The right choice for the ultimate yield!

LS ELECTRIC strives to maximize your profits in gratitude for choosing us as your partner.

Programmable Logic Control

Standalone Motion Controller

XGT Series

User Manual

XMC-E32A(/DC)
XMC-E16A(/DC)
XMC-E08A(/DC)
XMC-E32C(/DC)

Safety Instructions

- Read this manual carefully before installing, wiring, operating, servicing or inspecting this equipment.
- Keep this manual within easy reach for quick reference.

LS ELECTRIC

Before using the product …

For your safety and effective operation, please read the safety instructions thoroughly before using the product.

► Safety Instructions should always be observed in order to prevent accident or risk with the safe and proper use the product.

► Instructions are divided into “Warning” and “Caution”, and the meaning of the terms is as follows.

⚠ **Warning**	This symbol indicates the possibility of serious injury or death if some applicable instruction is violated.
⚠ **Caution**	This symbol indicates the possibility of severe or slight injury, and property damages if some applicable instruction is violated.

Moreover, even classified events under its caution category may develop into serious accidents relying on situations. Therefore we strongly advise users to observe all precautions properly just like warnings.

►The marks displayed on the product and in the user’s manual have the following meanings.

⚠ Be careful! Danger may be expected.

⚠ Be careful! Electric shock may occur.

► The user’s manual even after read shall be kept available and accessible to any user of the product.

Safety Instructions for Design Process

⚠ Caution

- **Design the analog input / output signal or pulse input / output line at least 100mm away from high voltage line or power line so that it is not affected by noise or magnetic field change.** It may cause malfunction due to noise.
- **If there is a lot of vibration in the installation environment, take measures to prevent direct vibration from being applied to the PLC.** It may cause electric shock, fire or malfunction.
- **If metallic dust is present in the installation environment, take measures to prevent metallic dust from entering the product.** It may cause electric shock, fire or malfunction.

Safety Instructions on Installation Process

⚠ Caution

- **Use PLC only in the environment specified in PLC manual or general standard of datasheet.** If not, electric shock, fire, abnormal operation of the product may be caused.
- **Before install or remove the module, be sure PLC power is off.** If not, electric shock or damage on the product may be caused.
- **Be sure that every module is securely attached after adding a module or an extension connector.** If the product is installed loosely or incorrectly, abnormal operation, error or dropping may be caused. In addition, contact failures under poor cable installation will be causing malfunctions as well.
- **Make sure that the I / O connector is securely fastened.** It may cause wrong input or output.

Safety Instructions for Wiring Process

⚠ Warning

- **Prior to wiring works, make sure that every power is turned off.** If not, electric shock or damage on the product may be caused.

⚠ Caution

- **Check rated voltages and terminal arrangements in each product prior to its wiring process.** Applying incorrect voltages other than rated voltages and misarrangement among terminals may cause fire or malfunctions.
- **Secure terminal screws tightly applying with specified torque.** If the screws get loose, short circuit, fire or abnormal operation may be caused. Securing screws too tightly will cause damages to the module or malfunctions, short circuit, and dropping.
- **Be sure to earth to the ground using Class 3 wires for PE terminals which is exclusively used for PLC**. If the terminals not grounded correctly, abnormal operation or electric shock may be caused.
- **Don't let any foreign materials such as wiring waste inside the module while wiring,** which may cause fire, damage on the product or abnormal operation.

Safety Instructions for Test-Operation and Maintenance

⚠ Warning

- **Don't touch the terminal when powered**. Electric shock or abnormal operation may occur.
- **Prior to cleaning or tightening the terminal screws, let all the external power off including PLC power.** If not, electric shock or abnormal operation may occur.

⚠ Caution

- **Do not make modifications or disassemble each module.**
  Fire, electric shock or abnormal operation may occur.
- **Prior to installing or disassembling the module, let all the external power off including PLC power.** If not, electric shock or abnormal operation may occur.
- **Keep any wireless equipment such as walkie-talkie or cell phones at least 30cm away from PLC**. If not, abnormal operation may be caused.

Safety Instructions for Waste Disposal

⚠ Caution

- **Product or battery waste shall be processed as industrial waste.** The waste may discharge toxic materials or explode itself.

Revision History

Version	Date	Remark	Revised position
V 1.0	2017.4	First Edition	-
V 1.1	2017.11	Added ‘speed unit’ and ‘filter time constant’ in encoder parameter	5-12, 18
		Added LS_ReadCamTableMasterPos function block	6-141~142
		Added G-Code of ‘G21’	9-12, 26
		Added Acceleration/Constant speed/Deceleration Operation flags	Appendix 1-6
		Added appendix chapter of ‘Using EtherCAT slaves from other companies’	Appendix 6
V1.2	2018.6	Setting range of encoder position filter constant is corrected	5-12~13, 7-24
		Added LS_OnOffCam function block	6-143~145
		Added LS_RotaryKnifeCamGen function block	6-146~148
		Added LS_CrossSealCamGen function block	6-149~151
		Revised TransitionMode about LS_MoveLinearTimeAbsolute and LS_MoveLinearTimeRelative	6-160, 6-162
		Added error code (0x1124, 0x1170~0x1179)	A 2-14, A2-16
V1.3	2018.12	Added XMC-E32C function block	Ch15, A7
V1.4	2019.5	Added XMC-E16A and XMC-E08A function blocks	Ch2, Ch6, Ch13
		Added wiring specifications	Chapter 3
		Set up Ethercat Network/Added Flags	Chapter 4
		Set up NC Spindles, Encoder Latch, Added Position Control Range Expansion	Chapter 5
		Added LS_OnOffCamEx, NC_RetraceMove, NC_BlockSkip, NC_DryRun, NC_ToolMode, NC_ReadToolMode, NC_MirrorImage, NC_SpindleControl, NC_BlockOptionalSkip, NC_ManualToolComp, NC_ChgSpindleGear, FILE_OPEN, FILE_CLOSE, FILE_WRITE, FILE_READ, FILE_SEEK	Chapter 6
		Added Parameters	Chapter 7
		Added NC Control Expansion Function	Chapter 9
		Added Built-in Cnet Function	Chapter 15
		Added Flags according to NC Control Expansion Function	Appendix 1
		Added Error Information according to Cnet, NC Control Expansion Function	Appendix 2
		Added the Explanation of Cnet Communications Code Words	Appendix 7
		Added EtherCAT Diagnostic Function	Appendix 8
V1.5	2020.7	Format and contents modification according to the change of company name(LSIS → LS ELECTRIC)	-

1.6	’23.01	● Modify Domain (www.lselectric.co.kr -> www.ls-electric.com) ● [Additional XMC OS V2.10 version-up functionality reflected] ● Added Expansion homing (LS_Home) ● Added Cross-coupled control(LS_CrossCoupledControlOn) ● Added Reset axis error 2(MC_Reset2) ● Added Total axis servo On/Off (MC_PowerAll) ● Added Total axis homing (MC_HomeAll) ● Added Total axis expansion homing (LS_HomeAll) ● Added Total axis prompt Stop (MC_StopAll) ● Added Total axis Stop (MC_HaltAll) ● Added Reset all axis error 2 (MC_Reset2All) ● Added Setting all axis current position (MC_SetPositionAll) ● Added Group Servo On/Off (MC_GroupPower) ● Added Read slave SDO data(timeout) (LS_ReadSDO2) ● Added Write slave SDO data(timeout) (LS_WriteSDO2) ● Added Master position loop control (LS_MasterPLoopControl) ● Added Cam Profile/Point Specification Extensions ● Added cam block setup function ● Supplement BufferMode Description ● Added Node Switch Operation Mode Description ● Add connection function description below EtherCAT settings ● Added Read Motion Information (MC_ReadMotionInfo) ● Added Read current Position (MC_ReadActualPosition) ● Added Read current Velocity (MC_ReadActualVelocity) ● Added Read current torque (MC_ReadActualTorque) ● Added Read command Position (MC_ReadCommandedPosition) ● Added Read command Velocity (MC_ReadCommandedVelocity) ● Added Read command torque (MC_ReadCommandedTorque) ● Added Expansion Gear operation (MC_GearInEx) ● Added Expanded position specified electronic gear operation (MC_GearInPosEx) ● Added Expansion Variable gear operation (LS_VarGearInEx) ● Added Expansion Variable positioning gear operation (LS_VarGearInPosEx) ● Added Read axis group parameter (LS_ReadGroupParameter) ● Added Write axis group parameter (LS_WriteGroupParameter) ● Added Axis group vel/acc override(MC_GroupSetOverride) ● Added Group current position setting (MC_GroupSetPosition2)	Cover, Chapter 15 Chapter 6, Chapter 8

- Supplemented the description (MC_GroupSetPosition)
- Added Flag Content (_AGxx_Disabled)
- Added Prepare to allow external TCP connections. (TCP Server)
- Added Perform TCP socket accept action.(TCP Server operation)
- Added Connects to a relative TCP port.(TCP Client operation) (SOCKET_TCPCONNECT)
- Added Receive data from a TCP socket (SOCKET_TCPRECV)
- Added Send data from a TCP socket (SOCKET_TCPSEND)
- Added Performs a request to create a UDP socket. (SOCKET_UDPCREATE)
- Added Receive data from a UDP socket (SOCKET_UDPRECV)
- Added Send data from a UDP socket (SOCKET_UDPSEND)
- Added Check information for each channel of socket service (SOCKET_SVCINFO)
- Added Initializes the receive buffer of a TCP or UDP socket. (SOCKET_BUFCLEAR)
- Added Close the socket service channel.(SOCKET_CLOSE)
- Added Read local Ethernet IP, SUBNET MASK, GATEWAY (L_NET_INFO)
- Added Local Ethernet IP, SUBNET MASK, GATEWAY settings (SET_IP)
- Added Feature Description (MoveLink)
- Added a description of the 3D arc interpolation function
- Added Coordinate System Operation Look Ahead Function Content
- Added Coordinate System Output Filter Feature Content
- Added contents of the maximum acceleration limit function for coordinate system operation.
- Added a description of the function of the terminology reverse instrumentation
- Added action method in case of coordinate system operation 0x20C7 error
- Added full NC2 channel functionality
- Added EtherCAT Multiframe Description — Chapter 4
- Supplemented Flag Description (_EC_MASTER_STATE, SLVxx_EC_STATE) — Appendix 1
- Added Error Code (0x10B3, 0x10B4, 0x10B5, 0x10B6, 0x10B7, 0x10B7, 0x3362, 0x36B3, 0x0E90~0E96, 0x1023, 0x2130, 0x0F64, 0x0F75, 0x1220, 0x2120~0x2126, 2110~0x2113) — Appendix 2

V1.7	'24.01	● Modify explanations of command in-position width ● Added contents of speed control operation mode ● Added contents of coordinate operation maximum allowance acceleration/deceleration. ● Added contents of drive absolute position error detection ● Added contents of override mode ● Added contents of interpolation operation blending angle limit ● Added contents of interpolation operation blending allowance angle ● Added contents of coordinate look ahead setting ● Add contents of coordinate system output filter time constants ● Added contents of rotation operation/linear operation conversion ratio ● Added explanations of axis input range ● Modify the command list missing ● Added explanations of MC stop command operation during homing ● Modify input/output type mistake of LS_ReadCamBlockProperty, LS_WriteCamBlockProperty ● Added PID, LINAC, SLINAC commands ● Added explanations of absolute coordinate system application condition during homing ● Added error code list	Chapter5 Chapter6 Chapter8 Appendix 2
V1.8	'24.06	● Changed quality warranty period	-

Thank you for purchasing PLC of LS ELECTRIC Co., Ltd.
Before use, make sure to carefully read and understand the User's Manual about the functions, performances, installation and programming of the product you purchased in order for correct use and importantly, let the end user and maintenance administrator to be provided with the User's Manual.

The User's Manual describes the product. If necessary, you may refer to the following description and order accordingly. In addition, you may connect our website (http://www.ls-electric.com/) and download the information as a PDF file.
Relevant User's Manuals

Title	Description
XG5000 User's Manual (for XGK, XGB)	XG5000 software user manual describing online function such as programming, print, monitoring, debugging by using XGK, XGB CPU.
XG5000 User's Manual (for XGI, XGR)	XG5000 software user manual describing online function such as programming, print, monitoring, debugging by using XGI, XGR CPU.
XGK/XGB Instructions & Programming User's Manual	User's manual for programming to explain how to use instructions that are used PLC system with XGK, XGB CPU.
XGI/XGR/XEC Instructions & Programming User's Manual	User's manual for programming to explain how to use instructions that are used PLC system with XGI, XGR, XEC CPU.
XGK CPU User's Manual (XGK-CPUA/E/H/S/U)	XGK-CPUA/CPUE/CPUH/CPUS/CPUU user manual describing about XGK CPU module, power module, base, IO module, specification of extension cable and system configuration, EMC standard.
XGI CPU User's Manual (XGI-CPUU/CPUH/CPUS)	XGI-CPUU/CPUH/CPUS user manual describing about XGI CPU module, power module, base, IO module, specification of extension cable and system configuration, EMC standard.
XGR Redundant Series User's Manual	XGR- CPUH/F, CPUH/T user manual describing about XGR CPU module, power module, extension drive, base, IO module, specification of extension cable and system configuration, EMC standard.
XG-PM User's Manual	XG-PM software user manual describing online function such as motion programing, monitoring, debugging by using Positioning Module.

The User's Manual is based on XMC-E32A.

1) The differences according to product name and figure, please check the chapter of specifications, function blocks, analog and serial interfaces.
2) This manual is based on the following versions.

XMC-E32A	XMC-E16A	XMC-E08A	XMC-E32C	XG5000
V1.40	V1.40	V1.40	V1.40	V4.28

◎ Table of Content ◎

Chapter 1 Overview

This user's manual describes the standard of motion controller, installation method, the method to use each function, programming and the wiring with external equipment.

1.1 Characteristics

The characteristics of motion controller are as follows. Motion controllers are used for overall control purposes that require the functions listed below.

(1) Various motion control function

It has various functions needed for motion control system such as position control, speed control etc.

(a) It supports various motion control commands.

1) It supports a number of function blocks.
2) It supports a number of motion function blocks compliant to PLCopen standards.
3) Motion programs are supported in the form of LD or ST by using XG5000.

(b) It can control actual/virtual axis of up to 32 axes, virtual axis of 4 axes. The actual axis of 32 axes can be set and operated as a virtual axis. EtherCAT I/O of 64 slaves, and supports digital input 8 points, digital output 16 points,, analog input 2 channels, analog output 2 channels and encoder input of 2 channels.

(c) Various sing-axis operations are available.

1) Position control
2) Velocity Control
3) Synchronous Control
4) Torque Control
5) Multi-axis simultaneous start

(d) Various multi-axis group operations are available.

1) Circular interpolation
2) Linear interpolation
3) Helical interpolation
4) Group homing / Changes group position

(e) Switching control in operation is available.

1) Position/Speed control switching
2) Position/Torque control switching
3) Speed/Torque control switching

(f) Cam Control is available.

By using XG5000, you can set profile data for cam control and use program commands to start and cancel cam operation, and to skip cam. Users can create up to 32 cam profiles.

(g) Various Homing Control Function.

As for a homing method, you can use Homing method supported by each servo drive model. (Refer to the instruction manual of each servo drive for more detailed homing methods and servo parameter settings.)

(h) For the Acceleration/Deceleration method, trapezoidal acceleration/deceleration and S-shaped acceleration/deceleration is supported, and S-shaped acceleration/deceleration can be implemented by setting jerk on a motion function block.

(2) Speed-up of execution of the motion program

Through realization of speed-up of processing at the time of start-up operation, the motion program set as main task can be performed at up to 0.5ms intervals.

In addition, there is no delay time between axes in Simultaneous start and interpolation start.

(3) Connection with the servo driver through EtherCAT*1

(a) Direct connection to servo drives of up to 32 units and EtherCAT I/O of up to 64 units can be achieved through EtherCAT.

(b) Since the connection between motion control module and servo drive is made using Ethernet cables. So wiring is simple.

(c) You can easily check and set up the servo driver information and parameter at the Motion Control module

(d) Max. Connection distance is 100m.

(4) Able to realize the absolute position system

You can realize the absolute position system just by connecting to the servo drive using the absolute position encoder and in case of ON/OFF, it can know the current position of the motor without homing. In order to use the absolute position encoder of the L7N/L7NH series servo drive, a battery must be installed.

(5) Easy maintenance

As retain registers, parameters, cam data and location data are stored within the motion controller, data can be stored without delay, and there is no limited number of writes.

(6) Self-diagnosis, monitoring and test are available with strong software package, XG5000.

(a) Monitoring function (Module & Servo driver)

(b) Trace function

(c) Trend function

(d) Reading and saving module program/parameter

(e) Reading and saving servo parameter

(f) Creation of CAM data

(g) Providing details about errors and the solution for it

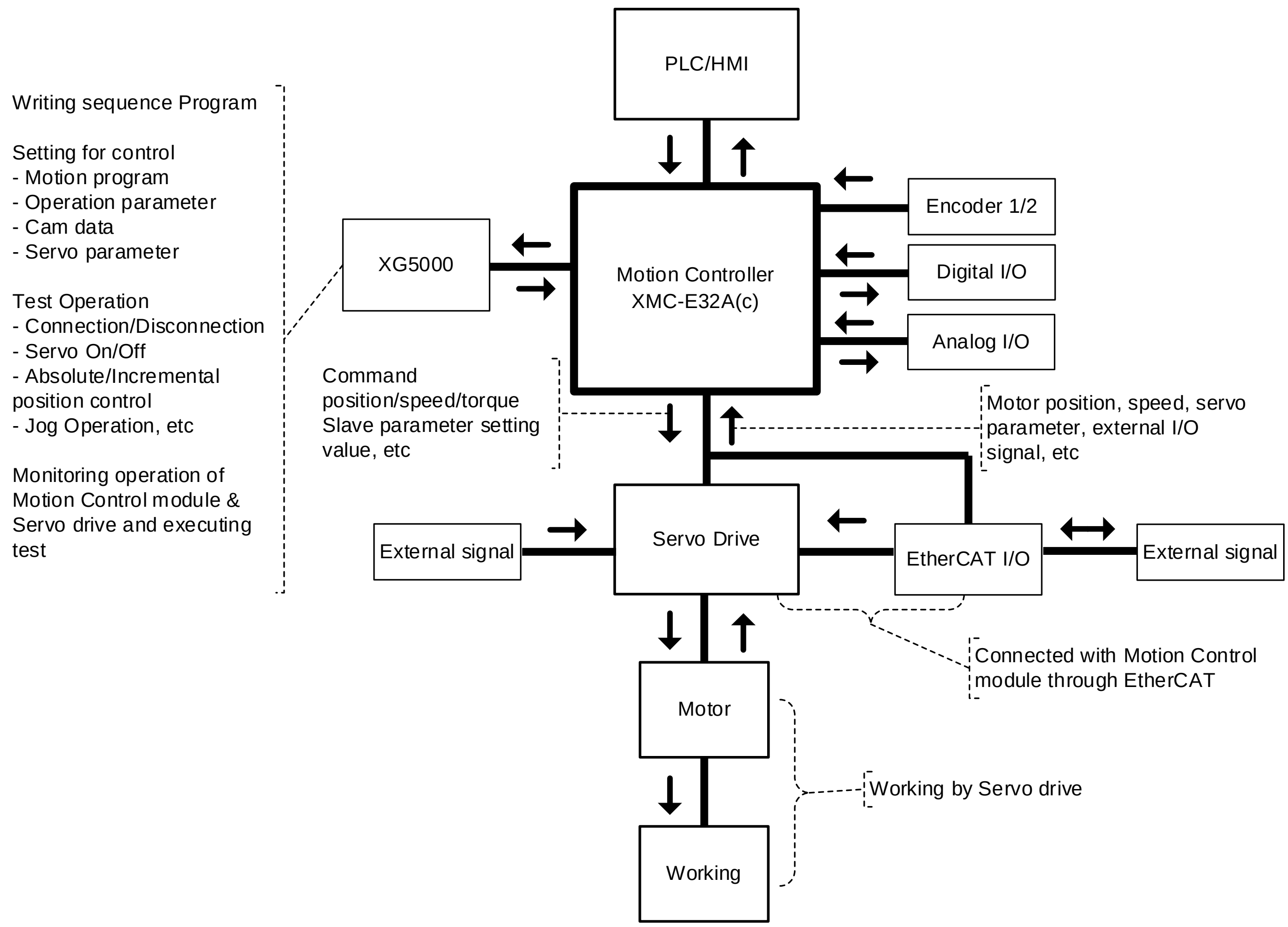
(h) Print function of various forms

Notes

1. What is EtherCAT?
   EtherCAT, Open Industrial Ethernet Solution, is developed by Beckhoff at 2002 and at 2003, November EtherCAT Technology Group (ETG-http://www.ethercat.org) is organized and it opens its technology. At 2005, February, that is authorized as IEC standard specification. Because of fast control speed and easiness for use and maintenance, it is widely used in the industrial field and conforming its performance
   In our positioning module, data communication with service driver is done with master-slave method through EtherCAT, and electric Ethernet Cable is used.

1.2 Signal Flow of Motion Controller

The flow of system using the motion controller is as follows.



1.3 Function Overview of Motion Controller

Describe Representative functions of motion controller (Coordinate & Linear Interpolation, Circular Interpolation & Stop) briefly.

1.3.1 Position Control

Execute positioning control for the designated axis from starting position (current position) to goal position. (the position to move to)

(1) Control by Absolute coordinates

(a) Execute positioning control from starting position to goal position designated in motion function block.

(b) Position control is executed based on position (the origin position) specified in the homing.

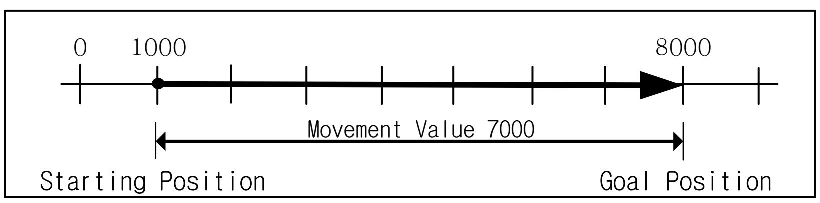
(c) Moving direction is decided by starting position and goal position.

- ■ Starting Position < Goal Position : Forward Positioning Operation
- ■ Starting Position > Goal Position : Reverse Positioning Operation

[Example]

- ■ Starting Position : 1000
- ■ target Position : 8000

Value of Forward movement is 7000 (7000=8000-1000)

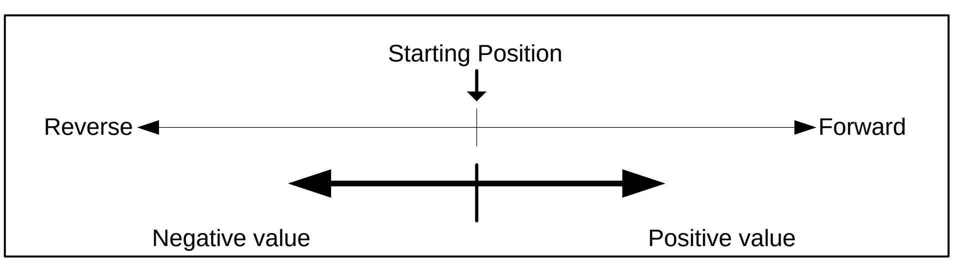


(2) Control by Incremental Coordinates

(a) Execute positioning control from starting position as much as goal movement value.
The difference from absolute coordinates control is that the goal position is movement value, not position value.

(b) Moving direction depends on sign of movement value.

- ■ Positive value (+ or 0) : Positioning operation with forward direction
- ■ Negative value (-) : Positioning operation with reverse direction

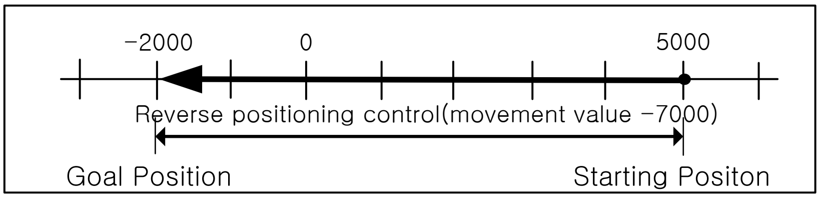


[Example]

■ Starting Position : 5000

■ Target Position : -7000

In this condition, it moves reversely and stops at -2000.



1.3.2 Interpolation Control

(1) Linear Interpolation Control

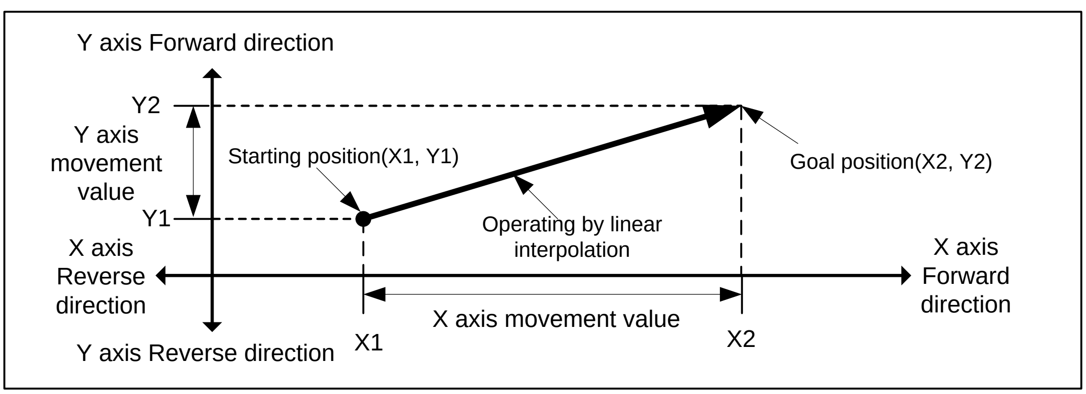
Execute Linear interpolation control with designated axis at start position. (Current position)
Combination of interpolation axis is unlimited and maximum 10 axes linear interpolation control is available.

(a) Linear interpolation by absolute coordinates

1) Execute Linear interpolation from starting position to goal position designated by positioning data.
2) Positioning control is executed based on home position designated in homing.
3) Movement direction is designated by starting position & goal position of each axis.

■ Starting position < Goal position : Positioning operation with forward direction

■ Starting position > Goal position : Positioning operation with reverse direction

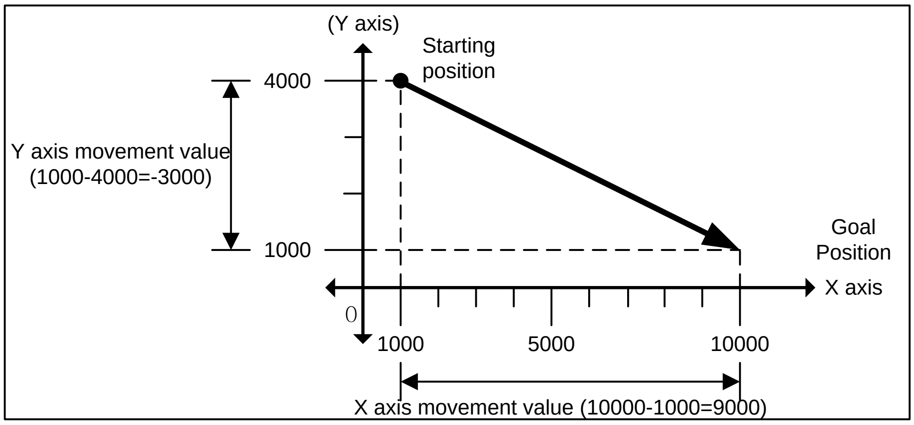


[Example]

■ Start position(1000, 4000),

■ Target position (10000, 1000)

The action is as follows in the condition above



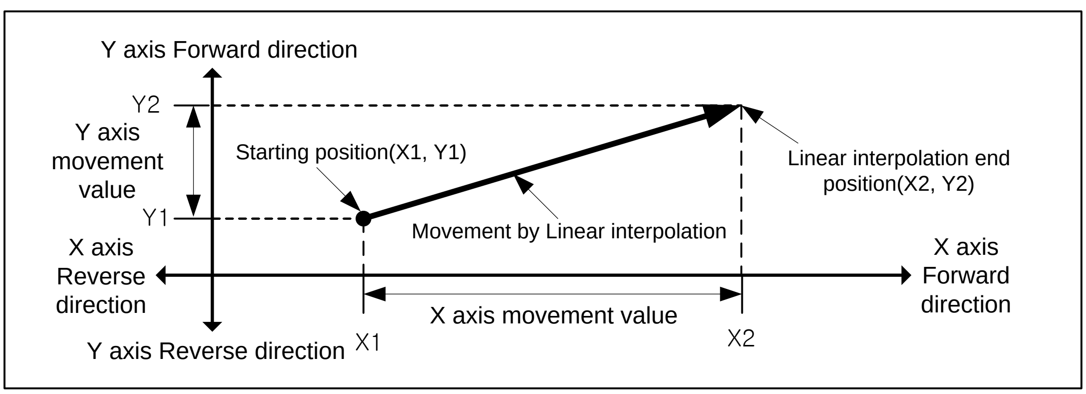
(b) Linear Interpolation by incremental coordinates

1) Linear interpolation is performed from the start address to the position including the target movement direction and movement amount for each axis.

2) Moving direction depends on movement value is positive or negative.

■ When the sign of movement distance is positive (+ or no sign): Positioning operation in forward direction (star ting position increase direction)

■ When the sign of movement distance is negative (-): Positioning operation in reverse direction (starting posit ion decrease direction)

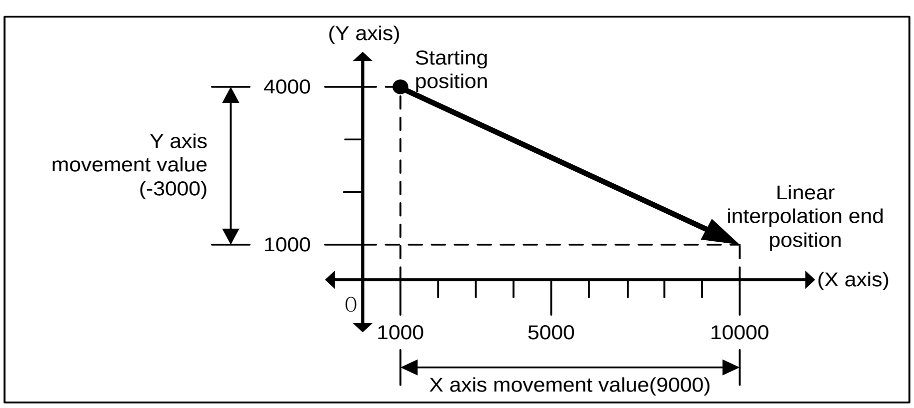


[Example]

■ Start position (1000, 4000),

■ Target position (9000, -3000)

The action is as follows in the condition above

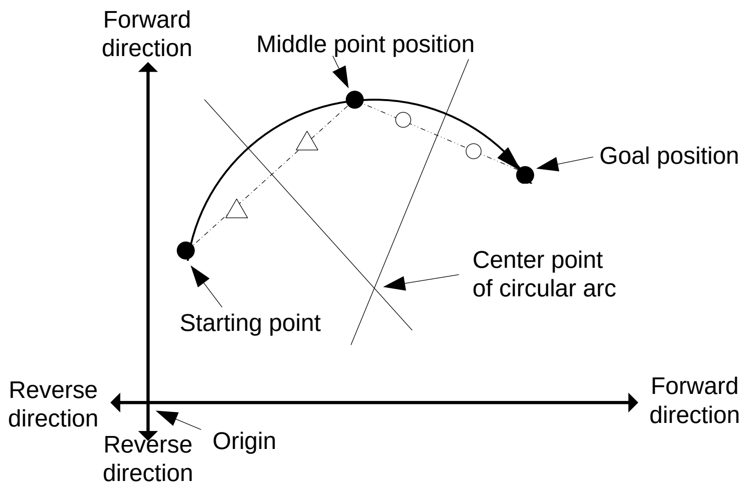


(2) Linear Interpolation Control

Execute interpolation operation along the trace of circle with 2 axes in forward direction that already designated for each axis. Circular interpolation has 3 types according to auxiliary point, Middle point method passing auxiliary point, Center point method using auxiliary point as center of circle and Radius method using auxiliary point as radius of circle.
The combination of 2 axes for circular interpolation is unlimited. Any of the two axes from the actual axes (1-axis to 32-axis) or virtual axes (1-axis to 36-axis) can be used.)

(a) Circular interpolation with middle point designation form.

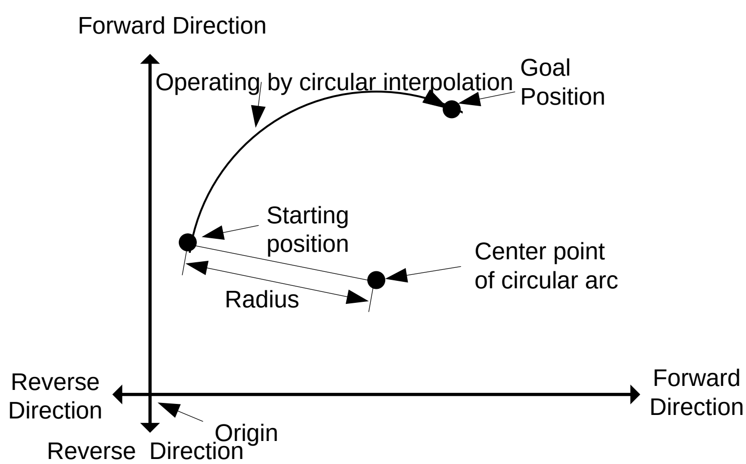
1) Starts operating at starting position and executes circular interpolation through the designated middle point.
2) There will be a circular arc whose center point is crossing point of perpendicular bisection between starting position and middle point or middle point and goal position.



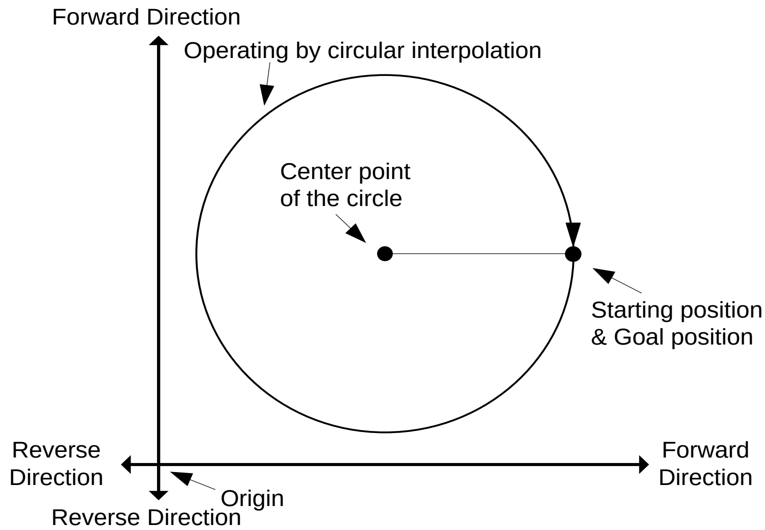
3) Movement direction is decided automatically depends on set target position and auxiliary point of circular interpolation.

(b) Circular interpolation with center point designation form

1) Starts operating from starting position and execute circular interpolation along trace of circle that has distance from starting point to designated center point as radius.



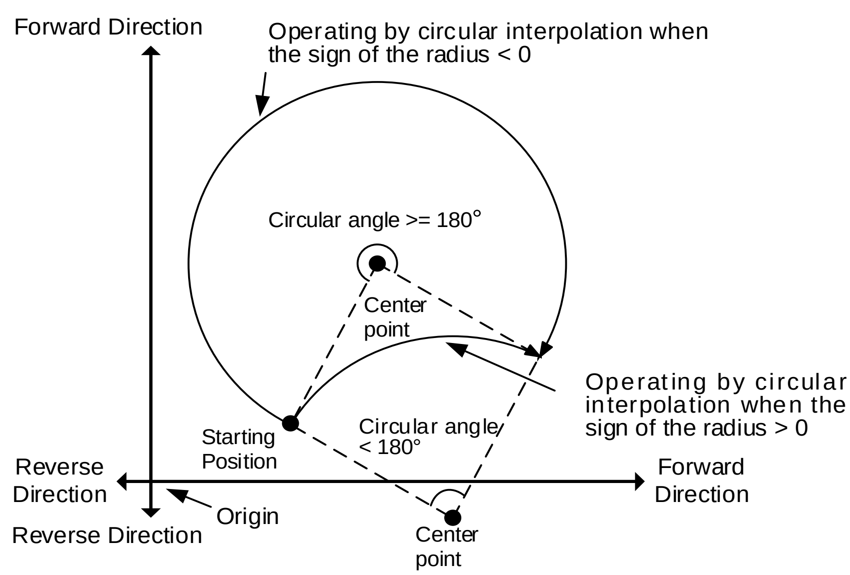
2) If the goal position is same as starting position, it is available to have an operation like a circle that has distance from starting point to auxiliary point as its radius.



3) The direction of movement is determined according to the selection of paths (CW, CCW) to be set at the time of motion function block.

(c) Circular interpolation with radius designation form

1) Starts operating from starting position and execute circular interpolation along trace of circular arc that has value designated in auxiliary point of main axis as it radius. An arc whose central point varies depending on the sign of the radius is drawn.



2) In circular interpolation of radius specification method, the target position cannot be set the same as starting position.

3) The operational directions and the size of the arc are determined by the path selection (CW, CCW) of circular interpolation commands and the sign of the radius.

(3) Helical interpolation Control

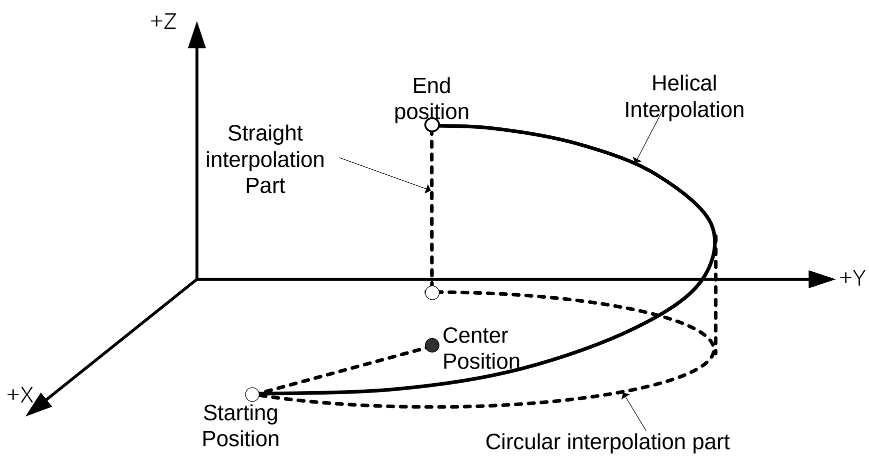
(a) Moves along the designated trace of circular arc depending on circular arc interpolation setting and executes Linear interpolation synchronously.

(b) There is no limit to the combination of axes to be used in helical interpolation, and three axes from actual axis (1 axis to 32 axes) or virtual axis (1 axes to 36 axes) are used. The three axes used for helical interpolation are [Axis Setting 1], [Axis Setting 2], and [Axis Setting 3] of the axis group. The corresponding axis settings are matched to the X, Y, and Z axes.

(c) Helical interpolation control is possible using the MC_MoveCircularAbsolute and MC_MoveCircularRelative commands.

(d) The starting position of the helical interpolation control in the figure below is the command position when the command is executed.

(e) The input variable EndPoiont of the MC_CircularAbsolute and MC_MoveCircularRelative commands sets the end position of the figure below. EndPoint[0] corresponds to the X axis, EndPoiont[1] corresponds to the Y axis, and EndPoiont[2] corresponds to the Z axis coordinates.



1.3.3 Speed Control

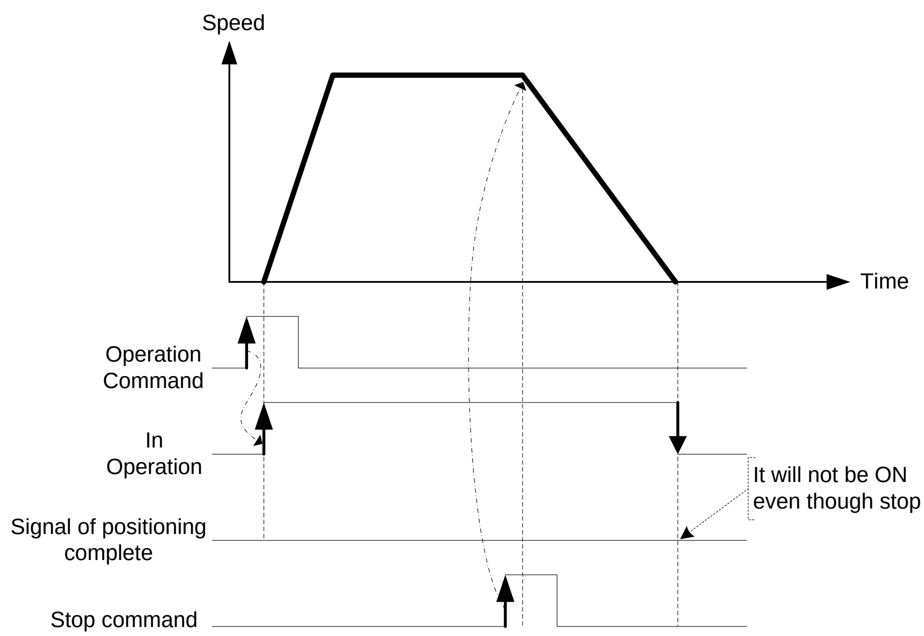
(1) Execution is made by speed control commands, and the operation proceeds at the established rate until buffer commands are executed, or stop commands are entered.

(2) Speed control has forward operation and reverse operation.

(a) Forward direction: In case of velocity > 0 and forward direction, or velocity < 0 and reverse direction

(b) Reverse direction: In case velocity > 0 and reverse direction, or velocity < 0 and reverse direction.

(3) Operation timing



1.3.4 Torque Control

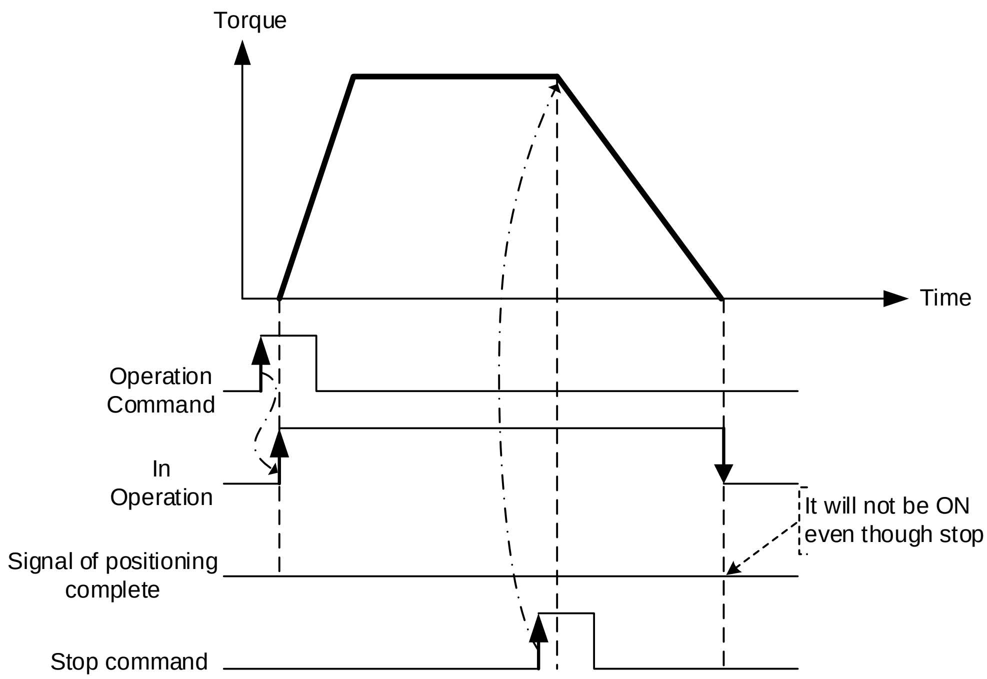
(1) The execution is made by the torque control command, and the operation is done in the set torque until the buffer c ommand or stop command is entered.

(2) Torque control includes forward operation and a reverse operation..

(a) Forward direction: In case of torque > 0 and forward direction, or torque < 0 and reverse direction

(b) Reverse direction: In case torque > 0 and reverse direction, or torque < 0 and reverse direction.

(3) Operation timing



Chapter 2 Specifications

2.1 General specifications

The following table shows the general specification of XGT series.

<table>
<tr><th>No.</th><th>Item</th><th colspan="4">Specifications</th><th>Relevant specifications</th></tr>
<tr><td>1</td><td>Ambient temperature</td><td colspan="4">0 ~ 55 ℃</td><td>-</td></tr>
<tr><td>2</td><td>Storage temperature</td><td colspan="4">-25 ~ +70 ℃</td><td>-</td></tr>
<tr><td>3</td><td>Ambient humidity</td><td colspan="4">5 ~ 95%RH (Non-condensing)</td><td>-</td></tr>
<tr><td>4</td><td>Storage humidity</td><td colspan="4">5 ~ 95%RH (Non-condensing)</td><td>-</td></tr>
<tr><td rowspan="8">5</td><td rowspan="8">Vibration resistance</td><td colspan="3">In case of occasional vibration</td><td>-</td><td>-</td></tr>
<tr><td>Frequency</td><td>Acceleration speed</td><td>Amplitude</td><td>Number</td><td rowspan="7">IEC61131-2</td></tr>
<tr><td>5 ≤ f < 8.4Hz</td><td>–</td><td>3.5mm</td><td rowspan="6">10 times for each direction of X, Y and Z</td></tr>
<tr><td>8.4 ≤ f ≤ 150Hz</td><td>9.8m/s^2{1G}</td><td>–</td></tr>
<tr><td colspan="3">In case of continuous vibration</td></tr>
<tr><td>Frequency</td><td>Acceleration speed</td><td>Amplitude</td></tr>
<tr><td>5 ≤ f < 8.4Hz</td><td>–</td><td>1.75mm</td></tr>
<tr><td>8.4 ≤ f ≤ 150Hz</td><td>4.9m/s^2{0.5G}</td><td>–</td></tr>
<tr><td>6</td><td>Shock resistance</td><td colspan="4">Maximum shock acceleration: 147㎨{15G}
Duration: 11ms
Pulse waveform: Half-sine (3 times for each direction of X, Y and Z)</td><td>IEC61131-2</td></tr>
<tr><td rowspan="5">7</td><td rowspan="5">Noise resistance</td><td>Square wave impulse noise</td><td colspan="3">AC: ± 1,500 V
DC:± 900 V</td><td>LS ELECTRIC standard</td></tr>
<tr><td>Electrostatic discharge</td><td colspan="3">Voltage : 4kV (contact discharging)</td><td>IEC61131-2
IEC61000-4-2</td></tr>
<tr><td>Radiated electromagnetic field noise</td><td colspan="3">80 ~ 1,000 ㎒, 10 V/m</td><td>IEC61131-2,
IEC61000-4-3</td></tr>
<tr><td rowspan="2">Fast transient /Burst noise</td><td>Classification</td><td>Power</td><td>Digital/analog I/O, Communication interface</td><td rowspan="2">IEC61131-2
IEC61000-4-4</td></tr>
<tr><td>Voltage</td><td>2kV</td><td>1kV</td></tr>
<tr><td>8</td><td>Operating atmosphere</td><td colspan="4">Free from corrosive gases and excessive dust</td><td>-</td></tr>
<tr><td>9</td><td>Altitude</td><td colspan="4">Up to 2,000m</td><td>-</td></tr>
<tr><td>10</td><td>Pollution degree</td><td colspan="4">Less than equal to 2</td><td>-</td></tr>
<tr><td>11</td><td>Cooling</td><td colspan="4">Air cooling</td><td>-</td></tr>
</table>

Notes

IEC (International Electrotechnical Commission)

An international nongovernmental organization which promotes internationally cooperated standardization in electric/electronic field, publishes international standards and manages applicable estimation system related with.

2) Pollution degree

An index indicating pollution degree of the operating environment which decides insulation performance of the devices. For instance, Pollution degree 2 indicates the state generally that only non-conductive pollution occurs. However, this state contains temporary conduction due to dew produced.

2.2 Power specification

The following table shows the power specifications of motion controller.

<table>
<tr><th colspan="3" rowspan="2">Item</th><th colspan="4">Specifications</th></tr>
<tr><th colspan="2">AC Power</th><th colspan="2">DC Power</th></tr>
<tr><td rowspan="8">Input</td><td colspan="2">Rated input voltage</td><td colspan="2">AC100V ~ AC240V</td><td colspan="2">DC19.2 ~ 28.8V</td></tr>
<tr><td colspan="2">Input frequency</td><td colspan="2">50/ 60 ㎐</td><td colspan="2">–</td></tr>
<tr><td colspan="2" rowspan="2">Input current</td><td colspan="2">0.7A or less (AC110V)</td><td colspan="2" rowspan="2">1.6A or less</td></tr>
<tr><td colspan="2">0.4A or less (AC120V)</td></tr>
<tr><td colspan="2">Inrush current</td><td colspan="2">120Apeak or less(AC240V, Phase 90 degree)</td><td colspan="2">100Apeak or less (DC28.8V)</td></tr>
<tr><td colspan="2">Leakage current</td><td colspan="2">3mA or less</td><td colspan="2">3mA or less</td></tr>
<tr><td colspan="2">Efficiency</td><td colspan="2">65% or more</td><td colspan="2">65% or more</td></tr>
<tr><td colspan="2">Permitted momentary power failure</td><td colspan="2">10ms or less</td><td colspan="2">10ms or less</td></tr>
<tr><td rowspan="8">Output</td><td rowspan="3">Output voltage</td><td>Voltage</td><td>Output voltage ripple range</td><td>Current</td><td>Output voltage ripple range</td><td>Current</td></tr>
<tr><td>+5V</td><td>4.90~5.20V</td><td>4A</td><td>4.90~5.15V</td><td>4A</td></tr>
<tr><td>+24V</td><td>21.1~26.9V</td><td>0.4A</td><td>–</td><td>–</td></tr>
<tr><td rowspan="3">Ripple & Noise</td><td>Output</td><td>Ripple</td><td>Noise</td><td>Ripple</td><td>Noise</td></tr>
<tr><td>+5V</td><td>100mVpp or less</td><td>200mVpp or less</td><td>100mVpp or less</td><td>200mVpp or less</td></tr>
<tr><td>+24V</td><td colspan="2">400mVpp or less</td><td colspan="2">–</td></tr>
<tr><td rowspan="2">Over current protection</td><td>+5V</td><td colspan="2">4.4A or higher</td><td colspan="2">4.4A or higher</td></tr>
<tr><td>+24V</td><td colspan="2">0.44A or higher</td><td colspan="2">–</td></tr>
</table>

For protection of the power supply, you are recommended to use the power supply with the maximum of 4A fuse.

Notes

1) Allowable instantaneous interruption time
It is the time to maintain the normal output voltage (normal operation) on the condition that the input voltage of (AC110/220V) is lower than the maximum/minimum (AC85/170V).

2) Over current protection
(1) If the current over the standard flows in DC5V, DC24V circuit, the over current protection device shutdowns the circuit to stop the system.
(2) If overcurrent occurs, after removing the causes such as shortage of current capacity, short circuit, etc., restart the system.

3) Over voltage protection
If the voltage over the standard is applied in DC5V circuit, the over voltage protection device shutdowns the circuit to stop the system.

4) Use UL approved power supply.
Use a UL certified product for the power supply. Use a power supply that meets Class 2 or LVLC (Limited Voltage Limited circuit).

2.3 Performance specifications

The following table shows the Performance specifications of motion controller.

2.3.1 Performance specifications

Item			XMC-E32A	XMC-E16A	XMC-E08A
			XMC-E32C	-	-
Operation Method			Main task/Periodic task: Fixed cyclic operation, reiterative operation initialization task: Only once at the time of entering the RUN		
Control cycle			Main Task Time: 0.5ms, 1ms, 2ms, 4ms Periodic task time: Multiple setting of main task		
I/O control method			Synchronism with main task cyclic (Refresh method)		
Program language			Ladder Diagram(Function block), Structured Text, G-Code		
Number of instruction	Operator		18		
	Basic function		202		
	Basic function block		174		
	Special function block		97		
Calculation processing Speed	Basic		6.25ns or more (General point/coil)		
	MOVE		5ns or more (Word type)		
	Arithmetic		30ns or more (Word type)		
Program Memory	Quantity		Max. 256		
	Capacity		10MB(Motion program,), 10MB(NC program)		
Data Memory	Automatic variable (A)		4,096KB(Retain setting available up to 2,048KB)		
	Input variable(I)		16KB		
	Output variable(Q)		16KB		
	Direct variable (M)		2,048KB(Retain setting available up to 1,024KB)		
	Flag variable	F	128KB		
		K	18KB		
		U	1KB		
		L*Note 3)	22KB		
		N*Note 3)	49KB		
Timer			No limit in points Time range: 0.001~ 4,294,967.295sec(1,193hour)		
Counter			No limit in points Counting range: 64 bit range		
Configuration of program			Initial program, Main task program, Periodic task program, NC program		
Operation mode			RUN, STOP		
Restart mode			Cold, Warm		
Self-diagnostics function			Cyclic error monitoring, time share over detection of task program, memory abnormal, power abnormal, etc.		
Data retention in case of power failure			Retain area setting in basic parameters or Retain setting when setting variables		

Item		XMC-E32A XMC-E32C	XMC-E16A -	XMC-E08A -
Control Axis/Slave	Real/Virtual Axes	Axis 32	Axis 16	Axis 8
	Dedicated Virtual Axis	Axis 4	Axis 2	Axis 1
	Slave (Including Real Axes)	64 slave	32 slave	16 slave
Module types		EtherCAT (CoE: CANopen over EtherCAT, FoE: File Access over EtherCAT)		
Communication Period		0.5ms, 1ms, 2ms, 4ms (Same as the Main Task Period)		
Servo drive support		Servo drive to support EtherCAT CoE		
Control unit		pulse, mm, inch, degree		
Control method		Position, Velocity, Torque (Servo drive support), Synchronous, Interpolation Control		
Position address range		± LREAL, 0		
Speed range		± LREAL, 0		
Torque unit		Rated torque % designation		
Acc./Dec. processing		Trapezoid type, S-type (Setting by specifying Jerk at a function block)		
Rage of Acc./Dec.		+LREAL*Note 1),0		
Manual operation		JOG Operation		
Cam Operation		32 profiles/ 32,768 points	16 profiles/ 16,384 points	8 profiles/ 8,192 points
Absolute position system		Available (When using an absolute encoder type servo drive)		
Digital I/O	Digital input	8 points		
	Digital output	16 points (Transistor)		
	Encoder input	2 channels Max. input: 500Kpps Input method: Line drive, Voltage input Input type: CW/CCW, Pulse/Direction, Phase A/B		
Analog input/output Note 2)	Analog input	2 channels Voltage input range: -10~10V / 0~10V / 1~5V / 0~5V Current input range: 4~20mA / 0~20mA Max. resolution: 14bit(1/16,000)		
	Analog output	2 channels Voltage Output range: Output range: -10~10V / 0~10V / 1~5V / 0~5V Max. resolution: 14bit(1/16,000)		
Serial communication note3)	Port	RS-232C : one port, RS-485 one port		
	Protocols supported	XGT dedicated protocol, Modbus protocol, User-defined protocol, LS bus (inverter protocol) support		
Coordinate system function (Robot)		Cartesian, Delta		
SD memory	Memory type	Micro SDHC		
	File system	FAT32		
	Maximum Capacity	32GB (Memory over 8GB can use only 8GB of overall area)		
	Service	Program back-up/restoration, Booting operation, Data logging		

Item		XMC-E32A	XMC-E16A	XMC-E08A
		XMC-E32C	-	-
Ethernet	Communication speed	Auto/10Mbps/100Mbps		
	Communication port	1 port		
	Communication distance	Max. 100m between nodes		
	Service	Loader service (XG5000) LS ELECTRIC protocol supported (XGT, MODBUS TCP) FTP server: Function to read/write files of the SD memory card from other devices SNTP client		
USB	Characteristics	USB 2.0, 1 channel		
	Service	The loader service (XG5000) is supported		
Error indication		Indicated by LED		
Weight		790g		

Note1) LREAL range: 2.2250738585072e-308 ~ 1.79769313486232e+308
Long real number (+LREAL) positive range: $0 < x \leq 1.79769313486232e+308$
Note2) the analog function is supported only by analog-type products (XMC-E32A/E16A/E08A).
Note3) the serial communication function is supported only by communication-type products (XMC-E32C).

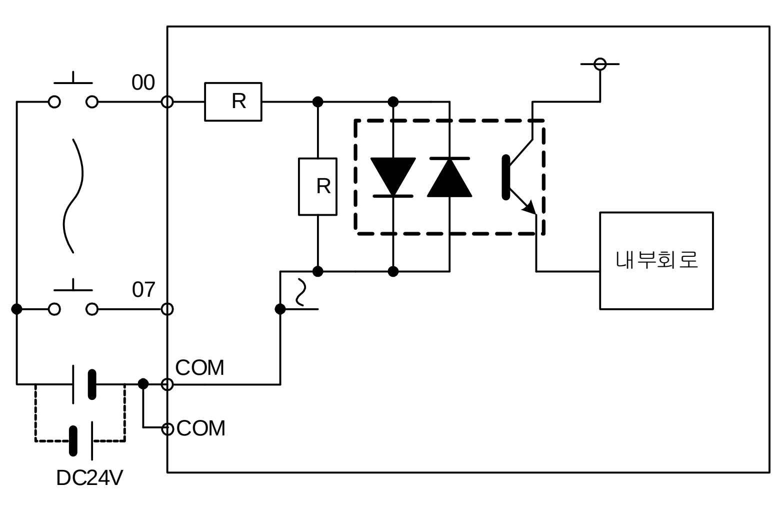
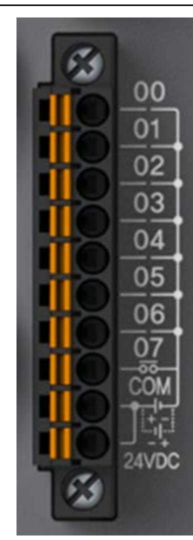
2.3.2 EtherCAT Communication specifications

Item	Specifications
Communication protocol	EtherCAT
Support specification	CoE(CANopen over EtherCAT)
Physical layer	100BASE-TX
Communication speed	100Mbps
Topology	Daisy Chain
Communication cable	Cat. 5 STP(Shielded Twisted-pair) cable
Number of maximum slave	64(Able to mapping Max. 32 drive to motion axis)
Communication period	0.5ms/1ms/2ms/4ms
Synchronous Jitter	Less than 1 μs
Synchronous communication	PDO(Process Data Object) Mapping through CoE
Non-synchronous communication	SDO(Service Data Object) communication through CoE
Communication setting	Set the communication configuration using XG5000
Maximum transmission distance	100m
Indicates the communication status	LED

2.3.3 Internal input/Output Specification

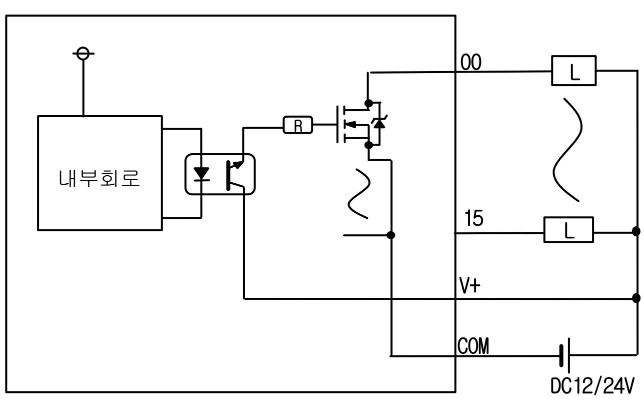
(1) Input specifications (source/sink type)

Item		Specifications
Input point		8 points
Insulation method		Photo coupler insulation
Rated input voltage		24V
Rated input current		About 5mA
Operation voltage range		DC20.4V~28.8V (within ripple rate 5%)
On voltage/On current		DC19V or more / 3mA or more
Off voltage/Off current		DC6V or less / 1mA or less
Input resistance		About 4.7 ㏀
Response time	Off→On	0.5/1/3/5/10/20/70/100 ㎳ (set by I/O parameter) Initial value: 3 ㎳
	On→Off	
Insulation voltage		AC560Vrms/3 Cycle (Altitude 2,000m)
Insulation resistance		Insulation resistance 10 ㏁ or more
Common method		8 point/COM

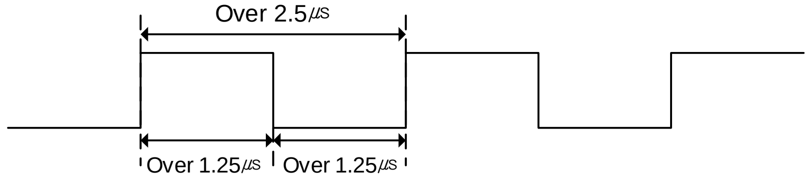
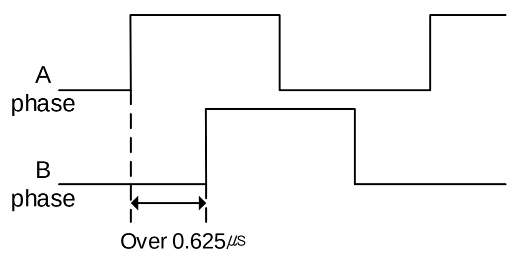
Circuit configuration	No.	Point	Type
	00	%IX0.0.0	
	01	%IX0.0.1	
	02	%IX0.0.2	
	03	%IX0.0.3	
	04	%IX0.0.4	
	05	%IX0.0.5	
	06	%IX0.0.6	
	07	%IX0.0.7	
	COM	-	
	COM	-	

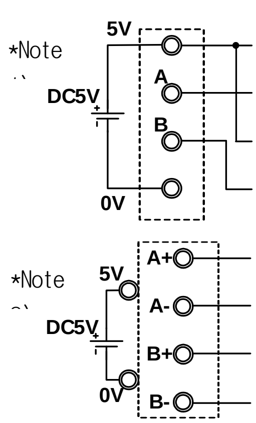
(2) Output specifications (sink type)

Item		Specifications
Output point		16 points
Insulation method		Photo coupler insulation
Rated load voltage		DC 12V / 24V
Used load voltage range		DC10.2V~26.4V
Max. load current		0.5A / 1 point, 2A / 1COM
Off leakage current		0.1mA or less
Maximum inrush current		4A/ 10ms or less
Maximum voltage drop(On)		DC 0.4V or less
Surge absorber		Zener diode
Response time	Off→On	1ms or less
	On→Off	1ms or less(Rated load, resistive load)
Common method		16 point / 1COM
External power	Voltage	DC12/24V±10% (Ripple voltage 4Vp-p or less)
	Current	10mA or less (DC24V connection)

Circuit configuration	No.	Point	Type
	00	%QX0.0.0	
	01	%QX0.0.1	
	02	%QX0.0.2	
	03	%QX0.0.3	
	04	%QX0.0.4	
	05	%QX0.0.5	
	06	%QX0.0.6	
	07	%QX0.0.7	
	08	%QX0.0.8	
	09	%QX0.0.9	
	10	%QX0.0.10	
	11	%QX0.0.11	
	12	%QX0.0.12	
	13	%QX0.0.13	
	14	%QX0.0.14	
	15	%QX0.0.15	
	V+	-	
	COM	-	

2.3.4 Input specifications of Encoder

Item	Specifications	
Input voltage	5V (3V ~ 6V)	In accordance with RS-422A Line Driver Level
input current	2 ㎃ to 7.5 ㎃	
Min. On guarantee voltage	2.5V	
Max. Off guarantee voltage	1V	
Input pulse	1) Pulse width  2) Phase difference 	When A phase input pulse is ahead of B phase input pulse : Position value increases When B phase input pulse is ahead of A phase input pulse : Position value decreases

Circuit configuration	Contact Point	Signal name	Type
	ENC1 A+	Encoder A+ input	
	ENC1 A-	Encoder A- input	
	ENC1B+	Encoder B+ input	
	ENC1B-	Encoder B- input	
	ENC2 A+	Encoder2 A+ input	
	ENC2 A-	Encoder2 A- input	
	ENC2 B+	Encoder2 B+ input	
	ENC2 B-	Encoder 2 B- input	

Caution

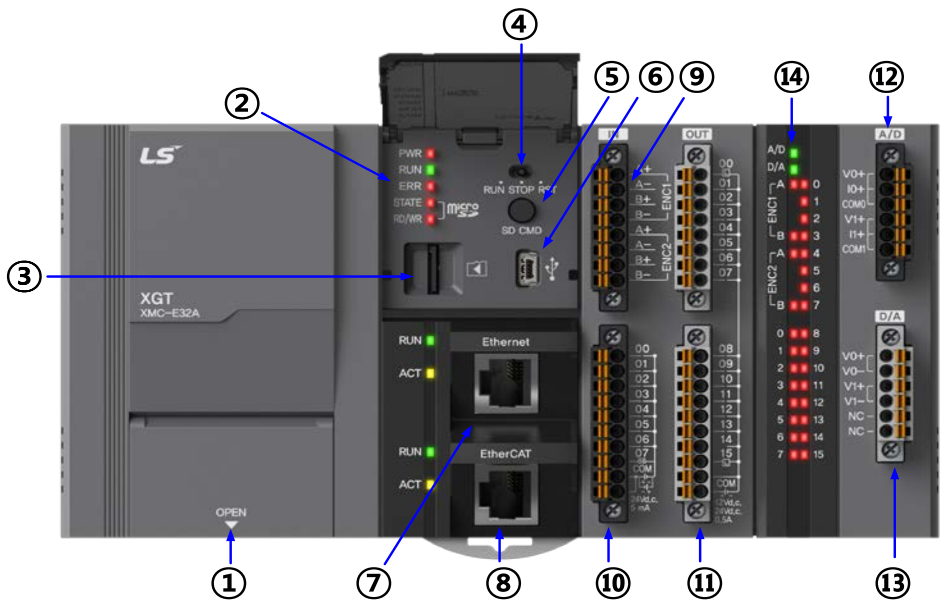
Note 1 : Encoder of 5V voltage output type(Open collector) When using 12V, 24V type MPG, change the input voltage from 5V to 12V or 24V and in case of 12V, connect 910Ω resistor to ENC1 A+(pin 1), ENC1 B+ (pin3), in case of 24V, 2.4㏀ resistor, before connecting the power source (adding PULL-UP resistor is needed)

Note 2 : Encoder of 5V voltage output type(Line driver)

2.4 Part names

2.4.1 Part names

(1) XMC-E32A



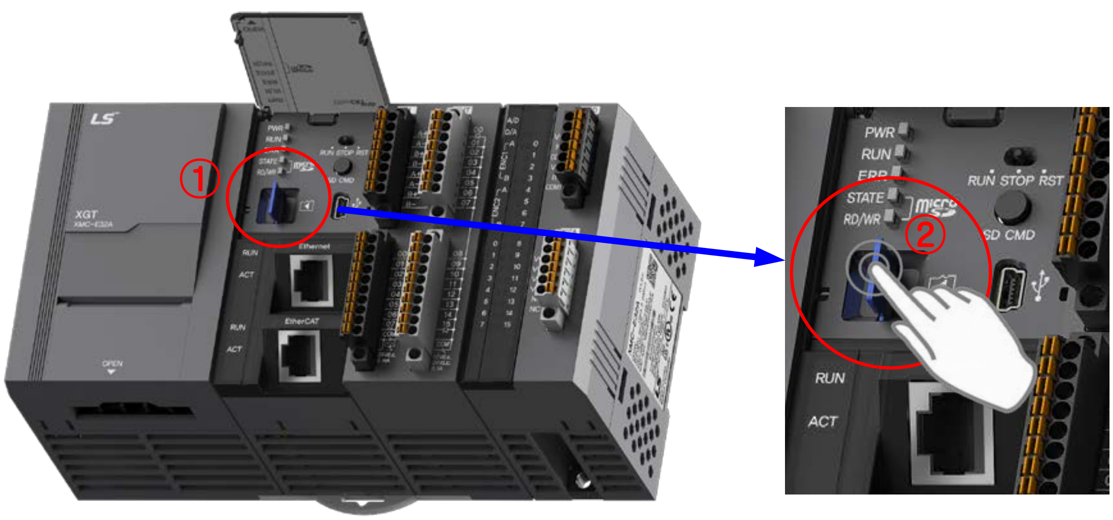
No.	Name	Content
①	Power input	AC 110/220V power input, LG terminal, DC24V output
②	Status display	Displays the motion controller's operation mode. - PWR(Red light on): The power is supplied - RUN(Green light on): During RUN mode - ERR(Flickering Red light): Occurrence of errors during operation - STATE (Red light on/Flickering Red light): When the SD card is installed, the red light is turned On; when the SD card error occurs, the red light is flickering. - RD/WR(Flickering Red light): During SD memory reads or writes
③	SD card connector	Connector with the SD memory card
④	Mode switch	Sets the motion controller's operation mode. - RUN: Program Execution - STOP: program stop - RST: Program's operation is reset.
⑤	SD card command button	Press to button less than 3 second. - Additional function(back-up, recover, compare) operation in according to script setting Press to button over 3 second. - SD Power On/Off Pressing to button and power on - Boot operation
⑥	USB connector	Port to access to XG5000
⑦	Ethernet connector	Port to communicate Ethernet
⑧	EtherCAT port	Port to communicate EtherCAT

No.	Name	Content
⑨	Encoder input connector	Connector that accepts encoder input signal
⑩	Digital input connector	Connector that accepts digital input signal
⑪	Digital output connector	A connector that outputs a digital output signal
⑫	Analog input connector	Connector that accepts analog input signal
⑬	Analog output connector	A connector that outputs analog output signal
⑭	Input/output display	Digital input/oupt, Analog input/output, Encoder input

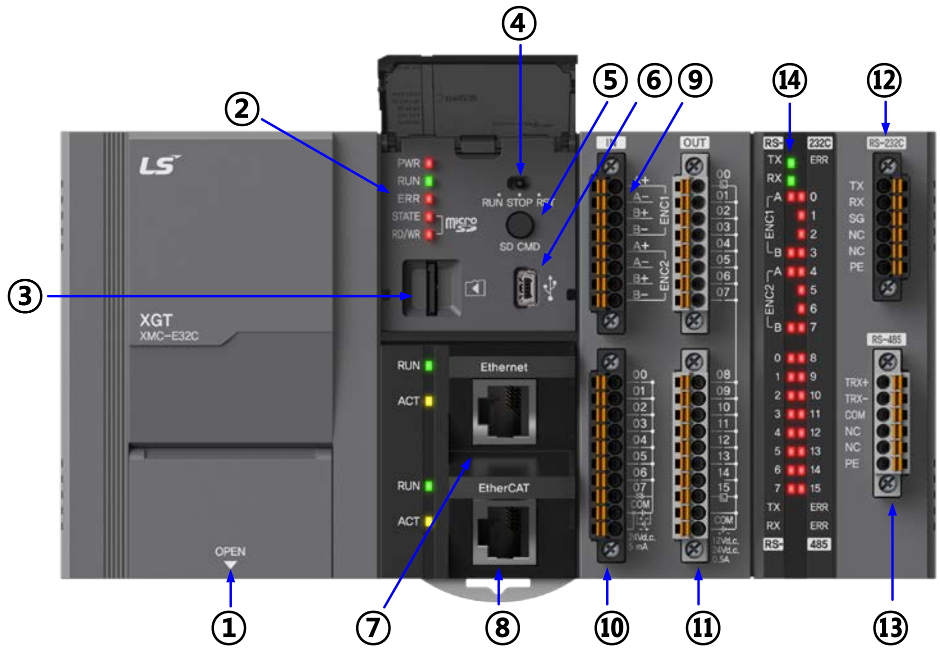
※ With exception of the outer part where the product name is shown, the XMC-E16A and XMC-E08A are identical to the XMC-E32A.

Caution

After inserting the SD memory into the SD memory slot entrance (display①) as shown below, press the middle part of SD memory (②) and install it completely. If the SD memory is not inserted correctly in the SD memory slot (direction reversed, up / down (left / right) tilting or twisting) It may not operate normally.



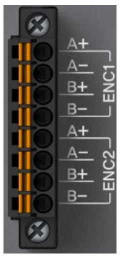
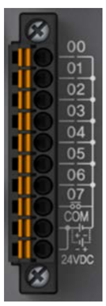
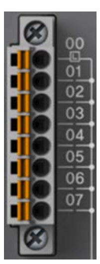
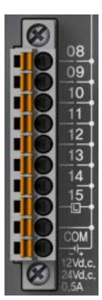
(2) XMC-E32C

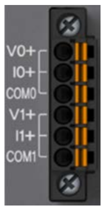
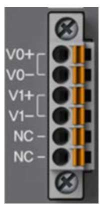


No.	Name	Content
① ~ ⑪	Same as XMC-E32A	-
⑫	Serial communication port (connector)	The port to execute RS-232C communication (connector)
⑬	Serial communication port (connector)	The port to execute RS-485 communication (connector)
⑭	Input/output display	Digital I/O, Encoder input, Communication interface status display

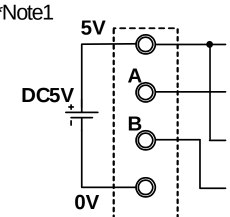
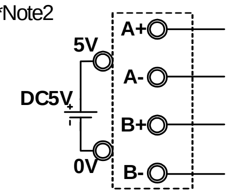
2.4.2 Specification of Interface with External Device

(1) Pin arrangement of connector

Pin arrangement	Signal name		Signal direction
	ENC1 A+	Encoder1 A+ input	Input
	ENC1 A-	Encoder1 A- input	
	ENC1B+	Encoder1 B+ input	
	ENC1B-	Encoder1 B- input	
	ENC2 A+	Encoder2 A+ input	
	ENC2 A-	Encoder2 A- input	
	ENC2 B+	Encoder2 B+ input	
	ENC2 B-	Encoder2 B- input	
	IN0	Input signal 0	Input
	IN1	Input signal 1	
	IN2	Input signal 2	
	IN3	Input signal 3	
	IN4	Input signal 4	
	IN5	Input signal 5	
	IN6	Input signal 6	
	IN7	Input signal 7	
	COM	Input signal Common	Input
	OUT0	Output signal 0	Output
	OUT1	Output signal 1	
	OUT2	Output signal 2	
	OUT3	Output signal 3	
	OUT4	Output signal 4	
	OUT5	Output signal 5	
	OUT6	Output signal 6	
	OUT7	Output signal 7	
	OUT8	Output signal 8	
	OUT9	Output signal 9	
	OUT10	Output signal 10	
	OUT11	Output signal 11	
	OUT12	Output signal 12	
	OUT13	Output signal 13	
	OUT14	Output signal 14	
	OUT15	Output signal 15	
	24V	DC 24V	Input
	GND	DC 24V GND	

Pin arrangement	Signal name		Signal direction
	V0+	Analog voltage input channel 0	Input
	I0+	Analog current input channel 0	
	COM0	Analog input 0 common	
	V1+	Analog voltage input channel 1	
	I1+	Analog current input channel 1	
	COM1	Analog channel1 common	
	V0+	Analog voltage output channel 0+	Output
	V0-	Analog voltage output channel 0-	
	V1,	Analog voltage output channel 1+	
	V1-	Analog voltage output channel 1-	
	NC	No Connection	
	NC	No Connection	

(2) Encoder part internal circuit

Classification	Signal	
*Note1 	ENC1A+	Encoder1 A+ input
	ENC1A-	Encoder1 A- input
	ENC1B+	Encoder1 B+ input
	ENC1B-	Encoder1 B- input
*Note2 	ENC2A+	Encoder2 A+ input
	ENC2A-	Encoder2 A- input
	ENC2B+	Encoder2 B+ input
	ENC2B-	Encoder2 B- input

Caution

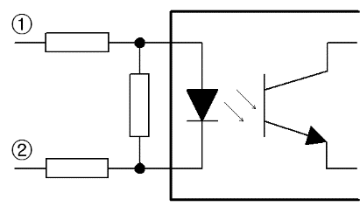
*Note 1

Wiring of encoder 1 is an example about 5V voltage output (open collector) type. When using 12V, 24V type MPG, change the input voltage from 5V to 12V or 24V and in case of 12V, connect 910Ω resistor to ENC1 A+(pin 1), ENC1 B+ (pin3), in case of 24V, 2.4kΩ resistor, before connecting the power source (adding PULL-UP resistor is needed)

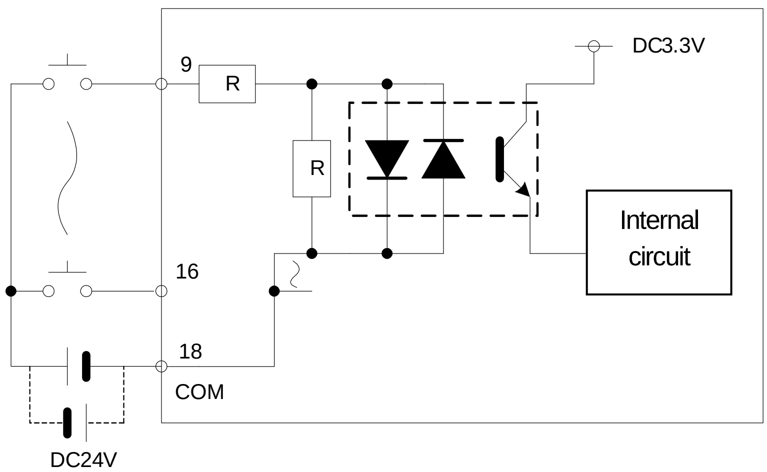
*Note 2

Wiring of encoder 2 is example about 5V voltage output type (line driver)

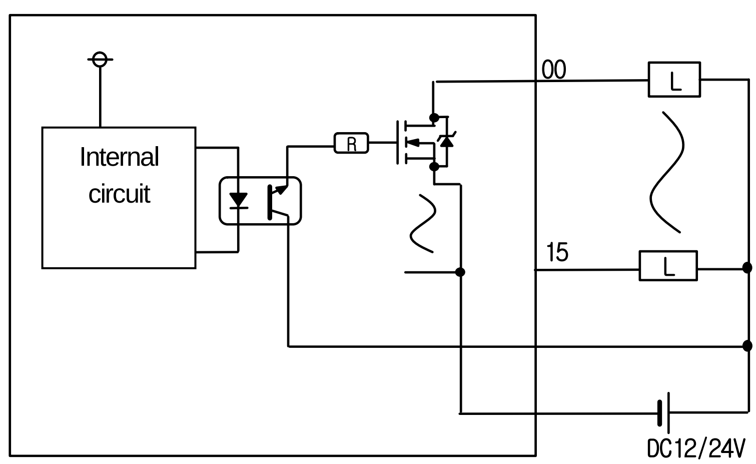
This describes the internal circuit of the module when connecting the encoder.

Input/output classification	Internal circuit	No.	terminal	Signal name
Input		①	A+	A phase pulse input +
		②	A-	A phase pulse input-
		①	B+	B phase pulse input +
		②	B-	B phase pulse input-

(3) Input part internal circuit



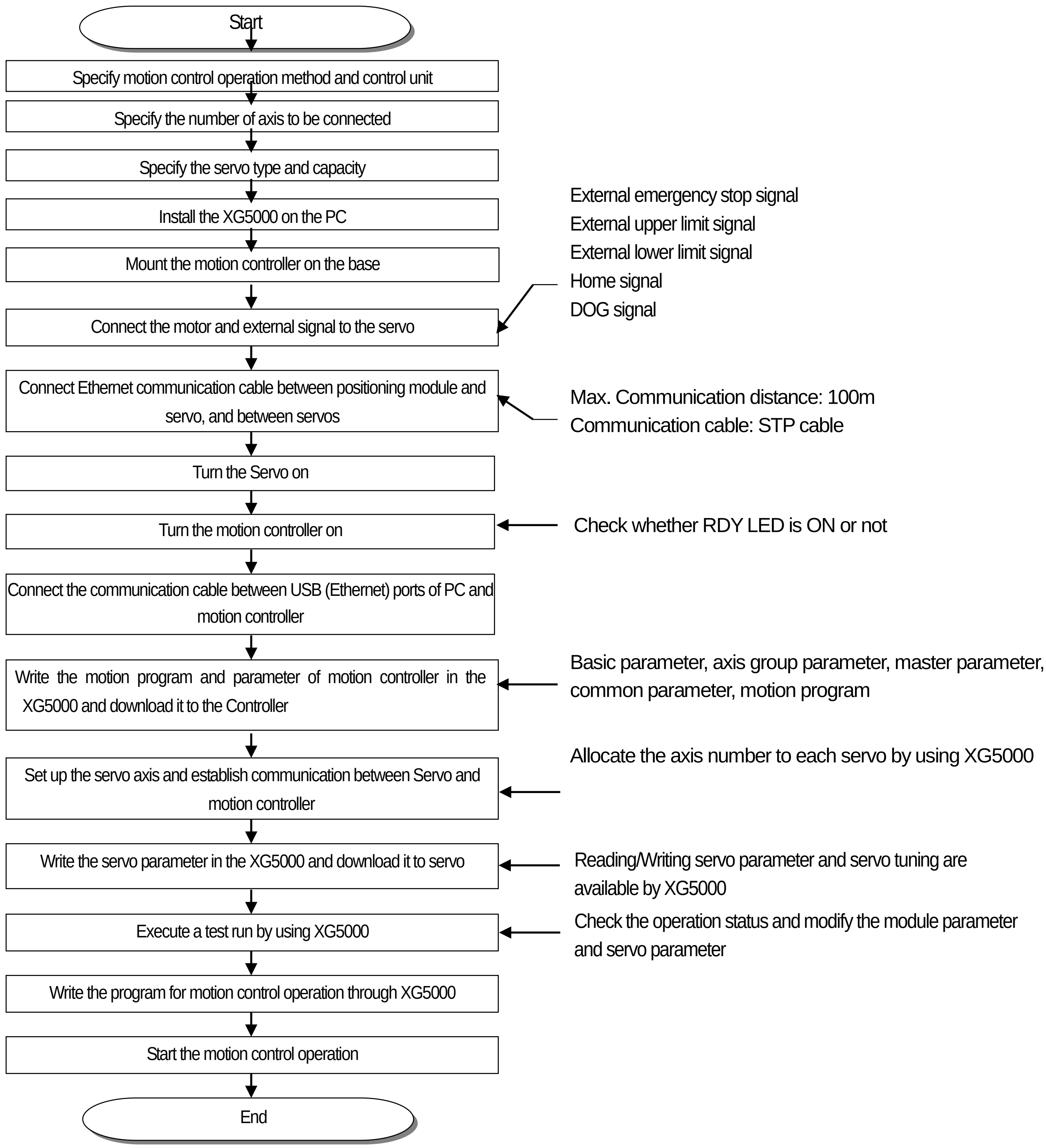
(4) Output part internal circuit



Chapter 3 Operation Order and Installation

3.1 Operation procedure

Here describes the Operation order of motion controller.



3.2 Installation

3.2.1 Safety Precautions

⚠ Danger

▸ Please design protection circuit at the external of Controller for entire system to operate safely because an abnormal output or a malfunction may cause accident when any error of external power or malfunction of Controller.

(1) It should be installed at the external side of Controller to emergency stop circuit, protection circuit, interlock circuit of opposition action such as forward /reverse operation and interlock circuit for protecting machine damage such as upper/lower limit of positioning.

.

(2) If Controller detects the following error, all operation stops and all output is off.
Available to hold output according to parameter setting
(a) When over current protection equipment or over voltage protection operates
(b) When self-diagnosis function error such as WDT error in Controller occurs

▸ When error about IO control part that is not detected by Controller, all output is off.
Design Fail Safe circuit at the external of Controller for machine to operate safely. Please refer to 3.5.1 Fail-Safe Circuit Configuration Example
(1) Because of error of output device, Relay, TR, etc., output may not be normal. About output signal that may cause the heavy accident, design supervisory circuit to external.

▸ When load current is more than rating or over current by load short flows continuously, danger of heat, fire may occur so design safety circuit to external such as fuse.

▸ Design for external power supply to be done first after Controller power supply is done. If external power supply is done first, it may cause accident by misoutput, misoperation.

▸ In case communication error occurs, for operation status of each station, refer to each communication manual.
There may be a risk of accidents due to incorrect output and malfunction.

▸ In case of controlling the Controller while peripheral is connected to Controller, configure the interlock circuit for system to operate safely. During operation, in case of executing program change, operation status change, familiarize the manual and check the safety status. Especially, in case of controlling long distance Controller, user may not response to error of Controller promptly because of communication error or etc.
Limit how to take action in case of data communication error between Controller and external device adding installing interlock circuit at the Controller program.

⚠ Danger

- Don’t close the control line or communication cable to main circuit or power line.
  Limit how to take action in case of data communication error between Controller and external device adding installing interlock circuit at the Controller program.
  Distance should be more than 100mm. It may cause malfunction by noise.

- In case of controlling lamp load, heater, solenoid valve, etc. in case of Off -> On, large current (10 times of normal current) may flows, so consider changing the module to module that has margin at rated current.

- Process output may not work properly according to difference of delay of Controller main power and external power for process (especially DC in case of) Controller power On-Off and of start time.
  For example, in case of turning on Controller main power after supplying external power for process, DC output module may malfunction when Controller is on, so configure the circuit to turn on the Controller main power first
  Or in case of external power error or Controller error, it may cause the malfunction.

- Not to lead above error to entire system, part causing breakdown of machine or accident should be configured at the external of Controller.

3.2.2 Installation Environment

This controller has a good reliability regardless of installation environment but cares should be taken in the following items to guarantee the reliability and safety of the system.

(1) Environment Condition

(a) Install the control panel available for water-proof, anti-vibration.

(b) The place free from continuous impact or vibration.

(c) The place not exposed to direct rays.

(d) The place with no dew phenomena by rapid temperature change.

(e) The place where surrounding temperature maintains 0-55℃.

(2) Installation Environment

(a) In case of processing the screw hole or wiring, cares should be taken not to put the wiring remnants to PLC inside.

(b) Install on the good place to operate.

(c) Do not install the high voltage machine on the same Panel.

(d) The distance from duct or surrounding module shall be more than 50mm.

(e) Ground to the place where surrounding noise environment is good enough.

3.2.3 Precautions of Handling

Here describes the notices in handling the positioning module from opening to installation.

(1) Do not fall down or apply the strong impact.

(2) Do not remove PCB from the case. Doing so may cause failure of the module and/or printed-circuit board.

(3) In wiring, cares should be taken not to put the wiring remnants or foreign materials to the upper part of Controller. If something entered, it should be removed.

3.2.4 Attachment/Detachment of motion controller

⚠ Caution

- Motion controller must be mounted to hook for fixation properly before its fixation.
  The Controller may be damaged from over-applied force.
- If module is not mounted properly, it may cause malfunction.
- Do not drop or impact the module case, terminal block connector.

Caution in handling

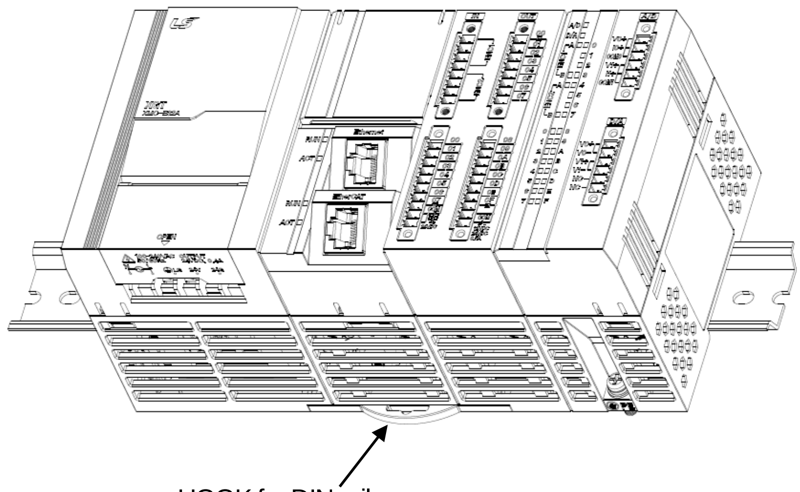
Use motion controller in the range of general specification specified by manual.
In case of usage out of range, it may cause electric shock, fire, malfunction, damage of product.

(1) Installation of motion controller

Motion controller has a hook for DIN rail (rail width: 35mm) so that can be installed at DIN rail.

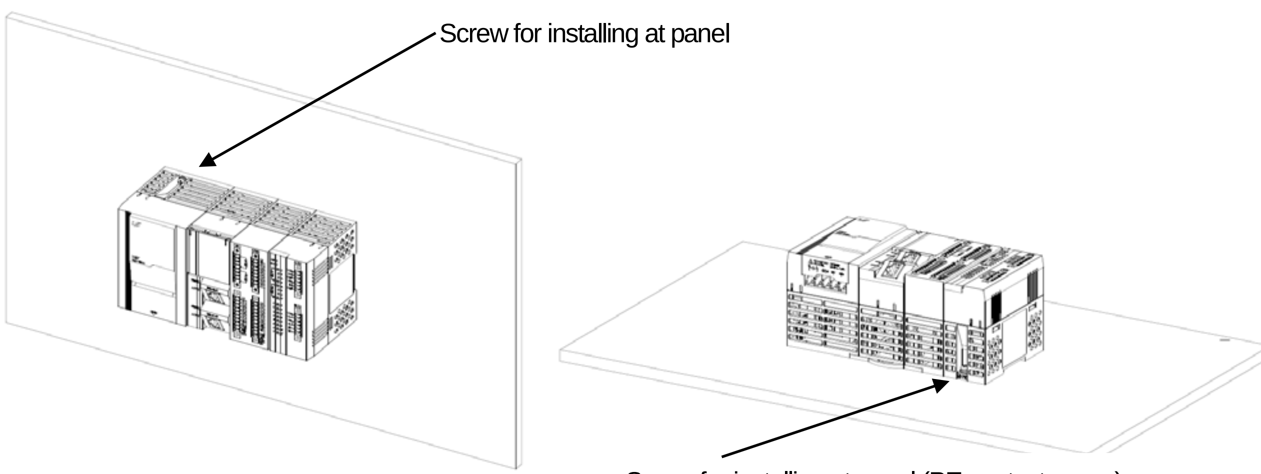
(a) In case of installing at DIN rail

- Pull the hook as shown below for DIN rail at the bottom of motion controller and install it at DIN rail.
- Push the hook to fix the module at DIN rail after installing motion controller at DIN rail.



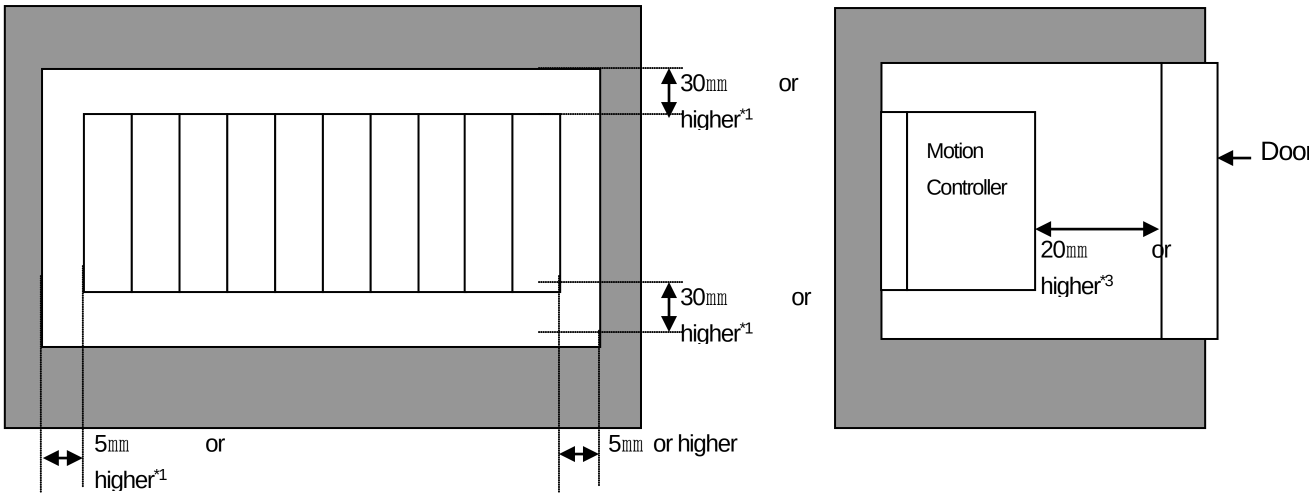
(b) In case of installing at panel

- You can install motion controller onto a panel directly using screw hole.
- Use M4 type screw to install the product onto a panel.
- This product is designed so that PE and panel come in contact with each other through a screw at the bottom right of the product. When installing on the panel, be sure to connect the screw in the bottom right side.



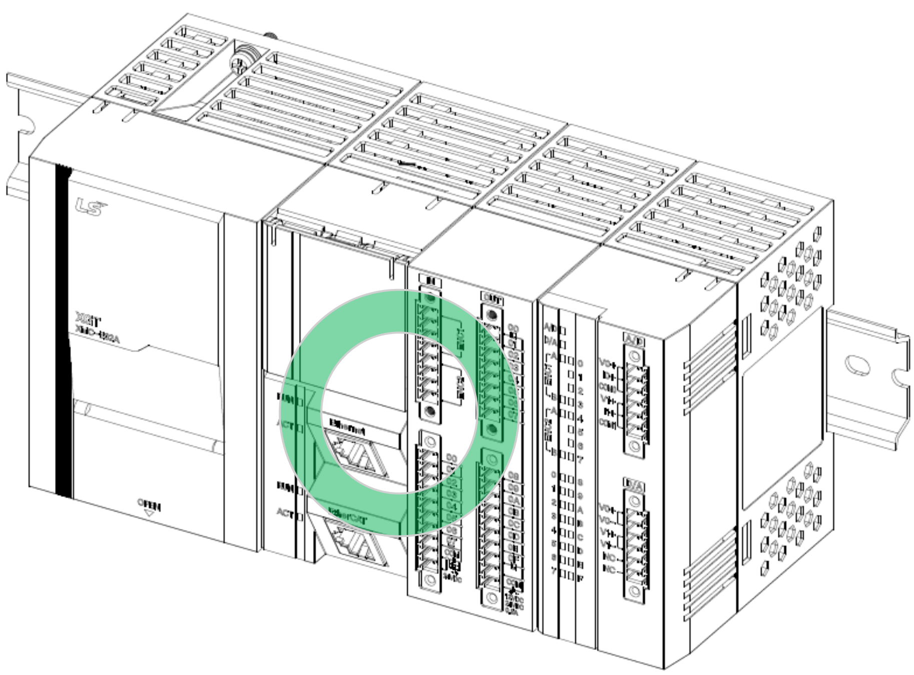
(2) Controller equipment location

Keep the following distance between module and structure or part for ventilation, easy detachment and attachment.

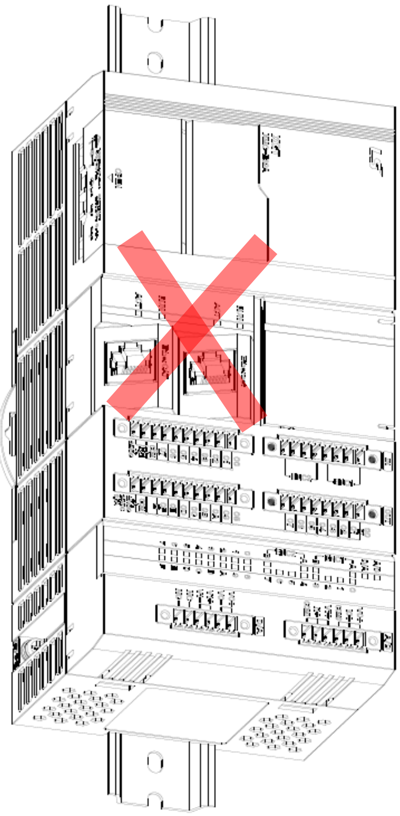


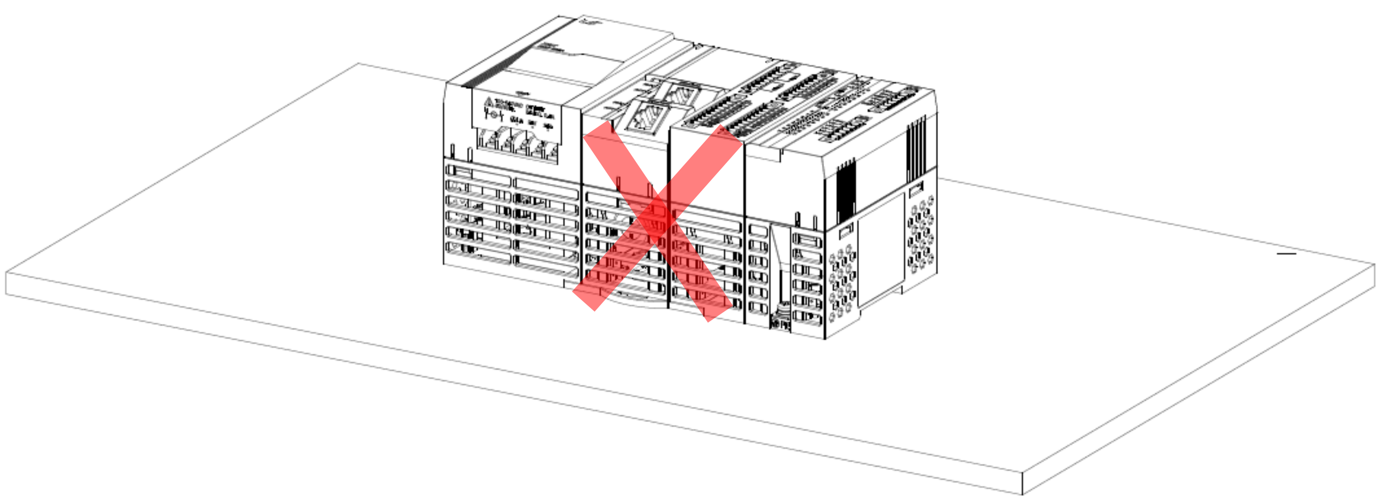
(3) Controller equipment direction

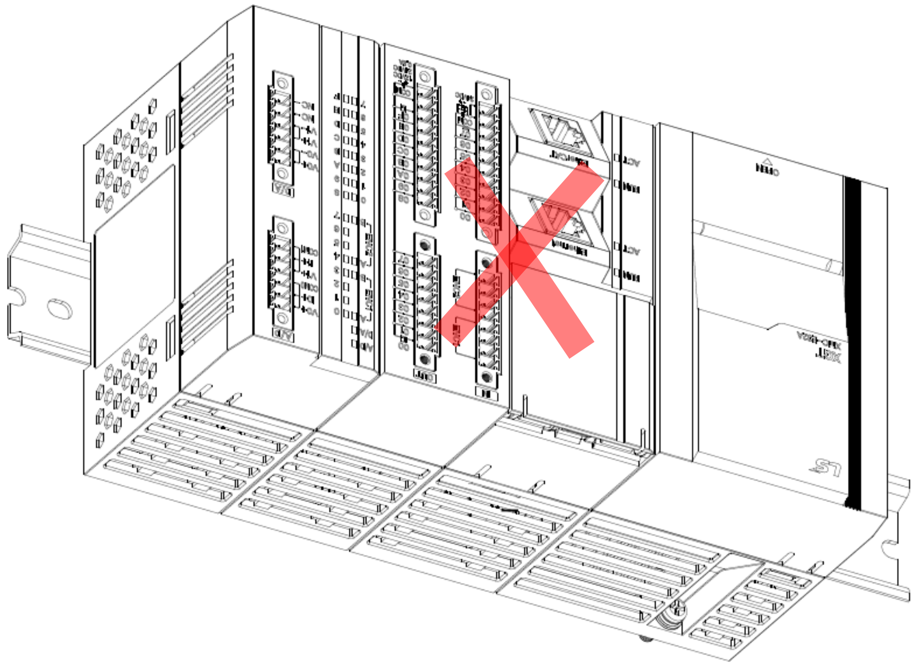
(a) For easy ventilation, install as shown below.

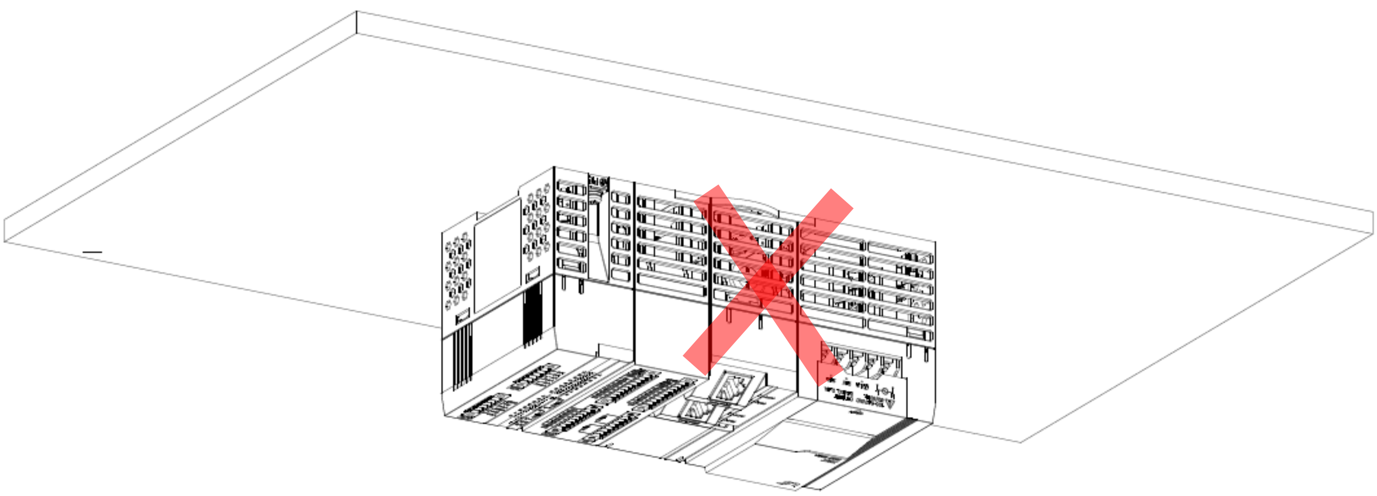


(b) Don't install as shown below.



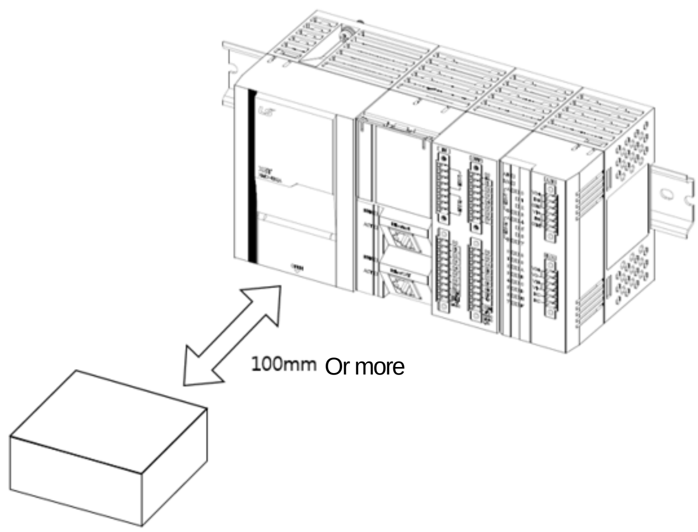


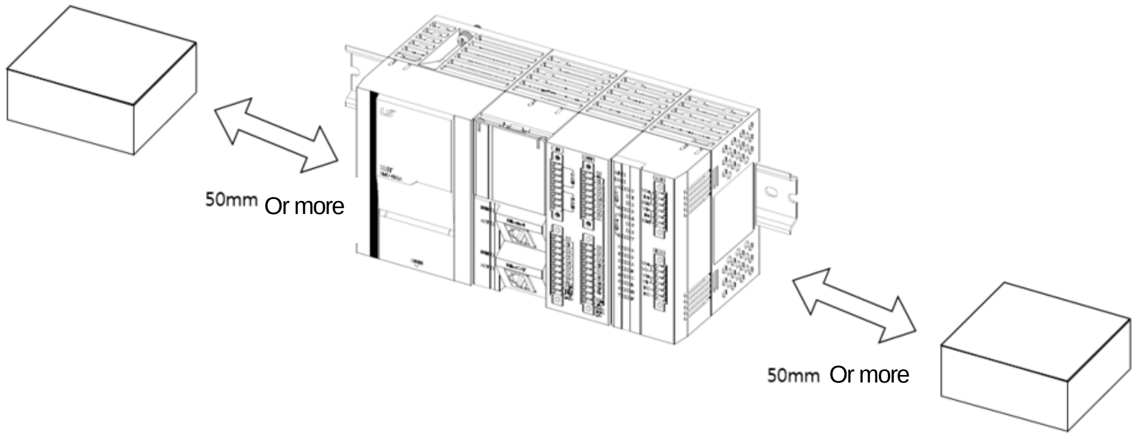




(4) Distance with other device

To avoid radiation noise or heat, keep the distance between motion controller and device (connector and relay) as far as the following figure.





3.3 Precautions in Wiring

3.3.1 Precautions of wiring

(1) The length of connecting cable between controller and drive machine shall be as short as possible.

(2) Use a separate cable from AC power for the signal line connected to the motion controller and the EtherCAT slave device, and make sure that it is not affected by surges or induced noise from the AC side.

(3) The wires should be selected considering surrounding temperature, allowable current and it is recommended to be more than max. size AWG22(0.3㎟).

(4) In wiring, if it is too close to the high temperature machine or material or it is directly contacted to the oil for a long time, the short-circuit will occur that may cause the damage or malfunction.

(5) Make sure to check the polarity before applying the external contact signal to the terminal board.

(6) In case of wiring the high voltage cable and power cables together, the induction noise occurs that may cause the malfunction or failure.

(7) In case of wiring by the pipe, the grounding of pipe is required.

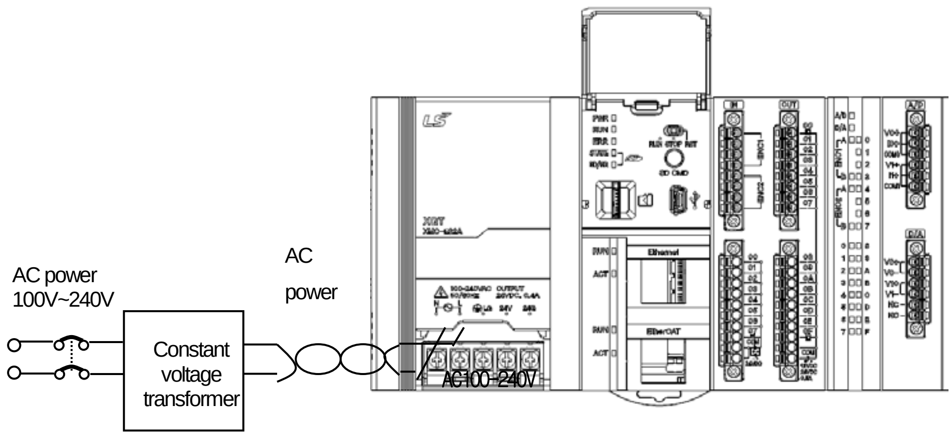
(8) Connect the line between controller and EtherCAT slave device by using more than STP CAT-5 in wiring between controller and drive unit.

(9) When a communication error(0x0F50, 0x0F51, 0x1F00, 0x1011, 0x2011, etc.) occurs in operation of controller, attach Ferrite Core to communication cable connecting controller to EtherCAT slave device and run the controller because it may be caused by noise interference in wiring between controller and EtherCAT slave device.

(10) When using the wiring connector for encoder signal and external I/O signal, install it on the place where there is no dust or corrosive gas.

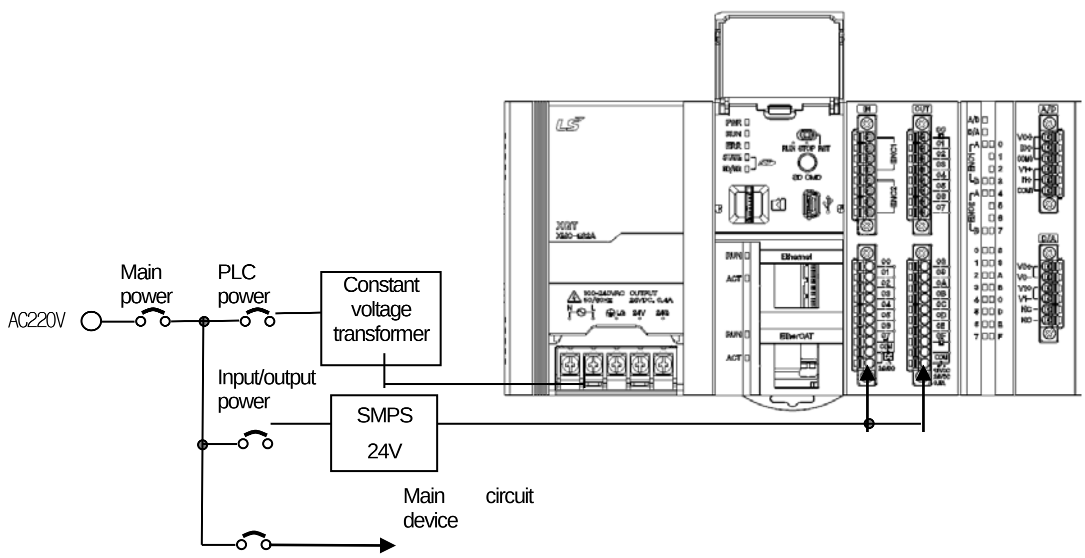
3.3.2 Power Wiring

(1) In case voltage regulation is larger than specified, connect constant voltage transformer.



(2) Connect the power source of which inter-cable or cable-ground noise is small.
(When there is much noise, connect insulated transformer.)

(3) Isolate the controller power, I/O devices and power devices as follows.



(4) If using DC24V of the controller, do not connect DC24V of several power modules in parallel.

(5) AC power cables should be compactly twisted and connected in the shortest distance.

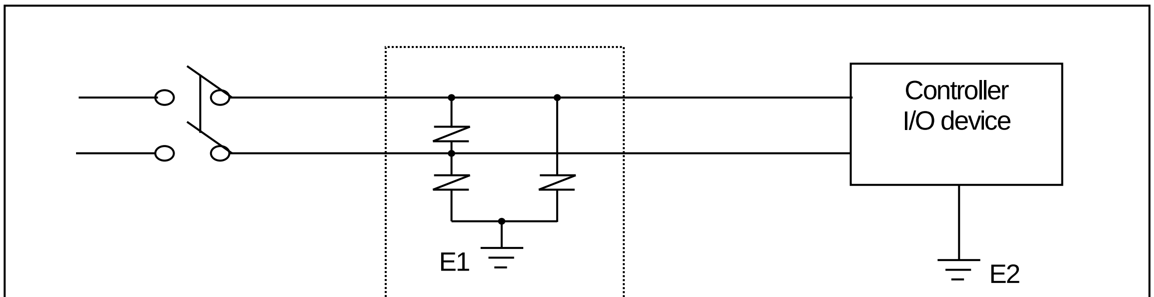
(6) AC power cables should be as thick as possible ($2mm^2$) to reduce voltage drop.

(7) AC power cables should not be installed close to main circuit cable (high voltage/high current) and I/O signal cable. They should be 100mm away from such cables

(8) Use the shielded insulation trans or noise filter when a noise invasion is expected.

(9) Wiring of each input power should be twisted as short as possible and the wiring of shielding transformer or noise filter should not be arranged via a duct.

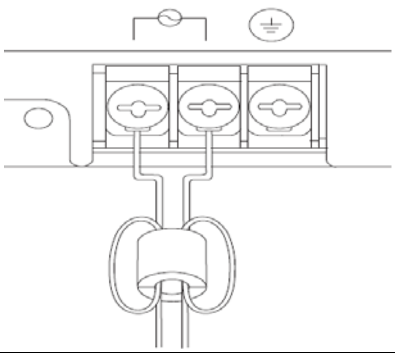
(10) To prevent surge from lightning, use the lightning surge absorber as presented below.



Surge absorber to prevent lighting

Notes

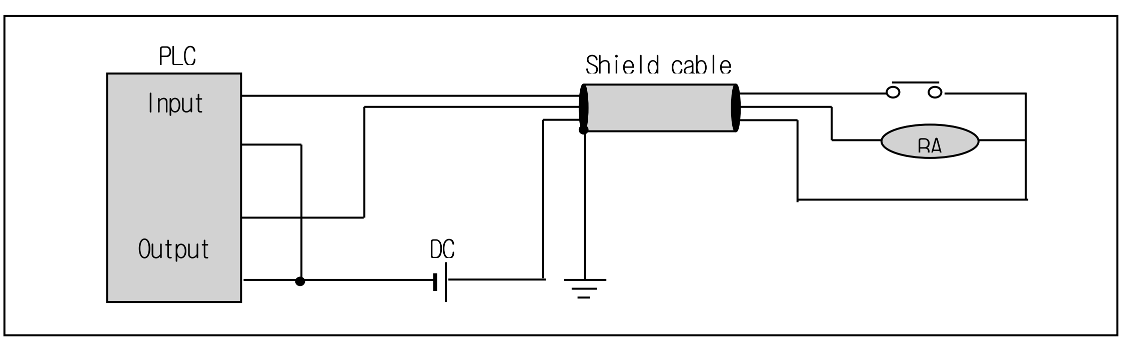
1) Isolate the grounding (E1) of lightning surge absorber from the grounding (E2) of the controller.
2) Select a lightning surge absorber type so that the max. Voltage may not the specified allowable voltage of the absorber.
3) When you use the magnetic lighting contactor (MC) for AC/DC, it is recommended to use a ferrite core close to the power. terminals as shown below



Ferrite core
(CU1330B, E-TECH)
(CU1330G, E-TECH)
(ZCAT3035-1330, TDK)
(TU1335,ESONG EMC)

3.3.3 I/O devices wiring

(2) The size of I/O device cable is limited to 0.3~2 mm^2 Please select a wire within the standard considering the convenience of wiring.
(2) Please isolate input signal line from output signal line.
(3) I/O signal lines should be wired 100mm and more away from high voltage·high current main circuit cable.
(4) Batch shield cable should be used and the motion controller side should be grounded unless the main circuit cable and power cable cannot be isolated.



(5) When applying pipe-wiring, make sure to firmly ground the piping.

3.3.4 Grounding wiring

(1) This controller has two types of grounding systems such as LG and PE.

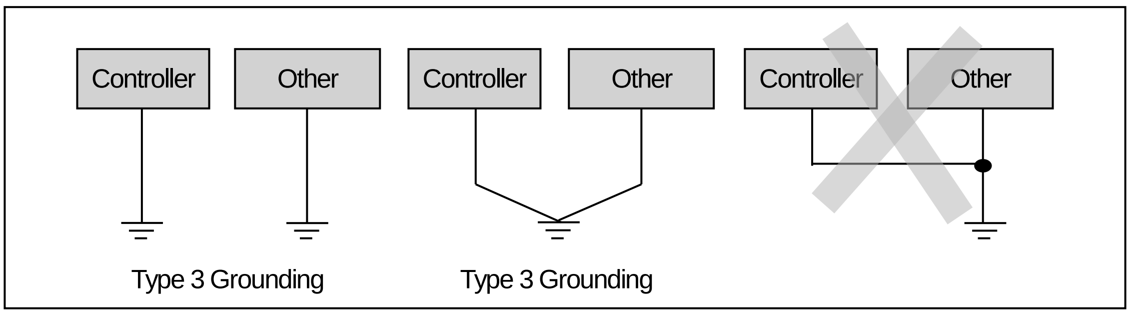
(2) LG () is grounding for a power filter and used as a noise countermeasure. This controller performs sufficient noise countermeasures, but it is recommended to use LG if there is no specific reason. For the location of LG, please refer to the names of each part of 2.4.1.

(3) PE () is grounding to prevent an electric shock. It should be in contact with ground portion to prevent accidents. When Din rail is installed, it is in contact with the DIN rail. When the panel is installed, it is contact with the panel through the screw for panel installation. For the location of PE, please refer to Section 3.2.3 Installing and Removing the Motion Controller.

(4) Please refer to the following instructions for LG grounding.

(a) For grounding, please make sure to use the exclusive grounding. For grounding construction, apply type 3 grounding (grounding resistance lower than 100Ω)

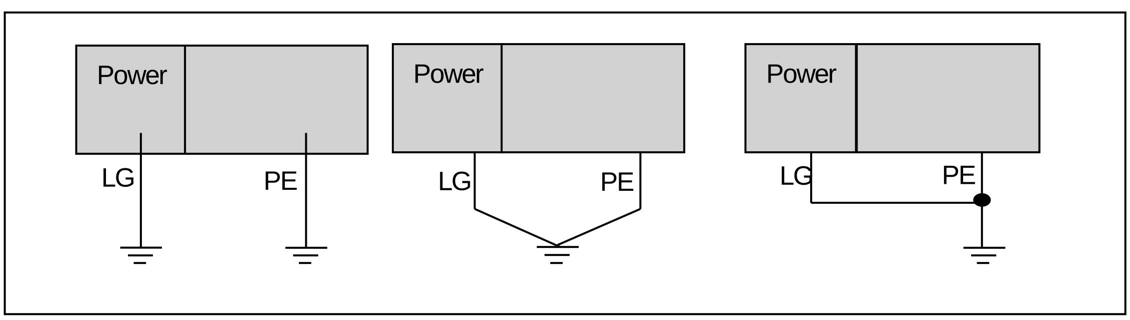
(B) If the exclusive grounding is not possible, use the common grounding as presented in B) of the figure below.



(a) Exclusive grounding: best (b) common grounding: good (c) common grounding: defective

(c) Use the grounding cable more than 2 mm^2. To shorten the length of the grounding cable, place the grounding point as close to the controller as possible.

(d) If any malfunction from grounding is detected, separate the PE and LG.



(a) Exclusive grounding: best (b) common grounding: good (c) common grounding: defective

(5) PE() is basically in contact with DIN rail. However, if DIN rail is coated, it may not be grounded. In this case, use a screw and connect ground wiring to the PE () terminal.

3.3.5 Cable Specification for Wiring

The specifications of cable used for wiring are as follows.

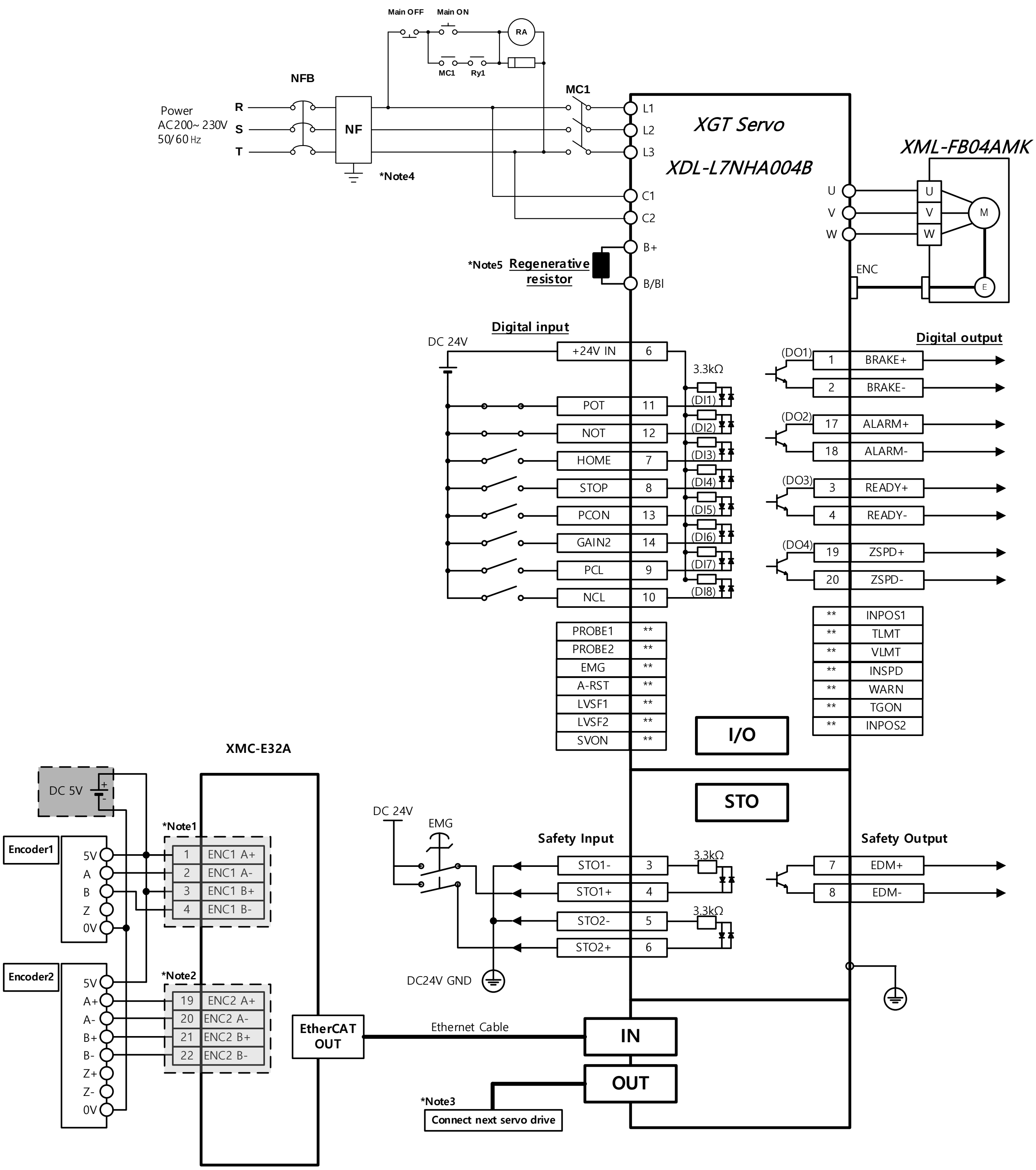
Types of external connection	Cable specification (mm²)	
	lower limit	Upper limit
Digital input	0.18 (AWG24)	1.5 (AWG16)
Digital output	0.18 (AWG24)	1.5 (AWG16)
Analog input/output	0.18 (AWG24)	1.5 (AWG16)
Communication	0.18 (AWG24)	1.5 (AWG16)
Main power	1.5 (AWG16)	2.5 (AWG12)
Grounding(LG)	1.5 (AWG16)	2.5 (AWG12)

Use the following specifications for the power cable.

Connection type	Cable specification (mm²)	Thermal resistance	Thermal resistance	Tightening Torque
Power and protection ground	1.5(AWG16)~2.5(AWG12)	Copper	60℃	0.51N•m

3.3.6 Connection of Servo Drive

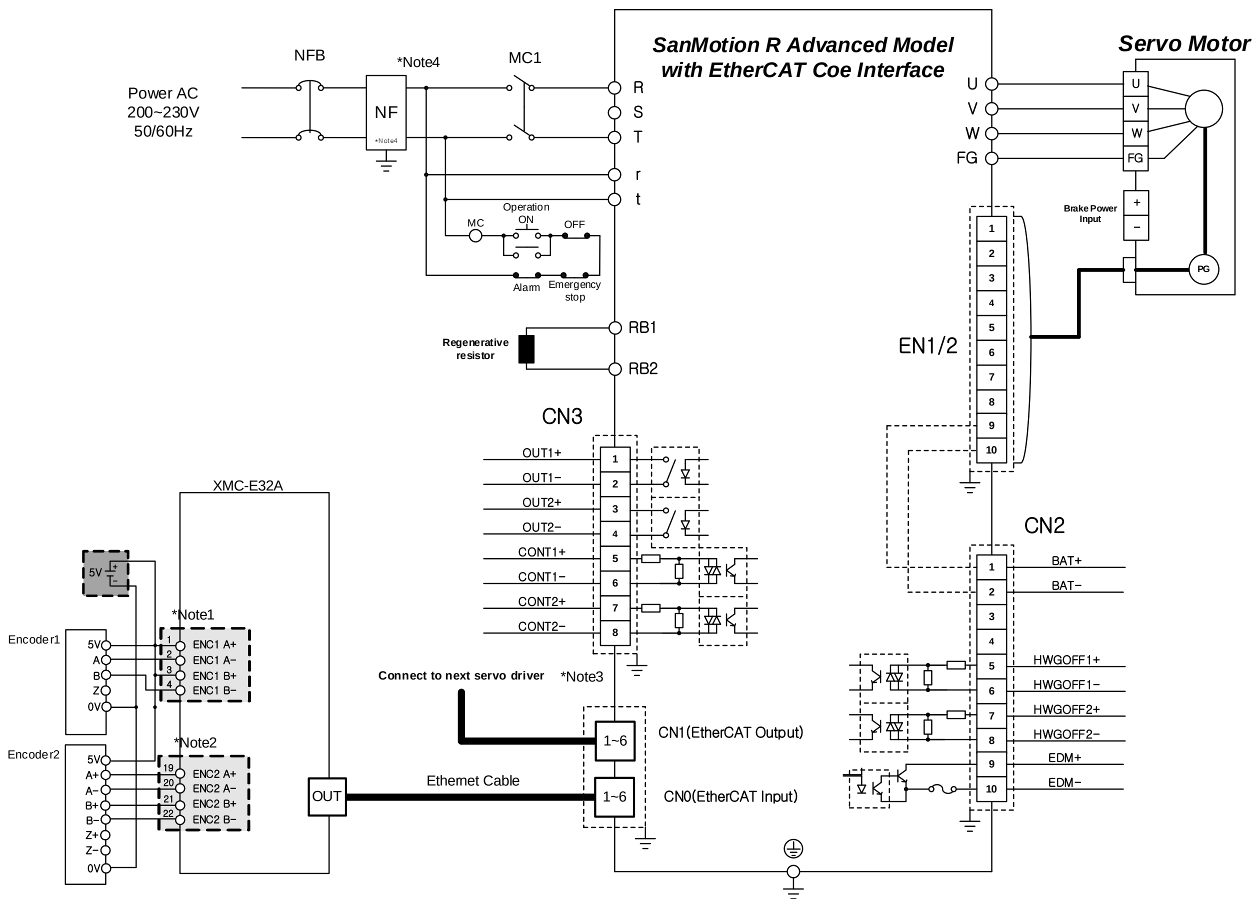
(1) This is an example of wiring from motion controller to XGT Servo L7NH series (EtherCAT servo drive/motor). Refer to manual of each drive for details on installation and wiring.



Caution

*Note 1
Wiring of encoder 1 is an example about 5V voltage output (open collector) type.
*Note 2
Wiring of encoder 2 is example about 5V voltage output type (line driver)
*Note 3
When connecting more than 2 servo drivers, connect first servo driver's IN to the motion controller's OUT and for other servo drivers, connect previous servo driver's OUT to next servo driver's IN. Last servo driver's OUT doesn't need to be connected. Axis number can be set through "Slave/Axis Connection" in XG5000. However, when the servo drive is automatically connected, the axis order is automatically set according to the connection order.
*Note 4
NF is abbreviation of Noise Filer. It is necessary to prevent the noise from coming in.
*Note 5
Use after making a short circuit between terminals B and BI as regenerative resistor of L7NHA001B~L7NHA004B (50[W], 100[Ω]), L7NHA008B~L7NHA010B (100[W], 40[Ω]), L7NHA020B~ L7NHA035B (150[W], 13[Ω]) is contained inside. In case of a high regeneration capacity due to frequent acceleration/deceleration, open the shorting pin (B, BI) and connect external resistor to B and BI to use.

(2) This is wiring example connecting SanMotion R Advanced Model EtherCAT servo drive/motor to motion controller. For detail on installation and wiring, refer to the driver manual.



Caution

*Note 1

Wiring of encoder 1 is an example about 5V voltage output (open collector) type.

*Note 2

Wiring of encoder 2 is example about 5V voltage output type (line driver)

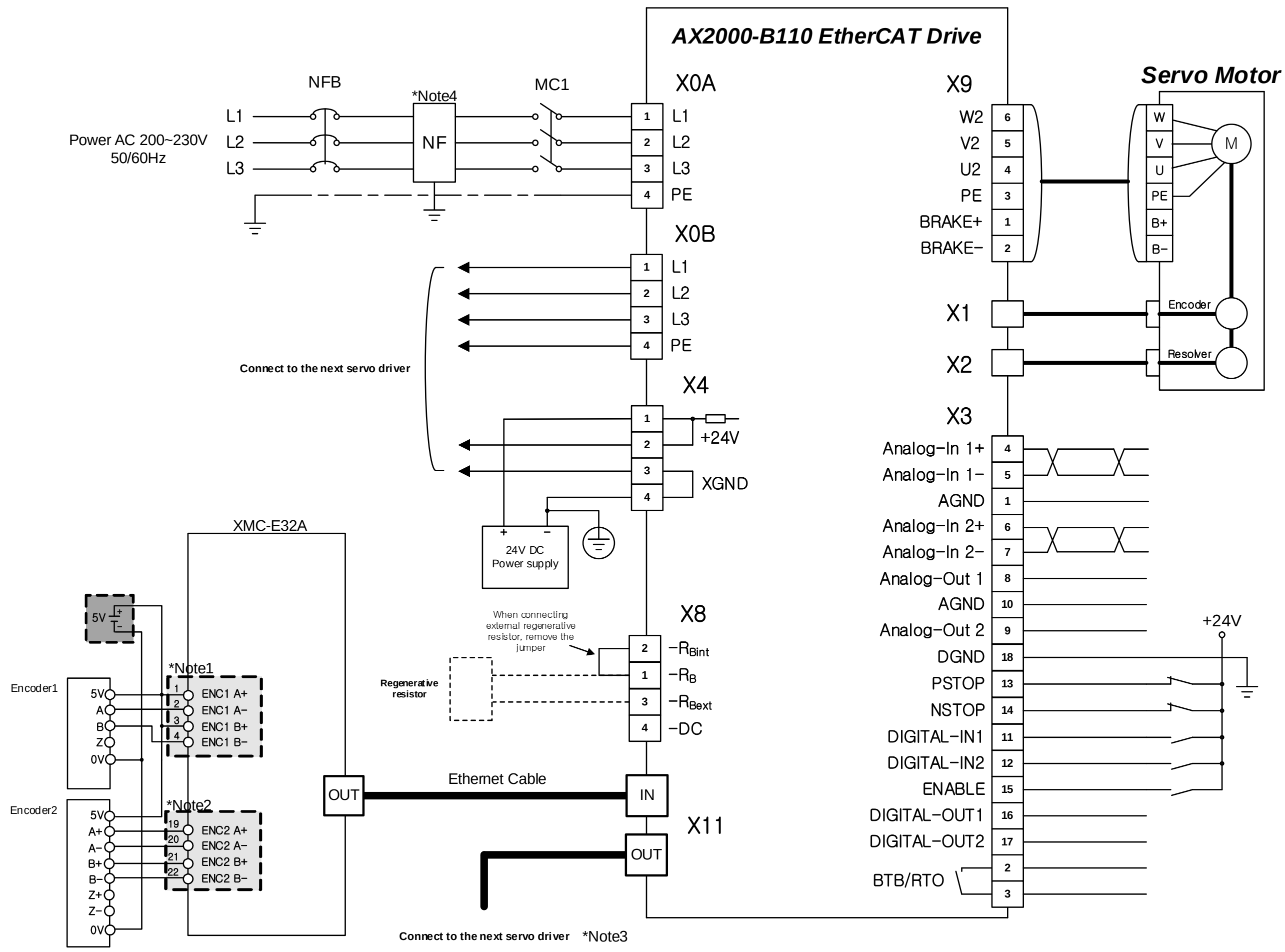
*Note 3

When connecting more than 2 servo drivers, connect first servo driver's IN to the motion controller's OUT and for other servo drivers, connect previous servo driver's OUT to next servo driver's IN. Last servo driver's OUT doesn't need to be connected. Axis number can be set through "Slave/Axis Connection" in XG5000. However, when the servo drive is automatically connected, the axis order is automatically set according to the connection order.

*Note 4

NF is abbreviation of Noise Filer. It is necessary to prevent the noise from coming in.

(3) This is wiring example connecting BeckHoff AX2000 servo drive/motor to motion controller. For detail on installation and wiring, refer to the driver manual.



Caution

*Note 1

Wiring of encoder 1 is an example about 5V voltage output (open collector) type.

*Note 2

Wiring of encoder 2 is example about 5V voltage output type (line driver)

*Note 3

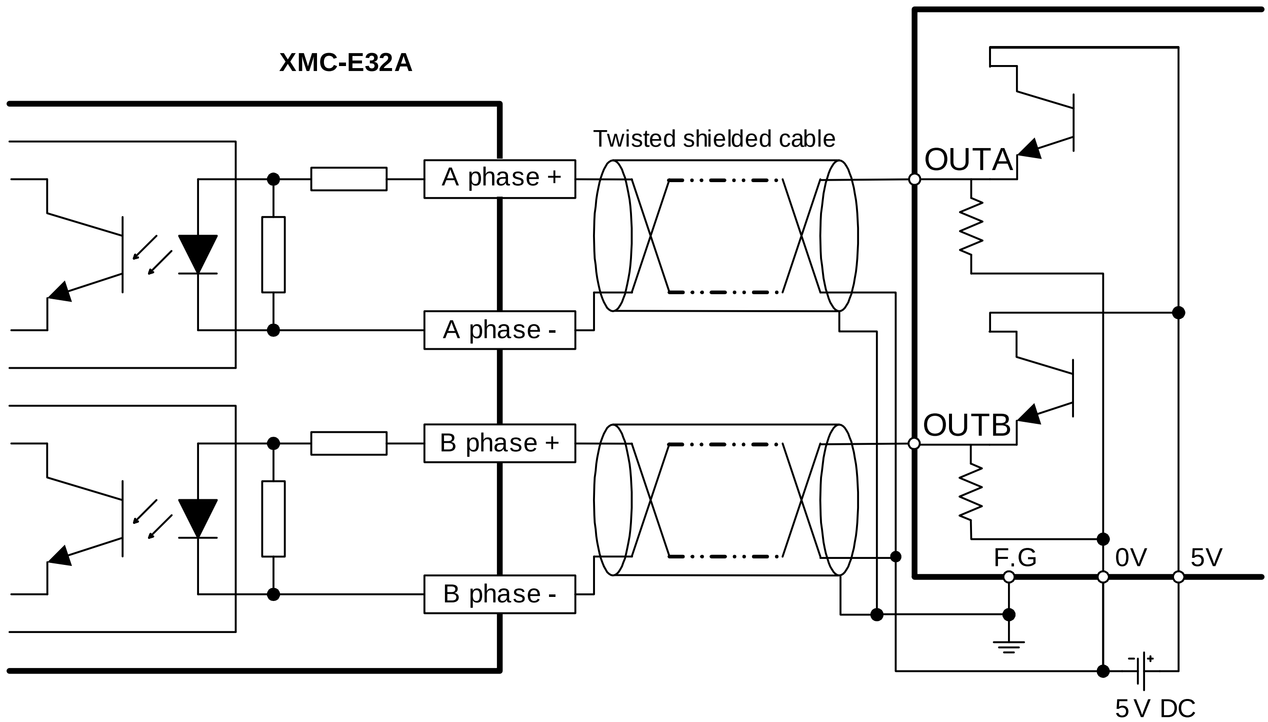
When connecting more than 2 servo drivers, connect first servo driver’s IN to the positioning module’s OUT and for other servo drivers, connect previous servo driver’s OUT to next servo driver’s IN. Last servo driver’s OUT doesn’t need to be connected. Axis number can be set through “Slave/Axis Connection” in XG5000. However, when the servo drive is automatically connected, the axis order is automatically set according to the connection order.

*Note 4

NF is abbreviation of Noise Filer. It is necessary to prevent the noise from coming in.

3.3.7 Encoder Input (DC5V Voltage Output) Wiring Example

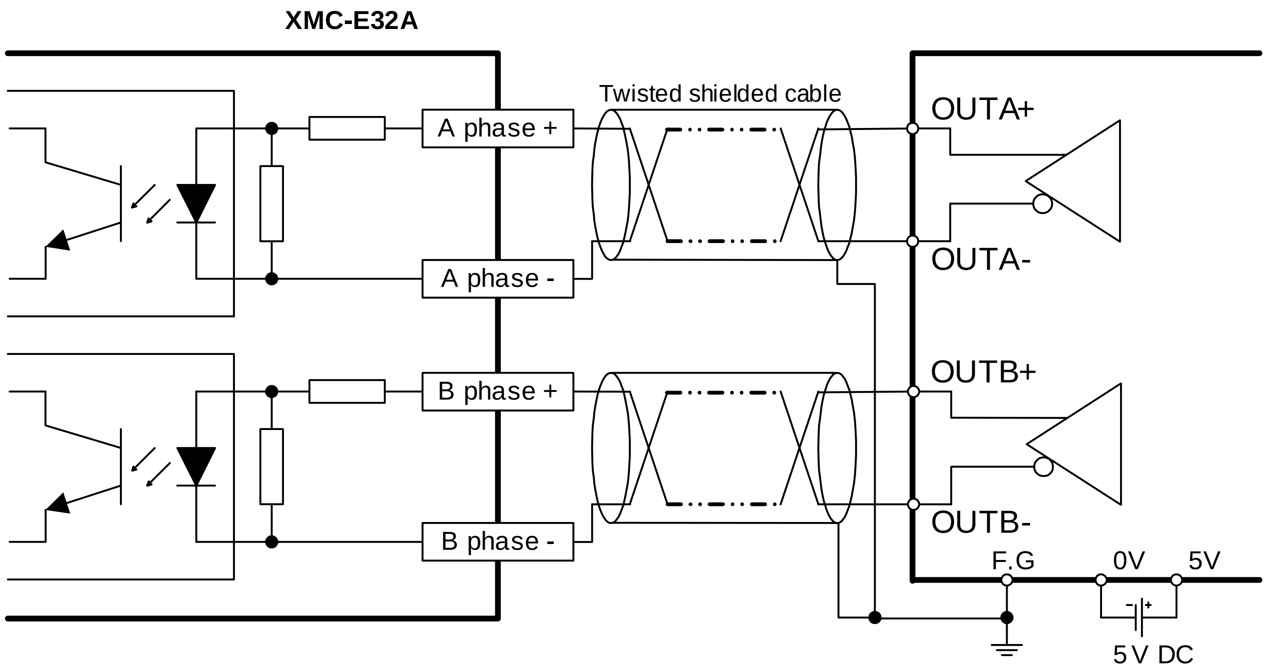
When pulse generator is a voltage output type, wiring example of motion controller and encoder input part is as follows.
In case that pulse generator is totem-pole output which is used as voltage output, wiring method is same with above.



Notes

Before Wiring, please consider maximum output distance of pulse generator.

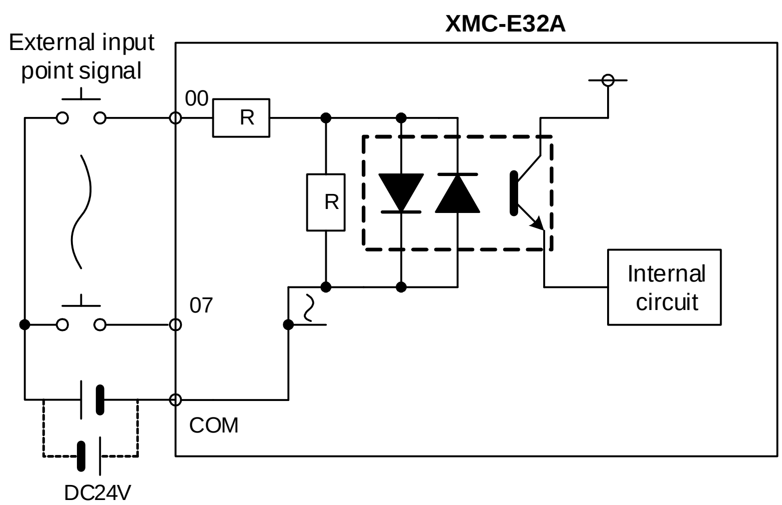
3.3.8 Encoder Input (5V Line Driver Output) Wiring Example



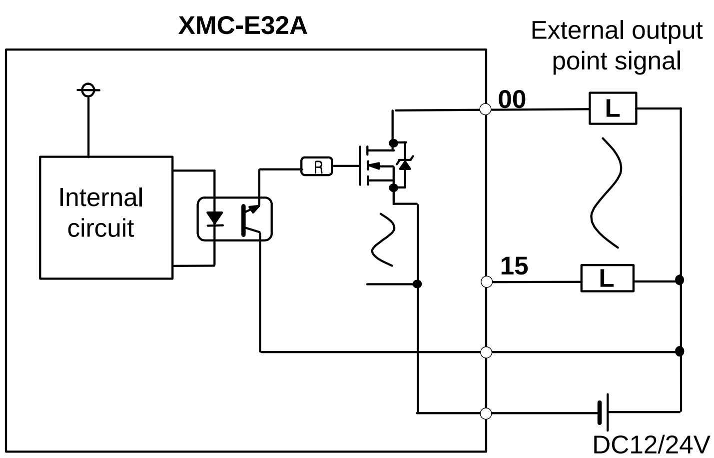
Notes

Before Wiring, please consider maximum output distance of pulse generator.

3.3.9 External input signal wiringexample



3.3.10 External Output signal wiringexample



3.4 EMC

3.4.1 EMC Standard

(1) EMC Standard

The EMC Directive specifies the products must 'be so constructed that they do not cause excessive electromagnetic interference (emissions) 'and 'are not unduly affected by electromagnetic interference (immunity)'. The applicable products are requested to meet these requirements. This section summarizes the precautions on conformance to the EMC Directive of the machinery assembled using motion control.

The details of these precautions are based on the requirements and the applicable standards control. However, LSIS will not guarantee that the overall machinery manufactured according to these details conforms to the below-described directives.

The method of conformance to the EMC directive and the judgment on whether or not the machinery conforms to the EMC Directive must be determined finally by the manufacturer of the machinery.

(2) EMC Standard

The standards applicable to the EMC Directive are listed below.

Specifications	Test item	Test details	Standard value
EN50081-2	EN55011 Radiated noise ＊2	Electromagnetic emissions from the product are measured	30~230 ㎒ QP: 50 dB㎶/m ＊1 230~1000 ㎒ QP: 57 dB㎶/m
	EN55011 conducted noise	Electromagnetic emissions from the product to the power line is measured.	150~500 ㎑ QP: 79 dB Mean : 66 dB 500~230 ㎒ QP: 73 dB Mean : 60 dB
EN61131-2	EN61000-4-2 Electrostatic immunity	Immunity test in which static electricity is applied to the case of the equipment	15 ㎸ Aerial discharge 8 ㎸ Contact discharge
	EN61000-4-4 Fast transient burst noise	Immunity test in which burst noise is applied to the power line and signal lines	Power line: 2 ㎸ Digital /O: 1 ㎸ Analog I/O, signal lines: 1 ㎸
	EN61000-4-3 Radiated field AM modulation	Immunity test in which field is irradiated to the product	10Vm,26~1000 ㎒ 80%AM modulation@ 1 ㎑
	EN61000-4-12 Damped oscillatory wave immunity	Immunity test in which a damped oscillatory wave is superimposed on the power line	Power line: 1 ㎸ Digital I/O (24V or higher): 1 ㎸

＊1: QP: Quasi-peak value, Mean: Average value

＊2: The motion controller is an open type device (device installed to another device) and must be installed in a conductive control panel.

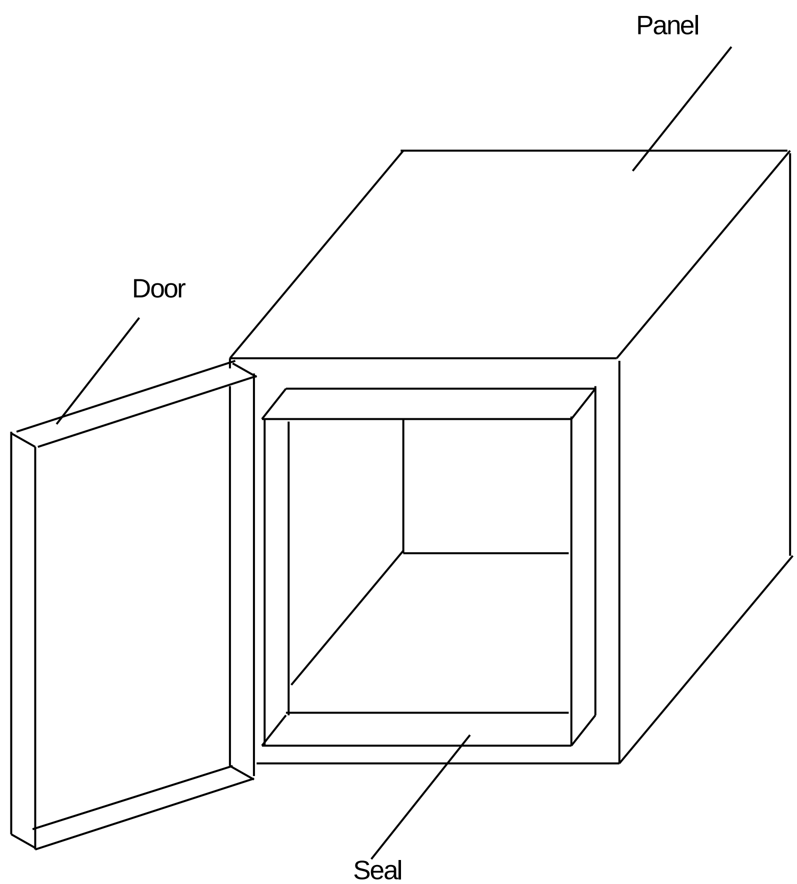
(3) Control Panel

The motion controller is an open type device (device installed to another device) and must be installed in a control panel. This is needed to prevent electric shock by touching motion controller and reduce the motion controller-generated noise. Install the motion controller in a metallic panel to reduce motion controller-generated EMI (Electro-magnetic interference), the specifications for the control panel are as follows

1) Control panel

The motion controller control panel must have the following features:

(a) Use SPCC (Cold Rolled Mild Steel) for the control panel.
(b) The steel plate should be thicker than 1.6mm.
(c) Use isolating transformers to protect the power supply from external surge voltage.
(d) The control panel must have a structure which the radio waves does not leak out. For example, make the door as a box-structure so that the panel body and the door are overlapped each other. This structure reduces the surge voltage generate by PLC.



(e) To ensure good electrical contact with the control panel or base plate, mask painting and weld so that good surface contact can be made between the panel and plate.

2) Connection of power and earth wires

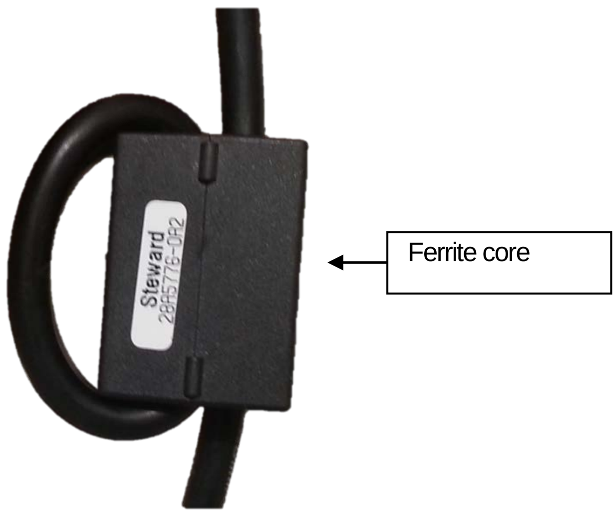
Earthing and power supply wires for the motion controller system must be connected as described below.

(a) Earth the control panel with a thick wire so that a low impedance connection to ground can be ensured even at high frequencies.

(b) The function of LG (Line Ground) and PE (Frame Ground) terminals is to pass the noise generated in the PLC system to the ground, so an impedance that is as low as possible must be ensured.

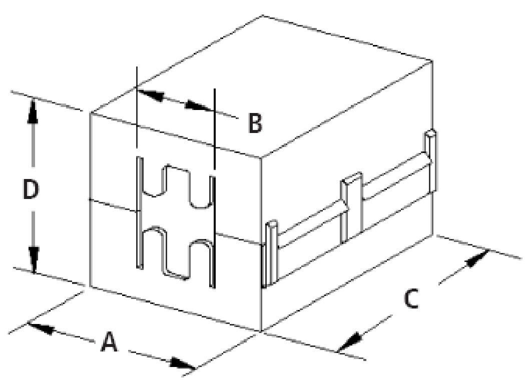
(c) The earthing wire itself can generate the noise, so wire as short and thick to prevent from acting as an antenna.

(d) Attach ferrite core under the power cable to satisfy CE specification.



[Reference product - ferrite core]

Vendor	Product name	Dimensions (mm)				Cable Maximum diameter (mm)	Note
		A	B	C	D		
Laird	28A3851-0A2	30.00	13.00	33.70	30.00	12.85	www.lairdtech.com
Laird	28A5776-0A2	29.20	20.00	42.00	42.00	19.40	www.lairdtech.com
Coilmaster	C2L RU130B	31.50	13.00	33.00	31.50	13.00	www.coilmaster.com.tw
TDK	ZCAT3035-1330	30.00	13.00	34.00	30.00	13.00	www.tdk.com



(4) Requirement to Conform to the Low-Voltage Directive

The low-voltage directive requires each device that operates with the power supply ranging from 50V to 1000VAC and 75V to 1500VDC to satisfy the safety requirements.
Cautions and installation and wiring of the motion controller series to conform to the low-voltage directive are described in this section.
The described contents in this manual are based on the requirements and the applicable standards control.
However, LSIS will not guarantee that the overall machinery manufactured according to these details conforms to the above regulation.
The method of conformance to the EMC directive and compliance to the EMC Directive must be determined by the manufacturer of the machinery.

1) Standard applied for motion controller

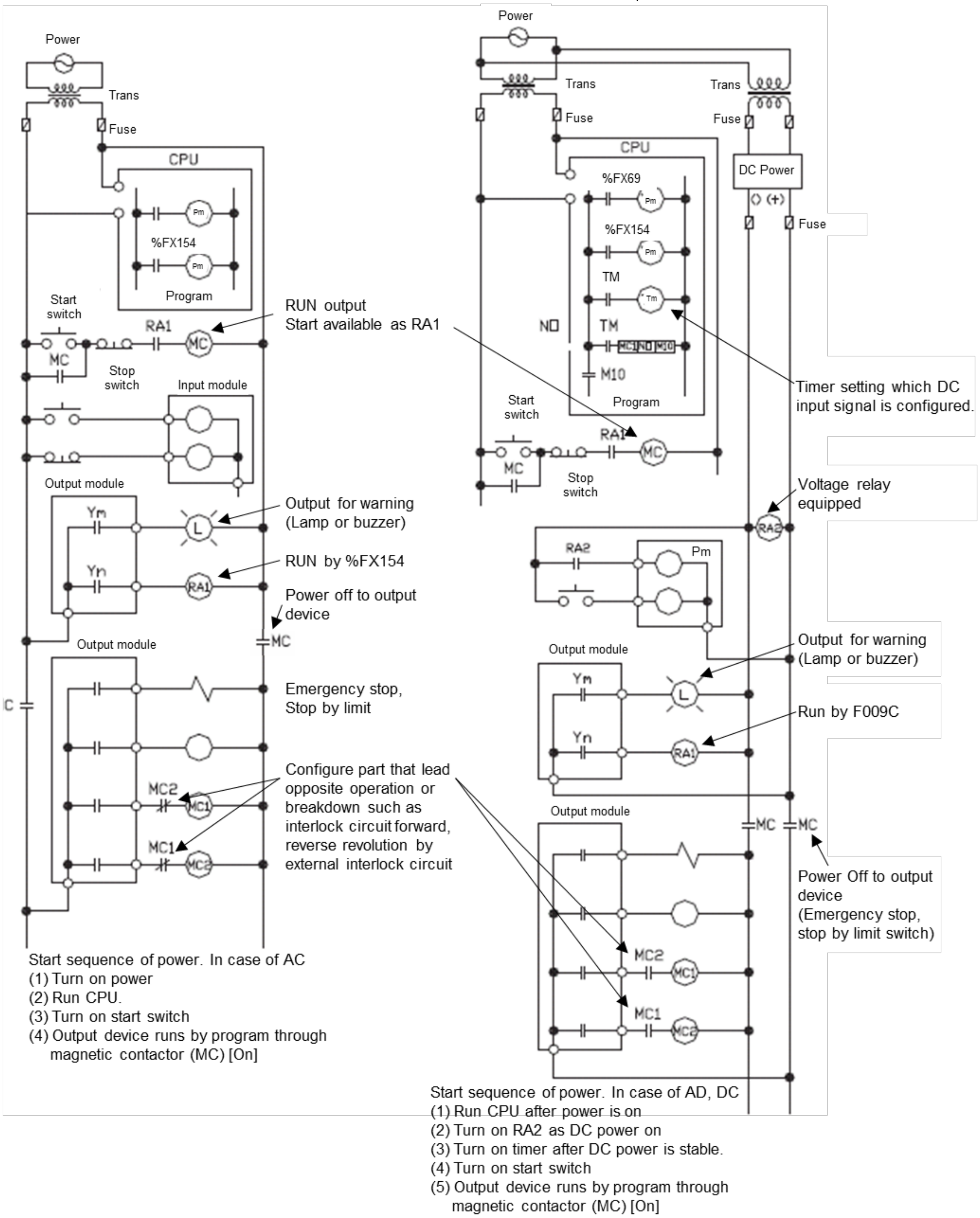
The motion controller follow EN6100-1 (safety of devices used in measurement rooms, control rooms, or laboratories).
And the motion controller modules which operate at the rated voltage of AC50V/DC75V or above are also developed to conform the above standard.

3.5 Fail Safe

3.5.1 Fail Safe Circuit

(1) Example of system design

In case of AC   in case of AC , DC



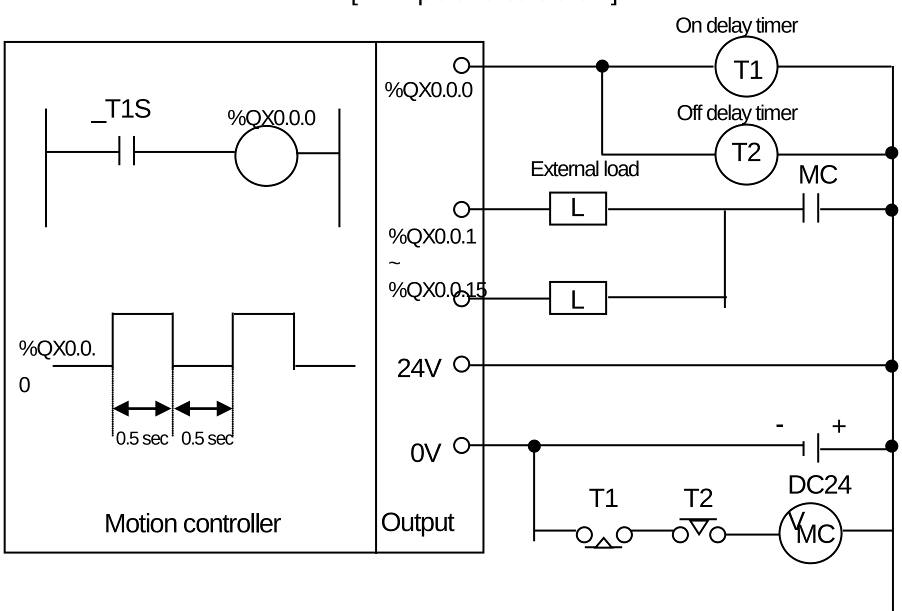
(2) Fail safe measures in case of motion controller failures

Failures of the motion controller and memory are detected by self-diagnosis but if there are some problems with I/O control part, etc, the failure may not be detected from the motion controller. In this case, it can be different depending on the failure status, all contacts may be On or Off so normal operation or safety of the controlled subject cannot be guaranteed.

We have done our best to assure quality but in case there are some problems with the PLC, please configure the fail safe circuit on the outside to prevent damage of the equipment or accident due to some cause.

The below is the example of system configuration with the fail sage circuit.

[Example of failsafe circuit]



3.6 Maintenance

Be sure to perform daily and periodic maintenance and inspection in order to maintain the moiton controller in the best conditions.

3.6.1 Maintenance and Inspection

The I/O module mainly consists of semiconductor devices and its service life is semi-permanent. However, periodic inspection is requested for ambient environment may cause damage to the devices.
When inspecting one or two times per six months, check the following items.

Check Items		Judgment	Corrective Actions
Power supply		Within change rate of input voltage (Refer to general specifications)	Hold it with the allowable range.
Power supply for input/output		Input/Output specification of each module	Hold it with the allowable range of each module.
Operating atmosphere	Temperature	0 ~ + 55℃	Adjust the operating temperature and humidity with the defined range.
	Humidity	5 ~ 95% RH	
	Vibration	No vibration	Use vibration resisting rubber or the vibration prevention method.
There will be no shaking		No play allowed	Securely enrage the hook.
Check the connecting screws		Screws should not be loose.	Retighten terminal screws.
Spare parts		Check the number of Spare parts and their Store conditions	Cover the shortage and improve the conditions.

3.6.2 Daily inspection

The following table shows the inspection and items which are to be checked daily.

Check Items		Check Points	Judgment	Action
Connection conditions of base		Check the screws.	Screws should not be loose.	Retighten Screws.
Connecting conditions of terminal block or extension cable		Check the connecting screws	Screws should not be loose.	Retighten Screws.
		Check the distance between solderless terminals.	Proper clearance should be provided.	Check screw
		Input/output connector	Connector should not be loose.	Check screw
Display LED	PWR LED	Check LED ON	On (Off indicates an error)	
	RUN LED	Check that the LED is On during Run.	On (flickering or Off indicates an error)	
	ERR LED	Check that the LED is Off during Run.	Flickering indicates an error	
	Input LED	Check LED turns On and Off	On when input is On, Off when input is off.	
	Output LED	Check LED turns On and Off	On when output is On, Off when output is off	

3.6.3 Periodic Inspection

Check the following items once or twice every six months, and perform corrective actions as needed.

<table>
<tr><th colspan="2">Check Items</th><th>Checking Methods</th><th>Judgment</th><th>Action</th></tr>
<tr><td rowspan="3">operating atmosphere</td><td>Ambient temperature</td><td rowspan="3">Measured with a thermometer/hygrometer measure corrosive gas</td><td>0 ~ 55 °C</td><td rowspan="3">Adjust to meet general specification (environment standard in control panel)</td></tr>
<tr><td>Ambient humidity</td><td>5 ~ 95%RH</td></tr>
<tr><td>Pollution degree</td><td>There should be no corrosive gases</td></tr>
<tr><td rowspan="2">Controller Status</td><td>Looseness, Ingress</td><td>The module should be move the unit</td><td>The module should be mounted securely.</td><td>Retighten screws or hook</td></tr>
<tr><td>Dust or foreign material</td><td>Visual checks</td><td>No dust or foreign material</td><td>Place the product horizontally so that dust does not enter the ventilation holes, and remove dust or foreign material with a dry cloth. Be careful not to let foreign material into the ventilation holes.</td></tr>
<tr><td rowspan="3">Connection status</td><td>Loose terminal screws</td><td>Re-tighten screws</td><td>Screws should not be loose.</td><td>Tighten</td></tr>
<tr><td>Distance between terminals</td><td>Visual checks</td><td>Proper clearance</td><td>Check screw</td></tr>
<tr><td>Loose connectors</td><td>Visual checks</td><td>Screws should not be loose.</td><td>Retighten connector mounting screws</td></tr>
<tr><td colspan="2">Line voltage check</td><td>Measure voltage between input terminals</td><td>Refer to general specifications</td><td>Change supply power</td></tr>
</table>

3.7 Troubleshooting

The following explains contents, diagnosis and corrective actions for various errors that can occur during system operation.

3.7.1 Basic Procedure of Troubleshooting

System reliability not only depends on reliable equipment but also on short downtimes in the event of fault.
The short discovery and corrective action are needed for speedy operation of system. The following shows the basic instructions for troubleshooting.

(1) Visual check
Check the following points.
- Machine operating condition (in stop and operation status)
- Power On/Off
- Status of I/O devices
- Condition of wiring (I/O wires, extension and communications cables)
- Display states of various indicators (such as POWER LED, RUN LED, ERR LED and I/O LED). After checking them, connect peripheral devices and check the operation status of the motion controller and the program contents.

(2) Error Check
Observe any change in the error conditions during the following.
- Switch to the STOP position, and then turn the power on and off.

(3) Range limit
Estimate what is the trouble because using the above method.
- Inside or outside of the motion controller? I/O module or another module?
- Is it input/output? Is it others?
- Motion program?

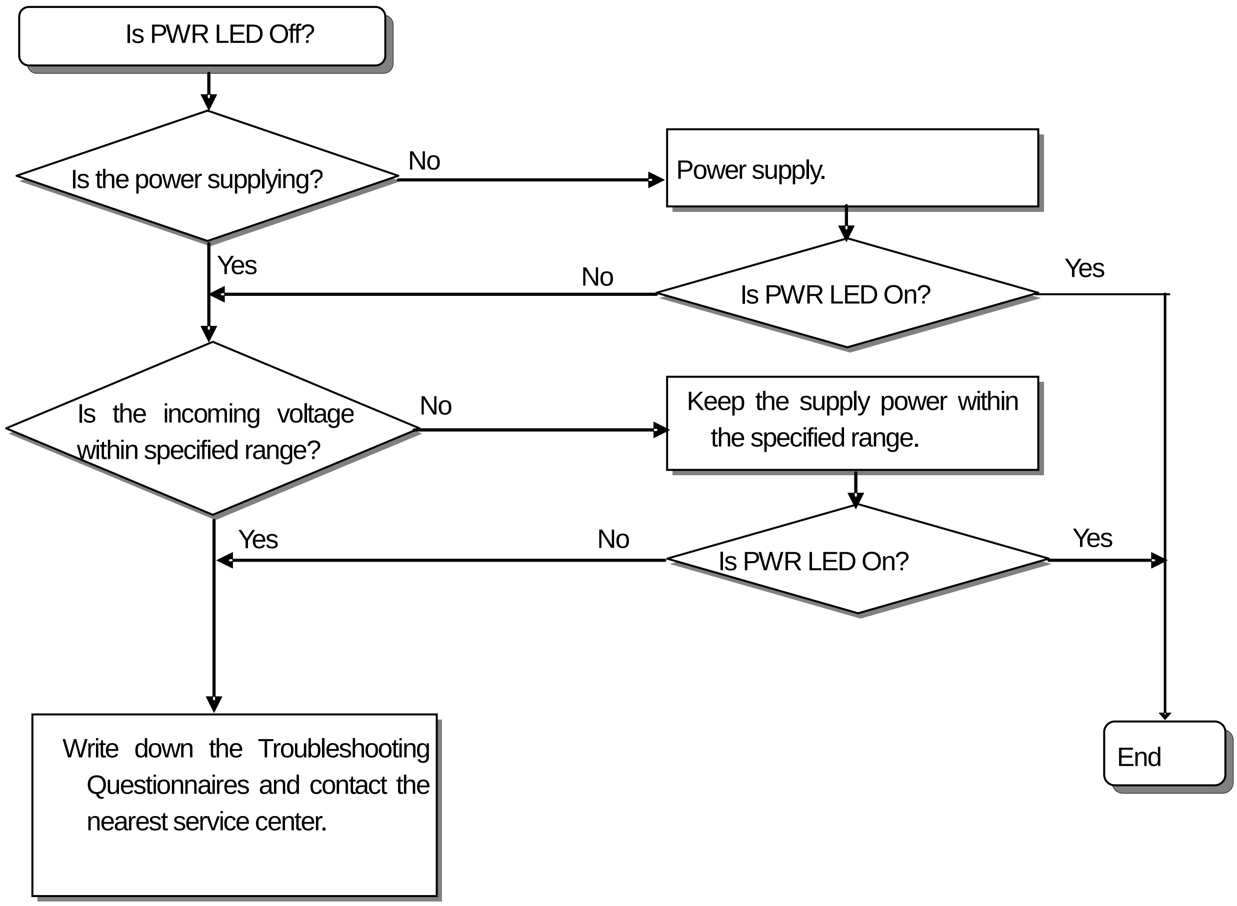
3.7.2 Troubleshooting

This section explains the procedure for determining the cause of troubles as well as the errors and corrective actions.

Trouble contents		
When Power (PWR) LED is Off	⇨	Measure when Power LED is off
Err. LED flicker codes	⇨	Measure when ERR LED is blinking
When RUN, STOP LED is OFF	⇨	Measure when RUN, STOP LED is OFF
In case the input/output module operates abnormally	⇨	Measure in case the I/O module does not operate normally
When PROGRAM WRITE does not work	⇨	Measure when PROGRAM WRITE does not work.

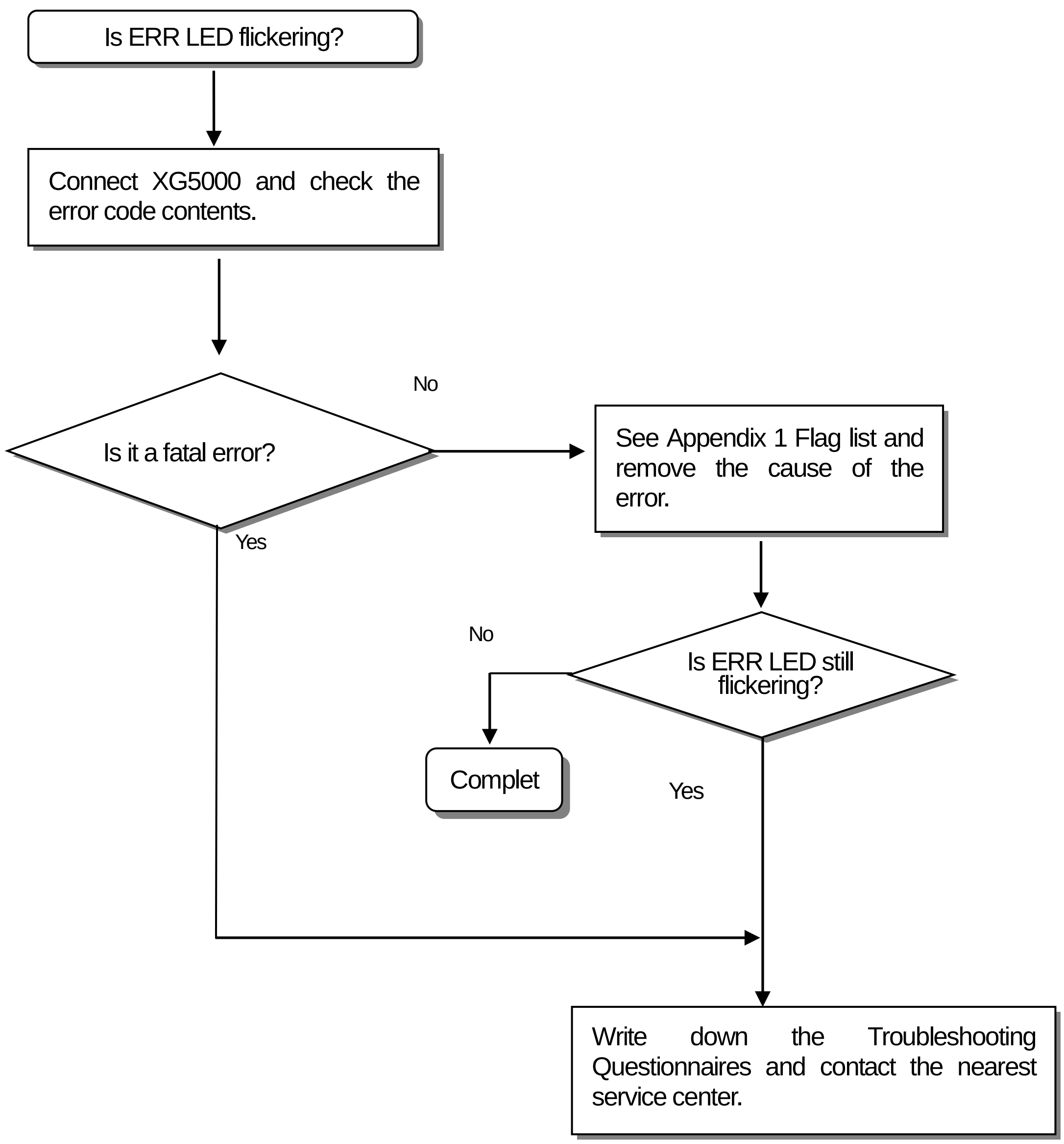
(1) Measure when PWR(Power) LED is off

Flowchart used when the POWER LED is turned Off.



(2) Measure when ERR(error) LED is blinking

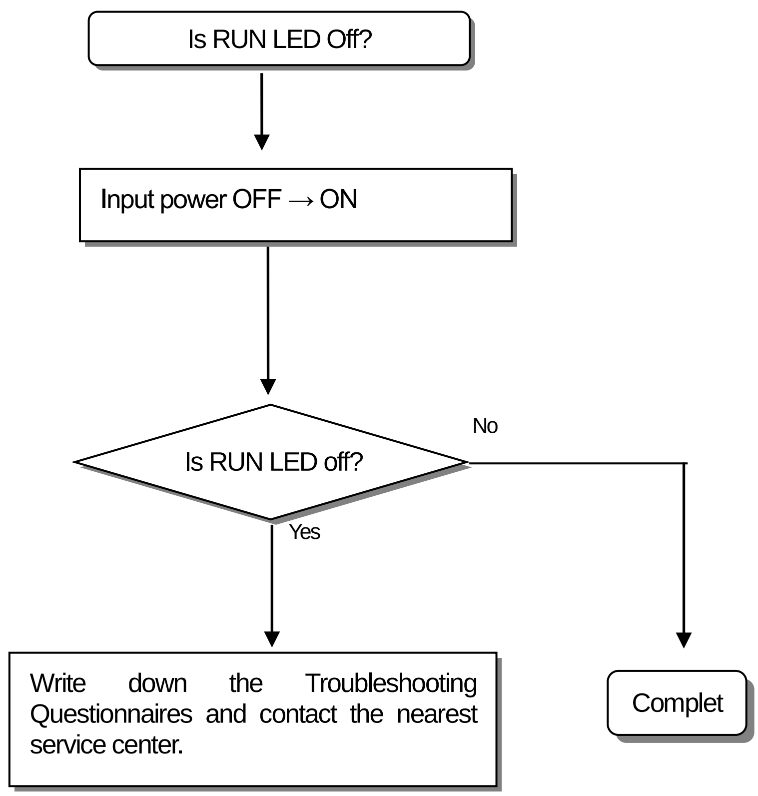
Here describes the measure procedure when ERR LED is flickering during power supply, or when operation starts, or during operation.



Notes

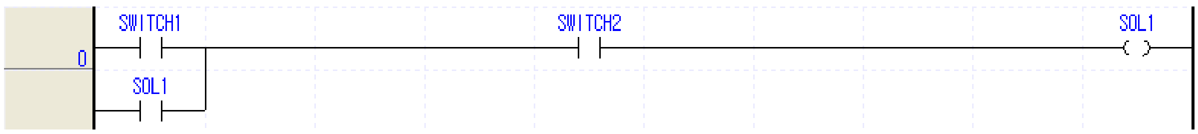
In case of minor error, and the does not stop but you should check the error contents promptly and take an action. If not, it may cause the critical error.

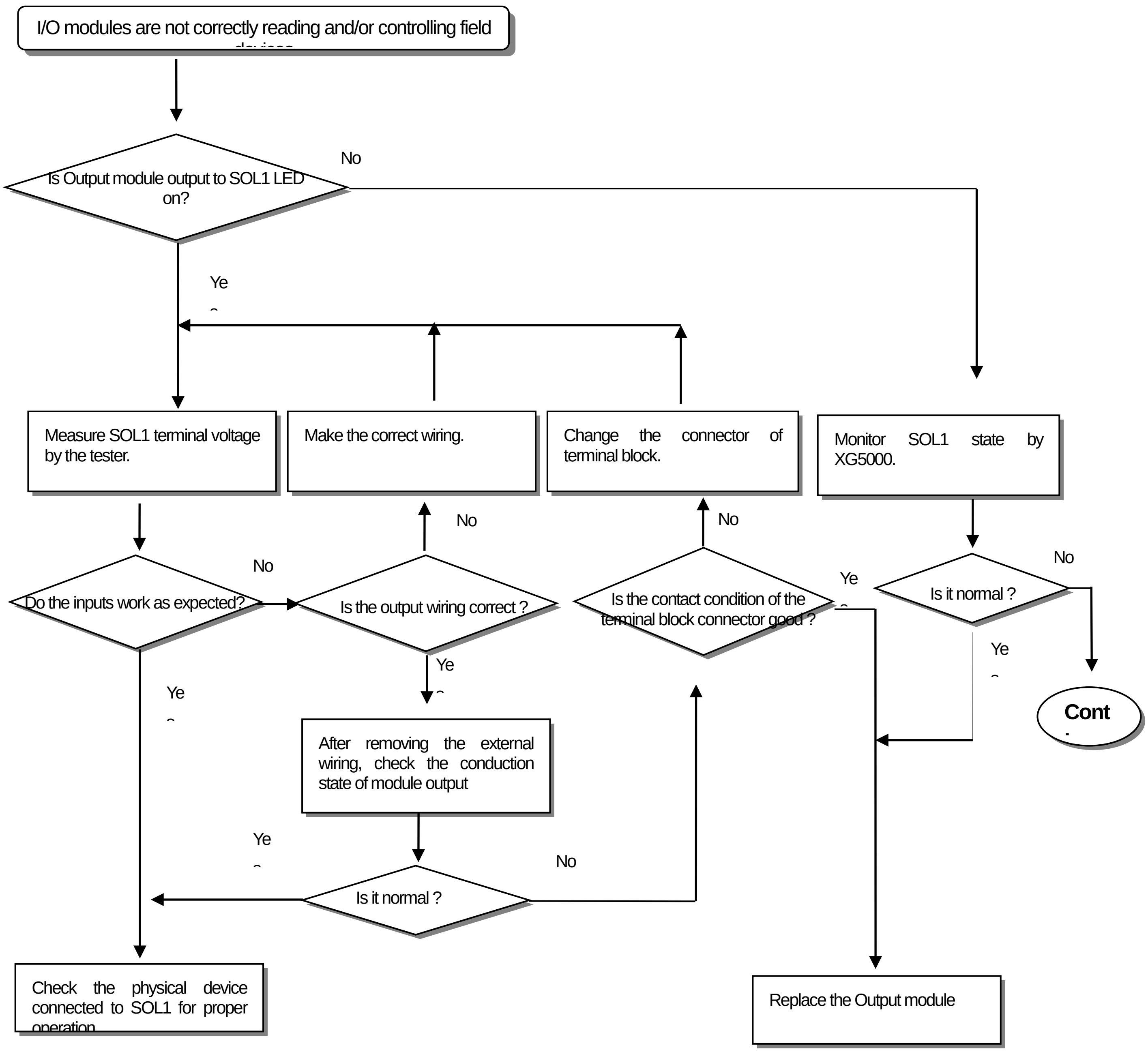
(3) Use PLC RUN LED Off flowchart on page 10-8
This section describes the action sequence when the RUN LED is turned off when the power is turned on, when the operation starts, or during operation.

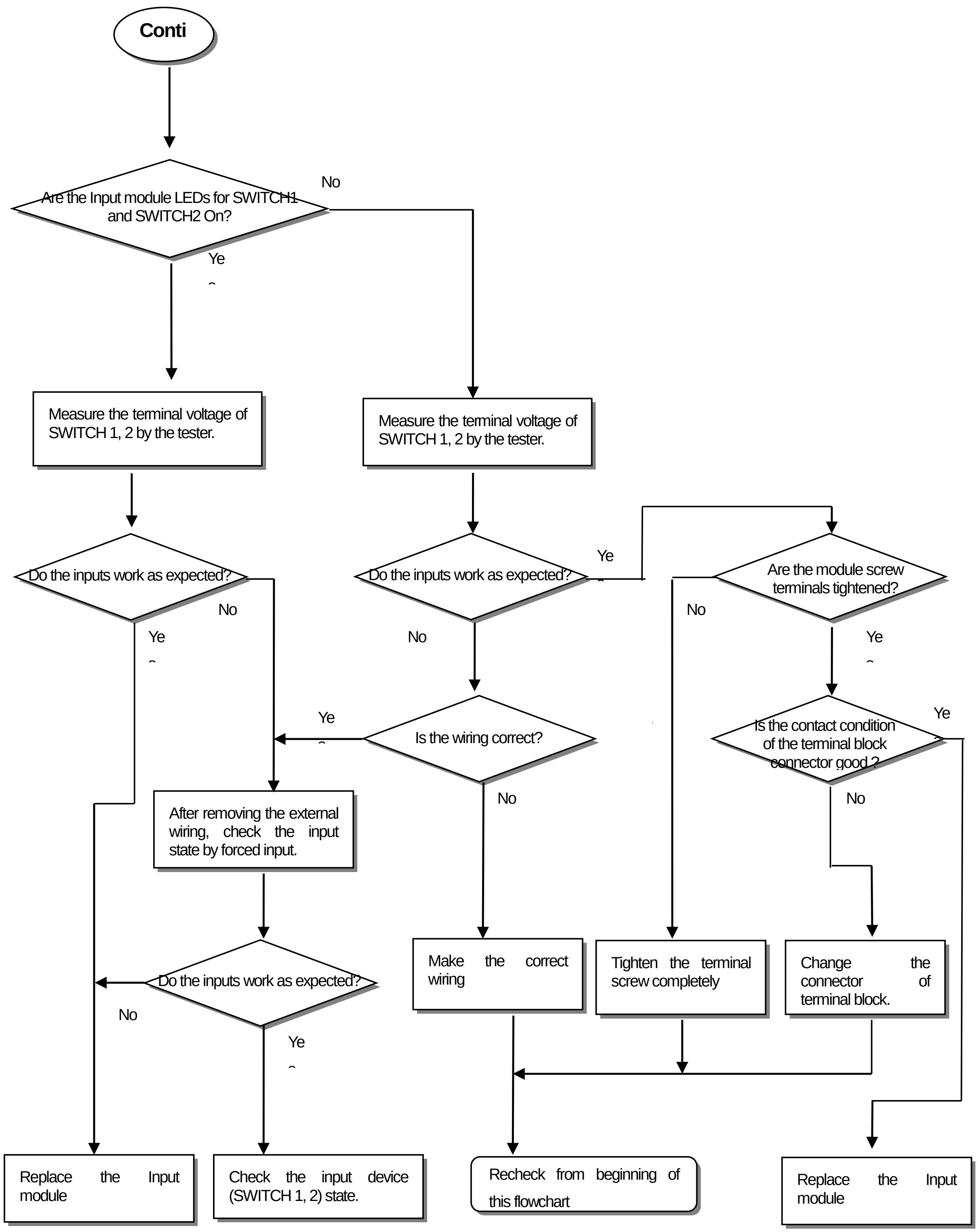


(4) Measure in case the I/O module does not operate normally

Here describes the measure procedure when I/O Module does operation normally during operation, as shown on the program example below.






Conti
Are the Input module LEDs for SWITCH1 and SWITCH2 On?
No
Ye
Measure the terminal voltage of SWITCH 1, 2 by the tester.
Measure the terminal voltage of SWITCH 1, 2 by the tester.
Do the inputs work as expected?
No
Ye
Do the inputs work as expected?
Ye
No
Are the module screw terminals tightened?
No
Ye
Is the wiring correct?
Ye
No
Is the contact condition of the terminal block connector good ?
Ye
No
After removing the external wiring, check the input state by forced input.
Do the inputs work as expected?
No
Ye
Make the correct wiring
Tighten the terminal screw completely
Change the connector of terminal block.
Replace the Input module
Check the input device (SWITCH 1, 2) state.
Recheck from beginning of this flowchart
Replace the Input module

3.7.3 Troubleshooting Questionnaire

If the trouble occurs when using motion controller, fill in the following questionnaire and then contact customer's service center by phone or FAX.

- For errors relating to special or communication modules, use the questionnaire included in the User's manual of the unit.

1. Customer's Contact Number: Phone) ____________________
   FAX) ____________________
2. Module type: ( )
3. Applied Module details
   Controller model ( ) OS version No. ( ) Serial No. ( )
   – XG5000 Version number used for program compiling :( )
4. General description of the device or system used as the control object:

5. The module type of the motion controller:
   – Operation by the mode setting switch ( ) – Operation by the XG5000 or communications ( )
6. LED of the motion controller turned Off ? Yes ( ), No ( )
7. Error message content by XG5000:
8. Action method for the error code in the article 7:
9. Trouble shooting method for other error action:
10. Characteristics of the error
    - Repetitive ( ): Periodic ( ), Related to a particular sequence ( ), Related to environment ( )
      Environment related ( )
    - Sometimes ( ): General error interval:
11. Detailed description for the error phenomena:

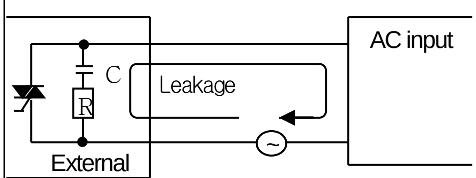
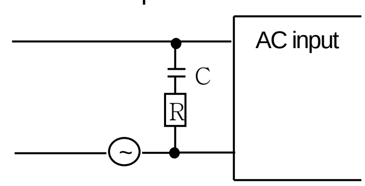
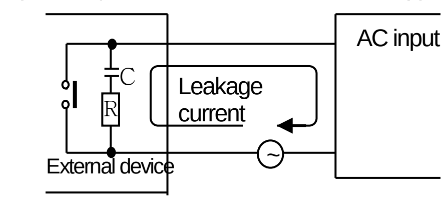
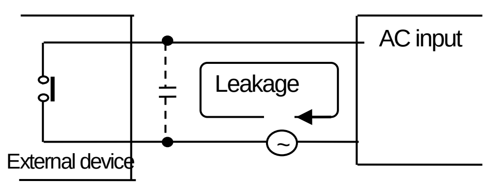
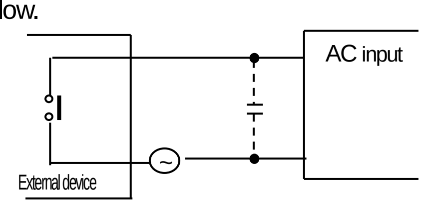
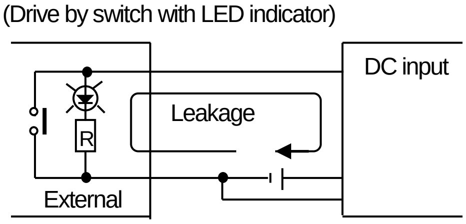
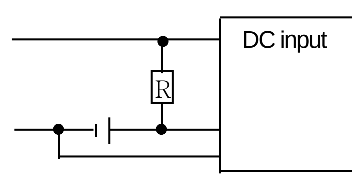
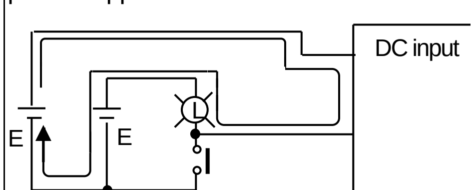
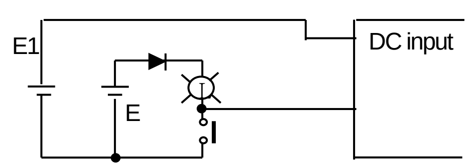
12. Configuration diagram of applied system:

3.7.4 Cases

Input circuit troubles and corrective actions

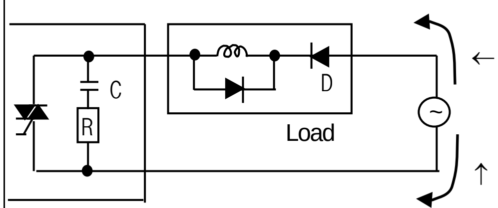
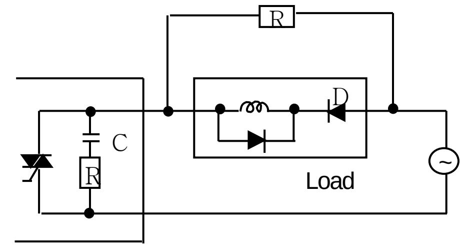
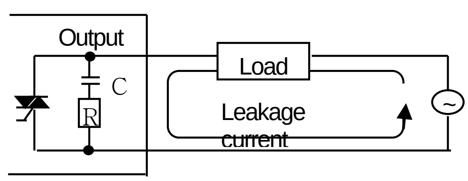
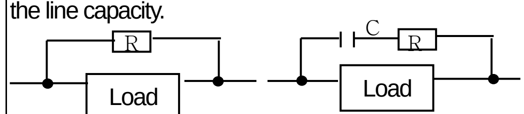
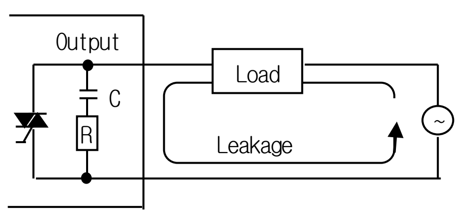
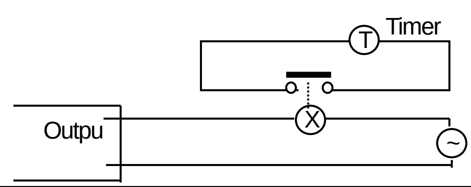
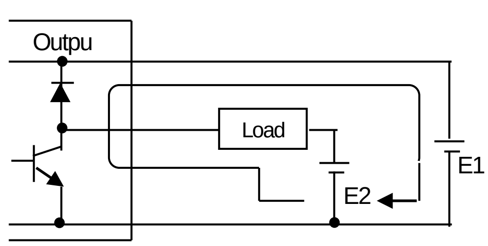
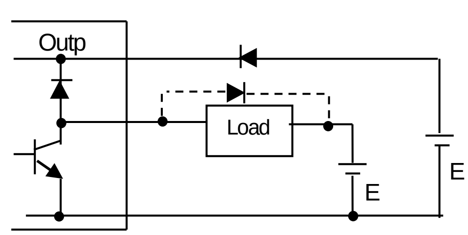
(1) Input Circuit Error Type and Corrective Actions

The followings describe possible troubles with input circuits, as well as corrective actions.

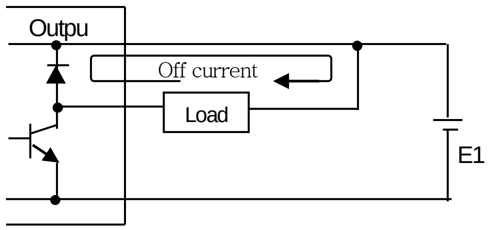
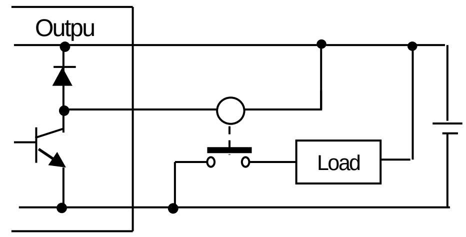
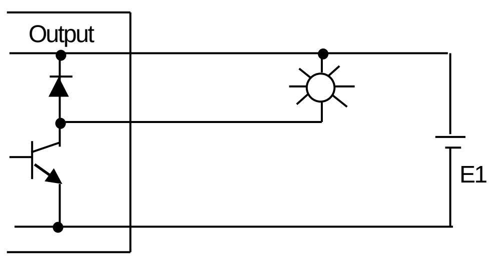
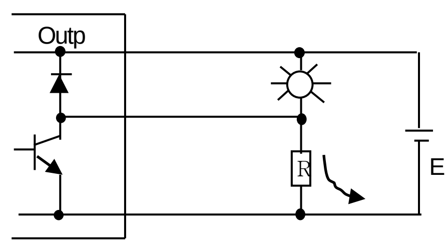
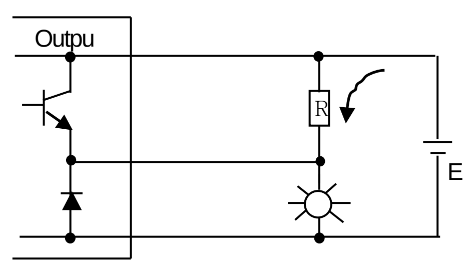
Phenomena	Causes	Measures
Input signal does not turn off	Leakage current of external device (Such as a drive by proximity switch)  	•Connect an appropriate register and capacity, which will make the voltage lower across the terminals of the input module. 
Input signal not OFF(Neon lamp may be still on)	Leakage current of external device (Drive by a limit switch with neon lamp)  	•CR values are determined by the leakage current value. – Recommended value C : 0.1 ~ 0.47㎌ R : 47 ~ 120 Ω (1/2W) or make up another independent display circuit.
Input signal doesn't turn off.	Leakage current due to line capacity of wiring cable.  	•Install the power on the external device as below. 
Input signal doesn't turn off.	Leakage current of external device (Drive by switch with LED indicator)  	•Connect an appropriate register, which will make the voltage higher than the OFF voltage across the input module terminal and common terminal. DC input 
Input signal doesn't turn off.	• Loop current due to the use of two different power supplies.  •E1 > E2, looped	•Use only one power supply. •Connect a sneak current prevention diode. 

(2) Output Circuit Error Type and Corrective Actions

Here describes the trouble examples of output circuit and its measures.

<table>
<tr><th>Phenomena</th><th>Causes</th><th>Measures</th></tr>
<tr><td>Over voltage applied to the load if case of output contact OFF</td><td>•Load is half-wave rectified inside
(in some cases, it is true of a solenoid)
•If power polarity is ←, C is charged, if polarity is ↑, voltage charged to C + power voltage is applied to both sides of diode (D). Max. Voltage is about 2□ 2.
Max. Voltage is about 2√2.

Note) If used as above, output element does not make trouble but the function of diode (D) built-in the load becomes low which causes the trouble.</td><td>•A resistor of tens of kΩ to hundreds of kΩ is connected in parallel to the load
</td></tr>
<tr><td>The load does not turn off.</td><td>•Leakage through a surge suppressor connected in parallel to the load.
</td><td>•Connect C and R across the load, which are of registers of tens KΩ .
When the wiring distance from the output module to the load is long, there may be a leakage current due to the line capacity.
</td></tr>
<tr><td>When the load is C–R timer, time constant fluctuates.</td><td>•Leakage through a surge suppressor connected in parallel to the load.
</td><td>•Drive the relay using a contact and drive the C–R type timer using the since.
•Use other timer than the C–R contact some timers.
Note) Have half-ware rectified internal circuits therefore, be cautious.
</td></tr>
<tr><td>The load does not turn off.(for DC)</td><td>•Circulating current by using two different power sources

•E1< E2, looped
•E1 is off (E2 is on), looped.</td><td>•Use only one power supply.
•Connect a sneak current prevention diode.

If the load is the relay, etc, connect a counter-electromotive voltage absorbing code as shown by the dot line.</td></tr>
</table>

Output circuit troubles and corrective actions (continued).

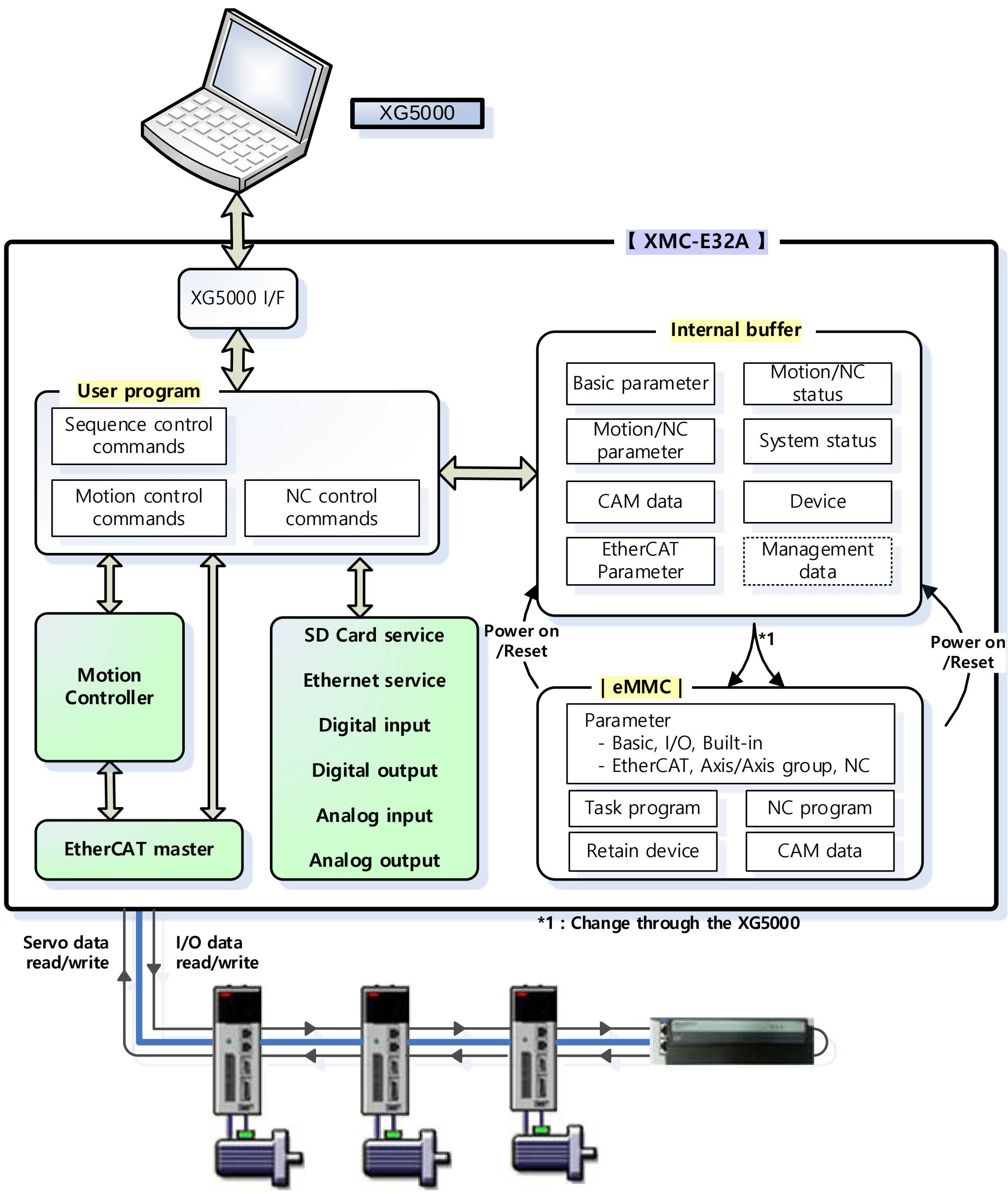
Phenomena	Causes	Measures
The load off response time is long. .	•Over current at off state [The large solenoid current fluidic load (L/R is large) such as is directly driven with the transistor output.  •The off response time can be delayed by one or more second as some loads make the current flow across the diode at the off time of the transistor output.	•Insert a small L/R magnetic contact and drive the load using the same contact. 
Output transistor is destroyed.	Surge current of the white lamp on.  A surge current of 10 times or more when turned on.	•To suppress the surge current make the dark current of 1/3 to 1/5 rated current flow.  Sink type transistor output  Source\ type transistor output

Chapter 4 Motion Control Operation

This chapter describes structure, parameter and device of motion controller.

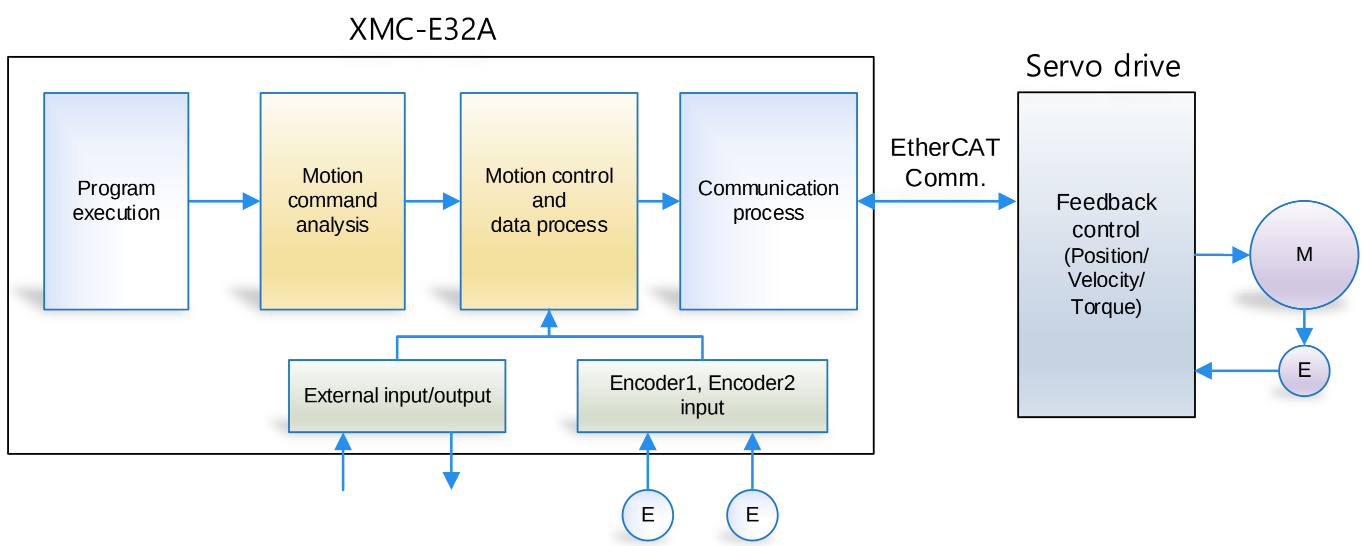
4.1 Structure of Motion Controller

This picture describes process of parameter and operation data saved in the controller.



4.2 Motion Control Configuration

Motion controller can control up to 32 axes of actual motor axis and 4 virtual axes through EtherCAT. Among 32 axes, you can control the axes that are not connected to the slave by setting them as virtual axes and 4 axes are provided for the virtual axes only. Motion control block diagram of motion controller is shown below.



4.3 Motion Control Tasks

The following describes tasks of the motion controller.

4.3.1 Task Types

There are 3 types of motion control tasks: main task, periodic task and initialization task.

The main task completes the motion within the period set by the user, and it performs I/O refresh, program process, motion control and processes EtherCAT synchronous communication. The set period of the main task is 0.5/1/2/4ms and it can be set in the basic parameter of the motion controller. (For smooth motion control, the maximum main task is limited to 4 ms.)

The period of the periodic task can be set in multiples of the main task's period set by the user, and the periodic task is processed in the remaining time after the main task is completed during the period of each task. Therefore, the periodic task can be performed over a number of main task periods.

The initialization task is only performed once at the beginning when the motion controller is entering the RUN mode, and it is normally used for setting the initial data of the system and the parameter.

Task types	Number of Programs	Operation
Main task	Up to 256	· It performs I/O refresh, processing of programs assigned to main task and motion control. · It performs the above tasks at a time for each of the established control period (main task cycle). · It has higher priority than periodic task. · It uses programs that require synchronized control and high-speed operation processing through allocation since it is possible to process program fast. · Period possible to be set: 0.5ms, 1ms, 2ms, 4ms
Periodic task		· It performs processing of programs assigned to main task. · It is performed for the remaining time after implementation of main task operation within the control period, and can be performed over multiple cycles. · Since it has lower priority than main task in the execution of motion control commands within main task program, the motion control commands executed in the main task program are processed first. · It uses programs of processing other monitoring data and control of device that doesn't require high-speed processing through allocation. · Period possible to be set: 1ms ~ 100ms (Set to a multiple of the main task cycle)
Initial Task		· It performs processing of programs assigned to the initialization task after implementing I/O refresh. · If the initial task completion (_INIT_DONE) flag is set by the initialization task program, the task is completed, and the execution of the main task and periodic task program starts.

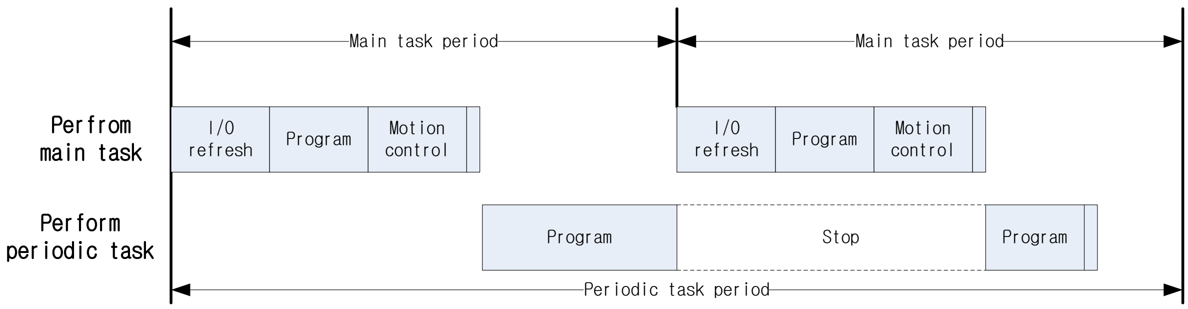
Notes

If the main task cycle is set outside the setting range, an error 0x0260 occurs.
If the periodic task cycle is not set to the multiple of the main task, an error 0x0261 occurs.
If the error occurs, check the task cycle.

4.3.2 Task Operation

(1) Overall task operation

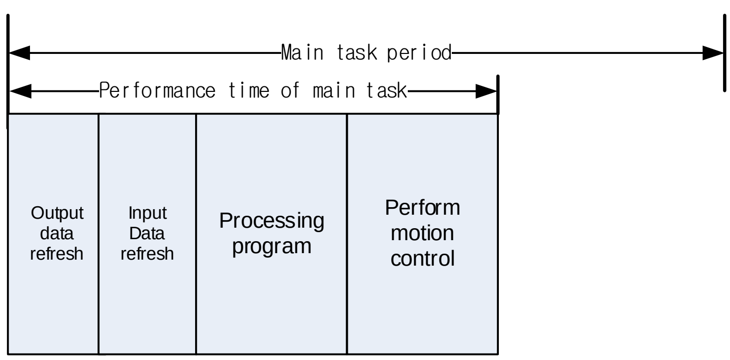
The task is composed of the main task and periodic task. The main task performs I/O refresh and processes program as well as motion control motion according to the processing of the program during the control period. The periodic task is performed in the control period in the remaining time after the main task is completed and it can be completed after going through many control periods.



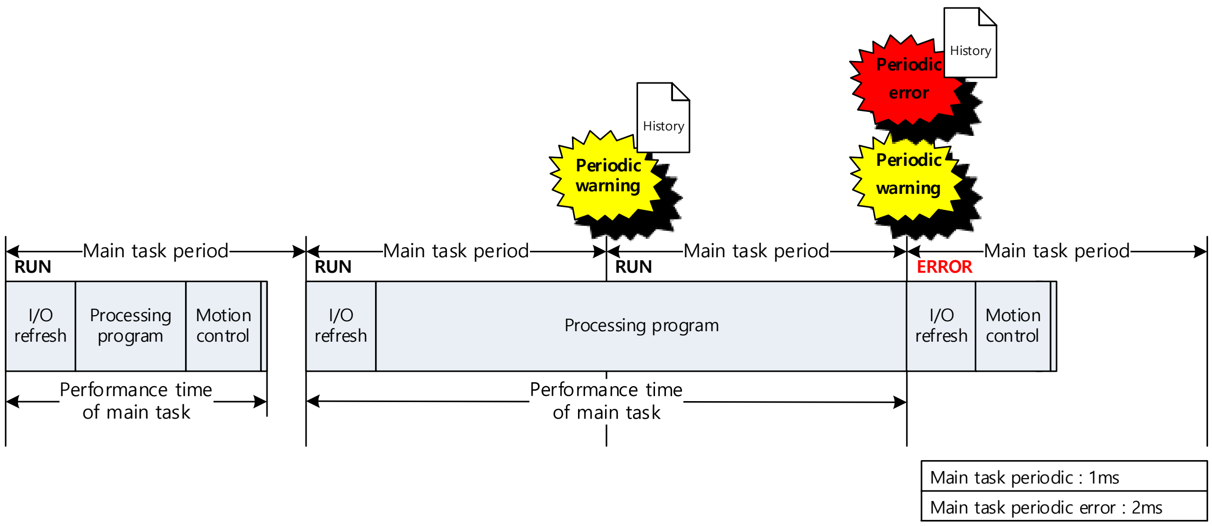
(2) Main task operation

The main task must be performed in the set task period, and if the performance of the main task exceeds the set main task period, an error occurs and if motion controller is in RUN state, it is changed to STOP state. If the main task execution is not completed during the 'main task cycle error' detection time, the operation is stopped immediately, and an error is generated if the motion controller is in the RUN state. The motion controller enters the ERR state.

1) Performance time of main task ≤ Main task period



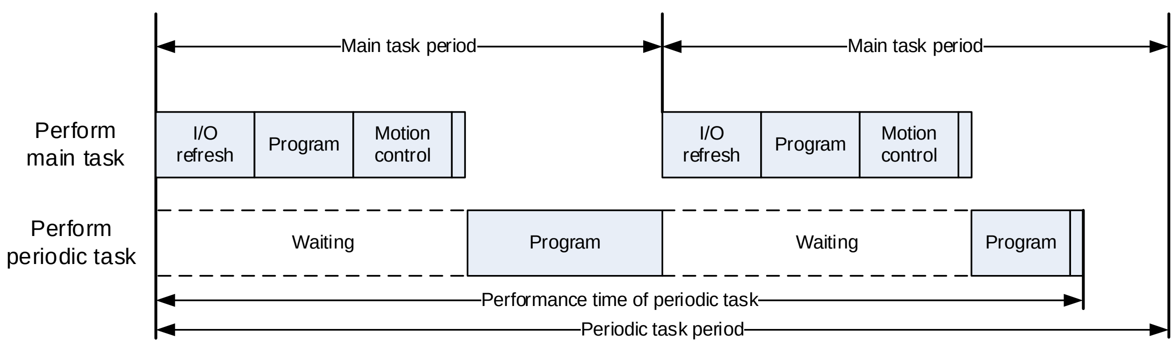
2) Performance time of main task > Main task period



(3) Cycle task operation

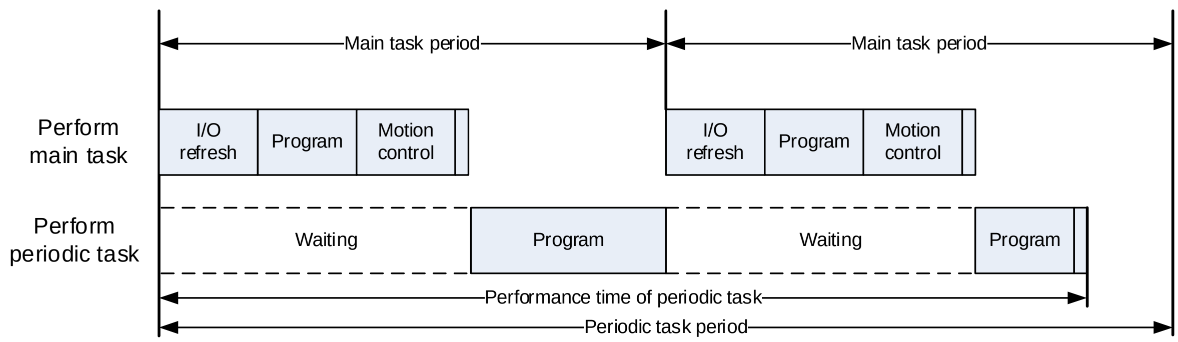
The periodic task is performed in the remaining time after performing the main task in the set control period and it can be performed over many control periods depending on the performance time of the task If the execution of periodic task exceeds the set periodic task cycle, a warning occurs. If the periodic task execution is not completed during the 'periodic task cycle error' detection time, the operation is stopped immediately, and an error is generated if the motion controller is in the RUN state. The motion controller enters the ERR state.

1) Performance time of periodic task ≤ Periodic task period



(4) Initial task operation

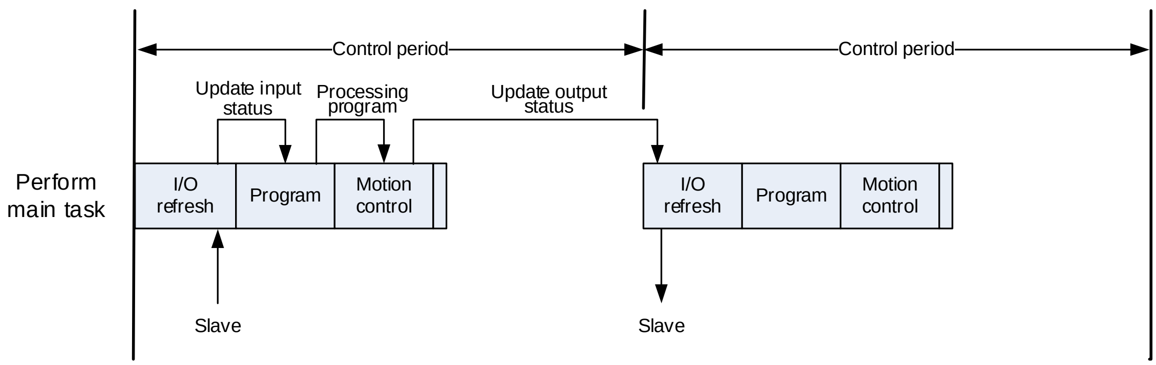
The initialization task is a task performed only once at the beginning when motion controller is entering the RUN mode. It is mainly used to set the initial data of the system and the parameter. The initialization task is executed until the initialization task execution completion (_INIT_DONE) flag is set before the execution of the main task, and it is terminated when the user sets the _INIT_DONE flag in the program. Only when the initialization task is terminated, the main task and periodic task programs are executed. The initialization task cycle inherits the main task cycle. The initialization task operates in the main task cycle and is included in the main task execution time.



4.3.3 Motion command execution

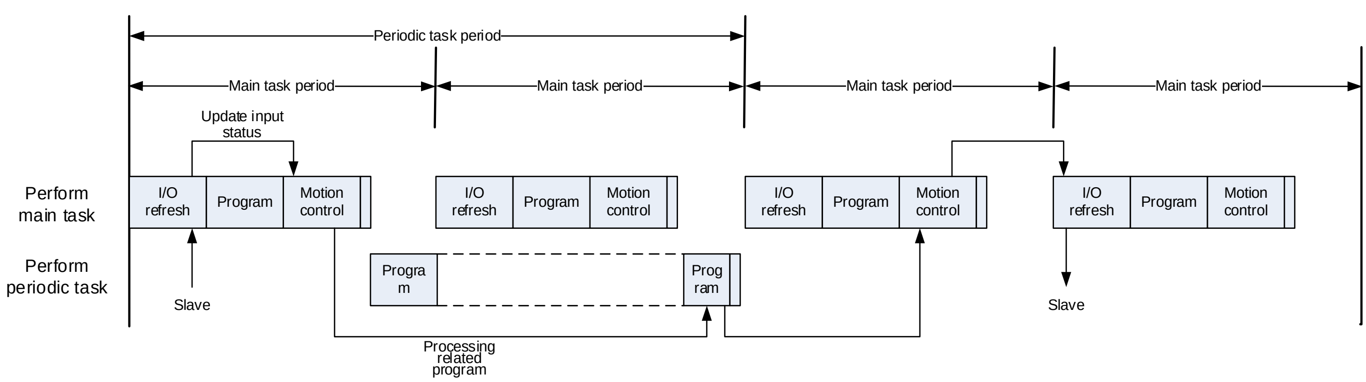
(1) Execution of motion commands in the main task

Execution of motion instruction of the main task is shown in the figure below. According to the I/O refresh motion of the main task, the input value of slave and the system parameters are updated. Motion control is performed in the main task based on this information. The outcome of the performance is output in slave module at the I/O refresh time of the next control period.



(2) Execution of motion commands in the periodic task

Execution of motion instruction in the periodic task is shown in the figure below. According to the I/O refresh motion of the main task, the input value of slave and the system parameters are updated. With this information, motion control is performed in the main task. The program of the periodic task is performed by this result, and motion control is performed with this result while the main task is being performed in the control period after the performance of the periodic task. Also the outcome of this motion control performance is output in slave at the I/O refresh time of the next control period



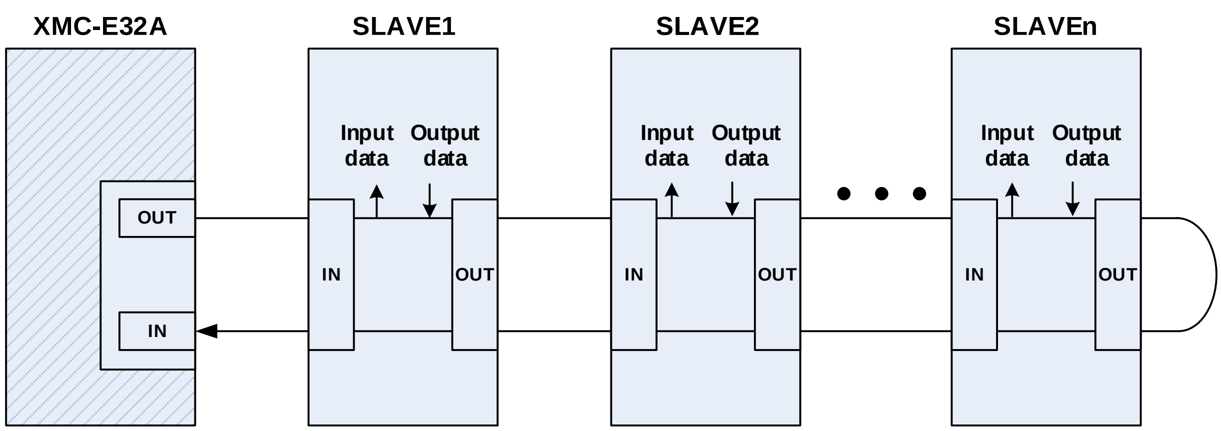
If the periodic task program is executed within several main tasks, the motion control result of each main task is reflected in the periodic task program. Please keep this in mind when writing the periodic task program.

4.4 EtherCAT Communication

The communication of EtherCAT(Ethernet for Control Automation Technology) is explained here.

4.4.1 What is EtherCAT

EtherCAT is a high-performance industrial network system which uses Real-Time Ethernet based on the Ethernet developed by Beckhoff Company in Germany. EhterCAT is a communication between the master and the slave, and it provides a short communication cycle time by transmitting Ethernet Frame at a high speed between each nodes. When data Frame transmitted from the master to the slave passes through the slave, EtherCAT communication sends the received data to the relevant data Frame at the same time as the slave receives the transmission data. In other words, EtherCAT passes one communication frame to all slaves in turn without transmitting data to each slave node in the network, When a communication frame passes through each slave, each slave reads and writes data to its own area in the frame. Communication Frame goes through the last slave and returns, passes through all slaves and is delivered back to the master, and high-speed data transmission is performed.



4.4.2 CoE(CANopen over EtherCAT)

Motion controller uses the slave and EtherCAT to communicate and uses CoE(CANopen over EtherCAT) as the protocol for information exchange.

In CoE, parameter and data information of the slave are composed of Object Dictionary. Object Dictionary contains the information used in the configuration of the device and communication, and it is a group of the object (parameter) which can be accessed through the network.

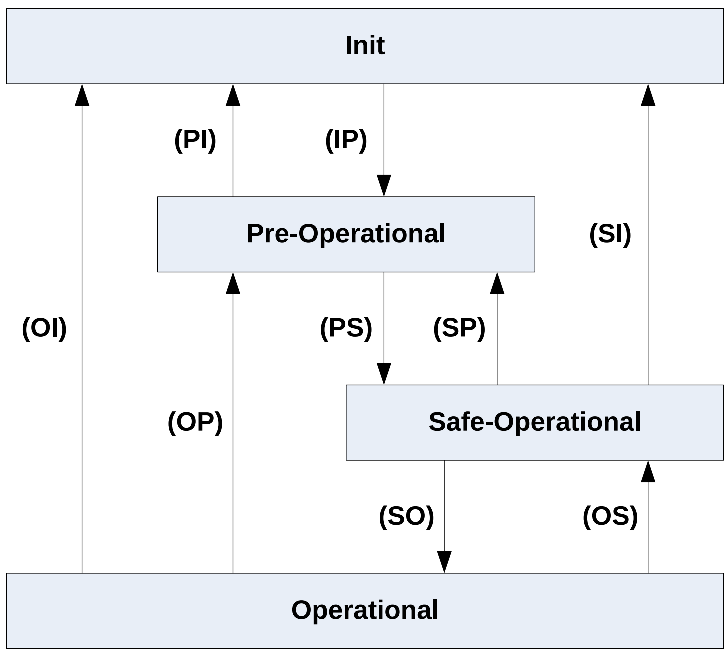
In the communication between master-slave using CoE, there are a communication which uses Process Data Object (PDO) and synchronously transmits information, and a Service Data Object (SDO) communication which occurs asynchronously.

Motion controller regularly performs process data communication to receive and send input/output signal and to control the position of EtherCAT slave (servo drive). It also performs service data communication in terms of an error state in the slave and the parameter reading/writing whenever there is a request.

Type	Communication time	Content
Process Data Communication (PDO Communication)	Synchronous (main task period)	servo drive position control data, input/output of data, etc.
Service Data Communication (SDO Communication)	Asynchronous (in request)	servo parameter reading/writing, servo error information reading, etc.

4.4.3 EtherCAT State Machine

The state and motion between states of EtherCAT communication are shown in the figure below.



The communication between the master-slave of EtherCAT communication begins from the Initial state and progresses to the Operational state. In the motion controller, the slave servo drive can be controlled with a normal process data communication when it is in operational state.

If a communication error occurs while the motion controller performs the slave and EhterCAT communication at operational state, the communication state is changed to the Initial state and the communication between the slaves is discontinued. In this case, the factor of communication error should be removed and reconnect with the slave to restart the communication.

4.4.4 EtherCAT Process Data Objective(PDO)

The synchronous data communication in EtherCAT communication of motion controller occurs through process data object (PDO). TxPDO which is transmitted from the slave to motion controller, and RxPDO which is transmitted from motion controller to the slave. In RxPDO and TxPDO, data which are going to be synchronous communication can be put together to be set as the example of the figure below shows among the data defined in the Object Dictionary.

Object Dictionary

Index	Sub-Index	Description	Type
0x6040	-	Control Word	UINT
	⋮		
0x6060	-	Mode of Operation	BYTE
	⋮		
0x607A	-	Target Position	DINT

RxPDO

Index	Sub-Index
0x6040	0x00
0x6060	0x00
0x607A	0x00

Object Dictionary

Index	Sub-Index	Description	Type
0x6041	-	Status Word	UINT
	⋮		
0x6061	-	Operation Display	BYTE
	⋮		
0x6064	-	Actual Position	DINT

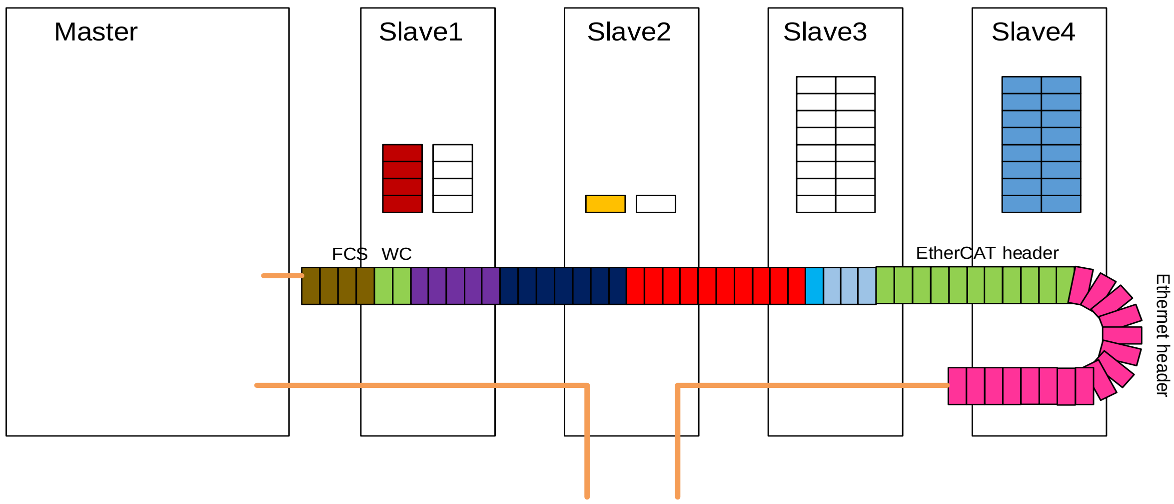
TxPDO

Index	Sub-Index
0x6040	0x00
0x6064	0x00
0x6061	0x00

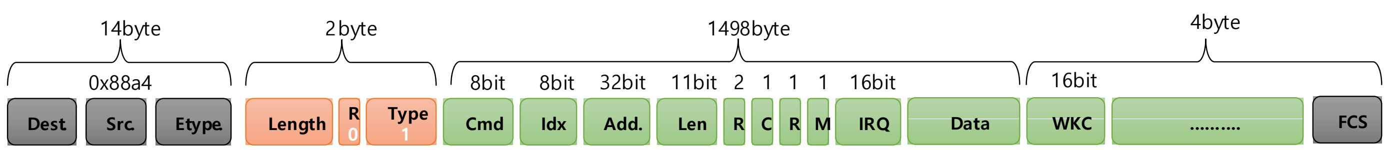
Slave manufacturers sometimes set many RxPDO and TxPDO in advance and provide Slave Information File including this information in xml format. When initially setting and test operating this slave information file, it should be transmitted to the motion controller using the XG5000. This slave information file should be analyzed and communicated to the PDO data which is optimized for controlling.

4.4.5 MultiFrame setting function

In EtherCAT communication, “frame” refers to a set of data transmitted once per cycle when exchanging data between master and slave during communication. In the figure below, the buffer moving along the yellow line, that is, a set of contiguous blocks is a frame.

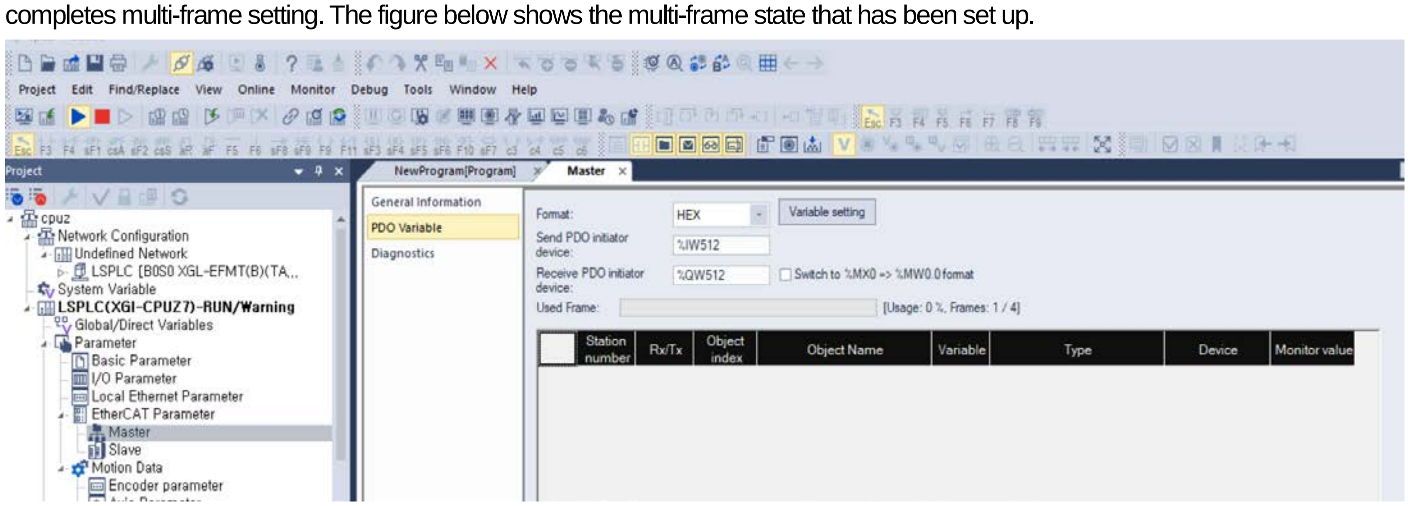


In general, the exchange data size between the master and the slave is often small. In this case, the master performs Ethernet communication with the slave in a single frame. At this time, considering the essential elements, the single frame has a specification consisting of up to 1518 bytes (14+2+1498+4). The figure below shows the configuration of a single frame.



On the other hand, if the number of slaves to communicate increases or the number of process data increases, the data size to be exchanged exceeds the standard size of 1518 bytes of single frame. At this time, data exchange is performed by transmitting multiple frames by composing an additional frame. At this time, multiple frames are called multi-frames.

When multi-frame is set by user setting multiple slaves and a sufficient number of PDOs, the program utility automatically completes multi-frame setting. The figure below shows the multi-frame state that has been set up.



Multi-frame is a convenient function that can be processed at once when there is a lot of data to be exchanged with the slave.

4.4.6 Specification of Motion Controller EtherCAT Communication

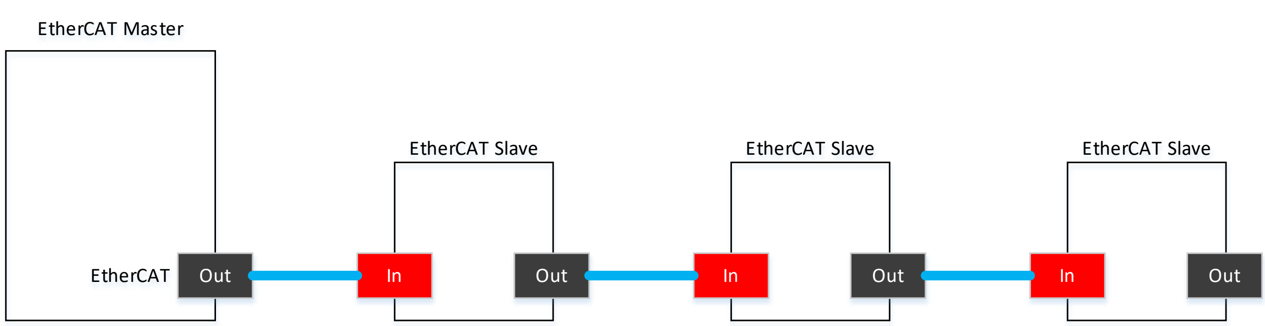
Item	Specifications
Communication protocol	EtherCAT
Support specification	CoE(CANopen over EtherCAT)
Physical layer	100BASE-TX
Communication speed	100Mbps
Topology	Daisy Chain
Communication cable	Cat. 5 STP(Shielded Twisted-pair) cable
Number of maximum slave	64(Able to mapping Max. 32 drive to motion axis)
Communication period	0.5ms/1ms/2ms/4ms (According to the main task cycle)
Synchronous Jitter between slave	0.5ms/1ms/2ms/4ms
Synchronous communication	PDO(Process Data Object) Mapping through CoE
Non-synchronous communication	SDO(Service Data Object) communication through CoE
Communication setting	Set the communication configuration using XG5000

4.4.7 EtherCAT Network Connection

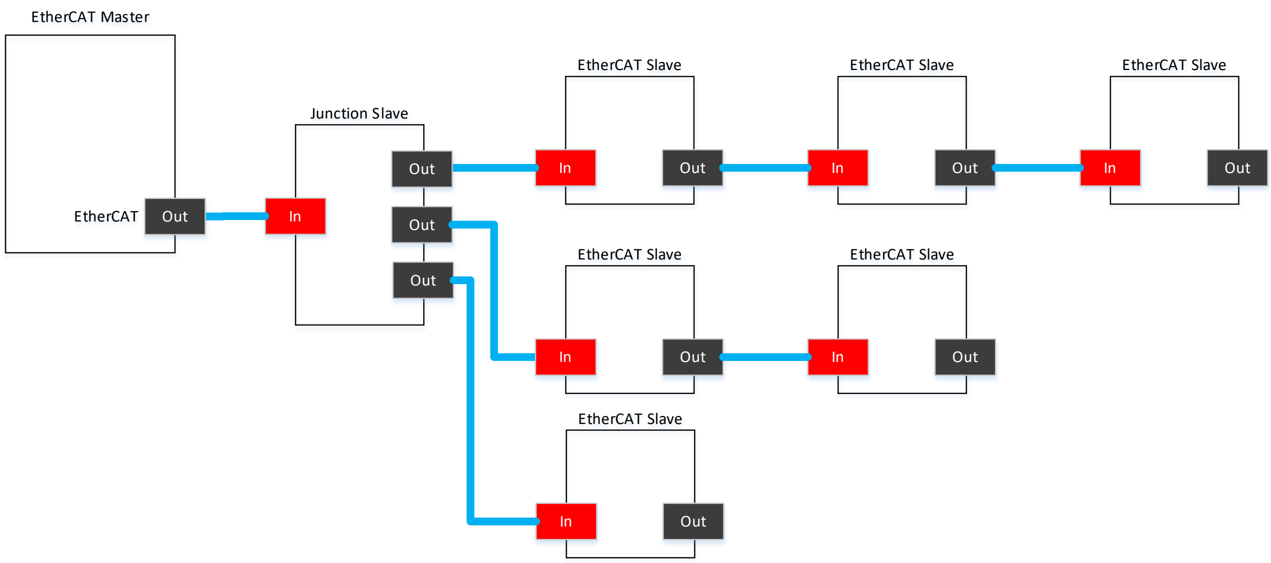
(1) Supported Topology

XMC generally constitutes a network using a daisy chain connection. It can constitute branches using junction slaves. XMC also can use cable duplication by constituting a network with rings using junction slaves.

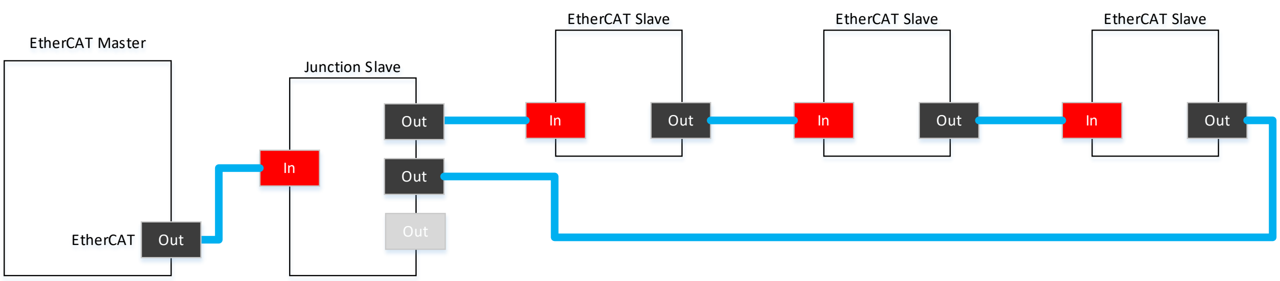
1) Not-using Branches



2) Using Branches



3) Cable Duplication Configuration



(2) Precautions

1) When connecting to EtherCAT slaves, be careful of not connecting to In/Out in reverse.

2) Equipment with several junction slaves is not allowed.

3) When using cable duplication with junction slaves, do not use the remaining ports of the junction slave.

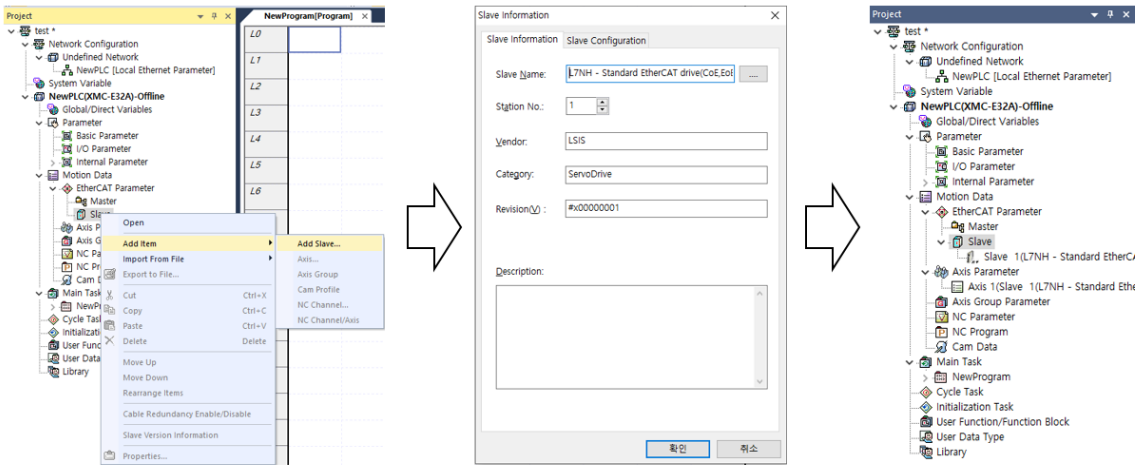
4) Junction slaves must use products that provide the EtherCAT reference clock.

5) When connecting to EtherCAT, configuration of the slave setting should be identical with the real configuration.

4.4.8 EtherCAT Network Setting

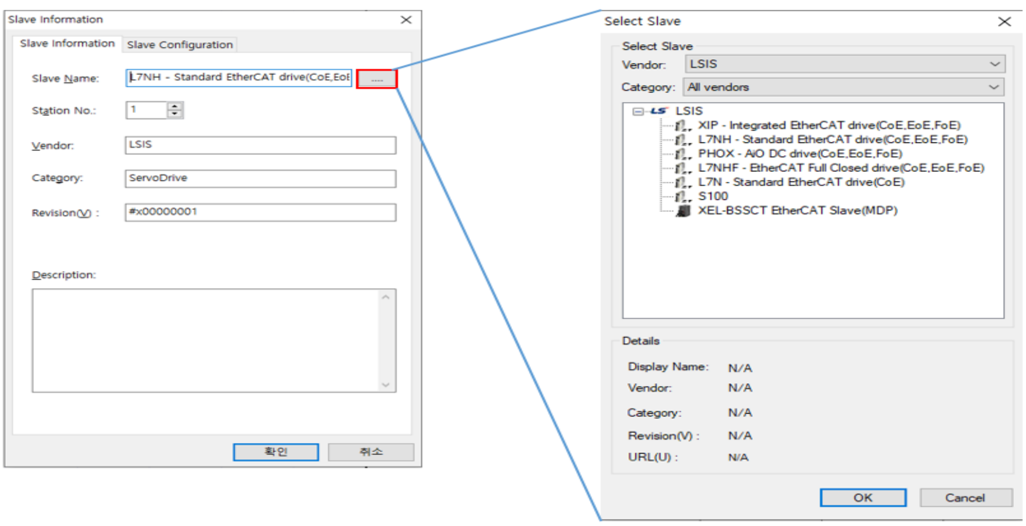
(1) Set Network in the Project Tree

1) To add slaves on the EtherCAT network, select “Add Items - Add Slaves” by clicking on the right mouse button over slaves in the project tree.



2) When the slave information window displays, check a slave name, select the confirmation button and add slaves.

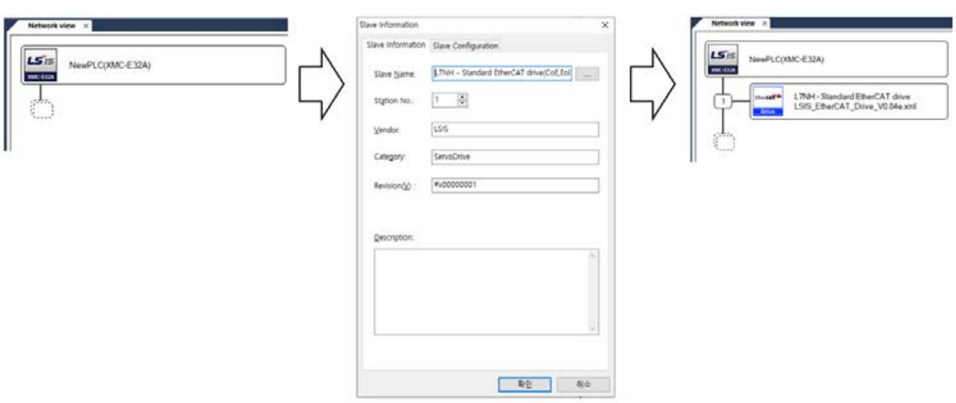
3) If adding other slaves, select slaves in the slave selection window by clicking the “...” button next to a slaves name.



4) Check if slaves are correctly added in the project window.

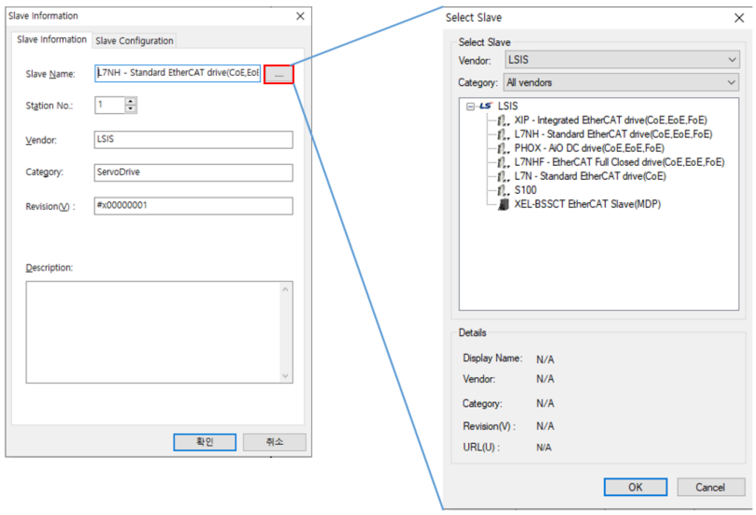
(2) Set Network in the EtherCAT Network Screen

1) Select View-EtherCAT Network from the menu.



2) Double-click in the EtherCAT Network Screen.

3) Confirm a slave name in the slave information window and click the confirm button.



4) When adding other slaves, click the "..." button in the slave information window.

5) Confirm if a slave is correctly added in the network view screen.

(3) Set Cable Duplication in the EtherCAT Network Screen

A single motion controller supports cable duplication that uses junction slaves. To use the cable duplication function, the cable duplication setting is needed for EtherCAT network parameters when supporting EtherCAT Network cable duplication.

1) Click and add the junction slave by clicking in the EtherCAT Network Screen.

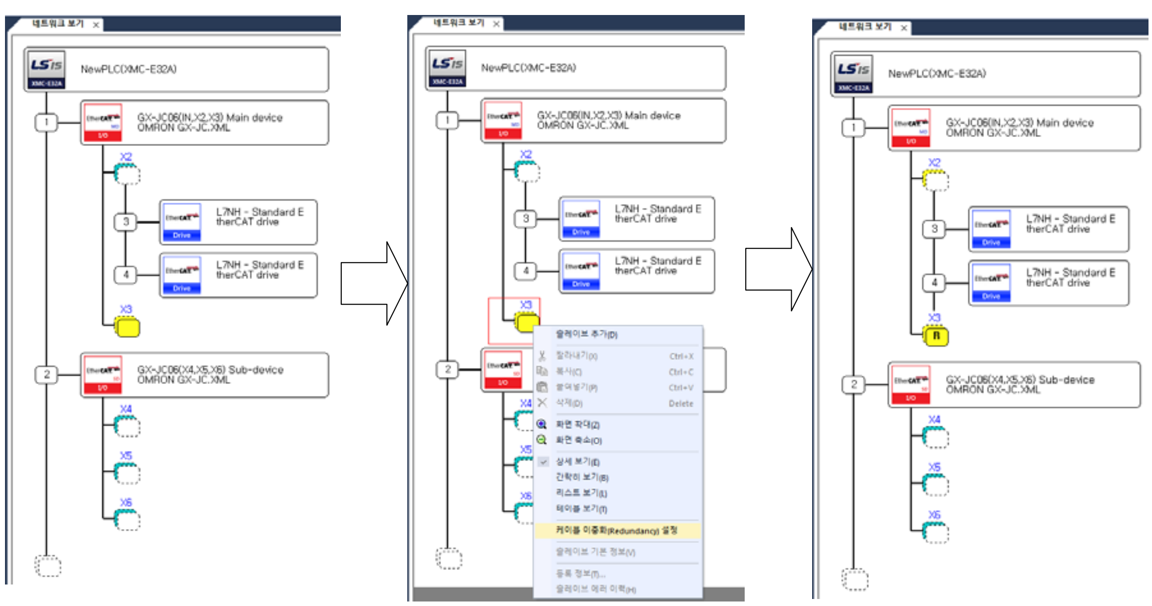
2) Click that represents the port of a junction slave and add slaves to the port 2 (X2) of a junction slave.

3) When adding slaves to the port 2, the display of the port changes into , which signifies the duplication setting is possible.

4) Select the port 3 (X3) or the port 5 (X5) of the junction slave and click the right mouse button.

5) Select the Cable Duplication Setting Menu.

6) Confirm if duplication is set to the port of a junction slave. A duplication port is marked as .



4.4.9 CiA 402 Operation Mode Supported

The CiA402 profile is a profile of drives and motion controllers in the Can open specification. Single motion controller supports the CiA402 profile that supports operation modes such as CSP/CSV/CST/Homing/Velocity. To use operation modes, slaves should support the corresponding operation mode and the corresponding essential PDO should be included in the operation mode that you want to use in the slave PDO setting.

1) Essential PDO by Operation Mode

Velocity mode	
Index	Name
6040h	Control word
6041h	Status word
6042h	vl target velocity
6044h	vl velocity actual value

Homing mode	
Index	Name
6040h	Control word
6041h	Status word
6098h	Homing method

CSV mode	
Index	Name
6040h	Control word
6041h	Status word
606Ch	Velocity actual value
60FFh	Target velocity

CSP mode	
Index	Name
6040h	Control word
6041h	Status word
607Ah	Target position
6064h	Position actual value

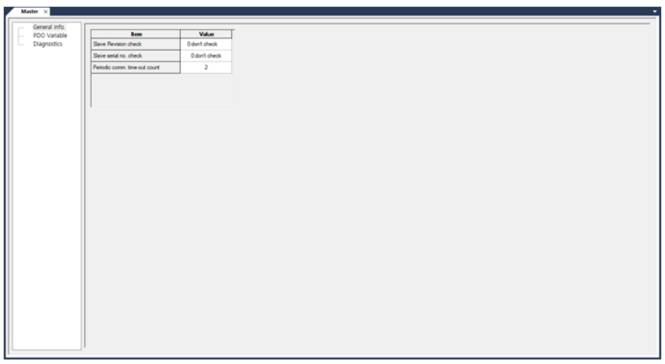
CST mode	
Index	Name
6040h	Control word
6041h	Status word
6071h	Target torque
6077h	Torque actual value

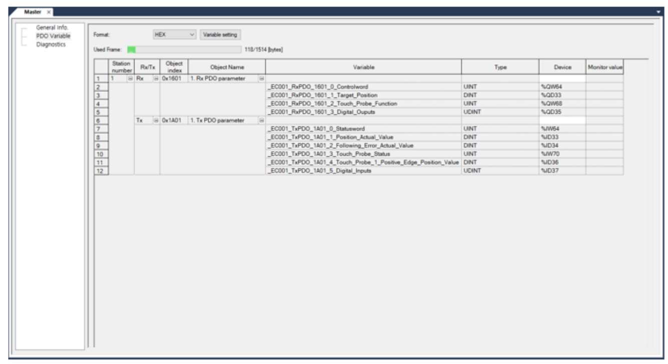
2) Confirm the Supported Mode of Slaves

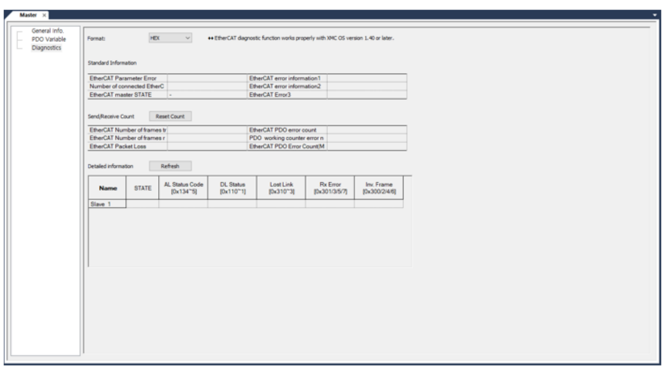
A Single Motion Controller provides the function that users can confirm whether to support the CiA402 profile of the connected slave. For more information see Appendix 8 EtherCAT Diagnosis Function.

4.4.10 EtherCAT Master Setting

The EtherCAT Master Setting consists of General Information/PDO Variable Information/Diagnosis Information. In the General Information Tab you can set Check Revision of Slaves when connecting to EtherCAT/Check Serial Numbers/the Number of the fixed-cycle communication timeout. In the PDO Variable Information Tab you can see all objects mapped to the PDO of the current slave at a glance and monitor a real value while on-line. In the Diagnosis Information Tab you can see status diagnosis flags of slaves at a glance. For more information about the EtherCAT Master Setting see Chapter 5 Memory, Parameter and I/O Signal - (4) EtherCAT Parameter. For more information about EtherCAT Diagnosis Information see Appendix 8 EtherCAT Diagnosis Function.



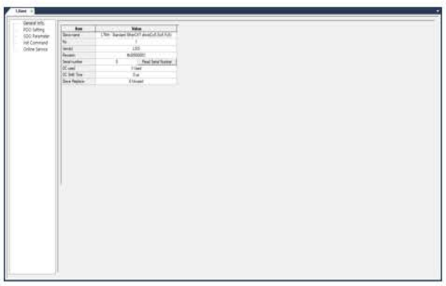
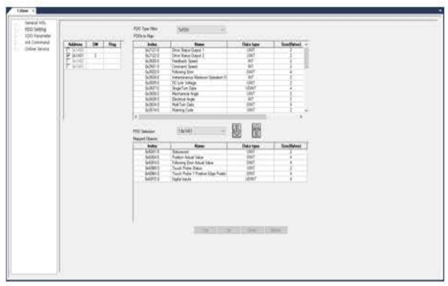
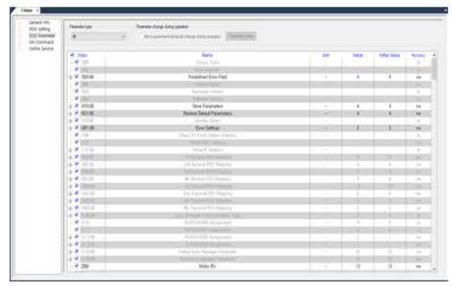
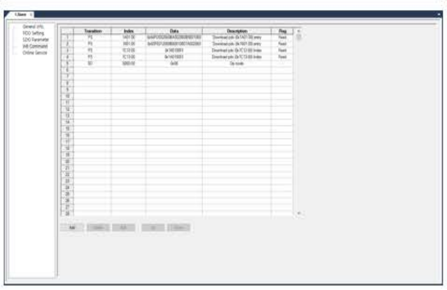
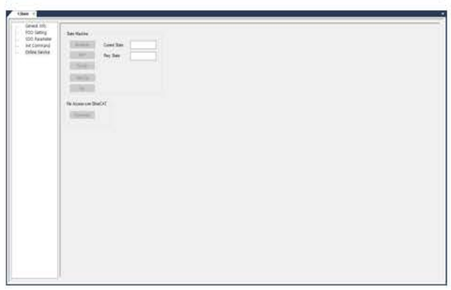




In the EtherCAT Slave Setting you can perform whether to Use the DC of Slaves/PDO Setting/Market Demand Setting. For more information see Chapter 5 Memory, Parameter and I/O Signal - (4) EtherCAT Parameter.

4.4.11 EtherCAT Slave Setting

The EtherCAT Slave Setting consists of General Information/PDO Setting/SDO Parameter/Start Command/On-line Service. In the General Information Tab you can set whether to Use the DC of Slaves/DC Shift Time/Whether to Use the Replace Function during Connection. The PDO Setting Tab provides the function that modifies the PDO mapping information of slaves. The SDO Parameter Tab shows objects registered with the object dictionary of slaves and provides an EEPROM storage function and a modification function during operation. In the Start Command Tab, you can set the initial operation when connecting by specifying the SDO write operation according to the transition of slave status. For more information see Chapter 5 Memory, Parameter and I/O Signal - (4) EtherCAT Parameter.

4.4.12 EtherCAT Error Information Flags

A single motion controller sets error flags when an error occurs during connecting to the EtherCAT. It also provides detailed error information by giving error information flags.

Variable	Type	Memory Allocation	Description
_EC_COMM_ERR	BOOL	%FX65602	EtherCAT Communication timeout error
_EC_ERR_INFO1	STRING	%FB8272	EtherCAT error information 1
_EC_ERR_INFO2	STRING	%FB8304	EtherCAT error information 2

(1) If the network setting is different from the number of slaves while connecting to EtherCAT

Variable	Error message
_EC_ERR_INFO1	NetConfig SlaveQtyxx Actualxx

(2) If the network setting is different from real slaves while connecting to EtherCAT

Variable	Error messages
_EC_ERR_INFO1	Check Slave xx VendorID

(3) If the network setting is different from real slaves while connecting to EtherCAT

Variable	Error messages
_EC_ERR_INFO1	Check Slave xx Product Code

(4) If the network cables are connected differently from the setting while connecting to EtherCAT

Variable	Error messages
_EC_ERR_INFO1	Slave xx DL Statusxx

(5) If the PDO working counter error occurs while connecting to EtherCAT

Variable	Error messages
_EC_ERR_INFO1	PDO Communication Error

(6) If a slave does not respond while connecting to EtherCAT

Variable	Error messages
_EC_ERR_INFO1	ECAT Communication Error

(7) If the slave error occurs while connecting to EtherCAT

Variable	Error messages
_EC_ERR_INFO1	PDO Communication ALStatus Error
_EC_ERR_INFO2	Slave xx AL Status Codexx

4.4.13 EtherCAT Master Status Diagnosis Flag

A single motion controller provides flags to diagnose the EtherCAT status.

Variable	Type	Memory Allocation	Description	Cause
_EC_TRANSMITTED_OK	UDINT	%FD2084	Number of Transmitted Frames	The EtherCAT frame is damaged due to noise
_EC_RECEIVED_OK	UDINT	%FD2085	Number of Received Frames	
_EC_CRCERR_CNT	UDINT	%FD2086	CRC error frame reception	The EtherCAT frame is damaged
_EC_CARRIER_SENSE_ERR	UDINT	%FD2088	EtherCAT Carrier sense error	Another device rather that the EtherCAT device is connected
_EC_COLLISION_CNT	UDINT	%FD2087	The mumber of Collision Frames	The repeater hub is connected
_EC_LINKOFF_CNT	UDINT	%FD2089	Link-off count	The EtherCAT cable is not connected
_EC_OVERSIZE_FRAME	UDINT	%FD2090	Oversize frame reception	The EtherCAT frame is damaged Another device rather that the EtherCAT device is connected
_EC_UNDERSIZE_FRAME	UDINT	%FD2091	Undersize frame reception	The EtherCAT frame is damaged Another device rather that the EtherCAT device is connected
_EC_JABBER_FRAME	UDINT	%FD2092	Jabber frame reception	The EtherCAT frame is damaged Another device rather that the EtherCAT device is connected
_EC_PDO_ERR_CNT_TOTAL	UDINT	%FD2104	PDO Error Count (Accumulated)	An error has occurred during PDO communication
_EC_PDO_ERR_CNT_MAX	UDINT	%FD2106	PDO Error Count (Max.)	An error has occurred during PDO communication
_EC_LOST_FRAME	UDINT	%FD2105	Number of Damaged Frames	The EtherCAT frame is damaged due to noise

4.4.14 EtherCAT Slave Status Diagnosis Flag

A single motion controller provides flags to diagnose the EtherCAT status of slaves. When errors occurred during EtherCAT communication, the diagnosis information of slaves can detect the slave with problems. For more information see Appendix 8 EtherCAT Diagnosis Function.

Variable	Description
_SLVxx_ALStatus	Shows the AL status of slave applications.
_SLVxx_ALStatusCode	Shows the error code of slave applications.
_SLVxx_DLStatus	Shows the link status information of slaves.
_SLVxx_LinkLostCount	Shows the link stop event counter for each port of slaves.
_SLVxx_InValidFrameCounterA/B/C/D	The count increases if there are errors in frame formats such as Preamble, SFD and CRC. The whole bit sequence corresponds to the damaged frame. Errors can occur in frames.
_SLVxx_RxErrorCounterA/B/C/D	The count increases if individual symbols are not valid. Errors can occur both in and out of frames.
_SLVxx_ForwardedRXErrCounter	Abnormal frames detected through the previous slaves show the received count.

4.4.15 Using the Third Party EtherCAT Slave

To use the third party EtherCAT slave other than the slave basically provided by a single motion controller, the EtherCAT Slave Information (ESI) is needed first. A manufacturer of EtherCAT slaves provides the ESI file that is defined as the EtherCAT standard. Users can directly register the ESI file given by a manufacturer with XG5000. To register the ESI file there are two ways: one way that XG5000 automatically identify the ESI file and the other way that user directly registers it.

(1) Automatic Identification of the ESI File

1) End XG5000 if executing it.

2) Put the ESI file in the EtherCATXML folder of the XG5000 installation folder.

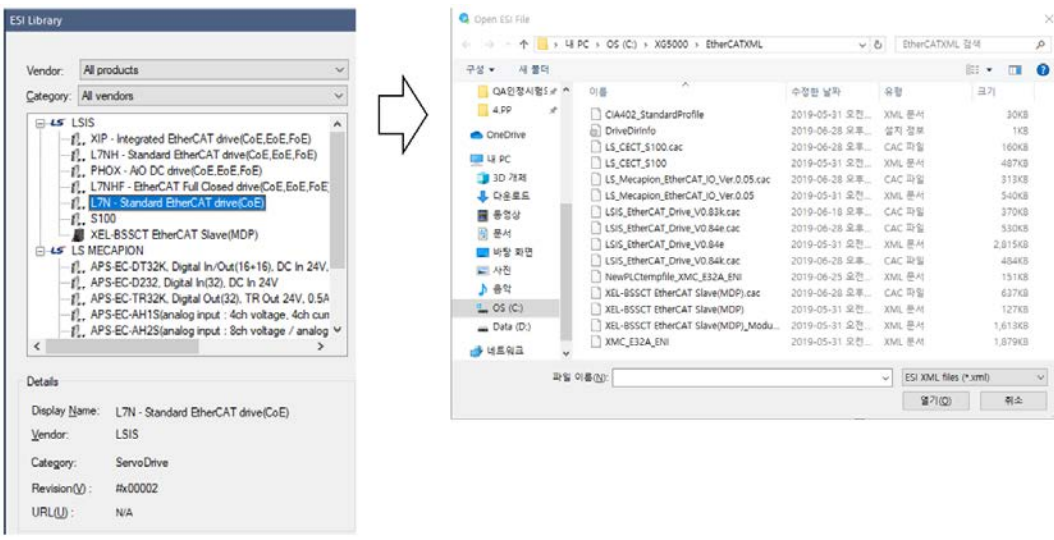
3) Execute XG5000.

(2) Direct Registration of the ESI File

1) Click the right mouse button in the ESI library window of XG5000.

2) Select 'Add the ESI File' from the shortcut menu.

3) Select the ESI file in the dialog box, 'Open the ESI File' and select the button, 'Open'.



4.5 Motion Control Program

4.5.1 Program Execution

(1) program Program Configuration

A motion control program consists of all functional elements necessary to execute a specific control. The program runs on the built-in RAM, and the program is backed up in eMMC.

Programs according to these functional elements are classified as follows.

Program	Contents of process
Main task program	Process the command which is executed in every 'main task period'.
Periodic task program	Process the command which is executed in every 'periodic task period'.
Initializing task program	Execute the command which is executed once in case of motion controller RUN.

Caution

Since the motion control program is stored in eMMC when the power is off, the number of program writes is limited to 100,000 times. Please be careful when using the program.

4.5.2 Operation mode

(1) RUN mode

This is a mode which normally performs the motion program calculation.

1) Processing two axes a mode is changed

Initialization is performed in the data area at the beginning, and possibility of performance is decided by examining the validity of the motion program.

2) The contents of operation processing

Motion program, motion command calculation, input/output data processing, and EtherCAT communication are performed.

(2) STOP mode

This is a mode in stop state which does not perform the motion program calculation.

1) Processing when changing the mode

Every output data is in Off state.

2) The contents of operation processing

This performs EtherCAT communication.

3) You can execute the command executed in the command window without motion program operation.

The command executed in the command window is performed, and EtherCAT communication is executed.

(4) Change in operation modes

Operation mode of motion controller can be changed as follows.

Operation mode	Note
during RUN	Motion controller performs program.
STOP → RUN	Motion controller changes from STOP mode to RUN mode.
RUN → STOP	Motion controller changes from RUN mode to STOP mode.
During STOP	TEST command can be performed only in case motion controller is in STOP mode in XG5000.

Chapter 5 Memory and parameter, I/O Signal

5.1 Memory

5.1.1 Program and data memory

(1) Program Memory

The configuration of the memory related to the program embedded in the motion controller is as follows.

Item	Size(KB)	Content
Parameter	9,251KB	User parameter area
Motion program	10,240KB	User program related to motion
NC Program	10,240KB	User program related to NC
Program operation table	4,478KB	Table area related to the program
System operation	47,104KB	System OS area

(2) Data Type Memory

The details and size of the data memory embedded in the motion controller are as follows.

Item		Size(KB)	Content
User Device	Automatic variable (A)	4,096KB	Automatic variable area
	Direct variable (M)	2,048KB	Internal device area
	Input variable(I)	16KB	Built-in digital input, TxPDO data of EtherCAT slaves
	Output variable(Q)	16KB	Built-in digital output, RxPDO data of the EtherCAT slave
	System variable(F)	128KB	Variables related to motion control status and module operation status
	Special variable(U)	1KB	Built-in analog operations and state variables
	Special variable(K)	18KB	SD memory, data log and encoder flag area
	Etc	456KB	UDF/B Internal purposes and NC Local variables
History		88KB	User history (error / mode / system / power / motion error)
System area		43,008KB	Internal operating area of the system and other functions

5.1.2 Device

(1) Device type

Types of device supported in motion control module are shown in the Table below.

Type	Capacity	Usage
Automatic variable (A)	4,096KB	Assigned when adding a symbolic variable to the automatic variable area (able to set 2,408KB of retain)
Direct variable (M)	2,048KB	Internal device area (able to set 1,024KB of retain by selecting in the area of basic parameter)
Input variable(I)	16KB	Built-in digital input, TxPDO data of the EtherCAT slave
Output variable(Q)	16KB	Built-in digital output, RxPDO data of the EtherCAT slave
System variable(F)	128KB	Variables related to motion control status and module operation status
Special variable(U)	1KB	Built-in analog operations and state variables
Special variable(K)	18KB	SD memory, data log and encoder flag area

(a) Automatic variable

- This is a variable to be automatically assigned the position of variables by compiler; user does not need to specify the position of internal variable. The variables, which user sets but does not assign specific position, are assigned to automatic variable.
- The automatic variables that do not have Retain Settings are initialized to 0 when power is applied again or at Stop-to-Run.

(b) Direct variable

- (a) This is a variable which user forces the position of memory to be assigned by using the name and number of a device directly.
- (b) The range of address assignment where direct variable is available is as follows.

Size of Variable	Designated range of Variable address
X(bit)	%MX0 ~ %MX16777215
B(byte)	%MB0 ~ %MB2097151
W(word)	%MW0 ~ %MW1048575
D(double word)	%MD0 ~ %MD524287
L(long word)	%ML0 ~ %ML262143

(c) Input variable (I)

1) This is a variable assigned to built-in digital input and TxPDO of EtherCAT slaves.

2) Built-in digital input is 8 points.

3) Input variable is expressed as follows.

%I[size prefix]n

No.	Description
Size prefix	X(1 bit), B(1 byte), W(1 word), D(1 double word), L(1 long word)
n	n data based on [size prefix] among data

%I[size prefix]n1.n2.n3

No.	Description
Size prefix	X(1 bit), B(1 byte), W(1 word), D(1 double word), L(1 long word)
n1	0~127 block assigned
n2	0~15 block assigned
n3	64 bit assignment. n3 data based on [size prefix]

Example) %IW64 = %IB128 = %IW1.0.0 = %IB1.0.0.

Also %IW1 = %IB2 = %IW0.0.1 = %IB0.0.2.

4) Device depending on the input variable expression is assigned as follows.

Device	Description
%IX0	Built-in digital input 0
%IX1	Built-in digital input 1
%IX2	Built-in digital input 2
%IX3	Built-in digital input 3
%IX4	Built-in digital input 4
%IX5	Built-in digital input 5
%IX6	Built-in digital input 6
%IX7	Built-in digital input 7
%IW64 ~	TxPDO mapping data of EtherCAT slaves

(d) Output variable (Q)

1) This is a variable assigned to built-in digital output and RxPDO of EtherCAT slaves.
2) Built-in digital output is 16 points.
3) Input variable is expressed as follows.

%Q [size prefix]n

No.	Description
Size prefix	X(1 bit), B(1 byte), W(1 word), D(1 double word), L(1 long word)
N	n data based on [size prefix] among data

%Q [size prefix] n1.n2.n3

No.	Description
Size prefix	X(1 bit), B(1 byte), W(1 word), D(1 double word), L(1 long word)
n1	0~127 block assigned
n2	0~15 block assigned
n3	64 bit assignment. n3 data based on [size prefix]

Example) %QW64 = %QB128 = %QW1.0.0 = %QB1.0.0.

Also %QW1 = %QB2 = %QW0.0.1 = %QB0.0.2.

4) Device depending on the output variable expression is assigned as follows.

Device	Description
%QX0	Built-in digital output 0
%QX1	Built-in digital output 1
%QX2	Built-in digital output 2
%QX3	Built-in digital output 3
%QX4	Built-in digital output 4
%QX5	Built-in digital output 5
%QX6	Built-in digital output 6
%QX7	Built-in digital output 7
%QX8	Built-in digital output 8
%QX9	Built-in digital output 9
%QX10	Built-in digital output 10
%QX11	Built-in digital output 11
%QX12	Built-in digital output 12
%QX13	Built-in digital output 13
%QX14	Built-in digital output 14
%QX15	Built-in digital output 15
%QW64 ~	RxPDO mapping data of EtherCAT slaves

(e) Special variable

1) This is a variable assigned to built-in analog input and output.
2) Built-in analog input is 2 channels, and built-in analog output is 2 channels.
3) Built-in analog variable is expressed as follows.

%I [size prefix]n1.n2.n3

No.	Description
Size prefix	X(1 bit), B(1 byte), W(1 word), D(1 double word), L(1 long word)
n1	0: Motion Controller
n2	1: Built-in Analog
n3	n3 data based on [size prefix] among n2 data

4) Device depending on the analog variable expression is assigned as follows.

Built-in analog input

Variable	Type	Device	Description
_01_AD0_ACT	BOOL	%UX0.1.16	CH0 RUN
_01_AD0_AVGTYPE	Byte	%UB0.1.34	Channel 0 Average processing
_01_AD0_AVGVAL	WORD	%UW0.1.18	Channel 0 Average value
_01_AD0_DATA	WORD	%UW0.1.5	CH0 conversion value
_01_AD0_DATATYPE	Byte	%UB0.1.26	Channel 0 data type setting
_01_AD0_ERR	BOOL	%UX0.1.32	Channel 0 error
_01_AD0_FILTCONST	WORD	%UW0.1.15	Channel 0 Filter constant
_01_AD0_HOLDVAL	BOOL	%UX0.1.320	Channel 0 valid conversion value hold setting.
_01_AD0_HOOR	BOOL	%UX0.1.48	Channel 0 upper limit alarm
_01_AD0_IDD	BOOL	%UX0.1.72	Channel 0 input disconnection detection
_01_AD0_LOOR	BOOL	%UX0.1.56	Channel 0 lower limit alarm
_01_AD0_RANGE	Byte	%UB0.1.22	Channel 0 range setting.
_01_AD0_RUN	BOOL	%UX0.1.160	Channel 0 run setting
_01_AD1_ACT	BOOL	%UX0.1.17	Channel 1 Active
_01_AD1_AVGTYPE	Byte	%UB0.1.35	Channel 1 Average processing
_01_AD1_AVGVAL	WORD	%UW0.1.19	Channel 1 Average value
_01_AD1_DATA	WORD	%UW0.1.6	Channel 1 Output data
_01_AD1_DATATYPE	Byte	%UB0.1.27	Channel 1 data type setting
_01_AD1_ERR	BOOL	%UX0.1.33	Channel 1 error
_01_AD1_FILTCONST	WORD	%UW0.1.16	Channel 1 Filter constant
_01_AD1_HOLDVAL	BOOL	%UX0.1.321	Channel 1 valid conversion value hold setting.
_01_AD1_HOOR	BOOL	%UX0.1.49	Channel 1 upper limit alarm
_01_AD1_IDD	BOOL	%UX0.1.73	Channel 1 input disconnection detection
_01_AD1_LOOR	BOOL	%UX0.1.57	Channel 1 lower limit alarm
_01_AD1_RANGE	Byte	%UB0.1.23	Channel 1 Range setting
_01_AD1_RUN	BOOL	%UX0.1.161	Channel 1 Operation setting
_01_AD_ACT_ARY	ARRAY[0..1] OF BOOL	%UX0.1.16	Active stats per channel (Array)
_01_AD_AVGTYPE_ARY	ARRAY[0..1] OF BYTE	%UB0.1.32	Average process by channel
_01_AD_AVGVAL_ARY	ARRAY[0..1] OF WORD	%UW0.1.18	Average value by channel

Variable	Type	Device	Description
_01_AD_DATATYPE_ARY	ARRAY[0..1] OF BYTE	%UB0.1.26	Data type setting by channel
_01_AD_DATA_ARY	ARRAY[0..1] OF WORD	%UW0.1.5	Conversion values by channel
_01_AD_ERR_ARY	ARRAY[0..1] OF BOOL	%UX0.1.32	Error by channel
_01_AD_FILTCONST_ARY	ARRAY[0..1] OF WORD	%UW0.1.15	Filter constant by channel
_01_AD_HOLDVAL_ARY	ARRAY[0..1] OF BOOL	%UX0.1.320	Valid conversion value hold setting by channel
_01_AD_HOOR_ARY	ARRAY[0..1] OF BOOL	%UX0.1.48	Alarm (Upper Limit) by channel
_01_AD_IDD_ARY	ARRAY[0..1] OF BOOL	%UX0.1.72	Input disconnection detection by channel
_01_AD_LOOR_ARY	ARRAY[0..1] OF BOOL	%UX0.1.56	Alarm (Lower Limit) by channel
_01_AD_RANGE_ARY	ARRAY[0..1] OF BYTE	%UB0.1.22	Range setting by channel
_01_AD_RUN_ARY	ARRAY[0..1] OF BOOL	%UX0.1.160	Operation setting by channel

Built-in analog output

Variable	Type	Device	Description
_01_DA0_ACT	BOOL	%UX0.1.24	Channel 0(Voltage) Active
_01_DA0_DATA	WORD	%UW0.1.8	Channel 0(Voltage) Input data
_01_DA0_DATATYPE	Byte	%UB0.1.28	Channel 0 data type setting
_01_DA0_ERR	BOOL	%UX0.1.40	Channel 0(Voltage) Error
_01_DA0_INTP	BOOL	%UX0.1.64	Channel 0(Voltage) Interpolation enabled
_01_DA0_INTPMTHD	Byte	%UB0.1.46	Channel 0(Voltage) Interpolation method
_01_DA0_INTPTIME	Byte	%UB0.1.48	Channel 0(Voltage) Interpolation time setting
_01_DA0_INTPVAL	WORD	%UW0.1.25	Channel 0(Voltage) Interpolation value
_01_DA0_OUTEN	BOOL	%UX0.1.112	Channel 0(Voltage) Output enable
_01_DA0_OUTSTAT	WORD	%UW0.1.21	Channel 0 output Status setting
_01_DA0_RANGE	Byte	%UB0.1.24	Channel 0 Range setting
_01_DA0_RUN	BOOL	%UX0.1.168	Channel 0 Operation setting
_01_DA1_ACT	BOOL	%UX0.1.25	Channel 1(Voltage) Active
_01_DA1_DATA	WORD	%UW0.1.9	Channel 1(Voltage) Input data
_01_DA1_DATATYPE	Byte	%UB0.1.29	Channel 1 data type setting
_01_DA1_ERR	BOOL	%UX0.1.41	Channel 1(Voltage) Error
_01_DA1_INTP	BOOL	%UX0.1.65	Channel 1(Voltage) Interpolation enabled
_01_DA1_INTPMTHD	Byte	%UB0.1.47	Channel 1(Voltage) Interpolation method
_01_DA1_INTPTIME	Byte	%UB0.1.49	Channel 1(Voltage) Interpolation time setting

_01_DA1_INTPVAL	WORD	%UW0.1.26	Channel 1(Voltage) Interpolation value
_01_DA1_OUTEN	BOOL	%UX0.1.113	Channel 1(Voltage) Output enable
_01_DA1_OUTSTAT	WORD	%UW0.1.22	Channel 1 output Status setting
_01_DA1_RANGE	Byte	%UB0.1.25	Channel 1 range setting.
_01_DA1_RUN	BOOL	%UX0.1.169	Channel 1 run setting
_01_DA_ACT_ARY	ARRAY[0..1] OF BOOL	%UX0.1.24	Active stats by channel (Array)
_01_DA_DATATYPE_ARY	ARRAY[0..1] OF BYTE	%UB0.1.28	Data type setting by channel

Variable	Type	Device	Description
_01_DA_DATA_ARY	ARRAY[0..1] OF WORD	%UW0.1.8	Input values by voltage channel
_01_DA_ERR_ARY	ARRAY[0..1] OF BOOL	%UX0.1.40	Error by channel
_01_DA_INTPMTHD_ARY	ARRAY[0..1] OF BYTE	%UB0.1.46	Interpolation method setting by channel
_01_DA_INTPTIME_ARY	ARRAY[0..1] OF BYTE	%UB0.1.48	Interpolation time setting by channel
_01_DA_INTPVAL_ARY	ARRAY[0..1] OF WORD	%UW0.1.25	Interpolation value by channel
_01_DA_INTP_ARY	ARRAY[0..1] OF BOOL	%UX0.1.64	Interpolation enabled by channel
_01_DA_OUTEN_ARY	ARRAY[0..1] OF BOOL	%UX0.1.112	Output enable by channel
_01_DA_OUTSTAT_ARY	ARRAY[0..1] OF WORD	%UW0.1.21	Output Status setting by channel
_01_DA_RANGE_ARY	ARRAY[0..1] OF BYTE	%UB0.1.24	Range setting by channel
_01_DA_RUN_ARY	ARRAY[0..1] OF BOOL	%UX0.1.168	Operation setting by channel

Built-in analog common

Variable	Type	Device	Description
_01_ERR	BOOL	%UX0.1.0	Module error
_01_RDY	BOOL	%UX0.1.15	Module ready
_01_SETTINGERR	WORD	%UW0.1.27	Setting error information

(f) Special variable (K)

1) These variables are assigned to the SD memory, data log and embedded encoder flag area.
2) The built-in encoder input is 2 channels.
3) For the memory allocated to the Special Variable, please refer to the Appendix 1 Flag List 7) SD memory Flag ~ 9) Encoder Flag.

(g) System variable(F)

1) These variables are assigned to the status variable of motion control status and system statues.
2) For details on the kinds of flags, please refer to the Appendix 1 Flag List 1) System Flag.

(2) Retain setting

Default (automatic) variable retain is used when wanting to keep and use the data that occurs while operating or the data required for an operation even in the case of restarting after the motion controller has stopped.

Certain part of the device in M area can be used as retain area by setting the basic parameter. The following table summarizes the features of retain settable device.

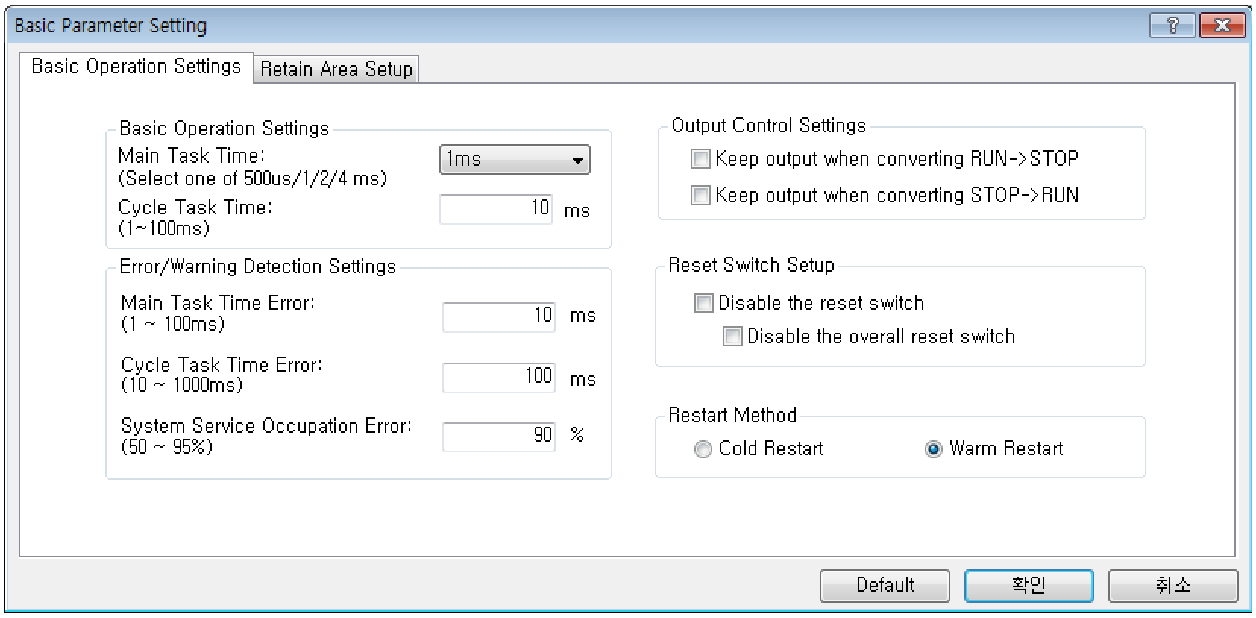
Device	Retain setting	Characteristics
Default	O	Enable retain setting when adding variable to automatical variable area
M	O	Enable retain setting in parameter by internal contact point area
I	X	Built-in digital input, TxPDO data of EtherCAT slaves
Q	X	Built-in digital output, RxPDO data of the EtherCAT slave

5.1.3 Parameter

(1) Basic parameter

Explain Basic parameter of the motion control module. For more details on operation, refer to Chapter 10 CPU Functions.

(a) Basic operation setting



1) Main Task time

- Set the motion period of the main task. The period can be set by selecting one in 0.5ms/1ms/2ms/4ms.
- Set the control time of performing in the main task of motion controller considering the execution time of program.
- When the execution time of the main task exceeds the main task period, an error occurs and if motion control module is in RUN state, it is changed to STOP state, the operation of the motion controller is stopped immediately, and an error is generated.

2) Periodic task time

Set the motion period of the periodic task. The period can be set in multiples of the main task between 1 ~ 100ms.

The periodic task is performed in the remaining time after performing the main task in the control period, and therefore, it can be performed through a number of control periods.

3) Main task cycle errors

- It sets the run time of the main task that causes errors when the task runs beyond the set time. The setting range is 1~100ms.

4) Periodic task cycle errors

- It sets the run time of the periodic task that causes errors when the task runs beyond the set time. The setting range is 10~1000ms.

5) Task program occupancy rate warning

- If the task program occupancy rate exceeds the set value because there are many main task programs or periodic task programs, the task program occupancy rate warning occurs. It can be set in the range of 50~95%. If the task program occupancy rate exceeds 100%, the task program occupancy rate error occurs, and it changes to the ERROR state.

6) Output Control setting

- When an error occurs in module or changing the motion mode, decide whether to maintain the data output or not.

Select	Operation
Maintain the output when switching from RUN to STOP	Decide whether to output the data normally during the operation mode of motion controller is switching from RUN to STOP.
Maintain the output when switching from STOP to RUN	Decide whether to output the data normally during the operation mode of motion controller is switching from STOP to RUN.

7) Setting whether or not to turn off the Reset switch

- Sets whether or not to perform reset operation with the switch on the front panel of the product. The item can be set to 'Allowed' or 'Prohibited'.

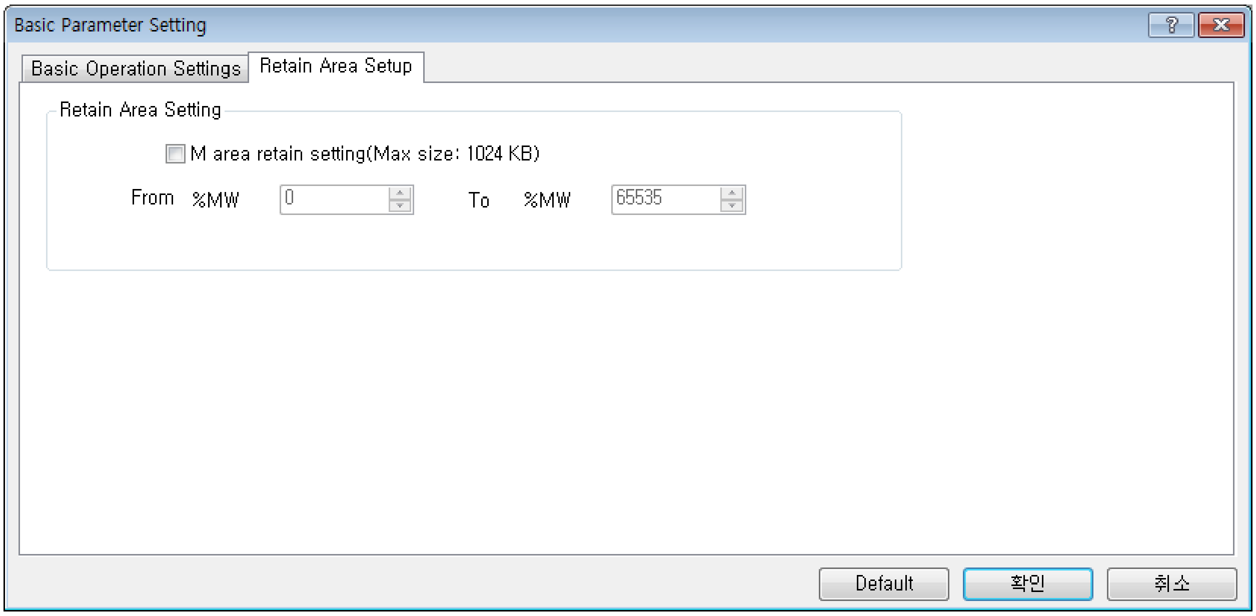
8) Restart mode

Restart by motion controller reset or turning on the power after turning off is divided into cold restart and warm restart. With regard to restart mode, variables can be set in 3 different types such as default, initialization, and retain; and the initialization of variables set by restart mode is as follows.

Variable	Cold restart	Warn Restart
Default	Initialize to '0'	Initialize to '0'
Retain	Initialize to '0'	Retain previous value
Initialization	Initialize to user defined value	Initialize to user defined value
Retain & initialization	Initialize to user defined value	Retain previous value

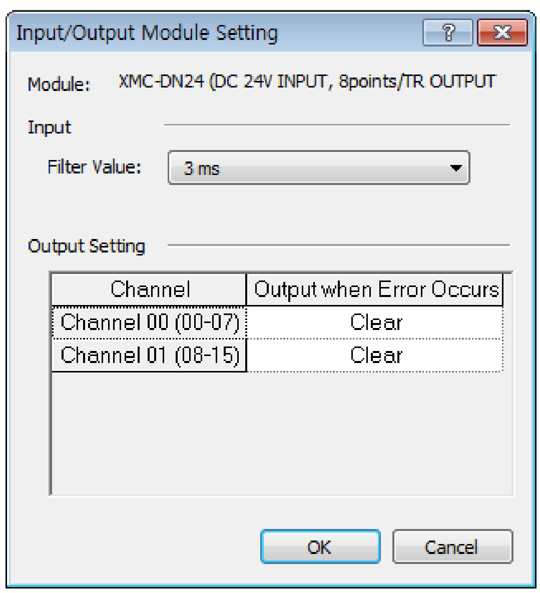
(b) Memory area setting

This is a parameter item which sets the retain area. Retain area can be set by checking the "M Area retain set" to activate retain setting. Retain can be set up to 1,024Kbyte, and if the beginning and ending addresses are set to be retain in M area, the value of relevant area is maintained even when turning off the power.



(2) I/O parameter

(a) Built-in input/output setting

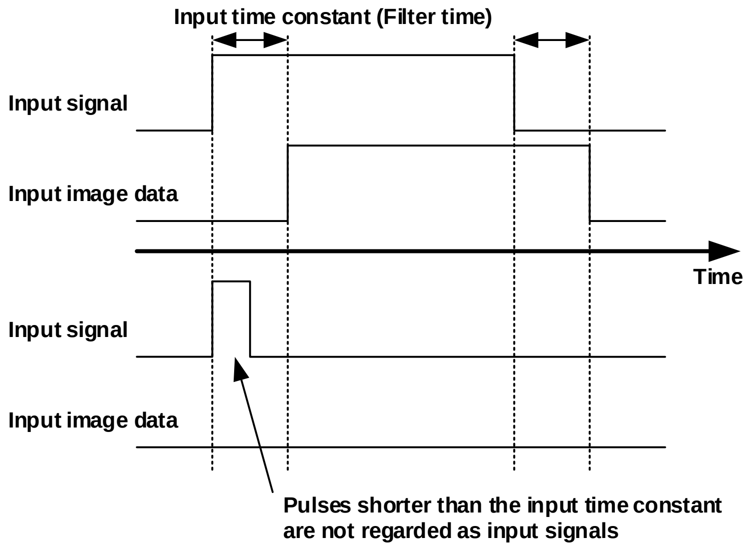


1) Input filter function

The built-in input part of the motion controller has an input filter function to prevent the external noise signal flowing into the input signal. In environments where there is a lot of noise or in the case of the equipment where the pulse width of the input signal acts as an important factor, the system may be subjected to incorrect input depending on the state of the input signal. In order to prevent such mistaken input, the input filter function does not accept the signal that is shorter than the time set by the user as input.

The input filter time can be set 1ms~100ms.

The following shows the timing diagram of the input filter function.



2) Setting output in case of error

The built-in output part of the motion controller provides the emergency output function to determine whether the output state is maintained or cleared when the operation is stopped due to errors.

When an error occurs, the output setting can be set in units of 8 points.

When the emergency output is set to 'No' (Clear), the output is turned off when the operation is stopped due to errors of the motion controller; and the output status is maintained by selecting 'Hold'.

(b) Built-in analog setting

For more details on the built-in analog, refer to Chapter 13 Built-in Analog Function.

(3) Built in parameter

(a) Data Log

For more details on the data log, refer to Chapter 11 Data Log Function.

(b) Encoder

Encoder parameter is explained as follows.

Item	Content	Setting range	Initial values
Encoder 1 unit	Set display unit of encoder position.	0: pulse 1: mm 2: inch 3:degree	0: pulse
Encoder1 Pulses per rotation	Set Encoder1 pulses per rotation	1 ~ 4294967295	8192 pls
Encoder1 Travel per rotation	Set the movement amount of the load side moved per encoder 1 rotation.	0.000000001 ~ 4294967295	10 pls
Encoder1 Pulse input	Set the input mode in accordance with the output shape of encoder.	0:CW/CCW 1 phase 1 multiplication 1: PULSE/DIR 1 multiplication 2: PULSE/DIR 2 multiplication 3: PHASE A/B 1 multiplication 4: PHASE A/B 2 multiplication 5: PHASE A/B 4 multiplication	3: PHASE A/B 1 multiplication
Encorder1 max. value	Set position display range of encoder.	Long real(LREAL)	2147483647 pls
Encoder1 Min. value			-2147483648 pls
Encoder1 Speed unit	Set the encoder speed display unit.	0: Unit/sec 1: Unit/min 2: rpm	0: Unit/sec
Encoder1 input filter value	Limit the frequency of pulse input to encoder.	0: Unused 1: 500kPPS 2: 200kPPS 3. 100kPPS 4: 10kPPS 5: 1kPPS 6: 0.2kPPS	0: Unused
Encoder1 position filter time constant	Set the time constant (in hours) of the filter to calculate the encoder's position average.	0 ~ 1000	0 ms
Encoder 1 Position Latch	Set whether to use the Encoder 1 Position Latch function using the input contact (%IX0.0.0).	0: Unused 1: used	0: Unused

Item	Content	Setting range	Initial values
Encoder 2 unit	Set display unit of encoder position.	0: pulse 1: mm 2: inch 3:degree	0: pulse
Encoder1 Pulses per rotation	Set Encoder1 pulses per rotation	1 ~ 4294967295	8192 pls
Encoder1 Travel per rotation	Set the movement amount of the load side moved per encoder 1 rotation.	0.000000001 4294967295	10 pls
Encoder2 Pulse input	Set the input mode in accordance with the output shape of encoder.	0:CW/CCW 1 phase 1 multiplication 1: PULSE/DIR 1 multiplication 2: PULSE/DIR 2 multiplication 3: PHASE A/B 1 multiplication 4: PHASE A/B 2 multiplication 5: PHASE A/B 4 multiplication	3: PHASE A/B 1 multiplication
Encorder2 max. value	Set position display range of encoder.	Long real(LREAL)	2147483647 pls
Encoder2Min. value			-2147483648 pls
Encoder2Speed unit	Set the encoder speed display unit.	0: Unit/sec 1: Unit/min 2: rpm	0: Unit/sec
Encoder2input filter value	Limit the frequency of pulse input to encoder.	0: Unused 1: 500kPPS 2: 200kPPS 3. 100kPPS 4: 10kPPS 5: 1kPPS 6: 0.2kPPS	0: Unused
Encoder2position filter time constant	Set the time constant (in hours) of the filter to calculate the encoder's position average.	0 ~ 1000	0 ms
Encoder 2 Position Latch	Set whether to use the Encoder 2 Position Latch function using the input contact (%IX0.0.1).	0: Unused 1: used	0: Unused

(a) Encoder unit

This is to set the display unit of encoder position, and each control target can be set by pulse, mm, inch, and degree. In case of the synchronous operation having the encoder as a center, the unit must be set by the same unit with it of the synchronous operation axis.

Notes

When the encoder unit is different from the synchronous operation axis, it operates by the synchronous ratio regardless of the unit.

[Example]
- Encoder unit: pulse
- Encoder resolution: 4096 pulse
- Unit of Synchronous operation axis: mm
- Master axis : Slave axis = 2 : 1

Encoder1 travel of synchronous operation axis per rotation = 4,096 x1 /2 = 2,048 [mm]

(b) Encoder Pulses per rotation

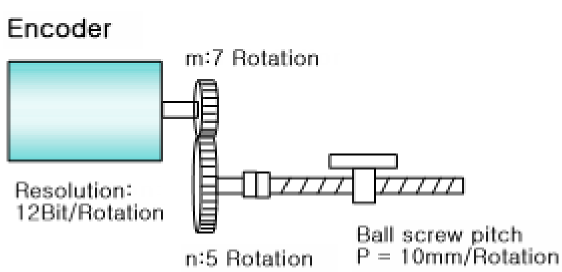
When using mm, inch, and degree for the encoder unit, set the number of purses per encoder rotation.

(c) Encoder Travel per rotation

When using mm, inch, and degree for the encoder unit, set the amount of movement of the load side moved per encoder rotation.

[Example]

When the machine which is moved by ball screw is connected to the encoder with gear, the setting of the encoder unit / Encoder Pulses per rotation / Encoder Travel per rotation is as follows.



- Encoder unit: mm
- Encoder Pulses per rotation = Encoder resolution x Encoder side gear ratio
  = 4096 x 7
  = 2,8672 pls
- Encoder Travel per rotation = Ball screw pitch x Machine side gear ratio
  = 10.0 mm x 5
  = 50.0 mm

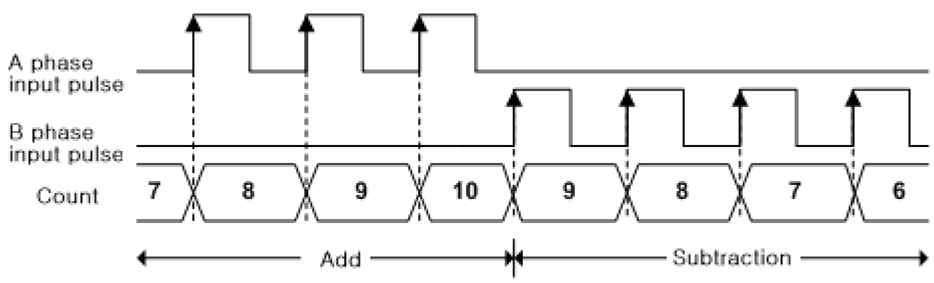
(d) Encoder Pulse input

If you want to use by signal of a manual pulse generator or encoder, can select suitable signal of a manual pulse generator or encoder for using. One among CW/CCW (x1), PULSE/DIR (x1), PULSE/DIR (x2), PHASE A/B (x1), PHASE A/B (x2), and PHASE A/B (x4) must be selected and set for the encoder input signal.

1) CW/CCW 1 phase 1 multiplication

Counts at the rising edge of A-phase input or B phase input. Increase count value if B-phase input is Low state at the rising edge of A-phase input and decrease count value if A-phase input is Low state at the rising edge of B-phase input.

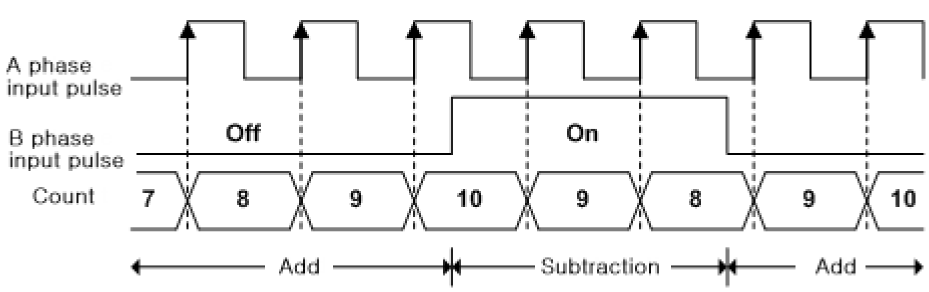
Increasing/Decreasing classification	A-phase input pulse high	A-phase input pulse low
B-phase input High	-	decreasing count
B-phase input pulse Low	Increasing count	-



2) PULSE/DIR 1 multiplication

Count operation is performed when a phase input pulse increases, whether to be added or subtracted is decided by B phase.

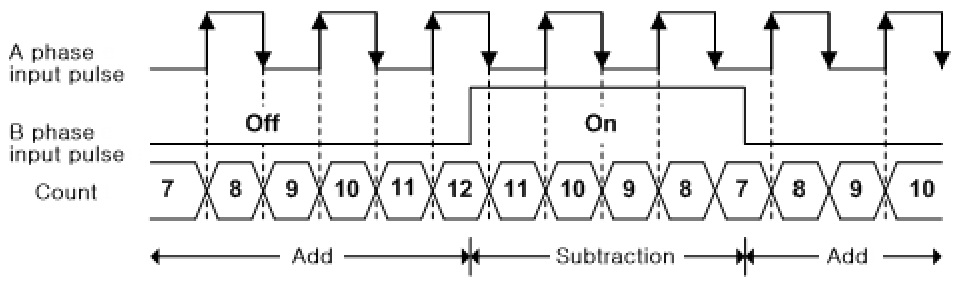
Increasing/Decreasing classification	A-phase input pulse rising	A-phase input pulse falling
B-phase input pulse off	Increasing count	-
B-phase input pulse on	decreasing count	-



3) PULSE/DIR 2 multiplication

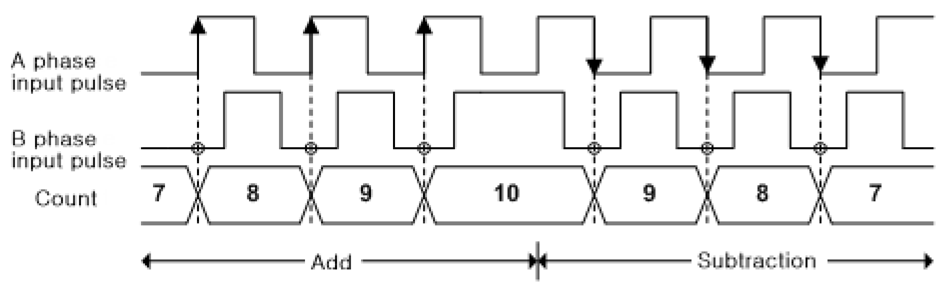
Count operation is performed when a phase input pulse increases and decreases, and whether to be added or subtracted is decided by B phase.

Increasing/Decreasing classification	A-phase input pulse rising	A-phase input pulse falling
B-phase input pulse off	Increasing count	Increasing count
B-phase input pulse on	decreasing count	decreasing count



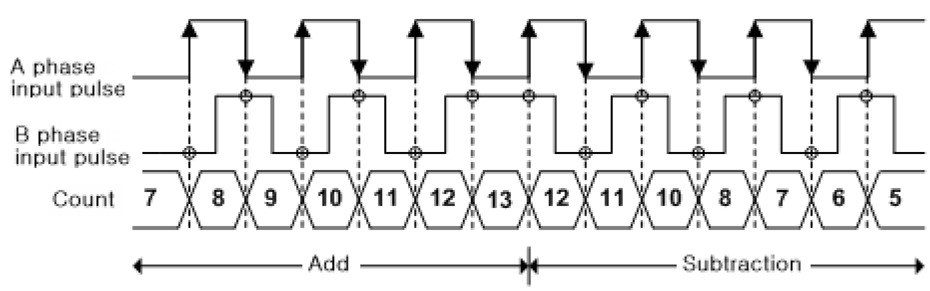
4) PHASE A/B 1 multiplication

Add operation is performed in case of the increase in A phase pulse when the phase of A phase input pulse is ahead of B phase input pulse, and subtraction operation is performed in case of the decrease in A phase pulse when the phase of B phase input pulse is ahead.



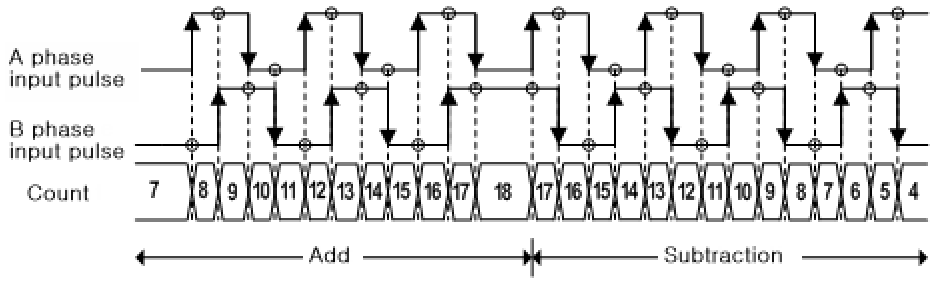
5) PHASE A/B 2 multiplication

Count operation is performed when both increase and decrease in a phase input pulse. Add operation is performed when the phase of A phase is input ahead of B phase, and subtraction operation is performed when the phase of B phase is input ahead of A phase.



6) PHASE A/B 4 multiplication

A-phase input pulse and B-phase input pulse count at rising. If A-phase input is antecedent to B-phase input, increasing operation starts, and if B-phase input is antecedent to A-phase input, decreasing operation starts

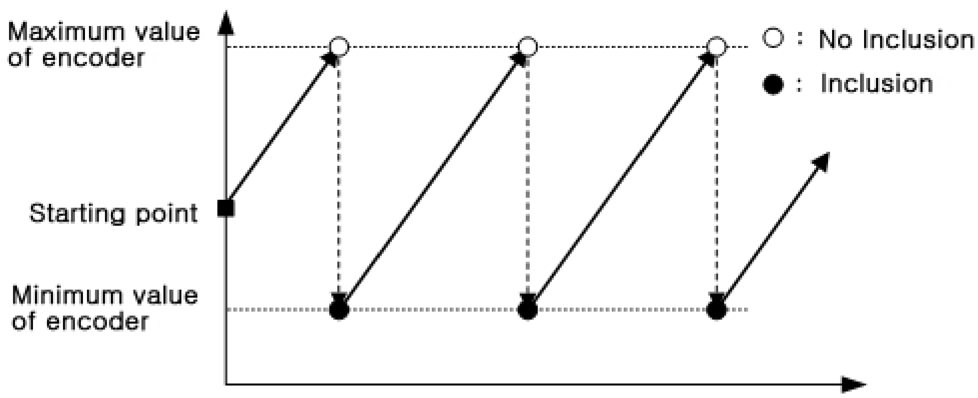


(e) Encoder maximum value and minimum value

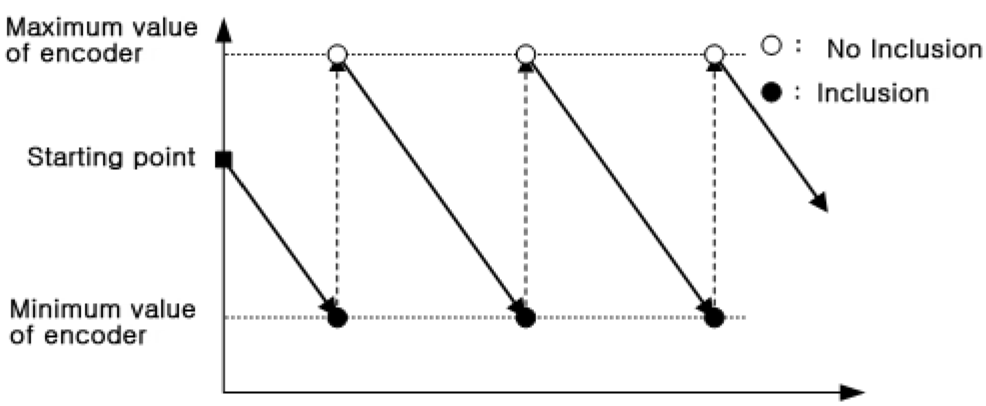
1) The range of the encoder value is set to the maximum and minimum values of encoder when counting the input pulse from the encoder signal of servo drive or manual pulse generator and indicating it to encoder value.

2) Operations are as shown in the figure below.

- In case of the increase in the encoder value



- In case of the decrease in the encoder value



(f) Encoder speed unit

This is used to set the speed display unit of the encoder and sets the reference unit of the speed value.

0: If it is set to ‘0: unit/sec’, it is applied by the rate of change per second form the position of the relevant unit set in the「unit」 parameter. For example, if the ‘Unit’ setting is mm, the speed display unit is 'mm/s'.

1: Unit/min', it is applied as the rate of change per minute of the corresponding unit position set in ‘Unit’ parameter. For example, if the ‘Unit’ setting is mm, the speed display unit is 'mm/min'.

When set to ‘2: rpm’, it is applied as the rpm. To display the rpm, it is used the values set in the ‘Number of pulses per rotation’ and ‘Travel distance per rotation’ parameter.

(g) Encoder input filter value

Set the filter value to limit the frequency of the pulse input to the encoder.

Possible values are 0 ~ 6 and the meaning of each value is as follows.

0: Does not limit the frequency of pulses input to the encoder.

1: Limit the frequency of the pulse input to the encoder to 500kPPS.

2: Limit the frequency of the pulse input to the encoder to 200kPPS.

3: Limit the frequency of the pulse input to the encoder to 100kPPS.

4: Limit the frequency of the pulse input to the encoder to 10kPPS.

5: Limit the frequency of the pulse input to the encoder to 1kPPS.

6: Limit the frequency of the pulse input to the encoder to 0.2kPPS.

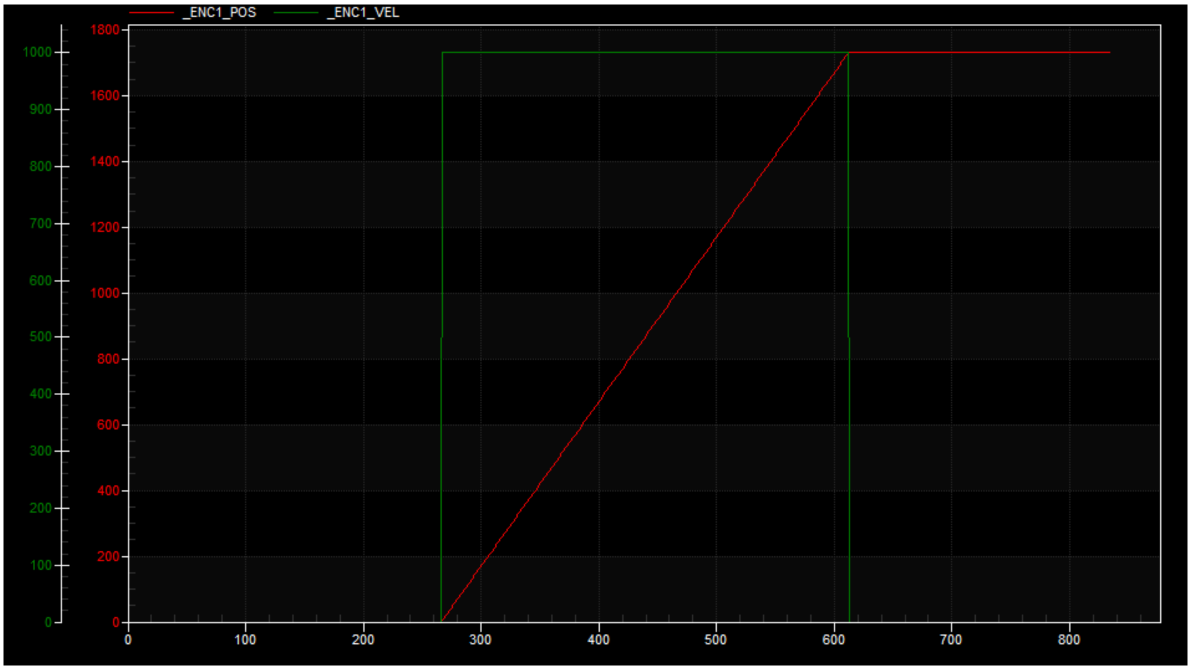
(h) Encoder position filter time constant

Set the time for calculating the position average of the encoder input from the outside. (Unit: ms) When set to '0', the position filter time constant is not applied.

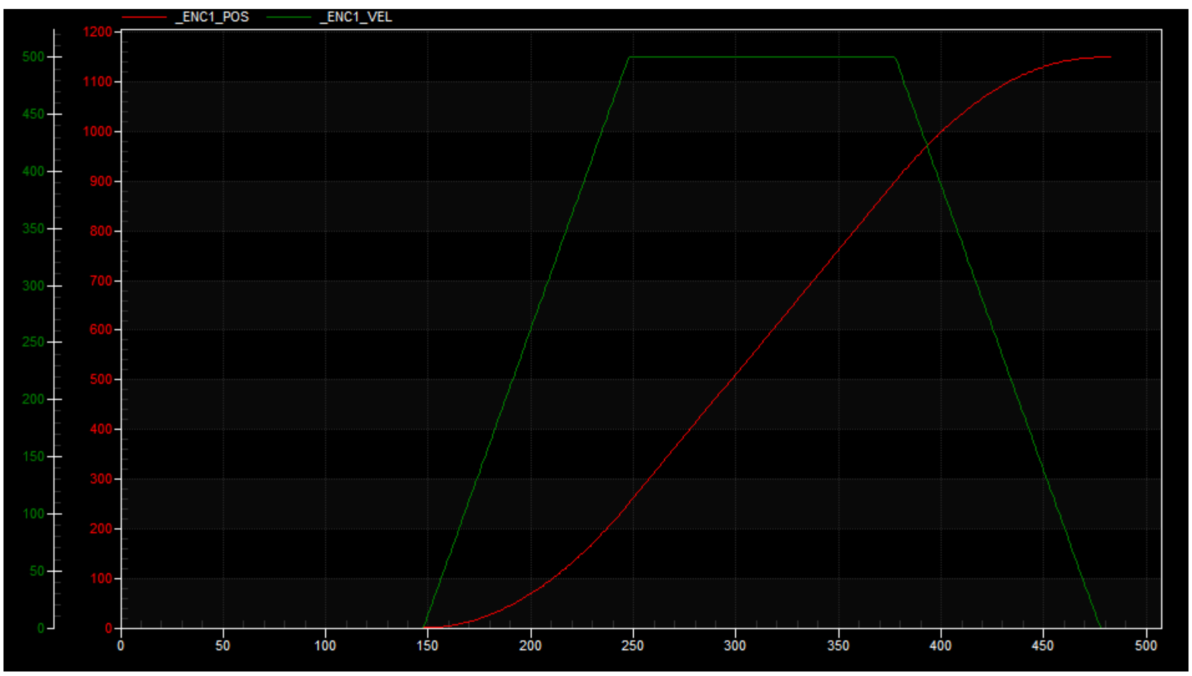
If the deviation of the current position is severe, such as when the ‘Unit’ setting of the encoder is ‘0: pulse’, a stable position can be obtained by applying the position average to the current position.

The following is the trace of the current position input from the external encoder. You can check the difference of the trace position according to the position filter time constant value.

1) Position filter time constant = 0 ms



2) Position filter time constant = 1,000 ms



(4) EtherCAT parameter

It describes the items related to EtherCAT network settings.

When modifying the EtherCAT parameters, make sure to write the EtherCAT parameters in the Project Write menu.

(a) Master

It sets the master functions related to the EtherCAT slave connection when connecting to the network.

The items for master setting are as follows.

1) Registration information

Item	Content	Setting range	Initial values
Slave Revision Check	Specify whether to check the revision information of the parameter matches the revision value of the actual slave when connecting to the network.	0: Do not check 1: Check	0: Do not check
Slave Serial Number Check	Specify whether to check the serial number information of the parameter matches the serial number value of the actual slave when connecting to the network.	0: Do not check 1: Check	0: Do not check
Count of periodic communication time-out	Specify the basic number of times to generate periodic communication time-out errors	1 ~ 8	2

a) Slave Revision Check

When connecting to the network, it determines whether to proceed with the connection by comparing the revision information set in the slave parameter with the one of the actual connected slave.

The operations according to the set values are as follows.

- '0: Do not check'

  The communication connection process is continued without comparing the revision information set in the slave parameter and the one in the connected slave.

- '1: Check'

  It compares the revision information set in the slave parameter with the one in the connected slave, and if a discrepancy is found, the network configuration mismatch error (error code: 0x0F1F) occurs and the communication connection process is terminated.

When the criteria of 'Slave Revision Check' are set to '0: Do not check', if the slave that is incompatible with the Revision of the slave parameter is connected, it may not operate normally. Therefore, make sure to check the compatibility between the Revisions before use.

b) Slave Serial Number Check

When connecting to the network, it determines whether or not to continue the connection process by comparing the serial number information set in the slave parameter and the one of the actual connected slave.

The operations according to the set values are as follows.

- '0: Do not check'

  The communication connection process is continued without comparing the serial number information set in the slave parameter and the one in the connected slave.

- '1: Check'

  It compares the serial number information set in the slave parameter with the one in the connected slave, and if a discrepancy is found, the network configuration mismatch error (error code: 0x0F1F) occurs and the communication connection process is terminated.

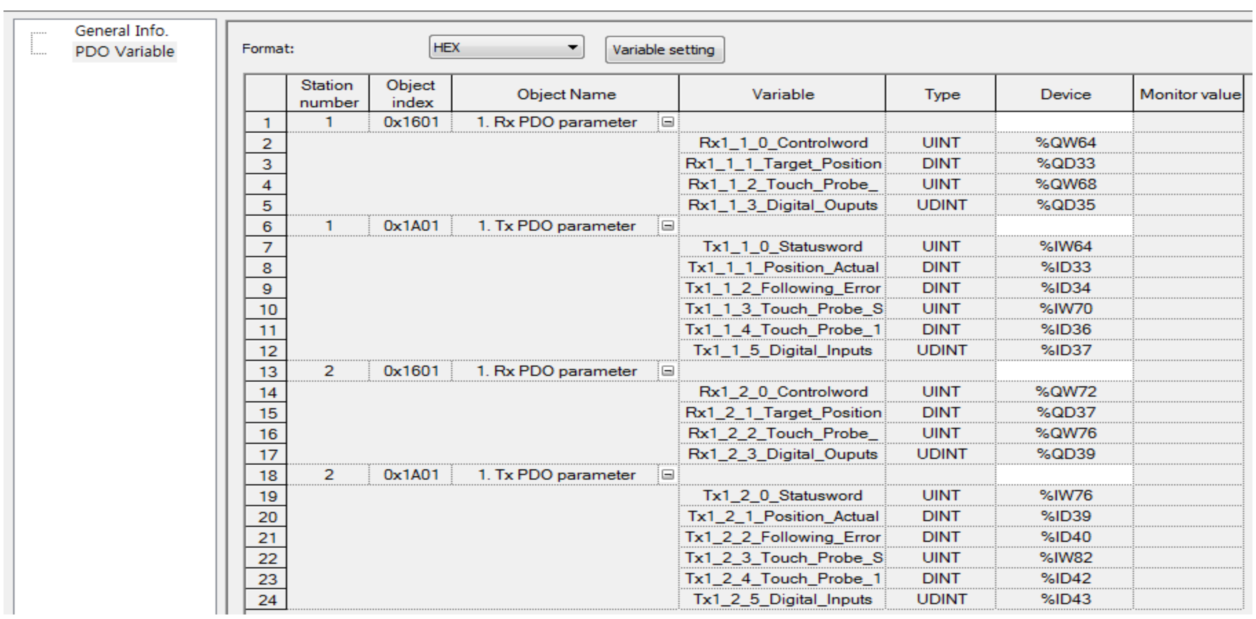
If the 'Slave Serial Number Check' is set to 1: Check', you can see the changes of the network when the network configuration order is changed or the slave is replaced so it is useful for maintenance such as resetting the slave parameters, etc. You need to reset the serial number in XG5000 to connect to the changed network configuration.

c) Count of periodic communication time-out

It specifies the basic number of times to generate time-out errors if the periodic data is not received during the periodic communication between the motion control module and the slave device.
When the communication time-out error occurs frequently in various noise environments (power surges, inductive noise or noise interference between the motion control module and the slave devices' wiring, etc.), set the set value higher.
The available setting range is between 1 to 8 times.

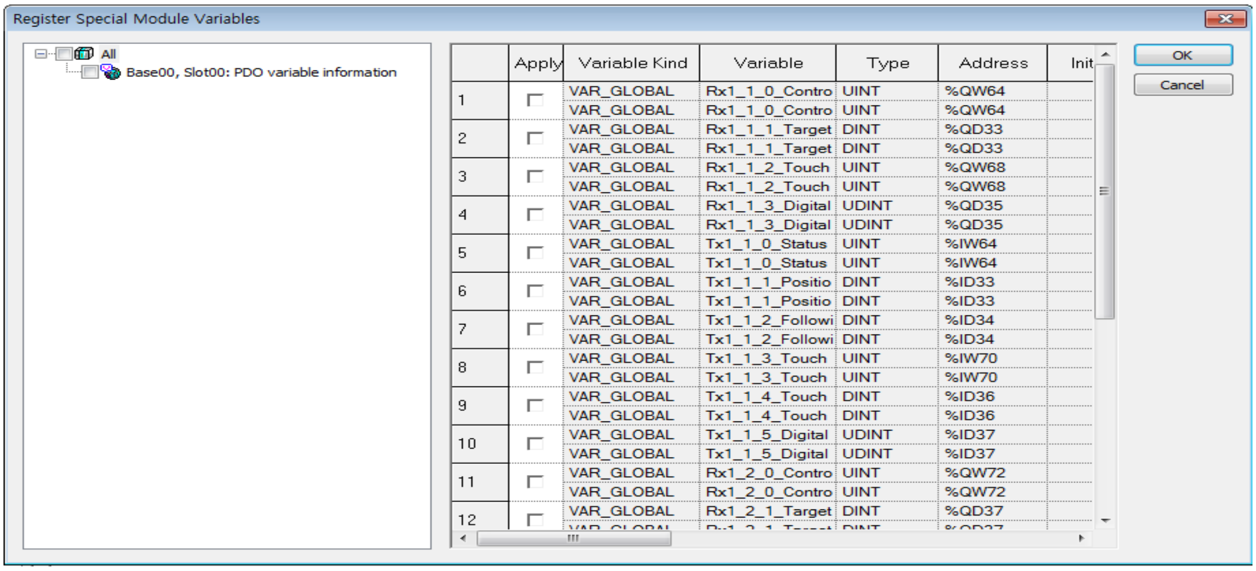
2) PDO variable information

It shows the information on the memory allocation of PDO data of the registered slave.



	Station number	Object index	Object Name	Variable	Type	Device	Monitor value
1	1	0x1601	1. Rx PDO parameter				
2				Rx1_1_0_Controlword	UINT	%QW64	
3				Rx1_1_1_Target_Position	DINT	%QD33	
4				Rx1_1_2_Touch_Probe_	UINT	%QW68	
5				Rx1_1_3_Digital_Ouputs	UDINT	%QD35	
6	1	0x1A01	1. Tx PDO parameter				
7				Tx1_1_0_Statusword	UINT	%IW64	
8				Tx1_1_1_Position_Actual	DINT	%ID33	
9				Tx1_1_2_Following_Error	DINT	%ID34	
10				Tx1_1_3_Touch_Probe_S	UINT	%IW70	
11				Tx1_1_4_Touch_Probe_1	DINT	%ID36	
12				Tx1_1_5_Digital_Inputs	UDINT	%ID37	
13	2	0x1601	1. Rx PDO parameter				
14				Rx1_2_0_Controlword	UINT	%QW72	
15				Rx1_2_1_Target_Position	DINT	%QD37	
16				Rx1_2_2_Touch_Probe_	UINT	%QW76	
17				Rx1_2_3_Digital_Ouputs	UDINT	%QD39	
18	2	0x1A01	1. Tx PDO parameter				
19				Tx1_2_0_Statusword	UINT	%IW76	
20				Tx1_2_1_Position_Actual	DINT	%ID39	
21				Tx1_2_2_Following_Error	DINT	%ID40	
22				Tx1_2_3_Touch_Probe_S	UINT	%IW82	
23				Tx1_2_4_Touch_Probe_1	DINT	%ID42	
24				Tx1_2_5_Digital_Inputs	UDINT	%ID43	

If you want to register the variable name and use it in the program, you can register the variable to be used in the program by selecting "Register Variable".



(b) Slave

1) General Information

Check the information of EtherCAT slave to be used for network connection. It can be identified on the Slave Information tab displayed after executing ‘Open’ of each slave connected to the sub-trees of [EtherCAT parameters]-[Slave] on the XG5000 project tree. To add slaves (servo drive, EtherCAT I/O, etc.), the EtherCAT parameters should be written in the Project Write. The general information items of the slave are as follows

Item	Content	Setting range	Initial values
Slave name	Select the slave and displays the name of the selected slave	XML	-
Station number	Display the station number to be applied to the selected slave.	-	1(Increases automatically when adding the slave)
Vendor	The vendor name of the selected slave is automatically displayed.	Unavailable	-
Version	The revision of the selected slave is automatically displayed.	Unavailable	-
Serial No	The serial number of the selected slave is displayed.	Unavailable	0x1600 PDO Map information
Whether DC is used	Set whether or not to use the DC of the slave.	0: unused 1: used	1: used
Replacement function during connection	Set whether the slave can be replaced during the EtherCAT communication.	0: unused 1: used	0: unused

a) Slave name

It selects the slave to be connected to the motion control module and displays the name of the selected slave. L7NH servo drive is selected as the initial value when adding the slave to the slave data.

When selecting the slave, the slave information is retrieved from the XML file in the folder below to display the available list.

→\EtherCATXML folder in XG5000 installation folder’

If there is a slave to be newly added, copy the corresponding XML file to the above folder and then, restart XG5000 or execute the 'ESI Rescan' menu which is activated by right-clicking in the 'ESI Library' window.

b) Station number

It displays the station number applied to the selected slave. The display range is from 1 to 64 and it cannot be arbitrarily changed by a user.

To change the slave station number, select the slave in the project tree and among menus by right-clicking, execute the 'Properties' menu and then, change the station number on the slave information.

However, the station number is automatically set according to the order of connection when the slave is connected automatically.

c) Vendor

The vendor name of the selected slave is automatically displayed. The user cannot change it arbitrarily.

d) Version

The Revision information of the selected slave is displayed automatically. The user cannot change it arbitrarily.

e) Serial number

The serial number of the selected slave is displayed. When "Read Serial Number" is executed during EtherCAT communication, serial number of the current product is displayed.

f) Whether DC is used

If the slave supports the DC function, it is automatically set form the XML file. If you do not want to use the DC function, select ‘0: Unused’.

Notes

DC (Distributed Clock): It is used to synchronize the EtherCAT master with the EtherCAT slave, enabling high-precision synchronous control between the EtherCAT slaves.
The DC shares the time information between the EtherCAT master and EtherCAT slave to synchronize each slave. In order to share the time information, the first slave connected to the motion control module provides the Reference Clock. The Reference Clock distributes the time information to each slave in every communication cycle.

g) Replacement function during connection

While using the cable duplication function, if a slave device previously not in operation due to network disconnection or a failure is restored and connected to the network, this function detects the connection and connects to the network of the individual slave device without having to reconnect the overall network. Then, it provides the connection with the network of the individual slave without reconnection of the whole network.For more details on the function, refer to 8.3.6 ‘Replacement during connection’.

2) PDO setting

RxPDO sets the synchronous data which is transmitted from the motion controller to the slave in every communication cycle. The RxPDO items supported by the relevant slave are automatically set when selecting the slave. You can use the ‘Edit’ function to add or delete objects you want.

Notes

For the slave used as the motion axis, when editing the RxPDO object, the following objectives must be included as they are essential items used in the motion control module.

0x6040:0	Control word
0x607A:0	Target position

The synchronous data allocated here is automatically assigned to I/O devices and it can be registered as I/O variables and referred in the user program.

For example, the 'Controlworld' object of RxPDO synchronous data of L7N servo drive connected to the slave 1 is registered as I/O flag Rx1_1_0_ControlWord (%QW64).

TxPDO sets the synchronous data read from the slave of the motion controller in every communication cycle. When selecting the slave, the TxPDO items supported by the relevant slave are set automatically. You can use the 'Edit' function to add or delete objects you want. When editing the PDO object, the following objects must be included as they are essential items used in the motion control module.

Notes

For the slave used as the motion axis, when editing the TxPDO object, the following objectives must be included as they are essential items used in the motion control module.

0x6041:0	Status word
0x6064:0	Actual position

The synchronous data allocated here is automatically assigned to I/O devices and it can be registered as I/O variables and referred in the user program.

For example, the"Statusword' object of TxPDO synchronous data of L7N servo drive connected to the slave 2 is registered as I/O flag Tx_1_2_0_StatusWord(%IW68).

3) SDO Parameter

Set the SDO (Service Data Object) parameters operated in the slave.

- The parameters are not stored on the motion controller but are operated on the slave.
- For the setting and operation of the parameters, refer to the Appendix 3 Setting Example.

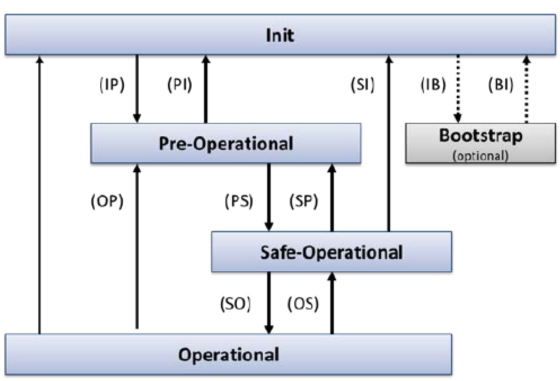
4) Start command

- It is the function to set the specific object during transition of the slave during EtherCAT connecting operation.
- It is used for initialization of the slave parameters as well as slave Rx and TxPDO address assignment and item settings.
- It is provided up to 50 per a slave.

Item	Content	Setting range	Initial values
Transition	Set the transition process in which the object setting function operates.	IP, PS, SO, SP, OP, OS	None
Index	Set the index and sub-index of the object.	XML	-
Data	Set the data to be configured for the object.	Variable depending on data type	-
Description	Add the statement for the object you want to set.	-	-
Flag	Display the flag of the relevant 'Start' command.	Fixed -	-

Notes

The configuration of the transition follows the below EtherCAT state transition diagram.



5) Online service

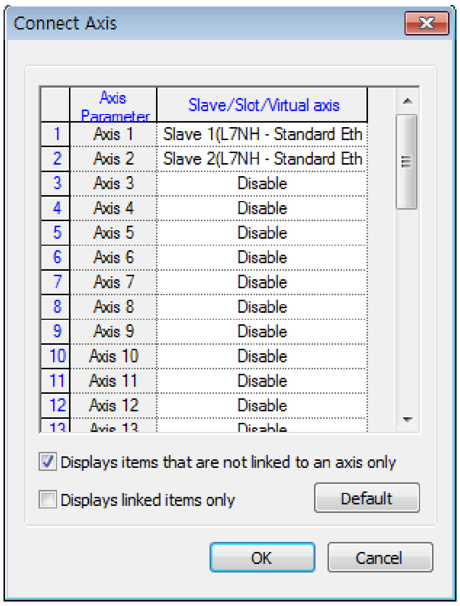
- For more information on the online service, refer to Chapter 08 Motion Control Functions -8.5.FoE Functions.

(5) Axis parameter

(a) Axis/Slave connection

There are two types of axes that can be controlled by the motion controller; a real axis and a virtual axis. The actual axis is the axis allocated to the actual EtherCAT slave, and the virtual axis is arbitrarily generated and controlled within the motion controller. The slave registered as the EtherCAT slave can be assigned as the axis that can be controlled by the motion controller.

You can set the axis in the project tree by selecting [Axis parameters] - [Add item] - [Axis], or [Axis parameters] - [Axis / slave connection].



The axes can be set to "Slave", "Virtual axis", "Disabled". The axis that is set to ‘Disabled’ is not included in the axis parameters.

(b) Axis parameter

1) Basic setting

Basic parameter among basic settings is explained as follows.

Item	Content	Setting range	Initial values
Unit	Set the command position unit of the axis.	0: pulse 1: mm 2: inch 3:degree	0: pulse
No. of pulse per 1 rotation	Set the number of pulses per rotation of motor which corresponds encoder resolution.	1 ~ 4294967295	524288 pls
Transfer distance per 1 rotation	Set the movement amount of the load side moved per rotation of motor.	0.000000001 ~ 4294967295	10 pls
Speed command unit	Set the command speed unit of the axis.	0: unit/sec 1: unit/ min 2: rpm	0: unit/sec

Item	Content	Setting range	Initial values
Speed limit value	Set the maximum speed in case of the speed command of each axis.	Long real (LREAL) positive number	20000000 pls/s
Emergency stop deceleration	Set the deceleration used in the sudden stop conditions.	0 or Long real (LREAL) positive number	0 pls/s^2
Encoder select	Set the type of encoder to be used.	0: Incremental encoder 1: Absolute encoder	0: Incremental encoder
Gear ratio(Motor)	Set gear ratio between the motor and the load.	1~65535	1
Gear ratio(Machine)		1~65535	1
Operation mode of the reverse rotation	Specify the operation method in case operation direction is reversed in the input conditions of newly executed command.	0: Deceleration stop 1: Prompt stop	0: Deceleration stop
Position Control Range Expansion	Set whether to use the used function by expanding a controllable position range when controlling positions.	0: Unused 1: used	0: Unused
Operation mode of Speed Contro	Set the operation mode to use during speed control.	0:CSP 1:CSV	0:CSP
The maximum acceleration of axis group operation	Set the maximum allowable acceleration of axis in axis group look-ahead operation.	Long real (LREAL) positive number	0
The maximum allowable decceleration of axis group operation	Set the maximum allowable deceleration of axis in axis group look-ahead operation.	Long real (LREAL) positive number	0

a) Unit

This is used to set the command unit during motion control, and depending on the control target, the unit of pulse, mm, inch, and degree can be set for each axis.

When changing the setting of the unit, other parameters or variable values are not changed. Therefore, when changing the units, the relevant parameters must be reset so that they can be adjusted to the setting range of the relevant unit.

b) 1 Pulse per revolution

When using mm, inch, and degree for the motion control command units and indicating the speed in rpm, the number of pulses required per motor rotation is set to be used.

c) Travel per rotation

Set the movement amount of the load side per motor rotation when using mm, inch and degree for motion control command unit. How the machine moves from a rotation of motor is determined by the structure of the machine.

d) Speed command unit

The base unit of the value of the speed used for the motion control command is set.

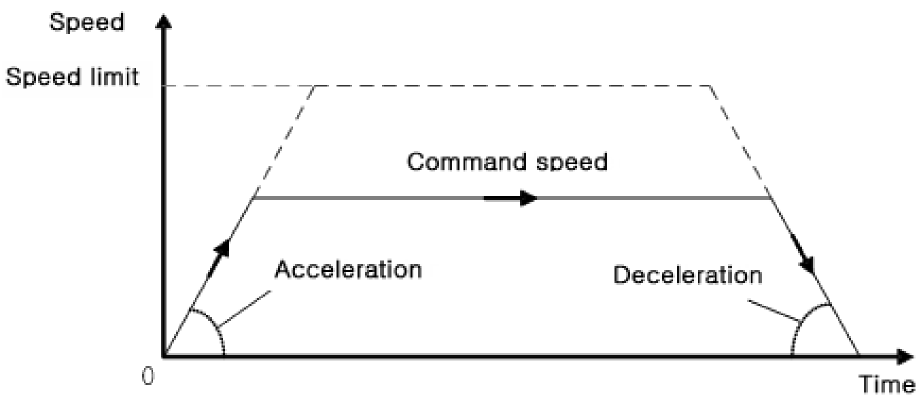
If it is set to ‘0: unit/sec’, it is applied by the rate of change per second of the relevant unit position set in the「unit」 parameter. For example, if the setting of the 「unit」 is in mm, the unit of the speed command is ‘mm/s’.

If it is set to ‘1: unit/min’, it is applied by the rate of change per minute form the position of the relevant unit set in the「unit」

parameter. For example, if the setting of the 「unit」 is in mm, the unit of the speed command is 'mm/min'.
When set to '2: rpm', it is applied as the rpm. If the speed command unit is rpm and it is internally changed to the unit speed, values set in the 「Pulses per rotation」 and 「Travel per rotation」 parameters are used.
When changing the setting of the speed command unit, other parameters or variable values are not changed. Therefore, the related parameters must also be reset according to the setting range of the relevant unit.

e) Speed limit value

Speed limit refers to the maximum rate of the available setting of motion control operation.
When operating the relevant axis, the operation speed should be set below the speed limit set.



f) Emergency stop deceleration

Deceleration in the event of a sudden stop sets the deceleration for situations where a sudden stop needs to be made while operating the axis due to internal or external factors.
Conditions for emergency stop are as follows.

- In case the software upper limit/lower limit is detected
- In case the operation speed of the serve axis exceeds the speed limit in synchronized operation (gear, cam)
- In case the setting for「error level of tracking error」 is '1: alarm' and the error of tracking error occurs
- In case the emergency stop command is executed during the test operation in XG5000
- In case an error occurs in the command executed while axis is currently operating during the checking of execution conditions

  (Except for occasions when restarting the command or ContinuousUpdate is activated.)

g) Select the Spindle Encoder

Set the type of encoder to be used. When using the absolute position system, select 1: absolute encoder.

The following shows the setting of “Encoder select”

Item	Setting Value	Content
Encoder select	0: Incremental encoder	After power on/off, the previous location of servo motor is not maintained. After power of/off, origin fix state is off.
	1: absolute encoder	The absolute position system is activated. After power on/off, the previous location of servo motor is maintained. Origin fix state maintain last condition before power on/off.

In order to use the absolute positioning system, the following three conditions must be satisfied.

- Servo motor: using absolute encoder
- Servo drive SDO parameter absolute encoder setting (0x2005) = 0
- Encoder selection of axis parameter – ‘1: Absolute encoder’ setting

h) Gear ratio (Motor, Machine)

Set gear ratio between the motor and the load. If it is a structure that the load side rotates n times when the motor side rotates m times, set the gear ratios as below.

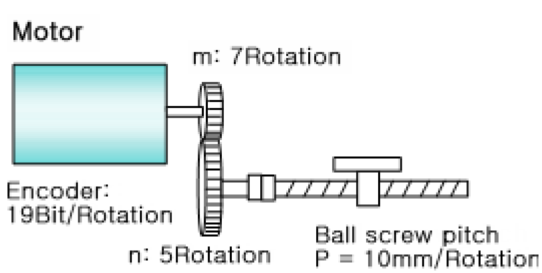
• Motor side gear ratio = m

• Machine side gear ratio = n

If the 「unit」 setting is ‘0: pulse’, this parameter is invalid.

[Example]

When the machine which is moved by ball screw is connected to the encoder with gear, the setting of the encoder unit / Encoder Pulses per rotation / Encoder Travel per rotation is as follows.



· unit: mm
· Pulses per rotation = 524288 (19Bit Encoder)
· Travel per rotation = Ball screw pitch
  = 10.0 mm
· Gear ratio(Motor) = 7
· Machine side gear ratio = n

Notes

If [Unit]is set to ‘0: pulse’ in the above [Setting example], it will move to the position corresponding to the number of encoder pulses without regards to the motor side gear ratio or machine side gear ratio.

That is, the instructions of 524,288 * 7/5 = 734,003 pulse should be issued in order to move 10mm.

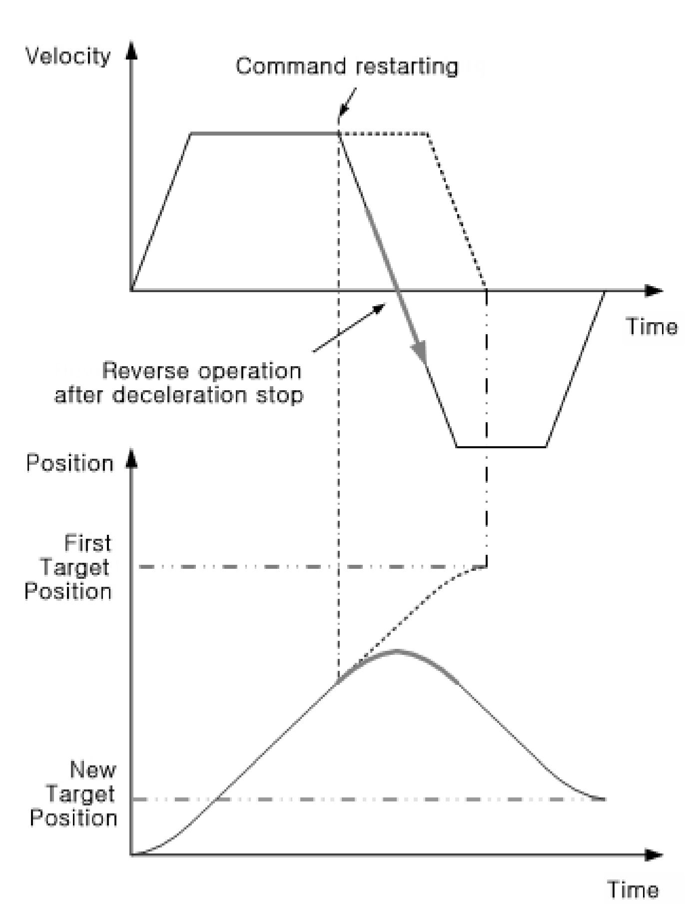
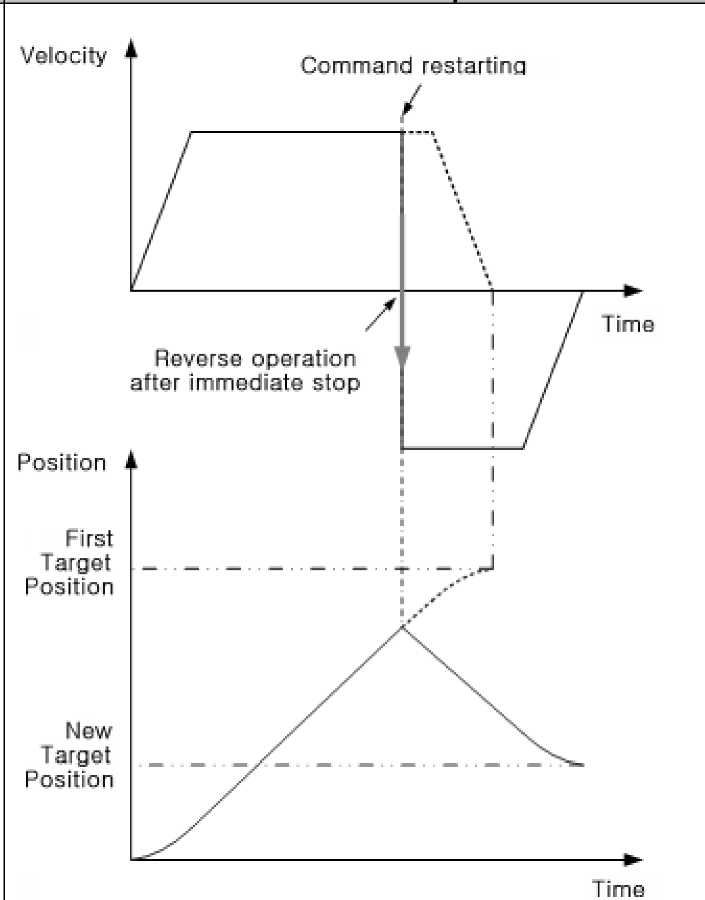
i) Operation mode of the reverse rotation

Specify the operation method in case operation direction is reversed in the input conditions of newly executed command. When starting or restarting the command which the BufferMode is Aborting, or activating ConinuousUpdate, in case where the command condition and the current operating direction are in reverse of each other, stop it by following the method set in the parameter, and start operation in the set speed.

If the BufferMode is not Aborting, it is run in the specified continuous running method in the BufferMode rather than the method set in the parameter.

- '0: Deceleration stop'
  When the operation direction is reversed by the condition of newly executed command, make a deceleration pause to 0 speed and continue accelerating to the target position or operate at the targeted speed.

- '1: Prompt stop'
  When the operation direction is reversed by the condition of newly executed command, stop immediately and continue operating in the opposite direction in the same operation speeds to the target position or at the targeted speed.

0: Deceleration stop	1: Immediate stop
	

j) Position Control Range Expansion

Set whether to use the used function by expanding a controllable position range when controlling positions. An error occurs when a position exceeds the position control range after conversing the unit position set to LREAL into the pulse unit when specifying the target position in motion control commands.

The position control range according to parameter setting is as follows.

Item	Setting value	Position Control Range
Position Control Range Expansion	0: Unused	Integer type 32 bits, -231 ~ 231-1 (-2,147,483,648 ~ 2,147,483,647)
	1: used	Integer type 48 bits, -231 ~ 231-1 (-140,737,488,355,328 ~ 140,737,488,355,327)

k) Speed control operation mode

Set the operation mode to use during speed control command like MC_MoveVelocity command

The speed control operation mode can be set to 0: CSP 1: CSV

l) Axis group maximum allowable acceleration/deceleration.

The maximum allowable deceleration of axis group operation maximum allowable acceleration/deceleration operation is applied when using look-ahead function in axis group operation.

When exceeding the maximum allowable acceleration/deceleration during operating axis group look-ahead function, operate decelerating not to exceed axis acceleration. For details, refer to chapter 8.4.13 Axis group operation maximum acceleration limit function.

2) Expansion setting

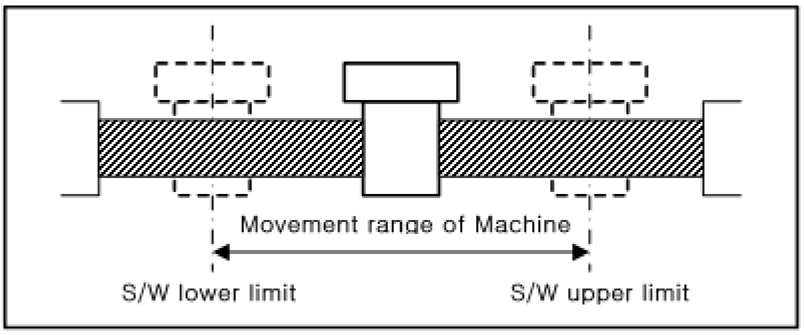
The following explains extended parameter of operation parameter

Item	Content	Setting range	Initial values
S/W upper limit	Set the range of software limit function	Long real(LREAL)	2147483647 pls
S/W low limit			-2147483648 pls
Infinite running repeat position	Set repeated position range value in case of being used as infinite running repetition mode	Long real (LREAL) positive number	360 pls
Infinite running repeat	Set whether to allow infinite running repetition operation function.	0: Disable 1:Enable	0: Disable
Command in-position range	Set the range where in-position signal is On before completion of positioning.	0 or Long real (LREAL) positive number	0 pls
Tracking error over-range value	Set the value to detect more than position deviation.	0 or Long real (LREAL) positive number	0 [unit]
Tracking error level	Set the error level more than deviation.	0: warning 1: alarm	0: warning
Current pos. compensation amount	Set the compensation threshold to indicate the current position value as the target position value.	0 or Long real (LREAL) positive number	0 [unit]
Current speed filter time constant	Set the time to calculate movement average of the current speed.	0 ~ 100	0 ms
Error reset monitoring time	Set the monitoring time in case of resetting error that occurs in servo drive	1 ~ 1000	100 ms
Software limit during speed control	Set whether the soft limit is detected during the speed control.	0: Not detect 1 : detect	0: Not detect
Override mode	Set the method of applying the input value, when override command is executed.	0: Specified by ratio 1: Specified by unit	0: Specified by ratio
JOG high speed	Set the values of speed / acceleration / deceleration / jerk which is referred in jog operation command	Long real (LREAL) positive number	100000 pls/s
Jog low speed			10000 pls/s
JOG Acceleration		0 or Long real (LREAL) positive number	100000 pls/s^2
JOG Deceleration			100000 pls/s^2
JOG Jerk			0 pls/s^3
Backlash compensation amount	Set the Backlash compensation amount.	0~65535	0
Drive absolute position error detection	Set the operation mode while the drive absolute position error occurs	0: not detect 1: homing initialization	0: not detect

a) Software upper/lower limits

This is a function which sets the available range of the movement of the machine in the way of software by setting the upper limit & lower limit and allows the machine not to be operated beyond the set range. That is, this function is used to prevent any breakaway by incorrect operation position setting and incorrect operation by user program fault.

External input upper/lower limit can be also set besides the software upper/lower limit.



The range check of the software upper limit and lower limit is conducted at the beginning of operation and during the operation.

If the soft upper limit and lower limit is detected, an error occurs and the module suddenly stops a motor. Therefore, check the cause of the error and use it after resetting the error when restarting the operation.

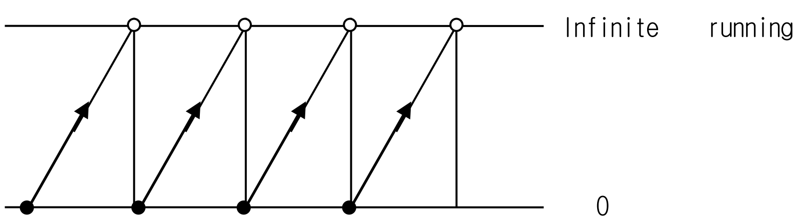
If the software upper/lower limit was set by default values (upper limit: 2,147,483,647, lower limit: -2,147,483,648) or same value, then it would not detect software upper/lower limit.

b) Infinite running repeat position

When using in infinite running repeat mode, set the position value which is repeated.

This is applied when the setting of extended parameter, 「Infinite running repeat」 parameter, is '1:Enable'.

When the 「Infinite running repeat」 parameter is '1:Enable', the command position and current position is indicated as "0~ (infinite running repeat position of -1). 「Unit」 = 0: pulse based



c) Infinite running repeat

Set whether to allow infinite running repetition operation function.

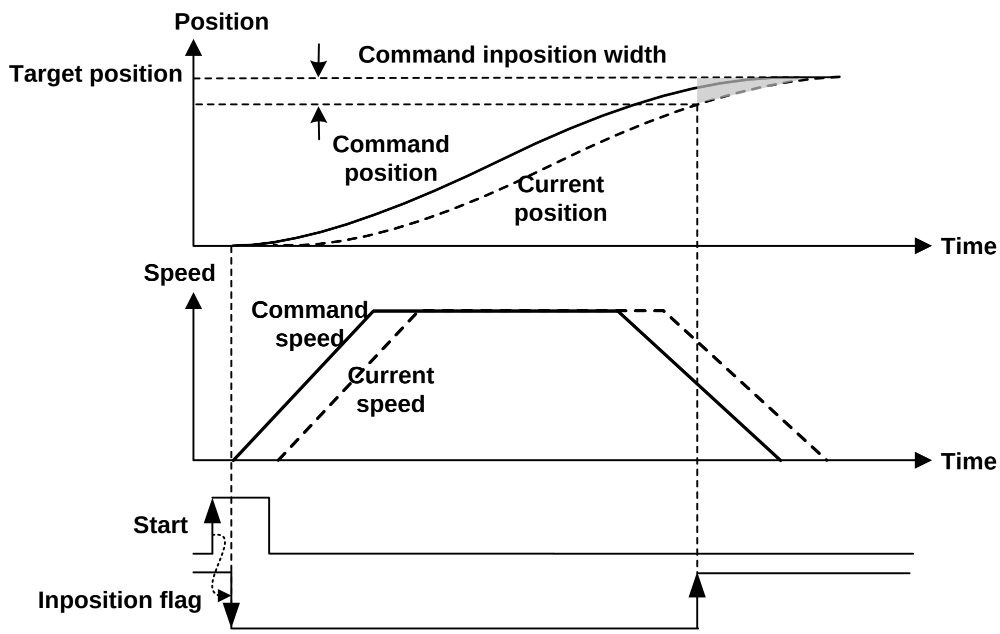
If this parameter is set to '1: Enable', the display of the command position and current position is updated periodically and automatically in the range set in the infinite length repetition position.

You must set it to '0: Disable' when you are not using the infinite running repeat operation function.

d) Command in-position range

//This item sets the distance to the target position where in-position flag (_AXxx_INPOS) is On.
When starting up the motion control, the in-position flag (_AXxx_INPOS) is Off, and it is On when the current position goes inside the「Command in-position range」from the target position.
In-position flag can be used as a trigger when executing other assistant work before completing the position control.//

This item sets the distance to target position where the In-position flag (_AXxx_INPOS) becomes on.
During operating motion control, the in-position flag((_AXxx_INPOS) turns off and when the difference between target command position and current position is less than the command in-position width, the in-position flag((_AXxx_INPOS) turns on. the in-position flag can be used as trigger when executing other auxiliary tasks before completing position control.



e) Exceeding value of tracking error

Set the value to detect more than position deviation. If a value exceeds this range, the 「Over deviation warning (_AXxx_DEV_WARN)」 or 「Over deviation alarm(_AXxx_DEV_ERR)」flag is On.
If this set value is 0, it won't detect the value over the deviation. You can set whether you want it to be a warning or an alarm for over deviation in the 「Error level of tracking error」of the expanded parameter.

f) Tracking error level

Set whether to make it a warning or an alarm when the value over deviation is detected.
The operations according to the set values are as follows.

- '0: warning'
  When an error occurs in tracking error, the 「Over deviation warning (_AXxx_DEV_WARN)」flag is On, and warning error of tracking error (error code: 0x101D)) occurs. The axis continues to operate without stopping.
- '1: alarm'
  When an error occurs in tracking error, the 「Over deviation alarm (_AXxx_DEV_ERR)」flag is On, and the alarm error of tracking error (error code: 0x101C) occurs. The axis suddenly stops at the「Emergency stop deceleration」of basic parameter.

In the following situations, the error in tracking error is not examined.

- In case the「Tracking error over-range value」is 0
- In case of the operation in homing or torque control

g) Current position compensation amount

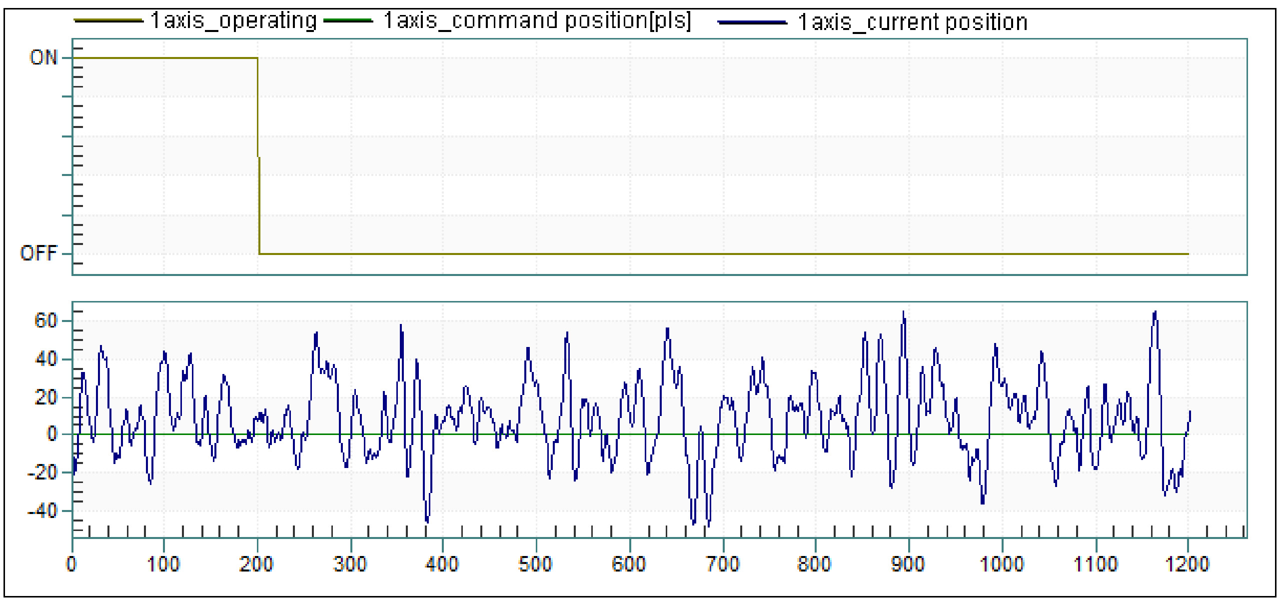
Current position compensation amount is a parameter unit used to display the current position value as the command position when the servo motor's current position value is not displayed as a fixed value but changed slightly depending on the personal setting of the user application and the servo drive.

When it is not in operation and if the difference of the command position and the current position is within the amount of compensation in displaying current position, the current position value is displayed as a command position value. When it is in operation, Current position compensation amount is not reflected, and the actual position value is displayed.

The following is an example of application of Current position compensation amount according to the value of Current position compensation amount when the command position is '0'.

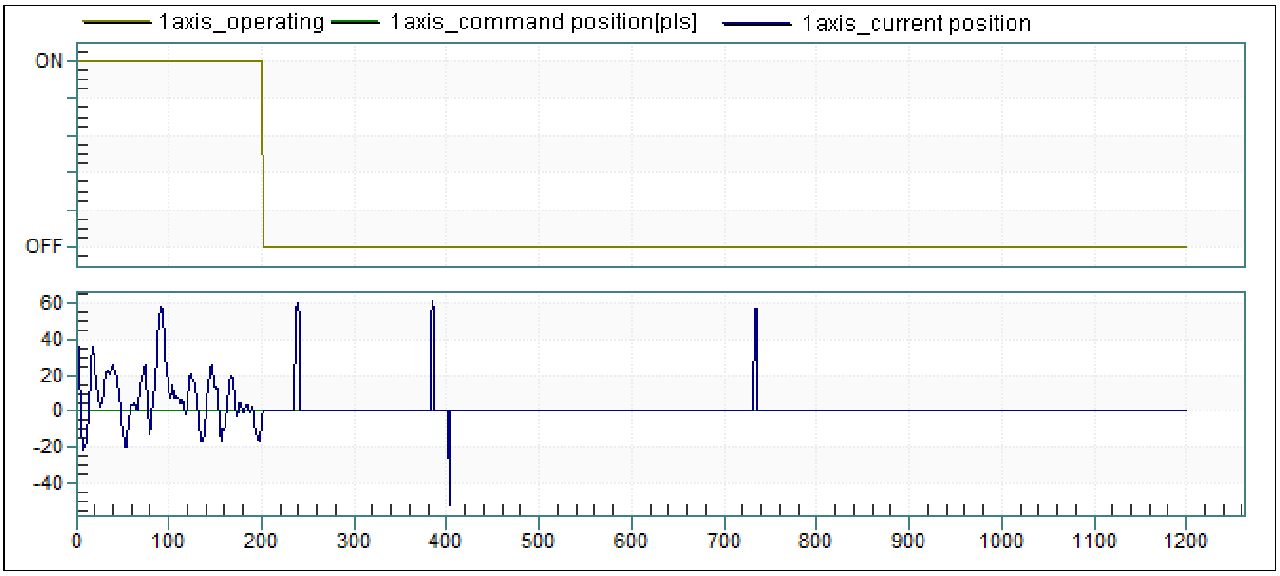
① Current position compensation amount = 0 pls

Position value of the actual motor is displayed as the current position value even after the end of operation.



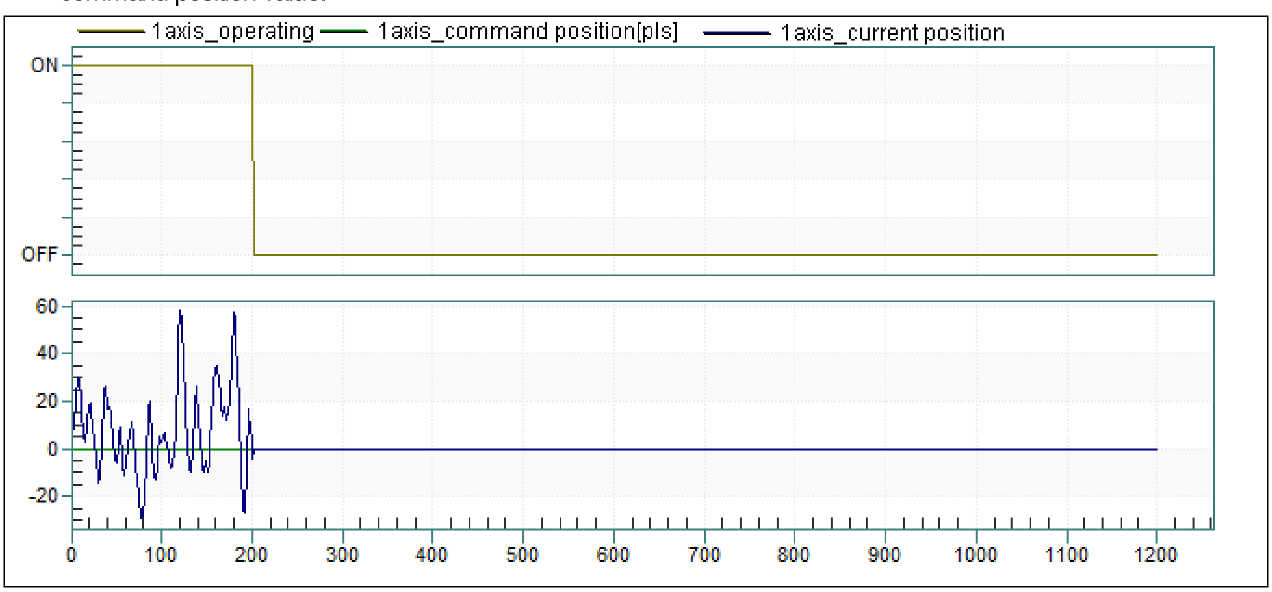
② Current position compensation amount = 50 pls

If the current position value is within ±50 of command position after the end of operation, it is displayed as the command position value.



③ Current position compensation amount = 100 pls

If the current position value is within ±100 of command position after the end of operation, it is displayed as the command position value.



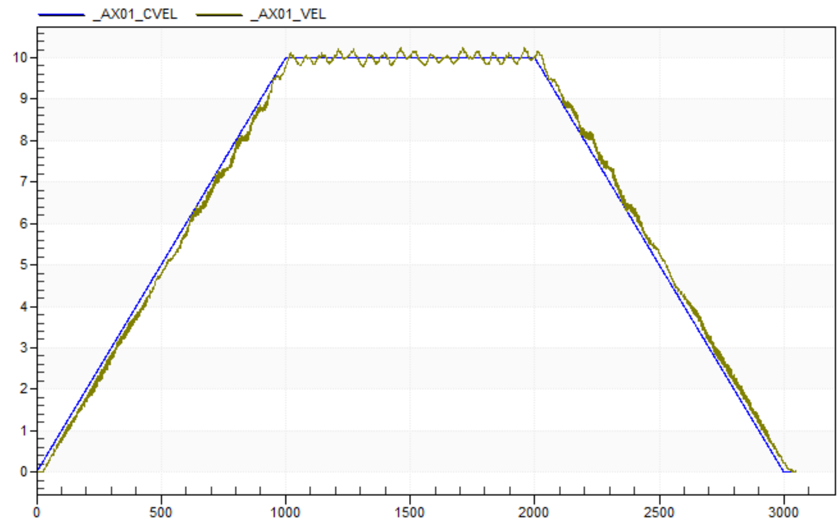
h) Current speed filter time constant

Set the time to calculate movement average of the current speed. (unit: ms) Current speed filter time constant is not applied if it is set to '0'.

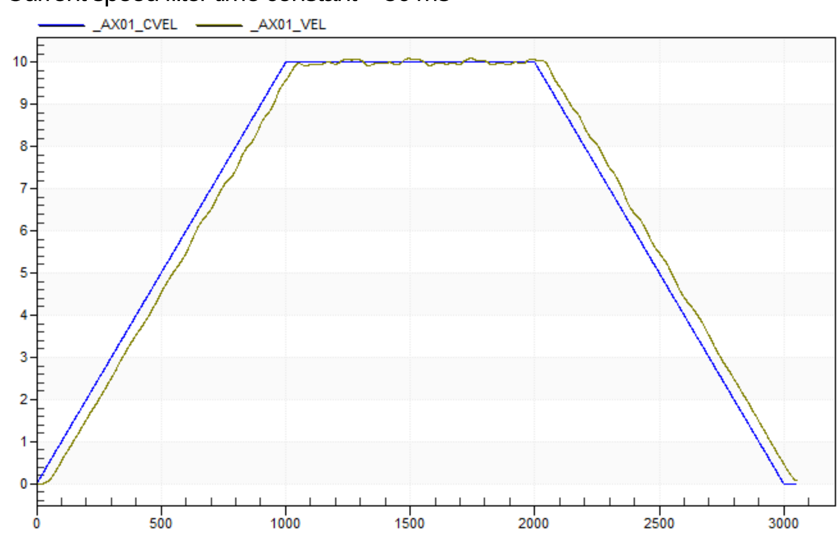
When the speed of axis is slow or there are wide variations in current speed (ex. 「unit」 setting is '0: pulse', stable speed can be achieved by applying the average of movement to the current speed.

You can check the differences in current speed depending on the value of Current speed filter time constant in the list below which traces command speed and current speed at 10 mm/s of command speed.

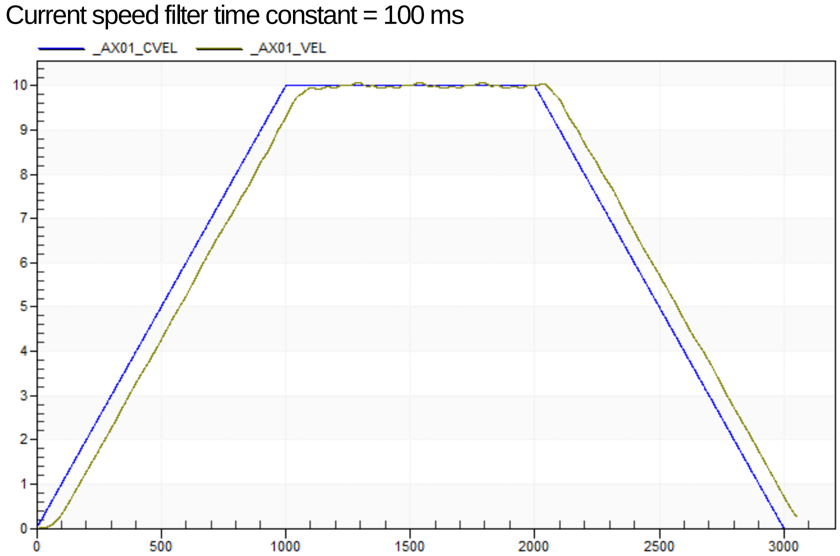
① Current speed filter time constant = 0 ms



② Current speed filter time constant = 50 ms



③ Current speed filter time constant = 100 ms



i) Error reset monitoring time

Set the monitoring time in case of resetting error that occurs in servo drive (Unit: ms)

If the error which occurred in the servo drive within the error reset monitoring time, error reset monitoring is terminated and error reset time out error of servo drive (error code: 0x1070) is occurred.

j) Software limit during speed control

When software limit is detected during the operation at fixed speed by speed control, this is used to stop the motor.

The operations according to the set values are as follows.

- ‘0: Not detect’
  If it is under the speed control even when the software limit function is activated, software limit is not detected.
- ‘1 : detect’
  If it is under the speed control even when the software limit function is activated, software limit is detected.

Even when the parameter value is set to '1: detect', if the software upper limit/lower limit is set to the initial value (upper limit: 2,147,483,647, lower limit:-2,147,483,648) or the same value, software limit is not detected.

k) JOG high speed / JOG low speed

Jog speed is related to Jog operation (a kind of manual operation) and has 2 types of operation: Jog low speed operation and Jog high speed operation.

Jog is operated in the pattern with the areas of acceleration, fixed speed, and deceleration. Therefore, the acceleration area is controlled by jog acceleration time and the deceleration area is controlled by jog deceleration time.

Setting range of JOG high speed cannot exceed the speed limit. Also, JOG high speed must be the same with or bigger than JOG low speed.

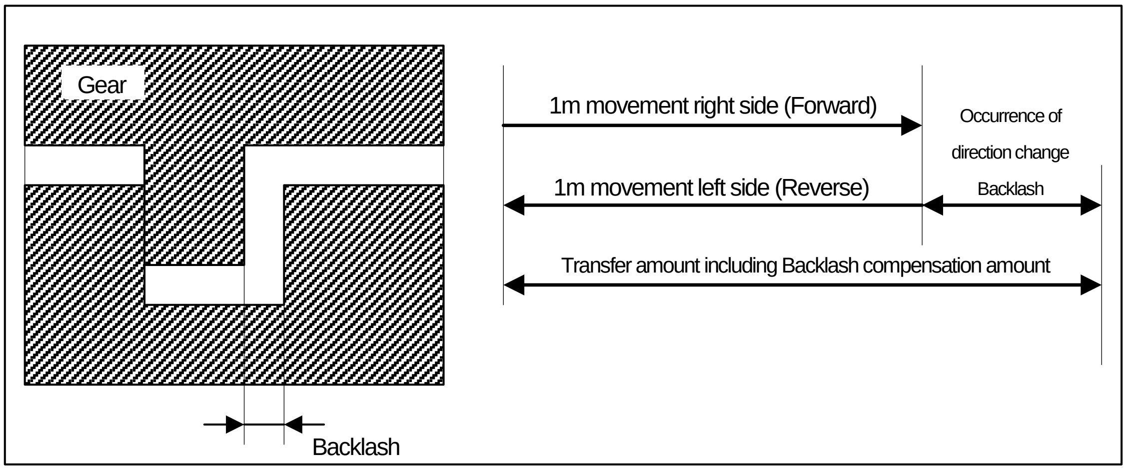
l) JOG acceleration, JOG deceleration, JOG jerk

Set the values of acceleration, deceleration, and jerk which are applied in the case of JOG high speed and JOG low speed operation. If JOG acceleration is 0, it is operated immediately at JOG set speed without acceleration area at the beginning of JOG operation. If JOG deceleration is 0, it is stopped immediately at 0 without deceleration area at the stop of JOG operation. If JOG jerk is 0, the form of acceleration/deceleration is in a linear as acceleration is fixed

m) Backlash compensation amount

If a gear, a screw and more is combined to the motor axis, the tolerance of the machine does not work by wear, when the rotation direction changes, is called ‘Backlash’. Therefore, when changing the rotation direction, a backlash compensation amount should be added to the amount traveled to avoid the machine position tolerance.

As presented in the following figure, if the position is moved 1m to the right and again 1m to the left, it is not possible to reach the original position by backlash.



The backlash compensation amount can be set to the pulse unit from 0 to 65535. If the 'Unit' parameter value is not '0: pulse', Please set a range as follows:

0 ≤ [Backlash compensation amount X (Pulse number per rotation / Transfer distance per rotation) X (Gear ratio on the motor / Gear ratio on the machine)] ≤ 65535

Notes

1. The 'backlash compensation amount' parameter generally operates only on the axis connected to the S axis of the NC channel that is for the spindle axis on which the spindle device does not support the backlash compensation function. For the NC channel/axis excluding general axes and the NC S axis, use the backlash function that a servo drive supports.
2. The 'backlash compensation amount' is output by adding the amount traveled to the original position when the traveling direction of machine changes after becoming the origin fix state. The backlash compensation amount applies only to the position control operation such as the spindle orientation operation but does not apply to the speed control.

n) Drive absolute position error detection function

The homing status can be initialized in drive using drive absolute error detection function if initializing homing status while an absolute position error is detected. It can be used only the drive that supports an absolute position valid signals.

	Drive
LS	L7NH, PEGASUS, iX7NH

① When the absolute positon of drive occurs damaged situation (the encoder cable open) in the homing status, it is used to initializing homing status and prevent malfunction of operation caused by an absolute position error.

② This function only operates drive that supports an absolute position valid signal.

③ The drive absolute position error detection of extended parameter is 1: the function operates when homing status is set to initialization.

④ If detecting that an absolute position valid signal of drive change from On to Off through EtherCAT communication, the homing status of the relevant axis is initialized and axis error (absolute position abnormality detection error: 0x1240) occurs.

⑤ If the absolute position valid signal is off while the communication is connected through EtherCA connection command, the homing status of relevant axis is initialized.

⑥ When changing the drive absolute position error detection of extended parameter from 0: not detect to 1: drive absolute position error detection, if the absolute position valid signal of relevant axis is off, the homing status of relevant axis is initialized and an axis error (an absolute position abnormality detedction error: 0x1241) is occurred.

⑦ If executing the current position setting function while drive absolute position valid signal of relevant axis is off, the axis errors (an absolute position abnormality detection error during executing the current position setting: 0x1096) occurs

3) NC Spindle Axis Setting

Explain about the NC Spindle Axis Setting of axis parameters.

Item	Content	Setting range	Initial values
Select the Spindle Encoder	Set the method that an encoder attached to a motor of the spindle axis is connected.	0: Unused 1: Motor ENC 2: Built-in ENC1 3: Built-in ENC2 4: EtherCAT ENC	0: Unused
Number of pulses per rotation of the spindle EtherCAT encoder	If the 'spindle encoder selection' parameter setting value is '4: EtherCAT ENC', set number of pulses per rotation of an encoder.	1 ~ 4294967295	8192 pls
Spindle EtherCAT encoder position variable/address	If the 'spindle encoder selection' parameter setting value is '4: EtherCAT ENC', set the device where the current position of the encoder is saved.	%ID0 ~ %ID4095 %MD0 ~ %MD524287	%ID0
The P Gain of the Spindle Positioning Mode	Set the P gain value that the spindle axis uses when controlling position.	1 ~ 500 Hz	30 Hz
The Feed Forward Gain of the Spindle Positioning Mode	Set the feed forward gain value that the spindle axis uses when controlling position.	0 ~ 100 %	0 %

a) Select the Spindle Encoder

The spindle axis is basically operated by speed control. But there are some cases where the position control operation such as the orientation operation is needed according to NC operation. Setting of the 'Spindle Encoder Selection' parameter is needed to operate the spindle axis by position control. In addition, setting of the 'Spindle Encoder Selection' parameter is needed to execute the homing operation to make the spindle axis become the origin fix state.

- 0: Unused

  If the 'spindle encoder selection' parameter is '0: Disable', a spindle axis can only perform the speed control-based operation. Errors occur when running the position control operation.

- 1: Motor ENC

  Set it if the position confirmation is possible by accepting the input of encoder signals of a motor on drives such as a servo drive. The position actual value (0x6064:0) object should be set in the setting of the EhterCAT Slave TxPDO.

- 2: Built-in ENC1

If confirming a position by attaching a separate encoder to the motor of the spindle axis, the position can be confirmed by connecting encoder signals to the built-in ENC1 of a motion controller.
When using the 'built-in ENC1', set the 'built-in parameter - encoder' as follows:

- Unit of Encoder 1 = 0: pulse
- Encorder1 max. value = 2147483647 pls
- Encorder1 min. value = -2147483648 pls

· 3: Built-in ENC2

If confirming a position by attaching a separate encoder to the motor of the spindle axis, the position can be confirmed by connecting encoder signals to the built-in ENC2 of a motion controller.
When using the 'built-in ENC2', set the 'built-in parameter - encoder' as follows:

- Unit of Encoder 2 = 0: pulse
- Encorder2 max. value = 2147483647 pls
- Encorder2 min. value = -2147483648 pls

· '4: EtherCAT ENC'

Set it if confirming a position by attaching a separate encoder to the motor of the spindle axis and connecting it to the EtherCAT slave (high speed counter device). To confirm the encoder position value read from the EtherCAT slave, the following parameter setting is needed.

- Number of pulses per rotation of the spindle EtherCAT encoder
- Spindle EtherCAT encoder position variable/address

b) Number of pulses per rotation of the spindle EtherCAT encoder

If the 'spindle encoder selection' parameter is set to '4: EtherCAT ENC', set the resolution of the encoder attached to a motor.

c) Spindle EtherCAT encoder position variable/address

If the 'spindle encoder selection' parameter is set to '4: EtherCAT ENC', the position value of the encoder read from the EtherCAT slave sets the saved variable/address. You can specify the input variable (I) and direct variable (M).
The address value that can be set according to variables is as follows:

- %ID0 ~ %ID4095
- %MD0 ~ %MD524287

d) The P Gain of the Spindle Positioning Mode

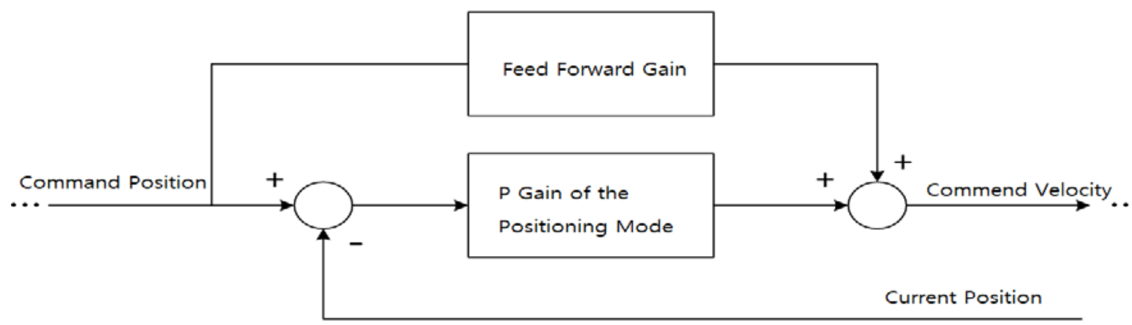
The spindle axis is basically operated with speed control, and the speed command value is output to the device that controls the spindle axis. However, some spindle functions may require position control operation. When the spindle axis executes position control operations such as the homing operation and orientation operation, the position loop that calculates the output speed with the command position and the current position operates.
The 'P Gain of the Spindle Positioning Mode' sets all the responses from the position loop. The bigger the gain value is set the higher the responses become. However, as vibration can occur according to load if setting the gain value too big, adjust the gain value accordingly after taking proper actions for safety.

e) The Feed Forward Gain of the Spindle Positioning Mode

The 'Feed Forward Gain of the Spindle Positioning Mode' is used to add the control value that is proportional to a speed command of the spindle axis from the position loop to the output speed. The bigger the gain value is set the less the tolerance is. But as vibration or overshoot can occur according to load if setting the gain value too big, adjust the gain value accordingly after taking proper actions for safety.

Notes

From the position loop of the spindle axis, the 'P Gain of the Spindle Positioning Mode' and 'Feed Forward Gain of the Spindle Positioning Mode' values are operated as follows.



4) NC Spindle homing Setting

Explain about the NC Spindle Homing Setting of axis parameters.

Item	Content	Setting range	Initial values
How to conduct the homing operation	Set the homing operation method that is run when executing the NC_Home command on the spindle axis.	0: Servo drive supported 33: Reverse direction, Z phase 34: Forward direction, Z phase 35: Set the homing of the current position	0: Servo drive supported
Switch navigation speed of the homing operation	Set the operated speed to detect switch signals after starting the homing operation.	Long real (LREAL) positive number	60 rpm
Zero navigation speed of the homing operation	Set the operated speed to detect zero signals after starting the homing operation.		12 rpm
Acceleration/deceleration of the homing operation	Set acceleration/deceleration to accelerate and decelerate to the target speed after starting the homing operation.	0 or Long real (LREAL) positive number	1000 deg/s^2
Z phase variable/address	Set the device where the Z phase signal used as the Zero signal of the homing operation is saved.	%IX0 ~ %IX131071 %MX0 ~ %MX16777215	%IX0
Orientation velocity	When the M19 Orientation command is executed on the NC program, set the Orientation offset position and velocity, and the traveling direction.	Long real (LREAL) positive number	60 rpm
Orientation direction		0: forward direction 1: reverse direction	0: forward direction
Orientation offset		0 ~ 360	0

a) Homing operation method

Set the homing operation method that is run when executing the NC_Home command on the spindle axis.

- 0: Servo drive supported

If the spindle drive connected to the spindle axis is a servo drive, the homing operation supported by the servo drive is executed. The servo drive parameter used for the homing operation is Homing method (0x6098:0).

- 33: Reverse direction, Z phase

Set the Z phase position as the homing after executing the homing operation on the NC function module of a motion controller and starting reverse operation.

When executing the NC_Home command, execute the homing operation with parameters; zero navigation speed of the homing operation and acceleration/deceleration of the homing operation. Set the device where the Z phase signal used as the Zero signal of the homing operation for ‘Z phase variable/address’ parameters is saved.

- 34: Forward direction, Z phase

Set the Z phase position as the homing after executing the homing operation on the NC function module of a motion controller and starting forward operation.

When executing the NC_Home command, execute the homing operation with parameters; zero navigation speed of the homing operation and acceleration/deceleration of the homing operation. Set the device where the Z phase signal used as the Zero signal of the homing operation for ‘Z phase variable/address’ parameters is saved.

- 35: Set the homing of the current position

It is used when setting the current position of the spindle axis to the homing.

b) Switch navigation speed of the homing operation, Zero navigation speed of the homing operation, Acceleration/deceleration of the homing operation

‘Homing operation method’ parameter is ‘33: reverse direction, Z phase’, ‘34: forward direction, Z phase’, and the homing operation is executed on the NC function module of a motion controller, set velocity and acceleration/deceleration.

c) Z phase variable/address

‘Homing operation method’ parameter is ‘33: reverse direction, Z phase’, ‘34: forward direction, Z phase’, set variables/addresses where the Z phase signal used as the Zero signal of the homing operation is saved. You can specify the input variable (I) and direct variable (M).

The address value that can be set according to variables is as follows:

- %IX0 ~ %IX131071
- %MX0 ~ %MX16777215

d) Orientation velocity

When the M19 Orientation command is executed on the NC program, set the command speed of the Orientation operation.

e) Orientation direction

When the M19 Orientation command is executed on the NC program, set the operation direction of the Orientation operation.

- 0: forward direction
- 1: reverse direction

f) Orientation offset

When the M19 Orientation command is executed on the NC program, set the target position value of the Orientation operation. After starting operation in the direction set to the ‘Orientation direction’ parameter, stop it at the position set to the ‘Orientation offset’. The range of the set value is 0 to 360 degrees.

5) NC Spindle Control Setting

Explain about the NC Spindle Control Setting of axis parameters.

Item	Content	Setting range	Initial values
The tolerance range to reach the spindle rotation command speed	Determine whether to reach the command speed of the spindle axis by the set value.	0 ~ 100 %	95 %
The tolerance RPM to reach the spindle rotation zero speed	Determine whether to reach the zero speed of the spindle axis by the set value.	0~ 100 rpm	5 rpm

a) The tolerance range to reach the spindle rotation command speed

When an axis operates as the spindle axis as it is connected to the NC S axis, the range is used to set the range to confirm whether the current speed value of the spindle axis reaches the target speed.

The 'signal to confirm whether to reach the spindle command speed' (_NC01_SpindleCVelAgr) flag turns On, if satisfying the following conditions:

→ 'Actual transfer speed of the S axis' ≥ ['Target speed of spindle (S command value)' X ('Tolerance range to reach the spindle rotation command speed' / 100)]

b) The tolerance RPM to reach the spindle rotation zero speed

When an axis operates as the spindle axis as it is connected to the NC S axis, the range is used to set the RPM speed value to confirm whether the current speed value of the spindle axis reaches the zero speed.

The 'signal to confirm whether to reach the spindle command speed' (_NC01_SpindleZeroVel) flag turns On, if satisfying the following conditions:

→ 'Actual transfer speed of the S axis' ≤ 'Tolerance RPM to reach the spindle rotation zero speed'.

(6) Axis Group parameter

1) Basic setting

Basic setting item is explained as follows.

Item	Content	Setting range	Initial values
Configuration axis 01~ 10	Set the axis which form axis group.	None 1Axis~ 32Axis(Real/Virtual axis), 33Axis ~ 36Axis(Virtual axis)	None
Interpolation speed max	Set max speed of operation about axis group.	Long real (LREAL) positive number	20000000 u/s
Override mode	Set the override mode.	0: Specified by ratio 1: Specified by unit	0: Specified by ratio
Interpolation operation blending angle limit	Set the whether to use the interpolation operation blending angle limit function.	0: Disable 1: Enable	0: Disable
Interpolation operation blending allowable angle	When using the interpolation operation blending angle limit, set the blending allowable maximum angle.	2 ~ 178, real (LREAL)	178 degree
Coordinate system look ahead settings	Set the number of Coordinate system look ahead buffer.	1 ~ 10, integer	1
Coordinate output filter time	Set the value of Coordinate output	0 ~ 100, real(LREAL)	0 ms

constant	filter time constant.		
Rotation operation/linear operation conversion ratio	Set the ratio of rotation operation/linear operation conversion.	Long real (LREAL) positive number	0

(a) Configuration axis setting

Set the number of each axis which belongs to the relevant axis group. Each axis group can include up to 10 axes.

Virtual axis can also be set in the axis group parameter.

Axis setting must be set in order in axis group which executes circular interpolation or helical interpolation command. In other words, 「axis setting 1」 is X-axis of the arc, 「axis setting 2」 is Y-axis of the arc, and 「axis setting 3」 is Z-axis of helical interpolation.

Therefore, if circular interpolation command is executed when setting the axis group, errors occur as follows.

- In case the axis group is comprised of 4 axes (error code:0x20A9)
- In case the set value of 「axis setting 1」 or 「axis setting 2」 is 'none'(error code: 0x20AA)
- In case the set value of「axis setting 3」is 'none' and the remaining axes are set (Error code: 0x20AA)

(b) Interpolation speed max

This refers to the configurable maximum speed of interpolation control operation when controlling interpolation with axes which belongs to the relevant axis group. In case of interpolation operation of the relevant axis group, interpolation speed must be set below the set Interpolation speed max.

c) Override mode

When executing override command at relevant axis group, set how to apply the input values.

0: Specified by ratio, 1: Specified by unit can be set. If setting 0: Specified by ratio, the factor value of override command operates as ratio for the current setting item. If setting 1: Specified by unit, the factor value operates absolute specified value of setting item.

d) Interpolation operation blending angle limit, Interpolation operation blending allowable angle

Set the blending angle limit function for the relevant axis group. You can set to operation after the deceleration stop in the place where a shock is expected to occur due to the large angle difference between the two operations of blending angle limit. You can set 0: disable , 1: enable in interpolation operation blending angle limit and set the value of 2 to 178 in the interpolation operation blending allowable angle. For details, refer to chapter 8.4.11 Interpolation operation blending angle limit function.

e) Look ahead setting

Set the look ahead function for relevant axis group. The look ahead function is function that determines whether the current operation is acceleration/deceleration by determining the movement distance/speed/acceleration/jerk of the next command in the continuously executed operation. If using the look ahead function, you can solve the deceleration problem due to insufficient deceleration distance when blending continuously the short operation of movement distance. The look ahead setting is 1 to 10. For details, refer to chapter 8.4.14 Look ahead function.

f) Coordinate output filter time constant

By setting the time constant of output filter function for the relevant axis group. If setting the output filter function during executing axis group motion, alleviating acceleration shock that occurs in axis and executing more soft operation. For details, refer to chapter 8.4.15 Coordinate operation output filter function.

g) Rotation operation/linear operation conversion ratio (Angular distance conversion constant)

Set the angular distance conversion constant for the relevant axis group. The angular distance conversion constant is used when calculating the movement distance during executing the coordinate system linear interpolation operation in an axis group. When the values of Rotation operation/linear operation conversion ratio is 0, determining not to be used.

$$\begin{aligned}\text{Movement distance} = \text{Max}(&\text{sqrt}((\text{Target positionX} - \text{current positionX})^2 + (\text{Target positionY} - \text{current positionY})^2 \\ &+ ((\text{Target positionZ} - \text{current positionZ})^2, \text{angle distance conversion constant} \\ &* \text{sqrt}((\text{Target positionA} - \text{current positionA})^2 + (\text{Target positionB} - \text{current positionB})^2 \\ &+ ((\text{Target positionC} - \text{current positionC})^2\ ))\end{aligned}$$

2) Coordinates system setting

Coordinate system setting item is explained as follows.

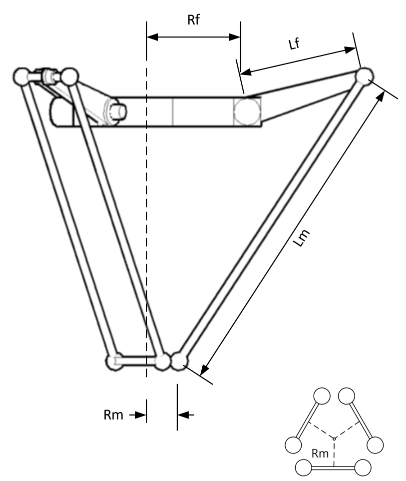
Item	Content	Setting range	Initial values
Coordinate system	Set the type of robot that is applied in the operation of coordinate system.	0: none, 1: XYZ 2: Delta3 3: Delta3R 4: LinearDelta3 5: LinearDelta3R	0: none
Coordinate system parameter	Set the parameters of the machine depending on the type of coordinate system.	-	-

(a) XYZ

XYZ is a robot where the axis set in 「axis setting 1」 in X-axis, the axis set in 「axis setting2」 in Y-axis, and the axis set in 「axis setting 3」 in Z-axis make a one-to-one correspondence and move in Cartesian coordinate. If the type of coordinate system is set to XYZ, there is no need to set the coordinate system parameters.

(b) Delta3/3R

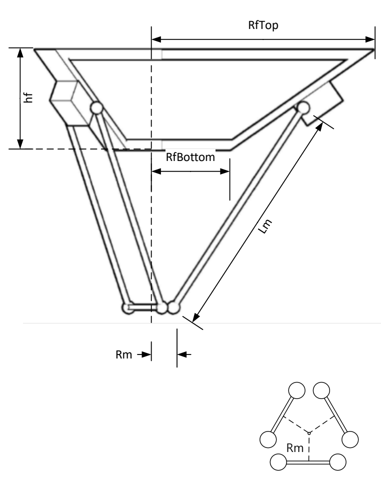
Delta is the delta robot consisting of three rotation axes. If you set the coordinate system type as Delta, you need to set the parameters of the five coordinate systems; Rf / Rm / Lf / Lm



Parameter	Description
Fixed frame radius (Rf)	Length from the center of the fixed frame to the link of the fixed frame(mm)
Link length of fixed frame (Lf)	Link length of fixed frame (mm)
Link length of the moving frame (Lm)	Link length of the moving frame (mm)
Moving frame radius Rm)	Length from the center of the moving frame to the link of the moving frame(mm) (In the left figure, the X, Y coordinates of the fixed frame and the moving frame are the same)

(c) LinearDelta3/3R

LinearDelta is the delta robot consisting of three linear axes. If you set the coordinate system type as LinearDelta, you need to set the parameters of the five coordinate systems; Lm / Hf / RfTop / RfBottom / Rm.



Parameter	Description
Link length of the moving frame (Lm)	Link length of the moving frame (mm)
Height of fixed frame (Hf)	Height of fixed frame (mm)
Radius of fixed frame (Rf) (RfTop)	Radius of fixed frame (mm)
Radius of fixed frame (Rf) (RfBottom)	Radius of fixed frame (mm)
Radius of moving frame (Rm)	Length from the center of the moving frame to the link of the moving frame(mm) (In the left figure, the X, Y coordinates of the fixed frame and the moving frame are the same)

d) T-Gantry/R

For T-Gantry robots, the machine parameter setting does not need additionally.

3) Tools setting

Tool setting item is explained as follows.

Item	Content	Setting range	Initial values
X-axis offset	Set the X axis offset at the end(tool) of robot	Long real(LREAL)	0
Y-axis offset	Set the Y axis offset at the end(tool) of robot	Long real(LREAL)	0
Z-axis offset	Set the Z axis offset at the end(tool) of robot	Long real(LREAL)	0

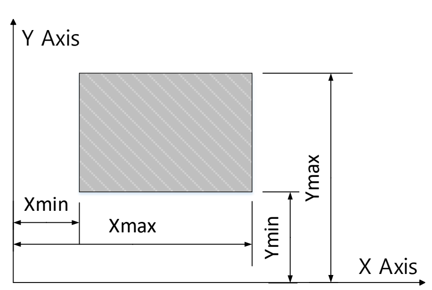
The tool setting parameter enables the position of the tool to be set as an offset so that the end of the tool can be controlled when using a separate tool at the end of the robot.

4) Work space setting

Work space setting item is explained as follows.

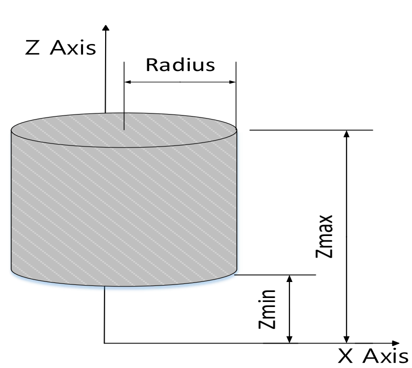
Item	Content	Setting range	Initial values
Work space type	Set the type of work space.	0: Rectangle 1: Cylinder 2: Delta 3: Sector	0: Rectangle
Workspace error check	Set whether or not an error occurs when leaving the workspace.	0: Disable 1:Enable	0: Disable
Work space parameter	Set the parameter in according to work space type.	Long real(LREAL)	0

(a) Rectangle



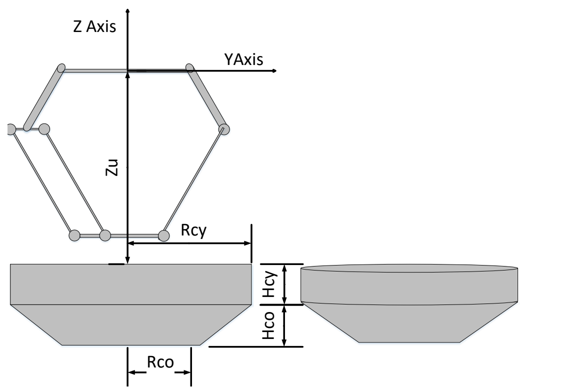
Parameter	Description
Work space parameter1	X min(mm)
Work space parameter2	X max(mm)
Work space parameter3	Y min(mm)
Work space parameter4	Y max(mm)
Work space parameter5	Z min(mm)
Work space parameter6	Z max(mm)
-	-

(b) Cylinder



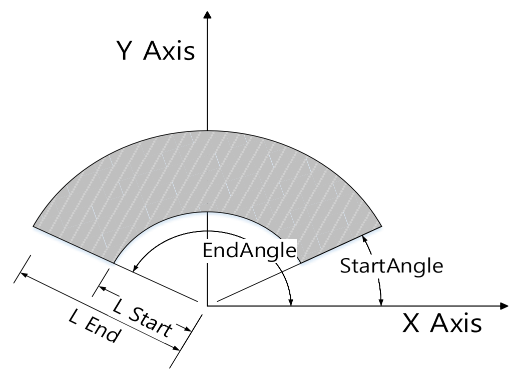
Parameter	Value
Work space parameter1	Radius(mm)
Work space parameter2	Z min(mm)
Work space parameter3	Z max(mm)

(c) Delta



Parameter	Value
Work space parameter1	Zu(mm)
Work space parameter2	Hcy(mm)
Work space parameter3	Hco(mm)
Work space parameter4	Rcy(mm)
Work space parameter5	Rco(mm)

(d) Sector



Parameter	Value
Work space parameter1	L start(mm)
Work space parameter2	L end(mm)
Work space parameter3	Z min(mm)
Work space parameter4	Z max(mm)
Work space parameter5	SartAngle(degree)
Work space parameter6	EndAngle(degree)

5) PCS setting

PCS setting item is explained as follows.

The PCS parameter sets the origin of the workpiece to PCS to facilitate the operation of moving over a specific workpiece in the coordinate system operation. In the PCS coordinate system operation, the coordinate system operation is performed with the set PCS as the origin.

Item	Content	Setting range	Initial values
X-axis move	Set X-axis move distance from MCS origin to PCS origin.	Long real(LREAL)	0 mm
Y-axis move	Set Y-axis move distance from MCS origin to PCS origin.	Long real(LREAL)	0 mm
Z-axis move	Set Z-axis move distance from MCS origin to PCS origin.	Long real(LREAL)	0 mm
X-axis rotation	Set X-axis rotation value of PCS.	-360~360	0 deg
Y-axis rotation	Set Y-axis rotation value of PCS..	-360~360	0 deg
Z-axis rotation	Set Z-axis rotation value of PCS.	-360~360	0 deg

6) JOG operation setting of the coordinate system

JOG operation setting item is explained as follows.

The JOG speed parameters of the coordinate system set the speed during JOG operation.

Item	Content	Setting range	Initial values
XYZ low speed	Set the low-speed JOG operation of the linear axis in the coordinate system operation.	Long real(LREAL) It should be less than or equal to XYZ high speed.	1 mm/sec
ABC low speed	Set the low-speed JOG operation of the rotary axis in the coordinate system operation.	Long real(LREAL) It should be less than or equal to ABC high speed.	1 deg/sec
XYZ high speed	Set the high-speed JOG operation of the linear axis in the coordinate system operation.	Long real(LREAL)	5 mm/sec
ABC high speed	Set the high-speed JOG operation of the rotary axis in the coordinate system operation.	Long real(LREAL)	5 deg/sec

(7) NC Parameter

Set the parameters related to NC control.

- For more details, please refer to Chapter 9 NC control function -9.4 NC parameters.

(8) Cam Data

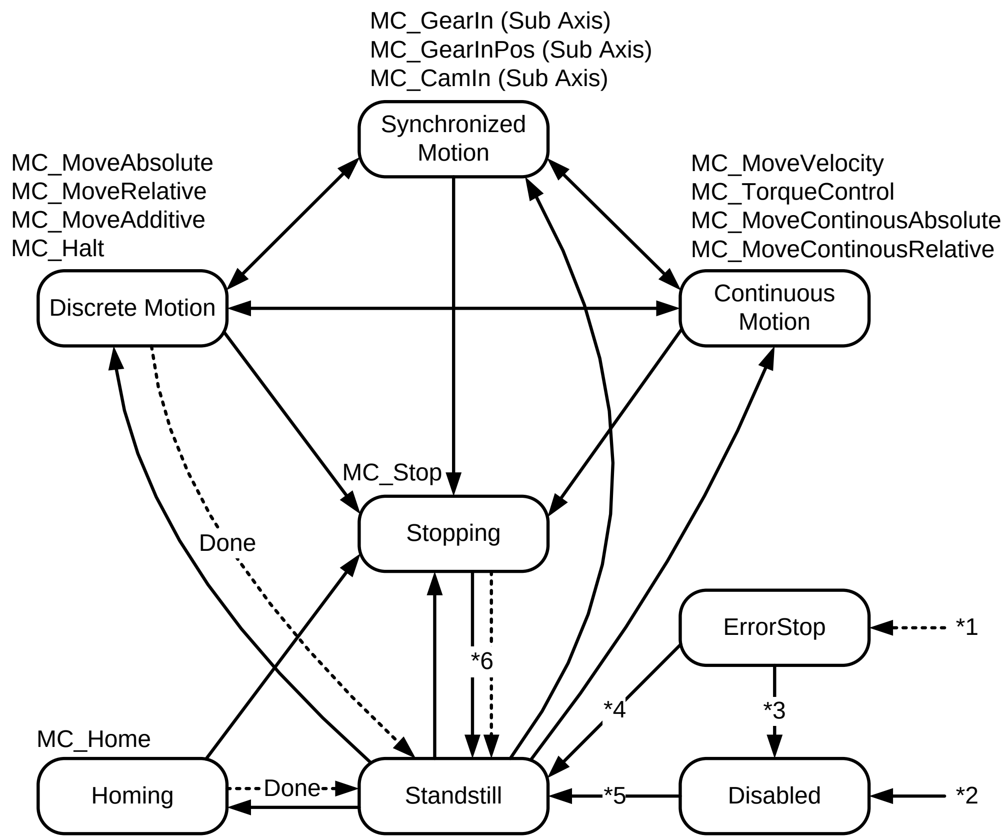
- Set the CAM profile data for CAM operation.
- For more details, refer to Chapter 8, Motion Control Function -8.2.11 (3) CAM Operation in the Synchronous Control Section.

Chapter 6 Motion Function Block

6.1 Common Elements of Motion Function Blocks

6.1.1 The state of axis

Each axis in the motion controller is changed to the relevant state depending on the situation and command. The changing structure of each situation is shown in the figure below.



*1 ErrorStop:
: in case axis error occurs regardless of the current state of axis

*2 Disabled:
: in case MC_Power.Enable input is Off when axis error does not occur

*3 →ErrorStop → Disabled
: in case MC_Reset command has issued when MC_Power.Status output is Off

*4 →ErrorStop → Standstill
: in case MC_Reset command has issued when MC_Power.Status output is on and MC_Power.Enable input is On

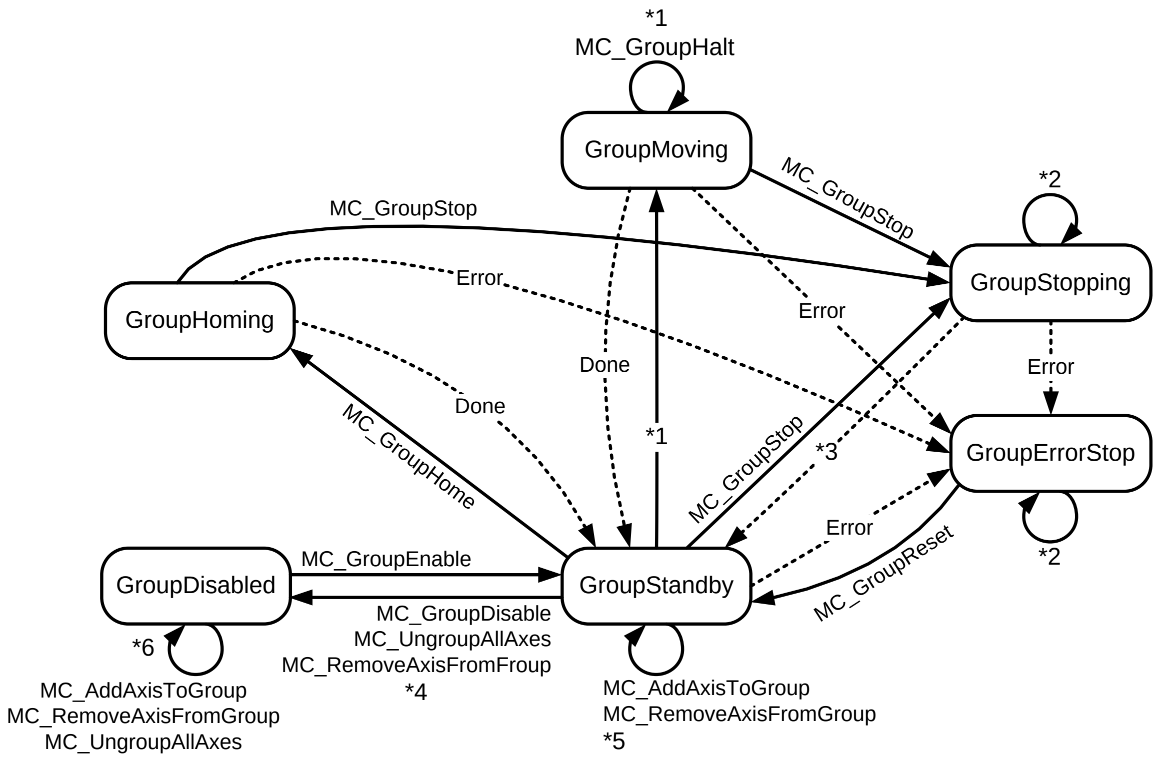
*5 →Disabled → Standstill
: in case of turning On MC_Power.Enable input when MC_Power.Status output is On

*6 →Stopping → Standstill
: in case of turning Off MC_Stop.Execute input when MC_Stop.Done output is On

The state of axis	Description
Disabled (Disabled)	Disabled state indicates the state in which no command is given to a single axis, and no error occurs. In case there is no motion controller at the time of first operation, each axis begins in the disabled state. Afterwards, axis status is changed to standstill state in case servo-on status emerges when Enable input of servo On/Off (MC_Power) motion function block is On. The axis becomes disabled state when Enable input of serve On/Off (MC_Power) motion function block is Off in case of not being in ErrorStop state. In case there is motion function block which is currently being performed, the command is interrupted. (CommandAborted output of the existing motion function blocks is On.)
ErrorStop (ErrorStop)	No matter which state the current axis is in, it is changed to ErrorStop state when axis error occurs, and the axis decelerates to stop. In the state where error occurs, ErrorStop state is maintained even though servo On/Off (MC_Power) motion function block is executed. The motion axis which is in ErrorStop state maintains stationary state, and any command except for error reset is not executed.
Standstill (Standstill)	When the power of axis is activated, there is no error in the axis and any command is not made, the axis state indicates StandStill state.
Homing (Homing)	Homing state indicates the axis is in homing operation.
Stop (Stopping)	In case Stop immediately (MC_Stop) function block is executed, the axis state is changed to stopping state. When the axis is in stopping state, other motion commands cannot be given to the axis until the Stop is completed (until Done output is activated). If Done output is On, and Execute input is On, the state is switched to Standstill status.
Continuous Motion (Continuous Motion)	It indicates state where operation continues until the current axis becomes operation stop status.
Discrete Motion (Discrete Motion)	It indicates reduced operating status with target position.
Synchronized Motion Synchronized Motion	Synchronized motion indicates axis is in synchronized operation.

6.1.2 The State of Group

Each group in motion controller is changed to the relevant state depending on the situation and command. The changing structure of each situation is shown in the figure below



*1 GroupMoving
: In case of performing the motion function block of general group operation

*2 GroupStopping, GroupErrorStop
The relevant motion function block is not performed when different motion function block is performed in GroupStopping or GroupErrorStop state, and when MC_GroupReset function block is performed in GroupErrorStop state, the state of the relevant group is changed to GroupStandby.

*3 → GroupStopping → GroupStandby
When MC_GroupStop.DONE output is On and MC_SroupStop.EXECUTE input is Off

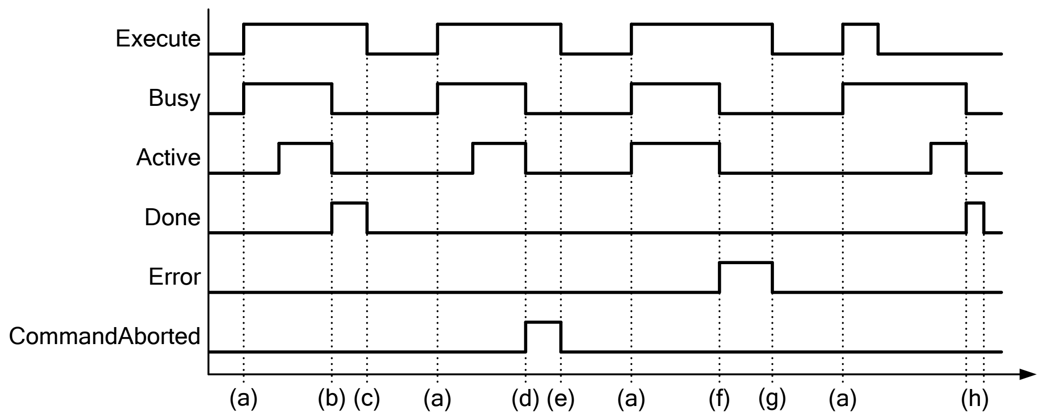
*4 → GroupStandby → GroupDisabled
In case there is no axis belonging to the group when performing the axis remove command (MC_RemoveAxisFromGroup, MC_UnGroupAllAxes)

*5 GroupStandby
In case more than one axis belongs to the group when performing the axis add or remove command in group (MC_AddAxisToGroup, MC_RemoveAxisFromGroup)

*6 GroupDisabled
When performing MC_GroupDisable or MC_UnGroupAllDisable function block, the relevant group is changed to GroupDisabled state regardless of its current state.

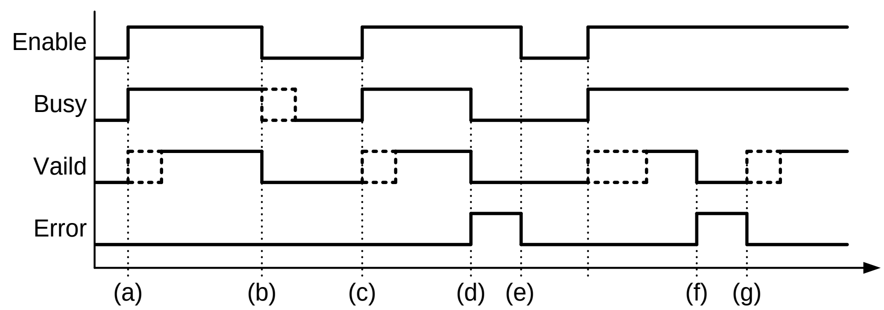
6.1.3 Basic I/O Variable

(1) Edge operation motion function block



Variable	Description
Execute	This is an input to run the relevant function block in Edge operation (Off→On) function block. (Figure a state)
Busy	This is an output to indicate the relevant motion function block is currently running ((= not completed), and it indicates the output of motion function block can be changed. Busy output is On in the rising Edge of Execute input (Figure a state), and it is Off when Done output is On (Figure b state), CommandAborted output is On (Figure d state), or Error output is On (Figure f state).
Active	This indicates the relevant motion function block is actually controlling axis. When running many motion function block to one axis (in case only one motion function block is controlling and other notion function blocks are Buffered), Active output is On in only one motion function block which is controlling, and in motion function blocks which are Buffered, Busy output is On.
Done	This is an output to indicate operation of the relevant motion function block has been successfully completed. If Done output is On, Busy and Active output is Off. (Figure d state) Done output is Off when Execute input is Off (Figure e state), if Execute output was Off when Done output became On, it remains On only during 1 scan (Figure h state).
Error	This is an output to indicate an error occurs while running motion function block. Error output is Off when Execute input is Off (Figure f state). If Execute output was Off when Error output became On, it remains On only during 1 scan (Figure h state).
ErrorID	This outputs error code regarding the relevant error when an error occurs while running motion function block. ErrorID output and elimination time are same with Error output.
CommandAborted	This indicates the relevant motion function block is interrupted by the other motion function block. CommandAborted output is Off when Execute input is Off (Figure g state). If Execute output was Off when Done output became On, it remains On only during one scan.
※ When Execute input is On in Edge operation (Execute input) motion function block, depending on the state of axis, one output in Busy, Done, Error, and CommandAborted output is On. Busy, Done, Error, and CommandAborted output are available to be On one at a time, and if one output in four is On, other three outputs become Off.	

(2) Motion function block for level motion



Variable	Description
Enable	This runs motion function block in the rising Edge (Figure a state), and stops it in the falling Edge (Figure b state). In other words, it is executed when the Enable input is a level action.
Busy	This is an output to indicate the relevant motion function block is currently running ((= not completed), and it indicates the output of motion function block can be changed. Busy output is On in the rising Edge of Enable input (Figure b state), and it remains on while motion function is in operation.
Valid	This is an output to indicate the relevant motion function block is successfully performed and output & motion are valid. Valid output is Off when Enable input is Off (Figure b state).
Error	This is an output to indicate an error occurs while running motion function block. If an error which cannot be automatically restored occurs while motion function block is in operation, Error output is On, Busy & Valid output is Off (Figure d state), and motion function block stops operating. Error output is Off when Enable input is Off (Figure e state). If an error which can be automatically restored occurs while function block is in operation, Error output is On and Valid input is Off (Figure f state). When the error in the relevant motion function block is restored, Error output is Off, and operation is resumed (Figure g state).
ErrorID	This outputs error code regarding the relevant error when an error occurs while running motion function block. ErrorID output and elimination time are same with Error output.
※Valid and Error outputs are not On at the same time.	

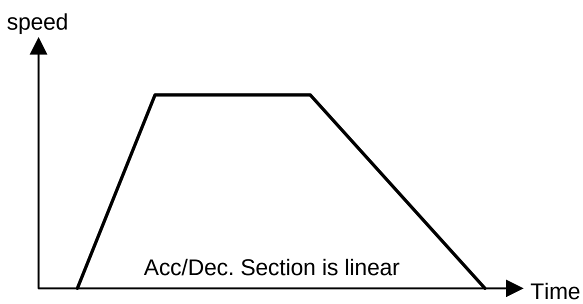
Notes

1. Axis input[note1)]

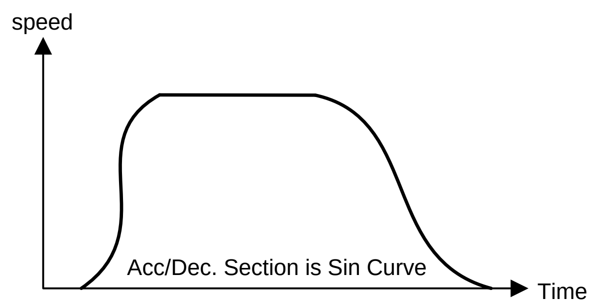
Each motion function block can be specified by Axis input to the axis which is subject to the relevant command. Motion controller can control 1~32 real/virtual axes and 33~36 virtual axes, and 1001~1002 encoders can be used as main axis depending on motion function block. Therefore, values of 1~32, 33~36, and 1001~1002 can be input in Axis input depending on motion function block. When it is out of the range which is available to set in each motion function block, "error 0x0006" occurs.

2. Jerk

Jerk sets the amount of change of acceleration/deceleration. If Jerk is set to a non-zero value, the speed profile becomes S-shaped, which can reduce the impact of the machine during acceleration / deceleration. If Jerk value is set larger, acceleration / deceleration is performed close to the straight line. If Jerk value is set to 0, acceleration / deceleration operation speed profile becomes linear.



< Jerk = 0 >



< Jerk Acceleration / Deceleration >

Note 1) Explain the range to set axis input variables on the basis of XMC-E32A.

6.1.4 BufferMode input

This is an input which can specify whether to wait until the existing command is completed or to cancel the existing motion function block and execute the command in case the axis is already running other motion function block when running motion function block in a certain axis. The number between 0-5 can be specified, and if it is out of the range, "error 0x101A" occurs in the axis command and "error 0x201A" occurs in the axis group command. The values which are available to be set in BufferMode are as below.

No.	Buffer mode	Description
0	**Aborting**	Execute the command immediately. The existing command in operation is interrupted.
1	**Buffered**	It executes commands after the completion of the existing command in operation.
2	**BlendingLow**	It conducts a combination operation that helps blend into side with lower velocity by comparing the velocity of the existing command and the command to make.
3	**BlendingPrevious**	It conducts a combination operation that makes the combination with velocity of the existing commands.
4	**BlendingNext**	It conducts a combination operation that makes the combination with velocity of commands to make.
5	**BlendingHigh**	It conducts a combination operation that helps blend into side with priority velocity by comparing the velocity of the existing command and the command to make.

Notes

The maximum number of commands that can wait for their execution in buffers on axis control is 100. An error (error code: 0x1022) occurs when executing more than 100 commands in a buffer mode.

6.1.5 Changes in Parameters during Execution of Motion Function Block

The parameter of the relevant command can be changed at the time motion function block is running, and the detailed operations are as below.

- When executing Edge operation motion function block in the Off state of ContinuousUpdate input (turn On the Execute input), the relevant motion function block is operated by application of the parameter at the time when Execute input was On (rising Edge). In this case, the change of the parameter input value in the middle of execution of motion function block does not affect operation.
  When wanting to change the parameter while the relevant motion function block is in operation, change the parameter and turn On Execute input again.
- When executing Edge operation motion function block in the On state of ContinuousUpdate input (turn On the Execute input), the parameter of the time when Execute input was On (rising Edge) is applied at first.
  When changing the parameter while ContinuousUpdate input is On, the relevant motion function block operates reflecting the every change in parameter.
  But, if you change the parameter at the completion or after the stop of the operation of the relevant motion function block (Busy output is Off), the change is not reflected any more. (Parameter changing operation using ContinuousUpdate does not rerun the motion function block which is completed or interrupted, In other words, ContinuousUpdate operation is applied only to the motion function block which is currently running.)
- For a function block without ContinuousUpdate input, the changed parameters can be applied by re-executing the function block (Execute input is On) before the command is completed.
- As for level operation motion function block, it is operated by the application of the parameter at the time when Enable input was On (rising Edge), and continuous change of parameter is available while Enable input is On.

6.1.6 Group Operation Route Change Settings

When the axis group of the current motion controller is executing a command, other command can be issued to the relevant axis group. At this point, the path, which the next command will achieve, can specify how the existing command will be connected to the existing path. The parameter of connection track is specified in TransitionParameter input.

No.	Transition Mode	Description
0	TMNone	Do not generate a connection track.
3	TMCornerDistance	Generate a connection track which specifies the corner distance of a connection track and draws circular arcs at the specified corner distance.

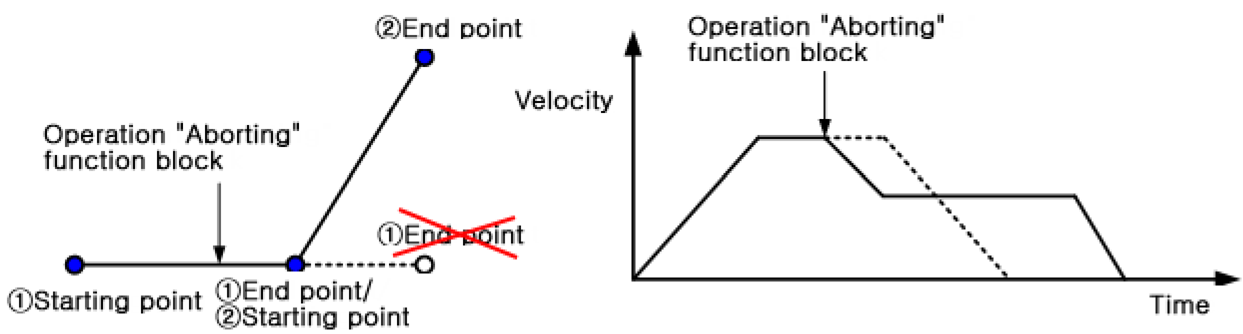
Notes

The maximum number of buffers that can wait for execution on axis group control is 100. An error (error code: 0x2022) occurs when executing more than 100 commands in a buffer mode.

- **TransitionMode "TMNone"**

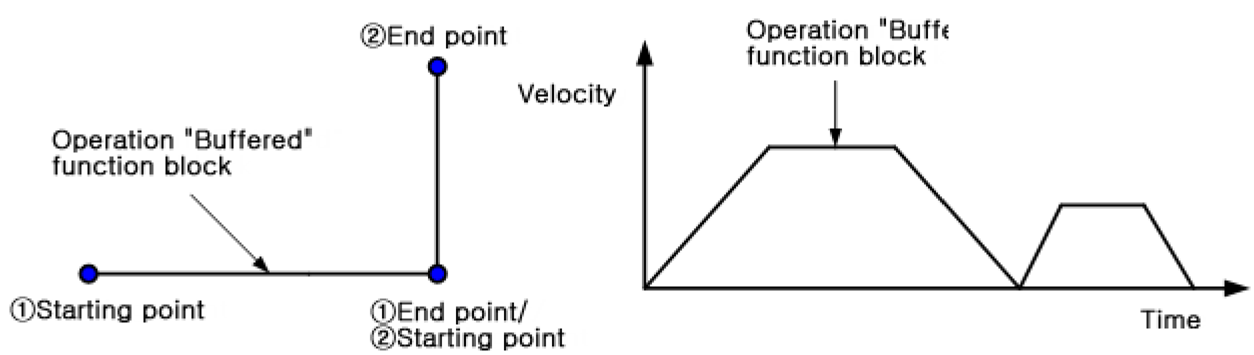
Do not generate a connection track.

The Figure below shows the case when running BufferMode of motion function block in the setting of 'Aborting'. The Figure in the left shows that motion function block ② is executed in the setting of 'Aborting' while motion function block ① is running. Motion function block ① is forced to be terminated at 'end point ① / starting point ②' without reaching 'end point ①'. The Figure in the right shows that the next motion function block is executed at the moment of the execution of 'Aborting' function block.



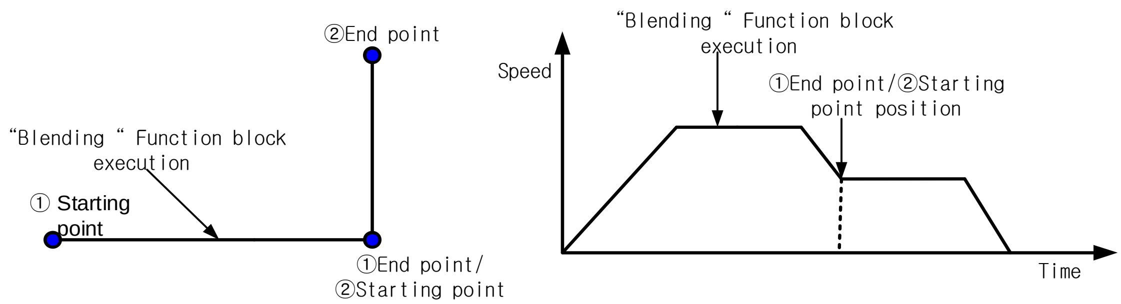
<In case BufferMode is specified as "Aborting">

The Figure below shows that the case when running BufferMode of motion function block in the setting of 'Buffered'. The Figure in the left shows that motion function block ② is executed in the setting of 'Buffered' while motion function block ① is running. Motion function block ② is executed after motion function block ① has reached target position. The Figure in the right shows that when 'Buffered' function block is executed, the next motion function block is executed after it reaches original target position.



<In case BufferMode is specified as "BufferMode">

The Figure below shows the case when running BufferMode of motion function block in the setting of ‘BlendingXXXX’. The Figure in the left shows that motion function block ② is executed in the setting of ‘BlendingXXXX’ while motion function block ① is running. Accelerates/decelerates from the target position of motion function block ① to the speed set in motion function block ② (Low, Previous, Next, High) and performs continuous operation. If you look at the figure on the right, when the ‘BlendingXXXX’ function block is executed, the next motion function block is executed at the speed (Low, Previous, Next, High) set in ‘BlendingXXXX’ after reaching the original target position.



<In case BufferMode is specified as “BlendingXXXX”>

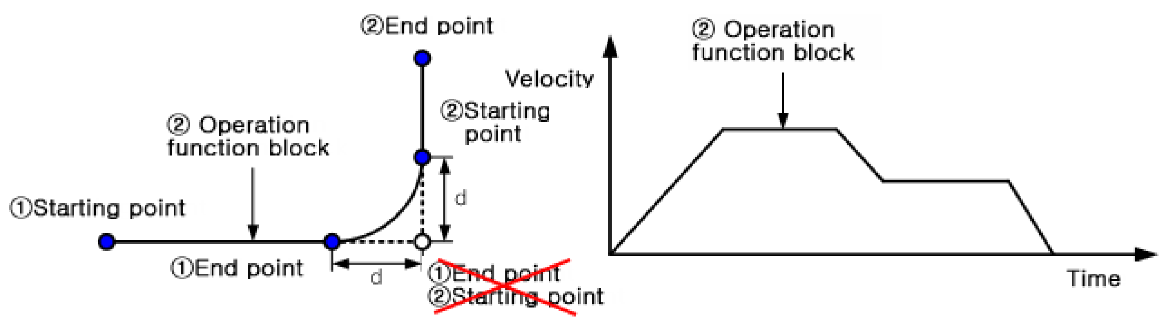
■ **TransitionMode “TMCornerDistance”**

The radius of a connection track is specified and the connection track which draws a circle having specified radius is output. This mode is operated only when BufferMode is “BlendingXXXX”, and it is operated in “TMNone” when BufferMode is “Aborting” or “Buffered”.

When drawing a connection track, the maximum speed of the path complies with the specified speed in BufferMode, and the length of radius complies with the value specified in TransitionParameter.

The Figure below shows the generation of a connection track which draws radius circle in two linear interpolation commands. The Figure in the left shows that motion function block ② is executed in the setting of “TMcornerDistance” while motion function block ① is running. The original target position of motion function block ① was end point ① / starting point ②, but straight-line motion is stopped and circular motion is started at the point ahead as far as radius 'd' (end point ①). Circular operation starts at end point ① and finishes at starting point ②, and executes motion function block ②.

The Figure in the right shows that the speed does not stop in the middle of two function blocks and continues.



<In case BufferMode is specified as “BlendingLow” and TransitionMode is specified as “TMCornerDistance”>

6.1.7 Motion Function Block Errors

Errors occurring in ErrorID variable of motion function block are as follows.

STAT	Content	Description
0x0000	Normal	In case motion function block is normally executed, “O” is displayed on ErrorID.
0x0005	The current motion controller does not support the motion function block.	The motion function block is not executed in the version of current motion controller. Check the version in which the motion function block can be executed.
0x0006	Axis number of motion function block (Axis input) or encoder number (Encoder input) exceeded allowable range.	Set axis and encoder numbers with a range by product.
0x0007	Axis group number of motion function block (AxesGroup input) exceeded allowable range.	Set axis group number to a value between 1 and 16.
0x0008	NC channel of function block exceeded allowable range.	Check the range of NC channel, and set again.
0x0009	Slave number of function block (Slave input) exceeded allowable range.	Check the range of slave number, and set again.
0x000B	Input of function block exceeded allowable range.	Check the input range of function block, and set again.
0x000C	Array input of function block exceeded allowable range.	Check the array input size of function block, and set again.
0x0012	Internal execution error of motion function block occurred during the execution of the motion function block.	Check the version of XG5000 and XMC-E32A. Please check the supported version of XG5000 and controller.
0x0013	Motion response error occurred during the execution of motion function block.	Check the version of XG5000 and XMC-E32A. Please check the supported version of XG5000 and controller.
0x0014	CAM ID (CamTableID input) of function block exceeded allowable range.	Check the CAM ID range of function block, and set again.
0x0E00 : 0x0FFF	It indicates a common error of the motion controller. For more details, refer to ‘error information and measures in APPENDIX 2’.	
0x1000 : 0x1FFF	It indicates error that occurs in relation to axis control of motion controller. For more details, refer to ‘error information and measures in APPENDIX 2’.	
0x2000 : 0x2FFF	It indicates error that occurs in relation to axis control of motion controller. For more details, refer to ‘error information and measures in APPENDIX 2’.	
0x3000 : 0x3FFF	It indicates error that occurs in relation to NC control of motion controller. For more details, refer to ‘error information and measures in APPENDIX 2‘.	

6.2 Motion Function Block

NO.	Name	Detailed description	Operation condition	Module O/S	XG5000
Single-axis command					
1	MC_Power	Servo On/Off	Level	V1.0	V4.20
2	MC_Home	Return to Origin Point	Edge	V1.0	V4.20
3	MC_Stop	Prompt stop	Edge	V1.0	V4.20
4	MC_Halt	Stop	Edge	V1.0	V4.20
5	MC_MoveAbsolute	Absolute positioning operation	Edge	V1.0	V4.20
6	MC_MoveRelative	Relative positioning operation	Edge	V1.0	V4.20
7	MC_MoveAdditive	Additive positioning operation	Edge	V1.0	V4.20
8	MC_MoveVelocity	Specified velocity operation	Edge	V1.0	V4.20
9	MC_MoveContinuousAbsolute	Specified velocity operation after Absolute position operation	Edge	V1.0	V4.20
10	MC_MoveContinuousRelative	Specified velocity operation after Relative position operation	Edge	V1.0	V4.20
11	MC_TorqueControl	Torque control	Edge	V1.0	V4.20
12	MC_SetPosition	Setting the current position	Edge	V1.0	V4.20
13	MC_SetOverride	Velocity/Acceleration override	Level	V1.0	V4.20
14	MC_ReadParameter	Parameter Read	Level	V1.0	V4.20
15	MC_WriteParameter	Parameter write	Edge	V1.0	V4.20
16	MC_Reset	Reset axis error	Edge	V1.0	V4.20
17	MC_TouchProbe	Touch probe	Edge	V1.0	V4.20
18	MC_AbortTrigger	Abort trigger events	Edge	V1.0	V4.20
19	MC_MoveSuperImposed	SuperImposed Operation	Edge	V1.0	V4.20
20	MC_HaltSuperImposed	SuperImposed operation halt	Edge	V1.0	V4.20
21	LS_Home	Expansion homing	Edge	V2.0	V4.70
22	MC_Reset2	Reset axis error 2	Edge	V2.0	V4.70
23	MC_ReadMotionInfo	Read Motion Information	Level	V2.0	V4.70
24	MC_ReadActual	Read Current Position	Level	V2.0	V4.70
25	MC_ReadActual	Read Current Velocity	Level	V2.0	V4.70
26	MC_ReadActual	Read Current Torque	Level	V2.0	V4.70
27	MC_ReadCommanded	R	Level	V2.0	V4.70
28	MC_ReadCommanded	Read Command Speed	Level	V2.0	V4.70
29	MC_ReadCommanded	Read Command Torque	Level	V2.0	V4.70
Multi-axis command					
30	MC_CamIn	Cam Operation	Edge	V1.0	V4.20
31	MC_CamOut	Camming stop	Edge	V1.0	V4.20
32	MC_GearIn	Electrical gearing run	Edge	V1.0	V4.20
33	MC_GearOut	Electrical gearing disengage	Edge	V1.0	V4.20
34	MC_GearInPos	Electrical gearing by specifying the position	Edge	V1.0	V4.20
35	MC_Phasing	Phase compensation	Edge	V1.0	V4.20
36	MC_PowerAll	Total axis servo On/Off	Level	V2.0	V4.70
37	MC_HomeAll	Total axis homing	Edge	V2.0	V4.70

NO.	Name	Detailed description	Operation condition	Module O/S	XG5000
38	LS_HomeAll	Total axis expansion homing	Edge	V2.0	V4.70
39	MC_StopAll	Total axis prompt Stop	Edge	V2.0	V4.70
40	MC_HaltAll	Total axis Stop	Edge	V2.0	V4.70
41	MC_Reset2All	Reset all axis error 2	Edge	V2.0	V4.70
42	MC_SetPositionAll	Setting all axis current position	Edge	V2.0	V4.70
43	MC_GearInEx	Expanded electronic gear operation	Edge	V2.0	V4.70
44	MC_GearInPosEx	Expanded position specified electronic gear operation	Edge	V2.0	V4.70
Group command					
45	MC_AddAxisToGroup	Adds one axis to the group	Edge	V1.0	V4.20
46	MC_RemoveAxisFromGroup	Removes one axis from the group	Edge	V1.0	V4.20
47	MC_UngroupAllAxes	Removes all axes from the group	Edge	V1.0	V4.20
48	MC_GroupEnable	Group Enable	Edge	V1.0	V4.20
49	MC_GroupDisable	Group Disable	Edge	V1.0	V4.20
50	MC_GroupHome	Performs the search home of all axes in the group	Edge	V1.0	V4.20
51	MC_GroupSetPosition	Sets the position of all axes in the group without moving	Edge	V1.0	V4.20
52	MC_GroupStop	Stop the group immediately	Edge	V1.0	V4.20
53	MC_GroupHalt	Stop the group	Edge	V1.0	V4.20
54	MC_GroupReset	Reset the group error	Edge	V1.0	V4.20
55	MC_MoveLinearAbsolute	Absolute positioning linear interpolation operation	Edge	V1.0	V4.20
56	MC_MoveLinearRelative	Relative positioning linear interpolation operation	Edge	V1.0	V4.20
57	MC_MoveCircularAbsolute	Absolute position circular interpolation operation	Edge	V1.0	V4.20
58	MC_MoveCircularRelative	Relative position circular interpolation operation	Edge	V1.0	V4.20
59	MC_GroupPower	Group Servo On/Off	Level	V2.0	V4.70
60	LS_ReadGroupParameter	Read axis group parameter	Level	V2.0	V4.70
61	LS_WriteGroupParameter	Write axis group parameter	Edge	V2.0	V4.70
62	MC_GroupSetOverride	Axis group velocity/acceleration override	Level	V2.0	V4.70
LS command					
63	LS_Connect	Connect servo drives	Edge	V1.0	V4.20
64	LS_Disconnect	Disconnect servo drives	Edge	V1.0	V4.20
65	LS_ReadSDO	Read slave SDO data	Edge	V1.0	V4.20
66	LS_WriteSDO	Write slave SDO data	Edge	V1.0	V4.20
67	LS_SaveSDO	Save slave SDO data	Edge	V1.0	V4.20
68	LS_ReadSDO2	Read slave SDO data(timeout)	Edge	V2.0	V4.60
69	LS_WriteSDO2	Write slave SDO data(timeout)	Edge	V2.0	V4.60
70	LS_EncoderPreset	Setting Encoder current position	Edge	V1.0	V4.20
71	LS_Jog	Jog Operation	Level	V1.0	V4.20
72	LS_ReadCamData	Read Cam data	Edge	V1.0	V4.20
73	LS_WriteCamData	Write CAM data	Edge	V1.0	V4.20

NO.	Name	Detailed description	Operation condition	Module O/S	XG5000
74	LS_WriteCamBlockProperty	CAM block property write	Edge	V2.0	V4.60
75	LS_ReadCamBlockProperty	CAM block property read	Edge	V2.0	V4.60
76	LS_WriteCamBlockNode	CAM block node write	Edge	V2.0	V4.60
77	LS_ReadCamBlockNode	CAM block node read	Edge	V2.0	V4.60
78	LS_GenerateCamData	CAM data generation	Edge	V2.0	V4.60
79	LS_ReadEsc	Read from ESC	Edge	V1.0	V4.20
80	LS_WriteEsc	Write to ESC	Edge	V1.0	V4.20
81	LS_CamSkip	Skip CAM	Edge	V1.0	V4.20
82	LS_VarCamIn	Variable CAM operation	Edge	V1.0	V4.20
83	LS_VarGearIn	Variable gear operation	Edge	V1.0	V4.20
84	LS_VarGearInPos	Variable positioning gear operation	Edge	V1.0	V4.20
85	LS_ReadCamTableSlavePos	Read the slave location of the CAM table	Edge	V1.0	V4.20
86	LS_InverterWriteVel	Write inverter speed	Edge	V1.0	V4.20
87	LS_InverterReadVel	Read inverter speed	Level	V1.0	V4.20
88	LS_InverterControl	Write inverter control word	Edge	V1.0	V4.20
89	LS_InverterStatus1	Read inverter status 1	Level	V1.0	V4.20
90	LS_InverterStatus2	Read inverter status 2	Level	V1.0	V4.20
91	LS_SyncMoveVelocity	Speed control operation (csv mode)	Edge	V1.0	V4.20
92	LS_ReadCamTableMasterPos	Read CAM table master position	Edge	V1.1	V4.23
93	LS_OnOffCam	On/off CAM operation	Edge	V1.1	V4.23
94	LS_RotaryKnifeCamGen	RotaryKnife cam profile generation	Edge	V1.1	V4.23
95	LS_CrossSealCamGen	Cross sealer cam profile generation	Edge	V1.2	V4.25
96	LS_OnOffCamEx	Expanded on/off CAM operation	Edge	V1.4	V4.28
97	LS_VarGearInEx	Expansion Variable gear operation	Edge	V2.0	V4.60
98	LS_VarGearInPosEx	Expansion Variable positioning gear operation	Edge	V2.0	V4.60
Coordinate system command					
99	MC_SetKinTransform	Machine Information Setting	Edge	V1.0	V4.20
100	MC_SetCartesianTransform	PCS Setting	Edge	V1.0	V4.20
101	LS_SetWorkSpace	Work space setting	Edge	V1.0	V4.20
102	LS_MoveLinearAbsolute	Linear Interpolation Operation for Absolute Position of Coordinate System	Edge	V1.0	V4.31
103	LS_MoveLinearRelative	linear interpolation operation for relative position of coordinate system	Edge	V1.0	V4.31
104	LS_MoveLinearTimeAbsolute	linear interpolation operation for absolute position of coordinate system	Edge	V1.0	V4.20
105	LS_MoveLinearTimeRelative	linear interpolation operation for relative position of coordinate system	Edge	V1.0	V4.20
106	MC_MoveCircularAbsolute2D	Circular interpolation operation for absolute position of coordinate system	Edge	V1.0	V4.20
107	MC_MoveCircularRelative2D	Circular interpolation operation for relative position of coordinate system	Edge	V1.0	V4.20

NO.	Name	Detailed description	Operation condition	Module O/S	XG5000
108	MC_TrackConveyorBelt	Synchronization setting of the conveyor belt	Edge	V1.0	V4.20
109	MC_TrackRotaryTable	Synchronization setting of the rotary table	Edge	V1.0	V4.20
110	LS_RobotJog	JOG operation of the coordinate system	Level	V1.0	V4.20
111	LS_SetMovePath	Set path operation data	Edge	V1.0	V4.20
112	LS_ResetMovePath	Delete path operation data	Edge	V1.0	V4.20
113	LS_GetMovePath	Read path operation data	Edge	V1.0	V4.20
114	LS_RunMovePath	Perform path operation	Edge	V1.0	V4.20
115	MC_MoverCircularAbsolute3D	Arc Interpolation Operation for Absolute Position of Coordinate System	Edge	V2.0	V4.60
116	MC_MoveCircuularRelavive3D	Arc Interpolation Operation for Relative Position of Coordinate System	Edge	V2.0	V4.60
117	LS_ForwardKIn	Forward Kinematic operation	Edge	V2.0	V4.60
118	LS_InverseKin	Reverse Kinematic operation	Edge	V2.0	V4.60
NC control command					
119	NC_LoadProgram	Specify NC program	Edge	V1.0	V4.20
120	NC_BlockControl	Specify Single/Optional block operation	Level	V1.0	V4.20
121	NC_Reset	NC reset	Edge	V1.0	V4.20
122	NC_Emergency	Emergency stop	Level	V1.0	V4.20
123	NC_CycleStart	Start automatic operation	Edge	V1.0	V4.20
124	NC_FeedHold	Feed Hold	Level	V1.0	V4.20
125	NC_Home	Homing	Edge	V1.0	V4.20
126	NC_RapidTraverseOverride	Rapid traverse override	Level	V1.0	V4.20
127	NC_CuttingFeedOverride	Cutting feed override	Level	V1.0	V4.20
128	NC_SpindleOverride	Spindle override	Level	V1.0	V4.20
129	NC_McodeComplete	M Code operation completed	Edge	V1.0	V4.20
130	NC_ScodeComplete	S Code operation completed	Edge	V1.0	V4.20
131	NC_TcodeComplete	T Code operation completed	Edge	V1.0	V4.20
132	NC_ReadParameter	Read NC parameters	Level	V1.0	V4.20
133	NC_WriteParameter	Write NC parameters	Edge	V1.0	V4.20
134	NC_RetraceMove	Reverse operation	Level	V1.3	V4.28
135	NC_BlockSkip	Block skip	Level	V1.3	V4.28
136	NC_DryRun	Dry run operation	Level	V1.3	V4.28
137	NC_ToolMode	Tool retract/recover operation	Edge	V1.3	V4.28
138	NC_ReadToolMode	Read tool retract/recover modes	Level	V1.3	V4.28
139	NC_MirrorImage	Mirror image	Level	V1.3	V4.28
140	NC_SpindleControl	Spindle operation control	Level	V1.3	V4.28
141	NC_BlockOptionalSkip	NC optional block skip	Edge	V1.3	V4.28
142	NC_ManualToolComp	Manual measurement of NC compensation value	Edge	V1.3	V4.28
143	NC_ChgSpindleGear	NC spindle gear change	Edge	V1.3	V4.28

NO.	Name	Detailed description	Operation condition	Module O/S	XG5000
144	NC_ExpoerProgram	NC program Export	Edge	V2.0	V4.70
145	NC_ImportProgram	NC program Import	Edge	V2.0	V4.70
File command					
146	FILE_OPEN	Open files in SD memory cards	Edge	V1.4	V4.28
147	FILE_CLOSE	Close files in SD memory cards	Edge	V1.4	V4.28
148	FILE_WRITE	Write files in SD memory cards	Edge	V1.4	V4.28
149	FILE_READ	Read files in SD memory cards	Edge	V1.4	V4.28
150	FILE_SEEK	Move positions to access in SD memory cards	Edge	V1.4	V4.28
Communication command					
151	SOCKET_TCPLISTEN	External TCP Server connection ready	Edge	V2.0	V4.70
152	SOCKET_TCPACCEPT	External TCP Server connection accept	Edge	V2.0	V4.70
153	SOCKET_TCPCONNECT	TCP Client connection	Edge	V2.0	V4.70
154	SOCKET_TCPRECV	TCP socket data receive	Edge	V2.0	V4.70
155	SOCKET_TCPSEND	TCP socket data send	Edge	V2.0	V4.70
156	SOCKET_UDPCREATE	UDP socket create request	Edge	V2.0	V4.70
157	SOCKET_UDPRECV	UDP socket data receive	Edge	V2.0	V4.70
158	SOCKET_UDPSEND	UDP socket data send	Edge	V2.0	V4.70
159	SOCKET_SVCINFO	Information for each channel of socket service	Edge	V2.0	V4.70
160	SOCKET_BUFCLEAR	Socket Service Buffer Initialization	Edge	V2.0	V4.70
161	SOCKET_CLOSE	Socket service channel close	Edge	V2.0	V4.70
162	L_NET_INFO	Local Ethernet information read	Edge	V2.0	V4.70
163	SET_IP	Local Ethernet setting	Edge	V2.0	V4.70
Other command					
164	PID	PID calculation	Level	V1.0	V4.20
165	LINAC	LINAC	Level	V1.0	V4.20
166	SLINAC	SLINAC	Level	V1.0	V4.20

6.2 Motion Function Block

6.2.1 Setting Range by Product

Important

(1) The motion function block names of XMC-E32A, XMC-E16A, XMC-E08A and XMC-E32C are all identical. This 'Chapter 06 Function Block' basically describes XMC-E32A.

(2) The motion function block of XMC-E32A and XMC-E32C has an identical setting range of input variables and the motion function block of XMC-E32A, XMC-E16A and XMC-E08A has a different setting range of input variables that set IDs of axes, slaves and CAM tables.

(a) Setting range of input variables that set axes

Input variable	XMC-E32A/XMC-E32C	XMC-E16A	XMC-E08A
Axis	1~32: Real/Virtual Axes, 33~36: Virtual Axes	1~16: Real/Virtual Axes, 17~18: Virtual Axes	1~8: Real/Virtual Axes, 9: Virtual Axes
Master	1~32: Real/Virtual Axes, 33~36: Virtual Axes, 1001~ 2: Encoder	1~16: Real/Virtual Axes, 17~18: Virtual Axes, 1001~ 1002: Encoder	1~8: Real/Virtual Axes, 9: Virtual Axes, 1001~ 1002: Encoder
Slave*Note 1)	1~32: Real/Virtual Axes, 33~36: Virtual Axes	1~16: Real/Virtual Axes, 17~18: Virtual Axes	1~8: Real/Virtual Axes, 9: Virtual Axes

(b) Setting range of input variables that set slaves

Input variable	XMC-E32A/XMC-E32C	XMC-E16A	XMC-E08A
Slave*Note 2)	1~ 64: Slave	1~ 32: Slave	1~ 16: Slave

(c) Setting range of input variables of the CAM table ID

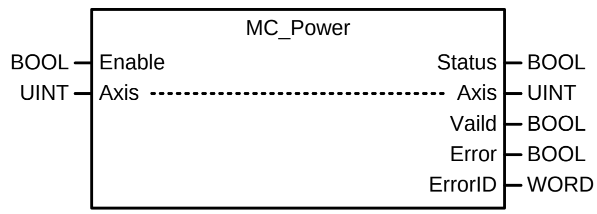
Input variable	XMC-E32A/XMC-E32C	XMC-E16A	XMC-E08A
CamTableID	1~32: CAM table number	1~16: CAM table number	1~8: CAM table number

Note 1) An input variable of the function block (MC_CAMIN, etc.) that sets operation of axes.

Note 2) An input variable of the function blocks (LS_ReadSDO, LS_WriteSDO, LS_SaveSDO) that sets operation of slaves.

6.3 Single-Axis Motion Function Block

6.3.1 Servo On/Off (MC_Power)

Motion Function Block		
		
Input-Output		
UINT	Axis	Specify the axis to be commanded (See 6.2.1 Setting Range by Product)
Input		
BOOL	Enable	Servo motor of the relevant axis is servo On while input is activated.
Output		
BOOL	Status	Indicate the power permission status of the relevant axis.
BOOL	Valid	Indicate the validity of motion function block output. (same with Status output here)
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

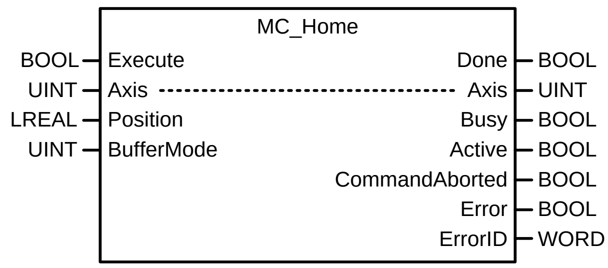
(1) This motion function block is to give servo On/Off command to the relevant axis.

(2) When enable input changes from Off to On, the Servo On command is given to the relevant axis. When it changes from On to Off, the Servo Off command is given to it.

(3) If servo On command is executed when the axis is in 'Disable' state, the axis state is 'StandStill', and failure in servo On brings 'ErrorStop' state.

6.3.2 Perform the search home (MC_Home)

Motion Function Block



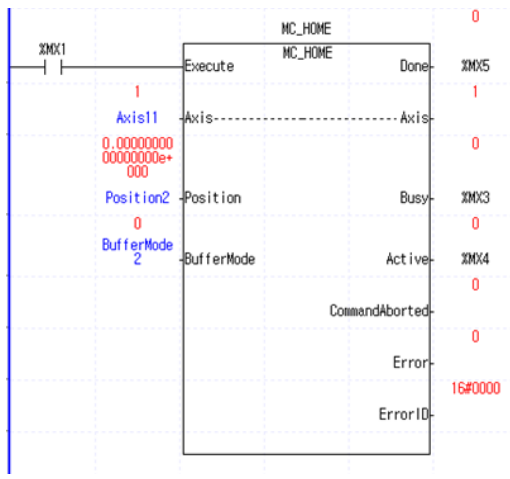
Input-Output		
UINT	Axis	Specify the axis to be commanded (See 6.2.1 Setting Range by Product)
Input		
BOOL	Execute	Start the homing operation in rising Edge.
LREAL	Position	Specify the absolute position of axis when reference signal is detected. (When the command position value is converted to a pulse, it is set within the range of -2147483648 to 2147483647.)
UINT	BufferMode	Specify the sequential operation setting of motion function block. (Refer to 6.1.4.BufferMode)
Output		
BOOL	Done	Indicate the completion state of motion function block.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that the current motion function block is controlling the relevant axis.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted by other command.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block is to give a homing command to the relevant axis.
(2) Homing method is operated as specified in the operation parameter of the relevant axis in advance.
(3) As for Position input, absolute position of axis is specified when Reference Signal is detected or homing is completed.
(4) While this motion function block is running, the axis is 'Homing' state, and when the command is completed, it is switched to 'Standstill'.
(5) If the function block is re-executed (Execute input is On) before the instruction is terminated, the changed parameters are applied. Only Position input can be updated.

(6) Example Program

This example shows execution of MC_HOME command when the current command position is 100,000.

(a) Function block setting

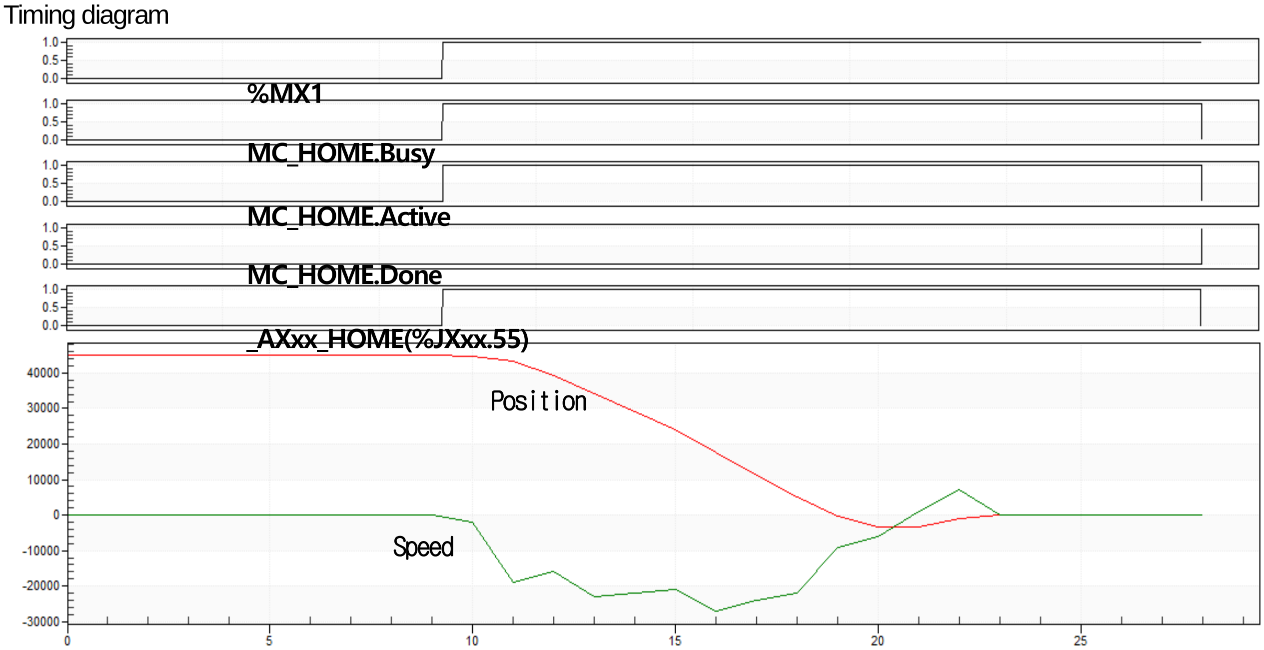


(b) Parameter Settings

- Set the Homing method in SDO parameters to 33.

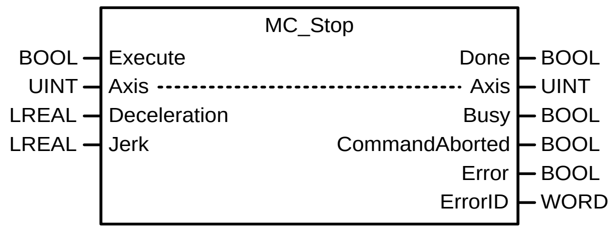
Index	Name	Unit	Current Value	Initial Value	Access
607A	Target Position	UU	0	0	rw
607C	Home Offset	UU	0	0	rw
607D:00	Software Position Limit	-	2	2	rw
607F	Maximum Profile Velocity	UU/s	2147483647	2147483647	rw
6080	Maximum Motor Speed	rpm	0	0	ro
6081	Profile Velocity	UU/s	200000	200000	rw
6083	Profile Acceleration	UU/s^2	200000	200000	rw
6084	Profile Deceleration	UU/s^2	200000	200000	rw
6085	Quick Stop Deceleration	UU/s^2	2000	2000	rw
6087	Torque Slope	0.1%/s	1000	1000	rw
6091:00	Gear Ratio	-	2	2	rw
6098	Homing Method	-	33	34	rw
6099:00	Homing Speeds	-	2	2	rw
609A	Homing Acceleration	UU/s^2	200000	200000	rw
60B0	Positon Offset	UU	0	0	rw
60B1	Velocity Offset	UU/s	0	0	rw
60B2	Torque Offset	0.1%	0	0	rw

(c) Timing diagram



6.3.3 Stop immediately (MC_Stop)

Motion Function Block



Input-Output		
UINT	Axis	Specify the axis to be commanded (See 6.2.1 Setting Range by Product)
Input		
BOOL	Execute	Give immediate stop command to the relevant axis in the rising Edge.
LREAL	Deceleration	Specify the deceleration in time of stop. [u/s^2]
LREAL	Jerk	Specify the change rate of acceleration/deceleration. [u/s^3]
Output		
BOOL	Done	Indicate that the speed of the relevant axis reaches 0.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted while it is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block is to give a emergency stop command to the relevant axis.

(2) When executing immediate stop (MC_Stop) motion function block, the existing motion function block being executed in the relevant axis is stopped, and the axis state changed to 'Stopping'. When the relevant axis is in 'Stopping' state, other motion function block cannot be executed in the relevant axis until the stopping is completed (until the Done output is activated).

(3) CommandAborted output indicates that the current motion function block is interrupted while it was executed. Other motion function block cannot interrupt immediate stop (MC_Stop) motion function block while immediate stop (MC_Stop) motion function block is running, therefore, CommandAborted output is On in general when the power of servo is blocked or servo Off command is executed.

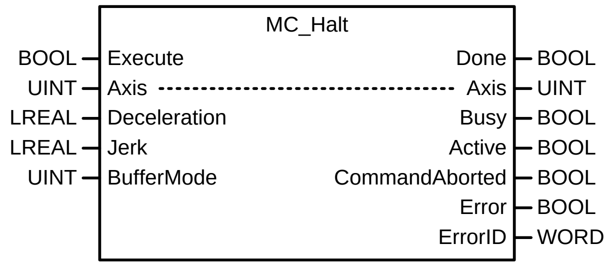
(4) If Execute input is On or the speed of axis is not 0, the axis is in 'Stopping' state, and when Done output is On and Execute input is Off, it is switched to 'Standstill' state.

(5) If the function block is re-executed (Execute input is On) before the instruction is terminated, the changed parameters are applied.
Only Deceleration and Jerk input can be updated

(6) When executing Stop immediately of motion function block during homing, 0x6040.4 Homing Bit is turned Off and 0x6040.8 Halt Bit On to immediately stop the axis. The axis stop according to homing bit and halt bit refer to the user manual of each servo drive. For example, The LS servo drive stops depending on Quick stop deceleration when homing bit is off during homing.

6.3.4 Stop (MC_Halt)

Motion Function Block



Input-Output		
UINT	Axis	Specify the axis to be commanded (See 6.2.1 Setting Range by Product)
Input		
BOOL	Execute	Give stop command to the relevant axis in the rising Edge.
LREAL	Deceleration	Specify deceleration in time of stop. [u/s^2]
LREAL	Jerk	Specify the change rate of acceleration/deceleration. [u/s^3]
UINT	BufferMode	Specify the sequential operation setting of motion function block. (Refer to 6.1.4.BufferMode)
Output		
BOOL	Done	Indicate that the speed of the relevant axis reaches 0.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that the current motion function block is controlling the relevant axis.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted while it is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block is to give a stop command to the relevant axis.

(2) The axis is 'DiscreteMotion' state while this motion function block is running, and when the speed of the relevant axis is 0, ‘Done’ output is On and changed to 'Standstill' state.

(3) BufferMode can be selected, unlike MC_Stop command. Halt command (MC_Halt) can be stopped by another motion function block.

(4) If the function block is re-executed (Execute input is On) before the instruction is terminated, the changed parameters are applied.
Deceleration, Jerk input can be updated.

6.3.5 Absolute positioning operation (MC_MoveAbsolute)

Motion Function Block

```
                     MC_MoveAbsolute
BOOL  — Execute                              Done — BOOL
UINT  — Axis  ------------------------------ Axis — UINT
BOOL  — ContinuousUpdate                     Busy — BOOL
LREAL — Position                           Active — BOOL
LREAL — Velocity                   CommandAborted — BOOL
LREAL — Acceleration                        Error — BOOL
LREAL — Deceleration                      ErrorID — WORD
LREAL — Jerk
UINT  — Direction
UINT  — BufferMode
```

Input-Output		
UINT	Axis	Specify the axis to be commanded (See 6.2.1 Setting Range by Product)
Input		
BOOL	Execute	Give an absolute position operation command to the relevant axis in the rising edge
BOOL	ContinuousUpdate	Specify the update setting of input value. (Refer to 6.1.5.Changes in Parameters during Execution of Motion Function Block).
LREAL	Position	Specify the target position.
LREAL	Velocity	Specify the maximum speed. [u/s]
LREAL	Acceleration	Specify the acceleration. [u/s^2]
LREAL	Deceleration	Specify the deceleration. [u/s^2]
LREAL	Jerk	Specify the change rate of acceleration/deceleration. [u/s^3]
UINT	Direction	Specify the operation direction. (0~4: 0-Not specified, 1-Forward direction, 2-Shortest distance, 3-Reverse direction, 4-Current direction)
UINT	BufferMode	Specify the sequential operation setting of motion function block. (Refer to 6.1.4.BufferMode)
Output		
BOOL	Done	Indicate whether to reach the specified position.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that the current motion function block is controlling the relevant axis.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted while it is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block is to give the relevant absolute position operation commands.

(2) Set the operation direction of the axis in infinite length repetition operation in Direction input, and if infinite length repetition operation is set to Prohibited, Direction input is ignored. When Direction input is the shortest distance (=2), the relevant axis doing Infinite length repetition operation automatically selects the direction which allows the shortest distance. The available range is 0-4 (0-Not specified, 1-Forward direction, 2-Shortest distance, 3-Reverse direction, 4-Current direction), and "error 0x1017" occurs in case of excess of the range.

(3) On condition that there is no motion function block is on standby after the current motion function block, If the speed is 0 after

reaching the target point, operation is completed and Done output is On.

(4) The axis is in 'DiscreteMotion' state while this motion function block is running, and it is switched to 'Standstill' state when operation is completed.

(5) If the function block is re-executed (Execute input is On) before the instruction is terminated, the changed parameters are applied.
Only Position, Velocity, Acceleration, Deceleration, Jerk, Direction input can be updated.

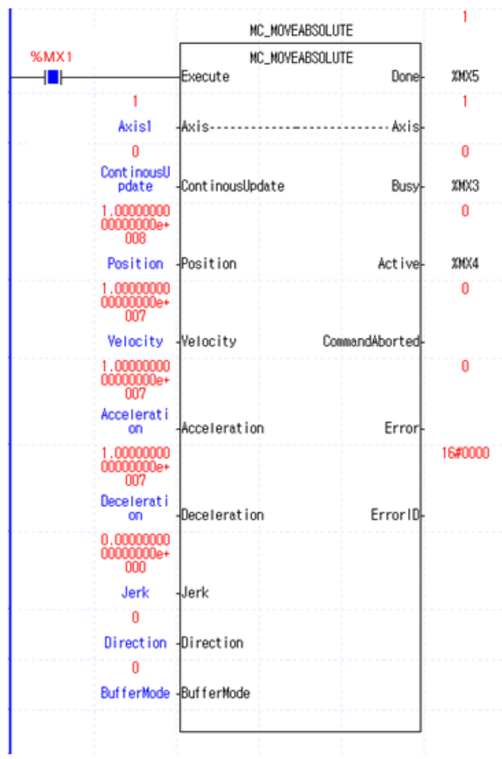
(6) Velocity input can be set to 0 or changed.

(7) During the deceleration operation, even if the Velocity and Acceleration inputs are changed by using the ContinuousUpdate function or the command re-execution function, the deceleration operation is not affected and the previous deceleration operation continues.

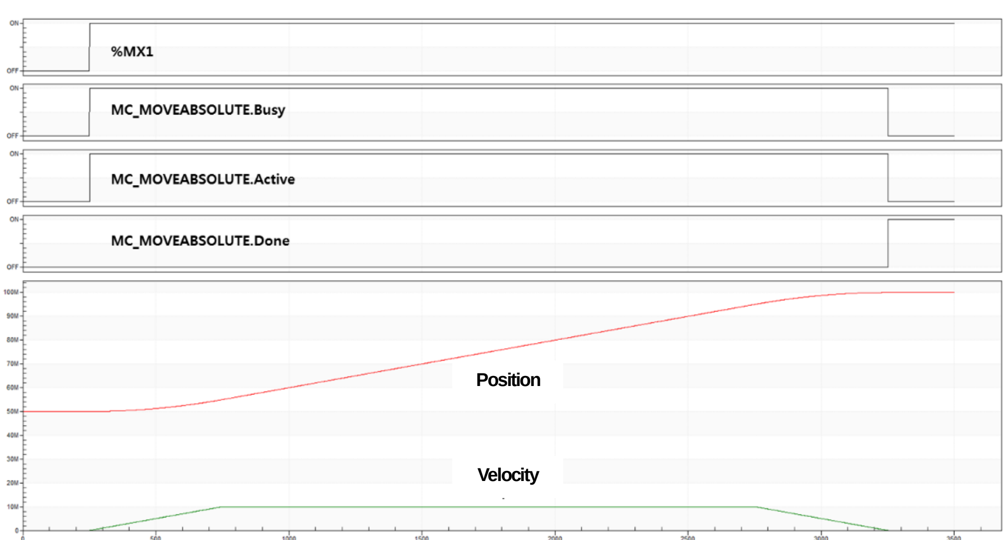
(8) Example program

This example shows the movement from the current command position of 50,000,000 to the 100,000,000 position.

(a) Function block setting



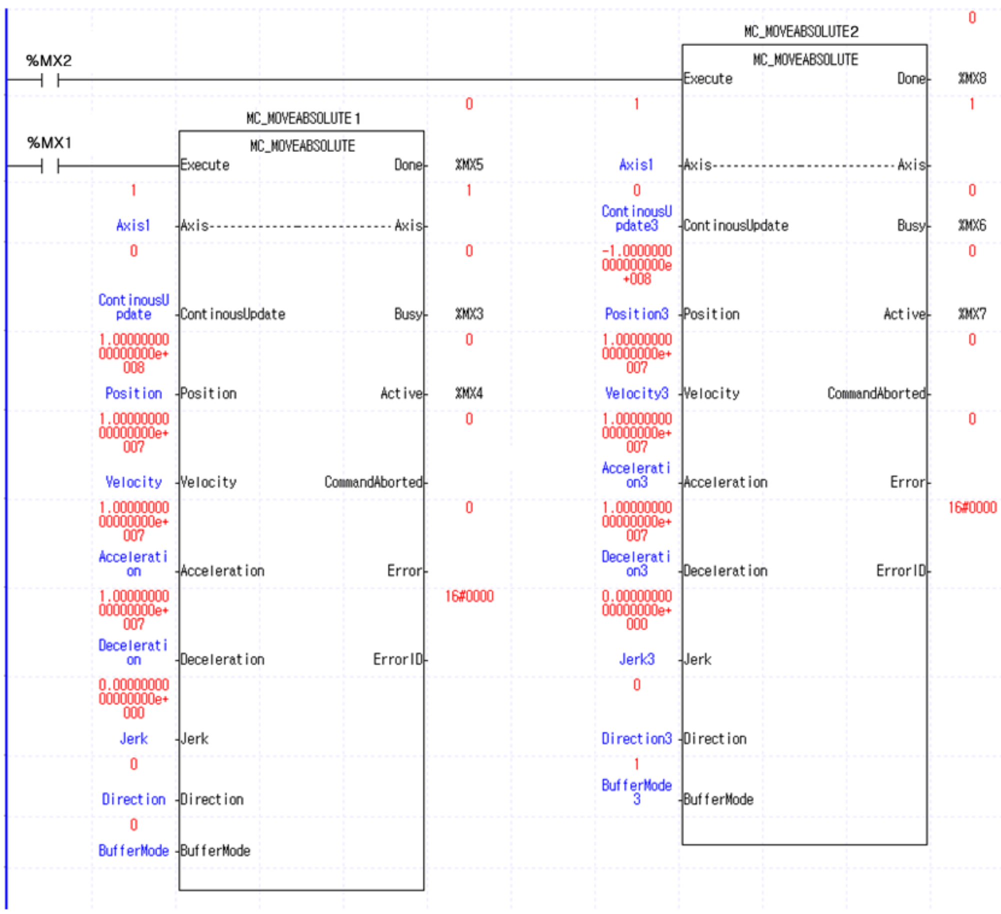
(b) Timing diagram



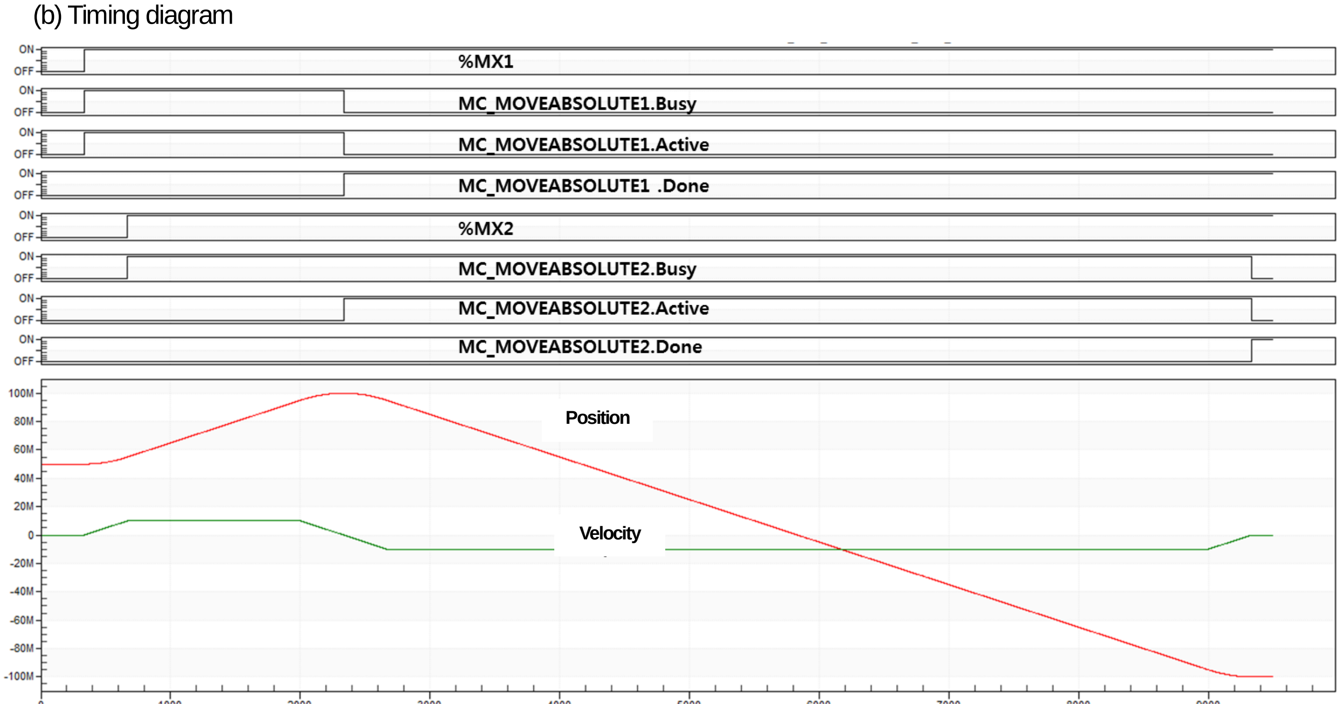
(9) Application example program

This example shows the execution of another function block with BufferMode set to 1 while moving from the current command position of 50,000,000 to the 100,000,000 position, to move to the -100,000,000 position.

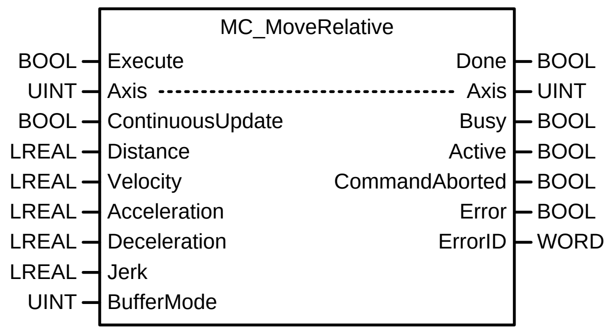
(a) Function block setting



(b) Timing diagram



6.3.6 Relative positioning operation (MC_MoveRelative)

Motion Function Block		
		
Input-Output		
UINT	Axis	Specify the axis to be commanded (See 6.2.1 Setting Range by Product)
Input		
BOOL	Execute	Give an absolute position operation command to the relevant axis in the rising edge
BOOL	ContinuousUpdate	Specify the update setting of input value. Reference
LREAL	Distance	Specify the target distance.
LREAL	Velocity	Specify the maximum speed. [u/s]
LREAL	Acceleration	Specify the acceleration. [u/s^2]
LREAL	Deceleration	Specify the deceleration. [u/s^2]
LREAL	Jerk	Specify the change rate of acceleration/deceleration. [u/s^3]
UINT	BufferMode	Specify the sequential operation setting of motion function block. (Refer to 6.1.4.BufferMode)
Output		
BOOL	Done	Indicate whether to reach the specified distance.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that the current motion function block is controlling the relevant axis.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted while it is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) Output the number of error occurred while motion function block is running.
(2) Relative position motion (MC_MoveRelative) is the motion function block which moves as far as the target distance specified in Distance input from the current position.
(3) Moving direction is decided depending on the sign of the target distance specified in Distance input, and positive (+ or No sign) moving direction leads to the forward direction, and negative (-) moving direction leads to the reverse direction.
(4) If there is no motion function block is on standby after the current motion function block and the speed is 0 after moving to the target distance, operation is completed and Done output is On.
(5) The axis is in "DiscreteMotion" state when this motion function block is running, and it is switched to "StandStill" state when operation is completed.
(6) If the function block is re-executed (Execute input is On) before the instruction is terminated, the changed parameters are applied. Only Distance, Velocity, Acceleration, Deceleration, Jerk input can be updated.

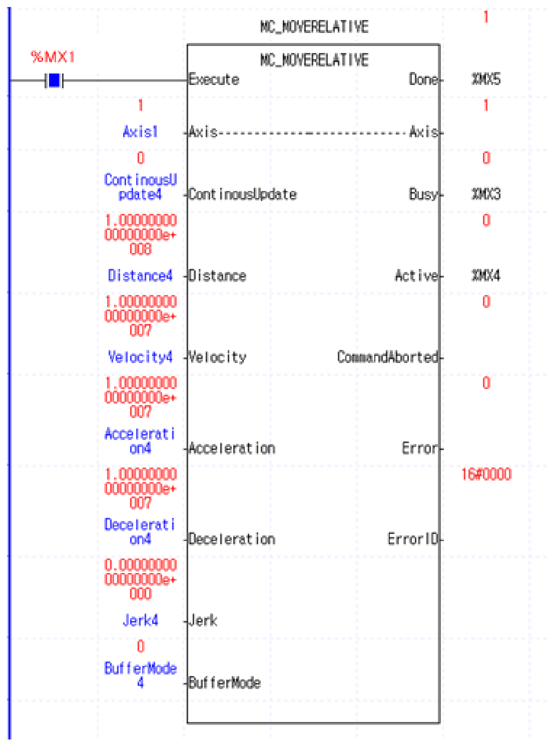
(7) Velocity input can be set to 0 or changed.

(8) During the deceleration operation, even if the Velocity and Acceleration inputs are changed by using the ContinuousUpdate function or the command re-execution function, the deceleration operation is not affected and the previous deceleration operation continues.

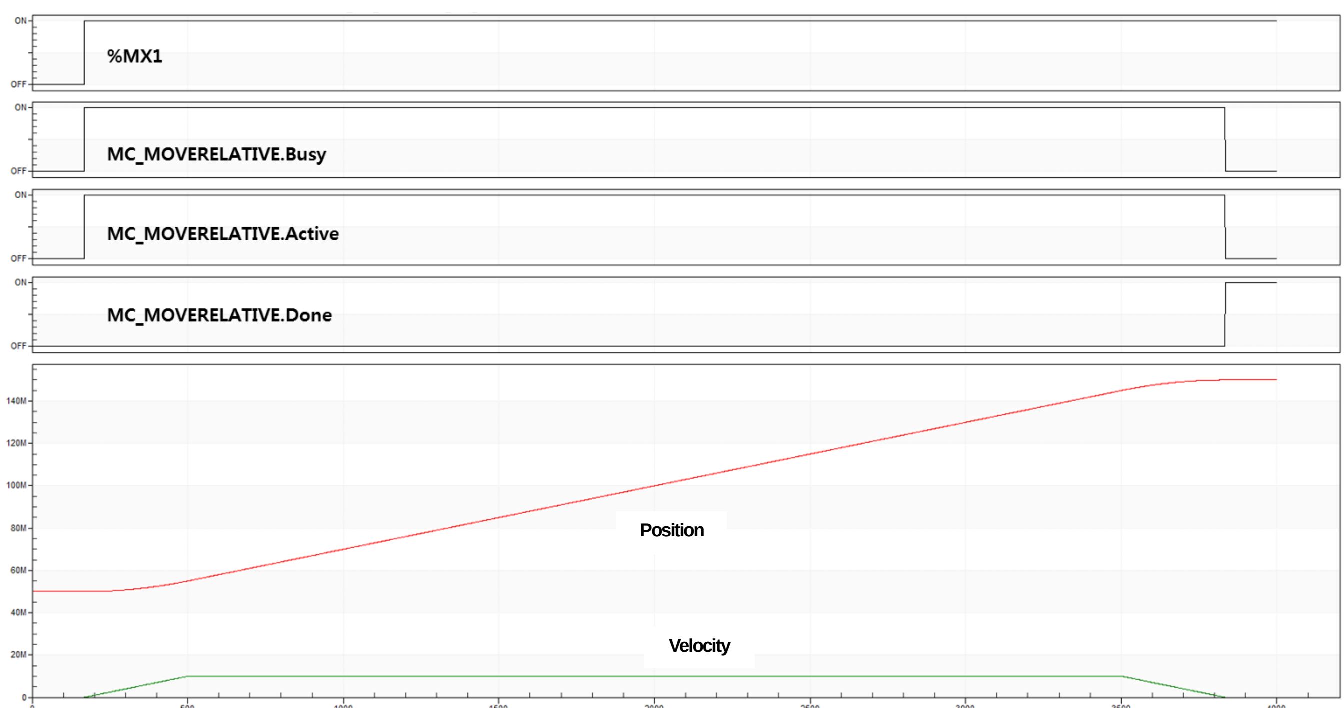
(9) Example program

This example shows the movement from the current command position of 50,000,000 to the 150,000,000 position by moving the distance corresponding to the set value (100,000,000).

(a) Function block setting



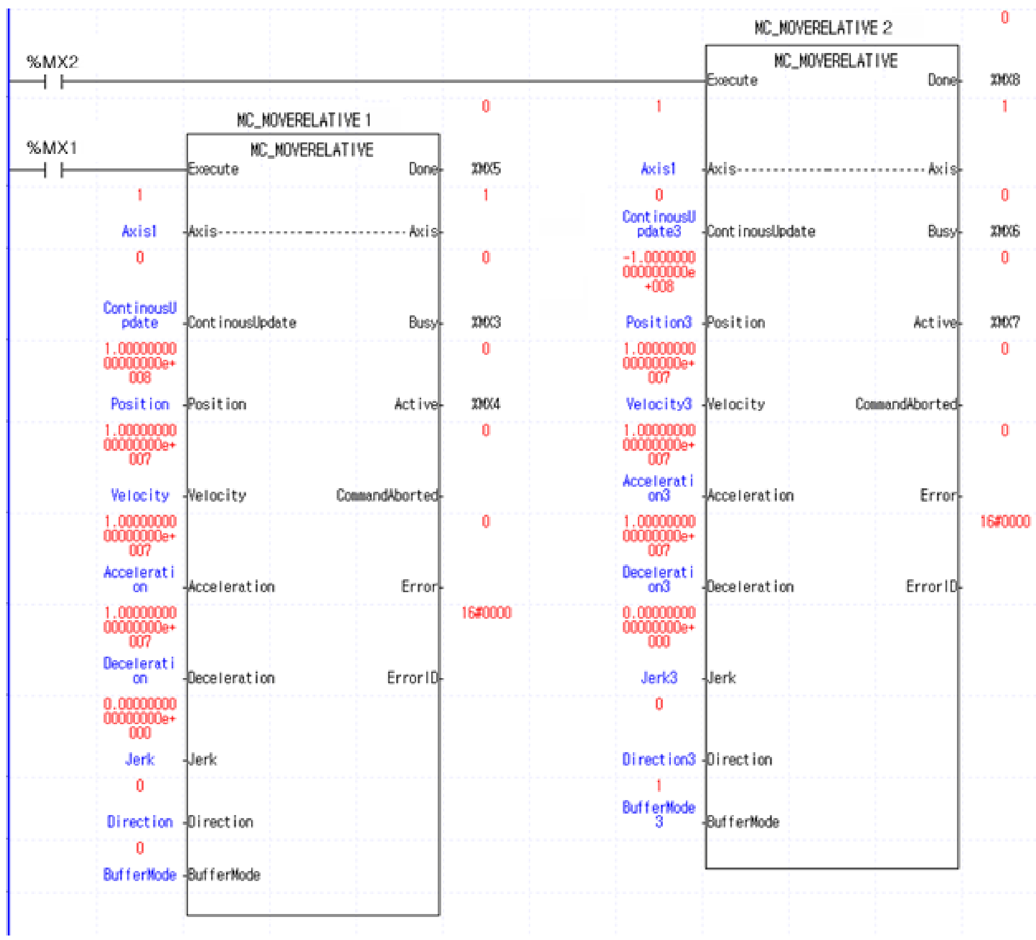
(b) Timing diagram



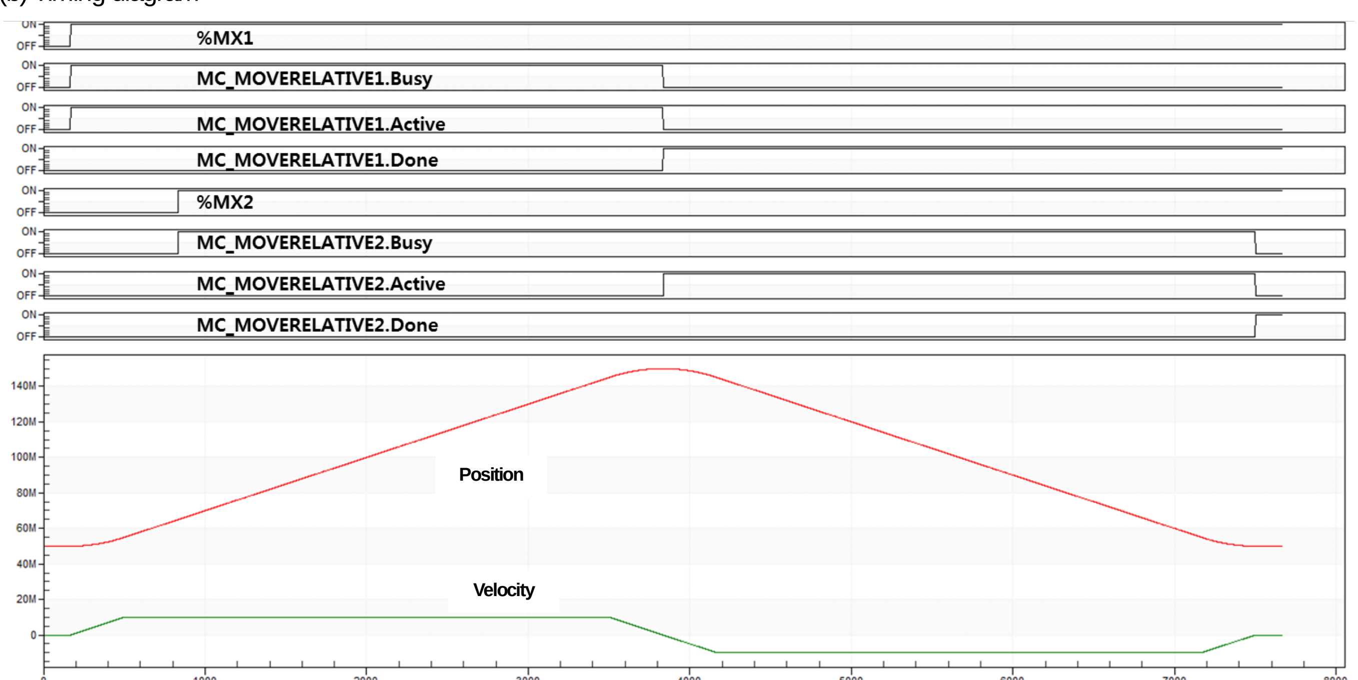
(10) Application example program

This example shows the execution of another function block with BufferMode set to 1 while moving from the current command position of 50,000,000 to the 150,000,000 position, to move to the 50,000,000 position.

(a) Function block setting



(b) Timing diagram



6.3.7 Additive positioning operation (MC_MoveAdditive)

Motion Function Block

MC_MoveAdditive

BOOL	Execute	Done	BOOL
UINT	Axis	Axis	UINT
BOOL	ContinousUpdate	Busy	BOOL
LREAL	Distance	Active	BOOL
LREAL	Velocity	CommandAborted	BOOL
LREAL	Acceleration	Error	BOOL
LREAL	Deceleration	ErrorID	WORD
LREAL	Jerk		
UINT	BufferMode		

Input-Output		
UINT	Axis	Specify the axis to be commanded (See 6.2.1 Setting Range by Product)
Input		
BOOL	Execute	Give an absolute position operation command to the relevant axis in the rising Edge.
BOOL	ContinuousUpdate	Specify the update setting of input value. Reference
LREAL	Distance	Specify the target distance.
LREAL	Velocity	Specify the maximum speed. [u/s]
LREAL	Acceleration	Specify the acceleration. [u/s^2]
LREAL	Deceleration	Specify the deceleration. [u/s^2]
LREAL	Jerk	Specify the change rate of acceleration/deceleration. [u/s^3]
UINT	BufferMode	Specify the sequential operation setting of motion function block. (Refer to 6.1.4.BufferMode)
Output		
BOOL	Done	Indicate whether to reach the specified distance.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that the current motion function block is controlling the relevant axis.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted while it is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block is to give the relevant additive position operation commands.

(2) Additive position motion (MC_MoveAdditive) is the motion function block which additionally moves as far as the position specified in Distance input from the final target position of the currently running motion function block or the latest motion function block executed in 'DiscreteMotion' state. If the current axis is executing motion function block ‘ContinuousMotion’ state, it executes operation based on the position where additive position motion (MC_MoveAdditve) is executing.

(3) The direction of the movement is determined by the positivity/negativity of the set distance. Positive distance (+ or no sign) means forward movement, and negative distance (-) means reverse movement.

(4) When reaching the target position without motion function block on standby after the current motion function block, 'Done' output is On.

(5) The axis is in 'DiscreteMotion' state while this motion function block is running, and it is switched to 'Standstill' state when operation is completed.

(6) If the function block is re-executed (Execute input is On) before the instruction is terminated, the changed parameters are applied. Only Distance, Velocity, Acceleration, Deceleration, Jerk input can be updated.

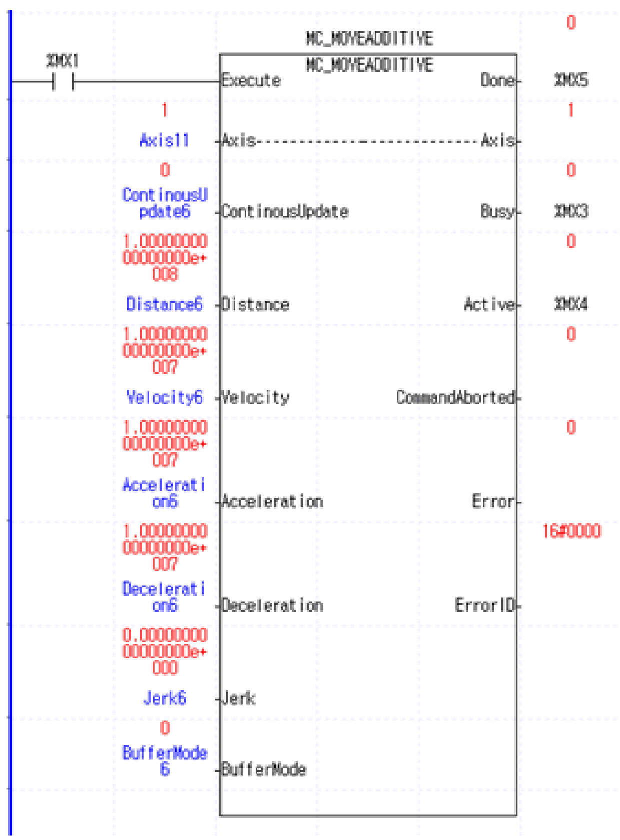
(7) Velocity input can be set to 0 or changed.

(8) During the deceleration operation, even if the Velocity and Acceleration inputs are changed by using the ContinuousUpdate function or the command re-execution function, the deceleration operation is not affected and the previous deceleration operation continues.

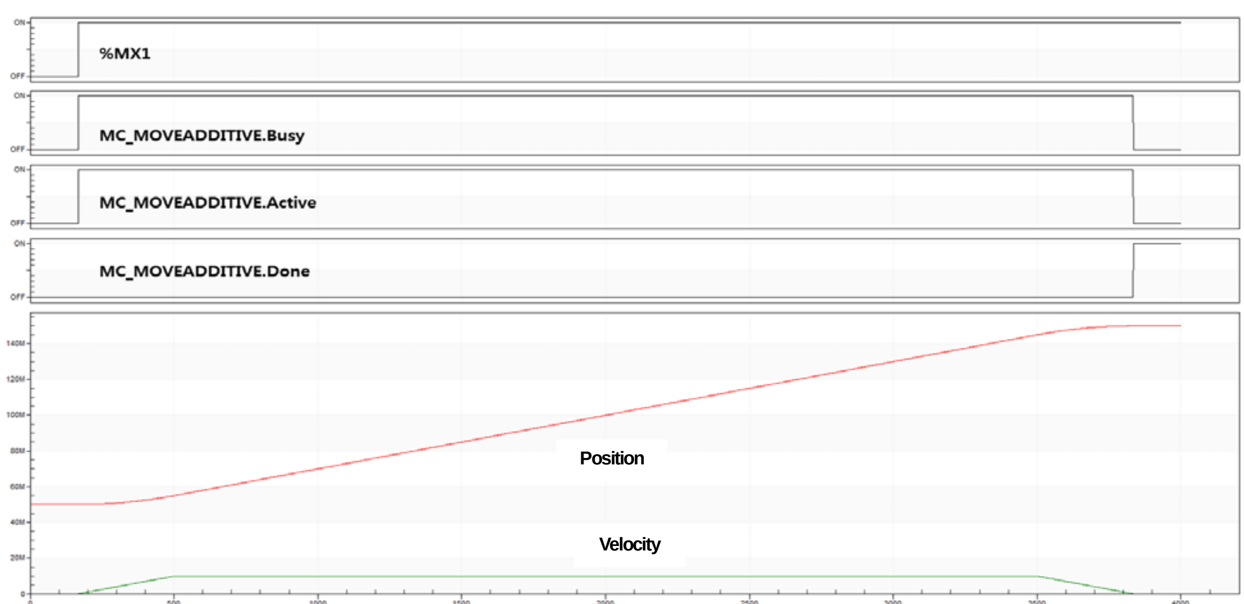
(9) Example program

This example shows the movement from the current command position of 50,000,000 to the 150,000,000 position by moving the distance corresponding to the set value (100,000,000).

(a) Function block setting



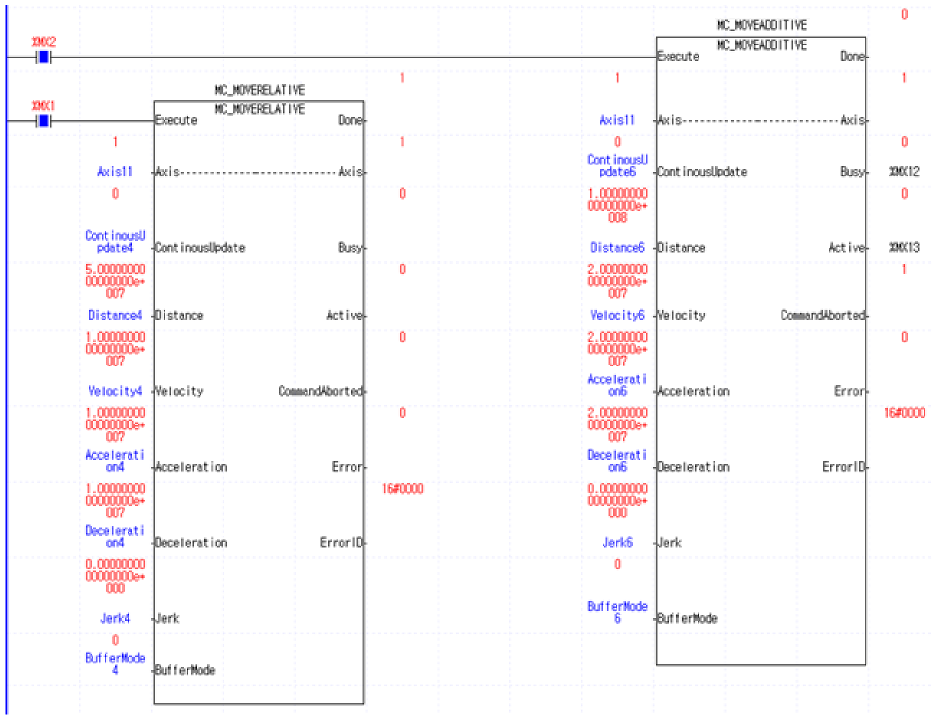
(b) Timing diagram



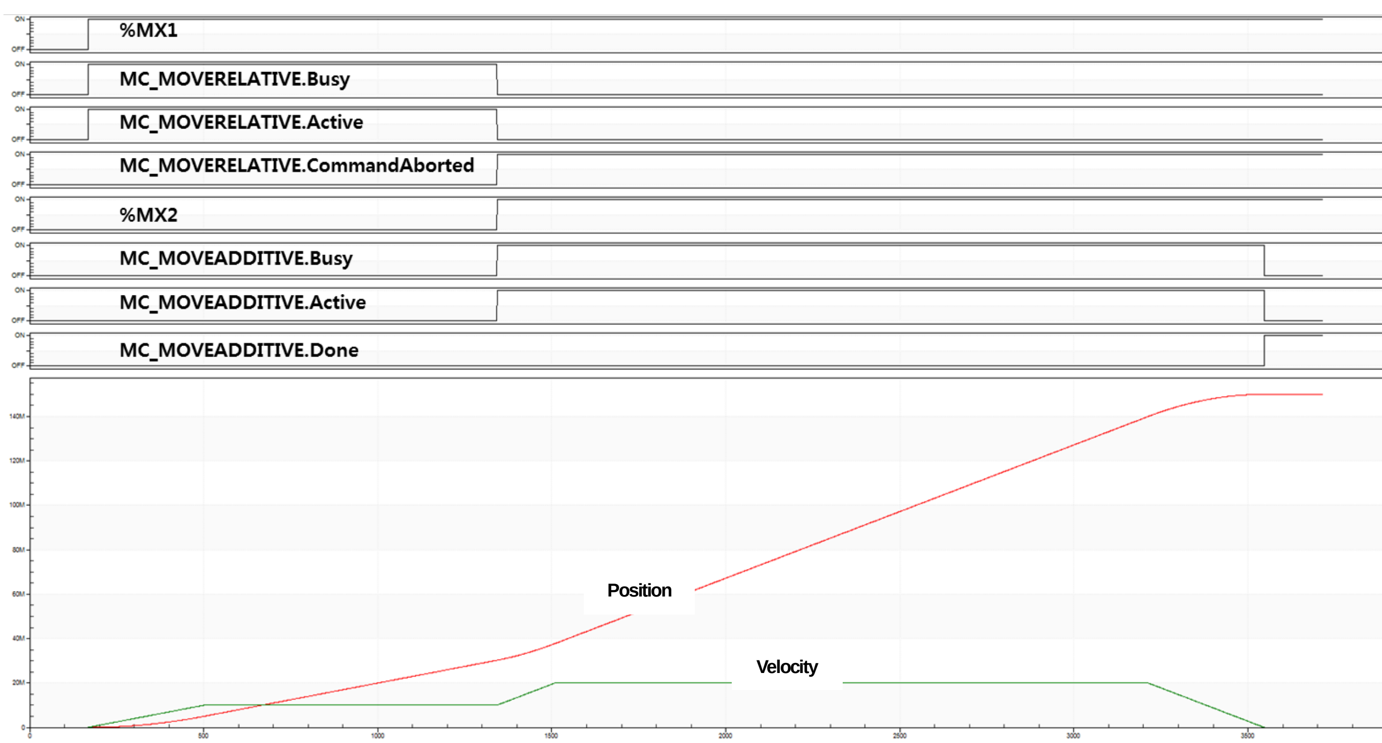
(10) Application example program

This example shows the execution of MC_MOVEADDITIVE function block while moving from current command position of 0 to the 50,000,000 position, to move an additional 100,000,000 to the 150,000,000 position.

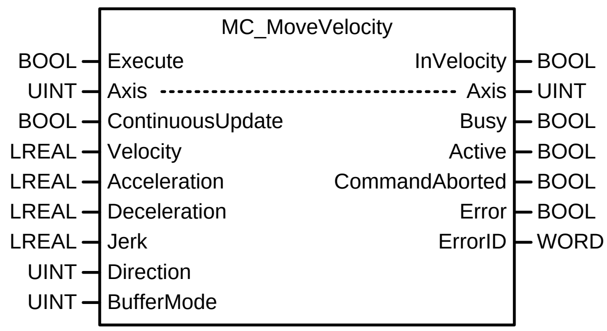
(a) Function block setting



(b) Timing diagram



6.3.8 Specified velocity operation (MC_MoveVelocity)

Motion Function Block		
		
Input-Output		
UINT	Axis	Specify the axis to be commanded (See 6.2.1 Setting Range by Product)
Input		
BOOL	Execute	Give an absolute position operation command to the relevant axis in the rising edge
BOOL	ContinuousUpdate	Specify the update setting of input value. (Refer to 6.1.5.Changes in Parameters during Execution of Motion Function Block)
LREAL	Velocity	Specify the operation speed. [u/s]
LREAL	Acceleration	Specify the acceleration. [u/s^2]
LREAL	Deceleration	Specify the deceleration. [u/s^2]
LREAL	Jerk	Specify the change rate of acceleration/deceleration. [u/s^3]
UINT	Direction	Specify the operation direction. (1 ~ 3 : 1-Forward direction, 2-Reverse direction, 3-Current direction)
UINT	BufferMode	Specify the sequential operation setting of motion function block. (Refer to 6.1.4.BufferMode)
Output		
BOOL	InVelocity	Indicate whether to reach the specified speed.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that the current motion function block is controlling the relevant axis.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted while it is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block is to give specified velocity operation command to the relevant axis.
(2) In order to stop the specified speed operation, you can make a stop command or execute another motion function block.
(3) Specify the operation speed in Velocity input. Positive sign (+ or No sign) of the operation speed value leads to forward direction, and negative (-) sign leads to reverse direction.
(4) Specify the operation direction in Direction input. But, the operation direction is affected by the sign of the specified speed value by Velocity input. For example, if you specify the negative number for the Velocity value and reverse direction for Direction input, the relevant axis lastly does forward direction operation.
(5) The output InVelocity is turned on when the axis reaches the specified speed, and it is turned off when the specified speed operation is stopped.

(6) The axis is in 'ContinuousMotion' state while this motion function block is running.

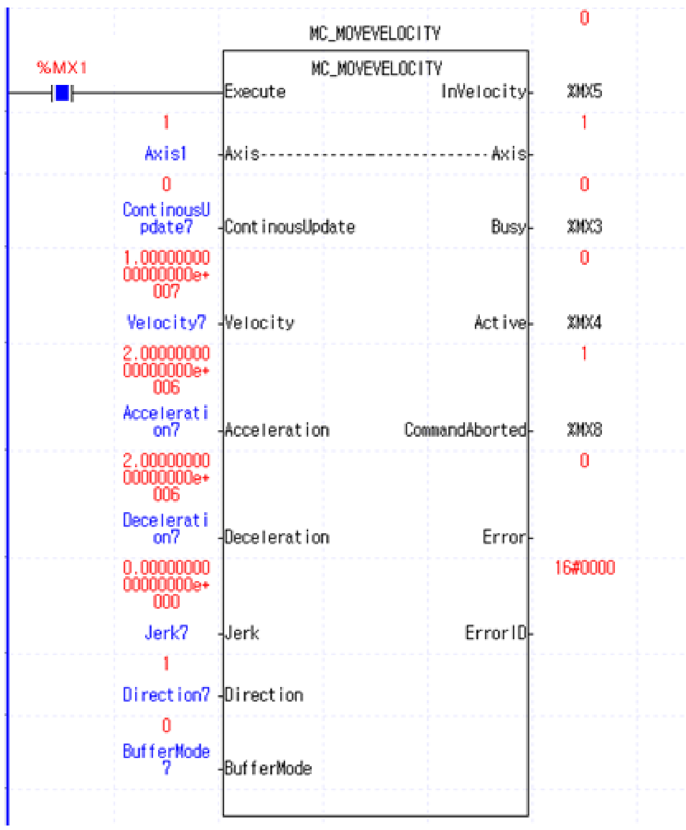
(7) If the function block is re-executed (Execute input is On) before the instruction is terminated, the changed parameters are applied.
Only Distance, Velocity, Acceleration, Deceleration, Jerk, Direction input can be updated.

(8) Velocity input can be set to 0 or changed.

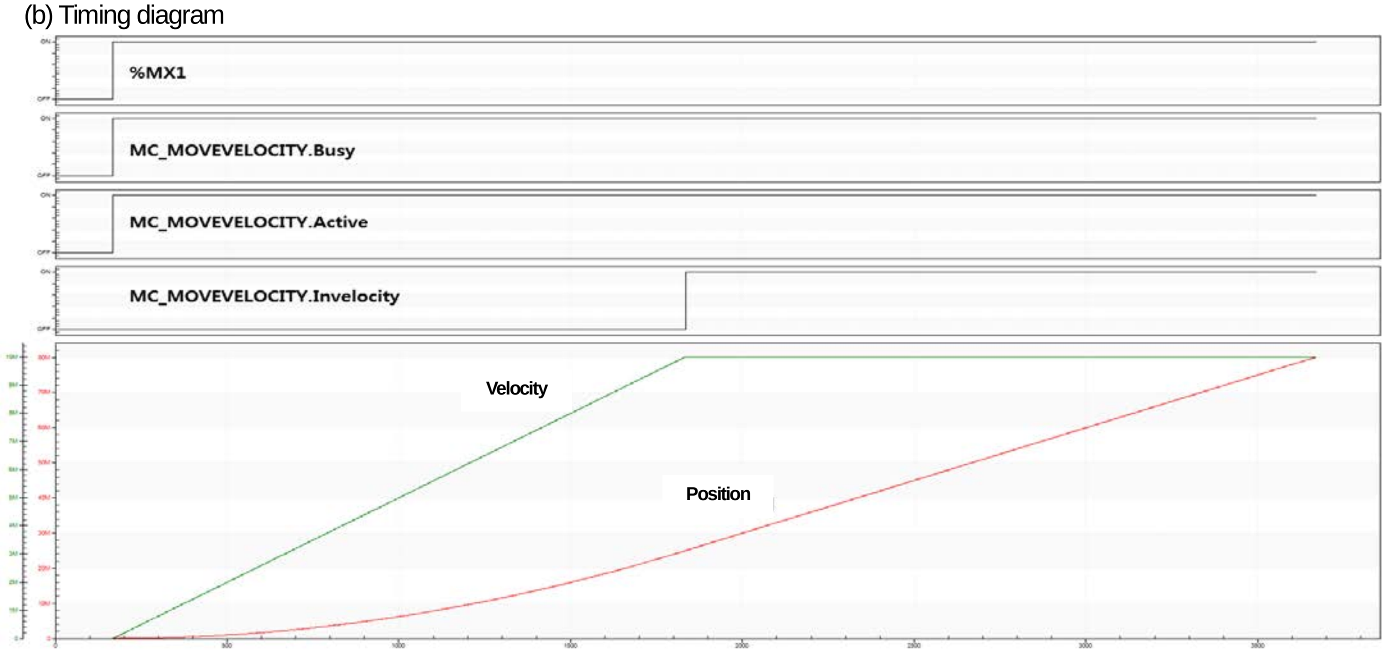
(9) Example program

This example program shows the movement at a velocity of 10,000,000. Once the set velocity is reached, InVelocityoutput is on.

(a) Function block setting



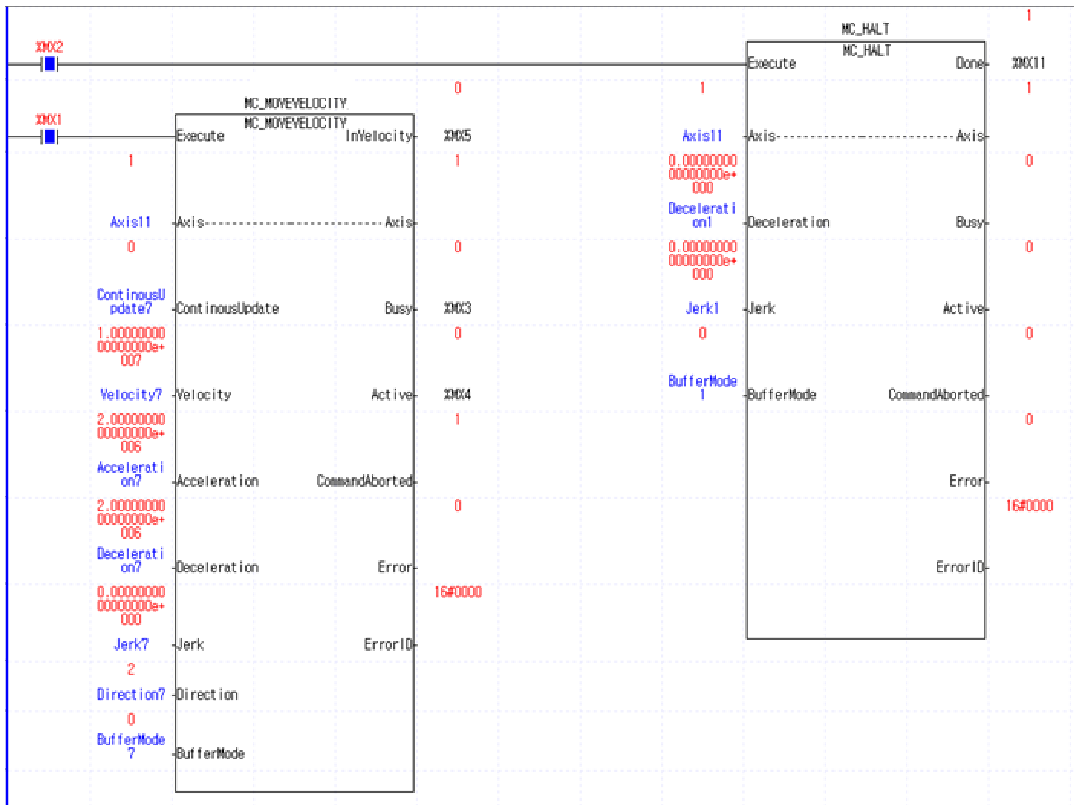
(b) Timing diagram



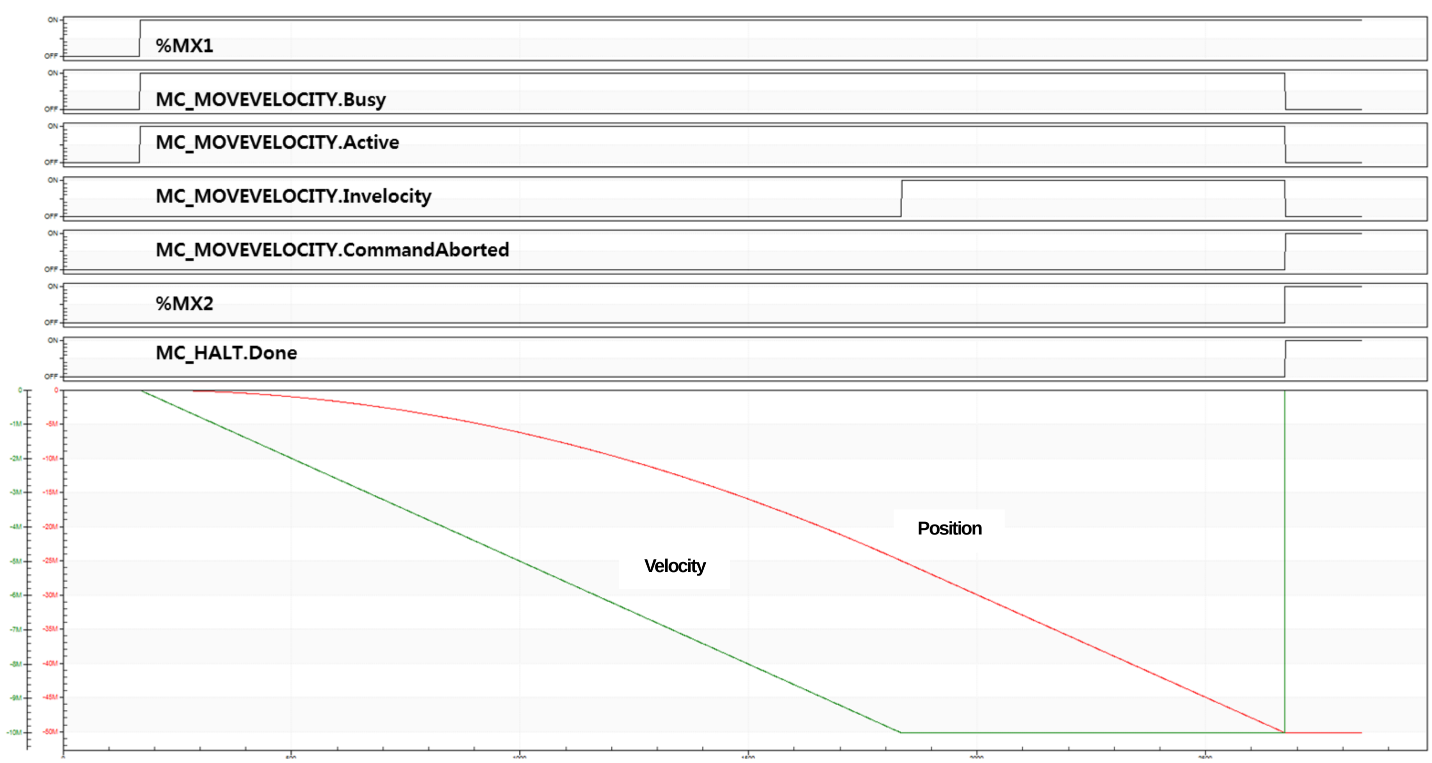
(10) Application example program

This example program shows that it stops running due to the execution of MC-Halt function block, while moving in the reverse direction at a velocity of 10,000,000.

(a) Function block setting

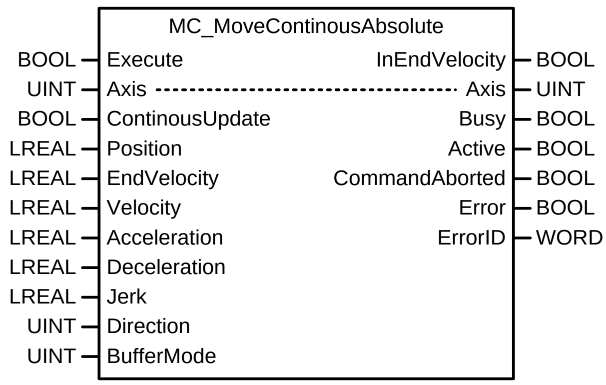


(b) Timing diagram



6.3.9 Absolute position operation ending with specified velocity operation (MC_MoveContinuousAbsolute)

Motion Function Block



Input-Output		
UINT	Axis	Specify the axis to be commanded (See 6.2.1 Setting Range by Product)
Input		
BOOL	Execute	Give an absolute position operation command to the relevant axis in the rising edge
BOOL	ContinuousUpdate	Specify the update setting of input value. Reference
LREAL	Position	Specify the target position.
LREAL	EndVelocity	Specify the operation speed after reaching the target position. [u/s]
LREAL	Velocity	Specify the maximum speed to reach the target position. [u/s]
LREAL	Acceleration	Specify the acceleration. [u/s^2]
LREAL	Deceleration	Specify the deceleration. [u/s^2]
LREAL	Jerk	Specify the change rate of acceleration/deceleration. [u/s^3]
UINT	Direction	Specify the operation direction in case of infinite length repeat operation. (0~4: 0-Not specified, 1-Forward direction, 2-Shortest distance, 3-Reverse direction, 4-Current direction)
UINT	BufferMode	Specify the sequential operation setting of motion function block. (Refer to 6.1.4.BufferMode)
Output		
BOOL	InEndVelocity	Indicate the operation at the specified speed after reaching the target position.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that the current motion function block is controlling the relevant axis.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted while it is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block is to give Specified velocity operation after relative position operation command to the relevant axis.
(2) When executing MC_MoveContinuousAbsolute, the relevant axis moves to the position specified in Position and operates at the specified speed in EndVelocity if there is no motion function block is on standby.

(3) Giving a stop command or execution of other motion function block allow to interrupt speed operation.

(4) Set the operation direction of the axis in infinite length repetition operation in Direction input, and if infinite length repetition operation is set to Prohibited, Direction input is ignored. When Direction input is the shortest distance (=2), the relevant axis selects the direction which allows the shortest distance and operates if it does infinite length repetition operation. The range can be set to 0~4(0-No specified, 1-Forward direction, 2-Shortest distance, 3-Reverse direction, 4-Current direction), if the value outside the range is set and motion function block is executed, Error is On and “0x1017” occurs in ErrorID.

(5) Output InEndVelocity is on when the relevant axis starts speed operation after reaching the specified position, and when the specified operation is interrupted, it is Off.

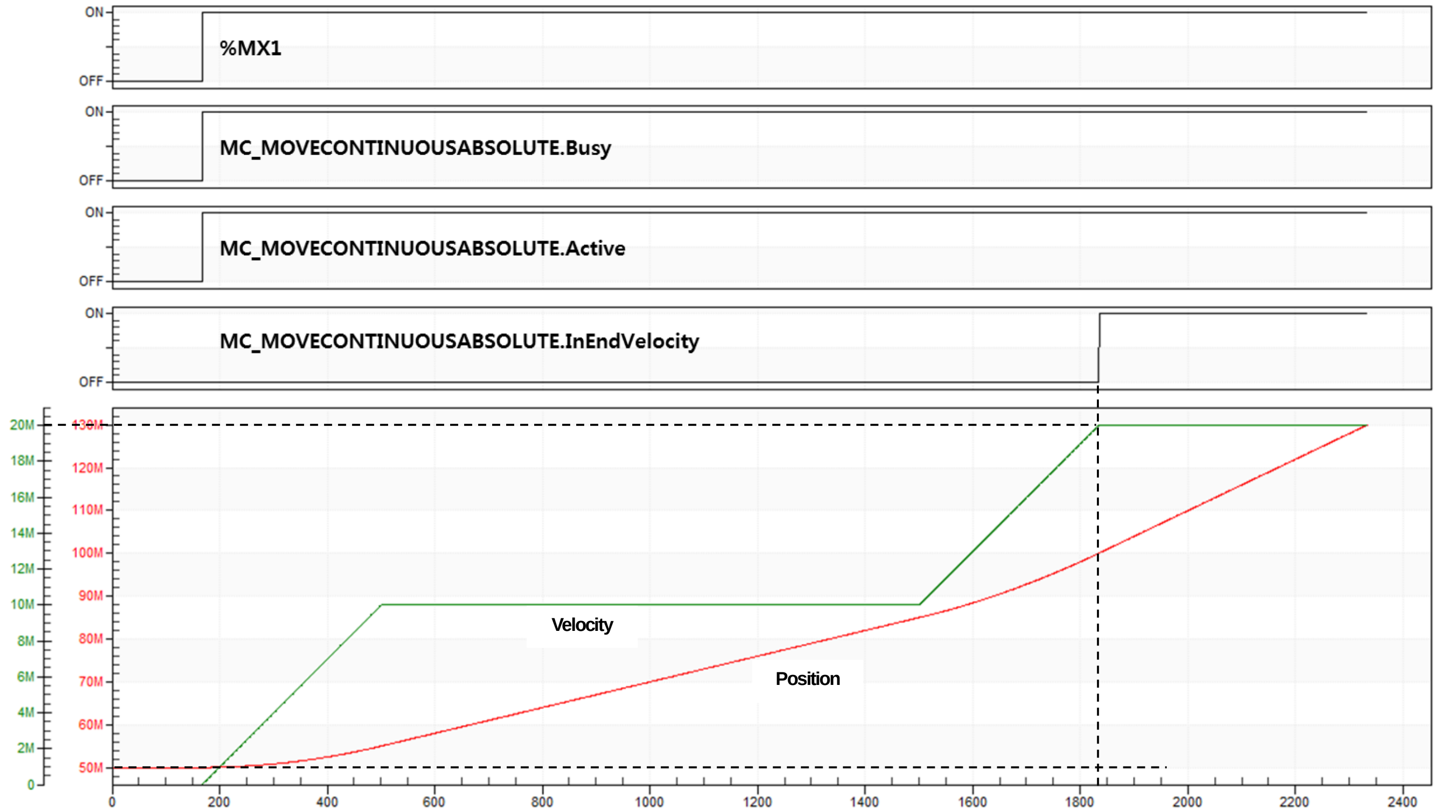
(6) The axis is in 'ContinuousMotion' state while this command is executing.

(7) If the function block is re-executed (Execute input is On) before the instruction is terminated, the changed parameters are applied.
Only Position, EndVelocity, Velocity, Acceleration, Deceleration, Jerk, Direction input can be updated. (However, in case of InEndVelocity=On, it is reflected only EndVelocity inputs.

(8) Velocity and EndVelocityy input can be set to 0 or changed.

(9) Example program
This example program shows the operation at a speed of 20,000,000 after moving from the current command position of 50,000,000 to the 100,000,000 position. Once the set position is reached, InEndVelocity output is on.
(a) Function block setting

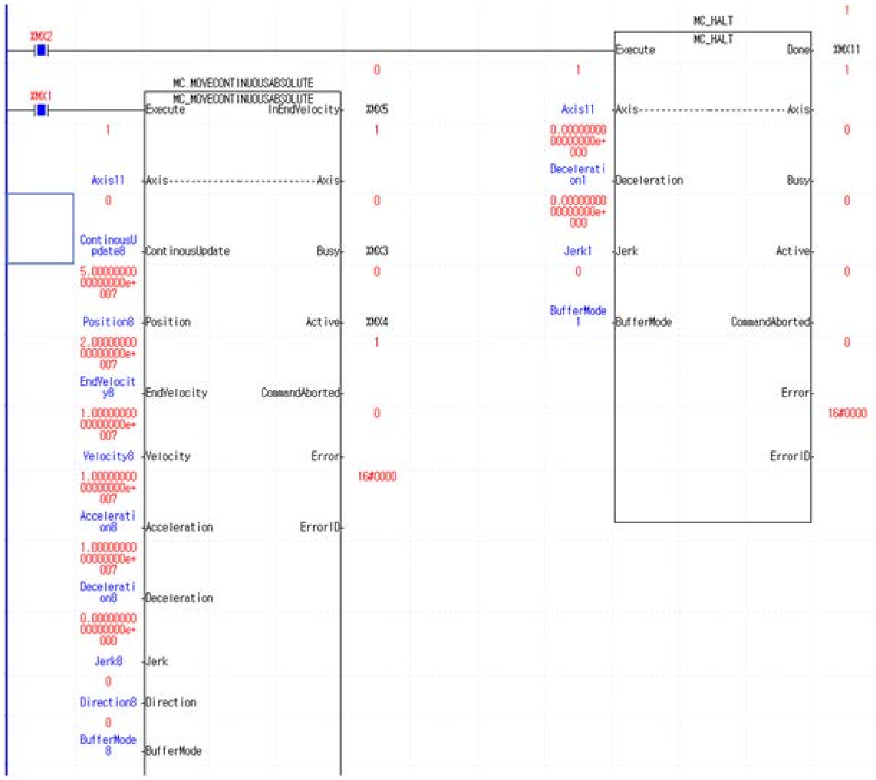
(b) Timing diagram



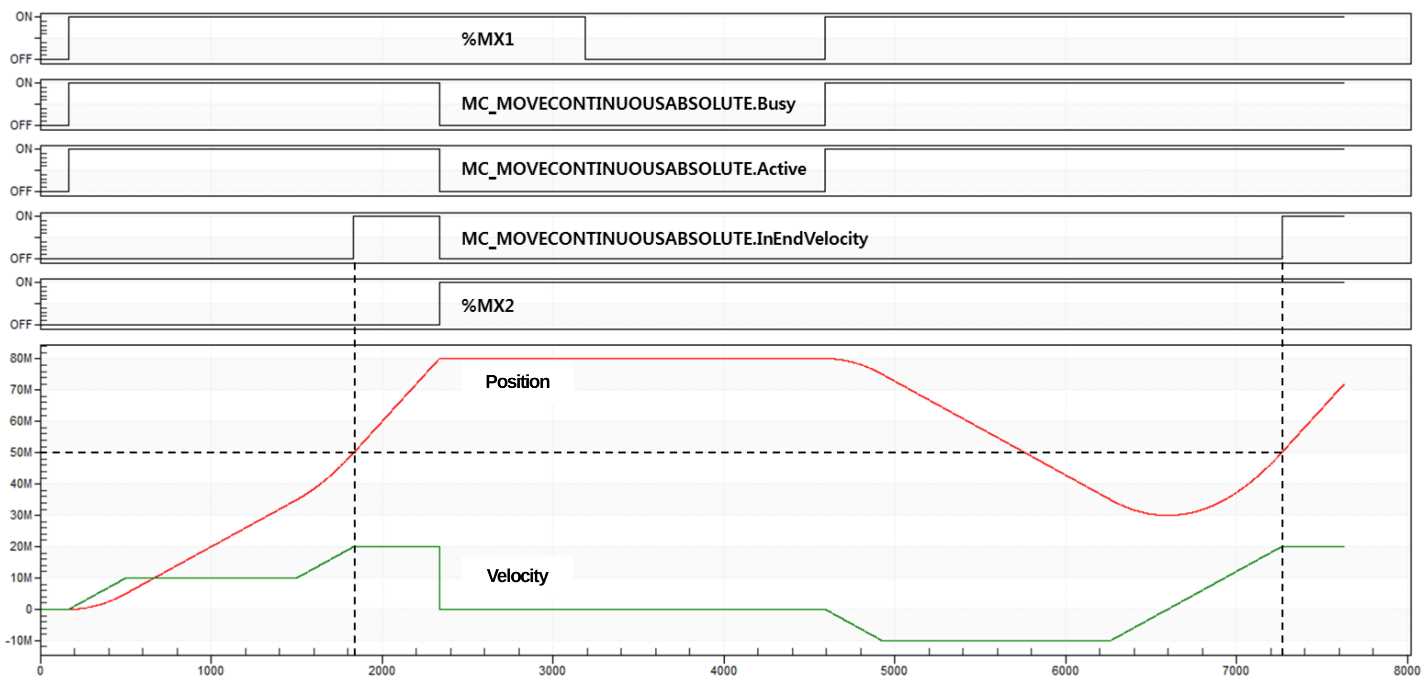
(10) Application example program

This example program shows the movement in the direction of the same speed when re-executing the function block after stopping the execution of MC-Halt function block, while moving from the current command position of 0 to the 50,000,000, then operating at a speed of 20,000,000.

(a) Function block setting

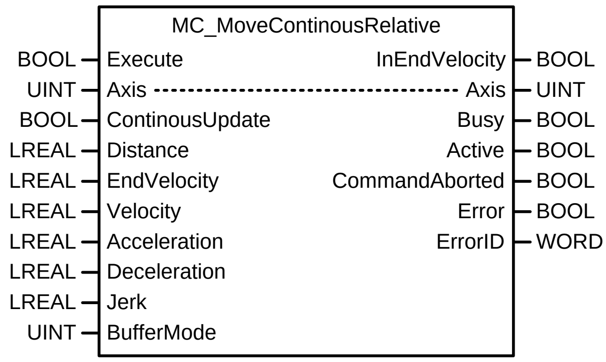


(b) Timing diagram



6.3.10 Relative position operation ending with specified velocity operation (MC_MoveContinuousRelative)

Motion Function Block



Input-Output		
UINT	Axis	Specify the axis to be commanded (See 6.2.1 Setting Range by Product)
Input		
BOOL	Execute	Give an absolute position operation command to the relevant axis in the rising edge
BOOL	ContinuousUpdate	Specify the update setting of input value. Reference
LREAL	Distance	Specify the target distance.
LREAL	EndVelocity	Specify the operation speed after reaching the target position. [u/s]
LREAL	Velocity	Specify the maximum speed to reach the target position. [u/s]
LREAL	Acceleration	Specify the acceleration. [u/s^2]
LREAL	Deceleration	Specify the deceleration. [u/s^2]
LREAL	Jerk	Specify the change rate of acceleration/deceleration. [u/s^3]
UINT	BufferMode	Specify the sequential operation setting of motion function block. (Refer to 6.1.4.BufferMode input)
Output		
BOOL	InEndVelocity	Indicate the operation at the specified speed after reaching the target position.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that the current motion function block is controlling the relevant axis.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted while it is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block gives MC_MoveContinuousRelative command to the relevant axis.
(2) When executing MC_MoveContinuousRelative, the relevant axis operates at the speed specified in EndVelocity after moving the distance specified in Distance if there is no motion function block is on standby.
(3) Giving a stop command or operation of other motion function block allow to interrupt specified velocity motion.
(4) Output InEndVelocity is On when the relevant axis starts speed operation and reaches the specified speed after moving the specified distance, and when specified velocity motion is interrupted, it is Off.
(5) The axis is in 'ContinuousMotion' state while this motion function block is running.
(6) If the function block is re-executed (Execute input is On) before the instruction is terminated, the changed parameters are applied.

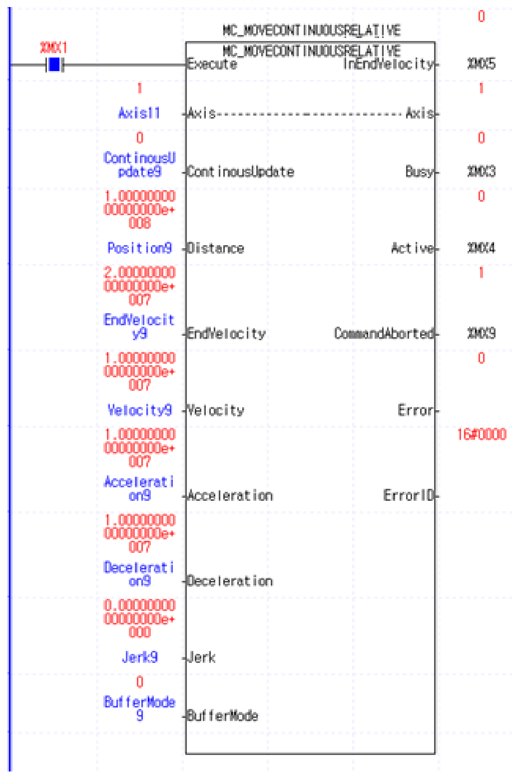
Only Distance, EndVelocity, Velocity, Acceleration, Deceleration, Jerk input can be updated. (However, in case of InEndVelocity=On, it is reflected only EndVelocity inputs.

(7) Velocity and EndVelocityy input can be set to 0 or changed.

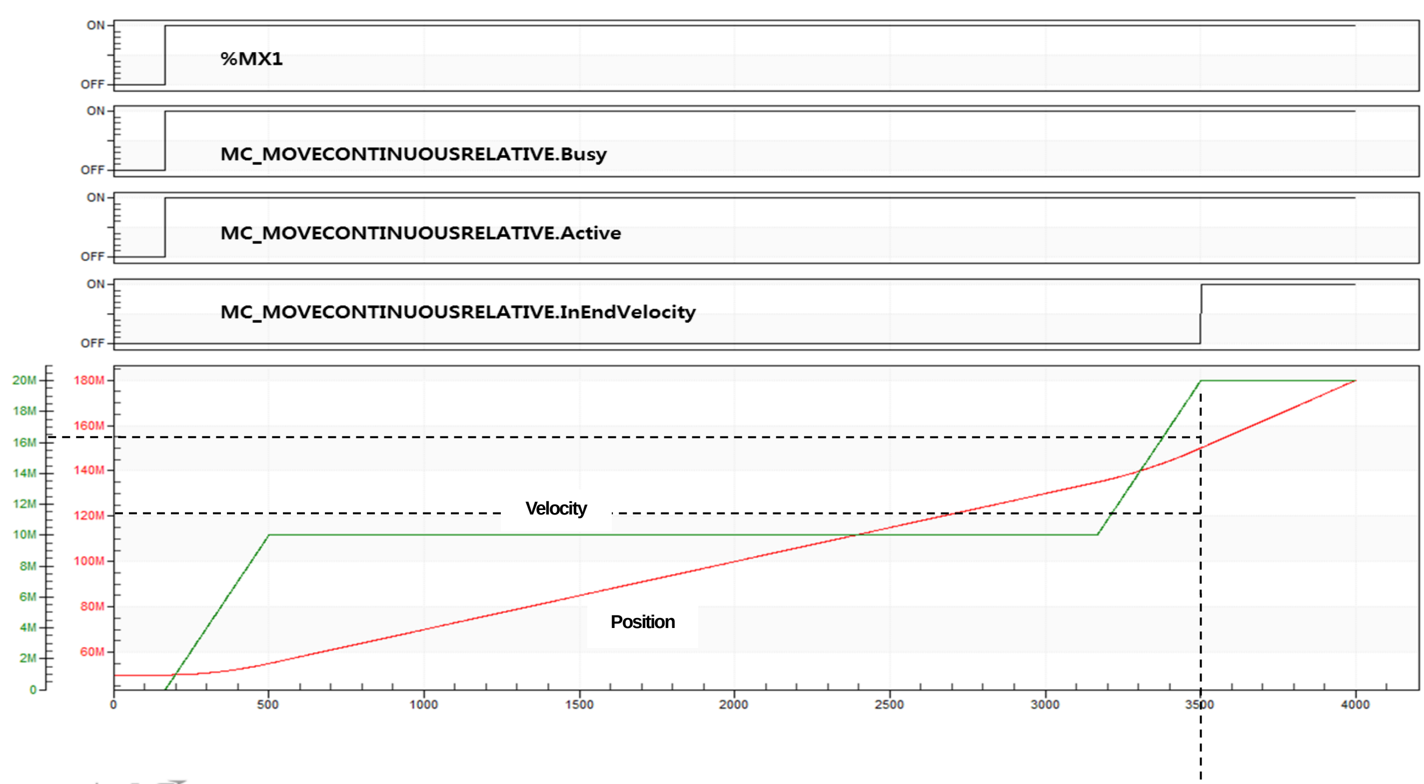
(8) Example program

This example program shows the operation at a velocity of 20,000,000 after moving from the current command position of 50,000,000 to the 150,000,000 position by moving the distance corresponding to the set value (100,000,000). Once the set position is reached, InEndVelocity output is on.

(a) Function block setting



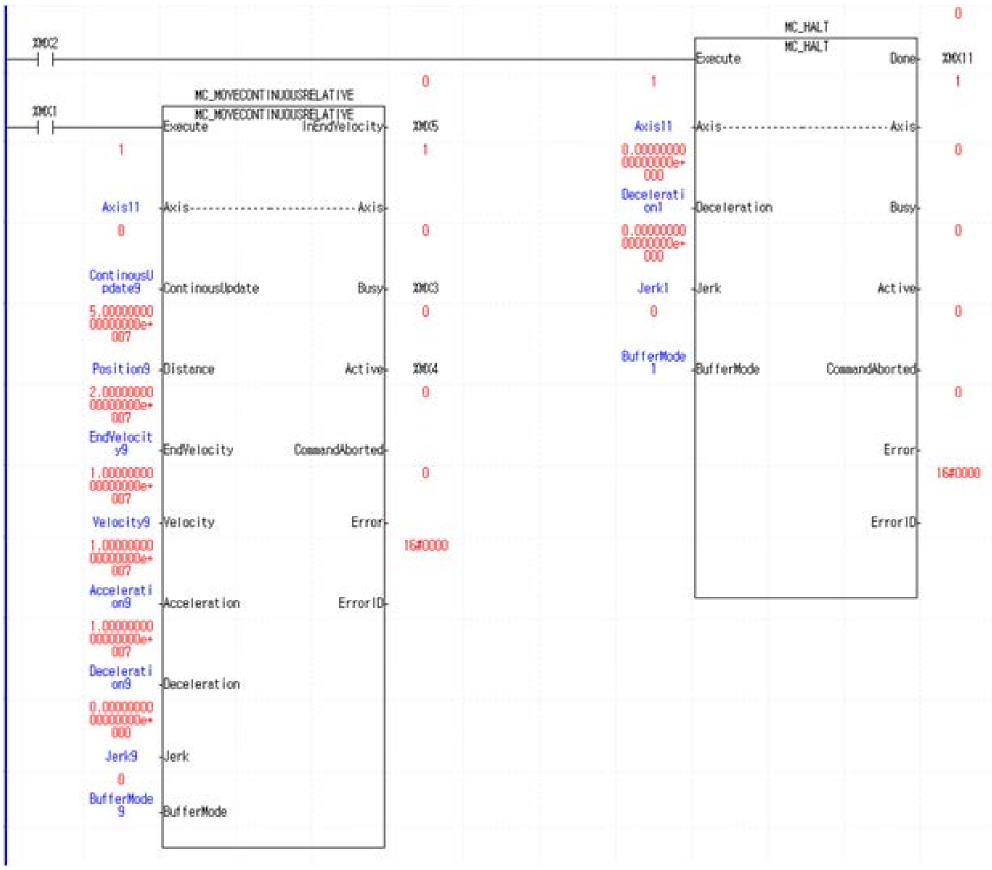
(b) Timing diagram



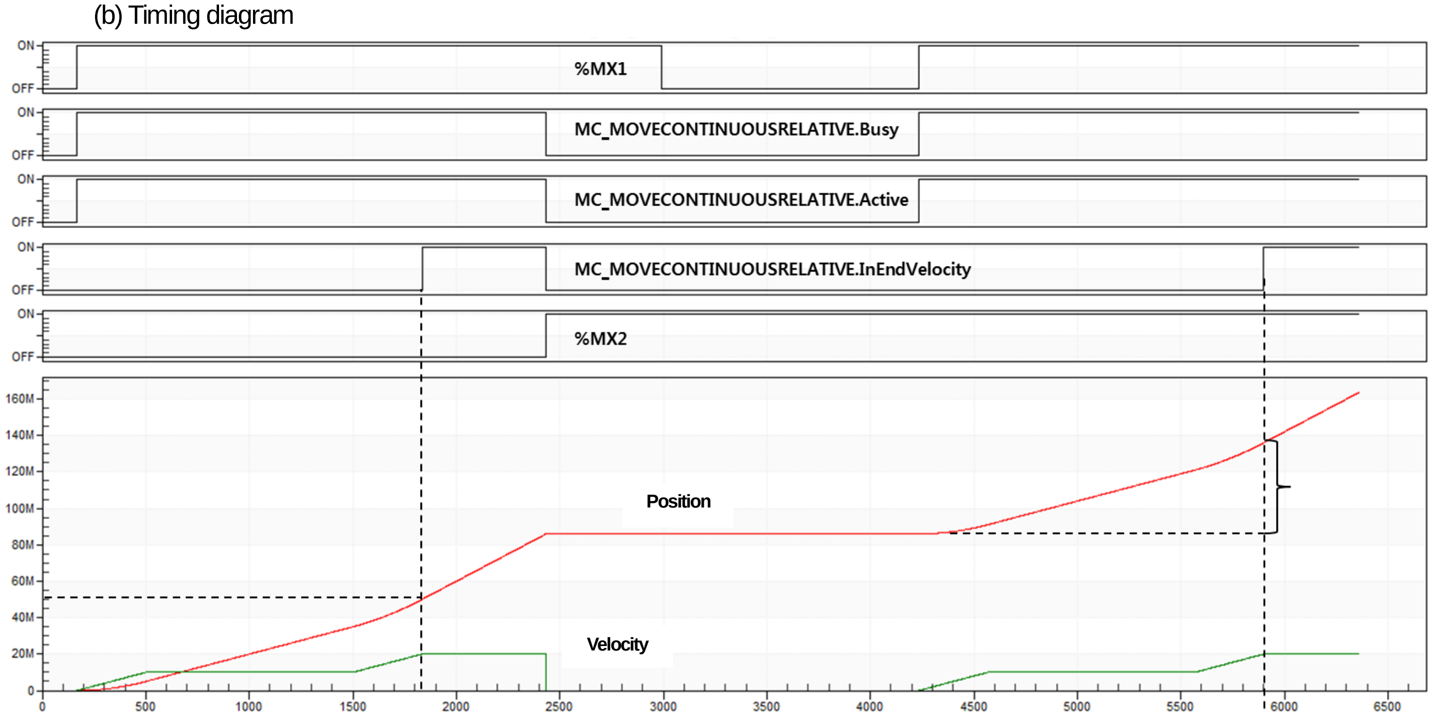
(9) Application example program

This example program shows the movement at a velocity of 20,000,000 after moving from the current command position of 0 to the 50,000,000 position, then operating at a velocity of 20,000,000, stopping by executing MC_Halt function block, moving to the same relative position (20,000,000) by re-executing the function block.

(a) Function block setting

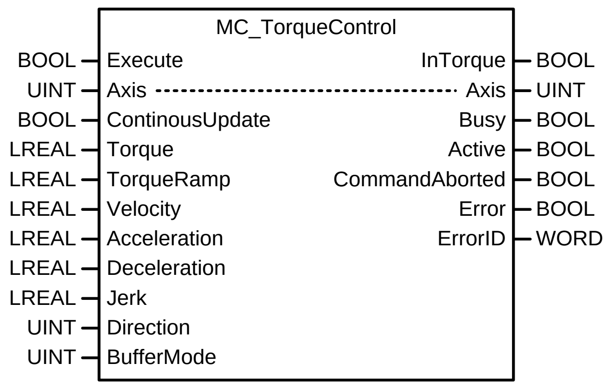


(b) Timing diagram



6.3.11 Torque Control (MC_TorqueControl)

Motion Function Block



Input-Output		
UINT	Axis	Specify the axis to be commanded (See 6.2.1 Setting Range by Product, Real axis)
Input		
BOOL	Execute	Give an absolute position operation command to the relevant axis in the rising edge
BOOL	ContinuousUpdate	Specify the update setting of input value. Reference
LREAL	Torque	Specify the target torque. [u]
LREAL	TorqueRamp	Specify the ascending slope of torque. [u/s]
LREAL	Velocity	Unused
LREAL	Acceleration	Unused
LREAL	Deceleration	Unused
LREAL	Jerk	Unused
UINT	Direction	Specify the operation direction. (1~2 : 1-Forward direction, 2-Reverse direction)
UINT	BufferMode	Specify the sequential operation setting of motion function block. (Refer to 6.1.4 BufferMode input)
Output		
BOOL	InTorque	Indicate that the input torque value and currently operating torque value are same.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that the current motion function block is controlling the relevant axis.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted while it is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block is to give torque control command to the relevant axis.
(2) When executing torque control (MC_Torque), the relevant axis performs the control to keep the torque value specified in Torque input.
(3) Giving a stop command or operation of other motion function block allow to interrupt specified velocity motion.
(4) Specify the gradient to reach the target torque value in TorqueRamp input.
(5) Specify the operation direction in Direction input. When setting the value outside the range and executing motion function block,

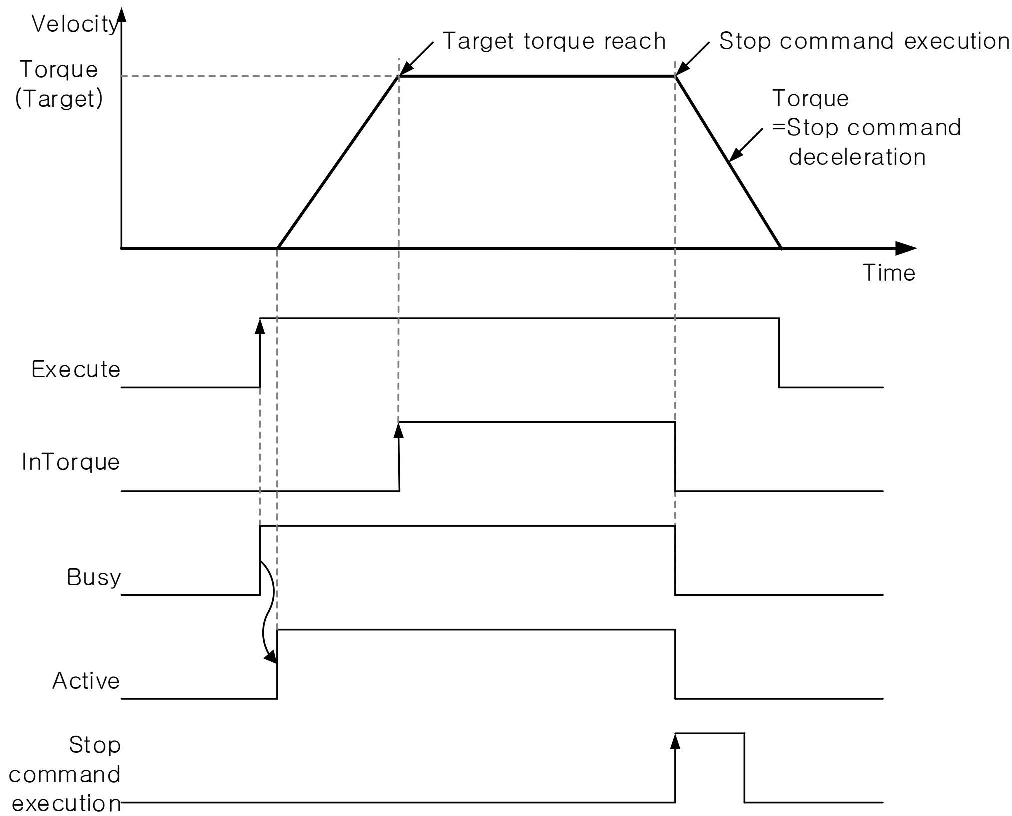
Error is On and “0x1017” occurs in ErrorID.

(6) Output InTorque is On when the relevant axis reaches the specified torque, and when torque control operation is interrupted, it is Off.

(7) The axis is in 'ContinuousMotion' state while this motion function block is running.

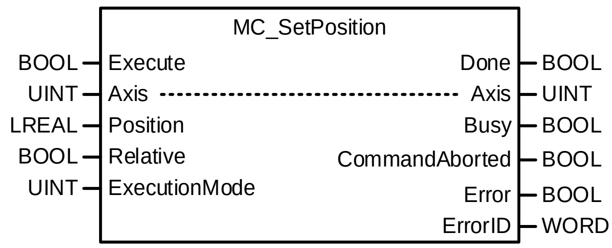
(8) If the function block is re-executed (Execute input is On) before the instruction is terminated, the changed parameters are applied. Only Torque, TorqueRamp, Direction input can be updated.

(9) Timing diagram



6.3.12 Setting the current position (MC_SetPosition)

Motion Function Block



Input-Output		
UINT	Axis	Specify the axis to be commanded (See 6.2.1 Setting Range by Product)
Input		
BOOL	Execute	Specify the current position of the relevant axis in the rising Edge.
LREAL	Position	Specify the position.
BOOL	Relative	0: Position value=Absolute position, 1: Position value=Relative position
UINT	ExecutionMode	0: Immediately applied the position value, 1: Applied at the same point with 'Buffered' of Buffermode
Output		
BOOL	Done	Indicate the completion state of motion function block.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted while it is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block is to set the current position of the relevant axis.

(2) Specify the position in Position input. When executing motion function block, if Relative input is Off, the position of the relevant axis is replaced by the value of Position input, and if Relative input is On, the value of Position input is added to the current position of the relevant axis.

(3) ExecutionMode input specifies the setting point. 0 means to be set immediately after motion function block, and 1 means to be set at the same point with 'Buffered' in sequential operation setting. The value unable to be set causes "error0x101B".

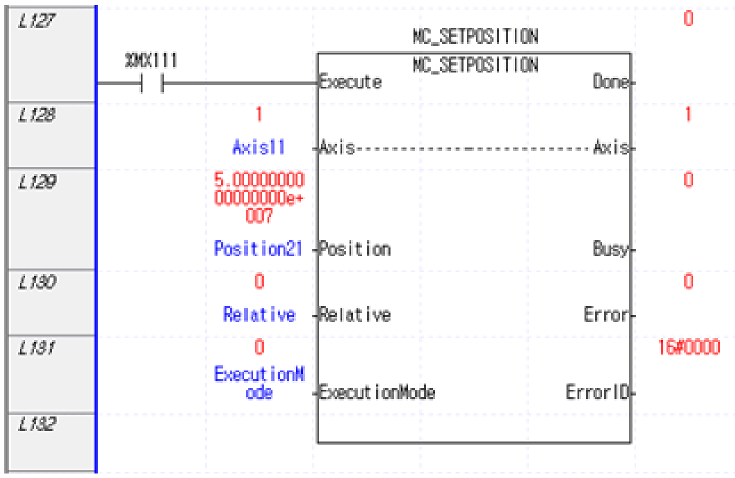
0 (mcImmediately): Change the parameter value immediately after executing function block (rising Edge in Execute input). If the relevant axis is in running, operation can be affected.

1 (mcQueued): Changed at the same point with 'Buffered' in Buffermode. (Refer to 6.1.4 Buffermode input)

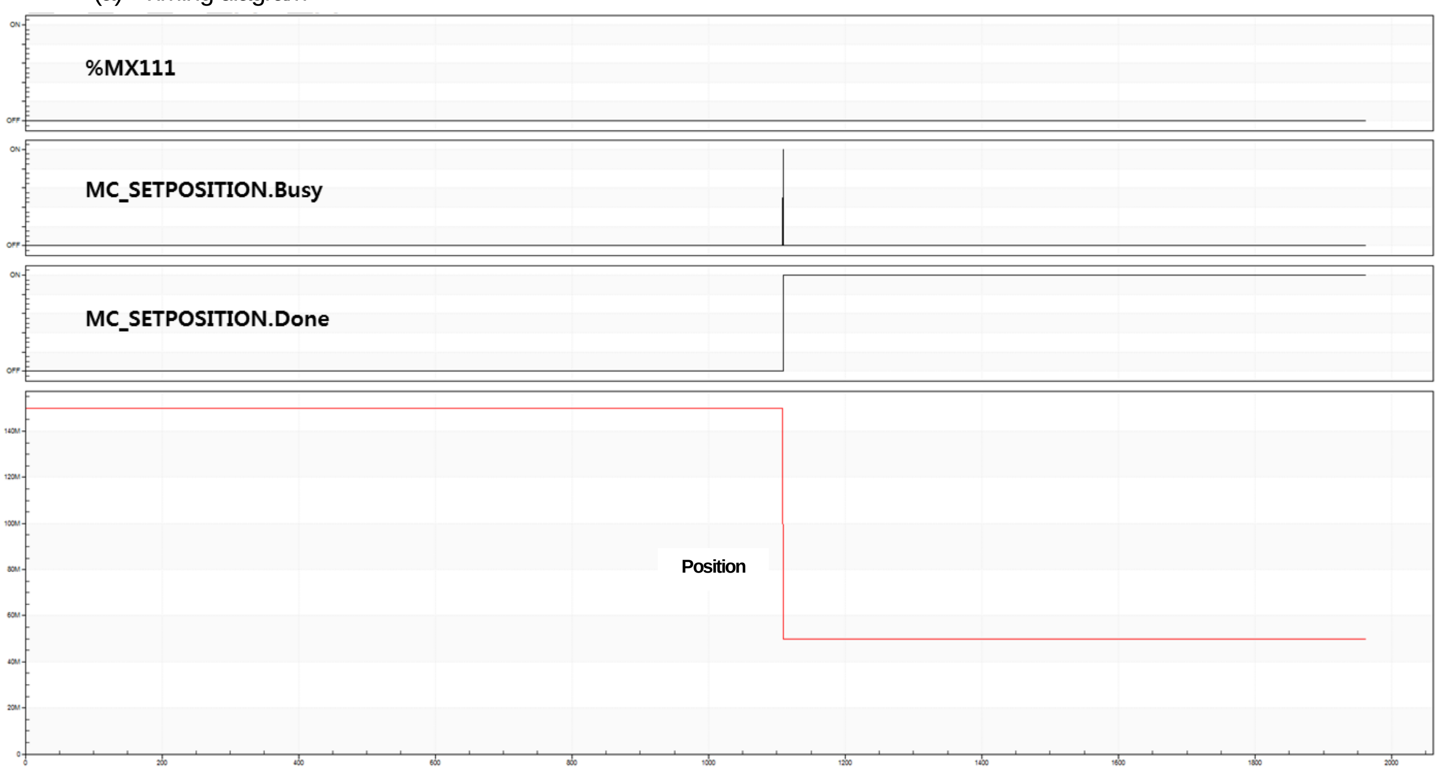
(4) Example program

This example program shows the setting of the current position to 200,000,000 position by adding a relative position (Relative=1) corresponding to the set value (50,000,000) from the current position of 150,000,000.

(a) Function block setting

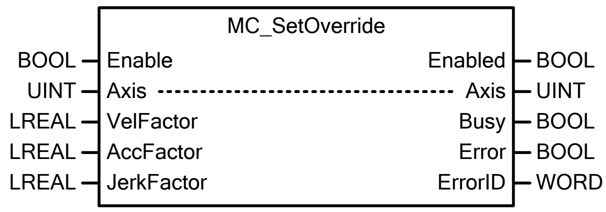


(a) Timing diagram



6.3.13 Velocity/Acceleration override (MC_SetOverride)

Motion Function Block



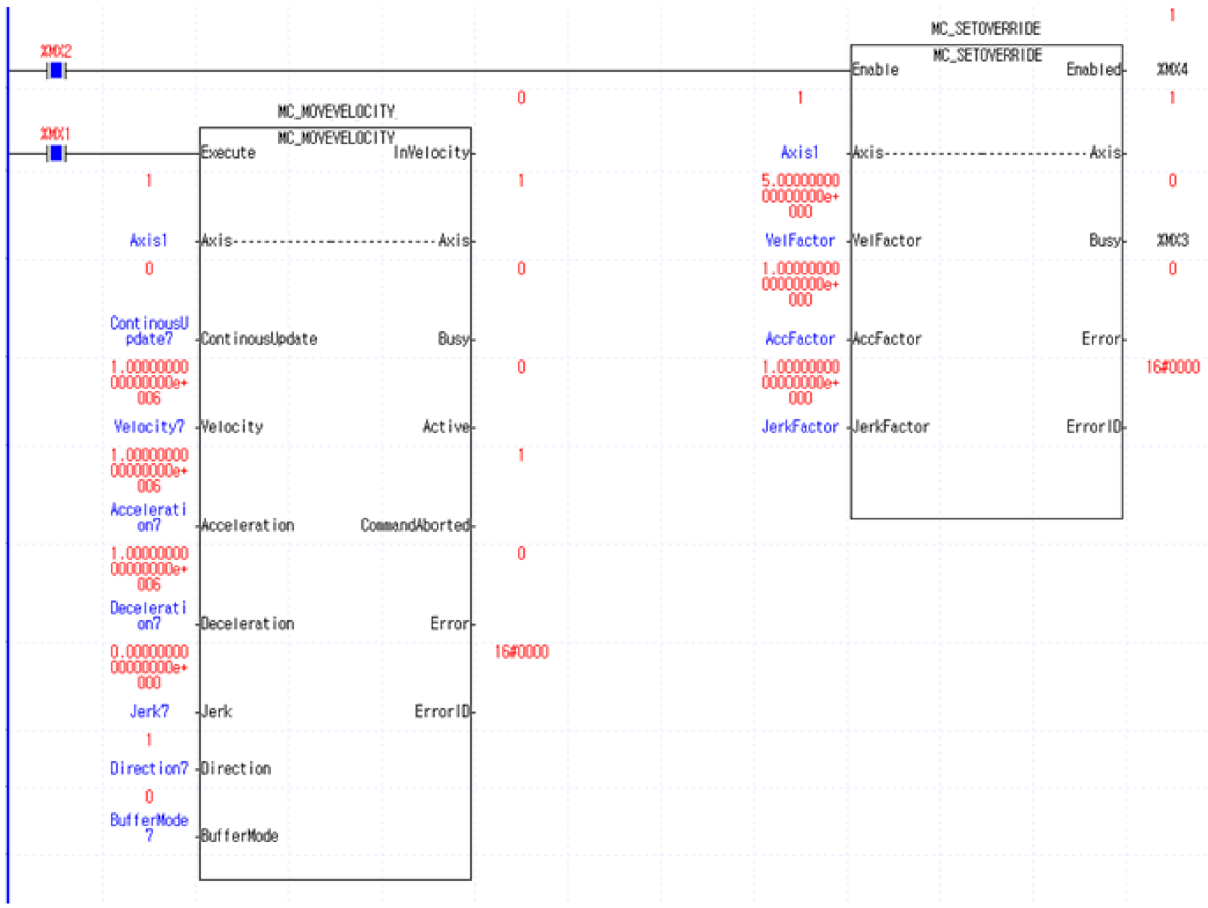
Input-Output		
UINT	Axis	Specify the axis to be commanded (See 6.2.1 Setting Range by Product)
Input		
BOOL	Enable	Execute override operation in the relevant axis while input is activated.
LREAL	VelFactor	Specify the override rate of speed.
LREAL	AccFactor	Specify the override rate of acceleration/deceleration.
LREAL	JerkFactor	Specify the override rate of the change rate of acceleration.
Output		
BOOL	Enabled	Indicate that the override rate was applied successfully.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block is to override the speed of the relevant axis, acceleration, and the change rate of acceleration.
(2) Override rate which is applied to the relevant axis can be specified and changed while Enable input is On. If Enable input is Off, override rate right before the Off is maintained.
(3) Specify the speed override ratio for the VelFactor input. If the specified value is 0.0, the relevant axis stops but it is not changed to 'StandStill' state.
(4) Specify acceleration/deceleration and override rate of jerk (change rate of acceleration) in AccFactor and JerkFactor input respectively.
(5) Negative number cannot be input in each Facotr, and if it is input, "error 0x10C1" occurs.
(6) Default of each override rate is 1.0, and it means 100% of the command speed of function block currently running.
(7) Override operation does not affect the serve axis of the relevant axis.

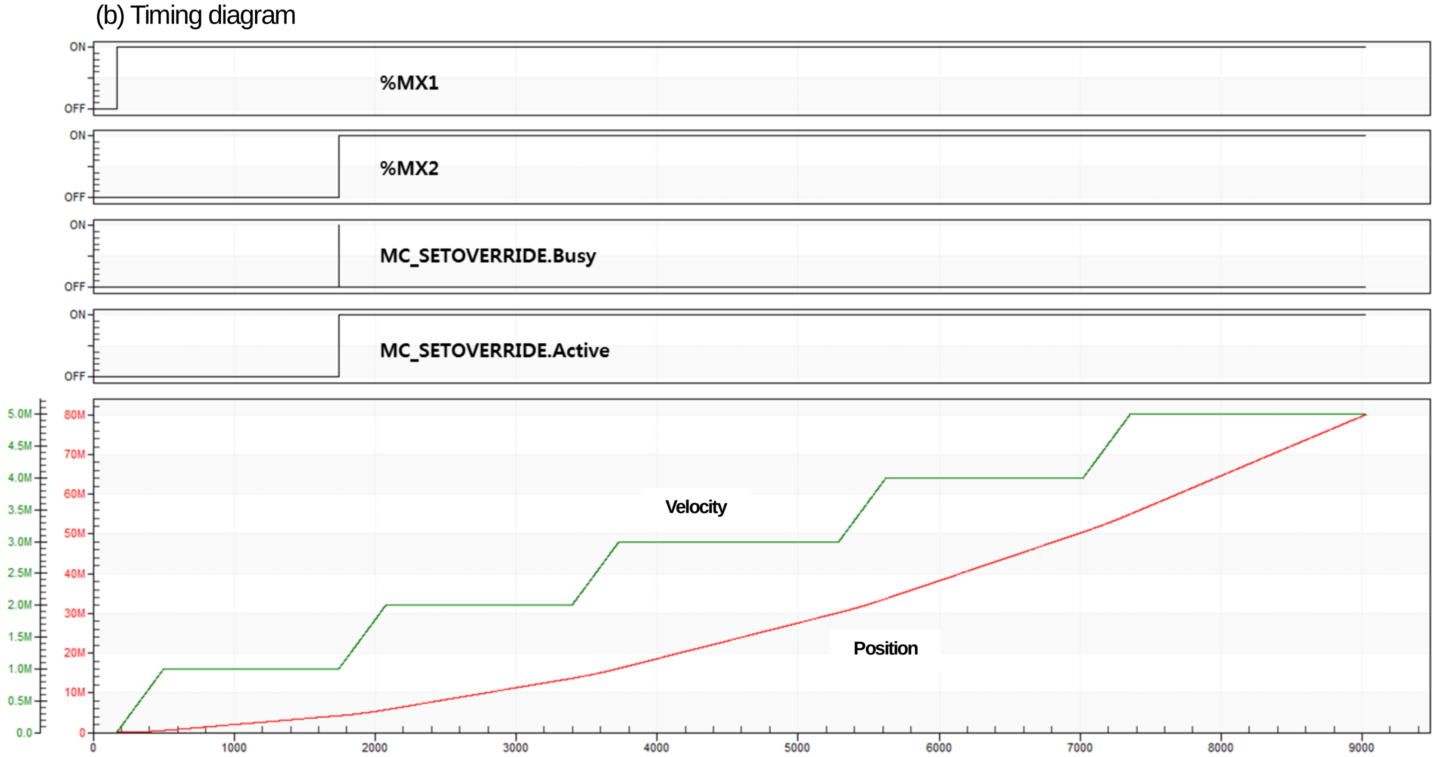
(8) Example program

This example shows the operation by changing the current velocity to 2,000,000/ 3,000,000/ 4,000,000/ 5,000,000 if VelFactor is changed to 2/3/4/5 at the current velocity of 1,000,000.

(a) Function block setting

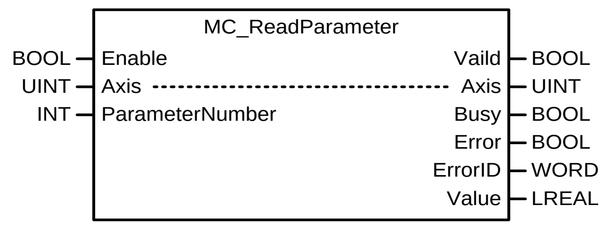


(b) Timing diagram



6.3.14 Read parameter (MC_ReadParameter)

Motion Function Block



Input-Output		
UINT	Axis	Specify the axis to be commanded (See 6.2.1 Setting Range by Product)
Input		
BOOL	Enable	The relevant parameters are output while the input is enabled.
INT	ParameterNumber	Specify the number of parameter to read.
Output		
BOOL	Vaild	Indicate whether the output of the current motion function block is valid.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.
LREAL	Value	Output the values of the parameters.

(1) This command is a motion function block which outputs parameter of the relevant axis.
(2) The value of the relevant parameter is continuously output in Value while Enable input is On.
(3) Specify the number of parameter to read in ParameterNumber input.

(4) The numbers of parameter are as below.

No.	Parameter	Item	Note	OS*Note 2)
0	Basic Setting	Unit	0: pulse, 1: mm, 2: inch, 3:degree	-
1		1 Pulse per revolution	1 ~ 4,294,967,295[pulse]	-
2		1 Travel per rotation	0.000000001 ~ 4,294,967,295 [Unit]	-
3		Speed command unit	0: Unit/sec, 1: Unit/min, 2: rpm	-
4		Speed limit value	LREAL Positive number [Unit/s, rpm] Change according to Unit, Pulses per rotation, Travel per rotation, Speed command unit	-
5		Emergency stop deceleration	0 or LREAL Positive number [Unit/s^2]	-
6		Encoder select	0:Incremental Encoder,1:Absolute Encoder	-
7		Gear ratio(Motor)	1 ~ 65,535	-
8		Gear ratio(Machine)	1 ~ 65,535	-
9		Operation mode of the reverse rotation	0: Deceleration stop, 1: Prompt stop	-
46		Position Control Range Expansion	0: Disable, 1:	-
47		Operation mode of Speed Control	0: CSP(Cyclic Sync. Position), 1: CSV(Cyclic Sync. Velocity)	-
48		Coordinate Systems Operation maximum allowable acceleration	0 or LREAL Positive number [Unit/s^2]	-
49		Coordinate Systems Operation maximum allowable deceleration	0 or LREAL Positive number [Unit/s^2]	-
10	Expansion Setting	S/W upper limit	LREAL*Note 1) [units]	-
11		S/W low limit	LREAL*Note 1) [units]	-
12		Infinite running repeat position	LREAL Positive number [Unit]	-
13		Infinite running repeat	0:disabled,1:enabled	-
14		Command Inposition range	0 or Long real (LREAL) positive number*1) [Unit]	-
15		Exceeding value of tracking error	0 or Long real (LREAL) positive number*1) [Unit]	-
16		Current position compensation amount	0 or Long real (LREAL) positive number*1) [Unit]	-
17		Current speed filter time constant	0 ~ 100	-
18		Error reset monitoring time	1 ~ 1000 [ms]	-
19		S/W limit during speed control	0: Not detect , 1 : detect	-
20		Tracking error level	0: Warning, 1: Alarm	-
21		Jog high speed	LREAL Positive number (Jog low speed ~speed limit) [Unit/s]	-
22		Jog low speed	LREAL Positive number (< Jog high speed) [Unit/s]	-
23		JOG Acceleration	0 or LREAL Positive number [Unit/s^2]	-
24		JOG Deceleration	0 or LREAL Positive number [Unit/s^2]	-
25		JOG Jerk	0 or LREAL Positive number [Unit/s^3]	-
26		Override mode	0: Specified by ratio, 1: Specified by unit	-
29		Backlash compensation amount	0 or Long real (LREAL) positive number*1) [Unit]	V1.30
27	NC Spindle Control Setting	Identifying range to reach the spindle rotation command speed	0~100%	
28		Identifying RPM to reach the spindle rotation zero speed	0 ~ 100rpm	
30	NC Spindle	Select the Spindle Encoder	0: Not Use, 1: Motor ENC, 2: Built-in ENC1, 3: Built-in ENC2, 4: EtherCAT ENC	V1.30

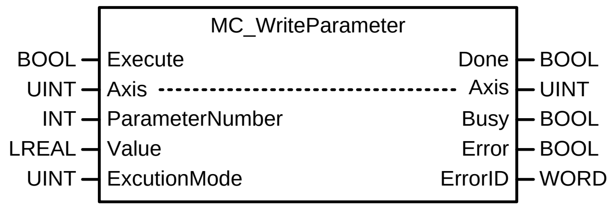
No.	Parameter	Item	Note	OS*Note 2)
31	Axis Setting	Number of pulses per rotation of the spindle EtherCAT encoder	1 ~ 4294967295	V1.30
32		Spindle EtherCAT encoder position variable	0: I device, 1: M device	V1.30
33		Spindle EtherCAT encoder position address	0~4095 (Spindle EtherCAT encoder position variable = 0: I) 0~524287 (Spindle EtherCAT encoder position variable = 1: M)	V1.30
34		The P Gain of the Spindle Positioning Mode	1~500 Hz	V1.30
35		The Feed Forward Gain of the Spindle Positioning Mode	0~100 %	V1.30
No.	**Parameter**	**Item**	**Note**	**OS**
36	NC Spindle Home Setting	How to conduct the homing operation	0: Servo drive supported, 33:Reverse direction, Z phase, 34: Forward direction, Z phase, 35: Set the homing of the current position	V1.30
37		Switch navigation speed of the homing operation	LREAL Positive number	V1.30
38		Zero navigation speed of the homing operation	Zero navigation speed of the origin operation ≤ Switch navigation speed of the origin ≤ Limit value of speed	V1.30
39		Acceleration/deceleration of the homing operation	0 or LREAL Positive number [Unit/s^2]	V1.30
40		Z phase variable	0: I device, 1: M device	V1.30
41		Z phase address	0~131071 (Z phase variable = 0: I) 0~16777215 (Z phase variable = 0: M)	V1.30
42		Orientation velocity	Long real (LREAL) positive number*note1) (≤ Limit value of speed)	V1.30
43		Orientation diredtion	0:forward direction , 1:reverse direction	V1.30
44		Orientation offset	0~360	V1.30
100	Built-in Encoder Setting	Encorder 1 unit	0: pulse, 1: mm, 2: inch, 3:degree	-
101		Encorder1 pulse per rotation	1 ~ 4,294,967,295[pulse]	-
102		Encorder1 travel per rotation	0.000000001 ~ 4,294,967,295 [Unit]	-
103		Encorder1 pulse input	0:CW/CCW 1 multiplier, 1:PULSE/DIR 1 multiplier 2: Pulse/Dir 2 multiplier, 3: Phase A/B 1 multiplier 4: Phase A/ B 2 multiplier, 5: Phase A/B 4 multiplier	-
104		Encorder1 max. value	(Encoder1 min. value + 1) ~ 2147483647	-
105		Encoder1 Min. value	-2147483648~(Encoder1 Max. value-1)	-
106		Encorder1 input filter value	0: Disable, 1:500kPPS 2: 200kPPS, 3: 100kPPS 4: 10kPPS, 5: 1kPPS 6: 0.1kPPS	-
107		Encorder1 Speed unit	0: Unit/sec, 1: Unit/min, 2: rpm	V1.10
108		Encorder1 position filter time constant	0 ~ 1000 ms	V1.10
109		Encoder 1 Position Latch	0: Disable, 1: Able	V1.40
200		Encorder 2 unit	0: pulse, 1: mm, 2: inch, 3:degree	-
201		Encorder 2 1 pulse per rotation	1 ~ 4,294,967,295[pulse]	-
202		Encorder 2 1 travel per rotation	0.000000001 ~ 4,294,967,295 [Unit]	-
203		Encorder2 pulse input	0:CW/CCW 1 multiplier, 1:PULSE/DIR 1 multiplier	-

No.	Parameter	Item	Note	OS*Note 2)
			2: Pulse/Dir 2 multiplier, 3: Phase A/B 1 multiplier 4: Phase A/ B 2 multiplier, 5: Phase A/B 4 multiplier	
204		Encorder2 max. value	(Encoder2 min. value + 2) ~ 2147483647	-
205		Encoder2 Min. value	-2147483648~(Encoder2 Max. value-1)	-
206		Encoder2 input filter value	0: Disable, 1:500kPPS 2: 200kPPS, 3: 100kPPS 4: 10kPPS, 5: 1kPPS 6: 0.1kPPS	-
207		Encoder2 Speed unit	0: Unit/sec, 1: Unit/min, 2: rpm	V1.10
208		Encoder2 position filter time constant	0 ~ 1000 ms	V1.10
209		Encoder 2 Position Latch	0: Disable, 1: Able	V1.40

*Note1) LREAL range: 2.2250738585072e-308 ~ 1.79769313486232e+308
LREAL positive range: 0 < x ≤1.79769313486232e+308
* note2) OS version is the version information to which the parameter is applied.

6.3.15 Write parameter (MC_WriteParameter)

Motion Function Block



Input-Output		
UINT	Axis	Specify the axis to be commanded (See 6.2.1 Setting Range by Product)
Input		
BOOL	Execute	Rising Edge corresponding parameters of input is written.
INT	ParameterNumber	Specify the number of parameter to write.
LREAL	Value	Specify the value of parameter to write.
UINT	ExecutionMode	Specify the time when parameter is written.
Output		
BOOL	Vaild	Indicate whether parameter is successfully written.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block is to write the value specified in parameter of the relevant axis.
(2) The parameters will be written in the rising edge of the Execute input.
(3) Specify the number of parameter to write in ParameterNumber input. The value unable to be set causes "error 0x10F0”.
(4) Specify the value to write in parameter for Value input.
(5) In ExecutionMode, correct the time when parameter is written and the values below can be set. The value unable to be set causes "error 0x101B”.
 0 (mcImmediately): Change the parameter value immediately after executing function block (rising Edge in Execute input). If the relevant axis is in running, operation can be affected.
 1 (mcQueued): Changed at the same point with ‘Buffered’ in Buffermode. (Refer to 6.1.4.BufferMode input)

(6) The numbers of parameter are as below.

No.	Parameter	Item	Note	OS
0	Basic Setting	Unit	0: pulse, 1: mm, 2: inch, 3:degree	-
1		1 Pulse per revolution	1 ~ 4,294,967,295[pulse]	-
2		1 Travel per rotation	0.000000001 ~ 4,294,967,295 [Unit]	-
3		Speed command unit	0: Unit/sec, 1: Unit/min, 2: rpm	-
4		Speed limit value	LREAL Positive number [Unit/s, rpm] Change according to Unit, Pulses per rotation, Travel per rotation, Speed command unit	-
5		Emergency stop deceleration	0 or LREAL Positive number [Unit/s^2]	-
6		Encoder select	0:Incremental Encoder,1:Absolute Encoder	-
7		Gear ratio(Motor)	1 ~ 65,535	-
8		Gear ratio(Machine)	1 ~ 65,535	-
9		Operation mode of the reverse rotation	0: Deceleration stop, 1: Prompt stop	-
46		Position Control Range Expansion	0: Disable, 1: Able	
47		Operation mode of Speed Control	0: CSP (Cyclic Sync. Position), 1: CSV (Cyclic Sync. Velocity)	
48		Coordinate Systems Operation maximum allowable acceleration	0 or LREAL Positive number [Unit/s^2]	
49		Coordinate Systems Operation maximum allowable deceleration	0 or LREAL Positive number [Unit/s^2]	
10	Expansion Setting	S/W upper limit	LREAL*Note 1) [units]	-
11		S/W low limit	LREAL*Note 1) [units]	-
12		Infinite running repeat position	LREAL Positive number [Unit]	-
13		Infinite running repeat	0:disabled,1:enabled	-
14		Command Inposition range	0 or Long real (LREAL) positive number*1) [Unit]	-
15		Exceeding value of tracking error	0 or Long real (LREAL) positive number*1) [Unit]	-
16		Current position compensation amount	0 or Long real (LREAL) positive number*1) [Unit]	-
17		Current speed filter time constant	0 ~ 100	-
18		Error reset monitoring time	1 ~ 1000 [ms]	-
19		S/W limit during speed control	0: Not detect , 1 : detect	-
20		Tracking error level	0: Warning, 1: Alarm	-
21		Jog high speed	LREAL Positive number (Jog low speed ~speed limit) [Unit/s]	-
22		Jog low speed	LREAL Positive number (< Jog high speed) [Unit/s]	-
23		JOG Acceleration	0 or LREAL Positive number [Unit/s^2]	-
24		JOG Deceleration	0 or LREAL Positive number [Unit/s^2]	-
25		JOG Jerk	0 or LREAL Positive number [Unit/s^3]	-
26		Override mode	0: Specified by ratio, 1: Specified by unit	-
29		Backlash compensation amount	0 or Long real (LREAL) positive number*1) [Unit]	V1.30
27	NC Spindle Control Setting	Identifying range to reach the spindle rotation command speed	0~100%	
28		Identifying RPM to reach the spindle rotation zero speed	0~ 100rpm	
30	NC Spindle	Select the Spindle Encoder	0: Not Use, 1: Motor ENC, 2: Built-in ENC1, 3: Built-in ENC2, 4: EtherCAT ENC	V1.30

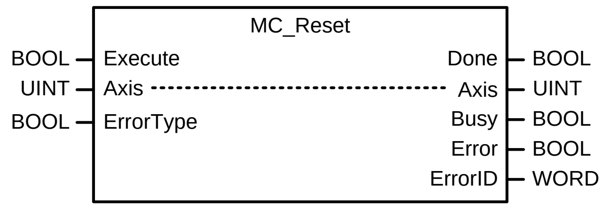
No.	Parameter	Item	Note	OS
31	Axis Setting	Number of pulses per rotation of the spindle EtherCAT encoder	1 ~ 4294967295	V1.30
32		Spindle EtherCAT encoder position variable	0: I device, 1: M device	V1.30
33		Spindle EtherCAT encoder position address	0~4095 (Spindle EtherCAT encoder position variable = 0: I) 0~524287 (Spindle EtherCAT encoder position variable = 1: M)	V1.30
34		The P Gain of the Spindle Positioning Mode	1~500 Hz	V1.30
35		The Feed Forward Gain of the Spindle Positioning Mode	0~100 %	V1.30
36	NC Spindle Home Setting	How to conduct the homing operation	0: Servo drive supported, 33:Reverse direction, Z phase, 34: Forward direction, Z phase, 35: Set the homing of the current position	V1.30
37		Switch navigation speed of the homing operation	LREAL Positive number	V1.30
38		Zero navigation speed of the homing operation	Zero navigation speed of the origin operation ≤ Switch navigation speed of the origin ≤ Limit value of speed	V1.30
39		Acceleration/deceleration of the homing operation	0 or LREAL Positive number [Unit/s^2]	V1.30
40		Z phase variable	0: I device, 1: M device	V1.30
41		Z phase address	0~131071 (Z phase variable = 0: I) 0~16777215 (Z phase variable = 0: M)	V1.30
42		Orientation velocity	Long real (LREAL) positive number[*note1] (≤ Limit value of speed)	V1.30
43		Orientation diredtion	0:forward direction , 1:reverse direction	V1.30
44		Orientation offset	0~360	V1.30
100	Built-in Encoder Setting	Encoder 1 unit	0: pulse, 1: mm, 2: inch, 3:degree	-
101		Encoder 1 1 pulse per rotation	1 ~ 4,294,967,295[pulse]	-
102		Encoder 1 1 travel per rotation	0.000000001 ~ 4,294,967,295 [Unit]	-
103		Encorder1 pulse input	0:CW/CCW 1 multiplier, 1:PULSE/DIR 1 multiplier 2: Pulse/Dir 2 multiplier, 3: Phase A/B 1 multiplier 4: Phase A/ B 2 multiplier, 5: Phase A/B 4 multiplier	-
104		Encorder1 max. value	(Encoder1 min. value + 1) ~ 2147483647	-
105		Encoder1 Min. value	-2147483648~(Encoder1 Max. value-1)	-
106		Encoder1 input filter value	0: Disable, 1:500kPPS 2: 200kPPS, 3: 100kPPS 4: 10kPPS, 5: 1kPPS 6: 0.1kPPS	-
107		Encoder1 Speed unit	0: Unit/sec, 1: Unit/min, 2: rpm	V1.10
108		Encoder1 position filter time constant	0 ~ 1000 ms	V1.10
109		Encoder 1 Position Latch	0: Disable, 1: Able	V1.40
200		Encorder 2 unit	0: pulse, 1: mm, 2: inch, 3:degree	-
201		Encorder 2 1 pulse per rotation	1 ~ 4,294,967,295[pulse]	-
202		Encorder 2 1 travel per rotation	0.000000001 ~ 4,294,967,295 [Unit]	-
203		Encorder2 pulse input	0:CW/CCW 1 multiplier, 1:PULSE/DIR 1 multiplier 2: Pulse/Dir 2 multiplier, 3: Phase A/B 1 multiplier 4: Phase A/ B 2 multiplier, 5: Phase A/B 4 multiplier	-

No.	Parameter	Item	Note	OS
204		Encorder2 max. value	(Encoder1 min. value + 2) ~ 2147483647	-
205		Encoder2 Min. value	-2147483648~(Encoder2 Max. value-1)	-
206		Encoder2 input filter value	0: Disable, 1:500kPPS 2: 200kPPS, 3: 100kPPS 4: 10kPPS, 5: 1kPPS 6: 0.1kPPS	-
207		Encoder2 Speed unit	0: Unit/sec, 1: Unit/min, 2: rpm	V1.10
208		Encoder2 position filter time constant	0 ~ 1000 ms	V1.10
209		Encoder 2 Position Latch	0: Disable, 1: Able	V1.40

*Note1) LREAL range: 2.2250738585072e-308 ~ 1.79769313486232e+308
LREAL positive range: 0 < x ≤1.79769313486232e+308

6.3.16 Reset axis error (MC_Reset)

Motion Function Block



Input-Output		
UINT	Axis	Specify the axis to be commanded (See 6.2.1 Setting Range by Product)
Input		
BOOL	Execute	Reset the axis error in the rising Edge of input.
BOOL	ErrorType	The types of error to be reset (0: Axis error, 1: Common error)
Output		
BOOL	Done	Indicates whether the axis error was successfully reset
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

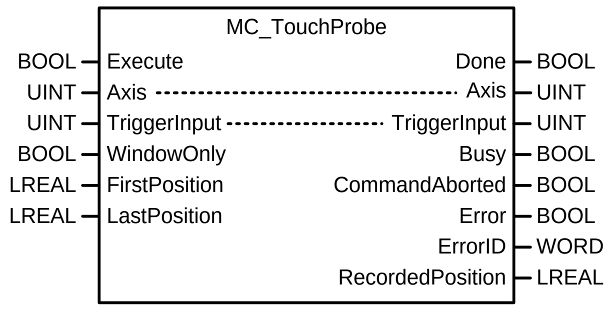
(1) This motion function block is to reset the error of the relevant axis. When setting ErrorType to '0' and executing motion function block in case the relevant axis is in ' ErrorStop' state, every axis error is reset and the axis state is switched to 'StandStill' or 'Disabled' state.

(2) If ErrorType is set to ‘1’ and motion function block is executed,
Common error occurred in the relevant module is reset.

(3) Motion function block is executed in the rising Edge of Execute input.

6.3.17 Touch probe (MC_TouchProbe/MC_TouchProbeEx)

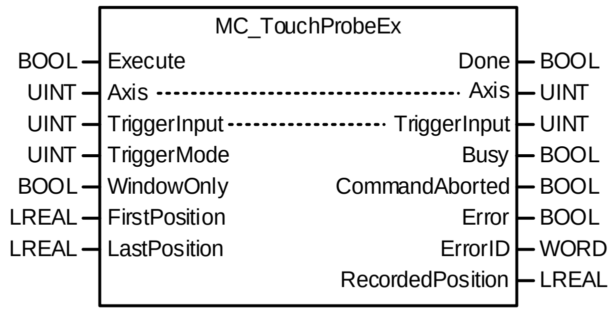
Motion Function Block



Input-Output		
UINT	Axis	Specify the axis to be commanded (See 6.2.1 Setting Range by Product, Real axis)
UINT	TriggerInput	Specify the signal to be used as a trigger. (0: Touch probe 1 rising edge, 1: Touch probe 2 rising edge)
Input		
BOOL	Execute	TouchProbe function starts at the rising Edge of input.
BOOL	WindowOnly	Activate the window mode.
LREAL	FirstPosition	Specify the starting position of allowable area in the window mode.
LREAL	LastPosition	Specify the end position of allowable area in the window mode.
Output		
BOOL	Done	Indicate that the trigger signal is successfully recorded.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted while it is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.
Read	RecordedPosition	Output the axis position where the trigger occurs.

(1) This motion function block is to execute 'TouchProbe' function which records the axis position at the time when the trigger event occurs.
(2) TouchProbe function starts at the rising Edge of Execute input.
(3) Specify the signal to be used as a trigger in TriggerInput. The value unable to be set causes "error 0x10E1".
(4) If the touch probe command satisfies the trigger condition, the Done output is On, and the position value at the trigger point is saved in the RecordedPosition output. The RecordedPosition output is not initialized and retains the previously stored value even if the Execute input changes to On or Off or an error occurs. The RecordedPosition output is updated when the touch probe command is re-executed to normally satisfy the trigger condition.
(5) When activating the window mode, allowable area where accepts the trigger signal of axis can be set. Operation timing of each signal when the window mode is activated is as below.

Motion Function Block



Input-Output		
UINT	Axis	Specify the axis to be commanded (See 6.2.1 Setting Range by Product, Real axis)
UINT	TriggerInput	Specify the signal to be used as a trigger. (0: Touch probe 1 rising edge, 1: Touch probe 2 rising edge, 2: Touch probe 1 falling edge, 3: Touch probe 2 falling edge) 4: Touch probe 1 index[Enc. Z] pulse, 5: Touch probe 2 index[Enc. Z] pulse)
Input		
BOOL	Execute	TouchProbe function starts at the rising Edge of input.
UINT	TriggerMode	Set the trigger mode. (0: single trigger, 1 : continuous trigger)
BOOL	WindowOnly	Activate the window mode.
LREAL	FirstPosition	Specify the starting position of allowable area in the window mode.
LREAL	LastPosition	Specify the end position of allowable area in the window mode.
Output		
BOOL	Done	Indicate that the trigger signal is successfully recorded.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted while it is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.
REAL	RecordedPosition	Output the axis position where the trigger occurs.

(1) This motion function block is an extended touch probe function block that includes the functions of the existing touch probe (MC_TouchProbe).

(2) Expansion TouchProbe function starts at the rising Edge of Execute input.

(3) Specify the signal to be used as a trigger in TriggerInput. The input range (0~1) of the existing (MC_TouchProbe) function block and the input range (0~5) of the extended function touch probe (MC_TouchProbeEx) function block are different. The value unable to be set causes "error 0x10E2".

(4) When setting the touch probe index [Enc.Z] pulse (4, 5) in TriggerInput, the index pulse rising edge signal of the encoder installed in the motor of the axis to be commanded is used as the position recording trigger. Therefore, the touch probe 1 index [Enc. Z] Pulse and touch probe 2 index [Enc. Z] pulse records the same data and selects whether to use touch probe 1 or touch probe 2 for the data storage channel.

(5) TriggerMode specifies the trigger mode. The value unable to be set causes "error 0x10E3". If TriggerMode is set to continuous trigger, position recording continues according to the signal set in TriggerInput until execution is stopped with the MC_AbortTrigger function block.

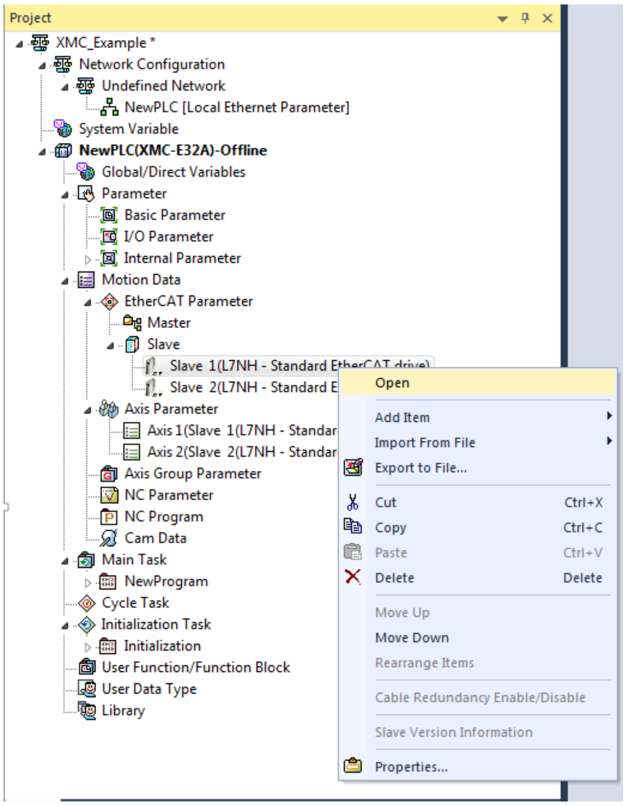
(6) Similar to the MC_TouchProbe function block, the setting can be canceled with the MC_AbortTrigger function block.

(7) When activating the window mode, allowable area where accepts the trigger signal of axis can be set. Operation timing of each signal when the window mode is activated is as below.

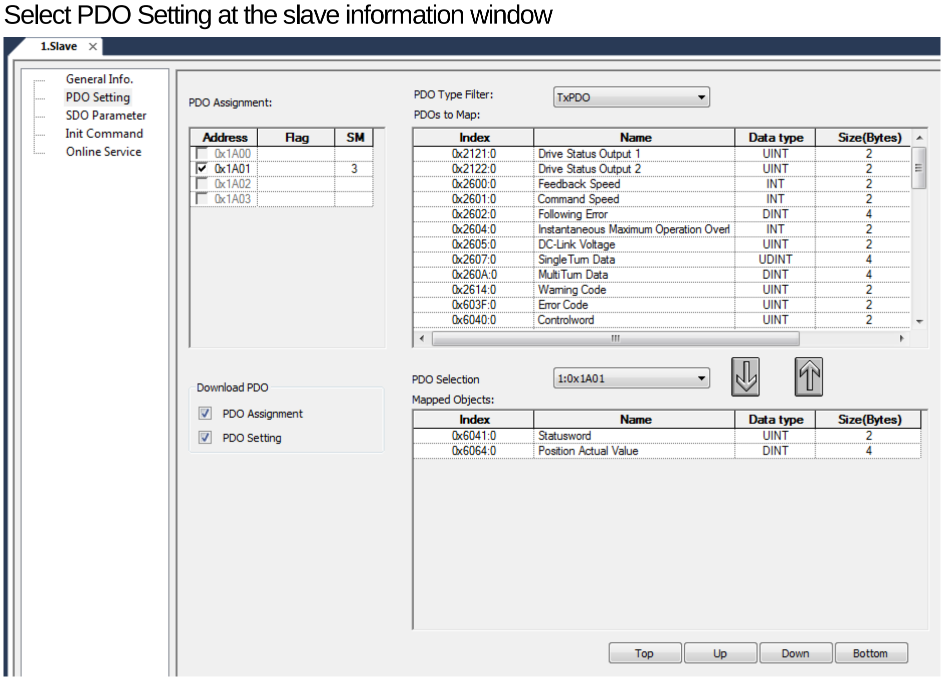
Notes

In the case of using Touch Probe, please set the slave parameters before use.

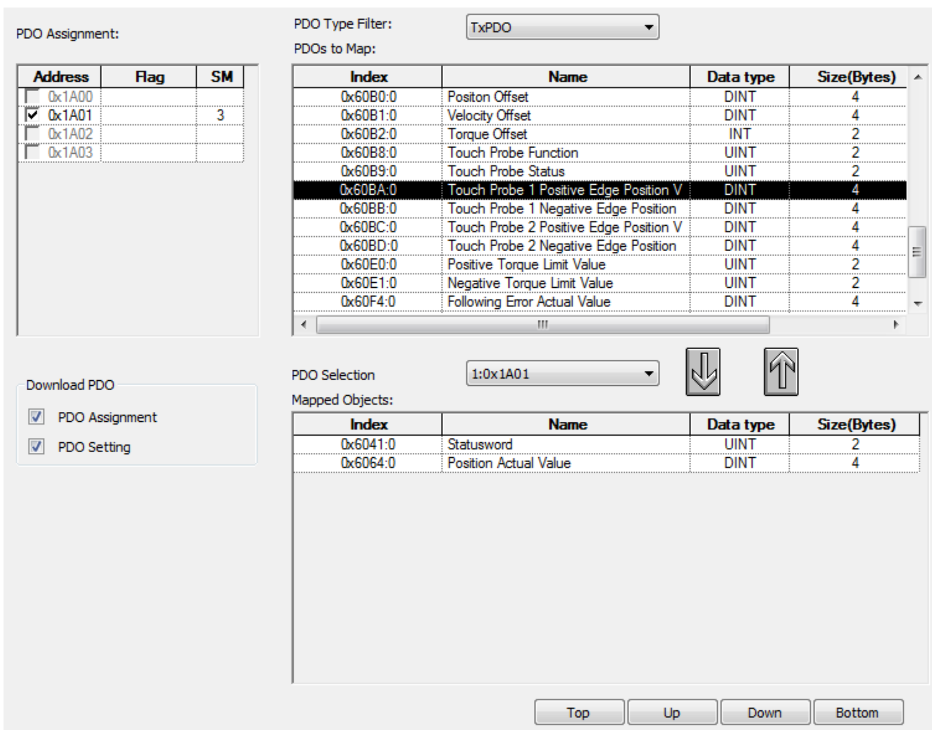
1. At XG50000, click the registration information of the servo drive.



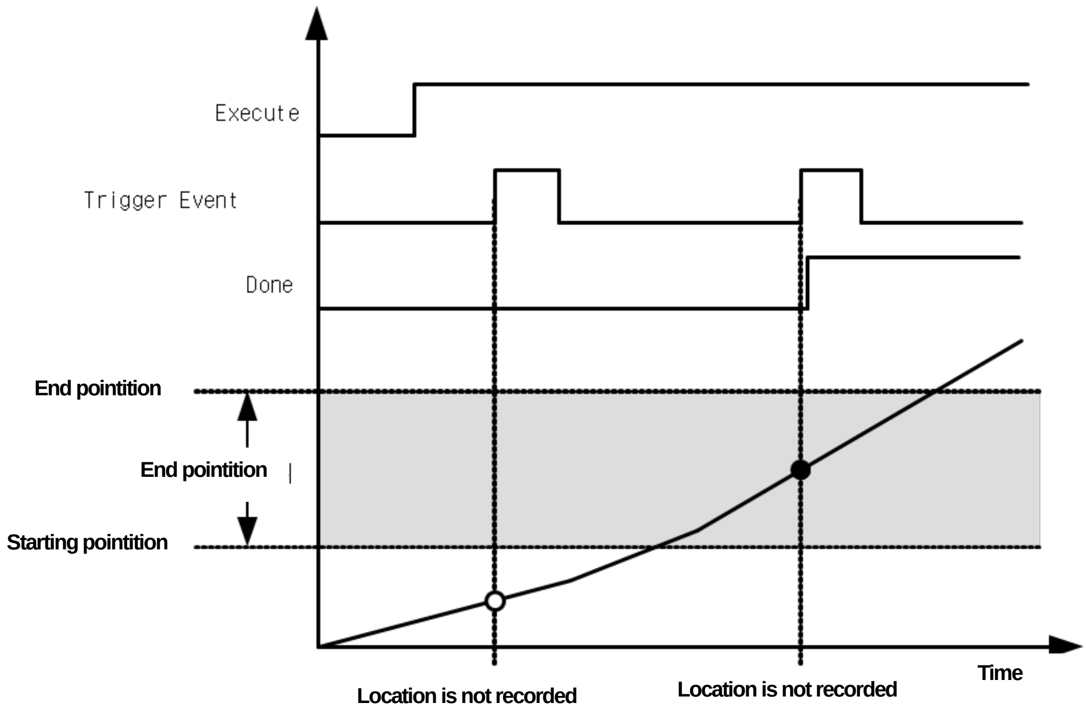
2. Select PDO Setting at the slave information window



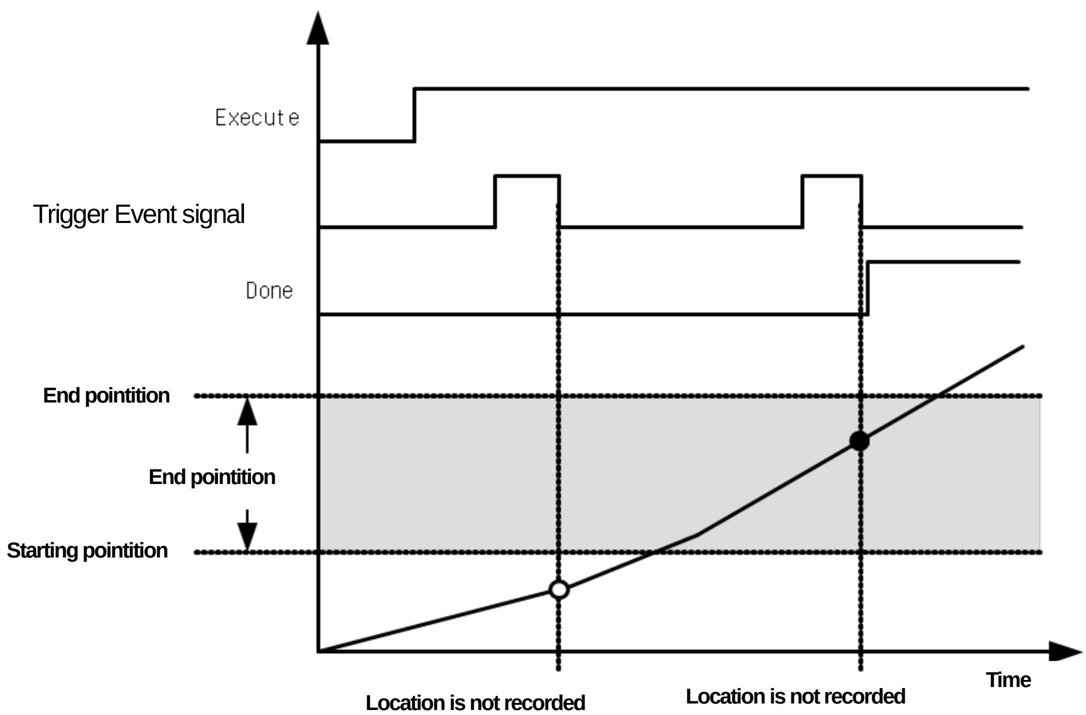
3. Select Touch Probe item in the edit window and click the arrow (downward), and include it in the PDO communication data. Touch Probe related PDO item are as follows.
   1) RxPDO
      Touch probe function(0x60B8)
   2) TxPDO
      Touch probe Status(0x60B9)
      Touch Probe 1 forward direction position value (0x60BA)
      Touch Probe 1 backward direction position value (0x60BB)
      Touch Probe 2 forward direction position value (0x60BC)
      Touch Probe 2 backward direction position value (0x60BD)



4. After PDO item is edited, must write ‘EtherCAT parameter’ in motion controller.

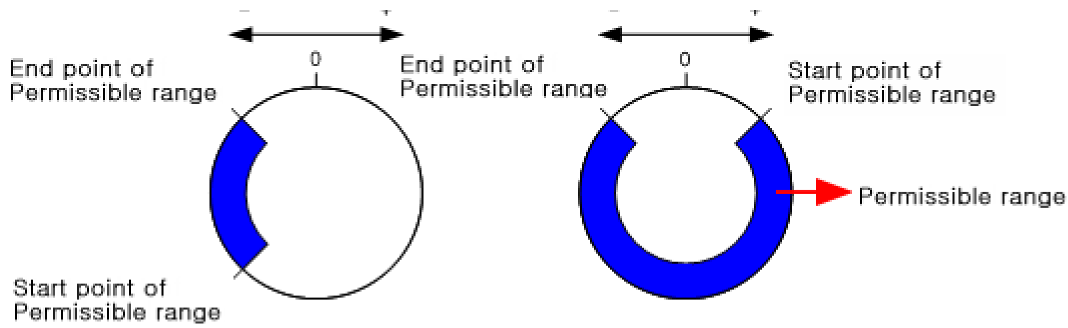


< When the TriggerInput input value is the rising edge of the touch probe and the touch probe function is in window mode, the operation timing >

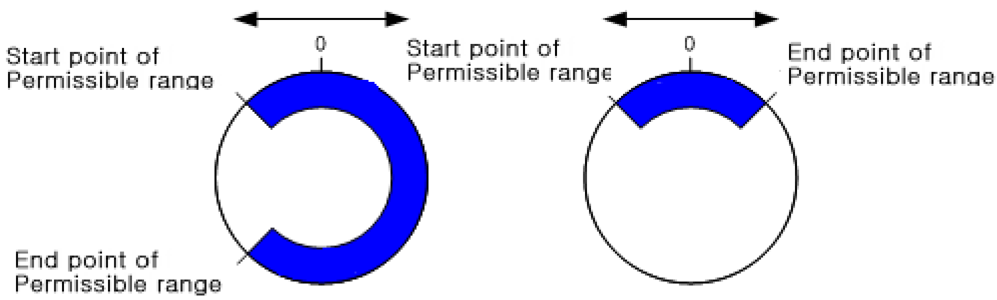


< When the TriggerInput input value is the falling edge of the touch probe and the touch probe function is in window mode, the operation timing >

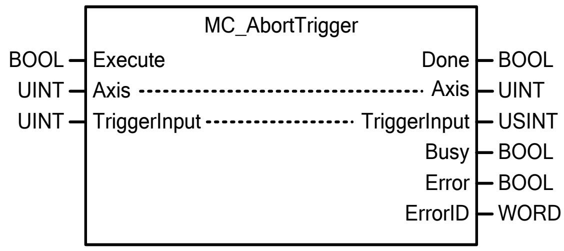
● In case of Permissible range start point < Permissible range end point



● In case of Permissible range start point > Permissible range end point



6.3.18 Abort trigger events (MC_AbortTrigger)

Motion Function Block		
		
Input-Output		
UINT	Axis	Specify the axis to be commanded. (See 6.2.1 Setting Range by Product, Real axis)
UINT	TriggerInput	Specify the trigger signal to be disengaged. (0: Touch probe 1, 1: Touch probe 2)
Input		
BOOL	Execute	The trigger on standby in the relevant axis in the rising Edge is disengaged.
Output		
BOOL	Done	Indicate the completion state of motion function block.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block is to disengage the trigger which is on standby in the relevant axis.
(2) Specify the trigger signal to be disengaged in TriggerInput. The value unable to be set causes "error 0x10E1".

6.3.19 SuperImposed operation (MC_MoveSuperImposed)

Motion Function Block

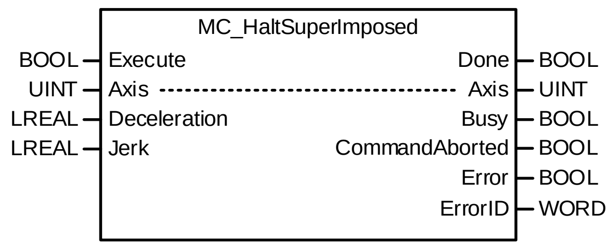
	MC_MoveSuperImposed		
BOOL	Execute	Done	BOOL
UINT	Axis ----------------------------------	Axis	UINT
BOOL	ContinuousUpdate	Busy	BOOL
LREAL	Distance	Active	BOOL
LREAL	VelocityDiff	CommandAborted	BOOL
LREAL	Acceleration	Error	BOOL
LREAL	Deceleration	ErrorID	WORD
LREAL	Jerk	CoveredDistance	LREAL

Input-Output		
UINT	Axis	Specify the axis to be commanded (See 6.2.1 Setting Range by Product)
Input		
BOOL	Execute	Give a SuperImposed operation command to the relevant axis in the rising Edge.
BOOL	ContinuousUpdate	Specify the update setting of input value. (Refer to 6.1.5.Changes in Parameters during Execution of Motion Function Block)
LREAL	Distance	Specify the target distance. [u]
LREAL	VelocityDiff	Specify the added velocity. [u/s]
LREAL	Acceleration	Specify the added acceleration. [u/s^2]
LREAL	Deceleration	Specify the added deceleration. [u/s^2]
LREAL	Jerk	Specify the added change rate of acceleration/deceleration. [u/s^3]
Output		
BOOL	Done	Indicate whether to reach the specified distance.
BOOL	Busy	Indicate that execution of the function block is not completed.
BOOL	Active	Indicate that the current motion function block is controlling the relevant axis.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted by other command
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the error number that occurred while the function block is running.
LREAL	CoveredDistance	Indicate the distance moved with SuperImposed operation after SuperImposed command.

(1) This motion function block is a command issuing SuperImposed operation order to the relevant axis.
(2) SuperImposed is a command ordering to move from the current position at the time of the command to the target distance set by Distance input.
(3) The direction of the movement is determined by the positivity/negativity of the set distance. Positive distance (+ or no sign) means forward movement, and negative distance (-) means reverse movement.
(4) After moving the target distance, when the velocity reaches 0, the command is completed and Done output is on.

6.3.20 SuperImposed operation Stop ((MC_HaltSuperImposed)

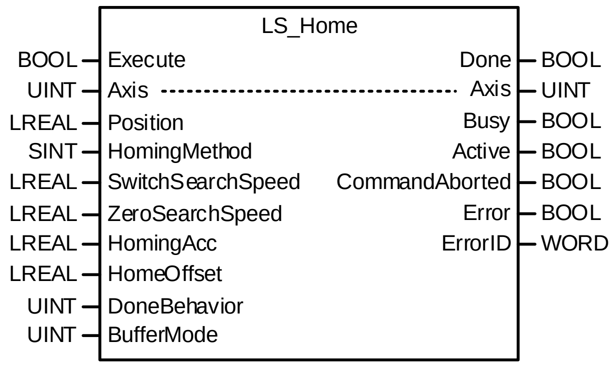
Motion Function Block		



Input-Output		
UINT	Axis	Specify the axis to be commanded (See 6.2.1 Setting Range by Product)
Input		
BOOL	Execute	Give a SuperImposed operation halt command to the relevant axis in the rising Edge.
LREAL	Deceleration	Specify deceleration in time of stop. [u/s^2]
LREAL	Jerk	Specify the change rate of acceleration/deceleration. [u/s^3]
Output		
BOOL	Done	Indicates that SuperImposed operation of the relevant axis is stopped.
BOOL	Busy	Indicate that execution of the function block is not completed.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted while it is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the error number that occurred while the function block is running.

(1) This motion function block is a command issuing an order to halt superImposed operation to the relevant axis.
(2) Halt command for SuperImposed operation is a command ordering to decelerate and halt at a given acceleration and jerk at the time of performing the command.
(3) After moving the target distance, when the velocity reaches 0, the command is completed and Done output is on.

6.3.21 Perform the search expansion home (LS_Home)

Motion Function Block		
		
Input-Output		
UINT	Axis	Specify the axis to be commanded (See 6.2.1 Setting Range by Product)
Input		
BOOL	Execute	Start the homing operation in rising Edge.
LREAL	Position	Specify the absolute position of axis when reference signal is detected. (When the command position value is converted to a pulse, it is set within the range of -2147483648 to 2147483647.)
SINT	HomingMethod	Set the homing method (0x6098)
LREAL	SwitchSearchSpeed	Set the switch search speed (0x6099:1) when returning to the homing.
LREAL	ZeroSearchSpeed	Set the zero search speed (0x6099:2) when returning to the homing.
LREAL	HomingAcc	Set acceleration/deceleration (0x609A) when returning to homing.
LREAL	HomeOffset	Set Offest (0x607C) when returning to homing.
UINT	DoneBehavior	Set the movement (0x201E) after homing is completed. (Used by drives other than our own L7N, the value has no effect on third-party drives)
UINT	BufferMode	Specify the sequential operation setting of motion function block. (Refer to 6.1.4 BufferMode input)
Output		
BOOL	Done	Indicate the completion state of motion function block.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that the current motion function block is controlling the relevant axis.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted by other command.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block is to give a expansion homing command to the relevant axis.
(2) The homing method operates as set in the function block. (The homing method may be different for each drive, so please refer to the user manual of the drive you are using. When using our XDL-N series servo drive, please refer to 8.1 Origin Determination)
(3) As for Position input, absolute position of axis is specified when Reference Signal is detected or homing is completed.
(4) While this motion function block is running, the axis is 'Homing' state, and when the command is completed, it is switched to 'Standstill'.
(5) If the function block is re-executed (Execute input is On) before the instruction is terminated, the changed parameters are

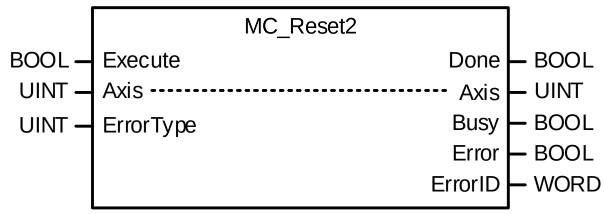
applied.
Only Position input can be updated.

(6) The input range for SwitchSearchSpeed, ZeroSearchSpeed, and HomingAcc is 0 ≤ Parameter ≤ 1073741824 in pulse units, and the input range for HomeOffset is -2147483648≤ Parameter ≤2147483647 in pulse units. If a value is input in a unit other than pulse unit, it is converted to pulse unit and checked if the value is out of range, and an error occurs if it is out of range.

(7) In the case of DoneBehavior, this is a function to set the movement method after homing is completed. It is used for drives other than our own L7N, and the value set in the function block does not affect when a third-party drive is used.
0: After homing is completed, the motor does not rotate and the Home Offset value becomes Zero Position.
1: After homing is completed, the motor moves as much as the Home Offset value, and it becomes the Zero Position at the position where the movement is completed.

6.3.22 Reset axis error 2(MC_Reset2)

Motion Function Block

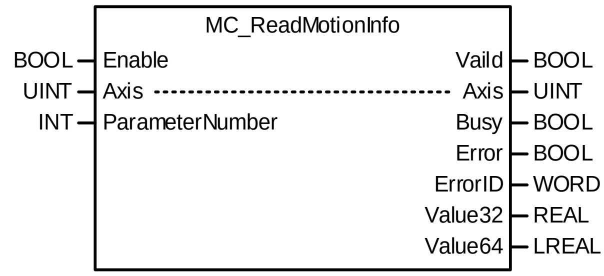


Input-Output		
UINT	Axis	Specify the axis to be commanded (See 6.2.1 Setting Range by Product)
Input		
BOOL	Execute	Reset the axis error in the rising Edge of input.
UINT	ErrorType	The types of error to be reset (0: Axis error, 1: Common error, 2: Axis error and Common error)
Output		
BOOL	Done	Indicates whether the axis error was successfully reset
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block is to reset the error of the relevant axis. When setting ErrorType to '0' and executing motion function block in case the relevant axis is in ' ErrorStop' state, every axis error is reset and the axis state is switched to 'StandStill' or 'Disabled' state.
(2) If ErrorType is set to ‘1’ and motion function block is executed, Common error occurred in the relevant module is reset. If ErrorType is set to ‘2’ and motion function block is executed, errors and common errors occurring in the set axis are reset.
(3) Motion function block is executed in the rising Edge of Execute input.
(4) “Error 0x1023” occurs when a value greater than 2 is entered for ErrorType
(5) BufferMode can be selected, unlike MC_Stop Halt command (MC_Halt) can be stopped by another motion function block.

6.3.23 Read Motion Information (MC_ReadMotionInfo)

Motion Function Block		



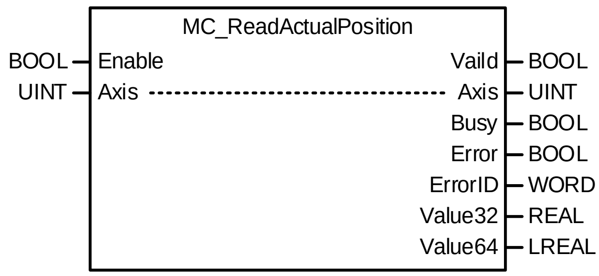
Input-Output		
UINT	Axis	Specify the axis to be commanded (See 6.2.1 Setting Range by Product)
Input		
BOOL	Enable	The relevant parameters are output while the input is enabled.
INT	ParameterNumber	Specify the number of parameter to read.
Output		
BOOL	Valid	Indicate whether the output of the current motion function block is valid.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.
REAL	Value32	Outputs the parameter value as a real number.
LREAL	Value64	Outputs the parameter value as a Long Real Number.

(1) This command is a motion function block which outputs motion information of the relevant axis.
(2) While Enable input is On, the value of the item set by ParameterNumber is continuously output to Value32 and Value64.
(3) Specify the number of item to read in ParameterNumber input. The number is as follows.

Item	ParameterNumber	Unit
Current position	**0**	Position display unit set on the commanded axis
Command position	**1**	Position display unit set on the commanded axis
Current Velocity	**2**	Speed display unit set on the commanded axis
Command speed	**3**	Speed display unit set on the commanded axis
Current Torque	**4**	Rated torque %
Command Torque	**5**	Rated torque %

6.3.24 Read current Position (MC_ReadActualPosition)

Motion Function Block

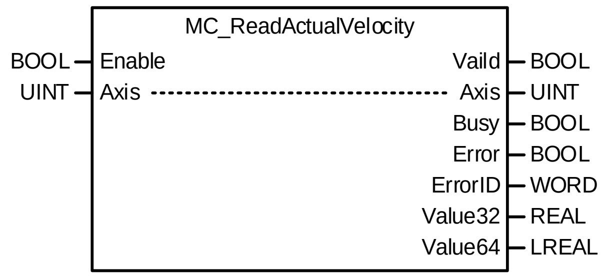


Input-Output		
UINT	Axis	Specify the axis to be commanded (See 6.2.1 Setting Range by Product)
Input		
BOOL	Enable	The relevant parameters are output while the input is enabled.
Output		
BOOL	Valid	Indicate whether the output of the current motion function block is valid.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.
REAL	Value32	Outputs the current position value as a real number.
LREAL	Value64	Outputs the current position value as a Long Real Number.

(1) This command is a motion function block which outputs current position of the relevant axis.
(2) While Enable input is On, the current position value is continuously output to Value32 and Value64 in the unit of position set on the axis.

6.3.25 Read current Velocity (MC_ReadActualVelocity)

Motion Function Block

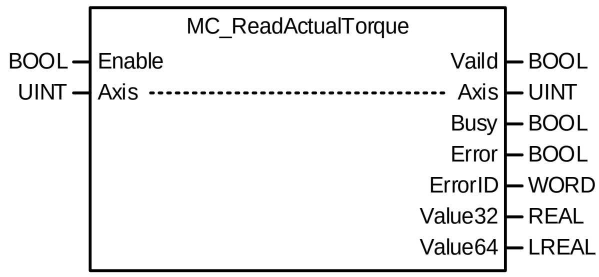


Input-Output		
UINT	Axis	Specify the axis to be commanded (See 6.2.1 Setting Range by Product)
Input		
BOOL	Enable	The relevant parameters are output while the input is enabled.
Output		
BOOL	Valid	Indicate whether the output of the current motion function block is valid.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.
REAL	Value32	Outputs the current speed value as a real number.
LREAL	Value64	Outputs the current speed value as a Long Real Number.

(1) This speed is a motion function block which outputs current speed of the relevant axis.
(2) While Enable input is On, the current speed value is continuously output to Value32 and Value64 in the unit of speed set on the axis.

6.3.26 Read current torque (MC_ReadActualTorque)

Motion Function Block

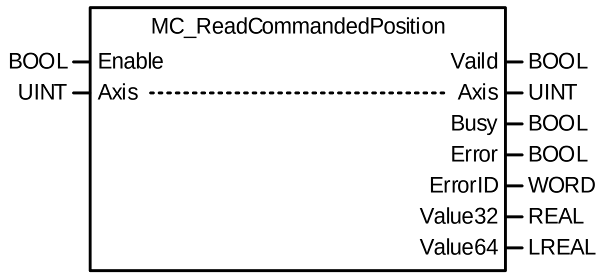


Input-Output		
UINT	Axis	Specify the axis to be commanded (See 6.2.1 Setting Range by Product)
Input		
BOOL	Enable	The relevant parameters are output while the input is enabled.
Output		
BOOL	Valid	Indicate whether the output of the current motion function block is valid.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.
REAL	Value32	Outputs the current torque value as a real number.
LREAL	Value64	Outputs the current torque value as a Long Real Number.

(1) This command is a motion function block which outputs current torque of the relevant axis.
(2) While the Enable input is On, the current torque value is continuously output as a % value of the rated torque in Value32 and Value 64.
(3) If the actual torque value (0x6077) is not included in the servo drive TxPDO setting, the value is not displayed normally.

6.3.27 Read command Position (MC_ReadCommandedPosition)

Motion Function Block



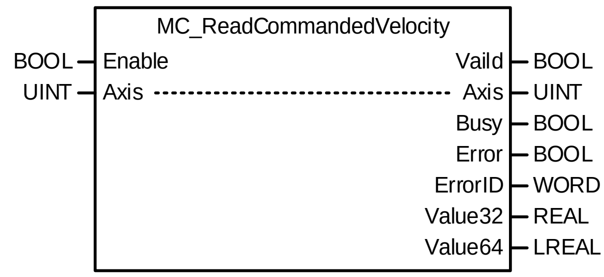
Input-Output		
UINT	Axis	Specify the axis to be commanded (See 6.2.1 Setting Range by Product)
Input		
BOOL	Enable	The relevant parameters are output while the input is enabled.
Output		
BOOL	Valid	Indicate whether the output of the current motion function block is valid.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.
REAL	Value32	Outputs current position value as a real number.
LREAL	Value64	Outputs command position value as a Long Real Number.

(1) This command is a motion function block which outputs command position of the relevant axis.

(2) While Enable input is On, the command position value is continuously output to Value32 and Value64 in the unit of position set on the axis.

6.3.28 Read command Velocity (MC_ReadCommandedVelocity)

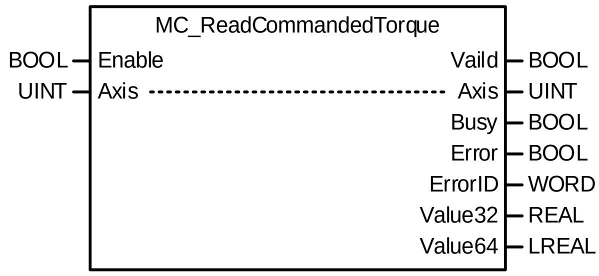
Motion Function Block



Input-Output		
UINT	Axis	Specify the axis to be commanded (See 6.2.1 Setting Range by Product)
Input		
BOOL	Enable	The relevant parameters are output while the input is enabled.
Output		
BOOL	Valid	Indicate whether the output of the current motion function block is valid.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.
REAL	Value32	Outputs command speed value as a real number.
LREAL	Value64	Outputs command position value as a Long Real Number.

(1) This command is a motion function block which outputs command speed of the relevant axis.
(2) While Enable input is On, the command speed value is continuously output to Value32 and Value64 in the unit of speed set on the axis.

6.3.29 Read command torque (MC_ReadCommandedTorque)

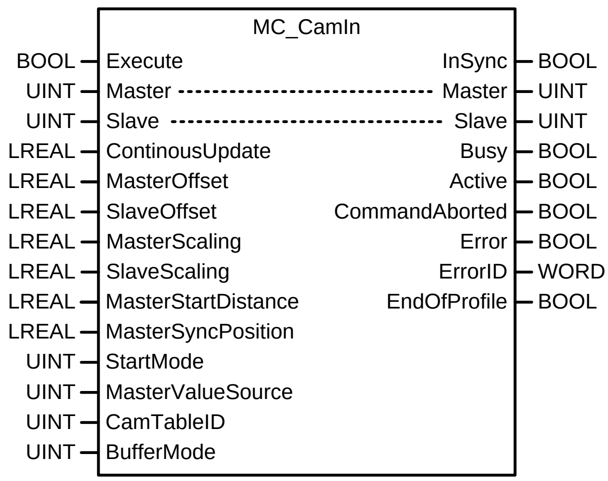
Motion Function Block		
		
Input-Output		
UINT	Axis	Specify the axis to be commanded (See 6.2.1 Setting Range by Product)
Input		
BOOL	Enable	The relevant parameters are output while the input is enabled.
Output		
BOOL	Valid	Indicate whether the output of the current motion function block is valid.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.
REAL	Value32	Outputs command torque value as a real number.
LREAL	Value64	Outputs command torque value as a Long Real Number.

(1) This command is a motion function block which outputs command torque of the relevant axis.
(2) While the Enable input is On, the command torque value is continuously output as a % value of the rated torque in Value32 and Value 64.
(3) If the actual torque value (0x6077) is not included in the servo drive TxPDO setting, the value is not displayed normally.

6.4 Multi-Axis Motion Function Block

6.4.1 Cam operation (MC_CamIn)

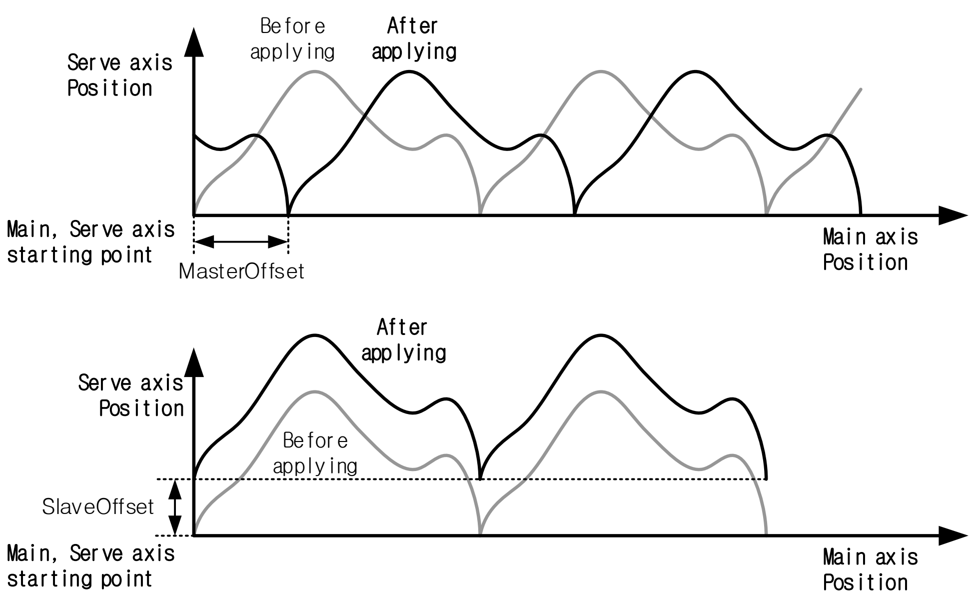
Motion Function Block



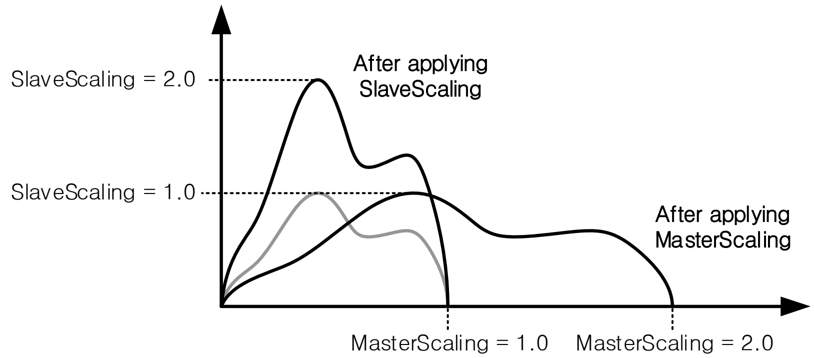
Input-Output		
UINT	Master	Set main axis (See 6.2.1 Setting Range by Product), Available to set variables only
UINT	Slave	Set the serve axis. (See 6.2.1 Setting Range by Product), Available to set variables only
Input		
BOOL	Execute	Give cam operation command to the relevant axis in the rising Edge.
BOOL	ContinuousUpdate	Specify the update setting of input value. Reference
LREAL	MasterOffset	Set the offset value of the main axis.
LREAL	SlaveOffset	Set the offset value of the slave cam table.
LREAL	MasterScaling	Specify the magnification of the main axis.
LREAL	SlaveScaling	Specify the magnification of the slave axis cam table.
LREAL	MasterStartDistance	Specify the position of the main-axis where the cam operation of the master axis starts.
LREAL	MasterSyncPosition	Specify the starting point at cam table when cam operation starts.
UINT	StartMode	Set the cam operation mode. 0: Cam table is applied as an absolute value. (mcAbsolute) 1: Cam table is applied as a relative value based on the command start position. (mcRelative)
UINT	MasterValueSource	Select the source of the main axis for cam operation. 0: Synchronizes to the command position of the main axis. 1: Synchronizes to the current position of the main axis.
UINT	CamTableID	Specify the cam table to operate. (refer toSee 6.2.1 Setting Range by Product)
UINT	BufferMode	Specify the sequential operation setting of motion function block. (Refer to 6.1.4 BufferMode input)

Output		
BOOL	InSync	Indicate that cam operation is normally being fulfilled. (Indicate that the serve axis is following the cam table.)
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that the current motion function block is controlling the relevant axis.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted while it is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

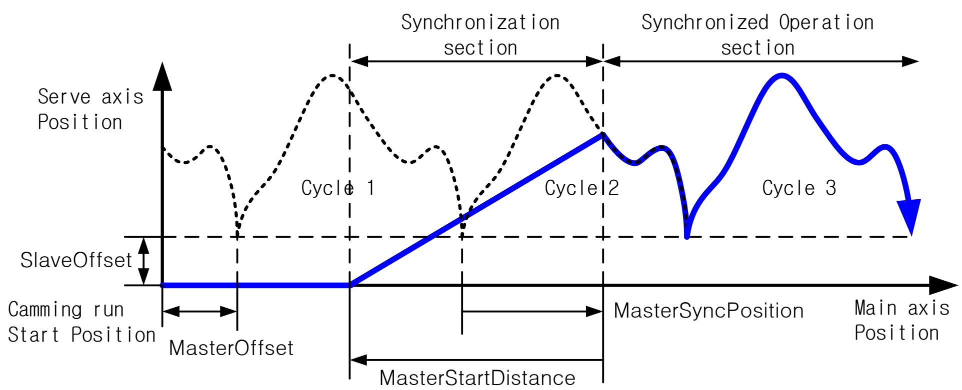
(8) This motion function block is to operate the serve axis cam depending on the main axis.
(9) Cam operation command can be given to the serve axis even if the main axis is in stop state.
(10) To halt cam operation, MC_CamOut command should be issued on the sub-axis, or another motion function block should be operated (in case of Aborting).
(11) If this motion function block is aborted by another command (BufferMode=0 of newly executed command), the cam operation is stopped, and the CommandAborted output is on.
(12) If another command is executed by Buffered while this motion function block is being executed (BufferMode=1~5 of newly executed command), the operation of the cam profiled cycle is terminated, and then the newly executed command is run subsequently. InSync / Busy / Active / CommandAborted / Error output of MC_CamIn function block are all Off.
(13) The axis is in 'Synchronized Motion' while this motion function block is running.
(14) Set the offset of the cam table to be applied in MasterOffset and SlaveOffset. MasterOffset determines the offset from the master axis start point, and Slaveoffset determines the offset from the slave axis start point. Refer to the Figure below. Using offset may change the start position for cam operation, causing an abrupt operation. In such a case, MasterSyncPosition, MasterStartDistance should be used.



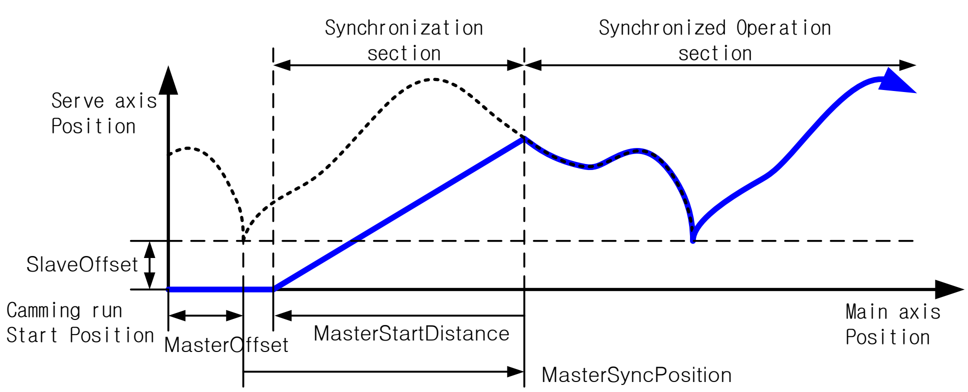
(15) Set the magnification of cam data to be applied in MasterScaling and SlaveScaling. MasterScaling determines the scale rate of the main-axis data, and SlaveScaling determines the scale rate of the sub-axis data. Refer to the Figure below.



(16) MasterSyncPosition specifies the position of the master axis within the table where the synchronization of actual cam operation is completed, and MasterStartDistance specifies the relative position of the master axis where the synchronization starts. Synchronization starts at a position as far away as MasterStartDistance from MasterSyncPosition. If unable to start synchronized operation at Cycle 1 as shown below (if the distance from the start position to the synchronized operation start position is shorter than MasterStartDistance), synchronized operation starts at Cycle 2.



<When MasterScaling is 1.0>

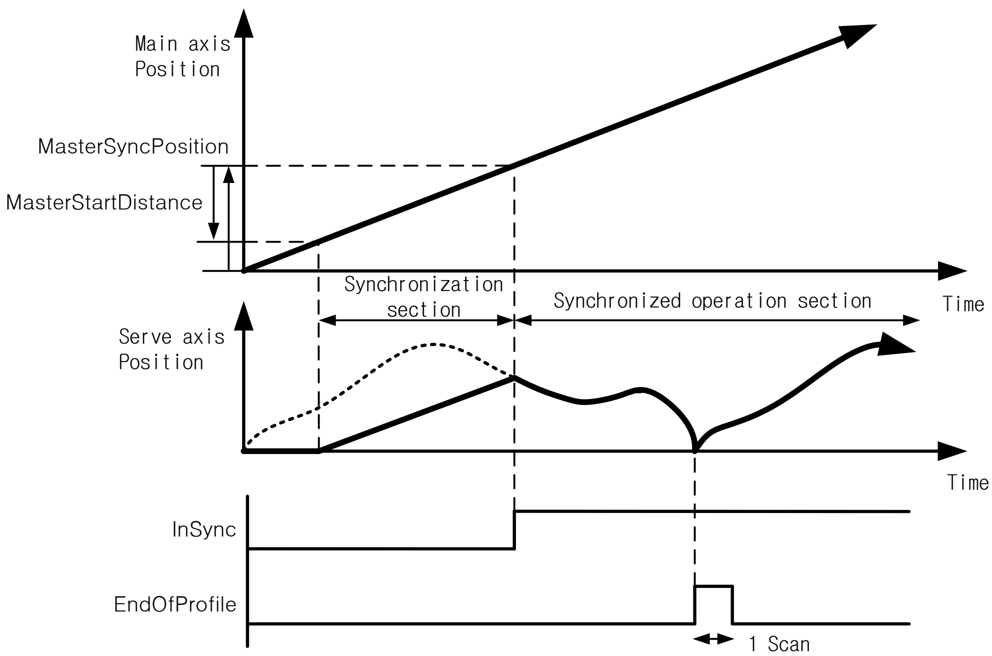


<When MasterScaling is 2.0>

Actual synchronization position can vary depending on MasterScaling and SlaveScaling because MasterSyncPosition is a value based on the inside of cam table, but MasterOffset and MasterStartDistance value remain unaffected.

(9) The changed parameters can be applied when ContinuousUpdate input is On. Only MasterOffset, SlaveOffset, MasterScaling, SlaveScaling, MasterStartDistance, MasterSyncPosition can be updated (However, In InSync=On case, MasterOffset, SlaveOffset, MasterScaling, SlaveScaling can be updated.)

(10) Once cam operation starts normally, InSync output is On, and EndOfProfile output is 1 scan On every time one cam table operation is completed.



(11) Cam operation mode is set in StartMode. The setting range is either 0 or 1. If the input value exceeds the setting range, an error occurs. If it is set to 0, the cam table start position is set to the main-axis position of 0. If it is set to 1, the cam table start position is set to the current position of the main-axis.

(12) MasterValueSource selects the source of the main axis to be synchronized. If set to 0, the serve axis performs cam operations based on the command position of the main axis calculated in the motion controller, and if set to 1, the serve axis performs cam operations based on the current position received via communication from the servo drive of the main axis.

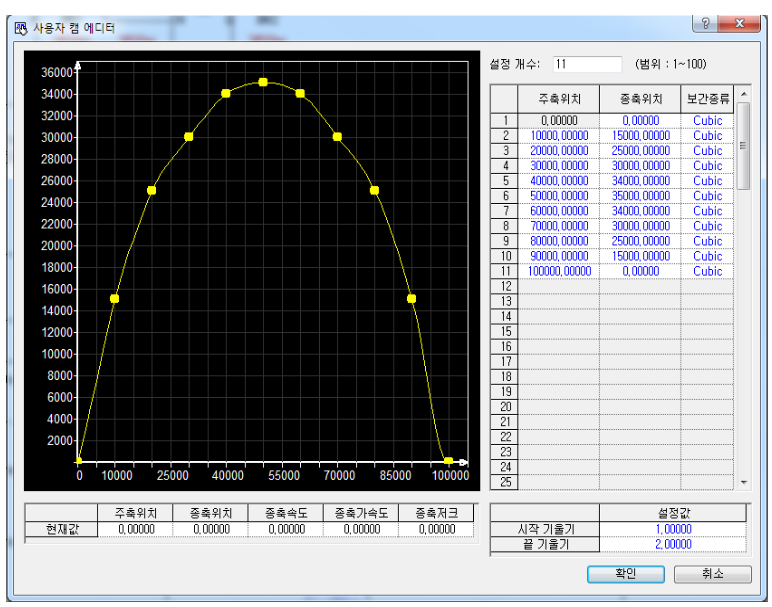
(13) CamTableID sets the number of cam table to be applied to cam operation. The setting range is from 1 to 32. If the input value exceeds the setting range, an error "0x1115" occurs at the motion function block.

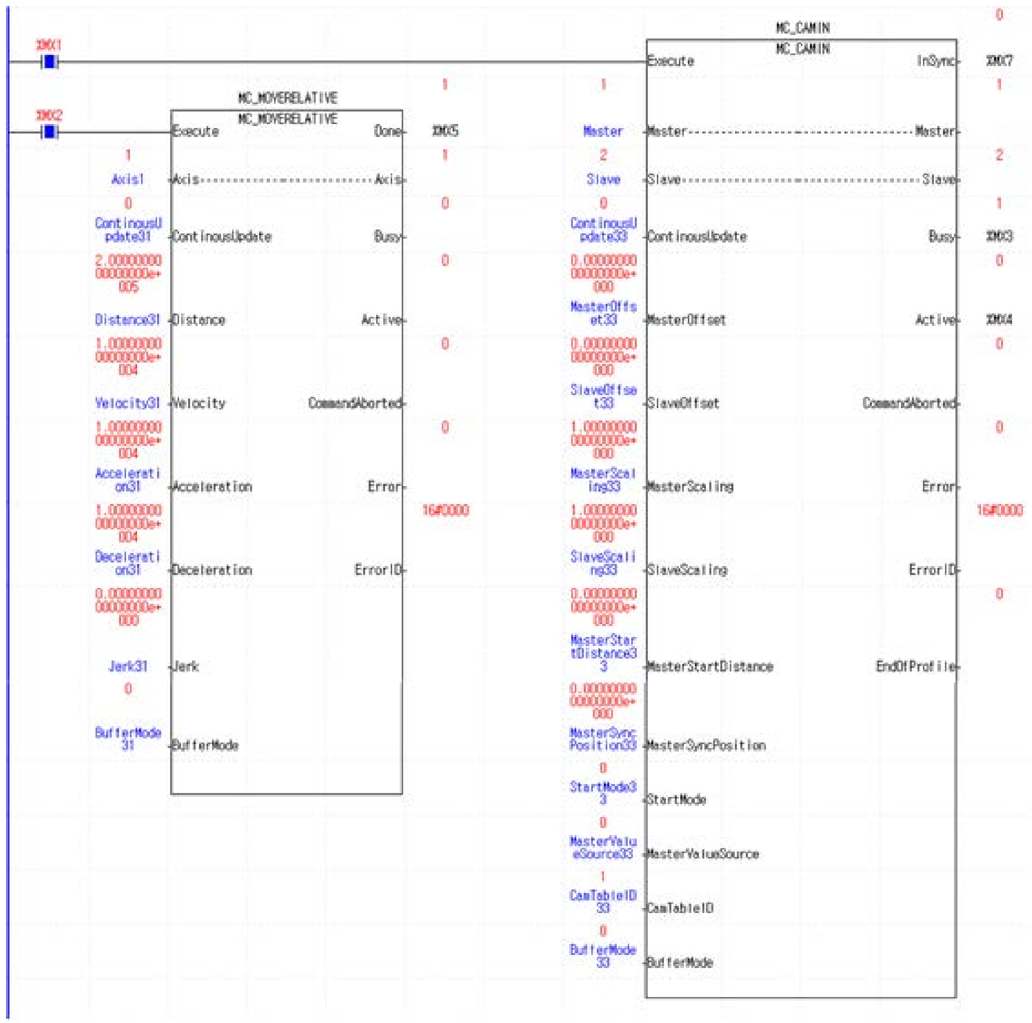
(14) The corresponding axis is in a "SynchronizedMotion" state when this motion function block is running.

(15) Example program

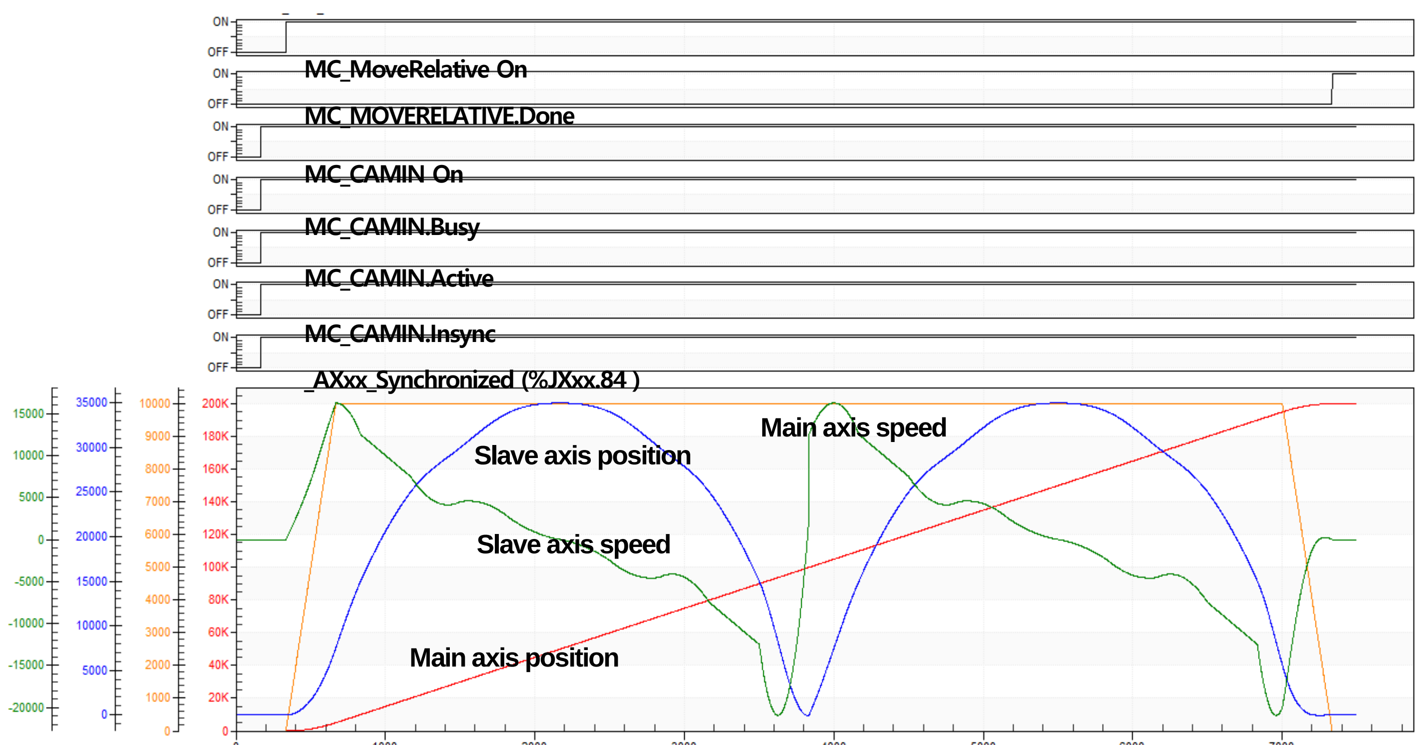
This example shows generating a cam profile, executing MC_CAMIN command on the sub-axis, moving the main-axis to the 200,000 position

(a) Function block setting





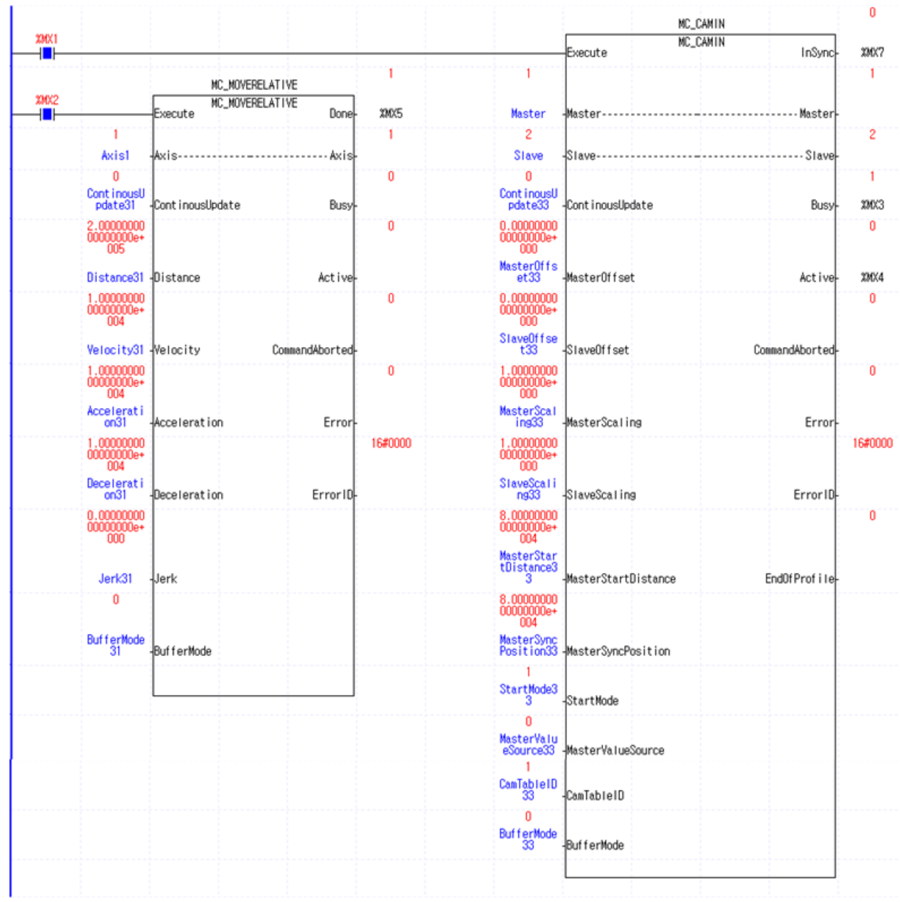
(b) Timing diagram



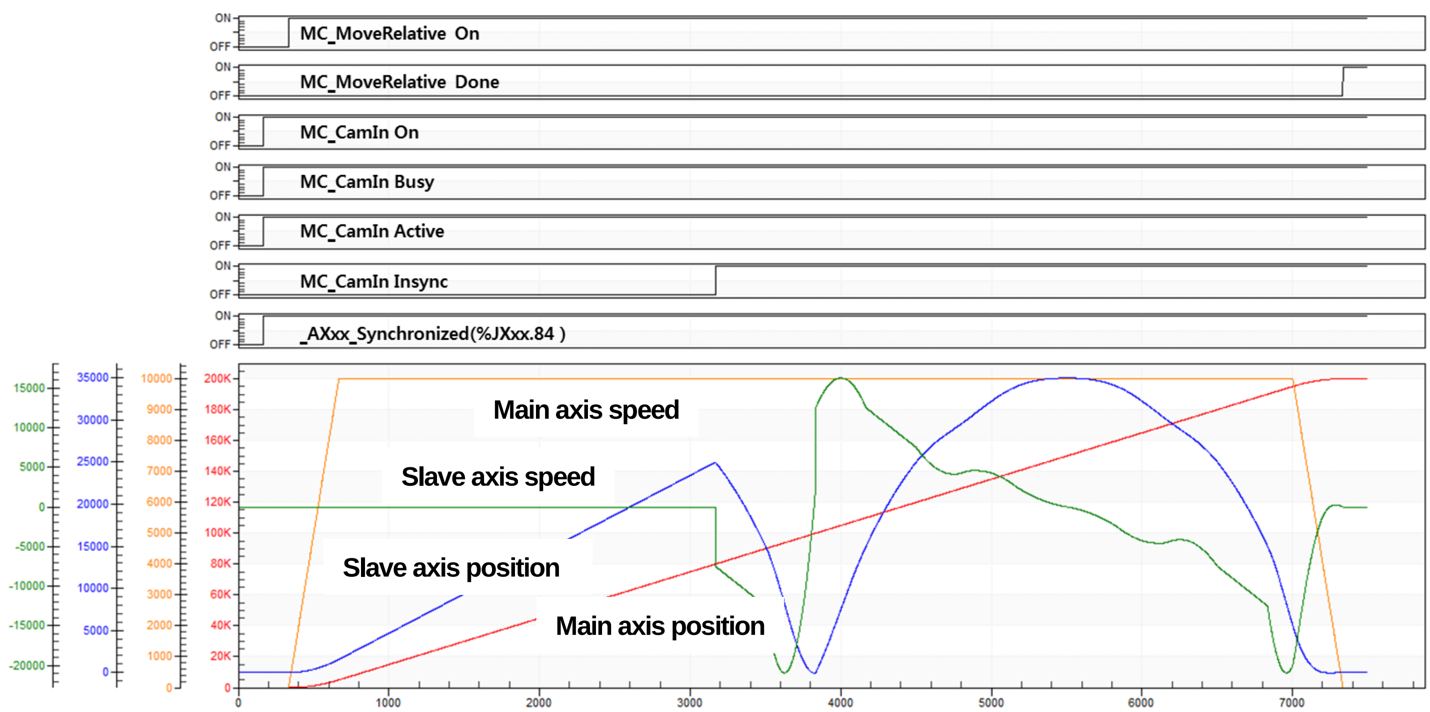
(16) Application example program

This example shows the movement of the main-axis from 0 to 200,000 positions after generating a cam profile and then executing MC_CAMIN command on the sub-axis.

(a) Function block setting



(b) Timing diagram



6.4.2 Cam operation Disable(MC_CamOut)

Motion Function Block

```
                    MC_CamOut
BOOL ─ Execute                        Done ─ BOOL
UINT ─ Slave ----------------------- Slave ─ UINT
                                      Busy ─ BOOL
                                     Error ─ BOOL
                                   ErrorID ─ WORD
```

Input-Output		
UINT	Slave	Set the serve axis. (refer toSee 6.2.1 Setting Range by Product)
Input		
BOOL	Execute	Give cam operation stop command to the relevant axis in the rising Edge.
Output		
BOOL	Done	Indicate the completion state of motion function block.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block immediately disengages cam operation running in the serve axis.

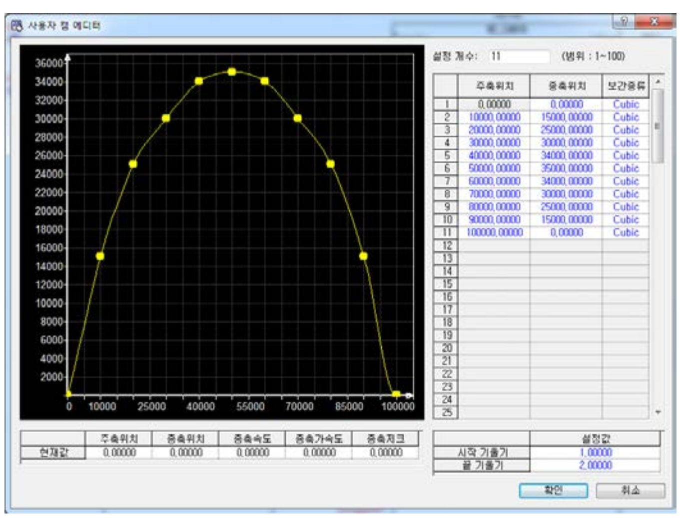
(2) If motion function block of which BufferMode is Aborting in the serve axis where cam operation is running, cam operation is automatically disengaged and the relevant motion function block is executed. To execute cam operation abort (MC_CamOut) motion function block, the relevant axis do operation which keeps the speed at the time when cam operation is disengaged. If you want to completely stop the serve axis, use stop (MC_Halt) or immediate stop (MC_Stop) motion function block.

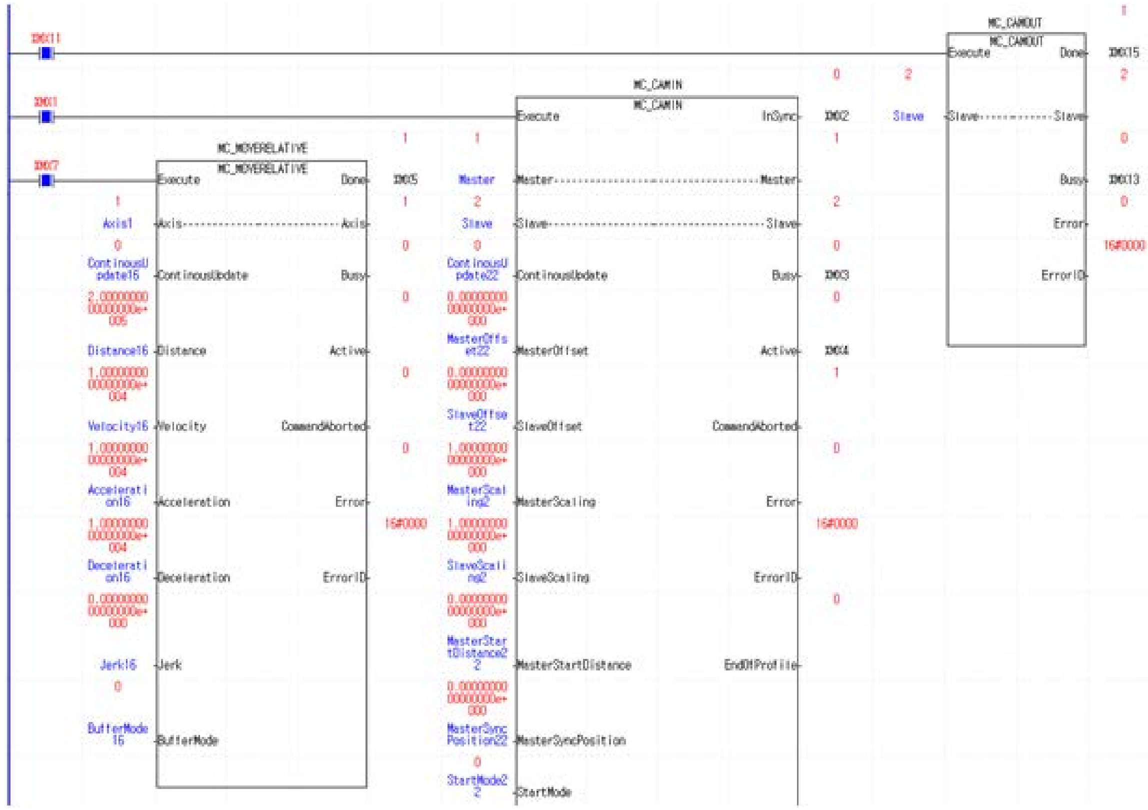
(3) When MC_CamOut motion function block is executed, the InSync output of MC_CamIn function block and the Synchronized status flag (_AXxx_Synchronized) is off.

(4) Example program

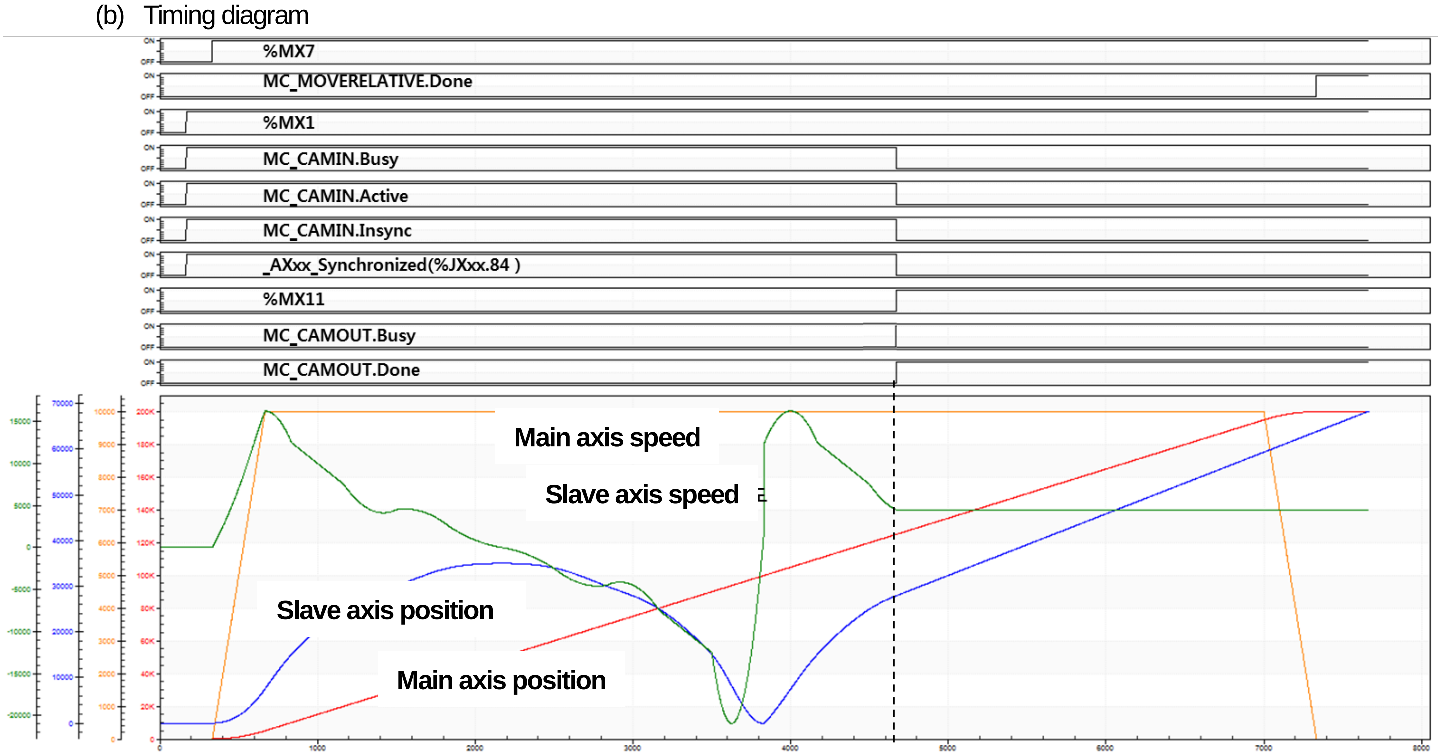
This example shows generating a cam profile, executing MC_CAMIN command on the sub-axis, moving main-axis to the 200,000 position and then executing MC_CAMOUT The sub-axis maintains the velocity at the time when the cam operation is terminated.

(a) Function block setting

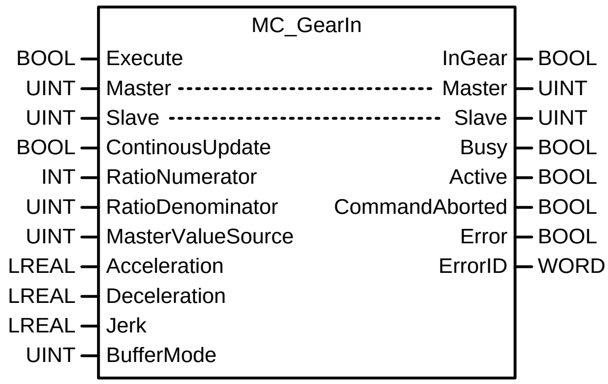




(b) Timing diagram



6.4.3 Gear operation (MC_GearIn)

Motion Function Block		
		
Input-Output		
UINT	Master	Set main axis (refer toSee 6.2.1 Setting Range by Product)
UINT	Slave	Set the serve axis. (refer toSee 6.2.1 Setting Range by Product)
Input		
BOOL	Execute	Give gear operation command to the relevant axis in the rising Edge.
BOOL	ContinuousUpdate	Specify the update setting of input value. (Refer to 6.1.5 Chnges in Parameters during Execution of motion Function Block)
INT	RatioNumerator	Specify the numerator of gear ratio. (-32768 ~ 32767)
UINT	RatioDenominator	Specify the denominator of gear ratio. (0 ~ 65535)
UINT	MasterValueSource	Select data of the main axis to be synchronized. 0: Synchronize in the command position of the main axis. 1: Synchronize in the current position of the main axis.
LREAL	Acceleration	Specify the acceleration at the beginning of gear operation synchronization. [u/s^2]
LREAL	Deceleration	Specify the deceleration at the beginning of gear operation synchronization. [u/s^2]
LREAL	Jerk	Specify the change rate of acceleration/deceleration. [u/s^3]
UINT	BufferMode	Specify the sequential operation setting of motion function block. (Refer to 6.1.4 BufferMode input)
Output		
BOOL	InGear	Indicate that gear operation is running by applying gear ration.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that the current motion function block is controlling the relevant axis.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted while it is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block is an operation to synchronize the speed of the main axis and the serve axis depending on gear ratio which is set.

(2) Giving gear operation abort (MC_GearOut) commands to the relevant axis or execution of other motion function block allow disengaging gear operation.

(3) RatioNumerator and RatioDenominator set the numerator and denominator to be applied to the serve axis respectively. If the numerator is set to negative number, the rotation direction of the serve axis is the opposite of the main axis.

(4) MasterValueSource select the data of the main axis which is a standard of synchronization. If it is set to 0, synchronization

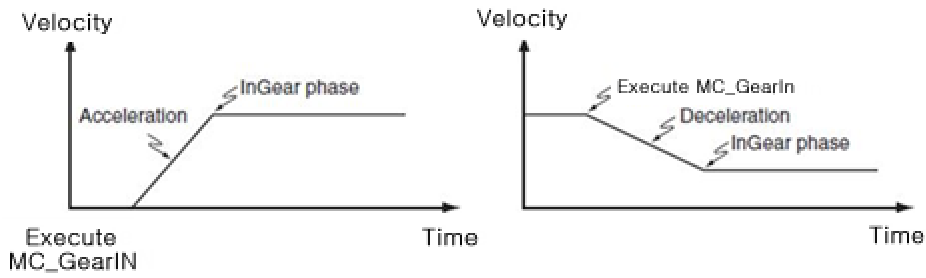
operation is based on the command position of the main axis of motion controller, and if it is set to 1, synchronization operation is based on the current position. Other values set besides these two make Error of motion function block On and cause “0x1114” in ErrorID.

(5) When this motion function block is executed, the sub axis accelerates/decelerates at a speed suitable for the gear ratio and synchronizes with the main axis.

(6) If this motion function block is aborted by another command (BufferMode=0 of newly executed command), the cam operation is stopped, and the CommandAborted output is on.

(7) If another command is executed by Buffered while this motion function block is being executed (BufferMode=1~5 of newly executed command), the status of gear operation (InGear phase) is terminated, and then the newly executed command is run subsequently. InGear / Busy / Active / CommandAborted / Error output of this function block are all Off.

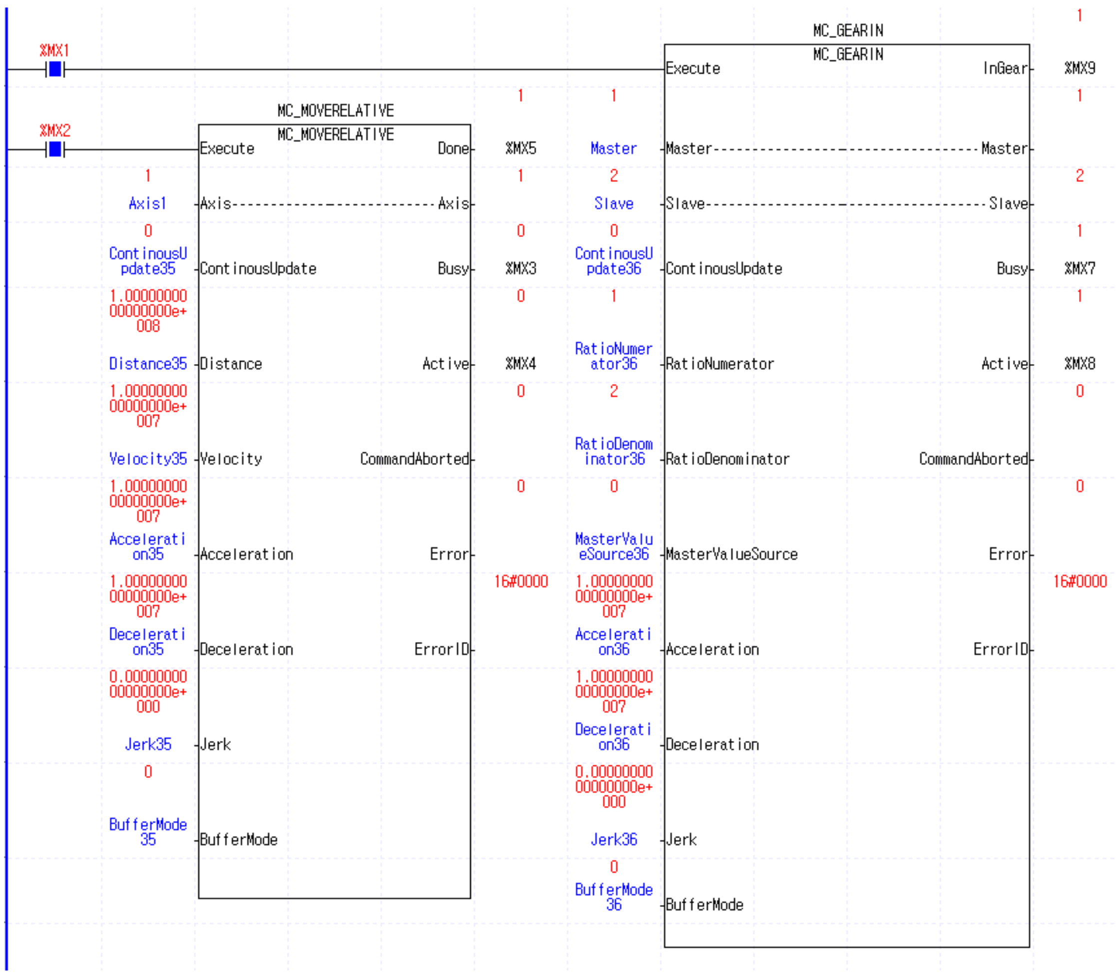
(8) The serve axis is in 'SynchronizedMotion' while this motion function block is running.



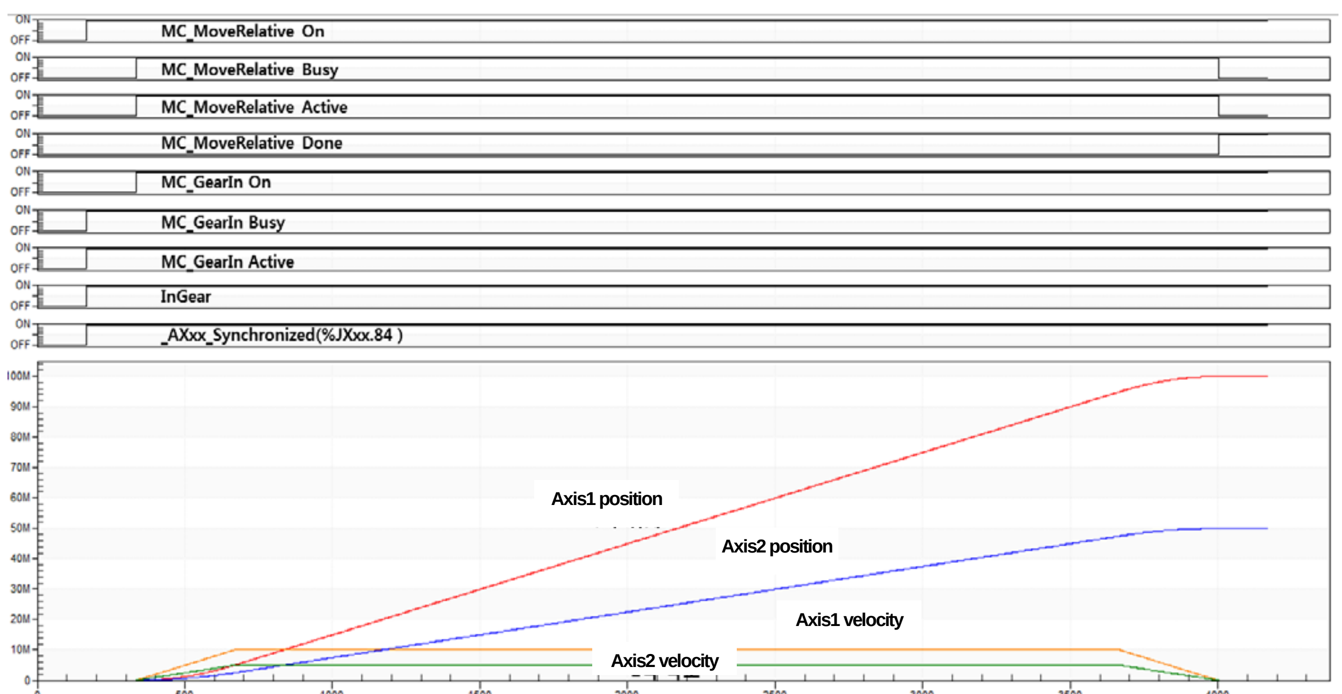
(9) The changed parameters can be applied when ContinuousUpdate input is On.
Only RatioNumerator, RatioDenominator, Acceleration, Deceleration input can be updated. (However, in case of InGear=On case, RatioNumerator, RatioDenominator input can be updated)

(8) Example program
This example shows the operation of 2–axis up to 50,000,000 when moving 1-axis (main-axis) to 100,000,000 after executing MC_GearIn command on axis 2(sub-axis) at the current position of 0.

(a) Function block setting

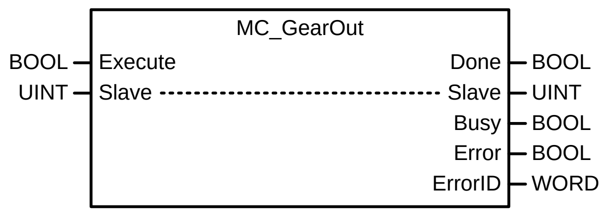


(b) Timing diagram



6.4.4 Gear operation Disable (MC_GearOut)

Motion Function Block



Input-Output		
UINT	Slave	Set the serve axis. (refer toSee 6.2.1 Setting Range by Product)
Input		
BOOL	Execute	Specify the sequential operation setting of motion function block.
Output		
BOOL	Done	Indicate the completion state of motion function block.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

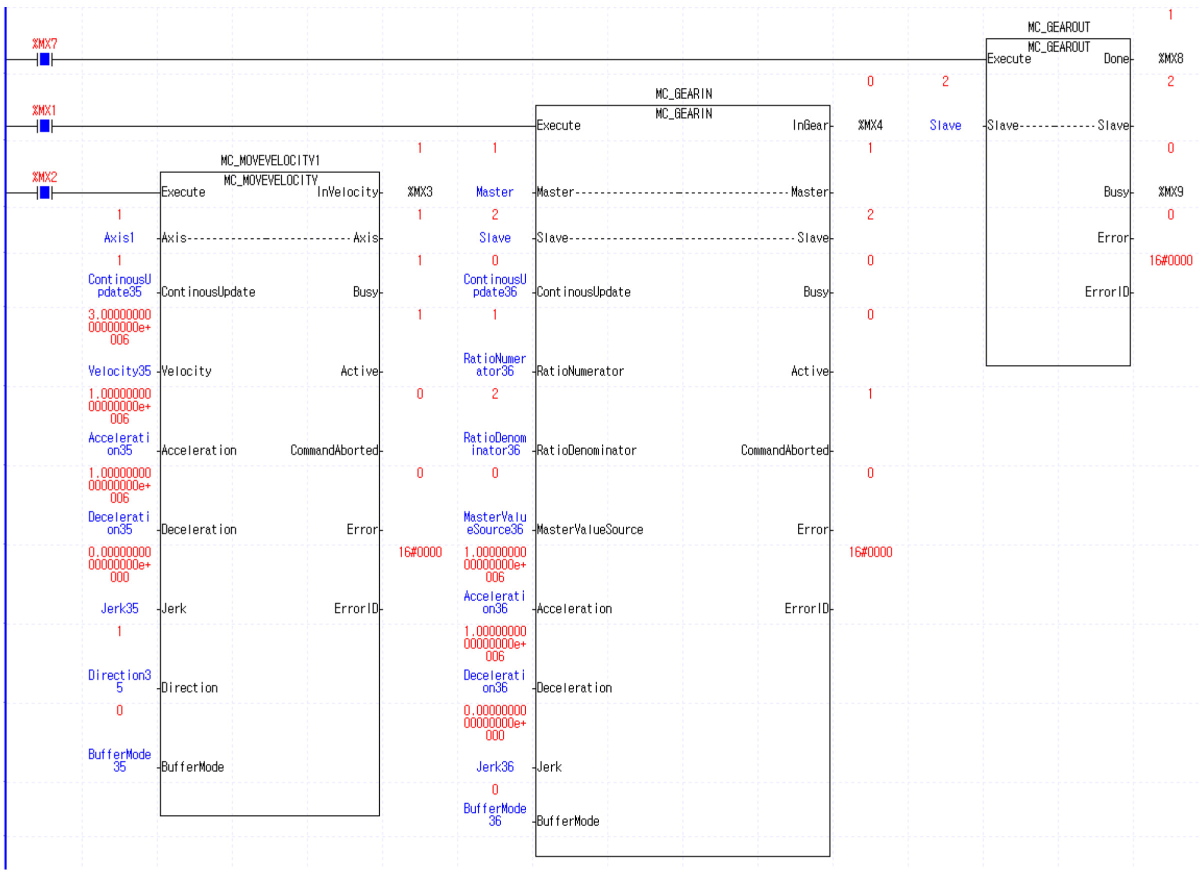
(1) This motion function block immediately disengages gear operation running in the spindle.

(2) If motion function block of which BufferMode is Aborting in the spindle where cam operation is running, gear operation is automatically disengaged and the relevant motion function block is executed. If gear operation abort (MC_GearOut) motion function block is only to be executed, the relevant axis performs operation to maintain the speed at the time when gear operation is disengaged. If you want to completely stop the serve axis, use stop (MC_Halt) or immediate stop (MC_Stop) motion function block.

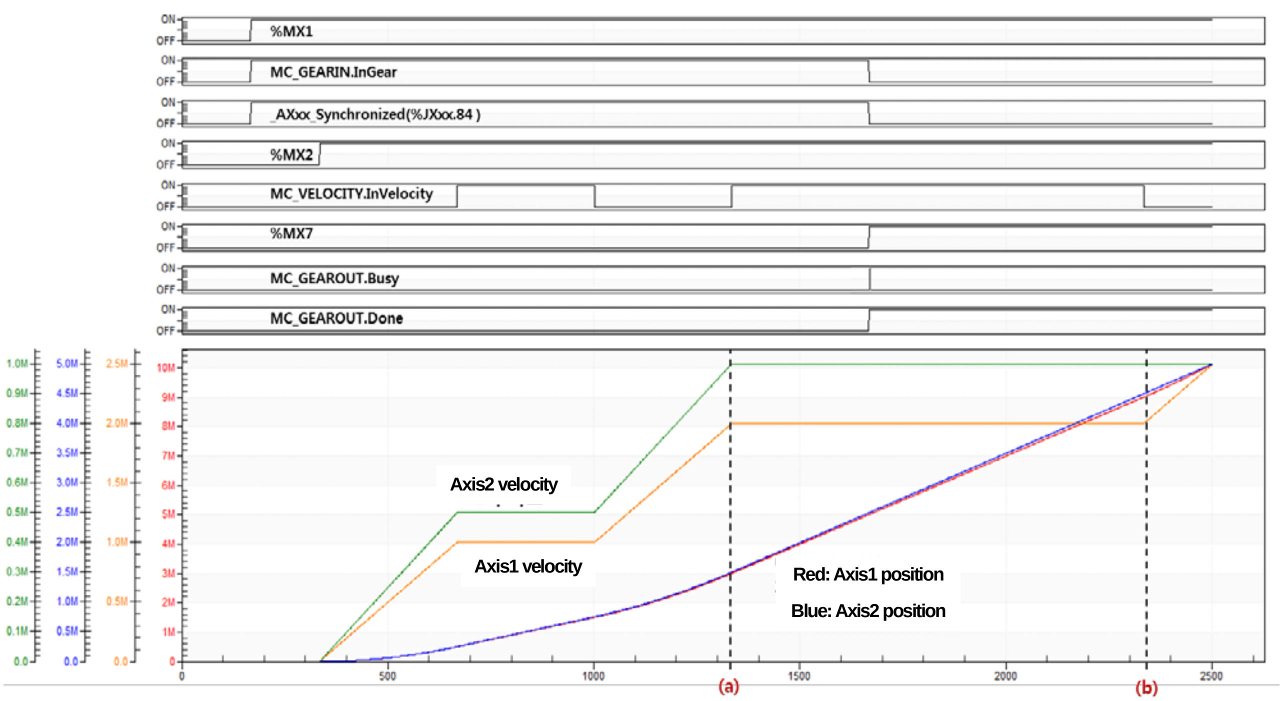
(3) Example program

After the execution of MC GearIn command in sub-axis at the current position of 0, ContinuousUpdate of main-axis is set to 1, and then the velocity is gradually changed (1,000,000 →2,000,000 →3,000,000). This examples show the operation to ensure that the velocity of sub-axis is no longer changed (b) by executing MC_GearOut command when the velocity of main-axis is 2,000,000.

(a) Function block setting

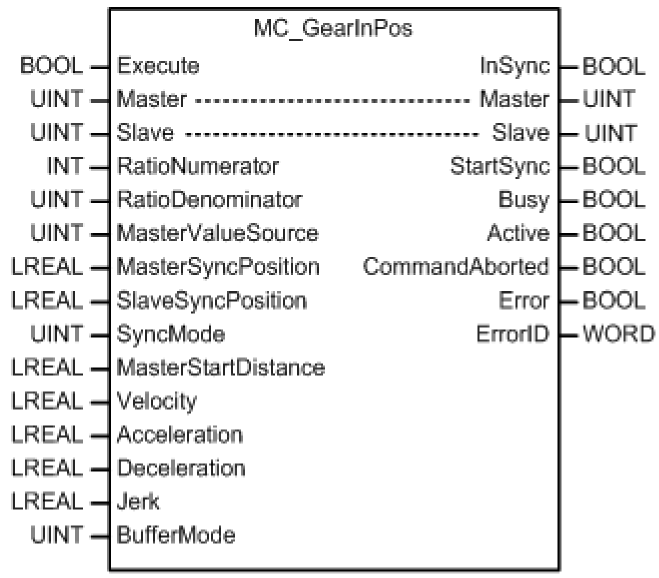


(b) Timing diagram



6.4.5 Gearing by specifying the position (MC_GearInPos)

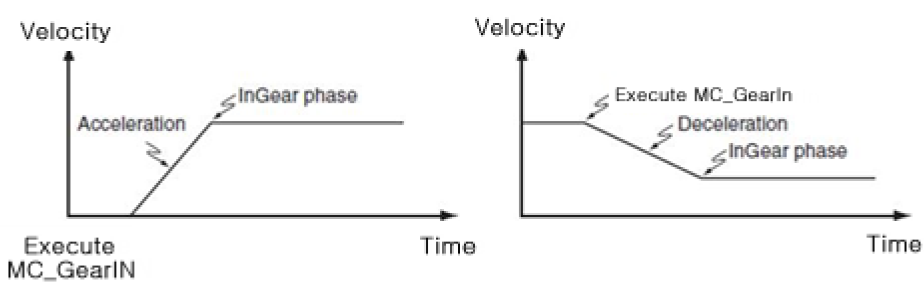
Motion Function Block

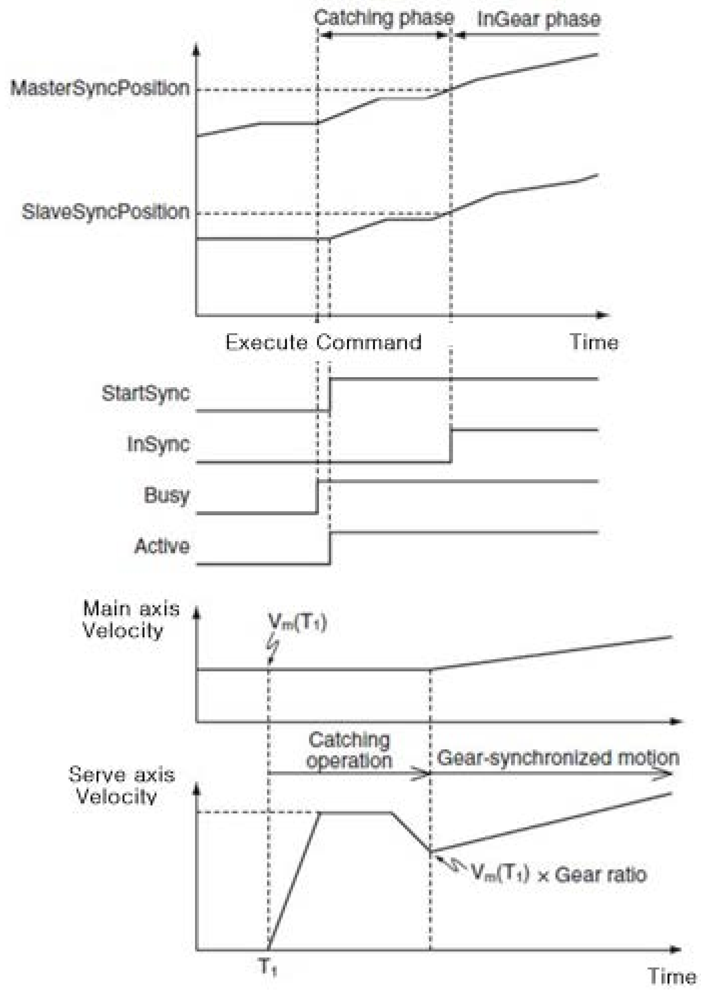


Input-Output		
UINT	Master	Set main axis (refer toSee 6.2.1 Setting Range by Product)
UINT	Slave	Set the serve axis. (refer toSee 6.2.1 Setting Range by Product)
Input		
BOOL	Execute	Give gear operation command to the relevant axis in the rising Edge.
INT	RatioNumerator	Specify the numerator of gear ratio. (-32768~32767)
UINT	RatioDenominator	Specify the denominator of gear ratio. (0~65535)
UINT	MasterValueSource	Select the standard of the main axis value to be synchronized. 0(mcSetValue): Synchronize in the target position of the main axis. 1(mcActualValue): Synchronize in the current position of the main axis.
LREAL	MasterSyncPosition	Specify the position of the main axis where gear operation starts.
LREAL	SlaveSyncPosition	Specify the position of the spindle where gear operation starts.
LREAL	SyncMode	Unused
LREAL	MasterStartDistance	Specify the distance of the main axis where synchronization starts.
LREAL	Velocity	Specify the maximum speed of the spindle at the beginning of synchronization. [u/s]
LREAL	Acceleration	Specify the maximum acceleration of the spindle at the beginning of synchronization. [u/s^2]
LREAL	Deceleration	Specify the maximum deceleration of the spindle at the beginning of synchronization. [u/s^2]
LREAL	Jerk	Unused
UINT	BufferMode	Specify the sequential operation setting of motion function block. (Refer to 6.1.4.BufferMode input)

Output		
BOOL	InSync	Indicate that gear operation is normally being fulfilled as the specified gear ratio is applied.
BOOL	StartSync	Indicate synchronization is starting.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that the current motion function block is controlling the relevant axis.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted while it is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

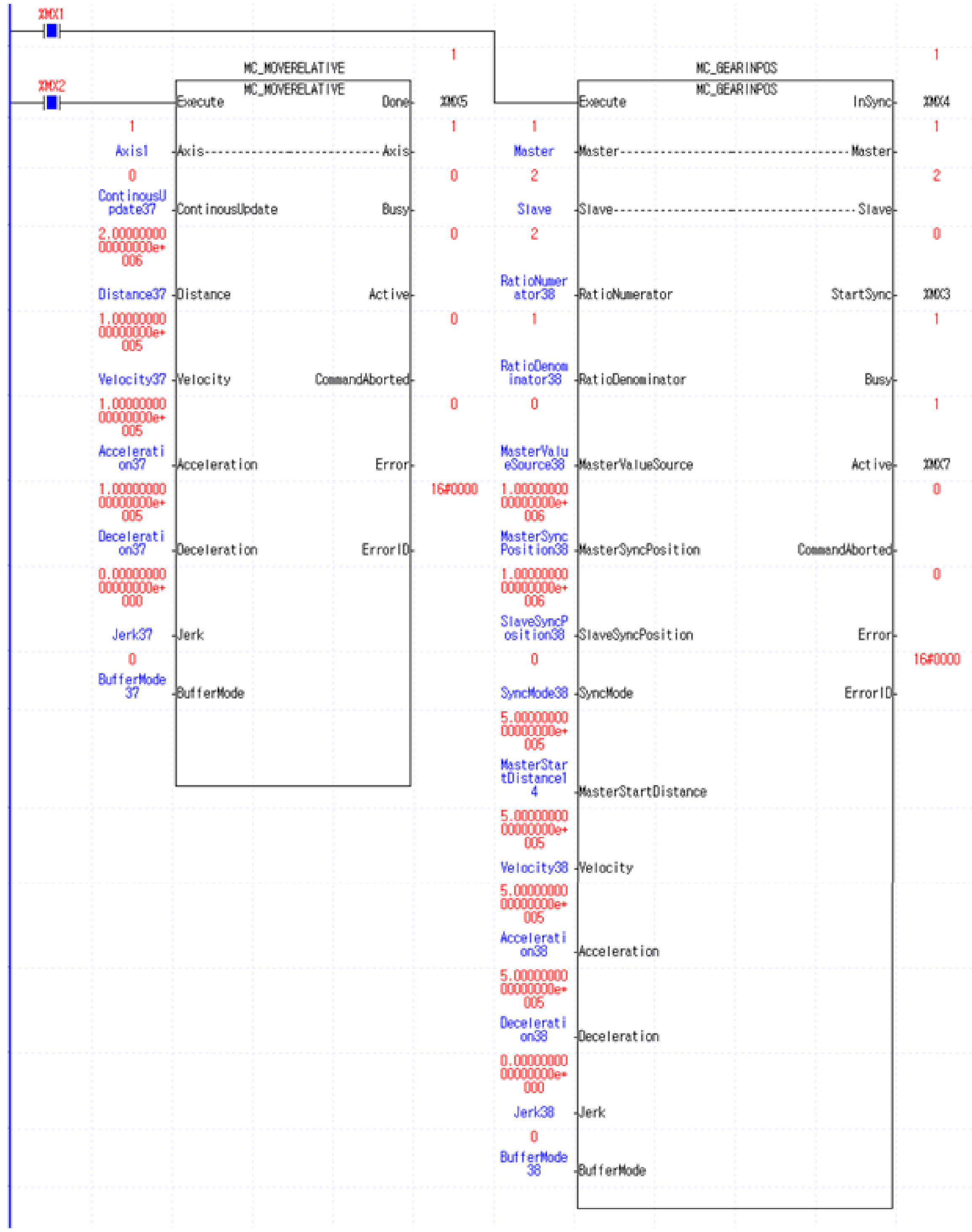
(1) This motion function block is an operation to synchronize the speed of the main axis and the spindle in the set position depending on gear ratio which is set in the specific position.
(2) Giving gear operation abort (MC_GearOut) commands to the spindle or operation of other motion function block allow stopping gear operation.
(3) RatioNumerator and RatioDenominator set the numerator and denominator of gear ratio to be applied to the spindle respectively. If the numerator is set to negative number, the rotation direction of the spindle goes into reverse of the main axis.
(4) MasterValueSource selects the source of the main axis to be synchronized. If it is set to 0 (mcSetValue), synchronization is performed by putting the target position of the main axis in the current motion control period as a source, and if it is set to 1(mcActualValue), synchronization is performed by putting the current position of the main axis got feedback from the current motion control period as a source. Other values set besides these two cause "error 0x10D1".
(5) Input the positions of the main axis and the spindle where gear operation is completed synchronization in MasterSyncPosition input and SlaveSyncPosition input respectively. Input the distance where the spindle starts synchronization in MasterStartDistance input, and the spindle starts synchronization at the position away the distance set in MasterStartDistance input from the position set in MasterSyncPosition input.
(6) Once synchronization starts, StartSync output is On. When synchronization is completed and gear operation starts, StartSync output is Off and InSync output is On.
(7) If this motion function block is aborted by another command (BufferMode=0 of newly executed command), the cam operation is stopped, and the CommandAborted output is on.
(8) If another command is executed by Buffered while this motion function block is being executed (BufferMode=1~5 of newly executed command), the status of gear operation (InGear phase) is terminated, and then the newly executed command is run subsequently. InSync / Busy / Active / CommandAborted / Error output of this function block are all Off.
(9) The serve axis is in 'SynchronizedMotion' while this motion function block is running.



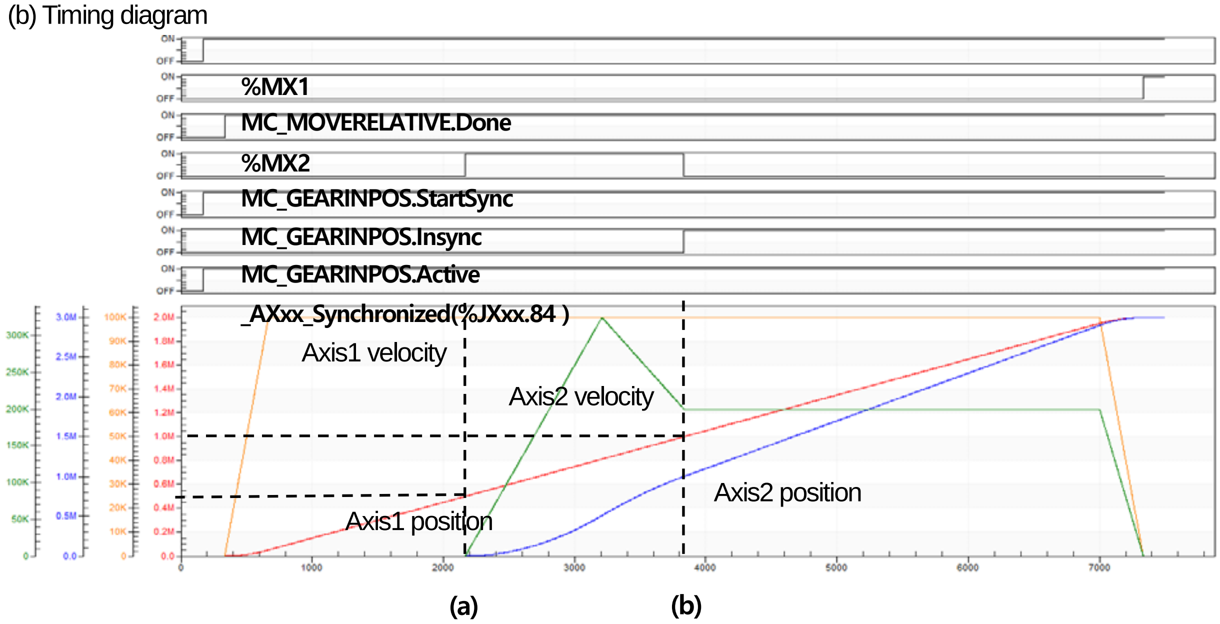


(10) If the function block is re-executed (Execute input is On) before the instruction is terminated, the changed parameters are applied.
Only RatioNumerator, RatioDenominator, MasterSyncPosition, SlaveSyncPosition, MasterStartDistance, Velocity, Acceleration, Deceleration input can be updated. (However, in case of InGear=On, RatioNumerator, RatioDenominator input can be updated.

(9) Example program
This example program executes MC_GearInPos function block in which sub-axis starts synchronization from a position away as long as the distance of MasterStartDistance (500,000) from MasterSyncPosition(1,000,000), and executes MC_MoveRelative for relative movement to the 2,000,000 position. Once synchronization starts, StartSyncoutput is on (a) and when the synchronization is completed and gear operation starts, StartSync output is off, and InSyncoutput is on.
(a) Function block setting


%MX1
%MX2
MC_MOVERELATIVE
Execute
Done
%MX5
Axis1
Axis
ContinousUpdate37
ContinousUpdate
Busy
Distance37
Distance
Active
Velocity37
Velocity
CommandAborted
Acceleration37
Acceleration
Error
Deceleration37
Deceleration
ErrorID
16#0000
Jerk37
Jerk
BufferMode37
BufferMode
MC_GEARINPOS
InSync
%MX4
Master
Slave
RatioNumerator38
RatioNumerator
StartSync
%MX3
RatioDenominator38
RatioDenominator
MasterValueSource38
MasterValueSource
%MX7
MasterSyncPosition38
MasterSyncPosition
SlaveSyncPosition38
SlaveSyncPosition
SyncMode38
SyncMode
MasterStartDistance14
MasterStartDistance
Velocity38
Acceleration38
Deceleration38
Jerk38
BufferMode38

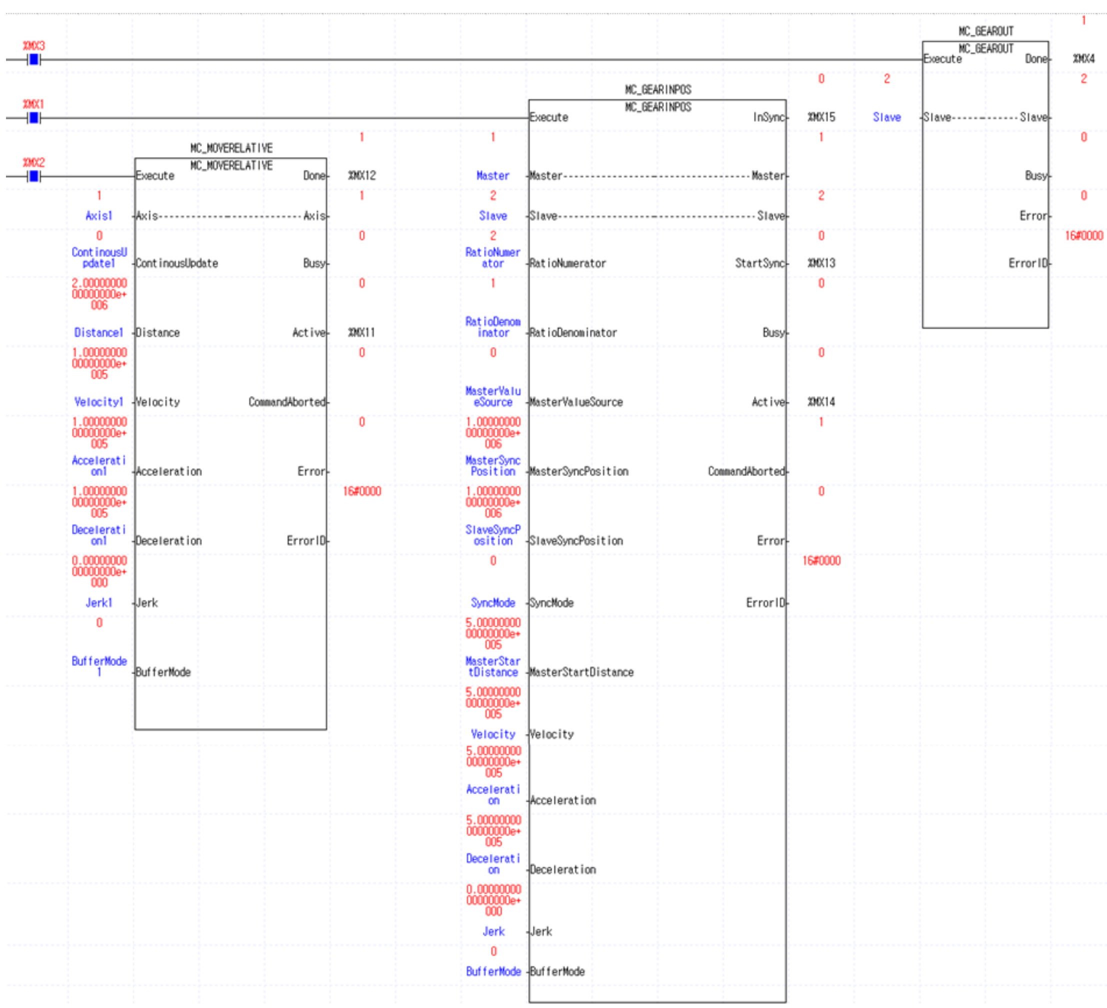
(b) Timing diagram



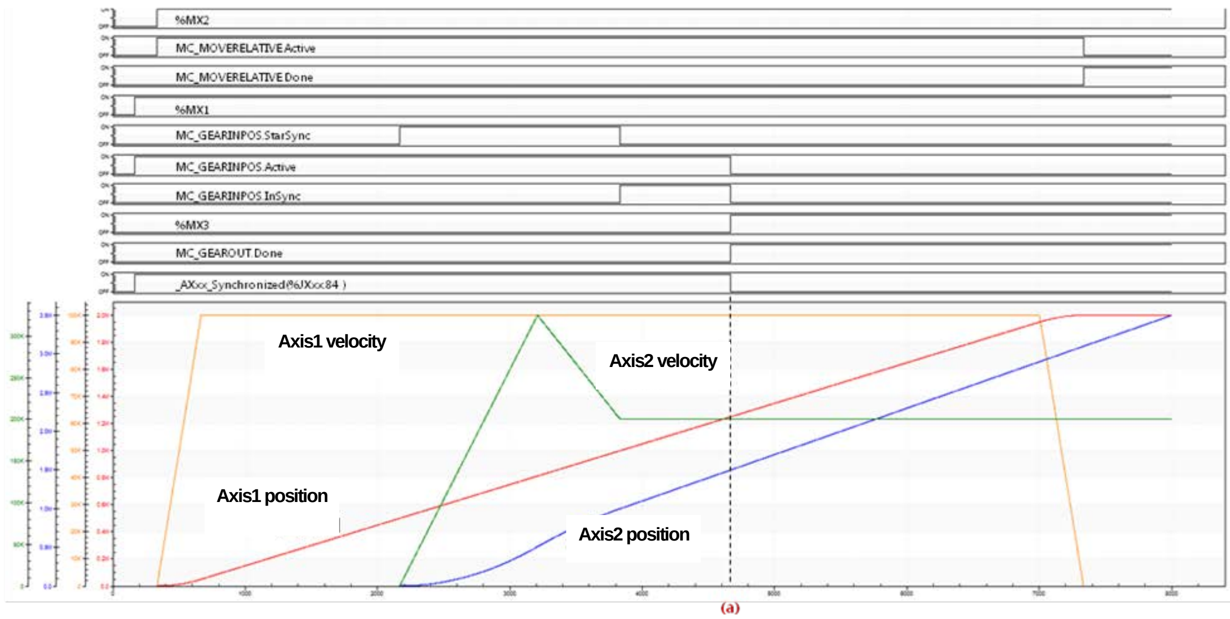
(10) Application example program

This example program shows MC_GearInPos Active and InSync being off and gear operation being terminated when MC_GearOut command is issued on 2-axis at (a) position during the motion shown in the basic example program. (Gear operation termination can be verified by 1-axis that stops and 2-axis that continues to operate)

(a) Function block setting

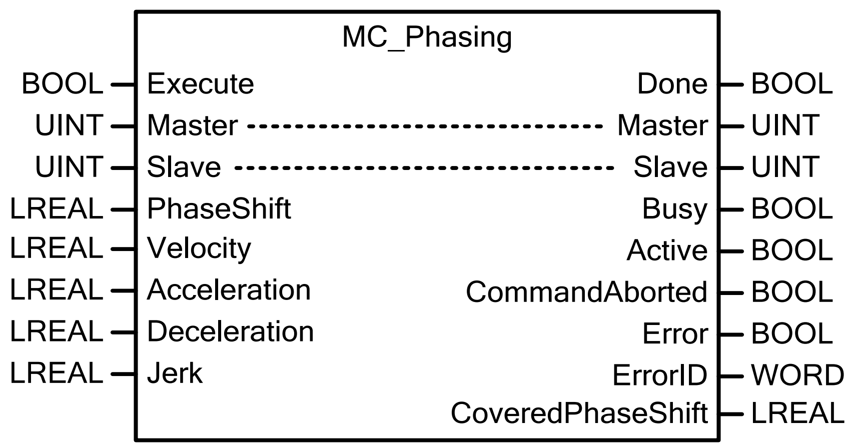


(b) Timing diagram



6.4.6 Phase compensation (MC_Phasing)

Motion Function Block



Input-Output		
UINT	Master	Set main axis (refer toSee 6.2.1 Setting Range by Product)
UINT	Slave	Set the serve axis. (refer toSee 6.2.1 Setting Range by Product)
Input		
BOOL	Execute	Give a phase compensation command to the relevant axis in the rising Edge
LREAL	PhaseShift	Specify the main axis compensation amount.
LREAL	Velocity	Specify the operation speed. [u/s]
LREAL	Acceleration	Specify the acceleration. [u/s^2]
LREAL	Deceleration	Specify the deceleration. [u/s^2]
LREAL	Jerk	Specify the change rate of acceleration/deceleration. [u/s^3]
Output		
BOOL	Done	Indicate whether to reach the specified phase compensation distance.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that the current motion function block is controlling the relevant axis.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted while it is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.
LREAL	CoveredPhaseShift	Continuously output the compensation amount reflected while the phase compensation is running

(1) This motion function block performs phase correction of axis during synchronous control operation. Phase correction is performed on the main-axis position referred to by sub-axis in synchronous control operation, to perform synchronous control operation of the sub-axis to the corrected main-axis position.

(2) Once phase correction command is executed, the current position of the main-axis is phase-corrected using the phase shift setting at PhaseShift- Velocity / Acceleration /Deceleration / Jerk.

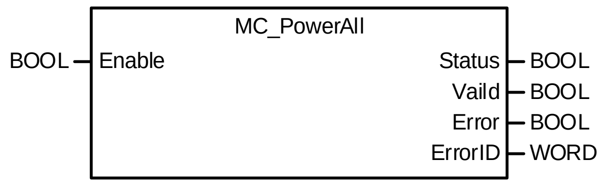
(3) Phase correction does not change the actual command position or current position of the main-axis. Phase correction is performed on the main-axis position referred to by sub-axis in synchronous control operation. In other words, the main-axis does not know that phase correction is executed by the sub-axis.

(4) Phase correction of the same amount can be performed again from the current position by re-executing the function block (Execute input is on) before the command is completed. In other words, phase shift is a relative value from the execution point.

(5) After executing phase correction command, when the phase shift is reached, Done output is on.

6.4.7 All axis Servo On/Off (MC_PowerAll)

Motion Function Block

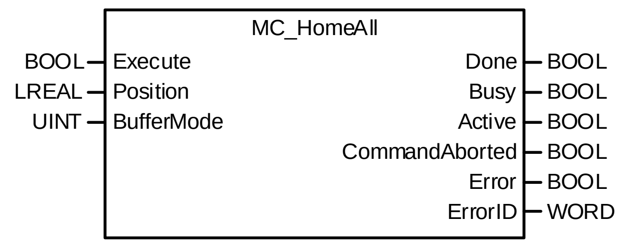


Input		
BOOL	Enable	Servo motor of the all axis is servo On while input is activated.
Output		
BOOL	Status	Indicate the power permission status of the all axis.
BOOL	Valid	Indicate the validity of motion function block output. (same with Status output here)
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block is to give servo On/Off command to the all axis.
(2) When the enable input is changed from Off to On, the servo on command is issued to the servo ready axis among all axes, and when the enable input is changed from On to Off, the servo off command is issued.
(3) “Error 0x0E90” occurs when there is an axis that has failed to turn servo on/off among configuration axes.
(4) If servo On command is executed when the axis is in 'Disable' state, the axis state is 'StandStill', and failure in servo On brings 'ErrorStop' state.

6.4.8 All axis Homing (MC_HomeAll)

Motion Function Block



Input		
BOOL	Execute	Start the homing operation in rising Edge.
LREAL	Position	Specify the absolute position of axis when reference signal is detected. (When the command position value is converted to a pulse, it is set within the range of -2147483648 to 2147483647.)
UINT	BufferMode	Specify the sequential operation setting of motion function block. (Refer to 6.1.4 BufferMode input)
Output		
BOOL	Done	Indicate the completion state of motion function block.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that the current motion function block is controlling the relevant axis.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted by other command.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block is to give a homing command to the all axis.
(2) Homing method is operated as specified in the operation parameter of the each axis in advance.
(3) As for Position input, absolute position of axis is specified when Reference Signal is detected or homing is completed.
(4) While this motion function block is running, the axis is 'Homing' state, and when the command is completed, it is switched to 'Standstill'.
(5) “Error 0x0E91” occurs when there is an axis that has failed to turn homing among configuration axes.
(5) Unlike the MC_Home command, even if the input parameter value is changed before the command is finished and the function block is re-executed (Execute input is On), it is not updated unlike the MC_Home command.

6.4.9 All axis Expansion Homing (LS_HomeAll)

Motion Function Block

LS_HomeAll

BOOL	Execute	Done	BOOL
LREAL	Position	Busy	BOOL
SINT	HomingMethod	Active	BOOL
LREAL	SwitchSearchSpeed	CommandAborted	BOOL
LREAL	ZeroSearchSpeed	Error	BOOL
LREAL	HomingAcc	ErrorID	WORD
LREAL	HomeOffset		
UINT	DoneBehavior		
UINT	BufferMode		

Input

BOOL	Execute	Start the homing operation in rising Edge.
LREAL	Position	Specify the absolute position of axis when reference signal is detected. (When the command position value is converted to a pulse, it is set within the range of -2147483648 to 2147483647.)
SINT	HomingMethod	Set the homing method (0x6098)
LREAL	SwitchSearchSpeed	Set the switch search speed (0x6099:1) when returning to the homing.
LREAL	ZeroSearchSpeed	Set the zero search speed (0x6099:2) when returning to the homing.
LREAL	HomingAcc	Set acceleration/deceleration (0x609A) when returning to homing.
LREAL	HomeOffset	Set Offest (0x607C) when returning to homing.
UINT	DoneBehavior	Set the movement (0x201E) after homing is completed. (Used by drives other than our own L7N, the value has no effect on third-party drives)
UINT	BufferMode	Specify the sequential operation setting of motion function block. (Refer to 6.1.4 BufferMode input)

Output

BOOL	Done	Indicate the completion state of motion function block.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that the current motion function block is controlling the relevant axis.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted by other command.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block is to give a expansion homing command to the all axis.
(2) The homing method operates as set in the function block. (The homing method may be different for each drive, so please refer to the user manual of the drive you are using. When using our XDL-N series servo drive, please refer to 8.1 Origin Determination)
(3) As for Position input, absolute position of axis is specified when Reference Signal is detected or homing is completed.
(4) While this motion function block is running, the axis is 'Homing' state, and when the command is completed, it is switched to 'Standstill'.
(5) “Error 0x0E92” occurs when there is an axis that has failed to turn homing among configuration axes.
(6) Unlike the LS_Home command, even if the input parameter value is changed before the command is finished and the function block is re-executed (Execute input is On), it is not updated unlike the MC_Home command.

(7) The input range for SwitchSearchSpeed, ZeroSearchSpeed, and HomingAcc is 0 ≤ Parameter ≤ 1073741824 in pulse units, and the input range for HomeOffset is -2147483648≤ Parameter ≤2147483647 in pulse units. If a value is input in a unit other than pulse unit, it is converted to pulse unit and checked if the value is out of range, and an error occurs if it is out of range.

(8) In the case of DoneBehavior, this is a function to set the movement method after homing is completed. It is used for drives other than our own L7N, and the value set in the function block does not affect when a third-party drive is used.
0: After homing is completed, the motor does not rotate and the Home Offset value becomes Zero Position.
1: After homing is completed, the motor moves as much as the Home Offset value, and it becomes the Zero Position at the position where the movement is completed.

6.4.10 All axis Stop immediately (MC_StopAll)

Motion Function Block

```
                 MC_StopAll
BOOL  - Execute                        Done - BOOL
LREAL - Deceleration                   Busy - BOOL
LREAL - Jerk                 CommandAborted - BOOL
                                      Error - BOOL
                                    ErrorID - WORD
```

Input		
BOOL	Execute	Give immediate stop command to the relevant axis in the rising Edge.
LREAL	Deceleration	Specify the deceleration in time of stop. [u/s^2]
LREAL	Jerk	Specify the change rate of acceleration/deceleration. [u/s^3]
Output		
BOOL	Done	Indicate that the speed of the relevant axis reaches 0.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted while it is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block is to give a Emergency Stop command to the all axis.

(2) When all axis execute immediate stop (MC_StopAll) motion function block, the existing motion function block being executed in the all axis is stopped, All the axis state changed to 'Stopping'. When the relevant axis is in 'Stopping' state, other motion function block cannot be executed in the relevant axis until the stopping is completed (until the Done output is activated).

(3) CommandAborted output indicates that the current motion function block is interrupted while it was executed. Other motion function block cannot interrupt immediate stop (MC_StopAll) motion function block while all axis execute immediate stop (MC_StopAll) motion function block is running, therefore, CommandAborted output is On in general when the power of servo is blocked or servo Off command is executed.

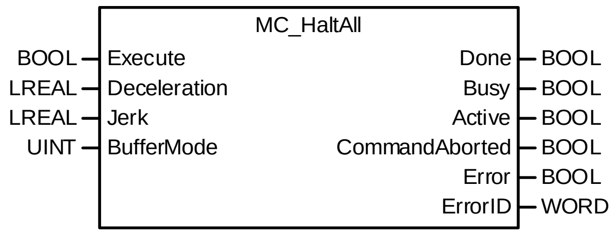
(4) If Execute input is On or the speed of axis is not 0, the axis is in 'Stopping' state, and when Done output is On and Execute input is Off, it is switched to 'Standstill' state.

(5) “Error 0x0E93” occurs when there is an axis that has failed to immediate stop among configuration axes.

(5) Unlike the MC_Home command, even if the input parameter value is changed before the command is finished and the function block is re-executed (Execute input is On), it is not updated unlike the MC_Stop command.

6.4.11 All axis Stop (MC_HaltAll)

Motion Function Block



Input		
BOOL	Execute	Give stop command to the relevant axis in the rising Edge.
LREAL	Deceleration	Specify deceleration in time of stop. [u/s^2]
LREAL	Jerk	Specify the change rate of acceleration/deceleration. [u/s^3]
UINT	BufferMode	Specify the sequential operation setting of motion function block. (Refer to 6.1.4 BufferMode input)
Output		
BOOL	Done	Indicate that the speed of the relevant axis reaches 0.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that the current motion function block is controlling the relevant axis.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted while it is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block is to give a stop command to the all axis.

(2) The axis is 'DiscreteMotion' state while this motion function block is running, and when the speed of the relevant axis is 0, ‘Done’ output is On and changed to 'Standstill' state.

(3) BufferMode can be set like the MC_Halt command, and if there is at least one axis whose stop command (MC_Halt) is interrupted by another motion function block, it is displayed as Error.

(4) “Error 0x0E94” occurs when there is an axis that has failed to stop among configuration axes.

(3) Unlike the MC_Halt command, even if the input parameter value is changed before the command is finished and the function block is re-executed (Execute input is On), it is not updated unlike the MC_Stop command.

6.4.12 Reset All axis error 2(MC_Reset2All)

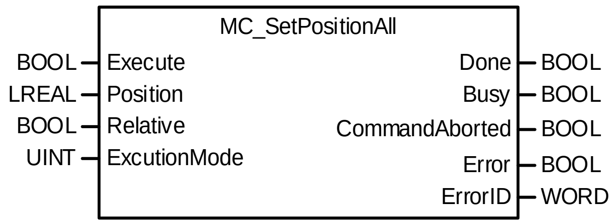
Motion Function Block

Input		
BOOL	Execute	Reset the axis error in the rising Edge of input.
UINT	ErrorType	The types of error to be reset (0: Axis error, 1: Common error, 2: Axis error and Common error)
Output		
BOOL	Done	Indicates whether the axis error was successfully reset
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block is to reset the error of the all axis. When setting ErrorType to '0' and executing motion function block in case the axis is in ' ErrorStop' state, every axis error is reset and the axis state is switched to 'StandStill' or 'Disabled' state.
(2) If ErrorType is set to ‘1’ and motion function block is executed, Common error occurred in the relevant module is reset. If ErrorType is set to ‘2’ and motion function block is executed, errors and common errors occurring in the set axis are reset.
(3) Motion function block is executed in the rising Edge of Execute input.
(4) “Error 0x0E95” occurs when there is an axis that has failed to error reset among configuration axes.

6.4.13 All axis Setting the current position (MC_SetPositionAll)

Motion Function Block



Input		
BOOL	Execute	Specify the current position of the relevant axis in the rising Edge.
LREAL	Position	Specify the position.
BOOL	Relative	0: Position value=Absolute position, 1: Position value=Relative position
UINT	ExecutionMode	0: Immediately applied the position value, 1: Applied at the same point with 'Buffered' of Buffermode
Output		
BOOL	Done	Indicate the completion state of motion function block.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted while it is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block is to set the current position of the all axis.

(2) Specify the position in Position input. When executing motion function block, if Relative input is Off, the position of the relevant axis is replaced by the value of Position input, and if Relative input is On, the value of Position input is added to the current position of the relevant axis.

(3) "Error 0x0E96" occurs when there is an axis that has failed to turn current position setting among configuration axes.

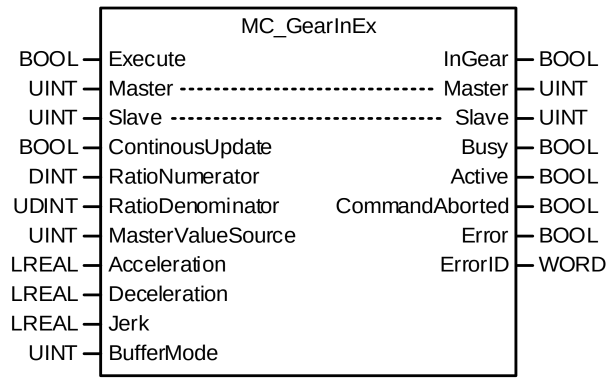
(4) ExecutionMode input specifies the setting point. 0 means to be set immediately after motion function block, and 1 means to be set at the same point with 'Buffered' in sequential operation setting. If a value that cannot be set is set, "Error 0x101B" occurs on the relevant axis.

0 (mcImmediately): Change the parameter value immediately after executing function block (rising Edge in Execute input). If the relevant axis is in running, operation can be affected.

1 (mcQueued): Changed at the same point with 'Buffered' in Buffermode. (Refer to 6.1.4 BufferMode input)

6.4.14 Expansion Gear operation (MC_GearInEx)

Motion Function Block



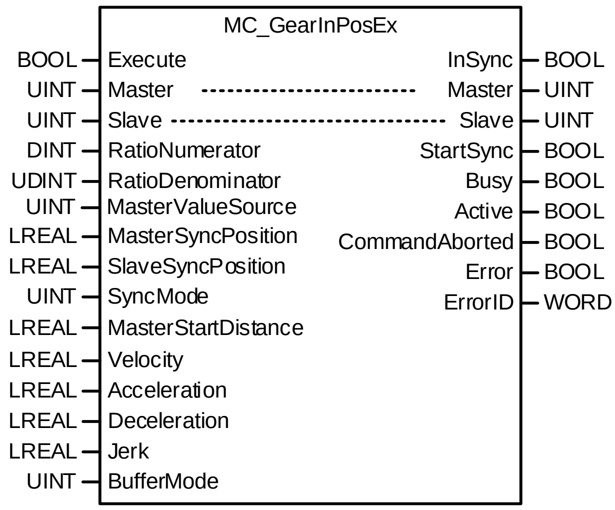
Input-Output		
UINT	Master	Set main axis (refer toSee 6.2.1 Setting Range by Product)
UINT	Slave	Set the serve axis. (refer toSee 6.2.1 Setting Range by Product)
Input		
BOOL	Execute	Give gear operation command to the relevant axis in the rising Edge.
BOOL	ContinuousUpdate	Specify the update setting of input value. (Refer to 6.1.5 Changes in Parameters during Execution of Motion Function Block)
DINT	RatioNumerator	Specify the numerator of gear ratio. (-2147483648 ~ 2147483647)
UDINT	RatioDenominator	Specify the denominator of gear ratio. (0 ~ 4294967295)
UINT	MasterValueSource	Select data of the main axis to be synchronized. 0: Synchronize in the command position of the main axis. 1: Synchronize in the current position of the main axis.
LREAL	Acceleration	Specify the acceleration at the beginning of gear operation synchronization. [u/s^2]
LREAL	Deceleration	Specify the deceleration at the beginning of gear operation synchronization. [u/s^2]
LREAL	Jerk	Specify the change rate of acceleration/deceleration. [u/s^3]
UINT	BufferMode	Specify the sequential operation setting of motion function block. (Refer to 6.1.4 BufferMode input)
Output		
BOOL	InGear	Indicate that gear operation is running by applying gear ration.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that the current motion function block is controlling the relevant axis.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted while it is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) In this motion function block, RatioNumerator input and RatioDenominator input are expanded to DINT and UDINT respectively.

(2) The rest of the settings and functions are the same as those of the MC_GearIn function block. Please refer to the description of the MC_GearIn function block.

6.4.15 Extended electrical gearing by specifying the position (MC_GearInPosEx)

Motion Function Block

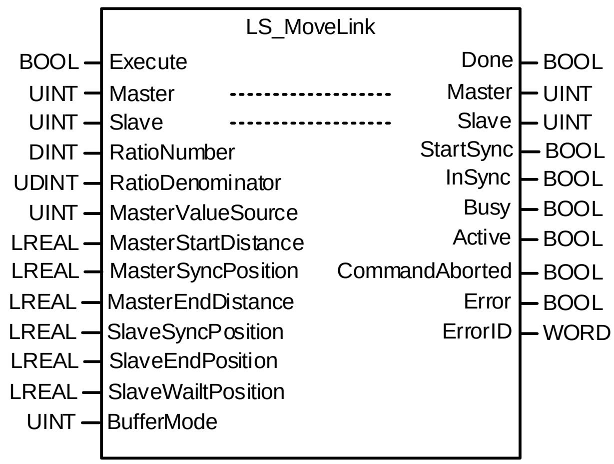


Input-Output		
UINT	Master	Set main axis (refer toSee 6.2.1 Setting Range by Product)
UINT	Slave	Set the serve axis. (refer toSee 6.2.1 Setting Range by Product)
Input		
BOOL	Execute	Give gear operation command to the relevant axis in the rising Edge.
DINT	RatioNumerator	Specify the numerator of gear ratio. (-2147483648 ~ 2147483647)
UDINT	RatioDenominator	Specify the denominator of gear ratio. (0 ~ 4294967295)
UINT	MasterValueSource	Select the standard of the main axis value to be synchronized. 0(mcSetValue): Synchronize in the target position of the main axis. 1(mcActualValue): Synchronize in the current position of the main axis.
LREAL	MasterSyncPosition	Specify the position of the main axis where gear operation starts.
LREAL	SlaveSyncPosition	Specify the position of the spindle where gear operation starts.
LREAL	SyncMode	Unused
LREAL	MasterStartDistance	Specify the distance of the main axis where synchronization starts.
LREAL	Velocity	Specify the maximum speed of the spindle at the beginning of synchronization. [u/s]
LREAL	Acceleration	Specify the maximum acceleration of the spindle at the beginning of synchronization. [u/s^2]
LREAL	Deceleration	Specify the maximum deceleration of the spindle at the beginning of synchronization. [u/s^2]
LREAL	Jerk	Unused
UINT	BufferMode	Specify the sequential operation setting of motion function block. (Refer to 6.1.4 BufferMode input)

Output		
BOOL	InSync	Indicate that gear operation is normally being fulfilled as the specified gear ratio is applied.
BOOL	StartSync	Indicate synchronization is starting.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that the current motion function block is controlling the relevant axis.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted while it is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

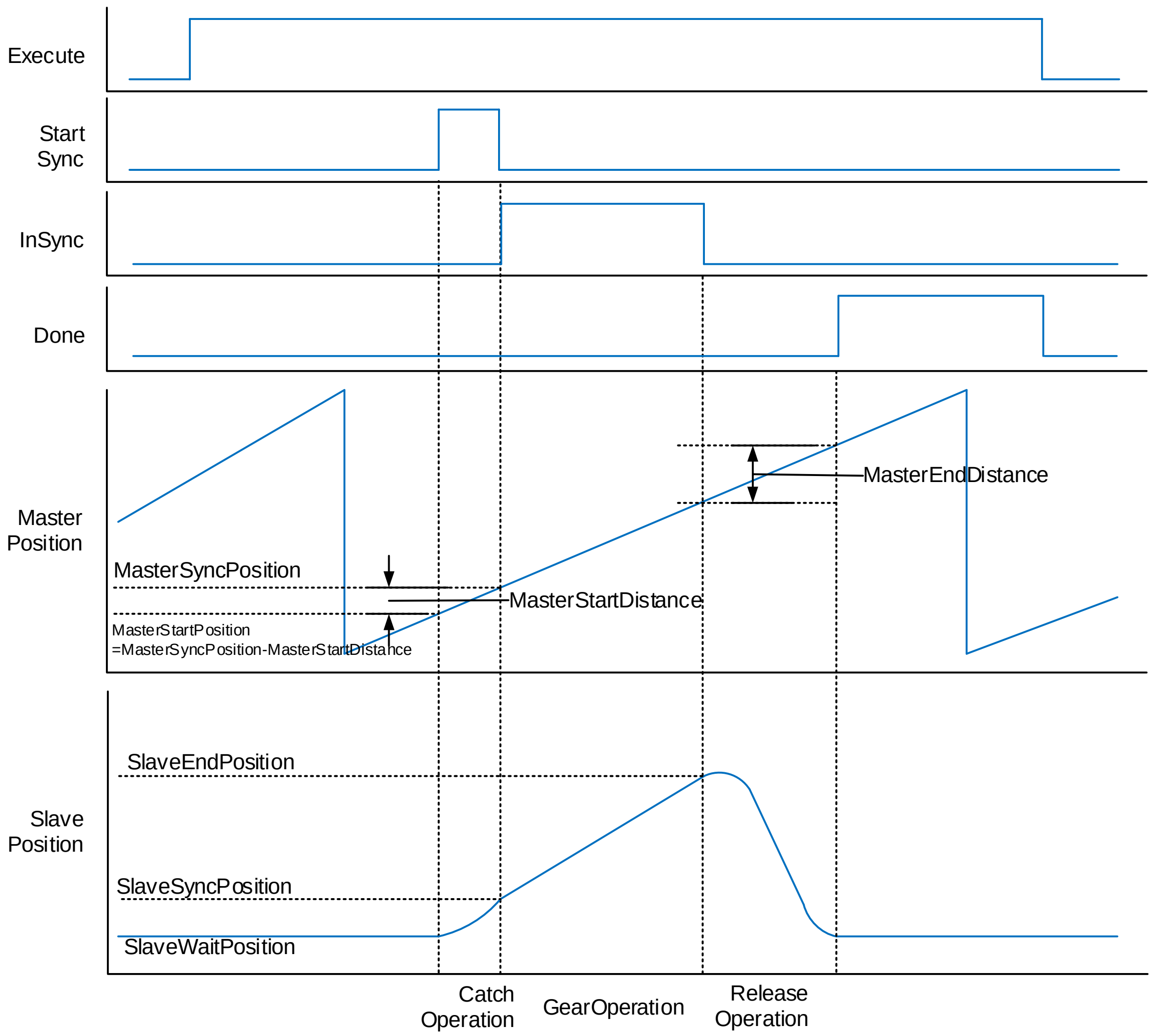
(1) In this motion function block, RatioNumerator input and RatioDenominator input are expanded to DINT and UDINT respectively.
(2) The rest of the settings and functions are the same as those of the MC_GearInPos function block. Please refer to the description of the MC_GearInPos function block.

6.4.16 Link operation (LS_MoveLink)

Motion Function Block		
		
Input-Output		
UINT	Master	Set main axis (1~32: Real/Virtual Axes,33~36: Virtual Axes, 1001~1002: Encoder)
UINT	Slave	Set the serve axis. (1~32: Real/Virtual Axes,33~36: Virtual Axes)
Input		
BOOL	Execute	Give an link operation command to the relevant axis in the rising edge
DINT	RatioNumerator	Specifies the numerator value of the gear ratio.
UDINT	RatioDenominator	Specifies the denominator value of the gear ratio.
UINT	MasterValueSource	Select the standard of the main axis value to be synchronized. 0(mcSetValue): Synchronize in the target position of the main axis. 1(mcActualValue): Synchronize in the current position of the main axis.
LREAL	MasterStartDistance	Specify the move distance of the main axis where synchronization starts.
LREAL	MasterSyncPosition	Specify the position of the main axis where sync. operation starts.
LREAL	MasterEndDistance	Specifies the movement distance of the main axis of operation to stop synchronization. (Movement distance of main axis from SlaveEndPosition to SlaveWaitPosition)
LREAL	SlaveSyncPosition	Specify the position of the main axis where sync. operation starts.
LREAL	SlaveEndPosition	Specify the position of the main axis where sync. operation end.
LREAL	SlaveWaitPosition	Specifies the position where the slave axis waits.
UINT	BufferMode	Specify the sequential operation setting of motion function block. (Refer to 6.1.4 BufferMode input)
Output		
BOOL	Done	Indicate that gear operation is normally being fulfilled as the specified gear ratio is applied.
BOOL	StartSync	Indicates that tracking operation is being performed.
BOOL	InSync	Indicate that gear operation is normally being fulfilled as the specified gear ratio is applied.

BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that the current motion function block is controlling the relevant axis.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted while it is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(5) This motion function block is an operation to synchronize the speed of the main axis and the spindle in the set position depending on gear ratio which is set in the specific position.
(6) In link operation, the main axis must always operate in the forward direction.
(7) Link operation consists of three operations: catch operation, gear operation, and release operation.
(8) Link operation is terminated when the subordinate axis reaches SlaveWaitPosition. To stop the link operation, it is necessary to give a stop (MC_Stop) command to the subordinate axis or operate another motion function block.
(9) RatioNumerator and RatioDenominator set the numerator and denominator of gear ratio to be applied to the spindle respectively. If the numerator is set to negative number, the rotation direction of the spindle goes into reverse of the main axis. If the numerator value is set to 0, “Error 0x1091” occurs.
(10) “Error 0x1192” occurs when the SlaveEndPosition value is set smaller than the SlaveSyncPosition when the gear ratio set by the RatioNumerator and RatioDenominator values is positive.
(11) MasterValueSource selects the source of the main axis to be synchronized. If it is set to 0 (mcSetValue), synchronization is performed by putting the target position of the main axis in the current motion control period as a source, and if it is set to 1(mcActualValue), synchronization is performed by putting the current position of the main axis got feedback from the current motion control period as a source. Other values set besides these two cause "error 0x1193”.
(11) In the MasterSyncPosition input and SlaveSyncPosition input, input the positions of the main and sub-axis, respectively, to complete the following operation and start the gear operation. In the MasterSyncPosition input and SlaveSyncPosition input, input the positions of the main and sub-axis, respectively, to complete the following operation and start the gear operation.
(12) When gear operation starts and the main axis reaches the SlaveEndPosition by the main axis, the gear operation ends and the main axis moves as much as the distance set in the MasterEndDistance input, while the subordinate axis performs the release action to reach the SlaveWaitPositon. When the release operation is completed, the link operation is terminated.
(13) The MasterStartDistance value should always be set to a value greater than 0. If the MasterStartDistance value is set to less than 0, “Error 0x1195” occurs.
(14) If SlaveEndPosition and SlaveWaitPosition are set identically, release operation is performed before reaching SlaveEndPosition. The position of the main axis where the release operation starts is the position obtained by subtracting the movement distance set in the MasterEndDistance input from the position of the main axis corresponding to SlaveEndPosition.
(15) The MasterEndDistance value should always be set to a value greater than 0. If the MasterEndDistance value is set to less than 0, “Error 0x1196” occurs.
(16) Once tracking operation starts, StartSync output is On. When the tracking operation is completed and gear operation starts, the StartSync output is turned off and the InSync output is turned on. When the release operation starts, InSync output is Off. When the position of the sub-axis reaches the SlaveWaitPosition and the release operation ends, the Done output is On.
(17) If this motion function block is aborted by another command (BufferMode=0 of newly executed command), the cam operation is stopped, and the CommandAborted output is on.
(18) If another command is executed by Buffered while this motion function block is being executed (BufferMode=1~5 of newly executed command), the status of gear operation (InGear phase) is terminated, and then the newly executed command is run subsequently. InSync / Busy / Active / CommandAborted / Error output of this function block are all Off.
(19) The serve axis is in 'SynchronizedMotion' while this motion function block is running.


Execute
Start
Sync
InSync
Done
Master
Position
MasterSyncPosition
MasterStartDistance
MasterStartPosition
=MasterSyncPosition-MasterStartDistance
MasterEndDistance
Slave
Position
SlaveEndPosition
SlaveSyncPosition
SlaveWaitPosition
Catch
Operation
GearOperation
Release
Operation

6.5 Group Motion Function Block

6.5.1 Add Group axis (MC_AddAxisToGroup)

Motion Function Block

```
                      MC_AddAxisToGroup
BOOL ─ Execute                              Done ─ BOOL
UINT ─ AxesGroup ------------------ AxesGroup ─ UINT
UINT ─ Axis ---------------------------------- Axis ─ UINT
UINT ─ IdentInGroup                         Busy ─ BOOL
                                            Error ─ BOOL
                                          ErrorID ─ WORD
```

Input-Output		
UINT	AxesGroup	Set the group where the relevant axis is added. (1 ~ 16: 1Group ~ 16 Group)
UINT	Axis	Set the axis to be added to the relevant group. (refer toSee 6.2.1 Setting Range by Product)
Input		
BOOL	Execute	Give group axis addition command to the relevant axis in the rising Edge.
UINT	IdentInGroup	Set the ID of the relevant axis to be used in the relevant group. (1 ~ 10)
Output		
BOOL	Done	Indicate the completion state of motion function block.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block adds Axis specified axis to the axis group specified in AxesGroup input.

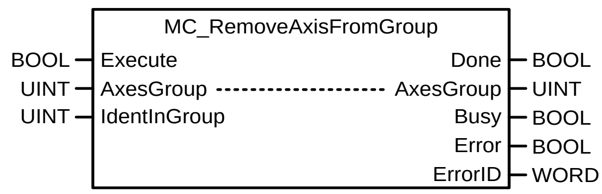
(2) ID in the axis group specified to IdentInGroup must have unique value for each axis. (ID of each axis must be different.) Maximum 10 axes can be included in each axis group, axis ID can be specified in the range of 1~10. If the specified axis number is outside the range, “error 0x0006” occurs, and if numbers in the axis group overlap, “error 0x2051” occurs.

(3) Axis group setting can be performed in the same way at XG5000 axis group parameter setting.

Group	Name	Axis group 1
Axis Group Parameter	Axis 01	0: None
	Axis 02	0: None
	Axis 03	0: None
	Axis 04	0: None
	Axis 05	0: None
	Axis 06	0: None
	Axis 07	0: None
	Axis 08	0: None
	Axis 09	0: None
	Axis 10	0: None
	Intp. speed Max	20000000 u/s

6.5.2 Remove Group axis (MC_RemoveAxisFromGroup)

Motion Function Block



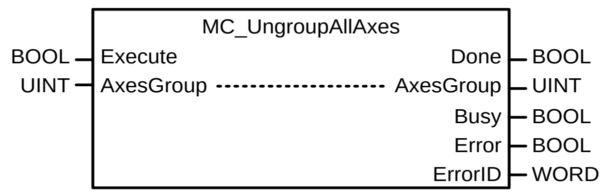
Input-Output		
UINT	AxesGroup	Set the group where the relevant axis is removed. (1 ~ 16: 1Group ~ 16 Group)
Input		
BOOL	Execute	Give group axis exclusion command to the relevant group in the rising Edge.
UINT	IdentInGroup	Set the axis number in the relevant group to be removed from the relevant group. (1 ~ 10)
Output		
BOOL	Done	Indicate the completion state of motion function block.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block removes the axis which is specified to IdentInGroup in the axis group specified in AxesGroup input.

(2) If the execution of group axis exclusion is tried when the axis group is not in GroupDisabled, GroupStandBy, and GroupErrorStop state, "error 0x2003 or 0x2004 or 0x2005" occurs and the axis is not removed. In other words, the axis cannot be removed when the axis group does not completely stop.

6.5.3 Removes all axes from the group (MC_UngroupAllAxes)

Motion Function Block



Input-Output		
UINT	AxesGroup	Set the group where every axis is to be removed. (1 ~ 16: 1Group ~ 16 Group)
Input		
BOOL	Execute	Give MC_UngroupAllAxes command to the relevant group in the rising Edge.
Output		
BOOL	Done	Indicate the completion state of motion function block.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

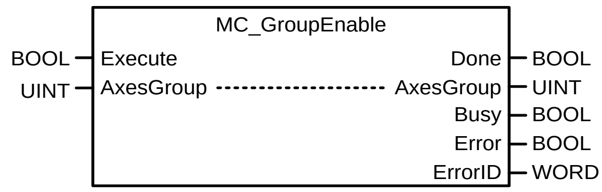
(1) This motion function block removes every axis which belongs to the axis group specified in AxesGroup input.

(2) If this motion function block is executed when the axis group is not in GroupDisabled, GroupStandBy, and GroupErrorStop state, "error 0x2003 or 0x2004 or 0x2005" occurs and the axis is not removed. In other words, the axis cannot be removed when the axis group does not completely stop.

(3) When the axis which belongs to the group is successfully removed, the relevant group is switched to GroupDisabled state.

6.5.4 Group Enable (MC_GroupEnable)

Motion Function Block

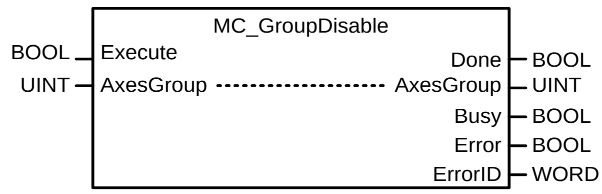


Input-Output		
UINT	AxesGroup	Set the group to be activated. (1 ~ 16: 1Group ~ 16 Group)
Input		
BOOL	Execute	Give group activation command to the relevant group in the rising Edge.
Output		
BOOL	Done	Indicate the completion state of motion function block.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block is to activate the axis group specified in AxesGroup input.
(2) When giving this command to the axis group in GroupDisable state, the relevant axis group is switched to GroupStandby state.
(3) This motion function block does not affect the power state of each axis in the relevant group.

6.5.5 Group Disable (MC_GroupDisable)

Motion Function Block

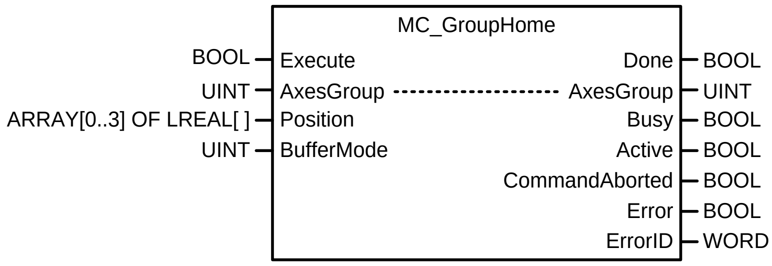


Input-Output		
UINT	AxesGroup	Set the group to be deactivated. (1 ~ 16: 1Group ~ 16 Group)
Input		
BOOL	Execute	Give group disablement command to the relevant group in the rising Edge.
Output		
BOOL	Done	Indicate the completion state of motion function block.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block is to deactivate the axis group specified in AxesGroup input.
(2) The axis group which executes this motion function block is switched to GroupDisabled.
(3) This motion function block does not affect the power state of each axis in the relevant group.

6.5.6 Performs the search home of all axes in the group (MC_GroupHome)

Motion Function Block

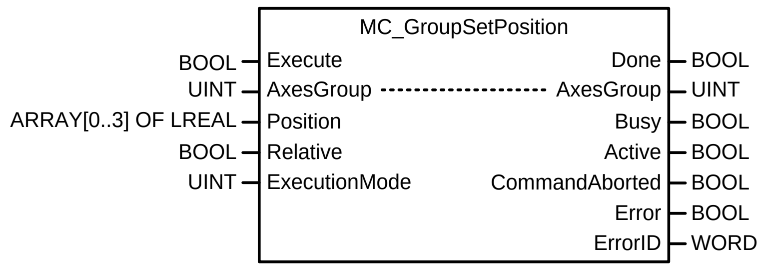


Input-Output		
UINT	AxesGroup	Set the group returning to home. (1 ~ 16: 1Group ~ 16 Group)
Input		
BOOL	Execute	Give group homing command to the relevant group in the rising Edge.
LREAL[ ]	Position	Specify the absolute position of each axis when reference signal is detected.
UINT	BufferMode	Specify the sequential operation setting of motion function block. (Refer to 6.1.4.BufferMode input)
Output		
BOOL	Done	Indicate the completion state of motion function block.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that the current motion function block is controlling the relevant axis.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted while it is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block is to give homing command to the axis group specified in AxesGroup input.
(2) Homing method is operated as specified in servo parameter of the relevant axis in advance.
(3) In Position input, specify the absolute position to the array to be set when homing is completed or Reference Signal is detected. Values in the array and the axis in the group correspond in the order of [①, ②, … ⑨, ⑩]. (①~⑩ are the axis ID in the axis group)
(4) The axis group is in 'GroupHoming' state while this motion function block is running, and it is switched to 'GroupStandby' state when motion function block is completed.
(5) If the function block is re-executed (Execute input is On) before the instruction is terminated, the changed parameters are applied.
Only Position input can be updated.

6.5.7 Sets the position of all axes in the group without moving (MC_GroupSetPosition)

Motion Function Block



Input-Output		
UINT	AxesGroup	Select the group to set the current position. (1 ~ 16: 1Group ~ 16 Group)
Input		
BOOL	Execute	Give group current position setting command to the relevant group in the rising Edge.
LREAL[ ]	Position	Specify the position.
BOOL	Relative	0: Position value=Absolute position, 1:Position value=Incremental position
UINT	ExecutionMode	0: Immediately applied the position value, 1: Applied at the same point with 'Buffered' of Buffermode
Output		
BOOL	Done	Indicate the completion state of motion function block.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that the current motion function block is controlling the relevant axis.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted while it is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block sets the current position of the relevant axis group.

(2) Specify the position of each axis in the group to the array. When executing this motion function block, if Relative input is Off, the position of the relevant axis is replaced by the Position input value, and if Relative input is On, the Position input value is added to the current position of the relevant axis. Values in the array and the axis in the group correspond in the order of [①, ②, … ⑨, ⑩]. (①~⑩ are the axis ID in the axis group)

(3) ExecutionMode input specifies the setting point. When the input is 0, setting is performed upon executing the command. When the input is 1, setting is performed at the same time as "Buffered" at the sequential operation. The value unable to be set causes "error 0x201B".

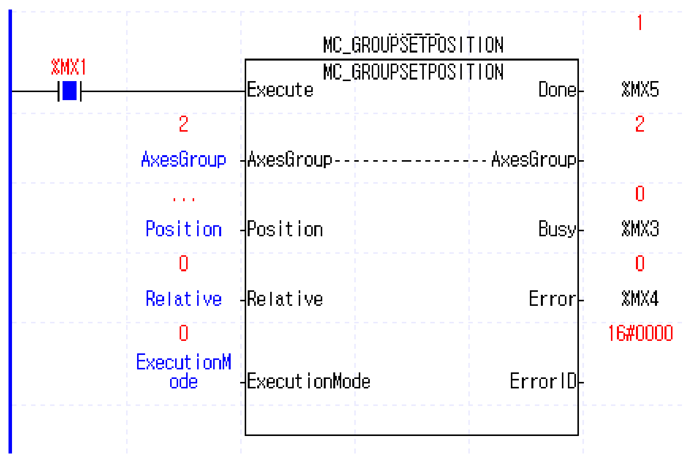
0 (mcImmediately): Change the value of parameter immediately after the execution of motion function block (rising Edge in Execute input). If the relevant axis is in running, operation can be affected.

1 (mcQueued): Changed at the same point with 'Buffered' in Buffermode. (Refer to 6.1.4.BufferMode input).

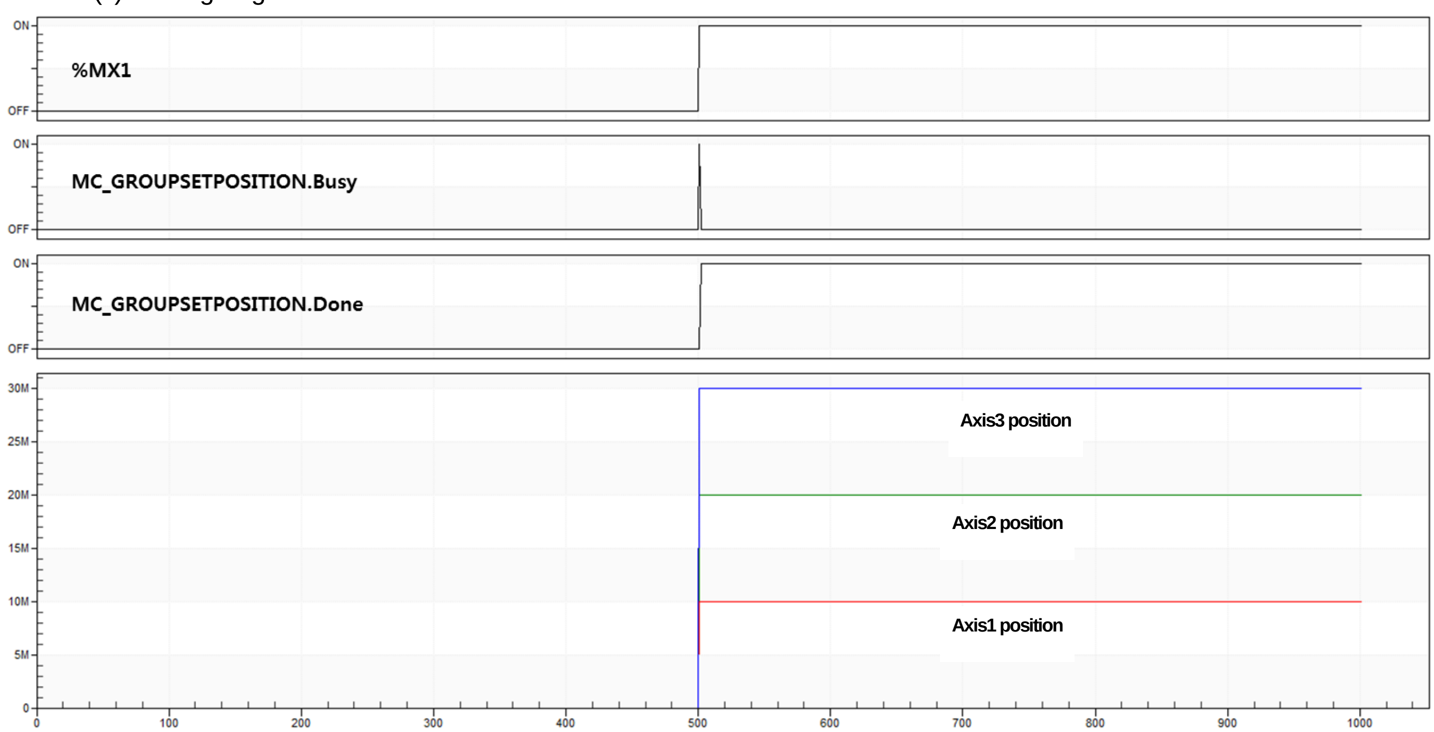
(4) Example program

This example shows the change of the current position to position values (10,000,000/20,000,000/30,000,000) set in the position variables when executing MC_GroupSetPosition function block at the status where 1-axis, 2-axis and 3-axis are set as a single group.

(a) Function block setting

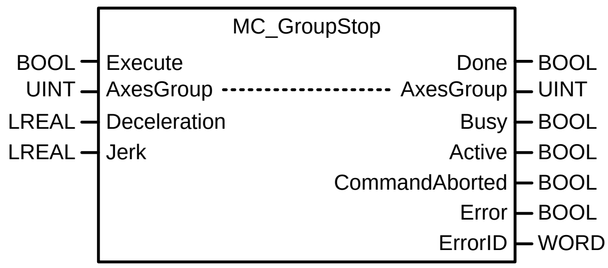


(b) Timing diagram



6.5.8 Stop the group immediately (MC_GroupStop)

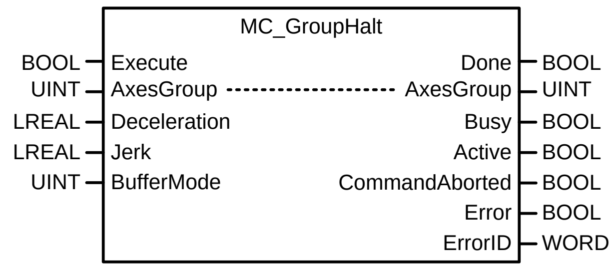
Motion Function Block



Input-Output		
UINT	AxesGroup	Set the group to stop immediately. (1 ~ 16: 1Group ~ 16 Group)
Input		
BOOL	Execute	Give group immediate stop command to the relevant group in the rising Edge.
LREAL	Deceleration	Specify the deceleration in time of stop. [u/s^2]
LREAL	Jerk	Specify the change rate of acceleration/deceleration. [u/s^3]
Output		
BOOL	Done	Indicate the completion state of motion function block.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that the current motion function block is controlling the relevant axis.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted while it is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block is to give an emergency stop command to the relevant axis group.
(2) The relevant axis group moves on the route which it was following until it completely stops.
(3) When executing group immediate stop (MC_GroupStop) motion function block, motion function block which the relevant axis group is performing is interrupted, and the axis is changed to 'GroupStopping'. When the relevant axis group is in ‘GroupStopping’ state, other motion function block cannot be given to the relevant axis until the stop is completed (until Done output is On).
(4) CommandAborted output indicates that the current motion function block is interrupted while it was executed. Because other motion function block cannot interrupt group immediate stop (MC_GroupStop) command while group immediate stop (MC_GroupStop) command is being executed, CommandAborted output is On when the power of servo is cut, servo Off command is executed, or servo connection is disconnected.
(5) If Execute input is On or the speed of the axis is not 0, the axis is in ' GroupStopping' state, and if Done output is On and Execute input is Off, the axis is switched to ' GroupStandBy' state.
(6) If the function block is re-executed (Execute input is On) before the instruction is terminated, the changed parameters are applied.
Deceleration, Jerk input can be updated.

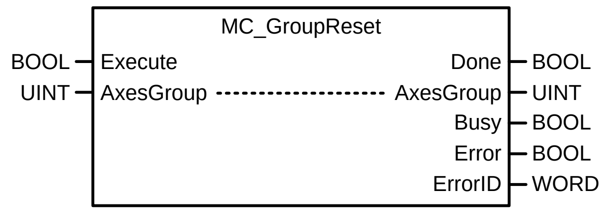
6.5.9 Stop the group (MC_GroupHalt)

Motion Function Block		
		
Input-Output		
UINT	AxesGroup	Set the group to stop. (1 ~ 16: 1Group ~ 16 Group)
Input		
BOOL	Execute	Give group stop command to the relevant group in the rising Edge.
LREAL	Deceleration	Specify the deceleration in time of stop. [u/s^2]
LREAL	Jerk	Specify the change rate of acceleration/deceleration. [u/s^3]
UINT	BufferMode	Specify the sequential operation setting of motion function block. (Refer to 6.1.4.BufferMode input)
Output		
BOOL	Done	Indicate the completion state of motion function block.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that the current motion function block is controlling the relevant axis.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted while it is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block is to give a stop command to the relevant axis.
(2) The relevant axis group moves on the route which it was following until it completely stops.
(3) The axis is in 'GroupMoving' state while this motion function block is running, and if the axis group completely stops, 'Done' output is On and the group state is changed to 'GroupStandBy' state.
(4) If the function block is re-executed (Execute input is On) before the instruction is terminated, the changed parameters are applied.
Only Deceleration and Jerk input can be updated

6.5.10 Reset the group error (MC_GroupReset)

Motion Function Block



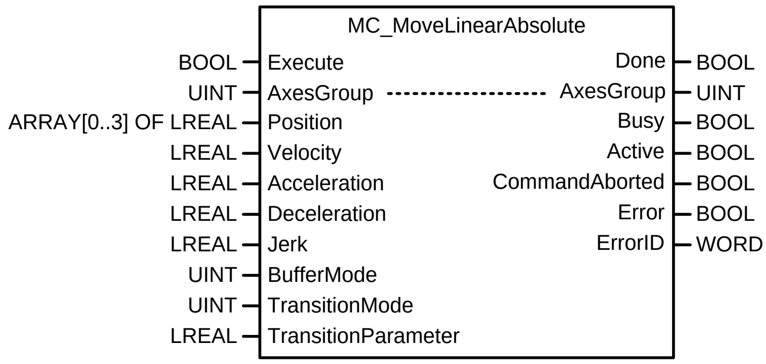
Input-Output		
UINT	AxesGroup	Set the group to do error reset. (1 ~ 16: 1Group ~ 16 Group)
Input		
BOOL	Execute	Give group error reset command to the relevant group in the rising Edge.
Output		
BOOL	Done	Indicate the completion state of motion function block.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block is to reset the error of the relevant axis group. When the relevant axis is in 'GroupErrorStop', the execution of motion function block resets the error occurred in the current relevant axis and switches the axis group to 'GroupStandBy' state.

(2) When executing this motion function block, every error occurred in each axis in the group is reset. (This has the same effect with when executing the axis error reset (MC_Reset) command in each axis.)

6.5.11 Absolute position linear Interpolation operation (MC_MoveLinearAbsolute)

Motion Function Block



Input-Output		
UINT	AxesGroup	Set the group to perform absolute position linear interpolation operation. (1 ~ 16: 1Group ~ 16 Group)
Input		
BOOL	Execute	Give absolute position linear interpolation operation command to the relevant group in the rising Edge.
LREAL[ ]	Position	Specify the target position of each axis.
LREAL	Velocity	Specify the maximum speed of the route. [u/s]
LREAL	Acceleration	Specify the maximum acceleration. [u/s^2]
LREAL	Deceleration	Specify the maximum deceleration. [u/s^2]
LREAL	Jerk	Specify the change rate of acceleration/deceleration. [u/s^3]
UINT	BufferMode	Specify the sequential operation setting of motion function block. (Refer to 6.1.4.BufferMode input)
UINT	TransitionMode	Specify the route change mode of group operation. (Refer to to 6.1.6.TransitionMode)
LREAL	TransitionParameter	Specify the parameter of the route change setting of group operation. (Refer to to 6.1.6.TransitionMode)
Output		
BOOL	Done	Indicate whether to reach the specified position.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that whether or not motion function block is controlling the group.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted while it is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block is to give an absolute position linear interpolation command to the axis group specified in AxesGroup input.

(2) When this motion function block is executed, interpolation control is performed in a linear path from the current position to the target position of each axis, and the moving direction is decided by the starting point and the target point of each axis.

Beginning position < Target position: Forward direction operation

Beginning position > Target position: Reverse direction operation

(3) In Position input, specify the target position of each axis in the group as matrix. The values in the array and the axis in the group correspond in the order of[①, ②, … ⑨, ⑩]. (①~⑩ are the axis ID in the axis group)

(4) Specify the speed, acceleration, deceleration, and the change rate of acceleration/deceleration of interpolation route in Velocity, Acceleration, Deceleration, and Jerk inputs respectively.

(5) Velocity is to set the interpolation speed of the axis group, and it indicates the integrated speed of each axis.

The operation speed of each configuration axis is calculated as follows.

$$\text{Interpolation speed (F)} = \text{Target speed specified in the Velocity}$$

$$\text{Interpolation movement amount (S)} = \sqrt{S_1^{\,2} + S_2^{\,2} + \ldots + S_3^{\,2} + S_4^{\,2}}$$

$$\text{Configuration axis 1 speed } (V_1) = \text{Interpolation speed (F)} \times \frac{\text{Configuration axis 1 movement amount } (S_1)}{\text{Interpolation movement amount (S)}}$$

$$\text{Configuration axis 2 speed } (V_2) = \text{Interpolation speed (F)} \times \frac{\text{Configuration axis 2 movement amount } (S_2)}{\text{Interpolation movement amount (S)}}$$

$$\text{Configuration axis 3 speed } (V_3) = \text{Interpolation speed (F)} \times \frac{\text{Configuration axis 3 movement amount } (S_3)}{\text{Interpolation movement amount (S)}}$$

$$\text{Configuration axis 4 speed } (V_4) = \text{Interpolation speed (F)} \times \frac{\text{Configuration axis 4 movement amount } (S_4)}{\text{Interpolation movement amount (S)}}$$

(6) Refer to chapter 8.2.6 linear interpolation control part in motion controller's manual for more details.

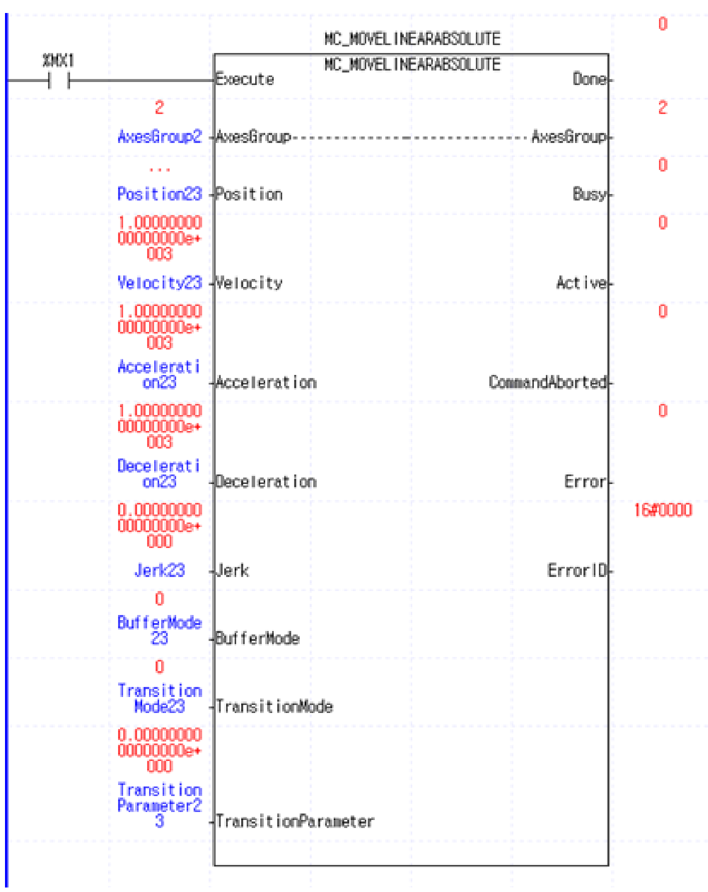
(7) If the function block is re-executed (Execute input is On) before the instruction is terminated, the changed parameters are applied. Only Velocity, Acceleration, Deceleration, Jerk, Position input can be updated.

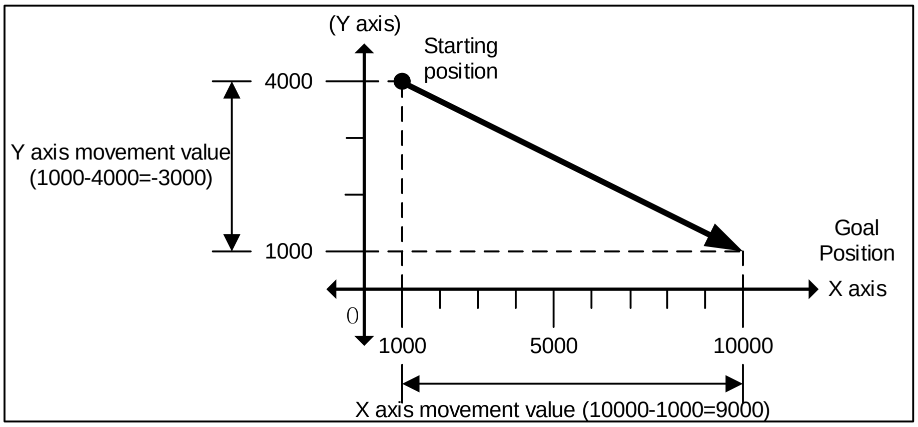
(8) Velocity input can be set to 0 or changed.

(9) Example program

This example shows the linear interpolation to the target position (10000, 1000) when the current command position is (1000, 4000).

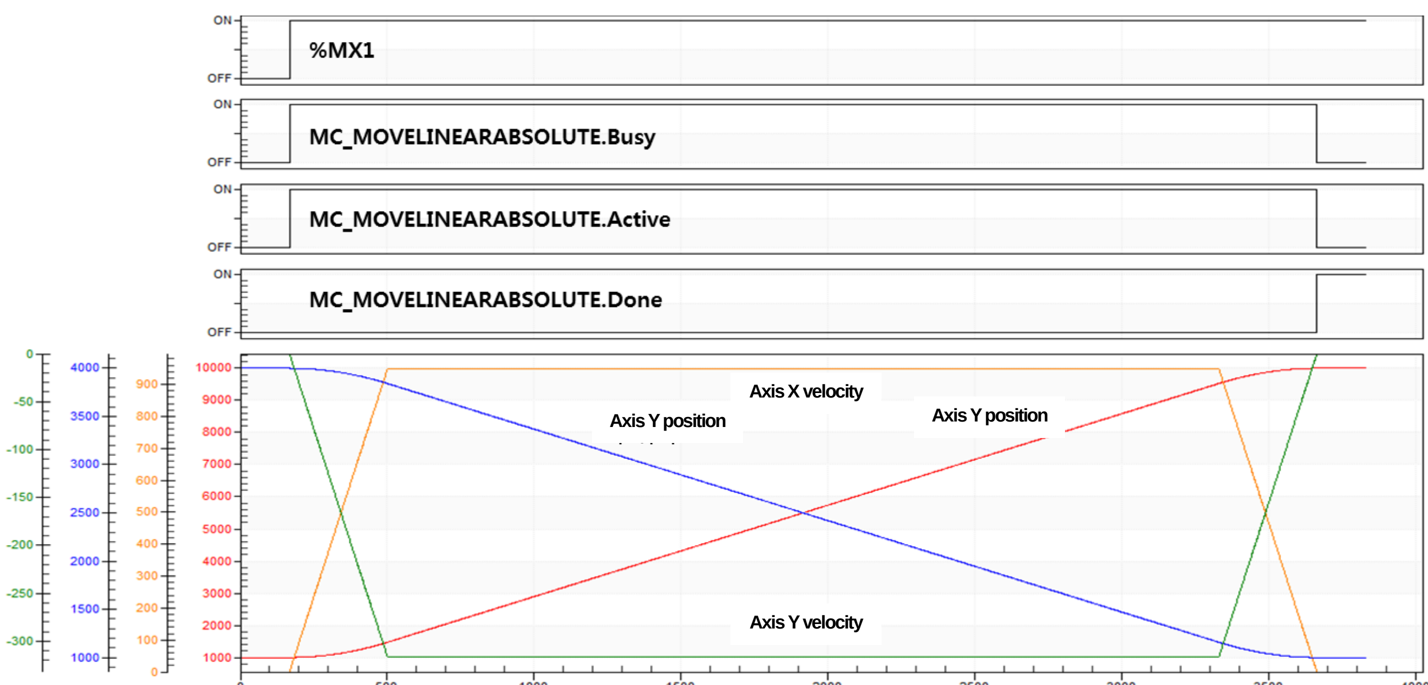
(a) Function block setting



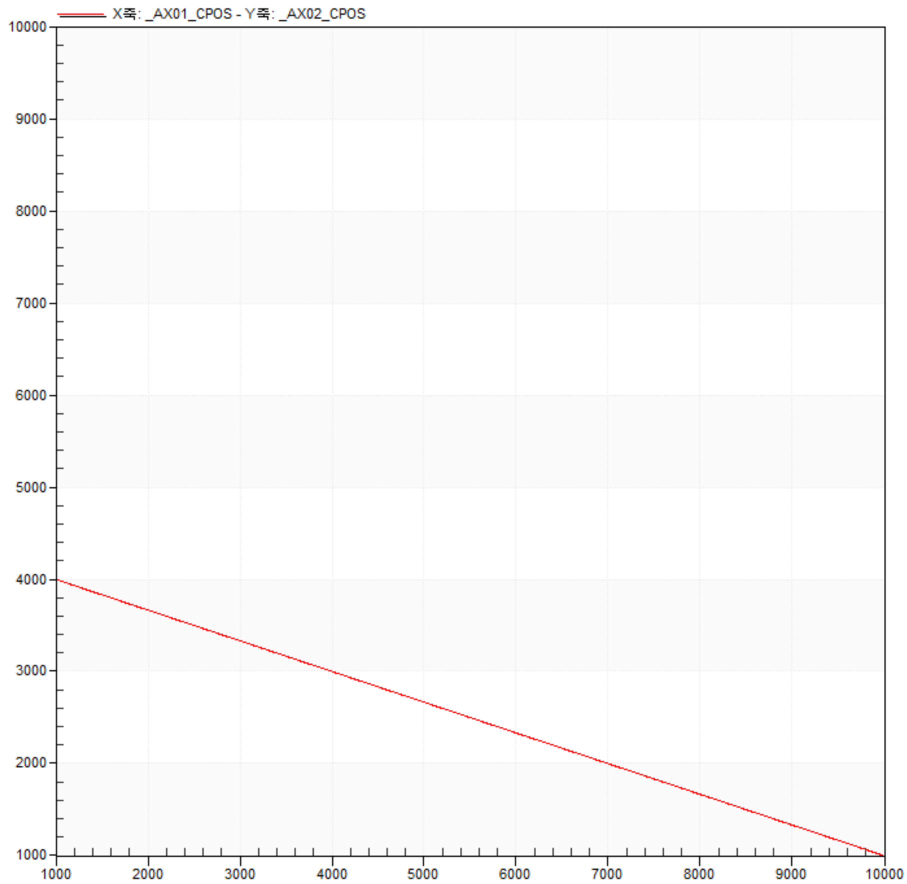


1	<GLOBAL>	%JL1.3	1.0000000000000000e+003	LREAL	_AX01_CPOS
2	<GLOBAL>	%JL2.3	4.0000000000000000e+003	LREAL	_AX02_CPOS
3	<GLOBAL>	%JL1.4	0.0000000000000000e+000	LREAL	_AX01_CVEL
4	<GLOBAL>	%JL2.4	0.0000000000000000e+000	LREAL	_AX02_CVEL
5	Group	GroupHome	On	BOOL	
6	Group	GourpHalt	Off	BOOL	
7	Group	⊟ Position23		ARRAY[0..3] OF LREAL	
8	Group	Position23 [0]	1.0000000000000000e+004	LREAL	
9	Group	Position23 [1]	1.0000000000000000e+003	LREAL	
10	Group	Position23 [2]	0.0000000000000000e+000	LREAL	
11	Group	Position23 [3]	0.0000000000000000e+000	LREAL	
12					

(b) Timing diagram

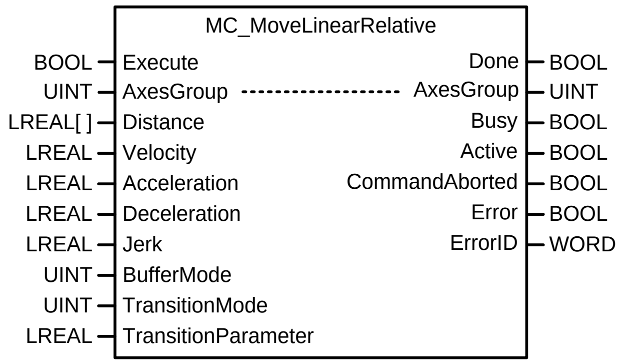


(c) XY graph


X축: _AX01_CPOS - Y축: _AX02_CPOS
10000
9000
8000
7000
6000
5000
4000
3000
2000
1000
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000

6.5.12 Incremental position linear Interpolation operation (MC_MoveLinearRelative)

Motion Function Block



Input-Output		
UINT	AxesGroup	Set the group to do relative position linear interpolation operation. (1 ~ 16: 1Group ~ 16 Group)
Input		
BOOL	Execute	Give relative position linear interpolation operation command to the relevant group in the rising Edge.
LREAL[ ]	Distance	Set the target distance of each axis.
LREAL	Velocity	Specify the maximum speed of the route. [u/s]
LREAL	Acceleration	Specify the maximum acceleration. [u/s^2]
LREAL	Deceleration	Specify the maximum deceleration. [u/s^2]
LREAL	Jerk	Specify the change rate of acceleration/deceleration. [u/s^3]
UINT	BufferMode	Specify the sequential operation setting of motion function block. (Refer to 6.1.4.BufferMode input)
UINT	TransitionMode	Specify the route change mode of group operation. (Refer to 10.1.6.TransitionMode)
LREAL	TransitionParameter	Specify the parameter of the route change setting of group operation. (Refer to 10.1.6.TransitionMode)
Output		
BOOL	Done	Indicate whether to reach the specified position.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that whether or not motion function block is controlling the group.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted while it is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block is to give a relative position linear interpolation command to the axis group specified in AxesGroup input.

(2) When this motion function block is executed, interpolation control performed in a linear path from the current position to the target position of each axis, and the moving direction is decided by the sign of the target distance of each axis.

Target distance > 0: Forward direction operation

Target distance < 0: Reverse direction operation

(3) In Distance input, specify the target distance of each axis in the group as array. The specified array and the axis in the group correspond in the order of specified axis ID [ID1 target distance, ID2 target distance, …].

(4) Set the speed, acceleration, deceleration, and the change rate of acceleration/deceleration of interpolation route in Velocity, Acceleration, Deceleration, and Jerk inputs respectively.

(5) Velocity is to set the interpolation speed of the axis group, and it indicates the integrated speed of each axis.

The operation speed of each configuration axis is calculated as follows.

$$\text{Interpolation speed (F)} = \text{Target speed specified in the Velocity}$$

$$\text{Interpolation movement amount (S)} = \sqrt{S_1^2 + S_2^2 + \ldots + S_3^2 + S_4^2}$$

$$\text{Configuration axis 1 speed } (V_1) = \text{Interpolation speed (F)} \times \frac{\text{Configuration axis 1 movement amount } (S_1)}{\text{Interpolation movement amount (S)}}$$

$$\text{Configuration axis 2 speed } (V_2) = \text{Interpolation speed (F)} \times \frac{\text{Configuration axis 2 movement amount } (S_2)}{\text{Interpolation movement amount (S)}}$$

$$\text{Configuration axis 3 speed } (V_3) = \text{Interpolation speed (F)} \times \frac{\text{Configuration axis 3 movement amount } (S_3)}{\text{Interpolation movement amount (S)}}$$

$$\text{Configuration axis 4 speed } (V_4) = \text{Interpolation speed (F)} \times \frac{\text{Configuration axis 4 movement amount } (S_4)}{\text{Interpolation movement amount (S)}}$$

(6)Refer to chapter 8.2.6 linear interpolation control part in motion controller's manual for more details.

(7) If the function block is re-executed (Execute input is On) before the instruction is terminated, the changed parameters are applied.

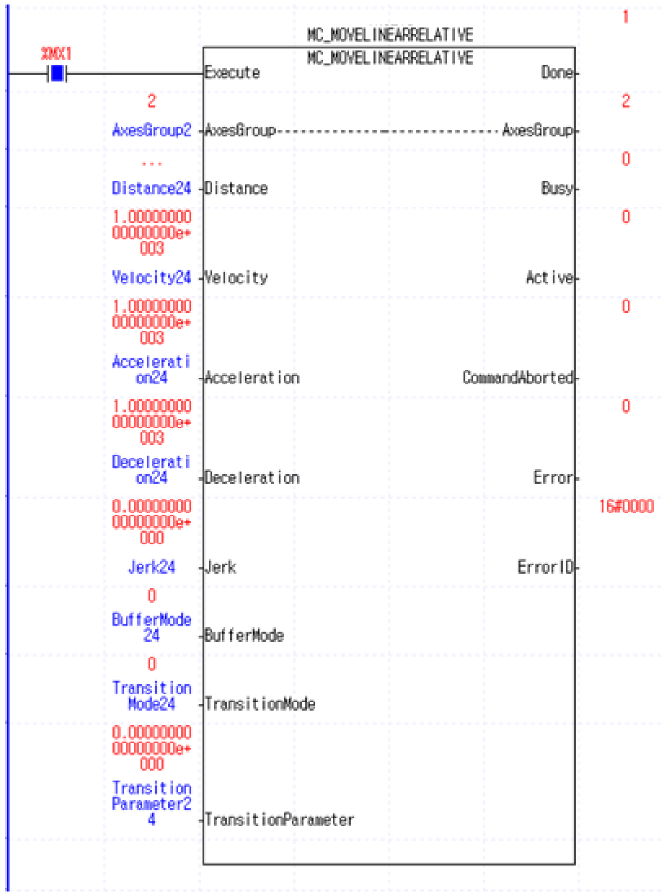
Only Velocity, Acceleration, Deceleration, Jerk, Position input can be updated.

(8) Velocity input can be set to 0 or changed.

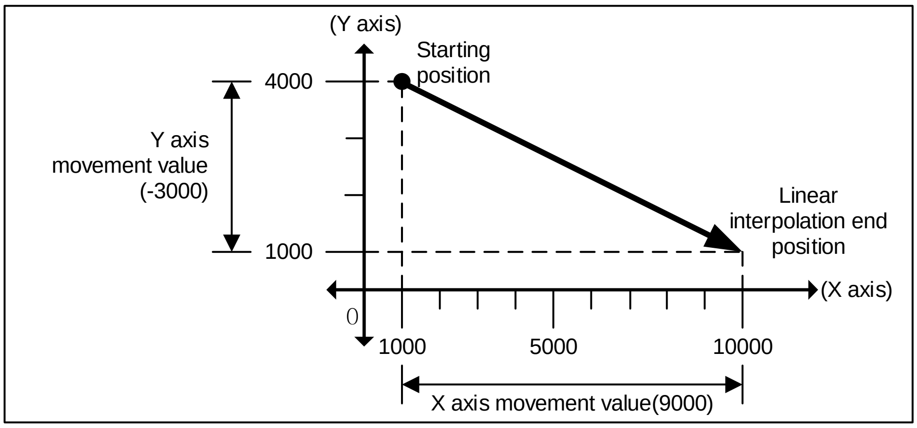
(9) Example program

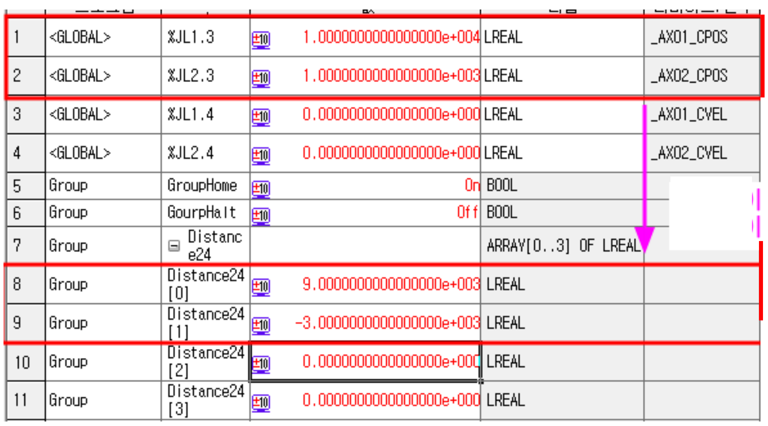
This example shows the linear interpolation to the target position (10000, 10000) by moving the target distance (X-axis: 9000, Y-axis: -3000) when the current command position is (1000, 4000).

(a) Function block setting



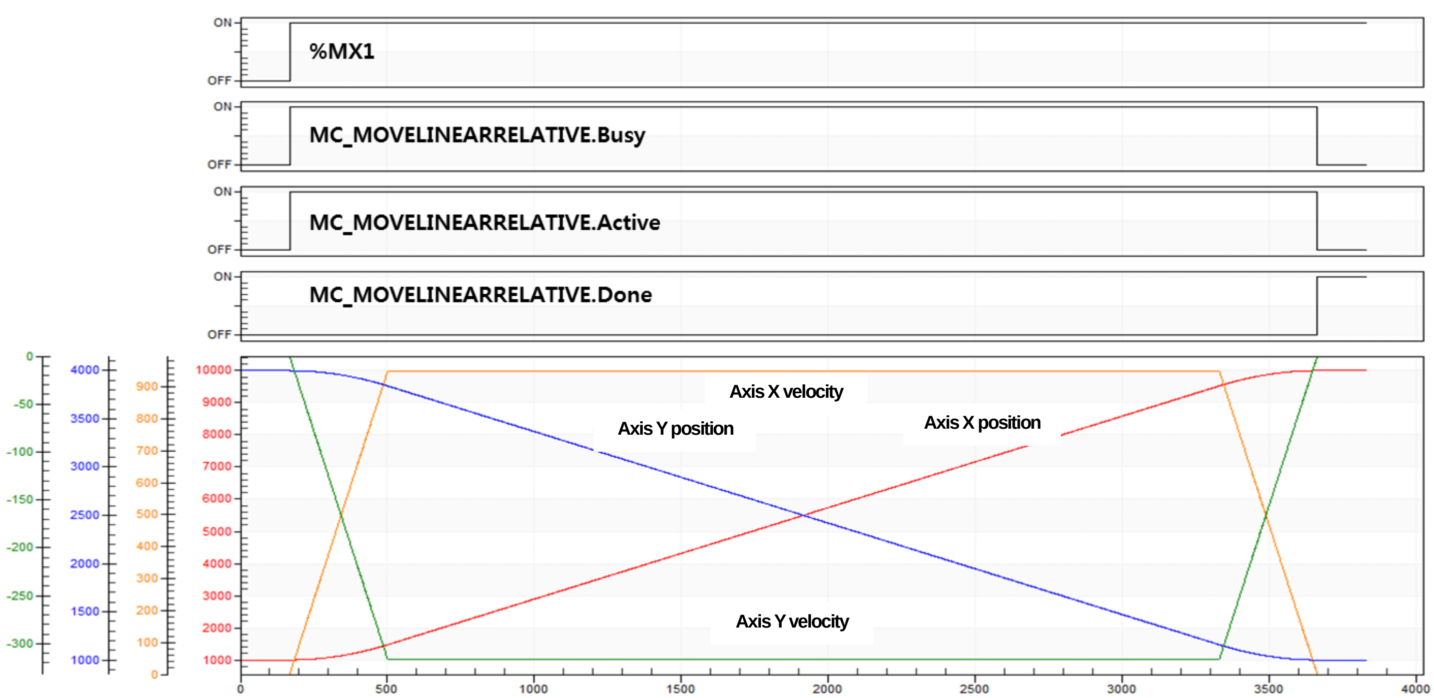
%MX1
MC_MOVELINEARRELATIVE
MC_MOVELINEARRELATIVE
Execute
Done
1
2
AxesGroup2
AxesGroup
AxesGroup
2
...
Distance24
Distance
Busy
0
1.00000000
00000000e+
003
Velocity24
Velocity
Active
0
1.00000000
00000000e+
003
Acceleration24
Acceleration
CommandAborted
0
1.00000000
00000000e+
003
Deceleration24
Deceleration
Error
0
0.00000000
00000000e+
000
Jerk24
Jerk
ErrorID
16#0000
0
BufferMode24
BufferMode
0
TransitionMode24
TransitionMode
0.00000000
00000000e+
000
TransitionParameter24
TransitionParameter



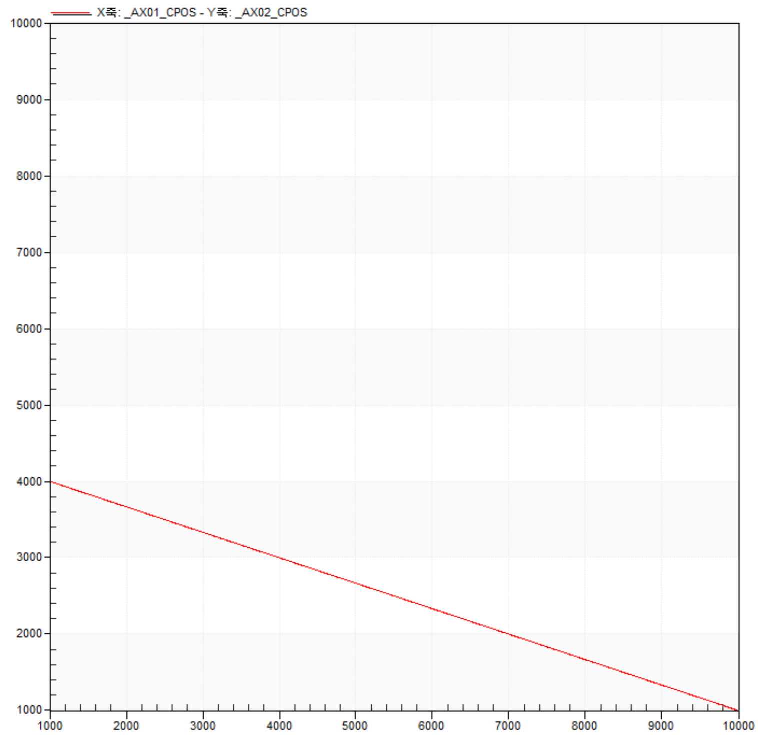


1	<GLOBAL>	%JL1.3	1.0000000000000000e+004	LREAL	_AX01_CPOS
2	<GLOBAL>	%JL2.3	1.0000000000000000e+003	LREAL	_AX02_CPOS
3	<GLOBAL>	%JL1.4	0.0000000000000000e+000	LREAL	_AX01_CVEL
4	<GLOBAL>	%JL2.4	0.0000000000000000e+000	LREAL	_AX02_CVEL
5	Group	GroupHome	On	BOOL	
6	Group	GourpHalt	Off	BOOL	
7	Group	Distance24		ARRAY[0..3] OF LREAL	
8	Group	Distance24[0]	9.0000000000000000e+003	LREAL	
9	Group	Distance24[1]	-3.0000000000000000e+003	LREAL	
10	Group	Distance24[2]	0.0000000000000000e+000	LREAL	
11	Group	Distance24[3]	0.0000000000000000e+000	LREAL	

(b) Timing diagram

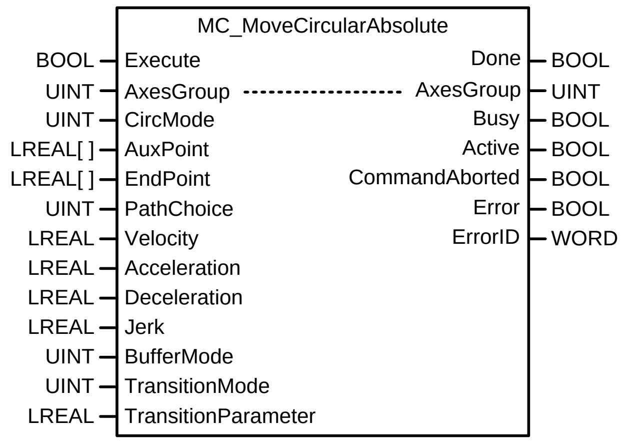


(c) XY graph


X축: _AX01_CPOS - Y축: _AX02_CPOS
10000
9000
8000
7000
6000
5000
4000
3000
2000
1000
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000

6.5.13 Absolute position circular Interpolation operation (MC_MoveCircularAbsolute)

Motion Function Block



Input-Output		
UINT	AxesGroup	Set the group to do absolute position circular interpolation operation. (1 ~ 16: 1Group ~ 16 Group)
Input		
BOOL	Execute	Give absolute position circular interpolation operation command to the relevant group in the rising Edge.
UINT	CirMode	Circular interpolation method setting [0: Midpoint, 1: Central point, 2: Radius]
LREAL[ ]	AuxPoint	Specify the position of auxiliary point depending on the circular interpolation method in an absolute coordinate.
LREAL[ ]	EndPoint	Set the circular end point as an absolute coordinate.
BOOL	PathChoice	Circular route selection 0: clockwise direction, 1: counter-clockwise direction
LREAL	Velocity	Specify the maximum speed of the route. [u/s]
LREAL	Acceleration	Specify the maximum acceleration. [u/s^2]
LREAL	Deceleration	Specify the maximum deceleration. [u/s^2]
LREAL	Jerk	Specify the change rate of acceleration/deceleration. [u/s^3]
UINT	BufferMode	Specify the sequential operation setting of motion function block. (Refer to 6.1.4.BufferMode input)
UINT	TransitionMode	Unused
LREAL	TransitionParameter	Unused
Output		
BOOL	Done	Indicate whether to reach the specified position.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that whether or not motion function block is controlling the group.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted while it is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block is to give an absolute position circular interpolation command to the axis group specified in AxesGroup input.

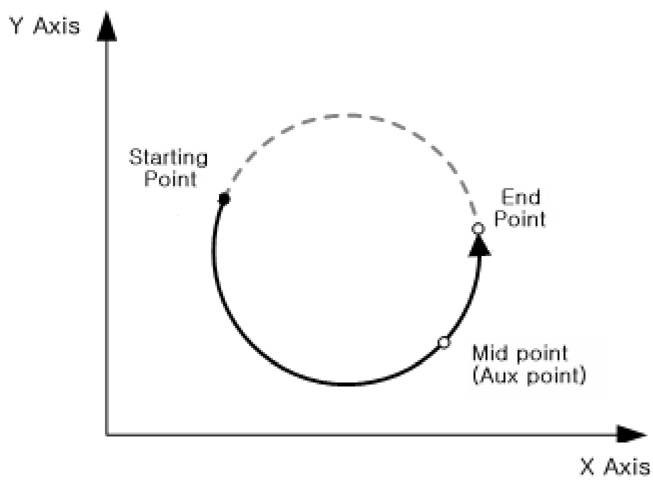
(2) When this motion function block starts, each axis performs circular trajectory interpolation control referring to the auxiliary point input, and the movement direction is determined by Path Choice input. If PathChoice input is set to 0, circular interpolation is operated in a clockwise direction, and if it is set to 1, circular interpolation is operated in a counter-clockwise direction.

(3) Specify the absolute position of the auxiliary point to refer when doing circular interpolation of each axis in AuxPoint and EndPoint inputs as array. The input arrangement and the axes of the group correspond to the designated axis IDs [ID1, ID2, ID3, ···], in that order. (The 3 LEAL type sized array should be entered in Position input as there are 3 axes which comprise the group to give a circular interpolation operation command.)

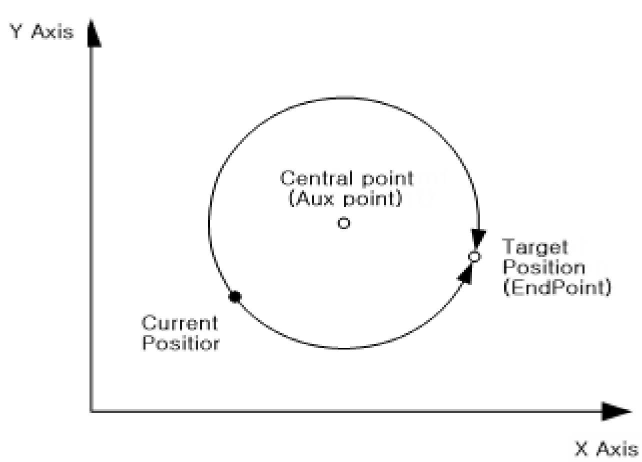
(4) In Velocity, Acceleration, Deceleration, Jerk inputs, the acceleration, deceleration, change rate of acceleration, velocity of the interpolation path are specified, respectively.

(5) Set the circular interpolation method in CircMode input. The circular interpolation methods corresponding to CircMode values are as follows.

(a) Circular Interpolation Using Midpoint Specification (CircMode = 0)
This method performs circular interpolation by starting operation at the start position, passing the designated midpoint, and reaching the target position. In the figure below, the start position corresponds to the axes group coordinate at the start of the command, the midpoint corresponds to the coordinate input for the AuxPoint, and the target position corresponds to the absolute coordinate input for the EndPoint.

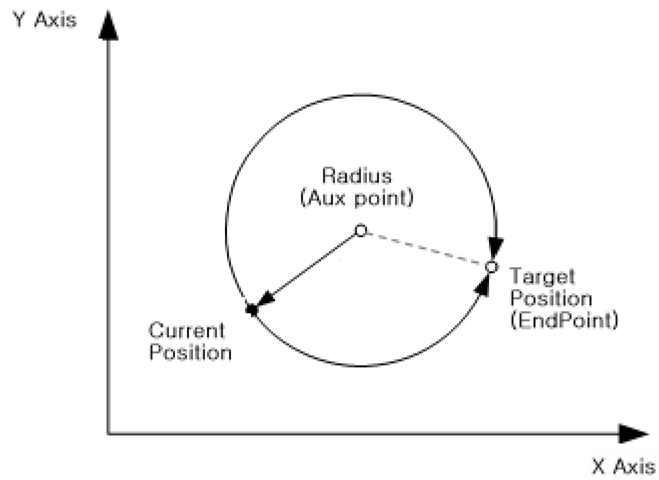


(b) Circular Interpolation Using Center Point Specification (CircMode = 1)
This method performs circular interpolation to the target position by starting operation at the current position, and following a circular trajectory of which diameter corresponds to the distance to the designated center point. In the figure below, the current position corresponds to the axes group coordinate at the start of the command, the center point corresponds to the coordinate input for the AuxPoint, and the target position corresponds to the absolute coordinate input for the EndPoint.



(c) Circular Interpolation using Radius Speciation (CircMode = 2)

This method performs circular interpolation to the target position by starting operation at the current position, and following a circular trajectory with a designated radius from the current position to the target position. In the figure below, the current position corresponds to the axes group coordinate at the start of the command, the radius corresponds to the X coordinate input for the AuxPoint, and the target position corresponds to the absolute coordinate input for the EndPoint.



(6) Refer to chapter 8.2.7 linear interpolation control part in motion controller's manual for more details.

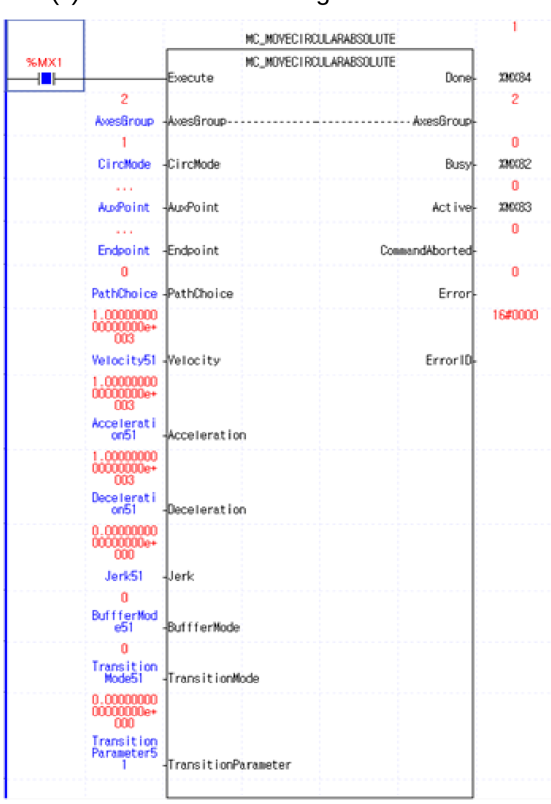
(7) If the function block is re-executed (Execute input is On) before the instruction is terminated, the changed parameters are applied.
Only, Velocity, Acceleration, Deceleration, Jerk, AuxPoint, Endpoint input can be updated.

(8) Velocity input can be set to 0 or changed.

(9) Example program

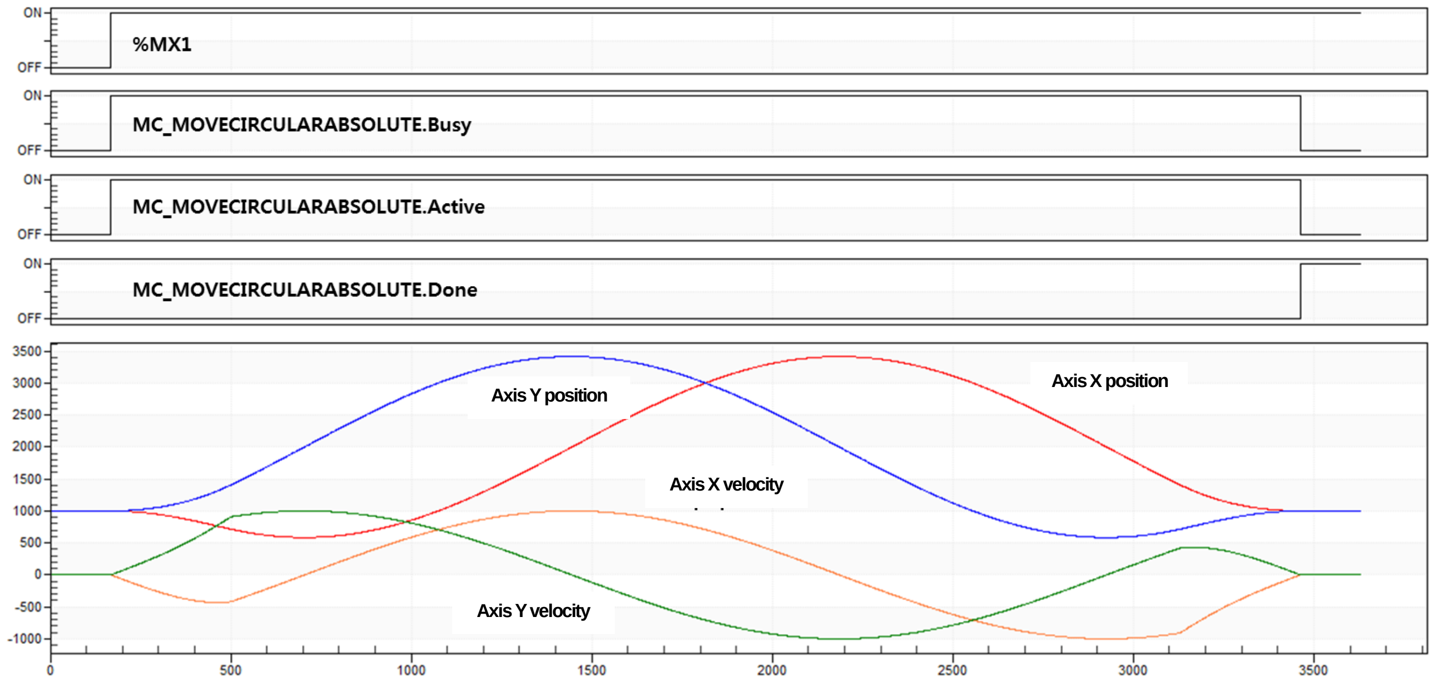
This example shows the circular interpolation to the target position (1000, 1000) by moving clock-wise after setting the center point (2000,2000) specification method when the current command position is (1000, 1000).

(a) Function block setting

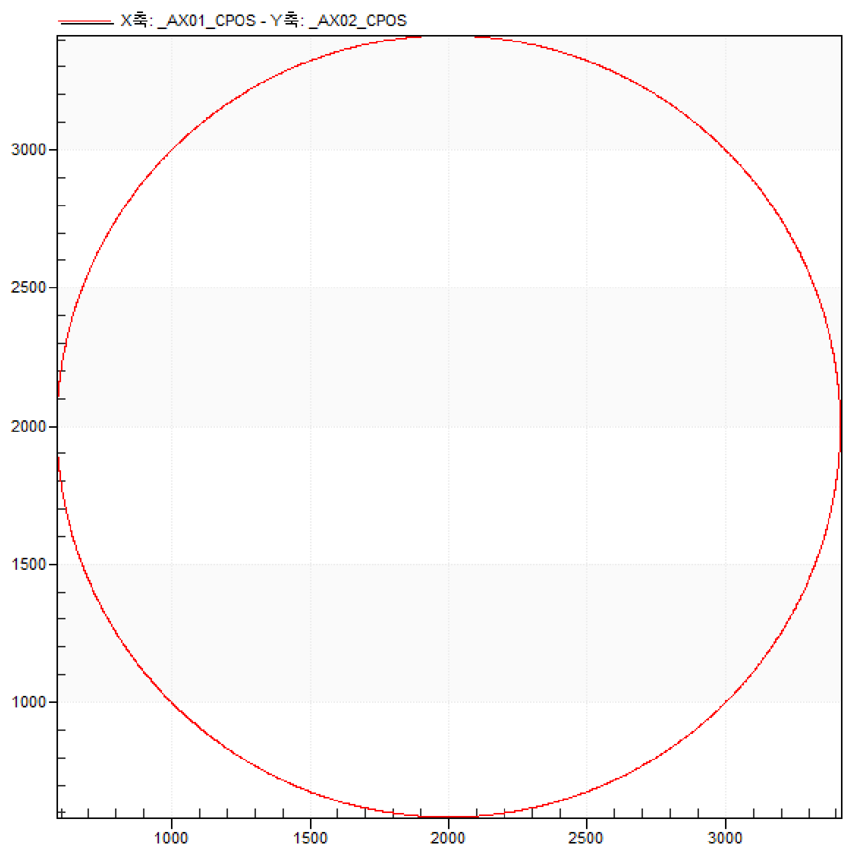


1	<GLOBAL>	%JL1.3	1.0000000000000000e+003	LREAL	_AX01_CPOS
2	<GLOBAL>	%JL2.3	1.0000000000000000e+003	LREAL	_AX02_CPOS
3	<GLOBAL>	%JL1.4	0.0000000000000000e+000	LREAL	_AX01_CVEL
4	<GLOBAL>	%JL2.4	0.0000000000000000e+000	LREAL	_AX02_CVEL
5	Group	⊟ AuxPoint		ARRAY[0..2] OF LREAL	
6	Group	AuxPoint[0]	2.0000000000000000e+003	LREAL	
7	Group	AuxPoint[1]	2.0000000000000000e+003	LREAL	
8	Group	AuxPoint[2]	0.0000000000000000e+000	LREAL	
9	Group	⊟ Endpoint		ARRAY[0..2] OF LREAL	
10	Group	Endpoint[0]	1.0000000000000000e+003	LREAL	
11	Group	Endpoint[1]	1.0000000000000000e+003	LREAL	
12	Group	Endpoint[2]	0.0000000000000000e+000	LREAL	

(b) Timing diagram

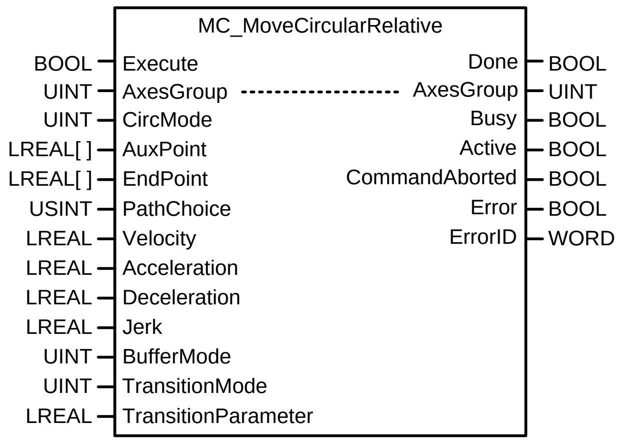


(c) XY graph



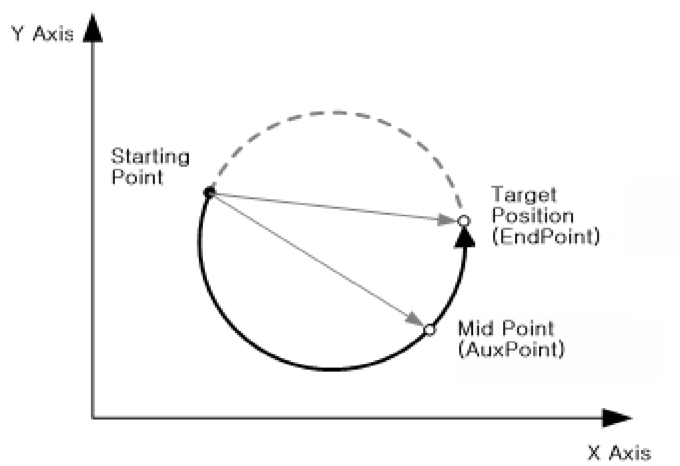
6.5.14 Incremental position circular Interpolation operation (MC_MoveCircularRelative)

Motion Function Block		

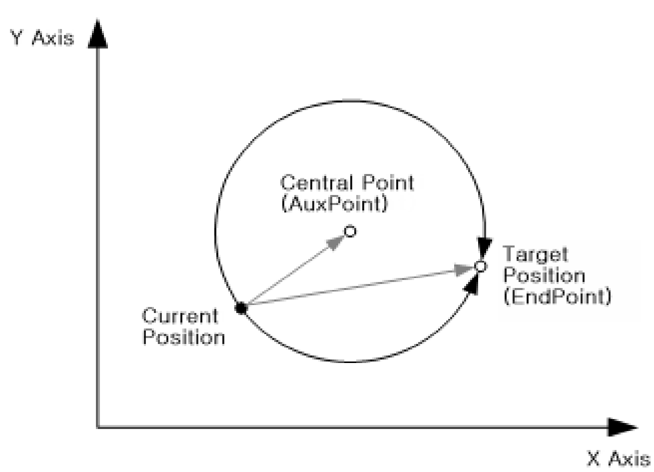


Input-Output		
UINT	AxesGroup	Set the group to do relative position circular interpolation operation. (1 ~ 16: 1Group ~ 16 Group)
Input		
BOOL	Execute	Give relative position circular interpolation operation command to the relevant group in the rising Edge.
UINT	CirMode	Circular interpolation method setting [0: Midpoint, 1: Central point, 2: Radius]
LREAL[ ]	AuxPoint	Specify the position of auxiliary point depending on the circular interpolation method as the relative coordinate based on the starting point.
LREAL[ ]	EndPoint	Specify the end point of circular arc as the relative coordinate based on the starting point.
BOOL	PathChoice	Circular route selection 0: clockwise direction, 1: counter-clockwise direction
LREAL	Velocity	Specify the maximum speed of the route. [u/s]
LREAL	Acceleration	Specify the maximum acceleration. [u/s^2]
LREAL	Deceleration	Specify the maximum deceleration. [u/s^2]
LREAL	Jerk	Specify the change rate of acceleration/deceleration. [u/s^3]
UINT	BufferMode	Specify the sequential operation setting of motion function block. (Refer to 6.1.4.BufferMode input)
UINT	TransitionMode	Unused
LREAL	TransitionParameter	Unused
Output		
BOOL	Done	Indicate whether to reach the specified position.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that whether or not motion function block is controlling the group.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted while it is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block issues relative position circular interpolation command on the axes group designated by AxesGroup input.
(2) When this motion function block starts, each axis performs circular trajectory interpolation control referring to the auxiliary point input, and the movement direction is determined by Path Choice input. If PathChoice input is set to 0, circular interpolation is operated in a clockwise direction, and if it is set to 1, circular interpolation is operated in a counter-clockwise direction.
(3) At AuxPoint and EndPoint input, designate the arrangement of the relative position of auxiliary points to refer to for circular interpolation of each axis. The input arrangement and the axes of the group correspond to the designated axis IDs [ID1, ID2, ID3, ···], in that order. (The 3 LEAL type sized array should be entered in Position input as there are 3 axes which comprise the group to give a circular interpolation operation command.)
(4) In Velocity, Acceleration, Deceleration, Jerk inputs, the acceleration, deceleration, change rate of acceleration, velocity of the interpolation path are specified, respectively.
(5) Set the circular interpolation method in CircMode input. The circular interpolation methods corresponding to CircMode values are as follows.
   (a) Circular interpolation of midpoint specifying method (BORDER, CircMode = 0) In this method, operation starts at the current position and it does circular interpolation through the pecified position of the central point to the target position. In the figure below, the current position corresponds to the axes group coordinate at the start of the command, the midpoint corresponds to the coordinate input for the AuxPoint, and the target position corresponds to the relative coordinate input for the EndPoint.

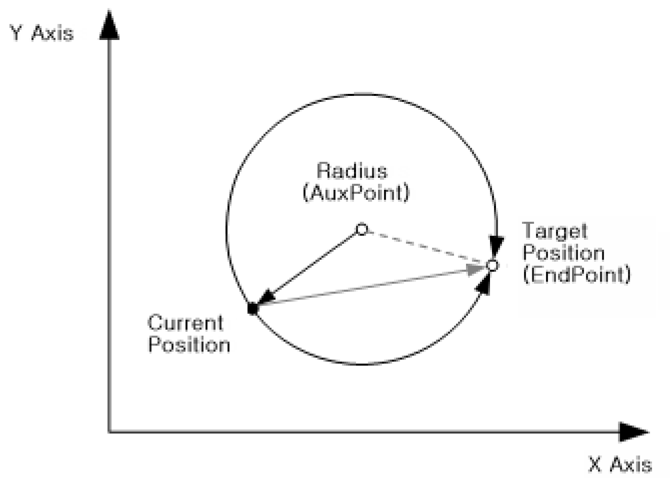


   (b) Circular interpolation of central point specifying method In this method, operation starts at the current position, and it does circular interpolation to the target position along the circular path, which has a radius of the distance to the specified central position. In the figure below, the current position corresponds to the axes group coordinate at the start of the command, the center point corresponds to the coordinate input for the AuxPoint, and the target position corresponds to the relative coordinate input for the EndPoint.



(c) Circular interpolation with radius designation form

In this method, operation starts at the current position, and it does circular interpolation to the target position along the circular path which has a radius of the value specified in the radius. In the figure below, the current position corresponds to the axes group coordinate at the start of the command, the diameter corresponds to the X coordinate input for the AuxPoint, and the target position corresponds to the relative coordinate input for the EndPoint.



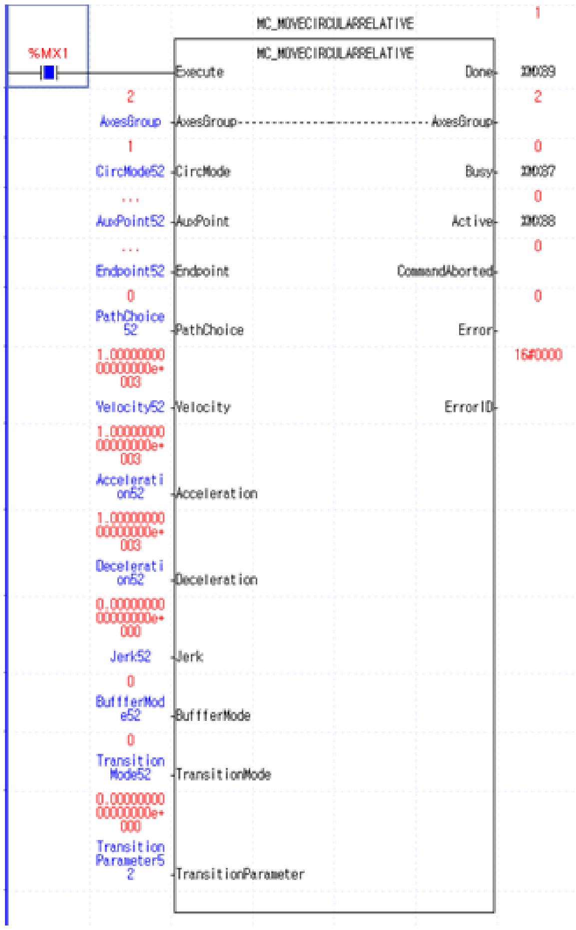
(6) Refer to chapter 8.2.7 linear interpolation control part in motion controller's manual for more details.

(7) If the function block is re-executed (Execute input is On) before the instruction is terminated, the changed parameters are applied.
Only, Velocity, Acceleration, Deceleration, Jerk, AuxPoint, Endpoint input can be updated.

(8) Velocity input can be set to 0 or changed.

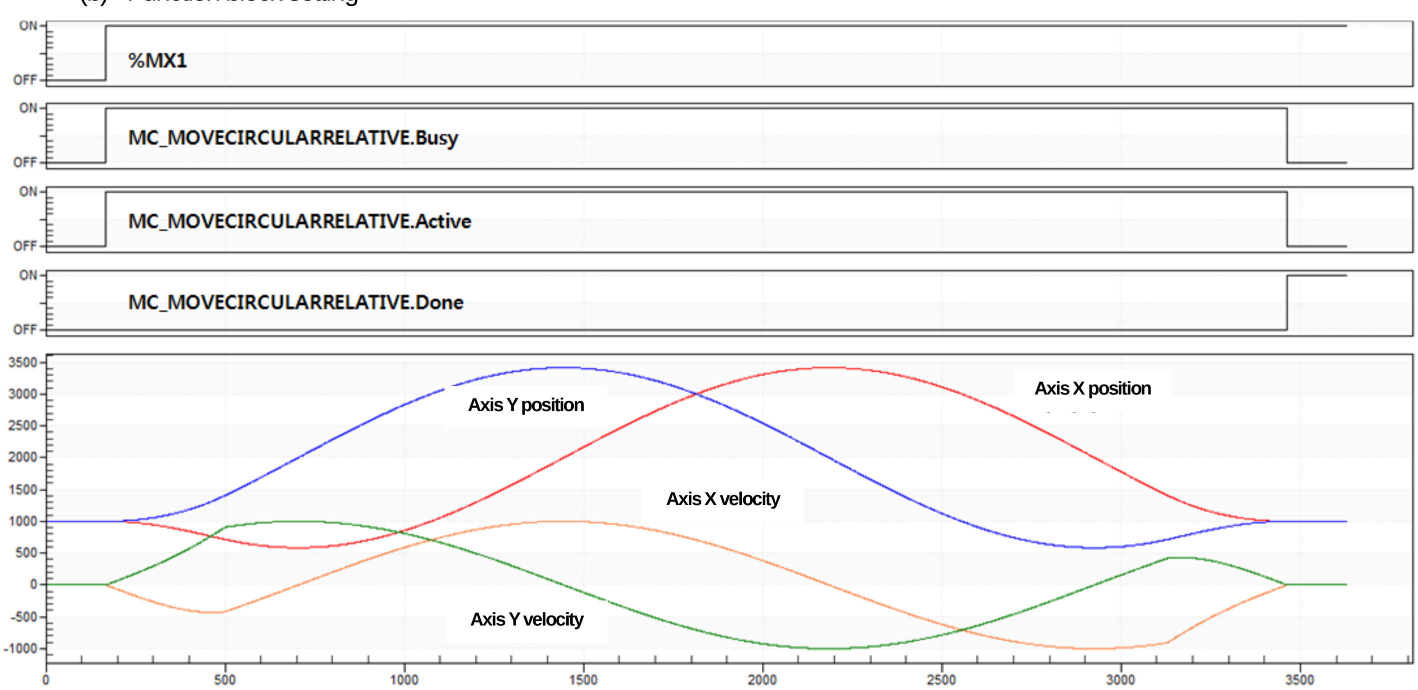
(9) Example program
This example is to set the center point specification method when the current command position is (1000, 1000) (set the relative position from the current position to the center point to set: 1000, 1000), and move clock-wise to perform circular interpolation to the target position (set the relative position from the current position to the target position: 0, 0).

(a) Function block setting

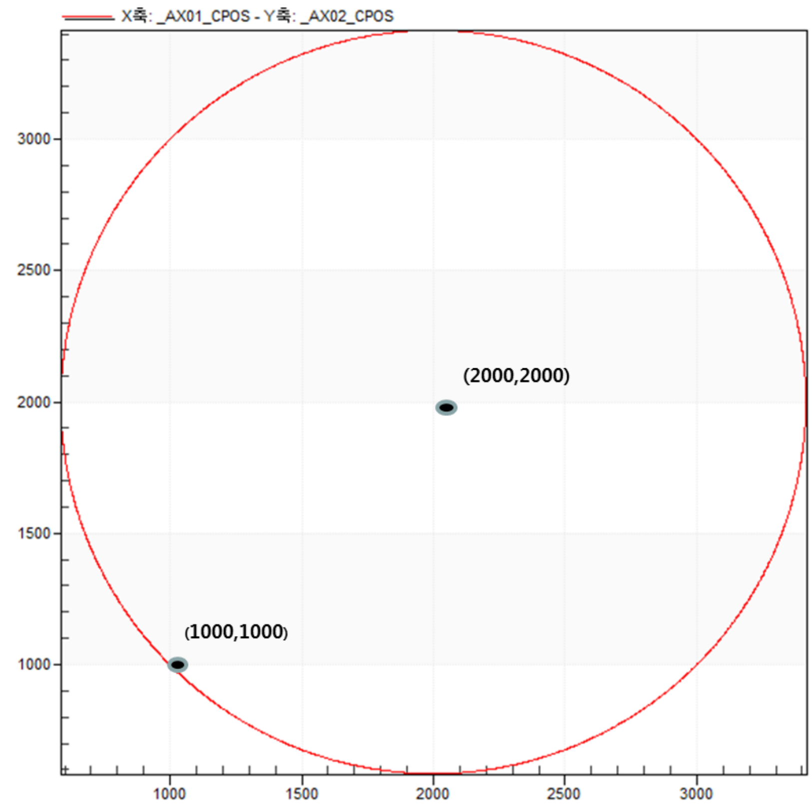


1	<GLOBAL>	%JL1.3	1.0000000000000000e+003	LREAL	_AX01_CPOS
2	<GLOBAL>	%JL2.3	1.0000000000000000e+003	LREAL	_AX02_CPOS
3	<GLOBAL>	%JL1.4	0.0000000000000000e+000	LREAL	_AX01_CVEL
4	<GLOBAL>	%JL2.4	0.0000000000000000e+000	LREAL	_AX02_CVEL
5	Group	AuxPoint52		ARRAY[0..2] OF LREAL	
6	Group	AuxPoint52[0]	1.0000000000000000e+003	LREAL	
7	Group	AuxPoint52[1]	1.0000000000000000e+003	LREAL	
8	Group	AuxPoint52[2]	0.0000000000000000e+000	LREAL	
9	Group	Endpoint52		ARRAY[0..2] OF LREAL	
10	Group	Endpoint52[0]	0.0000000000000000e+000	LREAL	
11	Group	Endpoint52[1]	0.0000000000000000e+000	LREAL	
12	Group	Endpoint52[2]	0.0000000000000000e+000	LREAL	

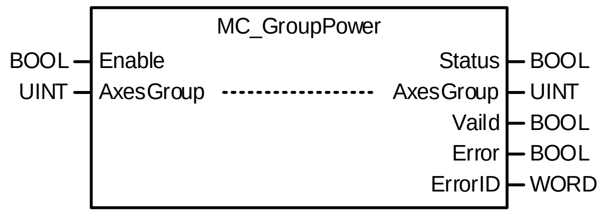
(b) Function block setting



(c) XY graph


X축: _AX01_CPOS - Y축: _AX02_CPOS
3000
2500
2000
1500
1000
(2000,2000)
(1000,1000)
1000
1500
2000
2500
3000

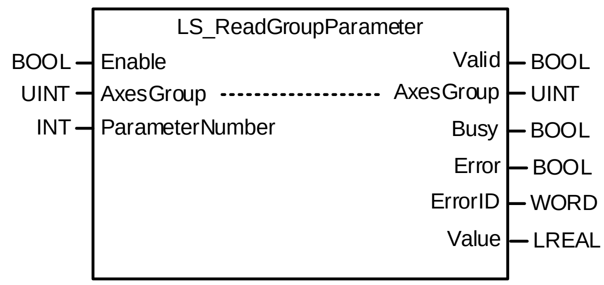
6.5.15 Axis Group Servo On/Off (MC_GroupPower)

Motion Function Block		
		
Input-Output		
UINT	AxesGroup	Set the group to do relative position circular interpolation operation. (1 ~ 16: 1Group ~ 16 Group)
Input		
BOOL	Enable	Group servo motor of the relevant axis is servo On while input is activated.
Output		
BOOL	Status	Indicate the power permission status of the Axis Group.
BOOL	Valid	Indicate the validity of motion function block output. (same with Status output here)
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block is to give servo On/Off command to relevant axis group.
(2) When Enable input changes from Off to On, the Servo On command is given to the relevant axis group. When it changes from On to Off, the Servo Off command is given to it.
(3) If servo On command is executed when the axis is in 'Disable' state, the axis state is 'StandStill', and failure in servo On brings 'ErrorStop' state.

6.5.16 Read axis group parameter (LS_ReadGroupParameter)

Motion Function Block



Input-Output		
UINT	AxesGroup	Set the axis group to read parameters from. (1~16:1Group ~ 16Group)
Input		
BOOL	Enable	The relevant parameters are output while the input is enabled.
INT	ParameterNumber	Specify the number of parameter to read.
Output		
BOOL	Valid	Indicate whether the output of the current motion function block is valid.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.
LREAL	Value	Output the values of the parameters.

(1) This motion function block is to output parameter of the relevant axis group.
(2) The value of the relevant parameter is continuously output in Value while Enable input is On.
(3) ParameterGroup input specifies the parameter group number to read.
If the value is set outside the range, “error 0x2120” occurs.

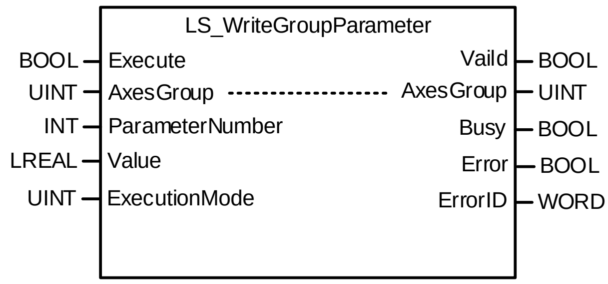
(4) The numbers of axis group parameter are as below.

No.	Parameter	Item	Note	OS*Note 2)
1	Standard setting	Configuration axis 01	0 : none, 1~36	-
2		Configuration axis 02	0 : none, 1~36	-
3		Configuration axis 03	0 : none, 1~36	-
4		Configuration axis 04	0 : none, 1~36	-
5		Configuration axis 05	0 : none, 1~36	-
6		Configuration axis 06	0 : none, 1~36	-
7		Configuration axis 07	0 : none, 1~36	-
8		Configuration axis 08	0 : none, 1~36	-
9		Configuration axis 09	0 : none, 1~36	-
10		Configuration axis 10	0 : none, 1~36	-
11		Interpolation speed max	LREAL Positive number(unit/s)	
42		Coordinate system look ahead settings	1 ~ 10	-
43		Coordinate output filter time constant	0 ~ 100(ms)	-
45		Interpolation operation blending angle limit	0: Disable, 1: Able	-
46		Interpolation operation blending allowable angle	2 ~ 178 (deg)	-
12	Coordinate setting	Coordinate system	0:none, 1:xyz, 2:Delta3, 3:Delta3R, 4:LinearDelta3, 5:LinearDelta3R, 6:T-Gantry, 7:T-GantryR	-
13		Coordinates parameter 1	0 or LREAL Positive number [mm]	-
14		Coordinates parameter 2	0 or LREAL Positive number [mm]	-
15		Coordinates parameter 3	0 or LREAL Positive number [mm]	-
16		Coordinates parameter 4	0 or LREAL Positive number [mm]	-
17		Coordinates parameter 5	0 or LREAL Positive number [mm]	-
18		Coordinates parameter 6	0 or LREAL Positive number [mm]	-
19	Tool setting	X-axis offset	Long real(LREAL)	
20		Y-axis offset	Long real(LREAL)	
21		Z-axis offset	Long real(LREAL)	
22	Work space setting	Work space type	0:Ractangle, 1:Cylinder, 2:Delta, 3:Sector	
23		Workspace error check	0:disabled,1:enabled	
24		Work space parameter1	Long real(LREAL)	
25		Work space parameter2	Long real(LREAL)	
26		Work space parameter3	Long real(LREAL)	
27		Work space parameter4	Long real(LREAL)	
28		Work space parameter5	Long real(LREAL)	
29		Work space parameter6	Long real(LREAL)	
30		Work space parameter7	Long real(LREAL)	
31		Work space parameter8	Long real(LREAL)	
32	PCS Setting	X-axis move	Long real(LREAL)	
33		Y-axis move	Long real(LREAL)	
34		Z-axis move	Long real(LREAL)	
35		X-axis rotation	-360 ~ 360 (deg)	
36		Y-axis rotation	-360 ~ 360 (deg)	

No.	Parameter	Item	Note	OS*Note 2)
37		Z-axis rotation	-360 ~ 360 (deg)	
38	Coordinate system jog operation settings	XYZ low speed	LREAL number (mm/sec), It should be less than or equal to XYZ high speed.	
39		ABC low speed	LREAL number (deg/sec), It should be less than or equal to ABC high speed.	
40		XYZ high speed	LREAL number (mm/sec)	
41		ABC high speed	Long real (LREAL) (deg/sec)	

6.5.17 Write axis group parameter (LS_WriteGroupParameter)

Motion Function Block



Input-Output		
UINT	AxesGroup	Set the axis group to give commands from. (1~16:1Group ~ 16Group)
Input		
BOOL	Execute	At the rising edge, the corresponding parameter is reflected in the axis group.
INT	ParameterNumber	Specify the number of parameter to write.
LREAL	Value	Specify the value of parameter to write.
UINT	ExecutionMode	Specify the time when parameter is written.
Output		
BOOL	Valid	Indicate whether parameter is successfully written.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block is to write the value specified in parameter of the relevant axis group.
(2) The parameters will be written in the rising edge of the Execute input.
(3) ParameterNumber input specifies the number of the parameter to be written. The value unable to be set causes "error 0x2120".
(4) Specify the value to write in parameter for Value input.
(5) In ExecutionMode, correct the time when parameter is written and the values below can be set. The value unable to be set causes "error 0x2120".
0 (mcImmediately): Change the parameter value immediately after executing function block (rising Edge in Execute input). If the relevant axis is in running, operation can be affected.
1 (mcQueued): Changed at the same point with 'Buffered' in Buffermode. (6.1. 4. Refet to BufferMode)

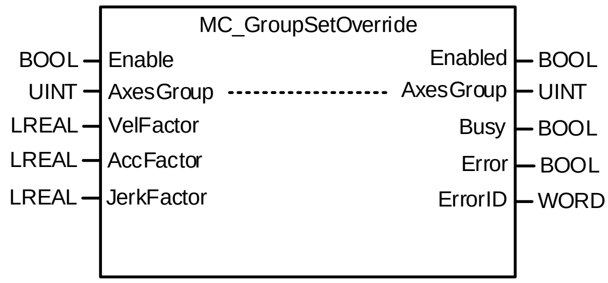
(6) The numbers of axis group parameter are as below.

No.	Parameter	Item	Note	OS*Note 2)
1	Standard setting	Configuration axis 01	0 : none, 1~36	-
2		Configuration axis 02	0 : none, 1~36	-
3		Configuration axis 03	0 : none, 1~36	-
4		Configuration axis 04	0 : none, 1~36	-
5		Configuration axis 05	0 : none, 1~36	-
6		Configuration axis 06	0 : none, 1~36	-
7		Configuration axis 07	0 : none, 1~36	-
8		Configuration axis 08	0 : none, 1~36	-
9		Configuration axis 09	0 : none, 1~36	-
10		Configuration axis 10	0 : none, 1~36	-
11		Interpolation speed max	LREAL Positive number(unit/s)	
42		Coordinate system look ahead settings	1 ~ 10	-
43		Coordinate output filter time constant	0 ~ 100(ms)	-
45		Interpolation operation blending angle limit	0: Disable, 1: Able	-
46		Interpolation operation blending allowable angle	2 ~ 178 (deg)	-
12	Coordinate setting	Coordinate system	0:none, 1:xyz, 2:Delta3, 3:Delta3R, 4:LinearDelta3, 5:LinearDelta3R, 6:T-Gantry, 7:T-GantryR	-
13		Coordinates parameter 1	0 or LREAL Positive number [mm]	-
14		Coordinates parameter 2	0 or LREAL Positive number [mm]	-
15		Coordinates parameter 3	0 or LREAL Positive number [mm]	-
16		Coordinates parameter 4	0 or LREAL Positive number [mm]	-
17		Coordinates parameter 5	0 or LREAL Positive number [mm]	-
18		Coordinates parameter 6	0 or LREAL Positive number [mm]	-
19	Tool setting	X-axis offset	Long real(LREAL)	
20		Y-axis offset	Long real(LREAL)	
21		Z-axis offset	Long real(LREAL)	
22	Work space setting	Work space type	0:Ractangle, 1:Cylinder, 2:Delta, 3:Sector	
23		Workspace error check	0:disabled,1:enabled	
24		Work space parameter2	Long real(LREAL)	
25		Work space parameter2	Long real(LREAL)	
26		Work space parameter3	Long real(LREAL)	
27		Work space parameter4	Long real(LREAL)	
28		Work space parameter5	Long real(LREAL)	
29		Work space parameter6	Long real(LREAL)	
30		Work space parameter7	Long real(LREAL)	
31		Work space parameter8	Long real(LREAL)	
32	PCS Setting	X-axis move	Long real(LREAL)	
33		Y-axis move	Long real(LREAL)	
34		Z-axis move	Long real(LREAL)	
35		X-axis rotation	-360 ~ 360 (deg)	
36		Y-axis rotation	-360 ~ 360 (deg)	

No.	Parameter	Item	Note	OS*Note 2)
37		Z-axis rotation	-360 ~ 360 (deg)	
38	Coordinate system jog operation settings	XYZ low speed	LREAL number (mm/sec), It should be less than or equal to XYZ high speed.	
39		ABC low speed	LREAL number (deg/sec), It should be less than or equal to ABC high speed.	
40		XYZ high speed	LREAL number (mm/sec)	
41		ABC high speed	Long real (LREAL) (deg/sec)	

6.5.18 Axis group override (MC_GroupSetOverride)

Motion Function Block



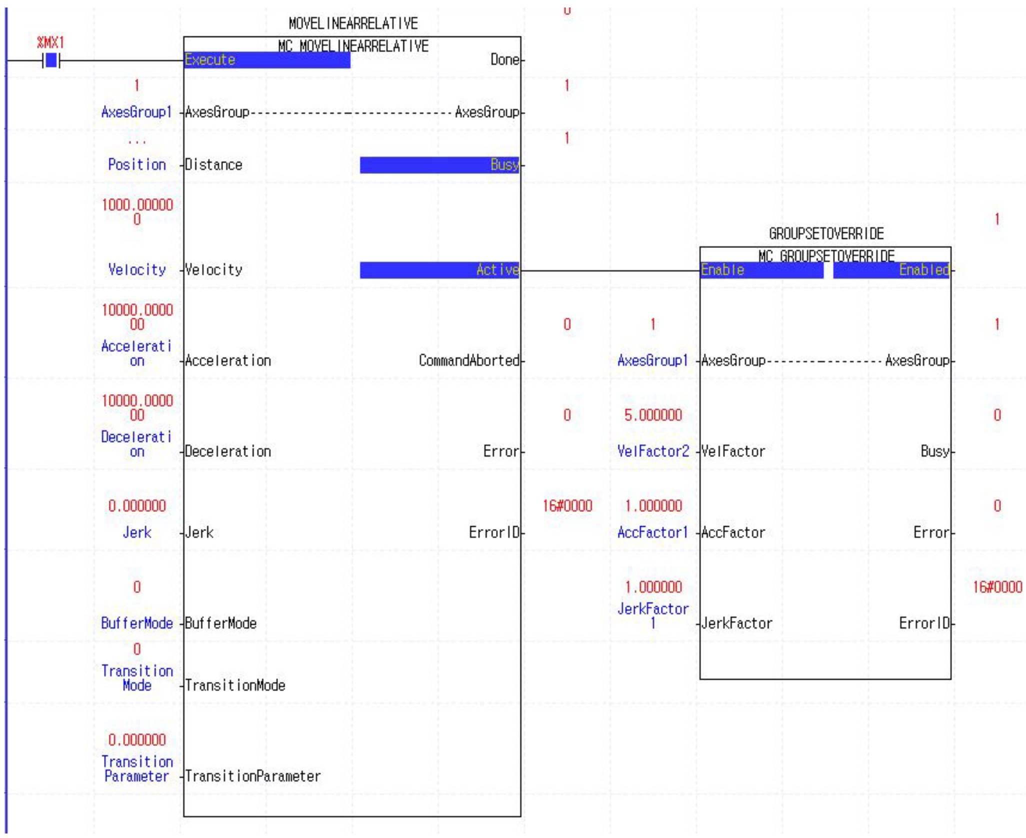
Input-Output		
UINT	AxesGroup	Set the axis group to give commands from. (1~16:1Group ~ 16Group)
Input		
BOOL	Enable	Execute override operation in the relevant axis group while input is activated.
LREAL	VelFactor	Specify the override rate of the speed.
LREAL	AccFactor	Specify the override rate of acceleration/deceleration.
LREAL	JerkFactor	Specify the override ratio of the rate of change for acceleration.
Output		
BOOL	Enabled	Indicate that the override rate was applied successfully.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block is to override the speed of the relevant axis group, acceleration, and the change rate of acceleration.
(2) Override rate which is applied to the relevant axis can be specified and changed while Enable input is On. If Enable input is Off, override rate right before the Off is maintained.
(3) Specify the speed override ratio for the VelFactor input. If the specified value is 0.0, the relevant axis group stops but it is not changed to 'StandStill' state.
(4) Specify acceleration/deceleration and override rate of jerk (change rate of acceleration) in AccFactor and JerkFactor input respectively.
(5) Negative number cannot be input in each Facotr, and if it is input, "error 0x2111" occurs.
(6) Default of each override rate is 1.0, and it means 100% of the command speed of function block currently running.
(7) Override operation does not affect the serve axis of the relevant axis group.
(8) Override command cannot be used during synchronous operation.

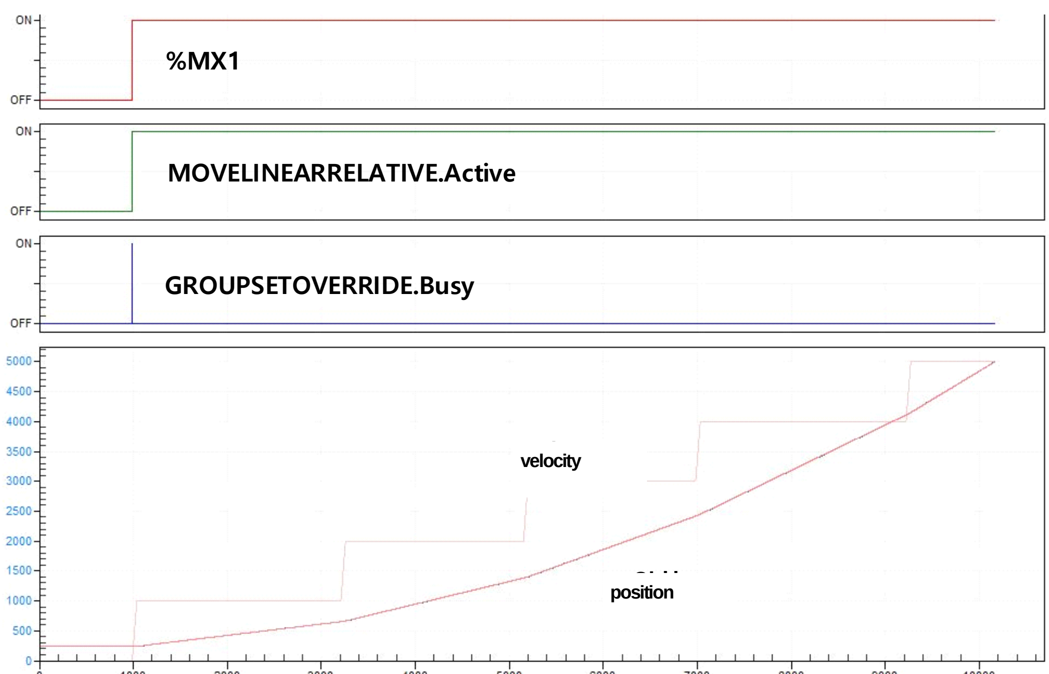
(9) Example program

This example shows the operation by changing the current velocity to 2,000/ 3,000/ 4,000/ 5,000 if VelFactor is changed to 2/3/4/5 at the current velocity of 1,000.

(a) Function block setting

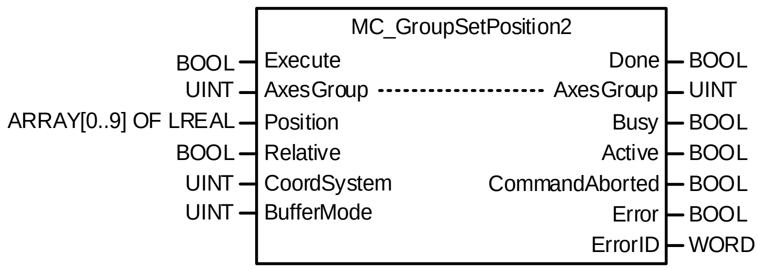


(b) Timing diagram



6.5.19 Group current position setting (MC_GroupSetPosition2)

Motion Function Block



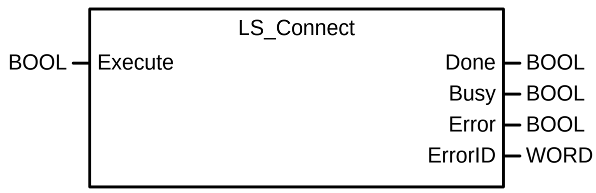
Input-Output		
UINT	AxesGroup	Select the group to set the current position. (1 ~ 16: 1Group ~ 16 Group)
Input		
BOOL	Execute	Give group current position setting command to the relevant group in the rising Edge.
LREAL[ ]	Position	Specify the position.
BOOL	Relative	0: Position value=Absolute position, 1:Position value=Incremental position
UINT	CoordSystem	Select the coordinate system type.(0:ACS 1:MCS 2:PCS)
UINT	BufferMode	Specify the sequential operation setting of motion function block. (6.1. 4. Refer to BufferMode input)
Output		
BOOL	Done	Indicate the completion state of motion function block.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that the current motion function block is controlling the relevant axis.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted while it is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) The motion function block does not move the axis and sets the position on all axes of the relevant axis group.
(2) The new coordinates are specified as an array in the Position input.
(3) Set the corresponding coordinate system in CoordSystem input. The MC_GroupSetPosition function block moves the position of the specified coordinates and affects the upper coordinate system. (Selecting ACS affects MCS and PCS)
(4) If the Relative input is Off, the Position input value is applied as a new position in the specified coordinate system. If the Relative input is On, the Position input value is added to the current position of the designated coordinate system.

6.6 Dedicated Function Block

6.6.1 Communication connection (LS_Connect)

Motion Function Block



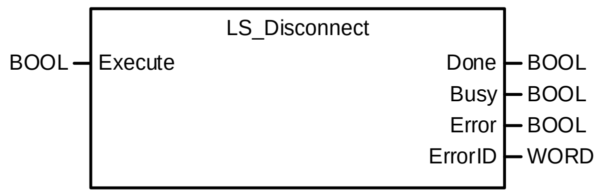
Input		
BOOL	Execute	Give communication connection command to the relevant motion controller in the rising Edge.
Output		
BOOL	Done	Indicate whether to complete communication connection.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block is to give a command to connect communication with servo drive or external input/output apparatus to the motion controller.

(2) When slave devicees are normally connected, Done is On and Busy is Off.

(3) If an error occurs during the communication connection, Error is On and error number is output in ErrorID according to the cause.

6.6.2 Communication Disconnect (LS_Disconnect)

Motion Function Block		
		
Input		
BOOL	Execute	Give communication disconnection command to the relevant motion controller in the rising Edge.
Output		
BOOL	Done	Indicate whether to complete communication disconnection.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block gives a command which orders the motion controller to disconnect the communication with servo drive or external input/output apparatuses.

(2) If communication slave is disconnected, Done is On and Busy is off. If an error occurs during the execution of communication disconnection, Error is On and error number is output in ErrorID according to the error situation.

6.6.3 Read SDO (LS_ReadSDO)

Motion Function Block

```
                 LS_ReadSDO
BOOL ─ Execute                  Done ─ BOOL
UINT ─ Slave-------------------Slave ─ UINT
UINT ─ Index                    Busy ─ BOOL
UINT ─ SubIndex                Error ─ BOOL
UINT ─ Length                ErrorID ─ WORD
                               Value ─ DINT
```

Input-Output		
UINT	Slave	Set the slave to be given a command. (refer toSee 6.2.1 Setting Range by Product)
Input		
BOOL	Execute	Give SDO reading command to the relevant slave in the rising Edge.
UINT	Index	Set the Index of slaver Object to be read. (0x0000~ 0x9FFF)
UINT	SubIndex	Set the SubIndex of slave Object to be read. (0 ~ 255)
UINT	Length	Set the distance of slave Object to be read by Byte. (1 ~ 4)
Output		
BOOL	Done	Indicate that SDO is successfully read.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.
DINT	Value	Output the value of SDO.

(1) This motion function block is to read the SDO (CoE Object) value of servo drive in the relevant axis, and reads the SDO value of the position specified in Index and SubIndex of the axis specified by Axis input as much as the size of Length and indicates it on Value output.

(2) Value output is eliminated to 0 when motion function block is running, and it is output as the read value when the running is completed (Done output is On).

(3) Index input can be set as below. If the value is set outside the range, "error 0x1F12" occurs.

Setting Value	Content
16#0000 ~ 16#0FFF	Data Type Description
16#1000 ~ 16#1FFF	Communication objects
16#2000 ~ 16#5FFF	Manufacturer Specific Profile Area
16#6000 ~ 16#9FFF	Standardized Device Profile Area

(4) The value between 0~255 can be entered in SubIndex, and if the value is set outside the range, "error 0x1F12" occurs.

(5) In Length, values ranging from 1 to 4 can be entered, which mean 1 to 4 bytes. If the value is set outside the range, "error 0x1F12" occurs.

6.6.4 Write SDO (LS_WriteSDO)

Motion Function Block		

```
                 LS_WriteSDO
BOOL  — Execute                      Done  — BOOL
UINT  — Slave ---------------------- Slave — UINT
UINT  — Index                        Busy  — BOOL
UINT  — SubIndex                     Error — BOOL
UINT  — Length                     ErrorID — WORD
DINT  — Value
```

Input-Output		
UINT	Slave	Set the slave to be given a command. (refer toSee 6.2.1 Setting Range by Product)
Input		
BOOL	Execute	Give SDO writing command to the relevant slave in the rising Edge.
UINT	Index	Set the Index of slave Object to be written. (0x0000~ 0x9FFF)
UINT	SubIndex	Set the SubIndex of slave Object to be written. (0 ~ 255)
UINT	Length	Set the distance of slave Object to be written by Byte. (1 ~ 4)
DINT	Value	Set the value to be written in SDO.
Output		
BOOL	Done	Indicate that SDO is successfully read.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block is to write the SDO (CoE Object) value of the relevant slave, and it writes the value entered in Value as the size of the Length in SDO of the position specified as Index and SubIndex of the slave specified in slave input.

(2) Index input can be set as below. If the value is set outside the range, "error 0x1F12" occurs.

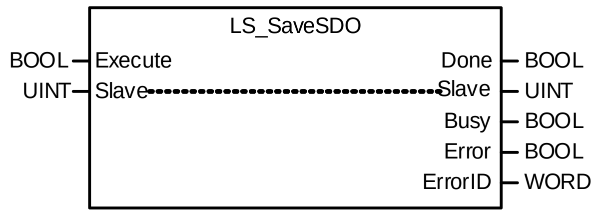
Setting Value	Content
16#0000 ~ 16#0FFF	Data Type Description
16#1000 ~ 16#1FFF	Communication objects
16#2000 ~ 16#5FFF	Manufacturer Specific Profile Area
16#6000 ~ 16#9FFF	Standardized Device Profile Area

(3) The value between 0~255 can be entered in SubIndex, and if the value is set outside the range, "error 0x1F12" occurs.

(4) In Length, values ranging from 1 to 4 can be entered, which mean 1 to 4 bytes. If the value is set outside the range, "error 0x1F12" occurs.

6.6.5 Save SDO (LS_SaveSDO)

Function Block type		

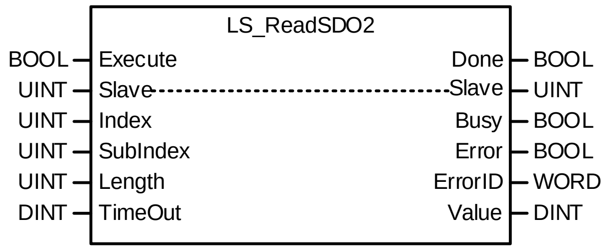


Input-Output		
UINT	Slave	Specify the Slave to be commanded (See 6.2.1 Setting Range by Product)
Input		
BOOL	Execute	Give SDO saving command to the relevant slave in the rising Edge.
Output		
BOOL	Done	Indicate that SDO is successfully save.
BOOL	Busy	Indicate that execution of the function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the error number that occurred while the function block is running.

(1) This motion function block is a command to save SDO of the designated slave to the memory of the slave.

6.6.6 Read SDO (LS_ReadSDO2)

Motion Function Block



Input-Output		
UINT	Slave	Set the slave to be given a command. (refer toSee 2.2.1 Setting Range by Product)
Input		
BOOL	Execute	Give SDO reading command to the relevant slave in the rising Edge.
UINT	Index	Set the Index of slaver Object to be read. (0x0000~ 0x9FFF)
UINT	SubIndex	Set the SubIndex of slave Object to be read. (0 ~ 255)
UINT	Length	Set the distance of slave Object to be read by Byte. (1 ~ 4)
DINT	TimeOut	Timeout of SDO read command (ms)
Output		
BOOL	Done	Indicate that SDO is successfully read.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.
DINT	Value	Output the value of SDO.

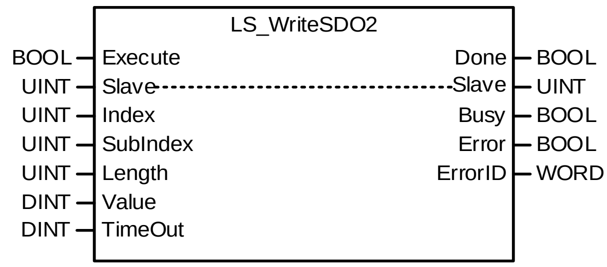
(1) This motion function block is a motion function block that reads the SDO (CoE Object) value of the corresponding slave, and reads the SDO value at the location specified as Index and SubIndex of the slave specified as Slave input as much as the Length size within the TimeOut setting time and displayed on the Value output. .
(2) Value output is eliminated to 0 when motion function block is running, and it is output as the read value when the running is completed (Done output is On).
(3) Index input can be set as below. If the value is set outside the range, "error 0x1F12" occurs.

Setting Value	Content
16#0000 ~ 16#0FFF	Data Type Description
16#1000 ~ 16#1FFF	Communication objects
16#2000 ~ 16#5FFF	Manufacturer Specific Profile Area
16#6000 ~ 16#9FFF	Standardized Device Profile Area

(4) The value between 0~255 can be entered in SubIndex, and if the value is set outside the range, "error 0x1F12" occurs.
(5) In Length, values ranging from 1 to 4 can be entered, which mean 1 to 4 bytes. If the value is set outside the range, "error 0x1F12" occurs.
(6) If SDO read is not performed within the time (ms) set in TimeOut, "Error 0x1F21" is generated.

6.6.7 Write SDO (LS_WriteSDO2)

Motion Function Block



Input-Output		
UINT	Slave	Set the slave to be given a command. (refer toSee 2.2.1 Setting Range by Product)
Input		
BOOL	Execute	Give SDO writing command to the relevant slave in the rising Edge.
UINT	Index	Set the Index of slave Object to be written. (0x0000~ 0x9FFF)
UINT	SubIndex	Set the SubIndex of slave Object to be written. (0 ~ 255)
UINT	Length	Set the distance of slave Object to be written by Byte. (1 ~ 4)
DINT	Value	Set the value to be written in SDO.
DINT	TimeOut	Timeout of SDO Write command (ms)
Output		
BOOL	Done	Indicate that SDO is successfully read.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

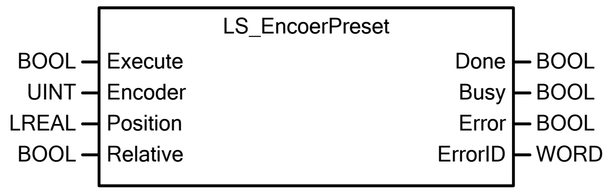
(1) This motion function block is to write the SDO (CoE Object) value of the relevant slave, and it writes the value entered in Value as the size of the Length in SDO value of the position specified as Index and SubIndex of the slave specified in slave input.
(2) Index input can be set as below. If the value is set outside the range, "error 0x1F12" occurs.

Setting Value	Content
16#0000 ~ 16#0FFF	Data Type Description
16#1000 ~ 16#1FFF	Communication objects
16#2000 ~ 16#5FFF	Manufacturer Specific Profile Area
16#6000 ~ 16#9FFF	Standardized Device Profile Area

(3) The value between 0~255 can be entered in SubIndex, and if the value is set outside the range, "error 0x1F12" occurs.
(4) In Length, values ranging from 1 to 4 can be entered, which mean 1 to 4 bytes. If the value is set outside the range, "error 0x1F12" occurs.
(5) If SDO read is not performed within the time (ms) set in TimeOut, "Error 0x1F14" is generated.

6.6.8 Encoder preset position setting (LS_EncoderPreset)

Function Block type

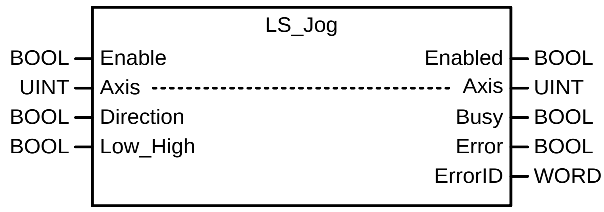


Input		
BOOL	Execute	Specify the position of the relevant encoder in the rising Edge.
UINT	Encoder	Set the encoder to set the position. (1: Encoder 1, 2:Encoder 2)
LREAL	Position	Specify the position to set. [u]
BOOL	Relative	0: Absolute coordinate position 1: Relative coordinate position
Output		
BOOL	Done	Indicate the completion state of motion function block.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block is to set the current position of the relevant encoder.
(2) Specify the position in Position input. When executing motion function command, if Relative input is Off, the position of the current axis is replaced with the Position input value, and if the Relative input is On, the Position input value is added to the current position of the relevant axis.

6.6.9 JOG operation (LS_Jog)

Function Block type		



Input-Output		
UINT	Axis	Specify the axis to be commanded. (refer toSee 6.2.1 Setting Range by Product)
Input		
BOOL	Enable	Give jog command to the relevant axis while input is On.
BOOL	Direction	Set the rotation direction in jog (0:forward direction , 1:reverse direction)
BOOL	Low_High	Set the JOG speed in JOG operation. (0: Jog low speed operation, 0: Jog high speed operation)
Output		
BOOL	Enabled	Indicate that the relevant axis is in jog.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block is to make the relevant axis perform jog operation.

(2) Jog is a manual operation function for test and is used to confirm the position address for system operation, wiring condition check, and teaching. Jog can be used by dividing the speed into high speed and low speed.

(3) When Enable input is On (in jog), if the value set in Low/High is changed, speed change occurs without stop in jog, and if the value set in JOG_DIR is changed, Jog is continued by changing the direction after the deceleration pause.

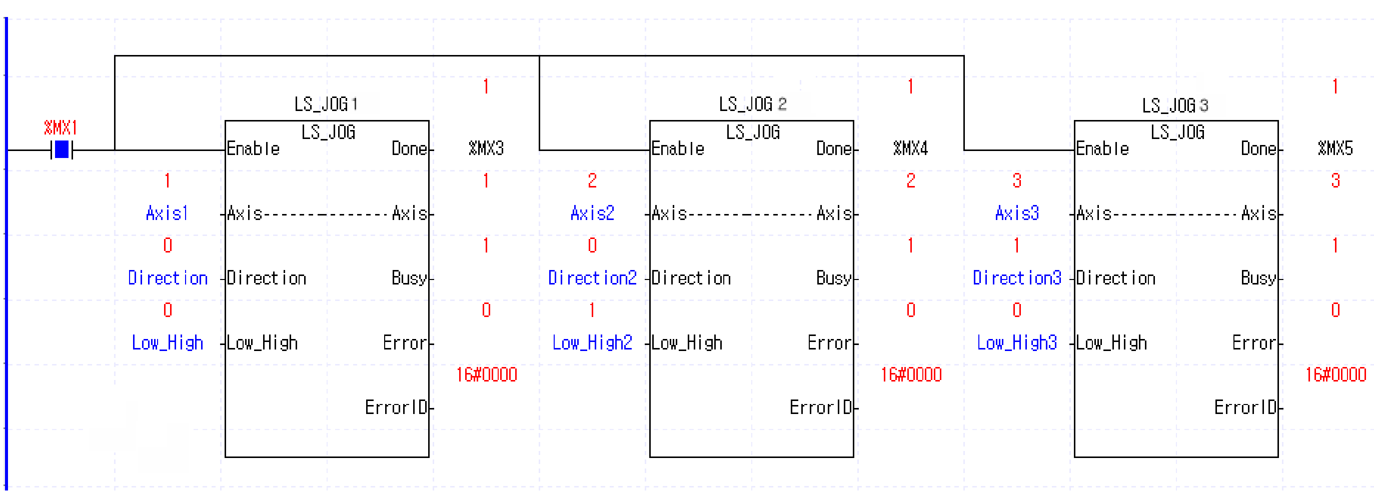
(4) If jog operation is stopped by another command, “Error 16#0015” occurs.

(5) Example program

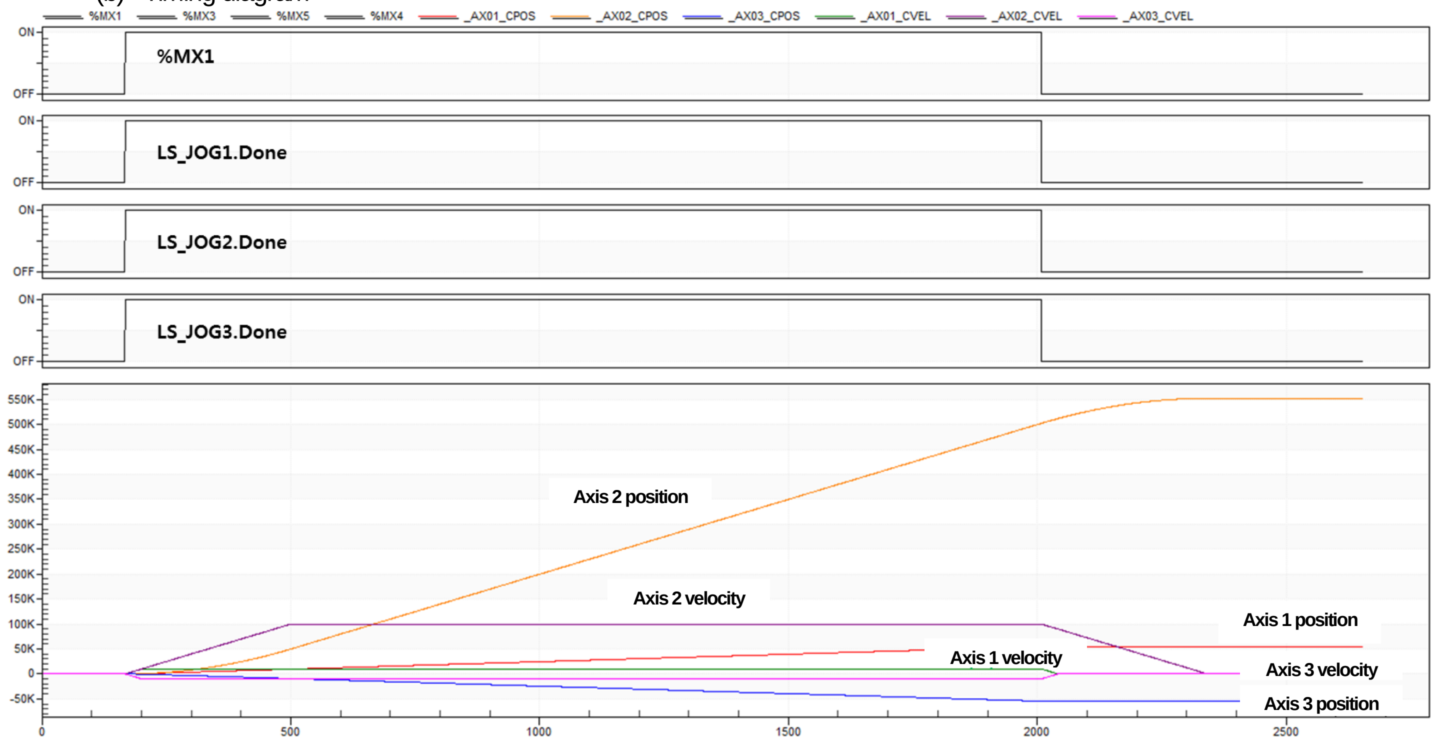
This example shows jog operation under the following settings when the current command position is 0.

Axis 1: Forward/low speed, Axis 2: Forward/high speed, Axis 3: Reverse/low speed,

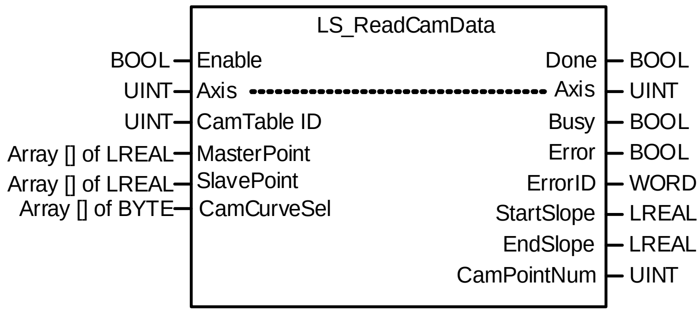
(a) Function block setting



(b) Timing diagram



6.6.10 Read Cam data (LS_ReadCamData)

Function Block type		
		
Input-Output		
UINT	Axis	Specify the Axis to be commanded (See 6.2.1 Setting Range by Product)
Input		
BOOL	Enable	Read the relevant cam data while input is On.
UINT	CamTableID	Specify the cam table to read. (refer toSee 6.2.1 Setting Range by Product)
LREAL[]	MasterPoint	MasterPoint value of the cam table is displayed on the areas of which front address is the set device.
LREAL[]	SlavePoint	SlavePoint value of the cam table is displayed on the areas of which front address is the set device.
BYTE[]	CamCurveSel	Cam curve type of the cam table is displayed on the areas of which front address is the set device. (0: Linear, 1: Cubic)
Output		
BOOL	Vaild	Indicate the validity of the function block output.
BOOL	Busy	Indicate that execution of the function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the error number that occurred while the function block is running.
LREAL	StartSlope	Output the StartSlope value of the relevant cam table.
LREAL	EndSlope	Output the EndtSlope value of the relevant cam table.
UINT	CamPointNum	Output the cam data point number of the relevant cam table.

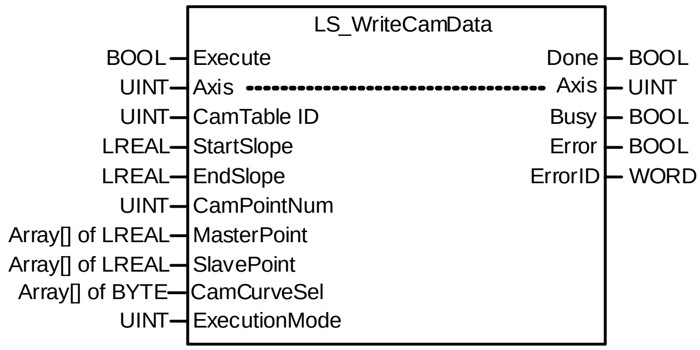
(1) This function block displays the data of the cam table.

(2) While Enable input is activated, the data values of the cam table are displayed in succession.

(3) The first address of the variables to store "Main-axis Position" and "Sub-axis Position" read from the cam profile is set at the MasterPoint and the SlavePoint. If the size of the array variable is set smaller than the number of data in the cam table, the data of the entire cam table may not be read because the cam data is read only by the array size.

6.6.11 Write Cam data (LS_WriteCamData)

Function Block type



Input-Output		
UINT	Axis	Specify the Axis to be commanded (See 6.2.1 Setting Range by Product)
Input		
BOOL	Execute	Give the cam data writing command in the rising Edge of the input.
UINT	CamTableID	Specify the ID of the cam table to write. (refer toSee 6.2.1 Setting Range by Product)
LREAL	StartSlope	Specify the StartSlope value of the cam table to write.
LREAL	EndSlope	Specify the StartSlope value of the cam table to write.
UINT	CamPointNum	Specify the cam data point number of the cam table to write.
LREAL[]	MasterPoint	Of the cam data to write, set the leading address of the device where Master Point value is stored.
LREAL[]	SlavePoint	Of the cam data to write, set the leading address of the device where Slave Point value is stored.
BYTE[]	CamCurveSel	Of the cam data to write, set the leading address where cam curve type is stored. (0: Linear, 1: Cubic)
UINT	ExecutionMode	Set the timing to write the cam data. 0 - Immediately applied, 1: Applied at the same point with 'Buffered' of Buffermode
Output		
BOOL	Done	This represents successful cam data writing.
BOOL	Busy	Indicate that execution of the function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the error number that occurred while the function block is running.

(1) This motion function block is a command to write the data value of the cam table. Of the cam table data set by CamTableID input, use the value of the device set at MasterPoint and Slave Point at the value set at StartSlope and EndSlope and the set number at CamPointNum as the MasterPoint and SlavePoint values.

(2) CamTableID input can be set to between 1 and 32. Setting a value outside the above range will cause "Error 16#000B"

(3) You can enter a value below the number of settings set in the existing cam profile into CamPointNum. If the CamPointNum value is larger than the exiting cam profile, an error 16#111C"occurs.

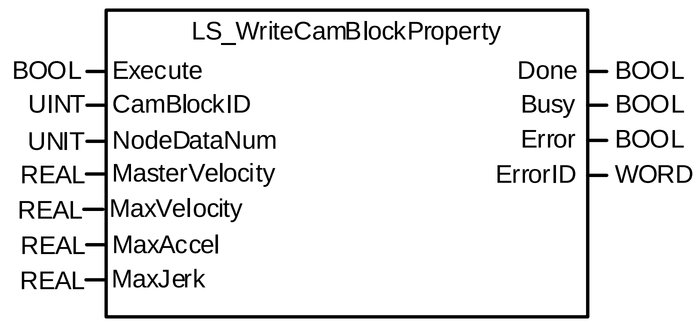
(4) If the size of MasterPoint / SlavePoint / CamCurveSel array is set to a value smaller than CamPointNum, an "error 16#000B" occurs.

(5) ExecutionMode input sets the setting timing. When the input is 0, setting is performed upon executing the command. When the input is 1, setting is performed at the same time as "Buffered" at the sequential operation. If the value that cannot be set is applied, "Error 16 # 000B" occurs.

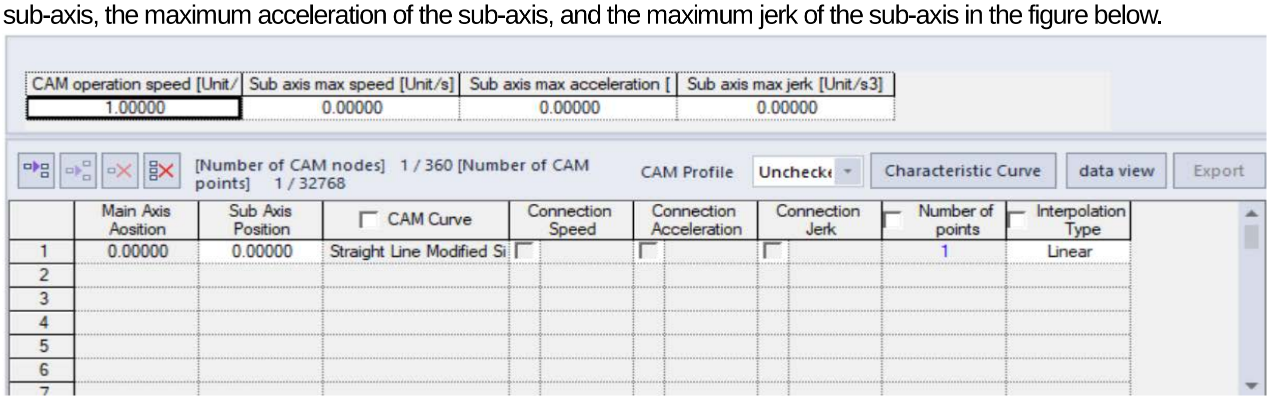
0(mcImmediately) : Itchanges the (Upward Edge of Execute input) parameter value upon executing the function block. If the relevant axis is in running, operation can be affected.

1(mcQueued) : It is changed at the same point of time as in "Buffered" of Buffermode. (6.1. 4. Command buffering: refer to BufferiMode)

6.6.12 Write Cam Block Property (LS_WriteCamBlockProperty)

Motion Function Block		
		
Input		
BOOL	Execute	Write cam block data at the rising edge of the input.
UINT	CamBlockID	Enter the setting block number. (1~100)
UINT	NodeDataNum	Enter the number of configured nodes.
REAL	CamCycleTime	Enter the time value for 1 cycle of CAM operation. (sec)
REAL	MaxVelocity	Enter the maximum cam operation speed value.
REAL	MaxAccel	Enter the maximum cam operation acceleration value.
REAL	MaxJerk	Enter the maximum cam operation jerk value.
Output		
BOOL	Done	Indicate the validity of the function block output.
BOOL	Busy	Indicate that execution of the function block is not completed.
BOOL	Active	Indicates whether an error occurs or not.
WORD	ErrorID	Output the error number that occurred while the function block is running.

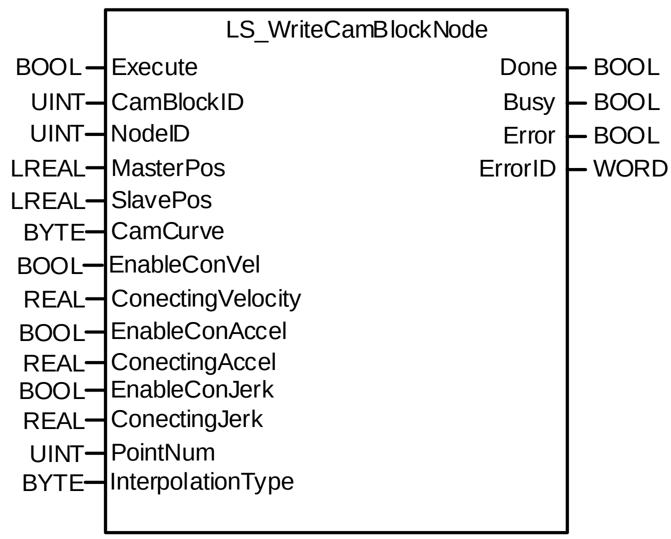
(1) The cam block function creates a cam profile by setting the cam curve, connection speed, connection acceleration, and connection jerk. This motion function block is a command to write properties during the cam profile creation stage to perform cam operation of the main and subordinate axes. The properties refer to the cam operation speed, the maximum speed of the sub-axis, the maximum acceleration of the sub-axis, and the maximum jerk of the sub-axis in the figure below.



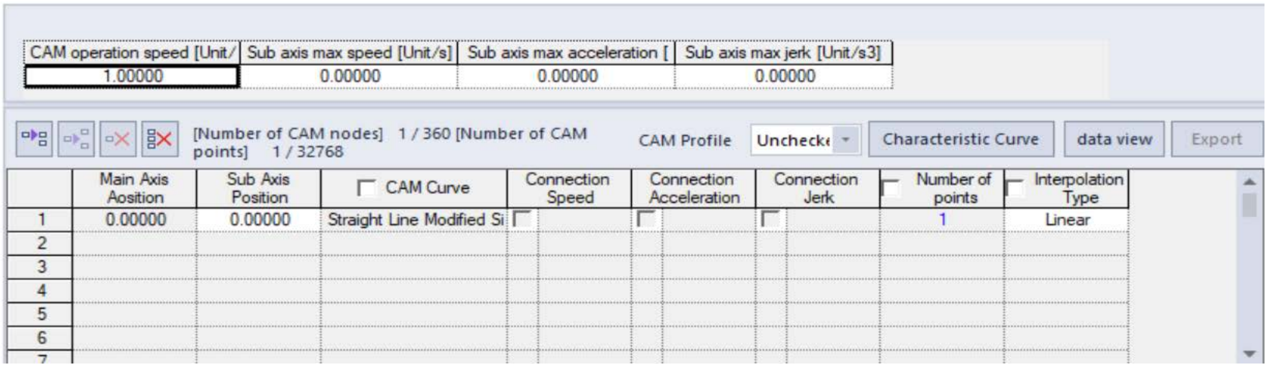
(2) Function block is the first command used among cam block related function blocks LS_WriteCamblockProperty, LS_WriteCamblockNode, LS_GenerateCamData, LS_ReadCamblockProperty, and LS_ReadCamblockNode.

(3) You can check the settings of LS_WriteCamblockProperty through the LS_ReadCamblockProperty command. If a value outside the set range is input to CamBlockID, Error becomes ‘1’ and ErrorID ‘0x00C1’ occurs.

6.6.13 Write Cam Block node (LS_WriteCamBlockNode)

Motion Function Block		
		
Input		
BOOL	Execute	Write cam block data at the rising edge of the input.
UINT	CamBlockID	Enter the setting block number. (1~100)
UINT	NodeID	Enter the setting node number. (0~360)
LREAL	MasterPos	Enter the position of the main axis of the cam block.
LREAL	SlavePos	Enter the position of the sub axis of the cam block.
Byte	CamCurve	Enter the Cam curve type of the cam block.
BOOL	EnableConVel	Enter whether to use the setting for the connection speed.
LREAL	ConnectingVelocity	Enter the connection speed of the cam block.
BOOL	EnableConAccel	Enter whether to use the setting for the connection acceleration.
LREAL	ConnectingAccel	Enter the connection acceleration of the cam block.
BOOL	EnableConJerk	Enter whether to use the setting for the connection jerk.
LREAL	ConnectingJerk	Enter the connection jerk of the cam block.
UINT	PointNum	Enter the number of cam points of the cam block.
Byte	InterpolationType	Enter the interpolation method of the cam block. (0:Linear 1:Cubic)
Output		
BOOL	Done	Indicate the validity of the function block output.
BOOL	Busy	Indicate that execution of the function block is not completed.
BOOL	Active	Indicates whether an error occurs or not.
WORD	ErrorID	Output the error number that occurred while the function block is running.

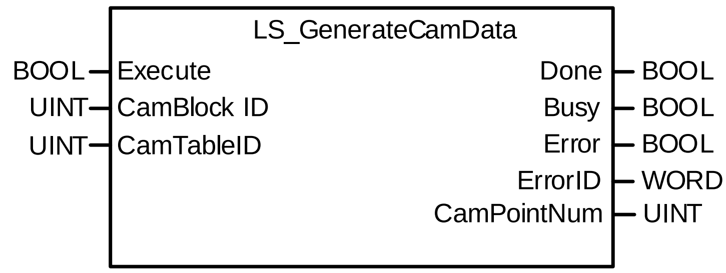
(1) The cam block function creates a cam profile by setting the cam curve, connection speed, connection acceleration, and connection jerk. This motion function block is a command to write node parameter during the cam profile creation stage to perform cam operation of the main and subordinate axes. Node parameters refer to main axis position, subordinate axis position, cam curve, connected speed, connected acceleration, number of points, and interpolation type in the figure below.



(2) This function block is a command used after the cam block related function block LS_WriteCamblockProperty is executed. Also, if there is no cam block, execute WriteCamBlockProperty to designate the number of nodes in the cam block.
(3) Enter the cam block number to write in CamBlockID.
(4) Enter the number of nodes in the cam block in NodeDataNum. (Ex 20)
(5) You can check the details of cam curve and interpolation type in Chapter 8, Cam Block Function.
(6) You can check the settings of LS_WriteCamblockNode through the LS_ReadCamblockNode command.
(7) If a value outside the set range is input to CamBlockID, Error becomes ‘1’ and ErrorID ‘0x00C1’ occurs.
(8) If a value outside the set range is input to Node ID, Error becomes ‘1’ and ErrorID ‘0x00C3’ occurs.
(9) If a value outside the set range is input to PointNum, Error becomes ‘1’ and ErrorID ‘0x00C4’ occurs.

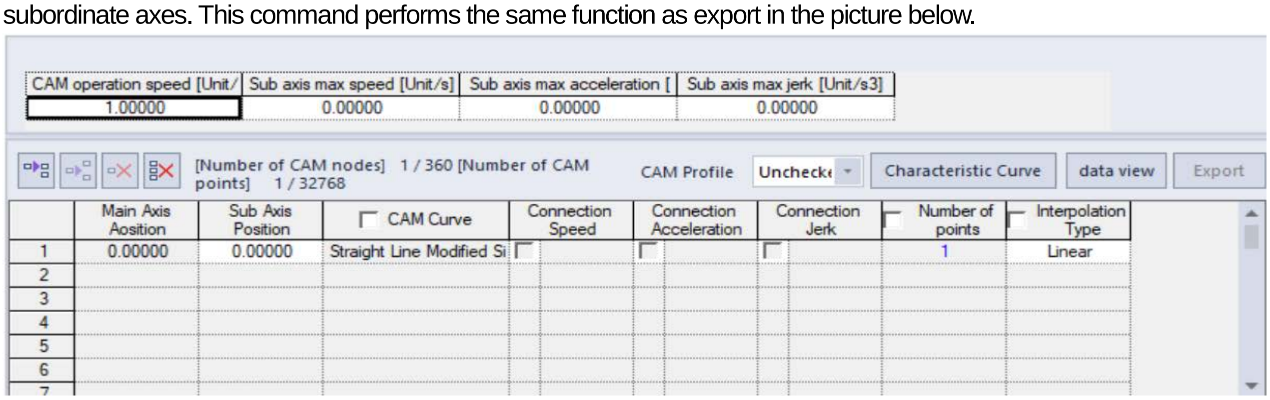
6.6.14 Generate Cam data (LS_GenerateCamData)

Motion Function Block



Input		
BOOL	Execute	Write cam block data at the rising edge of the input.
UINT	CamBlockID	Enter the setting block number. (1~100)
UINT	CamTableID	Specify the cam table Number to generate. (1~32)
Output		
BOOL	Done	Indicate the validity of the function block output.
BOOL	Busy	Indicate that execution of the function block is not completed.
BOOL	Active	Indicates whether an error occurs or not.
WORD	ErrorID	Output the error number that occurred while the function block is running.
UINT	CamPointNum	Outputs the number of generated cam data.

(1) The cam block function creates a cam profile by setting the cam curve, connection speed, connection acceleration, and connection jerk. This motion function block is the cam profile creation stage to perform cam operation of the main and subordinate axes. This command performs the same function as export in the picture below.

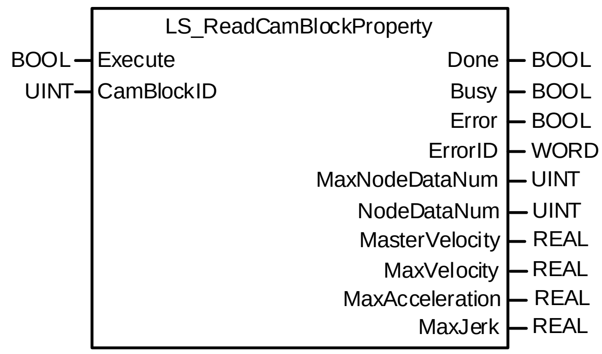


(2) This motion function block is a function block that creates a camp profile from the stored cam block data.

(3) The function block is executed at the rising edge of the Execute input.

6.6.15 Read Cam Block Property (LS_ReadCamBlockProperty)

Motion Function Block



Input		
BOOL	Execute	Write cam block data at the rising edge of the input.
UINT	CamBlockID	Input the number of cam block to read. (1~100)
Output		
BOOL	Done	Indicate the validity of the function block output.
BOOL	Busy	Indicate that execution of the function block is not completed.
BOOL	Active	Indicates whether an error occurs or not.
WORD	ErrorID	Output the error number that occurred while the function block is running.
UINT	NodeDataNum	Output the number of configured nodes.
REAL	MaxVelocity	Output the maximum cam operation speed value.
REAL	MaxAccel	Output the maximum cam operation acceleration value.
REAL	MaxJerk	Output the maximum cam operation jerk value.

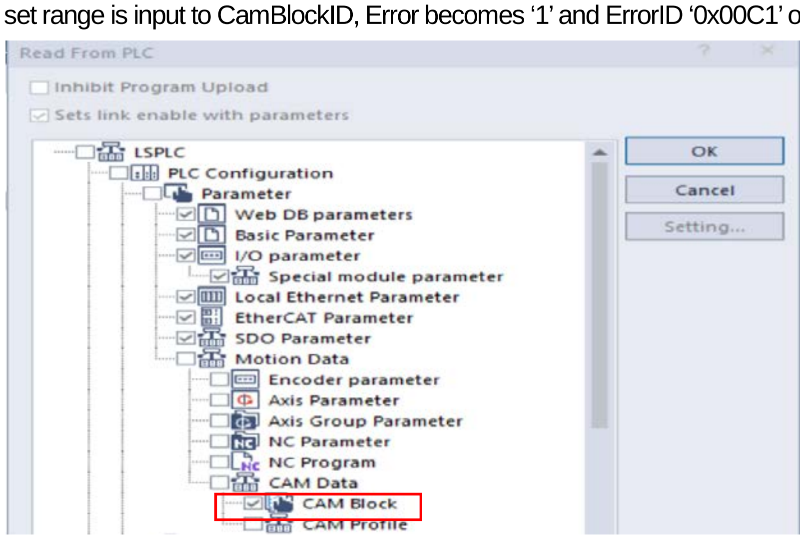
(1) The cam block function creates a cam profile by setting the cam curve, connection speed, connection acceleration, and connection jerk. This command can check the content set by LS_WriteCamblockProperty. This motion function block is a function block that reads the attribute data of the designated cam block.

(2) The function block is executed at the rising edge of the Execute input.

(3) In BlockID, enter the cam block number. A value between 1 and 100 can be entered for the BlockID value.

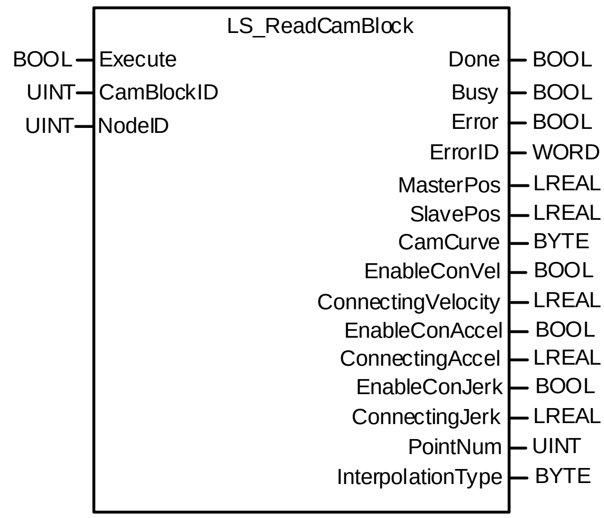
(4) If a value outside the set range is input to CamBlockID, Error becomes ‘1’ and ErrorID ‘0x00C1’ occurs.

(5) If a value outside the set range is input to CamBlockID, Error becomes ‘1’ and ErrorID ‘0x00C1’ occurs.



6.6.16 Read Cam Block node (LS_ReadCamBlockNode)

Motion Function Block

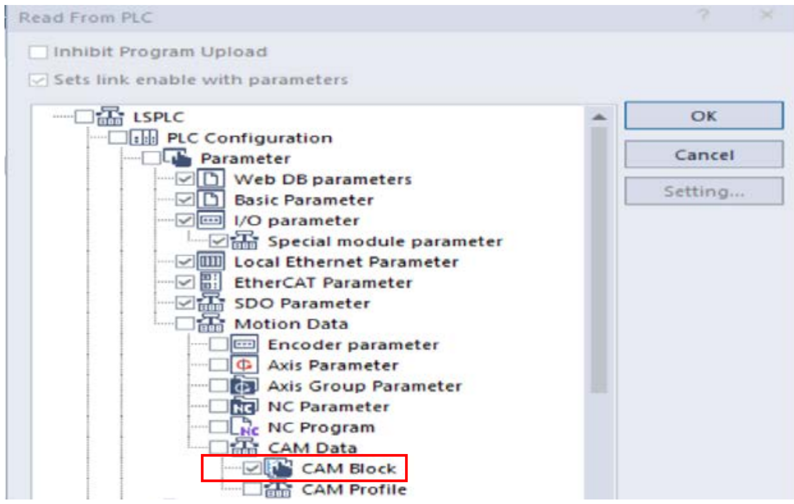


Input		
BOOL	Execute	Write cam block data at the rising edge of the input.
UINT	CamBlockID	Input the number of cam block to read. (1~100)
UINT	NodeID	Input the number of node to read. (1~100)
Output		
BOOL	Done	Indicate the validity of the function block output.
BOOL	Busy	Indicate that execution of the function block is not completed.
BOOL	Active	Indicates whether an error occurs or not.
WORD	ErrorID	Output the error number that occurred while the function block is running.
LREAL	MasterPos	Output the position of the main axis set in the cam block.
LREAL	SlavePos	Output the position of the sub axis set in the cam block.
Byte	CamCurve	The cam curve type set in the cam block is output..
BOOL	EnableConVel	It outputs whether the setting for the connection speed set in the cam block is used or not.
LREAL	ConnectingVelocity	Outputs the connection speed set in the cam block.
BOOL	EnableConAccel	It outputs whether the setting for the connection acceleration set in the cam block is used or not.
LREAL	ConnectingAccel	Outputs the connection acceleration set in the cam block.
BOOL	EnableConJerk	It outputs whether the setting for the connection jerk set in the cam block is used or not.
LREAL	ConnectingJerk	Outputs the connection jerk set in the cam block.
UINT	PointNum	Enter the number of cam points of the cam block.
Byte	InterpolationType	Outputs the number of cam points set in the cam block.

(1) The cam block function creates a cam profile by setting the cam curve, connection speed, connection acceleration, and connection jerk. This command can check the content set by LS_WriteCamblocNode. This motion function block is a function block that reads the node data of the designated cam block.

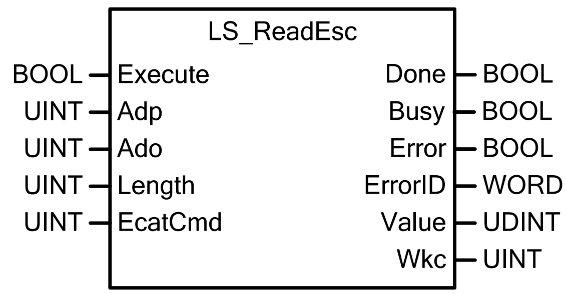
(2) The function block is executed at the rising edge of the Execute input.

(3) In BlockID, enter the cam block number. A value between 1 and 100 can be entered for the BlockID value.
(4) If a value outside the set range is input to CamBlockID, Error becomes ‘1’ and ErrorID ‘0x00C1’ occurs.
(5) If a value outside the set range is input to Node ID, Error becomes ‘1’ and ErrorID ‘0x00C3’ occurs.
(6) If a value outside the set range is input to CamBlockID, Error becomes ‘1’ and ErrorID ‘0x00C1’ occurs.



6.6.17 Read ESC (LS_ReadEsc)

Function Block type



Input		
BOOL	Execute	Give the ESC reading command to the slave controller in the rising Edge.
UINT	Adp	Set the slave controller address according to the EcatCmd.
UINT	Ado	Set the slave controller ESC address.
UINT	Length	Set the data length to read. (1 ~ 4 Byte)
UINT	EcatCmd	Set the EtherCAT command. (1: APRD, 4: FPRD, 7: BRD)

Output		
BOOL	Done	This represents successful ESC reading to complete normally.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.
UDINT	Value	Output the ESC reading value of the slave controller
UINT	Wkc	After the execution of the command, Working Counter value is displayed.

(1) This motion function block is a function block to read the data of the address in Ado set from the ESC (EtherCAT Slave Controller) of the designated slave device.

(2) Value and Wkc (Working Counter) is displayed as 0 when the motion function block is executed. When the execution is completed (Done output is on), the read data value is displayed at Value, and the Working Counter value is displayed at Wkc.

(3) Adp input specifies the EtherCAT slave device address, and the following values can be set depending on EcatCmd settings. If EcatCmd setting is 7(BRD), Adp input value is ignored. If a value outside the range is set for Adp input, “Error 0x0F60” occurs.

EcatCmd	Adp range
1 (APRD)	0x0000: The first slave connected 0xFFFF: The second slave connected 0xFFFE: The third slave connected : 0xFFC1: The 64th slave connected
4 (FPRD)	1 ~ 64: slave 1~slave 64
7 (BRD)	-

(4) In Length, values ranging from 1 to 4 can be entered, which mean 1 to 4 bytes. If the value is set outside the range, “error 0x0F61” occurs.

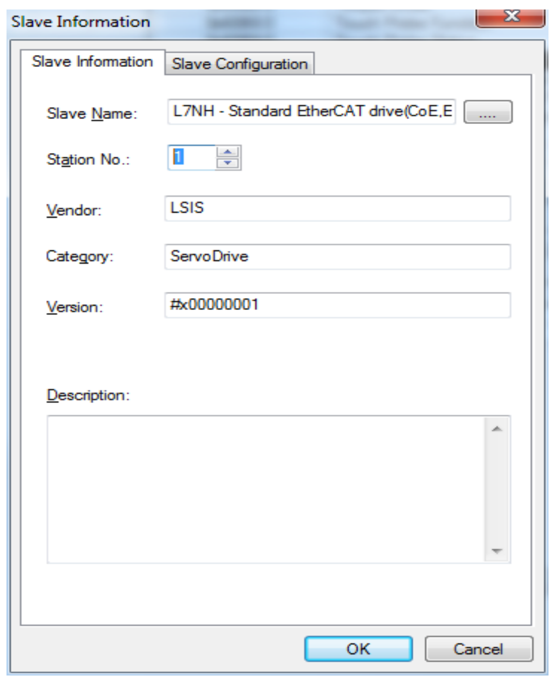
(5) At EcatCmd, the type of command to use when reading ESC (EtherCAT Slave Controller) is specified. The following three commands can be used. Setting a value outside the range at EcatCmd will cause "Error 0x0F62”.

1) 1 - APRD (Auto Increment Physical Read)

This command is used when reading the slave device data following the order of physical connection before normal communication connection by the master. A slave device receiving Adp with 0 value will read data of the size designated by Length. Adp of each slave device increases when EtherCAT frame is received. For example, if EcatCmd is 1, and Adp is set to 0xFFFF, when executing ESC read function block, reading is not performed because the Adp at the time of receiving EtherCAT frame from the first slave device is not 1, only increasing Adp by 1. When the second slve device receives EtherCAT frame, reading is performed because the Adp value of the first slave value increased by 1 to 0. The Adp setting values depending on the slave device connection order are as follows.

Slave device	Setting value
The first slave connected	0
The second slave connected	0xFFFF
:	:
The 64th slave connected	0xFFC1

2) 4 - FPRD (Configured Address Physical Read)
This order is used to read the data by designating the station address of the slave device after normal communication connection by the master. If the Station Address of the slave device set by EtherCAT master matches the transmitted Adp, the slave device reads data of the size designated by Length in the Ado area. The Station Address of slave device set by master can be checked in slave information dialog box when the slave is added.



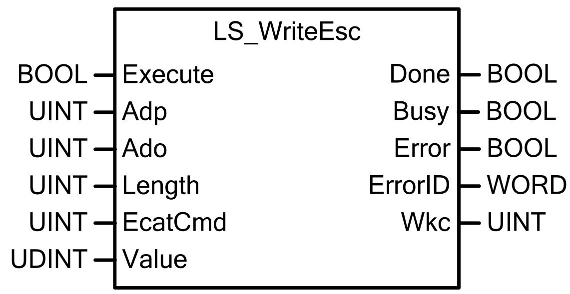
3) 7 – Read Broadcast (Broadcast Read)
All connected slave devices read data of the size set by Length in the Ado area, and saves the result after Bitwise-OR. The designated address value at Adp is ignored, and Wkc increase by 1due to all slaves that performed normal read operation

(6) Wkc stands for Working Counter. If data is successfully read at the designated slave device, it increases by 1. If EcatCmd is 7(BRD), it increases by 1 due to all slaves that performed normal read operation.

(7) After the execution of ESC read command, if normal data read operation is executed from the designated slave device, Done output is on.

6.6.18 Write ESC (LS_WriteEsc)

Function Block type



Input		
BOOL	Execute	Give the ESC writing command to the slave controller in the rising Edge.
UINT	Adp	Set the slave controller address according to the EcatCmd.
UINT	Ado	Set the slave controller ESC address.
UINT	Length	Set the data length to write. (1 ~ 4 Byte)
UINT	EcatCmd	Set the EtherCAT command. (2: APWR, 5: FPWR, 8: BWR)
UDINT	Value	Output the ESC writing value of the slave controller
Output		
BOOL	Done	This represents successful ESC writing to complete normally.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.
UINT	Wkc	After the execution of the command, Working Counter value is displayed.

(1) This motion function block writes data using the address set by Ado to ESC (EtherCAT Slave Controller) of the slave device set by Adp.

(2) Wkc value is displayed as 0 when the motion function block is executed, and the Working Counter value is displayed when execution is completed (Done output is on). Wkc increases by 1 through each slave device designated by EcatCmd and Adp.

(3) Adp input specifies the EtherCAT slave device address, and the following values can be set depending on EcatCmd settings. If EcatCmd setting is 8(BWR), Adp input value is ignored. If a value outside the range is set for Adp input, “Error 0x0F70” occurs.

EcatCmd	Adp range
2 (APWR)	0x0000: The first slave connected 0xFFFF: The second slave connected 0xFFFE: The third slave connected : 0xFFC1: The 64th slave connected
5 (FPWR)	1 ~ 64: slave 1~slave 64
8 (BWR)	-

(4) In Length, values ranging from 1 to 4 can be entered, which mean 1 to 4 bytes. If the value is set outside the range, “error 0x0F71” occurs.

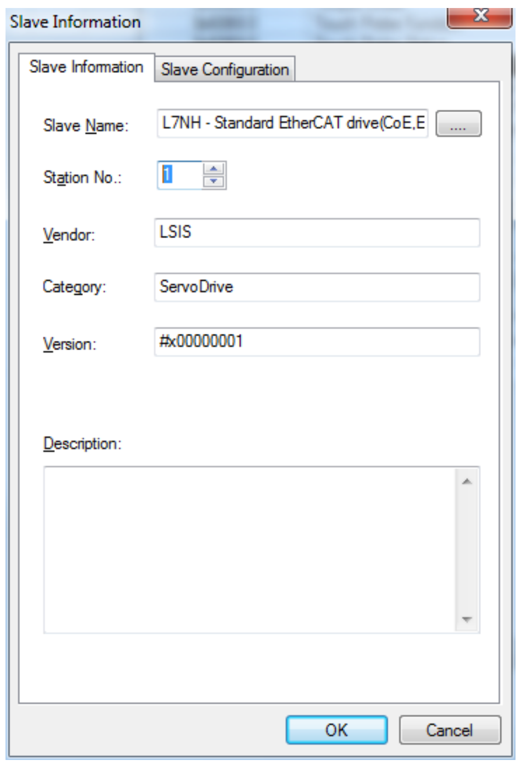
(5) At EcatCmd, the type of command to use when reading ESC (EtherCAT Slave Controller) is specified. The following write commands can be used. Setting a value outside the range at EcatCmd will cause "Error 0x0F72”.

1) 2- APWR(Auto Increment Physical Write)

This command is used when reading the slave device data following the order of physical connection before normal communication connection by the master. A slave device receiving Adp with 0 value will read data of the size designated by Length. Adp of each slave device increases when EtherCAT frame is received. For example, if EcatCmd is 2, and Adp is set to 0xFFFF, when executing ESC read function block, reading is not performed because the Adp at the time of receiving EtherCAT frame from the first slave device is not 0, only increasing Adp by 1. When the second slave device receives EtherCAT frame, writing is performed because the Adp value of the first slave value increased by 1 to 0. The Adp setting values depending on the slave device connection order are as follows.

Slave device	Setting value
The first slave connected	0
The second slave connected	0xFFFF
:	:
The 64th slave connected	0xFFC1

2) 5 – FPWR(Configured Address Physical Write)
This order is used to write the data by designating the station address of the slave device after normal communication connection by the master. If the Station Address of the slave device set by EtherCAT master matches the transmitted Adp, the slave device writes data of the size designated by Length in the Ado area. The Station Address of slave device set by master can be checked in slave information dialog box when the slave is added.



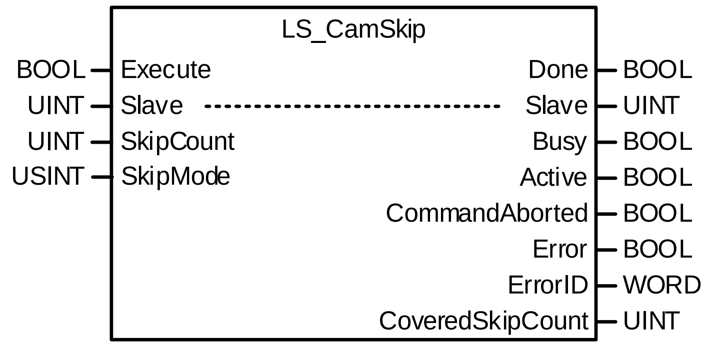
3) 8 – Write Broadcast BWR, Broadcast Write)
All connected slave devices write data of the size set by Length in the Ado area. The designated address value at Adp is ignored, and Wkc increase by 1 due to all slaves that performed normal write operation.

(6) Wkc stands for Working Counter. If data is successfully written at the designated slave device, it increases by 1. If EcatCmd is 8(BWR), it increases by 1 through each slave device that performed normal write operation.

(7) After the execution of ESC write command, if normal data write operation is executed in the specified slave device, Done output is on.

6.6.19 Skip Cam (LS_CamSkip)

Motion Function Block

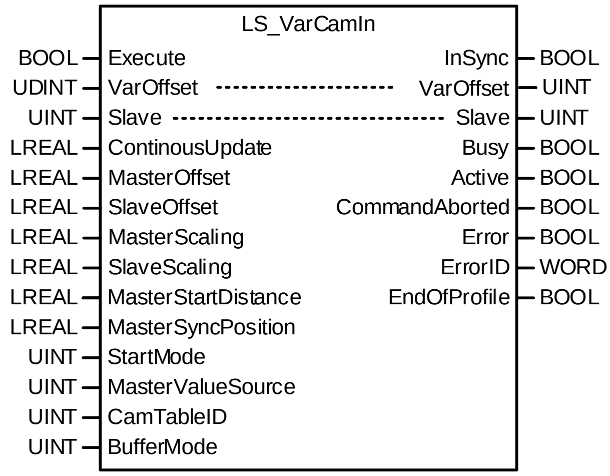


Input-Output		
UINT	Slave	Set the sub axis. (refer toSee 6.2.1 Setting Range by Product)
Input		
BOOL	Execute	Give cam skip command on the axis in the rising Edge.
UINT	SkipCount	Set the number of cam cycles to skip.
USINT	SkipMode	Set cam skip mode. (Reservation)
Output		
BOOL	Done	Indicate the completion state of cam skip operation.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that the current axis is controlling the cam skip.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted while it is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.
LREAL	CoveredSkipcount	Output the number of cam cycle skipped.

(1) This motion function block commands Cap Skip command which skip cam operation cycles as designated for the cam currently in operation.
(2) SkipCount determines the number of cam cycles to skip. If 0 is entered, SkipCount Error (Error 0x111E) is displayed.
(3) When Cam Skip command is issued on a sub-axis where cam operation is underway, the current cam cycle ends, and the skip operation starts. During cam skip, the sub-axis is in stand-by at the end of the cam table.
(4) CoveredSkipCount displays the number of cam cycles skipped. The count increases with each skipped cycle, and becomes 0 when Done output is off after the function block motion is completed
(5) Done output is on when the set number of cycles are skipped after executing Cam Skip command.

6.6.20 Variable Cam operation (LS_VarCamIn)

Motion Function Block



Input-Output		
UDINT	VarOffset	Set the offset value of the M device where the variable to be used as the main axis is located.
UINT	Slave	Set the sub axis. (refer toSee 6.2.1 Setting Range by Product)
Input		
BOOL	Execute	Give cam operation command to the relevant axis in the rising Edge.
BOOL	ContinuousUpdate	Specify the update setting of input value. (Refer to 6.1.5.Changes in Parameters during Execution of Motion Function Block)
LREAL	MasterOffset	Set the offset value of the main axis.
LREAL	SlaveOffset	Set the offset value of the slave cam table.
LREAL	MasterScaling	Specify the magnification of the main axis.
LREAL	SlaveScaling	Specify the magnification of the slave axis cam table.
LREAL	MasterStartDistance	Specify the position of the main-axis where the cam operation of the master axis starts.
LREAL	MasterSyncPosition	Specify the starting point at cam table when cam operation starts.
UINT	StartMode	Set the cam operation mode. 0: Cam table is applied as an absolute value. (mcAbsolute) 1: Cam table is applied as a relative value based on the command start position. (mcRelative)
UINT	MasterValueSource	Select the source of the main axis for cam operation. 0: Synchronizes to the command position of the main axis. 1: Synchronizes to the current position of the main axis.
UINT	CamTableID	Specify the cam table to operate. (refer toSee 6.2.1 Setting Range by Product)
UINT	BufferMode	Specify the sequential operation setting of motion function block. (Refer to 6.1.4.BufferMode input)

Output		
BOOL	InSync	Indicate that cam operation is normally being fulfilled. (Indicate that the serve axis is following the cam table.)
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that the current motion function block is controlling the relevant axis.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted while it is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

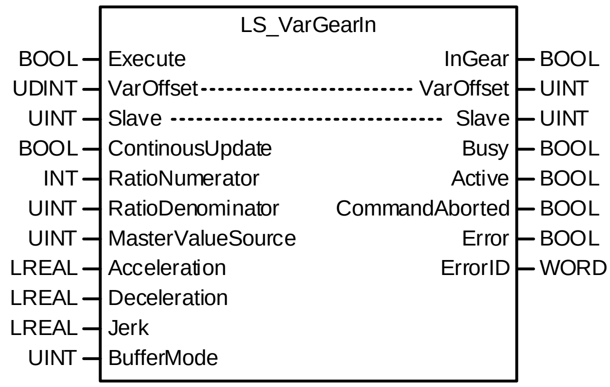
(1) This motion function block is the function block that operates the sub-axis CAM along the main axis by setting the variable value designated by offset as the main axis.

(2) The variable value specified as the main axis should be the LREL type. Example). When specifying the variable to be allocated to the memory by %ML100 as the main axis value, %ML100 should be LREAL type, and the offset value specifying a variable is UDINT type and you should input 100 to the VarOffset.

(3) Remaining settings and functions are the same as the MC_CamIn function block.

6.6.21 Variable gear operation (LS_VarCamIn)

Motion Function Block



Input-Output		
UDINT	VarOffset	Set the offset value of the M device where the variable to be used as the main axis is located.
UINT	Slave	Set the sub axis. (refer toSee 6.2.1 Setting Range by Product)
Input		
BOOL	Execute	Give gear operation command to the relevant axis in the rising Edge.
BOOL	ContinuousUpdate	Specify the update setting of input value. (Refer to 6.1.5.Changes in Parameters during Execution of Motion Function Block)
INT	RatioNumerator	Specify the numerator of gear ratio. (-32768 ~ 32767)
UINT	RatioDenominator	Specify the denominator of gear ratio. (0 ~ 65535)
UINT	MasterValueSource	Select data of the main axis to be synchronized. 0: Synchronize in the command position of the main axis. 1: Synchronize in the current position of the main axis.
LREAL	Acceleration	Specify the acceleration at the beginning of gear operation synchronization. [u/s^2]
LREAL	Deceleration	Specify the deceleration at the beginning of gear operation synchronization. [u/s^2]
LREAL	Jerk	Specify the change rate of acceleration/deceleration. [u/s^3]
UINT	BufferMode	Specify the sequential operation setting of motion function block. (Refer to 6.1.4 BufferMode Input)
Output		
BOOL	InGear	Indicate that gear operation is running by applying gear ration.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that the current motion function block is controlling the relevant axis.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted while it is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

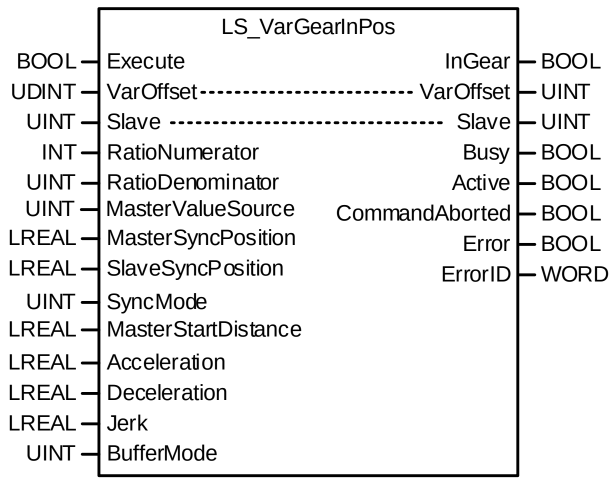
(1) This motion function block is the function block that drives the main axis and the sub axis in gear operation (speed synchronization) by setting the variable value designated by offset as the main axis.

(2) The variable value specified as the main axis should be the LREL type. Example). When specifying the variable to be allocated to the memory by %ML100 as the main axis value, %ML100 should be LREAL type, and the offset value specifying a variable is UDINT type and you should input 100 to the VarOffset.

(3) Remaining settings and functions are the same as the MC_GearIn function block.

6.6.22 Variable gearing by specifying the position (LS_VarGearInPos)

Motion Function Block



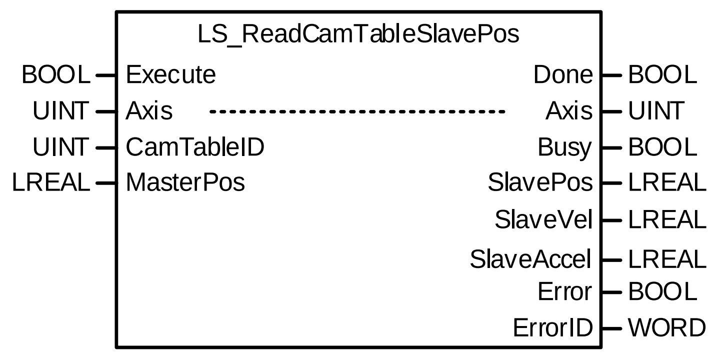
Input-Output		
UDINT	VarOffset	Set the offset value of the M device where the variable to be used as the main axis is located.
UINT	Slave	Set the sub axis. (refer toSee 6.2.1 Setting Range by Product)
Input		
BOOL	Execute	Give gear operation command to the relevant axis in the rising Edge.
INT	RatioNumerator	Specify the numerator of gear ratio. (-32768~32767)
UINT	RatioDenominator	Specify the denominator of gear ratio. (0~65535)
UINT	MasterValueSource	Select the standard of the main axis value to be synchronized. 0(mcSetValue): Synchronize in the target position of the main axis. 1(mcActualValue): Synchronize in the current position of the main axis.
LREAL	MasterSyncPosition	Specify the position of the main axis where gear operation starts.
LREAL	SlaveSyncPosition	Specify the position of the spindle where gear operation starts.
LREAL	SyncMode	Unused
LREAL	MasterStartDistance	Specify the distance of the main axis where synchronization starts.
LREAL	Velocity	Specify the maximum speed of the spindle at the beginning of synchronization. [u/s]
LREAL	Acceleration	Specify the maximum acceleration of the spindle at the beginning of synchronization. [u/s^2]
LREAL	Deceleration	Specify the maximum deceleration of the spindle at the beginning of synchronization. [u/s^2]
LREAL	Jerk	Unused
UINT	BufferMode	Specify the sequential operation setting of motion function block. (Refer to 6.1.4 BufferMode Input)
Output		
BOOL	InSync	Indicate that gear operation is normally being fulfilled as the specified gear ratio is applied.
BOOL	StartSync	Indicate synchronization is starting.

BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that the current motion function block is controlling the relevant axis.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted while it is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block is the function block that synchronizes the main axis and the servo axis according to the gear ratio set at the specific position by setting the variable value designated by the offset as the main axis
(2) The variable value specified as the main axis should be the LREL type. Example). When specifying the variable to be allocated to the memory by %ML100 as the main axis value, %ML100 should be LREAL type, and the offset value specifying a variable is UDINT type and you should input 100 to the VarOffset.
(3) Remaining settings and functions are the same as the MC_GearInPos function block.

6.6.23 Read the slave location of the CAM table (LS_ReadCamTableSlavePos)

Motion Function Block



Input-Output		
UINT	Axis	Set the command axis. (refer toSee 6.2.1 Setting Range by Product)
Input		
BOOL	Execute	Give ReadCamTableSlavePos operation command to the relevant axis in the rising Edge.
UINT	CamTableID	Specify the cam table to read. (refer toSee 6.2.1 Setting Range by Product)
LREAL	MasterPos	Specify the position of the main axis on the CAM table.
Output		
BOOL	Done	Indicate the completion state of ReadCamTabelSlavePos operation.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
LREAL	SlavePos	It outputs the position of the slave.
LREAL	SlaveVel	It outputs the speed of the slave. [u/s]
LREAL	SlaveAccel	It outputs the acceleration of the slave. [u/s^2]
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

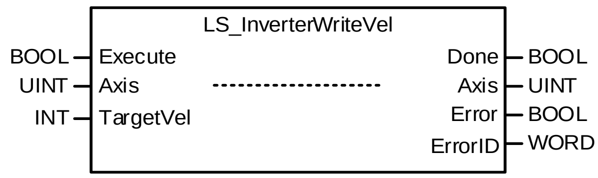
(1) This motion function block outputs the position of the serve axis according to the position of the main axis in the specified CAM table.

(2) Set the position value of the main axis to be read in the CAM table as the MasterPos value. Offset / gear ratio / phase correction operation, etc. applied to the command axis are not reflected in the SlavePos output.

(3) When reading the slave position on the CAM table is completed, the ‘Done Output’ will be turned on.

6.6.24 Write inverter speed (LS_InverterWriteVel)

Motion Function Block



Input-Output		
UINT	Axis	Specify the axis to be commanded. (See 6.2.1 Setting Range by Product, Real axis)
Input		
BOOL	Execute	Give InverterWriteVel operation command to the relevant axis in the rising Edge.
INT	TargetVel	The inverter speed to be set (-30000 ~ 30000, unit: rpm)
Output		
BOOL	Done	Indicate the completion state of InverterWriteVel operation.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

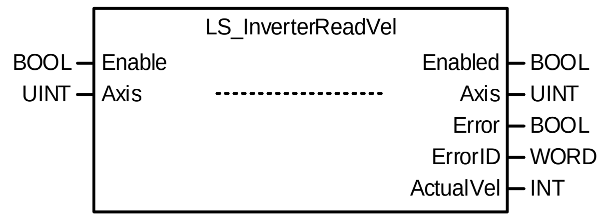
(1) This motion function block is the function block that sets the speed of the inverter to operate when controlling the inverter by the axis

(2) If you set the speed in TargetVel and execute the function block, the inverter connected to the axis will operate at the corresponding speed.

(3) The speed value set in TargetVel is in units of rpm, and can be set to the value from -30000 to 30000.

6.6.25 Read inverter speed (LS_InverterReadVel)

Motion Function Block

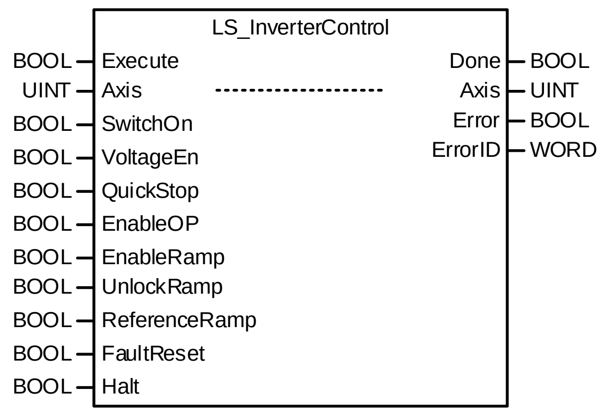


Input-Output		
UINT	Axis	Specify the axis to be commanded. (See 6.2.1 Setting Range by Product, Real axis)
Input		
BOOL	Enable	While the condition is ON, the speed of the inverter connected to the axis is read.
Output		
BOOL	Enabled	It indicates whether reading the inverter speed is being executed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.
INT	ActualVel	Speed value of the read inverter

(1) This motion function block is the function block that reads the speed of the connected inverter when controlling the inverter by the axis.
(2) When the function block is executed, the current speed of the inverter connected to the axis is read and displayed in ActualVel.
(3) The speed value set in ActualVel is in units of rpm, and can be displayed as the value from -30000 to 30000.

6.6.26 Write inverter control word (LS_InverterControl)

Motion Function Block



Input-Output		
UINT	Axis	Specify the axis to be commanded. (See 6.2.1 Setting Range by Product, Real axis)
Input		
BOOL	Execute	Set the inverter control word in the rising Edge.
BOOL	SwitchOn	Switch On
BOOL	VoltageEn	Voltage Enable
BOOL	QuickStop	Quick Stop
BOOL	EnableOP	Enable operation
BOOL	EnableRamp	Enable ramp
BOOL	UnlockRamp	Unlock ramp
BOOL	ReferenceRamp	Reference ramp
BOOL	FaultReset	Fault Reset
BOOL	Halt	Halt
Output		
BOOL	Done	It indicates whether or not the inverter control word setting is done normally.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block is the function block that sets the control word of the connected inverter when controlling the inverter by the axis.

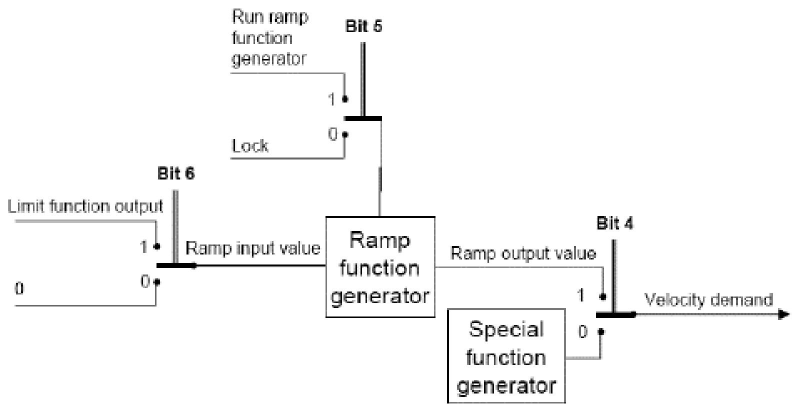
(2) In order to operate the inverter, the control word must be set to enable operation.

(3) Please refer to the following.

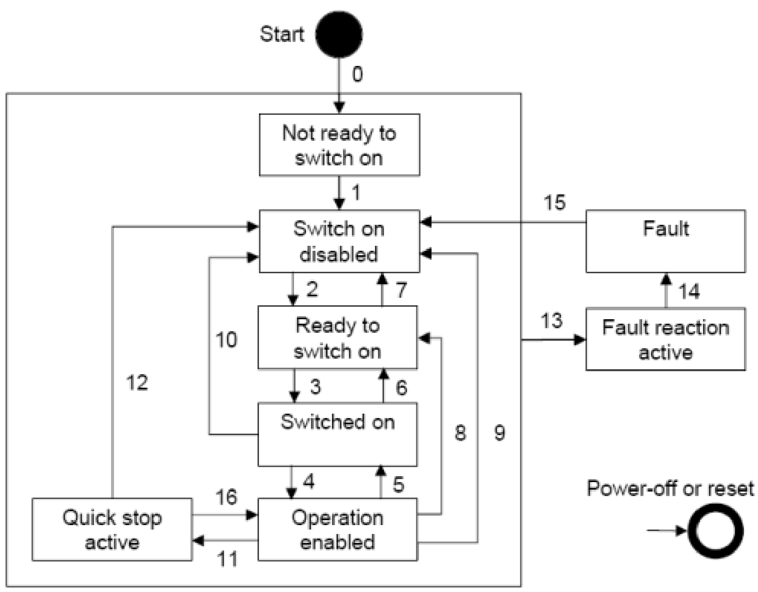
Command bit used in Enable Operation

Bit	Value	Description
4 (Enable Ramp)	0	Holding previous operation status
	1	Inverter operation by command bit
5 (Unlock Ramp)	0	Holding of output frequency
	1	Operatin to target freqency
6 (Reference Ramp)	0	Input target frequency as 0
	1	Input target frequency as settting value
8 (Halt)	X	Unused

Inverter status according to the bit setting of the control word



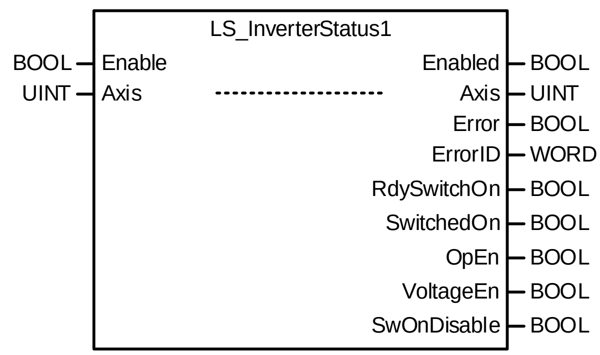
Change the inverter status according to the bit setting of the control word



Command	Bits of the *controlword*					Transitions
	Bit 7	Bit 3	Bit 2	Bit 1	Bit 0	
Shutdown	0	X	1	1	0	2,6,8
Switch on	0	0	1	1	1	3
Switch on + enable operation	0	1	1	1	1	3 + 4 (NOTE)
Disable voltage	0	X	X	0	X	7,9,10,12
Quick stop	0	X	0	1	X	7,10,11
Disable operation	0	0	1	1	1	5
Enable operation	0	1	1	1	1	4,16
Fault reset	↑	X	X	X	X	15
NOTE Automatic transition to Enable operation state after executing SWITCHED ON state functionality.						

6.6.27 Read inverter status 1 (LS_InverterStatus1)

Motion Function Block



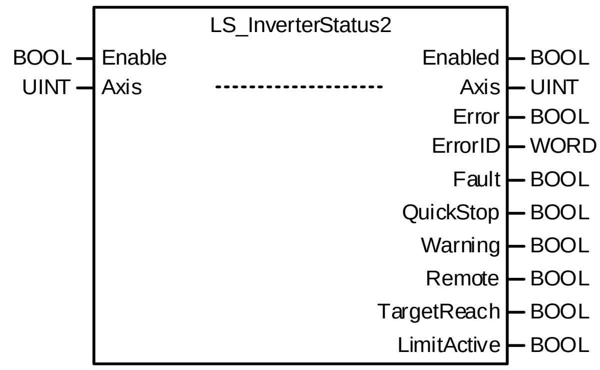
Input-Output		
UINT	Axis	Specify the axis to be commanded. (See 6.2.1 Setting Range by Product, Real axis)
Input		
BOOL	Enable	Read the “Status 1” of the inverter while the input is enabled.
Output		
BOOL	Enabled	It indicates the state of reading the “Status 1” of the inverter.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the error number that occurred while the function block is running.
BOOL	RdySwitchOn	Ready to Switch On
BOOL	SwitchedOn	Switched On
BOOL	OpEn	Operation Enabled
BOOL	VoltageEn	Voltage Enabled
BOOL	SwOnDisable	Switch On Disable

(1) This motion function block is the function block that reads and displays the "Status 1" of the connected inverter when controlling the inverter by the axis.
(2) RdySwitchOn, SwitchedOn, OpEn, VoltageEn, SwOnDisable are respectively the lower bit values of the Status Word among the inverter PDO Data.

RdySwitchOn	Bit 0
SwitchedOn	Bit 1
OpEn	Bit 2
VoltageEn	Bit 4
SwOnDisable	Bit 6

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
nu	nu	nu	Nu	lla	tr	rm	nu	w	sod	qs	Ve	f	oe	so	rtso

6.6.28 Read inverter status 2 (LS_InverterStatus2)

Motion Function Block		
		
Input-Output		
UINT	Axis	Specify the axis to be commanded. (See 6.2.1 Setting Range by Product, Real axis)
Input		
BOOL	Enable	Read the “Status 2” of the inverter while the input is enabled.
Output		
BOOL	Enabled	It indicates the state of reading the “Status 2” of the inverter.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the error number that occurred while the function block is running.
BOOL	Fault	Fault(trip)
BOOL	QuickStop	Quick stop
BOOL	Warning	Warning
BOOL	Remote	Remote
BOOL	TargetReach	Target Reached
BOOL	Limit Active	Internal Limit active

(1) This motion function block is the function block that reads and displays the "Status 2" of the connected inverter when controlling the inverter by the axis.

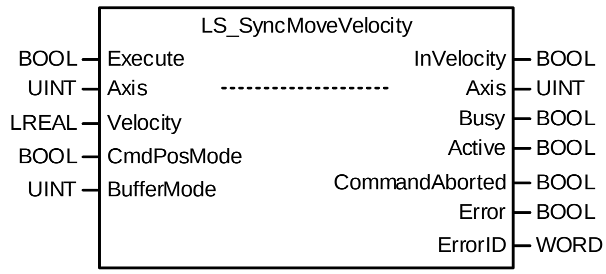
(2) Fault, QuickStop, Warning, Remote, TagetReach, LimiActive are respectively the lower bit values of the Status Word among the inverter PDO Data.

Fault	Bit 3
QuickStop	Bit 5
Warning	Bit 7
Remote	Bit 9
TargetReach	Bit10
LimitActive	Bit11

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
nu	nu	nu	Nu	lla	tr	rm	nu	w	sod	qs	Ve	f	oe	so	rtso

6.6.29 Speed control operation (CSV mode) (LS_SyncMoveVelocity)

Motion Function Block

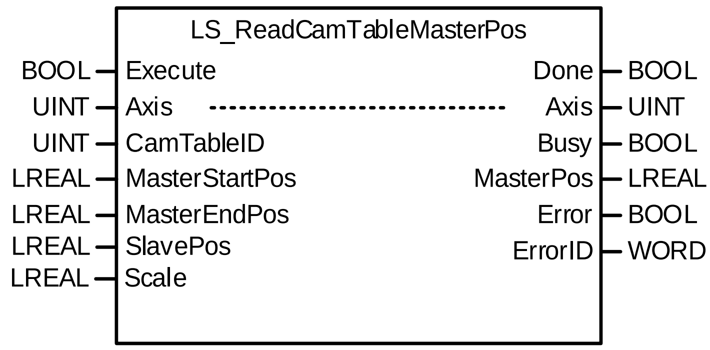


Input-Output		
UINT	Axis	Specify the axis to be commanded. (refer toSee 6.2.1 Setting Range by Product)
Input		
BOOL	Execute	In the rising Edge, it performs speed control operation through the CSV mode.
LREAL	Velocity	Setting the operation speed.
BOOL	CmdPosMode	0: Apply the current position to the command position.
UINT	BufferMode	Specify the sequential operation setting of motion function block. (0: Aborting, 1: Buffered) (Refer to 6.1.4.BufferMode input)
Output		
BOOL	Done	Indicate the state of motion function block completion.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that the current motion function block is controlling the relevant axis.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted while it is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the error number that occurred while the function block is running.

(1) This motion function block is the function block that allows speed control using the CSV (Cyclic Synchronous Velocity) mode of CiA402 profile on the set axis.
(2) In order to stop the specified speed operation, you can make a stop command or execute another motion function block.
(3) Velocity input specifies the speed to operate. Positive sign (+ or No sign) of the operation speed value leads to forward direction, and negative (-) sign leads to reverse direction.
(4) CmdPosMode is used to set the update methods of the current position at the time of command. Only the initial value of 0 is available and the current position of the command is updated using the feedback current position.
(5) The output InVelocity is turned on when the axis reaches the specified speed, and it is turned off when the specified speed operation is stopped.
(6) When this Motion Function Block is running, the axis status is 'Continuous Motion'.

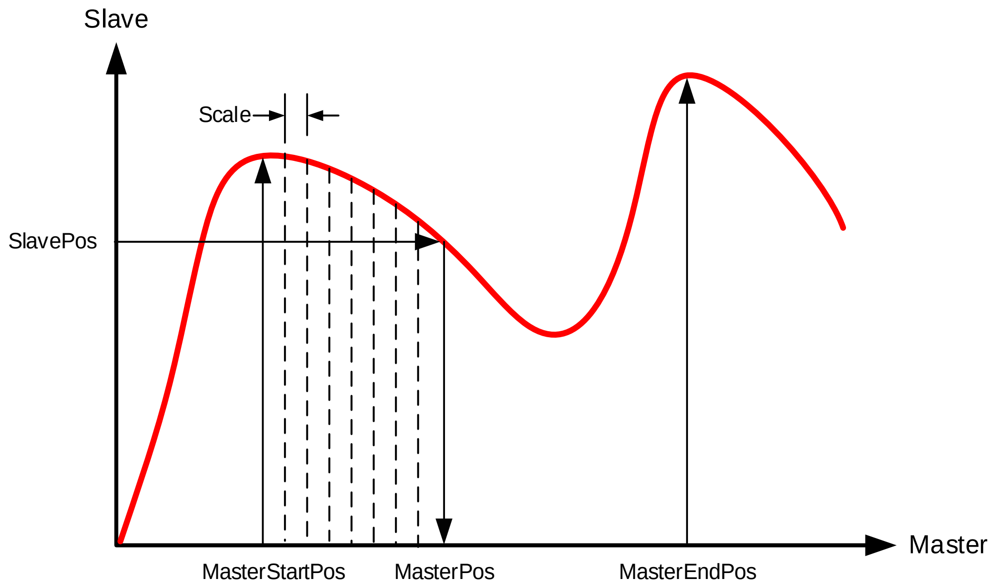
6.6.30 Read CAM table master position (LS_ReadCamTableMasterPos)

Motion Function Block



Input-Output		
UINT	Axis	Set the command axis. (refer toSee 6.2.1 Setting Range by Product)
Input		
BOOL	Execute	Give cam table master position reading command to the relevant axis in the rising Edge.
UINT	CamTableID	Specify the cam table to read. (1~32)
LREAL	MasterStartPos	Start position to read position of cam main axis
LREAL	MasterEndPos	End position to read position of cam main axis
LREAL	SlavePos	Position of cam serve axis
LREAL	Scale	Accuracy of main axis position reading
Output		
BOOL	Done	Indicate that the cam table main axis reading is successfully completed.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
LREAL	MasterPos	It outputs the position of the slave.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block outputs the position of the main axis corresponding to the position of the serve axis set in SlavePos, among the values between MasterStartPos and MasterEndPos in the specified cam table.



(2) Set the position of serve axis to read in the cam table as SlavePos value. Offset/Gear ratio/Phase correction operation applied to the command axis is not reflected in the MasterPos output.

(3) When the cam table master position reading operation is completed, the Done output turns on.

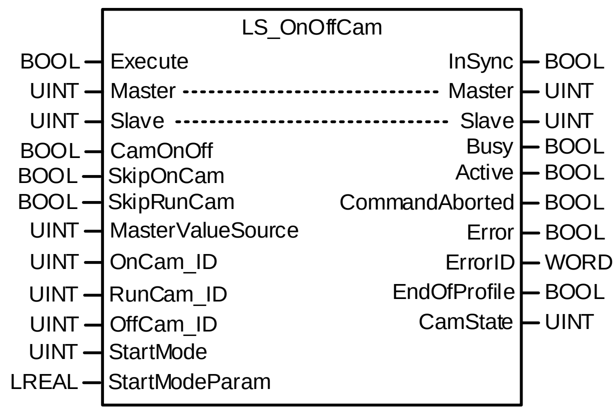
(4) The ‘Scale’, which is the accuracy value of the cam table master position reading, can’t input 0. If the ‘Scale’ is 0, an error (error number: 0x0B) occurs. If the ‘Scale’ value is large, an error may occur between the magnified MasterPos value and the actual spindle position. Also, if the ‘Scale’ value is small, the execution time of the function block may become long.

(5) If the position of the main axis corresponding to the position of the serve axis set in SlavePos does not exist among the values between MasterStartPos and MasterEndPos, Error is On and "0x1124" occurs in ErrorID.

(6) The value of MasterEndPos must be greater than the value of MasterStartPos. If the MasterEndPos value is less than or equal to MasterStartPos, Error is On and "0x0B" occurs in ErrorID.

6.6.31 OnOff CAM Operation (LS_OnOffCam)

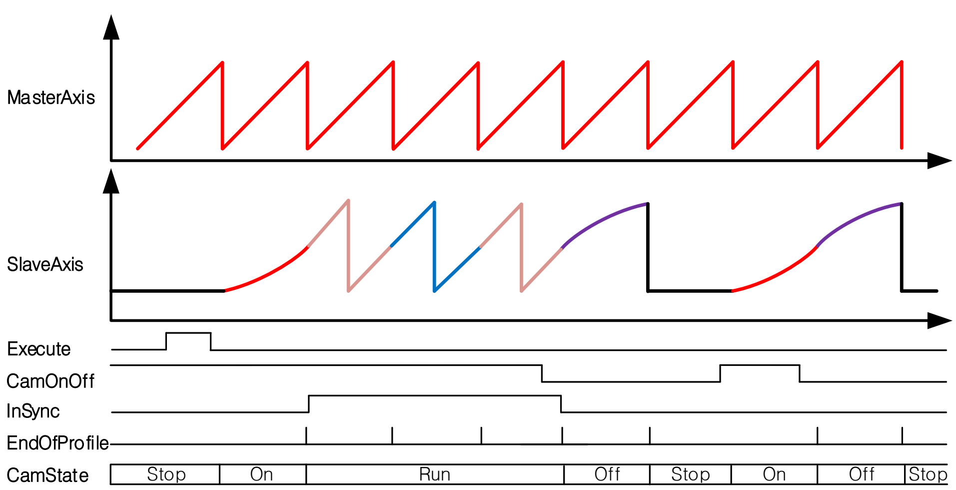
Motion Function Block



Input-Output		
UINT	Master	Set main axis (refer toSee 6.2.1 Setting Range by Product)
UINT	Slave	Set the sub axis. (See 6.2.1 Setting Range by Product), Available to set variables only
Input		
BOOL	Execute	Give the OnOff cam operation command to the relevant axis on the rising Edge.
BOOL	CamOnOff	Set whether the Cam operation is On or Off. 1: Complete OnCam and switch to RunCam. 0: Complete OffCam in RunCam and switch the cam to the stop status
BOOL	SkipOnCam	Exclude OnCam from OnOff cam operation and carry out RunCam->OffCam in order.
BOOL	SkipRunCam	Exclude RunCam from OnOff cam operation and carry out OnCam->OffCam in order.
UINT	MasterValueSource	Select the source of the main axis for cam operation. 0: Synchronizes to the command position of the main axis. 1: Synchronizes to the current position of the main axis.
UINT	OnCam_ID	Specify the cam table to operate in the OnCam state. (refer toSee 6.2.1 Setting Range by Product)
UINT	RunCam_ID	Specify the cam table to operate in the RunCam state. (refer toSee 6.2.1 Setting Range by Product)
UINT	OffCam_ID	Specify the cam table to operate in the OffCam state. (refer toSee 6.2.1 Setting Range by Product)
UINT	StartMode	Select the method to start the cam operation. 0: Start when CamOnOff is set to 1. 1: Start when CamOnOff is set to 1 and the main axis reaches the position set in StartModeParam. 2: Start when CamOnOff is set to 1 and the main axis moves the distance set in StartModeParam. 3: Use the profile generated with LS_CrossSealCamGen.
LREAL	StartModeParam	Set the parameter according to the method for starting the cam operation.

Output		
BOOL	InSync	Indicates that cam operation has entered the RunCam state.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate whether the current motion function block is controlling the relevant axis.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted by other command.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.
BOOL	EndOfProfile	Indicates the end of the current cam operation.
UINT	CamState	0: Stop status 1:OnCam operating 2:RunCam operating 3:OffCam operating

(1) This motion function block uses three cam tables to carry out the cam operation that is switched to Stop state->OnCam->RunCam or RunCam->OffCam->Stop state depending on the CamOnOff input.



(2) The cam operation runs under a state where Execute is the rising Edge. The cam operation does not stop even if Execute is changed to Off during the operation. To stop the OnOffCam operation, you must give the MC_CamOut command or run another motion function block.

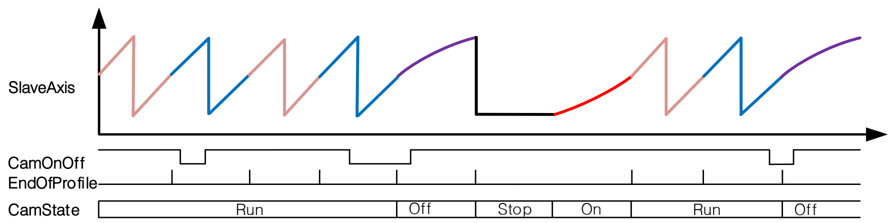
(3) If StartMode is set to 0, OnCam runs immediately when 1 is input in CamOnOff. If StartMode is set to 1, OnCam does not run immediately when 1 is input in CamOnOff. But OnCam runs when the position of the main axis passes by the position set in StartModeParam. If StartMode is set to 2, OnCam runs when 1 is input in CamOnOff and then the main axis moves the distance set in StartModeParam.

(4) If you are using a cam generated with the LS_CrossSealCamGen function block, set StartMode to 3. If StartMode is set to 3 and the length of OnCam_ID is 270, the same operation is conducted as if StartMode is set to 1 and StartModeParam is 270. If OnCam_ID is 180, the same operation is conducted as if StartMode is set to 1 and StartModeParam is set to 0.

(5) The EndOfProfile signal outputs On when passing the end of a profile during operation of each OnCam/OffCam/RunCam cam profile.

(6) If the CamOnOff signal is Off, the operation is performed to switch to RunCam->OffCam->Stop state. If the CamOnOff signal is switched from Off to On in the RunCam state, the RunCam state is maintained if OffCam is not yet executed. In a state

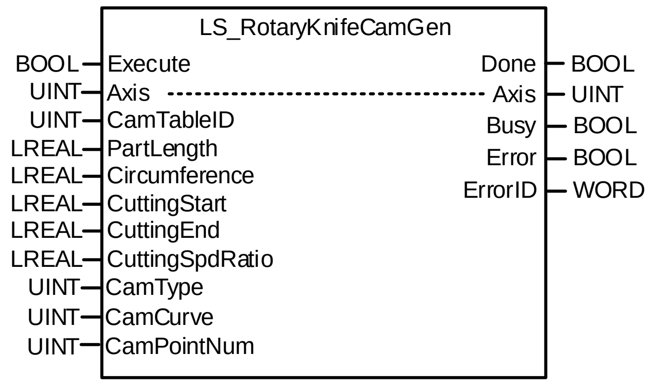
where OffCam is executed, the state switches to the OnCam->RunCam state again after switching to the OffCam->Stop state. (When turning off CamOnOff in RunCam, the operation must be maintained until the EndOfProfile signal is generated.)



(7) If the SkipOnCam signal is On, RunCam is executed instantly without OnCam. If CamOnOff turns off after executing RunCam, perform the operation to switch to RunCam->OffCam->Stop state. In an operation where the SkipOnCam signal is On, the operation is executed from the middle of RunCam.
(8) If the SkipRunnCam signal is On, OffCam is executed without RunCam being executed after OnCam is executed. If CamOnOff is On at this time, the operation repeats in the order of OnCam->OffCam->Stop->OnCam->OffCam->Stop.
(9) To stop the OnOffCam operation completely, use the halt (MC_Halt) or immediate stop (MC_Stop) motion function block.
(10) The CamState value is output as Stop(0) / OnCam(1) / RunCam(2) / OffCam(3) depending on the state of cam operation.
(11) Once the cam operation set in RunCam_ID is executed, InSync output turns On.
(12) MasterValueSource selects the source of the main axis to be synchronized. If set to 0, the serve axis performs cam operations based on the command position of the main axis calculated in the motion controller, and if set to 1, the serve axis performs cam operations based on the current position received via communication from the servo drive of the main axis.
(13) RunCam_ID sets the cam profile to execute during the operation of OnOffCam. The cam profile to execute during operation of OnOffCam is set to RunCam_ID. Before executing RunCam in the Stop state, the cam profile to run is set to OnCam_ID. OffCam_ID sets the cam profile to execute before RunCam reaches the Stop state. The setting range for each ID is 1-32, and an input value outside of the range causes a "0x1115" error in the motion function block.
(14) Any changes made to the MasterValueSource/OnCam_ID/RunCam_ID/OffCam_ID value during operation are not reflected.
(15) The value of the major axis can be changed during OnCam/RunCam/OffCam operation.(V1.50 or higher)
(16) The corresponding axis is in a "SynchronizedMotion" state when this motion function block is running.
(17) For more information, see Chapter 8.6 RotaryKnife Operation under Chapter 8 Motion Control Function.

6.6.32 RotaryKnife cam profile generation (LS_RotaryKnifeCamGen)

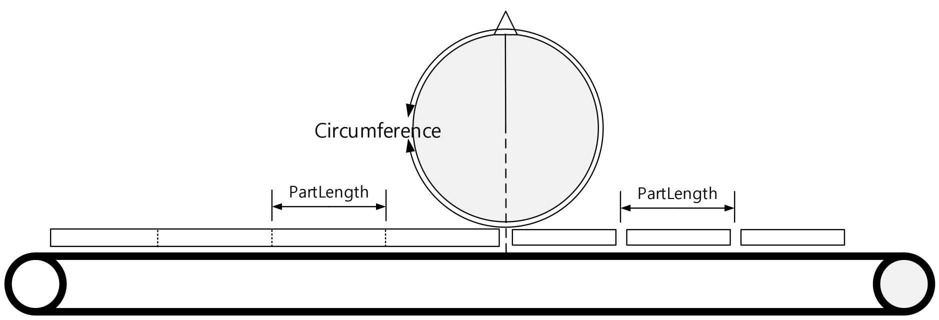
Function Block type



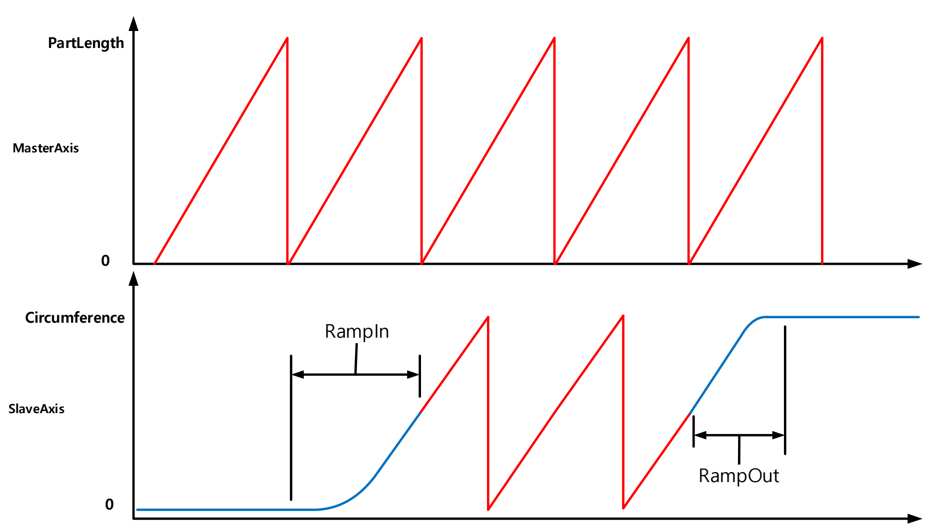
Input-Output		
UINT	Axis	Specify the axis to be commanded (See 6.2.1 Setting Range by Product)
Input		
BOOL	Execute	Performs cam profile generation in the rising Edge.
UINT	CamTableID	Set the cam table ID where the profile is stored. (refer toSee 6.2.1 Setting Range by Product)
LREAL	PartLength	Set the length of the object to cut by the RotaryKnife.
LREAL	Circumference	Set the circumference of the RotaryKnife.
LREAL	CuttingStart	Set the position for the RotaryKnife to start cutting.
LREAL	CutingEnd	Set the position for the RotaryKnife to end cutting.
LREAL	CuttingSpdRatio	Adjust the synchronization speed by a percentage while the RotaryKnife is cutting. (If 100 is entered, the cutting speed is synchronized 1:1 with the main axis.)
UINT	CamType	Set the type of the cam profile to generate. (0:ALL 1:RampIn 2:Running 3:RampOut) (4:sALL 5:sRampIn 6:Running 7:sRampOut)
UINT	CamCurve	Set the cam curve type used in cam profile generation. (0:Linear 1:Cubic)
UINT	CamPointNum	Set the number of cam points used for the cam profile.
Output		
BOOL	Done	Indicates that the cam profile is generated successfully.
BOOL	Busy	Indicates that the execution of the motion function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Outputs the error ID occurred while the motion function block is running.

(1) This motion function block generates the cam profile which performs the RotaryKnife action.

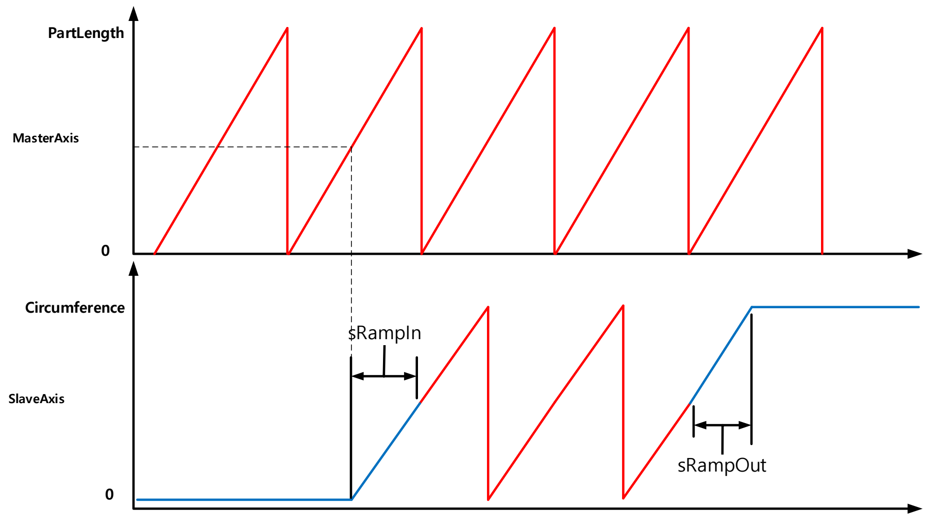
(2) Use the cam profile generated through LS_RotaryKnifeCamGen in the LS_OnOffCam function block.
(3) On the PartLength input, enter the length of the object to perform cutting using the RotaryKnife.
(4) On the Circumference input, enter the circumference of the RotaryKnife.



(5) On the CuttingStart input, enter the starting position for the RotaryKnife to start cutting. On the CuttingStart input, enter the ending position for the RotaryKnife to end cutting. The speed of the conveyor and the RotaryKnife are synchronized between CuttingStart and CuttingEnd. (If you want a cutting region of 10 when the Circumference is 360, set CuttingStart to 175 and CuttingEnd to 185.)
(6) On the generated cam profile, the movement amount of the main axis is 360Degree in ratio to PartLength. This means that you must set the gear ratio of the motor and the machine in the parameter so that 1 rotation of the main axis equals PartLength.
(7) On the generated cam profile, the movement amount of the serve axis is 360Degree in ratio to the Circumference. This means that you must set the gear ratio of the motor and the machine in the parameter so that 1 rotation of the serve axis equals the Circumference.
(8) For CuttingStart, you cannot enter a value that is less than 1/8 of the Circumference or greater than CuttingEnd. A "0x1172" error occurs if there is an error in the CuttingStart value.
(9) For CuttingEnd, you cannot enter a value that is greater than 7/8 of the Circumference or smaller than CuttingEnd. A "0x1172" error occurs if there is an error in the CuttingEnd value. To set the cutting region to the minimum, set CuttingEnd and CuttingStart as equal values.
(10) On the CamType, enter the type of cam profile to generate. Available values are 1: RampIn 2:Running 3:RampOut 5:sRampIn 6:Running 7:sRampOut. If you enter 0, RampIn/Running/RampOut will be generated at once. The Running type generates a cam profile which performs repeated cutting actions. The RampIn type generates a profile that includes the stop state to the action of the Running cam profile performing the cutting action. The RampOut type generates a profile to switch RotaryKnife from a running state to a stop state. A "0x1176" error occurs if the CamType value is outside of the range.



(11) The sRampIn and sRampOut types generate a shortened cam profile of RampIn and RampOut respectively. When operating using sRampIn and sRampOut and you want to main axis to reach the 1/2Circumference position of the serve axis, the main axis must start at the 1/2 position of PartLength.

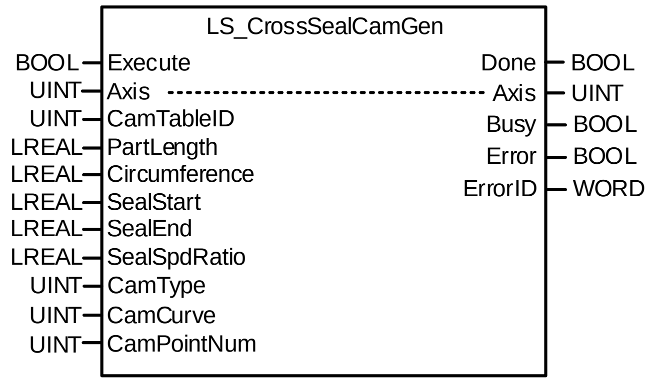


(12) On the CuttingSpdRatio input, set the speed ratio for the cutting region. If CuttingSpdRatio is set to 100, a cam profile is generated which operates by synchronizing 1:1 with the speed of the main axis in the cutting region. As the CuttingSpdRatio value is higher, the faster the synchronization speed on the cutting region. The setting range of CuttingSpdRatio is 50-200 and a "0x1174" error occurs if there is an error in the CuttingSpdRatio value.

(13) On the CamCurve, enter the curve of the cam profile to generate. If you enter 0: Linear, a cam profile is generated using linear interpolation. Once you select linear interpolation, you must specify the number of cam profile points to generate by setting CamPointNum. Take care when setting the number of points as too little can lead to a shock due to the acceleration or deceleration of cam operation and too many can lead to an overload in the program due to the amount of computing resources for saving cam profiles. If you enter 1: Cubic, a cam profile is generated that uses cubic interpolation. A "0x1176" error occurs if the CamCurve value is outside of the range.

(14) The minimum number of cam points that can be set in CamPointNum is 10, and the maximum number of cam points is 1024. Error 0x1177 occurs when CamPointNum has an error.

6.6.33 Cross sealer cam profile generation (LS_CrossSealCamGen)

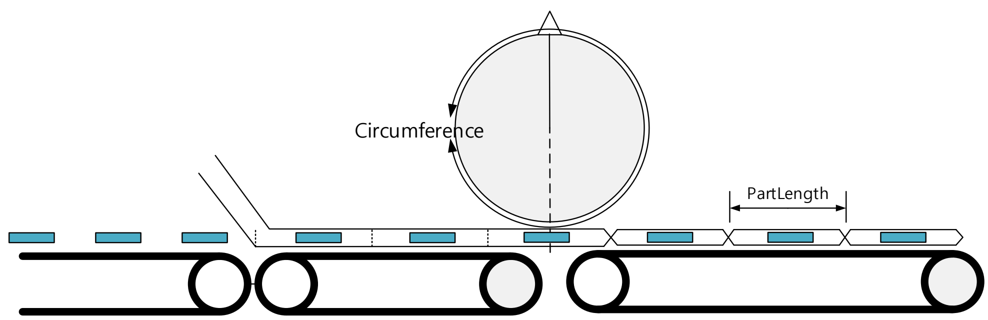
Function Block type		
		
Input-Output		
UINT	Axis	Specify the axis to be commanded (See 6.2.1 Setting Range by Product)
Input		
BOOL	Execute	Performs cam profile generation in the rising Edge.
UINT	CamTableID	Set the cam table ID to store the cam profile. (refer toSee 6.2.1 Setting Range by Product)
LREAL	PartLength	Set length of the object sealed by the cross sealer.
LREAL	Circumference	Set circumference of the cross sealer.
LREAL	SealStart	Set the position for the cross sealer to start sealing.
LREAL	SealEnd	Set the position for the cross sealer to end sealing.
LREAL	SealSpdRatio	Adjust the synchronization speed in percentage while the cross sealer is sealing. (If 100 is entered, the sealing speed is synchronized 1:1 with the main axis.)
UINT	CamType	Set the type of the cam profile to generate. (0:ALL 1:RampIn 2:Running 3:RampOut) (4:sALL 5:sRampIn 6:Running 7:sRampOut)
UINT	CamCurve	Set the cam curve type used in cam profile generation. (0:Linear 1:Cubic)
UINT	CamPointNum	Set the number of cam points used for the cam profile.
Output		
BOOL	Done	Indicates that the cam profile is generated successfully.
BOOL	Busy	Indicates that the execution of the motion function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Outputs the error ID occurred while the motion function block is running.

(1) This motion function block generates the cam profile which performs the cross sealer action. Use the cam profile generated through LS_CrossSealCamGen in the LS_OnOffCam function block.

(2) On the PartLength input, enter the length of the object to perform sealing using the cross sealer.

(3) On the Circumference input, enter the circumference of cross sealer.

(4) Both the main and serve axes of the generated cam profile is output within the 0-360 range. For the PartLength and Circumference values, you must enter the distance moved by the main axis when the main and serve axes move in 360 value.



(5) On the SealStart input, enter the starting position for the cross sealer to start sealing. On the sealEnd input, enter the position where the cross sealer completes the sealing. The speed of conveyor and the cross sealer are synchronized between SealStart and SealEnd. If you want a sealing region of 10 when the Circumference is 360, set SealStart to 175 and SealEnd to 185.

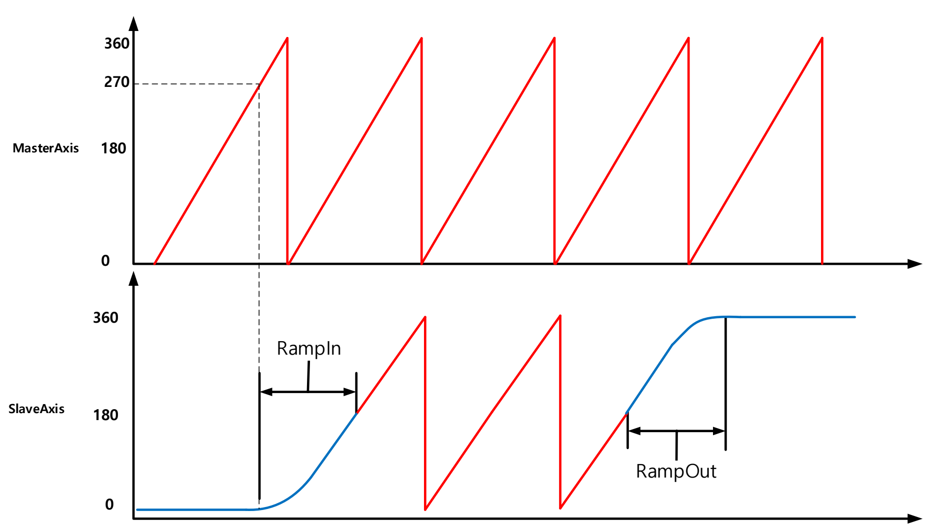
(6) On the generated cam profile, the movement amount of the main axis is 360 in ratio to PartLength. This means that you must set the gear ratio of the motor and the machine in the parameter so that when the main axis moves 360, the real distance equals PartLength.

(7) On the generated cam profile, the movement amount of the serve axis is 360 in ratio to Circumference. This means that you must set the gear ratio of the motor and the machine in the parameter so that when the serve axis moves 360, the real distance equals Circumference.

(8) For SealStart, you cannot enter a value that is less than 1/8 of the Circumference or greater than SealEnd. A "0x1172" error occurs if there is an error in the SealStart value.

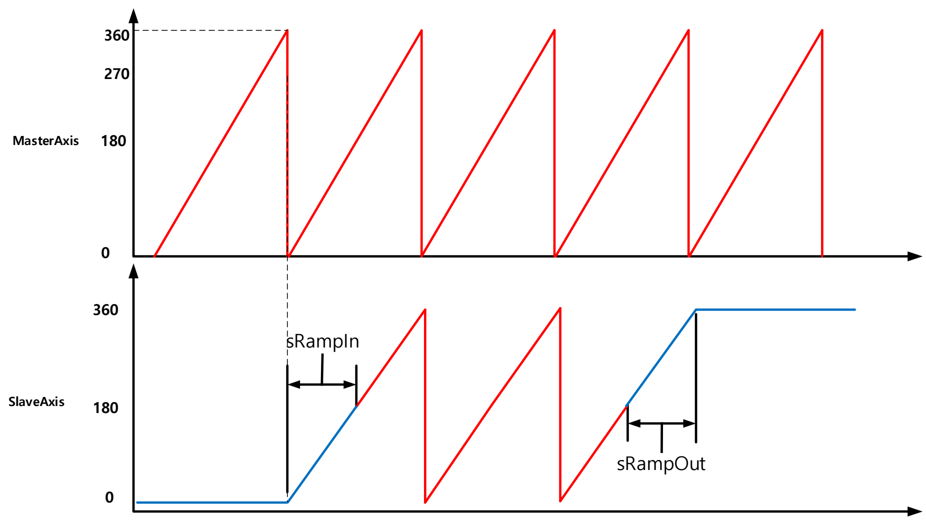
(9) For SealEnd, you cannot enter a value that is greater than 7/8 of the Circumference or smaller than SealEnd. A "0x1172" error occurs if there is an error in the SealEnd value. To set the sealing region to the minimum, set SealEnd and SealStart as equal values.

(10) On the CamType, enter the type of cam profile to generate. Available values are 1: RampIn 2: Running 3:RampOut 5:sRampIn 6:Running 7:sRampOut. If you enter 0, RampIn/Running/RampOut will be generated at once. The Running type generates a cam profile which performs repeated sealing actions. The RampIn type generates a profile that includes the stop state to the action of the Running cam profile performing the sealing action. The RampOut type generates a profile to switch the cross sealer from a running state to a stop state. A "0x1176" error occurs if the CamType value is outside of the range.



(11) The cam profile generated in the LS_CrossSealCamGen function is similar to the cam profile generated in the LS_RotaryCutCamGen. For the RampIn profile, the operation starts when the main axis is at 270 and not at 0. The profile also starts to perform sealing when the main axis is at 180 degrees.

(12) The sRampIn and sRampOut types generate a shortened cam profile of RampIn and RampOut respectively. When operating using sRampIn and sRampOut, the cam operation starts when the main axis is at 0.

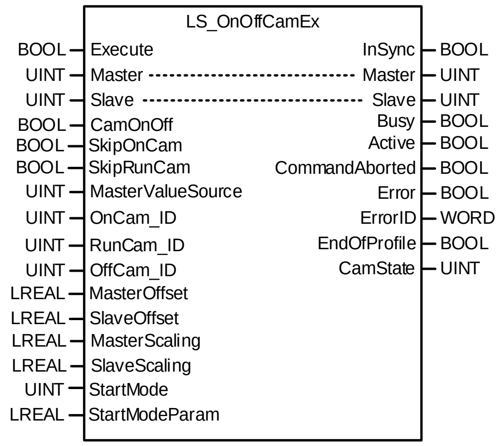


(13) On the SealSpdRatio input, set the speed ratio for the sealing region. If SealSpdRatio is set to 100, a cam profile is generated which operates by synchronizing 1:1 with the speed of the main axis in the sealing section. The higher the SealSpdRatio value, the faster the synchronization speed in the cutting region. The setting range of SealSpdRatio is 50-200 and a "0x1174" error occurs if there is an error in the SealSpdRatio value.

(14) On the CamCurve, enter the curve of the cam profile to generate. 0: Enter linear, a cam profile is generated using linear interpolation. Once you select linear interpolation, you must specify the number of cam profile points to generate by setting CamPointNum. Take care when setting the number of points as too little can lead to a shock due to the acceleration or deceleration of cam operation and too many can lead to an overload in the program due to the amount of computing resources for saving cam profiles. 1: Enter cubic, a cam profile is generated that uses cubic interpolation. A "0x1176" error occurs if the CamCurve value is outside of the range.

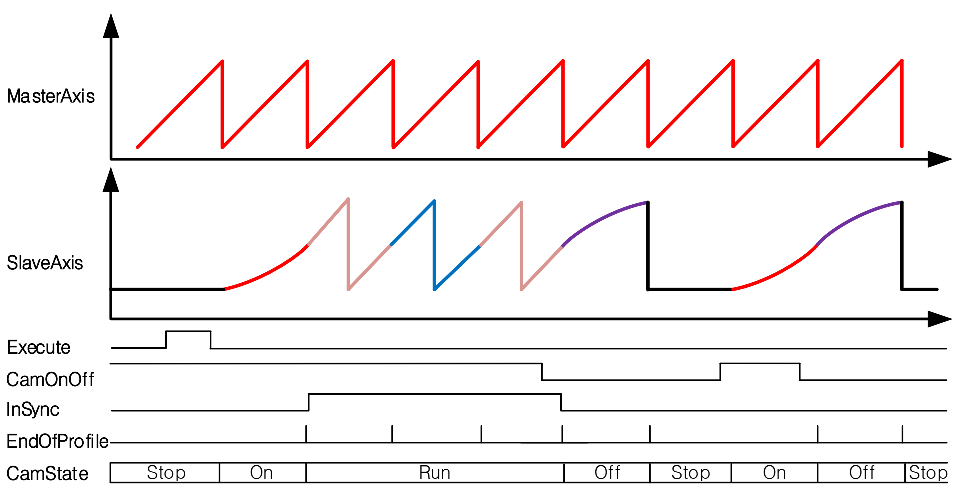
(15) The minimum number of cam points that can be set in CamPointNum is 10, and the maximum number of cam points is 1024. Error 0x1177 occurs when CamPointNum has an error.

6.6.34 Expand OnOff CAM Operation (LS_OnOffCamEx)

Motion Function Block		
		
Input-Output		
UINT	Master	Set main axis (See 6.2.1 Setting Range by Product), Available to set variables only
UINT	Slave	Set the sub axis. (See 6.2.1 Setting Range by Product), Available to set variables only
Input		
BOOL	Execute	Give the OnOff cam operation command to the relevant axis on the rising Edge.
BOOL	CamOnOff	Set whether the Cam operation is On or Off. 1: Complete OnCam and switch to RunCam. 0: Complete OffCam in RunCam and switch the cam to the stop status
BOOL	SkipOnCam	Exclude OnCam from OnOff cam operation and carry out RunCam->OffCam in order.
BOOL	SkipRunCam	Exclude RunCam from OnOff cam operation and carry out OnCam->OffCam in order.
UINT	MasterValueSource	Select the source of the main axis for cam operation. 0: Synchronizes to the command position of the main axis. 1: Synchronizes to the current position of the main axis.
UINT	OnCam_ID	Specify the cam table to operate in the OnCam state.
UINT	RunCam_ID	Specify the cam table to operate in the RunCam state.
UINT	OffCam_ID	Specify the cam table to operate in the OffCam state.
LREAL	MasterOffset	Set the offset value of the main axis.
LREAL	SlaveOffset	Set the offset value of the serve axis.
LREAL	MasterScaling	Specify the magnification of the main axis.
LREAL	SlaveScaling	Specify the magnification of the serve axis.
UINT	StartMode	Select the method to start the cam operation. 0: Start when CamOnOff is set to 1. 1: Start when CamOnOff is set to 1 and the main axis reaches the position set in StartModeParam. 2: Start when CamOnOff is set to 1 and the main axis moves the distance set in StartModeParam. 3: Use the profile generated with LS_CrossSealCamGen.
LREAL	StartModeParam	Set the parameter according to the method for starting the cam operation.

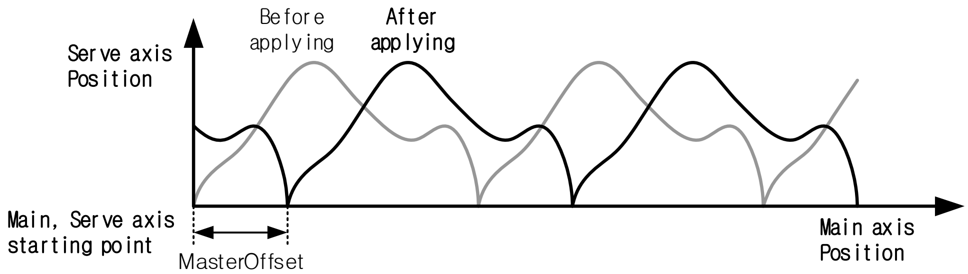
Output		
BOOL	InSync	Indicates that cam operation has entered the RunCam state.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate whether the current motion function block is controlling the relevant axis.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted by other command.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.
BOOL	EndOfProfile	Indicates the end of the current cam operation.
UINT	CamState	0: Stop status 1:OnCam operating 2:RunCam operating 3:OffCam operating

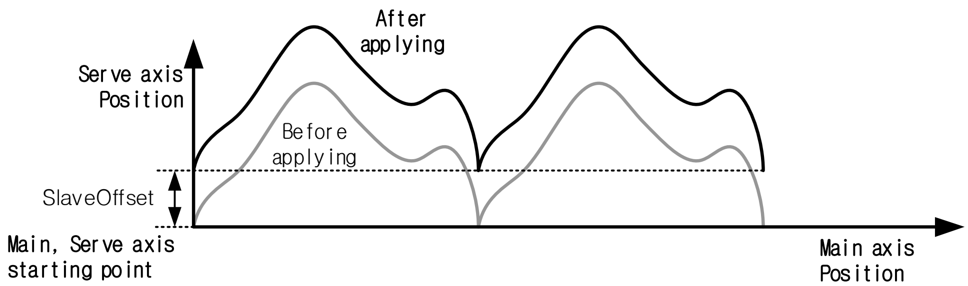
(1) This motion function block uses three cam tables to carry out the cam operation that is switched to stop state->OnCam->RunCam or RunCam->OffCam->Stop state depending on the CamOnOff input.



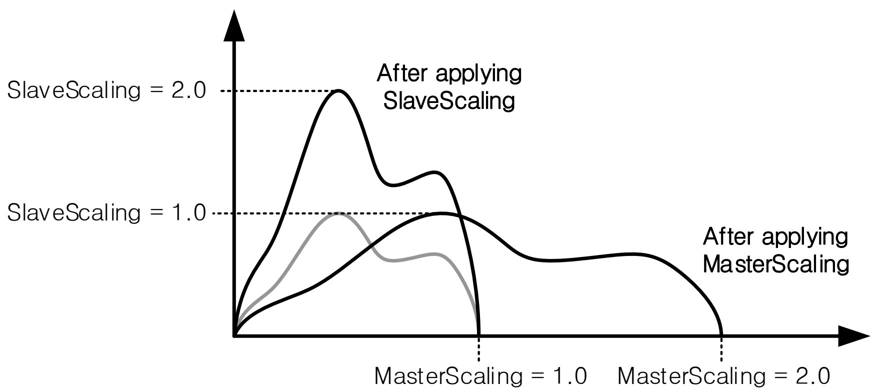
(2) The cam operation runs under a state where Execute is the rising Edge. The cam operation does not stop even if Execute is changed to Off during the operation. To stop the OnOffCam operation, you must give the MC_CamOut command or run another motion function block.

(3) Set the offset of the cam table to be applied in MasterOffset and SlaveOffset. MasterOffset determines the offset from the master axis start point, and Slaveoffset determines the offset from the slave axis start point. Refer to the Figure below.





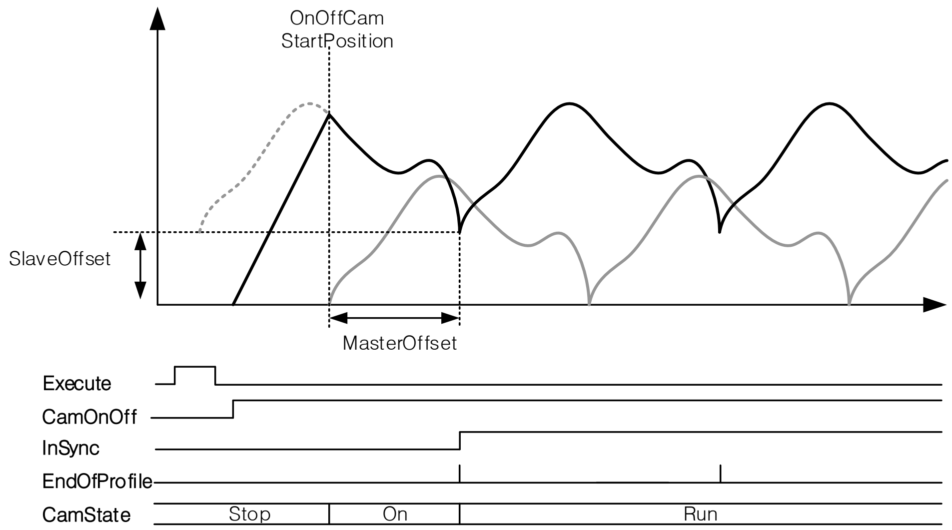
(4) Set the magnification of cam data to be applied in MasterScaling and SlaveScaling. MasterScaling determines the scale rate of the main-axis data, and SlaveScaling determines the scale rate of the sub-axis data. Refer to the Figure below.



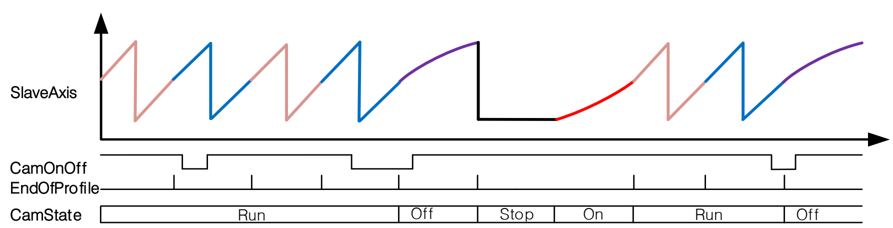
(5) If StartMode is set to 0, OnCam runs immediately when 1 is input in CamOnOff. If StartMode is set to 1, OnCam does not run immediately when 1 is input in CamOnOff. But OnCam runs when the position of the main axis passes by the position set in StartModeParam. If StartMode is set to 2, OnCam runs when 1 is input in CamOnOff and then the main axis moves the distance set in StartModeParam.

(6) If you are using a cam generated with the LS_CrossSealCamGen function block, set StartMode to 3. If StartMode is set to 3 and the length of OnCam_ID is 270, the same operation is conducted as if StartMode is set to 1 and StartModeParam is 270. If OnCam_ID is 180, the same operation is conducted as if StartMode is set to 1 and StartModeParam is set to 0.

(7) When MasterOffset/SlaveOffset is set, if 1 is input in CamOnOff, operation starts at the OnOffCam start position set in StartMode and StartModeParam. The OnOffCam operation is conducted when reaching the OnOffCam start position. If MasterOffset/SlaveOffset is set, StartMode is set to 0 and the OnOffCam operation is conducted, impact can occur when starting operation.



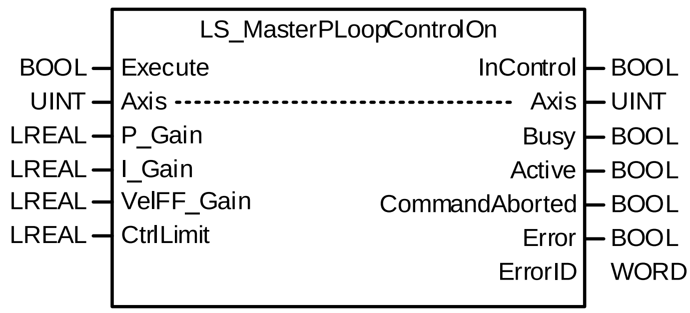
(8) The EndOfProfile signal outputs On when passing the end of a profile during operation of each OnCam/OffCam/RunCam cam profile.
(9) If the CamOnOff signal is Off, the operation is performed to switch to RunCam->OffCam->Stop state. If the CamOnOff signal is switched from Off to On in the RunCam state, the RunCam state is maintained if OffCam is not yet executed. In a state where OffCam is executed, the state switches to the OnCam->RunCam state again after switching to the OffCam->Stop state. (When turning off CamOnOff in RunCam, the operation must be maintained until the EndOfProfile signal is generated.)



(10) If the SkipOnCam signal is On, RunCam is executed instantly without OnCam. If CamOnOff turns off after executing RunCam, perform the operation to switch to RunCam->OffCam->Stop state. In an operation where the SkipOnCam signal is On, the operation is executed from the middle of RunCam.
(11) If the SkipRunnCam signal is On, OffCam is executed without RunCam being executed after OnCam is executed. If CamOnOff is On at this time, the operation repeats in the order of OnCam->OffCam->Stop->OnCam->OffCam->Stop.
(12) To stop the OnOffCam operation completely, use the halt (MC_Halt) or immediate stop (MC_Stop) motion function block.
(13) The CamState value is output as Stop(0) / OnCam(1) / RunCam(2) / OffCam(3) depending on the state of cam operation.
(14) Once the cam operation set in RunCam_ID is executed, InSync output turns On.
(15) MasterValueSource selects the source of the main axis to be synchronized. If set to 0, the serve axis performs cam operations based on the command position of the main axis calculated in the motion controller, and if set to 1, the serve axis performs cam operations based on the current position received via communication from the servo drive of the main axis.
(16) RunCam_ID sets the cam profile to execute during the operation of OnOffCam. The cam profile to execute during operation of OnOffCam is set to RunCam_ID. Before executing RunCam in the Stop state, the cam profile to run is set to OnCam_ID. OffCam_ID sets the cam profile to execute before RunCam reaches the Stop state. The setting range for each ID is 1-32, and an input value outside of the range causes a "0x1115" error in the motion function block.
(17) Any changes made to the MasterValueSource/OnCam_ID/RunCam_ID/OffCam_ID value during operation are not reflected.
(18) The value of the major axis can be changed during OnCam/RunCam/OffCam operation.(V1.5 or higher)
(19) The corresponding axis is in a "SynchronizedMotion" state when this motion function block is running.
(20) For more information, see Chapter 8.6 RotaryKnife Operation under Chapter 8 Motion Control Function.

6.6.35 Master position loop control (LS_MasterPLoopControlOn)

Motion Function Block



Input-Output		
UINT	Axis	Set the command axis. (refer to See 6.2.1 Setting Range by Product)
Input		
BOOL	Execute	Execute master position loop control command to the relevant axis in the rising Edge.
LREAL	P_Gain	Set the master position loop control proportional gain. (1/ 1000 Unit)
LREAL	I_Gain	Set the master position loop control integral gain. (1/ 1000 Unit)
LREAL	VelFF_Gain	Set the master position loop control speed FeedForward gain. (1/ 1000 Unit)
LREAL	CtrlLimit	Set the master position loop control output limit value.
Output		
BOOL	InControl	Indicates that the master position loop control is operating normally.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that machine information setting of the current axis is running.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted while it is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block is used to execute the 'position loop control' in the servo drive connected as a slave in the CPU of the standalone motion controller (XMC).

(2) When the command is executed, the execution mode of the servo drive input to the axis is changed to CSV mode, and the difference (position error) between the command position and the current position of the axis is input to the PI-FF controller operated by the master.

(3) The target speed (0x60FF) object must be added to the RxPDO List, and the CSV operation mode must be supported in the servo drive.

(4) It can be executed on any axis that can be commanded, and after execution, the axis can be driven with an existing motion command.

(5) It can be executed only when the axis is not in operation. If it is executed while the axis is operating, an error "0x1230" occurs. When the gain tuning of the controller is required, it is applied when the gain is modified while the instruction is being executed. Gain correction is possible even when the axis is running.

(6) Only positive numbers can be input for the master position loop control output limit specified in CtrlLimit, and the input value acts as the output upper limit, and the value obtained by multiplying the input value by -1 acts as the output lower limit.

(7) If both the master position loop control output limit and integral gain items are not 0, Anti Wind-up operates for the controller integral operation.

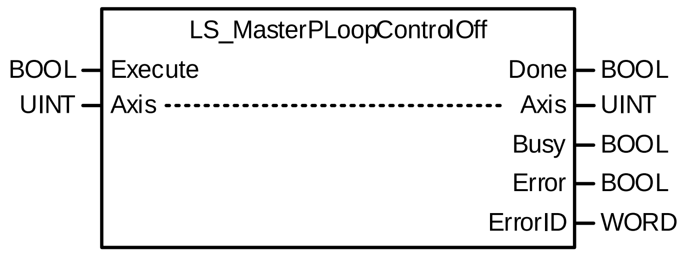
(8) If the gain of the master position loop control is not appropriate, the system may become unstable.

(9) If necessary, the drive's speed controller gain should be tuned. Please refer to the user manual of the drive.

(10) If homing start is executed while master position loop control is in operation, error “0x10B8” occurs. When homing start is required, cancel the master position loop control and execute homing start.

6.6.36 Master position loop control Release (LS_MasterPLoopControlOff)

Motion Function Block

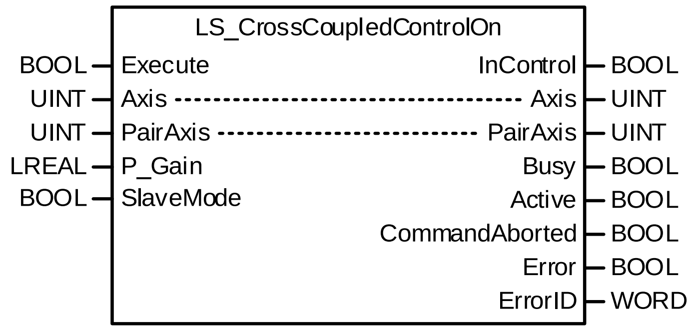


Input-Output		
UINT	Axis	Set the command axis. (refer toSee 6.2.1 Setting Range by Product)
Input		
BOOL	Execute	Execute master position loop control release command to the relevant axis in the rising Edge.
Output		
BOOL	Done	Indicates that master position loop control release has been completed successfully.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block is used to release the position loop control being executed by the CPU of the standalone motion controller (XMC).
(2) If the release command is executed on the axis where the master position loop control is running, the InControl status of the function block where the command was executed is turned off, the CommanAborted output is turned on, and the master position loop control is released.
(3) If the release command is executed on the axis that does not execute the master position loop control, the Done output turns on without any action.
(4) Release command can only be executed when the axis is not in operation.

6.6.37 Cross-coupled control (LS_CrossCoupledControlOn)

Motion Function Block



Input-Output		
UINT	Axis	Set the command axis. (refer toSee 6.2.1 Setting Range by Product)
UINT	PairAxis	Set the connection axis. (refer toSee 6.2.1 Setting Range by Product)
Input		
BOOL	Execute	Execute Cross- coupled control command to the relevant axis in the rising Edge.
LREAL	P_Gain	Set the master position loop control proportional gain. (1/ 1000 Unit)
BOOL	SlaveMode	Select the slave operation mode. (Unused)
Output		
BOOL	Done	Indicates that the Cross-coupled control operating normally.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that machine information setting of the current axis is running.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted while it is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block is used to execute cross-coupled control between two axes under 'position loop control'.

(2) You need to set the Axis and PairAxis different axes where the master position loop controller is running. If you input the same value to Axis and PairAxis and execute it, an error "0x1235" occurs in the function block,If the master position loop controller inputs and executes an axis that is not being executed, an error "0x1236" occurs in the function block.

(3) One cross-coupled control function block can be executed. If the command is re-executed while it is not released, the error "0x1238" occurs in the function block if it is input differently from the Axis and PairAxis set in the previously executed Cross-coupled control function block and executed.

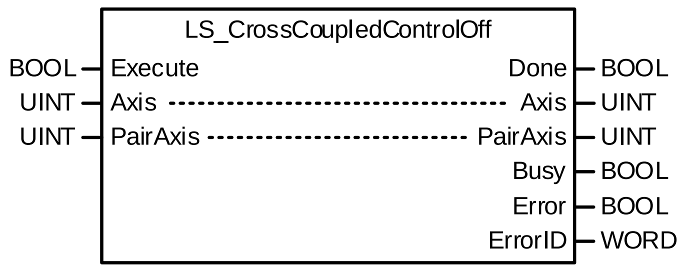
(4) It can be executed only when the axis is not in operation. If it is executed while the axis is operating, an error "0x1234" occurs. When the gain tuning of the controller is required, it is applied when the gain is modified while the instruction is being executed. Gain correction is possible even when the axis is running.

(5) Since the cross-coupled control function reduces the difference in position error (position command value – current position value) that occurs in each axis and the pair axis, the difference between the position command value and the current position value of one axis is caused by an obstacle or an external load. When this occurs, control output is generated as much as the position error of the same size on the other axis. During this process, the accumulated overload of the servo drive may increase and a continuous overload alarm may occur. To avoid this phenomenon, it is necessary to set the maximum torque or torque limit value smaller than the overload detection load factor so that the drive does not become a continuous overload condition. When adjusting overload-related parameters, the drive may be damaged, so pay special attention to safety.

(6) If the gain of cross-coupled control is not appropriate, the system may become unstable.

6.6.38 Cross-coupled control Release (LS_CrossCoupledControlOff)

Motion Function Block

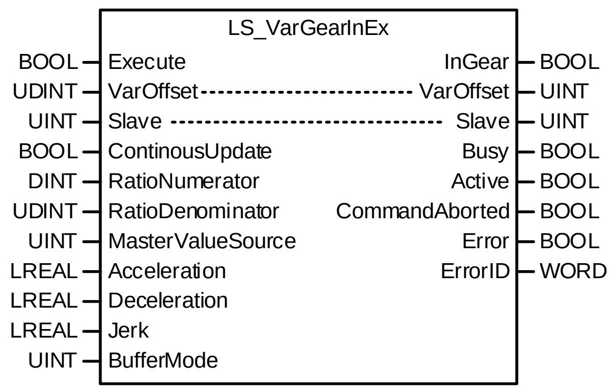


Input-Output		
UINT	Axis	Set the command axis. (refer toSee 6.2.1 Setting Range by Product)
UINT	PairAxis	Set the connection axis. (refer toSee 6.2.1 Setting Range by Product)
Input		
BOOL	Execute	Execute Cross- coupled control release command to the relevant axis in the rising Edge.
Output		
BOOL	Done	Indicates that Cross-coupled control release has been completed successfully.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(7) This motion function block is used to release the Cross-coupled control being executed by the CPU of the standalone motion controller (XMC).
(8) If a release command is executed on two axes where cross-coupled control is running, cross-coupled control is released, the InControl status of the function block where the command was executed is turned off, and the CommandAborted output is turned on.
(9) If the release command is executed on an axis that is not under cross-coupled control, an error “0x1238” occurs in the function block.
(10) It can be executed only when the axis is not in operation. If it is executed while the axis is operating, an error “0x1234” occurs.

6.6.39 Expansion Variable gear operation (LS_VarGearInEx)

Motion Function Block

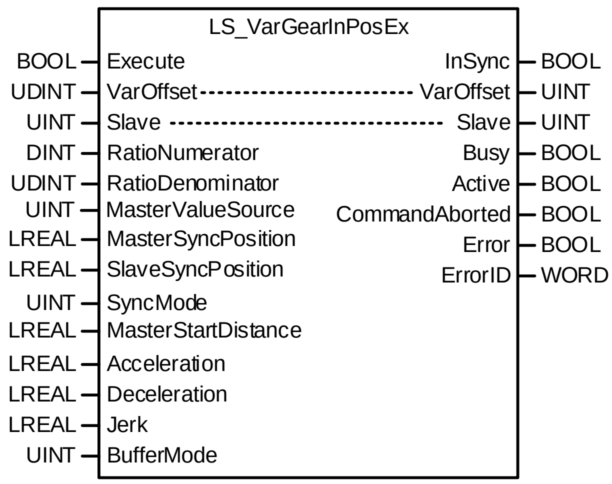


Input-Output		
UDINT	VarOffset	Set the offset value of the M device where the variable to be used as the main axis is located.
UINT	Slave	Set the sub axis. (refer toSee 6.2.1 Setting Range by Product)
Input		
BOOL	Execute	Give gear operation command to the relevant axis in the rising Edge.
BOOL	ContinuousUpdate	Specify the update setting of input value. (Refer to 6.1.5.Changes in Parameters during Execution of Motion Function Block)
DINT	RatioNumerator	Specify the numerator of gear ratio. (-2147483648 ~ 2147483647)
UDINT	RatioDenominator	Specify the denominator of gear ratio. (0 ~ 4294967295)
UINT	MasterValueSource	Select data of the main axis to be synchronized. 0: Synchronize in the command position of the main axis. 1: Synchronize in the current position of the main axis.
LREAL	Acceleration	Specify the acceleration at the beginning of gear operation synchronization. [u/s^2]
LREAL	Deceleration	Specify the deceleration at the beginning of gear operation synchronization. [u/s^2]
LREAL	Jerk	Specify the change rate of acceleration/deceleration. [u/s^3]
UINT	BufferMode	Specify the sequential operation setting of motion function block. (Refer to 6.1.4.BufferMode input)
Output		
BOOL	InGear	Indicate that gear operation is running by applying gear ration.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that the current motion function block is controlling the relevant axis.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted while it is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) In this motion function block, RatioNumerator input and RatioDenominator input are expanded to DINT and UDINT respectively.

(2) The rest of the settings and functions are the same as those of the LS_VarGearIn function block. Please refer to the description of the LS_VarGearIn function block.

6.6.40 Epansion Variable electrical gearing by specifying the position (LS_VarGearInPosEx)

Motion Function Block		
		
Input-Output		
UDINT	VarOffset	Set the offset value of the M device where the variable to be used as the main axis is located.
UINT	Slave	Set the sub axis. (refer toSee 6.2.1 Setting Range by Product)
Input		
BOOL	Execute	Give gear operation command to the relevant axis in the rising Edge.
DINT	RatioNumerator	Specify the numerator of gear ratio. (-2147483648 ~ 2147483647)
UDINT	RatioDenominator	Specify the denominator of gear ratio. (0 ~ 4294967295)
UINT	MasterValueSource	Select the standard of the main axis value to be synchronized. 0(mcSetValue): Synchronize in the target position of the main axis. 1(mcActualValue): Synchronize in the current position of the main axis.
LREAL	MasterSyncPosition	Specify the position of the main axis where gear operation starts.
LREAL	SlaveSyncPosition	Specify the position of the spindle where gear operation starts.
LREAL	SyncMode	Unused
LREAL	MasterStartDistance	Specify the distance of the main axis where synchronization starts.
LREAL	Velocity	Specify the maximum speed of the spindle at the beginning of synchronization. [u/s]
LREAL	Acceleration	Specify the maximum acceleration of the spindle at the beginning of synchronization. [u/s^2]
LREAL	Deceleration	Specify the maximum deceleration of the spindle at the beginning of synchronization. [u/s^2]
LREAL	Jerk	Unused
UINT	BufferMode	Specify the sequential operation setting of motion function block. (Refer to 6.1.4.BufferMode input)
Output		
BOOL	InSync	Indicate that gear operation is normally being fulfilled as the specified gear ratio is applied.
BOOL	StartSync	Indicate synchronization is starting.
BOOL	Busy	Indicate that the execution of motion function block is not completed.

BOOL	Active	Indicate that the current motion function block is controlling the relevant axis.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted while it is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

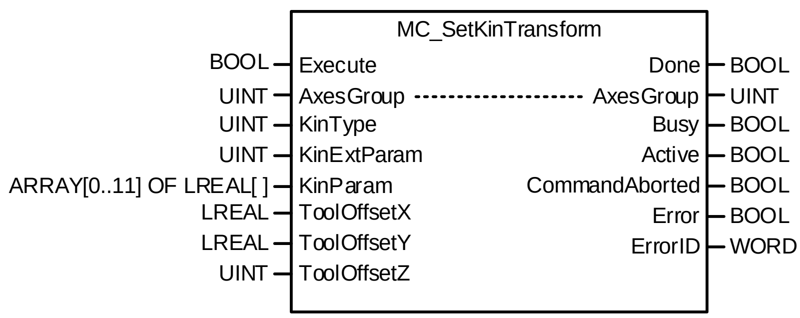
(1) In this motion function block, RatioNumerator input and RatioDenominator input are expanded to DINT and UDINT respectively.

(2) The rest of the settings and functions are the same as those of the LS_VarGearInPos function block. Please refer to the description of the LS_VarGearInPos function block.

6.7 Coordinate System Operation Function Block

6.7.1 Setting the Machine information position (MC_SetKinTransform)

Motion Function Block



Input-Output		
UINT	AxesGroup	Set the axes group to set the machine information. (1 ~ 16: 1Group ~ 16 Group)
Input		
BOOL	Execute	Give the machine information setting command on the axis in the rising Edge.
UINT	KinType	Set the machine type. (1:XYZ, 2:Delta3, 3:Delta3R, 4:LinearDelta3,□5:LinearDelta3R, 6:T-Gantry, 7:T-GantryR)
UINT	KinExtParam	Unused
LREAL[]	KinParam	Set the machine information.
LREAL	ToolOffsetX	Set the X axis offset of at the end of the machine.
LREAL	ToolOffsetY	Set the Y axis offset of at the end of the machine.
LREAL	ToolOffsetZ	Set the Z axis offset of at the end of the machine.
Output		
BOOL	Done	Indicate the machine information setting is successfully completed.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that machine information setting of the current axis is running.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted while it is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block sets the ACS and MCS conversion based on the machine model defined in advance at AxesGroup.
(2) The same setting can be applied to the XG5000 group parameter settings.

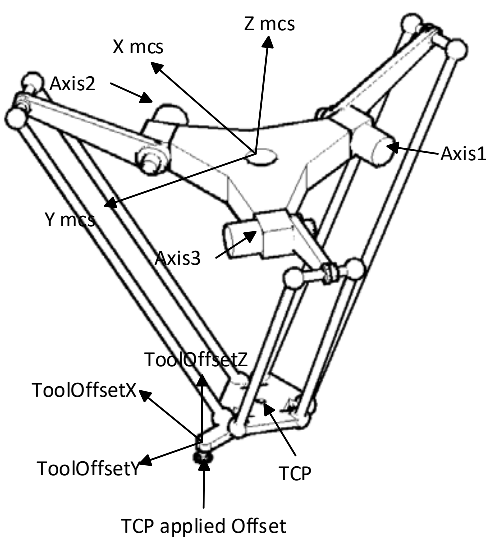
Group	Name	Axis group 1
Coordinate system configuration	Coordinate system Type	0: None
	Coordinate system parameter1	0
	Coordinate system parameter2	0
	Coordinate system parameter3	0
	Coordinate system parameter4	0
	Coordinate system parameter5	0
	Coordinate system parameter6	0
Tool configuration	X-axis offset	0 mm
	Y-axis offset	0 mm
	Z-axis offset	0 mm

(3) The KinType input is used to set the type of the device. You can set the device as shown below.

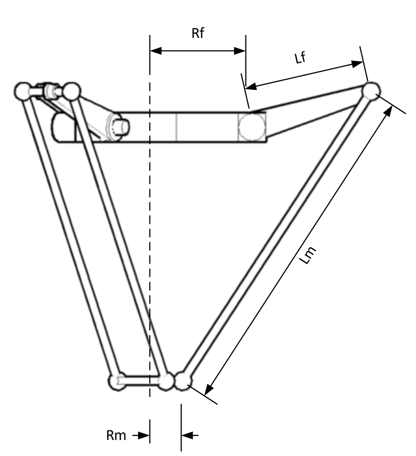
1) 0: none
2) 1: XYZ
3) 2: Delta3
4) 3: Delta3R
5) 4: LinearDelta3
6) 5: LinearDelta3R
7) 6: T-Gantry
8) 7: T-GantryR

(4) KinParam input is used to set the device information. (It is not set for XYZ type.)

(5) ToolOffsetX / ToolOffsetY / ToolOffsedZ are the functions to set the offset at the end point of the device. In order to cope with the case where a separate device is connected to the end of the TCP of the robot, the tool offset function is provided separately from the device information.



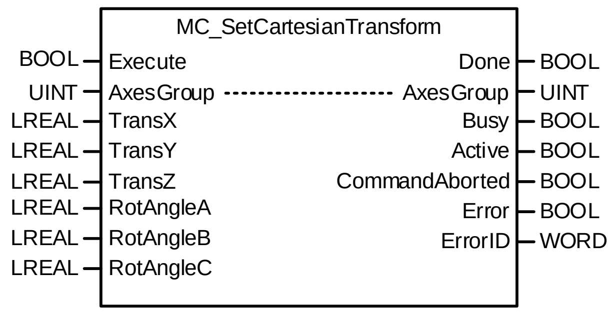
(6) When using Delta3, the device setting information is as follows. For more information, refer to 8.4.4 Machine information setting.



Parameter	Description
KinParam[0]	Lf:Link length of the fixed frame(mm)
KinParam[1]	Lm: Link length of the moving frame(mm)
KinParam[2]	Rf: Length from the center of the fixed frame to the link of the fixed frame (mm)
KinParam[3]	Rm: Length from the center of the moving frame to the link of the moving frame (mm)

6.7.2 PCS setting (MC_SetCartesianTransform)

Motion Function Block



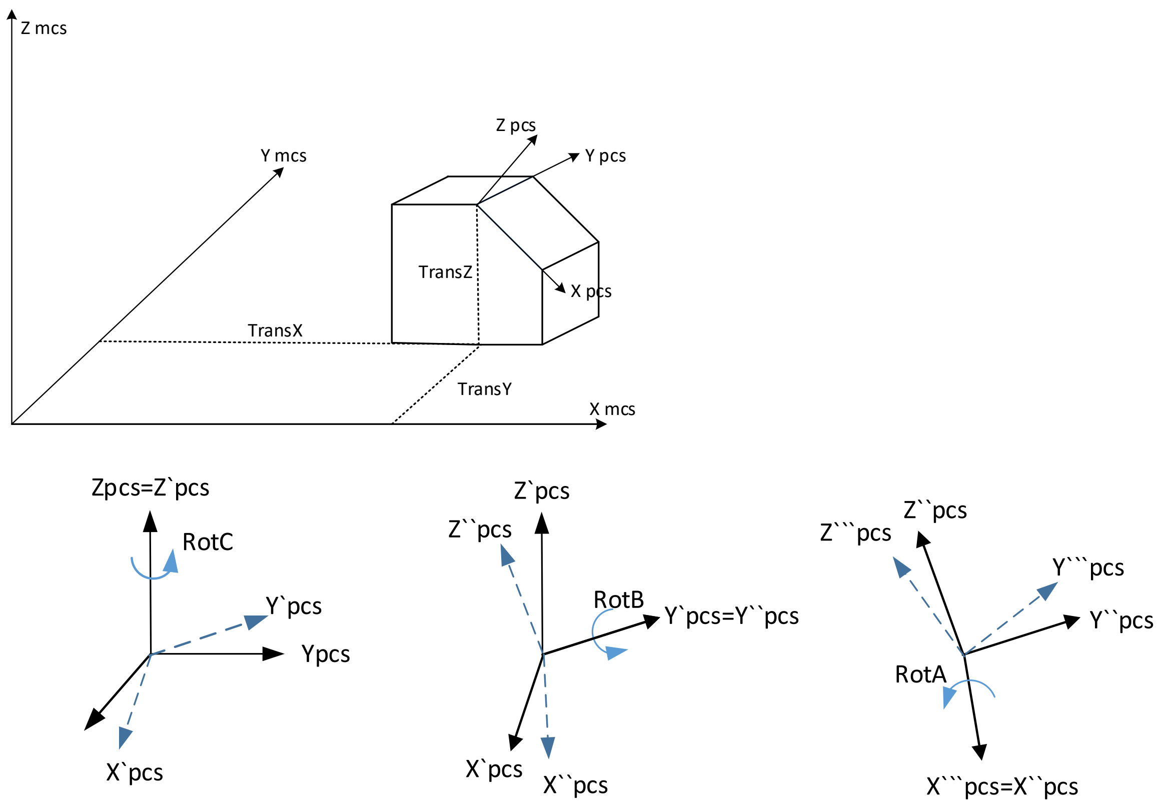
Input-Output		
UINT	AxesGroup	Set the group to set the PCS. (1 ~ 16: 1Group ~ 16 Group)
Input		
BOOL	Execute	Give the PCS setting command on the axes group in the rising Edge.
LREAL	TransX	Movement from MCS to X Axis(mm)
LREAL	TransY	Movement from MCS to Y Axis(mm)
LREAL	TransZ	Movement from MCS to Z Axis(mm)
LREAL	RotAngleA	X Axis rotation amount (degree) (reserved)
LREAL	RotAngleB	Y Axis rotation amount (degree) (reserved)
LREAL	RotAngleC	Z Axis rotation amount (Degree)
Output		
BOOL	Done	Indicate the PCS setting is successfully completed.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that machine information setting of the current axis is running.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted while it is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block sets the perpendicular coordinate conversion between MCS and PCS at AxesGroup.

(2) Axis group setting can be performed in the same way at XG5000 axis group parameter setting.

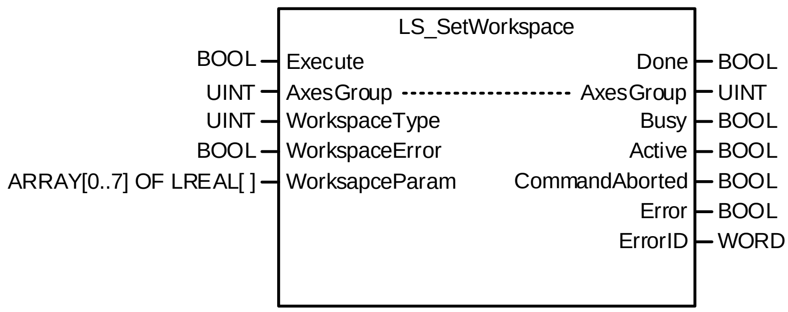
Group	Name	Axis group 1
PCS Configuration	X-axis feed amount	0 mm
	Y-axis feed amount	0 mm
	Z-axis feed amount	0 mm
	X-axis rotation	0 deg
	Y-axis rotation	0 deg
	Z-axis rotation	0 deg

(3) TransX/TransY/TransZ is the move distance from MCS origin to PCS origin. RotA/RotB/RotC is the rotation value of PCS, RotA is the value that rotates PCS on the X-axis of PCS, RotB is the value that rotates PCS on the Y-axis of PCS, RotC is the value that rotates PCS on the Z-axis of PCS. The rotation of PCS must be done the order of RotC, RotB, RotA. Refer to chapter 8.4.3 PCS setting in motion controller's manual for more details.



6.7.3 Work space setting (LS_SetWorkspace)

Motion Function Block



Input-Output		
UINT	AxesGroup	Set the axes group to set the workspace. (1 ~ 16: 1Group ~ 16 Group)
Input		
BOOL	Execute	Give the work space setting command on the axes group in the rising Edge.
UINT	WorkspaceType	Set the work space type (0:Rectangle 1:Cylinder 2:Delta3 3:Sector)
BOOL	WorkspaceError	Set whether an error occurs or not when a coordinate system operation exceeds the work space.
LREAL[]	WorkspaceParam	Set the parameter of the work space.
Output		
BOOL	Done	Indicate the PCS setting is successfully completed.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that machine information setting of the current axis is running.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted while it is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block sets the work space based on the coordinate system at the axes group designated by AxesGroup input.

(2) The same setting can be applied to the XG5000 group parameter settings.

Workspace configuration	Workspace type	0: Rectangle
	Workspace error check	0: Disable
	Workspace Parameter1	170 mm
	Workspace Parameter2	-170 mm
	Workspace Parameter3	170 mm
	Workspace Parameter4	-170 mm
	Workspace Parameter5	-380 mm
	Workspace Parameter6	-580 mm
	Workspace Parameter7	0
	Workspace Parameter8	0

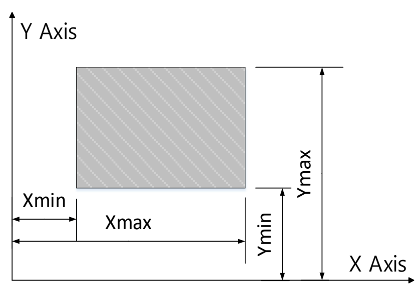
(3) WorkspaceType can be selected from 4 types (0: Rectangle 1: Cylinder 2: Delta3 3: Sector).

(4) WorkspaceError input determines whether an error occurs when a coordinate system operation exceeds the work space.

(5) WorkspaceParam input sets the parameters depending on the work space type.

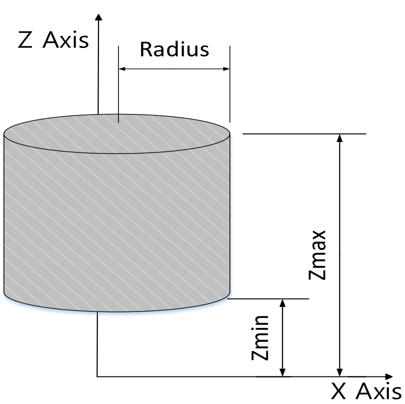
(6) The parameter setting of work space is explained as follows. Refer to chapter 8.4.5 Workspace setting in motion controller's manual for more details.

1) Rectangle



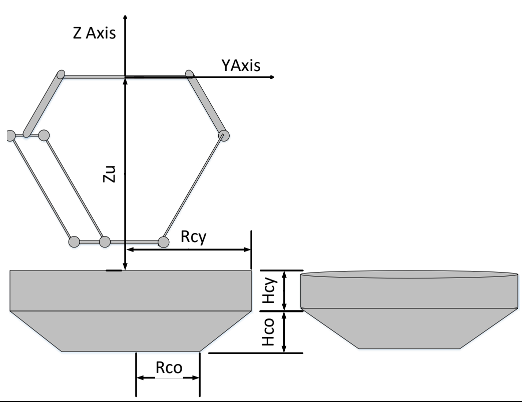
Parameter	Value
WorkspaceParam[0]	X max(mm)
WorkspaceParam[1]	X min(mm)
WorkspaceParam[2]	Y max(mm)
WorkspaceParam[3]	Y min(mm)
WorkspaceParam[4]	Z max(mm)
WorkspaceParam[5]	Z min(mm)

2) Cylinder



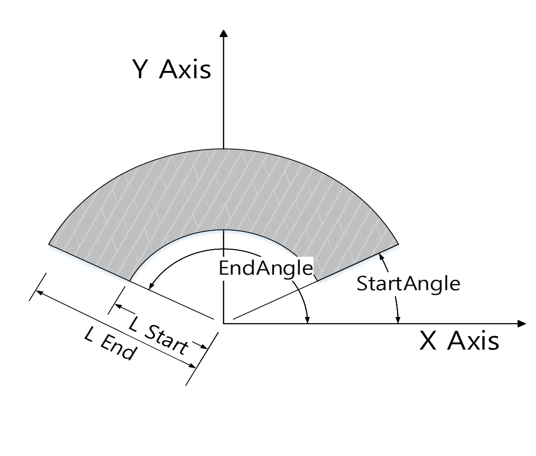
Parameter	Value
WorkspaceParam[0]	Radius(mm)
WorkspaceParam[1]	Z max(mm)
WorkspaceParam[2]	Z min(mm)

3) Delta



Parameter	Value
WorkspaceParam[0]	Zu(mm)
WorkspaceParam[1]	Hcy(mm)
WorkspaceParam[2]	Hco(mm)
WorkspaceParam[3]	Rcy(mm)
WorkspaceParam[4]	Rco(mm)
WorkspaceParam[5]	-

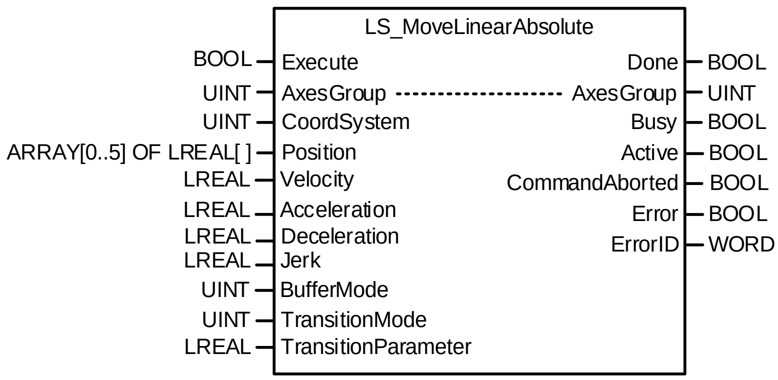
4) Sector



Parameter	Value
WorkspaceParam[0]	L end (mm)
WorkspaceParam[1]	L start(mm)
WorkspaceParam[2]	Z max(mm)
WorkspaceParam[3]	Z min(mm)
WorkspaceParam[4]	EndAngle(degree)
WorkspaceParam[5]	StartAngle(degree)

6.7.4 Linear Interpolation Operation for Absolute Position of Coordinate System (LS_MoveLinearAbsolute)

Motion Function Block



Input-Output		
UINT	AxesGroup	Set the axes group to set the absolute position time linear interpolation. (1 ~ 16: 1Group ~ 16 Group)
Input		
BOOL	Execute	Give the linear interpolation command on the axes group in the rising Edge.
UINT	CoordSystem	Set the coordinate system type (1:MCS 2:PCS)
LREAL[]	Position	Enter the target position of the end point of the machine.
LREAL	Velocity	Specify the maximum speed. [u/s]
LREAL	Acceleration	Specify the maximum acceleration. [u/s^2]
LREAL	Deceleration	Specify the maximum deceleration. [u/s^2]
LREAL	Jerk	Specify the change rate of acceleration/deceleration. [u/s^3]
UINT	BufferMode	Specify the sequential operation setting of motion function block. (Refer to 6.1.4.BufferMode input)
UINT	TransitionMode	Specify the route change mode of group operation. (Refer to 6.1.6.TransitionMode)
LREAL	TransitionParameter	Specify the parameter of the route change setting of group operation. (Refer to 6.1.6.TransitionMode)
Output		
BOOL	Done	Indicates that the coordinate system absolute position linear interpolation operation has been successfully completed.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that machine information setting of the current axis is running.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted while it is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block issues absolute position linear interpolation command based on coordinate system on the axes group designated by AxesGroup input

(2) When this motion function block is executed, interpolation control is performed in a linear trajectory from the machine end point

of each axes group to the target position.

(3) Specify the speed, acceleration, deceleration, and the change rate of acceleration/deceleration of interpolation operation in Velocity, Acceleration, Deceleration, and Jerk inputs respectively.

(4) Velocity is to set the maximum speed of interpolation operation, and the standard travel distance for interpolation operation is the combined distance of the target position position value (Position[0]~Position[2]) and the angle value (Position[3]~Position[5]).) multiplied by the combined distance of the axis group parameter and the angular distance conversion constant of the axis group parameter, the larger of the values is used as the standard travel distance. If the position value of the target position is the same as the position value of the current position and the angular distance conversion constant value of the axis group parameter is 0, the reference travel distance value of the coordinate system operation is used as the combined distance value of the angle value of the target position.

Travel distance

$$= \mathrm{MAX}\left(\begin{array}{c} \mathrm{sqrt}((targer\ positionX - current\ positionX)^2 + (target\ positionY - currnet\ positionY)^2 + (target\ positionZ - current\ positionZ)^2), \\ \text{angular distance conversion constant value} * \mathrm{sqrt}((target\ positionA - current\ positionA)^2 + (target\ positionB - current\ positionB)^2 + (target\ positionC - current\ positionC)^2) \end{array}\right)$$

(5) If the function block is re-executed (Execute input is On) before the instruction is terminated, the changed parameters are applied.

(6) Only Velocity, Acceleration, Deceleration, Jerk, Position input can be updated.

(7) Velocity input can be set to 0 or changed.

(8) Example program

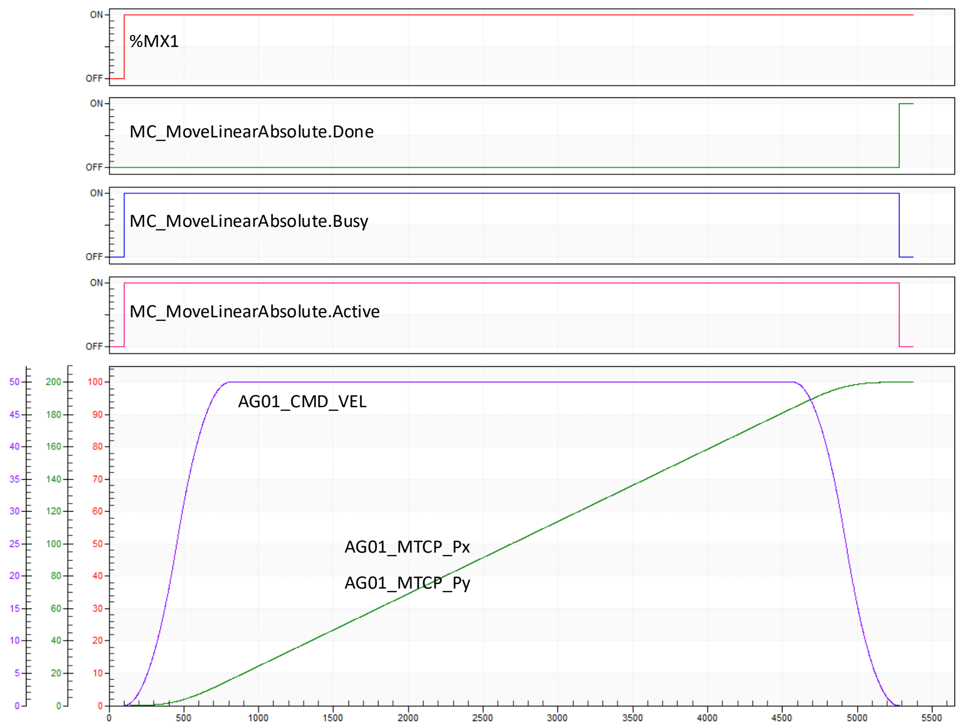
This example shows the linear interpolation to the target position of MCS (100, 200,0) when the current command position is 0,0,0 of MCS coordinate system.

(a) Function block setting

	변수/디바이스	값	타입	디바이스/변수	설명문
1	_AG01_MTCP_Px	0,000000	LREAL	%FL3367	축그룹 01 X축 좌표 (MCS
2	_AG01_MTCP_Py	0,000000	LREAL	%FL3368	축그룹 01 Y축 좌표 (MCS
3	_AG01_MTCP_Pz	0,000000	LREAL	%FL3369	축그룹 01 Z축 좌표 (MCS
4	⊟ LinIntpPosition		ARRAY[0		
5	LinIntpPosition[0]	100,000000	LREAL		
6	LinIntpPosition[1]	200,000000	LREAL		
7	LinIntpPosition[2]	0,000000	LREAL		
8	LinIntpPosition[3]	0,000000	LREAL		
9	LinIntpPosition[4]	0,000000	LREAL		
10	LinIntpPosition[5]	0,000000	LREAL		

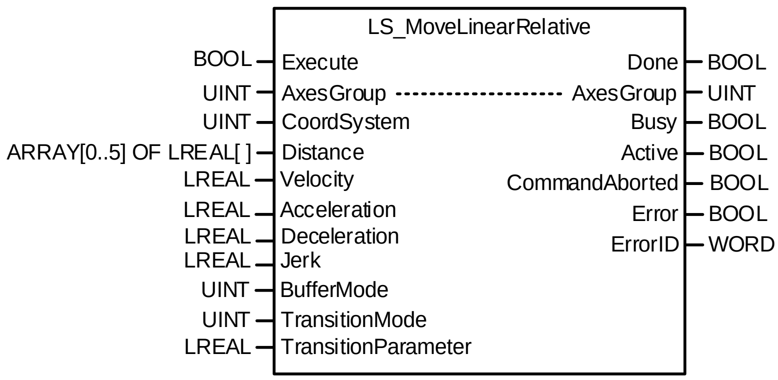
Target position

(b) Timing diagram



6.7.5 Linear Interpolation Operation for Incremental Position of Coordinate System (LS_MoveLinearRelative)

Motion Function Block



Input-Output		
UINT	AxesGroup	Set the axes group to set the Incremental position linear interpolation. (1 ~ 16: 1Group ~ 16 Group)
Input		
BOOL	Execute	Give the incremental position linear interpolation command on the axes group in the rising Edge.
UINT	CoordSystem	Set the coordinate system type (1:MCS 2:PCS)
LREAL[]	Distance	Enter the movement distance of the end point of the machine.
LREAL	Velocity	Specify the maximum speed. [u/s]
LREAL	Acceleration	Specify the maximum acceleration. [u/s^2]
LREAL	Deceleration	Specify the maximum deceleration. [u/s^2]
LREAL	Jerk	Specify the change rate of acceleration/deceleration. [u/s^3]
UINT	BufferMode	Specify the sequential operation setting of motion function block. (Refer to 6.1.4.BufferMode input)
UINT	TransitionMode	Specify the route change mode of group operation. Reference
LREAL	TransitionParameter	Specify the parameter of the route change setting of group operation. (Refer to 6.1.6.TransitionMode)
Output		
BOOL	Done	Indicates that the coordinate system incremental position linear interpolation operation has been successfully completed.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that machine information setting of the current axis is running.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted while it is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block issues Incremental position linear interpolation command based on coordinate system on the axes group designated by AxesGroup input

(2) When this motion function block is executed, interpolation control is performed in a linear trajectory from the machine end point of each axes group to the target position.

(3) Specify the speed, acceleration, deceleration, and the change rate of acceleration/deceleration of interpolation operation in Velocity, Acceleration, Deceleration, and Jerk inputs respectively.

(4) Velocity is to set the maximum speed of interpolation operation, and the standard travel distance for interpolation operation is the combined distance of the target position (Position [0]~Position[2]) and the angle value (Position[3]~Position[5]) The larger value of the product of the composite distance of and the angular distance conversion constant of the axis group parameter is used as the standard travel distance. If the position value of the target position is the same as the position value of the current position and the angular distance conversion constant value of the axis group parameter is 0, the reference travel distance value of the coordinate system operation is used as the combined distance value of the angle value of the target position.

$$\text{Travel distance}$$
$$= \text{MAX}\begin{pmatrix} \text{sqrt}((targer\ positionX - current\ positionX)^2 + (target\ positionY - currnet\ positionY)^2 + (target\ positionZ - current\ positionZ)^2), \\ \text{angular distance conversion constant value} * \text{sqrt}((target\ positionA - current\ positionA)^2 + (target\ positionB - current\ positionB)^2 + (target\ positionC - current\ positionC)^2) \end{pmatrix}$$

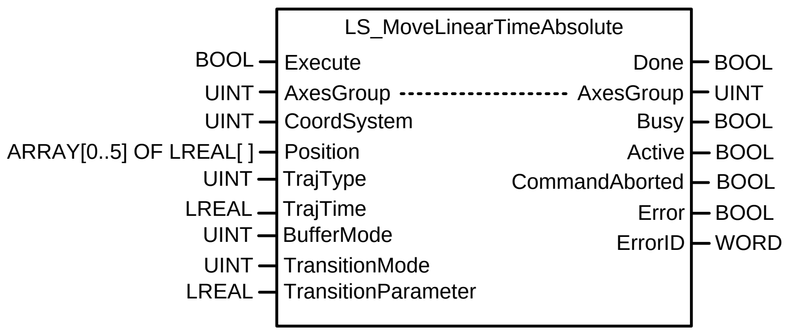
(5) If the function block is re-executed (Execute input is On) before the instruction is terminated, the changed parameters are applied.

(6) Only Distance, Velocity, Acceleration, Deceleration, Jerk, Distance input can be updated.

(7) Velocity input can be set to 0 or changed.

6.7.6 Time Linear Interpolation Operation for Absolute Position of Coordinate System (LS_MoveLinearTimeAbsolute)

Motion Function Block



Input-Output		
UINT	AxesGroup	Set the axes group to set the absolute position time linear interpolation. (1 ~ 16: 1Group ~ 16 Group)
Input		
BOOL	Execute	Give the time linear interpolation command on the axes group in the rising Edge.
UINT	CoordSystem	Set the coordinate system type (1:MCS 2:PCS)
LREAL[]	Position	Enter the target position of the end point of the machine.
UINT	TrajType	Enter the operation acc/dec type.(0:Trapezoid 1:Sine1 2:Sine2)
LREAL	TrajTime	Set the time taken to reach the target position.(msec)
UINT	BufferMode	Specify the sequential operation setting of motion function block. (Refer to 6.1.4.BufferMode input)
UINT	TransitionMode	Specify the route change mode of group operation. (Refer to 6.1.6.TransitionMode)
LREAL	TransitionParameter	Specify the parameter of the route change setting of group operation. (Refer to 6.1.6.TransitionMode)
Output		
BOOL	Done	Indicate the PCS setting is successfully completed.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that machine information setting of the current axis is running.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted while it is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block issues absolute position/time linear interpolation command based on coordinate system on the axes group designated by AxesGroup input

(2) When this motion function block is executed, interpolation control is performed in a linear trajectory from the machine end point of each axes group to the target position.

(3) TrajType input sets the type of velocity, acceleration, deceleration of interpolation trajectory. The type can be selected from three

types: Trapezoid/Sine1/Sine2.

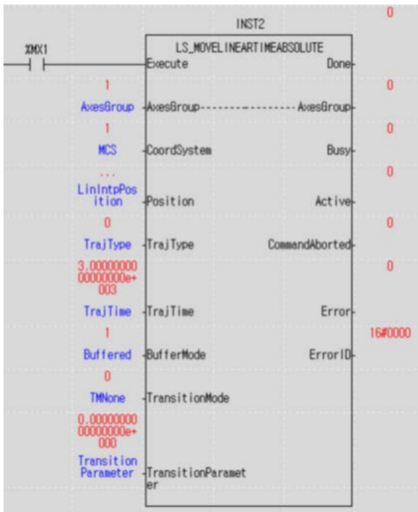
(4) TrajTime sets the time taken to reach the target position.

(5) Please refer to 8. 4. 6 Coordinate System Absolute Position/Time Linear Interpolation Control for further details.

(6) Example program

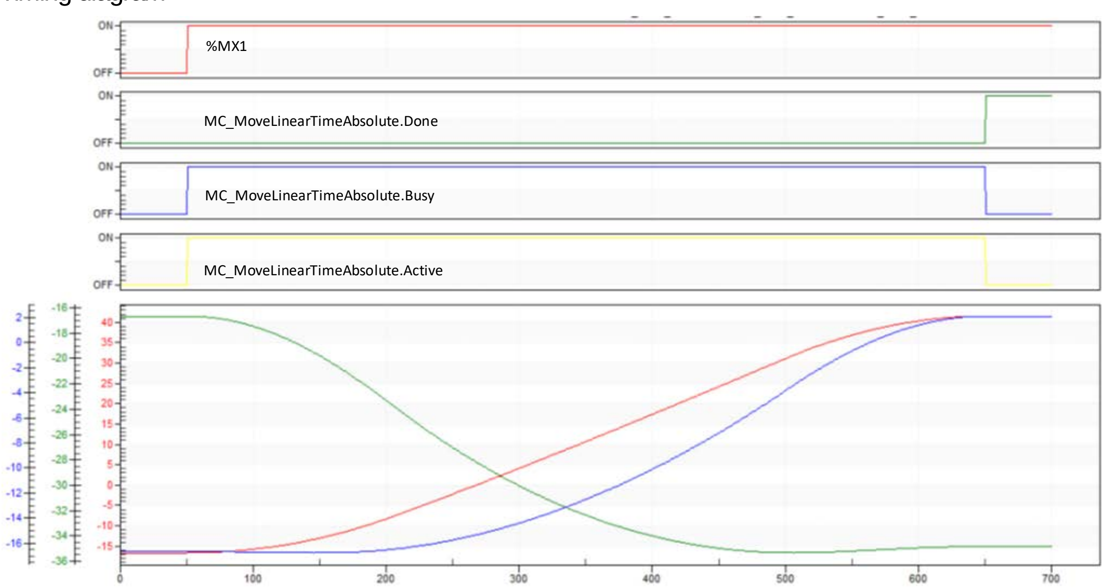
This example shows the linear interpolation to the target position of MCS (100, 200,-380) when the current command position is 0,0,-380 of MCS coordinate system.

(a) Function block setting



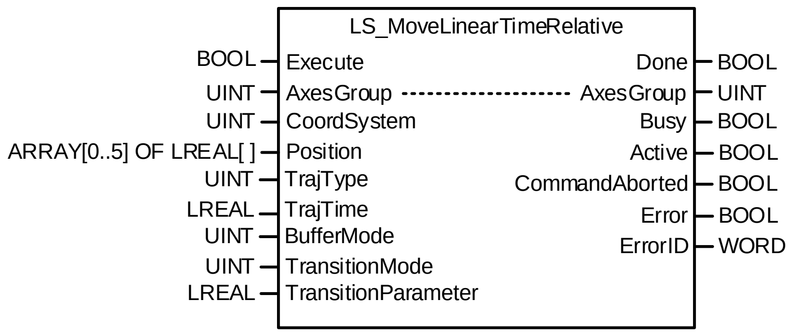
	변수/디바이스	값	타입	설명문
1	_AX01_CPOS	-1.3572036743164063e+001	LREAL	축 01 현재스캔 명령위치
2	_AX02_CPOS	-1.6653305053710937e+001	LREAL	축 02 현재스캔 명령위치
3	_AX03_CPOS	-1.6653121948242188e+001	LREAL	축 03 현재스캔 명령위치
4	_AX04_CPOS	0.0000000000000000e+000	LREAL	축 04 현재스캔 명령위치
5	⊟ LinIntpPosition		ARRAY[0..5] OF LREAL	
6	LinIntpPosition[0]	1.0000000000000000e+002	LREAL	
7	LinIntpPosition[1]	2.0000000000000000e+002	LREAL	Target position
8	LinIntpPosition[2]	-3.8000000000000000e+002	LREAL	
9	LinIntpPosition[3]	0.0000000000000000e+000	LREAL	
10	LinIntpPosition[4]	0.0000000000000000e+000	LREAL	
11	LinIntpPosition[5]	0.0000000000000000e+000	LREAL	
12				

(b) Timing diagram



6.7.7 Time Linear Interpolation Operation for Incremental Position of Coordinate System (LS_MoveLinearTimeRelative)

Motion Function Block



Input-Output		
UINT	AxesGroup	Set the axes group to set the Incremental position time linear interpolation. (1 ~ 16: 1Group ~ 16 Group)
Input		
BOOL	Execute	Give the time linear interpolation command on the axes group in the rising Edge.
UINT	CoordSystem	Set the coordinate system type (1:MCS 2:PCS)
LREAL[]	Position	Enter the target position of the end point of the machine.
UINT	TrajType	Enter the operation acc/dec type.(0:Trapezoid 1:Sine1 2:Sine2)
LREAL	TrajTime	Set the time taken to reach the target position.(msec)
UINT	BufferMode	Specify the sequential operation setting of motion function block. (Refer to 6.1.4.BufferMode input)
UINT	TransitionMode	Specify the route change mode of group operation. Refer to (6.1.6.TransitionMode)
LREAL	TransitionParameter	Specify the parameter of the route change setting of group operation. Refer to(6.1.6.TransitionMode)
Output		
BOOL	Done	Indicate the PCS setting is successfully completed.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that machine information setting of the current axis is running.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted while it is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block issues relative position/time linear interpolation command based on coordinate system on the axes group designated by AxesGroup input
(2) When this motion function block is executed, interpolation control is performed in a linear trajectory from the machine end point of each axes group to the target position.
(3) TrajType input sets the type of velocity, acceleration, deceleration of interpolation trajectory. The type can be selected from three types: Trapezoid/Sine1/Sine2.

(4) TrajTime sets the time taken to reach the target position.
(5) Please refer to 8. 4. 7 Coordinate System Relative Position/Time Linear Interpolation Control for further details.

6.7.8 Circular interpolation operation for absolute position of coordinate system (MC_MoveCircularAbsolute2D)

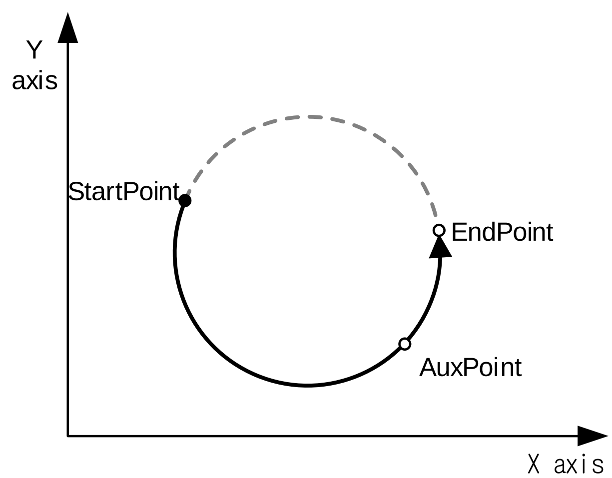
Motion Function Block

```
                        MC_MoveCircularAbsolute2D
     BOOL ─ Execute                              Done ─ BOOL
     UINT ─ AxesGroup ------------------- AxesGroup ─ UINT
     UINT ─ CircMode                             Busy ─ BOOL
 LREAL[ ] ─ AuxPoint                           Active ─ BOOL
 LREAL[ ] ─ EndPoint                 CommandAborted ─ BOOL
     UINT ─ PathChoice                          Error ─ BOOL
    LREAL ─ Velocity                          ErrorID ─ WORD
    LREAL ─ Acceleration
    LREAL ─ Deceleration
    LREAL ─ Jerk
     UINT ─ CoordSystem
     UINT ─ BufferMode
     UINT ─ TransitionMode
    LREAL ─ TransitionParameter
```

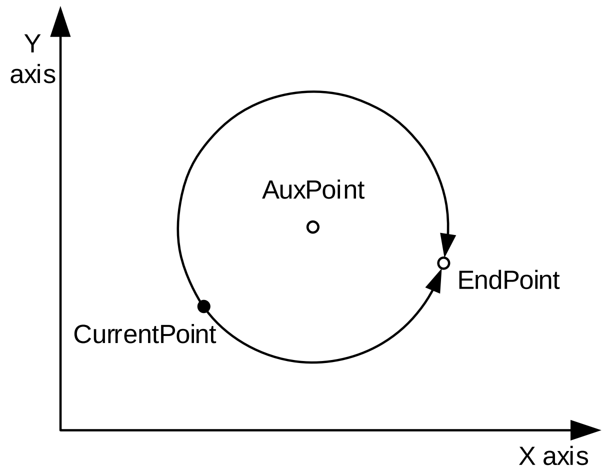
Input-Output		
UINT	AxesGroup	Set the group to do absolute position circular interpolation operation. (1 ~ 16: 1Group ~ 16 Group)
Input		
BOOL	Execute	Give absolute position circular interpolation operation command to the relevant group in the rising Edge.
UINT	CircMode	Circular interpolation method setting [0: Midpoint, 1: Center point, 2: Radius]
LREAL[ ]	AuxPoint	Specify the position of auxiliary point depending on the circular interpolation method in an absolute coordinate.
LREAL[ ]	EndPoint	Set the circular end point as an absolute coordinate.
BOOL	PathChoice	Circular route selection 0: clockwise direction, 1: counter-clockwise direction
LREAL	Velocity	Specify the maximum speed of the route. [u/s]
LREAL	Acceleration	Specify the maximum acceleration. [u/s^2]
LREAL	Deceleration	Specify the maximum deceleration. [u/s^2]
LREAL	Jerk	Specify the change rate of acceleration/deceleration. [u/s^3]
UINT	CoordSystem	Set the coordinate system type (1:MCS 2:PCS)
UINT	BufferMode	Specify the sequential operation setting of motion function block. (Refer to 6.1.4.BufferMode input)
UINT	TransitionMode	Unused
LREAL	TransitionParameter	Unused
Output		
BOOL	Done	Indicate whether to reach the specified position.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that whether or not motion function block is controlling the group.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted while it is running.
BOOL	Error	Indicates whether an error occurs or not.

WORD	ErrorID	Output the number of error occurred while motion function block is running.

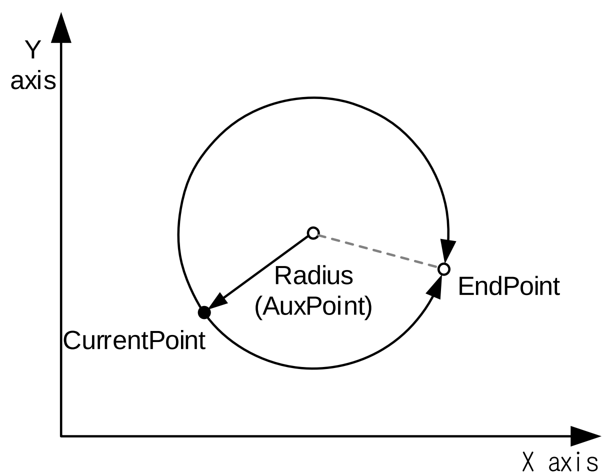
(1) This motion function block issues absolute position circular interpolation command based on coordinate system on the axis group designated by AxesGroup input.
(2) When this motion function block starts, each axis performs circular trajectory interpolation control referring to the auxiliary point input, and the movement direction is determined by Path Choice input. If PathChoice input is set to 0, circular interpolation is operated in a clockwise direction, and if it is set to 1, circular interpolation is operated in a counter-clockwise direction.
(3) Specify the absolute position of the auxiliary point to refer when doing circular interpolation of each axis in AuxPoint and EndPoint inputs as array. The input corresponds in the order of X, Y, Z, unlike MC_MoveCircularAbsolute.
(4) In Velocity, Acceleration, Deceleration, Jerk inputs, the acceleration, deceleration, change rate of acceleration, velocity of the interpolation path are specified, respectively.
(5) Set the circular interpolation method in CircMode input. The circular interpolation methods corresponding to CircMode values are as follows.
(a) Circular Interpolation Using Midpoint Specification (CircMode = 0)
This method performs circular interpolation by starting operation at the start position, passing the designated midpoint, and reaching the target position. In the figure below, the start position corresponds to the axes group coordinate at the start of the command, the midpoint corresponds to the coordinate input for the AuxPoint, and the target position corresponds to the absolute coordinate input for the EndPoint.



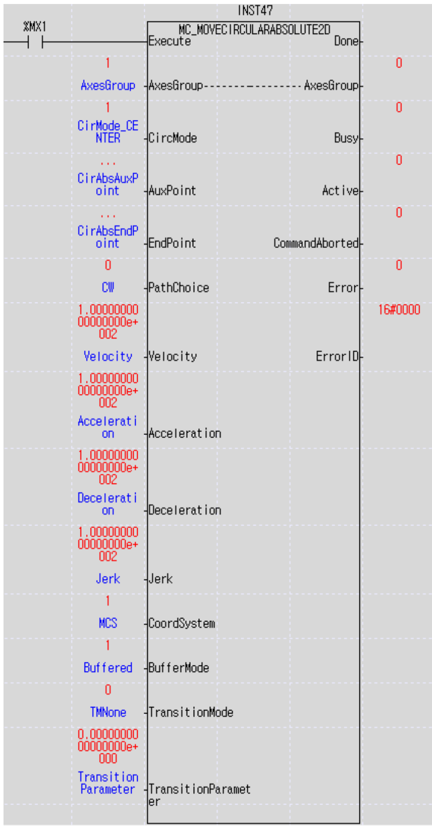
(b) Circular Interpolation Using Center Point Specification (CircMode = 1)
This method performs circular interpolation to the target position by starting operation at the current position, and following a circular trajectory of which diameter corresponds to the distance to the designated center point. In the figure below, the current position corresponds to the axes group coordinate at the start of the command, the center point corresponds to the coordinate input for the AuxPoint, and the target position corresponds to the absolute coordinate input for the EndPoint.



(c) Circular Interpolation using Radius Speciation (CircMode = 2)
This method performs circular interpolation to the target position by starting operation at the current position, and following a circular trajectory with a designated radius from the current position to the target position. In the figure below, the current position corresponds to the axes group coordinate at the start of the command, the radius corresponds to the X coordinate input for the AuxPoint, and the target position corresponds to the absolute coordinate input for the EndPoint.

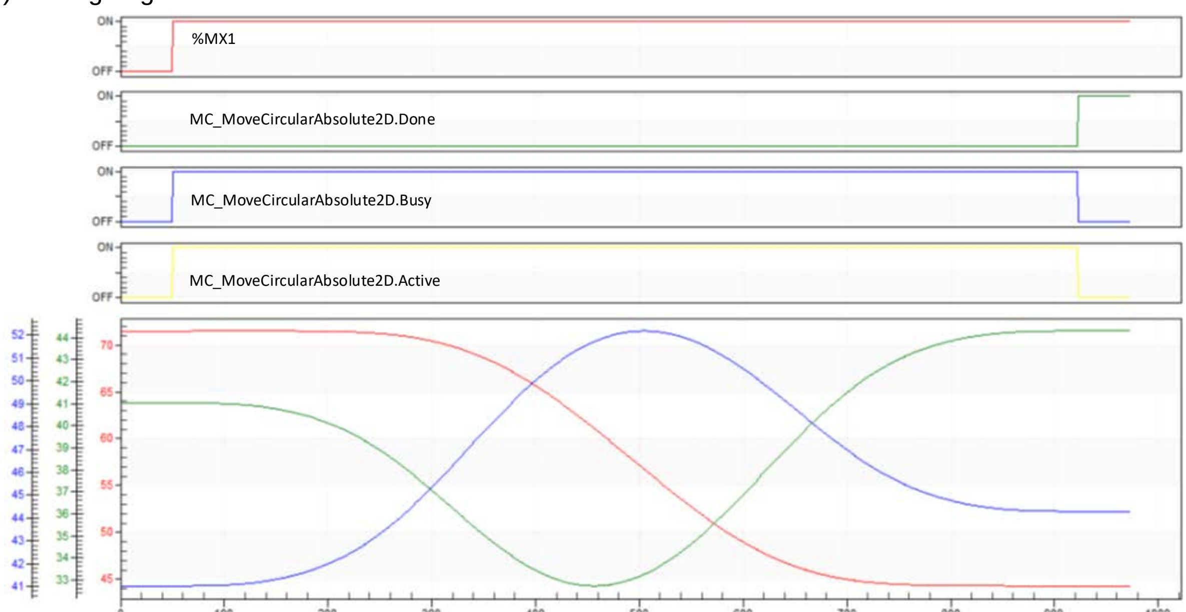


(6) Refer to chapter 8.4.7 circular interpolation control in motion controller's manual for more details.
(7) If the function block is re-executed (Execute input is On) before the instruction is terminated, the changed parameters are applied.
Only, Velocity, Acceleration, Deceleration, Jerk, AuxPoint, Endpoint input can be updated.
(8) Velocity input can be set to 0 or changed.
(9) Example program
This example is to set the center point at (0, 75, -580) when the current command position is MCS (0,150,-580), and perform circular interpolation to the target position MCS(0,0,-580) by moving in a clockwise direction.
(a) Function block setting



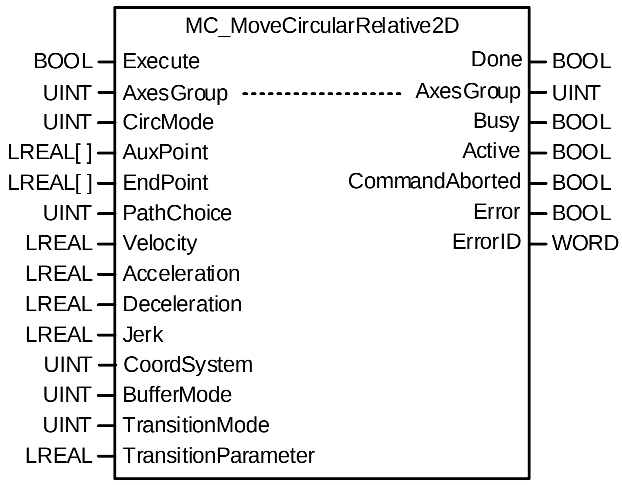
	변수/디바이스	값	타입	설명문
1	_AX01_CPOS	7.1485269050584748e+001	LREAL	축 01 현재스캔 명령위치
2	_AX02_CPOS	4.1026455810822632e+001	LREAL	축 02 현재스캔 명령위치
3	_AX03_CPOS	4.1026455810822632e+001	LREAL	축 03 현재스캔 명령위치
4	⊟ CirAbsAuxPoint		ARRAY[0..2] OF LREAL	
5	CirAbsAuxPoint[0]	0.0000000000000000e+000	LREAL	Center
6	CirAbsAuxPoint[1]	7.5000000000000000e+001	LREAL	point
7	CirAbsAuxPoint[2]	-5.8000000000000000e+002	LREAL	
8	⊟ CirAbsEndPoint		ARRAY[0..2] OF LREAL	
9	CirAbsEndPoint[0]	0.0000000000000000e+000	LREAL	Target
10	CirAbsEndPoint[1]	0.0000000000000000e+000	LREAL	position
11	CirAbsEndPoint[2]	-5.8000000000000000e+002	LREAL	

(b) Timing diagram



6.7.9 Circular interpolation operation for Incremental position of coordinate system (MC_MoveCircularRelative2D)

Motion Function Block



Input-Output		
UINT	AxesGroup	Set the group to do relative position circular interpolation operation. (1 ~ 16: 1Group ~ 16 Group)
Input		
BOOL	Execute	Give relative position circular interpolation operation command to the relevant group in the rising Edge.
UINT	CircMode	Circular interpolation method setting [0: Midpoint, 1: Central point, 2: Radius]
LREAL[ ]	AuxPoint	Specify the position of auxiliary point depending on the circular interpolation method as the relative coordinate based on the starting point.
LREAL[ ]	EndPoint	Specify the end point of circular arc as the relative coordinate based on the starting point.
BOOL	PathChoice	Circular route selection 0: clockwise direction, 1: counter-clockwise direction
LREAL	Velocity	Specify the maximum speed of the route. [u/s]
LREAL	Acceleration	Specify the maximum acceleration. [u/s^2]
LREAL	Deceleration	Specify the maximum deceleration. [u/s^2]
LREAL	Jerk	Specify the change rate of acceleration/deceleration. [u/s^3]
UINT	CoordSystem	Set the coordinate system type (1:MCS 2:PCS)
UINT	BufferMode	Specify the sequential operation setting of motion function block. (Refer to 6.1.4.BufferMode input)
UINT	TransitionMode	Unused
LREAL	TransitionParameter	Unused
Output		
BOOL	Done	Indicate whether to reach the specified position.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that whether or not motion function block is controlling the group.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted while it is running.

BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

This motion function block issues relative position circular interpolation command on the axes group designated by AxesGroup input.

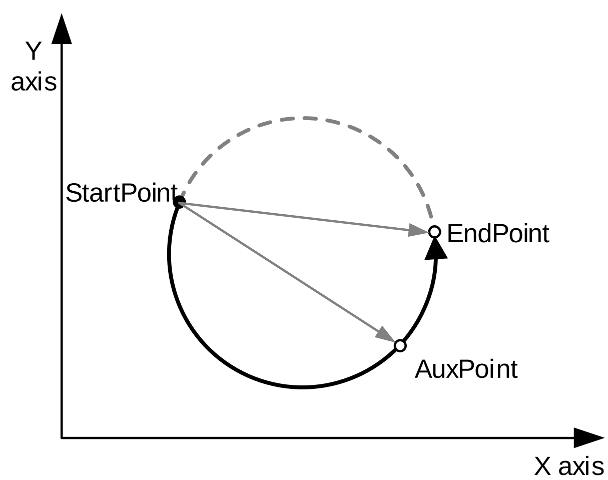
(1) When this motion function block starts, each axis performs circular trajectory interpolation control referring to the auxiliary point input, and the movement direction is determined by Path Choice input. If PathChoice input is set to 0, circular interpolation is operated in a clockwise direction, and if it is set to 1, circular interpolation is operated in a counter-clockwise direction.

(2) At AuxPoint and EndPoint input, designate the arrangement of the relative position of auxiliary points to refer to for circular interpolation of each axis. The input arrangement and the axes of the group correspond to the designated axis IDs [ID1, ID2, ID3, ···], in that order. (The 3 LEAL type sized array should be entered in Position input as there are 3 axes which comprise the group to give a circular interpolation operation command.)

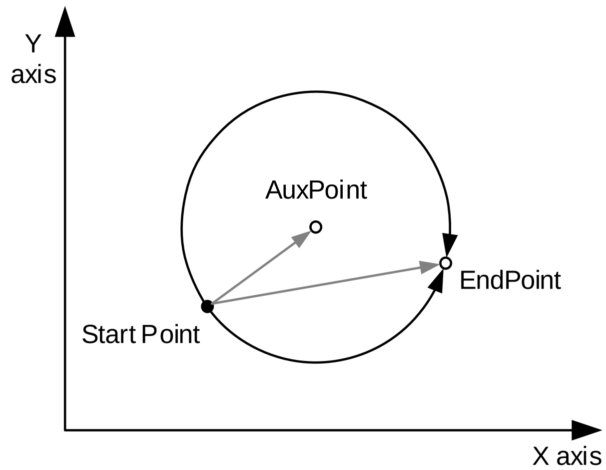
(3) In Velocity, Acceleration, Deceleration, Jerk inputs, the acceleration, deceleration, change rate of acceleration, velocity of the interpolation path are specified, respectively.

(4) Set the circular interpolation method in CircMode input. The circular interpolation methods corresponding to CircMode values are as follows.

(a) Circular interpolation of midpoint specifying method (BORDER, CircMode = 0) in this method, operation starts at the current position and it does circular interpolation through the pecified position of the central point to the target position. In the figure below, the current position corresponds to the axes group coordinate at the start of the command, the midpoint corresponds to the coordinate input for the AuxPoint, and the target position corresponds to the relative coordinate input for the EndPoint.

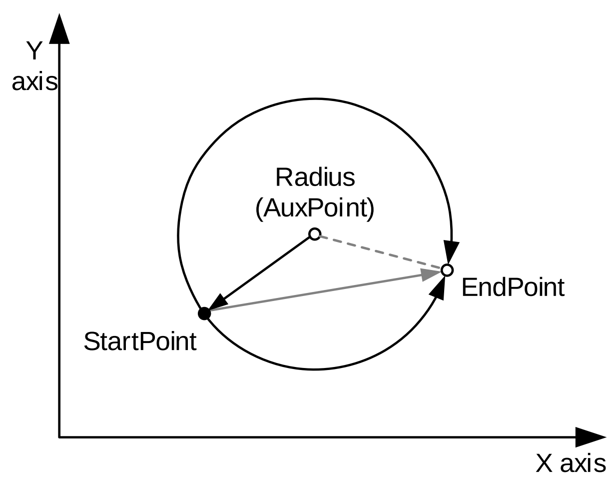


(b) Circular interpolation of central point specifying method in this method, operation starts at the current position, and it does circular interpolation to the target position along the circular path, which has a radius of the distance to the specified central position. In the figure below, the current position corresponds to the axes group coordinate at the start of the command, the center point corresponds to the coordinate input for the AuxPoint, and the target position corresponds to the relative coordinate input for the EndPoint.



(c) Circular interpolation with radius designation form

In this method, operation starts at the current position, and it does circular interpolation to the target position along the circular path which has a radius of the value specified in the radius. In the figure below, the current position corresponds to the axes group coordinate at the start of the command, the diameter corresponds to the X coordinate input for the AuxPoint, and the target position corresponds to the relative coordinate input for the EndPoint.



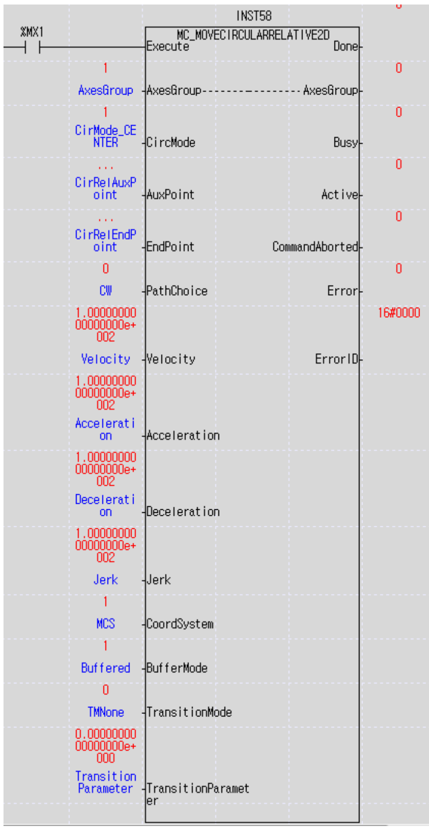
(5) Refer to chapter 8.4.7 circular interpolation control in motion controller's manual for more details.

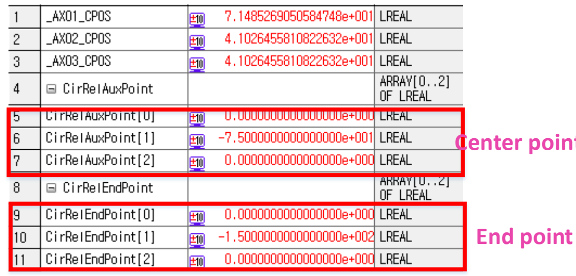
(6) If the function block is re-executed (Execute input is On) before the instruction is terminated, the changed parameters are applied.
Only, Velocity, Acceleration, Deceleration, Jerk, AuxPoint, Endpoint input can be updated.

(7) Velocity input can be set to 0 or changed.

(8) Example program
This example is to set the center point specification method when the current command position is (1000, 1000) (set the relative position from the current position to the center point to set: 1000, 1000), and move clock-wise to perform circular interpolation to the target position (set the relative position from the current position to the target position: 0, 0).

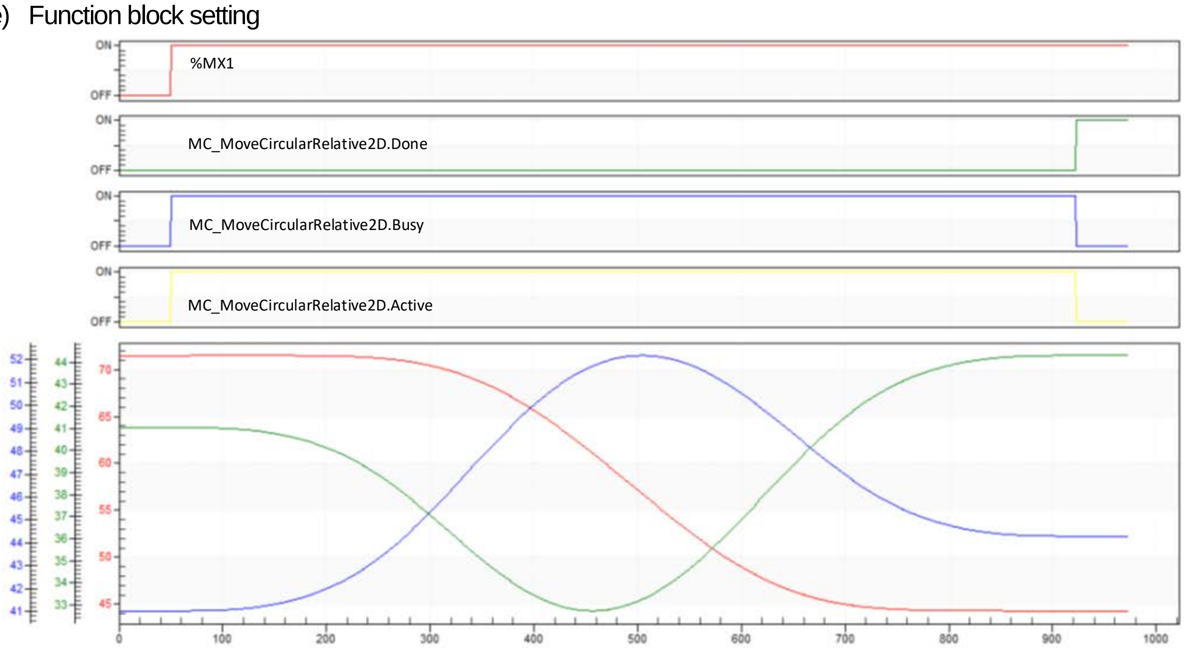
(d) Function block setting





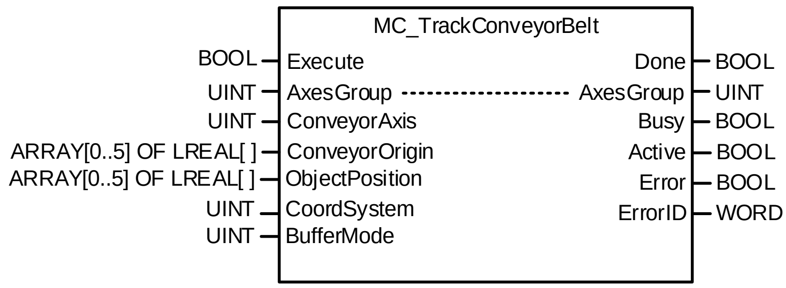
1	_AX01_CPOS	7.1485269050584748e+001	LREAL
2	_AX02_CPOS	4.1026455810822632e+001	LREAL
3	_AX03_CPOS	4.1026455810822632e+001	LREAL
4	⊟ CirRelAuxPoint		ARRAY[0..2] OF LREAL
5	CirRelAuxPoint[0]	0.0000000000000000e+000	LREAL
6	CirRelAuxPoint[1]	-7.5000000000000000e+001	LREAL
7	CirRelAuxPoint[2]	0.0000000000000000e+000	LREAL
8	⊟ CirRelEndPoint		ARRAY[0..2] OF LREAL
9	CirRelEndPoint[0]	0.0000000000000000e+000	LREAL
10	CirRelEndPoint[1]	-1.5000000000000000e+002	LREAL
11	CirRelEndPoint[2]	0.0000000000000000e+000	LREAL

(e) Function block setting



6.7.10 Synchronization setting of conveyor belt (MC_TrackConveyorBelt)

Motion Function Block

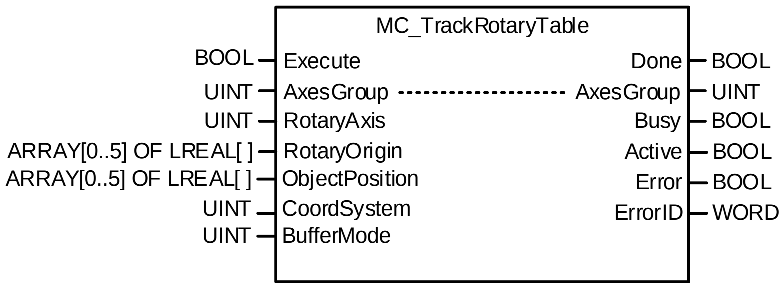


Input-Output		
UINT	AxesGroup	Set the group to perform Conveyor Belt Synchronization. (1 ~ 16: 1Group ~ 16 Group)
Input		
BOOL	Execute	Give the conveyor belt synchronized setting command on the axes group in the rising Edge.
UINT	ConveyorAxis	Set the conveyor axis.(1 ~ 32 : Axis 1~Axis 32)
LREAL[]	ConveyorOrigin	Enter the position from the MCS home position to the conveyor origin point.
LREAL[]	ObjectPosition	Input the conveyor home position to the object to work on.
UINT	CoordSystem	Set the coordinate system type.(2:PCS)
UINT	BufferMode	Specify the sequential operation setting of motion function block. (Refer to 6.1.4.BufferMode input)
Output		
BOOL	Done	Indicate the PCS setting is successfully completed.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that machine information setting of the current axis is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block sets conveyor belt synchronized operation for the axes group designated by AxesGroup input.
(2) This motion function block is not directly involved in operation. When this function block is executed, the coordinate system operation using the PCS coordinate system is synchronized to the designated conveyor belt axis.
(3) ConveyorAxis can be set to between 1 and 32. An axis belonging to the axes group set as AxesGroup cannot be designated.
(4) The operation parameter of the axis designated as ConveyorAxis must be in mm/inch.
(5) Infinite running repeat must be set for the operation parameter of the axis designated as ConveyorAxis
(6) Synchronized conveyor operation is terminated by performing coordinate system operation using the PCS coordinate system or performing PCS setting with MC_SetCartesianTransform function block.
(7) Refer to chapter 8.4.9 synchronized conveyor operation in motion controller's manual for more details

6.7.11 Synchronization setting of the rotary table (MC_TrackRotaryTable)

Motion Function Block

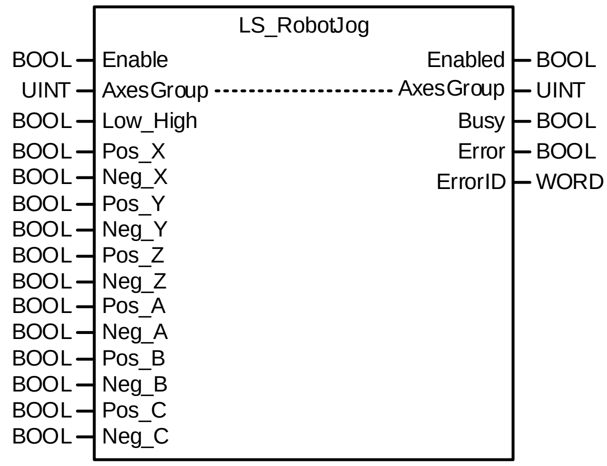


Input-Output		
UINT	AxesGroup	Set the group to perform Rotary table synchronization. (1 ~ 16: 1Group ~ 16 Group)
Input		
BOOL	Execute	Give the conveyor belt synchronized setting command on the axes group in the rising Edge.
UINT	RotaryAxis	Set the rotary table axis.(1 ~ 32 : Axis 1~Axis 32)
LREAL[]	RotaryOrigin	Enter the position from the MCS home position to the rotary table origin point.
LREAL[]	ObjectPosition	Input the rotary table home position to the object to work on.
UINT	CoordSystem	Set the coordinate system type.(2:PCS)
UINT	BufferMode	Specify the sequential operation setting of motion function block. (Refer to 6.1.4.BufferMode input)
Output		
BOOL	Done	Indicate the PCS setting is successfully completed.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that machine information setting of the current axis is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block sets rotary table synchronized operation for the axes group designated by AxesGroup input.
(2) This motion function block is not directly involved in operation. When this function block is executed, the coordinate system operation using the PCS coordinate system is synchronized to the designated conveyor belt axis.
(3) RotaryAxis can be set to between axis 1 and axis 32 belonging to the axes group set as AxesGroup cannot be designated.
(4) The operation parameter of the axis designated as RotaryAxis must be in mm/inch.
(5) Infinite running repeat must be set for the operation parameter of the axis designated as RotaryAxis
(6) Synchronized rotary table operation is terminated by performing coordinate system operation using the PCS coordinate system or performing PCS setting with MC_SetCartesianTransform function block.
(7) Refer to chapter 8.4.10 synchronized rotary table operation in motion controller's manual for more details

6.7.12 JOG operation of the coordinate system (LS_RobotJog)

Function Block type

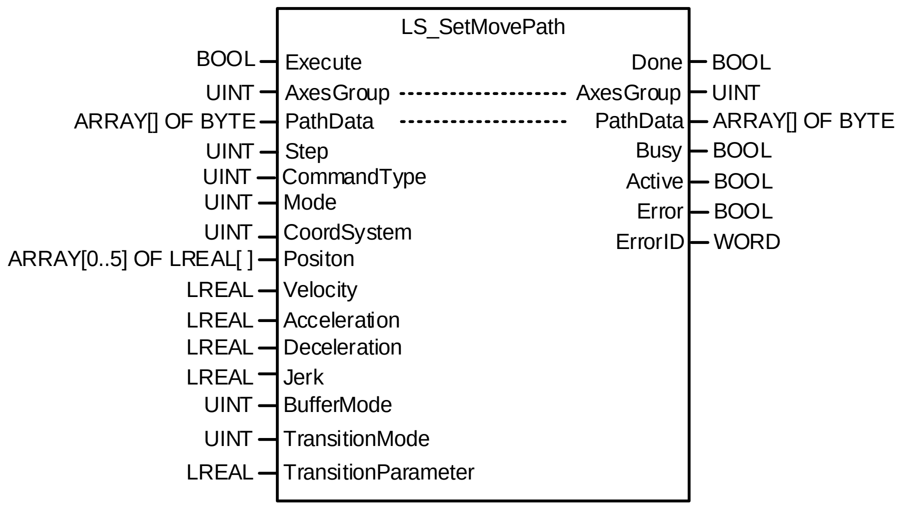


Input-Output		
UINT	AxesGroup	Specify the axis group to be commanded. (1 ~ 16: 1Group ~ 16 Group)
Input		
BOOL	Enable	While the input is ON, the JOG operation command is sent to the relevant axis group.
BOOL	Low_High	Set the JOG speed in JOG operation. (0: Jog low speed operation, 0: Jog high speed operation)
BOOL	Pos_X	Set the linear operation direction at direction in jog (X axis + direction)
BOOL	Neg_X	Set the linear operation direction at direction in jog (X axis- direction)
BOOL	Pos_Y	Set the linear operation direction at direction in jog (Y axis + direction)
BOOL	Neg_Y	Set the linear operation direction at direction in jog (Y axis - direction)
BOOL	Pos_Z	Set the linear operation direction at direction in jog (Z axis + direction)
BOOL	Neg_Z	Set the linear operation direction at direction in jog (Z axis - direction)
BOOL	Pos_A	Set the rotary operation direction at JOG operation. (X-axis counterclockwise rotation)
BOOL	Neg_A	Set the rotary operation direction at JOG operation. (X-axis clockwise rotation)
BOOL	Pos_B	Set the rotary operation direction at JOG operation. (Y-axis counterclockwise rotation)
BOOL	Neg_B	Set the rotary operation direction at JOG operation. (Y-axis clockwise rotation)
BOOL	Pos_C	Set the rotary operation direction at JOG operation. (Z -axis counterclockwise rotation)
BOOL	Neg_C	Set the rotary operation direction at JOG operation. (Z -axis clockwise rotation)
Output		
BOOL	Enabled	It indicates that the axis group is in the process of JOG operation.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block executes the JOG operation of the coordinate system for the corresponding axis group.
(2) Jog is a manual operation function for test and is used to confirm the position address for system operation, wiring condition check, and teaching. Jog can be used by dividing the speed into high speed and low speed.
(3) It can be respectively applied to both high speed and low speed. If you change the value set in Low / High when the Enable input is On (JOG operation status), the speed will change without stopping JOG operation.
(4) If both the forward (Pox_) / reverse (Neg_) inputs are set for the same axis, the axis will stop.

6.7.13 Set path operation data (LS_SetMovePath)

Motion Function Block



Input-Output		
UINT	AxesGroup	Set the group to set the path operation data. (1 ~ 16: 1Group ~ 16 Group)
BYTE[]	PathData	Set the location where the path data is stored.
Input		
BOOL	Execute	In the rising Edge, it sends the command for setting the path operation data to the corresponding axis group.
UINT	Step	Enter the step number of the path data. (The step number is affected by the size of the data set in PathData.)
UINT	CommandType	Select the type of path operation. 0: none 1: Linear interpolation operation for the absolute position of the coordinate system, 2: Linear interpolation operation for the relative position of the coordinate system 3: Circular interpolation operation for the absolute position of the coordinate system, 4: Circular interpolation operation for the relative position of the coordinate system
UINT	Mode	Select the method and path for circular interpolation operation of the coordinate system 0/1/2: Clockwise, Midpoint/Central point/Radius 3/4/5: counter-clockwise Midpoint/Central point/Radius
UINT	CoordSystem	Set the coordinate system type (1:MCS 2:PCS)
LREAL[]	Position	Enter the target position of the end point of the machine. In the circular interpolation, the circular interpolation end point should be set in Position [0] Position [1] Position [2]. In the circular interpolation, the Central point/Waypoint should be set in Position [3] Position [4] Position [5]. In the circular interpolation, the Radius should be in Position [3].
LREAL	Velocity	Specify the maximum speed of the route. [u/s]
LREAL	Acceleration	Specify the maximum acceleration. [u/s^2]

LREAL	Deceleration	Specify the maximum deceleration. [u/s^2]
LREAL	Jerk	Specify the change rate of acceleration/deceleration. [u/s^3]
UINT	BufferMode	Specify the sequential operation setting of motion function block. (Refer to 6.1.4.BufferMode input)
UINT	TransitionMode	Unused
LREAL	TransitionParameter	Unused

Output		
BOOL	Done	Indicate that the path data setting is completed successfully.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that machine information setting of the current axis is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

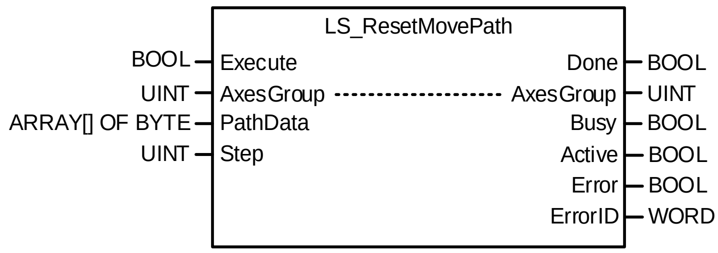
(1) This motion function block is the function block that sets the path data for the axis group specified in the AxesGroup input.
(2) The step value can be set from 0, and the size of one step is 96 Bytes.
(3) The path data is saved in the area of data set in PathData. The variable set in PathData should be set to 96 times or more of the number of the steps to use.
(4) The CommandType value selects the operation method for the path operation. If the CommandType value is set to 0, it is considered that the data for the corresponding step is not set during path operation.
(5) The Mode value sets the direction of the circular interpolation when performing the circular interpolation operation.
(6) The value of BufferMode should be set to 1(Buffered).
(7) Set position[] value to the TCP target position. Depending on the robot type, some Position variable areas may not be applied. Data input in the unapplied areas is not reflected in coordinate system operation.

Variable	Meaning	Unit
Position[0]	X axis position	mm
Position[1]	Y axis position	mm
Position[2]	Z axis position	mm
Position[3]	X axis rotation amount	degree
Position[4]	Y axis rotation amount	degree
Position[5]	Z axis rotation amount	degree

(8) For more details, refer to Section 8.4.10, "Path Operation of the Coordinate System ".

6.7.14 Delete path operation data (LS_ResetMovePath)

Motion Function Block

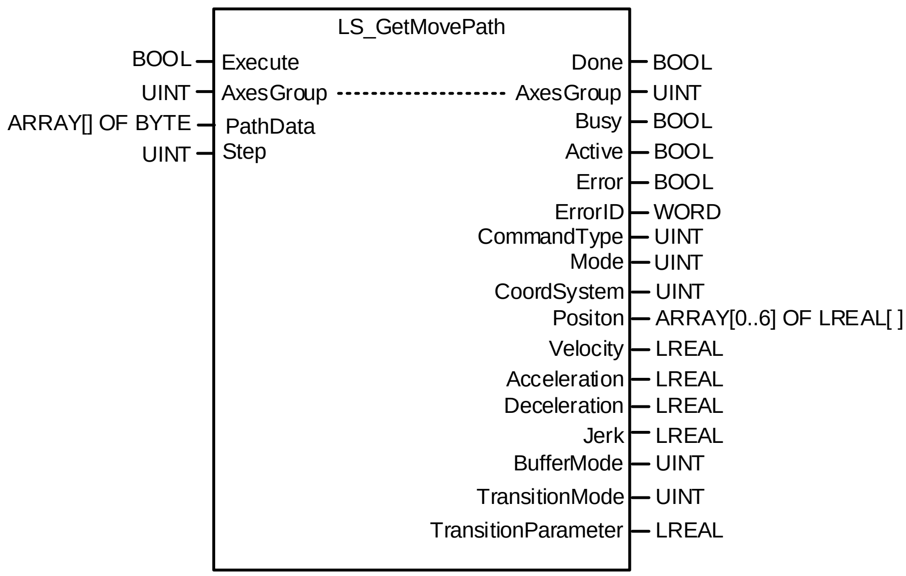


Input-Output		
UINT	AxesGroup	Set the group to delete the path operation data. (1 ~ 16: 1Group ~ 16 Group)
Input		
BOOL	Execute	In the rising Edge, the command for deleting the path operation data is sent to the corresponding axis group.
BYTE[]	PathData	Set the location where the path data is stored.
UINT	Step	Enter the step number of the path data. (The step number is affected by the size of the data set in PathData.)
Output		
BOOL	Done	Indicate the deleting the path data is done successfully.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that machine information setting of the current axis is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(1) This motion function block is the function block to delete the path data of the axis group specified in the AxesGroup input.
(2) The step value can be set from 0, and the size of one step is 96 Bytes.
(3) The path data is saved in the area of data set in PathData. The variable set in PathData should be set to 96 times or more of the number of the steps to use.
(4) For more details, refer to Section 8.4.10, "Path Operation of the Coordinate System ".

6.7.15 Read path operation data (LS_GetMovePath)

Motion Function Block



Input-Output		
UINT	AxesGroup	Set the group to set the path operation data. (1 ~ 16: 1Group ~ 16 Group)
Input		
BOOL	Execute	In the rising Edge, it sends the command for setting the path operation data to the corresponding axis group.
BYTE[]	PathData	Set the location where the path data is stored.
UINT	Step	Enter the step number of the path data. (The step number is affected by the size of the data set in PathData.)
Output		
BOOL	Done	Indicate that the path data setting is completed successfully.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that machine information setting of the current axis is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.
UINT	CommandType	Output the type of path operation. 0: none 1: Linear interpolation operation for the absolute position of the coordinate system, 2: Linear interpolation operation for the relative position of the coordinate system 3: Circular interpolation operation for the absolute position of the coordinate system, 4: Circular interpolation operation for the relative position of the coordinate system
UINT	Mode	Output the operation mode.
UINT	CoordSystem	Output the coordinate system type.(1:MCS 2:PCS)
LREAL[]	Position	Output the target position.
LREAL	Velocity	Output the maximum speed of the path. [u/s]

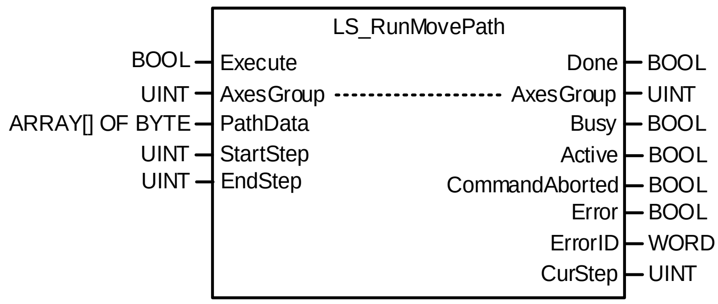
LREAL	Acceleration	Output the maximum acceleration. [u/s^2]
LREAL	Deceleration	Output the maximum deceleration. [u/s^2]
LREAL	Jerk	Output the change rate of acceleration/deceleration. [u/s^3]

UINT	BufferMode	Specify the sequential operation setting of motion function block. (Refer to 6.1.4.BufferMode input)
UINT	TransitionMode	Unused
LREAL	TransitionParameter	Unused

(1) This motion function block is the function block to read the path data to the axis group specified in AxesGroup input.
(2) The step value can be set from 0, and the size of one step is 96 Bytes.
(3) The path data is saved in the area of data set in PathData. The variable set in PathData should be set to 96 times or more of the number of the steps to use.
(4) For more details, refer to Section 8.4.10, "Path Operation of the Coordinate System ".

6.7.16 Perform path operation (LS_RunMovePath)

Motion Function Block

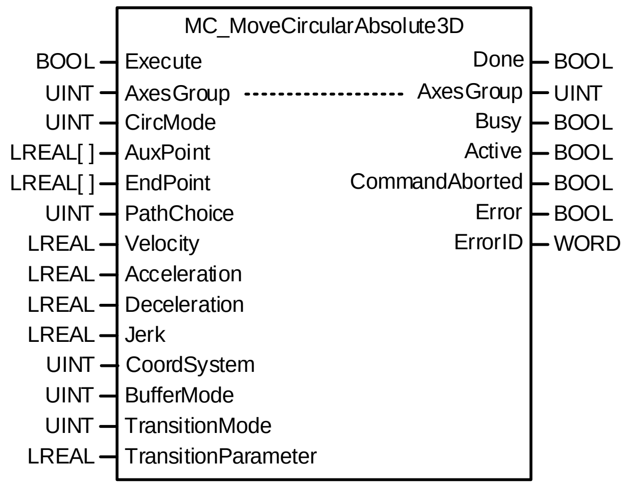


Input-Output		
UINT	AxesGroup	Set the group to run the path operation data. (1 ~ 16: 1Group ~ 16 Group)
Input		
BOOL	Execute	In the rising Edge, it sends the command for setting the path operation data to the corresponding axis group.
BYTE[]	PathData	Set the location where the path data is stored.
UINT	StartStep	Enter the start step number of the path data. (The step number is affected by the size of the data set in PathData.)
UINT	EndStep	Enter the end step number of the path data. (The step number is affected by the size of the data set in PathData.)
Output		
BOOL	Done	Indicate that the path data setting is completed successfully.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that machine information setting of the current axis is running.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted while it is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.
UINT	CurStep	Output the currently running step number.

(1) This motion function block is the function block to execute the path operation for the axis group specified in the AxesGroup input.
(2) The step value can be set from 0, and the size of one step is 96 Bytes.
(3) The path data is saved in the area of data set in PathData. The variable set in PathData should be set to 96 times or more of the number of the steps to use.
(4) The difference between StartStep and EndStep cannot be set to 100 or more. (Up to 100 step operations can be executed at one time.)
(5) If the CommandType of path data is 0 during the path operation, the operation is terminated even if EndStep is not reached.
(6) If the path operation is executed, the current step number in operation is output to the CurStep.
(7) For more details, refer to Section 8.4.10, "Path Operation of the Coordinate System ".

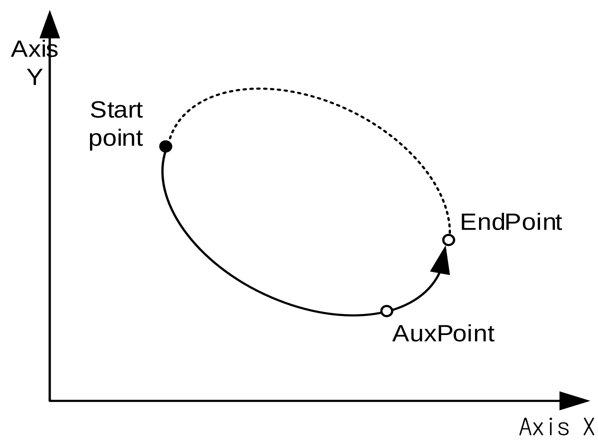
6.7.17 3D Circular interpolation operation for absolute position of coordinate system (MC_MoveCircularAbsolute3D)

Motion Function Block

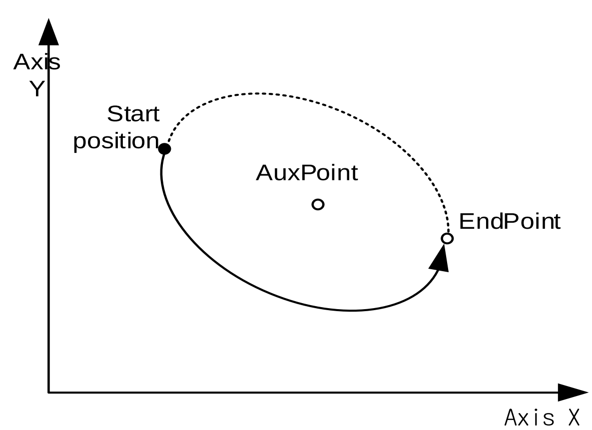


Input-Output		
UINT	AxesGroup	Set the group to do absolute position circular interpolation operation. (1 ~ 16: 1Group ~ 16 Group)
Input		
BOOL	Execute	Give absolute position circular interpolation operation command to the relevant group in the rising Edge.
UINT	CircMode	Circular interpolation method setting [0: Midpoint, 1: Central point, 2: Radius]
LREAL[ ]	AuxPoint	Specify the position of auxiliary point depending on the circular interpolation method in an absolute coordinate.
LREAL[ ]	EndPoint	Set the circular end point as an absolute coordinate.
BOOL	PathChoice	Circular route selection(0: clockwise direction, 1: counter-clockwise direction)
LREAL	Velocity	Specify the maximum speed of the route. [u/s]
LREAL	Acceleration	Specify the maximum acceleration. [u/s^2]
LREAL	Deceleration	Specify the maximum deceleration. [u/s^2]
LREAL	Jerk	Specify the change rate of acceleration/deceleration. [u/s^3]
UINT	CoordSystem	Set the coordinate system type (1:MCS 2:PCS)
UINT	BufferMode	Specify the sequential operation setting of motion function block. (Refer to 6.1.4.BufferMode input)
UINT	TransitionMode	Unused
LREAL	TransitionParameter	Unused
Output		
BOOL	Done	Indicate whether to reach the specified position.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that whether or not motion function block is controlling the group.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted while it is running.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

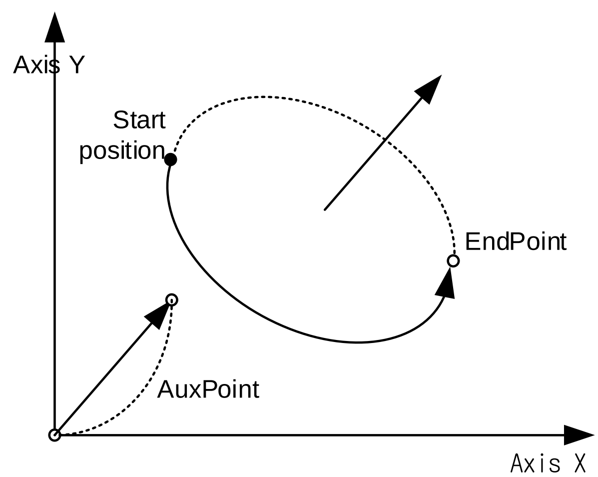
(10) This motion function block issues absolute position 3D circular interpolation command based on coordinate system on the axis group designated by AxesGroup input.
(11) Unlike MC_MoveCircularAbsolute2D function block that performs circular interpolation on the XY plane, MC_MoveCircularAbsolute3D function block performs circular interpolation on 3D coordinates according to AuxPoint and EndPoint.
(12) When this motion function block starts, each axis performs circular trajectory interpolation control referring to the auxiliary point input, and the movement direction is determined by Path Choice input. If PathChoice input is set to 0, circular interpolation is operated in a clockwise direction, and if it is set to 1, circular interpolation is operated in a counter-clockwise direction.
(13) In Velocity, Acceleration, Deceleration, Jerk inputs, the acceleration, deceleration, change rate of acceleration, velocity of the interpolation path are specified, respectively.
(14) Set the circular interpolation method in CircMode input. The circular interpolation methods corresponding to CircMode values are as follows.
(a) Circular interpolation of midpoint specifying method (BORDER, CircMode = 0) in this method, operation starts at command executing current position and it does circular interpolation through the specified position of the central point to the target position. In the figure below, the start position corresponds to the coordinate system position at the start of the command, the midpoint corresponds to the coordinate input for the AuxPoint, and the target position corresponds to the absolute coordinate input for the EndPoint.



(b) Circular Interpolation Using Center Point Specification (CircMode = 1)
This method performs circular interpolation to the target position by starting operation at the current position executed command, and following a circular trajectory of which diameter corresponds to the distance to the designated center point. In the figure below, the current position corresponds to the axes group coordinate at the start of the command, the center point corresponds to the coordinate input for the AuxPoint, and the target position corresponds to the absolute coordinate input for the EndPoint. In the center point method, the start position, center point position, and target position cannot be located on the same straight line.

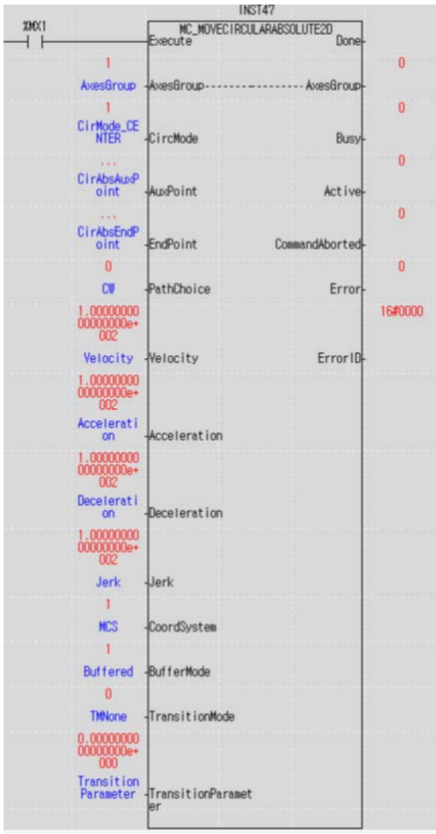


(c) Circular interpolation control using radius designation method (CircMode = 2)
Starts operation from the current position where the command is executed, and performs circular interpolation to the target position along the trajectory of a circular arc whose radius is the specified radius. In the radius designation method, the vertical vector of the arc is set in AuxPoint during circular interpolation. The length of the vertical vector is set to the radius of the circle. When AuxPoint is set to (50,0,0), circular interpolation set to radius 50 is performed on the YZ plane. In the figure below, the coordinates of the axis group at the start of the command correspond to the current position, and the coordinates entered in the EndPoint are matched to the target point as absolute values.



(15) Refer to chapter 8.4.11 3D circular interpolation control in motion controller's manual for more details.
(16) If the function block is re-executed (Execute input is On) before the instruction is terminated, the changed parameters are applied.
Only, Velocity, Acceleration, Deceleration, Jerk, AuxPoint, Endpoint input can be updated.
(17) Velocity input can be set to 0 or changed.
(18) Example program
This example is to set the center point at (0, 75, -580) when the current command position is MCS (0,150,-580), and perform circular interpolation to the target position MCS(0,0,-580) by moving in a clockwise direction.

(c) Function block setting

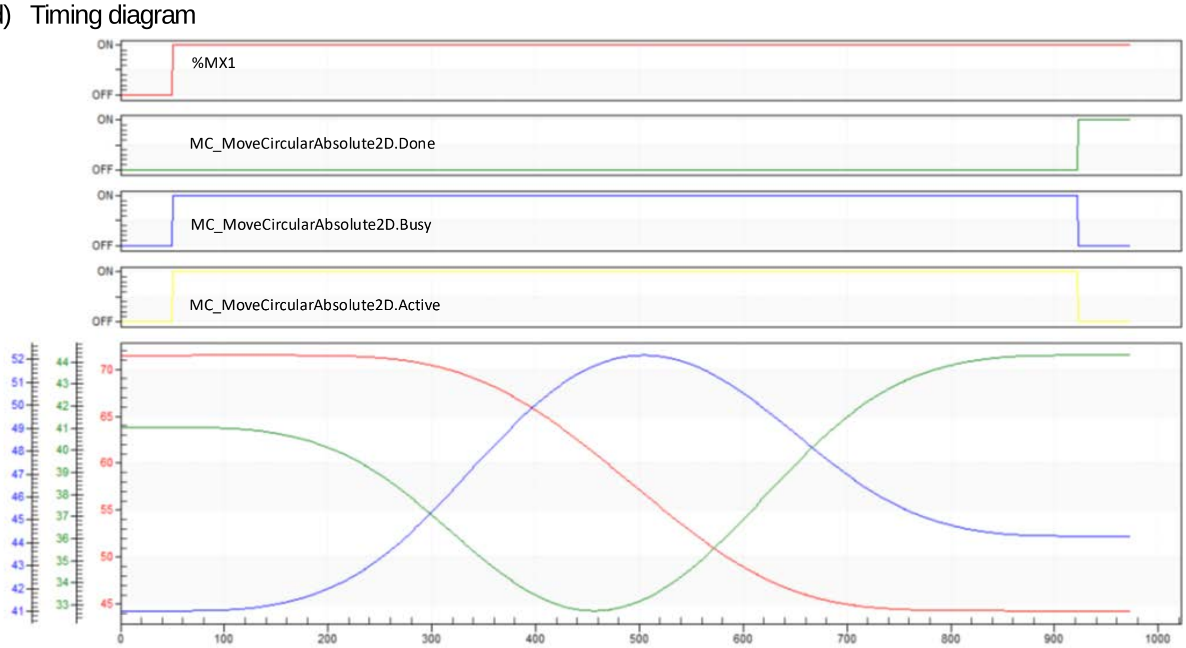


1	_AX01_CPOS	7.1485269050584748e+001	LREAL
2	_AX02_CPOS	4.1026455810822632e+001	LREAL
3	_AX03_CPOS	4.1026455810822632e+001	LREAL
4	⊟ CirAbsAuxPoint		ARRAY[0..2] OF LREAL
5	CirAbsAuxPoint[0]	0.0000000000000000e+000	LREAL
6	CirAbsAuxPoint[1]	7.5000000000000000e+001	LREAL
7	CirAbsAuxPoint[2]	-5.8000000000000000e+002	LREAL
8	⊟ CirAbsEndPoint		ARRAY[0..2] OF LREAL
9	CirAbsEndPoint[0]	0.0000000000000000e+000	LREAL
10	CirAbsEndPoint[1]	0.0000000000000000e+000	LREAL
11	CirAbsEndPoint[2]	-5.8000000000000000e+002	LREAL

CenterPoint

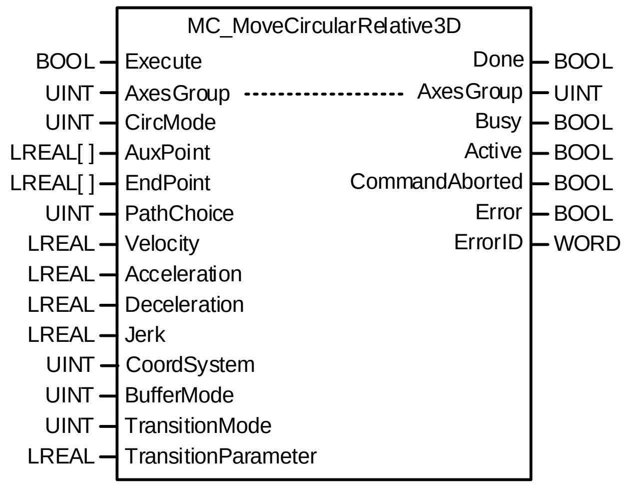
EndPoint

(d) Timing diagram



6.7.18 3D Circular interpolation operation for Incremental position of coordinate system (MC_MoveCircularRelative3D)

Motion Function Block



Input-Output		
UINT	AxesGroup	Set the group to do relative position circular interpolation operation. (1 ~ 16: 1Group ~ 16 Group)
Input		
BOOL	Execute	Give relative position circular interpolation operation command to the relevant group in the rising Edge.
UINT	CircMode	Circular interpolation method setting [0: Midpoint, 1: Central point, 2: Radius]
LREAL[ ]	AuxPoint	Specify the position of auxiliary point depending on the circular interpolation method as the relative coordinate based on the starting point.
LREAL[ ]	EndPoint	Specify the end point of circular arc as the relative coordinate based on the starting point.
BOOL	PathChoice	Circular route selection 0: clockwise direction, 1: counter-clockwise direction
LREAL	Velocity	Specify the maximum speed of the route. [u/s]
LREAL	Acceleration	Specify the maximum acceleration. [u/s^2]
LREAL	Deceleration	Specify the maximum deceleration. [u/s^2]
LREAL	Jerk	Specify the change rate of acceleration/deceleration. [u/s^3]
UINT	CoordSystem	Set the coordinate system type (1:MCS 2:PCS)
UINT	BufferMode	Specify the sequential operation setting of motion function block. (Refer to 6.1.4.BufferMode input)
UINT	TransitionMode	Unused
LREAL	TransitionParameter	Unused
Output		
BOOL	Done	Indicate whether to reach the specified position.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that whether or not motion function block is controlling the group.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted while it is running.

BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

(9) This motion function block issues Incremental position 3D circular interpolation command based on coordinate system on the axis group designated by AxesGroup input.

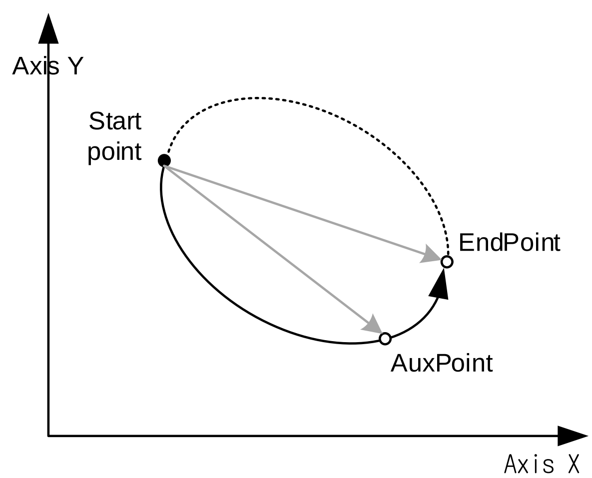
(10) When this motion function block starts, each axis performs circular trajectory interpolation control referring to the auxiliary point input, and the movement direction is determined by Path Choice input. If PathChoice input is set to 0, circular interpolation is operated in a clockwise direction, and if it is set to 1, circular interpolation is operated in a counter-clockwise direction.

(11) At AuxPoint and EndPoint input, designate the relative position of auxiliary points to refer to for circular interpolation.

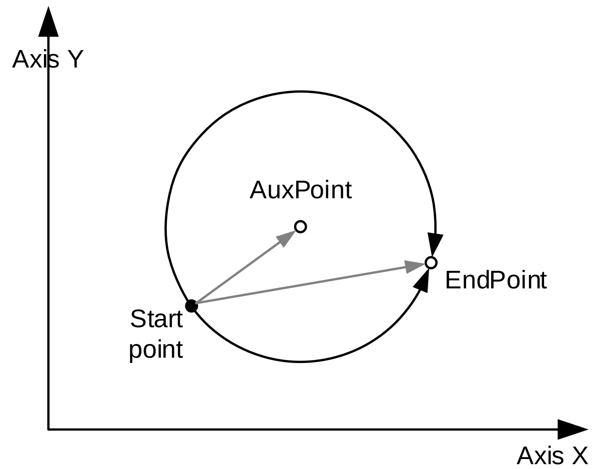
(12) In Velocity, Acceleration, Deceleration, Jerk inputs, the acceleration, deceleration, change rate of acceleration, velocity of the interpolation path are specified, respectively.

(13) Set the circular interpolation method in CircMode input. The circular interpolation methods corresponding to CircMode values are as follows.

(d) Circular interpolation of midpoint specifying method (BORDER, CircMode = 0) In this method, operation starts at the current position and it does circular interpolation through the pecified position of the central point to the target position. In the figure below, the current position corresponds to the coordinate system position at the start of the command, the midpoint corresponds to the coordinate input for the AuxPoint, and the target position corresponds to the relative coordinate input for the EndPoint.

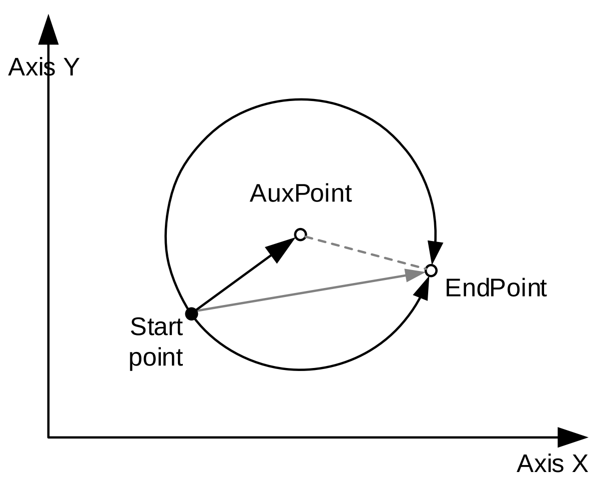


(e) Circular interpolation of central point specifying method in this method, operation starts at the current position, and it does circular interpolation to the target position along the circular path, which has a radius of the distance to the specified central position. In the figure below, the current position corresponds to the axes group coordinate at the start of the command, the center point corresponds to the coordinate input for the AuxPoint, and the target position corresponds to the relative coordinate input for the EndPoint.



(f) Circular interpolation with radius designation form

In this method, operation starts at the current position, and it does circular interpolation to the target position along the circular path which has a radius of the value specified in the radius. In the figure below, the coordinates of the axis group at the start of the command correspond to the current position, and the coordinates entered in the EndPoint are matched to the target point as incremental values. In the radius designation method, the vertical vector of the arc is set in AuxPoint during circular interpolation. The length of the vertical vector is set to the radius of the circle. When AuxPoint is set to (50,0,0), circular interpolation set to radius 50 is performed on the YZ plane.



(14) Refer to chapter 8.4.11 3D circular interpolation control in motion controller's manual for more details.

(15) If the function block is re-executed (Execute input is On) before the instruction is terminated, the changed parameters are applied.

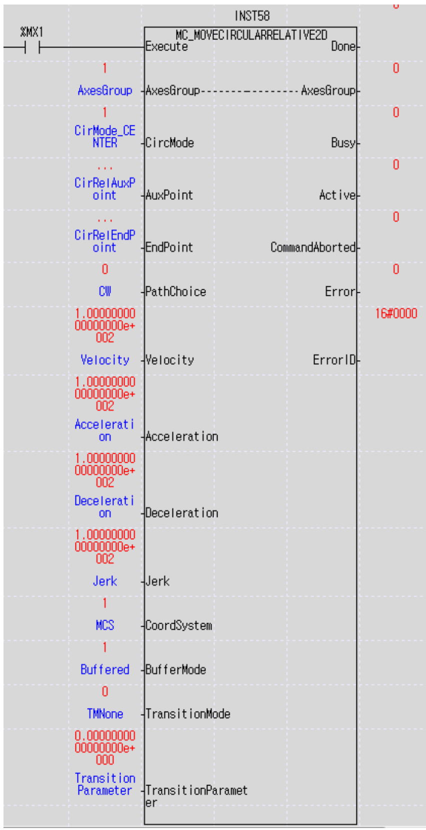
Only, Velocity, Acceleration, Deceleration, Jerk, AuxPoint, Endpoint input can be updated.

(16) Velocity input can be set to 0 or changed.

(17) Example program

This example is to set the center point specification method when the current command position is (1000, 1000) (set the relative position from the current position to the center point to set: 1000, 1000), and move clock-wise to perform circular interpolation to the target position (set the relative position from the current position to the target position: 0, 0).

(f) Function block setting

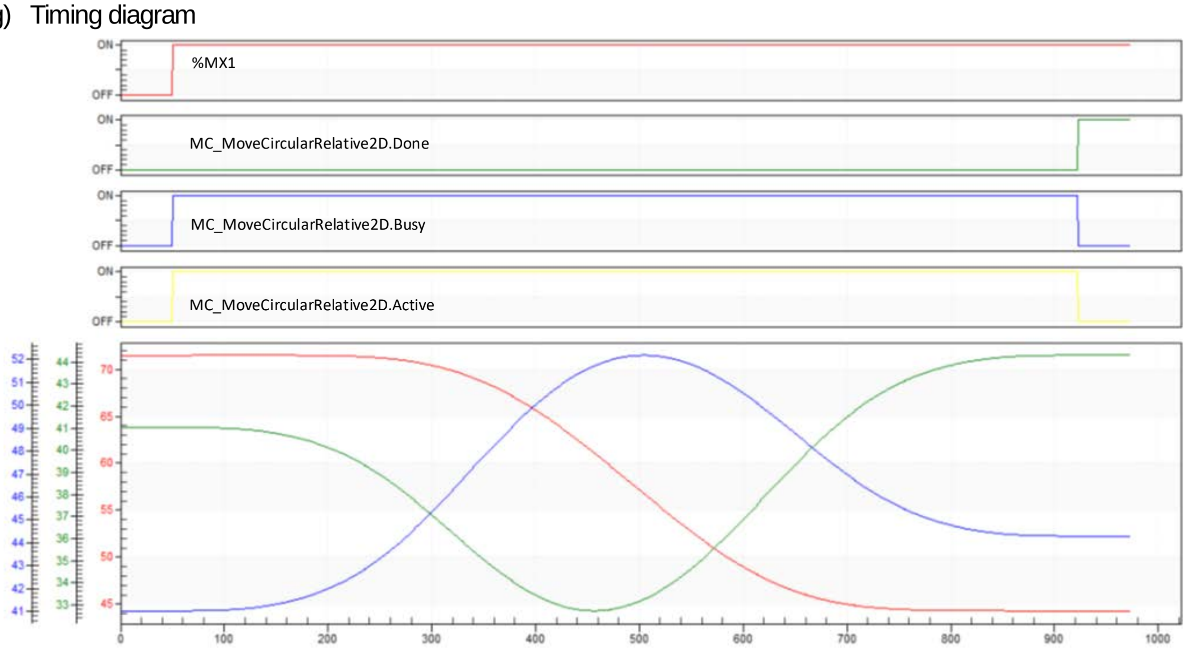


1	_AX01_CPOS	7.1485269050584748e+001	LREAL
2	_AX02_CPOS	4.1026455810822632e+001	LREAL
3	_AX03_CPOS	4.1026455810822632e+001	LREAL
4	⊟ CirRelAuxPoint		ARRAY[0..2] OF LREAL
5	CirRelAuxPoint[0]	0.0000000000000000e+000	LREAL
6	CirRelAuxPoint[1]	-7.5000000000000000e+001	LREAL
7	CirRelAuxPoint[2]	0.0000000000000000e+000	LREAL
8	⊟ CirRelEndPoint		ARRAY[0..2] OF LREAL
9	CirRelEndPoint[0]	0.0000000000000000e+000	LREAL
10	CirRelEndPoint[1]	-1.5000000000000000e+002	LREAL
11	CirRelEndPoint[2]	0.0000000000000000e+000	LREAL

Center point

End point

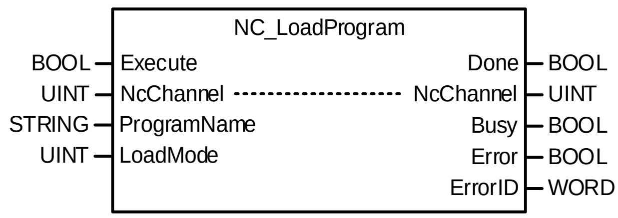
(g) Timing diagram



6.8 NC Control Function Block

6.8.1 Specify NC program (NC_LoadProgram)

Motion Function Block



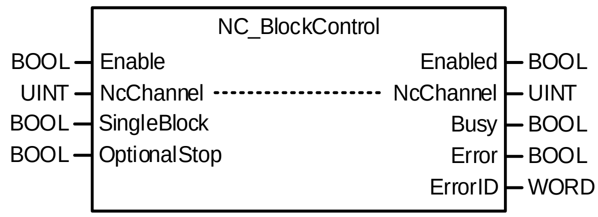
Input-Output		
UINT	NcChannel	Set the NC channel to make the command.
Input		
BOOL	Execute	Set programs to run in the rising Edge.
STRING	ProgramName	Set the name of the program to be executed.
UINT	LoadMode	Unused (Only'0'is settable.)
Output		
BOOL	Done	Indicate the state of motion function block completion.
BOOL	Busy	Indicate that execution of the function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the error number that occurred while the function block is running.

(1) This motion function block is the function block to specify the NC program to be executed when NC control is performed.

(2) When the program to be operated by the channel set in NC channel is set to ProgramName and the function block is executed, the program is designated as the one to be executed.

6.8.2 Specify block operation (NC_BlockControl)

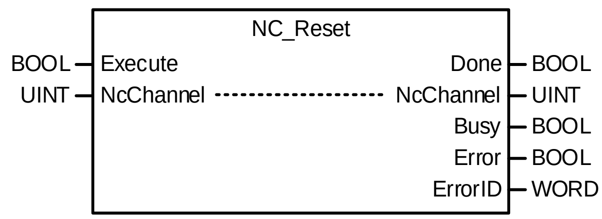
Motion Function Block



Input-Output		
UINT	NcChannel	Set the NC channel to make the command.
Input		
BOOL	Enable	While the input is enabled, the corresponding channel becomes the status of Single Block or Optional Stop.
BOOL	SingleBlock	Set the Single Block operation signal.
BOOL	OptionalStop	Set the Optional Stop operation signal.
Output		
BOOL	Enabled	Indicate the state of Block Control completion.
BOOL	Busy	Indicate that execution of the function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the error number that occurred while the function block is running.

(1) This motion function block determines the method to execute the program under the NC control.
(2) If SingleBlock is set to '1', NC_CycleStart executes one block at a time and stops after execution. If SingleBlock becomes '1' during the automatic operation and NC_BlockControl function block is executed, it will be stopped after terminating the currently executing block.
(3) If OptionalStop is set to '1', and M01 is commanded during the program, it will wait until NC_CycleStart function block is executed again.
(4) When both SingleBlock and OptionalStop are set to '1', SingleBlock setting is applied.

6.8.3 Reset (MC_Reset)

Motion Function Block		
		
Input-Output		
UINT	NcChannel	Set the NC channel to make the command.
Input		
BOOL	Execute	In the rising Edge, the NC is reset.
Output		
BOOL	Done	Indicate the state of motion function block completion.
BOOL	Busy	Indicate that execution of the function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the error number that occurred while the function block is running.

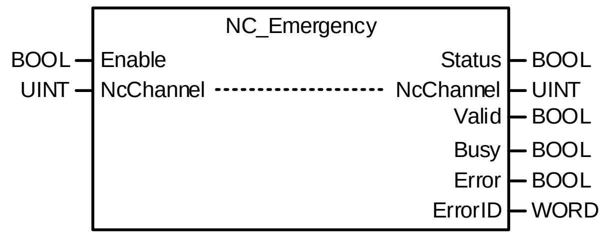
(1) This motion function block is to make the NC reset state under the NC control.
(2) If NC_Reset is executed during the automatic operation, it stops the automatic operation and changes into the reset state.
(3) The Reset state is as follows.

Item		Status
Setting Data	Offset Value	Hold
	Parameter	Hold
Various Data	Program in Memory	Hold
	Contents in the buffer storage	MDI: hold Cancel the rest
	Display of Sequence Number	Hold
	One shot G code	Cancel
	Modal G code	Hold
	F	Hold
	S, T, M	Hold
	K (Number of repeats)	Cancel
Work coordinate value		Hold
Action in operation	Movement	Cancel
	Dwell	Cancel
	Issuance of M, S, T code	Cancel
	Tool Length compensation	MDI: hold Parameter the rest
	Cutter compensation	MDI: hold Cancel the rest
	Storing called subprogram number	MDI: hold Cancel the rest

Item		Status
Output Signal	CNC Alarm signal AL	Extinguish if there is no cause for the alarm
	Reference position return completion LED	Hold Cancel(Emergency Stop)
	S, T, B Code	Hold
	M Code	Cancel
	M, S, T strobe signal	Cancel
	Spindle revolution signal(S analog signal)	Hold
	CNC ready signal MA	Hold
	Servo ready signal SA	ON
	Cycle Start LED	Cancel
	Feed hold LED	Cancel

6.8.4 Emergency stop (NC_Emergency)

Motion Function Block

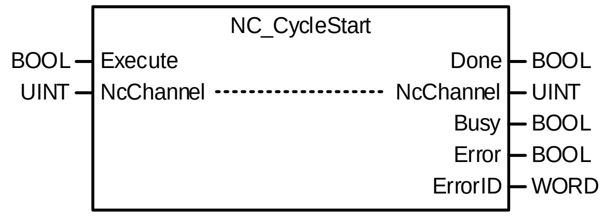


Input-Output		
UINT	NcChannel	Set the NC channel to make the command.
Input		
BOOL	Enable	The emergency stop is executed while the input is '1'.
Output		
BOOL	Status	Indicate the status of the emergency stop.
BOOL	Valid	Indicate the validity of the function block output. (Same as the Status output)
BOOL	Busy	Indicate that execution of the function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the error number that occurred while the function block is running.

(1) This motion function block is to execute the emergency stop on the corresponding NC channel under the NC control.
(2) If the emergency stop is executed, the current operation must be stopped immediately.

6.8.5 Start automatic operation (NC_CycleStart)

Motion Function Block

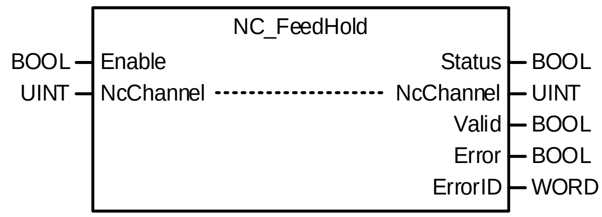


Input-Output		
UINT	NcChannel	Set the NC channel to make the command.
Input		
BOOL	Execute	Start the automatic operation in the rising Edge.
Output		
BOOL	Done	Indicate the state of motion function block completion.
BOOL	Busy	Indicate that execution of the function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the error number that occurred while the function block is running.

(1) This motion function block is to execute the automatic operation on the corresponding NC channel under the NC control.
(2) The program set in NC_LoadProgram is automatically operated.
(3) When the automatic operation is stopped due to M00, M01 (Optional Stop) and single block, the automatic operation is restarted.

6.8.6 Feed hold (NC_FeedHold)

Motion Function Block		

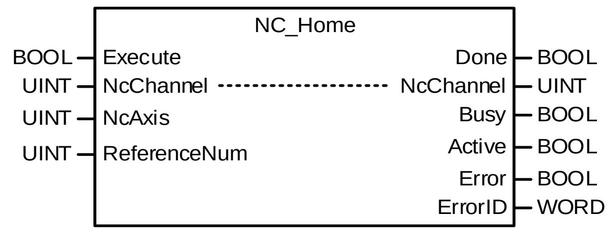


Input-Output		
UINT	NcChannel	Set the NC channel to make the command.
Input		
BOOL	Enable	The NC channel will be in Feed Hold status while the input is enabled.
Output		
BOOL	Status	Indicate the Feed Hold status.
BOOL	Valid	Indicate the validity of the function block output.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the error number that occurred while the function block is running.

(1) This motion function block is to make the Feed Hold command to the corresponding NC channel under the NC control.
(2) If the NC_FeedHold is executed during the automatic operation, the automatic operation is stopped.
(3) If the NC_CycleStart is performed during the execution of the NC_FeedHold command, the NC_CycleStart command is ignored.

6.8.7 NC Homing (NC_Home)

Motion Function Block



Input-Output		
UINT	NcChannel	Set the NC channel to make the command.
Input		
BOOL	Execute	Start the automatic operation in the rising Edge.
UINT	NcAxis	Set the channel axis. (1~10: X=1, Y=2, … B=8, C=9, S=10)
UINT	ReferenceNum	Select the origin type. 1~4: first origin ~ fourth origin
Output		
BOOL	Done	Indicate the state of motion function block completion.
BOOL	Busy	Indicate that execution of the function block is not completed.
BOOL	Active	Indicate that the current Function Block is controlling the axis.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the error number that occurred while the function block is running.

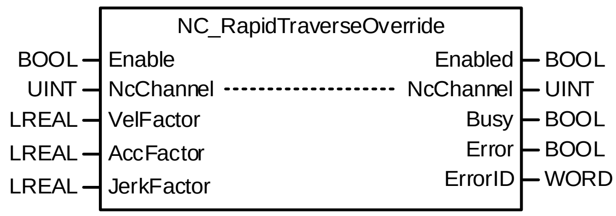
(1) This motion function block performs homing to the corresponding NC channel under the NC control.

(2) Homing to the 1st origin, 2nd origin, 3rd origin, and 4th origin is executed according to the values set in ReferenceNum. The origin coordinates can be set for each axis parameters of NC parameters in XG5000.

Group	Name	X Axis
Axis Settings	Command direction for the Modular Axis	0: Bidirectional
Home Settings	Position of 2nd home	0 mm
	Position of 3rd home	0 mm
	Position of 4rd home	0 mm

6.8.8 Rapid traverse override (NC_RapidTraverseOverride)

Motion Function Block

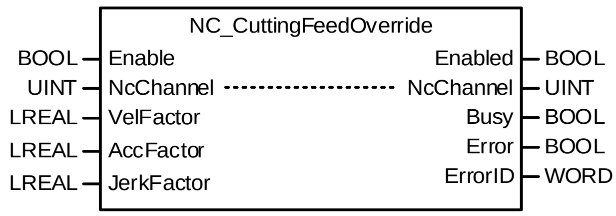


Input-Output		
UINT	NcChannel	Set the NC channel to make the command.
Input		
BOOL	Enable	Execute the Rapid Traverse Override operation on the channel while the input is enabled.
LREAL	VelFactor	Specify the override rate of the speed. (1.0=100%)
LREAL	AccFactor	Specify the override rate of acceleration/deceleration.
LREAL	JerkFactor	Specify the override ratio of the rate of change for acceleration.
Output		
BOOL	Enabled	Indicate that the override rate was applied successfully.
BOOL	Busy	Indicate that execution of the function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the error number that occurred while the function block is running.

(1) This motion function block makes the Rapid Traverse Override command for the corresponding NC channel under the NC control.
(2) Specify the speed override ratio for the VelFactor input. If the specified value is 0.0, the axis stops.
(3) The default value of each factor is 1.0, which means 100% of the command speed of the currently executing function block.
(4) Specify the acceleration / deceleration for the AccFactor input and the override rate of the jerk (rate of change of acceleration) for the JerkFactor input, respectively.
(5) Negative numbers cannot be entered into each factor.

6.8.9 Cutting feed override (NC_CuttingFeedOverride)

Motion Function Block

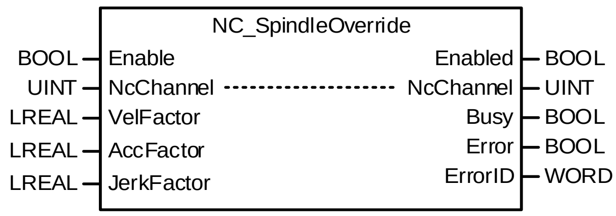


Input-Output		
UINT	NcChannel	Set the NC channel to make the command.
Input		
BOOL	Enable	Execute the Cutting Feed Override operation on the channel while the input is enabled.
LREAL	VelFactor	Specify the override rate of the speed. (1.0=100%)
LREAL	AccFactor	Specify the override rate of acceleration/deceleration.
LREAL	JerkFactor	Specify the override ratio of the rate of change for acceleration.
Output		
BOOL	Enabled	Indicate that the override rate was applied successfully.
BOOL	Busy	Indicate that execution of the function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the error number that occurred while the function block is running.

(1) This motion function block makes the Cutting Feed Override command for the corresponding NC channel under the NC control.
(2) Specify the speed override ratio for the VelFactor input. If the specified value is 0.0, the axis stops.
(3) The default value of each factor is 1.0, which means 100% of the command speed of the currently executing function block.
(4) Specify the acceleration / deceleration for the AccFactor input and the override rate of the jerk (rate of change of acceleration) for the JerkFactor input, respectively.
(5) Negative numbers cannot be entered into each factor.

6.8.10 Spindle override (NC_SpindleOverride)

Motion Function Block

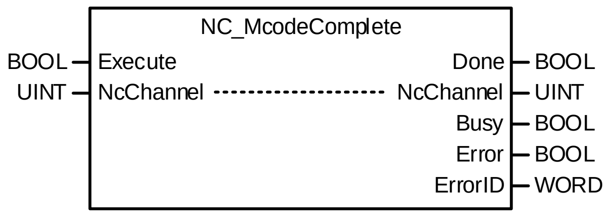


Input-Output		
UINT	NcChannel	Set the NC channel to make the command.
Input		
BOOL	Enable	Execute the Spindle Override operation on the channel while the input is enabled.
LREAL	VelFactor	Specify the override rate of the speed. (1.0=100%)
LREAL	AccFactor	Specify the override rate of acceleration/deceleration.
LREAL	JerkFactor	Specify the override ratio of the rate of change for acceleration.
Output		
BOOL	Enabled	Indicate that the override rate was applied successfully.
BOOL	Busy	Indicate that execution of the function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the error number that occurred while the function block is running.

(1) This motion function block makes the Spindle Override command for the corresponding NC channel under the NC control.
(2) Specify the speed override ratio for the VelFactor input. If the specified value is 0.0, the axis stops.
(3) The default value of each factor is 1.0, which means 100% of the command speed of the currently executing function block.
(4) Specify the acceleration / deceleration for the AccFactor input and the override rate of the jerk (rate of change of acceleration) for the JerkFactor input, respectively.
(5) Negative numbers cannot be entered into each factor.

6.8.11 M Code operation completed (NC_McodeComplete)

Motion Function Block

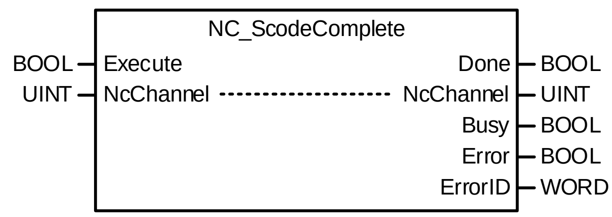


Input-Output		
UINT	NcChannel	Set the NC channel to make the command.
Input		
BOOL	Execute	Set the completion of the M Code operation on the corresponding the channel.
Output		
BOOL	Done	Indicate the state of motion function block completion.
BOOL	Busy	Indicate that execution of the function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the error number that occurred while the function block is running.

(1) This motion function block makes the completion command of the M Code operation for the corresponding NC channel under the NC control.
(2) It is the command to check the M code on the corresponding channel and set that the M code operation is completed.

6.8.12 S Code operation completed (NC_ScodeComplete)

Motion Function Block

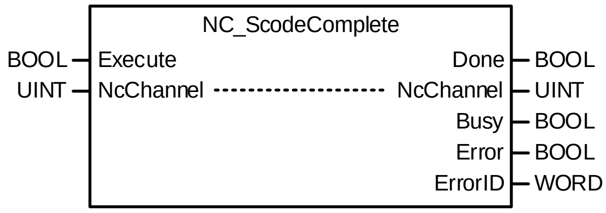


Input-Output		
UINT	NcChannel	Set the NC channel to make the command.
Input		
BOOL	Execute	Set the completion of the S Code operation on the corresponding the channel.
Output		
BOOL	Done	Indicate the state of motion function block completion.
BOOL	Busy	Indicate that execution of the function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the error number that occurred while the function block is running.

(1) This motion function block makes the completion command of the S Code operation for the corresponding NC channel under the NC control.
(2) It is the command to check the S code on the corresponding channel and set that the S code operation is completed.

6.8.13 T Code operation completed (NC_TcodeComplete)

Motion Function Block



Input-Output		
UINT	NcChannel	Set the NC channel to make the command.
Input		
BOOL	Execute	Set the completion of the T Code operation on the corresponding the channel.
Output		
BOOL	Done	Indicate the state of motion function block completion.
BOOL	Busy	Indicate that execution of the function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the error number that occurred while the function block is running.

(1) This motion function block makes the completion command of the T Code operation for the corresponding NC channel under the NC control.
(2) It is the command to check the T code on the corresponding channel and set that the T code operation is completed.

6.8.14 Read NC parameter (NC_ReadParameter)

Motion Function Block		

```
                    NC_ReadParameter
BOOL — Enable                                Valid — BOOL
UINT — NcChannel -------------------- NcChannel — UINT
UINT — NcAxis                                 Busy — BOOL
 INT — ParameterGroup                        Error — BOOL
 INT — ParameterNumber                     ErrorID — WORD
                                             Value — LREAL
```

Input-Output		
UINT	NcChannel	Set the NC channel to make the command.
Input		
BOOL	Enable	The relevant parameters are output while the input is enabled.
UINT	NcAxis	Set the channel axis. (1~10: X=1, Y=2, … B=8, C=9, S=10) If it is set to 0, 'Read Channel Parameters' will be executed.
INT	ParameterGroup	Specify the group of the parameters to read.
INT	ParameterNumber	Specify the group number of the parameters to read.
Output		
BOOL	Vaild	Indicate the validity of the function block output.
BOOL	Busy	Indicate that execution of the function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the error number that occurred while the function block is running.
LREAL	Value	Output the values of the parameters.

(1) This motion function block is to read and output the parameters of the channel and channel / axis of the corresponding channel.
(2) While the Enable input is active, the values of the relevant parameters are output continuously.
(3) ParameterGroup input specifies the parameter group number to read.
(4) ParameterNumber input specifies the number in the group of the parameters to be read.
(5) The group number and the number in the group of each parameter are as follows.

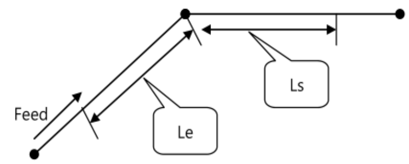
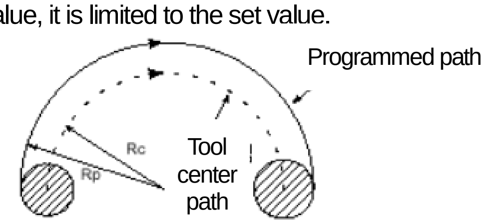
Parameter	Group	No.	Item	Description
1. Channel Parameter	1. Standard settings	1	Target machining quantity	Set the target machining quantity. (0 ~ 2,147,483,647) It is possible to write NC parameters during NC channel automatic operation.
		2	Target machining quantity at M99 repeated machining	Set the target machining quantity for repeated machining with M99. If the set value matches the current machining quantity, the cycle automatically stops.(0 ~ 2,147,483,647) It is possible to write NC parameters during NC channel automatic operation.
		3	Check of decimal point	Set whether to check decimal point of the NC program. 0: Decimal point check (Mm if there is a decimal point, um if there is no decimal point) 1: No decimal point check (mm)
		4	Keep workpiece coordinate system	Set whether to keep the workpiece coordinate system when resetting. 0:Hold 1: Do not hold

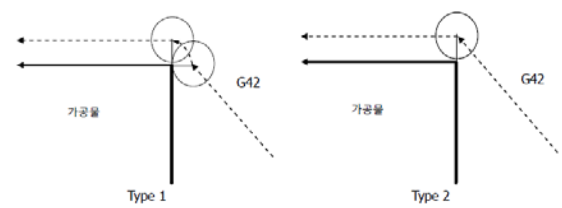
Parameter	Group	No.	Item	Description
1. Channel Parameter	1. Standard settings	5	Whether to call the macro when the T code is commanded	Set whether to call the macro program (9000.nc ~ 9009.nc) when the T code is commanded. 0: Do not call 1: Call
		6	Dwell Method	Set the dwell function (G04) to use the data corresponding to X, P as time or the number of revolutions of the spindle. If the data is set to the number of revolutions of the spindle, it is applied in the status of feed per revolution (G95). 0: times 1: Rotation number
		7	Select a progress block at reset	Set whether to initialize to the start block of the program at reset. ※If you want to set to 0 (keep the current block), the parameters of "Keep workpiece coordinate system" should be set to 0 (keep). 0: Keep the current block 1: Initialize to the start block of the main program 2: Initialize to the current block of the main program It is possible to write NC parameters during NC channel automatic operation.
		8	Whether or not to search the Statement Number	The number of buffers that can store the program's Statement Number (N__) is limited to 1,000 in the system. This buffer is needed if the program changes the sequence using a GOTO statement. If more than 1,000 blocks have the N__ command, an alarm will occur. This parameter is used to input whether or not to execute such Statement Number search. Because high- capacity CAM programs do not have GOTO using the Statement Number and in the majority of cases, there are more than 1,000 Statement Numbers, you should set this parameter as 1. 0: Search 1: Do not search
		12	Minimum command unit	When decimal point check is applied, set the minimum unit of the commanded value. (0 ~ 0.999mm)
		18	Whether to use G22 No Travelling Area	0: 'No Travelling Area' is valid. 1: 'No Travelling Area' is invalid. It is possible to write NC parameters during NC channel automatic operation.
		19	Set the inner/outer side of G22 No Travelling Area	0: Inside 1: Outside
		20	Whether to use the 3rd 'No Travelling Area'	0: 'No Travelling Area' is valid. 1: 'No Travelling Area' is invalid. It is possible to write NC parameters during NC channel automatic operation.
		21	Reverse buffer operation size	Set the buffer size required for NC reverse operation. It is possible to write NC parameters during NC channel automatic operation.

Parameter	Group	No.	Item	Description
1. Channel Parameter	1. Standard settings	22	Rotary axis of Cylindrical interpolation	In the cylindrical interpolation mode, the axis maps the axis of rotation during the circular interpolation. The axes are X, Y, Z and perform the circular interpolation by mapping the axis of rotation to the selected axis. For example, if the axis of rotation is mapped to the X axis under the state of the XY plane (G17), the width becomes the axis of rotation and the height becomes Y axis. When ZX (G18) is selected as the plane, the width becomes the Z axis and the height becomes the axis of rotation. However, if you set the plane to YZ (G19), you cannot perform the circular interpolation on the commanded axis of rotation. 0: X axis, 1: Y axis, 2: Z axis
		23	Linear axis for interpolating the polar coordinate	0: Unused 1: X, 2: Y, 3: Z, 4: A, 5: B, 6: C, 7: U, 8: V, 9: W
		24	Rotary axis for interpolating the polar coordinate	0: Unused 1: X, 2: Y, 3: Z, 4: A, 5: B, 6: C, 7: U, 8: V, 9: W
		25	Main spindle axis number	Set the number of an axis to be used as the main spindle axis in the NC channel. Set the system that does not use the spindle axis to 0. To automatically control spindle commands in the NC function module, set it exactly the same as the axis number connected to the NC S axis. 0: Unused 1 ~ 36: Axis 1 ~ Axis 36
		33	Monitoring time for in-position completion	0 ~ 65,535ms It is possible to write NC parameters during NC channel automatic operation.
		34	The spindle operation of the spindle axis. How to treat M/S-code[Note1)]	0: Automatic operation continues 1: Operation continues after completing commands It is possible to write NC parameters during NC channel automatic operation.
	2. Circular milling setting Setting	1	Regenerate the circular center when the circular alarm occurs	Set whether to recreate the central point of the arc without generating an arc alarm when the distance between the start point and the end point exceeds the tolerance of the difference between the two radii under I, J, K circular commands. 0: An alarm occurs. 1: The central point of the arc is regenerated.
		2	Speed-limiting function for the circular milling ON/OFF	0: Unused 1: used It is possible to write NC parameters during NC channel automatic operation.
		3	Tolerance of arc radius	Set the tolerance of the difference between the two radii at the start point and the end point under the circular arc command. If this value is large, the accuracy of the end part of the arc may be degraded. When set to 0, it is recognized as 0.001. (0~ 1 unit, real number) It is possible to write NC parameters during NC channel automatic operation.
		5	Circular radius with the speed-limiting function for the arc machining	(0 ~ 10,000 unit, real number) It is possible to write NC parameters during NC channel automatic operation.

Parameter	Group	No.	Item	Description
		6	Upper cutting speed limit of the circular milling	The maximum speed is limited to the set value for the circular arc below "Circular radius with the speed-limiting function for the circular milling". It is possible to write NC parameters during NC channel automatic operation. (0 ~ 10,000 unit/min, real number)
		7	Lower cutting speed limit of the circular milling	If "Speed-limiting function for the circular milling ON/OFF" is set to ON, the cutting speed is limited to the set value or more. (0 ~ 10,000 unit/min, real number) It is possible to write NC parameters during NC channel automatic operation.
		9	Circular milling acceleration	Set the acceleration at the circular milling. It is possible to write NC parameters during NC channel automatic operation.
		10	Circular milling deceleration	Set the deceleration at the circular milling. It is possible to write NC parameters during NC channel automatic operation.
		11	Circular milling jerk	Set the jerk at the circular milling. It is possible to write NC parameters during NC channel automatic operation.

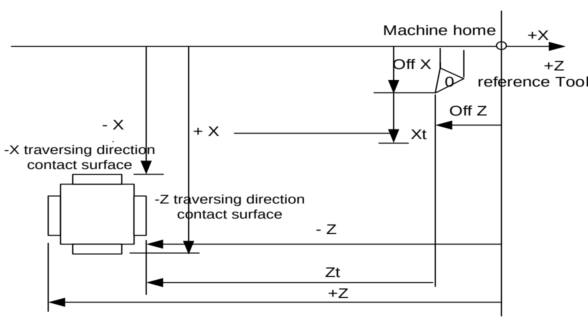
Parameter	Group	No.	Item	Description
1. Channel Parameter	3. Cutting feed setting	1	Set the upper speed limit of the cutting feed	If the cutting speed exceeding the set value is commanded, the cutting speed is limited to the set value and an alarm occurs. (0 ~ 100,000 unit/min, real number) It is possible to write NC parameters during NC channel automatic operation.
		2	Set the lower speed limit of the cutting feed	It is applied only when the cutting speed is not commanded in the feed mode per minute. (0 ~ 100,000 unit/min, real number) It is possible to write NC parameters during NC channel automatic operation.
		4	How to Accelerate/Decelerate Interpolation Operation	1: Acceleration / deceleration before interpolation
		5	Corner speed limit function mode setting	0: Unused 1: Angle difference mode 2: Speed difference mode It is possible to write NC parameters during NC channel automatic operation.
		6	Acceleration and deceleration type of the cutting feed	0: Line type 1: S type
		7	Operating method of the continuous blocks for acceleration / deceleration before interpolation	When executing the consecutive blocks, it creates the connecting trajectory that draws an arc on the corner of the connecting trajectory with the speed set with the next block. 1: When it is set to Buffered, the circular arc is not inserted. 1: Buffered 2: Blending Low 3: Blending Previous 4: Blending Next 5: Blending High
		9	Acceleration at the time of cutting feed (before interpolation)	Acceleration at the time of cutting feed It is possible to write NC parameters during NC channel automatic operation.
		10	Deceleration at the time of cutting feed (before interpolation)	Deceleration at the time of cutting feed It is possible to write NC parameters during NC channel automatic operation.
		11	Jerk at the time of cutting feed (before interpolation)	Jerk at the time of cutting feed It is possible to write NC parameters during NC channel automatic operation.
		18	Allowable angle setting (angle mode)	Set the allowable angle in angle mode. It is possible to write NC parameters during NC channel automatic operation.
		19	Allowable speed difference setting (speed difference mode)	Set the allowable speed difference in speed difference mode. It is possible to write NC parameters during NC channel automatic operation.
		20	Deceleration speed setting	Setting the Acc. speed. It is possible to write NC parameters during NC channel automatic operation.
		21	Speed multiplier with respect to the linear axis of the axis of rotation(speed difference mode)	Set the speed multiplier for the linear axis of the rotary axis in speed difference mode. It is possible to write NC parameters during NC channel automatic operation.
		22	Cutting feed acceleration and deceleration time constant	Set the time constant for acceleration/deceleration after interpolation in cutting feed.

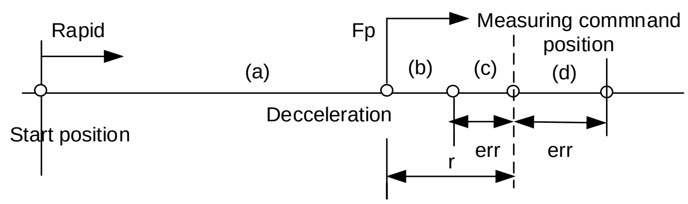
Parameter	Group	No.	Item	Description
			(acceleration and deceleration after interpolation)	
	4. Rapid traverse setting	1	Feed rate of rapid traverse (G00) block in dry run mode	In dry run mode, set the feed rate of the G00 block to the manual feed rate or the rapid feed rate. 0: Dry run speed 1: Rapid feed rate
	7. Automatic corner override	1	Start section of automatic corner override setting	Sets the override area (Ls) of the starting block that forms the corner when an automatic corner override command is commanded.  (0 ~ 999.999 mm, 0 ~ 99.999 inch, real number)
		2	End section of automatic corner override setting	Sets the override area (Le) of the ending block that forms the corner when an automatic corner override command is commanded. (0 ~ 999.999 mm, 0 ~ 99.999 inch, real number)
		3	Scale of Auto Corner Override setting	Set the multiplier applied to the feed rate in the automatic corner override section. ※ Auto Corner Override Feed rate = Current Command Feed rate x Multiplier x 0.01 (10~100%, real number)
		5	The angle between the automatic corner override settings	Sets the maximum angle of a corner that determines which blocks apply automatic corner override. (-999.999 ~ 999.999 deg, real number)
		6	The type of automatic corner override settings	Sets the application type of automatic corner override. If you set Always Apply, the automatic corner override function is applied by judging the angle of the corner in every block. 0: Applies only to inner corners for tool radius compensation 1: always applied
		7	Minimum rate of internal circular cutting speed	Set the minimum rate of internal circular cutting speed. The internal circular cutting speed is determined by Rc/Rp * F during circular arc feed during compensation. If Rc/Rp is less than or equal to the set value, it is limited to the set value.  (0.01~1.00%, real number)
		8	Automatic corner override velocity	This is an item to directly input the speed of the deceleration section of automatic corner override. If the value is 0, the value obtained by multiplying the current cutting speed (Feed) by the multiplier of the automatic corner override setting is applied as an override. If the command speed is smaller than this value, it is fed at the command speed. (0 ~ 99999.999 mm/min, 0 ~ 9999.999 inch/min, real number)

Parameter	Group	No.	Item	Description
	8. Tool radius Compensation	129	How to apply the compensation value of the tool diameter	Set the method of applying the compensation amount of the tool diameter when compensating the tool diameter. 0: Apply the diameter value 1: Apply the radius value
		130	Compensation type of the tool diameter	Tool diameter Sets the type of traversing method at the beginning and end of the calibration.  0: Type 1(Bypass traverse) 1: Type 2(Direct traverse)
		131	Whether to check the tool interference during tool diameter compensation	Set whether to check the tool interference during tool diameter compensation 0: Do not check 1: Check
		1	Compensation amount of the tool diameter 1	Compensation amount 1 to be used to compensate the tool diameter
		……	……	……
		128	Compensation amount of the tool diameter 128	Compensation amount 128 to be used to compensate the tool diameter
	9. Tool length Compensation	1	Compensation amount 1 of the tool length	Compensation amount 1 to be used to compensate the tool length
		……	……	……
		128	Compensation amount 128 of the tool length	Compensation amount 128 to be used to compensate the tool length
	10. Work-piece Coordinate System	1	Whether to use the workpiece coordinate system shift amount.	Set whether to use the workpiece coordinate system shift amount. 0: Unused 0: Used

Parameter	Group	No.	Item	Description
1. Channel Parameter	10. Work-piece Coordinate System	11	Workpiece coordinate system shift amount 1	Set the workpiece coordinate system shift amount for the X axis.
		……	……	Set the workpiece coordinate system shift amount for the 7 axes; Y, Z, A, B, C, U, V.
		19	Workpiece coordinate system shift amount 9	Set the workpiece coordinate system shift amount for the W axis.
		41	G54 workpiece coordinate system value 1	Set the G54 workpiece coordinate system value for the X axis.
		……	……	Set the G54 workpiece coordinate system values for the 7 axes; Y, Z, A, B, C, U, V.
		49	G54 workpiece coordinate system value 9	Set the G54 workpiece coordinate system value for the W axis.
		51	G55 workpiece coordinate system value 1	Set the G55 workpiece coordinate system value for the X axis.
		……	……	Set the G55 workpiece coordinate system values for the 7 axes; Y, Z, A, B, C, U, V.
		59	G55 workpiece coordinate	Set the G55 workpiece coordinate system value for

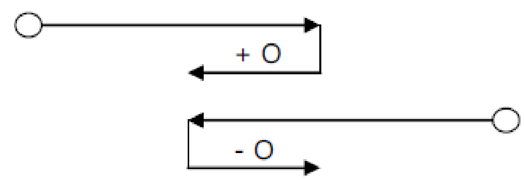
Parameter	Group	No.	Item	Description
			system value 9	the W axis.
		61	G56 workpiece coordinate system value 1	Set the G56 workpiece coordinate system value for the X axis.
		……	……	Set the G56 workpiece coordinate system values for the 7 axes; Y, Z, A, B, C, U, V.
		69	G56 workpiece coordinate system value 9	Set the G56 workpiece coordinate system value for the W axis.
		71	G57 workpiece coordinate system value 1	Set the G57 workpiece coordinate system value for the X axis.
		……	……	Set the G57 workpiece coordinate system values for the 7 axes; Y, Z, A, B, C, U, V.
		79	G57 workpiece coordinate system value 9	Set the G57 workpiece coordinate system value for the W axis.
		81	G58 workpiece coordinate system value 1	Set the G58 workpiece coordinate system value for the X axis.
		……	……	Set the G58 workpiece coordinate system values for the 7 axes; Y, Z, A, B, C, U, V.
		89	G58 workpiece coordinate system value 9	Set the G58 workpiece coordinate system value for the W axis.
		91	G59 workpiece coordinate system value 1	Set the G59 workpiece coordinate system value for the X axis.
		……	……	Set the G59 workpiece coordinate system values for the 7 axes; Y, Z, A, B, C, U, V.
		99	G59 workpiece coordinate system value 9	Set the G59 workpiece coordinate system value for the W axis.
	11. Macro Program	1	Whether to apply the single block stop function to the macro program	Set whether to apply the single block stop function to the macro program(9000.nc ~ 9999.nc) 0: Stop 1: Do not stop It is possible to write NC parameters during NC channel automatic operation.
		2	Display the macro program block	Set whether to display the progress status of the block on the screen when operating the macro program (9000.nc ~ 9999.nc). 0: Not display 1: Display It is possible to write NC parameters during NC channel automatic operation.
		10	Macro program call G code (9010.nc)	Set the G code number to call the macro program (9010.nc ~ 9019.nc) that can be called by the G code. The setting values 0, 1, 2, 3 are ignored. (0~255.9, real number)
		……	……	
		19	Macro program call G code (9019.nc)	Set the G code number to call the macro program (9010.nc ~ 9019.nc) that can be called by the G code. ※The setting values 0, 1, 2, 3 are ignored. (0~255.9, real number)

Parameter	Group	No.	Item	Description
1. Channel Parameter	11. Macro Program	20	Macro program call M code (9020.nc)	Assign the M code number to call the macro program (9020.nc ~ (9020.nc ~ 9029.nc) with the M code. ※ 0, 30 of the input values are ignored. (0~255, integer)
		……	……	
		29	Macro program call M code (9029.nc)	Assign the M code number to call the macro program (9020.nc ~ (9020.nc ~ 9029.nc) with the M code. 0, 30 of the input values are ignored. (0~255, integer)
		9	T code call macro program number	Enter the number of the macro program (9000.nc ~ 9009.nc) to be called when the T code is commanded. (9000~9009, integer) It is possible to write NC parameters during NC channel automatic operation.
	13. Automatic tool offsets	1	+Measurement reference distance X for automatic tool offset	Set the measurement reference distance from the automatic tool offset to the contact surface in the +traversing direction of X.  (-99999.999 ~ 99999.999 mm, -9999.999 ~ 9999.999 inch, real number)
		2	+Measurement reference distance Y for automatic tool offset	Set the measurement reference distance from the automatic tool offset to the contact surface in the +traversing direction of Y. (-99999.999 ~ 99999.999 mm, -9999.999 ~ 9999.999 inch, real number)
		3	+Measurement reference distance Z for automatic tool offset	Set the measurement reference distance from the automatic tool offset to the contact surface in the +traversing direction of Z. (-99999.999 ~ 99999.999 mm, -9999.999 ~ 9999.999 inch, real number)
		4	- Measurement reference distance X for automatic tool offset	Set the measurement reference distance from the automatic tool offset to the contact surface in the- traversing direction of X. (-99999.999 ~ 99999.999 mm, -9999.999 ~ 9999.999 inch, real number)
		5	- Measurement reference distance Y for automatic tool offset	Set the measurement reference distance from the automatic tool offset to the contact surface in the -traversing direction of Y. (-99999.999 ~ 99999.999 mm, -9999.999 ~ 9999.999 inch, real number)
		6	- Measurement	Set the measurement reference distance from the automatic tool offset to

Parameter	Group	No.	Item	Description
			reference distance Z for automatic tool offset	the contact surface in the -traversing direction of Z. (-99999.999 ~ 99999.999 mm, -9999.999 ~ 9999.999 inch, real number)
		7	Automatic tool deceleration start length of automatic tool offset	Sets the application section (r) of deceleration speed for automatic tool length measurement.  (0 ~ 99.999 mm, 0 ~ 9.999 inch, real number)
		8	Automatic tool measurement detection range of automatic tool offset	Sets the detection range (err) for automatic tool length measurement. (0 ~ 99.999 mm, 0 ~ 9.999 inch, real number)
		9	Automatic tool deceleration velocity of automatic tool offset	Set the automatic tool length measurement deceleration speed (Fp) (0.001 ~ 999.999 mm, 0.0001 ~ 99.9999 inch, real number)
	14. Default Setting	1	Modal traverse of default settings	If there is no G00 or G01, select the G code to be applied as the default modal. 0: Rapid Feed(G00) 1: Cutting Feed(G01)
		2	Modal plane of default settings	If there is no G code instruction for G17, G18, G19 group, select the G code to be applied as the default modal. 0: XY plane(G17) 1: XZ plane(G18) 2: YZ plane(G19)
		3	Modal absolute / increment with default settings	If there is no G code instruction for G90, G91 group, select the G code to be applied as the default modal. 0: Absolute command (G90) 1: Incremental command(G91)
		4	Modal inch / metric with default settings	0: G20 1: G21
		5	Check the modal prohibited area with default settings	If there is no G code instruction for G22, G23 group, select the G code to be applied as the default modal. 0: Stroke On(G22) 1: Stroke off(G23)
	15. Spindle Setting	4	A reference axis when controlling constant surface speed[Note1)]	Set the reference axis that operates in connection with a spindle when controlling constant surface speed. 0: Unused 1: X, 2: Y, 3: Z, 4: A, 5: B, 6: C, 7: U, 8: V, 9: W

Parameter	Group	No.	Item	Description
				It is possible to write NC parameters during NC channel automatic operation.
		5	The maximum number of spindle rotation when controlling constant surface speed[Note1)]	Set the maximum number of spindle rotation when controlling constant surface speed. When being commanded by the S code of G92 (set the maximum speed of the master axis), the S code data is saved as this parameter value. (0 ~ 100,000, real number) It is possible to write NC parameters during NC channel automatic operation.
		6	The minimum number of spindle rotation when controlling constant surface speedNote1)	Set the minimum number of spindle rotation when controlling constant surface speed. (0 ~ 100,000, real number) It is possible to write NC parameters during NC channel automatic operation.
	16. Relative Setting	1	Relative coordinate's offset value #1	Set the relative coordinate's offset value for the X axis.
		2	Relative coordinate's offset value #2	Set the relative coordinate's offset value for the Y axis.
		3	Relative coordinate's offset value #3	Set the relative coordinate's offset value for the Z axis.
		4	Relative coordinate's offset value #4	Set the relative coordinate's offset value for the A axis.
		5	Relative coordinate's offset value #5	Set the relative coordinate's offset value for the B axis.
		6	Relative coordinate's offset value #6	Set the relative coordinate's offset value for the C axis.
		7	Relative coordinate's offset value #7	Set the relative coordinate's offset value for the U axis.
		8	Relative coordinate's offset value #8	Set the relative coordinate's offset value for the V axis.
		9	Relative coordinate's offset value #9	Set the relative coordinate's offset value for the W axis.

Parameter	Group	No.	Item	Description
2. Channel/Axis Parameter	1. Axis setting	2	Setting the direction for the modular axis	Set the traverse command for the axis set as the modular axis. 0: Unidirectional 1: Bidirectional
	2. Home	1	Coordinates of the 2nd origin	Set the coordinates of the 2nd origin.
		2	Coordinates of the 3rd origin	Set the coordinates of the 2rd origin.
		3	Coordinates of the 4th origin	Set the coordinates of the 4th origin.
	3. Rapid feed	2	Rapid feed acceleration	The set value is used as the acceleration of the G00 block. It is possible to write NC parameters during NC channel automatic operation.
		3	Rapid feed deceleration	The set value is used as the deceleration of the G00 block. It is possible to write NC parameters during NC channel automatic operation.
		4	Rapid feed jerk	The set value is used as the jerk of the G00 block. It is possible to write NC parameters during NC channel automatic operation.
		5	Rapid feed speed	The set value is used as the feed speed of the G00 block. (0~100000 unit/min, real number) It is possible to write NC parameters during NC channel automatic operation.
		6	Dry run speed	Used at dry run speed It is possible to write NC parameters during NC channel automatic operation.
	4. Feed area	1	Minimum value of the G22 Feed Prohibited Area range for the X, Y, and Z axis.	Set the minimum value of the G22 Feed Prohibited Area range for the X, Y, and Z axis. (-100,000~100,000 unit, real number) It is possible to write NC parameters during NC channel automatic operation.
		2	Maximum value of the G22 Feed Prohibited Area range for the X, Y, and Z axis.	Set the maximum value of the G22 Feed Prohibited Area range for the X, Y, and Z axis. (-100,000~100,000 unit, real number) It is possible to write NC parameters during NC channel automatic operation.
		3	Minimum value of the 3rd Feed Prohibited Area range for the X, Y, and Z axis.	Set the minimum value of the 3rd Feed Prohibited Area range for the X, Y, and Z axis. (-100,000~100,000 unit, real number) It is possible to write NC parameters during NC channel automatic operation.
		4	Maximum value of the 3rd Feed Prohibited Area range for the X, Y, and Z axis.	Set the maximum value of the 3rd Feed Prohibited Area range for the X, Y, and Z axis. (-100,000~100,000 unit, real number) It is possible to write NC parameters during NC channel automatic operation.
	5. Sub setting	1	Scaling setting	When using the scaling function (G51) function, this parameter is applied to the axis without the scaling

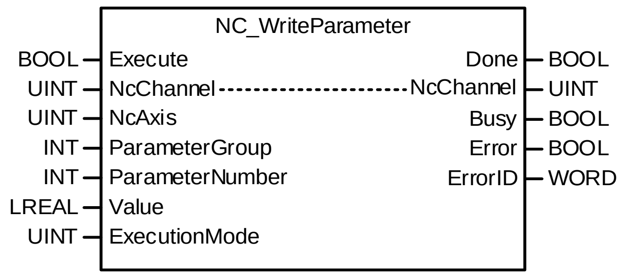
Parameter	Group	No.	Item	Description
				amount command. When set to a negative value, it is performed as a mirror image function. X, Y, Z, A, B, C, U, V, W, S 9 axis values setting (-9999~9999, integer)
		2	Overrun feed rare of single direction positioning	Set the overrun feed rate of the 9 axes; X, Y, Z, A, B, C, U, V, W when using the single direction positioning function (G60). After stopping at the position separated by the set value for the G60 command block's axis, it moves to the command position to eliminate the effect of backlash.  (-100 ~ 100 unit, real number)

Notes

Note 1) Available to use it on O/S V1.30 or greater of a single motion controller

6.8.15 Write NC parameter (MC_WriteParameter)

Motion Function Block

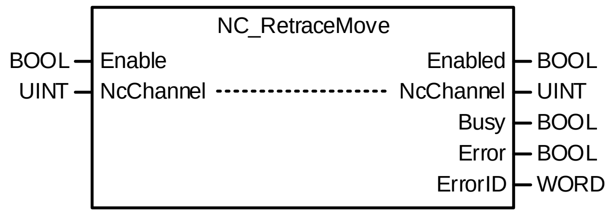


Input-Output		
UINT	NcChannel	Set the NC channel to make the command.
Input		
BOOL	Execute	The NC parameter is written in the rising Edge of the input.
UINT	NcAxis	Set the channel axis. (1~10: X=1, Y=2, … B=8, C=9, S=10) When set to 0, 'Write Channel Parameters' is executed.
INT	ParameterGroup	Specify the group of the parameter to be written.
INT	ParameterNumber	Specify the number in the group of the parameter to be written.
LREAL	Value	Specify the value of parameter to write.
UINT	ExecutionMode	Rsvd.
Output		
BOOL	Vaild	Indicate the validity of the function block output.
BOOL	Busy	Indicate that execution of the function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the error number that occurred while the function block is running.

(1) This motion function block is the function block that writes the values specified in the parameters of the NC channel and channels/axes.
(2) The parameters will be written in the rising edge of the Execute input.
(3) ParameterGroup input specifies the group number of the parameter to be written.
(4) ParameterNumber input specifies the number in the group of the parameter to be written. If the value that cannot be set is applied, "Error 16 # 000B" occurs.
(5) In the Value input, specify the value to be written in the parameter.
(6) For the group number and the number in the group of each parameter, refer to 6.8.14 Reading NC parameters.
(7) Some items of all parameters can be written when automatic operation is performed, that is, NC_CycleStart is performed and automatic operation is in progress. For this, refer to 6.8.14 Reading NC Parameter as well.

6.8.16 Reverse Operation (NC_RetraceMove)

Motion Function Block

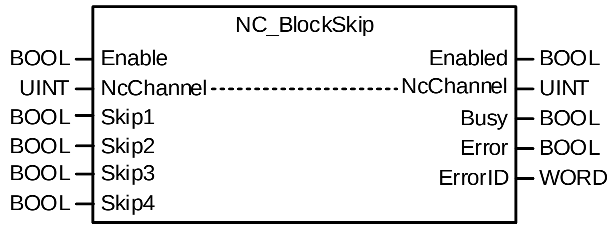


Input-Output		
UINT	NcChannel	Specify the NC channel to make commands (1 ~ 4 : Channel 1 ~ Channel 4)
Input		
BOOL	Enable	Execute the reverse operation command on the relevant channel while the input is enabled.
Output		
BOOL	Enabled	Indicate that the function block is successfully applied.
BOOL	Busy	Indicate that execution of the function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the error number that occurred while the function block is running.

(1) This motion function block is the function block that makes the reverse operation command on the relevant NC channel.
(2) While the Enable input is active, execute operation in the reverse direction.
(3) The reverse operation is possible only for G00, G01, G02 and G03 blocks.
(4) To use the function block, the NC channel parameter setting must set the 'reverse driving buffer size'.

6.8.17 Block skip (NC_BlockSkip)

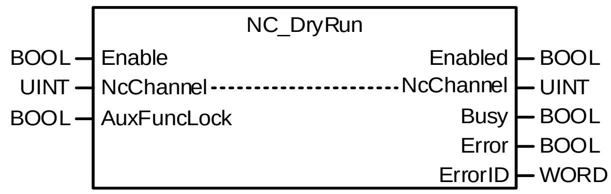
Motion Function Block



Input-Output		
UINT	NcChannel	Specify the NC channel to make commands (1 ~ 4 : Channel 1 ~ Channel 4)
Input		
BOOL	Enable	While an input is enabled, execute the block skip operation set in the relevant channel.
BOOL	Skip1	Set block skip 1. (G31/G31.1/G37/G37.1)
BOOL	Skip2	Set block skip 2. (G31.2/G37.2)
BOOL	Skip3	Set block skip 3. (G31.3/G37.3)
BOOL	Skip4	Set block skip 4. (G31.4/G37.4)
Output		
BOOL	Enabled	Indicate that the function block is successfully applied.
BOOL	Busy	Indicate that execution of the function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the error number that occurred while the function block is running.

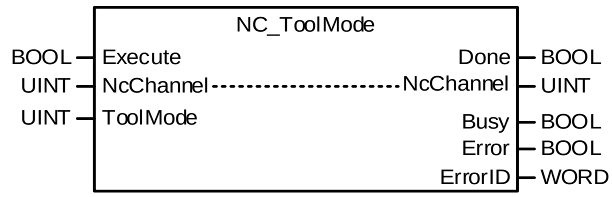
(1) This motion function block is the function block that the relevant NC channel gives commands such as block skip or automatic measurement of tool length.
(2) While the Enable input is active, skip blocks such as Skip1 (G31/G31.1), Skip2 (G31.2), Skip3 (G31.3) and Skip4 (G31.4).
(3) When the Enable input is active, if there are commands such as G31/G31.1 (Skip1), G31.2 (Skip2), G31.3 (Skip3) and G31.4 (Skip4), skip the block that is currently performed and then, the next block is performed. If there are M/S/T codes, the relevant code is performed and then, the next block is performed.
(4) When the Enable input is active, if there are commands such as G37/G37.1 (Skip1), G37.2 (Skip2), G37.3 (Skip3) and G37.4 (Skip4), perform the automatic measurement of tool length command in the block that is currently performed.
(5) If the function block is executed, the skipped position can be known as the current position of machine is saved in flags for each NC channel/axis.

6.8.18 Dry Run (NC_DryRun)

Motion Function Block		
		
Input-Output		
UINT	NcChannel	Specify the NC channel to make commands (1 ~ 4 : Channel 1 ~ Channel 4)
Input		
BOOL	Enable	The relevant parameters are output while the input is enabled.
BOOL	AuxFuncLock	If an input is active, ignore an auxiliary function code (M/S/T).
Output		
BOOL	Enabled	Indicate that the function block is performed.
BOOL	Busy	Indicate that execution of the function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the error number that occurred while the function block is running.

(1) This motion function block is the function block that conducts a dry run on the relevant NC channel.
(2) While the Enable input is active, conduct a dry run.
(3) When conducting a dry run, conduct 0: Dry running speed operation and 1: High transfer speed operation according to the set parameter in G00.
(4) If the AuxFuncLock input is active, strobe signals of auxiliary function codes (M/S/T) excluding M00, M01, M02, M30, M98 and M99 are not output.

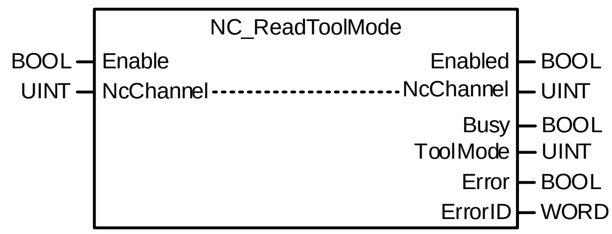
6.8.19 Tool Retract/Recover Operation (NC_ToolMode)

Motion Function Block		
		
Input-Output		
UINT	NcChannel	Specify the NC channel to make commands (1 ~ 4 : Channel 1 ~ Channel 4)
Input		
BOOL	Execute	Give the tool retract or recover operation command to the rising Edge of input.
UINT	ToolMode	Give the tool retract (1) or recover (2) operation command.
Output		
BOOL	Done	Indicate that the function block is successfully applied.
BOOL	Busy	Indicate that execution of the function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the error number that occurred while the function block is running.

(1) This motion function block is the function block that the relevant NC channel gives commands, such as tool retract or recover operation commands, to the relevant channel.
(2) Give the tool retract or recover operation command to ToolMode in the rising Edge of the Execute input.
(3) When conducting the tool retract operation, the retract operation should be conducted by the JOG operation. While the retract operation is conducted by the JOG operation, a position at the time that an operation axis changes is remembered up to 10 times.
(4) When conducting the tool retract operation, the JOG operation must be written not to select more than 2 axes simultaneously.
(5) When conducting the tool recover operation, the tool should be recovered at the remembered position.

6.8.20 Read Tool Retract/Recover Modes (NC_ReadToolMode)

Motion Function Block

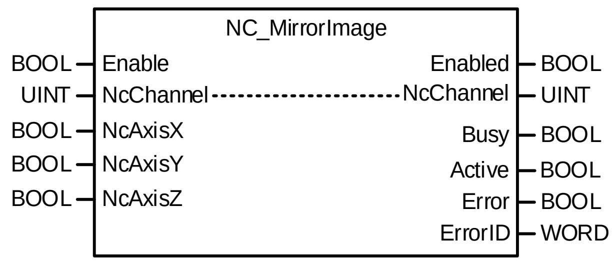


Input-Output		
UINT	NcChannel	Specify the NC channel to make commands (1 ~ 4 : Channel 1 ~ Channel 4)
Input		
BOOL	Enable	While an input is active, confirm a status during tool retract/recover.
Output		
BOOL	Enabled	Indicate that the function block is performed.
BOOL	Busy	Indicate that execution of the function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the error number that occurred while the function block is running.
UINT	ToolMode	Indicate whether a status is tool retract (1) or tool recover (2).

(1) This motion function block is the function block that the relevant NC channel gives the command to confirm whether a status is tool retract or tool recover to the relevant channel.
(2) While the Enable input is active, the ToolMode output helps you to understand whether a status is tool retract (1) or tool recover (2).
(3) When the status is tool retract, more than 2 axes should not be operated.

6.8.21 Mirror Image (NC_MirrorImage)

Motion Function Block

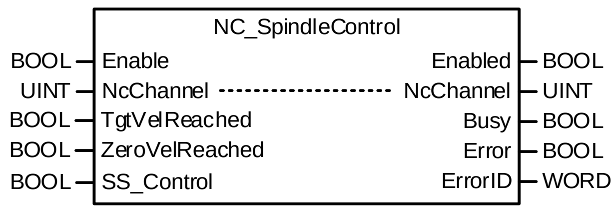


Input-Output		
UINT	NcChannel	Specify the NC channel to make commands (1 ~ 4 : Channel 1 ~ Channel 4)
Input		
BOOL	Enable	While an input is enabled, execute the reverse operation on the specified axis in the relevant channel.
BOOL	NcAxisX	Give the reverse operation signal to the X-axis.
BOOL	NcAxisY	Give the reverse operation signal to the Y-axis.
BOOL	NcAxisZ	Give the reverse operation signal to the Z-axis.
Output		
BOOL	Enabled	Indicate that the function block is performed.
BOOL	Busy	Indicate that execution of the function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the error number that occurred while the function block is running.

(1) This motion function block is the function block that conducts the operation to reverse the transfer position on NC axes (X, Y and Z) in the relevant channel.
(2) While the Enable input is active, conduct an operation by reversing the transfer position on the set axis.
(3) Conduct the reverse operation only for G00, G01, G02, G03, G31.x and G37.x among G codes.
(4) If NC_MirrorImage function block is executed during automatic operation start, error “0x3362” occurs, When it is necessary to execute a function block during automatic operation startup, it can be executed after executing the NC_BlockControl function block in SingleBlock operation mode.

6.8.22 Spindle operation control (NC_SpindleControl)

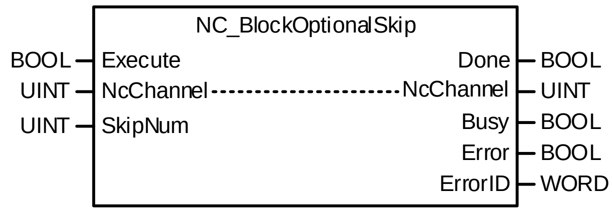
Motion Function Block



Input-Output		
UINT	NcChannel	Specify the NC channel to make commands (1 ~ 4 : Channel 1 ~ Channel 4)
Input		
BOOL	Enable	While an input is enabled, conduct the specified operation on the main spindle in the relevant channel.
BOOL	TgtVelReached	Convey whether the main spindle reaches the target speed to the NC function module. 0: The target speed is not reached 1: The target speed is reached
BOOL	ZeroVelReached	Convey whether the main spindle reaches the zero speed to the NC function module. 0: The zero speed is not reached 1: The zero speed is reached
BOOL	SS_Control	Start the SS control mode of the main spindle. (future support) 0: Start SS controls 1: End SS controls
Output		
BOOL	Enabled	Indicate that the function block is performed.
BOOL	Busy	Indicate that execution of the function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the error number that occurred while the function block is running.

(1) This motion function block performs the user's specified operations for the main spindle on the NC channel specified by the function block if the spindle control is conducted on NC.
(2) When the spindle axis of the channel is not enabled to be automatically operated on the NC function module, the error '0x36D0' occurs.
(3) When the axis specified as the main spindle of the channel is not ready for operation, the error '0x36D1' occurs.
(4) For more information about automatic operation on the NC function module, see '9.5.1 How to Operate Spindle Axes'.

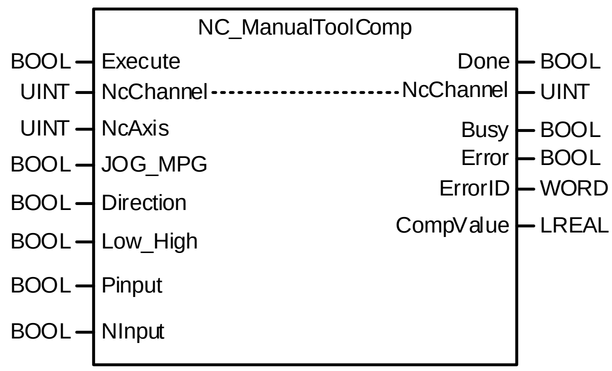
6.8.23 NC optional block skip (NC_BlockOptionalSkip)

Motion Function Block		
		
Input-Output		
UINT	NcChannel	Specify the NC channel to make commands (1 ~ 4 : Channel 1 ~ Channel 4)
Input		
BOOL	Execute	Give the skip signal to SkipNum in the rising Edge of input.
UINT	SkipNum	Specify a signal to do a block skip (1 ~ 9) (1: /or/1, 2: /2, 3: /3, 4: /4, 5: /5, 6: /6, 7: /7, 8: /8, 9: /9) Abort the skip function if it is set to 0 and executed.
Output		
BOOL	Done	Indicate that the function block is successfully applied.
BOOL	Busy	Indicate that execution of the function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the error number that occurred while the function block is running.

(1) This motion function block is the function block that gives the optional skip command to the NC channel.
(2) Skip the block that "/n" is used in front of the block of NC programs according to SkipNum input values in the rising edge of the Execute input. For example, if SkipNum is 3, skip the block that /3 is written in front of it. After skipping the block that is currently conducted, the next block is conducted. If there are M/S/T codes, the relevant code is implemented and the next block is executed.
(3) When setting 0 to SkipNum and executing commands, the skip function is aborted.
(4) When setting values other than 0~9 to SkipNum, the error "0x36A0" occurs.

6.8.24 Manual Measurement of Compensation Amount (NC_ManualToolComp)

Motion Function Block

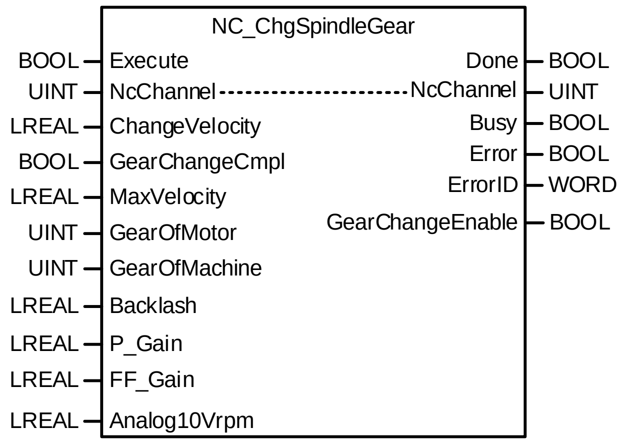


Input-Output		
UINT	NcChannel	Specify the NC channel to make commands (1 ~ 4 : Channel 1 ~ Channel 4)
UINT	NcAxis	Set the channel axis. (1~3: X=1, Y=2, Z=3)
Input		
BOOL	Execute	Give the compensation amount input mode setting command in the rising Edge of input.
BOOL	JOG_MPG	Choose operation methods (0: JOG, 1: MPG)
BOOL	Direction	The direction in JOG operation (0: Forward direction, 1: Reverse direction)
BOOL	Low_High	The speed in JOG operation (0: Low speed, 1: High speed)
BOOL	PInput	The input signal measured in forward direction
BOOL	NInput-	The input signal measured in reverse direction
Output		
BOOL	Done	Indicate that the function block is successfully applied.
BOOL	Busy	Indicate that execution of the function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the error number that occurred while the function block is running.
LREAL	CompValue	Output the calculated compensation amount.

(1) This motion function block is the function block that gives the manual measurement of the tool compensation amount command to the axis set to NcAxis in the relevant NC channel.
(2) Give the manual measurement of the tool compensation amount command in the rising Edge of the Execute input.
(3) If a command is conducted, start the operation selected in JOG_MPG. If the signal selected in PInput or NInput becomes 1, stop the operation and calculate a compensation amount by using the relevant position value.
(4) The compensation amount is calculated by the following formula:
Compensation amount = Position of an axis when PInput/NInput is On - Measured reference position
(5) The measured reference position is selected from “+ measured reference distance X of automatic tool offsets” to “- measured reference distance Z of automatic tool offsets” of channel parameters according to an axis. For example, if NcAxis is selected as Y and NInput is On, the value set in “- measured reference distance Y of automatic tool offsets” becomes the measured reference position.
(6) The calculated compensation amount is output to CompValue and Done becomes 1.
(7) If PInput and NInput are On simultaneously, the amount is recognized with PInput.
(8) If setting axes other than X~Z to NcAxis and executing function blocks, the error “0x36B0” occurs.

6.8.25 NC spindle gear change (NC_ChgSpindleGear)

Motion Function Block		



Input-Output		
UINT	NcChannel	Specify the NC channel to make commands (1 ~ 4 : Channel 1 ~ Channel 4)
Input		
BOOL	Execute	Give the spindle gear conversion command in the rising Edge of input.
LREAL	ChangeVelocity	Set the speed value to change
BOOL	GearChangeCmpl	The signal that gear changes have been completed. Each setting value of Operand is set as the relevant parameter after turning On
LREAL	MaxVelocity	The maximum setting value for velocity parameters
UINT	GearOfMotor	The gear ratio parameter of a motor
UINT	GearOfMachine	The gear ratio parameter of machine
LREAL	Backlash	Backlash value
LREAL	P_Gain	P gain setting value
LREAL	FF_Gain	Setting value for the Feed Forward gain
LREAL	Analog10Vrpm	Disabled
Output		
BOOL	Done	Indicate that the function block is successfully applied.
BOOL	Busy	Indicate that execution of the function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the error number that occurred while the function block is running.
BOOL	GearChangeEnable	Indicate whether to change gears

(1) This motion function block is the function block that gives the spindle gear change command to the relevant NC channel.
(2) The spindle gear change command is given in the rising Edge of the Execute input.
(3) If the command is executed, change the current spindle speed into the value set in ChangeVelocity that can change gears.
(4) If the speed of the spindle axis is changed to less than the value set in ChangeVelocity and the GearChangeEnable output turns On, users run sequence programs to change gears and enter On in GearChangeCmpl when gear change is completed.
(5) If the GearChangeCmpl turns On, users set values of the following items set in the function block in parameters and run the spindle with the changed settings.

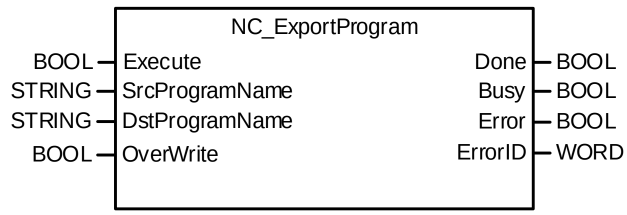
Speed limit values (MaxVelocity)

Gear ratio of a motor (GearOfMotor)
Gear ratio of machine (GearOfMachine)
Backlash compensation amount (Backlash)
P gain in a position mode (P_Gain)
Feed Forward gain in a position mode (FF_Gain)

(6) If setting the ChageVelocity value to values bigger than speed limit values of the relevant axis and executing the function block, the error “0x36C0” occurs.
(7) If setting the MaxVelocity value to values less than 0 and executing the function block, the error “0x36C1” occurs.
(8) If setting the GearOfMotor value to values less than 0 or bigger than 65535 and executing the function block, the error “0x36C2” occurs.
(9) If setting the GearOfMachine value to values less than 0 or bigger than 65535 and executing the function block, the error “0x36C3” occurs.
(10) If setting the Backlash value to values less than 0 and executing the function block, the error “0x36C4” occurs.
(11) If setting the P_Gain value to values less than 0 or bigger than 500 and executing the function block, the error “0x36C5” occurs.
(12) If setting the FF_Gain value to values less than 0 or bigger than 100 and executing the function block, the error “0x36C6” occurs.

6.8.26 Save NC program to SD Card (NC_ExportProgram)

Motion Function Block



Input		
BOOL	Execute	Save the program to the SD card on the rising edge.
STRING	SrcProgramName	Set the program name stored in PLC. Combination of English and numbers
STRING	DstProgramName	Set the program name to be saved on the SD card. Combination of English and numbers
BOOL	OverWrite	0: Overwrite Protection, 1: Overwrite Permission
Output		
BOOL	Done	Indicate the state of motion function block completion.
BOOL	Busy	Indicate that execution of the function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the error number that occurred while the function block is running.

(1) This motion function block is a function block that saves the NC program stored in the PLC to the SD card.

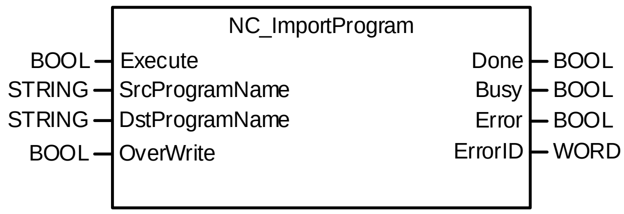
(2) After setting the ProgramName to be read from PLC and the ProgramName to be saved to the SD card and executing the function block, the corresponding program is saved to the SD card. (Save Path: XMC-Exxx\MACxxxxxxxxxxxx\Inst)

(3) If a file with the same name as DstProgramName exists on the SD card, the following processing is performed according to the OverWrite setting value.

OverWrite	Operation
0	It does not overwrite the file and occurs an error.
1	Overwrites the file.

6.8.27 Save SD card NC program to PLC (NC_ImportProgram)

Motion Function Block



Input		
BOOL	Execute	At the rising edge, the program stored in the SD card is saved in the PLC.
STRING	SrcProgramName	Set the program name stored in SD. Combination of English and numbers
STRING	DstProgramName	Set the program name to be saved on the SD. Combination of English and numbers
BOOL	OverWrite	0: Overwrite Protection, 1: Overwrite Permission
Output		
BOOL	Done	Indicate the state of motion function block completion.
BOOL	Busy	Indicate that execution of the function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the error number that occurred while the function block is running.

(1) This motion function block is a function block that saves the NC program stored in the SD card to the PLC.

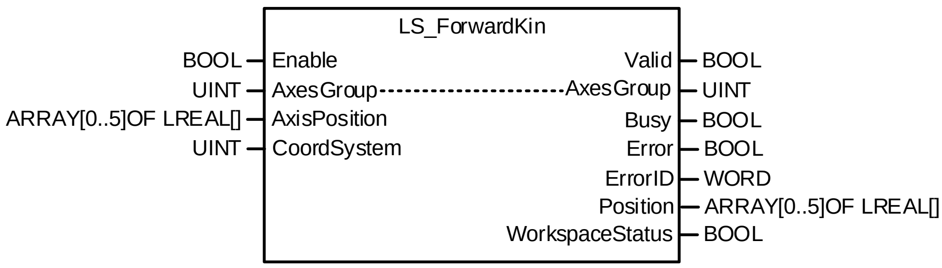
(2) The NC program file must be saved in the corresponding path of the SD card. (Save Path: XMC-Exxx\MACxxxxxxxxxxxx\Inst)

(3) After setting the ProgramName to be read from SD and the ProgramName to be saved to the PLC and executing the function block, the corresponding program is saved to the PLC.

(4) If a file with the same name as DstProgramName exists on the PLC, the following processing is performed according to the OverWrite setting value.

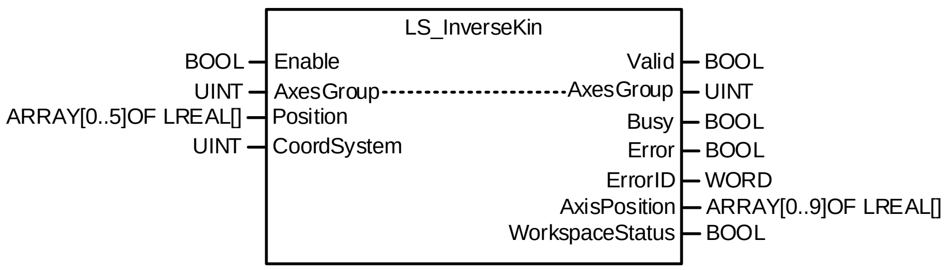
OverWrite	Operation
0	It does not overwrite the file and occurs an error.
1	Overwrites the file.

6.8.28 Forward Kinematic operation (LS_ForwardKin)

Motion Function Block		
		
Input-Output		
UINT	AxesGroup	Set the group to perform the forward kinematic function (1~16: 1group~16group)
Input		
BOOL	Enable	While the input is active, a forward kinematic operation is performed on the designated axis group.
LREAL[]	AxisPosition	Enter the position of each axis of the axis group.
UINT	CoordSystem	Select the coordinate system type.(1:MCS 2:PCS)
Output		
BOOL	Valid	Indicate the validity of function block output.
BOOL	Busy	It indicates that the function block is being executed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the error number that occurred while the function block is running.
LREAL[]	Position	It indicates the position of the instrument end point calculated by the selected coordinate system type.
BOOL	WorkspaceStatus	Indicates whether a working radius error has occurred.

(1) This motion function block converts the position (ACS) of each axis input in AxesGroup into the selected MCS/PCS coordinate system and outputs it as Position.
(2) When a workspace error is detected, the WorkspaceStatus value is output as ‘1’.
(3) This function block does not perform direct operation.

6.8.29 Inverse Kinematic operation (LS_InverseKin)

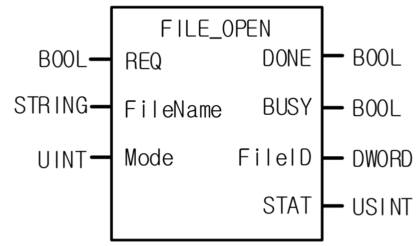
Motion Function Block		
		
Input-Output		
UINT	AxesGroup	Set the group to perform the inverse kinematic function (1~16: 1group~16group)
Input		
BOOL	Enable	While the input is active, a inverse kinematic operation is performed on the designated axis group.
LREAL[]	Position	Enter the target position of the end point of the machine.
UINT	CoordSystem	Select the coordinate system type.(1:MCS 2:PCS)
Output		
BOOL	Done	Indicates that the execution of the motion function block is completed.
BOOL	Busy	Indicate that execution of the function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the error number that occurred while the function block is running.
LREAL[]	AxisPosition	It indicates the position of each axis to reach the target position of the end point of the machine.
BOOL	WorkspaceStatus	Indicates whether a working radius error has occurred.

(1) This motion function block converts the MCS/PCS coordinate system selected in AxesGroup into ACS and outputs it as AxisPosition.
(2) When a workspace error is detected, the WorkspaceStatus value is output as ‘1’.
(3) This function block does not perform direct operation.

6.9 File instruction

6.9.1 Open File in SD Memory Cards (FILE_OPEN)

Motion Function Block



Input		
BOOL	REQ	Set programs to run in the rising Edge.
STRING	FileName	Set file names to specify
UINT	Mode	File open mode.
Output		
BOOL	Done	Indicate the state of motion function block completion.
BOOL	Busy	Indicate that execution of the function block is not completed.
DWORD	FileID	ID of the file that is opened.
USINT	STAT	Output the error number that occurred while the function block is running.

(1) Opens the file specified by 'FileName' on the SD memory card (FileName is a combination of letters and numbers).
(2) When executing 'open', an operation changes according to the setting value of the mode.

Mode	Operation
0	Open a file to read and write. If there is no file, create a new file. If files with the same name exist, delete the contents of the files and create new ones.
1	Open a file to read and continuously write. If there is no file, create a new file. If files with the same name exist, perform continuous writing from the end of the files when writing.
2	Open a file as read-only.

(3) Read a file from the beginning upon FILE_READ after FILE_OPEN. However, as reading a file from the last part when performing FILE_READ after FILE_WRITE, perform Read after moving a position to FILE_SEEK.
(4) The ID of a file opened when a file is normally opened is output as 'FileID'.
(5) 'FileID' is used when FILE_WRITE, FILE_READ, FILE_SEEK and FILE_CLOSE commands are executed.
(6) When FILE_OPEN is normally performed, STAT = 0. And when an error occurs, STAT information is as follows:
(7) The maximum number of FILE_OPEN is 50. (Including data log files)

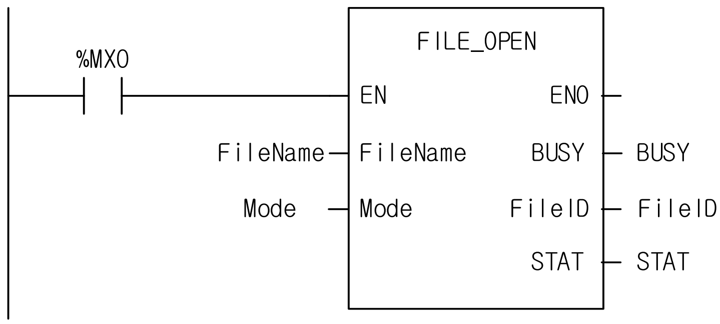
STAT	Error status
0	Normal
1	Failed to access SD memory cards
2	File is already openfile is already open
3	If it is mode 2 and there is no file in the Inst folder. If SD card is not installed.
4	If you have more than 50 files open
5	If the Mode is a value other than 0~2

(8) The file with FILE_OPEN must be closed by executing the FILE_CLOSE command after using it.
(9) If the PLC mode is changed, the status that a file is opened continues. Therefore, execute the FILE_OPEN command again after closing the file.

■ **Program Example**

(1) LD

- FileName = 'ABC', Mode = 0



(a) If the execution condition (%MX0) is On, the FILE_OPEN function is executed.

(b) If SD cards are normally installed, open a file that is able to read and write with FileName = 'ABC'. If files with the same name as ABC exists, delete the contents of the files and a new file is opened.

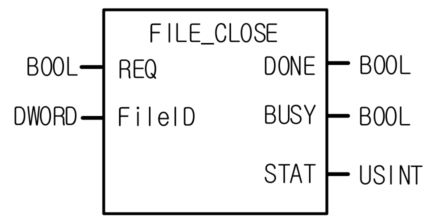
(c) According to the status of SD cards or files, STAT displays an error. If normal operation, 0 is output.

(2) ST

INST_FILE_OPEN (REQ:=%MX0, FileName:='ABC', Mode:=0, DONE=>DONE, BUSY=>BUSY, FileID=>FileID, stat=>stat);

6.9.2 Close file in SD Memory Cards (FILE_CLOSE)

Motion Function Block



Input		
BOOL	REQ	Set programs to run in the rising Edge.
DWORD	FileID	ID of the file that is opened.
Output		
BOOL	Done	Indicate the state of motion function block completion.
BOOL	Busy	Indicate that execution of the function block is not completed.
USINT	STAT	Output the error number that occurred while the function block is running.

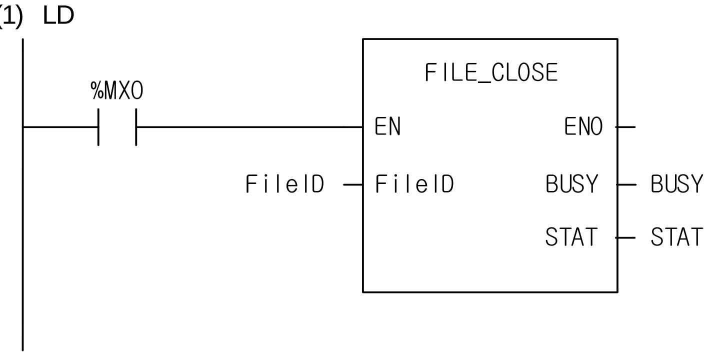
(1) Close the file specified with 'FileID' in SD memory cards.

(2) When FILE_CLOSE is normally performed, STAT = 0. And when an error occurs, STAT information is as follows:

STAT	Error status
0	Normal
1	Failed to access SD memory cards
2	If there are no opened files

■ **Program Example**

(1) LD



(a) The output value, FileID must be entered after FILE_OPEN is normally executed.

(b) If the execution condition (%MX0) is On, the FILE_CLOSE function is executed.

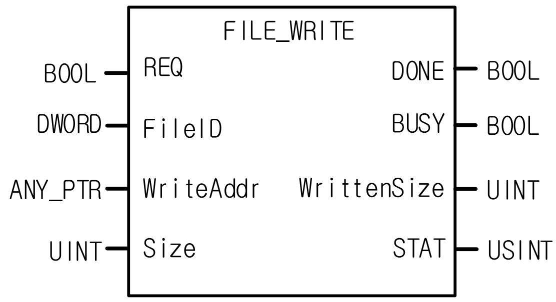
(c) According to the status of SD cards or files, STAT displays an error. If normal operation, 0 is output.

(2) ST

INST_FILE_CLOSE (REQ:=%MX0, FileID:=FileID, DONE=>DONE, BUSY=>BUSY, stat=>stat);

6.9.3 Write file in SD Memory Cards (FILE_WRITE)

Motion Function Block



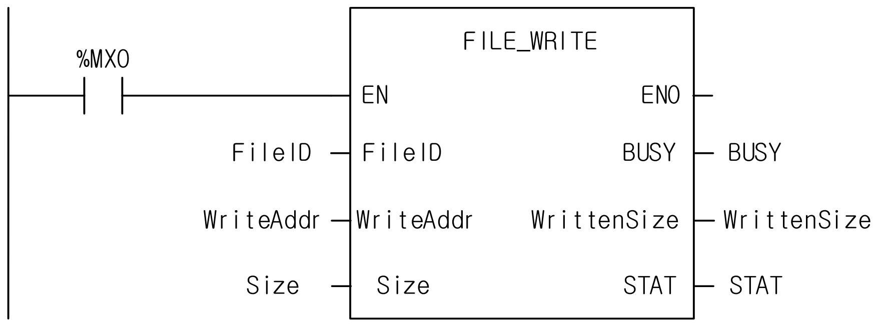
Input		
BOOL	REQ	Set programs to run in the rising Edge.
DWORD	FileID	ID of the file that is opened.
ANY_PTR	WriteAddr	Address of data to write
UINT	Size	Number of data to write
Output		
BOOL	Done	Indicate the state of motion function block completion.
BOOL	Busy	Indicate that execution of the function block is not completed.
UINT	WrittenSize	Number of data that writing is completed.
USINT	STAT	Output the error number that occurred while the function block is running.

(1) Write the file opened with ‘FileID’ in SD memory cards.
(2) The data to write is the contents of WriteAddr and executes writing as many as Sizes.
(3) If WriteAddr is announced as an Array type, write data in the array as many as Sizes to write.
(4) If an Array type, the size of data to write is WriteAddr data type x Size. (If Byte, the data type is 1)
(5) If WriteAddr is announced as a data type, write only the relevant data value irrespective of Size values.
(6) If writing, BUSY = 1. Upon completion, BUSY = 0 and DONE = 1.
(7) When executing normally FILE_WRITE, the size of data that completed Write is output to WrittenSize.
(8) Upon normal completion, STAT = 0. And when an error occurs, STAT information is as follows:
(9) When forcing to remove SD cards before FILE_CLOSE, data is not normally saved.

STAT	Error status
0	Normal
1	Failed to access SD memory cards
2	The status that files with FileID are not opened
3	If a file is opened as read-only
4	If the Size is 0 (an Array type), If the Size is over 65535

■ **Program Example**

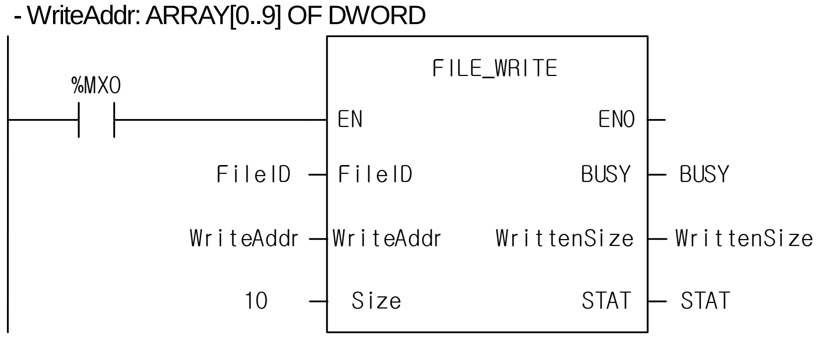
(1) LD



(a) The output value, FileID must be entered after FILE_OPEN is normally executed.
(b) If the execution condition (%MX0) is On, the FILE_WRITE function is executed.
(c) WriteAddr is able to be set as an array type or a data type.
(d) When setting it as an array type, data in an array range can be written into SD cards. For example, if setting it as 10 DWORD arrays, 10 array values can be written with [0] ~ [9] using the Size.
(e) If setting it as a data type, only the relevant value is written into SD cards. The Size value is not valid.
(f) Upon normal execution, WriteSize displays the data size that is actually write.
(g) According to the status of SD cards or files, STAT displays an error. If normal operation, 0 is output.

※ **WriteAddr Array Type Examples**

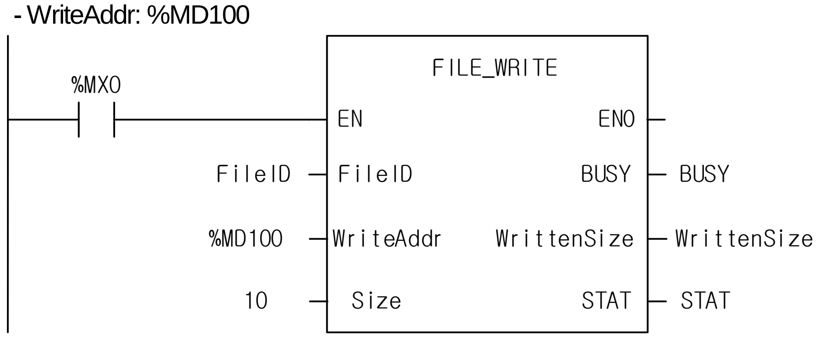
- WriteAddr: ARRAY[0..9] OF DWORD



(a) If the execution condition (%MX0) is On, the FILE_WRITE function is executed.
(b) As WriteAddr is an array type and the Size is 10, write [0] ~ [9] data of WriteAddr.
(c) As writing 10 DWORD data, WrittenSize displays 40 and STAT outputs 0 after writing is completed.

※ **WriteAddr Array Type Examples**

- WriteAddr: %MD100



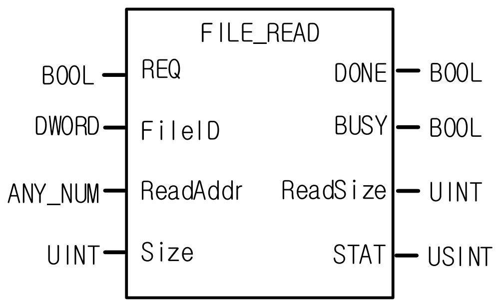
(a) If the execution condition (%MX0) is On, the FILE_WRITE function is executed.
(b) As the Size is 10 but WriteAddr is a data type, write the set value of %MD100.
(c) As they are DWORD data, WrittenSize displays 4 and STAT outputs 0.

(2) ST

INST_FILE_WRITE (REQ:=%MX0, FileID:=FileID, WriteAddr:=WriteAddr, Size:=Size, DONE=>DONE, BUSY=>BUSY, WrittenSize=>WrittenSize, stat=>stat)

6.9.4 Read Files in SD Memory Cards (FILE_READ)

Motion Function Block



Input		
BOOL	REQ	Set programs to run in the rising Edge.
DWORD	FileID	ID of the file that is opened.
ANY_NUM	ReadAddr	The address starting with data to read.
UINT	Size	Number of data to read.
Output		
BOOL	Done	Indicate the state of motion function block completion.
BOOL	Busy	Indicate that execution of the function block is not completed.
UINT	ReadSize	Number of data that reading is completed.
USINT	STAT	Output the error number that occurred while the function block is running.

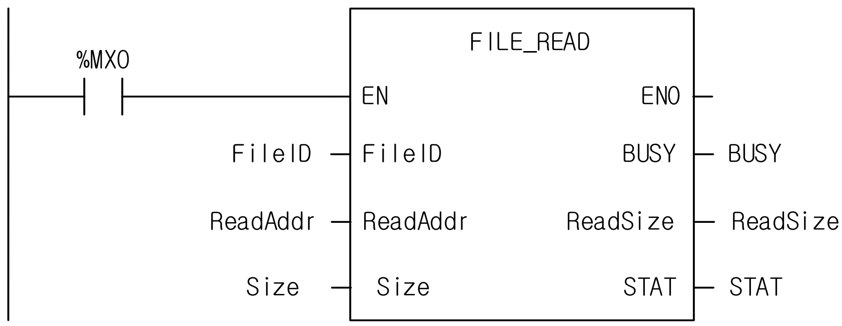
(1) Read the file opened with 'FileID' in SD memory cards.
(2) After executing FILE_OPEN, read a file from the beginning. After executing FILE_WRITE, make a file pointer read it from the last position.
(3) If a position shift is needed, it is performed after shifting position with the FILE_SEEK command.
(4) The read data is saved in ReadAddr and read data as much as the number of Sizes.
(5) If ReadAddr is announced as an Array type, read data in the array as many as Sizes to read.
(6) If an Array type, the size of data to write is ReadAddr data type x Size. (If Byte, the data type is 1)
(7) If ReadAddr is announced as a data type, read only the value of a data type irrespective of Size values.
(8) If reading, BUSY = 1. Upon completion, BUSY = 0 and DONE = 1.
(9) When executing normally FILE_READ, the size of data that completed reading is output to ReadSize.
(10) Upon normal completion, STAT = 0. And when an error occurs, STAT information is as follows:

STAT	Error status
0	Normal
1	Failed to access SD memory cards
2	The status that files with FileID are not opened
3	If the Size is 0 (an Array type) or there is actually no data to read

(11) If a file pointer points to the end of a file, 'STAT = 3' is output as there is no data to read.

■ **Program Example**

(1) LD

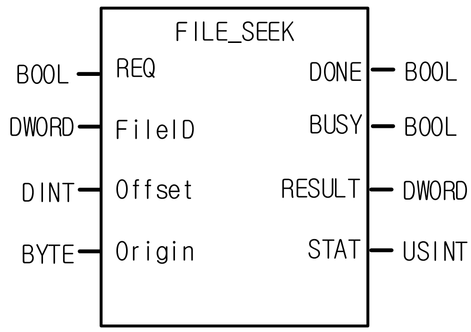


(a) The output value, FileID must be entered after FILE_OPEN is normally executed.
(b) If the execution condition (%MX0) is On, the FILE_READ function is executed.
(c) ReadAddr is able to be set as an array type or a data type.
(d) When setting it as an Array type, data of the file saved in SD cards can be read as many as Sizes in the set array. For example, if ReadAddr is set as 10 DWORD arrays, read data saved in SD cards in array as many as Sizes. If setting it as a data type, read it only with the relevant data value.
(e) The Size value is not valid. Upon normal execution, ReadSize displays the data size that is actually read.
(f) According to the status of SD cards or files, STAT displays an error. If normal operation, 0 is output.

(2) ST
INST_FILE_READ (REQ:=%MX0, FileID:=FileID, ReadAddr:=ReadAddr, Size:=Size, DONE=>DONE, BUSY=>BUSY, ReadSize=>ReadSize, stat=>stat);

6.9.5 Seek the Position to Access in SD Memory Card (FILE_SEEK)

Motion Function Block



Input		
BOOL	REQ	Set programs to run in the rising Edge.
DWORD	FileID	ID of the file that is opened.
DINT	Offset	A position from Origin to Offset
Byte	Origin	A reference position.
Output		
BOOL	Done	Indicate the state of motion function block completion.
BOOL	Busy	Indicate that execution of the function block is not completed.
DWORD	Result	Output the changed position.
USINT	STAT	Output the error number that occurred while the function block is running.

(1) Specify position to access in the file opened with 'FileID' in SD memory cards.

(2) A reference position is set as 3 modes as follows:

Origin value	Origin position
0	Front of file
1	The current position of a file pointer
2	End of file

(3) Move the position of a file pointer by adding the setting value of a reference position to the Offset value.

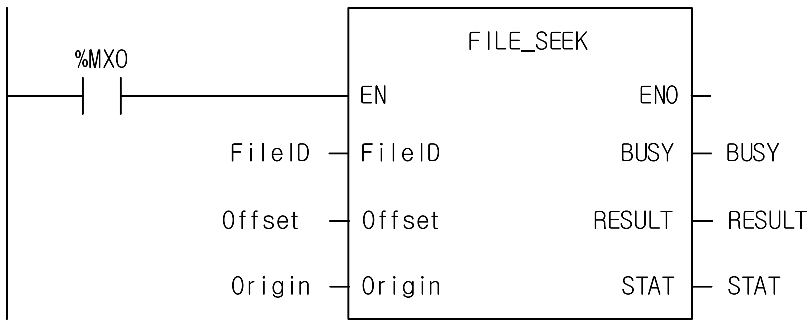
(4) If operation, BUSY = 1. Upon completion, BUSY = 0 and DONE = 1.

(5) Upon normal completion, RESULT displays the moved position and STAT = 0. And when an error occurs, STAT information is as follows:

STAT	Error status
0	Normal
1	Failed to access SD memory cards
2	The status that files with FileID are not opened
3	If the value of a position to move is less than the value of origin

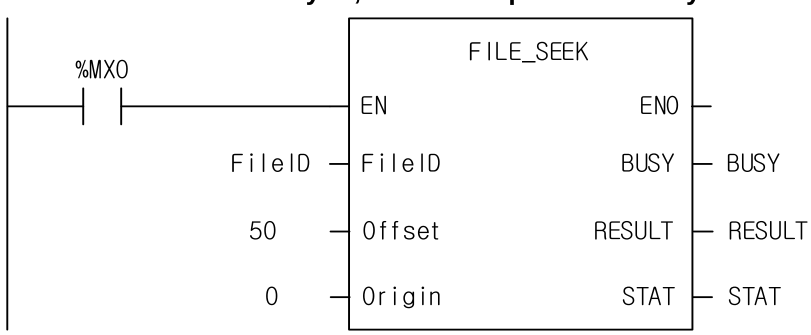
■ **Program Example**

(1) LD

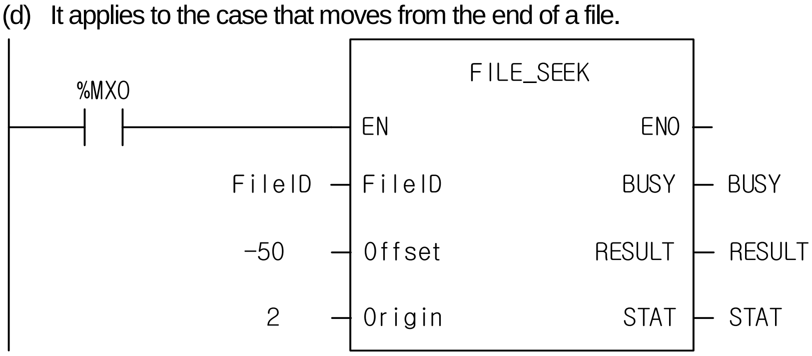


(a) The output value, FileID must be entered after FILE_OPEN is normally executed.
(b) If the execution condition (%MX0) is On, the FILE_SEEK function is executed.
(c) Move a file pointer by adding the Offset value to the setting of origin. For example, if you want to move to the beginning of a file, set it as Offset = 0 and Origin = 0. If you move to the position of 20 bytes, you can set it as Offset = 20 and Origin = 0.
(d) Upon normal execution, RESULT displays the current file pointer.
(e) According to the status of SD cards or files, STAT displays an error. If normal operation, 0 is output.

※ When a file size is 100 bytes, move to the position of 50 bytes



(a) If the execution condition (%MX0) is On, the FILE_SEEK function is executed.
(b) After moving to the beginning point of a file as Origin = 0, move to the position where Offset = 50 is added.
(c) RESULT displays the moved 50 bytes.
(d) It applies to the case that moves from the end of a file.



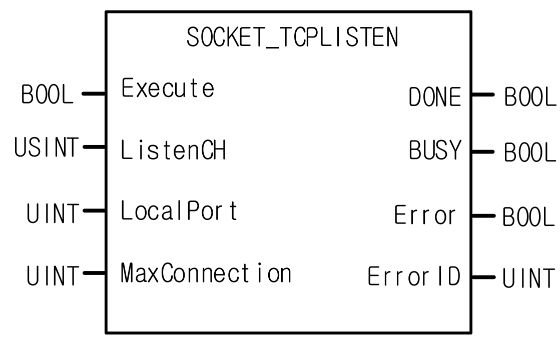
(2) ST

INST_FILE_SEEK(REQ:=%MX0, FileID:=FileID, Offset:=Offset, Origin:=Origin, DONE=>DONE, BUSY=>BUSY, RESULT=>RESULT, stat=>stat);

6.10 Communication commands

6.10.1 Prepare to allow external TCP connections. (TCP Server)

Function Block



Input		
BOOL	Execute	Action on rising edge
USINT	ListenCH	Socket service channel input to wait for external connection(1~16)
UINT	LocalPort	Specifies own port number(1~65535)
UINT	MaxConnection	Maximum number of external connections(1~15)
Output		
BOOL	Done	Function block completion status
BOOL	Busy	During execution of function block
BOOL	Error	Whether an error occurred
UINT	ErrorID	ID in case of error - in normal operation: 0

(1) Function
- (a) Listen operation of the socket service channel (ListenCH) set to the local TCP port number (LocalPort) is performed.
- (b) Create external connection queue as many as MaxConnection.
- (c) After the command ends normally ('DONE' value is TRUE), multiple client connections are allowed through the SOCKET_TCPACCEPT command.
- (d) Refer to the example program "14.4 SOCKET SERVICE".
- (e) Refer to the detailed operation specification "Command operation timing chart".

Tip

Close the service channel initialized with this command using the SOCKET_CLOSE command.
The maximum number of socket service channels is 16.
When the connection with the client is completed and the same LocalPort is reused after CLOSE, the use of the corresponding port may be temporarily restricted. Refer to INVALID_ADDRESS_OR_PORT of occurrence error

(2) Error Status Information

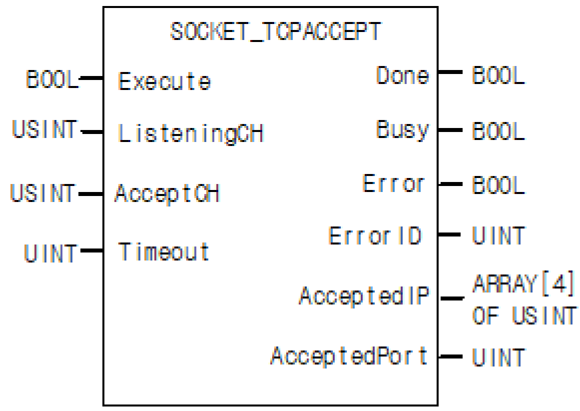
ID	Error name	Cause	Solutions
0	No Error	Normal status	-
1	INVALID_PARAMETER	Command input parameter error	- Normal range parameter input
2	SYSTEM_RESERVED_	Attempt to use the system	- Change Local Port input value

	PORT	reserved port	
3	ALREADY_USED_CH	Reassign channels already in use	− Use after checking the usage status of the service channel
10	SOCKET_SERVICE_PROCEDURE	Common errors that occur during service processing	− Command operation retry through exception handling − Review input values − System check
12	INVALID_ADDRESS_OR_PORT	Input IP, PORT value abnormal, unavailable status	− After the connection is completed after specifying the local port, when attempting to connect using the same local port, the error may persist for up to 60 seconds for a certain period of time to initialize the port used for the existing connection. In this case, retry the command operation through exception handling − Check whether a port that is already in use is specified − Local Port normal input value, check the range

※ Errors occurring outside the list Refer to “SOCKET SERVICE Common Error”

6.10.2 **Prepare to allow external TCP connections. (TCP Server)**

Function Block



Input		
BOOL	Execute	Action on rising edge
USINT	ListenCH	Input the listening service channel assigned through SOCKET_TCPLISTEN(1~16)
USINT	AcceptCH	Input socket service channel to accept connection with client(1~16)
UINT	TIMEOUT	Time-out
Output		
BOOL	Done	Function block completion status
BOOL	Busy	During execution of function block
BOOL	Error	Whether an error occurred
UINT	ErrorID	ID in case of error - in normal operation: 0
ARRAY[4] OF USINT	AcceptedIP	Client-side IP with established connection(0~255)
UINT	AcceptedPort	Client-side Port with established connection(1~65535)

(1) Function
- (a) Through SOCKET_TCPLISTEN, it accepts the connection of an external connection (Client) that comes into the service channel (TCP Server) in the Listening status(Same function as Accept function)
- (b) Refer to the example program “14.4 SOCKET SERVICE”.
- (c) Refer to the detailed operation specification “Command operation timing chart”.

Tip

Before executing the corresponding command, it is necessary to set the server through SOCKET_TCPLISTEN.
Close the service channel initialized with this command using the SOCKET_CLOSE command.
The maximum number of socket service channels is 16.

(2) Error Status Information

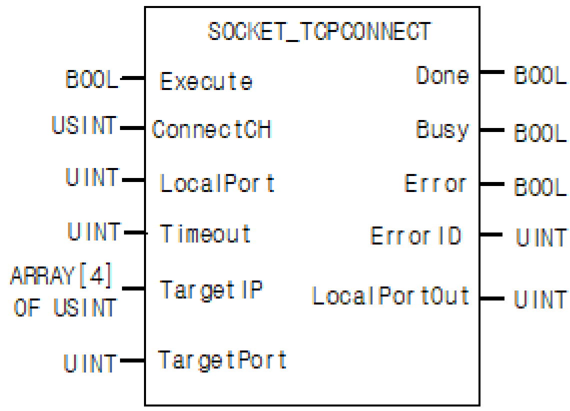
ID	Error name	Cause	Solutions
0	No Error	Normal status	Normal status
1	INVALID_PARAMETER	-	-
3	ALREADY_USED_CH	Command input parameter error	Command input parameter error
4	WAIT_TIMEOUT	Normal range parameter input	Normal range parameter input

5	COMMAND_INSTANCE_POOL_FULL	Reassign channels already in use	Reassign channels already in use
7	SOCKET_CH_NOT_INITIALIZE	Use after checking the usage status of the AcceptCH	Use after checking the usage status of the AcceptCH
10	SOCKET_SERVICE_PROCEDURE	Command timeout occurrence	Command timeout occurrence
11	SOCKET_CLOSED	Check the other side (server) connection standby status	Check the other side (server) connection standby status

※ Errors occurring outside the list Refer to “SOCKET SERVICE Common Error”

6.10.3 Connects to a relative TCP port.(TCP Client operation) (SOCKET_TCPCONNECT)

Function Block



Input		
BOOL	Execute	Action on rising edge
USINT	ConnectCH	Enter the socket service channel to use for Connect(1~16)
UINT	LocalPort	Connect with the specified port number. However, if it is 0, an empty TCP port other than the well-known port is automatically assigned.(1~65535)
UINT	Timeout	Timeout time(*) 0: no timeout 1~65535 : input value * 1sec Waiting
ARRAY[4] OF USINT	TargetIP	Destination IP to connect(0~255)
UDINT	TargetPort	Destination Port to connect(1~65535)
Output		
BOOL	Done	Function block completion status
BOOL	Busy	During execution of function block
BOOL	Error	Whether an error occurred
UINT	ErrorID	ID in case of error - in normal operation: 0
UINT	LocalPortOut	Local port number finally assigned after connection is established(1~65535)

(1) Function
- (a) Attempts to connect with the destination side address ‘TargetIP’ and ‘TargetPort’. (Same function as connect function)
- (b) Operation that appears differently for each product and when commands are executed
  - ■ When running after entering an unreachable IP to TargetIP
    - XGI-CPUZ: Error On, Occurred ErrorID 16
    - XMC: After maintaining Busy status as much as the value entered in Timeout, Error On,occured ErrorID 4(Wait indefinitely when Timeout is 0)
  - ■ When executing after inputting the IP of the FEnet module that operates as a client to the TargetIP
    - XGI-CPUZ: Error On, Occurred ErrorID 18
    - XMC: Error On, Occurred ErrorID 10.
- (c) Refer to the example program “14.4 SOCKET SERVICE”.
- (d) Refer to the detailed operation specification “Command operation timing chart”.

Tip

Close the service channel initialized with this command using the SOCKET_CLOSE command.
The maximum number of socket service channels is 16.
After the connection with the server and CLOSE, when reusing the same LocalPort, the use of the corresponding port may be temporarily restricted. Refer to INVALID_ADDRESS_OR_PORT of occurrence error
It is recommended to set the automatic port assignment unless it is absolutely necessary to specify the LocalPort. (LocalPort input value 0)

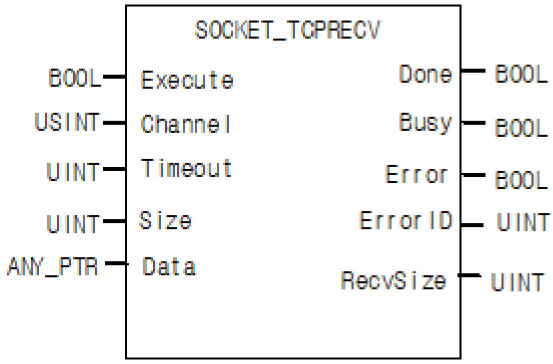
(2) Error Status Information

ID	Error name	Cause	Solutions
0	No Error	Normal status	−
1	INVALID_PARAMETER	Command input parameter error	− Normal range parameter input
2	SYSTEM_RESERVED_PORT	Attempt to use the system reserved port	− Change Local Port input value
3	ALREADY_USED_CH	Reassign channels already in use	− Use after checking the usage status of the service channel
4	WAIT_TIMEOUT	Command timeout occurrence	− Check the other side (server) connection standby status
5	COMMAND_INSTANCE_POOL_FULL	Exceeded command instance usage	− Decrease the frequency of command execution
10	SOCKET_SERVICE_PROCEDURE	Common errors that occur during service processing	− Command operation retry through exception handling − Checking the destination side status − Review input values − System check
11	SOCKET_CLOSED	Attempts to process a command on a socket that has already been closed, or a connection that has been aborted by the peer.	− Check whether SOCKET_CLOSE of the corresponding channel is operating during connection processing
12	INVALID_ADDRESS_OR_PORT	Input IP, PORT value abnormal, unavailable status	− When connecting using the same local port after specifying a local port and completing the connection, the error may last up to 60 seconds for a certain period of time to initialize the port used for the existing connection. − Check whether a port that is already in use is specified − Local Port normal input value, check the range

※ Errors occurring outside the list Refer to “SOCKET SERVICE Common Error”

6.10.4 Receive data from a TCP socket (SOCKET_TCPRECV)

Function Block



Input		
BOOL	Execute	Action on rising edge
USINT	Channel	Socket service channel(TCP)(1~16)
UINT	Timeout	Time-out 0: no timeout 1~65535 : input value * 1sec Waiting
UINT	Size	Save Size Data size to read from RECVDATA(0~1400)
ANY_PTR	Data	Data receive buffer(address)
Output		
BOOL	Done	Function block completion status
BOOL	Busy	During execution of function block
BOOL	Error	Whether an error occurred
UINT	ErrorID	ID in case of error - in normal operation: 0
UINT	RecvSize	Receive data size(0~1400)

(1) Function
- (a) The received data of the socket service channel set in 'Channel' is saved in 'Data'. The size of data to be saved is designated as 'Size'.
- (b) The actual received data size is output in 'RecvSize'
- (c) If there is no data received, it waits for the received data for the time set in the timeout (Timeout).
- (d) Refer to the example program "14.4 SOCKET SERVICE".
- (e) Refer to the detailed operation specification "Command operation timing chart".

Tip

If data is not received and the socket is closed with the SOC_CLOSE command, it will be terminated abnormally even if the timeout is less than the timeout period.
Using a socket service channel initialized for TCP purposes
The total size of the data entered in Data must be greater than or equal to the value entered in Size.

- ■ After TCP connection and disconnection of the destination side, when the command is operated
  - − XGI-CPUZ: Error On, ErrorID 11 occurs when executing after reading all the contents of the receive queue After re-execution, Done On occurs
  - − XMC: After maintaining Busy status as much as the value entered in Timeout, Error On, occured ErrorID 4(Wait indefinitely when Timeout is 0)
- ■ After TCP connection, enter a value of 0 in Size and execute

- XGI-CPUZ: Done On occurs
  - ◆ In case the destination side sends a 0 size message, response to previous
  - ◆ If the connection has already been terminated by the destination side at the time before the reception request
- After Busy, ErrorID 4 occurs (If Timeout value is 0, Busy state is maintained indefinitely)
  - ◆ In case the destination side sends a 0 size message response to previous

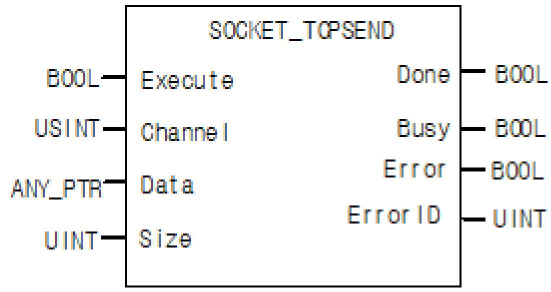
(2) Error Status Information

ID	Error name	Cause	Solutions
0	No Error	Normal status	-
1	INVALID_PARAMETER	Command input parameter error	Normal range parameter input
4	WAIT_TIMEOUT	Command timeout occurrence	● Check the data send status of the destination side Wait for send to continue
5	COMMAND_INSTANCE_POOL_FULL	Exceeded command instance usage	Decrease the frequency of command execution
7	SOCKET_CH_NOT_INITIALIZE	Control attempts through uninitialized channels	Check whether the input socket channel is initialized
10	SOCKET_SERVICE_PROCEDURE	Common errors that occur during service processing	● Command operation retry through exception handling ● Review input values System check
11	SOCKET_CLOSED	Attempts to process a command on a socket that has already been closed, or a connection that has been aborted by the peer.	Check normal connection status
13	CONNECTION_TIMEDOUT	Connection time out occurred	Check connection status of destination side
15	HOST_DOWN	Host device error	● Check host device status Check for abnormalities in cables and connection paths
16	HOST_UNREACH	Host connection disable	● Check host device status Check for abnormalities in cables and connection paths
18	CONNECTION_REFUSED	Connection denied	Check Connection denied status of destination side

※ Errors occurring outside the list Refer to “SOCKET SERVICE Common Error”

6.10.5 Send data from a TCP socket (SOCKET_TCPSEND)

Function Block



Input		
BOOL	Execute	Action on rising edge
USINT	Channel	Socket service channel(TCP)(1~16)
ANY_PTR	Data	Send data buffer(address)
UINT	Size	Send data size(0~1400)
Output		
BOOL	Done	Function block completion status
BOOL	Busy	During execution of function block
BOOL	Error	Whether an error occurred
UINT	ErrorID	ID in case of error - in normal operation: 0

(1) Function
  (a) Data is transmitted through the socket service channel set as 'Channel'. The size of data to be sending is designated as ‘Size’.
  (b) Refer to the example program “14.4 SOCKET SERVICE”.
  (c) Refer to the detailed operation specification “Command operation timing chart”.

Tip

Using a socket service channel initialized for TCP purposes
The total size of the data entered in Data must be greater than or equal to the value entered in Size.

(2) Error Status Information

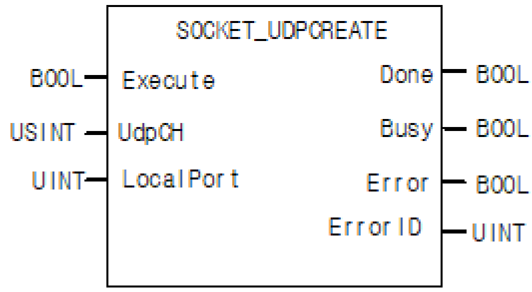
ID	Error name	Cause	Solutions
0	No Error	Normal status	−
1	INVALID_PARAMETER	Command input parameter error	− Normal range parameter input
7	SOCKET_CH_NOT_INITIALIZE	Control attempts through uninitialized channels	− Check whether the input socket channel is initialized
10	SOCKET_SERVICE_PROCEDURE	Common errors that occur during service processing	− Command operation retry through exception handling − Review input values − System check
11	SOCKET_CLOSED	Attempts to process a command on a socket that has already been	− Check normal connection status

		closed, or a connection that has been aborted by the peer.	
13	CONNECTION_TIMEDOUT	Connection time out occurred	- Check connection status of destination side
15	HOST_DOWN	Host device error	- Check host device status - Check for abnormalities in cables and connection paths
16	HOST_UNREACH	Host connection disable	- Check host device status - Check for abnormalities in cables and connection paths
18	CONNECTION_REFUSED	Connection denied	- Check Connection denied status of destination side

※ Errors occurring outside the list Refer to “SOCKET SERVICE Common Error”

6.10.6 Performs a request to create a UDP socket. (SOCKET_UDPCREATE)

Function Block



Input		
BOOL	Execute	Action on rising edge
USINT	UdpCH	Socket service channel to use with UDP
UINT	LocalPort	Local UDP port number(1~65535)
Output		
BOOL	Done	Function block completion status
BOOL	Busy	During execution of function block
BOOL	Error	Whether an error occurred
UINT	ErrorID	ID in case of error - in normal operation: 0

(1) Function
  (a) Opens the socket service channel ‘UdpCH’ designated by the local UDP port number ‘LocalPort’. (Socket function Socket(), executing Bind()).
  (b) Refer to the example program “14.4 SOCKET SERVICE”.
  (d) Refer to the detailed operation specification “Command operation timing chart”.

Tip

Close the socket service channel opened with this command using the SOCKET_CLOSE command.
The maximum number of socket service channels is 16.
If the port specified by LocalPort is already in use or used for TCP purposes, the use of the port may be temporarily restricted.
Refer to INVALID_ADDRESS_OR_PORT of occurrence error.

(2) Error Status Information

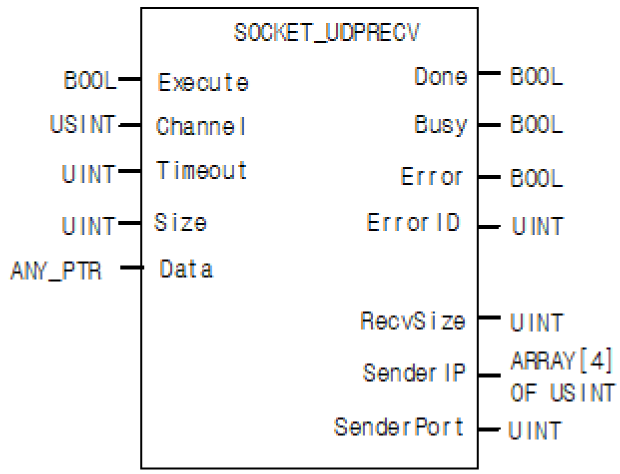
ID	Error name	Cause	Solutions
0	No Error	Normal status	−
1	INVALID_PARAMETER	Command input parameter error	− Normal range parameter input
2	SYSTEM_RESERVED_PORT	Attempt to use the system reserved port	− Change Local Port input value
3	ALREADY_USED_CH	Reassign channels already in use	− Use after checking the usage status of the service channel
10	SOCKET_SERVICE_PROCEDURE	Common errors that occur during service processing	− Command operation retry through exception handling − Review input values

			- System check
12	INVALID_ADDRESS_OR_PORT	Input IP, PORT value abnormal, unavailable status	- When the local port designated port executes a command immediately after TCP connection and disconnection, the error may last up to 60 seconds for a certain period of time to initialize the port used for the existing connection, and retry the command operation through exception handling. - Check whether a port that is already in use is specified - Local Port normal input value, check the range

※ Errors occurring outside the list Refer to “SOCKET SERVICE Common Error”

6.10.7 Receive data from a UDP socket (SOCKET_UDPRECV)

Function Block



Input		
BOOL	Execute	Action on rising edge
USINT	Channel	Socket service channel(UDP)(1~16)
UINT	Timeout	0: no timeout 1~65535 : input value * 1sec
UINT	Size	Save Size(0~1400) Data size to read from receive buffer
ANY_PTR	Data	Received data
Output		
BOOL	Done	Function block completion status
BOOL	Busy	During execution of function block
BOOL	Error	Whether an error occurred
UINT	ErrorID	ID in case of error - in normal operation: 0
UINT	RecvSize	Receive data size(0~1400) Data size received in RecvData
ARRAY[4] OF USINT	SenderIP	Send side IP address(0~255)
UINT	SenderPort	Send side port number(1~65535)

(1) Function

(a) The received data of the socket service channel set in ‘Channel’ is saved in ‘Data’. The size of data to be saved is designated as ‘Size’.

(b) The actual received data size is output in ‘RecvSize’

(c) The IP and Port numbers of the send are stored in ‘SenderIP’ and ‘SenderPort’, respectively.

(d) If there is no data received, it waits for the received data for the time set in the timeout (Timeout).

(e) Refer to the example program “14.4 SOCKET SERVICE”.

(f) Refer to the detailed operation specification “Command operation timing chart”.

Tip

If data is not received and the socket is closed with the SOCKET_CLOSE command, it will be terminated abnormally even if the timeout is less than the timeout period.
Using a socket service channel initialized for UDP purposes
The total size of the data entered in Data must be greater than or equal to the value entered in Size.

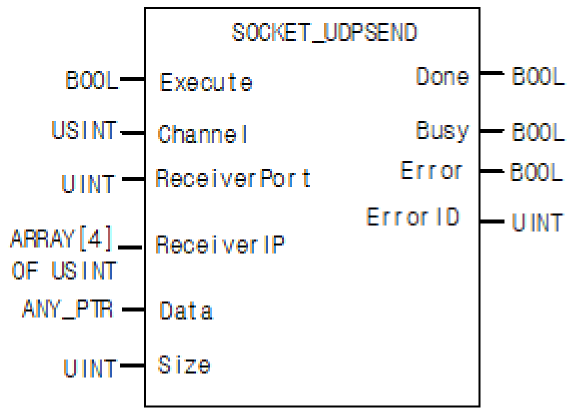
(g) Error Status Information

ID	Error name	Cause	Solutions
0	No Error	Normal status	-
1	INVALID_PARAMETER	Command input parameter error	- Normal range parameter input
4	WAIT_TIMEOUT	Command timeout occurrence	- Check the data send status of the destination side - Wait for send to continue
5	COMMAND_INSTANCE_POOL_FULL	Exceeded command instance usage	- Decrease the frequency of command execution
7	SOCKET_CH_NOT_INITIALIZE	Control attempts through uninitialized channels	- Check whether the input socket channel is initialized
10	SOCKET_SERVICE_PROCEDURE	Common errors that occur during service processing	- Command operation retry through exception handling - Review input values - System check
11	SOCKET_CLOSED	Attempts to process a command on a socket that has already been closed, or a connection that has been aborted by the peer.	- Check normal connection status
13	CONNECTION_TIMEDOUT	Connection time out occurred	- Check connection status of destination side
15	HOST_DOWN	Host device error	- Check host device status - Check for abnormalities in cables and connection paths
16	HOST_UNREACH	Host connection disable	- Check host device status - Check for abnormalities in cables and connection paths
18	CONNECTION_REFUSED	Connection denied	- Check Connection denied status of destination side

※ Errors occurring outside the list Refer to “SOCKET SERVICE Common Error”

6.10.8 Send data from a UDP socket (SOCKET_UDPSEND)

Function Block



Input		
BOOL	Execute	Action on rising edge
USINT	Channel	Socket service channel(UDP)(1~16)
UINT	ReceiverPort	Receive side port number(1~65535)
ARRAY[4] OF USINT	ReceiverIP	Receive side IP address(0~255)
ANY_PTR	Data	Send data
UINT	Size	Send data size(0~1400)
Output		
BOOL	Done	Function block completion status
BOOL	Busy	During execution of function block
BOOL	Error	Whether an error occurred
UINT	ErrorID	ID in case of error - in normal operation: 0

(1) Function

(a) Data is transmitted through the socket service channel set as 'Channel'. The size of data to be sending is designated as 'Size'.

(b) The receiving node is designated as 'ReceiverIP' and 'ReceiverPort'.

(c) Refer to the example program "14.4 SOCKET SERVICE".

(d) Refer to the detailed operation specification "Command operation timing chart".

Tip

Using a socket service channel initialized for UDP purposes
The total size of the data entered in Data must be greater than or equal to the value entered in Size.

(2) Error Status Information

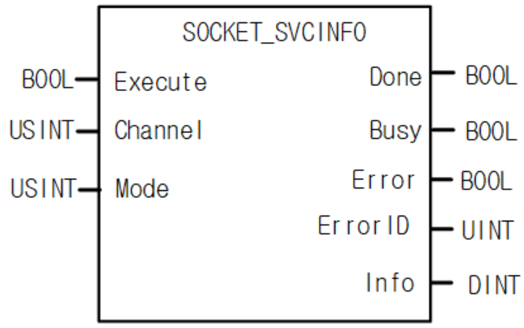
ID	Error name	Cause	Solutions
0	No Error	Normal status	-
1	INVALID_PARAMETER	Command input parameter error	Normal range parameter input

7	SOCKET_CH_NOT_INITIALIZE	Control attempts through uninitialized channels	Check whether the input socket channel is initialized
10	SOCKET_SERVICE_PROCEDURE	Common errors that occur during service processing	● Command operation retry through exception handling ● Review input values System check
11	SOCKET_CLOSED	Attempts to process a command on a socket that has already been closed, or a connection that has been aborted by the peer.	Check normal connection status
13	CONNECTION_TIMEDOUT	Connection time out occurred	Check connection status of destination side
15	HOST_DOWN	Host device error	● Check host device status Check for abnormalities in cables and connection paths
16	HOST_UNREACH	Host connection disable	● Check host device status Check for abnormalities in cables and connection paths
18	CONNECTION_REFUSED	Connection denied	Check Connection denied status of destination side

※ Errors occurring outside the list Refer to “SOCKET SERVICE Common Error”

6.10.9 Check information for each channel of socket service (SOCKET_SVCINFO)

Function Block



Input		
BOOL	Execute	Action on rising edge
USINT	Channel	Socket service channel(1~16)
USINT	Mode	Socket service information read mode setting 0: Channel type 1: Receive buffer remaining data size
Output		
BOOL	Done	Function block completion status
BOOL	Busy	During execution of function block
BOOL	Error	Whether an error occurred
UINT	ErrorID	ID in case of error - in normal operation: 0
DINT	Info	Details according to Mode setting Mode: 0 0: SocketUndefined 1: SocketEstablishing 2: SocketForTcpClient 3: SocketForTcpServer 4: SocketForTcpServerPeer 5 : SocketForUdp Mode: 1 Receive buffer remaining data size

(1) Function
  (a) Acquire detailed information according to the mode of the socket service channel set as 'Channel'.
  (b) Refer to the example program "14.4 SOCKET SERVICE".
  (c) Refer to the detailed operation specification "Command operation timing chart".

Tip

Used after initialization of socket service channel.

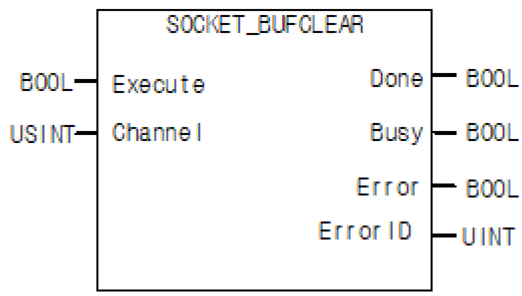
(2) Error Status Information

ID	Error name	Cause	Solutions
0	No Error	Normal status	–
1	INVALID_PARAMETER	Command input parameter error	– Normal range parameter input
10	SOCKET_SERVICE_PROCEDURE	Common errors that occur during service processing	– Command operation retry through exception handling – Review input values – System check

※ Errors occurring outside the list Refer to "SOCKET SERVICE Common Error"

6.10.10 Initializes the receive buffer of a TCP or UDP socket. (SOCKET_BUFCLEAR)

Function Block



Input		
BOOL	Execute	Action on rising edge
USINT	Channel	Socket service channel(1~16)
Output		
BOOL	Done	Function block completion status
BOOL	Busy	During execution of function block
BOOL	Error	Whether an error occurred
UINT	ErrorID	ID in case of error - in normal operation: 0

(1) Function
- (a) Initializes the receive buffer of the socket service channel set as ‘Channel’.
- (b) Initializes the remaining receive buffer at the time of command call.
- (d) Refer to the example program “14.4 SOCKET SERVICE”.
- (e) Refer to the detailed operation specification “Command operation timing chart”.

Tip

Used after initialization of socket service channel.

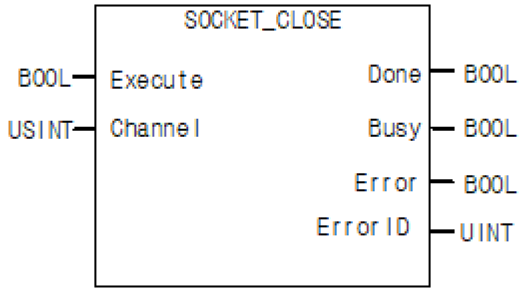
(2) Error Status Information

ID	Error name	Cause	Solutions
0	No Error	Normal status	−
1	INVALID_PARAMETER	Command input parameter error	− Normal range parameter input
7	SOCKET_CH_NOT_INITIALIZE	Control attempts through uninitialized channels	− Check whether the input socket channel is initialized
10	SOCKET_SERVICE_PROCEDURE	Common errors that occur during service processing	− Command operation retry through exception handling − Review input values − System check

※ Errors occurring outside the list Refer to “SOCKET SERVICE Common Error”

6.10.11 Close the socket service channel. (SOCKET_CLOSE)

Function Block



Input		
BOOL	Execute	Action on rising edge
USINT	Channel	Socket service channel(1~16)
Output		
BOOL	Done	Function block completion status
BOOL	Busy	During execution of function block
BOOL	Error	Whether an error occurred
UINT	ErrorID	ID in case of error - in normal operation: 0

(1) Function
  (a) Close the socket service channel set as ‘Channel’.
  (b) Refer to the example program “14.4 SOCKET SERVICE”.
  (c) Refer to the detailed operation specification “Command operation timing chart”.

Tip

Used after initialization of socket service channel.

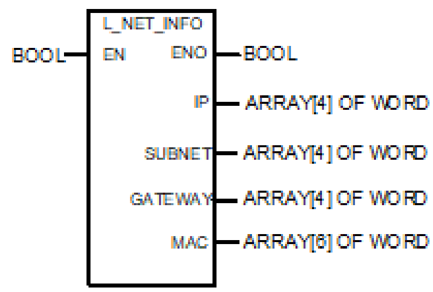
(2) Error Status Information

ID	Error name	Cause	Solutions
0	No Error	Normal status	−
1	INVALID_PARAMETER	Command input parameter error	− Normal range parameter input
7	SOCKET_CH_NOT_INITIALIZE	Control attempts through uninitialized channels	− Check whether the input socket channel is initialized
10	SOCKET_SERVICE_PROCEDURE	Common errors that occur during service processing	− Command operation retry through exception handling − Review input values − System check

※ Errors occurring outside the list Refer to “SOCKET SERVICE Common Error”

6.10.12 Read local Ethernet IP, SUBNET MASK, GATEWAY (L_NET_INFO)

Function Block



Input		
BOOL	EN	Function execution when 1
Output		
BOOL	ENO	Output 1 when executed without error
ARRAY[4] OF WORD	IP	Local Ethernet IP address
ARRAY[4] OF WORD	SUBNET	Local Ethernet subnet mask
ARRAY[4] OF WORD	GATEWAY	Local Ethernet subnet mask
ARRAY[6] OF WORD	MAC	Local Ethernet Mac address

(1) Function

(a) The L_NET_IP command allows you to read the IP address, subnet mask, and gateway information of the local Ethernet.

(b) After executing the command, the IP address of the local Ethernet is displayed as follows.

- Local Ethernet Information Set

  IP ADDRESS: 192.168.0.100
  Subnet: 255.255.255.0
  Gateway: 192.168.0.1
  MAC ADDRESS: 00-16-EA-50-AB-CD

- Display Information

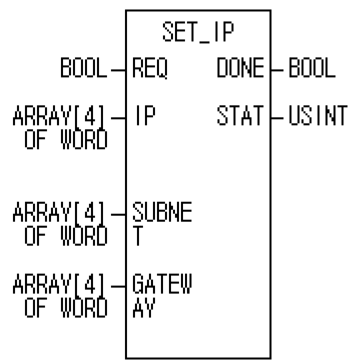
IP[0] : 192 (0x00C0 IP[1] : 168 (0x00A8) IP[2] : 0 (0x0000) IP[3] : 100 (0x00C8) SUBNET[0] : 255 (0x00FF) SUBNET[1] : 255 (0x00FF) SUBNET[2] : 255 (0x00FF) SUBNET[3] : 0 (0x0000)	GATEWAY[0] : 192 (0x00C0) GATEWAY[1] : 168 (0x00A8) GATEWAY[2] : 0 (0x0000) GATEWAY[3] : 1 (0x0001) MAC[0] : 0x0000 MAC[1] : 0x0016 MAC[2] : 0x00EA MAC[3] : 0x0050 MAC[4] : 0x00AB MAC[5] : 0x00CD

(2) Program Example

ST	L_NET_INFO(IP=>(*ARRAY[0..3]_OF_WORD*),SUBNET=>(*ARRAY[0..3]_OF_WORD*), GATEWAY=>(*ARRAY[0..3]_OF_WORD*), MAC=>(*ARRAY[0..5]_OF_WORD*))

6.10.13 Local Ethernet IP, SUBNET MASK, GATEWAY settings (SET_IP)

Function Block



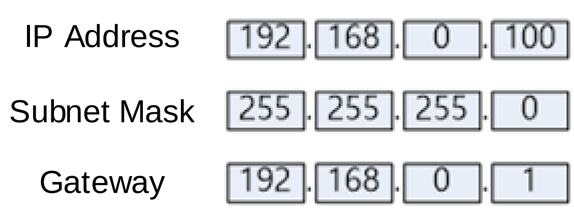
Input		
BOOL	REQ	Function block execution request
ARRAY[4] OF WORD	IP	Local ethernet IP address to set
ARRAY[4] OF WORD	SUBNET	Local ethernet subnet mask to set
ARRAY[4] OF WORD	GATEWAY	Local ethernet gateway to set
Output		
BOOL	DONE	Maintain 1 after initial operation
USINT	STAT	Complete and ERR information

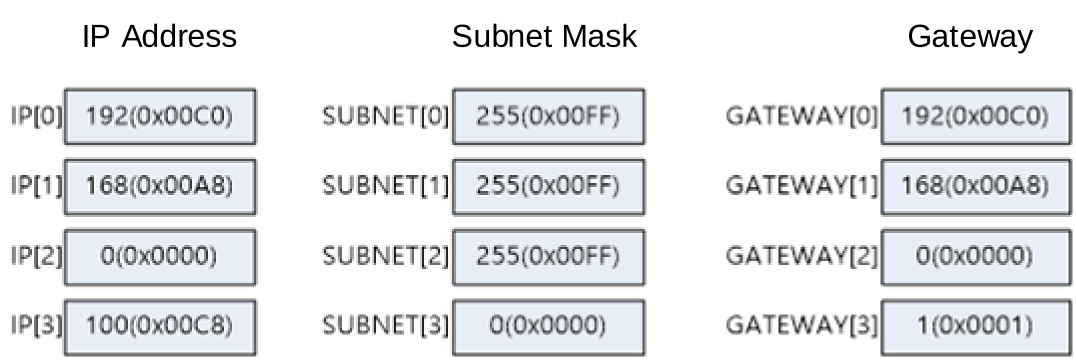
(1) Function

(a) The SET_IP command allows you to set the IP address, subnet mask, and gateway of the local Ethernet.

(b) Only available with XGI-CPUUN with local Ethernet.

(c) When setting the IP address, subnet mask, and gateway, you need to set the IP address, subnet mask, and gateway as shown below.





(2) Error Status Information

If the local Ethernet parameter is abnormal or the command is duplicated, the following error may occur

STAT	Contents	Details
0	Normal performance	Command execution complete
11	Above user setting value	User set IP / SUBNET / GATEWAY setting value is not valid
12	Above the default setting	Above existing local Ethernet parameter setting (Local Ethernet parameters have never been downloaded or parameter errors are present)
13	Duplicate request error	If the instruction is already being executed (The instruction can not be duplicated)
14	Timeout	Timeout processed because command execution is not completed

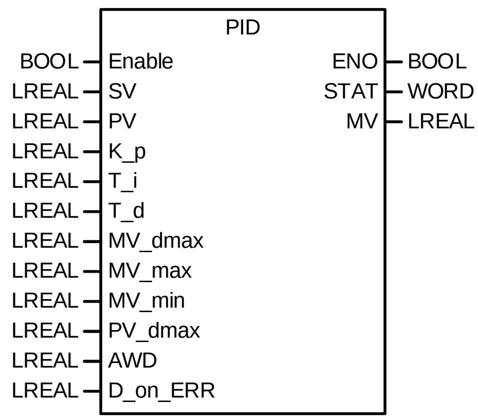
(3) Program Example

ST	INST_SET_IP(REQ:=REQ_BOOL, IP:=ARY_IP, SUBNET:=ARY_SUBNET, GATEWAY:=ARY_GATEWAY, DONE=>DONE_BOOL,STAT=>STAT_USINT)

6.11 Other Commands

6.11.1 PID calculation (PID)

Function Block



Input		
BOOL	Enable	Execute the fuction block
LREAL	SV	Set value (SV)
LREAL	PV	Present value (PV)
LREAL	K_p	P constant (K_p)
LREAL	T_i	I constant (T_i)[sec]
LREAL	T_d	D constant (T_d)[sec]
LREAL	MV_dmax	MV variation limit
LREAL	MV_max	MV maximum limit
LREAL	MV_min	MV minimum limit
LREAL	PV_dmax	PV variation limit
BOOL	AWD	Anti Wind-up prohibit (0: operation, 1: prohibit)
BOOL	D_on_ERR	DSelect Differential Calculation Source (0:PV, 1:ERR)
Output		
BOOL	ENO	Display that PID calculation is executing normally.
WORD	STAT	PID status alarm
LREAL	MV	Manupulate value (MV)

(1) The function block receives the set value (SV) and the present value (PV) of control object, executes PID calculation to output Manupulated value.

(2) The set value input is current status of control object. The status is expressed as a number, you should convert to PV and input according to system gain. For example, in a system where PV is sensed as 5000 when the temperature is 50℃, SV is set to 5000 when controlling as the temperature to 50℃.

(3) The present value input is an the indicator of the current status of control object, in general, the input from the sensor saved in the device of CPU through an input devices such as A/D conversion module, and the value must be transfered and inputed as the PV input.

(4) The K_p input set the proportional constant of the current PID calculation. K_p is multipled by P,I,D(proportional,integral, differential) term of the PID control, if K_p is increase, proportion, differential effect are increase and the integral effect is decrease. In particular, if K_p input is 0, PID control is not operation.

(5) The T_i input set the integral time constant of the relevant loops. T_i is divided by I (integral) term of the PID control, if T_i is increase, the integral effect is decrease. In particular, if T_i input is 0, I control is not operation.

(6) The T_d input sets the differential time constant of the relevant loops. T_d is multiplied by the D (differential) term of the PID control, if T_d is increase, the differential effect is increases. If T_d input is 0, D control is not operation.

(7) The PV_max input limits PV variation of the loop. In actuall control, the PV does not always reflect exact status of the system. The unwanted signal such as malfunction of sense, noise or disturbance can be mixed and reflected in PV. The PV change abruptly and casues a large change in PID output. To prevent this situation, when the PV changes more than setting value in _PID[B]_[L]dPV_max, at first it prevent change more than setting value. On the other hand, when PV_dmax set in too small, change of the system is slowly reflected and the convergence time takes longer, set to meet the system features. In particular, when the setting value set to “0”, the PV variation limit function does not operate.

(8) The MV_max input limits MV variation of the loop. Prevent overload and block the syatem error by limit maximum value of controller output transferring to output devices. Also transferring the unwanted value like overflow is prevented.

(9) The MV_min input limits MV minimum value of the loop. Prevent the syatem error previously by limit mimimum value of controller output transferring to output devices. Also transferring the unwanted value like overflow is prevented.

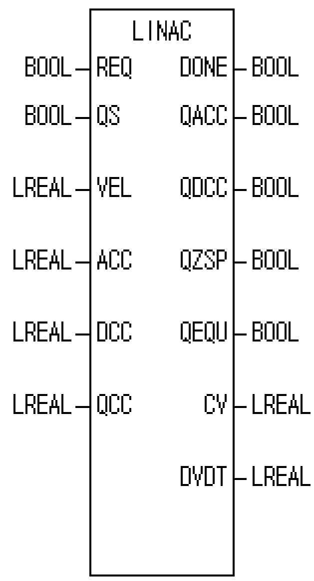
(10) D_on_ER input sets the D calculation source of the PID loop to ERR. D calculation is calculated with ERR or PV, when calculating D calculation using ERR, the D response is abruptly changed by the user, so excessive input can be applied to to driver instantly. To prevent this, the method using PV in D calculation is applied and the default value is also set to execute D calculation using PV. If using ERR without this algorism, this bit is turned On. If the bit is off, the PID executes D calculates with the PV value, if it is on,it executes D calculation with EER value.

(11) The AWD input is the input to activate or deactivate the Anti Wind-up function. If the input turns on, the Anti Wind-up function is disabled.

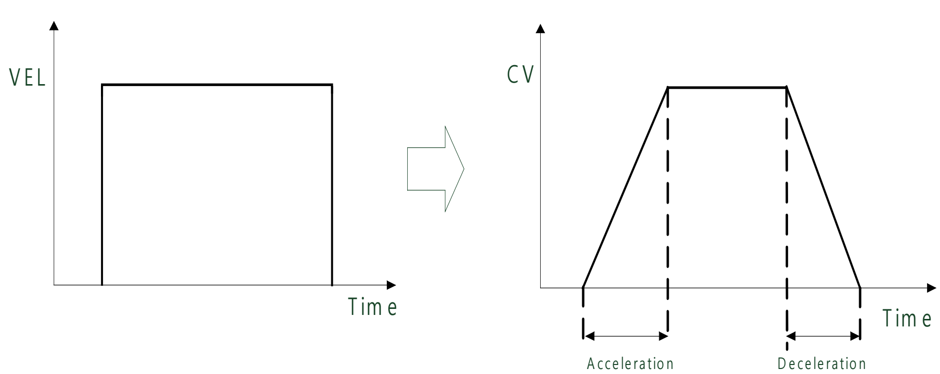
(12) Each bits of STAT output displays the status or abnormal status of the PID controller. Each bit is on only at the moment the opearion occurs and returns to off when the operation is canceled. Lower 8 bits of STAT display each error status of the loops, and the upper 8 bits display control status of the loops. The assignment of each bits are as follows.

Bit	Status
0	Alarm when skipping operation because T_s setting is too small.
2	Alarm that the PV variation is limited.
3	Alarm that the MV variation is limited.
4	Alarm that the MV maximum is limited.
5	Alarm that the MV minimum is limited.
8	Alarm that PID calculation is in operation.
15	Alarm that the Anti Wind-up is in operation during PID operation.

6.11.2 LINAC (Accelerator/Decelerator command)

Function Block		
		
Input		
BOOL	REQ	Execute the function block
BOOL	QS	Input emergency stop
LREAL	VEL	Set the target speed [u/s].
LREAL	ACC	Set acceleration [u/s^2].
LREAL	DCC	Set deceleration [u/s^2].
LREAL	QCC	Set emergency stop deceleration [u/s^2].
Output		
BOOL	DONE	Function block completion status
BOOL	QACC	Display whether accelerating.
BOOL	QDCC	Display whether decelerating.
BOOL	QZSP	Display whether zero speed of current speed.
BOOL	QEQU	Display whether the target speed matches the current speed.
LREAL	CV	Display the current speed.
LREAL	DVDT	Display current acceleration / deceleration.

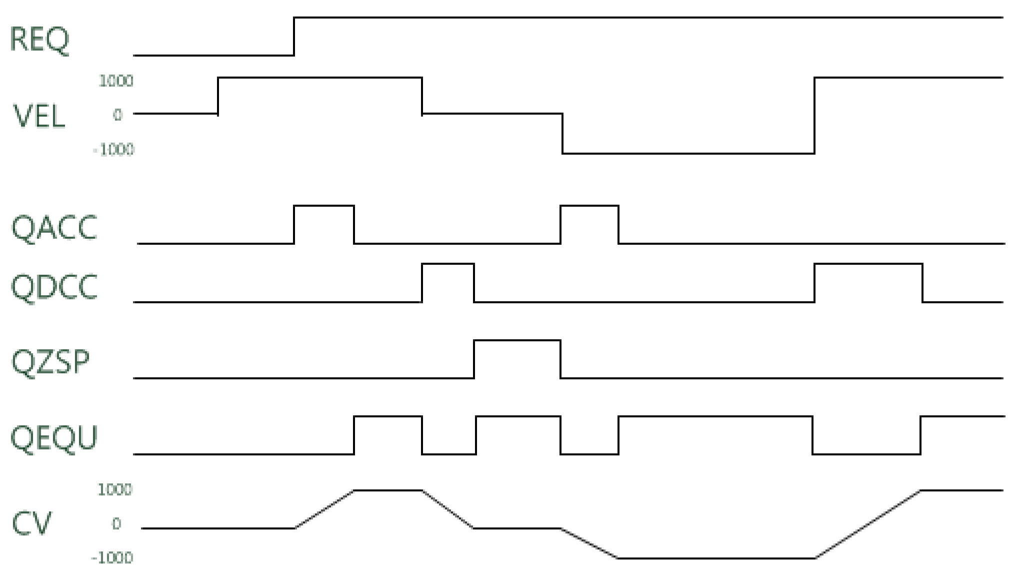
(1) The function block outputs the speed value reached by applying constant acceleration/deceleration to input speed to reach.



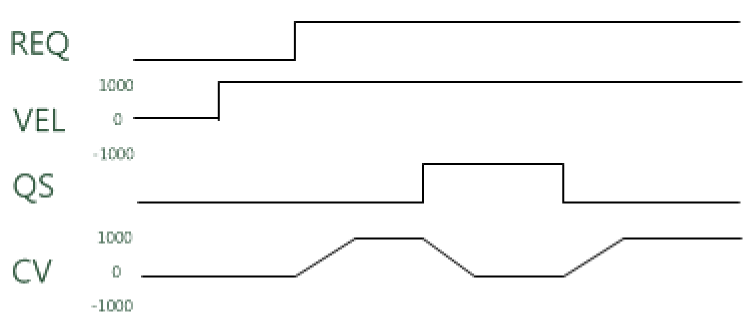
(2) REQ input is using ACC/DCC/QCC value of function block in rising edge and ACC/DCC/QCC value does not change during

operation.

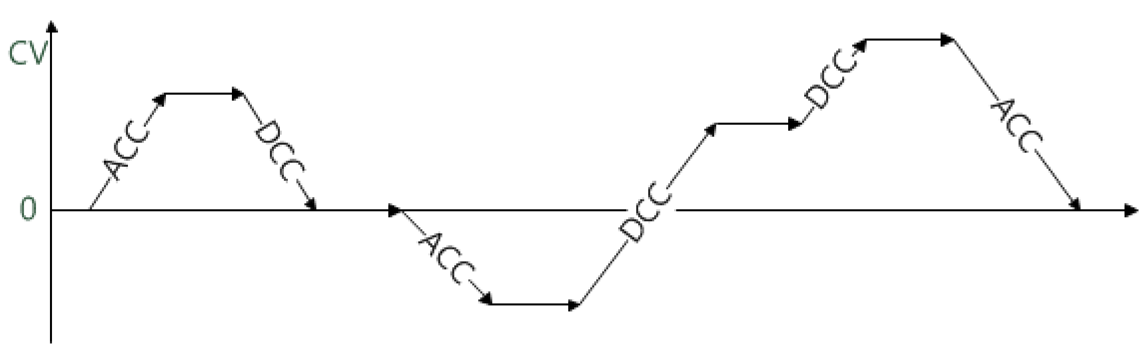
(3) QACC/QDCC/QZSP/QEQU output is as follows during operation.



(4) If QS value is “1” decelerate (rapid deceleration) as deceleration speed set in QCC. If QS value changes to “0” release deceleration and acelerate/decelerate as inputted target speed.

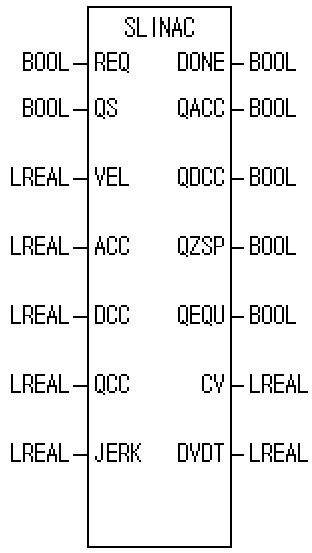


(5) If the speed of stop status is “0” accelerate to the direction of inputted target speed and decelerate reverse direction. If the stop operation is operation at 0 speed, the direction of acceleration/deceleration is changed.



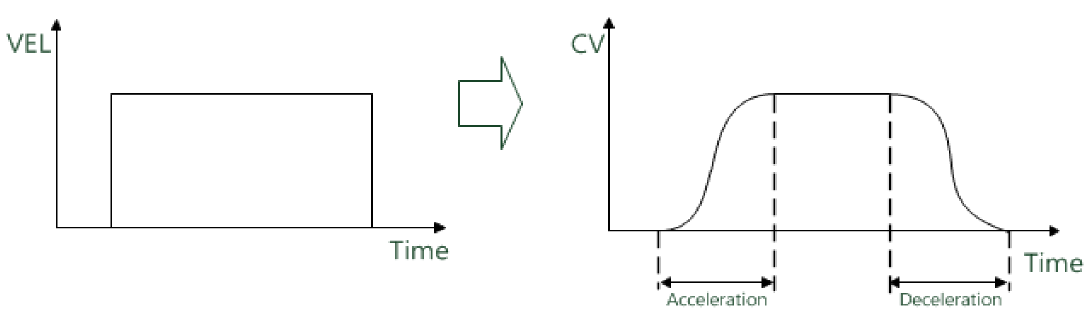
6.11.3 SLINAC (S-curve Accelerator/Decelerator command)

Function block



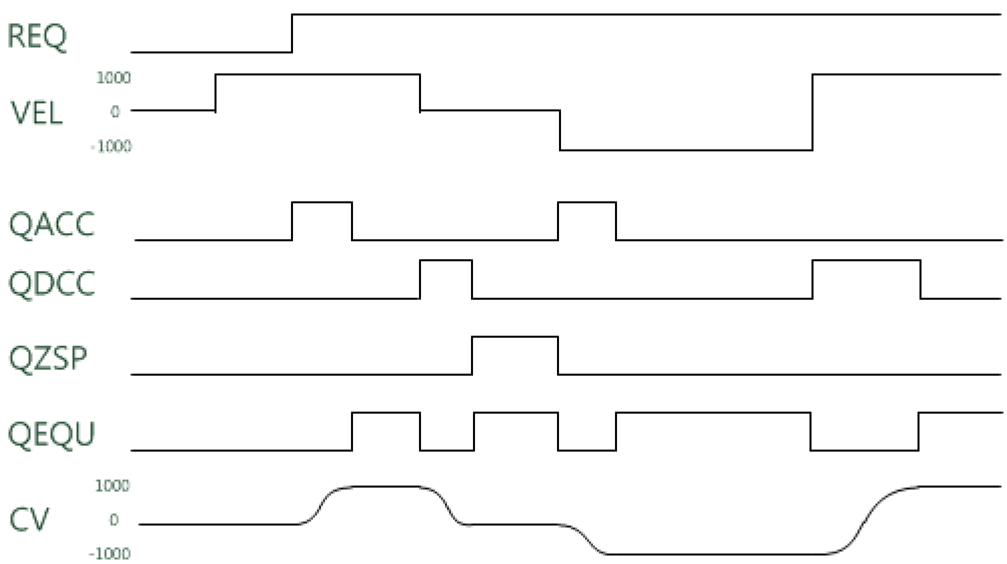
Input		
BOOL	REQ	Execute the function block
BOOL	QS	Input emergency stop
LREAL	VEL	Set the target speed [u/s].
LREAL	ACC	Set acceleration [u/s^2].
LREAL	DCC	Set deceleration [u/s^2].
LREAL	QCC	Set emergency stop deceleration [u/s^2].
LREAL	JERK	Set acceleration/deceleration change rate. [u/s^3]
Output		
BOOL	DONE	Function block completion status
BOOL	QACC	Display whether accelerating.
BOOL	QDCC	Display whether decelerating.
BOOL	QZSP	Display whether zero speed of current speed.
BOOL	QEQU	Display whether the target speed matches the current speed.
LREAL	CV	Display the current speed.
LREAL	DVDT	Display current acceleration / deceleration.
BOOL	DONE	Display completion status of function block.

(1) The function block outputs the speed value reached by appling acceleration/deceleration with JERK up to the input speed.



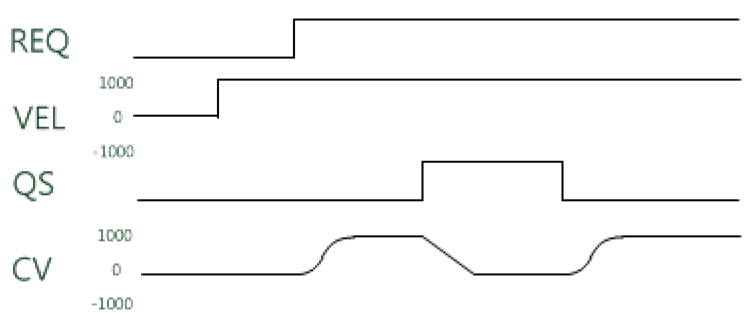
(2) REQ input is using ACC/DCC/QCC value of function block in rising edge and ACC/DCC/QCC value does not change during operation.

(3) QACC/QDCC/QZSP/QEQU output is as follows during operation.



(4) If changing the target speed before reaching target speed, the overshoot or undershoot can be occurred.

(5) If QS value is “1” decelerate (rapid deceleration) as deceleration speed set in QCC. If QS value changes to “0” release deceleration and acelerate/decelerate as inputted target speed.



Chapter 7 Program

7.1 Program Configuration and Operation Method

The program of the motion controller is divided into main task program, periodic task program and initialization task program.
The features of each program in execution are as follows.

7.1.1 Configuration of program

The motion controller's initialization, main and periodic task programs are executed based on the cycle. Each task has a fixed cycle and is set by the user in the default parameters. There are two configurable cycles: main task cycle and periodic task cycle. The initialization task adopts the cycle of the main task.

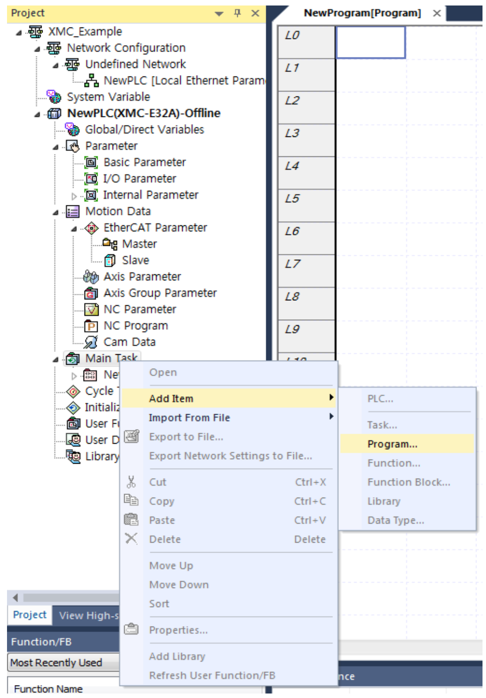
Classification	Content
Initializing task program	• The initialization task program is the first task program executed after the motion controller completes its own initialization required for operation when power is applied. It runs until the INIT_DONE command is executed. • When the initialization program is executed, only the initialization program is executed and the main task program and the periodic task program do not run until the INIT_DONE command is executed. • Even while the initialization task program runs, I / O Refresh and other functions are executed normally. • The initialization task program is used to program various operations required for initial setting of the motion controller.
Main task program	• This program is executed at intervals of the main task's cycle set in the motion controller. • The main task's cycle can be set in the "Main Task Cycle" of the basic parameters, and you can select one among 500 μs, 1 ms, 2 ms, or 4 ms. • When the run time of the main task program exceeds the set main task cycle, the cycle warning occurs. If the main task program is not completed during detecting the cycle error, the cycle error occurs.
Periodic task program	• The program is executed every periodic task cycle set in the motion controller. • The periodic task cycle can be set in the "Periodic Task Cycle" of the basic parameters and must be set to a multiple of the configured main task cycle. • The periodic task program runs in the spare time after the motion controller executes the main task program every main task cycle and then, it runs repeatedly every periodic task cycle.

For more details on the execution of the main task program and periodic task program, refer to "4.3 Motion control task".

7.1.2 How to Set the Program

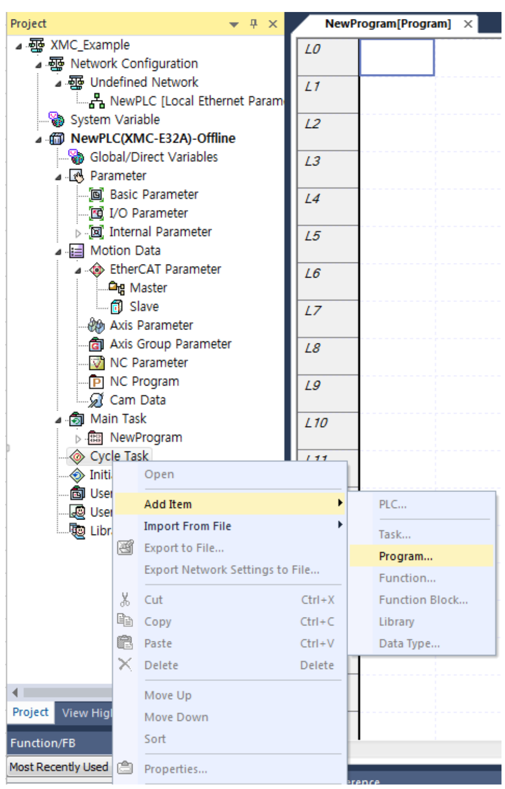
1) How to set the main task program

In the main task location, click the right mouse button and click 『Add item』-『Program』.
(However, when creating the project, the main task program is already created.)



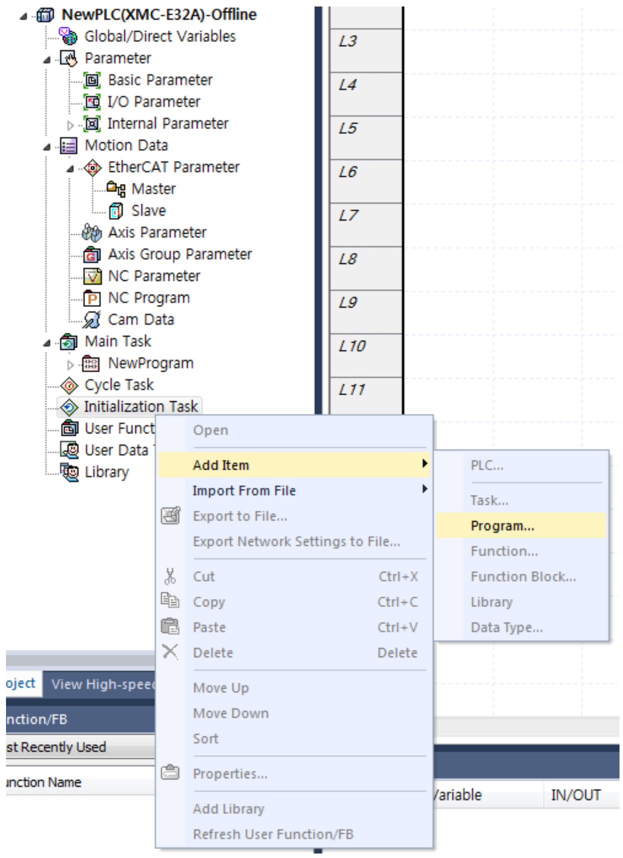
2) How to set the cycle task program

In the cycle task location, click the right mouse button and click 『Add item』-『Program』.

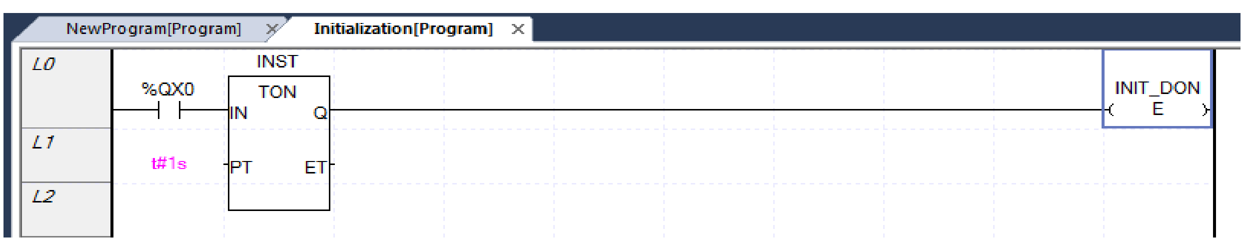


3) How to set the initialization task program

(1) In the initialization task location, click the right mouse button and click 『Add item』-『Program』.



(2) Create the necessary initialization program. Make sure to write the INIT_DONE command in the initialization task program. (When the operating condition of the INIT_DONE is executed, the initialization task is terminated and the scan program will run.)

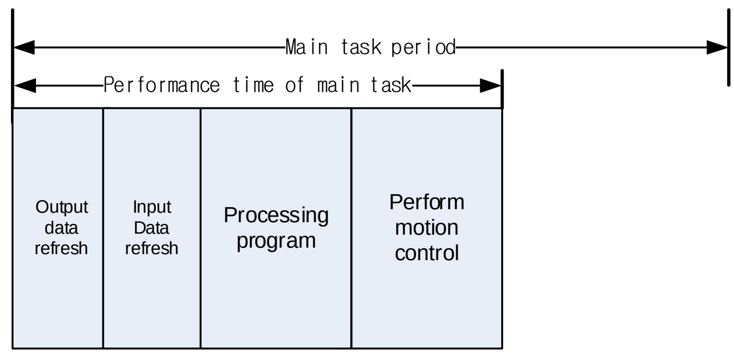


7.1.3 Run Time of the Program

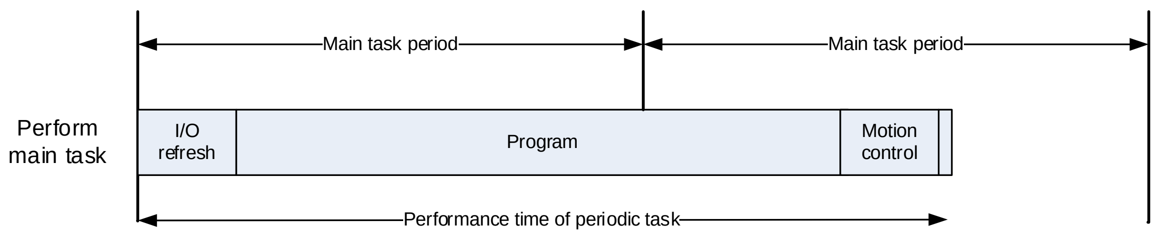
The execution time (scan time) of each task program is calculated as follows. It means the time required from the start of each control cycle to the time when the motion control execution is done, that is, the time required to complete the task.

1) Run time of the main task

It means the time from the start of the main task cycle to the time when the motion control execution is done.

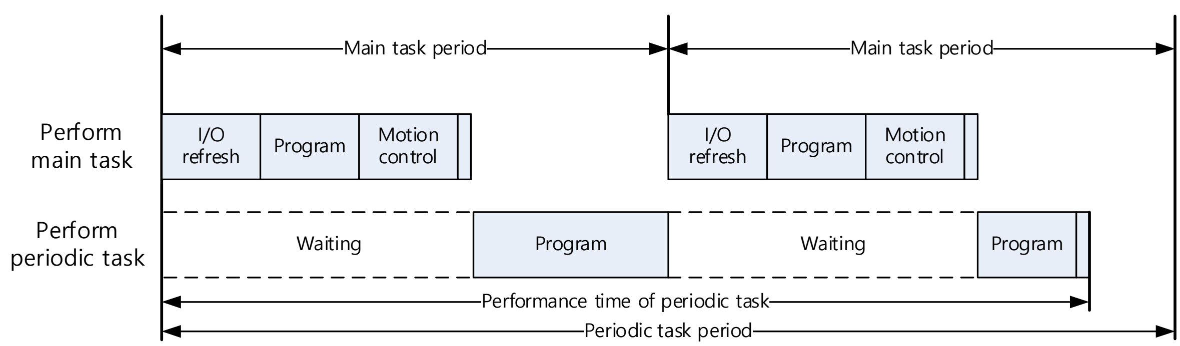


If the main task is not completed for one cycle, just measure the time from the start of the main task cycle to the time when the main task is completed as shown below.



2) Run time of the periodic task

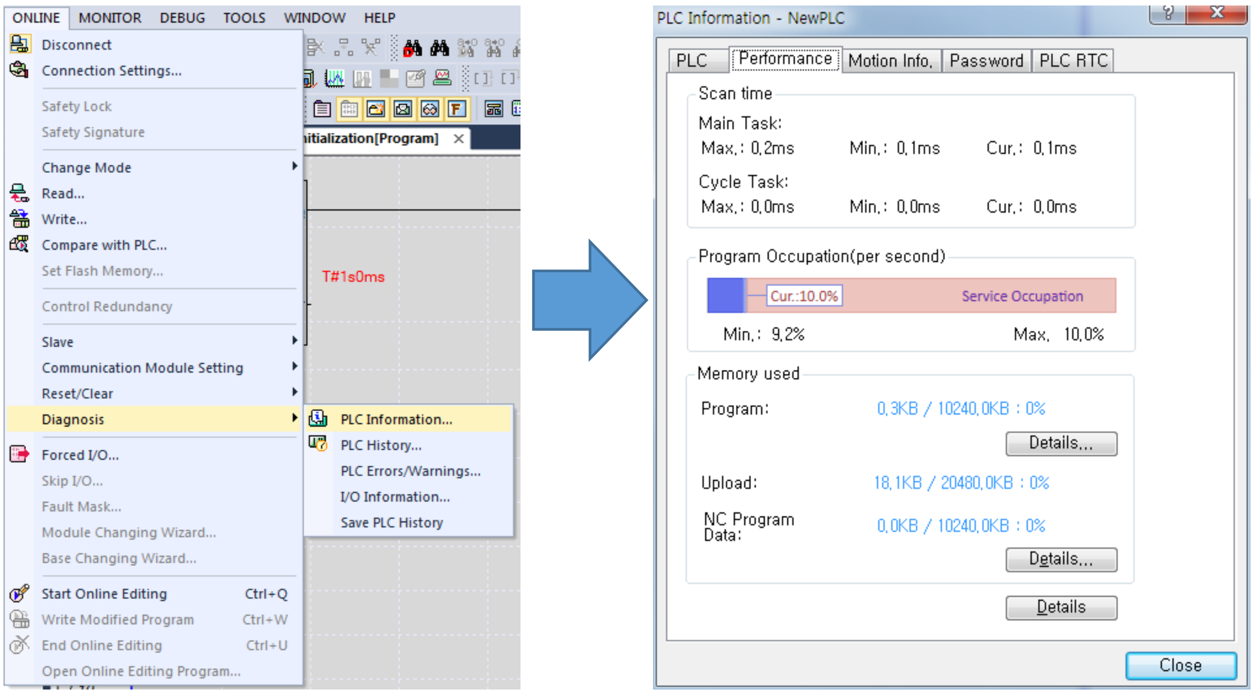
It means the time from the start of the periodic task cycle to the time when the periodic task program is done. The periodic task can be executed in several main task cycles because it runs after the main task is completed.



3) How to check the task run time

The task run time of the motion controller can be checked by using XG5000 or the flag as follows.

(1) Using XG5000: Click 『Online』 - 『Diagnosis』 - 『PLC Information』 - 『Performance』.



(2) Using the flag: The scan time is stored in the system flag (F) area below.

WORD	Flag Name	Description
%FW512	_PTASK_SCAN_MAX	Maximum scan time of main task program (Unit:100us)
%FW513	_PTASK_SCAN_MIN	Minimum scan time of main task program (Unit:100us)
%FW514	_PTASK_SCAN_CUR	Current scan time of main task program (Unit:100us)
%FW515	_CTASK_SCAN_MAX	Maximum scan time of periodic task program (Unit:100us)
%FW516	_CTASK_SCAN_MIN	Minimum scan time of periodic task program (Unit:100us)
%FW517	_CTASK_SCAN_CUR	Current scan time of periodic task program (Unit:100us)

7.2 Status Information Reading

In the program of motion control modules, each axis, status of axis group and operating status of the motion control module can be checked with the flag.

Most of the program examples of chapter 7 is created using flags that indicate axis and status of axis group.

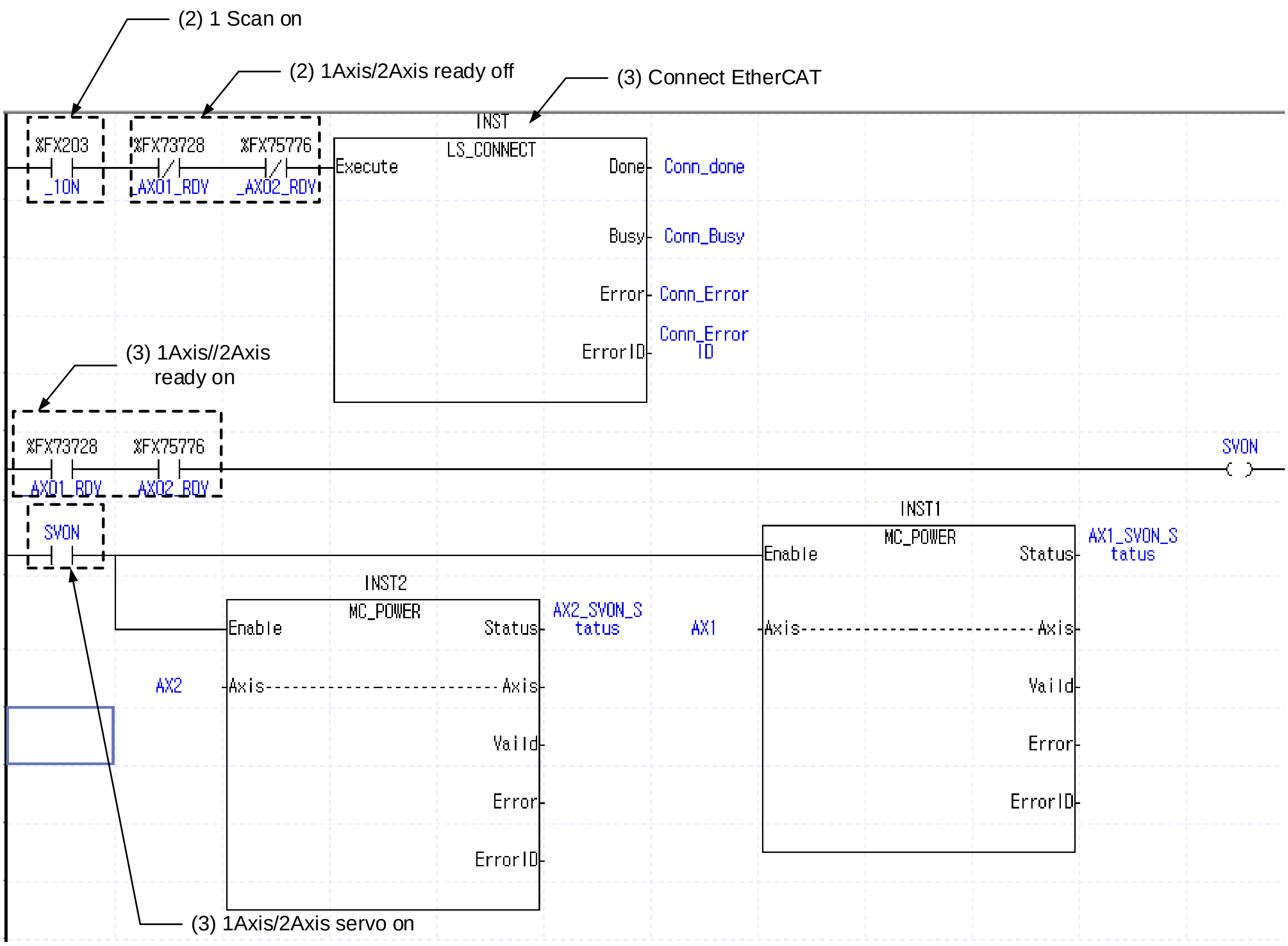
Flags that indicate the status information can be used directly in the program.

For more information on the types and functions of flags, refer to “Appendix 1. Flag list”.

7.3 Discrete Motion Program

7.3.1 Operation Ready

These are example programs that make access to servo drive connected with Ethernet cable and get the connected servo drive to be On to operate EtherCAT servo drive.



(1) The above examples assume situation in which two axes of 1-axis and 2-axis are connected to the motion control module. Assume the axis is connected to an EtherCAT slave and this slave is a servo drive.

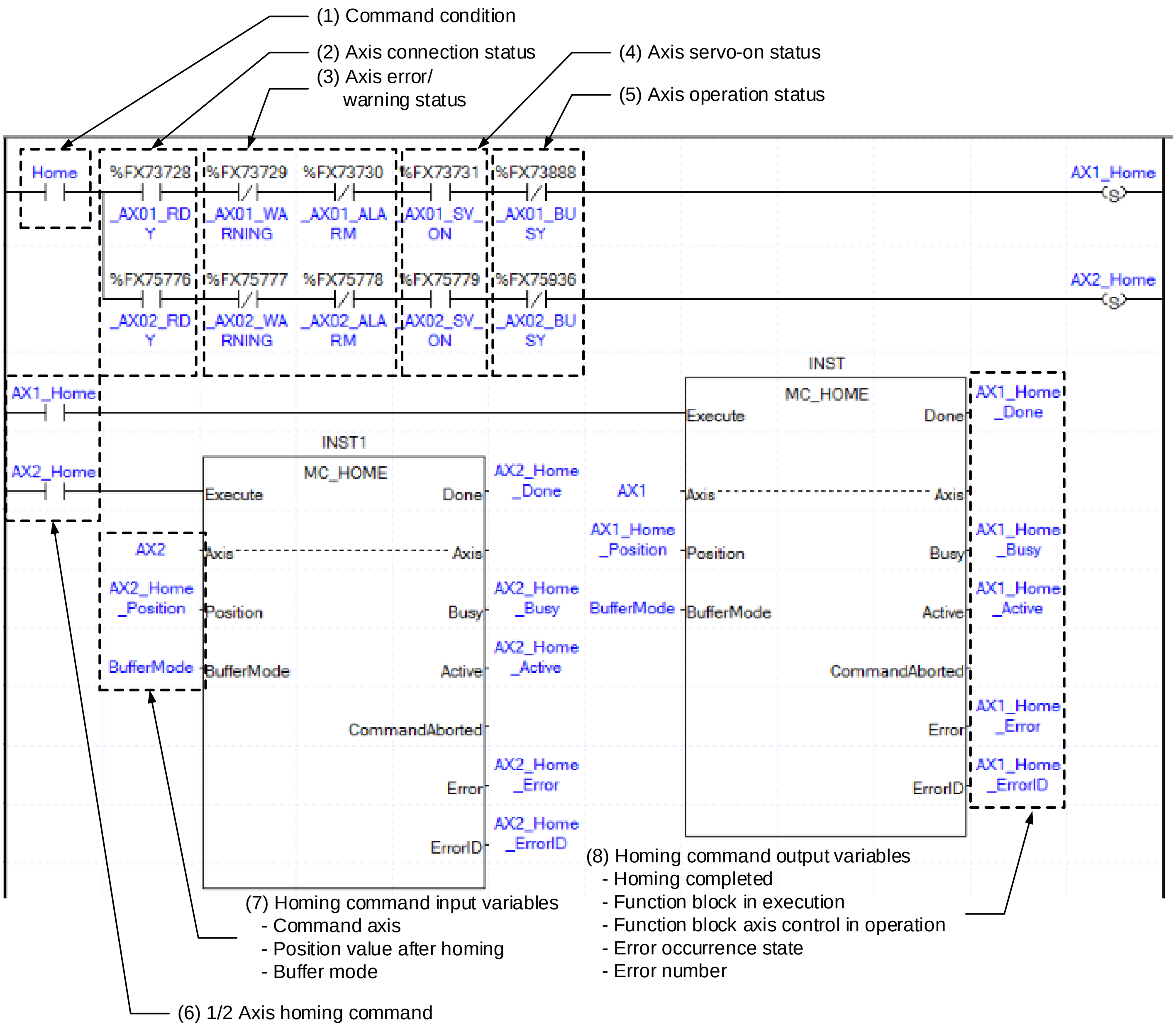
(2) In case 1-axis and 2-axis are not connected when the motion control module enters the RUN, start the connection of EtherCAT communication between motion control module and servo drive using motion function block for communication connection (LS_CONNECT).

(3) If the connection of EtherCAT communication between motion control modules and servo drives is normally performed, servo On/Off (MC_Power) command is issued to each axis by getting “SVON” contact to be On.

(4) In case there is no error in servo drive of the connected 1-axis and 2-axis, the servo is normally On, and it is ready to operate 1-axis and 2-axis.

7.3.2 Return to Origin Point operation

Homing is carried out to set the origin of the machine after the power is applied. Since homing is performed in the servo drive, homing methods may vary depending on servo drive manufacturers. In motion control module, the completion of homing command and error situation is monitored, and the position of the origin after homing is applied to control.



(1) Command condition

: It is a condition to make the axis perform homing operation.

(2) Axis connection status flag

: When the axis to be operated is connected to motion control module, and EtherCAT communication with motion control module is normally performed, it is On.

(3) Axis error/warning status flag

: If there are errors and warnings in the axis, it is On.

(4) Axis servo-on status flag

: If the axis is in servo-on state, it is On, and servo-off state, it becomes Off.

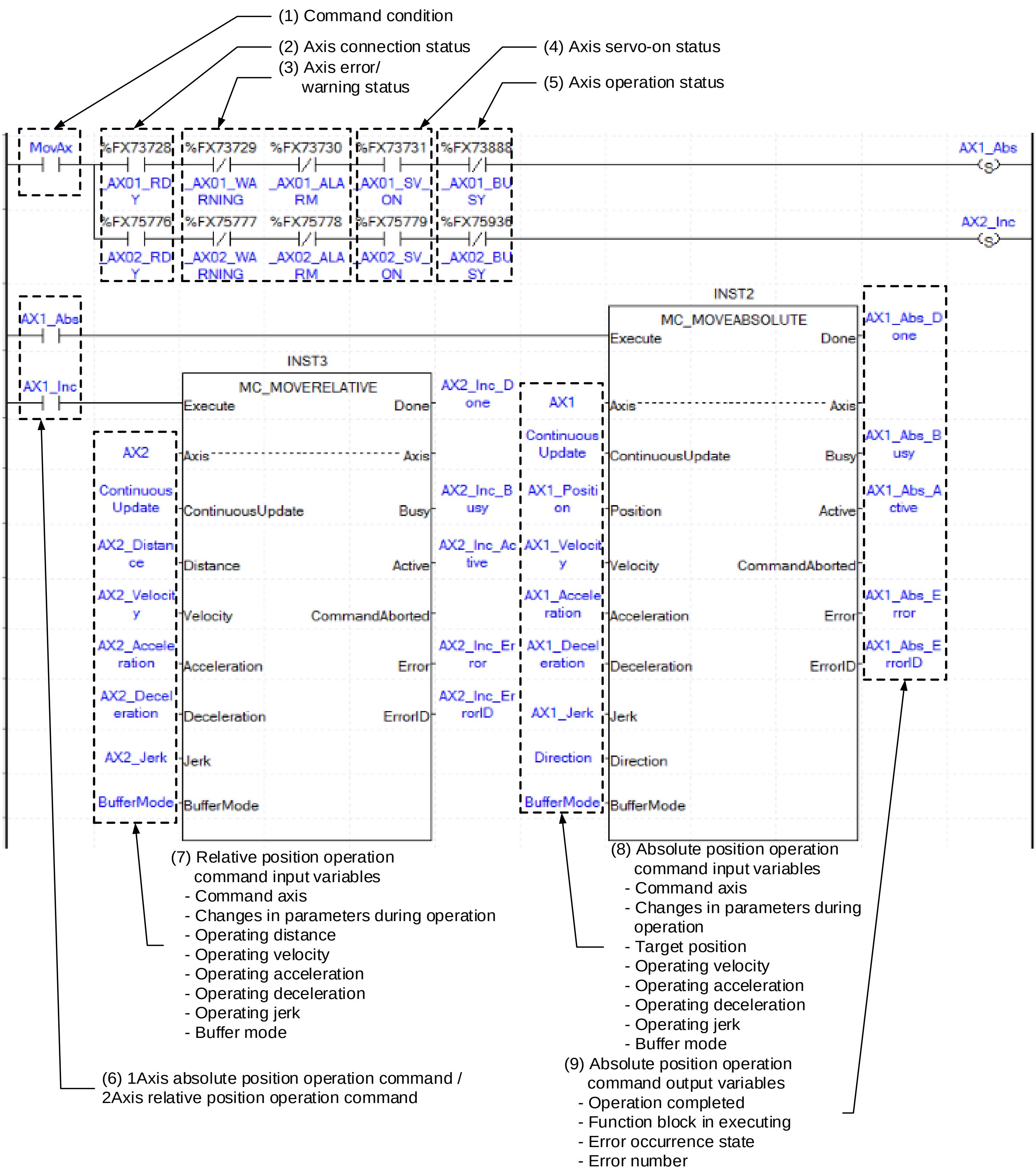
(5) Axis operation status flag
: If the axis is in operation, it is On.
(6) 1/2 axis homing command
: In example programs, homing (MC_Home) motion function block is performed under the following conditions.
- Homing condition is On
- The axis is normally connected
- There should be no errors and warnings.
- Servo-on state
- Not in operation.

Conditions to execute motion function block may vary depending on systems.
(7) Homing command input variables
: These are input variables to perform homing (MC_Home) motion function block.
- Command axis: It sets the axis in which motion function block is executed.
- Position value after homing: It sets the position value when homing is completed.
- Buffer mode: It sets the point of time when motion function block is executed. That is, it sets whether to execute immediately or execute after the completion of commands which are currently being performed. For more details on Function Block execution mode, refer to “6.1.4 Buffer Mode input”.

(8) Homing command output variable
: It is a variable to store output value generated when homing (MC_Home) motion function block is executed.
- Homing completed: If homing operation is completed, it is On.
- Function Block in execution: If motion function block is being performed, it is On, and homing completion is On, it is Off.
- Function Block axis control in operation: In case motion function block is controlling the axis, it is On.
- Error occurrence state: In case error occurs while motion function block is being executed, it is On.
- Error number: In case error occurs, the number that corresponds to error is generated.
- For more details on the output of motion function block, refer to “Edge operation motion function block” of “6.1.3 basic I/O variable”.

7.3.3 Absolute / Relative Operation

It is a program for absolute position and relative position operation using motion control module. The absolute positon is based on the origin and, and relative position the current position.



(1) Command condition
: It is a condition to make the axis perform position control operation.
(2) Axis connection status flag
: When the axis to be operated is connected to motion control module, and EtherCAT communication with motion control module is normally performed, it is On.
(3) Axis error/warning status flag
: If there are errors and warnings in the axis, it is On.
(4) Axis servo-on status flag
: If the axis is in servo-on state, it is On, and servo-off state, it becomes Off.
(5) Axis operation status flag
: If the axis is in operation, it is On.
(6) 1-axis absolute position operation / 2-axis relative position operation commands
: In example programs, absolute position operation (MC_MoveAbsolute) is performed in 1-axis, and relative position operation (MC_MoveRelative) in 2-aixs under the following conditions.
- The axis operation condition is On.
- The axis is normally connected
- There should be no errors and warnings.
- Servo-on state
- Not in operation.

Conditions to execute motion function block may vary depending on systems.

(7) Relative position operation command input variables
: These are input variables to perform relative position operation (MC_MoveRelative) motion function block.
- Command axis: It sets the axis in which motion function block is executed.
- Changes in parameters during operation: It sets whether to apply to the operation by changing the input variables of motion function block. For more information, refer to “6.1.5 Changes in parameters during execution of motion function block”.
- Operating distance: It sets distance to perform relative coordinate operation. Based on the current position, + value means forward direction, and – value means reverse direction value.
- Operating velocity: It sets velocity to perform relative coordinate operation.
- Operating acceleration, operating deceleration, operating jerk: It sets values to be applied in speed control operation respectively.
- Buffer mode: It sets the point of time when motion function block is executed. That is, it sets whether to execute immediately or execute after the completion of commands which are currently being performed. For more details, refer to “6.1.4 Buffer Mode input”.

(8) Absolute position operation command input variables
: These are input variables to perform absolute position operation (MC_MoveAbsolute) motion function block.
- Command axis: It sets the axis in which motion function block is executed.
- Changes in parameters during operation: It sets whether to apply to the operation by changing the input variables of motion function block. For more information, refer to “6.1.5 Changes in parameters during execution of motion function block”.
- Target position: It sets the position that moves to absolute coordinate operation.
- Operating velocity: It sets the velocity when absolute position operation is performed to the target position.
- Operating acceleration, operating deceleration, operating jerk: It sets values to be applied in absolute coordinate operation respectively.
- Operating direction: It sets direction when moving to the target position. In case of 1, movement to the target position is made through forward direction operation, in case of 2, operation is made in the direction that can reach the target area in the shortest distance based on the current position, in case of 3, reverse direction, and in case of 4, movement to the target position is made through operation in the direction of the current operation.
- Buffer mode: It sets the point of time when motion function block is executed. That is, it sets whether to execute immediately or execute after the completion of commands which are currently being performed. For more details, refer to “6.1.4 Buffer Mode input”.

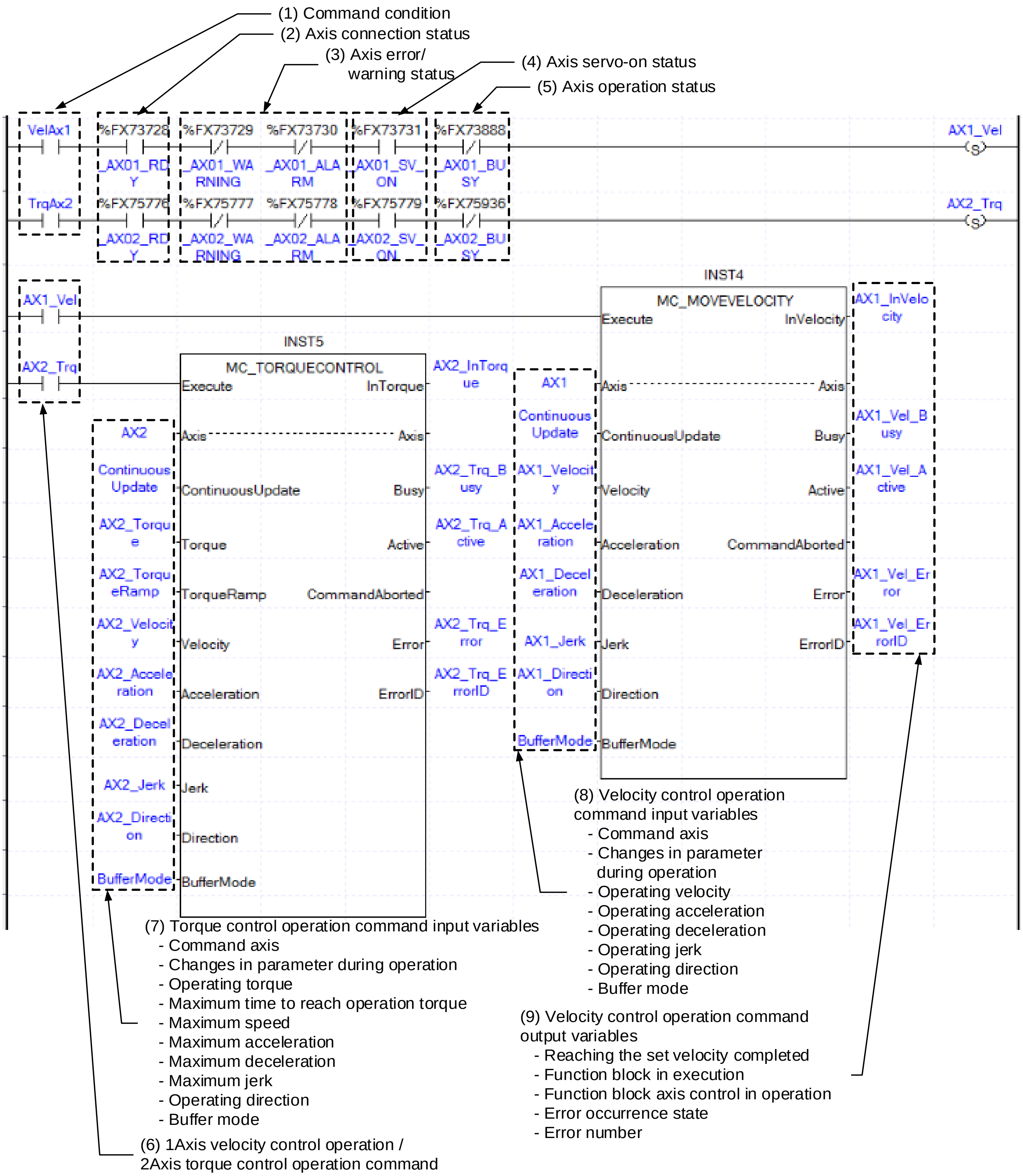
(9) Absolute position operation command output variable

: It is a variable to store output values generated when absolute position operation (MC_MoveAbsolute) motion function block is executed.

- Operation completed: When absolute coordinate operation is completed, it is On.
- Function Block in execution: When motion function block is executed, it is On, and operation completion is On, it becomes Off.
- Function Block axis control in operation: In case motion function block is controlling the axis, it is On.
- Error occurrence state: In case error occurs while motion function block is being executed, it is On.
- Error number: In case error occurs, the number that corresponds to error is generated.
- For more details on the output of function block, refer to “Edge operation motion function block” of “6.1.3 Basic input and output variables”.

7.3.4 Speed/ Torque Control Operation

These are example programs for speed control and torque control operation using motion control modules. In case of the torque control, torque control of servo drive is used, and in motion control module, command for executing torque control is issued, and execution completion and status is monitored.



(1) Command condition

: It is a condition to make the axis perform speed control/torque control operations.

(2) Axis connection status flag

: When the axis to be operated is connected to motion control module, and EtherCAT communication with motion control module is normally performed, it is On.

(3) Axis error/warning status flag

: If there are errors and warnings in the axis, it is On.

(4) Axis servo-on status flag

: If the axis is in servo-on state, it is On, and servo-off state, it becomes Off.

(5) Axis operation status flag

: If the axis is in operation, it is On.

(6) 1-axis speed control operation/ 2-axis torque control operation commands

: In example programs, specified velocity operation (MC_MoveVelocity) motion function block is executed in 1-axis, and toque control operation (MC_TorqueControl) motion function block is executed in 2-axis under the following conditions.

- The axis operation condition is On.
- The axis is normally connected
- There should be no errors and warnings.
- Servo-on state
- Not in operation.

Conditions to execute motion function block may vary depending on systems.

(7) Torque control operation command input variables

: These are input variables to execute torque control operation (MC_TorqueControl) motion function block.

- Command axis: It sets the axis in which motion function block is executed.
- Changes in parameters during operation: It sets whether to apply to the operation by changing input variable values of the motion function block. For more information, refer to “6.1.5 Changes in parameters during execution of motion function block”.
- Operation torque: It sets torque values in torque control operation.
- The maximum time to reach operation torque: It sets the maximum slope from the current torque until changed to the set torque. Its unit is [Unit/s].
- Maximum speed, maximum acceleration, maximum deceleration, and maximum jerk: Not used.
- Operating direction: It sets direction to be operated with torque control. In case of 1, it operates in forward direction and in case of 2, in reverse direction.
- Buffer mode: It sets the point of time when motion function block is executed. That is, it sets whether to execute immediately or execute after the completion of commands which are currently being performed. For more details, refer to “6.1.4 Buffer Mode input”.

(8) Speed control operation command input variables

: These are input variables to execute specified velocity operation (MC_MoveVelocity) motion function block.

- Command axis: It sets the axis in which motion function block is executed.
- Changes in parameters during operation: It sets whether to apply to the operation by changing input variable values of the motion function block. For more information, refer to “6.1.5 Changes in parameters during execution of motion function block”.
- Operating velocity: It sets velocity in speed control operation.
- Operating acceleration, operating deceleration, operating jerk: It sets values to be applied in speed control operation respectively.
- Operating direction: It sets directions in speed control operation. In case of 1, it operates in forward direction, in case of 2, in reverse direction, and in case of 3, it operates in direction of the current operation.
- Buffer mode: It sets the point of time when motion function block is executed. That is, it sets whether to execute immediately or execute after the completion of commands which are currently being performed. For more details, refer to “6.1.4 Buffer Mode input”.

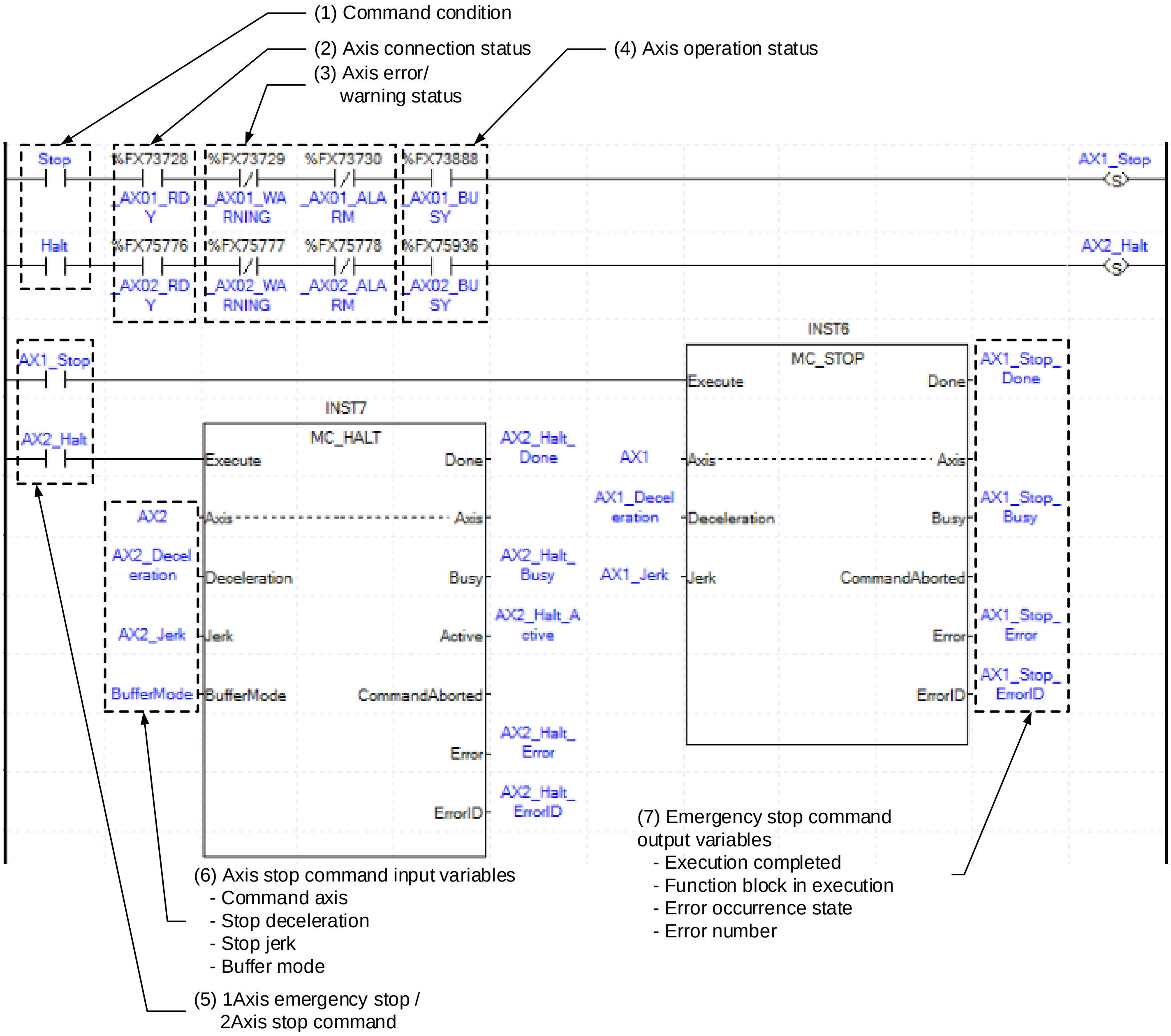
(9) Speed control operation command output variable

: It is a variable to store output values generated when specified velocity operation (MC_MoveVelocity) motion function block is executed.

- Reaching the set speed completed: When the set speed is reached through speed control operation, it is On.
- Function Block in execution: When motion function block is executed, it is On, and operation completion is On, it becomes Off.
- Function Block axis control in operation: In case motion function block is controlling the axis, it is On.
- Error occurrence state: In case error occurs while the motion function block is being executed, it is On.
- Error number: In case error occurs, the number that corresponds to error is generated.
- For more details on the output of function block, refer to “Edge operation motion function block” of “6.1.3 Basic input and output variables”.

7.3.5 Axis stop

It is an example program to stop the axis in operation. The motion function block to stop the axis in operation includes “Immediate Stop (MC_Stop)” and “Halt (MC_Halt)”. As a command to stop the axis, “Halt (MC_Halt)” performs “Halt (MC_Halt)”, the stop status is aborted by other motion function blocks during the stop, and other motion function blocks can be executed. For more details, refer to “Chapter 6 Command”.



(1) Command condition

: It is a condition to give emergency stop/axis stop commands to the axis.

(2) Axis connection status flag

: When the axis to be operated is connected to motion control module, and EtherCAT communication with motion control module is normally performed, it is On.

(3) Axis error/warning status flag
: If there are errors and warning in the axis, it is On.

(4) Axis operation status flag
:If the axis is in operation, it is On.

(5) 1-axis emergency stop / 2-axis axis stop commands
: In example programs, immediate stop (MC_Stop) motion function block is executed in 1-axis, and halt (MC_Halt) motion function block is executed in 2-axis under the following conditions.
- The axis stop condition is On.
- The axis is normally connected
- There should be no errors and warnings.
- In operation

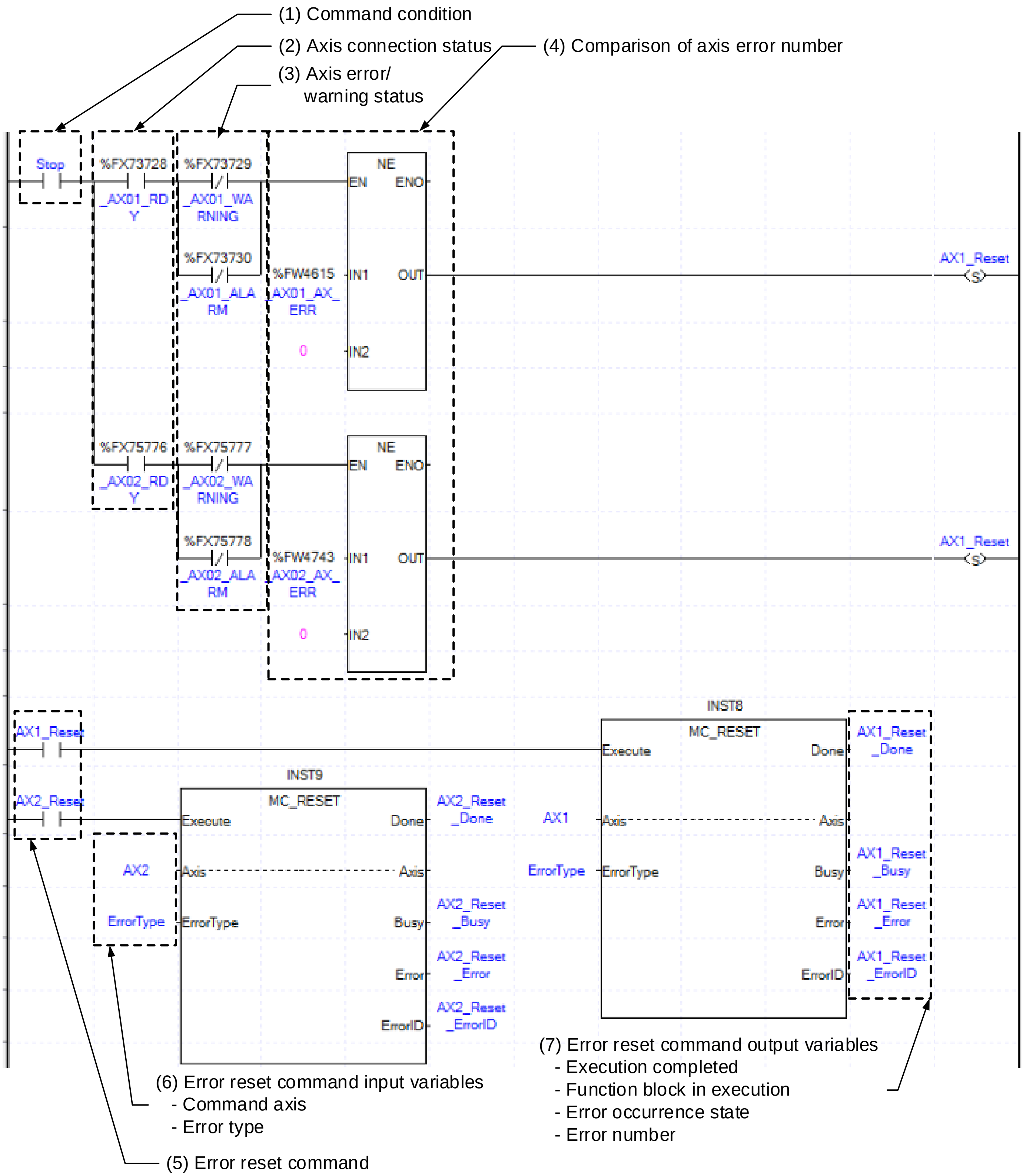
Conditions to execute motion function block may vary depending on systems.

(6) Axis stop command input variables
: These are input variables to execute Halt (MC_Halt) motion function block.
- Command axis: It sets the axis in which motion function block is executed.
- Stop deceleration: Its sets deceleration from operating speed at the time of axis stop to a stop.
- Stop jerk: it sets the jerk at the stop time.
- Buffer mode: It sets the point of time when motion function block is executed. That is, it sets whether to execute immediately or execute after the completion of commands which are currently being performed. For more details, refer to “6.1.4 Buffer Mode input”.

(7) Emergency stop command output variables
: It is a variable to store output values generated when Immediate Stop (MC_Stop) motion function block is executed.
- Execution completed: In case the axis stop, it is On.
- Function Block in execution: When motion function block is executed, it is On, and the execution is completed, It becomes Off.
- Error occurrence state: In case error occurs while the motion function block is being executed, it is On.
- Error number: In case error occurs, the number that corresponds to error is generated.
- For more details on the output of motion function block, refer to “Edge operation motion function block” of “6.1.3 basic I/O variable”.

7.3.6 Error handling

It is an example program to check the errors that occurred on the axis and conduct error reset.



(1) Command condition
: It is a condition to give error reset commands to the axis.

(2) Axis connection status flag
: When the axis to be operated is connected to motion control module, and EtherCAT communication with motion control module is normally performed, it is On.

(3) Axis error/warning status flag
: If there are errors and warnings in the axis, it is On.

(4) Comparison of axis error number
: In example programs, a case where the value of error number flag on the axis is not 0 is determined to be error reset condition through a comparison.

(5) Error reset command
: In example programs, axis error reset (MC_Reset) motion function block is executed under the following conditions.
- The axis operation condition is On.
- The axis is normally connected
- There should be error and warnings.
- Error number is not 0.

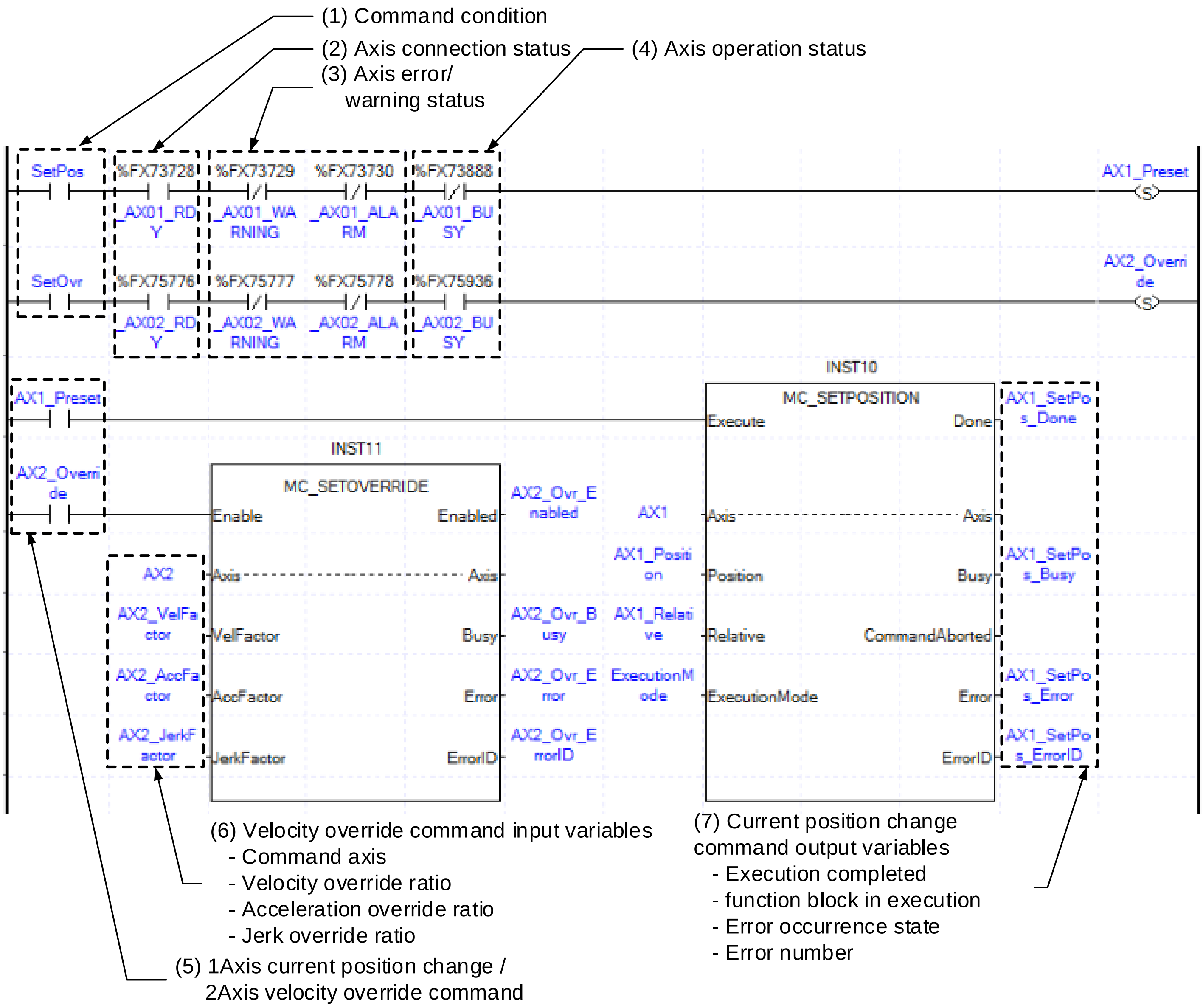
Conditions to execute motion function block may vary depending on systems.

(6) Error reset command input variables
: These are input variables to execute axis error reset (MC_Reset) motion function block.
- Command axis: It sets the axis in which motion function block is executed.
- Error type: The type of error for error rest is set. 0 represents axis error, and 1 common error.

(7) Error reset command output variables
: It is a variable to store output values generated when axis error reset (MC_Reset) motion function block is executed.
- Execution completed: When the execution of motion function block is completed, it is On.
- Function Block in execution: If motion function block is being performed, it is On, and execution is completed, it becomes Off.
- Error occurrence state: In case error occurs while motion function block is being executed, it is On.
- Error number: In case error occurs, the number that corresponds to error is generated.
- For more details on the output of motion function block, refer to “Edge operation motion function block” of “6.1.3 basic I/O variable”.

7.3.7 Operation Change

It is an example program to change the current location of the axis and speed in operation.



(1) Command condition

: It is a condition to give current location change/operating speed change commands to the axis.

(2) Axis connection status flag

: When the axis to be operated is connected to motion control module, and EtherCAT communication with motion control module is normally performed, it is On.

(3) Axis error/warning status flag

: If there are errors and warnings in the axis, it is On.

(4) Axis operation status flag

: If the axis is in operation, it is On.

(5) 1-axis current location change/2-axis speed override command

: In the example program, the current location setting (MC_SetPosition) motion function block is executed under the following conditions.

- The current location change condition is On.
- The axis is normally connected
- There should be no errors and warnings.
- The axis is not in operation.

In addition, speed/acceleration override (MC_SetOverride) motion function block is executed under the following conditions.

- The operating speed change condition is On.
- The axis is normally connected
- There should be no errors and warnings.
- The axis is in operation.

Conditions to execute motion function block may vary depending on systems.

(6) Speed override command input variables

: These are input variables to execute speed/acceleration override (MC_SetOverride) motion function block.

- Command axis: It sets the axis in which motion function block is executed.
- Speed override ratio: It sets the ratio of the speed to change in comparison with operating speed that is currently set.
- Acceleration override ratio: It sets the ratio of the acceleration to change in comparison with acceleration value which is currently set.
- Jerk override ratio: It sets the ratio of the jerk to change in comparison with jerk value that is currently set. That is, if 2 is set to the value of the ratio, double the currently set value is set. .

(7) Current location change command output variables

: These are variables to store output values generated when the current location setting (MC_SetPosition) motion function block is executed.

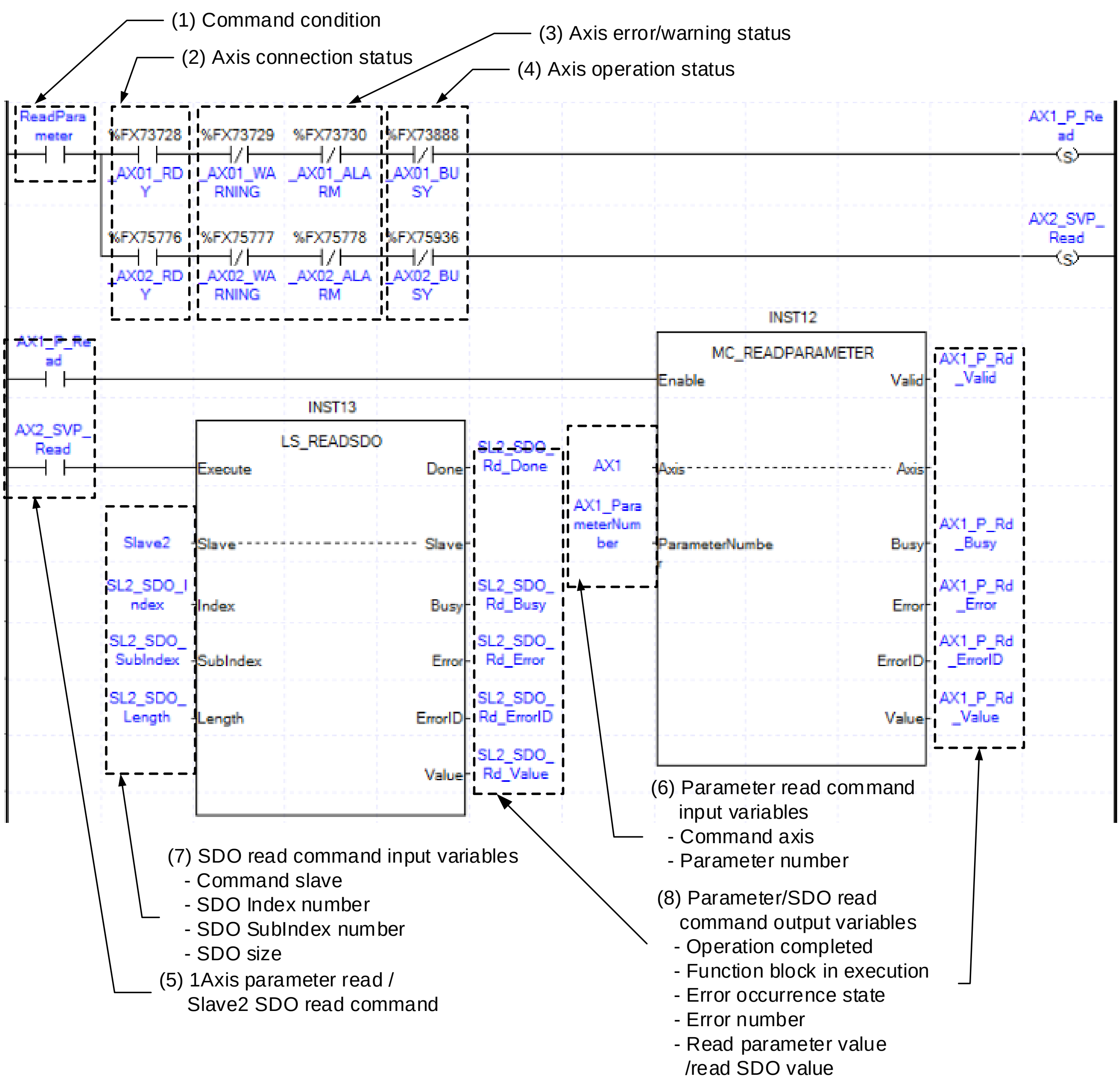
- Execution completed: When the execution of motion function block is completed, it is On.
- Function Block in execution: When motion function block is executed, it is On, and the execution is completed, It becomes Off.
- Error occurrence state: In case error occurs while motion function block is being executed, it is On.
- Error number: In case error occurs, the number that corresponds to error is generated.
- For more details on the output of motion function block, refer to “Edge operation motion function block” of “6.1.3 basic I/O variable”.

7.3.8 Read/ Write Parameter

Parameter read/write commands include “Parameter Write (MC_WriteParameter)” and Parameter Read (MC_ReadParameter)” as well as “SDO Write (LS_WriteSDO)” and “SDO Read (LS_ReadSDO)”.

“Parameter Write (MC_WriteParameter)” and “Parameter Read (MC_ReadParameter)” are commands to write and read operation parameters of the axis or encoder parameter, and “SDO Write (LS_WriteSDO)” and “SDO Read (LS_ReadSDO)” are commands to read or write SDO data of the connected EtherCAT slaves. If the slave of the connected EtherCAT is a servo drive, the SDO data becomes servo parameters. The following is an example of the program to read or change the operating parameters and servo parameters using Read / Write commands servo drive and the parameter when slave 2 is a servo drive and it is connected to 2 axes.

■ Parameter Read



(1) Command Condition
: It is a condition to read parameters and serve parameters of the axes.
(2) Axis connection status flag
: When the axis to be operated is connected to motion control module, and EtherCAT communication with motion control module is normally performed, it is On.
(3) Axis error/warning status flag
: If there are errors and warnings in the axis, it is On.
(4) Axis operation status flag
: If the axis is in operation, it is On.
(5) 1-axis parameter write/ 2-axis servo parameter read commands
: In example programs, Parameter Read (MC_ReadParameter) motion function block is executed in 1-axis, and Servo Parameter Read (LS_ReadSDO) motion function block is executed in 2-axis under the following conditions.
- Parameter read condition is On.
- The axis is normally connected
- There should be no errors and warnings.
- Not in operation.

Conditions to execute motion function block may vary depending on systems.
(6) Parameter read command input variables
: These input variables to execute Parameter Read (MC_ReadParameter) motion function block.
- Command axis: It sets the axis in which motion function block is executed.
- Parameter number: It sets the parameter numbers to read with motion function block.
  Numbers by parameter are as follows.

No.	Parameter	Item	Note
0	Standard setting	Unit	0: pulse, 1: mm, 2: inch, 3:degree
1		1 Pulse per revolution	1 ~ 4,294,967,295[pulse]
2		1 Travel per rotation	0.000000001 ~ 4,294,967,295 [Unit]
3		Speed command unit	0: Unit/sec, 1: Unit/min, 2: rpm
4		Speed limit value	LREAL Positive number [Unit/s, rpm] Change according to Unit, Pulses per rotation, Travel per rotation, Speed command unit
5		Emergency stop deceleration	0 or LREAL Positive number [Unit/s^2]
6		Encoder select	0:Incremental Encoder,1:Absolute Encoder
7		Gear ratio(Motor)	1 ~ 65,535
8		Gear ratio(Machine)	1 ~ 65,535
9		Operation mode of the reverse rotation	0:disabled,1:enabled
46		Position Control Range Expansion	0: Disable, 1: Able
10	Expansion setting	S/W upper limit	LREAL[*Note 1)] [units]
11		S/W low limit	LREAL[*Note 1)] [units]
12		Infinite running repeat position	LREAL Positive number [Unit]
13		Infinite running repeat	0:disabled,1:enabled
14		Command Inposition range	0 or Long real (LREAL) positive number[*1]) [Unit]
15		Exceeding value of tracking error	0 or Long real (LREAL) positive number[*1]) [Unit]
16		Current position compensation amount	0 or Long real (LREAL) positive number[*1]) [Unit]
17		Current speed filter time constant	0 ~ 100
18		Error reset monitoring time	1 ~ 1000 [ms]
19		S/W limit during speed control	0: Not detect , 1 : detect
20		Tracking error level	0: Warning, 1: Alarm

21		Jog high speed	LREAL Positive number (Jog low speed ~speed limit) [Unit/s]
22		Jog low speed	LREAL Positive number (< Jog high speed) [Unit/s]
23		JOG Acceleration	0 or LREAL Positive number [Unit/s^2]
24		JOG Deceleration	0 or LREAL Positive number [Unit/s^2]
25		JOG Jerk	0 or LREAL Positive number [Unit/s^3]
26		Override mode	0: Specified by ratio, 1: Specified by unit
29		Backlash compensation amount	0 or Long real (LREAL) positive number[*1]) [Unit]

No.	Parameter	Item	Note
27	NC setting	Identifying range to reach the spindle rotation command speed	0~100%
28		Identifying RPM to reach the spindle rotation zero speed	0~ 100rpm
30	NC Spindle Axis Setting	Select the Spindle Encoder	0: Not Use, 1: Motor ENC, 2: Built-in ENC1, 3: Built-in ENC2, 4: EtherCAT ENC
31		Number of pulses per rotation of the spindle EtherCAT encoder	1 ~ 4294967295
32		Spindle EtherCAT encoder position variable	0: I device, 1: M device
33		Spindle EtherCAT encoder position address	0~4095 (Spindle EtherCAT encoder position variable = 0: I) 0~524287 (Spindle EtherCAT encoder position variable = 1: M)
34		The P Gain of the Spindle Positioning Mode	1~500 Hz
35		The Feed Forward Gain of the Spindle Positioning Mode	0~100 %
36	NC spindle home setting	Home operation method	0: Servo drive supported, 33:Reverse direction, Z phase, 34: Forward direction, Z phase, 35: Set the homing of the current position
37		Switch navigation speed of the homing operation	LREAL Positive number Zero navigation speed of the origin operation ≤ Switch navigation speed of the origin ≤ Limit value of speed
38		Zero navigation speed of the homing operation	-
39		Acceleration/deceleration of the homing operation	0 or LREAL Positive number [Unit/s^2]
40		Z phase variable	0: I device, 1: M device
41		Z phase address	0~131071 (Z phase variable = 0: I) 0~16777215 (Z phase variable = 0: M)
42		Orientation velocity	Long real (LREAL) positive number[*note1]) (≤ Limit value of speed)
43		Orientation direction	0:forward direction , 1:reverse direction
44		Orientation offset	0~360
100	Built-in encoder setting	Encoder 1 unit	0: pulse, 1: mm, 2: inch, 3:degree
101		Encoder 1 1 pulse per rotation	1 ~ 4,294,967,295[pulse]
102		Encoder 1 1 travel per rotation	0.000000001 ~ 4,294,967,295 [Unit]
103		Encoder1 pulse input	0:CW/CCW 1 multiplier, 1:PULSE/DIR 1 multiplier

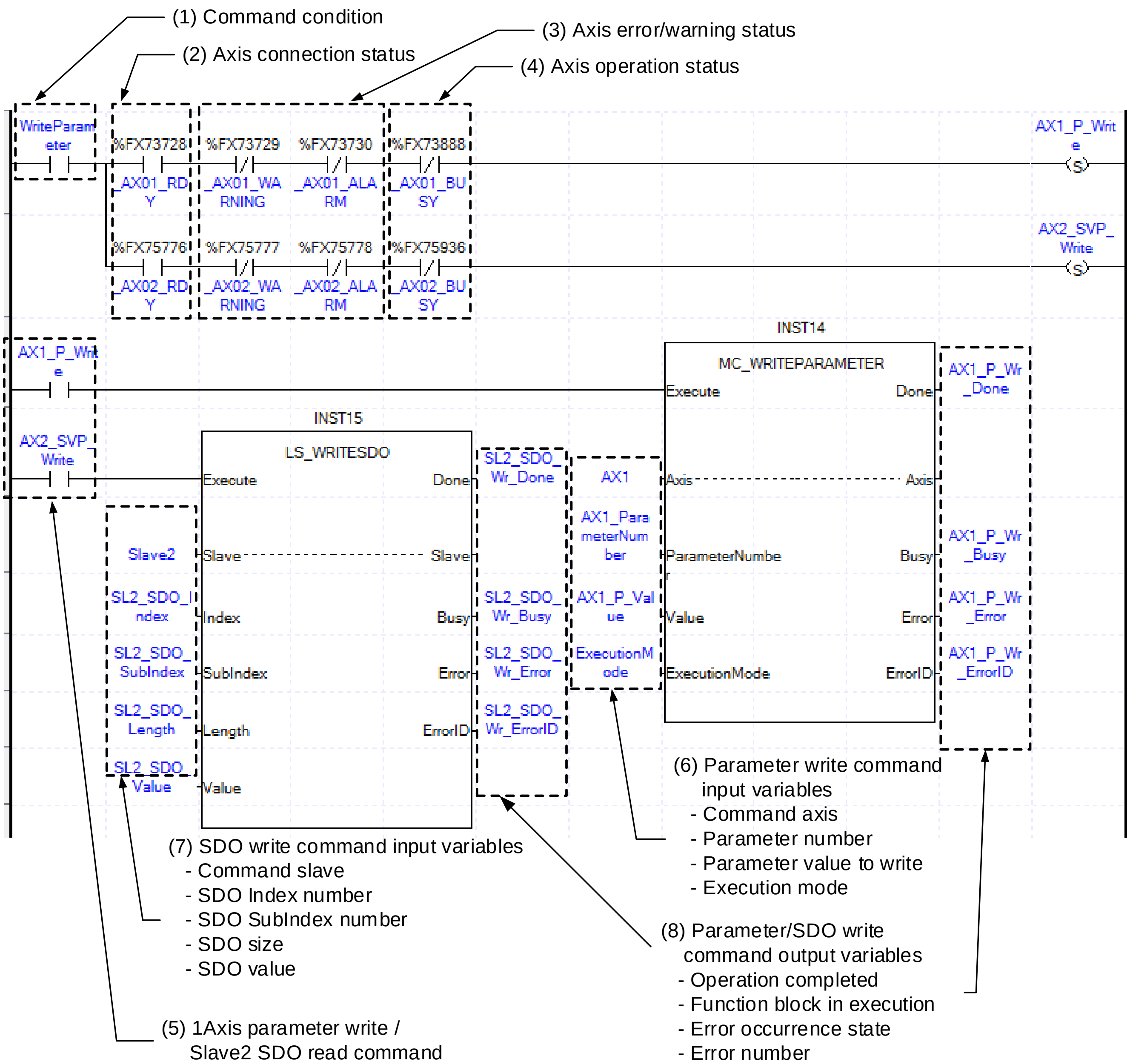
No.	Parameter	Item	Note
			2: Pulse/Dir 2 multiplier, 3: Phase A/B 1 multiplier 4: Phase A/ B 2 multiplier, 5: Phase A/B 4 multiplier
104		Encorder1 max. value	(Encoder1 min. value + 1) ~ 2147483647
105		Encoder1 Min. value	-2147483648~(Encoder1 Max. value-1)
106		Encoder1 input filter value	0: Disable, 1:500kPPS 2: 200kPPS, 3: 100kPPS 4: 10kPPS, 5: 1kPPS 6: 0.1kPPS
107		Encoder1 Speed unit	0: Unit/sec, 1: Unit/min, 2: rpm
108		Encoder1 position filter time constant	0 ~ 1000 ms
200		Encoder 2 unit	0: pulse, 1: mm, 2: inch, 3:degree
201		Encoder 2 1 pulse per rotation	1 ~ 4,294,967,295[pulse]
202		Encoder 2 1 travel per rotation	0.000000001 ~ 4,294,967,295 [Unit]
203		Encorder2 pulse input	0:CW/CCW 1 multiplier, 1:PULSE/DIR 1 multiplier 2: Pulse/Dir 2 multiplier, 3: Phase A/B 1 multiplier 4: Phase A/ B 2 multiplier, 5: Phase A/B 4 multiplier
204		Encorder2 max. value	(Encoder1 min. value + 2) ~ 2147483647
205		Encoder2 Min. value	-2147483648~(Encoder2 Max. value-1)
206		Encoder2 input filter value	0: Disable, 1:500kPPS 2: 200kPPS, 3: 100kPPS 4: 10kPPS, 5: 1kPPS 6: 0.1kPPS
207		Encoder2 Speed unit	0: Unit/sec, 1: Unit/min, 2: rpm
208		Encoder2 position filter time constant	0 ~ 1000 ms

* Note1) LREAL range: 2.2250738585072e-308 ~ 1.79769313486232e+308
  Long real number (LREAL) positive range: 0 < x ≤1.79769313486232e+308

(7) SDO data read command input variable
: These are input variables to execute Servo Parameter Read (LS_ReadSDO) motion function block.
- Command axis: It sets the axis in which motion function block is executed.
- Servo parameter index number, SubIndex number, size
  : Each value is set in servo parameters to read. Refer to the instruction manual of the servo drive for index number, subindex number and size of servo parameters.

(8) Parameter read/Servo parameter read command output variables
: These are variables to store output values generated when Parameter Read (MC_ReadParameter) and Servo Parameter Read (LS_ReadSDO) motion function block is executed.
- Operation completed: If values of parameters and servo parameters is read, it is On.
- Function Block in execution: When motion function block is executed, it is On, and operation completion is On, it becomes Off.
- Error occurrence state: In case error occurs while the motion function block is being executed, it is On.
- Error number: In case error occurs, the number that corresponds to error is generated.
- Read parameter values/Read servo parameter values: Values of parameters and servo parameters read by the execution of motion function block is stored.

■ Parameter write



(1) Command condition

: It is a condition to write parameters and servo parameters of the axes.

(2) Axis connection status flag

: When the axis to be operated is connected to motion control module, and EtherCAT communication with motion control module is normally performed, it is On.

(3) Axis error/warning status flag

: If there are errors and warnings in the axis, it is On.

(4) Axis operation status flag

: If the axis is in operation, it is On.

(5) 1-axis parameter write/ 2-slave SDO write commands

: In example programs, Parameter write (MC_WriteParameter) motion function block is executed in 1-axis, and SDO write (LS_WriteSDO) motion function block is executed in 2-slave under the following conditions.

- Parameter write condition is On
- The axis is normally connected
- There should be no errors and warnings.
- Not in operation.

Conditions to execute function block may vary depending on systems.

(6) SDO write command input variables

: These are input variables to execute Parameter Write (MC_WriteParameter) motion function block.

- Command axis: It sets the axis in which motion function block is executed.
- Parameter number: It set parameter numbers to write with the motion function block.
- Parameter values to write: Values to write in the parameters are set.
- Execution mode: It specifies the point of time when parameters are written. If it sets 0, it changes parameter values upon executing motion function block. If it sets 1, it is changed to the same point of time with “Buffered” of BufferMode. (Refer to 6.1.4 BufferMode)

(7) SDO write command input variables

: These are input variables to execute SDO write (LS_WriteSDO) motion function block.

- Command axis: It sets the axis in which motion function block is executed.
- Servo parameter index number, subIndex number, size

  : Each value is set according to servo parameters to write. Refer to instruction manual of the servo drive for index number, subindex number and size of servo parameters.
- Values of servo parameters to write: Values to be written in the servo parameters is set.

(8) Parameter write/Servo parameter write command output variable

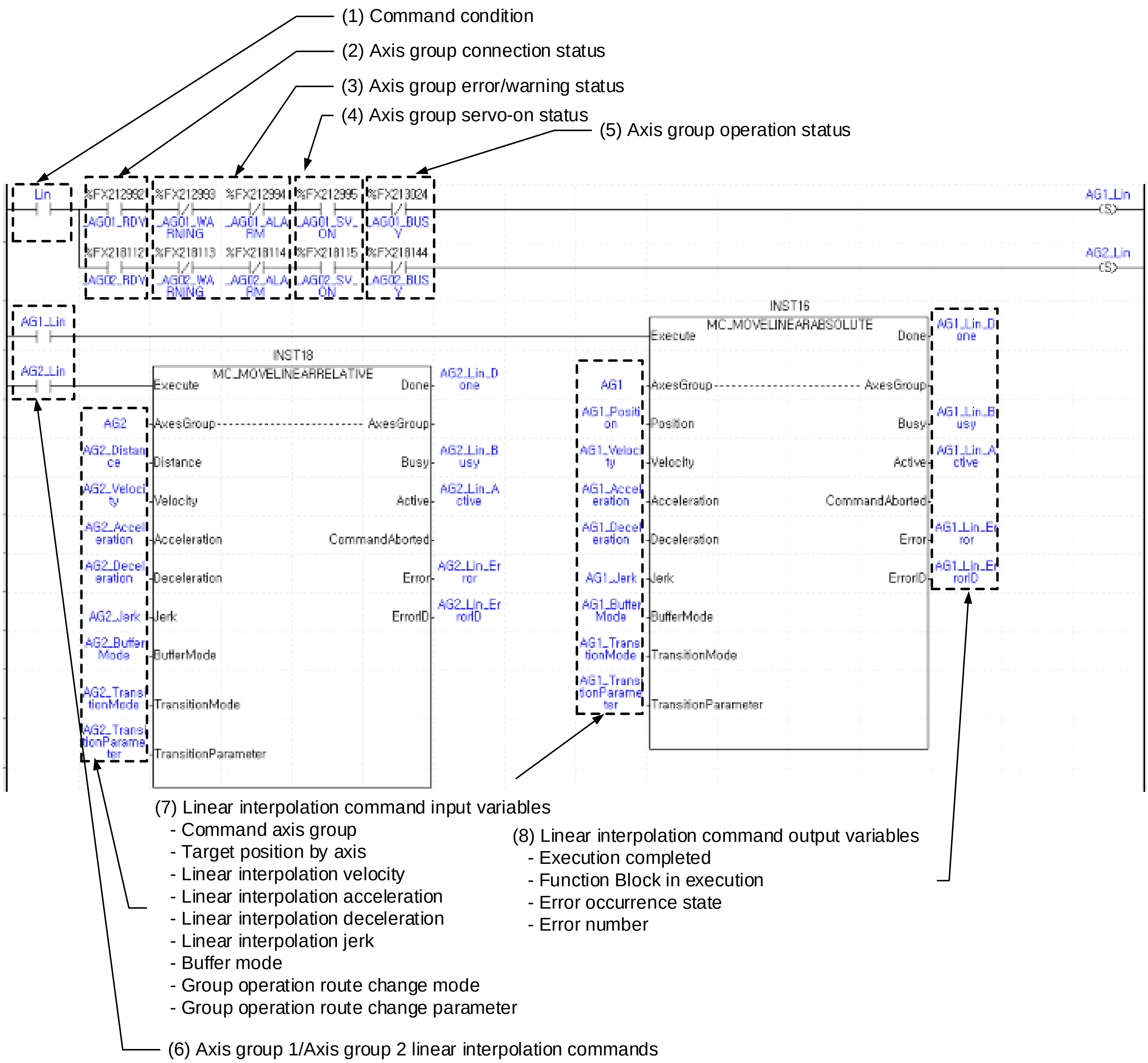
: It is a variable to store output values generated when Parameter write (MC_WriteParameter) and SDO write (LS_WriteSDO) motion function block is executed.

- Operation completed: If values of the parameters and servo parameters are written, it is On.
- Function Block in execution: When motion function block is executed, it is On, and operation completion is On, it becomes Off.
- Error occurrence state: In case error occurs while motion function block is being executed, it is On. As for error number, the number that corresponds to error is generated in case error occurs.

7.4 Multi-Axis Operation Program

7.4.1 Linear Interpolation operation

It is an example program to operate linear interpolation with axes set to the same group. In the example program, 1-axis and 2-axis are assumed to be included in the same axis group. Refer to the example program of "7.4.5 Axis group processing" to include an axis in axis group or remove the axis from axis group.



(1) Command condition
: It is a condition to give linear interpolation command to the axis group.

(2) Axis group connection state flag
: In case axes of the axis group to be operated are connected to motion control module, and EtherCAT communication with motion control module is normally performed, it is On.

(3) Axis group error/Warning status flag
: If there are errors and warnings in axes included in the axis group, it is On.

(4) Axis group servo-on status
: If axes included in the axis group are in servo-on state, it is On.

(5) Axis group operation status flag
: If axes of the axis group are in operation, it is On.

(6) Axis group 1 absolute position linear interpolation/Axis group 2 relative position linear interpolation commands
: In example programs, absolute position linear interpolation operation (MC_MoveLinearAbsolute) is executed in axis group 1, and relative position linear interpolation operation (MC_MoveLinearRelative) motion function block in axis 2 under the following conditions.
- Linear interpolation operation condition is On.
- Axes included in the axis group are normally connected.
- There should be no errors and warnings.
- Axes of the axis group are not in operation.

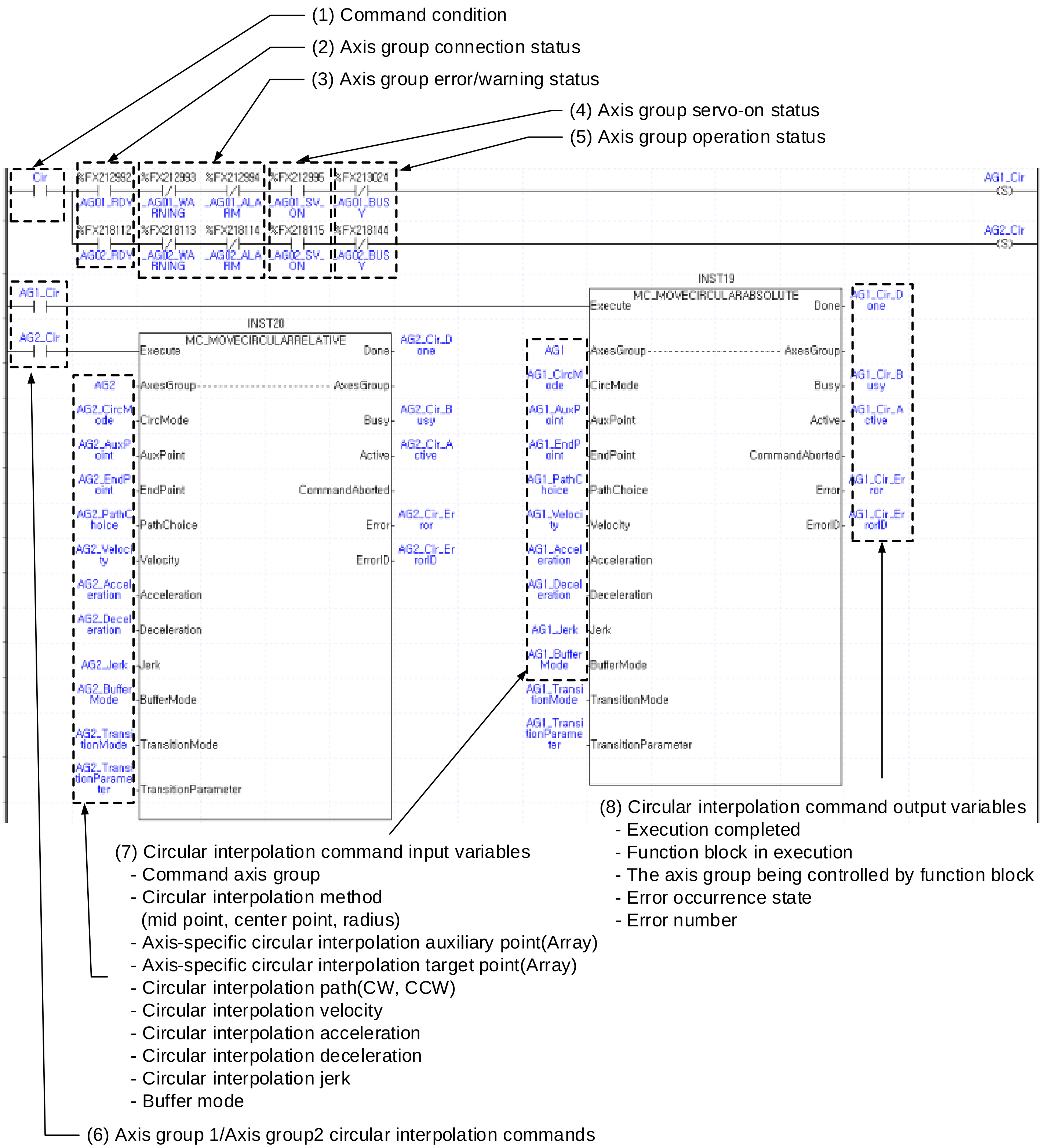
Conditions to execute motion function block may vary depending on systems.

(7) Linear interpolation command input variables
: These are input variables to execute absolute position linear interpolation operation (MC_MoveLinearAbsolute) and relative position linear interpolation operation (MC_MoveLinearRelative) motion function block.
- Command axis group: It sets axis group in which motion function block is executed.
- Target position by axis: Array variables are set, and linear interpolation operation target position of axes included in axis group is set in order.
- Linear interpolation speed: It sets target speed to execute linear interpolation, when the speed refers to the interpolation speed.
- Linear interpolation acceleration, deceleration, jerk: they set values to be applied when performing linear interpolation.
- Buffer mode: It sets the point of time when motion function block is executed. That is, it sets whether to execute immediately or execute after the completion of commands which are currently being performed. For more details, refer to “6.1.4 Buffer Mode input”.
- Group operation route change mode and group operation route change parameter
  : It specifies in which way the axis group in operation is connected to the trace the existing commands describe when linear interpolation command is given. Refer to “6.1.6 Group operation route change settings”.

(8) Linear interpolation command output variable
: It is a variable to store output values generated when absolute position linear interpolation operation (MC_MoveLinearAbsolute) and relative position linear interpolation operation (MC_MoveLinearRelative) motion function block is executed.
- Execution completed: When the execution of function block is completed, it is On.
- Function Block in execution: When motion function block is executed, it is On, and the execution is completed, It becomes Off.
- Error occurrence state: In case error occurs while the motion function block is being executed, it is On.
- Error number: In case error occurs, the number that corresponds to error is generated.
- For more details on the output of motion function block, refer to “Edge operation motion function block” of “6.1.3 basic I/O variable”.

7.4.2 Circular Interpolation operation

It is an example program to operate circular interpolation operation with axes set to the same group. In the example program, 1-axis and 2-axis are assumed to be included in the same axis group. Refer to the example program of “7.4.5 Axis group processing” to include an axis in axis group or remove the axis from axis group.



(1) Command condition

: It is a condition to give circular interpolation command to the axis group.

(2) Axis group connection state flag

: In case axes of the axis group to be operated are connected to motion control module, and EtherCAT communication with motion control module is normally performed, it is On.

(3) Axis group error/Warning status flag

: If there are errors and warnings in axes included in the axis group, it is On.

(4) Axis group servo-on status

: If axes included in the axis group are in servo-on state, it is On.

(5) Axis group operation status flag

: If axes of the axis group are in operation, it is On.

(6) Axis group 1 absolute position circular interpolation/Axis group 2 relative position circular interpolation commands

: In example programs, absolute position circular interpolation operation (MC_MoveCircularAbsolute) is executed in axis group 1, and relative position circular interpolation operation (MC_MoveCircularRelative) motion function block in axis 2 under the following conditions.

- Circular interpolation operation condition is On.
- Axes included in the axis group are normally connected.
- There should be no errors and warnings.
- Axes of the axis group are not in operation.

Conditions to execute motion function block may vary depending on systems.

(7) Circular interpolation command input variables

: These are input variables to execute absolute position circular interpolation operation (MC_MoveCircularAbsolute) and relative position circular interpolation operation (MC_MoveCircularRelative) motion function block.

- Command axis group: It sets axis group in which motion function block is to be executed.
- Target position by axis: Array variables are set, and linear interpolation operation target position of axes included in axis group is set in order.
- Circular interpolation method: It sets a method to execute circular interpolation through selection among mid-point method, center point method and radius method.
- Axis-specific circular interpolation auxiliary point: It takes a form of array and sets auxiliary point required for circular interpolation in the order of axes included in axis group.
- Axis-specific circular interpolation target point: It takes a form of array and sets target position in the order of axes included in axis group.
- Circular interpolation velocity: It sets target speed to execute circular interpolation, when the speed refers to interpolation speed.
- Circular interpolation acceleration, deceleration, jerk: Values to be applied when circular interpolation is performed are set.
- Buffer mode: It sets the point of time when motion function block is executed. That is, it sets whether to execute immediately or execute after the completion of commands which are currently being performed. For more details, refer to “6.1.4 Buffer Mode input”.

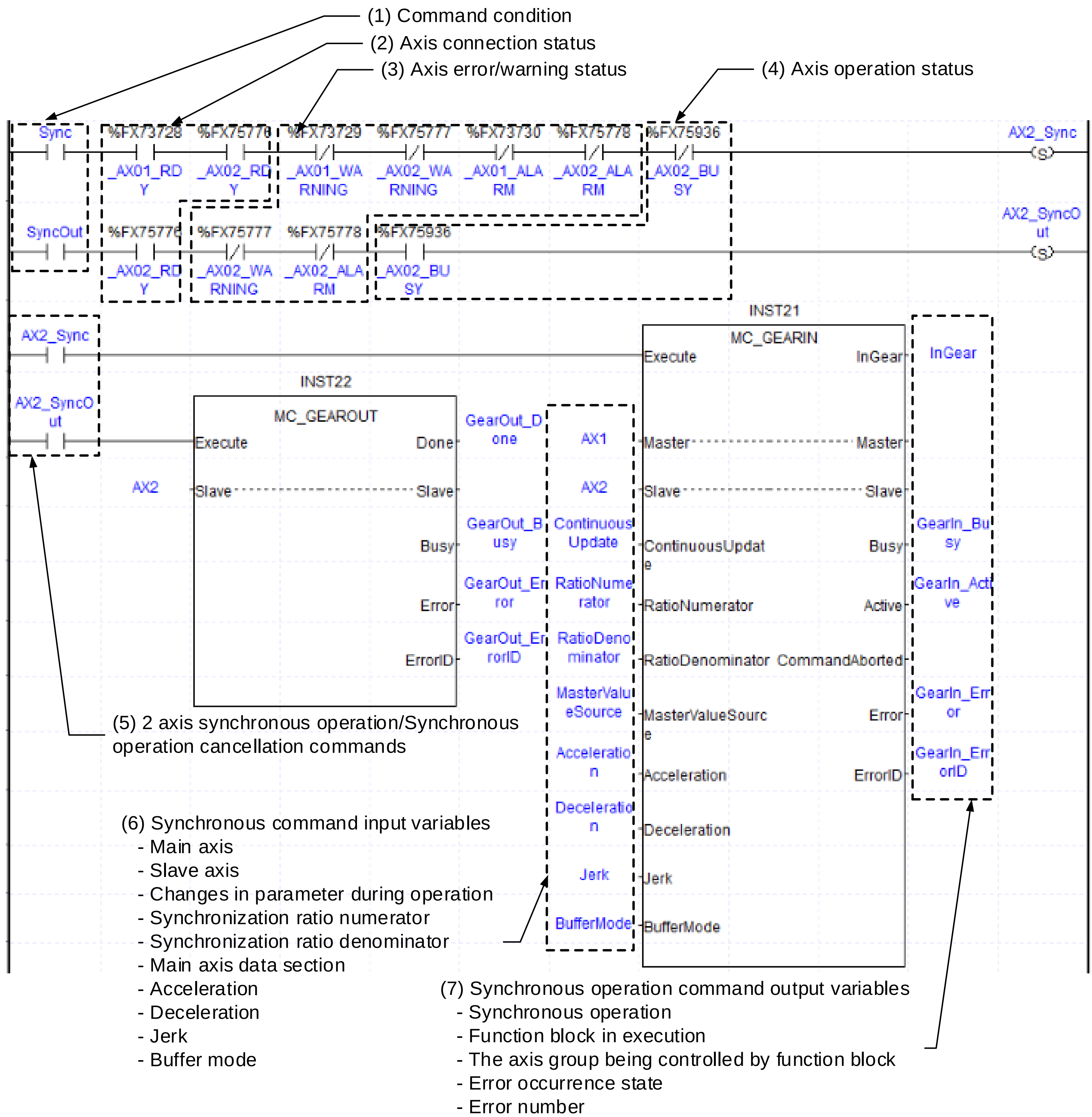
(8) Circular interpolation command output variable

: It is a variable to store output values generated when absolute position circular interpolation operation (MC_MoveCircularAbsolute) and relative position circular interpolation operation (MC_MoveCircularRelative) motion function block is executed.

- Execution completed: When the execution of motion function block is completed, it is On.
- Function Block in execution: When motion function block is executed, it is On, and the execution is completed, It becomes Off.
- The axis group being controlled by function block: When motion function block controls the axis group, it is On.
- Error occurrence state: In case error occurs while the motion function block is being executed, it is On.
- Error number: In case error occurs, the number that corresponds to error is generated.
- For details on the output of motion function block, refer to “Edge motion commands” of “6.1.3 Basic I/O Variable”.

7.4.3 Synchronous operation

It is an example program on the synchronous operation in which serve axis moves in synchronization ratio set in the main axis.



(1) Command condition
: It is a condition to give synchronous operation/synchronous operation cancellation commands to the axis.
(2) Axis connection status flag
: When the axis to be operated is connected to motion control module, and EtherCAT communication with motion control module is normally performed, it is On.
(3) Axis error/warning status flag
: If there are errors and warnings in the axis, it is On.
(4) Axis operation status flag
: If the axis is in operation, it is On.
(5) 2Axis synchronous operation/Synchronous operation cancellation commands
: In the example program, electronic gear operation (MC_GearIn) motion function block is executed under the following conditions.
- Synchronous operation condition is On.
- The axis and main axis is normally connected.
- There should be no errors and warnings.
- The axis is not in operation.

In addition, electronic gear cancellation (MC_GearOut) motion function block is executed under the following conditions.
- Synchronous operation cancellation condition is On.
- The axis is normally connected
- There should be no errors and warnings.
- The axis is in operation.

Conditions to execute motion function block may vary depending on systems.
(6) Synchronous command input variables
: These are input variables to execute electronic gear operation (MC_GearIn) motion function block.
- Main axis: It sets serve axis of synchronous operation.
- Serve axis: It sets the axis in which synchronous operation is to be performed.
- Synchronization ratio numerator: It sets numerator value among synchronization ratio to be operated by synchronization of the operation of main axis.
- Synchronization ratio denominator: It sets denominator among synchronization ratio to be operated by synchronization of the operation of main axis.
- The speed of serve axis in a state of gear operation (InGear) is set as follows.

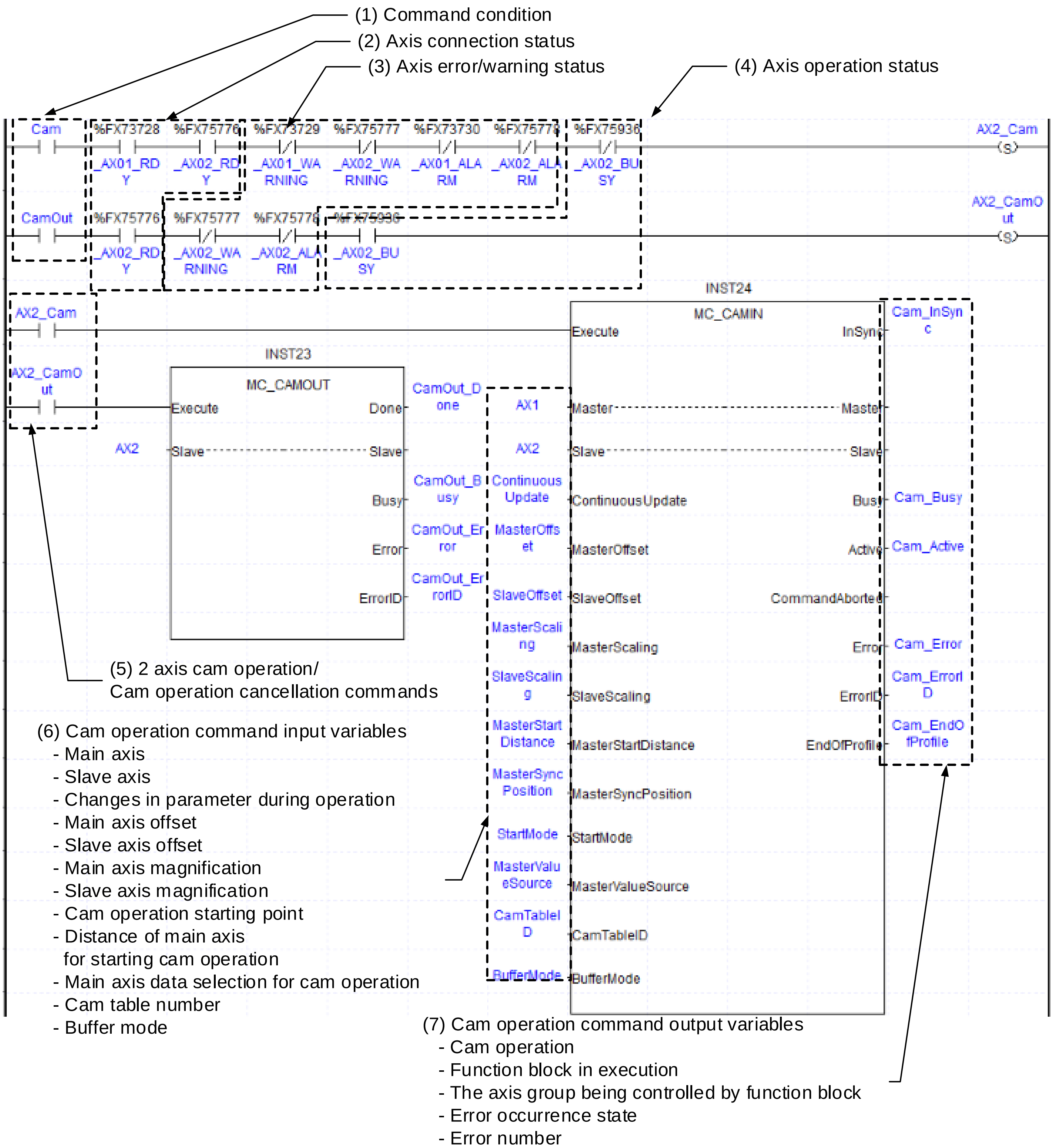
Serve axis speed = Main axis speed x (Synchronization ratio numerator/Synchronization denominator)

- Main axis data selection: It selects whether the data of main axis is set to command speed or current speed.
In case command speed is set, synchronization is achieved based on the speed of main axis calculated in motion control module. In case current speed is set, synchronization is achieved by using speed data of main axis servo drive transmitted through the communication.
- Acceleration, deceleration, jerk: Each value is set in synchronous operation.
- Buffer mode: It sets the point of time when motion function block is executed. That is, it sets whether to execute immediately or execute after the completion of commands which are currently being performed. For more details, refer to “6.1.4 Buffer Mode input”.

(7) Synchronous operation command output variable
: It is a variable to store output values generated when electronic gear operation (MC_GearIn) motion function block is executed.
- Synchronous operation: When serve axis is normally synchronized in main axis after the execution of motion function block, it is On.
- Function Block in execution: When motion function block is executed, it is On, and the execution is completed, It becomes Off.
- Error occurrence state: In case error occurs while the motion function block is being executed, it is On.
- Error number: In case error occurs, the number that corresponds to error is generated.
- For details on the output of motion function block, refer to “Edge motion commands” of “6.1.3 Basic I/O Variable”.

7.4.4 CAM Operation

It is an example program on the cam operation that moves in synchronization based on cam (CAM) profile in which serve axis is set.



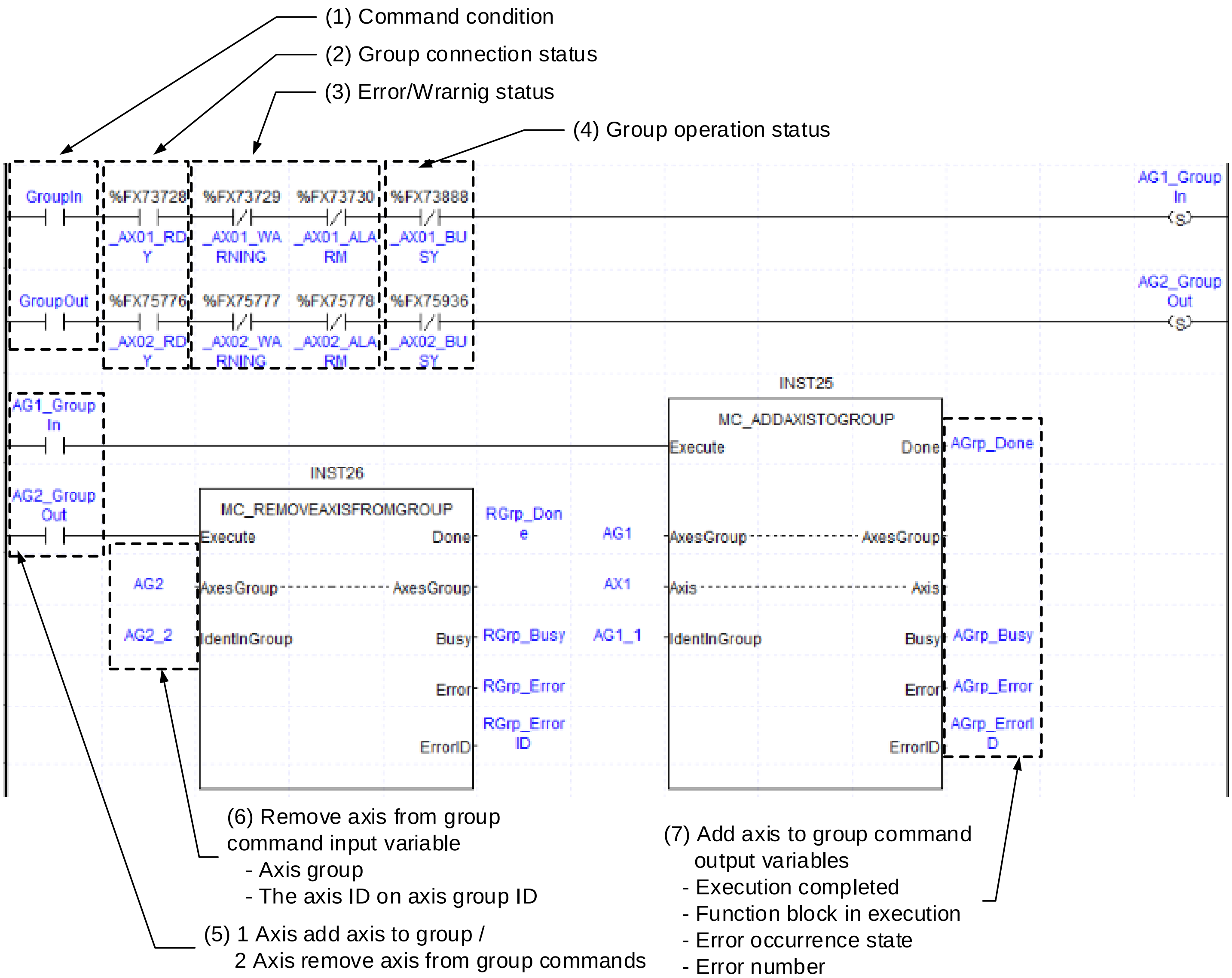
(1) Command condition
: It is a condition to give cam operation/cam operation cancellation commands to the axis.
(2) Axis connection status flag
: When the axis to be operated is connected to motion control module, and EtherCAT communication with motion control module is normally performed, it is On.
(3) Axis error/warning status flag
: If there are errors and warnings in the axis, it is On.
(4) Axis operation status flag
: If the axis is in operation, it is On.
(5) 2-axis cam operation/Cam operation cancellation commands
: In the example program, cam operation (MC_CamIn) motion function block is executed under the following conditions.
- Cam operation condition is On.
- The axis and main axis are normally connected.
- There should be no errors and warnings.
- The axis is not in operation.

In addition, cam operation cancellation (MC_CamOut) motion function block is executed under the following conditions.
- Cam operation cancellation condition is On.
- The axis is normally connected
- There should be no errors and warnings.
- The axis is in operation.

Conditions to execute motion function block may vary depending on systems.
(6) Cam operation command input variables
: These are input variables to execute cam operation (MC_CamIn) motion function block.
- Main axis: It sets main axis of cam operation.
- Serve axis: It sets the axis in which cam operation is executed.
- Main axis offset: It sets offset values of main axis data to be used when cam table data is applied.
- Serve axis offset: It sets offset values of serve axis data to be used when cam table data is applied.
- Main axis magnification: It sets magnification of main axis data to be used when cam table data is applied.
- Serve axis magnification: It sets magnification of serve axis data to be used when cam table data is applied.
- Cam operation starting point: It sets the position of main axis which will be the starting point of cam table.
- Distance of main axis for starting cam operation: It sets the distance of main axis in which actual cam operation starts.
- Main axis data selection for cam operation: It selects main axis data which will be a basis of cam operation among main axis command position and main axis current position.
- Cam table number: It sets cam data number to conduct cam operation.
- For details on cam operation command input variables, refer to “6.4.1 Cam operation (MC_CamIn)”.
- Changes in parameters during operation: It sets whether to apply to the operation by changing input variable values of the function block. For more information, refer to “6.1.5 Changes in parameters during execution of motion function block”.
- Buffer mode: It sets the point of time when motion function block is executed. That is, it sets whether to execute immediately or execute after the completion of commands which are currently being performed. For more details, refer to “6.1.4 Buffer Mode input”.

(7) Cam operation command output variable
: It is a variable to store output values generated when cam operation (MC_CamIn) motion function block is executed.
- Cam operation: It is on when serve axis is synchronized in main axis according to cam data after the execution of motion function block.
- Function Block in execution: When motion function block is executed, it is On, and the execution is completed, It becomes Off.
- Error occurrence state: In case error occurs while the motion function block is being executed, it is On.
- Error number: In case error occurs, the number that corresponds to error is generated.
- For details on the output of motion function block, refer to “Edge motion commands” of “6.1.3 Basic I/O Variable”.

7.4.5 Axis Group processing



(1) Command condition

: It is a condition to give add axis to group/remove axis from group commands to the axis.

(2) Axis connection status flag

: When the axis to be operated is connected to motion control module, and EtherCAT communication with motion control module is normally performed, it is On.

(3) Axis error/warning status flag

: If there are errors and warnings in the axis, it is On.

(4) Axis operation status flag

: If the axis is in operation, it is On.

(5) 1-axis add axis to group/2-axis remove axis from group commands

: In the example program, add axis to group (MC_AddAxisToGroup) motion function block is executed under the following conditions.

- Add axis to group condition is On.
- The axis is normally connected

- There should be no errors and warnings.
- The axis is not in operation.

In addition, group axis exclusion (MC_RemoveAxisFromGroup) motion function block is executed under the following conditions.

- Remove axis from group condition is On.
- The axis is normally connected
- There should be no errors and warnings.
- The axis is not in operation.

Conditions to execute motion function block may vary depending on systems.

(6) Remove axis from group command input variables

: These are variables to execute group axis exclusion (MC_RemoveAxisFromGroup) motion function block.

- Axis group: It sets the group to exclude the axis.
- It sets ID values granted when the axis is included in axis group.

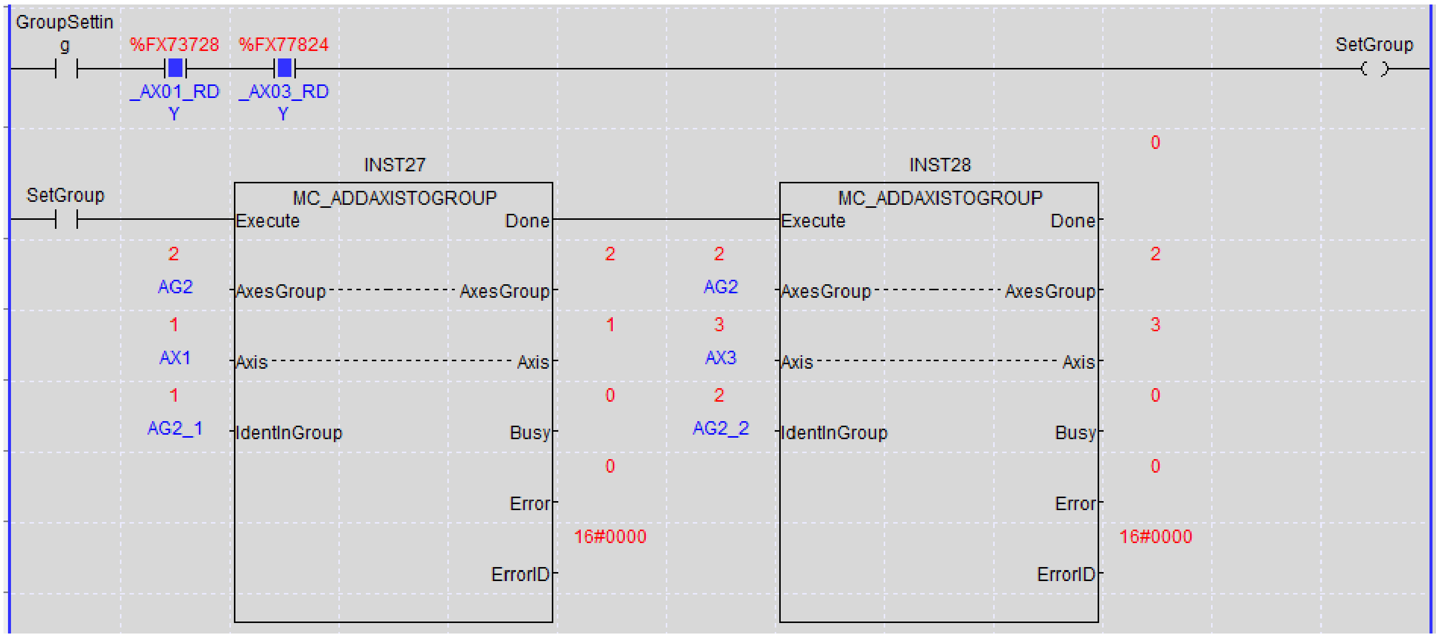
(7) Add axis to group command output variable

: It is a variable to store output values generated when add axis to group (MC_AddAxisToGroup) motion function block is executed.

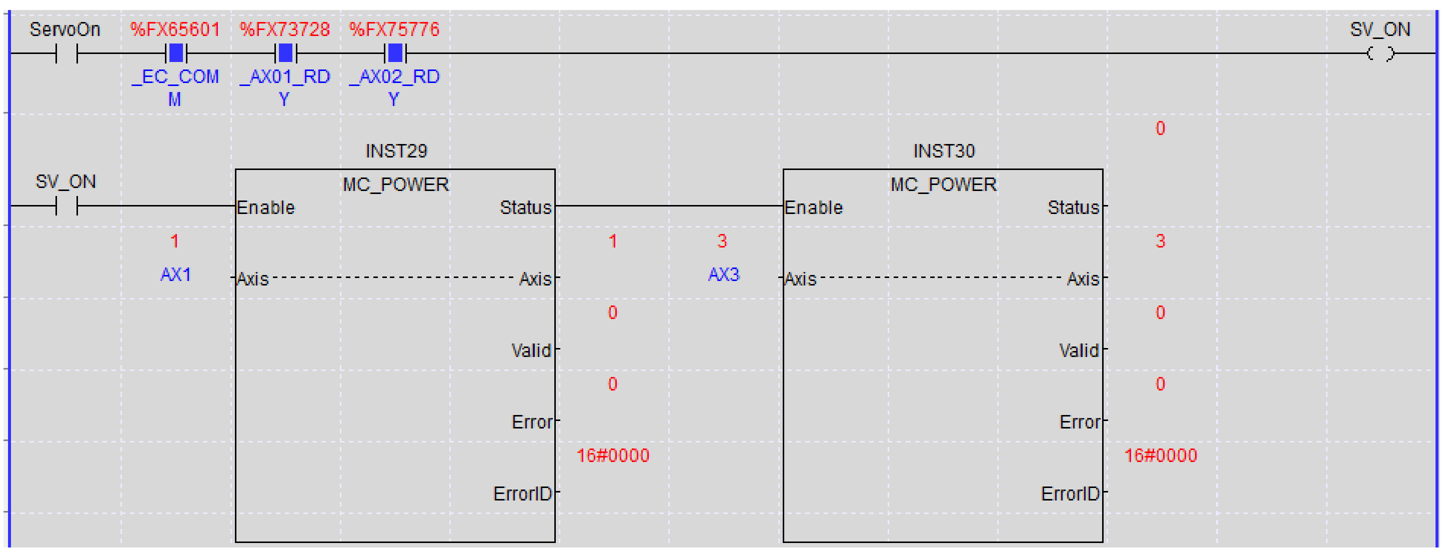
- Execution completed: When motion function block is normally executed, it is On.
- Function Block in execution: When motion function block is executed, it is On, and the execution is completed, It becomes Off.
- Error occurrence state: In case error occurs while the motion function block is being executed, it is On.
- Error number: In case error occurs, the number that corresponds to error is generated.
- For details on the output of motion function block, refer to “Edge motion commands” of “6.1.3 Basic I/O Variable”.

7.4.6 Axis Group Operation Example

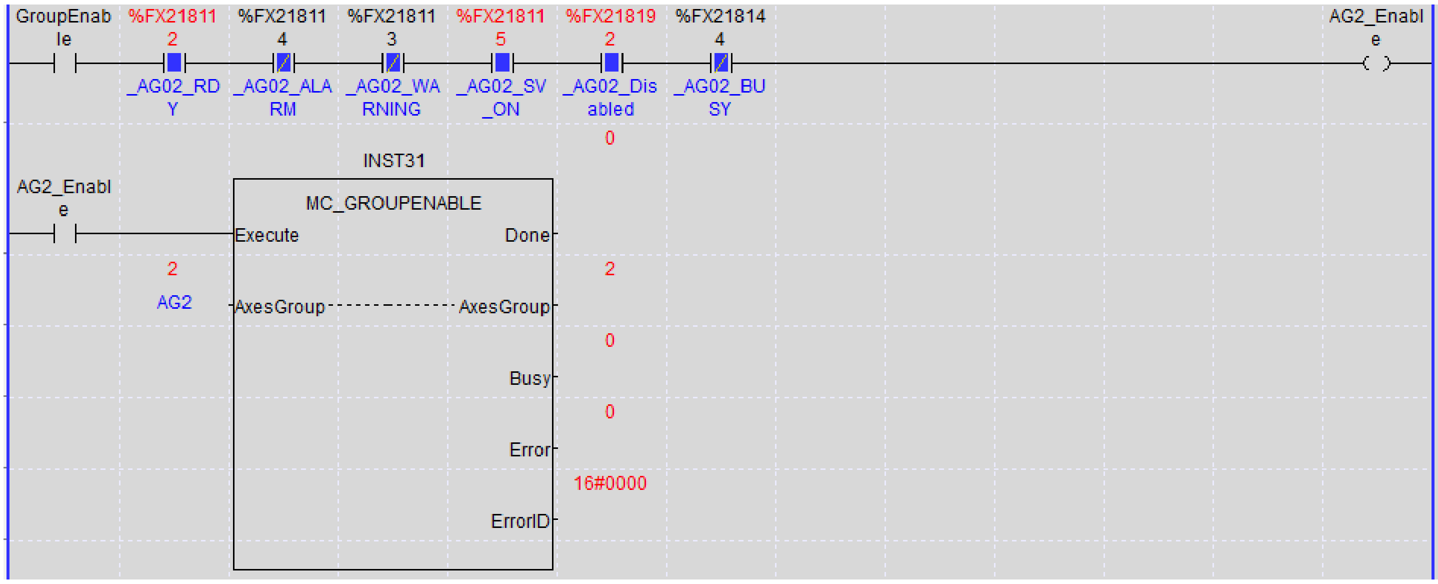
1. Group Configuration



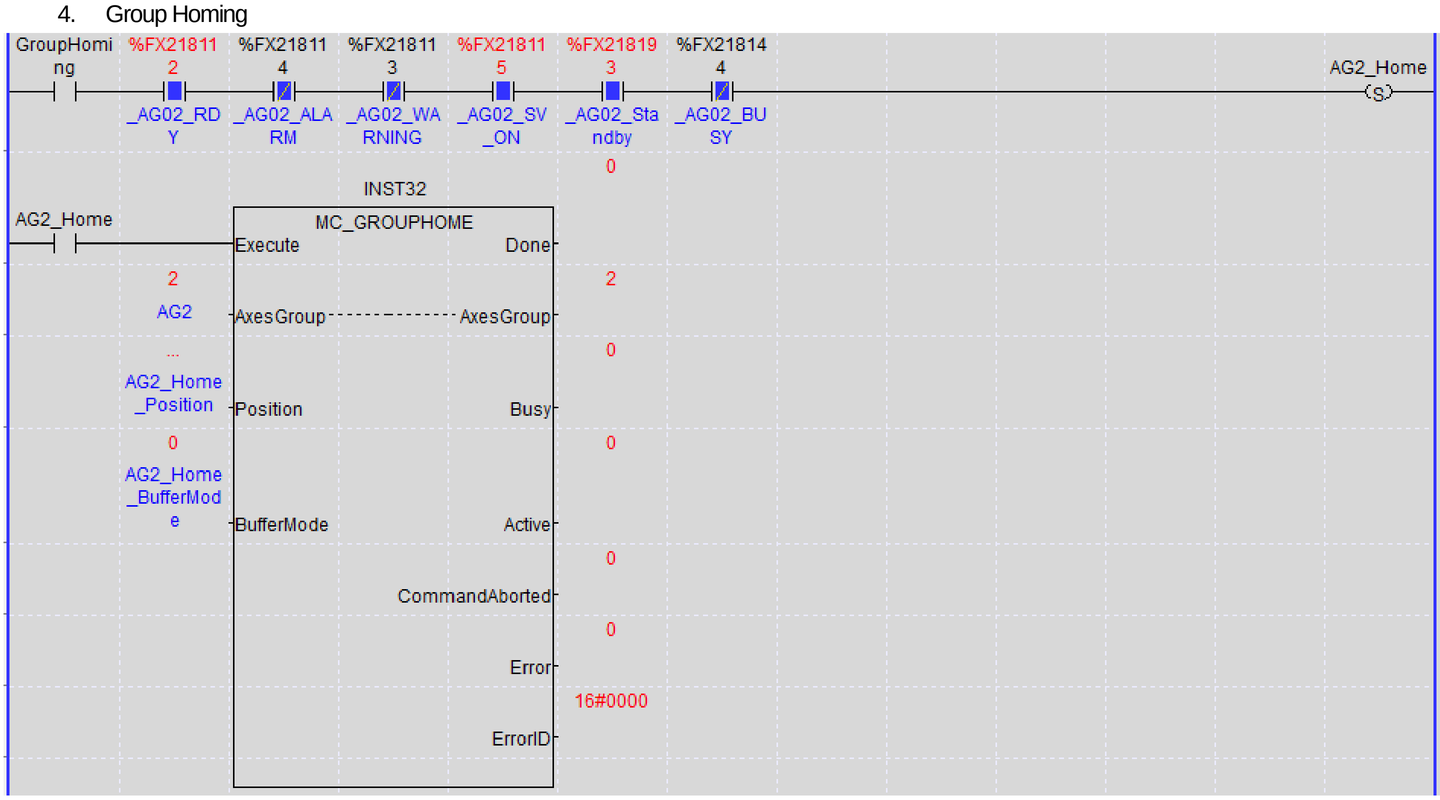
2. Servo On



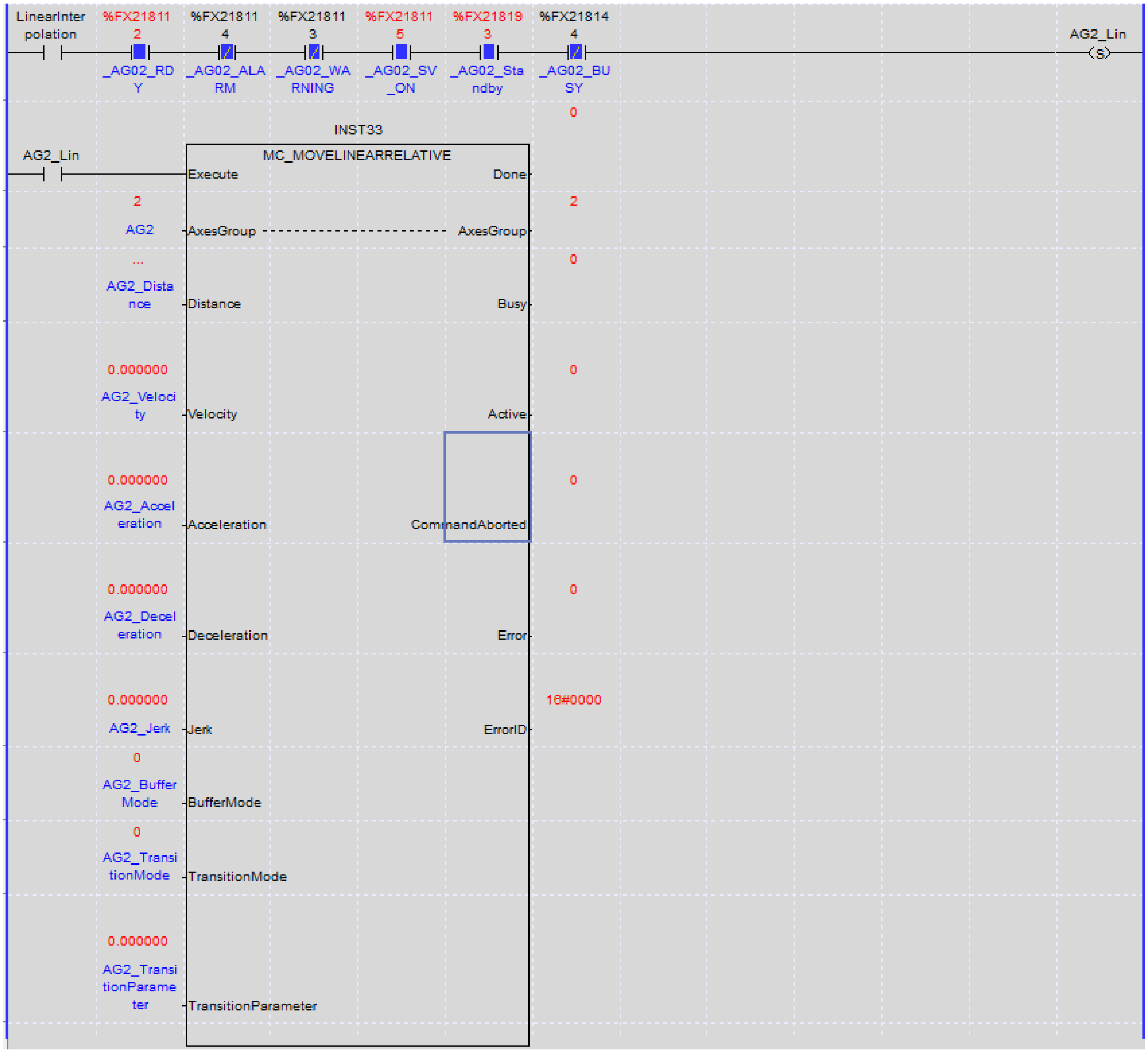
3. Group Enable



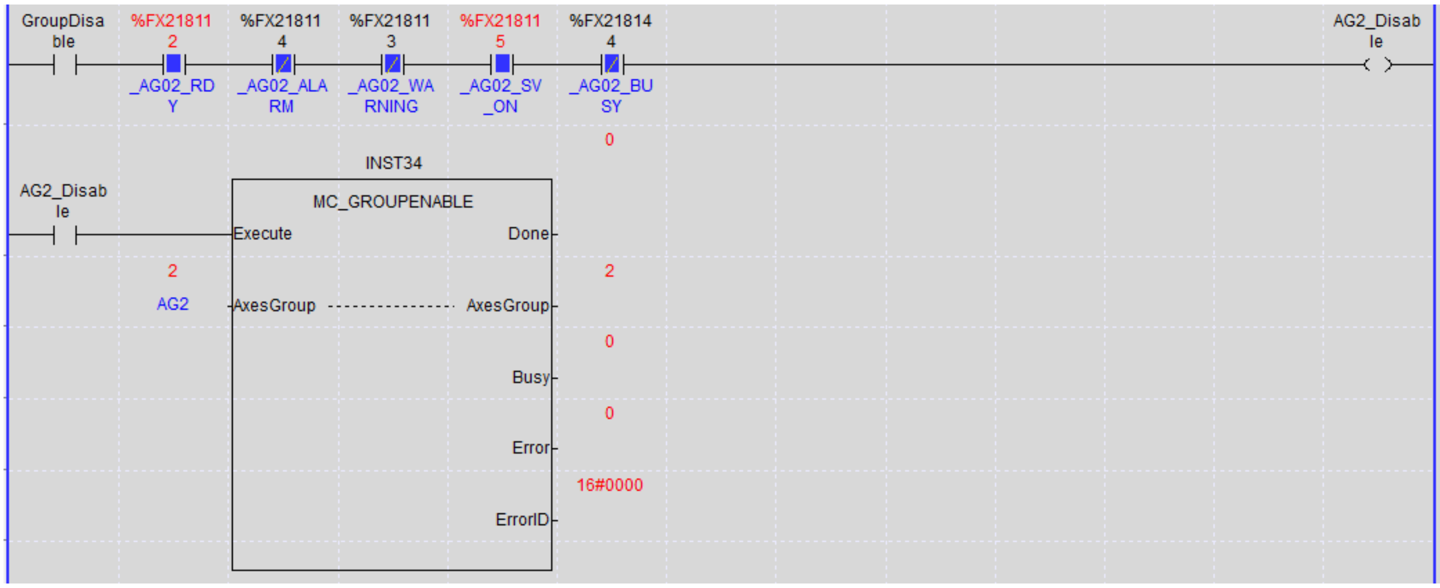
4. Group Homing



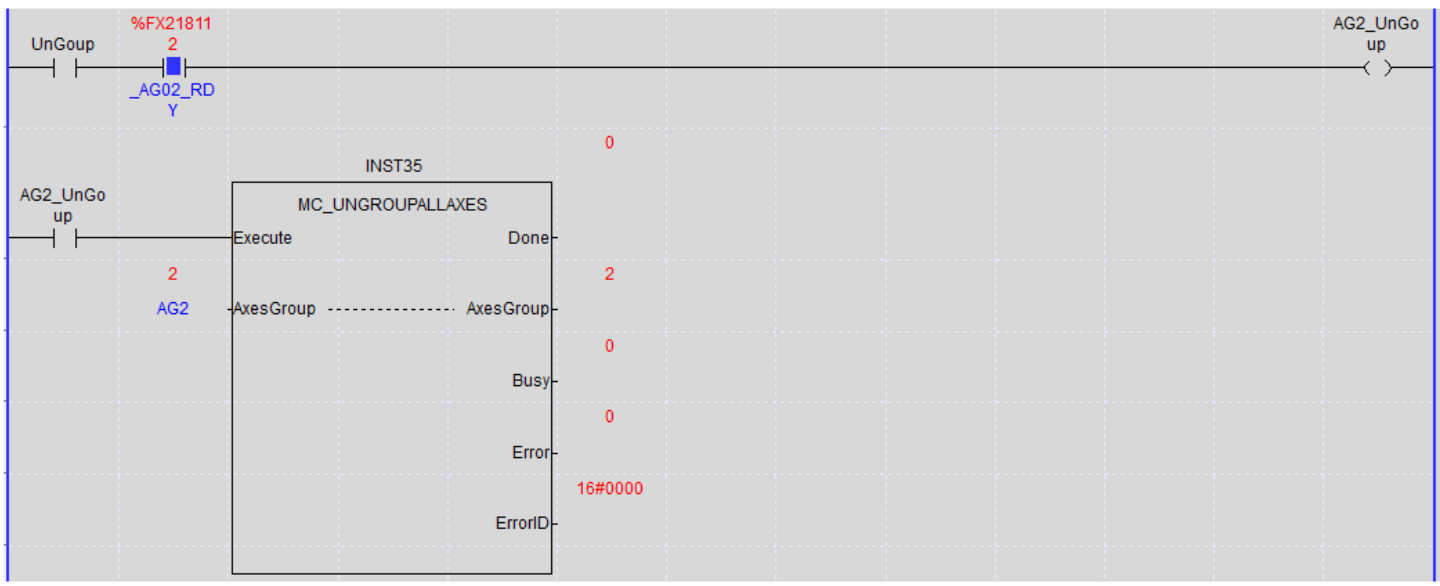
5. Linear Interpolation



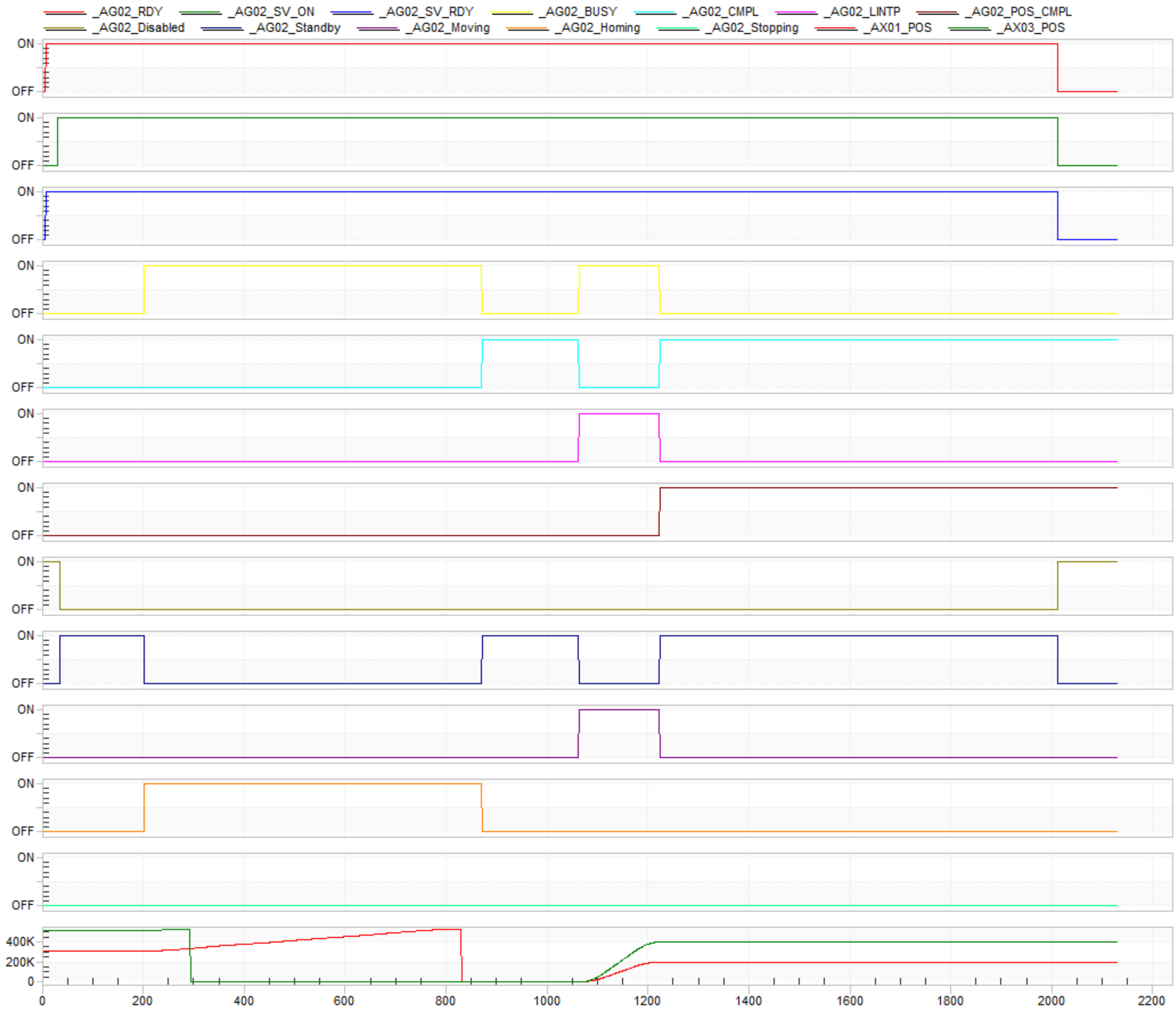
6. Group Disable

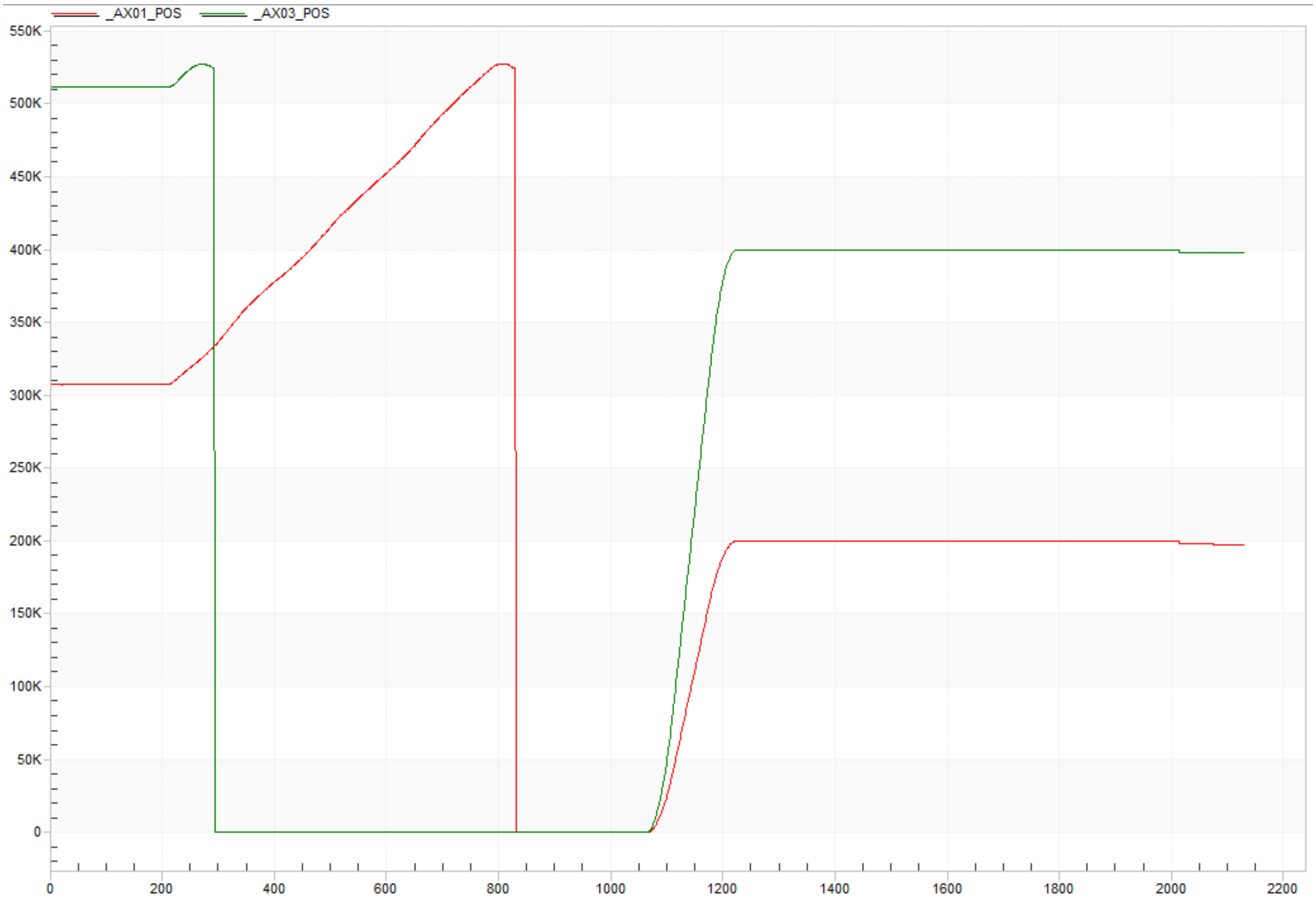


7. Ungroup



8. Timing diagram





_AX01_POS
_AX03_POS
550K
500K
450K
400K
350K
300K
250K
200K
150K
100K
50K
0
0
200
400
600
800
1000
1200
1400
1600
1800
2000
2200

7.5 I/O Processing Program

Motion controller has the input of 8 points and output of 16 points internal, and it can expand input and output points using external EtherCAT input/output modules. EtherCAT input and EtherCAT output modules possible to be mounted on the outside can be expanded up to 64 stations and up to 1,024 points.

7.5.1 Handling of Input signal

Internal input signals and signals inputted in external input module can be used in the program using an internal flag of the motion control module.

For more information on the types and functions of flags, refer to “Appendix 1. Flag list”.

7.5.2 Handling of Output Signal

Internal output signals and signals inputted in external output module can be used in the program using an internal flag of the motion control module.

For more information on the types and functions of flags, refer to “Appendix 1. Flag list”.

Chapter 8 Motion Control Function

8.1 Origin Determination

In case the position control function of motion controller is used, the origin must be determined first to execute commands based on the absolute coordinate position. The position value of absolute coordinates is the distance based on the predetermined origin(0 position). The origin determination means setting the origin of the machine for position control using absolute coordinates.

8.1.1 Origin Determination

(1) Methods to determine the origin

There are two methods to determine the origin of the machine as below.

(a) Homing

It is a method to determine the origin of the machine by moving the machine using a sensor connected to servo drive with homing (MC_Home) motion function block.

When homing command is executed, the origin determination becomes the origin indetermination status, and homing is successfully completed, it becomes the origin determination status.

(b) Current position setting

After moving the machine to a certain position by using JOG operation (LS_Jog) or relative coordinate position control (MC_MoveRelative) motion function block, the position can be set to the specific position with the current position location setting (MC_Setposition) motion function block. In this case, the position is recognized as an absolute coordinate and becomes origin determination status. The origin determination status of axis can be identified with motion axis flag AXxx_HOME_CMPL. (xx: axis number)

(2) Origin determination when using absolute encoders

In case of using absolute encoder in servo drive, absolute data value is maintained by battery backup even if the power is off. Motion control module can continue to maintain the origin determination status by reading the current position from the value of absolute encoder and calculating absolute coordinate position when it is connected to servo drive.

To this end, the encoder selection of basic parameters among operating parameters should be set to ‘1: Absolute encoder’ in case of using absolute encoder. Even though the power of motion control module and servo drive is off after the establishment of origin determination status, the previous origin determination status is maintained by calculating absolute coordinate position when servo drive is connected in case encoder selection parameter is ‘1: Absolute encoder’ when the power is re-applied.

In absolute coordinate system using absolute encoder as above, the absolute coordinate position can be controlled without the origin determination even after power off/on.

To use the absolute coordinate systems, the following conditions should be required.

(a) The servo motor should support the absolute encoder.
(b) Set the absolute encoder setting of servo drive to “absolute encoder enable”.
(c) Set the encoder setting in the axis parameter of motion module to “absolute encode”.

(3) Change to the origin indetermination status

The absolute position control operation cannot be performed since motion control module becomes the origin indetermination status in the following cases.

(d) In case of re-connection after servo drive power off when using an incremental encoder
(e) In case of re-connection after PLC power off/on when using an incremental encoder

(f) In case homing is not normally completed after the execution of homing command

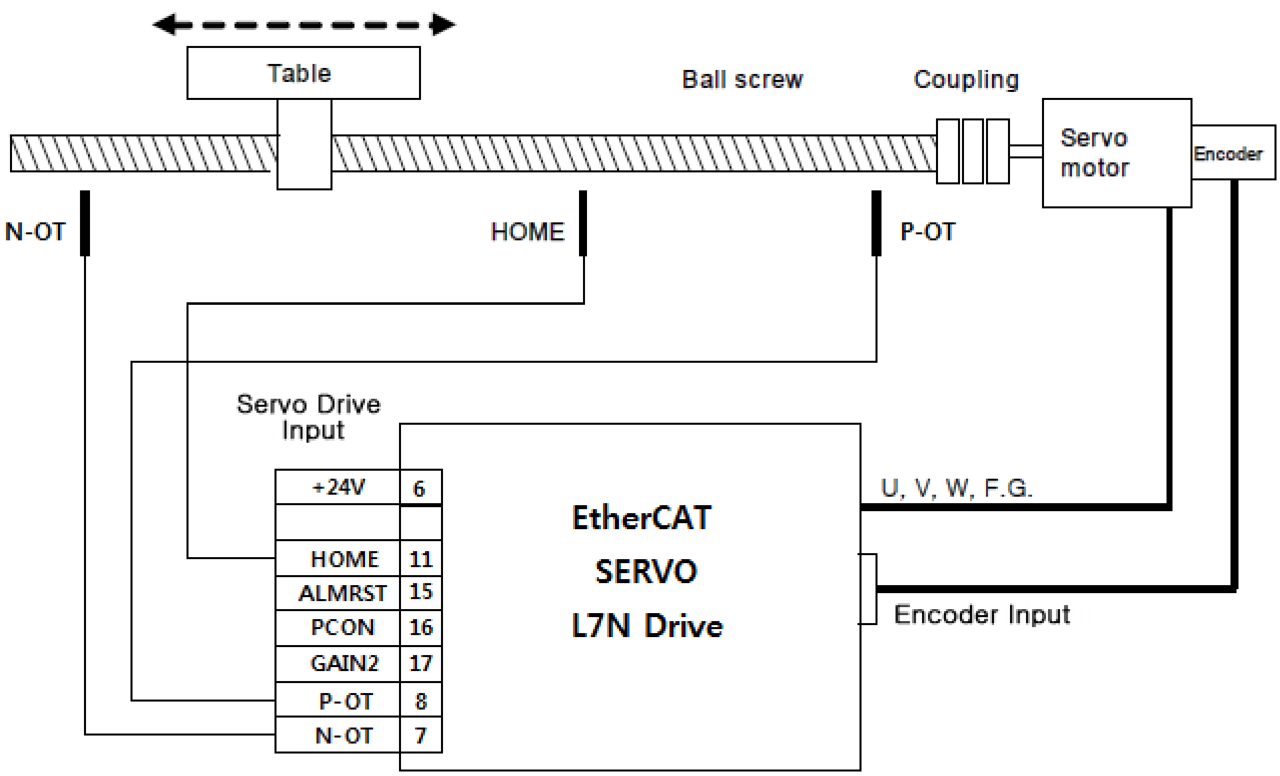
In case of the origin indetermination status as above, the origin determination should be executed for absolute coordinate position control operation.

8.1.2 Homing

(4) Operation

Before performing the homing, parameters related to the homing of servo drive must be set in each axis. Before performing the homing, parameters related to the homing of servo drive must be set in each axis. When the origin position is determined by homing, the origin detection signal is not recognized during the motion control operation.

The contact performed at the time of homing is entered through connector of servo drive (EtherCAT CoE support servo drive). Typical wiring is as follows.



For the performance of homing, a method suitable for the system of users for homing operation mode (EtherCAT CoE support drives: Refer to instruction manual for the relevant drive) should be selected.

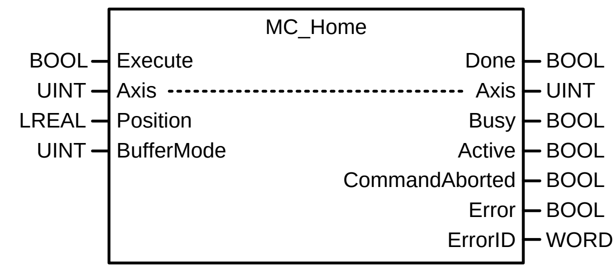
In motion control module, actual operation after starting homing is performed in servo drive, and homing method to support complies with servo drive. Before setting the homing, homing-related parameters are to be set in servo parameters of the axis.

■ Example of setting homing parameters

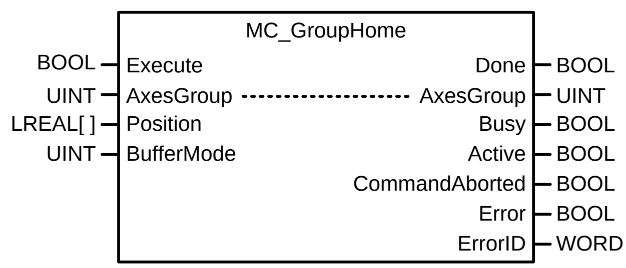
Index	Name	Unit	Current Value	Initial Value	Access
6098	Homing Method	-	0x22	0x22	rw
6099:00	Homing Speeds	-	0x02	0x02	rw
6099:01	Speed during search for switch	Vel.Unit	0x000000A0	0x000000A0	rw
6099:02	Speed during search for zero	Vel.Unit	0x00000020	0x00000020	rw
609A	Homing Acceleration	Acc.Unit	0x0000C350	0x0000C350	rw

- Relevant motion function block

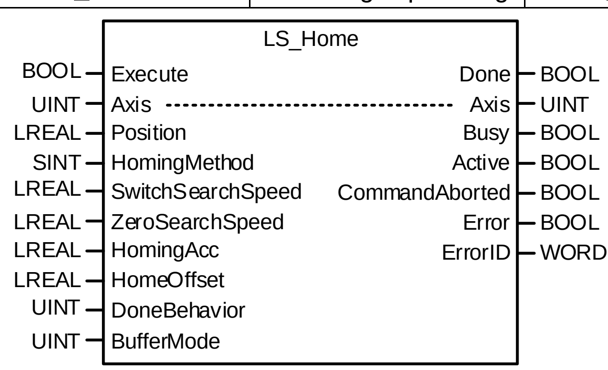
Name	Description	Operation condition
MC_Home	Perform homing	Edge



Name	Description	Operation condition
MC_GroupHome	Perform group homing	Edge

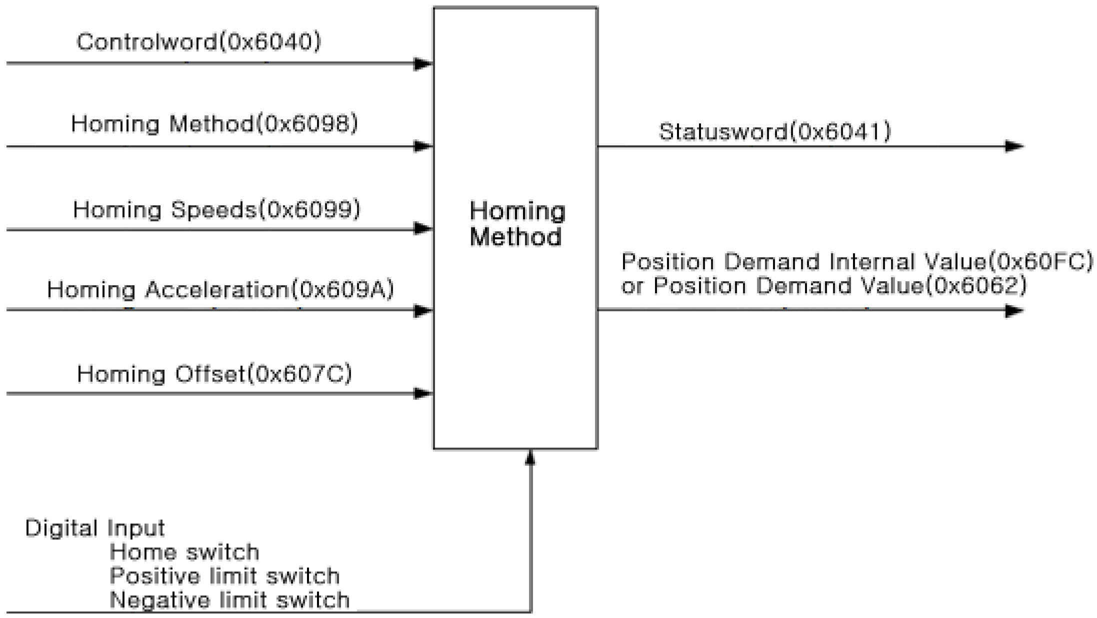


Name	Description	Operation condition
MC_Home	Perform group homing	Edge



(5) XDL- N Series servo drive homing parameters and operation

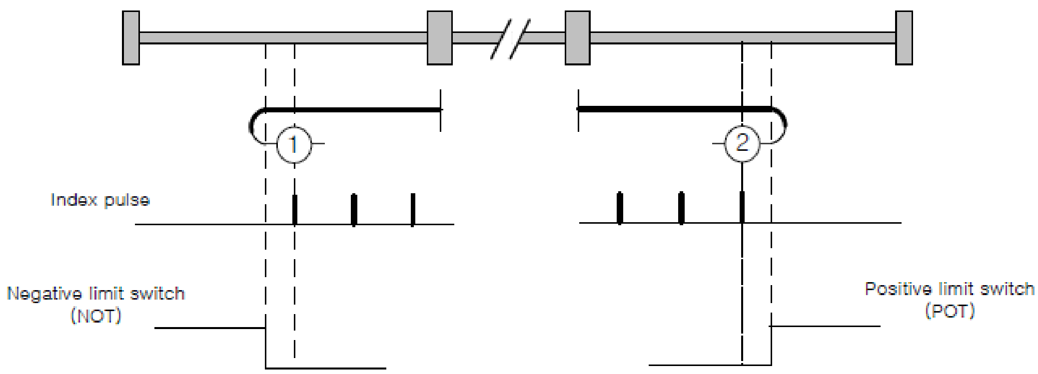
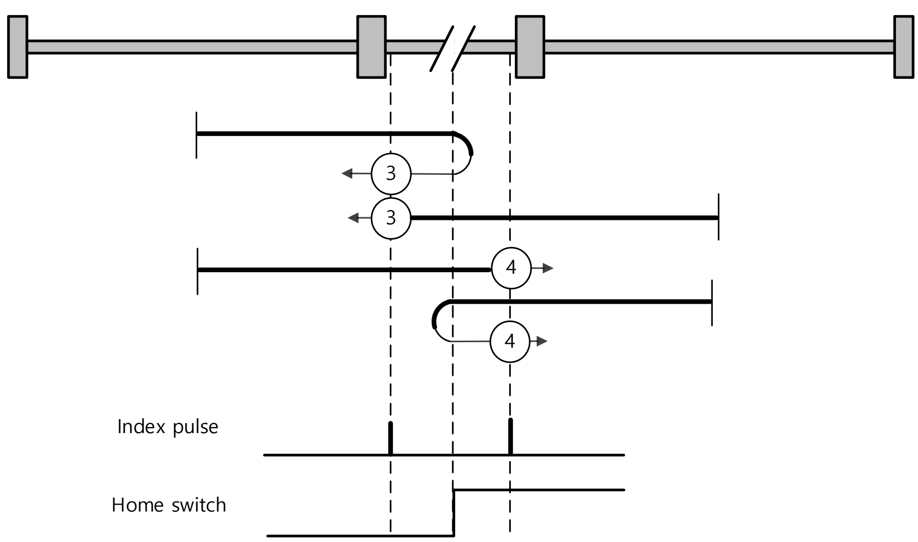
The following figure shows input and output definitions of homing-related XDL N series servo drive parameters. The velocity, acceleration and homing methods can be specified.

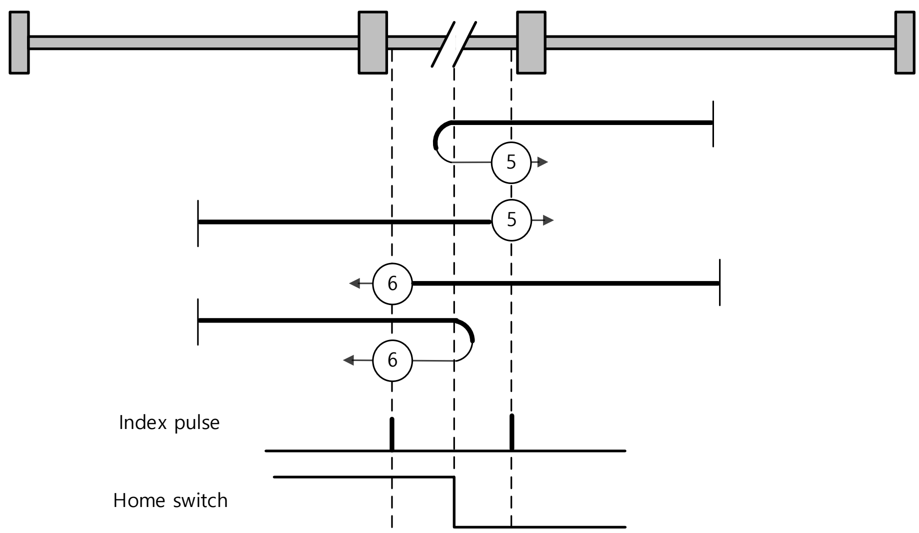


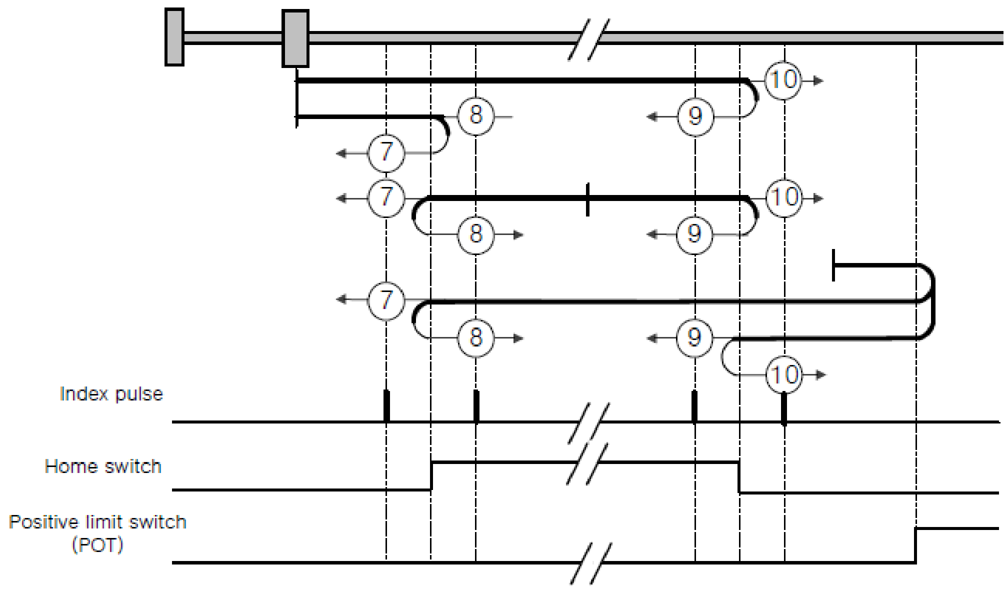
(a) Homing-related Parameters

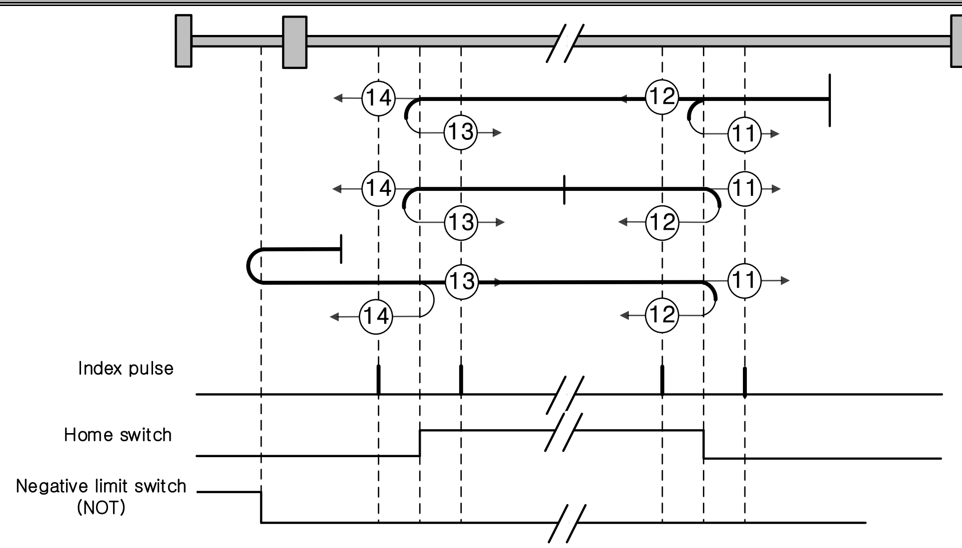
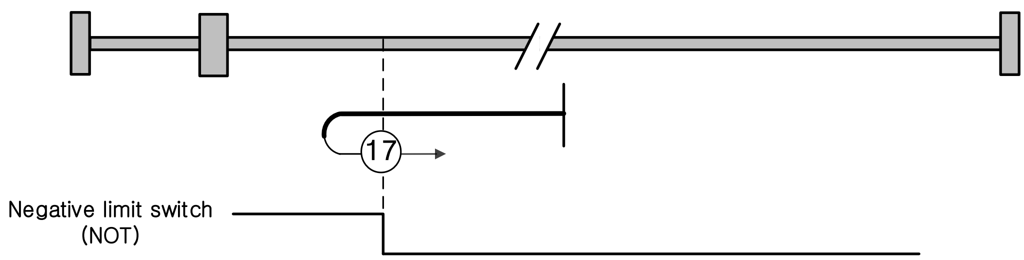
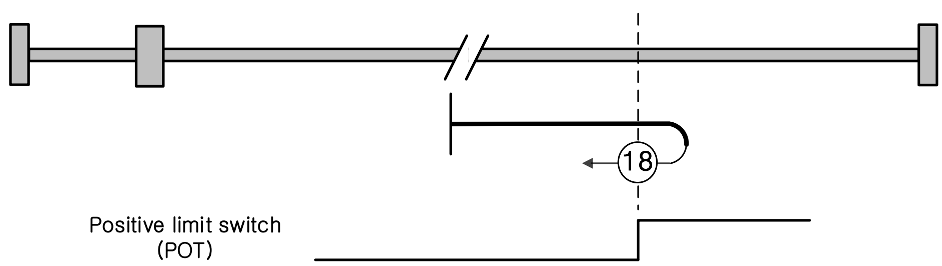
Index	Sub	Name	Data Type	Unit
0x6040	-	Control word	UINT	-
0x6041	-	Status word	UINT	-
0x607C	-	Homing Offset → Cannot be set (can be set in function block)	DINT	[pls]
0x6098	-	Homing Method	SINT	-
0x6099	-	Homing Speeds	-	-
	0	Item Number	USINT	-
	1	Speed during search for Switch	UDINT	[pls/s]
	2	Speed during search for zero	UDINT	[pls/s]
0x607D	-	Software Position Limit	-	-
	0	Item Number	USINT	-
	1	Min position limit	DINT	[pls]
	2	Max position limit	DINT	[pls]
0x609A	-	Homing acceleration	UDINT	[pls/s^2]

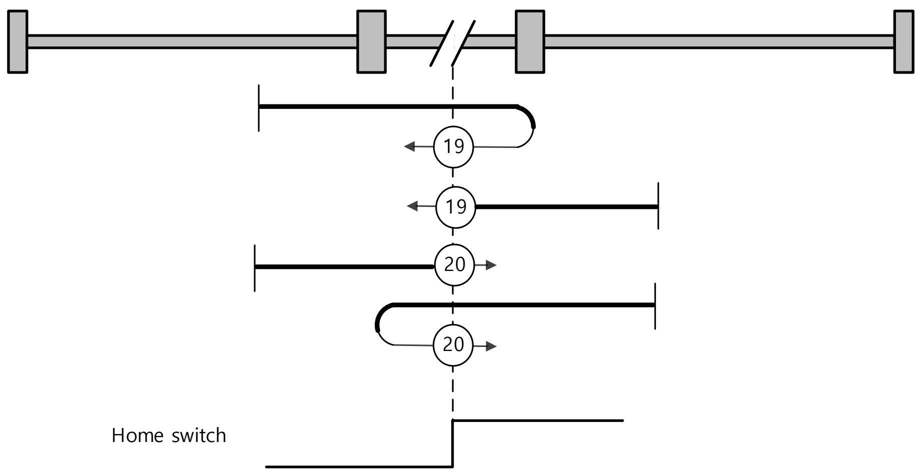
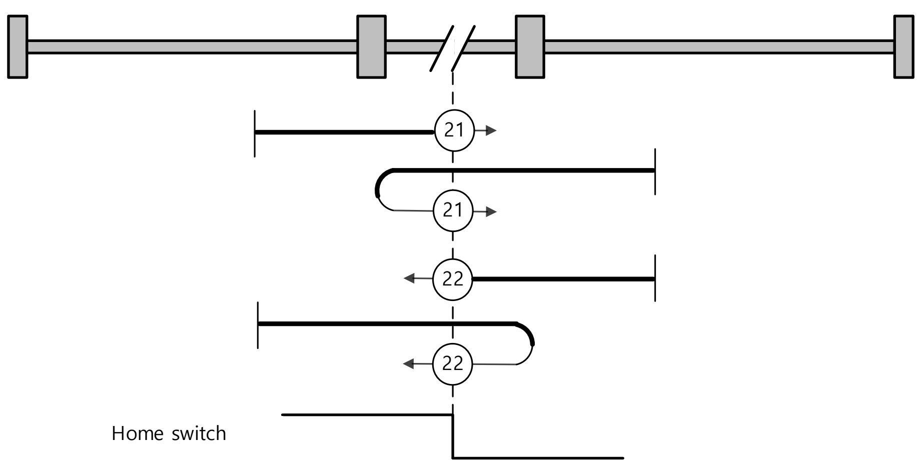
(b) Homing Method (0x6098)

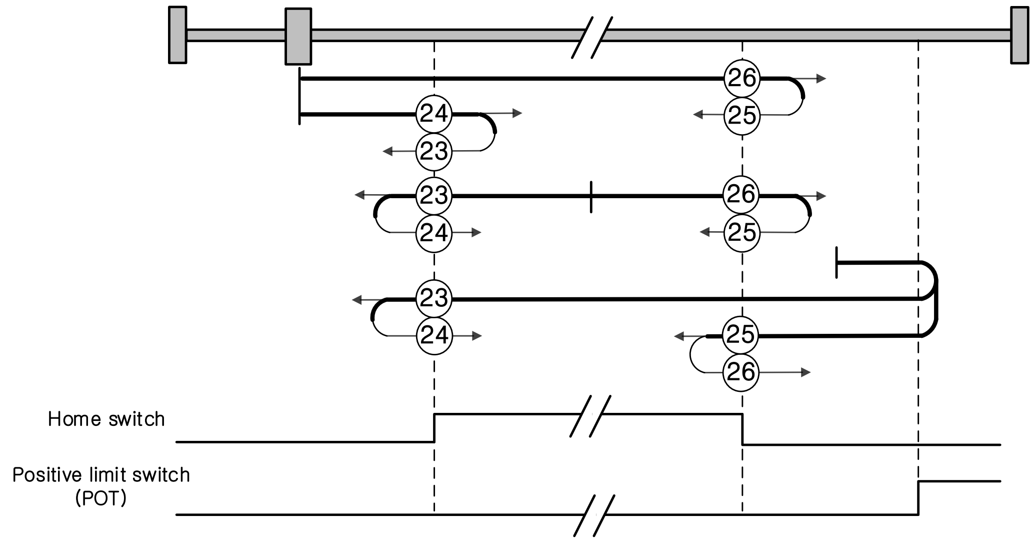
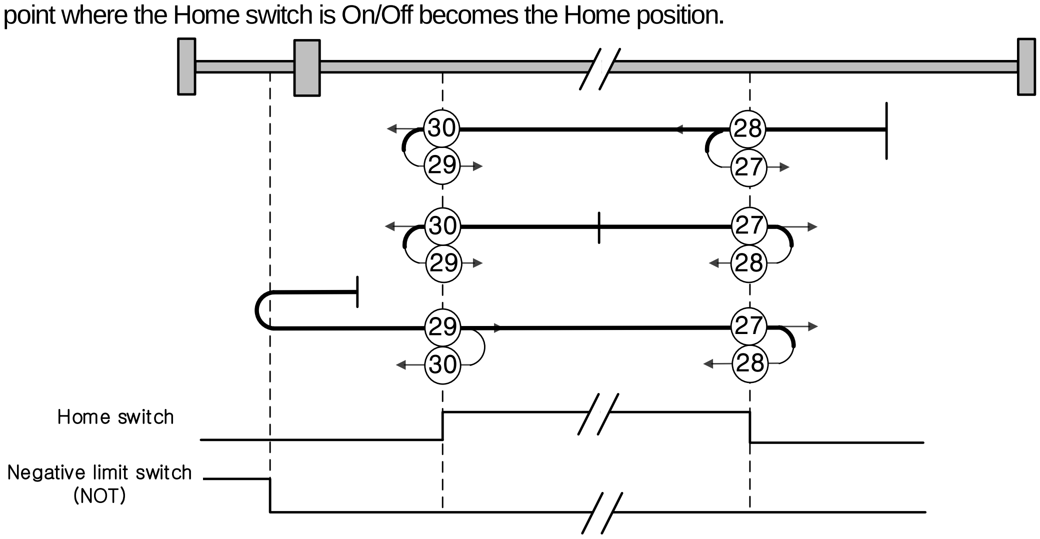
Value	Description	L7NH Supported or not
0	No Homing	Unavailable
1, 2	(1) If NOT switch is Off, the initial movement direction becomes forward direction (CW). If NOT switch is On, change of direction is made. The location that meets the first index pulse during operation in reverse direction (CCW) after NOT switch is On becomes the Home position. (2) If POT switch is Off, the initial movement direction becomes reverse direction (CCW). If POT switch is On, change of direction is made. The location that meets the first index pulse during operation in reverse direction (CW) after POT switch is On becomes the Home position.  	Supported
3, 4	Methods (3) and (4) change the direction when the Home switch is On. After the home switch is On, the point where the first index pulse meets during operation becomes the origin position.  	Unavailable

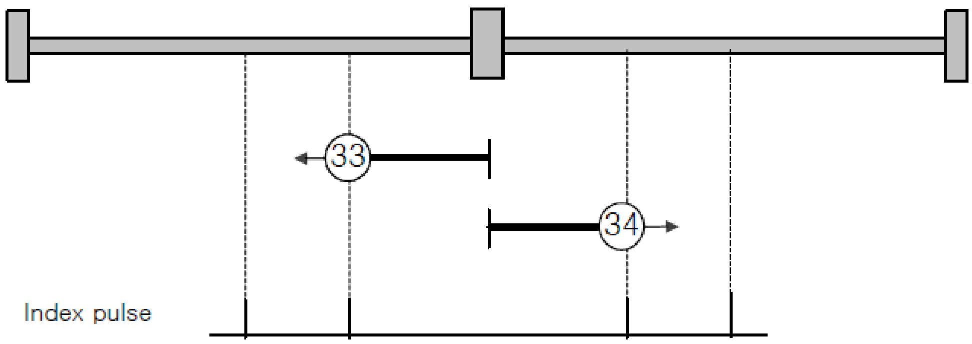
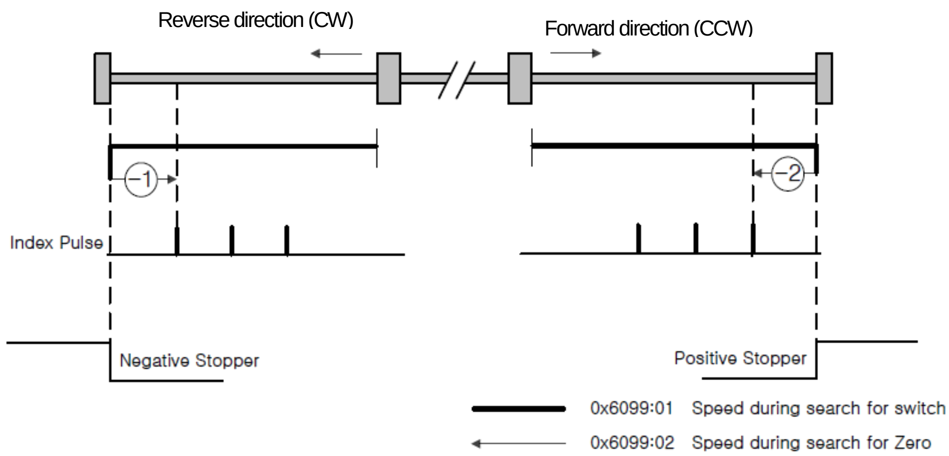
Value	Description	L7NH Supported or not
5, 6	Methods (5) and (6) change the direction when the Home switch is off. After the home switch is Off, the point where the first index pulse meets during operation becomes the origin position.  	Unavailabl e
7~10	Through (7) to (10) methods, the origin position is determined by the Home switch and POT switch. (7) Upper figure: If POT switch is Off, operation is made at switch search velocity, and the initial movement direction becomes reverse direction CCW. If the Home switch is On, change of direction is made. Afterwards, the location that meets the first index pulse during operation in forward direction CW becomes the Home position, and operation is made at Zero search velocity. (7) Middle figure: If POT switch is Off, and the Home switch is On, operation is made at switch search velocity, and the initial movement direction becomes forward direction CW. If the Home switch is Off, the speed is changed to Zero search velocity. Afterwards, the location that meets the index pulse first during operation in forward direction CW becomes the Home position. (7) Below figure: If POT switch is Off, and the Home switch is On, operation is made at switch search velocity, and the initial movement direction becomes forward direction CCW. If POT switch is On, change of direction is made. When the Home switch is changed from On to Off, operation is made at Zero search velocity, and the location where that meets index pulse first during continuous operation in forward direction CW becomes the Home position. (8) to (10) methods have the same positioning concept in homing with the above (7) method except for the initial operational direction and motions according to the Home switch polarity. Refer to the figure below.	Supported

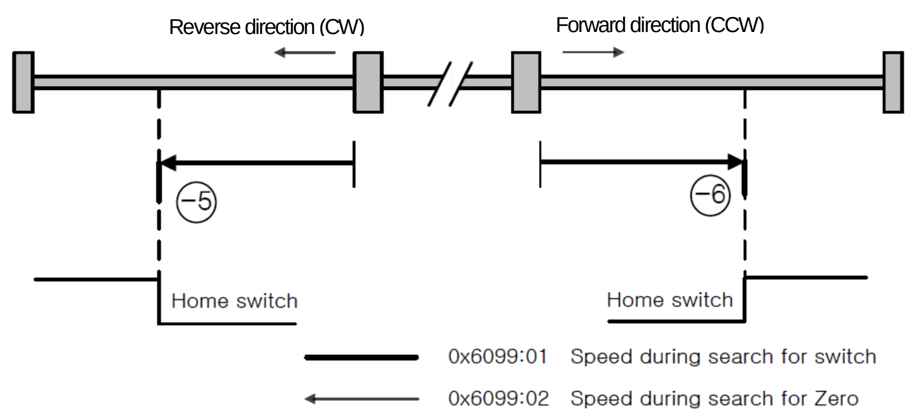
<table>
<tr><th>Value</th><th>Description</th><th>L7NH Supported or not</th></tr>
<tr><td></td><td>
</td><td></td></tr>
<tr><td>11~14</td><td>Through (11) to (14) methods, the origin position is determined by the Home switch and NOT switch.
(11) Upper figure: If NOT switch is Off, operation is made at switch search velocity, and the initial movement direction becomes reverse direction CW. If the Home switch is On, change of direction is made. Afterwards, the location that meets the first index pulse during operation in forward direction CCW becomes the Home position, and operation is made at Zero search velocity.
(11) Middle figure: If NOT switch is Off, and the Home switch is On, operation is made at switch search velocity, and the initial movement direction becomes forward direction CCW. If the Home switch is Off, the speed is changed to Zero search velocity. Afterwards, the location that meets the index pulse first during operation in forward direction CCW becomes the Home position.
(11) Below figure: If NOT switch is Off, and the Home switch is On, operation is made at switch search velocity, and the initial movement direction becomes forward direction CW. If NOT switch is On, change of direction is made. When the Home switch is changed from On to Off, operation is made at Zero search velocity, and the location where that meets index pulse first during continuous operation in forward direction CCW becomes the Home position.
(11) to (14) methods have the same positioning concept in homing with the above (11) method except for the initial operational direction and motions according to the Home switch polarity. Refer to the figure below.</td><td>Supported</td></tr>
</table>

Value	Description	L7NH Supported or not
		
17	Methods (17) change the direction when the NOT switch is On. The position where the NOT switch is turned off during operation after being turned on becomes the origin position. 	Unavailable
18	Methods (18) change the direction when the POT switch is On. The position where the POT switch is turned off during operation after being turned on becomes the origin position. 	Unavailable

Value	Description	L7NH Supported or not
19, 20	The Home position is determined as method (3),(4), but index pulse is not used. In addition, the point where the Home switch is On/Off becomes the Home position. 	Unavailable
21, 22	The Home position is determined as method (5),(6), but index pulse is not used. In addition, the point where the Home switch is On/Off becomes the Home position. 	Unavailable

Value	Description	L7NH Supported or not
23 , 24 , 25, 26	The Home position is determined as method (7)~(10), but index pulse is not used. In addition, the point where the Home switch is On/Off becomes the Home position. 	(24) Available, others Unavailable
27 , 28 , 29, 30	The Home position is determined as method (11)~(14), but index pulse is not used. In addition, the point where the Home switch is On/Off becomes the Home position. 	(28) Available, others Unavailable

Value	Description	L7NH Supported or not
33, 34	The location that meets index pulse first during movement in the reverse direction CCW/forward direction CW becomes the Home position. 	Supported
35	Homing operation starting point becomes the Home position.	Supported
-1	The first movement direction is CW direction, and it operates at the switch search speed. If it collides with the reverse stopper (Negative Stopper), it waits according to the torque limit value (0x2409) when returning to the origin using the stopper and the time set value (0x240A) when returning to the origin using the stopper, and then changes the direction. After that, it detects the first index pulse during operation at the zero search speed and operates to the index position (Home). 	Supported L7NH dedicated
-2	The home position is determined as in the method (-1), but the initial movement direction is CCW, and it collides with a positive stopper.	Supported L7NH dedicated

Value	Description	L7NH Supported or not
-3	The first movement direction is CW direction, and it operates at the switch search speed. If it hits a reverse stopper (Negative Stopper), it waits according to the torque limit value (0x2409) when returning to the origin using the stopper and the time set value (0x240A) when returning to the origin using the stopper, and then the return to the origin is completed.	Supported L7NH dedicated
-4	The home position is determined as in the method (-3), but the initial movement direction is CCW, and it collides with a positive stopper.	Supported L7NH dedicated
-5	The first movement direction is CW direction, and it operates at the switch search speed. When the home switch is On, the home position return is completed after decelerating to a stop. It returns to the origin using only the home switch, and if it meets the limit switch during homing, it decelerates and stops after a homing error occurs. 	Supported L7NH dedicated
-6	The home position is determined as in the method of (-5), but the initial movement direction is CW.	Supported L7NH dedicated

Note) ▬ : Speed during search for switch (0x6099:01), → : Speed during search for zero (0x6099:02)

(6) MC_STOP (immediate stop) command operation during homing

L7NH series can give a stop command with MC_STOP command during homing. At this time, the deceleration operation varies according to the Quick Stop Option Code (0x605A) parameter.

(a) Stop operation according to Quick Stop Option Code (0x605A)

0x605A	Quick Stop Option Code						ALL
Variable tyype	Setting range	Initial value	Unit	Accesibility	PDO assignment	Chaging property	Store
INT	0 to 4	2	-	RW	No	Always	Yes

Setting Value	Description
0 , 3 , 4	Deceleration stop with the value set in homing acceleration (0x609A)
1, 3	Deceleration stop with the value set in Quick Stop Deceleration (0x6085)

8.2 Control Operation Type

Motion control modules execute control through programs set in motion control program. Kinds of motion control operations include speed position control, speed velocity control, speed torque control, interpolation control, switching control between position/velocity, switching control between position/torque, and switching control between velocity/torque.

8.2.1 Single-axis Position Control

It conducts position control of the axis specified after the execution by motion function block (「Relative position operation (MC_MoveRelative)」 and 「Absolute position operation (MC_MoveAbsolute)」) from starting position (current stop position) to target position (position of the point to move)

(1) Control by absolute coordinate method (「Absolute position operation (MC_MoveAbsolute)」)

(a) It conducts position control from starting position to target position (location specified in 'Position' of absolute pos ition operation command).

(b) Position control is executed based on position (the origin position) specified in the homing.

(c) In direction input, the direction to operate is specified. It is valid only if operation parameter 「Infinite running repetition」 setting is '1: Enable'.

- Setting value: 0-Not specified, 1–Forward direction, 2–Shortest distance direction, 3–Reverse direction, 4–Current direction
- When the shortest direction distance is specified, the operation is made by selecting the direction that can go to the shortest direction automatically depending on the form of the axis.
- Motions according to the direction (Direction) input are as follows.

1) 0- Not specified

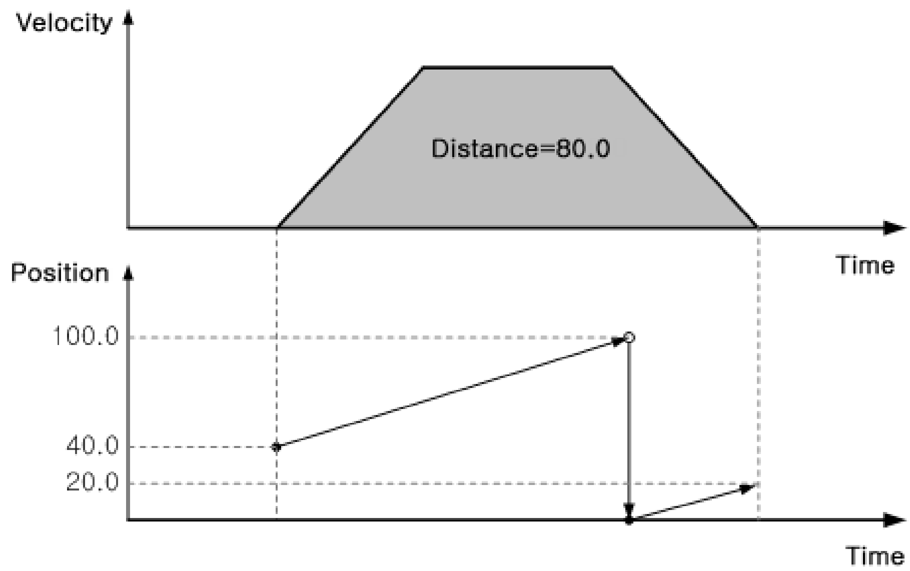
The position value that exceeds repetitive length repetition position can be specified. In case of setting the position value that exceeds the infinite running repetition position, the difference from target position to current position becomes positioning distance. The command position after the absolute position operation is calculated by the following equation.

Command position = Target position – (Infinite running repetition position x n)

(n: Integer value in which infinite running repetition position x n does not exceed the target position)

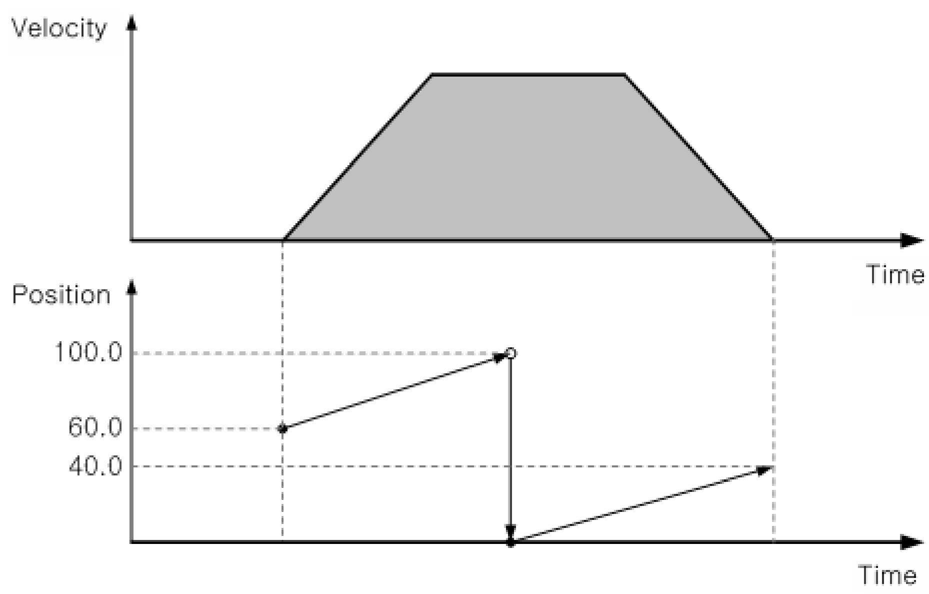
[Example] The absolute position operation is executed with the following settings.

- Infinite running repetition position: 100.0
- Current position : 40.0
- Goal Position : 120.0
- Command position after the absolute position operation = 120.0 – (100.0 x 1) = 20.0



2) 1-Forward direction
Positioning is executed toward the absolute position of forward direction. In case the target position is set with the range that exceeds infinite running repetition position, error (error code: 0x1081) occurs.

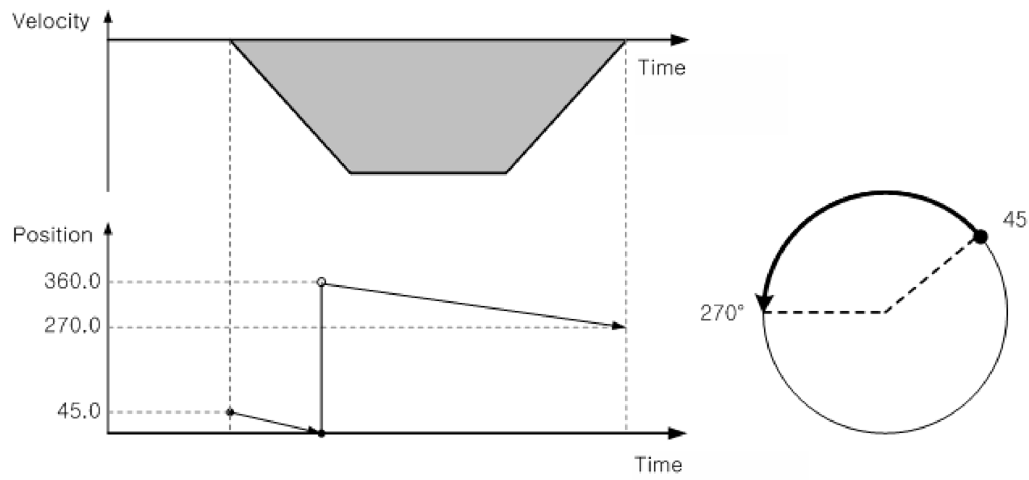
[Example] The absolute position operation is executed with the following settings.
- Infinite running repetition position: 100.0
- Current position : 60.0
- Goal Position : 40.0



3) 2-Shortest distance direction
Positioning is executed by automatically determining the direction of rotation possible to move through shorter distance from the starting position to target position. That is, positioning toward closer direction to target position based on the starting position is carried out.
In case the target position is set with the range that exceeds infinite running repetition position, error (error code: 0x1081) occurs.

[Example] The absolute position operation is executed with the following settings.

- Infinite running repetition position: 360.0
- Current position : 45.0
- Goal Position : 270.0
- Since the movement distance is 225.0°in case of the operation in forward direction, and 135.0°in case of the operation in reverse direction, operation is made in reverse direction, the shortest distance direction.

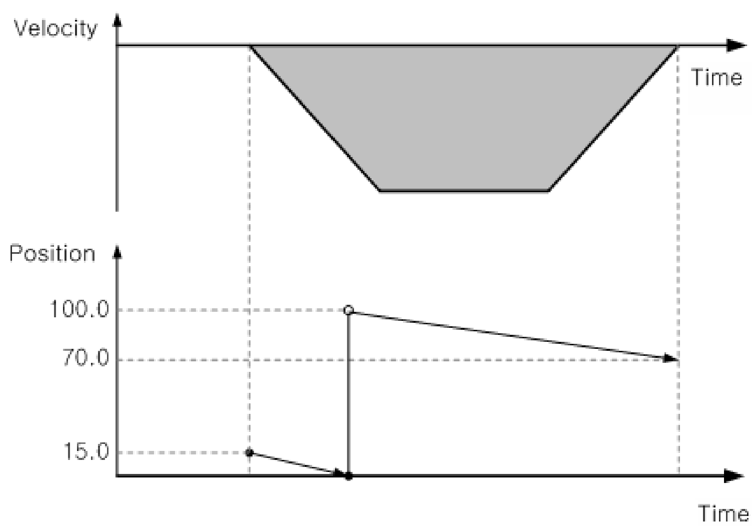


4) 3-Reverse direction

Positioning is executed toward the absolute position of reverse direction. In case the target position is set with the range that exceeds infinite running repetition position, error (error code: 0x1081) occurs.

[Example] The absolute position operation is executed with the following settings.

- Infinite running repetition position: 100.0
- Current position : 15.0
- Goal Position : 70.0



5) 4-Current direction

Positioning is executed depending on the current operating direction.

In case the current operating direction is forward, operation is made in the same way as in Direction='1-forward direction' setting.

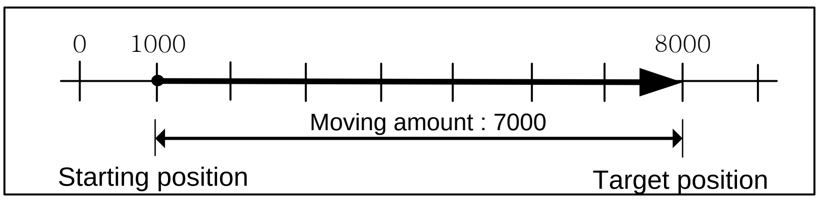
In case the current operating direction is reverse, operation is made in the same way as in Direction=‘3 reverse direction’ setting.

(d) In case operation parameter 「Infinite running repetition」 setting is ‘0: disable’, operating direction is determined as follows.

- Starting position < target position: Positioning operation in forward direction
- Starting position > target position: Positioning operation in reverse direction

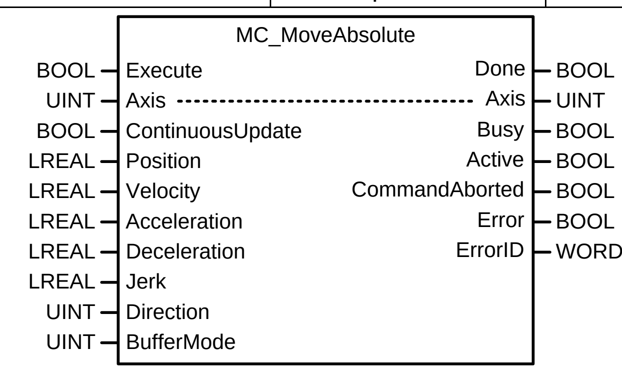
[Example] Executes Absolute coordinate, single-axis position control with the following setting

■ Starting Position: 1000
Goal Position: 8000
The moving amount to forward direction is 7000 (7000=8000-1000).



■ Related Motion Function Block

Name	Description	Operation condition
MC_MoveAbsolute	Absolute positioning operation	Edge



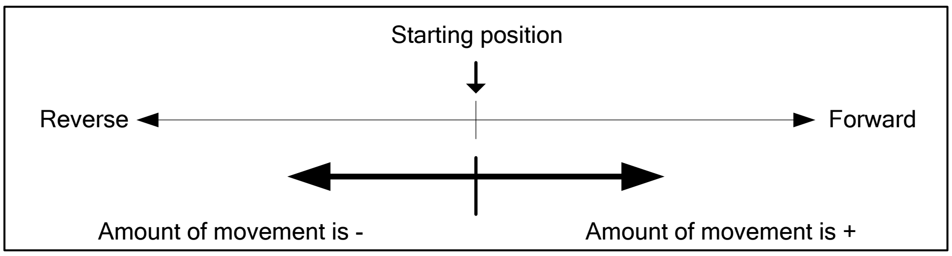
(2) Control by Incremental method (「Relative positioning operation(MC_MoveRelative)」)

(a) It moves the object as much as the target moving amount from start position. Unlike the target position of the absolute coordinate, the value specified on target position is not position value. That is a transfer amount from the starting position.

(b) Transfer direction is determined by the sign of moving amount.

■ The target position value is positive (+ or unsigned): forward direction (current position increase) positioning

■ The target position value is negative (-): reverse direction (current position decrease)

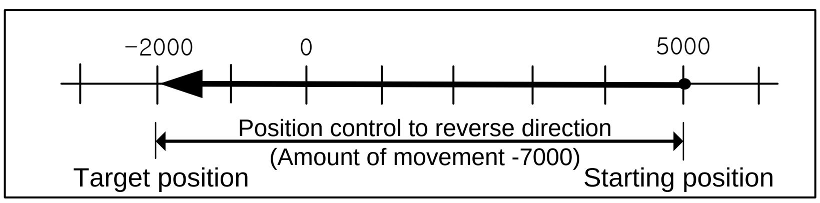


[Example] Executes Absolute coordinate, single-axis position control with the following setting

■ Starting Position: 5000

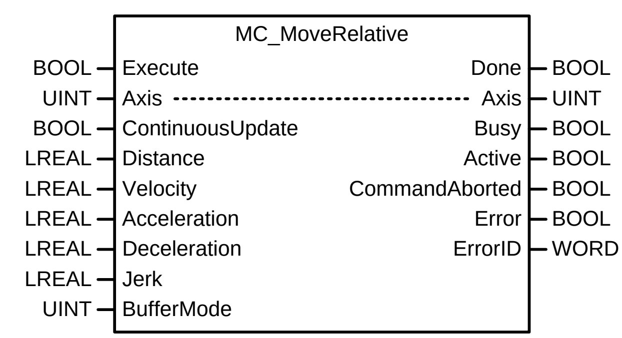
Goal Position: -7000

It goes to reverse direction and stops at the –2000.



■ Related Motion Function Block

Name	Description	Operation condition
MC_MoveRelative	Relative positioning operation	Edge



8.2.2 Single-axis Velocity Control

Execution is made by motion function block(「Specified velocity operation (MC_MoveVelocity)」), and operation is performed at the set velocity until stop condition is inputted.

(1) Features of control

(a) Speed control operation of the specified axis is executed using specified velocity and acceleration/deceleration. The velocity control is executed through a method to transmit the target position value that corresponds to the target velocity using position control of servo drive.

(b) In direction input, the direction to operate is specified.
(However, the forward direction is based on the operating direction specified with the target velocity (Velocity) input. For example, if a negative value is specified in target velocity (Velocity) value, and reverse direction in direction (Direction) input, the axis is finally operated in forward direction.)
■ Setting value: 1-Forward, 2-Reverse, 3-Curent direction

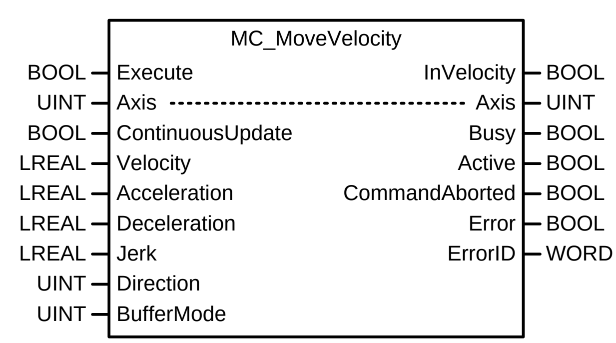
(c) Negative number can be set for target velocity (Velocity) input value. In case the target velocity setting value is negative number, operating direction becomes the opposite direction of the previously specified direction.
■ Forward operation
- Velocity > 0, Direction=1: Forward
- Velocity < 0, Direction=2: Reverse
■ Reverse operation
- Velocity > 0, Direction=2: Reverse
- Velocity < 0, Direction=1: Forward

(d) After reaching the target velocity, InVelocity output of the function block is On (On). If there is a pending command, the pending command is executed after InVelocity output is On.

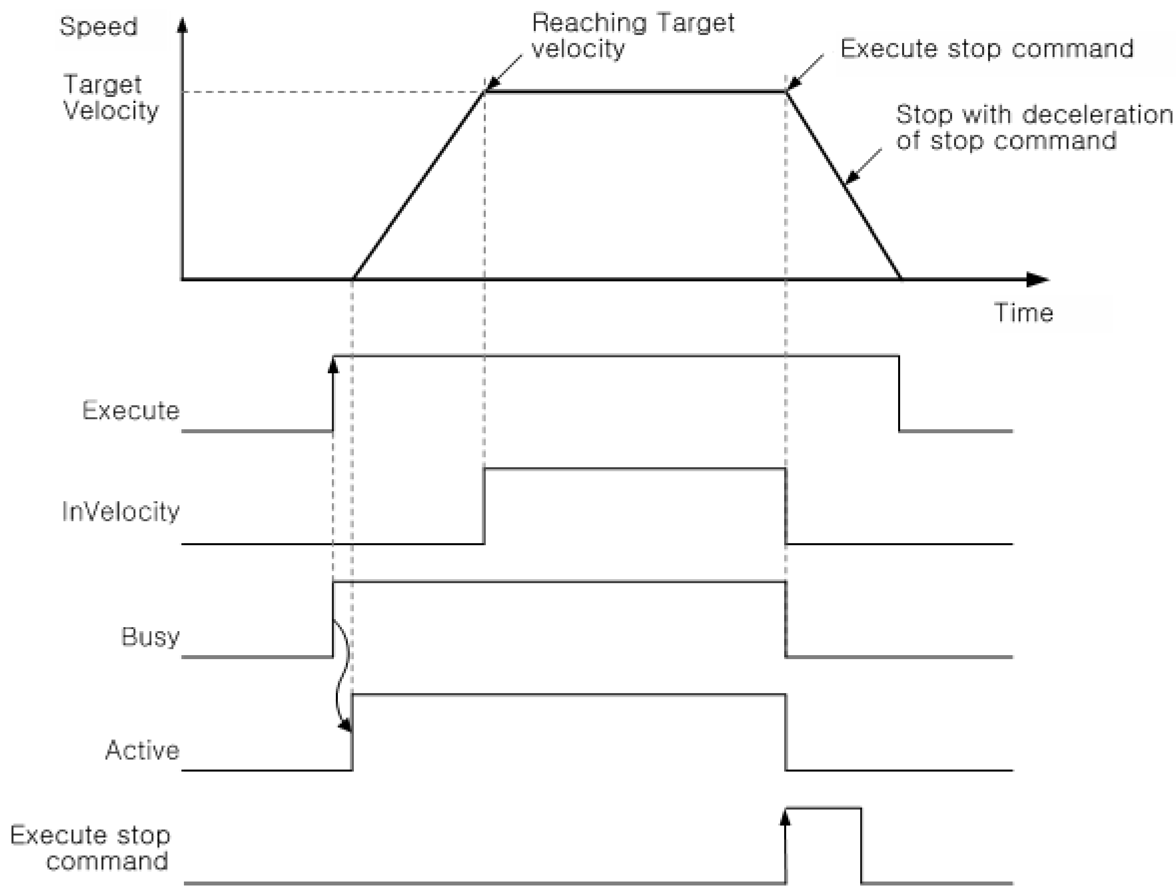
(e) The speed control which is currently being executed is stopped with halt (MC_Halt) or immediate stop (MC_Stop) motion function block.

(2) Related Motion Function Block

Name	Description	Operation condition
MC_MoveVelocity	Specified velocity operation	Edge



(3) Operation timing



8.2.3 Single-axis torque Control

If motion function block(「Torque control(MC_TorqueControl)」) is executed, torque control of the axis is made with the set torque value.

(1) Features of control

(a) Torque control of the specified axis is made using target torque value and torque rising slope.

(b) Torque rising slope (TorqueRamp) is the rate of change in torque per second to the target torque, and time to reach the target torque can be calculated as follows.

Time to reach the target torque(s) = target torque (Torque) / torque rising slope (TorqueRamp)

(c) Torque control mode is executed using torque control mode of servo drive.

(d) Target torque values are rounded to two decimals and reflected in [0.1%] unit.

(e) Specify the operation direction in Direction input.

(However, the forward direction is based on the operating direction specified with the Torque input. For example, if a negative value is specified in Torque value, and reverse direction in direction (Direction) input, the axis is finally operated in forward direction.)

■ Setting value: 1-Forward, 2-Reverse, 3-Curent direction

(f) Negative number can be set for Torque (target torque) input value. In case the target torque setting value is negative number, operating direction becomes the opposite direction of the previously specified direction.

■ Forward operation

- Torque < 0, Direction=1:
- Torque > 0, Direction=2:

■ Reverse operation

- Torque > 0, Direction=2:
- Torque < 0, Direction=1:

(g) The setting range of the target torque value is as follows.

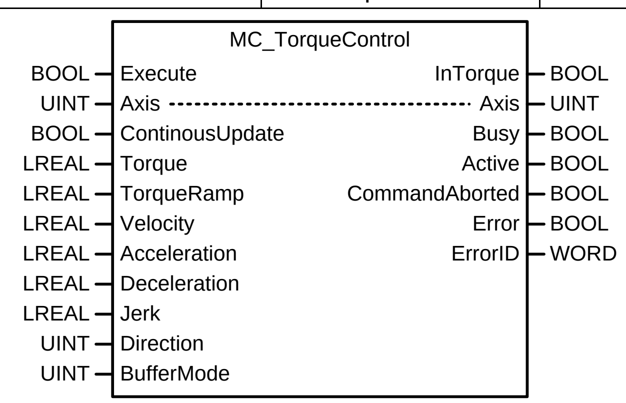
-1000.0 % ~ 1000 %

(h) After reaching the target torque, Intorque output of function block is On. In case there is a pending command, the pending command is executed after InTorque output is On.

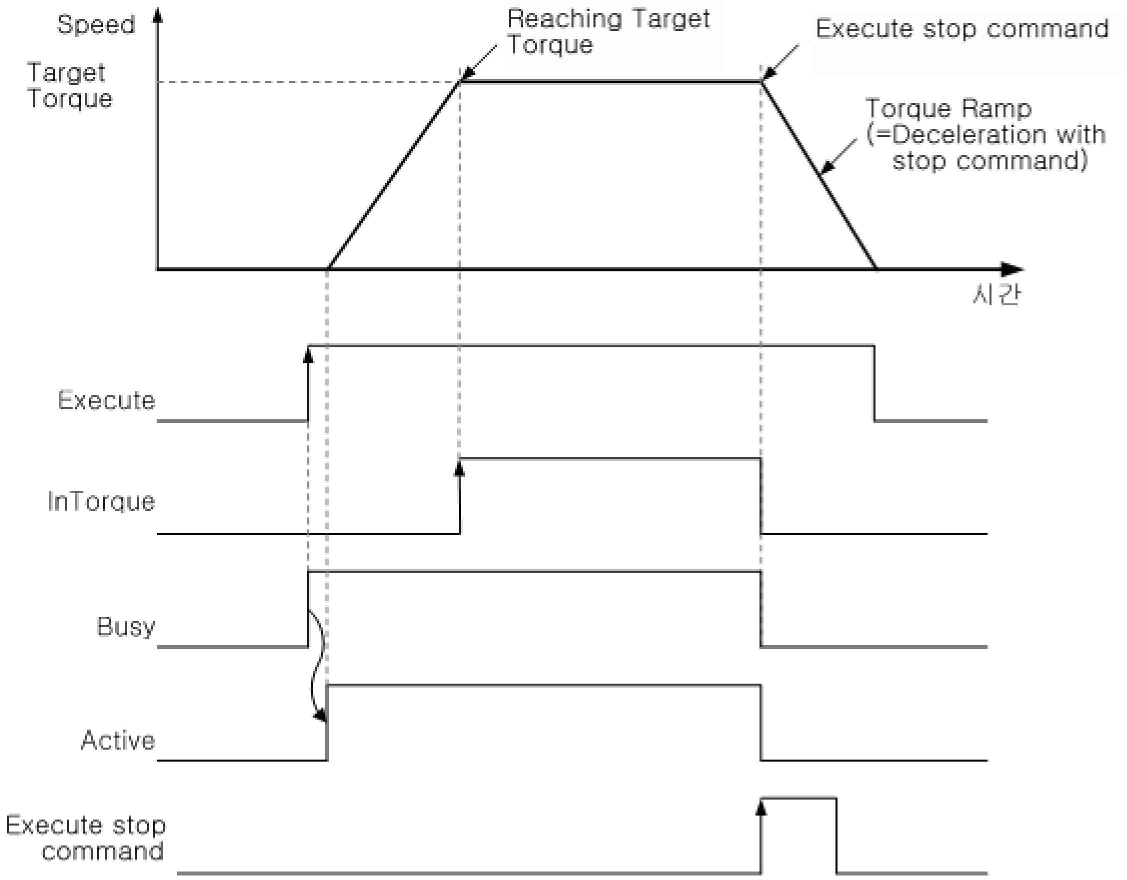
(i) Torque control which is currently being executed is stopped with halt (MC_Halt) or immediate stop (MC_Stop) motion function block.

(2) Related Motion Function Block

Name	Description	Operation condition
MC_TorqueControl	Torque Control	Edge



(3) Operation timing



8.2.4 Specified Velocity Operation after Position Operation

Speed control of the axis specified after being executed by motion function block (「Specified speed operation after relative position operation (MC_MoveContinuousRelative)」 and 「Specified speed operation after absolute position operation (MC_MoveContinuousAbsolute)」) is carried out after the execution of position control that ends with end rate specified from starting position (current stop position) to target position (position of point to move) at the rate specified in end velocity (EndVelocity) if there are no pending commands.

(1) Features of control

(a) Position control that ends with end rate specified from staring position to target position is carried out.

(b) Position control is executed based on position (the origin position) specified in the homing.

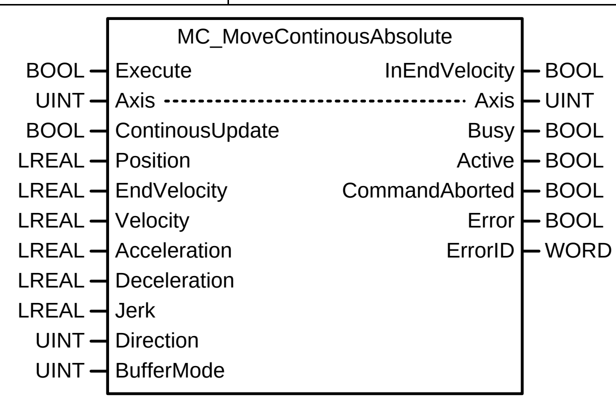
(c) In case of 「Specified speed operation after the absolute position operation (MC_MoveContinuousAbsolute)」, the direction to operate is specified in direction input. It is valid only if operation parameter「Infinite running repetition」setting is '1: Enable'.

- Setting value: 0-Not specified, 1–Forward direction, 2–Shortest distance direction, 3–Reverse direction, 4–Current direction

(d) The end rate is reached after the completion of position control operation to target position, InEndVelocity output of function block is On. If there is a pending command, the pending command is executed after InEndVelocity output is On.

(2) Related Motion Function Block

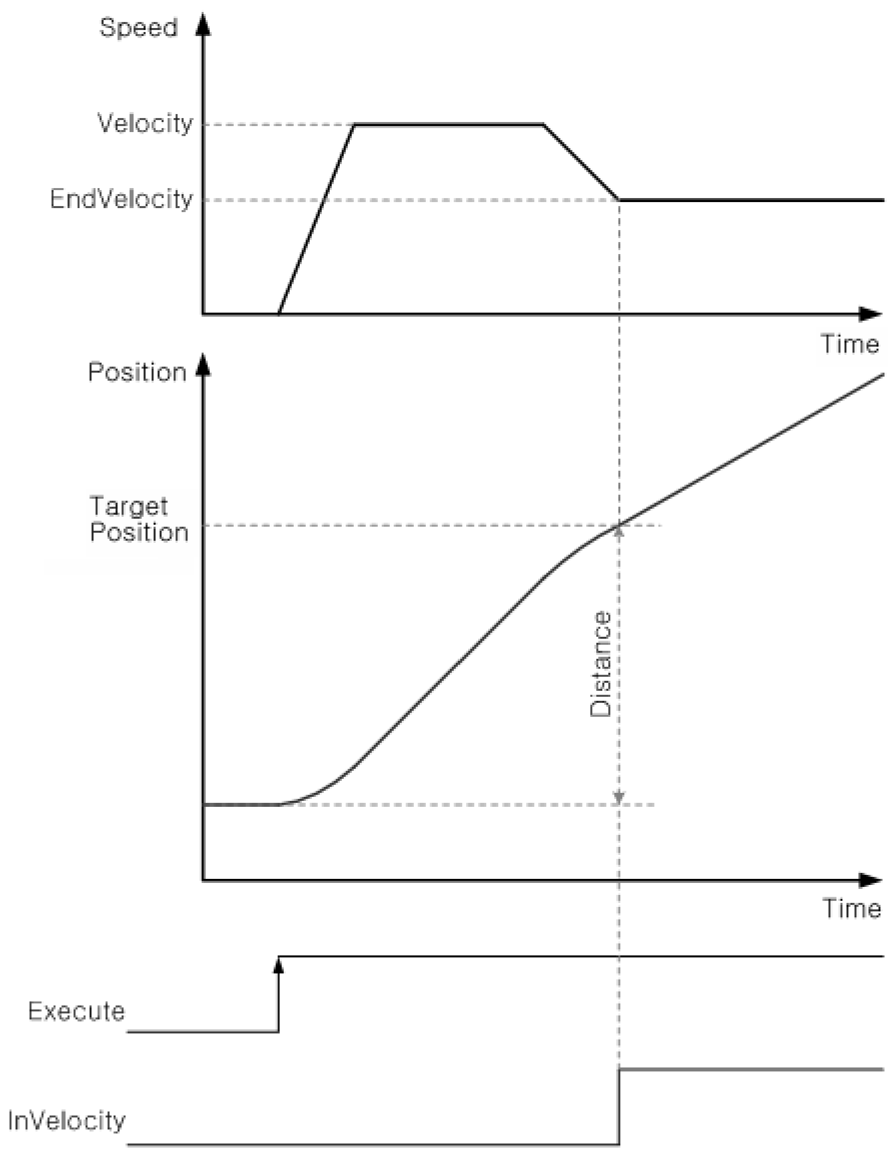
Name	Description	Operation condition
MC_ MoveContinuousAbsolute	Specified velocity operation after Absolute position operation	Edge



Name	Description	Operation condition
MC_ MoveContinuousRelative	Specified velocity operation after Relative position operation	Edge

```
                 MC_MoveContinousRelative
  BOOL — Execute                   InEndVelocity — BOOL
  UINT — Axis ----------------------------- Axis — UINT
  BOOL — ContinousUpdate                    Busy — BOOL
 LREAL — Distance                         Active — BOOL
 LREAL — EndVelocity            CommandAborted — BOOL
 LREAL — Velocity                          Error — BOOL
 LREAL — Acceleration                    ErrorID — WORD
 LREAL — Deceleration
 LREAL — Jerk
  UINT — BufferMode
```

(3) Operation timing



8.2.5 Switching Control

In motion control module, switching control means real-time control switch between position control / velocity control / torque control.

In case the control mode that is currently being executed (position control, velocity control, torque control) are intended to change to a different control mode immediately, BufferMode of commands is to be set to Aborting, and relevant motion function block is to be executed.

(1) Position-Speed switching control

When specified speed operation (MC_MoveVelocity) is executed in the axis in absolute/relative position operation, the position control is switched to velocity control. The velocity at the time of being changed to velocity control is operated continuously from the velocity operated with the previous position control to the target velocity of the currentvelocity control. The next operation can be continued by conducting halt (MC_Halt) during operation with velocity control, performing operation stop with immediate stop (MC_Stop) motion function block or executing other motion function block.

(2) Speed-position switching control

When absolute/relative/additive position control (MC_MoveAbsolute, MC_MoveRelative, MC_MoveAdditive) motion function block is executed in the axis in specified speed operation during velocity control, the velocity control is switched to position control. The velocity at the time of being changed to position control is operated continuously from the velocity operated with the previous velocity control to the target velocity of the current position control. The next operation can be continued by conducting halt (MC_Halt) during operation with position control, performing operation stop with immediate stop (MC_Stop) motion function block or executing other motion function block.

(3) Position-torque switching control

When torque control (MC_TorqueControl) motion function block is executed in the axis in absolute/relative position operation during position control, the position control is switched to torque control. The torque at the time of being changed to torque control is operated continuously from the current torque value operated with the previous position control to the target torque of the torque control. The next operation can be continued by conducting halt (MC_Halt) during operation with torque control, performing operation stop with immediate stop (MC_Stop) motion function block or executing other motion function block.

(4) Torque-position switching control

When absolute/relative/additive position control (MC_MoveAbsolute, MC_MoveRelative, MC_MoveAdditive) motion function block is executed in the axis in specified torque control operation during torque control, the torque control is switched to position control. When switching to position control, the torque value is reduced to 0 to continue operating position control after stopping. The next operation can be continued by conducting halt (MC_Halt) during operation with position control, performing operation stop with immediate stop (MC_Stop) motion function block or executing other motion function block.

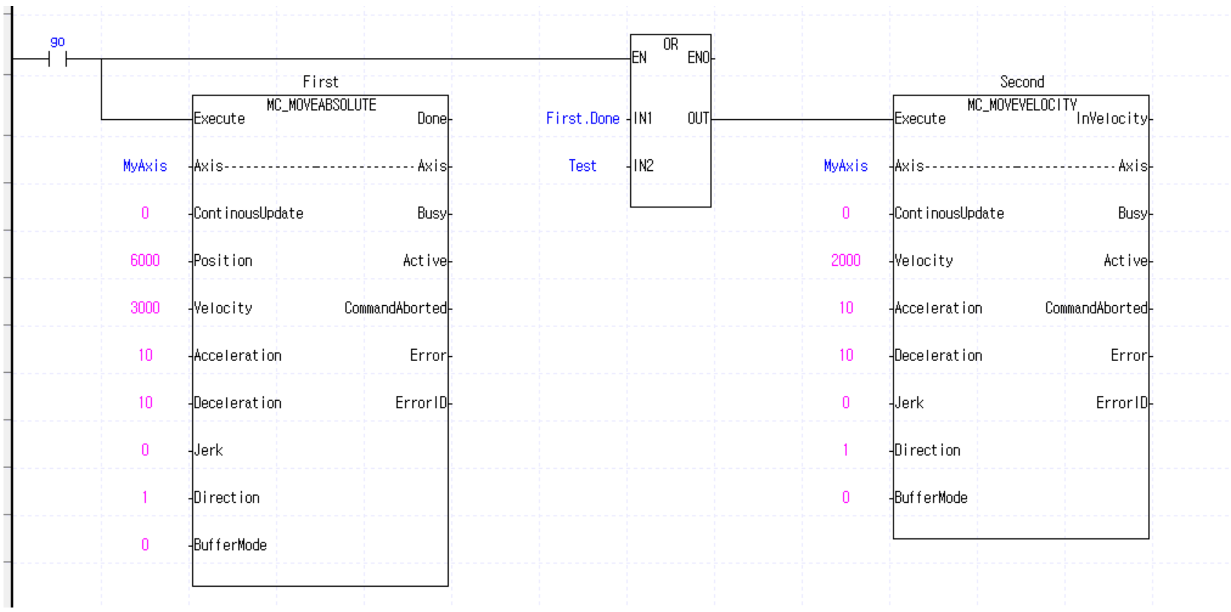
(5) Velocity –torque switching control

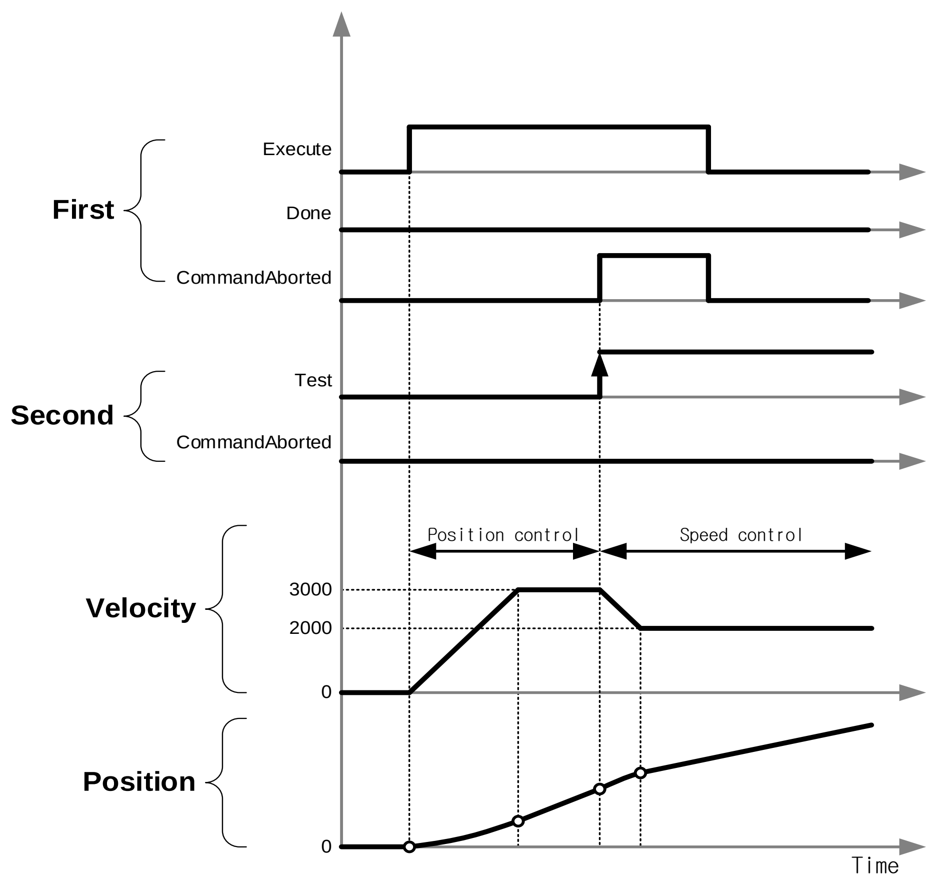
When torque control (MC_TorqueControl) motion function block is executed in the axis in specified speed operation during velocity control, the velocity control is switched to torque control. The torque at the time of being changed to torque control is operated continuously from the current torque value operated with the previous velocity control to the target torque of the torque control. The next operation can be continued by conducting halt (MC_Halt) during operation with torque control, performing operation stop with immediate stop (MC_Stop) motion function block or executing other motion function block.

(6) Torque- velocity switching control

When the specified speed operation (MC_MoveVelocity) motion function block is executed while the axis in torque control operation is performing torque control, torque control is converted to speed control. When switching to speed control, the torque value is reduced to 0 to continue operating speed control after stopping. The next operation can be continued by conducting halt (MC_Halt) during operation with velocity control, performing operation stop with immediate stop (MC_Stop) motion function block or executing other motion function block.

(7) Example of using switching control





8.2.6 Axis Group Control

Axis group control is a function to control the trajectory of moving objects by setting involved multiple axes into one axis group. For axis group control, axis group is to be set.

Axis group operation includes linear interpolation, circular interpolation and helical interpolation.

As for coordinate system in which axis group control is operated, only Cartesian coordinate system is supported

(1) Axis group settings

For axis group control, axis group should be set and enabled prior to the execution of operation.

Configuration axis can be specified, and axis group is set using XG5000. In addition, the use of motion function block makes it possible to add axes to axis group or remove them from it.

When axis group is configured, axis group operation can be executed after enabling the axis group.

All axis groups are in the Disabled state and the _AGxx_Disabled flag is set to On before the axis groups are activated by the group enable command (MC_GroupEnable).

(a) Adds one axis to the group

It means adding an axis to the axis group. The configuration axis specified into IdentlnGroup is added to the axis group specified in AxesGroup input. It can be executed only in case where the axis group is in group disablement (GroupDisabled) and group standby (GroupStandBy) state.

Name	Description	Operation condition
MC_AddAxisToGroup	Adds one axis to the group	Edge

```
                 MC_AddAxisToGroup
BOOL ─ Execute                          Done ─ BOOL
UINT ─ AxesGroup ---------------- AxesGroup ─ UINT
UINT ─ Axis ------------------------------ Axis ─ UINT
UINT ─ IdentInGroup                     Busy ─ BOOL
                                        Error ─ BOOL
                                      ErrorID ─ WORD
```

(b) Removes one axis from the group

It means removing an axis from the axis group. The configuration axis specified into IdentlnGroup is removed from the axis group specified in AxesGroup input.

It can be executed only in case where the axis group is in group disablement (GroupDisabled) and group standby (GroupStandBy) state.

If there are no axes remaining in the axis group by executing the 'Exclude Group Axis' command while the axis group is active, the relevant axis group is changed to the disabled state, and the _AGxx_Disabled flag is set to On.

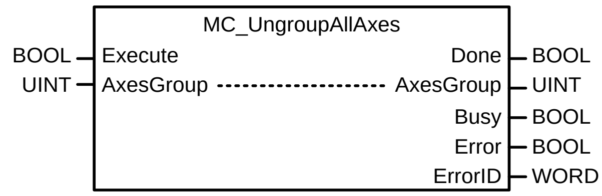
Name	Description	Operation condition
MC_RemoveAxisToGroup	Removes one axis from the group	Edge

```
               MC_RemoveAxisFromGroup
BOOL ─ Execute                          Done ─ BOOL
UINT ─ AxesGroup ---------------- AxesGroup ─ UINT
UINT ─ IdentInGroup                     Busy ─ BOOL
                                        Error ─ BOOL
                                      ErrorID ─ WORD
```

(c) Removes all axes from the group

It means removing all axes from the axis group.

Name	Description	Operation condition
MC_UngroupAllAxes	Removes all axes from the group	Edge



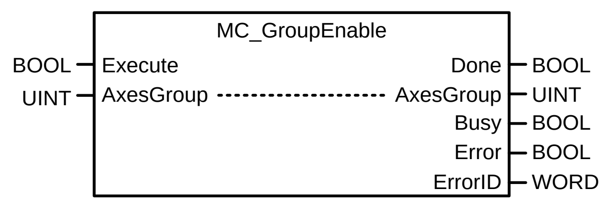
(d) Group Enable

It changes the status to enabled state in which axis group command can be executed.

The axis group cannot be enabled in the following cases.

- In case there is no axis group configuration axis, or axes included in the axis group is not connected to network
- In case the configuration axis of the axis group to be enabled belongs to other enabled axis group
- In case there is an axis in operation among configuration axes in the axis group
- In case the 'unit' of configuration axes in the axis group is not the same
-

Name	Description	Operation condition
MC_GroupEnable	Group Enable	Edge

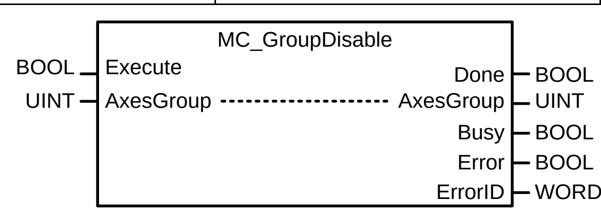


(e) Group Disable

It changes the axis group to be group disabled state.

In case the axis group is in operation, the axis group is changed to be disabled state after the immediate stop.

Name	Description	Operation condition
MC_GroupDisable	Group Disable	Edge



8.2.7 Linear Interpolation Control

Interpolation of multiple axes from staring point (current stop position) to target position is performed with linear trajectory by using relevant axes set in the axis group. Linear interpolation can be performed up to 10 axes.

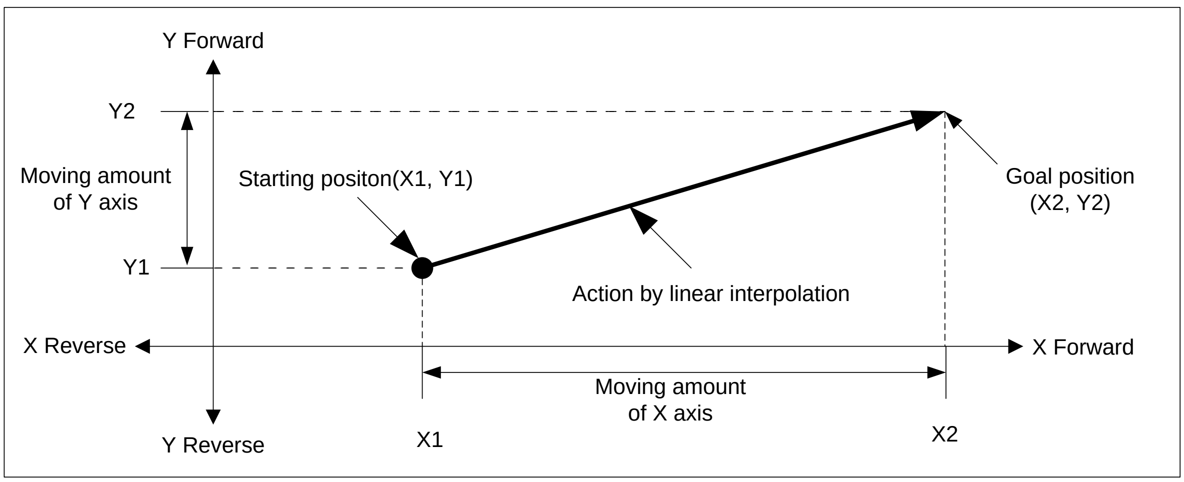
(1) Linear interpolation control with absolute coordinates (「Absolute positioning linear interpolation operation(MC_MoveLinearAbsolute)」)

(a) Executes linear interpolation from starting position to the target position designated on positioning data. Positioning control is carried out based on the position specified from homing.

(b) The direction of movement depends on the starting position and the target position for each axis.

▪ Starting position < Goal position: Positioning operation in forward direction

▪ Starting position > Goal position Positioning operation in reverse direction



(c) To stop the current interpolation control, use MC_GroupHalt or MC_GroupStop motion function block.

(d) The speed value set in absolute position liner interpolation operation (MC_MoveLinearAbsolute) motion function block means synthesis rate of axes that make up the axis group.

The operation speed of each configuration axis is calculated as follows.

$$Interpolatinspeed\ (F) = Operationspeedsetinpositiondata$$

$$Interpolatingmovingamount(S) = \sqrt{S_1^{\ 2} + S_2^{\ 2} + \ldots + S_{10}^2}$$

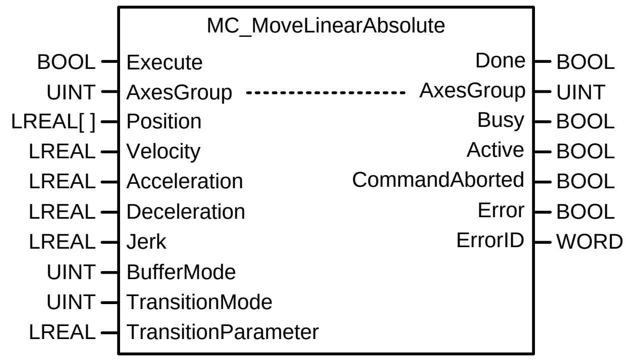
$$Axis1speed(V_1) = Interpolatingspeed\ (F) \times \frac{Main\ axismovingamount(S_1)}{Interpolatingmovingamount\ (S)}$$

$$Axis2\ \text{speed}\ (V_2) = Interpolatingspeed\ (F) \times \frac{Sub - axis1movingamount\ (S_2)}{Interpolatingmovingamount(S)}$$

$$Axis10\ \text{speed}\ (V_{10}) = Interpolatingspeed\ (F) \times \frac{Sub - axis2\ movingamount\ (S_{10})}{Interpolatingmovingamount\ (S)}$$

(e) Relevant motion function block

Name	Description	Operation condition
MC_MoveLinearAbsolute	Absolute positioning linear interpolation operation	Edge



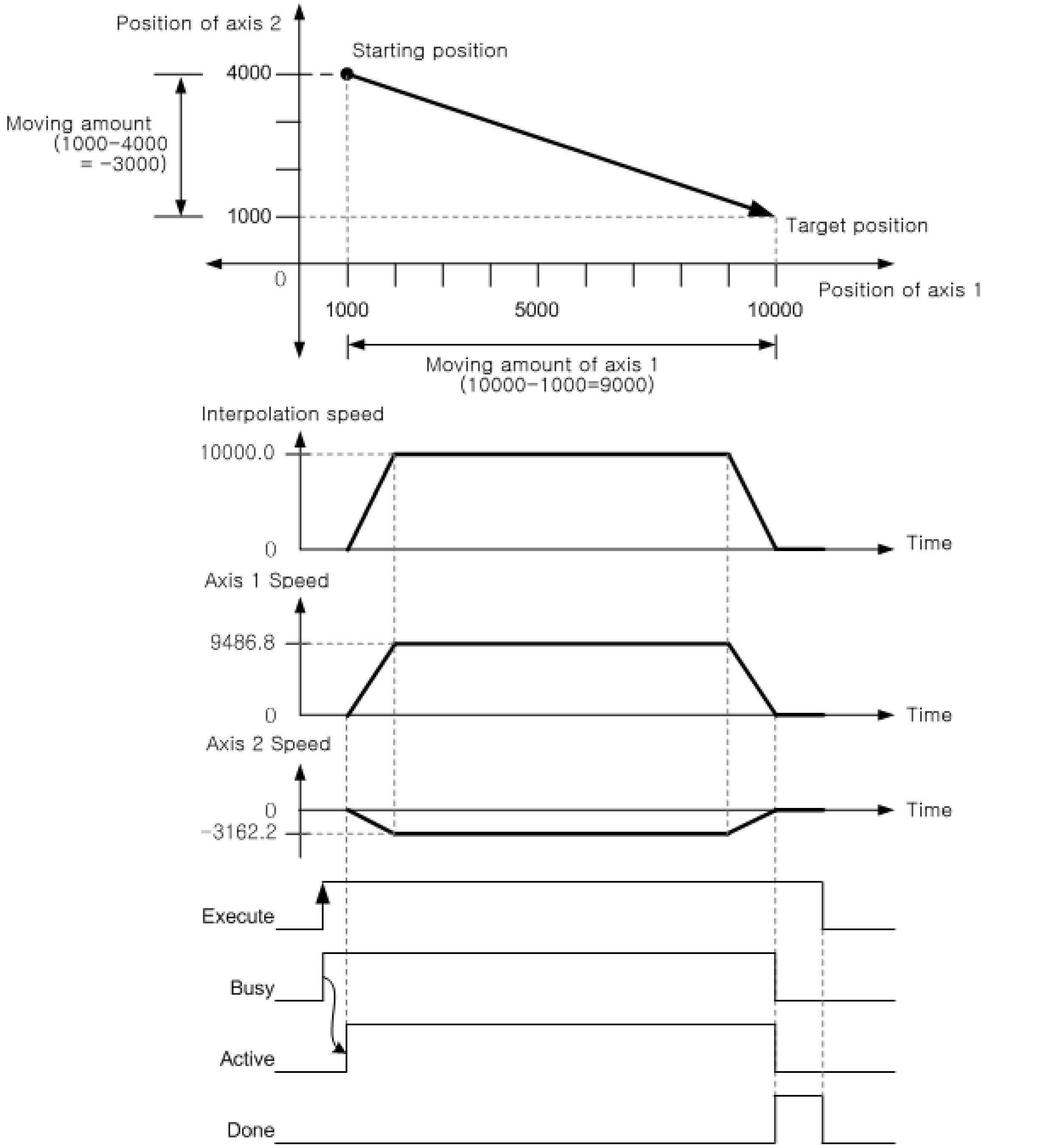
(f) Restrictions

Linear interpolation by absolute coordinate system cannot be executed in the following cases.

- In case there is an axis which is in the origin indetermination state among configuration axes (error code: 0x2090)
- In case the operation speed of configuration axis exceeds the speed limit of each axis (error code: 0x2091)
- In case there is an axis in infinite running repetition operation among configuration axes (error code: 0x2094)

(g) Operation Timing

- Start position:(1000.0, 4000.0)
- Target position:(10000.0, 1000.0)
- Target speed: 10000.0



※ Velocity of each configuration axis is approximate estimate.

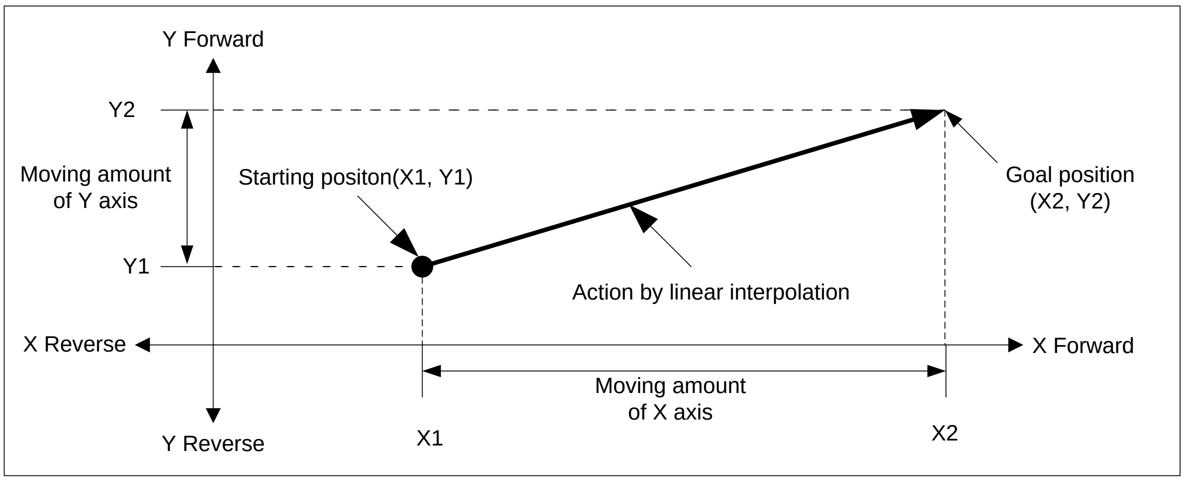
(2) Linear interpolation control with relative coordinates (「Relative positioning interpolation operation (MC_MoveLinearRelative)」)

(a) Linear interpolation is executed from starting position to movement direction targeted by each axis and position t hat includes movement direction. Positioning control is on basis of the current stop position.

(b) Moving direction depends on the sign of the goal position (Moving amount)

▪ When the sign of movement distance is positive (+ or no sign): Positioning operation in forward direction (starting position increase direction)

▪ When the sign of movement distance is negative (-):Positioning operation in reverse direction (starting position decrease direction)



(c) To stop the current interpolation control, use MC_GroupHalt or MC_GroupStop motion function block.

(d) The speed value set in relative position liner interpolation operation (MC_MoveLinearRelative) motion function blo ck means interpolation speed.

The operation speed of each configuration axis is calculated as follows.

$$Interpolation\ speed(F) = Target\ speed\ set\ in\ Velocit$$

$$Interpolation\ movement(S) = \sqrt{S1^2 + S2^2 + S3^2 \ldots\ldots + S10^2}$$

$$\text{Composition axis1 speed(V1)} = Interpolation\ speed(F)\ x\ \frac{\text{Composition axis1 movement amout(S1)}}{Interpolation\ movement\ amout(S)}$$

$$\text{Composition axis2 speed(V2)} = Interpolation\ speed(F)\ x\ \frac{\text{Composition axis2 movement amout(S2)}}{Interpolation\ movement\ amout(S)}$$

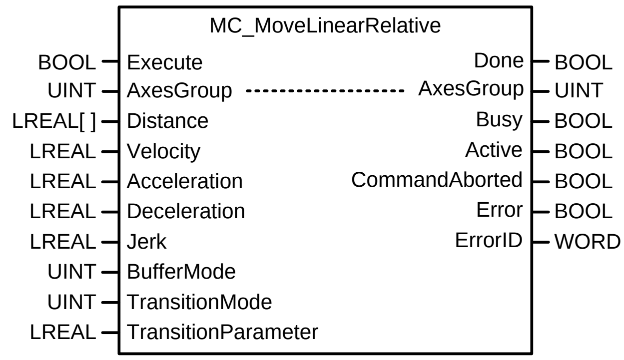
$$\text{Composition axis3 speed(V3)} = Interpolation\ speed(F)\ x\ \frac{\text{Composition axis3 movement amout(S3)}}{Interpolation\ movement\ amout(S)}$$

:

$$\text{Composition axis10 speed(V10)} = Interpolation\ speed(F)\ x\ \frac{\text{Composition axis10 movement amout(S10)}}{Interpolation\ movement\ amout(S)}$$

Relevant motion function block

Name	Description	Operation condition
MC_MoveLinearRelative	Relative positioning linear interpolation operation	Edge



(e) Restrictions

Linear interpolation by relative coordinate system cannot be executed in the flowing cases.

- In case there is an axis in infinite running repetition operation among configuration axes (error code: 0x2094)
- In case the operation speed of configuration axis exceeds the speed limit of each axis (error code: 0x2091)

8.2.8 Circular Interpolation Control

Interpolation operation is performed along the trajectory of the circle in the direction of axis progress set by using two axes set in the axis group.

There are three kinds of methods for circular interpolation such as midpoint method that passes through the position specified in auxiliary point, center point method that considers the position specified in auxiliary point as center point and radius method that takes the value specified in auxiliary point as the radius of an arc depending on 'CircMode' settings and auxiliary points.

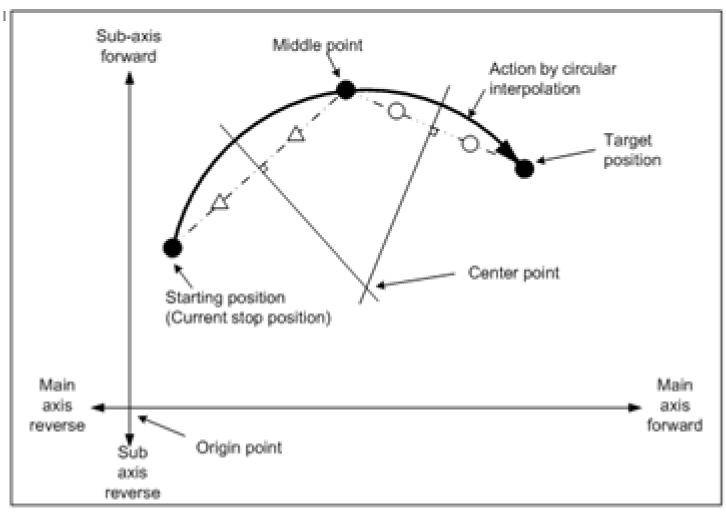
To stop the current interpolation control use MC_GroupHalt or MC_GroupStop motion function block.

(1) Circular interpolation with middle point designation form.

(a) Circular interpolation is executed from starting position to target position through midpoint position set in auxiliary point.

(b) To be made path of circular interpolation with start position, midpoint and a crossing which is perpendicular divi de equally position of midpoint and target position.

(c) Movement direction is decided automatically depends on set target position and auxiliary point of circular interpol ation.

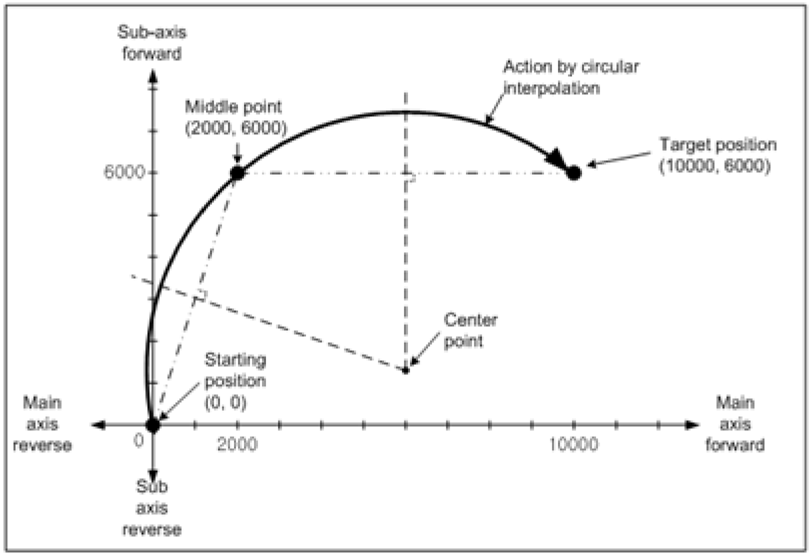


(d) Restrictions

Circular interpolation control using mid-point specification method cannot be performed in case of the following errors.

- In case there is an axis which is in the origin undetermined state among configuration axes at the time of absolute coordinate circular interpolation operation (error code: 0x20A0)
- In case the midpoint specified as auxiliary point is the same as the starting position or target position (error code: 0x20A4)
- In case starting position is the same as the target position (error code: 0x20A5)
- In case the calculated radius of the arc exceeds 2147483647pls (error code: 0x20A6)
- In case start position, auxiliary point position and target position are in a straight line (error code: 0x20A7)
- In case there is an axis in infinite running repetition operation among configuration axes (error code: 0x20A8)
- In case the number of configuration axes in the axis group is four (error code: 0x20A9)
- In case axis group configuration settings are not set in order (error code: 0x20AA)

(e) Operating Pattern
- Start position:(0.0, 0.0)
- Target position:(10000.0, 6000.0)
- Midpoint position:(2000.0, 6000.0)
- CircMode: Midpoint(0)
- Direction(PathChoice): - (Ignored in middle point method)

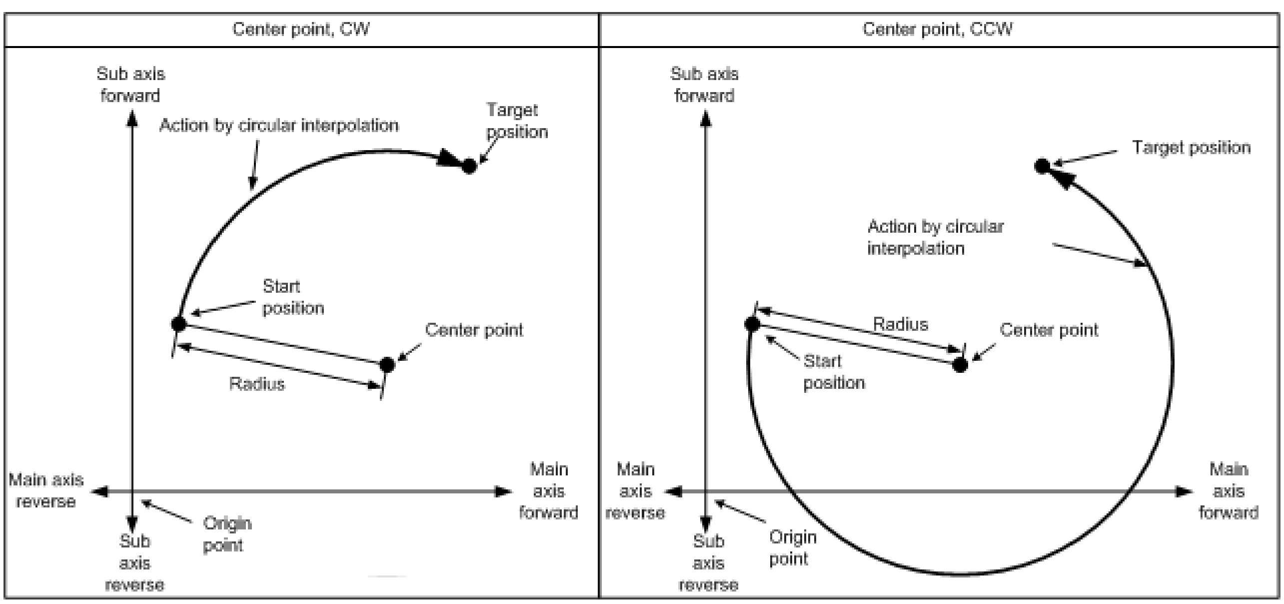


(2) Circular interpolation with center point designation form

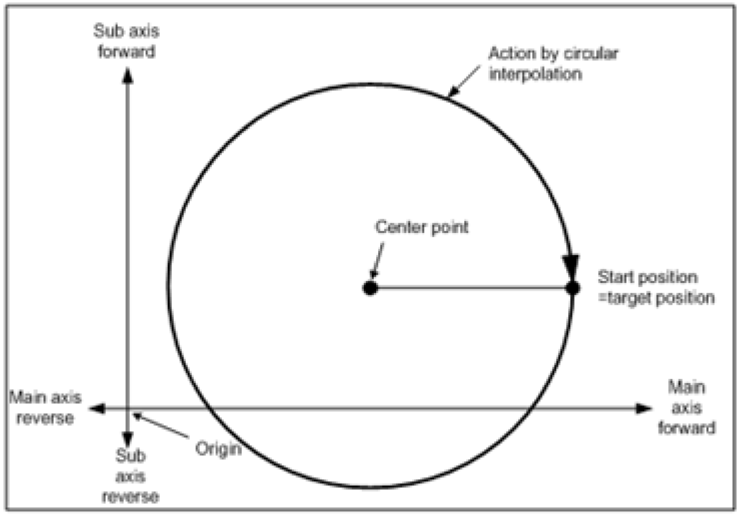
(a) Circular interpolation is performed by starting at the start position, and reaching the target position in a circular trajectory of which the diameter is the distance to the designated center point.

(b) Movement direction is determined by the direction set in “PathChoice” of absolute position circular interpolation operation (MC_MoveCircularAbsolute) or relative position circular interpolation operation (MC_MoveCircularRelative) motion function block.

- 0:「CW」 - perform circular interpolation in the clockwise direction from the start position.
- 1:「CCW」- perform circular interpolation in the counter-clockwise direction from the start position.



(c) If target position is same with start position, can progress circular interpolation. And the circle radius is distance from midpoint to starting position (=target position)



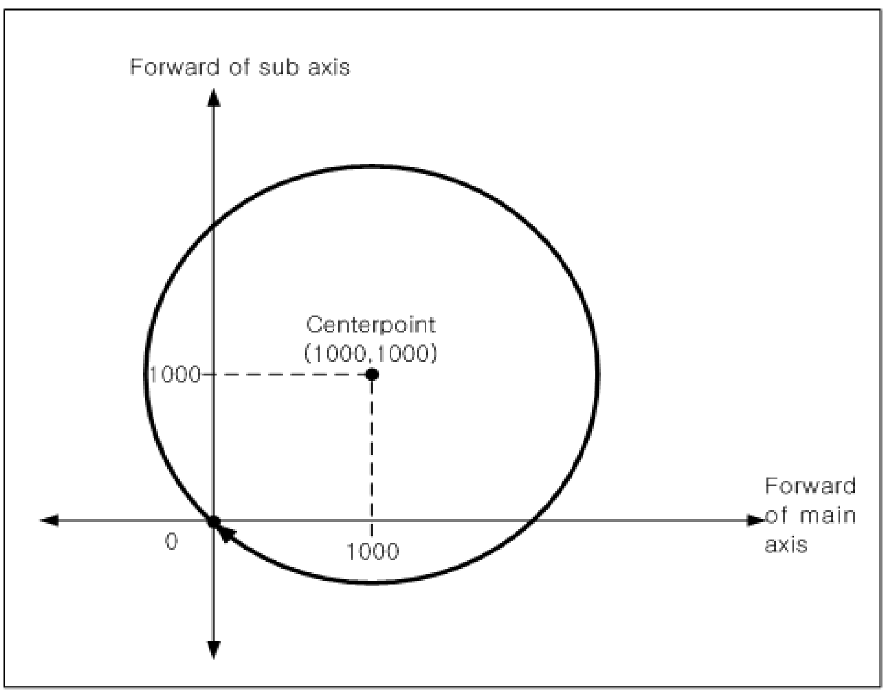
(d) Restrictions

Circular interpolation control using center point specification method cannot be performed in case of the following errors.

- In case there is an axis which is in the origin undetermined state among configuration axes at the time of absolute coordinate circular interpolation operation (error code: 0x20A0)
- In case the midpoint specified as auxiliary point is the same as the starting position or target position (error code: 0x20A4)
- In case the calculated radius of the arc exceeds 2147483647pls (error code: 0x20A6)
- In case start position, auxiliary point position and target position are in a straight line (error code: 0x20A7)
- In case there is an axis in infinite running repetition operation among configuration axes (error code: 0x20A8)
- In case the number of configuration axes in the axis group is four (error code: 0x20A9)
- In case axis group configuration settings are not set in order (error code: 0x20AA)

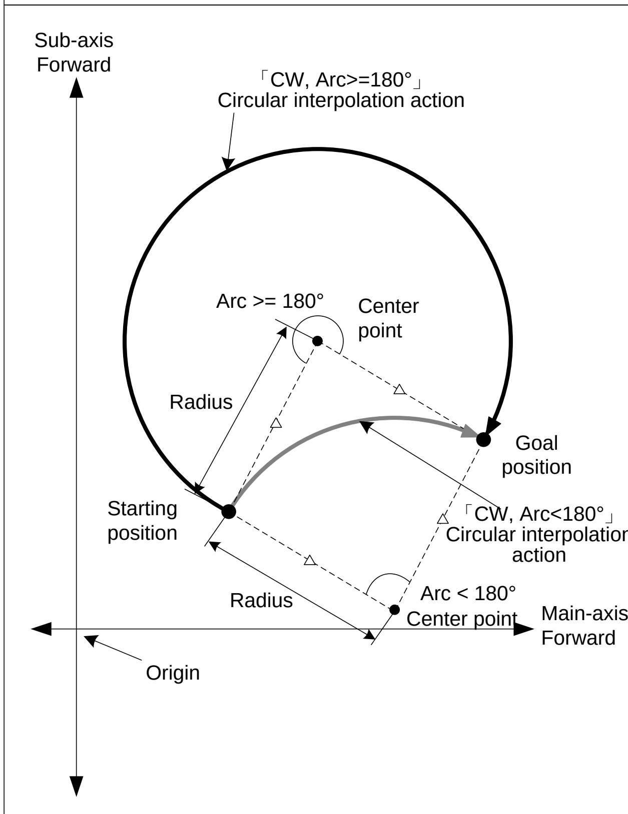
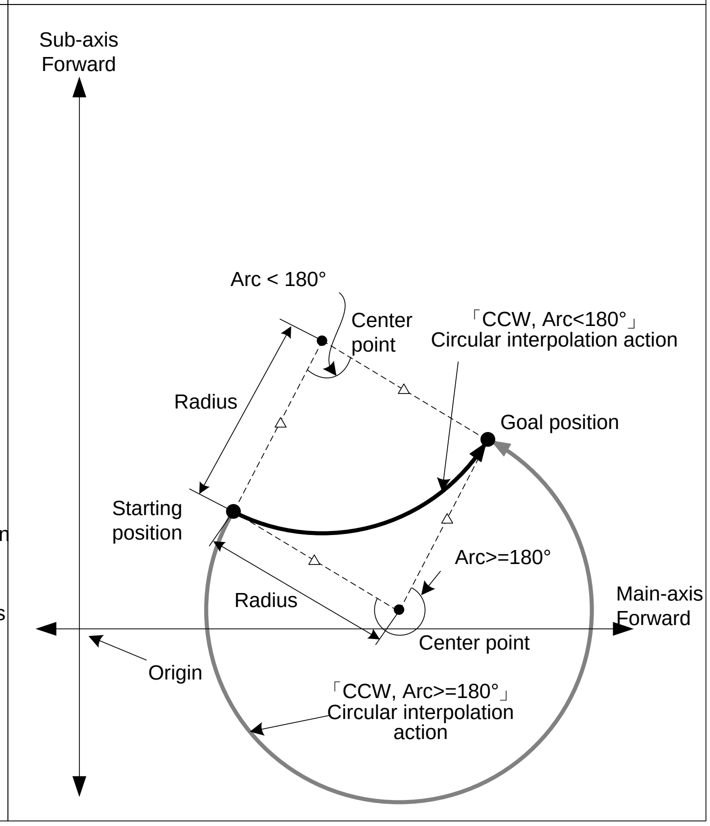
(e) Operating Pattern

- Start position:(0.0, 0.0)
- Target position:(0.0, 0.0)
- Auxiliary position:(1000.0, 1000.0)
- CircMode: Center Point(1)
- PathChoice: CW(0)



(3) Circular interpolation with radius designation form

(a) Circular interpolation is performed from starting position to target position along the trajectory of the arc that take s the value set in circular interpolation auxiliary point. The arc that has center point depending on the sign of r adius ((+): arc angle <180°, (-): arc angle>=180°) is drawn.

CW, Arc<180°　CW, Arc>=180°	CCW, Arc<180°　CCW, Arc>=180°
	

(b) In circular interpolation of radius specification method, the target position cannot be set the same as starting po sition.

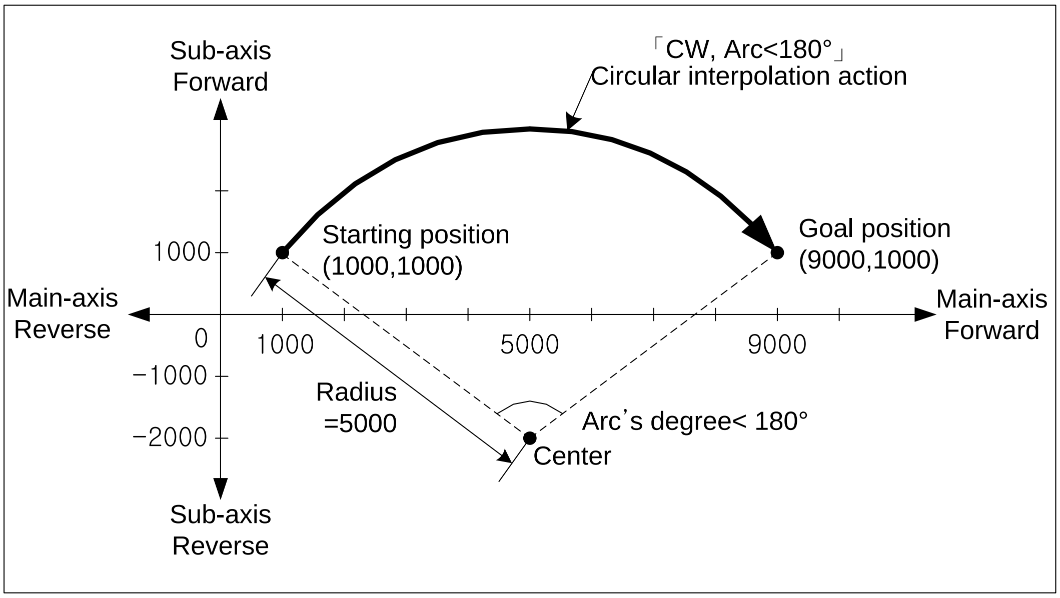
(c) Movement direction and the size of the arc is determined by the sign of auxiliary point and directions (CW, CC W) set in “PathChoice” of absolute position circular interpolation operation (MC_MoveCircularAbsolute) or relative po sition interpolation operation (MC_MoveCircularRelative) motion function block

(d) Restrictions

Circular interpolation control by radius specification method cannot be performed in case of the following errors.

- In case there is an axis which is in the origin undetermined state among configuration axes at the time of absolute coordinate circular interpolation operation (error code: 0x20A0)
- In case starting position is the same as the target position (error code: 0x20A5)
- In case the calculated radius of the arc exceeds 2147483647pls (error code: 0x20A6)
- In case start position, auxiliary point position and target position are in a straight line (error code: 0x20A7)
- In case there is an axis in infinite running repetition operation among configuration axes (error code: 0x20A8)
- In case the number of configuration axes in the axis group is four (error code: 0x20A9)
- In case axis group configuration settings are not set in order (error code: 0x20AA)

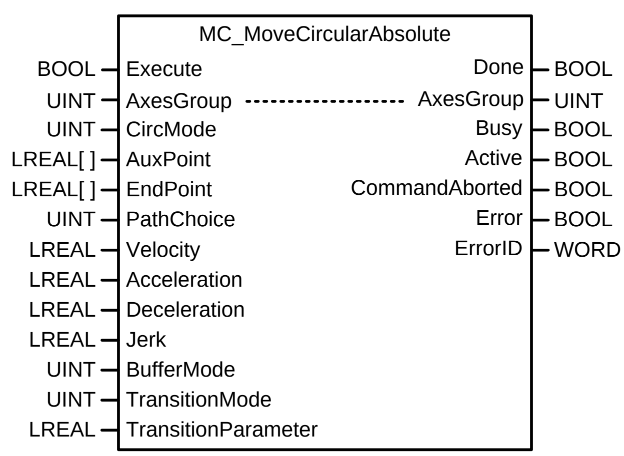
(e) Operating Pattern
- Start position:(1000.0, 1000.0)
- Target position:(9000.0, 1000.0)
- Auxiliary position:(5000.0, 0.0)
- CircMode: Center Point(2)
- PathChoice: CW(0)



(4) Relevant motion function block

(a) Absolute position circular interpolation operation

Name	Description	Operation condition
MC_MoveCircularAbsolute	Absolute position circular interpolation operation	Edge



(b) Relative position circular interpolation operation

Name	Description	Operation condition
MC_MoveCircularRelative	Relative position circular interpolation operation	Edge

MC_MoveCircularRelative

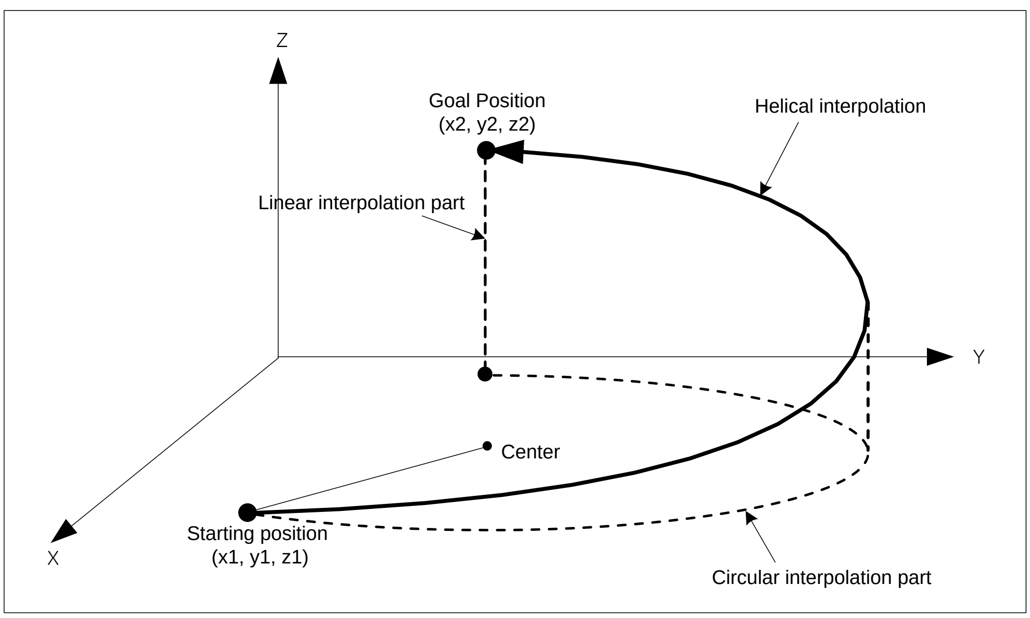
Type	Input	Output	Type
BOOL	Execute	Done	BOOL
UINT	AxesGroup	AxesGroup	UINT
UINT	CircMode	Busy	BOOL
LREAL[ ]	AuxPoint	Active	BOOL
LREAL[ ]	EndPoint	CommandAborted	BOOL
USINT	PathChoice	Error	BOOL
LREAL	Velocity	ErrorID	WORD
LREAL	Acceleration		
LREAL	Deceleration		
LREAL	Jerk		
UINT	BufferMode		
UINT	TransitionMode		
LREAL	TransitionParameter		

(5) Helical interpolation

(a) Three axes are used in the execution of circular interpolation commands (「Absolute positioning circular interpolation operation (MC_MoveCircularAbsolute)」, 「Relative positioning circular interpolation operation (MC_MoveCircularRelative)」). That is, two axes move the trajectory of the arc depending on circular interpolation settings, and one axis performs linear interpolation in synchronization with circular interpolation motion.

(b) Linear axis is the third axis of the circular interpolation axis group.

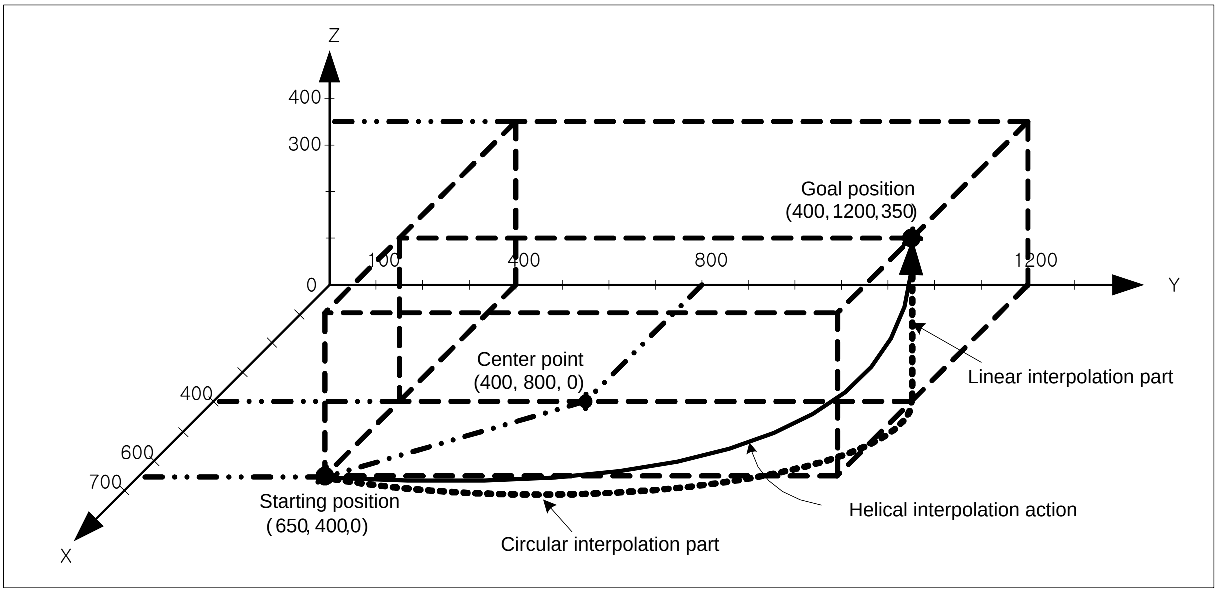
(c) For the execution of helical interpolation, the axis group of circular interpolation command needs to be set to 3-axis, and linear interpolation target position is to be set in the third axis of 'EndPoint'.



(d) Restrictions

The restrictions of helical interpolation are the same as those of circular interpolation according to the set circular interpolation modes.

(e) Operating Pattern
- Start position:(650.0, 400.0, 0)
- Target position:(400.0, 1200.350)
- Center Point position:(400.0, 800.0, 0)
- CircMode: Center Point(1)
- PathChoice: CCW(1)



8.2.9 Axis Control Buffer mode

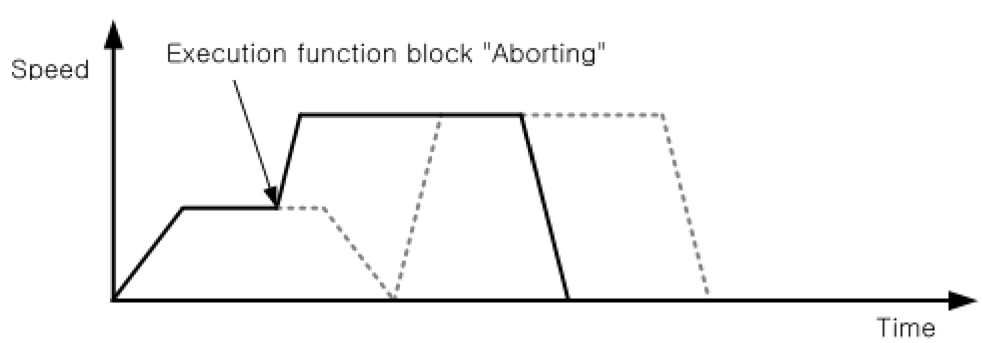
Cancellation of the existing axis motions and continued or continuous operation of them can be carried out by executing other motion function block while the axis is in operation. The motions are specified by entering buffer mode (BufferMode) in motion function block. The maximum number of commands that can wait for their execution in buffers on axis control is 100. An error (error code: 0x1022) occurs when executing more than 100 command in a buffer mode.
When operating using buffer mode, the status of the command buffer can be checked using the following flags.

Variable	Type	Description
_AXxx_CMDBUF_FULL	BOOL	Axis xx Buffered full of command buffers
_AXxx_CMDBUF_QUEUED	UINT	Axis xx Buffered number of command execution wait
_AXxx_CMDBUF_FREE	UINT	Axis xx Buffered number of executable commands

The values which are available to be set in BufferMode are as below.

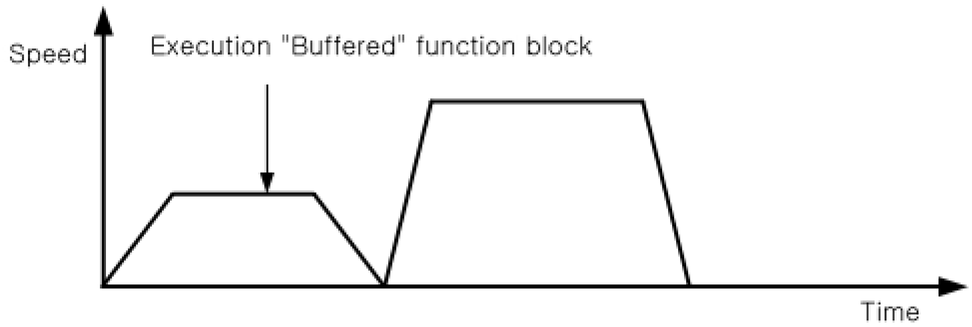
Buffer mode	Description
Aborting	Execute the command immediately. The existing command in operation is interrupted.
Buffered	It executes commands after the completion of the existing command in operation.
BlendingLow	It conducts a combination operation that helps blend into side with lower velocity by comparing the velocity of the existing command and the command to make.
BlendingPrevious	It conducts a combination operation that makes the combination with velocity of the existing commands.
BlendingNext	It conducts a combination operation that makes the combination with velocity of commands to make.
BlendingHigh	It conducts a combination operation that helps blend into side with priority velocity by comparing the velocity of the existing command and the command to make.

(1) Buffer mode“Aborting”
It aborts the existing commands in execution immediately and executes the next command. CommandAborted output of the existing motion function blocks is On.



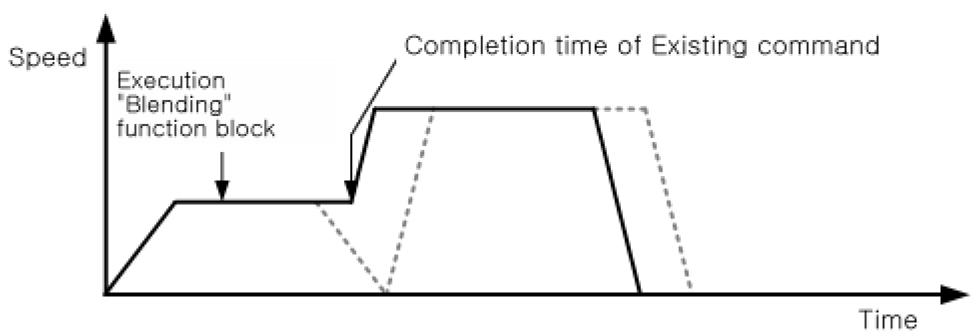
(2) Buffer mode“Buffered”

It execute the next command after the completion of the existing commands in execution (Done output is On).



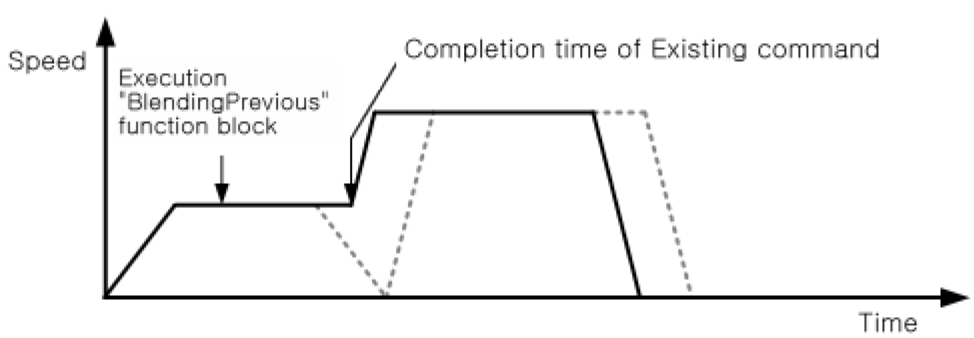
(3) Buffer mode“BlendingLow”

It combines operation so that operation can be made at lower velocity in a comparison between the target velocity of the existing commands in execution at the time of command completion and that of buffered command.



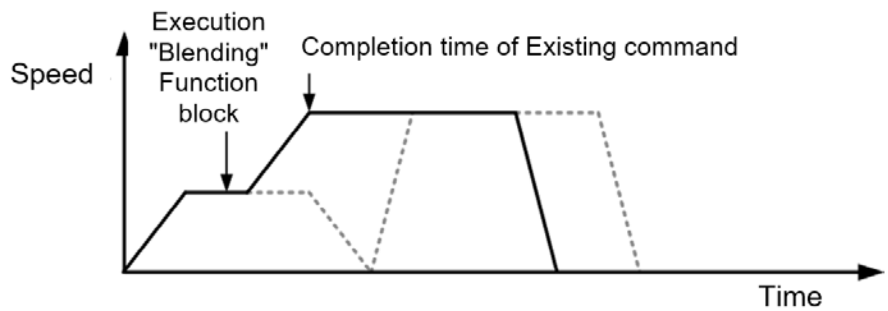
(4) Buffer mode“BlendingPrevious”

It executes the next command after acceleration/deceleration of the velocity to the target velocity of the next command buffered after maintaining the velocity of commands in execution at the point of time when the exiting commands are competed.



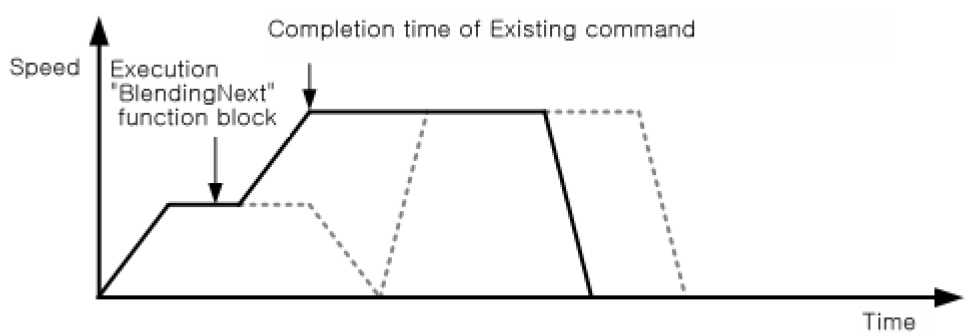
(5) Buffer mode“BlendingNext”

It executes the next command after acceleration/deceleration so that operation can be performed at the target velocity of the next command at the point of time when the existing commands in execution are completed.



(6) Buffer mode“BlendingHigh”

It combines operation so that operation can be made at higher velocity in a comparison between the target velocity of the existing commands in execution at the time of command completion and that of buffered command.



8.2.10 Axis Group Control Buffer Mode and Transition Mode

In axis group control as in speed control, motion commands can be executed continuously by using buffer mode, and the maximum number of runs that can be queued in the buffer is 10. The maximum number of buffers that can wait for execution on axis group control is 100. An error (error code: 0x2022) occurs when executing more than 100 command in a buffer mode. In addition, operation is possible by inserting curve between the two linear trajectories using transition mode.

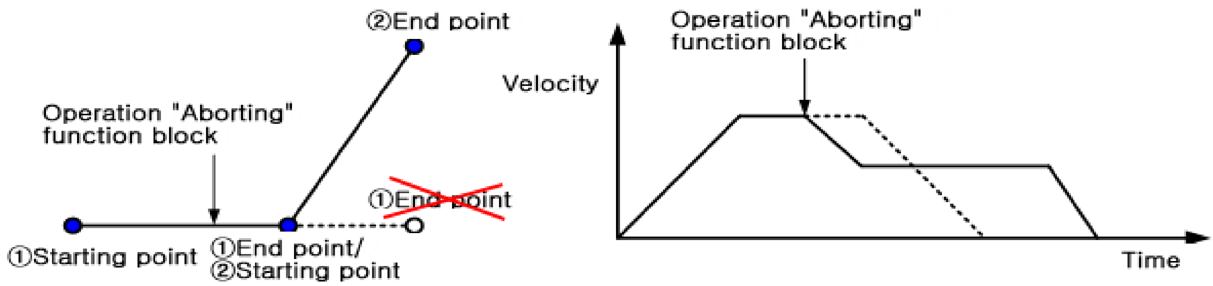
When operating using buffer mode, the status of the command buffer can be checked using the following flags.

Variable	Type	Description
_AGxx_CMDBUF_FULL	BOOL	Axis group xx Buffered full of command buffers
_AGxx_CMDBUF_QUEUED	UINT	Axis group xx Buffered number of command execution wait
_AGxx_CMDBUF_FREE	UINT	Axis group xx Buffered number of executable commands

(1) ‘BufferMode’

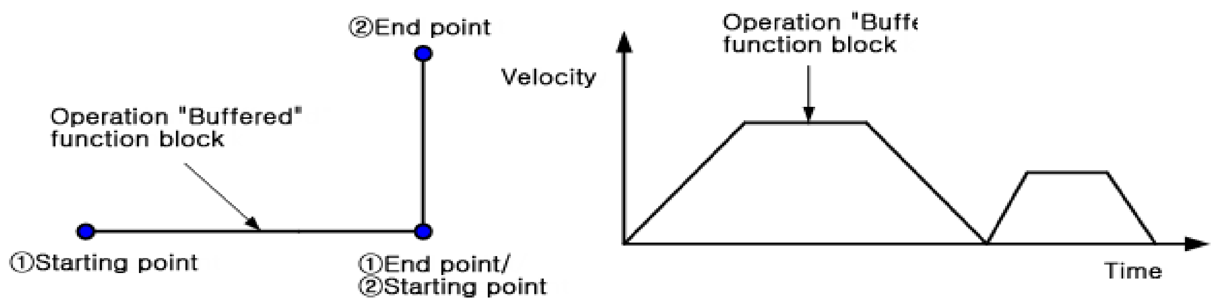
(a) Aborting

It aborts the motion that is currently running, and executes a new motion immediately.



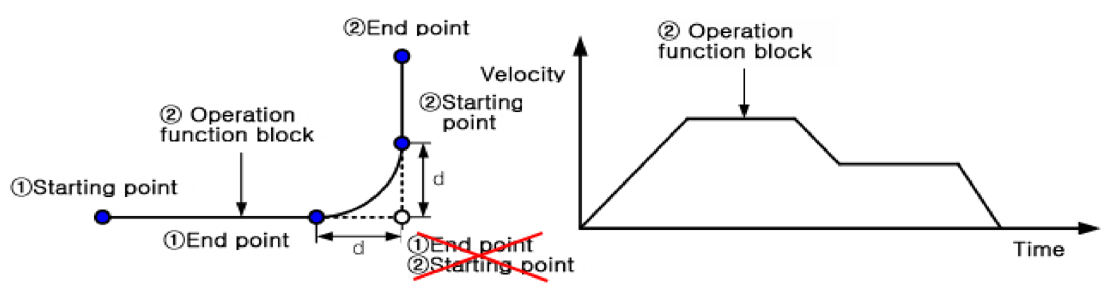
(b) Buffered

It executes the next command after completing motion operation that is being currently executed.
‘TransitionMode’ is not reflected.



(c) Blending

There is no stop between the two operations since the current motion is mixed with the next motion. The velocity may vary depending on blending modes (BlendingLow, BendingPrevious, BlendingNext, BlendingHigh).



※ Motions in case of the BlendingNext

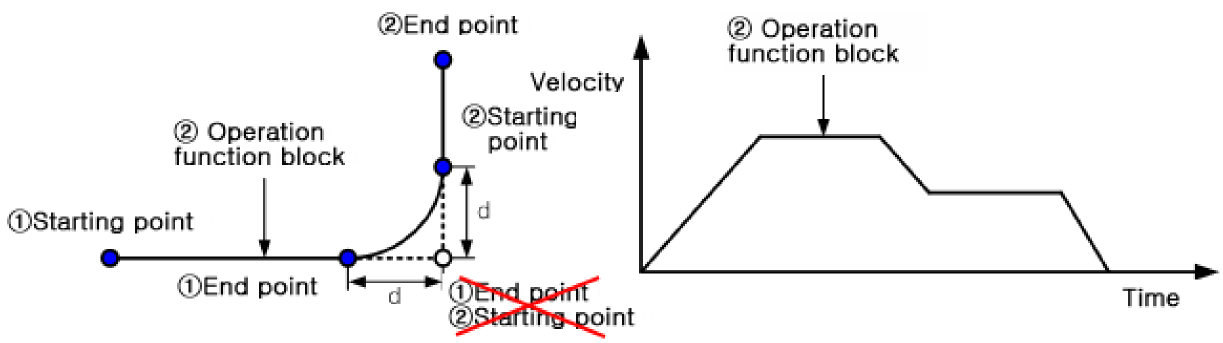
(2) 'TransitionMode'

(a) TMNone

Motion trajectory is not changed, and curve is not inserted between the two operations.
In case buffer mode is Blending in this setting, Buffered mode is operated.
Motions according to the buffer mode are the same as the above Aborting and Buffered.

(b) TMCornerDistance

Ther curve can be inserted by specifying the distance of two motion block corners. The conversion velocity is specified by the BufferMode.

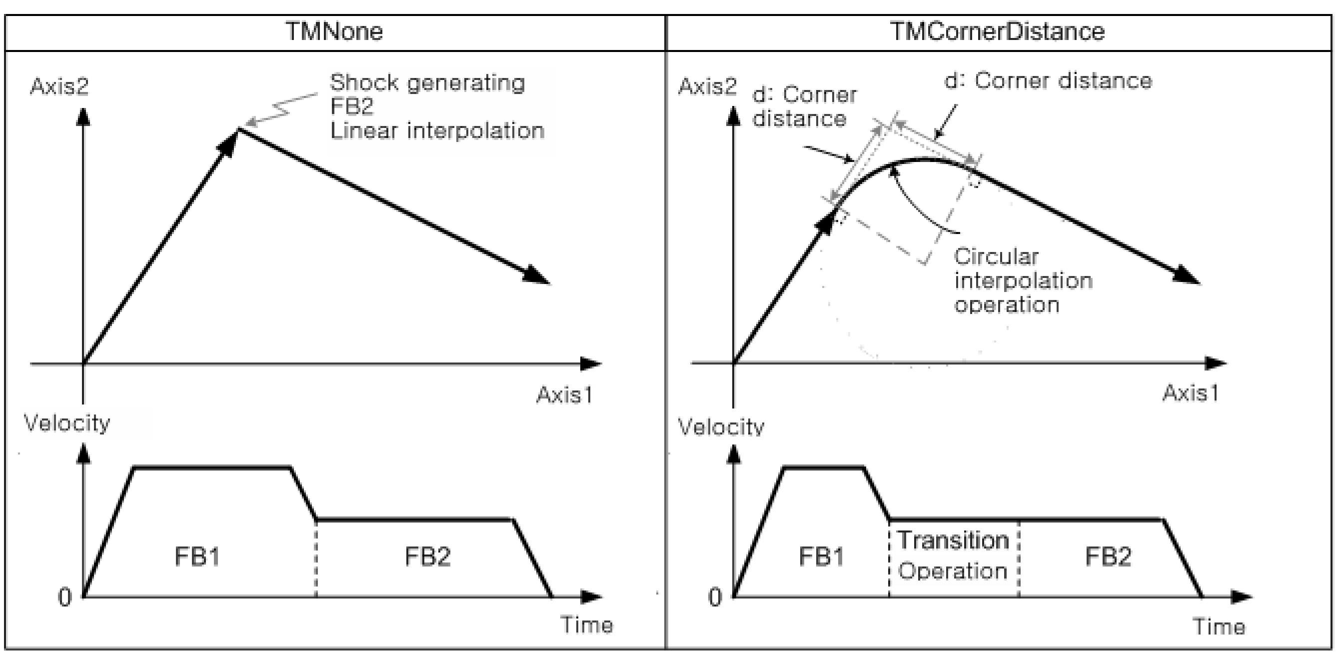


※ Motions in case of the BlendingNext

※ d: Curve insertion distance at the corner

※ In the MC_MoveLinearAbsolute and MC_MoveLinearRelative commands, it operates only when the number of axes in the axis group is two. In other cases, even if it is set to TMCornerDistance, it operates as TMNone and no error occurs.

(c) TransitionMode Compare



8.2.11 Synchronous Control

(1) Gear operation

(a) Gear operation makes speed synchronization of the master axis (or encoder) and the slave axis depending on the set ratio.

(b) Gear operation can be aborted with gear operation cancellation command.

(c) Gear ratio (=velocity synchronization ratio) is calculated as follows.

Gear ratio = Master axis ratio/Slave axis ratio

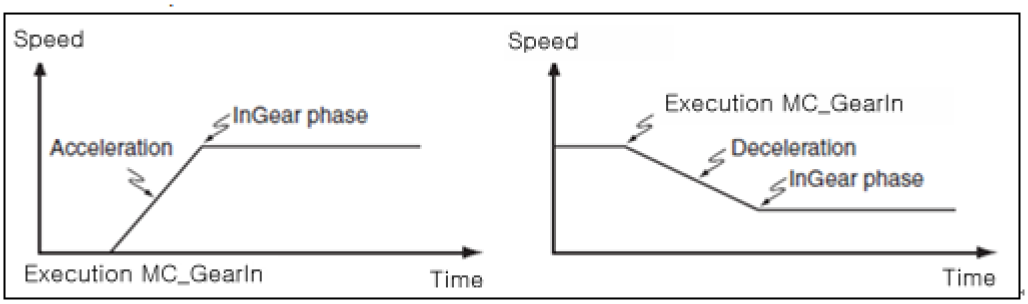
※ The master axis ratio < the slave ratio can also be set.

(d) Rotation direction of the slave axis is based on the forward direction of the master axis. In case gear ratio is positive (>0), rotation is made in forward direction, and that is negative (< 0), in reverse direction.

(e) The final operating velocity of the slave axis is calculated as follows.

Operation speed of the slave axis= operation speed of the master axis X Gear ratio Operation speed of the master axis X main axis ratio/subordinate axis ratio

(f) Acceleration/deceleration from the start of gear operation to target velocity can be set by using Acceleration and Deceleration input.



(g) Relevant motion function block

Name	Description	Operation condition
MC_GearIn	Gear operation	Edge

```
                 MC_GearIn
BOOL  ─ Execute                          InGear ─ BOOL
UINT  ─ Master ------------------------- Master ─ UINT
UINT  ─ Slave  -------------------------  Slave ─ UINT
BOOL  ─ ContinousUpdate                    Busy ─ BOOL
INT   ─ RatioNumerator                   Active ─ BOOL
UINT  ─ RatioDenominator       CommandAborted ─ BOOL
UINT  ─ MasterValueSource                 Error ─ BOOL
LREAL ─ Acceleration                    ErrorID ─ WORD
LREAL ─ Deceleration
LREAL ─ Jerk
UINT  ─ BufferMode
```

Name	Description	Operation condition
LS_VarGearIn	Variable Gearing run	Edge

LS_VarGearIn

Input type	Input	Output	Output type
BOOL	Execute	InGear	BOOL
UDINT	VarOffset	VarOffset	UINT
UINT	Slave	Slave	UINT
BOOL	ContinousUpdate	Busy	BOOL
INT	RatioNumerator	Active	BOOL
UINT	RatioDenominator	CommandAborted	BOOL
UINT	MasterValueSource	Error	BOOL
LREAL	Acceleration	ErrorID	WORD
LREAL	Deceleration		
LREAL	Jerk		
UINT	BufferMode		

Name	Description	Operation condition
MC_GearOut	Gearing disengage	Edge

MC_GearOut

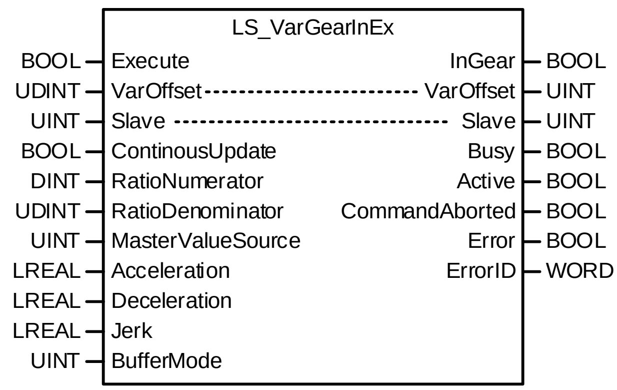
Input type	Input	Output	Output type
BOOL	Execute	Done	BOOL
UINT	Slave	Slave	UINT
		Busy	BOOL
		Error	BOOL
		ErrorID	WORD

Name	Description	Operation condition
MC_GearInEx	Expansion gear operation	Edge

MC_GearInEx

Input type	Input	Output	Output type
BOOL	Execute	InGear	BOOL
UINT	Master	Master	UINT
UINT	Slave	Slave	UINT
BOOL	ContinousUpdate	Busy	BOOL
DINT	RatioNumerator	Active	BOOL
UDINT	RatioDenominator	CommandAborted	BOOL
UINT	MasterValueSource	Error	BOOL
LREAL	Acceleration	ErrorID	WORD
LREAL	Deceleration		
LREAL	Jerk		
UINT	BufferMode		

Name	Description	Operation condition
LS_VarGearInEx	Expansion variable gear operation	Edge

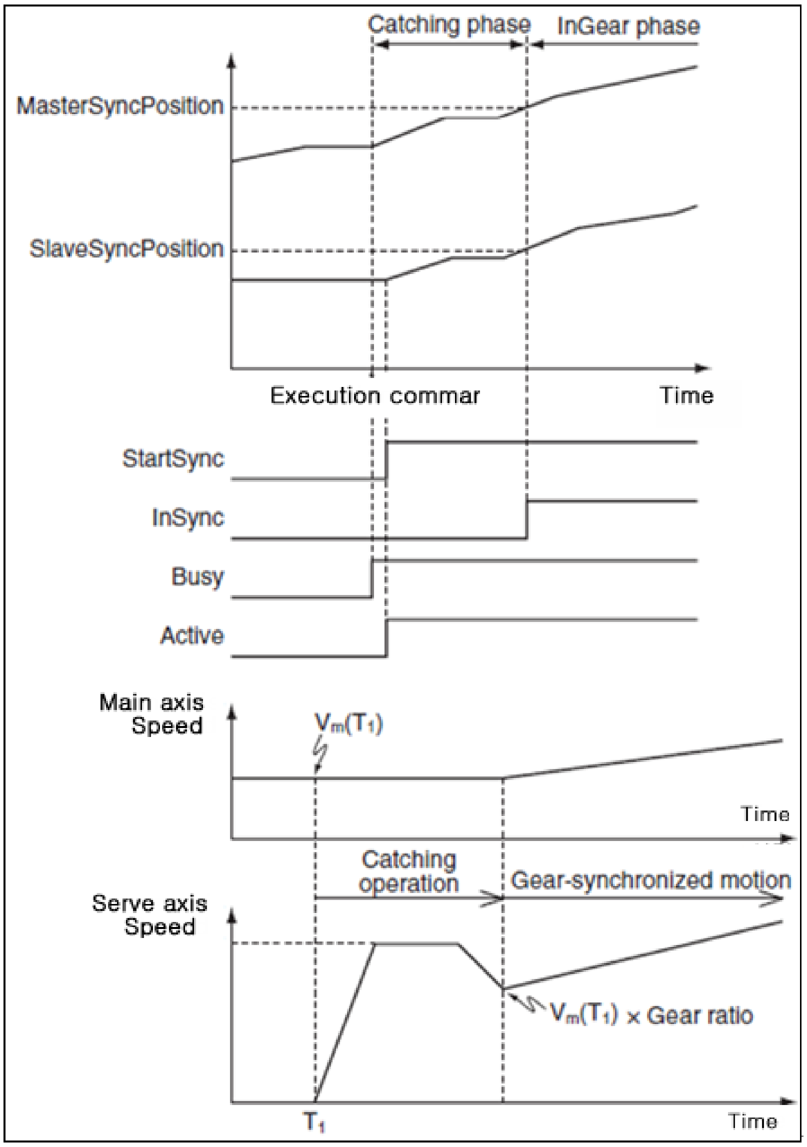


(2) Positioning gear operation

(a) Positioning gear operation makes speed synchronization of the master axis (or encoder) and the slave axis depending on the ratio set the same as in gear operation basically.

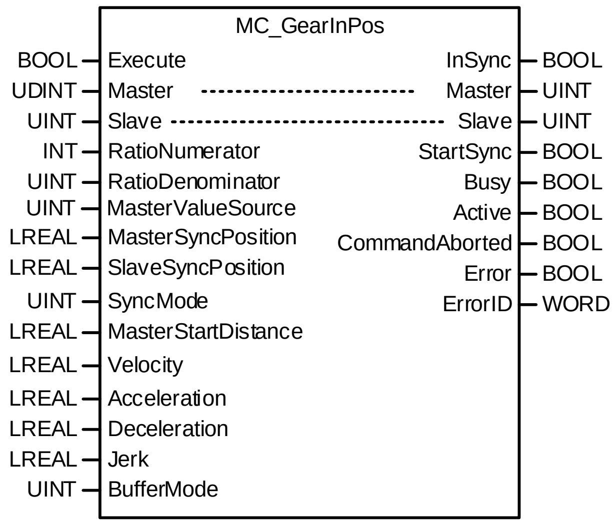
(b) The starting position in which the master axis and the slave axis are synchronized can be specified.

(c) Methods for operation are as follows.

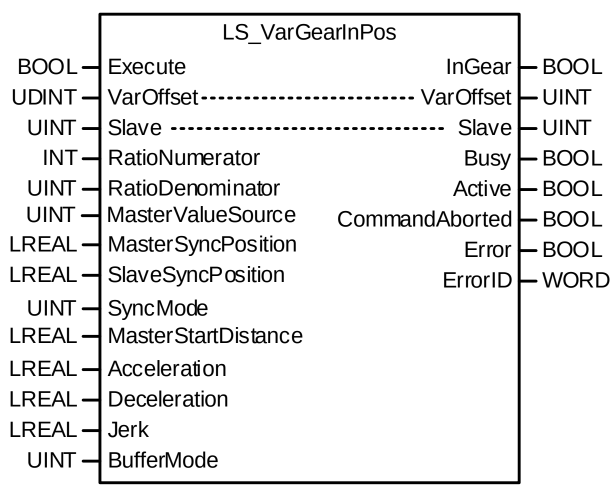

Catching phase
InGear phase
MasterSyncPosition
SlaveSyncPosition
Execution commar
Time
StartSync
InSync
Busy
Active
Main axis
Speed
Vm(T1)
Time
Catching
operation
Gear-synchronized motion
Serve axis
Speed
Vm(T1) × Gear ratio
T1
Time

(d) Relevant motion function block

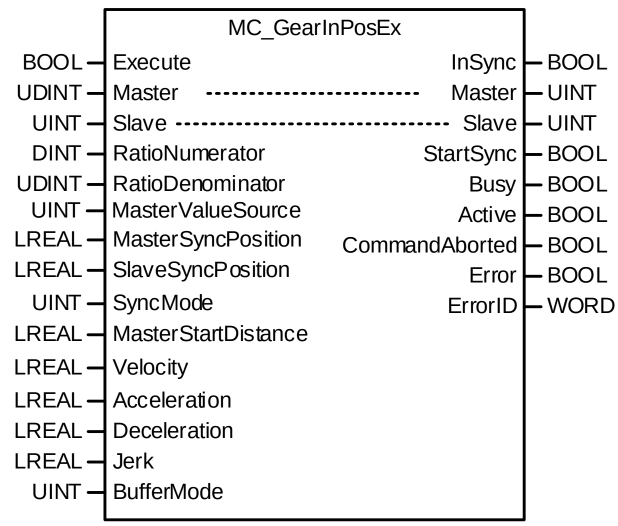
Name	Description	Operation condition
MC_GearInPos	Gearing by specifying the position	Edge



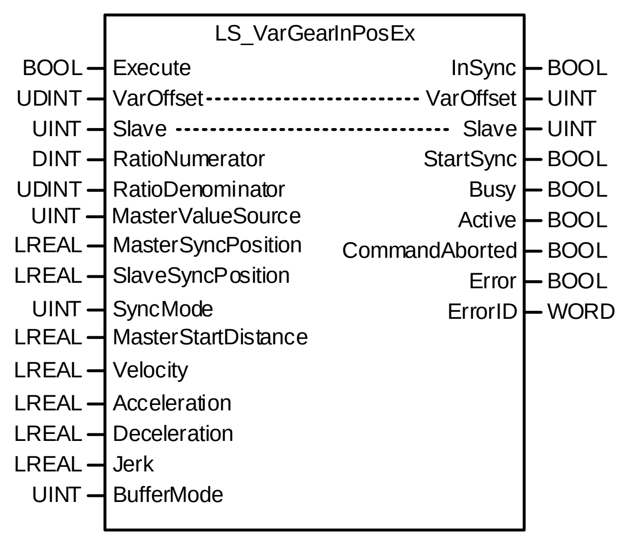
Name	Description	Operation condition
LS_VarGearInPos	Variable Gearing by specifying the position	Edge



Name	Description	Operation condition
MC_GearInPosEx	Expansion position specified gear operation	Edge



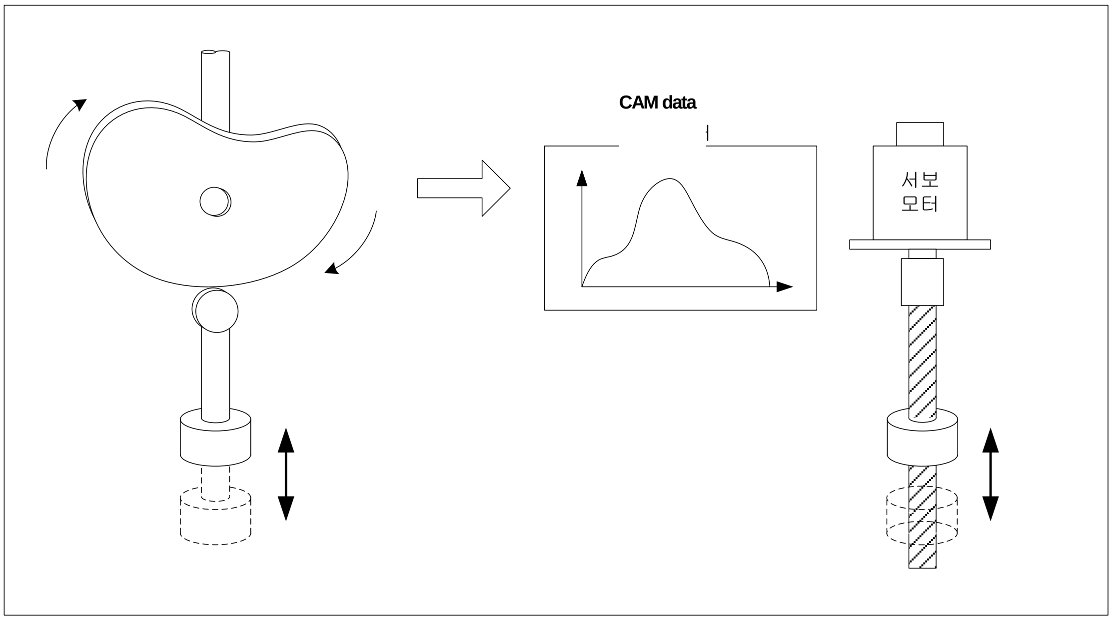
Name	Description	Operation condition
LS_VarGearInPosEx	Expansion position variable specified gear operation	Edge



(3) Cam Operation

(a) CAM operation controls cams by converting mechanical cam motion to the cam data set at the cam profile and synchronizing the data to the position of the motor designated as the main-axis.

(b) Mechanical cam operation in the past can be replaced with software cam motion using the cam data at the cam profiles.



(c) A total of 32 cam profiles can be generated, each of which can be applied to each axis regardless of their order.

(d) Each cam profile can consist of up to 32,768 cam data. (The number of cam data can be set variably for each cam profile, and a total of 32,768 data can be set in 32 cam profiles.)

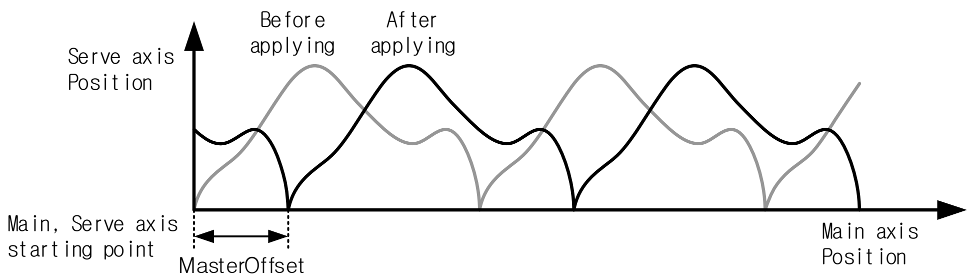
(e) To halt cam operation, MC_CamOut command should be issued on the sub-axis, or another motion function block should be operated (in case of Aborting).

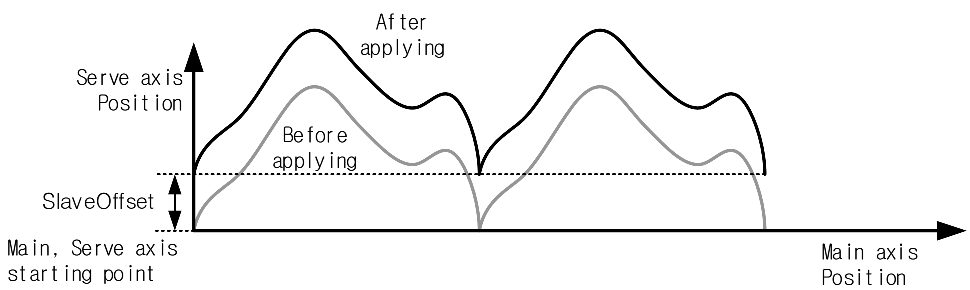
(f) Cam operation command's secondary data

Item	Content
Master	Set main axis (1~36: axis, 1001~1002: encoder),
MasterOffset	Set the offset value of the main axis.
SlaveOffset	Set the offset value of the slave cam table.
MasterScaling	Specify the magnification of the main axis.
SlaveScaling	Specify the magnification of the slave axis cam table.
MasterStartDistance	Specify the position of the main-axis where the cam operation of the master axis starts.
MasterSyncPosition	Specify the starting point at cam table when cam operation starts.
StartMode	Set the cam operation mode. 0: Cam table is applied as an absolute value. (mcAbsolute) 1: Cam table is applied as a relative value based on the command start position. (mcRelative)
MasterValueSource	Select the source of the main axis for cam operation. 0: Synchronizes to the command position of the main axis. 1: Synchronizes to the current position of the main axis.
CamTableID	Specify the cam table to operate.

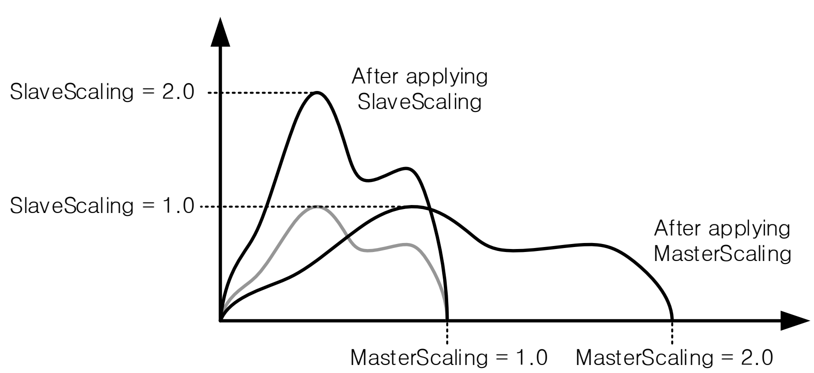
(ㄱ) Set the offset of the cam table to be applied in MasterOffset and SlaveOffset. MasterOffset determines the offset from the master axis start point, and Slaveoffset determines the offset from the slave axis start point. Refer to the Figure below.

Using offset may change the start position for cam operation, causing an abrupt operation. In such a case, MasterSyncPosition, MasterStartDistance should be used.

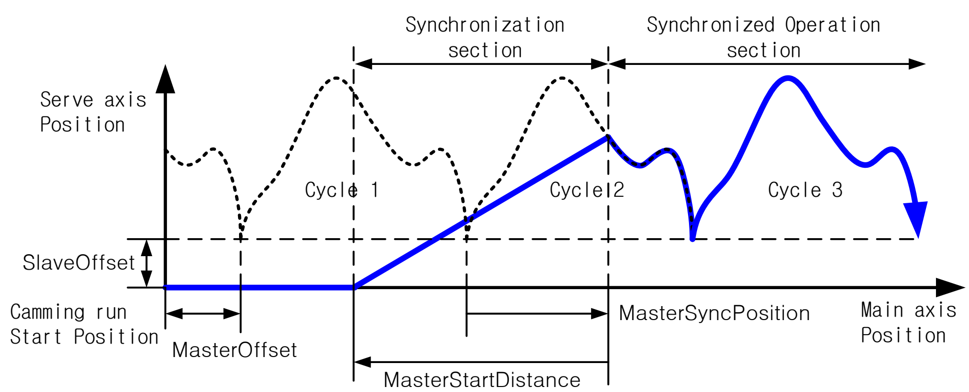




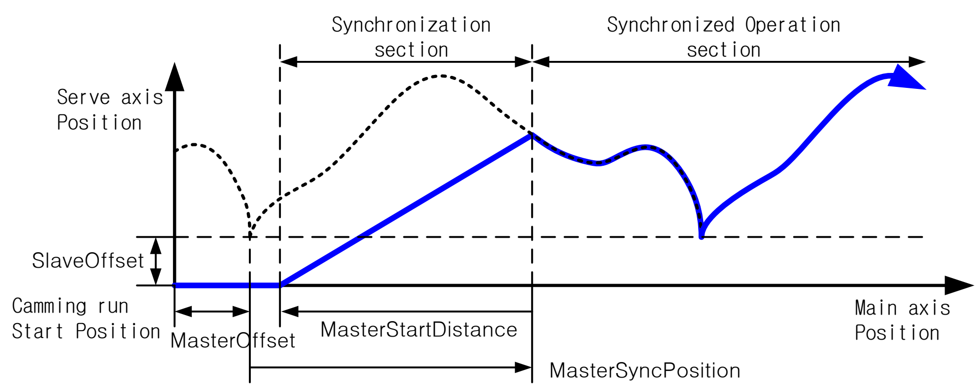
(ㄴ) Set the magnification of cam data to be applied in MasterScaling and SlaveScaling. MasterScaling determines the scale rate of the main-axis data, and SlaveScaling determines the scale rate of the sub-axis data. Refer to the Figure below.



MasterSyncPosition specifies the position of the master axis within the table where the synchronization of actual cam operation is completed, and MasterStartDistancespecifies the relative position of the master axis where the synchronization starts. Synchronization starts at a position as far away as MasterStartDistance from MasterSyncPosition. If unable to start synchronized operation at Cycle 1 as shown below (if the distance from the start position to the synchronized operation start position is shorter than MasterStartDistance), synchronized operation starts at Cycle 2.



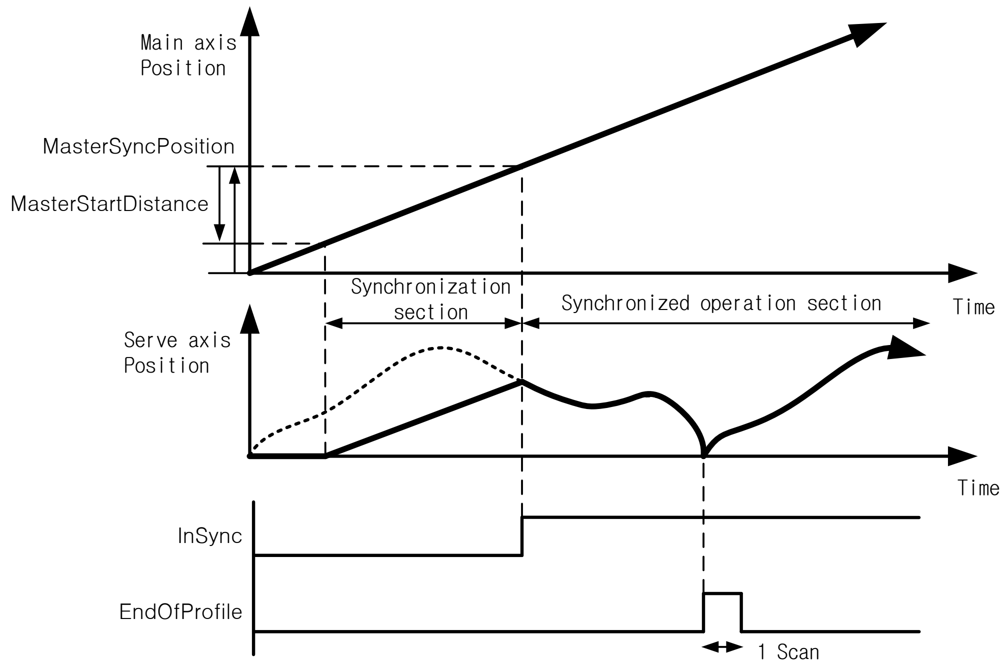
[When MasterScaling is 1.0]



[When MasterScaling is 2.0]

Actual synchronization position can vary depending on MasterScaling and SlaveScaling because MasterSyncPosition is a value based on the inside of cam table, but MasterOffset and MasterStartDistance value remain unaffected.

(ㄷ) Once cam operation starts normally, InSync output is On, and EndOfProfile output is 1 scan On every time one cam table operation is completed.



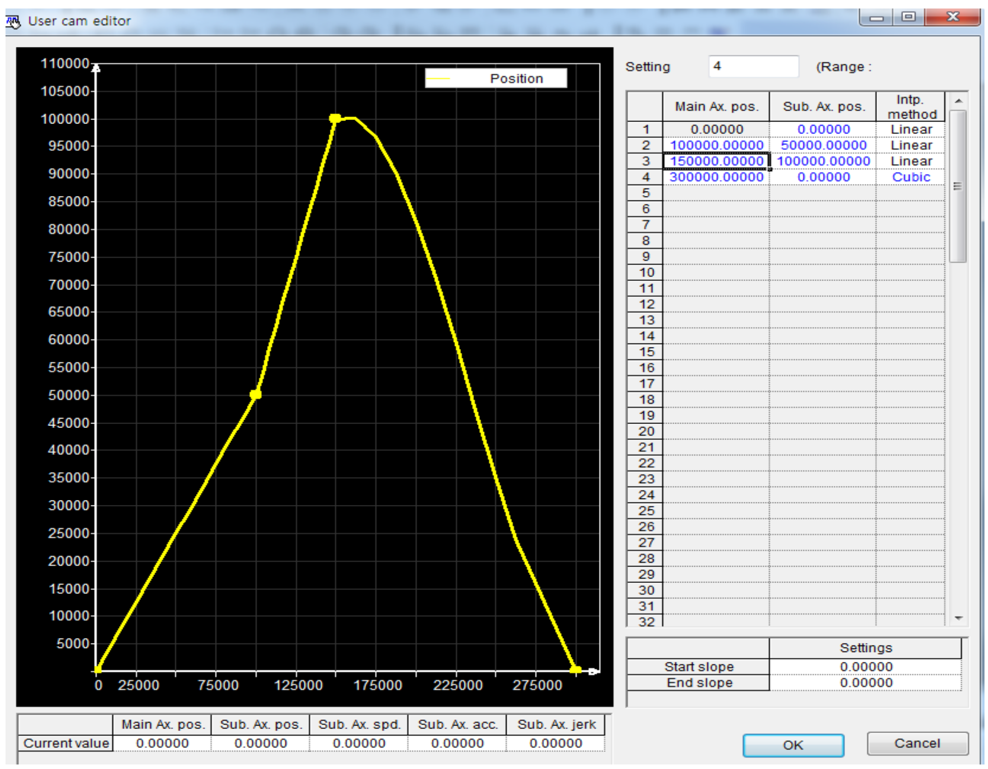
(ㄹ) Cam operation mode is set in StartMode. The setting range is either 0 or 1. If the input value exceeds the setting range, an error occurs. If it is set to 0, the cam table start position is set to the main-axis position of 0. If it is set to 1, the cam table start position is set to the current position of the main-axis.

(ㅁ) MasterValueSource selects the source of the main axis to be synchronized. If set to 0, the serve axis performs cam operations based on the command position of the main axis calculated in the motion controller, and if set to 1, the serve axis performs cam operations based on the current position received via communication from the servo drive of the main axis.

(ㅂ) CamTableID sets the number of cam table to be applied to cam operation. The setting range is from 1 to 32. If the input value exceeds the setting range, an error "0x1115" occurs at the motion function block.

(g) The items required when creating a cam profile are as follows.

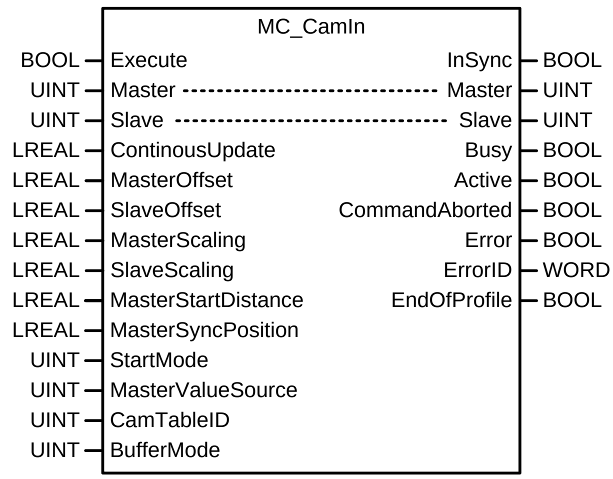
Item		Content
CAM data	Main axis position	Set the CAM position of sub axis corresponding to main axis
	Slave axis position	
	Interpolation type	Set the characteristic curve between the cam data. (Linear, Cubic)
Start inclination		If the interpolation type for the first or the last section is set to 'Cubic', set the start inclination and the end inclination for the Cubic operation.
End inclination		



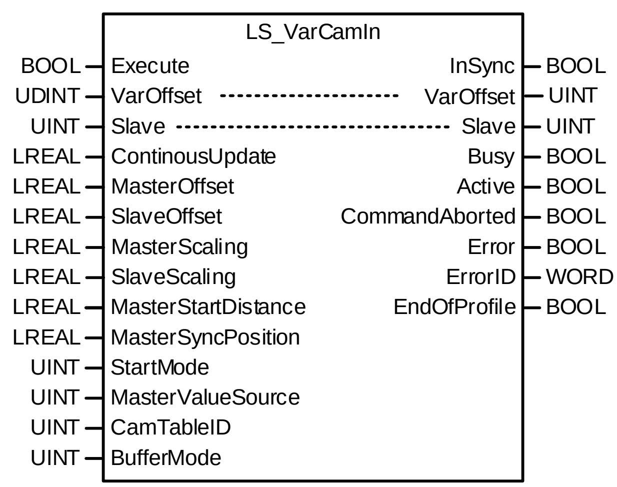
[Cam Profile setting example]

(h) Relevant motion function block

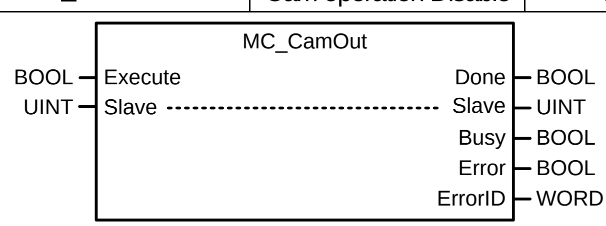
Name	Description	Operation condition
MC_CamIn	Cam Operation	Edge



Name	Description	Operation condition
LS_VarCamIn	Variable CAM operation	Edge



Name	Description	Operation condition
MC_CamOut	Cam operation Disable	Edge



(4) Skip CAM

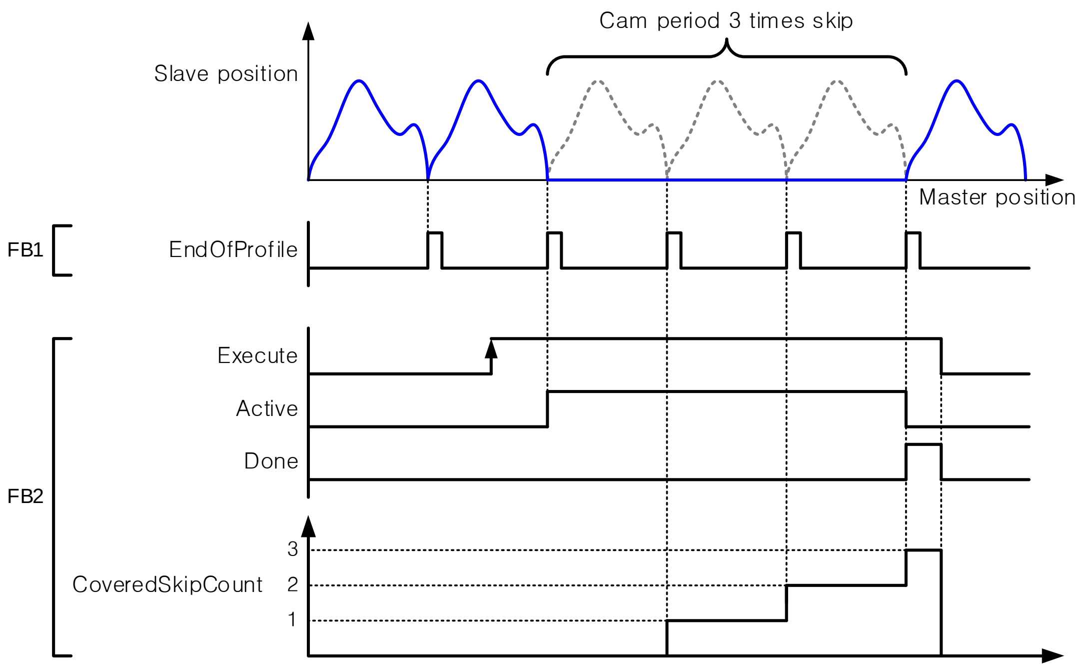
(a) This function skips the cam operation as many as the number of cam operation cycles that user wants in the axis where cam operation is underway.

(b) When Cam Skip command is issued on a sub-axis where cam operation is underway, the current cam cycle ends, and the skip operation starts. The sub-axis is in stand-by at the end position of the cam table in the cam skip motion.

(c) Cam skip command's secondary data

Item	Content
Slave	Set the sub axis to execute cam skip. (1~36: axis)
SkipCount	Set the number of cam cycles to skip.

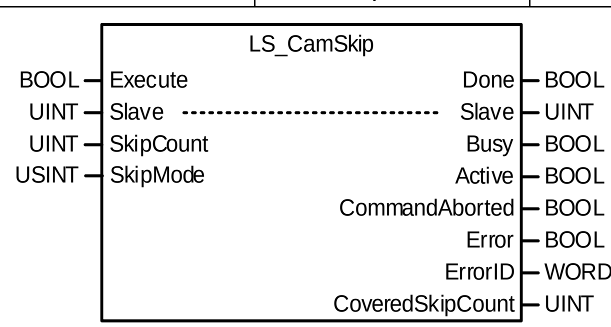
(d) After the execution of cam motion by MC_CamIn command (FB1), if three cycles are skipped using LS_CamSkip command (FB2), the output of each function block FB1 and FB2and the motion of the cam sub-axis are as displayed as shown in the figure below



(e) If Cam Skip command is re-executed during cam skip motion, or cam skip motion is aborted by another Cam Skip command, the SkimCount of the latter Cam Skip command applies, and a new cam skip motion starts from the beginning. In such a case, the number of cycles skipped at the time of re-execution is included in the cycles skipped after the re-execution. Therefore, the CoveredSkipCount value is 1 point larger than the SkipCount set by the user.

(f) Relevant motion function block

Name	Description	Operation condition
LS_CamSkip	Skip CAM	Edge



(5) Reverse Operation is Banned during Synchronized Operation

(a) In the state that synchronization control commands such as CAM and Gear are executed, this function stops synchronization control of the slave axis when the master axis operates in the reverse direction of synchronized operation referenced by the slave axis.

(b) After the master axis starts operation in the direction allowing synchronized operation again, when it passes by the position that started operation in the reverse direction, the slave axis starts synchronization control again.

(c) The operation direction of the master axis, which is referenced upon synchronized operation, can be set in the expanded setting of axis parameters.

Group	Parameter	Setting value
Extended parameter	Synchronization methods according to the operation direction of the master axis	0: Synchronization with both directions of the master axis 1: Synchronization with the forward direction of the master axis 2: Synchronization with the reverse direction of the master axis

(d) Operation according to parameter setting

(ㄱ) 0: Synchronization with both directions of the master axis

Perform the synchronization control operation according to synchronization commands executed on the slave axis by the operation direction of the master axis. Synchronization control operation is not separately limited by the operation direction of the master axis.

(ㄴ) 1: Synchronization with the forward direction of the master axis

If the master axis performs reverse operation when starting synchronization control. The slave axis stands still without starting synchronization control. After the master axis changed its operation in the forward direction, when the position of the master axis passes by the position that started synchronization control, the slave axis starts synchronization control operation.

If the master axis performs forward operation when starting synchronization control. The slave axis performs the synchronization control operation according to synchronization commands executed on the slave axis by the operation direction of the master axis. If the master axis changed its operation in the reverse direction, the slave axis stops synchronization control. After the master axis changed its operation in the forward direction, when the position of the master axis passes by the position that started operation in the reverse direction, the slave axis starts synchronization control again.

a) 2: Synchronization with the reverse direction of the master axis

a) If the master axis performs forward operation when starting synchronization control

The slave axis stands still without starting synchronization control. After the master axis changed its operation in the reverse direction, when the position of the master axis passes by the position that started synchronization control, the slave axis starts the synchronization control operation.

b) If the master axis performs reverse operation when starting synchronization control

The slave axis performs the synchronization control operation according to synchronization commands executed on the slave axis by the operation direction of the master axis. If the master axis changed its operation in the forward direction, the slave axis stops synchronization control. After the master axis changed its operation in the reverse direction again, when the position of the master axis passes by the position that started operation in the reverse direction, the slave axis starts synchronization control again.

(e) Confirm the direction of the master axis according to MasterValueSource

(ㄱ) 0: Synchronizes to the command position of the main axis.

If the command position value increases, it is judged as the forward operation of the master axis. If the command position value decreases, it is judged as the reverse operation of the master axis.

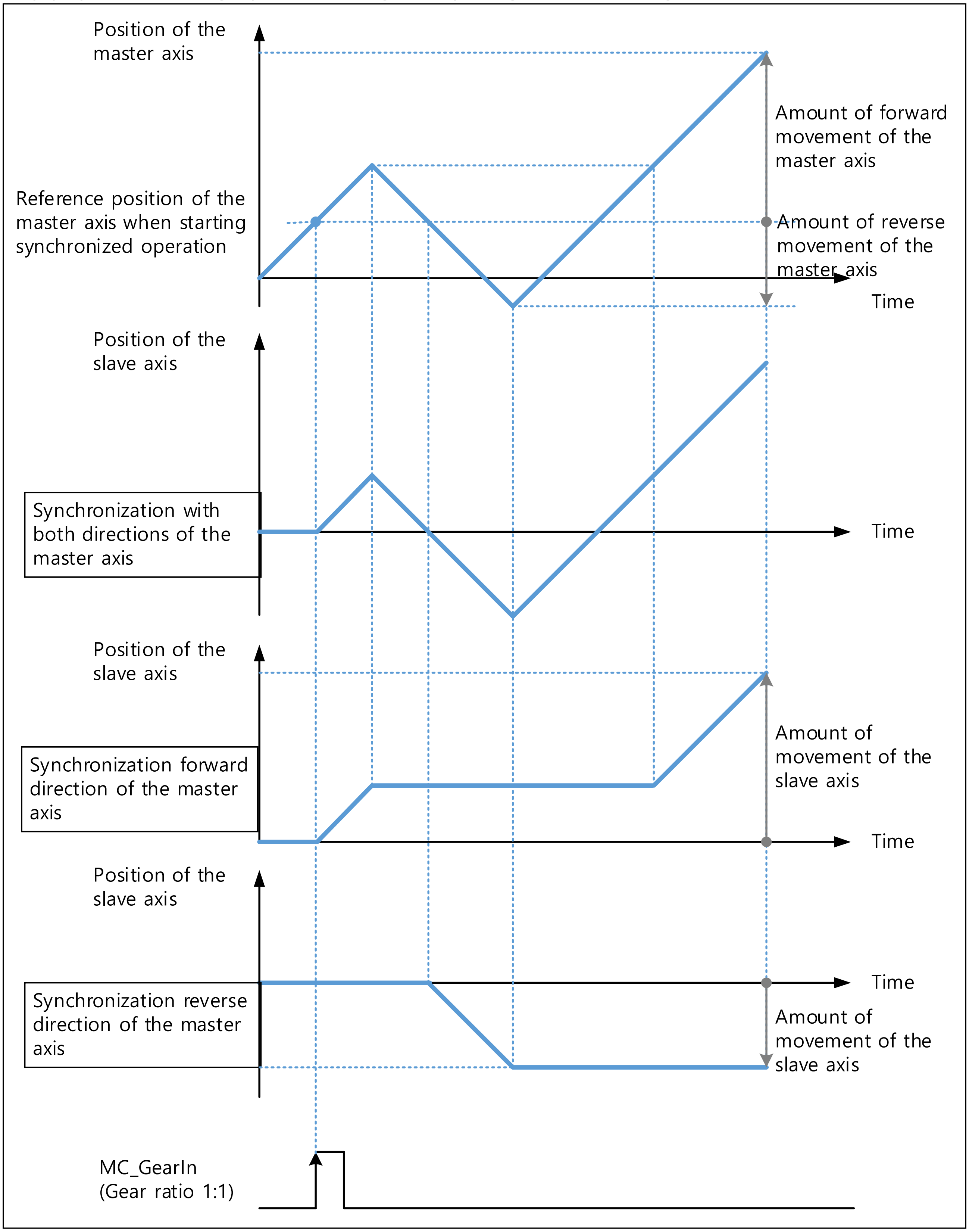
(ㄴ) 1: Synchronization with the forward direction of the master axis

If the current position value increases, it is judged as the forward operation of the master axis.
If the current position value decreases, it is judged as the reverse operation of the master axis.

※ If the master axis is an 'encoder' or a synchronization control command specifying variables, the operation according to the MasterValueSource input value is identical as there is no separation between the current position and separate command positions. In other words, if the encoder position or variable value increases irrespective of the MasterValueSource value, it is judged as forward operation.

(f) Operation Timing

(ㄱ) Operation according to parameter setting when operating the Gear with the gear ratio of 1:1



(g) Applying synchronization control commands

(ㄱ) CAM Operation

MC_CamIn, LS_VarCamIn, LS_OnOffCam

(ㄴ) Gear operation

MC_GearIn, LS_VarGearIn, MC_GearInPos, LS_VarGearInPos

(h) The version information that can use the function where reverse operation is banned during synchronized operation is as follows:

Item / Product name	Module O/S	XG5000
XMC-E32A **XMC-E16A** **XMC-E08A** **XMC-E32C**	V1.40	V4.28

8.2.12 Manual Control

(1) JOG Operation

(a) Jog operation makes positioning control by manual jog commands of users.

(b) Jog operation is possible even in the state in which the origin of the axis is not determined.

(c) Jog commands are executed even in the origin determined or undetermined status, which makes it possible to monitor changes in position values of the axis.

(d) Acceleration/Deceleration process and Jog speed

For processing acceleration and deceleration, acceleration and deceleration control is made based on the value set in Jog Acceleration/Deceleration/Jerk among [Operation parameter – expansion parameter] setting items.

Jog speed is set in Jog high-speed and Jog low-speed among [Operation parameter – expansion parameter] setting items.

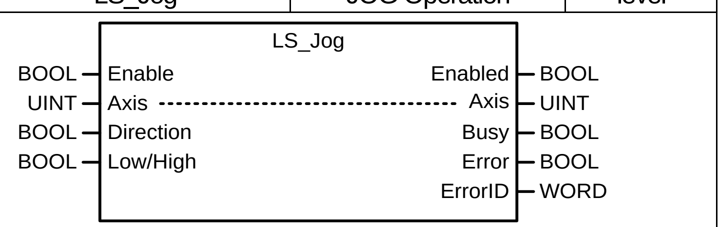
Jog high-speed should be set to at the speed limit or less or at least Jog low-speed among [Operation parameter – basic parameter] setting items.

(e) Related parameter setting

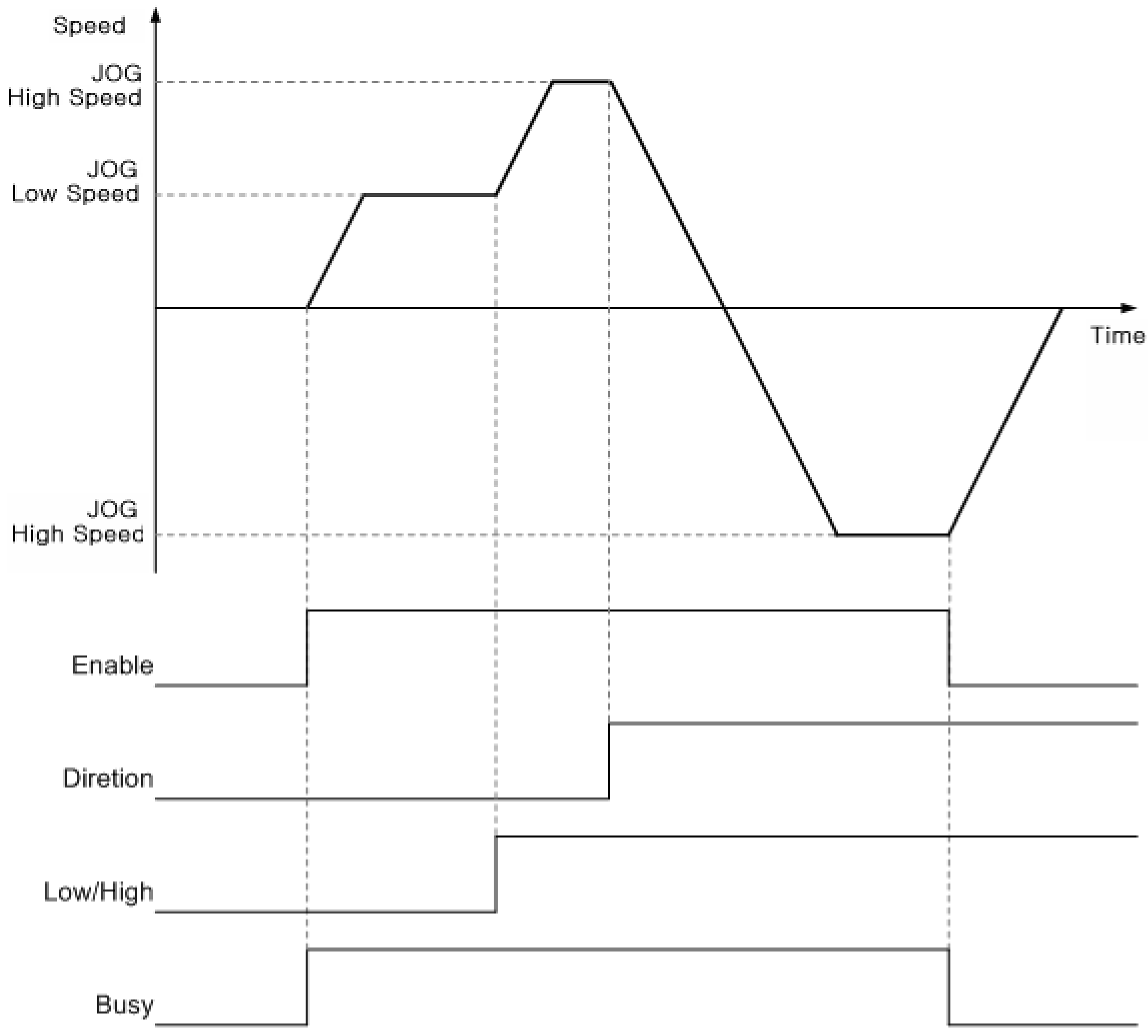
Item	Setting range	Initial value
JOG high speed	Long real (LREAL) positive number	100000 pls/s
Jog low speed		10000 pls/s
JOG Acceleration	0 or Long real(LREAL) Positive number	100000 pls/s^2
JOG Deceleration		100000 deg/s^2
JOG Jerk		0 deg/s^3

(f) Relevant motion function block

Name	Description	Operation condition
LS_Jog	JOG Operation	level



(g) Operation Timing



8.2.13 SuperImposed operation

SuperImposed operation executes the positioning control additionally as much as the moving distance designated in the current motion operation.

(1) Features of control

(a) When SuperImposed operation command is executed, the axis moves from the point at the time of command execution to the target distance specified in the Distance input.

(b) The direction of the movement is determined by the positivity/negativity of the set distance. Positive distance (+ or no sign) means forward movement, and negative distance (-) means reverse movement.

(c) The existing motion is not canceled, but its operation overlaps with SuperImposed operation.

(d) Even when the existing motion is completed, SuperImposed operation continues unless the amount of movement does not reach the one specified in the SuperImposedoperation.

(e) If the axis is not in operation, but in "StandStill" status, SuperImposed operation works the same way as MC_MoveRelative operation.

(f) The current SuperImposed operation can be halted with the MC_HaltSuperImposed command. After executing the command, SuperImposed operation decelerates and stops at the given acceleration and jerk. The existing motion which is currently being executed is not affected.

(2) Relevant motion function block

Name	Description	Operation condition
MC_MoveSuperImposed	SuperImposed Operation	Edge

```
                 MC_MoveSuperImposed
  BOOL ─ Execute                           Done ─ BOOL
  UINT ─ Axis ---------------------------- Axis ─ UINT
  BOOL ─ ContinuousUpdate                  Busy ─ BOOL
 LREAL ─ Distance                        Active ─ BOOL
 LREAL ─ VelocityDiff            CommandAborted ─ BOOL
 LREAL ─ Acceleration                     Error ─ BOOL
 LREAL ─ Deceleration                   ErrorID ─ WORD
 LREAL ─ Jerk                   CoveredDistance ─ LREAL
```

Name	Description	Operation condition
MC_HaltSuperImposed	SuperImposed operation halt	Edge

```
                 MC_HaltSuperImposed
  BOOL ─ Execute                           Done ─ BOOL
  UINT ─ Axis ---------------------------- Axis ─ UINT
 LREAL ─ Deceleration                      Busy ─ BOOL
 LREAL ─ Jerk                            Active ─ BOOL
                                 CommandAborted ─ BOOL
                                          Error ─ BOOL
                                        ErrorID ─ WORD
```

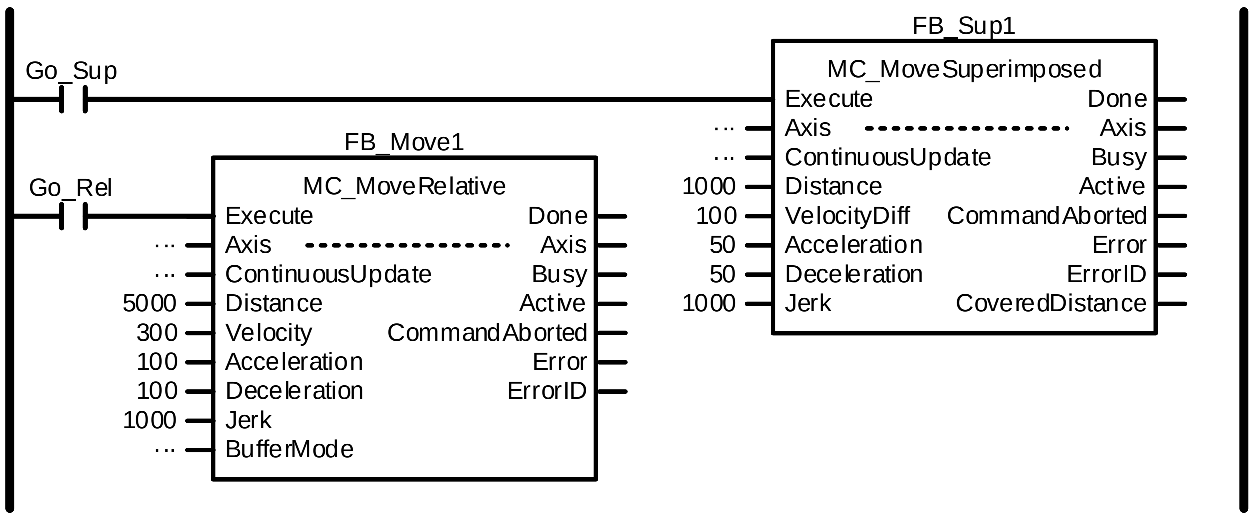
(3) Restrictions

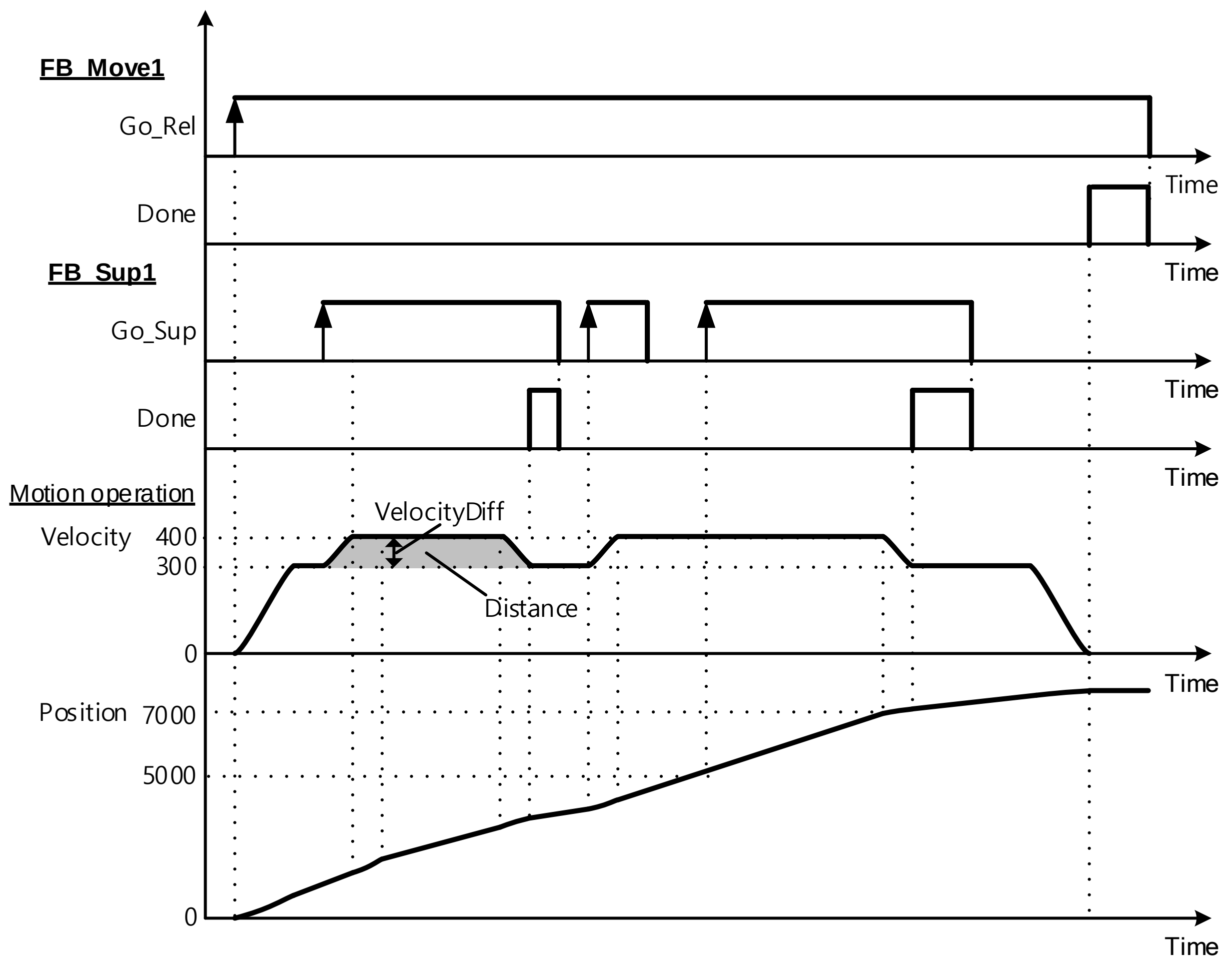
In the following cases, SuperImposed operation cannot be performed due to errors.

(a) SuperImposed command is executed during velocity control or torque control operation (Error Code: 0x1082)

(b) MC_HaltSuperImposed command is executed when SuperImposed operation is not being performed (Error Code: 0x1083)

(4) Operation Timing





8.2.14 Phase compensation control

Phase correction control performs phase correction for the main-axis of the axes during synchronization control operation. It performs a virtual movement of the position of the main-axis which the sub-axis refers to in synchronization control operation, and the sub-axis performs synchronized operation to the moved main-axis position.

(1) Features of control

(a) Phase correction order can be executed with respect to the axes where synchronized operation is underway as in gear operation or cam operation.

(b) Phase correction does not change the actual command position or current position of the main-axis, and phase correction is performed on the main-axis position referred to by sub-axis in synchronous control operation. In other words, the main-axis does not know that phase correction is executed by the sub-axis. Phase correction velocity is relative to the velocity of the current main-axis operation.

(c) If the main-axis is encoder, when phase correction control is executed, the operation uses the velocity limit of the sub-axis.

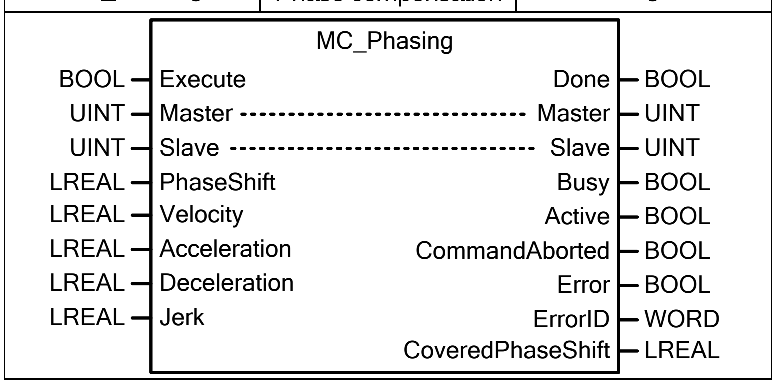
(d) The main axis position referenced by the subordinate axis during synchronous operation is “actual main axis position + phase compensation control position”.

(e) If the command is re-executed during the phase correction operation, phase correction is performed again from the current position. In other words, PhaseShift is operated in a relative value.

(f) If re-executing phase correction by setting PhaseShiftto 0 during the phase correction operation, the existing phase correction operation stops immediately.

(2) Relevant motion function block

Name	Description	Operation condition
MC_Phasing	Phase compensation	Edge



(3) Restrictions

Phase compensation control can not be executed in the case below.

(a) Sub-axis is not performing synchronization control operation (Error Code: 0x1130)

(b) The designated main-axis is the main-axis of the actual synchronized operation (Error Code: 0x1131)

(c) PhaseShift is outside the pulse unit position expression range (INT) (Error Code: 0x1132)

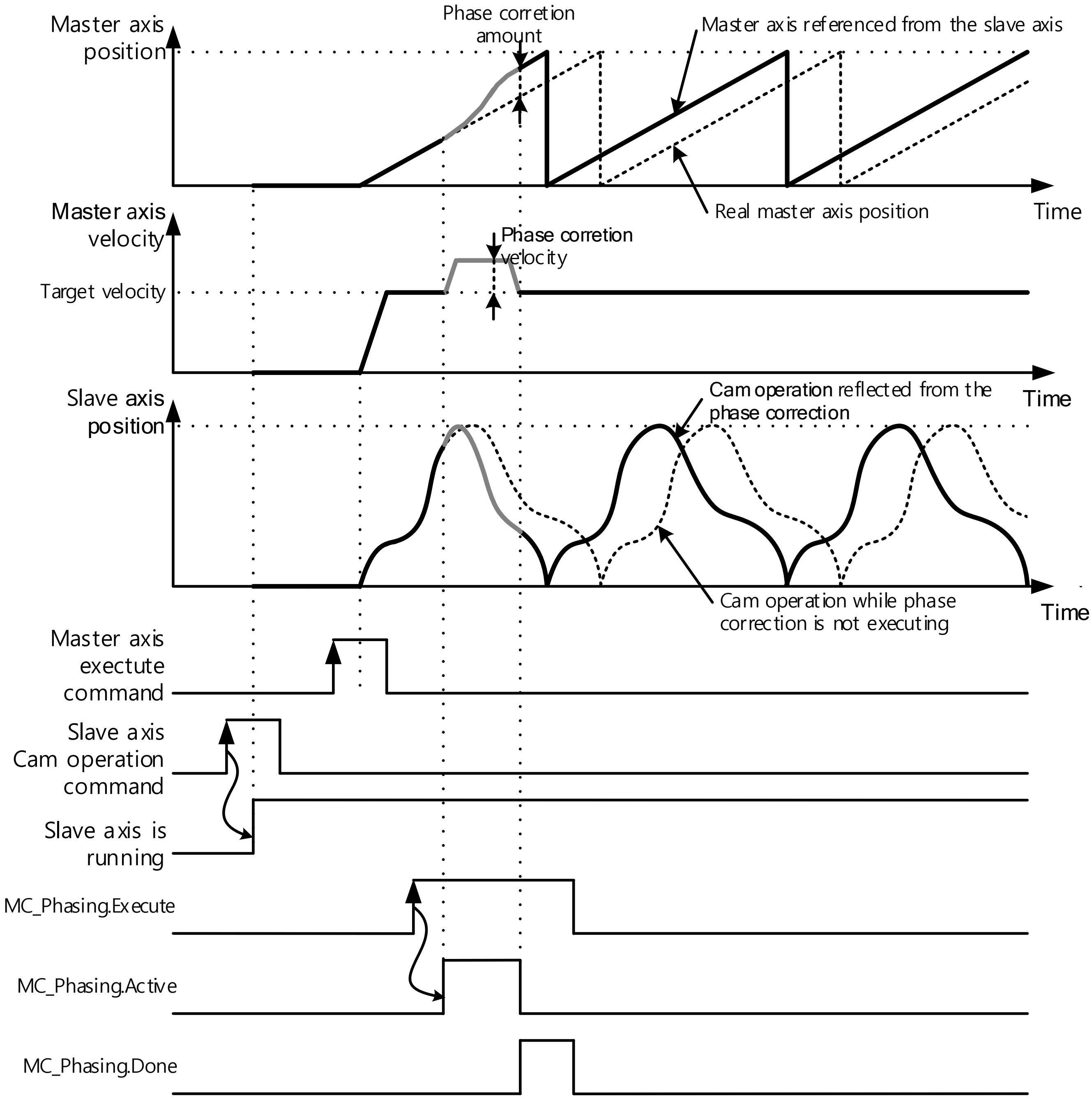
(d) Velocity setting is less than0, or exceeds the velocity limit for the main-axis (Error Code: 0x1133)

(e) Acceleration setting is less than 0 (Error Code: 0x1014)

(f) Deceleration setting is less than 0 (Error Code: 0x1015)

(g) Jerk setting is less than 0 (Error Code: 0x1016)

(4) Operation Timing



8.2.15 Cross-coupled Control

This is a function that provides stable control when synchronously controlling two physically coupled/connected axes such as Automated Guided Vehicles and Gantry Stages.

(1) Cross-coupled control

In the case of synchronous control of two physically coupled/connected axes, the motor output for controlling one of them may affect the other, resulting in poor control performance. In this case, if cross-coupeld control is applied, the position error occurring in each axis is shared with each other, so that the position error occurring in the other axis does not act as a factor that interferes with control.

Since the cross-coupled control uses a modified position control loop, it is designed to operate only when the master position loop control is executed.

(2) Master position control loop

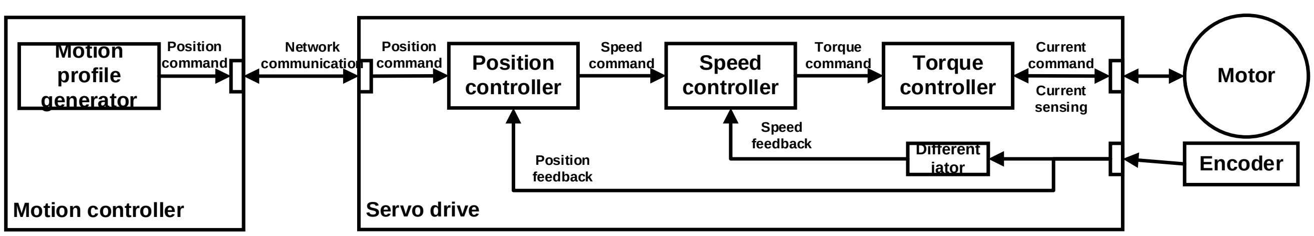
In the conventional operation of the motion controller, the drive operates in CSP mode, and the entire servo control loop is executed in the drive by passing the position command input through the direct start command from the motion controller to the drive.

When master position loop control is executed, the drive operates in CSV mode, and the positioning module generates a speed command by comparing the position command input through the direct start command from the motion controller with the current position fed back from the drive. This speed command is transmitted to the drive operating in CSV mode to execute the control servo loop. The position control part of the entire servo control loop is performed by the motion controller.

(3) Operation principles

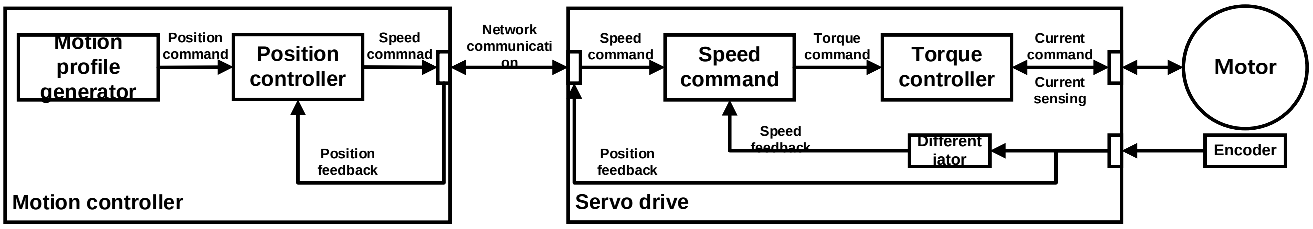
(a) Existing control system

Conventionally, the position command generated by the motion profile generator of the motion controller is transmitted to the servo drive through network communication. When a position command is transmitted to the servo drive, the entire servo loop for motor control is operated within the servo drive.



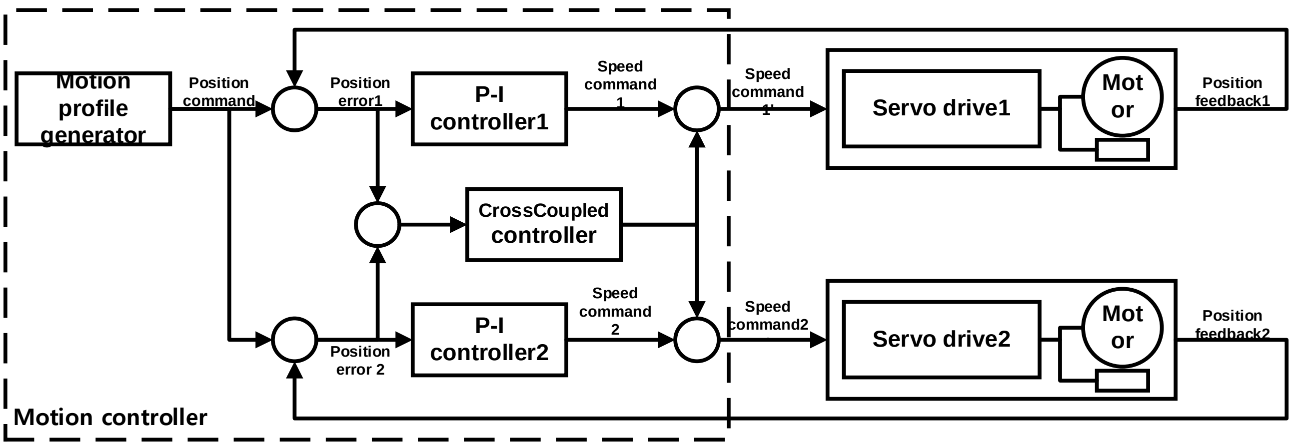
(b) Master position control loop

If the master position control loop is used, the position command generated by the motion profile generator is not directly transmitted to the servo drive, but is input to the position controller inside the positioning module. When the position feedback transmitted through network communication is input to the position controller, the difference (error) between the position command and the position feedback is PI (proportional integral) controlled to generate a speed command. The generated speed command is transmitted to the servo drive through network communication. When a speed command is transmitted to the servo drive, the remaining servo loop for motor control is operated within the servo drive. The position control loop is executed in the motion controller and the servo drive operates in CSV mode. The motor connected to the relevant axis can be operated with the general start command. The position controller is a PI controller. Control gain tuning of position controller is required. During control gain tuning (change), there is no need to cancel the master position control loop command, but change the gain or change the gain and turn on the contact while the master position control loop is running.



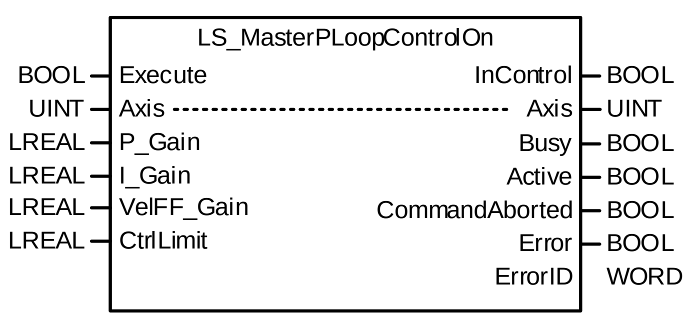
(c) Cross-coupled control

Cross-coupled control can be executed when the two connected axes each have the master position control loop activated. A simplified block diagram of cross-coupled control is shown below to make it easier to understand. Cross-coupled control is performed with position error 1 and position error 2 generated in servo system 1 and servo system 2. Cross-coupled controller is a proportional controller, and the difference between position error 1 and position error 2 is used as input. Controller proportional gain tuning is required. A speed command with a cross-coupled control command added to the speed command generated by the position controller is transmitted to the servo drive. When servo control is executed on one servo drive, the position error information of the partner motor is reflected through the cross-coupled control, so that the output for controlling the partner motor does not act as a disturbance when controlling the main motor.

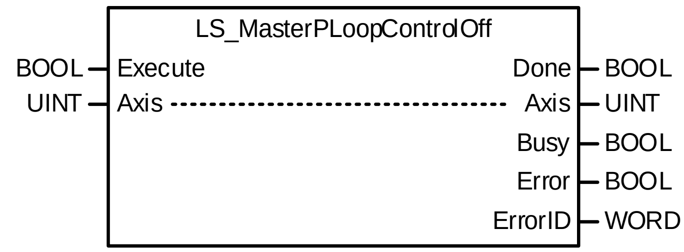


(4) Relevant motion function block

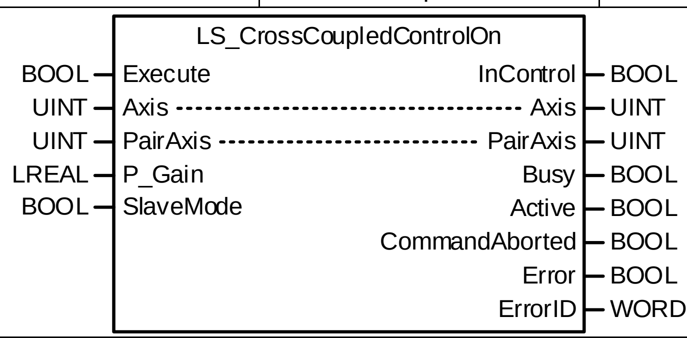
Name	Description	Operation condition
LS_MasterPLoopControlOn	Master position loop control	Edge



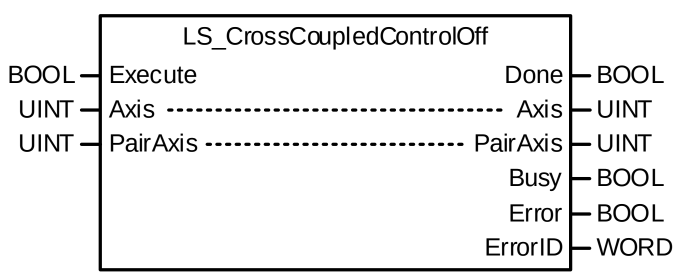
Name	Description	Operation condition
LS_MasterPLoopControlOff	Master position loop control disable	Edge



Name	Description	Operation condition
LS_CrossCoupledControlOn	Cross-coupled control	Edge



Name	Description	Operation condition
LS_ CrossCoupledControlOff	Cross-coupled control disable	Edge



(5) Restrictions

In the following cases, an error occurs and master position loop control/cross-coupled control cannot be executed.

(a) When executing or releasing master position loop control while the axis is in operation (Error code: 0x1230)

(b) When the driver to execute master position loop control does not support CSV operation mode (Error code: 0x1160)

(c) If there is no target speed (0x60FF) object in RxPDO of the axis to execute master position loop control (Error code: 0x1161)

(d) When the P gain input value of the master position loop control function block is negative (error code: 0x1231)

(e) When the I gain input value of the master position loop control function block is negative (error code: 0x1232)

(f) When the control output limit value of the master position loop control function block is negative (error code: 0x1233)

(g) When executing or releaseing Cross-coupled control while the axis is in operation (Error code: 0x1234)

(h) When the same axis is set in Axis and PairAxis of Cross-coupled control function block (Error code: 0x1235)

(i) In case master position loop control is not being executed for Axis and PairAxis set in Cross-coupled control function block (Error code: 0x1236)

(j) When the P gain input value of cross-coupled control function block is negative (error code: 0x1237)

(k) When a command is executed on an axis for which cross-coupled control is not being executed (Error code: 0x1238)

(l) When homing start is executed during master position loop control (Error code: 0x10B8)

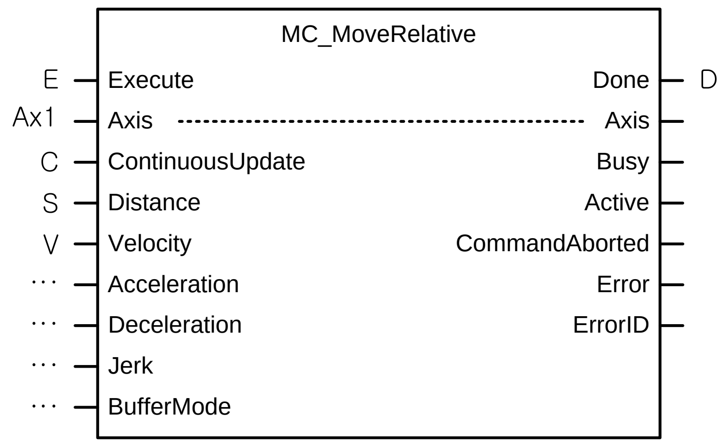
8.3 Other Functions

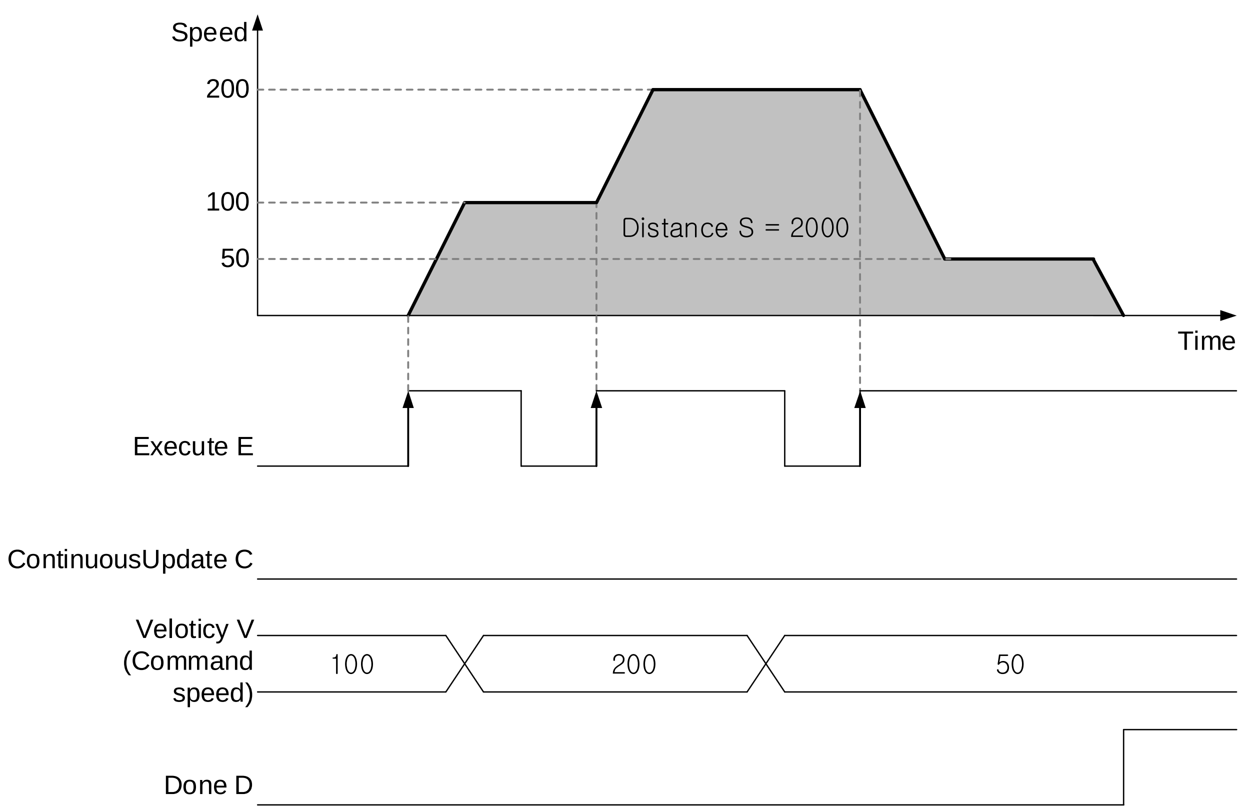
8.3.1 Modification Function of Control

(1) Changes in input variables of motion function block in execution

(a) In case there is no ContinuousUpdate input in motion function block, or execution (Execute input enabled) is made when ContinuousUpdate input is Off, the motion function block is operated with the input at the time when Execute input is On(rising Edge) applied. To operate by changing the input of the motion function block during operation, get the Execute input to be On after changing input value, and the changed value is immediately applied for operation.

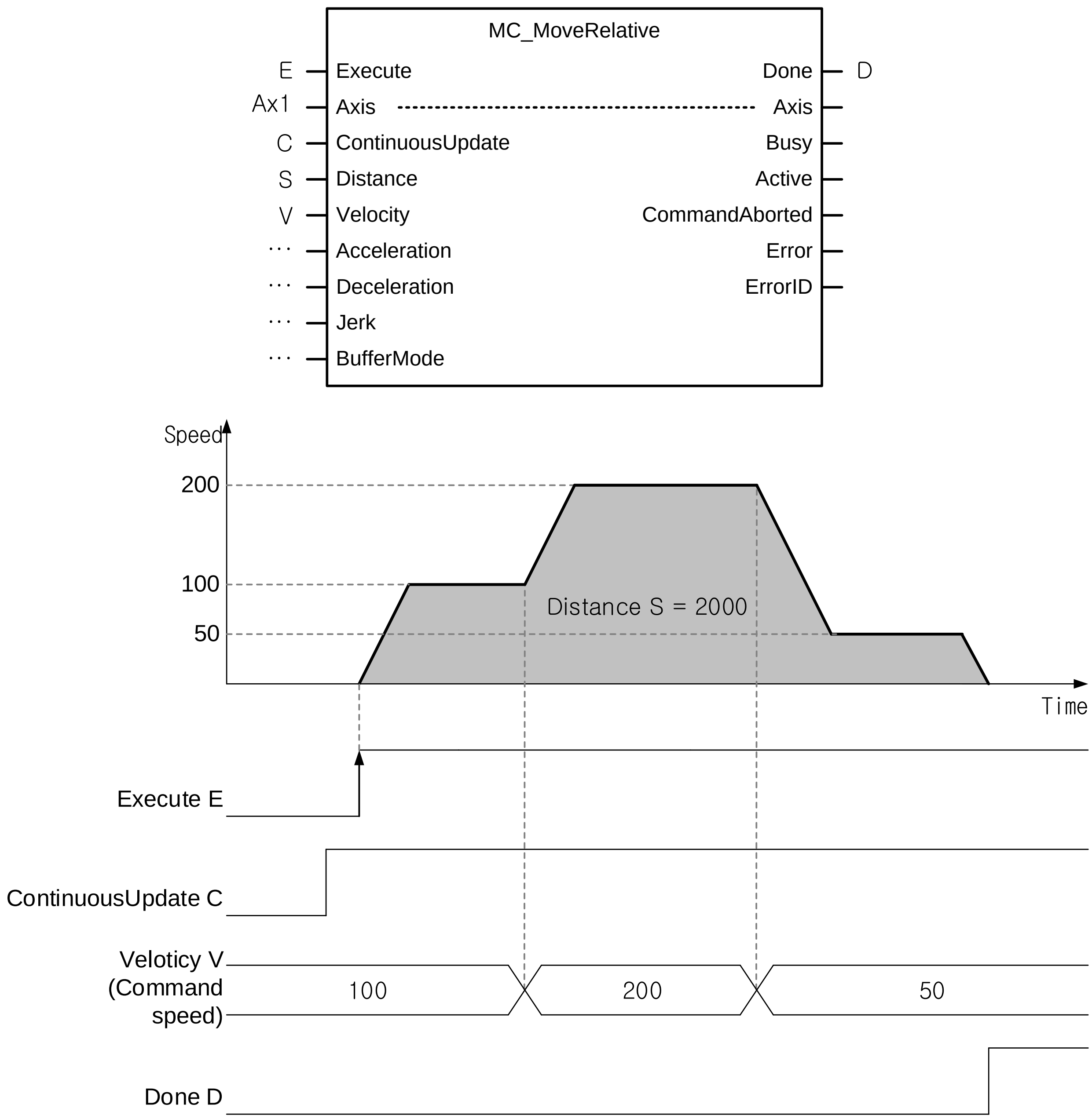
[Example] Input variable update of motion control command by re-execution of Execute





(b) In case ContinuousUpdate input is On in Edge operation motion function block, the input at the time when Execute input is On (rising Edge) is applied to the motion function block if Execute input is On,   That is the motion function block makes a motion to reflect the change if the input is changed while ContinuousUpdate input is On. However, changes in input are no longer reflected after the operation of the motion function block is completed or stopped (Busy output disabled).

[Example]Input variable update of the motion control command when ContinuousUpdate is On



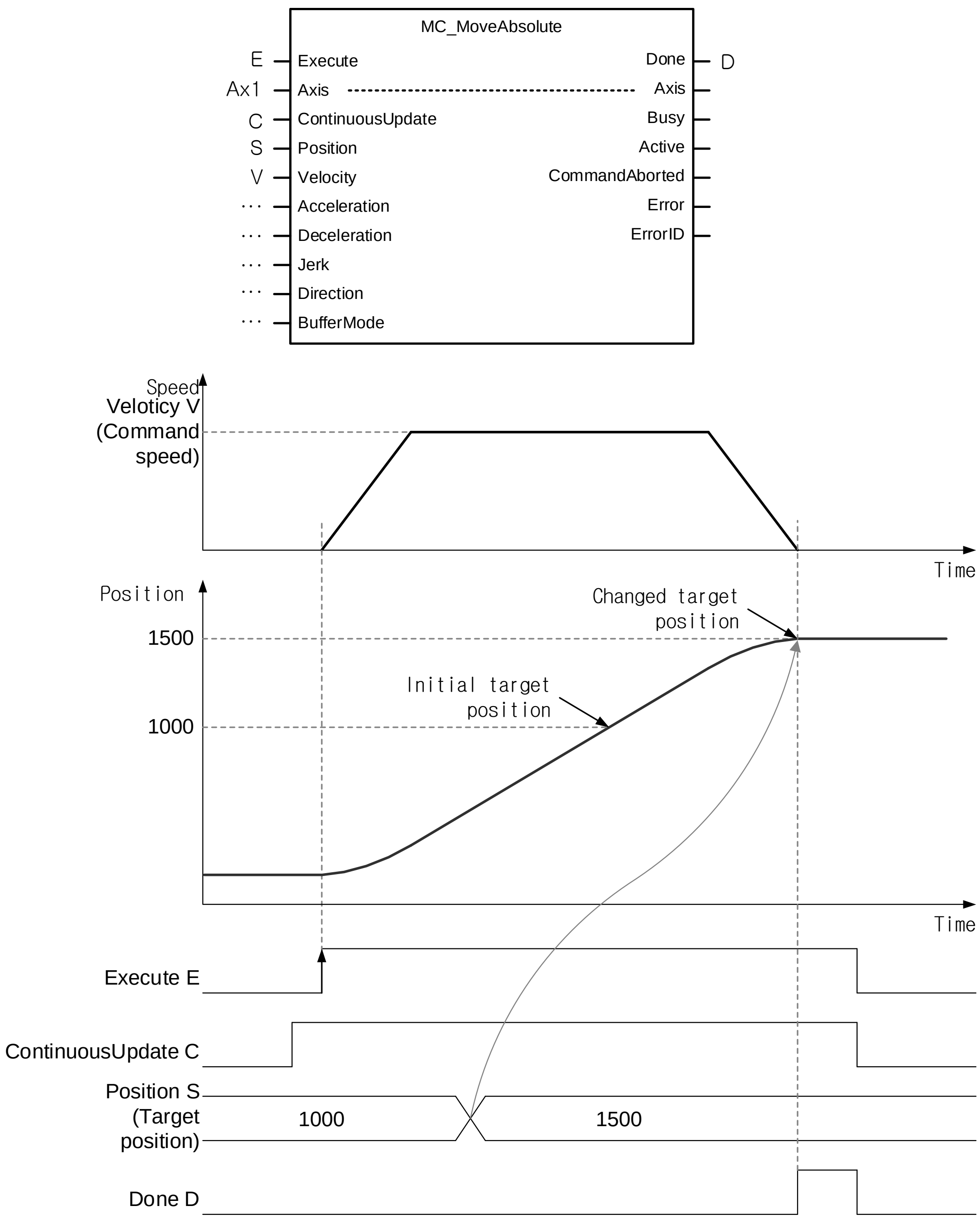
(c) In case of the level operation motion function block, input variables at the time when Enable input is On (rising Edge) are applied to operate, and input variables can be changed continuously while Enable input is On.

(2) Position override

(a) It is a function to override the target position of the axis in position operation. Override function is enabled by suing ContinuousUpdate input of the position operation motion function block. When the position operation motion function block is being executed, the position operation to reflect changed objectives is performed by tuning Execute input On again by changing the target position after turning ContinuousUpdate input of the motion function block On.

(b) In case the target position changed at the point in time when changes in the target position are reflected is greater than the position in case of the velocity being reduced to stop from the current velocity, positioning is made in the direction of the current movement. On the contrary, in case the changed position is smaller than the position in case of the velocity being reduced to stop from the current velocity, positioning is made in the direction of the target position by operating to the opposite direction after deceleration stop.

[Example] Position override using ContinuousUpdate



(3) Velocity, Acceleration/Deceleration, Jerk override

(a) It is a function to conduct velocity, acceleration/deceleration and jerk override of the specified axis

(b) It can override velocity, acceleration/deceleration, jerk to absolute value using ContinuousUpdate input of the motion function block in operation. When the operation motion function block is being executed, the operation to reflect changed velocity and acceleration is performed by tuning Execute input On again by changing the velocity and acceleration after turning ContinuousUpdate input of the motion function block On.

(c) For the execution of speed override operation at the rate on the current command speed, not an absolute value override (MC_SetOverride) motion function block is used for the override.

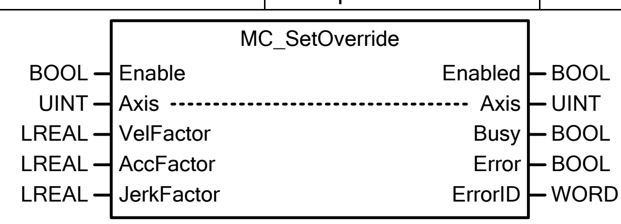
- In case the value is 1.0, the current operating speed, acceleration/deceleration, jerk is the same as before.
- In case VelFactor value specified is 0.0, the axis comes to a stop, but it cannot be changed to 'StandStil' state.
- If AccFactor value is 0.0, the changed velocity value is immediately applied without acceleration/deceleration.
- If JerkFactor value is 0.0, the acceleration/deceleration rate is immediately applied, and therefore the command velocity linearly accelerates/decelerates.
- The meaning of Factor value specified of override (MC_SetOverride) motion function block differs depending on the override item value of common parameters.

  0: percentage specified – Factor value operates at the rate on the current command speed

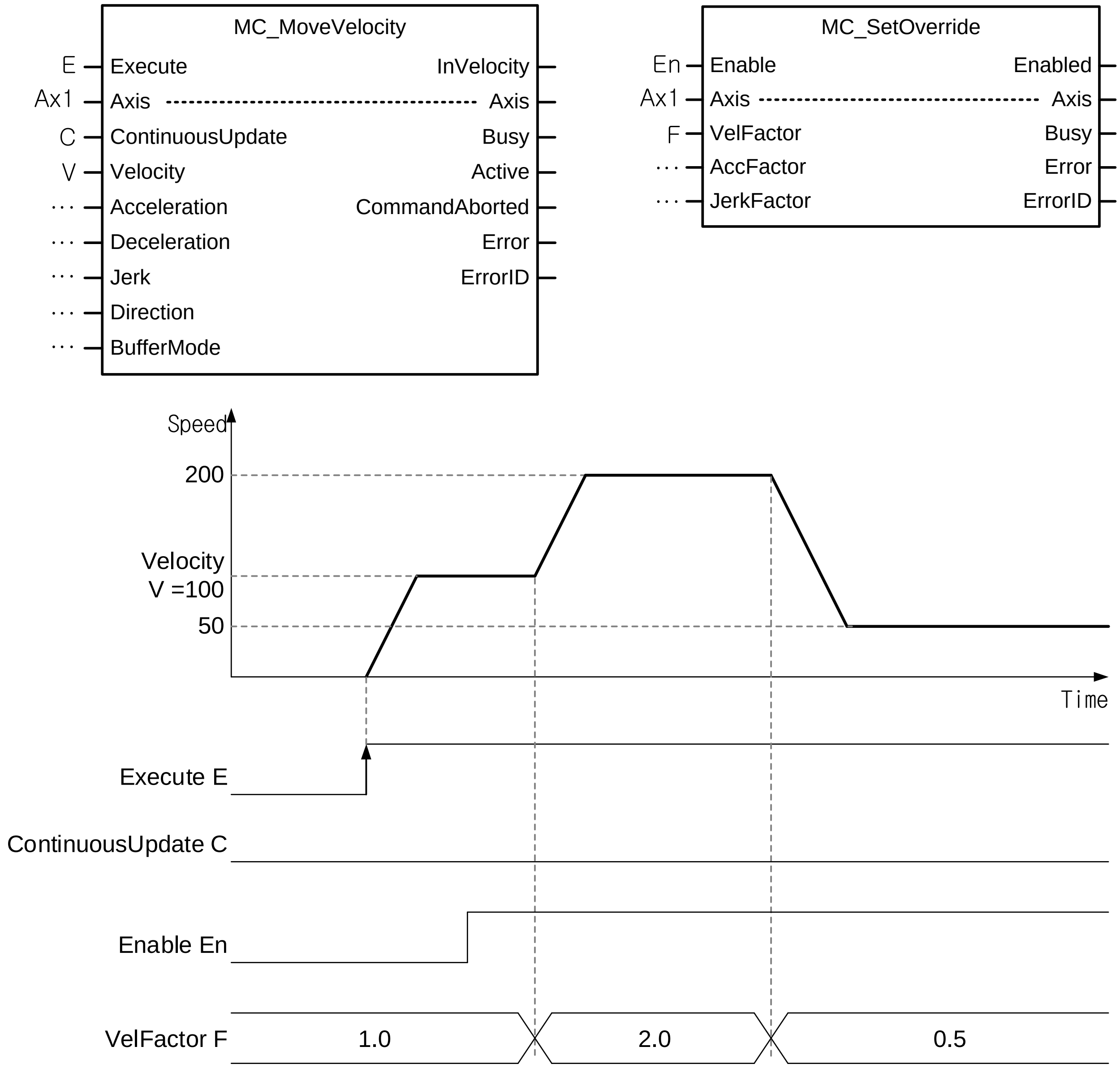
  1: unit value specified – Factor value is an absolute unit specified value of the set item

(d) Relevant motion function block

Name	Description	Operation condition
MC_SetOverride	Speed override	level



[Example] Changes in velocity using override (MC_SetOverride) motion function block



(4) Present Position Change

(a) It is a function to change the current position of the axis to the value specified by users.

(b) Specify the position in Position input. In case Relative input is Off state when command is executed, the position of the axis is replaced with the Position input value, and in case Relative input is On state, Position input value is added to the current position of the axis.

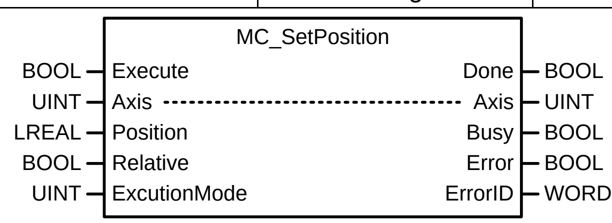
- 0: Absolute coordinate position
- 1: Relative coordinate position

(c) Set point can be specified with ExcutionMode input. When the input value is 0, the set value is set immediately after the execution of commands, and in case it is 1, it is set in the same time with ‘Buffered’ in a sequential operation setting.

- 0: Position value applied immediately
- 1: Applied at the same point with ‘Buffered’ of Buffermode

(d) Relevant motion function block

Name	Description	Operation condition
MC_SetPosition	Present position change	Edge



(5) Encoder preset

(a) It is a function to change the current encoder position value to any position value specified by users.

(b) In Encoder input, encoder to be changed is specified.

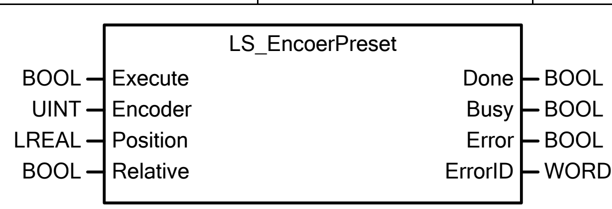
- 1: Encoder 1
- 2: Encoder 2

(c) In Position input, the encoder position is specified. In case Relative input is Off state when command is executed, the encoder position of the axis is replaced with the Position input value, and in case Relative input is On state, Position input value is added to the current position of the encoder.

- 0: Absolute coordinate position
- 1: Relative coordinate position

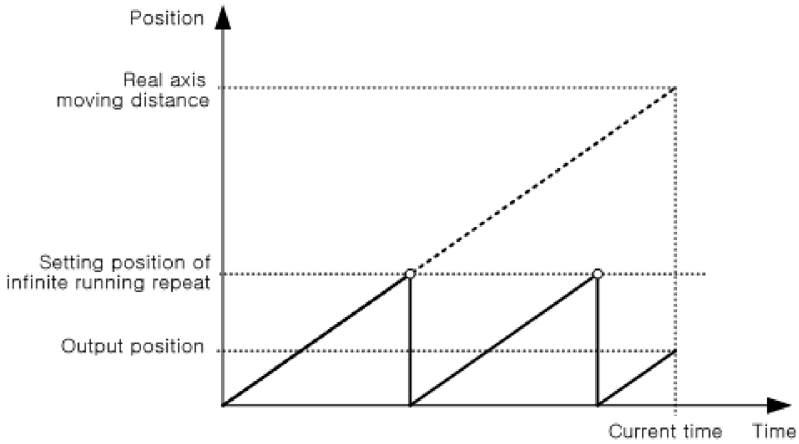
(d) Relevant motion function block

Name	Description	Operation condition
LS_EncoderPreset	Setting encoder current position	Edge



(6) Infinite running

(a) Infinite running repetition function is to perform periodic updates on the display values of the command position and current position automatically with values set in 'infinite running repetition position' among expansion parameters of operating parameters. The use of infinite running repetition positioning function makes it possible to determine the position with repeated position value on the same direction.



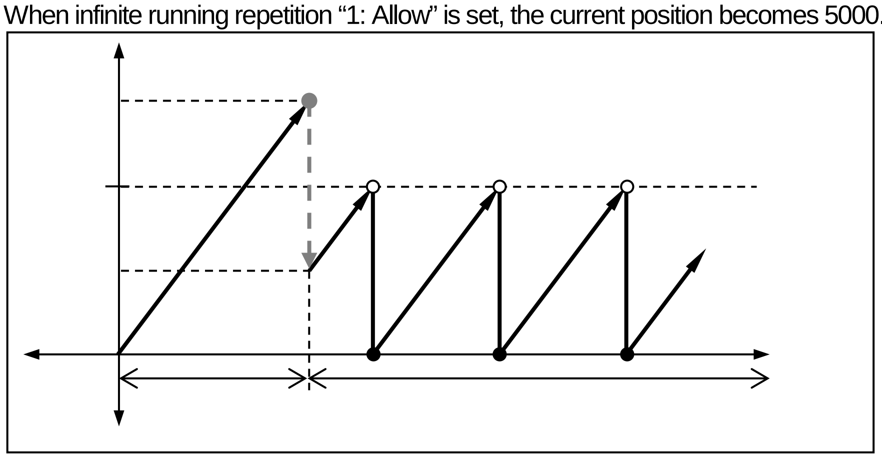
(b) The instant 'infinite running repetition' parameter among expansion parameters of operating parameters is set to Allow, the current position is automatically changed to value within the infinite running repletion position in case it is the value other than the range of infinite running repletion position.

[Example 1] In case the current position is -32100, and infinite running repetition position 10000

When infinite running repetition "1: Allow" is set, the current position becomes 7900.

[Example 2] In case the current position is 15000, and infinite running repetition position 10000

When infinite running repetition "1: Allow" is set, the current position becomes 5000.



(c) Infinite running repetition setting of driving axis can be made by using software package or axis parameter change function.

(d) Related parameter setting

Item	Content	Setting range	Initial value
Infinite running repeat position	Set repeated position range value in case of being used as infinite running repetition mode	Long real (LREAL) positive number	360 pls
Infinite running repeat	Set whether to allow infinite running repetition operation function.	0: disable 1:Enable	0: disable

8.3.2 Auxiliary Function of Control

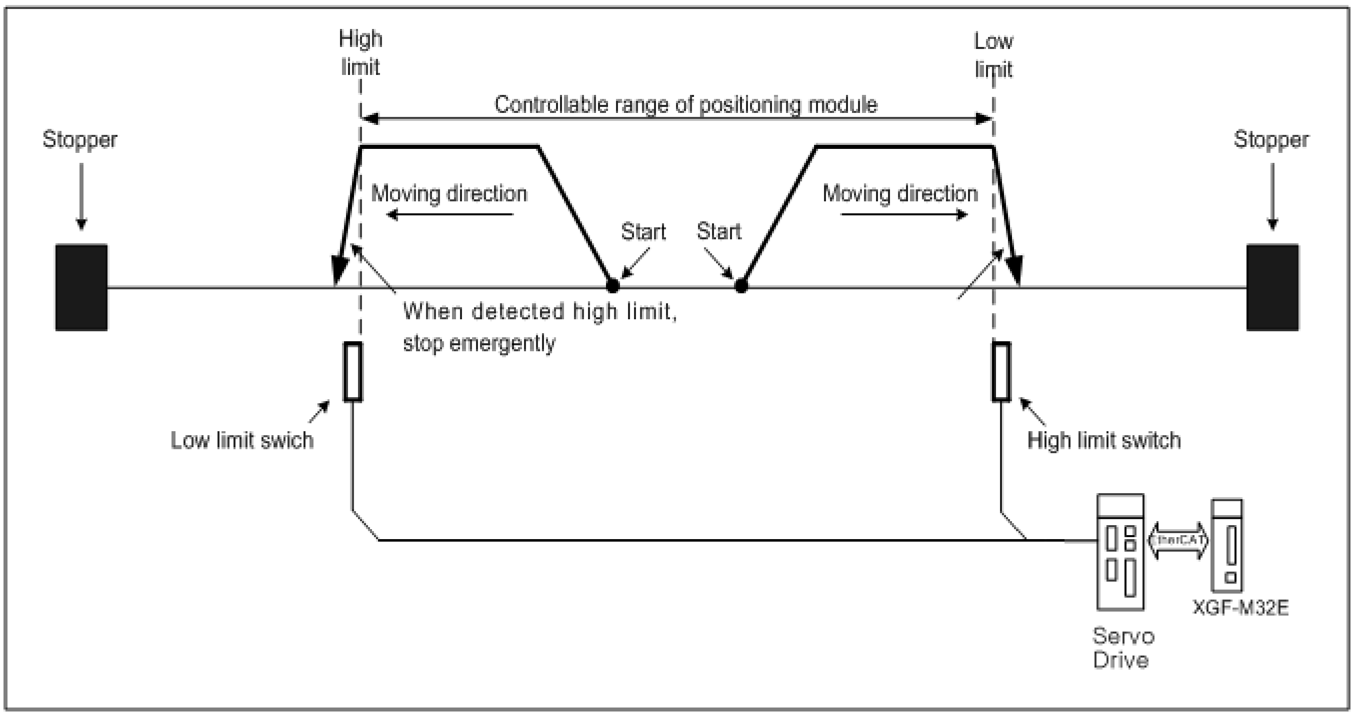
(1) hardware Upper/Lower Limits

(a) It is used to make a sudden stop of servo drive before reaching lower limit/upper limit of the machine side by installing high/low limit switch in the inside of the high/low limit, the physical operating range of the machine side. In this case, the range is out of the upper limit, error ‘0x1200’ occurs, and lower limit, error ‘0x1201’.

(b) Input of hardware high/low limit switch is connected to each servo drive, and operation is stopped by servo drive at the time of high/low limit detection, and module immediately terminates the motion which is currently being operated.

(c) In case of the stop due to the detection of hardware high/low limit signals, it is required to move inside the controllable range of motion control module with jog operation of the opposite direction of the detected signals.

(d) Hardware high/low limit motions are as follows.



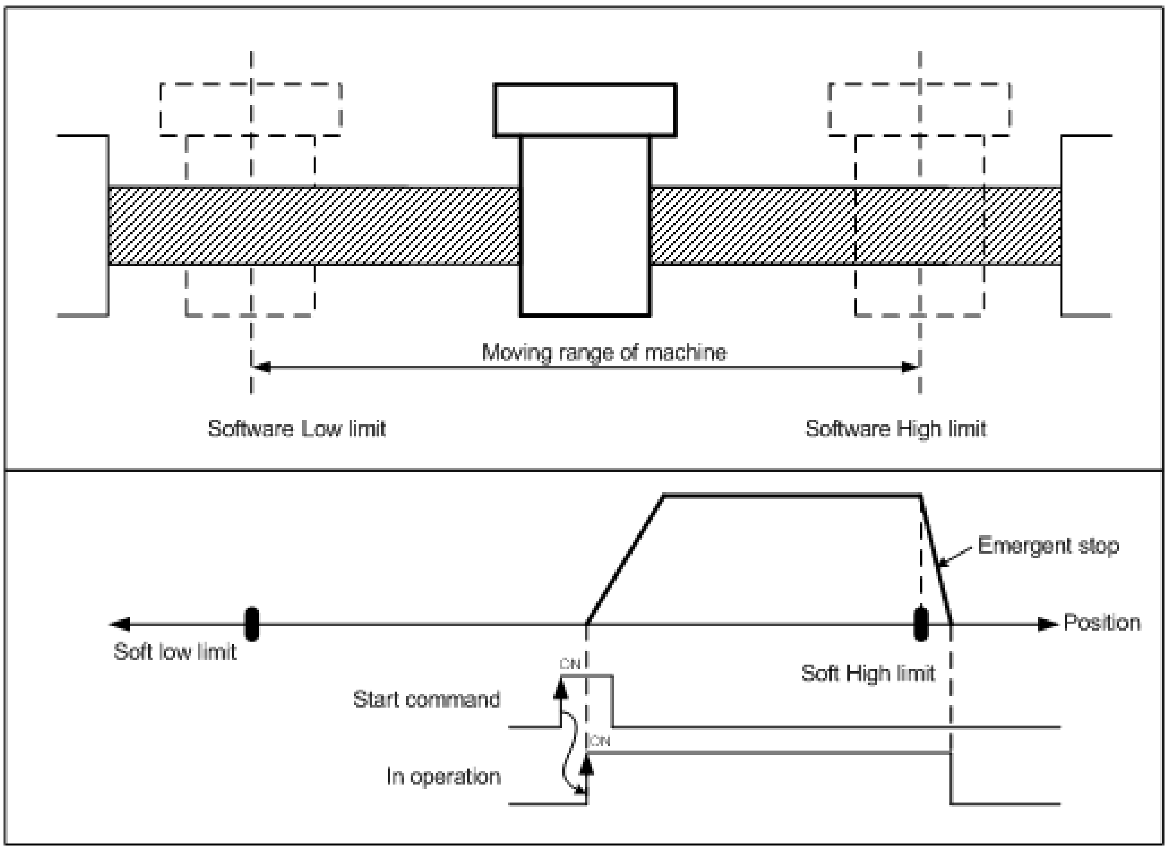
(2) Soft Upper/Lower Limits

(a) Software stroke high/low limit is a function that does not perform the operation in out of the range of soft high/low limit set by users.

(b) Software stroke high/low limit of each driving axis can be set by using software package or axis parameter change function.

(c) If the axis is outside the range of stroke, axis error occurs.

(d) When the axis is positioned outside the range of stroke, operation of the axis is impossible except for jog. Operation can be resumed by moving it inside the range of stroke through jog operation or resetting the current position to the inside the stroke range.



(e) Software high/low limit are don't detect in the following cases.

- If the soft upper limit and lower limit values are set as the maximum and minimum values based on a pulse position
  - a) If 'Position Control Range Expansion' = '0: Disable'
    - Soft upper limit: 2147483647, Soft lower limit: -2147483648
  - b) If 'Position Control Range Expansion' = '1: Enable'
    - Soft upper limit: 140737488355327, Soft lower limit: -140737488355328

  If 'Position Control Range Expansion' is set to '1: Enabled' and soft upper/lower limits are 2147483647 and -2147483648, the existing initial values, soft upper/lower limits are checked by these values.
- If the soft upper and lower limits are set to the same value (soft upper limit = soft lower limit)
- In case of the operation with speed control when expansion parameter "S/W limits during speed control" is set to " 0: Don't detect".

•

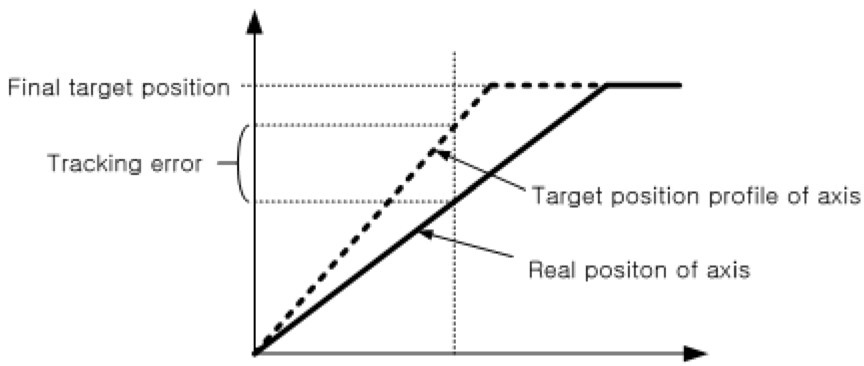
(f) Related parameter setting

Item	Content	Setting range	Initial value
S/W upper limit	Set the range of software limit function	Long real(LREAL)	2147483647 pls
S/W low limit			-2147483648 pls

(3) Position following error

(a) It is a function to output an error when driving axis is in position operation, or the actual position read from the axis is further beyond tracking tolerance than the target position of the position operation instruction profile.

(b) Position tracking tolerance of each driving axis can be set by using software package or axis parameter change function



(c) Whether to set abnormality to a warning or an alarm in case of the occurrence of tracking error can be set in Tracking Error Level of expansion parameter.

The operations according to the set values are as follows.

- '0: warning'
  When an error occurs in tracking error, the 「Over deviation warning (_AXxx_DEV_WARN)」flag is On, and warning error of tracking error (error code: 0x101D)) occurs. The axis continues to operate without stopping.
- '1: alarm'
  When an error occurs in tracking error, the 「Over deviation alarm (_AXxx_DEV_ERR)」flag is On, and the alarm error of tracking error (error code: 0x101C) occurs. The axis makes a sudden stop at 「Sudden stop deceleration」.

(d) In the following situations, the error in tracking error is not examined.

- In case the「Tracking error over-range value」is 0
- In case of the operation in homing or torque control

(e) Related parameter setting

Item	Content	Setting range	Initial value
Tracking error over-range value	Set the value to detect more than position deviation.	0 or Long real (LREAL) positive number	0
Tracking error level	Set the error level more than deviation.	0: warning 1: alarm	0: warning

(4) Latch (Touch Probe)

(a) It is a function to record the position of the axis when specific situation (Trigger event) occurs in the axis.

(b) Touch probe 1 and 2 can be selected to use according to trigger input (TriggerInput) settings.

- Trigger input (TriggerInput)=0: Latch function is performed when touch probe 1 signal is Off->On .
- Trigger input (TriggerInput)=1: Latch function is performed when touch probe 2 signal is Off->On .
- Trigger input (TriggerInput)=2: Latch function is performed when touch probe 1 signal is On->Off .
- Trigger input (TriggerInput)=3: Latch function is performed when touch probe 2 signal is On->Off .
- Trigger input (TriggerInput)=4: Latch function is performed when encoder index signal is Off->On .
- Trigger input (TriggerInput)=5: Latch function is performed when encoder index signal is Off->On .

(Items 0 to 1 can use functions in all touch probe commands [MC_TouchProbe, MC_TouchProbeEx], and items 2 to 5 can use functions in extended touch probes [MC_TouchProbeEx].)

(c) Depending on the trigger mode setting, you can set whether to record the trigger signal only once or every time a signal is generated continuously.

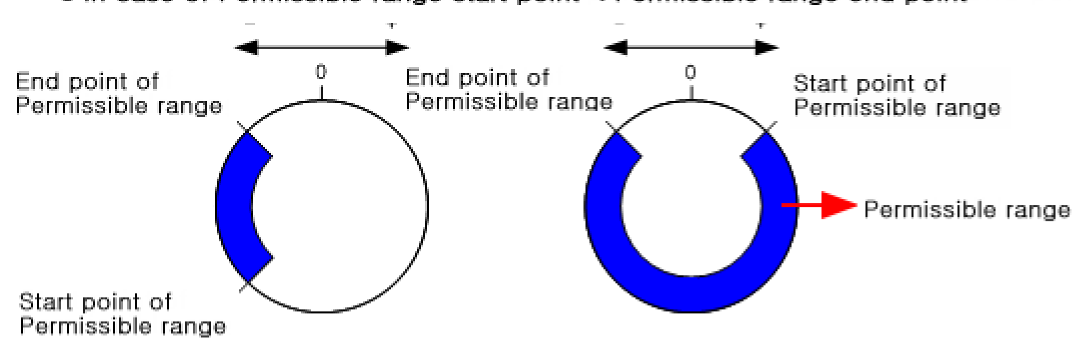
- Trigger Mode = 0: When the signal set in the trigger input occurs, the latch function is executed once.
- Trigger Mode = 1 : Every time the signal set in the trigger input occurs, the latch function is executed If the trigger mode is set to continuous trigger (1), the latch can be stopped with the MC_AbortTrigger command.

(Function is available in Extended Touch Probe [MC_TouchProbeEx].)

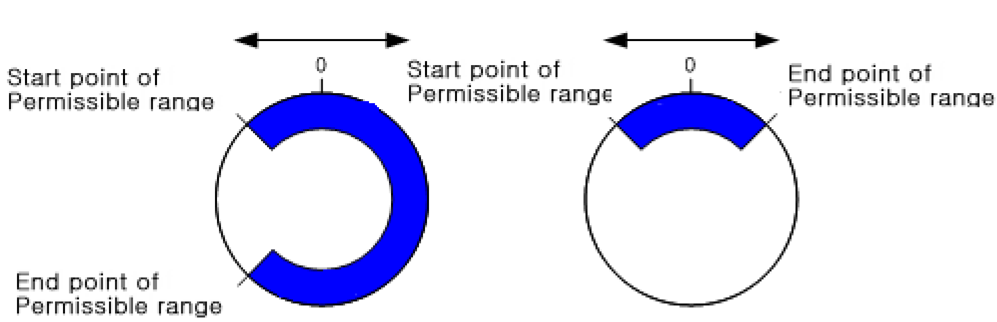
(d) Specify the area where the latch (touch probe) function operates.

- If you specify the allowable area, it operates only within the designated area.
- In the case of infinite length repeat operation (rotary axis), the relationship of the latch (touch probe) operation area according to the start and end positions of the allowable area is as follows.

● In case of Permissible range start point < Permissible range end point



● In case of Permissible range start point > Permissible range end point

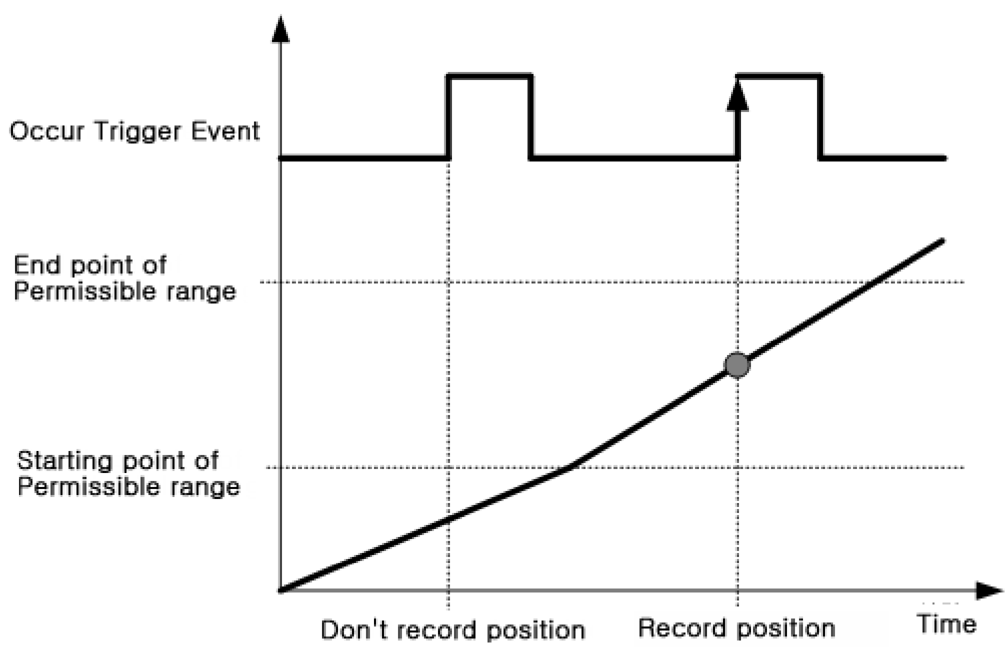


(e) To use the latch (touch probe) function, the following objects must be included in the PDO setting of the slave parameter.

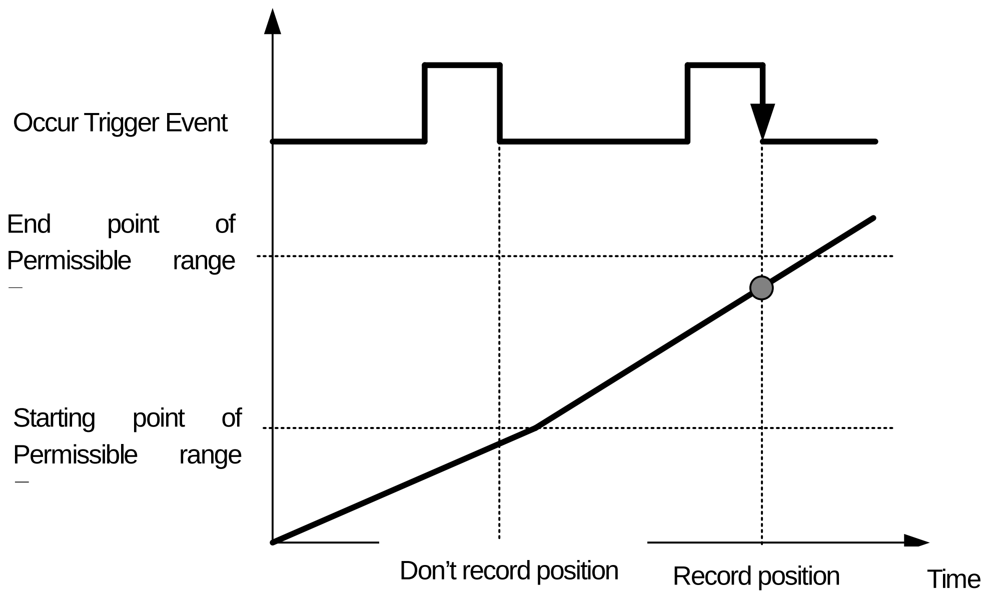
Trigger input	RxPDO	TxPDO
Touch probe 1 rising edge	0x60B8:0 Touch probe function	0x60B9:0 Touch probe Status 0x60BA:0 touch probe 1 forward direction position values
Touch probe 2 rising edge	0x60B8:0 Touch probe function	0x60B9:0 Touch probe Status 0x60BC:0 touch probe 2 forward direction position values
Touch probe 1 falling edge	0x60B8:0 Touch probe function	0x60B9:0 Touch probe Status 0x60BB:0 touch probe 1 reverse direction position values
Touch probe 2 falling edge	0x60B8:0 Touch probe function	0x60B9:0 Touch probe Status 0x60BD:0 touch probe 2 reverse direction position values
Touch probe 1 index pulse	0x60B8:0 Touch probe function	0x60B9:0 Touch probe Status 0x60BA:0 touch probe 1 forward direction position values
Touch probe 2 index pulse	0x60B8:0 Touch probe function	0x60B9:0 Touch probe Status 0x60BC:0 touch probe 2 forward direction position values

In case there are not above objects, an error (error code: 0x10E0) occurs when latch (touch probe) command is used.

(f) Operation Timing



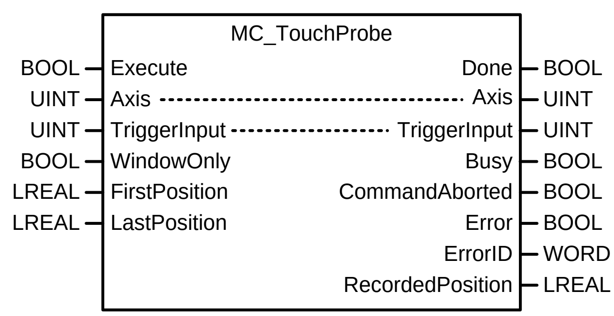
< When TriggerInput is set to touch probe rising edge (0 to 1) or encoder index pulse (4 to 5)>



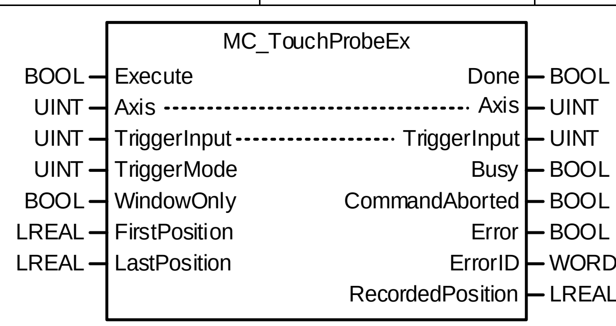
< When TriggerInput is set to touch probe falling edge (2~3)>

(g) Relevant motion function block

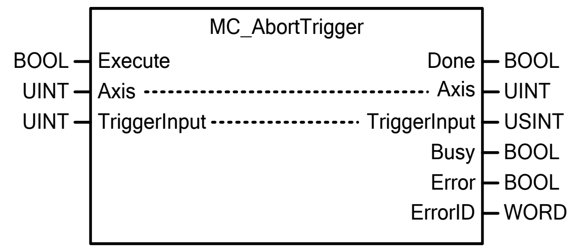
Name	Description	Operation condition
MC_TouchProbe	Touch probe	Edge



Name	Description	Operation condition
MC_TouchProbeEx	Expansion Touch Probe	Edge



Name	Description	Operation condition
MC_AbortTrigger	Abort trigger events	Edge



(5) Error reset monitoring

(a) When resetting an error occurred in the servo drive, the monitoring time can be set.

(b) In case an error occurs in servo drive at the time of resetting error that occurs in the axis due to error reset commands, whether servo drive error is properly reset can be verified by setting error reset monitoring time.

(c) If monitoring time is exceeded, error reset is not executed any more even if the error of the drive is not reset.. In this case, a servo drive error reset timeout error (error code: 0x1070) occurs.

(d) Error reset monitoring time of each driving axis can be set by using software package or axis parameter change function.

(e) Related parameter setting

Item	Content	Setting range	Initial value
Error reset monitoring time (error reset timeout)	Set the monitoring time in case of resetting error that occurs in servo drive	1 ~ 1000 ms	100 ms

8.3.3 Data manager function

(1) Parameter manager

(a) It is a function to read or change axis parameters stored in the module.

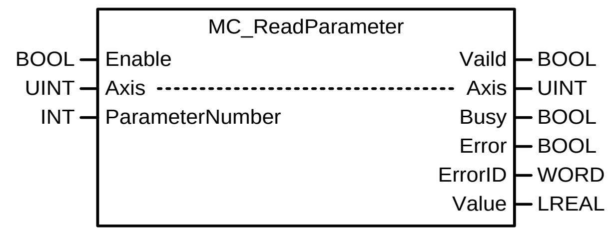
(b) It can change desired parameter values by specifying axis number and corresponding parameter number.

(c) Parameter value modified with parameter-write function is automatically stored in backup.ram in case there is no error.

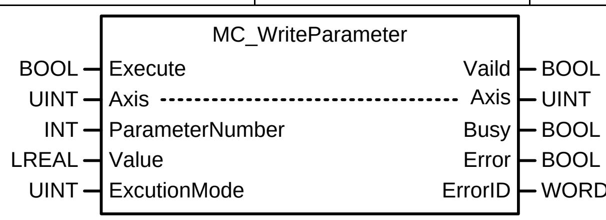
(d) For parameters to be set in "ParameterNumber", refer to the motion function block item.

(e) Relevant motion function block

Name	Description	Operation condition
MC_ReadParameter	Parameter Read	level



Name	Description	Operation condition
MC_WriteParameter	Parameter write	Edge



(2) Cam data manager

It is able to read and change the cam data in program by the cam data Cam data reading/writing command

(a) Read Cam data

(ㄱ) CmDataRead command reads the cam profile data designated by CamTable ID when Enable input is enabled and saves the data to the data area specified as MasterPoint and SlavePoint.

(ㄴ) The first address of the variables to store "Main-axis Position" and "Sub-axis Position" read from the camp profile is set at the MasterPoint and the SlavePoint. In CamCurveSel, set an array variable to store the 'interpolation type' value.

(ㄷ) If the size of the array variable set in MasterPoint/ SlavePoint/ CamCurveSel is smaller than the number of cam points, cam data is read as much as the size of the array, so the entire cam table data may not be read.

(b) Write CAM data

(ㄱ) When the Execute input is On, the cam data write command writes the values set in StartSlope and EndSlope and the number set in CamPointNum to the camp profile designated by CamTalble ID, as well as the values of the array variables set in MasterPoint and SlavePoint to 'main axis position' and 'sub axis position'. Write as a value and write the value of the set array variable to CamCurveSel as the 'interpolation type' value.

(ㄴ) If the size of the array variable set in MasterPoint / SlavePoint / CamCurveSel is set to a value smaller than CamPointNum, "Error 16#000B" occurs.

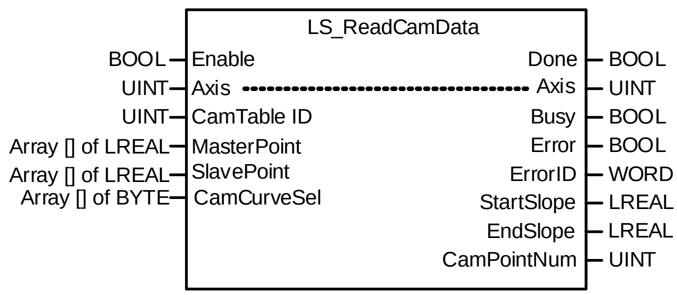
(ㄷ) CamTableID input can be set to between 1 and 32. Setting a value outside the above range will cause "Error 16#000B"

(ㄹ) In CamPointNum, less than the number of settings set in the cam profile can be entered. If a value outside the range is set, "Error 16#111C" occurs.

(c) Relevant motion function block

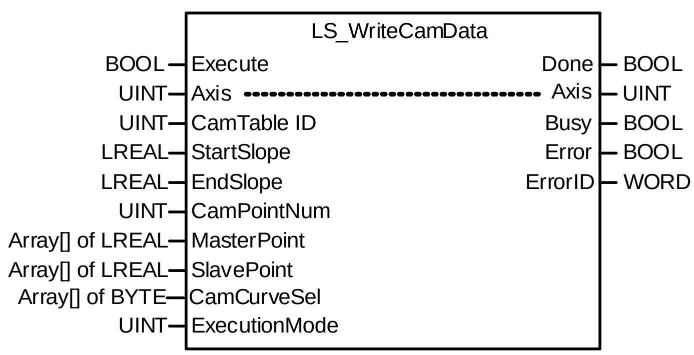
(ㄱ) Read Cam data

Name	Description	Operation condition
LS_ReadCamData	Read Cam data	level



(ㄴ) Write CAM data

Name	Description	Operation condition
LS_WriteCamData	Write CAM data	Edge



(3) SDO parameter management

(a) This function reads or changes SDO parameters of slave devices connected via network.

(b) Parameter values for a certain axis number and the corresponding object number can be read or changed. Parameter number is specified by Index and SubIndex. Parameter size is specified by Length

(c) Index input can be set as follows. If it is not set as the setting value, "Error 0x1F12" occurs.

Setting Value	Content
16#0000 ~ 16#0FFF	Data Type Description
16#1000 ~ 16#1FFF	Communication objects
16#2000 ~ 16#5FFF	Manufacturer Specific Profile Area
16#6000 ~ 16#9FFF	Standardized Device Profile Area

(d) The value between 0~255 can be entered in SubIndex, and if the value is set outside the range, "error 0x1F12" occurs.

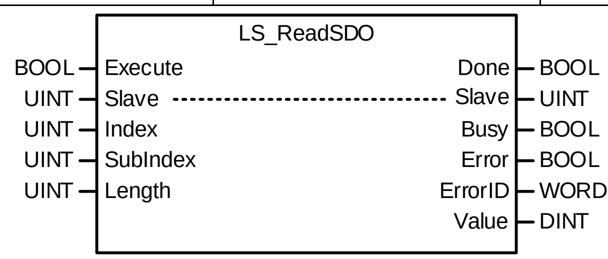
(e) In Length, values ranging from 1 to 4 can be entered, which mean 1 to 4 bytes. If the value is set outside the range, "error 0x1F12" occurs.

(f) The parameter values changed by SDO write function are not automatically stored to the ROM of the slave device. In order to store the changed parameters to the servo drive EEPROM, please use SDO Save command.

(g) Relevant motion function block

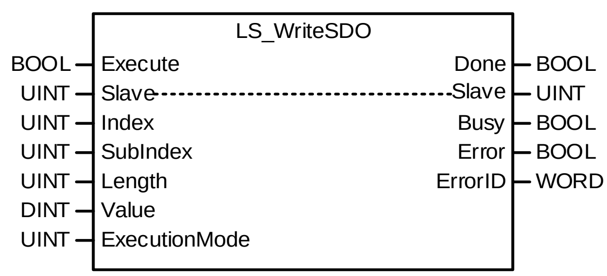
(ㄱ) Read from SDO

Name	Description	Operation condition
LS_ReadSDO	Read from SDO	level



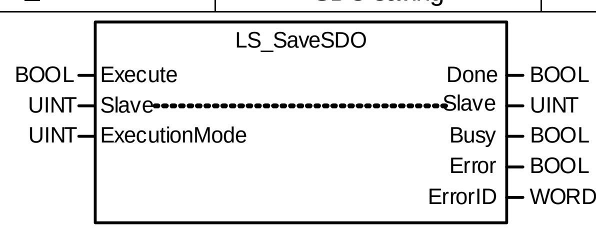
(ㄴ) Write to SDO

Name	Description	Operation condition
LS_WriteSDO	Write to SDO	Edge



(ㄷ) SDO saving

Name	Description	Operation condition
LS_SaveSDO	SDO saving	Edge

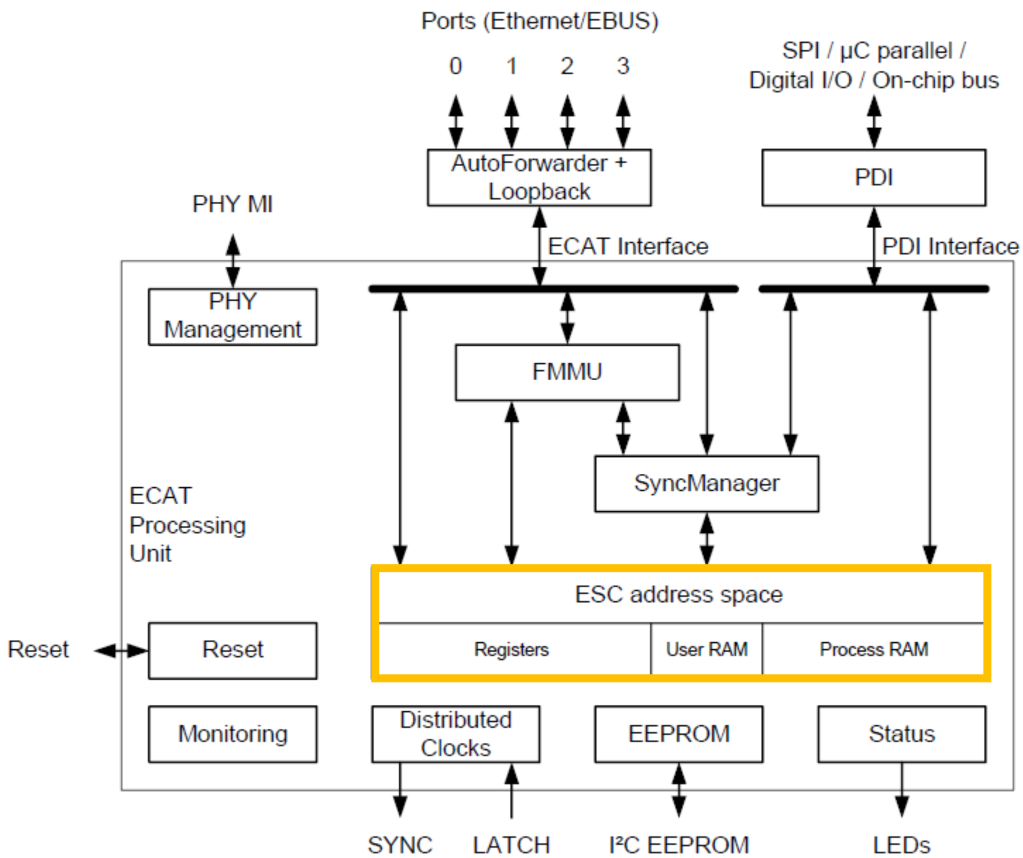


8.3.4 EtherCAT Communication Diagnosis Function

EtherCAT slave devices performsEtherCAT communication using ASIC, FPGA, or EtherCAT Slave Controller (ESC) included in the standard micro controller. The communication diagnosis function of EtherCAT reads and writes the ECS (EtherCAT Slave Controller) registers and memories of the slave device, allowing the user to check EtherCAT communication status and errors. EtherCAT communication diagnosis function can be used whether communication is normal or disconnected.

(1) EtherCAT Slave Controller (ESC)

(a) ECS is configured as follows. Diagnosis function commands can be used to read and write in the ESC address spaces shown in the block diagram below.



(b) ESC address space is configured as follows.

Address	Definition	Note
0x0000	ESC Register	ESC Information,
:		FMMU, SyncManager,
0x0FFF		Distributed Clocks (DC).
0x1000	Process Data RAM	Digital I/O Input Data,
:		Process Data RAM (1KB ~ 60KB)
0xFFFF		

※ Please refer to EtherCAT Slave Controller (ESC) data sheet for detailed information on register and Process RAM.

(2) Read from ESC

(a) This function reads data in ESC of the slave devices connected via network.

(b) Adp input specifies the EtherCAT slave device address, and the following values can be set depending on EcatCmd settings. If EcatCmd setting is 7(BRD), Adp input value is ignored.

EcatCmd	Adp range
1 (APRD)	0x0000: The first slave connected 0xFFFF: The second slave connected 0xFFFE: The third slave connected : 0xFFC1: The 64th slave connected
4 (FPRD)	1 ~ 64: slave 1~slave 64
7 (BRD)	-

(c) In Length, values ranging from 1 to 4 can be entered, which mean 1 to 4 bytes.

(d) At EcatCmd, the type of command to use when reading ESC (EtherCAT Slave Controller) is specified. The following three commands can be used.

(ㄱ) 1 - APRD (Auto Increment Physical Read)

This command is used when reading the slave device data following the order of physical connection before normal communication connection by the master. A slave device receiving Adp with 0 value will read data of the size designated by Length. Adp of each slave device increases when EtherCAT frame is received. For example, if EcatCmd is 1, and Adp is set to 0xFFFF, when executing ESC read function block, reading is not performed because the Adp at the time of receiving EtherCAT frame from the first slave device is not 1, only increasing Adp by 1. When the second slve device receives EtherCAT frame, reading is performed because the Adp value of the first slave value increased by 1 to 0. The Adp setting values depending on the slave device connection order are as follows.

Slave device	Setting value
The first slave connected	0
The second slave connected	0xFFFF
:	:
The 64th slave connected	0xFFC1

(ㄴ) 4 - FPRD (Configured Address Physical Read)

This order is used to read the data by designating the station address of the slave device after normal communication connection by the master. If the Station Address of the slave device set by EtherCAT master matches the transmitted Adp, the slave device reads data of the size designated by Length in the Ado area. The Station Address of slave device set by master can be checked in slave information dialog box when the slave is added.

(ㄷ) 7 – Read Broadcast (Broadcast Read)

All connected slave devices read data of the size set by Length in the Ado area, and saves the result after Bitwise-OR. The designated address value at Adp is ignored, and Wkc increase by 1due to all slaves that performed normal read operation

(e) Value and Wkc(Working Counter) is displayed as 0 when the motion function block is executed. When the execution is completed (Done output is on), the read data value is displayed at Value, and the Working Counter value is displayed at Wkc.

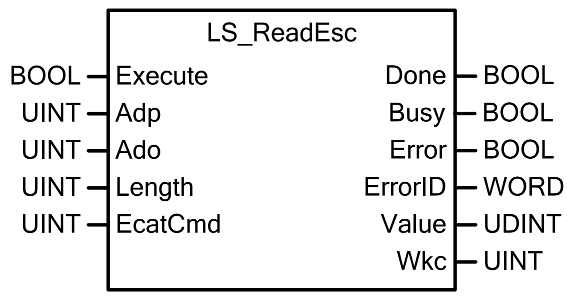
(f) Wkc stands for Working Counter. If data is successfully read at the designated slave device, it increases by 1. If EcatCmd is 7(BRD), it increases by 1 due to all slaves that performed normal read operation.

(g) After the execution of ESC read command, if normal data read operation is executed from the designated slave device, Done output is on.

(h) ESC read command and ESC write command cannot be simultaneously executed. If they are executed at the same time, the command of the program last executed is executed, and an error (0x1021) occurs in the preceding command.

(i) Relevant motion function block

Name	Description	Operation condition
LS_ReadEsc	Read from ESC	Edge



(j) In the following cases, ESC reading cannot be performed due to errors, properly.

(ㄱ) No slave device is connected to module (Error Code: 0x0F09)

(ㄴ) Adp setting value is outside the range (Error Code: 0x0F60)

(ㄷ) Length setting value is outside the range (Error Code: 0x0F61)

(ㄹ) EcatCmd setting value is outside the range (Error Code: 0x0F62)

(ㅁ) No response to ESC read command (Error Code: 0x0F63)

(3) Write to ESC

(a) This function writes data in ESC of the slave devices connected via network.

(b) Adp input specifies the EtherCAT slave device address, and the following values can be set depending on EcatCmd settings. If EcatCmd setting is 8(BWR), Adp input value is ignored.

EcatCmd	Adp range
2 (APWR)	0x0000: The first slave connected 0xFFFF: The second slave connected 0xFFFE: The third slave connected : 0xFFC1: The 64th slave connected
5 (FPWR)	1 ~ 64: slave 1~slave 64
8 (BWR)	-

(c) In Length, values ranging from 1 to 4 can be entered, which mean 1 to 4 bytes.

(d) At EcatCmd, the type of command to use when reading ESC (EtherCAT Slave Controller) is specified. The following three commands can be used.

(ㄱ) 2- APWR(Auto Increment Physical Write)

This command is used when reading the slave device data following the order of physical connection before normal communication connection by the master. A slave device receiving Adp with 0 value will read data of the size designated by Length. Adp of each slave device increases when EtherCAT frame is received. For example, if EcatCmd is 2, and Adp is set to 0xFFFF, when executing ESC read function block, reading is not performed because the Adp at the time of receiving EtherCAT frame from the first slave device is not 0, only increasing Adp by 1. When the second slave device receives EtherCAT frame, writing is performed because the Adp value of the first slave value increased by 1 to 0. The Adp setting values depending on the slave device connection order are as follows.

Slave device	Setting value
The first slave connected	0
The second slave connected	0xFFFF
:	:
The 64th slave connected	0xFFC1

(ㄴ) 5 – FPWR(Configured Address Physical Write)

This order is used to write the data by designating the station address of the slave device after normal communication connection by the master. If the Station Address of the slave device set by EtherCAT master matches the transmitted Adp, the slave device writes data of the size designated by Length in the Ado area. The Station Address of slave device set by master can be checked in slave information dialog box when the slave is added.

(ㄷ) 8 – Write Broadcast BWR, Broadcast Write)

All connected slave devices write data of the size set by Length in the Ado area. The designated address value at Adp is ignored, and Wkc increase by 1 due to all slaves that performed normal write operation.

(e) Wkc value is displayed as 0 when the motion function block is executed, and the Working Counter value is displayed when execution is completed (Done output is on). Wkc increases by 1 through each slave device designated by EcatCmd and Adp.

(f) Wkc stands for Working Counter. If data is successfully written at the designated slave device, it increases by 1. If EcatCmd is 8(BWR), it increases by 1 through each slave device that performed normal write operation.

(g) After the execution of ESC write command, if normal data write operation is executed in the specified slave device, Done output is on.

(h) Slave devices use ESC to perform EtherCAT communication. Therefore, changing ESC register values while executing connection/disconnection command or during normal EtherCAT communication may prevent the slave device from maintaining existing motions or cause communication errors. Therefore, using the following ESC Register causes an error without executing write motion. (Error code: 0x0F74)

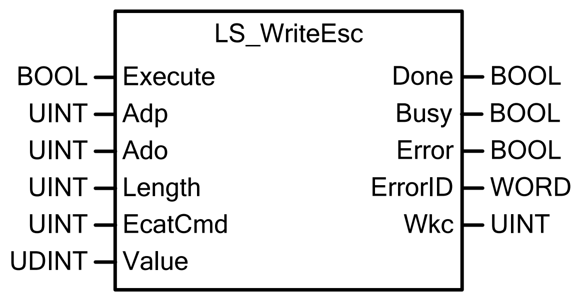
Ado range	Definition
0x0010 ~ 0x0011	Configured Station Address
0x0020 ~ 0x0021	Write Protection
0x0030 ~ 0x0031	
0x0040	ESC Reset ECAT
0x0100 ~ 0x0103	DL Control
0x0120 ~ 0x0121	AL Control
0x0600 ~ 0x06FF	FMMU
0x0800 ~ 0x087F	SyncManager
0x0900 ~ 0x09FF	Distributed Clocks

0x0120 (AL Control) register can be written after the connection of normal communication, not the execution of connection/disconnection command.

(i) ESC read command and ESC write command cannot be simultaneously executed. If they are executed at the same time, the command of the program last executed is executed, and an error (0x1021) occurs in the preceding command.

(j) Relevant motion function block

Name	Description	Operation condition
LS_WriteEsc	Write to ESC	Edge



(k) In the following cases, ESC writing cannot be performed due to errors, properly

(ㄱ) No slave device is connected to module (Error Code: 0x0F09)

(ㄴ) Adp setting value is outside the range (Error Code: 0x0F70)

(ㄷ) Length setting value is outside the range (Error Code: 0x0F71)

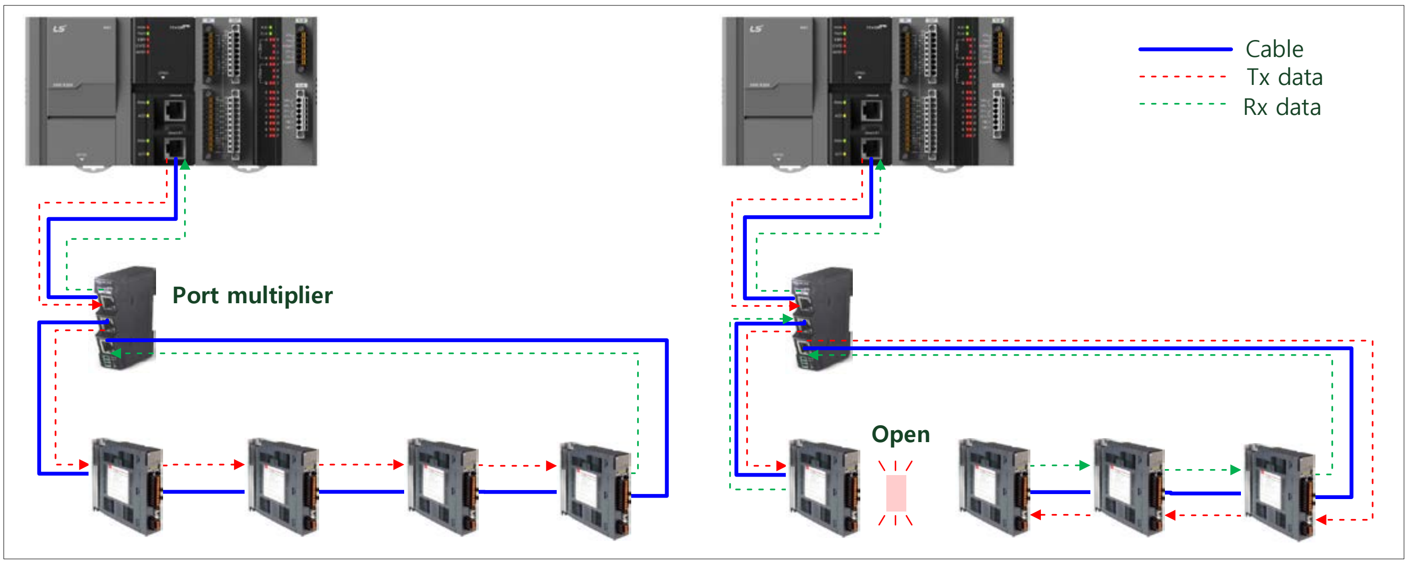
(ㄹ) EcatCmd setting value is outside the range (Error Code: 0x0F72)

(ㅁ) No response to ESC read command (Error Code: 0x0F73)

(ㅂ) Ado setting value is not correct (Error Code: 0x0F74)

8.3.5 Cable Duplication function

It provides cable duplication function using port multiplier. Constructing a ring topology using port multiplier will prevent the network between slaves from disconnecting even in case of a cable disconnection on one side. When the disconnected cable is re-connected, it is recovered to the original communication method.

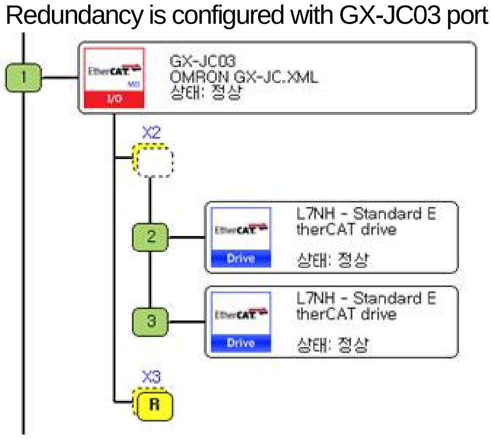


Port multiplier of up to 1 can be used. In case of using a port multiplier, it occupies an IO slave number. Therefore, a caution is required when using since the use of the port multiplier reduces the number of IO slaves available.

(1) EC_LINE_FAIL (cable disconnection) flag disconnection diagnosis

The EC_LINE_FAIL flag can be used to diagnose disconnection during redundancy configuration. In the case of a port multiplier, all ports are checked and the EC_LINE_FAIL flag is On when disconnection is detected in the port. In case of general slave, EC_LINE_FAIL flag is On only if there is a slave whose A port is disconnected.

(a) Redundancy is configured with GX-JC03 port multiplier and 2 L7NH slaves.



(b) The operation of the EC_LINE_FAIL flag according to the condition is as follows.
_EC_LINE_FAIL[0] : port multiplier
_EC_LINE_FAIL[1] : slave 2
_EC_LINE_FAIL[2] : slave 3

No	Condition	_EC_LINE_FAIL[0]	_EC_LINE_FAIL[1]	_EC_LINE_FAIL[2]
1	Only X3 is disconnected	On	Off	Off
2	Only X2 is disconnected	On	On	Off
3	Both X2 and X3 are disconnected	On	On	On

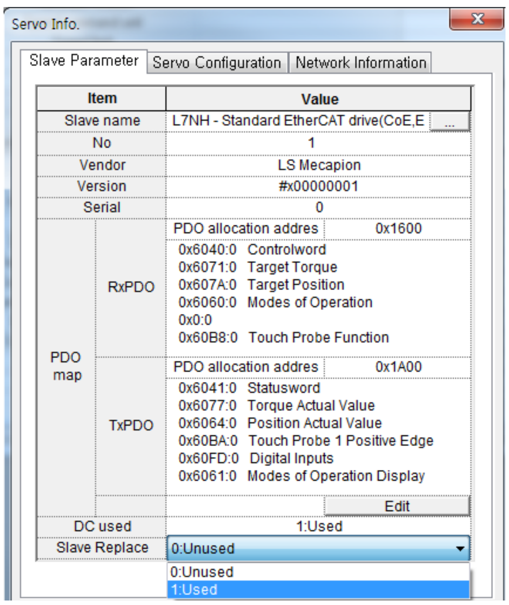
- When only X3 is disconnected, _EC_LINE_FAIL[0] of the port multiplier is ON and other LineFail flags are not ON because port A among general slaves (Slave 2, Slave 3) is not disconnected.
- When only X2 is disconnected, _EC_LINE_FAIL[0] of the port multiplier is ON and _EC_LINE_FAIL[1] is ON because port A of slave No. 2 is ON.
- If both X2 and X3 are disconnected, _EC_LINE_FAIL[0] of the port multiplier is turned on, and _EC_LINE_FAIL[1], _EC_LINE_FAIL[2] are turned on because all the port status information of the slave cannot be read.

8.3.6 Replacement during connection

While using the cable duplication function, if a slave device previously not in operation due to network disconnection or a failure is restored and connected to the network, this function detects the connection and connects to the network of the individual slave device without having to reconnect the overall network.

(2) Replace function during connection setting

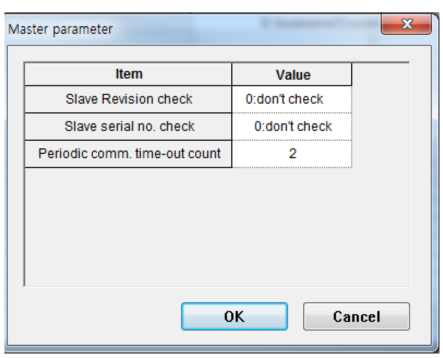
To set the slave device to use the function to replace slaves during connection, the "In-connection Replacement Function" at the slave information should be set to Enabled. In case of a slave for which the replacement function is not set to use, if the slave is removed from the network, the removal is regarded as a network error, which stops the entire network.



(3) Master parameter setting

When using the in-connection replacement function, the slaves being replaced should be identical to the replacing slave. To determine whether the slaves being replaced are identical, check whether the manufacturer/product codes match. In addition, check whether the revision/serial numbers are identical, depending on the master parameter settings.

.



(4) The way of slave exchange during connection

(a) Remove the input/output cable of the slave to be replaced during the network connection.

(b) Shut off the power of the slave.

(c) Apply the power to the slave to be replaced.

(d) Connect the cable of one side of the port.

(The simultaneous connection of input/output cables may prevent normal replacement.)

(d) When the slave communication is restored, connect the cable of the other side.

8.3.7 Encoder Position Latch Function

If using the encoder position latch function, you can use the encoder value that is not affected by the program task cycle through the built-in input by latching the value.
Encoder latch operates through the input of the built-in input channel 1 (%IX0) and channel 2 (%IX1).

(1) Parameter setting

To use the encoder position latch function, setting of the 'Encoder - Encoder 1/2 Position Latch' parameter is needed. This parameter sets whether to use the 'Encoder 1/2 Position Latch' function as follows:

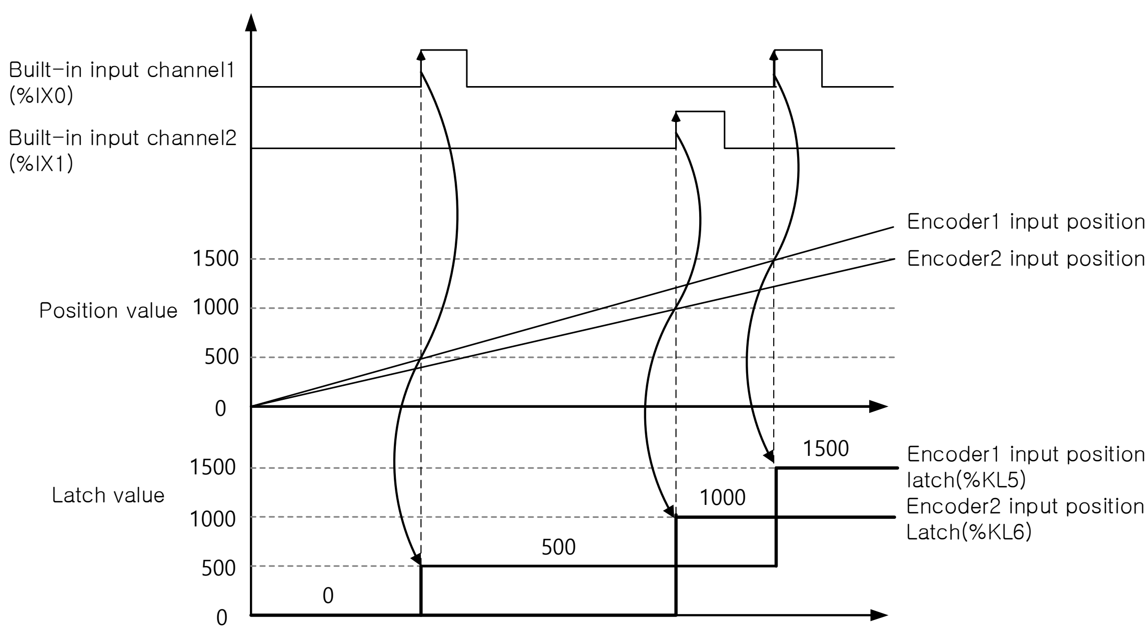
Item	Setting range	Initial values
Encoder 1 Position Latch	0: Unused 1: used	0: Unused
Encoder 1 Position Latch	0: Unused 1: used	0: Unused

(2) Trigger condition

Input condition	Latch values	Flag name (Device)
Built-in input 1 channel (%IX0) input rise	Encoder 1 Input Position Latch	_ENC1_POS_LATCH (%KL5)
Built-in input 2 channel (%IX1) input rise	Encoder 2 Input Position Latch	_ENC2_POS_LATCH (%KL6)

The relative operation is as follows:

If a signal is input into the built-in input channel 1, the position value is latched in the rising edge and saved as the encoder 1 input position latch value.
If a signal is input into the built-in input channel 2, the position value is latched in the rising edge and saved as the encoder 2 input position latch value.



8.3.8 Position Control Range Expansion

When executing the position control among motion functions, the target position value can be set in the range of 32-bit integer types based on the pulse position. However, if the resolution of a motor encoder is high or the transfer distance of the machine is long, position control is sometimes needed to exceed the range of 32-bit integer types.

In such case, the 'Position Control Range Expansion' function can position control by expanding the range of position control to the range of 48-bit integer types.

(1) Parameter setting

To use the 'Position Control Range Extension' function, you need to set the 'Axis Parameter – Basic Setting – Position Control Range Extension' parameter. This parameter sets whether to use the 'Position Control Range Extension' function as follows.

Item	Setting range	Initial values
Position Control Range Expansion	0: Unused 1: used	0: Unused

(2) Position Control Range

An error occurs when a position exceeds the position control range after conversing the unit position set to LREAL into the pulse unit when specifying the target position in motion control commands.

The range of position control according to whether to use the 'Position Control Range Expansion' function is as follows:

Item	Setting value	Position Control Range
Position Control Range Expansion	0: Unused	Integer 32bit ➔ -2^{31} ~ $2^{31}-1$ (-2,147,483,648 ~ 2,147,483,647)
	1: used	Integer 48bit ➔ -2^{47} ~ $2^{47}-1$ (-140,737,488,355,328 ~ 140,737,488,355,327)

If using the 'Position Control Range Expansion' function, the range of position control is expanded 65,536 times more than previously.

(3) Software Upper/Lower Limits

If 'soft upper/lower limit' parameters are set to the initial value (the soft upper limit = 2147483647 and the soft lower limit = -2147483648), operation changes according to 'Position Control Range Expansion' parameters as follows:

1) If 'Position Control Range Expansion' = '0: Disable'
   - If the soft upper limit is 2147483647 and the soft lower limit is -2147483648, the software upper/lower limits are not detected.
2) If 'Position Control Range Expansion' = '1: Enable'
   - If the soft upper limit is 2147483647 and the soft lower limit is -2147483648, the software upper/lower limits are examined with these values. When the position control exceeds these values, an axis stops suddenly after the software upper/lower limit error occurs.

※ If the soft upper limit and lower limit are set to the same value, the software upper/lower limits are not detected irrespective of the 'Position Control Range Expansion' setting.

(4) Restrictions

When the position value is specified for the next item, the position range after being converted into pulse unit is limited to the '32-bit integer type' value.

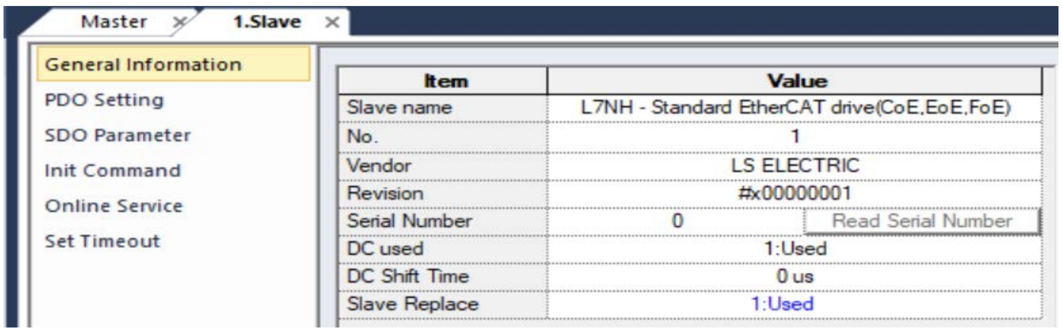
1) Transfer distance per 1 rotation
2) Infinite running repeat position
3) Command in-position range
4) Excessive following error value
5) Current pos. compensation amount
6) Phase compensation amount of the phase compensation command
7) Target distance of the SuperImposed operation command
8) If the linear interpolation, TransitionMode is TMCornerDistance, the value of TransionParameter

8.3.9 Connection function less than the set number

In the standalone motion controller, if the slave whose replacement function is set to '1: Use' during connection is not connected, the unconnected slave can be restored using the replacement function during connection in the future, and EtherCAT connection of the remaining connected slaves is allowed.

(1) Connection function setting less than the number of settings

When using less than the set number of connections, set the replacement function during connection of the slave that will allow less than the set number of connections to '1: Use'. If a slave whose replacement function is set to '0: not used' during connection is not connected, EtherCAT connection is not performed.



(2) Less than set number of connection function available version

In the version that does not support the connection function less than the set number, even if the replacement function is set to '1: Use' during slave connection, if it is not connected when performing EtherCAT connection, EtherCAT connection is not performed.

Item / Product name	Module O/S	XG5000
XMC-E32A XMC-E16A XMC-E08A XMC-E32C	V1.70	-

8.3.10 Node switch operation mode setting

In the standalone motion controller, the function to set the station number of the slave through the node switch is provided. When performing automatic connection with node switch set, the station number of each slave uses the value set in the node switch of the slave as the slave station number. After that, when performing EtherCAT connection through slave connection or function block command, the function to check whether the connection sequence of slaves matches the EtherCAT parameter setting with the station number of the slave set in the node switch, and to set the station number of the EtherCAT network by the node switch setting It provides a matching function. By setting node switch operation mode, you can set the node switch operation mode when connecting to EtherCAT.

(1) Node switch use conditions

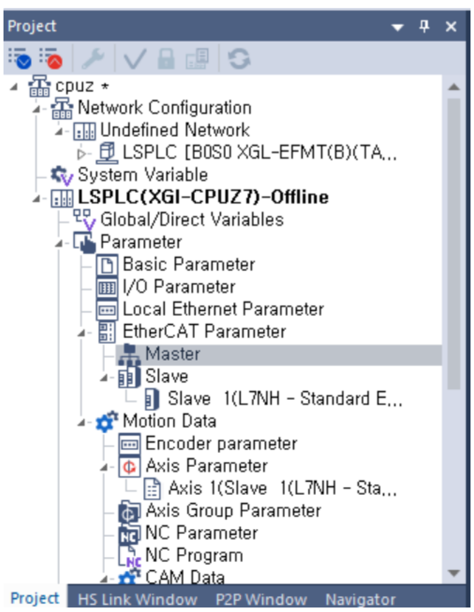
(a) All slaves in the EtherCAT network must have the node switch value set.

(b) The node switch setting value of all slaves must be set within the slave station number setting range (1~64).

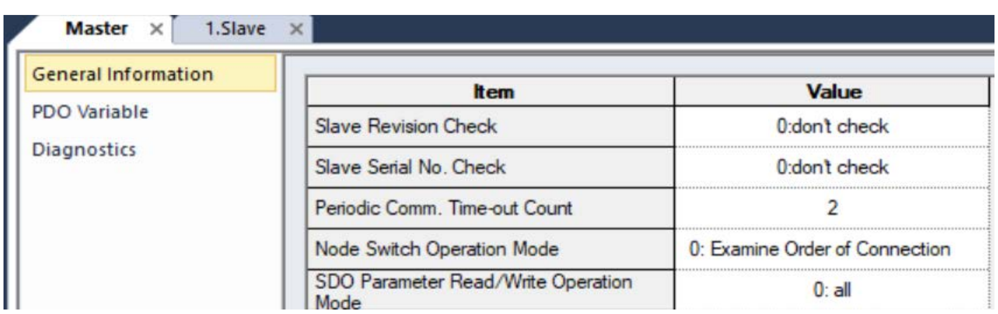
(c) There should be no slaves with duplicate node switch settings

(2) Node switch operation mode setting

(a) Node switch operation mode can be set by selecting Motion Data > EtherCAT Parameter > Master in the project tree.



(b) If you select the General Information tab, you can set the node switch operation mode. Node switch operation mode can be set to 3 modes: 0: Connection order check/1: No connection order check/2: Node switch use.



(3) Node switch operation mode

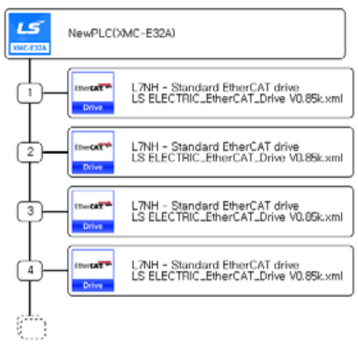
(a) 0: Connection order check: This is an operation mode in which an error occurs when the node switch set value does not match the station number setting of the EtherCAT parameter and the connection sequence.

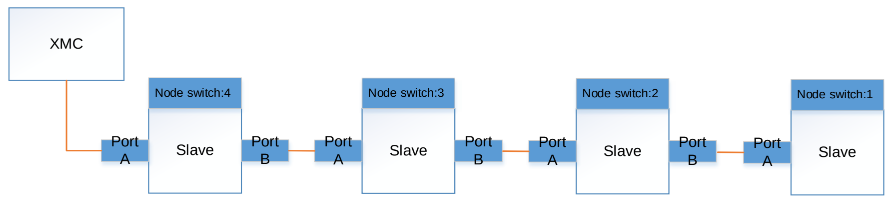
(b) 1: Connection order not check: Regardless of the connection order, it is an operation mode that connects to the setting of the slave with the station number that matches the value set in the node switch. The slaves moved by the node switch setting must use the same slave.

(c) 2: node switch not use: This is an operation mode that does not use node switches.

(4) Node switch operation mode differences

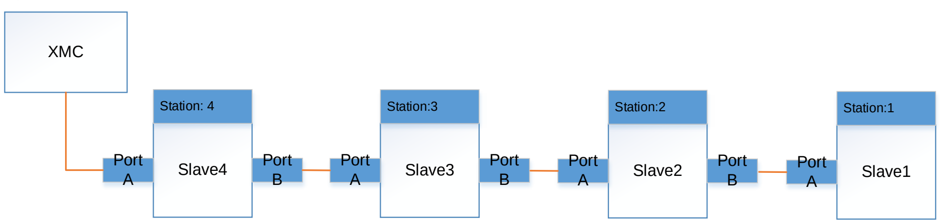
Connect 4 identical EtherCAT slaves, set station number 1 to station number 4 in order from the first slave, and set the node switch to 4 to 1 from the first slave to the actually connected slave. The operation according to the switch operation mode is as follows.



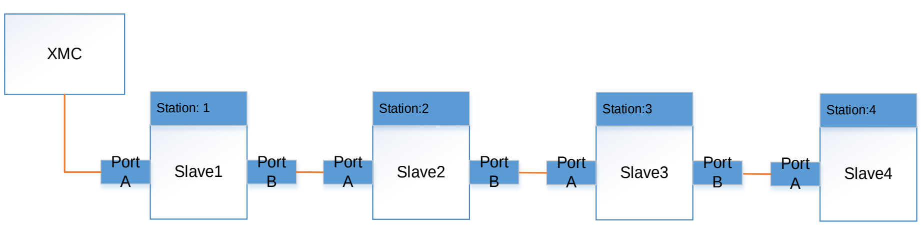


(a) When the node switch operation mode is ‘0: check connect order’, as the node switch setting is different from the connection order of EtherCAT parameter, 0xF0E error occurs and EtherCAT connection fails.

(b)When the node switch operation mode is ‘1: No check connect order’, as the station number is set by the node switch regardless of the connection order, the first slave is connected as slave 4, the second slave is slave 3, the third slave is slave 2, and the fourth slave is connected as slave 1.



(b) When the node switch operation mode is ‘2: No node switch use’,as node switch is not used, as set in EtherCAT parameter, the first slave is connected as slave 1, the second slave is slave 2, the third slave is slave 3, and the fourth slave is connected as slave 4.



(5) Available version of Node switch operation mode setting

Versions that support node switch operation mode setting function are as follows. In the version of the product that does not support node switch operation mode setting, it operates as set as ‘0: check connection order’.

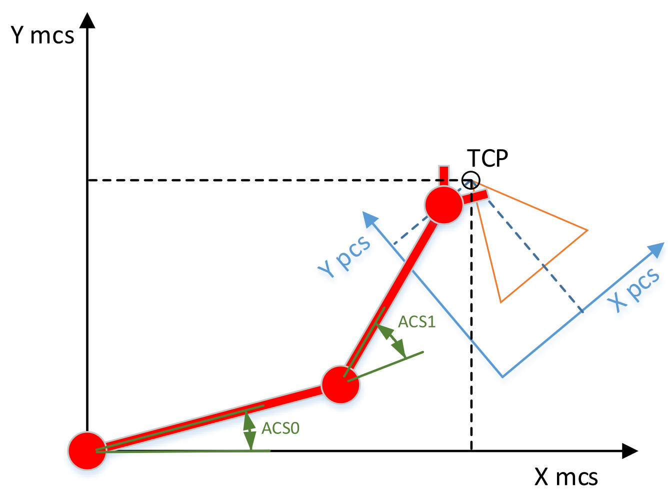
Item / Product name	Module O/S	XG5000
XMC-E32A XMC-E16A XMC-E08A XMC-E32C	V1.70	V4.53

8.4 Coordinate Systems Operation Functions

8.4.1 Overview of Coordinate Systems Operation

Different coordinate systems define various ways specifying certain positions or directions in the space. The figure below shows how to represent a certain TCP through each coordinate system. In the ACS coordinate system, TCP is represented as the rotation angle of a robot joint consisting of two links. In the MCS coordinate system, TCP position is represented based on the home position of MCS. In the PCS coordinate system, TCP position is represented based on the home position of TCP

TCP represented as PCS/MCS cannot be delivered to the motor connected to the robot for operation. To operate the motor connected to the robot, the values converted to ACS should be used, as it represents the actual movement of the motor. Therefore, for operation in a coordinate system, convert PCS to MCS coordinates through the Cartesian coordinate conversion, and convert the MCS coordinates to ACS coordinates through inverse kinematics conversion, and deliver the CA values to each motor to begin operation.



8.4.2 ACS/MCS/PCS/TCP

ACS: Axes Coordinate System (ACS) represents the actual movement of the physical motors.

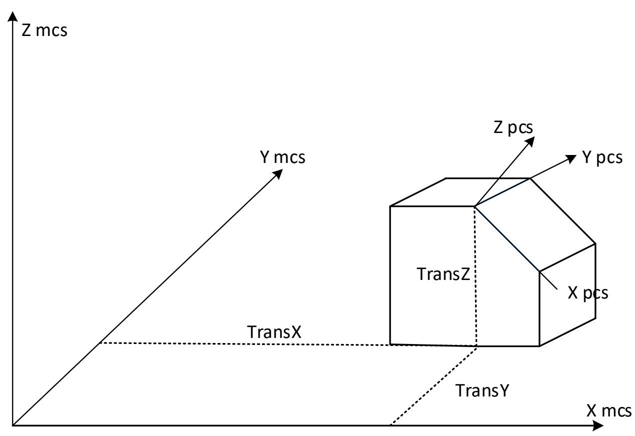
MCS: Machine Coordinate System (MCS) is related with machines (robots). It is the fixed home position of the mechanical system represented as the Cartesian coordinate system.

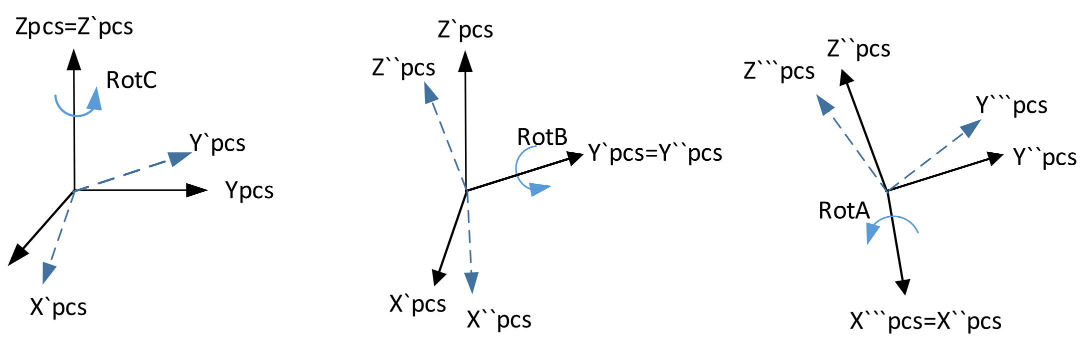
PCS: Product Coordinate System based on MCS represents the position of products being moved or rotated. PCS is linked to the products through a program, and a user can be changed.

TCP (Tool Center Point) is the center or end point of the tool as a position to which a machine (robot) is moved by the command. In case of operation using MCS or PCS, the target position is represented by TCP. TCP consists of 6 RotC data: Px,Py,Pz, representing movement along XYZ axes; RotA representing rotation along X axis; RotB representing rotation along Y axis; and RotC representing rotation along Z axis.

8.4.3 PCB setting

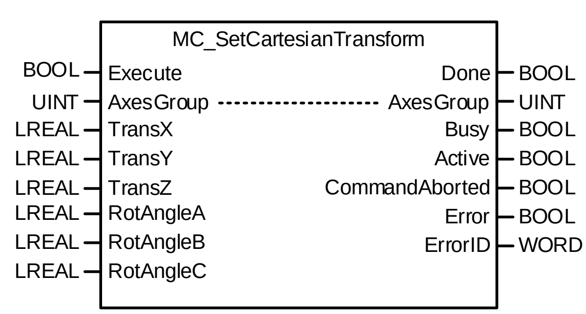
PCS represents TCP on the work stand. TCP is defined by rotation and movement from the origin point. The parameter to convert PCS into MCS can be set using MC_SetCartesianTransform function block or setting axes group parameter. In MC_SetCartesianTransform, TransX/TransY/TransZ represents the distance of movement from the MCS origin point to the PCS origin point. RotA/RotB/RotCare rotation values for PCS. RotA represents PCS rotation along X-axis. RotB represents PCS rotation along Y-axis. RotC represents PCS rotation along Z-axis PCS rotation is performed in the following order: RotC->RotB->RotA





(1) Relevant motion function block

Name	Description	Operation condition
MC_SetCartesianTransform	PCS setting	Edge



8.4.4 Machine Information Setting

To operate the robot using coordinate system operation, the type of the robot (machine) and the machine parameters should be set at the axes group parameter in advance. Machine parameters can be set using MC_SetKinTransform function block. XG5000 axes group parameters can be set using the same.

(1) Machine information, machine type setting

In the machine type settings, select the type of machine (robot) to perform coordinate system operation. XYZ / Delta3 / Delta3R / LinearDelta3 / LinearDelta3R can be selected as the robot type.

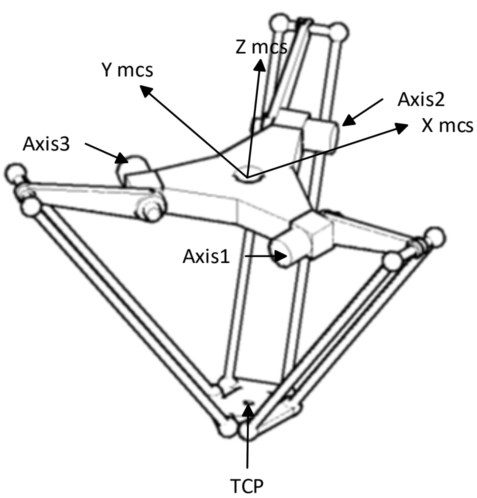
(a) XYZ (Cartesian coordinates) robot

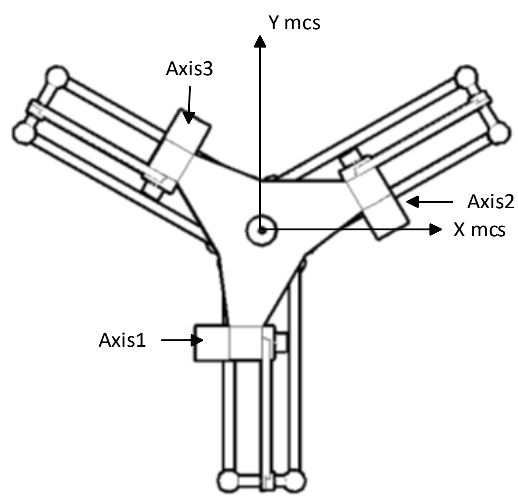
XYZ is a robot type with servo motors connected to X/Y/Z axes, which can perform the operation in Cartesian coordinates, and it does not require additional kinematic analysis between ACS and MCS.

(b) Delta3

MCS of the Delta robot

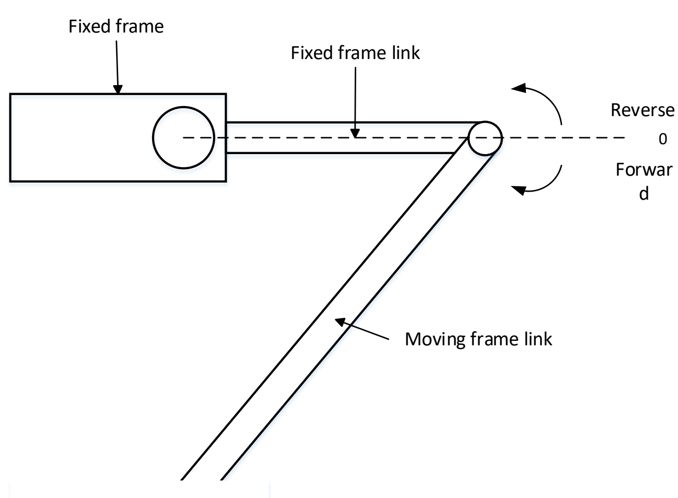
In case of a Delta robot, the center of Fixed Frame is defined as MCS. The relationship between each axis connected to Delta and MCS are as shown below.





ACS of the Delta robot

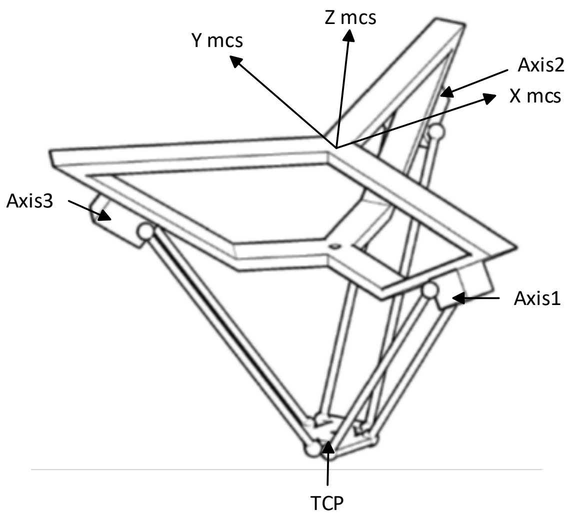
The direction of the link connected to Delta's fixed frame facing the floor is defined as the forward direction of the axis operating the link, and the other direction is defined as the opposite direction.

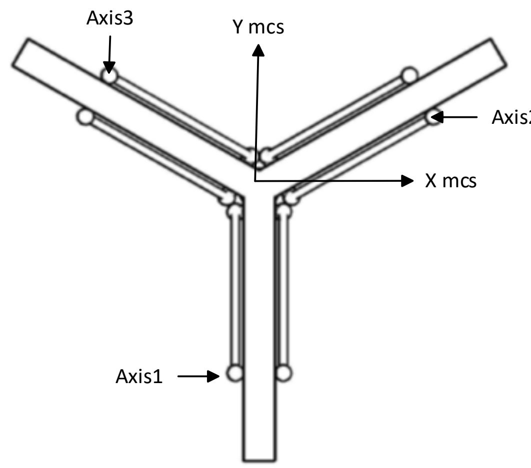


(c) LinearDelta3

MCS of the LinearDelta robot

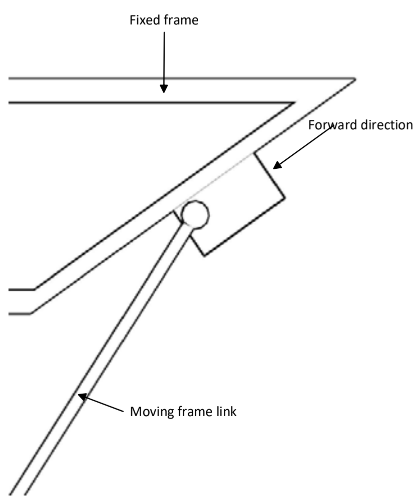
In case of a linearDelta robot, the center of Fixed Frame is defined as MCS. The relationship between each axis connected to linearDelta and MCS are as shown below.





ACS of the LinearDelta robot

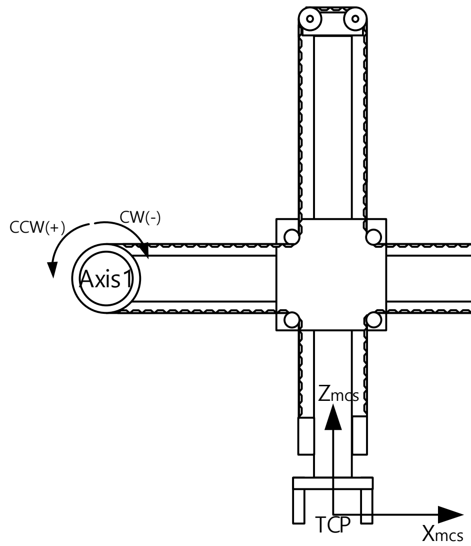
The utmost position that the linear axis of the LinearDelta robot moves toward the ceiling is defined as 0. The direction that the linear axis moves toward the bottom is defined as the forward direction, and the opposite direction is defined as the reverse direction.



(d) T-Gantry

MCS of the T-Gantry robot

The MCS of the T-Gantry robot defines the position of the end of a tool when the positions of Axis 1 and Axis 2 are 0 as the origin (0, 0) of MCS. The relationship between MCS and the axis connected with T-Gantry is shown in the following figure:



ACS of the T-Gantry robot

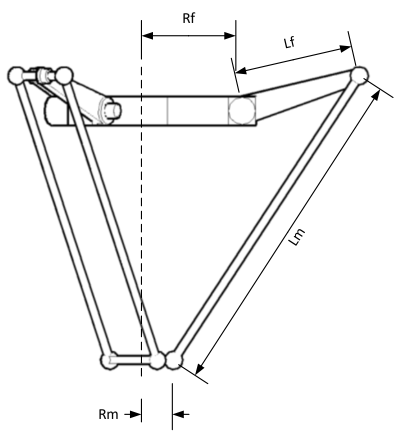
If the structure of the T-Gantry robot is the same as the above, define a counterclockwise operation as a forward one and a clockwise operation as a reverse one.

(2) Machine information, machine parameter setting

(a) XYZ

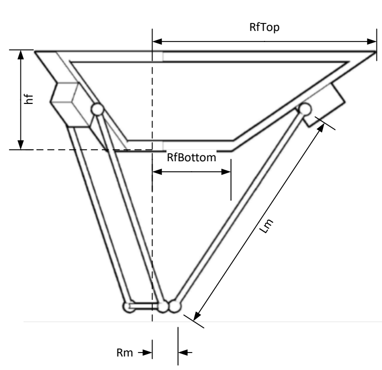
XYZ robot does not require separate machine parameters, as the position of each axis matches the XYZ coordinates of TCP.

(b) Delta3/Delta3R



Parameter	Description
KinParam[0]	Lf:Link length of the fixed frame(mm)
KinParam[1]	Lm: Link length of the moving frame(mm)
KinParam[2]	Rf: Length from the center of the fixed frame to the link of the fixed frame(mm)
KinParam[3]	Rm: Length from the center of the moving frame to the link of the moving frame(mm)

(c) LinearDelta3/LinearDelta3R



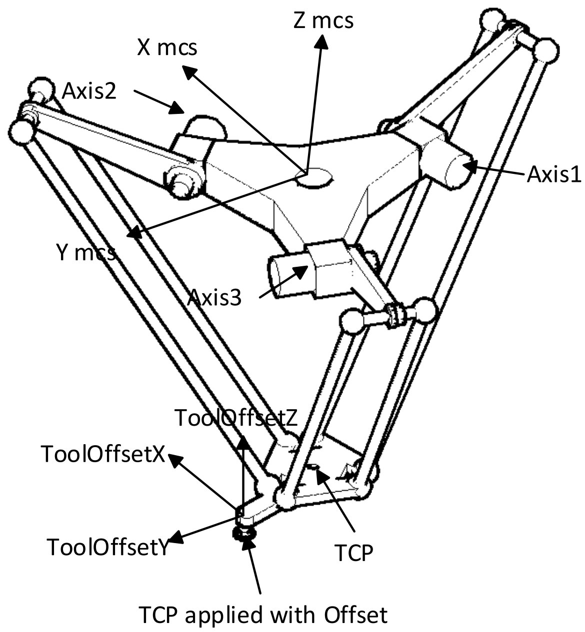
Parameter	Description
KinParam[0]	Lm: Link length of the moving frame(mm)
KinParam[1]	Hf:Fixed frame height (mm)
KinParam[2]	RfTop: Radius of top fixed frame(mm)
KinParam[3]	RfBottom: Radius of bottom fixed frame(mm)
KinParam[4]	Rm: Length from the center of the moving frame to the link of the moving frame(mm)

(d) T-Gantry

The T-Gantry robot does not need the parameter setting of a separate mechanism.

(3) Machine information, tool offset setting

A tool offset function is provided in addition to the machine information, as additional equipment may be connected to the end of the robot's TCP. Activating tool offset applies the offset to the TCP target position applied to the coordinate system operation.



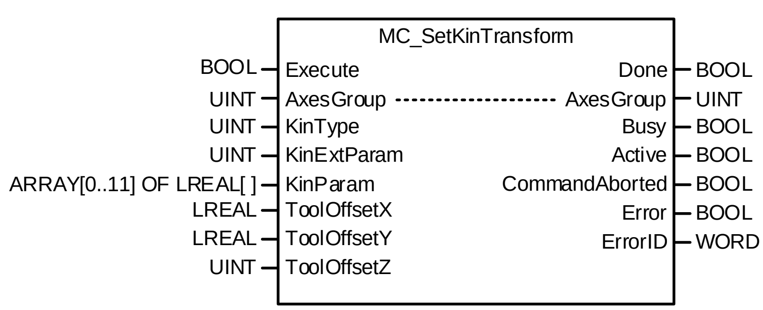
(4) Axis group, axis configuration setting

To perform coordinate system operation, the axes should be set to suit the machine type. Coordinate system operation is not performed if the number of axes or the axis unit does not match.

Machine type	Number of axis	Unit			
		Axis 1	Axis 2	Axis 3	Axis 4
XYZ	Axis 3	mm	mm	mm	-
Delta3	Axis 3	degree	degree	degree	-
Delta3R	Axis 4	degree	degree	degree	degree
LinearDelta3	Axis 3	mm	mm	mm	-
LinearDelta3R	Axis 4	mm	mm	mm	degree

(5) Related motion function block

Name	Description	Operation condition
MC_SetKinTransform	Machine Information Setting	Edge



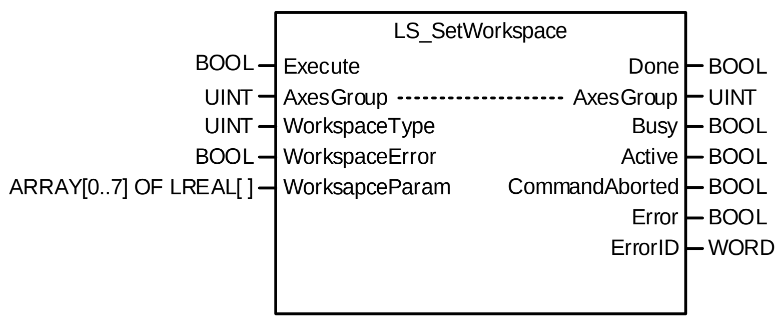
8.4.5 Workspace configuration

For coordinate system operation, in order to prevent machine damage or safety accident caused by the robot performing impossible motion, a work space function is provided to prevent the robot from going out of the preset work space. Coordinate system operation is not performed if the robot's current position or target position is outside the work space. Work space setting can be performed using LS_SetWorkspace function block. XG5000 axes group parameters can be set using the same

(1) Work Space Setting

Perform work space settings, and the occurrence of work space error can be set. Set the workspace type set in the WorkspaceType as the work space parameter set in the WorkspaceParam in the axis group specified in the set AxesGroup input. If WorksaceError value is set to 0, the operation continues without errors even when it goes out of the work space.
Work space settings cannot be performed while the axes group is in operation.

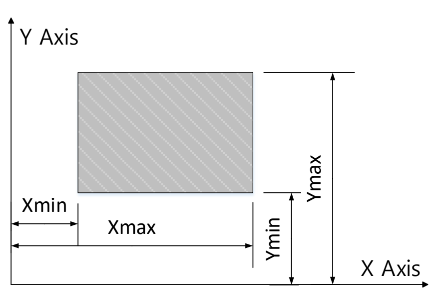
Name	Description	Operation condition
LS_SetWorkspace	Work Space Setting	Edge



(2) Workspace type and parameter

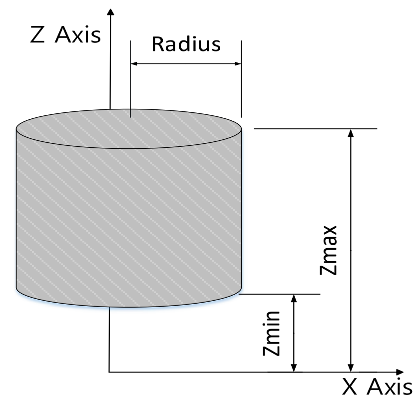
The work space type supports 4 types of Rectangle/Cylinder/Delta/Sector.

Rectangle



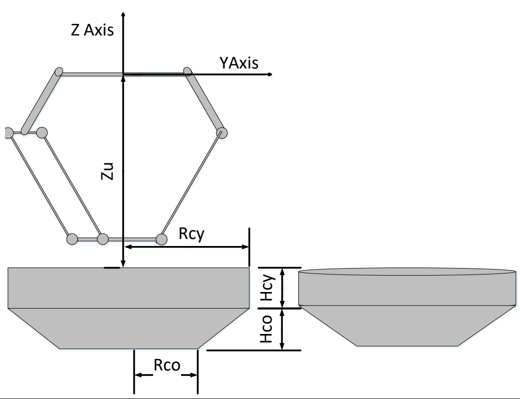
Parameter	Value
WorkspaceParam[0]	X max(mm)
WorkspaceParam[1]	X min(mm)
WorkspaceParam[2]	Y max(mm)
WorkspaceParam[3]	Y min(mm)
WorkspaceParam[4]	Z max(mm)
WorkspaceParam[5]	Z min(mm)

Cylinder



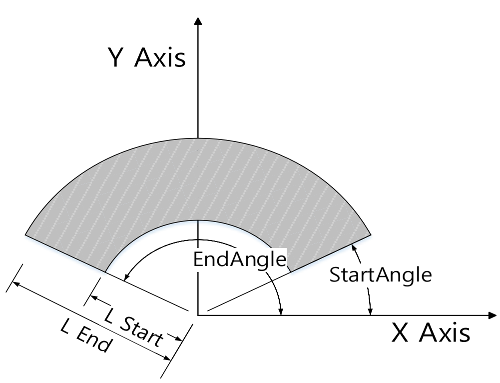
Parameter	Value
WorkspaceParam[0]	Radius(mm)
WorkspaceParam[1]	Z max(mm)
WorkspaceParam[2]	Z min(mm)

Delta



Parameter	Value
WorkspaceParam[0]	Zu(mm)
WorkspaceParam[1]	Hcy(mm)
WorkspaceParam[2]	Hco(mm)
WorkspaceParam[3]	Rcy(mm)
WorkspaceParam[4]	Rco(mm)
WorkspaceParam[5]	-

Sector



Parameter	Value
WorkspaceParam[0]	L end (mm)
WorkspaceParam[1]	L start(mm)
WorkspaceParam[2]	Z max(mm)
WorkspaceParam[3]	Z min(mm)
WorkspaceParam[4]	EndAngle(degree)
WorkspaceParam[5]	StartAngle(degree)

8.4.6 Time Linear Interpolation Operation for Absolute Position of Coordinate System

Use the related axes set in the axes group to perform interpolation control by moving the TCP from the current position to the target position in the set time in a linear trajectory.

(1) Executes linear interpolation from starting position to the target position designated on positioning data. Positioning control is carried out based on the position specified from homing.
(2) Set Position[ ] to define the TCP target position.

Variable	Meaning	Unit
Position[0]	X axis position	mm
Position[1]	Y axis position	mm
Position[2]	Z axis position	mm
Position[3]	X axis rotation amount	degree
Position[4]	Yaxis rotation amount	degree
Position[5]	Zaxis rotation amount	degree

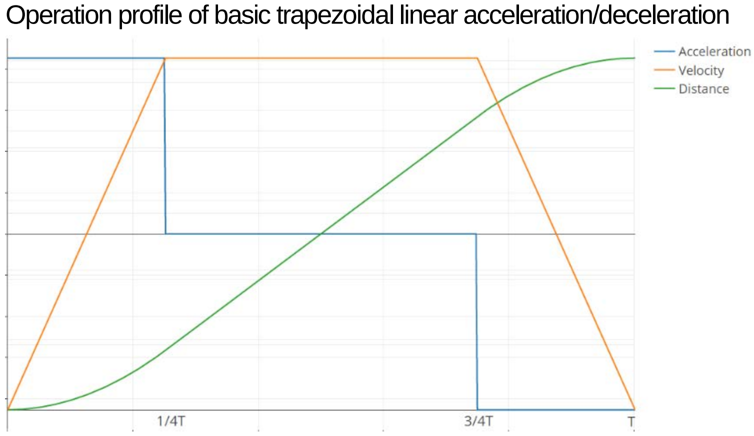
(3) Depending on the robot type, some Position variable areas may not be applied. Data input in the unapplied areas is not reflected in coordinate system operation.

Variable	Robot type	
	XYZ	Delta3
Position[0]	Apply	Apply
Position[1]	Apply	Apply
Position[2]	Apply	Apply
Position[3]	Disabled	Disabled
Position[4]	Disabled	Disabled
Position[5]	Disabled	Disabled

(4) Perform linear interpolation to reach the target TCP in the time set in TrajTime.
(5) TrajType input determines the type of acceleration/deceleration for reaching the interpolation trajectory. Three types of 0: Trapezoid/Sine1/Sine2are available.

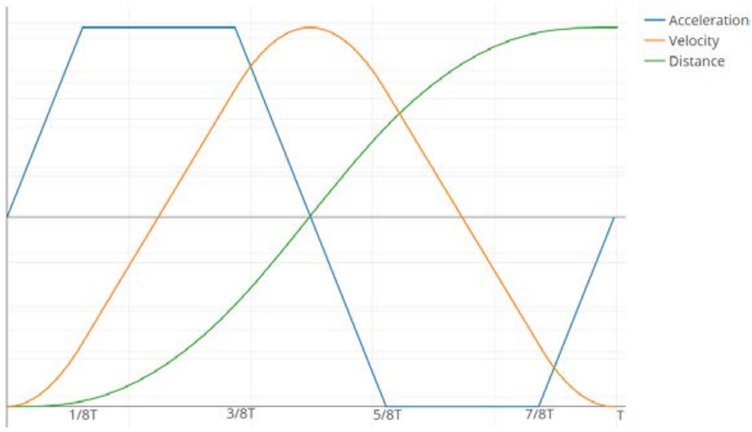
Trapezoid

Operation profile of basic trapezoidal linear acceleration/deceleration

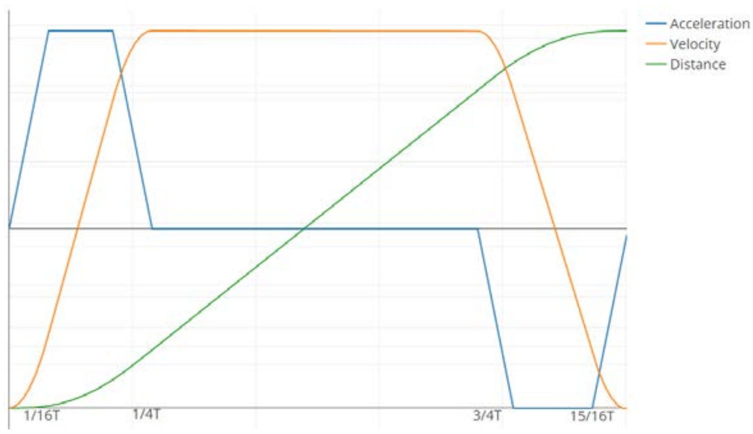


Sine1

The velocity profile of this operation type consists of sine curves. It is suitable for low-load high-velocity operation, and reduces impact on the motor caused by load changes.

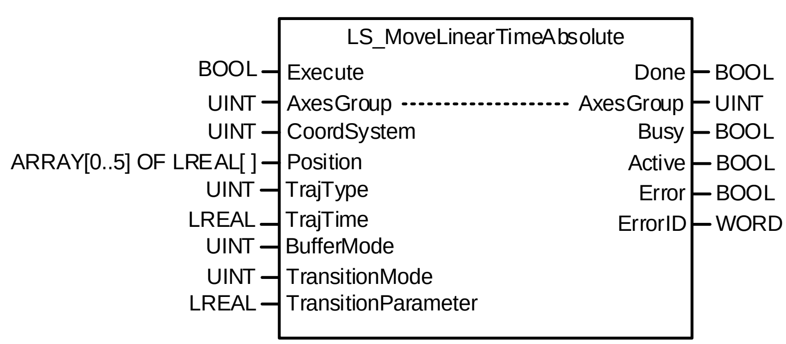


Sine2 (Sine With Constant)
This velocity profile of this operation type consists of sine curves and constant speed sections. It is suitable for high-load, medium-velocity operation.



(6) When CoordSystem input is set to 1, the robot operates using the Position values as MCS coordinates. When it is set to 2, the robot operates using the Position values as PCS coordinate system.
(7) To stop the current interpolation control, use MC_GroupHalt or MC_GroupStop motion function block.
(8) Relevant motion function block

Name	Description	Operation condition
MC_MoveLinearTimeAbsolute	Coordinate system absolute position time linear interpolation operation	Edge



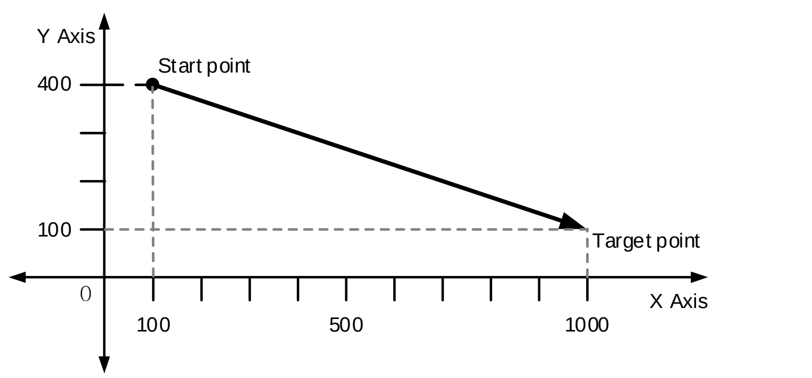
(9) Restrictions

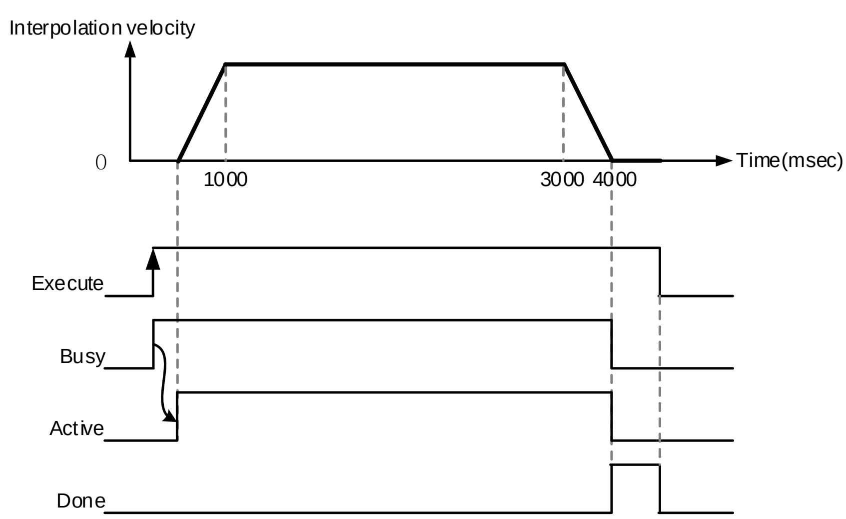
Coordinate system absolute position time linear interpolation control cannot be performed in case of the following errors

- CoordSystem input is set to a value other than 1 or 2 (Error Code: 0x20BC)
- The operation parameter unit of the component axes is not compatible with the coordinate system type (Error Code: 0x2063)
- In case there is an axis which is in the origin indetermination state among configuration axes (error code: 0x20B0)
- In case the operation speed of configuration axis exceeds the speed limit of each axis (error code: 0x20B9)
- In case there is an axis in infinite running repetition operation among configuration axes (error code: 0x20BA)

(10) Operation Timing

- Start position:(100.0, 400.0, 0.0)
- Target position:(1000.0, 100.0, 0.0)
- Target time: 4000msec
- Operation tyoe: 0





8.4.7 Circular Interpolation Operation for Coordinate System

Coordinate system-based circular interpolation operation is performed, where the TCP moves in a circular trajectory on the XY plane using the designated axis in the axes group. Coordinate system absolute position circular interpolation control involves the same setting and motion except that it is based on a coordinate system. There are three kinds of methods for circular interpolation such as midpoint method that passes through the position specified in auxiliary point, center point method that considers the position specified in auxiliary point as center point and radius method that takes the value specified in auxiliary point as the radius of an arc depending on 'CircMode' settings and auxiliary points. To stop the current interpolation control use MC_GroupHalt or MC_GroupStop motion function block.

(1) EndPoint/AuxPoint

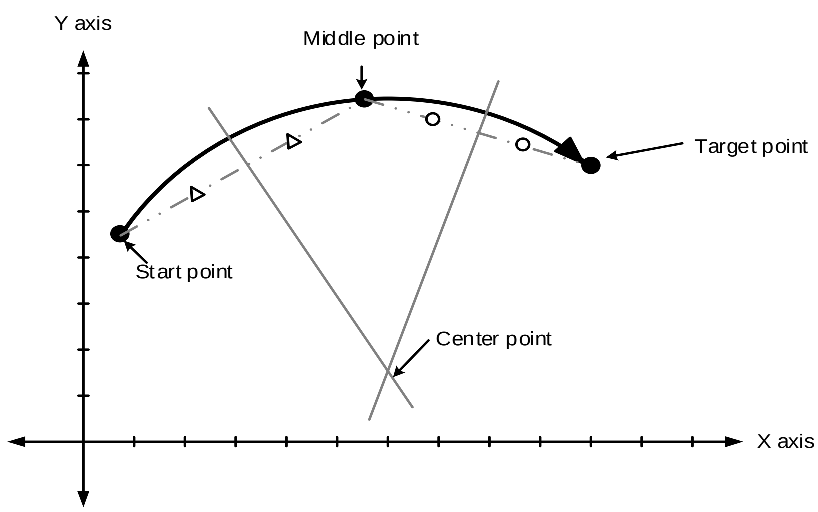
In case of coordinate system circular interpolation control, enter the Px,Py,Pz of TCP to EndPoint/AuxPoint. The RotA, RotB, RotC values, which determine the TCP posture, is not entered, instead maintaining the values at the start position.

(2) Circular interpolation with middle point designation form.

(a) Circular interpolation is executed from starting position to target position through midpoint position set in auxiliary point.

(b) To be made path of circular interpolation with start position, midpoint and a crossing which is perpendicular divi de equally position of midpoint and target position.

(c) Movement direction is decided automatically depends on set target position and auxiliary point of circular interpol ation.

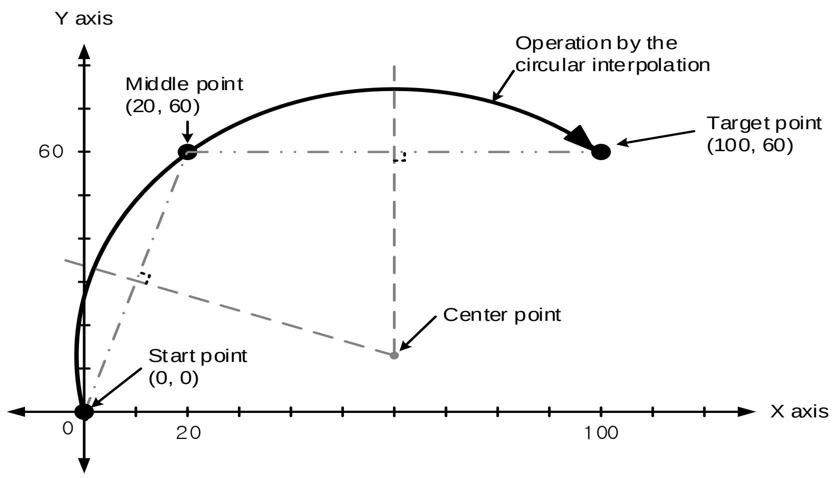


(d) Restrictions

Circular interpolation control using mid-point specification method cannot be performed in case of the following errors.

- During absolute coordinate circular interpolation, home position has not been determined in one or more of the component axes (Error Code: 0x20A0)
- In case the midpoint specified as auxiliary point is the same as the starting position or target position (error code: 0x20A4)
- In case starting position is the same as the target position (error code: 0x20A5)
- In case start position, auxiliary point position and target position are in a straight line (error code: 0x20A7)
- In case there is an axis in infinite running repetition operation among configuration axes (error code: 0x20A8)

(e) Operating Pattern
- Start position:(0.0, 0.0, 0.0)
- Target position:(100.0, 60.0, 0.0)
- Midpoint position:(20.0, 60.0)
- CircMode: Midpoint(0)
- Direction(PathChoice): - (Ignored in middle point method)

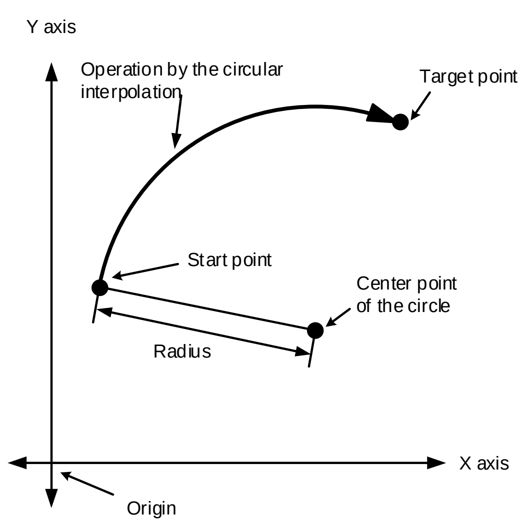
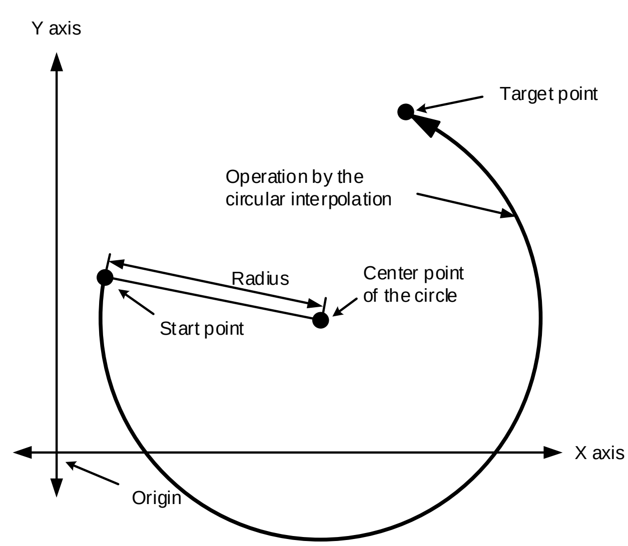


(3) Circular interpolation with center point designation form

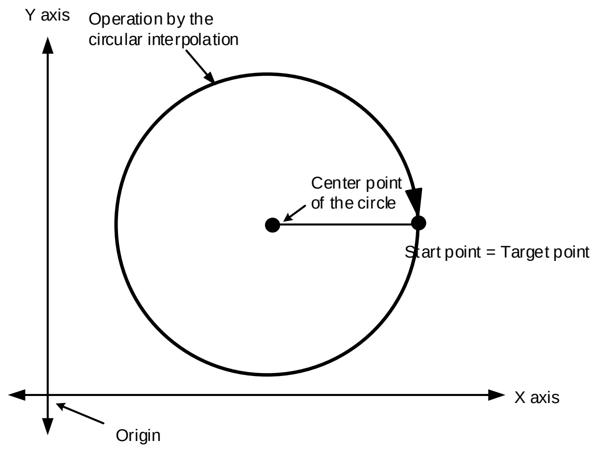
(a) Circular interpolation is performed by starting at the start position, and reaching the target position in a circular t rajectory of which the diameter is the distance to the designated center point.

(b) The movement direction is determined as the direction set in the absolute position circular interpolation operatio n (MC_MoveCircularAbsolute2D), the relative position circular interpolation operation (MC_MoveCircularRelative2D), or "PathChoice" of the motion function block.

- 0:「CW」 - perform circular interpolation in the clockwise direction from the start position.
- 1:「CCW」- perform circular interpolation in the counter-clockwise direction from the start position.

Center point, CW	Center point, CCW
	

(c) If target position is same with start position, can progress circular interpolation. And the circle radius is distance from midpoint to starting position (=target position)



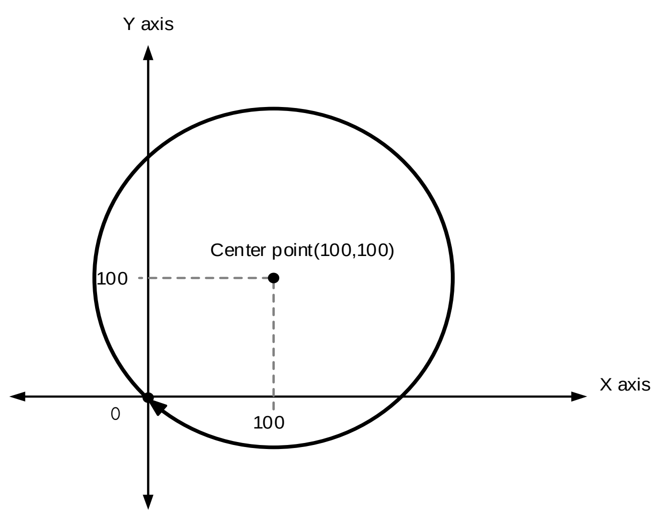
(d) Restrictions

Circular interpolation control using center point specification method cannot be performed in case of the following errors.

- During absolute coordinate circular interpolation, home position has not been determined in one or more of the component axes (Error Code: 0x20A0)
- In case the midpoint specified as auxiliary point is the same as the starting position or target position (error code: 0x20A4)
- In case start position, auxiliary point position and target position are in a straight line (error code: 0x20A7)
- In case there is an axis in infinite running repetition operation among configuration axes (error code: 0x20A8)

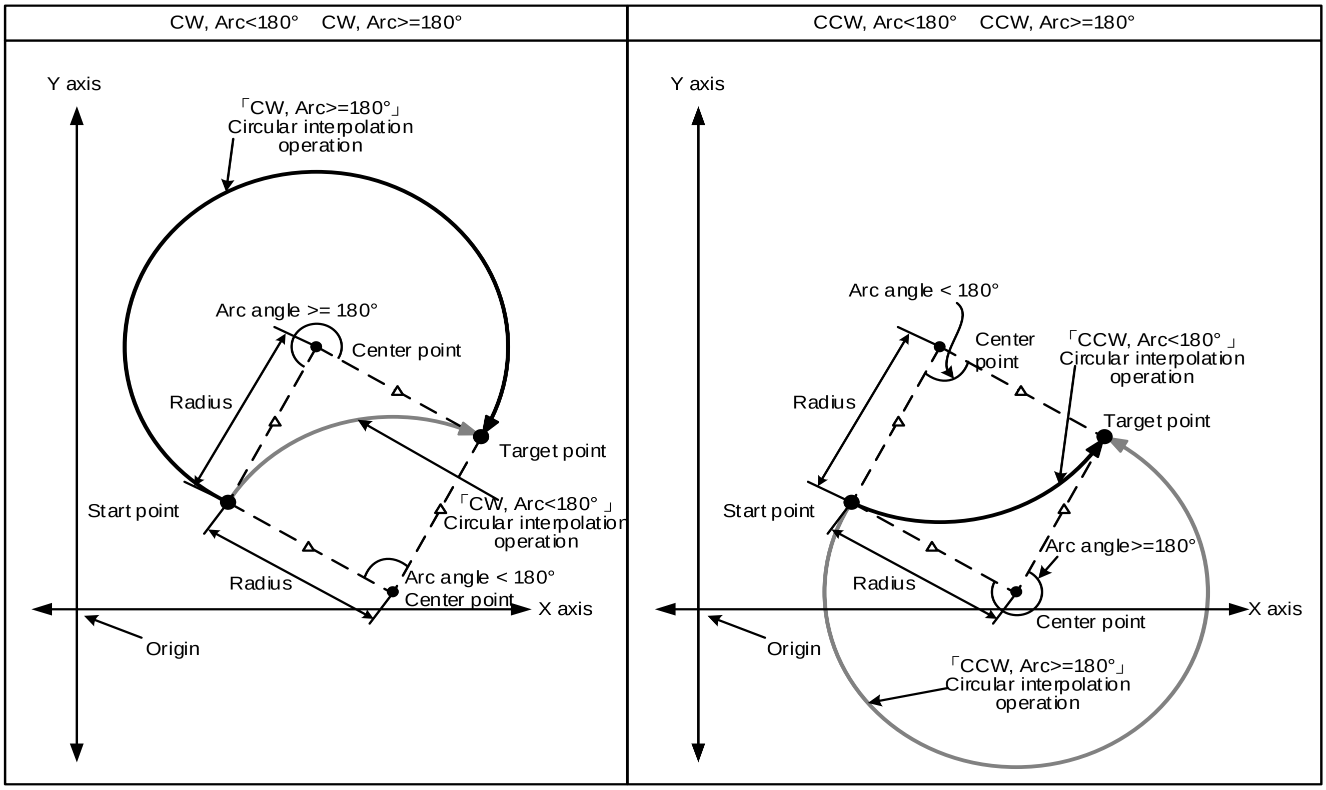
(e) Operating Pattern

- Start position:(0.0, 0.0, 0.0)
- Target position:(0.0, 0.0, 0.0)
- Auxiliary position:(100.0, 100.0, 0.0)
- CircMode: Center Point(1)
- PathChoice: CW(0)



(4) Circular interpolation with radius designation form

(a) Circular interpolation is performed from starting position to target position along the trajectory of the arc that takes the value set in circular interpolation auxiliary point. The arc that has center point depending on the sign of radius ((+): arc angle <180°, (-): arc angle>=180°) is drawn.



(b) In circular interpolation of radius specification method, the target position cannot be set the same as starting position.

(c) The movement direction and the size of the arc are determined by the signs of the auxiliary point and the direction set in the absolute position coordinate system circular interpolation operation (MC_MoveCircularAbsolute2D), the relative position coordinate system circular interpolation operation (MC_MoveCircularRelative2D), or “PathChoice” of the motion function block.

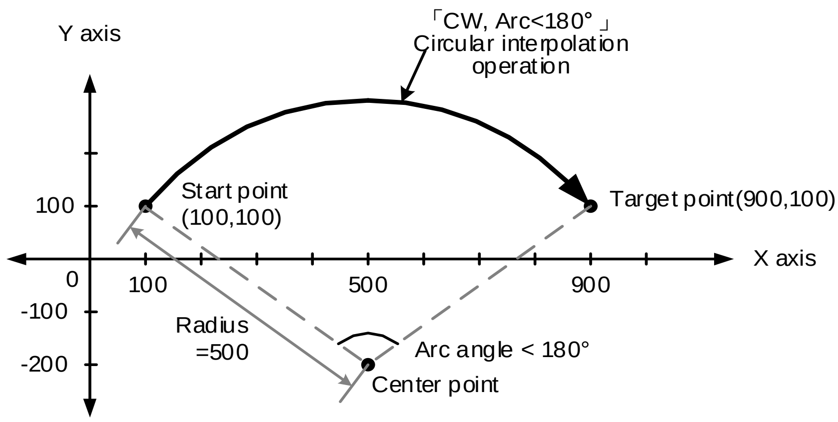
(d) Restrictions

Circular interpolation control by radius specification method cannot be performed in case of the following errors.

- During absolute coordinate circular interpolation, home position has not been determined in one or more of the component axes (Error Code: 0x20A0)
- In case starting position is the same as the target position (error code: 0x20A5)
- In case start position, auxiliary point position and target position are in a straight line (error code: 0x20A7)
- In case there is an axis in infinite running repetition operation among configuration axes (error code: 0x20A8)

(e) Operating Pattern

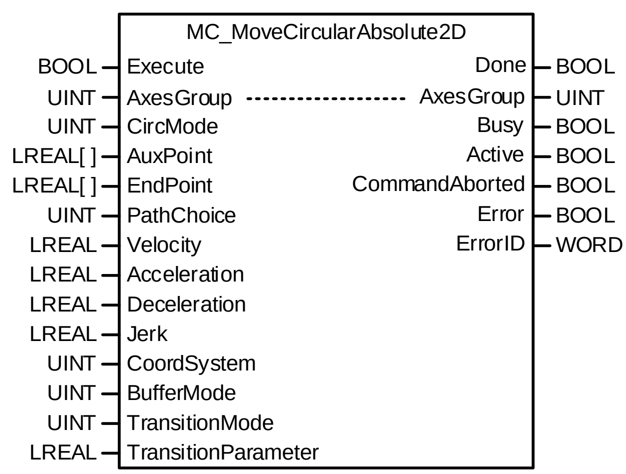
- Start position:(100.0, 100.0, 0.0)
- Target position:(900.0, 100.0)
- Auxiliary position:(500.0, 0.0)
- CircMode: Center Point(2)
- PathChoice: CW(0)



(5) Relevant motion function block

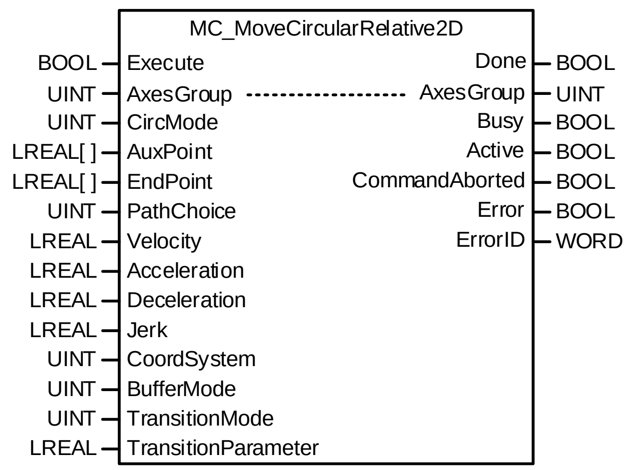
(a) Absolute position coordinate system circular interpolation operation

Name	Description	Operation condition
MC_MoveCircularAbsolute2D	Absolute position circular interpolation operation	Edge



(b) Relative position coordinate system circular interpolation operation

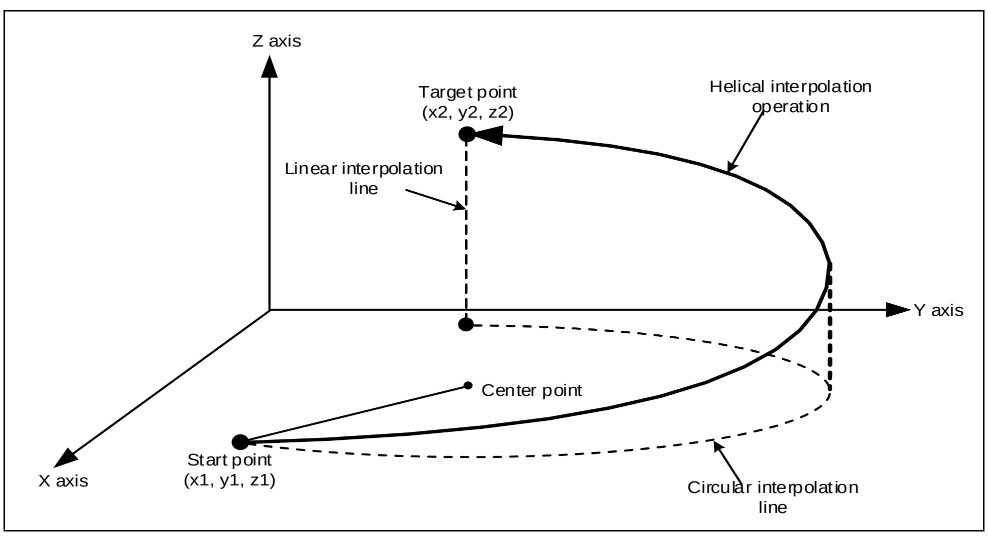
Name	Description	Operation condition
MC_MoveCircularRelative2D	Relative position circular interpolation operation	Edge



(6) Helical interpolation

(a) When circular interpolation commands (absolute position coordinate system circular interpolation operation (MC_ MoveCircularAbsolute2D), relative position coordinate system circular interpolation operation (MC_MoveCircularRe lative2D)) are executed, circular interpolation is performed by moving in a circular trajectory on the XY plane, w hile linear interpolation synchronized to the circular interpolation motion is performed with respect to Z-axis

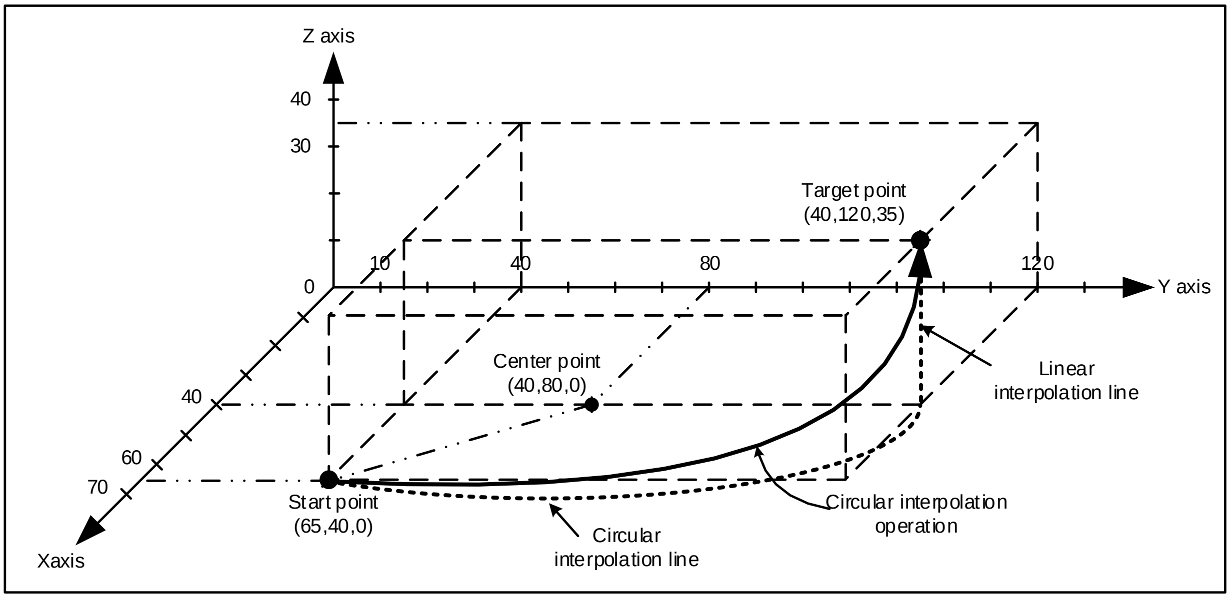
(b) To perform helical interpolation, set the target position for linear interpolation at Pz of ‘EndPoint’.



(c) Restrictions

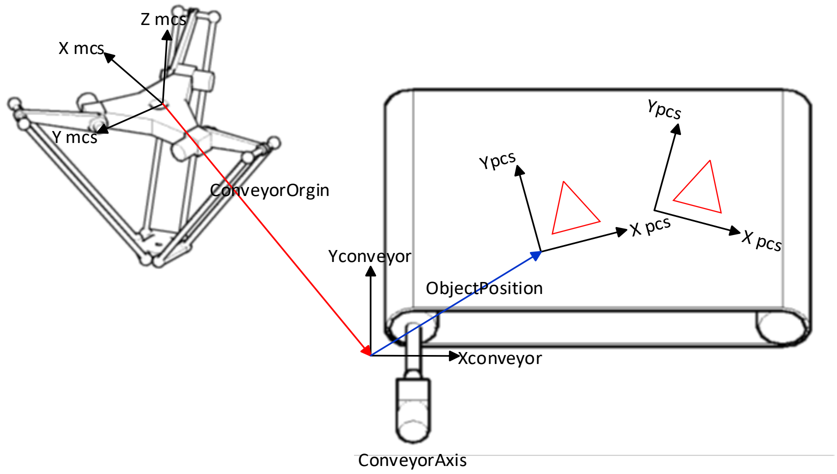
The restrictions of helical interpolation are the same as those of circular interpolation according to the set circular interpolation modes.

(d) Operating Pattern
- Start position:(65.0, 40.0, 0.0)
- Target position:(40.0, 120.0, 35.0)
- Center point position:(40.0, 80.0, 0.0)
- CircMode: Center Point(1)
- PathChoice: CCW(1)



8.4.8 Synchronized Operation for Conveyor Belt

In a coordinate-based operation, one of the axes group is designated as the conveyor axis, and the objects moving on the conveyor in a straight line are tracked.



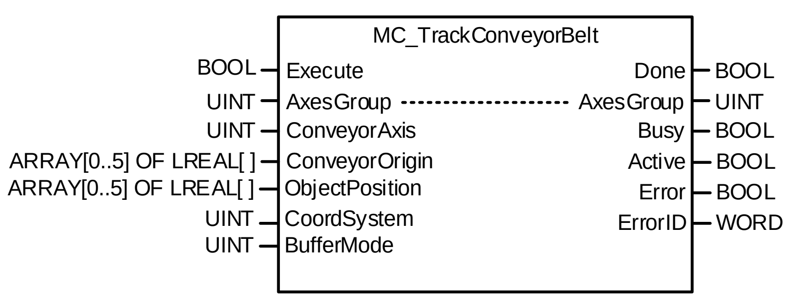
(1) Setting and disable of the conveyor belt synchronized operation

MC_TrackConveyorBelt function block performs the setting for conveyor belt synchronized operation. It is not directly involved in operation. After performing the setting for conveyor belt synchronization with MC_TrackConveyorBelt function block, coordinate system-based motion function blocks where the CoordSystem performed after the setting is set to PCS are synchronized to the conveyor belt for operation. After completing synchronized conveyor belt operation, to perform PCS operation which does not perform conveyor belt synchronized operation, the synchronized conveyor belt operation should be disabled. In order to disable synchronized conveyor belt operation by performing MCS operation or using MC_TrackConveyorBelt function block, the PCS coordinate system should be reset using MC_SetCartesian function block.

(2) Relevant motion function block

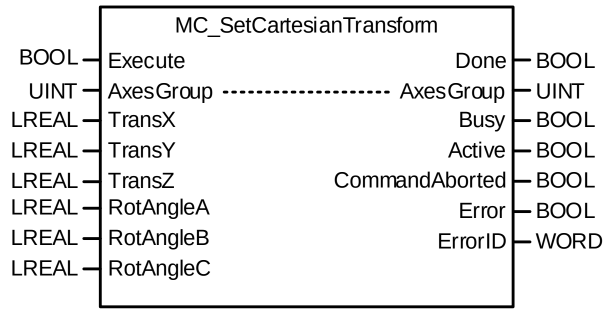
(a) Conveyor belt synchronized setting

Name	Description	Operation condition
MC_TrackConveyorBelt	Conveyor belt synchronized setting	Edge



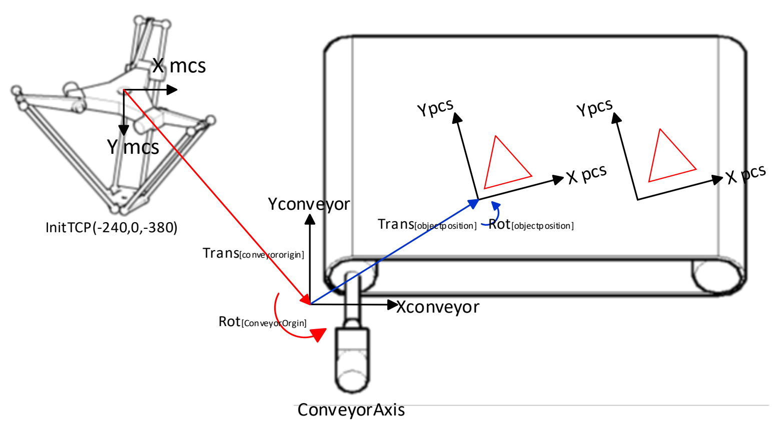
(b) Conveyor belt synchronized setting disable (PCS setting)

Name	Description	Operation condition
MC_SetCartesianTransform	PCS setting	Edge

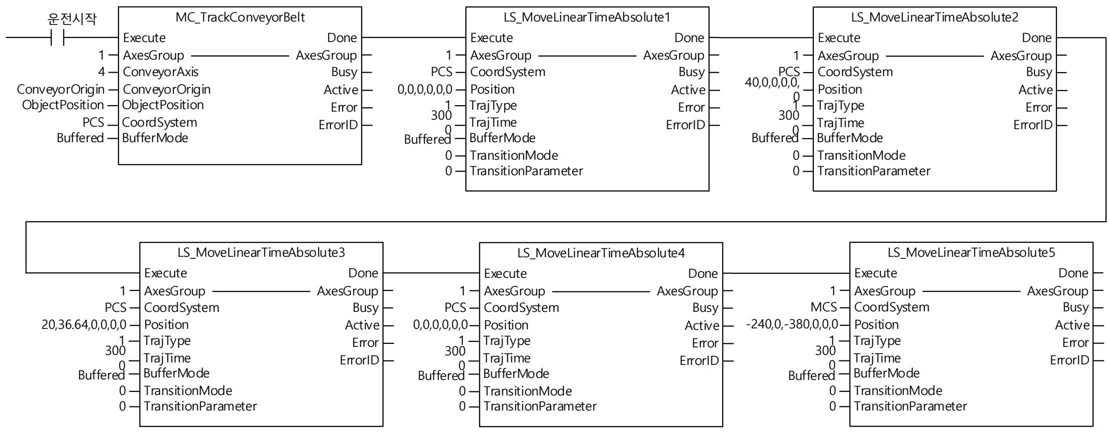


(3) Conveyor belt synchronized function operation example

The conveyor belt synchronization function begins with setting conveyor synchronization using MC_TrackConveyorBelt function block. For MC_TrackConveyorBelt function block, enter conveyor axis value at the ConveyorAxis input. Enter the position from the origin of the robot to the conveyor belt in ConveyorOrigin. Input the position of the work piece from the origin of the conveyor into ObjectPosition. Once MC_TrackConveyorBelt function block setting is complete, LS_MoveLinearTimeAbsolute function block where the subsequently applied CoordSystem input is set to PCS is operated in sync with the conveyor. Synchronized conveyor operation performs an operation of drawing a triangle on a product. After synchronized conveyor belt operation is completed, execute LS_MoveLinearTimeAbsolute function block where the CoordSystem is set to MCS to return to the previous status where the conveyor work is not yet performed.



FunctionBlock	CoordSystem	Position[]	Description
MoveLinearTimeAbsolute1	PCS	0.0, 0	Move to object origin
MoveLinearTimeAbsolute2	PCS	40.0, 0	Draw a triangle1
MoveLinearTimeAbsolute3	PCS	20.36. 64.0	Draw a triangle2
MoveLinearTimeAbsolute4	PCS	0.0, 0	Draw a triangle3
MoveLinearTimeAbsolute5	MCS	-240.0, -380	Move the robot to initial position



(4) Restrictions

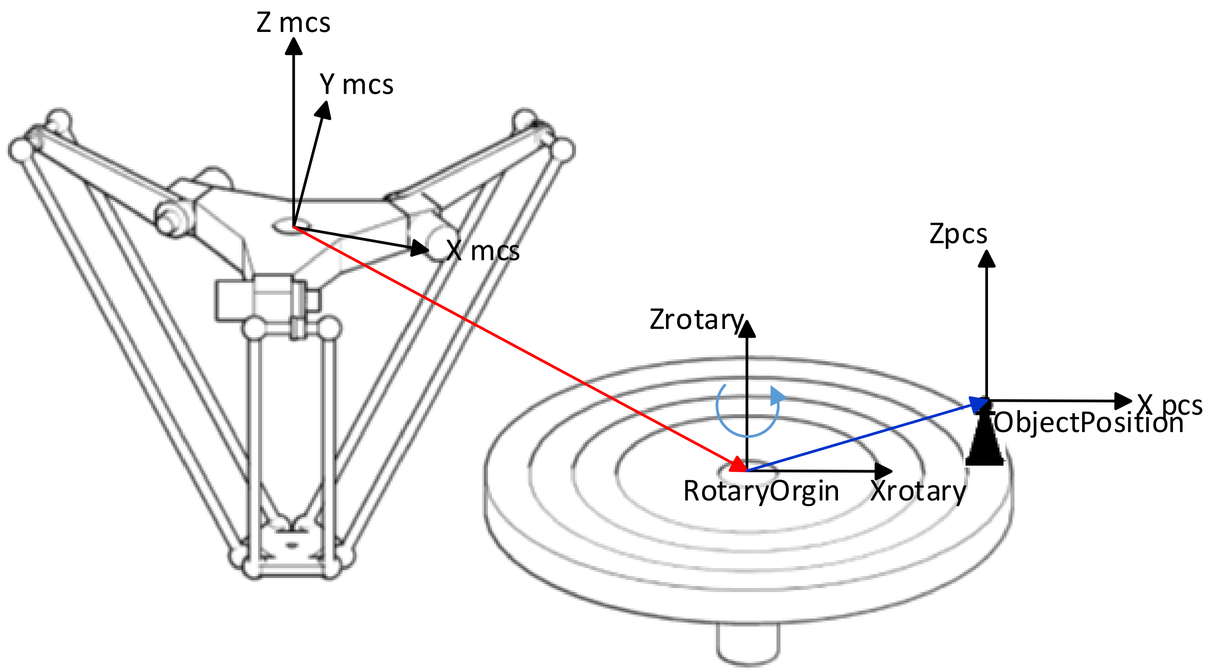
Conveyor belt synchronization cannot be set in the case of the following errors.

- Value other than 2 is set in CoordSystem and performed (Error Code: 0x20BC)
- Axis set in ConveyorAxis is not connected (Error Code: 0x20C3)
- The unit of operation parameter of the axis set in ConveyorAxis is not mm/inch(Error Code: 0x20C2)
- Axis set in ConveyorAxis is not set as the infinite running repeat operation (Error Code: 0x20C6)
- Axis set in ConveyorAxis is the component axis in the applicable axis group (Error Code: 0x20C1)
- In case there is an axis which is in the origin indetermination state among configuration axes (error code: 0x20B0)
- In case there is an axis in infinite running repetition operation among configuration axes (error code: 0x20BA)
- Target position not reachable by coordinate system operation (Error code: 0x20C7)

(If 0x20C7 error occurs, jog operation, MCS coordinate system operation, and axis operation can be performed to resume operation after leaving the error position.)

8.4.9 Synchronized Operation for Rotary table

In a coordinate-based operation, one of the axes group is designated as the rotary axis, and the objects moving on the rotary table are tracked.



(1) Setting and disable of the rotary table synchronized operation

MC_TrackRotaryTable function block performs the setting for rotary table synchronized operation. It is not directly involved in operation. After performing the setting for rotary table synchronization with MC_TrackRotaryTable function block, coordinate system-based motion function blocks where the CoordSystem performed after the setting is set to PCS are synchronized to the rotary table for operation. After completing synchronized rotary table operation, to perform PCS operation which does not perform rotary table synchronized operation, the synchronized rotary table operation should be disabled. In order to disable synchronized rotary table operation by performing MCS operation or using MC_TrackRotaryTable function block, the PCS coordinate system should be reset using MC_SetCartesian function block.

(2) Relevant motion function block

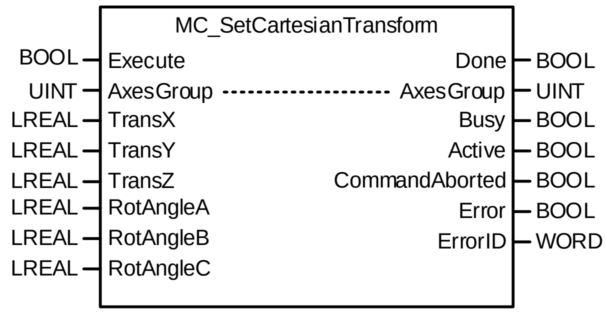
(a) Rotary table synchronized setting

Name	Description	Operation condition
MC_TrackRotaryTable	Rotary table synchronized setting	Edge

```
                                      MC_TrackRotaryTable
                       BOOL ─ Execute                              Done ─ BOOL
                       UINT ─ AxesGroup ------------------- AxesGroup ─ UINT
                       UINT ─ RotaryAxis                           Busy ─ BOOL
ARRAY[0..5] OF LREAL[ ] ─ RotaryOrigin                          Active ─ BOOL
ARRAY[0..5] OF LREAL[ ] ─ ObjectPosition                         Error ─ BOOL
                       UINT ─ CoordSystem                       ErrorID ─ WORD
                       UINT ─ BufferMode
```

(b) Rotary table synchronized setting disable (PCS setting)

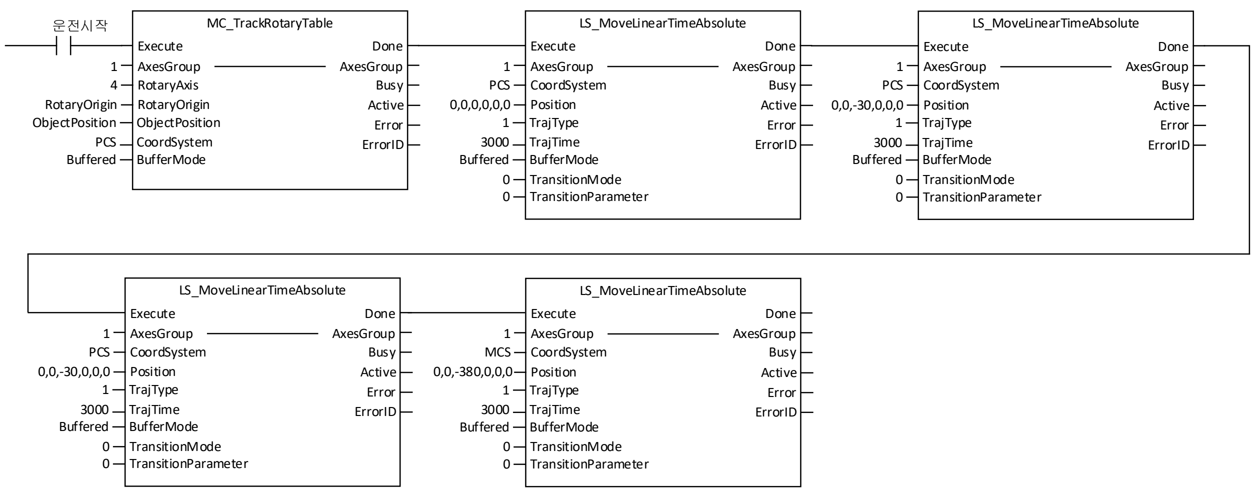
Name	Description	Operation condition
MC_SetCartesianTransform	PCS setting	Edge



(3) Rotary table synchronized function operation example

The rotary table synchronization function begins with setting rotary synchronization using MC_TrackRotaryTable function block. For MC_TrackRotaryTable function block, enter rotary axis value at the RotaryAxis input, enter the rotary table center position from the robot's origin point at the RotaryOrigin. Enter the position of the product origin point from the rotary table center point at the ObjectPosition input. Once MC_TrackRotaryTable function block setting is complete, LS_MoveLinearTimeAbsolute function block where the subsequently applied CoordSystem input is set to PCS is operated in sync with the rotary. Synchronized rotary operation performs to track the object with moving to Z positive direction and Z negative direction. After synchronized rotary table operation is completed, execute LS_MoveLinearTimeAbsolute function block where the CoordSystem is set to MCS to return to the previous status where the rotary work is not yet performed.

FunctionBlock	CoordSystem	Position[]	Description
MoveLinearTimeAbsolute1	PCS	0.0, 0	Move to object center
MoveLinearTimeAbsolute2	PCS	0.0, -30	Track the Object 1
MoveLinearTimeAbsolute4	PCS	0.0, 30	Track the Object 2
MoveLinearTimeAbsolute5	MCS	0.0, -380	Move the robot to initial position



(4) Restrictions

Rotary table synchronization cannot be set in the case of the following errors.

- Value other than 2 is set in CoordSystem and performed (Error Code: 0x20BC)
- Axis set in RotaryAxis is not connected (Error Code: 0x20C3)
- The unit of operation parameter of the axis set in RotaryAxis is not degree(Error Code: 0x20C2)
- Axis set in RotaryAxis is not set as the infinite running repeat operation (Error Code: 0x20C6)
- Axis set in RotaryAxis is the component axis in the applicable axis group (Error Code: 0x20C1)
- In case there is an axis which is in the origin indetermination state among configuration axes (error code: 0x20B0)
- In case there is an axis in infinite running repetition operation among configuration axes (error code: 0x20BA)
- Target position not reachable by coordinate system operation (Error code: 0x20C7)

(If 0x20C7 error occurs, jog operation, MCS coordinate system operation, and axis operation can be performed to resume operation after leaving the error position.)

8.4.10 Path Operation Function for Coordinate System

The coordinate system path operation function stores operation command information in a specific memory area and sequentially executes the stored operation commands to indirectly perform coordinate system operations such as coordinate system linear interpolation operation/circular interpolation operation.

(1) Coordinate system path operation settings

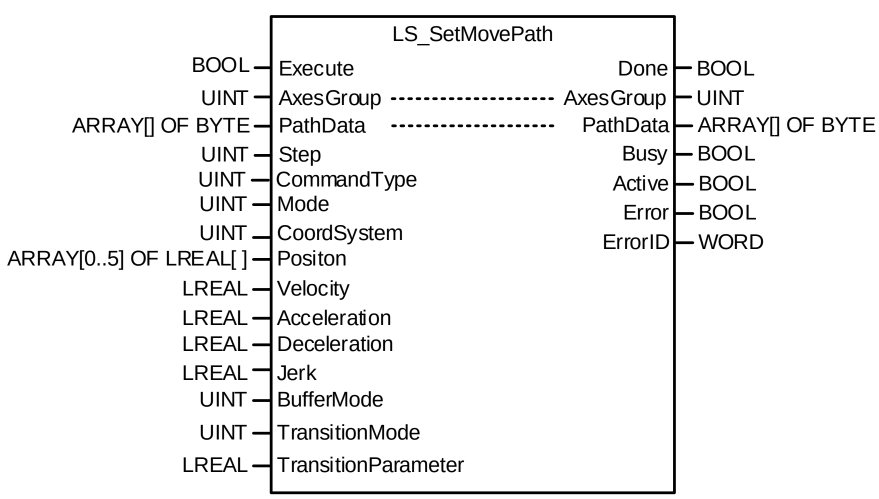
The data of the coordinate system path operation is set using the LS_SetMovePath function block. The path data set in the LS_SetMovePath is stored in the array variable specified as the PathData input. The array variable specified as input of PathData should use an array that is large enough to store the coordinate system path data as an input. Since the size of one step of the coordinate system path data is 96 Bytes, the PathaData should use at least 96 arrays. The sequence of path operation to be set in Step is specified. In CommandType, the type of command to be operated is designated.(0: None 1: Coordinate system absolute position linear interpolation operation 2: Coordinate system relative position linear interpolation operation 3: Coordinate system absolute position circular interpolation operation 4: Coordinate system relative position circular interpolation operation) Mode input is an input for selecting the path of an arc if circular interpolation is selected in the CommandType, and you can select the direction of the arc (0: clockwise 1: counterclockwise). Position is an input for setting the target position, and inputs of the X, Y, Z, A, B and C directions are entered sequentially

The coordinate system path operation is performed using the LS_RunMovePath function block. When the coordinate system path operation is executed, the path data of steps designated as StartStep and EndStep are sequentially executed. Even if the EndStep is not reached at the time of path operation, the path operation is terminated if the CommandType value of the step is set to 0. The step number which is currently being executed during the coordinate system path operation is displayed via CurStep.

(2) Relevant motion function block

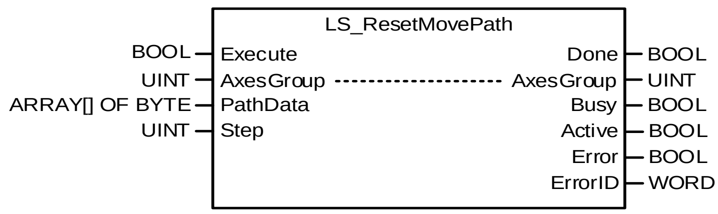
(a) Coordinate system path data setting

Name	Description	Operation condition
LS_SetMovePath	Coordinate system path data setting	Edge



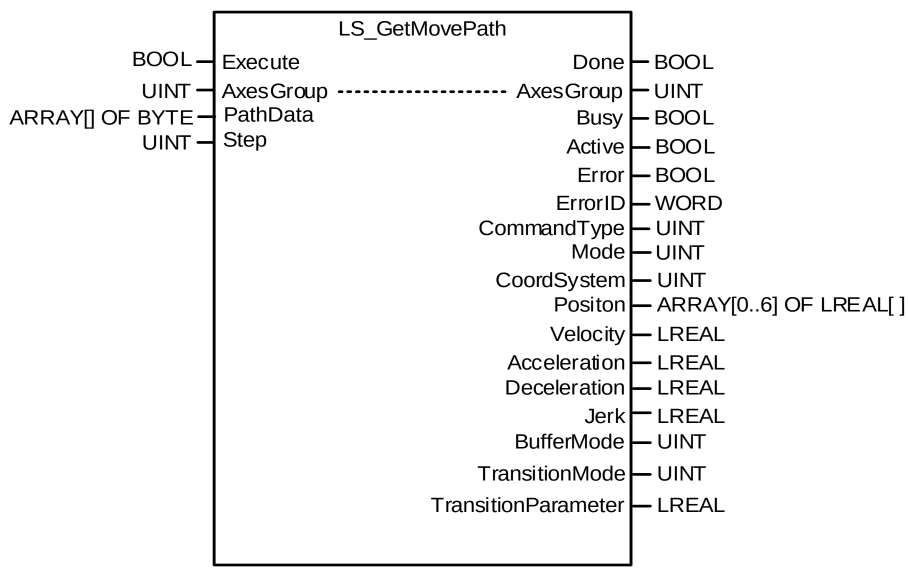
(b) Coordinate system path data remove

Name	**Description**	**Operation condition**
LS_ResetMovePath	Coordinate system path data remove	Edge



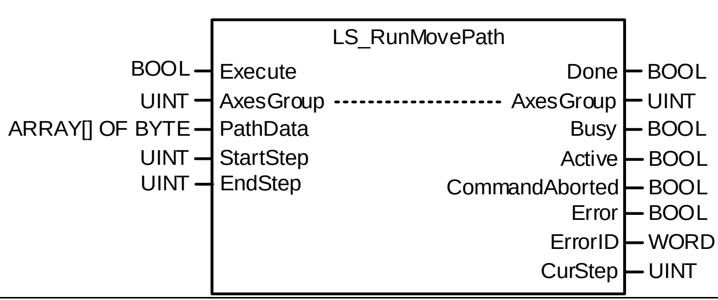
(c) Coordinate system path data read

Name	Description	Operation condition
LS_GetMovePath	Coordinate system path data read	Edge



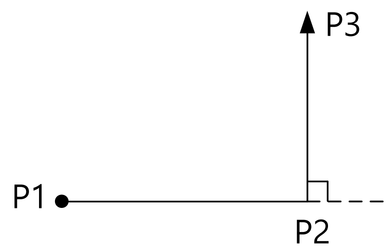
(d) Path Operation for Coordinate System

Name	Description	Operation condition
LS_RunMovePath	Coordinate system path operaton	Edge

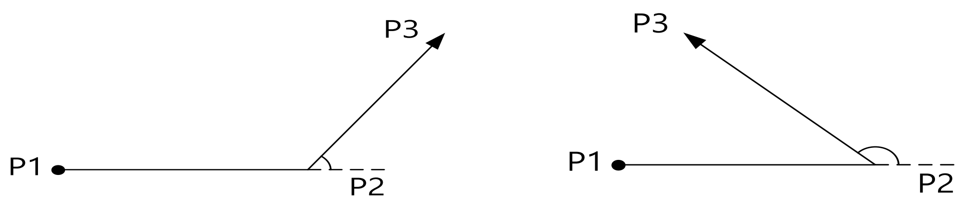


8.4.11 Interpolation operation blending angle limit function

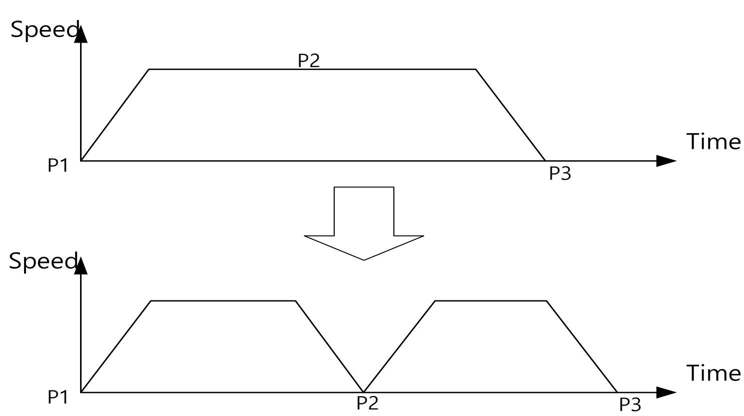
If you use the interpolation blending function during group or coordinate system operation, you can combine the operations and continuously operate according to the set method without stopping between the two operations.
On the plane operation trajectory, the angle between the current position and the target position of the next executed command exists with the target position of the first executed command as the vertex. An example of a plane operation in which the angle between two operations is 90 degrees and passes through the positions P1 → P2 → P3 in order is as follows.



To combine the operations so that the two operations are continuous, the operation that started from the current position (P1) at the first target position (P2) and went to the target position 1 (P2) stops at an infinite deceleration speed and starts at the target position 1 (P2). Then, the operation to the target position 2 (P3) starts the operation with infinite acceleration. In this way, at the position where the two operations are combined, the system is impacted as it is operated with infinite acceleration/deceleration to maintain the combined speed of the interpolation operation.



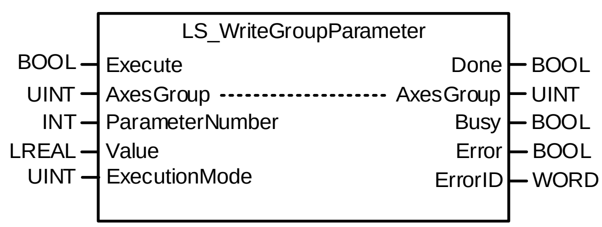
When the angle difference between the two operations is small, the magnitude of the shock generated is small, and the damping of the system itself may seem insignificant.
If a large impact is expected due to the large difference in angle between the two operations, if the interpolation operation blending angle limit function is used, even if the interpolation operation blending is set for the two operations, deceleration stops at the expected impact point (P2) and then accelerates operation.



The unit of the allowable angle is ° (degrees, deg) and can be set from 2° to 178°.
If the angle of the two operations is greater than the set allowable angle and the TransitionMode is set to “TMNone”, it stops at the target point of the first operation and continues the next operation,TMCornerDistance”, the operation continues after stopping at the start and end points of the arc, respectively.
Interpolation operation blending angle limit function works in group operation and coordinate system operation. However, when the coordinate system 3D circular interpolation command is combined, the interpolation operation blending angle limit function does not work.

(1) nterpolation operation blending angle limit function setting
The setting of the interpolation operation blending angle limit function is applied by setting the axis group parameter item.

Name	Description	Operation condition
MC_WriteParameter	Axis group parameter setting	Edge



No.	Parameter	Item	Note
45	Standard setting	Interpolation operation blending angle limit	0: Disable, 1: Able
46	Standard setting	Interpolation operation blending allowable angle	2 ~ 178

8.4.12 3D Circular Interpolation Operation for Coordinate System

Coordinate system-based circular interpolation operation is performed, where the TCP moves in a circular trajectory on the 3D plane using the designated axis in the axes group. There are three kinds of methods for circular interpolation such as midpoint method that passes through the position specified in auxiliary point, center point method that considers the position specified in auxiliary point as center point and radius method that takes the value specified in auxiliary point as the radius of an arc depending on 'CircMode' settings and auxiliary points. To stop the current interpolation control use MC_GroupHalt or MC_GroupStop motion function block.

(1) EndPoint/AuxPoint

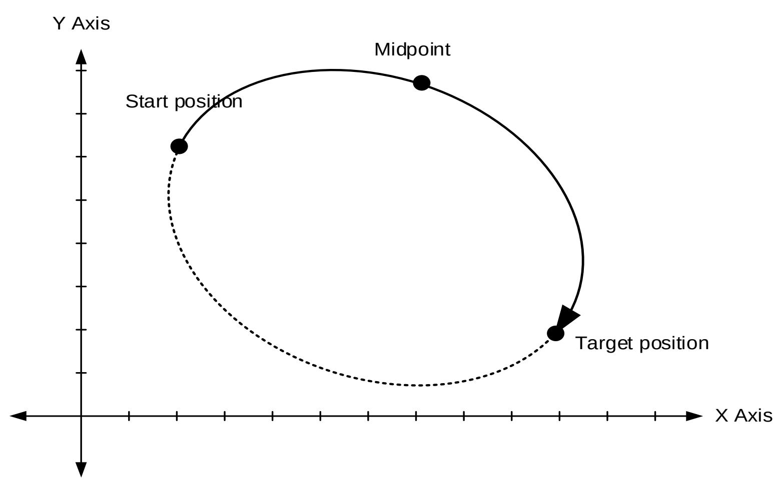
In case of coordinate system circular interpolation control, enter the Px,Py,Pz of TCP to EndPoint/AuxPoint. The RotA, RotB, RotC values, which determine the TCP posture, is not entered, instead maintaining the values at the start position.

(2) Circular interpolation with middle point designation form.

(a) Circular interpolation is executed from starting position to target position through midpoint position set in auxilia ry point.

(b) It determines the plane on which 3D circular interpolation will be performed with the starting position, midpoint, and target position.

(c) Movement direction is decided automatically depends on set target position and auxiliary point of circular interp olation.

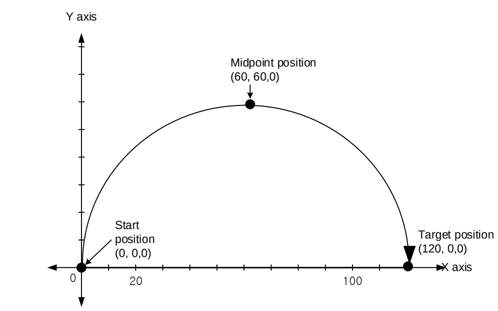
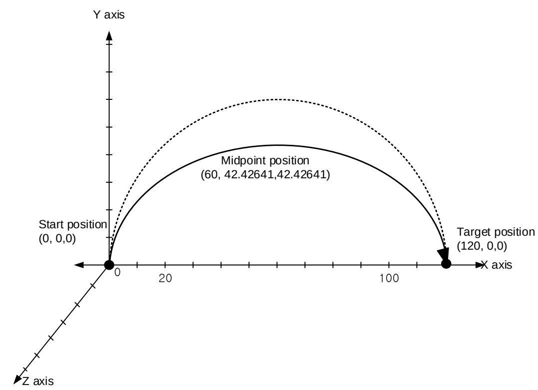
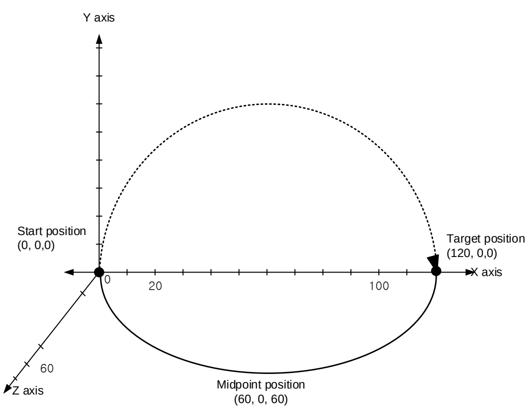
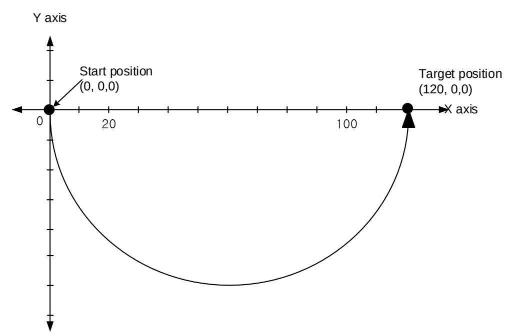


(d) Restrictions

Circular interpolation control using mid-point specification method cannot be performed in case of the following errors.

- During absolute coordinate circular interpolation, home position has not been determined in one or more of the component axes (Error Code: 0x20A0)
- In case the midpoint specified as auxiliary point is the same as the starting position or target position (error code: 0x20A4)
- In case starting position is the same as the target position (error code: 0x20A5)
- In case start position, auxiliary point position and target position are in a straight line (error code: 0x20A7)
- In case there is an axis in infinite running repetition operation among configuration axes (error code: 0x20A8)

(e) Operating Pattern

<table>
<tr>
<td>Start position:(0.0, 0.0, 0.0)
Target position:(120.0, 0.0, 0.0)
Midpoint position:(60.0, 60.0)
CircMode: Midpoint(0)

</td>
<td>Start position:(0.0, 0.0, 0.0)
Target position:(120.0, 0.0, 0.0)
Midpoint position:(60.0, 42.4561, 42.45641)
CircMode: Midpoint(0)

</td>
</tr>
<tr>
<td>Start position:(0.0, 0.0, 0.0)
Target position:(120.0, 0.0, 0.0)
Midpoint position:(60.0, 0, 60.0)
CircMode: Midpoint(0)

</td>
<td>Start position:(0.0, 0.0, 0.0)
Target position:(120.0, 0.0, 0)
Midpoint position:(60.0, -60.0, 0.0)
CircMode: Midpoint(0)

</td>
</tr>
</table>

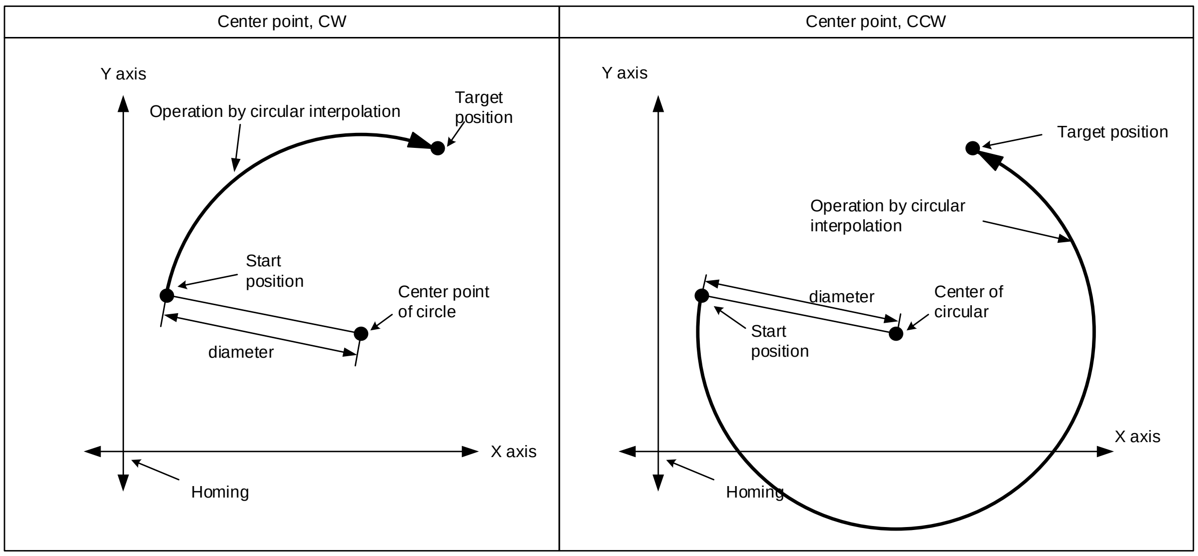
(3) Circular interpolation with center point designation form

(a) Circular interpolation is performed by starting at the start position, and reaching the target position in a circular trajectory of which the diameter is the distance to the designated center point.

(b) It determines the plane on which 3D circular interpolation will be performed with the starting position middle poi nt and target position. The starting position and the center point target position cannot be set to be located i n a straight line.

(c) The movement direction is determined by the direction set in “PathChoice” of the motion function block.

- 0:「CW」 - perform circular interpolation in the clockwise direction from the start position.
- 1:「CCW」- perform circular interpolation in the counter-clockwise direction from the start position.

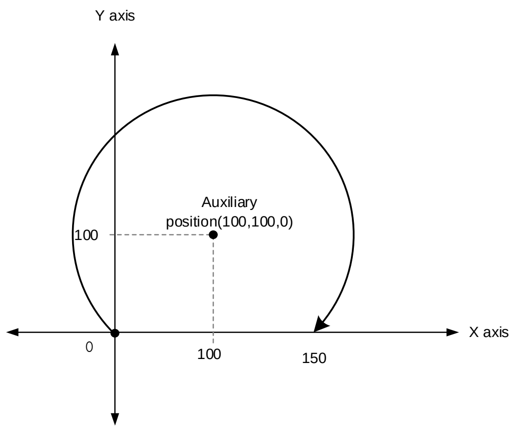
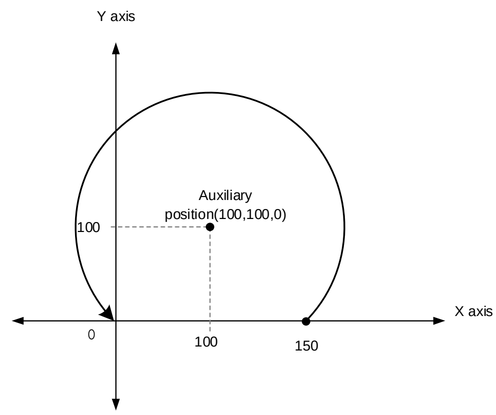
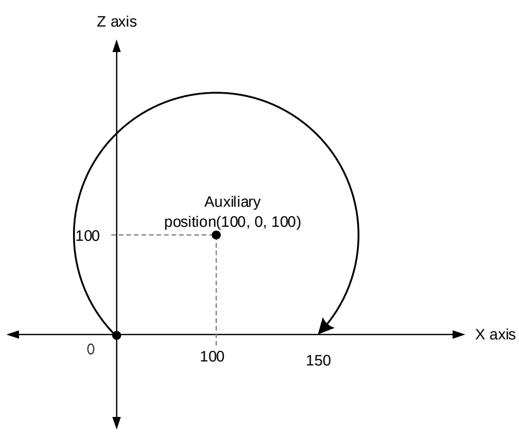
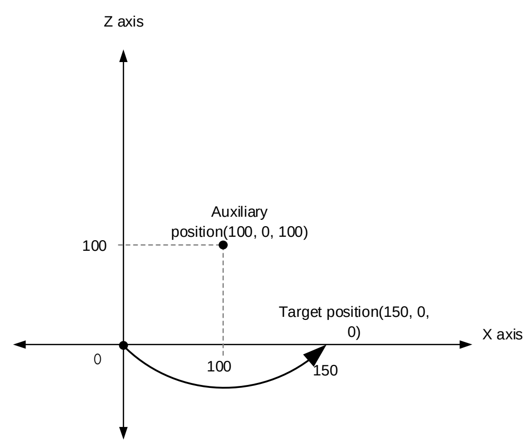


(d) Restrictions

Circular interpolation control using center point specification method cannot be performed in case of the following errors.

- During absolute coordinate circular interpolation, home position has not been determined in one or more of the component axes (Error Code: 0x20A0)
- In case the midpoint specified as auxiliary point is the same as the starting position or target position (error code: 0x20A4)
- In case start position, auxiliary point position and target position are in a straight line (error code: 0x20A7)
- In case there is an axis in infinite running repetition operation among configuration axes (error code: 0x20A8)

(e) Operating Pattern

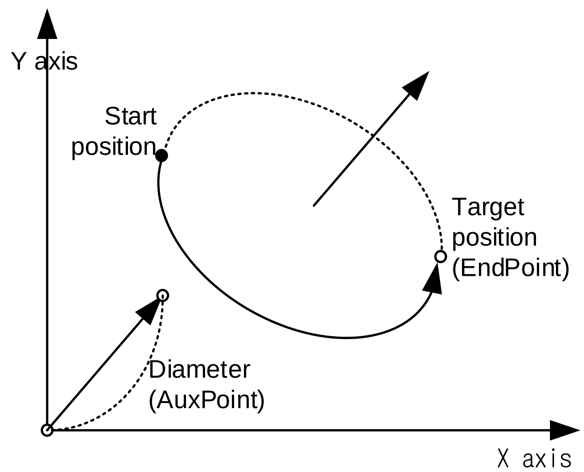
<table>
<tr>
<td>Start position:(0.0, 0.0, 0.0)
Target position:(150.0, 0.0, 0.0)
Auxiliary position:(100.0, 100.0, 0.0)
CircMode: Center Point(1)
PathChoice: CW(0)

</td>
<td>Start position:(150.0, 0.0, 0.0)
Target position:(0.0, 0.0, 0.0)
Auxiliary position:(100.0, 100.0, 0.0)
CircMode: Center Point(1)
PathChoice: CW(0)

</td>
</tr>
<tr>
<td>Start position:(0.0, 0.0, 0.0)
Target position:(150.0, 0.0, 0.0)
Auxiliary position:(100.0, 100.0, 0.0)
CircMode: Center Point(1)
PathChoice: CW(0)

</td>
<td>Start position:(0.0, 0.0, 0.0)
Target position:(150.0, 0.0, 0.0)
Auxiliary position:(100.0, 100.0, 0.0)
CircMode: Center Point(1)
PathChoice: CCW(1)

</td>
</tr>
</table>

(4) Circular interpolation with radius designation form

(a) Starts operation from the starting position and performs circular interpolation up to the target position along the trajectory of the arc whose radius is the length value of the vector set at the circular interpolation auxiliary p oint. The plane to perform circular interpolation is determined according to the vector set at the auxiliary point.

(b) In circular interpolation of radius specification method, the target position cannot be set the same as starting p osition.

(c) The auxiliary point represents the vertical vector of the arc, and the vector value is set according to the right-hand rule. The length of the vector represents the radius.



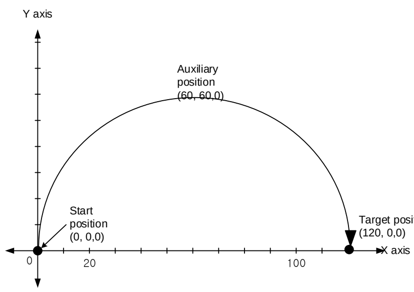
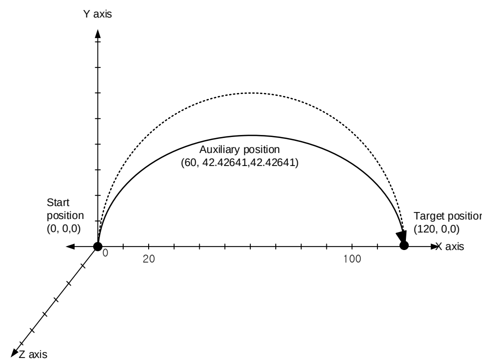
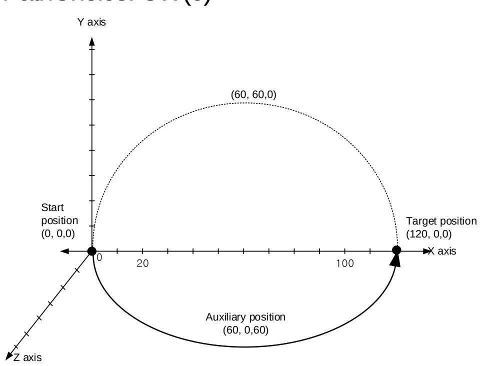
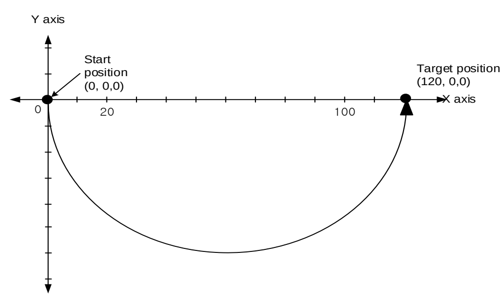
(d) The movement direction and the size of the arc are determined by the direction set in the absolute position c oordinate system circular interpolation operation (MC_MoveCircularAbsolute3D), the relative position coordinate s ystem circular interpolation operation (MC_MoveCircularRelative3D), or “PathChoice” of the motion function bloc k.

(e) Restrictions

Circular interpolation control by radius specification method cannot be performed in case of the following errors.

- During absolute coordinate circular interpolation, home position has not been determined in one or more of the component axes (Error Code: 0x20A0)
- In case starting position is the same as the target position (error code: 0x20A5)
- In case there is an axis in infinite running repetition operation among configuration axes (error code: 0x20A8)

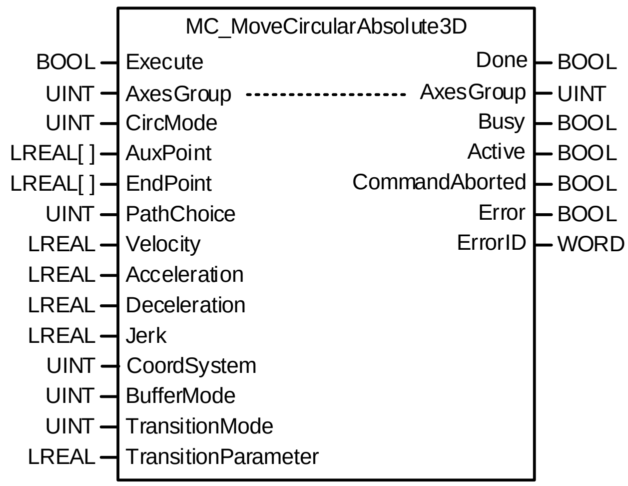
(f) Operating Pattern

<table>
<tr>
<td>Start position:(0.0, 0.0, 0.0)
Target position:(120.0, 0.0, 0)
Auxiliary position:(0.0, 0.0, 60.0)
CircMode: Center Point(2)
PathChoice: CW(0)

</td>
<td>Start position:(0.0, 0.0, 0.0)
Target position:(120.0, 0.0, 0)
Auxiliary position:(0.0, -42.42641, 42.42641)
CircMode: Center Point(2)
PathChoice: CW(0)

</td>
</tr>
<tr>
<td>Start position:(0.0, 0.0, 0.0)
Target position:(120.0, 0.0, 0)
Auxiliary position:(0.0, -60.0, 0.0)
CircMode: Center Point(2)
PathChoice: CW(0)

</td>
<td>Start position:(0.0, 0.0, 0.0)
Target position:(120.0, 0.0, 0)
Auxiliary position:(0.0, 0.0, -60.0)
CircMode: Center Point(2)
PathChoice: CW(0)

</td>
</tr>
</table>

(5) Relevant motion function block

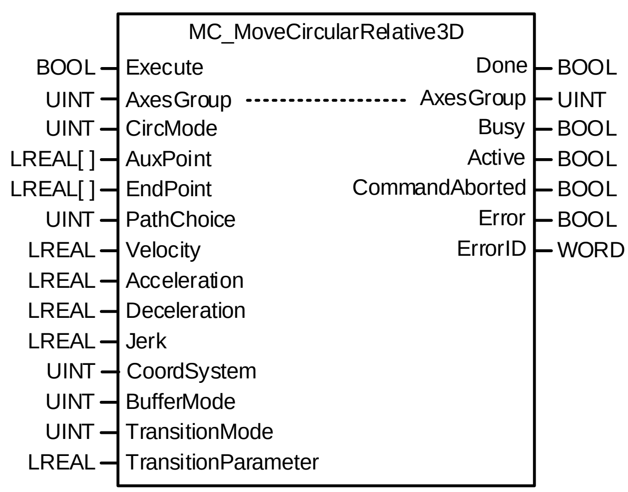
(a) Absolute position coordinate system 3D circular interpolation operation

Name	Description	Operation condition
MC_MoveCircularAbsolute3D	Absolute position 3D circular interpolation operation	Edge



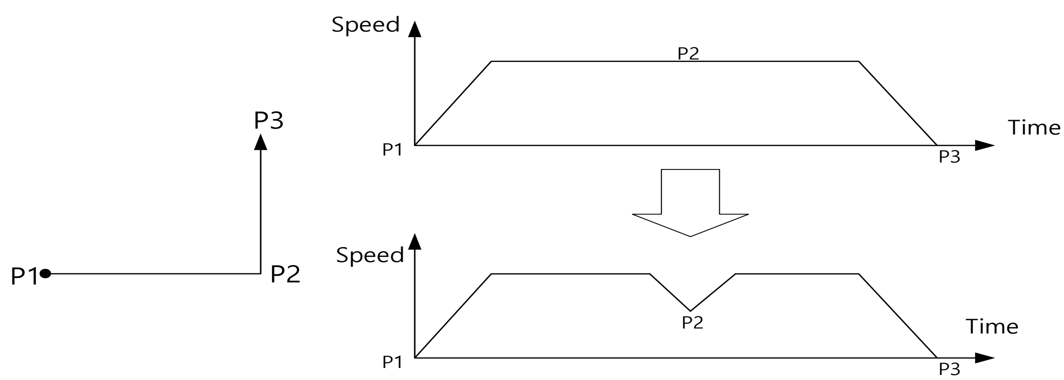
(b) Incremental position coordinate system 3D circular interpolation operation

Name	Description	Operation condition
MC_MoveCircularRelative3D	Incremental position 3D circular interpolation operation	Edge



8.4.13 Coordinate system operation maximum acceleration limit function

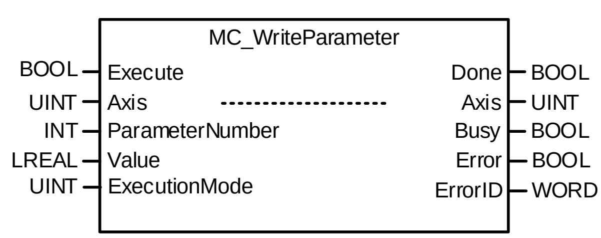
The maximum acceleration limit function of coordinate system operation is when the operation exceeding the maximum acceleration set for each axis is performed according to the driving path when the BufferMode is set to Blending and the coordinate system operation is continuously performed, It is a function to operate by limiting the speed of coordinate system operation so as not to exceed the maximum acceleration. When the coordinate system operation maximum acceleration limit function is not used If the coordinate system operation is performed continuously at the same speed along the path of P1 to P3 as shown below, the speed at the point P2 is kept constant, but the maximum acceleration limit function of the coordinate system operation is not used. In operation in this state, deceleration is performed in the P2 section so as not to exceed the maximum acceleration of the axis.



(1) Coordinate system operation maximum acceleration setting

The maximum acceleration limit of the coordinate system operation is applied by setting the axis parameter items of the coordinate system components. Coordinate system operation maximum acceleration setting / coordinate system operation maximum deceleration setting value of axis parameter can be set separately, when it is set to 0, it is set to not use the maximum acceleration limit function of the coordinate system operation.

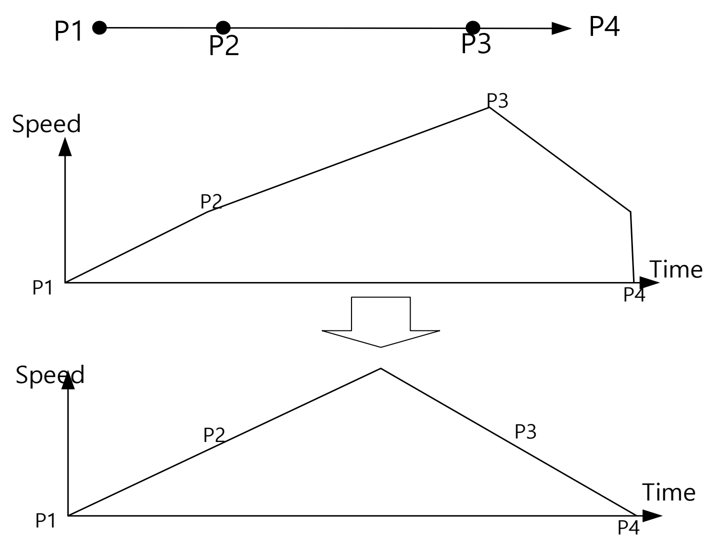
Name	Description	Operation condition
MC_WriteParameter	Axis Parameter Setting	Edge



No.	Parameter	Item	Note
48	Standard setting	Coordinate Systems Operation maximum allowable acceleration	0 or Long real(LREAL) Positive number
49		Coordinate Systems Operation maximum allowable deceleration	0 or Long real(LREAL) Positive number

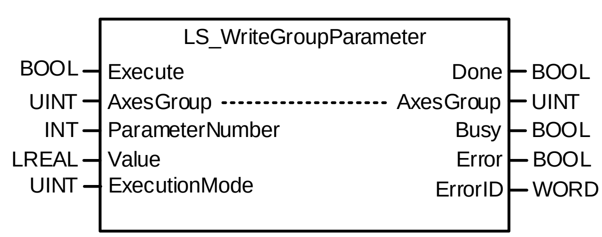
8.4.14 Coordinate system operation look ahead function

The coordinate system operation look-ahead function is used when the BufferMode is set to Blending in the coordinate system operation and continuous coordinate system operation is performed,it is a function to determine whether the current operation is accelerated/decelerated by judging the movement distance/speed/acceleration/jerk of the next command executed in Buffered. If the operation speed is set at a speed higher than the movement distance and acceleration/deceleration in the P1~P4 section without using the look-ahead function of the coordinate system and continuous operation is performed, the deceleration section is lower than the deceleration in the last P3~P4 section. It is short, so there may be a problem that the deceleration cannot be sufficiently slow. If you use the coordinate system look-ahead function, you can solve the problem that acceleration/deceleration cannot be performed because there is not enough moving distance to perform acceleration/deceleration when a short distance is set at a high speed during coordinate system operation and continuous operation is performed.



(1) Coordinate system operation look ahead setting
The coordinate system operation look-ahead setting is applied by setting the axis group parameter item.

Name	Description	Operation condition
MC_WriteParameter	Axis group parameter setting	Edge



No.	Parameter	Item	Note
42	Standard setting	Coordinate system operation look ahead settings	1~10

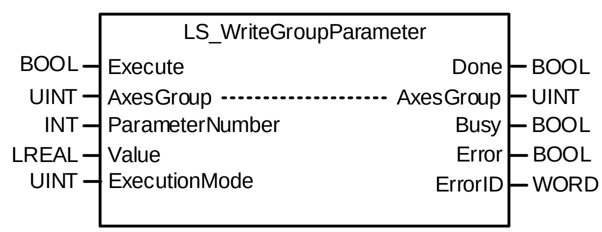
8.4.15 Coordinate system operation output filter function

Coordinate system operation output filter function sets a filter on the output of coordinate system operation, it is a function to perform smoother operation by alleviating the acceleration shock generated on the shaft. By using the coordinate system operation output filter function, it reduces the impact and noise that are added to the axis when performing the coordinate system operation. If the coordinate system operation output filter function is used, the set target position may not be passed because the operation of the vertex trajectory where the acceleration shock occurs is corrected in a round shape. As the set coordinate system operation output filter time constant increases, the degree of roundness of the vertex trajectory also increases.

(1) Coordinate system operation output filter setting

Coordinate system operation output filter setting is applied by setting the axis group parameter item. If set to 0, it is set to not use the coordinate system output filter function.

Name	Description	Operation condition
MC_WriteParameter	Axis group parameter setting	Edge



No.	Parameter	Item	Note
43	Standard setting	Coordinate system output filter function.	0 ~ 100ms

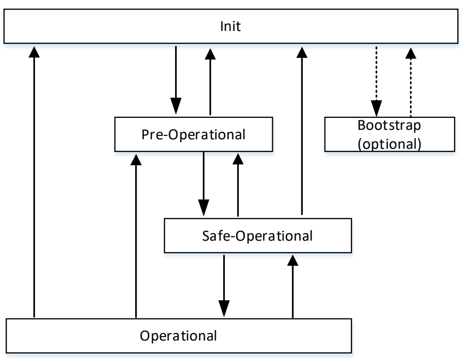
8.5 FoE (File Access over EtherCAT) Function

8.5.1 Overview of FoE Function

FoE is a function that supports firmware download from the motion controller to the slave which is in bootloader state through the EtherCAT network as a simple file access protocol provided by EtherCAT communication. In order to use the FoE function, both master and slave should support the FoE protocol. Therefore, it is necessary to check whether the FoE is supported prior to using the function.

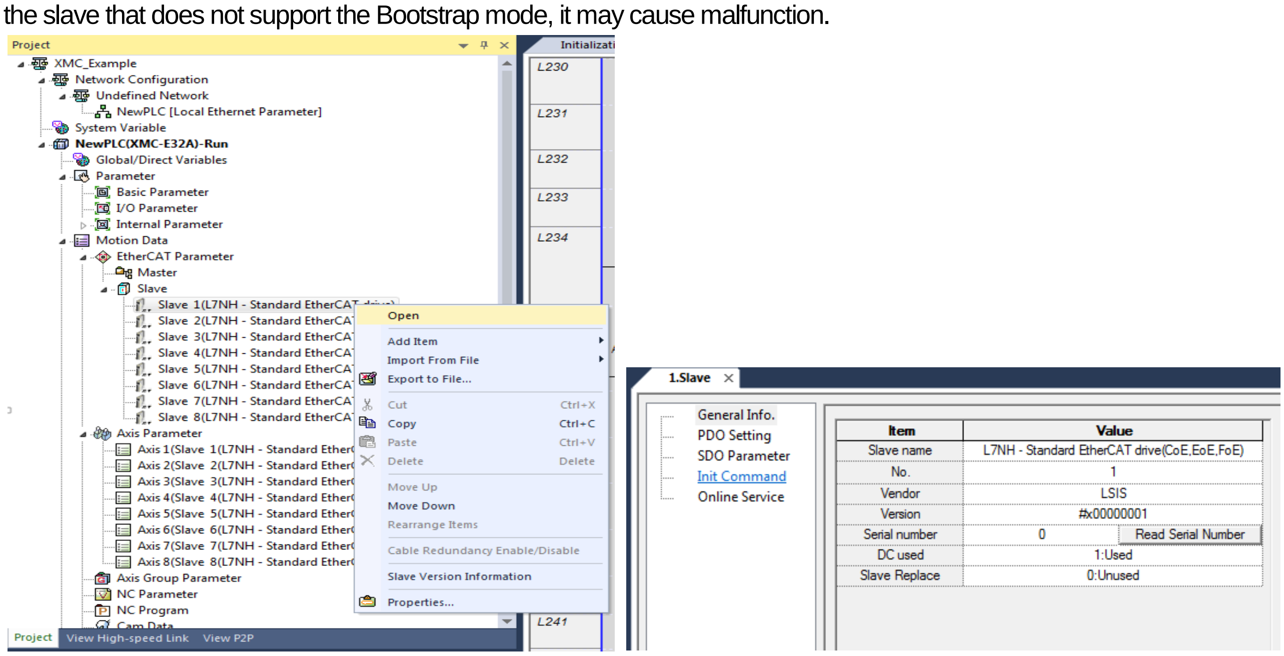
8.5.2 FoE Download

The slave operates as a state machine depending on the functions it actually supports. Since the FoE function is supported only in Boot (Bootstrap) mode, which is the bootloader state, the salve state should be converted to the Boot mode in order to use the FoE function. If the motion controller performs a network connection with full servo connection command, the state of all the connected slaves will be changed from Init to Op (Operational) mode. Therefore, the mode should be switched in the order of Op->Init->Boot to switch from the Op mode to the Boot mode. The FoE download is executed while the Boot mode is running. After the FoE download is completed, you should perform a mode switch to Boot->Init again



(1) StateMachine setting

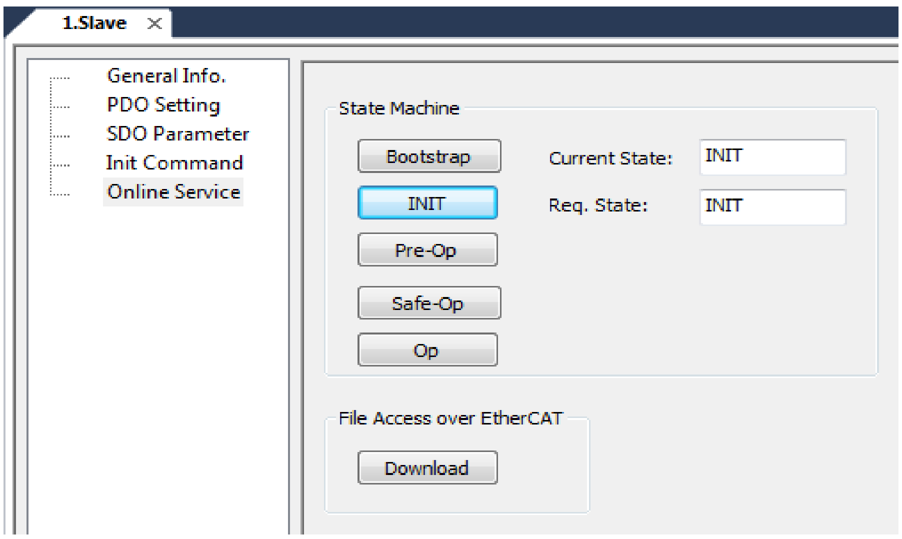
The StateMachine setting is executed by selecting the slave in a project tree while the slave is connected, the Shortcut Menus>> Registered Information and the Online Service tab from the slave information dialog box. The current slave's StateMachine state and the entered requests are displayed on the screen. Since the Bootstrap is not supported depending on the slave in most cases, it is necessary to check whether the slave supports the Bootstrap mode. If the Bootstrap mode is set in the slave that does not support the Bootstrap mode, it may cause malfunction.



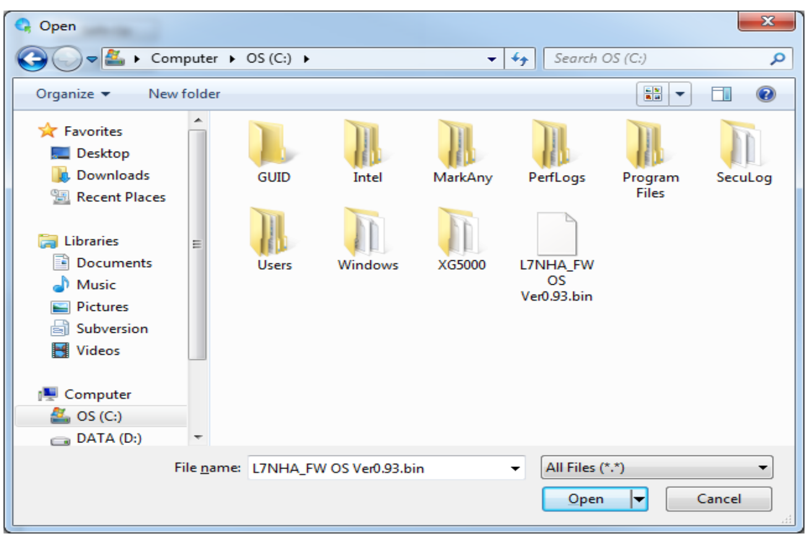
(2) file download

Download a file using the FoE protocol. FoE download can be executed when the StateMachine state is in Boot model. The procedures for downloading the FoE files may vary depending on the slave.

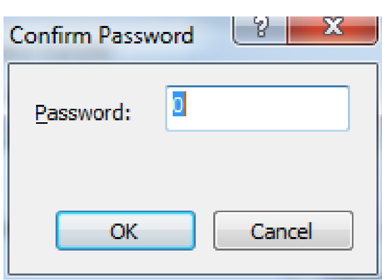
1) Please refer to the salve instruction manual. Change OpMode to Boot and click the Download button in the FoE to start the download.



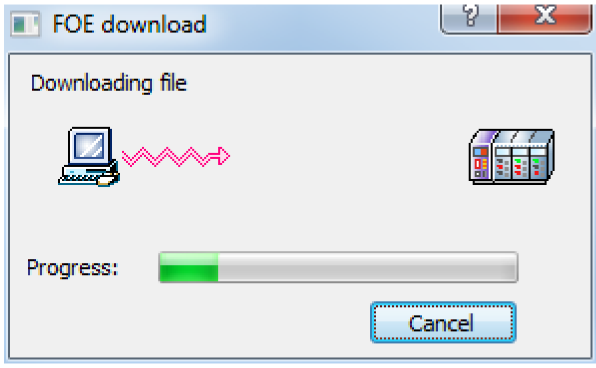
2) Select the file you want to download from the open dialog box.



3) Enter the password (number) in the password confirmation dialog box.



4) Select the OK button to download the file.



5) When the download is completed, change the StateMachine state to Init.

8.6 CAM data function expansion

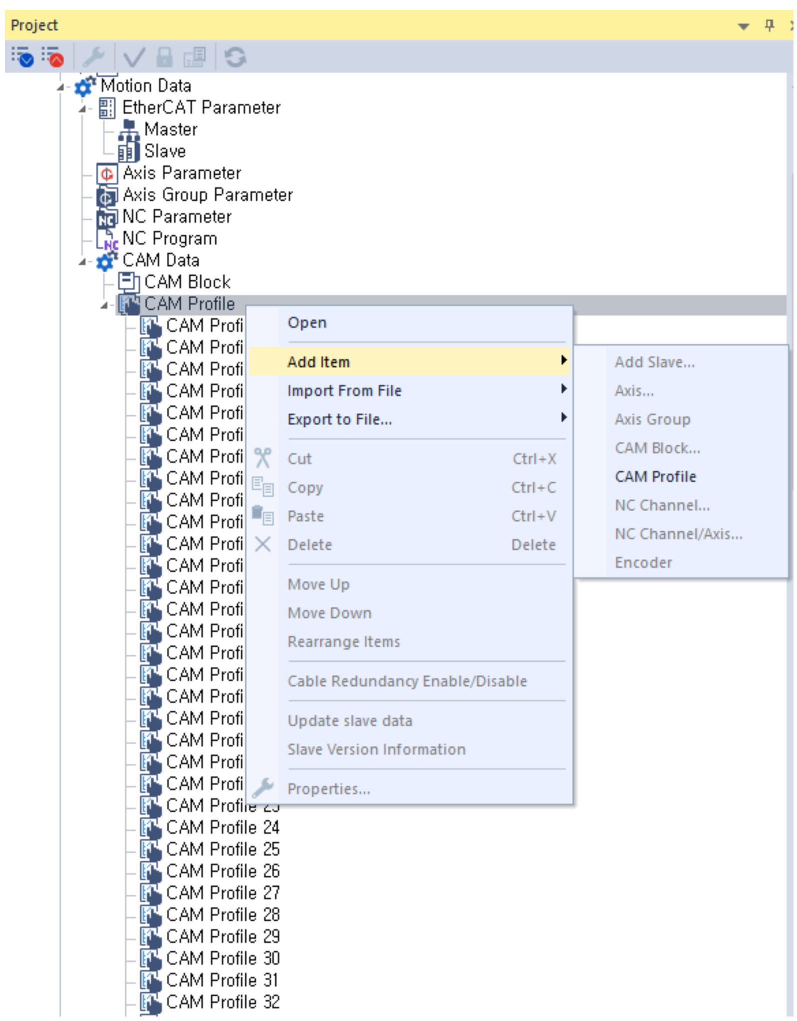
8.6.1 Cam Data expansion overview

Camp profile can be created by editing cam profile of XG-5000 or LS_WriteCamData and LS_GenerateCamData of function block. Previously, the number of cam profiles could be up to 32 and the number of cam points could be up to 32,768. In the Cam Profile and Cam Point Expansion Update Standard, 100 cam profiles and 65,536 cam points are possible. This cam data extension chapter describes the basic guide for cam profile creation based on the extension standard and the mutual compatibility of the previous/extended XMC controller and XG-5000. You can check the cam profile creation method in XG5000 and the creation method using function block instructions in Chapters 6, 7, 8 of this manual and in the XG5000 manual. That is, it is the same as the way it was previously worked. Therefore, it describes the range of parameters related to profile creation and compatibility with previous standards. Therefore, it describes the range of parameters related to profile creation and compatibility with previous standards.

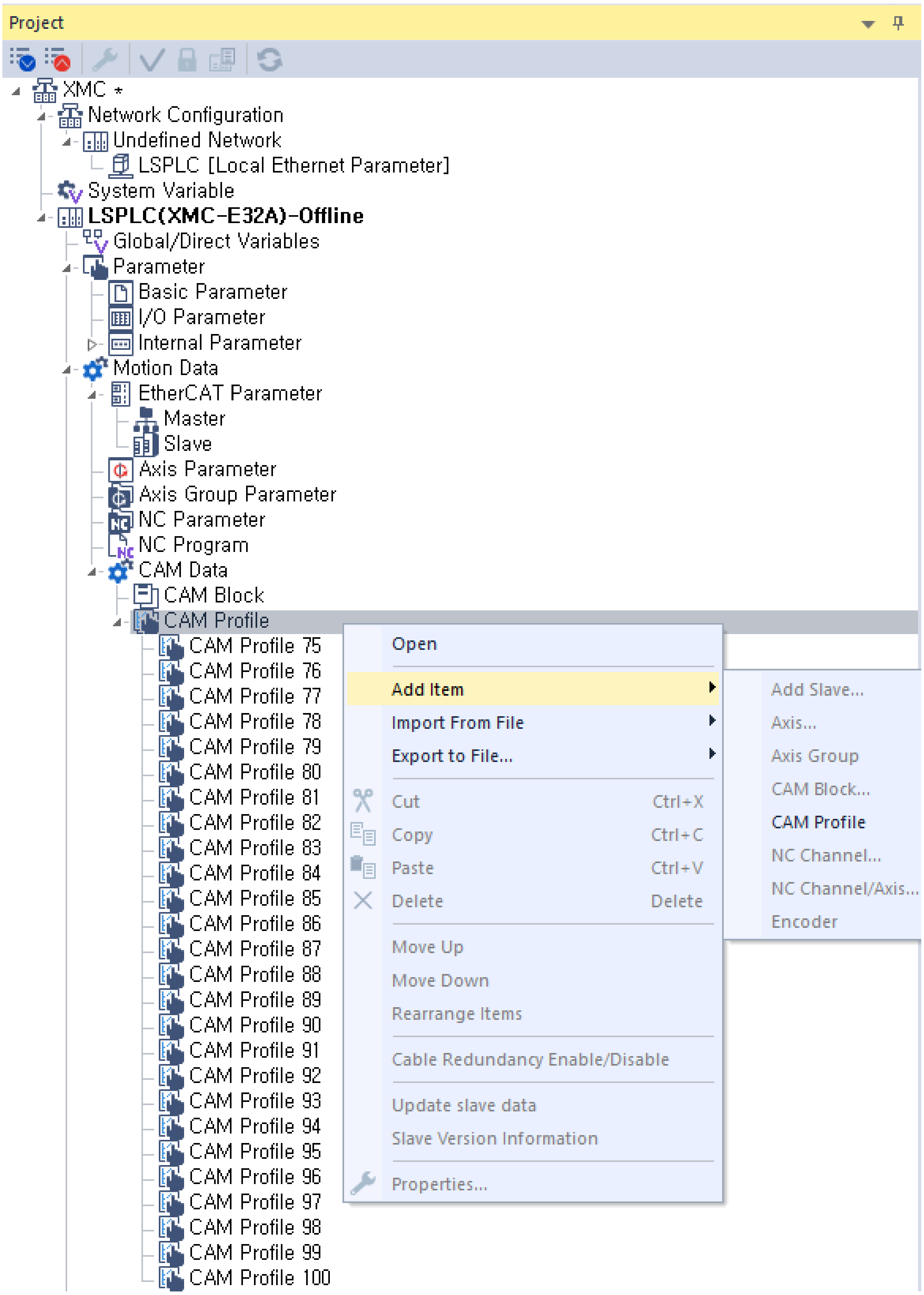
8.6.2 Cam Data expansion

(1) Cam profile expansion

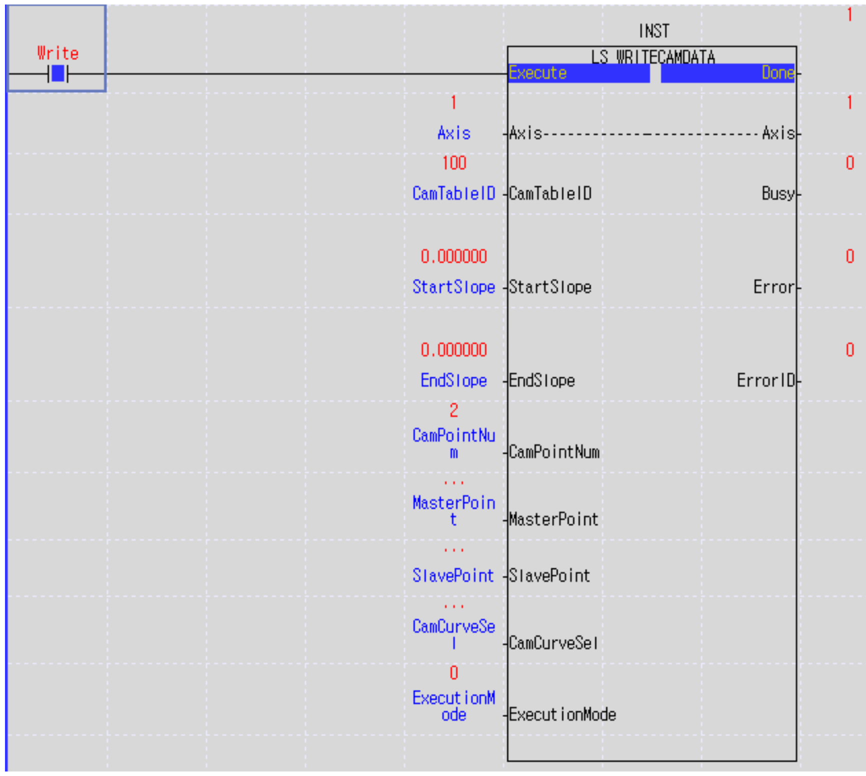
For camp profile, select the cam table from the cam data of the project tree, right-click and select Add Item -> camp profile to add the camp profile. Previously, up to 32 camp log files could be added.



However, up to 100 cam profiles can be created in the cam table extended update standard.



In the case of commands, up to 32 cam table IDs can be entered for LS_WriteCamData/LS_GenerateCamData. However, in the new standard, up to 100 can be entered.



(2) Cam Point Expansion

Cam point is added by selecting the cam block from the cam data of the project tree, right-clicking the mouse, and selecting Add Item -> Cam Block. If you select the applied cam block and open the window, it appears as shown in the figure below. Previously, up to 32768 camp points could be added.

[Number of CAM nodes] 271 / 360 [Number of CAM points] 24301 / 32768 CAM Profile Unchecke Characteristic Curve data view Export

	Main Axis Aosition	Sub Axis Position	CAM Curve	Connection Speed	Connection Acceleration	Connection Jerk	Number of points	Interpolation Type
1	0.00000	0.00000	Straight Line Modified Si				1	Linear
2	1.00000	1.00000	Straight Line Modified Si				90	Linear
3	2.00000	2.00000	Straight Line Modified Si				90	Linear
4	3.00000	3.00000	Straight Line Modified Si				90	Linear
5	4.00000	4.00000	Straight Line Modified Si				90	Linear
6	5.00000	5.00000	Straight Line Modified Si				90	Linear
7	6.00000	6.00000	Straight Line Modified Si				90	Linear

Up to 65,536 cam points can be input in the extended update standard.

Setting value: 100

(Max count : 65536, Remain count : 65536)

	Main Ax. pos.	Sub. Ax. pos.	Intp. method
1	0.00000	0.00000	Linear
2	1.00000	1.00000	Linear
3	2.00000	2.00000	Linear
4	3.00000	3.00000	Linear
5	4.00000	4.00000	Linear
6	5.00000	5.00000	Linear
7	6.00000	6.00000	Linear
8	7.00000	7.00000	Linear
9	8.00000	8.00000	Linear
10	9.00000	9.00000	Linear
11	10.00000	10.00000	Linear

In the case of commands, LS_WriteCamData/LS_GenerateCamData can input up to 32,768 cam points. However, in the new standard, up to 65,536 can be entered.

8.6.3 Cam data extension function compatibility check (XG5000 / XMC)

The standards defined to maintain compatibility between the XG5000 and XMC controllers of the updated standard according to the cam data extension and the XG5000 and XMC controllers of the standard before the update are as follows. Depending on the total number of four cases, the range implemented when creating a cam profile is different, and the number of cases is as follows. In 8.6.3, the scope of function implementation according to the number of four cases is described.

(1) XG5000 (Expansion specifications) / XMC(Expansion specifications)

The first case is when a cam profile is created using the updated XG5000 as an extension standard and read/write is performed on the updated XMC. As shown in the figure in 8.6.2, in the extended specification XG5000, up to 100 profiles can be created and read/write to the extended specification XMC controller is possible. However, to maintain compatibility when writing by creating cam profiles and cam points of less than 32, 32,768 cam points, the extended standard XMC controller makes it possible to read from the previous standard XG5000. On the other hand, in case of exceeding 32, 32,768, writing is performed and then reading is attempted by executing the previous standard XG5000, the written cam data cannot be read.

(2) XG5000 (Expansion specifications) / XMC(Previous specifications)

When performing read/write to XMC of the previous standard by creating a cam profile in XG5000 of extended standard, XG5000 checks the updated version so that less than 32, 32,768 cam profiles and cam points can be read and written. When writing more than 32, 32,768 cam profiles and cam points, XG5000 cannot create more than cam data by itself.

(3) XG5000(Previous specifications) / XMC(Expansion specifications)

In XG5000 of the previous standard, it is possible to create up to 32, 32,768 cam profiles and cam points as before. First, when reading/writing to the extended standard XMC, reading is the same as described in (1) at the end of the second half. In other words, it is possible to read from the previous standard XG5000 only when the cam profiles and cam points of 32, 32,768 or less are written to the extended standard XMC. In the case of writing, it is possible to write cam profiles and cam points of up to 32, 32,768 or less, as in the existing method. In addition, if 32 or less cam profiles and 32,768 cam points are written in the extended standard XMC in the extended standard XG5000, it can be read in the previous standard XG5000 On the other hand, if you try to read from the previous standard after writing 32/32768 or more in the extended standard XG5000 to the extended standard XMC, it cannot be read and the profile is not created in the XG5000 cam profile tree.

(4) XG5000 (Previous specifications) / XMC (Previous specifications)

As before, up to 32, 32,768 cam profiles and cam points can be created, and read/write is also possible according to the existing number.

Additionally, regardless of the number of cam profiles, if the cam table ID value is 32 or less, the previous standard is applied, and if it is 33 or more, it is applied as the expanded standard. For example, if only one cam table ID value of 50 is written in XG5000 of extended standard, cam information written in XMC cannot be read in XG5000 of previous standard.

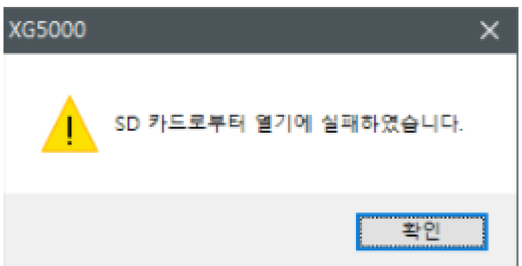
8.6.4 Cam data extension function compatibility check (SD memory card/XG5000/ XMC)

How to use the SD memory card is described in detail in Chapter 12 SD Additional Functions. Please refer to Chapter 12 for how to use it. Even in the case of SD memory cards, the range of use of the import/export function varies according to the applied standards of XG5000 and XMC.

When trying to save the contents of XMC controller or XG5000 to SD card, the contents are as follows.

If the number of created cam profiles exceeds the last cam table ID value of 32 in the extended standard, that is, from 33, the backup is made so that only XG5000 of the extended standard can be read.

If you try to open the backed-up cam profile data with a cam table ID value of 33 or higher in XG5000 of the previous standard, it will fail and an alarm window will appear as shown in the figure below.



Conversely, in case of backing up the contents of which cam profile values of which the number of cam profiles are less than or equal to 32 in the previous standard and extended standard, it is possible to open both in XG5000 of the previous/extended standard.

Taken together, the number of cases is:

case 1. Write a cam profile whose number of cam profiles is less than or equal to 32 in the XG5000 of the previous standard -> Download to the XMC controller of the previous/extended standard -> Attempt to back up to the memory card -> Can be opened in the XG5000 of the previous/extended standard

case 2. Write a cam profile whose number of cam profiles is less than or equal to 32 in the XG5000 of the expansion standard -> Download to the XMC controller of the previous/extended standard -> Attempt to back up to the memory card -> Can be opened in the XG5000 of the previous/extended standard

case 3. Write a cam profile whose number of cam profiles is more than 33 in the XG5000 of the expansion standard -> Download to the XMC controller of the extended standard -> Attempt to back up to the memory card -> Can be opened in the XG5000 of the extended standard

case 4. Write a cam profile whose number of cam profiles is more than 33 in the XG5000 of the previous standard -> Download to the XMC controller of the extended standard -> Attempt to back up to the memory card -> Can not be opened in the XG5000 of the extended standard

case 5. Write a cam profile whose number of cam profiles is less than or equal to 32 in the XG5000 of the previous standard -> Download to the XMC controller of the previous/extended standard -> Attempt to back up XG5000 contents to memory card -> Can be opened in the XG5000 of the previous/extended standard

case 6. Write a cam profile whose number of cam profiles is less than or equal to 32 in the XG5000 of the expansion standard -> Download to the XMC controller of the previous/extended standard -> Attempt to back up XG5000 contents to memory card -> Can be opened in the XG5000 of the previous/extended standard

case 7. Write a cam profile whose number of cam profiles is more than 33 in the XG5000 of the expansion standard -> Download to the XMC controller of the previous/extended standard -> Attempt to back up XG5000 contents to memory card -> Can be opened in the XG5000 of the extended standard

case 8. Write a cam profile whose number of cam profiles is more than 33 in the XG5000 of the expansion standard -> Download to the XMC controller of the previous/extended standard -> Attempt to back up XG5000 contents to memory card -> Can not be opened in the XG5000 of the extended standard

Additionally, regardless of the number of cam profiles, if the cam table ID value is 32 or less, the previous standard is applied, and if it is 33 or more, it is applied as the expanded standard. For example, if only one cam table ID value is set to 50, the SD card is recognized as an extended standard and operates in the manner described in the above case.

8.7 CAM block Functions

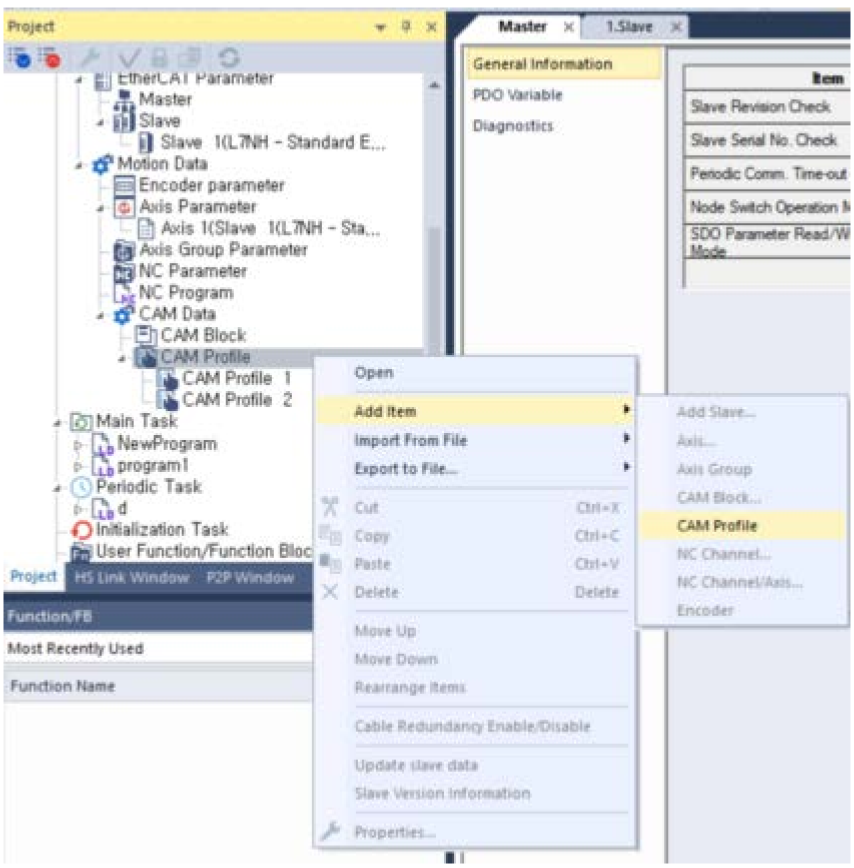
8.7.1 Overview of CAM block Function

The cam block function creates a cam profile by setting the cam curve, connection speed, connection acceleration, and connection jerk. In the cam block function, various cam curves such as StraightLine/ModifiedSine/ModefiedTrapezoid/DoubleHarmonic/TripleHarmonic can be set, up to 100 blocks can be added, 360 nodes can be set per block, and a total of 6000 nodes can be set. You can create a cam profile on the cam block setting screen, edit the cam block while driving, and create a camp profile file from the cam block. Cam block cannot be used for CAM operation alone. To perform cam operation with cam block, you must go through the process of creating a camp profile.

8.7.2 Cam block setting enable

1) Cam block adding

To add a cam block, select the cam block from the cam data of the project tree, right-click the mouse, and select Add Item > Cam Block to add the cam block. Up to 100 cam blocks can be added.



2) CamBlock-Property

	Main Axis Aosition	Sub Axis Position	CAM Curve	Connection Speed	Connection Acceleration	Connection Jerk	Number of points	Interpolation Type
1	0.00000	0.00000	Straight Line Modified Si				1	Linear

Cam operation speed: Input the operation speed of the main axis of the cam block. The entered speed value is used to calculate the maximum speed/maximum acceleration/maximum jerk of the sub axis.

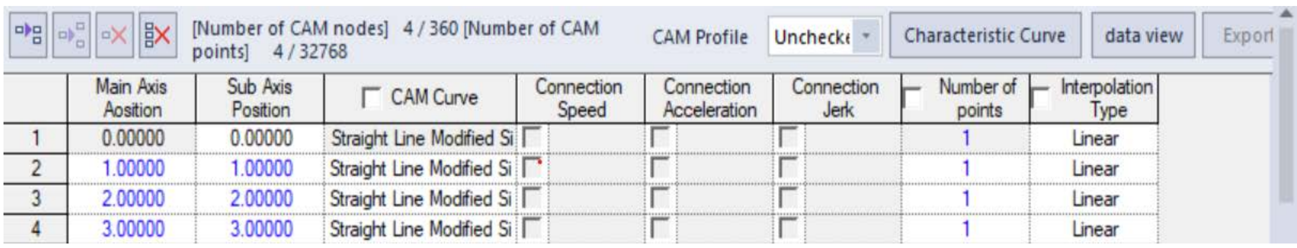
Maximum speed of sub axis: Enter the maximum speed of sub axis. The maximum speed value of the sub axis is used to check whether the maximum speed value is exceeded when setting the cam block and does not affect the creation of the camp profile.

Maximum speed of sub axis: Enter the maximum speed of sub axis. The maximum acceleration of the sub axis is used to check whether the maximum acceleration is exceeded when setting the cam block and does not affect the creation of the camp profile.

Maximum jerk of sub axis: enter the maximum jerk of sub axis. The jerk value of the sub axis is used to check whether the jerk value is exceeded when setting the cam block and does not affect the creation of the camp profile.

3) CamBlock Node Data

Cam block data is composed of each node data. One node data consists of main axis position/sub axis position/cam curve/connection speed/connection acceleration/connection jerk/number of points/interpolation type. Up to 360 nodes can be added to the cam block.



Main axis position: Set the main axis position of the camp profile to be created.

Main axis position: Set the main axis position of the camp profile to be created.

Sub axis position: Set the sub axis position of the camp profile to be created. (Linear/ModifiedSine/ModifiedTrepezoid/…/5thCurve/7thCurve)

Connection speed: Set the speed of the end of the current node. To set the connection speed, the cam curve of the current node or the next node must be set to a cam curve that can set the connection speed, and a check box must be set when setting.

Connection Acceleration: Set the last acceleration of the current node. To set the connection acceleration, the cam curve of the current node or the next node must be set to a cam curve that can set the acceleration speed, and a check box must be set when setting.

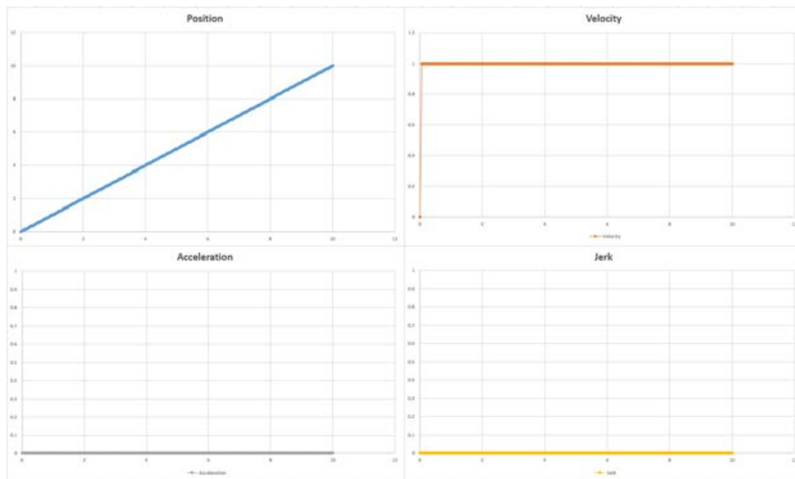
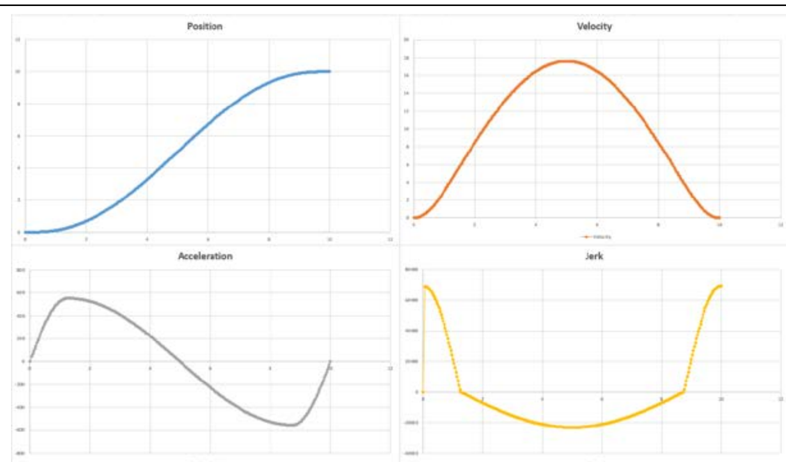
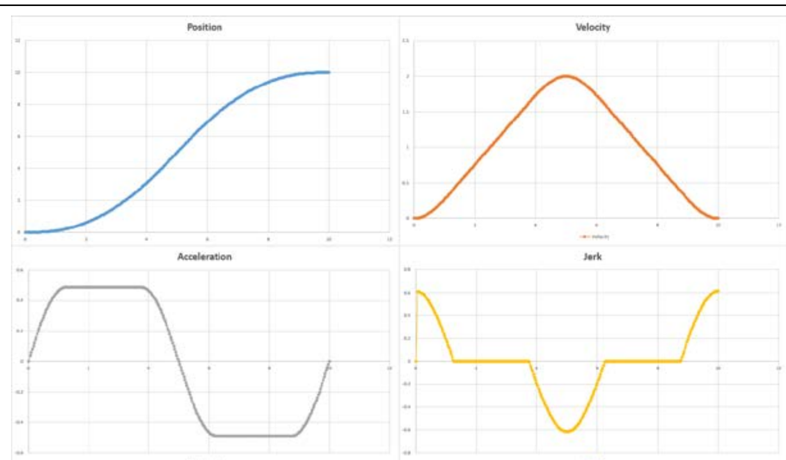
Connection jerk: Set the last jerk of the current node. To set the connection jerk, the cam curve of the current node or the next node must be set to a cam curve that can set the connection jerk, and a check box must be set when setting.

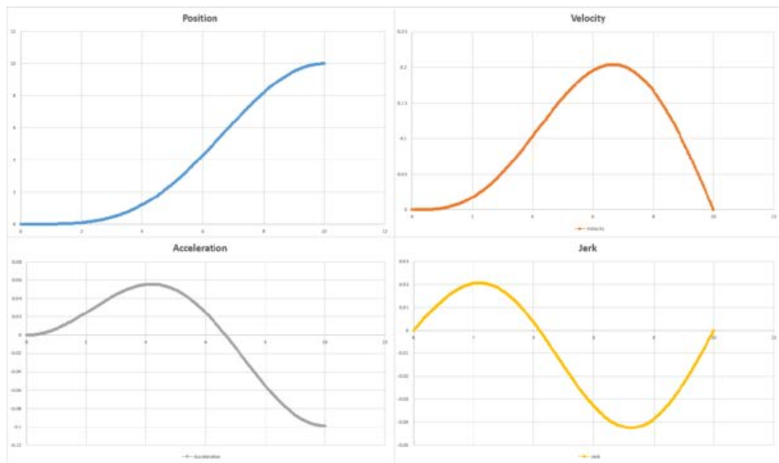
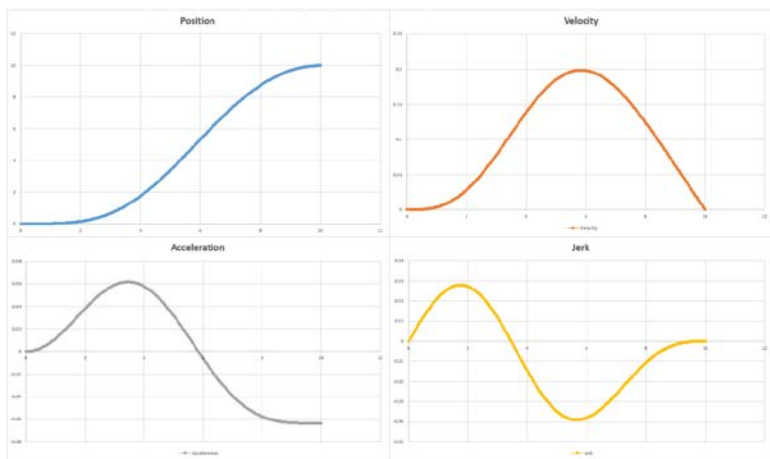
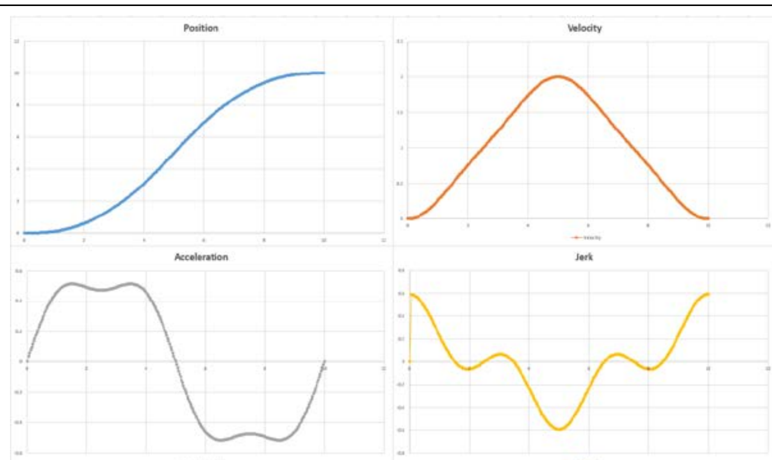
Number of points: Set how many cam points to create the current node when creating a cam profile for cam blocks.

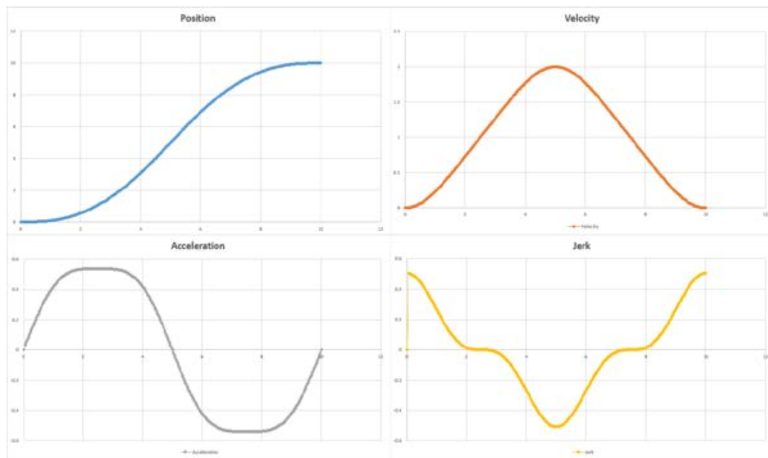
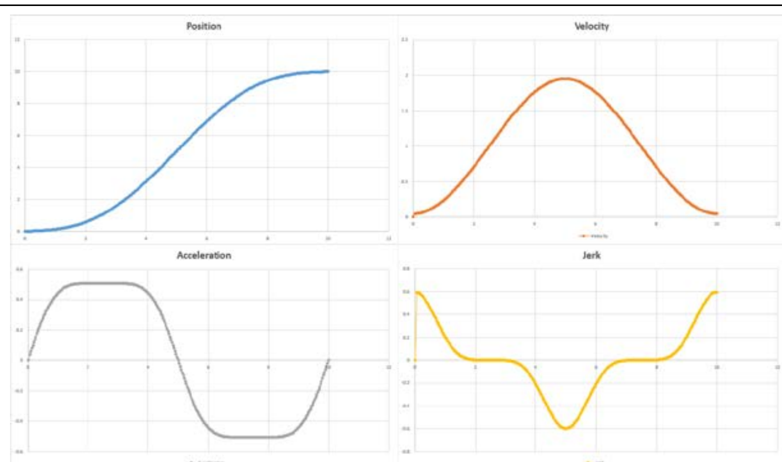
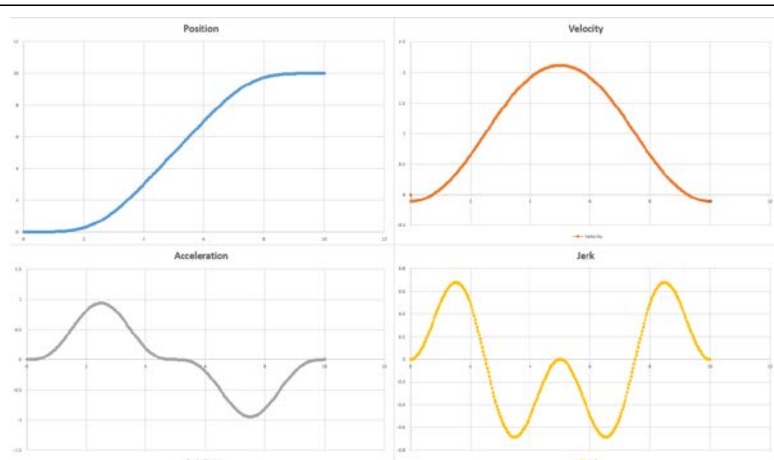
Interpolation type: Set the interpolation type of the camp profile. (Linear/Qubic)

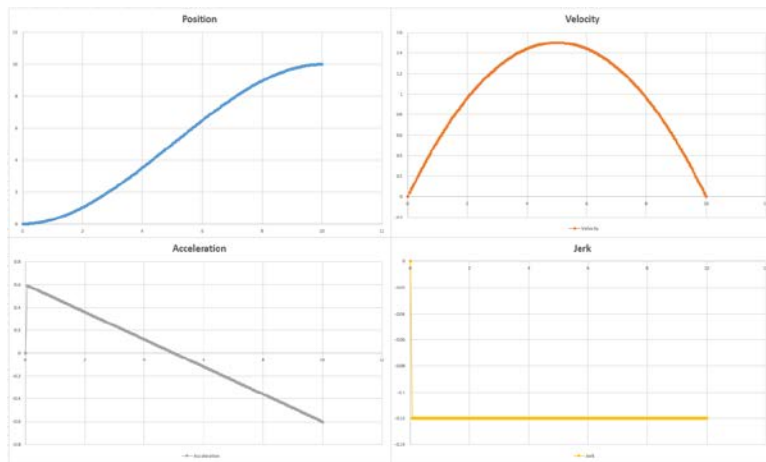
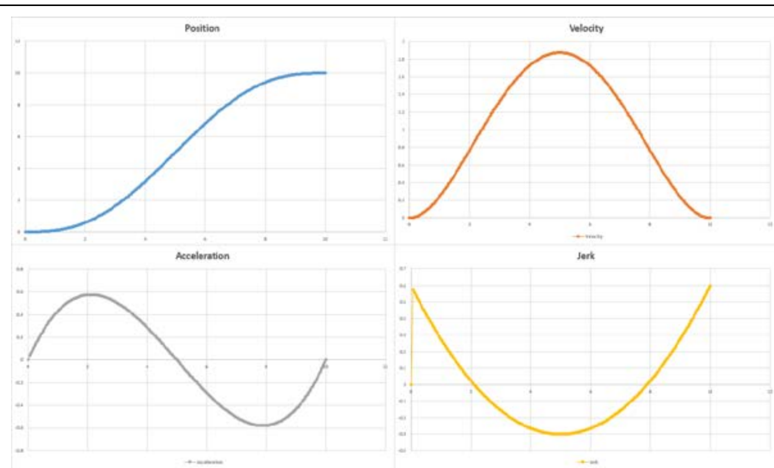
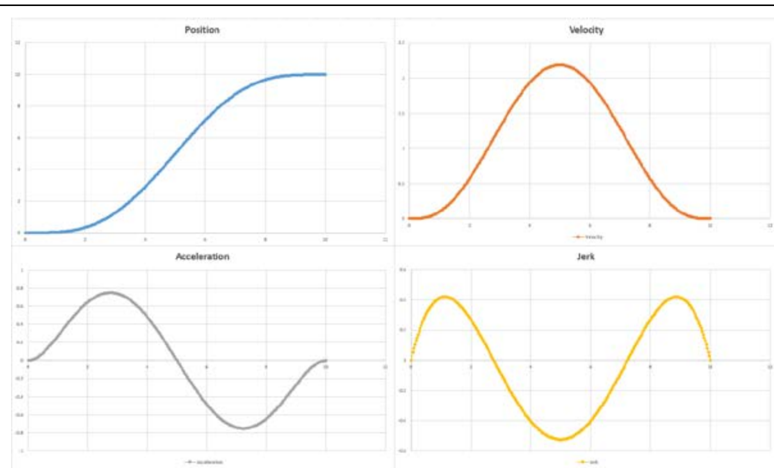
4) Cam curve

There are a total of 12 types of cam curves that can be set in the cam block function. The characteristics of each cam curve are as follows.

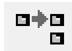
CamCurve	Characteristic curve
0:StraightLine	
1:ModifiedSine	
2:ModifiedTrapezoidal	

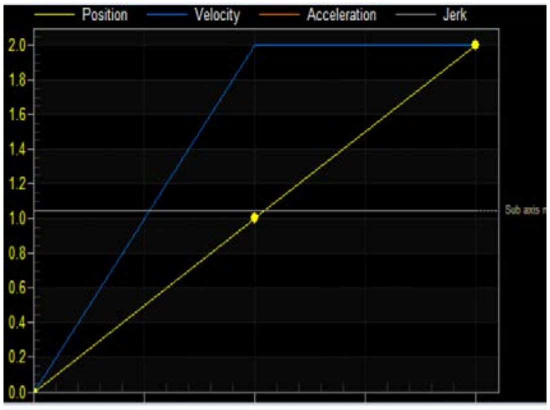
3:DoubleHarmonic	
4:TripleHarmonic	
5: THMTG1-3	

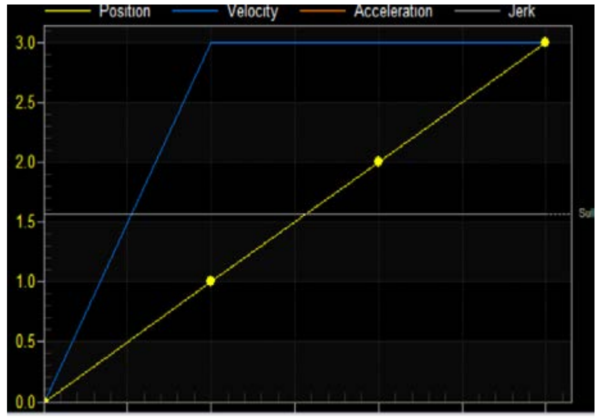
6: THMTF1-3	
7: THMTF1-3-5	
8: TH0J0A	

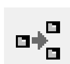
9: 3rd curve	
10: 5th curve	
11: 7th curve	

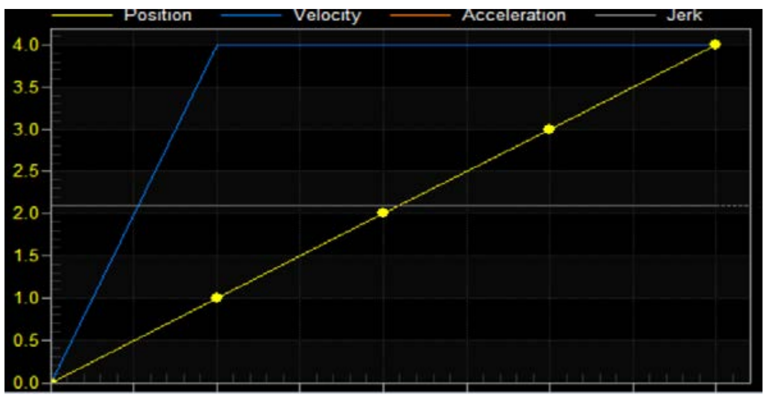
5) Cam block node data adding

If you click the  button on the cam block screen, one last node is added.

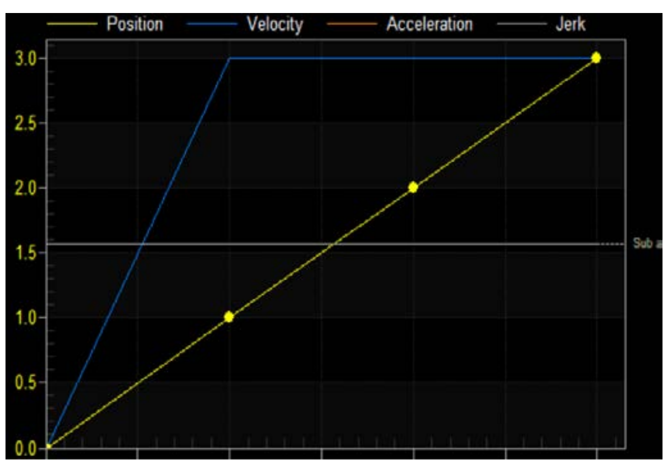




If you click the  button on the cam block screen, one node is added at the selected position.



If you select node data and click  the button, the selected node data is deleted.

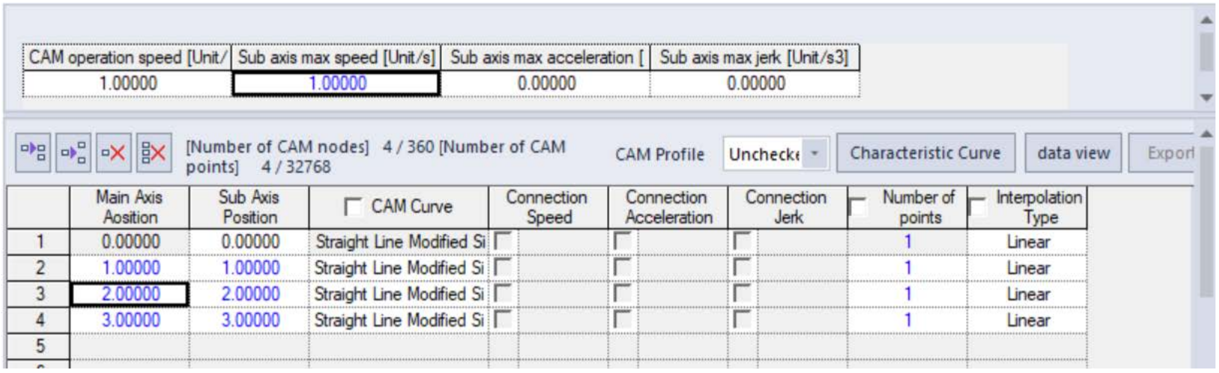


If you select node data and click  the button, the total node data is deleted.

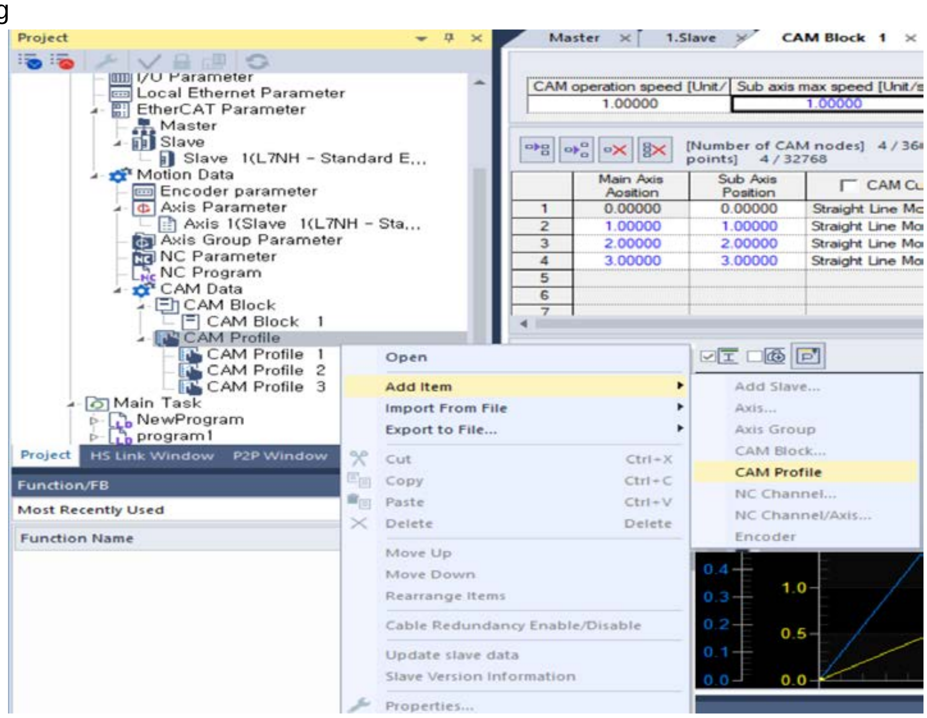
6) Export camp file

In order to perform CAM operation with CAM block data, you must first create a CAM profile from CAM block. The procedure for exporting camblock to camp file is as follows.

(1) CAM block setting



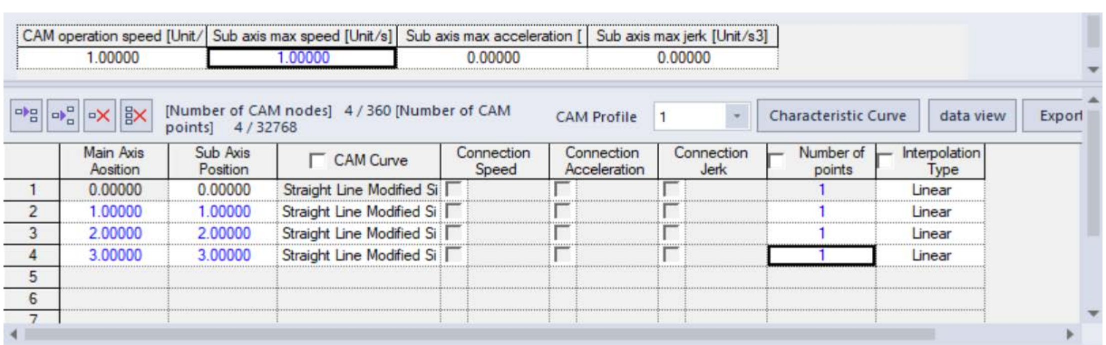
(2) Cam profile adding



1

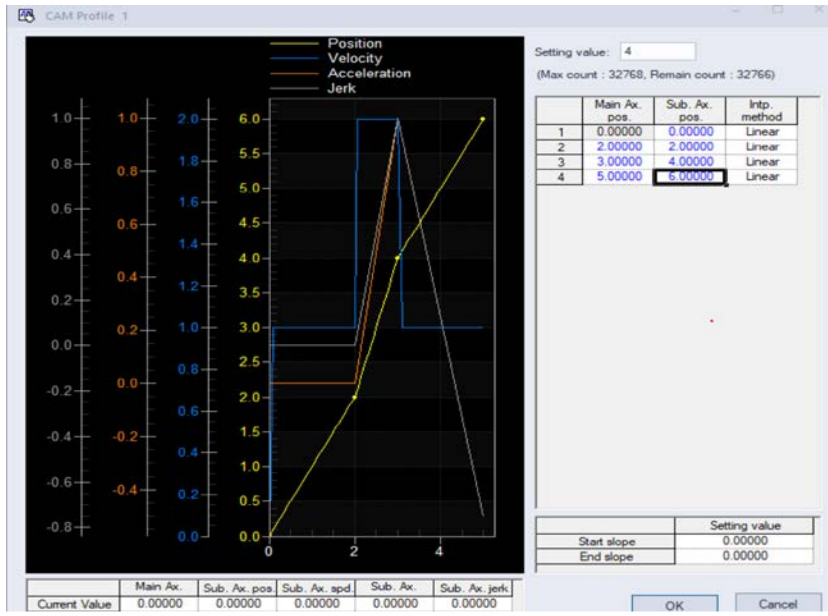
(3) Cam profile export

Select the camp profile file to be created as Camblock and click the Export button.



(4) Check the created cam profile

Open the camp profile to check the created camp profile.



7) Cam block setting function (command)

Provides functions to read/write/save cam blocks and create camp profile files while driving. To create a camp profile file by setting a cam block, first set the cam block properties and the number of nodes using the LS_WriteCamBlockProperty function block. After that, set each node data (main axis position/subordinate axis position/cam curve/number of points, etc.) with LS_WriteCamBlockNode function block. When the cam block node data setting is completed, the camp profile file is created and saved from the cam block data in the designated camp profile file using the LS_GenerateCamData function block. For more information about cam block setting function, refer to Chapter 2 Function Block.

Chapter 9 NC Control Function

Chapter 9 describes how the motion controller user creates the motion program of the G-code format.
The motion controller can program motions through a kind of scripting language called the G code. Chapter 9 describes the basic terms and conceptual explanations for the G code programming, and explains how to configure the program. It also describes the command scripts such as G, M, and F supported by the motion controller.

9.1 NC Commands

9.1.1 NC Command Definition

The motion control is used by the user to move the machine from any position to the desired position at the specified speed or to control the input / output. At this time, the axis that moves the machine will operate in various ways depending on each control environment. It is the motion control program that is used to control the movement of various axes, and in the motion controller, when the G code is used among the control information constituting the motion control program this is called the NC command.

9.1.2 Definition of the Command Character

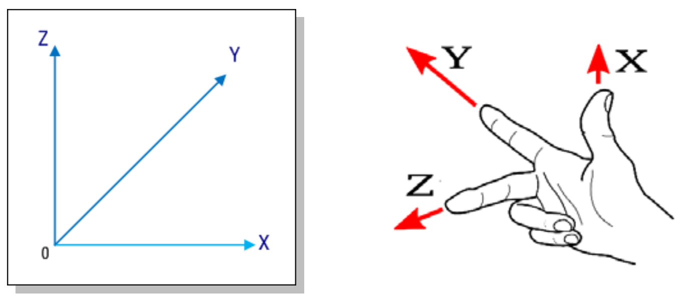
The NC commands of the motion controller have specific Descriptions for certain alphabetical characters. The types and Descriptions of reserved character used in NC commands are as follows.

Reserved character	Meaning
Character set	
(	Start comment
)	End comment
[	Left parenthesis
]	Right parenthesis
+,-,*,/	Arithmetic
A(AND), O(OR)	Comparison operation
=	Assignment operator
0~9	Numeric data
;	Block end
#	Variable
Address character	
X	X axis of the XYZ rectangular coordinate system (Primary Axis)
Y	Y axis of the XYZ rectangular coordinate system (Secondary Axis)
Z	Z axis of the XYZ rectangular coordinate system (Third Axis)
A	In the XYZ rectangular coordinate system, the rotation axis parallel to the X axis. (When it is set to the rotation axis in the parameters setting)
B	In the XYZ rectangular coordinate system, the rotation axis parallel to the Y axis. (When it is set to the rotation axis in the parameters setting)
C	In the XYZ rectangular coordinate system, the rotation axis parallel to the Z axis. (When it is set to the rotation axis in the parameters setting)
U	1st additional linear axis (rotation axis, when it is set to the rotation axis in the parameters setting)

Reserved character	Meaning
V	2nd additional linear axis (rotation axis, when it is set to the rotation axis in the parameters setting)
W	3rd additional linear axis (rotation axis, when it is set to the rotation axis in the parameters setting)
S	Used to specify the speed of revolution or to make the position command for the spindle axis
G	Preparatory Functions
F	Feed rate
M	Miscellaneous Functions
S	Specify the rotation speed
I	Rotational center coordinate value for X-axis circular interpolation
J	Rotational center coordinate value for Y-axis circular interpolation
K	Rotational center coordinate value for Z-axis circular interpolation
N	Statement Number (Sequence Number)
O	Used for the same purpose as N
P	Used to represent the optional data of Preparatory Functions and Miscellaneous Functions.
Q	Secondary option data representation. Drill's cutting depth and so on.
R	Circle radius
T	Tool functions
L	Repetition number
H	Tool length offset number
Other character	
IF	Conditional branch operation
GOTO	Branch jump
WHILE, DO	Loop iteration declaration
END	End loop
%, ;, ( )	Comments
/	Optional Block Skip
LE,GE,EQ,LT,GT,NE	Compare instructions
SIN, COS, TAN, ATAN, SQRT, ABS, ROUND, AND, OR, FIX, FUP	Mathematical function

9.1.3 Coordinate System

The coordinate system means the space to be used as a basis for operating the machine. The motion controller uses the right-handed rectangular coordinate system and supports four modes: machine coordinate system, work coordinate system, local coordinate system, and relative coordinate system.



[Basic coordinate system]

(1) machine coordinate system

Each machine used for motion control has its own specific position setting, and the coordinate system is set based on this specific position. This particular position is the "machine origin" of the machine, and the coordinate system based on this machine origin is the "machine coordinate system". The "machine origin" and its accompanying "machine coordinate system" differ depending on the machine to which the motion controller is applied. Accordingly, please refer to the instruction manual of the applicable machine.

Generally, when power is applied and the machine is started, homing is performed first. After homing, the reset machine position is reset to "0" position and at this time, the machine coordinate system is changed to the origin position. However, in the case of machinery equipped with the absolute encoder-positioning feedback, the absolute position is maintained independent of homing.

For more details on setting the machine coordinate system, refer to the G53 command description.

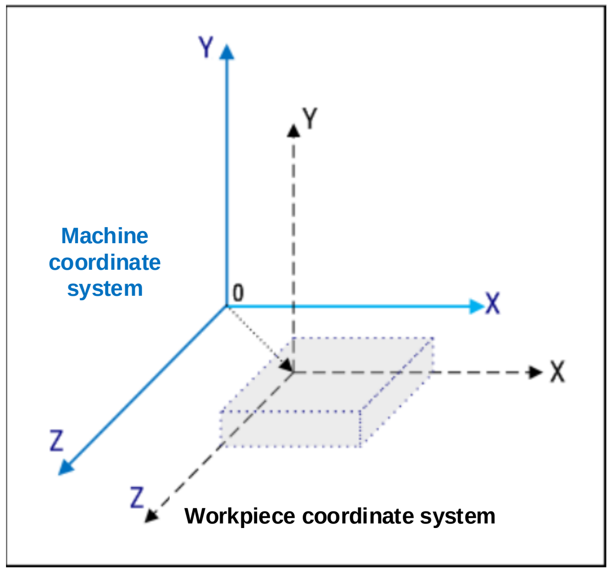
(2) Workpiece coordinate system

The workpiece coordinate system means the coordinate system whose origin is the machining reference point of the product.

Generally, the origin of the workpiece coordinate system is set by the workpiece coordinate system setting command. When the workpiece coordinate system is set, since then, the command operates at the new coordinate whose origin is the machining start point of the product. For the setting (G92) and selection (G54 ~ G59) of the workpiece coordinate system, please refer to the description section of the commands.

(3) Local coordinate system

It is called the local coordinate system to set the reference point at any position on the workpiece coordinate system and make the command when programing with the workpiece coordinate system. It refers to the coordinate system created newly within the program of the workpiece coordinate.



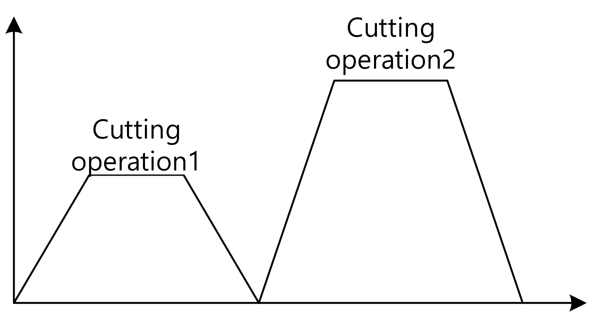
[Each coordinate system and offset]

9.1.4 How to Accelerate/Decelerate Interpolation Operation

The NC control function can set how to accelerate and decelerate operation during cutting feed. A method of acceleration and deceleration has an acceleration/deceleration before interpolation mode and an acceleration/deceleration after interpolation mode. According to each mode, the form of acceleration and deceleration zones of an axis in cutting feed varies.

(1) Acceleration and deceleration before interpolation

The acceleration/deceleration before interpolation mode performs operation by calculating the form of acceleration and deceleration before the cutting feed operation is performed. When the acceleration/deceleration before interpolation mode performs the cutting feed operation, the position of each axis stops at a command position when the cutting feed ends because the operation is performed by previously calculating the acceleration and deceleration profiles.



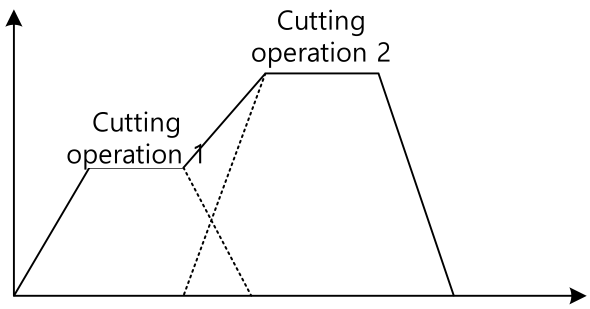
[Speed profiles]

The speed profile of the cutting feed operation can be changed by setting acceleration/ cutting feed deceleration/cutting feed jerk values in the cutting feed setting of the NC channel parameter.

Cutting Feed Settings	Upper speed limit of the cutting feed	10000 mm/m
	Lower speed limit of the cutting feed	1000 mm/m
	Acc./Dec. method of the interpolation	1: Acc./Dec. before Intp.
	Corner Speed Limit Mode	0: Unused
	Cutting Acceleration/Deceleration Type	0: Linear
	Blocks opr. for Acc./Dec. before Intp.	1: Buffered
	Cutting feed acceleration (Before-Intp.)	200 mm/s2
	Cutting feed deceleration (Before-Intp.)	200 mm/s2
	Cutting feed jerk (Before-Intp.)	0 mm/s3
	Allowable Angle (Angle mode)	0 deg
	Allowable Speed Difference(Speed Difference Mode)	0 mm/m
	Deceleration Speed Setting	0 mm/m
	The speed multiplier with respect to the Linear Axis of the Rotary Shaft (Speed Difference Mode)	0
	Number of correction for cutting feed deceleration (decrease after interpolation)	0 msec

(2) Acceleration and deceleration after interpolation

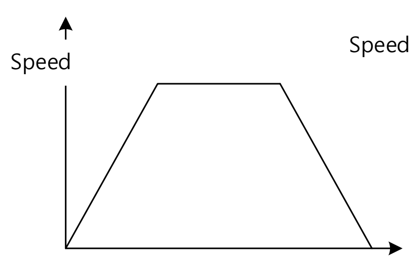
The acceleration/deceleration after interpolation mode performs the operation by calculating the form of acceleration and deceleration while the cutting feed operation is performed. When the acceleration/deceleration after interpolation mode performs the cutting feed operation, the position of each axis cannot reach a command position when the cutting feed ends because acceleration and deceleration profiles are calculated during operation. But the acceleration/deceleration after interpolation mode performs smoother operation in the operation of consecutive cutting feed commands than the acceleration/deceleration before interpolation mode.

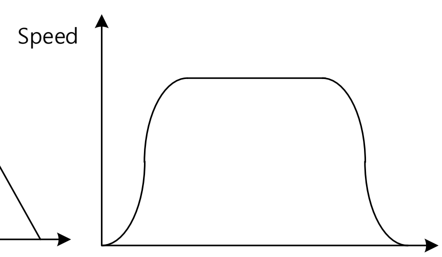


[Speed profiles]

1) The form of acceleration and deceleration in the acceleration/deceleration after interpolation mode
The speed profile of acceleration and deceleration zones can be set by setting the form of acceleration and deceleration of the cutting feed in the cutting feed setting of the NC channel parameter. 0: Line-type 1: S-type, Two types can be set.

Cutting Feed Settings	Upper speed limit of the cutting feed	10000 mm/m
	Lower speed limit of the cutting feed	1000 mm/m
	Acc./Dec. method of the interpolation	1: Acc./Dec. before Intp.
	Corner Speed Limit Mode	0: Unused
	Cutting Acceleration/Deceleration Type	0: Linear
	Blocks opr. for Acc./Dec. before Intp.	1: Buffered
	Cutting feed acceleration (Before-Intp.)	200 mm/s2
	Cutting feed deceleration (Before-Intp.)	200 mm/s2
	Cutting feed jerk (Before-Intp.)	0 mm/s3
	Allowable Angle (Angle mode)	0 deg
	Allowable Speed Difference(Speed Difference Mode)	0 mm/m
	Deceleration Speed Setting	0 mm/m
	The speed multiplier with respect to the Linear Axis of the Rotary Shaft (Speed Difference Mode)	0
	Number of correction for cutting feed deceleration (decrease after interpolation)	0 msec





2) The time constant of acceleration and deceleration after interpolation

Enter the time of acceleration and deceleration needed to reach the specified speed value after starting the cutting feed. The time constant of acceleration and deceleration after interpolation sets the time constant of cutting feed acceleration and deceleration in msec in the setting of the cutting feed of the NC channel parameter.

Cutting Feed Settings	Upper speed limit of the cutting feed	10000 mm/m
	Lower speed limit of the cutting feed	1000 mm/m
	Acc./Dec. method of the interpolation	1: Acc./Dec. before Intp.
	Corner Speed Limit Mode	0: Unused
	Cutting Acceleration/Deceleration Type	0: Linear
	Blocks opr. for Acc./Dec. before Intp.	1: Buffered
	Cutting feed acceleration (Before-Intp.)	200 mm/s2
	Cutting feed deceleration (Before-Intp.)	200 mm/s2
	Cutting feed jerk (Before-Intp.)	0 mm/s3
	Allowable Angle (Angle mode)	0 deg
	Allowable Speed Difference(Speed Difference Mode)	0 mm/m
	Deceleration Speed Setting	0 mm/m
	The speed multiplier with respect to the Linear Axis of the Rotary Shaft (Speed Difference Mode)	0
	Number of correction for cutting feed deceleration (decrease after interpolation)	0 msec

3) Corner speed limit function

The function is to improve the problem that curvature occurs in a corner when setting the time constant value bigger if using acceleration and deceleration after interpolation. It provides two modes such as an angle mode and a speed differential mode. Corner speed limit function in the cutting feed setting of the NC channel parameter 0: Disable 1: Angle mode 2: Speed differential mode can be set.

Cutting Feed Settings	Upper speed limit of the cutting feed	10000 mm/m
	Lower speed limit of the cutting feed	1000 mm/m
	Acc./Dec. method of the interpolation	1: Acc./Dec. before Intp.
	Corner Speed Limit Mode	0: Unused
	Cutting Acceleration/Deceleration Type	0: Linear
	Blocks opr. for Acc./Dec. before Intp.	1: Buffered
	Cutting feed acceleration (Before-Intp.)	200 mm/s2
	Cutting feed deceleration (Before-Intp.)	200 mm/s2
	Cutting feed jerk (Before-Intp.)	0 mm/s3
	Allowable Angle (Angle mode)	0 deg
	Allowable Speed Difference(Speed Difference Mode)	0 mm/m
	Deceleration Speed Setting	0 mm/m
	The speed multiplier with respect to the Linear Axis of the Rotary Shaft (Speed Difference Mode)	0
	Number of correction for cutting feed deceleration (decrease after interpolation)	0 msec

4) Angle mode

If the angle of the specified block is less than the allowed angle, the next block starts to traverse after the speed in a corner is decelerated up to the setting value of the deceleration speed. In the result, you can get a sharp corner according the setting value of the deceleration speed.

Cutting Feed Settings	Upper speed limit of the cutting feed	10000 mm/m
	Lower speed limit of the cutting feed	1000 mm/m
	Acc./Dec. method of the interpolation	1: Acc./Dec. before Intp.
	Corner Speed Limit Mode	0: Unused
	Cutting Acceleration/Deceleration Type	0: Linear
	Blocks opr. for Acc./Dec. before Intp.	1: Buffered
	Cutting feed acceleration (Before-Intp.)	200 mm/s2
	Cutting feed deceleration (Before-Intp.)	200 mm/s2
	Cutting feed jerk (Before-Intp.)	0 mm/s3
	Allowable Angle (Angle mode)	0 deg
	Allowable Speed Difference(Speed Difference Mode)	0 mm/m
	Deceleration Speed Setting	0 mm/m
	The speed multiplier with respect to the Linear Axis of the Rotary Shaft (Speed Difference Mode)	0
	Number of correction for cutting feed deceleration (decrease after	0 msec

5) Speed differential mode

If the speed difference between the end point of the specified existing block and the starting point of a new block is greater than the setting value of the allowed speed difference, the next block starts to traverse after being decelerated up to the setting value of the deceleration speed. You can get a sharp corner according the setting value of the deceleration speed.

Cutting Feed Settings	Upper speed limit of the cutting feed	10000 mm/m
	Lower speed limit of the cutting feed	1000 mm/m
	Acc./Dec. method of the interpolation	1: Acc./Dec. before Intp.
	Corner Speed Limit Mode	0: Unused
	Cutting Acceleration/Deceleration Type	0: Linear
	Blocks opr. for Acc./Dec. before Intp.	1: Buffered
	Cutting feed acceleration (Before-Intp.)	200 mm/s2
	Cutting feed deceleration (Before-Intp.)	200 mm/s2
	Cutting feed jerk (Before-Intp.)	0 mm/s3
	Allowable Angle (Angle mode)	0 deg
	Allowable Speed Difference(Speed Difference Mode)	0 mm/m
	Deceleration Speed Setting	0 mm/m
	The speed multiplier with respect to the Linear Axis of the Rotary Shaft (Speed Difference Mode)	0
	Number of correction for cutting feed deceleration (decrease after	0 msec

9.2 Program Configuration

9.2.1 NC Program

The NC program is a file consisting of the commands with control information about the axis. NC program can be added to 'Motion data - NC program' in XG5000. The NC program is used in the form of ‘program name.extension’ when saving the NC program as a file, and the file name extension of the NC program used is ‘nc’. The NC program is divided into two types, ''main program'' and ''sub program'', depending on the nature of the file.

The name of the ''main program'' can be assigned by a user with any name. For example, such as ''main.nc'' or ''main control .nc'', any name can be applied in English or Korean. The ''subprogram'' can be named using the four-digit numbers. (Eg: 1234) The ''subprogram'' can be disabled as needed, and the nested call is allowed up to 9 levels.

(1) Main program

The main program is the program that controls the whole flow of the motion program. The name of the main program can be written in any name and in the case of the nested call, there are 10 levels based on the main.

(2) Sub program

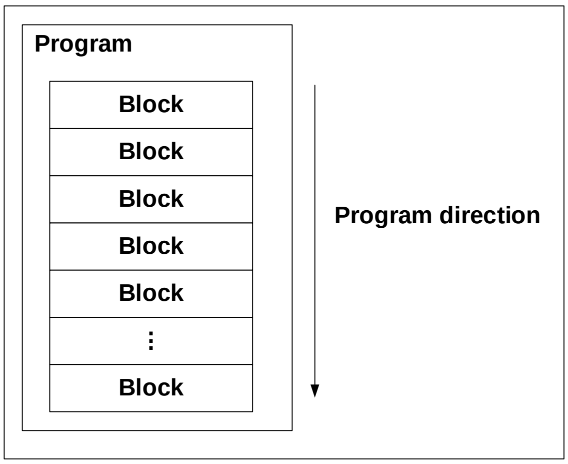
The sub program is executed by the main program’s call, and the only 4-digit numbers can be applied to the program file name. (0000 to 9999) The extension is the ''nc'' which is the same as the main program. The subprogram can be called directly from the main program so it must be written in numeric names only and distinguished by them.

9.2.2 NC Program Configuration

(1) Basic configuration of the program

The NC program is created with various instructions (G / M codes and instructions) that can be recognized by the motion controller and it consists of a set of blocks with the information for each operation command.

The NC program is written in the ST language. If the character that is not specified by the NC command (G / M codes and instructions) is used, an error occurs. The program starts with the first block one by one.



(2) Configuration of Blocks

It consists of the basic NC commands and command information for driving the machine.

One block corresponds to one line of the program. The maximum number of the characters that can be used in a block is 300, including the space characters. A maximum of 10 NC commands can be used per one block. When the number of characters available in one block or the limit of the NC command is exceeded, an error occurs.

N~~	G~~	X~.~	Y~.~	Z~.~	F~.~ (M/S/T …)
Statement Number	Preparation command (G/M codes and commands)	Command information (Command information of the coordinate)			Auxiliary command (G/M codes and commands)

The blocks of the NC program are normally input as shown in the table above. Unless there is a particularly limited notation in the same block, the order of each motion command does not matter. However, it is executed in the order of command, and for the Modal command, it continues to run until another NC command is made.

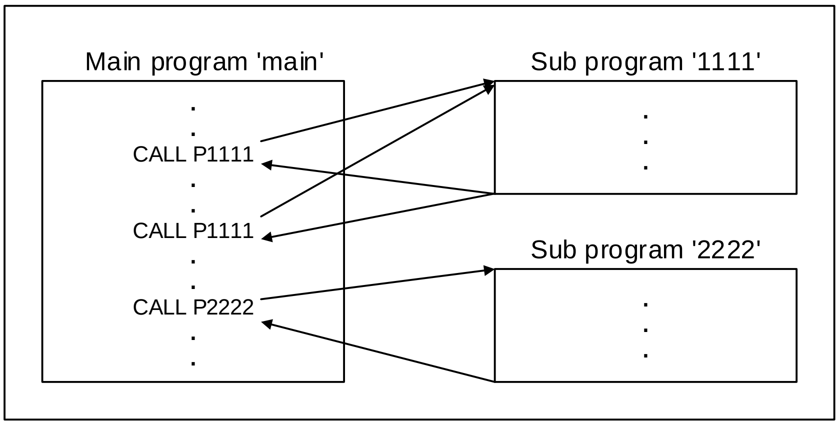
At the head of each block, there is a Statement Number which contains the motion control's sequence information. The Statement Number is often referred in the GOTO and CALL instructions, and does not have to be used unless it is necessary. The Statement Number can also be written alone in a block.

The command information depends on the preparatory commands (G / M codes and instructions) used earlier. Although you do not use the command information, no error will occur. However, even if you need the command information, the preparatory commands that do not use the command information will be ignored.

The auxiliary commands (G / M codes and instructions) are used with the preparatory commands to further refine the operating information of the preparatory commands.

(3) The main program and the subprogram

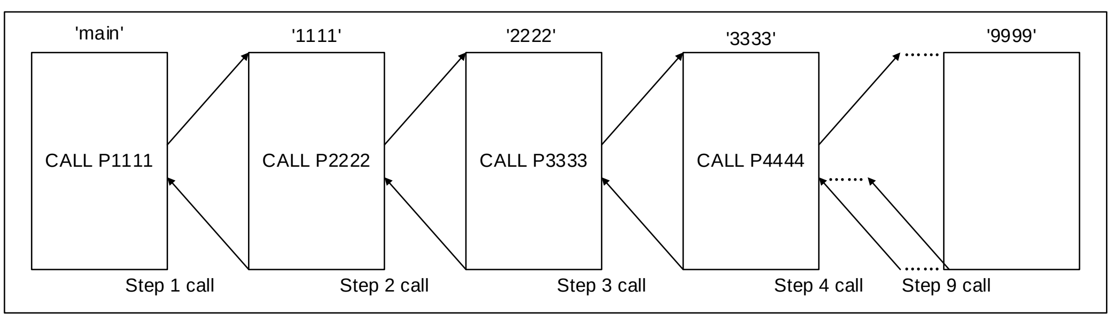
Basically, you can configure and control the NC program with the main program only. However, if a series of identical commands or control intervals are repeated, you can make such parts into the subprogram and call it in case of necessity. When the subprogram is called, the details of the subprogram will run thereafter. When all commands of the subprogram have been executed, the details of the main program will run again.



(4) Call multiplicity of the subprogram

The main program can call a subprogram, and the subprogram can call another subprogram. If calling the subprogram for the first time is the Call Step 1, the NC program of the motion controller can be called up to 9 steps as shown below. If the subprogram call multiplicity goes beyond step 9, an error occurs.

The motion controller's NC program supports the multiple calls up to 9 steps but it does not allow "recursive calls of the subprogram" that call the subprogram itself or "recursive calls" that may lead to call deadlock between subprograms. If the subprogram is called in this way, an error will occur.



(5) Call and return of the subprogram

The subprogram is called by M98. The procedures for call and return of the subprogram are as follows.

```
M98    P_ Q_ R_ L_
M99    P_
```

M98: Call of the subprogram

M99: End of the subprogram

P _: For M98, the name of the subprogram (_ is a 4-digit number.)
For M99, the statement Number of the block to return

Q _: Statement Number of the start block of the subprogram (if omitted, it starts from the first block).

R _: Statement Number of the end block of the subprogram (if omitted, it proceeds until M99.)

L _: Repeat count of the subprogram calls

In addition to M98, you can call the reserved sub program using the macro program calling M code set in NC channel parameter. This method can be used only in the main program, and when used in the subprogram, it operates with the general M code.

If there is no program to call, or if the syntax or argument is wrong, an error occurs. For alarms occurred, please refer to Appendix 2'Error Information & Solutions'.

(6) Repeat of the main program

When M99 is commanded in the main program, the details of the main program can be executed through the repeat mode. If P_ is not commanded, it is repetitively executed from the first block of the main program. If L_ is not commanded, it runs repeatedly and infinitely.

```
M99    P_ L_
```

M99: Repeat of the main program

P _: Statement Number of the repeated start block

L _: Repeat count of calls

9.2.3 Data

(1) Data type used in the NC program

The NC program uses the numerical data for each axis command, feed rate command, DWELL command, macro variables, etc. When each operator is applied, the constant is used directly in the program.

At this time, the range of data type that can be used for the NC program of motion controller is as follows. If the wrong data range is applied, an alarm will occur. For the location of the alarm information, etc., please refer to Appendix 2, Errors Information & Solutions'.

Available range of the integer -2,147,483,648 ~ 2,147,483,647 - Range of expressible real numbers Can be expressed up to 12 digits including a decimal point and a sign.

(2) Unit

For the numbers used for the NC program, the system of units applied changes depending on whether or not to enter a decimal point ("."). This depends on how the decimal point check item is set among the NC channel parameters of the motion controller.

Group	Name	Channel 1
Basic Settings	Target Machining Quantity	0
	Target machining quantity at M99 repetition	0
	Check of decimal point	1: Unused
	Keep workpiece coordinate system	0: Keep
	Macro call on T-code command	0: Do not call
	DWELL Method	0: Time
	Block selection at NC reset	0: Keep the Current Block
	Statement number search	0: Search
	Minimum command unit	0 mm
	Whether to use G22 [No traveling area]	0: Used
	Inner/Outer side of G22 [No traveling area]	0: Inner side
	Whether to use the 3rd [No traveling area]	0: Used
	Rotary Axis of Cylindrical Interpolation	0: X Axis
	Linear axis for interpolating the polar coordi	0: None
	Rotary axis for interpolating the polar coordi	0: None
	Monitoring time for in-position completion	5000 ms

If the parameter is set to "0: Check", the value changes internally, as shown in the following example.

X100 → 100 um = 100 / 1000.0 = 0.1mm X100, → 100.0 mm = 100 / 1.0 = 100.0 mm ex) X10.4 → 10.4 mm = 10.4 / 1.0 = 10.4 mm X104 → 104 um = 104 / 1000.0 = 0.104 mm

The unit conversion specified above is effective only when it is entered directly into the axis's coordinate information.

If the decimal point check parameter is set to "0: Check", the system of units used for the G04 (TIME) command is as follows.

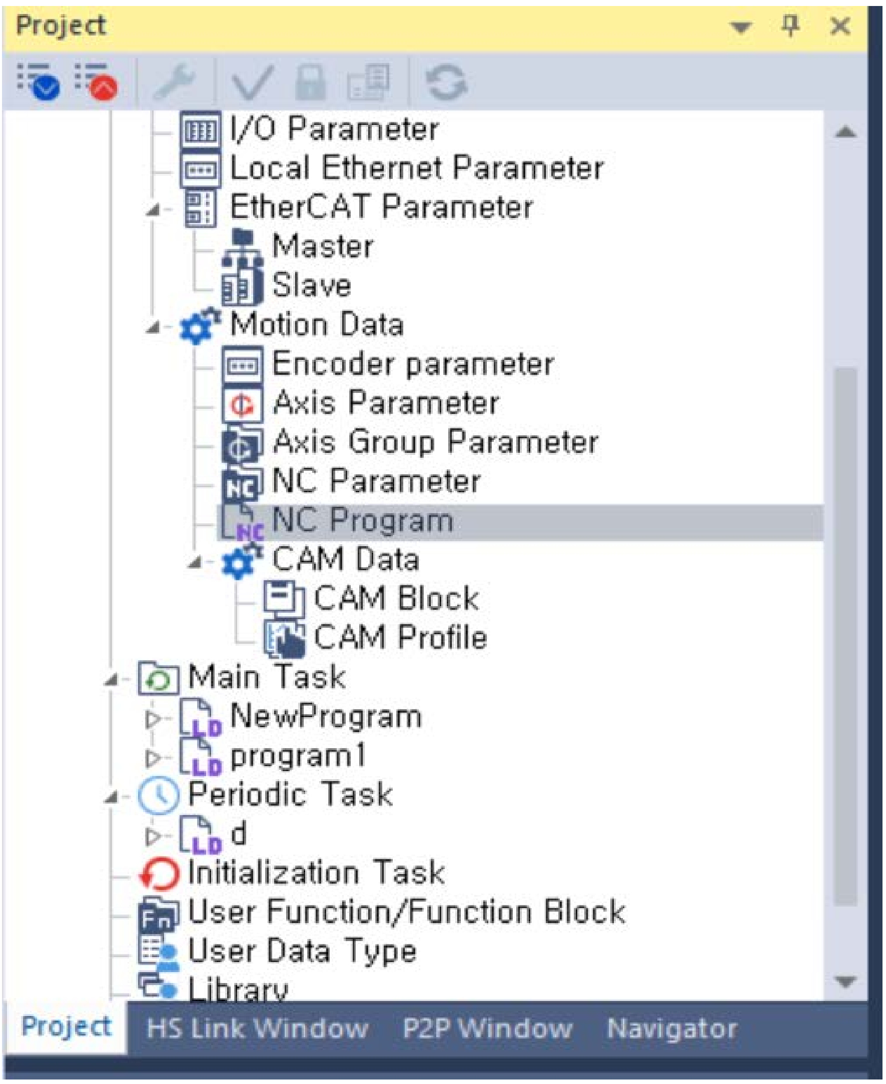
G04 X1 → 1 msec = 1 / 1000.0 = 0.001 sec G04 X1.0 → 1.0 sec = 1 / 1.0 = 1 sec

9.2.4 NC program Write

Programs added to 'Motion Data – NC Program' in XG5000 can basically be downloaded when the motion controller's operation mode is in the stop state. However, the NC program can be downloaded even when the operation mode is in the run state, and there are some restrictions in this case.

(1) NC program download during RUN.

This is a function to download the NC program created by the user while the motion controller's operation mode is run. In this case, only 'NC program' should be selected in the project write window as follows.



(2) Restrictions

If the user-written NC program is downloaded while the motion controller's operation mode is RUN, there are the following restrictions depending on the operating conditions.

1) When the NC channel is in automatic operation

- After the NC operation currently in automatic operation is completed, it is changed to the downloaded NC program. That is, even if the program currently in automatic operation is changed, the changed program does not operate immediately. The point in time when the downloaded NC program is applied during run is always after the automatic operation of the NC channel is finished.

- If the NC program currently being operated is repeated infinitely with the M99 command in the last block of the program, automatic operation cannot be ended in the normal way,automatic operation is stopped with the NC_Reset command or the motion controller is changed to 'stop' mode and automatic operation is stopped and then changed to the downloaded program.
- If the number of repetitions is specified in the NM99 command, automatic operation stops after repeating the number of repetitions and then the downloaded program is changed.

2) If the NC_LoadProgram command is executed during internal write processing after downloading the NC program
   - 0x3605 error occurs.
   - Error contents: the program designation command cannot be executed while writing the downloaded NC program during run.
3) If the NC_CycleStart command is executed during internal write processing after downloading the NC program
   - 0x350F error occurs.
   - Error contents: the automatic operation start command cannot be executed while writing the downloaded NC program during run.

Notes

NC program download function during run is supported from the next version onwards.

1. XG5000 - V4.29
2. XMC OS - V1.50

9.3 NC Commands

The NC command is basically described based on the three types of data: the type of motion to be moved, the target position and the target speed. The basic formats of the position command and speed command are as follows.

9.3.1 Basic Format of the NC Position Command

The motion controller supports two types of commands; the command method using "X, Y, Z, A, B, C, U, V, W, S", the command method using "I, J, K" type

(1) Absolute/relative position specified (X, Y, Z, A, B, C, U, V, W, S)

```
X_ Y_ Z_ A_ B_ C_ U_ V_ W_ S_
```

X_ Y_ Z_ ~ S_: Command the position of the axis

The position command of the axis, "X_ Y_ Z_ ... "is mainly used to specify the position on the coordinate of the point where each axis should be finally moved at the time of interpolation or traverse command.

The position of the coordinate you want to move should be specified behind "X, Y, Z, … S". The number of axes specified with the G code is limited to 10. The motion controller can control up to 32 axes by motion control but the maximum number of axes by the NC program is 10 axes and more axes can be controlled by the axis command of the motion control program.
In the case of the absolute command, specify the coordinate value (coordinate value of the end point of feed) of the feed target point. If it is the relative command, specify the increment value from the current position to the feed target point.

When you specify the position of the coordinate you want to move, the operating mode of the command differs depending on whether to use the value that includes "." or not ".". Please refer to the position formula in "9.2.3 System of units of data (2) "

```
G90
G00 X100. Y100, Z100, U100,            % Rapid Traverse, X-axis 100, Y-axis 100, Z-axis 100, U-axis 100
G01 X150. Y200
U300,
Z325,
M02
```

(2) Specify the center point of the arc (I, J, K)

```
I_ J_ K_
```

I_ J_ K_: Central point-position command of an arc for circular interpolation

The position command of axis, "I_ J_ K_" is used to command the position of each origin point on the coordinate to the individual axis when commanding the central point of the arc for circular interpolation.

You can specify the location of the central point of the arc behind "I_ J_ K_".
"I_ J_ K_" must specify the increment value from the current position to the origin.

When you specify the position of the coordinate, the operating mode of the command depends on whether to use the value that includes "." or not ".". Please refer to the position formula in "9.2.3 System of units of data (2) "

```
G90
G02 X100. Y100, I50, J50,      % clockwise circular interpolation, X-axis 100, Y-axis 100, Central point(X50, Y50)
M02
```

The above program shows the traverse target point, X-axis 100, Y-axis 100 and the coordinate of the origin direct the clockwise circular interpolation with X-axis 50, Y-axis 50.

(3) Speed command (F)
The speed command has the function to instruct the speed of the interpolation command.
If there is no separate speed command, it operates at the basic speed set at the lower limit of the cutting feed among NC channel parameters.

Group	Name	Channel 1
Cutting Feed Settings	Upper speed limit of the cutting feed	10000 mm/m
	Lower speed limit of the cutting feed	1000 mm/m
	Acc./Dec. method of the interpolation	1: Acc./Dec. before Intp.
	Blocks opr. for Acc./Dec. before Intp.	1: Buffered
	Cutting feed acceleration (Before-Intp.)	200 mm/s2
	Cutting feed deceleration (Before-Intp.)	200 mm/s2
	Cutting feed jerk (Before-Intp.)	0 mm/s3

The speed command is valid for the interpolation command only and has no effect on the rapid traverse command.

```
F_
```

: F Speed command

The speed command specifies the operating speed of the interpolation command.
The speed command can be instructed with each interpolation command or instructed independently.
Since the speed command is the modal command, once it is instructed, it is valid for the operations of the interpolation command until another speed command is made.
In the case of the speed command for linear interpolation, the operating speed is calculated in the same manner as the speed formula of "9.2.3 System of units of data (2)".
In the case of circular interpolation command, the speed command is calculated as linear velocity in tangential direction.

```
G90
F5000
G02 X100. Y100, I50, J50,        % Clockwise circular interpolation speed: 5000
G01 X400. Y250, Z300, F3500      % Linear interpolation,speed: 3500
M02
```

9.3.2 NC Command List

The NC commands (G / M code and other commands) used for the motion controller are as follows.

Classification	Program command	Function
G code command	G00	Rapid positioning control
	G01	Linear interpolation feed control
	G02	Clockwise circular / helical interpolation
	G03	Counter clockwise circular / helical interpolation
	G04	Dwell function
	G09	Exact Stop
	G10	Data set
	G15	Cancel polar coordinates
	G16	Command polar coordinates
	G17	Select the circular interpolation plane (XY plane)
	G18	Select the circular interpolation plane (ZX plane)
	G19	Select the circular interpolation plane (YZ plane)
	G20	Inch input
	G21	meter input
	G22	Stroke check function ON
	G23	Stroke check function OFF
	G27	Homing check
	G28	automatic homing
	G29	Return at the auto-origin
	G30	Automatic 2nd, 3rd and 4th homing
	G31	Skip function 1
	G31.1	Skip function 1
	G31.2	Skip function 2
	G31.3	Skip function 3
	G31.4	Skip function 4
	G37	Automatic tool length measurement 1
	G37.1	Automatic tool length measurement 1
	G37.2	Automatic tool length measurement 2
	G37.3	Automatic tool length measurement 3
	G37.4	Automatic tool length measurement 4
	G39	Tool radius compensation corner circular interpolation
	G40	Cancel compensation of tool diameter
	G41	Compensate the tool diameter to the left
	G42	Compensate the tool diameter to the right
	G43	+ Compensate the tool length in the direction of +
	G44	- Compensate the tool length in the direction of +
	G45	Tool position offset increase
	G46	Tool position offset decrease
	G47	Tool position offset double increase
	G48	Tool position offset double decrease
	G49	Cancel compensation of the tool length
	G50	Scaling/Mirror image cancel

Classification	Program command	Function
	G51	Scaling/Mirror image setting
	G52	Set the local coordinate system
	G53	Select the machine coordinate system
	G54	Select the workpiece coordinate system 1
	G55	Select the workpiece coordinate system 2
	G56	Select the workpiece coordinate system 3
	G57	Select the workpiece coordinate system 4
	G58	Select the workpiece coordinate system 5
	G59	Select the workpiece coordinate system 6
	G60	Single direction positioning
	G61	Exact Stop Mode
	G62	Automatic corner override mode
	G63	Tapping mode
	G64	Cutting mode (acceleration and deceleration after interpolation)
	G65	Macro call
	G66	Macro modal call
	G67	Macro modal call cancel
	G68	Coordinate rotation
	G69	Coordinate rotation cancel
	G74	Counter tapping cycle
	G80	Fixed cycle cancellation
	G81	Drill cycle/Spot drill cycle
	G82	Drilling dwell cycle/Counter boring cycle
	G84	Tapping cycle
	G90	Absolute command
	G91	Incremental command
	G92	Set the workpiece coordinate system, the max. speed of the master axis
	G94	Feed mode command per minute
	G95	Feed mode command per revolution
	G96	Constant surface speed control
	G97	Constant surface speed control release
	G98	Return to initial point at a canned cycle
	G99	Return to R-point at a canned cycle
	G107	Cylindrical interpolation mode setting
	G112	Interpolation mode of the polar coordinate ON
	G113	Interpolation mode of the polar coordinate OFF

Classification	Program command	Function
M code	M00	Program's operation is stopped.
	M01	Selective stop
	M02	PROGRAM END
	M03	Forward rotation of the master axis
	M04	Reverse rotation of the master axis
	M05	Master axis stop
	M06	Tool change
	M08	Coolant ON
	M09	Coolant OFF
	M30	Program End
	M98	Auxiliary program call
	M99	End of the auxiliary program
Position command	X, Y, Z, A, B, C, U, V, W, S	Specify the location of the axis
	I, J, K	Rotating central point coordinate of each axis for circular interpolation
Speed command	F	Feed rate command
Dwell time	X	Specified Dwell time
Other instruction	N	Specify the Statement Number
	P	Specify the call number of the subprogram
	IF	Conditional branch instruction and conditional operation
	GOTO	Branch instruction
	WHILE, DO	Execute a certain program repetitively
	END	Loop end
	%, ;	Comment processing command
	LE, GE, EQ, LT, GT, NE	Compare instructions
	AND, OR, XOR, +, -, *, /	Operation instruction
	=	Assignment operator
	SIN, COS, TAN, ATAN, SQRT, ABS, ROUND, AND, OR, FIX, FUP	Mathematical operation function

9.3.3 Description of the NC Command

(1) G code

The G code defines the types of the commands such as feed and machining method of each axis during machining, and it is the command to carry out mechanical drive and operation of the NC program, etc.
There are two types of G code as shown below.

Classification	Meaning
One-shot G code instruction	G code instruction that is valid only in the block to which the G code is commanded
Modal G code command	G code instruction that is valid until it is released by another command from the block to which the G code is commanded

Modal command

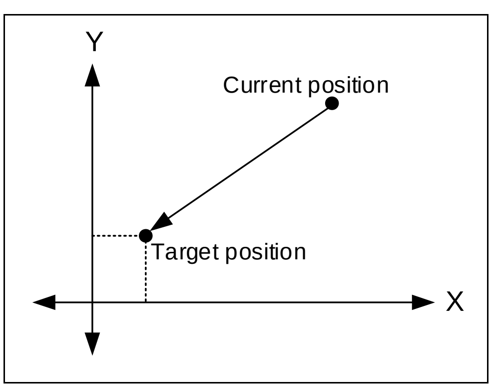
```
G01 X10. F100
Y100,
Z300,
G00, X100
```

The G01 is a modal command as shown in the above program so the G01 command will be executed until the G00 command is made since the G01 is commanded even if G01 command is not separately specified.

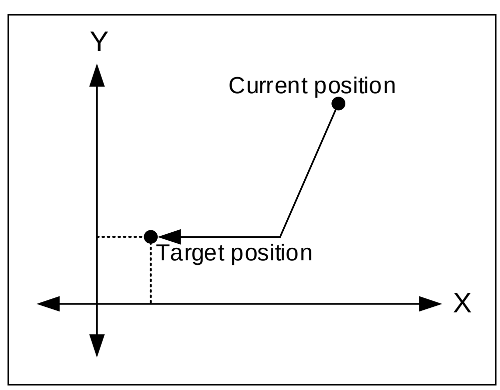
1) Rapid traverse (G00)

```
(G90, G91) G00 X_ Y_ Z_ A_ B_ C_ U_ V_ W_ S_
```

G90, G91: Absolute/Incremental command
G00: Rapid positioning control command
X_ Y_ Z_ A_ B_ C_ U_ V_ W_ S: feed target position



[Current location and moving target]



[Type of transport section]

As shown in the left figure above, the Rapid Traverse (G00) transfers the specified axis quickly from the X, Y point of the coordinate given by the command information or the current position to the position incremented by the command information. Under the G00 command, the feed rate moves according to the G00 feed rate set for each axis.
G00 is traversed independently for each axis. Since the axis with short travel distance first reaches the target point, the shape of the travel section is not a straight line as above.

The Rapid Traverse command is a modal command so once it is instructed, it is valid for the axis traversing command until another traverse command is made.

```
G90
G00 X100 Y100
Z100
```

The above program is the example set to absolute command using G90. After that, it rapidly traverses the X and Y axis to (100,100) points and then, traverses the Z axis to 100 points again.

The parameters related to the Rapid Traverse are as follows.

NC parameter	Group	Parameter name
NC channel/axis parameter	Rapid feed	Rapid feed acceleration
		Rapid feed deceleration
		Rapid feed jerk
		Rapid feed speed

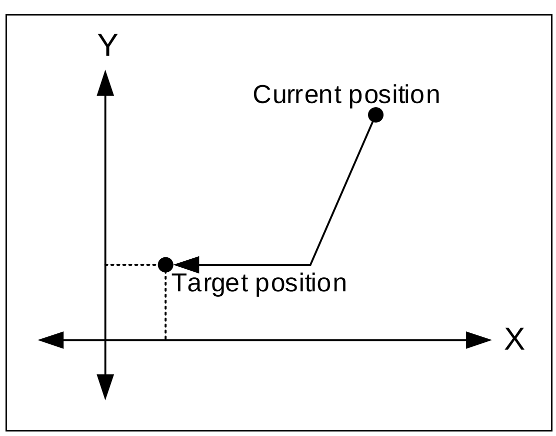
NC Parameter	Group and parameter name
NC channel/axis Parameter	(see table below)

Group	Name	X Axis
Rapid Traverse Settings	Rapid traverse acceleration	500 mm/s2
	Rapid traverse deceleration	500 mm/s2
	Rapid traverse jerk	0 mm/s3
	Rapid traverse speed	10000 mm/m

2) Linear Interpolation (G01)

```
(G90, G91) G01 X_ Y_ Z_ A_ B_ C_ U_ V_ W_ S_ (F_)
```

G90, G91: Absolute/Incremental command
G01: Interpolation feed control command
X_ Y_ Z_ A_ B_ C_ U_ V_ W_ S: feed target position
F_: Feed rate



The linear interpolation (G01) is the function that simultaneously traverses each axis in a straight line to the commanded position at the speed set by the F command in order to perform the desired machining (eg. cutting) as shown in the figure above.

In the case of the incremental command (G91), it moves to the position incremented by the command information in a straight line from the current axis position.

Since the G01 command is a modal command, once it is instructed, it continues to be valid for the axis feed command until another feed command is made. The feed speed command can be instructed with the "F" code. There are two feed methods; feed per revolution and feed per minute. Normally, the feed per minute is applied.

Feed rate of the X axis: Dx Feed rate of the Y axis: Dy Feed rate of the Z axis: Dz D= $\sqrt{(Dx^2+Dy^2+Dz^2)}$ X axis's feed rate Fx=Dx/D ×F Y axis's feed rate Fy=Dy/D ×F Z axis's feed rate Fz=Dz/D ×F

The feed rate of each axis differs depending on the distance of each axis as shown in the above formula.

```
G90
G01 X50 Y35 F3000        % Interpolation feed control, target position to traverse(X=50, Y=35), speed 3000
G91
X100 Y55                 % Interpolation feed control
```

The above program shows the example that executes the interpolation feed control at the speed 3000 to the points of the X axis 50 and the Y axis 35 under the absolute command and then, executes the interpolation feed control to the points incremented by 100 from the X axis and 55 from the Y axis under the incremental command.
"X100. Y55.", as mentioned above, the G01 and F code are modal commands so they operate under interpolation feed control without a separate command.

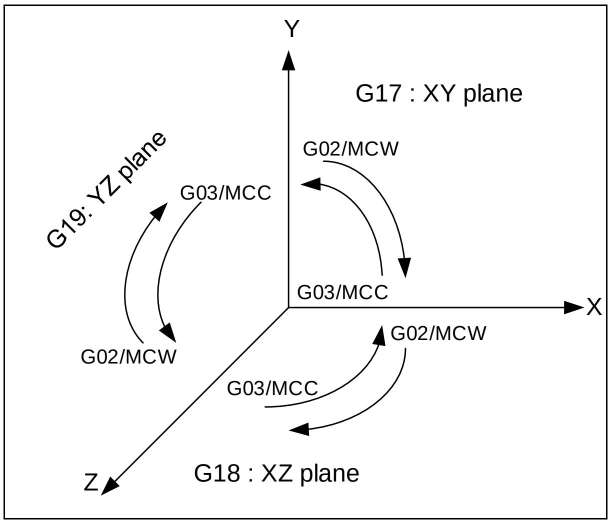
3) Circular arc interpolation (G02/G03)

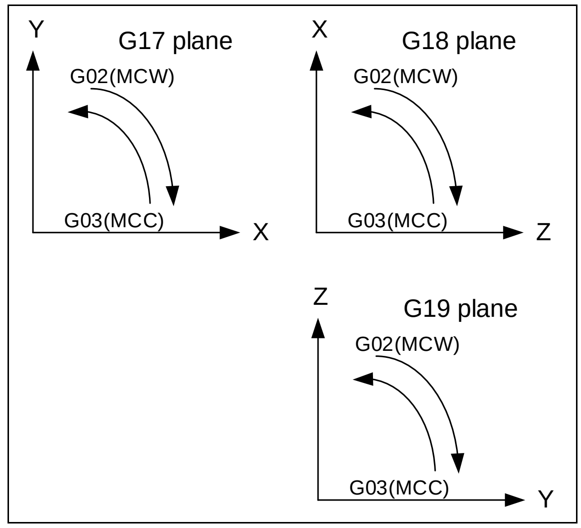
```
(G90, G91) G17 (G02, G03) X_ Y_ (I_ J_, R_) (F_)
(G90, G91) G18 (G02, G03) X_ Z_ (I_ K_, R_) (F_)
(G90, G91) G19 (G02, G03) Y_ Z_ (J_ K_, R_) (F_)
```

G90, G91: Absolute/Incremental command
G17, G18, G19: Specify the plane to execute the circular interpolation
G02, G03: Clockwise, counter clockwise circular interpolation
X_ Y_ Z_: Target position to traverse
I_ J_ K_/R_: Reference point or radius of an arc
F_: Feed rate

The circular interpolation (G02 / G03) is the command to execute the rotary feed at the speed specified by "F" based on the commanded or calculated central point to the target position to traverse. In the circular interpolation command, the speed means the linear velocity in tangential direction.

For the circular interpolation command, you should select the plant to execute circular interpolation before the command. The NC program has the command for specifying each plane; G17 is defined as XY plane, G18 as ZX plane, and G19 as YZ plane as shown below. If you enter the command information beyond the selected plane, an error will occur.





The circular interpolation can be executed by setting the central point of an arc as the command information or by setting the circular radius as the command information. In order to execute the circular interpolation, one of the above two methods must applied for commanding.

When applying the Reference Point Method using I, J, and K and the Radius Method using the "R" code simultaneously between the circular interpolation commands, I, J, and K commands will be ignored and then, the circular interpolation will be executed under the radius command using “R”.

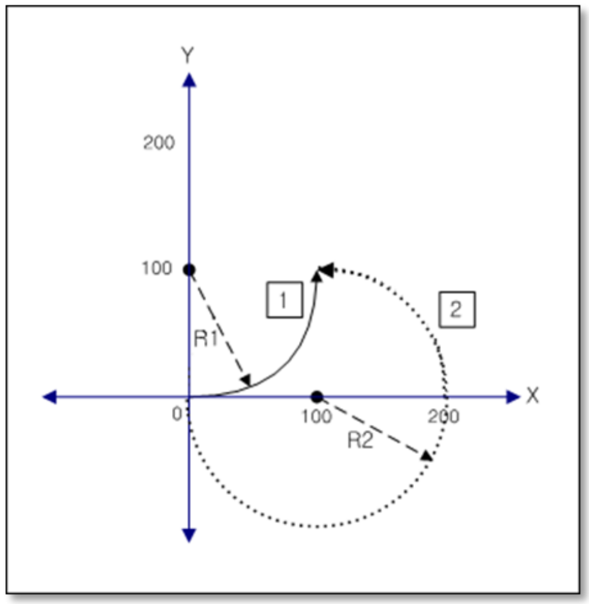
When instructing the circular interpolation with specifying the central point, the central point command information (I, J, K) of the circular interpolation command is always the position incremented from the start position to the central point regardless of the absolute / incremental command (G90 / G91). The central point command information, I, J, and K correspond to X, Y, and Z, respectively. If the value of I, J, K command information is "0", it can be omitted.

When the current position and the target position to traverse are the same, the circular interpolation with specifying the central point can command a 360-degree perfect circle.

The circular interpolation with specifying the R (radius) does not designate the central point to determine the arc section, but only the R(radius) which forms the arc from the current position to the target position to traverse.

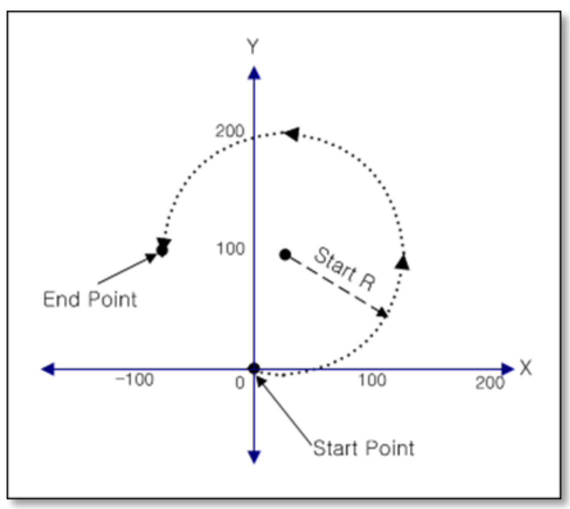
When executing the circular interpolation using the R (radius) designation method, the central point of the arc can have two shapes. At this time, the motion controller performs the circular interpolation by selecting the central point with the shortest arc to the target point to traverse as shown below.

Unlike the circular interpolation with specifying the central point, the circular interpolation with the R (radius) designation cannot command a 360-degree perfect circle.



[Circular interpolation with specifying the radius(R)]

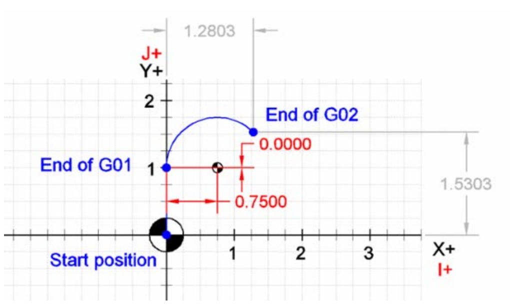
During the circular interpolation, if the start and end radii of the arc are different, an alarm occurs. If it is within the error radius, it traverses to the original trajectory and then reaches the final position with a straight line.



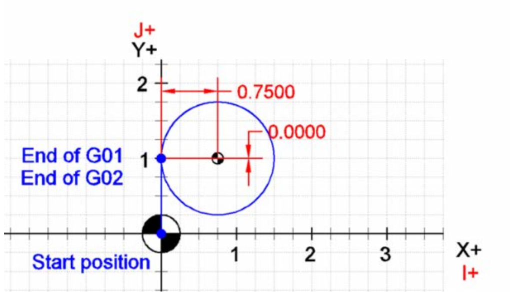
[Circular interpolation of sections with different radius of rotation]

G90
G00 X0 Y0 Z0
% XY plane
G90
G17
G02 X50 Y50 I50 F100 % Clockwise circular interpolation, Central point(X=X+50, Y=0), speed 100
G03 X0 Y0 R50 % Counter clockwise circular interpolation, R(Radius)=50
G91 % relative coordinate
G03 X100 Y100 J100 % Counter clockwise circular interpolation, Central point(X=0, Y=Y+100)
G02 X-100 Y-100 R100 % Clockwise circular interpolation, R(Radius)=100

Example 1 - Circular interpolation clockwise
G01 Y1.0 F8.0; % Linear interpolation (Y=1.0), Speed 8
G02 X1.2803 Y1.5303 I0.75 % Circular interpolation clockwise, R (radius) = 0.75, Target position (X=1.2803, Y1.5303)



Example 2 - Circle clockwise
G01 Y1.0 F8.0; % Linear interpolation (Y=1.0), Speed 8
G02 I0.75 % Circular interpolation clockwise, R (radius) = 0.75



% ZX plane
G90
G18
G02 Z50 X50 K50 F200 % Clockwise circular interpolation, Central point(X=X+0, Z=Z+50), speed 200
G03 Z0 X0 R50 % Counter clockwise circular interpolation, R(Radius)=50

```
G91
G03 Z100   X100.   I100          % Counter clockwise circular interpolation, Central point(X=X+100, Z=Z+0)
G02 Z-100 X-100 R100             % Clockwise circular interpolation, R(Radius)=100,
                                 % Target position to traverse(X=X-100, Z=Z-100)

% YZ plane
G90
G19
G02 Y50 Z50 J50 F300             % Clockwise circular interpolation, Central point(Y=Y+50, Z=Z+0), speed 300
G03 Y0   Z0   R50          % Counter clockwise circular interpolation, R(Radius)=50
G91
G03 Y100   Z100   K100           % Counter clockwise circular interpolation, Central point(Y=Y+0, Z=Z+100)
G02 Y-100 Z-100 R100             % Clockwise circular interpolation, R(Radius)=100

% Perfect circle
G17
G02 I50   % Clockwise circular interpolation(360-degree perfect circle), Central point(X=X+50, Y=Y+0)
G03 J50        % Counter clockwise circular interpolation(360-degree perfect circle), Central point(X=X+50, Y=Y+0)
G02 I50 J50    % Clockwise circular interpolation(360-degree perfect circle), Central point(X=X+50, Y=Y+50)

% I ignore R apply
G02 X-100 I30 R50
```

The parameters related to the circular interpolation are as follows.

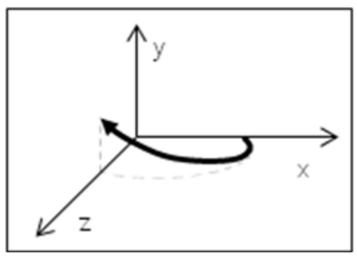
NC Parameter	Group and parameter name		
	Group	Name	Channel 1
NC Channel Parameter	Circular Machining Settings	Operation when the circular alarm occurs	0: Generate an alarm
		Speed limitation for circular interpolation	0: Unused
		Tolerance of arc radius	0 mm
		Circular radius with speed limitation	0 mm
		Upper speed limit of circular interpolation	1000 mm/m
		Lower speed limit of circular interpolation	100 mm/m
		Circular interpolation acceleration	100 mm/s2
		Circular interpolation deceleration	100 mm/s2
		Circular interpolation jerk	0 mm/s3

4) Helical interpolation(G02/G03)

```
(G90, G91) G17 (G02, G03) X_ Y_ (I_ J_, R_) Z_ F_
(G90, G91) G18 (G02, G03) X_ Z_ (I_ K_, R_) Y_ F_
(G90, G91) G19 (G02, G03) Y_ Z_ (J_ K_, R_) X_ F
```

G90, G91: Absolute/Incremental command
G17, G18, G19: Specify the plane to execute the circular interpolation
G02, G03: Clockwise, counter clockwise circular interpolation
X_ Y_ Z_: Target position to traverse
I_ J_ K_/R_: Reference point or radius of an arc
F_: Feed rate

The helical interpolation instructs another axis whose plane is not specified in the circular interpolation command, and the axis is synchronized in a straight line and traverses with the progress of the circular interpolation. That is, when XY plane G17 is commanded, the Z axis can be transferred.




```
G90
G00 X0 Y0 Z0

% XY plane
G90
G17
G02 X50 Y50 I50 Z10 F100 % Clockwise circular interpolation, Central point(X=X+50, Y=0), Z position 10, speed 100
G03 X0    Y0    Z 20 R50          % counter clockwise circular interpolation, R(Radius)=50, Z position 20
```

5) Dwell function

```
G04 (X_, P_)
```

G04: dwell command
X_, P_: Dwell time command information(sec, msec)

The DWELL command (G04) is the command to stop for the time specified following "X" or "P" and then, execute the next block.
X's unit is sec, P's unit is in msec.

```
G90
G00 X0. Y0, Z0,
G01 X100. Y100, F1500
G04 P100                    % Dwell time : 100 msec
G00 X500. Y500,
G04 X1              % Dwell time : 1 sec
G91
G01 X100. Y100, F1500
```

The parameter related to the DWELL command is as follows.

NC Parameter	Group and parameter name
NC Channel Parameter	(see table below)

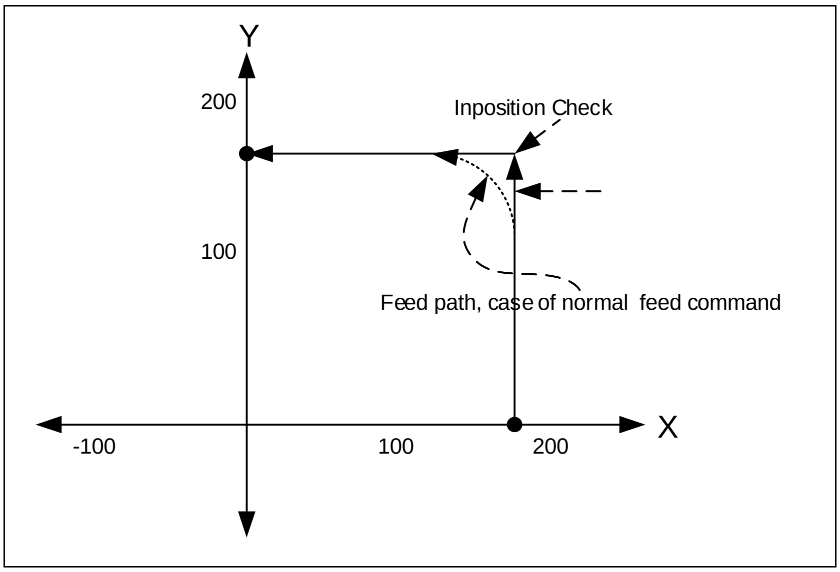
Group	Name	Channel 1
Basic Settings	Target Machining Quantity	0
	Target machining quantity at M99 repetition	0
	Check of decimal point	1: Unused
	Keep workpiece coordinate system	0: Keep
	Macro call on T-code command	0: Do not call
	DWELL Method	0: Time
	Block selection at NC reset	0: Keep the Current Block
	Statement number search	0: Search
	Minimum command unit	0 mm
	Whether to use G22 [No traveling area]	0: Used
	Inner/Outer side of G22 [No traveling area]	0: Inner side
	Whether to use the 3rd [No traveling area]	0: Used
	Rotary Axis of Cylindrical Interpolation	0: X Axis
	Linear axis for interpolating the polar coordi	0: None
	Rotary axis for interpolating the polar coordi	0: None
	Monitoring time for in-position completion	5000 ms

6) Exact Stop(G09)

G09

Exact Stop (Precision stop command)

In normal feed / cutting operations, the corner section decelerates the current block and accelerates the next block because it is affected by physical inertia when accelerating or decelerating the axis traverse. That is why 'Rounding' occurs. This function performs the 'Inposition Check' and proceeds to the next block as shown below to put the commanded block in the instructed position of feed / cutting exactly.



This function is a one-shot command so it is valid in the corresponding command only.
If the G09 command is used for the simple feed command like "G01", the 'Inposition Check' is performed at the target position to traverse.
If machining such as cutting is performed using this function, fine stopping phenomenon occurs at the connecting intersection point of the curved surfaces, resulting in some disadvantages; bad condition of the machined surface, significant wear of the tool, and long machining time.

```
G90
G00 X0. Y0, Z0
G09 G01 X100. Y100, F5000          % Linear feeding through the Exact Stop
X200, Y250,                          % linear feeding
G10
```

The above program is the example of using the Exact Stop (G09) command for linear feeding. The G09 command in the above program is a one-shot command so "X200. Y250." command is not affected by the G09 command.

7) data setting (G10)

(G90, G91) G10 (L10) (P_ R_)
(G90, G91) G10 (L12) (P_ R_)

G10: data setting
L10: H Data setting(tool diameter)
L12: D Data setting(tool length)
P_: offset number
R_: tool compensation amount
G90, G91: The compensation value is set and the value previously set is updated.

This command is used in the diameter of a tool, the compensation value of tool length and reference position offset of the coordinate system, and can be input by program. It can be used if the real-time compensation value changes.

G10 G90 L10 P10 R5	%Set 5 to H10, compensation No. 5.
G10 G91 L12 P5 R10	%Set 10 to D5, compensation No. 10.

The parameter items related to data setting commands are as follows:

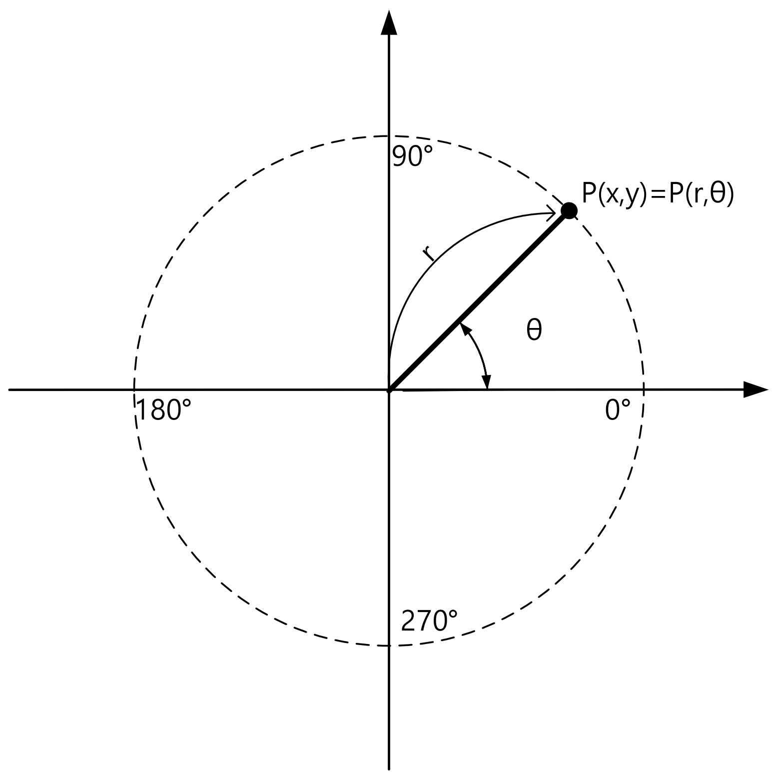
NC Parameter	Group and parameter name		
NC Channel Parameter		How to apply the compensation value	0: Apply the diameter value
		Compensation type of the tool diameter	0: Bypass Traverse
		Check the tool interference	0: Do not check
		Tool diameter compensation amount 1	0 mm
	Tool Diameter Compensation	Tool diameter compensation amount 61	0 mm
		Tool diameter compensation amount 62	0 mm
		Tool diameter compensation amount 63	0 mm
		Tool diameter compensation amount 64	0 mm
		Tool diameter compensation amount 65	0 mm
		Tool diameter compensation amount 127	0 mm
		Tool diameter compensation amount 128	0 mm
		Tool length compensation amount 1	0 mm
		Tool length compensation amount 2	0 mm
		Tool length compensation amount 24	0 mm
		Tool length compensation amount 25	0 mm
		Tool length compensation amount 26	0 mm
		Tool length compensation amount 27	0 mm
		Tool length compensation amount 28	0 mm
		Tool length compensation amount 127	0 mm
		Tool length compensation amount 128	0 mm

8) Command polar coordinates(G15, G16)

```
G15
G16 X_ Y_
```

G15: Cancel polar coordinates
G16 : Command polar coordinates
X : Polar coordinates arc radius
Y : Polar coordinates angle command

A polar coordinate command is used if the current coordinate system works with the polar coordinate system, not the right-angle coordinate system. You can work by entering pivot, angle and radius.



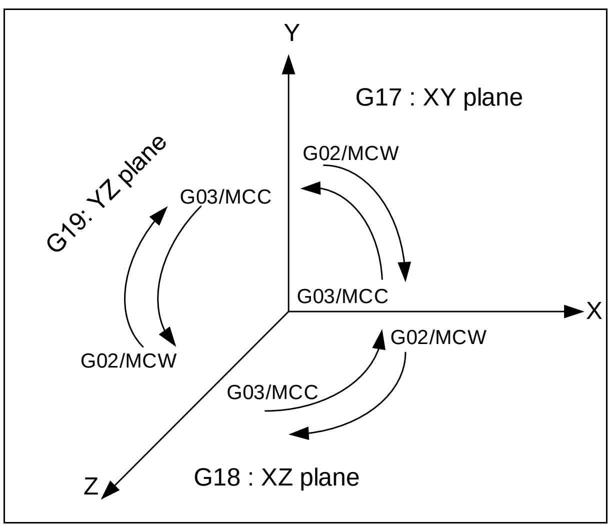
If the polar coordinate system is used, the circular interpolation plane appears differently. For this part, see the circular interpolation plane (G17). The reference position is set the same with the reference position of the right-angle coordinate system. If it is an incremental command, the current position becomes the reference position of the polar coordinate.

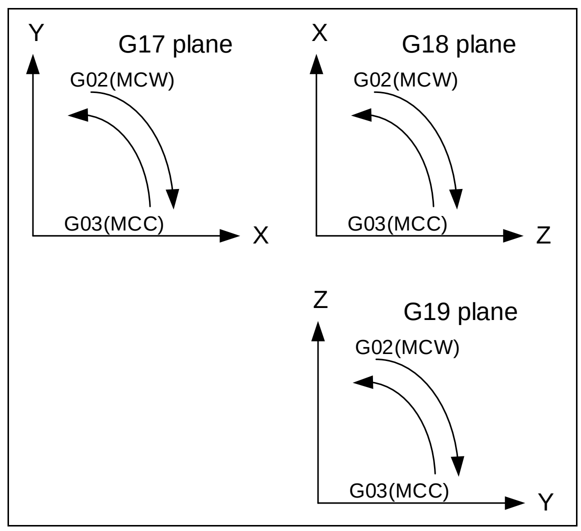
9) 9) Selecting the plane for circular interpolation (G17, G18, G19)

(G90, G91) G17 (G02, G03) X_ Y_ (I_ J_ / R_) F_
(G90, G91) G18 (G02, G03) X_ Z_ (I_ K_ / R_) F_
(G90, G91) G19 (G02, G03) Y_ Z_ (J_ K_ / R_) F

G90, G91: Absolute/Incremental command
G17: X-Y plane
G18: Z-X plane
G19: Y-Z plane
G02, G03: Clockwise, counter clockwise circular interpolation
X_ Y_ Z_: Target position to traverse
I_ J_ K_/R_: Reference point or radius of an arc
F_: Feed rate

This command specifies two planes to perform the circular interpolation.





The parameters related to the command to select planes for circular interpolation are as follows.

NC Parameter	Group and parameter name		
NC Channel Parameter	Default Settings	Default modal G-code for TRAVERSE	0: G00
		Default modal G-code for PLANE	0: G17
		Default modal G-code for ABS/INC	0: G90
		Default Modal Inch/Metric	0: G20
		Default modal G-code for Limit check	0: G22
		Default Modal Scale	0: G50

10) Inch/meter input (G20, G21)

```
G20
G21
```

G20: Inch input
G21: meter input

This command sets whether the position unit to be input is inch or meter. Even if this command is executed, the system of units including a position displayed on screen or an internal offset keeps the system of units set by parameters.
The command to shift this system of units should be used at the beginning of a program. When shifting to other system of units in the middle of a single program, unstable results can be caused. If the whole part of a plan is different from parameter units, use the command at the beginning of the program.

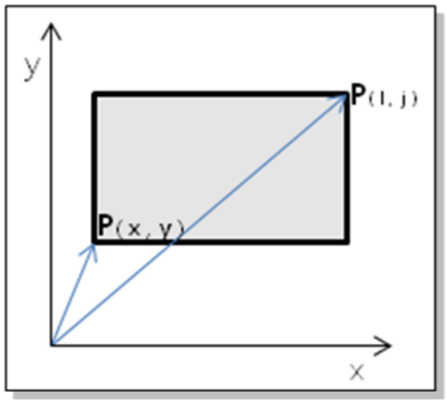
NC Parameter	Group and parameter name		
NC Channel Parameter	Default Settings	Default modal G-code for TRAVERSE	0: G00
		Default modal G-code for PLANE	0: G17
		Default modal G-code for ABS/INC	0: G90
		Default Modal Inch/Metric	0: G20
		Default modal G-code for Limit check	0: G22
		Default Modal Scale	0: G50

11) Stroke function enable/disable(G22, G23)

```
G22 (X_ Y_ Z_) (I_ J_ K_)
G23
```

G22: Stroke check function ON
G23: Stroke check function Off
X_ Y_ Z_: Enter the lower limit position based on the machine origin of each coordinate.
I_ J_ K_: Enter the upper limit position based on the machine origin of each coordinate.

This command sets the Soft Limit of the coordinate system. You can input the lower limit coordinate (X, Y, Z) and upper limit coordinate (I, J, K) of each stroke. If it is out of this range, the error will be displayed. Then, in a manual mode, it can be entered into the working area or be driven after turning off the stroke check function. This is a modal command so it continues to be valid once it is commanded.
For A, B, C, U, V, W, S axes other than X, Y and Z axes, it should be set by the parameters. At this time, the parameter, 'Whether to use G22 No Travelling Area' should be set to 0 to use the G22 command.
This command does not operate on a single axis, and works normally only when values are entered in two or more axes (plane or space).



The parameters related to Enable /Disable stroke are as follows.

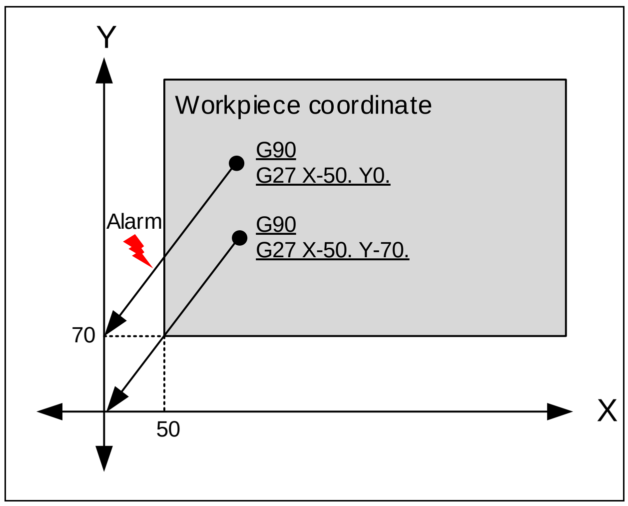
<table>
<tr><th>NC Parameter</th><th>Group and parameter name</th></tr>
<tr><td>NC Channel Parameter</td><td>
<table>
<tr><th>Group</th><th>Name</th><th>Channel 1</th></tr>
<tr><td rowspan="16">Basic Settings</td><td>Target Machining Quantity</td><td>0</td></tr>
<tr><td>Target machining quantity at M99 repetition</td><td>0</td></tr>
<tr><td>Check of decimal point</td><td>1: Unused</td></tr>
<tr><td>Keep workpiece coordinate system</td><td>0: Keep</td></tr>
<tr><td>Macro call on T-code command</td><td>0: Do not call</td></tr>
<tr><td>DWELL Method</td><td>0: Time</td></tr>
<tr><td>Block selection at NC reset</td><td>0: Keep the Current Block</td></tr>
<tr><td>Statement number search</td><td>0: Search</td></tr>
<tr><td>Minimum command unit</td><td>0 mm</td></tr>
<tr><td>Whether to use G22 [No traveling area]</td><td>0: Used</td></tr>
<tr><td>Inner/Outer side of G22 [No traveling area]</td><td>0: Inner side</td></tr>
<tr><td>Whether to use the 3rd [No traveling area]</td><td>0: Used</td></tr>
<tr><td>Rotary Axis of Cylindrical Interpolation</td><td>0: X Axis</td></tr>
<tr><td>Linear axis for interpolating the polar coordi</td><td>0: None</td></tr>
<tr><td>Rotary axis for interpolating the polar coordi</td><td>0: None</td></tr>
<tr><td>Monitoring time for in-position completion</td><td>5000 ms</td></tr>
</table>
</td></tr>
</table>

12) Homing check(G27)

(G90, G91) G27 X_ Y_ Z_ A_ B_ C_ U_ V_ W_ S_

G90, G91: Absolute/Incremental command
G27: homing check
X_ Y_ Z_ A_ B_ C_ U_ V_ W_ S_: Target coordinate

Through this command, it traverses to the specified X, Y, Z coordinate. When the current position is the origin after the traverse is done, homing is completed. If it is not the origin, the alarm occurs. When this command is instructed, the compensations of tool diameter and tool length are canceled.

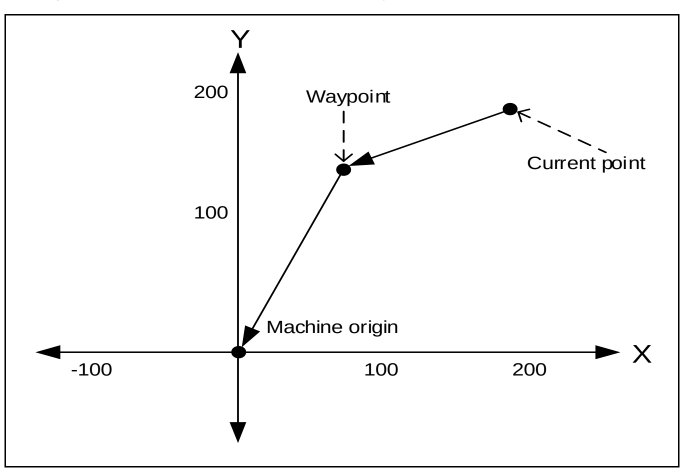


13) Automatic homing(G28)

(G90, G91) G28 X_ Y_ Z_ A_ B_ C_ U_ V_ W_ S_

G90, G91: Absolute/Incremental command
G28: Auto-homing command
X_ Y_ Z_ A_ B_ C_ U_ V_ W_ S_: Coordinate of waypoint of each axis to be homed

It is the command to automatically return the axis to the machine reference point. When the G28 command is encountered during the program execution, each axis is moved to the machine origin at the rapid traverse rate. At this time, it stops by the commanded position of each axis on the way.



The axes without receiving the auto-homing command do not move.
The incremental commands are available for axis positioning.
If the axis position command is "0", it returns directly to the machine origin without dropping by waypoints.

G90
G01 X100. Y100, Z100 F552. % Linear interpolation, target position to traverse(X=100, Y=100), speed 552
G28 X40. Y55. Z32. % Auto-homing, waypoint(X=40, Y=55, Z=32)
G91 G01 X50. Y50. F550.

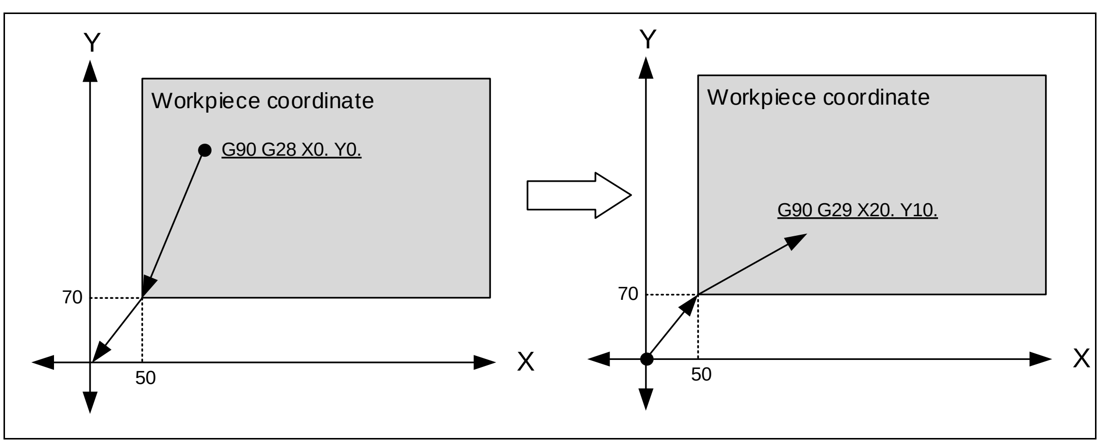
The above program is the example of moving the position of axes transferred to X, Y, Z axes linearly to the machine origin by using the G28 auto-homing command.

14) Automatic homing(G29)

(G90, G91) G29 X_ Y_ Z_ A_ B_ C_ U_ V_ W_ S_

G90, G91: Absolute/Incremental command
G29: Auto-homing command
X_ Y_ Z_ A_ B_ C_ U_ V_ W_ S_: return coordinate

This command is used when the traverse is done after the auto-homing(G28), 2nd, 3rd, 4th homing(G30) is instructed. It traverses rapidly (G00) to the returning coordinate via the waypoint that was used for homing. If the homing command has not been previously executed, the machine origin becomes the midpoint and it traverses to the returning coordinate. In this command, tool diameter compensation and tool length compensation are not applied. All the axes that have been commanded at the time of origin return have been traversed before the midpoint has been traversed, and only the coordinates that have been commanded at return coordinate are traversed thereafter.

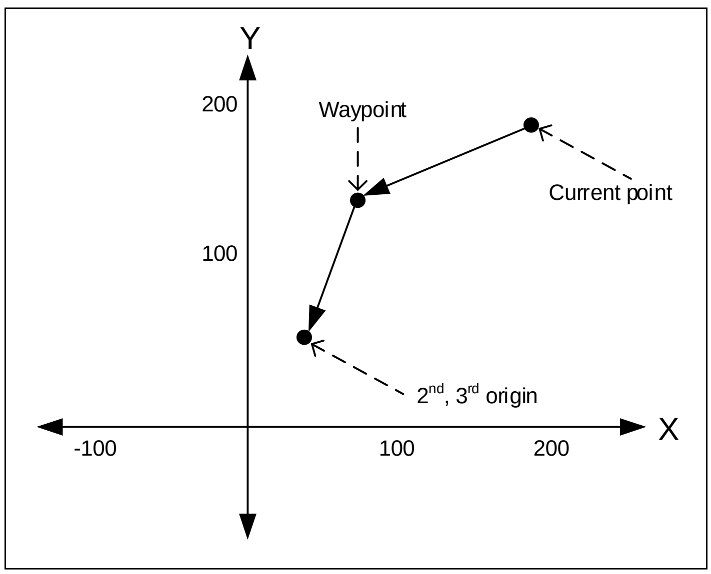


15) The 2nd, 3rd and 4th homing(G30)

(G90, G91) G30 (P2, P3, P4) X_ Y_ Z_ U_

G90, G91: Absolute/Incremental command
G30: Auto-homing command
P2: the 2th origin
P3: the 3rd origin
P4: the 4th origin
X_ Y_ Z_ A_ B_ C_ U_ V_ W_ S_: Coordinate of waypoint of each axis to be homed

This command automatically returns each commanded axis to the preconfigured 2nd, 3rd, 4th origin. If the G30 instruction is encountered during the program execution, each axis is moved to the specified 2nd or 3rd or 4th origin at the rapid traverse rate. At this time, it stops by the commanded position of each axis on the way.



The 2nd, 3rd, 4th origin coordinate of each axis should be specified separately in the NC channel / axis parameters.
Through the P2, P3, and P4 commands, for the 2nd and 3rd homing to be instructed currently, you can specify which origin is selected between the 2nd, 3rd origin. "P2" refers to the 2nd origin and "P3" refers to the 3rd origin.
The incremental commands are available for axis positioning.

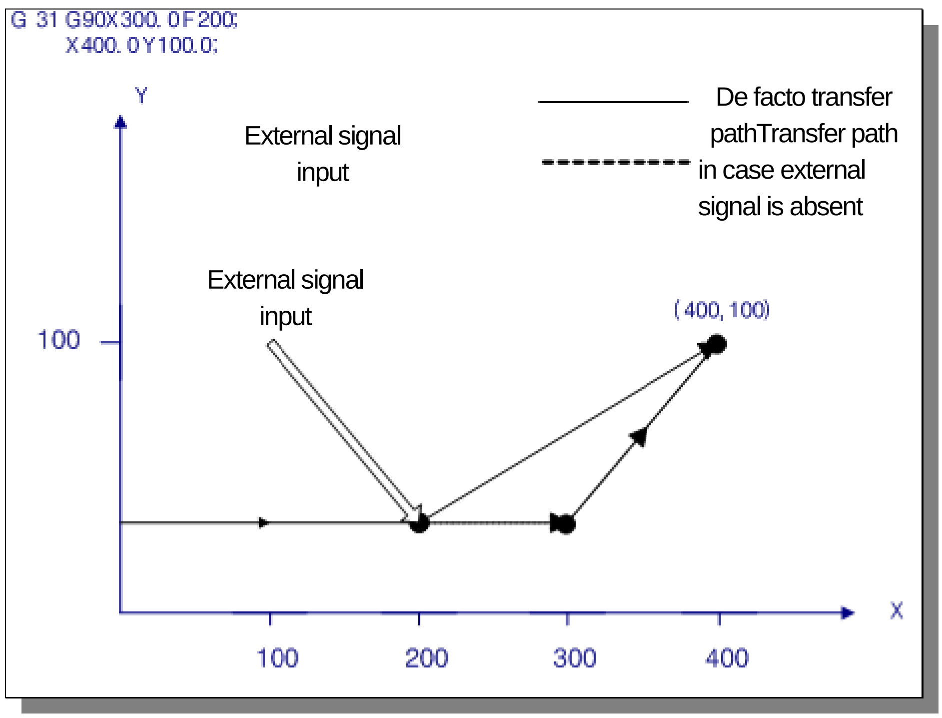
NC Parameter	Group and parameter name
NC Channel/Axis Parameter	<table><tr><th>Group</th><th>Name</th><th>X Axis</th></tr><tr><td rowspan="3">Home Settings</td><td>Position of 2nd home</td><td>0 mm</td></tr><tr><td>Position of 3rd home</td><td>0 mm</td></tr><tr><td>Position of 4rd home</td><td>0 mm</td></tr></table>

16) Skip function (G31 / G31.1 / G31.2 / G31.3 / G31.4)

{G31 / G31.1 / G31.2 / G31.3 / G31.4} X_ Y_ Z_ A_ B_ C_ U_ V_ W_ S_ F_

G31 / G31.1 / G31.2 / G31.3 / G31.4:
X_ Y_ Z_ A_ B_ C_ U_ V_ W_ S_: feed position
F_: Feed rate

The method of command and the type of axis feed are applied the same as those for linear interpolation (G01). Once the SKIP signal is input with the NC_BLOCKSKIP command, feed is stopped and it is progressed in the next block. This function is used for measuring a size of work-piece or for knowing a particular location during machining. At this time, the location of the machine at the point that it is stopped is saved in the flags by axis. For more information of the relevant flags, see “Appendix 1 Flag Alarm”.



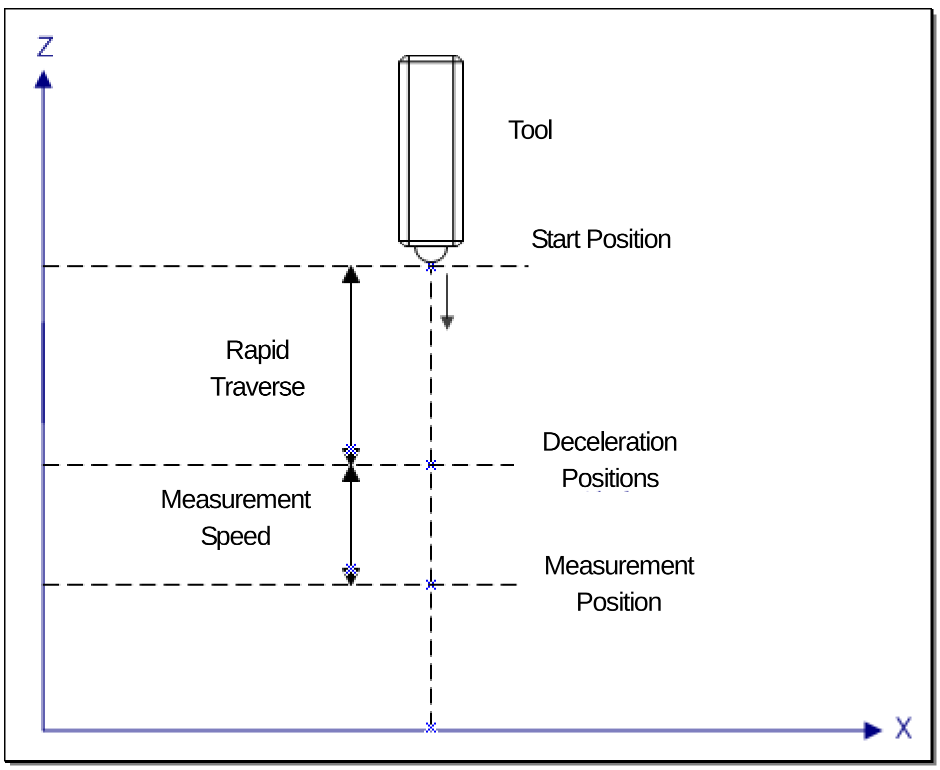
G90 G54 G00 X0. Y0. Z0.	
G31 X300. F200.;	- Input of external signal SKIP 1
G31.2 Y200. ;	- Input of external signal SKIP 2
G31.3 X0. ;	- Input of external signal SKIP 3
G31.4 Y0. ;	- Input of external signal SKIP 4
M02; program End	

17) Automatic tool length measurement (G37 / G37.1 / G37.2 / G37.3 / G37.4)

{G37 / G37.1 / G37.2 / G37.3 / G37.4} X_ Y_ Z_

G37: Auto tool length measurement 1 G code
G37.1: Auto tool length measurement 1 G code
G37.2: Auto tool length measurement 2 G code
G37.3: Auto tool length measurement 3 G code
G37.4: Auto tool length measurement 4 G code
X_ Y_ Z_ : measuring location

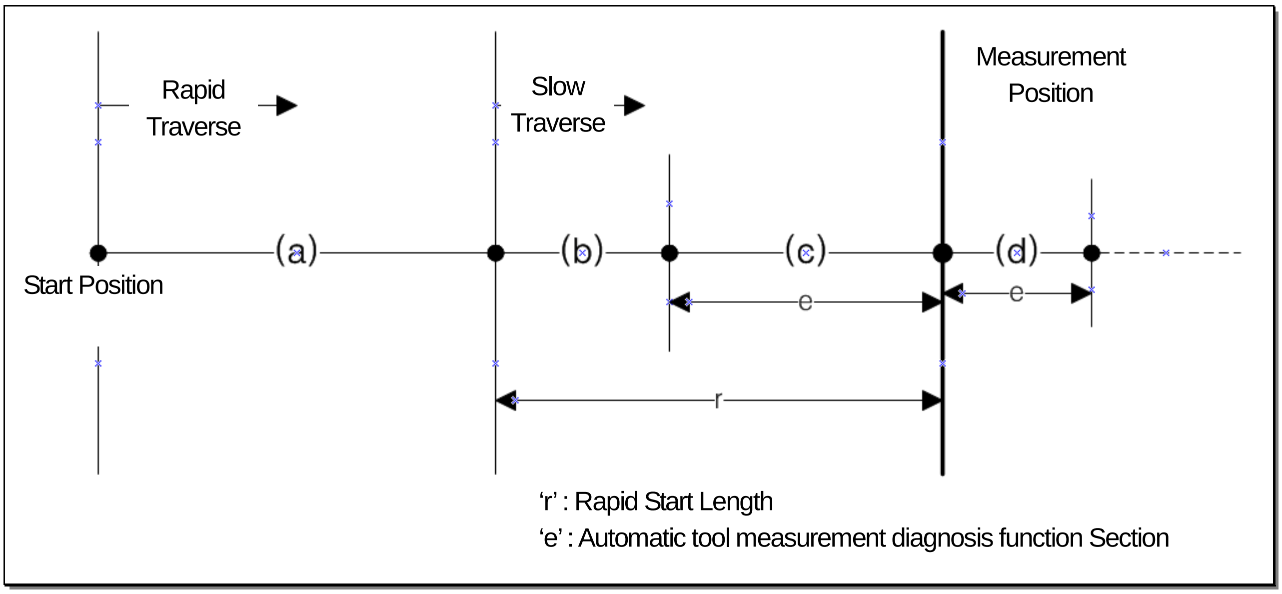
It is used when tool measuring equipment is installed on a particular position of the machine. It automatically measures the compensation value of a tool and compensates. A tool moves to a measuring location during which it slows down and moves until a signal to reach the measuring location displays from measuring equipment. It calculates a new tool compensation value (tool length compensation value) from the difference between the actually nominated position and the position where the auto tool measure signal happens.



(1) Auto tool measurement signal (SKIP signal)
If a SKIP signal is input with the NC_BLOCKSKIP command, the feed command for the relevant block is closed and tool measurement is carried out. G code corresponding to each SKIP signal of the NC_BLOCKSKIP command is shown as follows, and following G code, one of X, Y, or Z is nominated.

SKIP signal	G code
SKIP1	G37.1(G37)
SKIP2	G37.2
SKIP3	G37.3
SKIP4	G37.4

(2) Type of axis feed during auto tool measurement
Once the command of auto tool measurement is given, it rapidly moves a tool at first ((a) section). If a tool comes into the section of speed reduction set by a parameter, the tool is moved at a low speed set by the parameter ((B) section). If the tool reaches the measuring point and the measuring signal (SKIP signal of the NC_BLOCKSKIP command) is ON, tool movement is stopped.



When a measurement signal is detected in an (c),(d) area other than the section, a tool measurement alarm occurs.
If no measuring signal is detected until out of (d) range, a tool measurement alarm occurs.
Parameter items related to automatic tool length measurement are as follows.

NC Parameter	Group and parameter name		
NC Channel parameter	Automatic Tool Offset	+ Measurement Reference X of Automatic Tool Offset	0
		+ Measurement Reference Y of Automatic Tool Offset	0
		+ Measurement Reference Z of Automatic Tool Offset	0
		- Measurement Reference X of Automatic Tool Offset	0
		- Measurement Reference Y of Automatic Tool Offset	0
		- Measurement Reference Z of Automatic Tool Offset	0
		Automatic Tool Deceleration Start Length of Automatic Tool Offset	0
		Automatic Tool Measurement of Automatic Tool Offset Detectable Section	0
		Automatic Tool Deceleration Speed of Automatic Tool Offset	0

(3) Auto tool compensation amount calculation
A new compensation value is produced out by adding the difference between the coordinate values of a tool at the time of reaching the measuring location and the nominated measuring location to the currently used compensation value.

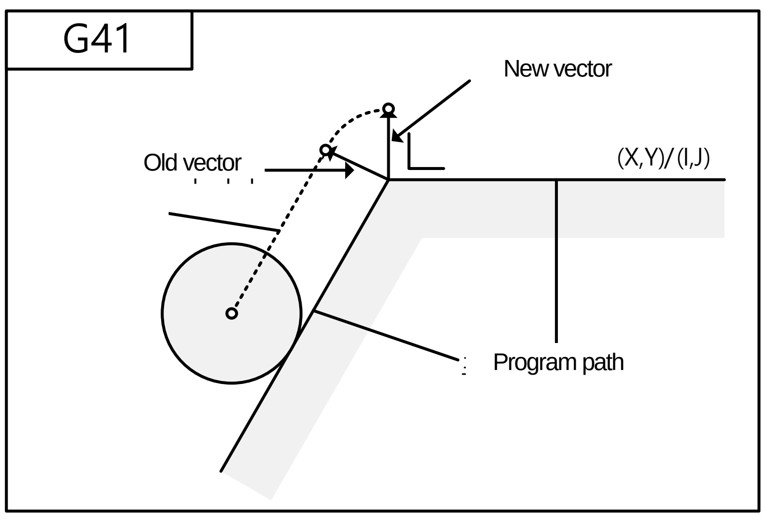
18) Compensation value = Current compensation value + (Location where measuring signal becomes ON – 19) Nominated measuring point)

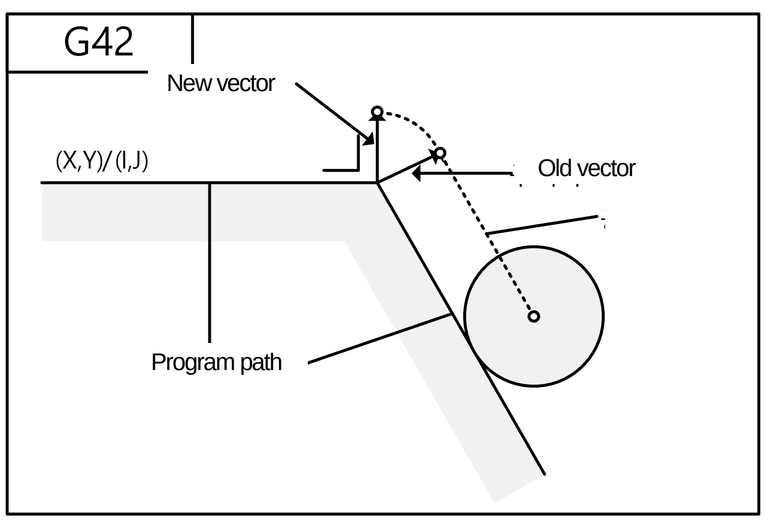
18) Corner circular arc interpolation (G39)

{G39} X_ Y_ / I_ J_

G39 : Handle corner offset with circular interpolation
X_ Y_ / I_ J_ : Command the vector for the next block

The corner off circular interpolation with the tool radius of the corner as the radius is carried out by the next command specified in the state of G01, G02 or G03. If specifying X and Y, the new vector that moves to the left (G41) or the right (G42) from a starting point of a right angle with X and Y is created. A tool moves according to a circular arc from the end of an old vector to the end of a new vector. X, Y and Z are displayed by the absolute value or the incremental value corresponding to G90 or G91. And I, J and K are always displayed by the incremental value at the end point.





The command of G39 can be given only during the offset mode (the state that G41 or G42 are commanded previously). Left and right rotation of a circular arc is decided by whether it is (G41) or (G42). This command carries out circular interpolation if there is the G function (G00/G01/G02/G03/G33).

19) Disable tool diameter compensation (G40)

```
{G40} [G00/G01] X_ Y_ Z_ A_ B_ C_ U_ V_ W_ S_
```

G40: Disable tool diameter compensation
X_ Y_ Z_ A_ B_ C_ U_ V_ W_ S_: Instruct the vector of the next command block

The G40 command is to cancel the tool diameter compensation. When G40 is commanded in the mode of G00 and G01, the mode will change from Enable Tool Diameter Compensation to Disable Tool Diameter Compensation. The offset in the G40 mode is always 0, and the center path of the tool matches the programmed path. The program should always be terminated in the G40 mode. If it ends in the G41 / G42 mode, the program will be terminated at a distance offset by the compensation amount. In addition, it is not possible to cancel the tool diameter compensation in the circular interpolation (G02, G03).

```
G40 X_ Y_
```

20) tool diameter compensation (G41, G42)

```
{G41/G42} [G00 / G01] X_ Y_ Z_ A_ B_ C_ U_ V_ W_ S_ D_
```

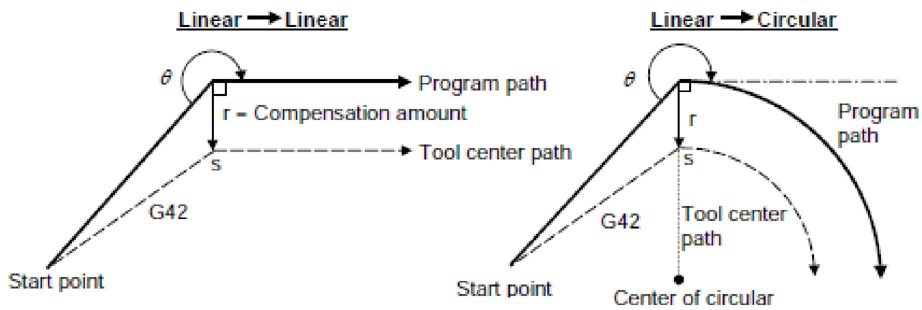
G41: Left compensation of the tool diameter
G42: Right compensation of the tool diameter
D_: The offset number that stores the tool diameter compensation value
X_ Y_ Z_ U_: Instruct the vector of the next command block

(1) Start – Up mode

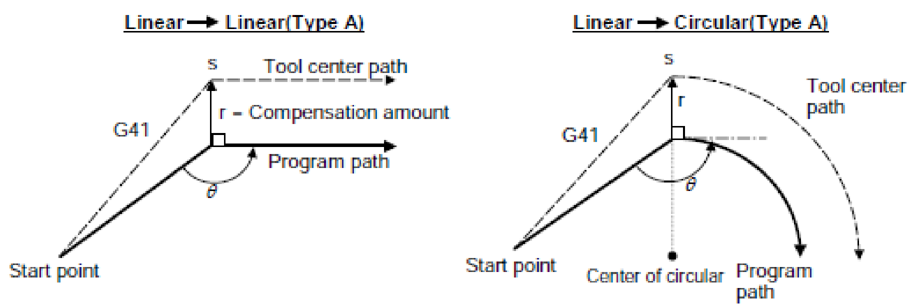
When the tool diameter compensation is started by commanding G41 / G42 in the status of Disable Tool Diameter Compensation, it is called the Start-Up mode.

The case where G41 / G42 are commanded or the axis motion block is commanded for the first time after it is instructed, it is called the Start-Up mode.

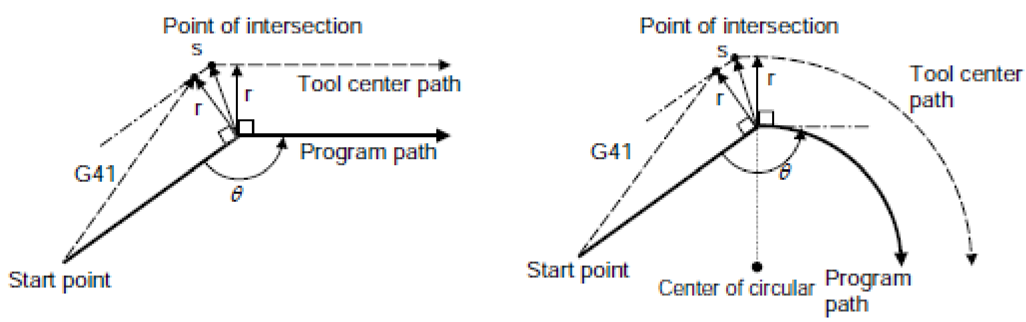
In the Start-Up block, the axis motion command must be greater than the tool radius. In the start-up or cancellation mode, the arc command [G02 / G03] is not executed. When such commands are made, an alarm occurs.



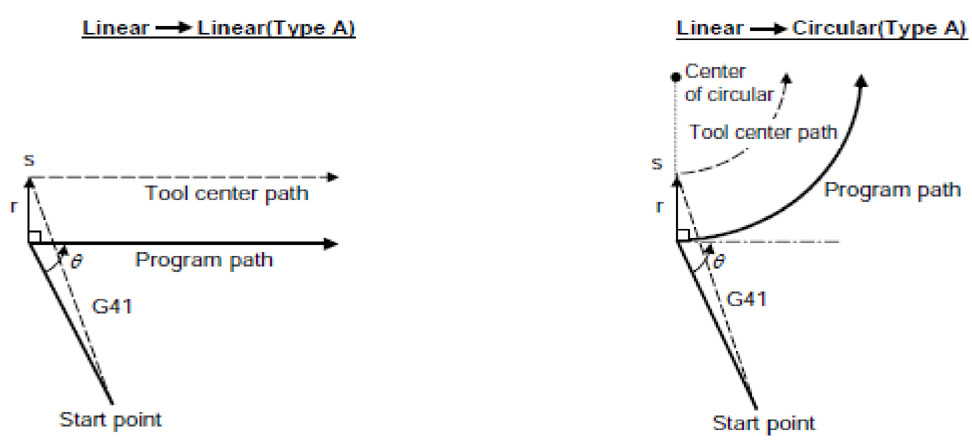
[Tool path of the inner corner]



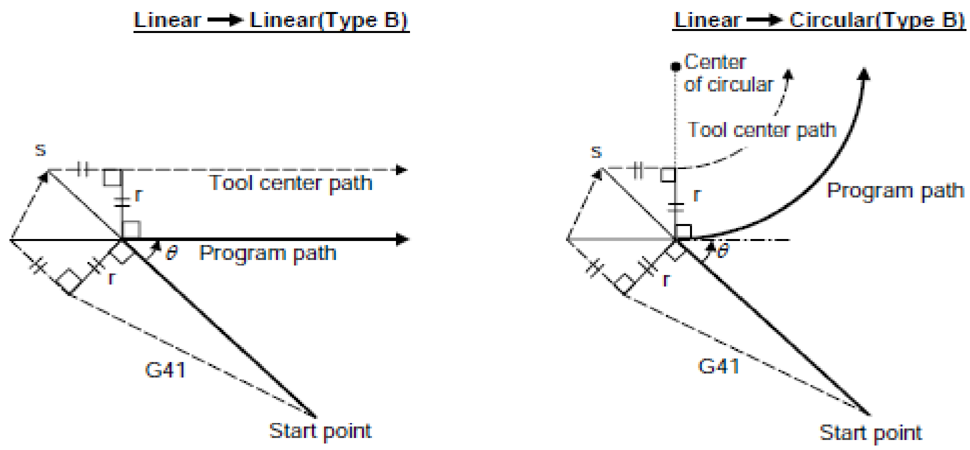
[Tool path of the outer corner (obtuse angle)] (Type A)



[Tool path of the outer corner (obtuse angle)] (Type B)

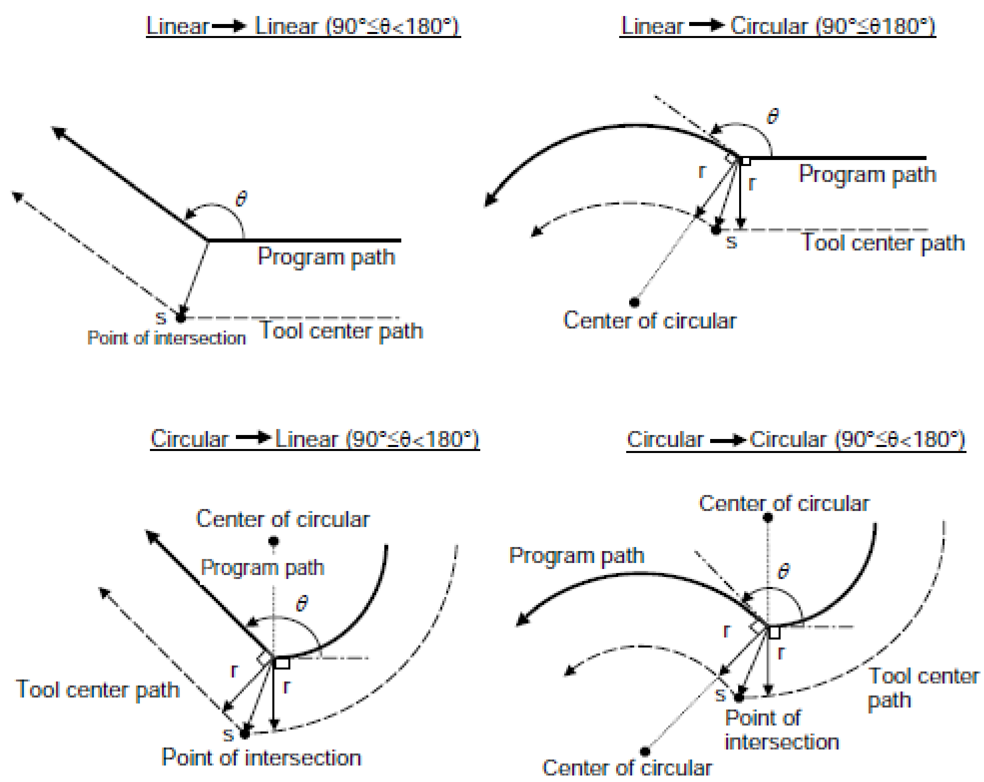


[Tool path of the outer corner (acute angle)] (Type A)

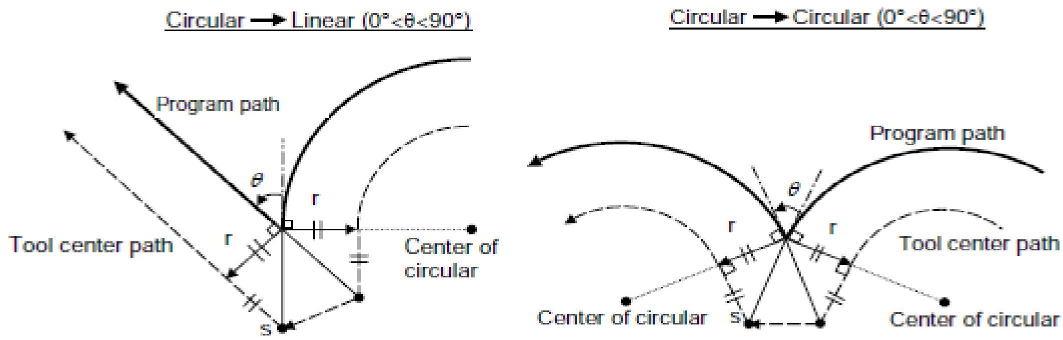


[Tool path of the outer corner (acute angle)] (Type B)

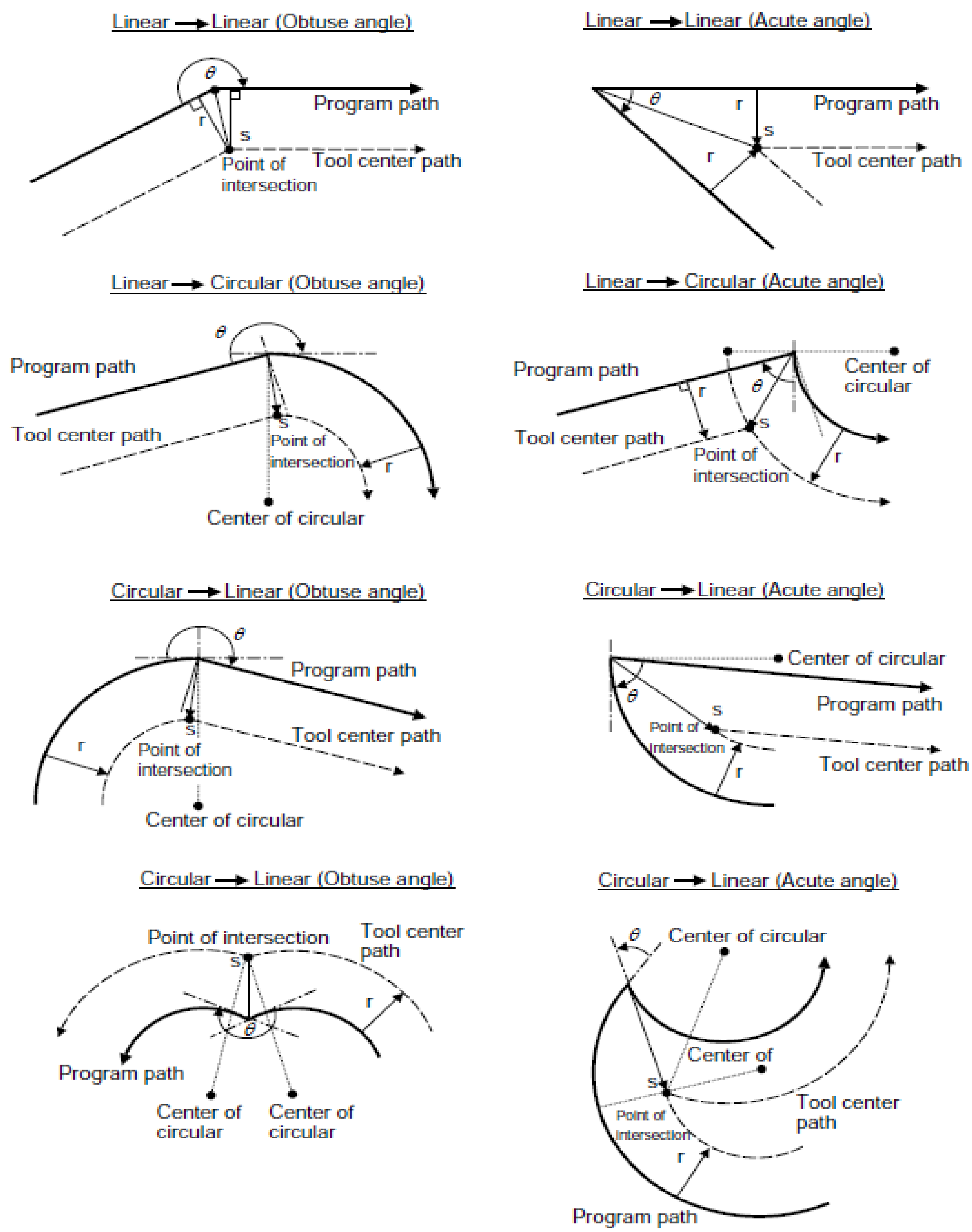
(2) Compensation mode



[outer wall Machining(obtuse angle)]

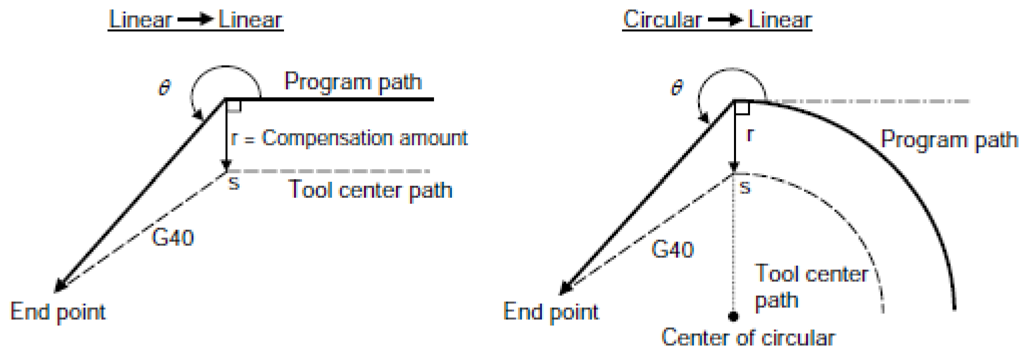


[outer wall Machining(acute angle)]

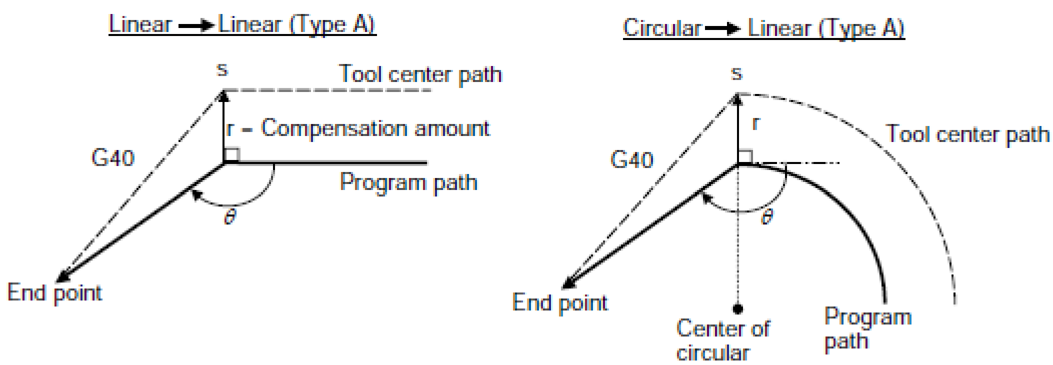


[Inner wall machining]

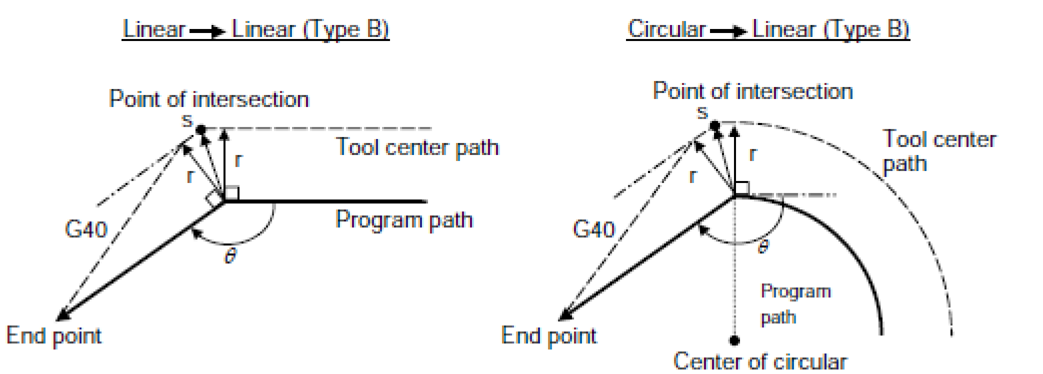
(3) Cancel mode



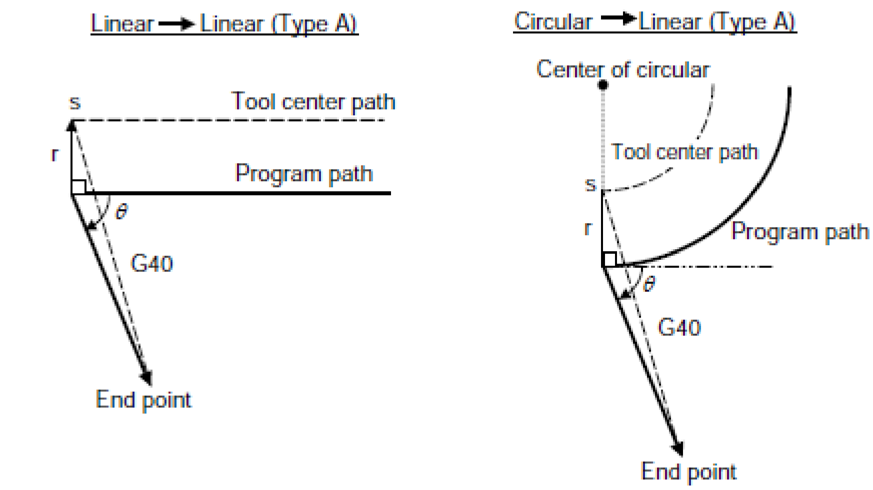
[Tool path of the inner corner]



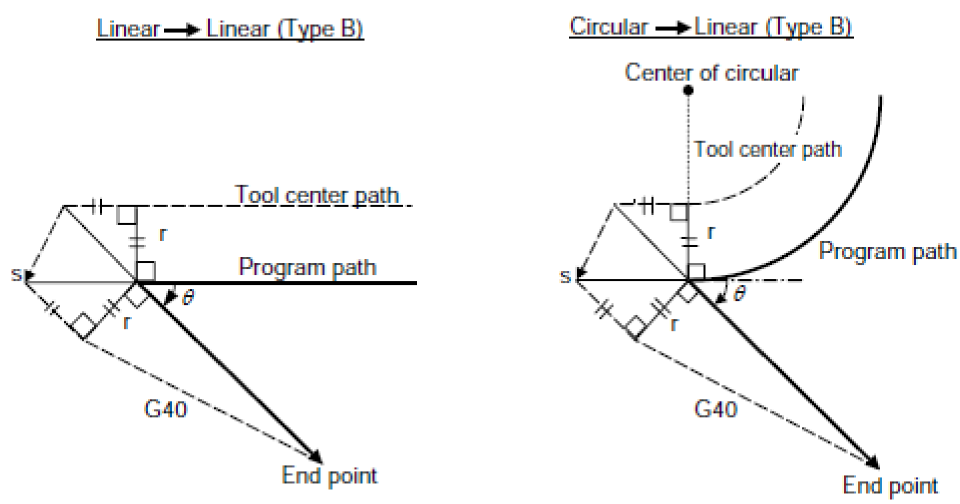
[Tool path of the outer corner (obtuse angle)] (Type A)



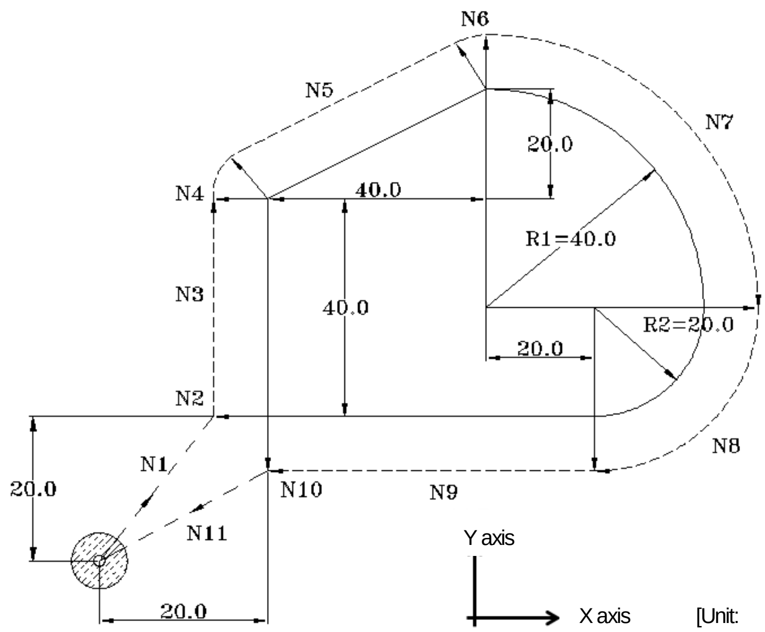
[Tool path of the outer corner (obtuse angle)] (Type B)



[Tool path of the outer corner (acute angle)] (Type A)



[Tool path of the outer corner (acute angle)] (Type B)



N1 G91 G17 G00 G41 X20. Y20, D08 (D08 tool offset number)
N2 G01 Z-25. F100 (The radius value of the tool is entered in the corresponding number)
N3 Y40. F250
N4 G39 X40. Y20, (Circular arc compensation path)
N5 X40 Y20.
N6 G39 X40. (Circular arc compensation path)
N7 G02 X40. Y-40, R40.0
N8 X-20. Y-20. R20
N9 G01 X-60.
N10 G00 Z25.
N11 G40 X-20. Y-20,
N12 M30

The parameters related to tool diameter correction are as follows.

NC Parameter	Group	Name	Channel 1
NC Channel Parameter	Tool diameter compensation	How to Apply the Compensation Value of th	0: Apply the diameter value
		Compensation Type of the Tool Diameter	0: Bypass Traverse
		Whether to check the tool interference duri	0: Do not check
		Compensation amount of the tool diameter	0 mm
		Compensation amount of the tool diameter	0 mm
		Compensation amount of the tool diameter	0 mm
		Compensation amount of the tool diameter	0 mm
		Compensation amount of the tool diameter	0 mm
		Compensation amount of the tool diameter	0 mm
		Compensation amount of the tool diameter	0 mm
		Compensation amount of the tool diameter	0 mm
		Compensation amount of the tool diameter	0 mm
		Compensation amount of the tool diameter	0 mm
		Compensation amount of the tool diameter	0 mm
		Compensation amount of the tool diameter	0 mm
		Compensation amount of the tool diameter	0 mm

* The parameter related to the tool diameter compensation amount is "tool diameter compensation amount 1 ~ tool diameter compensation amount 128".

21) tool length compensation (G43, G44, G49)

```
G43 Z_ H_
G49 Z_
```

G43: Tool length + length compensation
G44: Tool length + length compensation
G49: Cancel the tool length compensation
Z_: Z-axis movement command (in the case of the G17 plane)
Absolute command and incremental command possible
H_: Offset number storing the tool length compensation value

If the offset data that is designated by H code is added to the coordinate values of the terminal of the motion command for the Z axis that is programmed by Absolute Command or Incremental Command, G43 is added so the coordinate value becomes the terminal. A length value can have +, -.

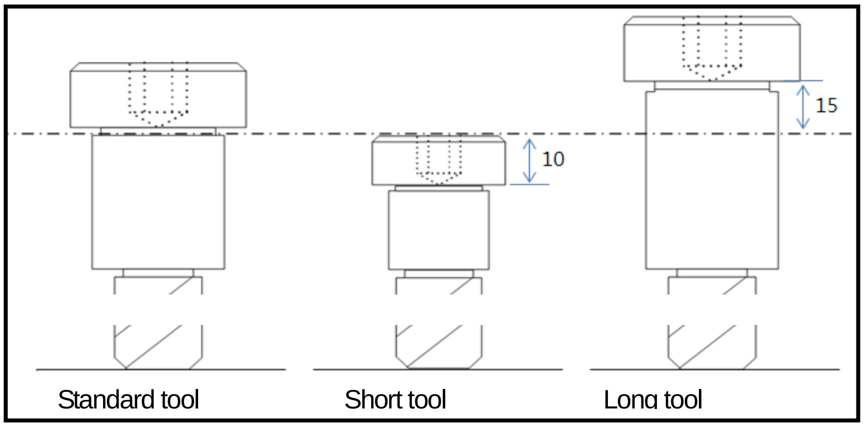
If the motion command of Z is omitted, the length offset is applied in the + direction for G43 and the - direction for G44 in the next block that has the command of Z.

When measuring the tool length first, use the following method

(1) Place the workpiece with a wide top surface on a table.
(2) Bring the end of the reference tool into contact with the plane of the workpiece.
(3) Compensate the Z-axis value.
(4) Replace with the tool to be measured and bring the tip of the tool into contact with the plane.

(5) The Z-axis value of the relative coordinate system in that state is stored in the memory as the tool compensation amount.

With the above settings, the correction amount is set to a - value for a short tool and a + value for a long tool with respect to the reference tool. Therefore, tool length compensation can always be specified only with the G43 during the program.



G43, G44 and G49 are a modal code, which are effective until another code appears. Therefore, the program makes commands G43/G44 right after tool replacement. After finishing the tool work, if commanding G49 before tool replacement, tool length compensation is canceled.

Caution 1. To cancel the offset compensation, command the G49 or H00.

Caution 2. The offset number can be specified up to H00 - H128, and the offset number 00, namely, the offset amount corresponding to H00 always means 0, and it is not possible to set the offset amount corresponding to H00.

Caution 3. It is recommended to create the program like the Z axis movement command for the Enable/Disable Tool Length Compensation commands. The reason is that if it is commanded in the same way as G43 H01, it moves by the tool length (or length compensation amount) input in the length compensation address 01 and if the only G49 is commended, it moves in the opposite direction by the tool length compensation executed before G49, if the tool length compensation value is "+", it may move downward by the tool length from the current position and cause the tool collision.

Therefore, it is recommended to instruct Enable/Disable Tool Length Compensation commands like the Z-axis movement command, and to make it larger than the tool length value.

The parameters related to the tool length compensation are shown below.

<table>
<tr><th>NC Parameter</th><th>Group and parameter name</th></tr>
<tr><td>NC Channel Parameter</td><td>
<table>
<tr><th>Group</th><th>Name</th><th>Channel 1</th></tr>
<tr><td rowspan="5">Tool Length Compensation</td><td>Compensation amount of the tool length 62</td><td>0 mm</td></tr>
<tr><td>Compensation amount of the tool length 63</td><td>0 mm</td></tr>
<tr><td>Compensation amount of the tool length 64</td><td>0 mm</td></tr>
<tr><td>Compensation amount of the tool length 65</td><td>0 mm</td></tr>
<tr><td>Compensation amount of the tool length 66</td><td>0 mm</td></tr>
</table>
* The parameter related to the tool length compensation amount is "tool length compensation amount 1 ~ length compensation amount 128".
</td></tr>
</table>

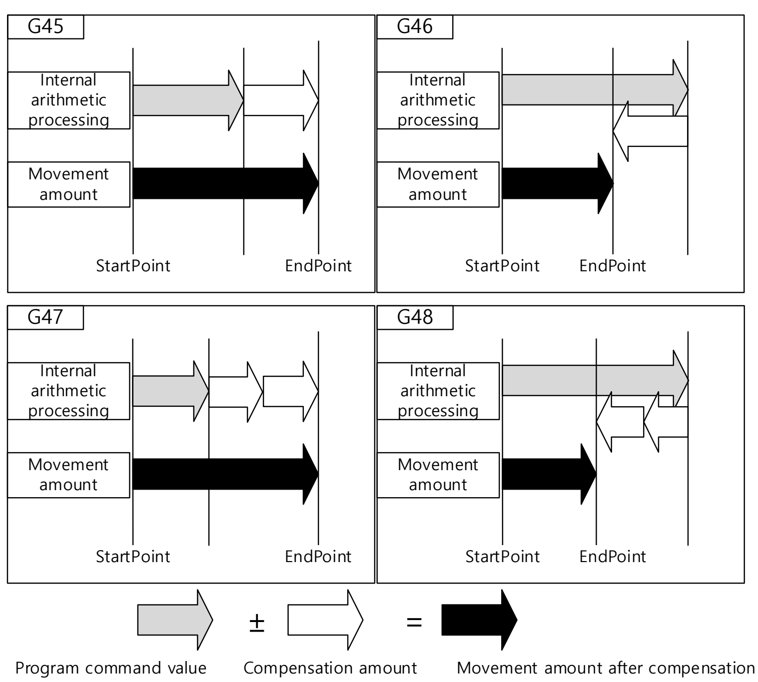
22) Tool position offset (G45~ G48)

```
G45 {G01 / G02 / G03}    X_ Y_ D_
G46 {G02 / G03}    X_ Y_ D_
G47 {G02 / G03}    X_ Y_ D_
G48 {G02 / G03}    X_ Y_ D_
```

G45 : Tool position offset increase
G46 : Tool position offset decrease
G47 : Tool offset double increase
G48 : Tool offset double decrease

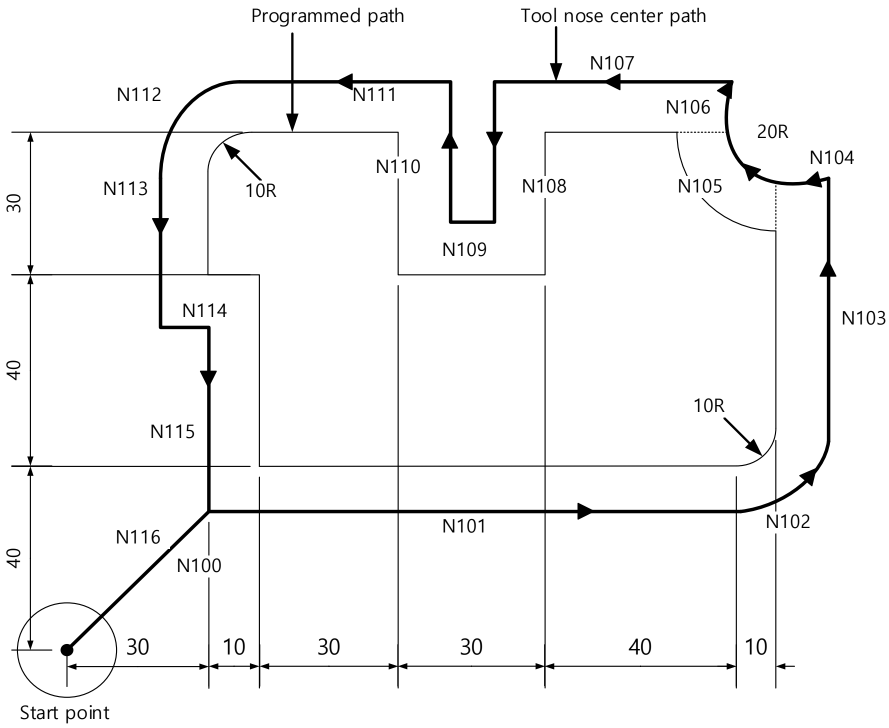
D_ : Tool diameter compensation number
X_ Y_ Z_ : Coordinate value of the target location to move through circular interpolation

During machining the distance between tools can be increased or decreased as much as the tool diameter rather than the width of the work-piece. The tool position compensation function (G45~G48) is the function to increase and decrease the distance to go as much as the tool diameter. The G codes are not modal commands. Therefore, they are valid only for the commanded block. However, the increased or decreased compensation value is continuously valid for the commanded axis. To avoid confusion about H code when designating the compensation value, D code is used. The compensation number is a modal command so that it is continuously valid until another value (compensation number) is commanded or reset. Therefore, it is not needed to command each block.
The compensation number, 0, means that the compensation value is always 0. Increase and decrease are conducted toward the move command. In the case of the absolute command, it increases and decreases toward the commanded position.



Generally the compensation value sets tool radius or tool length with the value of an amount. If the compensation value is set by a negative value, increase and decrease is conducted the other way. It is like setting G45 and G46, G47 and G48 conversely.

For circular arc interpolation, only 1/4 circle and 3/4 circle can be done. In other words, move toward the circular arc that radius is increased or decreased as much as the compensation value and the center of circular arc is the same. Half or perfect circle repeatedly commands 1/4 and 3/4 circles.



N100 G91 G46 G00 X40.0 Y40.0 D01
N101 G45 G01 X100.0 F200
N102 G45 G03 X10.0 Y10.0 J10.0
N103 G45 G01 Y40.0
N104 G46 X0
N105 G46 G02 X-20.0 Y20.0 J20.0
N106 G45 G01 Y0
N107 G47 X-30.0
N108 Y-30.0
N109 G48 X-30.0
N110 Y30.0
N111 G45 X-30.0
N112 G45 G03 X-10.0 Y-10.0 J-10.0
N113 G45 G01 Y-20.0
N114 X10.0
N105 Y-40.0

23) Scaling (G50, G51, Scaling Function)

Scaling function is to reduce or enlarge the size of the programmed shape and then, to program. It can be applied variously by designating the whole magnifications or different magnifications of each axis. The tool offset data is excluded from the subject for scaling.

(1) Increase/decrease by same magnification

```
G50
G51 X_ Y_ Z_ P_
```

G50: Scaling Cancel
G51: Scaling Setting
X_ Y_ Z_: The center coordinate value for the scaling command should be absolute G90. If X_Y_Z_ are omitted, the current location where G51 is commanded becomes the center of scaling.
P_: It is possible to give a command in the form of a real number to an address that designates magnification of scaling. The unit of magnification is always 1/1000.

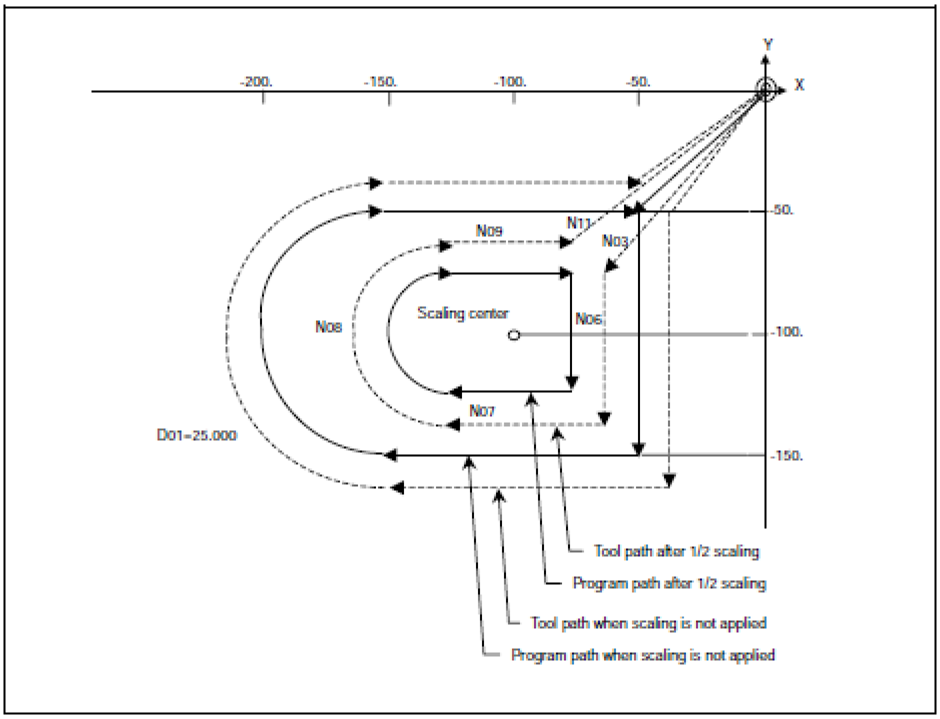
For the motion command after this command, scaling is performed by magnification of P that is designated based on the center of scaling. The scaling mode can be canceled by G50.

The plane selection of circular interpolation can be independently used and should be used with the same block as a circular interpolation command.

(2) Increase/decrease by same magnification

```
G50
G51 X_ Y_ Z_ I_ J_ K_
```

G50: Scaling Cancel
G51: Scaling Setting
X_ Y_ Z_: The center coordinate value for the scaling command should be absolute G590. If X_Y_Z_ are omitted, the location where G51 is commanded becomes the center of scaling.
I_ J_ K_: Scaling magnification for each axis is designated in the relation of I = X axis, J = Y axis, K = Z axis.
It is always 1/1000.



N01 G92 X0 Y0 Z0
N02 G90 G51 X-100. Y-100, P0.5
N03 G00 G43 Z-200. H02
N04 G41 X-50. Y-50, D01
N05 G01 Z-250. F1000
N06 Y-150. F200
N07 X-150
N08 G02 Y-50. J50,
N09 G01 X-50.
N10 G00 G49 Z0
N11 G40 G50 X0 Y0
N12 M02

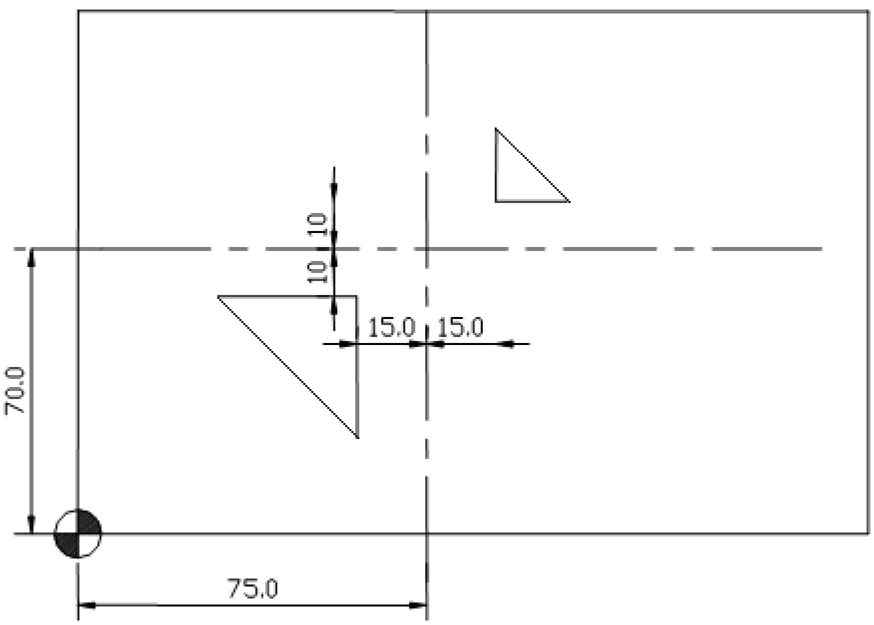
24) Mirror Image(G50, G51, Mirror Image)

```
G50
G51 X_ Y_ Z_ I-_ J-_ K-_
```

G50:
Scaling/Mirror image setting
X_ Y_ Z_: The center coordinate value of mirror image should be absolute G90. If X_Y_Z_ are omitted, the location where G51 is commanded becomes the center of the mirror image.
I-_ J-_ K-_: You should designate in negative number by the use of '-'. The mirror image magnification for each axis should be designated in the relation of I = X axis, J = Y axis, and K = Z axis. It is always 1/1000.

Ex) I-1000: Mirror image of X (1 time)
J-1500: Mirror image of Y (1.5 times)
K-2000: Mirror image of Z (2 times)
If the mirror image magnification for each axis is same, you should use P_ command instead of I_ J_ K_ like a scaling command. You should designate in a negative number by the use of '-'.
(2 times increase of mirror image = P-2000, 1/2 decrease of mirror image = P-500)

The mirror image function is made by adding the mirror function to the scaling function. Therefore, it is possible to reduce/enlarge the programmed forms and at the same time, it is possible to symmetrically move. If you designate I_ J_ K_ (or P_) in positive numbers when using mirror image G51, the scaling function works. If you designate in negative numbers, the mirror image function works.



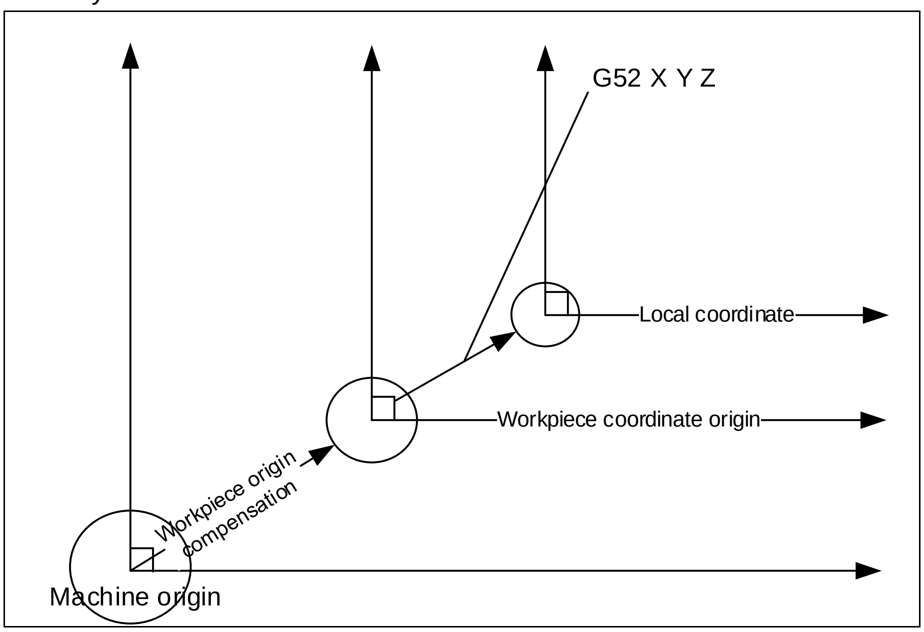
G54 G91 XZ0.
G54 G90 G00 X0. Y0,
G52 G90 X90. Y80, (Setting of local coordinate system 1)
G00 X0. Y0,
M98 P1234 (Auxiliary program call)
Z2
G52 X60. Y80, (Setting of local coordinate system 2)
G51 X0. Y0, I-1500, J-1500, (1.5 times increase of scale)
(Mirror image for X and Y ON)
M98 P1234
G50
G52 X60. Y60, (Local coordinate cancel)

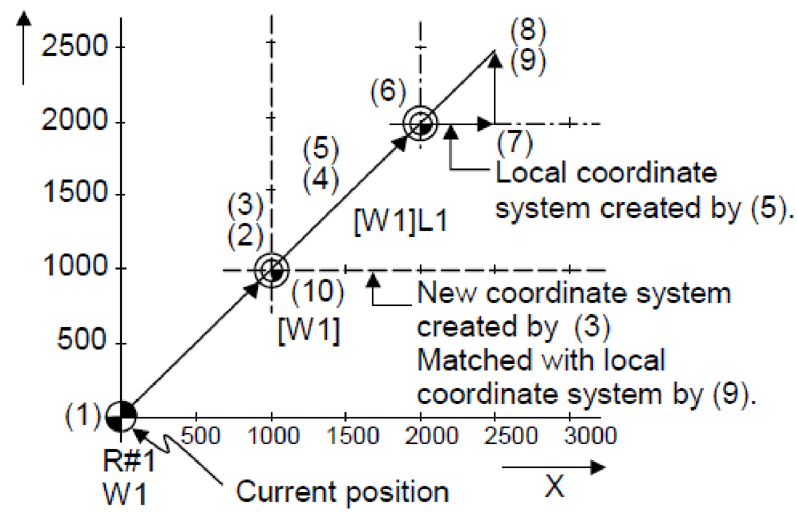
25) Set the local coordinate system

G52 X_ Y_ Z_

G52: Set the local coordinate system

The local coordinate system is the coordinate system that sets and uses the reference point at an arbitrary point based on the set workpiece coordinate system when creating the program by the workpiece coordinate system. With the local coordinate system command, the new coordinate system, namely, the local coordinate system can be set in all workpieces coordinate systems (G54 to G59). The origin of each local coordinate system is the X_Y_Z_ locations specified by each workpiece coordinate system. The local coordinate system is cleared to 0 when a new workpiece coordinate system is set.





G28X0Y0

G00G90X1, Y1,

G92X0Y0

G00X500Y500

G52X1, Y1,

G00X0Y0

G01X500F100

Y500

G52X0Y0

G00X0Y0

26) Select the machine coordinate system(G53)

```
G90 G53 X_ Y_ Z_
```

G90: absolute command
G53: Select the machine coordinate system
X_ Y_ Z_: feed position

G53 is the command to use the machine coordinate system and the tool moves rapidly to the X_Y_Z_ position above in the machine coordinate system. G53 is the stand-alone G code so it is valid in the commanded block only. It is still valid in the absolute command (G90) but becomes ineffective in the incremental command (G91). If you want to move the tool to the machine-specific position such as a tool change position, you should program it in the machine coordinate system with G53. The tool diameter compensation, tool length compensation, and tool position compensation must be canceled before the G53 command, otherwise, it will be moved to the compensated state. In addition, since the machine coordinate system must be set before the G53 is commanded, manual homing or homing with G28 should be executed after turning on the power.

```
G40 G80
G53 G90 X-140 Y-120 Z0       (Move to position X-140 Y-120 Z0 of machine coordinate system)
G92 X0 Y0 Z150               (Reset by changing the workpiece coordinate system)
G30 G91 Z0
G54 G00 G90 X0 Y0
M30
```

27) Select workpiece Coordinate System1~6(G54, G55, G56, G57, G58, G59)

```
G54 X_ Y_ Z_
G55 X_ Y_ Z_
G56 X_ Y_ Z_
G57 X_ Y_ Z_
G58 X_ Y_ Z_
G59 X_ Y_ Z_
```

G54: Select the workpiece coordinate system 1
G55: Select the workpiece coordinate system 2
G56: Select the workpiece coordinate system 3
G57: Select the workpiece coordinate system 4
G58: Select the workpiece coordinate system 5
G59: Select the workpiece coordinate system 6
X_ Y_ Z_: Position of the workpiece coordinate system

The coordinate system used for workpiece machining is called the workpiece coordinate system. This is the coordinate system that allows the operator to create a program conveniently on the basis of drawings, and to set any point of the workpiece to be machined as the origin by applying the NC program as it is. After turning on the power, it is necessary to executing homing for proper application of the coordinate system. When using G54 ~ G59, it is not necessary to set the coordinate system with G92.

```
G40 G80
G28 G91 X0 Y0 Z0        (Returning to machine origin where the waypoint is the current position value [G91mode])
G54 G00 G90 X0 Y0 Z0    % Use the 54 workpiece coordinate system and traverse it to the origin rapidly. i.e. G54
                        % Traverse to the origin of the coordinate system rapidly
M30
```

28) Single direction positioning (G60)

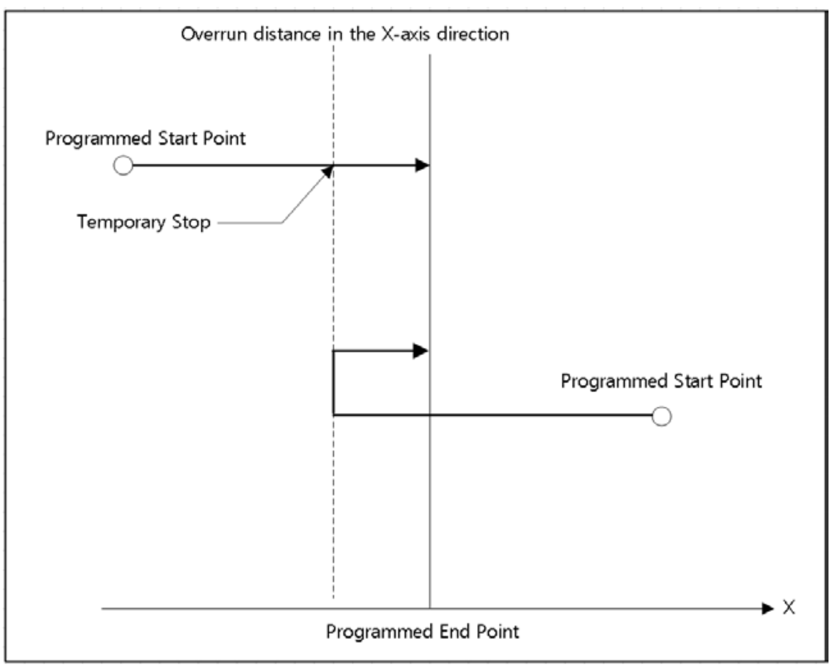
G90 G60 X_ Y_ Z_ U_

G60: Single Direction Positioning command
G00: Positioning command
X_ Y_ Z_ U_: Target position to traverse

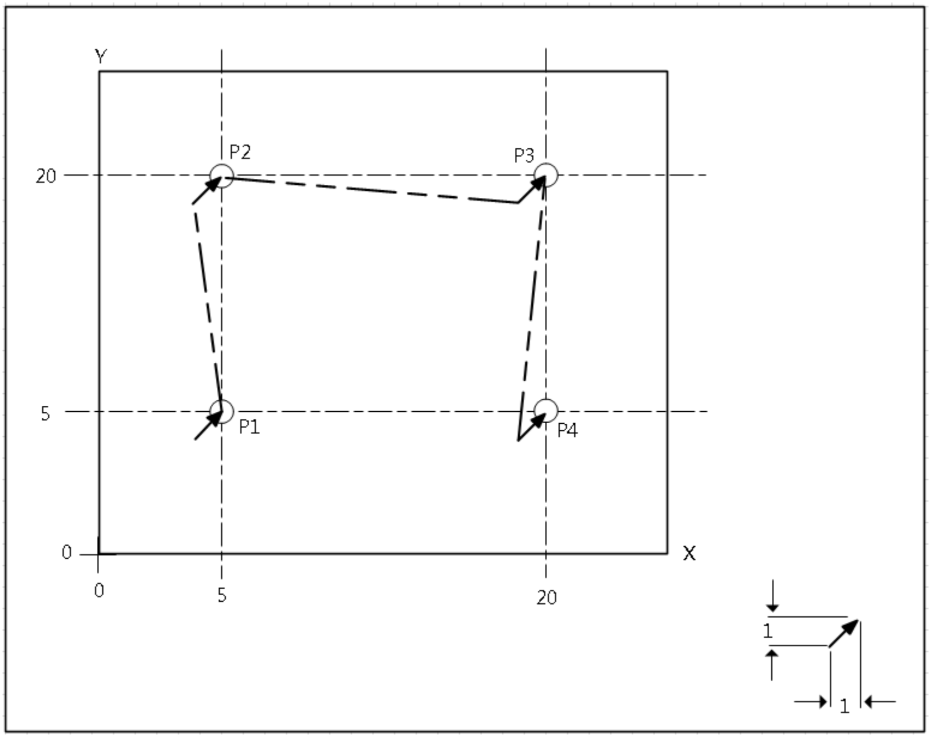
Single Direction Positioning (G60) is the function used for tool traversing, which replaces the Rapid Traverse or runs last. After stopping at the position separated by the overrun stroke set for the commanded positioning direction, it moves to the end position and obtains the effect of backlash compensation.
Therefore, the G60 command is applied, it always moves from the same direction to the target position.
The overrun amount is stored in the parameters.
Please refer to the overrun feed amount of the single direction positioning of NC channel / axis parameters.

NC Parameter	Group and parameter name
NC Channel/Axis Parameter	Group: Auxiliary Function; Name: Overrun distance in single dir. positioning; X Axis: 0 mm

Note that the single direction positioning command does not eliminate the backlash physically.



If the positioning direction is specified for the X + direction as shown in the figure, it will always move from the same direction to the target position.
If the overrun amount is not set or the feed amount is 0, the single direction positioning command is not applied.
In addition, it does not apply to the Z axis in the drill cycle, and it is not affected by the mirror image for the set direction.



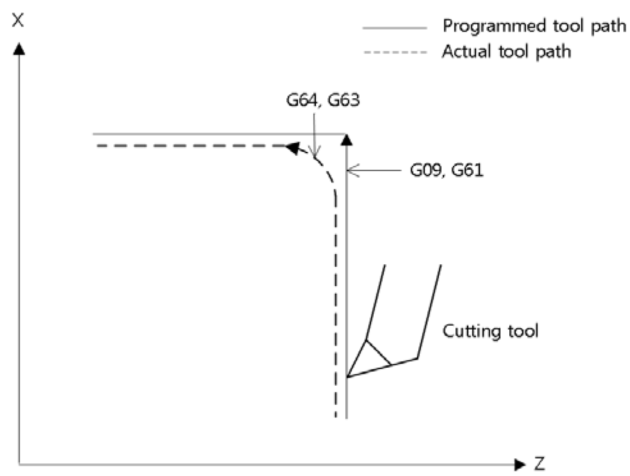

```
G21
G17 G40 G80 T01
M06
G90 G54 G60 X5.0 Y5.0
S1200 M03 T02
G43 Z2.5 H01 M08
G99 G82 R2.5 Z-2.0 P200
F150.0
G60 Y20.0
G60 X20.0
G60 Y5.0
G80 Z2.5 M09
G28 Z2.5 M05
M01
```

29) Exact Stop Mode(G61)

G61

G61: Exact Stop Mode command

The Exact Stop mode (G61) command is used to avoid not exactly reaching the location designated in the previous block due to NC's continuous execution between continuous blocks in the cutting feed.
If exact stop mode is commanded, motion speed is reduced to 0 in the end point of motion of a block. After confirming if reaching the command position, execute the next block. Exact Stop mode (G61) is a modal command while G09 is a one-shot command. G61 is continuously applied to the cutting feed until G62 (automatic corner override mode), G63 (tapping mode) and G64 (cutting mode) are commanded.



[Figure] Tool Path between Continuous Blocks

```
G54 G00 X0 Y0
G61 G01 X10 F50
Y100
G64 X0
Y0
M30
```

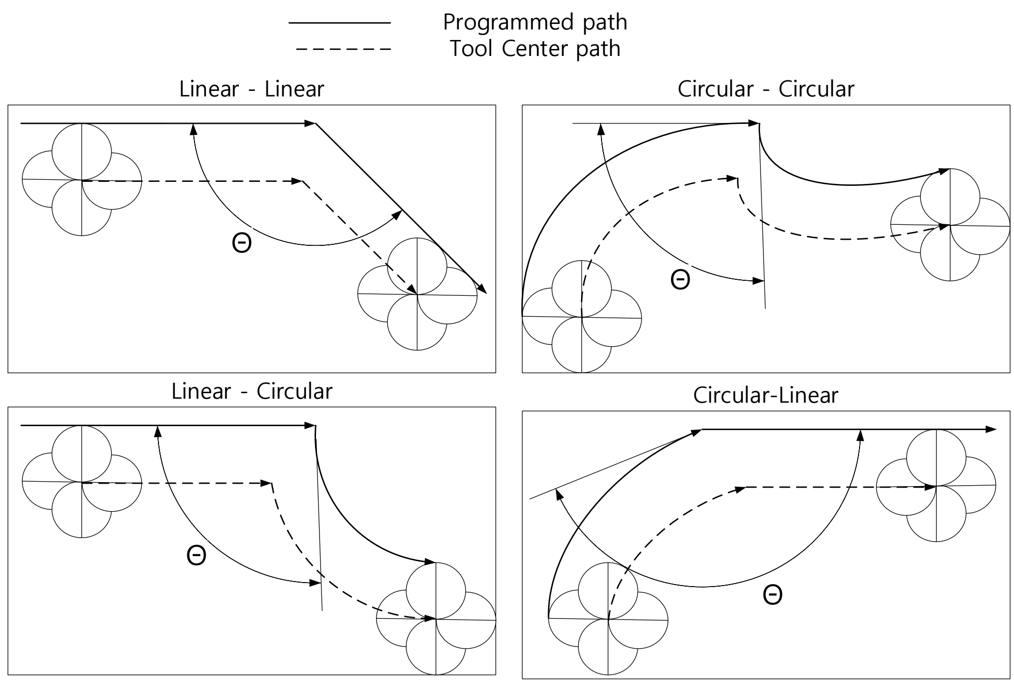
30) Automatic corner override mode(G62)

G62

G62 : Automatic corner override mode command

This function enables smooth feed by automatically reducing the tool feed speed to prevent the load on the tool from increasing while moving to the inner corner in the tool diameter compensation mode (G41/G42).

There are 4 types of inner corners to which automatic corner override mode is applied. Corner angle Θ ranges from 2° ≤ Θ ≤ Θp ≤178°.



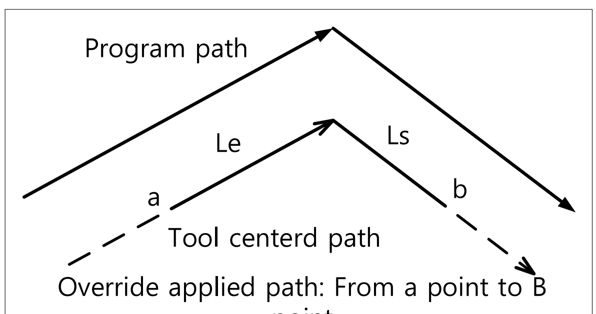
Θp is a parameter as shown below.

NC Parameter	Automatic corner override		
NC Channel Parameter	Automatic Corner Override	Start Interval of Automatic Corner Override	0
		End Interval of Automatic Corner Override	0
		Scale of Automatic Corner Override	100
		Angle Between Auto Corner Override	178
		Cutting Feed Override Settings	0: Applies to internal corners only when calibrating tool path
		Minimum rate of internal arc cutting speed	0
		Automatic Corner Override Speed	0

Override is applied to the feed rate from the start section to the end section of the corner when it is determined as the inner corner.
The start section and end section are saved in parameters. The save location is as follows.

NC Parameter	Automatic corner override		
NC Channel Parameter	Automatic Corner Override	Start Interval of Automatic Corner Override	0
		End Interval of Automatic Corner Override	0
		Scale of Automatic Corner Override	100
		Angle Between Auto Corner Override	178
		Cutting Feed Override Settings	0: Applies to internal corners only when calibrating tool path
		Minimum rate of internal arc cutting speed	0
		Automatic Corner Override Speed	0

Linear distances Le and Ls to which feedrate override is applied are the linear distances between a point on the tool center path and the intersection of the corners.



The amount of override in the corner section where the override is applied is set as a parameter as follows.

NC Parameter	Automatic corner override		
NC Channel Parameter	Automatic Corner Override	Start Interval of Automatic Corner Override	0
		End Interval of Automatic Corner Override	0
		Scale of Automatic Corner Override	100
		Angle Between Auto Corner Override	178
		Cutting Feed Override Settings	0: Applies to internal corners only when calibrating tool path
		Minimum rate of internal arc cutting speed	0
		Automatic Corner Override Speed	0

The feed rate per minute is set as follows.

Deceleration section feed rate =	Command feed rate × auto corner override amount × feed rate override amout

Automatic corner override is not applied during acceleration/deceleration before interpolation.
Also, if the override amount is set to 0, the feed does not proceed.

```
G54 G00 X0. Y0, Z0,
G41 D1
G62 G01 X50. F100,
G03 X100. Y50, R10,
G01 X0.
Y100,
M30
```

* Always Automatic Corner Override
Automatic corner override can be applied to the entire cutting feed as well as tool diameter compensation.
f the parameter setting is 0, it is the normal automatic corner override mode, and if it is 1, the automatic corner override function is always available.

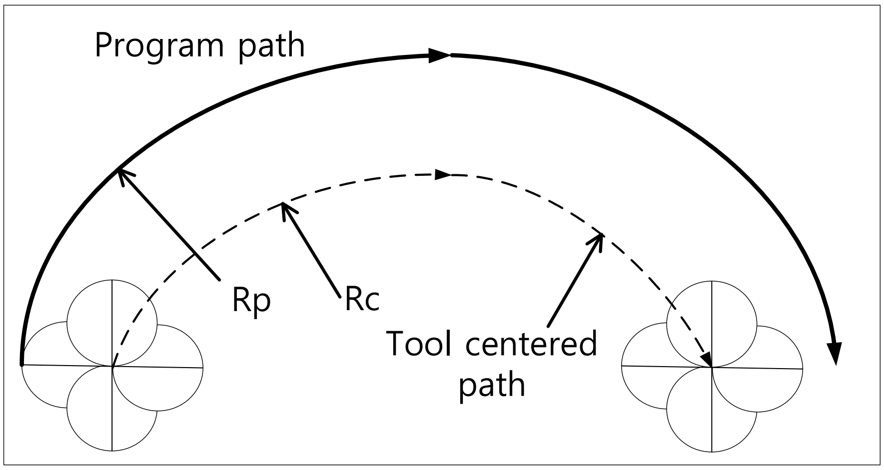
NC Parameter	Automatic corner override		
NC Channel Parameter	Automatic Corner Override	Start Interval of Automatic Corner Override	0
		End Interval of Automatic Corner Override	0
		Scale of Automatic Corner Override	100
		Angle Between Auto Corner Override	178
		Cutting Feed Override Settings	0: Applies to internal corners only when calibrating tool path
		Minimum rate of internal arc cutting speed	0
		Automatic Corner Override Speed	0

* Automatic Corner Feed Rate
the feed rate can be set directly during automatic corner override.
It is applied when the value set in the parameter is greater than 0, and when the set speed is greater than the command speed, it operates at the command speed.

NC Parameter	Automatic corner override		
NC Channel Parameter	Automatic Corner Override	Start Interval of Automatic Corner Override	0
		End Interval of Automatic Corner Override	0
		Scale of Automatic Corner Override	100
		Angle Between Auto Corner Override	178
		Cutting Feed Override Settings	0: Applies to internal corners only when calibrating tool path
		Minimum rate of internal arc cutting speed	0
		Automatic Corner Override Speed	0

* Inter Circular Cutting Feed Rate Change
The inner arc section is automatically decelerated during tool diameter compensation in the same way as the inner corner.
If Rp is very large compared to Rc, Rc/Rp becomes close to 0 and the feed rate becomes 0. Therefore, the minimum value of Rc/Rp can be set in the parameter and used.

NC Parameter	Automatic corner override		
NC Channel Parameter	Automatic Corner Override	Start Interval of Automatic Corner Override	0
		End Interval of Automatic Corner Override	0
		Scale of Automatic Corner Override	100
		Angle Between Auto Corner Override	178
		Cutting Feed Override Settings	0: Applies to internal corners only when calibrating tool path
		Minimum rate of internal arc cutting speed	0
		Automatic Corner Override Speed	0



If there is a circular arc in the corner while automatic corner override is applied, the feed rate is applied as follows.

Deceleration section feed rate = Command feed rate × Rx/Rp × auto corner override amount × feed rate override amout

31) Tapping Mode (G63)

G63

G63: Tapping Mode Command

Tapping mode (G63) command carries out the cutting feed the same as cutting mode. A tool is not decelerated at the end point of a block and executes the next block.
Cutting feedrate override is fixed at 100% and the function of Feed Hold is not applied.
This function, as a modal command (group 15), is continuously applied to the cutting feed until G61 (Exact Command Mode), G62 (Automatic Corner Override Mode) and G64 (Cutting Mode) are commanded.

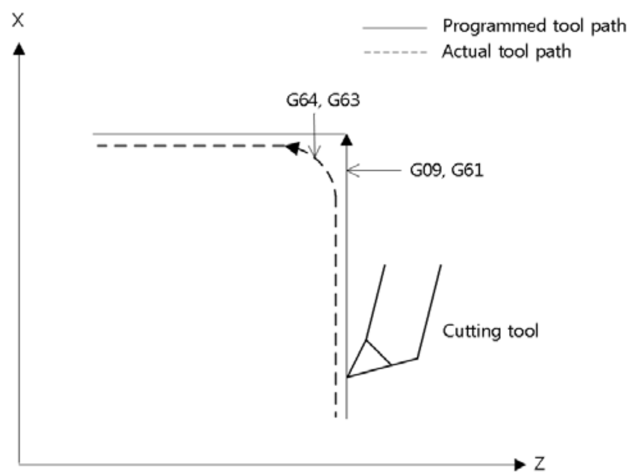
```
G54 G00 X0. Y0, Z0,
G63 G01 X50. F100,
Y100,
M30
```

32) Cutting mode(G64)

G64

G64: Cutting Mode command

The Cutting mode (G64) command is the feed mode that is set as the default of the cutting feed mode. A tool is not decelerated at the end point of a block and executes the next block.
But as in the real feed, the current block is previously decelerated and the next block is previously accelerated, a phenomenon of rounding happens at the corner. This function, a modal command (group 15), is valid until G61 (Exact Command Mode), G62 (Automatic Corner Override Mode) and G63 (Tapping Mode) are commanded.



[Figure] Tool Path between Continuous Blocks

```
G54 G00 X0 Y0
G64 G01 X50 F100
Y100
G61 X0
Y0
M30
```

33) Macro call(G65)

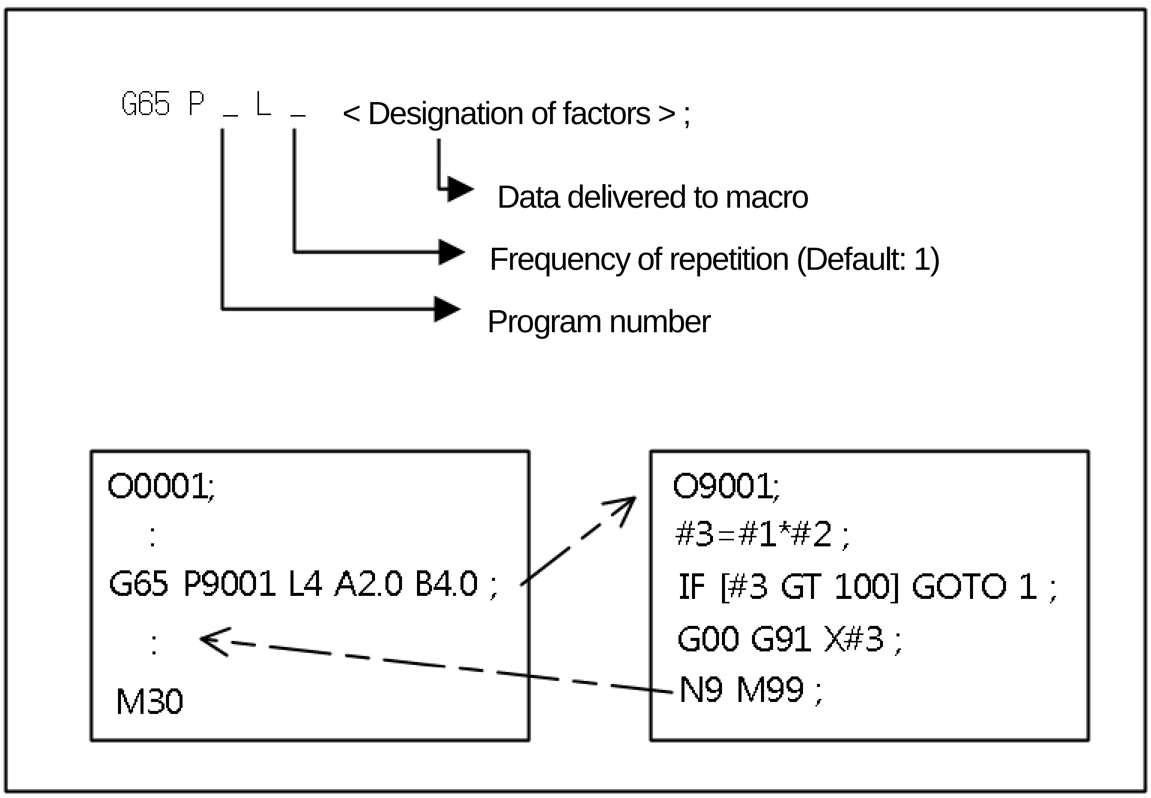
G65 P_ L_ <Factors designated>

G65: Macro call
P_: Program Number
L_: Frequency of repetition

A custom macro can be called with the macro call (G65) command.

A custom macro has several features different from a Sub Program.

- ✓ Factors (Data) can be designated and operation among factors is possible.
- ✓ Factors are classified into local variables, common variables and system variables.
- ✓ Variables always have the values in the form of a real number.
- ✓ The program number should be 4-digits.



Custom macro that is designated as program addresses can be called with macro call (G65) command.
And data variables can be delivered by designating factors.
If it is needed to designate frequency of repetition, set the range value of 1~9999 after L.
If input of repetition frequency is omitted, 1 is set as default.

Designation of factors have 3 methods as follows:

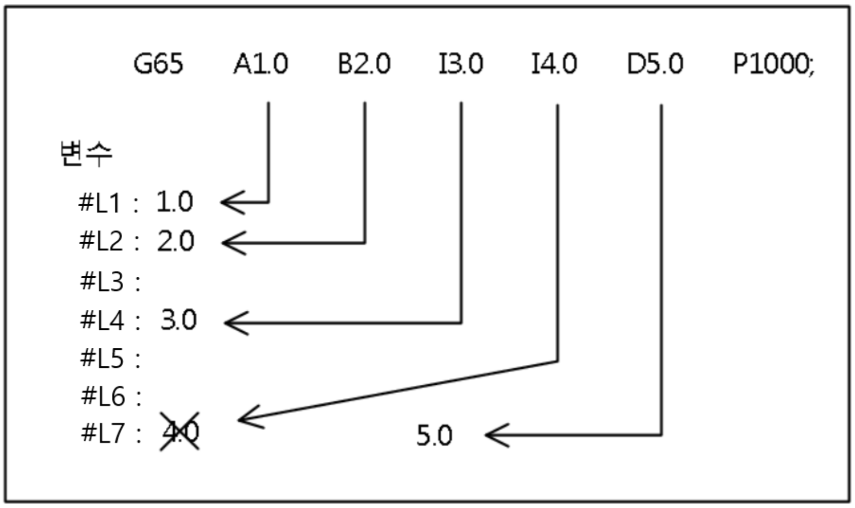
- Factors designated 1
  - Use one alphabet from A to Z except G, L, O, N and P.
  - I, J and K should be designated in alphabetic order but other variables should not be in alphabetic order.

Address	Variable number in macro	Address	Variable number in macro	Address	Variable number in macro
A	#L1	I	#L4	T	#L20
B	#L2	J	#L5	U	#L21
C	#L3	K	#L6	V	#L22
D	#L7	M	#L13	W	#L23
E	#L8	Q	#L17	X	#L24
F	#L9	R	#L18	Y	#L25
H	#L11	S	#L19	Z	#L26

- Factors designated 2
  - It is possible to designate factors with address A, B, C. It is possible to designate the factor consisting of a set of addresses I, K, J up to a maximum of 10 sets.
  - The number in () means the order of designating factors is not used when it is actually commanded.
  -

Address	Variable number in macro	Address	Variable number in macro	Address	Variable number in macro
A	#L1	K(3)	#L12	J(7)	#L23
B	#L2	I(4)	#L13	K(7)	#L24
C	#L3	J(4)	#L14	I(8)	#L25
I(1)	#L4	K(4)	#L15	J(8)	#L26
J(1)	#L5	I(5)	#L16	K(8)	#L27
K(1)	#L6	J(5)	#L17	I(9)	#L28
I(2)	#L7	K(5)	#L18	J(9)	#L29
J(2)	#L8	I(6)	#L19	K(9)	#L30
K(2)	#L9	J(6)	#L20	I(10)	#L31
I(3)	#L10	K(6)	#L21	J(10)	#L32
J(3)	#L11	I(7)	#L22	K(10)	#L33

◆ Combination designations
- Even if factors are designated by combining designation of factors 1 and 2, alarms do not occur.
- In this case, factor address follow designation 1. Variable



As in the figure, in case of a combination designation, the address according to the designation of factors 1 is valid. Therefore, D5.0, not I(2)4.0, is valid in #L7 address.

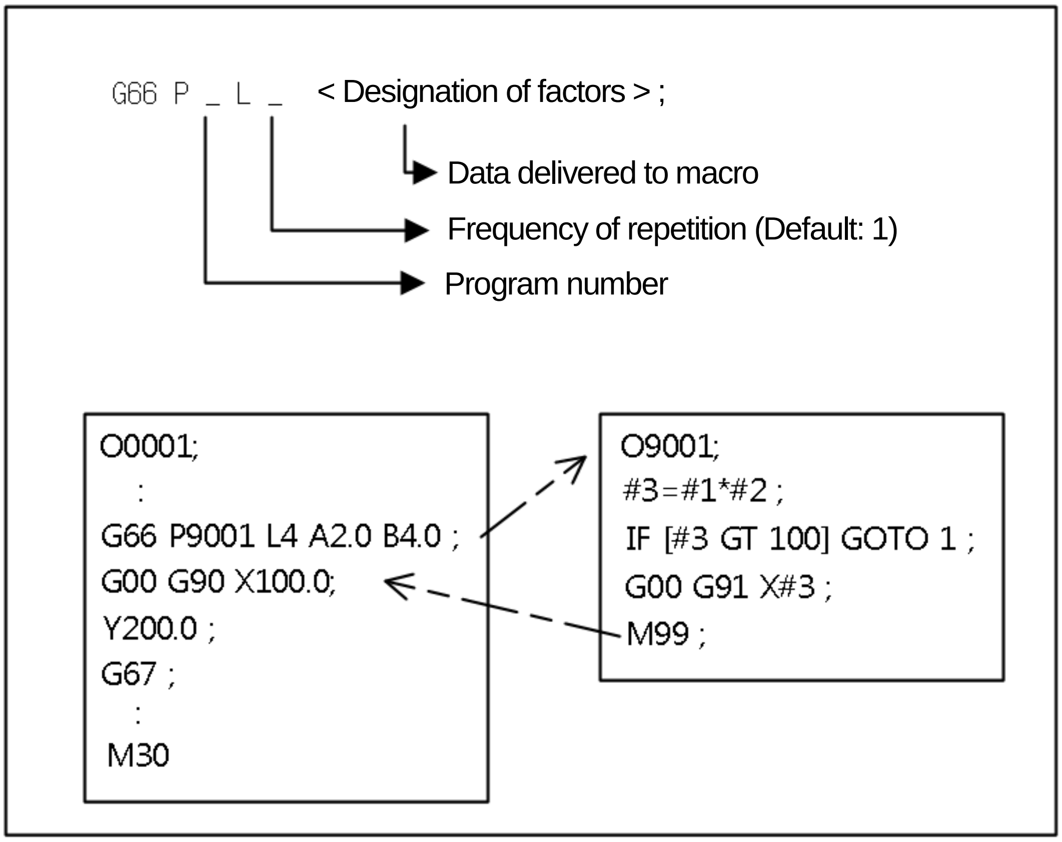
34) Modal modal call/cancel (G66/G67)

G66 P_ L_ <Factors designated>

G67

G66: Macro modal call
G67: Macro modal call cancel
P_: Program Number
L_: Frequency of repetition

Factor designation of the macro modal call (G66)/cancel (G67) command is the same as macro all (G65).



After executing the block that motion is commanded during a macro call, the macro call is carried out.
If G66 is commanded again during a macro modal call (G66), the next modal call can be carried out.
If G67 is commanded, the macro call is not carried out from the next block.
G66 and G67 always exist in the same program.

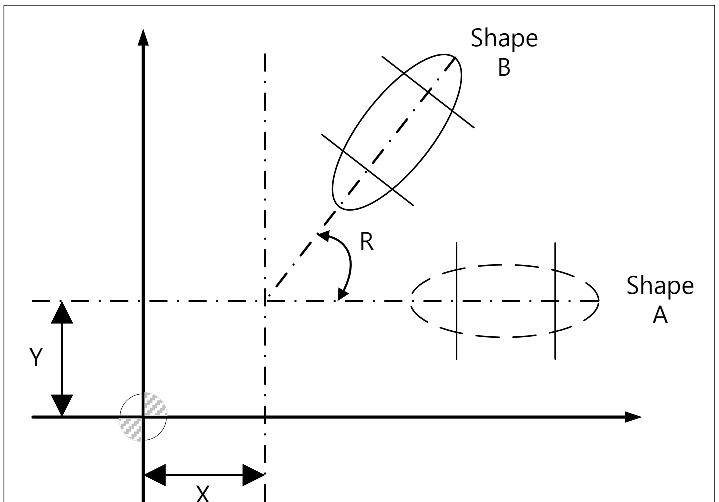
35) Coordinate Rotation(G68 / G69)

```
G17 G68 X_ Y_ R_
G18 G68 Z_ X_ R_
G19 G68 Y_ Z_ R_
G69
```

G68 : Coordinate System Rotation
G69 : Coordinate System Rotation Cancel

G17 X_ Y_ : G17 plane and coordinate value of rotation center.
G18 Z X_ : G18 plane and coordinate value of rotation center.
G19 Y_ Z_ : G19 plane and coordinate value of rotation center.
R_ : rotation angle

By commanding the rotation center and rotation angle of the shape through the coordinate system rotation command (G18), it can be rotated by an arbitrary angle. In addition, the rotated shape can be created and used as an auxiliary program.



In coordinate rotation in the G17 plane, the counterclockwise rotation direction is the + direction of the rotation angle command.

G68 command should be commanded in G00, G01 mode.
If there is a decimal point in the rotation angle, it is an angle unit.
G69 can be commanded in the same block as other commands.
G27, G28, G29, G30, G92 must be commanded in G69 mode.
Among tool diameter compensation, G68 and G69 commands are available. At this time, the rotation plane must be the plane where tool radius compensation is performed.
It is also possible to call a subprogram while changing the rotation angle.
When the coordinate system rotation is commanded in the scaling mode (G51), the scale is applied to the coordinate system center point. However, the scale is not applied to the rotation angle. If there is an axis feed command, the rotation is applied after the scale is applied.

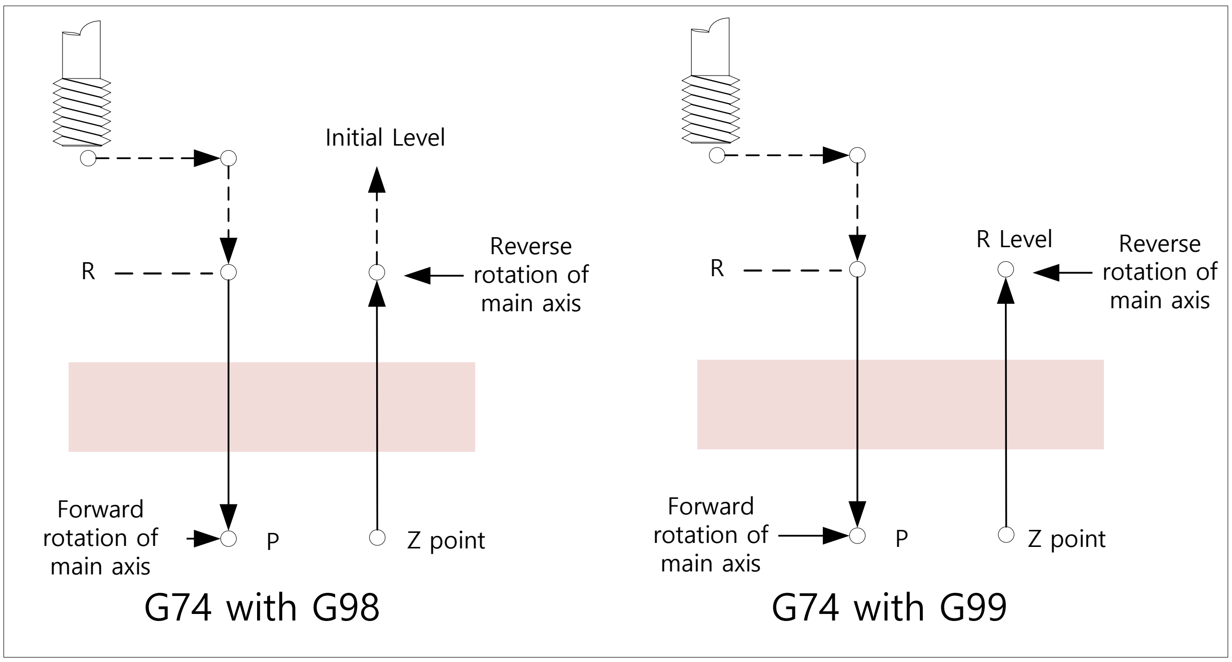
36) Counter Tapping Cycle

G74 [G90/G91] [G98/G99] X_ Y_ Z_ R_ F_ K_

G74 : Counter Tapping Cycle

X_ Y_ : Position of the hole
Z_ : Depth of the hole
R_ : Position of R point
F_ : Cutting feed rate
K_ : Number of repeats

The Counter Tapping Cycle (G74) command is the function that is useful in creating counter screws. After it rapidly moves to R point after positioning, execute tapping up to the Z point. Dwell is carried out when tapping ends. It is the cycle function that the master axis directly revolves (M3) and moves to the R point after stopping.



M4 S100	%main axis start
G90 G98 G74 X300. Y-250. Z-150. R-120. F120.	%After positioning, Screw hole 1 cutting, Return to initial point
Y-550.	%After positioning, Screw hole 2 cutting, Return to initial point
Y-750.	%After positioning, Screw hole 3 cutting, Return to initial point
X1000,	%After positioning, Screw hole 4 cutting, Return to initial point
Y-550.	%After positioning, Screw hole 5 cutting, Return to initial point
Y-750.	%After positioning, Screw hole 6 cutting, Return to initial point
G80 G28 G91 X0 Y0 Z0	%Reference return
M5	%main axis stop

37) Canned Cycle Cancel (G80)

G80

G80 : Canned Cycle Cancel

The Canned Cycle Cancel (G80) command is the function that releases all drilling data and aborts the canned cycle function. As it aborts canned cycle data such as R point and Z point, R point and Z point become 0 through incremental command.

```
M3 S100                                   %main axis  start
G90 G99 G88 X300. Y-250. Z-150. R-120. F120.   %After positioning, Screw hole 1 cutting, Return to R point level
Y-550.                                    %After positioning, Screw hole 2 cutting, Return to R point level
Y-750.                                    %After positioning, Screw hole 3 cutting, Return to R point level
X1000.                                    %After positioning, Screw hole 4 cutting, Return to R point level
Y-550.                                    %After positioning, Screw hole 5 cutting, Return to R point level
G98Y-750.                                 %After positioning, Screw hole 6 cutting, Return to initial level
G80 G28 G91 X0 Y0 Z0                      %Reference return
M5                                        %main axis stop
```

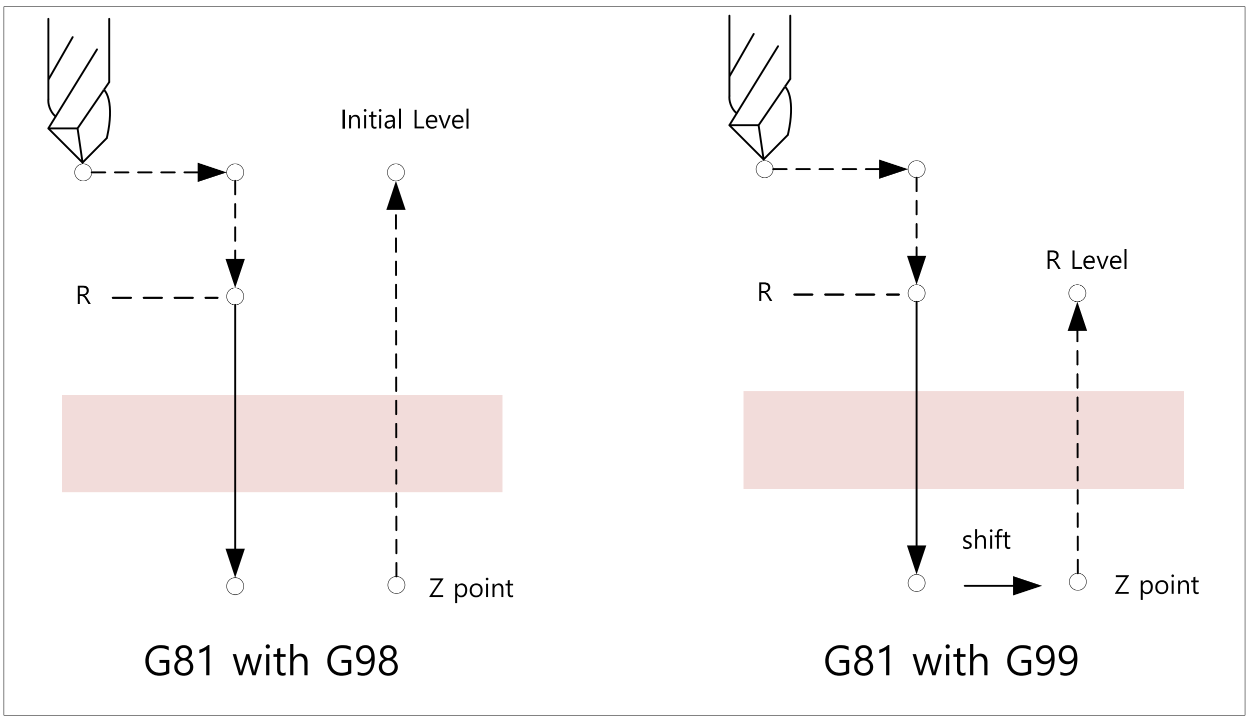
38) Drilling Cycle/Spot Drilling Cycle

G81 [G90/G91] [G98/G99] X_ Y_ Z_ R_ F_ K _

G81 : Drilling Cycle/Spot Drilling Cycle

X_ Y_ : Position of the hole
Z_ : Depth of the hole
R_ : Position of R point
F_ : Cutting feed rate
K_ : Number of repeats

The Drilling Cycle/Spot Drilling Cycle (G80) commands are used for general drilling, reaming and spot boring. The tool does cutting feed to the bottom of a hole and then goes out of it at a rapid traverse.



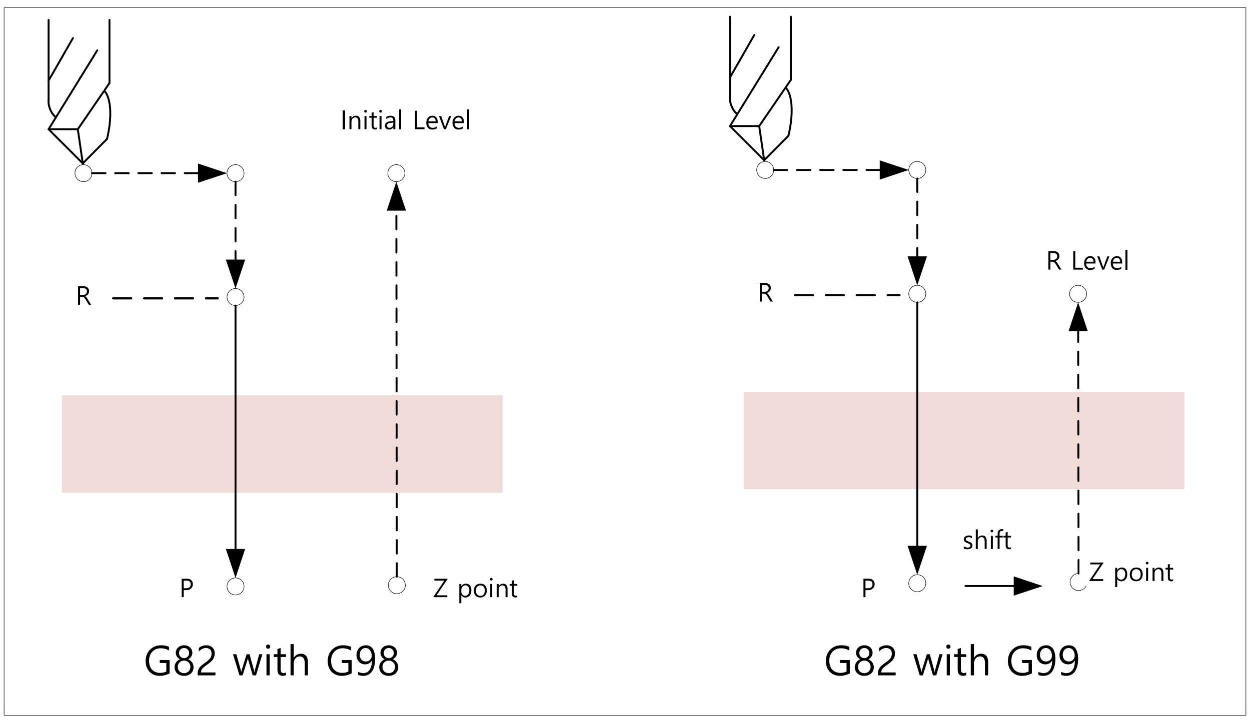
39) Drilling dwell cycle/Counter boring cycle

G82 [G90/G91] [G98/G99] X_ Y_ Z_ R_ P_ F_ K _

G81 : Drilling Cycle/Spot Drilling Cycle

X_ Y_ : Position of the hole
Z_ : Depth of the hole
R_ : Position of R point
P_ : Dwell time at the bottom of a hole
F_ : Cutting feed rate
K_ : Number of repeat

The Drilling Dwell Cycle / Counter Boring Cycle (G82) commands are used for general drilling. After cutting feed to the bottom of a hole, perform the set dwell. If dwell is omitted, it is same as G81.



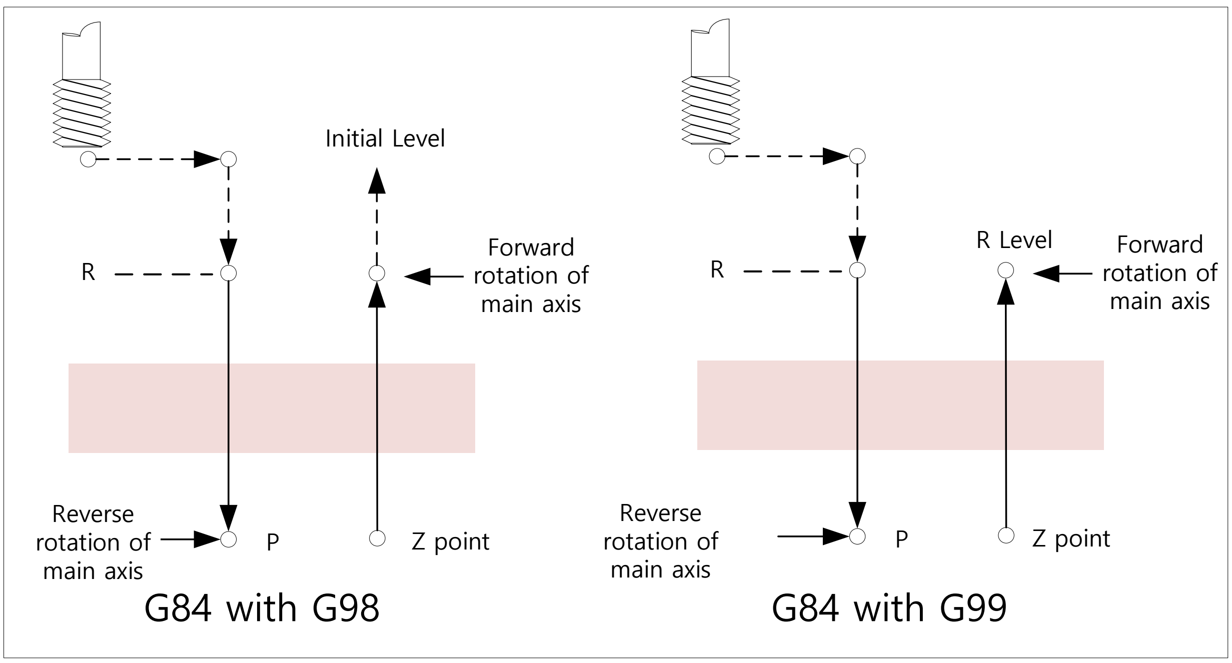
40) Tapping cycle(G84)

G84 [G90/G91] [G98/G99] X_ Y_ Z_ R_ P_ F_ K _

G84 : Tapping Cycle

X_ Y_ : Position of the hole
Z_ : Depth of the hole
R_ : Position of R point
F_ : Cutting feed rate
K_ : Number of repeats

The Tapping Cycle (G84) command is used for creating screws. This tapping cycle makes a screw by directly revolving (M3) the master axis with a right screw tap spindle and counterclockwise revolving (M4) at the bottom of a hole to escape.

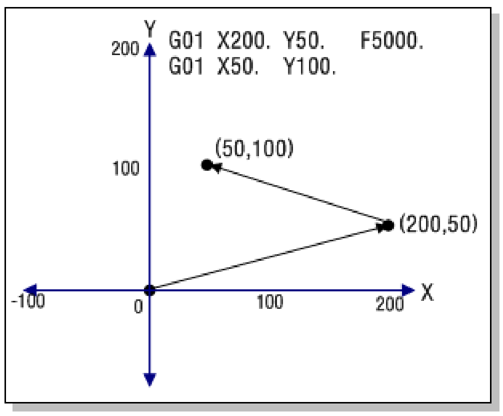


41) Absolute command(G90)

G90 G01 X_ Y_ Z_ A_ B_ C_ U_ V_ W_

G90: Absolute position command
G01/ G00: linear interpolation/Positioning
X_ Y_ Z_ A_ B_ C_ U_ V_ W_: feed target position

The absolute command (G90) is the method of commanding the feed position based on the currently set coordinate system.
The feed end point uses the value calculated from the origin of the currently specified coordinate system, regardless of where the current position is on the coordinate. The absolute command (G90) is a modal command, and once it is commanded, it still acts as a valid command unless another position command for the feed target is set.



G90	% absolute command
G01 X200 Y50 F5000	% Linear interpolation, target position to traverse(X=200, Y=50), speed 5000
X50 Y100	% Linear interpolation, target position to traverse(X=50, Y=100), speed 5000

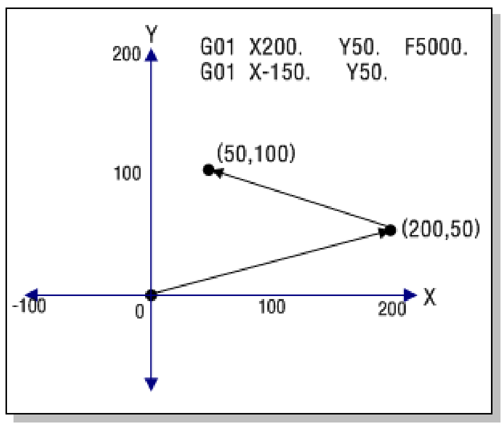
The above program represents the movement of the above figure with the G code.

42) Incremental command(G91)

G91 G01 X_ Y_ Z_ A_ B_ C_ U_ V_ W_

G91: Incremental command
G01/ G00: linear interpolation/Positioning
X_ Y_ Z_ A_ B_ C_ U_ V_ W_: feed target position

The incremental command (G91) is the method to instruct the movement amount to the target point to traverse for the current position based on the currently set coordinate system.
The incremental command (G91) is a modal command, and once it is commanded, it still acts as a valid command unless another position command for the feed target is set.



G91 %Incremental command	
G01 X200 Y50 F5000.	% Linear interpolation, target position to traverse(X=200, Y=50), speed 5000
X-150 Y50	% Linear interpolation, target position to traverse(X=50, Y=100), speed 5000

The above program represents the movement of the above figure with the G code.

43) Work-piece Coordinate System Setting, Maximum Spindle Speed (G92)

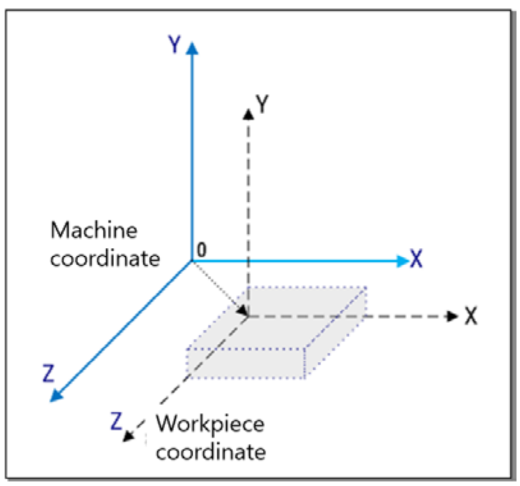
When setting the Work-piece Coordinate System

G92 X_ Y_ Z_ A_ B_ C_ U_ V_ W_

G92: Workpiece coordinate system command
X_ Y_ Z_ A_ B_ C_ U_ V_ W_: Input of offset value by axis

The work-piece coordinate system setting is a command to shift the coordinate system in current use as much as the input offset.
It is used when resetting the coordinate system based on the reference point of the workpiece.
It operates based on the coordinate system shifted (G54 ~G59) before selecting the work-piece coordinate system after the relevant command.

Note: It should be used after reference position return.



% Used after homing	
G90 X10 Y10	% workpiece coordinate system X=10 Y=10 move to position
G92 X100 Y100	% Shift work coordinate system X=10,Y10 to X=100,Y=100
G90 X10 Y10	% workpiece coordinate system -80, -80 move to position

When setting the maximum main axis speed

G92 S_

G92: The command to set the maximum spindle speed
S_: The maximum revolution speed of the spindle in rpm

Set the upper limit for input speed when controlling constant surface speed (G96). When controlling constant surface speed, if the input value of speed is greater than the command value of G92, the spindle revolves at the set speed of G92.

G92 S100	% When controlling constant surface speed, the maximum revolution speed of the spindle is designated at 100 rpm
G96 S1000	% When controlling constant surface speed, cutting speed is 1000 m/min

44) Feed mode command per minute (G94)

```
G94 G01 X_ F_
```

G94: Feed mode command per minute
G01: Linear interpolation feed command
X_: Coordinate value of the target position to move through the linear interpolation feed
F_: Speed command

It is the command to set the input unit to the user input unit (mm, degree) per minute.
Under the command, for the F input unit, the unit / min (mm / min, deg / min) is applied.

```
G94 G01 X10 F10      % If the unit is mm, the feed rate is commanded in 10mm / min.
```

45) Feed mode command per revolution (G95)

```
G95 G01 X_ F_
```

G95: Feed rate per revolution of the master axis
G01: Linear interpolation feed command
X_: Coordinate value of the target position to move through the linear interpolation feed
F_: Speed command

It is the command to set the input unit to the feed rate per revolution of the master axis.
Under the command, for the F input unit, the unit /rev(mm/rev) is applied.

```
G95 G01 X10 F10      % If the unit is mm, set the speed to 10mm / rev.
```

46) Constant surface speed control (G96)

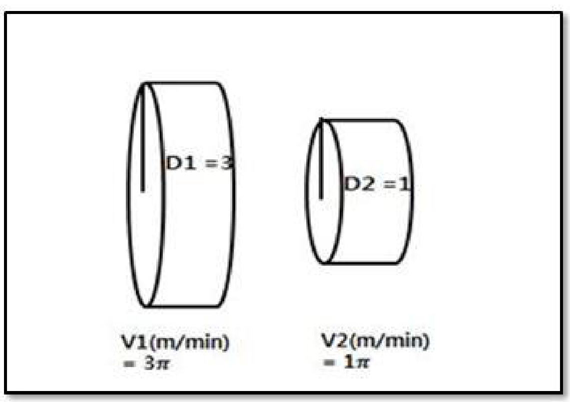
G96 S_

G96: Constant surface speed (= circumferential speed) control command
S_: Cutting speed command (m/min)

It is the function used when machining materials with an inconstant diameter.
Keep the cutting speed constant and revolve the spindle by calculating the rotation number of the spindle according to change in material diameters.
As the cutting amount is constantly kept when using the relevant command, improve uniformity of the surface roughness of a work-piece. The relevant setting is kept before G97 (Constant surface speed control cancel command) is commanded.

$$N = \frac{1000\,V(\text{속도})}{\pi D(\text{지름})}$$

* Note: As a diameter becomes 0 and the N value reaches infinity when machining the center of a material, if you want to execute G96, designate the maximum rotation number by setting G92 (the maximum spindle speed) to a code before executing G96 for safety.
If selecting G96 first, the speed of constant surface speed control, S, should be input. If you have selected G96 previously, you don't need to input it.



[Figure] Comparison of Radius and Speed of Work-piece

G96 S1000 % Command Cutting speed by 1000m/min

The parameter items related to constant surface speed control are as follows:

NC Parameter	Group and parameter name		
NC Channel Parameter	Spindle Setting	Reference Axis for Constant Speed Contr	0: None
		Max. Speed of Spindle at Constant Speed Control	0
		Min. Speed of Spindle at Constant Speed Control	0

47) Constant surface speed control disable(G97)

G97 S_

G96: Constant surface speed control cancel = Constant rotation number control
S_: Set the rotation speed of the spindle in rpm

The modal code operates before G96 (constant surface speed control) is commanded. The code is used when rotating the spindle at the constant speed.

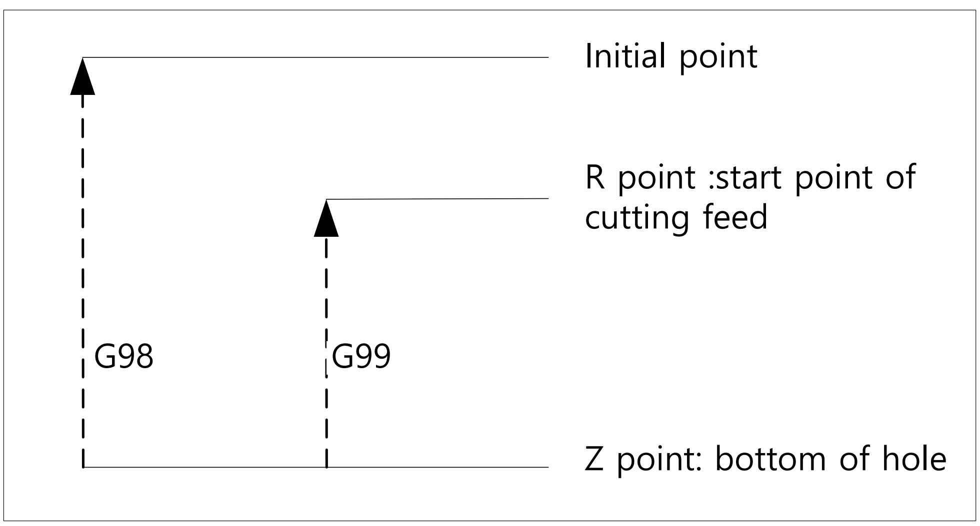
G97 S100 % Command Cutting speed by 100rpm

48) Return to Initial Point at a Canned Cycle (G98)

G98

G98 : Return to Initial Point at a Canned Cycle

Rapidly return to the initial point (the height of Z axis when the canned cycle mode setting is commanded) after completing boring.

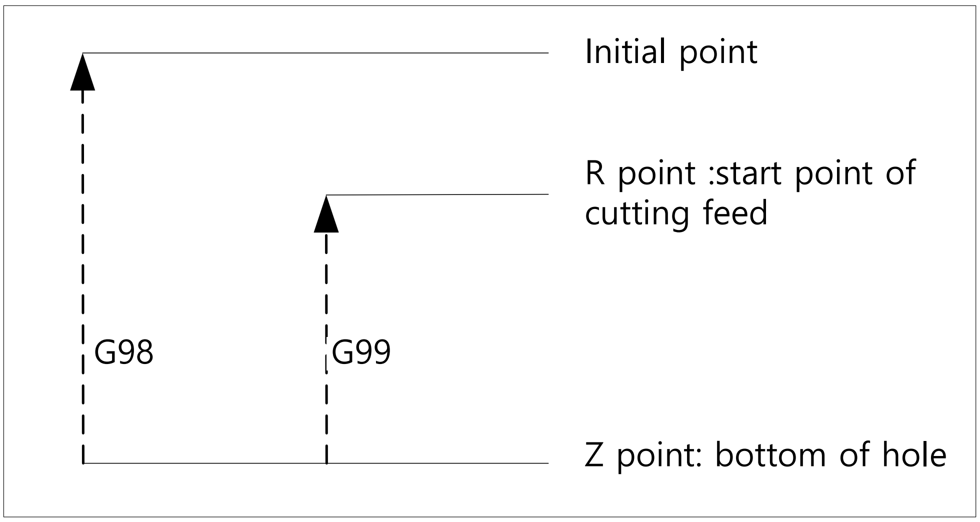


49) Return to R Point at a Canned Cycle (G99)

G99

G99 : Return to Initial Point, R point (initial point of cutting feed) at a Canned Cycle

Rapidly return to R point after completing boring.

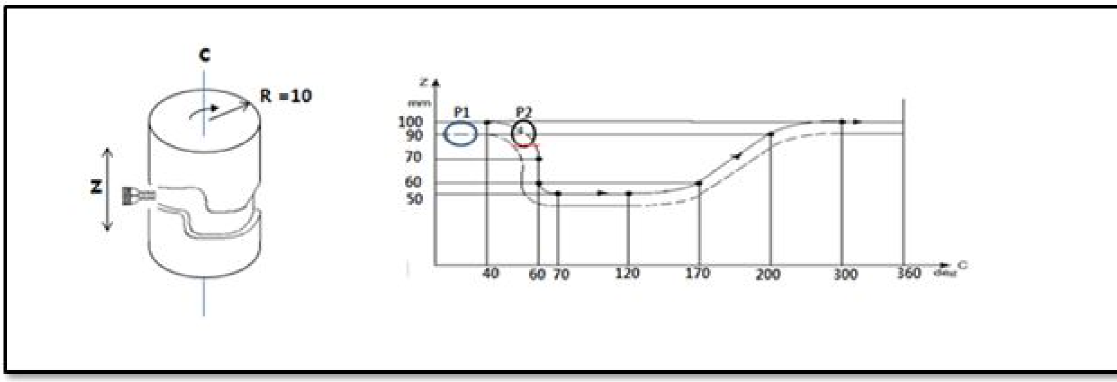


50) Cylindrical interpolation mode setting (G107)

G107 C_

G107: Cylindrical interpolation command
C_: Set the rotation axis and radius of a cylinder (If the value is 0, the cylindrical interpolation is canceled.)

The cylindrical interpolation is a type of contour control, which is the control mode for machining a cylindrical surface. It can be easily created when grooving the cylindrical CAM.
It performs the circular (G02, G03) and linear interpolation (G01) with other axes by converting the movement amount of the rotation axis specified by the angle into the linear axis distance of the circumference.




```
G107 C10
G90 G01 G18 Z0 C0     % Select the circular interpolation plane (ZX) Z0 C10 linear interpolation command

C40                   % P1: Linear interpolation
G03 Z70 C60 R3        % P2: Circular interpolation CCW direction (G03) feed position command (Z70 C60)
Circular arc's radius (3)

G107 C0               % Cancel the cylindrical interpolation
```

Caution
In the cylindrical interpolation mode, the circular arc radius command can be done with R only.
In the cylindrical interpolation mode, the positioning command (G00) is not available.
In the cylindrical interpolation mode, the coordinate system command is not available.

51) Polar coordinate interpolation mode ON/OFF (G112, G113)

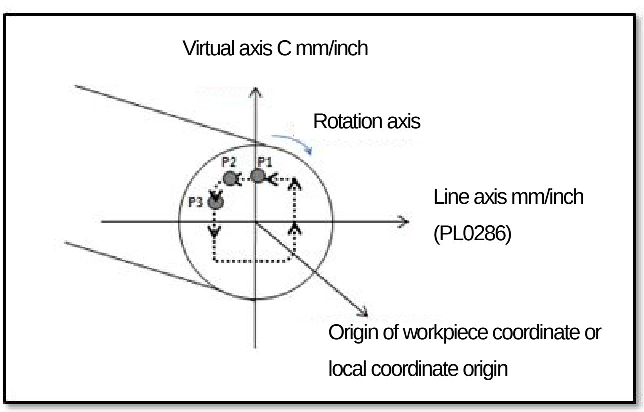
G112 G01/G02/G03 G113

G112: Set the polar coordinate interpolation mode, it is maintained until the G113 is commanded.
G01/G02/G03: Command to the rectangular coordinate system of the linear and rotary axis.
G113: Disable the polar coordinate interpolation mode

The polar coordinate interpolation converts the command entered into the rectangular coordinate system into the movement of the linear axes (X, Y, Z: tool movement) and the rotary axes (A, B, C: workpiece). This is executed on the polar coordinate interpolation plane created by the linear axis and the imaginary axis that is orthogonal to the linear axis.
The linear and the rotation axes should set in the parameters before the polar coordinate interpolation.

<table>
<tr><th>NC Parameter</th><th>Group and parameter name</th></tr>
<tr><td>NC Channel Parameter</td><td>
<table>
<tr><th>Group</th><th>Name</th><th>Channel 1</th></tr>
<tr><td rowspan="16">Basic Settings</td><td>Target Machining Quantity</td><td>0</td></tr>
<tr><td>Target machining quantity at M99 repetition</td><td>0</td></tr>
<tr><td>Check of decimal point</td><td>1: Unused</td></tr>
<tr><td>Keep workpiece coordinate system</td><td>0: Keep</td></tr>
<tr><td>Macro call on T-code command</td><td>0: Do not call</td></tr>
<tr><td>DWELL Method</td><td>0: Time</td></tr>
<tr><td>Block selection at NC reset</td><td>0: Keep the Current Block</td></tr>
<tr><td>Statement number search</td><td>0: Search</td></tr>
<tr><td>Minimum command unit</td><td>0 mm</td></tr>
<tr><td>Whether to use G22 [No traveling area]</td><td>0: Used</td></tr>
<tr><td>Inner/Outer side of G22 [No traveling area]</td><td>0: Inner side</td></tr>
<tr><td>Whether to use the 3rd [No traveling area]</td><td>0: Used</td></tr>
<tr><td>Rotary Axis of Cylindrical Interpolation</td><td>0: X Axis</td></tr>
<tr><td>Linear axis for interpolating the polar coordi</td><td>0: None</td></tr>
<tr><td>Rotary axis for interpolating the polar coordi</td><td>0: None</td></tr>
<tr><td>Monitoring time for in-position completion</td><td>5000 ms</td></tr>
</table>
</td></tr>
</table>

In this mode, the tool diameter can be compensated and the polar coordinate interpolation is performed for the compensation path of the tool diameter.
It is mainly used for grinding of the CAM shaft, etc.



```
G112                          % Polar coordinate interpolation mode On
G01 C10 F100                  % P1: C10 travel
G01 X-8                       % P2: X -8 position travel
G03 X-10. C8, R2,             % P3: X -10 Y 8

G113: Disable the polar coordinate interpolation mode
```

Caution

In the polar coordinate interpolation, only the straight line (G01) and circular interpolation (G02 / G03) can be used.

The command unit of a virtual axis is the same as a linear axis. The coordinate of the virtual axis becomes 0 under the G112 command.

The F command in the polar coordinate interpolation is the linear velocity (the relative velocity of the workpiece and the tool).

The circular interpolation imagines the X and Y planes so the distance from the starting point to the central point is commanded by I and J only.

(2) M code

Operating the machine through the motion control requires the functions for various mechanical operations in addition to the functions such as feed and interpolation using the G codes. In order to control the machine using the functions other than those supported by the G codes, the motion controller supports the M codes.
The M codes of the motion controller support the Pause, the function indicating the end of the NC program and the functions to interface with each motion control flag.
M codes can be commanded again only after all operations of the codes that have been already commanded are completed.

```
Mxx
```

Mxx: Auxiliary command

The machine sequence function corresponding to the "Mxx" code is activated.

```
G90
G01 X100. Y100,
G01 X150. Y200
M00        % Pause command
G01 X10. Y10,
G01 Y30. Z30,
M02
```

Actual operations may differ because the machine manufacturer determines which function of the machine is to be given to the M code. However, the general code table that is commonly used is as follows.

M code	Function	Meaning
M00	Program's operation is stopped. Program Stop	Automatic operation stops when M00 is commanded during the automatic operation. The modal information is valid up to the present like the single block stop, and the automatic operation is continued by pressing the cycle start button.
M01	Selective stop Optional Stop	This function that is the same as the M00 function is valid when the Optional Stop Switch is On. This command is ignored if the switch is not turned on.
M02	Program End End of Program	This is the command indicating the end of the program. After the operation of the block is completed, the master axis and Coolant stop. Then, the cursor returns to the beginning of the program. All commands are RESET by the same operation as M30.
M03	Forward rotation of the master axis [CW]	Forward rotation of the master axis Before this command, you must adjust the gear shift and the number of revolutions of the master axis in advance.
M04	Reverse rotation of the master axis [CCW]	Reverse rotation of the master axis Before this command, you must adjust the gear shift and the number of revolutions of the master axis in advance.

M code	Function	Meaning
M05	Master axis stop	Master axis stop It is used to change the direction of rotation or to shift gears.
M06	Tool change	Tool change Depending on the type of automatic tool changer (ATC), it is also used as the specific macro program call function.
M08	Coolant On	A coolant motor is operated. Before this command, the auto switch of the coolant on the machine’s control panel must be set to On. If the switch is off, the program will not proceed.
M09	Coolant Off	A coolant motor is stopped.
M30	Program End End of Tape	At the end of the program, the program returns to its beginning again. All commands are RESET.
M98	Auxiliary program call	The function to call an auxiliary program while the automatic program runs
M99	End of the auxiliary program	The function to terminate the auxiliary program. [Used even if the main program is executed repeatedly]

(3) Other Operation instructions of the NC program

Other instructions of the motion controller are the commands that control the progress of the program that is not supported by G code, M code, or logical / numerical operation function. Using the variables and instructions, it can program flexible and complicated forms of operations synchronized with the G / M code. These operation functions used in the NC program are similar to those used in the motion program but since they are operated directly in the NC program, it is possible to develop programs that can operate the machine more flexibly. In terms of the difference from the operation processed in the motion program, the motion program operates in the fixed cycle mode, while the operation of the NC program is executed through one flow only except that it is specified as the iteration statement.

The motion controller supports the variables available in the NC program as the macro variables. The macro variable replaces the part where the variable is used with its own value. By using the macro variable, you can give flexibility to the machine control through a controller.

1) Variable (#)

1-1) Local variable

#Ni (N = X, D, W, L; i= 1, 2, 3, …) #N[Expression]

For the variables, # followed by a variable type and a number. Multiple variables are separated by a number after the #. The constraints on using variables are as follows.

It is possible to use the variable instead of the value following the address.
ex) F[#L103] → F100 (when, #L103 = 100)
Z-[#L110]→ Z-250 (when, #L110 = 250).

When the variable number is used as the variable
ex) #L100 = 105
#L105 = -500
##L100 ; wrong expression
#[#L100] ; Expressed using brackets.

A value exceeding the maximum command value set for each address cannot be specified.
ex) #L140 = 1000
G[#L140] ; Maximum command value OVER.

The value of the variable that is not yet defined is 0

1-2) Global variable

#MNi (N = X, D, W, L; i= 1, 2, 3, …) #MN[Expression]

For the global variables, # followed by a device type, M with a variable type and a number. The range of input values available for each variable type is different. The constraints on using variables are the same as the local variable.

1-3) System variable

#FNi (N = X, D, W, L; i= 1, 2, 3, …)

For the system variables, # followed by a device type, M with a variable type and a number. The range of input values available for each variable type is different. The constraints on using variables are as follows.

- For the system variable, read only.

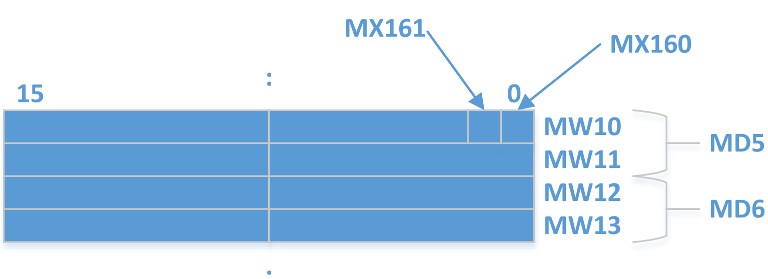
2) Usage according to the range of variables

The range of the variable is limited by the variable's number. It is divided into the local variable, global variable, and system variable according to the range, and the details are as follows.

2-1) Size of each variable

Type	Size	Note
Local variable	4KB	Size per program (10 program)
Global variable(M)	2MB	
System flag(F)	128KB	

2-2) Addressing method according to each data type



Data type	Addressing method
Bit(X)	Addressing the leaner bitwise value from bit 0 after the 'X' indicator
16 bit(W)	Addressing the value in word units (16bit) after the 'W' indicator
32 bit(D)	Addressing the value in double word units(32bit) after the 'D' indicator
64 bit(L)	Addressing the value in long word units(64bit) after the 'L' indicator

2-3) Performing the address range check for each data type

Type	Local variable	Global variable	System flag
Bit	#X0 ~ #X32767	#MX0 ~ #MX16777215	#FX0 ~ #FX1048575
16 bit	#W0 ~ #W2047	#MW0 ~ #MW1048575	#FW0 ~ #FW65535
32 bit	#D0 ~ #D1024	#MD0 ~ #MD524287	#FD0 ~ #FD32767
64 bit	#L0 ~ #L511	#ML0 ~ #ML262143	#FL0 ~ #FL16383

3) Specify the Statement Number(N)

```
N_
```

N_: Specify the Statement Number

"N" for specifying the Statement Number is the command used in instructions such as IF, GOTO, etc., and displays the corresponding block so that other commands can recognize it. Since the NC program of the motion controller does not memorize the block's line number of each program separately, the "N" command is used for the block to be displayed.

The "N" command can be used with other commands in the block or it can be used alone in one block.
However, the number used for the "N" command must be unique for each program's motion file (.nc file). If there is a duplicated Statement Number, an Error will occur.
The Statement Numbers that can be specified with the "N" command range from 0 to 2147483647. If you use a number exceeding this range, an error is generated. In addition, the number of Statement Numbers that can be specified in one motion program is up to 1000. If the total number of specified Statement Numbers exceeds 1000, an error occurs.
The statement numbers ordered commanded with the "N" do not have to be used in numerical order on the program. That is, the Statement Numbers can be used in random order.

4) Conditional statements(IF)

```
IF [……] GOTO N_
```

IF: IF statements
[……]: IF statements condition
GOTO N_: Specify the Statement Number to be branched

When the condition following"IF"is met, the conditional statement branches to the block with the Statement Number "N_" specified immediately after it. If the condition is not met, it starts sequentially from the block immediately below.
For the conditions following "IF"in the conditional statement, you can create them by comparing local variables with constants, comparing between local variables, and comparing between constants. In the condition after the "IF" in the conditional statement, variables and constants used in the condition can be applied up to 2 including variables and constants.
It cannot branch to another motion file (.nc file) or another motion program.

5) Branch Instruction(GOTO)

```
GOTO N_
```

GOTO: Unconditional branch to the block designated as "N_"
N_: Specify the Statement Number to Jump

The branch instruction is the function to branch unconditionally to the block marked with the Statement Number "N_" that is specified after "GOTO".

It cannot branch to another motion file (.nc file) or to another motion program.

6) Repeat statement(DO, WHILE)

```
WHILE [<Conditional expression>] DO n
(n = 1, 2, 3, …)
~
END n
```

WHILE: Condition repeat statement
DO n: repeat until n declaration
[……]: Conditional statement
END n: End of block to repeat

When the <conditional expression> is met, it repeats from the block following DO n to the END n block.When the <conditional expression> is not met, it jump to the next block of END. WHILE [<conditional expression>] can be omitted and If omitted, it repeats infinitely from DO n to END n.

WHILE [<conditional expression>] Do n and END n are always used as a pair, and by the identification number n, the pair is identified. If another loop is selected in the Iteration loop, it is distinguished by the pair of identification factors.

7) Operation command

There are substitution of variables and integers, the four fundamental arithmetic operations, Mathematical operation, etc. for available operations. The types of commands are shown in the table below. When using multiple operations in combination, the priority is given in order of variable, multiplication / division, addition / subtraction. The brackets "[]" are used to set the priority.

Classification	Expression	Note
Substitutution, replacement	#Li = #Lj	
Add	#Li = #Lj + #Lk	
Subtract	#Li = #Lj - #Lk	
OR	#Li = #Lj OR #Lk	
XOR	#Li = #Lj XOR #Lk	
Multiply	#Li = #Lj * #Lk	
Divide	#Li = #Lj / #Lk	
AND	#Li = #Lj AND #Lk	
Sin	#Li = SIN[#Lj]	
Cos	#Li = COS[#Lj]	
Tan	#Li = TAN[#Lj]	
Atan	#Li = ATAN[#Lj]	
Sqrt	#Li = SQRT[#Lj]	
Abs	#Li = ABS[#Lj]	
Round	#Li = ROUND[#Lj]	Rounding off to the nearest integer
And	#Li = AND[#Lj]	
Or	#Li = OR[#Lj]	
Fix	#Li = FIX[#Lj]	Decimal point abandon operation
Fup	#Li = FUP[#Lj]	Decimal point round-up operation

8) annotation(;, %, ())

```
%, ;, ()
```

% : annotation
; : annotation
( ) : annotation
Command description
All characters, expressions and numbers following "%" or ";" are commented out and ignored during the program execution.
"%" is valid in one line only.
The statement between the parentheses "(" and ")" is also treated as a comment.

```
G90
G00 X100. Y100, Z100, U100,        % From here, the whole line is commented out.
M02
```

9) Optional Block Skip

This is used as the command of / at the head of a block. It selects whether to perform the current block through an external signal. The command should be located in the forefront of the block and can be separated by using a number at the end of the block. For example, you make it with 2 and can set all the blocks set by 2 not to execute later / with the NC_BlockOptionalSkip function block.

/Numbers from 1 to 9 can be set later. In other words, you can set with /1 ~ /9. / means /1. The block that is set not to execute remains until skip is released by setting SkipNum to 0 in the NC_BlockOptionalSkip command.

10) Example of program operation instruction usage

```
G90

% % symbol comment description
; Comment description after a colon

#L100=1                          % Substitute a constant 1 in the local variable # L100
#L102=3                          % Substitute a constant 3 in the local variable # L102

IF [#L100 EQ 1] GOTO N3          % Comparison of conditions using local variables in IF STATEMENT
#L101 = #L100 + #L102            % Numerical operation using the local variables
N3                               % Specify the Statement Number
G02 X100. Y100, I50, J50,
G01 X[#L102]                     % Same operation as G01 X3 (since 3 is substituted in # L102)

N150 WHILE [#L100 LE [360.-#L102]] DO 210     % Repeat up to N210 until the condition is met

N200 WHILE [#L101 GE 10.] DO 220              % Repeat up to N220 until the condition is met

G0 Z10.
#L101 = #L101 + 10. (INCREASE)   % calculation formula

END 220                  % Iteration end for the DO 220 statement

#L100 = #L100 + #L102
#L101 = 50.

END 210                          % Iteration end for the DO 210 statement

M02
```

9.4 NC Parameter

NC parameter is channel parameter and axis parameter.
The each parameter is as follows.

Parameter	Group	No.	Item	Description
1. Channel Parameter	1. Standard settings	1	Target machining quantity	Set the target machining quantity. (0 ~ 2,147,483,647) It is possible to write NC parameters during NC channel automatic operation.
		2	Target machining quantity at M99 repeated machining	Set the target machining quantity for repeated machining with M99. If the set value matches the current machining quantity, the cycle automatically stops. (0 ~ 2,147,483,647) It is possible to write NC parameters during NC channel automatic operation.
		3	Check of decimal point	Set whether to check decimal point of the NC program. 0: Decimal point check (Mm if there is a decimal point, um if there is no decimal point) 1: No decimal point check (mm)
		4	Keep workpiece coordinate system	Set whether to keep the workpiece coordinate system when resetting. 0:Hold, 1: Do not hold
		5	Whether to call the macro when the T code is commanded	Set whether to call the macro program (9000.nc ~ 9009.nc) when the T code is commanded. 0: Do not call 1: Call
		6	Dwell Method	Set the dwell function (G04) to use the data corresponding to X, P as time or the number of revolutions of the spindle. If the data is set to the number of revolutions of the spindle, it is applied in the status of feed per revolution (G95). 0: times 1: Rotation number
		7	Select a progress block at reset	Set whether to initialize to the start block of the program at reset. If you want to set to 0 (keep the current block), the parameters of "Keep workpiece coordinate system" should be set to 0 (keep). 0: Keep the current block 1: Initialize to the start block of the main program 2: Initialize to the current block of the main program It is possible to write NC parameters during NC channel automatic operation.

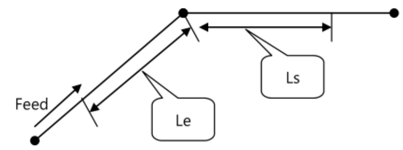
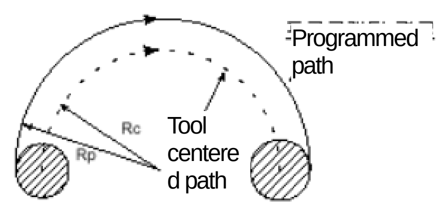
Parameter	Group	No.	Item	Description
1. Channel Parameter	1. Standard settings	8	Whether or not to search the Statement Number	The number of buffers that can store the program's Statement Number (N__) is limited to 1,000 in the system. This buffer is needed if the program changes the sequence using a GOTO statement. If more than 1,000 blocks have the N__ command, an alarm will occur. This parameter is used to input whether or not to execute such Statement Number search. Because high- capacity CAM programs do not have GOTO using the Statement Number and in the majority of cases, there are more than 1,000 Statement Numbers, you should set this parameter as 1. 0: Search 1: Do not search
		12	Minimum command unit	When decimal point check is applied, set the minimum unit of the commanded value. (0 ~ 0.999mm)
		18	Whether to use G22 No Travelling Area	0: 'No Travelling Area' is valid. 1: 'No Travelling Area' is invalid. It is possible to write NC parameters during NC channel automatic operation.
		19	Set the inner/outer side of G22 No Travelling Area	0: Inside 1: Outside
		20	Whether to use the 3rd 'No Travelling Area'	0: 'No Travelling Area' is valid. 1: 'No Travelling Area' is invalid. It is possible to write NC parameters during NC channel automatic operation.
		21	Reverse buffer operation size	Set the number of blocks that can be reversed during reverse operation. 0: Disable(Use of high-speed processing) 1~50: block number It is possible to write NC parameters during NC channel automatic operation.
		22	Rotary axis of Cylindrical interpolation	In the cylindrical interpolation mode, the axis maps the axis of rotation during the circular interpolation. The axes are X, Y, Z and perform the circular interpolation by mapping the axis of rotation to the selected axis. For example, if the axis of rotation is mapped to the X axis under the state of the XY plane (G17), the width becomes the axis of rotation and the height becomes Y axis. When ZX (G18) is selected as the plane, the width becomes the Z axis and the height becomes the axis of rotation. However, if you set the plane to YZ (G19), you cannot perform the circular interpolation on the commanded axis of rotation. 0: X axis,

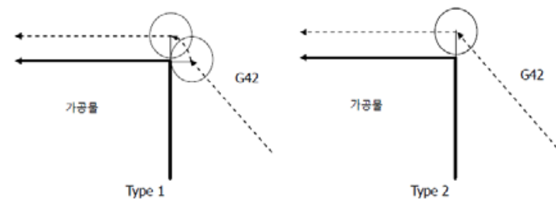
Parameter	Group	No.	Item	Description
				1: Y axis, 2: Z axis
		23	Linear axis for interpolating the polar coordinate	0: Unused 1: X, 2: Y, 3: Z, 4: A, 5: B, 6: C, 7: U, 8: V, 9: W
		24	Rotary axis for interpolating the polar coordinate	0: Unused 1: X, 2: Y, 3: Z, 4: A, 5: B, 6: C, 7: U, 8: V, 9: W
		25	Main spindle axis number	Set the number of an axis to be used as the main spindle axis in the NC channel. Set the system that does not use the spindle axis to 0. To automatically execute spindle commands in the NC function module, set it exactly the same as the axis number connected to the NC S axis. 0: Unused 1 ~ 36: Axis 1 ~ Axis 36
		33	Monitoring time for in-position completion	0 ~ 65,535ms It is possible to write NC parameters during NC channel automatic operation.
		34	Spindle operation of the spindle axis Methods to handle M/S-codes	If automatically controlling spindle commands in the NC function module, set the methods to handle M/S-codes related to the spindle operation. When executing M/S-code commands, conduct automatic control of the spindle axis without M/S-code operation completion commands. To execute the next block, select '0: Continue automatic operation'. If a user wants to automatically control the spindle axis after completing to use M/S-code operation completion commands, select '1: Operation continues after completing commands'. 0: Automatic operation continues 1: Operation continues after completing commands It is possible to write NC parameters during NC channel automatic operation.

Parameter	Group	No.	Item	Description
1. Channel Parameter	2. Circular milling setting	1	Regenerate the circular center when the circular alarm occurs	Set whether to recreate the central point of the arc without generating an arc alarm when the distance between the start point and the end point exceeds the tolerance of the difference between the two radii under the I, J, K circular commands. 0: An alarm occurs. 1: The central point of the arc is regenerated.
		2	Speed-limiting function for the circular milling ON/OFF	0: Unused 1: used It is possible to write NC parameters during NC channel automatic operation.
		3	Tolerance of arc radius	Set the tolerance of the difference between the two radii at the start point and the end point under the circular arc command. If this value is large, the accuracy of the end part of the arc may be degraded. When set to 0, it is recognized as 0.001. (0~ 1 unit, real number) It is possible to write NC parameters during NC channel automatic operation.
		5	Circular radius with the speed-limiting function for the arc machining	(0 ~ 10,000 unit, real number) It is possible to write NC parameters during NC channel automatic operation.
		6	Upper cutting speed limit of the circular milling	The maximum speed is limited to the set value for the circular arc below "Circular radius with the speed-limiting function for the circular milling " . It is possible to write NC parameters during NC channel automatic operation. (0 ~ 10,000 unit/min, real number)
		7	Lower cutting speed limit of the circular milling	If “Speed-limiting function for the circular milling ON/OFF” is set to ON, the cutting speed is limited to the set value or more. It is possible to write NC parameters during NC channel automatic operation. (0 ~ 10,000 unit/min, real number)
		9	Circular milling acceleration	Set the acceleration at the circular milling. It is possible to write NC parameters during NC channel automatic operation.
		10	Circular milling deceleration	Set the deceleration at the circular milling. It is possible to write NC parameters during NC channel automatic operation.
		11	Circular milling jerk	Set the jerk at the circular milling. It is possible to write NC parameters during NC channel automatic operation.

Parameter	Group	No.	Item	Description
1. Channel Parameter	3. Cutting feed setting	1	Set the upper speed limit of the cutting feed	If the cutting speed exceeding the set value is commanded, the cutting speed is limited to the set value and an alarm occurs. (0 ~ 100,000 unit/min, real number) It is possible to write NC parameters during NC channel automatic operation.
		2	Set the lower speed limit of the cutting feed	It is applied only when the cutting speed is not commanded in the feed mode per minute. (0 ~ 100,000 unit/min, real number) It is possible to write NC parameters during NC channel automatic operation.
		4	How to Accelerate/Decelerate Interpolation Operation	1: Acceleration / deceleration before interpolation
		5	Corner speed limit function mode setting	0: Unused 1: Angle difference mode 2: Speed difference mode It is possible to write NC parameters during NC channel automatic operation.
		6	Acceleration and deceleration type of the cutting feed	0: Line type 1: S type
		7	Operating method of the continuous blocks for acceleration / deceleration before interpolation	When executing the consecutive blocks, it creates the connecting trajectory that draws an arc on the corner of the connecting trajectory with the speed set with the next block. 1: When it is set to Buffered, the circular arc is not inserted. 1: Buffered 2: Blending Low 3: Blending Previous 4: Blending Next 5: Blending High
		9	Acceleration at the time of cutting feed (before interpolation)	Acceleration at the time of cutting feed It is possible to write NC parameters during NC channel automatic operation.
		10	Deceleration at the time of cutting feed (before interpolation)	Deceleration at the time of cutting feed It is possible to write NC parameters during NC channel automatic operation.
		11	Jerk at the time of cutting feed (before interpolation)	Jerk at the time of cutting feed It is possible to write NC parameters during NC channel automatic operation.
		18	Allowable angle setting (angle mode)	Set the allowable angle in angle mode. It is possible to write NC parameters during NC channel automatic operation.
		19	Allowable speed difference setting (speed difference mode)	Set the allowable speed difference in speed difference mode. It is possible to write NC parameters during NC channel automatic operation.
		20	Deceleration speed setting	Setting the Acc. speed.

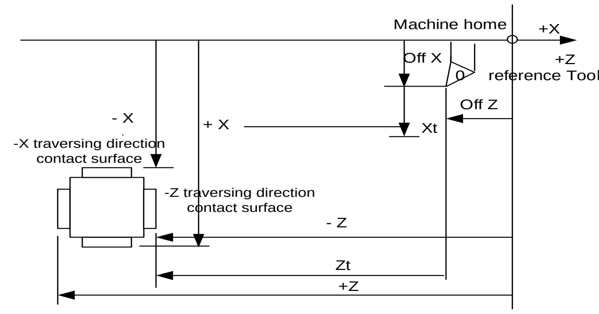
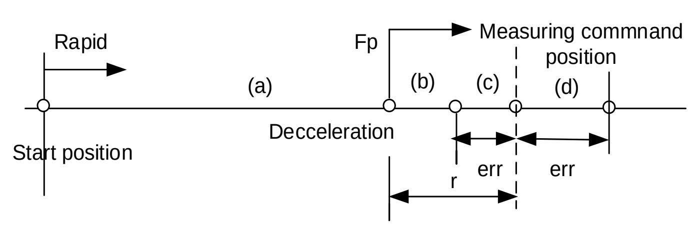
Parameter	Group	No.	Item	Description
				It is possible to write NC parameters during NC channel automatic operation.
		21	Speed multiplier with respect to the linear axis of the axis of rotation(speed difference mode)	Set the speed multiplier for the linear axis of the rotary axis in speed difference mode. It is possible to write NC parameters during NC channel automatic operation.
		22	Cutting feed acceleration and deceleration time constant (acceleration and deceleration after interpolation)	Set the time constant for acceleration/deceleration after interpolation in cutting feed.
	4. Rapid feed setting	1	Feed rate of rapid traverse (G00) block in dry run mode	In dry run mode, set the feed rate of the G00 block to the manual feed rate or the rapid feed rate. 0: Dry run speed 1: Rapid feed rate

Parameter	Group	No.	Item	Description
1. Channel Parameter	7. Automatic corner override	1	Start section of automatic corner override setting	Sets the override area (Ls) of the starting block that forms the corner when an automatic corner override command is commanded.  (0 ~ 999.999 mm, 0 ~ 99.999 inch, real number)
		2	End section of automatic corner override setting	Sets the override area (Le) of the ending block that forms the corner when an automatic corner override command is commanded. (0 ~ 999.999 mm, 0 ~ 99.999 inch, real number)
		3	The scale of automatic corner override settings	Set the multiplier applied to the feed rate in the automatic corner override section. ※ Auto Corner Override Feed rate = Current Command Feed rate x Multiplier x 0.01 (10~100%, real number)
		5	The angle between the automatic corner override settings	Sets the maximum angle of a corner that determines which blocks apply automatic corner override. (-999.999 ~ 999.999 deg, real number)
		6	The type of automatic corner override settings	Sets the application type of automatic corner override. If you set Always Apply, the automatic corner override function is applied by judging the angle of the corner in every block. 0: Applies only to inner corners for tool radius compensation 1: always applied
		7	Minimum rate of internal circular cutting speed	Set the minimum rate of internal circular cutting speed. The internal circular cutting speed is determined by Rc/Rp * F during circular arc feed during compensation. If Rc/Rp is less than or equal to the set value, it is limited to the set value.  (0.01~1.00%, real number)
		8	Automatic corner override velocity	This is an item to directly input the speed of the deceleration section of automatic corner override. If the value is 0, the value obtained by multiplying the current cutting speed (Feed) by the multiplier of the automatic corner override setting is applied as an override. If the command speed is smaller than this value, it is fed at the command speed. (0 ~ 99999.999 mm/min, 0 ~ 9999.999 inch/min, real number)

Parameter	Group	No.	Item	Description
1. Channel Parameter	8. Tool radius Compensation	129	How to apply the compensation value of the tool diameter	Set the method of applying the compensation amount of the tool diameter when compensating the tool diameter. 0: Apply the diameter value 1: Apply the radius value
		130	Compensation type of the tool diameter	Tool diameter Sets the type of traversing method at the beginning and end of the calibration.  0: Type 1(Bypass traverse) 1: Type 2(Direct traverse)
		131	Whether to check the tool interference during tool diameter compensation	Set whether to check the tool interference during tool diameter compensation 0: Do not check 1: Check
		1	Compensation amount of the tool diameter 1	Compensation amount 1 to be used to compensate the tool diameter
		……	……	……
		128	Compensation amount of the tool diameter 128	Compensation amount 128 to be used to compensate the tool diameter
	9. Tool length Compensation	1	Compensation amount 1 of the tool length	Compensation amount 1 to be used to compensate the tool length
		……	……	……
		128	Compensation amount 128 of the tool length	Compensation amount 128 to be used to compensate the tool length

Parameter	Group	No.	Item	Description
1. Channel Parameter	10. Work-piece Coordinate System	1	Whether to use the workpiece coordinate system shift amount.	Set whether to use the workpiece coordinate system shift amount. 0: Unused 0: Used
		11	Workpiece coordinate system shift amount 1	Set the workpiece coordinate system shift amount for the X axis.
		……	……	Set the workpiece coordinate system shift amount for the 7 axes; Y, Z, A, B, C, U, V.
		19	Workpiece coordinate system shift amount 9	Set the workpiece coordinate system shift amount for the W axis.
		41	G54 workpiece coordinate system value 1	Set the G54 workpiece coordinate system value for the X axis.
		……	……	Set the G54 workpiece coordinate system values for the 7 axes; Y, Z, A, B, C, U, V.
		49	G54 workpiece coordinate system value 9	Set the G54 workpiece coordinate system value for the W axis.
		51	G55 workpiece coordinate system value 1	Set the G55 workpiece coordinate system value for the X axis.
		……	……	Set the G55 workpiece coordinate system values for the 7 axes; Y, Z, A, B, C, U, V.
		59	G55 workpiece coordinate system value 9	Set the G55 workpiece coordinate system value for the W axis.
		61	G56 workpiece coordinate system value 1	Set the G56 workpiece coordinate system value for the X axis.
		……	……	Set the G56 workpiece coordinate system values for the 7 axes; Y, Z, A, B, C, U, V.
		69	G56 workpiece coordinate system value 9	Set the G56 workpiece coordinate system value for the W axis.
		71	G57 workpiece coordinate system value 1	Set the G57 workpiece coordinate system value for the X axis.
		……	……	Set the G57 workpiece coordinate system values for the 7 axes; Y, Z, A, B, C, U, V.
		79	G57 workpiece coordinate system value 9	Set the G57 workpiece coordinate system value for the W axis.
		81	G58 workpiece coordinate system value 1	Set the G58 workpiece coordinate system value for the X axis.
		……	……	Set the G58 workpiece coordinate system values for the 7 axes; Y, Z, A, B, C, U, V.
		89	G58 workpiece coordinate system value 9	Set the G58 workpiece coordinate system value for the W axis.
		91	G59 workpiece coordinate system value 1	Set the G59 workpiece coordinate system value for the X axis.
		……	……	Set the G59 workpiece coordinate system values for the 7 axes; Y, Z, A, B, C, U, V.
		99	G59 workpiece coordinate system value 9	Set the G59 workpiece coordinate system value for the W axis.

Parameter	Group	No.	Item	Description
1. Channel Parameter	11. Macro Program	1	Whether to apply the single block stop function to the macro program	Set whether to apply the single block stop function to the macro program(9000.nc ~ 9999.nc) 0: Stop 1: Do not stop It is possible to write NC parameters during NC channel automatic operation.
		2	Display the macro program block	Set whether to display the progress status of the block on the screen when operating the macro program (9000.nc ~ 9999.nc). 0: Not display 1: Display It is possible to write NC parameters during NC channel automatic operation.
		10	Macro program call G code (9010.nc)	Set the G code number to call the macro program (9010.nc ~ 9019.nc) that can be called by the G code. The setting values 0, 1, 2, 3 are ignored. (0~255.9, real number)
		……	……	
		19	Macro program call G code (9019.nc)	Set the G code number to call the macro program (9010.nc ~ 9019.nc) that can be called by the G code. ※ The setting values 0, 1, 2, 3 are ignored. (0~255.9, real number)
		20	Macro program call M code (9020.nc)	Assign the M code number to call the macro program (9020.nc ~ (9020.nc ~ 9029.nc) with the M code. ※ 0, 30 of the input values are ignored. (0~255, integer) ※ It can be used only in the main program, and when used in the subprogram, it operates with the general M code.
		……	……	
		29	Macro program call M code (9029.nc)	Assign the M code number to call the macro program (9020.nc ~ (9020.nc ~ 9029.nc) with the M code. 0, 30 of the input values are ignored. (0~255, integer) ※ It can be used only in the main program, and when used in the subprogram, it operates with the general M code.
		9	T code call macro program number	Enter the number of the macro program (9000.nc ~ 9009.nc) to be called when the T code is commanded. (9000~9009, integer) It is possible to write NC parameters during NC channel automatic operation.

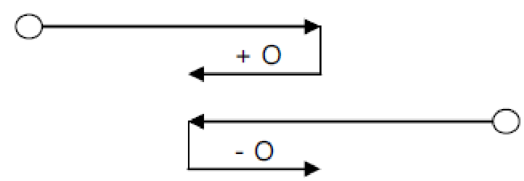
Parameter	Group	No.	Item	Description
1. Channel Parameter	13. Automatic tool offsets	1	+Measurement reference distance X for automatic tool offset	Set the measurement reference distance from the automatic tool offset to the contact surface in the +traversing direction of X.  (-99999.999 ~ 99999.999 mm, -9999.999 ~ 9999.999 inch, real number)
		2	+Measurement reference distance Y for automatic tool offset	Set the measurement reference distance from the automatic tool offset to the contact surface in the +traversing direction of Y. (-99999.999 ~ 99999.999 mm, -9999.999 ~ 9999.999 inch, real number)
		3	+Measurement reference distance Z for automatic tool offset	Set the measurement reference distance from the automatic tool offset to the contact surface in the +traversing direction of Z. (-99999.999 ~ 99999.999 mm, -9999.999 ~ 9999.999 inch, real number)
		4	- Measurement reference distance X for automatic tool offset	Set the measurement reference distance from the automatic tool offset to the contact surface in the - traversing direction of X. (-99999.999 ~ 99999.999 mm, -9999.999 ~ 9999.999 inch, real number)
		5	Measurement reference distance Y for automatic tool offset	Set the measurement reference distance from the automatic tool offset to the contact surface in the -traversing direction of Y. (-99999.999 ~ 99999.999 mm, -9999.999 ~ 9999.999 inch, real number)
		6	- Measurement reference distance Z for automatic tool offset	Set the measurement reference distance from the automatic tool offset to the contact surface in the -traversing direction of Z. (-99999.999 ~ 99999.999 mm, -9999.999 ~ 9999.999 inch, real number)
		7	Automatic tool deceleration start length of automatic tool offset	Sets the application section (r) of deceleration speed for automatic tool length measurement.  (0 ~ 99.999 mm, 0 ~ 9.999 inch, real number)
		8	Automatic tool measurement detection range of automatic tool offset	Sets the detection range (err) for automatic tool length measurement. (0 ~ 99.999 mm, 0 ~ 9.999 inch, real number)

Parameter	Group	No.	Item	Description
		9	Automatic tool deceleration velocity of automatic tool offset	Set the automatic tool length measurement deceleration speed (Fp) (0.001 ~ 999.999 mm, 0.0001 ~ 99.9999 inch, real number)

Parameter	Group	No.	Item	Description
1. Channel Parameter	14. Default Setting	1	Modal traverse of default settings	If there is no G00 or G01, select the G code to be applied as the default modal. 0: Rapid Feed(G00) 1: Cutting Feed(G01)
		2	Modal plane of default settings	If there is no G code instruction for G17, G18, G19 group, select the G code to be applied as the default modal. 0: XY plane(G17) 1: XZ plane(G18) 2: YZ plane(G19)
		3	Modal absolute / increment with default settings	If there is no G code instruction for G90, G91 group, select the G code to be applied as the default modal. 0: Absolute command (G90) 1: Incremental command(G91)
		4	Modal inch / metric with default settings	0: G20 1: G21
		5	Check the modal prohibited area with default settings	If there is no G code instruction for G22, G23 group, select the G code to be applied as the default modal. 0: Stroke On(G22) 1: Stroke off(G23)
	15. Spindle setting	4	A reference axis when controlling constant surface speed	Set the reference axis that operates in connection with a spindle when controlling constant surface speed. 0: Unused 1: X, 2: Y, 3: Z, 4: A, 5: B, 6: C, 7: U, 8: V, 9: W It is possible to write NC parameters during NC channel automatic operation.
		5	The maximum number of spindle rotation when controlling constant surface speed	Set the maximum number of spindle rotation when controlling constant surface speed. When being commanded by the S code of G92 (set the maximum speed of the master axis), the S code data is saved as this parameter value. (0 ~ 100,000, real number) It is possible to write NC parameters during NC channel automatic operation.
		6	The minimum number of spindle rotation when controlling constant surface speed	Set the minimum number of spindle rotation when controlling constant surface speed. (0 ~ 100,000, real number) It is possible to write NC parameters during NC channel automatic operation.
	16. Relative Setting	1	Relative coordinate’s offset value #1	Set the relative coordinate’s offset value for the X axis.
		2	Relative coordinate’s offset value #2	Set the relative coordinate’s offset value for the Y axis.
		3	Relative coordinate’s offset value #3	Set the relative coordinate’s offset value for the Z axis.
		4	Relative coordinate’s offset value	Set the relative coordinate’s offset value for the A

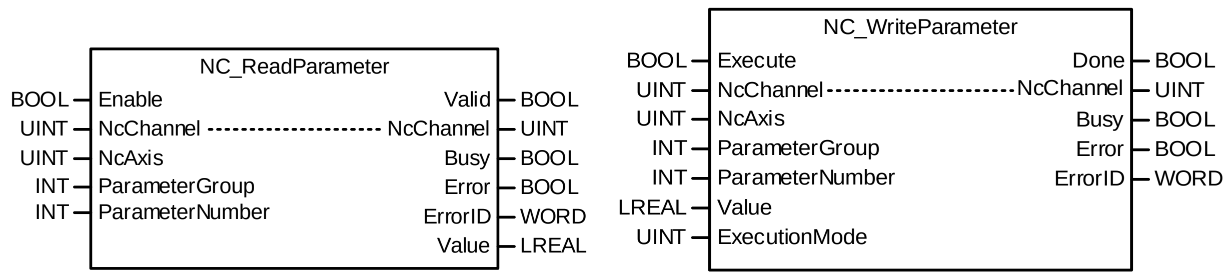
Parameter	Group	No.	Item	Description
			#4	axis.
		5	Relative coordinate's offset value #5	Set the relative coordinate's offset value for the B axis.
		6	Relative coordinate's offset value #6	Set the relative coordinate's offset value for the C axis.
		7	Relative coordinate's offset value #7	Set the relative coordinate's offset value for the U axis.
		8	Relative coordinate's offset value #8	Set the relative coordinate's offset value for the V axis.
		9	Relative coordinate's offset value #9	Set the relative coordinate's offset value for the W axis.

Parameter	Group	No.	Item	Description
2. Channel/Axis Parameter	1. Axis setting	2	Setting the direction for the modular axis	Set the traverse command for the axis set as the modular axis. 0: Unidirectional 1: Bidirectional
	2. Home	1	Coordinates of the 2nd origin	Set the coordinates of the 2nd origin.
		2	Coordinates of the 3rd origin	Set the coordinates of the 3rd origin.
		3	Coordinates of the 4th origin	Set the coordinates of the 4th origin.
	3. Rapid feed	2	Rapid feed acceleration	The set value is used as the acceleration of the G00 block. It is possible to write NC parameters during NC channel automatic operation.
		3	Rapid feed deceleration	The set value is used as the deceleration of the G00 block. It is possible to write NC parameters during NC channel automatic operation.
		4	Rapid feed jerk	The set value is used as the jerk of the G00 block. It is possible to write NC parameters during NC channel automatic operation.
		5	Rapid feed speed	The set value is used as the feed speed of the G00 block. (0~100000 unit/min, real number) It is possible to write NC parameters during NC channel automatic operation.
		6	Dry run speed	Used at dry run speed It is possible to write NC parameters during NC channel automatic operation.
	4. Feed area	1	Minimum value of the G22 Feed Prohibited Area range for the X, Y, and Z axis.	Set the minimum value of the G22 Feed Prohibited Area range for the X, Y, and Z axis. (-100,000~100,000 unit, real number) It is possible to write NC parameters during NC channel automatic operation.
		2	Maximum value of the G22 Feed Prohibited Area range for the X, Y, and Z axis.	Set the maximum value of the G22 Feed Prohibited Area range for the X, Y, and Z axis. (-100,000~100,000 unit, real number) It is possible to write NC parameters during NC channel automatic operation.
		3	Minimum value of the 3rd Feed Prohibited Area range for the X, Y, and Z axis.	Set the minimum value of the 3rd Feed Prohibited Area range for the X, Y, and Z axis. (-100,000~100,000 unit, real number) It is possible to write NC parameters during NC channel automatic operation.
		4	Maximum value of the 3rd Feed Prohibited Area range for the X, Y, and Z axis.	Set the maximum value of the 3rd Feed Prohibited Area range for the X, Y, and Z axis. (-100,000~100,000 unit, real number) It is possible to write NC parameters during NC channel automatic operation.
	5. Sub setting	1	Scaling setting	When using the scaling function (G51) function, this

Parameter	Group	No.	Item	Description
				parameter is applied to the axis without the scaling amount command. When set to a negative value, it is performed as a mirror image function. X, Y, Z, A, B, C, U, V, W, S set values for each of the 9 axes . . (-9999~9999, integer)
		2	Overrun feed rare of single direction positioning	Set the overrun feed rate of the 9 axes; X, Y, Z, A, B, C, U, V, W when using the single direction positioning function (G60). After stopping at the position separated by the set value for the G60 command block's axis, it moves to the command position to eliminate the effect of backlash.  (-100 ~ 100 unit, real number)

The parameter number, group number, and item number in the table above can be used as operands of NC_ReadParameter or NC_WriteParameter when reading or writing NC parameters in a motion program.
For the function block, refer to 6.8.14 and 6.8.15.

Motion Function Block



Input-Output		
UINT	NcChannel	Set the NC channel to make the command.
Input		
UINT	NcAxis	Set the channel axis. (1~10: X=1, Y=2, … B=8, C=9, S=10) When set to 0, 'Write Channel Parameters' is setting
INT	ParameterGroup	Specify the number in the group of the parameter.
INT	ParameterNumber	Specify the item number in the group of the parameter.

9.5 Spindle Functions

A spindle is a rotating axis that is used to equip a work-piece or a cutting tool in machine tools. The NC control of a motion controller provides methods to control the spindle axis and various operation functions.

9.5.1 Spindle device

This part explains basic settings to use the spindle axis and spindle devices supported to control the spindle axis in the NC control of a motion controller.

(1) Supported devices

The spindle axis in the NC control of a motion controller can set only by the slave connected by EtherCAT. Kinds of EtherCAT slaves that can be connected by the spindle axis are as follows:

1) EtherCAT Servo drive

To connect with the NC spindle axis, the EtherCAT servo drive should support the csv (cyclic synchronous position) mode among operation modes of the CiA402 drive profile.

To operate the csv operation mode, the following objects should be registered with the EtherCAT PDO setting.

Type	Index	Parameter name
TxPDO	0x60410	Status word
	0x606C0	Velocity actual value
RxPDO	0x6040:0	Control word
	0x60FF:0	Target velocity

※ The position actual value (0x6064:0) object can be used instead of the velocity actual value (0x606C:0) object.

2) EtherCAT inverter

To connect with the NC spindle axis, the EtherCAT inverter should support the vl (velocity, frequency converter) mode among operation modes of the CiA402 drive profile.

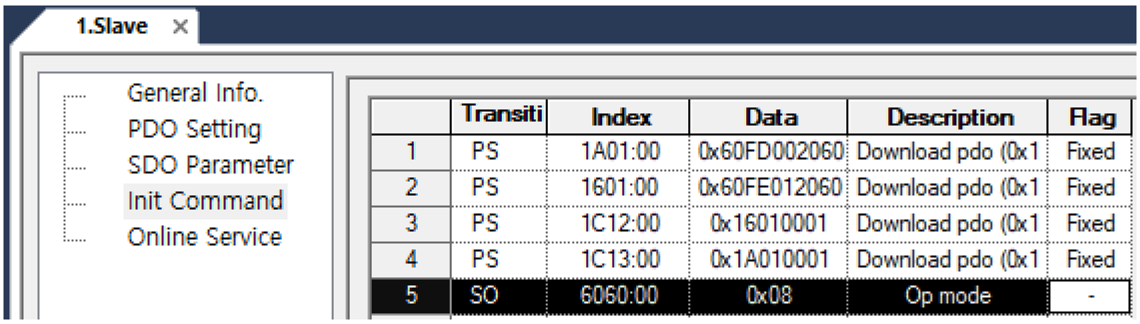
To operate the vl operation mode, the following objects should be registered with the EtherCAT PDO setting.

Type	Index	Parameter name
TxPDO	0x6041:0	Status word
	0x6044:0	vl velocity actual value
RxPDO	0x6040:0	Control word
	0x6042:0	vl target velocity

Notes

If the basic operation mode of drive profiles is not set to csv or vl according to supported devices, set the operation mode to csv (9) or vl (2) using the ‘EtherCAT parameter – Slave – Start command’.

[Ex.] Set the operation mode to csv (9) in the start command of the LS ELECTRIC L7NH servo drive



	Transiti	Index	Data	Description	Flag
1	PS	1A01:00	0x60FD002060	Download pdo (0x1	Fixed
2	PS	1601:00	0x60FE012060	Download pdo (0x1	Fixed
3	PS	1C12:00	0x16010001	Download pdo (0x1	Fixed
4	PS	1C13:00	0x1A010001	Download pdo (0x1	Fixed
5	SO	6060:00	0x08	Op mode	-

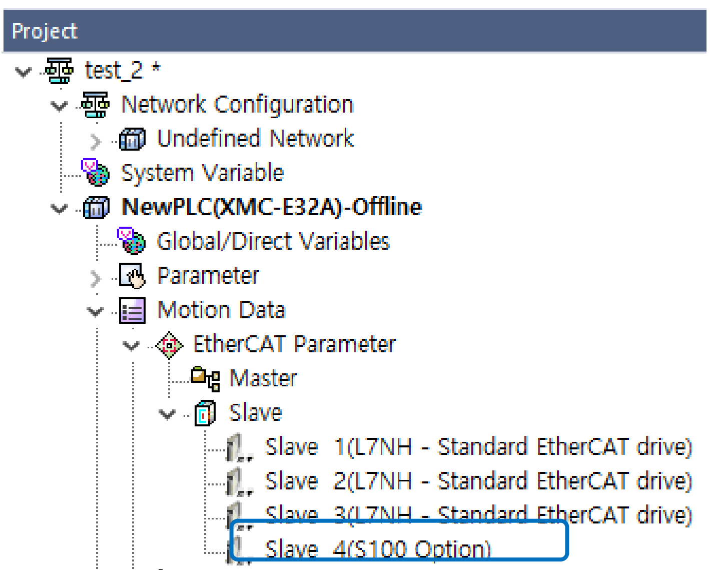
(2) How to set the spindle axis

To use the spindle axis in the NC control, set motion data in the following order:

1) EtherCAT parameter – Slave registration

Register the EtherCAT device to be used as the spindle axis with ‘EtherCAT parameter - Slave’.

[Ex.] Register the LS ELECTRIC S100 inverter as the EtherCAT ‘slave 4’



※ Confirm if essential objects for the vl operation mode is registered by confirming the PDO setting of slaves

PDO Type Filter: TxPDO

Index	Name	Data type	Size(Bytes)
0x6041:0	Status word	UINT	2
0x6044:0	vl velocity actual value	INT	2

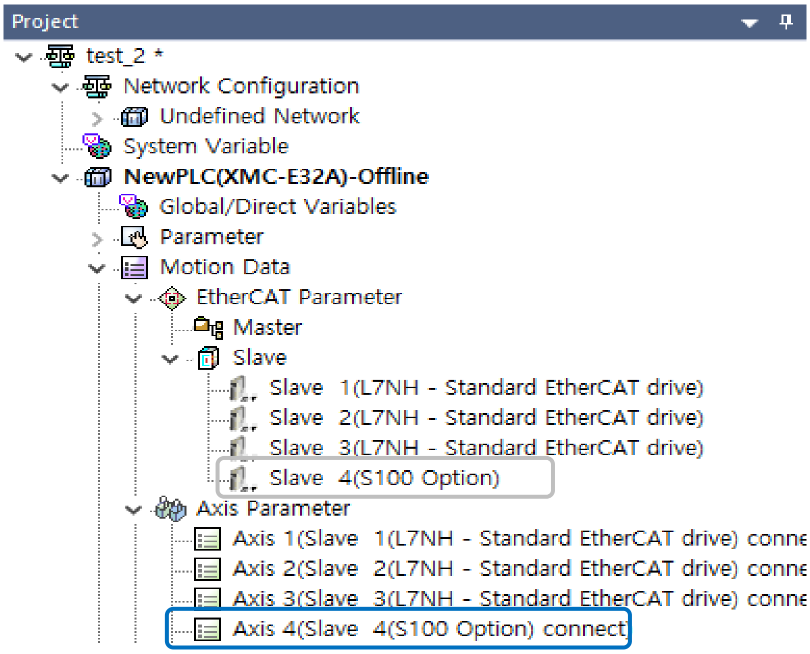
PDO Type Filter: RxPDO

Index	Name	Data type	Size(Bytes)
0x6040:0	Control word	UINT	2
0x6042:0	vl target velocity	INT	2

2) Axis - Connect the EtherCAT slave

Connect the EtherCAT device to be used as the spindle axis to the axis of the axis parameter.

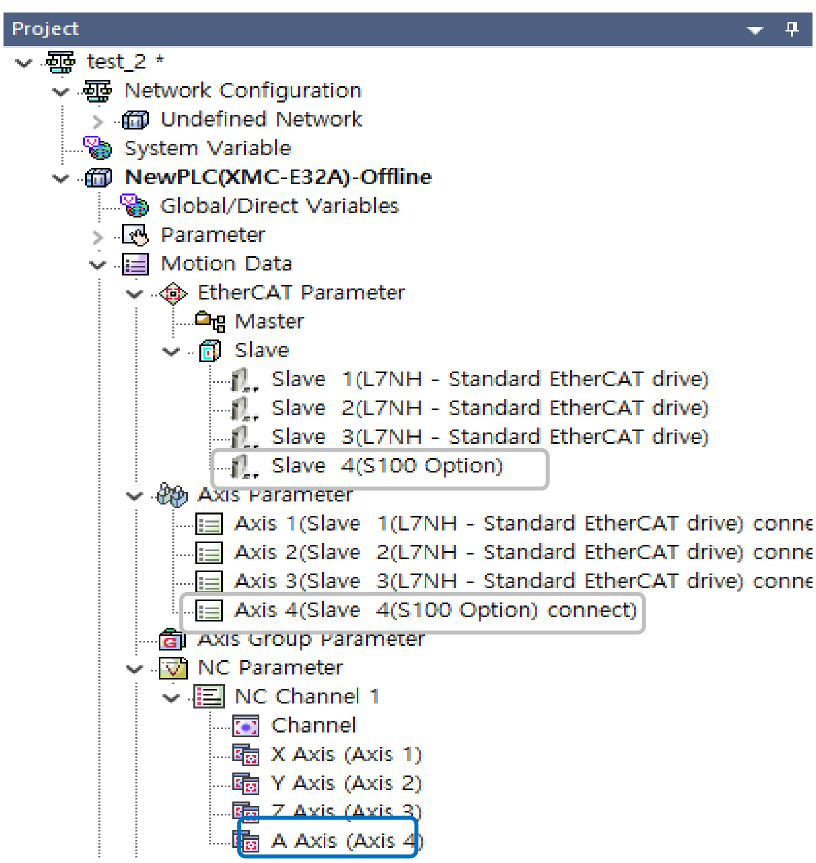
[Ex.] Connect the slave 4 (S100 Option) to ‘Axis 4’



3) NC channel/Axis - Connection of an axis

Connect the axis to be used as the spindle axis among axes registered with the axis parameter to NC channel S axis, or spindle axis.

[Ex.] Connect Axis 4 (connection of Slave 4 (S1000 Option)) to the NC S axis



4) NC channel parameter - Set how to operate the spindle axis

Set how to handle the spindle M/S-code of the spindle axis and the main spindle axis number according to how to operate the spindle axis.

[Ex.] Set 'Axis 4' connected to the NC channel S axis to the 'main spindle axis number'

Group	Name	Channel 1
Basic Settings	Spindle Operation of Spindle Axis M/S-code	0: Automatic 1: After Command Execution
	Minimum command unit	0 mm
	Whether to use G22 [No traveling area]	0: Used
	Inner/Outer side of G22 [No traveling area]	0: Inner side
	Whether to use the 3rd [No traveling area]	0: Used
	Reverse driving buffer size	0
	Rotary axis of cylindrical interpolation	0: X Axis
	Linear axis for interpolating the polar coordinate	0: None
	Rotary axis for interpolating the polar coordinate	0: None
	Main spindle axis number	0:Disabled
	Monitoring time for in-position completion	5000 ms

9.5.2 How to Operate the Spindle Axis

Users can set how to operate the spindle axis in the NC control of a motion controller. The spindle axis that is set to the main spindle axis number according to values of the M-code and S-code can be automatically operated or users can control the spindle axis directly using motion commands after the NC function module confirms the values of the M-code and S-code in a task program.

(1) Automatic operation in the NC function module

1) Operation

If the block where the spindle-related M-code (M03, M04, M05, M19) and S-code are used in the NC program is executed, the spindle axis set to the main spindle axis number is automatically operated in the NC function module. Users do not need to control the spindle axis by separately confirming the values of the M-code or S-code in a task program.

They can select how to continue operation when the block where the spindle-related M-code (M03, M04, M05, M19) and S-code are used is executed according to 'How to handle the spindle operation M/S-code of the spindle axis' of the NC channel parameter.

(a) 0: Automatic operation continues (Described based on NC channel 1).

a) If the block where the spindle-related M-code (M03, M04, M05, M19) and S-code are used is executed, the relevant spindle operation is automatically executed.

b) The 'NC Channel 01 M Code Output Strobe Signal' (_NC01_McodeStrobe) is not generated.

c) Users can confirm the M-code data with the 'NC Channel 01 M Code Data Output' (_NC01_McodeData).

d) The 'NC channel 01 S Output Strobe Signal' (_NC01_ScodeStrobe) is not generated.

e) Users can confirm the S-code data with the 'NC Channel 01 S Code Data Output' (_NC01_ScodeData).

(b) 1: Operation continues after completion' (Described based on NC channel 1).

a) If the block where the spindle-related M-code (M03, M04, M05, M19) is used is executed, the relevant block stops automatic operation of the NC program until the 'M Code Operation Completion' (NC_McodeComplete) command is executed.

b) If the 'M Code Operation Completion' (NC_McodeComplete) command is executed, the spindle operation corresponding to the spindle-related M-code is automatically executed in the NC function module and the program of the next block conducts automatic operation.

c) If the block where the S-code is used is executed, the relevant block stops automatic operation of the NC program until the 'S Code Operation Completion' (NC_ScodeComplete) command is executed.

d) If the 'S Code Operation Completion' (NC_ScodeComplete) command is executed, the spindle operation corresponding to the S-code is automatically executed in the NC function module and the program of the next block is automatically operated.

e) Users can confirm that the M-code is executed with the 'NC Channel 01 M Code Output Strobe Signal' (_NC01_McodeStrobe). They also can confirm the M-code data with the 'NC Channel 01 M Code Data Output' (_NC01_McodeData).

f) Users can confirm that the M-code is executed with the 'NC Channel 01 S Code Output Strobe Signal' (_NC01_ScodeStrobe).They also can confirm the S-code data with the 'NC Channel 01 S Code Data Output' (_NC01_ScodeData).

(c) Flag

Program	Variable/Device	Type	Device/Variable	Comment
<GLOBAL>	%FW32840	WORD	_NC01_MainSpindle	Check NC channel 01 main spindle axis number
<GLOBAL>	%FX526080	BOOL	_NC01_McodeStrobe	NC Ch. 01 M code output strobe signal
<GLOBAL>	%FD16441	DWORD	_NC01_McodeData	NC Ch. 01 M code data output
<GLOBAL>	%FX526144	BOOL	_NC01_ScodeStrobe	NC Ch. 01 S code output strobe signal
<GLOBAL>	%FD16443	DWORD	_NC01_ScodeData	NC Ch. 01 S code data output

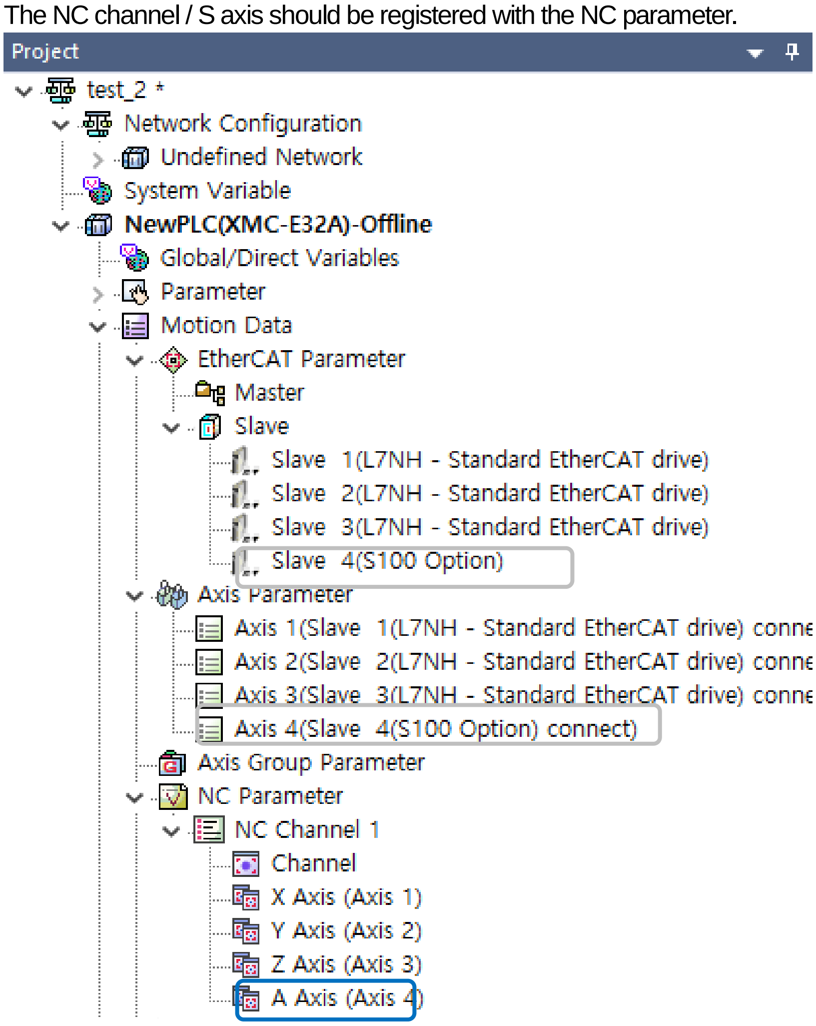
※ If the NC spindle axis conducts automatic operation in the NC function module, users can confirm the number of the axis set as the main spindle axis in the NC Channel 01 Main Spindle Axis' (_NC01_MainSpindle) flag.

2) Parameter setting

To conduct automatic operation of the spindle axis in the NC function module, parameters should be set as follows:

(a) NC parameter

The NC channel / S axis should be registered with the NC parameter.



(b) NC channel parameter - Main spindle axis number

The number of axes connected to the NC channel / S axis should be set identically to the 'main spindle axis number'.

Group	Name	Channel 1
Basic Settings	Spindle Operation of Spindle Axis M/S-code	0: Automatic 1: After Command Execution
	Minimum command unit	0 mm
	Whether to use G22 [No traveling area]	0: Used
	Inner/Outer side of G22 [No traveling area]	0: Inner side
	Whether to use the 3rd [No traveling area]	0: Used
	Reverse driving buffer size	0
	Rotary axis of cylindrical interpolation	0: X Axis
	Linear axis for interpolating the polar coordinate	0: None
	Rotary axis for interpolating the polar coordinate	0: None
	Main spindle axis number	0:Disabled
	Monitoring time for in-position completion	5000 ms

(2) User operation in a task program

1) Operation

If the block where the spindle-related M-code (M03, M04, M05, M19) and S-code are used in the NC program, users can control the spindle axis by separately confirming the values of the M-code or S-code in a task program.

(a) Treatment order (Described based on NC channel 1).

a) If the block where the M-code (M03, M04, M05, M19) and S-code are used in the NC program, the relevant block stops automatic operation of the NC program.

b) Users can confirm that the M-code is executed with the 'NC Channel 01 M Code Output Strobe Signal' (_NC01_McodeStrobe) in a task program. They also can learn what spindle operation is executed by confirming the M-code data with the 'NC Channel 01 M Code Data Output' (_NC01_McodeData).

c) Conduct the spindle axis operation corresponding to the M-code value by confirming the 'NC Channel 01 M Code Data Output' (_NC01_McodeData) value.

[Ex.] When executing the 'M03 S3000' block in the NC program

As 3 is saved in the 'NC Channel 01 M Code Data Output' (_NC01_McodeData) and 3000 is saved in the 'NC Channel 01 S Code Data Output' (_NC01_ScodeData), write a program to operate the spindle axis at 3000 rpm by using the LS_SyncMoveVelocity command after confirming the values.

[Ex.] When executing the 'M05' block in the NC program

As 5 is saved in the 'NC Channel 01 M Code Data Output' (_NC01_McodeData), write a program to stop the spindle axis by using the MC_Halt or MC_Stop commands after confirming the values.

d) After executing operation of the spindle axis corresponding to the M-code or S-code, execute the 'S Code Operation Completion' (NC_McodeComplete) command to automatically operate the next block program.

(b) Flag

Variable/Device	Type	Device/Variable	Comment
%FW32840	WORD	_NC01_MainSpindle	Check NC channel 01 main spindle axis number
%FX526080	BOOL	_NC01_McodeStrobe	NC Ch. 01 M code output strobe signal
%FD16441	DWORD	_NC01_McodeData	NC Ch. 01 M code data output
%FX526144	BOOL	_NC01_ScodeStrobe	NC Ch. 01 S code output strobe signal
%FD16443	DWORD	_NC01_ScodeData	NC Ch. 01 S code data output

※ The value of the'NC Channel 01 Confirm the Main spindle axis number' (_NC01_MainSpindle) flag becomes 0.

2) Parameter setting

In order that users directly control the spindle axis in a task program, not to conduct its automatic operation in the NC function module, parameters should be set as follows:

(a) NC channel parameter - Main spindle axis number

Set the 'main spindle axis number' to '0: Disable'.

NC Channel parameter

Group	Name	Channel 1
Basic Settings	Minimum command unit	0 mm
	Whether to use G22 [No traveling area]	0: Used
	Inner/Outer side of G22 [No traveling a	0: Inner side
	Whether to use the 3rd [No traveling ar	0: Used
	Reverse driving buffer size	0
	Rotary axis of cylindrical interpolation	0: X Axis
	Linear axis for interpolating the polar coordinate	0: None
	Rotary axis for interpolating the polar coordinate	0: None
	Main spindle axis number	0:Disabled
	Monitoring time for in-position completi	5000 ms

9.5.3 Spindle Related Parameter

If controlling the spindle axis in the NC control of a motion controller, explain the relevant parameter.

(1) Axis Parameter

Item	Description	Setting range	Initial values
Backlash compensation amount	Set the value to compensate backlash of machine.	0 or Long real (LREAL) positive number	0 [unit]
Select the Spindle Encoder	Set the method that an encoder attached to a motor of the spindle axis is connected.	0: Unused 1: Motor ENC 2: Built-in ENC1 3: Built-in ENC2 4: EtherCAT ENC	0
Number of pulses per rotation of the spindle EtherCAT encoder	If the 'spindle encoder selection' parameter setting value is '4: EtherCAT ENC', set number of pulses per rotation of an encoder.	1 ~ 4294967295	8192 pls
Spindle EtherCAT encoder position variable/address	If the 'spindle encoder selection' parameter setting value is '4: EtherCAT ENC', set the device where the current position of the encoder is saved.	%ID0 ~ %ID4095 %MD0 ~ %MD524287	%ID0
The P Gain of the Spindle Positioning Mode	Set the P gain value that the spindle axis uses when controlling position.	1 ~ 500 Hz	30 Hz
The Feed Forward Gain of the Spindle Positioning Mode	Set the feed forward gain value that the spindle axis uses when controlling position.	0 ~ 100 %	0 %
How to conduct the homing operation	Set the homing operation method when executing the NC_Home command to the spindle axis.	0: Servo drive supported 33: Reverse direction, Z phase 34: Forward direction, Z phase 35: Set the homing of the current position	0
Switch navigation speed of the homing operation	Set the operated speed to detect switch signals after starting the homing operation.	Long real (LREAL) positive number	60 rpm
Zero navigation speed of the homing operation	Set the operated speed to detect zero signals after starting the homing operation.		12 rpm
Acceleration/deceleration of the homing operation	Set acceleration/deceleration to accelerate and decelerate to the target speed after starting the homing operation.	0 or Long real (LREAL) positive number	1000
Z phase variable/address	Set the device where the Z phase signal used as the Zero signal of the homing operation is saved.	%IX0 ~ %IX131072 %MX0 ~ %MX16777215	%IX0
Orientation velocity	When the M19 Orientation command is executed on the NC program, set the Orientation position	Long real (LREAL) positive number	60 rpm

Orientation diredtion	(offset) and velocity, and the traveling direction.	0:forward direction , 1:reverse direction	0-Forward direction
Orientation offset		0 ~ 360	0
The tolerance range to reach the spindle rotation command speed	Determine whether to reach the command speed of the spindle axis by the set value.	0 ~ 100 %	95 %
The tolerance RPM to reach the spindle rotation zero speed	Determine whether to reach the zero speed of the spindle axis by the set value.	0~ 100 rpm	5 rpm

(2) Channel parameter

Group	No.	Item	Description
1. Standard settings	25	Main spindle axis number	Set the number of an axis to be used as the main spindle axis in the NC channel. Set the system that does not use the spindle axis to 0. To automatically control spindle commands in the NC function module, set it exactly the same as the axis number connected to the NC S axis. 0: Unused 1 ~ 36: Axis 1 ~ Axis 36
	34	The spindle operation of the spindle axis. How to treat M/S-code	If automatically controlling spindle commands in the NC function module, set the methods to handle M/S-codes related to the spindle operation. When executing M/S-code commands, conduct automatic control of the spindle axis without M/S-code operation completion commands. To execute the next block, select '0: Continue automatic operation'. If a user wants to automatically control the spindle axis after completing to use M/S-code operation completion commands, select '1: Operation continues after completing commands'. 0: Automatic operation continues 1: Operation continues after completing commands

(3) Channel / S axis parameter

Group	No.	Item	Description
3. Rapid feed	2	Rapid feed acceleration	When the block where the spindle-related M-code (M03, M04, M05, M19) and S-code used in the NC program is executed, it is used as acceleration, deceleration and jerk values.
	3	Rapid feed deceleration	
	4	Rapid feed jerk	

9.5.4 Spindle operation function

Users can set how to operate the spindle axis in the NC control of a motion controller. If the spindle axis that is set to the main spindle axis number according to values of the M-code and S-code is automatically operated in the NC function module, explain the spindle operation function.

(1) Forward operation

1) Operation

When executing the M03 block in the NC program, the spindle axis is operated forward at the speed set in the S-code. Conduct operation up to the target speed using 'rapid traverse acceleration/deceleration/jerk' of channels / S axis parameters.

After starting the M03 forward operation, the program of the next block is automatically operated.

2) Status

After starting the M03 forward operation, the flag value is set as follows: (Described based on NC channel 1).

Variable	Description	Value
_NC01_TVelOfSpindle	NC channel 01 target speed of the spindle (S command value)	Designated values of S-code in a program
_NC01_CVelOfSpindle	NC channel 01 Spindle Command Velocity	Values of the current command velocity of the spindle
_NC01_SpindleStop	NC channel 01 Signal to confirm spindle stop status	Off
_NC01_SpindleCW	NC channel 01 Signal to confirm spindle CW status	On
_NC01_SpindleCCW	NC channel 01 Signal to confirm spindle CCW status	Off
_NC01_SpindleCVelAgr	NC channel 01 Signal to confirm the status of reaching spindle command velocity	After reaching the target velocity, turn On
_NC01_SpindleZeroVel	NC channel 01 Signal to confirm the status of reaching zero velocity of the spindle	Off
_NC01S_ForwardRun	NC channel 01 Traversing axes toward the axis S +	On
_NC01S_ReverseRun	NC channel 01 Traversing axes toward the axis S -	Off
_NC01S_SpindleRun	NC channel 01 Axis S spindle operation	On

(2) Reverse operation

1) Operation

When executing the M04 block in the NC program, the spindle axis is operated in reverse at the speed set in the S-code. Conduct operation up to the target speed using 'rapid traverse acceleration/deceleration/jerk' of channels / S axis parameters.

After starting the M04 reverse operation, the program of the next block is automatically operated.

2) Status

After starting the M04 reverse operation, the flag value changes as follows: (Described based on NC channel 1).

Variable	Description	Value
_NC01_TVelOfSpindle	NC channel 01 target speed of the spindle (S command value)	Designated values of S-code in a program
_NC01_CVelOfSpindle	NC channel 01 Spindle Command Velocity	Values of the current command velocity of the spindle

_NC01_SpindleStop	NC channel 01 Signal to confirm spindle stop status	Off
_NC01_SpindleCW	NC channel 01 Signal to confirm spindle CW status	Off
_NC01_SpindleCCW	NC channel 01 Signal to confirm spindle CCW status	On
_NC01_SpindleCVelAgr	NC channel 01 Signal to confirm the status of reaching spindle command velocity	After reaching the target velocity, turn On
_NC01_SpindleZeroVel	NC channel 01 Signal to confirm the status of reaching zero velocity of the spindle	Off
_NC01S_ForwardRun	NC channel 01 Traversing axes toward the axis S +	Off
_NC01S_ReverseRun	NC channel 01 Traversing axes toward the axis S -	On
_NC01S_SpindleRun	NC channel 01 Axis S spindle operation	On

(3) Stop

1) Operation

If executing the M05 block in the NC program, when operating the spindle axis, stop the spindle axis.

Conduct operation using ‘rapid traverse acceleration/deceleration/jerk’ of channels / S axis parameters to stop the spindle axis.

After starting the M05 stop operation, the program of the next block is automatically operated.

2) Status

After starting the M05 stop operation, the flag value changes as follows: (Described based on NC channel 1).

Variable	Description	Value
_NC01_TVelOfSpindle	NC channel 01 target speed of the spindle (S command value)	Designated values of S-code in a program
_NC01_CVelOfSpindle	NC channel 01 Spindle Command Velocity	Values of the current command velocity of the spindle
_NC01_SpindleStop	NC channel 01 Signal to confirm spindle stop status	On when stopped
_NC01_SpindleCW	NC channel 01 Signal to confirm spindle CW status	Turn Off when stopping after keeping the previous status
_NC01_SpindleCCW	NC channel 01 Signal to confirm spindle CCW status	Turn Off when stopping after keeping the previous status
_NC01_SpindleCVelAgr	NC channel 01 Signal to confirm the status of reaching spindle command velocity	Off
_NC01_SpindleZeroVel	NC channel 01 Signal to confirm the status of reaching zero velocity of the spindle	On when stopped
_NC01S_ForwardRun	NC channel 01 Traversing axes toward the axis S +	Turn Off when stopping after keeping the previous status
_NC01S_ReverseRun	NC channel 01 Traversing axes toward the axis S -	Turn Off when stopping after keeping the previous status

(4) Speed change

1) Operation

When executing the S-code block during execution of M03 or M04 in the NC program, the current operating speed changes to the value of velocity set to S.

Conduct operation using 'rapid traverse acceleration/deceleration/jerk' of channels / S axis parameters to change velocity of the spindle axis.

After starting the S-code velocity change operation, the program of the next block is automatically operated.

2) Status

After starting the M05 stop operation, the flag value changes as follows: (Described based on NC channel 1).

Variable	Description	Value
_NC01_TVelOfSpindle	NC channel 01 target speed of the spindle (S command value)	Designated values of S-code in a program
_NC01_CVelOfSpindle	NC channel 01 Spindle Command Velocity	Values of the current command velocity of the spindle
_NC01_ScodeData	NC channel 01 S Code Data output	Designated values of S-code in a program
_NC01_SpindleCW	NC channel 01 Signal to confirm spindle CW status	Keep previous status
_NC01_SpindleCCW	NC channel 01 Signal to confirm spindle CCW status	Keep previous status
_NC01_SpindleCVelAgr	NC channel 01 Signal to confirm the status of reaching spindle command velocity	After reaching the target velocity, turn On
_NC01_SpindleZeroVel	NC channel 01 Signal to confirm the status of reaching zero velocity of the spindle	Off
_NC01S_ForwardRun	NC channel 01 Traversing axes toward the axis S +	Keep previous status
_NC01S_ReverseRun	NC channel 01 Traversing axes toward the axis S -	Keep previous status

(5) Home operation

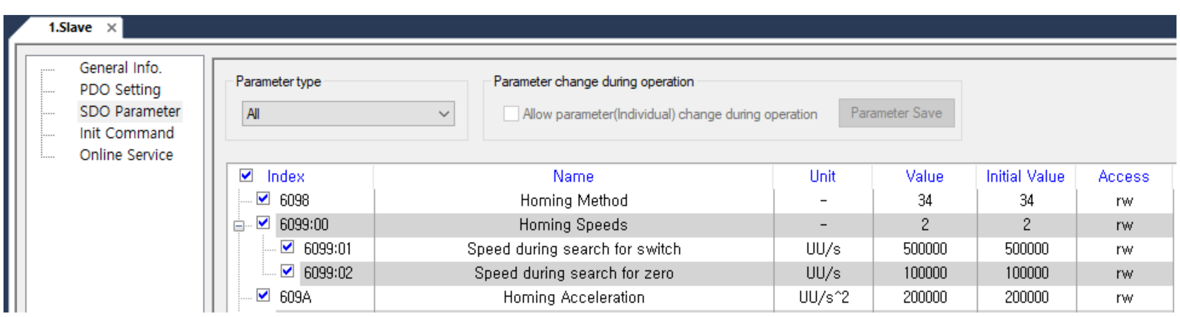
1) Operation

When executing commands by setting the NcAxis input of the NC_Home command to 10 (S axis) and the ReferenceNum input to '1: The 1st homing', execute the homing operation of the spindle axis according to the method set in the 'method for homing operation' of the axis parameter when executing commands.

Axis parameter - Operation according to the 'method for homing operation' is as follows:

(a) 0: Servo drive supported

If the axis connected to the spindle axis is a servo drive, the homing operation supported by the servo drive is executed. Servo drive parameters used for homing operation are as follows:



(b) 33: Reverse direction, Z phase

Set the Z phase position as the homing after executing the homing operation on the NC function module of a motion controller and starting reverse operation.

Operation after executing the NC_Home command is as follows: (Described based on NC channel 1).

a) The 'NC Channel 01 Axis S Homing Completion' (_NC01S_HomeCmpl) flag turns Off.

b) Start reverse operation at the 'Zero navigation speed of the homing operation' and 'Acceleration/deceleration of the homing operation'.

c) Stop acceleration if detecting the Z phase (the device set to 'Z phase variable/address' turns On).

d) Start forward operation at the 'Zero navigation speed of the homing operation' and 'Acceleration/deceleration of the homing operation'.

e) Stop at the location where the Z phase is detected.

f) The 'NC Channel 01 Axis S Homing Completion' (_NC01S_HomeCmpl) flag turns On.

g) The value of the 'NC channel 01 Axis S Command position in the Work-piece Coordinate system' (_NC01S_CmdPosInWC) is set to 0.

(c) 34: Forward direction, Z phase

Set the Z phase position as the homing after executing the homing operation on the NC function module of a motion controller and starting forward operation.

Operation after executing the NC_Home command is as follows: (Described based on NC channel 1).

a) The 'NC Channel 01 Axis S Homing Completion' (_NC01S_HomeCmpl) flag turns Off.

b) Start forward operation at the 'Zero navigation speed of the homing operation' and 'Acceleration/deceleration of the homing operation'.

c) Stop acceleration if detecting the Z phase (the device set to 'Z phase variable/address' turns On).

d) Start reverse operation at the 'Zero navigation speed of the homing operation' and 'Acceleration/deceleration of the homing operation'.

e) Stop at the location where the Z phase is detected.

f) The 'NC Channel 01 Axis S Homing Completion' (_NC01S_HomeCmpl) flag turns On.

g) The value of the 'NC channel 01 Axis S Command position in the Work-piece Coordinate system' (_NC01S_CmdPosInWC) is set to 0.

(d) 35: Set the homing of the current position

The current position of the spindle axis becomes a reference position. (Described based on NC channel 1). After executing the NC_Home command, the 'NC channel 01 Axis S Homing completion' (_NC01S_HomeCmpl) flag turns On and the value of the 'NC channel 01 Axis S Command position in the Work-piece Coordinate system' (_NC01S_CmdPosInWC) is set to 0.

2) Parameter setting

Parameters related to homing operation of the spindle axis are as follows:

Item	Description	Setting range
Select the Spindle Encoder	Set the method that an encoder attached to a motor of the spindle axis is connected.	1: Motor ENC 2: Built-in ENC1 3: Built-in ENC2 4: EtherCAT ENC
Number of pulses per rotation of the spindle EtherCAT encoder	If the 'spindle encoder selection' parameter setting value is '4: EtherCAT ENC', set number of pulses per rotation of an encoder.	1 ~ 4294967295
Spindle EtherCAT encoder position variable/address	If the 'spindle encoder selection' parameter setting value is '4: EtherCAT ENC', set the device where the current position of the encoder is saved.	%ID0 ~ %ID4095 %MD0 ~ %MD524287
How to conduct the homing operation	Set the homing operation method when executing the NC_Home command to the spindle axis.	0: Servo drive supported 33: Reverse direction, Z phase 34: Forward direction, Z phase 35: Set the homing of the current position
Switch navigation speed of the homing operation	Set the operated speed to detect switch signals after starting the homing operation.	Long real (LREAL) positive number
Zero navigation speed of the homing operation	Set the operated speed to detect zero signals after starting the homing operation.	
Acceleration/deceleration of the homing operation	Set acceleration/deceleration to accelerate and decelerate to the target speed after starting the homing operation.	0 or Long real (LREAL) positive number
Z phase variable/address	Set the device where the Z phase signal used as the Zero signal of the homing operation is saved.	%IX0 ~ %IX131072 %MX0 ~ %MX16777215

If the 'spindle encoder selection' parameter is '0: Disable', the homing operation cannot be executed. If satisfying the following conditions according to the 'spindle encoder selection' parameter, homing operation can be normally executed.

(a) '1: Motor ENC'

The position actual value (0x6064:0) object should be set in the setting of the EtherCAT Slave TxPDO.

(b) '2: Built-in ENC1'

a) Unit of Encoder 1 = 0: pulse

b) Encorder1 max. value = 2147483647 pls

c) Encorder1 min. value = -2147483648 pls

(c) '3: Built-in ENC2'

a) Unit of Encoder 2 = 0: pulse

b) Encorder2 max. value = 2147483647 pls

c) Encorder2 min. value = -2147483648 pls

(d) '4: EtherCAT ENC'

The 'Number of pulses per rotation of the spindle EtherCAT encoder' and the 'Spindle EtherCAT encoder position variable/address' parameters should be set.

3) Status

After starting homing operation, the status flag value changes as follows: (Described based on NC channel 1).

Variable	Description	Value
_NC01S_Homing	NC channel 01 Axis S reference position return operation	On when executing a command, Off after completing it
_NC01S_HomeCmpl	NC channel 01 Axis S homing completion	On after completing normally
_NC01S_CmdPosInWC	NC channel 01 Axis S Command position of Work-piece Coordinate system	0 after completing normally
_NC01S_CmdPosInMC	NC channel 01 Axis S Command position (Machine Coordinate system)	0 after completing normally

※ As the spindle axis changes into the speed control operation and the command position of the spindle axis is updated to the current position after completing the homing operation, the position cannot be exactly 0 due to motor vibration, etc.

4) Exclusive conditions

As the homing operation of the spindle axis cannot be executed under the following conditions, errors occur when executing the NC_Home command.

(a) If the 'spindle encoder selection' parameter is '0: Disable', (Error code - 0x3637)

(b) If the 'spindle encoder selection' parameter is '1: Moter ENC' and there is not the Position actual value (0x6064:0) object in the EtherCAT slave TxPDO setting, (Error code - 0x3638)

(c) If the 'spindle encoder selection' parameter is '2: Built-in ENC1' and the encoder 1 parameter setting does not satisfy the following, (Error code - 0x3639)

a) Unit of Encoder 1 = 0: pulse

b) Encorder1 max. value = 2147483647 pls

c) Encorder1 min. value = -2147483648 pls

(d) If the 'spindle encoder selection' parameter is '3: Built-in ENC2' and the encoder 2 parameter setting does not satisfy the following, (Error code - 0x363A)

a) Unit of Encoder 2 = 0: pulse

b) Encorder2 max. value = 2147483647 pls

c) Encorder2 min. value = -2147483648 pls

(6) Orientation

1) operation(Described based on NC channel 1).

When executing the M19 block in the NC program, move the spindle axis to the 'Spindle Orientation Offset' position of the spindle axis.

Start operation at the 'Orientation speed' of the axis parameter and toward the 'Orientation direction'. Conduct operation up to the target speed using 'rapid traverse acceleration/deceleration/jerk' of channels / S axis parameters. Stop at the 'Orientation Offset' position after starting the M19 Orientation operation and then, a program of the next block is automatically operated.

When executing the M19 Orientation operation, the detailed operation is as follows:

(a) If the spindle axis is operated by M03 or M04 commands, decelerate and stop.

(b) Start operation with 'Orientation velocity', NC axis S's rapid traverse acceleration/deceleration/jerk'. Orientation direction is operated toward the direction set to the 'Orientation direction' parameter.

(c) After starting operation, decelerate and stop at the 'Orientation Offset' position.

(d) Wait until the 'NC channel 01 Axis S In-position Detection' (_NC01S_INPOSITION) signal turns On.

(e) After detecting in-position, the 'NC channel 01 Confirm Spindle Orientation Status signal' (_NC01_SpindleOrient) flag turns On.

2) Parameter setting

Parameters related to Orientation operation of the spindle axis are as follows:

Item	Description	Setting range
Select the Spindle Encoder	Set the method that an encoder attached to a motor of the spindle axis is connected.	1: Motor ENC 2: Built-in ENC1 3: Built-in ENC2 4: EtherCAT ENC
Number of pulses per rotation of the spindle EtherCAT encoder	If the 'spindle encoder selection' parameter setting value is '4: EtherCAT ENC', set number of pulses per rotation of an encoder.	1 ~ 4294967295
Spindle EtherCAT encoder position variable/address	If the 'spindle encoder selection' parameter setting value is '4: EtherCAT ENC', set the device where the current position of the encoder is saved.	%ID0 ~ %ID4095 %MD0 ~ %MD524287
Orientation velocity	When the M19 Orientation command is executed on the NC program, set the Orientation position (offset) and velocity, and the traveling direction.	Long real (LREAL) positive number
Orientation diredtion		0:forward direction , 1:reverse direction
Orientation offset		0 ~ 360

If the 'spindle encoder selection' parameter is '0: Disable', the Orientation operation cannot be executed. If satisfying the following conditions according to the 'spindle encoder selection' parameter, Orientation operation can be normally executed.

(a) '1: Motor ENC'

The position actual value (0x6064:0) object should be set in the setting of the EtherCAT Slave TxPDO.

(b) '2: Built-in ENC1'

d) Unit of Encoder 1 = 0: pulse

e) Encorder1 max. value = 2147483647 pls
f) Encorder1 min. value = -2147483648 pls
(c) '3: Built-in ENC2'
d) Unit of Encoder 2 = 0: pulse
e) Encorder2 max. value = 2147483647 pls
f) Encorder2 min. value = -2147483648 pls
(d) '4: EtherCAT ENC'
The 'Number of pulses per rotation of the spindle EtherCAT encoder' and the 'Spindle EtherCAT encoder position variable/address' parameters should be set.

3) Status
After starting Orientation operation, the status flag value changes as follows: (Described based on NC channel 1).

Variable	Description	Value
_NC01_SpindleOrient	NC channel 01 Spindle Signal to confirm Orientation status	On when executing a command, Off after completing it
_NC01S_CmdPosInWC	NC channel 01 Axis S Command position of Work-piece Coordinate system	Orientation offset after completing normally
_NC01S_CmdPosInMC	NC channel 01 Axis S Command position (Machine Coordinate system)	Orientation offset after completing normally

※ As the spindle axis changes into the speed control operation and the command position of the spindle axis is updated to the current position after completing the Orientation operation, the position cannot be exactly 0 due to motor vibration, etc.

4) Exclusive conditions
As the Orientation operation of the spindle axis cannot be executed under the following conditions, errors occur when executing the M19 command.
(a) If the spindle axis is not the decision status of a reference position (Error code - 0x3872)
(b) If the 'spindle encoder selection' parameter is '0: Disable', (Error code - 0x3873)
(c) If the 'spindle encoder selection' parameter is '1: Moter ENC' and there is not the Position actual value (0x6064:0) object in the EtherCAT slave TxPDO setting, (Error code - 0x3874)
(d) If the 'spindle encoder selection' parameter is '2: Built-in ENC1' and the encoder 1 parameter setting does not satisfy the following, (Error code - 0x3875)
d) Unit of Encoder 1 = 0: pulse
e) Encorder1 max. value = 2147483647 pls
f) Encorder1 min. value = -2147483648 pls
(e) If the 'spindle encoder selection' parameter is '3: Built-in ENC2' and the encoder 2 parameter setting does not satisfy the following, (Error code - 0x3876)
d) Unit of Encoder 2 = 0: pulse
e) Encorder2 max. value = 2147483647 pls
f) Encorder2 min. value = -2147483648 pls

(7) Constant surface speed control
1) Operation
The function is used when machining a material with an inconstant diameter. When executing the G96 (constant surface

speed control) command in the NC program, the NC spindle axis controls constant surface speed.
Keep the cutting speed constant and revolve the spindle axis by calculating the rotation number of the spindle according change in material diameters.
For more information, see 9.3.3 Explanation of NC Commands (1) G code 35) Surface speed control (G96).

(8) Spindle override

1) Operation

Users can set an overdrive ratio over the spindle speed command, acceleration/deceleration and jerk by using the spindle overdrive command in a task program. The spindle overdrive can be executed during or before automatic operation of the NC program. It does not apply when controlling constant surface speed or conducting the homing operation.
For more information, see '6.8.10 Spindle overdrive (NC_SpindleOverride)'.

9.5.5 Spindle operation State

The NC control of a motion controller provides a flag that can confirm the operation status of the spindle axis.

(1) NC channel flag (Described based on NC channel 1).

Variable	Memory	Description
_NC01_ConstSurfSpeed	%FX524738	NC channel 01 Signal controlling constant surface speed
_NC01_TVelOfSpindle	%FL8203	NC channel 01 target speed of the spindle (S command value)
_NC01_CVelOfSpindle	%FL8204	NC channel 01 Spindle Command Velocity
_NC01_SpindleOverride	%FL8208	NC channel 01 Spindle Overdrive
_NC01_SpindleStop	%FX525376	NC channel 01 Signal to confirm spindle stop status
_NC01_SpindleCW	%FX525377	NC channel 01 Signal to confirm spindle CW status
_NC01_SpindleCCW	%FX525378	NC channel 01 Signal to confirm spindle CCW status
_NC01_SpindleOrient	%FX525379	NC channel 01 Spindle Signal to confirm Orientation status
_NC01_SpindleCVelAgr	%FX525380	NC channel 01 Signal to confirm the status of reaching spindle command velocity
_NC01_SpindleZeroVel	%FX525381	NC channel 01 Signal to confirm the status of reaching zero velocity of the spindle
_NC01_MainSpindle	%FW32840	NC channel 01 Confirm the main spindle axis number
_NC01_McodeStrobe	%FX526080	NC channel 01 M code output Strobe signal
_NC01_McodeData	%FD16441	NC channel 01 M Code Data output
_NC01_ScodeStrobe	%FX526144	NC channel 01 S code output Strobe signal
_NC01_ScodeData	%FD16443	NC channel 01 S Code Data output

(2) NC S channel flag (Described based on NC channel 1).

Variable	Memory	Description
_NC01S_Ready	%FX569344	NC channel 01 Axis S axis Ready
_NC01S_Warning	%FX569345	NC channel 01 Axis S Warning occurrence status
_NC01S_Alarm	%FX569346	NC channel 01 Axis S Alarm occurrence status
_NC01S_ServoOn	%FX569347	NC channel 01 Axis S Servo On/Off Status
_NC01S_ServoReady	%FX569348	NC Ch. 01 axis S servo ready
_NC01S_ServoAlarm	%FX569349	NC Ch. 01 axis X servo alarm occurrence
_NC01S_OprRdy	%FX569408	NC channel 01 Axis S Axis Operation Ready Status
_NC01S_LinkedAxNum	%FW35589	NC channel 01 Axis S Actual Axis Number of IPR Axis
_NC01S_Busy	%FX569472	NC channel 01 Axis Signal during Axis S traverse
_NC01S_Direction	%FX569473	NC channel 01 Axis S operation direction
_NC01S_ForwardRun	%FX569474	NC channel 01 Traversing axes toward the axis S +
_NC01S_ReverseRun	%FX569475	NC channel 01 Traversing axes toward the axis S -
_NC01S_SpindleRun	%FX569478	NC channel 01 Axis S spindle operation
_NC01S_HomeCmpl	%FX569539	NC channel 01 Axis S homing completion
_NC01S_CmdPosInWC	%FL8901	NC channel 01 Axis S Command position of Work-piece Coordinate system
_NC01S_ActualVel	%FL8903	NC channel 01 Axis S Actual Velocity of Traverse
_NC01S_CmdPosInMC	%FL8915	NC channel 01 Axis S Command position (Machine Coordinate system)
_NC01S_ActualPosInMC	%FL8917	NC channel 01 Axis S Current position (Machine Coordinate system)
_NC01S_AxErr	%FW35676	NC channel 01 Axis S Error Code Number
_NC01S_DrvErr	%FW35677	NC channel 01 Axis S Drive Error code Number

9.5.6 Spindle Related Commands

The NC control of a motion controller sets information of the current speed of the NC channel spindle axis and provides individual commands that support spindle operations such as gear conversion of the spindle axis.

(1) Spindle operation control (NC_SpindleControl)

1) operation(Described based on NC channel 1).

If the current speed of the spindle cannot be confirmed in the NC function module because there is no encoder in the spindle axis or the speed information of the spindle axis is not registered, users use the function to transmit the speed status of the spindle axis to the NC function module.

(a) Input of ‘Reaching Spindle Command Velocity (TgtVelReached)’

If the spindle axis is automatically operated in the NC function module, the ‘Signal to Confirm the Status of Reaching Spindle Command Velocity’ (_NC01_SpindleCVelAgr) turns On if the relevant input is On after the spindle axis starts operation at the target speed with M03 or M04.

(b) Input of ‘Reaching the Zero Velocity of the Spindle (ZeroVelReached)’

If the spindle axis is automatically operated in the NC function module, the ‘Signal to Confirm the Status of Reaching the Zero Velocity of the Spindle’ (_NC01_SpindleZeroVel) turns On if the relevant input is On after M03 or M04 block with the target speed of 0 is executed or M05 block is executed.

2) Function Block

The spindle operation control (NC_SpindleControl) function block is as follows:

Motion Function Block

```
                 NC_SpindleControl
BOOL ─ Enable                             Enabled ─ BOOL
UINT ─ NcChannel --------------------  NcChannel ─ UINT
BOOL ─ TgtVelReached                         Busy ─ BOOL
BOOL ─ ZeroVelReached                       Error ─ BOOL
BOOL ─ SS_Control                         ErrorID ─ WORD
```

Input-Output		
UINT	NcChannel	Designate the NC channel to give commands
Input		
BOOL	Enable	While an input is enabled, conduct the specified operation on the main spindle in the relevant channel.
BOOL	TgtVelReached	Convey whether the main spindle reaches the target speed to the NC control. 0: The target speed is not reached 1: The target speed is reached
BOOL	ZeroVelReached	Convey whether the main spindle reaches the zero speed to the NC control. 0: The zero speed is not reached 1: The zero speed is reached
BOOL	SS_Control	Start the SS control mode of the main spindle. (future support) 0: Start SS controls 1: End SS controls
Output		
BOOL	Enabled	Indicate that the function block is performed.
BOOL	Busy	Indicate that execution of the function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the error number that occurred while the function block is running.

For function block operation, see ‘6.8.22 Spindle Operation Control (NC_SpindleControl)’.

(2) Spindle gear change (NC_ChangeSpindleGear)

1) Operation

The function is used to change parameter values related to gear conversion and to change the velocity of the spindle axis at the speed that gear conversion can be conducted in order to change gears connected to the NC channel spindle axis.

(a) 'Setting of Velocity Values to Change' (ChangeVelocity) Input

If the spindle gear conversion command is executed, the speed of the spindle axis is changed to the value set in ChangeVelocity and the GearChangeEnable output turns On.

Users operate sequence programs to conduct gear conversion by confirming the GearChangeEnable output.

(b) 'Signal to Complete Gear Conversion' (GearChangeCmpl) Input

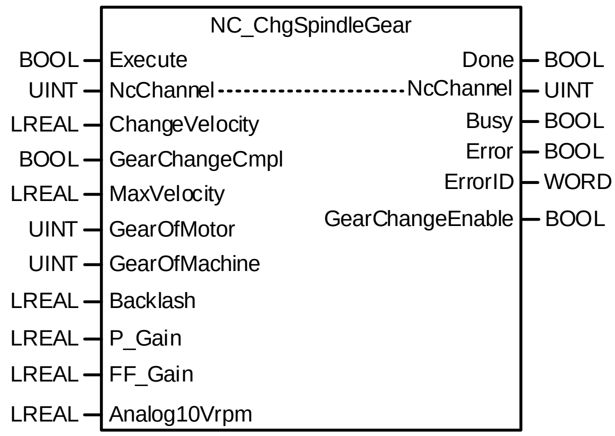
After the speed of the spindle axis is changed to the ChangeVelocity speed, users confirm if the GearChangeEnable output is On and input On in 'Signal to Complete Gear Conversion' (GearChangeCmpl) after executing gear conversion.

If the 'Signal to Complete Gear Conversion' (GearChangeCmpl) is On, set values of the following items set in the function block input to parameters and operate the spindle with the changed setting.

a) Speed limit values (MaxVelocity)
b) Gear ratio of a motor (GearOfMotor)
c) Gear ratio of machine (GearOfMachine)
d) Backlash compensation amount (Backlash)
e) P gain in a position mode (P_Gain)
f) Feed Forward gain in a position mode (FF_Gain)

2) Function Block

The spindle gear conversion (NC_ChangeSpindleGear) function block is as follows:

Motion Function Block		
		
Input-Output		
UINT	NcChannel	Designate the NC channel to give commands
Input		
BOOL	Execute	Give the spindle gear conversion command in the rising Edge of input.
LREAL	ChangeVelocity	Set the speed value to change
BOOL	GearChangeCmpl	The signal that gear changes have been completed. Each input value is set to the relevant parameter after this input turns On
LREAL	MaxVelocity	The maximum setting value for velocity parameters
UINT	GearOfMotor	The gear ratio parameter of a motor
UINT	GearOfMachine	The gear ratio parameter of machine
LREAL	Backlash	Backlash compensation amount (Backlash)
LREAL	P_Gain	P gain setting value
LREAL	FF_Gain	Feed forward gain setting value
LREAL	Analog10Vrpm	Disabled
Output		
BOOL	Done	Indicate that the function block is successfully applied.
BOOL	Busy	Indicate that execution of the function block is not completed.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Output the error number that occurred while the function block is running.
BOOL	GearChangeEnable	Indicate whether to change gears

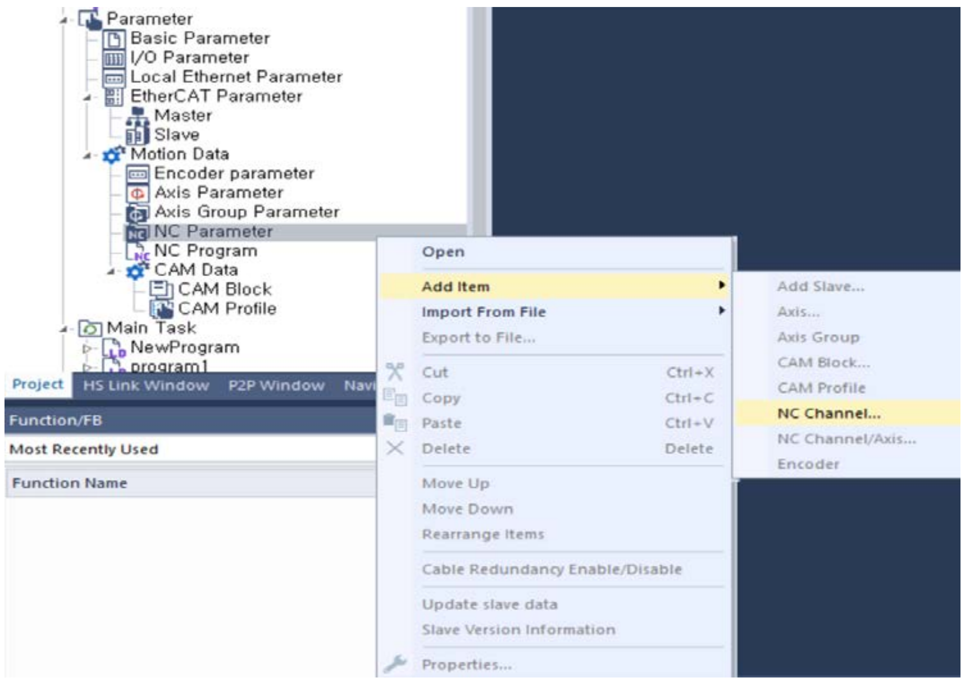
※ For function block operation, see '6.8.24 NC Spindle Operation Control (NC_ChgSpindleGear)'.

9.6 NC 2 channel function

Two NC channels operating independently of each other can be operated simultaneously. The two NC channels can specify the same or different NC programs and can be executed separately.

9.6.1 How to add NC channel 2

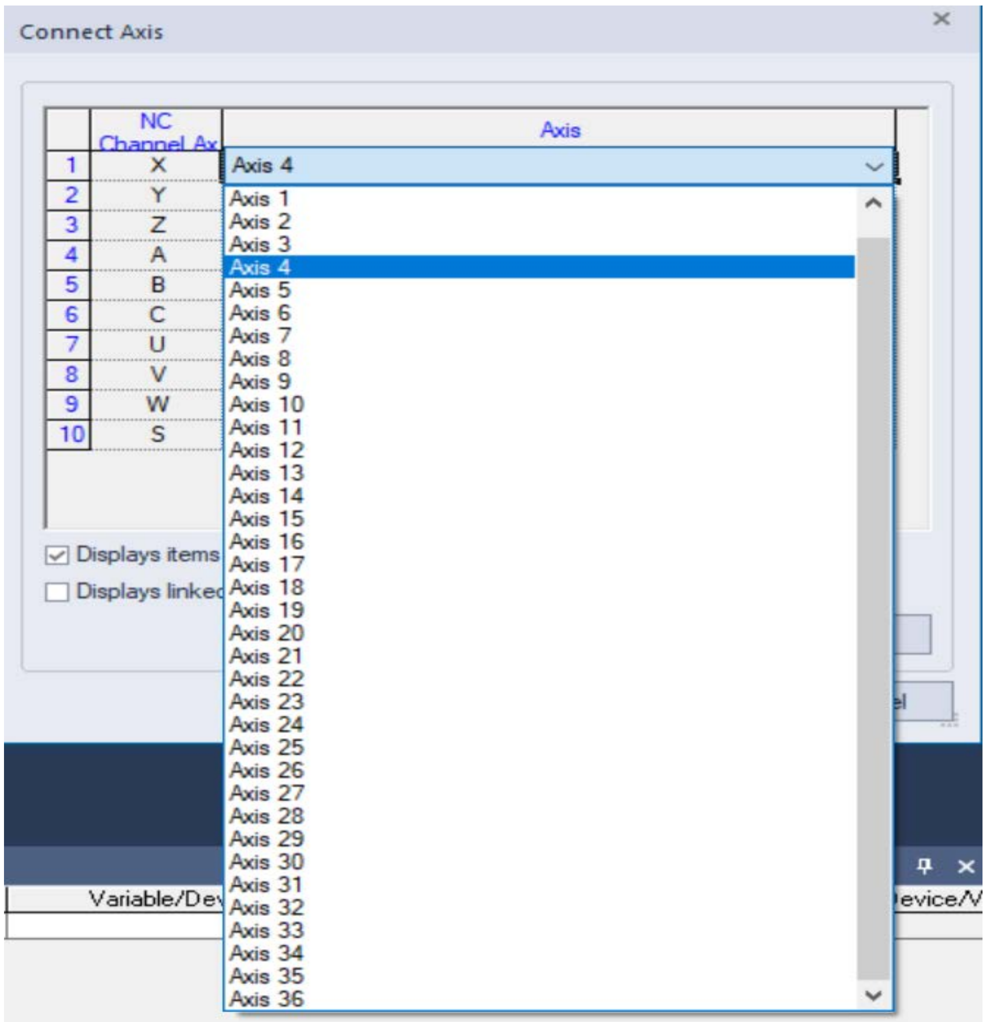
With the existing NC channel 1 added, as shown in the figure below, 'Add Item - NC Channel… Click ' to add NC channel 2.



9.6.2 NC channel/axis connection of NC channel 2

Axis set in NC channel 1 cannot be duplicated in NC channel 2. Please set the axis not set in NC channel 1 to NC channel 2.

Because NC channels are independent, the channel axes of NC channel 1 (X, Y, Z, …, S) and the channel axes of NC channel 2 (X, Y, Z, …, S) have the same name but behave independently.

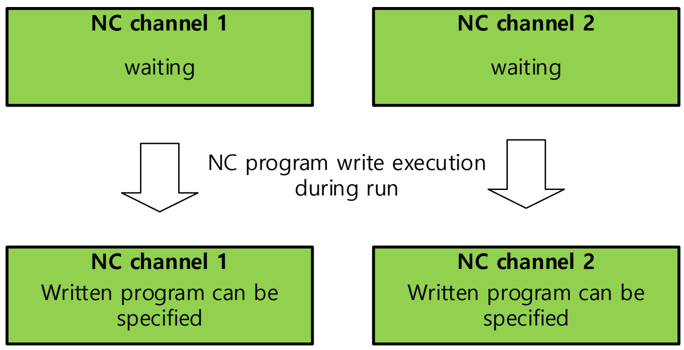


9.6.3 Operation when writing NC program during run

Describes the operation when NC program write is executed during run depending on whether the NC channel program is executed or not.

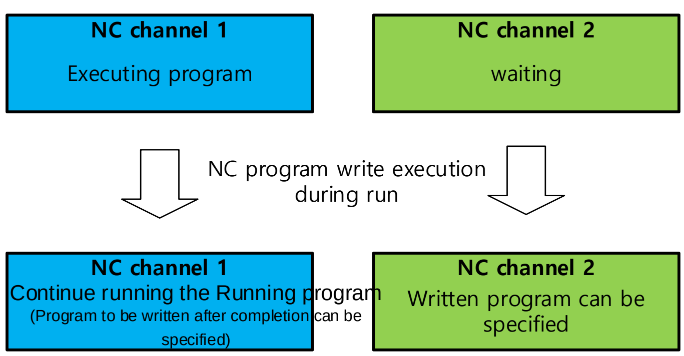
(1) If neither channel is running a program

If the NC program write function is executed during the run while neither channel is executing a program, the latest contents entered in XG5000 are reflected in the internal NC program storage.



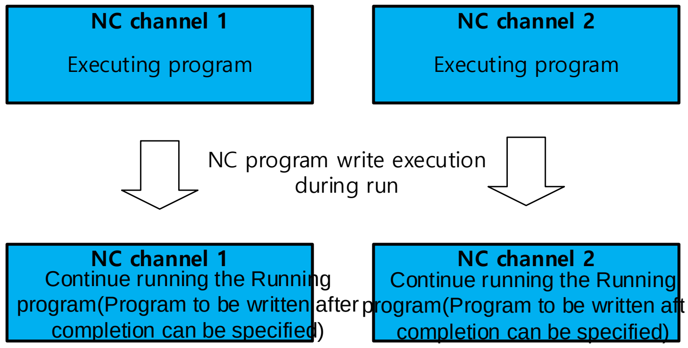
(2) When a program is running on one channel

If the NC program write function is executed during the run while the program is being executed in one channel, the latest content entered in XG5000 is reflected in the internal NC program storage, but the program of the currently running channel is operated unchanged. You can assign NC program (NC_LoadProgram) the latest contents to a channel that is not running. Even if the program with the same name as the running program in the running channel is modified, the previously running program is not changed.



(3) If both channels are running programs

If the NC program write function is executed during the run while both channels are running programs, the latest contents entered in XG5000 are reflected in the internal NC program storage, but the programs of the two currently running channels are operated unchanged. You can designate the latest contents of the NC program (NC_LoadProgram) from the channel where the operation is completed to the channel that is not running. Even if the program with the same name as the running program in the running channel is modified, the previously running program is not changed.



NC program during run depending on whether NC channel program is executed or not

9.6.4 NC 2 Channel Precautions

Describes the precautions for NC 2 channel.

(1) When using 'PLC Erase' function in XG5000 V4.60 or lower version

In XG5000 V4.60 or lower version, if NC parameters are erased by 'PLC Clear' function, even if NC parameters are written again, 0x3100 (NC channel parameter data is abnormal), 0x3200 (NC channel/axis) The parameter data is abnormal.) error may still occur. In this case, the error can be resolved by writing NC parameters using XG5000 V4.70 or higher version.

Chapter 10 CPU Function

10.1 Task Design

10.1.1 Task Overview

There are 3 types of motion control tasks: main task, periodic task and initialization task.

Types of Tasks	Number of Programs	Motions
Main task	Up to 256	• It performs I/O refresh, processing of programs assigned to main task and motion control. • It performs the above tasks at a time for each of the established control period (main task cycle). • It has higher priority than periodic task. • It uses programs that require synchronized control and high-speed operation processing through allocation since it is possible to process program fast. • Period possible to be set: 1ms, 2ms, 4ms
Periodic task		• It performs processing of programs assigned to main task. • It is performed for the remaining time after implementation of main task operation within the control period, and can be performed over multiple cycles. • Since it has lower priority than main task in the execution of motion control commands within main task program, the motion control commands executed in the main task program are processed first. • It uses programs of processing other monitoring data and control of device that doesn't require high-speed processing through allocation. • Period possible to be set: 1ms ~ 100ms (Set to a multiple of the main task cycle)
Initialization task		• It performs processing of programs assigned to the initialization task after implementing I/O refresh. • It is performed only once at the time of entering the RUN mode. • It is executed first when entering RUN mode. If the initial task completion (_INIT_DONE) flag is set by the initialization task program, the task is completed, and the execution of the main task and periodic task program starts.

(1) Main task and periodic task

Both the main task and the periodic task are executed at fixed intervals. The interval at which the main task and periodic task are executed is called the 'task cycle'. The main task can be set in cycles of 500㎲, 1㎳, 2㎳ and 4㎳, and the periodic task can be set in multiple of the main task cycle. However, the periodic task cycle can be set in ㎳ unit.

In the task, 1 to 256 programs can be used. The programs are executed in the order in which they are assigned.

I/O refresh refers to the exchange of data between the digital I/O module and the analog module. The I/O refresh is performed at the beginning of the main task execution per cycle.

(2) Task partitioning

All programs should be assigned to one task. Users are required to assign the task according to the characteristics of the created program by referring to the table below.

Tasks	Appropriate programs
Main task	• The execution cycle of I/O refresh should be strictly observed. • Highest execution priority • High-priority motion control is included.
Periodic task	• Program that can be operated regardless of I/O refresh • Program that has lower execution priority than the main task and requires periodic execution
Initialization task	• Execution is required prior to the main task program execution during the RUN operation •Device initialization and initial value setting program

10.1.2 Task Specification

The specifications of the tasks are as follows.

Items	Specifications
Types of tasks	• Main task • Periodic task • Initialization task
Number of task programs	• Up to 256
Main task cycle	• 500 ㎲, 1 ㎳, 2 ㎳, 4 ㎳
Periodic task cycle	• Can be set to a multiple of the main task from 1 to 100 ㎳
Initialization task cycle	• Same as the main task cycle

Periodic task cycle that can be set depending on the main task cycle

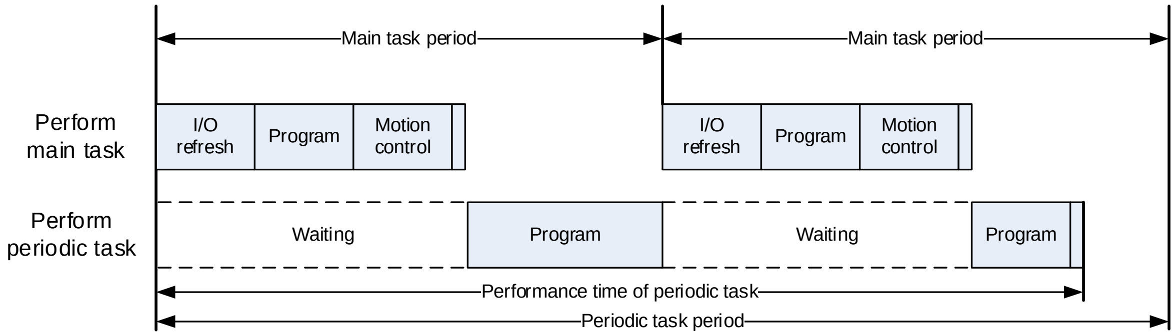
Main task cycle	Periodic task cycle that can be set
500 ㎲	1㎳, 2㎳, 3㎳, 4㎳, 5㎳, … 95㎳, 96㎳, 97㎳, 98㎳, 99㎳, 100㎳
1 ㎳	1㎳, 2㎳, 3㎳, 4㎳, 5㎳, … 95㎳, 96㎳, 97㎳, 98㎳, 99㎳, 100㎳
2 ㎳	2㎳, 4㎳, 6㎳, 8㎳, 10㎳, … 92㎳, 94㎳, 96㎳, 98㎳, 100㎳
4 ㎳	4㎳, 8㎳, 12㎳, 16㎳, 20㎳, … 84㎳, 88㎳, 92㎳, 96㎳, 100㎳

10.1.3 Basic Operation of Task

Several tasks of XMC-E32A cannot be executed at the same time. Each task is executed according to its priority, and the main task program has higher priority than the periodic task program.

If the main task program execution cycle is reached during the execution of the periodic task program, the main task program is executed. Therefore, while the main task program is executed in accordance with the cycle, the periodic task program can be executed in several main task cycles. If you use the periodic task program, you should write the program by referring to this point.

In addition, the cycle should be set so that the task program execution can be completed within the set period. If the task exceeds the set period, a warning is issued. If the task execution is not completed until the periodic error detection time, the system switches to the ERROR state.

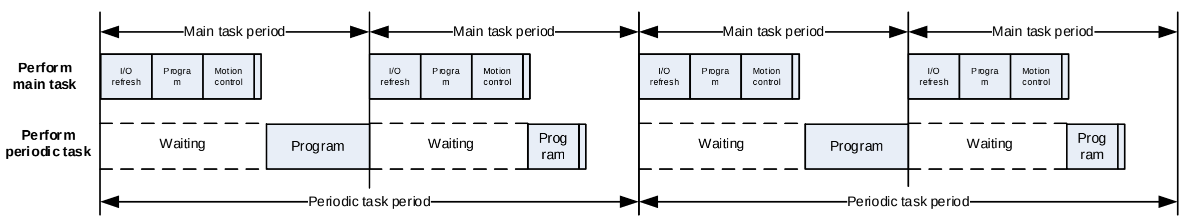


In the main task execution, the double line display after the program execution the motion control or periodic task execution indicates that the task execution is completed.

(1) Cycle of main task and periodic task

The main task and periodic task are the ones that are executed repeatedly in cycles. Both tasks have a task execution cycle, and the periodic task cycle can be set to a multiple of the main task cycle.

For example, if the main task cycle is 1㎳, and the periodic task cycle is 2㎳, the periodic task is executed every time the main task is executed twice.

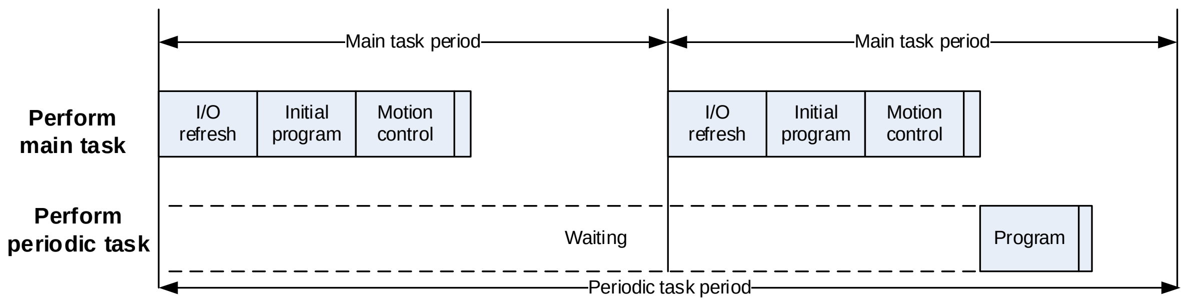


(2) Initialization task

The initialization task is executed until the initialization task execution completion (_INIT_DONE) flag is set before the execution of the main task, and it is terminated when the user sets the _INIT_DONE flag in the program. Only when the initialization task is terminated, the main task and periodic task programs are executed. The initialization task cycle inherits the main task cycle.

Notice
If the initialization task execution completion (_INIT_DONE) flag is set by the user-written initialization task program, the execution of the initialization task program is terminated, and the execution of the main task and periodic task programs are started. The initialization task operates in the main task cycle and is included in the main task execution time.

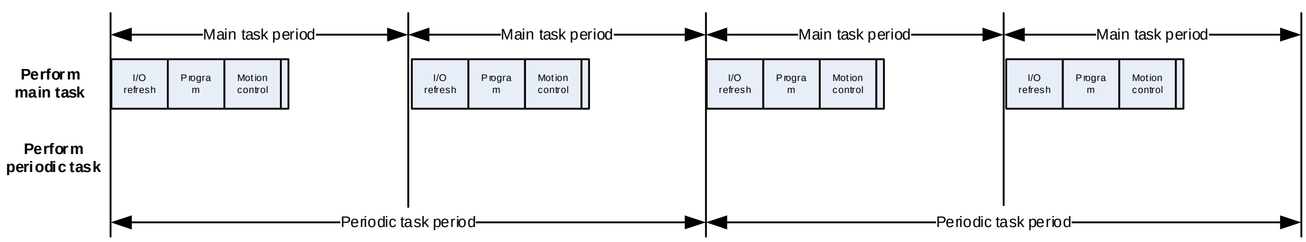
When the initialization program execution is completed, and the initialization task execution is terminated as shown below, the main task program and periodic task program are executed.



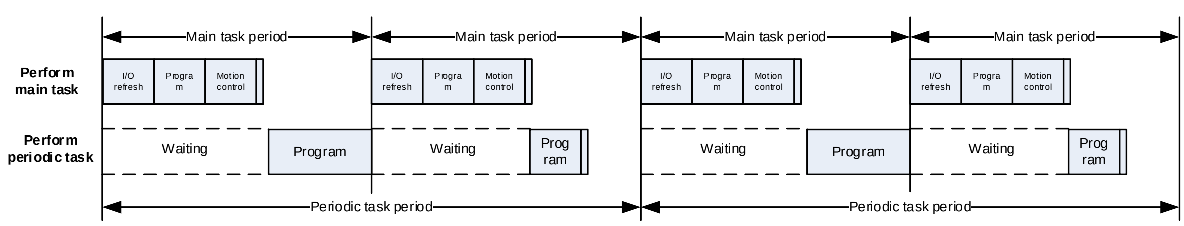
10.1.4 Examples of Task Execution Sequence

Below are descriptions of the execution sequence for the main task and periodic task.

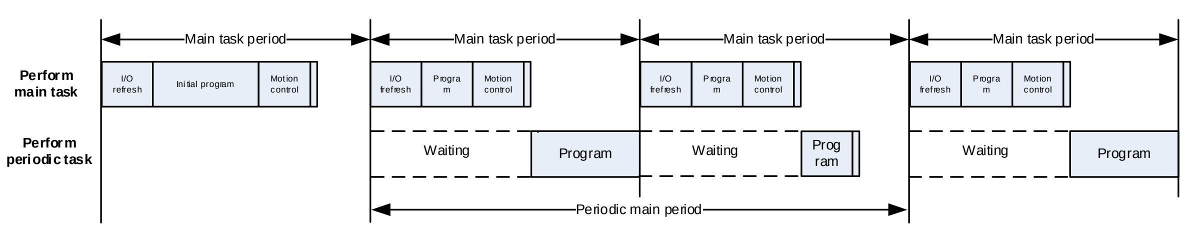
(1) If there is only main task program



(2) If there main/periodic task programs



(3) If there is an initialization task program

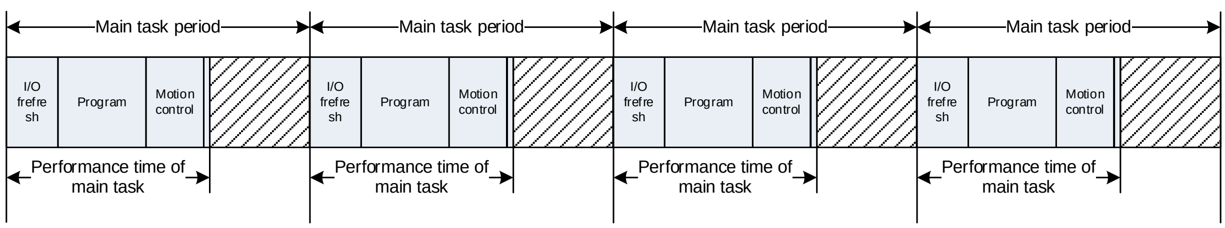


10.1.5 System Service Processing

System service includes the following services.

System Service Names	Contents
USB service	• Processing of service requests in XG5000
Built-in Ethernet port service	• Processing of service requests in XG5000 • Communication (P2P) service processing • FTP service processing
SD memory card service	• SD memory card command execution • Data logging

The system service is executed after the completion of the main task or the periodic task and s run at an idle time when the task is not running. The system service is executed in the shaded section as shown below.

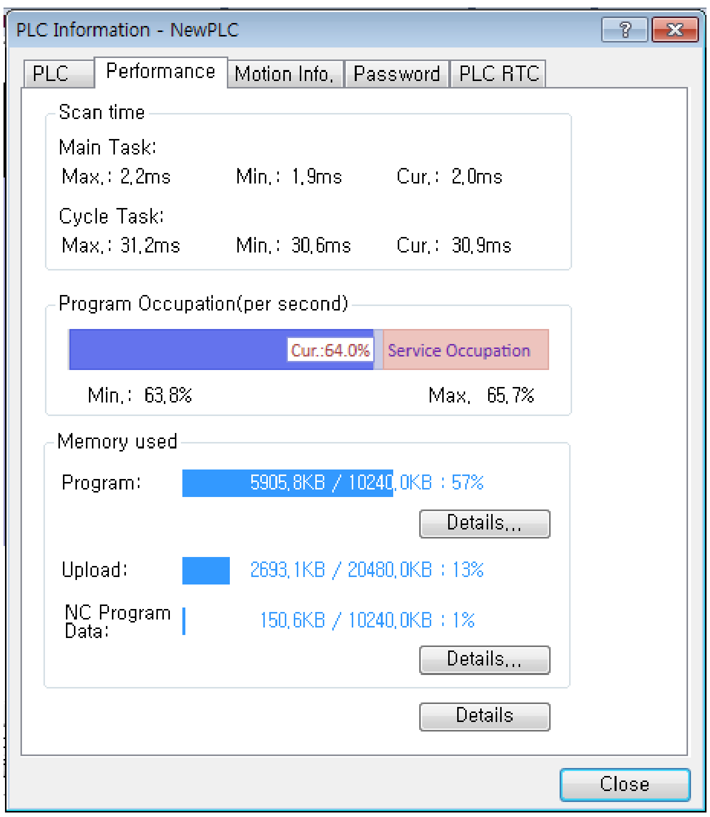


The order of priority of the system service and each task is main task > periodic task > system service, and the main task has the highest priority. When the main task execution cycle is reached while the system service is running, the system service is paused, and the main task is executed. In addition, if the main task execution cycle is reached while the system service is running, the system service is paused, and the periodic task is executed. When the execution of both the main task and the periodic task is completed, the paused system service is executed sub sequentially.

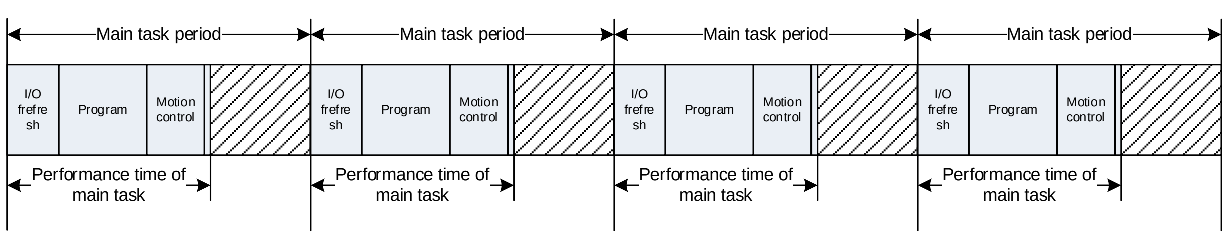
10.1.6 Program Occupancy Rate Operation

Program occupancy rate refers to the ratio of the task execution time per second during the system RUN operation. If there is only main task, the sum of the main task execution time is displayed as a percentage. If there is a periodic task, the main task and periodic task execution time is calculated and displayed as a percentage.

In the figure below, the program occupancy rate is currently 64%, which means that the main task and periodic task are executed about 64% of the time for one second, and the system service is running for the remaining time.

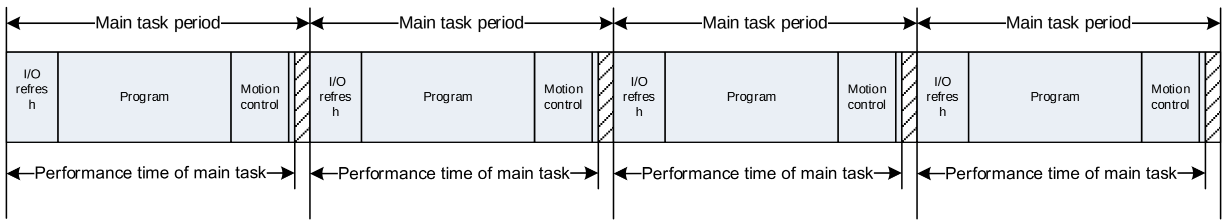


If there is no periodic task as shown below, the system service can be executed in the shaded section if the main task execution is completed.

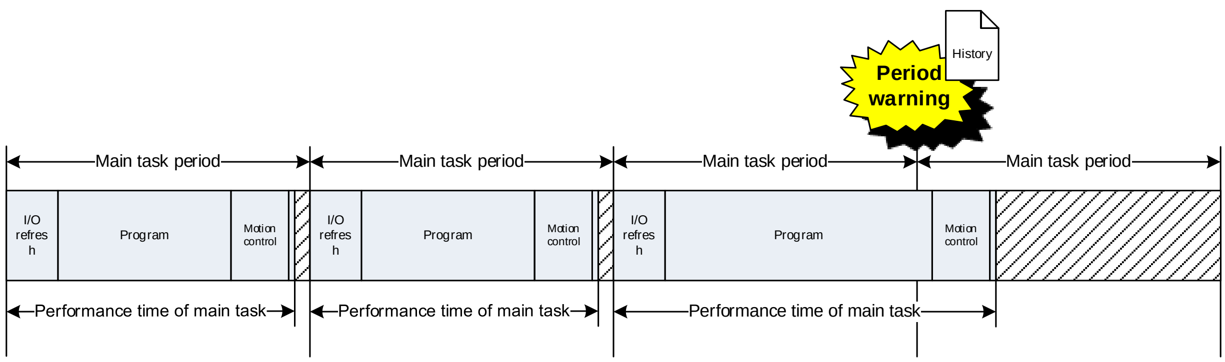


If the program occupancy rate is high (system service occupancy rate is low) as shown below, the system service may not be performed normally. In the basic parameter, a user can set the value ranging from 50 to 95%, and if the set value is exceeded, the task program occupancy rate warning is generated. If the program occupancy rate exceeds 100%, the system enters the ERROR state.

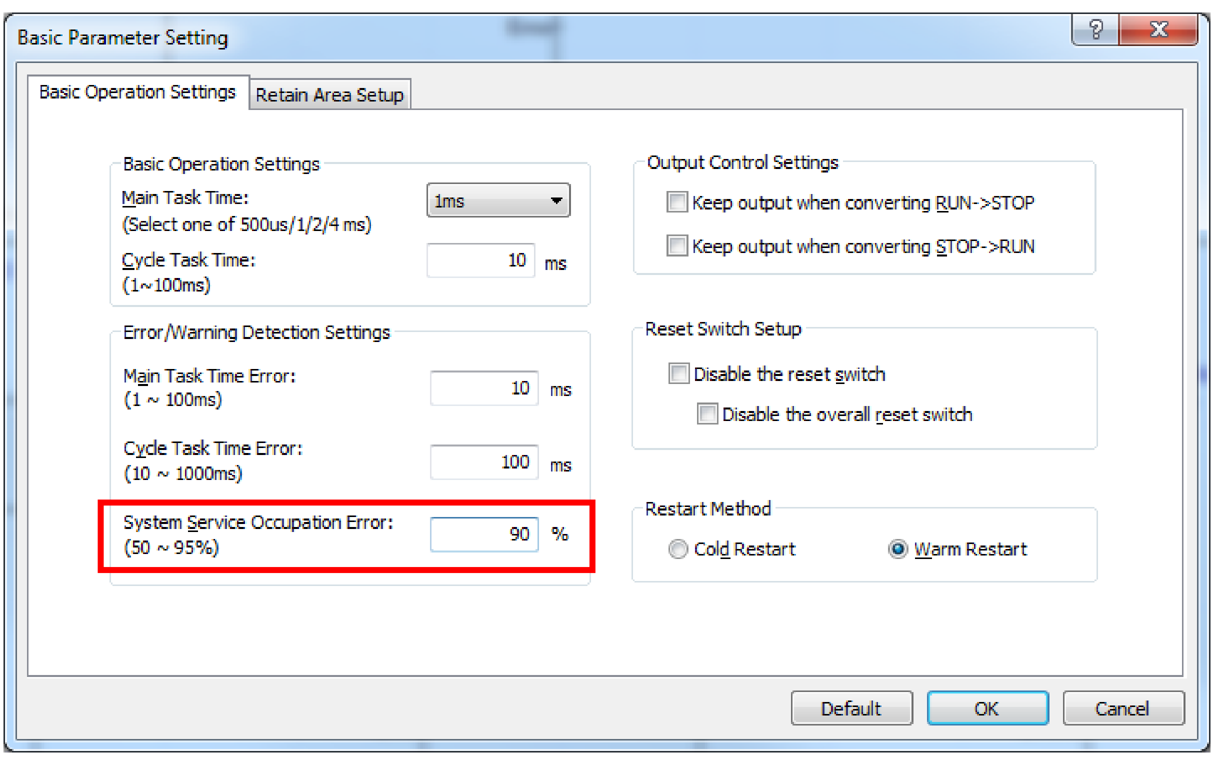
※ Adjust the main task cycle so that the program occupancy rate does not exceed 90%, if possible.



If the cycle warning is generated as shown below, the program occupancy rate may increase.

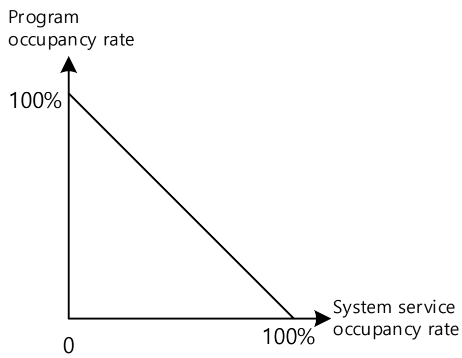


The value for setting the task program occupancy rate excess warning detection can be set from 50% to 95% in the basic parameter settings.



If the task program occupancy rate exceeds the set value, the task occupancy rate excess warning (_TASK_PRM_USAGE_OVER_WAR) is generated. If the task program occupancy rate exceeds 100%, the system state switches from RUN to ERROR, and the task program occupancy rate excess error (_TASK_PRM_USAGE_OVER_ER) is generated.

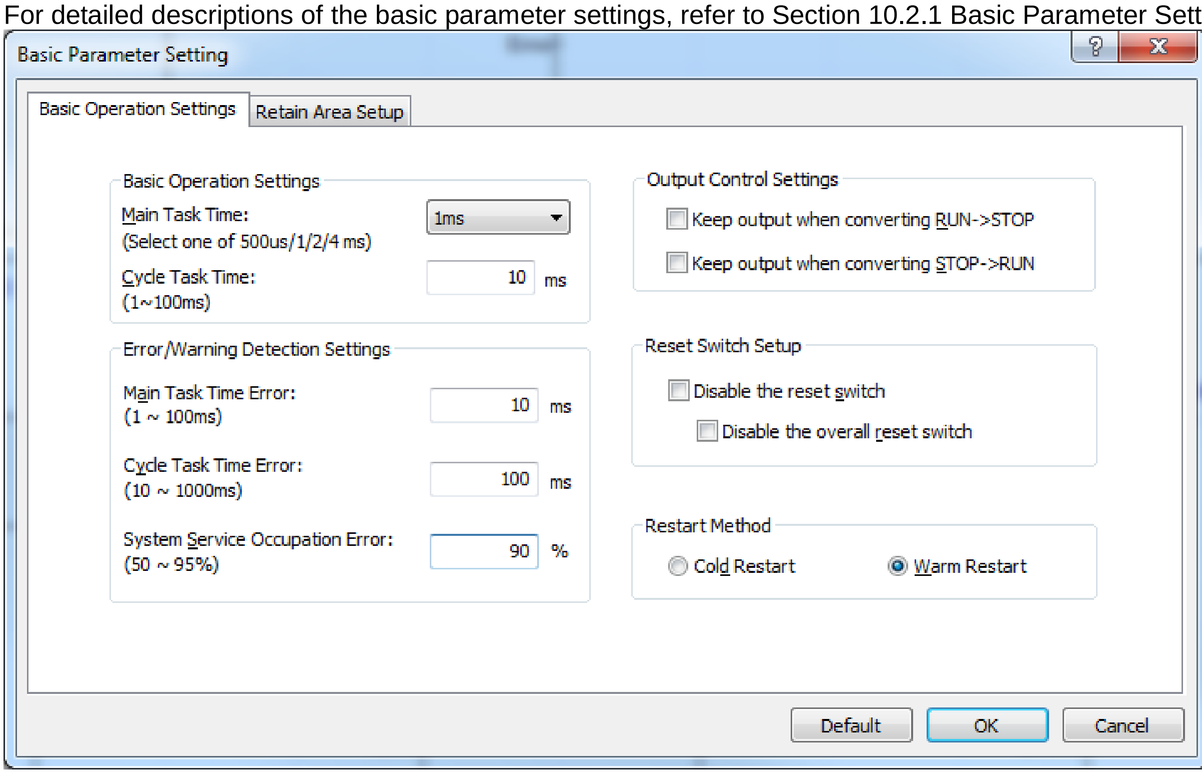
The program occupancy rate is inversely proportional to the system occupancy rate. If the program occupancy rate is 20%, the system occupancy rate is 80%. But if the program occupancy rate is 80%, the system occupancy rate is 20%.



The increase in the task program occupancy rate means that the main task and the periodic task occupy a large portion in one cycle, and thus the time required for the system service execution is reduced. Please make sure that the program occupancy rate does not exceed 95%. If it exceeds 95%, change the main task cycle.

10.1.7 Task Setting Items

To execute the task program, the following task-related items should be set. Each item is reflected immediately when the basic parameter items are transmitted. Even if the periodic cycle is not used, the cycle should be set. For detailed descriptions of the basic parameter settings, refer to Section 10.2.1 Basic Parameter Settings.



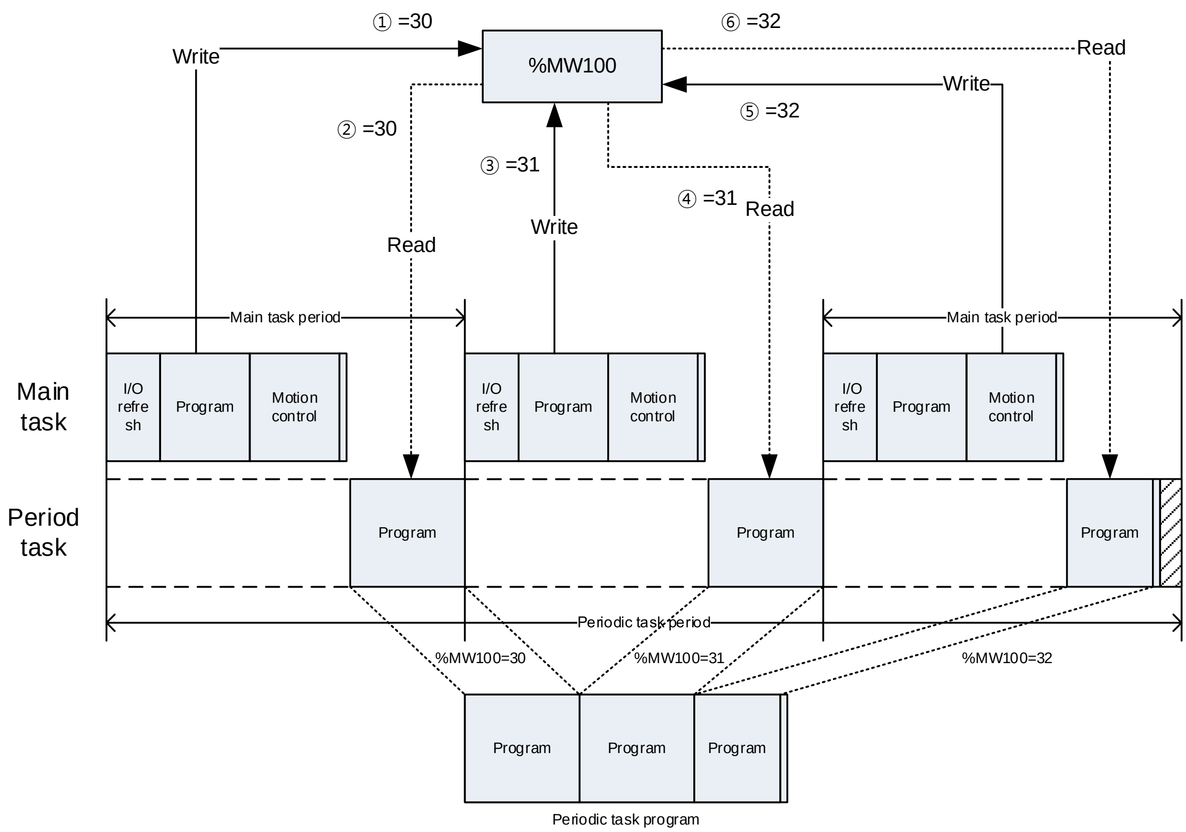
Items	Descriptions	Setting Values	Default
Main task cycle	Sets the time for the main task	500 ㎲, 1 ㎳, 2 ㎳, 4 ㎳	1 ㎳
Periodic task cycle	Sets the time of the periodic task as the multiple of the main task cycle	1~100 ㎳	10 ㎳
Main task cycle error	Sets the main task execution time when the main task is executed beyond the set time	1~100 ㎳	10 ㎳
Periodic task cycle error	Sets the periodic task execution time that causes an error when the periodic task is executed beyond the set time	10~1000 ㎳	100 ㎳
Task program occupancy rate warning	If the task program occupancy rate exceeds the set value because there are many main task programs or periodic task programs, the task program occupancy rate warning is generated. If the task program occupancy rate exceeds 100%, the task program occupancy rate error occurs, and it switches to the ERROR state.	50~95%	90%

10.1.8 Methods on How to Use Variables between Tasks

Extra attention should be given when reading and writing the same global variables in the main task and the periodic task.

If the value of %MW100 is read and written in the main task and periodic task programs as shown below, the value of %MW100 will be changed continuously depending on the usage position in the periodic task.

① Write the value of %MW100 to 30 in the main task program
② The value of %MW100 is 30 when read from the periodic task
③ Write the value of %MW100 to 31 in the main task program
④ The value of %MW100 is 31 when read from the periodic task
⑤ Write the value of %MW100 to 32 in the main task program
⑥ The value of %MW100 is 32 when read from the periodic task



Since the value of %MW100 is continuously changed to 30, 31 and 32 in one cycle of the periodic task, the value may be different depending on the location of the device use. If in the periodic task program, the value is written in the global variable (example: %MW100) used in the main task program, the result of the main task program operation may be affected.

※ Please be careful when programming to avoid using the same device between the main task program and the periodic task program, if possible.

10.1.9 Task Flags

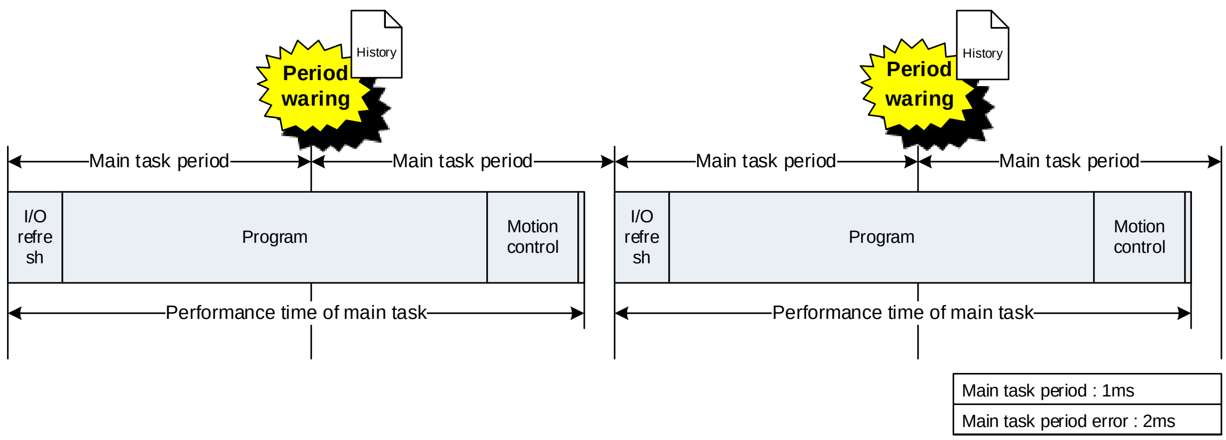
Below are descriptions of the task flags.

Flag name	Type	Device	Description
_PROGRAM_RATIO_MAX	UINT	%FW518	User program maximum execution occupancy (1sec)
_PROGRAM_RATIO_MIN	UINT	%FW519	User program minimum execution occupancy (1sec)
_PROGRAM_RATIO_CUR	UINT	%FW520	User program current execution occupancy (1sec)
_PTASK_SCAN_WR	BOOL	%FX20486	Main task scan value initialization
_PTASK_CYCLE_WAR_NUM	UINT	%FW748	Main task period exceeded warning count
_PTASK_CYCLE_WAR	BOOL	%FX129	Main task period exceeded warning
_PTASK_SCAN_MAX	UINT	%FW512	Main task max. scan time(Unit:100 us)
_PTASK_SCAN_MIN	UINT	%FW513	Main task min. scan time(Unit:100 us)
_PTASK_SCAN_CUR	UINT	%FW514	Main task current scan time(Unit:100 us)
_CTASK_SCAN_WR	BOOL	%FX20487	Periodic task scan value initialization
_CTASK_CYCLE_WAR_NUM	UINT	%FW749	Periodic task period exceeded warning count
_CTASK_CYCLE_WAR	BOOL	%FX130	Periodic task period exceeded warning
_CTASK_SCAN_MAX	UINT	%FW515	Periodic task max. scan time(Unit:100us)
_CTASK_SCAN_MIN	UINT	%FW516	Periodic task min. scan time(Unit:100us)
_CTASK_SCAN_CUR	UINT	%FW517	Periodic task current scan time(Unit:100us)
_CTASK_CYCLE_ER	BOOL	%FX92	Periodic task period error
_PTASK_CYCLE_ER	BOOL	%FX91	Main task period error
_INIT_DONE	BOOL	%FX20496	Completion of initialization task
_INIT_RUN	BOOL	%FX24	Executing the initial task
_TASK_PRM_USAGE_OVER_WAR	BOOL	%FX135	Task program occupancy excess warning
_TASK_PRM_USAGE_OVER_ER	BOOL	%FX94	Task program occupancy excess error

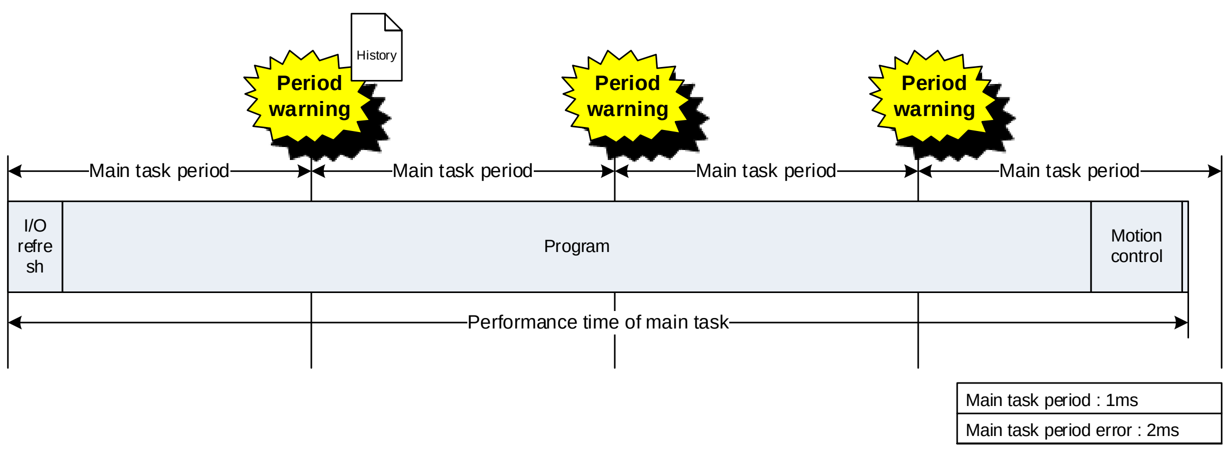
10.1.10 Task-Related Warning/Error

(1) Task cycle over warning

If the main task or the periodic task exceeds the cycle set by a user, the cycle over warning is generated. The warning is stored in the error history.

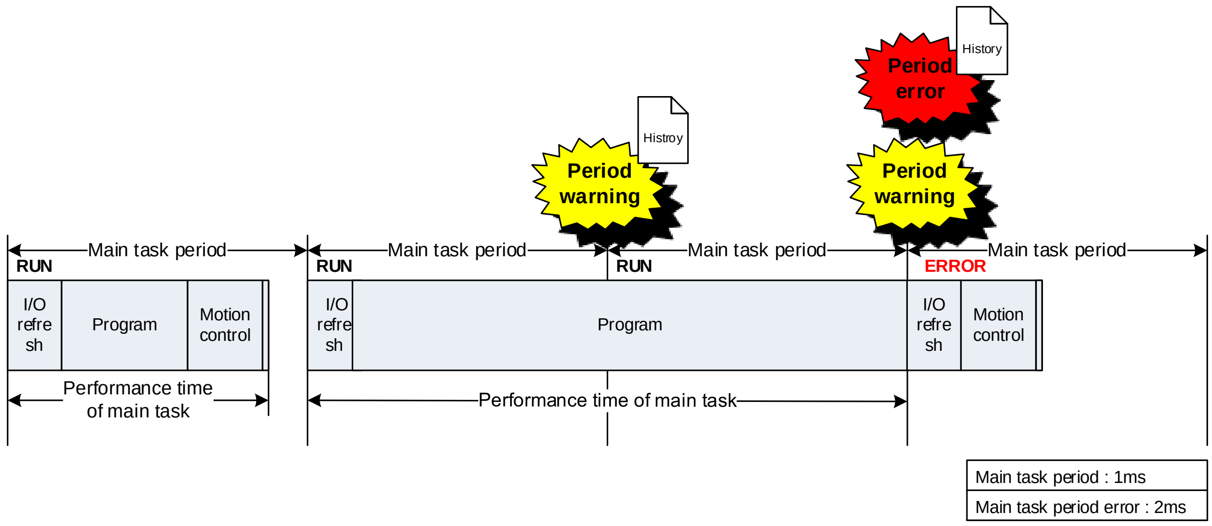


If the task execution is completed in the previous cycle as shown in the figure below, the history is stored. If the cycle over warning is continuously generated, the cycle over warning history is saved only for the first occurrence of the warning. The saved history can be checked in the error history.



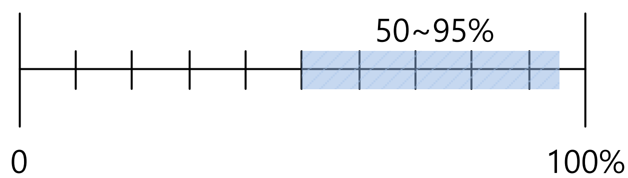
(2) Task cycle error

If the task is executed by exceeding the cycle error time set in the basic parameter, a cycle over error occurs. Refer to Section 0 Task Program Occupancy Rate Excess Warning/Error for corrective actions taken in the case of an error.

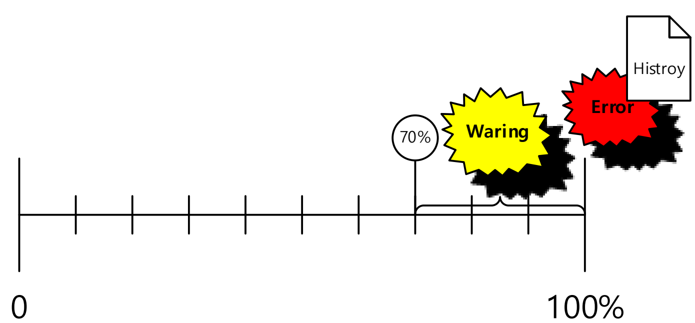


(3) Program occupancy rate excess warning/error

In the basic parameter, the program occupancy rate excess warning detection setting value can be set to 50~95%. If it exceeds the value set by a user, the program occupancy rate excess warning is generated, and if it is 100%, the program occupancy rate excess error occurs.



As shown in the figure below, if the program occupancy rate excess warning detection value of the basic parameter is set to 70%, a warning is generated when the program occupancy rate ranges from 70 to 99%, and an error occurs when it is 100%.

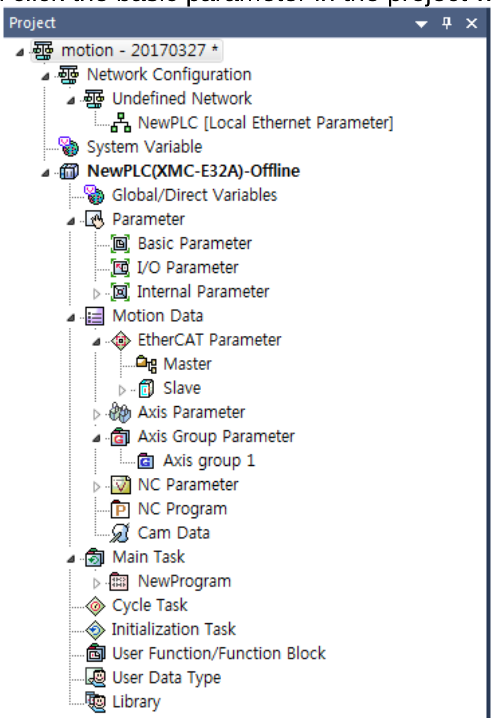


10.2 Parameter Setting

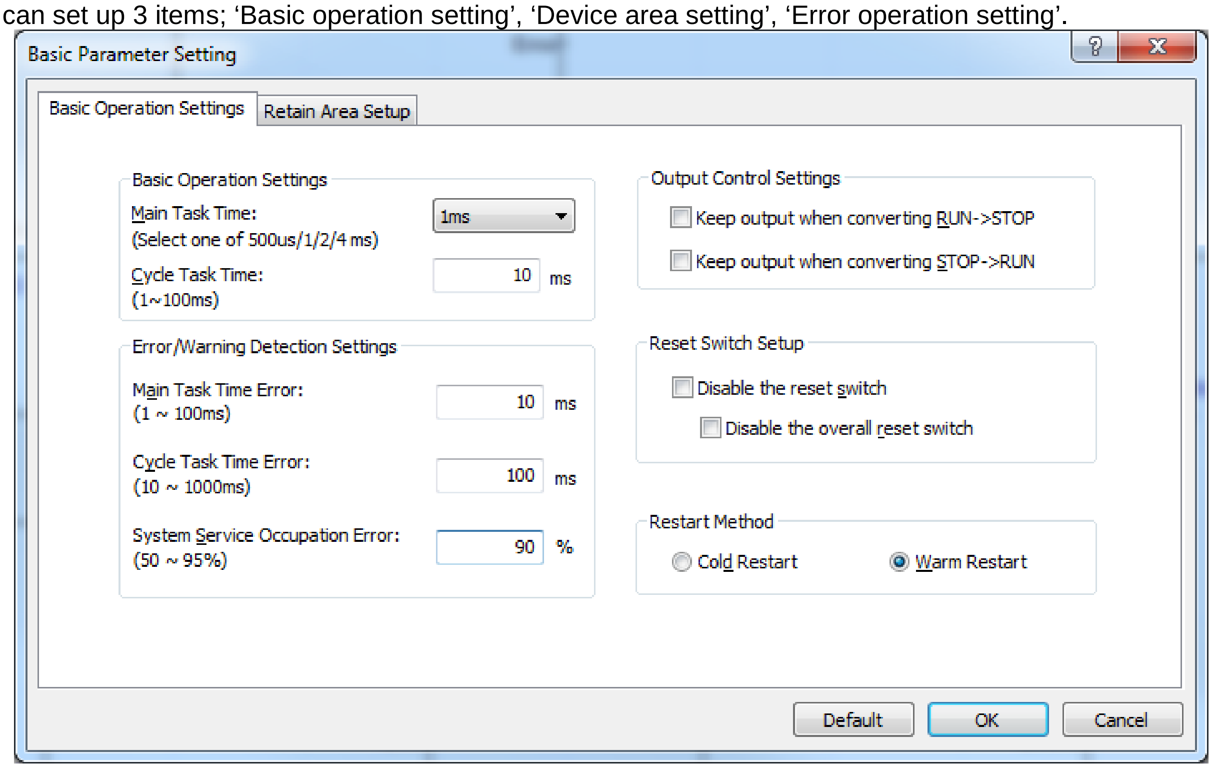
This section describes motion controller’s parameter setting.

10.2.1 Basic Parameter Setting

If you click the basic parameter in the project window, the below screen will be displayed.



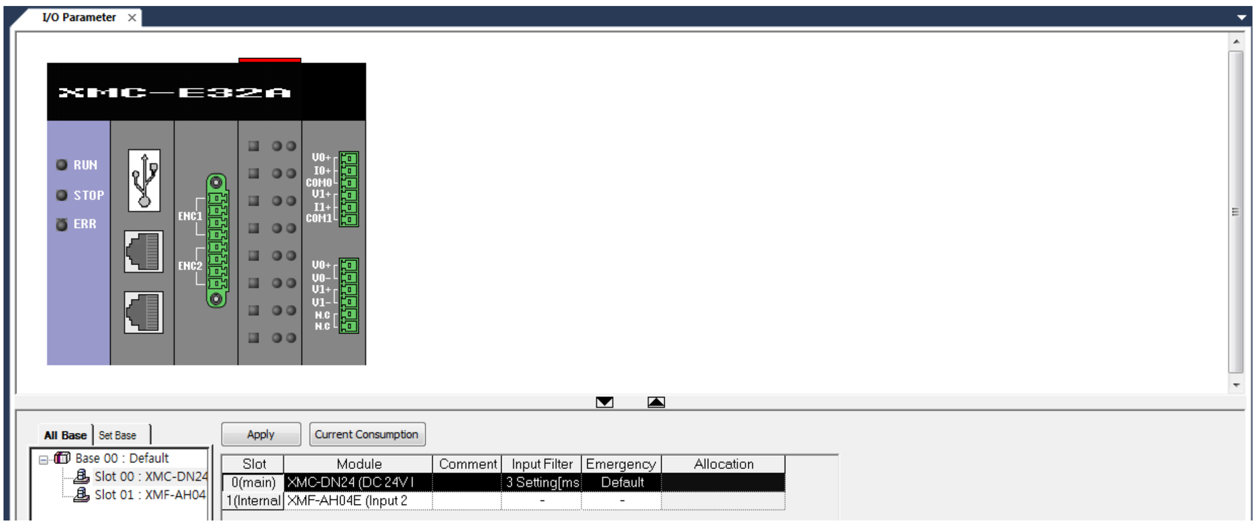
You can set up 3 items; ‘Basic operation setting’, ‘Device area setting’, ‘Error operation setting’.



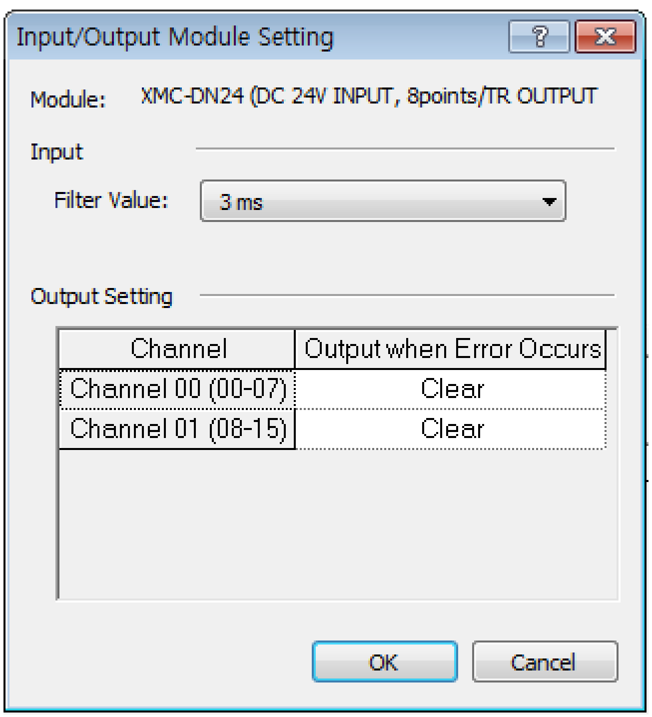
Classification	Items	Descriptions	Setting Values
Basic operations	Main task cycle	Sets the time of the main task	500us, 1ms, 2ms, 4ms
	Periodic task cycle	Sets the time of the periodic task	1~100ms(a multiple of the main task)
	Main task cycle error	Sets the main task execution time that causes an error when the main task is executed beyond the set time	1~100ms
	Periodic task cycle error	Sets the periodic task execution time that causes an error when the periodic task is executed beyond the set time	10~1000ms
	Task program occupancy rate warning	If the task program occupancy rate exceeds the set value because there are many main task programs or periodic task programs, the task program occupancy rate warning is generated. If the task program occupancy rate exceeds 100%, the task program occupancy rate error occurs, and it switches to the ERROR state.	50~95%
	Output maintenance in case of Run->Stop transition	Maintains output when Run->Stop transition is allowed	Allowed/Prohibited
	Output maintenance in case of Stop → Run transition	Maintain output when Stop->Run transition is allowed	Allowed/Prohibited
	Reset switch operation shutdown	Sets whether or not to perform reset operation with the switch on the front panel of the product	Allowed/Prohibited
	Overall Reset switch operation shutdown	Sets whether or not to perform the overall reset operation with the switch on the front panel of the product	Allowed/Prohibited
	Restart mode	Selects restart mode	Cold, Warm
Memory area settings	Retain area settings	Sets the range to retain for the M area	Among %MW0~%MW1048575, 524,288 Word settings

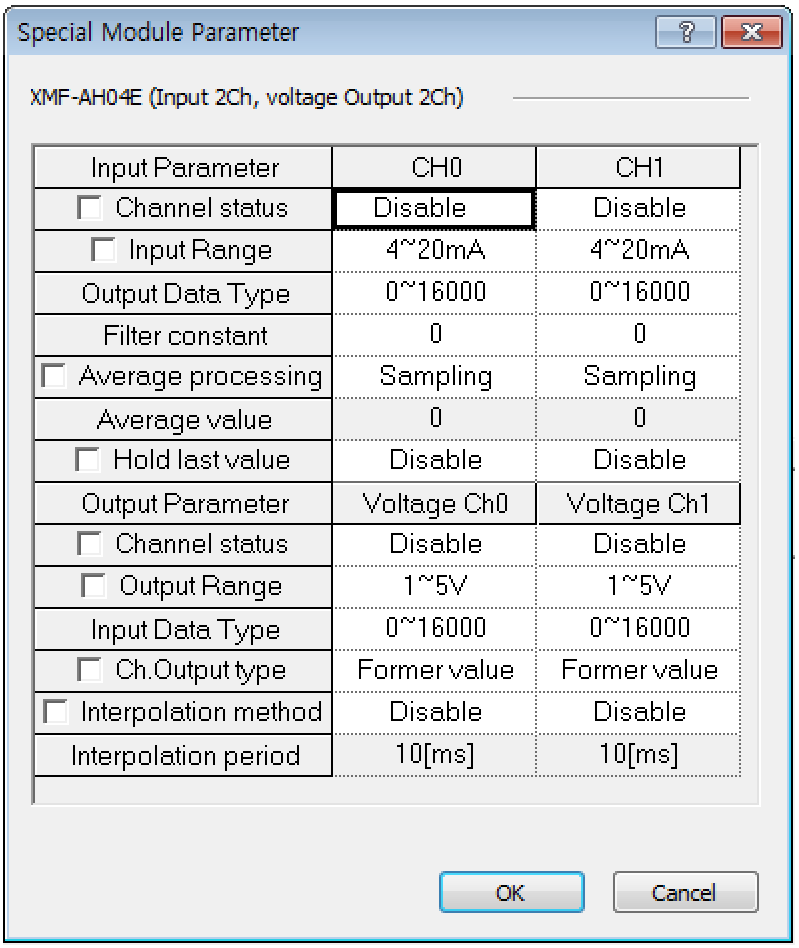
10.2.2 I/O Parameter Setting

It is the function to set up and reserve the information for each I/O. If you click 『I/O Parameter』 in the project window, the below setting window will be displayed.



If you click the 『Module』 in the 『slot』 position, the list of each module will be displayed. Then, choose the module that is matched with the actual system to be configured. The selected slot will be displayed as below.




Special Module Parameter
XMF-AH04E (Input 2Ch, voltage Output 2Ch)
Input Parameter CH0 CH1
Channel status Disable Disable
Input Range 4~20mA 4~20mA
Output Data Type 0~16000 0~16000
Filter constant 0 0
Average processing Sampling Sampling
Average value 0 0
Hold last value Disable Disable
Output Parameter Voltage Ch0 Voltage Ch1
Channel status Disable Disable
Output Range 1~5V 1~5V
Input Data Type 0~16000 0~16000
Ch.Output type Former value Former value
Interpolation method Disable Disable
Interpolation period 10[ms] 10[ms]
OK
Cancel

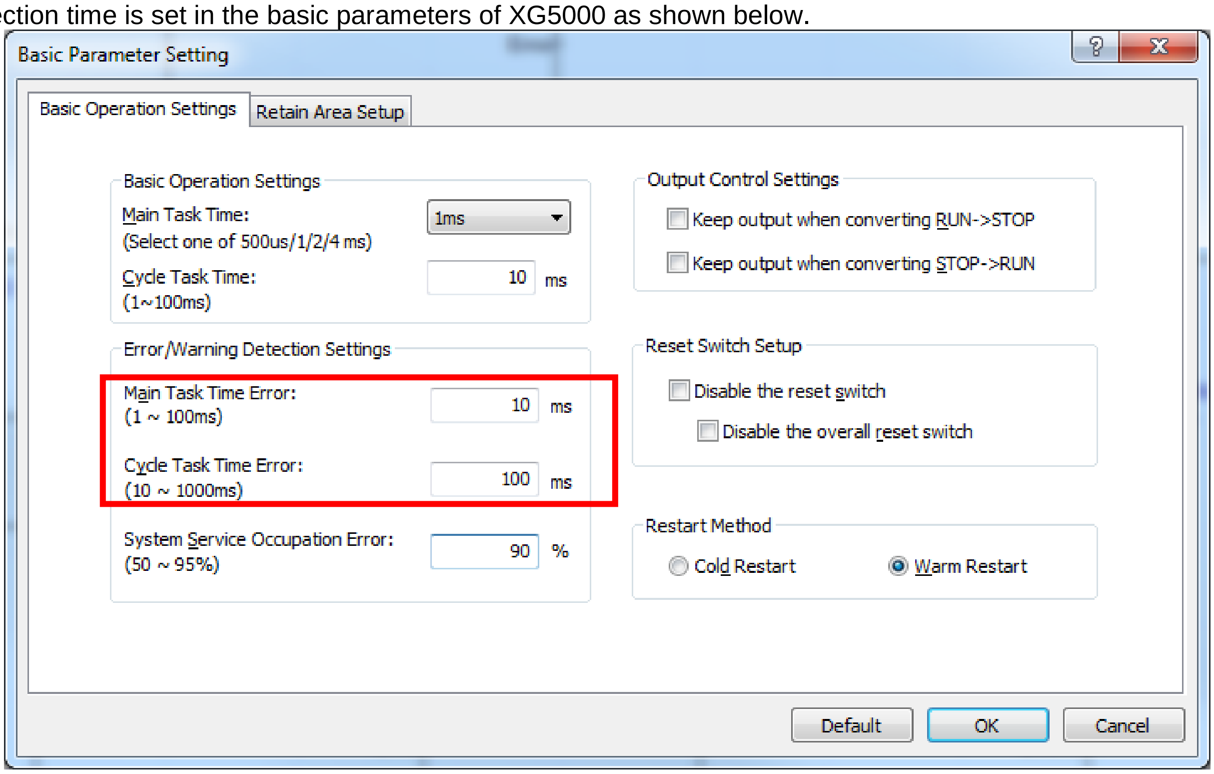
10.3 Self-Diagnosis Function

The Self-Diagnosis function is the function for the CPU part to diagnose the motion controller system for defects. In case errors occur during supplying the power to the motion controller system or during operation, it detects errors to prevent malfunction of the system and preventive maintenance.

10.3.1 Main Task/Periodic Task Cycle Error

Main task/periodic task cycle error is a function to a software error of the motion controller or a periodic error caused by the user program.

(1) It is used to detect that the program is executed for the time that exceeds the user's intended period due to an operation delay caused by the main task/periodic task program error. The main task/periodic task cycle error detection time is set in the basic parameters of XG5000 as shown below.



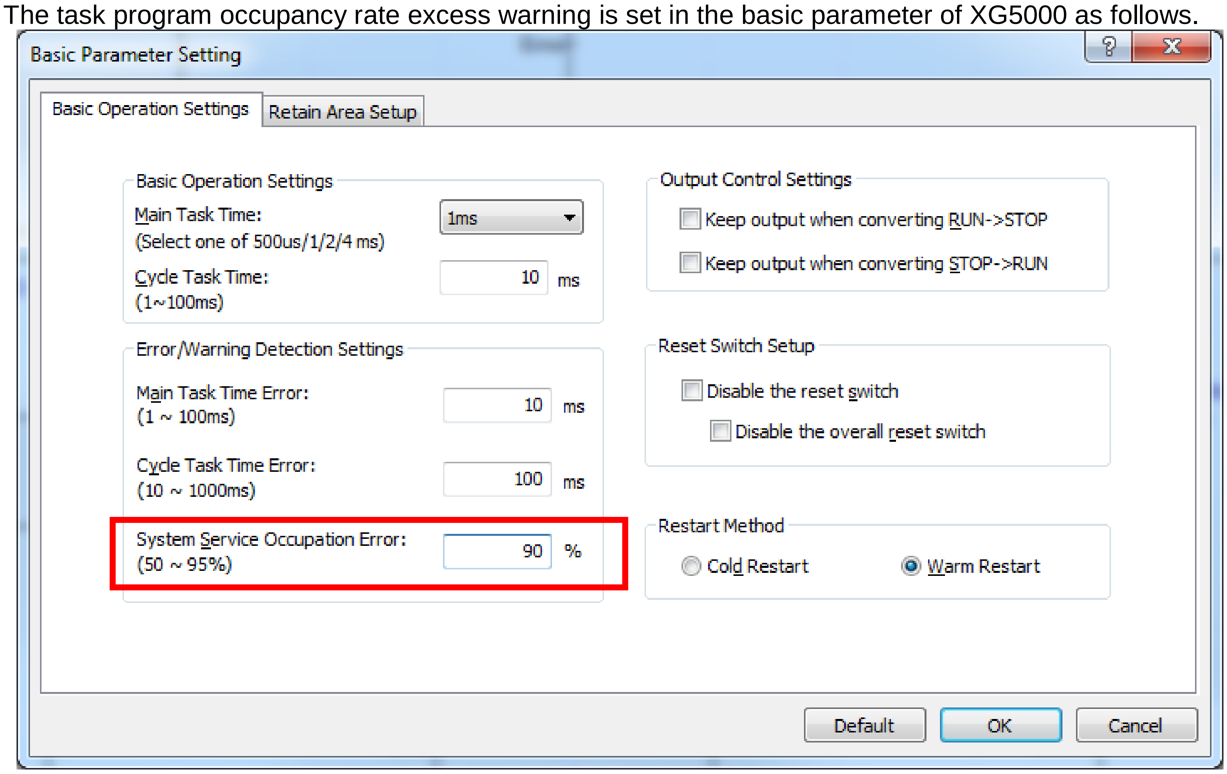
(2) While the program is running, the elapsed scan time is monitored, and if the set detection time is exceeded, the operation of the motion controller is stopped immediately, and an error is generated.

(3) When the main task/periodic task error occurs, the error is cleared if the power is turned on again, or the mode is switched to Stop mode

10.3.2 Task Program Occupancy Rate Excess Warning /Error

If the occupancy rate of the program increases due to the execution of the main/periodic tasks, the system service cannot be executed. To prevent this, this function allows the user to detect the task program occupancy rate excess warning/error. (System service: Services, excluding the main/periodic/ initialization task)

(1) The task program occupancy rate excess warning is set in the basic parameter of XG5000 as follows.



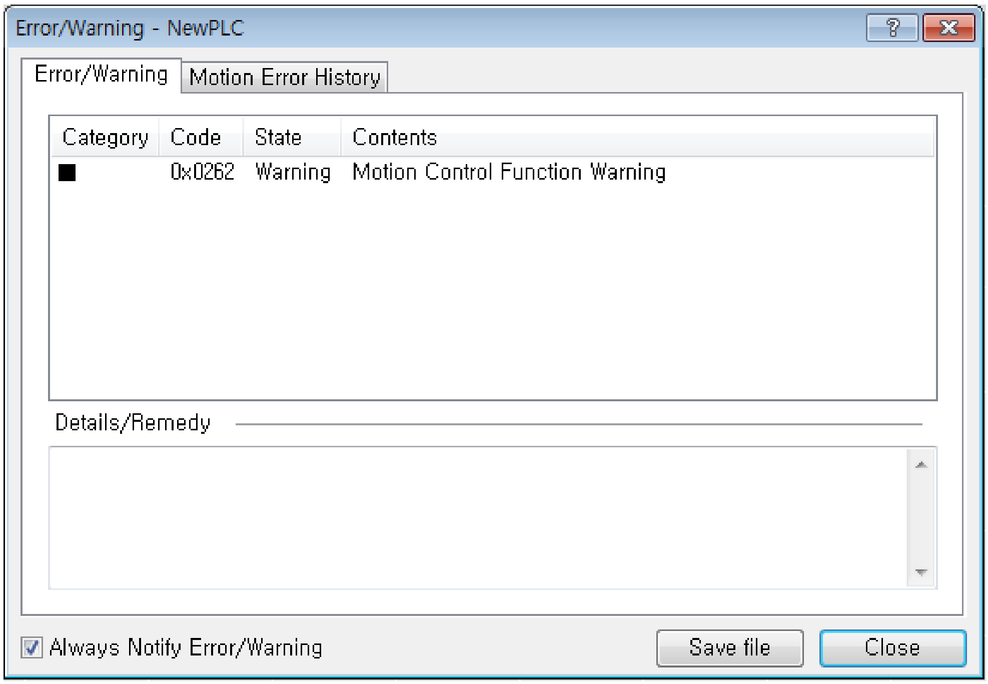
(2) If the set occupancy rate is exceeded while monitoring the task program occupancy rate during the execution of program, the task program occupancy rate excess warning is generated. If the task program occupancy rate is 100% in the state where the warning occurs, the task program occupancy rate error is generated.

(3) The following measures are required when the task program occupancy rate warning/error occurs.
(a) Secure the time for the system service to operate by reducing the amount of the user program execution within the main task/periodic task.
(b) Secure the time for the system servic to operate by increasing the execution cycle of the main task/periodic task of the basic parameter.
(c) Increase the task program occupancy rate excess warning setting of the basic parameter.

10.3.3 Error History Storage Function

The motion controller is designed to record the error history when errors occur, identify the cause of the errors and correct them.

Click on the 『Online』 - 『Diagnostics』 - 『PLC Error/Warning』 items of the menu to see the current errors and error history. Please refer to the details and action contents for each error item and eliminate the cause of error.



Items	Descriptions	Remarks
Error/Warning	Displays the current error/warning	-
Error history	Displays the error/warning that occur in chronological order	Save up to 100

Notice

The saved error history is deleted by clicking 'Clear' in the error/warning window.
If the error history exceeds 1,024, it is removed from the earliest history, and the latest 1,024 history is saved.

10.3.4 Failure Management

(1) Failure Types

The troubles are caused by failure of the motion controller itself, system configuration's error, error detection of operational results, etc. They can be divided into the failure mode stopping the operation for system safety; minor failure mode that informs a user of failure warning and resumes the operation.

The failures of the motion controller system are mainly caused by the below.

- Motion controller hardware's problems
- Operational error during execution of user programs
- Detection of errors caused by external device failure

(2) Operation mode in case of failures

In case failures occur, the motion controller system records the failure details in the special flag (F area) and determines whether resuming the operation based on the failure mode.

- In case of the motion controller hardware's failure

  In case there are problems with the motion controller, power, etc. that the motion controller cannot works normally, the system will be stopped; In case of minor failures such as a battery's low voltage, the warning is displayed and the operation will be resumed.

- Computational error during execution of user programs

  In case of the numeric operation error (Ex.: in case the denominator of division operation is 0) occurred during execution of user programs, the details will be displayed in the error flag and the system will resume the operation. If the operational time exceeds the operation delay monitoring set time during operation or equipped I/O modules cannot be normally controlled, the system will be stopped.

- Detection of errors caused by external device failure

  The failure of the external control device can be detected by the motion controller's user program; in case of detecting failures, the system will be stopped; in case of detecting minor failures, only the detection status will be displayed and the operation will be continued. (For the detailed use of the function to detect external device's failures, refer to the 10.3.5 Failure Diagnosis Function for the External Device.)

The information on failures occurrence is saved in the special relay (F area). Among F area flags, the information related to the failures are as below.

Double Word	Bit	Flag Name	Function	Description
%FD0	%FX2	_ERROR	Error	Error status
%FD2	-	_CNF_ER	System error	Reports the failure status of the system.
	%FX70	_ANNUM_ER	External device failure	Failures are detected from the external device.
	%FX72	_BPRM_ER	Basic parameters	There are some problems with the basic parameters.
	%FX73	_IOPRM_ER	IO parameters	There are some problems with I/O parameters.
	%FX74	_SPPRM_ER	Special module parameters	Abnormal special module parameters
	%FX75	_CPPRM_ER	Communication module parameters	Abnormal communication module parameters
	%FX76	_PGM_ER	Program error	There are some errors with the program.
	%FX78	_SWDT_ER	System Watch dog	The system Watchdog works.
	%FX80	_SWDT_ER	System Watch dog	The system Watchdog works.
	%FX85	_ENCPRM_ER	Encoder parameter error	Abnormal encoder parameter
	%FX86	_AXISPRM_ER	Axis parameter	Abnormal axis parameter
	%FX87	_GROUPPRM_ER	Axis group parameter	Abnormal axis group parameter
	%FX88	_ECPRM_ER	EtherCAT parameter	Abnormal EtherCAT parameter
	%FX89	_NCPRM_ER	NC parameter	Abnormal NC parameter
	%FX90	_NCPGM_ER	NC program	Error of NC parameter
	%FX91	_PTASK_CYCLE_ER	Main task	Period error of main task
	%FX92	_CTASK_CYCLE_ER	Periodic task	Period error of periodic task
	%FX93	_SYSTEM_ER	System error	System error
	%FX94	_TASK_PRM_USAGE_OVER_ER	Occupancy rate over error of task program	The task program occupancy rate exceeds 100%

Double Word	Bit	Flag Name	Function	Description
%FD4	-	_CNF_WAR	System warning	Reports the minor failure status of the system.
	%FX128	_RTC_ER	RTC data error	Abnormal RTC data
	%FX129	_PTASK_CYCLE_WAR	Main task	Period warning of main task
	%FX130	_CTASK_CYCLE_WAR	Periodic task	Period warning of periodic task
	%FX131	_ABSD_ER	Shutdown caused by abnormal operation	Stoppage caused by abnormal operation.
	%FX132	_MOTION_CONTRO_WAR	Motion control warring	Motion control function warring
	%FX134	_ANNUM_WAR	External device failure	Minor failures are detected from the external device.
	%FX135	_TASK_PRM_USAGE_OVER_WAR	Occupancy rate over warring of task program	The task program occupancy rate exceeds.
%FD7	%FX224	_ERR	Calculation error	In case of calculation error, this is ON during 1 scan
	%FX227	_ALL_OFF	Overall output OFF	When overall output is OFF, this is ON
	%FX229	_LER	Operational error latch	It maintains 0 in case of operational error.
	%FX247	_ARY_IDX_ERR	Array index range over	In case of range over error of array index, this is ON during 1 scan
	%FX248	_ARY_IDX_LER	Array index range over latch	In case of range over error of array index, this is ON during 1 scan
	%FX249	_UDF_STACK_ERR	UDF stack over	In case of over error of UDF stack, this is ON during 1 scan
	%FX250	_UDF_STACK_LER	UDF stack over latch	In case of over error of UDF stack, this is ON
%FW202	-	_ANC_ERR	Information on the external device's failure	Displays the information on the external device's failure
%FW203	-	_ANC_WAR	Information on the external device's minor failure	Displays the information on the external device's minor failure

Word	Bit	Flag Name	Function	Description
%FW1282	-	_ANC_ERR	Information on the external device's failure	Displays the information on the external device's failure
%FW1283	-	_ANC_WAR	Information on the external device's minor failure	Displays the information on the external device's minor failure

Notice

For more details on the whole flags, refer to the Appendix 1 Flag Table of the Outline of this manual.

10.3.5 Failure Diagnosis Function for the External Device

It is the function to detect the failure of the external device connected to the motion controller to realize stoppage of the system and warning easily. Through this function, you can detect the external device's failure without complex programming and can monitor the failure position without special devices (XG5000, etc.) or programs.
You can use the failure diagnosis function for the external devices as below.

(1) Failure types of external devices
• The failures of external devices are divided into the two types; failure (error) detected by combination of user programs and special relay (F area) requires stoppage of the motion controller operation; minor failure (warning) that continues the motion controller's operation and displays the detection status only.

(2) Flag to detect failures of external devices
The following flag types are used to diagnose failures of external devices.

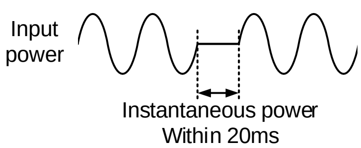
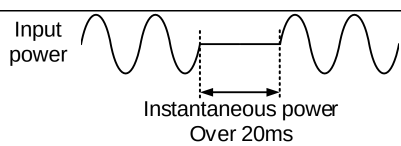
Word	Bit	Flag Name	Function	Description
%FW1282	-	_ANC_ERR	Information on the external device's failures	Input the error code of user-defined serious failure of external device.
%FW1283	-	_ANC_WAR	Information on the external device's MINOR failures	Input the error code of user-defined minor failure of external device.
-	%FX70	_ANNUM_ER	detection of external serious error	It is On when the external device's serious failure occurs.
-	%FX134	_ANNUM_WAR	detection of external slight error	It is On when the external device's minor failure occurs.
-	%FX20482	_CHK_ANC_ERR	Request detection of external serious error	It is the command flag asking to detect the external device's serious failure.
-	%FX20483	_CHK_ANC_WAR	Request detection of external slight error minor failure	It is the command flag asking to detect the external device's minor failure.

(3) How to detect the external device's serious failures
The following programming is used to detect the external device's serious failures.
(a) Save the error code that can be distinguished by external device's serious failures in %FW1282 (_ANC_ERR) through the MOVE command as below. (Input the values excluding 0)
(b) In case the external device's serious failures occur, %FX20482 (_CHK_ANC_ERR)flag will be On.
(c) When the main task program is completed, the motion controller checks whether %FX20482 (_CHK_ANC_ERR) is ON and detects serious failures.
(d) If the external device's serious failures occur, the motion controller will be in error status and will stop the operation. Then, %FX70 (_ANNUM_ER) is ON and %FX20482 (_CHK_ANC_ERR) flag is automatically Off. All outputs works based on IO parameter's emergency output settings.
(e) When failures occur, through XG5000, a user can figure out the causes of failures by monitoring %FW1282 (_ANC_ERR) flag.

(4) How to detect the external device's minor failures
The following programming is used to detect the external device's minor failures.
(a) Save the error code that can be distinguished by external device's serious failures in %FW1282 (_ANC_ERR) through the MOVE command as below. (Input the values excluding 0)
(b) In case the external device's minor failures occur, %FX20483 (_CHK_ANC_ERR)flag will be On.
(c) When the main task program is completed, the motion controller checks whether %FX20483 (_CHK_ANC_ERR) is ON and detects serious failures.
(d) If the external device's minor failures occur, %FX134 (_ANNUM_WAR)flag will be ON and continue to operation. Then, %FX20483 (_CHK_ANC_ERR) is automatically Off.
(e) When minor failures occur, through XG5000, a user can figure out the causes of failures by monitoring %FW20483 (_ANC_WAR)flag.
(d) If you input 0 again to %FW1283 (_ANC_WAR) after removing the causes of failures and turn ON%FX20483 (_CHK_ANC_WAR) again, detection of minor failures is canceled.

10.3.6 Instantaneous Power Failure Protection Function

Instantaneous power failure is detected when the input power voltage supplied to the motion controller becomes lower than the standard.
If the instantaneous power failure is detected, the following operation processing is performed.

Power Failure Time	Operation Processing
	1. If the instantaneous power failure occurs for the first time, the internal timer starts, and the operation is performed (without stopping) as before. 2. If the instantaneous power failure is canceled (within 20ms of the reference time), the internal timer start-up is stopped, and the operation is performed as before.
	1. If the power is not supplied during the excess of 20ms after the first occurrence of the instantaneous power failure, restart operation is performed in the same way as the power input.

Notice

Instantaneous power failure refers to the state where the voltage of the power supply specified by the motion controller in the power condition is lowered as it exceeds the allowable fluctuation range, and the power failure for a short time (several ms to dozens of ms) is called instantaneous power failure.

10.4 RTC Function

The motion controller has the embedded clock (RTC) function that keeps running by battery backup even when the power is off. The time data of the embedded RTC can be used for time management such as the system's operating history or failure history, etc. The RTC's current time is updated every scan by the flags for the system's operating state information.

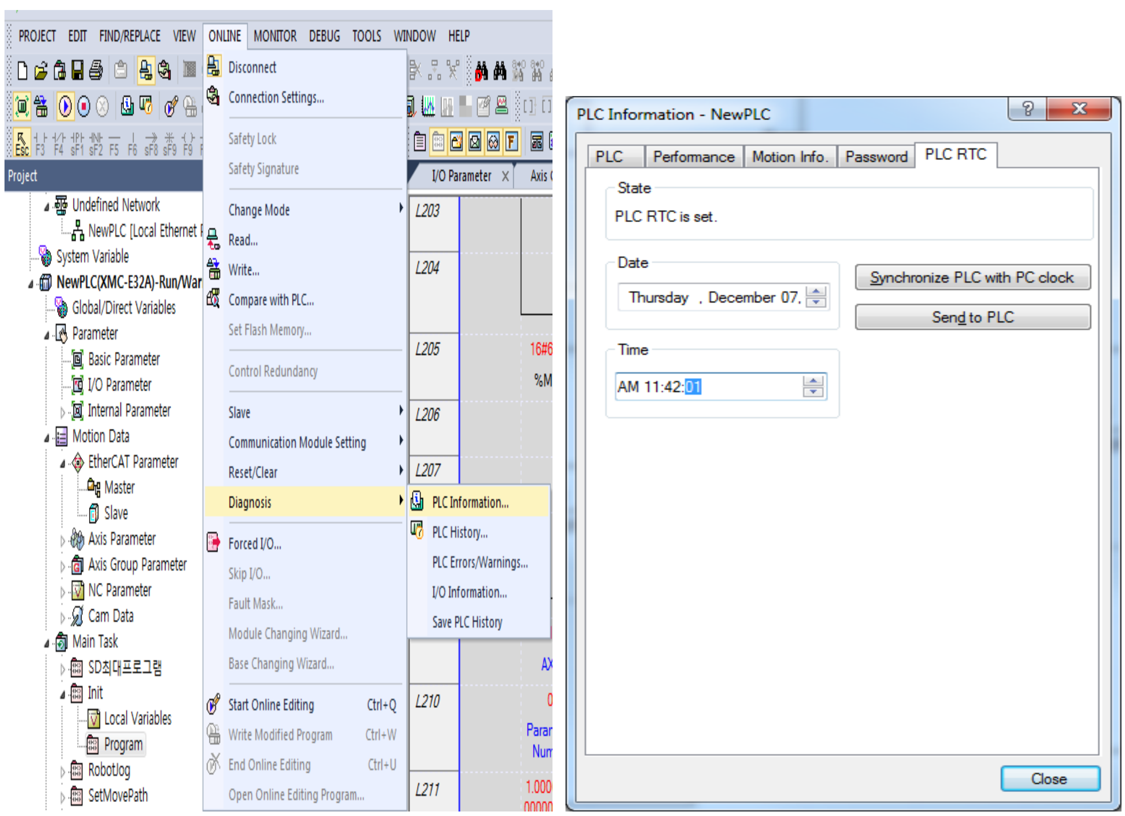
10.4.1 How to Use the RTC

(1) Reading/Setting clock data

(a) Reading the data from XG5000 and setting

1) Click『Online』-『Diagnosis』- 『motion controller information』.

2) Click the motion controller clock tab of『motion controller information』.



3) If you want to send the time of the PC to the motion controller, click 'Synchronization with PC clock' button.

4) If you want to set up the user defined time, after changing set values of the data and time box, click 'Send to motion controller'.

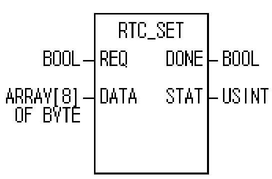
(b) Reading with the special relay

You can monitor the data by the special relay as shown in the below example.

Memory	Flag name	Function	Data	Description
%FB52	_RTC_TIME	RTC data[]		
%FB52~%FB59	_RTC_TIME[0]	RTC data(year)	h16	Year 2016
	_RTC_TIME[1]	RTC data(month)	h11	November
	_RTC_TIME[2]	RTC data(day)	h08	8th day
	_RTC_TIME[3]	RTC data(hour)	h19	At 7 pm
	_RTC_TIME[4]	RTC data(minute)	h12	12 minutes
	_RTC_TIME[5]	RTC data(second)	h54	54 seconds
	_RTC_TIME[6]	RTC data(weekday)	h02	Tuesday
	_RTC_TIME[7]	RTC data(a hundred years)	h20	2000s(decade)
%FW30	_RTC_DATE	RTC current date	2016-11-08	November 8, 2016
%FW31	_RTC_WEEK	RTC current weekday	2	Tuesday
%FD16	_RTC_TOD	RTC current hour	19:17:14.345	19:17:14.345

(c) Example of modifying clock data through the program

A user can set up the clock data through the program using RTC-SET function blocks as below.

Function block	I/O variable	Description
	REQ	It executes the function block in rising edge.
	DATA	Time data to input (Refer to the below table.)
	DONE	If the process is performed normally, 1 is output.
	STAT	In case of error, it outputs error codes.

Variable	Details	Example	Variable	Details	Example
DATA[0]	Year	16#16	DATA[4]	Minute	16#30
DATA[1]	Month	16#11	DATA[5]	Second	16#11
DATA[2]	Day	16#30	DATA[6]	Day of Week	-
DATA[3]	Hour	16#12	DATA[7]	Age	16#20
In case of 12:30:11, 30th, November, 2016, you do not need to input the separate day data since the day of week corresponding to the date is automatically set up.					

(d) Example of modifying clock data through the system flags
You can set up the clock data by filling up the clock data in the below area and turning on %FX20480 (_RTC_WR) without using function blocks.

Memory	Flag name	Description
%FB2568	_RTC_TIME_USER	Time to set
%FB2568~ %FB2575	_RTC_TIME_USER[0]	Time to set (year)
	_RTC_TIME_USER[1]	Time to set (month)
	_RTC_TIME_USER[2]	Time to set (day)
	_RTC_TIME_USER[3]	Time to set (hour)
	_RTC_TIME_USER[4]	Time to set (minute)
	_RTC_TIME_USER[5]	Time to set (second)
	_RTC_TIME_USER[6]	Time to set(weekday)
	_RTC_TIME_USER[7]	Time to set (a hundred of years)

(e) How to express day of the week

Number	0	1	2	3	4	5	6
Day	Sun.	Mon.	Tue.	Wed.	Thu.	Fri.	Sat.

(2) Time error
The RTC's error may be different depending on usual temperature.

Operation temperature	Maximum difference(Second/1 Day)	General case(Second/1 Day)
0℃	-12.26 ~ -1.03	-6.64
25℃	-10.37 ~ 0.86	-4.75
55℃	-13.09 ~ -1.86	-7.47

Notice

• The clock data may not be stated when the product is sent out from a factory so you need to set up clock data correctly before using the product.
• If you apply unavailable clock data to the RTC, it will not work normally.
Ex.) 25:00, 32th, 14 month
• In case the RTC stops due to battery problem or errors occur, when you input new clock data to the RTC, the error will be cleared.

10.5 Remote Function

In the motion controller, you can change the operation mode through the key switch attached to the module or through communication. For remote operation, put the basic unit's mode change switch on STOP position.

(1) The kinds of remote operations are as below.
- Access to XG5000 and operation through the USB port installed in the basic unit
- You can operate the other motion controllers connected to the network by using the motion controller's communication functions when XG5000 is connected to the basic unit.
- You can control the motion controller's operation status with HMI software, etc. though the dedicated communication

(2) Remote RUN/STOP
- It is the function to execute RUN/STOP through communication modules through the outside.
- This convenient function can be helpfully used when the motion controller is installed in the bad palace to operate or you need to RUN/STOP the CPU modules of a control panel from the outside.

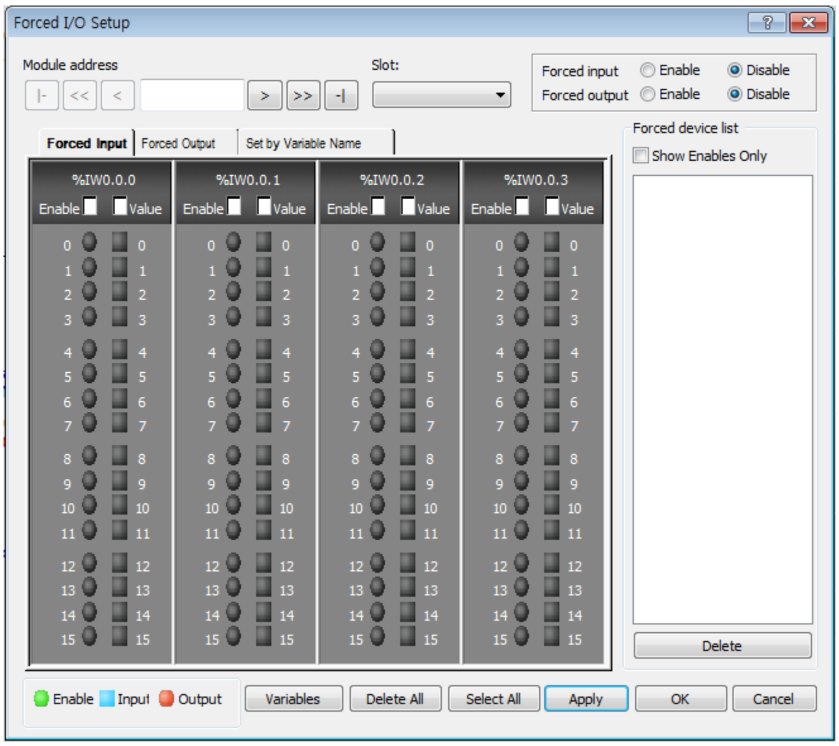
(3) Remote reset
- It is the function to reset the CPU module by remote control when errors occur.
- 'Reset' and 'Overall Reset' are available.

10.6 I/O Forced On/Off Functions

The forced I/O function is used to turn On/Off I/O areas by force regardless of the results of program execution.

10.6.1 Forced I/O Setting Method

Click『Online』-『 Forced I/O setting 』.



The below table represents the items related to the forced I/O setting.

Item		Description	Remarks
Movement of address		You can select the base and slot.	
Apply		You can set the forced input and output Enable / Unable	
Individual	Flag	You can set the forced I/O Enable / Unable by bit.	
	Data	You can set the forced I/O data (On/Off) by bit.	
View variables/comments		You can check the set input, output variables.	
Select All		You can set the forced I/O Enable under the condition that the whole I/O areas are On.	
Delete All		You can delete the forced I/O Enable under the condition that the whole I/O areas are Off.	
Set device		It displays the I/O area where even one bit is set.	

10.6.2 Time to Process the Forced I/O On/Off and Processing Method

(1) Forced input

When the forced input is set, among the data read from the input model at the time of Refresh, the data of the contact set as the forced On/Off is replaced by the forced set data to update the input image area. Accordingly, during program operation, among the actual input data, the forced set area is operated with the results replaced by the forced set data.

(2) Forced output

After completing the operation of user programs, at the time of output Refresh, among the data of the output image areas including the operation results, the data of the contact set as the forced On/Off is replaced by the forced set data, and then, they are output. Accordingly, in contrast with the forced input, in the case of the forced output, the data of the output image area shows the same data with the program operation results but the actual output changes by the forced output On/Off settings.

(3) Instructions to use the Forced I/O functions

- It work from the time of setting each I/O‘Enable’ after setting the forced data.
- Although the actual I/O modules are not equipped, the forced input can be set.
- In spite of Off → On of the power, change of operation modes and operation by the reset key. The previously set On/Off data is stored in the motion controller.
- Even in STOP mode, the forced input and output data is not eliminated.
- When you try to set the new data from the beginning, cancel all settings of I/O by using ‘Delete All’ before use.

(4) Operations in case of errors

- When errors occur after setting the forced output, it works based on「Output Hold when errors occur」 of output control settings in the basic parameters and 「Emergency Output」 of the I/O parameters. In case of error occurrence, if you select the emergency output as 「Clear」 after setting Output Hold when errors occur」, the output is off when errors occur; if you choose 「Hold」, the output status will be maintained.

10.7 Function Saving the Operation History

There are 5 types of operation history; error history, mode conversion history, power down history and system history. The occurrence time, frequency, operating details of each event are saved in the memory and you can conveniently monitor the data through XG5000. The operation history is saved in the motion controller unless it is deleted through XG5000.

10.7.1 Error History

It saves the error history occurred during operation.

- The error code, date, time, error details are saved..
- The histories can be saved up to 1,024 EA.
- It is automatically canceled when the memory backup is cleared due to the battery's low voltage, etc.

10.7.2 Mode Conversion History

It saves the information on the changed mode and time when changing the operation mode.

- It saves the data, time, mode conversion details.
- The histories can be saved up to 1,024 EA.

10.7.3 Power Down History

On or Off time of the power is saved as the ON/OFF information.

- ON/OFF information, date and time are saved.
- The histories can be saved up to 2,048 EA.

10.7.4 System History

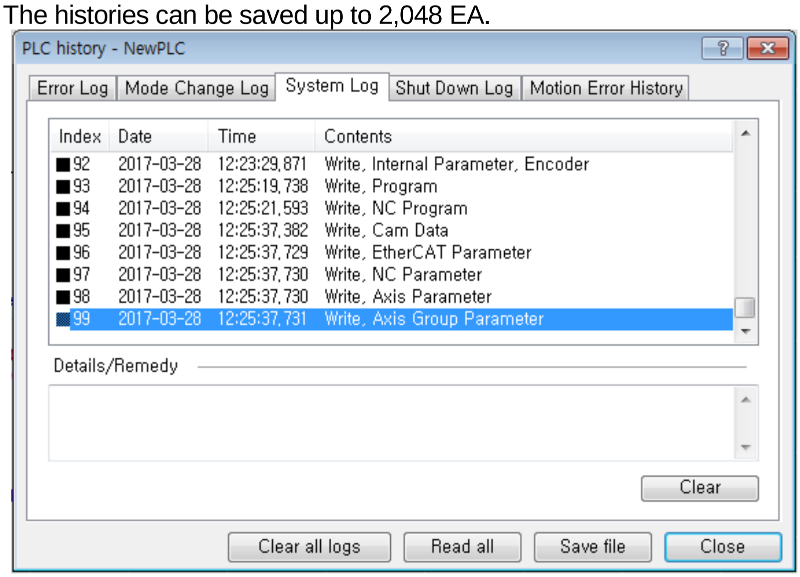
It saves the operation history of the system occurred during operation.

- The date, time and details of operation changes are saved.
- The histories related to system operation are saved; XG5000 operation information, change of the key switch position, etc.
- The histories can be saved up to 1,024 EA.

10.7.5 Motion Error History

It saves the error history occurred during motion control.

- The error code, date, time, error details are saved..
- The histories related to system operation are saved; XG5000 operation information, change of the key switch position, etc.
- The histories can be saved up to 2,048 EA.



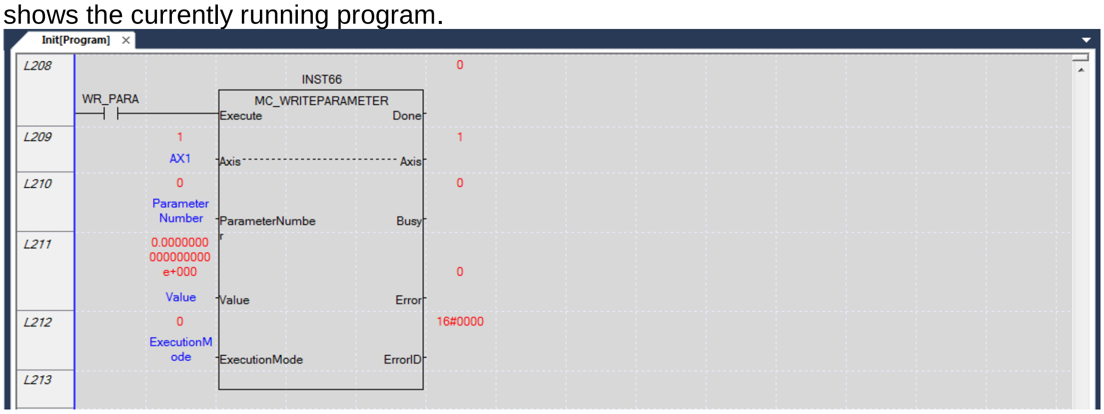
10.8 Program Modification during Operation(Modification during RUN)

You can modify the programs and communication parameters without stopping control operations during running the motion controller. The below describes the basic modification method. For more details on Modification during RUN, refer to the XG5000 manual.

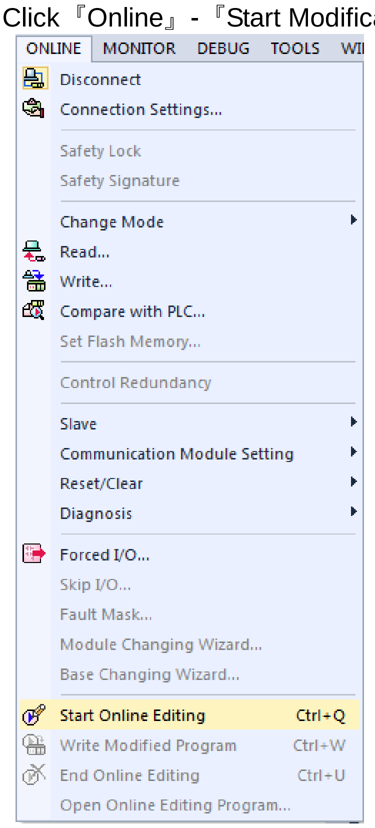
The items that can be modified during RUN are limited to programs, network parameters.
You cannot modify adding tasks, deletion, parameters, etc. during RUN.

10.8.1 Modification Procedures during RUN

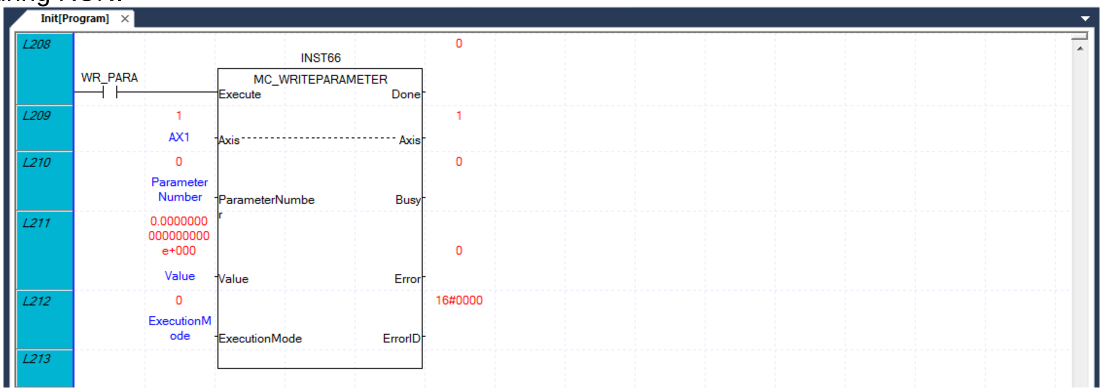
(1) It shows the currently running program.



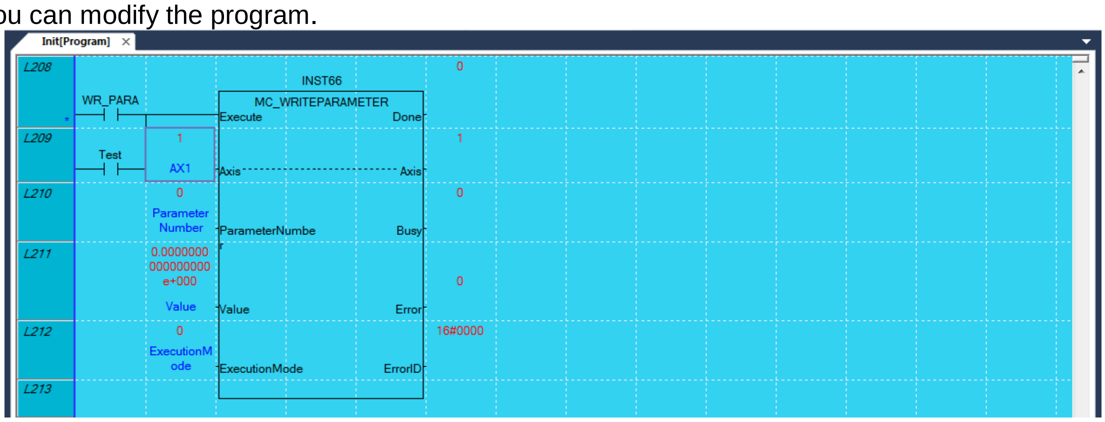
(2) Click『Online』-『Start Modification During RUN』.



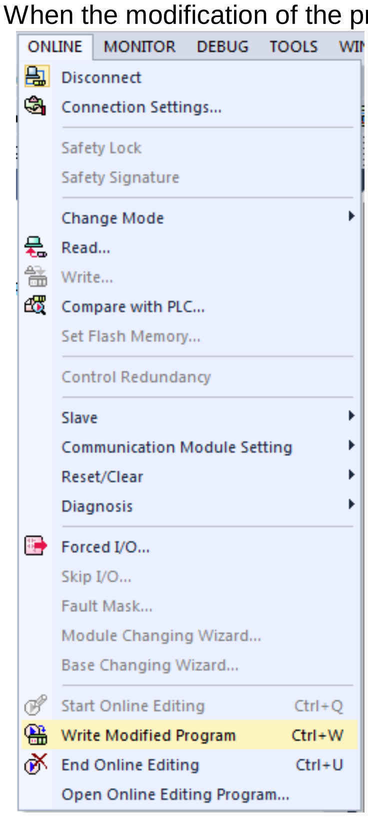
(3) Then, the background color of the program window changes and it is converted into the mode of modification during RUN.

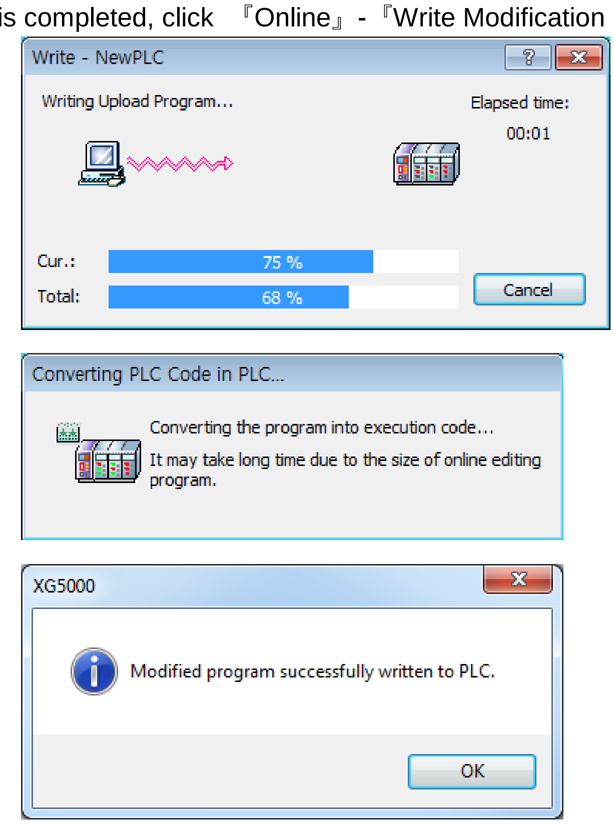


(4) You can modify the program.

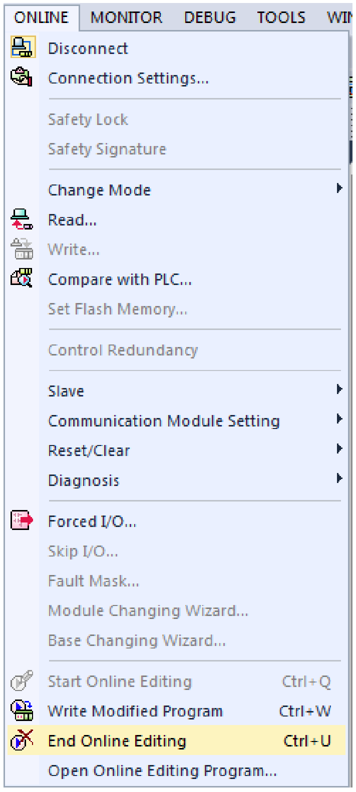


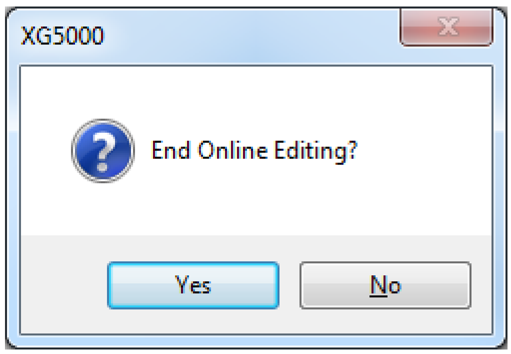
(5) When the modification of the program is completed, click 『Online』 - 『Write Modification During RUN』



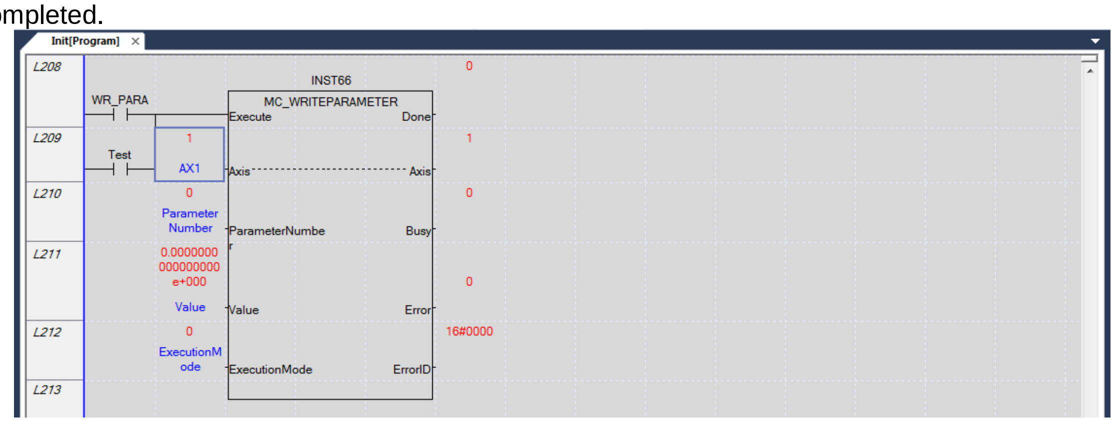


(6) When Write Program is completed, click 『Online』 - 『End Modification During RUN』.





(7) The background color of the program window changes into the original one and modification during RUN is completed.



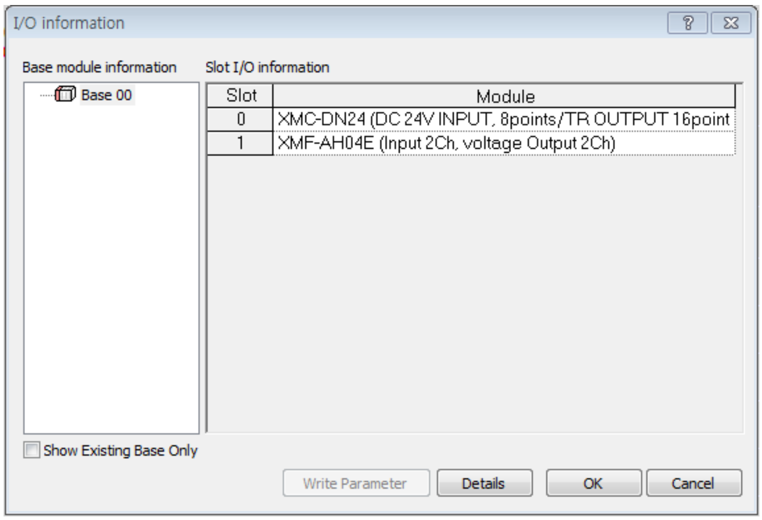
Notice

- For Modification of communication parameters during RUN, after changing the network configuration items of XG5000 in the RUN status without going into the Modification during RUN menu, click 『Online』 - 『Write』 and choose ‘Network Parameter’ to execute Write.

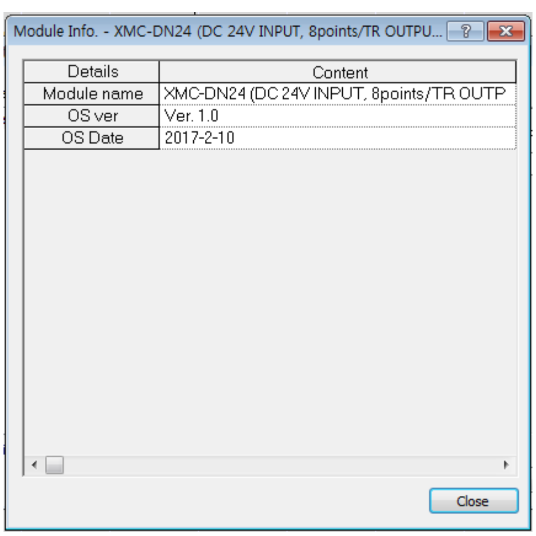
10.9 Read I/O Information

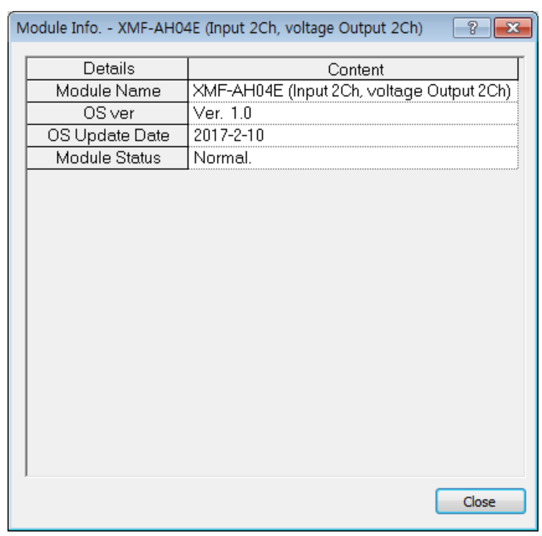
It is the function to monitor each module's information comprising the motion controller system.

(1) If you click 『Online』-『Diagnosis』-『I/O Information』, the information of each module of connected systems will be monitored.



(2) If you click 'Detailed Information' after choosing the module, the details on the module will be displayed.

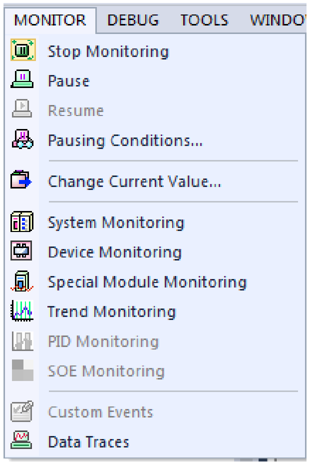




10.10 Monitoring Functions

It is the function to monitor the motion controller system’s general information.

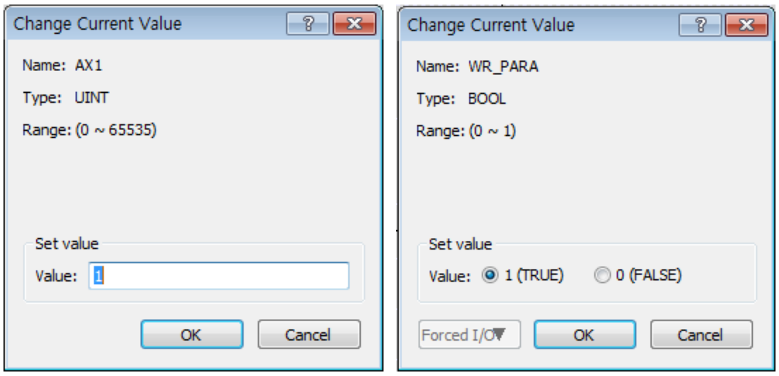
(1) If you click 『Monitor』, the submenu will be displayed as below.



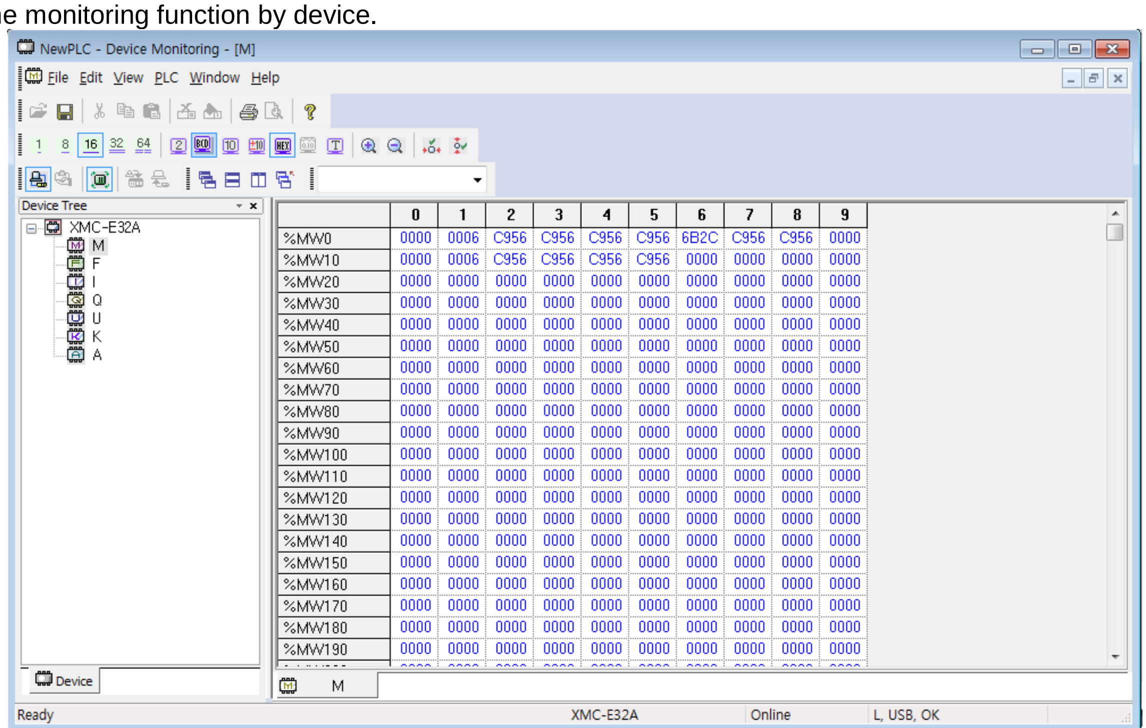
(2) The below table provides the descriptions on each item.

Items	Descriptions	Remarks
Start/End monitor	Specifies the startup and end of the monitor.	Changes every time you click
Suspend monitor	Suspends the monitor.	
Restart monitor	Executes the suspended monitor again.	
Monitor suspension setting	It is the function to suspend the monitor when the set device’s value is matched with the conditions.	Restarts when you click ‘Restart Monitor’
Changing the current value	Changes the currently selected device’s current value.	
System monitor	Monitors the current system’s general information.	
Device monitor	It is the function to monitor each device.	
Trend monitor	Monitors the set device’s trend.	For more details, refer to the XG5000 manual.
User event	Monitors the set device’s value when the event specified by a user occurs.	
Data trace	Traces the set device’s value.	

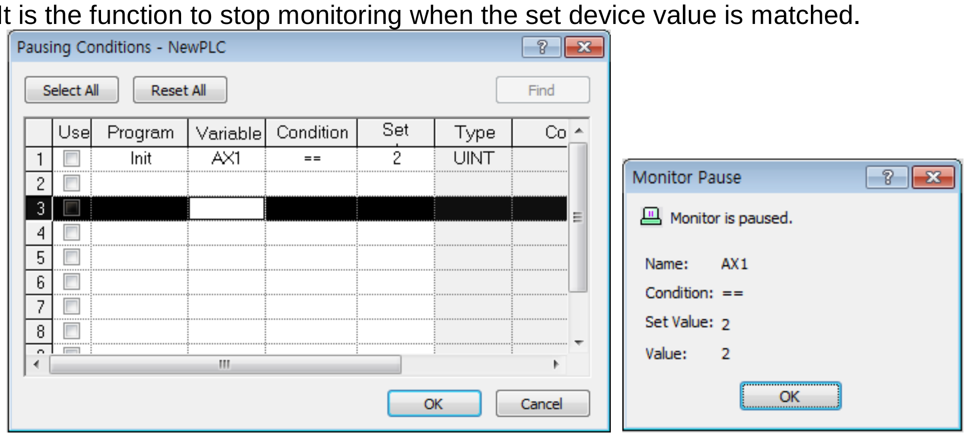
(a) Changing the current value
It is the function to change the current value of each selected device in the program window.



(b) Device monitor
It is the monitoring function by device.



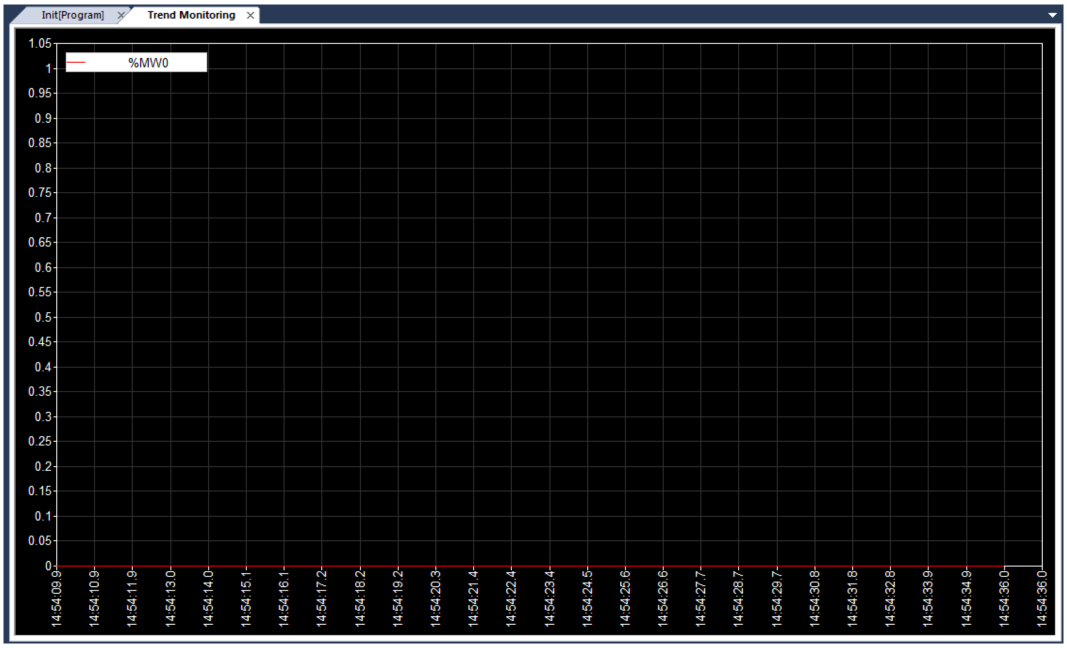
(c) Monitor suspension setting
It is the function to stop monitoring when the set device value is matched.



(d) Trend Monitor

It is the function to represent the set device value in a graphic form. The value represented on the graph is not the data collected by the motion controller at the right timing but the value read from XG5000 through the communication function. Accordingly, communication delay can occur so it may not be matched with the actual data collected at the right cycle.

You are recommended to use the Trend Monitor function to check the rough data trend.



(e) Data trace

It is a function to collect device values set at the time of event set by the user or at a desired time and monitor them with graph or data. Unlike trend monitor, it collects actual data at a sampling period set by the user, and thus is used check actual data at a certain point in time.

1) Data Trace condition setting
2) Device setting
3) Data display

Note

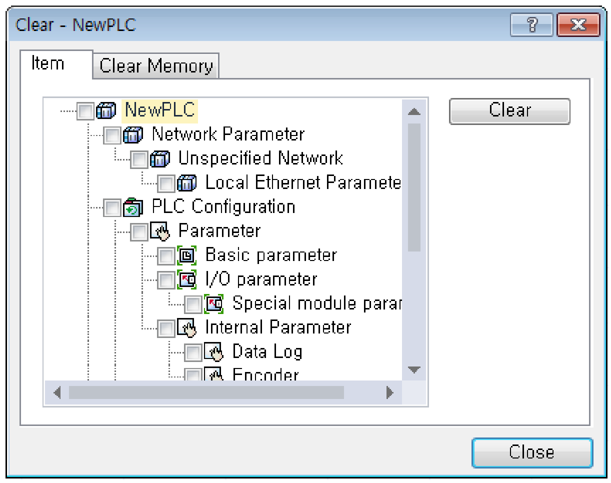
For more details on the monitoring function, refer to XG5000 manual.

10.11 Function to Delete All of the Motion Controller

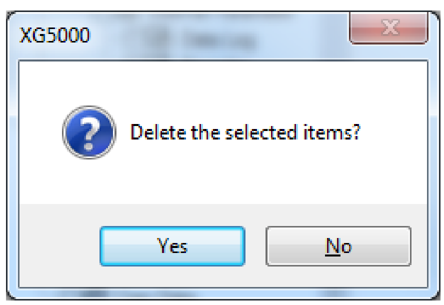
The function to delete all of motion controller is the initialization function to delete all programs, parameters, passwords, data stored in the motion controller.

(1) How to delete all of motion controller

(a) Click『Online』-『Reset/Clear』-『Clear PLC』.



(b) If you choose 『Yes』 in the dialog box, the window for selecting the connection method with the motion controller to be deleted is created.



(c) If you select “Yes” in the confirmation window, the data value will be cleared to “0”. Since the parameter may need a default value depending on the data, an error may occur after the erase operation. If an error occurs, it is necessary to write the data as an initial value once.

10.12 Built-in Input/Output Function

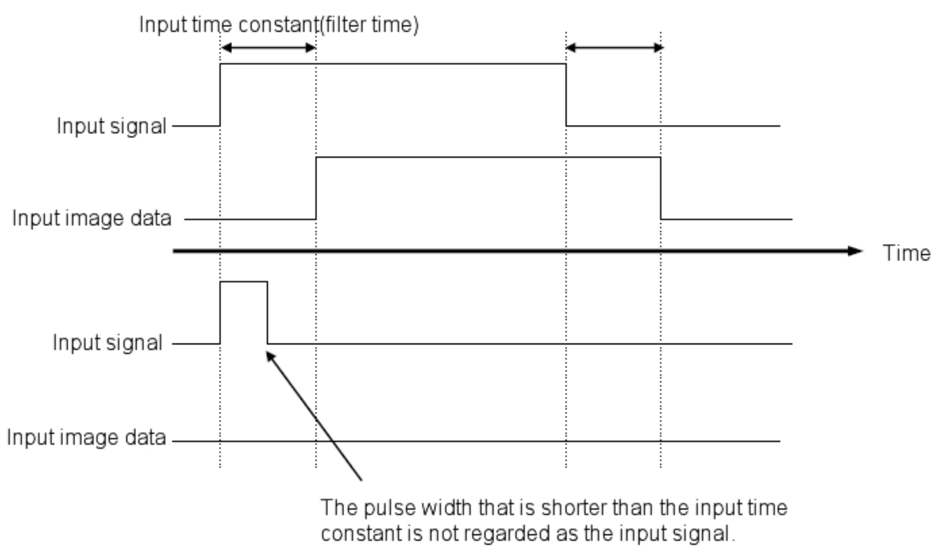
10.12.1 Input Filter Function

The motion controller's input modules have the input filter function to prevent the external noise signal flowed into the input signal. For more details on the input filter function, refer to the below.

(1) Purposes and Operations of the input filter function

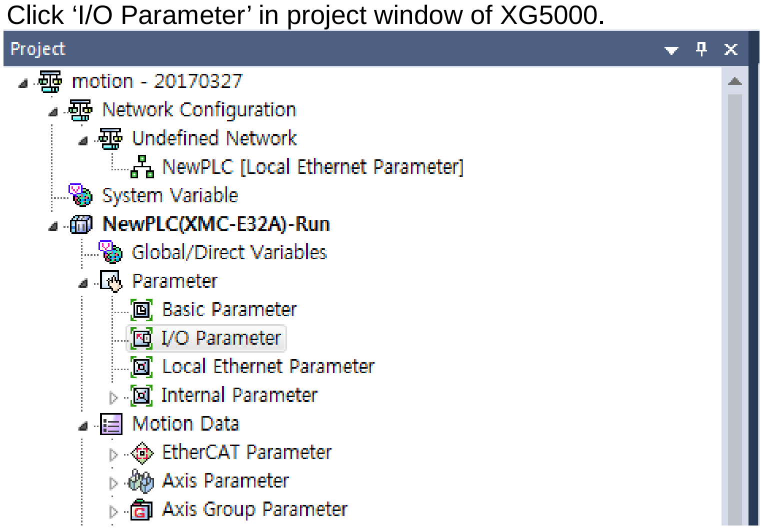
Under the environment with serious noise or in the case of the equipment that is greatly affected by the input signal's pulse width, the system may receive incorrect input depending on the input signal status. To prevent such incorrect input, the input filter function does not regard the signal that is shorter than the set time by a user as input. In the case of the motion controller, you can set the input filter time in the range of 1ms~100ms.

The below timing chart represents the operations of the input filter function.

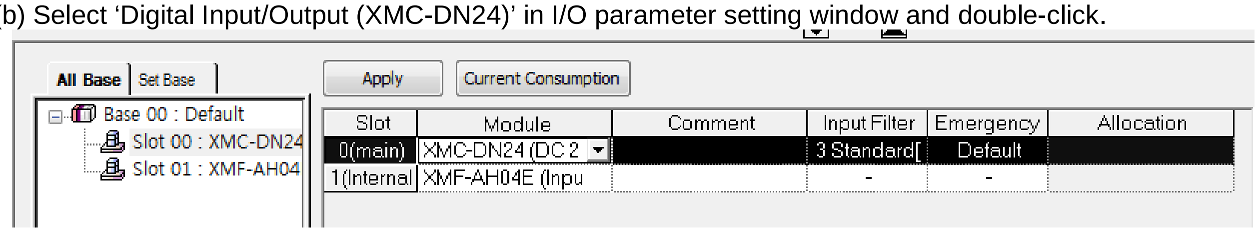


(2) Input filter setting method

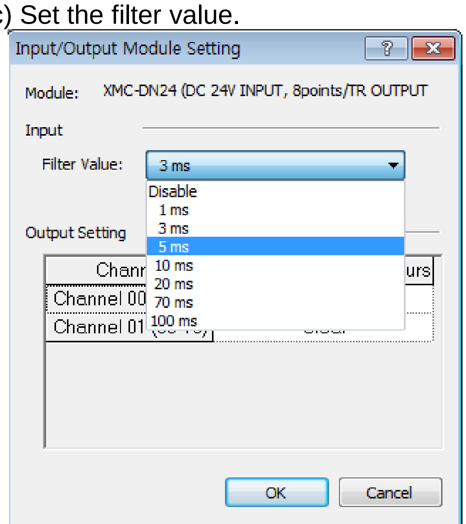
(a) Click 'I/O Parameter' in project window of XG5000.



(b) Select ‘Digital Input/Output (XMC-DN24)’ in I/O parameter setting window and double-click.



(c) Set the filter value.



Notice

When the filter value is set, the cycle of the main task should be set to a value smaller than the set filter value.

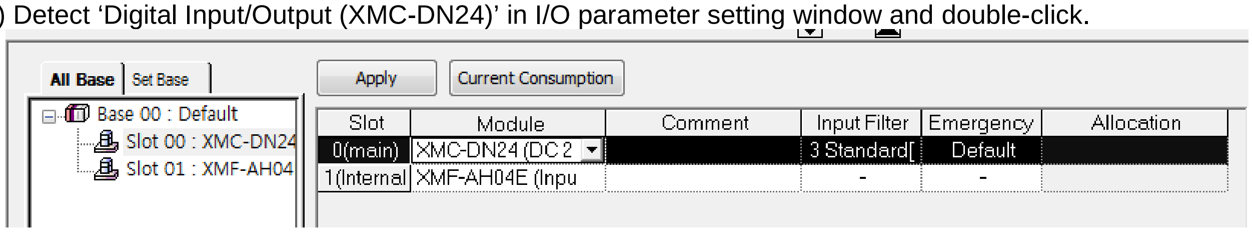
For example, if the filter value is set to 3ms, the cycle of the main task should be set to 1ms or 2ms.

10.12.2 Emergency Output Function

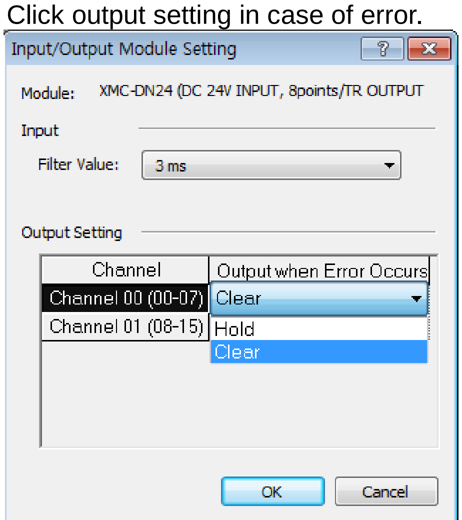
The XMC's output module supports the emergency output function to determine whether maintaining the output status of the output module or clearing it when the motion controller is stopped due to errors.
You can set the emergency output by 8 points. For more details on how to set the emergency output, refer to the below.

(1) Output status setting in case of error

(a) Detect 'Digital Input/Output (XMC-DN24)' in I/O parameter setting window and double-click.



(b) Click output setting in case of error.



If you select [Clear] as the output setting when an error occurs, the output is turned off when the operation is stopped due to an error that occurs in the motion controller. If you select [Hold], the output status is maintained.

10.13 Reading of Serial Number Information

It is a function to monitor serial number information of motion controller.

(1) It can be monitored as follows through variables.

Memory	Flag name	Data	Description
%FB80	_SERIAL_NUM		Serial number data[]
	_SERIAL_NUM[0]	h08	Serial number 1~2th digit
	_SERIAL_NUM[1]	h08	Serial number 3~4th digit
	_SERIAL_NUM[2]	h08	Serial number 5~6th digit
	_SERIAL_NUM[3]	h08	Serial number 7~8th digit
	_SERIAL_NUM[4]	h08	Serial number 9~10th digit
	_SERIAL_NUM[5]	h08	Serial number 11~12th digit
	_SERIAL_NUM[6]	h08	Serial number 13~14th digit
	_SERIAL_NUM[7]	h08	Serial number 15~16th digit
	_SERIAL_NUM[8]	h08	Serial number 17~18th digit
	_SERIAL_NUM[9]	h08	Serial number 19~20th digit
	_SERIAL_NUM[10]	h08	Serial number 21~22th digit
	_SERIAL_NUM[11]	h08	Serial number 23~24th digit
	_SERIAL_NUM[12]	h08	Serial number 25~26th digit
	_SERIAL_NUM[13]	h08	Serial number 27~28th digit
	_SERIAL_NUM[14]	h08	Serial number 29~30th digit
	_SERIAL_NUM[15]	h08	Serial number 31~32th digit
	_SERIAL_NUM[16]	h08	Serial number 33~34th digit
	_SERIAL_NUM[17]	h08	Serial number 35~36th digit
	_SERIAL_NUM[18]	h08	Serial number 37~38th digit
	_SERIAL_NUM[19]	h08	Serial number 39~40th digit

Ex) If the serial number is 123456789, the flag are displayed as follows
(The unused area of the serial number is displayed as 0)

Monitor 1

	PLC	Program	Variable/Device	Value	Type	Device/Variable	Comment
1	NewPLC	<GLOBAL>	⊟ _SERIAL_NUM		ARRAY[	%FB80	Serial Number
2	NewPLC	<GLOBAL>	_SERIAL_NUM[0]	16#50	BYTE		
3	NewPLC	<GLOBAL>	_SERIAL_NUM[1]	16#07	BYTE		
4	NewPLC	<GLOBAL>	_SERIAL_NUM[2]	16#21	BYTE		
5	NewPLC	<GLOBAL>	_SERIAL_NUM[3]	16#31	BYTE		
6	NewPLC	<GLOBAL>	_SERIAL_NUM[4]	16#BA	BYTE		
7	NewPLC	<GLOBAL>	_SERIAL_NUM[5]	16#20	BYTE		
8	NewPLC	<GLOBAL>	_SERIAL_NUM[6]	16#00	BYTE		
9	NewPLC	<GLOBAL>	_SERIAL_NUM[7]	16#00	BYTE		
10	NewPLC	<GLOBAL>	_SERIAL_NUM[8]	16#00	BYTE		
11	NewPLC	<GLOBAL>	_SERIAL_NUM[9]	16#00	BYTE		
12	NewPLC	<GLOBAL>	_SERIAL_NUM[10]	16#00	BYTE		
13	NewPLC	<GLOBAL>	_SERIAL_NUM[11]	16#00	BYTE		
14	NewPLC	<GLOBAL>	_SERIAL_NUM[12]	16#00	BYTE		
15	NewPLC	<GLOBAL>	_SERIAL_NUM[13]	16#00	BYTE		
16	NewPLC	<GLOBAL>	_SERIAL_NUM[14]	16#00	BYTE		
17	NewPLC	<GLOBAL>	_SERIAL_NUM[15]	16#00	BYTE		
18	NewPLC	<GLOBAL>	_SERIAL_NUM[16]	16#00	BYTE		
19	NewPLC	<GLOBAL>	_SERIAL_NUM[17]	16#00	BYTE		
20	NewPLC	<GLOBAL>	_SERIAL_NUM[18]	16#00	BYTE		
21	NewPLC	<GLOBAL>	_SERIAL_NUM[19]	16#00	BYTE		
22							

Chapter 11 Datalog Function

11.1 Overview

Motion controller comes with built-in datalog function. This chapter describes the specification and usage of the datalog function.

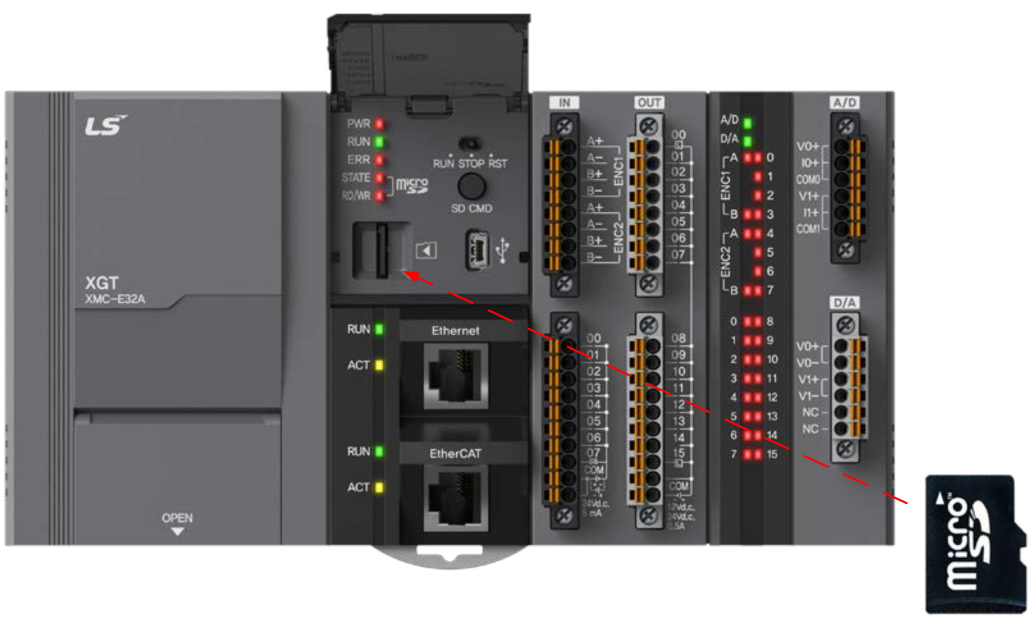
11.1.1 Features

Using the motion controller internal datalog function, you can collect run data of motion controller and save them into a SD memory card in the CSV (Comma-Separated Values) format just with a simple parameter configuration. The function has the following features.

(1) Easy Motion Controller Device Data Saving

You can save motion controller's various device data with just a simple parameter configuration. It eliminates the need to construct a network to collect large volumes of run data, thereby saving system costs.

In addition, it eliminates problems that might be caused in network-based data collection, such as communication cutoff or cable disconnection.



(2) Precise Data Collection

This function allows you to collect precise data for main task, by 1ms or in accordance with other various run conditions.

In addition, you can use the trigger function to save data before/after the trigger. Or you can use the event function to save data changes from the event occurrence. This allows for easy analysis of the system's run status, which also saves system maintenance costs.

(3) Large-volume Operation Data

The function supports up to 8GB SD memory card, which allows for saving run data over a long period of time

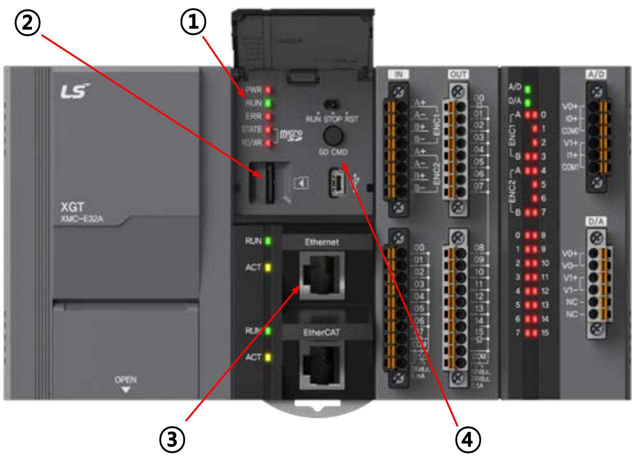
(4) FTP Interface

Files saved in the datalog can be read remotely using FTP, making it easier to verify data fluctuations.

11.1.2 Part Names

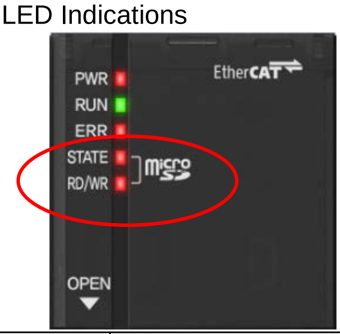
The names of pars related to datalog function are as follows.

(1) Part names



	Names	Description
①	Status LED	Indicates run status of SD memory and datalog.
②	SD memory mounting slot	A slot where SD memory is mounted.
③	Internal Ethernet Port	The port is used when transmitting files using the FTP function of the internal Ethernet.
④	SD CMD Button	It is used for SD PWR ON, OFF or SD additional functions • Push the button for 0.7[sec] ~ 3[sec] : SD additional function • Push the button above 3[sec] : SD PWR ON, OFF

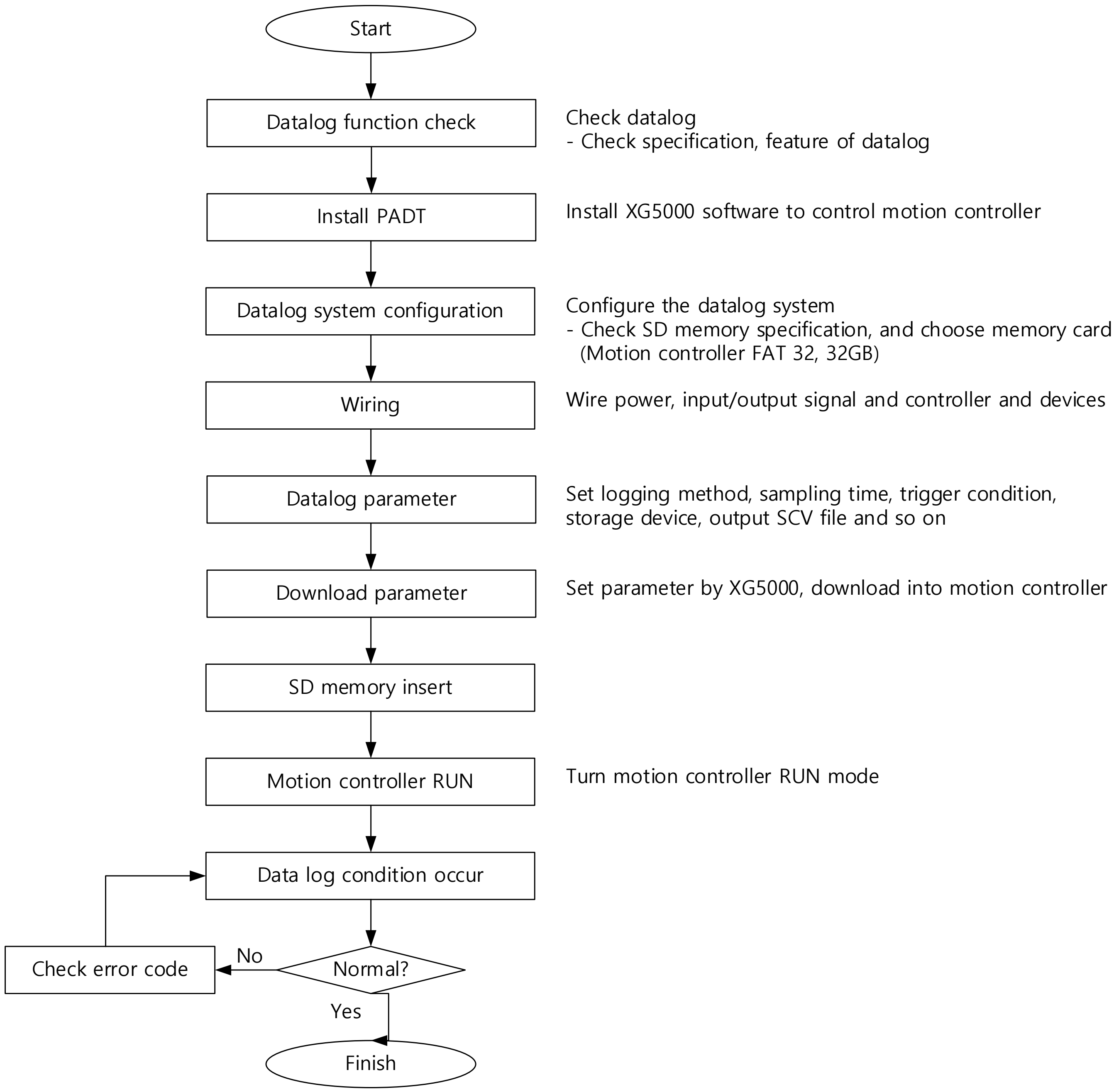
(2) LED Indications



Names	Description	Specifications
PWR	Indicates motion controller power supply status	Turns on during power ON.
RUN	Indicates motion controller run	Turns on during RUN, and turns off at STOP, ERR.
ERR	Indicates motion controller error status	Flashes when error occurs
STATE	Indicates the status of SD memory mounted.	Turns on : SD card mounted, status normal Flashes : SD card mounted, error occurred (flashes at 2s interval) Turns off: SD card removed
RD/WR	Indicates SD card control status	Flashes : Reading or writing SD card (flashes at 100ms interval) Turns off : Access to SD card terminated

11.1.3 Operation Sequence

Datalog is performed in the following sequence.

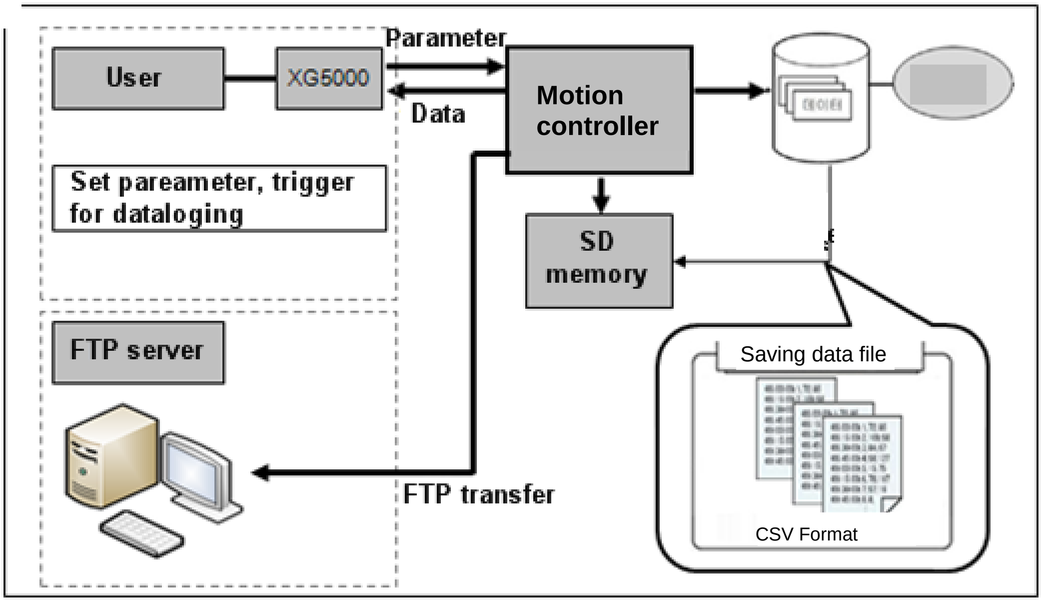


Note

1. The SD memory should be formatted in FAT 32 format to be used for motion controller datalog function.
2. The maximum storage of SD memory supported is 32GB.

11.1.4 Control Signal Flow

The datalog function saves the motion controller device values into the SD memory or exchanges the value with external device or software, in accordance with the following data flow.



11.2 Performance Specifications

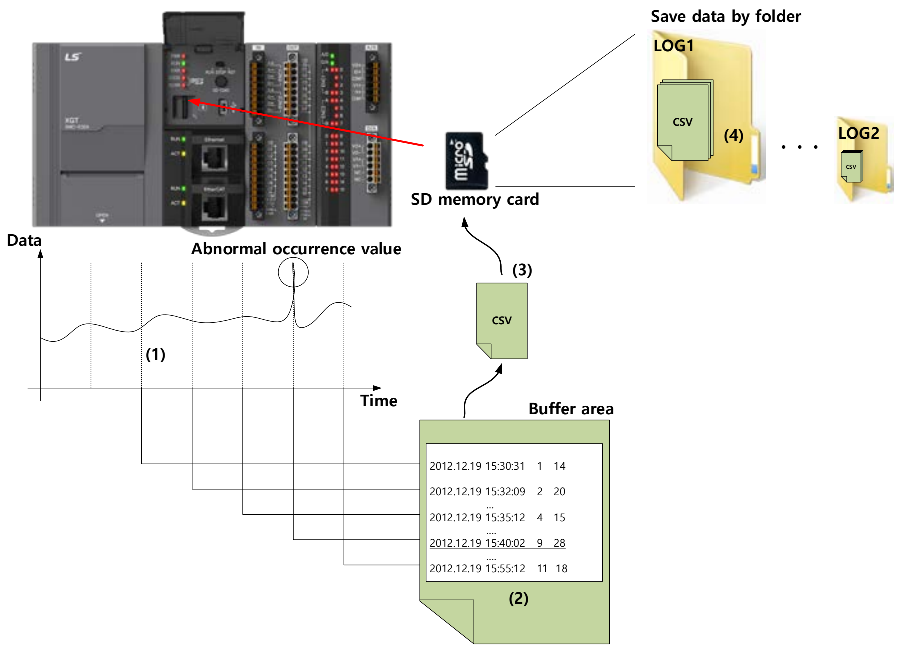
Items			Specifications	Note
Function Configuration	Group Configuration		Up to 16 groups	
	Configuration Data		Up to 64 per group	
	Data Collection Type		regular / trigger / event	
	File Format		CSV	
	File Size		Up to 16MByte	
	Data Type		BIT, BYTE, WORD, DWORD, LWORD, SINT, INT, DINT, LINT USINT, UINT, UDINT, ULINT, REAL, LREAL, STRING	
	Save Data Type		Decimal, Hexadecimal, Exponent, character string	
Regular Save	Sampling Cycle		Main-task Cycle, Designation Cycle, Designation Time	
	Sampling Object		64 per file	
	File Conversion	Conversion Timing	Designate with File Size 10 ~ 16,384KB Designate with No. of Save Lines 1,000~50,000	
		Maximum No. of Files	256 per folder	
Trigger Save	Single Condition		Bit: elevation/descent Word: small, big, same, different, big or same, small or same	
	Operation Condition		AND, OR condition	
	Trigger Save Range		Up to 69,905 data per group	
	Files Conversion	Conversion Timing	Designate with File Size 10 ~ 16,384KB Designate with No. of Save Lines 1,000~50,000	
		Maximum No. of Files	256 per folder	
Event Save	Single Condition		Bit: ON, OFF, elevation, descent, transfer Word: small, big, same, different, big or same, small or same	
	Operation Condition		AND, OR condition	
	Files Conversion	Conversion Timing	Designate with File Size 10 ~ 16,384KB Designate with No. of Save Lines 1,000~50,000	
		Maximum No. of Files	256 per folder	
Formatting Function	Formatting Type		Quick Format	
	Cluster Size		32kByte	
	Volume Label		LSIS (fixed)	
SD memory	Power Input		2.7 ~ 3.6VDC	
	Card Size		15mm * 11mm * 1.0mm	
	Maximum Capacity		Up to 32GB (Only 8GB can be available for above 8GB memory size)	
	Memory Type		Micro SD, SDHC (Recommended manufacturer: SanDisk, Transcend)	
	File System		FAT 32	

Note

SanDisk, Transcend SD memories are recommended for internal datalog. Use of SD memory from other manufacturer may result in unexpected run. Please choose your SD memory card with caution.

11.3 Specification Functions

Datalog function refers to storing device values of motion controller at a set interval or when the trigger condition occurs. Thus collected data are saved into the SD memory card in CSV format.



11.3.1 Data Type and Device

You can save device memories using motion controller's datalog function. When the clock function is normal, the memory is saved along with the time information.

If the clock function is abnormal, the time information is saved as the default value, which is 2000/ 01/01 00:00:00.000.

(1) Data Type

The data types and character strings that can be saved using the internal datalog function of motion controller is as follows.

Data Type	Output	Size (including ',' BYTE)
BOOL	0 or 1	2
BYTE	00 ~ FF	3
WORD	0000 ~ FFFF	5
DWROD	00000000 ~ FFFFFFFF	9
LWORD	00000000 00000000 ~ FFFFFFFF FFFFFFFF	17
SINT	-128 ~ 127	5
INT	-32,768 ~ 32,767	7
DINT	-2,147,483,648 ~ 2,147,483,647	12
LINT	-576,460,752,303,423,488 ~ 576,460,752,303,423,487	21
USINT	0 ~ 255	4
UINT	0 ~ 65,535	6
UDINT	0 ~ 4,294,967,295	11

Data Type	Output	Size (including ‘,’ BYTE)
ULINT	0 ~ 1,152,921,504,606,846,975	20
REAL	-3.402823466e+038 ~ -1.175494351e-038 or 0 or 1.175494351e-038 ~ 3.402823466e+038	17
LREAL	-1.7976931348623157e+308 ~ -2.2250738585072014e-308 or 0 or 2.2250738585072014e-308 ~ 1.7976931348623157e+308	24
STRING	Fixed Character (up to 32 characters	33

<table>
<tr><th>ASCII Code</th><th>Indication</th><th>ASCII Code</th><th>Indication</th><th>ASCII Code</th><th>Indication</th><th>ASCII Code</th><th>Indication</th></tr>
<tr><td>0x20</td><td>SP</td><td>0x2A</td><td>*</td><td>0x3E</td><td>></td><td>0x7B</td><td>{</td></tr>
<tr><td>0x21</td><td>!</td><td>0x2B</td><td>+</td><td>0x3F</td><td>?</td><td>0x7C</td><td>|</td></tr>
<tr><td>0x22</td><td>“</td><td>0x2D</td><td>-</td><td>0x41 ~ 0x5A</td><td>English (upper case)</td><td>0x7D</td><td>}</td></tr>
<tr><td>0x23</td><td>#</td><td>0x2E</td><td>.</td><td>0x5B</td><td>[</td><td>0x7E</td><td>~</td></tr>
<tr><td>0x24</td><td>$</td><td>0x2F</td><td>/</td><td>0x5C</td><td>\</td><td></td><td></td></tr>
<tr><td>0x25</td><td>%</td><td>0x30 ~ 0x39</td><td>Number</td><td>0x5D</td><td>]</td><td></td><td></td></tr>
<tr><td>0x26</td><td>&</td><td>0x3A</td><td>:</td><td>0x5E</td><td>^</td><td></td><td></td></tr>
<tr><td>0x27</td><td>‘</td><td>0x3B</td><td>;</td><td>0x5F</td><td>_</td><td></td><td></td></tr>
<tr><td>0x28</td><td>(</td><td>0x3C</td><td><</td><td>0x60</td><td>`</td><td></td><td></td></tr>
<tr><td>0x29</td><td>)</td><td>0x3D</td><td>=</td><td>0x61 ~ 0x7A</td><td>English (lower case)</td><td></td><td></td></tr>
</table>

(2) Device Available for Saving

The devices that can be used to save files using the internal datalog function of motion controller are as follows.

Data Type	Description	Note
BOOL	I, Q, M, K, A, F, U	
WORD	I, Q, M, K, A, F, U	

(3) Calculates data unit when saving buffer

The motion controller for data saving supported by internal datalog is BYTE. Therefore, operation of data that accumulates inside the buffer during data collection is performed as follows.

(Unit: BYTE)

Type	Calculation Unit
BOOL	1
BYTE	1
WORD	2
DWORD	4
LWORD	8
INT	2
SINT	1
DINT	4
LINT	8
UINT	2
USINT	1
UDINT	4
ULINT	8
REAL	4
LREAL	8
STRING	32

(4) Data Conversion

Data are collected in the following order, and converted into the set types.

(a) 2 WORD Data (DWORD, DINT, UDINT, REAL)

Ex) %MW0: 0x1234, %MW1: Converts to 0x0000 → 0000 1234

Sequence	#2	#1
Device	%MW1	%MW0

(b) 4 WORD Data (LWORD, LINT, ULINT, LREAL)

Ex) %MW0: 0x1234, %MW1:0x5678, %MW2:0x000, %MW3: Converts to 0x000 → 0000 0000 5678 1234

Sequence	#4	#3	#2	#1
Device	%MW3	%MW2	%MW1	%MW0

(c) Character String Conversion

- Unlike other types, character strings are saved up to 32 characters, and converted into 2 characters per word. If a 0x00 value exists during conversion, conversion is performed up to that character string, and further conversion is not performed.

Ex) 16 words without 0x00→ 32 characters

16 words with 0x00→ character string converted up to 0x00

- When converting character strings, characters which do not correspond with ASCII (see 11.3.1) are all converted to Null.

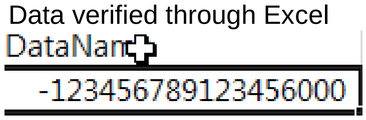
Sequence	#16	#15	#14	…	#1
Device	%MW14	%MW13	…	%MW0	%MW14

Note

If the data are saved using the LINT type, the following may not be represented when verifying the data through Excel.

☞ Actual save data

☞ Data verified through Excel



In such cases, you can view the normal data by reading the data using Word Pad.

Note

Float conversion, such as REAL type, supports IEEE754 standards as follows.

BIT 31 **BIT 0**

Sign (S)	Exponent E)	Fix d Decimal Point (F)

Sign (S): 1 BIT
Exponent (E): 8 BIT
Fixed Decimal Point (F): 23 BIT

Conversion Value:$(-1)^{S}$ X $(1+FX2^{-23})X2^{(E-127)}$

0< Exponent (E) < 255 → integer
Exponent (E) = 0, Fixed Decimal Point (F) = 0 → 0 (ZERO)
Exponent (E) = 0, Fixed Decimal Point (F) > 0 → Conversion value close to 0
Exponent (E) = 255, Fixed Decimal Point (F) = 0 → INFINITY
Exponent (E) = 255, Fixed Decimal Point (F) > 0 → NAN

Note

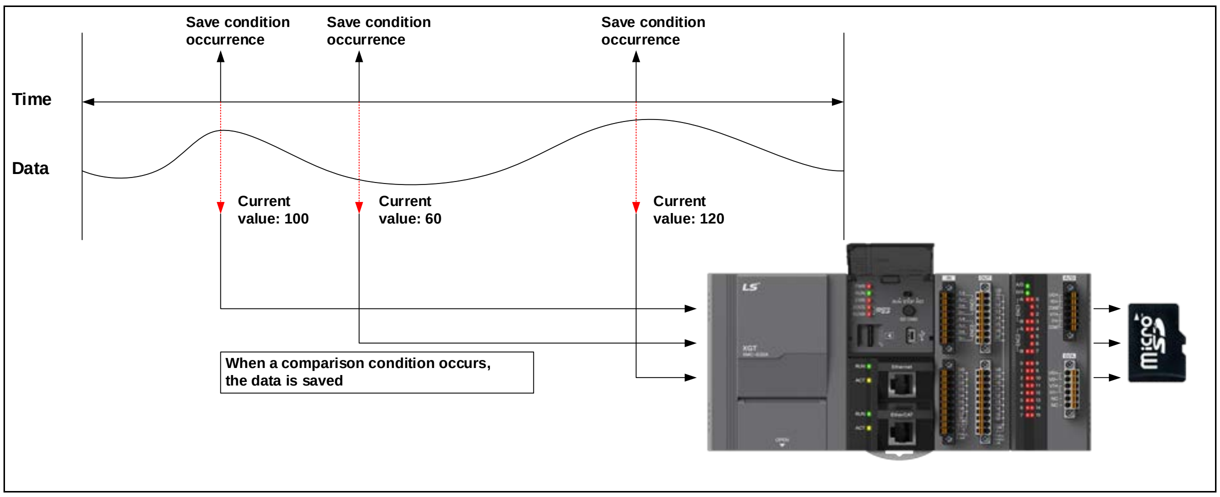
In case of REAL, LREAL types, –NaN, +NaN are saved for undefined data, and -INF, +INF character strings are saved for data with infinite range.
Please verify the data save range before use.

11.3.2 Data Save Method

The datalog function saves data using one of the three methods that follows.

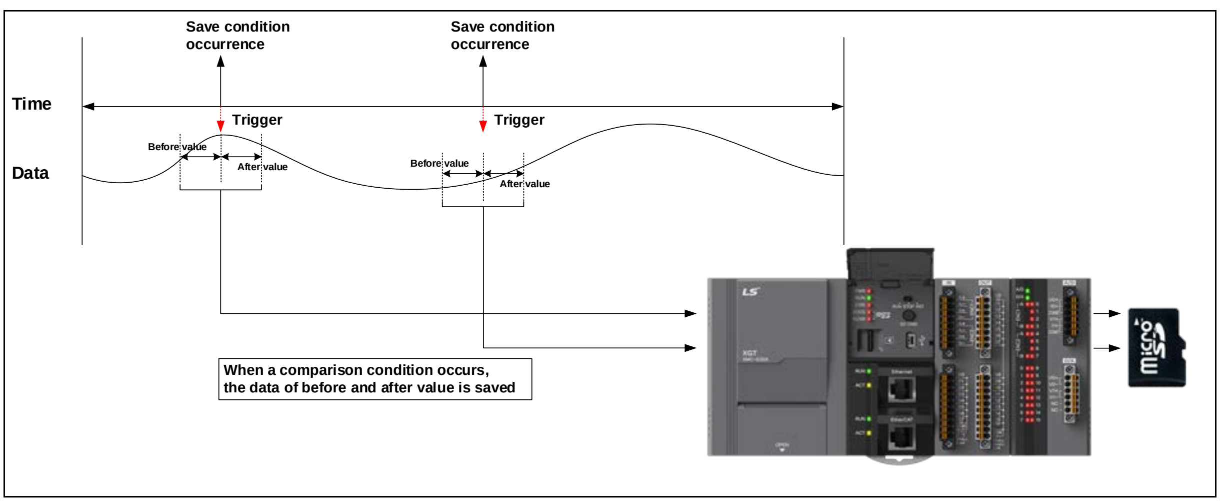
(1) Regular Save

Regular Save refers to saving data at main task or at a set interval That is, data at the time of save condition are saved, without considering the status before or after the save condition. This method is useful for collecting certain data at a certain interval.

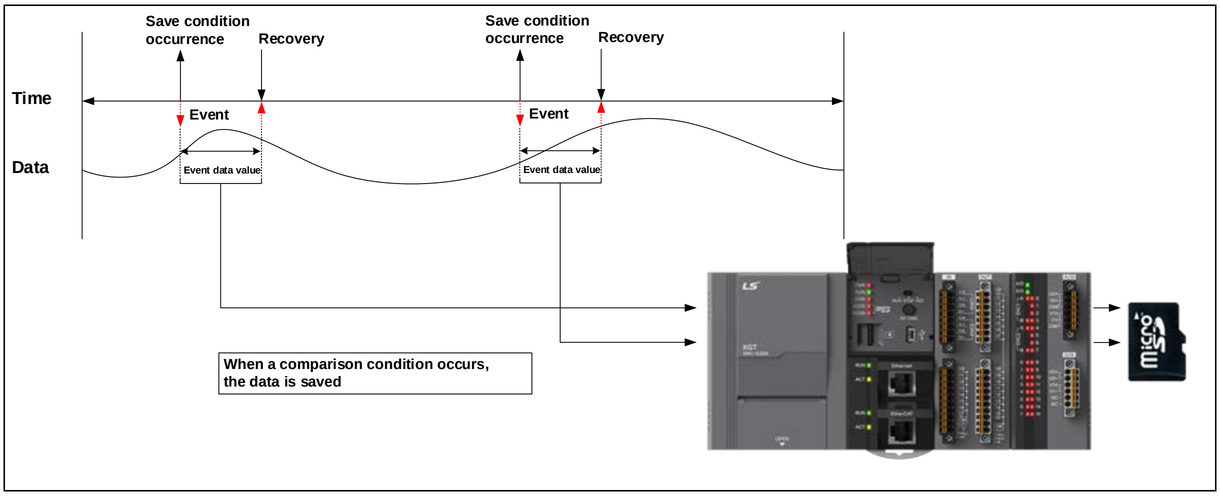


(2) Trigger Save

Trigger Save refers to saving a set number of data before and after the relevant point: the number of data are set by parameter. This method is useful when you want to view data from a certain period before and after a certain event.



(3) Event Save
Event Save refers to monitoring the device value collected, and saving the present data when a certain event condition is satisfied. This method is useful for analyzing fluctuation of event values and timing by saving data from the event occurrence to the event termination.



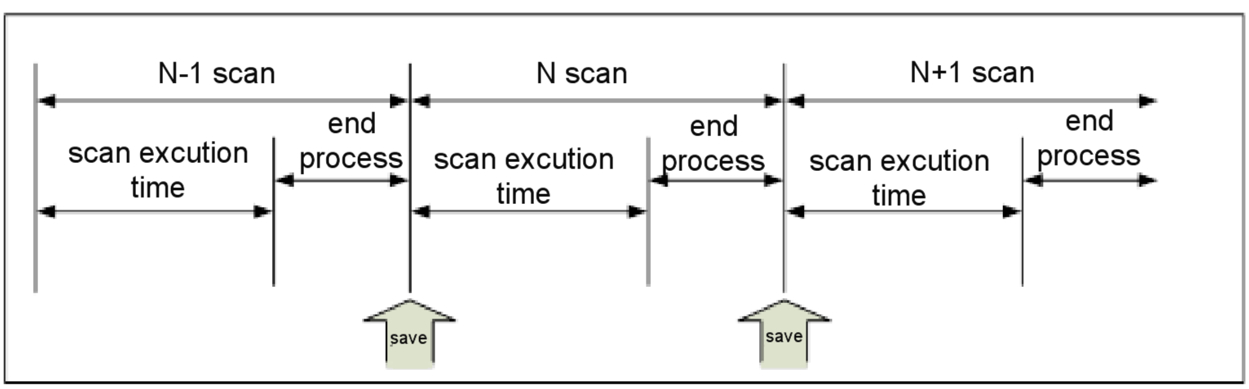
11.3.3 Data Sampling Condition

The datalog function classifies the data save conditions and intervals as follows, depending on the parameter setting.

(1) Regular Save
The following are condition setting items for Regular Save.

Setting	Operation	Note
Save at every main task	Data are saved after End of each main task	
Save at certain interval	Data are saved after END of each main task after lapse of set time	
Save at certain time	Data are saved after END of each main task after lapse of set time	

(a) Save at every scan
When using the scan interval save method, data are collected after END of each main task. If the volume of stored data is large, a scan watchdog timer error may occur.



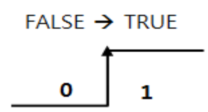
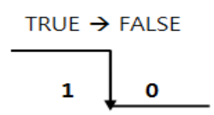
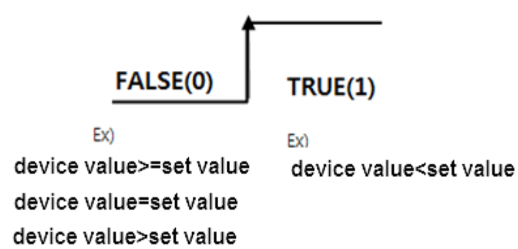
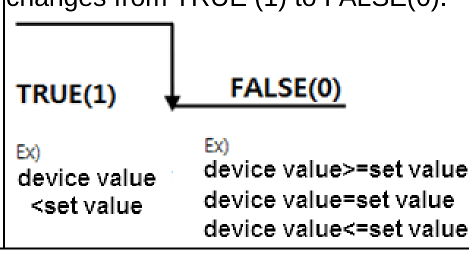
(b) Designation Cycle Save
It samples data when a set interval arrives.

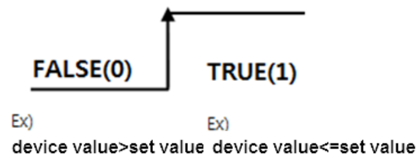
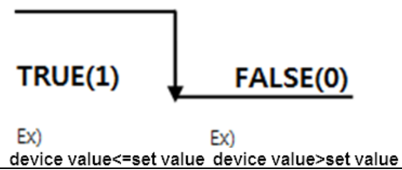
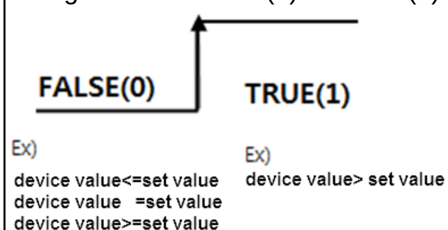
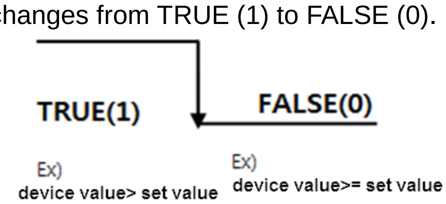
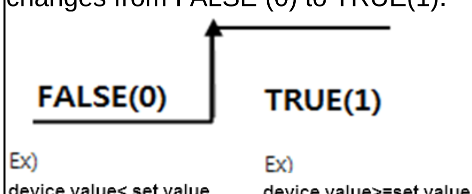
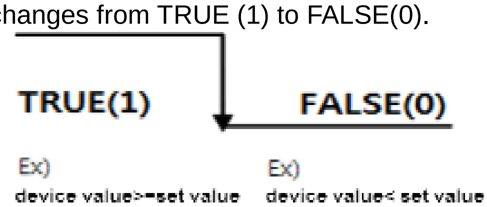
(c) Designation Time Save
It samples data when a set interval arrives.

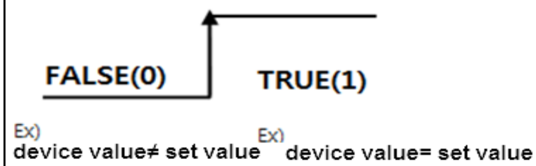
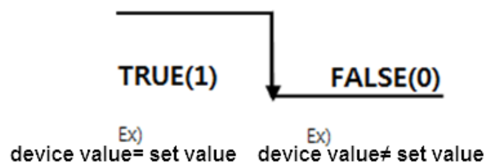
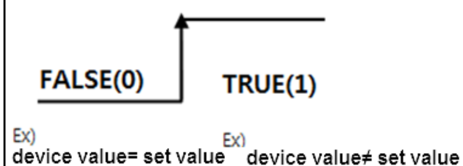
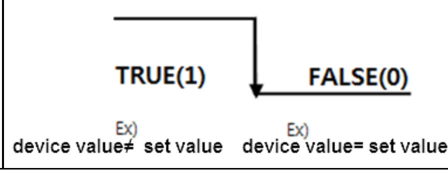
Note

1. The data collection is performed at the interval set by the parameter.
2. Each group has its buffer area, where certain data are collected an then saved into the SD memory.
3. In case of data loss, DLxx_Ovf flag will be on.

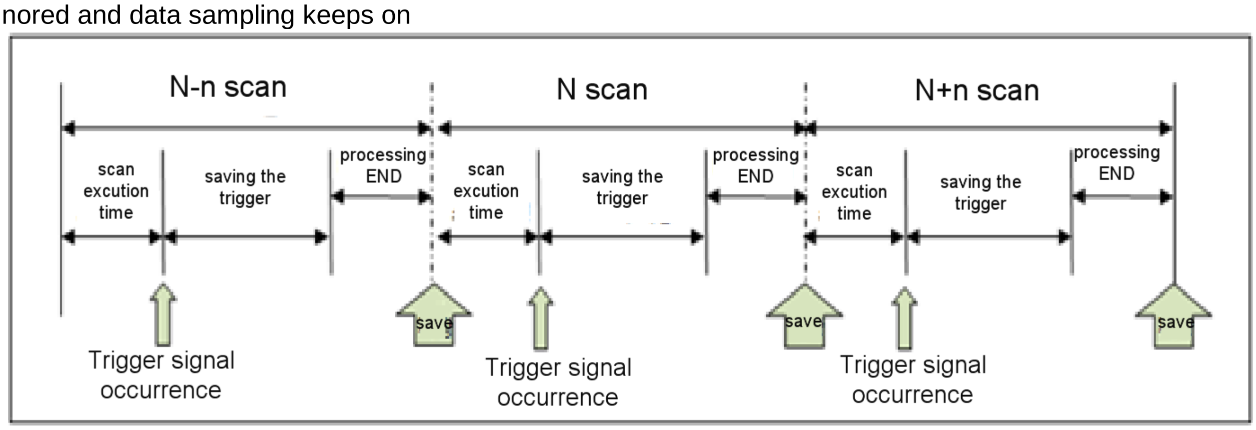
(2) Trigger Save
Save data in the preset number of collection data. The following are condition setting items for Regular Save.

<table>
<tr><th></th><th>Trigger Occurrence Condition</th><th>Device Set Condition</th><th>Operation</th><th>Note</th></tr>
<tr><td rowspan="2">BIT Condition</td><td>Elevation</td><td rowspan="2"></td><td>Saves data at elevation edge of set device bit value

</td><td rowspan="2"></td></tr>
<tr><td>Descent</td><td>Saves data at descent edge of set device bit value

</td></tr>
<tr><td rowspan="2">Word Condition</td><td>Elevation</td><td rowspan="2">Small</td><td>Samples data at the elevation edge when Device Set condition changes from FALSE (0) to TRUE (1).

</td><td></td></tr>
<tr><td>Descent</td><td>Samples data at the descent edge when Device Set Condition changes from TRUE (1) to FALSE(0).

</td><td></td></tr>
</table>

	Trigger Occurrence Condition	Device Set Condition	Operation	Note
Word Condition	Elevation	Small or Same	Samples data at the descent edge when Device Set Condition changes from FALSE (0) to TRUE(1). 	
	Descent		Samples data at the descent edge when Device Set Condition changes from TRUE (1) to FLASE(0). 	
	Elevation	Large	Samples data at the elevation edge when Device Set Condition changes from FALSE (0) to TRUE(1). 	
	Descent		Samples data at the descent edge when Device Set Condition changes from TRUE (1) to FALSE (0). 	
	Elevation	Large or Same	Samples data at the elevation edge when Device Set Condition changes from FALSE (0) to TRUE(1). 	
	Descent		Samples data at the descent edge when Device Set Condition changes from TRUE (1) to FALSE(0). 	

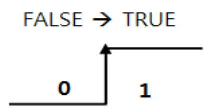
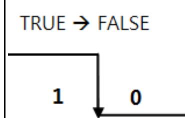
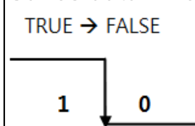
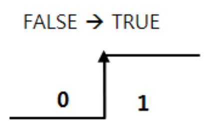
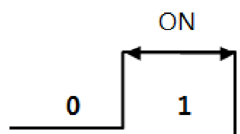
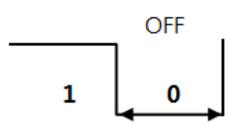
	Trigger Occurrence Condition	Device Set Condition	Operation	Note
Word Condition	Elevation	Same	Samples data at the elevation edge when Device Set Condition changes from FALSE (0) to TRUE(1). 	
	Descent		Samples data at the descent edge when Device Set Condition changes from TRUE (1) to FALSE(0). 	
	Elevation	Different	Samples data at the elevation edge when Device Set Condition changes from FALSE (0) to TRUE(1). 	
	Descent		Samples data at the descent edge when Device Set Condition changes from TRUE (1) to FALSE (0). 	

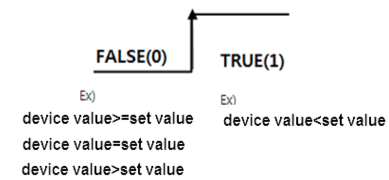
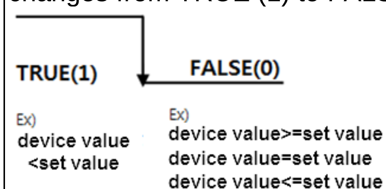
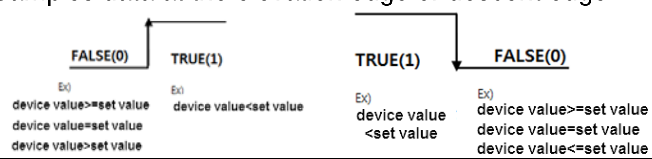
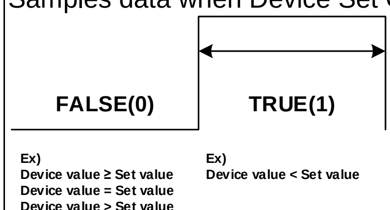
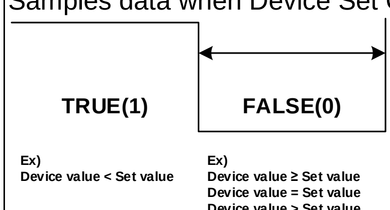
Trigger occurrence condition is decided by main task END. If trigger occurs again when data sampling, the trigger is ignored and data sampling keeps on

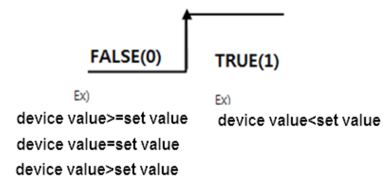
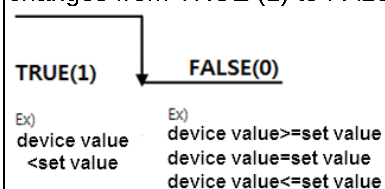
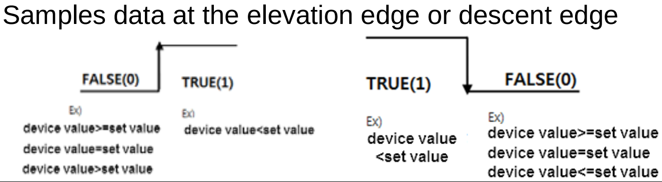
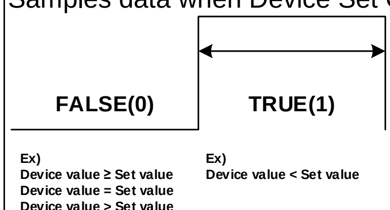
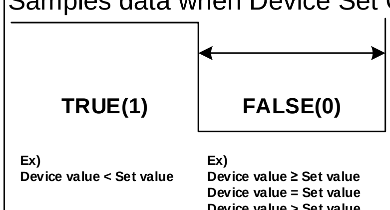


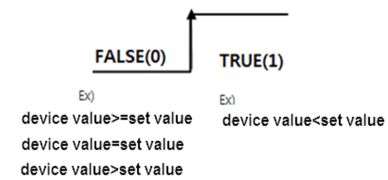
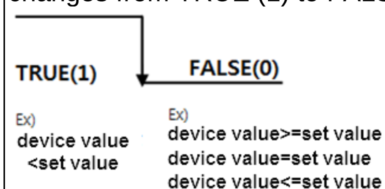
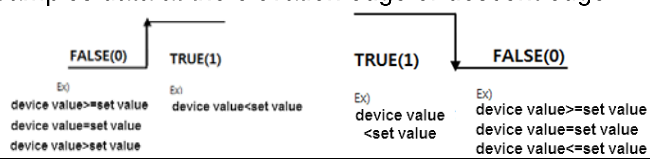
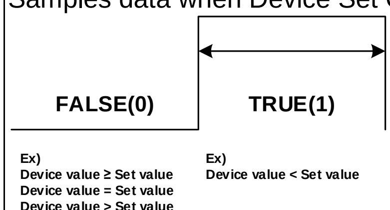
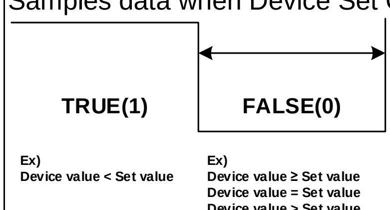
(3) Event Save

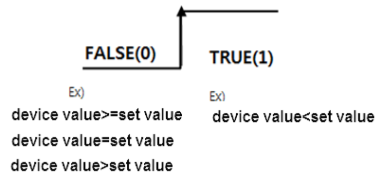
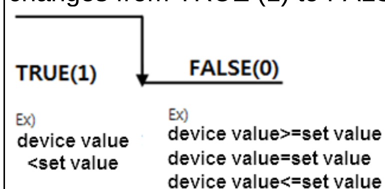
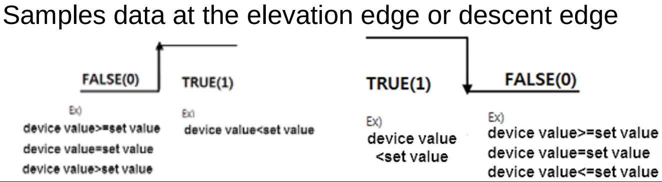
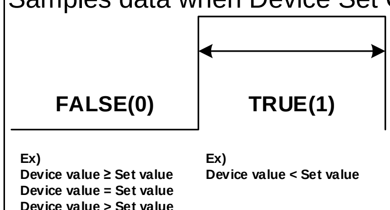
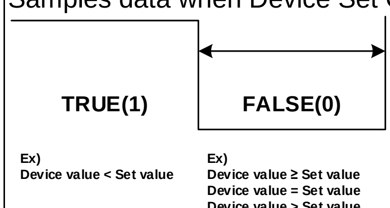
Event Save runs with similar conditions to Trigger Save. Event Save refers to saving data when the event occurs, until the conditions are not satisfied.

	Trigger Occurrence Condition	Device Set Condition	Operation	Release Value Setting
Bit Condition	Elevation		Saves data at elevation edge of set device bit value 	
	Descent		Saves data at descent edge of set device bit value 	
	Transfer		Saves data when set device bit value is transferred  or 	
	ON		Saves data when set device bit value is ON 	
	OFF		Saves data when set device bit value is OFF 	

<table>
<tr><th></th><th>Trigger Occurrence Condition</th><th>Device Set Condition</th><th>Operation</th><th>Release Value Setting</th></tr>
<tr><td rowspan="5">Word Condition</td><td>Elevation</td><td rowspan="5">Small</td><td>Samples data at the elevation edge when Device Set Condition changes from FALSE (0) to TRUE (1).

</td><td>Setting Available</td></tr>
<tr><td>Descent</td><td>Samples data at the descent edge when Device Set Condition changes from TRUE (1) to FALSE (0).

</td><td></td></tr>
<tr><td>Transfer</td><td>Samples data at the elevation edge or descent edge

</td><td></td></tr>
<tr><td>ON</td><td>Samples data when Device Set Condition is TRUE (1)

</td><td></td></tr>
<tr><td>OFF</td><td>Samples data when Device Set Condition is FALSE (0)

</td><td></td></tr>
</table>

	Trigger Occurrence Condition	Device Set Condition	Operation	Release Value Setting
Word Condition	Elevation	Small or Same	Samples data at the elevation edge when Device Set Condition changes from FALSE (0) to TRUE (1). 	Setting Available
	Descent		Samples data at the descent edge when Device Set Condition changes from TRUE (1) to FALSE (0). 	
	Transfer		Samples data at the elevation edge or descent edge 	
	ON		Samples data when Device Set Condition is TRUE (1) 	
	OFF		Samples data when Device Set Condition is FALSE (0) 	

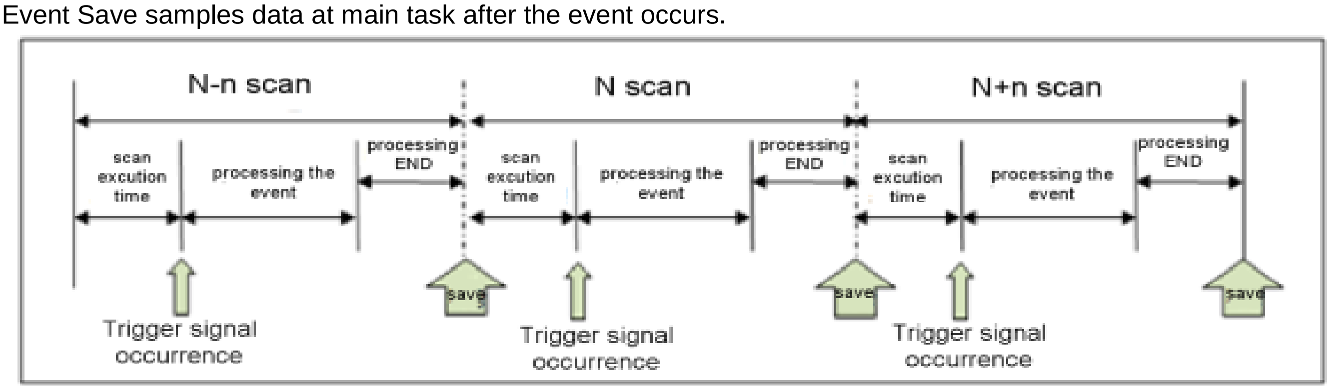
	Trigger Occurrence Condition	Device Set Condition	Operation	Release Value Setting
Word Condition	Elevation	Large	Samples data at the elevation edge when Device Set Condition changes from FALSE (0) to TRUE (1).  	Setting Available
	Descent		Samples data at the descent edge when Device Set Condition changes from TRUE (1) to FALSE (0).  	
	Transfer		Samples data at the elevation edge or descent edge  	
	ON		Samples data when Device Set Condition is TRUE (1)  	
	OFF		Samples data when Device Set Condition is FALSE (0)  	

	Trigger Occurrence Condition	Device Set Condition	Operation	Release Value Setting
Word Condition	Elevation	Large or Same	Samples data at the elevation edge when Device Set Condition changes from FALSE (0) to TRUE (1).  	Setting Available
	Descent		Samples data at the descent edge when Device Set Condition changes from TRUE (1) to FALSE (0).  	
	Transfer		Samples data at the elevation edge or descent edge  	
	ON		Samples data when Device Set Condition is TRUE (1)  	
	OFF		Samples data when Device Set Condition is FALSE (0)  	

	Trigger Occurrence Condition	Device Set Condition	Operation	Release Value Setting
Word Condition	Elevation	Same	Samples data at the elevation edge when Device Set Condition changes from FALSE (0) to TRUE (1). FALSE(0) TRUE(1) Ex) device value>=set value device value=set value device value>set value Ex) device value<set value	Setting Available
	Descent		Samples data at the descent edge when Device Set Condition changes from TRUE (1) to FALSE (0). TRUE(1) FALSE(0) Ex) device value <set value Ex) device value>=set value device value=set value device value<=set value	
	Transfer		Samples data at the elevation edge or descent edge FALSE(0) TRUE(1) Ex) device value>=set value device value=set value device value>set value Ex) device value<set value TRUE(1) FALSE(0) Ex) device value <set value Ex) device value>=set value device value=set value device value<=set value	
	ON		Samples data when Device Set Condition is TRUE (1) FALSE(0) TRUE(1) Ex) Device value ≥ Set value Device value = Set value Device value > Set value Ex) Device value < Set value	
	OFF		Samples data when Device Set Condition is FALSE (0) TRUE(1) FALSE(0) Ex) Device value < Set value Ex) Device value ≥ Set value Device value = Set value Device value > Set value	

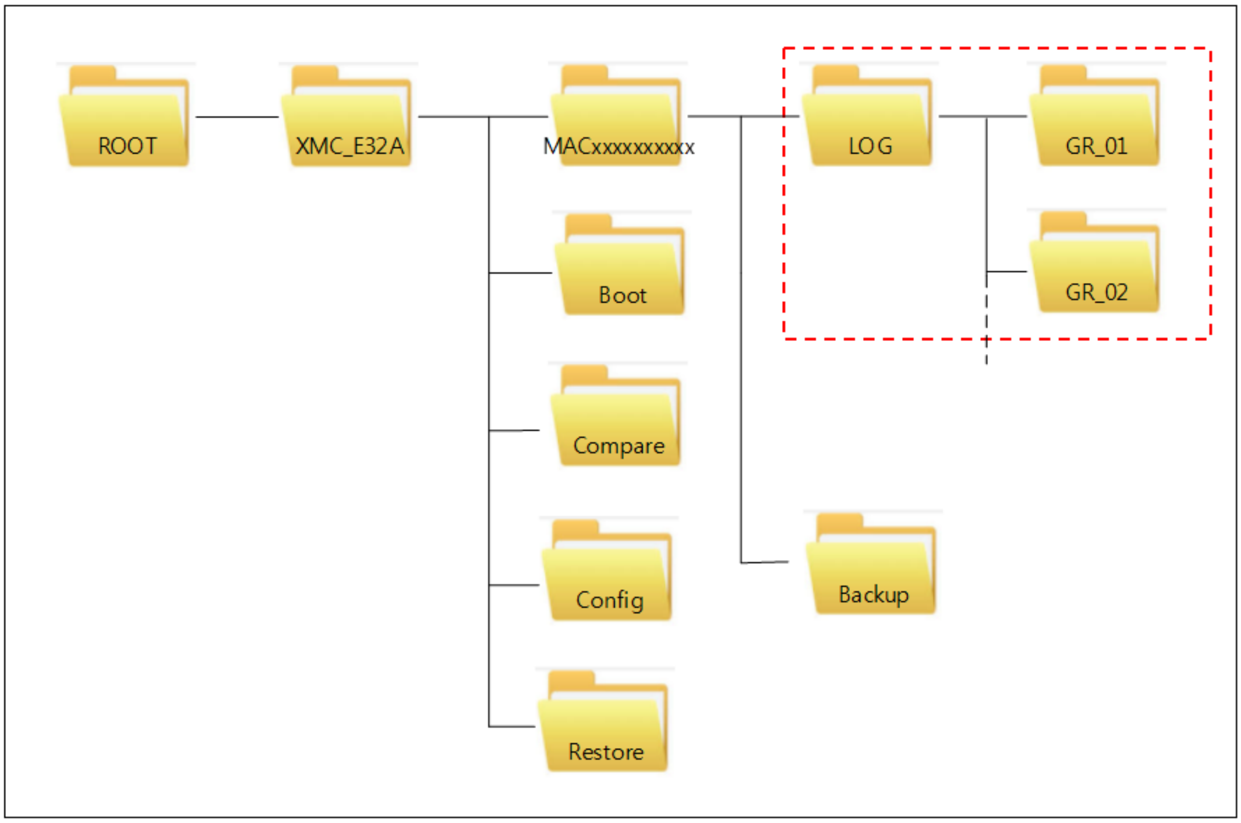
	Trigger Occurrence Condition	Device Set Condition	Operation	Release Value Setting
Word Condition	Elevation	Not Same	Samples data at the elevation edge when Device Set Condition changes from FALSE (0) to TRUE (1). FALSE(0) TRUE(1) Ex) device value>=set value, device value=set value, device value>set value Ex) device value<set value	Setting Available
	Descent		Samples data at the descent edge when Device Set Condition changes from TRUE (1) to FALSE (0). TRUE(1) FALSE(0) Ex) device value <set value Ex) device value>=set value, device value=set value, device value<=set value	
	Transfer		Samples data at the elevation edge or descent edge FALSE(0) TRUE(1) Ex) device value>=set value, device value=set value, device value>set value Ex) device value<set value TRUE(1) FALSE(0) Ex) device value <set value Ex) device value>=set value, device value=set value, device value<=set value	
	ON		Samples data when Device Set Condition is TRUE (1) FALSE(0) TRUE(1) Ex) Device value ≥ Set value, Device value = Set value, Device value > Set value Ex) Device value < Set value	
	OFF		Samples data when Device Set Condition is FALSE (0) TRUE(1) FALSE(0) Ex) Device value < Set value Ex) Device value ≥ Set value, Device value = Set value, Device value > Set value	

When Event Save method is used, data are saved after END of each main task where the set bit condition occurred. Event Save samples data at main task after the event occurs.



11.3.4 Save Folder Structure

Data saved by datalog are saved in the following file structure.



(1) Folder Name: Folder name is fixed. Creating additional folder other than the structures show in in the Figure below in the SD memory, datalog function does not show normal function. Please be careful.

(2) Data Save Folder: This folder saves log data generated by datalog. Each parameter setting group uses different folders. The file names are created in accordance with the following rules. The data folder name can be as long as 32 characters (in case of English, no space). (The folder name indicated in the folder structure diagram is arbitrary. Users can change the names.)

11.3.5 CSV File Format

CSV files generated by datalog function follow the following specifications

Items	Description
Separation Character	Comma (,)
Line Change Code	CR, LF(0x0D, 0x0A)
Character Code	ASCII Code
Field Data	Decimal, Hexadecimal, Exponent, character string
File Size	Up to 16Mbyte

Header File

Data File

	A	B	C	D	E	F
1	Remark	Project = NewPLC				
2	Remark	Filename = FILE0001.CSV				
3	Remark	Start Date = 2016/11/09/16:27:37.267				
4	Remark	Controller = XMC-E32A				
5	Remark	LogType = Trigger				
6	Remark	DataType	INT	WORD	INT	WORD
7	Remark	Device	mw1	mw2	mw5	mw6
8						
9	TIME	INDEX	DataName	DataName	DataName	DataName
10	2016/11/09/16:27:37.267	441	20521	h'1029	4137	h'1029
11	2016/11/09/16:27:37.271	442	20522	h'102A	4138	h'102A

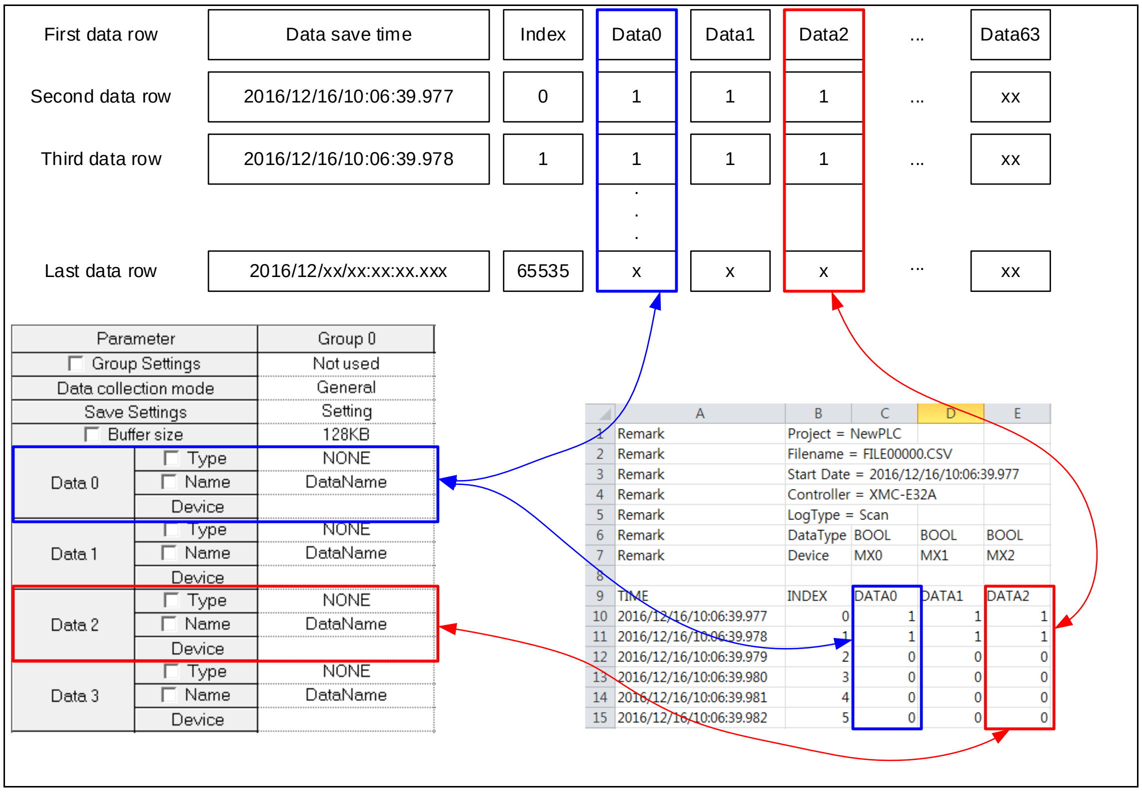
(1) Header File Structure

The header structure of datalog files saved in the SD memory is as follows

Remark	Project Name
Remark	Save File Name
Remark	File Creation Time
Remark	Motion Controller Type
Remark	Datalog Save Type
Remark	Data Conversion Type
Remark	Device

(2) Data File Structure

The internal structure of datalog files saved in the SD memory is as follows



Note

1. Index indicates the number of saved data
2. Data 0, Data 1, …, Data 63 indicate data names

Note

When you read a CSV file in Microsoft Office Excel, several data may be displayed in a single cell. This is because you are required to use the “symbol as the text qualifier” when opening the CSV file in Excel. In this case, if you open the CSV file in the following order, it will be displayed normally.

1. Select the text in the data menu after executing Excel and then select the CSV file you want to open
2. Select [Comma] as a separator and [None] as a text in the Text Wizard - Step 2 and then click Finish

(3) Data File Item Description

(a) First Data Line

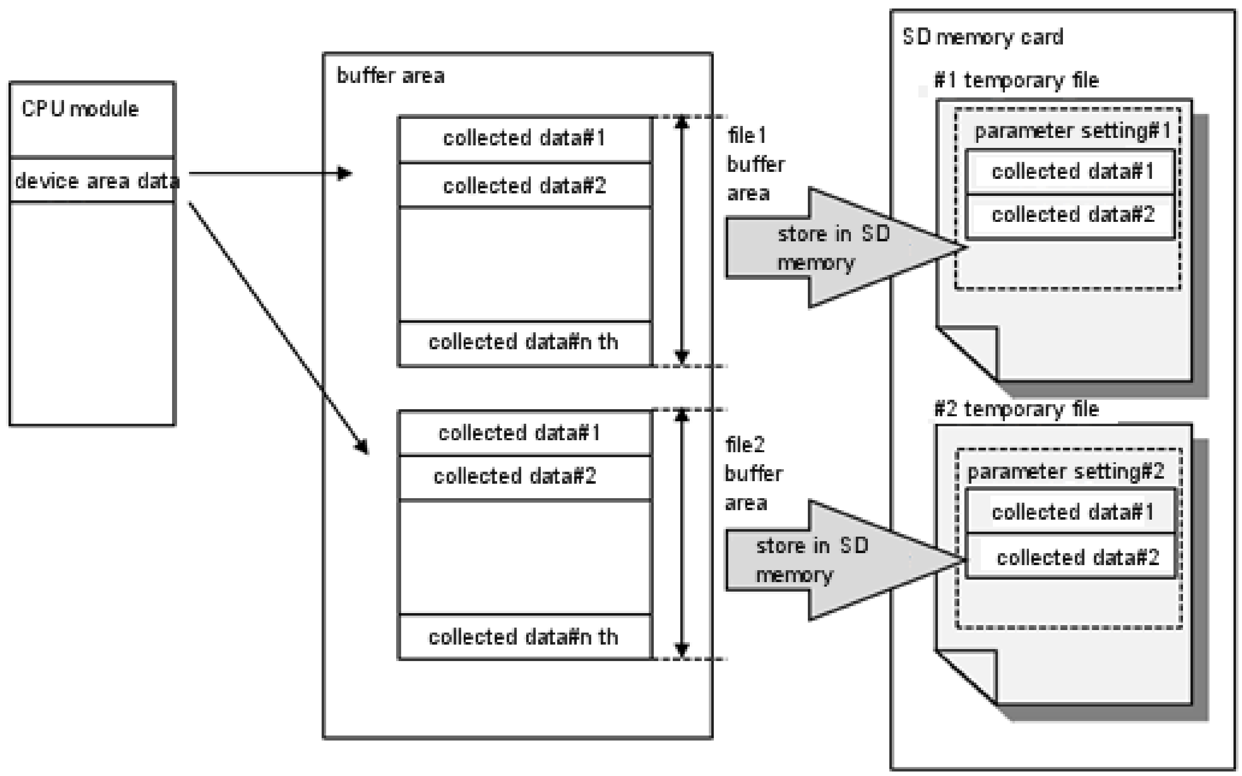
String Name	Output	Size (Word)
Temporary String	Indicates date and time with fixed characters	5
Index String	Indicates index name	2
Data String	Outputs the data name designated at data setting	1-64 (Depends on parameter setting)

(b) Data Row Repeat

Column Name	Output		Size (Byte)
Date and Time Column	String is output using the data output format set at CSV Output Setting. Ex) 2014/09/17 10:15:20:243		24
Index Column	Outputs counted numbers starting from 0 and up.		10
Data Column	BOOL	0 or 1	2
	BYTE	00 ~ FF	3
	WORD	0000 ~ FFFF	5
	DWORD	00000000 ~ FFFFFFFF	9
	LWORD	00000000 00000000 ~ FFFFFFFF FFFFFFFF	17
	SINT	-128 ~ 127	5
	INT	-32,768 ~ 32,767	7
	DINT	-2,147,483,648 ~ 2,147,483,647	12
	LINT	-576,460,752,303,423,488 ~ 576,460,752,303,423,487	21
	USINT	0 ~ 255	4
	UINT	0 ~ 65,535	6
	UDINT	0 ~ 4,294,967,295	11
	ULINT	0 ~ 1,152,921,504,606,846,975	20
	REAL	-3.402823466e+038 ~ -1.175494351e-038 or 0 or 1.175494351e-038 ~ 3.402823466e+038	17
	LREAL	-1.7976931348623157e+308 ~ -2.2250738585072014e-308 or 0 or 2.2250738585072014e-308 ~ 1.7976931348623157e+308	24
	STRING	Fixed Character (up to 32 characters)	33

11.3.6 How to Save CSV

Motion controller collects data whenever the sampling condition occurs, saves them into the SD memory as CSV files. When the data meet file conversion time, motion controller generates a new file in the SD memory card to perform data saving.



(1) File Conversion Test
Temporary files are converted to CSV files at the following points

At saving	Setting Range
When the designated number of saves have been completed in the temporary file	5,000 ~ 50,000
When the temporary file reaches the designated size	10KB ~ 16,384KB

(2) Operation in Case of Exceeding the Number of Save Files
When the number of maximum saved files set by the parameter is exceeded, the following run occurs in accordance with the set runs in case of file excess.

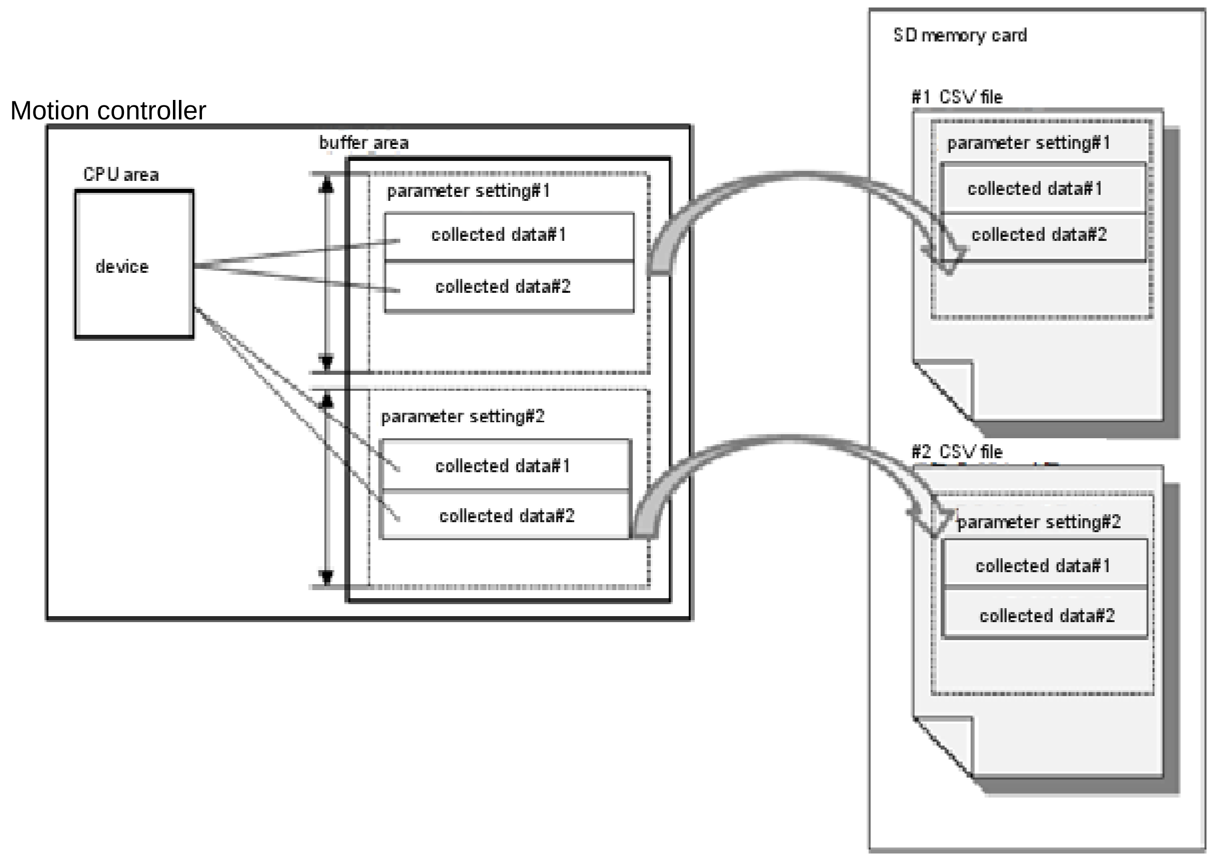
Operation Setting in Case of Excess	Operation	Note
Maintains the latest history	Overwrites and saves new data over the oldest file	
Maintains the initial history	Performs no more file saving	

Note

In case the SD memory is not capable of saving 256 files and the storage is full, it maintains the initial history saves file up to the full storage of SD memory, then stops data saving regardless of the the [History Setting] value in the parameter and generate the error code 6(%KW522)

11.3.7 Buffer Memory

Motion controller has an internal buffer memory for datalog function. Buffer memory refers to a volatile memory which temporarily stores collected data before saving them into the temporary file in the SD memory.



In accordance with the set sampling condition, the collected data are stored in the buffer memory first and then saved in to the temporary memory of the SD memory card when datalog condition occurs. Therefore, setting too fast data sampling condition or sampling too much data, data loss can be caused by buffer memory excess.

In case of data loss, **'C'** string is stored together with the loss data. In this case, adjust the storage period in the datalog parameter, or reduce the amount of collected data to prevent data loss.

11.3.8 Data Omission

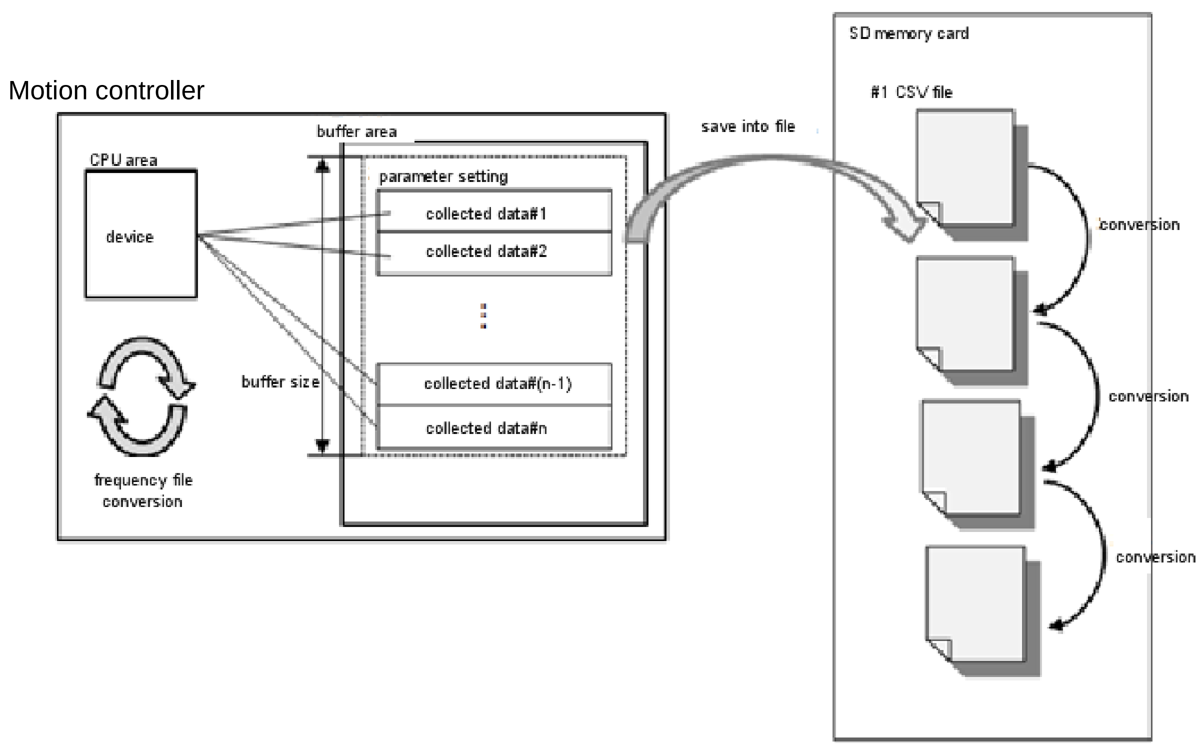
Data omission refers to situation where normal data collection is not possible. If data collection interval is set too short, data sampling might not be performed at every set interval, which in turn might cause data omission. Cases include the following.

(1) Buffer Excess

If data sampling condition is set too fast or too much data are being sampled, the speed of saving buffer memory values into the temporary file in the SD memory may be slower than the data collection speed, which causes the buffer storage to be exceeded and data omission.

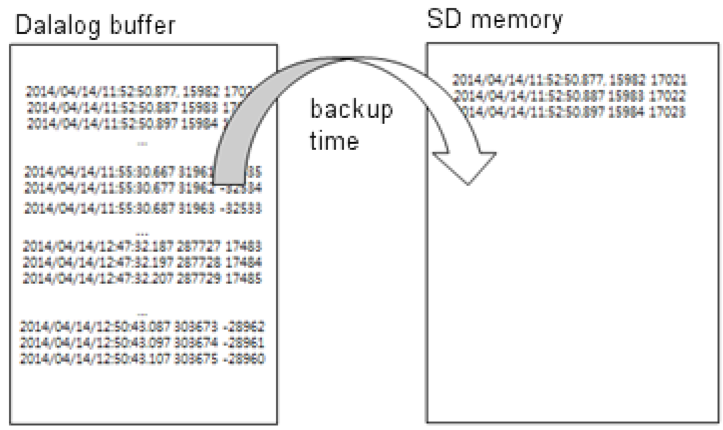
(2) Too Frequent File Conversion

Upon occurrence of file conversion condition, the temporary file should be converted to CSV file to create a new temporary file. Meanwhile, the buffer memory values cannot be saved into the temporary file. Therefore, too frequent occurrence of file conversion condition may cause the buffer memory storage to be exceeded, and thus leading to data omission.

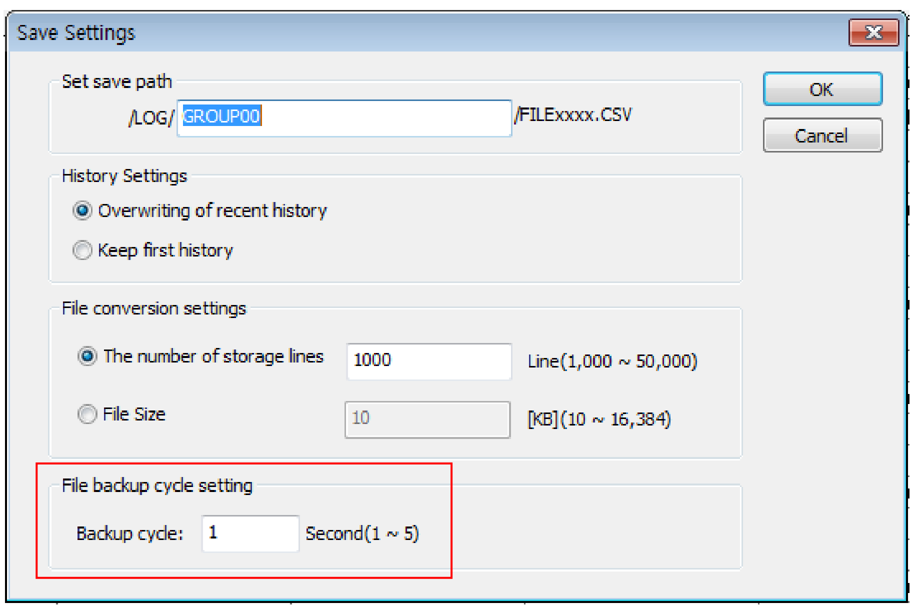


11.3.9 Files Backup Cycle

Data collected by datalog are not directly saved into the SD memory. They are saved into the designated buffer, and later saved in to the SD memory when a certain volume (4Kbyte) has been collected. When the data save interval is long and the volume of data to collect is not large, it takes a lot of time to save data into the SD memory. If collected data are saved only in the buffer before sudden shutoff or reset occurs, the saved data are all lost.



To prevent this, the collected data need to be saved into the data at certain intervals regardless of the storage. The data saved into the SD memory is not lost even in case of sudden power change. Backup time can be set at from 1 to 5 seconds. However, setting too short backup time may affect datalog performance.

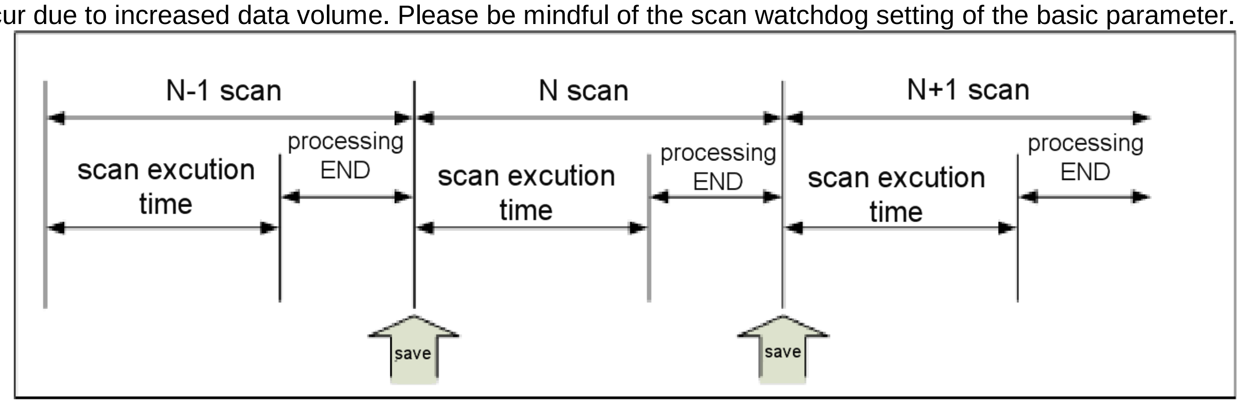


11.4 Regular Save

Among internal datalog functions of motion controller, Regular Save runs in two methods: Main task Save and Save at Designated Interval
Main task Saves refer to saving data at main task, and Save at Designated Interval refers to saving data at an interval set by the user.

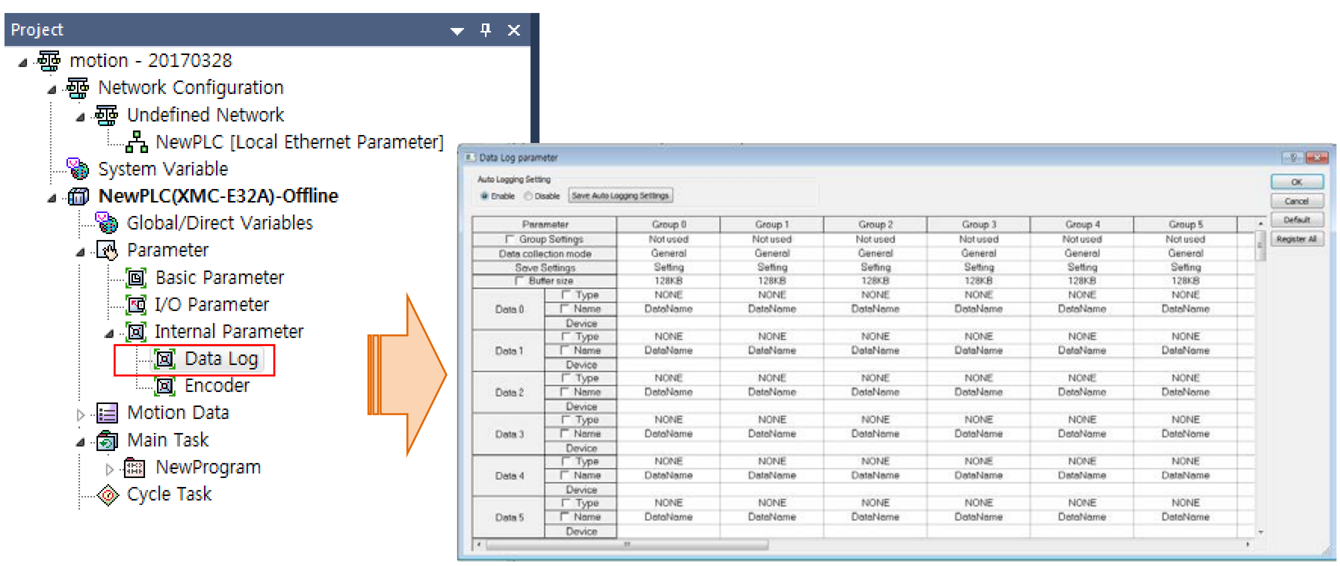
11.4.1 Save Method

(1) Operation Description
Among internal datalog functions of motion controller, Scan Saves refer to saving data at main task into the SD memory. When using the scan interval save method, data are saved after END of each main task. The collected data are accumulated in the motion controller internal buffer. When a certain amount is accumulated, these are saved into the SD memory. If the set interval is too short or the data to collect is too large, a scan watchdog timer error may occur due to increased data volume. Please be mindful of the scan watchdog setting of the basic parameter.

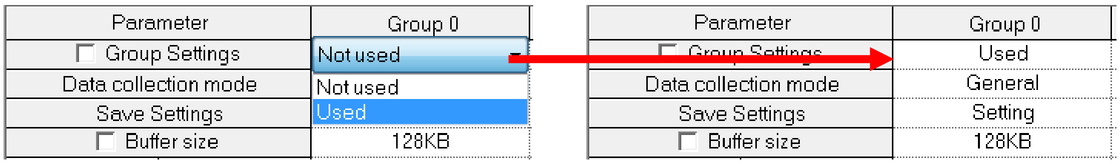


(2) Setting Method

(a) Choose XG5000 – [Project Window] - [internal parameter] - [datalog]
This activates the datalog parameter setting window.



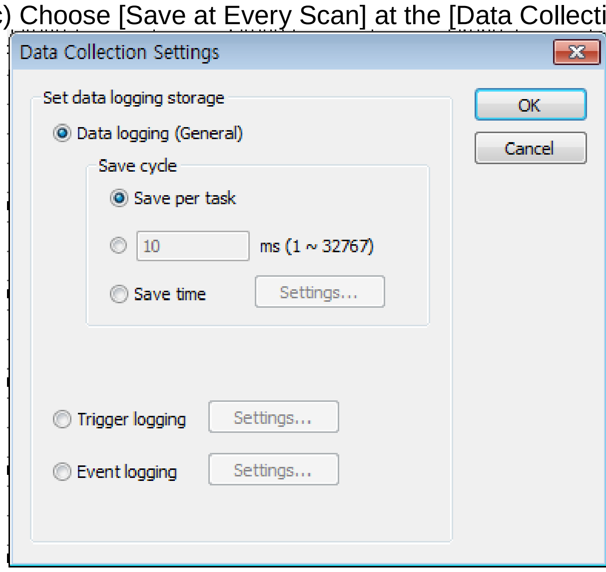
(b) Set the group to use on the datalog parameter window.



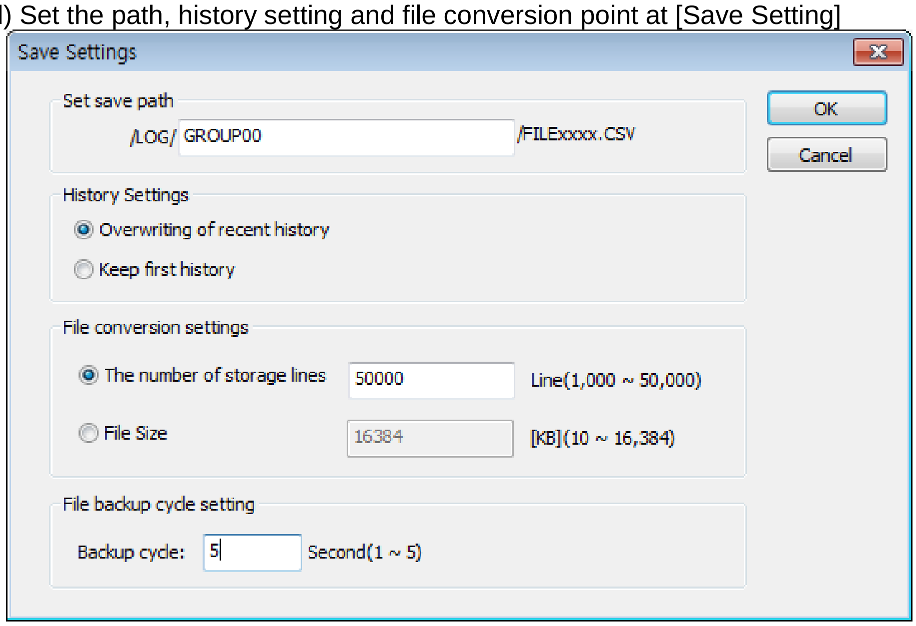
Note

It runs when both the datalog parameter and the datalog EN flag are set. In case either condition is omitted, the datalog run will not progress. Please verify whether both the datalog parameter and the datalog EN flag are set. (See 11.10, Flag List)

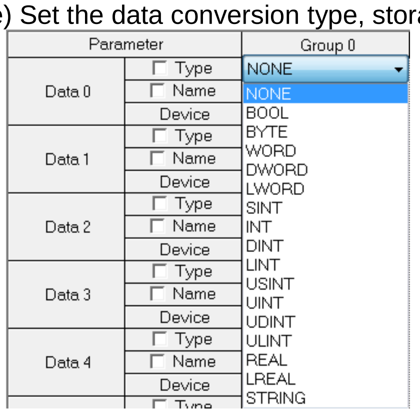
(c) Choose [Save at Every Scan] at the [Data Collection Method]

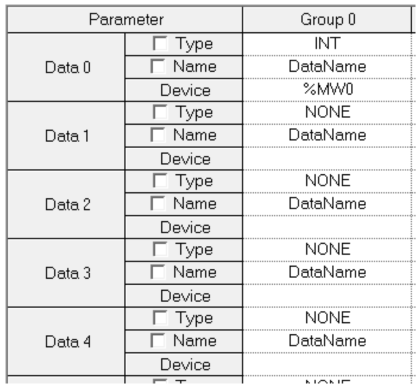


(d) Set the path, history setting and file conversion point at [Save Setting]



(e) Set the data conversion type, storage device and name





(f) Connect the SD memory card, and turn on the Datalog Enable Flag (%KW8224) when the _DL_Rdy(%KX8800) Flag is On to activate the function. Datalog will not be activated if the Enable Flag is ON while _DL_Rdy(%KX8800) Flag is OFF.

The following are Enable Flags for each datalog group

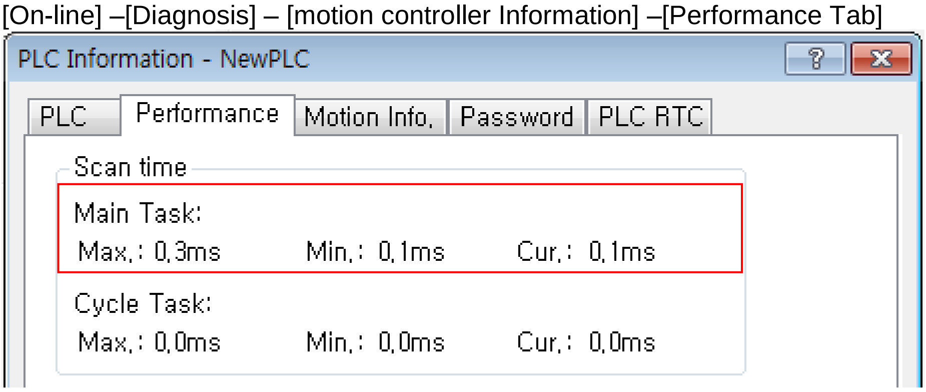
Item	Memory allocation	Type	Description
-	%KW514	WORD	Datalog Enable Flags
_DL00_Enable	%KX8224	BOOL	Group 00 Enable Flag 1: Operation, 0: Stop
_DL01_Enable	%KX8225	BOOL	Group 01 Enable Flag 1: Operation, 0: Stop
_DL02_Enable	%KX8226	BOOL	Group 02 Enable Flag 1: Operation, 0: Stop
_DL03_Enable	%KX8227	BOOL	Group 03 Enable Flag 1: Operation, 0: Stop
_DL04_Enable	%KX8228	BOOL	Group 04 Enable Flag 1: Operation, 0: Stop
_DL05_Enable	%KX8229	BOOL	Group 05 Enable Flag 1: Operation, 0: Stop
_DL06_Enable	%KX8230	BOOL	Group 06 Enable Flag 1: Operation, 0: Stop
_DL07_Enable	%KX8231	BOOL	Group 07 Enable Flag 1: Operation, 0: Stop
_DL08_Enable	%KX8232	BOOL	Group 08 Enable Flag 1: Operation, 0: Stop
_DL09_Enable	%KX8233	BOOL	Group 09 Enable Flag 1: Operation, 0: Stop
_DL10_Enable	%KX8234	BOOL	Group 10 Enable Flag 1: Operation, 0: Stop
_DL11_Enable	%KX8235	BOOL	Group 11 Enable Flag 1: Operation, 0: Stop
_DL12_Enable	%KX8236	BOOL	Group 12 Enable Flag 1: Operation, 0: Stop
_DL13_Enable	%KX8237	BOOL	Group 13 Enable Flag 1: Operation, 0: Stop
_DL14_Enable	%KX8238	BOOL	Group 14 Enable Flag 1: Operation, 0: Stop
_DL15_Enable	%KX8239	BOOL	Group 15 Enable Flag 1: Operation, 0: Stop

OFF the datalog Enable Flag to stop data saving.

When the SD memory still has data to save, the Log Ending _DLxx_Stoping (%KX8963) flag turns ON, and back to OFF once all data are saved.

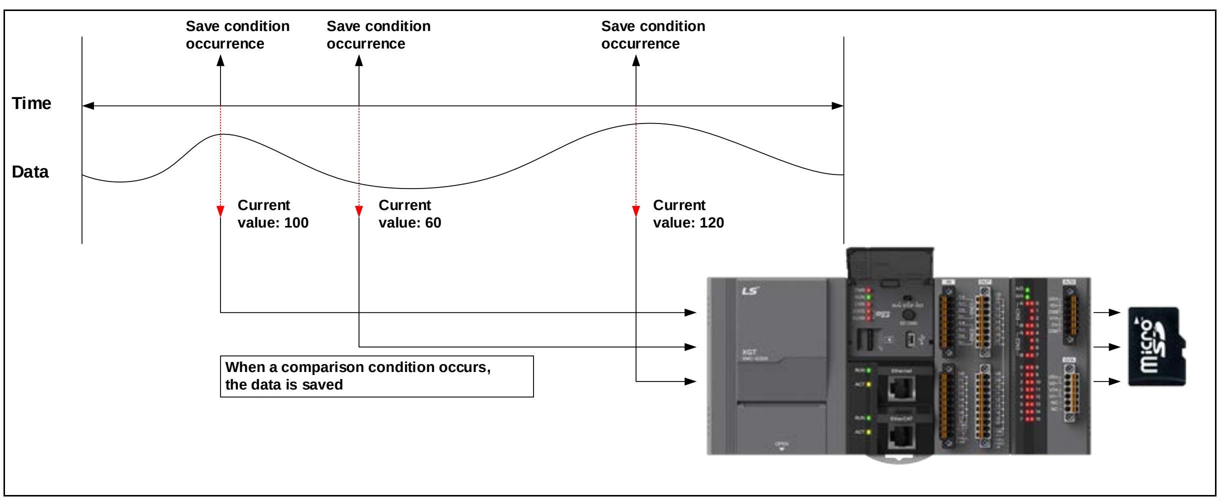
Note

When using Main task save, set the datalog parameters by referring to main task performance cycle. Setting too much data and too fast interval may cause data loss.
- Main task run time can be verified from the following menu.
  [On-line] –[Diagnosis] – [motion controller Information] –[Performance Tab]



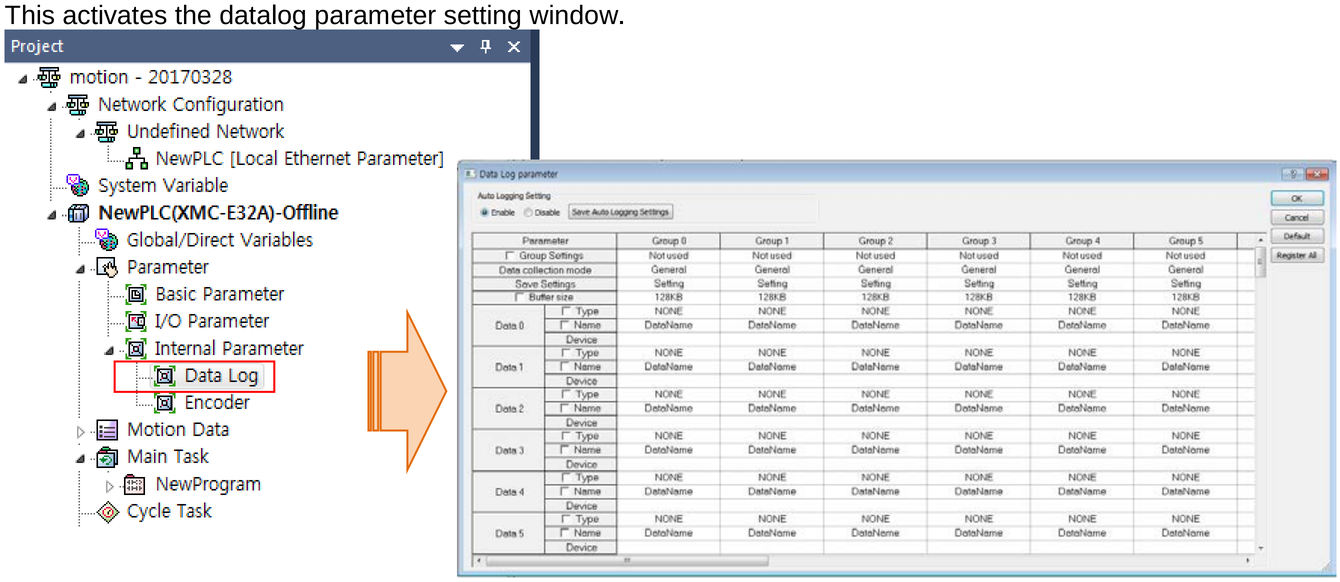
11.4.2 Save at Designated Interval

(1) Description of operation
Save at Designated Interval refers to saving data at intervals set by the user. It is different from Main task Save in that the former collects data at certain intervals, and is capable of saving data that change at certain intervals at more accurate points.

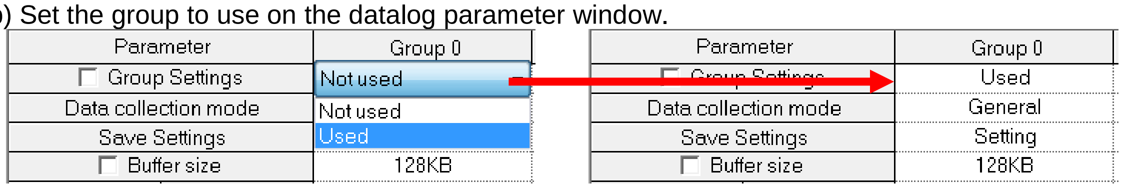


(2) Setting Method

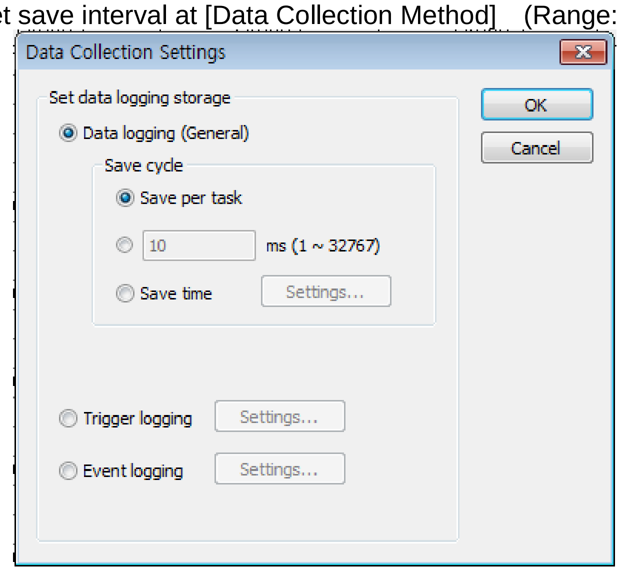
(a) Choose XG5000 – [Project Window] - [internal parameter] - [datalog]
This activates the datalog parameter setting window.



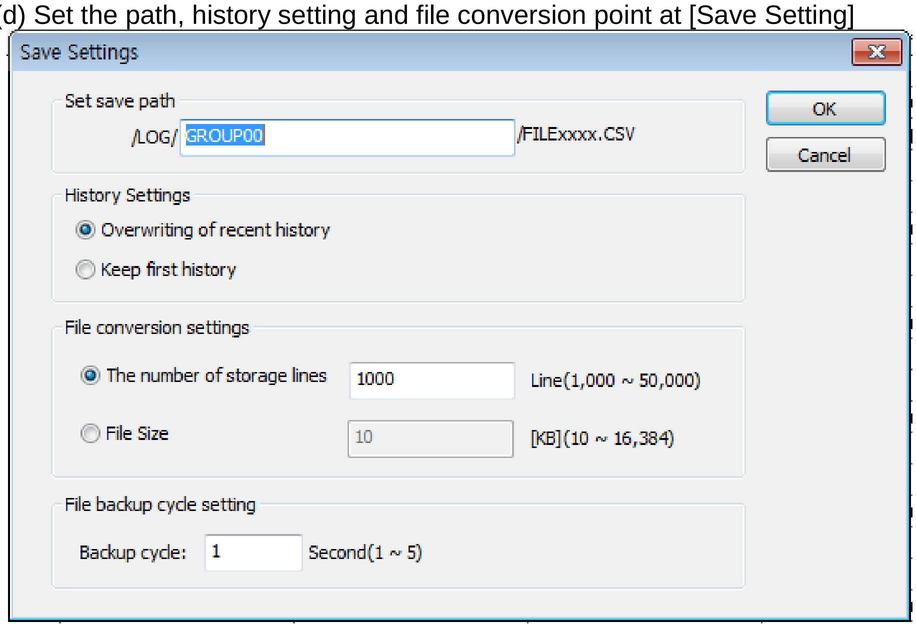
(b) Set the group to use on the datalog parameter window.



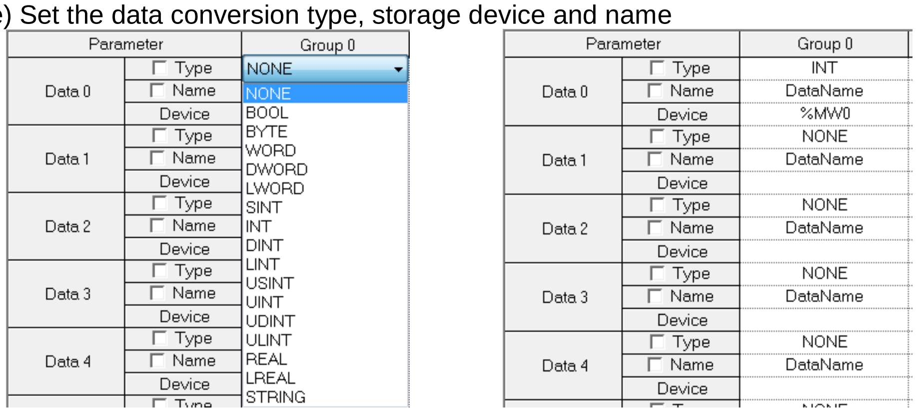
(c) Set save interval at [Data Collection Method]   (Range: 1~32,767ms)



(d) Set the path, history setting and file conversion point at [Save Setting]



(e) Set the data conversion type, storage device and name



(f) Connect the SD memory card, and turn on the Datalog Enable Flag (%KW8224) when the _DL_ Rdy(%KX8800) Flag is On to activate the function. Datalog will not be activated if the Enable Flag is ON while _DL_ Rdy (%KX8800) Flag is OFF.

The following are Enable Flags for each datalog group

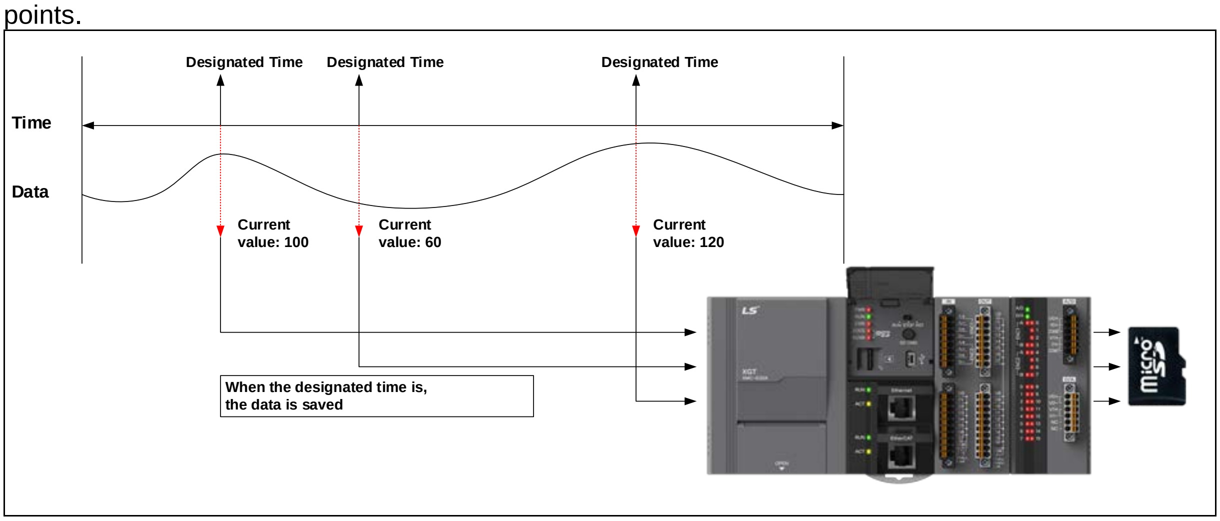
Item	Memory allocation	Type	Description
-	%KW514	WORD	Datalog Enable Flags
_DL00_Enable	%KX8224	BOOL	Group 00 Enable Flag 1: Operation, 0: Stop
_DL01_Enable	%KX8225	BOOL	Group 01 Enable Flag 1: Operation, 0: Stop
_DL02_Enable	%KX8226	BOOL	Group 02 Enable Flag 1: Operation, 0: Stop
_DL03_Enable	%KX8227	BOOL	Group 03 Enable Flag 1: Operation, 0: Stop
_DL04_Enable	%KX8228	BOOL	Group 04 Enable Flag 1: Operation, 0: Stop
_DL05_Enable	%KX8229	BOOL	Group 05 Enable Flag 1: Operation, 0: Stop
_DL06_Enable	%KX8230	BOOL	Group 06 Enable Flag 1: Operation, 0: Stop
_DL07_Enable	%KX8231	BOOL	Group 07 Enable Flag 1: Operation, 0: Stop
_DL08_Enable	%KX8232	BOOL	Group 08 Enable Flag 1: Operation, 0: Stop
_DL09_Enable	%KX8233	BOOL	Group 09 Enable Flag 1: Operation, 0: Stop
_DL10_Enable	%KX8234	BOOL	Group 10 Enable Flag 1: Operation, 0: Stop
_DL11_Enable	%KX8235	BOOL	Group 11 Enable Flag 1: Operation, 0: Stop
_DL12_Enable	%KX8236	BOOL	Group 12 Enable Flag 1: Operation, 0: Stop
_DL13_Enable	%KX8237	BOOL	Group 13 Enable Flag 1: Operation, 0: Stop
_DL14_Enable	%KX8238	BOOL	Group 14 Enable Flag 1: Operation, 0: Stop
_DL15_Enable	%KX8239	BOOL	Group 15 Enable Flag 1: Operation, 0: Stop

OFF the datalog Enable Flag to stop data saving.
When the SD memory still has data to save, the _DLxx_Stoping (%KX8963) turns ON, and back to OFF once all data are saved.

11.4.3 Save at Designated Time

(1) Description of operation

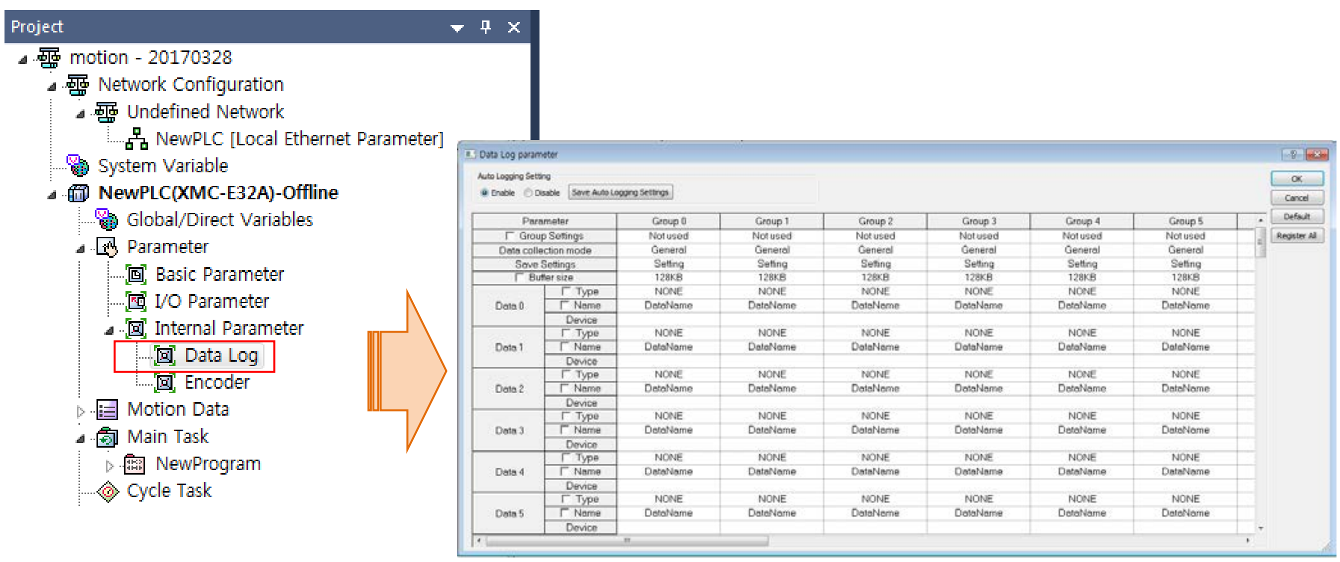
Save at Designated Interval refers to saving data at Designated Time set by the user. It is different from Designed Interval Save in that the former collects data at certain intervals, and is capable of saving data at more accurate points.



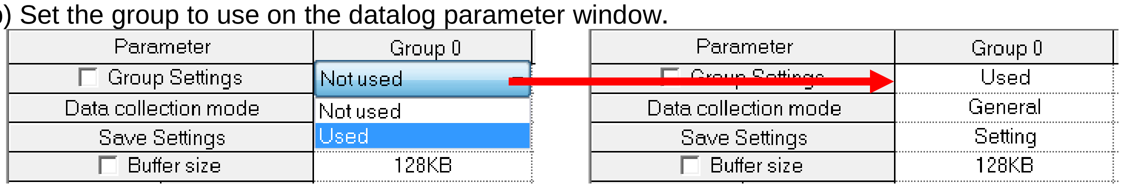
(2) Setting Method

(a) Choose XG5000 – [Project Window] - [internal parameter] - [datalog]

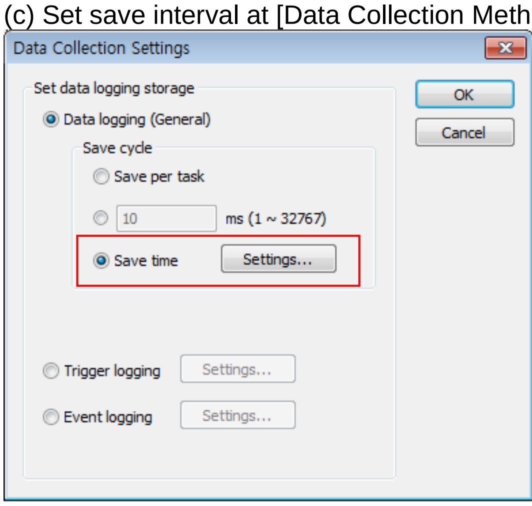
This activates the datalog parameter setting window.



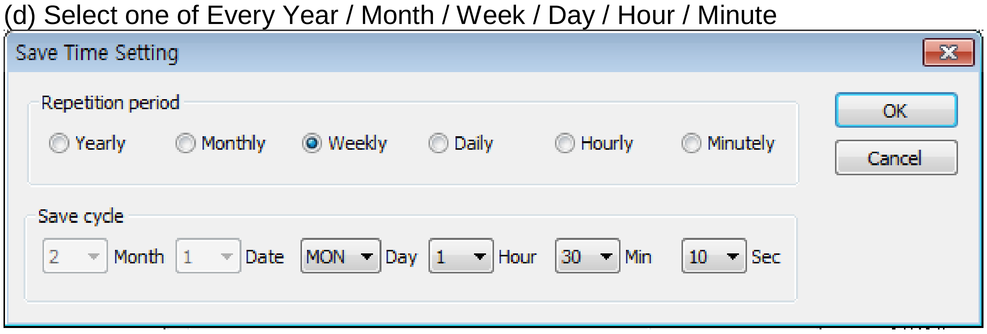
(b) Set the group to use on the datalog parameter window.



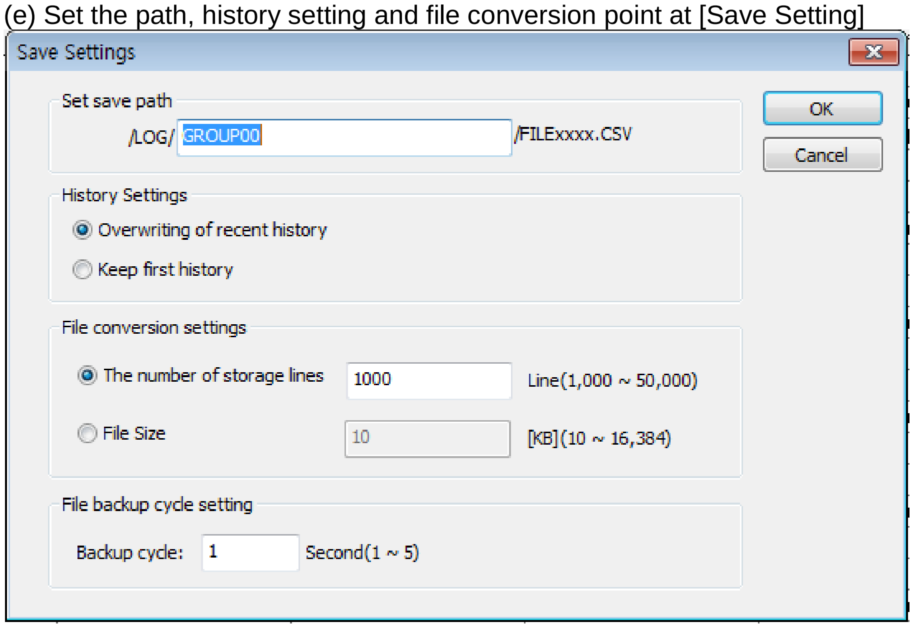
(c) Set save interval at [Data Collection Method] (Range: 1~32,767ms)



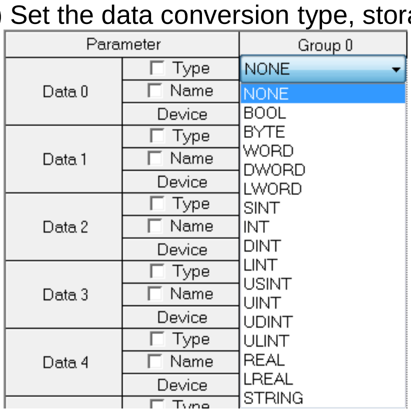
(d) Select one of Every Year / Month / Week / Day / Hour / Minute

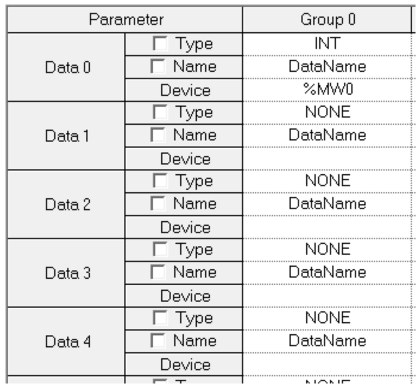


(e) Set the path, history setting and file conversion point at [Save Setting]



(f) Set the data conversion type, storage device and name





(g) Connect the SD memory card, and turn on the Datalog Enable Flag (%KW8224) when the _DL_ Rdy (%KX8800) Flag is On to activate the function. Datalog will not be activated if the Enable Flag is ON while _DL_ Rdy (%KX8800) Flag is OFF.

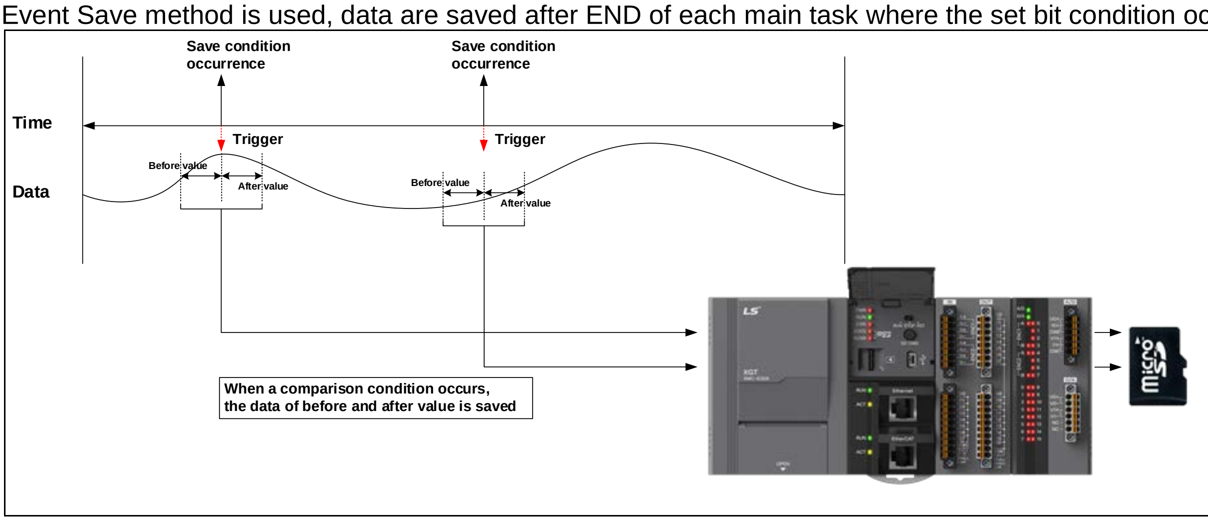
The following are Enable Flags for each datalog group

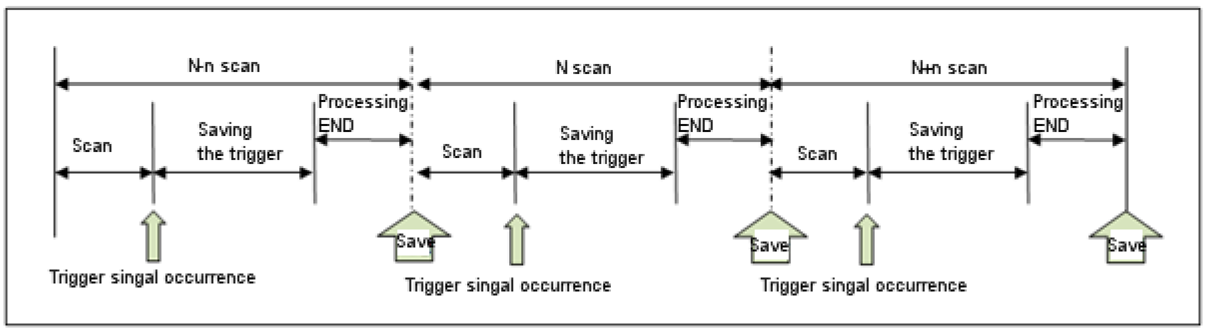
Item	Memory allocation	Type	Description
-	%KW514	WORD	Datalog Enable Flags
_DL00_Enable	%KX8224	BOOL	Group 00 Enable Flag 1: Operation, 0: Stop
_DL01_Enable	%KX8225	BOOL	Group 01 Enable Flag 1: Operation, 0: Stop
_DL02_Enable	%KX8226	BOOL	Group 02 Enable Flag 1: Operation, 0: Stop
_DL03_Enable	%KX8227	BOOL	Group 03 Enable Flag 1: Operation, 0: Stop
_DL04_Enable	%KX8228	BOOL	Group 04 Enable Flag 1: Operation, 0: Stop
_DL05_Enable	%KX8229	BOOL	Group 05 Enable Flag 1: Operation, 0: Stop
_DL06_Enable	%KX8230	BOOL	Group 06 Enable Flag 1: Operation, 0: Stop
_DL07_Enable	%KX8231	BOOL	Group 07 Enable Flag 1: Operation, 0: Stop
_DL08_Enable	%KX8232	BOOL	Group 08 Enable Flag 1: Operation, 0: Stop
_DL09_Enable	%KX8233	BOOL	Group 09 Enable Flag 1: Operation, 0: Stop
_DL10_Enable	%KX8234	BOOL	Group 10 Enable Flag 1: Operation, 0: Stop
_DL11_Enable	%KX8235	BOOL	Group 11 Enable Flag 1: Operation, 0: Stop
_DL12_Enable	%KX8236	BOOL	Group 12 Enable Flag 1: Operation, 0: Stop
_DL13_Enable	%KX8237	BOOL	Group 13 Enable Flag 1: Operation, 0: Stop
_DL14_Enable	%KX8238	BOOL	Group 14 Enable Flag 1: Operation, 0: Stop
_DL15_Enable	%KX8239	BOOL	Group 15 Enable Flag 1: Operation, 0: Stop

OFF the datalog Enable Flag to stop data saving.
When the SD memory still has data to save, the _DLxx_Stoping (%KX8963) turns ON, and back to OFF once all data are saved.

11.5 Trigger Save

Trigger Save refers to saving a set number of data before and after the relevant point: the number of data is set by parameter. This method is useful when you want to view data from a certain period before and after a certain event. When Event Save method is used, data are saved after END of each main task where the set bit condition occurred.





Note

After selecting Trigger Save, if the first trigger condition occurs and another trigger condition occurs while collecting data, the new trigger is ignored and the trigger reoccurrence flag value increases.

Note

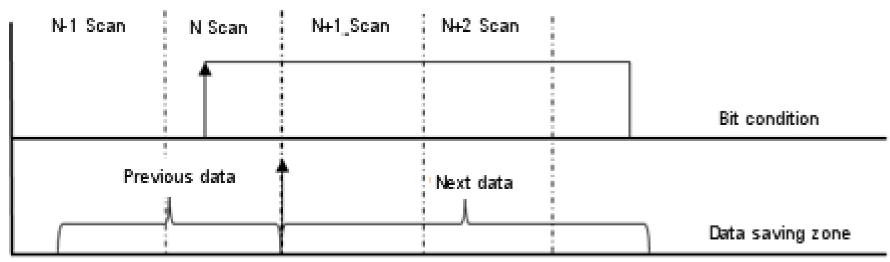
If a trigger occurs after collecting one less than the number of sample blocks prior to the trigger, only the number of blocks collected until then is stored, and the sample is started after the trigger. In this case, the total number of blocks collected can be less than the total number of trigger blocks

11.5.1 Trigger Condition

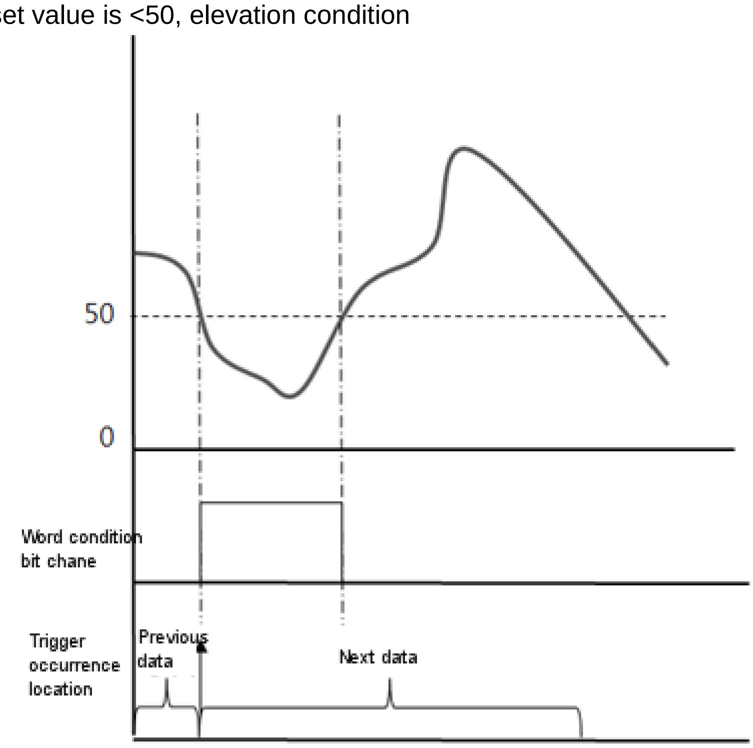
Trigger Save function runs under Single Condition, Multiple Condition. The setting item for single/multiple conditions are as follows. Multiple Condition runs by connecting Single Condition using AND, OR. Up to 4 Single Conditions can be set to form a condition. When the Trigger Condition occurs and data saving initiates, T character string is inserted into the first data string to indicate the trigger starting point.

(1) Single Condition
Single Condition runs under [BIT Condition], [WORD Condition].

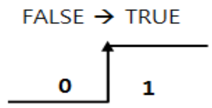
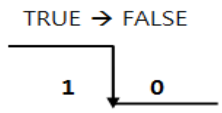
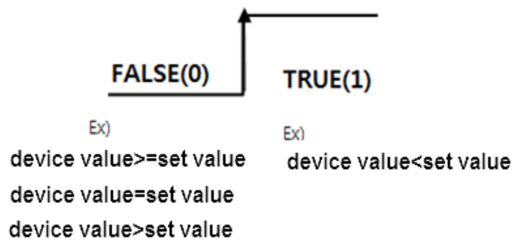
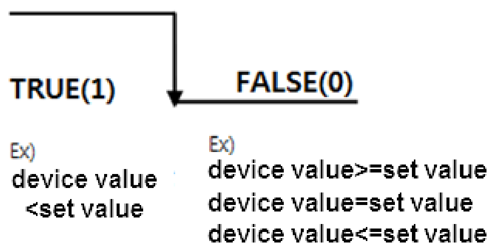
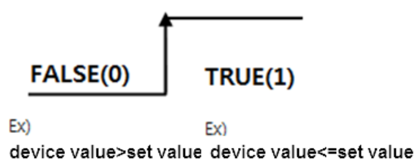
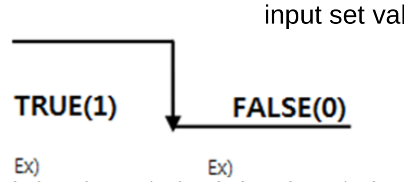
(a) BIT Condition
BIT condition checks the set device BIT value, and collects data by detecting trigger when the value is either [elevation] or [descent].

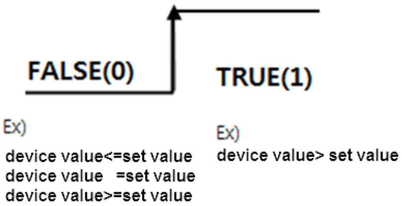
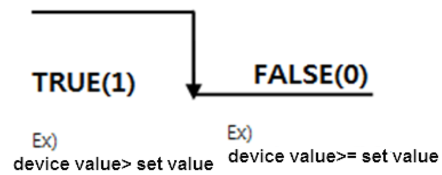
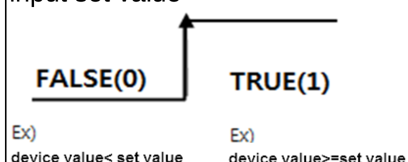
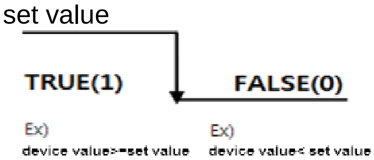
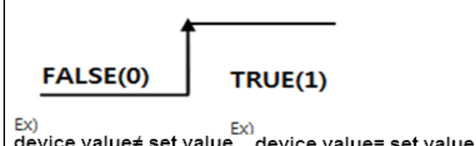
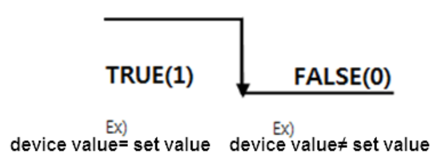


(b) WORD Condition
Word Condition compares the set device with the input value, and converts them into TRUE or FALSE. If the set device value satisfies the input condition, data are collected when the value is either [elevation] or [descent].
Ex) If set value is <50, elevation condition



(c) Condition Description

	Trigger Occurrence Condition	Device Set Condition	Operation	Note
Bit condition	Elevation		Saves data at elevation edge of set device bit value 	
	Descent		Saves data at descent edge of set device bit value 	
Word Condition	Elevation	Small	Saves data at the elevation edge of the relevant bit, when the set word device value is smaller than the input set value 	
	Descent		Saves data at the descent edge of the relevant bit, when the set word device value is smaller than the input set value 	
	Elevation	Small or Same	Saves data at the elevation edge of the relevant bit, when the set word device value is smaller than or the same as the input set value 	
	Descent		Saves data at the descent edge of the relevant bit, when the set word device value is smaller than or the same as the input set value 	

	Trigger Occurrence Condition	Device Set Condition	Operation	Note
Word Condition	Elevation	Large	Saves data at the descent edge of the relevant bit, when the set word device value is larger than the input set value  	
	Descent		Saves data at the descent edge of the relevant bit, when the set word device value is larger than the input set value  	
	Elevation	Large or Same	Saves data at the elevation edge of the relevant bit, when the set word device value is larger than or the same as the input set value  	
	Descent		Saves data at the descent edge of the relevant bit, when the set word device value is larger than or the same as the input set value  	
	Elevation	Same	Saves data at the elevation edge of the relevant bit, when the set word device value is the same as the input set value  	
	Descent		Saves data at the descent edge of the relevant bit, when the set word device value is the same as the input set value  	

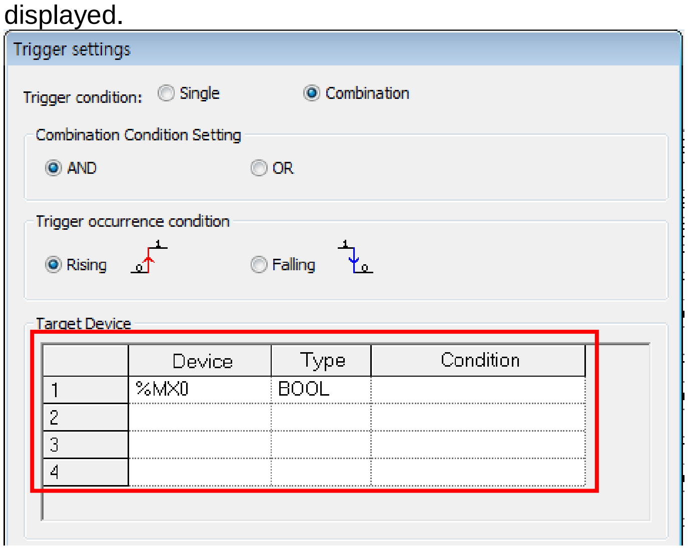
	Trigger Occurrence Condition	Device Set Condition	Operation	Note
Word Condition	Elevation	Different	Saves data at the elevation edge of the relevant bit, when the set word device value is different from the input set value FALSE(0) TRUE(1) Ex) device value= set value Ex) device value≠ set value	
	Descent		Saves data at the descent edge of the relevant bit, when the set word device value is different from the input set value TRUE(1) FALSE(0) Ex) device value≠ set value Ex) device value= set value	

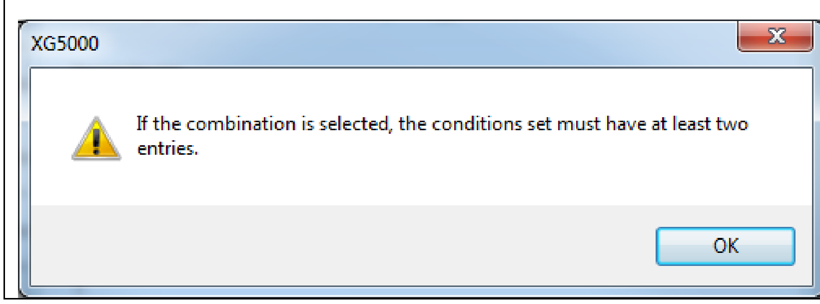
(2) Multiple Condition

Multiple condition refers to setting up to 4 single conditions and operating by performing the operations that fit the conditions. At least two Single Conditions should be set. Trigger Save begins when operation with the set single conditions satisfy the result. Multiple Condition runs under AND Calculation, OR Calculation.

Note

When less than 2 single conditions are set for trigger multiple condition, the following error message is displayed.

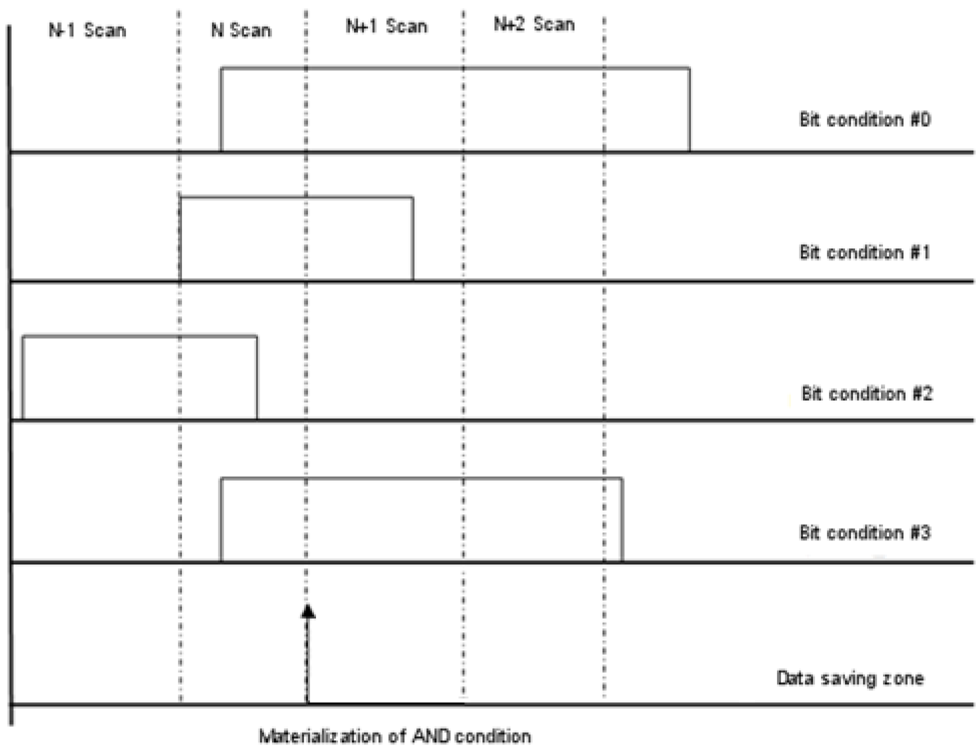




(a) AND Calculation
Trigger occurs when all relevant conditions are satisfied at a single main task.
The following figure shows an example of trigger save activated by trigger elevation and descent occurring within the same main task cycle.

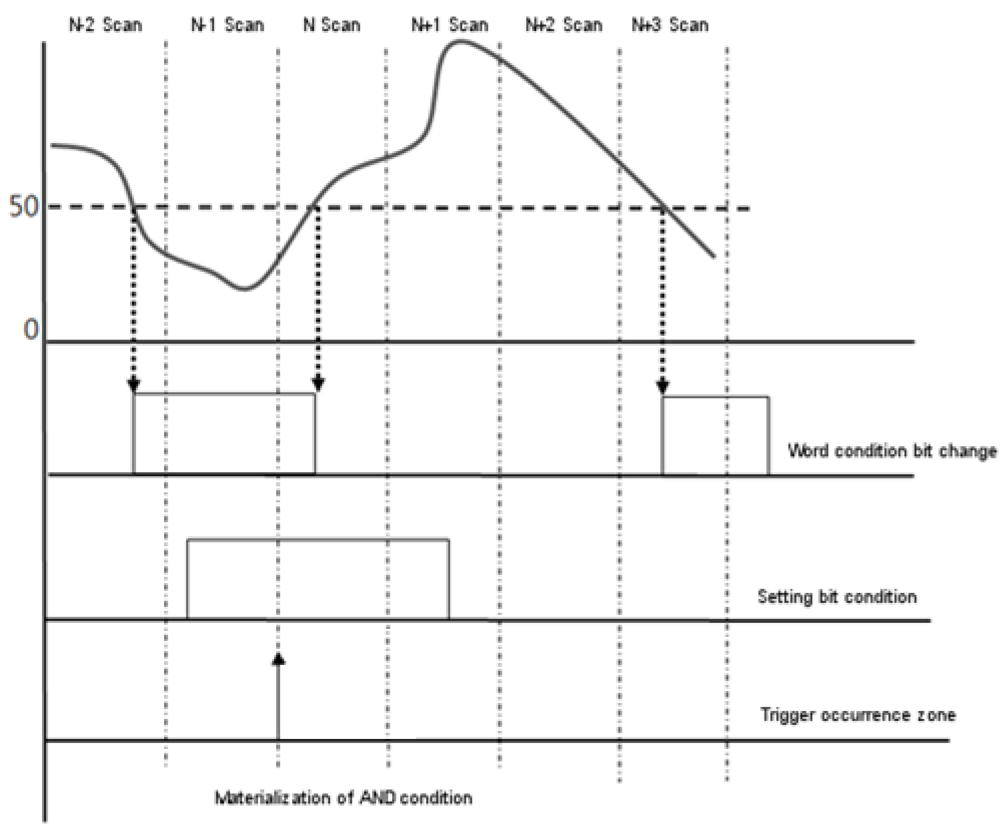
☞ When setting only with BIT condition

	Condition	Set Device	Trigger Occurrence Condition
Condition 0	BOOL	%MX1010	Elevation
Condition 1	BOOL	%IX1	
Condition 2	BOOL	%MX2010	
Condition 3	BOOL	%QX130	



☞ When setting with combination of BIT and WORD conditions

	Condition	Comparison Condition	Set Value	Set Device	Trigger Occurrence Condition
Condition 0	WORD	<	50	%MW10	Elevation
Condition 1	BOOL			%MX15	

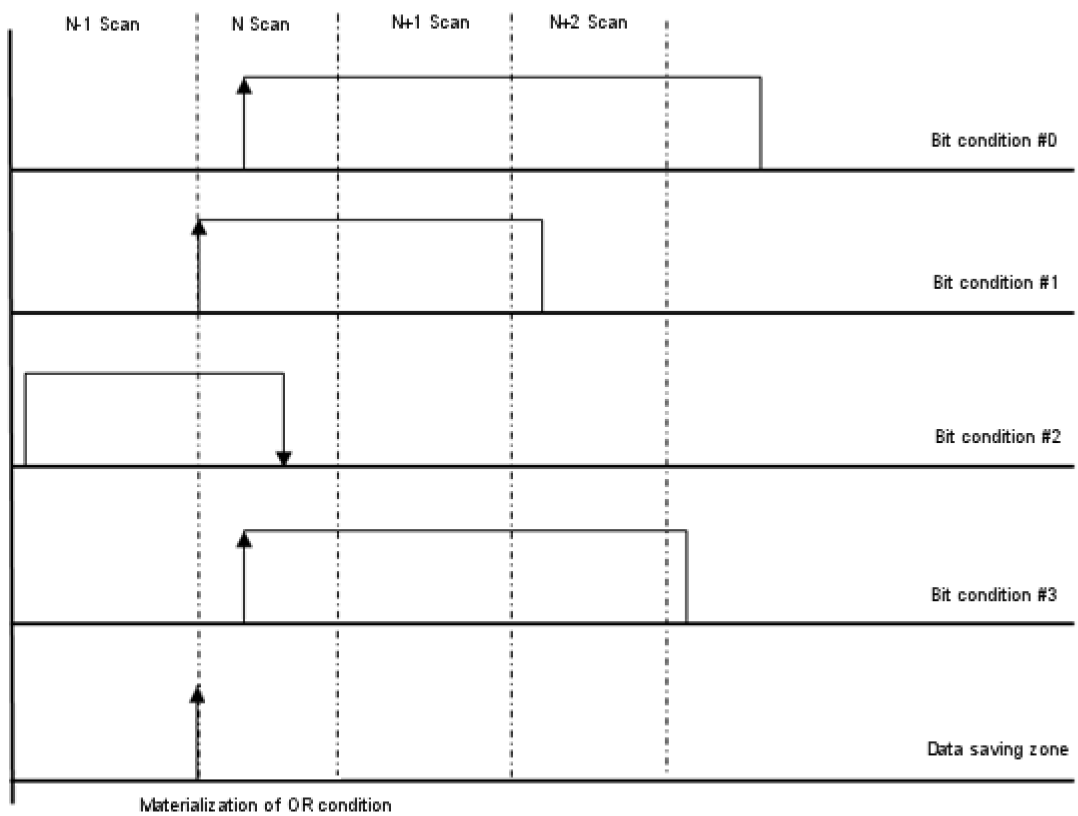


(b) OR Calculation

Trigger occurs when even one condition is satisfied at a single main task. After selecting Trigger Save, if the Trigger Condition is again satisfied before data saving is complete, the new trigger is ignored and the number of collisions flag value increases.

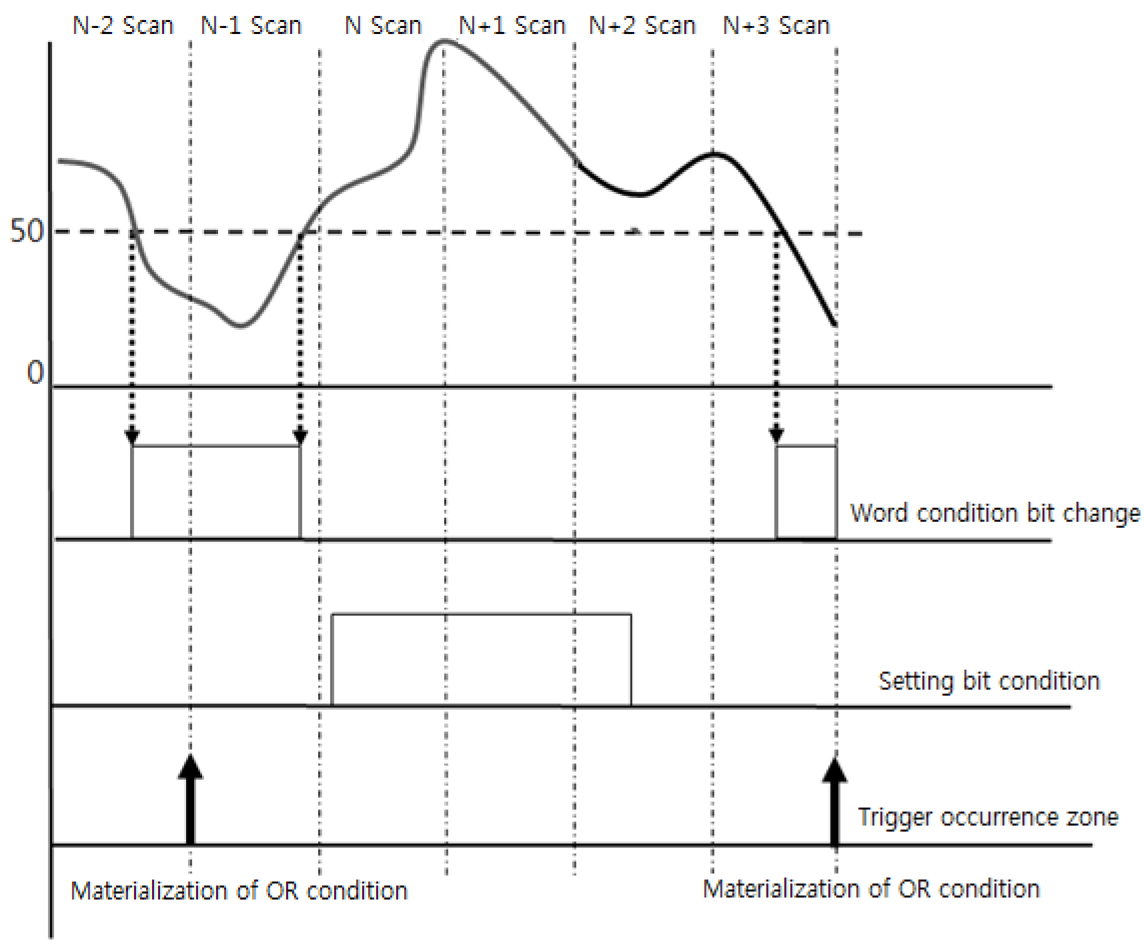
☞ When setting only with BOOL condition

	Condition	Set Device	Trigger Occurrence Condition
Condition 0	BOOL	%MX1010	Elevation
Condition 1	BOOL	%IX1	
Condition 2	BOOL	%MX2010	
Condition 3	BOOL	%QX130	



☞ When setting with combination of BIT and WORD conditions

	Condition	Comparison Condition	Set Value	Set Device	Trigger Occurrence Condition
Condition 0	WORD	<	50	%MW10	Elevation
Condition 1	BOOL			%MX15	



11.5.2 Trigger Sample Block Calculation

During Trigger Save, data collection progresses for each sample block. Sample block refers to the unit of collected data set by the datalog parameter, where sample refers to each data value. The number of trigger sample blocks and the total number of samples are calculated as follows.

No. of sample blocks = Trigger Buffer Space1) / {(No. of set data2) * size of set data3)) +(RTC data size4))}
No. of stored samples = sample block * No. of set data

(1) Trigger Buffer Space: 64KB ~ 2,048KB /Group
(2) No. of Set Data 64 (Maximum)
(3) Size of Set Data

Unit: Byte

Data Type	Data Size
BOOL	1
BYTE	1
WORD	2
DWORD	4
LWORD	8
INT	2
SINT	1
DINT	4
LINT	8
UINT	2
USINT	1
UDINT	4
ULINT	8
REAL	4
LREAL	8
STRING	32

Ex)
- Set the buffer memory: 128KB
- Number and type of setting data: 64 (BYTE)

Max. No. of sample blocks that can be set:
131,072 / { (64 * 1) + 29 } = 1409 sample blocks

(4) RTC size of data: 29 (fixed value)

11.5.3 Trigger Sample Calculation

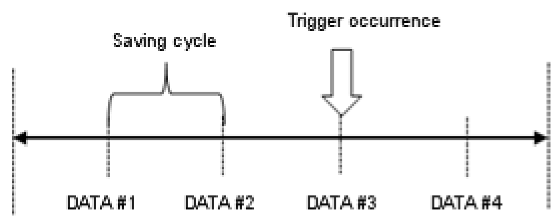
The item that can be set at the parameter is the total number of trigger sample blocks and the number of sample blocks before trigger condition. The number of sample blocks after trigger is determined by the two input values

Total Number of Trigger Samples (Setting Available)	= Number of Samples before Trigger Condition (Setting Available)	+	Number of Samples after Trigger Condition (Setting Available)

11.5.4 Trigger Sample Save Cycle

When Trigger Condition occurs, data collected are saved at the sampling interval set by the parameter.
The saving interval is as follows.

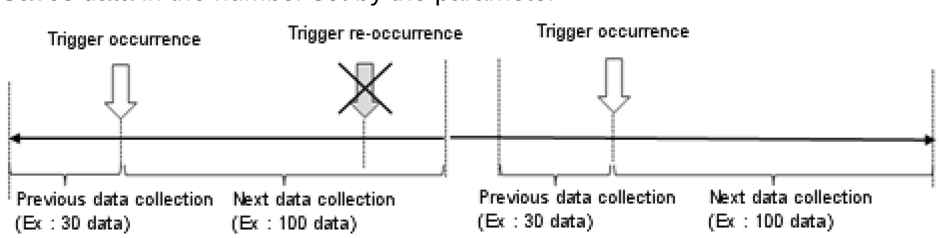
→ **Main task interval, 20ms, 50ms, 100 ms, 200 ms, 500 ms, 1000 ms, 2000 ms**



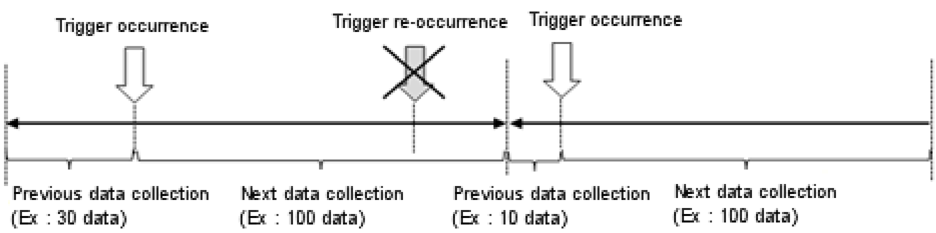
Caution
After selecting Trigger Save, if the Trigger Condition is again satisfied before data saving is complete, the new trigger is ignored and the trigger collisions number flag value increases. Trigger Condition is checked after saving the set number of trigger sample blocks, and then the data are saved.

11.5.5 Trigger Sample Save Section

(1) If Trigger occurs after the number of previous data set by the parameter
→ Saves data in the number set by the parameter



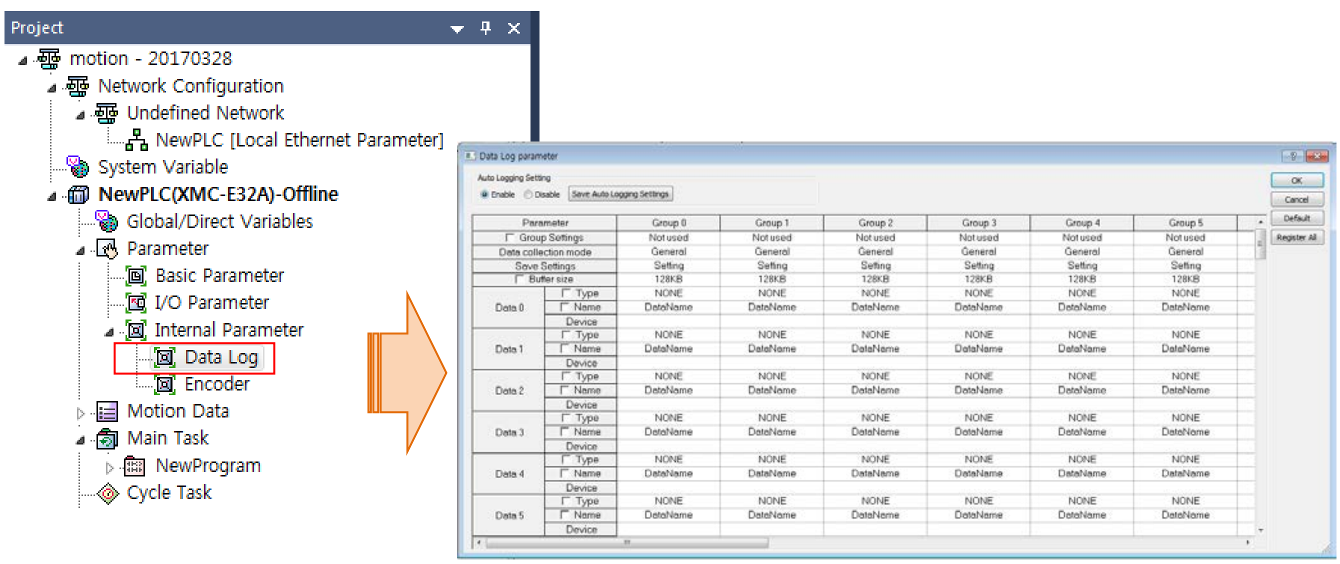
(2) If Trigger occurs before the number of previous data set by the parameter
→ Saves data in the number of transfer data collected, and then collects subsequent data
(Saves less number of data than the number set by the parameter)



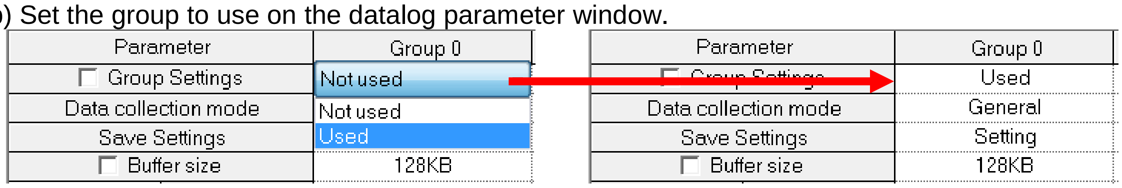
11.5.6 Setting Method

(1) Single BIT Condition

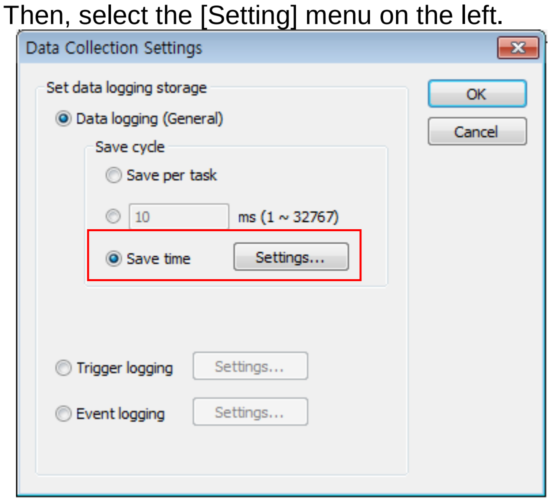
(a) Choose XG5000 – [Project Window] - [internal parameter] - [datalog]
This activates the datalog parameter setting window.



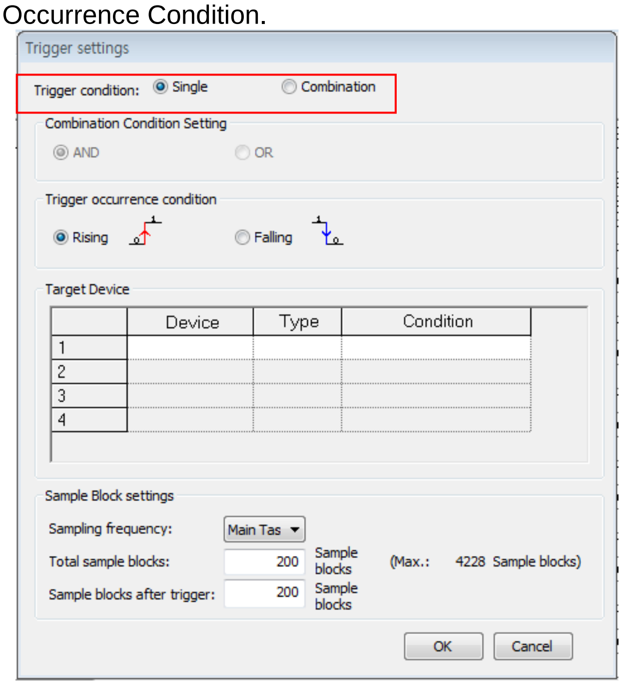
(b) Set the group to use on the datalog parameter window.



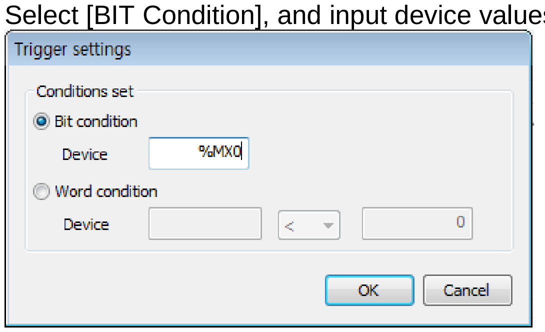
(c) Select [Trigger Logging] at [Data Collection Method] to activate [Setting] menu on the left.
Then, select the [Setting] menu on the left.



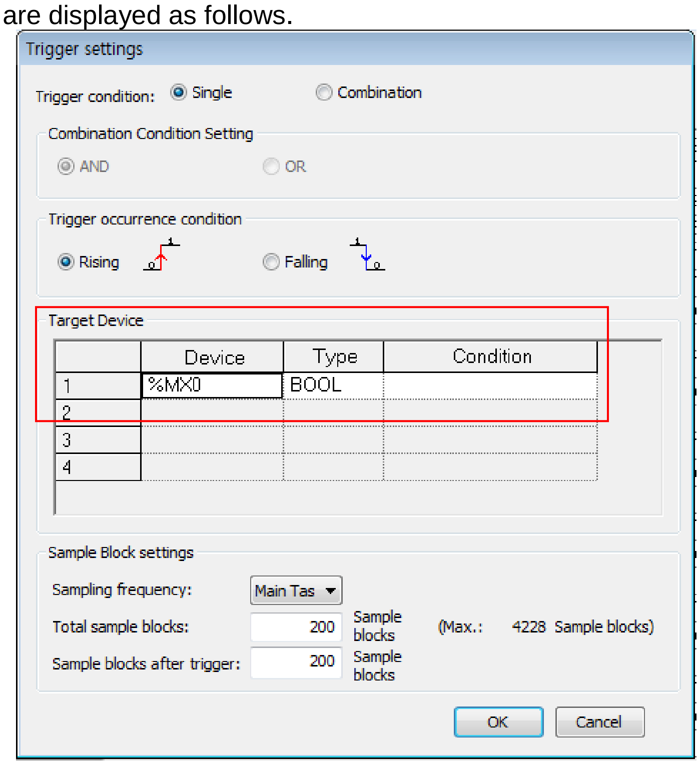
(d) Upon selection, the following window is activated for trigger setting.
Select [Single Condition] as the Trigger Condition. Select either [Elevation] or [Descent] as the Trigger Occurrence Condition.



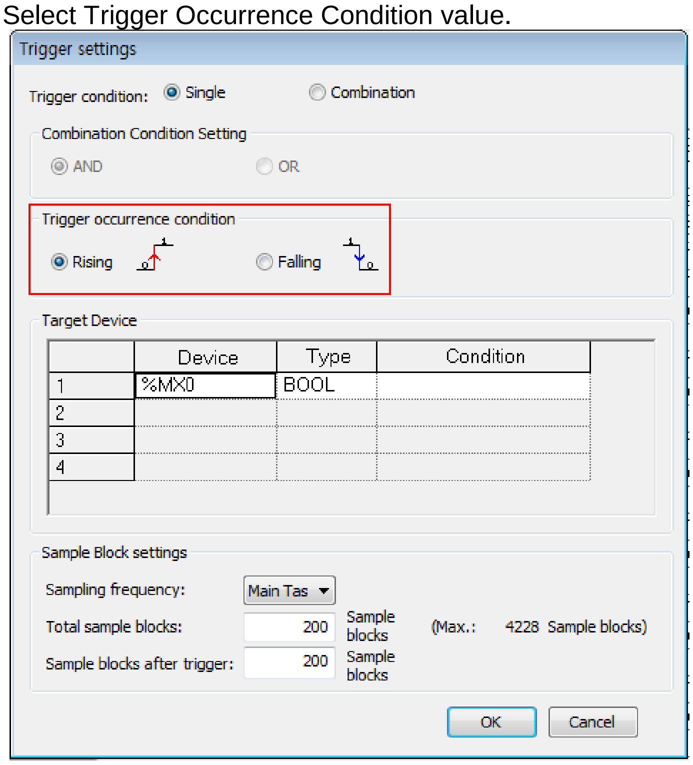
(e) Select the condition setting menu to activate the following setting window.
Select [BIT Condition], and input device values into the device window in BIT types.



When setting is complete, the window closes and the conditions initially set at the Trigger Setting Condition menu are displayed as follows.

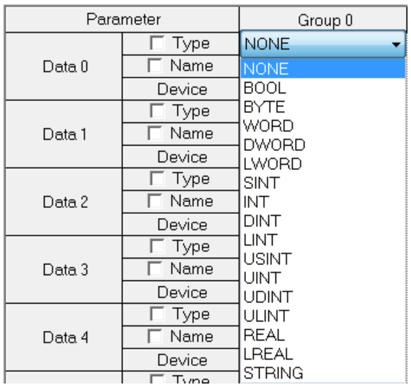


(f) Select Trigger Occurrence Condition value.



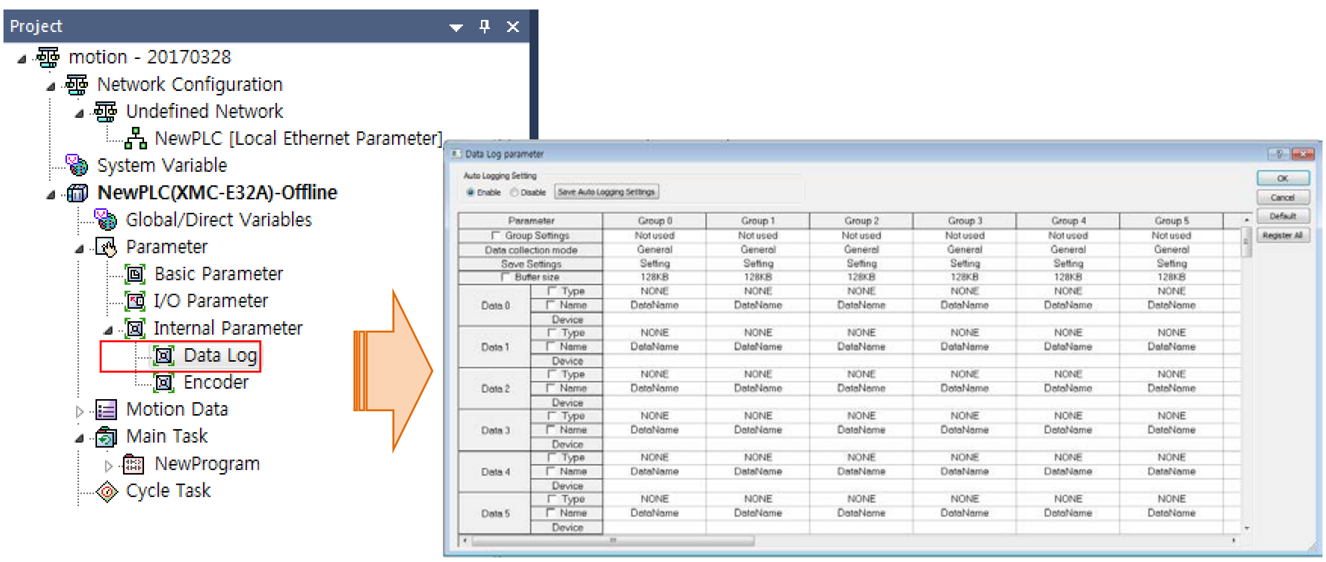
(g) Input sampling interval, total number of samples and number of samples after trigger, then finish Trigger setting. See [11.5.2 Trigger Sample Block Calculation] for operation of number of sample blocks.

(h) Device values set at the Datalog Basic Setting window are collected, and saved into the SD memory after type conversion.

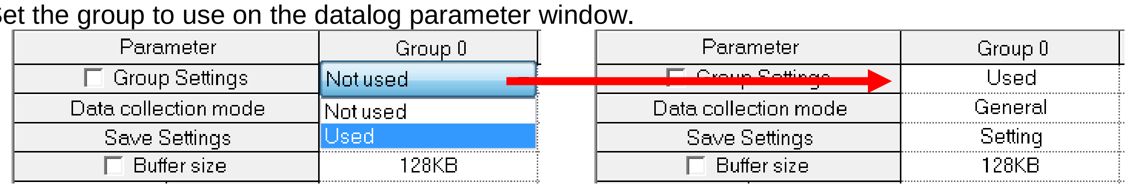


(2) Single WORD Condition

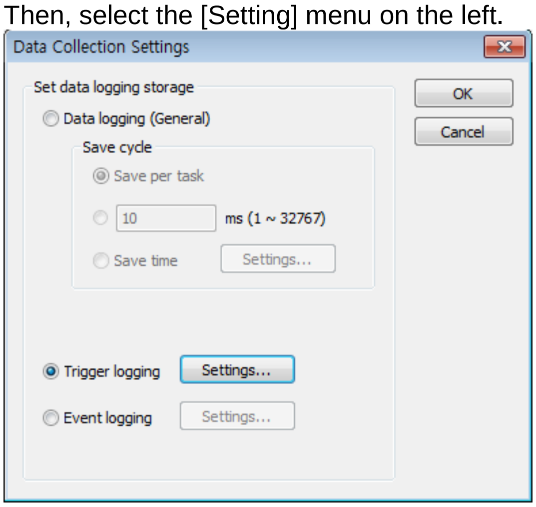
(a) Choose XG5000 – [Project Window] - [internal parameter] - [datalog]
This activates the datalog parameter setting window.



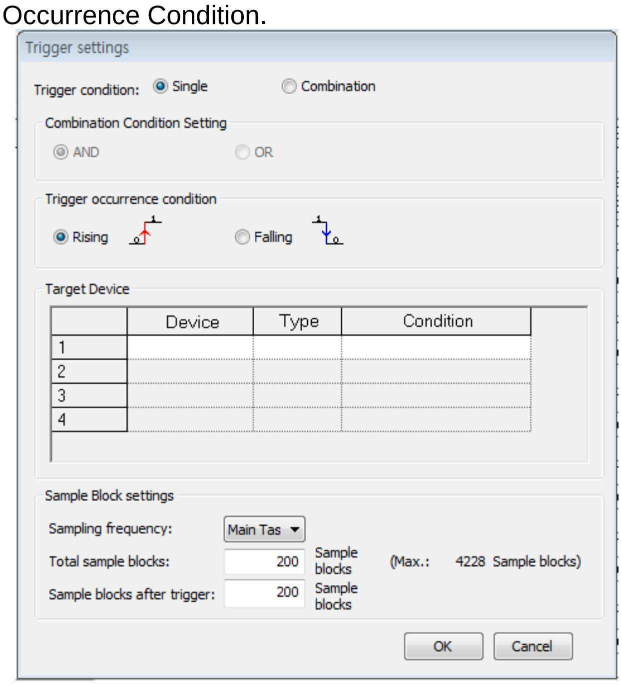
(b) Set the group to use on the datalog parameter window.



(c) Select [Trigger Logging] at [Data Collection Method] to activate [Setting] menu on the left. Then, select the [Setting] menu on the left.

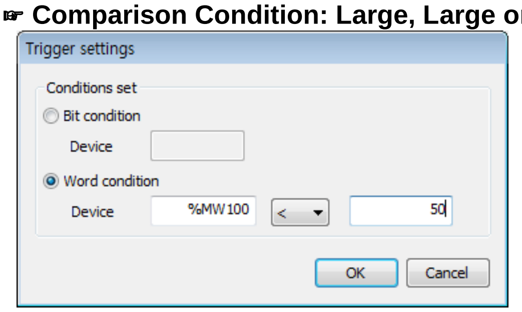


(d) Upon selection, the following window is activated for trigger setting. Select [Single Condition] as the Trigger Condition. Select either [Elevation] or [Descent] as the Trigger Occurrence Condition.

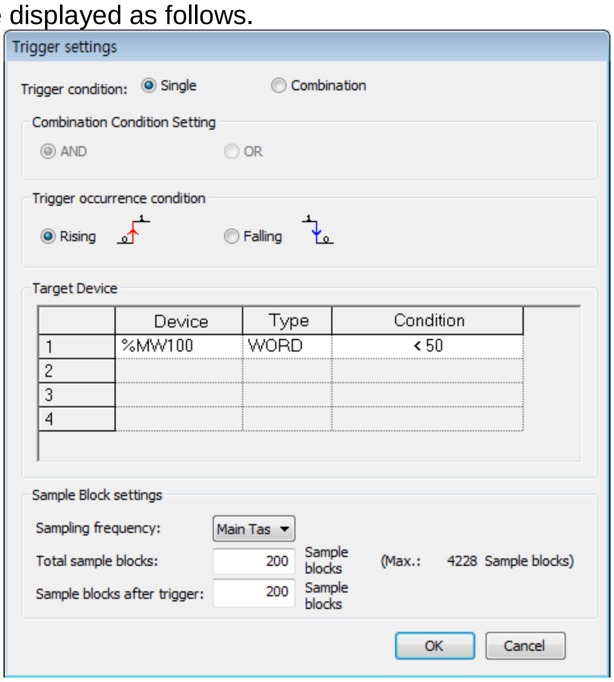


(e) Select the condition setting menu to activate the following setting window. Select [Word Condition], and input device values into the device window in BIT types, and input comparison condition and comparison values

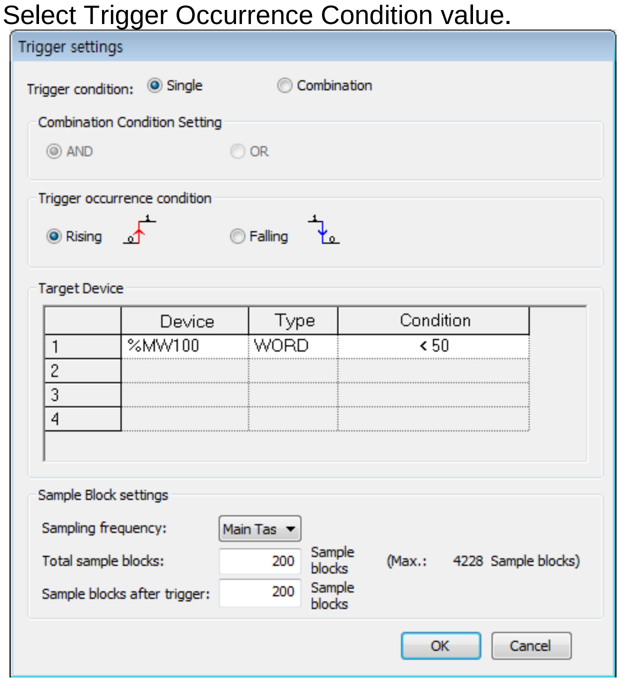
☞ **Comparison Condition: Large, Large or Same, Same, Small, Small or Same, Not Same.**



When setting is complete, the window closes and the conditions initially set at the Trigger Setting Condition menu are displayed as follows.



(f) Select Trigger Occurrence Condition value.



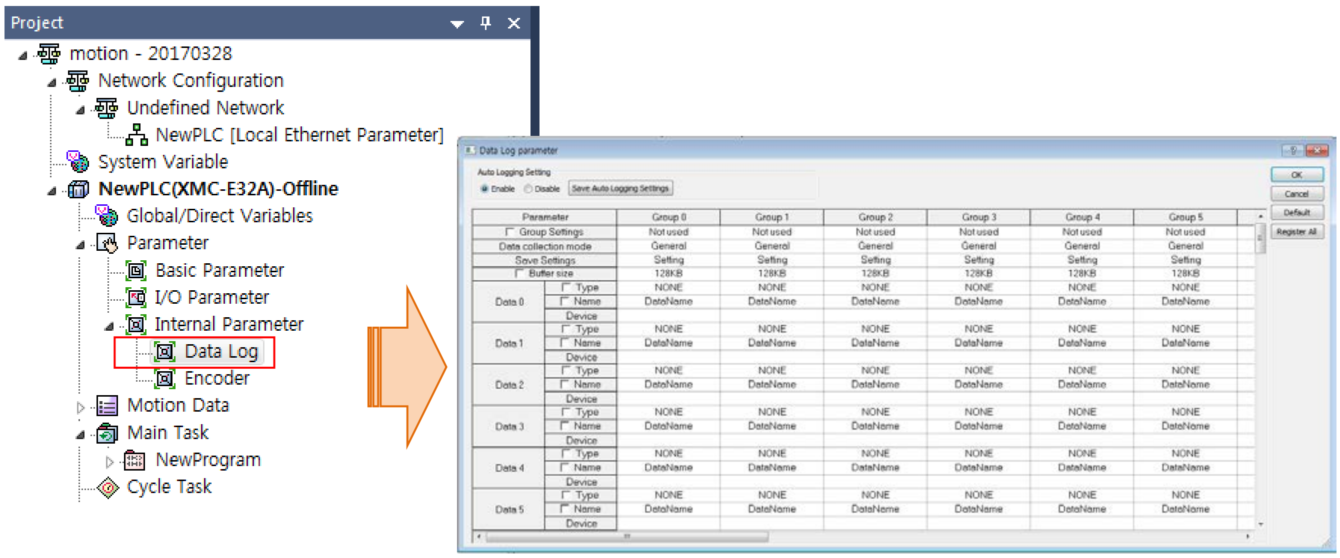
(g) Input sampling interval, total number of samples and number of samples after trigger, then finish Trigger setting. See [11.5.2 Trigger Sample Block Calculation] for operation of number of sample blocks.
(h) Device values set at the Datalog Basic Setting window are collected, and saved after type conversion.

Caution

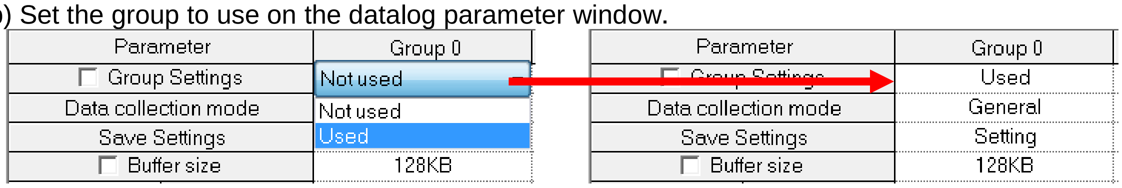
When inputting single, word condition set values, set device type as [BIT] and [WORD], respectively.

(3) Multiple AND Condition

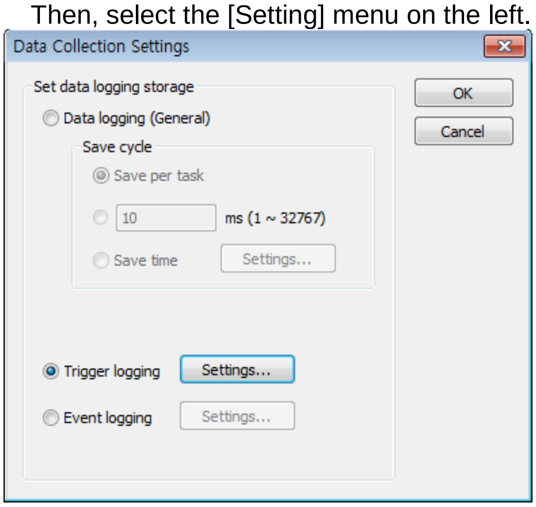
(a) Choose XG5000 – [Project Window] - [internal parameter] - [datalog]
This activates the datalog parameter setting window.



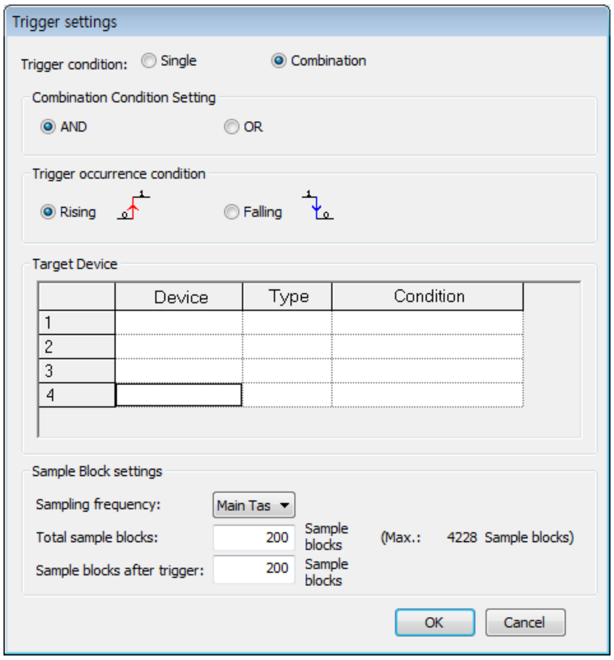
(b) Set the group to use on the datalog parameter window.



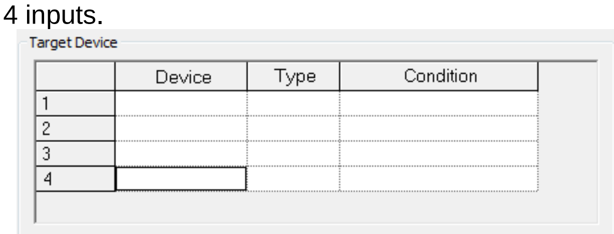
(c) Select [Trigger Logging] at [Data Collection Method] to activate [Setting] menu on the left.
Then, select the [Setting] menu on the left.



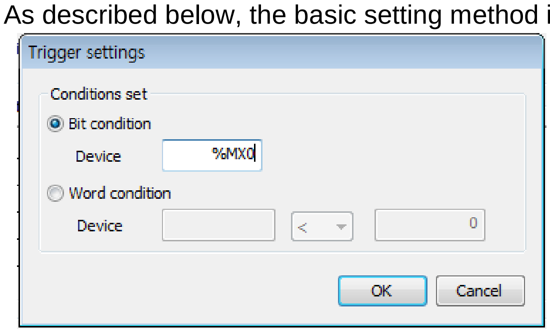
(d) Upon selection, the following window is activated for trigger setting.
Select [Multiple Condition] as Trigger Condition, Select either [Elevation] or [Descent] as the Trigger Occurrence Condition.

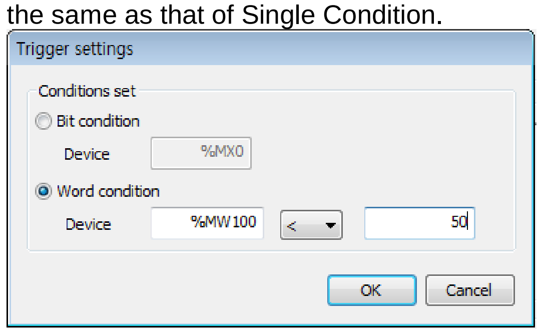


(e) Select [Trigger Condition] and [Multiple Condition] to activate the condition setting window which allows for up to 4 inputs.

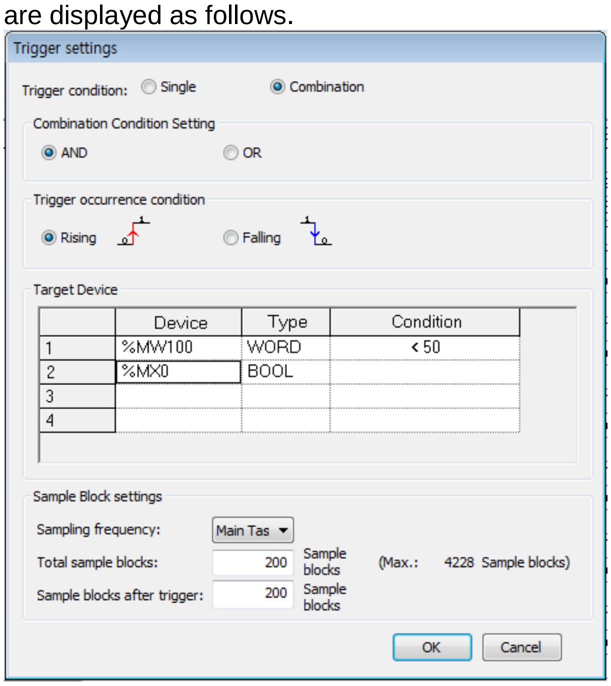


(f) Select each condition setting menu one by one, inputting specific set values.
[Multiple Condition] activates Trigger Condition by combining [Single Conditions] through operation to save data. As described below, the basic setting method is the same as that of Single Condition.

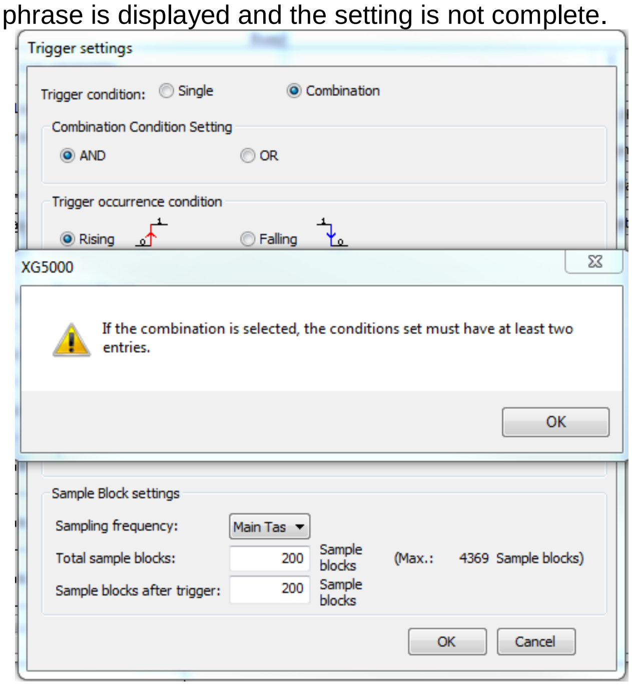




When setting is complete, the window closes and the conditions initially set at the Trigger Setting Condition menu are displayed as follows.

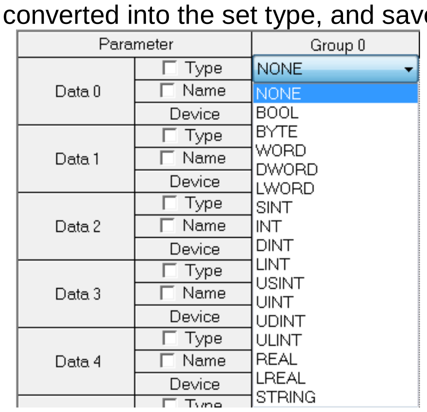


If only one [Condition Setting] is input after selecting Calculation Condition before finishing the setting, the following phrase is displayed and the setting is not complete.



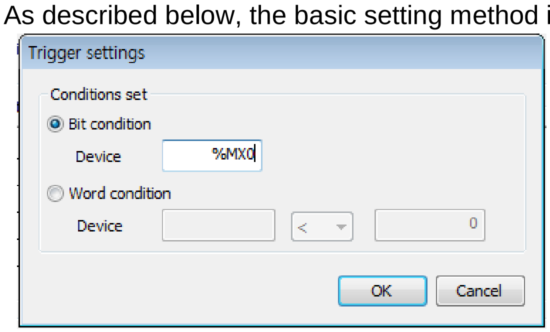
(g) Input sampling interval, total number of samples and number of samples after trigger, then finish Trigger setting.

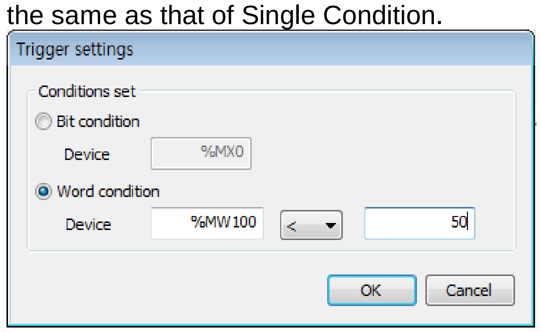
(h) Device values set at the Datalog Basic Setting window are collected when the Trigger Condition occurs, converted into the set type, and saved into the SD memory.ion.



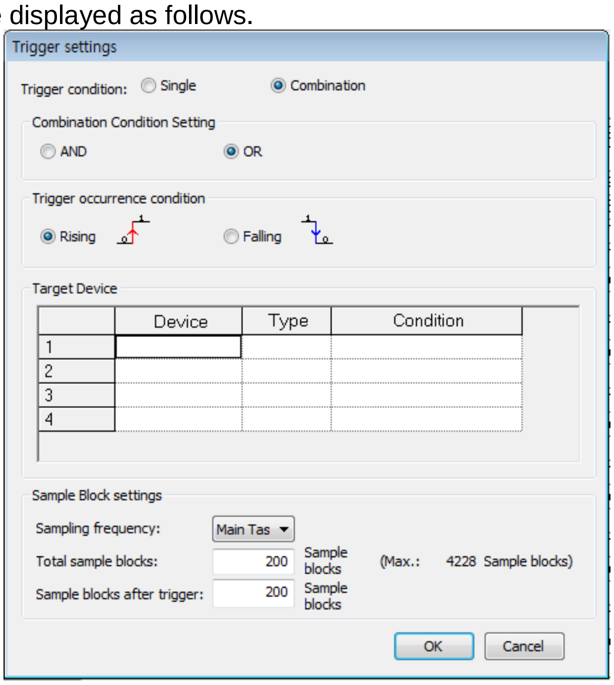
(4) Multiple OR Condition
[Trigger Setting] is identical to the [Multiple AND Calculation] above.

(a) Select each condition setting menu one by one, inputting specific set values.
[Multiple Condition] activates Trigger Condition by combining [Single Conditions] through operation to save data. As described below, the basic setting method is the same as that of Single Condition.

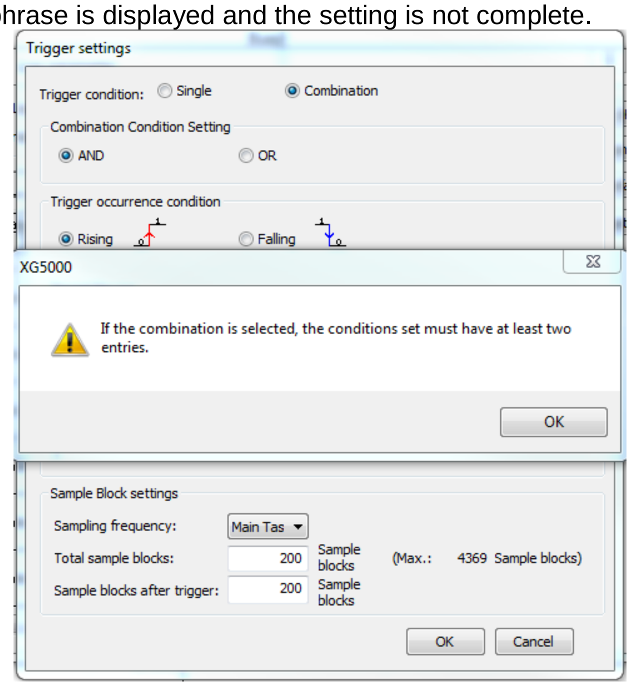




When setting is complete, the window closes and the conditions initially set at the Trigger Setting Condition menu are displayed as follows.

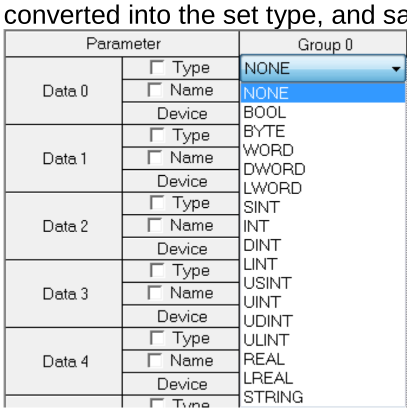


If only one [Condition Setting] is input after selecting Calculation Condition before finishing the setting, the following phrase is displayed and the setting is not complete.



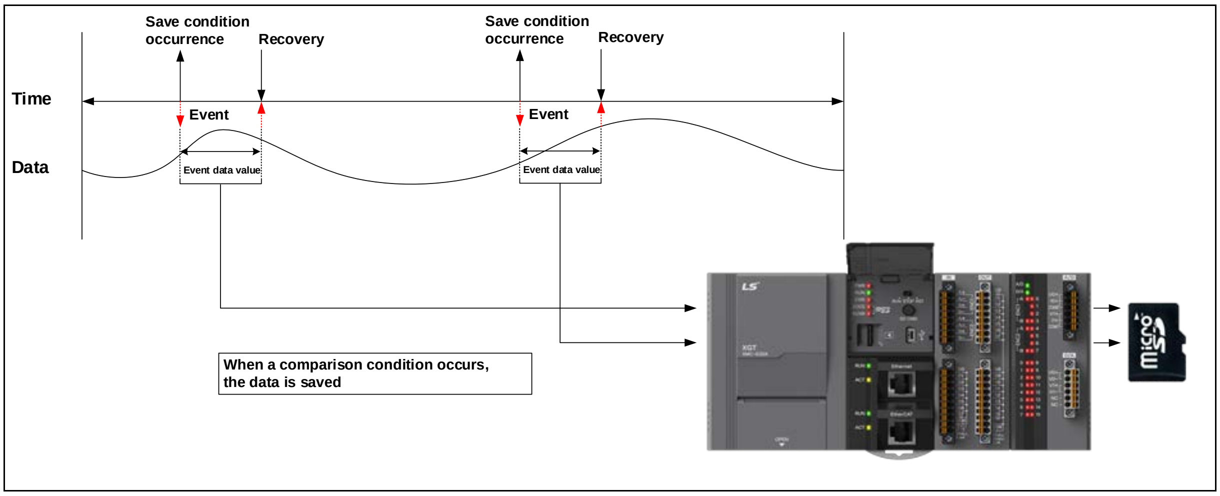
(b) Input sampling interval, total number of samples and number of samples after trigger, then finish Trigger setting.

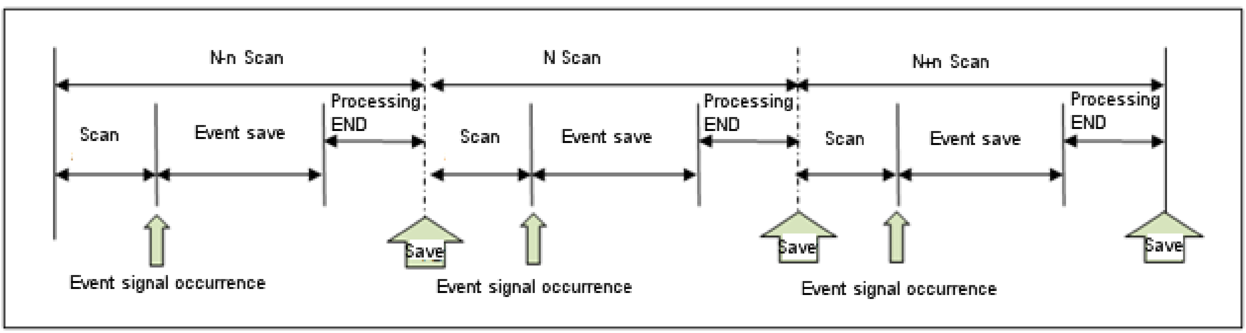
(c) Device values set at the Datalog Basic Setting window are collected when the Trigger Condition occurs, converted into the set type, and saved into the SD memory.



11.6 Event Save

Event Save refers to monitoring the device value collected, and saving the present data when a certain event condition is satisfied. This method is useful for analyzing fluctuation of event values and timing by saving data from the event occurrence to the event termination. Event Save refers to saving data when the event occurs, until the conditions are not satisfied.





Note

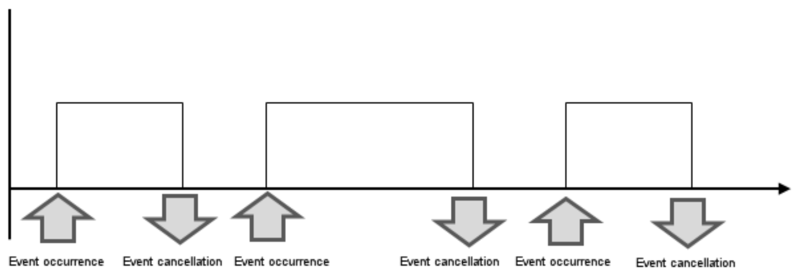
After selecting Trigger Save, if the first trigger condition occurs and another trigger condition occurs while collecting data, the new trigger is ignored.

11.6.1 Event Condition

Event Save function runs under Single Condition, Multiple Condition. The setting item for single/operation conditions are as follows. Multiple Condition runs by connecting Single Condition using operation. Up to 4 Single Conditions can be set to form a condition. When the Trigger Condition occurs and data saving initiates, E character string is inserted into the first data string to indicate the event occurrence.

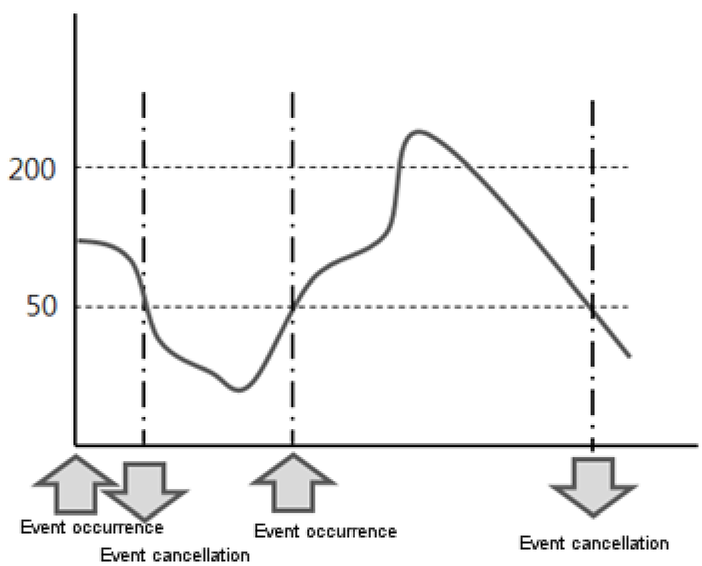
(1) Single Condition
Single Condition runs under BIT Condition, WORD Condition.

(a) BIT Condition
BIT condition checks the set device BIT value, and collects data by detecting trigger when the value is either [elevation], [descent], [transfer], [ON], or [OFF].



(b) WORD Condition
Word Condition compares the set device with the input value, and converts them into TRUE or FALSE. If the set device value satisfies the input condition, saves data when the value is [elevation], [descent], [transfer], [ON], or [OFF].

Ex) If set value is >50, elevation condition



(c) Release Value Setting

Among Event Save functions, release value setting can be done only in WORD Condition. It affects data save interval and frequency. Once the release value is set, the condition after event occurrence saves data until the release value is satisfied.

	Use Release Value Setting	Do Not Use Release Value Setting
%MW0 > 100	☞ Release Value Setting 50 Saves data until the setting value after event occurrence is 50	Saves data until the condition is met after event occurrence
%MW0 >= 100		
%MW0 == 100	Release Value Cannot be Set	
%MW0 < 100	☞ Release Value: 120 Saves data until the setting value after event occurrence is 120	
%MW0 <= 100		
%MW0 <> 100	Release Value Cannot be Set	

Note

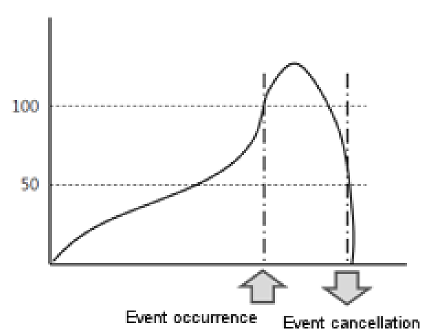
Release value can be set as follows. If the following is not complied with, an error window will appear and data input will not work. Check it when setting the parameter.

☞ Release value many not overlap with the range of set values.

Condition	Range of Release Value
Large	Set Value >= Release Value
Large or Same	Set Value > Release Value
small	Set Value <= Release Value
Small or Same	Set Value <= Release Value
Same	Setting Available
Not Same	

Example 1) In the word condition, if the value is set to %MW0>100, and the cancelation value is set to 50.

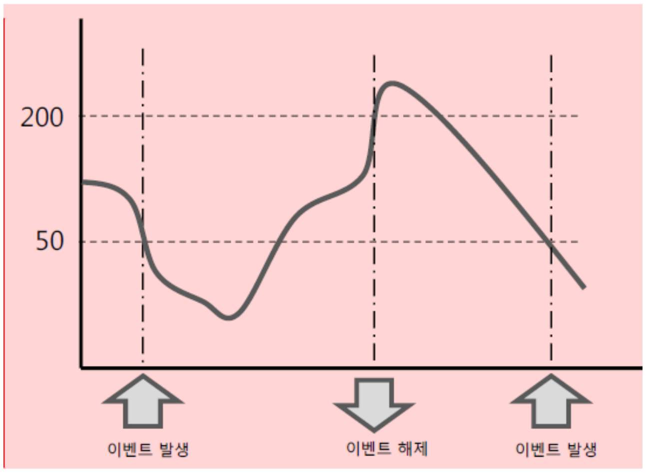
☞ If %MW0 exceeds 100, an event occurs, and the data is saved. However, since the cancelation value is set to 50, the data storage is performed until it reaches 50.



☞ When the condition is met, the data is saved at the point of time set in the even occurrence condition. [Rise], [Fall], [Transition] conditions store 1 block of data by operation, and [ON],[OFF] conditions store data until the conditions are not met by the level operation.

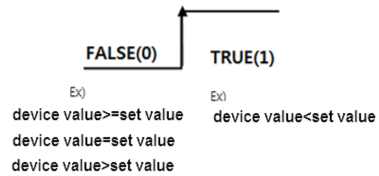
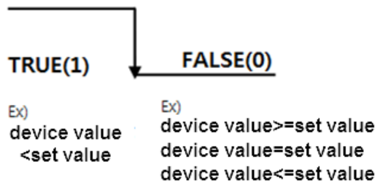
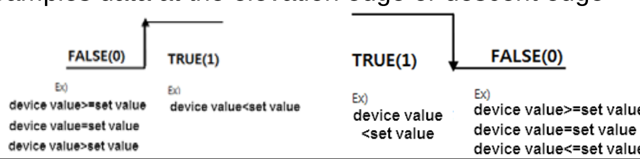
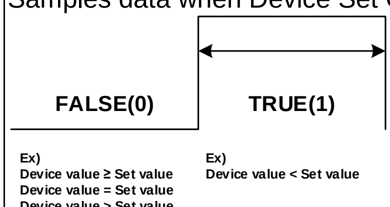
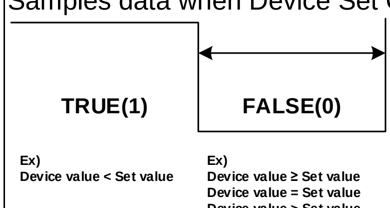
Example 2) In the word condition, if the value is set to %MW0<50, and the cancelation value is set to 200

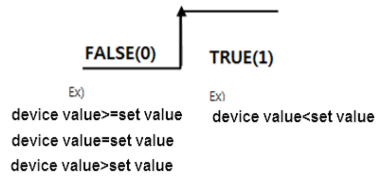
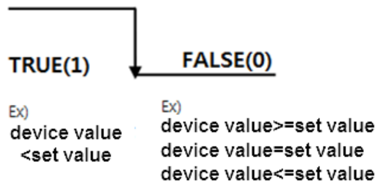
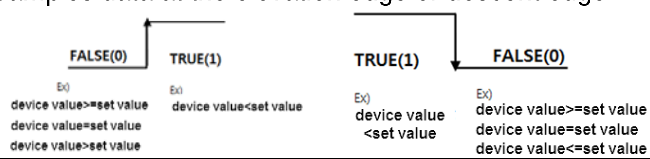
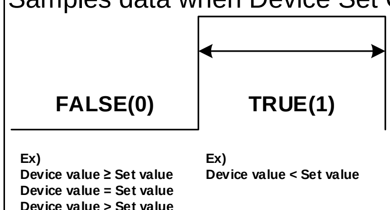
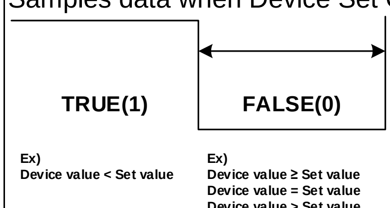
☞ If %MW0 is less than 50, an event occurs, and the data is saved. However, since the cancelation value is set to 200, the data is saved until it reaches 200.

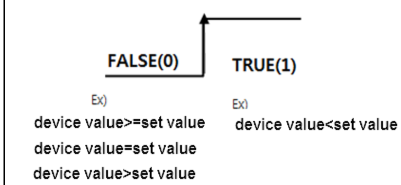
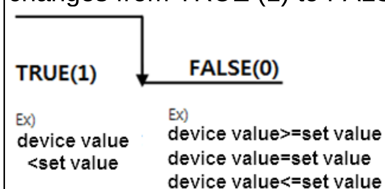
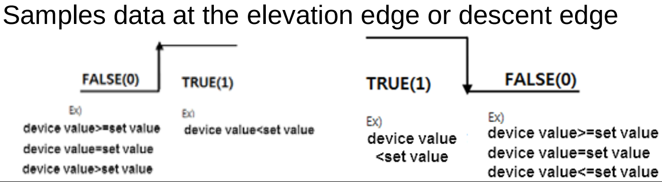
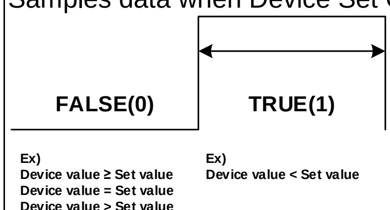
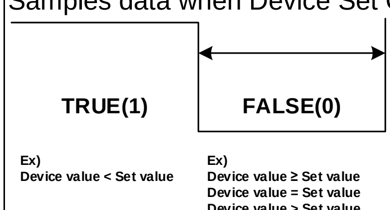


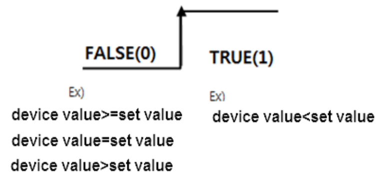
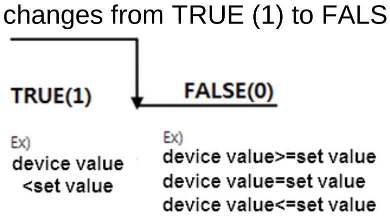
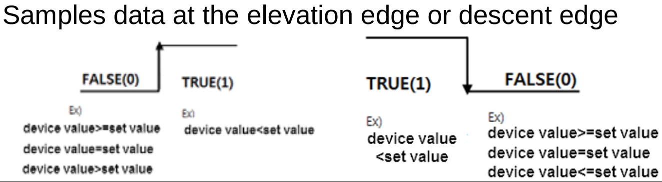
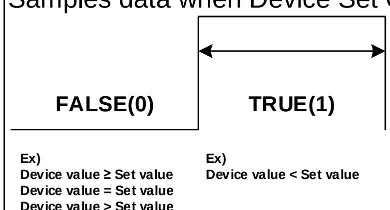
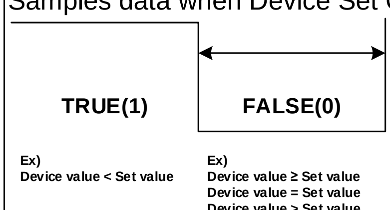
(d) Condition Description

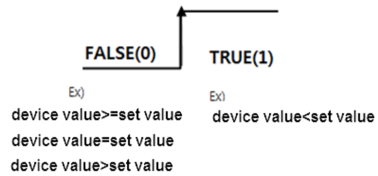
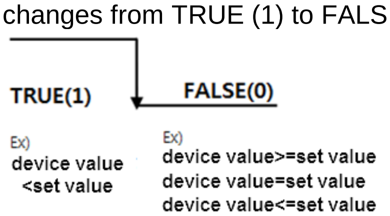
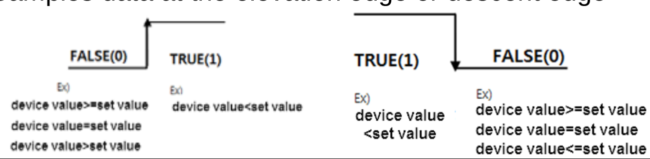
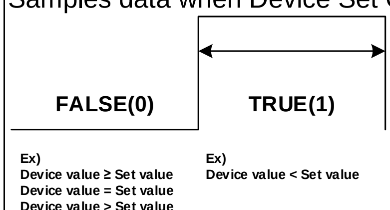
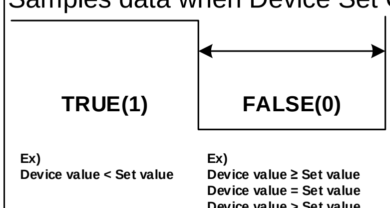
	Trigger Occurrence Condition	Device Set Condition	Operation	Release Value Setting
Bit Condition	Elevation		Saves data at elevation edge of set device bit value FALSE → TRUE 0 1	
	Descent		Saves data at descent edge of set device bit value TRUE → FALSE 1 0	
	Transfer		Saves data when set device bit value is transferred TRUE → FALSE 1 0 or FALSE → TRUE 0 1	
	ON		Saves data when set device bit value is ON ON 0 1	
	OFF		Saves data when set device bit value is OFF OFF 1 0	

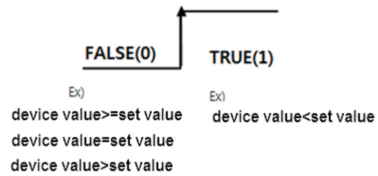
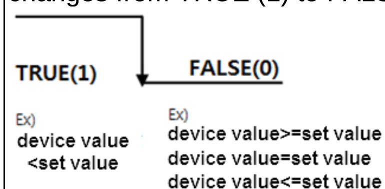
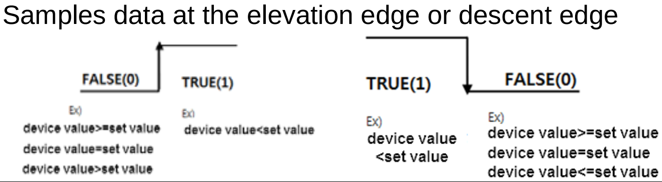
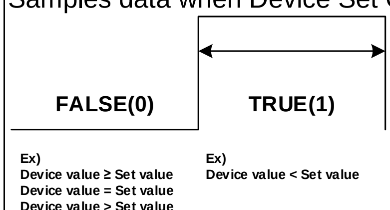
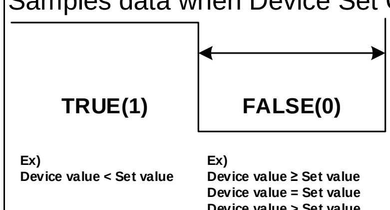
	Trigger Occurrence Condition	Device Set Condition	Operation	Release Value Setting
Word Condition	Elevation	Small	Samples data at the elevation edge when Device Set Condition changes from FALSE (0) to TRUE (1).  	Setting Available
	Descent		Samples data at the descent edge when Device Set Condition changes from TRUE (1) to FALSE (0).  	
	Transfer		Samples data at the elevation edge or descent edge  	
	ON		Samples data when Device Set Condition is TRUE (1)  	
	OFF		Samples data when Device Set Condition is FALSE (0)  	

	Trigger Occurrence Condition	Device Set Condition	Operation	Release Value Setting
Word Condition	Elevation	Small or Same	Samples data at the elevation edge when Device Set Condition changes from FALSE (0) to TRUE (1).  	Setting Available
	Descent		Samples data at the descent edge when Device Set Condition changes from TRUE (1) to FALSE (0).  	
	Transfer		Samples data at the elevation edge or descent edge  	
	ON		Samples data when Device Set Condition is TRUE (1)  	
	OFF		Samples data when Device Set Condition is FALSE (0)  	

<table>
<tr><th></th><th>Trigger Occurrence Condition</th><th>Device Set Condition</th><th>Operation</th><th>Release Value Setting</th></tr>
<tr><td rowspan="5">Word Condition</td><td>Elevation</td><td rowspan="5">Large</td><td>Samples data at the elevation edge when Device Set Condition changes from FALSE (0) to TRUE (1).

</td><td>Setting Available</td></tr>
<tr><td>Descent</td><td>Samples data at the descent edge when Device Set Condition changes from TRUE (1) to FALSE (0).

</td><td></td></tr>
<tr><td>Transfer</td><td>Samples data at the elevation edge or descent edge

</td><td></td></tr>
<tr><td>ON</td><td>Samples data when Device Set Condition is TRUE (1)

</td><td></td></tr>
<tr><td>OFF</td><td>Samples data when Device Set Condition is FALSE (0)

</td><td></td></tr>
</table>

<table>
<tr><th></th><th>Trigger Occurrence Condition</th><th>Device Set Condition</th><th>Operation</th><th>Release Value Setting</th></tr>
<tr><td rowspan="5">Word Condition</td><td>Elevation</td><td rowspan="5">Large or Same</td><td>Samples data at the elevation edge when Device Set Condition changes from FALSE (0) to TRUE (1).

</td><td>Setting Available</td></tr>
<tr><td>Descent</td><td>Samples data at the descent edge when Device Set Condition changes from TRUE (1) to FALSE (0).

</td><td></td></tr>
<tr><td>Transfer</td><td>Samples data at the elevation edge or descent edge

</td><td></td></tr>
<tr><td>ON</td><td>Samples data when Device Set Condition is TRUE (1)

</td><td></td></tr>
<tr><td>OFF</td><td>Samples data when Device Set Condition is FALSE (0)

</td><td></td></tr>
</table>

	Trigger Occurrence Condition	Device Set Condition	Operation	Release Value Setting
Word Condition	Elevation	Same	Samples data at the elevation edge when Device Set Condition changes from FALSE (0) to TRUE (1).  	Setting Available
	Descent		Samples data at the descent edge when Device Set Condition changes from TRUE (1) to FALSE (0).  	
	Transfer		Samples data at the elevation edge or descent edge  	
	ON		Samples data when Device Set Condition is TRUE (1)  	
	OFF		Samples data when Device Set Condition is FALSE (0)  	

	Trigger Occurrence Condition	Device Set Condition	Operation	Release Value Setting
Word Condition	Elevation	Not Same	Samples data at the elevation edge when Device Set Condition changes from FALSE (0) to TRUE (1). 	Setting Available
	Descent		Samples data at the descent edge when Device Set Condition changes from TRUE (1) to FALSE (0). 	
	Transfer		Samples data at the elevation edge or descent edge 	
	ON		Samples data when Device Set Condition is TRUE (1) 	
	OFF		Samples data when Device Set Condition is FALSE (0) 	

(2) Multiple Condition

Multiple Condition refers to setting up to 4 single conditions and operating by performing the runs that fit the conditions

Event condition occurs when operation with the set condition satisfies the result

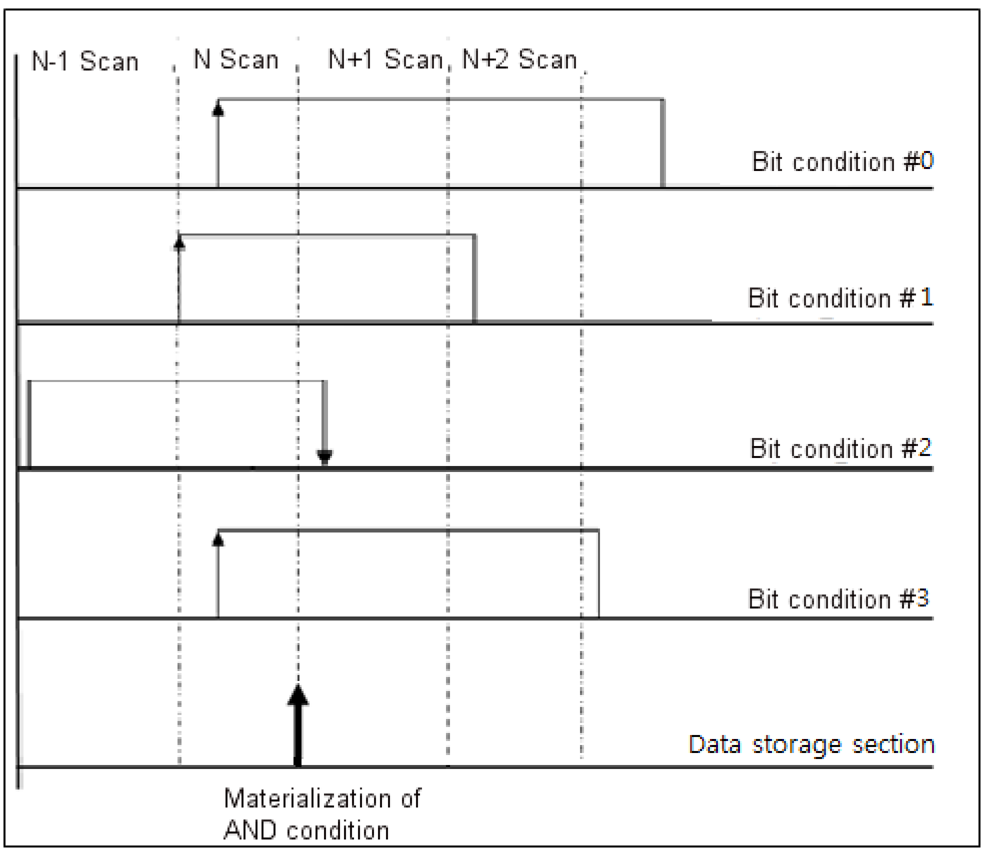
Setting	Operation	Note
AND Condition	Performs AND run with the set conditions, and saves data when the result is 1.	
OR Condition	Performs OR run with the set conditions, and saves data when the result is 1.	

(a) AND Calculation

Event occurs when all relevant conditions are satisfied at a single scan. The following is an example of activating Event Save.

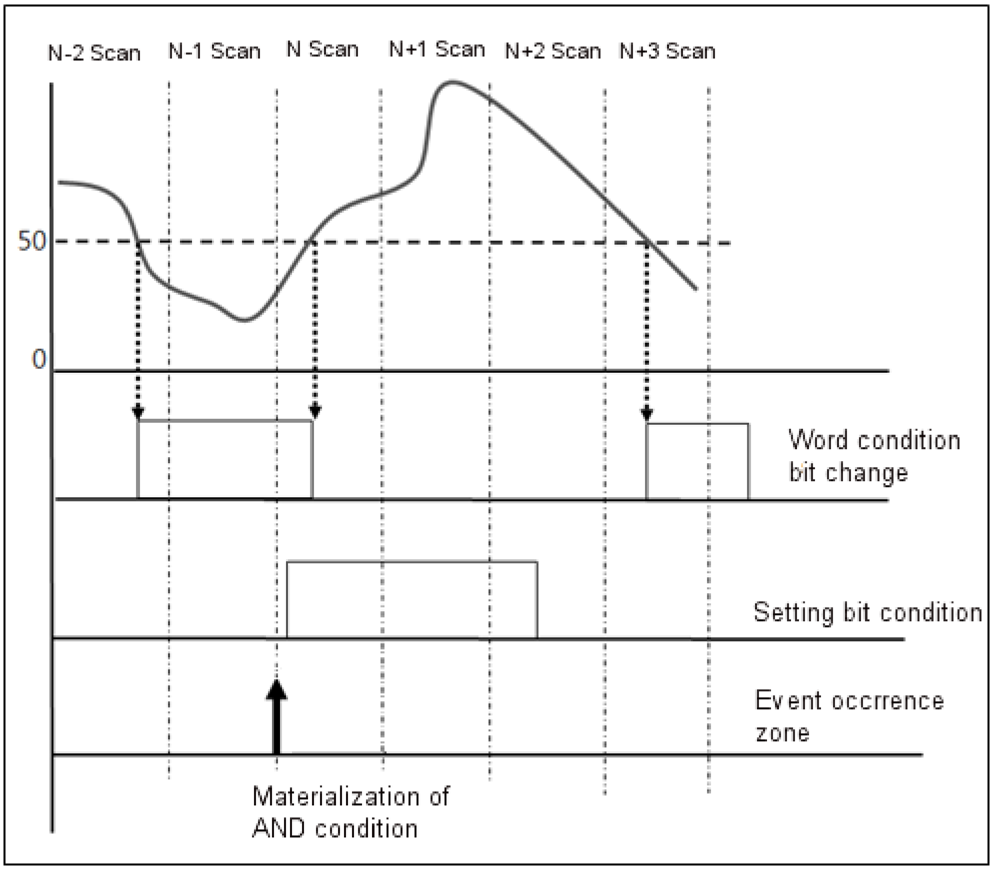
☞ When setting only with BIT condition

	Condition	Set Device	Event Occurrence Condition
Condition 0	BOOL	%MX1010	Elevation
Condition 1	BOOL	%IX1	
Condition 2	BOOL	%MX2010	
Condition 3	BOOL	%QX130	



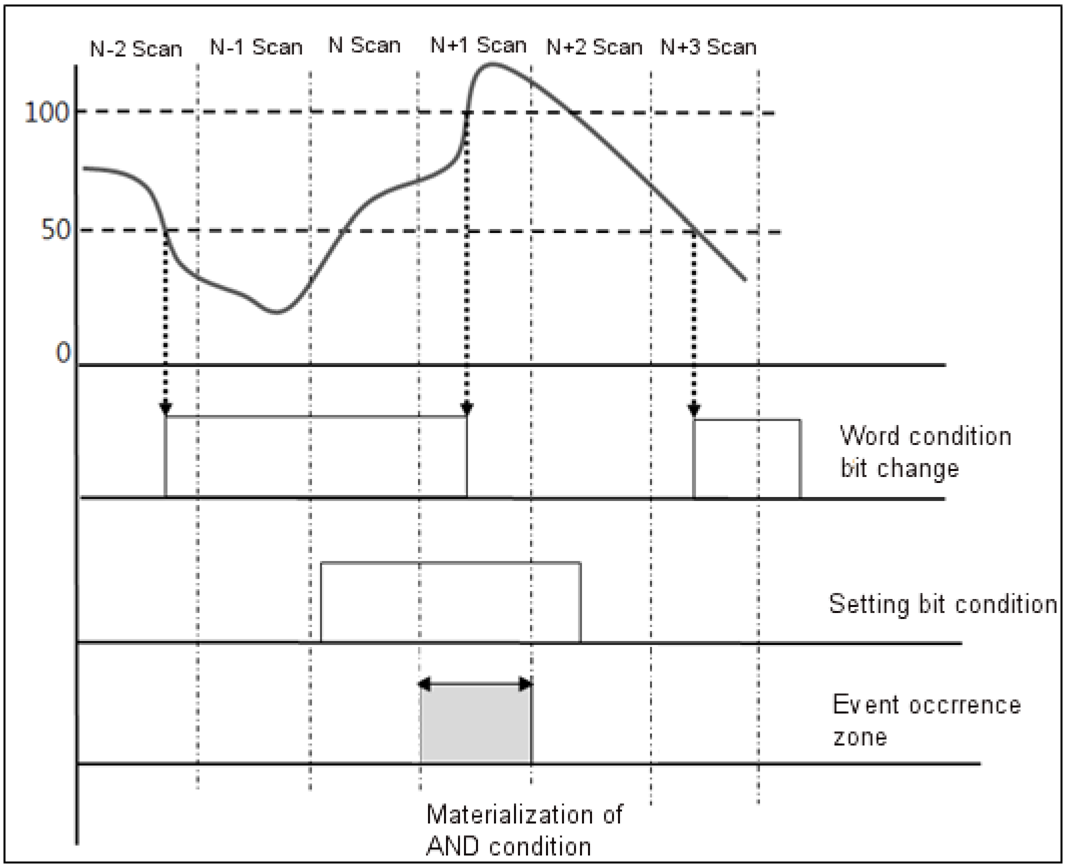
☞ When setting with combination of BIT and WORD conditions (no release value set)

	Condition	Comparison Condition	Set Value	Release Value	Set Device	Event Occurrence Condition
Condition 0	Word	<	50	-	%MW100	Elevation
Condition 1	BOOL				%MX15	



☞ When setting with combination of BIT and WORD conditions (release value set)

	Condition	Comparison Condition	Set Value	Release Value	Set Device	Event Occurrence Condition
Condition 0	WORD	<	50	100	%MW100	Elevation
Condition 1	BOOL				%MX15	

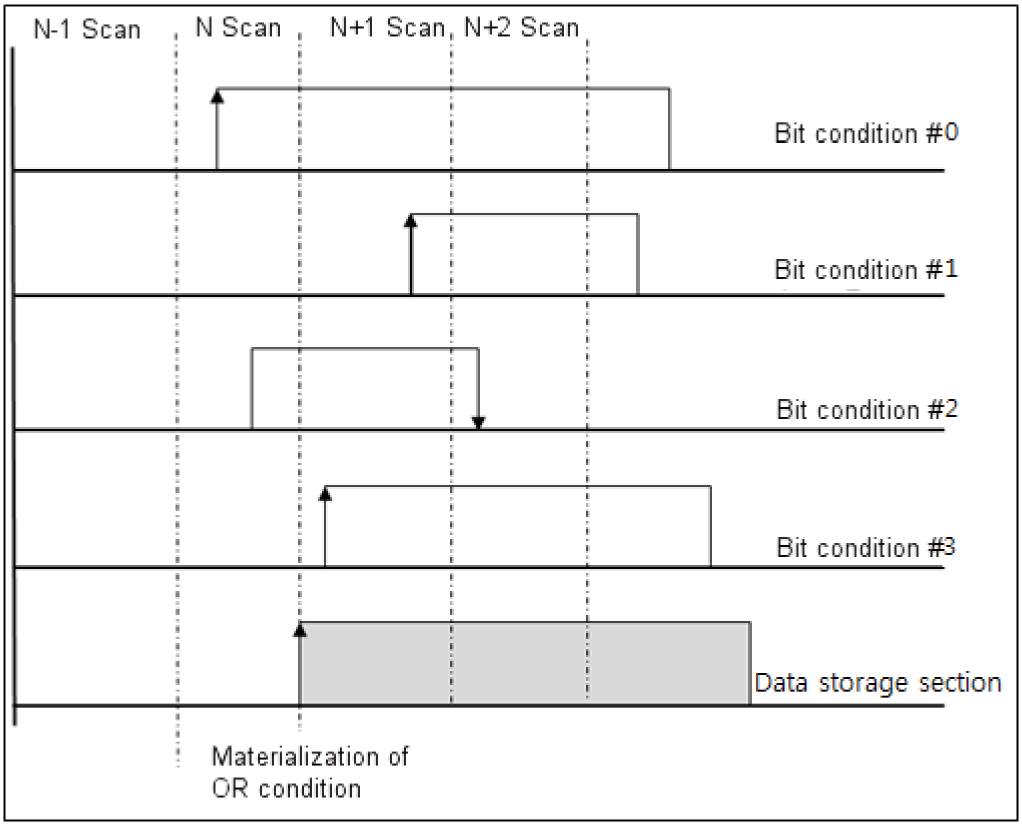


(b) OR Calculation

Event occurs when even one condition is satisfied at a single main task. After selecting Trigger Save, if the Trigger Condition is again satisfied before data saving is complete, and the trigger reoccurrence flag value increases.

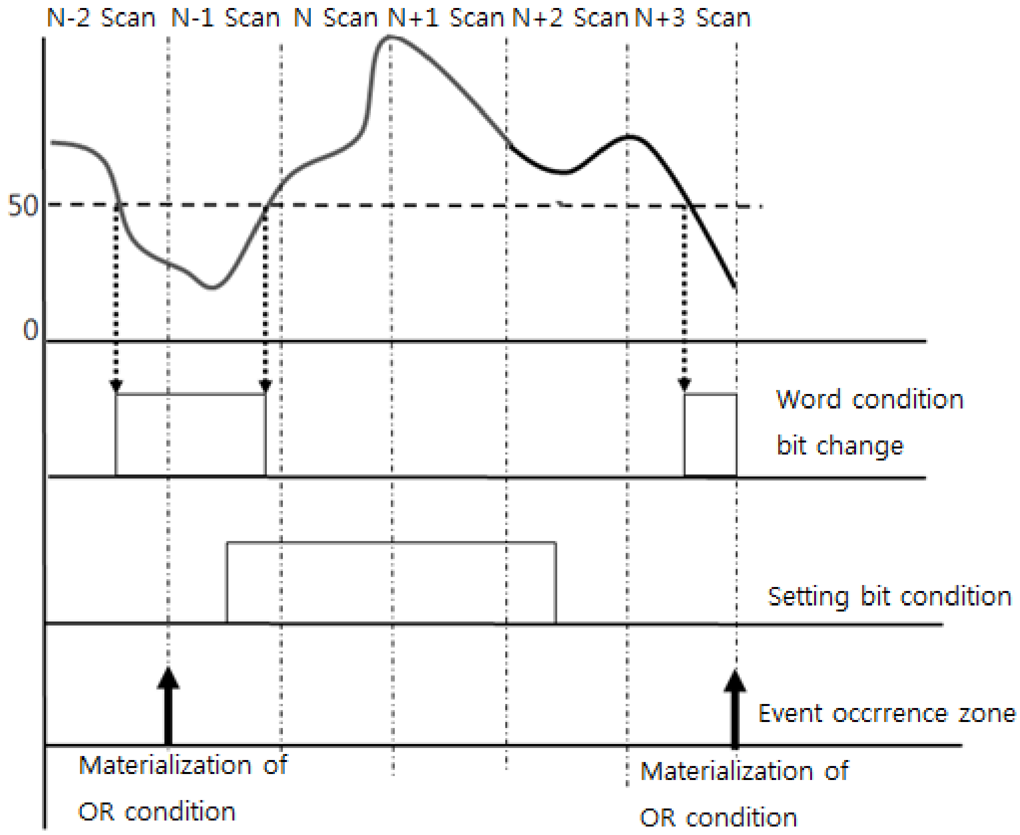
☞ When setting only with BIT condition

	Condition	Set Device	Event Occurrence Condition
Condition 0	BOOL	%MX1010	ON
Condition 1	BOOL	%IX1	
Condition 2	BOOL	%MX2010	
Condition 3	BOOL	%QX130	



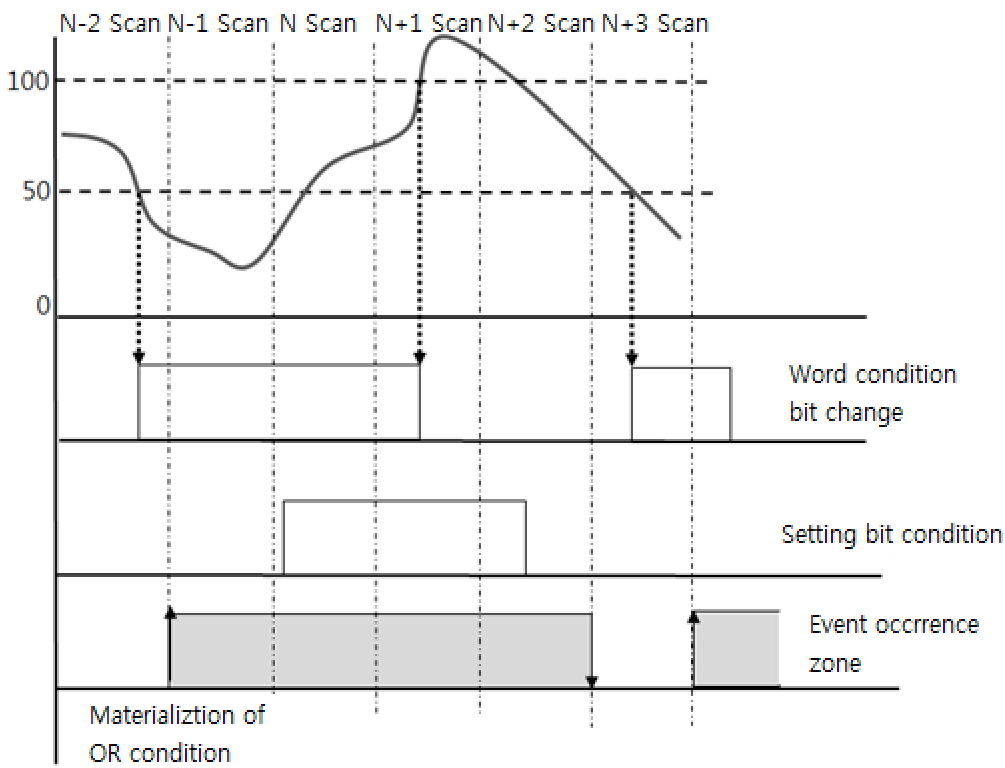
☞ When setting with combination of BIT and WORD conditions (no release value set)

	Condition	Comparison Condition	Set Value	Release Value	Set Device	Event Occurrence Condition
Condition 0	WORD	<	50	-	%MW10	Elevation
Condition 1	BOOL				%MX15	



☞ When setting with combination of BIT and WORD conditions (release value set)

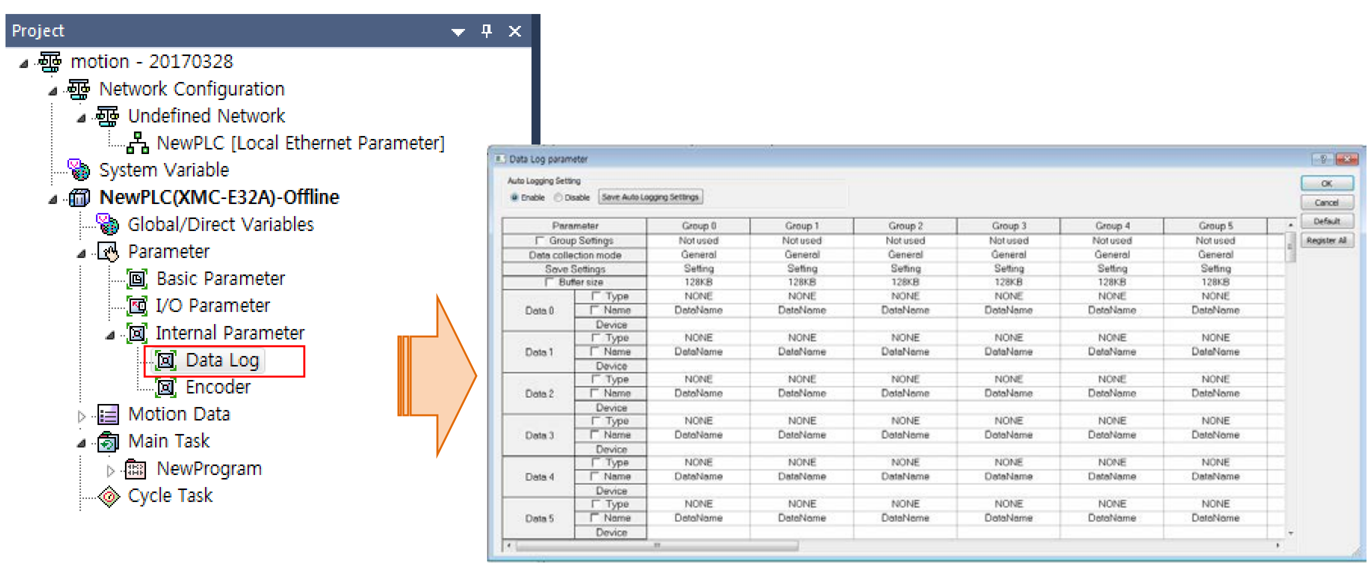
	Condition	Comparison Condition	Set Value	Release Value	Set Device	Event Occurrence Condition
Condition 0	Word	<	50	100	%MW10	ON
Condition 1	BOOL				%MX15	



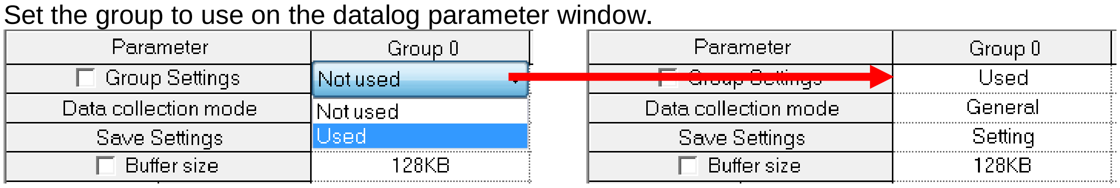
11.6.2 Setting Method

(1) Single BIT Condition

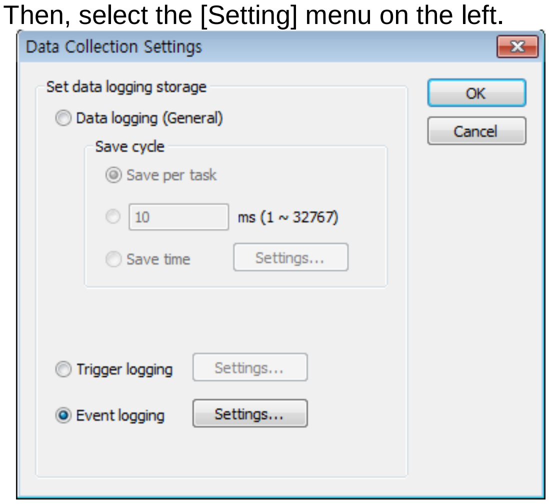
(a) Choose XG5000 – [Project Window] - [internal parameter] - [datalog]
This activates the datalog parameter setting window.



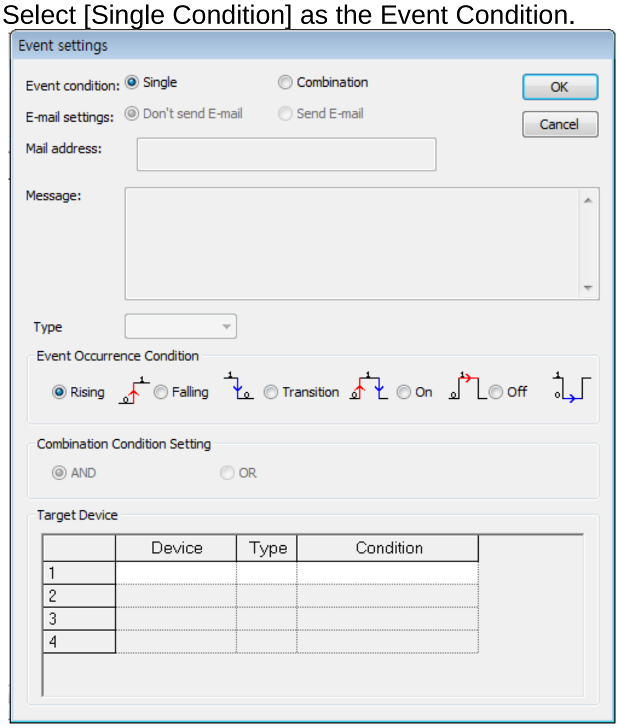
(b) Set the group to use on the datalog parameter window.



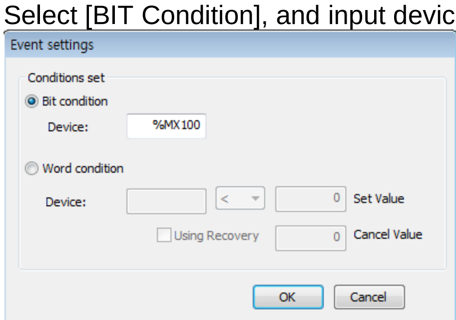
(c) Select [Event Logging] at [Data Collection Method] to activate [Setting] menu on the left.
Then, select the [Setting] menu on the left.



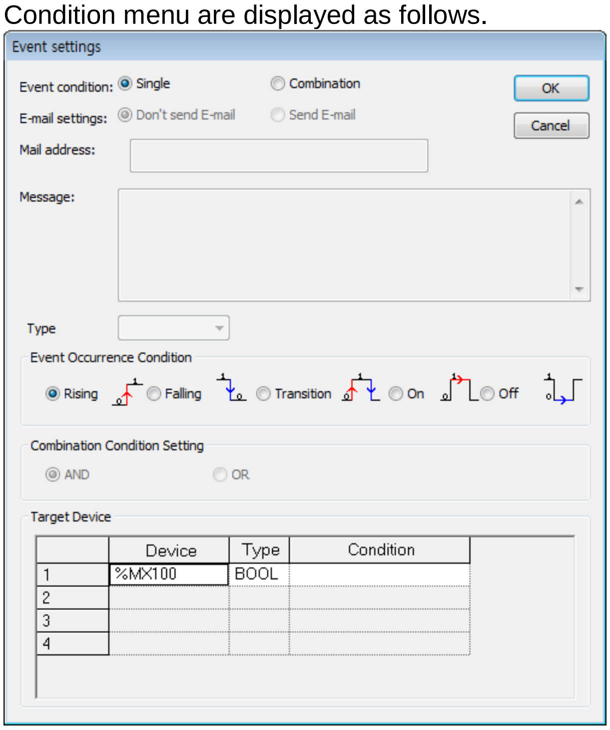
(d) Upon selection, the following window is activated for event setting.
Select [Single Condition] as the Event Condition.



(e) Select the condition setting menu to activate the following setting window.
Select [BIT Condition], and input device values into the device window in BIT types.



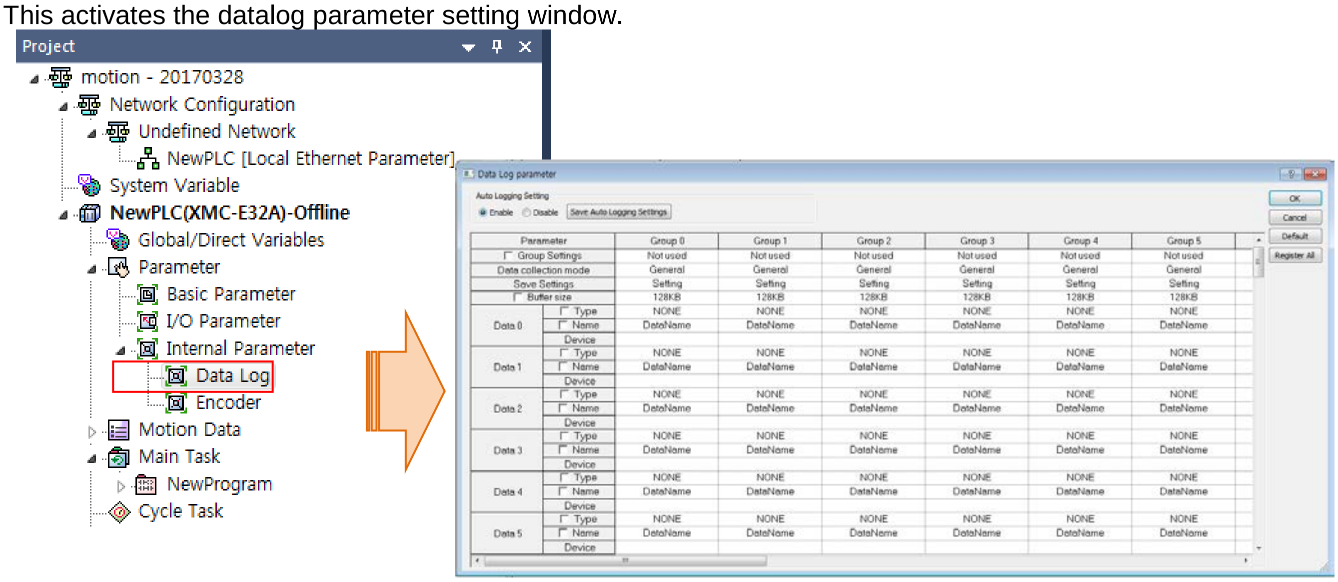
When setting is complete, the window closes and the conditions initially set at the Event Setting Condition menu are displayed as follows.



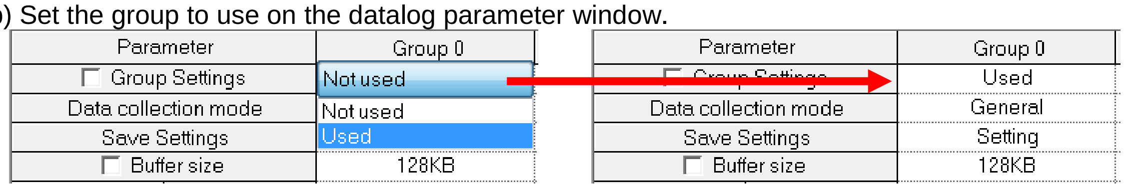
(f) Select the timing of data saving at the Event Occurrence Condition. The number and timing of data change depending on the set value.

(2) Single WORD Condition

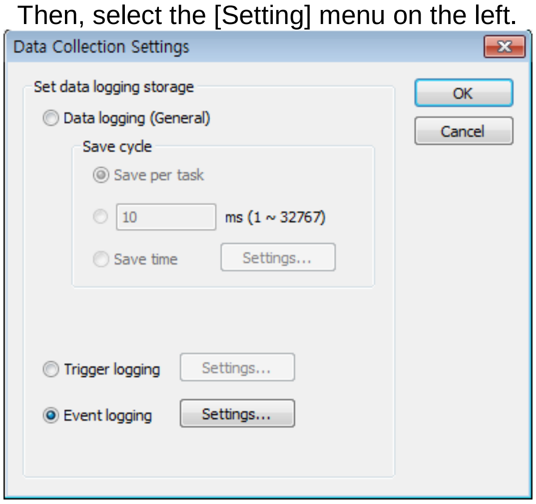
(a) Choose XG5000 – [Project Window] - [internal parameter] - [datalog]
This activates the datalog parameter setting window.



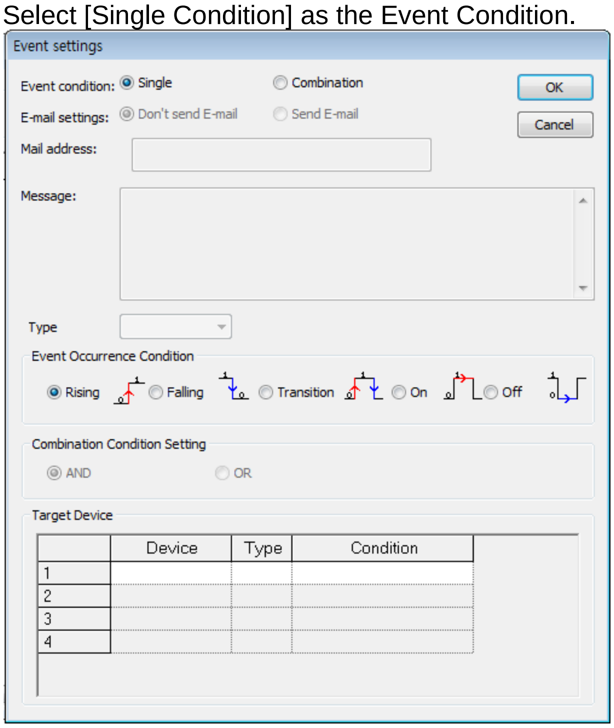
(b) Set the group to use on the datalog parameter window.



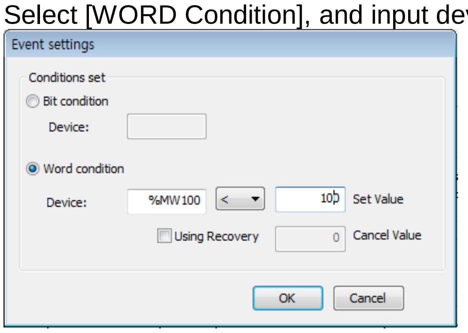
(c) Select [Event Logging] at [Data Collection Method] to activate [Setting] menu on the left.
Then, select the [Setting] menu on the left.



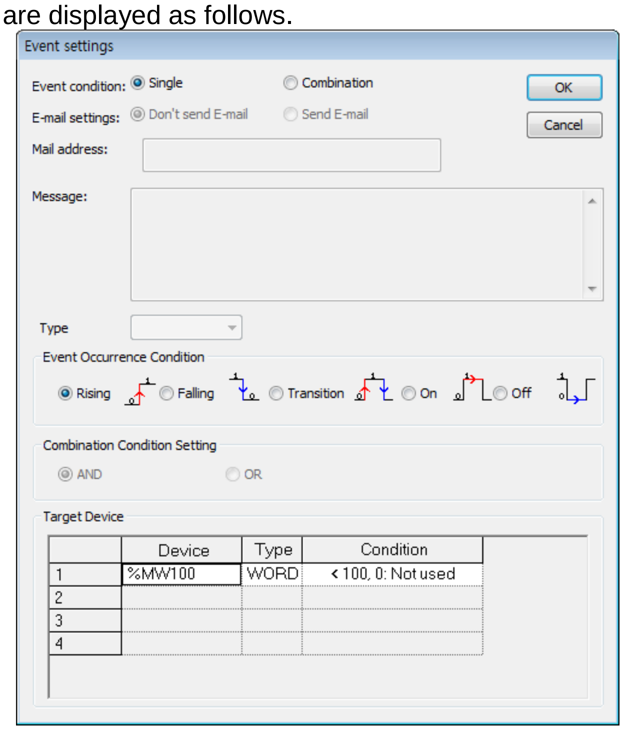
(d) Upon selection, the following window is activated for event setting.
Select [Single Condition] as the Event Condition.



(e) Select the condition setting menu to activate the following setting window.
Select [WORD Condition], and input device values into the device window in BIT types.



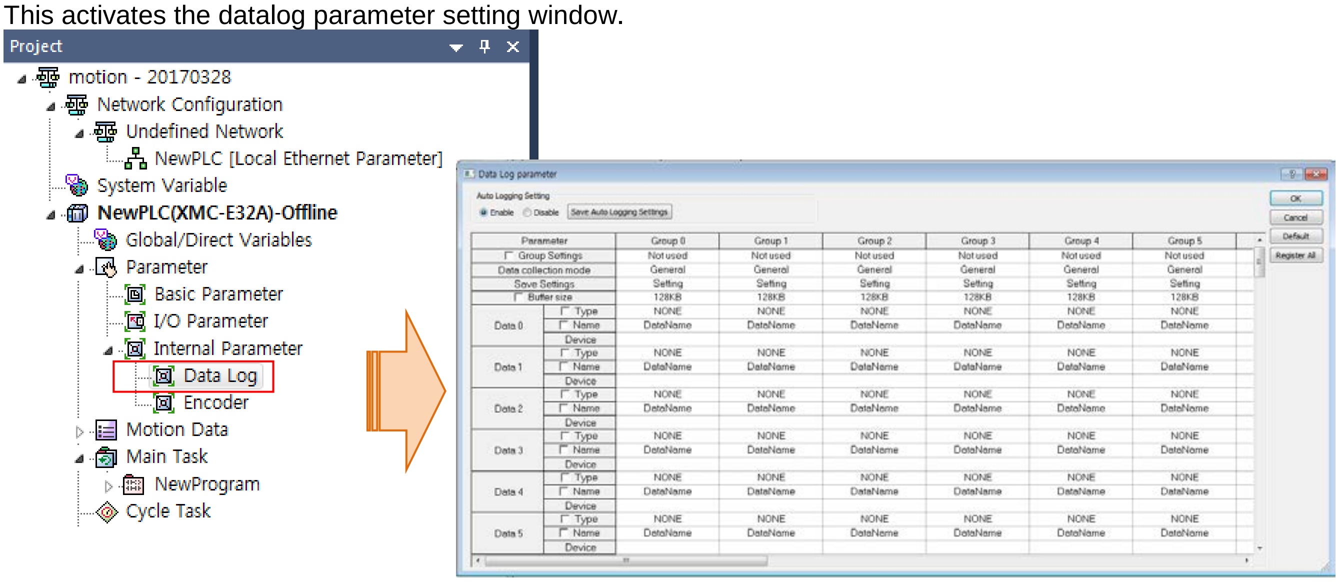
When setting is complete, the window closes and the conditions initially set at the Event Setting Condition menu are displayed as follows.



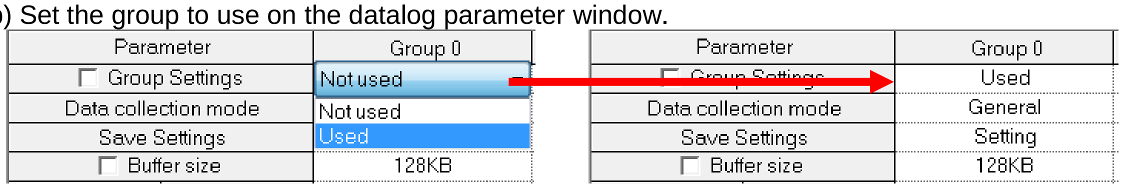
(f) Select the timing of data saving at the Event Occurrence Condition. The number and timing of data change depending on the set value.

(3) Multiple AND Condition

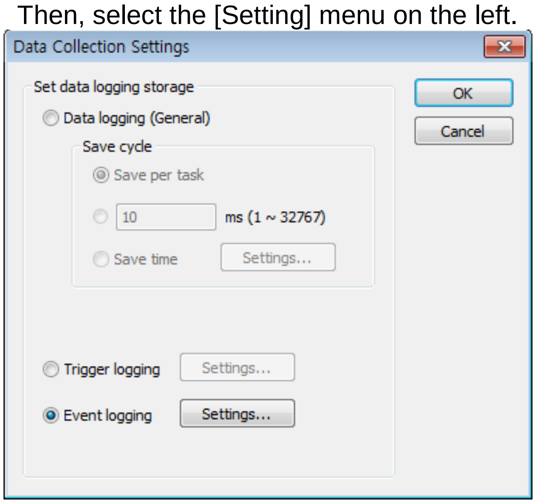
(a) Choose XG5000 – [Project Window] - [internal parameter] - [datalog]
This activates the datalog parameter setting window.



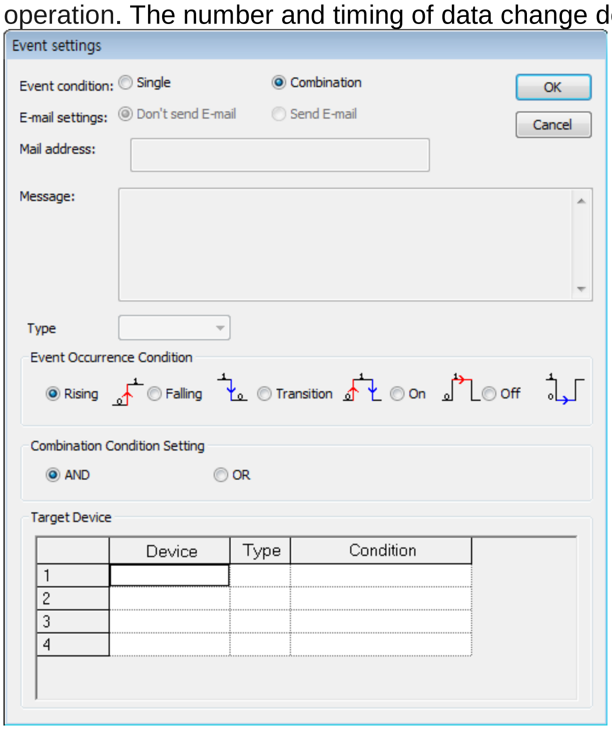
(b) Set the group to use on the datalog parameter window.



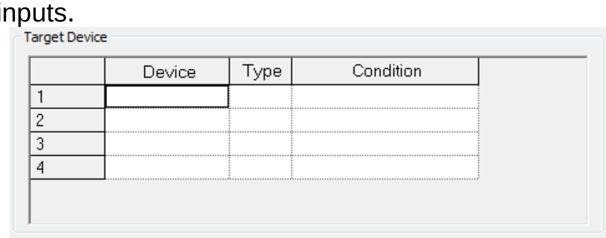
(c) Select [Event Logging] at [Data Collection Method] to activate [Setting] menu on the left.
Then, select the [Setting] menu on the left.



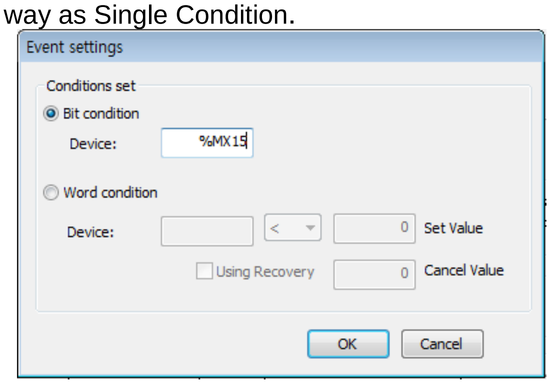
(d) Select the timing of data saving at the Event Occurrence Condition and set the operation condition to AND operation. The number and timing of data change depending on the set value and Time.

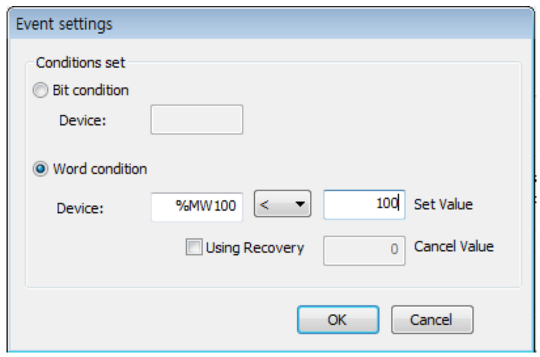


(e) Select [Event Condition] and [Multiple Condition] to activate the condition setting window which allows for up to 4 inputs.

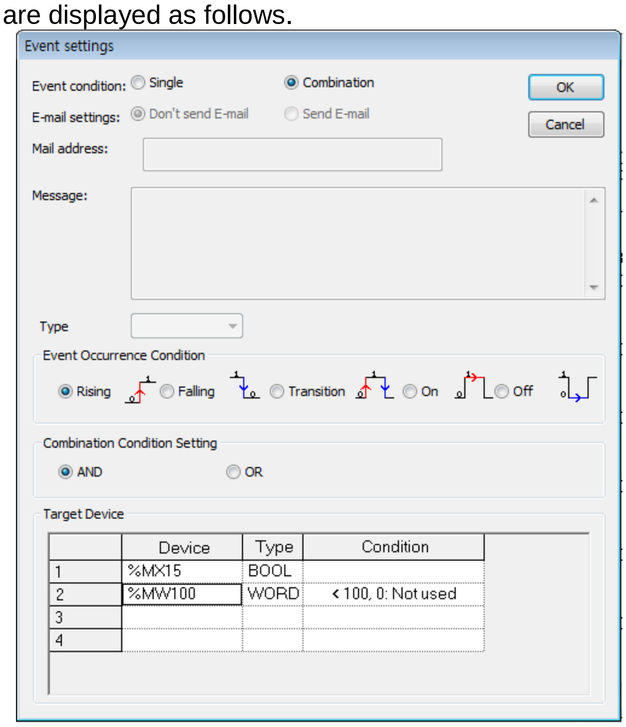


(f) Select each condition setting menu one by one, inputting specific set values. [Multiple Condition] activates Event Condition by calculating [Single Conditions] using the set run method. The basic setting is performed in the same way as Single Condition.

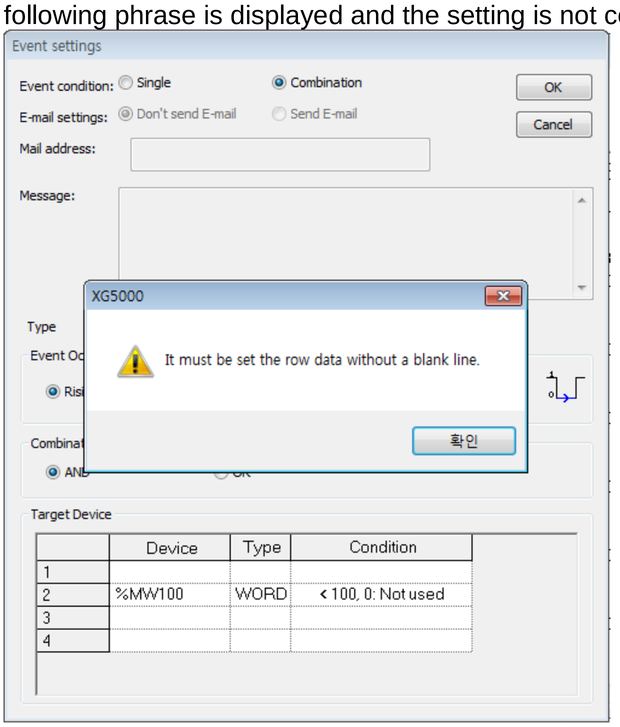




(g) When setting is complete, the window closes and the conditions initially set at the Event Setting Condition menu are displayed as follows.

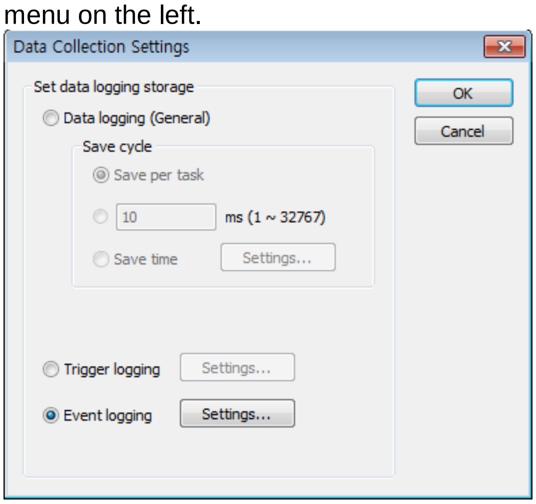


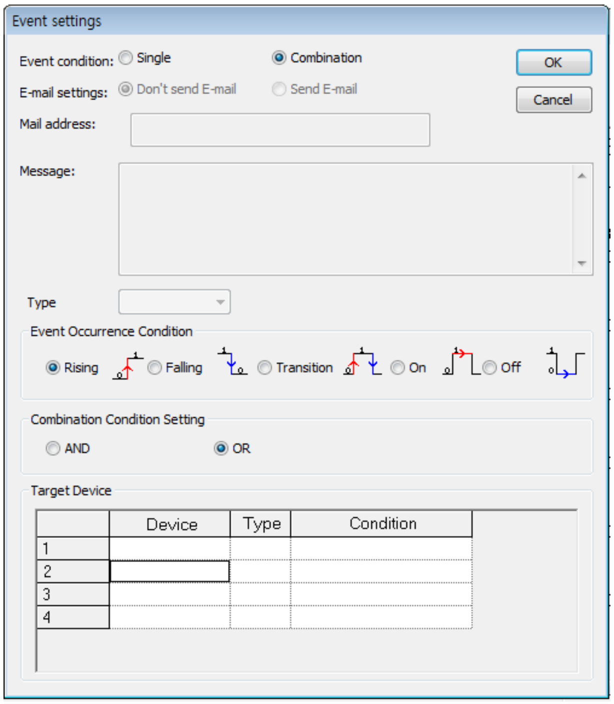
If only one [Condition Setting] is input after selecting Calculation Condition before finishing the setting, the following phrase is displayed and the setting is not complete.



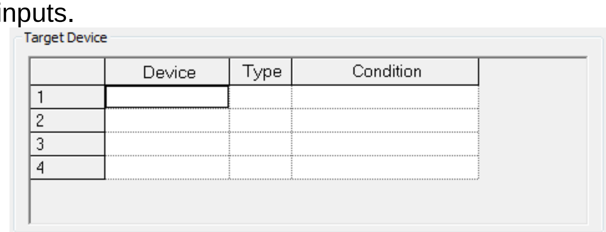
(4) Multiple OR Condition

(a) The same sequence as [AND Calculation Condition] applies up to the [Event Setting] menu.
(b) Select [Event Logging] at [Data Collection Method] to activate [Setting] menu on the left. Then, select the [Setting] menu on the left.



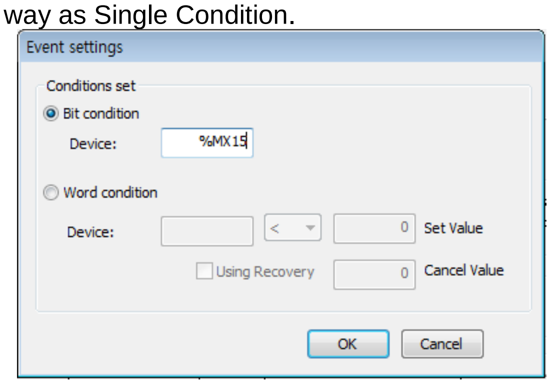


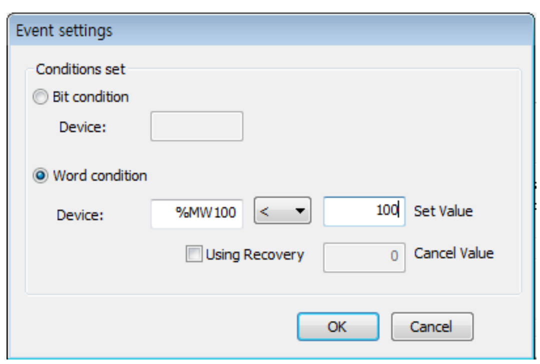
(c) Select [Event Condition] and [Multiple Condition] to activate the condition setting window which allows for up to 4 inputs.



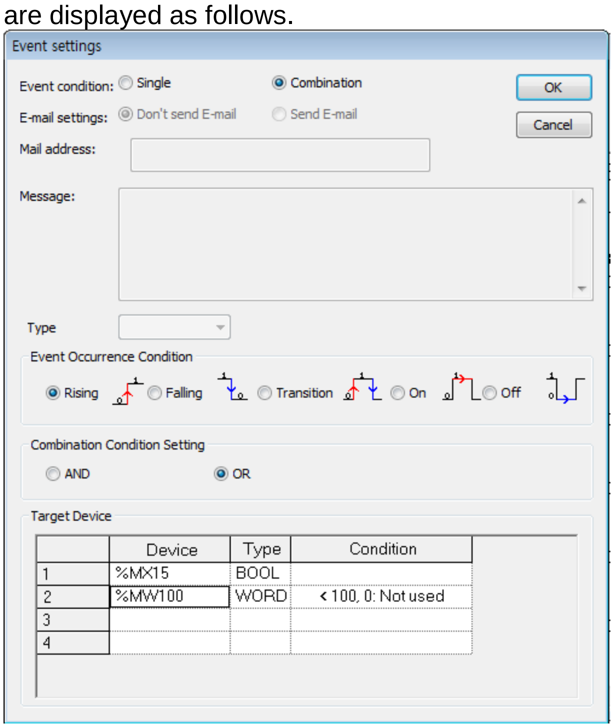
(d) Select the timing of data saving at the Event Occurrence Condition and set the operation condition to OR operation. The number and timing of data change depending on the set value.

(e) Select each condition setting menu one by one, inputting specific set values. [Multiple Condition] activates Event Condition by calculating [Single Conditions] using the set run method. The basic setting is performed in the same way as Single Condition.

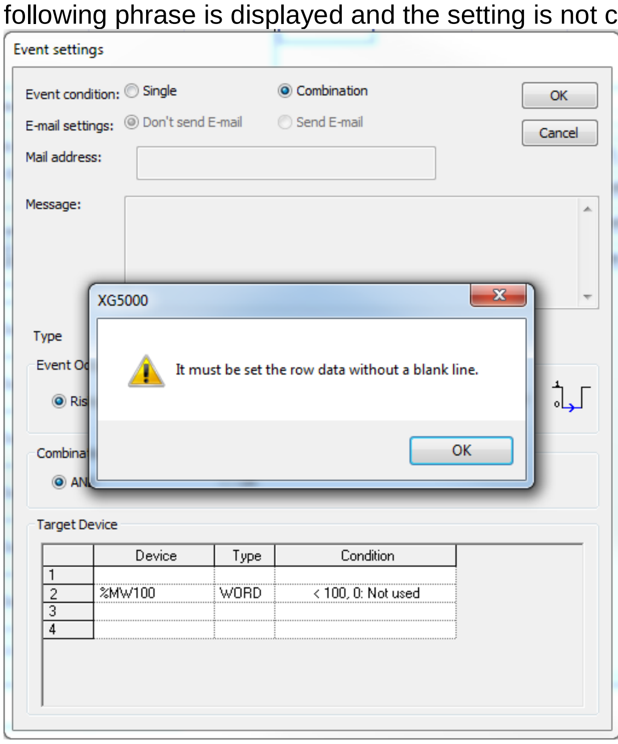




(f) When setting is complete, the window closes and the conditions initially set at the Event Setting Condition menu are displayed as follows.



(g) If only one [Condition Setting] is input after selecting Calculation Condition before finishing the setting, the following phrase is displayed and the setting is not complete.



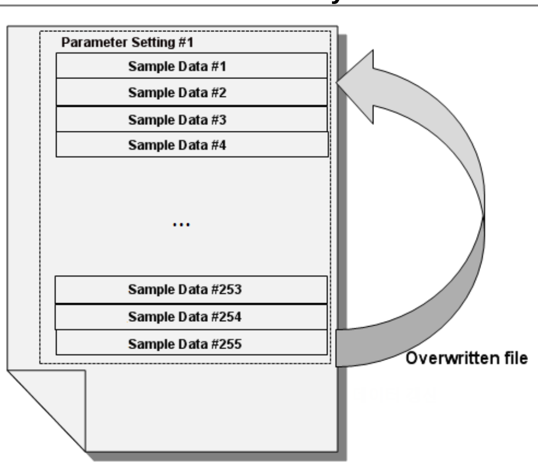
11.7 Additional Functions

This section provides detailed description of additional functions of internal datalog

11.7.1 File Save History Setting

When the maximum number of files are saved into the datalog, file save changes depending on whether [Overwrite with Latest History] or [Maintain First History] is chosen at the [History Setting

Overwrite with the latest history



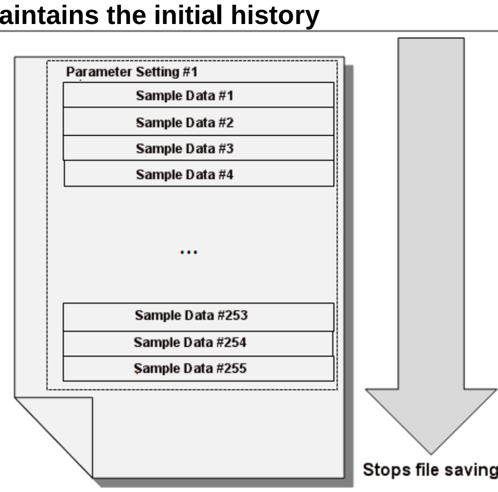
☞ Saves data in the maximum number of saved files (256 files/folder), and then goes back to the beginning to delete old files, and save the latest history.

☞ When the maximum files are saved after selecting [Overwrite with Latest History], the file save excess flag value increases. (See 5.10, Flag List)

☞ If the 10% or less of the SD memory storage is free, the data are written over the file first saved.

☞ The overwritten file has the same size as the previous one.

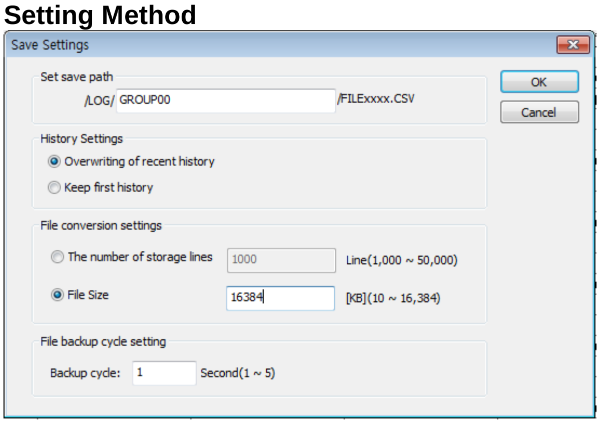
Maintains the initial history

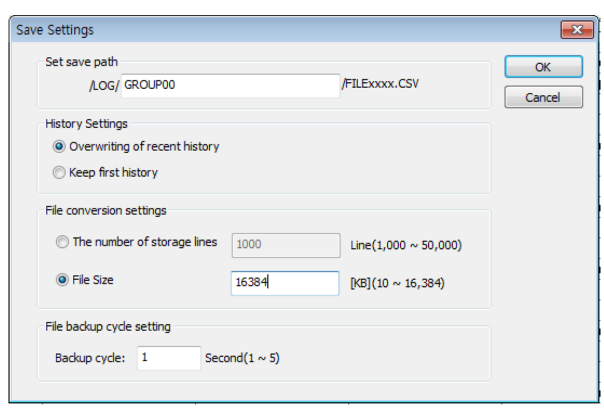


☞ Saves data in the maximum number of saved files (256 files/folder), and then stops file saving.

☞ If the 10% or less of the SD memory storage is free, stops file saving.

Setting Method





11.7.2 Formatting Function

Internal datalog supports SD memory formatting function. SD memory formatting is done through XG5000.
SD memory formatting is supported only when motion controller is in STOP mode.

(1) Formatting Specifications
The SD memory formatting supported by datalog has the following specifications.

Item	Set Specifications
File System1)	FAT32
Supported SD memory Capacity2)	2GByte ~ 32GByte
Allotted Cluster Size3)	32KByte (512 Sector4) x 8)
Volume Label5)	LSIS (fixed)
Motion controller Operation Mode6)	STOP (REMOTE available)
Formatting Mode7)	Fast Formatting

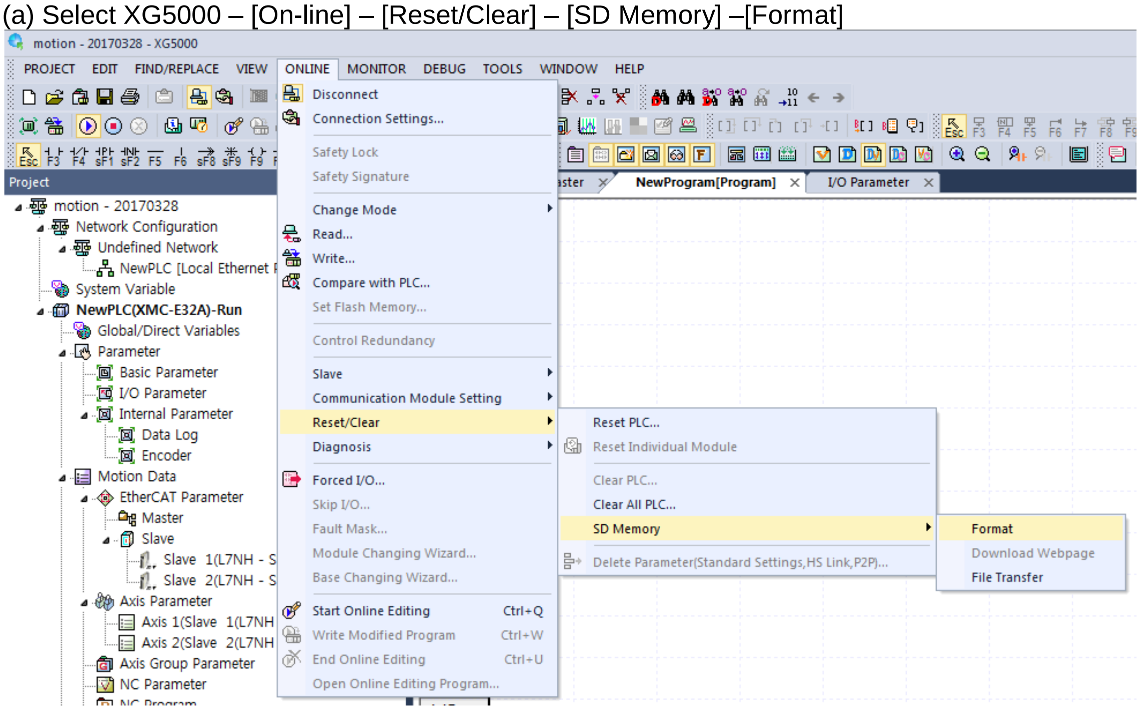
(a) File System: Rules of Saving Files into Disk
(b) Supported SD memory Capacity: MMC card not supported, 2GByte~ 32GByte SD memory supported (Micro SD, SDHC supported)
(c) Allotted Cluster Size: 32kByte
(d) Sector: Minimum Unit for Data Saving (Default: 512 Byte)
(e) Volume Label: SD memory Card Name
(f) Motion Controller Operation Mode: Operates only in STOP mode
(g) Formatting Mode: Fast-formats the SD memory only deletes the FAT and directory area within the file system.

Note

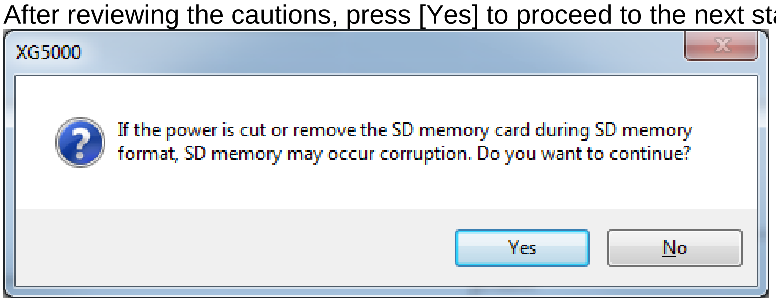
1. When performing [Formatting Function] at motion controller, all contents within the SD memory are deleted, followed by creation of a folder with the name set by the parameter.
2. If the SD card file system is not FAT32, the format function is not supported in the XG5000. In an attempt to format the SD card other than FAT32, a warning window “No file system other than FAT32 is supported” is created. Please use after formatting to FAT32 in PC

(2) Execution

(a) Select XG5000 – [On-line] – [Reset/Clear] – [SD Memory] –[Format]



(b) Before executing SD memory formatting, cautions for formatting process are activated..
After reviewing the cautions, press [Yes] to proceed to the next stage.

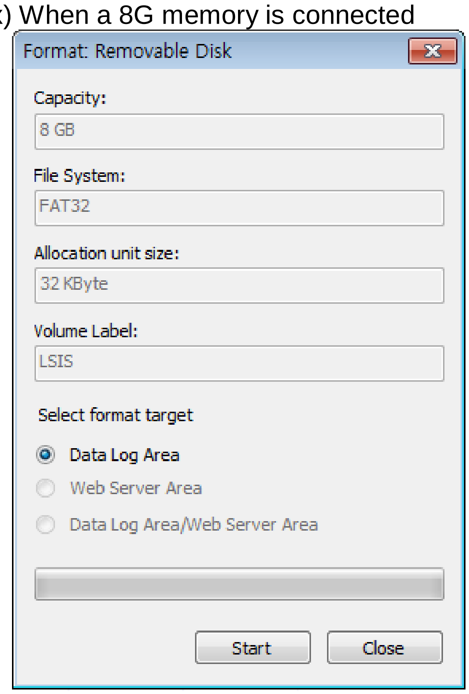


Caution

1. Detaching the SD memory with force, power off or reset during formatting may cause internal damage of the connected card, which may not show normal run afterwards.
2. If SD memory is being recognized when connected, the formatting can be performed after the operation is completed.
3. Check RD/WRLED and relevant flags when SD memory is connected. If the motion controller mode is changed while the formatting is in progress, the formatting will not be performed normally (supported only in STOP state).
4. Transition to RUN mode is not possible during formatting.

(c) Subsequently the formatting setting window is activated. The setting window is as follows.
The storage, file system and allotted unit size are Default values that are read when connecting the SD memory.
Also, only fast formatting is supported. Volume label should be in English, and can be as long as 10 characters.
After setting as indicated above, press [Start] to begin formatting. The status bar indicates the current progress.

Ex) When a 8G memory is connected



(3) Formatting Complete and Error Codes

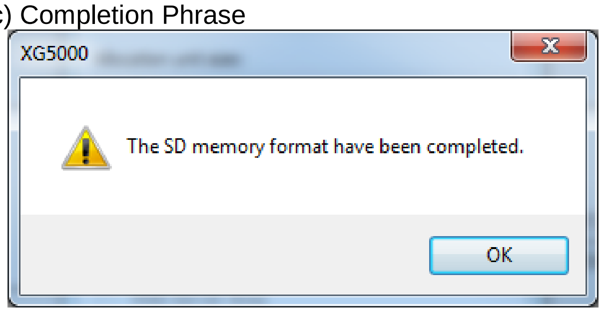
(a) Status Information

F Area Address		Flag Name	Description
%FW523		_SD_FmtInfo	SD memory formatting information
BOOL	%FX8368	_SD_FmtRun	SD memory formatting in progress
	%FX8369	_SD_FmtDone	SD memory formatting complete
	%FX8370	_SD_FmtNg	SD memory formatting failed
%FW524		_SD_FmtEcode	SD memory formatting error codes

(b) Error Code

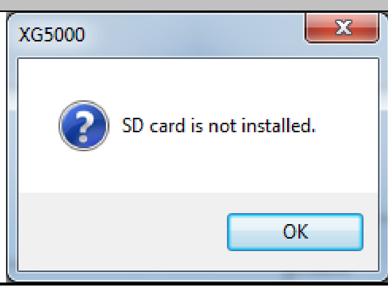
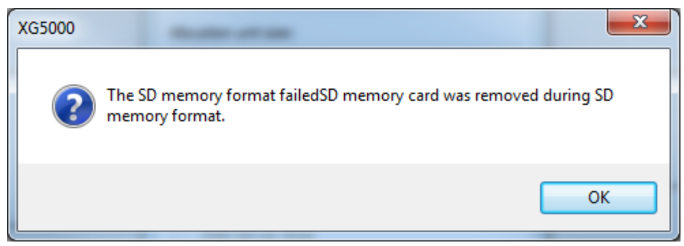
Error Code	Error Name	Error Description
0x0001	SD Detachment	When the SD memory card is forcibly removed during SD memory formatting
0x0002	File System Damage	When the file system is damaged during SD memory formatting

(c) Completion Phrase



_SD_FmtDone (%KX8369) Bit turns ON when formatting is complete. In this case, the following completion window appears.

If formatting failed, an error window appears along with the relevant code.

SD Detachment	File System Damage
	

11.7.3 Diagnosis Function

Datalog provides SD memory diagnosis function.
SD memories that do not comply with the following cannot be used. Datalog function will not be executed when such memories are connected.

(1) FAT32 File System Diagnosis

☞ The memory should be formatted using the FAT32 format, to allow for file saving.
Files will not be saved if it is formatted using other formats.

Caution

Since sudden power off may cause file system / file damage or saving of abnormal data.
Therefore, make sure to execute STOP flag or push the SD CMD button for 2 second when trying to stop datalog function, so as to ensure normal data saving.

11.8 CSV File Structure

11.8.1 File Save Format

The name of CSV files are created in the following form.

Name	F	I	L	E	0	0	0	0	0	.CSV
Description	File Name				Group Number		File Number			Extension
Range	Fixed Value				0~15		000 ~ 255			Fixed Value

The first 4 characters are fixed as 'FILE,' and the 5 ~ 6th numbers indicates the group number selected, and the following
7~9th numbers indicate the file number.
For example, the 8th file of the 'GROUP11' folder will be named 'FILE11008.CSV.'

11.8.2 File Name and Save Sequence

When executing datalog function after selecting a certain group, the file sequence progresses from Number 0. When executing datalog function on multiple groups, files are first created for Group 0, and progresses sequentially to Group 15.
Selecting [Do Not Use] at the [Group Setting] will stop fie saving in the current group, and files creating will move into the next group.

Group Name	Group 0	Group 1	Group 2	Group 3	Group 4	Group 5	…	Group 15
File Name and Creation Sequence	LOG0000	LOG1000	LOG2000	LOG3000	LOG4000	LOG5000		LOG9000
	LOG0255	LOG1255	LOG2255	LOG3255	LOG4255	LOG5255		LOG15255

Note

While the data value collected from motion controller is saved at the interval set by the parameter, saving into the SD memory is performed using Main Task Save method, starting from Group 0. However, it can change flexibly depending on file storage.

11.9 SD Memory Card

11.9.1 SD Memory Specifications

To use datalog function, the SD memory used should satisfy the following specifications.

Items	Description
Memory Capacity:	Up to 32 GB (supports SPI MODE, SD, SDHC) (Only 8GB can be available in more than 8GB memory)
File System	FAT32
Voltage Range	2.7 ~ 3.6V
Working Temperature Range	-25℃ ~ 85℃
Static Tolerance	Should satisfy IEC61000-4-2
Number of Detachments	Up to 10,000 times
Current Consumption	Up to 100mA (when reading, writing)
Number of Read/Writes	Up to 100,000 times (for SLC)
Size	15mm * 11mm * 1mm
Recommended Products	SanDisk, Transcend

Note

1. Datalog function of motion controller is capable of using all SD memories that satisfy the specifications above.
2. Optimal performance can be expected by using the recommended products (SanDisk, Transcend). Please use the recommended products unless required otherwise

11.9.2 Caution

Please pay attention to the following when using datalog function with SD memory card.

(1) Power Off during SD Memory Writing

(a) Power off or motion controller reset during writing of data collected by motion controller into the SD memory may damage the file system of the memory card. Although motion controller verifies the file system of the SD memory when applying electric power to convert the damaged files into usable files, such restoration may not be possible depending on the level of damage. When powering off motion controller, please perform power off after verifying that the SD memory writing is not being performed.

(b) Power off or motion controller reset during writing of data collected by motion controller into the SD memory causes all data saved in the buffer memory inside the buffer memory. Therefore, the data collected immediately before power off may not have been saved properly. When powering off motion controller, please perform power off after verifying that the SD memory writing is not being performed.

(2) Time Required when Suspending SD Memory Writing

In cases of using K area flag to turn off the datalog permission flag while data saving is in progress, all data collected before reception of the relevant flag command are saved into the SD memory, and then the datalog operation stops.

Therefore, a small time is required until the datalog function actually stops. The time required for datalog stop varies depending on the volume of data collected.

While the datalog stop is being performed, DLxx_Stoping (xx is the group number) flag is turned on, and when the stop is completed, _DLxx_Finish flag is turned on. During the stop, the size of the remaining dada to be saved to the SD card is displayed in the _DLxx_WaitingData flag.

(3) Removal of Memory Card during Read/Write in SD Memory

(a) Forcibly removing the SD memory from motion controller during writing or reading of data collected by motion controller may damage the file system of the memory card. Therefore, please remove SD memory after disabling the datalog function using the command flag. If SD memory is removed during read/write of the SD memory, the SD STATE LED flashes at 500ms interval.
The following figure shows the sequence of disconnecting or exchanging SD memory card.

(b) Power off or reset during datalog run may cause abnormal data saving. Also, the file system may be damaged and not recognized the SD memory and the files.

(4) Using the cover to prevent SD memory removal
Please set the direction correctly when connecting the SD memory to the motion controller.
If you want to remove the SD memory, press the SD memory deeply to remove the memory.
In addition, please use the cover to prevent the SD memory from being removed due to vibration.

11.9.3 Micro SD Memory Usage Capacity

Up to 32GB memory can be mounted on the motion controller, but only about 80% of 8GB can be used. Even if the SD card with more than 8GB is installed, the capacity corresponding to about 80% of 8GB, not 80% of the total capacity can be used, and therefore it can no longer be used when more than 6.4GB is used. This is to prevent excessive increase of SD memory access time when the data is stored.

Caution

1. SD memory state may affect main task time and saving performance. SD memory should be formatted before use.
2. When using the SD memory for a long time, formatting on a regular basis is required to maintain performance.

11.10 Flag List

11.10.1 Common Flag

Address	Type	Variable	Function	Description
%KX8800	BOOL	_DL_Rdy	Datalog ready	It is the flag indicating whether the datalog is ready.
%KX8192	BOOL	_DL_AutoLogStop	Stop Auto-logging	It is the flag indicating stop command input of auto-logging.
%KX8801	BOOL	_DL_Err	Datalog error state	It is the flag indicating error state of the datalog.
%KX8256	BOOL	_SD_Attach	SD attachment state	It is the flag indicating attachment state of SD memory.
%KX8257	BOOL	_SD_Rdy	SD memory ready	It is the flag indicating whether the SD memory is enabled.
%KX8258	BOOL	_SD_Err	SD memory error	It is the flag indicating error state of SD memory.
%KX8259	BOOL	_SD_Init	SD memory initializing state	It is the flag indicating initialization state of SD memory.
%KX8260	BOOL	_SD_Closing	SD memory closing state	It is the flag indicating closing state of SD memory.
%KX8261	BOOL	_SD_FATErr	File System Error	It is the flag indicating error state of SD memory file system
%KX8262	BOOL	_SD_AutoLogAct	Act Auto-logging…	It is the flag indicating acting state of auto-logging.
%KX8263	BOOL	_SD_Busy	SD memory busy state	It is the flag indicating busy state of SD memory.
%KX8264	BOOL	_SD_SpaceWarn	SD memory insufficient state	It is the flag indicating insufficient state of SD memory capacity.
%KX8265	BOOL	_SD_Detach	SD memory detachment state	It is the flag indicating detachment state of SD memory.
%KD259	UDINT	_SD_VolTot	SD memory storage capacity(GB)	The Capacity of attachment SD memory (GB) (In case of 8GB or more, it is displayed as 8GB)
%KD260	UDINT	_SD_VolAvail	Available storage capacity(KB)	The usable capacity of SD memory (KB)
%KW522	WORD	_SD_Ecode	SD memory error code	It is the flag indicating error number of SD memory.
%KW523	WORD	_SD_FmtInfo	SD memory format information	It is the flag indicating format information of SD memory.
%KX8368	BOOL	_SD_FmtRun	SD memory format operation state	It is the flag indicating that the SD memory is formatting.
%KX8369	BOOL	_SD_FmtDone	SD memory format complete state	It is the flag indicating that the format of SD memory is completed normally.
%KX8370	BOOL	_SD_FmtErr	SD memory format fail state	It is the flag indicating that the format of SD memory is failed

Address	Type	Variable	Function	Description
%KW524	WORD	_SD_FmtEcode	SD memory format error code	It is the flag indicating error number that occurred while formatting the SD memory.
%KW525	WORD	_SD_FmtProgress	SD memory format progress ratio (%)	It is the flag indicating format progress ration of SD memory. (0~100(%))
%KW526	WORD	_SD_AttachCnt	SD memory attachment count	It is the flag indicating attachment count of SD memory.
%KW527	WORD	_SD_DetachCnt	SD memory detachment count	It is the flag indicating detachment count of SD memory.
%KX8640	BOOL	_SD_AddfuncAct	SD additional function operation state	It is the flag indicating that the additional function of SD memory is operating.
%KX8641	BOOL	_SD_AddfuncErr	SD additional function error state	It is the flag indicating that the additional function of SD memory is error state.
%KX8642	BOOL	_SD_AddfuncDone	SD additional function complete state	It is the flag indicating that the additional function of SD memory is completed operation state.
%KX8643	BOOL	_SD_CmpResult	SD result of comparison	It is the flag indicating comparison operation result of SD memory
%KW541	WORD	_SD_AddfuncKind	SD type of additional function	It is the flag indicating type of that the additional function of SD memory is.
%KW542	WORD	_SD_AddfuncEcode	SD additional function error code	It is the flag indicating error number that occurred while operating the additional function of the SD memory.

11.10.2 Group Specific Flag

(1) Parameter Group 0 Flag

Address	Type	Variable	Function	Description
%KX8224	BOOL	_DL00_Enable	Group 00 datalog enable state	0: Stop, 1: Save
%KX8960	BOOL	_DL00_Rdy	Group 00 datalog ready	0: Not ready, 1: Ready
%KX8961	BOOL	_DL00_Act	Group 00 datalog operation state	0: Stop, 1: Saving
%KX8962	BOOL	_DL00_Err	Group 00 datalog error state	0: No error, 1: Error
%KX8963	BOOL	_DL00_Stoping	Group 00 datalog stopping state	0: Not stopping, 1: Stopping
%KX8964	BOOL	_DL00_Finish	Group 00 datalog finish state	0: Incomplete, 1: Complete
%KX8965	BOOL	_DL00_Trig	Group 00 trigger occurrence state	0: Stop, 1: Operating
%KX8966	BOOL	_DL00_TrigDone	Group 00 trigger complete state	0: Incomplete, 1: Complete
%KX8967	BOOL	_DL00_Evt	Group 00 event occurrence state	0: Stop, 1: Operating
%KX8968	BOOL	_DL00_Ovf	Group 00 buffer overflow state	0: normal, 1: overflow
%KW561	WORD	_DL00_Ecode	Group 00 datalog error code	-
%KW562	WORD	_DL00_FileIdx	Group 00 datalog file index number	range :0~255
%KW563	WORD	_DL00_FileRollcnt	Group 00 overwrite count	-
%KD282	UDINT	_DL00_FileSize	Group 00 file size(Byte)	-
%KD283	UDINT	_DL00_DataRow	Group 00 data row number	-
%KD284	UDINT	_DL00_RemainBuf	Group 00 remaining buffer size(Byte)	-
%KD285	UDINT	_DL00_WaitingData	Group 00 waiting data size(Byte)	-
%KW572	WORD	_DL00_OvfCnt	Group 00 buffer overflow count	-
%KW573	WORD	_DL00_TrigCnt	Group 00 trigger occurrence count	-
%KW574	WORD	_DL00_TrigOvrap	Group 00 trigger overlap count	-
%KW575	WORD	_DL00_EvtCnt	Group 00 event occurrence count	-

(2) Parameter group 1th~15th flag

Address	Size	Variable	Function	Description
%KW580	20Word	-	-	Parameter of group 1(the same structure with group 0)
%KW600	20Word	-	-	Parameter of group 2(the same structure with group 0)
%KW620	20Word	-	-	Parameter of group 3(the same structure with group 0)
%KW640	20Word	-	-	Parameter of group 4(the same structure with group 0)
%KW660	20Word	-	-	Parameter of group 5(the same structure with group 0)
%KW680	20Word	-	-	Parameter of group 6(the same structure with group 0)
%KW700	20Word	-	-	Parameter of group 7(the same structure with group 0)
%KW720	20Word	-	-	Parameter of group 8(the same structure with group 0)
%KW740	20Word	-	-	Parameter of group 9(the same structure with group 0)
%KW760	20Word	-	-	Parameter of group 10(the same structure with group 0)
%KW780	20Word	-	-	Parameter of group 11(the same structure with group 0)
%KW800	20Word	-	-	Parameter of group 12(the same structure with group 0)
%KW820	20Word	-	-	Parameter of group 13(the same structure with group 0)
%KW840	20Word	-	-	Parameter of group 14(the same structure with group 0)
%KW860	20Word	-	-	Parameter of group 15(the same structure with group 0)

* _DLxx_Enable (Datalogging Enable Flag per Group) is set to each Bit of %KW514

11.10.3 Error Code and Solution

Error codes related to datalog function is as follows.

Items	Error Code	Error Name	Cause and Solution	Note
Overall Error Codes	0x0000	No Error	-	
	0x0001	SD Card Recognition Error	It occurs when SD card is damaged, or SD which is not formatted to FAT32 is mounted. Format it with FAT32 and try mounting it again. If it is still not recognized, replace the SD card.	
	0x0002	Partition Information Error	Failed to read partition information. Format it with FAT32 and try mounting it again. If it is still not recognized, replace the SD card.	
	0x0003	File System Error	Format in FAT32 format and connect SD memory.	
	0x0004	SD Card Not Supported	Please connect SD Card with storage of 2GB~16GB.	
	0x0005	SD Card Capacity Check Failed	SD memory capacity test failed, and thus SD cannot be used. Replace SD memory or re-connect it after formatting.	
	0x0006	Lack of Free Space for SD Card	The available capacity of SD card is less than 20% of the maximum capacity. (In case of SD card with more than 8GB, about 6.4GB, which is about 80% of 8GB, is used.) Replace it with another SD card or secure the capacity by formatting and then re-connect it.	
	0x0007	Folder Creation Failed	Datalogging group folder cannot be created in SD card. Replace the SD card if it is damaged or re-connect it after formatting.	
Group-specific Error Codes	0x0100	Group No. xx Folder Creation Error	Format in FAT32 format and connect SD memory.	
	0x0200	Group No. xx File Open Error		
	0x0400	Group No. xx File Write Error		

11.11 Datalog Performance

11.11.1 Data Processing Time

This section describes the data storage time of datalog function.
The processing times described in this section do not represent absolute values, but actual measurement of each example.
The actual processing time varies depending on the scan time, volume of collected data, format of the collected data, type and storage of SD memory and number of files in the SD memory.

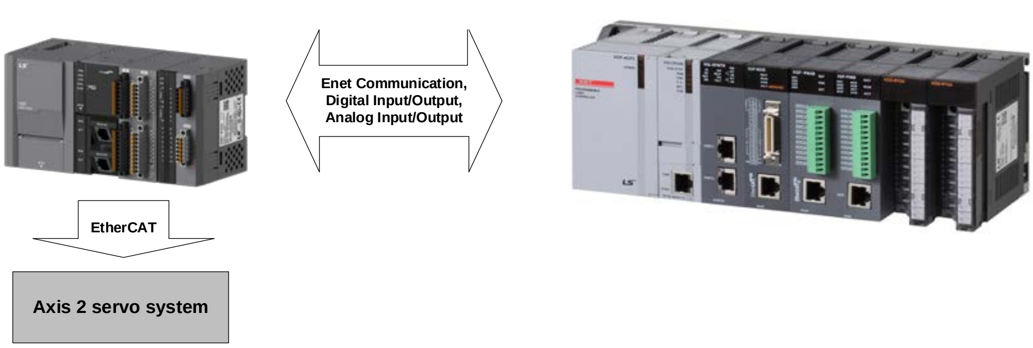
11.11.2 Save Performance by Main Task Interval

The following figures are save performance measurement by main task save intervals and number of set data saved. These measurements represent relative values. The actual vales may vary depending on the program, setting parameter and SD memory applied. You can use it as a reference when using datalog function.

(1) Set Condition
Data processing time was measured under the following conditions.

Item		Description	Note
Main task Interval		1ms, 2ms, 4ms	
Buffer Size		128kByte	
Data Collection Time		4 Word/ 10ms	
Datalog Setting	Sampling Method	Designated Main task Interval	
	Data	M Area, Type: Word	
	CSV Output	Time, Index information included	
	File Save	16MByte	
SD Memory Card		Transcend 16G	

(2) System configuration
The system for performance measurement was configured using the built-in function of motion controller.



(3) Measurement Results:
The storage performance according to the main task cyclic is shown in the following table.

(a) In case of WORD type

	Number of Devices				
	4 WORD (4 WORD * 1 Group)	8 WORD (8 WORD * 1 Group)	16 WORD (16 WORD * 1 Group)	32 WORD (32 WORD * 1 Group)	64 WORD (32 WORD * 2 Group)
1ms	Normal	Buffer overflow occurred	Buffer overflow occurred	Buffer overflow occurred	Buffer overflow occurred
2ms	Normal	Normal	Buffer overflow occurred	Buffer overflow occurred	Buffer overflow occurred
4ms	Normal	Normal	Normal	Normal	Buffer overflow occurred

(b) In case of LWORD type

	Number of Devices				
	4 LWORD (4 LWORD * 1 Group)	8 LWORD (8 LWORD * 1 Group)	16 LWORD (16 LWORD * 1 Group)	32 LWORD (32 LWORD * 1 Group)	64 LWORD (64 LWORD * 1 Group)
1ms	Normal	Buffer overflow occurred	Buffer overflow occurred	Buffer overflow occurred	Buffer overflow occurred
2ms	Normal	Normal	Buffer overflow occurred	Buffer overflow occurred	Buffer overflow occurred
4ms	Normal	Normal	Normal	Normal	Buffer overflow occurred

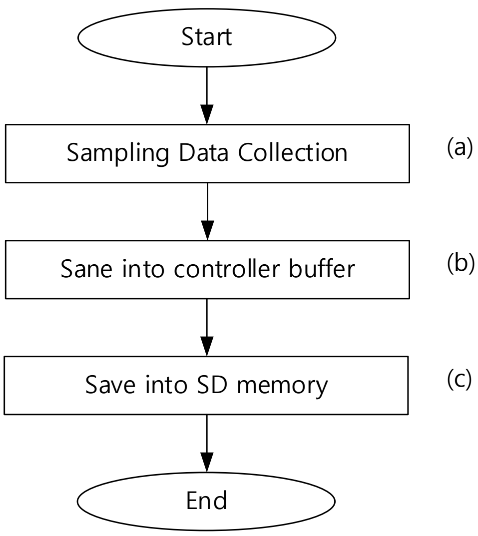
11.11.3 Save Process Time Verification

Datalog function does not guarantee saving of all data under any setting. It performs the maximum operation that motion controller is capable of at the time when datalog condition occurs. That is, since datalog processing time may fluctuate depending on the parameter setting, sampling data amount, scan time and run state of motion controller's other functions such as internal communication and position determination, it may not run as specified by the set collection condition in some cases. Therefore, it is recommended to use datalog function after verifying each processing time of the system before using datalog function.

(1) Save Process Time Verification

The following figure represents the flow from datalog function performed by motion controller to saving into SD memory.

Details are as follows.



Stage		Operation	Note
(a)	Data Collection	When datalog is started, data is collected and stored in the internal buffer. Data collection is performed in accordance with conditions set in the parameter (storage for each main task and specified cycle, etc.), data may not be collected in accordance with the set conditions depending on the number of data and function usage.	
(b)	Data Design	The collected data is designed and processed in a form that can be stored in the CSV file. After the design, it is stored in a buffer. If the data storage buffer is full, the designed data will wait until there is available free space in the buffer.	
(c)	Data Storage	It performs the operation of saving the designed data as a file in the SD card. If the data storage rate cannot keep up with the data collection rate, the internal buffer is exceeded, and data may be dropped.	

(2) Methods on how to check the data storage processing time
To confirm whether the collected data is stored normally in the SD card, check the following contents

Checklist	Contents and Solutions		Note
Buffer overflow flag	Contents	Check whether the number of times when the buffer overflow occurred in K area is 0. If it is not 0, data collection is faster than data collection time, and thus the data may not be stored. Insert ‘C” string in the saved file	
	Solutions	For the main task cycle, increase the main task cycle, and increase the sampling period in case of the specified cycle sampling. Reduce the amount of data collected per sampling. Only the necessary data is saved as a file (using the trigger storage function).	

Chapter 12 SD Additional Function

12.1 Overview

The motion controller has built-in additional functions using the SD card. This chapter describes the specifications and usage of the SD additional features.

12.1.1 Characteristics

Through the motion controller's SD additional features, you can perform the PLC update, backup, comparison, boot operation. These functions can be executed by operating the SD CMD buttons on the PLC.

(1) SD card setting function for SD additional features
- SD card setting through XG5000

(2) Motion controller update using the SD card
- Preventing leak of the motion controller program by using password setting
- Limit update using the motion controller's MAC address
- Motion controller's auto reset and operation mode can be set after updating
- Allows selective updating of projects using update flags

(3) Motion controller backup using the SD card
- Motion controller's program can be saved to the SD card without XG5000
- Motion controller's history also can be backed up.

(4) Comparison with the motion controller using the SD card
- You can compare the motion controller's parameters, motion data, motion controller programs, NC codes, CAM data.
- The comparison results can be saved and checked on the SD card.

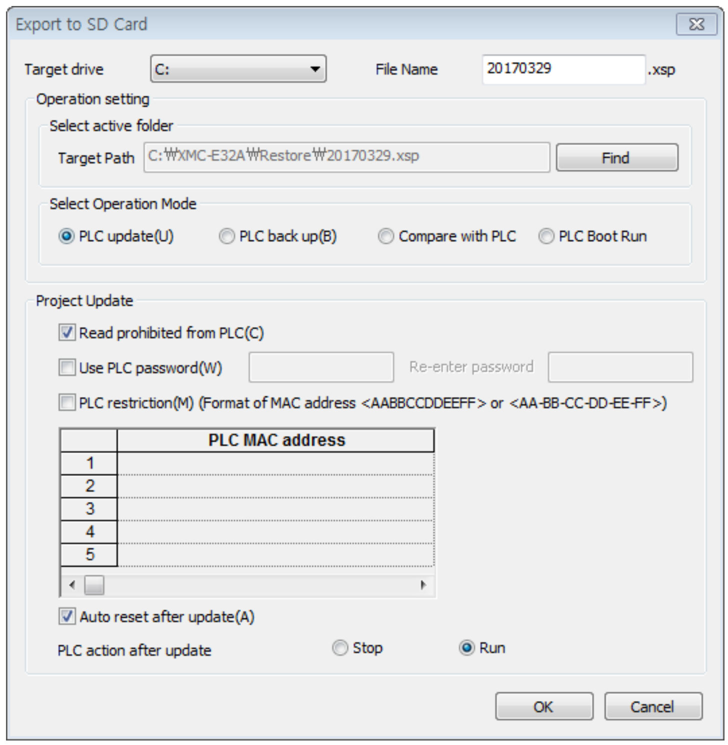
(5) Motion controller's boot operation using the SD card
- Motion controller program can be protected by password setting.
- Limiting the boot operation using the motion controller's MAC address

12.1.2 Export to the SD Card

Select [XG5000] - [Project] - [SD Card Setting] - [Export to SD Card] to launch the window where you can set the SD card. The function, 'Export to SD card' is available only when the XG5000 is not online.

(1) PLC Update

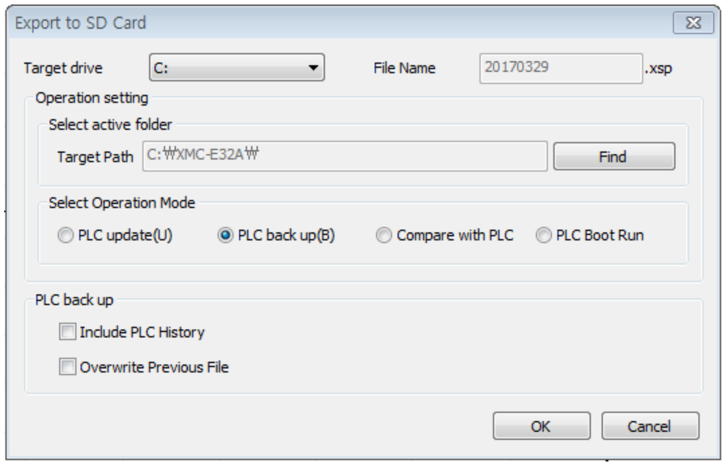
The PLC update function is to update the program stored in the SD card to the PLC.



The description of each item in the PLC update mode is as follows.

Classification	Content
Target drive	Select the storage medium to store the project data
File Name	Alphabet/Number only (However, when checking flag update, only numbers 0 to 255 are possible.)
Flag update	PLC update by selecting the desired project with the flag
Select action folder	Location where you will save the project data (folder)
Select operation mode	Set the PLC operations when inserting the SD card PlC Update: Updating the PLC using the data stored in the SD card
Do not read from PLC	When the PLC project is updated using the SD, 'Read from PLC' is prohibited.
Use PLC password	Check whether the PLC project can be updated using the SD including the PLC password setting
Limit PLC usage	Specify the PLC that can update the PLC project
Auto-reset after update is completed	Whether to execute the PLC reset after the PLC update is completed
PLC status after update is completed	Set the PLC operation mode after the PLC update is completed

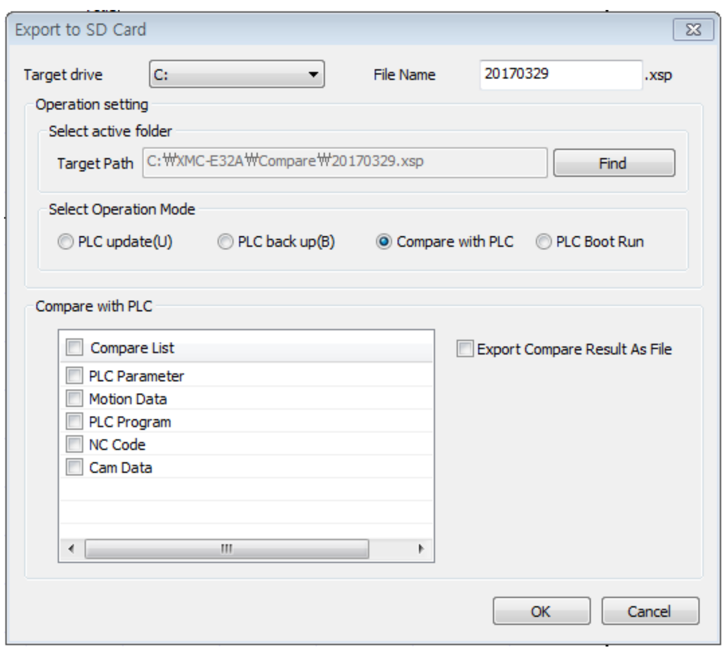
(2) PLC backup:
It is the function to back up the program stored in the PLC to the SD card.



In the PLC backup mode, the description of each item is as follows.

Classification	Content
Target drive	Select the storage medium to store the project data
Select action folder	Location where you will save the project data (folder)
Select operation mode	Set the PLC operations when inserting the SD card PLC backup: Saving the PLC project to the SD card
Include PLC history	Check whether to back up the history saved in the PLC together during the PLC backup
Overwrite existing file	If there is a file backed up to the SD card, checking whether to overwrite it.

(3) Comparison with PLC
This function allows you to compare the project stored in the PLC with the project stored in the SD card.

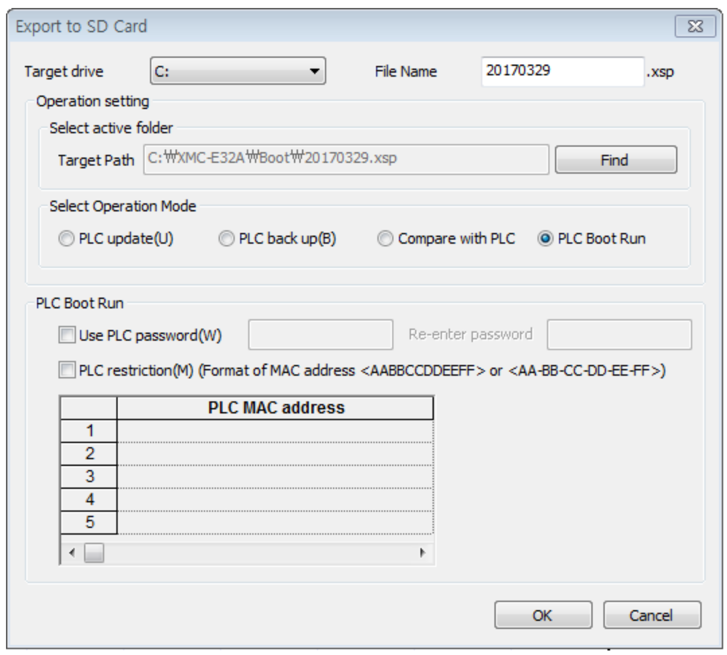


In the comparison mode with the PLC, the description of each item is as follows.

Classification	Content
Target drive	Select the storage medium to store the project data
Select action folder	Location where you will save the project data (folder)
Select operation mode	Set the PLC operations when inserting the SD card Compare PLC: Compare the projects stored in the PLC and SD card
Compare item	Only the desired items can be compared.
Save comparison results to a file	Check whether the comparison result is saved as csv type file.

(4) PLC boot operation

It is the function to start the PLC by using the project stored in the SD. If you turn Off, On the PLC after removing the SD, it runs by the program that was driven before boot operation.

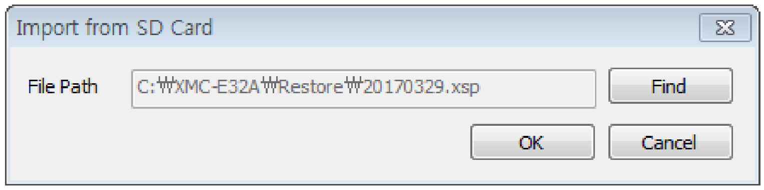


The description of each item in the PLC boot operation mode is as follows.

Classification	Content
Target drive	Select the storage medium to store the project data
Select action folder	Location where you will save the project data (folder)
Select operation mode	Set the PLC operations when inserting the SD card PLC Update: Updating the PLC using the data stored in the SD card
Use PLC password	Check whether the PLC boot operation can be updated using the SD
Limit PLC usage	Specify the PLC that can execute the PLC boot operation

12.1.3 Import from the SD Card

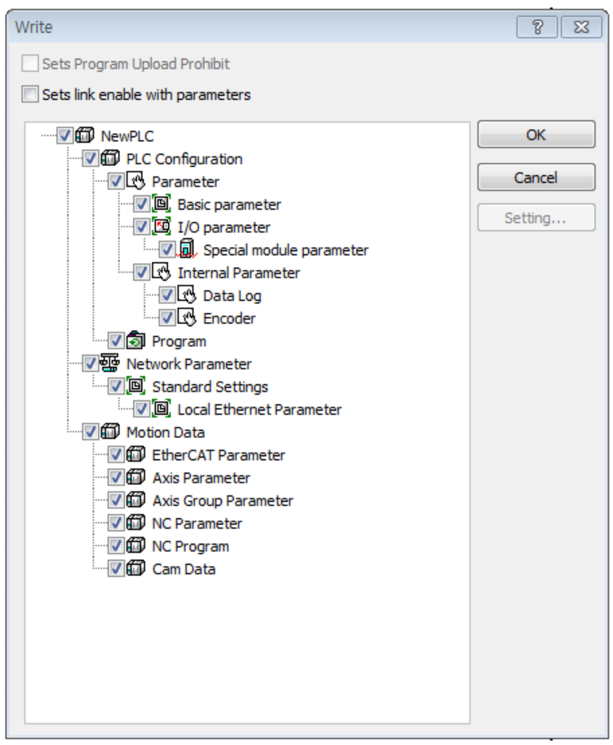
Select [XG5000] - [Project] - [SD card setting] - [Import from SD] to launch the window to read the file.
In the corresponding path, you can confirm the project saved in the SD of XG5000 is opened.



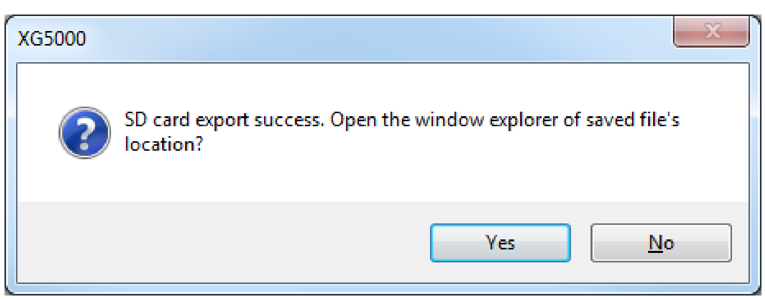
12.1.4 PLC update function

The PLC update function is available only when the PLC is in the STOP mode.

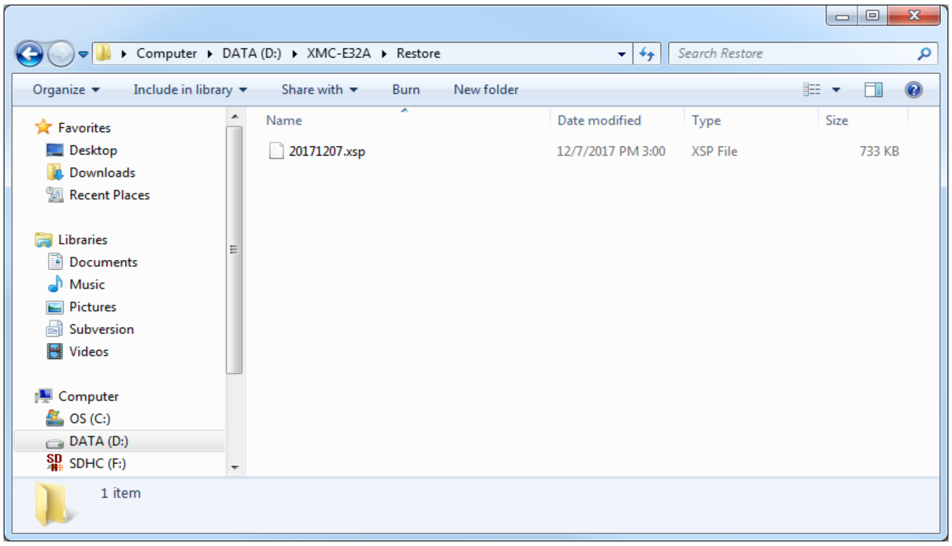
If you select 'PLC update' in 'Export to SD card' and click OK, the writing window will be created as below.



After completing 'Export to SD card' successfully, the window indicating successful completion is created and the saved drive is displayed.



When checking the saved drive, an add-on folder is created under the model folder and the file is created in the Confi g and Restore folder.



When the SD card is inserted into the SD card slot of the PLC, the flag (% KW541) of SD additional features is displayed according to the values set in the Config.

%KW541	SD additional functions mode
0	additional functions
1	PLC backup function
2	PLC update function
3	Comparison with PLC Function
4	Boot operation function

If you press the SD CMD button once for more than 0.7 second and less than 2 seconds, the flag (%KX8640) will be turned On and the PLC update operation will be executed while the SD RD/WR LED and SD additional features are running.

Besides the SD CMD button method, PLC update using flags is also possible. If the SD add-on function is PLC update, you can update any project stored in the SD card. However, when exporting to an SD card, the project name must be set from 0 to 255. In XG5000, input the project to be updated in the SD memory project number (%KB1088) flag, and when the SD memory project update permission (%KX8688) flag is turned on, PLC update operation is performed.

When the update is completed normally, the flag (%KX8640) is turned Off and the completion flag of SD additional features (%KX8642) is turned On while the SD RD/WR LED and SD additional features are running. If an error occurs during operation, the SD additional function error flag (%KX8641) is turned On and the error value is displayed in the SD additional function error code (%KW542).

(1) When 'Read-protected from PLC' is set, even if the update process is normal, reading from the PLC via XG5000 is prohibited. If the PLC password option is set and the password has not been set in the PLC, the password will be saved to the PLC along with the project update. In addition, if the password is already set in the PLC before perfor ming the project update using the SD, the update will be executed only when the password set as the option matc hes the password of the PLC.

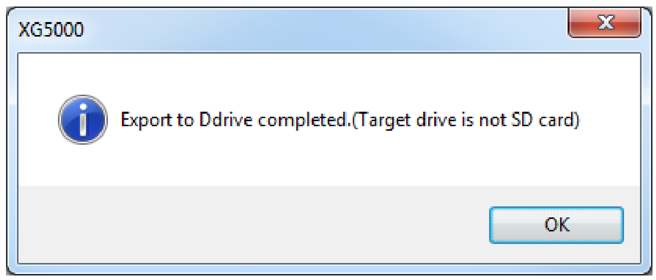
(2) If 'Limit PLC usage' option is set, the PLC update is performed only when the MAC address stored as the option ma tches the MAC address of the PLC.

(3) If auto-reset is set after update is completed, the PLC will be reset automatically after updating is done, and the PLC operation mode will be changed into the PLC status option specified value.

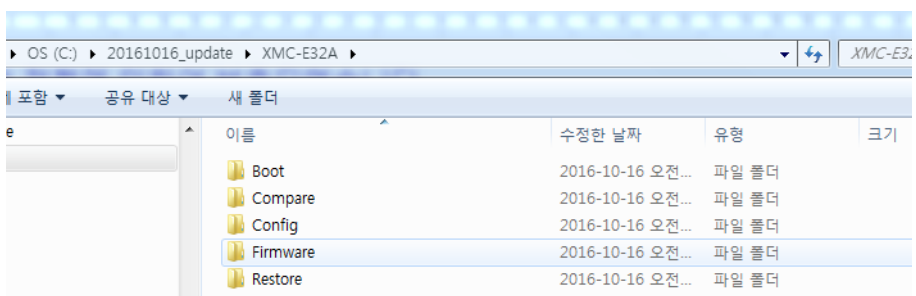
12.1.5 PLC backup function

This function backs up the project stored in the PLC to the SD card. The project of the PLC is backed up in the Backu p folder in the MAC address folder of the product and saved as a file. The PLC backup function can operate regardless of the PLC mode.

After completing 'Export to SD card' successfully, the window indicating successful completion is created and the saved drive is displayed.



When checking the saved drive, an add-on folder is created under the model folder and the file is created in the Config folder.



When the SD card is inserted into the SD card slot of the PLC, the flag (% KW541) of SD additional features is displayed according to the values set in the Config.

%KW541	SD additional functions mode
0	additional functions
1	PLC backup function
2	PLC update function
3	Comparison with PLC Function
4	Boot operation function

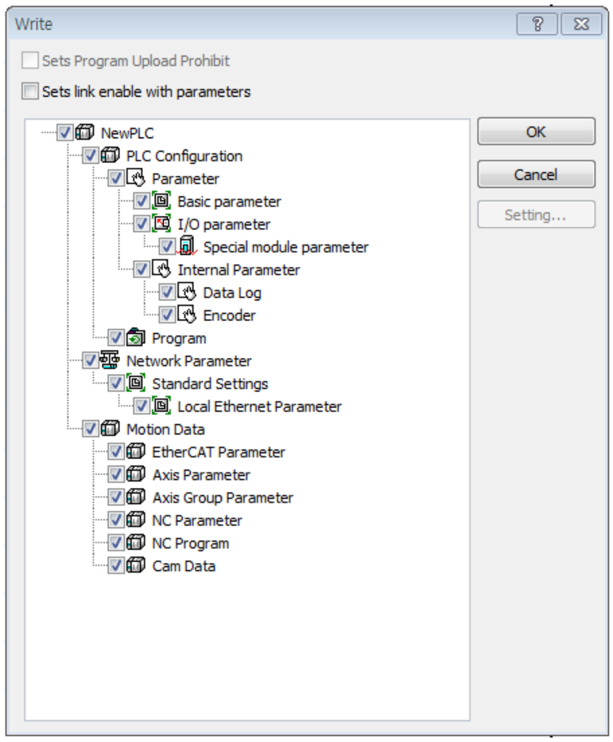
If you press the SD CMD button once for more than 0.7 second and less than 2 seconds, the flag (%KX8640) will be turned On and the PLC update operation will be executed while the SD RD/WR LED and SD additional features are running. When the backup is completed normally, the flag (%KX8640) is turned Off and the completion flag of SD additional features (%KX8642) is turned On while the SD RD/WR LED and SD additional features are running.

If an error occurs during operation, the SD additional function error flag (%KX8641) is turned On and the error value is displayed in the SD additional function error code (%KW542).

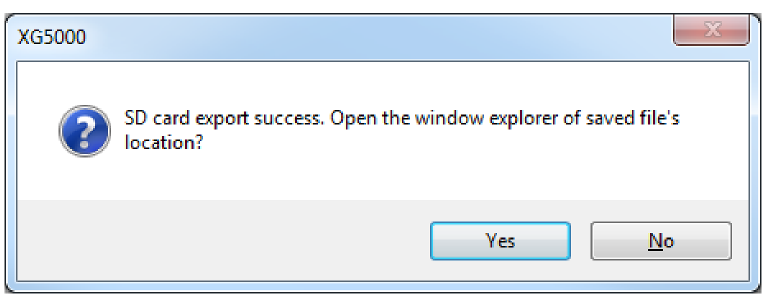
After removing the SD card, you can see the project will be saved under the Backup folder in the product's correspondin g path and the saved project will be opened when executing 'Import from SD' in XG5000.

12.1.6 Comparison with PLC Function

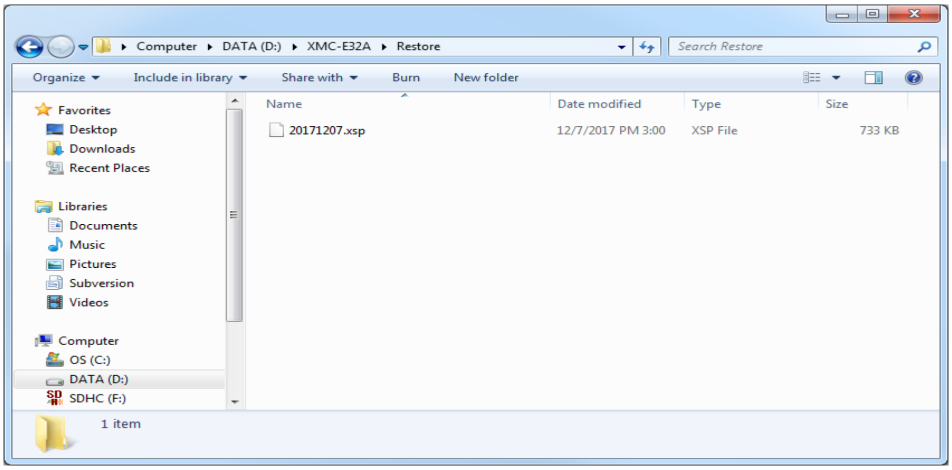
This function is used to compare the project stored in the PLC with the program stored in the SD card. The comparison result can be checked through the flag or .csv file. If you select 'Comparison with the PLC' in 'Export to SD card' and click OK, the writing window will be created as below.



After completing 'Export to SD card' successfully, the window indicating successful completion is created and the saved drive is displayed.



When checking the saved drive, an add-on folder is created under the model folder and the file is created in the Config, Compare folder.



When the SD card is inserted into the SD card slot of the PLC, the flag (% KW541) of SD additional features is displayed according to the values set in the Config.

.

%KW541	SD additional functions mode
0	additional functions
1	PLC backup function
2	PLC update function
3	Comparison with PLC Function
4	Boot operation function

If you press the SD CMD button once for more than 0.7 second and less than 2 seconds, the flag (%KX8640) will be turned On and the PLC update operation will be executed while the SD RD/WR LED and SD additional features are running.
When the comparison is completed normally, the flag (%KX8640) is turned Off and the completion flag of SD additional features (%KX8642) is turned On while the SD RD/WR LED and SD additional features are running.
If an error occurs during operation, the SD additional function error flag (%KX8641) is turned On and the error value is displayed in the SD additional function error code (%KW542).

(1) (1) When there is a discrepancy of the comparison, the SD comparison result flag (% KX8643) is turned Off and if it is the same, the SD comparison flag is turned On.

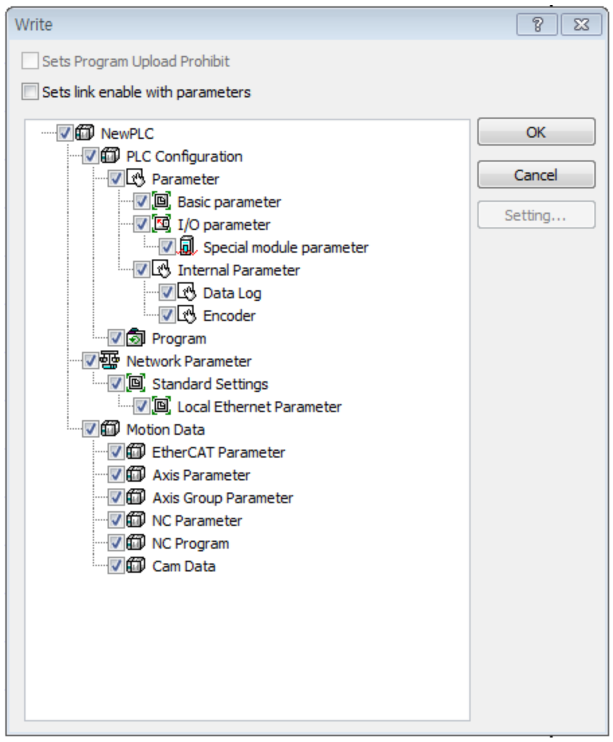
(2) If you check the item, 'Save comparison result to file' when executing ‘Export to SD card’, the result file (CmpResult.c

sv) is created in the ‘Compare’ folder and the comparison result is saved.

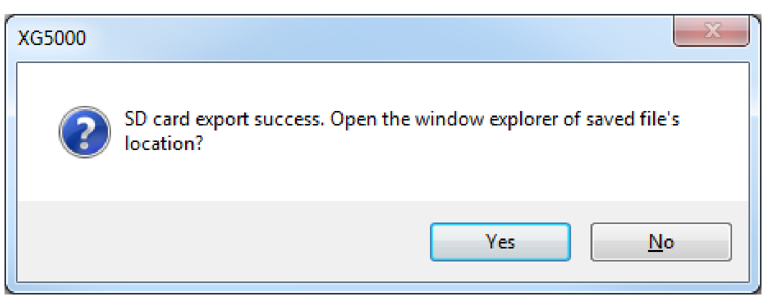
12.1.7 PLC Boot Operation

This function is to operate the PLC with the program saved in the SD, not the project saved in the PLC. The programs that were already running are stored in the PLC. If the PLC power is turned off, on after removing the SD card, it is driven by the existing program again.

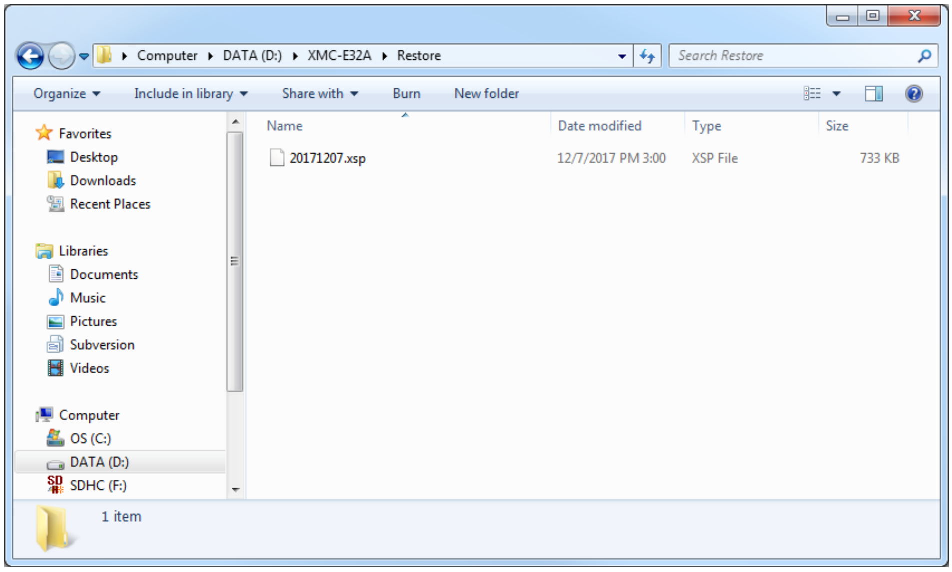
If you select 'PLC boot operation' in 'Export to SD card' and click OK, the writing window will be created as below.



After completing 'Export to SD card' successfully, the window indicating successful completion is created and the saved drive is displayed.



When checking the saved drive, an add-on folder is created under the model folder and the file is created in the Config, Boot folder.



The boot operation must be performed when the PLC is powered off. After installing the SD card in the PLC power off state, turn on the PLC power while pressing the SD CMD button.

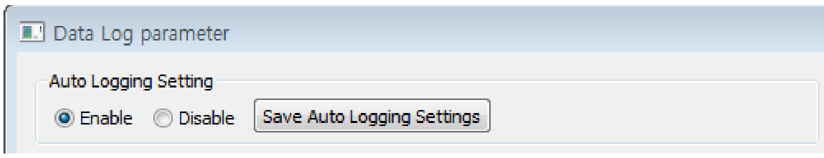
When the boot operation mode is completed normally, the flag (%KX8640) is turned Off and the completion flag of SD additional features (%KX8642) is turned On while the SD RD/WR LED and SD additional features are running.

If an error occurs during operation, the SD additional function error flag (%KX8641) is turned On and the error value is displayed in the SD additional function error code (%KW542).

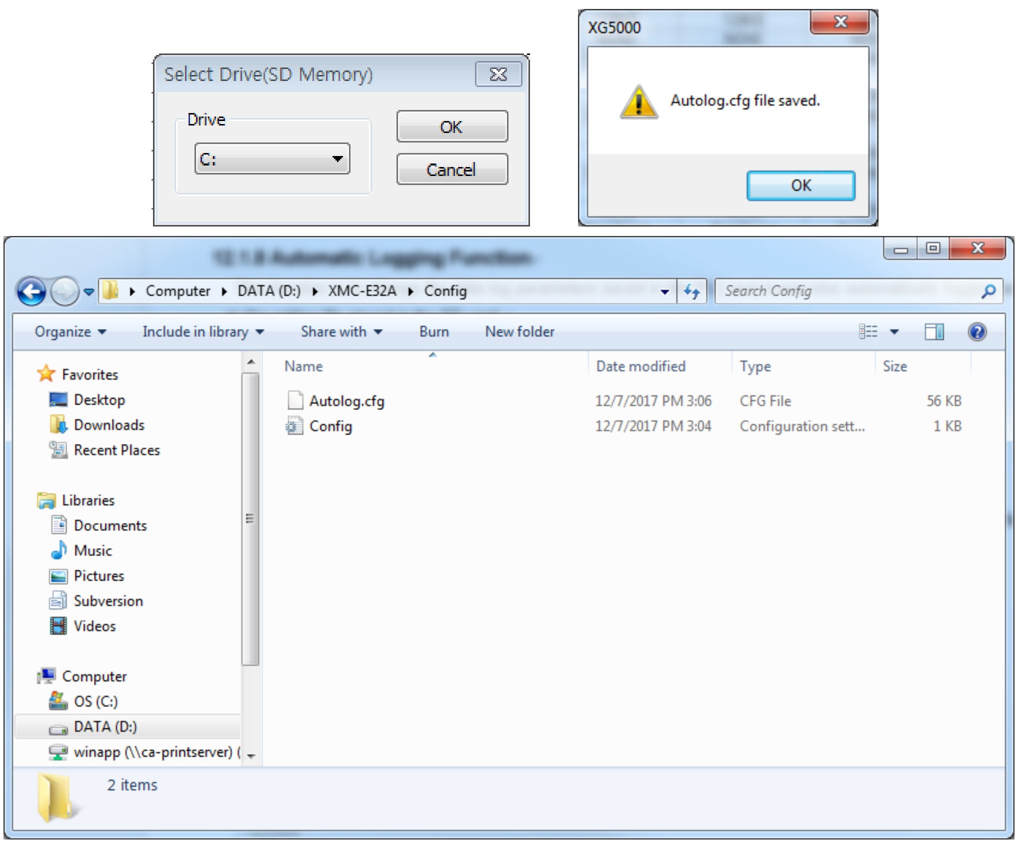
12.1.8 Automatic Logging Function

This function is to change the data log parameters saved in the PLC and it executes automatically logging according to the setting file stored in the SD card.

There are the items for automatic logging setting on the top left of the data log parameters window.



If you save the auto logging settings after setting the parameter related to data log, you can save the setting file to the desired path.



Auto-logging is executed when the power is turned On after the SD card is mounted while the PLC is off.

(1) If the parameters stored in the PLC are set to 'Prohibit automatic logging', the settings of Autolog.cfg will not be reflected but will operate according to the existing program settings.
(2) Since the corresponding parameters stored in the SD are saved in the PLC, if other SD card is inserted and the PLC power is turned off or on, same operations will be executed as auto logging settings.
(3) If the automatic logging stop (_DL_AutoLogStop: %KX8192) command flag is turned on during automatic logging,After that, logging is not resumed even if the group data logging enable flag is turned on or the automatic logging stop command flag is turned off.

12.1.9 Error Codes and Countermeasures

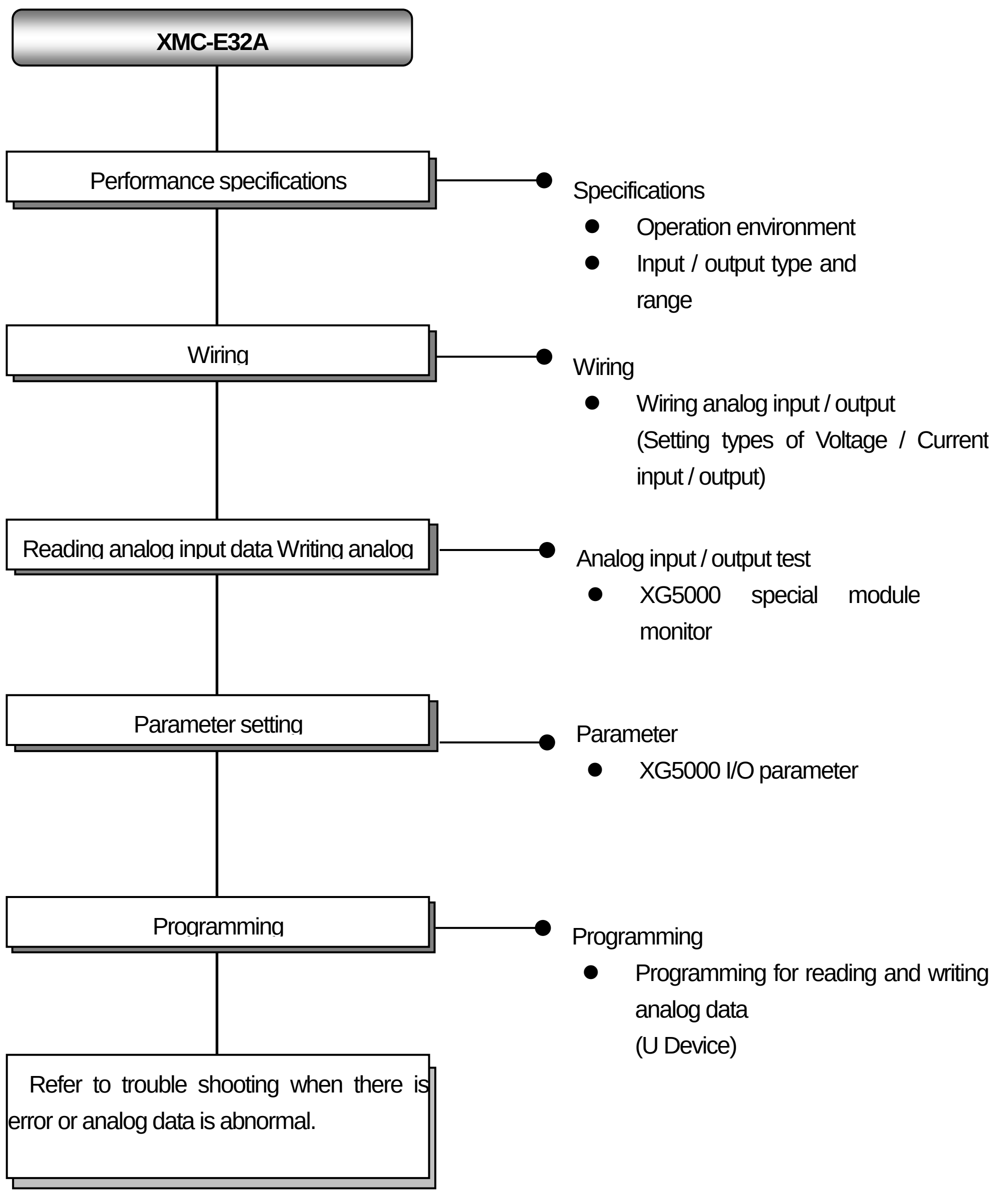
The error codes related to SD additional features are as follows. The error code is displayed together with the additional function mode. (For example, when there is no file password among PLC update functions (0x2005) 2: additional function mode, 5: error operation)

Category	Error code	Error name	Remarks
Whole error codes	0xX001	File error (file open failure, CRC error)	
	0xX002	Damaged file (damaged head, tail, etc.)	
	0xX003	Unsupported file version	
	0xX004	Model mismatch	
	0xX005	No password in file	
	0xX006	Password discrepancy	
	0xX007	MAC address mismatch	
	0xX008	File decryption error	
	0xX009	IO configuration mismatch	
	0xX00A	No save file	
	0xX00B	PLC mode is RUN status	
	0xX00C	No SD card	
	0xX00D	SD card error status	
	0xX00E	In the process of powering off the SD card	
	0xX00F	State that the SD card is powered off	

Chapter 13 Built-in Analog Functions

13.1 Overview

Before using the analog input and output function, follow steps below.



Notes

The analog function is supported only by analog-type products (XMC-E32A/E16A/E08A).

Performance specifications are as follows

1) Input Performance Specifications

<table>
<tr><th colspan="3">Item</th><th colspan="2">Performance specifications</th></tr>
<tr><td colspan="3">Number of channels</td><td colspan="2">2 channels</td></tr>
<tr><td rowspan="7">Analog
Input range</td><td colspan="2">Type</td><td>Voltage</td><td>Current</td></tr>
<tr><td colspan="2" rowspan="2">Range</td><td>DC 1~5V
DC 0~5V
DC 0~10V
DC -10~10V
(input resistance: 1 ㏀ or more)</td><td>DC 4~20㎃
DC 0~20㎃
(input resistance:250Ω)</td></tr>
<tr><td colspan="2">Current input or Voltage input can be selected through the external terminal wiring setting. (In voltage mode, use V+ and COM terminal for the channel. • In current mode, short V+ and I+ terminal and then use I+ and COM terminal.)</td></tr>
<tr><td rowspan="4">R
a
n
g
e</td><td>Unsigned Value</td><td colspan="2">0~16,000</td></tr>
<tr><td>Signed Value</td><td colspan="2">-8,000~8,000</td></tr>
<tr><td>Precise value</td><td>1,000~5,000(1~5V)
0~5,000(0~5V)
0~10,000(0~10V)
-10,000~10,000(±10V)</td><td>4,000~20,000(4~20㎃)
0~20,000(0~20㎃)</td></tr>
<tr><td>Percentile value</td><td colspan="2">0~10,000</td></tr>
<tr><td colspan="3" rowspan="2">Max. resolution</td><td colspan="2">1/16,000</td></tr>
<tr><td>0.250㎷(1~5V)
0.3125㎷(0~5V)
0.625㎷(0~10V)
1.250㎷(±10V)</td><td>1.0㎂(4~20㎃)
1.25㎂(0~20㎃)</td></tr>
<tr><td colspan="3">Precision</td><td colspan="2">±0.2% or less (When ambient temperature 25±5℃)
±0.3% or less (When ambient temperature 0±55℃)</td></tr>
<tr><td colspan="3">Max. conversion speed</td><td colspan="2">0.5ms/channel</td></tr>
<tr><td colspan="3">Absolute Max. input</td><td>DC ±15V</td><td>DC ±30㎃</td></tr>
<tr><td rowspan="8">Auxiliary function</td><td colspan="2">Filter</td><td colspan="2">Digital filter (4 to 64,000 ㎳)</td></tr>
<tr><td colspan="2" rowspan="4">Average function</td><td colspan="2">Time average (4 to 16,000 ㎳)</td></tr>
<tr><td colspan="2">Count average (2 to 64,000)</td></tr>
<tr><td colspan="2">Moving average (2 to 100)</td></tr>
<tr><td colspan="2">Weighted average (1 to 99%)</td></tr>
<tr><td colspan="2">Alarm function</td><td colspan="2">Disconnection (1 to 5 V DC, 4 to 20 ㎃ DC)</td></tr>
<tr><td colspan="2">Hold Last Value Function</td><td colspan="2">When input signal exceeds the effective range, holds the last effective value.</td></tr>
<tr><td colspan="2">Alarm Function</td><td colspan="2">When input signal exceeds the effective range, relevant flag turns on.</td></tr>
<tr><td colspan="3">Input terminal</td><td colspan="2">6 point terminal block</td></tr>
</table>

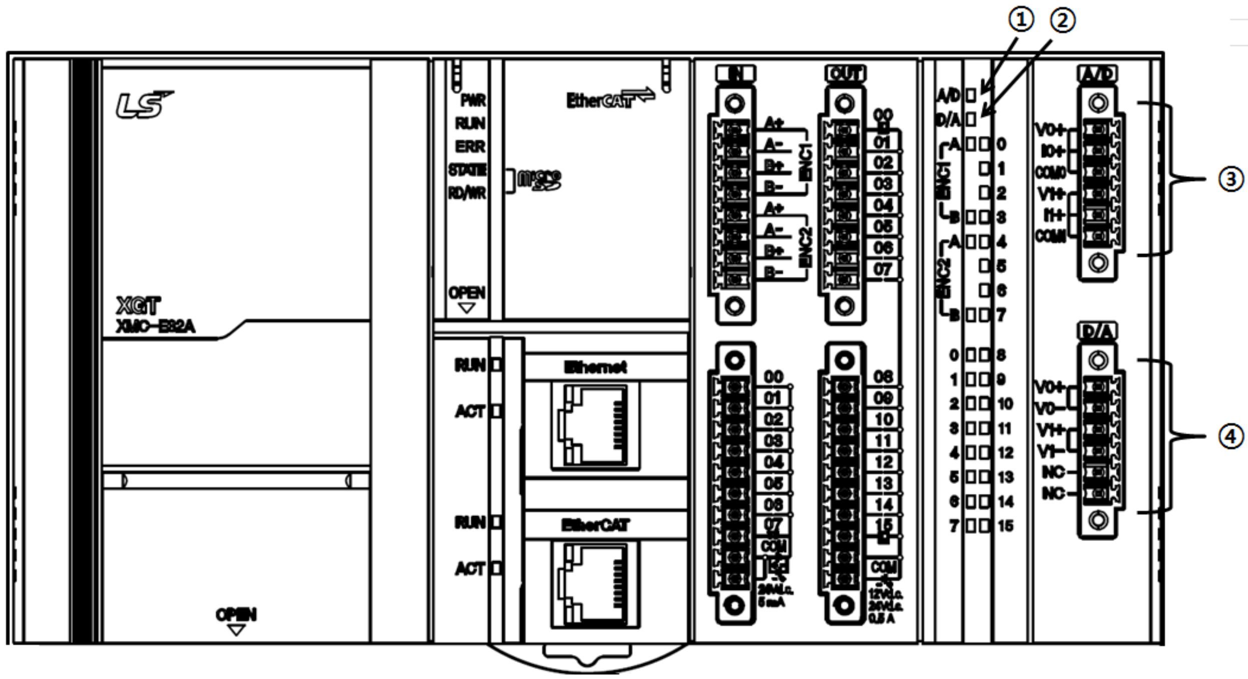
2) Output Performance Specifications

Item			Performance specifications
number of output channels			voltage 2 channels
	Range		DC 1~5V DC 0~5V DC 0~10V DC -10~10V (Load resistance: 1 ㏀ or more)
			Output ranges are set in user program or I/O parameter per each channel.
	Range	Unsigned Value	0~16,000
		Signed Value	-8,000~8,000
		Precise value	1,000~5,000(1~5V) 0~5,000(0~5V) 0~10,000(0~10V) -10,000~10,000(±10V)
		Percentile value	0~10,000
Max. resolution			1/16,000
			0.250㎷(1~5V) 0.3125㎷(0~5V) 0.625㎷(0~10V) 1.250㎷(±10V)
Precision			±0.2% or less (When ambient temperature is 25±5℃) ±0.3% or less (When ambient temperature 0~55℃)
Max. conversion speed			0.5ms/channel
Additional functions			Setting of channel output status - Select one among previous, minimum, average, maximum value Setting of interpolation method - Linear interpolation, S-type interpolation
Output terminal			6 point terminal block

3) Common performance specification

Item	Performance specifications
Insulation method	Photo-coupler and trans insulation between the input/output terminal and motion controller power (no insulation between channels)
I/O occupied points	Fixed type: 64 point

13.2 Name of Analog Part and Functions



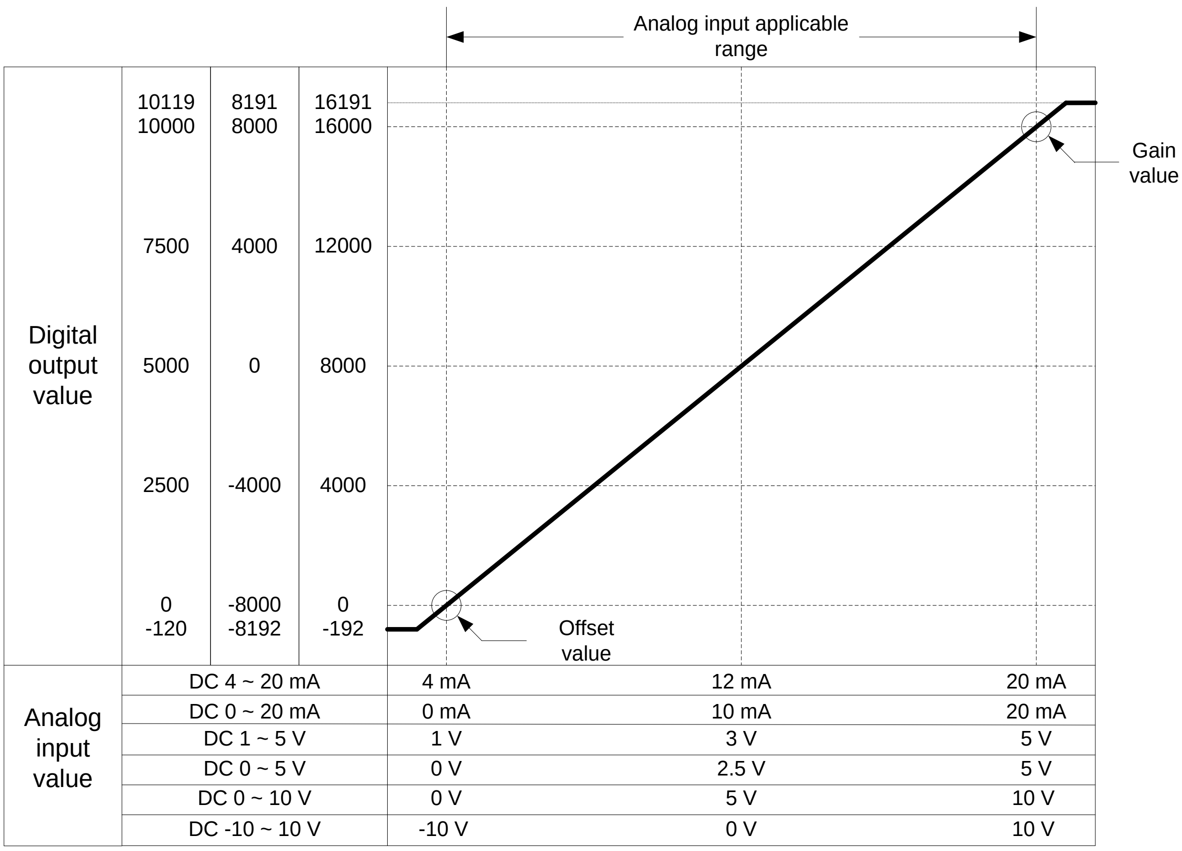
No.	Name	Content
①	A/D LED	Displays the operation status of analog input part On: Normal operation Blinks: Error occurs (Flickering 1s intervals) Off: Power off or module error
②	D/A LED	Displays the operation status of analog output part On: Normal operation Blinks: Error occurs (Flickering 1s intervals) Off: Power off or module error
③	Input connector	▶ Wiring input terminal block to connect with external device
④	Output connector	▶ Wiring output terminal block to connect with external device

13.3 Characteristic of I/O Control

Voltage/Current input ranges are able to set from each channel by using user program or I/O parameter.
Data output type of digital is defined as below.

1) Unsigned Value
2) Unsigned Value
3) Precise value
4) Percentile Value

13.3.1 Input characteristic



1) DC 4~20mA input range

Digital output range	Analog input current (㎃)						
	3.808	4	8	12	16	20	20.191
Unsigned Value (-192~16,191)	-192	0	4,000	8,000	12,000	16,000	16,191
Signed Value (-8,192~8,191)	-8,192	-8,000	-4,000	0	4,000	8,000	8,191
Precise value (3,808~20,191)	3,808	4,000	8,000	12,000	16,000	20,000	20,191
Percentile value (-120~10,119)	-120	0	2,500	5,000	7,500	10,000	10,119

2) DC 0~20mA input range

Digital output range	Analog input current (㎃)						
	-0.24	**0**	**5**	**10**	**15**	**20**	**20.239**
Unsigned Value (-192~16,191)	-192	0	4,000	8,000	12,000	16,000	16,191
Signed Value (-8,192~8,191)	-8,192	-8,000	-4,000	0	4,000	8,000	8,191
Precise value (-240~20,239)	-240	0	5,000	10,000	15,000	20,000	20,239
Percentile value (-120~10,119)	-120	0	2,500	5,000	7,500	10,000	10,119

3) DC 1 ~ 5V input range

Digital output range	Analog input voltage (V)						
	0.952	**1**	**2**	**3**	**4**	**5**	**5.047**
Unsigned Value (-192~16,191)	-192	0	4,000	8,000	12,000	16,000	16,191
Signed Value (-8,192~8,191)	-8,192	-8,000	-4,000	0	4,000	8,000	8,191
Precise value (952~5,047)	952	1,000	2,000	3,000	4,000	5,000	5,047
Percentile value (-120~10,119)	-120	0	2,500	5,000	7,500	10,000	10,119

4) DC 0 ~ 5V input range

Digital output range	Analog input voltage (V)						
	-0.06	**0**	**1.25**	**2.5**	**3.75**	**5**	**5.059**
Unsigned Value (-192~16,191)	-192	0	4,000	8,000	12,000	16,000	16,191
Signed Value (-8,192~8,191)	-8,192	-8,000	-4,000	0	4,000	8,000	8,191
Precise value (-60~5,059)	-60	0	1,250	2,500	3,750	5,000	5,059
Percentile value (-120~10,119)	-120	0	2,500	5,000	7,500	10,000	10,119

5) DC 0 ~ 10V input range

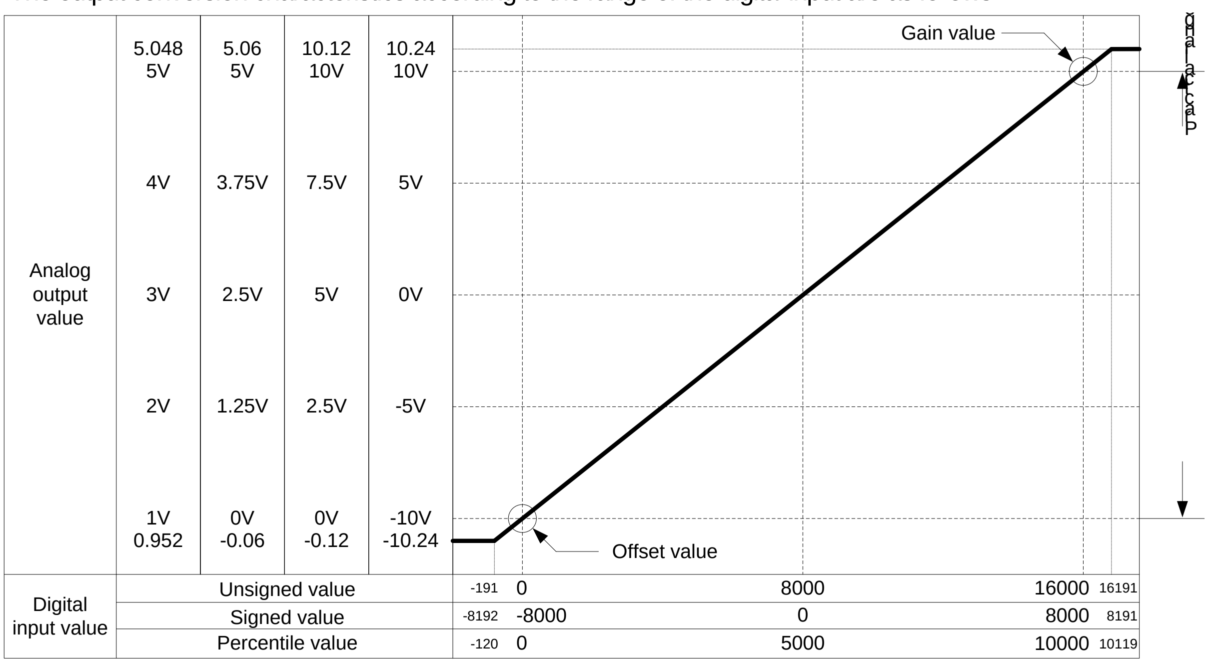
Digital output range	Analog input voltage (V)						
	-0.12	**0**	**2.5**	**5**	**7.5**	**10**	**10.119**
Unsigned Value (-192~16,191)	-192	0	4,000	8,000	12,000	16,000	16,191
Signed Value (-8,192~8,191)	-8,192	-8,000	-4,000	0	4,000	8,000	8,191
Precise value (-120~10,119)	-120	0	2,500	5,000	7,500	10,000	10,119
Percentile value (-120~10,119)	-120	0	2,500	5,000	7,500	10,000	10,119

6) DC -10 ~ 10V input range

Digital output range	Analog input voltage (V)						
	-10.24	**-10**	**-5**	**0**	**5**	**10**	**10.239**
Unsigned Value (-192~16,191)	-192	0	4,000	8,000	12,000	16,000	16,191
Signed Value (-8,192~8,191)	-8,192	-8,000	-4,000	0	4,000	8,000	8,191
Precise value (-10,240~10,239)	-10,240	-10,000	-5,000	0	5,000	10,000	10,239
Percentile value (-120~10,119)	-120	0	2,500	5,000	7,500	10,000	10,119

13.3.2 Output characteristic

The output conversion characteristics according to the range of the digital input are as follows.



1) DC 1 ~ 5V Output range

Digital Input	Analog Output voltage (V)						
	0.952	**1**	**2**	**3**	**4**	**5**	**5.047**
Unsigned Value (-192~16,191)	-192	0	4,000	8,000	12,000	16,000	16,191
Signed Value (-8,192~8,191)	-8,192	-8,000	-4,000	0	4,000	8,000	8,191
Precise value (952~5,047)	952	1,000	2,000	3,000	4,000	5,000	5,047
Percentile value (-120~10,119)	-120	0	2,500	5,000	7,500	10,000	10,119

2) DC 0 ~ 5V output range

Digital Input	Analog Output voltage (V)						
	-0.06	**0**	**1.25**	**2.5**	**3.75**	**5**	**5.059**
Unsigned Value (-192~16,191)	-192	0	4,000	8,000	12,000	16,000	16,191
Signed Value (-8,192~8,191)	-8,192	-8,000	-4,000	0	4,000	8,000	8,191
Precise value (-60~5,059)	-60	0	1,250	2,500	3,750	5,000	5,059
Percentile value (-120~10,119)	-120	0	2,500	5,000	7,500	10,000	10,119

3) DC 0 ~ 10V Output range

Digital Input	Analog Output voltage (V)						
	-0.12	**0**	**2.5**	**5**	**7.5**	**10**	**10.119**
Unsigned Value (-192~16,191)	-192	0	4,000	8,000	12,000	16,000	16,191
Signed Value (-8,192~8,191)	-8,192	-8,000	-4,000	0	4,000	8,000	8,191
Precise value (-120~10,119)	-120	0	2,500	5,000	7,500	10,000	10,119
Percentile value (-120~10,119)	-120	0	2,500	5,000	7,500	10,000	10,119

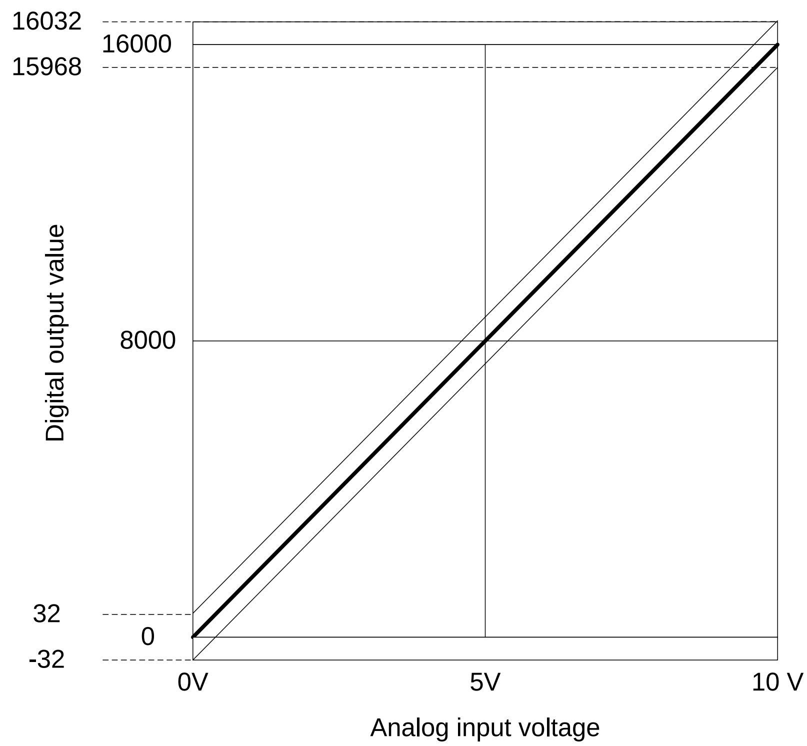
4) DC -10 ~ 10V Output range

Digital Input	Analog Output voltage (V)						
	-10.24	**-10**	**-5**	**0**	**5**	**10**	**10.239**
Unsigned Value (-192~16,191)	-192	0	4,000	8,000	12,000	16,000	16,191
Signed Value (-8,192~8,191)	-8,192	-8,000	-4,000	0	4,000	8,000	8,191
Precise value (-10,240~10,239)	-10,240	-10,000	-5,000	0	5,000	10,000	10,239
Percentile value (-120~10,119)	-120	0	2,500	5,000	7,500	10,000	10,119

13.4 Precision

13.4.1 Input Accuracy

Accuracy of digital output value does not change even if input range is changed.
Figure below shows the range of the accuracy with analog input range of 0 to 10 V and digital output type of unsigned value selected.
Accuracy is ±0.2% (Ambient temperature of 25±5℃).

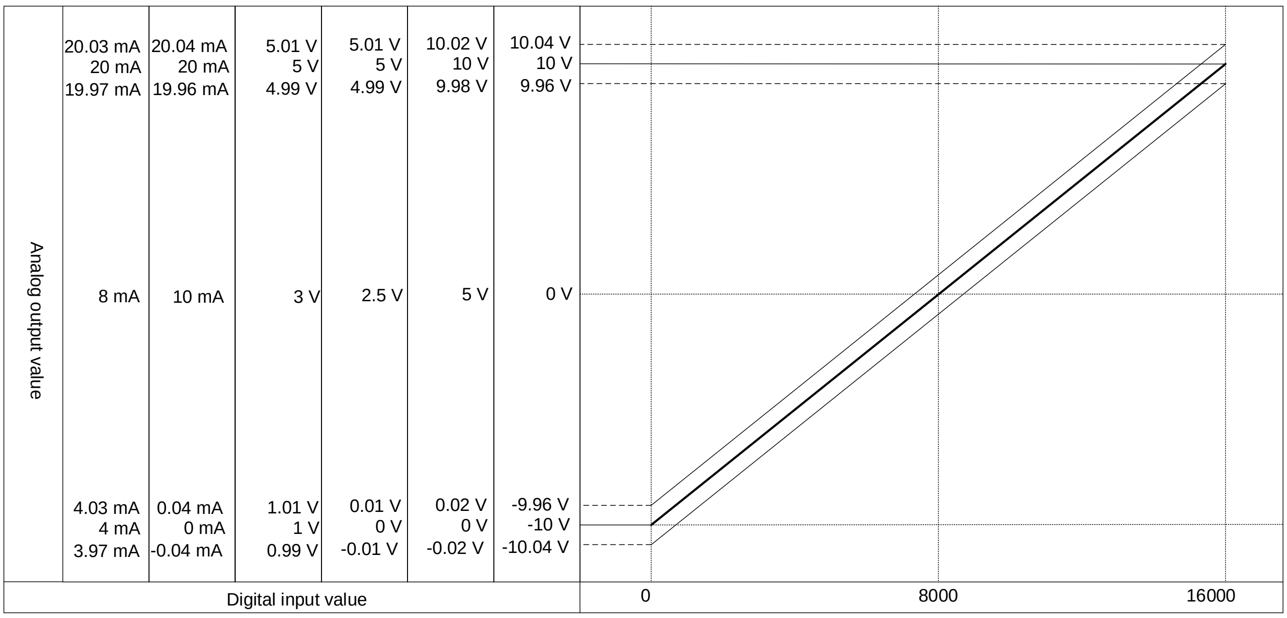


1) Accuracy when using 5 V input 16,000 × 0.2% = 32
   Therefore the range of the accuracy will become (8,000-32) to (8,000+32) = 7,968 to 8,032

2) Accuracy when using 10V input 16,000 × 0.2% = 32
   Therefore the range of the accuracy will become (16,000-32) to (16,000+32) = 15,968 to 16,032

13.4.2 Output Accuracy

Accuracy of digital output value does not change even if input range is changed.
When digital input range is selected with unsigned value, It indicates the range of precision variation.
Accuracy is ±0.2% (Ambient temperature of 25 ±5℃).



1) Accuracy when using -10 to 10V output 16000 × 0.2% = 32
   Accuracy range when using -10 V output will become (-10 V – 32 × 1.25 ㎷) ~ (-10 V + 32 × 1.25 ㎷) = -10.04 ~ -9.96 V, Accuracy range when using 10V output will become (10 V – 32 × 1.25 ㎷) ~ (10 V + 32 × 1.25 ㎷) = 9.96 ~ 10.04 V

2) Accuracy when using 1 to 5 V output 16000 × 0.2% = 32
   Accuracy range when using 1 V output will become (1 V – 32 × 0.25 ㎷) ~ (1 V + 32 × 0.25 ㎷) = 0.992 V ~ 1.008 V, Accuracy range when using 10V output will become (5 V – 32 × 0.25 ㎷) ~ (5 V + 32 × 0.25 ㎷) = 4.992 V ~ 5.008 V

13.5 Built-in Analog functions

Functions of embedded analog module are as described below.

Function item	Content
Channel run/stop setting	● Specify Run/Stop of the channel to execute A/D, D/A conversion. ● If the unused channel is set to Stop, whole Run time can be reduced.
Input / output voltage/current Range setting	● Specify analog input / output range to be used. ● Select range in parameter setting after selecting Voltage/Current input / output according to the wiring properly. ● Embedded analog module provides two kinds of current input / output ranges (4 to 20 mA, 0 to 20 mA) and four kinds of voltage input / output ranges (1 to 5 V, 0 to 5 V, 0 to 10 V,-10 to 10 V)
Input/Output data type Setting	● Specify digital input / output type. ● 4 data formats are provided in this module. (Unsigned value, Signed value, Precise value, Percentile value)
A/D conversion methods	● Sampling processing - Sampling process will be performed if A/D conversion type is not specified. ● Filter processing - Used to delay the sudden change of input value. ● Average processing - Outputs average A/D conversion value based on time or count. ● Detection alarm (Input disconnection) - After detecting disconnection of the input circuit, the alarm is displayed by a single flag. (Input signal range: 4~20㎃, 1~5V) ● Maintenance function of valid conversion value. - When valid conversion value is exceeded, whether conversion value retains will be able to set. ● Alarm function - When exceeding valid input range, alarm and maximum /minimum flag will be generated.
D/A output status setting	● Set the output status of channel when changing 'Run' to 'Stop'. ● The four kinds of output statuses **are provided** (Previous, Min, Mid, Max value)
Interpolation method setting	● Set linear interpolation, S-type interpolation method.

13.5.1 Sampling Processing

Collects analog input sign through general A/D conversion processing at a specific interval to convert to digital.
The time required for A/D conversion of analog input sign till saved on the memory depends on the number of channels used.

(Processing time) = (Number of channels used) x (Conversion speed)

(i.e.) If the number of channels used is 3, its process time will be 3 x 0.5 ㎳ = 1.5 ㎳

Sampling is used to calculate the sampling value of continuous analog sign at a specific interval.

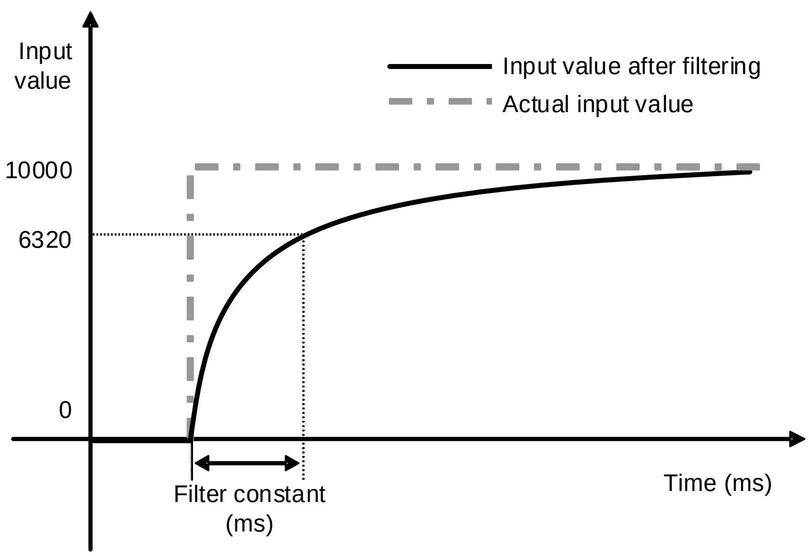
13.5.2 Filter Processing

The input value of the specified channel is calculated and output as follows with the previously filtered input value by the set filter value (time constant 63.2%).

$$\text{Filtered Value} = \frac{(\text{Pre} - \text{Filtered Input Value} \times \text{Filter Constant}) + (\text{Current Input Value} \times 0.5\ ms \times \text{Number of used channels})}{\text{Filter Constant} + (0.5\ ms \times \text{Number of used channels})}$$

The time required for A/D conversion of analog input sign till saved on the memory depends on the number of channels used.

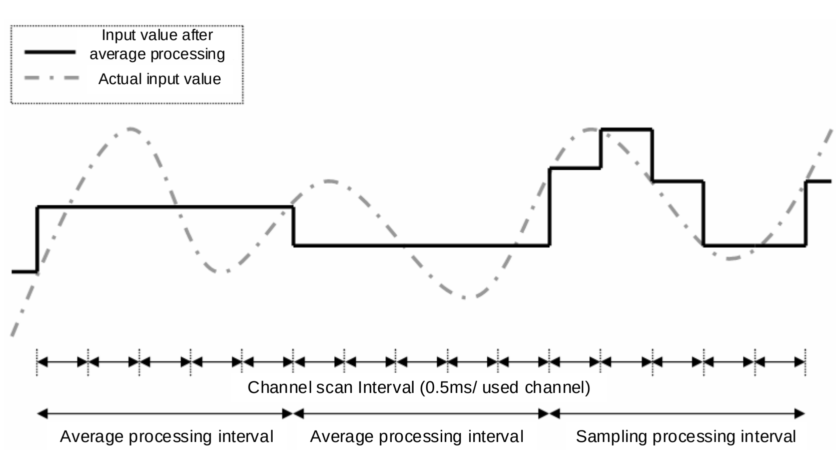
Setting range of Filter constant = 4 to 64,000 [ms]



As the above graph, if the input value rapidly decreases from 0 to 10,000, the input value will be filtered.
Specified time with filter constant is that the input value is the time to change by 63.2% of actual time constant.

13.5.3 Average Processing

1) Time Average
   Input value of specified channel accumulates during setting time and then the average value of the sum is shown with digital data.



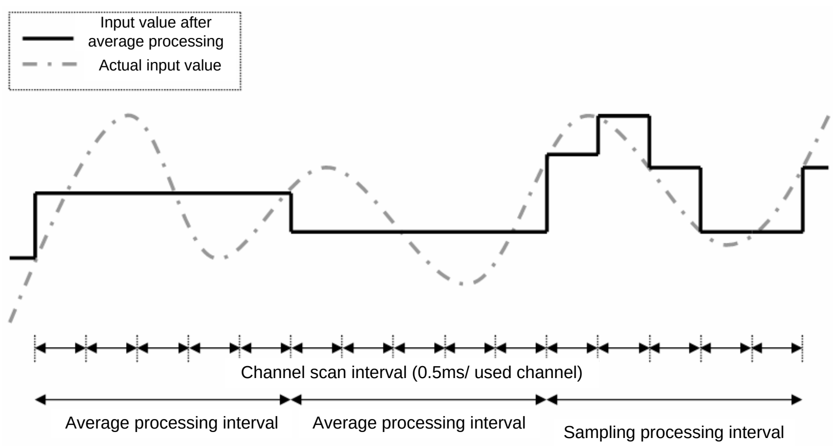
Setting range = 4 to 16,000 [ms]
In case of the time average, the average processing count is calculated by depending on the number of used channels.

$$\text{Average processing count} = \frac{\text{Average time}}{\text{Number of used channels x 0.5 ms}}$$

Time average is converted to count average in A/D conversion module internally, and then processed. In this case, remainder can be generated when dividing average time by (number of used channels X conversion speed). The remainder generated at this time is discarded, and the average number of processing is determined by the quotient of (set time) ÷ (number of channels used × conversion speed).

(i.e.) If the number of channels used is 4 and setting time is 151 ㎳,
Average processing count = 151 ㎳ ÷ (4 × 0.5 ㎳) = 75 counts·····remainder 1 → 75 counts

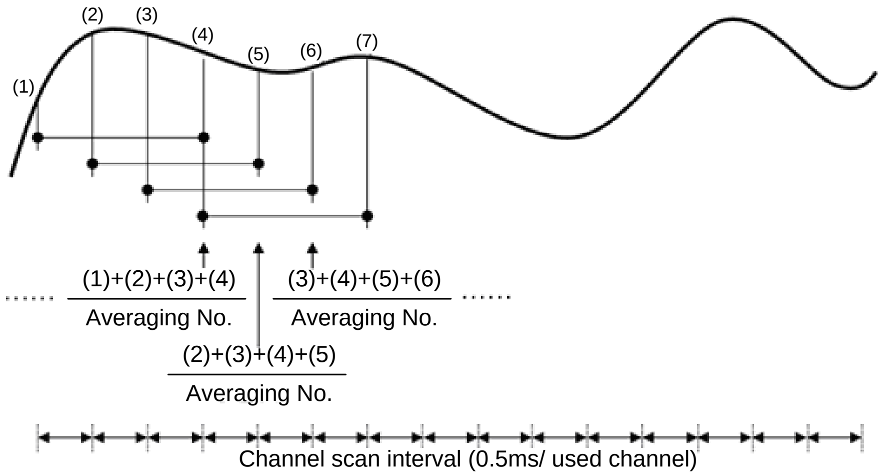
2) Count Average
Input value of specified channel accumulates during setting numbers and then the average value of the sum is shown with digital data



Setting range = 2 to 64,000 [times]
In case of count average, the average processing interval is calculated by depending on used channels.

Average processing interval [㎳] = Number of average count × Number of used channels × 0.5 ㎳

3) Moving Average
The inputs into the designated channel are accumulated for the presser number, its average is calculated, and output as digital data.
However, in moving average method, each scan provides its average value.



4) Weighted Average

Weighted average function processes transition of input data gradually by filter (delay) of input sampling data.

Setting range: 1 to 99 (%)

$$F[n] = (1-\alpha) \times A[n] + \alpha \times F[n-1]$$

F[n] : Current Weighted average output
A[n] : Current A/D Conversion value
F[n – 1] : Former Weighted average output
α : Weighted average constant
(0.01 to 0.99 : Weighted value of former value)

Setting Value	Filter Output Value				Description
	-	1 scan	2 scan	3 scan	
Not set	0	8,000	8,000	8,000	Not process weighted average
1	0	7,920	7,999	7,999	Apply 1% of former value
50	0	4,000	6,000	7,000	Apply 50% of former value
99	0	80	159	237	Apply 99% of former value

Notes

(1) In case of the time/number of average, every conversion time input value is not outputted. And precondition is retained until the average time/number is arrived.
.
2. Four kinds of average functions and introduced filtering functions that are above are able to deal with at the same time. When those are chosen at the same time, the top priority is filter function in the processing sequence. At that time digital data value is outputted as the final processing value.
(3) Number of used channel include input/output channel.

13.5.4 Detection Alarm (Input Disconnection)

When Input voltage (DC 1~5V) or Input current (DC 4~20 mA) is chosen for analog input range, the analog input module has a diagnostic function by checking for disconnection.
If the module shows disconnection, that means the parts of connections in the wiring connection are faulty. If so, check and take action.

1) Detection conditions
When input signal range of 4 to 20 mA and 1 to 5 V is used, disconnection of input circuit can be detected.
The detection conditions of each input signal range are as below.

Input signal range	Voltage/Current recognized as a disconnection
4~20㎃	0.8mA or less
1~5V	0.2V or less

2) When between used wiring and module is disconnected, the LED will be turned on/off 1s intervals.

3) Each channel can detect disconnection. However, Disconnection is only displayed for specified operation channel. The LED can commonly use the channel from 0 to 1. If one or more channel is disconnected, LED will be turned on/off.

Input connections	Channel run	AD LED status	Disconnection Flag
Normal	Operation	ON	OFF
	Stop	ON	OFF
Input wiring is disconnected or Input is not connected.	Operation	**Flicker (1 second)**	**On**
	Stop	ON	Off

4) In case of disconnection, disconnection flag of relevant channel will turn on and In case of connection, disconnection flag of relevant channel will turn off.

Disconnection Flag	Content	Status description
%UX0.1.72	Channel 0 disconnection	OFF : normal ON : disconnection
%UX0.1.73	Channel 1 disconnection	

5) In case of disconnection, the input value displays the lowest value among each input range.

13.5.5 Hold Last Value Function

When input signal exceeds the effective range, last input value is held.
This function can be set for each channel by I/O parameter setting or user program.

1) Used input range
In the channels that allow the hold last value function, the actual ranges provided within each digital conversion value are shown. For example, in case of operating output data type of unsigned value, original digital output value is shown from -192 to 16,191. However, if this function is allowed, it will be shown from 0 to 16,000.
It is recommended that the function should be setting when the input value is in the actual range.

(1) Digital output value depending on input range (unsigned value, signed value, percentile value)

Classification	Unsigned Value	Signed Value	Precise value	Percentile value
Function disabled	-192~16,191	-8,192~8,191	(2) Reference	-120~10,119
Function enabled	0~16,000	-8,000~8,000		0~10,000

(2) Digital output value depending on input range (Precise value)

Analog input range	Classification	Precise value
4~20㎃	Function disabled	3,808~20,191
	Function enabled	4,000~20,000
0~20㎃	Function disabled	-240~20,239
	Function enabled	0~20,000
1~5V	Function disabled	952~5,047
	Function enabled	1,000~5,000
0~5V	Function disabled	-60~5,059
	Function enabled	0~5,000
0~10V	Function disabled	-120~10,119
	Function enabled	0~10,000
-10~10V	Function disabled	-10,240~10,239
	Function enabled	-10,000~10,000

2) Operation
When operating with 4 ~ 20 ㎃ with this function enabled, the output for input value changes at the moment is as follows. (Output data type: In case of 0~16,000)

Input current (mA)	12㎃	3㎃	4㎃	12㎃	21㎃	20㎃
Digital output value	8,000	8,000	0	8,000	8,000	16,000
Note	-	**Previous value Hold**	-	-	**Previous value Hold**	-

13.5.6 Alarm Function

When the input signal is exceeded from valid value, the alarm will be shown through alarm flag of relevant channel.

1) Input detection condition
The detection conditions of each input signal range are as below.

Analog input range	Signal Difference (Difference)	Permission range	Lower limit alarm	Upper limit
4~20㎃	16㎃	1.2%	3.808㎃	20.192㎃
0~20㎃	20㎃		-0.240㎃	20.240㎃
1~5V	4V		0.952V	5.048V
0~5V	5V		-0.060V	5.060V
0~10V	10V		-0.120V	10.120V
-10~10V	20V		-10.240V	10.240V

2) Alarm indication for each channel
The alarm detection signal for each input channel is displayed in %UX0.1.48~%UX0.1.49 and %UX0.1.56~%UX0.1.57.
If input signal returns to the within of effective range, alarm detection signal also returns to the normal status automatically.

(1) Upper limit alarm

Variable name	Device assigned	Description	Status description
_01_AD0_HOOR	%UX0.1.48	CH0 upper limit alarm	OFF : normal On: Maximum alarm occurrence
_01_AD1_HOOR	%UX0.1.49	CH1 upper limit alarm	

(2) Lower limit alarm

Variable name	Device assigned	Description	Status description
_01_AD0_LOOR	%UX0.1.56	Channel 0 lower limit alarm	OFF : normal On: Maximum alarm occurrence
_01_AD1_LOOR	%UX0.1.57	Channel 1 lower limit alarm	

Notes

The channel conversion data will be 0 and Lower limit alarm flag will be ON if the input signal is out of the effective range as below when the input channel is enabled and hold last value function is enabled.

Analog input range	Hold last value function	Input signal	Lower limit alarm	Channel conversion value
4~20mA	On	3.808mA~4mA	On	0
		20mA~20.192mA		
0~20mA	ON	-0.24mA~0mA	On	0
		20mA~20.24mA		
1~5V	On	0.952V~1V	On	0
		5V~5.048V		
0~5V	On	-0.06V~0V	On	0
		5V~5.06V		
0~10V	On	-0.12V~0V	On	0
		10V~10.12V		
-10~10V	On	-10.24V~-10V	On	0
		10V~10.24V		

13.5.7 Setting Function of Channel Output Status

Set the output against stop and abnormal condition of motion controller.

1) Function
When initialization of module and error of XMC system are happened, use to prevent abnormal output.

2) Type
You can set an output status of channel among Previous, Min, Mid, Max value.
(1) Previous value: The last output operated normally is retained.
(2) Min: The Min value of each range is outputted.
(3) Mid: The Mid value of each range is outputted.
(4) Max: The Max value of each range is outputted.

3) Example
When the range of output channel is set to 0 ~ 10V and the output is 7V, if the system is changed from 'Run' to 'Stop', the output will be as follows depending on setting data of channel output status.
.

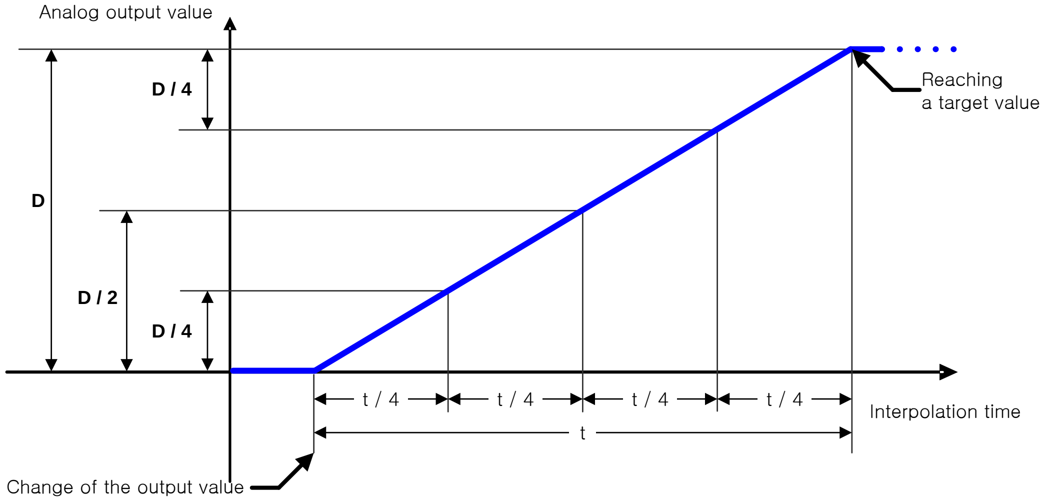
(1) Previous value: It retains the previous output, the 7V output.
(2) Min: It outputs 0V, which is the minimum value of the range.
(3) Mid: It outputs 5V, which is the middle value of that range.
(4) Maximum value: It outputs 10V, which is the maximum value of the range

13.5.8 Interpolation Method Setting

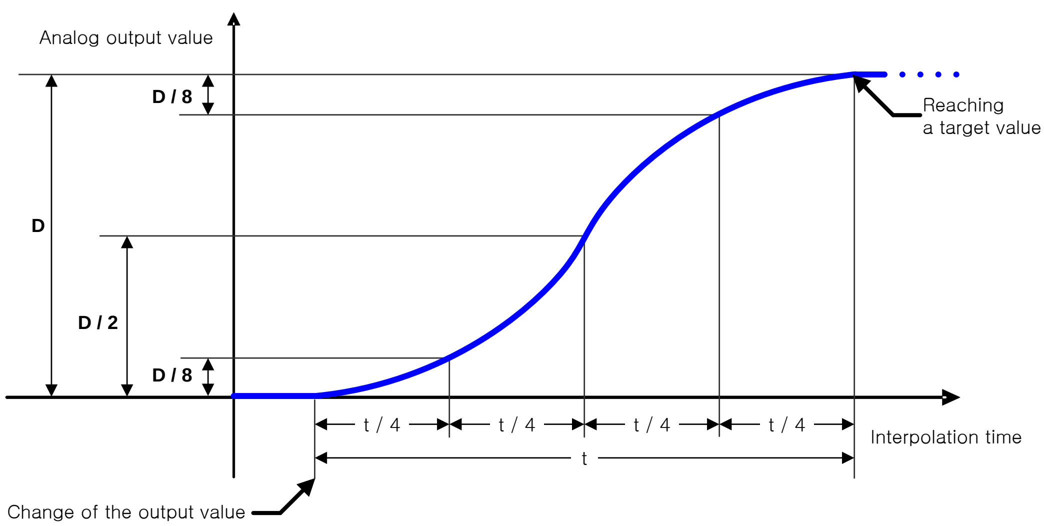
1) Function
The output signal of module is used to execute interpolation output depending on set interpolation time. When the voltage and current is outputted, it can be used to prevent transient response of load system as a suddenly changed output.

2) Interpolation method setting
Interpolation method can set the one among interpolation prohibition, linear interpolation S-type interpolation.

(a) Interpolation prohibition: It doesn't execute interpolation operation. And it outputs digital input value intact.
(b) Linear interpolation: The output is changed up to objective value with linear during the interpolation time.



(c) S-type interpolation: The output is changed up to objective value with S-type during the interpolation time.



3) Interpolation time setting
The interpolation time can be set with the one among 10[ms], 100[ms], 1[s], 60[s].
The output is changed depending on interpolation method setting during the set interpolation time.

4) Interpolation output value
The interpolation operation value that is currently being outputted can check in parameter area while using interpolation function.

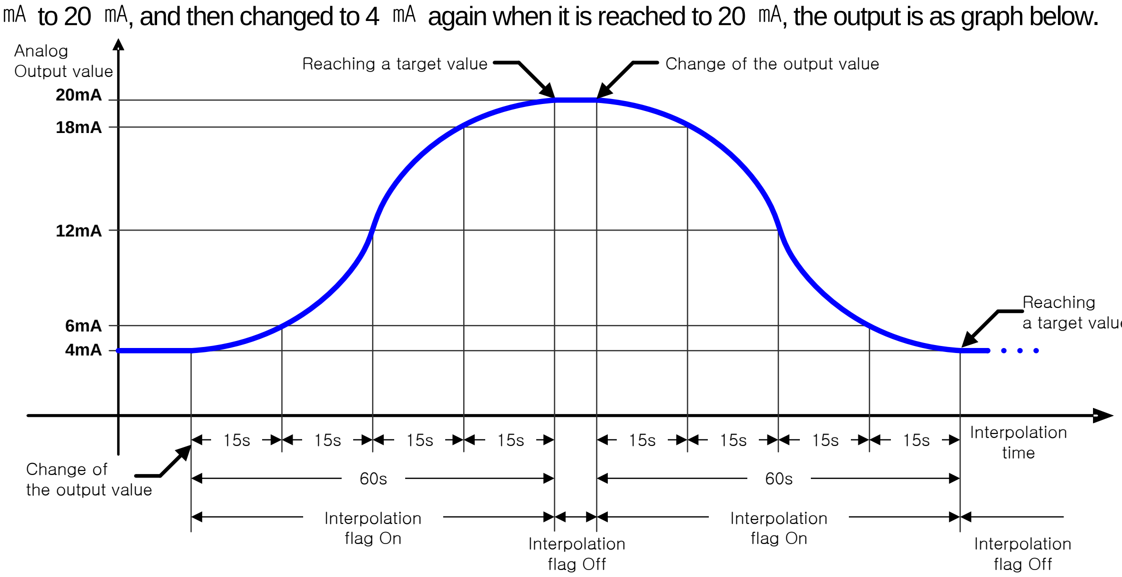
Variable name	Interpolation output value address	Content
_01_DA0_INTPVAL	%UW0.1.25	Channel 0 interpolation operation value
_01_DA1_INTPVAL	%UW0.1.26	Channel 1 interpolation operation value

5) Interpolation flag turns on while the interpolation is outputted. And when the interpolation output value is reached at objective value, it will turn off.

Variable name	Interpolation flag	content
_01_DA0_INTP	%UX0.1.64	Channel 0 interpolation output in operation
_01_DA1_INTP	%UX0.1.65	Channel 1 interpolation output in operation

※ Interpolation flag can be monitored when interpolation time is set to 1[s] or 60[s].

6) Example
The interpolation method is set to S-type interpolation and interpolation time is set to 60s. If the output is changed from 4 ㎃ to 20 ㎃, and then changed to 4 ㎃ again when it is reached to 20 ㎃, the output is as graph below.



Notes

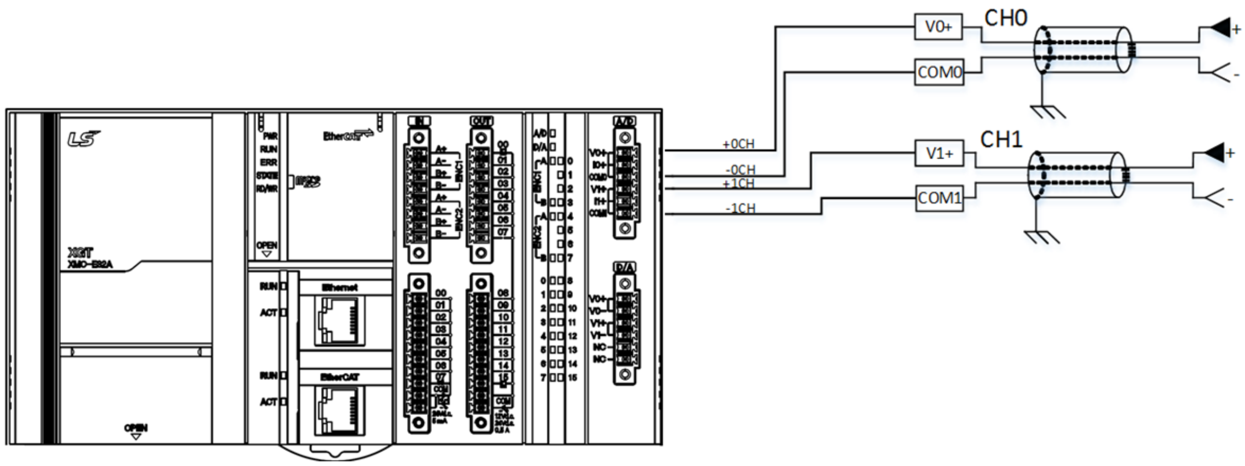
1) During the interpolation output, If the internal parameter is changed, the interpolation operation will be temporarily stopped and the output can be immediately changed to objective value.
2) If the change of internal parameter is needed, change the parameter during interpolation output after the flag turns off when the analog output value is not changed.

13.6 Wiring

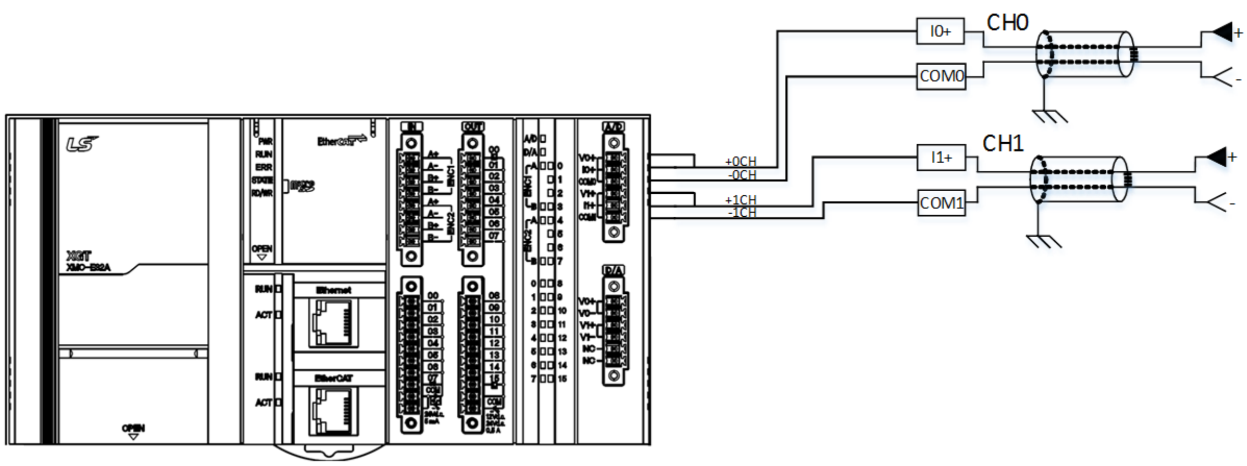
13.6.1 Example for Wiring Analog Input

1) The input resistance of current input circuit is 250 Ω (typ.).
2) The input resistance of voltage input circuit is 1 MΩ or more.
3) Set the operation mode only if you want to use channels.
4) Example for analog input wiring
   In voltage mode, use V+ and COM terminal for the channel. In current mode, short V+ and I+ terminal and then use I+ and COM terminal.

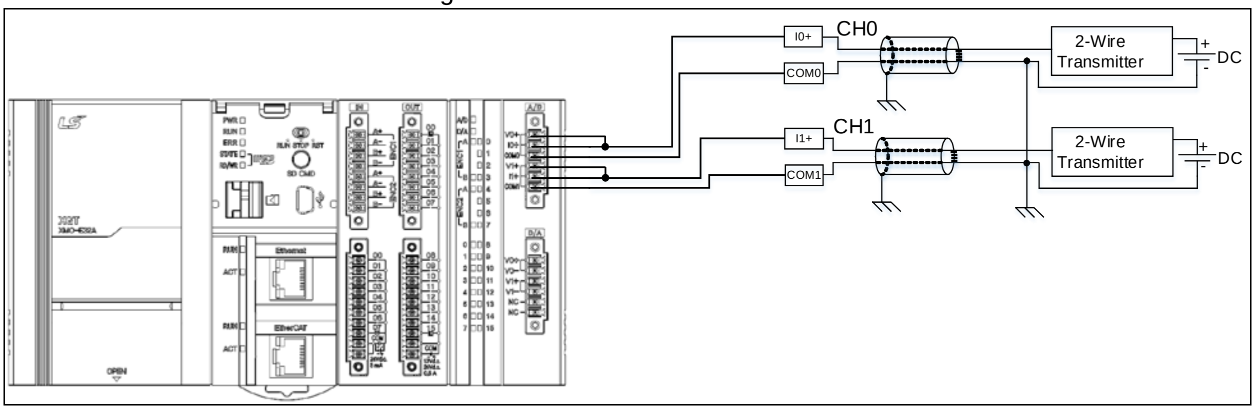
(1) Voltage wiring



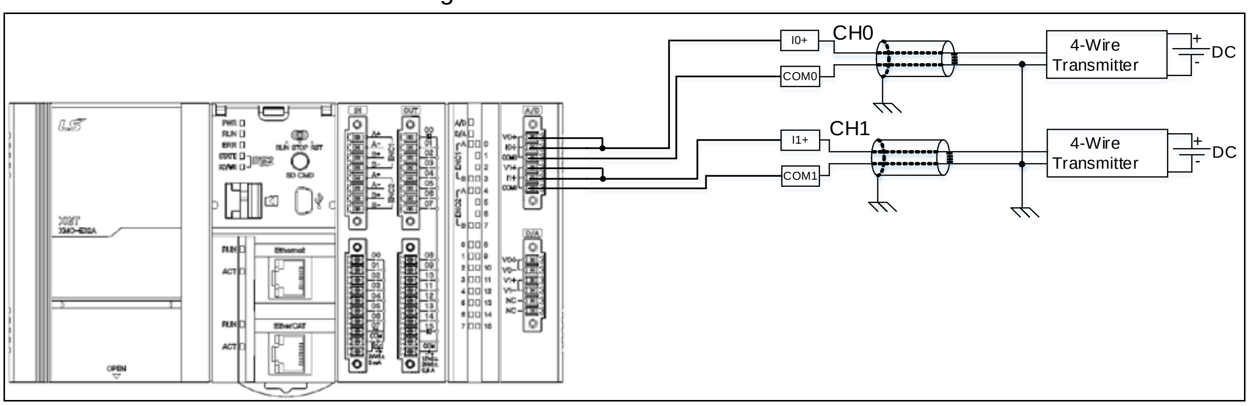
(2) Current wiring



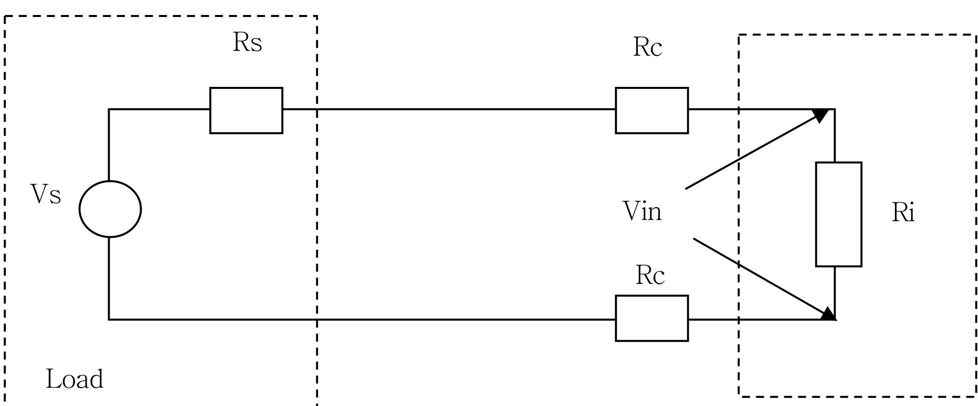
5) The example of analog input 2-Wire sensor/transmitter wiring (The current input)
Use I+ and COM terminal after connecting V+ with I+ terminal.



6) The example of analog input 4-Wire sensor/transmitter wiring (The current input)
Use I+ and COM terminal after connecting V+ with I+ terminal.



7) Relationship between voltage input accuracy and wiring length
In voltage input, the wiring (cable) length between transmitter or sensor and module influences digital-converted values of the module as specified below;
The values are as follows:



Analog input (voltage)

here

Rc: Resistance value due to line resistance of cable
Rs: Internal resistance value of transmitter or sensor
Ri: Internal resistance value (1㏁) of voltage input module

Vin: Voltage allowed to analog input module
% Vi: Tolerance of converted value (%) due to source and cable length in voltage input

$$Vin = \frac{Ri \times Vs}{[Rs + (2 \times Rc) + Ri]}$$

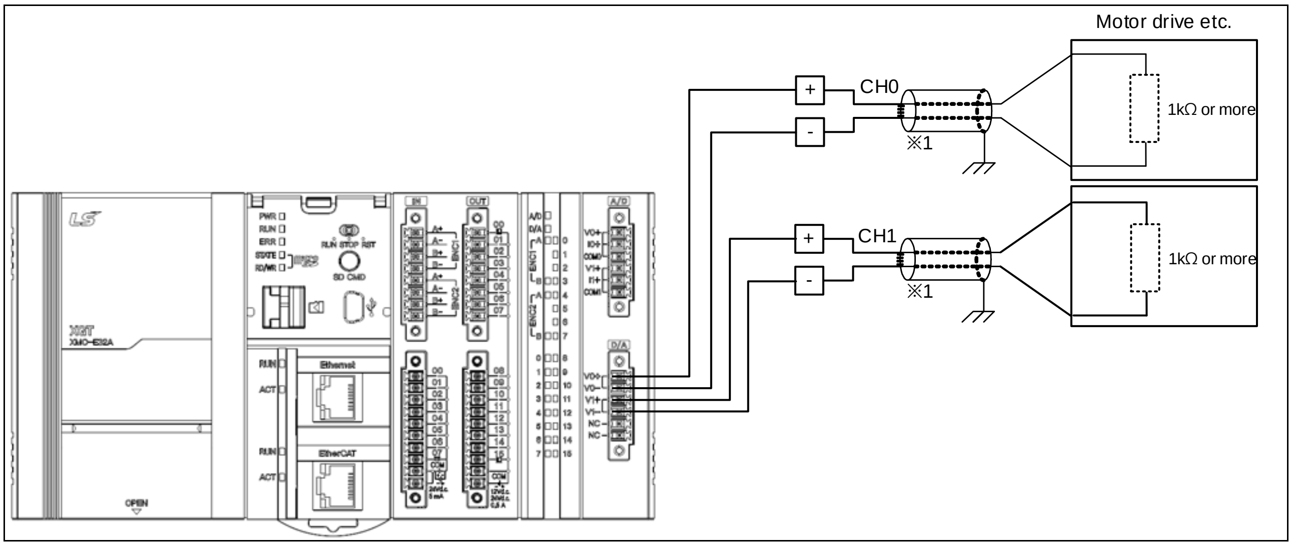
$$\%Vi = \left(1 - \frac{Vin}{Vs}\right) \times 100\%$$

Notes

1) While using a input voltage range among 1 to 5 V, 0 to 5 V, 0 to 10 V, -10 to 10 V If the external wiring is disconnected, It will take a certain amount of time to display output data value of 0 V. If you want to reduce that time, connect the resistance about 0.1 ㏁ to 1 ㏁ between input channel V+ and COM.

13.6.2 Example for Wiring Analog Output

1) Example for analog voltage ·current output wiring



※1: A twisted two core shielded wire should be used as wire.

13.7 Operation Parameter Setting

Built-in analog conversion module's operation parameters can be specified through XG5000's [I/O parameters].

1) Setting items

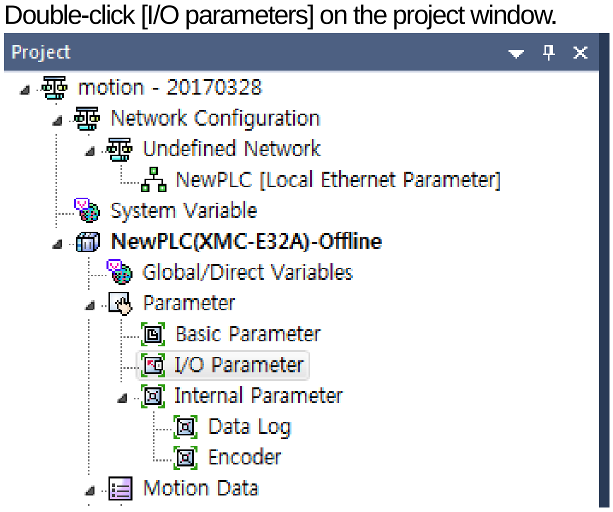
For the user's convenience of D/A conversion module, XG5000 provides GUI (Graphical User Interface) for parameters setting of D/A conversion module. Setting items available through [I/O parameters] on the XG5000 project window are as described below in the table.

Item	Content
[I/O parameter]	(a) Input parameter setting Specify the following setting items necessary for the module operation. 1) Operation Channel 2) Input range 3) Output Data Type 4) Filter constant 5) Average Method 6) Average value 7) Keep valid conversion values (b) Output parameter settings Specify the following setting items necessary for the module operation. 1) Operation Channel 2) Output range 3) Input data type 4) Channel output status setting 5) Interpolation method setting 6) Interpolation time (c) When the parameters set by user in XG5000 is downloaded, that data is saved in flash memory of motion controller.

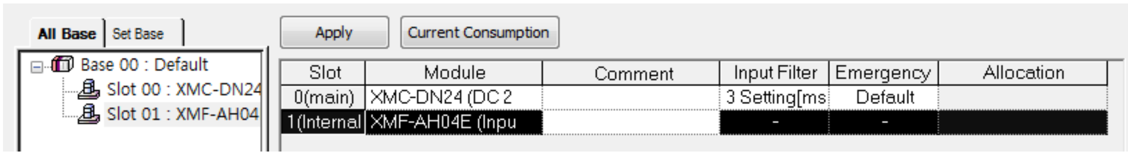
2) [I/O Parameter] Using method

(1) Run XG5000 to create a project. (Refer to XG5000 program manual for details on how to create the project)

(2) Double-click [I/O parameters] on the project window.

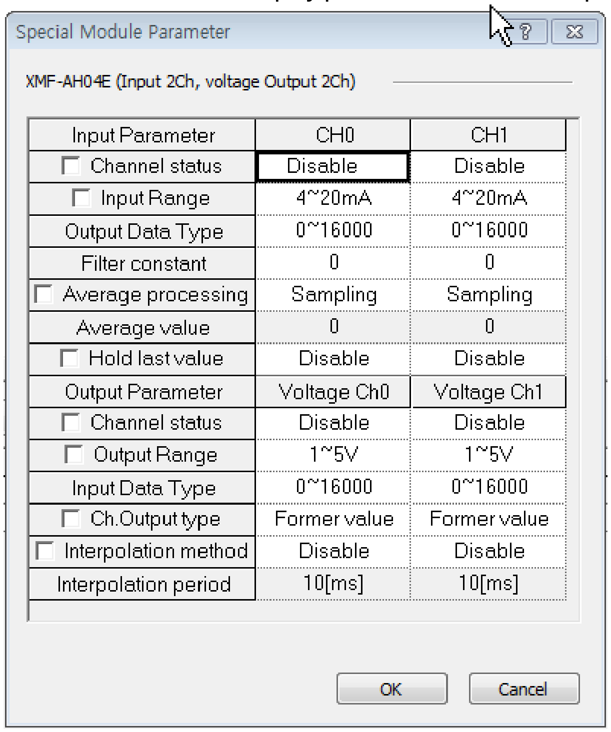


(3) [I/O Parameter setting] On the 'I/O Parameter setting' screen, find and clink the slot 1 (internal) which has embedded function.



(4) Click the arrow button on the screen above to display the screen where an applicable module can be selected.

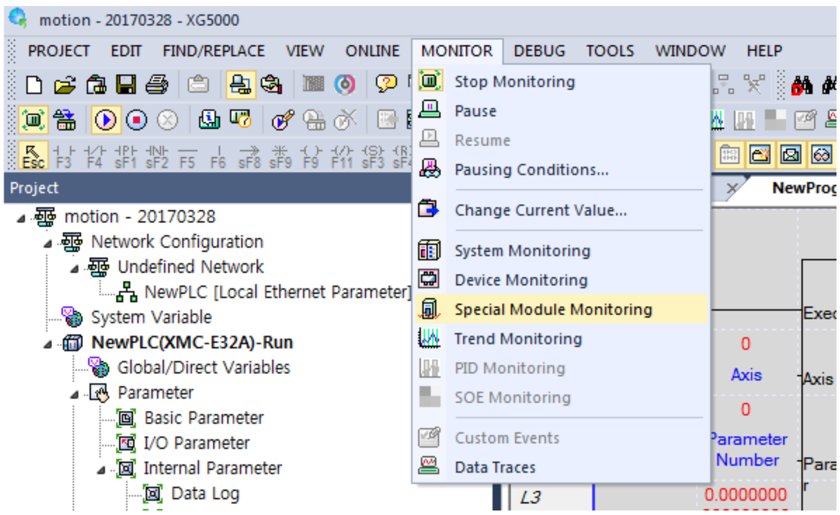
(5) A screen will be displayed for you to specify parameters for respective channels as below.
Click a desired item to display parameters to set for respective items.



13.8 Special Module Monitoring Functions

Functions of Special Module Monitoring are as described below.

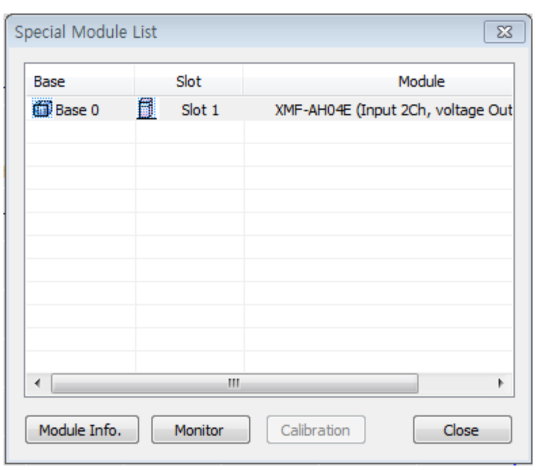
1) [Start of [Special Module Monitoring]
Go through [Online]-> [Connect] and [Monitor]-> [Special module Monitoring] to start. If the status is not online, [Special Module Monitoring] menu will not be activated.



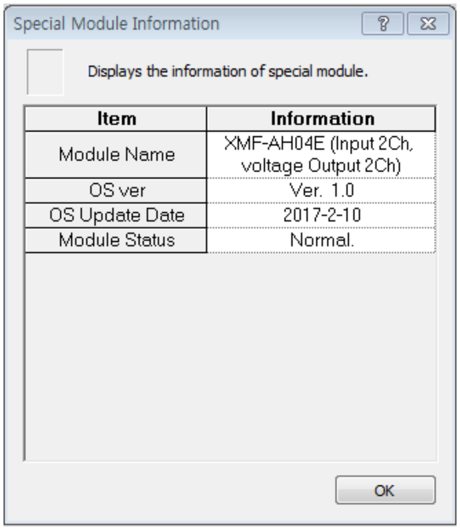
Notes

1) The screen may not normally be shown due to the lack of system resource. In this case, terminate all applications and try to start XG5000 again.
2) I/O parameter set in status of [Special Module Monitor] is temporally set to implement the test. Therefore, when the [special module monitor] status is terminated, the set I/O parameters are extinguished.
3) The test of [Special Module Monitor] is an examination function to check operation of the analog Input/output module when the sequence program is not made up.

2) How to use special module monitoring
(1) With XG5000 connected to motion controller on-line status), click [Monitor] → [Special Module Monitoring] to display 'Special Module Select' screen as below.
Window shows base/slot information in addition to 'special module type'.

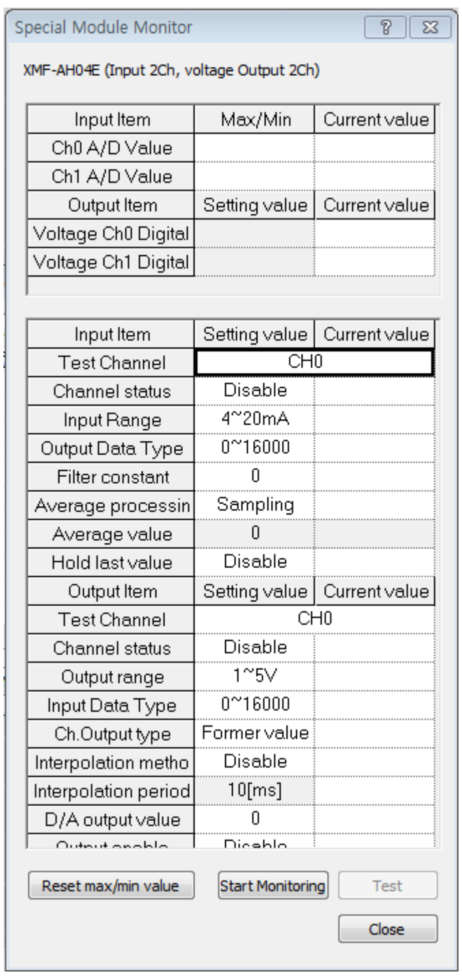


(2) Select “Special Module” and click [Module information] to display the information as shown below.



Item	Information
Module Name	XMF-AH04E (Input 2Ch, voltage Output 2Ch)
OS ver	Ver. 1.0
OS Update Date	2017-2-10
Module Status	Normal.

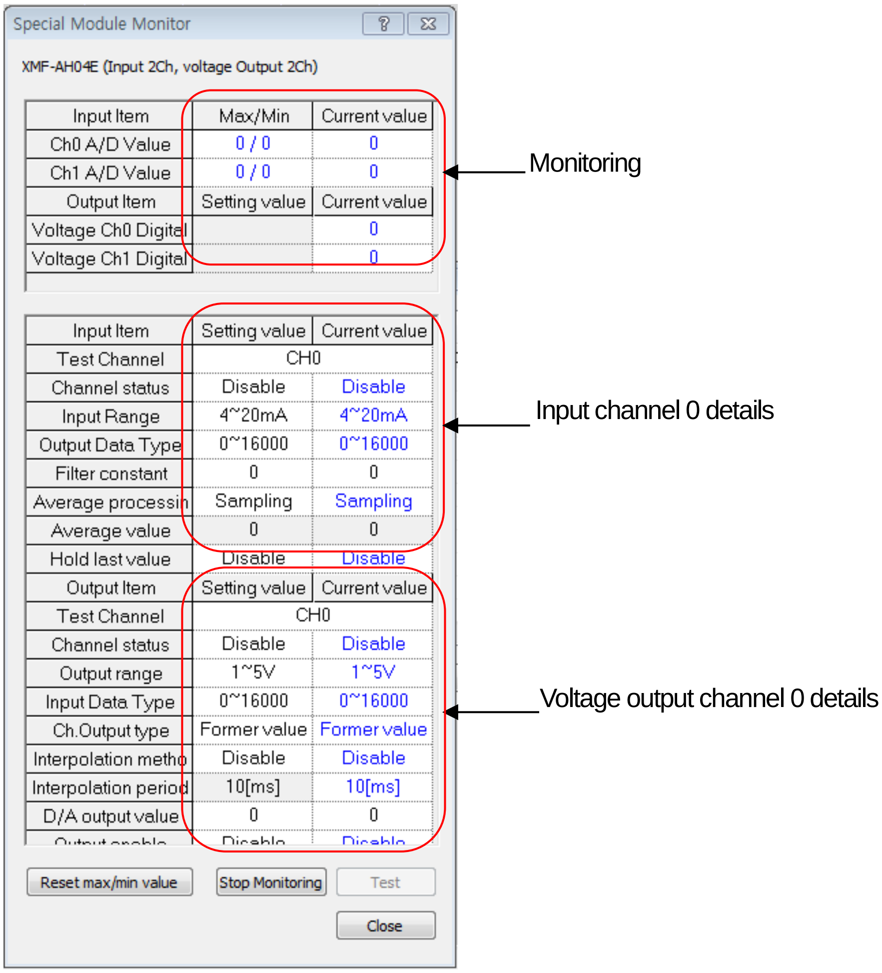
(3) Click [Monitor] on the “Special Module” screen in [Special Module List] to display [Special Module Monitoring] screen as shown below.



Input Item	Max/Min	Current value
Ch0 A/D Value		
Ch1 A/D Value		
Output Item	Setting value	Current value
Voltage Ch0 Digital		
Voltage Ch1 Digital		

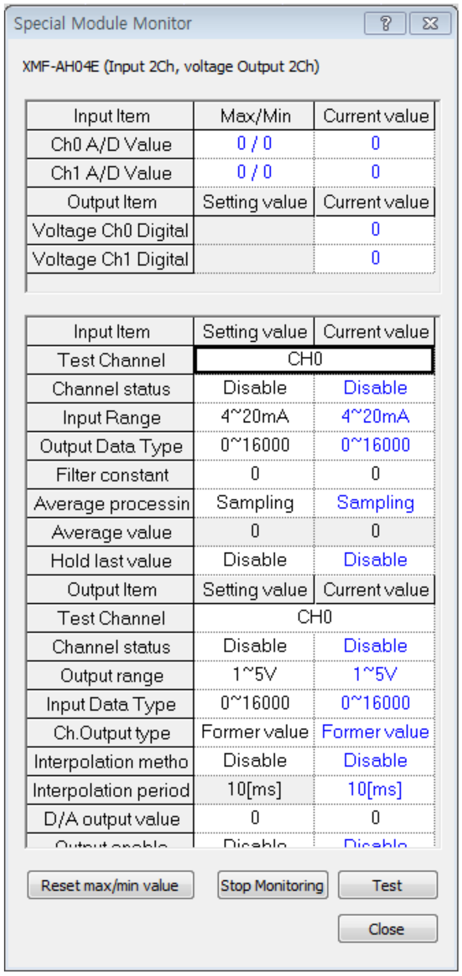
Input Item	Setting value	Current value
Test Channel	CH0	
Channel status	Disable	
Input Range	4~20mA	
Output Data Type	0~16000	
Filter constant	0	
Average processin	Sampling	
Average value	0	
Hold last value	Disable	
Output Item	Setting value	Current value
Test Channel	CH0	
Channel status	Disable	
Output range	1~5V	
Input Data Type	0~16000	
Ch.Output type	Former value	
Interpolation metho	Disable	
Interpolation period	10[ms]	
D/A output value	0	
Output enable	Disable	

(4) Start Monitoring: Click [Start Monitoring] to show digital input / output data of current operated channel.



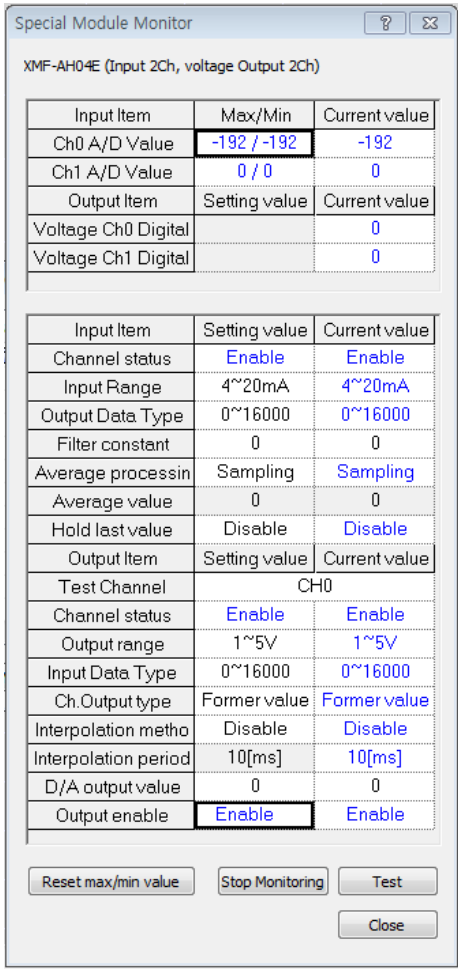
Execution screen of [Start Monitoring]

(5) Test: [Test] is a function to change the parameter of the embedded analog module which is presently set. In case of clicking the setting value in the bottom of the screen, you can change the parameter. [Test] is able to set only if operation status of motion controller is STOP.



Execution screen of [Test]

(6) Max/Min Value Monitor
Max/Min value of input channel in operation can be monitored.
However, visible Max/Min values are based on the present value.
So Max/Min value is not saved when [Monitoring/Test Screen] is closed.


Special Module Monitor

XMF-AH04E (Input 2Ch, voltage Output 2Ch)

Input Item	Max/Min	Current value
Ch0 A/D Value	-192 / -192	-192
Ch1 A/D Value	0 / 0	0
Output Item	Setting value	Current value
Voltage Ch0 Digital		0
Voltage Ch1 Digital		0

Input Item	Setting value	Current value
Channel status	Enable	Enable
Input Range	4~20mA	4~20mA
Output Data Type	0~16000	0~16000
Filter constant	0	0
Average processin	Sampling	Sampling
Average value	0	0
Hold last value	Disable	Disable
Output Item	Setting value	Current value
Test Channel	CH0	
Channel status	Enable	Enable
Output range	1~5V	1~5V
Input Data Type	0~16000	0~16000
Ch.Output type	Former value	Former value
Interpolation metho	Disable	Disable
Interpolation period	10[ms]	10[ms]
D/A output value	0	0
Output enable	Enable	Enable

Reset max/min value
Stop Monitoring
Test
Close

Execution screen of [Max/Min Value Monitor]

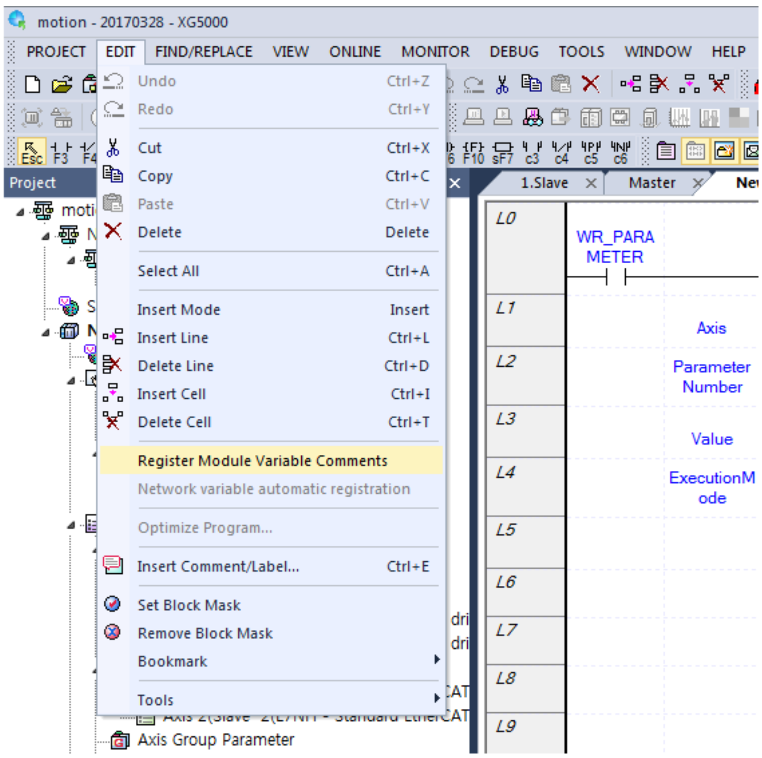
(7) close
[Close] is used to escape from the monitoring/test screen.
When the monitoring/test screen is closed, the max value, the min. value and the present value will not be saved any more.

13.9 Automatic Register U Devices

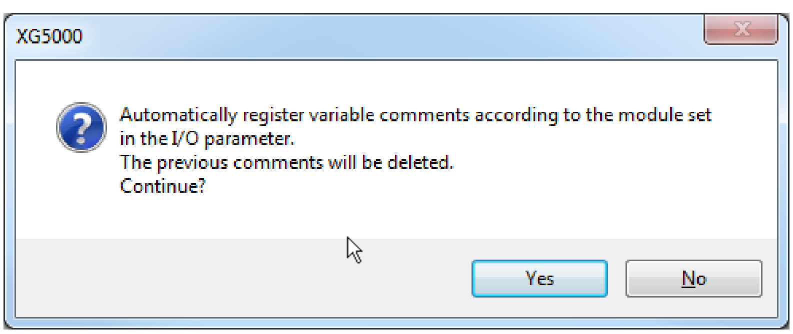
Register the variables for each module referring to the special module information that is set in the I/O parameter.
The user can modify the variables and comments.

1) Procedure

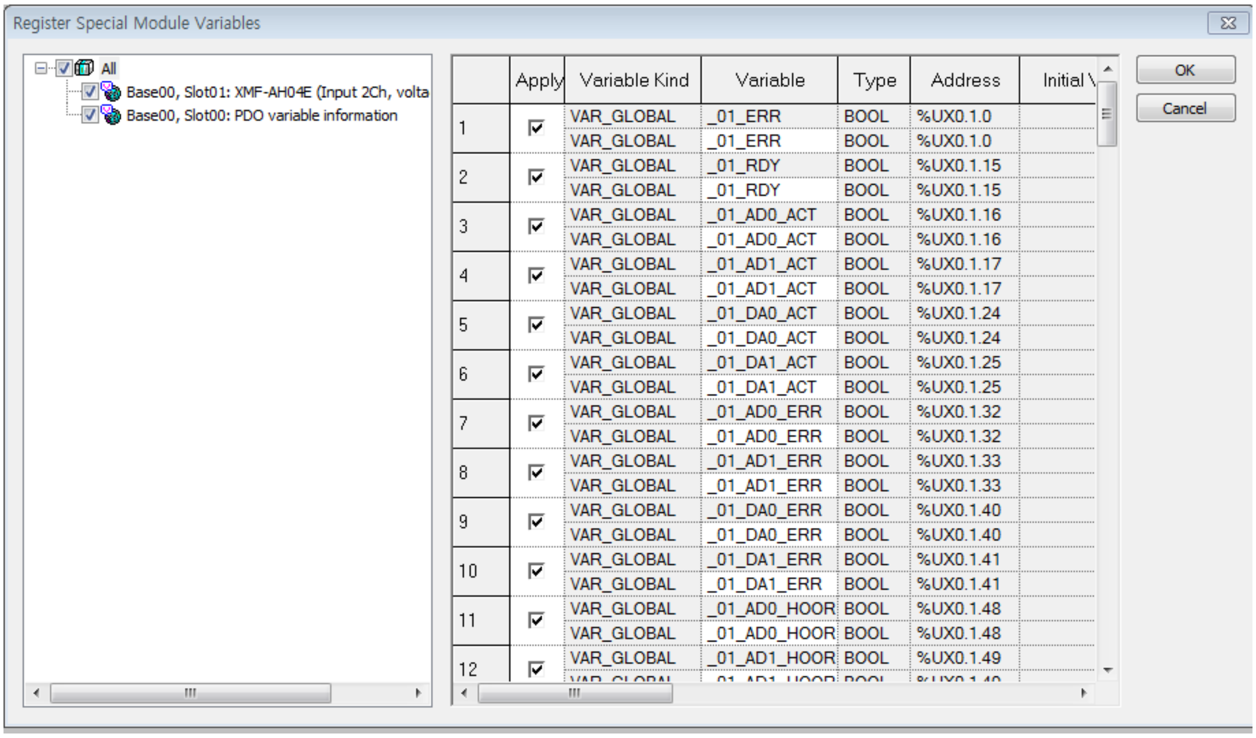
(1) Select ‘Automatic Register U Device(G)’ on ‘Edit’



(2) Click ‘Yes’.



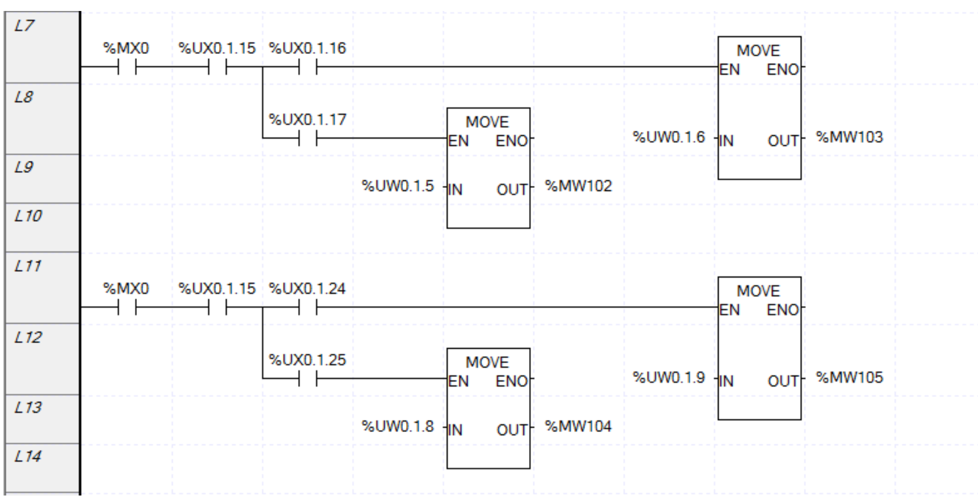
(3) As shown below, the variables are registered.



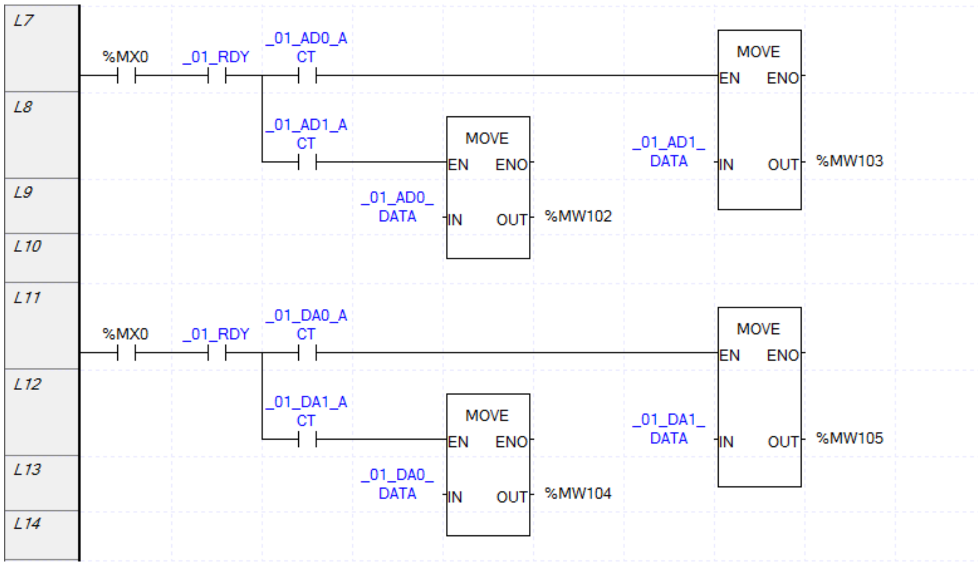
	Apply	Variable Kind	Variable	Type	Address	Initial \
1	☑	VAR_GLOBAL	_01_ERR	BOOL	%UX0.1.0	
		VAR_GLOBAL	_01_ERR	BOOL	%UX0.1.0	
2	☑	VAR_GLOBAL	_01_RDY	BOOL	%UX0.1.15	
		VAR_GLOBAL	_01_RDY	BOOL	%UX0.1.15	
3	☑	VAR_GLOBAL	_01_AD0_ACT	BOOL	%UX0.1.16	
		VAR_GLOBAL	_01_AD0_ACT	BOOL	%UX0.1.16	
4	☑	VAR_GLOBAL	_01_AD1_ACT	BOOL	%UX0.1.17	
		VAR_GLOBAL	_01_AD1_ACT	BOOL	%UX0.1.17	
5	☑	VAR_GLOBAL	_01_DA0_ACT	BOOL	%UX0.1.24	
		VAR_GLOBAL	_01_DA0_ACT	BOOL	%UX0.1.24	
6	☑	VAR_GLOBAL	_01_DA1_ACT	BOOL	%UX0.1.25	
		VAR_GLOBAL	_01_DA1_ACT	BOOL	%UX0.1.25	
7	☑	VAR_GLOBAL	_01_AD0_ERR	BOOL	%UX0.1.32	
		VAR_GLOBAL	_01_AD0_ERR	BOOL	%UX0.1.32	
8	☑	VAR_GLOBAL	_01_AD1_ERR	BOOL	%UX0.1.33	
		VAR_GLOBAL	_01_AD1_ERR	BOOL	%UX0.1.33	
9	☑	VAR_GLOBAL	_01_DA0_ERR	BOOL	%UX0.1.40	
		VAR_GLOBAL	_01_DA0_ERR	BOOL	%UX0.1.40	
10	☑	VAR_GLOBAL	_01_DA1_ERR	BOOL	%UX0.1.41	
		VAR_GLOBAL	_01_DA1_ERR	BOOL	%UX0.1.41	
11	☑	VAR_GLOBAL	_01_AD0_HOOR	BOOL	%UX0.1.48	
		VAR_GLOBAL	_01_AD0_HOOR	BOOL	%UX0.1.48	
12	☑	VAR_GLOBAL	_01_AD1_HOOR	BOOL	%UX0.1.49	

2) Save variables
   (1) The contents of 'View Variable' can be saved as a text file.
   (2) Select [Edit] → [Export to File].
   (3) The contents of 'View variable' are saved as a text file.

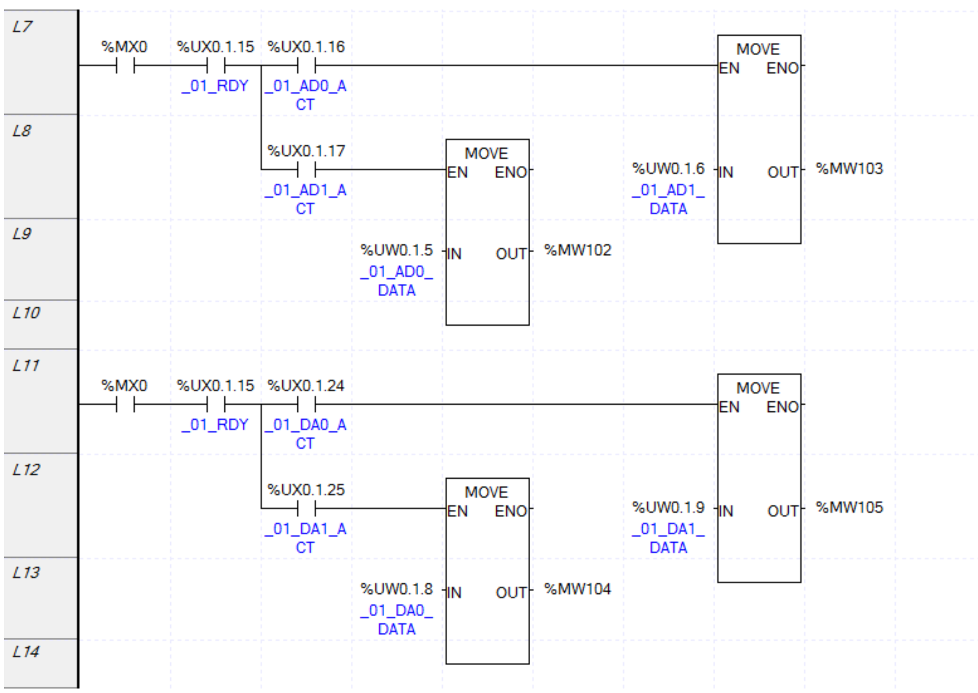
3) View variables in program
   (1) The example program of XG5000 is as shown below.



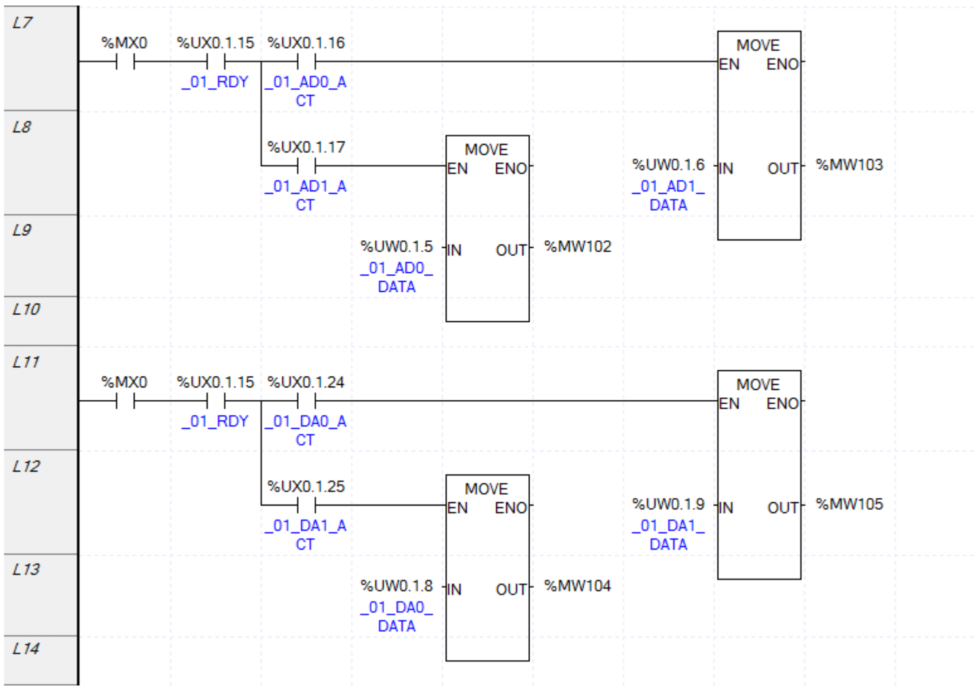
   (2) Select [View] → [Variables]. Click 'view variables on the 'view' of menu.



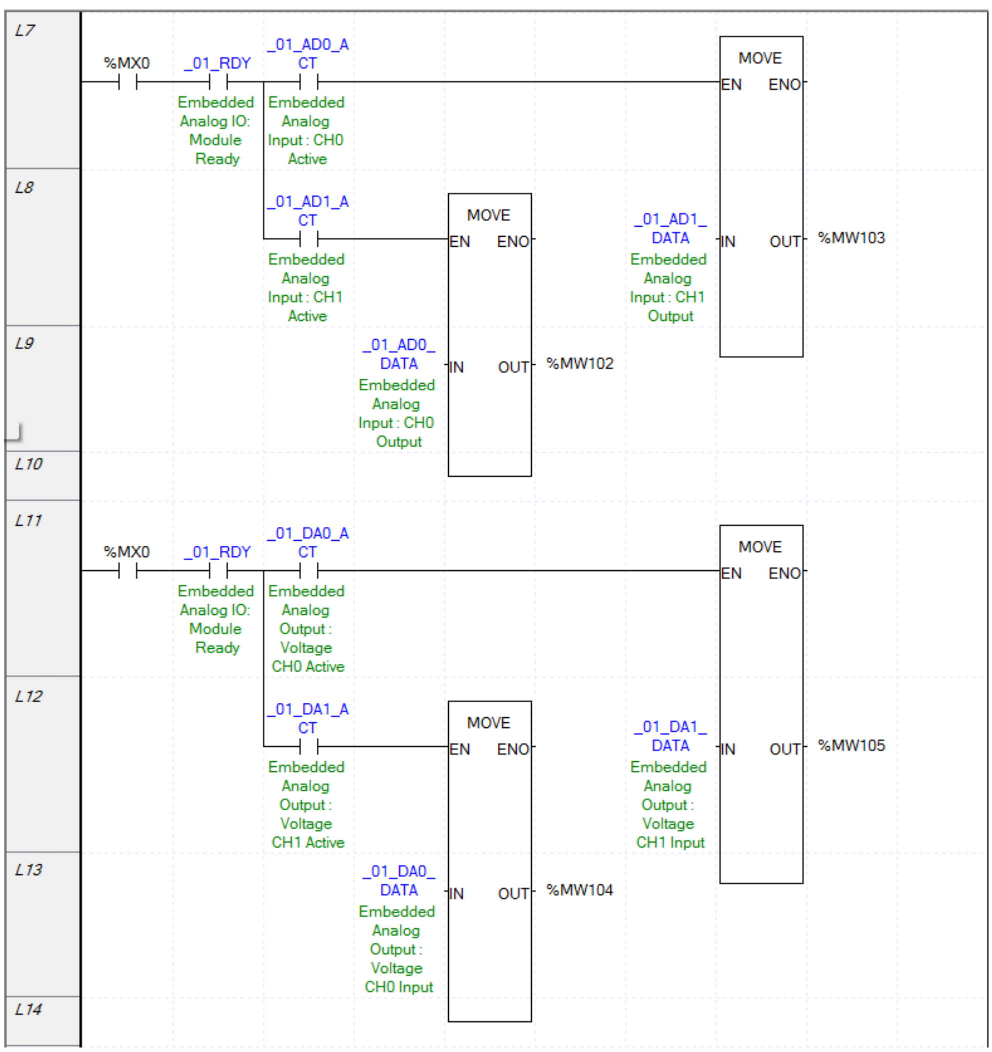
(3) Select [View] → [Devices/Variables]. Devices and variables are both displayed.



(4) Select [View] -> [Device/Comments]. Devices and comments are both displayed.



(5) Select [View] -> [Variables/Comments]. Variables and comments are both displayed.



13.10 Configuration and Function of Internal Memory

13.10.1 I/O Area of Built-in Analog Data

I/O area of built-in analog data is as displayed in table

Built-in analog input

Variable name	Type	Memory allocation	Description
_01_AD0_ACT	BOOL	%UX0.1.16	Channel 0 Active
_01_AD0_AVGTYPE	Byte	%UB0.1.34	Channel 0 Average processing
_01_AD0_AVGVAL	WORD	%UW0.1.18	Channel 0 Average value
_01_AD0_DATA	WORD	%UW0.1.5	Channel 0 output value
_01_AD0_DATATYPE	Byte	%UB0.1.26	Channel 0 Output data type setting
_01_AD0_ERR	BOOL	%UX0.1.32	Channel 0 error
_01_AD0_FILTCONST	WORD	%UW0.1.15	Channel 0 Filter constant
_01_AD0_HOLDVAL	BOOL	%UX0.1.320	Channel 0 Hold effective conversion value setting
_01_AD0_HOOR	BOOL	%UX0.1.48	Channel 0 upper limit alarm
_01_AD0_IDD	BOOL	%UX0.1.72	Channel 0 Input disconnection flag
_01_AD0_LOOR	BOOL	%UX0.1.56	Channel 0 lower limit alarm
_01_AD0_RANGE	Byte	%UB0.1.22	Channel 0 Range setting
_01_AD0_RUN	BOOL	%UX0.1.160	Channel 0 Operation setting
_01_AD1_ACT	BOOL	%UX0.1.17	Channel 1 Active
_01_AD1_AVGTYPE	Byte	%UB0.1.35	Channel 1 Average processing
_01_AD1_AVGVAL	WORD	%UW0.1.19	Channel 1 Average value
_01_AD1_DATA	WORD	%UW0.1.6	Channel 1 output value
_01_AD1_DATATYPE	Byte	%UB0.1.27	Channel 1 Output data type setting
_01_AD1_ERR	BOOL	%UX0.1.33	Channel 1 error
_01_AD1_FILTCONST	WORD	%UW0.1.16	Channel 1 Filter constant
_01_AD1_HOLDVAL	BOOL	%UX0.1.321	Channel 1 Hold effective conversion value setting
_01_AD1_HOOR	BOOL	%UX0.1.49	Channel 1 upper limit alarm
_01_AD1_IDD	BOOL	%UX0.1.73	CH1 input disconnection detection
_01_AD1_LOOR	BOOL	%UX0.1.57	Channel 1 lower limit alarm
_01_AD1_RANGE	Byte	%UB0.1.23	Channel 1 Range setting
_01_AD1_RUN	BOOL	%UX0.1.161	Channel 1 Operation setting
_01_AD_ACT_ARY	ARRAY[0..1] OF BOOL	%UX0.1.16	Run by channel
_01_AD_AVGTYPE_ARY	ARRAY[0..1] OF BYTE	%UB0.1.32	Average process by channel
_01_AD_AVGVAL_ARY	ARRAY[0..1] OF WORD	%UW0.1.18	Average value by channel
_01_AD_DATATYPE_ARY	ARRAY[0..1] OF BYTE	%UB0.1.26	Input data type by channel
_01_AD_DATA_ARY	ARRAY[0..1] OF WORD	%UW0.1.5	Conversion value by channel
_01_AD_ERR_ARY	ARRAY[0..1] OF BOOL	%UX0.1.32	Error by channel
_01_AD_FILTCONST_ARY	ARRAY[0..1] OF WORD	%UW0.1.15	Filter constant by channel

Variable name	Type	Memory allocation	Description
_01_AD_HOLDVAL_ARY	ARRAY[0..1] OF BOOL	%UX0.1.320	Specify to hold valid conversion value for each channel
_01_AD_HOOR_ARY	ARRAY[0..1] OF BOOL	%UX0.1.48	Alarm (Upper Limit) by channel
_01_AD_IDD_ARY	ARRAY[0..1] OF BOOL	%UX0.1.72	Input disconnection detection by channel
_01_AD_LOOR_ARY	ARRAY[0..1] OF BOOL	%UX0.1.56	Alarm (Lower Limit) by channel
_01_AD_RANGE_ARY	ARRAY[0..1] OF BYTE	%UB0.1.22	Range setting by channel
_01_AD_RUN_ARY	ARRAY[0..1] OF BOOL	%UX0.1.160	Operation setting by channel

Built-in analog output

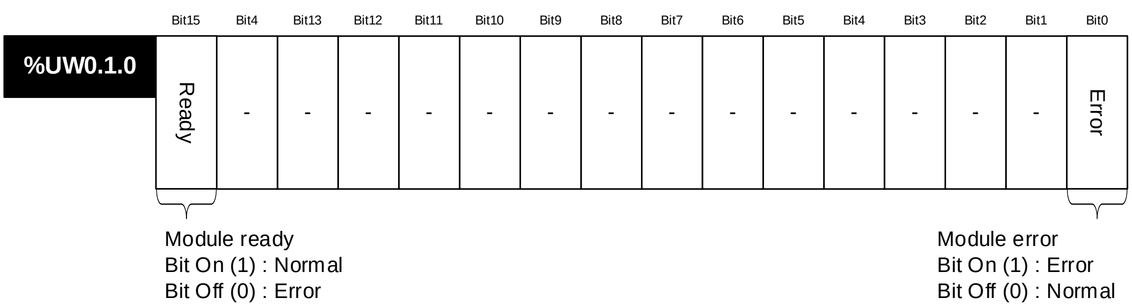
Variable name	Type	Memory allocation	Description
_01_DA0_ACT	BOOL	%UX0.1.24	Channel 0(Voltage) run
_01_DA0_DATA	WORD	%UW0.1.8	Channel 0(Voltage) Input data
_01_DA0_DATATYPE	Byte	%UB0.1.28	Channel 0 Output data type setting
_01_DA0_ERR	BOOL	%UX0.1.40	Channel 0(Voltage) Error
_01_DA0_INTP	BOOL	%UX0.1.64	Channel 0(Voltage) Interpolation Enabled
_01_DA0_INTPMTHD	Byte	%UB0.1.46	Channel 0(Voltage) Interpolation method
_01_DA0_INTPTIME	Byte	%UB0.1.48	Channel 0(Voltage) Interpolation time setting
_01_DA0_INTPVAL	WORD	%UW0.1.25	Channel 0(Voltage) Interpolation value
_01_DA0_OUTEN	BOOL	%UX0.1.112	Channel 0(Voltage) Output enable
_01_DA0_OUTSTAT	WORD	%UW0.1.21	Channel 0 output status setting
_01_DA0_RANGE	Byte	%UB0.1.24	Channel 0 Range setting
_01_DA0_RUN	BOOL	%UX0.1.168	Channel 0 Operation setting
_01_DA1_ACT	BOOL	%UX0.1.25	Channel 1(Voltage) run
_01_DA1_DATA	WORD	%UW0.1.9	Channel 1(Voltage) Input data
_01_DA1_DATATYPE	Byte	%UB0.1.29	Channel 1 Output data type setting
_01_DA1_ERR	BOOL	%UX0.1.41	Channel 1(Voltage) Error
_01_DA1_INTP	BOOL	%UX0.1.65	Channel 1(Voltage) Interpolation Enabled
_01_DA1_INTPMTHD	Byte	%UB0.1.47	Channel 1(Voltage) Interpolation method
_01_DA1_INTPTIME	Byte	%UB0.1.49	Channel 1(Voltage) Interpolation time setting
_01_DA1_INTPVAL	WORD	%UW0.1.26	Channel 1(Voltage) Interpolation value
_01_DA1_OUTEN	BOOL	%UX0.1.113	Channel 1(Voltage) Output enable
_01_DA1_OUTSTAT	WORD	%UW0.1.22	Channel 1 output status setting
_01_DA1_RANGE	Byte	%UB0.1.25	Channel 1 Range setting
_01_DA1_RUN	BOOL	%UX0.1.169	Channel 1 Operation setting
_01_DA_ACT_ARY	ARRAY[0..1] OF BOOL	%UX0.1.24	Run by channel
_01_DA_DATATYPE_ARY	ARRAY[0..1] OF BYTE	%UB0.1.28	Input data type by channel
_01_DA_DATA_ARY	ARRAY[0..1] OF WORD	%UW0.1.8	Input values by voltage channel
_01_DA_ERR_ARY	ARRAY[0..1] OF BOOL	%UX0.1.40	Error by channel
_01_DA_INTPMTHD_ARY	ARRAY[0..1] OF BYTE	%UB0.1.46	Interpolation method setting by channel
_01_DA_INTPTIME_ARY	ARRAY[0..1] OF BYTE	%UB0.1.48	Interpolation time setting by channel

Variable name	Type	Memory allocation	Description
_01_DA_INTPVAL_ARY	ARRAY[0..1] OF WORD	%UW0.1.25	Interpolation value by channel
_01_DA_INTP_ARY	ARRAY[0..1] OF BOOL	%UX0.1.64	Interpolation enabled by channel
_01_DA_OUTEN_ARY	ARRAY[0..1] OF BOOL	%UX0.1.112	Output enable setting by channel
_01_DA_OUTSTAT_ARY	ARRAY[0..1] OF WORD	%UW0.1.21	Output status setting by channel
_01_DA_RANGE_ARY	ARRAY[0..1] OF BYTE	%UB0.1.24	Range setting by channel
_01_DA_RUN_ARY	ARRAY[0..1] OF BOOL	%UX0.1.168	Operation setting by channel

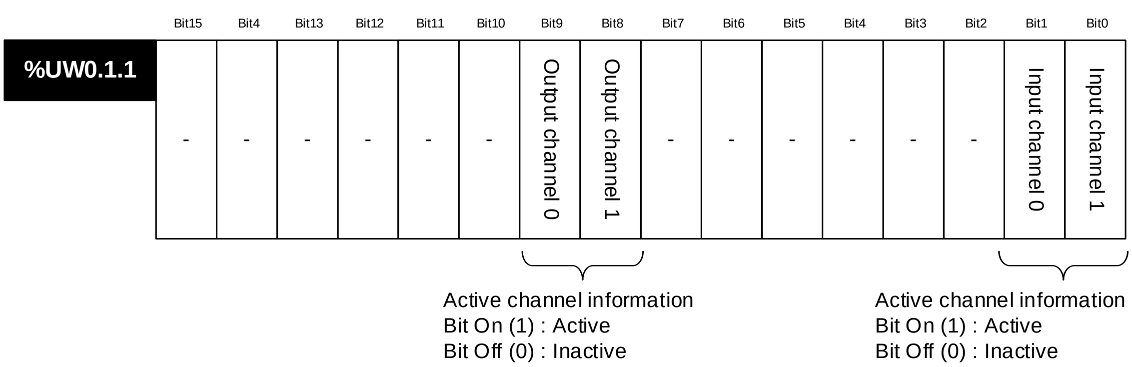
Built-in analog common

Variable name	Type	Memory allocation	Description
_01_ERR	BOOL	%UX0.1.0	Module error
_01_RDY	BOOL	%UX0.1.15	Module ready
_01_SETTINGERR	WORD	%UW0.1.27	Setting error information

1) Built-in analog module ready/error flag (_01_RDY/_01_ERR)
   (1) %UX0.1.15: It will be ON when module is powered or reset with D/A conversion ready to process A/D conversion.
   (2) %UX0.1.0 : It is a flag to display the error status of built-in analog module



2) Channel active information
   (1) This area shows the channel being used.
   (2) _01_AD0_ACT (%UX0.1.16): Input channel 0 active
   _01_AD1_ACT (%UX0.1.17): Input channel 1 active
   _01_DA0_ACT (%UX0.1.24): Output channel 0 active
   _01_DA0_ACT (%UX0.1.25): Output channel 1 active



3) Error channel information
   (1) This area shows the channel error status.
   (2) _01_AD0_ERR (%UX0.1.32): Input channel 0 error
       _01_AD1_ERR (%UX0.1.33): Input channel 1 error
       _01_DA0_ERR (%UX0.1.40): output channel 0 error
       _01_DA0_ERR (%UX0.1.41): output channel 1 error

4) Input upper/lower alarm flag
   (1) This area shows upper/lower alarm per channel status.
   (2) _01_AD0_HOOR (%UX0.1.48): Input channel 0 upper limit alarm
       _01_AD0_HOOR (%UX0.1.49): Input channel 1 upper limit alarm
       _01_AD0_LOOR (%UX0.1.56): Input channel 0 lower limit alarm
       _01_AD0_LOOR (%UX0.1.57): Input channel 1 Lower limit alarm

	Bit15	Bit4	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
%UW0.1.3	-	-	-	-	-	-	Input channel 0	Input channel 1	-	-	-	-	-	-	Input channel 0	Input channel 1

Lower alarm detection
Bit On(1) : Lower alarm status
Bit Off(0) : Normal

Upper alarm detection
Bit On(1) : Upper alarm status
Bit Off(0) : Normal

5) Disconnection input/interpolation output status
   (1) This area shows the channel detecting input disconnection and being outputting interpolation.
   (2) _01_DA0_INTP (%UX0.1.64): Output channel 0 outputting interpolation
       _01_DA0_INTP (%UX0.1.65): Output channel 1 outputting interpolation
       _01_AD0_IDD (%UX0.1.72): Input channel 0 lower Disconnection Detection
       _01_AD1_IDD (%UX0.1.73): Input channel 1 upper Disconnection Detection

	Bit15	Bit4	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
%UW0.1.4	-	-	-	-	-	-	Input channel 0 disconnection	Input channel 1 disconnection	-	-	-	-	-	-	Output channel 0 interpolation	Output channel 1 interpolation

Input disconnection detection
Bit On(1) : Disconnection
Bit Off(0) : Normal

Interpolation output status
Bit On(1) : Interpolation status
Bit Off(0) : Don't

6) Digital output value
   (1) This area shows converted(A/D) digital output value by channel in buffer memory (%UW0.1.5 ~%UW0.1.6)
   (2) Digital output value is stored in 16-bit binary number.
   (3) _01_AD0_DATA (%UW0.1.5): Input channel 0 conversion value.
       _01_AD1_DATA (%UW0.1.6): Input channel 1 conversion value.

	Bit15 ~ Bit0 (Bit15, Bit4, Bit13, Bit12, Bit11, Bit10, Bit9, Bit8, Bit7, Bit6, Bit5, Bit4, Bit3, Bit2, Bit1, Bit0)
%UW0.1.5	Input channel 0 conversion value
%UW0.1.6	Input channel 1 conversion value

7) Output permission setting
   (1) The output enable / disable for each channel can be set.
   (2) When the output permission is not set, the output of all channels will be prohibited
   (3) _01_DA0_OUTEN (%UX0.1.112): Output channel 0 Enable Output.
       _01_DA1_OUTEN (%UX0.1.113): Output channel 1 Enable Output.

	Bit15	Bit4	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
%UW0.1.7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Output channel 0	Output channel 1

Output permission information
Bit On(1) : Permission
Bit Off(0) : Prohibition

8) Output digital input values
   (1) Unsigned value (-192~16,191 / 0~16,191), Signed value (-8,192~8,191 / -8,000~8,191), Precise value(-952~5,047 / -60~5,059 / -120~10,119 / -10,240~10,239 / 3,808~20,191 / 0~20,239), Percentile value(-120~10,119 / 0~10,119) can be used within these ranges depending on the setting of input data type.
   (2) If the digital input value is not set, it will be handled as '0'.
   (3) _01_DA0_DATA (%UW0.1.8): Output channel 0 input data
       _01_DA1_DATA (%UW0.1.9): Output channel 1 input data

	Bit15	Bit4	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
%UW0.1.8	Output channel 0 digital data															
%UW0.1.9	Output channel 1 digital data															

9) Setting operation channel
   (1) If the operating channel is not set, overall channel status is STOP.
   (2) _01_AD0_RUN (%UX0.1.160): Input channel 0 operating setting
       _01_AD1_RUN (%UX0.1.161): Input channel 1 operating setting
       _01_DA0_RUN (%UX0.1.168): Output channel 0 operating setting
       _01_DA1_RUN (%UX0.1.169): Output channel 1 operating setting

	Bit15	Bit4	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
%UW0.1.10	-	-	-	-	-	-	Output channel 0	Output channel 1	-	-	-	-	-	-	Input channel 0	Input channel 1

Output channel setting
Bit On (1) : Operation
Bit Off (0) : Stop

Input channel setting
Bit On (1) : Operation
Bit Off (0) : Stop

10) Input range setting
   (1) The ranges of analog input voltage are DC 1~5V, DC 0~5V, DC 0~10V, DC -10~10V, the ranges of analog current input are DC 4~20mA, DC 0~20mA.

(2) When the input range is not set, it is processed as DC 4~20mA range.
(3) _01_AD0_RANGE (%UB0.1.22): Input channel 0 range setting.
_01_AD1_RANGE (%UB0.1.23): Input channel 1 range setting.

<table>
<tr><td></td><td>Bit15</td><td>Bit4</td><td>Bit13</td><td>Bit12</td><td>Bit11</td><td>Bit10</td><td>Bit9</td><td>Bit8</td><td>Bit7</td><td>Bit6</td><td>Bit5</td><td>Bit4</td><td>Bit3</td><td>Bit2</td><td>Bit1</td><td>Bit0</td></tr>
<tr><td>%UW0.1.11</td><td colspan="4">Don't Care</td><td colspan="4">Input channel 1</td><td colspan="4">Don't Care</td><td colspan="4">Input channel 0</td></tr>
</table>

Input range setting (4 Bit per channel)
0 : 4~20㎃
1 : 0~20㎃
2 : 1~5V
3 : 0~5V
4 : 0~10V
5 : -10~10V

11) Output range setting
(1) The ranges of analog output voltage are DC 1~5V, DC 0~5V, DC 0~10V, DC -10~10V.
(2) When the input range is not set or it is entered out of setting values, it is handled as range of DC 1~5V.
(3) _01_DA0_RANGE (%UB0.1.24): Output channel 0 range setting.
_01_DA1_RANGE (%UB0.1.25): Output channel 1 range setting.

<table>
<tr><td></td><td>Bit15</td><td>Bit4</td><td>Bit13</td><td>Bit12</td><td>Bit11</td><td>Bit10</td><td>Bit9</td><td>Bit8</td><td>Bit7</td><td>Bit6</td><td>Bit5</td><td>Bit4</td><td>Bit3</td><td>Bit2</td><td>Bit1</td><td>Bit0</td></tr>
<tr><td>%UW0.1.12</td><td colspan="4">Don't Care</td><td colspan="4">Output channel 1</td><td colspan="4">Don't Care</td><td colspan="4">Output channel 0</td></tr>
</table>

Output range setting (4 Bit per channel)
0 : 1~5V
1 : 0~5V
2 : 0~10V
3 : -10~10V

12) Built-in analog Input data type setting
(1) The range of digital input data (A/D conversion) can be specified for respective channels.
(2) If the input data range is not specified, the range of all the channels will be set to 0 ~ 16000.
(3) _01_AD0_DATATYPE (%UB0.1.26): Input channel 0 - A/D Conversion output data type setting.
_01_AD1_DATATYPE (%UB0.1.27): Input channel 1 - A/D Conversion output data type setting.

<table>
<tr><td></td><td>Bit15</td><td>Bit4</td><td>Bit13</td><td>Bit12</td><td>Bit11</td><td>Bit10</td><td>Bit9</td><td>Bit8</td><td>Bit7</td><td>Bit6</td><td>Bit5</td><td>Bit4</td><td>Bit3</td><td>Bit2</td><td>Bit1</td><td>Bit0</td></tr>
<tr><td>%UW0.1.13</td><td colspan="6">Don't Care</td><td colspan="2">Input channel 1</td><td colspan="6">Don't Care</td><td colspan="2">Input channel 0</td></tr>
</table>

A/D conversion output data type setting (2 Bit per channel)
0 : 1~16,000
1 : -8,000~8,000
2 : Precise value
3 : 0~10,000

4~20㎃ : 4,000~20,000
0~20㎃ : 0~20,000
1~5V : 1,000~5,000
0~5V : 0~5,000
0~10V : 0~10,000
-10~10V : -10,000~10,000

13) Built-in analog Output data type setting

(1) The range of digital input data (D/A conversion) can be specified for respective channels.

(2) If the input data range is not specified, the range of all the channels will be set to 0 ~ 16000.

(3) _01_DA0_DATATYPE (%UB0.1.28): Output channel 0 – D/A Conversion output data type setting.
_01_DA1_DATATYPE (%UB0.1.29): Output channel 1 – D/A Conversion output data type setting.

	Bit15	Bit4	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
%UW0.1.14	Don't Care						Output Channel 1		Don't Care						Output Channel 0	

D/A conversion input data type setting (2 Bit per channel)
0 : 1~16,000
1 : -8,000~8,000
2 : Precise value
3 : 0~10,000

1~5V : 1,000~5,000
0~5V : 0~5,000
0~10V : 0~10,000
-10~10V : -10,000~10,000

14) Filter constant setting

(1) When the filter constant is specified with 0, the filter will not be operated.

(2) If input filter constant is not set, the filter will not be operated

(3) _01_AD0_FILCONST (%UW0.1.15): Input channel 0 Filter constant
_01_AD0_FILCONST (%UW0.1.16): Input channel 1 Filter constant

	Bit15 … Bit0
%UW0.1.15	Input channel 0 filter constant(0 or 64,000ms)
%UW0.1.16	Input channel 1 filter constant(0 or 64,000ms)

15) Average process method setting

(1) When setting average process, the average process method is selected among time average, count average, moving average, or weighted average.

(2) If setting average process is not specified, all channels will not handle the average process.

(3) _01_AD0_AVGTYPE (%UB0.1.34): Input channel 0 average process method setting.
_01_AD1_AVGTYPE (%UB0.1.35): Input channel 1 average process method setting.

	Bit15	Bit4	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
%UW0.1.17	Don't Care						Input channel 1		Don't Care						Input channel 0	

Average process method setting (4 Bit per channel)
0 : Sampling
1 : Time average
2 : Count average
3 : Moving average
4 : Weighted average

16) Average values

(1) Set to range of 4 ~ 16,000 as time average value.

(2) Set to range of 2 ~ 64,000 as count average value.
(3) Set to range of 2 ~ 100 as moving average value.
(4) Set to range of 1~ 99 as weighted average value.
(5) If average process method is set to 0(sampling process) and average value is set to 0, the input channel will not do average process, and sampling value will be output.
(6) _01_AD0_AVGVAL (%UW0.1.18): Input channel 0 Average value.
_01_AD1_AVGVAL (%UW0.1.19): Input channel 1 Average value.

	Bit15 ~ Bit0
%UW0.1.18	Input channel 0 average value
%UW0.1.19	Input channel 1 average value

17) Hold last value setting
(1) In case that hold last value function is set at the same time, if the invalid value is come, the late valid value will only be retained.
For example, firstly, it is operated with 4~20mA. Secondly, 10mA comes in. Finally, the signal is immediately falling down to 3mA without falling down the current continually. In this case, relevant channels will retain the output value of 10mA.
(2) When this function is set, digital output value related with actual range of analog input is only shown.
Refer to the actual range of the analog from "chapter 13.3".
(3) For the detailed usage, refer to chapter 13.5.5 Hold Last Value Function.
(4) Setting of hold last value is as below.
_01_AD0_HOLDVAL (%UX0.1.320): Input channel 0 hold effective conversion value setting.
_01_AD1_HOLDVAL (%UX0.1.321): Input channel 1 hold effective conversion value setting.

	Bit15	Bit4	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
%UW0.1.20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Input channel 0	Input channel 1

Channel setting
Bit On (1) : Permission
Bit Off (0) : Prohibition

18) Output status setting
(1) When the motion controller is stopped, set the analog output status
(2) When the output status setting is not specified, output the previous value.

	Bit15 ~ Bit2	Bit1 ~ Bit0
%UW0.1.21	Don't care	Channel 0 status
%UW0.1.22	Don't care	Channel 1 status

Output channel status setting (2 Bit)
00 : Previous value
01 : Min value
02 : Mid value
03 : Max value

Variable	Memory	content	Setting
_01_DA0_OUTSTAT	%UW0.1.21	Channel 0 output status setting	Input data type setting (bit) → 00 : Previous value → 01 : Minimum value → 10 : Midian values → 11 : Maximum value
_01_DA1_OUTSTAT	%UW0.1.22	Channel 1 output status setting	

19) Interpolation method setting
   (1) Shows the setting of the interpolation method of each channel.
   (2) _01_DA0_INTPMTHD (%UB0.1.46): Output channel 0 interpolation method setting
   _01_DA1_INTPMTHD (%UB0.1.47): Output channel 1 interpolation method setting

	Bit15	Bit4	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
%UW0.1.23	Don't Care						Output channel 1		Don't Care						Output channel 0	

Interpolation method setting (2 Bit per channel)
0 : Prohibition
1 : Linear
2 : S-Curve

20) Interpolation time setting
   (1) Shows the setting of interpolation time of each channel.
   (2) _01_DA0_INTPTIME (%UB0.1.48): Output channel 0 interpolation time setting.
   _01_DA1_INTPTIME (%UB0.1.49): Output channel 1 interpolation time setting.

	Bit15	Bit4	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
%UW0.1.24	Don't Care						Output channel 1		Don't Care						Output channel 0	

Interpolation time setting (2 Bit per channel)
0 : 10[ms]
1 : 100[ms]
2 : 1[s]
3 : 60[s]

21) Interpolation operation value
   (1) Shows the interpolation operation value of each channel.
   (2) _01_DA0_INTPVAL (%UW0.1.25): Output channel 0 interpolation operation value.
   _01_DA1_INTPVAL (%UW0.1.26): Output channel 1 interpolation operation value.

	Bit15	Bit4	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
%UW0.1.25	Input channel 0 operation value															
%UW0.1.26	Input channel 1 operation value															

22) Setting error code

(1) Shows the error code of each channel.

(2) If it is normal, the error code is 0.

(3) _01_SETTINGERR (%UW0.1.27): Error information.

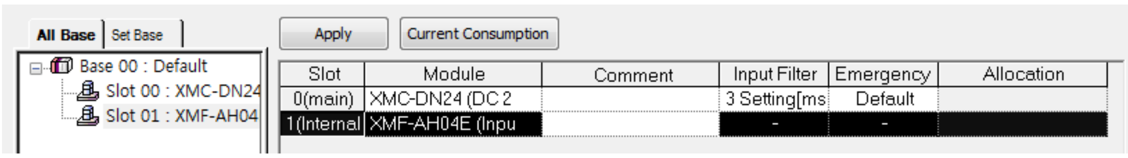
	Bit15	Bit4	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
%UW0.1.27	Setting error information															

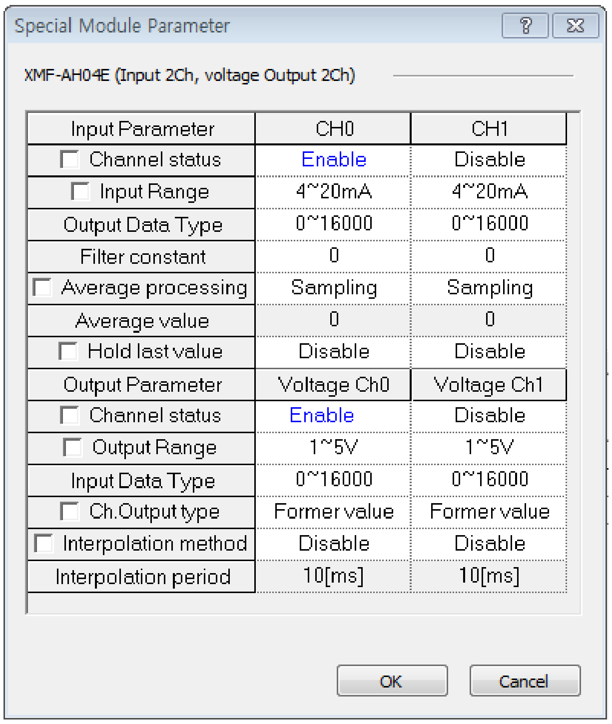
Type	Error code (Decimal)	LED sign	Description	Priority	Note
Error	10#	AD LED Flickering 1s intervals	Setting error of input channel range	1	#: channel number (Channel 0~1)
	20#		Setting error of input channel filter value	2	
	30#		Setting error of input channel average value	3	
	40#	DA LED Flickering 1s intervals	Setting error of output channel range	4	
	50#		Setting error of output channel digital input value range	5	
	60#		Output channel interpolation method setting error	6	

(4) When errors of two or more are caused, the high priority error code is saved. And when the same error code is caused in channels of two or more, the error code of low channel number is saved preferentially.

13.11 Example program

1) I/O Parameters window

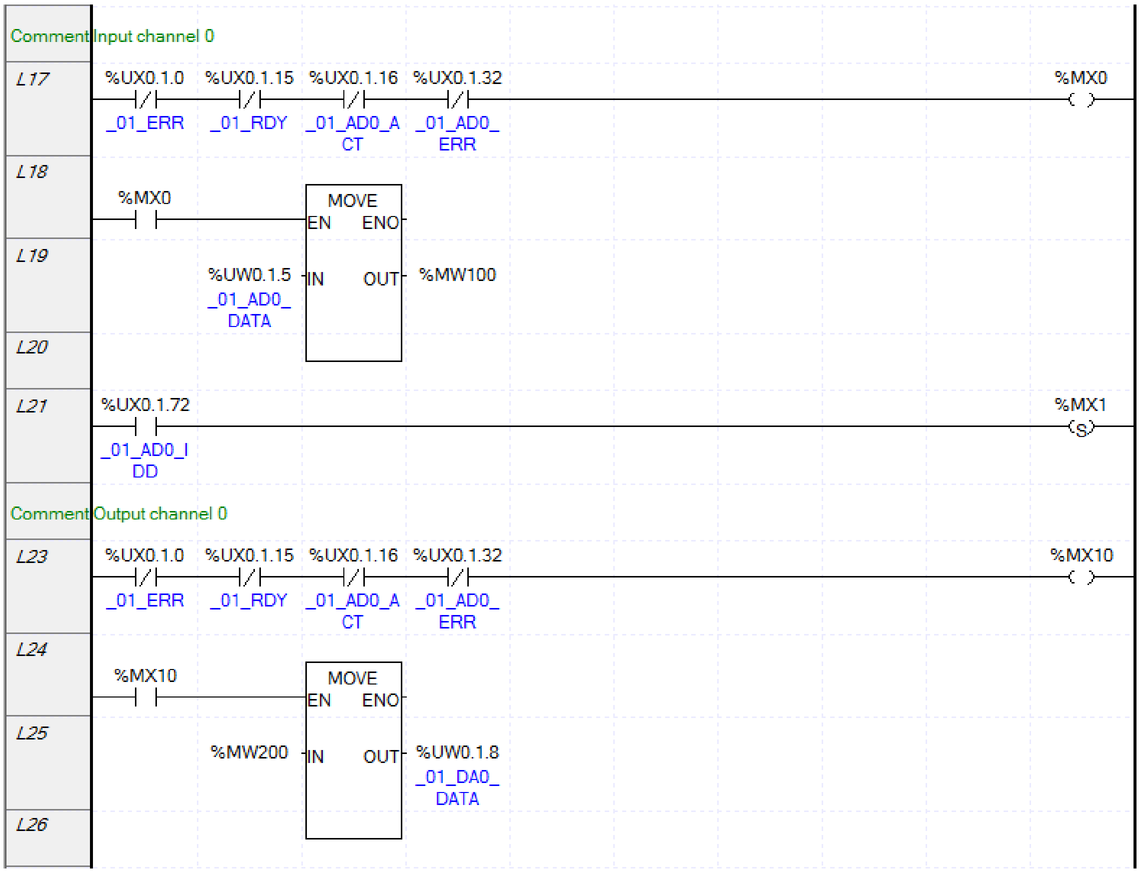




Input Parameter	CH0	CH1
Channel status	Enable	Disable
Input Range	4~20mA	4~20mA
Output Data Type	0~16000	0~16000
Filter constant	0	0
Average processing	Sampling	Sampling
Average value	0	0
Hold last value	Disable	Disable
Output Parameter	Voltage Ch0	Voltage Ch1
Channel status	Enable	Disable
Output Range	1~5V	1~5V
Input Data Type	0~16000	0~16000
Ch.Output type	Former value	Former value
Interpolation method	Disable	Disable
Interpolation period	10[ms]	10[ms]

(1) The input channel 0 is set with operation channel and the range is set with 4~20mA.
(2) The voltage output channel 0 is set with operation channel and the range is set with 1~5V.

2) Example of program



(1) Input Program Example

(가) The '%MX0' is on while the module normally operates.

%UX0.1.0 (module error)= OFF

%UX0.1.15 (module ready)= ON

%UX0.1.16 (channel 0 run) = ON

% UX0.1.32 (Channel 0 error)= OFF

(나) When the '%MX0' is on, conversion value (%UW0.1.5) of CH0 is moved to the '%MW100'.

(다) If the error is caused on CH0, %UX0.1.72 (CH0 disconnection) will be on and the '%MX1' will be on.

(2) Output Program Example

(가) The '%MX10' is on while the module normally operates.

%UX0.1.0 (module error)= OFF

%UX0.1.15 (module ready)= ON

%Ux0.1.24 (voltage output channel 0 run) = ON

%Ux0.1.40 (voltage output channel 0 error)= OFF

(나) When the '%MX10' is on, voltage channel 0 output status (%UX0.1.112) is on, and the output is permitted.

(다) If '%MX10' is on, '%MW200' data is moved to voltage channel 0 output value (%UW0.1.8) and then it is output.

13.12 Failure Diagnostics

The chapter describes diagnostics and measures method in case of any trouble occurs during use of built-in analog module.

13.12.1 LED Indication by Errors

Built-in analog module has two LEDs and it is possible to check whether it had any error with the indication of LEDs.

Item	Normal Status	When CH is disconnected	When parameter setting is error
AD LED	ON	Flickering 1s intervals	Flickering 1s intervals (Input parameter setting error)
DA LED	ON	-	Flickering 1s intervals (Output parameter setting error)
Module Operation	Normal operation Operation of all functions	Operation of all functions Shows minimum input value.	Operation of all functions (Operation as parameter default)
Corrective Actions	-	Check wiring	Check parameter setting

13.12.2 Check the Built-in Analog Module

The status of built-in analog module can be checked through the system monitor of XG5000.

1) Execution sequence

You can run it in one of the ways below.

(1) [Monitor] → [System Monitor] → Click the right button of mouse on the painting of module. → [Module Information]

(2) [Monitor] → [System Monitor] → Double click the painting of module

(3) [Monitor] → [Special Module Monitor] → Built-in Analog Module Selection → Click the module information

(4) [Online] → [I/O Information] → Built-in Analog Module Selection → Click the details

(5) [Online] → [I/O Information] → Built-in Analog Module Double click

2) CPU module information

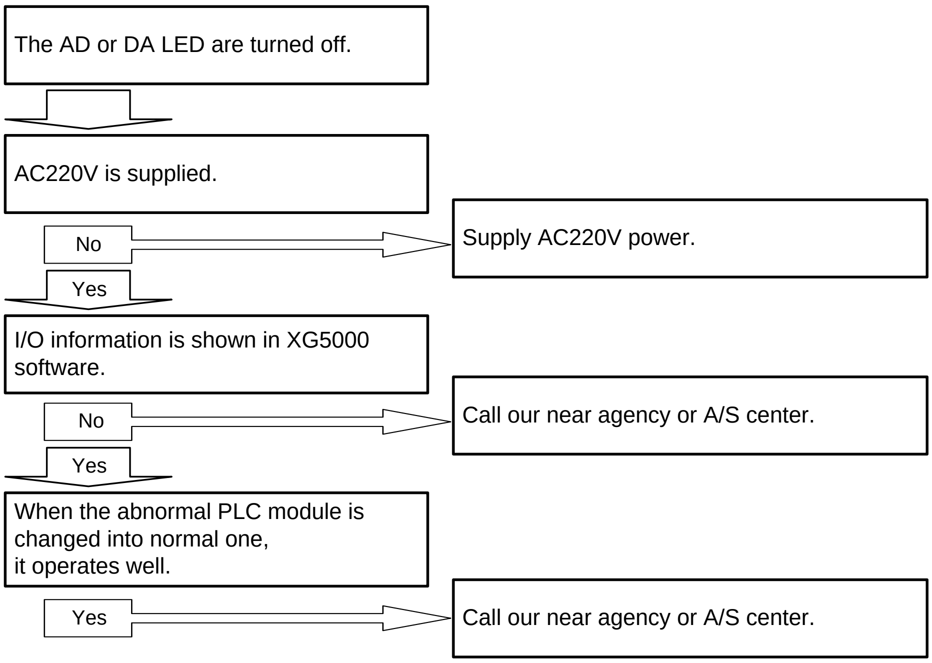
(1) OS version: OS version of module is shown.

(2) OS date: The OS prepared date of module is shown.

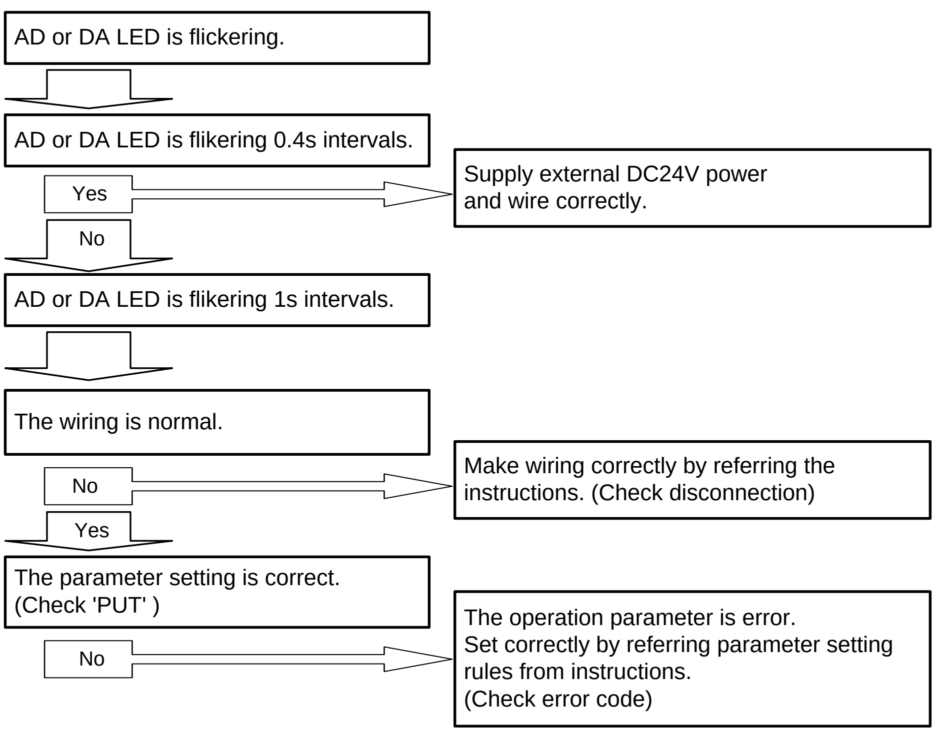
(3) Module Status: The present error code is shown.

13.12.3 Failure Diagnostics and Action method

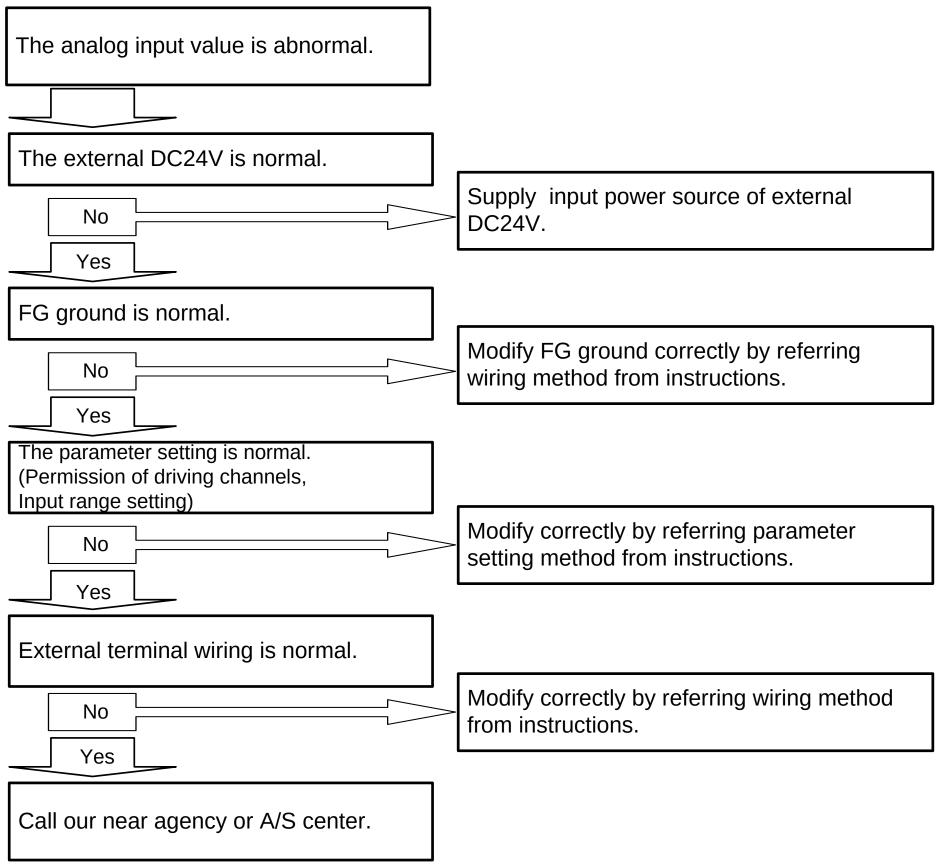
1) AD, DA LEDs are off



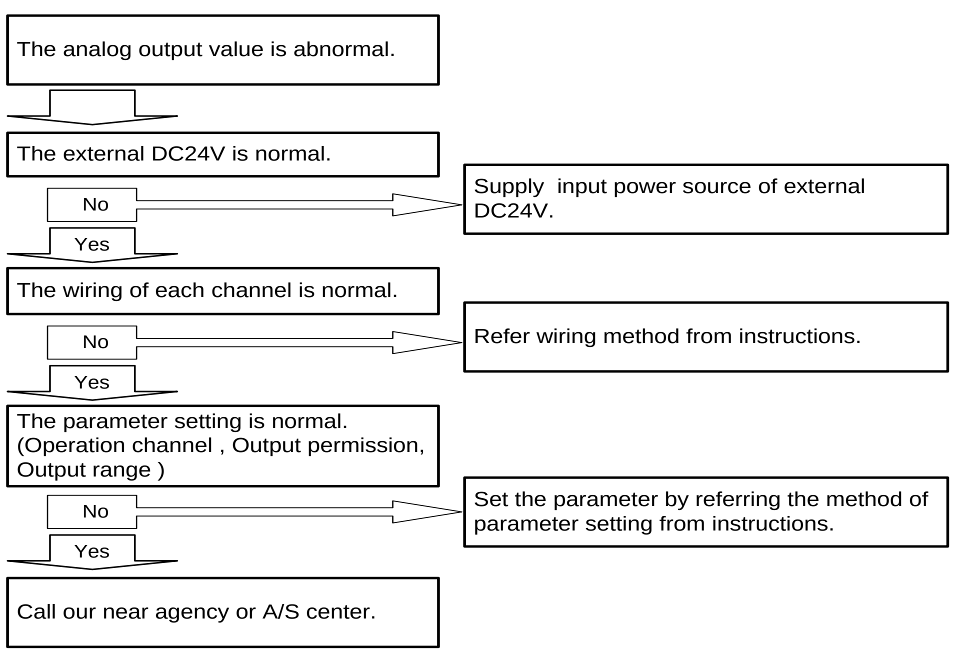
2) LED is flickering



3) The analog input value is abnormal.



4) The analog output value is abnormal.



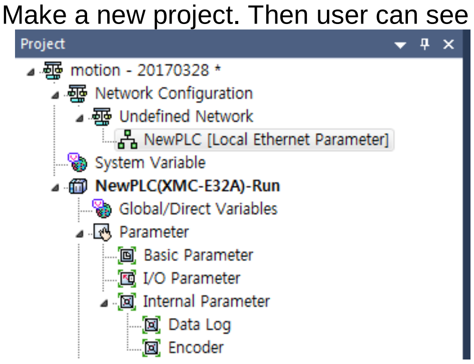
Chapter 14 Local Ethernet Function

14.1 Local Ethernet Function

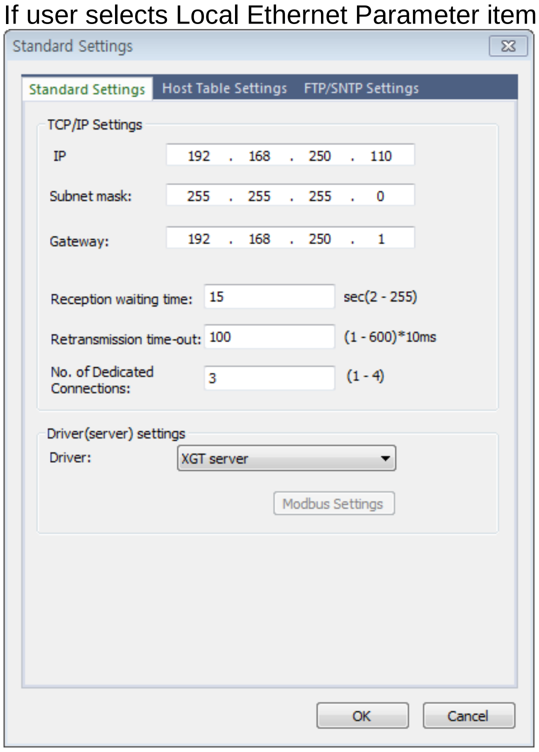
Motion controller can carry out the functions of Ethernet server using internal local Ethernet function.

14.1.1 Local Ethernet Parameter Settings

Make a new project. Then user can see Local Ethernet Parameters as shown below figure.



If user selects Local Ethernet Parameter item, Local Ethernet Parameter setting window will be displayed.



To use the Local Ethernet function, user should set the parameters.

(1) TCP/IP Setting

Classification	Description
IP address	Specify the IP Address of the applicable motion controller.
Subnet mask	Value necessary to check if destination station is on the same network of the applicable station.
Gateway	IP address of Gateway or Router to transmit/receive data through the public network or a network different from the network where the applicable FEnet module is included.
Reception waiting time	If there is no request during the specified time from the host PC or HMI (Human Machine Interface) connected for dedicated communication, it will end the dedicated service connection regardless of normal ending procedures supposing that the higher level system is with error. This time is used in dedicated service to reset the channel when any error occurs on the destination station or the cable is disconnected. (available range is 2 ~ 255 sec)
Retransmission time-out (10 ㎳)	It is the time it takes CPU to send a data to the destination station if the destination station does not answer the data sent by applicable station during setting time. (Applicable station considers it as a data missing.) (available range is 10 ㎳ ~ 6000 ㎳) * **Note**: Retransmission time-out should be set depending on the network situation. If the setting time is too long, it takes a long time to resend a data in case of data missing. This will deteriorate the network performance. But if the setting time is too short, there is a chance to make a frequent disconnection or increase the load to the network.
Number of dedicated connections	Number of TCP dedicated services accessible at a time. (Max.4)

(2) Driver (Server) setting

Classification	Description
XGT server	Set when operated as dedicated communication server (slave)
Modbus TCP/IP server	Set when operated as Modbus server driver (slave)

(3) Host table setting

Classification	Description
Enable host table	Access allowed to applicable module of IP address registered in host table. Unregistered client (IP address) is prohibited from connection when enabled.

(4) Available Device address

Device	Address	Size(Word)	Description
I	%IW0.0.0 ~ %IW127.15.3	8192	Available Read/Write/Monitor
Q	%QW0.0.0 ~ %QW127.15.3	8192	Available Read/Write/Monitor
M	%MW0 ~ %MW1048575	1048576	Available Read/Write/Monitor
U	%UW0 ~ %UW0.15.31	512	Available Read/Write/Monitor
F	%FW0 ~ %FW65535	65536	Available Read/Monitor
K	%KW0~%KW9125	9126	Available Read/Monitor

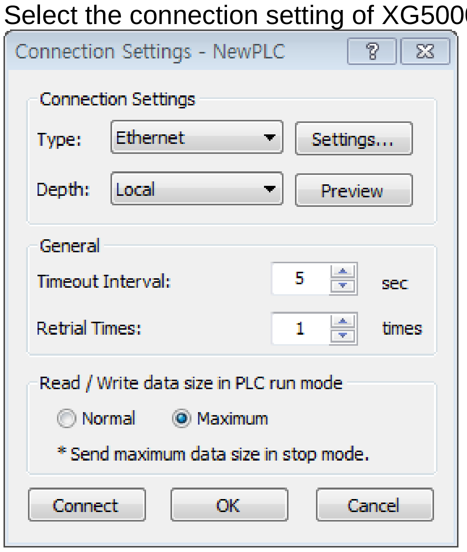
14.1.2 Local Ethernet Connection with XG5000

After finishing Local Ethernet Parameter settings, download the settings to the motion controller, then user can connect to XG5000.

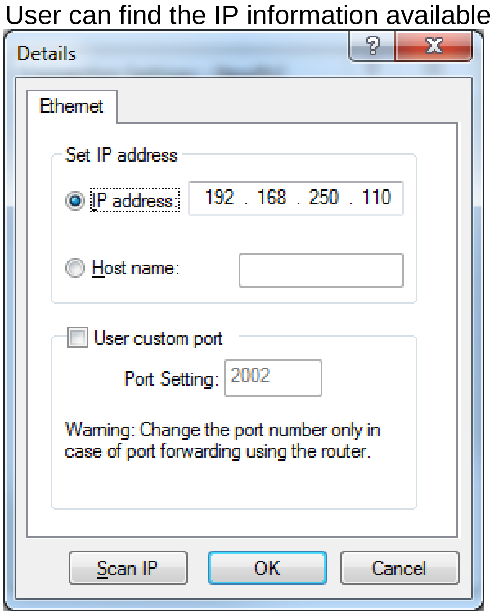
Select Online Settings and set the options as shown below figure.

(Notice: Motion controller's Ethernet port does not support the relay function about remote connection. Then, motion controller doesn't support the remote connection.)

Select the connection setting of XG5000. Then, select the options of connection option as below.

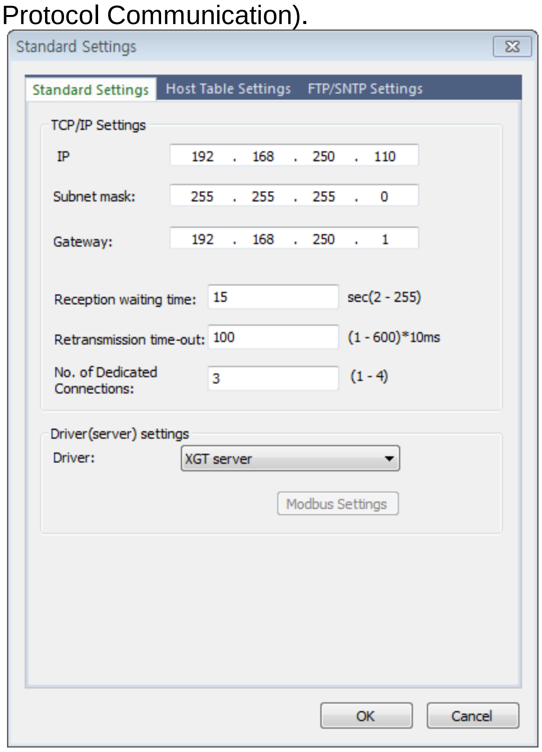


Click the setting button to specify Ethernet IP.   Click OK after specify the Ethernet IP set before.
User can find the IP information available now.



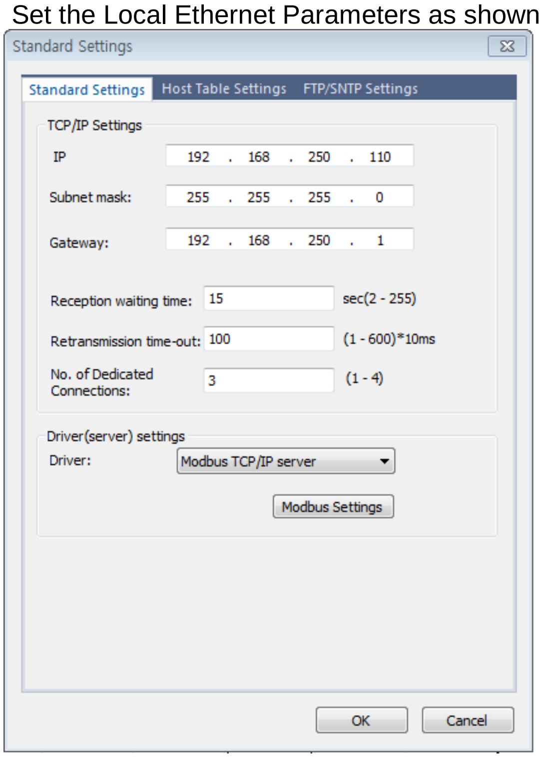
14.1.3 Local Ethernet Connection with XGT Server

Set the Local Ethernet Parameters as shown below figure. User can use it as a XGT Server (LS ELECTRIC dedicated Protocol Communication).

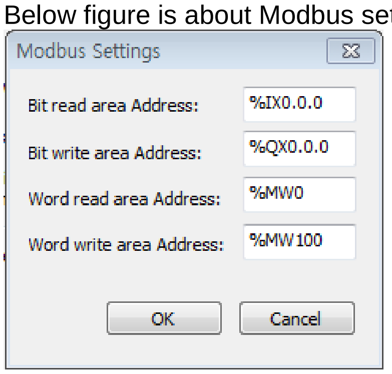


14.1.4 Local Ethernet Connection with TCP/IP Server

Set the Local Ethernet Parameters as shown below figure. User can use it as a Modbus server.



Below figure is about Modbus settings. .



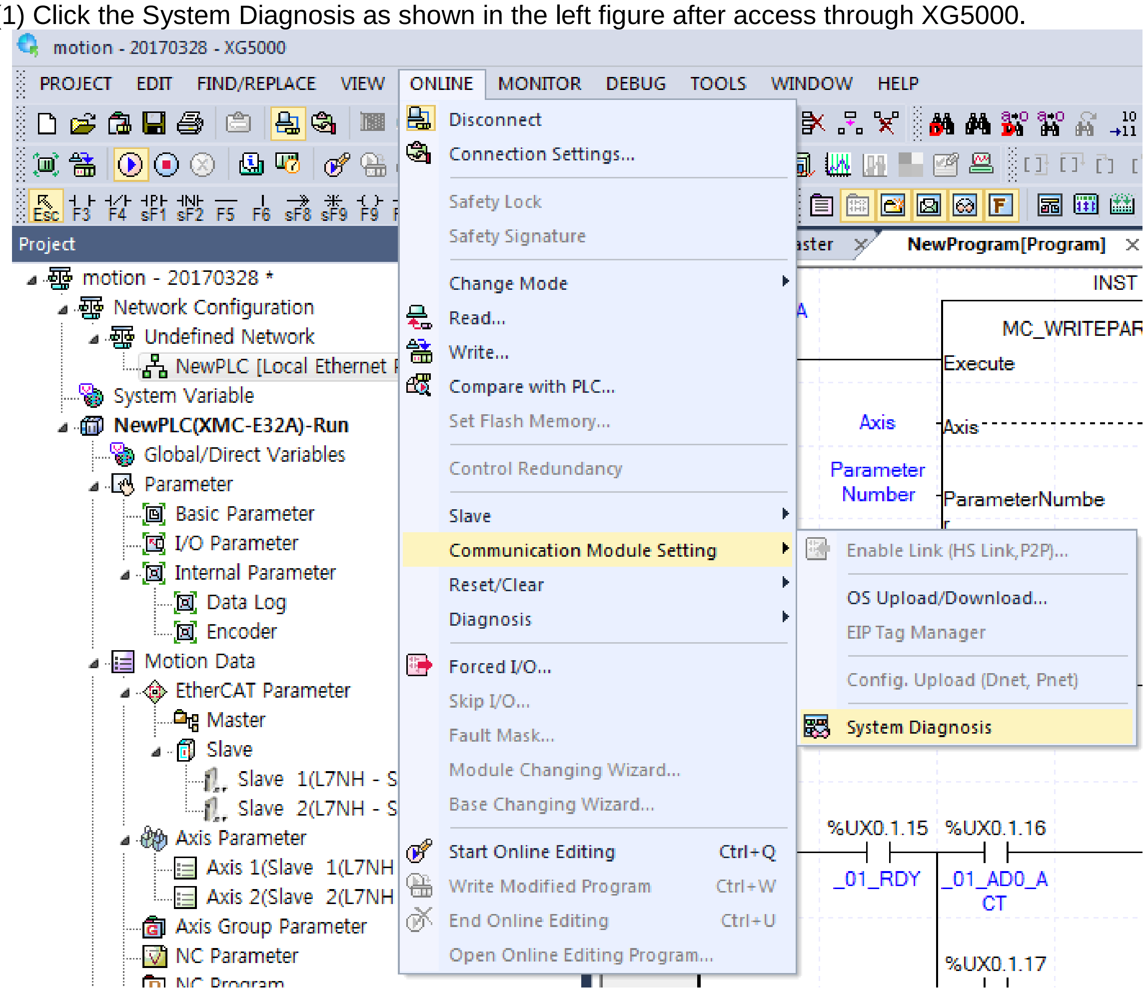
Note

1) Modbus TCP/IP server connection function allows RST packet transmission depending on the network condition.(TCP/IP protocol)
   So the user devices connecting to CPU module should have RST packet process.
2) Connection to user devices can be disconnected for retransmission time-out.
   For the retransmission timeout period, the connection is terminated after waiting twice as long as the previous waiting time after retransmission time setting value set by the user in the local Ethernet parameter(default value: 100)X10ms), the number of retransmissions(three times, twice the previous waiting time) and three transmissions.
   Retransmission time-out = retransmission time-out value(set in the Local Ethernet Parameter window) x 30ms
3) Too much Network loads can affect a scan time. So user should consider appropriate network loads for CPU scan time.

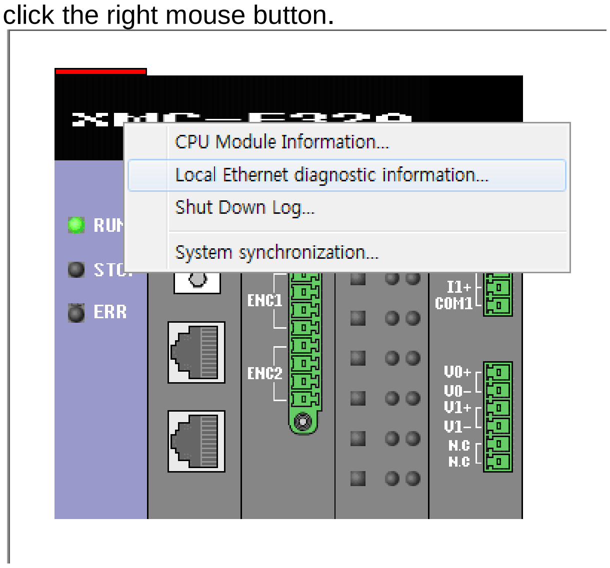
14.1.5 Local Ethernet Diagnosis Information Function

Motion controller provides local Ethernet diagnosis information function to monitor the status of local Ethernet.

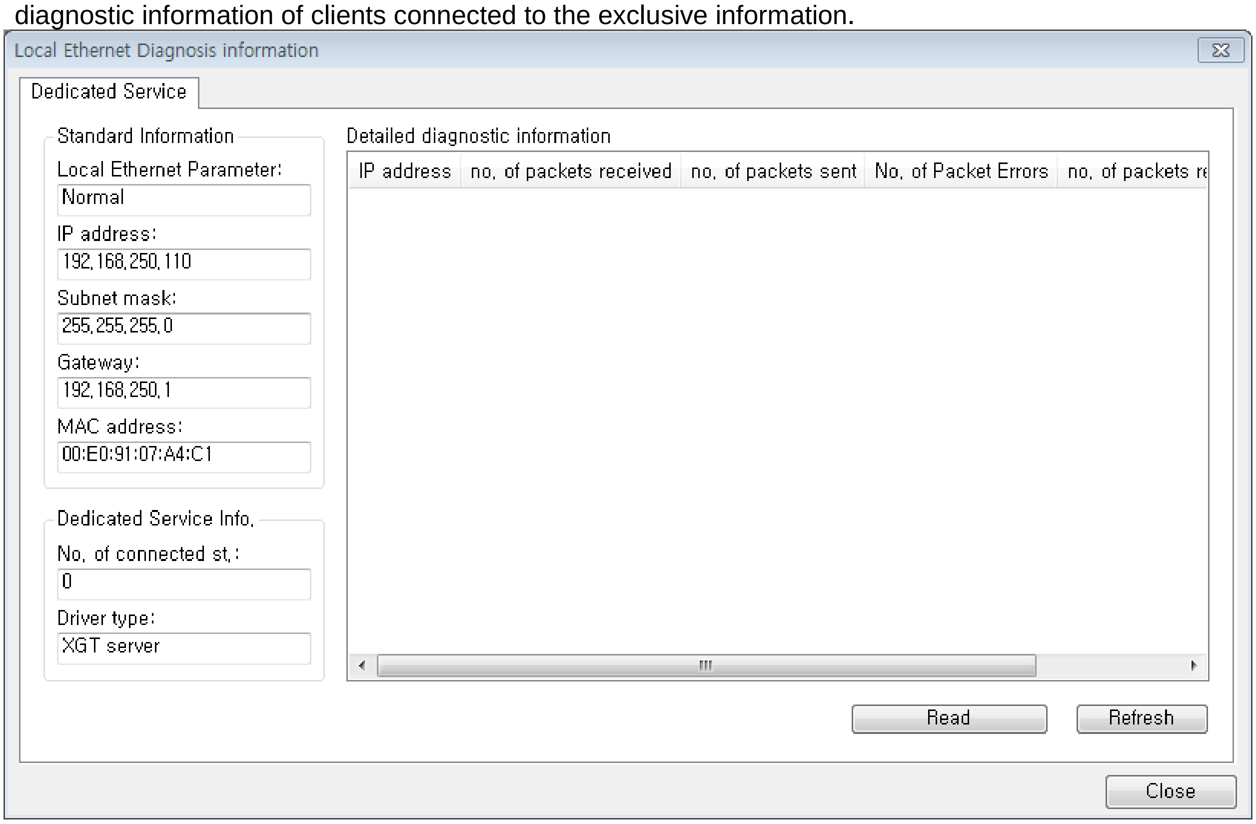
(1) Click the System Diagnosis as shown in the left figure after access through XG5000.



(2) Then, the current system is displayed as shown in the below figure. Put the mouse on the figure of the module and click the right mouse button.



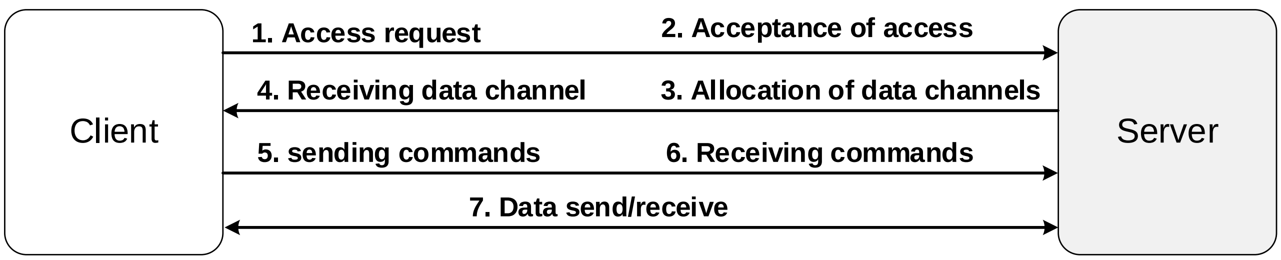
(3) If you click the loca1l Etnernet diagnostic information among the menus that occur when clicking the right mouse button, the local Etnernet diagnostic information window will be created as shown below. Through the local Ethernet information window, you can monitor the Ethernet basic information, exclusive service information and detailed diagnostic information of clients connected to the exclusive information.



14.2 FTP Server Functions

14.2.1 Outline

Motion controller supports the Transfer Protocol (File Transfer Protocol) to download the data log file from a remote site through built-in Ethernet port. The File Transfer Protocol is TCP/IP based protocol to be designed for file transfer and you can manage files in a remote site by using the File Transfer Protocol. The File Transfer Protocol that is divided into the server and the client transmits or receives files.

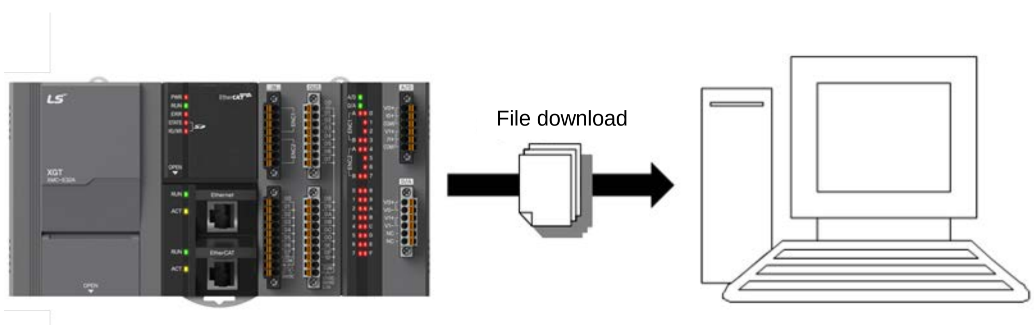


14.2.2 Support Functions

You can access to motion controller built-in FTP server through FTP client. After access, you can copy the data log file saved in the micro-SD card to the user’s PC where FTP client is installed or other devices. However, in terms of the FTP function, only the download function is provided to prevent arbitrary modification or changes of data log files through FTP.

(1) Read File (File download)

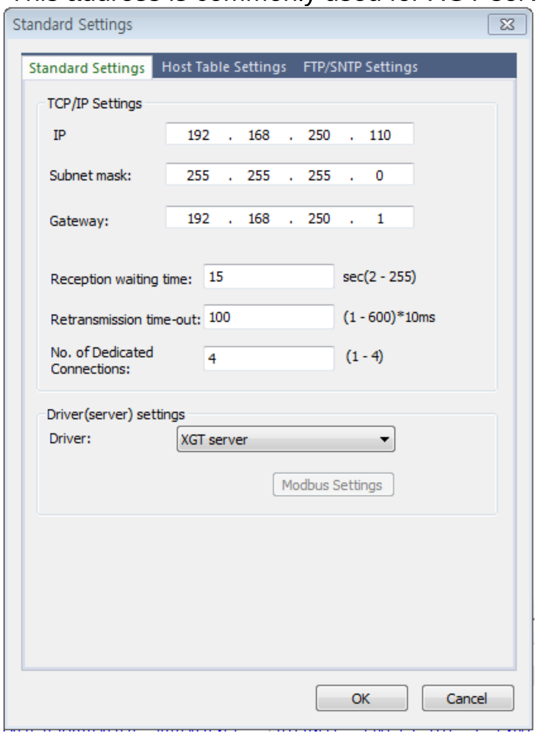
It is the function to import and save files to the devices with FTP client or PC (Personal Computer) from motion controller’s FTP server. The files are saved to the designated directory path of the devices with FTP client or PC.



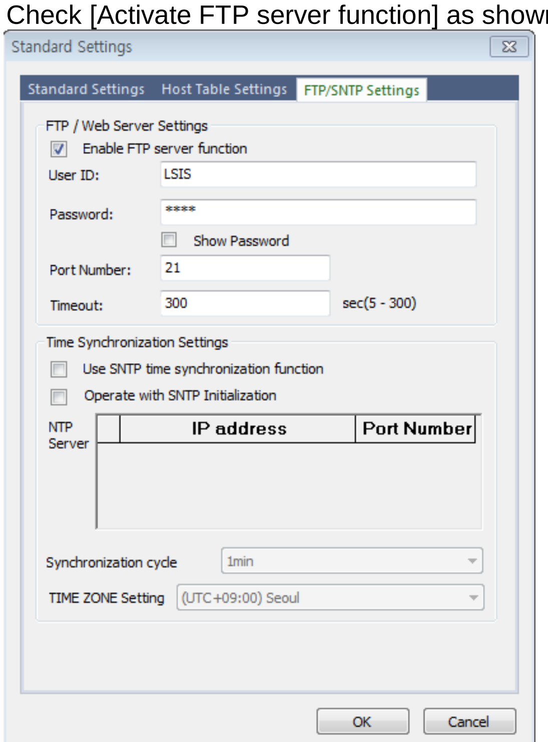
14.2.3 Setting FTP Server Parameters

You need to set parameters through XG5000 to use the FTP server function.

(1) Input the “TCP/IP setting” parameters in the window for setting FEnet basic.
  - Input the IP address, subnet mask, gateway, DNS server address.
  - This address is commonly used for XGT server, Modbus TCP/IP server, SNTP service, FTP service.

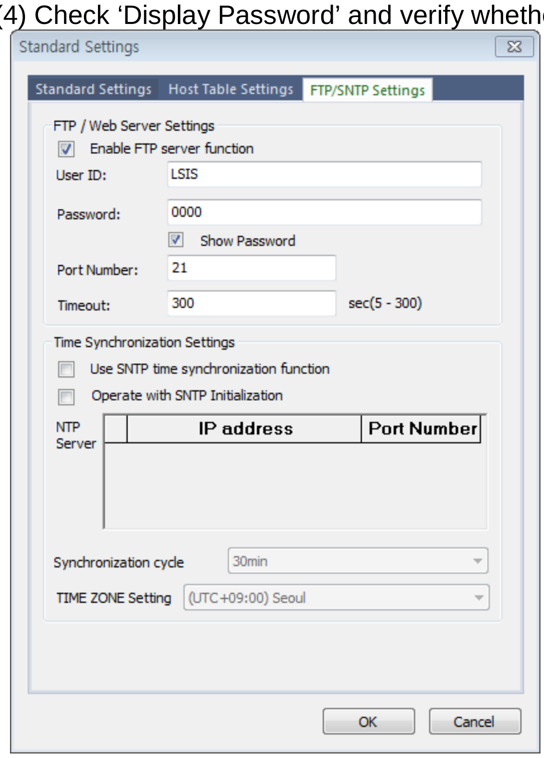


(2) Check [Activate FTP server function] as shown below figure.



(3) Enter the user ID and password to be used to access the FTP server.
- You can change the user ID and password through XG5000 only.

(4) Check 'Display Password' and verify whether the entered password is correct.



(5) Enter the port number. (Default: 21)

(6) Enter the timeout (Default: 20 sec)
- The timeout is the time required to disconneted automatically, if you do not use after connecting to the FTP server.

(7) If you press the OK button, setting the parameters to use FTP is completed.

(8) When you execute [Online] → [Write Parameter], the parameters are written in the motion controller.

Note

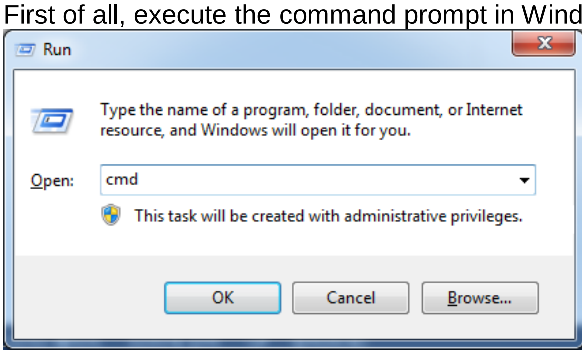
1. Unless you set the user ID and password, basic ID and password will be set initially
   - Default setting ID: LSIS
   - Default password: 0000
2. Rules for applying the user ID and password
   - You can enter the user ID and password that are composed of alphabetical characters and numbers but special characters are not available.
   - They must be case-sensitive and must not exceed the maximum of 8 digits.

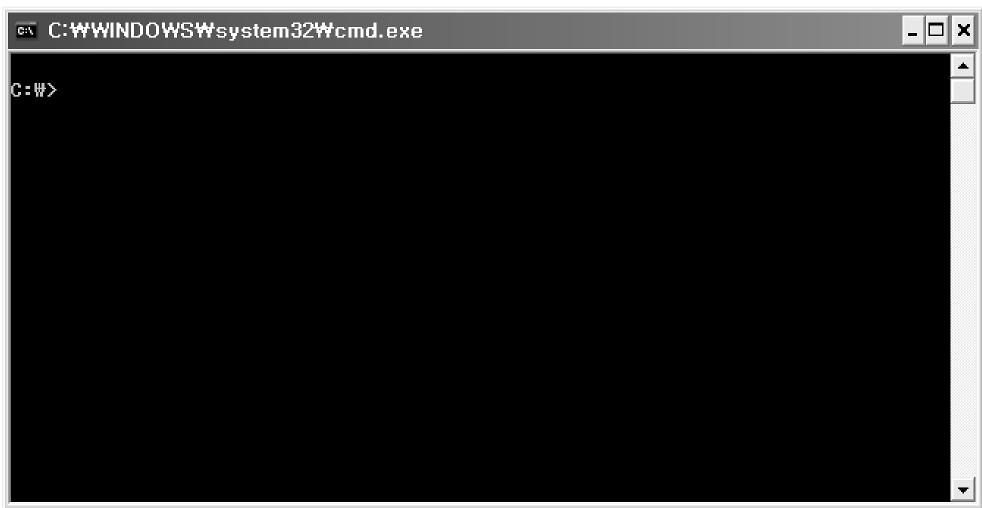
14.2.4 How to Access to the FTP Server

Just one user can access to the FTP server at a time so using Windows FTP client is recommended.

(1) How to use WINDOWS command prompt

(a) First of all, execute the command prompt in Windows

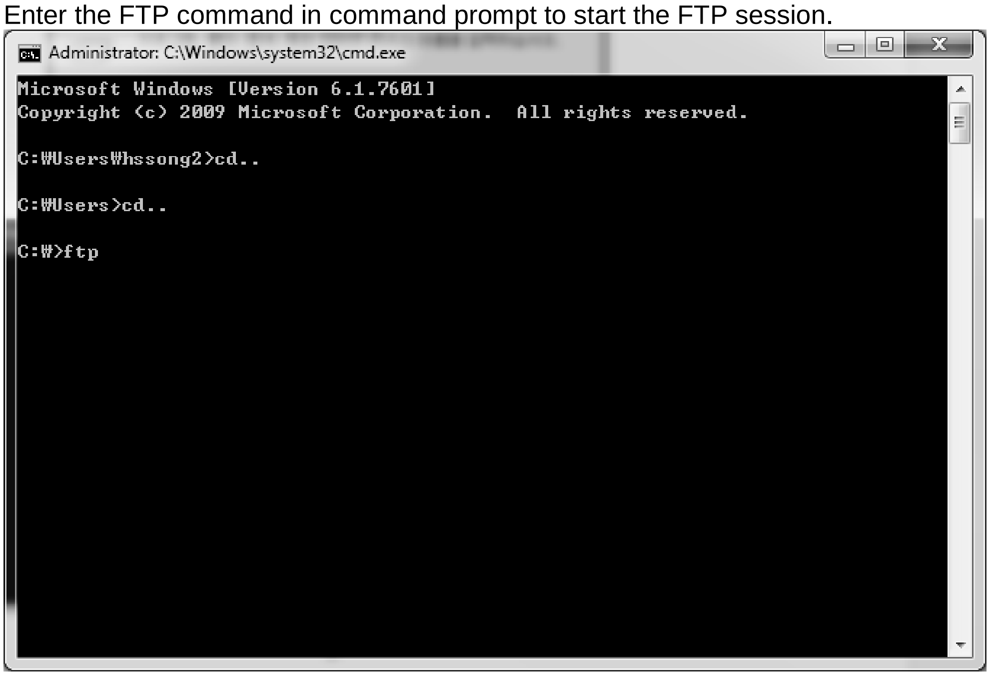




Note

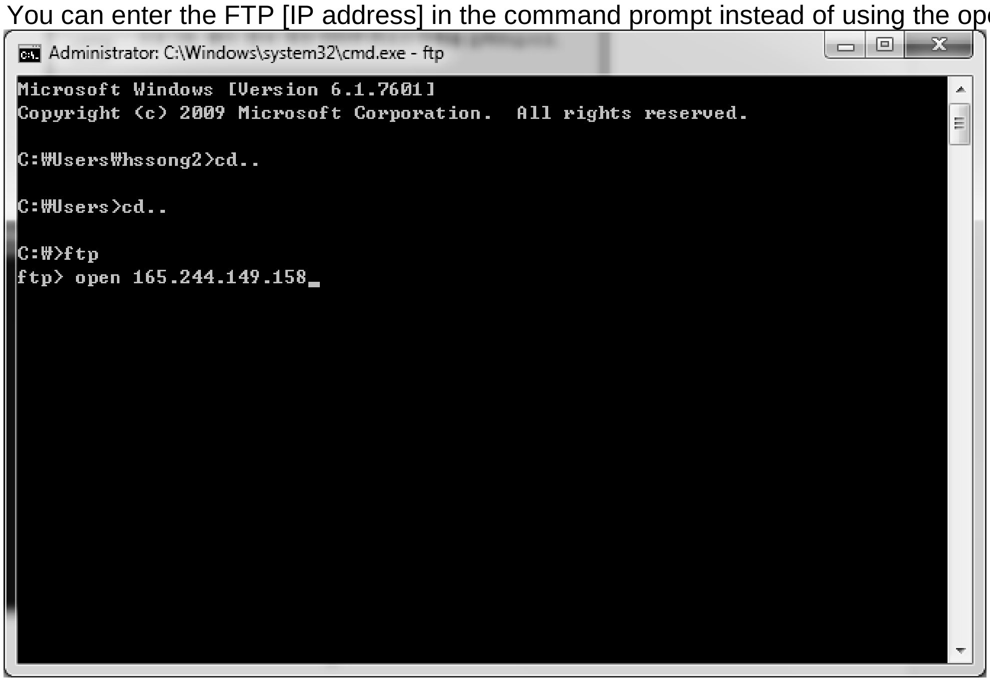
1. The compatibility with other commercial client programs other than Windows FTP client is not guaranteed.
2. It is not support the multiple sessions, only one user can access at the same time.
3. To execute the command prompt window, enter ‘cmd’ to the window or press [Start] → [All Programs] → [Auxiliary Programs] → [Command Prompt].
For more details on execution of command prompt, refer to the Windows manual.

(b) Enter the FTP command in command prompt to start the FTP session.

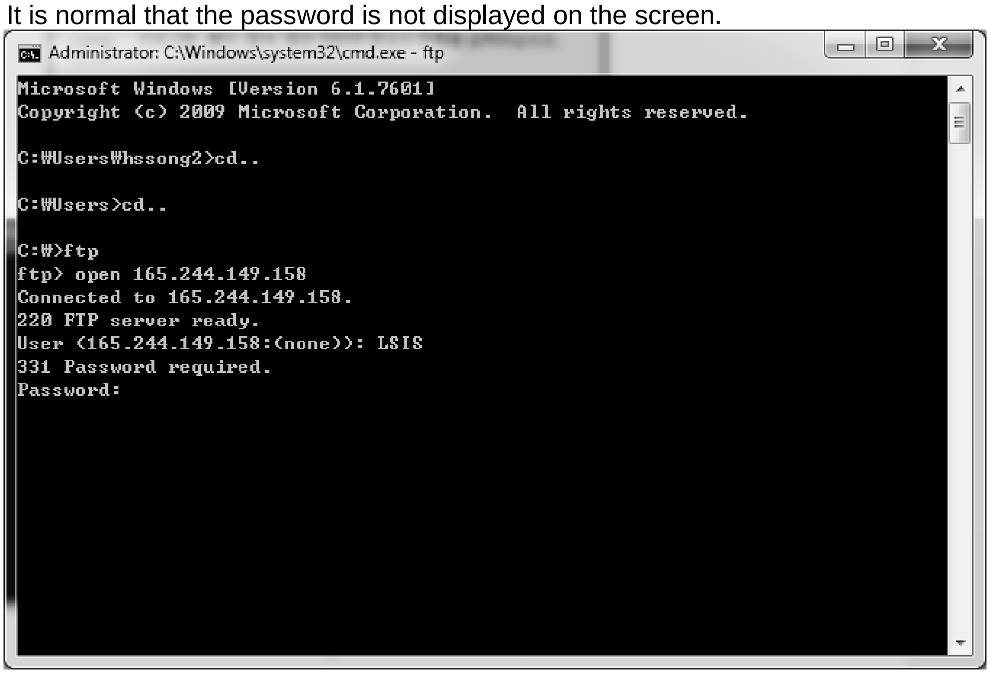


(c) Enter the ‘open [IP address]’to access to the FTP server

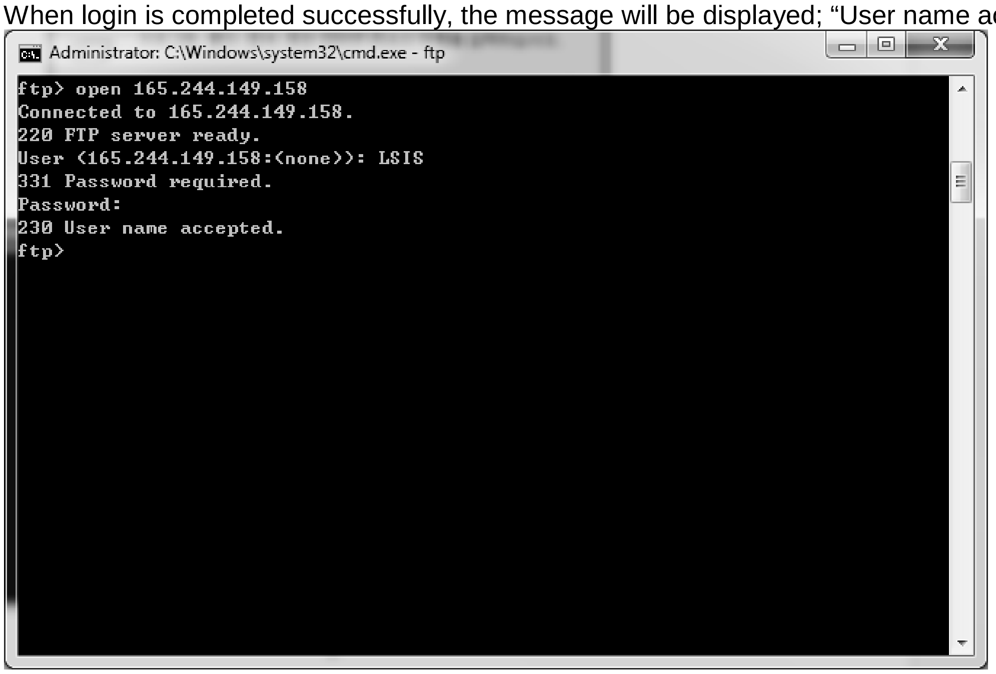
- You can enter the FTP [IP address] in the command prompt instead of using the open command.



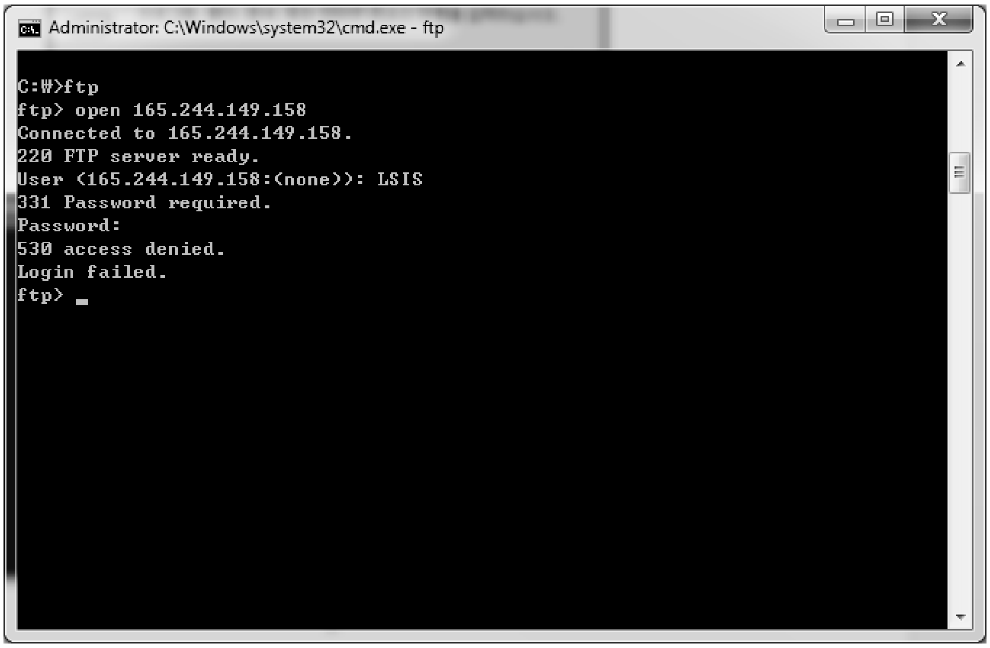
(d) Enter the user ID and password to access to the FTP server.
- It is normal that the password is not displayed on the screen.



(e) When login is completed successfully, the message will be displayed; “User name accepted.”

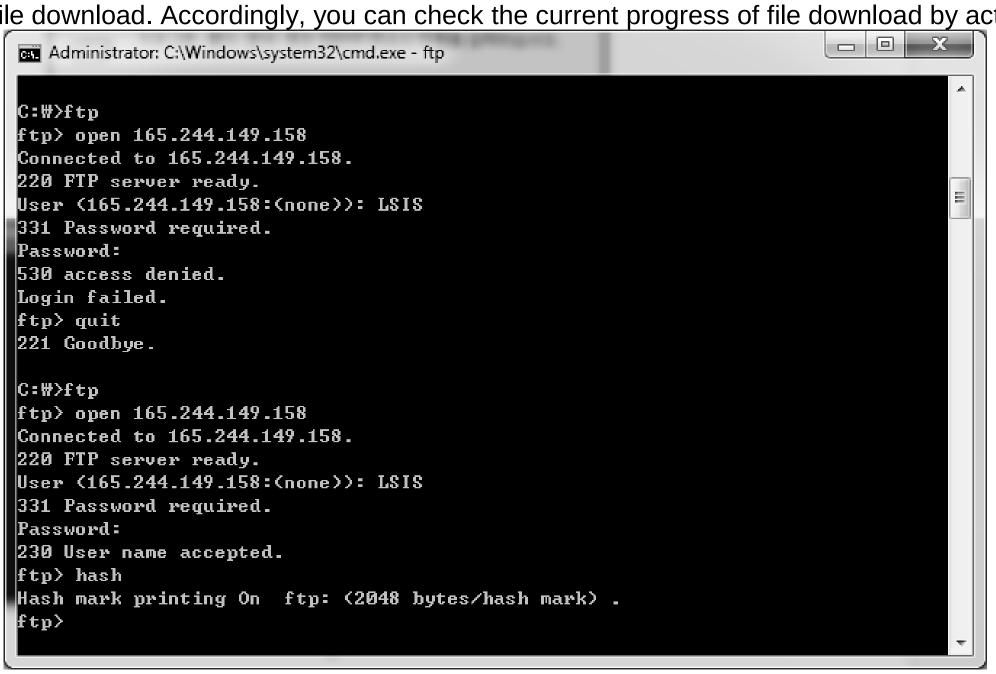


[Completion of FTP server access and login]



[Failure of FTP server login]

(f) In case you access to the FTP server through windows command prompts, you cannot see the progress status of file download. Accordingly, you can check the current progress of file download by activating the HASH function.



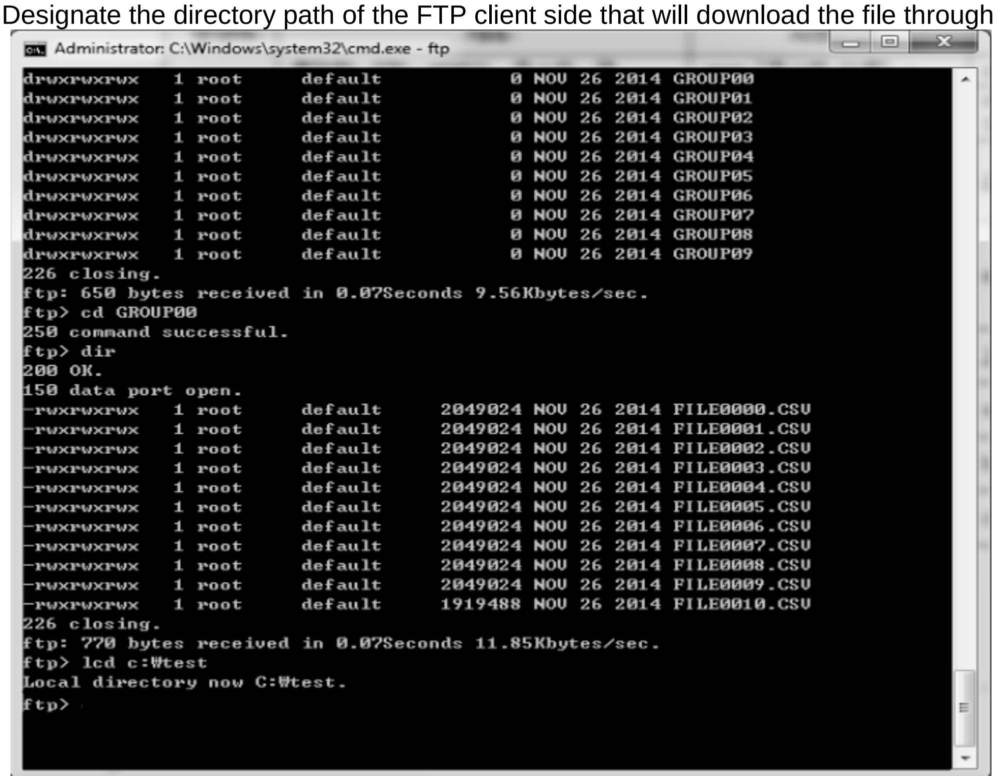
(g) You can view the directories and file lists that exist in the drive currently through the ‘DIR’ command.

```
Administrator: C:\Windows\system32\cmd.exe - ftp
Hash mark printing On  ftp: (2048 bytes/hash mark) .
ftp> dir
200 OK.
150 data port open.
drwxrwxrwx   1 root     default           0 NOV 20 2014 DATALOG
226 closing.
ftp: 65 bytes received in 0.09Seconds 0.76Kbytes/sec.
ftp> cd DATALOG
250 command successful.
ftp> dir
200 OK.
150 data port open.
drwxrwxrwx   1 root     default           0 NOV 26 2014 GROUP00
drwxrwxrwx   1 root     default           0 NOV 26 2014 GROUP01
drwxrwxrwx   1 root     default           0 NOV 26 2014 GROUP02
drwxrwxrwx   1 root     default           0 NOV 26 2014 GROUP03
drwxrwxrwx   1 root     default           0 NOV 26 2014 GROUP04
drwxrwxrwx   1 root     default           0 NOV 26 2014 GROUP05
drwxrwxrwx   1 root     default           0 NOV 26 2014 GROUP06
drwxrwxrwx   1 root     default           0 NOV 26 2014 GROUP07
drwxrwxrwx   1 root     default           0 NOV 26 2014 GROUP08
drwxrwxrwx   1 root     default           0 NOV 26 2014 GROUP09
226 closing.
ftp: 650 bytes received in 0.37Seconds 1.75Kbytes/sec.
ftp> _
```

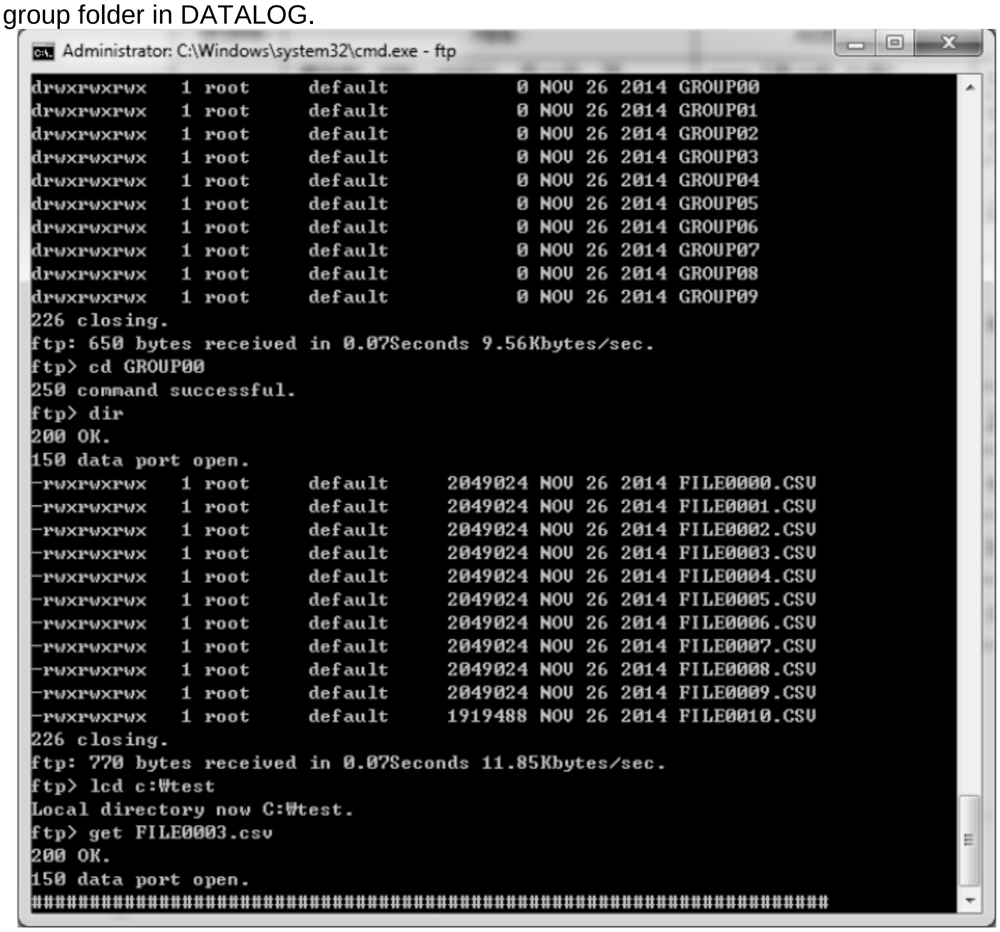
(h) You can go into the lower folder through the ‘cd [Folder name]’command. If you execute the ‘DIR’ command again after going into the lower folder, only the files ahasnd folder lists that exist in the lower folder will be displayed.

```
Administrator: C:\Windows\system32\cmd.exe - ftp
ftp> cd DATALOG
250 command successful.
ftp> dir
200 OK.
150 data port open.
drwxrwxrwx   1 root     default           0 NOV 26 2014 GROUP00
drwxrwxrwx   1 root     default           0 NOV 26 2014 GROUP01
drwxrwxrwx   1 root     default           0 NOV 26 2014 GROUP02
drwxrwxrwx   1 root     default           0 NOV 26 2014 GROUP03
drwxrwxrwx   1 root     default           0 NOV 26 2014 GROUP04
drwxrwxrwx   1 root     default           0 NOV 26 2014 GROUP05
drwxrwxrwx   1 root     default           0 NOV 26 2014 GROUP06
drwxrwxrwx   1 root     default           0 NOV 26 2014 GROUP07
drwxrwxrwx   1 root     default           0 NOV 26 2014 GROUP08
drwxrwxrwx   1 root     default           0 NOV 26 2014 GROUP09
226 closing.
ftp: 650 bytes received in 0.07Seconds 9.56Kbytes/sec.
ftp> cd GROUP00
250 command successful.
ftp> dir
200 OK.
150 data port open.
-rwxrwxrwx   1 root     default     2049024 NOV 26 2014 FILE0000.CSV
-rwxrwxrwx   1 root     default     2049024 NOV 26 2014 FILE0001.CSV
-rwxrwxrwx   1 root     default     2049024 NOV 26 2014 FILE0002.CSV
-rwxrwxrwx   1 root     default     2049024 NOV 26 2014 FILE0003.CSV
-rwxrwxrwx   1 root     default     2049024 NOV 26 2014 FILE0004.CSV
-rwxrwxrwx   1 root     default     2049024 NOV 26 2014 FILE0005.CSV
-rwxrwxrwx   1 root     default     2049024 NOV 26 2014 FILE0006.CSV
-rwxrwxrwx   1 root     default     2049024 NOV 26 2014 FILE0007.CSV
-rwxrwxrwx   1 root     default     2049024 NOV 26 2014 FILE0008.CSV
-rwxrwxrwx   1 root     default     2049024 NOV 26 2014 FILE0009.CSV
-rwxrwxrwx   1 root     default     1919488 NOV 26 2014 FILE0010.CSV
226 closing.
ftp: 770 bytes received in 0.07Seconds 11.85Kbytes/sec.
ftp> _
```

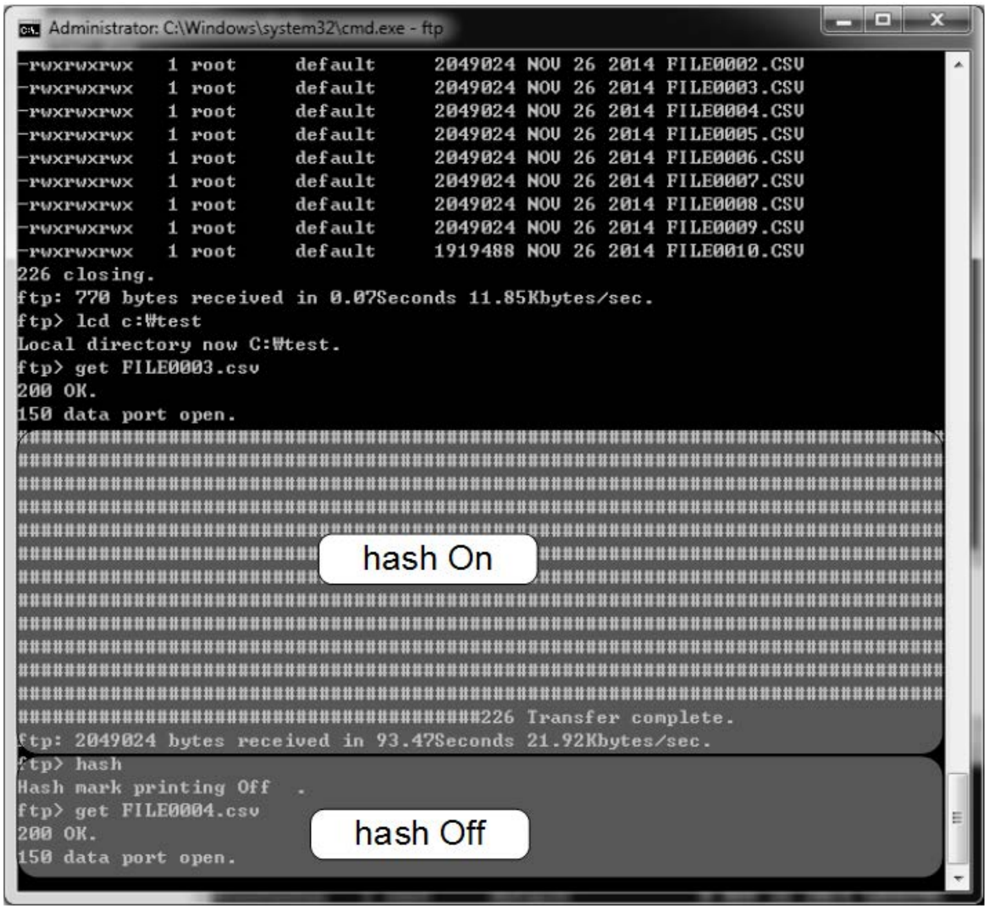
(i) Designate the directory path of the FTP client side that will download the file through the 'lcd' command.



(j) Select the file to be imported through the 'get' command and download it. At this time, the file is located in the sub-group folder in DATALOG.



(k) When the HASH function is activated and deactivated, the transmission status is shown in as below.



(2) FTP server command list

The windows FTP provides the below commands basically. You can check the further commands through '?' commands. There are also unserviceable functions to protect data log files so refer to the below list.

Commands	Operations	Commands	Operations
?	Displaying the available commands	trace	Setup/cancellation of packet trace
bye	Termination and end of the ftp session	type	Setting the file transfer type
cd	Changing remote working directory	user	Transmission of the new user information
close	Termination of the ftp session	verbose	Setup/cancellation of verbose mode
open	Connection to the remote ftp	quote	Sending random ftp commands
prompt	Executing interactive questions to multiple commands	recv	Receiving files
put	Sending one file (Not available)	dir	Enumerate the contents of remote directories
pwd	Printing the remote computer's working directory	disconnect	Termination of the ftp session
quit	Termination and end of the ftp session	get	Receiving files
lcd	Changing the local working directory	glob	Setup/cancellation of meta character extension of local file names
literal	Sending random ftp commands	hash	Setup/cancellation of '#' printing for the transmitted buffer
ls	Enumerate the contents of remote directories	help	Printing the local HELP information
status	Viewing the current status	cd ..	Move to the upper directory
rmdir	Remove the remote computer's directory (Not available)	mkdir	Make the remote computer's directory (Not available)
rename	Change file name (Not available)	delete	Delete the remote computer's file (Not available)
send	Send a file (Not available)		

(3) Command Usage

Commands	Description	Operations	Example
open	Attempting to access to the server by entering the specific FTP server's host name or IP.	open [host name]	open LSISHOST
		open [IP address]	open 166.0.1.254
dir	Showing the whole files saved to the basic unit's SD card with the file information	dir [drive volume:\]	dir B:\
get	Reading the specific file from the basic unit's SD card	get [File path and file name to be read from the server]	get LSIS.CSV
ls	Showing only the names of files saved the SD card of the basic unit	ls [drive volume:\]	ls B:\
quit	Braking and disconnecting the FTP server and FTP session	quit	quit
bye	Braking and disconnecting the FTP server and FTP session	Bye	bye
cd ..	Moving to the upper directory from the current one.	cd ..	cd ..

Note

1. You need to distinguish ASCII from Binary command depending on the file extension. If you transmit the file with a wrong mode, the file will not work properly.
   (a) File extension names using ASCII: html, htm, txt, cgi, pl, php, phtml, php3, sql, c, ph, py, etc.
   (b) File extension names using Binary: gif, jpg, swf, png, exe, asf, wmv, zip, rar, gzip, tar, gz, etc.
2. If you download the network setting parameters when accessing to the FTP server, the current download will stop and serious errors may occur in the relevant file so you cannot open it in the PC.
   Accordingly, if possible, you are recommended to disconnect the FTP server when downloading the network setting parameters.

(4) FTP response code list

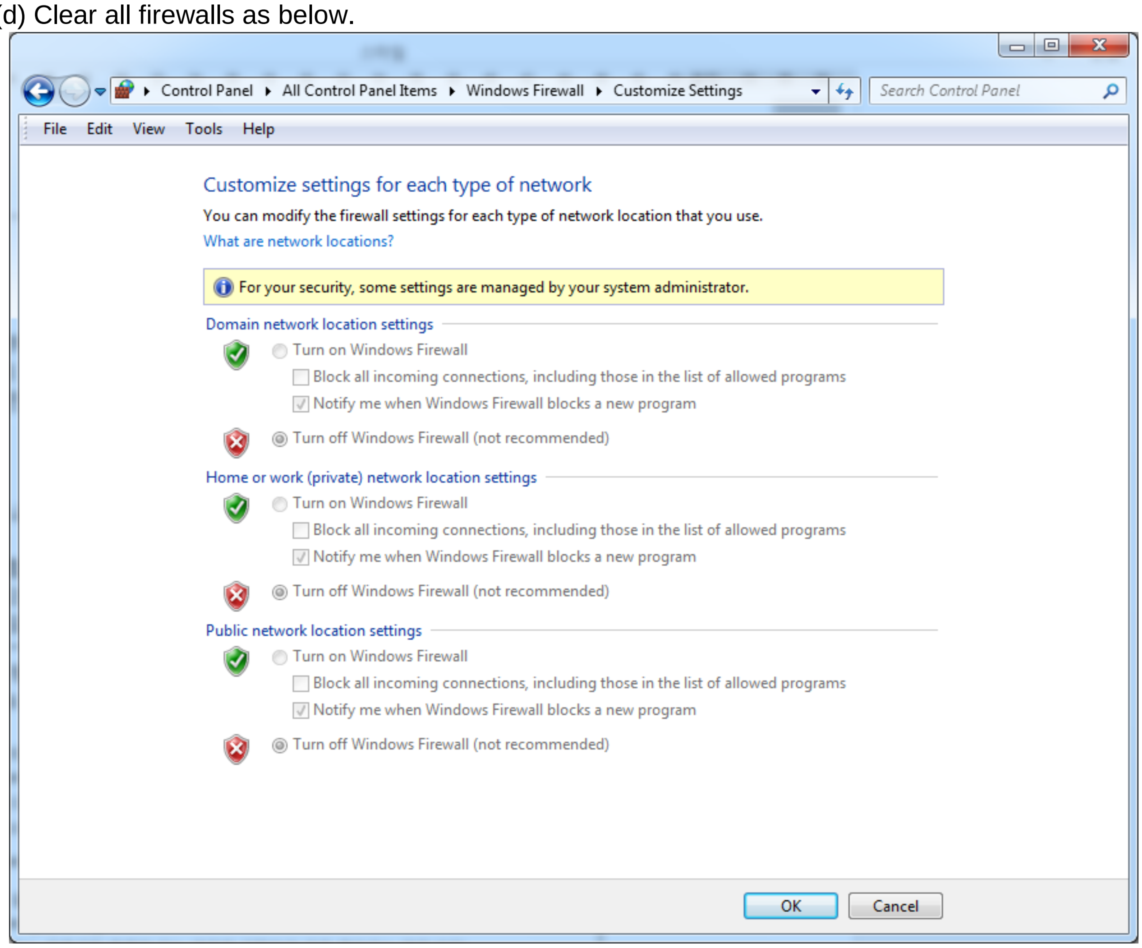
Response codes	Description
150	File status okay; about to open data connection
200	Command okay
202	Command not implemented, superfluous at this site
211	Type: ASCII, Structure: FILE, Mode: Stream
215	UNIX Type: L8 Version: Nucleus-ftpd
220	Nucleus FTP Server (Version 1.7) ready
221	Service closing control connection. Logged out if appropriate.
226	Closing data connection. Requested file action successful.
230	User logged in, proceed
250	Requested file action okay, completed.
331	User name okay, need password
500	Syntax error, command unrecognized.
502	Command not implemented. The server does not support this command.

14.2.5 Firewall Setting

When you access to the FTP server through Windows command prompts, FTP access may not be smooth since the FTP access is applied. When you have bad access, cancel a firewall or apply exception handling.
If the FTP access is not smooth, refer to the below.

(1) Cancellation of a firewall
Clear the window's own firewall.

(a) Execute the control panel.
(b) Execute the Windows Firewall.
(c) Execute setup or clear of the Windows Firewall.
(d) Clear all firewalls as below.



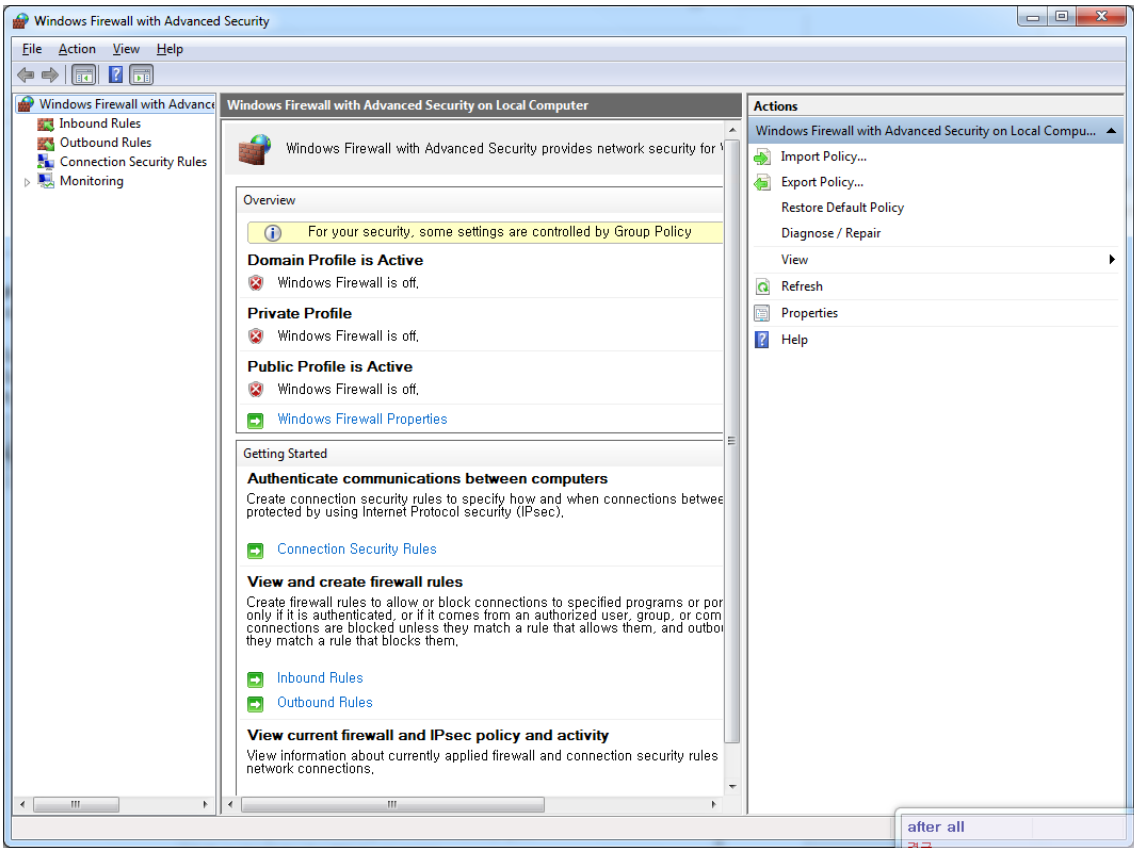
Note

If you clear the Windows Firewall, you may be exposed to various external intrusions so you are recommended to register the exceptional rules to a firewall.

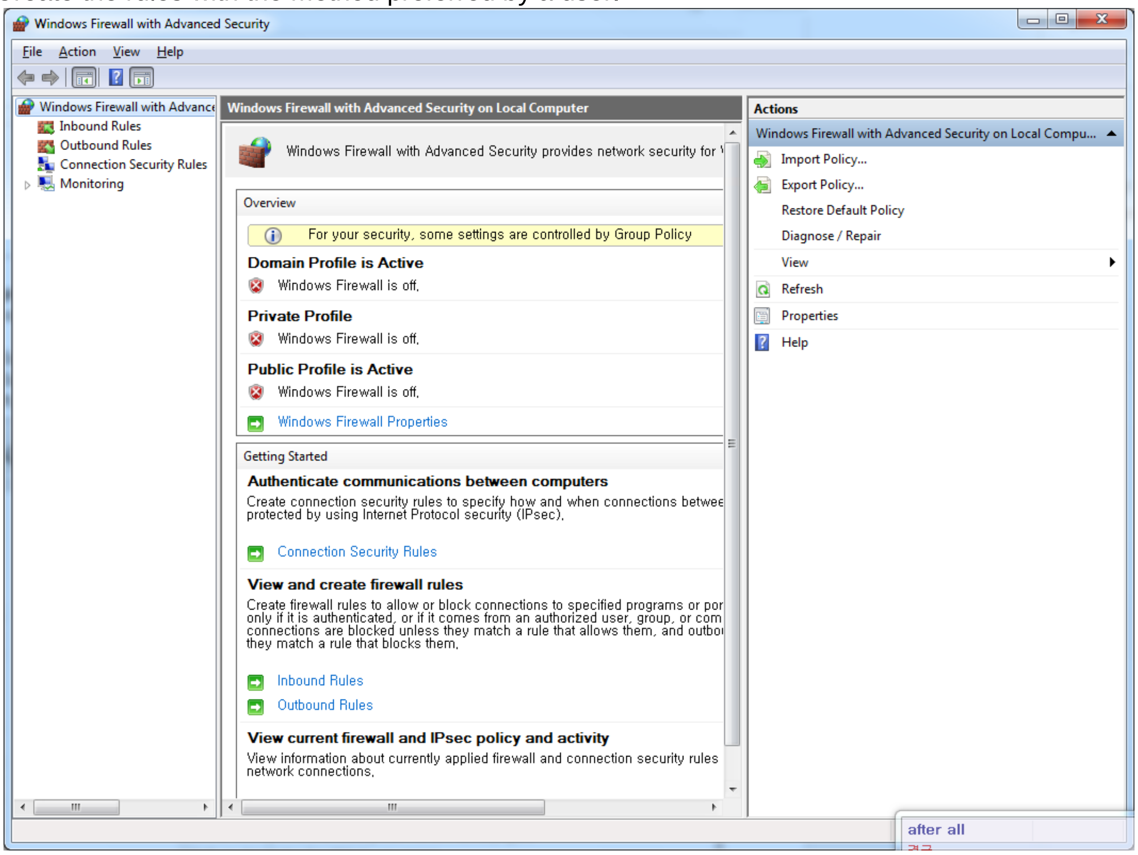
(2) Registration of exceptional rules

You can refer to the following procedures to register exceptional rules to a firewall.

(a) Execute the control panel.
(b) Execute the Windows Firewall.
(c) If you execute the advanced settings, the below screen will pop up.



(d) Choose the inbound rules.
(e) Choose ‘New Rules’ at the top of the right side.
(f) Create the rules with the method preferred by a user.



Note

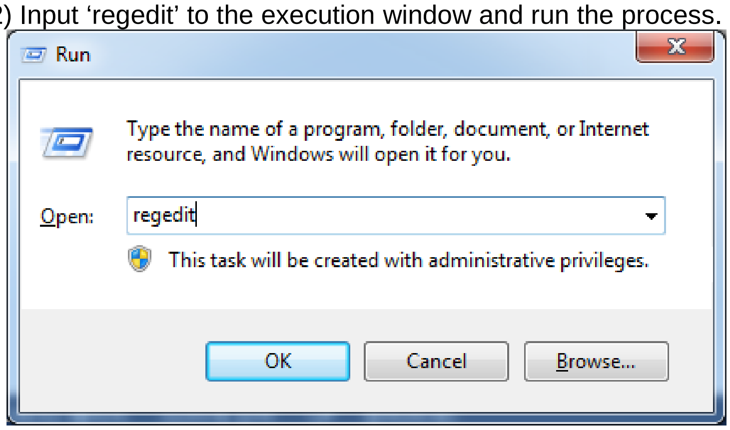
For registering exceptional rules, refer to the window manual.

14.2.6 Speed up of FTP

Motion controller's built-in FTP server is supposed to send one data packet per one scan to minimize the influence on the scan time. In this structure, if the response to the transmitted data packet is not received immediately, the next packet will not be sent until the response is obtained. However, windows is usually supposed to send the response after waiting until 2 packets are received or after 200ms, instead of responding all when receiving the data packet. Accordingly, you are recommended to set that ACK is sent whenever the TCP/IP of windows receives one packet through a register as below.

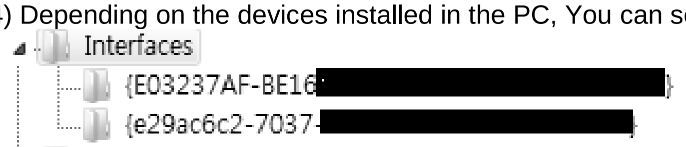
(1) Select the [Start] button of Windows for execution.(Shortcut key /Windows key + R)

(2) Input 'regedit' to the execution window and run the process.

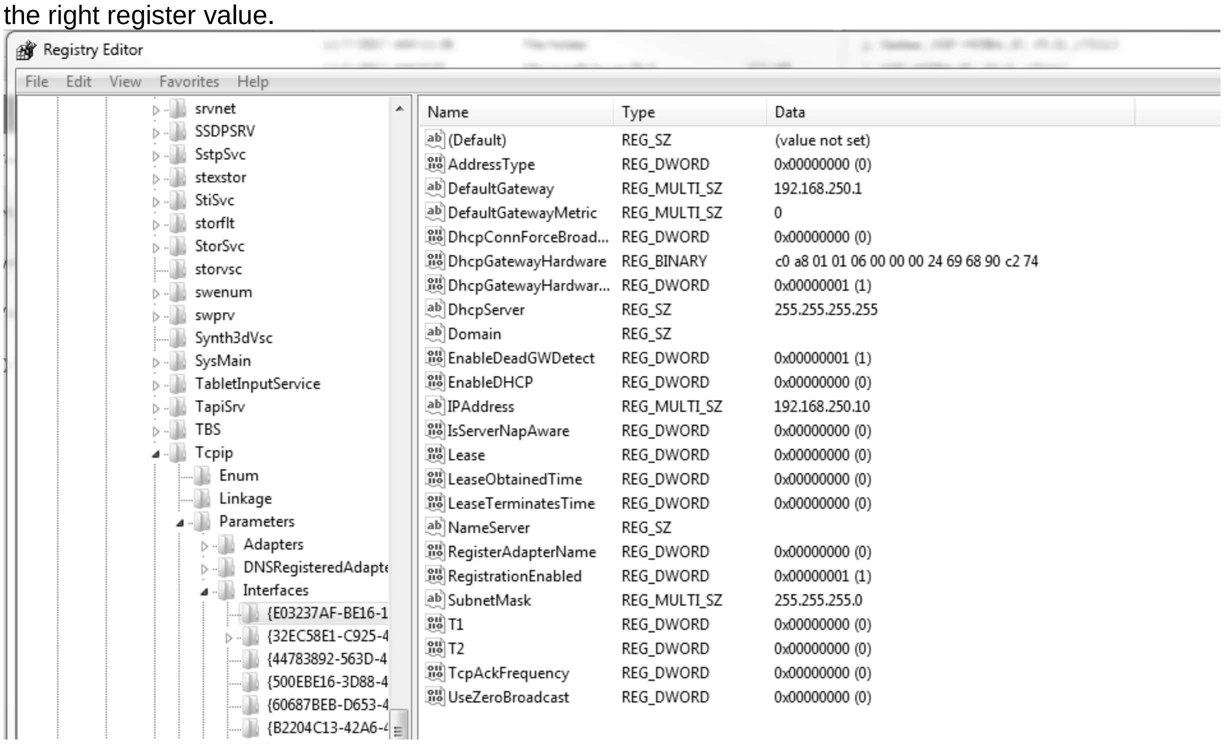


(3) Check the below path.
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\Tcpip\Parameters\Interfaces

(4) Depending on the devices installed in the PC, You can see the folders are created



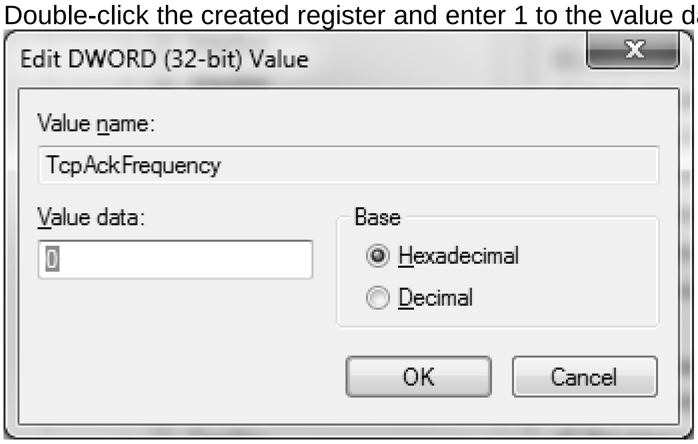
(5) If there are several register folders, select one by one and find the folder where the current PC's IP address is set in the right register value.



(6) Click with the right mouse button on the right screen of the relevant folder and select New]→ [DWORD(32bit) value].

(7) Enter the value name as shown below.
- Value name: TcpAckFrequency (It should be case-sensitive.)

(8) Double-click the created register and enter 1 to the value data.



(9) Reboot the computer.

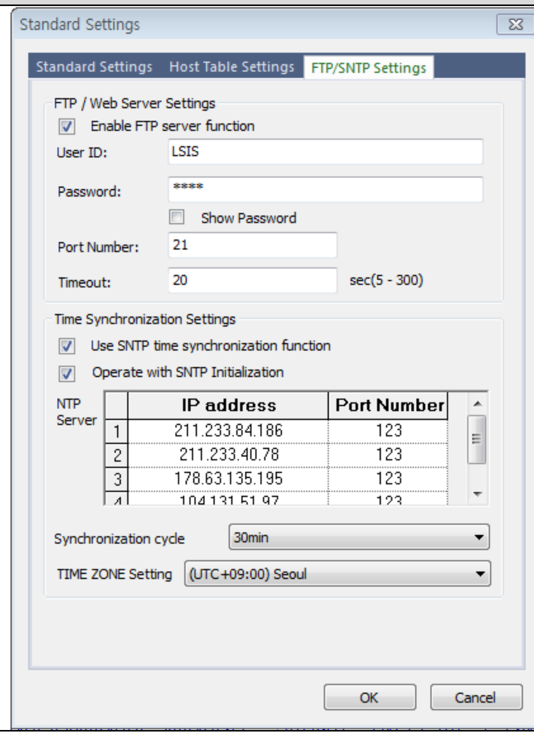
14.3 SNTP Client Functions

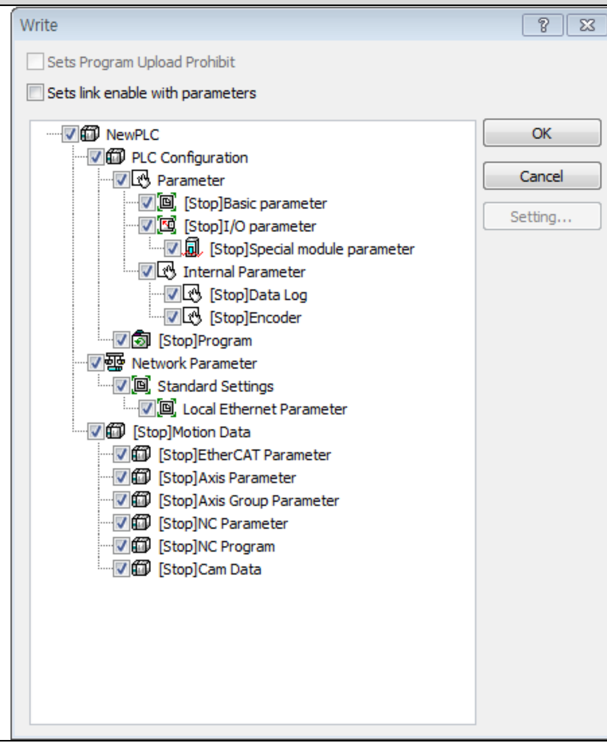
14.3.1 Outline of the Time Synchronization Protocol

Motion controller supports the NTP (Network Time Protocol) that obtains the time information by accessing to the SNTP(Simple Network Time Protocol)server and synchronizes time. The NTP is the protocol to synchronize the time of the PLC connected to the network.

14.3.2 SNTP Parameter Setting

You can set up the parameters to use the SNTP server function as shown below.

<table>
<tr><th>Procedures</th><th colspan="2">Description</th></tr>
<tr><td rowspan="2">1</td><td>Setting for SNTP</td><td>
</td></tr>
<tr><td colspan="2">1. Input the [TCP/IP setting] parameters in the Ethernet basic setting window.
- Enter the IP address, subnet mask, gateway, DNS server address.
- This address is commonly used for FTP server, SNTP client service, etc.
2. Check [SNTP Time Synchronization Enable].
3. Then, set up the SNTP server's IP address and Port No., synchronization cycle, TIME ZONE setting.</td></tr>
</table>

<table>
<tr><th>Procedures</th><th colspan="2">Description</th></tr>
<tr><td rowspan="2">2</td><td>Write parameters and Link Enable</td><td>
</td></tr>
<tr><td colspan="2">Select [Online] → [Write] in XG5000’s project window.</td></tr>
<tr><td>3</td><td colspan="2">If you click the [OK] button, ‘Write Parameters’ will be done.</td></tr>
</table>

Note

1. When parameter setting is done, the PLC reads periodically the time value from the SNTP server.
2. The SNTP server IP address is initially set as follows.

IP	Port
211.233.84.186	123
211.233.40.78	123
178.63.135.195	123
104.131.51.97	123

Note

3. If you want to use other SNTP servers, change the IP address and port No. of the SNTP server before input. Below is an example of public NTP server and port.

Server address	IP	Port	Support
time.apple.com	17.253.6.243	123	Apple
time.asia.apple.com	17.83.253.7	123	Apple
time.euro.apple.com	17.72.148.52	123	Apple
ntp.kornet.net	168.126.3.6	123	KT(Korea)
time.kriss.re.kr	210.98.16.100	123	KRISS(Korea)
time.nuri.net	211.115.194.21	123	inethosting(Korea)
time.nist.gov	132.163.4.102	123	NIST(Korea)
time.windows.com	191.233.81.105	123	MS
1.kr.pool.ntp.org	211.233.40.78	123	Navyism(Korea)
1.asia.pool.ntp.org	125.62.193.121	123	Navyism(Korea)
2.asia.pool.ntp.org	82.200.209.236	123	Navyism(Korea)
3.asia.pool.ntp.org	218.189.210.4	123	Navyism(Korea)

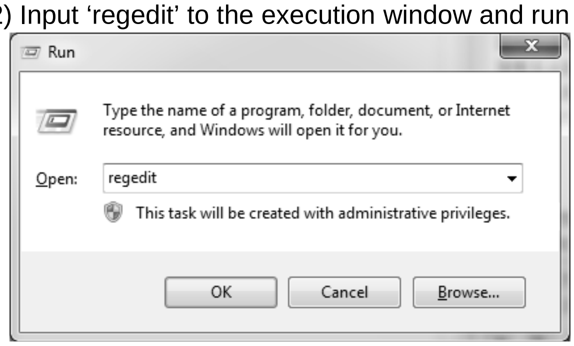
(4) If you cannot use a public NTP server, Please setup a local NTP server refer to **'14.3.3 How to setup a local NTP server'.**

14.3.3 How to Setup a Local NTP Server

If you cannot use a public NTP server, Please setup a local NTP server as follows:

(1) Select the [Start] button of Windows for execution.(Shortcut key /Windows key + R)

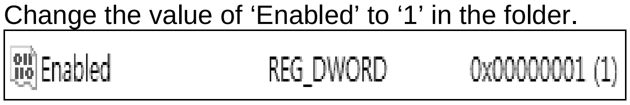
(2) Input 'regedit' to the execution window and run the process.



(3) Check the below path.

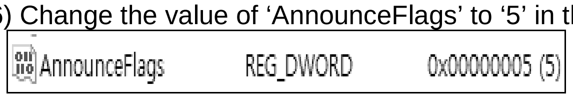
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W32Time\TimeProviders\NtpServer

(4) Change the value of 'Enabled' to '1' in the folder.



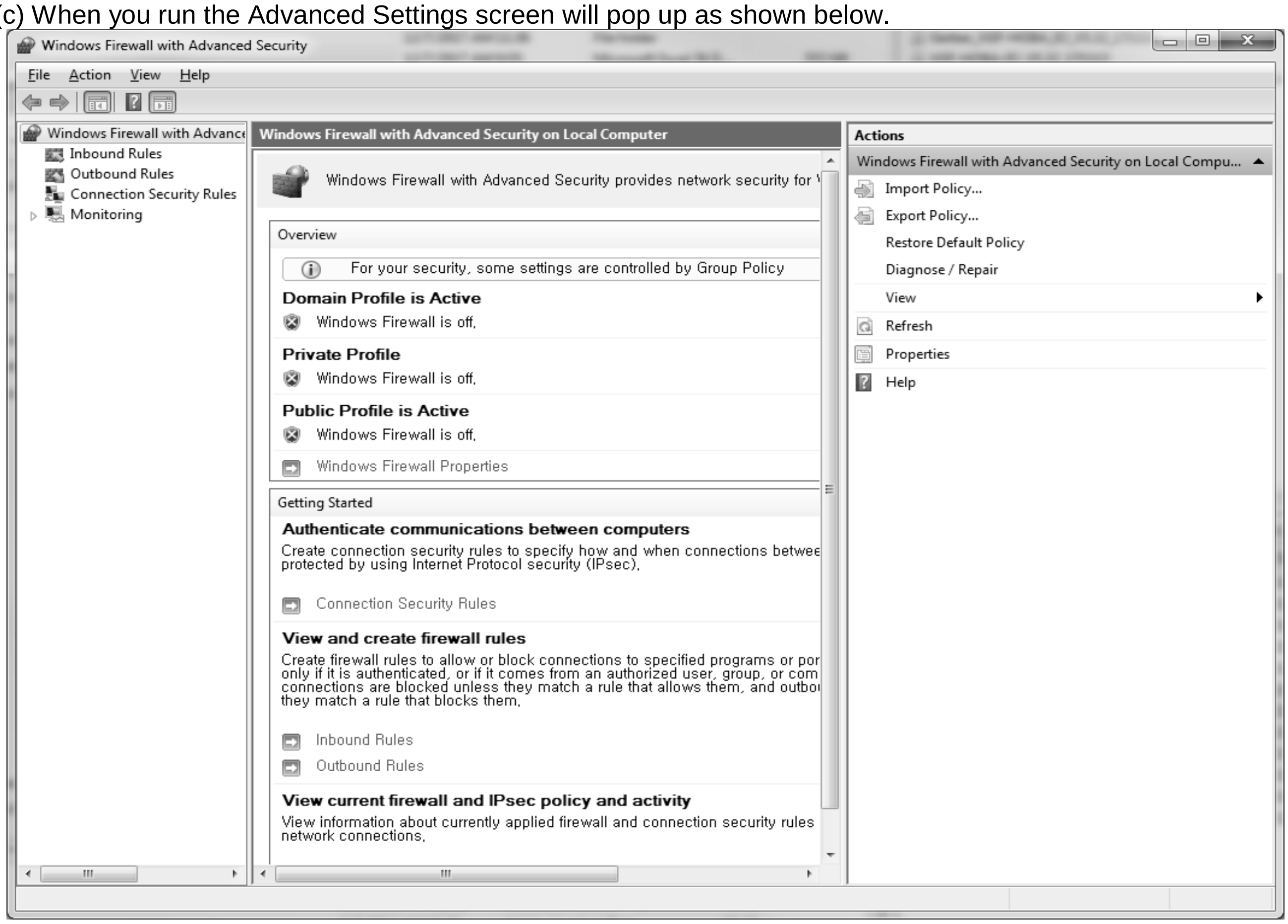
(5) Check the below path.
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W32Time\Config

(6) Change the value of 'AnnounceFlags' to '5' in the folder.

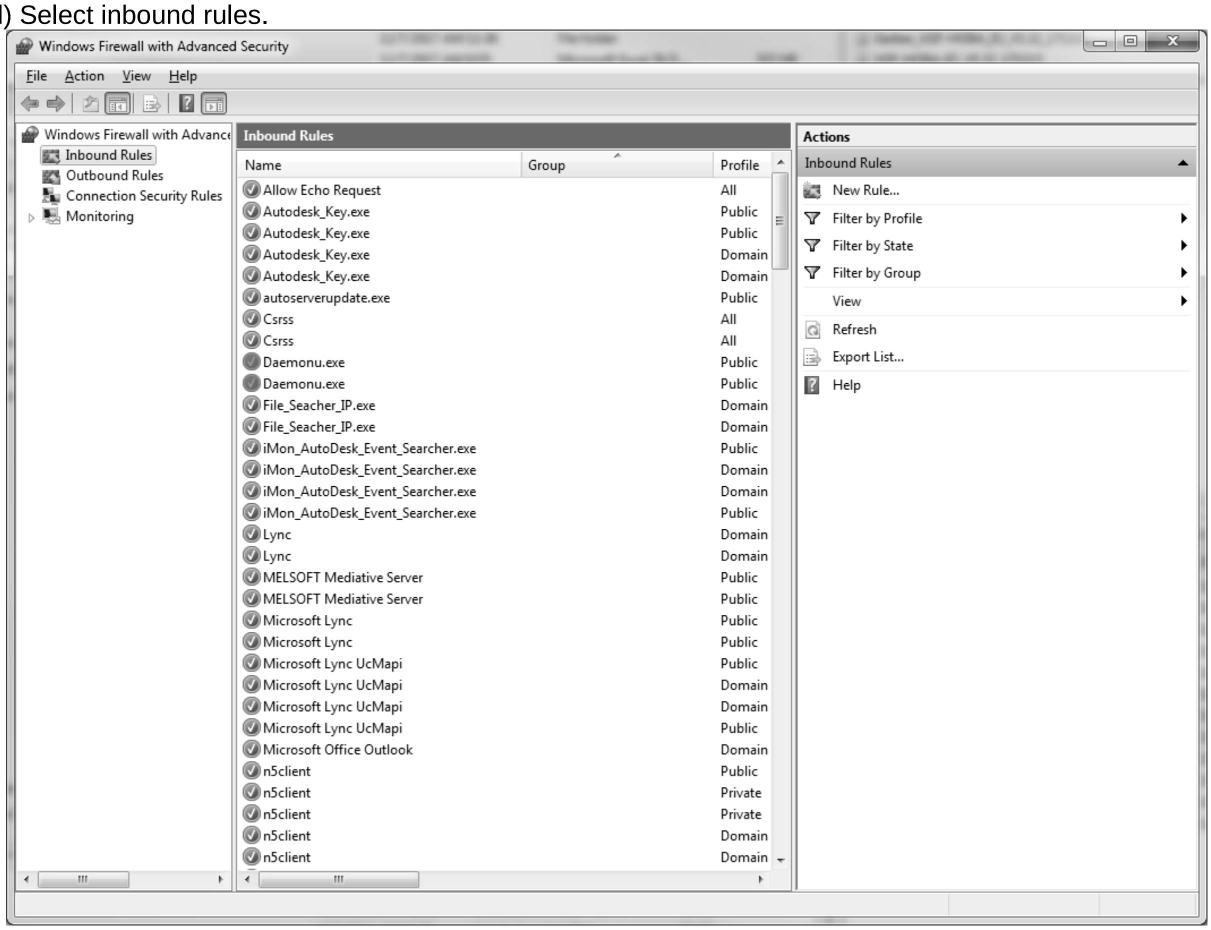


(7) Reboot the computer.

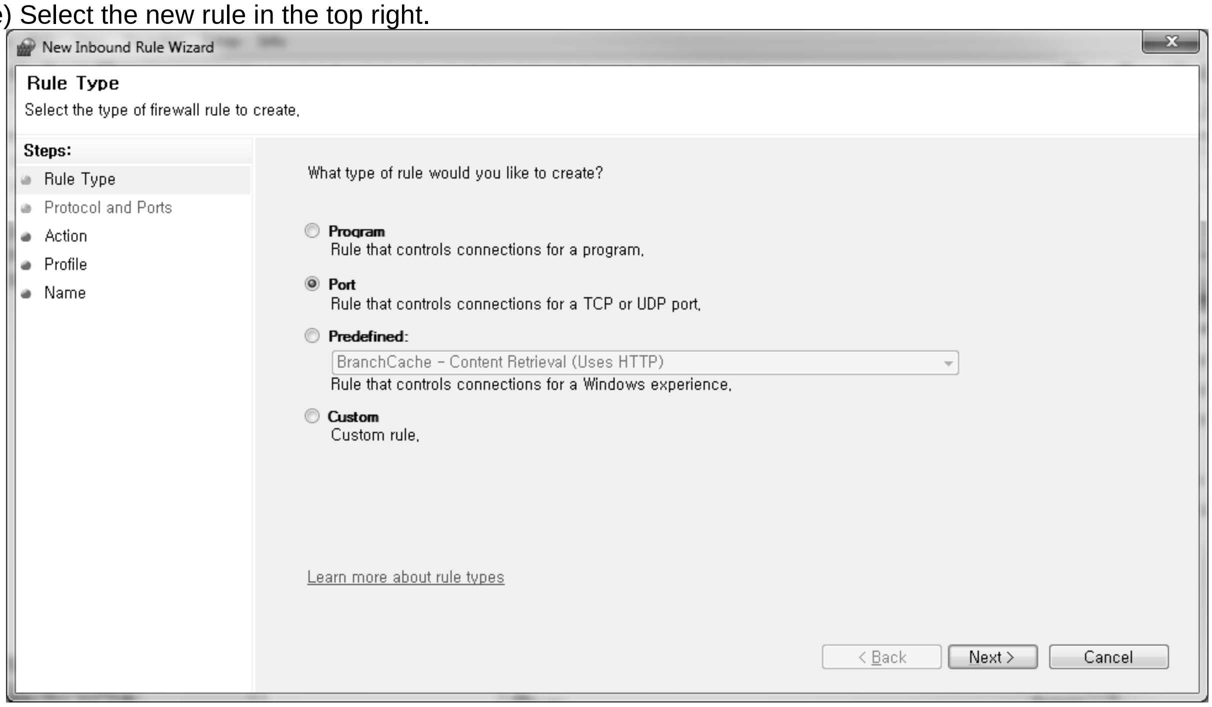
(8) Setup inbound firewall rules.
(a) Run the Control Panel.
(b) Run the Window Firewall
(c) When you run the Advanced Settings screen will pop up as shown below.



(d) Select inbound rules.

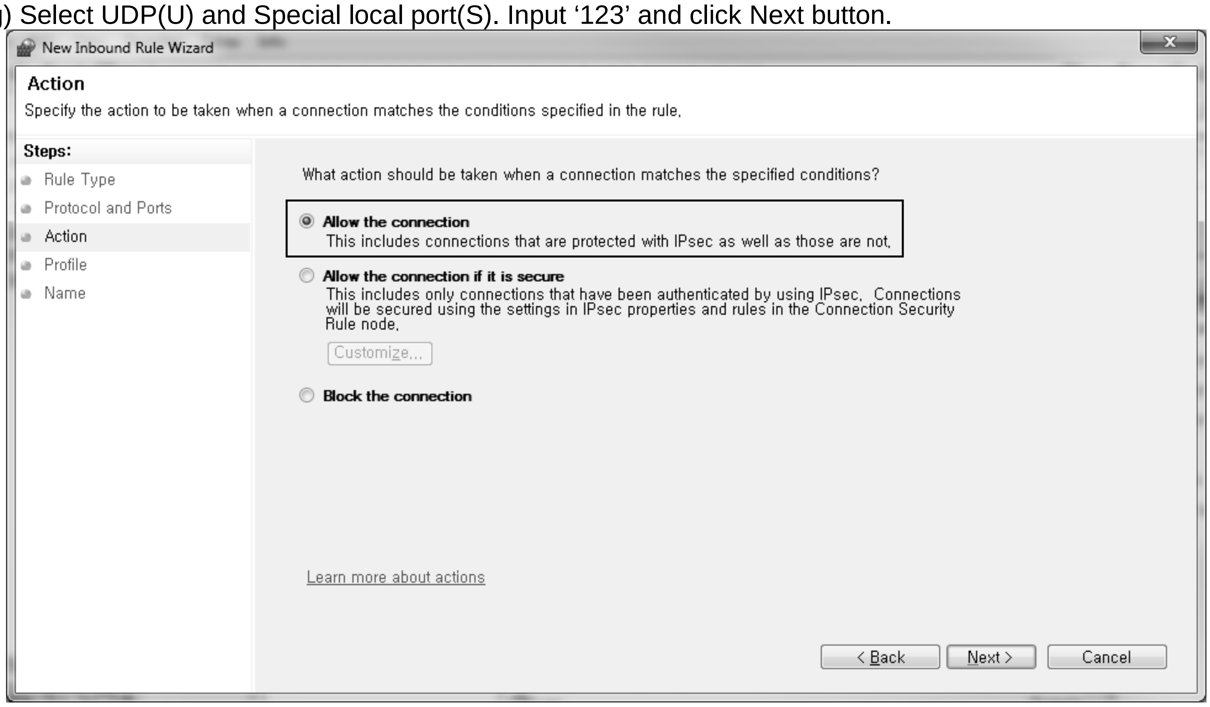


(e) Select the new rule in the top right.

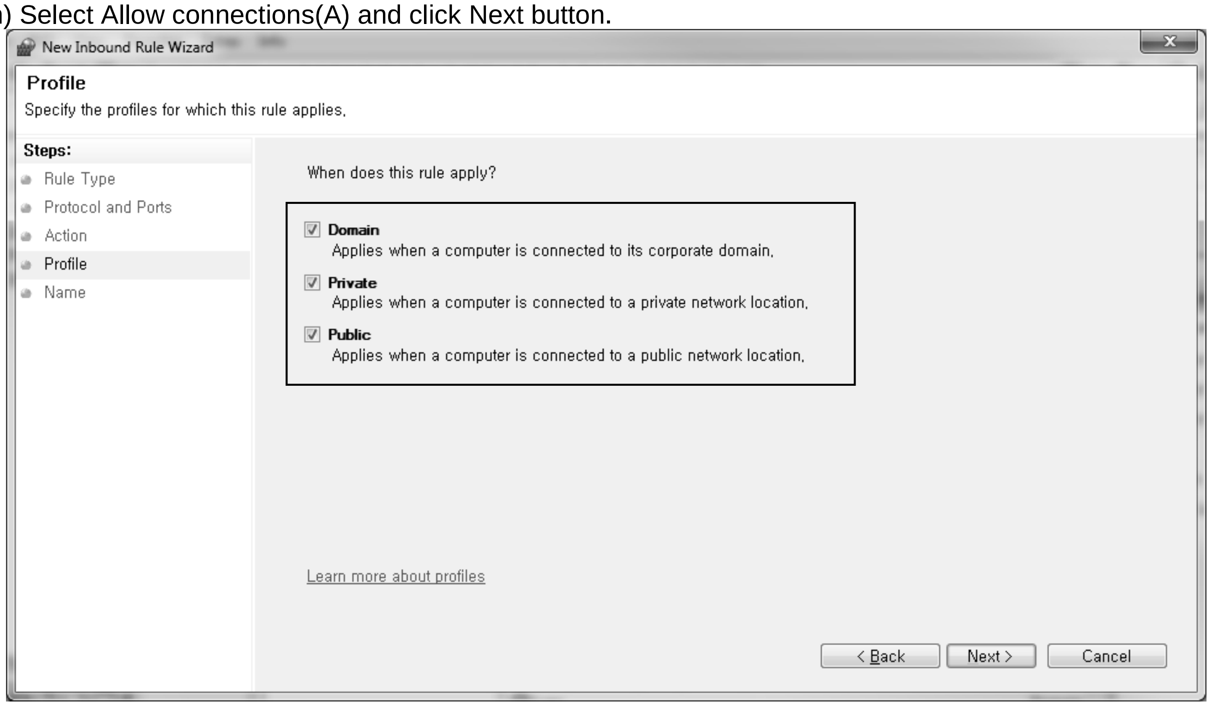


(f) Select the port and click Next button.

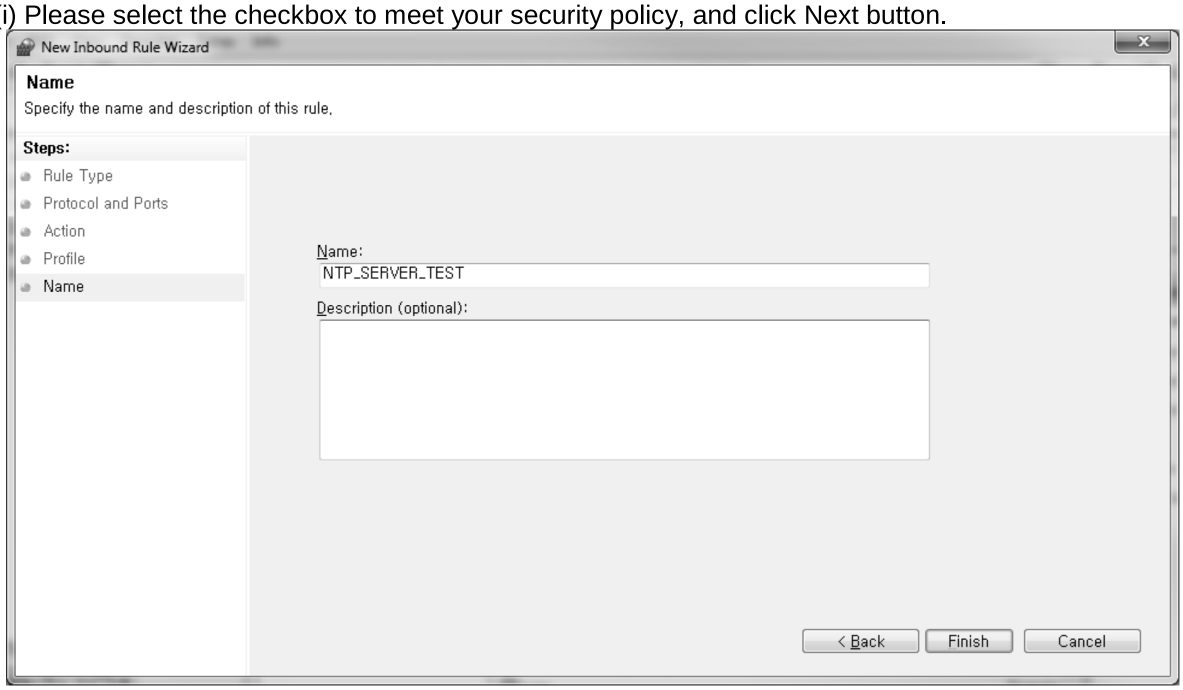
(g) Select UDP(U) and Special local port(S). Input ‘123’ and click Next button.



(h) Select Allow connections(A) and click Next button.



(i) Please select the checkbox to meet your security policy, and click Next button.



(j) Input the server name (anything) and description and click Finish button.

(k) Select the [Start] button of Windows for execution (Shortcut Key /Windowskey + R)

(l) Enter ‘CMD’ and click Confirm. (Administrator)

(m) In the command window, Input ‘net stop w32time’and press Enter key. And then, also input ‘net start w32time’and press Enter key.

(n) Input ‘ipconfig’ and press Enter key in the command window to find out the IP address of NTP server.

(o) Setting the parameters using IP address of NTP server. (refer to ‘14.3.2 SNTP server parameter setting’)

14.4 Socket Service

14.4.1 UDP (Receiver) sending/receiving program using socket service

(1) Program procedures

(a) Check the usage status of the socket service channel to be used with the SOCKET_STATUS command.

(b) Socket service channel allocation is performed with the SOCKET_UDPCREATE command.

(c) Receive data with the SOCKET_UDPRECV command.

(d) Send data with the SOCKET_UDPSEND command.

(e) If it fails in steps 3 and 4, the socket channel is returned with the SOCKET_CLOSE command, and it is executed again.

(2) ST Program – Variable

	Variable Type	Variable	Type	Description
1	VAR	INST_SOCKET_CLOSE	SOCKET_CLOSE	
2	VAR	INST_SOCKET_STATUS	SOCKET_STATUS	
3	VAR	INST_SOCKET_UDPCREATE	SOCKET_UDPCREATE	
4	VAR	INST_SOCKET_UDPRECV	SOCKET_UDPRECV	
5	VAR	INST_SOCKET_UDPSEND	SOCKET_UDPSEND	
6	VAR	SocketServiceCh	USINT	Socket service channel to use
7	VAR	Stage	UDINT	Program processing procedure
8	VAR	TestEnable	BOOL	Choose whether to start the test
9	VAR	UdpBuffer	ARRAY[0..1399] OF BYTE	memory of transmit/receive data
10	VAR	CycleCounter	UDINT	For setting the frequency of program execution

(3) ST Program

```
IF(TestEnable= TRUE) THEN
        CycleCounter := CycleCounter + 1;

        IF(CycleCounter > 20)THEN
                CycleCounter := 0;
```

```
                CASE Stage OF
                        0:      //Check and create socket service channel status to use
                                SocketServiceCh := 1;
        //Specify the socket channel to use
                                INST_SOCKET_STATUS(
        //instance reset
                                        REQ             := FALSE,
                                        SOCKCH          := SocketServiceCh);

                                INST_SOCKET_STATUS(
        //instance execution
                                        REQ             := TRUE,
                                        SOCKCH          := SocketServiceCh);

                                IF(INST_SOCKET_STATUS.DONE = TRUE AND

                                        INST_SOCKET_STATUS.SOCKETTYPE = 0) THEN         //For
channels that are performing normally and not in use
                                        INST_SOCKET_UDPCREATE(
        //instance execution
                                                REQ                             := TRUE,
                                                SRCPORT                         := 8000,
        //Set port to 8000
                                                OPENSOCKCH      := SocketServiceCh);
        //Socket service channel to use

                                        IF(INST_SOCKET_UDPCREATE.DONE = TRUE) THEN
        //in normal performance
                                                Stage := 1;
        //change Stage 1
                                        END_IF;

                                        INST_SOCKET_UDPCREATE(
        //instance reset
                                                REQ                             := FALSE,
                                                SRCPORT                         := 8000,
                                                OPENSOCKCH      := SocketServiceCh);
                                END_IF;
```

```
1:      //data receive
        INST_SOCKET_UDPRECV(
//instance execution
            REQ             := TRUE,

            SOCKCH      := SocketServiceCh,
//Socket service channel to created
            TIMEOUT         := 7,
//Receive Standby    for up to 7 sec
            SIZE            := 1400,        //specify the maximum
receive size (cannot set more than the buffer size)
            RECVDATA    := UdpBuffer);
//received data buffer

        IF(INST_SOCKET_UDPCREATE.DONE = TRUE) THEN
//in normal performance
            Stage := 2;
//change Stage 2
            INST_SOCKET_UDPRECV(
//instance reset
                REQ             := FALSE,
                SOCKCH          := SocketServiceCh,
                TIMEOUT         := 7,
                SIZE            := 1400,
                RECVDATA    := UdpBuffer);

        END_IF;

        IF(INST_SOCKET_UDPRECV.ERROR = TRUE)THEN
//When performing abnormal
            Stage := 0;
//change Stage 0
            INST_SOCKET_CLOSE(
//instance execution
                REQ         := TRUE,
                SOCKCH      := SocketServiceCh);
//Socket service channel to close
```

```
                    INST_SOCKET_CLOSE(
    //instance reset
                        REQ         := FALSE,
                        SOCKCH      := SocketServiceCh);

                    INST_SOCKET_UDPRECV(
    //instance reset
                        REQ              := FALSE,
                        SOCKCH           := SocketServiceCh,
                        TIMEOUT          :=7,
                        SIZE             := 1400,
                        RECVDATA    := UdpBuffer);

                END_IF;
            2:  //data sending
                INST_SOCKET_UDPSEND(
    //instance execution
                    REQ              := TRUE,
                    SOCKCH      := SocketServiceCh,
    //Socket service channel to created
                    DSTPORTNO   := INST_SOCKET_UDPRECV.FROMPORTNO,
    //Received port information
                    DSTIPADDR   := INST_SOCKET_UDPRECV.FROMIPADDR,
    //Received IP information
                    ENDDATA     := UdpBuffer,
    // Send data buffer
                    SIZE             := INST_SOCKET_UDPRECV.RECVSIZE);
    //send data size

                IF(INST_SOCKET_UDPSEND.DONE = TRUE) THEN            //in
normal performance
                    Stage := 1;
    //change Stage 1
                END_IF;

                IF(INST_SOCKET_UDPSEND.ERROR = TRUE)THEN
    //When performing abnormal
```

```
                                        Stage := 0;
            //change Stage 0
                                        INST_SOCKET_CLOSE(
            //instance execution
                                              REQ          := TRUE,

                                              SOCKCH       := SocketServiceCh);
            //Socket service channel to close

                                        INST_SOCKET_CLOSE(
            //instance reset
                                              REQ          := FALSE,
                                              SOCKCH       := SocketServiceCh);

                                  END_IF;

                                  INST_SOCKET_UDPSEND(
            //instance reset
                                        REQ                := FALSE,
                                        SOCKCH             := SocketServiceCh,
                                        DSTPORTNO    := INST_SOCKET_UDPRECV.FROMPORTNO,
                                        DSTIPADDR    := INST_SOCKET_UDPRECV.FROMIPADDR,
                                        SENDDATA     := UdpBuffer,
                                        SIZE               := INST_SOCKET_UDPRECV.RECVSIZE);


                  END_CASE;
            END_IF;
      END_IF;
```

14.4.2 UDP (Sender) sending/receiving program using socket service

(1) Program Procedures

(a) Check the usage status of the socket service channel to be used with the SOCKET_STATUS command.

(b) Socket service channel allocation is performed with the SOCKET_UDPCREATE command.

(c) Send data with the SOCKET_UDPSEND command.

(d) Receive data with the SOCKET_UDPRECV command.

(e) If it fails in steps (c) and (d), the socket channel is returned with the SOCKET_CLOSE command, and it is executed again.

(2) ST PROGRAM – VARIABLE

	Variable Type	Variable	Type	Description
1	VAR	INST_SOCKET_CLOSE	SOCKET_CLOSE	
2	VAR	INST_SOCKET_STATUS	SOCKET_STATUS	
3	VAR	INST_SOCKET_UDPCREATE	SOCKET_UDPCREATE	
4	VAR	INST_SOCKET_UDPRECV	SOCKET_UDPRECV	
5	VAR	INST_SOCKET_UDPSEND	SOCKET_UDPSEND	
6	VAR	SocketServiceCh	USINT	Socket service channel to use
7	VAR	Stage	UDINT	Program processing procedure
8	VAR	TestEnable	BOOL	Choose whether to start the test
9	VAR	UdpBuffer	ARRAY[0..1399] OF BYTE	memory of transmit/receive data
10	VAR	CycleCounter	UDINT	For setting the frequency of program execution

(3) ST PROGRAM

```
IF(TestEnable= TRUE) THEN
        CycleCounter := CycleCounter + 1;
        IF(CycleCounter > 20)THEN
                CycleCounter := 0;
                CASE Stage OF
                        0:      //Check and create socket service channel status to use
                                SocketServiceCh := 2;
        //Specify the socket channel to use
                                INST_SOCKET_STATUS(                               //instance reset
                                        REQ             := FALSE,
```

```
                    SOCKCH          := SocketServiceCh);

                INST_SOCKET_STATUS(                                     //instance
execution
                    REQ             := TRUE,
                    SOCKCH          := SocketServiceCh);

                IF(INST_SOCKET_STATUS.DONE = TRUE AND

                    INST_SOCKET_STATUS.SOCKETTYPE = 0) THEN     //For    channels
that are performing normally and not in use
                    INST_SOCKET_UDPCREATE(
    //instance execution
                        REQ                 := TRUE,
                        SRCPORT             := 8001,                //Set
port to 8001
                        OPENSOCKCH  := SocketServiceCh);        //Socket   service
channel to use

                    IF(INST_SOCKET_UDPCREATE.DONE = TRUE) THEN          //in
normal performance
                        Stage := 1;
    //change Stage 1
                    END_IF;

                    INST_SOCKET_UDPCREATE(
    //instance reset
                        REQ                     := FALSE,
                        SRCPORT                 := 8001,
                        OPENSOCKCH          := SocketServiceCh);
                END_IF;

            1:  //data sending
                DestIpAddr[0] := 192;
                DestIpAddr[1] := 168;
                DestIpAddr[2] := 0;
                DestIpAddr[3] := 111;
    //Initialize the IP value of the receiving side
```

```
                    INST_SOCKET_UDPSEND(
    //instance execution
                        REQ             := TRUE,
                        SOCKCH          := SocketServiceCh,
    //Socket service channel to created
                        DSTPORTNO       := 8001,                //Receiving side
port information
                        DSTIPADDR       := DestIpAddr,
    //Receiving side IP information
                        ENDDATA         := UdpBuffer,                // send
data buffer
                        SIZE            := 1400);   //size of received data

                    IF(INST_SOCKET_UDPSEND.DONE = TRUE) THEN    //in    normal
performance
                        Stage := 2;
    //change Stage 2
                    END_IF;

                    IF(INST_SOCKET_UDPSEND.ERROR = TRUE)THEN          //When
performing abnormal
                        Stage := 0;
    //change Stage 0
                        INST_SOCKET_CLOSE(                     //instance
execution
                            REQ         := TRUE,

                            SOCKCH      := SocketServiceCh);
    //Socket service channel to close

                        INST_SOCKET_CLOSE(                     //instance reset
                            REQ         := FALSE,
                            SOCKCH      := SocketServiceCh);

                    END_IF;

                    INST_SOCKET_UDPSEND(
    //instance reset
```

```
                    REQ              := FALSE,
                    SOCKCH           := SocketServiceCh,
                    DSTPORTNO        := INST_SOCKET_UDPRECV.FROMPORTNO,
                    DSTIPADDR        := INST_SOCKET_UDPRECV.FROMIPADDR,
                    SENDDATA         := UdpBuffer,
                    SIZE             := 1400);

          2:    //data receive
                INST_SOCKET_UDPRECV(
    //instance execution
                    REQ              := TRUE,

                    SOCKCH           := SocketServiceCh,
    //Socket service channel to created
                    TIMEOUT          := 7,
    //Receive Standby    for up to 7 sec
                    SIZE             := 1400,        //specify the maximum receive size
(cannot set more than the buffer size)
                    RECVDATA         := UdpBuffer);
    //received data buffer

                IF(INST_SOCKET_UDPCREATE.DONE = TRUE) THEN                  //in
normal performance
                    Stage := 1;
    //change Stage 1
                    INST_SOCKET_UDPRECV(
    //instance reset
                        REQ          := FALSE,
                        SOCKCH       := SocketServiceCh,
                        TIMEOUT      := 7,
                        SIZE         := 1400,
                        RECVDATA     := UdpBuffer);

                END_IF;

                IF(INST_SOCKET_UDPRECV.ERROR = TRUE)THEN                    //When
performing abnormal
                    Stage := 0;
```

```
        //change Stage 0
                                INST_SOCKET_CLOSE(                          //instance
execution
                                        REQ             := TRUE,
                                        SOCKCH          := SocketServiceCh);
        //Socket service channel to close

                                INST_SOCKET_CLOSE(                          //instance reset
                                        REQ             := FALSE,
                                        SOCKCH          := SocketServiceCh);

                                INST_SOCKET_UDPRECV(
        //instance reset
                                        REQ             := FALSE,
                                        SOCKCH          := SocketServiceCh,
                                        TIMEOUT         :=7,
                                        SIZE            := 1400,
                                        RECVDATA        := UdpBuffer);

                        END_IF;

                END_CASE;
        END_IF;
END_IF;
```

14.4.3 TCP server sending/receiving program using socket service

(1) Program Procedures

(a) Check the usage status of the server socket service channel to be used with the SOCKET_STATUS command.

(b) Server Socket service channel allocation is performed with the SOCKET_TCPLISTEN command.

(c) Check the usage status of the peer socket service channel to be used with the SOCKET_STATUS command.

(d) The SOCKET_TCPACCEPT command allows peer connection and performs socket service channel assignment.

(e) Receive data with the SOCKET_TCPRECV command.

(f) Send data with the SOCKET_TCPSEND command.

(g) If it fails in steps 3, 4, and 5, the socket channel is returned with the SOCKET_CLOSE command, and it is executed again.

(2) ST PROGRAM – VARIABLE

	Variable Type	Variable	Type	Description
1	VAR	INST_SOCKET_CLOSE	SOCKET_CLOSE	
2	VAR	INST_SOCKET_STATUS	SOCKET_STATUS	
3	VAR	INST_SOCKET_TCPACCEPT	SOCKET_TCPACCEPT	
4	VAR	INST_SOCKET_TCPLISTEN	SOCKET_TCPLISTEN	
5	VAR	INST_SOCKET_TCPRECV	SOCKET_TCPRECV	
6	VAR	INST_SOCKET_TCPSEND	SOCKET_TCPSEND	
7	VAR	PeerSocketServiceCh	USINT	Socket service channel to use(PEER)
8	VAR	ServerSocketServiceCh	USINT	Socket service channel to use(SERVER)
9	VAR	Stage	UDINT	Program processing procedure
10	VAR	TcpBuffer	ARRAY[0..1399] OF BYTE	memory of transmit/receive data
11	VAR	TestEnable	BOOL	Choose whether to start the test
12	VAR	CycleCounter	UDINT	For setting the frequency of program execution

(3) ST PROGRAM

```
IF(TestEnable= TRUE) THEN
        CycleCounter := CycleCounter + 1;
        IF(CycleCounter > 20)THEN
                CycleCounter := 0;
```

```
                    CASE Stage OF
                              0:         //specifying and creating a Server Socket Channel
                                         ServerSocketServiceCh     := 3;                                  //Specify the
socket channel to use(server)
                                         eerSocketServiceCh                 := 4;
          //Specify the socket channel to use(peer)
                                         INST_SOCKET_STATUS(                                              //instance reset
                                                   REQ                := FALSE,
                                                   SOCKCH             := ServerSocketServiceCh);

                                         INST_SOCKET_STATUS(                                              //instance
execution
                                                   REQ                := TRUE,
                                                   SOCKCH             := ServerSocketServiceCh);
          //Socket service channel to use(server)

                                         IF(INST_SOCKET_STATUS.DONE = TRUE AND
                                                   INST_SOCKET_STATUS.SOCKETTYPE = 0) THEN             //For channel
that are performing normally and not in use
                                                   INST_SOCKET_TCPLISTEN(
          //instance execution
                                                             REQ                := TRUE,
                                                             SRCPORT            := 8002,                         //set the
server side port to 8002
                                                             PEERCNT            := 1,                            //Set
the maximum connecting peers to 1.
                                                             OPENSOCKCH := ServerSocketServiceCh);           //Socket service
channel to use(server)

                                                   IF(INST_SOCKET_TCPLISTEN.DONE = TRUE) THEN                    //in
normal performance
                                                             Stage := 1;
          //change Stage 1
                                                   END_IF;

                                                   INST_SOCKET_TCPLISTEN(
          //instance reset
                                                             REQ                               := FALSE,
```

```
                        SRCPORT             := 8002,
                        PEERCNT             := 1,
                        OPENSOCKCH          := ServerSocketServiceCh);

        END_IF;

    1:  //specify peer socket channel and allow access
        INST_SOCKET_STATUS(                                     //instance reset
                REQ         := FALSE,
                SOCKCH      := PeerSocketServiceCh);

        INST_SOCKET_STATUS(                                     //instance
execution
                REQ         := TRUE,
                SOCKCH      := PeerSocketServiceCh);            //Socket service
channel to use(peer)

        IF(INST_SOCKET_STATUS.DONE = TRUE AND
                (INST_SOCKET_STATUS.SOCKETTYPE = 0 OR
                INST_SOCKET_STATUS.SOCKETTYPE = 1)) THEN //In case of normal
operation and not in use or in case of access waiting channel
                INST_SOCKET_TCPACCEPT(
    //instance execution
                        REQ                 := TRUE,
                        SERVERSOCKCH        := ServerSocketServiceCh,       //socket
service channel to created
                        OPENPEERSOCKCH      := PeerSocketServiceCh,   //peer socket
service channel to accept
                        TIMEOUT             := 5);                          //
connection Standby    for up to 5 sec

                IF(INST_SOCKET_TCPACCEPT.DONE = TRUE) THEN                  //in
normal performance
                        Stage := 2;
    //change Stage 2
                        INST_SOCKET_TCPACCEPT(
    //instance reset
                                REQ                 := FALSE,
```

```
                                        SERVERSOCKCH          := ServerSocketServiceCh,
                                        OPENPEERSOCKCH        := PeerSocketServiceCh,
                                        TIMEOUT               := 5);


                        END_IF;

                        IF(INST_SOCKET_TCPACCEPT.ERROR = TRUE) THEN      //When
performing abnormal
                                INST_SOCKET_TCPACCEPT(
        //instance reset
                                        REQ                   := FALSE,
                                        SERVERSOCKCH          := ServerSocketServiceCh,
                                        OPENPEERSOCKCH        := PeerSocketServiceCh,
                                        TIMEOUT               := 5);

                                IF(INST_SOCKET_TCPACCEPT.ERRID <> 4) THEN //If it is not a
timeout error
                                        Stage := 0;
        //change Stage 0
                                        INST_SOCKET_CLOSE(                  //instance
execution
                                                REQ           :=TRUE,
                                                SOCKCH        := ServerSocketServiceCh);
        //close server socket service channel

                                        INST_SOCKET_CLOSE(                  //instance reset
                                                REQ           :=FALSE,
                                                SOCKCH        := ServerSocketServiceCh);

                                END_IF;

                        END_IF;
                END_IF;
            2:  //data receiving and sending
                INST_SOCKET_TCPRECV(
        //instance execution
                        REQ                   := TRUE,
```

```
                    SOCKCH          := PeerSocketServiceCh,      //Connection
allowed peer socket service channel
                    TIMEOUT         := 5,
    //Receive Standby    for up to 5 sec
                    SIZE            := 1400,     //specify the maximum receive size
(cannot set more than the buffer size)
                    RECVDATA        := TcpBuffer);
    //received data buffer

                IF(INST_SOCKET_TCPRECV.DONE = TRUE)THEN           //in normal
performance
                    INST_SOCKET_TCPRECV(
    //instance reset
                        REQ         := FALSE,
                        SOCKCH      := PeerSocketServiceCh,
                        TIMEOUT     := 5,
                        SIZE        := 1400,
                        RECVDATA    := TcpBuffer);



                    INST_SOCKET_TCPSEND(
    //instance execution
                        REQ         := TRUE,
                        SOCKCH      := PeerSocketServiceCh,   //Connection allowed peer
socket service channel
                        SENDDATA    := TcpBuffer,                     // send
data buffer
                        SIZE        := INST_SOCKET_TCPRECV.RECVSIZE);  //send
size

                    IF(INST_SOCKET_TCPSEND.ERROR = TRUE)THEN             //When
performing abnormal
                        Stage := 1;
    //change Stage 1
                        INST_SOCKET_CLOSE(                           //instance
execution
                            REQ         := TRUE,
                            SOCKCH      := PeerSocketServiceCh);  //Connection
```

```
allowed peer socket service channel

                        INST_SOCKET_CLOSE(                        //instance reset
                                REQ             := FALSE,
                                SOCKCH          := PeerSocketServiceCh);

                END_IF;

                INST_SOCKET_TCPSEND(
        //instance reset
                        REQ                     := FALSE,
                        SOCKCH                  := PeerSocketServiceCh,
                        SENDDATA                := TcpBuffer,
                        SIZE                    := INST_SOCKET_TCPRECV.RECVSIZE);

        END_IF;

        IF(INST_SOCKET_TCPRECV.ERROR = TRUE)THEN                        //When
performing abnormal
                INST_SOCKET_TCPRECV(
        //instance reset
                        REQ                     := FALSE,
                        SOCKCH                  := PeerSocketServiceCh,
                        TIMEOUT                 := 5,
                        SIZE                    := 1400,
                        RECVDATA                := TcpBuffer);

                IF(INST_SOCKET_TCPRECV.ERRID <> 4) THEN             //If it is not a
timeout error
                        Stage := 1;
        //change Stage 1
                        INST_SOCKET_CLOSE(                        //instance
execution
                                REQ             := TRUE,

                                SOCKCH          := PeerSocketServiceCh);  //Connection
allowed peer socket service channel
```

```
                        INST_SOCKET_CLOSE(                //instance reset
                            REQ         := FALSE,
                            SOCKCH      := PeerSocketServiceCh);

                    END_IF;
                END_IF;
        END_CASE;

    END_IF;
END_IF;
```

14.4.4 TCP client sending/receiving program using socket service

(1) Program Procedures

(a) Check the usage status of the socket service channel to be used with the SOCKET_STATUS command.

(b) Connect to the other side (Server) with the SOCKET_TCPCONNECT command.

(c) Send data with the SOCKET_TCPSEND command.

(d) Receive data with the SOCKET_TCPRECV command.

(e) If it fails in steps 3 and 4, the socket channel is returned with the SOCKET_CLOSE command, and repeat from step 1

(2) ST PROGRAM – VARIABLE

	Variable Type	Variable	Type	Description
1	VAR	INST_SOCKET_CLOSE	SOCKET_CLOSE	
2	VAR	INST_SOCKET_STATUS	SOCKET_STATUS	
3	VAR	INST_SOCKET_TCPCONNECT	SOCKET_TCPCONNECT	
4	VAR	INST_SOCKET_TCPRECV	SOCKET_TCPRECV	
5	VAR	INST_SOCKET_TCPSEND	SOCKET_TCPSEND	
6	VAR	SocketServiceCh	USINT	Socket service channel to use(PEER)
7	VAR	Stage	UDINT	Socket service channel to use(SERVER)
8	VAR	TcpBuffer	ARRAY[0..1399] OF BYTE	Program processing procedure
9	VAR	TestEnable	BOOL	memory of transmit/receive data
10	VAR	DestIpAddr	ARRAY[0..3] OF USINT	Choose whether to start the test
11	VAR	CycleCounter	UDINT	For setting the frequency of program execution

(3) ST PROGRAM

```
IF(TestEnable= TRUE) THEN
        CycleCounter := CycleCounter + 1;
        IF(CycleCounter > 20)THEN
                CycleCounter := 0;
                CASE Stage OF
                0:      //Check socket service channel status to use
                        SocketServiceCh := 5;                               //Specify the socket
channel to use
                        INST_SOCKET_STATUS(                          //instance reset
                                REQ             :=FALSE,
```

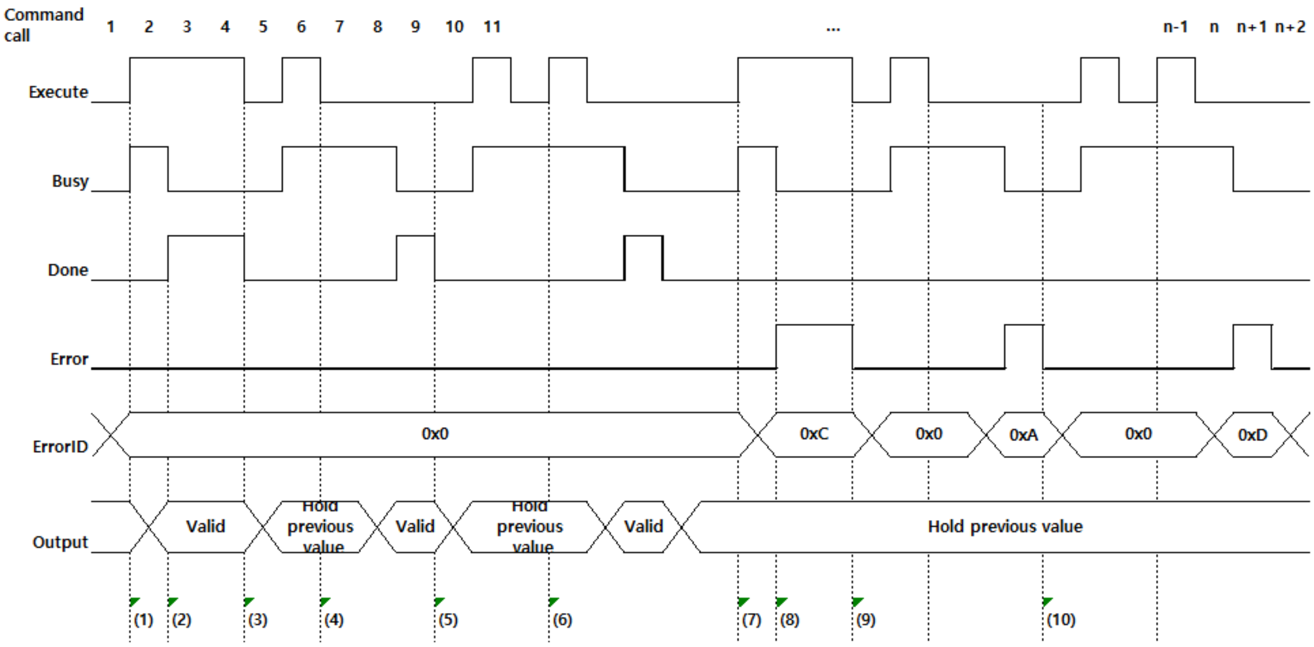
```
                SOCKCH          :=SocketServiceCh);

            INST_SOCKET_STATUS(                         //instance execution
                REQ             :=TRUE,
                SOCKCH          :=SocketServiceCh);

            IF(INST_SOCKET_STATUS.DONE = TRUE AND
                INST_SOCKET_STATUS.SOCKETTYPE = 0) THEN //For channel that are performing
normally and not in use
                Stage := 1;

            END_IF;

        1:  //try to connection
            DestIpAddr[0] := 192;
            DestIpAddr[1] := 168;
            DestIpAddr[2] := 0;
            DestIpAddr[3] := 111;                           //Initialize the IP value of
the connection
            INST_SOCKET_TCPCONNECT(                     //instance execution
                REQ                 := TRUE,
                SRCPORT             := 8003,                //Set port to 8003
                TIMEOUT             := 5,                   // connection   for up to 5
sec
                DSTIPADDR           := DestIpAddr,          //      connection      IP
information
                DSTPORT             := 8003,                //connection station Port
information
                OPENSOCKCH  := SocketServiceCh);        //Socket service channel to use

            IF(INST_SOCKET_TCPCONNECT.DONE = TRUE) THEN //in normal performance
                Stage := 2;                                 //change Stage 2
                INST_SOCKET_TCPCONNECT(                 //instance reset
                    REQ                 := FALSE,
                    SRCPORT             := 8003,
                    TIMEOUT             := 5,
                    DSTIPADDR           := DestIpAddr,
                    DSTPORT             := 7000,
```

```
                    OPENSOCKCH   := SocketServiceCh);

            END_IF;

            IF(INST_SOCKET_TCPCONNECT.ERROR = TRUE) THEN    //When performing abnormal

                Stage := 0;                                     //change Stage 0
                INST_SOCKET_TCPCONNECT(                     //instance reset
                    REQ             := FALSE,
                    SRCPORT         := 8003,
                    TIMEOUT         := 5,
                    DSTIPADDR       := DestIpAddr,
                    DSTPORT         := 8003,
                    OPENSOCKCH   := SocketServiceCh);

            END_IF;

        2:  //data sending
            INST_SOCKET_TCPSEND(                                //instance execution
                REQ         := TRUE,
                SOCKCH      := SocketServiceCh,                 //Connection confirmed
socket service channel
                SENDDATA    := TcpBuffer,                       // send data buffer
                SIZE        := 1400);                           //sending data size

            IF(INST_SOCKET_TCPSEND.DONE = TRUE) THEN        //in normal performance
                Stage := 3;                                     //change Stage 3
                INST_SOCKET_TCPSEND(                            //instance reset
                    REQ         := FALSE,
                    SOCKCH      := SocketServiceCh,
                    SENDDATA    := TcpBuffer,
                    SIZE        := 1400);

            END_IF;

            IF(INST_SOCKET_TCPSEND.ERROR = TRUE)THEN            //When performing
abnormal
                Stage := 0;                                     //change Stage 0
```

```
                INST_SOCKET_TCPSEND(                            //instance reset
                    REQ         := FALSE,
                    SOCKCH      := SocketServiceCh,
                    SENDDATA    := TcpBuffer,
                    SIZE        := 1400);

                INST_SOCKET_CLOSE(                          //instance execution
                    REQ         := TRUE,
                    SOCKCH      := SocketServiceCh);            //Socket service channel
to close

                INST_SOCKET_CLOSE(
                    REQ         := FALSE,
                    SOCKCH      := SocketServiceCh);            //instance reset

            END_IF;

        3:  //data receive
            INST_SOCKET_TCPRECV(                                //instance execution
                REQ                 := TRUE,
                SOCKCH              := SocketServiceCh,         //Connection confirmed
socket service channel
                TIMEOUT             := 3,                       //Receive Standby for
up to 3 sec
                SIZE                := 1400,                    //specify the maximum
receive size (cannot set more than the buffer size)
                RECVDATA            := TcpBuffer);              //received data buffer

            IF(INST_SOCKET_TCPRECV.DONE = TRUE)THEN         //in normal performance
                Stage := 2;                                     //change Stage 2
                INST_SOCKET_TCPRECV(                            //instance reset
                    REQ         :=FALSE,
                    SOCKCH      := SocketServiceCh,
                    TIMEOUT     := 3,
                    SIZE        := 1400,
                    RECVDATA    := TcpBuffer);

            END_IF;
```

```
                IF(INST_SOCKET_TCPRECV.ERROR = TRUE)THEN            //When      performing
abnormal
                    Stage := 0;                                     //change Stage 0
                    INST_SOCKET_TCPRECV(
                        REQ         := FALSE,
                        SOCKCH      := SocketServiceCh,
                        TIMEOUT     := 3,
                        SIZE        := 1400,
                        RECVDATA    := TcpBuffer);

                    INST_SOCKET_CLOSE(                          //instance execution
                        REQ         := TRUE,
                        SOCKCH      := SocketServiceCh);            //Socket  service  channel
to close

                    INST_SOCKET_CLOSE(                          //instance reset
                        REQ         := FALSE,
                        SOCKCH      := SocketServiceCh);


                END_IF;

            END_CASE;
        END_IF;
END_IF;
```

14.4.5 Command operation timing chart



(1) Start command operation with Rising Edge of Execute, initialize ErrorID
(2) Done occurs when normal operation is completed after the command operation (Busy) state, Output Parameter is valid (updated)
(3) When Execute Falling, Done Flag Reset.
(4) In case of command operation (Busy) state, even if Execute falls, the previous operation is continued
(5) Done occurs after completion of command operation (Busy), and if Execute status is 0, Done is maintained during one call
(6) Even if Rising Edge of Execute occurs during command operation (Busy), the previous operation is continued
(7) Start command operation with Rising Edge of Execute, initialize ErrorID
(8) Error occurs when abnormal operation is completed after command operation (Busy) status, ErrorID update
(9) Execute Falling 시 Error Flag Reset
(10) Done occurs after completion of command operation (Busy), and if Execute status is 0, Error is maintained during one call

※ Follows PLC Open FB operation standard

14.4.6 SOCKET SERVICE Common Error

ID	Error name	Contents	Note
0	No Error	Normal status	
1	INVALID_PARAMETER	Command input parameter error	Normal range parameter input
2	SYSTEM_RESERVED_PORT	Attempt to use the system reserved port	Change Local Port input value
3	ALREADY_USED_CH	Attempt to assign channels already in use	Use after checking the usage status of the service channel
4	WAIT_TIMEOUT	Command timeout occurrence	● Checking the destination side status ● SOCKET_TCPRECV, SOCKET_UDPRECV ■ Check whether the destination side data is sending ■ Check whether the destination side is connected ● SOCKET_TCPCONNECT ■ Check the destination side (server) access-allowed standby status ● SOCKET_TCPACCEPT Check the status of the destination side (client) connection attempt
5	COMMAND_INSTANCE_POOL_FULL	Exceeded command instance usage	Decrease the frequency of command execution
7	SOCKET_CH_NOT_INITIALIZE	Control attempts through uninitialized channels	● Use after initializing the service channel according to the purpose ■ UDP: SOCKET_UDPCREATE ■ TCP(server): SOCKET_TCPLISTEN & SOCKET_TCPACCEPT TCP(client): SOCKET_TCPCONNECT
9	SOCKET_NO_DATA	There is no received data	
10	SOCKET_SERVICE_PROCEDURE	Common errors that occur during service processing	● Command operation retry through exception handling ● Review input values System check
11	SOCKET_CLOSED	Attempts to process a command on a socket that has already been closed, or a connection that has been aborted by the peer.	Action to prevent SOCKET_CLOSE from operating for the service channel while SOCKET_TCPACCEPT, SOCKET_TCPCONNECT, SOCKET_TCPRECV, SOCKET_UDPRECV is in operation.
12	INVALID_ADDRESS_OR_PORT	Input IP, PORT value abnormal, unavailable status	● When a local port is specified, if the system internal initialization for the port is in progress, the error may persist for up to 60 seconds, and the command operation is retried through exception handling. ● Check whether a port that is already in use is specified Local Port normal input value, check the range
13	CONNECTION_TIMEDOUT	Connection time out occurred	Check remote side status
14	NETWORK_UNREACH	Network connection disable	● Check Local device status Check settings such as gateway, subnet mask, and route table
15	HOST_DOWN	Host device error	● Check Local device status Check for abnormalities in cables and connection paths , check H/W

16	HOST_UNREACH	Host connection disable	● Check Local device status Check for abnormalities in cables and connection paths , check H/W
17	ALREADY_CONNECTED	Reconnection of an already established socket	Retry after SOCKET_CLOSE of the corresponding socket service channel
18	CONNECTION_REFUSED	Connection denied	Check connection denied status of remote side
19	CONNECTION_ABORTED	Aborted while connecting	Abort remote side connection during connection

Chapter 15 Built-in Cnet Communications

15.1 Overview

Cnet with built-in motion controllers is a serial communication device that supports RS-232C and RS-485 protocols and has the following characteristics.

15.1.1 Characteristics

Cnet I/F is a serial communication device supporting RS-232C and RS-485 protocols and has the following characteristics.

1) It offers a user-friendly experience and is easy to connect with the products of other companies.
2) Using the XG5000 that operates in the Windows environment, a user can directly write communication speed and communication mode (protocol), and it is easy to connect with the products of other companies.
3) It provides the CnetI/F of RS-232C 1 port/ RS-485.
4) Read/write variable is possible using a dedicated protocol.
5) It provides a dedicated communication function for multi-drop configuration that con connect up to 32 units when RS-485 is used.
6) Various communication speed settings are possible.
   - RS-232C: 1200bps ~ 115,200bps / RS-485: 1200bps ~ 115,200bps available
7) 1:1/1: N communication is possible.
8) It supports full-duplex (RS-232C) and half-duplex (RS-485) communication methods.
9) It provides dedicated communication (user defined communication and XGT client/server communication), and Modbus client/server functions.
10) It provides a client mode (LS bus) for communication dedicated for LS Industrial Systems Inverter.
11) Smart server automatically recognizes the protocol (LS ELECTRIC dedicated protocol, Modbus RTU/ ASCII) and works.
12) It provides a repeater mode to convert RS-232C into RS-485 or uses it as an isolated repeater.
13) It has a built-in termination resistor, and thus termination can be set in the default parameter, when the termination resistance is needed.

Notice

The serial communication function is supported only for communication type products. (XMC-E32C)

15.2 Confirm before product check

To use the serial communication of the stand-alone motion controller, you need to prepare some things in advance.
Please check below for the correct use of the product.

15.2.1 Advance Preparation

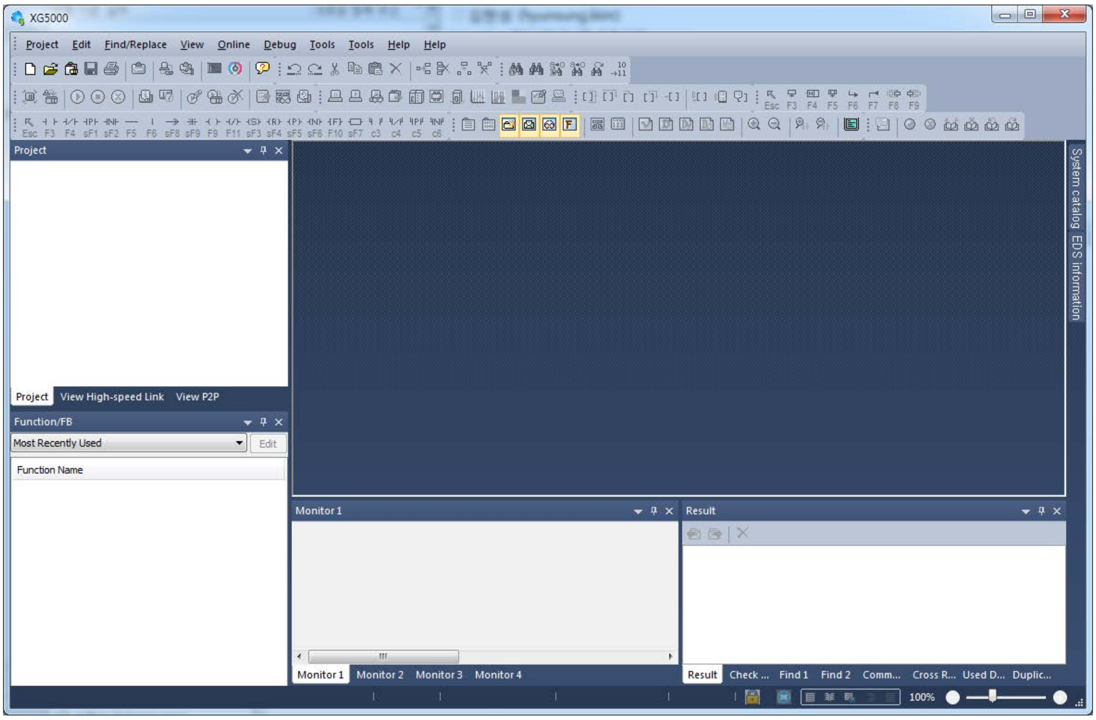
(1) Please download the XG5000 from LS Industrial Systems website below.
Internet web address: http://www.ls-electric.com/

(2) Please download the XG5000 from LS Industrial Systems website below.
The model name of the cable is as follows.

□ USB: USB-301A, □ RS-232C: K1C-050A

15.2.2 Install the XG5000

XG5000 is dedicated software for setting of basic parameter, writing of frame and diagnosis of all communication module including the Cnet I/F module. The following figure is initial screen of XG5000.



15.2.3 Check the product version

Before using the Cnet module, check the version of module.

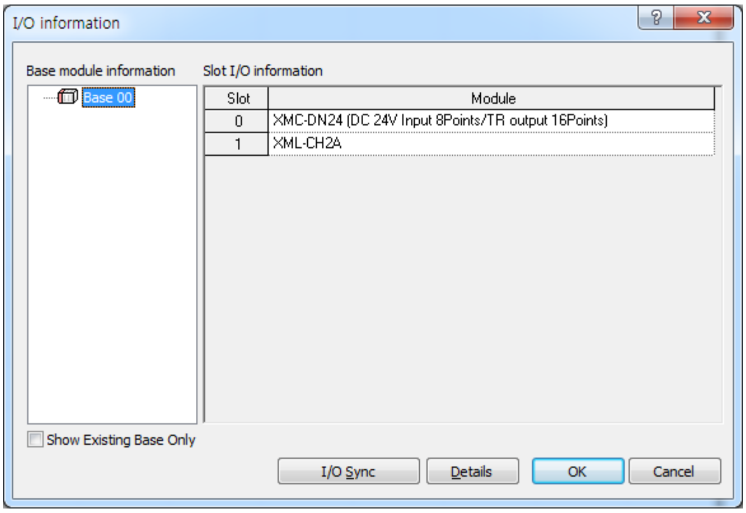
1) Check through XG5000

Here describes on how to read communication module information by online connection to communication module.
If interface with CPU is normal, it is available to get the following information.

(a) Execute the XG5000

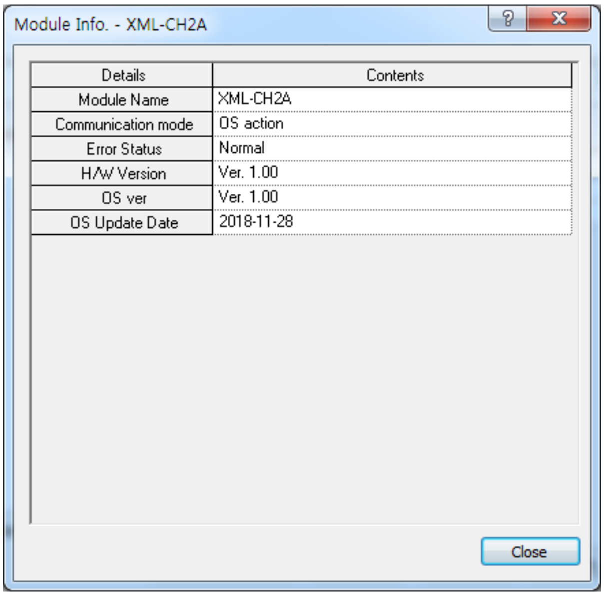
(b) Connect with XMC through online connection.

(c) If connection with XMC is established, execute the system diagnosis.
Execute the Communication module information in the system diagnosis screen.
Software information shows at the right bottom of screen.



(d) When click the XML-CH2A module for which you want to check the stand-alone motion controller product information, [Serial Communication Module Information] screen appears.

(e) Check the product version at the bottom right of this screen..



2) Check version written on the case label of the product

Each communication module has the product information label on the case. If online check is not possible, see the label on the case after removing it from base. Label is in the back of the case and type name of product and version information is indicated.

15.3 Specifications

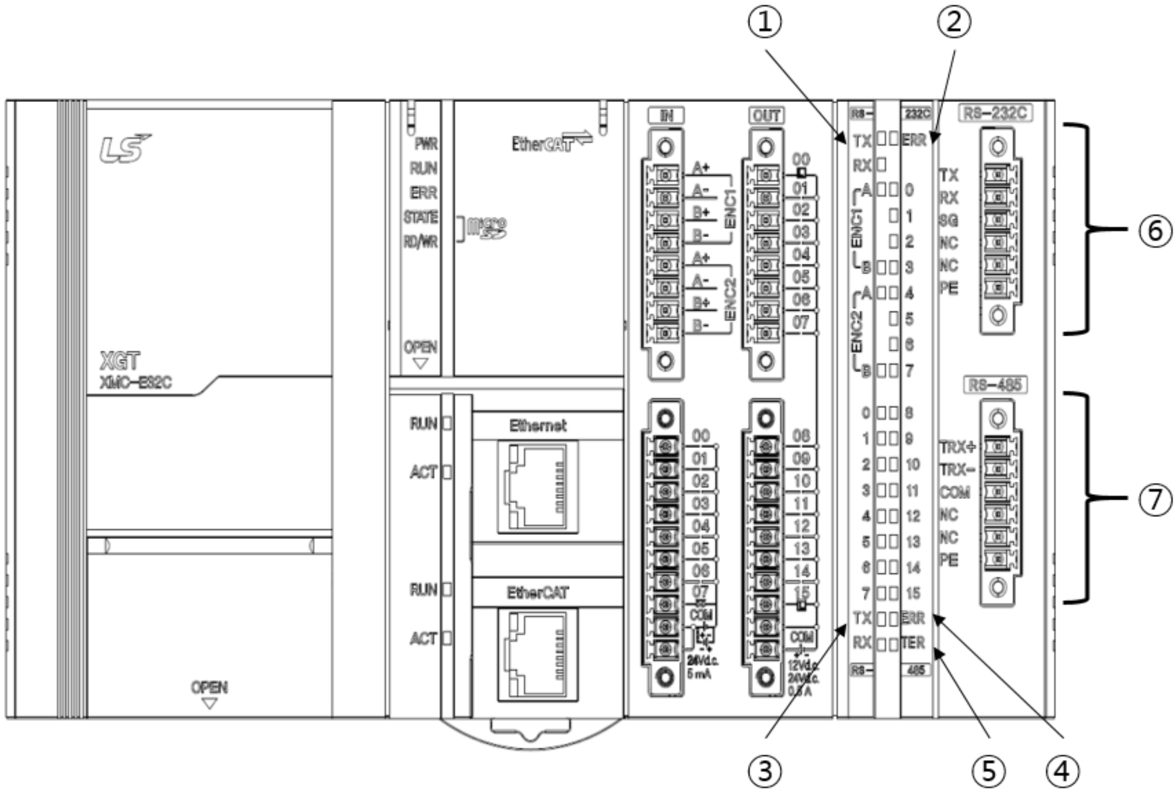
15.3.1 Performance Specifications

<table>
<tr><th colspan="2" rowspan="2">Item</th><th colspan="2">Specifications</th></tr>
<tr><th>Channel 1</th><th>Channel 2</th></tr>
<tr><td colspan="2">Serial communication method</td><td>RS-232C</td><td>RS-485</td></tr>
<tr><td colspan="2">Modem connection function</td><td>-</td><td>-</td></tr>
<tr><td rowspan="2">Operation mode(define operation for each channel)</td><td>P2P</td><td colspan="2">Operates as a communication client
- XGT dedicated protocol client
- Modbus ASCII/RTU client
- User defined communication
- LS bus client Note 1)</td></tr>
<tr><td>Server</td><td colspan="2">- XGT dedicated protocol server
- Modbus ASCII/RTU server</td></tr>
<tr><td rowspan="3">Data type</td><td>Data bit</td><td colspan="2">7 or 8</td></tr>
<tr><td>Stop bit</td><td colspan="2">1 or 2</td></tr>
<tr><td>Parity</td><td colspan="2">Even/Odd/None</td></tr>
<tr><td colspan="2">Synchronous method</td><td colspan="2">Asynchronous method</td></tr>
<tr><td colspan="2">Transmission speed(bps)</td><td colspan="2">1,200/2,400/4,800/9,600/19,200/38,400/57,600/115,200 bps can be selected</td></tr>
<tr><td colspan="2">Station No. setting</td><td colspan="2">Setting range: 0~255 Note 2)
Maximum number of stations: 32</td></tr>
<tr><td colspan="2">Transmission distance</td><td>Max. 15m</td><td>Max. 1200m</td></tr>
<tr><td colspan="2">Diagnosis function</td><td colspan="2">- Check operation according to LED status.
- XG5000 Diagnosis Service: Frame monitor, service status, loopback test, PLC history.</td></tr>
</table>

Notice

Note 1) It means a dedicated protocol with LS Inverter.
Note 2) In the client/server configuration, up to 32 stations can be set, and the station number setting ranges from 0 to 255.

15.3.2 Names and Roles of Built-in Cnet Components



Number	Name	Contents
①	TX LED RX LED	•RS-232C data LED ON/OFF: Transmitting/receiving communication data OFF: Standby for communication
②	ERR LED	•RS-232C error LED ON/OFF: Communication error OFF: No communication error
③	TX LED RX LED	•RS-485 data LED ON/OFF: Transmitting/receiving communication data OFF: Standby for communication
④	ERR LED	•RS-485 error LED ON/OFF: Communication error OFF: No communication error
⑤	TER LED	•Termination resistor LED ON: Termination resistor ON OFF: Termination resistor OFF
⑥	RS-232C communication connector	•Connector for connecting RS-232C communication signals
⑦	RS-485 communication connector	•Connector for connecting RS-485 communication signals

15.3.3 Cable Specifications

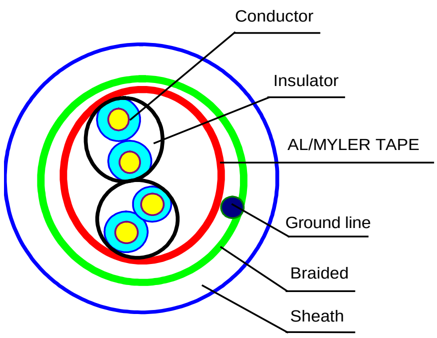
When communication is done using the RS-485 channel, a twisted pair cable for RS-422 should be used for excellent signal transmission and control characteristics.
The following shows the recommended cable specifications.

- Item name: Low-capacity LAN interface cable
- Model name: LIREV-AMESB
- Specification: 2P X 22AWG (D/0.254 TA)
- Manufacturer: LS Industrial Systems

	Test item	Unit	Characteristics	Test conditions
Electric characteristics	Conductor resistance	Ω/km	59	Room temp.
	Withstanding voltage(DC)	V/1min	Withstands for 1 min. at 500V	In air
	Insulation resistance	MΩ-km	1,000	room temp
	Static electricity capacity	pF/M	45 or less	1kHz
	Characteristics impedance	Ω	120 ± 12	10MHz

	Item			Single Cable
Characteristics of appearance.	Conductor	Cores	Pair	2
		Size	AWG	22
		Composition	NO./mm	1/0.643
		Outer dia.	mm	0.643
	Insulator	Thickness	mm	0.59
		Outer dia.	mm	1.94

Standard of Twisted Pair Cable



Structure

15.3.4 Termination Resistor

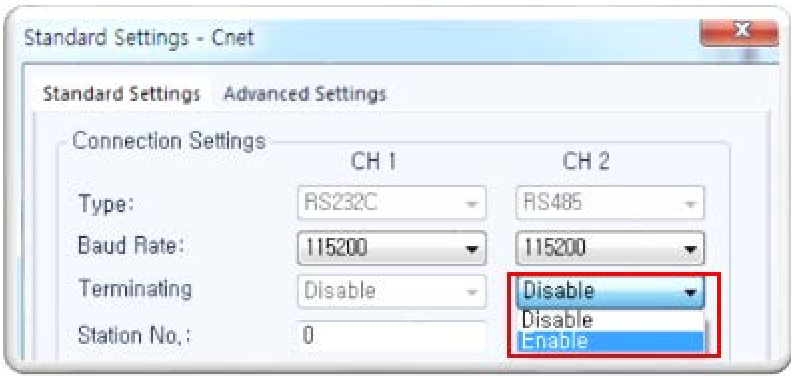
When communication is done via the RS-485 channel with built-in motion controller, a termination resistor should be connected from the outside.

The termination resistor is used to prevent signal distortion due to the reflected wave of the cable when use for long distance communication, the resistance (120Ω, 1/2W) equal to the characteristic impedance value of the cable should be connected to the end of the network.

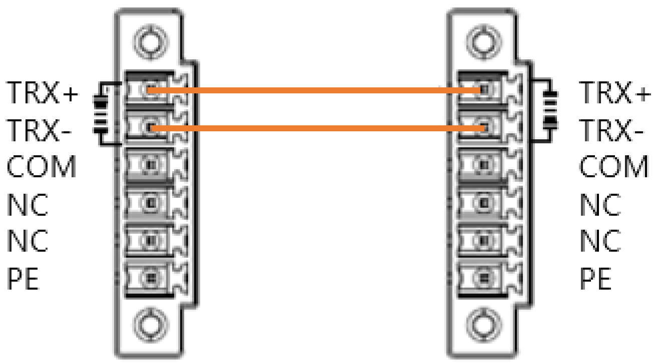
When using the recommended cable, connect the termination resistor to both ends of the line.
Even when using cables other than the recommended cable, connect a resistor with the same value as the characteristic impedance value to both ends of the line.

Recommended termination resistance: 1/2W, 120Ω, 5% error

1) Enabling fundamental parameter termination resistor



2) Disabling fundamental parameter termination resistor



[Termination reistor connection diagram in the RS-485 interface]

15.4 Perfomance Specifications

15.4.1 Operation Mode Setting

The operation mode of Cnet is decided by the standard setting parameters. It operates separately from each communication port with the operation modes available as described below.

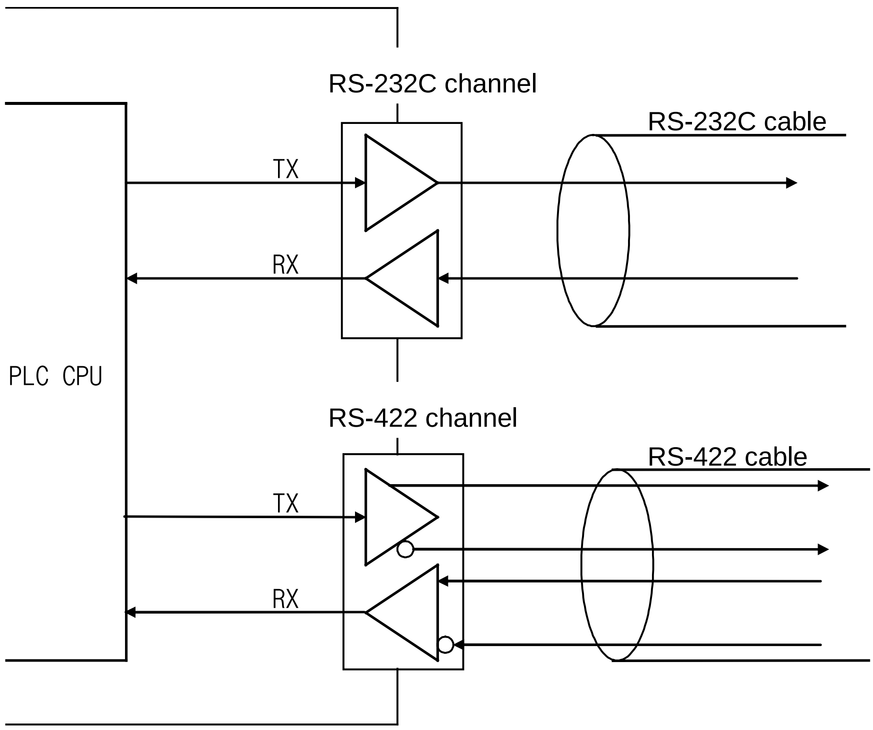
1) Server Mode
Operates as a server in the network. XGT server and Modbus server are optional.

(1) XGT server: dedicated communication protocol supported, memory Read, Write available.
(2) Modbus server: Modbus protocol supported, RTU/ASCII type optional.
(3) Setting necessary for conversion between Modbus protocol memory area and XGT memory area.
(4) Smart server: Smart server automatically analyzes the protocol (XGT, Modbus ASCII/RTU) automatically and operates as analyzed server.

2) P2P (Client) Mode
(1) Operates as a client in the network.
(2) Support dedicated communication protocol, Modbus protocol and LS inverter dedicated communication (LS BUS client).
(3) Up to 64 communication blocks can be specified for 1 Cnet module to define the independent operation.

15.4.2 Operation by Channel

Each communication port operates independently to allow simultaneous TX, RX in separate transmission specifications. Therefore, transmission specifications can be set per RS-232C and RS-485 channel, and the operation is started and stopped according to channels. Data flow of each channel is as below.



Notes

[Note 1] While in operation, mode change is not available. In order to change the mode, download the standard communication parameters and reset the communication module.

[Note 2] Although do not Reset the PLC separately, the changed mode will be applied when the download is completed.

15.4.3 Channel Operation in Repeater Mode

1) Repeater Mode

Repeater mode is a function that transmits data received from each channel to another channel.

(1) Repeater mode of Cnet is not offers Auto Speed

(2) Communication setting is equal to each channels and modem type is fixed to null modem.

(3) While in operation, mode change is not available. If you want to change the mode, download the standard communication parameter and RESET the module.

15.4.4 Serial Connection Methods

1) Connection method when using built-in RS-232C

Three-wire system is used for connection, as shown below.

Cnet		Pin number and signal direction	Computer/communication device
Pin number	Name		Name
1	TX		TXD
2	RX		RXD
3	SG		SG

2) Connection method when using built-in RS-485

Pin number	Name	Pin number and signal direction	External communication device
1	TRX+		485+
2	TRX-		485-
3	COM		COM

15.5 Installation and operation

15.5.1 Parameter information for communication mode

Communication functions available in Cnet with built-in motion controller can be classified into several types, as shown below.

1) Default setting parameters

This is the part that sets the media information, hardware information and basic protocol information of serial communication of Standalone Motion Controller

<table>
<tr><th rowspan="2">Parameter</th><th rowspan="2">Sub-menu</th><th rowspan="2">Setting item</th><th rowspan="2">Setting range</th><th colspan="2">Possibility of settings</th><th rowspan="2">Notice</th></tr>
<tr><th>Client</th><th>Server</th></tr>
<tr><td rowspan="10">Default settings</td><td rowspan="4">Connection settings</td><td>Communication type</td><td>RS-232C
RS-485</td><td>Possible</td><td>Possible</td><td>-</td></tr>
<tr><td>Communication speed(bps)</td><td>1200 ~ 115,200</td><td>Possible</td><td>Possible</td><td>-</td></tr>
<tr><td>Termination resistance</td><td>Enabled/Disabled</td><td>Possible</td><td>Possible</td><td></td></tr>
<tr><td>Station number</td><td>XGT communication: 0~255
Modbus: 0~255</td><td>Possible</td><td>Possible</td><td>Station number is not significant when setting client</td></tr>
<tr><td rowspan="6">Operation mode</td><td>P2P used</td><td rowspan="5">One mode is selectable</td><td>Possible</td><td>-</td><td>-</td></tr>
<tr><td>XGT server</td><td>-</td><td>Possible</td><td>-</td></tr>
<tr><td>Modbus ASCII server</td><td>-</td><td>Possible</td><td>-</td></tr>
<tr><td>Modbus RTU server</td><td>-</td><td>Possible</td><td>-</td></tr>
<tr><td>Smart server</td><td>-</td><td>Possible</td><td></td></tr>
<tr><td>Repeater mode</td><td>Enabled/Disabled</td><td>-</td><td>-</td><td>All services are stopped when repeater mode is set</td></tr>
<tr><td rowspan="8">Advanced settings</td><td rowspan="5">Connection settings</td><td>Data bit</td><td>7, 8</td><td>Possible</td><td>Possible</td><td>In Modbus ASCII, the number of data bits is 7.</td></tr>
<tr><td>Stop bit</td><td>1, 2</td><td>Possible</td><td>Possible</td><td>*Note 5)</td></tr>
<tr><td>Parity bit</td><td>NONE,ODD,EVEN</td><td>Possible</td><td>Possible</td><td>-</td></tr>
<tr><td>Parity reception error *Note 4)</td><td>Allowed/ Not allowed</td><td>Possible</td><td>Possible</td><td>-</td></tr>
<tr><td>Modem type</td><td>Null modem</td><td>Possible</td><td>Possible</td><td>-</td></tr>
<tr><td rowspan="3">Time settings</td><td>Response waiting time</td><td>0~50 (x 100ms)</td><td>Possible</td><td>-</td><td>*Note 1)</td></tr>
<tr><td>Delay time</td><td>0~255 (x 10ms)</td><td>Possible</td><td>Possible</td><td>*Note 2)</td></tr>
<tr><td>Inter-character waiting time</td><td>0~255 (x 10ms)</td><td>Possible</td><td>Possible</td><td>*Note 3)</td></tr>
</table>

Notice

1) Response waiting time: It means the waiting time until the reception of response frame after the completion of frame transmission.
   (1) Operation setting: It can e set when P2P is used in operation mode.
   (2) Response waiting time = Basic response waiting time + (response waiting time set value x 100ms) + inter-character waiting time
   (3) Basic response waiting time by communication speed
      1) 9,600~115,200bps: 100ms
      2) 7,200~2,400bps: 200ms
      3) 1,800~1,200bps: 400ms

2) Delay time setting: It means transmitting a frame after the delay time set by a user.
   (1) Client operation setting: Client operation can be set when communication type is RS-485.
   (2) Server operation setting: It is possible to transmit a frame after the delay time for the response of the server to be set.

3) Inter-character waiting time: It means the spacing between characters in one frame processed as one frame in case of a character that comes in the set time in one frame.
   (1) Operation setting: It can be set regardless of operation mode.

4) Parity reception error: When [Accept] is selected, data can be received even if there is an error in the reception parity bit.

5) Stop bit: It means the end of a single packet and checks the stop bit set at the time of data transmission. If the stop bit of the received data is smaller than the set stop bit, data cannot be received normally. In order to receive data normally, the stop bit should be configured identically.
   (1) When stop bit is set to 1, all frames can be received.
   (2) Normal communication may not be possible in communication with equipment where stop bit is set to 2.
   (3) When stop bit is set to 2, data of stop bit 1 cannot be received.
   (4) The parameters should be set the same for normal communication.

2) P2P setting parameters

This is to set communication frames.

Parameter	Sub-menu	Setting item	Setting range and contents	Possibility of settings(client)				
				XGT	Modbus ASCII	Modbus RTU	Inverter Dedicated communication	User frame definition
P2P	Communication module settings	Base	0~7	Possible	Possible	Possible	Possible	Possible
		Slot	0~11	Possible	Possible	Possible	Possible	Possible
	P2P channel	P2P deriver	User frame definition	-	-	-	-	Possible
			XGT client	Possible	-	-	-	-
			Modbus ASCII Client	-	Possible	-	-	-
			Modbus RTU Client	-	-	Possible	-	-
			LS Bus client*Note 5)	-	-	-	Possible	-
	P2P block	Channel	1, 2	Possible	Possible	Possible	Possible	Possible
		P2P function	READ	Possible	Possible	Possible	Possible	-
			WRITE	Possible	Possible	Possible	Possible	-
			SEND	-	-	-	-	Possible
			RECEIVE	-	-	-	-	Possible
		Start condition*Note 1)		Possible	Possible	Possible	Possible	Possible
		Method	Individual	Possible	Possible	Possible	-	-
			Continuous	Possible	Possible	Possible	Possible	-
		Data type	Bit	Possible	Possible	Possible	-	-
			Word	Possible	Possible	Possible	Possible	-
			1 bye	Possible	-	-	-	-
			2 bytes	Possible	-	-	-	-
			4 bytes	Possible	-	-	-	-
			8 bytes	Possible	-	-	-	-
		Number of variables*Note 2)		Possible	Possible	Possible	-	-
		Data size*Note 2)		Possible	Possible	Possible	Possible	-
		Partner station number		Possible	Possible	Possible	Possible	-
		Frame		-	-	-	-	Possible
		Settings*Note 3)		Possible	Possible	Possible	Possible	Possible
	User frame definition	Add group	Group name	-	-	-	-	
			Frame type: Transmission	-	-	-	-	Possible
			Frame type: Reception	-	-	-	-	Possible
	Frame*Note 4	Edit group	Group name	-	-	-	-	Possible
		Delete group		-	-	-	-	Possible
		Add frame	HEAD	-	-	-	-	Possible
			TAIL	-	-	-	-	Possible
			BODY	-	-	-	-	Possible

Notice

1) Start condition in user-defined frame communication is selectable only when the P2P function is SEND.
2) The number of variables and data size can be set only in the case of continuous mode in XGT client and Modbus ASCII/RTU client.
3) Settings in user-defined frame communication can be established only when the fixed size parameter or variable size parameter is selected.
4) Frame settings can be entered after the frame type and group name of the user frame definition are set.
5) LS Bus client is a function provided by the B type Cnet I/F module.

15.5.2 Device Information

1) Default settings

Communication type	Communication speed	Data bit	Stop bit	Parity bit	Modem type	Station number	Response waiting time	Delay time	Inter-character waiting time
RS-232C	1200 ~115,200	7~8	1~2	NONE~ ODD	Null modem	0~255	0~50	0~255	0~255
RS-485	1200 ~115,200	7~8	1~2	NONE~ ODD	Null modem	0~255	0~50	0~255	0~255

2) Modbus settings

Channel	Modbus Used/unused	Settings	Item	XMC-E32Cdefault
Channel 1	P2P used	Disable	-	-
	XGT server	Disable	-	-
	Modbus ASCII server	Enable	Bit Read Area Start Address:	%IX0.0.0
			Bit write Area start Address:	%QX0.0.0
			Word Read Area Start Address:	%MW0
			Word Write Area Start Address:	%MW100
	Modbus RTU server	Enable	Bit Read Area Start Address:	%IX0.0.0
			Bit Write Area Start Address:	%QX0.0.0
			Word Read Area Start Address:	%MW0
			Word Write Area Start Address:	%MW100
Channel 2	P2P used	Disable	-	-
	XGT server	Disable	-	-
	Modbus ASCII server	Enable	Bit Read Area Start Address:	%IX0.0.0
			Bit Write Area Start Address:	%QX0.0.0
			Word Read Area Start Address:	%MW0
			Word Write Area Start Address:	%MW100
	Modbus RTU server	Enable	Bit Read Address Start Address:	%IX0.0.0
			Bit Write Area Start Address:	%QX0.0.0
			Word Read Area Start Address:	%MW0
			Word Write Area Start Address:	%MW100

3) P2P channel settings

Operation mode	P2P deriver	TCP/UDP	Client/ Server	Partner station port	Destination port IP address
XGT server	-	-	-	-	-
P2P used	XGT client	-	-	-	-
	User frame definition	-	-	-	-
	LS Bus client (CH2: RS-485)	-	-	-	-
	Modbus ASCII client				
	Modbus RTU client	-	-	-	-

4) P2P block settings

<table>
<tr><th>Operation mode</th><th>P2P server</th><th>P2P function</th><th>Condition flag</th><th>Command type</th><th>Data type</th><th>Number of variables</th><th>Data size</th><th>Partner station number</th><th>Read area</th><th>Storage area</th><th>Address</th></tr>
<tr><td>XGT server</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td rowspan="26">P2P used</td><td rowspan="6">XGT client</td><td rowspan="3">Read</td><td rowspan="23">XGT device</td><td>Individual</td><td>BIT</td><td>1 ~ 4</td><td rowspan="2">Disable (blank)</td><td rowspan="24">0~63</td><td rowspan="6">XGT device</td><td rowspan="10">XGT device</td><td rowspan="26">N device calculation method</td></tr>
<tr><td>Individual</td><td>B/W/D/L</td><td>1 ~ 4</td></tr>
<tr><td>Continuous</td><td>B/W/D/L (XGI)</td><td>Disable (1)</td><td>1 ~ 120</td></tr>
<tr><td rowspan="3">Write</td><td>Individual</td><td>BIT</td><td>1 ~ 4</td><td rowspan="2">Disable (blank)</td></tr>
<tr><td>Individual</td><td>B/W/D/L (XGI)</td><td>1 ~ 4</td></tr>
<tr><td>Continuous</td><td>B/W/D/L (XGI)</td><td rowspan="17">Disable(1)</td><td>1 ~ 120</td></tr>
<tr><td rowspan="8">Modbus ASCII client</td><td rowspan="4">Read</td><td>Individual</td><td>BIT</td><td rowspan="2">Disable (blank)</td><td>00000 ~ 19999</td></tr>
<tr><td>Individual</td><td>WORD</td><td>30000 ~ 49999</td></tr>
<tr><td>Continuous</td><td>BIT</td><td>1 ~ 976</td><td>00000 ~ 19999</td></tr>
<tr><td>Continuous</td><td>WORD</td><td>1 ~ 61</td><td>30000 ~ 49999</td></tr>
<tr><td rowspan="4">Write</td><td>Individual</td><td>BIT</td><td rowspan="2">Disable (blank)</td><td rowspan="4">XGT device</td><td>00000 ~ 09999</td></tr>
<tr><td>Individual</td><td>WORD</td><td>40000 ~ 49999</td></tr>
<tr><td>Continuous</td><td>BIT</td><td>1~944</td><td>00000 ~ 09999</td></tr>
<tr><td>Continuous</td><td>WORD</td><td>1~59</td><td>40000 ~ 49999</td></tr>
<tr><td rowspan="8">Modbus RTU client</td><td rowspan="4">Read</td><td>Individual</td><td>BIT</td><td rowspan="2">Disable (blank)</td><td>00000 ~ 19999</td><td rowspan="4">XGT device</td></tr>
<tr><td>Individual</td><td>WORD</td><td>30000 ~ 49999</td></tr>
<tr><td>Continuous</td><td>BIT</td><td>1 ~ 2000</td><td>00000 ~ 19999</td></tr>
<tr><td>Continuous</td><td>WORD</td><td>1 ~ 125</td><td>30000 ~ 49999</td></tr>
<tr><td rowspan="4">Write</td><td>Individual</td><td>BIT</td><td rowspan="2">Disable (blank)</td><td rowspan="4">XGT device</td><td>00000 ~ 09999</td></tr>
<tr><td>Individual</td><td>WORD</td><td>40000 ~ 49999</td></tr>
<tr><td>Continuous</td><td>BIT</td><td>1~1968</td><td>00000 ~ 09999</td></tr>
<tr><td>Continuous</td><td>WORD</td><td>1~123</td><td>40000 ~ 49999</td></tr>
<tr><td rowspan="2">User frame definition</td><td>SEND</td><td>Transmitting body</td><td>-</td><td>-</td><td>1 ~ 1024</td><td>XGT device variable size parameter</td><td>-</td></tr>
<tr><td>RECEIVE</td><td>-</td><td>Receiving body</td><td>-</td><td>-</td><td>-</td><td>-</td><td>Memory specification</td></tr>
<tr><td rowspan="2">LS Bus client</td><td>Read</td><td rowspan="2">XGT device</td><td rowspan="2">Continuous</td><td rowspan="2">WORD</td><td rowspan="2">1</td><td rowspan="2">1 ~ 8</td><td rowspan="2">0~255</td><td>Inverter address value</td><td>XGT device</td></tr>
<tr><td>Write</td><td>XGT device</td><td>Inverter address value</td></tr>
</table>

1) User definition frame

Group	Frame	Segment	Remarks
Transmission 1	HEAD	Numerical constant	Max. 10Byte 12345678901234567890
		String constant	1234567890 (Internally registered as 3132..30)
	TAIL	Numerical constant	Max. 10Byte 12345678901234567890
		String constant	1234567890 (Internally registered as 3132..30)
		BCC	-
	BODY	Numerical constant	Max. 10Byte 12345678901234567890
		String constant	1234567890 (Internally registered as 3132..30)
		Variable size parameter	Up to 4 available
Reception 1	HEAD	Numerical constant	Max. 10Byte 12345678901234567890
		String constant	1234567890 (Internally registered as 3132..30)
	TAIL	Numerical constant	Max. 10Byte 12345678901234567890
		String constant	1234567890 (Internally registered as 3132..30)
		BCC	-
	BODY	Numerical constant	Max. 10Byte 12345678901234567890
		String constant	1234567890 (Internally registered as 3132..30)
		Fixed size parameter	Up to 4 available
		Variable size parameter	Only one variable size parameter can be set. Therefore, segment cannot be added to the variable size parameter.
No restrictions on the number of groups, frames and segments, except for size(0x4B00)			

15.5.3 Device Area Information

Area	Range	Size(Word)	Remarks
M	MW0 – MW1048575	1048576	Read/Write/Monitor available
K	KW0 – KW9215	9216	Read/Monitor available
F	FW0 – FW65535	65536	Read/Monitor available
L	LW0 – LW11263	11264	Read/Write/Monitor available
U	UW0.0.0– UW0.15.31	512	Read/Write/Monitor available
I	IW0.0.0 –IW127.15.3	8192	Read/Write/Monitor available
Q	QW0.0.0– QW127.15.3	8192	Read/Write/Monitor available

15.6 Cnet Communication System Configuration

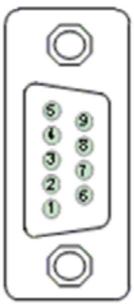
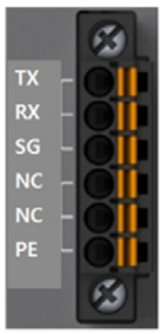
If you use Cnet of motion controller, you can configure various types of communication networks by connecting with PLC and PC of its company and other companies. Below are some examples of network system configurations.

1:1 Connection to PC (HMI) (No Modem)

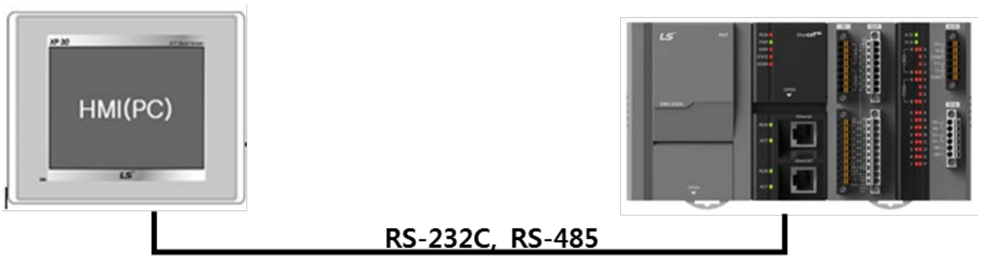
(1) The basic units of PC (HMI) and motion controller are connected by RS-232C or RS-485 channel, and PC (HMI) and PLC are connected 1:1 without model in this system.
(2) Most HMI(PC) operate as client stations, and XMC basic units act as server stations which respond to requests from HMI(PC).
(3) Since there is no model, communication distance is up to 15m when it is through RS-232C channel and up to 1500m when it is through RS-485.

1) 1:1 connection with general-purpose PC

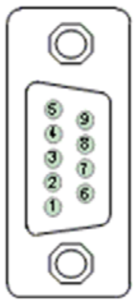
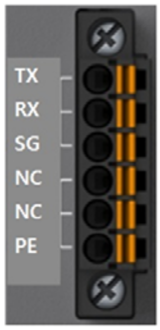
• Wiring method

outward appearance on PC side	PC Part Pin number	Connection number and signal direction	XMC basic unit Pin number	XMC basic unit Signal name	RS-232C terminal
 Female Type	1		1	TX	 TX RX SG NC NC PE
	2 (RXD)		2	RX	
	3(TXD)		3	SG	
	4		4	NC	
	5(GND)		5	NC	
	6		6	PE	
	7				
	8				
	9				

2) 1:1 connection with monitoring devices such as XGT Panel



• Wiring method (RS-232C)

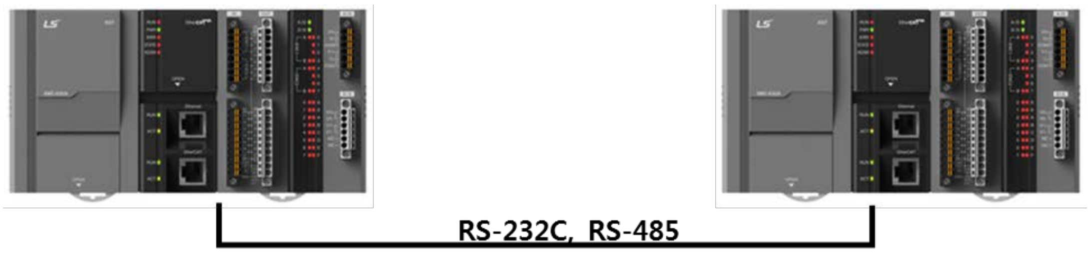
outward appearance on PC side	XP part Pin number	Connection number and signal direction	XMC basic unit Pin number	XMC basic unit Signal number	RS-232C terminal
 Female Type	1		1	TX	 TX RX SG NC NC PE
	2 (RXD)		2	RX	
	3(TXD)		3	SG	
	4		4	NC	
	5(GND)		5	NC	
	6		6	PE	
	7				
	8				
	9				

Note) In the case of PMU, 4 and 6, and 7 and 8 should be short-circuited for use.

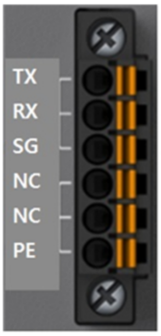
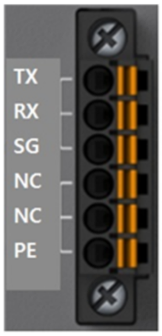
• Wiring method (RS-485)

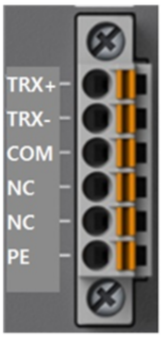
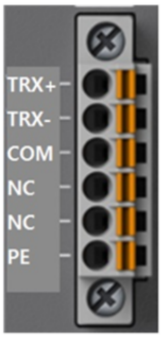
PMU side	**Connection number and signal direction**	**XMC RS-485 terminal**
485+	←→	TRX+
485-	←→	TRX-

3) 1:1 Connection to XMC Basic Unit



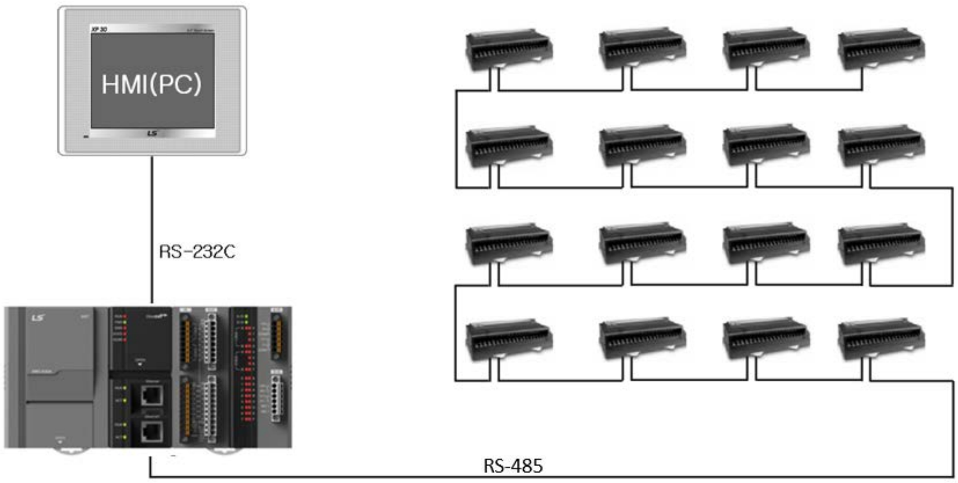
• Wiring method

RS-232C terminal	XMC basic unit		Connection number and signal direction	XMC basic unit		RS-232C terminal
	Pin number	Signal number		Pin number	Signal number	
	1	TX		1	TX	
	2	RX		2	RX	
	3	SG		3	SG	
	4	NC		4	NC	
	5	NC		5	NC	
	6	PE		6	PE	

RS-232C terminal	nection number		Connection number and signal direction	XMC basic unit		RS-232C terminal
	Pin number	Signal number		Pin number	Signal number	
	1	TRX+		1	TRX+	
	2	TRX−		2	TRX−	
	3	COM		3	COM	
	4	NC		4	NC	
	5	NC		5	NC	
	6	PE		6	PE	

4) HMI (PC), Dedicated Communication and RS-485 Communication

- Communication using HMI (PC) and RS-232C channel
- HMI (PC) operates as a client station, and Cnet I/F acts as a server station, when the module settings operate with RS-232C XGT server.
- RS-485 channel operates in P2P mode.
- GSL-TR4A (32 points of Smart I/O transistor output for Modbus) data transmission via RS-485 channel
- Reading data transmitted to GSL-TR4A in HMI (PC)



Type	Module settings		
	RS-232C	RS-485	Station number
PLC Cnet #1 station	XGT server	P2P	1

15.7 Communication parameter

15.7.1 Summary

Communication parameters can be divided into basic setting parameters and P2P setting parameters as shown below.

1) Default setting parameters

This is the part that sets the media information, hardware information and basic protocol information of serial communication of Standalone Motion Controller

<table>
<tr><th rowspan="2">Parameter</th><th rowspan="2">Sub-menu</th><th rowspan="2">Setting item</th><th rowspan="2">Setting range</th><th colspan="2">Possibility of settings</th><th rowspan="2">Notice</th></tr>
<tr><th>Client</th><th>Server</th></tr>
<tr><td rowspan="10">Default settings</td><td rowspan="4">Connection settings</td><td>Communication type</td><td>RS-232C
RS-485</td><td>Possible</td><td>Possible</td><td>-</td></tr>
<tr><td>Communication speed(bps)</td><td>1200 ~ 115,200</td><td>Possible</td><td>Possible</td><td>-</td></tr>
<tr><td>Termination resistance</td><td>Enabled/Disabled</td><td>Possible</td><td>Possible</td><td></td></tr>
<tr><td>Station number</td><td>XGT communication: 0~255
Modbus: 0~255</td><td>Possible</td><td>Possible</td><td>Station number is not significant when setting client</td></tr>
<tr><td rowspan="6">Operation mode</td><td>P2P used</td><td rowspan="5">One mode is selectable</td><td>Possible</td><td>-</td><td>-</td></tr>
<tr><td>XGT server</td><td>-</td><td>Possible</td><td>-</td></tr>
<tr><td>Modbus ASCII server</td><td>-</td><td>Possible</td><td>-</td></tr>
<tr><td>Modbus RTU server</td><td>-</td><td>Possible</td><td>-</td></tr>
<tr><td>Smart server</td><td>-</td><td>Possible</td><td></td></tr>
<tr><td>Repeater mode</td><td>Enabled/Disabled</td><td>-</td><td>-</td><td>All services are stopped when repeater mode is set</td></tr>
<tr><td rowspan="8">Advanced settings</td><td rowspan="5">Connection settings</td><td>Data bit</td><td>7, 8</td><td>Possible</td><td>Possible</td><td>In Modbus ASCII, the number of data bits is 7.</td></tr>
<tr><td>Stop bit</td><td>1, 2</td><td>Possible</td><td>Possible</td><td>*Note 5)</td></tr>
<tr><td>Parity bit</td><td>NONE,ODD,EVEN</td><td>Possible</td><td>Possible</td><td>-</td></tr>
<tr><td>Parity reception error *Note 4)</td><td>Allowed/ Not allowed</td><td>Possible</td><td>Possible</td><td>-</td></tr>
<tr><td>Modem type</td><td>Null modem</td><td>Possible</td><td>Possible</td><td>-</td></tr>
<tr><td rowspan="3">Time settings</td><td>Response waiting time</td><td>0~50 (x 100ms)</td><td>Possible</td><td>-</td><td>*Note 1)</td></tr>
<tr><td>Delay time</td><td>0~255 (x 10ms)</td><td>Possible</td><td>Possible</td><td>*Note 2)</td></tr>
<tr><td>Inter-character waiting time</td><td>0~255 (x 10ms)</td><td>Possible</td><td>Possible</td><td>*Note 3)</td></tr>
</table>

Notice

1) Response waiting time: It means the waiting time until the reception of response frame after the completion of frame transmission.
(1) Operation setting: It can e set when P2P is used in operation mode.
(2) Response waiting time = Basic response waiting time + (response waiting time set value x 100ms) + inter-character waiting time
(3) Basic response waiting time by communication speed
1) 9,600~115,200bps: 100ms
2) 7,200~2,400bps: 200ms
3) 1,800~1,200bps: 400ms

2) Delay time setting: It means transmitting a frame after the delay time set by a user.
(1) Client operation setting: Client operation can be set when communication type is RS-485.
(2) Server operation setting: It is possible to transmit a frame after the delay time for the response of the server to be set.

3) Inter-character waiting time: It means the spacing between characters in one frame processed as one frame in case of a character that comes in the set time in one frame.
(1) Operation setting: It can be set regardless of operation mode.

4) Parity reception error: When [Accept] is selected, data can be received even if there is an error in the reception parity bit.

5) Stop bit: It means the end of a single packet and checks the stop bit set at the time of data transmission. If the stop bit of the received data is smaller than the set stop bit, data cannot be received normally. In order to receive data normally, the stop bit should be configured identically.
(1) When stop bit is set to 1, all frames can be received.
(2) Normal communication may not be possible in communication with equipment where stop bit is set to 2.
(3) When stop bit is set to 2, data of stop bit 1 cannot be received.
(4) The parameters should be set the same for normal communication.

(1) P2P service
(a) This service allows the Cnet I/F module to act as a client on the network.
(b) It can read or write the memory of the other station when a predetermined event occurs.
(it can operate as XGT client or Modbus client.)
(c) This is used in communication with the equipment of other companies that does not support XGT or Modbus protocol or in transmission/reception of a frame desired by a user.
(d) It can define up to 64 independent P2P blocks per channel.

(2) Dedicated service(XGT server, Modbus ASCII server, Modbus RTU server)
(a) This service allows PC and peripheral devices to read and write information and data without the creation of a separate program in the PLC.
(b) It can operate as XGT server that support the XGT protocol and Modbus server that supports the Modbus RTU/ASCII protocol.

2) P2P setting parameters

This is to set communication frames.

Parameter	Sub-menu	Setting item	Setting range and contents	Possibility of settings(client)				
				XGT	Modbus ASCII	Modbus RTU	Inverter Dedicated communication	User frame definition
P2P	Communication module settings	Base	0~7	Possible	Possible	Possible	Possible	Possible
		Slot	0~11	Possible	Possible	Possible	Possible	Possible
	P2P channel	P2P deriver	User frame definition	-	-	-	-	Possible
			XGT client	Possible	-	-	-	-
			Modbus ASCII Client	-	Possible	-	-	-
			Modbus RTU Client	-	-	Possible	-	-
			LS Bus client*Note 5)	-	-	-	Possible	-
	P2P block	Channel	1, 2	Possible	Possible	Possible	Possible	Possible
		P2P function	READ	Possible	Possible	Possible	Possible	-
			WRITE	Possible	Possible	Possible	Possible	-
			SEND	-	-	-	-	Possible
			RECEIVE	-	-	-	-	Possible
		Start condition*Note 1)		Possible	Possible	Possible	Possible	Possible
		Method	Individual	Possible	Possible	Possible	-	-
			Continuous	Possible	Possible	Possible	Possible	-
		Data type	Bit	Possible	Possible	Possible	-	-
			Word	Possible	Possible	Possible	Possible	-
			1 bye	Possible	-	-	-	-
			2 bytes	Possible	-	-	-	-
			4 bytes	Possible	-	-	-	-
			8 bytes	Possible	-	-	-	-
		Number of variables*Note 2)		Possible	Possible	Possible	-	-
		Data size*Note 2)		Possible	Possible	Possible	Possible	-
		Partner station number		Possible	Possible	Possible	Possible	-
		Frame		-	-	-	-	Possible
		Settings*Note 3)		Possible	Possible	Possible	Possible	Possible
	User frame definition	Add group	Group name	-	-	-	-	
			Frame type Transmission	-	-	-	-	Possible
			Reception	-	-	-	-	Possible
	Frame*Note 4	Edit group	Group name	-	-	-	-	Possible
		Delete group		-	-	-	-	Possible
		Add frame	HEAD	-	-	-	-	Possible
			TAIL	-	-	-	-	Possible
			BODY	-	-	-	-	Possible

Notice

1) Start condition in user-defined frame communication is selectable only when the P2P function is SEND.
2) The number of variables and data size can be set only in the case of continuous mode in XGT client and Modbus ASCII/RTU client.
3) Settings in user-defined frame communication can be established only when the fixed size parameter or variable size parameter is selected.
4) Frame settings can be entered after the frame type and group name of the user frame definition are set.
5) LS Bus client is a function provided by the B type Cnet I/F module.

3) Transmission Standard

In order to use the Cnet I/F module, set the transmission standard like baud rate, data/stop bit. The basic setting item of Cnet I/F module should be same with transmission standard of system. The written basic setting values are saved in the XMC, those are kept regardless of power off until rewriting the values.

(1) Communication type

Check the Cnet I/F module and set the parameter about each channel accurately. If communication type set by parameter is different with channel type of mounted communication module, normal communication is impossible because CPU recognizes the channel type of mounted communication module.

(2) Parity bit

The Cnet I/F module defines three parity bit. Each parity bit has the following meaning like

Parity	Meaning	Reference
None	Not use parity bit	
Even	If the number of 1 is even in the one byte, it sends 0 at the parity bit.	
Odd	If the number of 1 is odd in the one byte, it sends 0 at the parity bit.	

(3) Operation mode

Each channel of Cnet I/F module operates as server or client, each channel operates independently.

Driver type	Meaning	Reference
P2P	Relevant port acts as client and it communicates by P2P parameter setting.	Refer to P2P setting
XGT server	Support the XGT dedicated communication and act as XGT server.	For dedicated service
Modbus ASCII server	Act as Modbus ASCII server	
Modus RTU server	Act as Modbus RTU server	
Smart server	After analyzes the protocol automatically, act as XGT/Modbus ASCII/Modbus RTU server	

If operation mode of Cnet channel is XGT server or Modbus, it supports the loader service with the dedicated service

a) XGT Server
Support reading/writing memory of the dedicated service

b) Modbus ASCII/RTU Server
1) This is used when network is configured with modbus protocol and Cnet I/F module acts as server.
2) Since modbus memory area is different with XGT memory area, memory mapping is necessary.

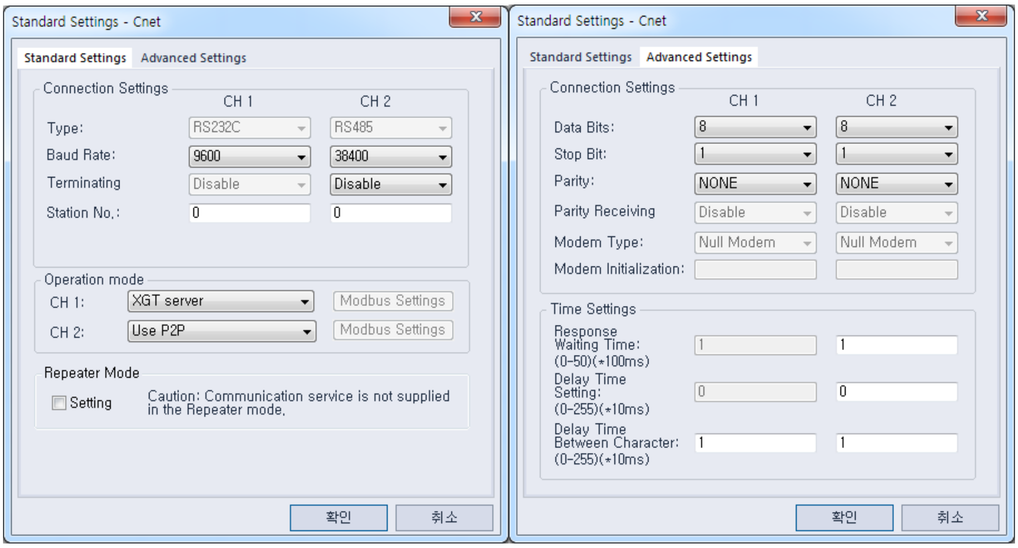
15.7.2 Downloading Parameters

To operate XMC-Cnet according to user-defined communication specification and mode, you should follow procedures below.

1) Communication setting contents

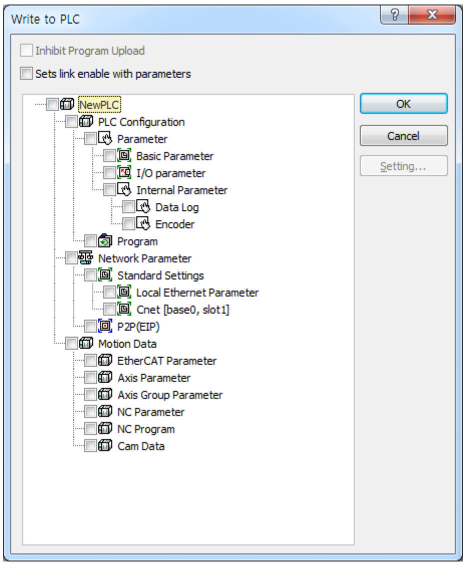
- Channel 1:RS-232C, 9600Bps, 8/1/NONE, XGT server, station 1
- Channel2: RS-485, 38400Bps, 8/1/ODD, P2P, station 2, response waiting time100ms
  Inter-character waiting time 0ms, XGT server

2) After Cnet module is registered, double click the Cnet module and the following default setting window appears.



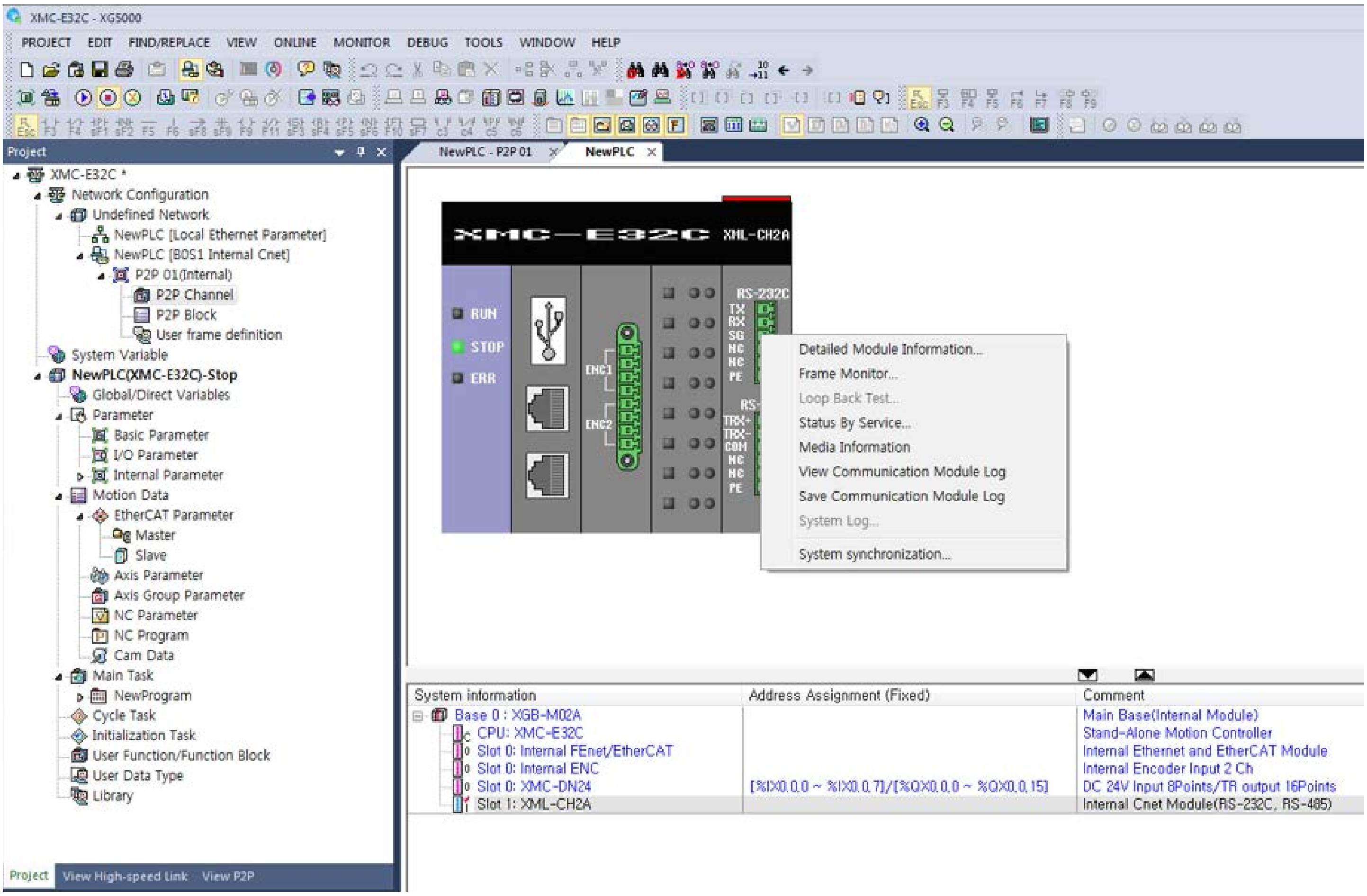
3) Download the parameter

Select [Online → Connection → Write] menu and [Confirm the Parameter Download Window] to perform download. After the download is complete, the parameters are applied immediately after the download is complete. At this time, if [Link Enable Setting Together] is checked, Link Enable is also set.



4) Check the operation

The status check and diagnostic method of the system and network through system diagnostics of XG5000 are as follows. Connect XG5000 to the basic unit and select “Online → Communication Module Setting → System Diagnostics” in the menu, or click the system diagnostics icon ( ) and the following window appears.



15.7.3 Server Function and P2P Service

Server function is a built-in function in the built-in Cnet and enables PC and peripheral devices to read and write information and data in PLC without creating separate program in PLC.
It operates as a server in the communication network and responds when memory read and write requests that follow the XGT dedicated protocol or Modbus protocol are received from the external device or PC.

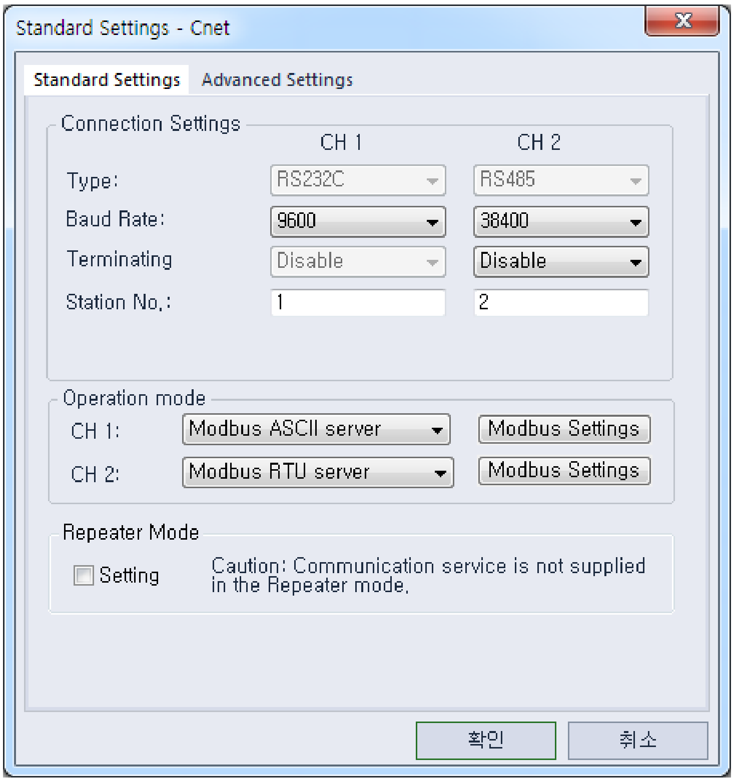
1) Server function

(1) XGT dedicated server

It is used in inter-company communication as its company-dedicated protocol service, and the characters used in all frames are composed of ASCII codes. When used in multi-drop mode, up to 32 stations can be connected for use. Make sure not to set duplicate station number on the same network, and the communication speed/stop bit/parity bit/data bit of all Cnet I/F modules on the network should be the same when used as multi-drop. Please refer to “15.6 XGT dedicated protocol“ for details on the protocol.

(2) Modbus server

It is used when the target device for communication operates as a Modbus client.
It supports both RTU mode and ASCII mode of Modbus and can be defined in the default setting window operation mode.

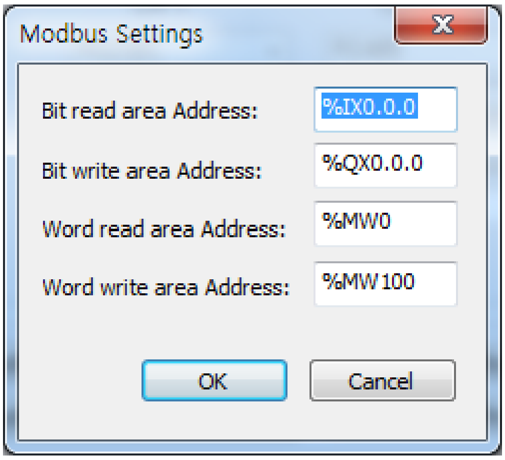


The maximum number of Modbus commands and response data supported by the Modbus RTU/ASCII driver is shown in the following table. The other client device should make requests only within the scope of the table below.

Code	Use	Address	Maximum number of response data
01	Read Coil Status	0XXXX	2000 Coils
02	Read Input Status	1XXXX	2000 Coils
03	Read Holding Registers	4XXXX	125 Registers
04	Read Input Registers	3XXXX	125 Registers
05	Force Single Coil	0XXXX	1 Coil
06	Preset Single Register	4XXXX	1 Register
15	Force Multiple Coils	0XXXX	1968 Coils
16	Preset Multiple Registers	4XXXX	120 Registers

[Modbus command code]

The area corresponding to the command code in the above table must be set in the Stand-alone Motion controller serial communication module memory. The [Modbus Settings] button becomes active when the Modbus ASCII server or Modbus RTU server is selected as the operation mode in the [Preferences] window. If you click on it, the [Modbus Settings] window will appear as shown below, and you can set the start address here.



[Modbus server memory setting]

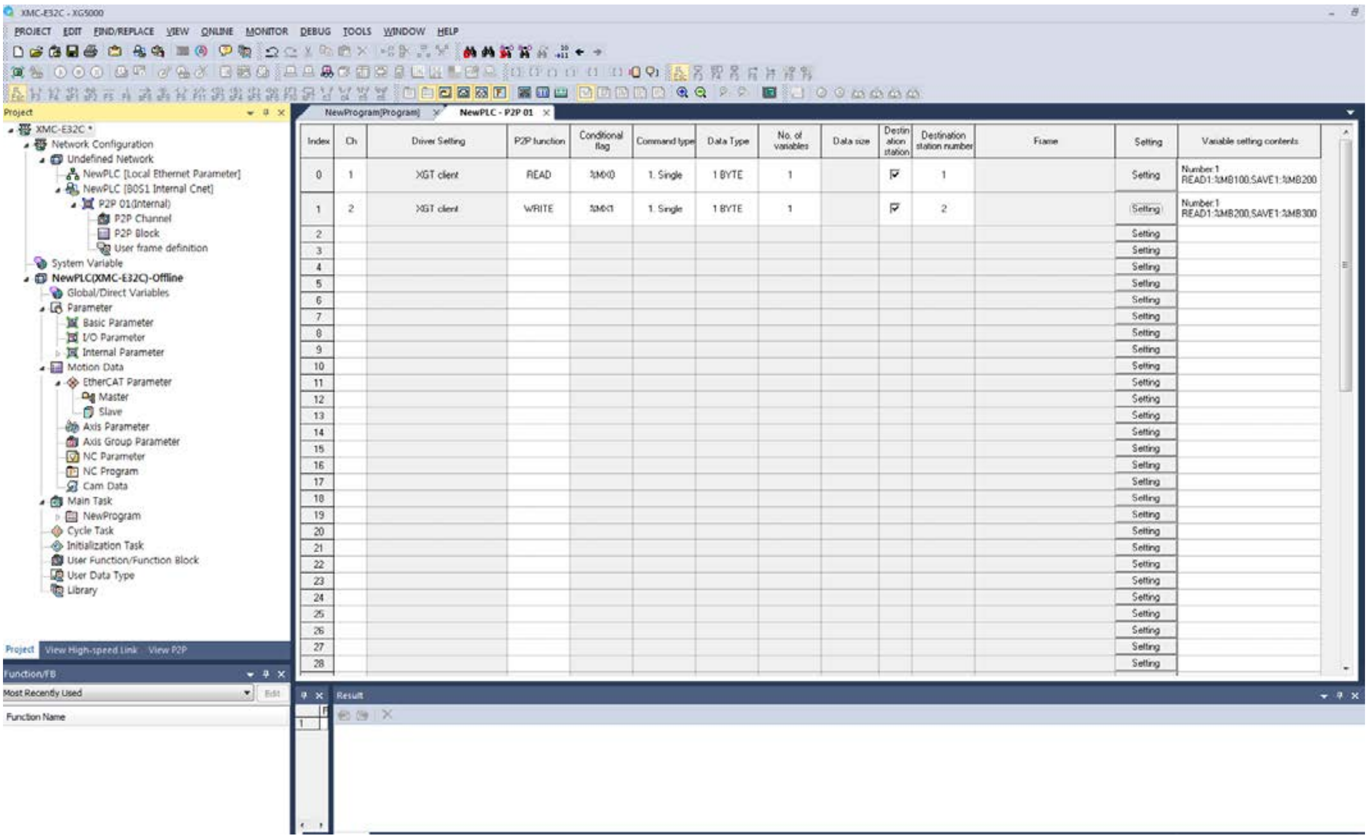
Details of respective setting item are as follows

Item	Description	Remarks
Bit read area address	XGT address applicable to Bit input area	Bit address
Bit write area address	XGT address applicable to Bit output area	Bit address
Word read area address	XGT address applicable to Word input area	Word address
Word write area address	XGT address applicable to Word output area	Word address

[Details of Modbus Area]

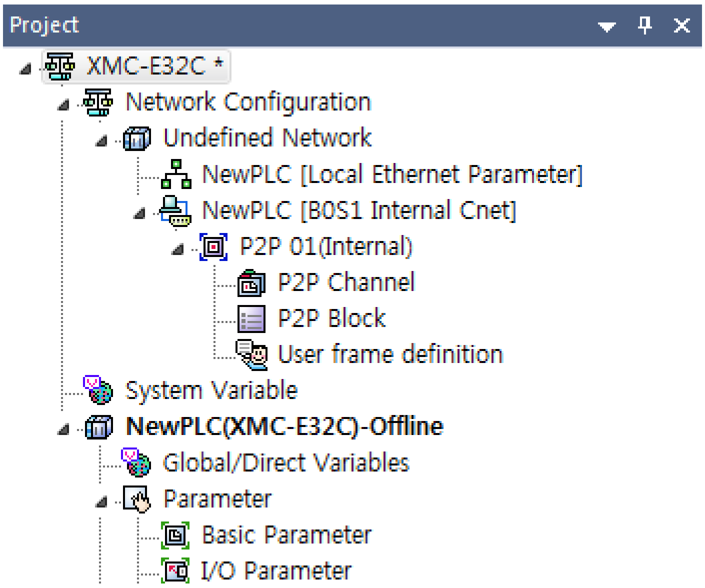
1) P2P Service

P2P service performs the client operation of the communication module, and there are four commands available in Cnet: Read/Write/Send/Receive. The registration and editing of P2P service are done in XG5000, and each P2P parameters consist of up to 64 P2P blocks. The following figure shows an example of the P2P parameter setting window of XG5000.



(1) P2P parameter configuration

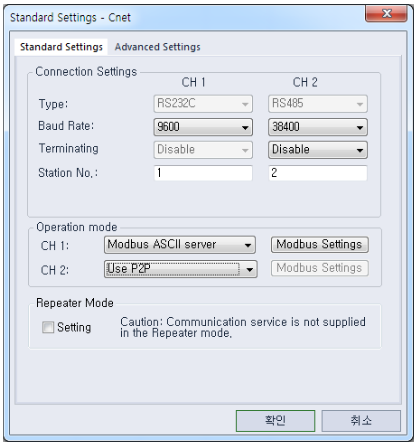
In order to use P2P service, a user should perform settings for desired operation in P2P parameter window. The P2P parameter consists of three pieces of information as shown below.

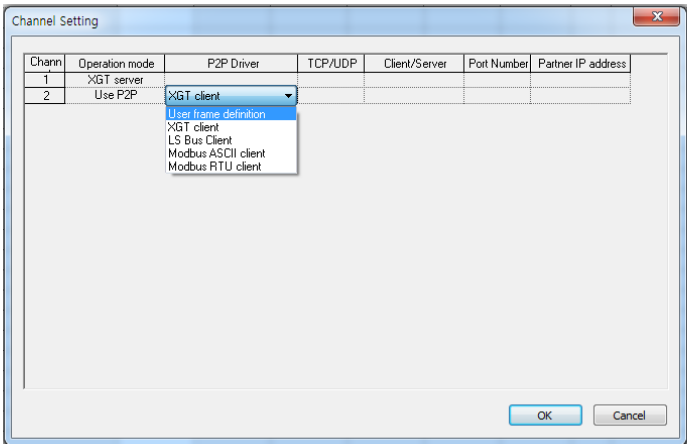


Division	Contents	Remarks
P2P channel	- Set P2P channel to define the communication protocol of P2P service to perform - XGT/Modbus available - It is possible to set independently for each channel. It is applied only when the operation mode of the default settings is “Use P2P”.	
P2P block	Set 64 P2P blocks that operate independently	
User-defined frame registration	Set frame for user-defined communication	

(2) Channel settings

Built-in Cnet I/F function is fixed to P2P No. 1 and provides two fixed communication channels. In the built-in Cnet I/F, dry type for P2P service can be defined respectively.





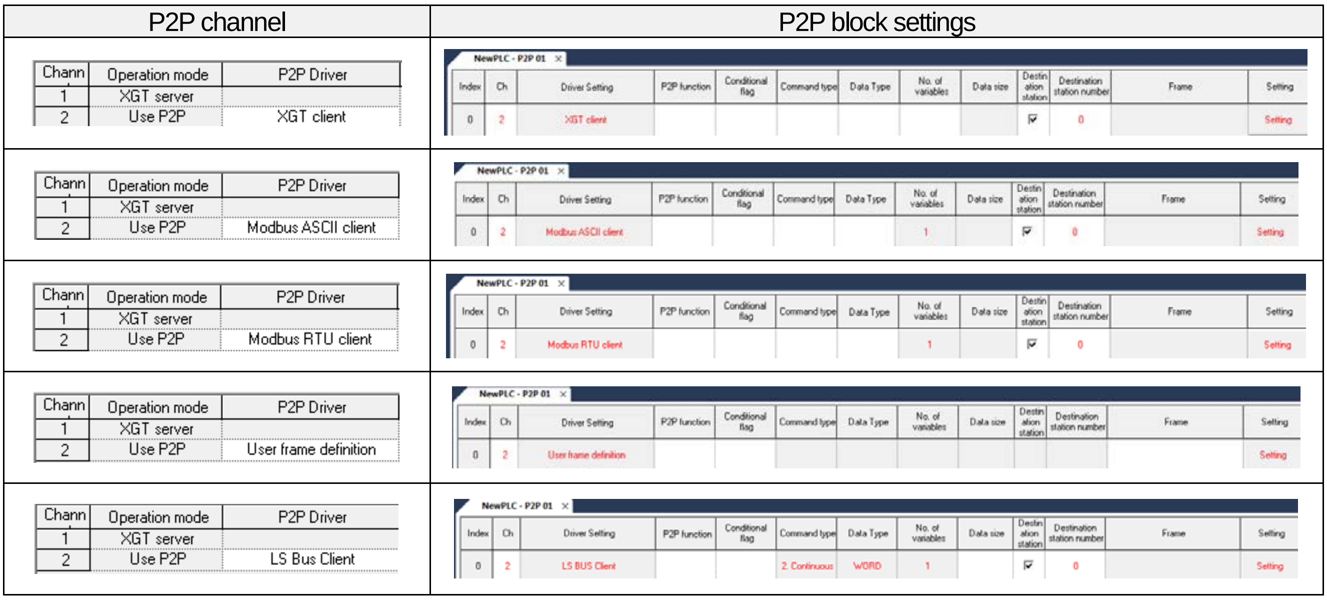
The following drivers can be selected when 'Enable P2P' is selected in the operation mode.

Driver	Meaning
None	P2P service is not used
User frame definition	Used when it sends/receives the desired user-defined frame
XGT client	Selected when it performs memory read/write of XGT
LS Bus client	Selected when it communicates with the inverter of its company
Modbus ASCII client	Selected when it operates as a Modbus client and is used in ASCII mode
Modbus RTU client	Selected when it operates as a Modbus client and is used in RTU mode

User frame definition communication is possible only when P2P driver is set to user frame with respect to the communication channel.

(3) P2P block setting

If you select the P2P block of the corresponding parameter in the [P2P Parameter Setting] window, the [P2P Block Setting] window appears. The block setting window of all protocols is as shown in the figure below. Depending on the protocol selected in the P2P channel, the active area is displayed differently.



P2P block settings

.

15.7.4 Start operation

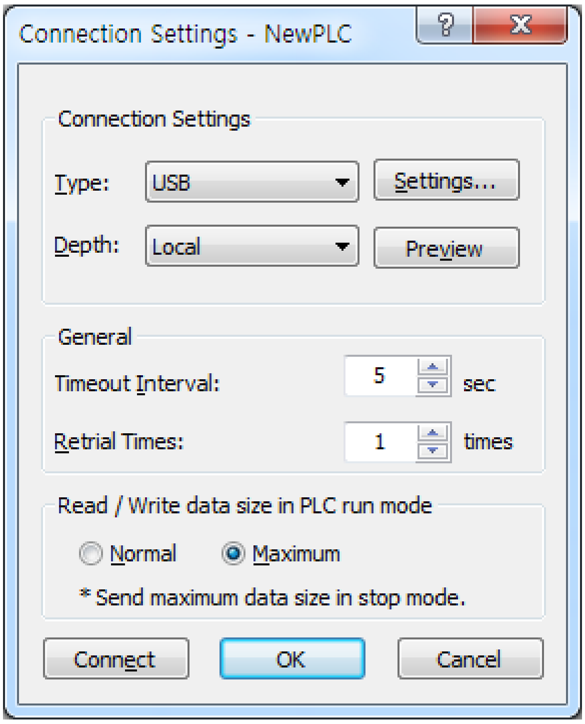
The operation mode of the serial communication of stand-alone motion controller can be roughly divided into P2P service and server function. To use each mode, follow the steps below.

1) When operating as a server

(1) Connection setup

(a) Select [Online] → [Connection Settings] or click the icon ().

(b) Set connection options for your environment and click [Connect].



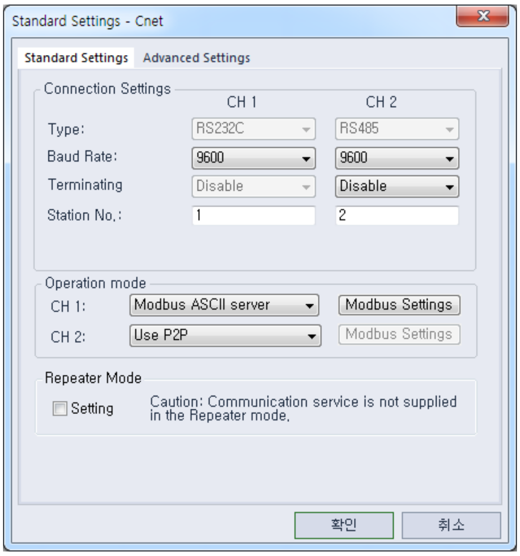
(2) Default setting

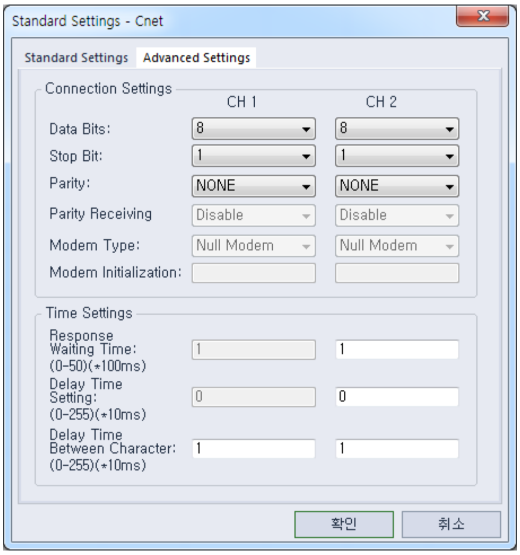
(a) Double-click the serial communication module in the project window to launch the [Preferences] window and set the connection. Set the communication type, communication speed, data bit, stop bit, and station number in the menu.

(b) The delay time can be set only when RS-485 is used, and the response wait time can be set only when using P2P as the operation mode in RS-485 communication.

(c) Cnet's repeater mode does not support Auto Speed.
When communicating via RS-485 channel, it is necessary to connect a terminating resistor from the outside.

- When used as a Modbus ASCII server, the data bit is 7





(3) Select operating mode

(a) Select the operation mode of the server to be used.

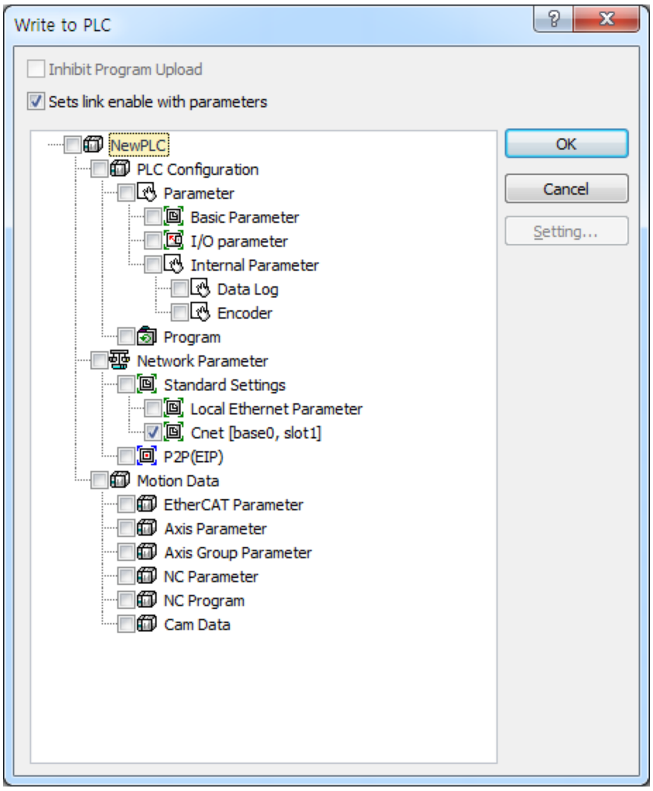
(b) Built-in Cnet supports XGT server, Modbus ASCII server, Modbus RTU server.

(4) Write parameters

(a) Select [Online] → [Write] or click the icon ( ).

(b) Check (✓) the module with the default settings and click [OK].

(c) Click the [OK] button, and when the parameter writing finishes, reset each module.

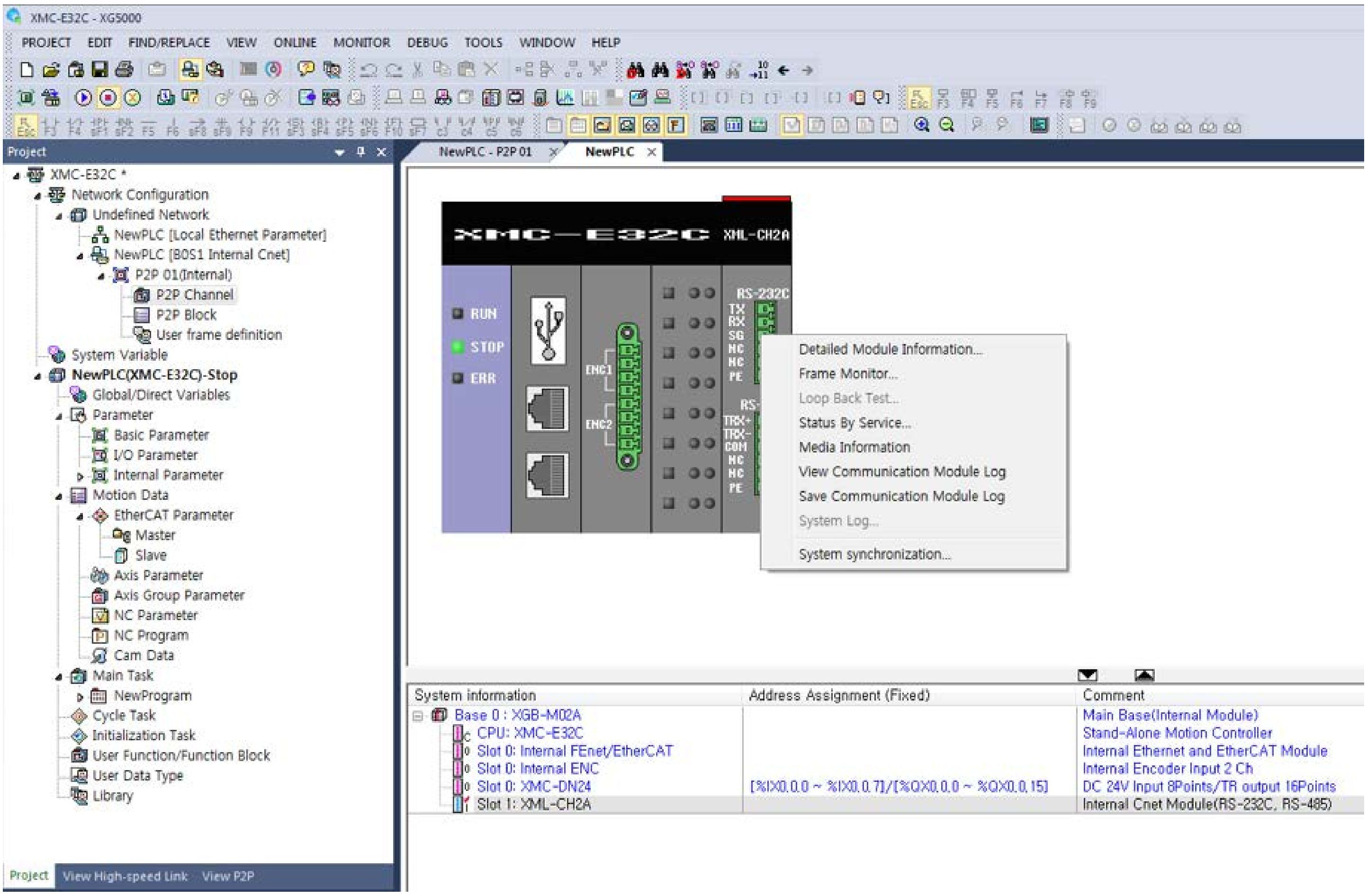


(5) Check operation

(a) Select [Online] → [Communication Module Settings] → [System Diagnosis] or click the icon ( ).

(b) Click the communication module whose status you want to diagnose and press the right mouse button.

(c) When the following screen appears, click [Frame Monitor] or [Service Status] to check the operation status.

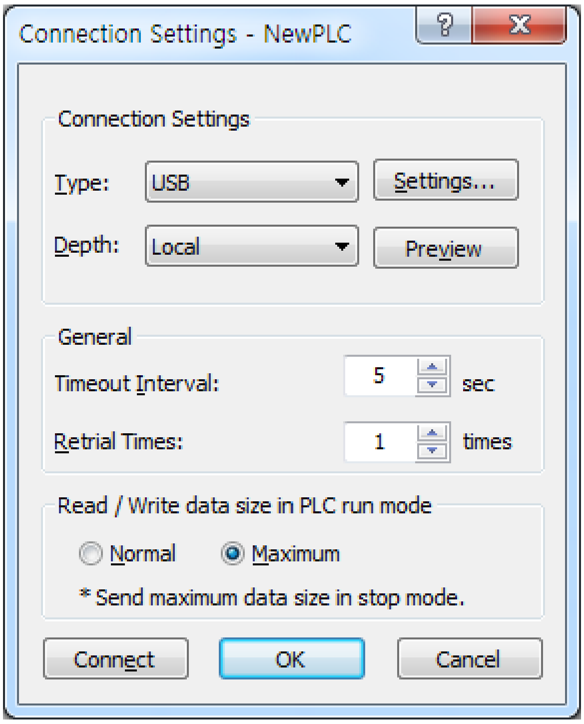


2) When operating as a P2P service (client)

(1) Connection setup

(a) Select [Online] → [Connection Settings] or click the icon ( ).

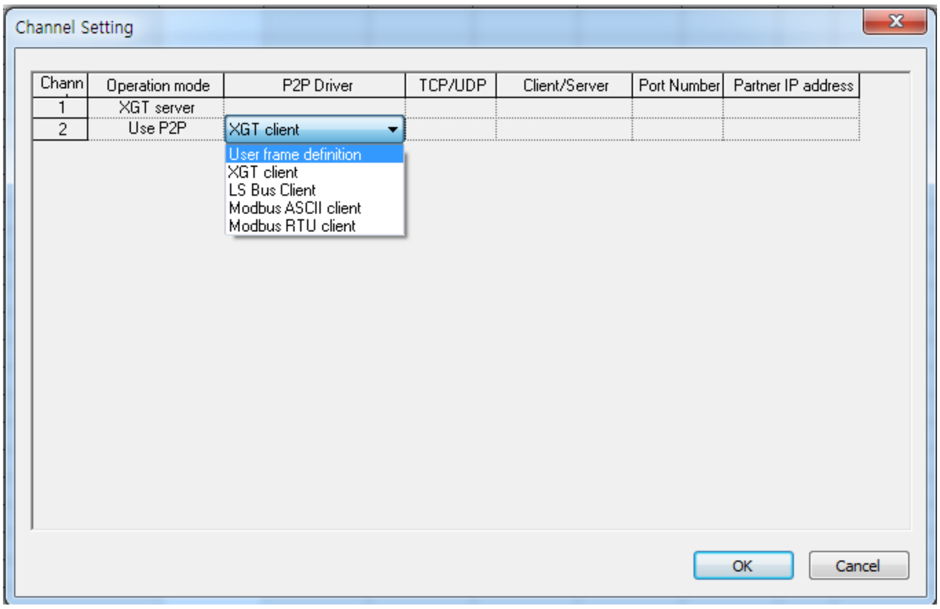
(b) Set connection options for your environment and click [Connect].



(2) Channel setting

(a) In the [P2P Parameter Setting] window, double-click the P2P channel to select the protocol for each channel.

(b) The P2P driver supports user frame definition, XGT client, LS bus client, Modbus RTU client, and Modbus ASCII client.

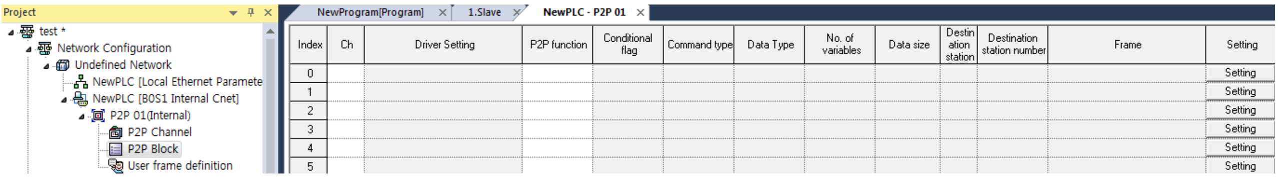


(3) P2P block setting

(a) Depending on the type of client selected in the channel setting, the P2P block setting value will be activated differently.

(b) Creates the contents of the cell that is in the active state for the protocol type.

* In the case of user frame definition, the frame must be created in the user frame definition before it can be used.

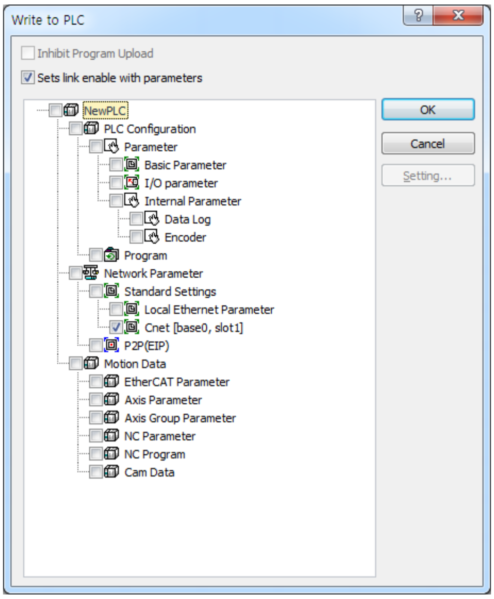


(4) Write parameters

(a) Select [Online] → [Write] or click the icon ( ).

(b) Check (✓) the module with the default settings and click [OK].

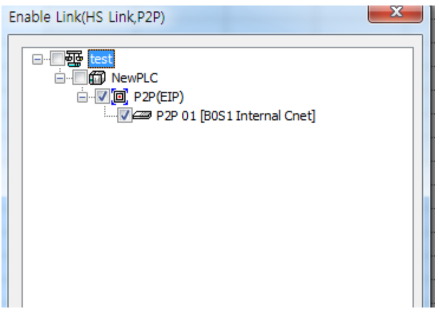
(c) Click the [OK] button, and when the parameter writing finishes, reset each module.



(5) Link enable

(a) Select [Online] → [Communication module settings] → [Link Enable] or click the icon ( )

(b) Check (✓) the configured P2P and click [Write].

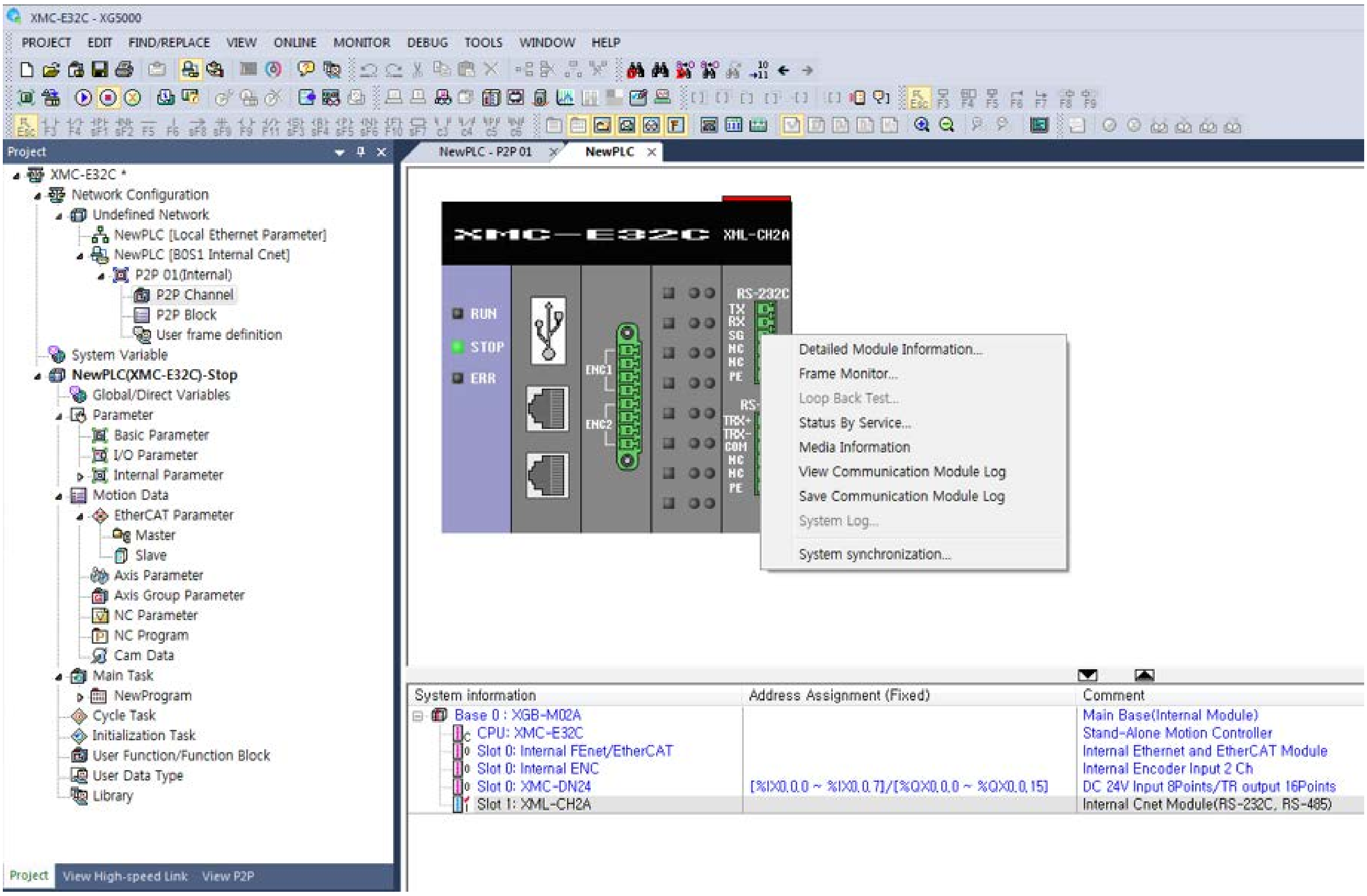


(6) Check operation

(a) Select [Online] → [Communication Module Settings] → [System Diagnosis] or click the icon ( ).

(b) Click the communication module whose status you want to diagnose and press the right mouse button.

(c) When the following screen appears, click [Frame Monitor] or [Service Status] to check the operation status.



15.7.5 Diagnostic Function of XG5000

1) Types of diagnostic functions

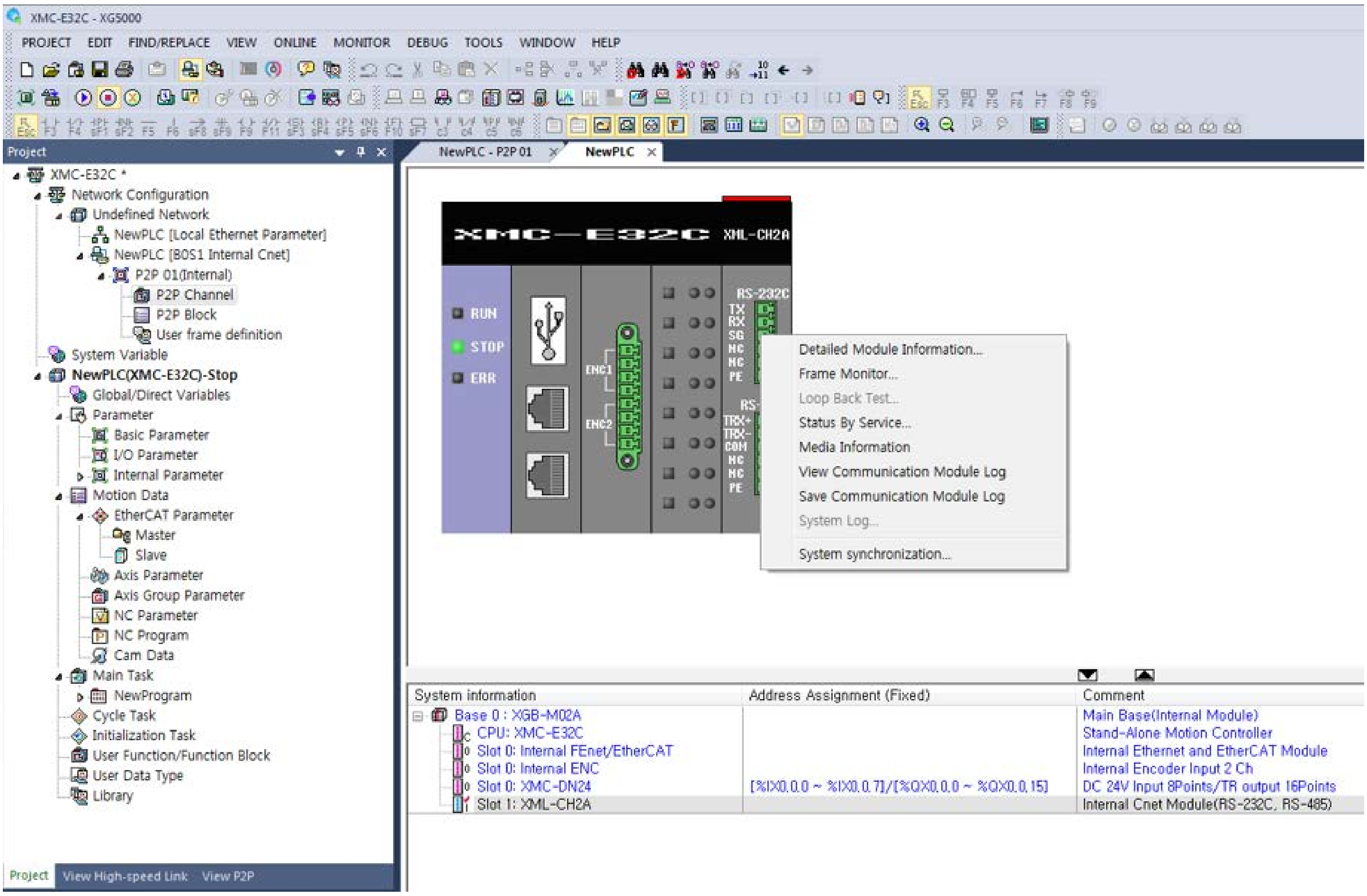
With XG5000, you can check/diagnose the status of network and various systems such as basic unit status, communication module information, service status information and frame monitor, etc.
The available diagnostic functions are as follows.

- ▶ XMC base unit status
- ▶ Communication module information
- ▶ Frame monitor
- ▶ Service status
- ▶ Media Information

(a) Select [Online] → [Communication Module Settings] → [System Diagnosis] or click the icon ().

(b) Click the communication module whose status you want to diagnose and press the right mouse button.

(c) When the following screen appears, click [Frame Monitor] or [Service Status] to check the operation status.



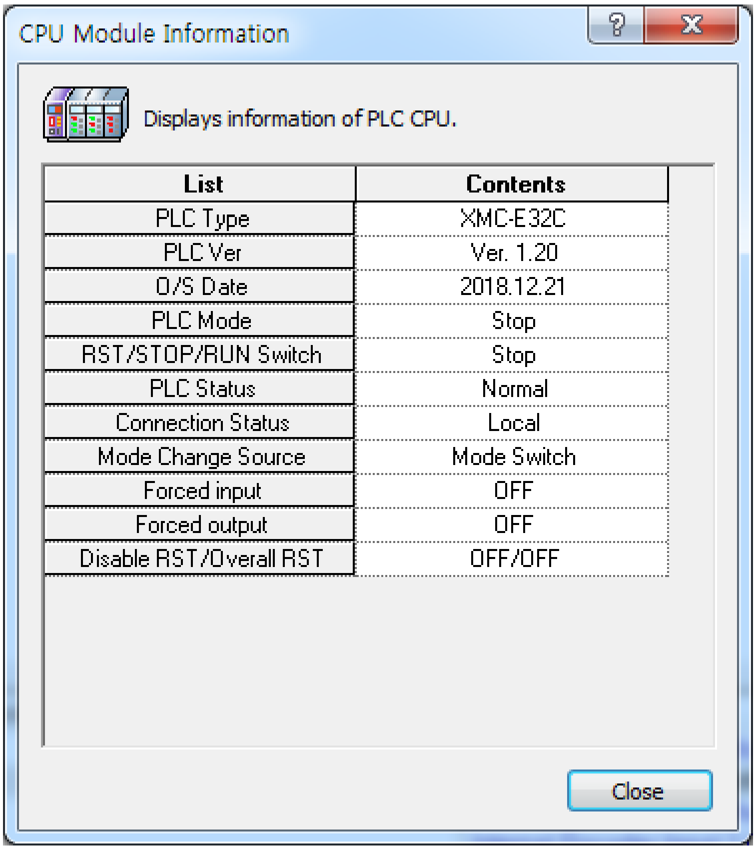
2) Check the XMC base unit status

(1) XMC unit information

(a) Select [Online] → [Communication Module Settings] → [System Diagnosis] or click the icon ( ).

(b) Click the XMC base unit and then right-click.

(c) If you click [CPU Module Information], the screen to check the status of the CPU module is displayed as shown below.

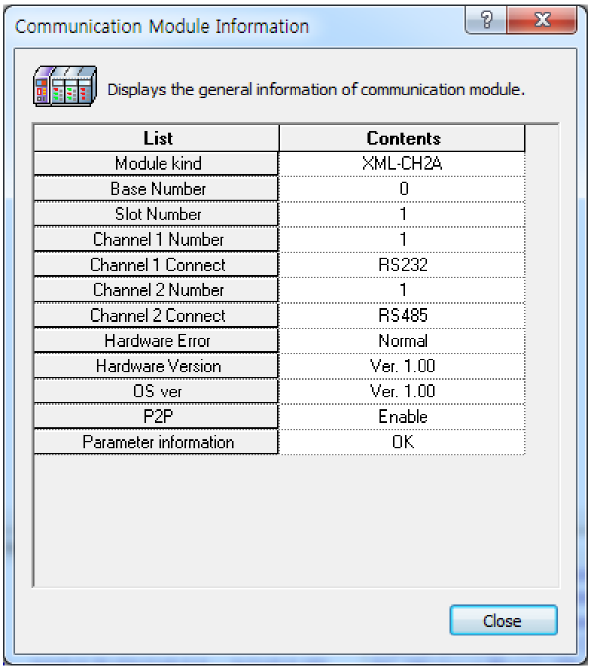


3) Communication module Information.

(a) Select [Online] → [Communication Module Settings] → [System Diagnosis] or click the icon ( ).

(b) Click the Cnet I / F module and click the right mouse button.

(c) If you click [Communication module information], the screen to check the status of communication module is displayed as below.



Items	**Contents**
Communication module type	Indicate the type of the communication module currently being diagnosed.
Base No.	Indicate the base information of the communication module currently being diagnosed. It is fixed to 0 and displayed in XGB PLC.
Slot No.	Indicate the slot number of the communication module currently being diagnosed. It is fixed to 0 and displayed in built-in communication
Station No.	Station number of the channel used in P2P and dedicated service
Connection method	Information of the communication type (RS-232C, RS-485) of the corresponding channel
Hardware error status	Indicate whether the hardware of the communication module is normal or not.
Hardware version	Version of communication module hardware
OS version	Indicate the version of the communication module OS
P2P	Indicate the P2P communication is enabled/disabled
System parameter information	Whether to download the default communication parameters Display the error information of default communication parameters

4) Frame monitor

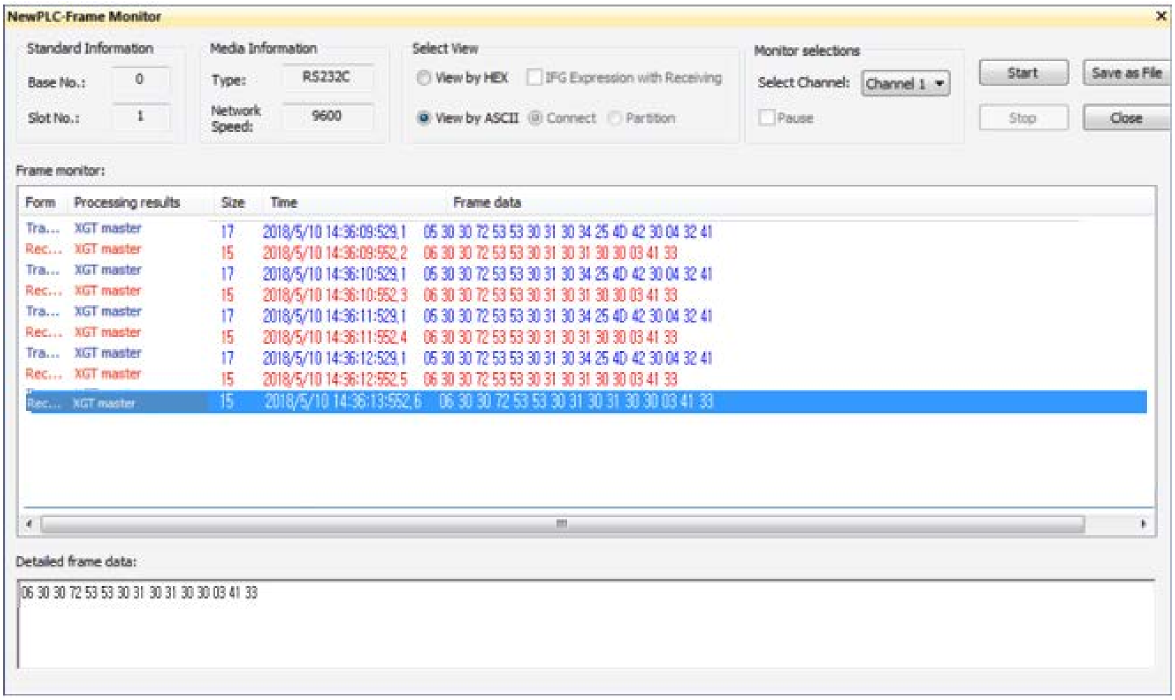
The frame monitor of XG5000 allows you to check whether the frame transmitted/received through the Cnet I/F module is normal or not.

(1) Frame monitor

(a) Select [Online] → [Communication Module Settings] → [System Diagnosis] or click the icon ( ).

(b) Click the Cnet I / F module and click the right mouse button.

(c) If you click [Frame Monitor], the screen to monitor the communication status will appear as shown below.



(2) Frame monitor details

Items	Contents

Basic information	Base No.	Base position of the communication module being monitored
	Slot No.	Slot position of the communication module being monitored
Monitor option	Channel selection	Select the channel to monitor
Frame monitor window	Type	Indicate the transmission frame and reception frame
	Processed results	Indicate the protocol type currently being used 1) XGT server 2) XGT client 3) Modbus server 4) Modbus client 5) User defined 6) Unknown: Frame that cannot be processed
	Size	Length of the monitored frame
	Time	Display the point of time for transmission/reception
	Frame data	Display the data of transmitted/received frame
View in HEX		Display the frame data with HEX values
View in ASCII		Display the frame data with ASCII values
Save file		Save the frame monitoring contents to a file
Start		Start of the frame monitoring operation
Stop		Stop the monitoring status
Close		Close the frame monitor window

5) Service status

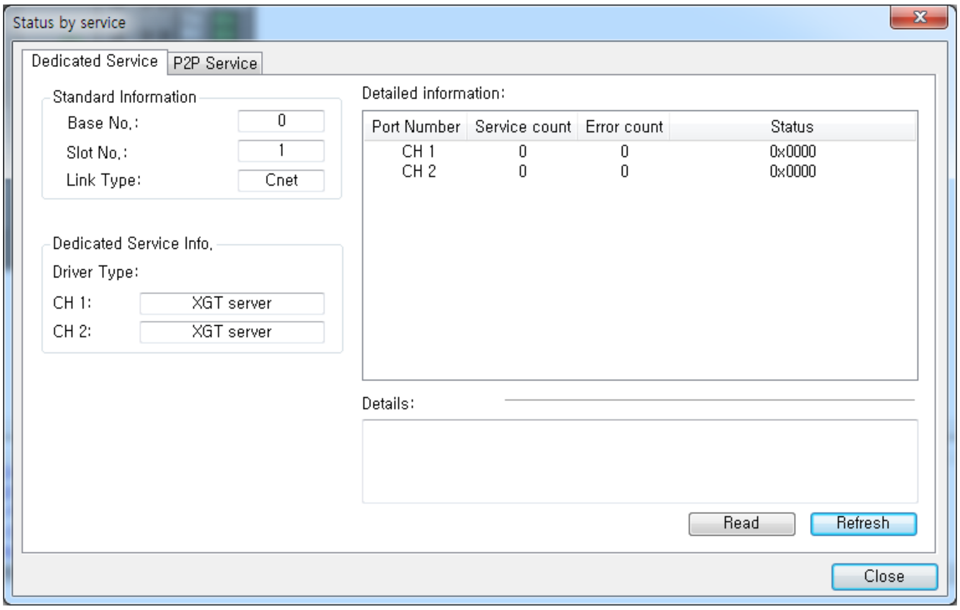
(1) Dedicated service

(a) Select [Online] → [Communication Module Settings] → [System Diagnosis] or click the icon ( ).

(b) Click the built-in Cnet and then right-click.

(c) Click [Service Status] and select [Dedicated Service] in the following screen.

(d) Click [Continue Reading] to check the status of each service.



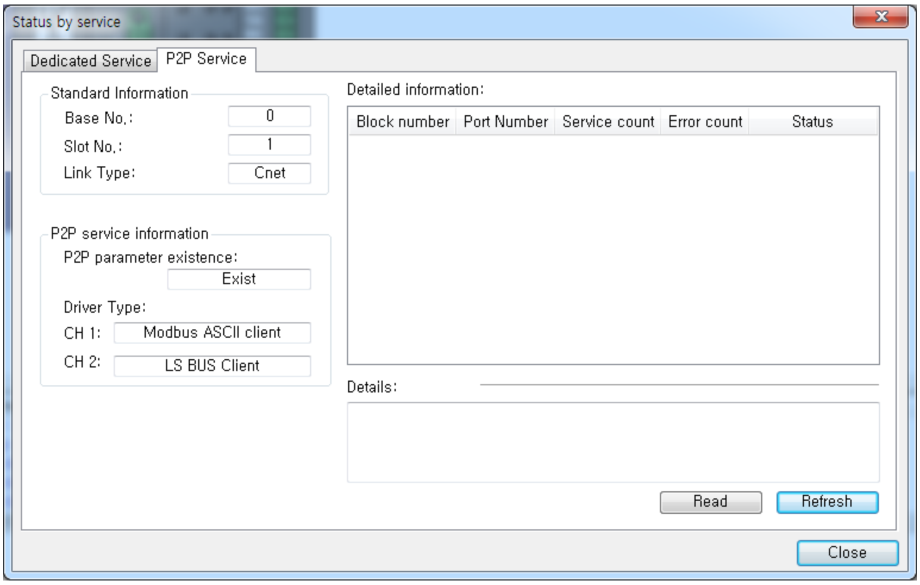
(2) P2P service

(a) Select [Online] → [Communication Module Settings] → [System Diagnosis] or click the icon ( ).

(b) Click the built-in Cnet and then right-click.

(c) Click [Service Status] and select [P2P Service] in the following screen.

(d) Click [Continue Reading] to check the status of each service.



(3) Service-specific details

Division	Item	Contents

<table>
<tr><td rowspan="2">Continuous Read/ Redo</td><td colspan="2">Continuous Read</td><td>Check the dedicated service status information every second</td></tr>
<tr><td colspan="2">Redo</td><td>Check the dedicated service status information at the time of execution</td></tr>
<tr><td rowspan="8">Dedicated service</td><td rowspan="3">Basic information</td><td>Base No.</td><td>Base position of the module using the dedicated service</td></tr>
<tr><td>Slot No.</td><td>Slot position of the module using the dedicated service</td></tr>
<tr><td>Link type</td><td>Type of the communication module being used</td></tr>
<tr><td colspan="2">Dedicated service information</td><td>Indicate the type of drives being used for each channel</td></tr>
<tr><td rowspan="4">Detailed information window</td><td>Port No.</td><td>Indicate the channel number</td></tr>
<tr><td>Service count</td><td>Indicate the number of dedicated service communications</td></tr>
<tr><td>Error count</td><td>Indicate the number of errors that occur during the dedicated service communication</td></tr>
<tr><td>Status</td><td>Display the dedicated service communication status</td></tr>
<tr><td rowspan="10">P2P service</td><td rowspan="3">Basic information</td><td>Base No.</td><td>Base position of the module using the P2P service</td></tr>
<tr><td>Slot No.</td><td>Slot position of the module using the P2P service</td></tr>
<tr><td>Link type</td><td>Type of the communication module being used</td></tr>
<tr><td rowspan="2">P2P service information</td><td>Presence of P2P parameters</td><td>Indicate whether the P2P parameter is downloaded or not</td></tr>
<tr><td>Driver type</td><td>P2P driver setting information for each channel
XGT client/MODBUS client/User definable</td></tr>
<tr><td rowspan="5">Detailed information</td><td>Block No.</td><td>Available from 0 to 63
Display only the currently registered block being operated</td></tr>
<tr><td>Port No.</td><td>Indicate the channel number</td></tr>
<tr><td>Status</td><td>Display service execution status information for each block</td></tr>
<tr><td>Service count</td><td>Indicate the number of times each block has been executed since the P2P service was performed</td></tr>
<tr><td>Error count</td><td>Indicate the number of errors that occur during the service</td></tr>
<tr><td rowspan="2">Continuous Read/ Redo</td><td colspan="2">Continuous Read</td><td>Check the P2P service status information every second</td></tr>
<tr><td colspan="2">Restart</td><td>Check the P2P service status information at the time of execution</td></tr>
</table>

(4) Service status code

The status codes provided by the Service Status are used to determine whether Cnet I/F performs normal communication or not.

Dedicated service		P2P service	
Status	Meaning	Status	Meaning
0	Normal communication	0	Normal communication
1	Reception frame header error (No ACK/NAK)	4	Maximum station setting error(when setting 0 to 255 stations or more)
2	Reception frame tail error(No Tail)	5	Time out occurs
3	Reception frame BCC error	FFFE	1.Modbus address error 2.When using commands other than Read/Write
9	The station number of the received frame is different from the number of its station (Its station number=0)	-	
0A	No response is received from the CPU		
0B	The received frame is longer than the Modbus maximum frame		
0C	The received frame is not Modbus ASCII / Modbus RTU		
0D	HEX conversion error in Modbus occurs		

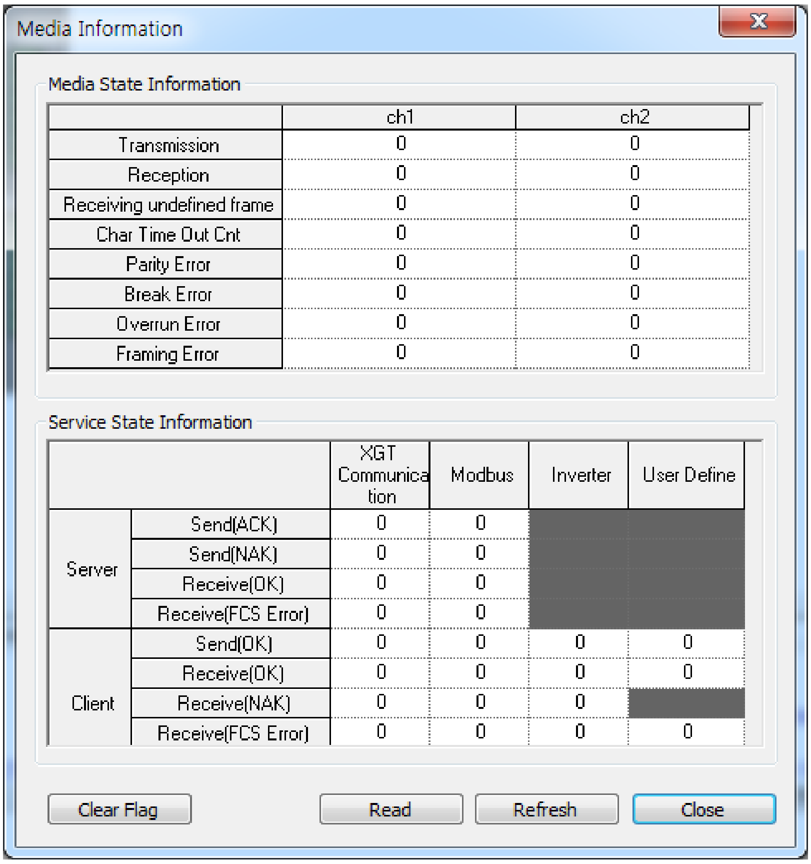
6) Media Information

It provides statistical value of media status and service status of built-in Cnet and is used to determine whether communication is normal or not.

(a) Select [Online] → [Communication Module Settings] → [System Diagnosis] or click the icon ( ).

(b) Click the built-in Cnet and then right-click.

(c) If you click [Media Info], the following screen appears.



15.8 XGT Communication

15.8.1 XGT Protocol Overview

The XGT protocol is a Cnet I/F module dedicated protocol developed by LS Industrial Systems.

In addition, it can check the communication status by monitoring the actual protocol when communicating with the other device through the frame monitoring of XG5000. The XGT protocol is divided into an XGT client that requests the other device to read and write data and an XGT server that responds to the contents requested from the XGT client.

(1) Data Read/Write

The monitoring function and device area read/write function inside the CPU module of the XGT client/server can be used to easily construct a communication system intended by a user.

(2) Frame monitor

The frame monitoring function of XG5000 that sets parameters of Cnet I/F module can actually check the frame of XGT client and server when communicating with the other station, and it analyzes the data through frame monitoring, check the error code when the error occurs and can solve the problems that occur during communication.

(3) XGT client, server

In communication using the XGT protocol, the XGT client requests the other device to read or write data, and the XGT server analyzes the data received from the XGT client and processes the requested command along with the ACK response when receiving a frame that conforms to the XGT protocol specification but transmits the NAK response including the error code to the XGT client.

(4) Functions provided by Cnet when using XGT protocol

(a) Operate the independent channel of RS-232C and RS-485
(b) Device Individual/Continuous Write
(c) Device Individual/Continuous Read
(d) Monitor variable registration
(e) Monitor execution
(f) 1:1 connection (its own link) system configuration

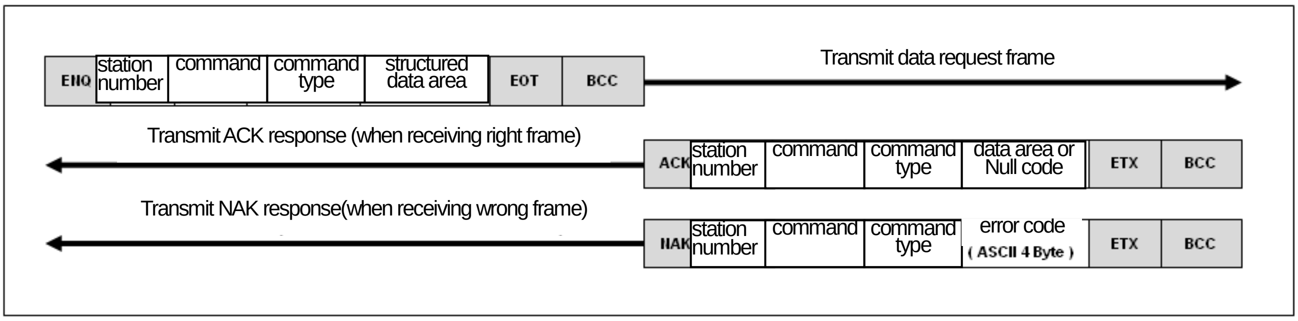
15.8.2 P2P service

The frame of the XGT protocol is largely divided into a frame in which a communication device that operates as an XGT client makes a request to read/write data and a frame that responds to the request frame of the client.

1) Frame structure

(1) Command frame sequence

When the request frame that is standardized from the XGT client is transmitted to the server, the server analyzes the received request server frame and transmits the ACK response when receiving the frame that conforms to the predetermined protocol rule. Otherwise, it transits the NAK frame to which the error code is attached.



(2) Basic frame structure of the XGT protocol

(a) Request frame (Equipment operating as the XGT client)

Header (ENQ)	Partner station number	Command	Command type	Structured data area	Tail(EOT)	Frame check (BCC)

(b) Response frame (Equipment operating as the XGT server)

1) ACK response frame(Received frame that meet the rules of the XGT protocol)

Header (ACK)	Its station number	Command	Command type	Structured data area or null code	Tail(ETX)	Frame check (BCC)

2) NAK response frame (Received frame that does not meet the rules of the XGT protocol)

Header (NAK)	Its station number	Command	Command type	Error code (ASCII type 4 byte)	Tail(ETX)	Frame check (BCC)

(3) Frame characteristics

(a) The numeric data of all frames is displayed in ASCII code for hexadecimal values unless otherwise specified.

(b) The items displayed in hexadecimal are as follows.

a) Station number

b) Command type in case command type is numeric (data type) when main command is R(r) or W(w)

c) All items that display all data sizes in the structured data area
d) Command registration number with respect to monitor registration and execution command
e) All contents of data

(c) In the case of hexadecimal data, ‘h’ is attached to the number as in h01, h12345, h34, h12 and h89AB, etc. to indicate that this data is hexadecimal.
(d) The maximum length of available frame is 256 bytes.
(e) The contents of control codes used are shown below.

Code	Hex	Name	Control contents
ENQ	05	Enquire	Start code of request frame
ACK	06	Acknowledge	Start code of ACK response frame
NAK	15	Not Acknowledge	Start code of NAK response frame
EOT	04	End of Text	ASCII code for ending request frame
ETX	03	End Text	ASCII code for ending response frame

(f) If the command is composed of lowercase letters, the BCC value is added to the frame check. If it consists of capital letters, the BCC value is not added.
Example)Device read command R(r)
- Lowercase letter r: BCC is added
- Capital letter R: BCC is not added

2) XGT Communication Commands

(1) Types of commands
The types of commands used in the dedicated communication are summarized below.

	Command	Processing contents

Item \ Division		Main command ASCII	Main command Hex	Command type ASCII	Command type Hex	
Read device	Individual read	r(R)	h72(h52)	SS	h5353	Read direct variables of bit and word type
	Continuous read	r(R)	h72(h52)	SB	h5342	Read direct variables of word type in block unit*Note 1)
Write device	Individual write	w(W)	h77(h57)	SS	h5353	Write data to direct variables of bit and word type
	Continuo write	w(W)	h77(h57)	SB	h5342	Write block units to direct variables of word type*Note 2)

Item \ Division	Command				Processing contents
	Main command		Registration No.		
	Frame example	Hex	Registration No.	Hex	
Monitor variable registration	x(X)	h78(h58)	00 ~ 09	h3030 ~ 3039	Register variables to monitor
Monitor execution	y(Y)	h79(h59)	00 ~ 09	h3030 ~ 3039	Execute monitor of registered variables

(2) Data type

Data type	Display frame example	Usage example
Bit	X(58h)	%MX000, %LX000, %KX000, %FX000, %IX0.0.0, %QX0.0.0, %UX0.0.0 , etc.
Byte	B(42h)	%MB000, %LB000, %KB000, %FB000, %IB0.0.0, %QB0.0.0, etc.
Word	W(57h)	%MW000, %LW000, %KW000, %FW000, %IW0.0.0, %QW0.0.0, %MW0, %RW0, %UW0.0, etc.
Double word*Note 3)	D(44h)	%MD000, %LD000, %KD000, %FD000, %ID0.0.0, %QD0.0.0, %MD0, etc.
Long word*Note 4)	L(4Ch)	%ML000, %LL000, %KL000, %FL000, %IL0.0.0, %QL0.0.0, %ML0, etc.

Notice

Note 1) Bit continuous read is not allowed in the case of continuous read.
Note 2) Bit continuous write is not allowed in the case of continuous write.
Note 3) Double word: 1 double word is 4 byes when converted into byte.
Note 4) Long word: 1 long word is 8 bytes when converted into byte

(3) Available device area

Area	Range	Size(Word)	Remarks
M	MW0 – MW1048575	1048576	Read/Write/Monitor available

K	KW0 – KW9215	9216	Read/Monitor available
F	FW0 – FW65535	65536	Read/Monitor available
L	LW0 – LW11263	11264	Read/Write/Monitor available
U	UW0.0.0– UW0.15.31	512	Read/Write/Monitor available
I	IW0.0.0 –IW127.15.3	8192	Read/Write/Monitor available
Q	QW0.0.0– QW127.15.3	8192	Read/Write/Monitor available

3)Individual writing of direct variables (W(w)SS)

This function is used to directly specify the PLC device memory to be used and write it according to the memory data type.

(1) Example of the individual write request frame of XGT client

Division	Header	Station number	Command	Command type	Number of blocks	Variable length	Variable name	Data	...	Tail	Frame check
Frame	ENQ	20	W(w)	SS	01	06	%MW100	00E2	...	EOT	BCC
Hex	h05	h3230	h57(77)	h5353	h3031	h3036	h254D57313030	h30304532	...	h04	

(2) Example of the response frame of XGT server

a)ACK response

Division	Header	Station number	Command	Command type	Tail	Frame check
Frame	ACK	20	W(w)	SS	ETX	BCC
Hex	h06	h3230	h57(77)	h5353	h03	

b) NAK response

Division	Header	Station number	Command	Command type	Error code (2 bytes)	Tail	Frame check
Frame	NAK	20	W(w)	SS	4252	ETX	BCC
Hex	h15	h3230	h57(77)	h5353	h34323532	h03	

c) Meanings of each item

Division	Descriptions
Number of blocks	▶ The number of blocks consisting of variable length + variable name - Maximum setting: 16 blocks - Setting range: 01(Hex value:3031) ~ 10(HexI value:3130)
Variable length	▶The number of characters in the variable name - Maximum setting: 16 - Setting range: 01(Hex value:3031) ~ 10(Hex value:3130) Example) If the variable name is %MW0, the variable length is h04 since the number of characters is 4. If the variable name is %MW000, the variable length is h06 since the number of characters is 6.
Variable name	▶ Address of write device - Setting range: 12 characters or less - Notice: Not allowed except for digits, upper/lower case frames, and '%'
Data	▶If the value you want to write in the %MW100 area is h A, the format of the data should be h000A. -Usage example If the data type you want to write is word, and the data to be written is h1234, its ASCII code conversion value is 31323334, and this content should be contained in the data area. That is, the highest value is transmitted first, and the lowest value should be transmitted for the last time.
Frame check	▶The BCC value is added if the command is a lowercase frame example (w), whereas the BCC value is not added if the command is an uppercase frame example (W). ▶If the command is the lowercase frame example (w), values ranging from ENQ to EOT are converted into Hex values, and only 1 low-order byte of the result obtained by adding one byte is added to the BCC.

Notice

(1) The device data type of each block should be the same.
(2) If the data type is a bit, the data to be written should be represented by 1 byte in hexadecimal.
That is, if bit value is 0, it should be h00(3030), and if it is 1, h01(3031).

(3) Usage example

This example supposes that ' hFF'is written in M0230 of station No. 1

a) Individual write request frame of XGT client

Division	Header	Station number	Command	Command type	Number of blocks	Variable length	Variable name	Data	Tail	Frame check
Frame	ENQ	01	W(w)	SS	01	06	%MW230	00FF	EOT	BCC
Hex	h05	h3031	h57(77)	h5353	h3031	h3036	h254D57323330	h30304646	h04	

b) Response frame of XGT server

- ACK response

Division	Header	Station number	Command	Command type	Tail	Frame check
Frame	ACK	01	W(w)	SS	ETX	BCC
Hex	h06	h3031	h57(77)	h5353	h03	

- NAK response

Division	Header	Station number	Command	Command type	Error code	Tail	Frame check
Frame	NAK	01	W(w)	SS	Error code (2 bytes)	ETX	BCC
Hex	h15	h3031	h57(77)	h5353	Error code (4 bytes)	h03	

4) Individual reading of direct variables(R(r)SS)

This function is used to direct specify and read the PLC device according to the data type.
It makes it possible to read 16 independent device memories at a time.

(1) Example of the individual read request frame of XGT client

Division	Header	Station number	Command	Command type	Number of blocks	Variable length	Variable name	· · ·	Tail	Frame check
Frame	ENQ	20	R(r)	SS	01	06	%MW100	· · ·	EOT	BCC
Hex	h05	h3230	h52(72)	h5353	h3031	h3036	h254D57313030	· · ·	h04	

(2) Example of the response frame of XGT server

a)ACK response

Division	Header	Station number	Command	Command type	Number of blocks	Number of data	Data	.....	Tail	Frame check
Frame	ACK	20	R(r)	SS	01	02	A9F3		ETX	BCC
Hex	h06	h3230	h52(72)	h5353	h3031	h3032	h41394633		h03	

b) NAK response

Division	Header	Station number	Command	Command type	Error code (2 bytes)	Tail	Frame check
Frame	NAK	20	R(r)	SS	1132	ETX	BCC
Hex	h15	h3230	h52(72)	h5353	h31313332	h03	-

(3) Meanings of each item

Division	Descriptions
Number of blocks	► The number of blocks consisting of variable length + variable name ▷Maximum setting: 16 blocks ▷Setting range: 01(ASCII code:3031) ~ 10(ASCII code:3130)
Variable length	►The number of characters in the variable name ▷Maximum setting: 16 ▷Setting range: 01(ASCII code:3031) ~ 10(ASCII code:3130) ▷Example) If the variable name is %MW0, the variable length is h04 since the number of characters is 4. If the variable name is %MW000, the variable length is h06 since the number of characters is 6.
Variable name	►Address of write device ▷Setting range: 12 characters or less ▷Notice: Not allowed except for digits, upper/lowercase frames, and '%'

Division	Descriptions

<table>
<tr><td>Number of data</td><td>▶It means the number of bytes of Hex type and is converted into ASCII.
▶ The number is determined by the data type(X,B,W,D,L) contained in the direct variable name of the external communication device request format
▷The number of data according to the variable type is shown below.

<table>
<tr><th>Data type</th><th>Available direct variables</th><th>Number of data</th></tr>
<tr><td>Bit(X)</td><td>%(P,M,L,K,F,T,C,I,Q,W,R)X</td><td>1</td></tr>
<tr><td>Byte(B)</td><td>%(P,M,L,K,F,T,C,I,Q,W,R)B</td><td>1</td></tr>
<tr><td>Word(W)</td><td>%(P,M,L,K,F,T,C,I,Q,W,R)W</td><td>2</td></tr>
<tr><td>Double word(D)</td><td>%(P,M,L,K,F,T,C,I,Q,W,R)D</td><td>4</td></tr>
<tr><td>Long word(L)</td><td>%(P,M,L,K,F,T,C,I,Q,W,R)L</td><td>8</td></tr>
</table></td></tr>
<tr><td>Data</td><td>▶ The value obtained by converting the data of area hexadecimal into ASCII code is stored
▷Usage example 1
If the number of data is h04 (ASCII code: h3034), it indicates that there are 4 bytes of Hex data in the data. That is, 4 bytes of hexadecimal data is converted into ASCII code in the data.

▷Usage example 2
If the number of data is h04, and the data is h12345678, its ASCII code conversion value is “31 32 33 34 35 36 37 38”, and this content is contained in the data area. That is, the highest value comes first, and the lowest number is transmitted for the last time.</td></tr>
</table>

Notice

1) If the data type is a bit, the read data is displayed in the form of a byte.
That is, if bit value is 0, it is displayed as h00, and if it is 1, h01.

(4) Usage example

This example supposes that 1 word is read from M0001 and M0020 of station No. 1,

(It is assumed that h1234 is contained in M0020, and M0001 contains h5678.)

a) Individual read request frame of XGT client

Division	Header	Station number	Command	Command type	Number of blocks	Variable length	Variable name	Variable length	Variable name	Tail	Frame check
Frame	ENQ	01	R(r)	SS	02	06	%MW020	06	%MW001	EOT	BCC
Hex	h05	h3031	h52(72)	h5353	h3032	h3036	h254D57303230	h3036	h254D5730303031	h04	

b) Response frame of XGT server

- ACK response

Division	Header	Station number	Command	Command type	Number of blocks	Number of data	Data	Number of data	Data	Tail	Frame check
Frame	ACK	01	R(r)	SS	02	02	1234	02	5678	ETX	BCC
Hex	h06	h3031	h52(72)	h5353	h3032	h3032	h31323334	h3032	h35363738	h03	

- NAK response

Division	Header	Station number	Command	Command type	Error code	Tail	Frame check
Frame	NAK	01	R(r)	SS	Error code(2 bytes)	ETX	BCC
Hex	h15	h3031	h52(72)	h5353	Error code(4 bytes)	h03	

5) Continuous writing of direct variables (W(w)SB)

This function is used to write the data of the specified length continuously from the specified address of the device.

(1) Example f the continuous write request frame of XGT client

Division	Header	Station number	Command	Command type	Variable length	Variable name	Number of data	Data	Tail	Frame check
Frame	ENQ	10	W(w)	SB	06	%MW100	02	11112222	EOT	BCC
Hex	h05	h3130	h57(77)	h5342	h3036	h254D57313030	h3034	h3131313132323232	h04	

(2) Example of the response frame of XGT server

a)ACK response

Division	Header	Station number	Command	Command type	Tail	Frame check
Frame	ACK	10	W(w)	SB	ETX	BCC
Hex	h06	h3130	h57(77)	h5342	h03	

b) NAK response

Division	Header	Station number	Command	Command type	Error code (Hex 2 bytes)	Tail	Frame check
Frame	ENQ	10	W(w)	SB	1132	ETX	BCC
Hex	h05	h3130	h57(77)	h5342	h31313332	h03	

(3) Meanings of each item

Division	Descriptions
Variable name	It means the start address of the device to perform continuous write.
Number of data	The number of data specifies the number according to the type of direct variable. That is, if the device data type is word, the number of data is 5, it means that 5 words need to be written. The maximum number of data is 120 bytes for Hex, and it is 240 bytes when converted to ASCII value.

(4) Usage example

This example supposes that 2 bytes of hAA15 is written in M000 of station No. 1.

a) Continuous write request frame of XGT client

Division	Header	Station number	Command	Command type	Variable length	Variable name	Number of data	Data	Tail	Frame check
Frame	ENQ	01	W(w)	SB	06	%MW000	01	AA15	EOT	BCC
Hex	h05	h3031	h57(77)	h5342	h3036	h254D5730 3030	h3031	h41413135	h04	

b) Response frame of XGT server

- ACK response

Format name	Header	Station number	Command	Command type	Tail	Frame check
Frame	ACK	01	W(w)	SB	ETX	BCC
Hex	h06	h3031	h57(77)	h5342	h03	

- NAK response

Format name	Header	Station number	Command	Command type	Error code	Tail	Frame check
Frame	NAK	01	W(w)	SB	Error code(2)	ETX	BCC
Hex	h15	h3031	h57(77)	h5342	Error code(4)	h03	

6) Continuous reading of direct variables(R(r)SB)

This function is to continuously read as much data as the specified amount from the designated address of the

PLC device.

(1) Example of continuous read request frame of XGT client

Division	Header	Station number	Command	Command type	Variable length	Variable name	Number of data (Up to 240 bytes)	Tail	Frame check
Frame	ENQ	10	R(r)	SB	06	%MW100	05	EOT	BCC
Hex	h05	h3130	h52(72)	h5342	h3036	h254D57313030	h3035	h04	

(2) Example of the response frame of XGT server

a)ACK response

Division	Header	Station number	Command	Command type	Number of blocks	Number of data	Data	Tail	Frame check
Frame	ACK	10	R(r)	SB	01	02	1122	ETX	BCC
Hex	h06	h3130	h52(72)	h5342	h3031	h3032	h31313232	h03	

b) NAK response

Division	Header	Station number	Command	Command type	Error code (Hex 2 byte)	Tail	Frame check
Frame	NAK	10	R(r)	SB	1132	ETX	BCC
Hex	h15	h3130	h52(72)	h5342	h31313332	h03	

(3) Meanings of each item

Division	Descriptions
Number of data	▶It means the number of bytes of Hex type and is converted to ASCII. ▶The number indicates the number of bytes. <table><tr><th>Data type</th><th>Available direct variables</th><th>Number of data</th></tr><tr><td>Bit(X)</td><td>%(P,M,L,K,F,T,C,I,Q,W,R)X</td><td>1</td></tr><tr><td>Byte(B)</td><td>%(P,M,L,K,F,T,C,I,Q,W,R)B</td><td>1</td></tr><tr><td>Word(W)</td><td>%(P,M,L,K,F,T,C,I,Q,W,R)W</td><td>2</td></tr><tr><td>Double word(D)</td><td>%(P,M,L,K,F,T,C,I,Q,W,R)D</td><td>4</td></tr><tr><td>Long word(L)</td><td>%(P,M,L,K,F,T,C,I,Q,W,R)L</td><td>8</td></tr></table>
Data	▶The data area contains the value obtained by converting Hex data to ASCII code ▷Usage example 1 If the memory type included in the direct variable name of the PC request format is W(word), and the number of data in the PC request format is 03, the number of PLC ACK response data after executing the command is h06(2*03 = 06 bytes)bytes, and this value is ASCII code value 3036. ▷ Usage example 2 In the above example, if the 3-word data contents are 1234,5678,9ABC in order, the actual ASCII code conversion value is 31323334 35363738 39414243, and this content is contained in the data area.

(4) Usage example

This example supposes that 2 words is read from M000 of station No. 10(h0A).

(It is assumed that the following data is contained in M000 and M001.)
M000 = h1234
M001 = h5678

a) Continuous read request frame of XGT client

Division	Header	Station number	Command	Command type	Variable length	Variable name	Number of data	Tail	Frame check
Frame	ENQ	0A	R(r)	SB	06	%MW000	02	EOT	BCC
Hex	h05	h3041	h52(72)	h5342	h3036	h254D30 3030	h3032	h04	

b) Response frame of XGT server
- ACK response

Division	Header	Station number	Command	Command type	Number of data	Data	Tail	Frame check
Frame	ACK	0A	R(r)	SB	04	12345678	ETX	BCC
Hex	h06	h3041	h52(72)	h5342	h3034	h3132333435363738	03	

- NAK response

Division	Header	Station number	Command	Command type	Error code	Tail	BCC
Frame	NAK	0A	R(r)	SB	Error code(2 bytes)	ETX	BCC
Hex	h15	h3041	h52(72)	h5342	Error code(4 bytes)	h03	

7) Registration and Execution of Monitor Variables

(1) Registration of monitor variables(X##)

Monitor register can register up to 32 (0 to 31) variables individually in combination with the actual variable reading command, and execute what is registered by monitor command after registration.

(a) Example of the monitor variable registration frame of XGT client

Structure	Header	Station number	Command	Registration number	Registration format	Tail	Frame check
Frame	ENQ	01	X(x)	09	Refer to registration format	EOT	BCC
Hex	h05	h3031	h58(78)	h3039	*Note 1)	h04	

(b) Example of the monitor variable response frame of XGT server

a) ACK response

Structure	Header	Station number	Command	Registration number	Tail	Frame check
Frame	ACK	01	X(x)	09	ETX	BCC
Hex	h06	h3031	h58(78)	h3039	h03	

b) NAK response

Structure	Header	Station number	Command	Registration number	Error code (Hex 2 bytes)	Tail	Frame check
Frame	NAK	01	X(x)	09	h1132	ETX	BCC
Hex	h15	h3031	h58(78)	h3039	h31313332	h03	

(c) Meanings of each item

Division	Descriptions
Registration number	Up to 32 numbers (0~31, h00~h1F) can be registered. If you register the already registered number again, what is currently executed is registered.
Registration format	It is used before EOT in the command among the device individual reading and continuous reading formats.

Notice

Note1) Please be sure to select one of the following registration formats for request formats.

▶Individual reading of device

RSS	Number of blocks(2 bytes)	Variable length(2 bytes)	Variable name(16 bytes)	...

1 block(Max. 16 blocks)

▶Continuous reading of device

RSB	Variable length (2 bytes)	Variable name (16 bytes)	Number of data

(d) Usage example

This example supposes that the device M0000 of station No. 1 is registered as No. 01.

a) Monitor variable registration frame of XGT client

Division	Header	Station number	Command	Registration number	Registration format				Tail	Frame check
					Command type	Number of blocks	Variable length	Variable name		
Frame	ENQ	01	X(x)	01	RSS	01	06	%MW000	EOT	BCC
Hex	h05	h3031	h58(78)	h3031	h525353	h3031	h3036	h255457303030	h04	

b) Monitor variable response frame of XGT

- ACK response

Division	Header	Station number	Command	Registration number	Tail	Frame check
Frame	ACK	01	X(x)	01	ETX	BCC
Hex	h06	h3031	h58(78)	h3031	h03	

- NAK response

Division	Header	Station number	Command	Registration number	Error code	Tail	Frame check
Frame	NAK	01	X(x)	01	Error code(2)	ETX	BCC
Hex	h15	h3031	h58(78)	h3031	Error code(4)	h03	

(2) Monitor execution(Y##)

Monitor execution is a function to execute the reading of device registered by monitor register. It specifies the registered number and executes the reading of device registered by the number.

(a) Example of the monitor execution frame of XGT client

Division	Header	Station number	Command	Registration number	Tail	Frame check
Frame	ENQ	10	Y(y)	09	EOT	BCC
Hex	h05	h3130	h59(79)	h3039	h03	

(b) Example of the monitor execution response frame of XGT server

a) ACK response

- If the registration format of registration number is individual reading of device

Division	Header	Station number	Command	Registration number	Number of blocks	Number of data	Data	Tail	Frame check
Frame	ACK	10	Y(y)	09	01	02	9183	ETX	BCC
Hex	h06	h3130	h59(79)	h3039	h3031	h3032	h39313833	h03	

-If the registration format of registration number is continuous reading of direct variable

Division	Header	Station number	Command	Registration number	Number of data	Data	Tail	Frame check
Frame	ACK	10	Y(y)	09	04	9183AABB	ETX	BCC
Hex	h06	h3130	h59(79)	h3039	h3034	h3931383341414242	h03	

b) NAK response

Division	Header	Station number	Command	Registration number	Error code (Hex 2 byte)	Tail	Frame check
Frame	NAK	10	Y(y)	09	1132	ETX	BCC
Hex	h15	h3130	h59(79)	h3039	h31313332	h03	

(c) Usage example

This example supposes that the reading of device registered as registration No. 1 in station No.1.

It is assumed that what is registered is device M000, and its number of blocks is 1.

a) Monitor execution registration frame of XGT client

Division	Header	Station number	Command	Registration number	Tail	Frame check
Frame	ENQ	01	Y(y)	01	EOT	BCC
Hex	h05	h3031	h59(79)	h3031	h04	

b) Monitor execution response frame of XGT server

- ACK response

Division	Header	Station number	Command	Registration number	Number of blocks	Number of data	Data	Tail	Frame check
Frame	ACK	01	Y(y)	01	01	02	2342	ETX	BCC
Hex	h06	h3031	h59(79)	h3031	h3031	h3032	h32333432	h03	

- NAK response

Division	Header	Station number	Command	Registration number	Error code	Tail	Frame check
Frame	NAK	01	Y(y)	01	Error code(2)	ETX	BCC
Hex	h15	h3031	h59(79)	h3031	Error code(4)	h03	

15.8.3 XGT communication function

1) Summary

XGT communication operates as XGT server or P2P service depending on what setting of Cnet I / F module operation mode. Each mode must be set to XG5000.

(1) XGT Server

(a) It makes it possible to read or write PLC information or data to PC or peripheral devices without writing a separate program in PLC.

(b) The XGT client responds to frames requested.

(2) P2P Service

(a) Cnet This service allows I / F module to operate as a client on the network.

(b) When a specified event occurs, the memory of the other station can be read or written.

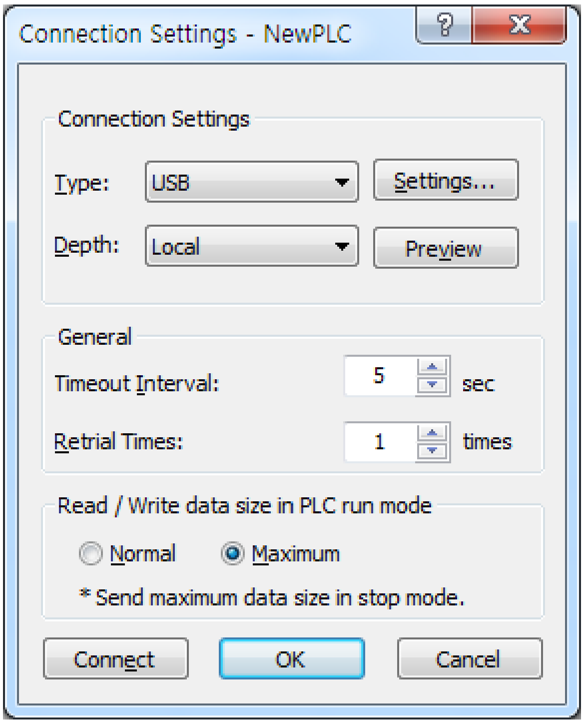
(c) Up to 64 independent P2P blocks can be defined per channel.

2) Setting parameter when using as XGT server

(1) Connection setup

(a) Select [Online] → [Connection Settings].

(b) Set connection options for your environment and click [Connect].

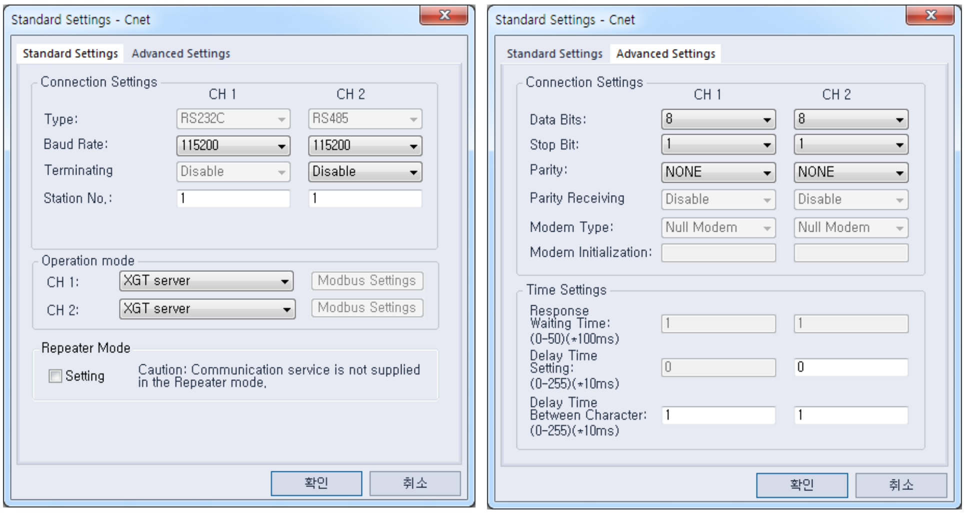


(2) Basic setting

(a) Double-click the corresponding XMC built-in Cnet to open the [Preferences] window and set the communication type, communication speed, modem type, data bit, stop bit and station number in the [Connection Settings] menu.

(b) The modem function is not available.

(c) The delay time can be set only when the communication type is RS485, and the response wait time can be set only when the communication type is RS485 and the operation mode is P2P.



(3) Select operating mode

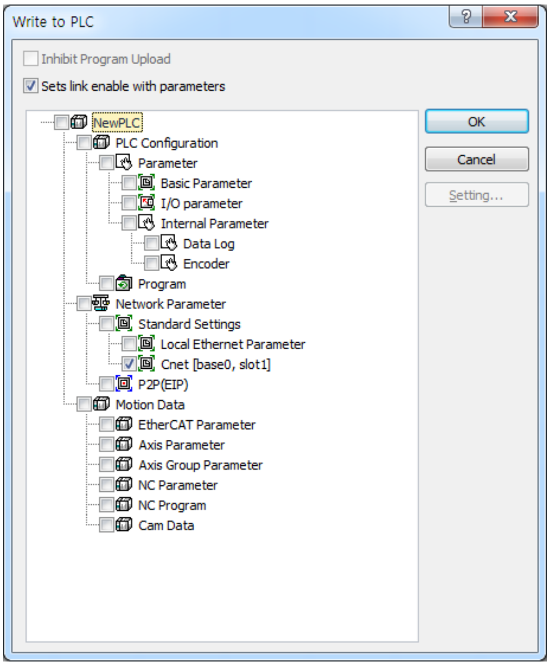
Select the XGT server.

(4) Write parameters

(a) Select [Online] → [Write] or click the icon ().

(b) Check (✓) the module with the default settings and click [OK].

(c) Click the [OK] button, and when the parameter writing finishes, reset each module.

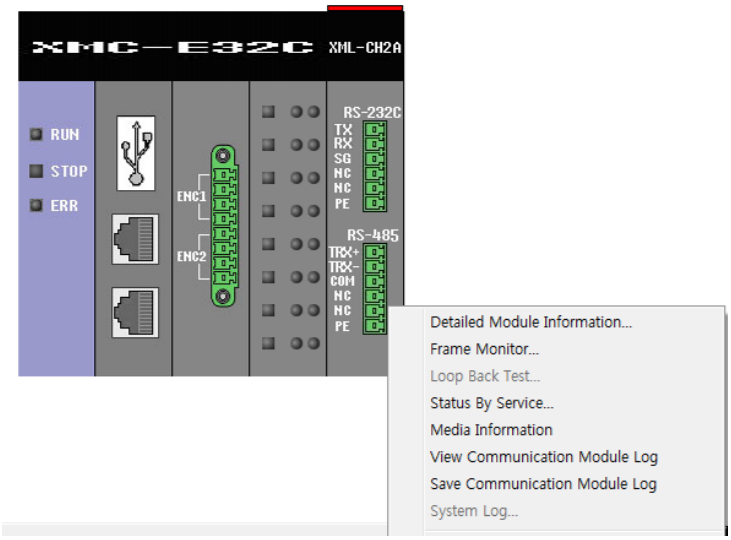


(5) Check operation

(a) Select [Online] → [Communication Module Settings] → [System Diagnosis] or click the icon ( ).

(b) Click the communication module whose status you want to diagnose and press the right mouse button.

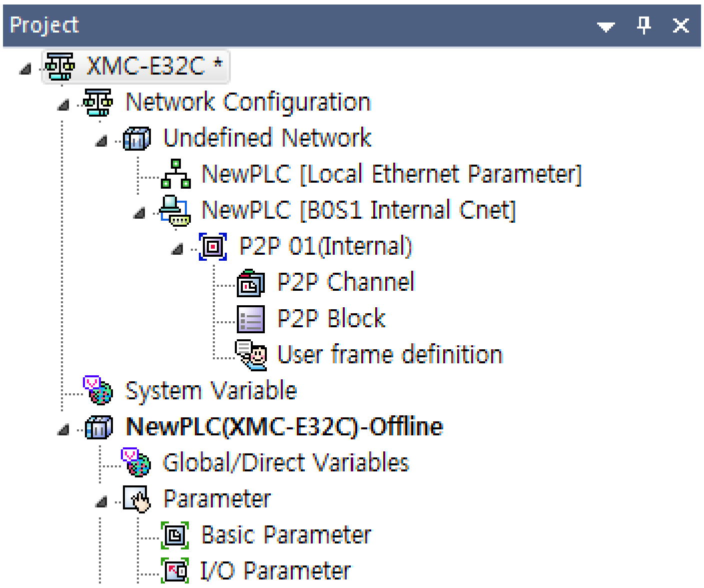
(c) When the following screen appears, click [Frame Monitor] or [Service Status] to check the operation status.



3) Setting parameter when using as XGT client

(1) P2P parameter configuration

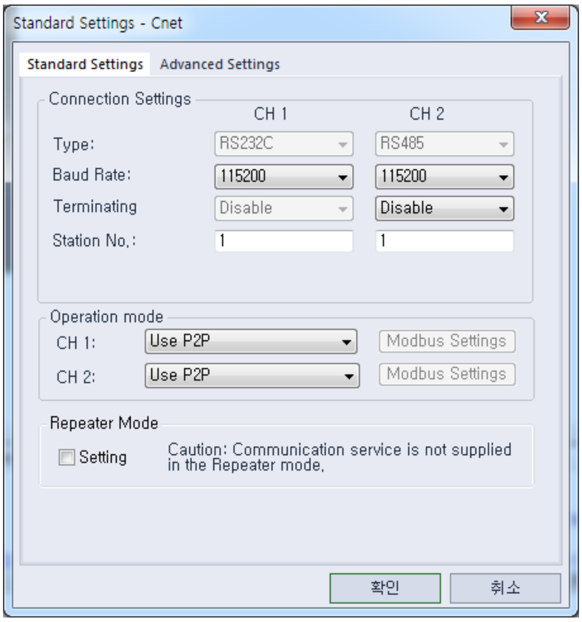
In order to use P2P service, a user should perform settings for desired operation in P2P parameter window. The P2P parameter consists of three pieces of information as shown below.

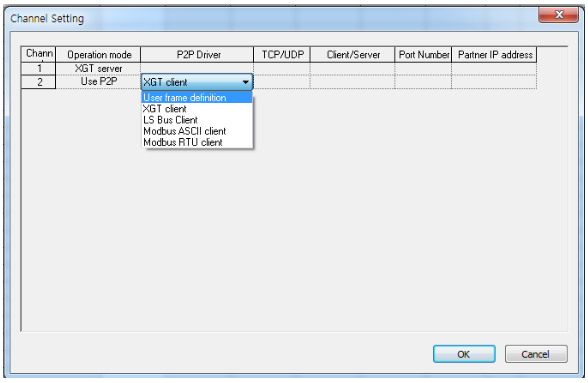


Division	Contents
P2P channel	- Set P2P channel to define the communication protocol of P2P service to perform - XGT/Modbus available - It is possible to set independently for each channel. It is applied only when the operation mode of the default settings is “Use P2P”.
P2P block	Set 64 P2P blocks that operate independently
User-defined frame registration	Set frame for user-defined communication

(2) P2P channel setting

Cnet I / F module provides two independent communication channels. You can define the driver type to perform P2P service for each channel. However, for the P2P channel to function as a client, be sure to select 'Use P2P' in the [Preferences] window. P2P channel setting according to operation mode is as follows





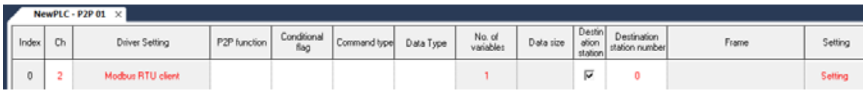
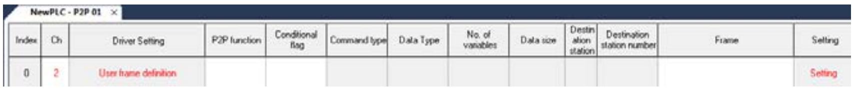
The following drivers can be selected when 'Enable P2P' is selected in the operation mode.

Driver	Meaning
None	P2P service is not used
User frame definition	Used when it sends/receives the desired user-defined frame
XGT client	Selected when it performs memory read/write of XGT
LS Bus client	Selected when it communicates with the inverter of its company
Modbus ASCII client	Selected when it operates as a Modbus client and is used in ASCII mode
Modbus RTU client	Selected when it operates as a Modbus client and is used in RTU mode

If the P2P driver is selected as XGT or Modbus, user frame definition cannot be used.

(3) P2P block settings

If you select the P2P block of the parameter in the P2P parameter setting window, P2P block setting window appears. The set contents for the P2P block vary depending on the channel status set by a user.

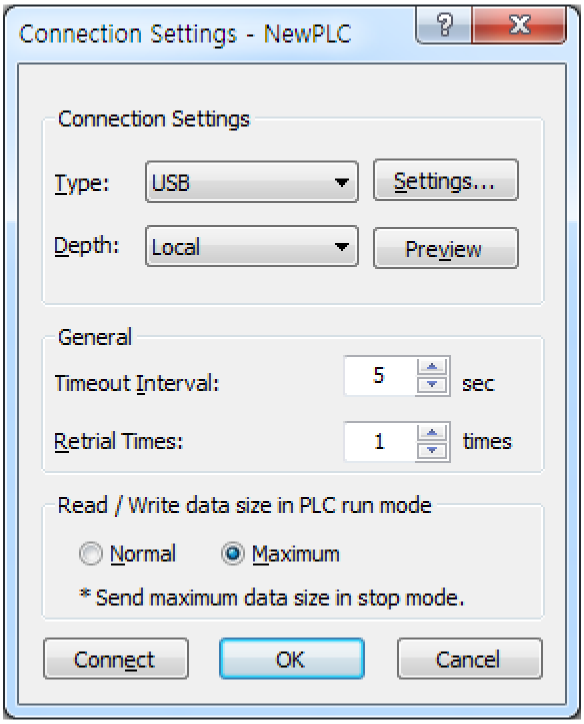
P2P channel	P2P block settings
Chann / Operation mode / P2P Driver 1 / XGT server / 2 / Use P2P / XGT client	
Chann / Operation mode / P2P Driver 1 / XGT server / 2 / Use P2P / Modbus ASCII client	
Chann / Operation mode / P2P Driver 1 / XGT server / 2 / Use P2P / Modbus RTU client	
Chann / Operation mode / P2P Driver 1 / XGT server / 2 / Use P2P / User frame definition	
Chann / Operation mode / P2P Driver 1 / XGT server / 2 / Use P2P / LS Bus Client	

(4) Setting parameter when using as XGT server

1) Connection setup

(a) Select [Online] → [Connection Settings].

(b) Set connection options for your environment and click [Connect].

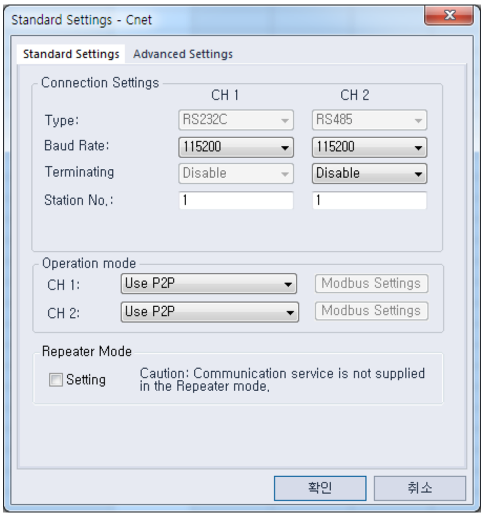


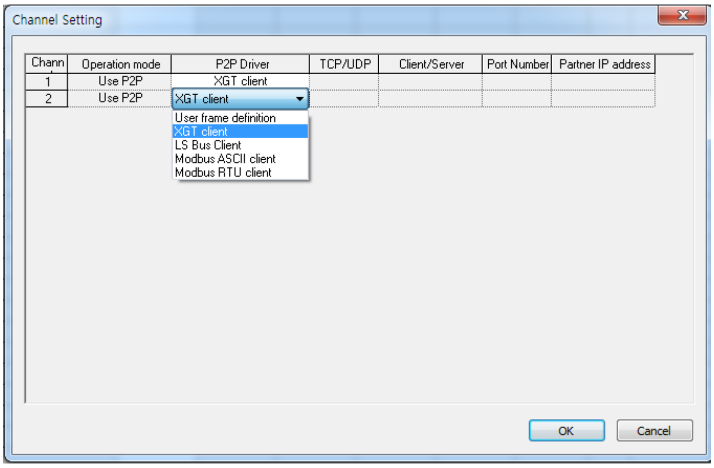
2) Basic setting

(a) Double-click the corresponding XMC built-in Cnet to open the [Preferences] window and set the communication type, communication speed, modem type, data bit, stop bit and station number in the [Connection Settings] menu.

(b) The modem function is not available.

(c) The delay time can be set only when the communication type is RS485, and the response wait time can be set only when the communication type is RS485 and the operation mode is P2P.

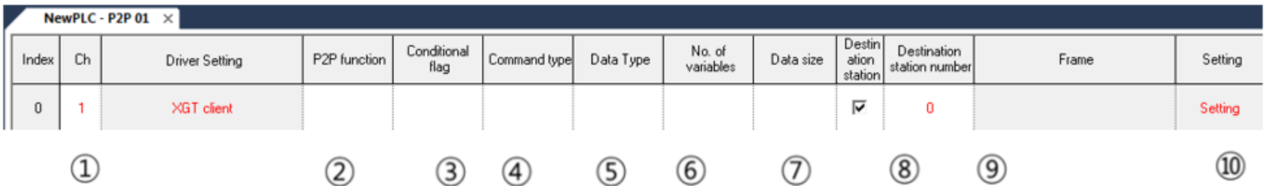


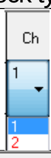
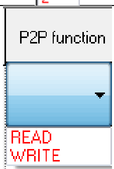


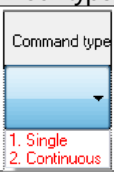
3) Operation mode

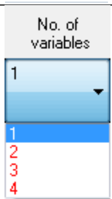
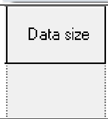
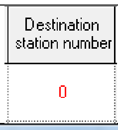
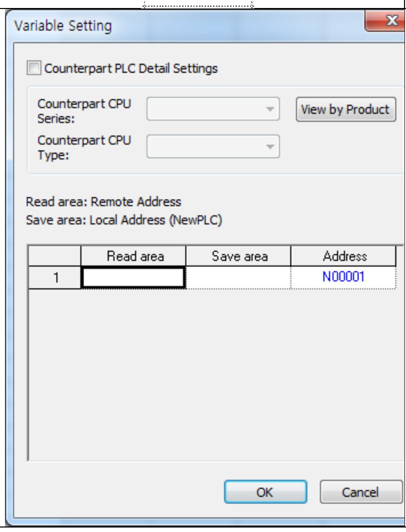
Select 'Use P2P'.

(a) Setting P2P parameters



Number	Classification	Block type	Contents
1	Channel		The set drive name changes according to the driver set in P2P driver.
2	P2P function		1. Read: Used to read arbitrary data from the partner station 2. Write: Used to write arbitrary data to the partner station
3	Start condition		1. Enter special flag or bit contact to select the time when data is transmitted and received 2. Example: _T20MS(cycle: 20ms), %MX0(when MX0 is On)

Number	Classification	Block type	Contents
4	Mode		1.Individual: Used to read or write data of up to 4 memory areas to the other station(Example: M01, M10, M20, M30) 2.Continuous: Used to read or write continuous data to the other station (Example: M01~M10)

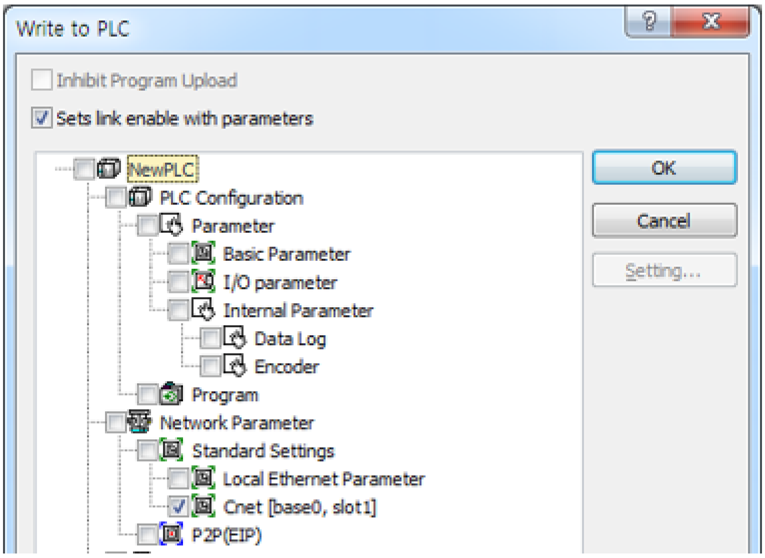
5	Data type		1. Individual mode: Data types are divided into five types: bit, 1 byte, 2 byte, 4 byte and 8 byte. 2. Continuous mode: Data types are divided into four types: 1 byte, 2 byte, 4 byte and 8 byte
6	Number of variables		1. This function is enabled only when the individual mode is selected, and the number of data to be transmitted and received is selected. However, the maximum allowable number of data is four. 2. If continuous mode is selected, it is fixed to 1.
7	Data size		1. This function is enabled only when the continuous mode is selected, and it is possible to set up to 120 bytes when the data type is 1 byte.
8	Partner station		1.It is checked automatically. If you do not use the block, click the check mark once again and the block will not work.
9	Partner station number		1. It means the station number of the partner station. 1) XGT dedicated communication: A total of 64 station numbers can be set from 0 to 63 stations.
10	Settings		1. When the P2P function is Read 1)Read area: Device area where data of the other station (server) is stored 2)Storage area: Device area of its station (server) to store data which is read from the other station 2. When the P2P function is Write 1)Read area: Device area where data of its station is stored 2)Storage area: Device area of the other station to store data of its station

(b) Write parameters

1) Select [Online] → [Write] or click the icon ().

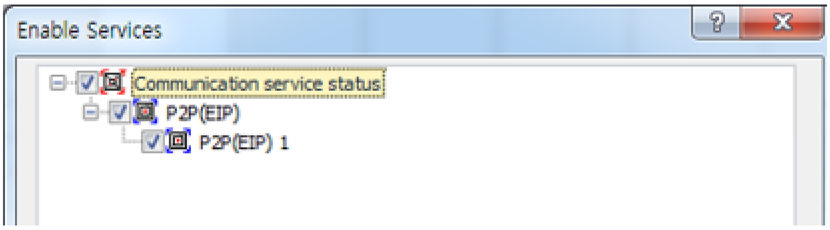
2) Check (✓) the module with the default settings and click [OK].

3) Click the [OK] button, and when the parameter writing finishes, reset each module.



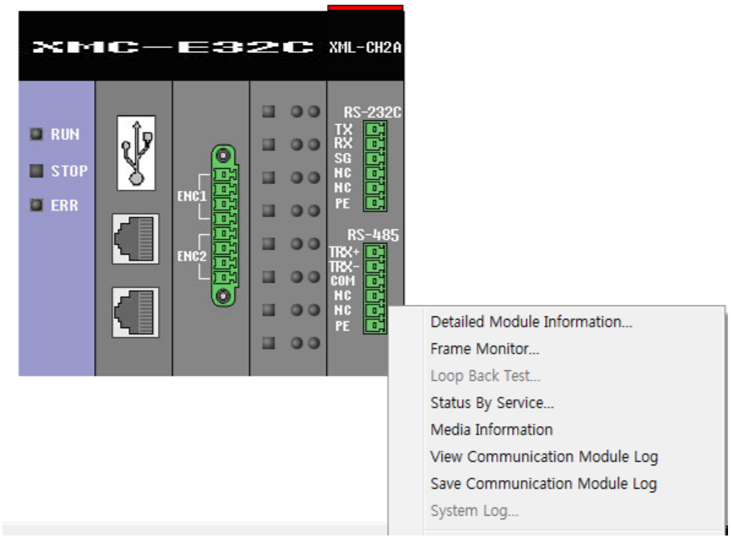
(c) Link Enable

1) Select [Online] → [Communication module settings] → [Link enable].
2) Check the P2P block to be used and click [Write].



(d) Check operation

1) Select [Online] → [Communication module settings] → [System diagnostics].
2) Click the communication module whose status you want to diagnose and press the right mouse button.
3) When the following screen appears, click [Frame Monitor] or [Service Status] to check the operation status.



4) Frame monitor

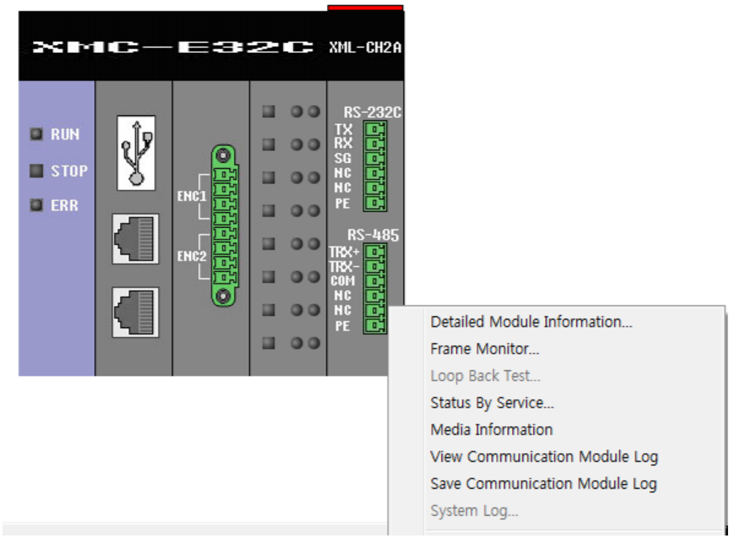
With the frame monitoring function provided by XG5000, you can check the frame that the client and server exchange.

(1) Check operation

a) Select [Online] → [Communication module settings] → [System diagnostics].

b) Click the communication module whose status you want to diagnose and press the right mouse button.

c) When the following screen appears, click [Frame Monitor] or [Service Status] to check the operation status.

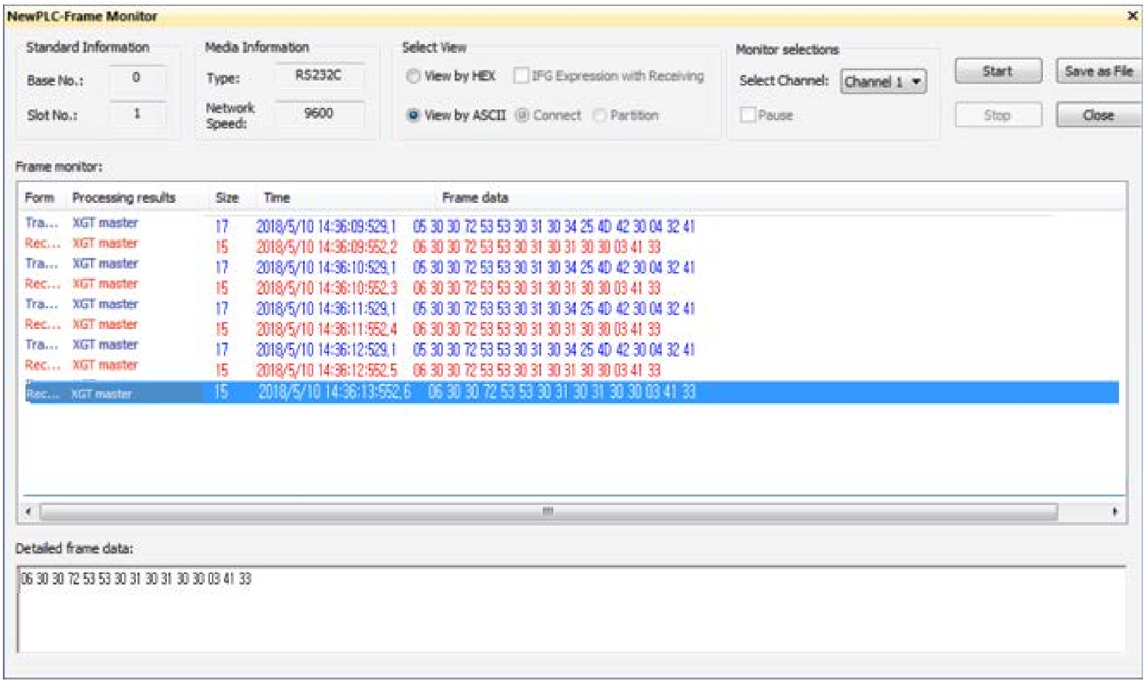


(2) Frame monitor

(a) Select the channel you want to monitor.

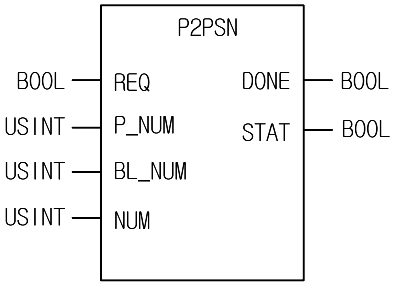
(b) Since the XGT protocol is ASCII communication, select the view in ASCII mode.

(c) Click [Start] to check the sending / receiving frame.



15.8.4 P2P Commands

1) P2PSN

Function block	Descriptions
	Input REQ :Function block execution request P_NUM : P2P number BL_NUM :Block number NUM :Station number Output DONE :Maintain 1 after initial operation STAT : Complete and ERR information

(1) Functions

a) The use of P2PSN command makes it possible to change the station number of P2P service partner during the run.

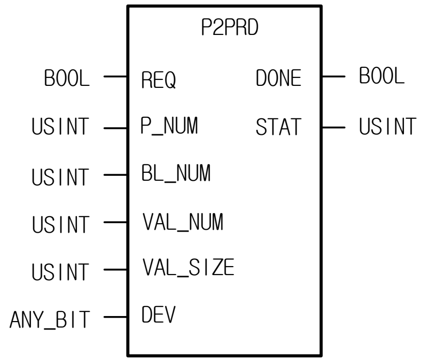
b) It changes BL_NUM block remote station number of P_NUM P2P to NUM.

(2) Errors

When an error occurs, the error number is displayed on STAT.

STAT_NUM	Contents	Details
1	P2P number setting	Occurs when a value other than cP_NUM(1~8) is set
2	Block number setting	Occurs when a value other than BL_NUM(0~63) is set
4	Slot does not exist	-
5	Module mismatch	Not a communication module
6	Module mismatch	Communication module that cannot be used for the command
7	Station number setting error	Occurs when a value other than NUM(0~63) is set

2) P2PRD

Function block	Descriptions
	Input REQ :Function block execution request P_NUM : P2P number BL_NUM :Block number VAL_NUM :Variable number VAL_SIZE :Variable size DEV :Device(only direct variables can be entered) Output DONE :Maintain 1 after initial operation STAT : Complete and ERR information

ANY type variable description	Variable name	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
	DEV	○	○	○	○	○															

(1) Functions

a) The P2PRD command changes the variable size of the P2P parameter block and the READ device area. (Individual/continuous reading can be changed.)

b) The P_NUM, BL_NUM and VAL_NUM are used to specify the P2P parameters, blocks and variables and change the variable size and device to VAL_SIZE(In the case of continuous reading, VAL_SIZE means variable size, and it is the size of variable type in the case of individual reading) and DEV, respectively. However, only direct variables can be entered for DEV. (e.g., %MW100)

(2) Errors

If the setting is outside the allowable range of P2P parameter set in XG5000, the corresponding error code occurs.

STAT_NUM	Contents	Details
1	P2P number setting error	Occurs when a value other than P_NUM(1~8) is set
2	Block number setting error	Occurs when a value other than BL_NUM(0~63) is set
3	Variable number setting error	Occurs when the variable number that is not allowed in the P2P parameter set in XG5000 is entered.
4	Slot does not exist	-
5	Module mismatch	Not a communication module
6	Module mismatch	Communication module that cannot be used for the command
10	Modbus setting error	Impossible to enter the offset of Modbus(e.g., 0x10000) because only direct variables can be entered for DEV
11	Variable size setting error	Occurs when the variable size that is not allowed in the P2P parameter set in XG5000 is entered
12	Data type setting error	Occurs when the variable type that is not allowed in the P2P parameter set in XG5000 is entered

3) P2PWR

Function block	Descriptions
P2PWR BOOL — REQ DONE — BOOL USINT — P_NUM STAT — USINT USINT — BL_NUM USINT — VAL_NUM USINT — VAL_SIZE ANY_BIT — DEV	Input REQ : Function block execution request P_NUM : P2P number BL_NUM :Block number VAL_NUM :Variable number VAL_SIZE :Variable size DEV :Device(only direct variables can be entered) Output DONE : Maintain 1 after initial operation STAT :Complete and ERR information

ANY type variable description	Variable name	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
	DEV	○	○	○	○	○															

(1) Functions

a) The P2PWR command changes the variable size of the P2P parameter block and the WRITE device area. (Individual/continuous writing can be changed.)

b) The P_NUM, BL_NUM and VAL_NUM are used to specify the P2P parameters, blocks and variables and change the variable size and device to VAL_SIZE(In the case of continuous writing, VAL_SIZE means variable size, and it is the size of variable type in the case of individual writing) and DEV, respectively. However, only direct variables can be entered for DEV. (e.g., %MW100)

(2) Errors

If the setting is outside the allowable range of P2P parameter set in XG5000, the corresponding error code occurs.

STAT_NUM	Contents	Details
1	P2P number setting error	Occurs when a value other than P_NUM(1~8) is set
2	Block number setting error	Occurs when a value other than BL_NUM(0~63) is set
3	Variable number setting error	Occurs when the variable number that is not allowed in the P2P parameter set in XG5000 is entered.
4	Slot does not exist	-
5	Module mismatch	Not a communication module
6	Module mismatch	Communication module that cannot be used for the command
10	Modbus setting error	Impossible to enter the offset of Modbus(e.g., 0x10000) because only direct variables can be entered for DEV
11	Variable size setting error	Occurs when the variable size that is not allowed in the P2P parameter set in XG5000 is entered
12	Data type setting error	Occurs when the variable type that is not allowed in the P2P parameter set in XG5000 is entered

15.9 LS Bus Protocol

15.9.1 LS Bus Protocol Architecture

LS Bus protocol communication is a protocol that is applied when communicating with the inverter of its company. The use of the data read/write function and monitoring function of various internal device areas makes it possible to easily construct a communication system intended by a user without settings specific to the inverter of its company.

LS Bus protocol functions provided by XMC are as follows.

- Continuous reading of device
- Continuous writing of device

1) Frame structure

(1) Basic structure

(a) Request frame (external communication device → Cnet)

Header (ENQ)	Station No.	Command	Structured data area	Frame check (BCC)	Tail (EOT)

(b) ACK response frame (Cnet → external communication device, when receiving data normally)

Header (ACK)	Station No.	Command	Structured data area	Frame check (BCC)	Tail (EOT)

(c) NAK response frame (Cnet → external communication device, when receiving data abnormally)

Header (NAK)	Station No.	Command	Error code(ASCII 4 Byte)	Frame check (BCC)	Tail (EOT)

Notice

(1) The numeric data of all frames is displayed in ASCII code for hexadecimal values unless otherwise specified. The items displayed in hexadecimal are as follows.
- Station number
- The command type supports R(read) and W(write).
- All contents of data

(2) For hexadecimal data, 'H' is prefixed to the number in the frame, as in H01, H12345, H34, H12, H89AB and etc. to indicate that the data is hexadecimal.
(3) The available frame length is up to 44 bytes.
(4) The control codes used are summarized below.

Code	Hex value	Name	Control contents
ENQ	H05	Enquire	Start code of request frame
ACK	H06	Acknowledge	Start code of ACK response frame
NAK	H15	Not Acknowledge	Start code of NAK response frame
EOT	H04	End of Text	Frame end ASCII code for request

(2) Command frame sequence

- Command request frame sequence

ENQ	Station No.	Command	Formatted data	BCC	EOT

ACK	Station No.	Command	Formatted data	BCC	EOT

(Inverter ACK response)

NAK	Station No.	Command	Formatted data	BCC	EOT

(Inverter NAK response)

2) List of commands

The command types used in the LS bus protocol are as follows.

Division / Item	Command type: Symbol	Command type: ASCII code	Processing contents
Continuous read	R	H52	Word-type inverter variables are read in word units.
Continuous write	W	H57	Word-type inverter variables are written in word units.

15.9.2 Command Details

1) Inverter continuous write (W)

This command is used to directly specify addresses in inverter and write them in word units.

(1) LS Bus client request format

Format name	Header	Station No.	Command	Data size	Inverter address	Data	.....	Frame check	Tail
Frame(example)	ENQ	H20	W	H6	0100	H00E2		BCC	EOT
ASCII code	H05	H3230	H57	H36	H30313030	H30304532		`	H04

Division	Descriptions
BCC	Only one low-order byte of the value obtained by adding one byte each to the ASCII value excluding the values of ENQ and EOT is converted into ASCII and added to BCC.
Data size	The number of words to be written is specified. When converted to ASCII, it ranges from H01 (ASCII value: 3031) to H08 (ASCII value: 3038).
Inverter address	The inverter address to be read is entered. It should be ASCII value with 4 characters, and only numbers are allowed.
Data	If the value to be written in the inverter address 0100 area is H'A, the data format should be H000A.

• Usage example

If the data type to be written is WORD, and the data to be written is H1234, its ASCII code conversion value is 31323334, and this content should be contained in the data area. That is, the highest value is transmitted first, and the lowest value should be transmitted for the last time.

Notice

• Only word is supported as a device data type.

(2) Inverter response format(ACK response)

Format name	Header	Station No.	Command	Data		Frame check	Tail
Frame(example)	ACK	H20	W	H00E2	…	BCC	EOT
ASCII value	H06	H3230	H57	H30304532			H04

Division	Descriptions
BCC	Only one low-order byte of the value obtained by adding one byte each to the ASCII value excluding the values of ENQ and EOT is converted into ASCII and added to BCC.

(3) Inverter response format(NAK response)

Format name	Header	Station No.	Command	Error code (ASC2 Byte)	Frame check	Tail
Frame(example)	NAK	H20	W	H12	BCC	EOT
ACSII value	H15	H3230	H57	H3132		H04

Division	Descriptions
BCC	Only one low-order byte of the value obtained by adding one byte each to the ASCII value excluding the values of ENQ and EOT is converted into ASCII and added to BCC.
Error code	The types of errors are indicated by the contents of 1Byte in Hex (2 Byte in ASCII code). Please refer to the error code of the inverter for details.

(4) Usage example

This example supposes that “H00FF” is written in 1230 of inverter station No. 1.

(a) XMC request format (XMC → inverter)

Format name	Header	Station No.	Command	Data length	Inverter address	Data	Frame check	Tail
Frame (example)	ENQ	H01	W	H1	1230	H00FF	BCC	EOT
ASCII value	H05	H3031	H57	H3031	H31323330	H30304646		H04

(b) ACK response after command execution (XMC ← inverter)

Format name	Header	Station No.	Command	Data	Frame check	Tail
Frame(example)	ACK	H01	W	H00FF	BCC	EOT
ASCII value	H06	H3031	H57	H30304646		H04

(c) NAK response after command execution (XMC ← inverter)

Format name	Header	Station No.	Command	Error code	Frame check	Tail
Frame(example)	NAK	H01	W	H12	BCC	EOT
ASCII value	H15	H3031	H57	Error code(2 Byte)		H04

2) Inverter continuous read(R)

This function is to continuously read as much data as the specified amount from the designated address of the PLC device.

(1) PC request format

Format name	Header	Station No.	Command	Inverter address	Number of data	Frame check	Tail
Frame (example)	ENQ	H10	R	0100	H5	BCC	EOT
ASCII value	H05	H3130	H52	H30313030	H35		H04

Division	Descriptions
BCC	Only one low-order byte of the value obtained by adding one byte each to the ASCII value excluding the values of ENQ and EOT is converted into ASCII and added to BCC.
Data size	The number of words to be written is specified. When converted to ASCII, it ranges from H01 (ASCII value: 3031) to H08 (ASCII value: 3038).
Inverter address	The inverter address to be read is entered. It should be ASCII value with 4 characters, and only numbers are allowed.

Notice

• Only word is supported as a device data type.

(2) Inverter response format(ACK response)

Format name	Header	Station No.	Command	Data		Frame check	Tail
Frame(example)	ACK	H20	R	H00E2	…	BCC	EOT
ASCII value	H06	H3230	H52	H30304532			H04

Division	Descriptions
BCC	Only one low-order byte of the value obtained by adding one byte each to the ASCII value excluding the values of ENQ and EOT is converted into ASCII and added to BCC.

(3) Inverter response format(NAK response)

Format name	Header	Station No.	Command	Error code (ASC2 Byte)	Frame check	Tail
Frame(example)	NAK	H20	R	H12	BCC	EOT
ACSII value	H15	H3230	H52	H3132		H04

Division	Descriptions
BCC	Only one low-order byte of the value obtained by adding one byte each to the ASCII value excluding the values of ENQ and EOT is converted into ASCII and added to BCC.
Error code	The types of errors are indicated by the contents of 1Byte in Hex(2 Byte in ASCII code). Please refer to the error code of the inverter for details

(4) Usage example

This example supposes that data 1 word from 1230 of the inverter station No. 1 is read.

(a) XMC request format (XMC → inverter)

Format name	Header	Station No.	Command	Inverter address	Data size	Frame check	Tail
Frame (example)	ENQ	H01	R	1230	H1	BCC	EOT
ASCII value	H05	H3031	H52	H31323330	H31		H04

(b) ACK response after command execution (XMC ← inverter)

Format name	Header	Station No.	Command	Data	Frame check	Tail
Frame(example)	ACK	H01	R	H1234	BCC	EOT
ASCII value	H06	H3031	H52	H31323334		H04

(c) NAK response after command execution (XMC ← inverter)

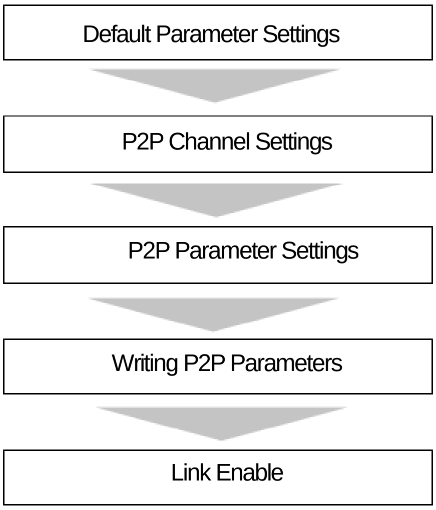
Format name	Header	Station No.	Command	Error code	Frame check	Tail
Frame(example)	NAK	H01	R	H12	BCC	EOT
ASCII value	H15	H3031	H52	H3132		H04

15.10 Modbus Protocol

15.10.1 Modbus Communication Setting Procedures

Modbus protocol is a standardized open protocol used for communication between client and server, and it can read/write data according to function code. The inter-device communication using the Modbus protocol uses client-server function that is processed by only one client

Procedures for sending/receiving data to/from communication devices by using Modbus communication are shown below.



15.10.2 Modbus Protocol

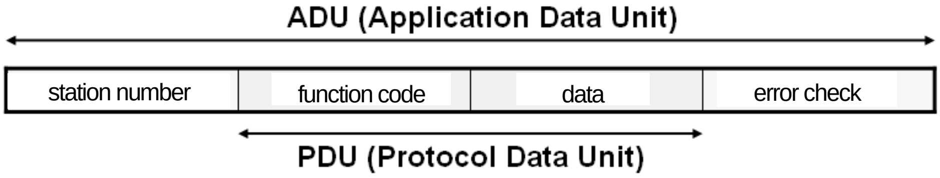
1) Protocol type

The communication modes of Modbus are divided into ASCII mode and RTU mode.

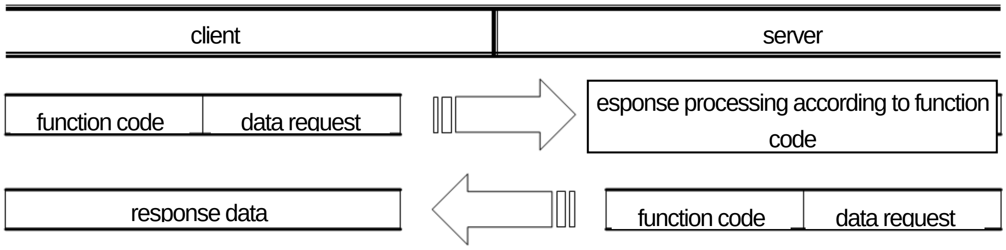
Characteristics		ASCII mode	RTU mode
Code system		ASCII code	8-bit binary code
Number of data per character	Start bit	1	1
	Data bit	7	8
	Parity bit	Even, Odd, None	Even, Odd, None
	Stop bit	1 or 2	1 or 2
Error check		LRC(Longitudinal Redundancy Check)	CRC (Cyclical Redundancy Check)
Start of frame		Colon (:)	3.5 character no-response time

2) Protocol configuration

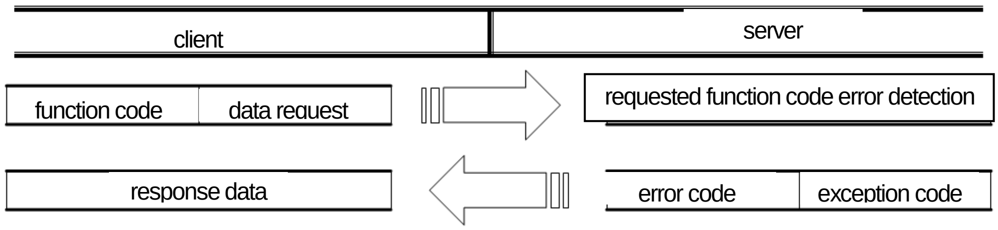
Modbus protocol is largely composed of PDU consisting of function code and data and ADU with the partner station number and error check added to PDU.



The processing procedure for normal Modbus communication is shown below.



When a Modbus communication error occurs, the server sends a response that includes the error code to the client as shown below.



When an abnormal frame is received, the server sends an error code and an exception code to the client. The error code is represented by adding 80(Hex) value to the function code, and the exception code indicates the details of the error. The contents for each code are shown below.

Code	Code name	Meaning
01	Function code error	Function code error
02	Address error	Address allowable range excess error
03	Data setting error	Data value is not allowed
04	Server station error	Server(slave)station is in error state
05	Retransmission request	Requests the client to make a request again at a proper time because the content to be processed is too huge for the server to handle at the moment
06	Processing time delay	The server station takes time to process. The master should make a request again.

15.10.3 Frame Structure

1) Frame structure in ASCII mode

The frame structure in Modbus ASCII mode is shown below.

Division	Start	Station No.	Function code	Data	Error check	End
Size(byte)	1	2	2	N	2	2

(1) Characteristics of ASCII mode
  (a) In ASCII mode, the start of a frame distinguished by a colon (:), which is a 1-byte ASCII code, and the end of a frame by ‘CRLF’.
  (b) It allows a maximum interval of 1 second between characters.
  (c) The error checking method uses LRC to convert ASCII by taking two’ s complement to the sum of a fame excluding the start/end of the frame to determine whether or not an error occurs.

(2) Address area
  (a) It consists of 2 bytes.
  (b) Station number can be set from 0 to 31 when using Cnet I/F.
  (c) Station No. 0 is used as the client station number.
  (d) The server responses with its own address in the response frame so that it can identify the client’s response when it responds.

(3) Data area
  (a) Data is transmitted using ASCII data, and the structure of data is changed according to each function code.
  (b) It responds with response data in response to a normal frame
  (c) Error code is used to respond when an abnormal frame is received.

(4) Error check area
  (a) The error checking method uses LRC to convert ASCII by taking two’ s complement to the sum of a fame excluding the start/end of the frame to determine whether or not an error occurs.

2) Frame structure in RTU mode

The frame structure in Modbus RTU mode is shown below.

Division	Start	Station No.	Function code	Data	Error check	End
Size(byte)	Idle time	1	1	N	2	Idle time

(1) Characteristics of RTU mode
(a) It communicates using hexadecimal numbers.
(b) The start character is the station number, and the end of a frame is distinguished by CRC error check.
(c) The start and end of the frame is distinguished by adding 1-bit idle time to the start and end of the frame.
(d) It has an interval of at least 3.5 character time between frames and recognizes the frame as an independent frame when 1.5 character time passes between characters.

(2) Address area
(a) It consists of 1 byte.
(b) Station number can be set from 0 to 31 when using XGT Cnet I/F module.
(c) Station No. 0 is used as the client station number.
(d) The server responses with its own address in the response frame so that it can identify the client’s response when it responds.

(3) Data area
(a) Data is transmitted using Hex data, and the data structure is changed according to each function code.
(b) It responds with response data in response to a normal frame
(c) Error code is used to respond when an abnormal frame is received.

(4) Error check area
(a) Two-byte CRC check method is used to determine whether the frame is normal or abnormal.

(5) Modbus address rule
(a) The address in the data starts from 0 and is equal to the value obtained by subtracting 1 from the Modbus memory. That is, Modbus address 2 is the same as the address 1 in the data.

3) Representation of data and address

The characteristics of representing the data and address of Modbus protocol are as follows.
(1) Hexadecimal (Hex.) data is used as a default format.
(2) Hex data is converted into ASCII code and used in ASCII mode.
(3) Hex data is used in RTU mode.
(4) The meanings for each function code are summarized below.

Code(Hex)	Use	Usage area	Address	Maximum response data
01	Bit individual/continuous write	Bit output	0XXXX	2000 bits
02	Bit individual/continuous read	Bit input	1XXXX	2000 bits
03	Word individual/continuous read	Word output	4XXXX	120 words
04	Word individual/continuous read	Word input	3XXXX	120 words
05	Bit individual write	Bit output	0XXXX	1 bit
06	Word individual write	Word output	4XXXX	1 word
0F	Bit continuous write	Bit output	0XXXX	1968 bits
10	Word continuous write	Word output	4XXXX	120 words

4) Reading data of bit format in bit output area (01)

(1) Reading bit of output area(function code: 01)

The structure of request and response frame when reading the data of bit format in output area is as follows. The tail of frame is applied only in ASCII code.

(a) Request frame

Frame	Station No.	Function code(01)	Address	Data size	Frame error check	Tail(CRLF)
Size(byte)	1	1	2	2	2	2

(b) Response frame (when receiving a normal frame)

Frame	Station No.	Function code(01)	Number of bytes	Data	Frame check error	Tail(CRLF)
Size(byte)	1	1	1	N	2	2

(c) Response frame (when receiving an abnormal frame)

Frame	Station No.	Error code	Exception code	Tail(CRLF)
Size(byte)	1	1	1	2

(2) Frame details

(a) Station number: It means the station number of the slave to read the bits of output area.

(b) Function code: '01' means bit continuous/individual reading of output area.

(c) Address: It means the start address of data to be read and consists of 2 bytes, when the start address conforms to the Modbus address rule.

(d) Data size: It is the size of data to read and consists of 2 bytes.

(e) Frame error check: It uses LRC in the case of ASCII mode or CRC error check method in the case of STU mode and consists of 2 bytes.

(f) Tail: It is applied only in the case of ASCII mode, and CRLF is added after LRC.

(g) Number of bytes: It means the number of bytes of data that responds.

(h) Data: It is sent in byte units with the address of request frame as the start address.

(i) Error code: It is represented by adding 80 (Hex) values to the function code and transmitted in 81(Hex) for the bit reading of output area.

(j) Exception code: It means the detailed error description and consists of 1 byte.

(3) Frame example

This is an example of reading bits from 20 to 38 in server station No. 1 that operates in Modbus RTU mode.

(a) Request frame

Division	Station No.	Function code	Address		Data size		Error check
			High-order byte	Low-order byte	High-order byte	Low-order byte	
Frame	01	01	00	13	00	13	CRC

(b) Response frame (when receiving a normal frame)

Division	Station No.	Function code	Number of bytes	Data			Error check
Frame	01	01	03	12	31	05	CRC

(c) Response frame (when receiving an abnormal frame)

Division	Station No.	Function code	Exception code	Error check
Frame	01	81	02	CRC

5) Reading data of bit type in bit input area (02)

(1) Bit reading of input area

The structures of request and response frames when reading the bit-type data of input area are as follows.
The tail of the frame is applied only in the case of ASCII mode.

(a) Request frame

Division	Station No.	Function code(02)	Address	Data size	Frame error check	Tail(CRLF)
Size(byte)	1	1	2	2	2	2

(b) Response frame (when receiving a normal frame)

Division	Station No.	Function code(02)	Number of bytes	Data	Frame error check	Tail(CRLF)
Size(byte)	1	1	1	N	2	2

(c) Response frame (when receiving an abnormal frame)

Division	Station No.	Error code	Exception code	Tail(CRLF)
Size(byte)	1	1	1	2

(2) Frame details

(a) Station number: It means the station number of the slave to read the bits of input area.
(b) Function code: '02' means bit continuous/individual reading of input area.
(c) Address: It means the start address of data to be read and consists of 2 bytes, when the start address conforms to the Modbus address rule.
(d) Data size: It is the size of data to read and consists of 2 bytes.
(e) Frame error check: It uses LRC in the case of ASCII mode or CRC error check method in the case of STU mode and consists of 2 bytes.
(f) Tail: It is applied only in the case of ASCII mode, and CRLF is added after LRC.
(g) Number of bytes: It means the number of bytes of data that responds.

(h) Data: It is sent in byte units with the address of request frame as the start address.
(i) Error code: It is represented by adding 80 (Hex) values to the function code and transmitted in 82(Hex) for the bit reading of output area.
(j) Exception code: It means the detailed error description and consists of 1 byte.

(3) Frame example
This is an example of reading bits from 20 to 28 in server station No. 1 that operates in Modubus RTU mode.

(a) Request frame

Division	Station No.	Function code	Address		Data size		Error check
			High-order byte	Low-order byte	High-order byte	Low-order byte	
Frame	01	02	00	13	00	13	CRC

(b) Response frame (when receiving a normal frame)

Division	Station No.	Function code	Number of bytes	Data			Error check
Frame	01	02	03	12	31	05	CRC

(c) Response frame (when receiving an abnormal frame)

Division	Station No.	Function code	Exception code	Error check
Frame	01	82	02	CRC

6) Reading data of word type in word output area (03)

(1) Word reading of output area
The structures of request and response frames when reading the word-type data of the output area are as follows. The tail of the frame is applied only in the case of ASCII mode.

(a) Request frame

Division	Station No.	Function code(03)	Address	Data size	Frame error check	Tail(CRLF)
Size(byte)	1	1	2	2	2	2

(b) Response frame (when receiving a normal frame)

Division	Station No.	Function code(03)	Number of bytes	Data	Frame check error	Tail(CRLF)
Size(byte)	1	1	1	N*2	2	2

(c) Response frame (when receiving an abnormal frame)

Division	Station No.	Error code	Exception code	Tail(CRLF)
Size(byte)	1	1	1	2

(2) Frame details
(a) Station number: It means the station number of the slave to read the word type data of output area.
(b) Function code: ‘03’ means word continuous/individual reading of input area.
(c) Address: It means the start address of data to be read and consists of 2 bytes, when the start address conforms to the Modbus address rule.
(d) Data size: It is the size of data to read and consists of 2 bytes.
(e) Frame error check: It uses LRC in the case of ASCII mode or CRC error check method in the case of STU mode and consists of 2 bytes.
(f) Tail: It is applied only in the case of ASCII mode, and CRLF is added after LRC.
(g) Number of bytes: It means the number of bytes of data that responds.

(h) Data: It is sent in byte units with the address of request frame as the start address, when the data is word type, so its size is the same as twice the number of bytes..
(i) Error code: It is represented by adding 80 (Hex) values to the function code and transmitted in 83(Hex) for the bit reading of output area.
(j) Exception code: It means the detailed error description and consists of 1 byte.

(3) Frame example
This is an example of reading the data of word type from 108 to 110 in server station No. 1 that operation in Modbus RTU mode.
(a) Request frame

Division	Station No.	Function code	Address		Data size		Error check
			High-order byte	Low-order byte	High-order byte	Low-order byte	
Frame	01	03	00	6B	00	03	CRC

(b) Response frame (when receiving a normal frame)

Division	Station No.	Function code	Number of bytes	Data						Error check
Frame	01	03	06	13	12	3D	12	40	4F	CRC

(c) Response frame (when receiving an abnormal frame)

Division	Station No.	Function code	Exception code	Error check
Frame	01	83	04	CRC

7) Reading data of word type in word input area (04)

(1) Word reading of input area
The structures of request and response frames when reading the word-type data of input area are as follows. The tail of the frame is applied only in the case of ASCII mode.
(a) Request frame

Division	Station No.	Function code(04)	Address	Data size	Frame error check	Tail(CRLF)
Size(byte)	1	1	2	2	2	2

(b) Response frame (when receiving a normal frame)

Division	Station No.	Function code(04)	Number of bytes	Data	Frame error check	Tail(CRLF)
Size(byte)	1	1	1	N*2	2	2

(c) Response frame (when receiving an abnormal frame)

Division	Station No.	Error code	Exception code	Tail(CRLF)
Size(byte)	1	1	1	2

(2) Frame details
(a) Station number: It means the station number of the slave to read the word type data of input area.
(b) Function code: '04' means word continuous/individual reading of input area.
(c) Address: It means the start address of data to be read and consists of 2 bytes, when the start address conforms to the Modbus address rule.
(d) Data size: It is the size of data to read and consists of 2 bytes.
(e) Frame error check: It uses LRC in the case of ASCII mode or CRC error check method in the case of STU mode and consists of 2 bytes.

(f) Tail: It is applied only in the case of ASCII mode, and CRLF is added after LRC.
(g) Number of bytes: It means the number of bytes of data that responds.
(h) Data: It is sent in byte units with the address of request frame as the start address, when the data is word type, so its size is the same as twice the number of bytes.
(i) Error code: It is represented by adding 80 (Hex) values to the function code and transmitted in 84(Hex) for the bit reading of output area.
(j) Exception code: It means the detailed error description and consists of 1 byte

(3) Frame example
This is an example of reading the data of word type stored in the input area No. 9 of server No. 1 that operates in Modbus RTU mode.

(a) Request frame

Division	Station No.	Function code	Address		Data size		Error check
			High-order byte	Low-order byte	High-order byte	Low-order byte	
Frame	01	04	00	08	00	01	CRC

(b) Response frame (when receiving a normal frame)

Division	Station No.	Function code	Number of bytes	Data		Error check
Frame	01	04	02	00	0A	CRC

(c) Response frame (when receiving an abnormal frame)

Division	Station No.	Function code	Exception code	Error check
Frame	01	84	04	CRC

8) Individual writing of bit-type data in bit output area (05)

(1) Bit individual writing of output area
The structures of request and response frames when writing data of bit type in output area are as follows. The tail of the frame is applied only in the case of ASCII mode.

(a) Request frame

Division	Station No.	Function code(05)	Address	Output value	Frame error check	Tail(CRLF)
Size(byte)	1	1	2	2	2	2

(b) Response frame (when receiving a normal frame)

Division	Station No.	Function code(05)	Address	Output value	Frame error check	Tail(CRLF)
Size(byte)	1	1	2	2	2	2

(c) Response frame (when receiving an abnormal frame)

Division	Station No.	Error code	Exception code	Tail(CRLF)
Size(byte)	1	1	1	2

(2) Frame details
(a) Station number: It means the station number of the slave to read the bit type data of input area.
(b) Function code: ‘05’ means bit continuous/individual reading of input area.
(c) Address: It means the start address of data to be written and consists of 2 bytes, when the start address conforms to the Modbus address rule.
(d) Output value: It is the bit value of address set in the address operates On, it is indicated by FF00(Hex), whereas if it operates Off, it is indicated by 0000(Hex).

(e) Frame error check: It uses LRC in the case of ASCII mode or CRC error check method in the case of STU mode and consists of 2 bytes.
(f) Tail: It is applied only in the case of ASCII mode, and CRLF is added after LRC.
(g) Number of bytes: It means the number of bytes of data that responds.
(h) Error code: It is represented by adding 80 (Hex) values to the function code and transmitted in 85(Hex) for the bit reading of output area.
(i) Exception code: It means the detailed error description and consists of 1 byte.

(3) Frame example
This is an example of turning the 9th bit of the output area On in server station No. 1 that operates Modbus RTU mode.
(a) Request frame

Division	Station No.	Function code	Address		Output value		Error check
			High-order byte	Low-order byte	High-order byte	Low-order byte	
Frame	01	05	00	08	FF	00	CRC

(b) Response frame (when receiving a normal frame)

Division	Station No.	Function code	Address		Output value		Error check
			High-order byte	Low-order byte	High-order byte	Low-order byte	
Frame	01	05	00	08	FF	00	CRC

(c) Response frame (when receiving an abnormal frame)

Division	Station No.	Function code	Exception code	Error check
Frame	01	85	04	CRC

9) Individual writing of word-type data in word output area (06)

(1) Word individual writing of output area
The structures of request and response frames when writing data of word type in output area are as follows.
The tail of the frame is applied only in the case of ASCII mode.
(a) Request frame

Division	Station No.	Function code(06)	Address	Output value	Frame error check	Tail(CRLF)
Size(byte)	1	1	2	2	2	2

(b) Response frame (when receiving a normal frame)

Division	Station No.	Function code(06)	Address	Output value	Frame error check	Tail(CRLF)
Size(byte)	1	1	2	2	2	2

(c) Response frame (when receiving an abnormal frame)

Division	Station No.	Error code	Exception code	Tail(CRLF)
Size(byte	1	1	1	2

(2) Frame details
(a) Station number: It means the station number of the slave to read the word type data of input area.
(b) Function code: '06' means word continuous/individual reading of input area.
(c) Address: It means the start address of data to be written and consists of 2 bytes, when the start address conforms to the Modbus address rule.
(d) Output value: It means the data value to be written in address set in the address.
(e) Frame error check: It uses LRC in the case of ASCII mode or CRC error check method in the case of

STU mode and consists of 2 bytes.
(f) Tail: It is applied only in the case of ASCII mode, and CRLF is added after LRC.
(g) Number of bytes: It means the number of bytes of data that responds.
(h) Error code: It is represented by adding 80 (Hex) values to the function code and transmitted in 86(Hex) for the bit reading of output area.
(i) Exception code: It means the detailed error description and consists of 1 byte.

(3) Frame example
This is an example of writing 0003(hex) in the 9th output area of word type in server station No. 1 that operates in Modbus RTU mode.
(a) Request frame

Division	Station No.	Function code	Address		Output value		Error check
			High-order byte	Low-order byte	High-order byte	Low-order byte	
Frame	01	06	00	08	00	03	CRC

(b) Response frame (when receiving a normal frame)

Division	Station No.	Function code	Address		Output value		Error check
			High-order byte	Low-order byte	High-order byte	Low-order byte	
Frame	01	06	00	08	00	03	CRC

(c) Response frame (when receiving an abnormal frame)

Division	Station No.	Function code	Exception code	Error check
Frame	01	86	02	CRC

10) Continuous writing of bit-type data in bit output area (0F)

(1) Bit continuous writing of output area
The structures of request and response frames for continuous writing of bit-type data in output area are as follows. The tail of the frame is applied only in the case of ASCII mode.
(a) Request frame

Division	Station No.	Function code (0F)	Address	Number of outputs	Data size	Output value	Frame error check	Tail(CRLF)
Size(byte)	1	1	2	2	1	N	2	2

(b) Response frame (when receiving a normal frame)

Division	Station No.	Function code(0F)	Address	Number of outputs	Frame error check	Tail(CRLF)
Size(byte)	1	1	2	2	2	2

(c) Response frame (when receiving an abnormal frame)

Division	Station No.	Error code	Exception code	Tail(CRLF)
Size(byte)	1	1	1	2

(2) Frame details
(a) Station number: It means the station number of the slave to read the bit type data of input area.
(b) Function code: ‘0F’ means bit continuous/individual reading of input area.
(c) Address: It means the start address of data to be written and consists of 2 bytes, when the start address conforms to the Modbus address rule.

(d) Number of outputs: It means the number of data to be written and consist of 2 bytes. Example) If the address writes 10 data consecutively from No. 20, the number of outputs is 00A (Hex).
(e) Data size: The number of outputs is represented by a byte value. That is, if the data size is 1, the number of continuous write data is 8. Example) When 10 consecutive bits of data is written, the data size is 2.
(f) Output value: It means the data value to be written in address set in the address.
(g) Frame error check: It uses LRC in the case of ASCII mode or CRC error check method in the case of STU mode and consists of 2 bytes.
(h) Tail: It is applied only in the case of ASCII mode, and CRLF is added after LRC.
(i) Number of bytes: It means the number of bytes that responds.
(j) Error code: It is represented by adding 80 (Hex) values to the function code and transmitted in 8F(Hex) for the bit reading of output area.
(k) Exception code: It means the detailed error description and consists of 1 byte.

(3) Frame example
This is an example of writing 10 consecutive bit values from the 20th address in the server No. 1 that operates in Modbus RTU mode.

Example) Data value to be continuously written

Bit value	1	1	0	0	1	1	0	1	0	0	0	0	0	0	0	1
Hex	C				D				0				1			
Address	27	26	25	24	23	22	21	20	-	-	-	-	-	-	29	28

(a) Request frame

Division	Statio n No.	Function code	Address		Number of outputs		Data size	Output value		Error check
			High-order byte	Low-order byte	High-order byte	Low-order byte		High-order byte	Low-order byte	
Frame	01	0F	00	13	00	0A	02	CD	01	CRC

(b) Response frame (when receiving a normal frame)

Division	Station No.	Function code	Address		Number of outputs		Error check
			High-order byte	Low-order byte	High-order byte	Low-order byte	
Frame	01	04	00	13	00	0A	CRC

(c) Response frame (when receiving an abnormal frame)

Division	Station No.	Function code	Exception code	Error check
Frame	01	8F	01	CRC

11) Continuous writing of word-type data in word output area (10)

(1) Word continuous writing of output area
The structures of request and response frames for continuous writing of word-type data in output area are as follows. The tail of the frame is applied only in the case of ASCII mode.

(a) Request frame

Division	Station No.	Function code(10)	Address	Number of outputs	Data size	Output value	Frame error check	Tail(CRLF)
Size(byte)	1	1	2	2	1	N*2	2	2

(b) Response frame (when receiving a normal frame)

Division	Station No.	Function code(10)	Address	Number of outputs	Frame error check	Tail(CRLF)
Size(byte)	1	1	2	2	2	2

(c) Response frame (when receiving an abnormal frame)

Division	Station No.	Error code	Exception code	Tail(CRLF)
Size(byte)	1	1	1	2

(2) Frame details

(a) Station number: It means the station number of the slave to read the bit type data of input area.
(b) Function code: '10' means bit continuous/individual reading of input area.
(c) Address: It means the start address of data to be written and consists of 2 bytes, when the start address conforms to the Modbus address rule.
(d) Number of outputs: It means the number of data to be written and consist of 2 bytes. Example) If the address writes 10 data consecutively from No. 20, the number of outputs is 00A (Hex).
(e) Data size: The number of outputs is represented by a byte value. That is, if the data size is 1, the number of continuous write data is 8.
Example) When 10 consecutive bits of data is written, the data size is 2.
(f) Output value: It means the data value to be written in address set in the address.
(g) Frame error check: It uses LRC in the case of ASCII mode or CRC error check method in the case of STU mode and consists of 2 bytes.
(h) Tail: It is applied only in the case of ASCII mode, and CRLF is added after LRC.
(i) Number of bytes: It means the number of bytes that responds
(j) Error code: It is represented by adding 80 (Hex) values to the function code and transmitted in 90(Hex) for the bit reading of output area.
(k) Exception code: It means the detailed error description and consists of 1 byte.

(3) Frame example

This is an example of writing 2 consecutive words from the 20th address in the server station No. 1 that operates in Modbus RTU mode.

Example) Data value to be continuously written

Hex	C	D	0	1	0	0	0	A
Address	20				21			

(a) Request frame

Division	Station No.	Function code	Address		Number of outputs		Data size	Output value				Error check
			High-order bit	Low-order bit	High-order byte	Low-order byte						
Frame	01	10	00	13	00	02	04	CD	01	00	0A	CRC

(b) Response frame (when receiving a normal frame)

Division	Station No.	Function code	Address		Number of outputs		Error check
			High-order byte	Low-order byte	High-order byte	Low-order byte	
Frame	01	10	00	13	00	02	CRC

(c) Response frame (when receiving an abnormal frame)

Division	Station No	Function code	Exception code	Error check
Frame	01	90	01	CRC

15.10.4 Modbus server

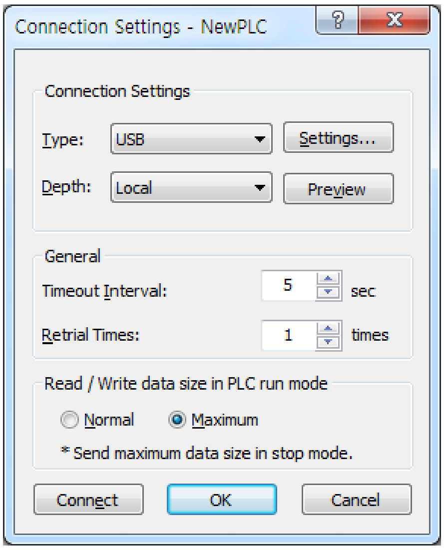
This is used when the external device that is trying to communicate operates as a Modbus client. It supports both ASCII mode and RTU mode of Modbus, each operation mode can be set in [Basic setting] window.

1) How to use a Modbus ASCII Server

(1) Connection setup

(a) Select [Online] → [Connection Settings] or click the icon ().

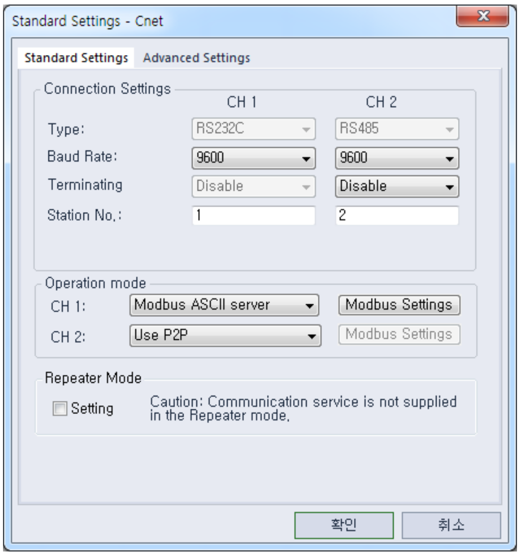
(b) Set connection options for your environment and click [Connect].

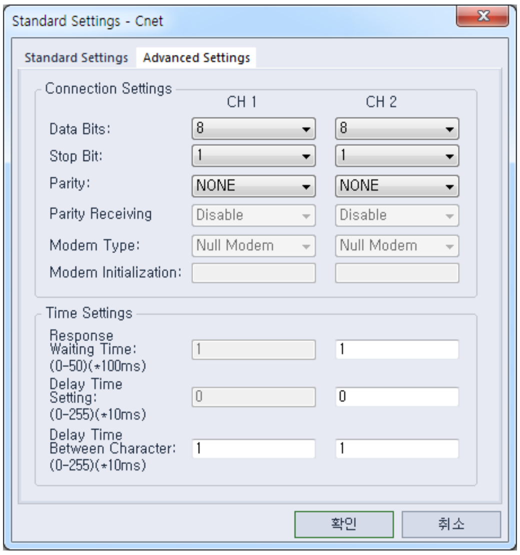


(2) Default setting

(a) Double-click the serial communication module in the project window to launch the [Preferences] window and set the connection. Set the communication type, communication speed, data bit, stop bit, and station number in the menu.

(b) The delay time can be set only when RS-485 is used, and the response wait time can be set only when using P2P as the operation mode in RS-485 communication.





(3) Select operating mode

Select the Modbus ASCII server

(4) Modbus settings

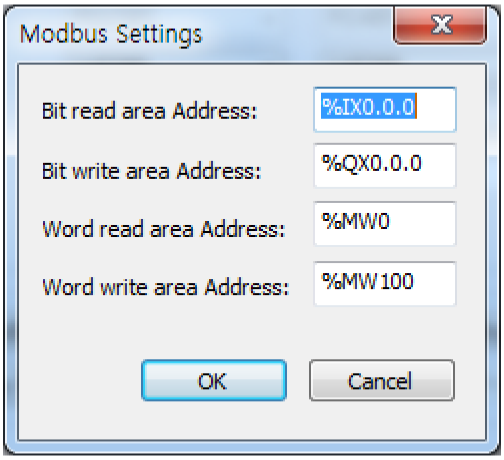
(a) When the Modbus ASCII server is selected as the operation mode, [Modbus setting] becomes active.

(b) Bit Read Area Start Address: Indicates the start address of the bit read area and consists of 5 digits.
The first four digits represent the word value, and the remaining digits represent the bit value.
Ex) IX0.0.0: I Device area 0-th bit of 0th word is set as start address of bit read area.

(c) Bit Write Area Start Address: Indicates the start address of the bit write area and consists of 5 digits.
The first four digits represent the word value, and the remaining digits represent the bit value.
Ex) QX0.0.0: The 0th bit of the tenth word of the Q device area is set as the start address of the bit read area.

(d) Word read area start address: It indicates the start address of the word read area and consists of 4 digits.
Ex) The 0th word of MW0: M device area is set as the start address of the word read area.

(e) Word write area start address: It is the start address of the word write area and consists of 4 digits.
Ex) MW100: It is the case that the 100th word of the M device area is set as the start address of the word write area.

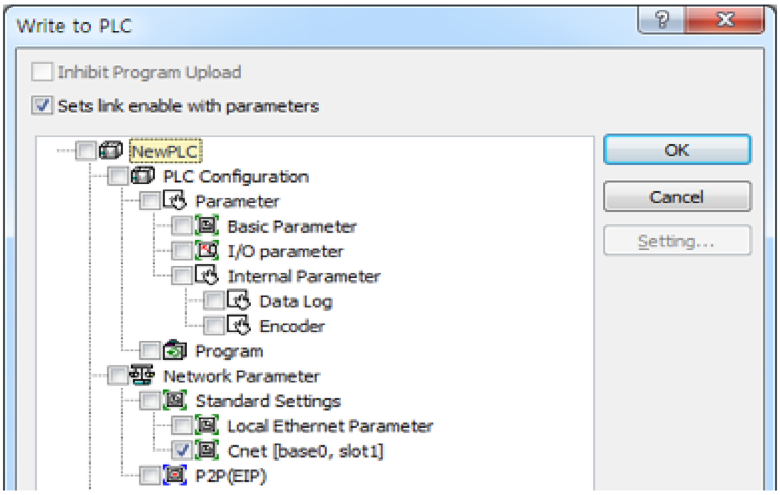


(5) Write parameters

(a) Select [Online] → [Write] or click the icon ( ).

(b) Check (✓) the module with the default settings and click [OK].

(c) Click the [OK] button, and when the parameter writing finishes, reset each module.

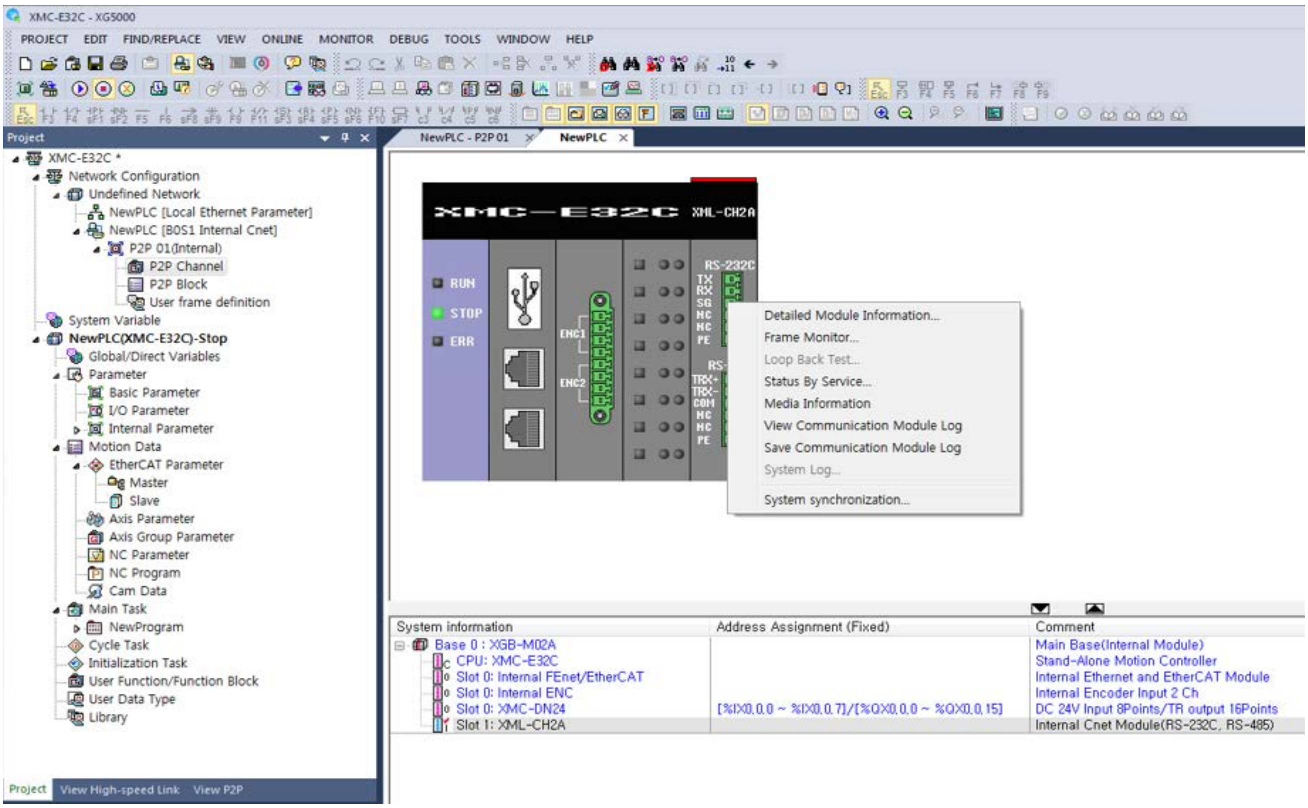


(6) Check operation

(a) Select [Online] → [Communication Module Settings] → [System Diagnosis] or click the icon ( ).

(b) Click the communication module whose status you want to diagnose and press the right mouse button.

(c) When the following screen appears, click [Frame Monitor] or [Service Status] to check the operation status.

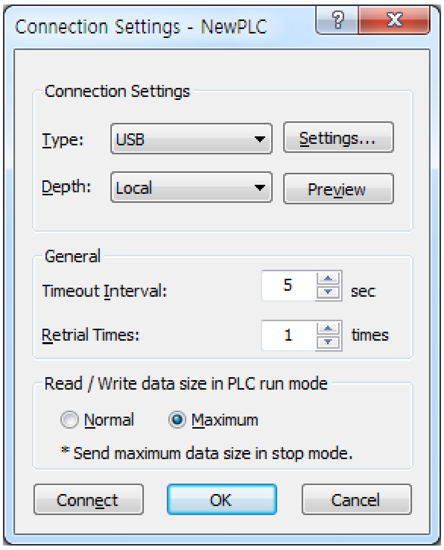


2) How to use a Modbus RTU Server

(1) Connection setup

(a) Select [Online] → [Connection Settings] or click the icon ().

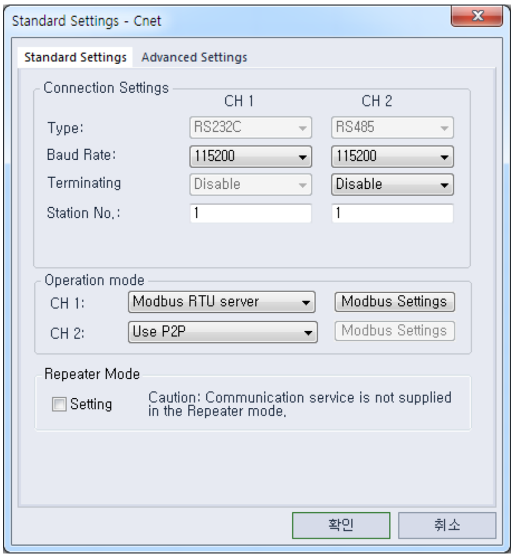
(b) Set connection options for your environment and click [Connect].

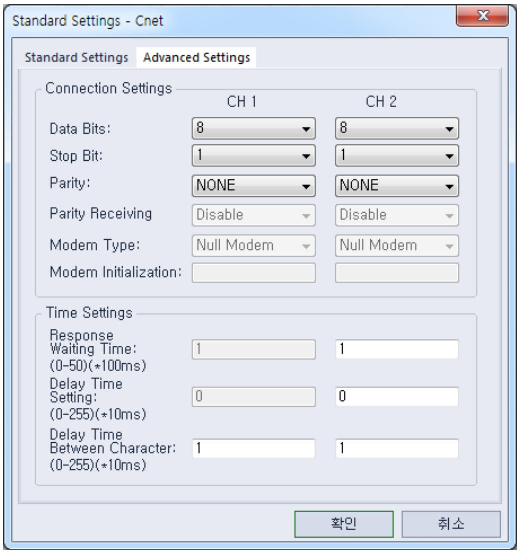


(2) Default setting

(a) Double-click the serial communication module in the project window to launch the [Preferences] window and set the connection. Set the communication type, communication speed, data bit, stop bit, and station number in the menu.

(b) The delay time can be set only when RS-485 is used, and the response wait time can be set only when using P2P as the operation mode in RS-485 communication.





(3) Select operating mode

Select the Modbus ASCII server

(4) Modbus settings

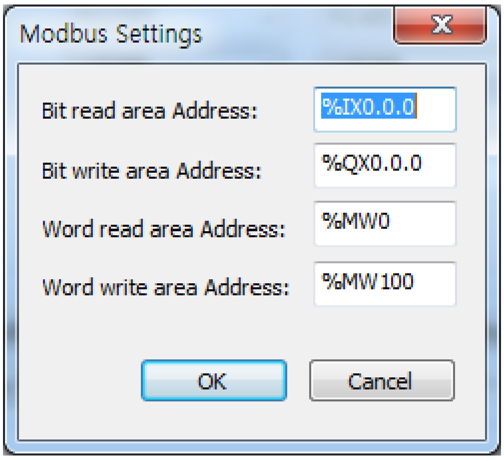
(a) When the Modbus ASCII server is selected as the operation mode, [Modbus setting] becomes active.

(b) Bit Read Area Start Address: Indicates the start address of the bit read area and consists of 5 digits.
The first four digits represent the word value, and the remaining digits represent the bit value.
Ex) IX0.0.0: I Device area 0-th bit of 0th word is set as start address of bit read area.

(c) Bit Write Area Start Address: Indicates the start address of the bit write area and consists of 5 digits.
The first four digits represent the word value, and the remaining digits represent the bit value.
Ex) QX0.0.0: The 0th bit of the tenth word of the Q device area is set as the start address of the bit read area.

(d) Word read area start address: It indicates the start address of the word read area and consists of 4 digits.
Ex) The 0th word of MW0: M device area is set as the start address of the word read area.

(e) Word write area start address: It is the start address of the word write area and consists of 4 digits.
Ex) MW100: It is the case that the 100th word of the M device area is set as the start address of the word write area.

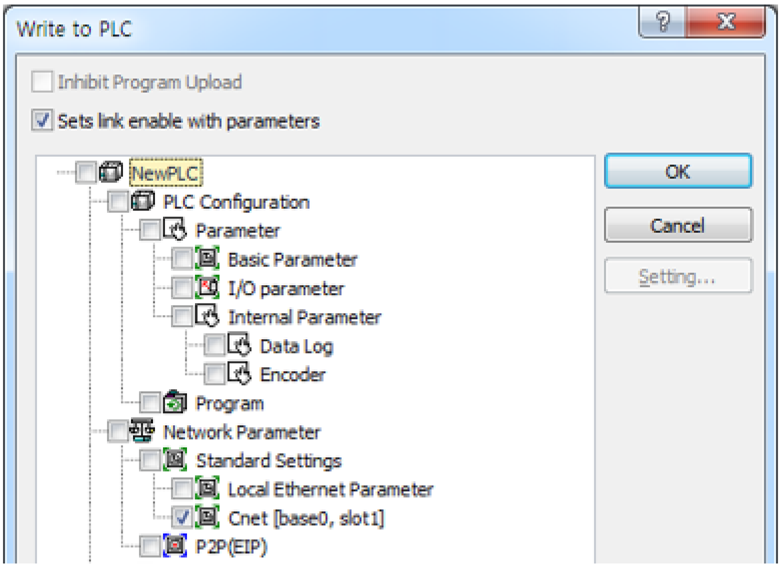


(5) Write parameters

(d) Select [Online] → [Write] or click the icon ( ).

(e) Check (✓) the module with the default settings and click [OK].

(f) Click the [OK] button, and when the parameter writing finishes, reset each module.

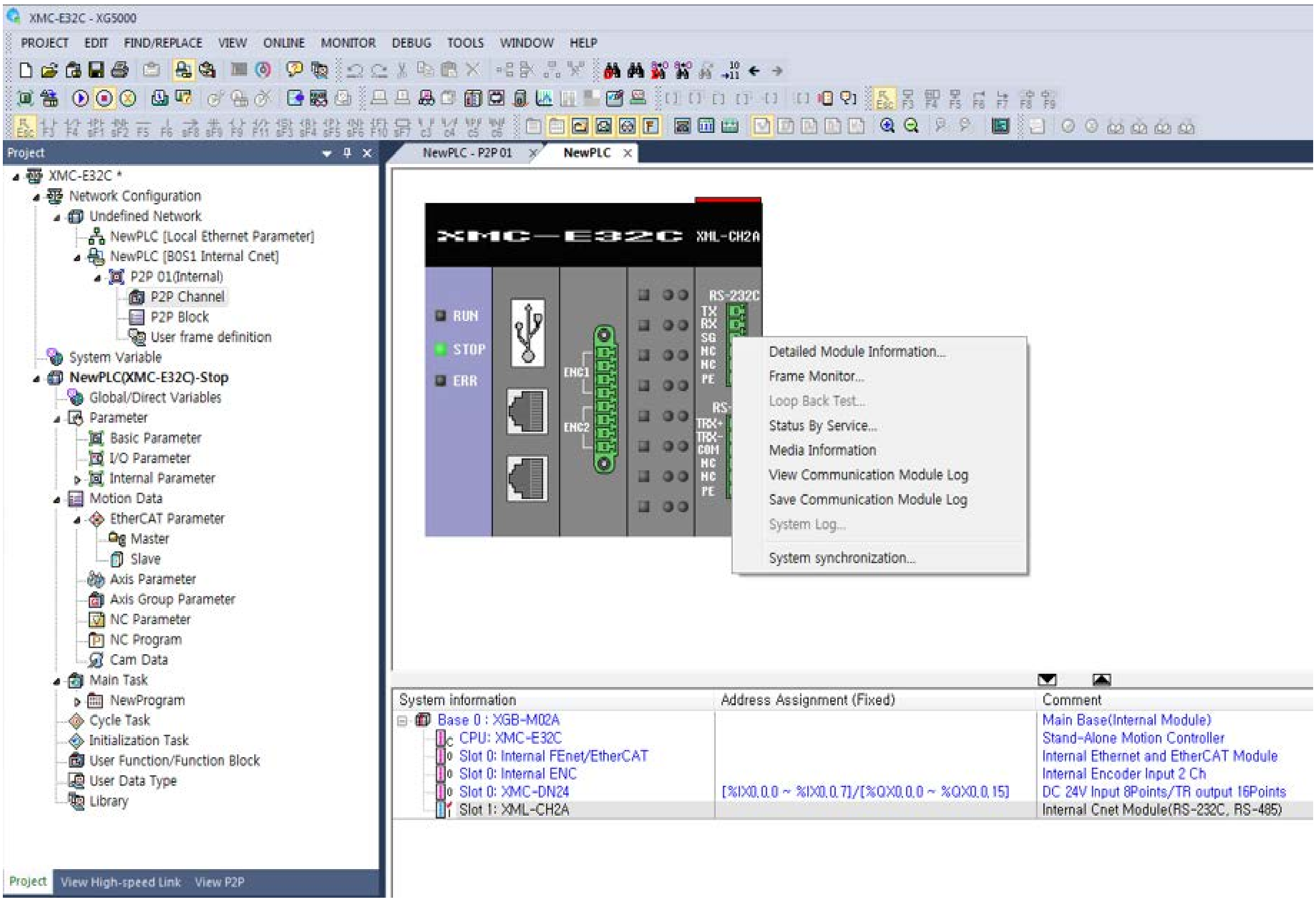


(6) Check operation

(a) Select [Online] → [Communication Module Settings] → [System Diagnosis] or click the icon ( ).

(b) Click the communication module whose status you want to diagnose and press the right mouse button.

(c) When the following screen appears, click [Frame Monitor] or [Service Status] to check the operation status.



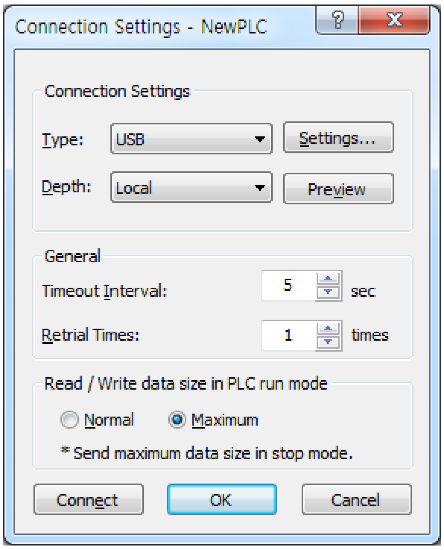
15.10.5 Modbus RTU / ASCII client

1) Default setting

(1) Connection setup

(a) Select [Online] → [Connection Settings] or click the icon ( ).

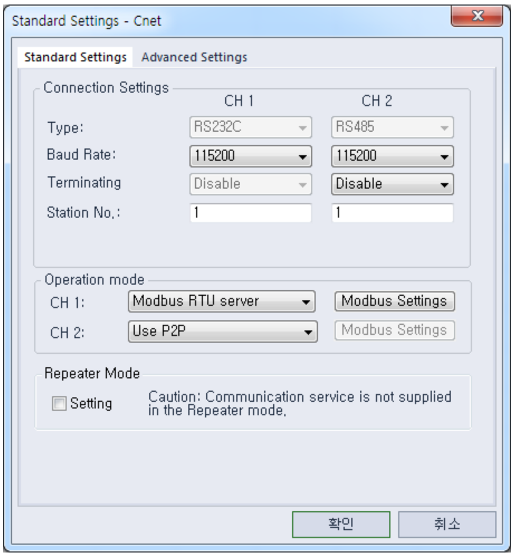
(b) Set connection options for your environment and click [Connect].

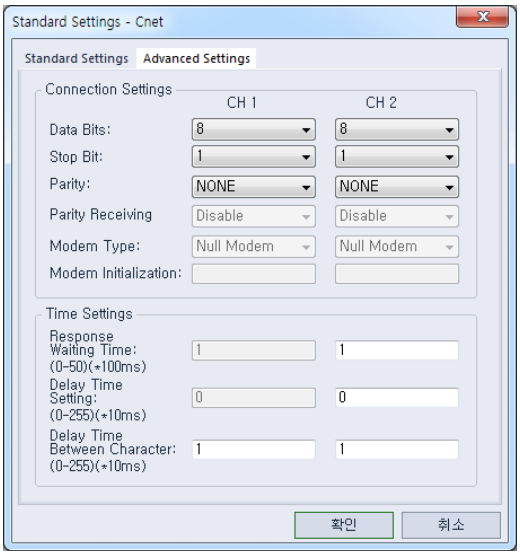


(2) Default setting

(a) Double-click the serial communication module in the project window to launch the [Preferences] window and set the connection. Set the communication type, communication speed, data bit, stop bit, and station number in the menu.

(b) The delay time can be set only when RS-485 is used, and the response wait time can be set only when using P2P as the operation mode in RS-485 communication.





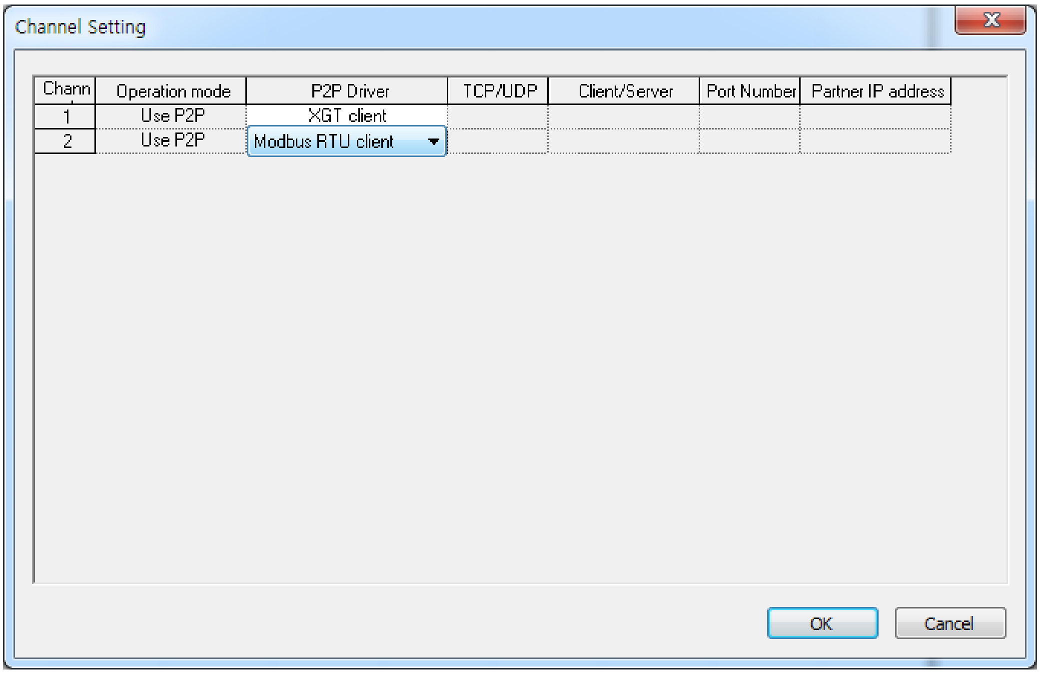
(3) Select operating mode

When using as a client, be sure to select 'Use P2P'.

(4) P2P channel setting

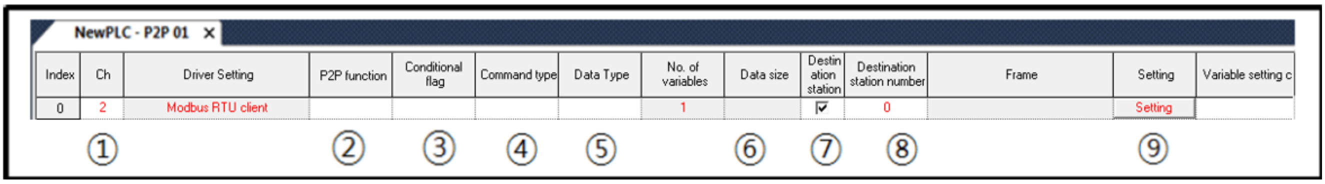
(a) Double-click the P2P channel to select the protocol for each channel.

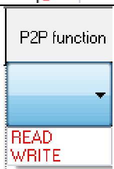
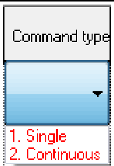
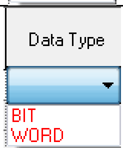
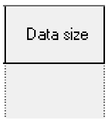
(b) The P2P driver supports user frame definition, XGT client, LS bus client, and Modbus RTU / ASCII client.

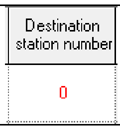
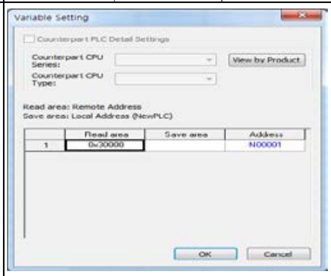
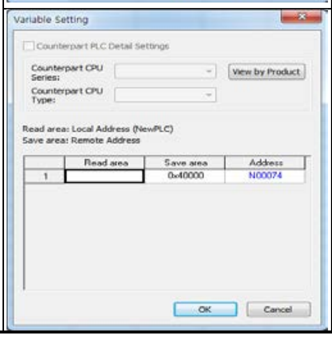


2) P2P Parameter setting

Operation in the Modbus RTU/ASCII client is divided into the Read command used to read and store the arbitrary area of the other station and the Write command written in the arbitrary area of the other station. The setting method of Modbus RTU client and Modbus ASCII client is the same as shown below.



Number	Classification	Block type	Meaning
1	Channel		The set drive name changes according to the driver set in P2P driver
2	P2P function		1. Read: Used to read arbitrary data from the other station 2. Write: Used to write arbitrary data to the other station
3	Start condition		1. Enter special flag or bit contact to select the time when data is transmitted and received 2. Example: _T20MS(cycle: 20ms), %MX0(when MX0 is On)
4	Mode		1.Individual: Used to read or write data of up to 4 memory areas to the other station(Example: M01, M10, M20, M30) 2.Continuous: Used to read or write continuous data to the other station (Example: M01~M10)
5	Data type		Data types can be selected among bits and words
6	Data size		▷It defines data size to be transmitted and received and is enabled only in continuous mode. 1. If the P2P function is Read 1) Modbus RTU client (1)Bit type: 1~2000 (2)Word type: 1~120 2) Modbus ASCII client (1)Bit type: 1~976 (2)Word type: 1~61 2. If the P2P function is Write 1) Modbus RTU client (1)Bit type: 1~1968 (2)Word type: 1~120 2) Modbus ASCII client (1)Bit type: 1~944 (2)Word type: 1~120

<table>
<tr><th>Number</th><th>Classification</th><th>Block type</th><th>Meaning</th></tr>
<tr><td>7</td><td>Partner station</td><td></td><td>It is checked automatically. If you do not use the block, click it once more, otherwise the block will not work.</td></tr>
<tr><td>8</td><td>Partner station number</td><td></td><td>1. It means the station number of the partner station.
1) Modbus: A total of 256 station numbers can be set from 0 to 255.</td></tr>
<tr><td rowspan="2">9</td><td rowspan="2">Settings</td><td></td><td>▶If the P2P function is Read
1.Read area: Data area start address of the partner station(server)
1)Bit: Bit input(0x10000), bit output(0x00000)
2)Word: Word input(0x30000), word output(0x40000)
2.Storage area: Data storage area of its station(client)</td></tr>
<tr><td></td><td>▶If the P2P function is Write
1.Read are: Data area of its station
2.Storage area: Data storage area start address of the other station
1)Bit: Bit input(0x10000), bit output(0x00000)
2)Word: Word input(0x30000), word output(0x40000)</td></tr>
</table>

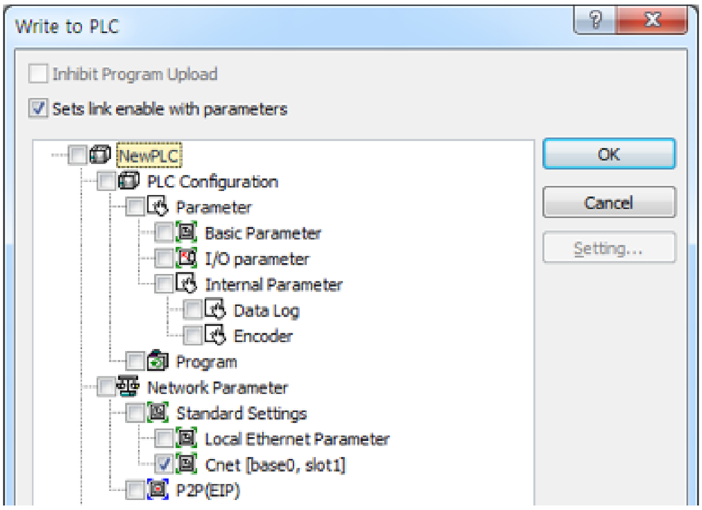
3) Writing P2P parameters

(1) Writing P2P parameters

(a) Select [Online] → [Write] or click the icon ( ).

(b) Check (✓) the module with the default settings and click [OK].

(c) Click the [OK] button, and when the parameter writing finishes, reset each module.



(2) Link Enable

(a) Select [Online] → [Communication module settings] → [Link enable].

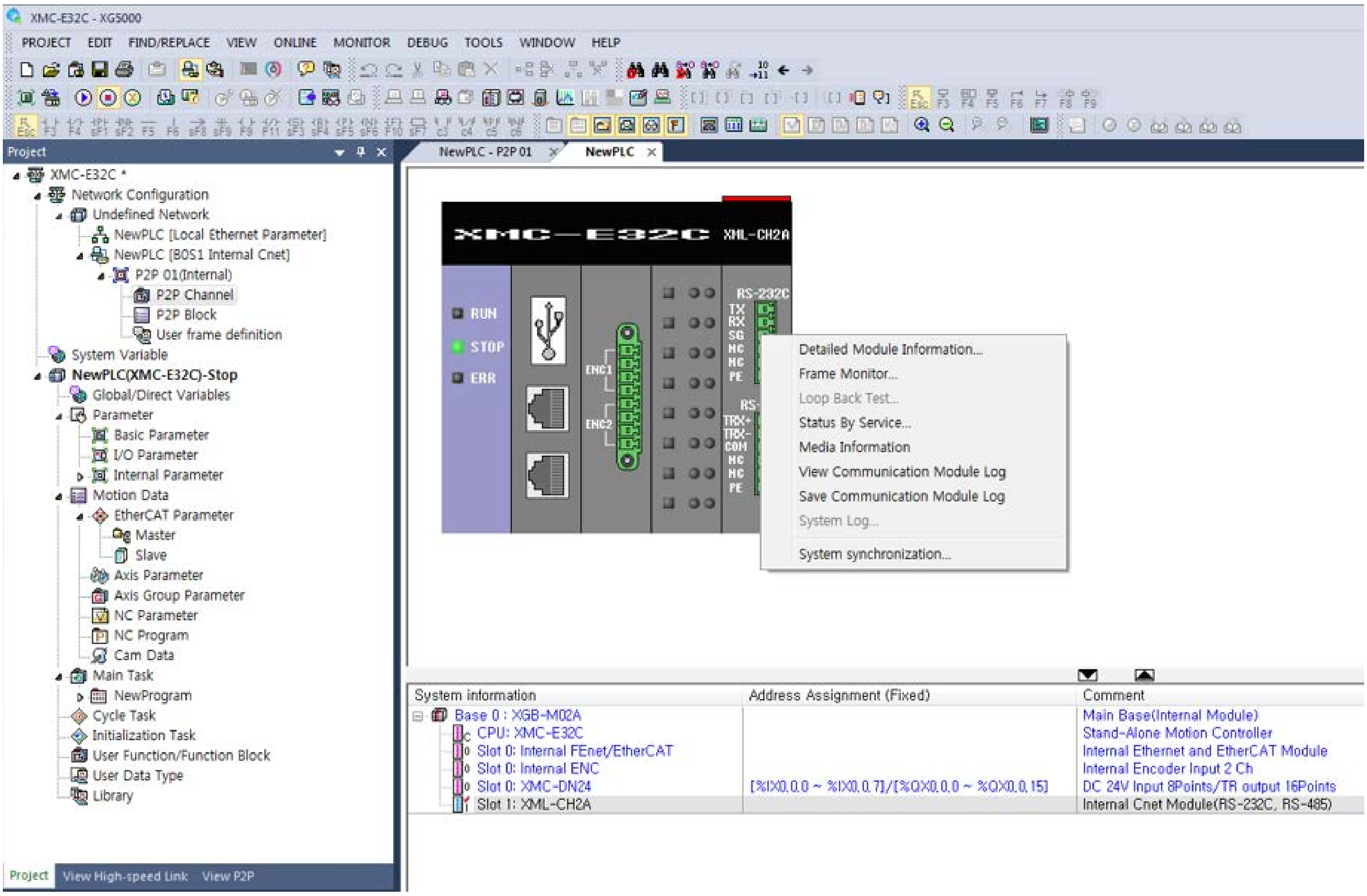
(b) Check the set P2P block and click Write.

(3) Check operation

(a) Select [Online] → [Communication Module Settings] → [System Diagnosis] or click the icon ().

(b) Click the communication module whose status you want to diagnose and press the right mouse button.

(c) When the following screen appears, click [Frame Monitor] or [Service Status] to check the operation status.



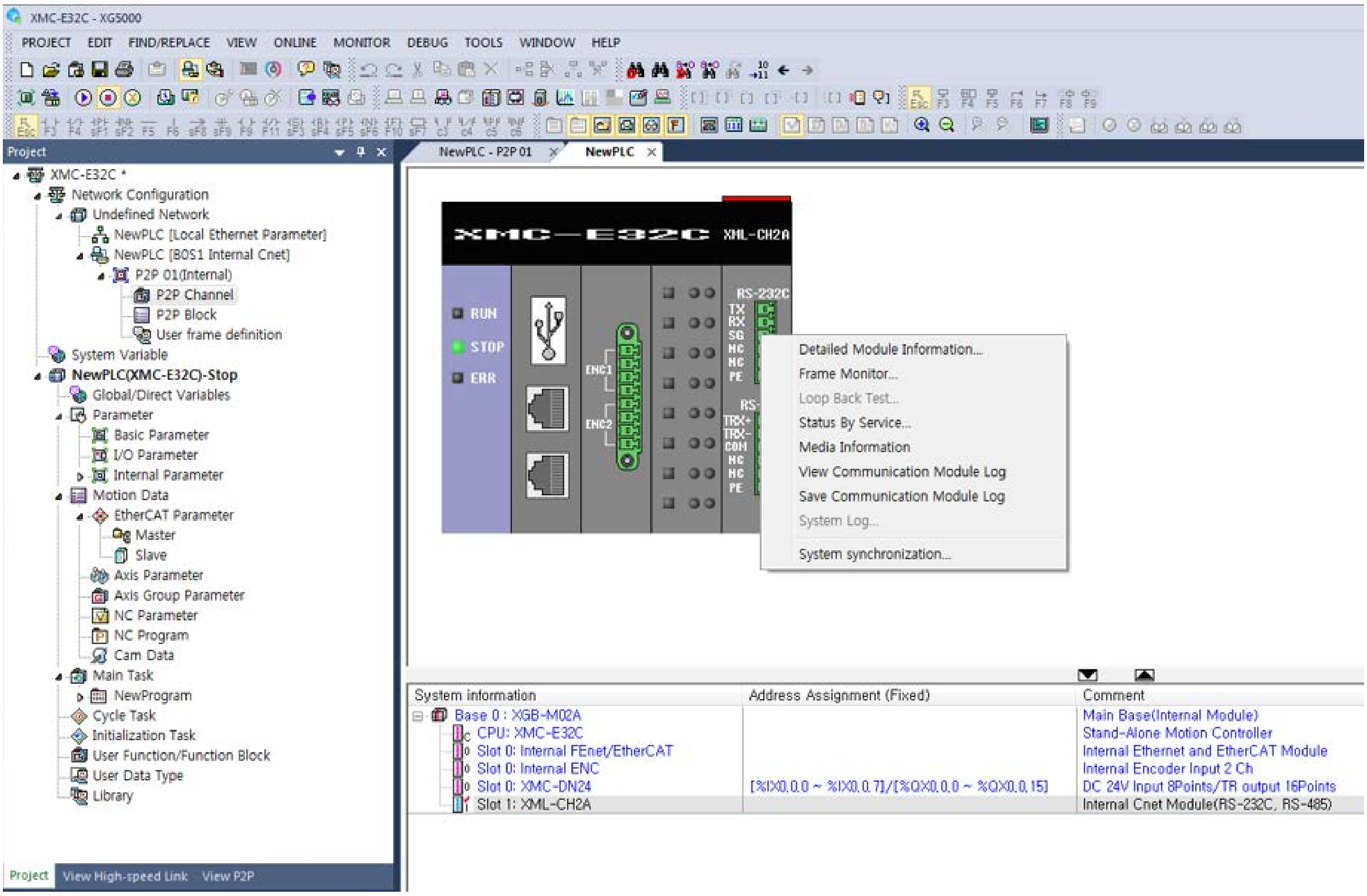
15.10.6 Frame monitor

With the frame monitor function of XG5000, you can check the frame that client and server actually send and receive.

(1) Check operation

(a) Select [Online] → [Communication Module Settings] → [System Diagnosis] or click the icon ( ).

(b) Click the communication module whose status you want to diagnose and press the right mouse button.



(c) When the following screen appears, click [Frame Monitor] or [Service Status] to check the operation status.

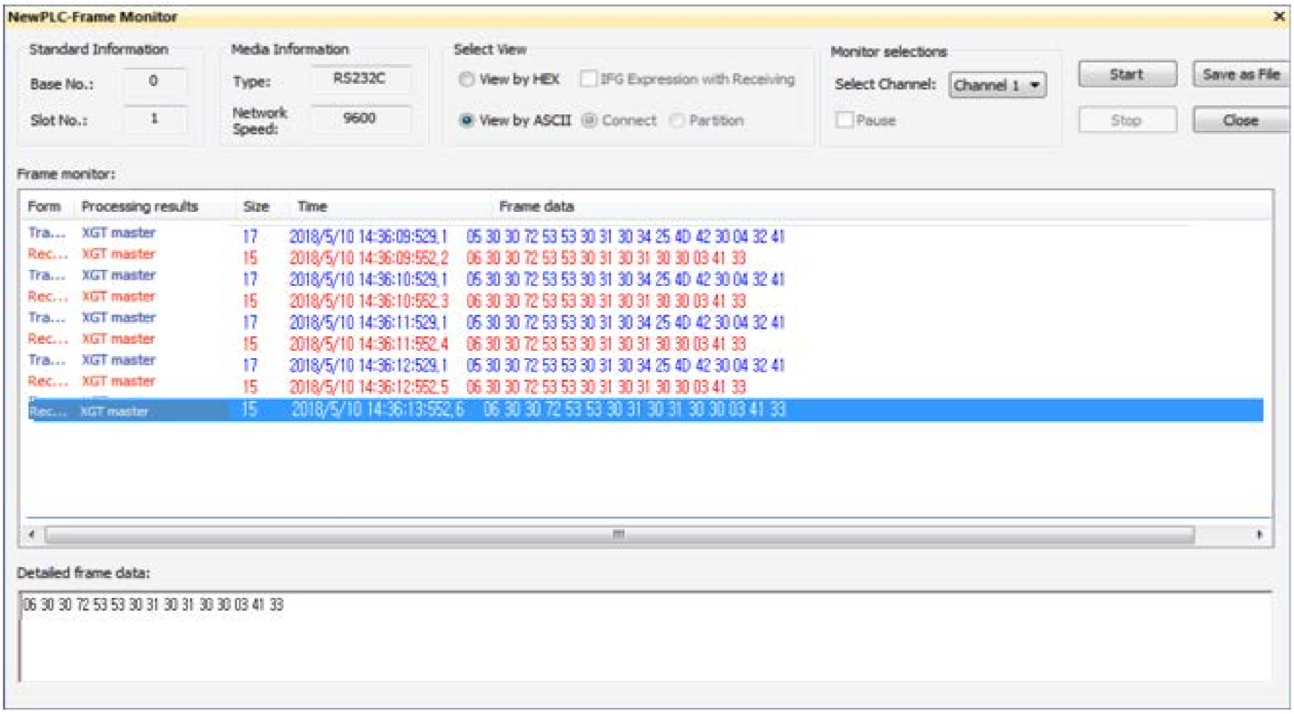
(2) Frame monitor

(a) Select the channel you want to monitor.

(b) When the protocol is in Modbus ASCII mode, select View as ASCII.

(c) Select Hex View when the protocol is in Modbus RTU mode.

(d) Click [Start] to check the sending / receiving frame.

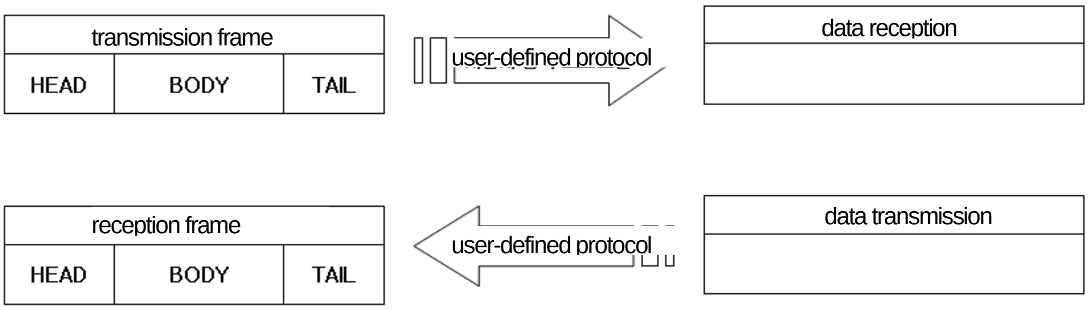


Items		Contents	Remarks
Basic information	Base No.	Base position of the communication module being monitored	
	Slot No.	Slot position of the communication module being monitored	
Monitor option	Channel selection	Select the channel to monitor	
Frame monitor window	Type	Indicate the transmission frame and reception frame	
	Processed results	Indicate the protocol type currently being used 1) XGT server 2) XGT client 3) Modbus server 4) Modbus client 5) User defined 6) Unknown: Frame that cannot be processed	
	Size	Length of the monitored frame	
	Time	Display the point of time for transmission/reception	
	Frame data	Display the data of transmitted/received frame	
View in HEX		Display the frame data with HEX values	
View in ASCII		Display the frame data with ASCII values	
Save file		Save the frame monitoring contents to a file	Saved in CSV format
Start		Start of the frame monitoring operation	
Stop		Stop the monitoring status	
Close		Close the frame monitor window	

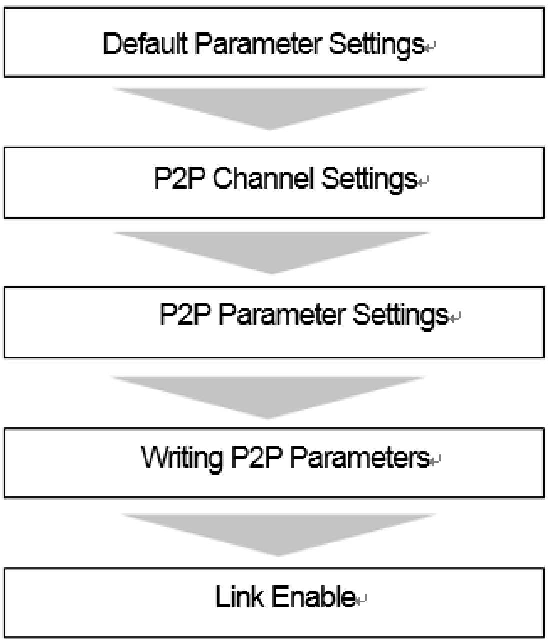
15.11 User Defined Communication Service

15.11.1 Summary

Because there are many kinds of communication protocols, it is actually impossible to mount all protocols in one communication module. Therefore, to solve this problem, XMC built-in Cnet provides user frame definition communication function. When this function is used, when connecting with other devices that do not use XGT protocol or Modbus protocol, users can create and communicate the protocol for their own use. At this time, the user must send frame and receive frame in the same way as the protocol of the external device to send / receive data.



Follow the procedure below when using user frame definition communication.



15.11.2 User-defined frame configuration

When a frame is created using the user-defined communication, the frame is largely divided into a head which indicates the beginning of the frame, a tail indicating the end, and a body, which is a data area. The head, tail and body are composed of each segment. In addition, the total size of one frame should be less than 1024 bytes.

Frame		
HEAD	BODY	TAIL
Segment 1	Segment 1	Segment 1
Segment 2	Segment 2	Segment 2
Segment 3	Segment 3	Segment 3
Segment N	Segment N	Segment N

(1) Head configuration

The input types of the segments composed of the head are largely divided into numerical constants and string constants. The numerical constant is expressed as a hexadecimal value, and the string constant means an ASCII character.

(2) Tail configuration

The input types of the segments that consist of the tail include numerical constants, string constants and BCC to check frame errors. The meaning of the numerical constant and string constant is the same as that used in the head. The BCC is a segment used to check errors in the transmission and reception frames, and only one can be set in the tail.

a) BCC error check

The meaning of the numerical constant and string constant is the same as that used in the head. The BCC is a segment used to check errors in the transmission and reception frames, and only one can be set in the tail. If BCC is set, BCC operation is performed for the transmission/reception frame according to the setting method. If the operation results are different, the corresponding frame is ignored. In this way, it serves as a means for improving the reliability of communication. The error checking methods of each BCC are shown below.

Classification	BCC method	Descriptions
General-purpose communication error detection method	Byte SUM	Use the low-order byte value of the result obtained by adding the data of the specified area in 1-byte unit
	Word SUM	Use the low-order word value of the result obtained by adding the data of the specified area in 1-word unit
	Byte XOR	Use the low-order byte of Exclusive OR result of the data of the designated area in 1-byte unit
	7 bit SUM	Use values except for the most significant bit of the byte SUM result value
	7 bit XOR	Use values except for the most significant bit of the byte XOR result value
	7 bit SUM#1	Add 20_H if the 7bit SUM result value is less than 20_H
	Byte SUM 2'S COMP	Take 2's complement with respect to the byte SUM result
	Byte SUM 1'S COMP	Take 1's complement with respect to the byte SUM result
	CRC 16	16 bit CRC error detection method
	CRC 16 IBM	16 bit IBM CRC error detection method
	CRC 16 CCITT	16 bit CCITT CRC error detection method
	MODBUS LRC	MODBUS LRC error detection method
Dedicated communication error detection method	LGIS CRC	Error detection method used in LGIS PLC
	DLE AB	Error detection method of Allen Bradley's DF1 Protocol
	DLE SIEMENS	Error detection method used in Siemens 3964R communication

In BCC settings, if BCC calculation method is classified as a dedicated communication method, there is no need to set BCC setting range and display method. On the other hand, it is classified as a general-purpose communication method, BCC setting range and display method should be set.

<table>
<tr><th colspan="2">Item</th><th>Contents</th></tr>
<tr><td rowspan="2">Start position</td><td>Area</td><td>Specify where to start BCC calculation from head/body/tail</td></tr>
<tr><td>Segment</td><td>Specify the segment position to start BCC calculation within the head/body/tail. 0 is included in the BCC calculation from the beginning of the frame</td></tr>
<tr><td rowspan="3">End position</td><td>Before BCC</td><td>Included in the calculation from start position to before BCC</td></tr>
<tr><td>End of are</td><td>Included in the calculation from start position to the end of the specified area</td></tr>
<tr><td>Setting</td><td>Included in the calculation from start position to the specified area segment position</td></tr>
<tr><td colspan="2">ASCII conversion</td><td>Convert result values to ASCII characters, the size increases two times.</td></tr>
<tr><td colspan="2">Initial value 0</td><td>The initial value of BCC calculation can be set to 0. If not specified, FF_H becomes the initial value.</td></tr>
</table>

(3) Body configuration

The input types of segments that make up the body vary depending on the transmission and reception. In the case of transmission, numerical and string constants are classified as variable size parameters, and the meaning of the numerical constant and the string constant is the same as that of the head.

a) Variable size parameters (in the case of reception frame)

The part of the frame that changes in size and content is defined as a variable size parameter segment. The variable size parameter can be set only for the body. In addition, unlike other segments, the additional segment cannot be set after the variable size parameter segment. When the variable size segment is used, there should be head or tail. If you register a frame only with the variable size parameter without head or tail, there is a possibility of error when receiving the frame depending on communication situation. Be sure to set head or tail for reliable communication. Even if the variable size parameter is used in the transmission frame, the function and characteristics are the same as those of the fixed size parameter of the reception because the size of the variable size parameter is specified in the P2P block settings.

b) Fixed size parameters (in the case of reception frame)

The part that changes in reception although the size is fixed is defined as a fixed size parameter segment. It can be set only in the body. In the case of the fixed size parameter, up to 4 parameters can be set in one body.

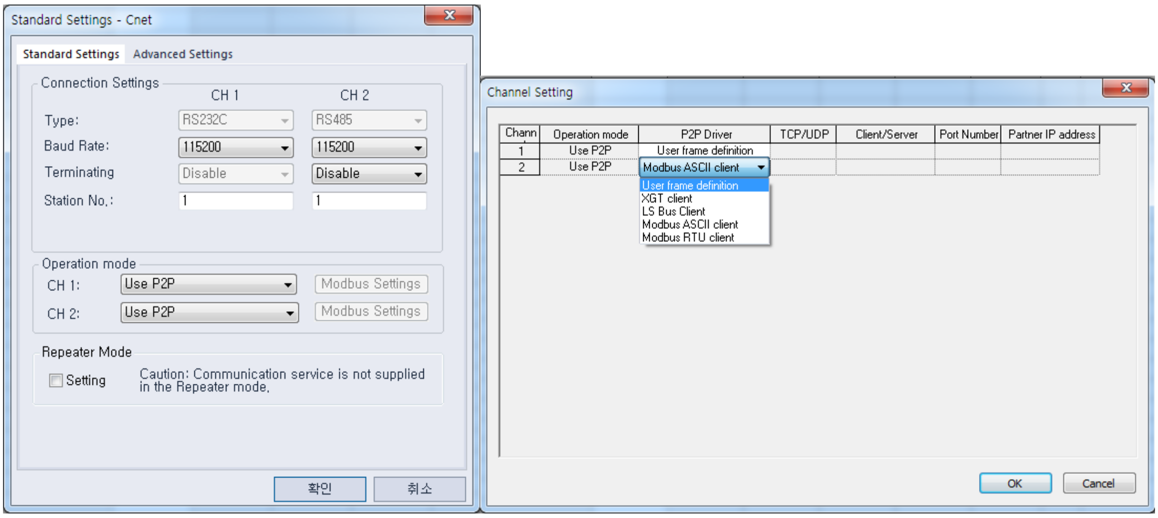
The specifications of the transmission and reception frames supported in the user-defined communication of Cnet I/F are summarized below.

<table>
<tr><th>Group</th><th>Frame</th><th>Segment</th><th>Remarks</th></tr>
<tr><td rowspan="8">Transmission frame</td><td rowspan="2">HEAD</td><td>Numerical constant</td><td>Max. 10 bytes</td></tr>
<tr><td>String constant</td><td>Max. 10 bytes</td></tr>
<tr><td rowspan="3">TAIL</td><td>Numerical constant</td><td>Max. 10 bytes</td></tr>
<tr><td>String constant</td><td>Max. 10 bytes</td></tr>
<tr><td>BCC</td><td>Only one BCC applicable</td></tr>
<tr><td rowspan="3">BODY</td><td>Numerical constant</td><td>Max. 10 bytes</td></tr>
<tr><td>String constant</td><td>Max. 10 bytes</td></tr>
<tr><td>Variable size parameter</td><td>Up to 4 available</td></tr>
<tr><td rowspan="9">Reception frame</td><td rowspan="2">HEAD</td><td>Numerical constant</td><td>Max. 10 bytes</td></tr>
<tr><td>String constant</td><td>Max. 10 bytes</td></tr>
<tr><td rowspan="3">TAIL</td><td>Numerical constant</td><td>Max. 10 bytes</td></tr>
<tr><td>String constant</td><td>Max. 10 bytes</td></tr>
<tr><td>BCC</td><td>Only one BCC applicable</td></tr>
<tr><td rowspan="4">BODY</td><td>Numerical constant</td><td>Max. 10 bytes</td></tr>
<tr><td>String constant</td><td>Max. 10 bytes</td></tr>
<tr><td>Fixed size parameter</td><td>Up to 4 can be set
3 fixed size parameters and 1 variable size parameter available</td></tr>
<tr><td>Variable size parameter</td><td>Only one variable size parameter can be set.
Impossible to add segment after variable size parameter</td></tr>
</table>

15.11.3 Create Frame

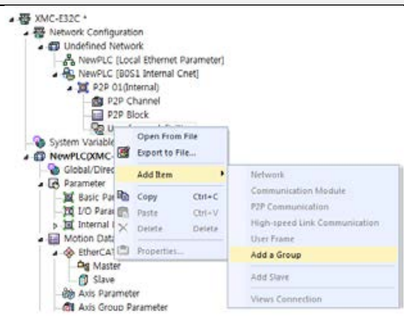
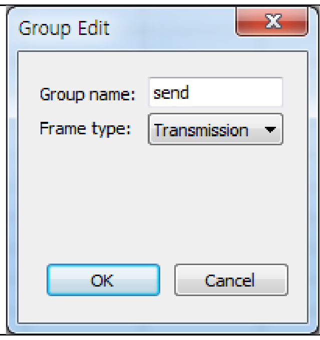
1) Channel settings

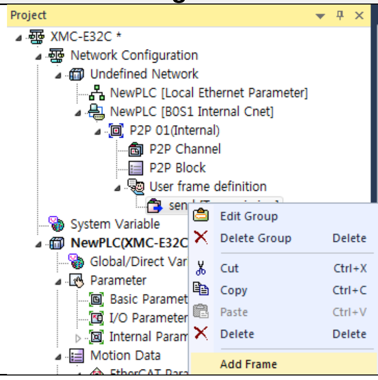
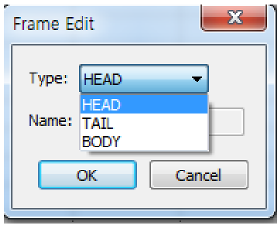
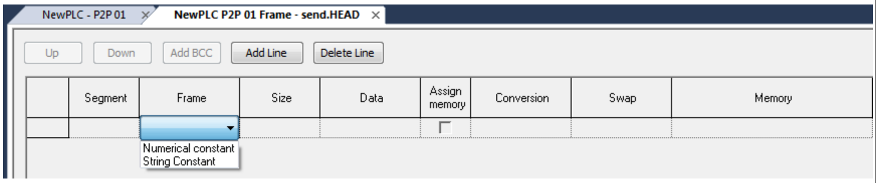
Cnet I/F can define the driver type for P2P service. However, if the P2P channel operates in the user frame definition, the operation mode of the default setting should be defined as P2P use.

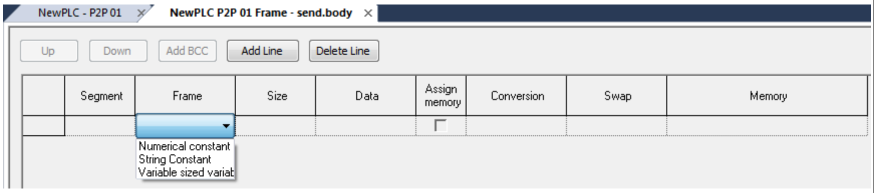


2) Creating transmission frame

The frame is largely divided into a head which indicates the beginning of the frame, a tail indicating the end, and a body, which is a data area. The method for creating the transmission frame is shown below.

Order	Setting contents	Setting method
1	User frame creation	
	1. Select user frame definition. 2. Click the right mouse button and select Add Group.	
2	Frame generation	
	1. The group name is the name of the frame a user wants to create. 2. Select transmission as a frame type because the transmission frame needs to be created at present.	

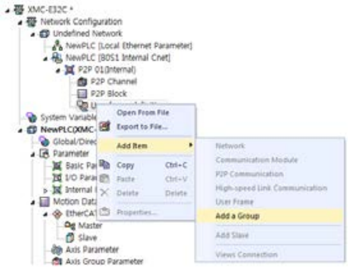
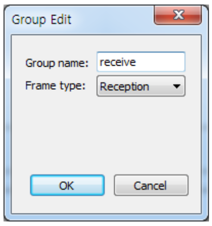
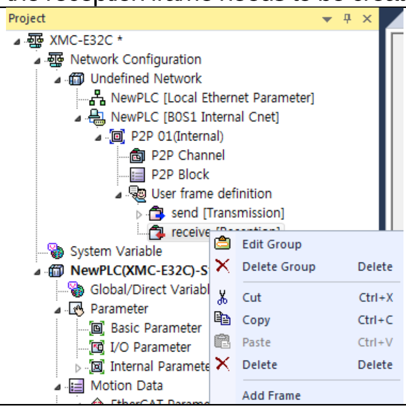
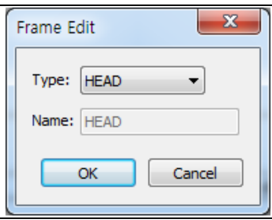
<table>
<tr><th>Order</th><th>Setting contents</th><th>Setting method</th></tr>
<tr><td rowspan="2">3</td><td>Frame generation</td><td></td></tr>
<tr><td colspan="2">1. Check frame generation.
2. Select the frame name and click the right mouse button.
3. Click Add Frame to create header, tail and body in the generated frame.
4. Edit Group: Used to change the frame name
5. Delete Group: Used to delete a frame.</td></tr>
<tr><td rowspan="2">4</td><td>HEAD,TAIL,BODY generation</td><td></td></tr>
<tr><td colspan="2">1. Click Add Frame and then select the type of frame to be created.
2. Type: HEAD,TAIL,BODY
3. Select HEAD.
4. Repeat the above process (No.3) to create tail and body.
5. The name of the frame editing window is enabled only when the type is BODY.
6. Multiple bodies can be created with different names.</td></tr>
<tr><td rowspan="2">5</td><td>HEAD registration</td><td></td></tr>
<tr><td colspan="2">1. Double click HEAD to create an editor screen.
2. Double click the editor screen or click the right mouse button to select Add Segment.
3. Select the type.
1) Numerical constant
(1) Define the fixed part of the frame as a constant
(2) The value of data item is Hex(hexadecimal)
2) String constant
(1) Register string constant in frame
(2) The value of data item is ASCII
4. Enter a value in data.
Example) Type: numerical constant
Data: 5(ENQ)
*You can edit, delete, insert and copy the segment by clicking the right mouse button in the generated segment .</td></tr>
</table>

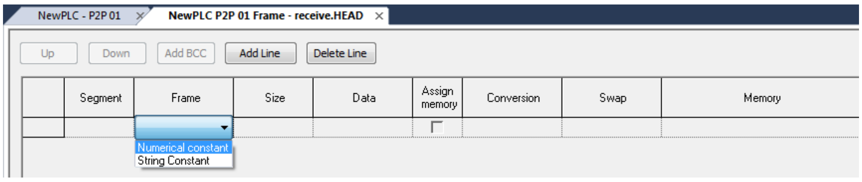
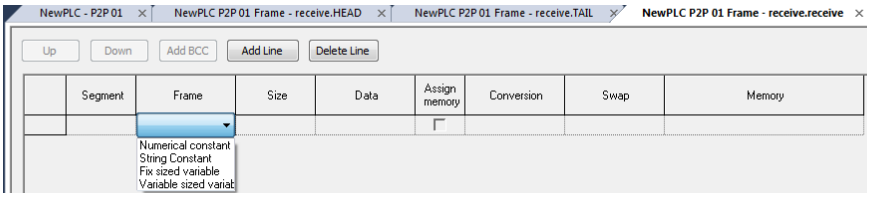
<table>
<tr><th>Order</th><th>Setting contents</th><th>Setting method</th></tr>
<tr><td>6</td><td>TAIL registration</td><td>1. Double click TAIL to create an editor screen.
2. The setting method is the same as that in the above process (No. 5).
3. Adding BCC is possible after the segment is created in the editor screen.</td></tr>
<tr><td rowspan="2">7</td><td>BODY registration</td><td>
</td></tr>
<tr><td colspan="2">1. Double click BODY to enable the editor screen and select the data type.
1) The meaning of numerical and string constants is the same as that in the header registration.

2) Variable type parameter
(1) Used to vary the length of the frame
(2) Up to 4 parameters can be set in one body
(3) Memory specification is automatically checked
(4) Control by byte

3)Conversion
▶Hex To ASCII: Convert the data read from the PLC memory into ASCII to configure the transmission frame
▶ASCII To Hex: Convert the data read from the PLC memory into Hex to configure the transmission frame
1) 4) Swap
▶2 byte swap: 2-byte swap of the data value (Example: 0x1234->0x3412)
▶4 byte swap: 4-byte swap of the data value (Example: 0x12345678->0x78564321
▶8 byte swap: 8-byte swap of the data value</td></tr>
</table>

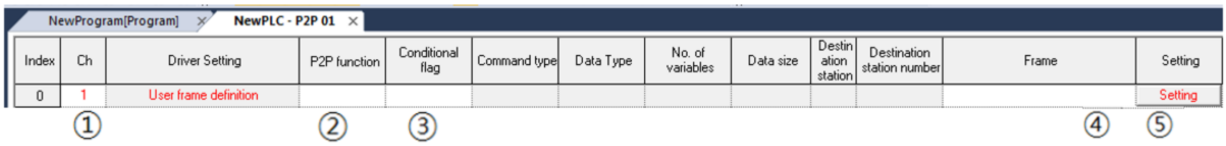
3) Creating reception frame

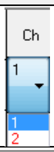
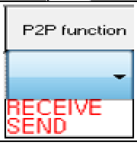
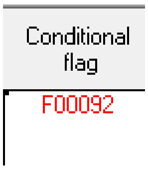
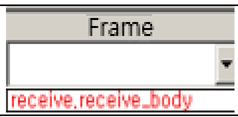
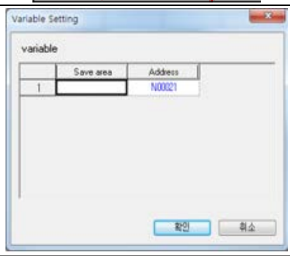
<table>
<tr><th>Order</th><th>Setting contents</th><th>Setting method</th></tr>
<tr><td rowspan="2">1</td><td>User frame creation</td><td></td></tr>
<tr><td colspan="2">1. Select user frame definition.
2. Click the right mouse button and select Add Group.</td></tr>
<tr><td rowspan="2">2</td><td>Frame generation</td><td></td></tr>
<tr><td colspan="2">1. The group name is the name of the frame a user wants to create.
2. Select reception as a frame type because the reception frame needs to be created at present</td></tr>
<tr><td rowspan="2">3</td><td>Frame generation</td><td></td></tr>
<tr><td colspan="2">1. Check frame generation.
2. Select the frame name and click the right mouse button.
3. Click Add Frame to create header, tail and body in the generated frame.
4. Edit Group: Used to change the frame name
5. Delete Group: Used to delete a frame.</td></tr>
<tr><td rowspan="2">4</td><td>HEAD,TAIL,BODY generation</td><td></td></tr>
<tr><td colspan="2">1. Click Add Frame and then select the type of frame to be created.
2. Type: HEAD,TAIL,BODY
3. Select HEAD.
4. Repeat the above process (No.3) to create tail and body.
5. The name of the frame editing window is enabled only when the type is BODY.
6. Multiple bodies can be created with different names.</td></tr>
</table>

<table>
<tr><th>Order</th><th>Setting contents</th><th>Setting method</th></tr>
<tr><td rowspan="2">5</td><td>HEAD registration</td><td>
</td></tr>
<tr><td colspan="2">1. Double click HEAD to create an editor screen.
2. Double click the editor screen or click the right mouse button to select Add Segment.
3. Select the type.
1) The meaning of each type is the same as that of the head registration shown in No. 5 in writing transmission frame.
4. Enter a value in data.</td></tr>
<tr><td>6</td><td>TAIL registration</td><td>1. Double click TAIL to create an editor screen.
2. The setting method is the same as that in the above process (No. 5).
3. Adding BCC is possible after the segment is created in the editor screen.</td></tr>
<tr><td rowspan="2">7</td><td>BODY registration</td><td>
</td></tr>
<tr><td colspan="2">1. Double click BODY to enable the editor screen and select the data type.
1) The meaning of numerical and string constants is the same as that in the header registration.
2) Variable type parameter
(1) Used in cases where the length of the frame varies
(2) One variable size parameter can be set, and segment cannot be added when the variable size parameter is set
(3) Possible to save data in PLC memory when checking Specify Memory
(4) Control by byte
3) Fixed size parameter
(1) Used if the frame length is fixed to a certain size
(2) Up to 4 parameters can be set in one body
(3) Possible to save data in PLC memory when checking Specify Memory
4) Specify Memory: Checked when the device area to be saved in PLC is set.
5) Conversion
▶Hex To ASCII: Convert the received data into ASCII to configure the reception frame
▶ASCII To Hex: Convert the received data into Hex to configure the reception frame
6) Swap
▶2 byte swap: 2-byte swap of the data value (Example: 0x1234->0x3412)
▶4 byte swap: 4-byte swap of the data value (Example: 0x12345678->0x78564321
▶8 byte swap: 8-byte swap of the data value</td></tr>
</table>

4) Parameter settings

In order to transmit and receive data by using the transmission and reception frames created through the user frame definition in P2P view of XG5000, you should set parameters through P2P block. The parameter setting method is shown below.



Number	Classification	Block type	Meaning
1	Channel		The set drive name changes according to the driver set in P2P driver.
2	P2P function		1. Receive: Used to receive arbitrary data by using a frame created according to the protocol of the partner station 2. Send: Used to send arbitrary data by suing a frame created according to the protocol of the partner station.
3	Start condition		1. Enter special flag or bit contact and select the point of time when data is transmitted and received. 2.In the user frame definition, it is enabled only when the P2P function is Send. 3. Example) _T20MS(cycle: 20ms),%MX01
4	Frame		1. I f Send is selected in the P2P function, select the body of the transmission frame created in the user frame.
			1. If RECEIVE is selected in the P2P function, select the body of the reception frame created in the user frame.
5	Settings		1. The setting in the P2P block of the user frame definition can be done only when Specify Memory of the fixed size parameter and variable size parameter created by a user is checked. 2. Storage area: It means the firs address of the area to store data received from the other station.

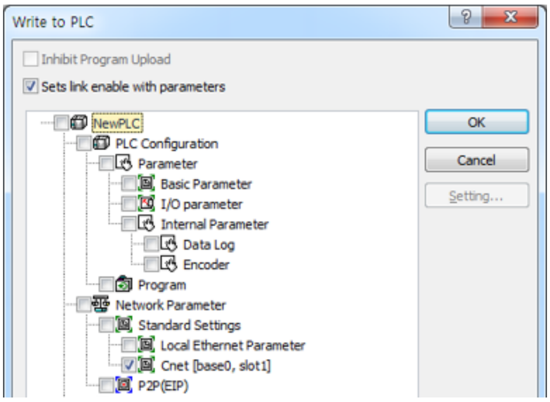
5) Writing parameters

(1) Writing parameter

(a) Select [Online] → [Write] or click the icon ( ).

(b) Check (✓) the module with the default settings and click [OK].

(c) Click the [OK] button, and when the parameter writing finishes, reset each module.

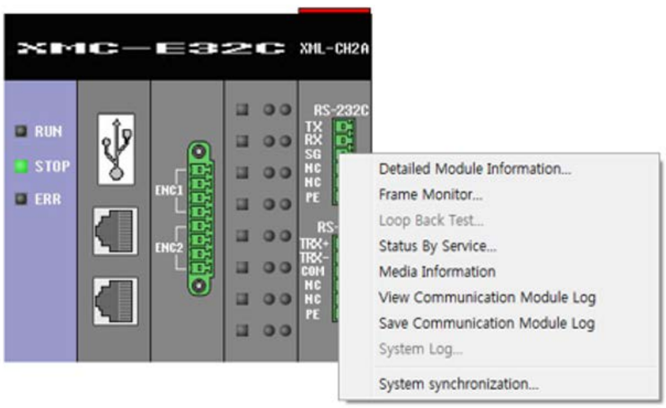


(2) Check operation

(a) Select [Online] → [Communication Module Settings] → [System Diagnosis] or click the icon ( ).

(b) Click the communication module whose status you want to diagnose and press the right mouse button.

(c) When the following screen appears, click [Frame Monitor] or [Service Status] to check the operation status.



15.11.4 Frame monitor

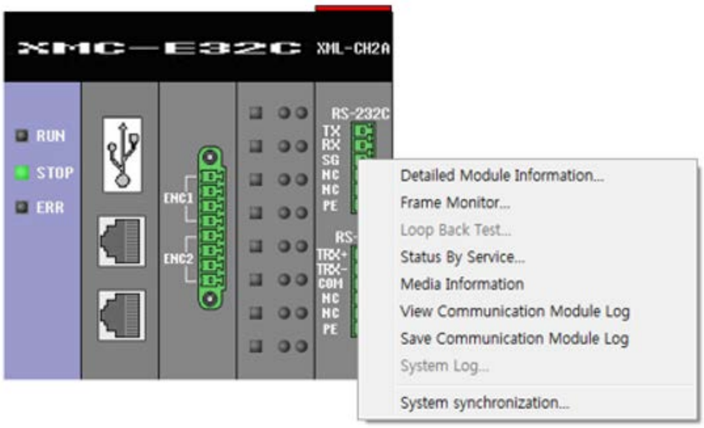
With the frame monitor function of XG5000, you can check the frame that client and server actually send and receive.

(1) Check operation

(a) Select [Online] → [Communication Module Settings] → [System Diagnosis] or click the icon ().

(b) Click the communication module whose status you want to diagnose and press the right mouse button.

(c) When the following screen appears, click [Frame Monitor] or [Service Status] to check the operation status.



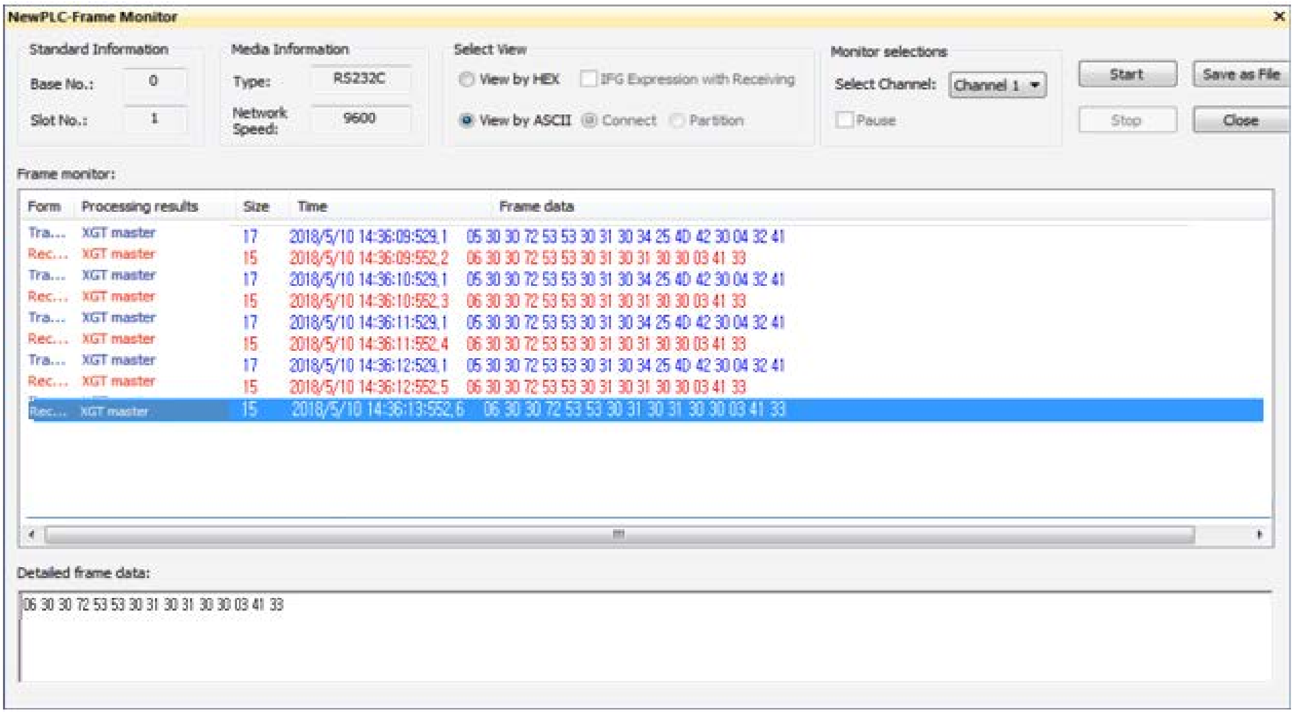
(2) Frame monitor

(a) Select the channel you want to monitor.

(b) When the protocol is in Modbus ASCII mode, select View as ASCII.

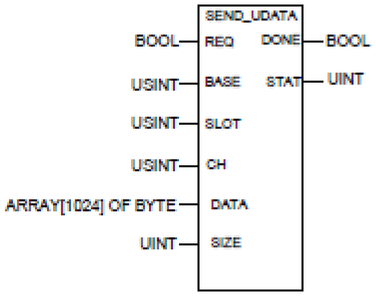
(c) Select Hex View when the protocol is in Modbus RTU mode.

(d) Click [Start] to check the sending / receiving frame.



15.11.5 User-defined communication commands

(1) SEND_UDATA

Function block	Descriptions
	Input REQ :Function block is executed(pulse operation) if 0 -> 1 BASE : Base SLOT : Slot CH : Channel(1 or 2) DATA :Data area to send SIZE : Data size to send Output DONE :1 is displayed when executed without error STAT : Status information

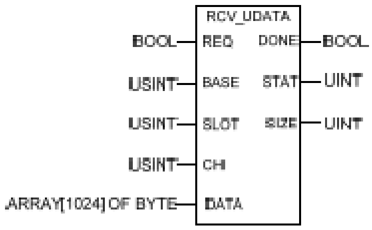
a) Functions

- This command is used to transmit user-defined data (hereafter referred to as DATA).
- For BASE and SLOT, enter the base and slot number on which the current CNET module is mounted.
- CH means channel number, and only 1 or 2 should be set.
- DATA should be declared as ARRAY OF BYTE type.
- The size of the array declared as SIZE ranges from 1 to 1024.(Unit: Byte)
- Store as much data as the number of SIZE from DATA [0] in the transmission buffer. (The number of data sizes that can be sent at a time is limited to 1024)
- When normal operation is performed, 1 is displayed in DONE and STAT, and status information is displayed when an error occurs.

b) Errors

STAT	Contents	Details
0	Initial state	Initial state before command execution
1	No error	Normal operation
2	Module setting error	The module is not mounted on the base slot, or it is not a CNET module
3	Channel setting error	It occurs when exceeding the input range(1, 2)
4	Array size error	The size of transmitted data exceeds 1024
5	Communication parameter setting error	The communication parameter of the Cnet module is not set to user-defined, and Link Enable is not done.
6	Command timeout error	There is no response from the module, or the maximum scan time (10 scans) is exceeded.

(2) RCV_UDATA

Function block	Description
	Input REQ :Function block is executed (pulse operation) if 0 -> 1 BASE : Base SLOT : Slot CH : Channel(1 or 2) DATA :Received data Output DONE :1 is displayed when executed without error STAT :Status information SIZE :Size of received data

a) Functions

- This command is used to store data of the frame received through CNET.
- For BASE and SLOT, enter the base and slot number on which the current CNET module is mounted.
- CH means channel number, and only 1 or 2 should be set.
- DATA should be declared as ARRAY OF BYTE type.
- The size of the array declared as SIZE ranges from 1 to 1024.(Unit: Byte)
- SIZE represents the size of the received data
- When normal operation is performed, 1 is displayed in DONE and STAT, and status information is displayed when an error occurs.

b) Errors

STAT	Contents	Details
0	Initial state	Initial state before command execution
1	No error	Normal operation
2	Module setting error	The module is not mounted on the base slot, or it is not a CNET module
3	Channel setting error	It occurs when exceeding the input range(1, 2)
4	Array size error	The size of transmitted data exceeds 1024
5	Communication parameter setting error	The communication parameter of the Cnet module is not set to user-defined, and Link Enable is not done.
6	Command timeout error	There is no response from the module, or the maximum scan time (10 scans) is exceeded.

15.12 Program examples

15.12.1 How to set each operation mode

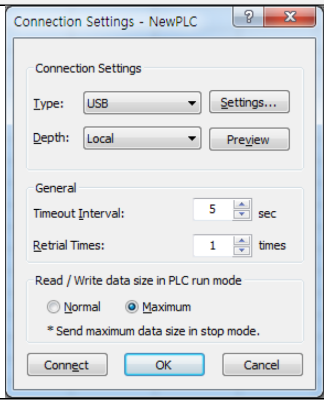
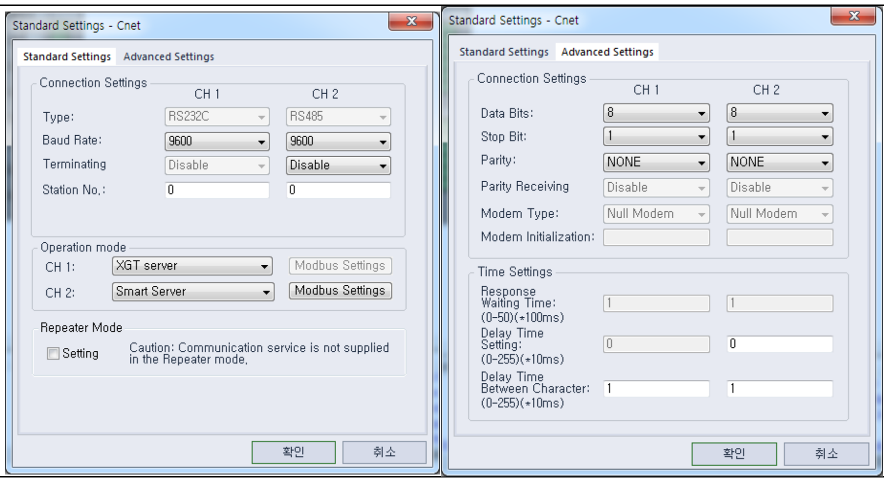
The operation mode of XMC built-in Cnet is classified into P2P service and server function.

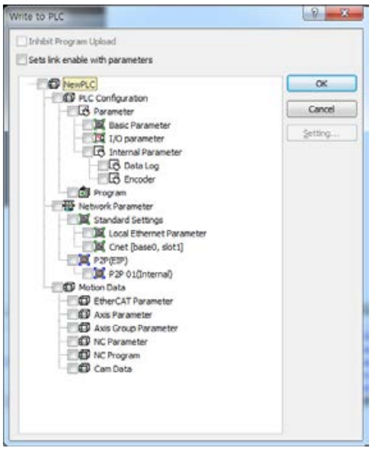
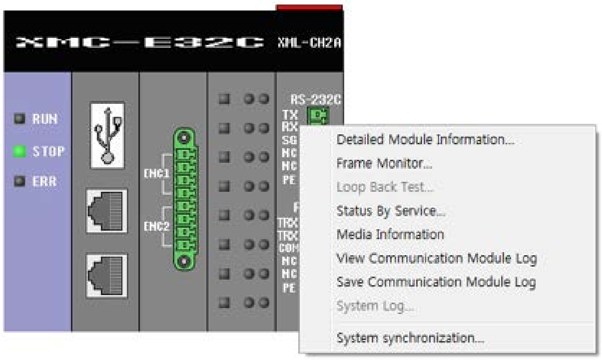
• P2P service: It operates as a client (master) and makes a request to read/write data from/to the partner station.
- XGT client
- LS BUS client
- Modbus RTU/ASCII client
- User frame definition

• Server: It operates as a server (slave) and responds according to the protocol type when requested by the client.
- XGT server
- Modbus RTU server

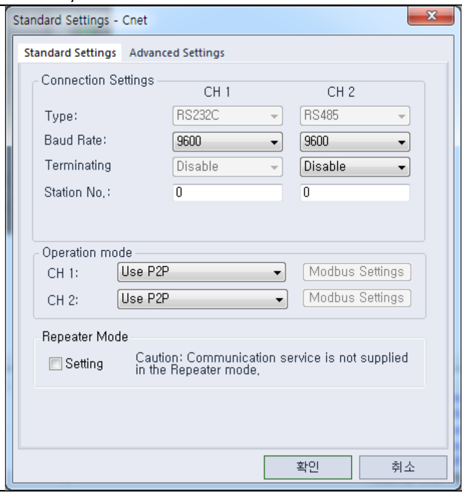
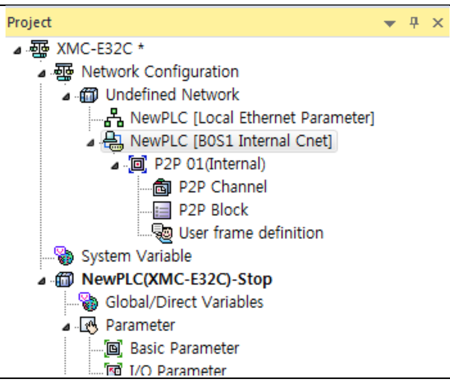
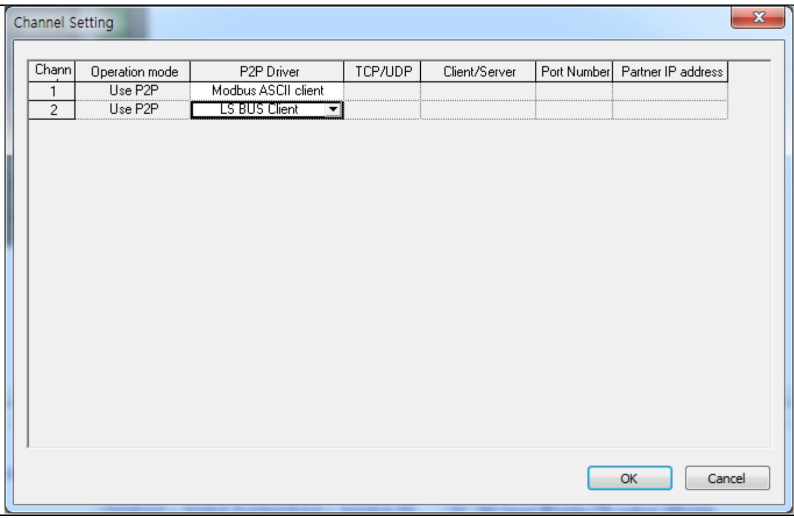
The setting method for each operation mode is as follows

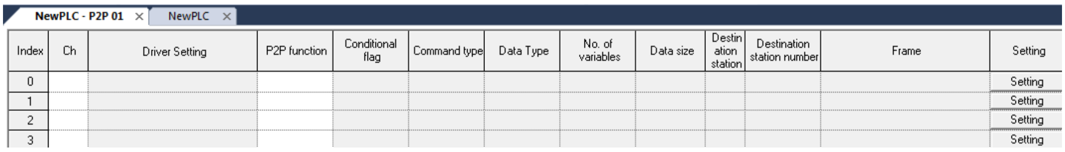
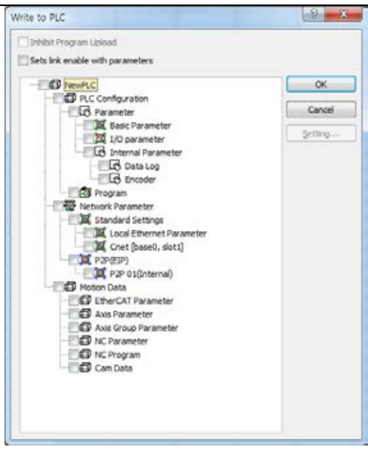
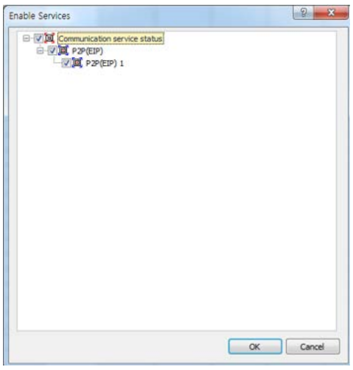
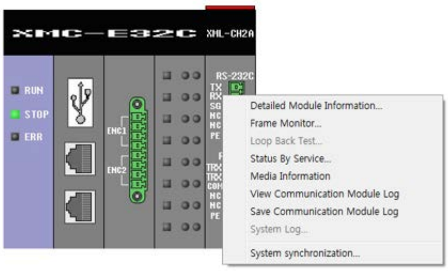
1) When operating as a server

Sequence	Setting process	Setting method
1	Connection settings	
		1. Select Online ->Connection Settings or click the icon ( ). 2. Set the connection option suitable for user's environment and click the Connect.
2	Open from PLC	Select the Read from Project->PLC or click the icon( ) to read the information of the module mounted on the current basic unit.
3	Default settings	
		1. Double click the Cnet I/F to execute the default setting window and set the communication type, communication speed, module type, data bit, stop bit and station number in the connection setting window. 2. Model initialization is only possible when the modem type is a dial-up modem, not a null modem. 3. Latency setting: it means that the frame is sent after the delay time set by a user. 1) Operation setting: It can be set when the communication type is RS-485. * When used as a Modbus ASCII server, the data bit should be set to 7.

<table>
<tr><th>Sequence</th><th>Setting process</th><th>Setting method</th></tr>
<tr><td>4</td><td>Operation mode selection</td><td>1. Select the operation mode of a server to be used by a user.
2. XMC Cnet I/F supports the XGT server, Modbus ASCII server and Modbus RTU server.</td></tr>
<tr><td rowspan="2">5</td><td>Writing parameters</td><td>
</td></tr>
<tr><td colspan="2">1. Select Online ->Write or click the icon ( ).
2. Click the OK button.
3. Click the [OK] button, and when the parameter writing finishes, reset each module.</td></tr>
<tr><td rowspan="2">6</td><td>Operation check</td><td>
</td></tr>
<tr><td colspan="2">1. Slect Online -> Comunication Module Settings -> System Diagnostics or click the icon( ).
2. Click the module, press the right mouse button.
3. Click [Frame Monitor] or [Service Status] to check the operation status.</td></tr>
</table>

2) When operating as a P2P service(client)

<table>
<tr><th>Sequence</th><th>Setting process</th><th>Setting method</th></tr>
<tr><td>1</td><td>Default settings</td><td>The steps 1 to 3 are the same as those in the above case.
*When set to Modbus ASCII client, the data bit is set to 7</td></tr>
<tr><td rowspan="2">2</td><td>Operation mode</td><td>
</td></tr>
<tr><td colspan="2">1.P2P use is selected as an operation mode.</td></tr>
<tr><td rowspan="2">3</td><td>P2P settings</td><td>
</td></tr>
<tr><td colspan="2">1. Right-click the Cnet module in the project tree and select [Add Item] → [P2P Communication].
2. Select the P2P number (01).</td></tr>
<tr><td rowspan="2">4</td><td>P2P channel settings</td><td>
</td></tr>
<tr><td colspan="2">1. Double click the P2P channel and select protocols for each channel.
2. P2P driver supports the user frame definition, XGT client and Modbus RTU/ASCII client.</td></tr>
</table>

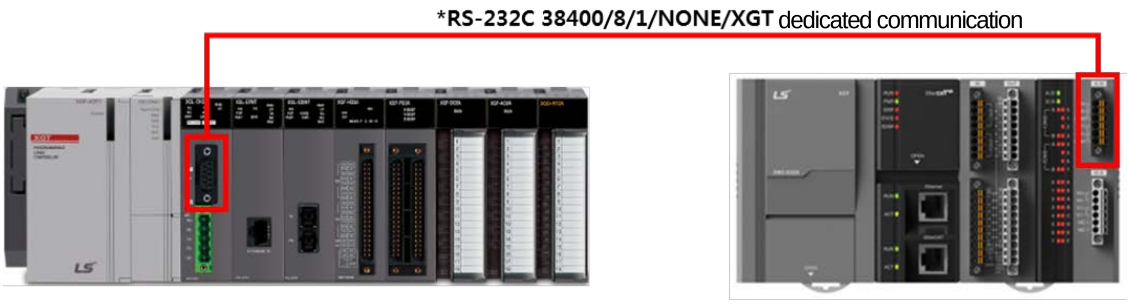
<table>
<tr><th>Sequence</th><th>Setting process</th><th>Setting method</th></tr>
<tr><td rowspan="2">5</td><td>P2P block settings</td><td>
</td></tr>
<tr><td colspan="2">1. P2P block setting values are enabled differently according to the type of client selected in the channel settings.
2. Create the frame in the enabled cell according to the protocol type.
*In the case of user frame definition, it is available only when the frame is created in the user frame definition.</td></tr>
<tr><td rowspan="2">6</td><td>Writing parameters</td><td>
</td></tr>
<tr><td colspan="2">1. Select Online ->Write or click the icon ( ).
2. In the default settings, check the basic settings and P2P that have been set up, and then click the [OK] button.
3. Click the [OK] button, and when the parameter writing finishes, reset each module.</td></tr>
<tr><td rowspan="2">7</td><td>Link Enable</td><td>
</td></tr>
<tr><td colspan="2">1. Select Online ->Communication Module Settings ->Link Enable or click the icon( ).
2.Check the P2P whose setting is completed and click Write.</td></tr>
<tr><td rowspan="2">8</td><td>Operation check</td><td>
</td></tr>
<tr><td colspan="2">1. Slect Online ->Communication Module Settings ->System Diagnostics or click the icon( ).
2. Click the module and then right-click.
3. Click [Frame Monitor] or [Service Status] to check the operation status.</td></tr>
</table>

15.12.2 Dedicated Communication Examples

What is a dedicated communication?

• It is a protocol defied by LS ELECTRIC and is classified into XGT client and XGT server
• XGT client: requests the server to read/write data
• XGT server: responds to the client's request

Here is an example of how to use XGT dedicated communication using XGT protocol.



<client> < server >

Example of dediated service system configuration

1) Client-side settings

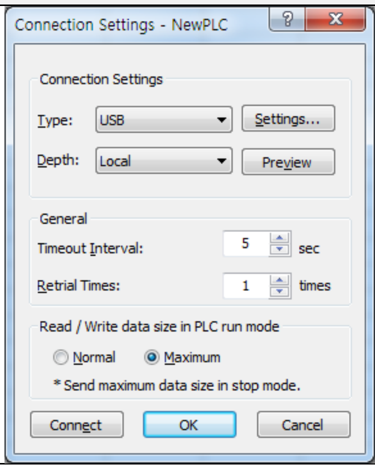
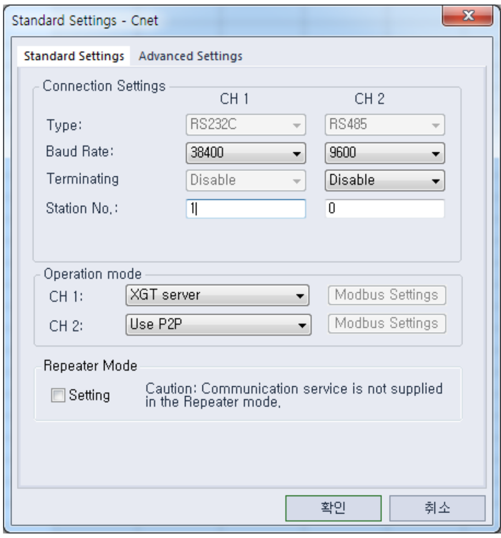
<table>
<tr><th colspan="2">Type</th><th>Setting contents</th></tr>
<tr><td colspan="2">Basic unit</td><td>XGK-CPUH</td></tr>
<tr><td colspan="2">Communication module</td><td>XGL-CH2B (Slot No.1)</td></tr>
<tr><td colspan="2">Communication type</td><td>RS-232C</td></tr>
<tr><td colspan="2">Communication speed</td><td>38,400</td></tr>
<tr><td colspan="2">Data bit</td><td>8</td></tr>
<tr><td colspan="2">Stop bit</td><td>1</td></tr>
<tr><td colspan="2">Parity bit</td><td>None</td></tr>
<tr><td colspan="2">Modem type</td><td>Null modem</td></tr>
<tr><td colspan="2">Operation cycle</td><td>200ms</td></tr>
<tr><td rowspan="2">Operation status</td><td>Write</td><td>Save 1 word of M100 on the client side in M110 on the server side</td></tr>
<tr><td>Read</td><td>Save 1 word of M110 on the server side in M110 on the client side</td></tr>
</table>

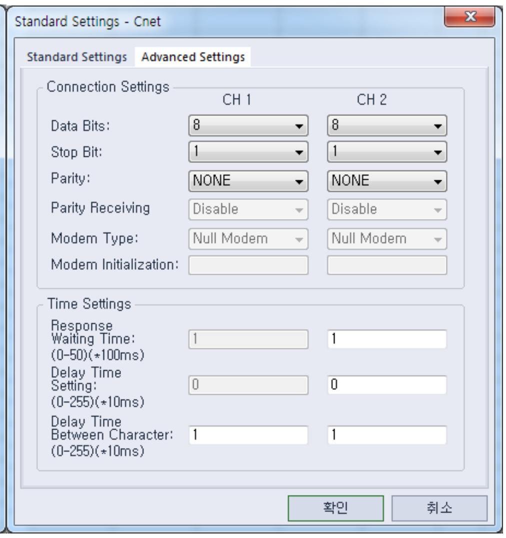
2) Server-side settings

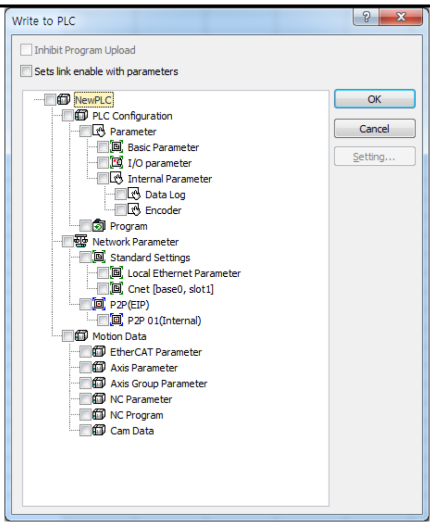
Type	Setting contents
Basic unit	XMC-E32C
Communication module	Basic unit built-in communication(RS-232C)
Communication type	RS-232C
Communication speed	38,400
Data bit	8
Stop bit	1
Parity bit	None
Modem type	Null modem
Station No.	1

3) XGT server-side settings

On the preceding system, set up the server as follows:

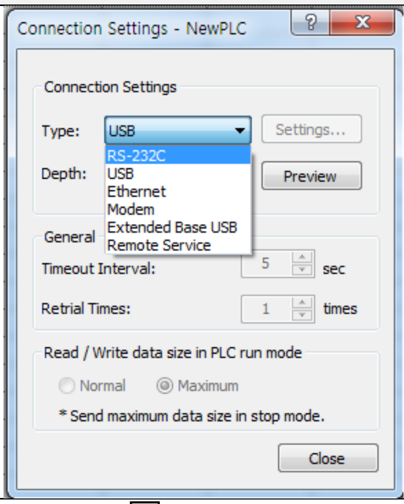
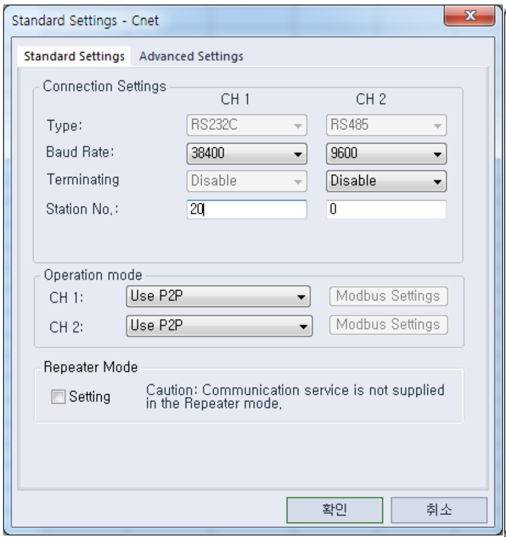
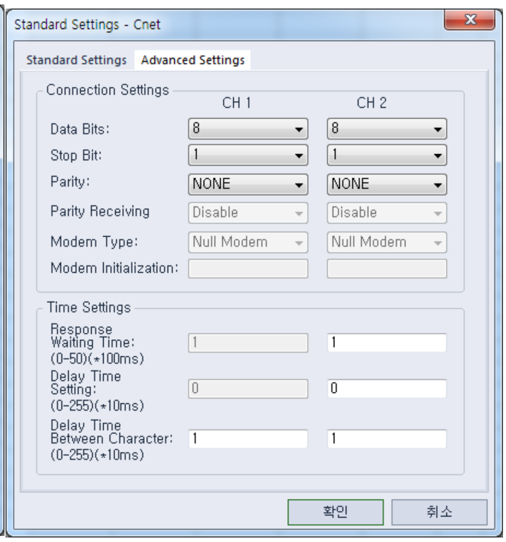
<table>
<tr><th>Sequence</th><th>Setting process</th><th>Setting method</th></tr>
<tr><td rowspan="2">1</td><td>Connection settings</td><td>
</td></tr>
<tr><td colspan="2">1. Select Online ->Connection Settings or click the icon ( ).
2. Set the connection option suitable for user's environment and click the Connect.</td></tr>
<tr><td>2</td><td>Oepn from PLC</td><td>Select the Read from Project->PLC or click the icon( ) to read the information of the module mounted on the current basic unit.</td></tr>
<tr><td rowspan="2">3</td><td>Default settings</td><td>


</td></tr>
<tr><td colspan="2">1. It is created the same as in the setting value on the server side of the exemplary system in the built-in communication channel 1.
2. Operation mode is set as XGT server since it operates as a dedicated communication server.</td></tr>
</table>

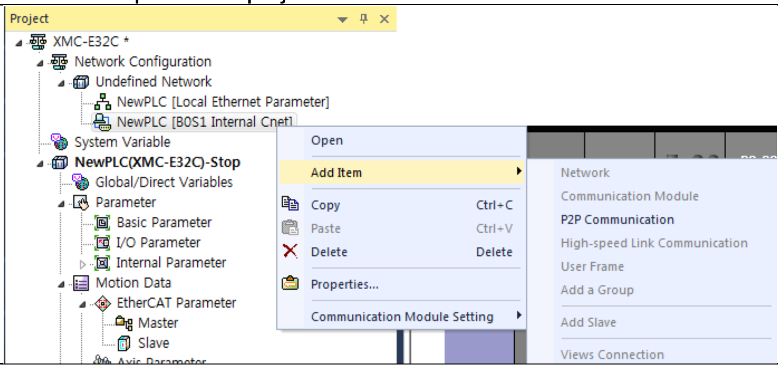
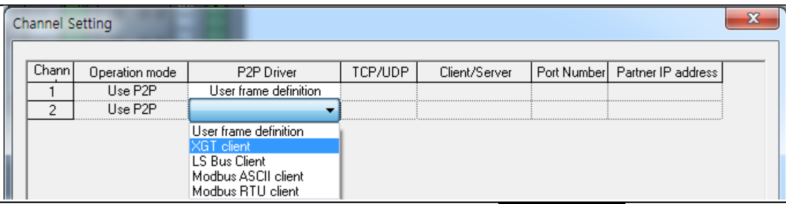
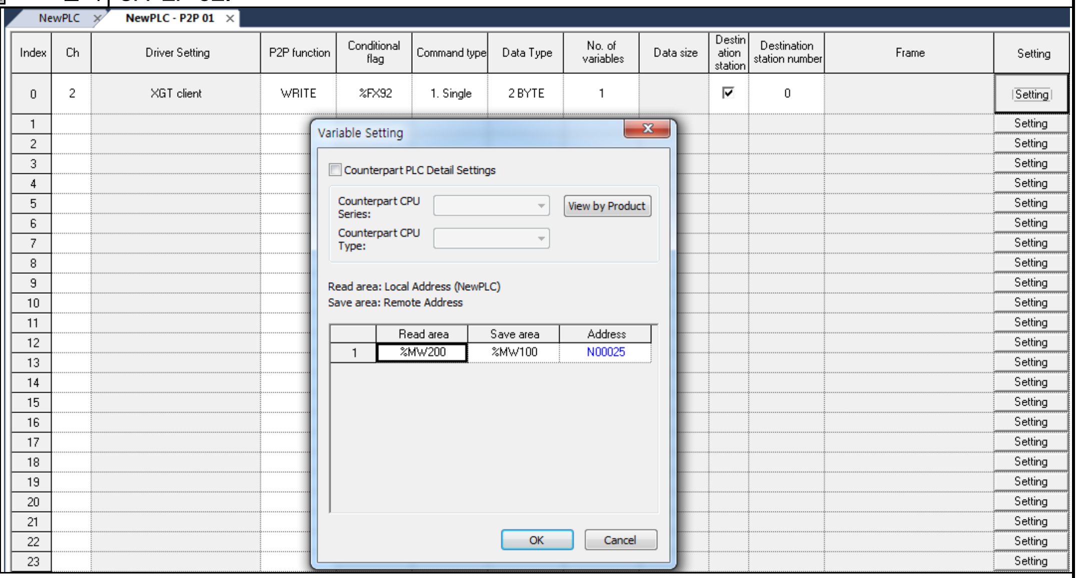
<table>
<tr><td rowspan="2">4</td><td>Writing parameters</td><td>
</td></tr>
<tr><td colspan="2">1. Select Online ->Write or click the icon ( ).
2. In the default settings, check the basic settings and P2P that have been set up, and then click the [OK] button.
3. Click the [OK] button, and when the parameter writing finishes, reset each module.</td></tr>
</table>

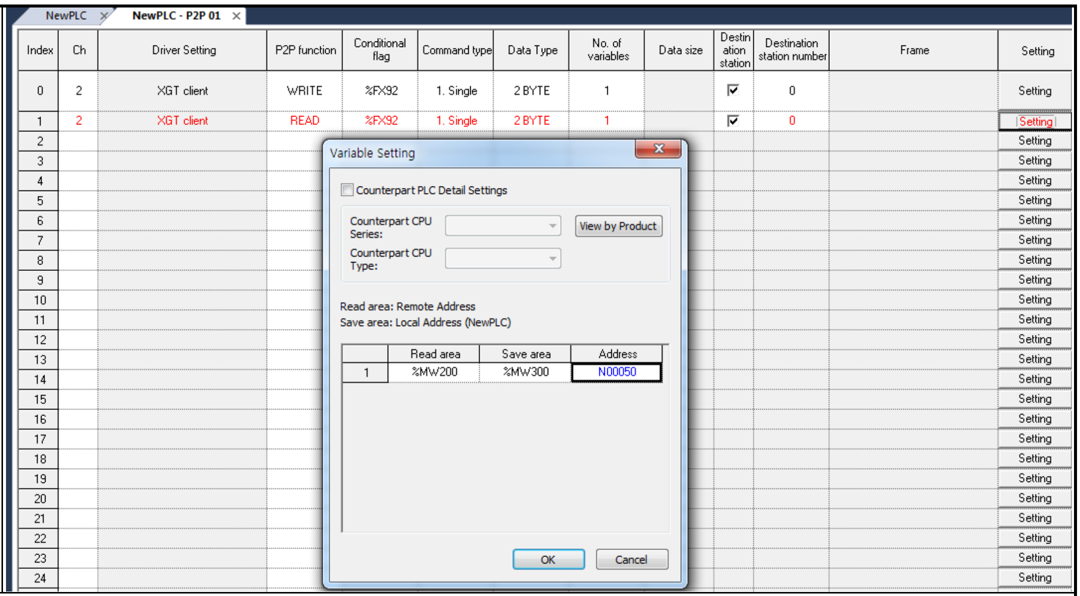
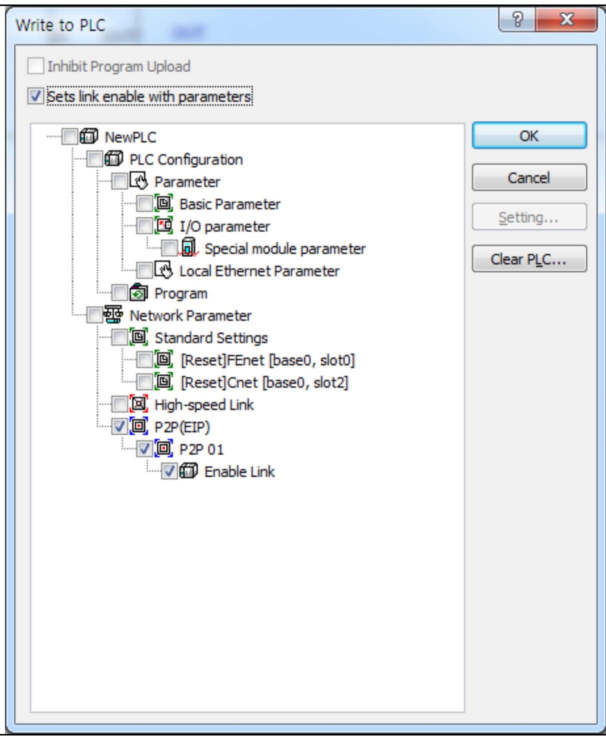
4) XGT client-side settings

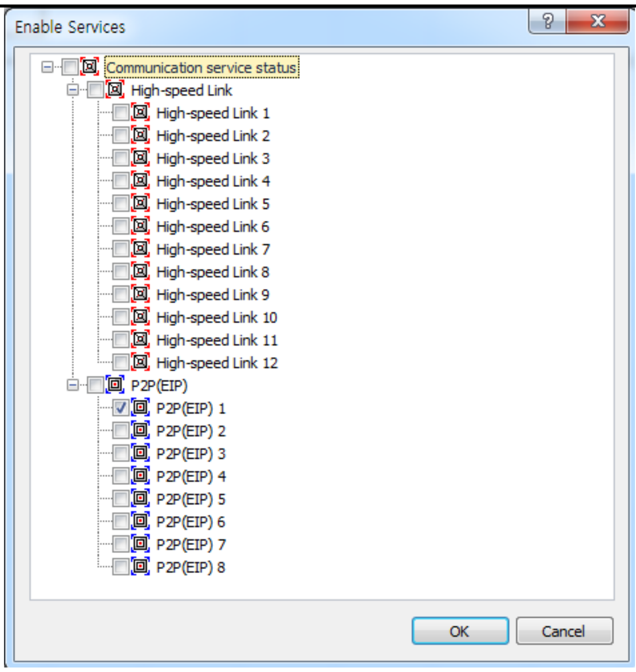
In order to operate the XBL-C21A on the client side as an XGT client, the default settings of the Cnet I/F module are done in the following order.

Sequence	Setting process	Setting method
1	Connection settings	
		1. Select Online ->Connection Settings or click the icon ( ). 2. Set the connection option suitable for user's environment and click the Connect.
2	Open from PLC	Select the Read from Project->PLC or click the icon ( ) to read the information of the module mounted on the current basic unit.
3	Default settings	
		1. Select XBL-C21A to create in the same way as in the setting value on the client side of the exemplary system in channel 2. 2. Set it to an arbitrary station number (0~255) since the station number setting is meaningless when operating as a client. 3. Operation mode should always be set to P2P use when operating as a client.

When the default setting is completed, P2P channel and P2P block should be set. The setting method is shown below.

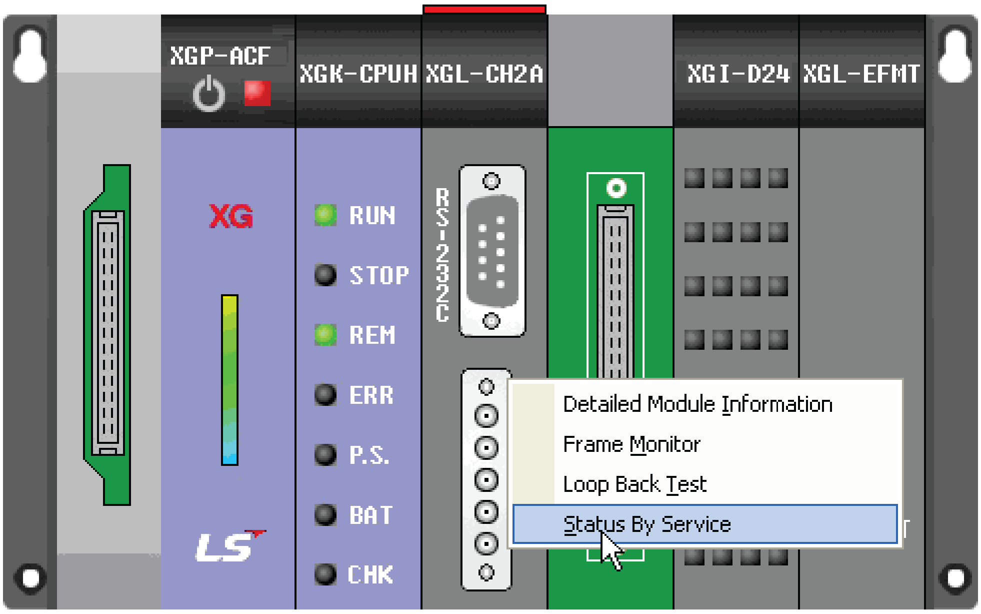
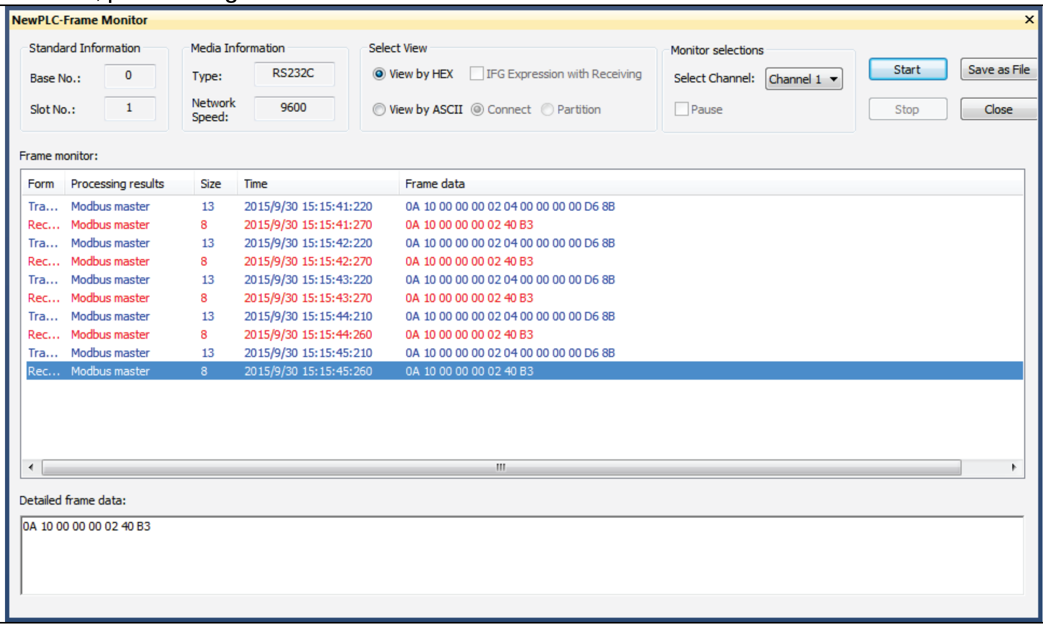
Sequence	Setting process	Setting method
1	P2P settings	Click [P2P] in the lower part of the project window.
2	Communication module settings	 1. Double click [P2P 02] in the project window.(P2P 01 is fixed as a built-in communication) 2. Select the slot number (No.1) that acts as a client and click [OK].
3	P2P channel settings	 1. Double click [P2P 채널] of P2P 02 to set the P2P driver of channel 2 as a [XGT Client] and click OK.
4		1. Double click [P2P 블록] of P2P 02.
5	Write operation settings	 1. Channel: Select the channel 2 set as the XGT client in P2P channel setting. 2. Select WRITE to perform the write operation. 3. Start condition: Use special flag F92 to transmit frame every 200㎳. 4. Method, data type: Select 2 Byte which means individual and word since I word is to be written. 5. Number of variables: Select 1 variable since it is 1 word. 6. Partner station number: Enter 1, which is the station number on the server side. 7. Settings: Set the Read area and Storage area and click [OK]. 1) Read area: Device address of data stored on the client side 2) Storage area: Device address of data stored on the server side * When all settings are completed normally, the text color of the index changes to black.

6	Read operation settings	
	1. Channel, start condition, method, data type, number of variables and partner station number are set as the same as write operation. 2. P2P function: Select READ since it performs read operation. 3. Settings: Set the read and storage area and click OK. 1) Read area: Device area of data stored in server station 2) Storage area: Device area of data to be stored in client	
7	Writing parameters	
	1. Select Online -> Write or click the icon ( ). 2. Click OK. 3. Click the OK button to complete the parameter write and then reset the module.	

<table>
<tr><td rowspan="2">8</td><td>Link Enable</td><td>
</td></tr>
<tr><td colspan="2">1. Select Online ->Communication Module Settings ->Link Enable or click the icon( ).
2. Check the P2P whose setting is completed and click Write.</td></tr>
</table>

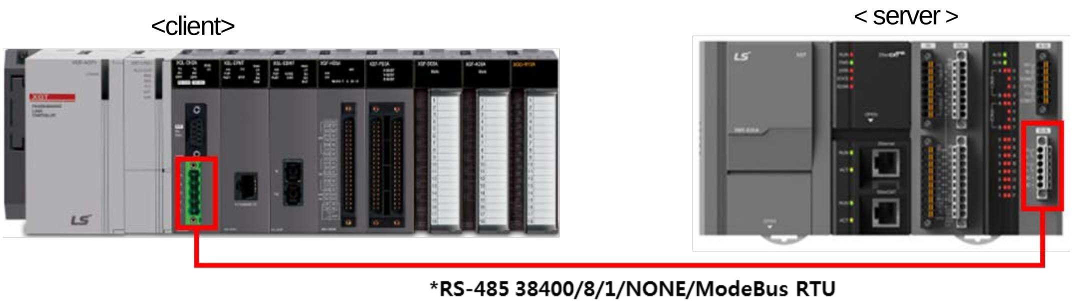
5) Operation status check

Frames transmitted and received through the frame monitoring of XG5000 can be analyzed to determine whether the communication status is normal. The frame monitoring method of Cnet I/F module is the same regardless of the protocol type, as shown below.

<table>
<tr><th>Sequence</th><th>Setting process</th><th>Setting method</th></tr>
<tr><td rowspan="2">1</td><td>System diagnostics</td><td>
</td></tr>
<tr><td colspan="2">1. Connect to XG5000 on the client side and select Online ->Communiation Module Settings ->System Diagnostics , or click the icon( ).
2. Click on the module, press the right mouse button and click the frame monitor.</td></tr>
<tr><td rowspan="2">2</td><td>Frame monitor</td><td>
</td></tr>
<tr><td colspan="2">1. Select channel 1 and click Start.
2. Since the dedicated service is ASCII communication, the frame that operates normally can be checked by selecting View in ACSCII.
*Select View in Hex for Modbus RTU, and select View in ASCII for Modbus ASCII.</td></tr>
</table>

15.12.3 Modbus Communication Examples

The system configuration of Modbus communication (Modbus RTU mode) example is shown in below, and the communication settings are summarized in the table below.



• XBL-C41A installed in slot No. 1 of client PLC

1) Client-side settings

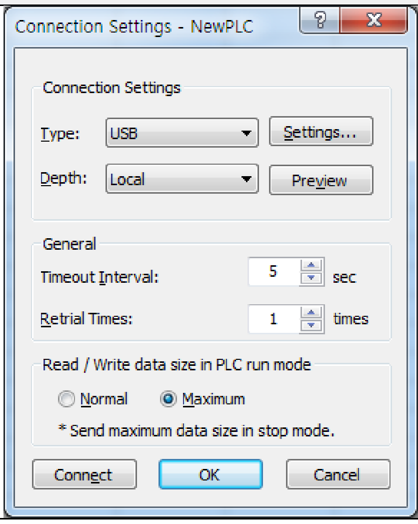
Basic unit		XGI-CPUUN
Communication module		XGL-CH2B(Slot No. 2)
Communication type		RS-485
Communication speed		38,400
Data bit		8
Stop bit		1
Parity bit		None
Operation cycle		200ms
Operation status	Write	▶Save 1 word of MW100 on the client side in the word write area M1 on the server side ▶Save 4 words from MW101 on the client side and to M5 in the word write area M2 ▶Save MX1 bit value on the client side in the bit write area M20 on the server side ▶Save MX2~MX17 bit value on the client side in the bit write area M21~M36 on the server side
	Read	▶Save 1 word value of M2 in the read area on the server side in MW160 on the client side ▶Save 4 words from M0 of the read area on the server side in MW150 to MW153 on the client side ▶Save bit values of M10 in the bit read area on the server side in the 1th bit of MX170 on the client side ▶Save bit values of M10 to M25 in the bit read area on the server side in MX180 to MX195 bits on the client side

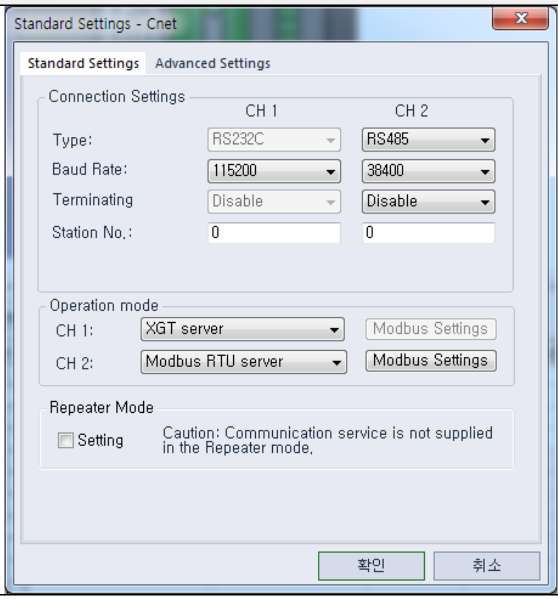
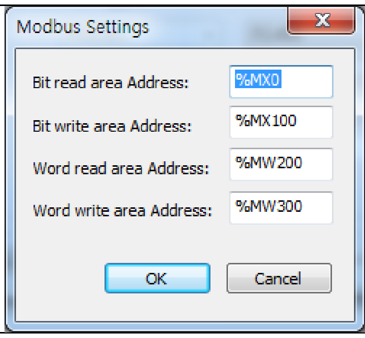
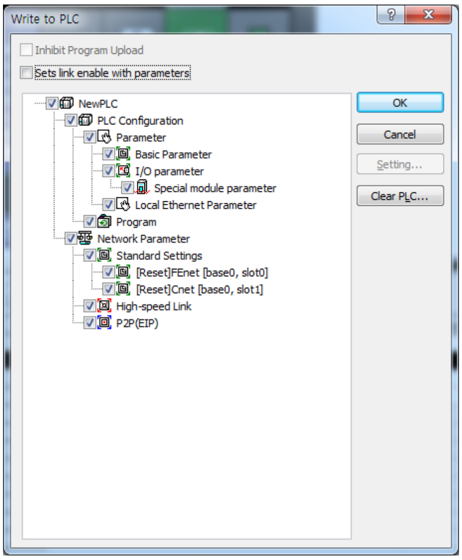
2) Server-side settings

<table>
<tr><td colspan="2">Basic unit</td><td>XMC-E32C</td></tr>
<tr><td colspan="2">Communication type</td><td>Built-in RS-485</td></tr>
<tr><td colspan="2">Communication speed</td><td>38,400</td></tr>
<tr><td colspan="2">Data bit</td><td>8</td></tr>
<tr><td colspan="2">Stop bit</td><td>1</td></tr>
<tr><td colspan="2">Parity bit</td><td>None</td></tr>
<tr><td colspan="2">Station number</td><td>1</td></tr>
<tr><td rowspan="4">Start address</td><td>Bit read</td><td>P0</td></tr>
<tr><td>Bit write</td><td>M0</td></tr>
<tr><td>Word read</td><td>P0</td></tr>
<tr><td>Word write</td><td>M0</td></tr>
</table>

3) Modbus RTU server-side settings

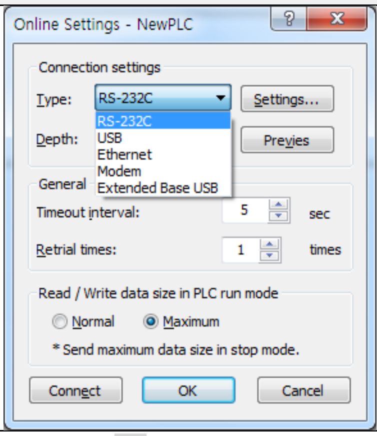
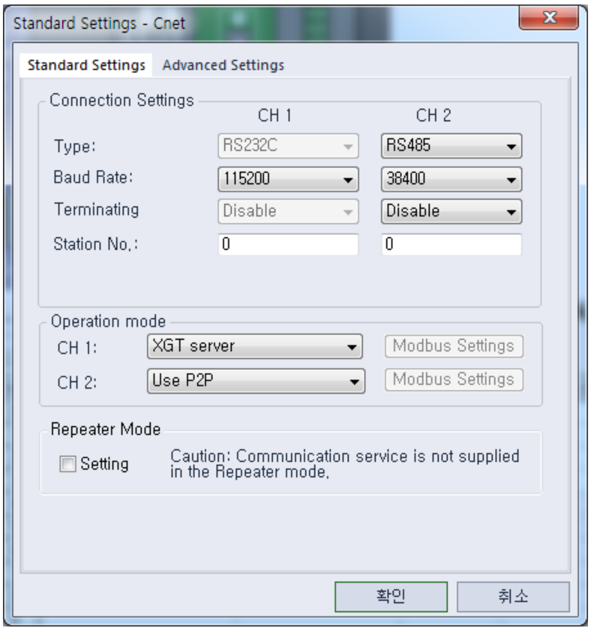
It is designed to operate the built-in communication channel of XMC-E32C as a Modbus RTU server and set as shown below.

<table>
<tr><th>Sequence</th><th>Setting process</th><th>Setting method</th></tr>
<tr><td rowspan="2">1</td><td>Connection settings</td><td>
</td></tr>
<tr><td colspan="2">1. Select Online -> Connection Settings or click the icon( ).
2. Set the connection option suitable for user’s environment and click Connection.</td></tr>
<tr><td>2</td><td>Open from PLC</td><td>Select the Read from Project->PLC or click the icon( ) to read the information of the module mounted on the current basic unit.</td></tr>
</table>

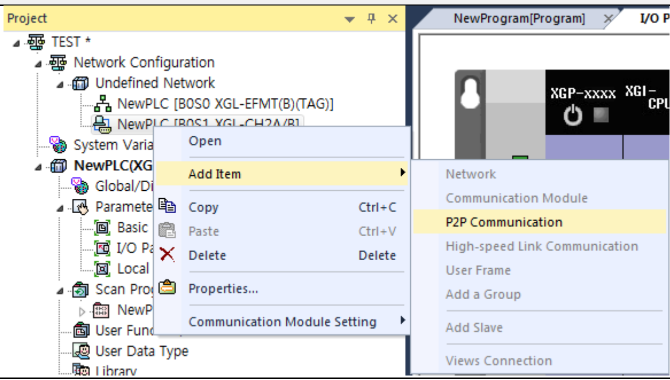
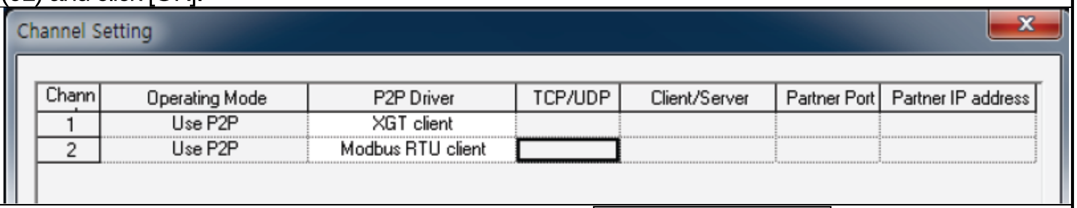
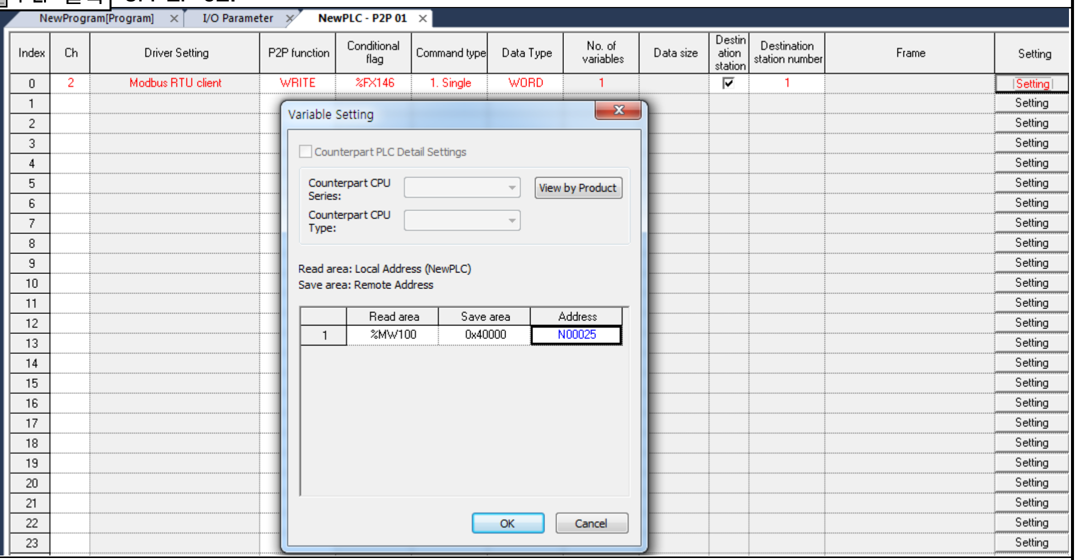
Sequence	Setting process	Setting method
3	Default settings	
		1. It is created the same as in the setting value on side server side in the built-in communication channel 2. 2. Modbus RTU server is set as operation mode.
4	Modbus settings	
		1. Start address of bit read area : MX0 2.Start address of bit write area : MX100 3. Start address of word read area : MW200 4.Start address of word write area : MW300
5	Writing parameters	
		1. Select Online -> Write Parameter or click the icon( ). 2. Click the OK button. 3. When the parameter write is completed after clicking the OK button, the changed parameter is automatically applied.

4) Modbus RTU Client Settings

In order to operate XBL-241A on the client side as an XGT client, the Cnet I/F module is set in the following order.

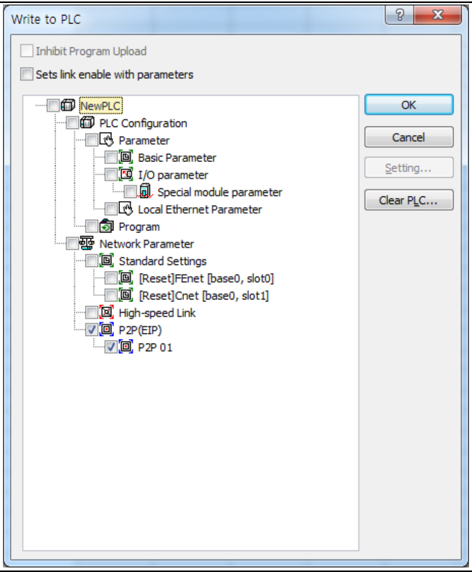
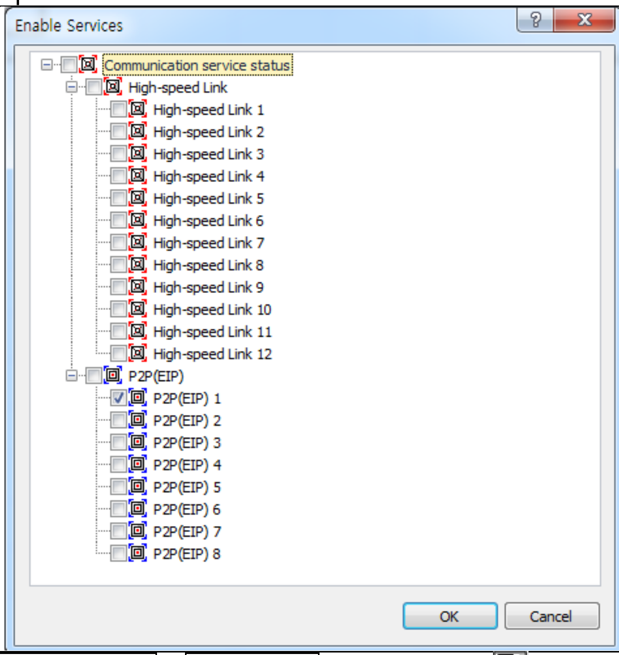
Sequence	Setting process	Setting method
1	Connection settings	
	1. Select Online ->Connection Settings or click the icon ( ). 2. Set the connection option suitable for user's environment and click the Connect.	
2	Open from PLC	Select the Read from Project->PLC or click the icon( ) to read the information of the module mounted on the current basic unit.
3	Default settings	
	1. It is created the same as in the setting value on the client side of the exemplary system in channel 2. 2. Set it to an arbitrary station number (0~255) since the station number setting is meaningless when acting as a client. 3. Operation mode should always be set to P2P use when operating as a client.	

When the default setting is completed, P2P channel and P2P should be set. The setting method is shown below.

<table>
<tr><th>Sequence</th><th>Setting process</th><th>Setting method</th></tr>
<tr><td rowspan="2">1</td><td>Communication module settings</td><td>
</td></tr>
<tr><td colspan="2">1. Click on the[Select Cnet]->[Click Right Mouse Button]->[Add Item]->[P2P Communication].
2. Select P2P No.(01) and click [OK].</td></tr>
<tr><td rowspan="2">2</td><td>P2P channel settings</td><td>
</td></tr>
<tr><td colspan="2">1.Double click P2P 채널 of P2P 02, set the P2P driver of channel 2 to Modbus RTU Client and click OK.</td></tr>
<tr><td>3</td><td colspan="2">1. Double click P2P 블록 of P2P 02.</td></tr>
<tr><td rowspan="2">4</td><td>Write operation settings(1)</td><td>
</td></tr>
<tr><td colspan="2">▶Save 1 word of MW100 on the client side in the word write area M1 on the server side
1. Channel: Select the channel 2 set as the Modbus RTU client in P2P channel setting.
2. P2P function: Select WRITE to perform the write operation.
3. Start condition: Use special flag F146 to transmit frame every 200㎳.
4. Method, data type: Select Individual or Word for I word.
5. Partner station number: Select station No. 1 on the server side.
7. Settings: Set the read area and storage area and click OK.
1) Read area: Device address of data stored on the client side(MW100)
2) Storage area: Device address of data stored on the server side(0x40001:M1)
* When all settings are completed, the text color of the index changes to black</td></tr>
</table>

<table>
<tr><td rowspan="2">5</td><td>Write operation settings(2)</td><td>Read area: Local Address (NewPLC)
Save area: Remote Address
<table><tr><th></th><th>Read area</th><th>Save area</th><th>Address</th></tr><tr><td>1</td><td>%MW101</td><td>0x4002</td><td>N00074</td></tr></table></td></tr>
<tr><td colspan="2">▶Save 4 words from MW101 on the client side in the word write area from M2 to M5 on the server side
1. Channel, P2P function, start address and partner station number: The same as in the sequence No. 4.
2. Method, data type: Select Continuous, Word since it is continuous 4-word writing.
3. Data size: Enter 4 since it is 4 words.
4. Settings: Set the read area and storage area and then click the OK button.
(1) Read area: Device address of data stored on the client side(MW101)
(2) Storage area: Device address of data to be stored in the server station(0x40002: M2)</td></tr>
<tr><td rowspan="2">6</td><td>Write operation settings(3)</td><td>Read area: Local Address (NewPLC)
Save area: Remote Address
<table><tr><th></th><th>Read area</th><th>Save area</th><th>Address</th></tr><tr><td>1</td><td>%MX1</td><td>0x0020</td><td>N00123</td></tr></table></td></tr>
<tr><td colspan="2">▶Save MX1 bit value on the client side in the bit write area M20 on the server side
1. Channel, P2P function, start condition, method and partner station number: The same as in the sequence 4.
2. Data type: Select Bit.
3. Settings: Set the read area and storage area and then click the OK button.
(1) Read area: Device area of data stored on the client side(MX1)
(2) Storage area: Device address of data to be stored in the server station (0x00020: M20)</td></tr>
<tr><td rowspan="2">7</td><td>Write operation settings(4)</td><td>Read area: Local Address (NewPLC)
Save area: Remote Address
<table><tr><th></th><th>Read area</th><th>Save area</th><th>Address</th></tr><tr><td>1</td><td>%MX2</td><td>0x0021</td><td>N00172</td></tr></table></td></tr>
<tr><td colspan="2">▶Save bit values ranging from MX2 to MX17 on the client side in the bit write area from M21 to M36 on the server side
1. Channel, P2P function, start condition and data type: The same as in the sequence No. 6.
2. Method: Select Continuous.
3. Data size: Enter 5.
4. Settings: Set the read area and storage area and then click the OK button.
(1) Read area: Device area of data stored on the client side(MX2)
(2) Storage area: Device address of data to be stored in the server station (0x00020: M21)</td></tr>
</table>

<table>
<tr><th>Sequence</th><th>Setting process</th><th>Setting method</th></tr>
<tr><td rowspan="2">8</td><td>Read operation settings(1)</td><td>Read area: Remote Address
Save area: Local Address (NewPLC)<table><tr><th></th><th>Read area</th><th>Save area</th><th>Address</th></tr><tr><td>1</td><td>0x30002</td><td>%MW160</td><td>N00197</td></tr></table></td></tr>
<tr><td colspan="2">▶Save 1 word value of M2 in the word read area on the server side in MW160 on the client side
1. Channel, start condition, method, data type and partner station number: The same as in the sequence No.4
2. P2P function: Select READ.
3. Settings: Set the read area and storage area and then click the OK button.
(1) Read area: Device area of data stored on the client side(0x30002)
(2) Storage area: Device address of data to be stored in the client (MW160)</td></tr>
<tr><td rowspan="2">9</td><td>Read operation settings(2)</td><td>Read area: Remote Address
Save area: Local Address (NewPLC)<table><tr><th></th><th>Read area</th><th>Save area</th><th>Address</th></tr><tr><td>1</td><td>0x30000</td><td>%MW150</td><td>N00246</td></tr></table></td></tr>
<tr><td colspan="2">▶Save 4 words from M0 of the read area on the server side in the address from MW150 to MW153 on the client side
1. Channel, function, start condition, method, data type and partner station number: The same as in the sequence No. 8
2. Method: Select Continuous
3. Settings: Set the read area and storage area and then click the OK button.
(1) Read area: Device area of data stored on the server side(0x30000)
(2) Storage area: Device address of data to be stored in the client (MW150)</td></tr>
<tr><td rowspan="2">10</td><td>Read operation settings(3)</td><td>Read area: Remote Address
Save area: Local Address (NewPLC)<table><tr><th></th><th>Read area</th><th>Save area</th><th>Address</th></tr><tr><td>1</td><td>0x10010</td><td>%MX170</td><td>N00295</td></tr></table></td></tr>
<tr><td colspan="2">▶Save the bit value of M10 in the bit read area on the server side in the 1th bit of MX170 on the client side
1. Channel, function, start condition, method and partner station number: The same as in the sequence No. 8.
2. Data type: Select Bit.
3. Settings: Set the read area and storage area and then click the OK button.
(1) Read area: Device address of data stored on the server side(0x10010)
(2) Storage area: Device address of data to be stored in the client(MX170)</td></tr>
<tr><td rowspan="2">11</td><td>Read operation settings(4)</td><td>Read area: Remote Address
Save area: Local Address (NewPLC)<table><tr><th></th><th>Read area</th><th>Save area</th><th>Address</th></tr><tr><td>1</td><td>0x10010</td><td>%MX180</td><td>N00344</td></tr></table></td></tr>
<tr><td colspan="2">▶Save the bit value of M10 in the bit read area from M10 to M25 on the server side in the bit of MX180 to MX195 on the client side
1. Channel, function, start condition, method and partner station number: The same as in the sequence No. 10.
2. Method: Select Continuous
3. Settings: Set the read area and storage area and then click the OK button.
(1) Read area: Device address of data stored on the server side(0x10010)
(2) Storage area: Device address of data to be stored in the client(MX180)</td></tr>
<tr><th>Seque</th><th>Setting</th><th>Setting method</th></tr>
</table>

nce	process	
12	Writing parameters	
	1. Select Online ->Write or click the icon ( ). 2. Click OK. 3. Click the OK button to complete the parameter write and then reset the module.	
13	Link Enable	
	1. Select Online ->Communication Module Settings ->Link Enable or click the icon( ). 2. Check the P2P whose setting is completed and click Write..	

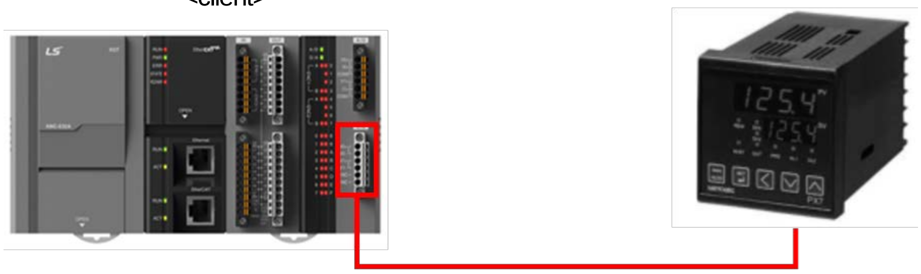
15.12.4 User-Defined Communication Examples

1) When communicating with other products

This chapter explains how to communicate using user frame definition communication.

▶ System configuration

<client> < server >



*RS-485 9600/8/1/NONE/. user-defined communication

In this example, the Cnet I / F and the external device to communicate by user defined communication are as follows.

Division	Basic unit	XMC-E32C	HANYOUNG Temperature controller PX7*Note 2)
	Communication module	Built-in RS-485	
Operation mode	Client		Server
Protocol	User frame definition		PC Link
Communication type	RS-485		RS-485
Communication speed	9,600		9,600
Data bit	8		8
Stop bit	1		1
Parity bit	None		None
Station No.	0		1
Latency*Note 1)	100ms		-
Operation	The current value and set temperature value of the temperature controller area read at intervals of 1 second, and then the current value is stored in M200, and the temperature set value in M210.		

[User-defined communication system configuration]

Note 1) Latency value is set in RS-422/485 communication and is designed to prevent the frame from being broken when communicating with the other device with slow response. The set value varies depending on the other device and distance, and the set value generally ranges from 50 to 100ms.

Note 2) Please refer to the user's manual (http://hynux.com) for details on the communication settings of the temperature controller used in this example.

- PC Link frame structure

The frame structure of PC Link, which is the communication protocol of HANYOUNG temperature controller used in this example, is as follows.

The frame of the temperature controller is executed as an ASCII string, and the contents of the defined D,I Register can be read and written. The type of protocol is divided into the STD standard protocol and the SUM protocol with a checksum added to the standard type. The type of protocol is selected by the parameter of the temperature controller. The standard protocol is "STD", and its structure starts with the start character STX(0x02) and ends with the termination character CR(0x0D) LF(0x0A).

Please refer to the PLC communication example of HANYOUNG NUX (http://hynux.com) for details on the command and data structure.

STX	station number	command	data	CR	LF
0x02	1~99			0x0D	0x0A

[Standard protocol architecture]

STX	station number	command	data	error code	CR	LF
0x02	1~99			Check Sum	0x0D	0x0A

[SUM protocol architecture]

- Exemplary frame creation

This example is when the current value and setting value of the temperature controller are stored in the M device area of the PLC and the table below shows the frames requesting reading of contiguous areas of data and the frames responding to requests to read data.

frame	STX	station No.	DRS	,	number of data	start address of D register	CR	LF
Size (Byte)	1	2	3	1	2	4	1	1

[Request frame]

frame	STX	station No.	DRS	,	OK	,	Data 1	,	Data N	CR	LF
Size (Byte)	1	2	3	1	2	1	4	1	4	1	1

[Request frame]

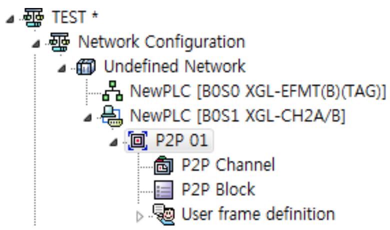
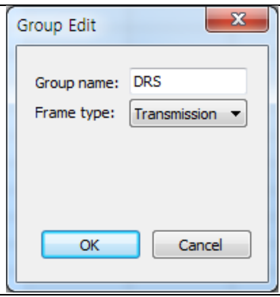
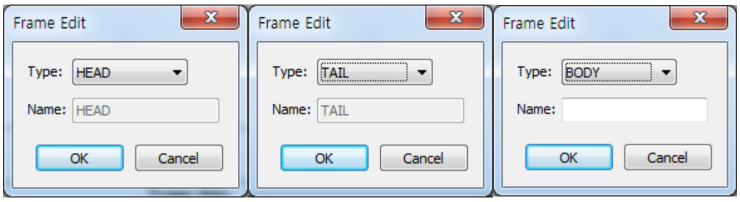
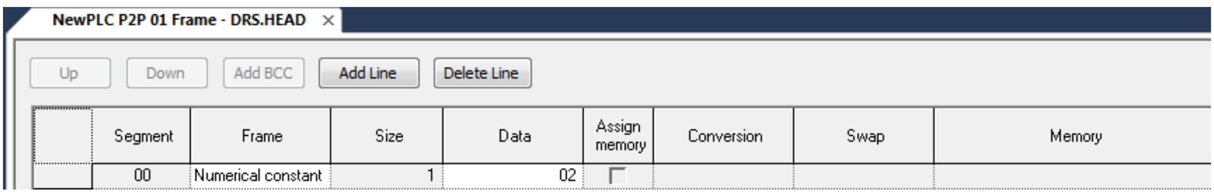
•DRS: It is a command to read the consecutive values of D register, and the number of data to be read on the frame and the start address of D register should be set.
• In the example, the number of data is 2, and 0 is entered as the start address of the current value.

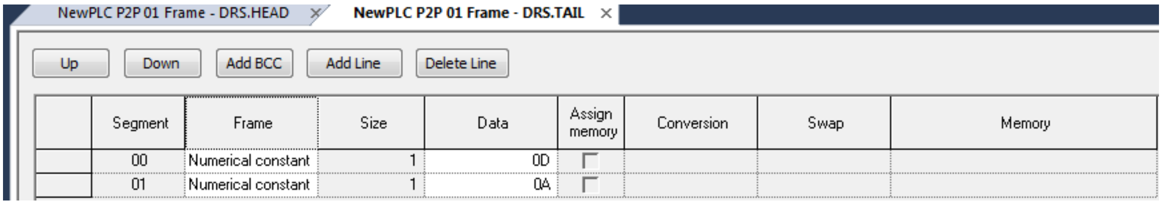
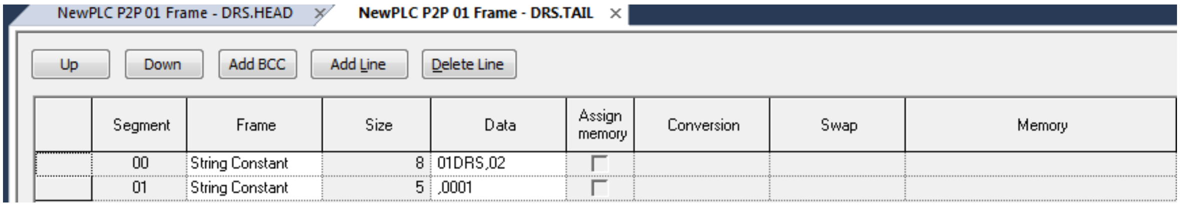
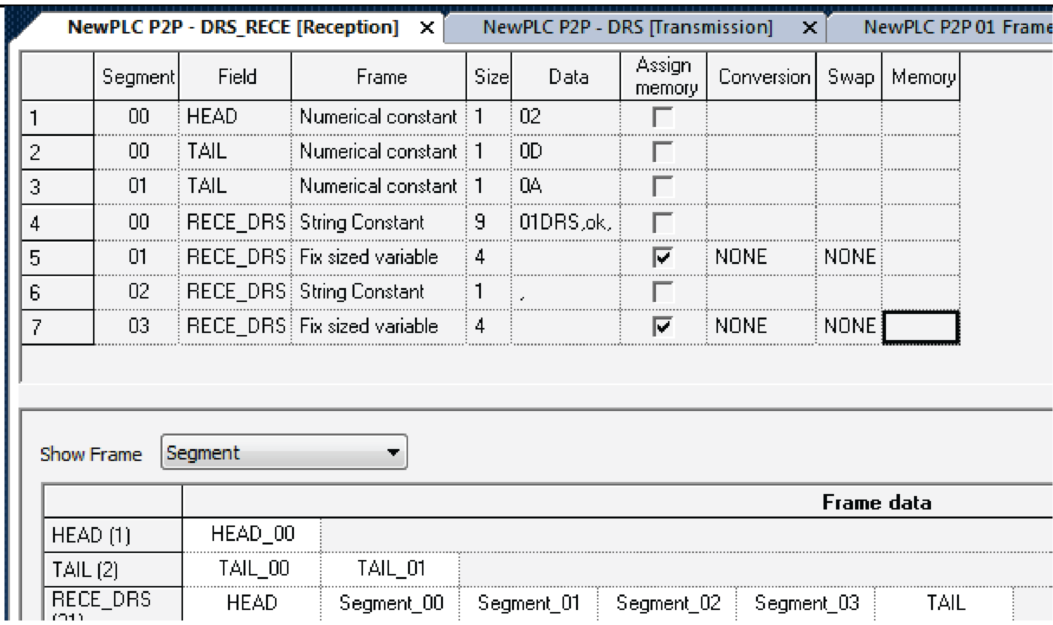
- Default parameter settings

  Refer to 15.12.2 for the basic setting method.

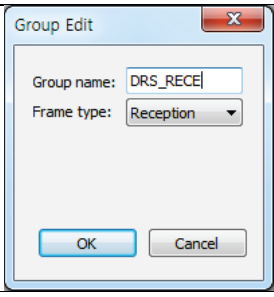
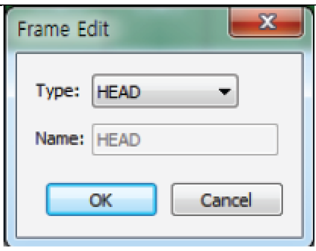
- Creation of data read request frame

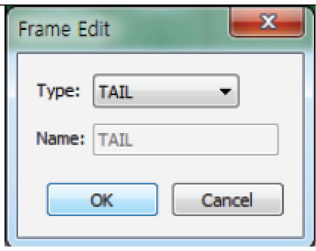
  It describes how to create a frame in XG5000 for user-defined communication.

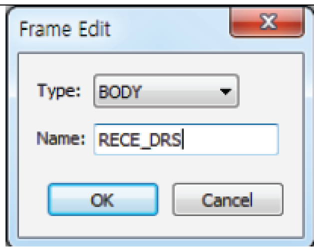
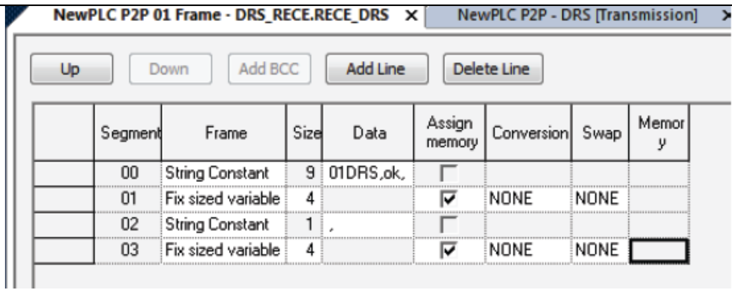
Data read request frame(Transmission frame)	
Sequence	Setting method
1	
	1. After completing the default settings, double click P2P 01 on the P2P screen. 2. Base and slot are set to 0 in built-in communication. Click OK. 3. Double click P2P channel and select the User Frame Definition in Channel 2
2	
	1. Click the User Frame Definition and press the right mouse button. 2. Select Add Group to enter the frame name (DRS) created in the group name and select Transmission in the frame type.
3	
	1. Press the right mouse button in the added DRS group, click Add Frame and add HEAD, TAIL and BODY, respectively. 2. In this example, test is entered as the name of the BODY.
4	
	1. Select the DRS.HEAD tab on the right editor screen and double click the editor window to create the segment setting screen. 2. Enter the numeric constant that represents the Hex value in ASCII code as a type and 2, which is the Hex value that means STX, as data in the segment addition.

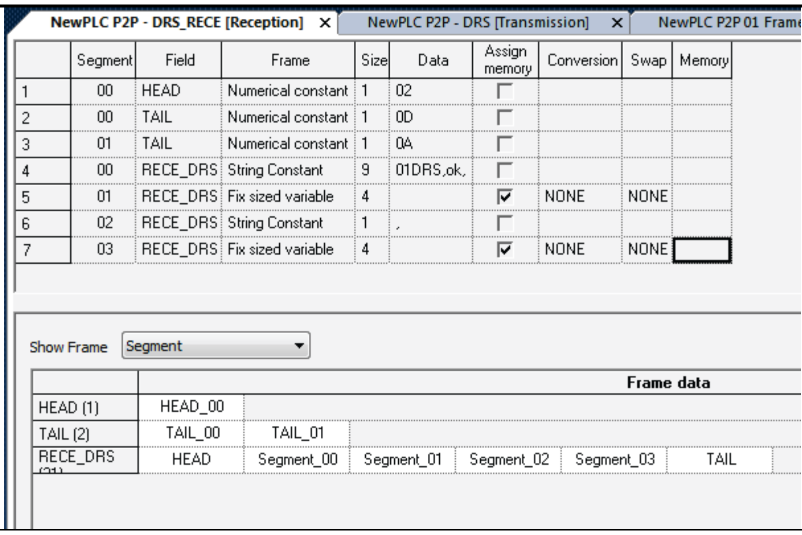
<table>
<tr><th>Sequence</th><th>Setting method</th></tr>
<tr><td rowspan="2">5</td><td>
</td></tr>
<tr><td>1. In the same way, enter the numerical constant that represents the Hex value in ASCII as a type and D,A, which are Hex values that mean CR and LF, as data in DRS.TAIL.</td></tr>
<tr><td rowspan="2">6</td><td>
</td></tr>
<tr><td>1. Double click the DRS.test tab and edit the segment as shown below.
2. Create a frame that makes a request to read the data value of two consecutive areas with No. 1 of D register of the station No. 1 as the start address.
3. When the frame is created through the segment editing by double clicking the editor screen, the size per segment is 10 or less.</td></tr>
<tr><td rowspan="2">7</td><td>
</td></tr>
<tr><td>1. The above shows the result of creating the full frame of the data read request frame.</td></tr>
</table>

▶ Creation of temperature controller response reception frame

<table>
<tr><th colspan="2">Response frame creation(Reception frame)</th></tr>
<tr><td>Sequence</td><td>Setting method</td></tr>
<tr><td rowspan="2">1</td><td>
</td></tr>
<tr><td>1. Frame is created in the same way as in the sequence No. 2 of the data read request frame creation, when the frame type is set to reception.
2. The frame name is DRS_RECE.</td></tr>
<tr><td rowspan="2">2</td><td>

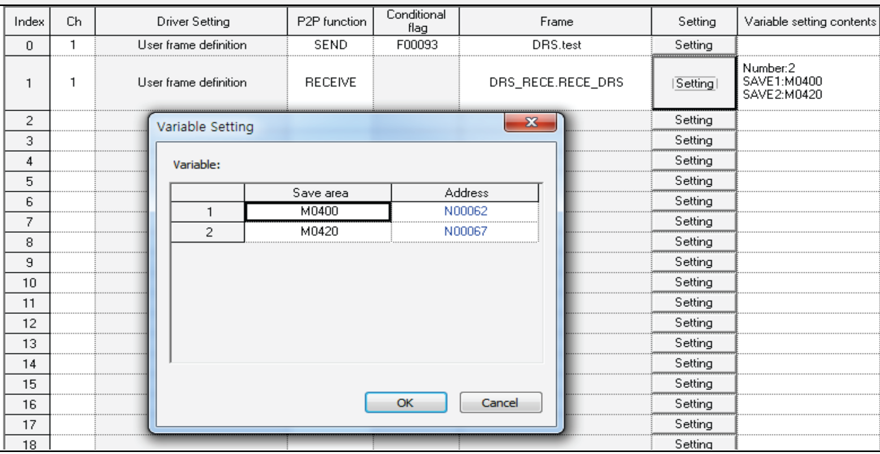



</td></tr>
<tr><td>1. Press the right mouse button in the added DRS_RECE group, click the Add Frame and add HEAD, TAIL, and BODY, respectively.
2. In this example, RECE_DRS is entered as the name of the BODY.</td></tr>
<tr><td>3</td><td>1. Method on how to create HEAD and TAIL is the same as the sequence No. 4 to 5 of the data read request frame.</td></tr>
<tr><td rowspan="2">4</td><td>
</td></tr>
<tr><td>1. As set in the operation of [Table15.10.1], the current temperature value should be saved in MB200, and the temperature setting value in MB210, and therefore the storage area of the 1st and 2nd data should be set.
2. Since the size of data 1 and 2 is 4 byes as identified in the structure of the response frame in the exemplary frame creation, the type is a fixed size variable, and the size is 4 when entered in the segment.
3. Check the Specify Memory to select the storage area of data.</td></tr>
</table>

<table>
<tr><td rowspan="2">5</td><td>
<table>
<tr><td colspan="3">NewPLC P2P - DRS_RECE [Reception]</td><td colspan="3">NewPLC P2P - DRS [Transmission]</td><td colspan="4">NewPLC P2P 01 Fram</td></tr>
<tr><td></td><td>Segment</td><td>Field</td><td>Frame</td><td>Size</td><td>Data</td><td>Assign memory</td><td>Conversion</td><td>Swap</td><td>Memory</td></tr>
<tr><td>1</td><td>00</td><td>HEAD</td><td>Numerical constant</td><td>1</td><td>02</td><td>☐</td><td></td><td></td><td></td></tr>
<tr><td>2</td><td>00</td><td>TAIL</td><td>Numerical constant</td><td>1</td><td>0D</td><td>☐</td><td></td><td></td><td></td></tr>
<tr><td>3</td><td>01</td><td>TAIL</td><td>Numerical constant</td><td>1</td><td>0A</td><td>☐</td><td></td><td></td><td></td></tr>
<tr><td>4</td><td>00</td><td>RECE_DRS</td><td>String Constant</td><td>9</td><td>01DRS,ok,</td><td>☐</td><td></td><td></td><td></td></tr>
<tr><td>5</td><td>01</td><td>RECE_DRS</td><td>Fix sized variable</td><td>4</td><td></td><td>☑</td><td>NONE</td><td>NONE</td><td></td></tr>
<tr><td>6</td><td>02</td><td>RECE_DRS</td><td>String Constant</td><td>1</td><td>,</td><td>☐</td><td></td><td></td><td></td></tr>
<tr><td>7</td><td>03</td><td>RECE_DRS</td><td>Fix sized variable</td><td>4</td><td></td><td>☑</td><td>NONE</td><td>NONE</td><td></td></tr>
</table>
Show Frame: Segment
<table>
<tr><td></td><td colspan="6">Frame data</td></tr>
<tr><td>HEAD (1)</td><td>HEAD_00</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>TAIL (2)</td><td>TAIL_00</td><td>TAIL_01</td><td></td><td></td><td></td><td></td></tr>
<tr><td>RECE_DRS (21)</td><td>HEAD</td><td>Segment_00</td><td>Segment_01</td><td>Segment_02</td><td>Segment_03</td><td>TAIL</td></tr>
</table>
</td></tr>
<tr><td>1. The above shows the result of creating the full frame of response that receives the data responded by the temperature controller.</td></tr>
</table>

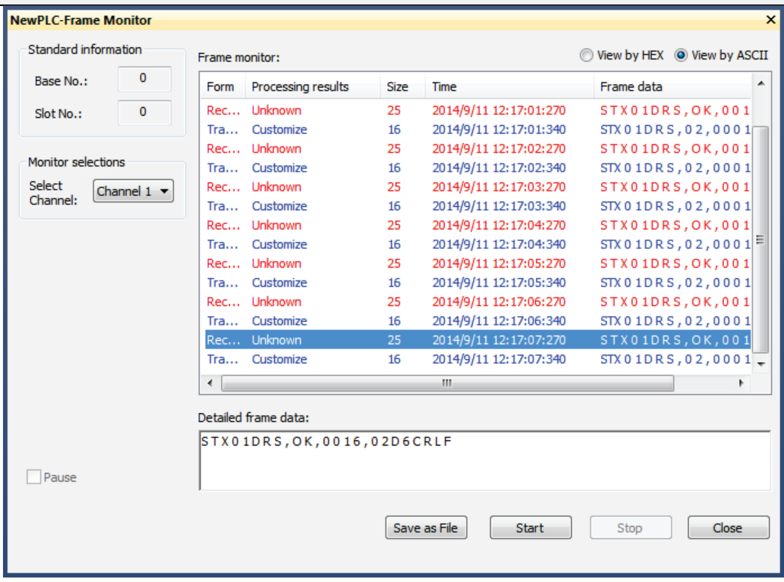
▶ Creation of P2P transmission/reception blocks

P2P transmission and reception blocks are created using the user-defined communication segment created earlier, as shown below.

Segment	Setting method
1	 1. Double click the P2P block of P2P 01. 2. Enter the channel (user frame definition) selected from the P2P channel. 3. Select SEND for the transmission frame and RECEIVE for the reception frame as a P2P function. 4. Start condition is enabled only when the P2P function is SEND. 5. Use F39 that means 1 second since the start condition reads data once every second. 6. Click the reception frame settings to set the storage area of the current temperature value and setting value.
2	Execute the parameter write and Link Enable.

▶Transmitted/received data check

It is designed to check whether the created frame is normally transmitted and received.

Sequence	Setting method
1	 1. Select Online ->Communication Moduel Settings ->System Diagnostics or click the icon( ). 2. Click the module and click the right mouse button. 3. Click [Frame Monitor] or [Service Status] to check the operation status. 4. If the frame is erroneously written, the message [unknown] is displayed in the processing result.
2	Check the data of device area set through the device monitoring of XG-5000.

15.13 Diagnostic Functions

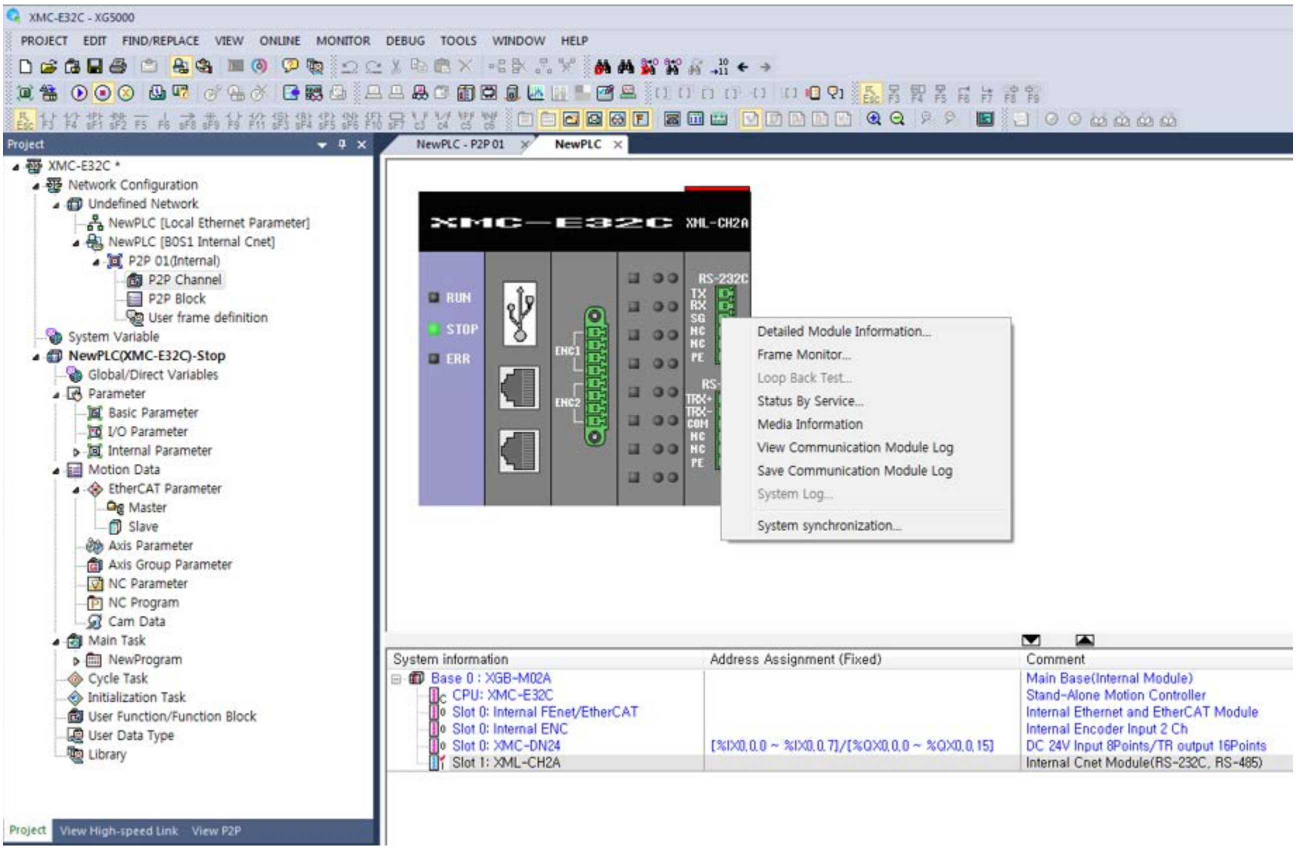
With XG5000, you can check/diagnose the status of network and various systems such as basic unit status, communication module information, service status information and frame monitor, etc.

The available diagnostic functions are as follows.

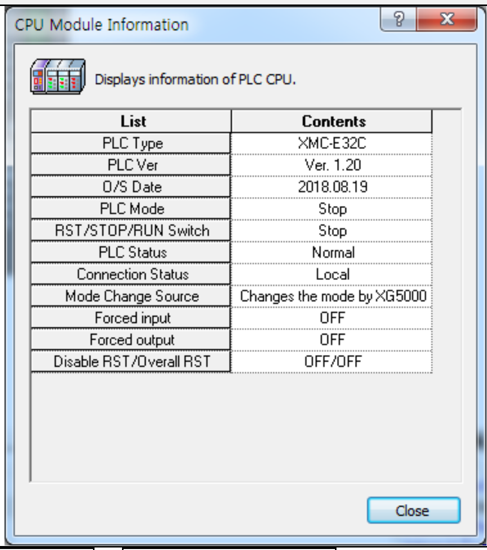
▶ CPU module information
▶ Communication module information
▶ Frame monitor
▶ Service status

15.13.1 Diagnostic Function of XG5000

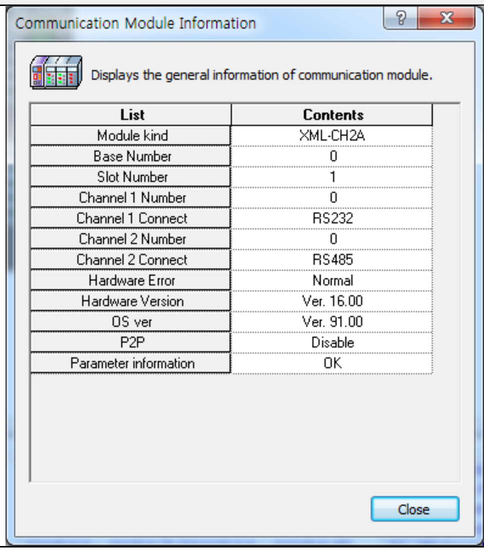
The status check and diagnostic method of the system and network through system diagnostics of XG5000 are as follows. Connect XG5000 to the basic unit and select “Online → Communication Module Setting → System Diagnostics” in the menu, or click the system diagnostics icon ( ) and the following window appears.



▶ **Basic unit status check**

<table>
<tr><th>Checked contents</th><th>Detailed results</th></tr>
<tr><td>CPU Module information</td><td>
</td></tr>
<tr><td colspan="2">1. Select Online>Communication Module Settings ->System Diagnostics or click the icon( ).
2. Click the CPU module and click the right mouse button.
3. Click [CPU Module Information] to check the status of the CPU module</td></tr>
</table>

▶ **Communication module information**

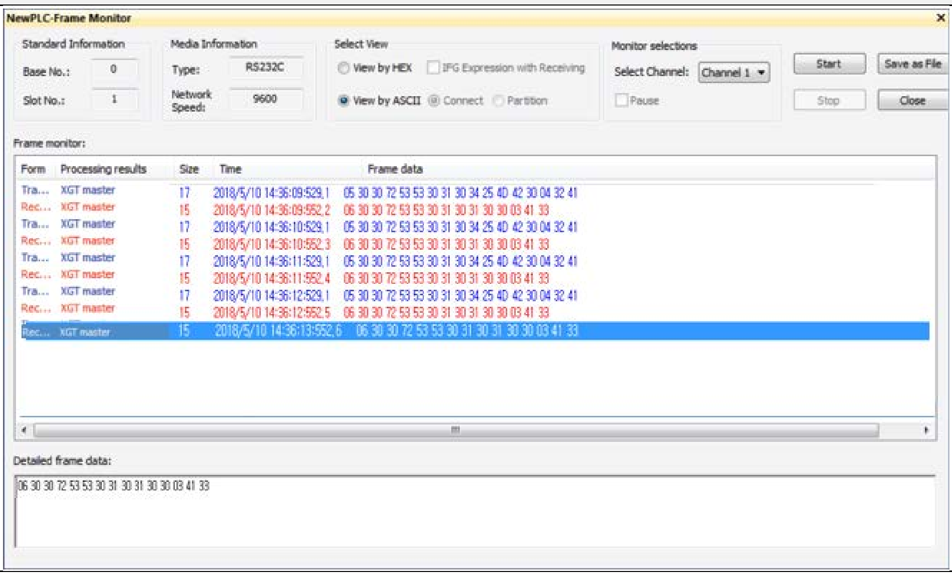
<table>
<tr><th>Checked contents</th><th>Detailed results</th></tr>
<tr><td>Communication module information</td><td>
</td></tr>
<tr><td colspan="2">1. Select Online>Communication Module Setting ->System Diagnostics or click the icon( ).
2. Click the Cnet I / F module and click the right mouse button.
3. Click [Communication module information] to check the status of communication module.</td></tr>
</table>

▶ **Contents of communication module information item**

Items	Contents
Communication module type	Indicate the type of the communication module currently being diagnosed.
Base No.	Indicate the base information of the communication module currently being diagnosed. It is fixed to 0 and displayed in XGB PLC.
Slot No.	Indicate the slot number of the communication module currently being diagnosed. It is fixed to 0 and displayed in built-in communication
Station No.	Station number of the channel used in P2P and dedicated service
Connection method	Information of the communication type (RS-232C, RS-422) of the corresponding channel
Hardware error status	Indicate whether the hardware of the communication module is normal or not.
Hardware version	Version of communication module hardware
OS version	Indicate the version of the communication module OS
P2P	Indicate the P2P communication is enabled/disabled
System parameter information	Whether to download the default communication parameters Display the error information of default communication parameters

▶ **Frame monitor**

The frame monitor of XG5000 allows you to check whether the frame transmitted/received through the Cnet I/F module is normal or not.

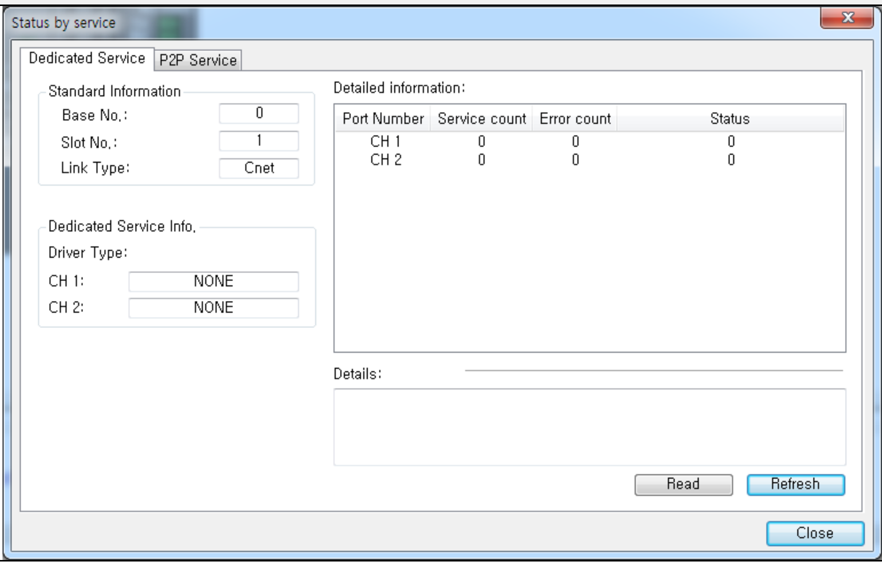
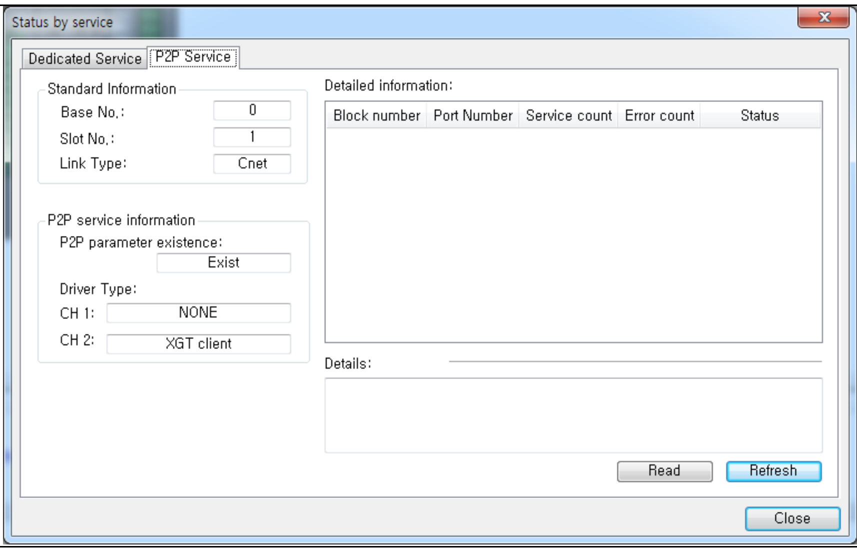
<table>
<tr><th>Checked contents</th><th>Detailed results</th></tr>
<tr><td>Frame monitor</td><td>
</td></tr>
<tr><td colspan="2">1. Select Online>Communication Module Setting ->System Diagnostics or click the icon( ).
2.You can monitor the communication data currently being communcated by cliking on the Cnet I/F module and pressing the right mouse button to click the [Frame Monitor].
3. Click [Frame Monitor] to monitor the communication status.</td></tr>
</table>

▶ Frame Monitor Details

Items		Contents
Basic information	Base No.	Base position of the communication module being monitored

	Slot No.	Slot position of the communication module being monitored
Monitor option	Channel selection	Select the channel to monitor
Frame monitor window	Type	Indicate the transmission frame and reception frame
	Processed results	Indicate the protocol type currently being used 1) XGT server 2) XGT client 3) Modbus server 4) Modbus client 5) User defined 6) Unknown: Frame that cannot be processed
	Size	Length of the monitored frame
	Time	Display the point of time for transmission/reception
	Frame data	Display the data of transmitted/received frame
View in HEX		Display the frame data with HEX values
View in ASCII		Display the frame data with ASCII values
Save file		Save the frame monitoring contents to a file
Start		Start of the frame monitoring operation
Stop		Stop the monitoring status
Close		Close the frame monitor window

▶ Service status

<table>
<tr><th>Checked contents</th><th>Detailed results</th></tr>
<tr><td>Dedicated service</td><td>
</td></tr>
<tr><td colspan="2">1. Select Online>Communication Module Setting ->System Diagnostics or click the icon( )
2. Click on the Cnet I/F, and then press the right mouse button and click the service status.
3. Click [Service Status], and then click [Dedicated Service].
4. Click [Continue Reading] to check the status of each service.</td></tr>
<tr><td>P2P service</td><td>
</td></tr>
<tr><td colspan="2">1. Select Online>Communication Module Setting ->System Diagnostics or click the icon( )
2. Click on the Cnet I/F, and then press the right mouse button and click the service status.
3. Click [Status by Service], and then click [P2P Service].
4. Click [Continue Reading] to check the status of each service.</td></tr>
</table>

▶ Details by service

Division	Item		Contents
Dedicated service	Basic information	Base No.	Base position of the module using the dedicated service
		Slot No.	Slot position of the module using the dedicated service
		Link type	Type of the communication module being used
	Dedicated service information		Indicate the type of drives being used for each channel
	Detailed information window	Port No.	Indicate the channel number
		Service count	Indicate the number of dedicated service communications
		Error count	Indicate the number of errors that occur during the dedicated service communication
		Status	Display the dedicated service communication status
P2P service	Basic information	Base No.	Base position of the module using the P2P service
		Slot No.	Slot position of the module using the P2P service
		Link type	Type of the communication module being used
	P2P service information	Presence of P2P parameters	Indicate whether the P2P parameter is downloaded or not
		Driver type	P2P driver setting information for each channel XGT client/MODBUS client/User definable
	Detailed information	Block No.	Available from 0 to 63 Display only the currently registered block being operated
		Port No.	Indicate the channel number
		Status	Display service execution status information for each block
		Service count	Indicate the number of times each block has been executed since the P2P service was performed
		Error count	Indicate the number of errors that occur during the service
Continuous Read/ Redo	Continuous Read		Check P2P service status information every second
	Restart		Check the P2P service status information at the time of execution

▶ **Service status code**

It is used to check the communication status of Cnet I / F module.

Dedicated service		P2P service	
Status	**Meaning**	**Status**	**Meaning**
0	Normal communication	0	Normal communication
1	Reception frame header error (No ACK/NAK)	4	Maximum station setting error(when setting 0 to 255 stations or more)
2	Reception frame tail error(No Tail)	5	Time out occurs
3	Reception frame BCC error	FFFE	1.Modbus address error 2.When using commands other than Read/Write
9	The station number of the received frame is different from the number of its station (Its station number=0)	-	
0A	No response is received from the CPU		
0B	The received frame is longer than the Modbus maximum frame		
0C	The received frame is not Modbus ASCII / Modbus RTU		
0D	HEX conversion error in Modbus occurs		

15.13.2 Protocol-specific error codes

It is related to the error code according to the protocol.

▶ XGT Client / Server

Error code	Error type	Error content and cause	Example of error frame
0003	Block number excess error	The number of blocks is larger than 16 at individual read/write requests	01rSS1105%MW10…
0004	Variable length error	Variable length is greater than the maximum size (16)	01rSS113%MW10000000000…
0007	Data type error	Data other than X,B,W,D and L is received	01rSS1105%MK10
0011	Data error	Date length area information is incorrect	01rSB05%MW%4
		It does not start with %	01rSS0105$MW10
		The area value of a variable is invalid	01rSS0105%MW^&
		In the case of bit writing, 00 or 01 should be written, but other values are written	01wSS0105$MX1011
0090	Monitor execution error	The execution of unregistered monitor is required	
0190	Monitor execution error	The range of monitor execution exceeds that of registration numbers requested by the client	
0290	Monitor registration error	The range of monitor registration exceeds that of registration numbers requested by the client	
1132	Device memory error	Character, which is not a device to be used, is entered	
1232	Data size error	The size of data exceeds 60 words that can be read or written at a time	01wSB05%MW1040AA5512…
1234	Reserved frame error	There is additional content that is not needed	01rSS0105%MW10000
1332	Data type mismatch error	In the case of individual read/write, all blocks should be requested with respect to the same data type	01rSS0205%MW1005%MB10
1432	Data value error	Data value cannot be converted to Hex value	01wSS0105%MW10AA%5
7132	Variable request area excess error	It is required beyond the area supported by each device.	01rSS0108%MWFFFFFF

▶ Modbus ASCII, Modbus RTU Client / Server

Code	Name	Detailed descriptions
01	Function code error	Function code error
02	Address error	Address allowable range excess error
03	Data setting error	Data value is not allowed
04	Server station error	Server(slave)station is in error state
05	Retransmission request	Requests the client to make a request again at a proper time because the content to be processed is to huge for the server to handle at the moment
06	Processing time delay	The server station takes time to process. The master should make a request again.

▶ P2P Client Error Code

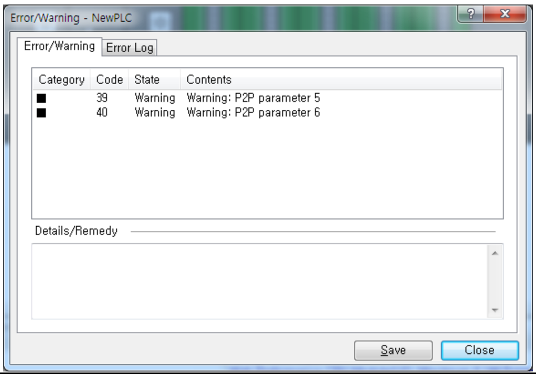
Code	Name	Detailed descriptions
05	P2P block timeout error	Exceeds P2P block reception response time

▶ Error code that occurs in PLC itself

Code	Name	Detailed descriptions
0015	PLC P2P client time out	Timeout error occurs with respect to P2P block in PLC, not communication module. The PLC waits for up to 5 seconds after the P2P requests to the communication module. After 5 seconds, the PLC generates a timeout error with respect to the P2P block.
0016	PLC P2P client error	Invalid device area access error

15.13.3 Troubleshooting

1) Solution when P2P parameter setting error warning occurs when connected to XG5000

Problem	Cause	Solution
P2P setting error warning when connected to XG5000  	P2P number without P2P settings is selected during the Link-Enable of XG5000	1. Check the P2P setting number in the Link-Enable of XG5000, delete the check of wrongly selected P2P number and click Write. 2. After disconnecting the connection of XG5000, connect to XG5000 and check whether the warning disappears.

2) If the communication is not done even though P2P setting is completed when operating as a client

Problem	Cause	Solution
Communication setting is completed, but Tx/Rx of LED of CnetI/F does not blink	Basic unit is in the stop state	Connect to XG5000, check operation mode and convert Stop to Run
	Communication default parameter mismatch between the client and the server	Connect to XG5000, select “File->Open from PLC” and check the communication default setting of the module that acts as client and server
	Link-Enable is not done	After performing P2P parameter, select the Link-Enable of the P2P and execute Write.

3) If the communication type is set to RS-485, and the response frame is missing when operating as a client

Problem	Cause	Solution
Frame monitoring after completing the settings of multiple P2P parameters in P2P block results in missing response frames	P2P start condition is faster than communication time	1. Change the P2P start condition of P2P block setting in consideration of communication time 2. Communication time= Transmission time + reception time • Transmission time= Start condition + basic unit Scan time + communication module Reaction time + data Delivery time • Reception time= Basic unit scan time + communication module reaction time + data delivery time
	Response time of the communication module acting as a server is slow	1.Increase the latency in the default network setting section of XG5000

4) Solution when transmitting and receiving data that cannot be subject to frame analysis

Problem	Cause	Solution
Transmitting and receiving data that cannot be subject to frame analysis	In a multi-drop connection, a plurality of servers transmit data at the same time	1. Execute 1:1 communication with the device acting as a server and check whether it is normal 2. In case of normal communication between all devices, a plurality of servers may transmit data at the same time in a multi-drop configuration. Take the Interlock and be sure not to transmit in this case
	Parity bit setting is not matched	Match the parity bit setting between Cnet I/F and device acting as a server
	Length of the stop bit is not set correctly	Match the stop bit setting between Cnet I/F and device acting as a server
	Communication speed setting is not matched	Match the communication speed setting between Cnet I/F and device acting as a server
	In multi-drop, termination resistance is not set	Check whether the termination resistor between Cnet I/F and device acting as a server is installed or not.

5) If it is unclear whether the error is caused by the client or the device operating as a server

Problem	Cause	Solution
It is unclear whether the communication error is caused by the client or the device acting as a server	-	1.Check Cnet I/F module - Check the installation status of the module - Check the wiring status 2.Check the basic unit status

6) Solution when normal or abnormal communication repeatedly occurs

Problem	Cause	Solution
Normal or abnormal communication repeatedly occurs	In a multi-drop connection, a plurality of servers transmit data at the same time	1. Execute 1:1 communication with the device acting as a server and check whether it is normal 2. In case of normal communication between all devices, a plurality of servers may transmit data at the same time in a multi-drop configuration. Take the Interlock and be sure not to transmit in this case
	Wiring connection failure of communication line	Replace cable or fix the cable connection securely
	Timing mismatch between transmission and reception signals when communication is set by half-duplex communication(RS-485)	Increase the latency of the device acting as a client and a server
	1. When the transmission processing is not completed, the following transmission process is required. 2. When the reception processing is not completed, the following reception process is required	Ensure the handshaking in the program

Appendix 1 Flag List

(1) Type of flag

(a) System flag

This flag indicates the operation, state, and information of motion controller.

Variable	Type	Address	Description
_SYS_STATE	DWORD	%FD0	PLC mode and states
_RUN	BOOL	%FX0	RUN
_STOP	BOOL	%FX1	STOP
_ERROR	BOOL	%FX2	ERROR
_LOCAL_CON	BOOL	%FX4	Local control
_REMOTE_CON	BOOL	%FX6	Remote mode ON
_RUN_EDIT_ST	BOOL	%FX8	Downloading a program at online editing mode
_RUN_EDIT_CHK	BOOL	%FX9	Processing online editing internally
_RUN_EDIT_DONE	BOOL	%FX10	Online editing done
_RUN_EDIT_NG	BOOL	%FX11	Online editing abnormal termination
_CMOD_KEY	BOOL	%FX12	Change operation mode by the switch
_CMOD_LPADT	BOOL	%FX13	Change operation mode by the local PADT
_FORCE_IN	BOOL	%FX16	Force input
_FORCE_OUT	BOOL	%FX17	Force output
_MON_ON	BOOL	%FX20	Monitoring mode
_USTOP_ON	BOOL	%FX21	STOP by STOP Function
_ESTOP_ON	BOOL	%FX22	STOP by ESTOP Function
_INIT_RUN	BOOL	%FX24	Executing the initial task
_PB1	BOOL	%FX28	Program code 1
_PB2	BOOL	%FX29	Program code 2
_CNF_ER	DWORD	%FD2	System errors(Significant error)
_ANNUM_ER	BOOL	%FX70	Significant error detection in external device
_BPRM_ER	BOOL	%FX72	Basic parameter error
_IOPRM_ER	BOOL	%FX73	IO configuration parameter error
_SPPRM_ER	BOOL	%FX74	Parameter error in Special module
_CPPRM_ER	BOOL	%FX75	Local Ethernet parameter error
_PGM_ER	BOOL	%FX76	Program error
_SWDT_ER	BOOL	%FX78	CPU abnormal ends
_ENCPRM_ER	BOOL	%FX85	Encoder parameter error
_AXISPRM_ER	BOOL	%FX86	Axis parameter error
_GROUPPRM_ER	BOOL	%FX87	Axis group parameter error
_ECPRM_ER	BOOL	%FX88	EtherCAT parameter error
_NCPRM_ER	BOOL	%FX89	NC Parameter Error

Variable	Type	Address	Description
_NCPGM_ER	BOOL	%FX90	NC Program Check Error
_PTASK_CYCLE_ER	BOOL	%FX91	Main Task Period Error
_CTASK_CYCLE_ER	BOOL	%FX92	Cycle Task Period Error
_SYSTEM_ER	BOOL	%FX93	System Error
_TASK_PRM_USAGE_OVER_ER	BOOL	%FX94	Task Program Occupancy Excess Error
_CNF_WAR	DWORD	%FD4	System warnings(Minor error)
_RTC_ER	BOOL	%FX128	Abnormal RTC data
_PTASK_CYCLE_WAR	BOOL	%FX129	Main Task Period Exceeded Warning
_CTASK_CYCLE_WAR	BOOL	%FX130	Cycle Task Period Exceeded Warning
_AB_SD_ER	BOOL	%FX131	Stop from abnormal operation
_MOTION_CONTROL_WAR	BOOL	%FX132	Motion Control Abnormal Warning
_ANNUM_WAR	BOOL	%FX134	Minor error detection in external device
_TASK_PRM_USAGE_OVER_WAR	BOOL	%FX135	Task Program Occupancy Excess Warning
_P2P_WAR	BOOL	%FX138	P2P Parameter Warning
_T20MS	BOOL	%FX192	20ms CLOCK
_T100MS	BOOL	%FX193	100ms CLOCK
_T200MS	BOOL	%FX194	200ms CLOCK
_T1S	BOOL	%FX195	1s CLOCK
_T2S	BOOL	%FX196	2s CLOCK
_T10S	BOOL	%FX197	10s CLOCK
_T20S	BOOL	%FX198	20s CLOCK
_T60S	BOOL	%FX199	60s CLOCK
_ON	BOOL	%FX201	Always ON
_OFF	BOOL	%FX202	Always OFF
_1ON	BOOL	%FX203	1 scan ON
_1OFF	BOOL	%FX204	1 scan OFF
_STOG	BOOL	%FX205	Every scan Toggle
_ERR	BOOL	%FX224	Calculation error flag
_ALL_OFF	BOOL	%FX227	All output OFF
_LER	BOOL	%FX229	Latch flag for calculation error
_ARY_IDX_ERR	BOOL	%FX247	Exceeding error from Index range when using array
_ARY_IDX_LER	BOOL	%FX248	Latch for exceeding error on Index range when using array
_UDF_STACK_ERR	BOOL	%FX249	UDF Stack Over Error Flag
_UDF_STACK_LER	BOOL	%FX250	UDF Stack Over Error Latch Flag
_CPU_TYPE	WORD	%FW18	CPU type (XMC-E08A: 0xA421, XMC-E16A: 0xA422, XMC-E32A: 0xA425, XMC-32C: 0xA426, LSMMT-E32A: 0xA429, LSMMT-E32C: 0xA42A)
_CPU_VER	WORD	%FW19	CPU version
_OS_VER	DWORD	%FD10	OS version
_OS_DATE	DWORD	%FD11	OS date
_OS_VER_PATCH	DWORD	%FD12	OS patch version

Variable	Type	Address	Description
_RTC_TIME	ARRAY[0..7] OF BYTE	%FB52	RTC Time
_RTC_DATE	DATE	%FW30	Current RTC date
_RTC_WEEK	UINT	%FW31	Current RTC day
_RTC_TOD	TIME_OF_DAY	%FD16	Current time of RTC(ms unit)
_KEY	DWORD	%FD17	Current state of the local key switch
_AC_F_CNT	UINT	%FW36	Short power interruptions count
_FALS_NUM	UINT	%FW37	FALS Command Usage Area
_SYS_ERR_TYPE	WORD	%FW38	System Error Detailed Flag
_ENCODER_HW_ERR	BOOL	%FX608	Encoder Input Handling H/W Setting Error
_BACKPLANE_IF_ERR	BOOL	%FX609	Backplane Interface Error
_SERIAL_NUM	ARRAY[0..19] OF BYTE	%FB80	Serial Number
_PTASK_SCAN_MAX	UINT	%FW512	Main Task Max. Scan Time(Unit:100us)
_PTASK_SCAN_MIN	UINT	%FW513	Main Task Min. Scan Time(Unit:100us)
_PTASK_SCAN_CUR	UINT	%FW514	Main Task Current Scan Time(Unit:100us)
_CTASK_SCAN_MAX	UINT	%FW515	Cycle Task Max. Scan Time(Unit:100us)
_CTASK_SCAN_MIN	UINT	%FW516	Cycle Task Min. Scan Time(Unit:100us)
_CTASK_SCAN_CUR	UINT	%FW517	Cycle Task Current Scan Time(Unit:100us)
_PROGRAM_RATIO_MAX	UINT	%FW518	User Program Maximum Execution Occupancy (1sec)
_PROGRAM_RATIO_MIN	UINT	%FW519	User Program Minimum Execution Occupancy (1sec)
_PROGRAM_RATIO_CUR	UINT	%FW520	User Program Current Execution Occupancy (1sec)
_PTASK_CYCLE_WAR_NUM	UINT	%FW748	Main Task Period Exceeded Warning Count
_CTASK_CYCLE_WAR_NUM	UINT	%FW749	Cycle Task Period Exceeded Warning Count
_RTC_WR	BOOL	%FX20480	User RTC Setting Request
_CHK_ANC_ERR	BOOL	%FX20482	Request for significant error detection in external device
_CHK_ANC_WAR	BOOL	%FX20483	Request for minor error detection in external device
_PTASK_SCAN_WR	BOOL	%FX20486	Main Task Scan Value Initialization
_CTASK_SCAN_WR	BOOL	%FX20487	Cycle Task Scan Value Initialization
_INIT_DONE	BOOL	%FX20496	Completion of initialization task
_ANC_ERR	WORD	%FW1282	Significant error information in external device
_ANC_WAR	WORD	%FW1283	Minor error information in external device
_RTC_TIME_USER	ARRAY[0..7] OF BYTE	%FB2568	User RTC Time

(b) Motion flag

The flag displayed following are as follows. It displays the state and data of the motion controller.

The flag related to axis is displayed as “_AXxx_...”(xx indicates the relevant axis No. : Decimal) and the flag related to axis group is displayed as “_AGyy_...”(yy indicates the axis group No. : Decimal).

1) Motion common flag

Variable	Type	Address	Description
_MC_RUN	BOOL	%FX65536	MC RUN
_MC_STOP	BOOL	%FX65537	MC STOP
_MC_TEST	BOOL	%FX65538	MC TEST
_MC_WARNING	BOOL	%FX65539	MC Common warning occurrence
_MC_ALARM	BOOL	%FX65540	MC Common alarm occurrence
_MC_COM_ERR	BOOL	%FX65541	MC Common error occurrence
_MC_COM_ERR_CODE	WORD	%FW4097	MC Common error code

2) Motion axis flag

The address information is the flag memory of axis 01. The address has 2,048bit (32LREAL) offsets per axis.

Variable	Type	Address	Description
_AXxx_RDY	BOOL	%FX73728	Axis xx ready
_AXxx_WARNING	BOOL	%FX73729	Axis xx warning occurrence
_AXxx_ALARM	BOOL	%FX73730	Axis xx alarm occurrence
_AXxx_SV_ON	BOOL	%FX73731	Axis xx servo On/Off
_AXxx_SV_RDY	BOOL	%FX73732	Axis xx servo ready
_AXxx_MSTSLV_STS	BOOL	%FX73733	Axis xx master/slave status
_AXxx_NC	BOOL	%FX73734	Axis xx NC operation
_AXxx_MST_INFO	UINT	%FW4609	Axis xx master axis information
_AXxx_AXIS_TYPE	UINT	%FW4610	Axis xx axis type
_AXxx_LINKED_NODE	UINT	%FW4611	Axis xx connected node information
_AXxx_LINKED_SLOT	UINT	%FW4612	Axis xx connected slot information
_AXxx_UNIT	UINT	%FW4613	Axis xx axis unit
_AXxx_VEL_UNIT	UINT	%FW4614	Axis xx speed unit
_AXxx_AX_ERR	WORD	%FW4615	Axis xx error code
_AXxx_SVON_INCMPL	BOOL	%FX73856	Axis xx servo on incomplete
_AXxx_COMM_WARN	BOOL	%FX73857	Axis xx communication warning
_AXxx_DEV_WARN	BOOL	%FX73858	Axis xx deviation warning
_AXxx_SV_ERR	BOOL	%FX73872	Axis xx servo drive error
_AXxx_HW_POT	BOOL	%FX73873	Axis xx positive limit detection
_AXxx_HW_NOT	BOOL	%FX73874	Axis xx negative limit detection
_AXxx_SW_POT	BOOL	%FX73875	Axis xx S/W positive limit detection
_AXxx_SW_NOT	BOOL	%FX73876	Axis xx S/W negative limit detection
_AXxx_SV_OFF	BOOL	%FX73877	Axis xx execution error of operation command in servo-off state
_AXxx_POS_OVR	BOOL	%FX73878	Axis xx exceeds the set range of positioning travel amount
_AXxx_VEL_OVR	BOOL	%FX73879	Axis xx exceeds the maximum velocity
_AXxx_DEV_ERR	BOOL	%FX73880	Axis xx deviation alarm
_AXxx_HOME_INCMPL	BOOL	%FX73881	Axis xx Execution of absolute position command in undetermined HOME

Variable	Type	Address	Description
_AXxx_COMM_ERR	BOOL	%FX73882	Axis xx communication alarm (EtherCAT maintenance lost communication)
_AXxx_BUSY	BOOL	%FX73888	Axis xx busy state of motion command
_AXxx_PAUSE	BOOL	%FX73889	Axis xx pause state of motion command (velocity is zero)
_AXxx_STOP	BOOL	%FX73890	Axis xx stop state by the stop command
_AXxx_CMD_FAIL	BOOL	%FX73891	Axis xx abnormal completion of motion command
_AXxx_CMD_CMPL	BOOL	%FX73892	Axis xx normal completion of motion command
_AXxx_DIR	BOOL	%FX73893	Axis xx operation direction
_AXxx_JOG	BOOL	%FX73894	Axis xx JOG operation
_AXxx_HOME	BOOL	%FX73895	Axis xx Homing operation
_AXxx_POS_CTRL	BOOL	%FX73896	Axis xx position control operation
_AXxx_VEL_CTRL	BOOL	%FX73897	Axis xx velocity control operation
_AXxx_TRQ_CTRL	BOOL	%FX73898	Axis xx torque control operation
_AXxx_LINTP	BOOL	%FX73899	Axis xx linear interpolation operation
_AXxx_CINTP	BOOL	%FX73900	Axis xx circular interpolation operation
_AXxx_SYNC	BOOL	%FX73901	Axis xx synchronous control operation
_AXxx_COORD	BOOL	%FX73902	Axis xx coordinated operation
_AXxx_BUSY_ACC	BOOL	%FX73917	Axis xx acceleration operation
_AXxx_BUSY_CONSTVEL	BOOL	%FX73918	Axis xx constant speed operation
_AXxx_BUSY_DEC	BOOL	%FX73919	Axis xx deceleration operation
_AXxx_POS_CMPL	BOOL	%FX73920	Axis xx positioning completion
_AXxx_INPOS	BOOL	%FX73921	Axis xx inposition detection
_AXxx_LATCH_CMPL	BOOL	%FX73922	Axis xx latch completion
_AXxx_HOME_CMPL	BOOL	%FX73923	Axis xx homing completion
_AXxx_Disabled	BOOL	%FX73936	Axis xx Disabled state
_AXxx_Standstill	BOOL	%FX73937	Axis xx Standstill state
_AXxx_Discrete	BOOL	%FX73938	Axis xx Discrete state
_AXxx_Continuous	BOOL	%FX73939	Axis xx Continuous state
_AXxx_Synchronized	BOOL	%FX73940	Axis xx Synchronized state
_AXxx_Homing	BOOL	%FX73941	Axis xx Homing state
_AXxx_Stopping	BOOL	%FX73942	Axis xx Stopping state
_AXxx_ErrorStop	BOOL	%FX73943	Axis xx ErrorStop state
_AXxx_CMD_TPOS	LREAL	%FL1156	Axis xx target position
_AXxx_CMD_CPOS	LREAL	%FL1157	Axis xx command position of current scan
_AXxx_CMD_VEL	LREAL	%FL1158	Axis xx command velocity
_AXxx_CMD_ACCDEC	LREAL	%FL1159	Axis xx command acceleration/deceleration
_AXxx_CMD_JERK	LREAL	%FL1160	Axis xx command jerk
_AXxx_CMD_TRQ	LREAL	%FL1161	Axis xx command torque

Variable	Type	Address	Description
_AXxx_ACT_POS	LREAL	%FL1162	Axis xx actual current position
_AXxx_ACT_VEL	LREAL	%FL1163	Axis xx actual current velocity
_AXxx_ACT_TRQ	LREAL	%FL1164	Axis xx actual current torque
_AXxx_POS_DEV	LREAL	%FL1165	Axis xx position deviation
_AXxx_DRV_ALARM	BOOL	%FX74624	Axis xx drive alarm state
_AXxx_DRV_WARNING	BOOL	%FX74625	Axis xx drive warning state
_AXxx_DRV_SV_ON	BOOL	%FX74626	Axis xx servo on status
_AXxx_DRV_POT	BOOL	%FX74627	Axis xx positive limit input
_AXxx_DRV_NOT	BOOL	%FX74628	Axis xx negative limit input
_AXxx_DRV_HOME	BOOL	%FX74629	Axis xx home input
_AXxx_DRV_LATCH1	BOOL	%FX74630	Axis xx LATCH1 input
_AXxx_DRV_LATCH2	BOOL	%FX74631	Axis xx LATCH2 input
_AXxx_DRV_PARAMBUSY	BOOL	%FX74632	Axis xx read/write operations of the SDO parameter
_AXxx_DRV_IN	DWORD	%FD2333	Axis xx drive inputs
_AXxx_DRV_ERR	WORD	%FW4668	Axis xx drive error code
_AXxx_CMDBUF_FULL	BOOL	%FX73951	Axis xx Buffered full of command buffers
_AXxx_CMDBUF_QUEUED	UINT	%FW4622	Axis xx Buffered number of command execution wait
_AXxx_CMDBUF_FREE	UINT	%FW4623	Axis xx Buffered number of executable commands

Reference) The flags of _AXxx_HOME(Flag used at home return command) and _AXxx_Homing(Operation status of PLC open standard) indicate the same state.

3) Motion axis group flag

The address information is the flag memory of axis 01. The address has 5,120bit (80LREAL) offsets per axis.

Variable	Type	Address	Description
_AGxx_RDY	BOOL	%FX212992	Axis group xx ready
_AGxx_WARNING	BOOL	%FX212993	Axis group xx warning occurrence
_AGxx_ALARM	BOOL	%FX212994	Axis group xx alarm occurrence
_AGxx_SV_ON	BOOL	%FX212995	Axis group xx servo On/Off
_AGxx_SV_RDY	BOOL	%FX212996	Axis group xx servo ready
_AGxx_ERR	WORD	%FW13313	Axis group xx error code
_AGxx_BUSY	BOOL	%FX213024	Axis group xx busy state of motion command
_AGxx_PAUSE	BOOL	%FX213025	Axis group xx pause state of motion command (velocity is zero)
_AGxx_STOP	BOOL	%FX213026	Axis group xx stop state by the stop command
_AGxx_CMD_FAIL	BOOL	%FX213027	Axis group xx command error exit status
_AGxx_CMD_CMPL	BOOL	%FX213028	Axis group xx command execution complete
_AGxx_LINTP	BOOL	%FX213029	Axis group xx linear interpolation operation
_AGxx_CINTP	BOOL	%FX213030	Axis group xx circular interpolation operation
_AGxx_HOME	BOOL	%FX213031	Axis group xx homing operation

_AGxx_SYNC	BOOL	%FX213032	Axis group xx synchronization operation
_AGxx_TLINTP	BOOL	%FX213033	Axis group xx coordinated time operation
_AGxx_CDMOVE	BOOL	%FX213034	Axis group xx coordinated direct operation
_AGxx_CCINTP	BOOL	%FX213035	Axis group xx coordinated circular interpolation operation
_AGxx_POS_CMPL	BOOL	%FX213056	Axis group xx positioning completion
_AGxx_Disabled	BOOL	%FX213072	Axis group xx Disabled state
_AGxx_Standby	BOOL	%FX213073	Axis group xx Standby state
_AGxx_Moving	BOOL	%FX213074	Axis group xx Moving state
_AGxx_Homing	BOOL	%FX213075	Axis group xx Homing state
_AGxx_Stopping	BOOL	%FX213076	Axis group xx Stopping state
_AGxx_ErrorStop	BOOL	%FX213077	Axis group xx ErrorStop state
_AGxx_CMDBUF_FULL	BOOL	%FX213087	Axis group xx Buffered Command buffer full.
_AGxx_CMDBUF_QUEUED	UINT	%FW13318	Axis group xx Buffered Command waiting number
_AGxx_CMDBUF_FREE	UINT	%FW13319	Number of axis group xx Buffered commands that can be executed
_AGxx_CMD_TPOS	ARRAY[0..9] OF LREAL	%FL3330	Axis group xx target position
_AGxx_CMD_CPOS	ARRAY[0..9] OF LREAL	%FL3340	Axis group xx command position of current scan
_AGxx_CMD_VEL	LREAL	%FL3350	Axis group xx target velocity
_AGxx_CMD_ACCDEC	LREAL	%FL3351	Axis group xx command acc./dec.
_AGxx_CMD_JERK	LREAL	%FL3352	Axis group xx command jerk
_AGxx_ACT_POS	ARRAY[0..9] OF LREAL	%FL3353	Axis group xx actual current position
_AGxx_ACT_VEL	LREAL	%FL3363	Axis group xx actual current velocity
_AGxx_CFG_AX_NUM	UINT	%FW13456	Axis group xx number of axes
_AGxx_CMDBUF_FULL	BOOL	%FX213087	Axis group xx Buffered full of command buffers
_AGxx_CMDBUF_QUEUED	UINT	%FW13318	Axis group xx Buffered number of command execution wait
_AGxx_CMDBUF_FREE	UINT	%FW13319	Axis group xx Buffered number of executable commands
_AGxx_CFG_A1	UINT	%FW13458	Axis group xx axis number of composition axis1
_AGxx_CFG_A2	UINT	%FW13459	Axis group xx axis number of composition axis2
_AGxx_CFG_A3	UINT	%FW13460	Axis group xx axis number of composition axis3
_AGxx_CFG_A4	UINT	%FW13461	Axis group xx axis number of composition axis4
_AGxx_CFG_A5	UINT	%FW13462	Axis group xx axis number of composition axis5
_AGxx_CFG_A6	UINT	%FW13463	Axis group xx axis number of composition axis6
_AGxx_CFG_A7	UINT	%FW13464	Axis group xx axis number of composition axis7
_AGxx_CFG_A8	UINT	%FW13465	Axis group xx axis number of composition axis8
_AGxx_CFG_A9	UINT	%FW13466	Axis group xx axis number of composition axis9
_AGxx_CFG_A10	UINT	%FW13467	Axis group xx axis number of composition axis10
_AGxx_MTCP_Px	LREAL	%FL3367	Axis group xx X axis position(MCS)

_AGxx_MTCP_Py	LREAL	%FL3368	Axis group xx Y axis position(MCS)
_AGxx_MTCP_Pz	LREAL	%FL3369	Axis group xx Z axis position(MCS)
_AGxx_MTCP_A	LREAL	%FL3370	Axis group xx X axis rotation(MCS)
_AGxx_MTCP_B	LREAL	%FL3371	Axis group xx X axis rotation(MCS)
_AGxx_MTCP_C	LREAL	%FL3372	Axis group xx Z axis rotation(MCS)
_AGxx_PTCP_Px	LREAL	%FL3373	Axis group xx X axis position(PCS)
_AGxx_PTCP_Py	LREAL	%FL3374	Axis group xx Y axis position(PCS)
_AGxx_PTCP_Pz	LREAL	%FL3375	Axis group xx Z axis position(PCS)
_AGxx_PTCP_A	LREAL	%FL3376	Axis group xx X axis rotation(PCS)
_AGxx_PTCP_B	LREAL	%FL3377	Axis group xx Y axis rotation(PCS)
_AGxx_PTCP_C	LREAL	%FL3378	Axis group xx Z axis rotation(PCS)

4) Master flag

Variable	Type	Address	Description
_EC_LINKUP_INFO	BOOL	%FX65600	Link Up/Down Information
_EC_COMM	BOOL	%FX65601	communication connection status
_EC_COMM_ERR	BOOL	%FX65602	Communication timeout error
_EC_PDO_ERR_CNT	UINT	%FW4102	PDO error count
_EC_SLAVE_RDY	ARRAY[0..63] OF BOOL	%FX65664	Slave ready
_EC_SDO_BUSY	ARRAY[0..63] OF BOOL	%FX65792	Slave SDO processing Busy
_EC_SDO_ERR	ARRAY[0..63] OF BOOL	%FX65920	Slave SDO processing error
_EC_LINE_FAIL	ARRAY[0..63] OF BOOL	%FX66048	Cable disconnection
_EC_MASTER_STATE	BYTE	%FB8264	Master EtherCAT STATE(1 : Init, 2 : PreOP, 3 : bootstrap, 4 : SafeOP, 8 : OP, 99 : CHG)
_EC_SLAVE_NUM	WORD	%FW4133	Number of EtherCAT slave connections
_EC_ERR_INFO1	STRING	%FB8272	EtherCAT error information 1
_EC_ERR_INFO2	STRING	%FB8304	EtherCAT error information 2
_EC_TRANSMITTED_OK	UDINT	%FD2084	number of frames transmitted
_EC_RECEIVED_OK	UDINT	%FD2085	number of frames received
_EC_CRCERR_CNT	UDINT	%FD2086	Receive CRC error frame
_EC_COLLISION_CNT	UDINT	%FD2087	Number of collision frames
_EC_CARRIER_SENSE_ERR	UDINT	%FD2088	Carrier sense error
_EC_LINKOFF_CNT	UDINT	%FD2089	Number of link offs
_EC_OVERSIZE_FRAME	UDINT	%FD2090	Receiving oversized Frames
_EC_UNDERSIZE_FRAME	UDINT	%FD2091	Receiving undersized Frames
_EC_JABBER_FRAME	UDINT	%FD2092	Receive Jabber Frame
_EC_PDO_CUR_TRANSCYCLE	UDINT	%FD2093	PDO transmission cycle(ns)
_EC_PDO_MAX_TRANSCYCLE	UDINT	%FD2094	Maximum PDO transmission cycle (ns)
_EC_PDO_MIN_TRANSCYCLE	UDINT	%FD2095	Minimum PDO transmission cycle (ns)
_EC_PDO_TRANS_JITTER	UDINT	%FD2096	PDO frame transmission jitter (ns)
_EC_PDO_ERR_CNT_TOTAL	UDINT	%FD2104	PDO error count(accumulation)
_EC_LOST_FRAME	UDINT	%FD2105	Frames lost
_EC_PDO_ERR_CNT_MAX	UDINT	%FD2106	PDO error count (Max)

_EC_ERR_INFO3	STRING	%FB8424	EtherCAT error information 3

5) Slave flag

Variable	Type	Address	Description
_SLVxx_EC_STATE	SINT	%FB47104	EtherCAT Slave xx STATE (1 : Init, 2 : PreOP, 3 : bootstrap, 4 : SafeOP, 8 : OP, 99 : CHG)
_SLVxx_LINK_STATUS	BYTE	%FB47105	EtherCAT Slave xx link information
_SLVxx_ERROR	WORD	%FW23553	EtherCAT Slave xx error
_SLVxx_VENDOR_ID	DWORD	%FD11777	EtherCAT Slave xx Vendor ID
_SLVxx_PRODUCT_CODE	DWORD	%FD11778	EtherCAT Slave xx Product Code
_SLVxx_REVISION_NUMBER	DWORD	%FD11779	EtherCAT Slave xx Revision Number
_SLVxx_ALStatus	AL status information	%FW23563	Slave xx AL status information
_SLVxx_ALStatusCode	AL error code	%FW23564	Slave xx AL error code
_SLVxx_DLStatus	Link status information	%FW23565	Slave xx Link status information
_SLVxx_LinkLostCount	Port A Number of link disconnection	%FD11783	Slave xx Port A Number of link disconnection
_SLVxx_InValidFrameCounterA	Port A Abnormal frame counter	%FB47136	Slave xx Port A Abnormal frame counter
_SLVxx_RxErrorCounterA	Port A Number of physical layer errors	%FB47137	Slave xx Port A Number of physical layer errors
_SLVxx_InValidFrameCounterB	Port B Abnormal frame counter	%FB47138	Slave xx Port B Abnormal frame counter
_SLVxx_RxErrorCounterB	Port B Number of physical layer errors	%FB47139	Slave xx Port B Number of physical layer errors
_SLVxx_InValidFrameCounterC	Port C Abnormal frame counter	%FB47140	Slave xx Port C Abnormal frame counter
_SLVxx_RxErrorCounterC	Port C Number of physical layer errors	%FB47141	Slave xx Port C Number of physical layer errors
_SLVxx_InValidFrameCounterD	Port D Abnormal frame counter	%FB47142	Slave xx Port D Abnormal frame counter
_SLVxx_RxErrorCounterD	Port D Number of physical layer errors	%FB47143	Slave xx Port D Number of physical layer errors
_SLVxx_ForwardedRXErrCounter	Number of transmitted abnormal frames	%FD11786	Slave xx Number of transmitted abnormal frames

6) NC channel flag

It displays the state of NC channel. NC channel flag is displayed as "_NCyy_..."

(yy indicates the NC channel No.(Decimal))

Variable	Type	Address	Description
_NCyy_Ready	BOOL	%FX524288	NC Ch. yy NC ready
_NCyy_Warning	BOOL	%FX524289	NC Ch. yy warning occurrence
_NCyy_Alarm	BOOL	%FX524290	NC Ch. yy alarm occurrence
_NCyy_ResetStatus	BOOL	%FX524291	NC Ch. yy reset state
_NCyy_CycStartBegin	BOOL	%FX524292	NC Ch. yy cycle start begin information
_NCyy_CycStartFinish	BOOL	%FX524293	NC Ch. yy cycle start finish information
_NCyy_TargetQtyCmpl	BOOL	%FX524294	NC Ch. yy target quantity reached signal
_NCyy_PrgmNormalCmpl	BOOL	%FX524295	NC Ch. yy normal completion of program execution
_NCyy_PwrFailInAuto	BOOL	%FX524296	NC Ch. yy power failure in automatic operation
_NCyy_ErrorCode	WORD	%FW32770	NC Ch. yy error code
_NCyy_IPR_HeartBeat	UDINT	%FD16386	NC Ch. yy IPR HeartBeat
_NCyy_IPR_Run	BOOL	%FX524384	NC Ch. yy IPR operation state (0:stop, 1:running)
_NCyy_IPR_WaitEoM	BOOL	%FX524400	NC Ch. yy waiting end of motion state (0: not waiting, 1:waiting)
_NCyy_IPR_EndOfMot	UINT	%FW32776	NC Ch. yy end of motion
_NCyy_IPR_AfBufSts	UINT	%FW32777	NC Ch. yy AutoFIFO buffer state (0: empty, another: buffer usage)
_NCyy_IPR_ErrorCode	UINT	%FW32778	NC Ch. yy IPR error code
_NCyy_PA_ErrorCode	UINT	%FW32779	NC Ch. yy program access error code
_NCyy_IPR_AlarmSts	ARRAY[0..4] OF DWORD	%FD16390	NC Ch. yy IPR alarm information
_NCyy_CycleStart	BOOL	%FX524672	NC Ch. yy cycle start state
_NCyy_FeedHold	BOOL	%FX524673	NC Ch. yy feed hold state
_NCyy_AutoOperation	BOOL	%FX524674	NC Ch. yy automatic operation state
_NCyy_RetraceMove	BOOL	%FX524675	NC channel yy Signal to confirm reverse operation
_NCyy_RapidTrvsOpr	BOOL	%FX524736	NC Ch. yy rapid traverse operation
_NCyy_CuttingFeedOpr	BOOL	%FX524737	NC Ch. yy cutting feed operation
_NCyy_ConstSurfSpeed	BOOL	%FX524738	NC channel yy Signal controlling constant surface speed
_NCyy_TargetVelocity	LREAL	%FL8200	NC Ch. yy target velocity (F command value)
_NCyy_CmdVelocity	LREAL	%FL8201	NC Ch. yy command velocity
_NCyy_TVelOfSpindle	LREAL	%FL8203	NC Ch. yy spindle target velocity (S command value)
_NCyy_CVelOfSpindle	LREAL	%FL8204	NC Ch. yy spindle command velocity
_NCyy_FeedOverride	LREAL	%FL8206	NC Ch. yy feed override

Variable	Type	Address	Description
_NCyy_RapidOverride	LREAL	%FL8207	NC Ch. yy rapid override
_NCyy_SpindleOverride	LREAL	%FL8208	NC Ch. yy spindle override
_NCyy_SpindleStop	BOOL	%FX525376	NC Ch. yy spindle stop state
_NCyy_SpindleCW	BOOL	%FX525377	NC Ch. yy spindle CW operation
_NCyy_SpindleCCW	BOOL	%FX525378	NC Ch. yy spindle CCW operation
_NCyy_SpindleOrient	BOOL	%FX525379	NC channel yy Signal to confirm spindle orientation status
_NCyy_SpindleCVelAgr	BOOL	%FX525380	NC Ch. yy spindle command velocity reached signal
_NCyy_SpindleZeroVel	BOOL	%FX525381	NC Ch. yy spindle zero velocity reached signal
_NCyy_SpindlePosCtrl	BOOL	%FX525382	NC channel yy Signal to confirm spindle position control mode status
_NCyy_SpindleSSCtrl	BOOL	%FX525383	NC channel yy Signal to confirm main axis SS control mode status
_NCyy _MainSpindle	UDINT	%FW32840	NC channel yy Confirm the main spindle axis number
_NCyy_DwellCount	UDINT	%FD16422	NC Ch. yy dwell count
_NCyy_ErrorBlockNum	UDINT	%FD16423	NC Ch. yy error block number
_NCyy_BlockCmdType	UINT	%FW32848	NC Ch. yy command type of current block
_NCyy_CurrentToolNum	UINT	%FW32856	NC Ch. yy current tool number
_NCyy_ToolRadiusComp	UINT	%FW32857	NC Ch. yy offset number of current tool radius compensation
_NCyy_ToolLengthComp	UINT	%FW32858	NC Ch. yy offset number of current tool length compensation
_NCyy_McodeStrobe	BOOL	%FX526080	NC Ch. yy M code output strobe signal
_NCyy_McodeDistCmpl	BOOL	%FX526081	NC Ch. yy M code distribution complete signal
_NCyy_McodeM00	BOOL	%FX526082	NC Ch. yy special M code output signal(M00)
_NCyy_McodeM01	BOOL	%FX526083	NC Ch. yy special M code output signal(M01)
_NCyy_McodeM02	BOOL	%FX526084	NC Ch. yy special M code output signal(M02)
_NCyy_McodeM30	BOOL	%FX526085	NC Ch. yy special M code output signal(M30)
_NCyy_McodeData	UDINT	%FD16441	NC Ch. yy M code data output
_NCyy_ScodeStrobe	BOOL	%FX526144	NC Ch. yy S code output strobe signal
_NCyy_ScodeDistCmpl	BOOL	%FX526145	NC Ch. yy S code distribution complete signal
_NCyy_ScodeData	UDINT	%FD16443	NC Ch. yy S code data output
_NCyy_TcodeStrobe	BOOL	%FX526208	NC Ch. yy T code output strobe signal
_NCyy_TcodeDistCmpl	BOOL	%FX526209	NC Ch. yy T code distribution complete signal
_NCyy_TcodeData	UDINT	%FD16445	NC Ch. yy T code data output
_NCyy_CycleTime	REAL	%FD16446	NC Ch. yy machining cycle time
_NCyy_TotalRunTime	REAL	%FD16447	NC Ch. yy total machining cycle time
_NCyy_PartCount	UDINT	%FD16448	NC Ch. yy machining quantity
_NCyy_PartCountByM99	UDINT	%FD16449	NC Ch. yy M99 machining quantity at repeat machining

Variable	Type	Address	Description
_NCyy_MainProgram	STRING	%FB65800	NC Ch. yy main program name
_NCyy_CurrentProgram	STRING	%FB65832	NC Ch. yy current running program name
_NCyy_MainBlkNum	UDINT	%FD16466	NC Ch. yy block number of main program
_NCyy_CurrentBlkNum	UDINT	%FD16468	NC Ch. yy block number of current running program
_NCyy_ModalG_OneShot	REAL	%FD16476	NC Ch. yy G code modal value group 0 - One shot
_NCyy_ModalG_Motion	REAL	%FD16477	NC Ch. yy G code modal value group 1 - Motion
_NCyy_ModalG_CmdMode	REAL	%FD16479	NC Ch. yy G code modal value group 3 - Command mode (ABS or INC)
_NCyy_ModalG_Mirror	REAL	%FD16480	NC channel yy G Code Modal Value Group 4 - Mirror
_NCyy_ModalG_Feed	REAL	%FD16481	NC Ch. yy G code modal value group 5 - Feed mode
_NCyy_ModalG_Unit	REAL	%FD16482	NC Ch. yy G code modal value group 6 - Unit
_NCyy_ModalG_TRComp	REAL	%FD16483	NC Ch. yy G code modal value group 7 - Tool radius compensation
_NCyy_ModalG_Stroke	REAL	%FD16485	NC Ch. yy G code modal value group 9 - Stroke check
_NCyy _ModalG_Scale	REAL	%FD16487	NC channel yy G Code Modal Value Group 11 - Scale
_NCyy _ModalG_Macro	REAL	%FD16488	NC channel yy G Code Modal Value Group 12 - Macro
_NCyy_ModalG_TLComp	REAL	%FD16489	NC Ch. yy G code modal value group 13 - Tool length compensation
_NCyy_ModalG_WpCoord	REAL	%FD16490	NC Ch. yy G code modal value group 14 - Workpiece coordinate system
_NCyy_ModalG_CutMode	REAL	%FD16491	NC channel yy G Code Modal Value Group 15 - CutMode
_NCyy_ModalG_Plane	REAL	%FD16492	NC Ch. yy G code modal value group 16 - Circular plane
_NCyy_ModalG_RPolar	REAL	%FD16496	NC Ch. yy G code modal value group 20 - Reverse polar coordinate interpolation
_NCyy_ModalG_CylIntp	REAL	%FD16498	NC Ch. yy G code modal value group 22 - Cylindrical interpolation
_NCyy _ModalG_Skip	REAL	%FD16499	NC channel yy G Code Modal Value Group 23 - Skip
_NCyy_ModalFeed	LREAL	%FL8254	NC Ch. yy modal feed
_NCyy_ModalScode	UDINT	%FD16510	NC Ch. yy modal S code
_NCyy_ModalSpindleM	UDINT	%FD16511	NC Ch. yy modal spindle M code
_NCyy_ModelMcode	UDINT	%FD16512	NC Ch. yy Modal M Code
_NCyy_ModelHcode	UDINT	%FD16513	NC Ch. yy Modal H Code
_NCyy_ModalWorkCoord	UDINT	%FD16514	NC Ch. yy Modal Workpiece Coordinate

7) NC channel/axis flag

It displays the state of axis configured on the NC channel. NC channel/axis flag is displayed as "_NCyy_X...", "NCyy_Y…" (yy indicates the NC channel No.(Decimal) and X,Y,Z,A,B,C,U,V,W is the assigned axis)

Variable	Type	Address	Description
_NC01X_Ready	BOOL	%FX532480	NC Ch. 01 axis X ready
_NC01X_Warning	BOOL	%FX532481	NC Ch. 01 axis X warning occurrence
_NC01X_Alarm	BOOL	%FX532482	NC Ch. 01 axis X alarm occurrence
_NC01X_ServoOn	BOOL	%FX532483	NC Ch. 01 axis X servo On/Off
_NC01X_ServoReady	BOOL	%FX532484	NC Ch. 01 axis X servo ready
_NC01X_ServoAlarm	BOOL	%FX532485	NC Ch. 01 axis X servo alarm occurrence
_NC01X_OprRdy	BOOL	%FX532544	NC Ch. 01 axis X operation ready
_NC01X_FeedMode	BOOL	%FX532552	NC Ch. 01 axis X axis feed mode (0: linear axis, 1: rotation axis)
_NC01X_LinkedAxNum	UINT	%FW33285	NC Ch. 01 axis X actual axis number of IPR axis
_NC01X_Busy	BOOL	%FX532608	NC Ch. 01 axis X busy state
_NC01X_Direction	BOOL	%FX532609	NC Ch. 01 axis X operation direction
_NC01X_ForwardRun	BOOL	%FX532610	NC Ch. 01 axis X running to positive direction
_NC01X_ReverseRun	BOOL	%FX532611	NC Ch. 01 axis X running to negative direction
_NC01X_RapidTraverse	BOOL	%FX532612	NC Ch. 01 axis X rapid traverse operation
_NC01X_CuttingFeed	BOOL	%FX532613	NC Ch. 01 axis X cutting feed operation
_NC01X_Homing	BOOL	%FX532614	NC Ch. 01 axis X homing operation
_NC01X_SpindleRun	BOOL	%FX532615	NC channel axis 01 X Spindle operation
_NC01X_PosCmpl	BOOL	%FX532672	NC Ch. 01 axis X positioning completion
_NC01X_Inposition	BOOL	%FX532673	NC Ch. 01 axis X in-position detection
_NC01X_HomeCmpl	BOOL	%FX532675	NC Ch. 01 axis X homing completion
_NC01_Mirror	BOOL	%FX532736	NC channel axis 01 X Signal to confirm Mirror
_NC01X_CmdPosInWC	LREAL	%FL8325	NC Ch. 01 axis X command position in workpiece coordinate system
_NC01X_CmdPosInRC	LREAL	%FL8326	NC Ch. 01 axis X command position in relative coordinate system
_NC01X_ActualVel	LREAL	%FL8327	NC Ch. 01 axis X actual current velocity
_NC01X_RemDistance	LREAL	%FL8329	NC Ch. 01 axis X remaining distance
_NC01X_PosDeviation	LREAL	%FL8330	NC Ch. 01 axis X servo position deviation (tracking error)
_NC01X_WcOffset	LREAL	%FL8334	NC Ch. 01 axis X offset value of workpiece coordinate system
_NC01X_WcBasicOffset	LREAL	%FL8335	NC Ch. 01 axis X basic offset value of workpiece coordinate system
_NC01X_WcShiftOffset	LREAL	%FL8336	NC Ch. 01 axis X shift offset value of workpiece coordinate system
_NC01X_LocalWcOffset	LREAL	%FL8337	NC Ch. 01 axis X offset value of local workpiece coordinate system

Variable	Type	Address	Description
_NC01X_CmdPosInMC	LREAL	%FL8339	NC Ch. 01 axis X command position in machine coordinate system
_NC01X_ActualPosInMC	LREAL	%FL8341	NC Ch. 01 axis X actual current position in machine coordinate system
_NC01X_SkipPosInMC	LREAL	%FL8342	NC channel axis 01 X Position value of machine displaying skip signal
_NC01X_AxErr	WORD	%FW33372	NC Ch. 01 axis X error code
_NC01X_DrvErr	WORD	%FW33373	NC Ch. 01 axis X drive error code

8) SD memory flag

Variable	Type	Address	Description
_SD_Attach	BOOL	%KX8256	SD attachment state
_SD_Rdy	BOOL	%KX8257	SD memory ready
_SD_Err	BOOL	%KX8258	SD memory error
_SD_Init	BOOL	%KX8259	SD memory initializing state
_SD_Closing	BOOL	%KX8260	SD memory closing state
_SD_FATErr	BOOL	%KX8261	File System Error
_SD_AutoLogAct	BOOL	%KX8262	Act Auto-logging
_SD_Busy	BOOL	%KX8263	SD memory busy state
_SD_SpaceWarn	BOOL	%KX8264	SD memory insufficient state
_SD_Detach	BOOL	%KX8265	SD memory detachment state
_SD_VolTot	UDINT	%KD259	SD memory storage capacity(GB)
_SD_VolAvail	UDINT	%KD260	Available storage capacity(KB)
_SD_Ecode	WORD	%KW522	SD memory error code
_SD_FmtInfo	WORD	%KW523	SD memory format information
_SD_FmtRun	BOOL	%KX8368	SD memory format operation state
_SD_FmtDone	BOOL	%KX8369	SD memory format complete state
_SD_FmtErr	BOOL	%KX8370	SD memory format fail state
_SD_FmtEcode	WORD	%KW524	SD memory format error code
_SD_FmtProgress	WORD	%KW525	SD memory format progress ratio (%)
_SD_AttachCnt	WORD	%KW526	SD memory attachment count
_SD_DetachCnt	WORD	%KW527	SD memory detachment count
_SD_AddfuncAct	BOOL	%KX8640	SD additional function operation state
_SD_AddfuncErr	BOOL	%KX8641	SD additional function error state
_SD_AddfuncDone	BOOL	%KX8642	SD additional function complete state
_SD_CmpResult	BOOL	%KX8643	SD result of comparison
_SD_AddfuncKind	WORD	%KW541	SD type of additional function
_SD_AddfuncEcode	WORD	%KW542	SD additional function error code

9) Data log flag

Variable	Type	Address	Description
_DL00_Enable	BOOL	%KX8224	Group 00 datalog enable state
_DL00_Rdy	BOOL	%KX8960	Group 00 datalog ready
_DL00_Act	BOOL	%KX8961	Group 00 datalog operation state
_DL00_Err	BOOL	%KX8962	Group 00 datalog error state
_DL00_Stoping	BOOL	%KX8963	Group 00 datalog stoping state
_DL00_Finish	BOOL	%KX8964	Group 00 datalog finish state
_DL00_Trig	BOOL	%KX8965	Group 00 trigger occurrence state
_DL00_TrigDone	BOOL	%KX8966	Group 00 trigger complete state
_DL00_Evt	BOOL	%KX8967	Group 00 event occurrence state
_DL00_Ovf	BOOL	%KX8968	Group 00 buffer overflow state
_DL00_Ecode	WORD	%KW561	Group 00 datalog error code
_DL00_FileIdx	WORD	%KW562	Group 00 datalog file index number
_DL00_FileRollcnt	WORD	%KW563	Group 00 overwrite count
_DL00_FileSize	UDINT	%KD282	Group 00 file size(Byte)
_DL00_DataRow	UDINT	%KD283	Group 00 data row number
_DL00_RemainBuf	UDINT	%KD284	Group 00 remaining buffer size(Byte)
_DL00_WaitingData	UDINT	%KD285	Group 00 waiting data size(Byte)
_DL00_OvfCnt	WORD	%KW572	Group 00 buffer overflow count
_DL00_TrigCnt	WORD	%KW573	Group 00 trigger occurrence count
_DL00_TrigOvlap	WORD	%KW574	Group 00 trigger overlap count
_DL00_EvtgCnt	WORD	%KW575	Group 00 event occurrence count

10) Encoder flag

Variable	Type	Address	Description
_ENC1_POS	LREAL	%KL0	Encoder1 input position
_ENC2_POS	LREAL	%KL1	Encoder2 input position
_ENC1_UNIT	UINT	%KW8	Encoder1 unit (0:pulse, 1:mm, 2:inch, 3:degree)
_ENC2_UNIT	UINT	%KW9	Encoder2 unit (0:pulse, 1:mm, 2:inch, 3:degree)
_ENC1_VEL	LREAL	%KL3	Encoder 1 velocity
_ENC2_VEL	LREAL	%KL4	Encoder 2 velocity
_ENC1_POS_LATCH	LREAL	%KL**5**	Encoder 1 Input Position Latch
_ENC2_POS_LATCH	LREAL	%KL6	Encoder 2 Input Position Latch

10) P2P flag

Variable	Type	Address	Description
_P2Pn_NDRxx	BOOL	Refer to XG5000 P2P Global/Direct Variable	P2P Parameter n times xx block service
_P2Pn_ERRxx	BOOL		P2P parameter n times xx block service abnormal completion
_P2Pn_STATUSxx	WORD		Error code is displayed when P2P parameter n times xx block service abnormal completion.
_P2Pn_SVCCNTxx	DWORD		Displays the number of P2P parameter n times xx block service normal execution.
_P2Pn_ERRCNTxx	DWORD		Displays the number of P2P parameter n times xx block service abnormal execution.

Appendix 2 Error Information and measurement

It describes the error information of the motion control module and how to deal with it.

(1) Function block error information

Error code		Error Description	Action
Hex	Dec		
0005	5	The current motion control module does not support the corresponding function block.	The motion function block is not executed in the version of current motion control module. Check the version in which the motion function block can be executed.
0006	6	Axis number of motion function block (Axis input) or encoder number (Encoder input) exceeded allowable range.	Set axis and encoder numbers with a range by product.
0007	7	Axis group number of function block (AxesGroup input) exceeded allowable range.	Set axis group number to a value between 1 and 16.
0008	8	NC channel of function block exceeded allowable range.	Check the range of NC channel, and set again.
0009	9	Slave number of function block (Slave input) exceeded allowable range.	Check the range of slave number, and set again.
000B	11	Input of function block exceeded allowable range.	Check the input range of function block, and set again.
000C	12	Array input of function block exceeded allowable range.	Check the array input size of function block, and set again.
0012	18	Internal execution error of function block occurred during the execution of the function block.	The current controller version may cause problems. Please check the supported version of XG5000 and controller.
0013	19	Motion response error occurred during the execution of function block.	The current controller version may cause problems. Please check the supported version of XG5000 and controller.
0014	20	CAM ID (CamTableID input) of function block exceeded allowable range.	Check the CAM ID range of function block, and set again.
0015	21	Jog operation was stopped by another command.	If re-execution of jog operation is necessary, turn Enable input off and then on.

(2) System error information

Error code		Error Description	Action
Hex	Dec		
000E	14	System error	A / S request if repeated on power up
0017	23	Program error	Startup after program modification and reloading
0019	25	Basic parameter error	Check retention status after uploading basic parameters If it is broken, fix it and download it again to check the operation. Exchange if the problem persists.
002B	43	Built-in parameter - Encoder error	Check retention status after uploading built-in parameters If it is broken, fix it and download it again to check the operation. Exchange if the problem persists.
002C	44	Axis parameter error	Re-download after modifying parameters
002D	45	Axis group parameter error	Re-download after modifying parameters

Error code		Error Description	Action
Hex	Dec		
002E	46	EtherCAT parameter error	Re-download after modifying parameters
002F	47	NC parameter error	Re-download after modifying parameters
0030	48	NC program detection error	Re-download after checking the program
0032	50	Significant error detection in external device	Repair and restart the wrong device by referring to the fault detection flag of the external device (according to the parameter).
0038	56	Main task period error	Check the main task and cycle flag and re-download after modifying the main task program or download by increasing the main task cycle of the basic parameter.
0039	57	Cycle task period error	Check the cycle task and cycle flag and re-download after modifying the cycle task program or download by increasing the cycle task cycle of the basic parameter
003A	58	Task program occupancy excess error	1) Secure the time for the system service to operate by reducing the amount of the user program execution within the main task/cycle task. 2) Please set the execution cycle of the main task/cycle task in the basic parameter higher to secure time for the internal service of the system to operate.
003B	59	Local Ethernet parameter check error	Re-download after modifying parameters
01F5	501	RTC data error	Using RTC clock function, resetting and ouccuring repeatly, request A/S.

(3) SD memory card error information

Error code		Error Description	Action
Hex	Dec		
Overall Error Codes			
0001	1	SD Card Recognition Error	Format in FAT32 format and connect SD memory.
0002	2	Partition Information Error	Format in FAT32 format and connect SD memory.
0003	3	File system error	Format in FAT32 format and connect SD memory.
0004	4	SD Card Not Supported	Please connect SD Card with storage of 2GB~512GB.
0005	5	SD card capacity check error	SD memory capacity test failed, and thus SD cannot be used. Please replace SD memory or format the memory before reconnecting.
0006	6	SD Card Capacity Exceeded	SD memory storage is fully used, and data cannot be saved. Please replace SD memory or format the memory before reconnecting. Available storage is less than 20%
0007	7	Folder Creation Failed	Failed to create folder on SD memory card (Please replace SD memory or format the memory before reconnecting.
Datalog group error code			
1000	4096	Group No x folder creating error	After formatting as FAT32, connect the SD memory.
2000	8192	Group No x file open error	After formatting as FAT32, connect the SD memory.
4000	16384	Group No x file wtite error	After formatting as FAT32, connect the SD memory.

Error code		Error Description	Action
Hex	**Dec**		
SD Additional Function error code			
0001	1	File error(file open failure, CRC error)	After creating fie and then in operation.
0002	2	Damaged file (damaging Head, Tail etc.)	After creating fie and then in operation.
0005	5	No password in file	In case the password set in SD card but it does not exsit in saved file. After setting password and then create file.
0006	6	Password mismatch	The password set in PLC does not match with saved on SD card. Check the password.
0007	7	MAC address mismatch	Setting MAC address does not match with MAC address of PLC. Check MAC address and reset the address.
000A	10	No saved file	There is no saved file on SD card. In operation after creating file.
000B	11	PLC mode is RUN status	Check after PLC mode change to STOP.

(4) Analog error information

Error code		Error Description	Action
Hex	Dec		
0064	0100	Input Channel 0 range setting error	Set it to a configurable input range.
0065	0101	Input Channel 1 range setting error	Set it to a configurable input range.
00C8	0200	Input Channel 0 filter value setting error	Set it to a configurable filter value.
00C9	0201	Input Channel 1 filter value setting error	Set it to a configurable filter value.
012C	0300	Input Channel 0 average value setting error	Set it to a configurable average value.
012D	0301	Input Channel 1 average value setting error	Set it to a configurable average value.
0190	0400	Output Channel 0 range setting error	Set it to a configurable output range.
0191	0401	Output Channel 1 range setting error	Set it to a configurable output range.
01F4	0500	Output Channel 0 Input value setting error	Set it to a configurable input value.
01F5	0501	Output Channel 1 Input value setting error	Set it to a configurable input value.
0258	0600	Output Channel 0 Interpolation method setting error	Set it to a configurable Interpolation method range.
0259	0601	Output Channel 1 Interpolation method setting error	Set it to a configurable Interpolation method range.

(5) Cnet error information

1) XGT server error information

Error code		Error Description	Action
Hex	Dec		
0003	0003	Block number excess error	Set request block count to 16 or less
0004	0004	Variable length error	Set variable length to 16 or less
0007	0007	Data type error	Set the data type to X,B,W,D,L.
0011	0017	Data error	Set with configurable data type, area and length.
0090	0144	Monitor execution error	Please register for monitor execution.
0190	0400	Monitor execution error	When executing the monitor, set the registration number range requested by the client to the configurable input range.
0290	0656	Monitor registration error	When registering monitor, set the registration number range requested by the client to the configurable input range.
1132	4402	Device memory error	Set it to an available device.
1232	4658	Data size error	Set the data size to 60 words or less.
1234	4660	Spare frame error	Remove unnecessary content of slack frames.
1332	4914	Data type mismatch error	Remove unnecessary content of slack frames
1432	5170	Data value error	Check if the data value can be converted to Hex.
7132	28978	Variable request area excess error	Set the device to an available zone.

2) Modbus server error information

Error code		Error Description	Action
Hex	Dec		
0001	0001	Function code error	This is a function code that the server device does not support. Set it after checking whether the function code is supported by the Modbus server.
0002	0002	Address error	Set the address range supported by the server device.
0003	0003	Data setting error	Set the data type supported by the server device.
0004	0004	Server station abnormality error	Check the error status of the server (slave) station.
0005	0005	Server station retry request	The server is in a state where it cannot be processed due to a large amount of processing. Request redelivery when client-side processing time is available.
0006	0006	Server station processing time delay	It is a state that takes time for the server station to process. A re-request must be made on the client side.

3) P2P client error information

Error code		Error Description	Action
Hex	Dec		
0005	0005	P2P block Time out error	Check server connection status and media settings.

4) PLC CPU self error information

Error code		Error Description	Action
Hex	Dec		
0015	0021	P2P client timeout error	There is no response within 5 seconds after the PLC makes a P2P request to the communication module. Check the communication module status.
0016	0022	P2P client device error	An invalid device range was used. Please reset your device.

(5) Motion error information

Error code		Error Description	Action	Alarm/ Warning
Hex	Dec			
0E00	3584	The command data range sent from XG5000 is out of the allowed value.	The current controller version may cause problems. Please check the supported version of XG5000 and controller.	Warning
0E01	3585	XG5000 test run function cannot be executed when the controller is in RUN status.	Please execute the XG5000 test operation after putting the controller in the STOP state.	Warning
0E02	3586	If there is an axis in operation, cam data cannot be written.	Write cam data while all axes are not in operation.	Warning
0E03	3587	If there is an axis in operation, encoder parameter cannot be written.	Write encoder parameter while all axes are not in operation.	Warning
0E04	3588	When EtherCAT communication is connected, EtherCAT parameters cannot be written.	After disconnecting EtherCAT communication, write network data.	Warning
0E10	3600	Encoder parameter data is abnormal.	Download the data from XG5000 again, and if it occurs repeatedly after re-executing, request A/S.	Alarm
0E11	3601	Encoder 1 pulse input type of encoder parameter exceeded the setting range.	Set encoder 1 input signal of Encoder parameter to be between 0 and 5.	Alarm
0E12	3602	Encoder 1 maximum value of encoder parameter is out of pulse unit expression value.	For encoder 1 maximum value of encoder parameter, input 1 or more based on pulse unit.	Alarm
0E13	3603	Encoder 1 minimum value of encoder parameter is out of pulse unit expression value.	For encoder 1 minimum value of encoder parameter, input 1 or more based on pulse unit.	Alarm
0E14	3604	Encoder 1 pulse input Max/Min of encoder parameter exceeded the setting range.	Set the minimum value of encoder 1 of encoder parameter to be smaller than the maximum value.	Alarm
0E15	3605	Encoder 2 pulse input type of encoder parameter exceeded the range.	Set encoder 2 input signal of Encoder parameter to be between 0 and 5.	Alarm
0E16	3606	Encoder 2 maximum value of encoder parameter is out of pulse unit expression value.	For encoder 2 maximum value of encoder parameter, input 1 or more based on pulse unit.	Alarm
0E17	3607	Encoder 2 minimum value of encoder parameter is out of pulse unit expression value.	For encoder 2 minimum value of encoder parameter, input 1 or more based on pulse unit.	Alarm
0E18	3608	Encoder 2 pulse input Max/Min of encoder parameter exceeded the setting range.	Set the minimum value of encoder 2 in the encoder parameter to be smaller than the maximum value.	Alarm
0E19	3609	It is not possible to set the encoder input more than the encoder setting of the encoder parameter.	Check the encoder related items in the encoder parameter and set it to a value within the range.	Alarm
0E1A	3610	Encoder 1 pulse count per rotation in the encoder parameter exceeded the setting range.	Set pulse count per rotation of encoder1 in the encoder parameter to be greater than 0 and less than or equal to 4294967295.	Alarm
0E1B	3611	Encoder 1 travel distance per rotation in the encoder parameter exceeded the setting range.	Set travel distance per rotation of encoder1 in the encoder parameter to be greater than 0.000000001 and less than or equal to 4294967295.	Alarm
0E1C	3612	Encoder 2 pulse count per rotation in the encoder parameter exceeded the setting range.	Set pulse count per rotation of encoder2 in the encoder parameter to be greater than 0 and less than or equal to 4294967295.	Alarm
0E1D	3613	Encoder 2 travel distance per rotation in the encoder parameter exceeded the setting range.	Set travel distance per rotation of encoder2 in the encoder parameter to be greater than 0.000000001 and less than or equal to 4294967295.	Alarm
0E1E	3614	Encoder 1 input filter value in the encoder parameter exceeded the setting range.	Set encoder 1 input signal in the encoder parameter to be between 0 and 6.	Alarm

Error code		Error Description	Action	Alarm/ Warning
Hex	Dec			
0E1F	3615	Encoder 2 input filter value in the encoder parameter exceeded the setting range.	Set encoder 2 input signal in the encoder parameter to be between 0 and 6.	Alarm
0E20	3616	Encoder 1 maximum and minimum values of encoder parameters are set not to include the current position of encoder 1.	Encoder 1 maximum and minimum values of encoder parameters are set to include the current position of encoder 1. Or to operate with the set parameters, use the encoder preset command to change the encoder current position to a value within the parameter range.	Alarm
0E21	3617	Encoder 2 maximum and minimum values of encoder parameters are set not to include the current position of encoder 2.	Encoder 2 maximum and minimum values of encoder parameters are set to include the current position User. Or to operate with the set parameters, use the encoder preset command to change the encoder current position to a value within the parameter range.	Alarm
0E22	3618	Encoder 1 position latch value in the encoder parameter exceeded the setting range.	Set encoder 1 position latch in the encoder parameter to be between 0 and 1.	Alarm
0E23	3619	Encoder 2 position latch value in the encoder parameter exceeded the setting range.	Set encoder 2 position latch in the encoder parameter to be between 0 and 1.	Alarm
0E30	3632	EtherCAT parameter data is abnormal.	Download the data from XG5000 again, and if it occurs repeatedly after re-executing, request A/S.	Alarm
0E31	3633	The number of constant period communication timeouts of the EtherCAT parameter exceeded the range.	Set the constant period communication timeout number of the EtherCAT parameter to a value between 1 and 8.	Alarm
0E32	3634	An error occurred while parsing EtherCAT parameters.	Check the EtherCAT parameters, and set again.	Alarm
0E40	3648	Connection command cannot be executed with EtherCAT parameter error.	Check the EtherCAT parameters, and set again.	Warning
0E41	3649	Network connection command is running.	Check that the network connection command is not entered again while the network connection command is being executed.	Warning
0E42	3650	A network disconnect command is running.	Check that the network disconnection command is not entered again while the network disconnection command is being executed.	Warning
0E43	3651	The connect/disconnect command cannot be executed with mode switching.	Check that no mode switching is executed during network connect/disconnect command operation.	Warning
0E44	3652	The connect/disconnect command cannot be executed with ESTOP Command.	Check that no ESTOP Command is executed during network connect/disconnect command operation.	Warning
0E50	3664	Unable to execute encoder preset command due to encoder parameter error.	Check the encoder related items of the encoder parameter to see if it is set to a value within the range, and set the encoder parameter to a normal value using XG5000.	Warning
0E51	3665	Preset command cannot be done because of the axis which using relevant encoder as a main axis	When there is an axis that operates the corresponding encoder as the main axis, check if the encoder preset command is input to the corresponding encoder.	Warning

Error code		Error Description	Action	Alarm/ Warning
Hex	Dec			
0E52	3666	The encoder preset position is out of the range of the maximum or minimum value of relevant encoder.	Set the encoder preset position to a range greater than or equal to the minimum value of relevant encoder and less than or equal to the maximum value.	Warning
0E53	3667	The encoder selection of the encoder preset command exceeded the range.	For encoder selection, set a value between 0 and 1 (0: Encoder 1, 1: Encoder 2).	Warning
0E54	3668	The command to set encoder current position cannot be executed because the built-in ENC1 is set as the encoder of the spindle axis automatically controlled by the NC function module.	Check that the built-in ENC1 is not set as the encoder connected to the spindle axis in the 'Spindle encoder selection' item of the axis parameters.	Warning
0E55	3669	The command a set encoder current position cannot be executed because the built-in ENC2 is set as the encoder of the spindle axis automatically controlled by the NC function module.	Check that the built-in ENC2 is not set as the encoder connected the spindle axis in the 'Spindle encoder selection' item of the axis parameters.	Warning
0E60	3680	A command cannot be executed with basic parameter data error.	Download the basic parameters from XG5000 again, and if it occurs repeatedly basic parameter error after re-executing, request A/S.	Alarm
0E61	3681	CAM data is abnormal.	Download the data from XG5000 again, and if it occurs repeatedly after re-executing, request A/S.	Alarm
0E80	3712	LS_SaveMotionData command is running.	Please execute it again after the previously executed LS_SaveMotionData command is completed.	Alarm
0E81	3713	An error occurred while executing the LS_SaveMotionData command.	Execute the LS_SaveMotionData command again in XG5000, and if it occurs repeatedly after re-executing, request A/S.	Alarm
0E82	3714	Exceeded the mode range of the LS_SaveMotionData command.	Set Mode to 0 or 1 (0: save only changed motion data, 1: save all motion data)	Alarm
0E90	3728	Error executing all axis command MC_PowerAll.	Check if the axis is in a state where the command can be executed and execute the command again.	Warning
0E91	3729	Error executing all axis command MC_HomeAll.	Check if the axis is in a state where the command can be executed and execute the command again.	Warning
0E92	3730	Error executing all axis command LS_HomeAll.	Check if the axis is in a state where the command can be executed and execute the command again.	Warning
0E93	3731	Error executing all axis command MC_StopAll.	Check if the axis is in a state where the command can be executed and execute the command again.	Warning
0E94	3732	Error executing all axis command MC_HaltAll.	Check if the axis is in a state where the command can be executed and execute the command again.	Warning
0E95	3733	Error executing all axis command MC_Reset2All.	Check if the axis is in a state where the command can be executed and execute the command again.	Warning
0E96	3734	Error executing all axis command MC_SetPositionAll.	Check if the axis is in a state where the command can be executed and execute the command again.	Warning
0F00	3840	Failed to change to EtherCAT INIT state.	Check the communication cable connection status and slave operation status (power on and error occurrence). Make sure communication cables are not exposed to noise.	Alarm

Error code		Error Description	Action	Alarm/ Warning
Hex	Dec			
0F06	3846	EtherCAT INIT status initialization (DC_INIT) error.	Check the communication cable connection status and slave operation status (power on and error occurrence). Make sure communication cables are not exposed to noise.	Alarm
0F09	3849	There is no EtherCAT slave connected to the controller.	If there is a slave connected to the controller, check that the communication cable between the controller and the EtherCAT slave is properly installed, that the power is normally supplied to the EtherCAT slave, or that the communication cable is not exposed to noise.	Alarm
0F0A	3850	The maximum number of connected slaves has been exceeded.	Check that there are not exceed 64 EtherCAT slaves connected to the controller	Alarm
0F0E	3854	There is a difference in node ID and network settings.	Check that the network cable connection sequence matches the network settings.	Alarm
0F0F	3855	There is wrong with the node ID setting.	Check if the node ID is duplicated or there is an error in the setting.	Alarm
0F10	3856	Failed to change to EtherCAT PREOP state.	Check the communication cable connection status and slave operation status (power on and error occurrence). Make sure communication cables are not exposed to noise.	Alarm
0F1E	3870	No network setting data.	Send slave parameters to the controller using XG5000.	Alarm
0F1F	3871	The network setting data and the connected slave are different.	Transmit the slave parameters after connecting the slave using the network slave automatic connection using XG5000.	Alarm
0F20	3872	Failed to change to EtherCAT SAFEOP state.	Check the communication cable connection status and slave operation status (power on and error occurrence). Make sure communication cables are not exposed to noise.	Alarm
0F30	3888	Failed to change to EtherCAT OP state.	Check the communication cable connection status and slave operation status (power on and error occurrence). Make sure communication cables are not exposed to noise.	Alarm
0F40	3904	Failed to change EtherCAT OP state to INIT state.	Check the communication cable connection status and slave operation status (power on and error occurrence). Make sure communication cables are not exposed to noise.	Alarm
0F50	3920	There is no response in the communication connection state.	Check the communication cable connection status and slave operation status (power on and error occurrence). Make sure communication cables are not exposed to noise.	Alarm
0F51	3921	A constant cycle communication error has occurred. (A communication error exceeding the number of master parameter constant cycle communication timeouts occurred.)	Check that the servo power is not turned off during communication, the communication cable is installed normally, or the communication cable is not exposed to noise.	Alarm
0F52	3922	A constant cycle communication error has occurred. (AlStatus error occurred in slave)	Please check the AL Status Code of the slave.	Alarm

Error code		Error Description	Action	Alarm/ Warning
Hex	Dec			
0F60	3936	The slave device address (Adp) setting value of the ESC read command exceeded the range.	Check and set the slave device address (Adp) range according to the EtherCAT command code (EcatCmd) setting value.	Warning
0F61	3937	The data size setting value of the ESC read command exceeded the range.	Set the data size setting value of the ESC read command to 1 to 4 (BYTE)	Warning
0F62	3938	The EtherCAT command code (EcatCmd) setting value of the ESC read command is incorrect.	Set the EtherCAT command code to one of 1 (APRD), 4 (FPRD), or 7 (BRD).	Warning
0F63	3939	There is no response from the slave device to the ESC read command.	Check whether the slave device designated as Adp is installed normally or if the Ado address value is in the read-allowed area.	Warning
0F64	3940	ESC read command cannot be executed when the slave is in Init state.	Change the status of the slave to PreOP, SafeOP, or OP, then execute again.	Warning
0F70	3952	The slave device address (Adp) setting value of the ESC write command exceeded the range.	Check and set the slave device address (Adp) range according to the EtherCAT command code (EcatCmd) setting value.	Warning
0F71	3953	The data size setting value of the ESC write command exceeded the range.	Set the data size setting value of the ESC write command to 1 to 4 (BYTE)	Warning
0F72	3954	The EtherCAT command code (EcatCmd) setting value of the ESC write command is incorrect.	Set the EtherCAT command code to one of 2 (APWR), 5 (FPWR), or 8 (BWR).	Warning
0F73	3955	There is no response from the slave device to the ESC write command.	Check whether the slave device designated as Adp is installed normally or if the Ado address value is in the write allowed area.	Warning
0F74	3956	The area designated by the ESC address (Ado) cannot be written while the communication connection/disconnection command is being executed or in the communication connection state.	In the ESC write command, set the writable ESC address (Ado) during communication connection/disconnection command execution or communication connection status.	Warning
0F75	3957	ESC write command cannot be executed when the slave is in Init state.	Change the status of the slave to PreOP, SafeOP, or OP, then execute again.	Warning
0FF2	4082	Normal operation related to encoder input cannot be executed due to the controller H/W error.	If it occurs repeatedly during retry, please request A/S.	Alarm
1000	4096	Axes are not ready to operating (No is not connected.)	Execute the command when the axis is ready for operation.	Warning
1001	4097	Cannot run in "Disabled" state.	Check the operable axis status of the command, and execute the command in the status where the command can be operated.	Warning
1002	4098	Cannot run in "Standstill" state.	Check the operable axis status of the command, and execute the command in the status where the command can be operated.	Warning
1003	4099	Cannot run in "Discrete" state.	Check the operable axis status of the command, and execute the command in the status where the command can be operated.	Warning
1004	4100	Cannot run in "Continuous" state.	Check the operable axis status of the command, and execute the command in the status where the command can be operated.	Warning
1005	4101	Cannot run in "Synchronized" state.	Check the operable axis status of the command, and	Warning

Error code		Error Description	Action	Alarm/ Warning
Hex	Dec			
			execute the command in the status where the command can be operated.	
1006	4102	Cannot run in "Homing" state.	Check the operable axis status of the command, and execute the command in the status where the command can be operated.	Warning
1007	4103	Cannot run in "stopped" state.	Check the operable axis status of the command, and execute the command in the status where the command can be operated.	Warning
1008	4104	Cannot run in "ErrorStop" state.	Check the operable axis status of the command, and execute the command in the status where the command can be operated.	Warning
100A	4106	Motion command cannot be executed when the affiliated axis group is active.	Execute the command after changing the axis group to GroupDisabled with the axis group disable command.	Warning
100B	4107	This command cannot be given to the virtual axis.	This command cannot be executed on the virtual axis. Check that the command is not executed on the virtual axis.	Warning
100C	4108	If it is registered as NC channel/axis and NC control operation is in progress, the corresponding command cannot be executed.	Check the operable axis status of the command, and execute the command in the status where the command can be operated.	Warning
100D	4109	The command cannot be executed because the axis is not active.	Check if the setting axis of the relevant command is registered in the axis parameter. Axis can be registered in axis parameter among motion data items of XG5000.	Warning
100E	4110	While the motion test operation command is being executed, it is changed to 'Run' and cannot continue operation.	Check that the controller has not changed to 'Run' state while the axis is running.	Warning
100F	4111	The controller is stopped by the ESTOP command and the axis operation cannot be continued.	Check that the controller is not stopped by ESTOP command during axis operation.	Alarm
1010	4112	The controller has changed to 'Stop' or 'Error' status and operation cannot be continued.	Check that the controller has not changed to 'Stop' or 'Error' state while the axis is running.	Alarm
1011	4113	Since the network connection was disconnected, operation cannot be continued.	Check if the network connection is not disconnected due to slave power error, network cable error, or noise inflow on the network cable while the axis is running.	Alarm
1012	4114	The absolute target position where the position setting value of the command is reflected is out of the pulse unit position expression range.	Set the command position so that it does not exceed the 32-bit INT range (-2147483648 ~ 2147483647) when the absolute target position value reflecting the position setting value of the command is converted into pulse units. (When using the 'Position Control Range Expansion' function, 48-bit INT range can be set.)	Warning
1013	4115	The operating speed value is less than 0 or exceeds the maximum speed value.	Set the operation speed value greater than 0 and less than the maximum speed value set for the axis.	Warning
1014	4116	Acceleration is set to negative.	Set the acceleration value to a value greater than or equal to 0.	Warning
1015	4117	Deceleration is set to negative.	Set the deceleration value to a value greater than or	Warning

Error code		Error Description	Action	Alarm/ Warning
Hex	Dec			
			equal to 0.	
1016	4118	Jerk is set to negative.	Set the Jerk value to a value greater than or equal to 0.	Warning
1017	4119	The direction setting value is out of range.	Check the range of the direction setting value of the command, and set it within the range. (Refer to Chapter 6 Commands and Functions)	Warning
1018	4120	The torque setting value is out of range.	Set the torque setting value within 1000%.	Warning
1019	4121	The torque ramp setting value is out of range.	Set torque ramp setting value to a value greater than or equal to 0.	Warning
101A	4122	The Buffer Mode setting value is out of input range.	Set a settable value (0 to 5) in the buffer mode.	Warning
101B	4123	The Execution Mode setting value is out of input range.	Set a settable value (0 to 1) in the Execution Mode.	Warning
101C	4124	A tracking error excess alarm occurs, and operation cannot be continued.	The deviation between the command position and the current position is out of the 'Exceeding value of tracking error'. In order not to occur an alarm, tune the servo drive or set the 'Exceeding value of tracking error' larger.	Alarm
101D	4125	Exceeding value of tracking error warning occurred.	The deviation between the command position and the current position is out of the 'Exceeding value of tracking error'. In order not to occur an alarm, tune the servo drive or set the 'Exceeding value of tracking error' larger.	Warning
101F	4127	The command position value transmitted to the servo drive is out of the pulse unit expression value.	When the command position value was converted into pulse units, it was out of the 32-bit area When the command position values is converted to a pulse, it set within the range of -2147483648 to 2147483647) (When using the 'Position Control Range Expansion' function, 48-bit INT range can be set.)	Alarm
1020	4128	Undefined axis command	The command is not executed in the current version of the controller. Check the version in which the motion function block can be executed.	Warning
1021	4129	The same command was executed, thus canceling the previously executed command.	The corresponding motion command can only be executed once per scan. Change the operating condition of the program so that one motion command is executed per scan.	Warning
1022	4130	The number of Buffered commands that can be executed has been exceeded.	The command cannot be executed because the command buffer of the corresponding axis is full. The number of Buffered Command that can be executed is 100. Adjust the command execution timing.	Warning
1023	4131	The value of the parameter exceeded the input range.	Check the parameter input range and execute the command.	Warning
1030	4144	Axis parameters cannot be written while the axis is running.	Write parameters when the relevant axis is not in operation.	Warning
1040	4160	Axis parameter data is abnormal.	Download the data from XG5000 again, and if it occurs repeatedly after re-executing, request A/S.	Alarm
1041	4161	Operation cannot be executed due to the axis	Check the Axis parameters, and set again.	Alarm

Error code		Error Description	Action	Alarm/ Warning
Hex	Dec			
		parameters error.		
1042	4162	The speed limit of the axis parameter cannot be set to a value less than 0.	Set the speed limit of the basic parameter to a value greater than zero.	Warning
1043	4163	Axis parameter software upper/lower value exceeds range.	Set the soft upper limit of the axis parameter to be greater than or equal to the soft lower limit.	Warning
1044	4164	The current speed filter time constant value of the axis parameter exceeds the range.	Set the parameter setting value from 0 to 100.	Warning
1045	4165	Error reset monitoring time of axis parameter exceeded the range.	Set the parameter setting value from 1 to 1000.	Warning
1046	4166	The set value of transfer distance per rotation exceeds the range	Set the parameter setting value to 0.000000001 or more and 4294967295 or less.	Warning
1047	4167	Infinite length repeat position setting value exceeded the range.	Set the parameter setting value greater than 0 and less than or equal to 2147483647 in pulse units.	Warning
1048	4168	The in-position width setting value is out of range.	Set the parameter setting value to be greater than or equal to 0 and less than or equal to 2147483647 in pulse units.	Warning
1049	4169	The set value of Tracking error over range exceeds the range	Set the parameter setting value to be greater than or equal to 0 and less than or equal to 2147483647 in pulse units.	Warning
104A	4170	The current position display compensation amount setting value exceeds the range.	Set the parameter setting value to be greater than or equal to 0 and less than or equal to 2147483647 in pulse units.	Warning
104B	4171	The set value of JOG high speed speed exceeds the range	Set the parameter setting value greater than 0, greater than the jog low speed value, and less than the speed limit value.	Warning
104C	4172	The set value of jog low speed speed exceeds the range	Set the parameter setting value greater than 0, less than the jog high speed speed value, and less than the speed limit value.	Warning
104D	4173	The set value of jog acceleration exceeds the range	Set the parameter setting value to 0 or higher.	Warning
104E	4174	The set value of jog deceleration exceeds the range	Set the parameter setting value to 0 or higher.	Warning
104F	4175	The set value of jog jerk exceeds the range	Set the parameter setting value to 0 or higher.	Warning
1050	4176	The set value of gear ratio of motor side exceeds the range	Set the parameter setting value from 1 to 65535.	Warning
1051	4177	The set value of gear ratio of mechanical side exceeds the range	Set the parameter setting value from 1 to 65535.	Warning
1052	4178	The set value of pulse per rotation exceeds the range	Set the parameter setting value greater than 0 and less than or equal to 4294967295.	Warning
1053	4179	The set value of connection device exceeds the range	Set the device number of the slave that can be supported. The node setting range is 0 (no connection device), 1 to 64.	Warning
1054	4180	The set value of axis type exceeds the range	Set the parameter setting value to '0: real axis' or '1: virtual axis'.	Warning
1055	4181	The set value of speed command unit exceeds the range	Set the parameter setting value to '0: unit /sec', '1: unit /min', '2: rpm'.	Warning
1056	4182	The set value of blacklash compenstion amount	Set the parameter setting value to be greater than or	Warning

Error code		Error Description	Action	Alarm/ Warning
Hex	Dec			
		exceeds the range.	equal to 0 and less than or equal to 65535 in pulse units.	
1060	4192	Servo On cannot be executed due to a servo drive error.	Check the error factor of the servo drive, remove the servo drive error, and then turn on the servo.	Alarm
1061	4193	Servo-on execution command was executed again during servo-on processing.	Check if the servo-on command is not executed again in the program or XG5000 during servo-on processing.	Warning
1062	4194	Servo-on could not be completed because the servo drive could not change to "ReadyToSwitchON" status.	Check the status of the servo drive. Servo-on command may not be executed under certain conditions.	Alarm
1063	4195	Servo on could not be completed because the servo drive could not be changed to "Switched on" status.	Check the status of the servo drive. Servo-on command may not be executed under certain conditions.	Alarm
1064	4196	Servo on could not be completed because the servo drive could not be changed to "operation enable" status.	Check the status of the servo drive. Servo-on command may not be executed under certain conditions.	Alarm
1065	4197	Servo-on cannot be completed because the "Quick Stop" function of the servo drive is activated.	Check the status of the servo drive. Servo-on command may not be executed under certain conditions.	Alarm
1066	4198	Servo off execution command was executed again during servo- Servo-Off processing.	Check if the servo off command is not executed again in the program or XG5000 during servo- Servo off processing.	Warning
1067	4199	Servo off execution command execution is not completed.	Check the the servo drive status.	Alarm
1070	4208	The servo error reset monitoring time has been exceeded.	The servo drive error was not removed after the error reset monitoring time set in the axis parameter passed. After removing the cause of the servo drive error, execute the error reset command again.	Warning
1080	4224	Commands using absolute coordinates cannot be executed in the absolute coordinates of the undecided homing state.	Execute the absolute coordinate operation command after making the homing state with the home return command or the current position setting command.	Warning
1081	4225	In the state where infinite running repeat is allowed, the target position is out of the infinite running repeat position range in the direction designation.	Set the target position within the infinite running repeat position in the direction designation	Warning
1082	4226	SuperImposed commands cannot be executed while operating with speed control or torque control.	Execute the SuperImposed command while not operating with speed control or torque control.	Warning
1083	4227	A SuperImposed stop command cannot be executed while not operating a SuperImposed.	Execute the SuperImposed stop operation command while operating SuperImposed.	Warning
1090	4240	The position value of the current position change command exceeded the range.	Execute the current position preset command after setting the position setting value above the soft lower limit value and below the soft upper limit value of the extended parameter.	Warning

Error code		Error Description	Action	Alarm/ Warning
Hex	Dec			
1091	4241	The current position change command cannot be executed during operation by homing, speed synchronization, cam, or torque control.	Execute the current position change command when the relevant axis is not in operation of one of homing, speed synchronization, cam, and torque control.	Warning
1092	4242	When the spindle axis is automatically controlled by the NC function module, if the 'Spindle encoder selection' item of the axis parameter is '0: Disabled', the axis current position setting command cannot be executed.	After correctly setting the encoder connection method connected to the spindle axis in the 'Spindle encoder selection' item of the axis parameter, execute the axis current position setting command.	Warning
1093	4243	If the 'Spindle encoder selection' item of the axis parameter is '1: Motor ENC', the 'Position actual value (0x6064)' object does not exist in the TxPDO setting of the EtherCAT slave connected to the spindle axis, so the axis current position setting command cannot be executed.	If the 'Spindle encoder selection' item of the axis parameter is '1: Motor ENC', add the 'Position actual value (0x6064)' object to the TxPDO setting of the EtherCAT slave connected to the spindle axis, resume EtherCAT connection, and then execute the axis current position setting command.	Warning
1094	4244	If the 'spindle encoder selection' item of the axis parameter is '2: Built-in ENC1', the axis current position setting command cannot be executed because the encoder 1 parameter setting is incorrect.	If the 'Spindle encoder selection' item of the axis parameter is '2: Built-in ENC1', set the encoder 1 unit = pulse, Encoder 1 maximum value = 2147483647, Encoder 1 minimum value = -2147483648 and then execute the axis current position setting command. Please run.	Warning
1095	4245	If the 'spindle encoder selection' item of the axis parameter is '3: Built-in ENC2', the axis current position setting command cannot be executed because the encoder 2 parameter setting is incorrect.	If the 'Spindle encoder selection' item of the axis parameter is '3: Built-in ENC2', set the encoder 2 unit = pulse, Encoder 2 maximum value = 2147483647, Encoder 2 minimum value = -2147483648 and then execute the axis current position setting command. Please run.	Warning
10A0	4256	Servo drive does not support torque control mode.	Execute torque control using a servo drive that supports CST mode of EtherCAT CoE.	Warning
10A1	4257	There is no target torque object (0x6071) setting that can execute torque control in the RxPDO entry setting in the slave parameter.	In the slave parameter of XG5000, set the target torque object (0x6071) that supports torque control in the RxPDO entry and send it to the controller.	Warning
10B0	4272	Servo drive does not support homing mode.	Execute homing using a servo drive that supports is 'homing mode of EtherCAT CoE.	Warning
10B1	4273	An error occurred during homing in the servo drive.	Check the error factor of the servo drive, remove the servo drive error, and then execute homing.	Warning
10B2	4274	The homing command cannot be executed while the axis is running.	After the axis stops operating, execute the homing command again in the Standstill state.	Warning
10B3	4275	The switch search speed value of the extended homing command exceeded the range.	When converting the switch search speed value to pulse unit, set it to a value between 0 and 1073741824.	Warning
10B4	4276	The ZERO search speed value of the extended homing command exceeded the range.	When converting the ZERO search speed value to pulse unit, set it to a value between 0 and 1073741824.	Warning
10B5	4277	The Homing acceleration value of the extended homing command exceeded the range.	When converting homing acceleration value to pulse unit, set it to a value between 0 and 1073741824.	Warning
10B6	4278	The Home offset value of the extended homing command exceeded the range.	When converting home offset value to pulse unit, set it to a value between -2147483648 and 2147483647.	Warning

Error code		Error Description	Action	Alarm/ Warning
Hex	Dec			
10B7	4279	Failed to write extended homing command parameters to the drive.	Among the parameters of the extended homing command, there is an item that is not supported by the drive. (Refer to the user manual of the drive) Use the basic homing command (MC_Home).	Warning
10C0	4288	Override command cannot be executed if it is not operating with position/speed control.	Execute the override command while operating by position control or speed control.	Warning
10C1	4289	The override factor of the override instruction exceeded the range.	Set the VelFactor, AccFactor, and JerkFactor values of the override command to 0 or more and execute the override command.	Warning
10C2	4290	After reflecting factor of the override command, the operation speed value exceeded the maximum speed value.	Override within the range that does not exceed the maximum speed value of the relevant axis.	Warning
10D0	4304	Gear ratio denominator od gear operation cannot be zero.	Set the gear ratio denominator to a non-zero value and execute the command.	Warning
10D1	4305	Gear operation MasterValueSource setting value exceeds the range.	Set the MasterValueSource input values to 0~1 and execute the command.	Warning
10D2	4306	Gear operation main axis setting is out of range.	Set axis and encoder numbers with a range by product.	Warning
10D3	4307	The gear operation main axis setting is the same as that of the sub axis.	Set the main axis as an axis different from the sub axis (command axis) and execute the command.	Warning
10D4	4308	Gear operation main axis are not ready to operating	Execute the command when the main axis is ready for operation.	Warning
10D5	4309	If the gear operation main axis is set as an encoder, the command cannot be executed due to the occurrence of a common parameter error.	Check the encoder related items of the encoder parameter to see if it is set to a value within the range, and set the encoder parameter to a normal value using XG5000.	Warning
10D6	4310	MC_GearInPos command cannot be executed when the main axis is operating under torque control.	Execute the MC_GearInPos command while the main axis is not operating under torque control.	Warning
10D7	4311	The speed of the gear operation sub-axis has exceeded the speed limit.	Reduce the speed of the main axis or set a different gear ratio so that the speed of the sub-axis during gear operation does not exceed the speed limit set for the sub-axis.	Warning
10D8	4312	If it is not in gear operation, the gear release command cannot be executed.	The gear release command can only be used during gear operation.	Warning
10D9	4313	The command cannot be executed because the target speed setting value of the positioning gear operation command is smaller than the current operation speed or gear operation speed.	Execute the command after setting the target speed setting value of the positioning gear operation command to be higher than the current operation speed or gear operation speed.	Alarm
10DA	4314	During the positioning gear operation operation, the subordinate axis cannot reach the subordinate axis synchronous position at the set target speed within the time the main axis operates to the main axis synchronous position.	Increase the target speed setting value of the positioning gear operation command or execute the command after adjusting MasterStartDistance so that the subordinate axis can move to the subordinate axis synchronous position within the time that the main axis operates to the main axis synchronous position.	Alarm
10DB	4315	Synchronous operation commands (gear, cam, etc.) cannot be executed when the main axis is in	Execute the synchronous operation command (gear, cam, etc.) while the main axis is not in homing	Warning

Error code		Error Description	Action	Alarm/ Warning
Hex	Dec			
		homing operation.	operation.	
10E0	4320	There is no object setting that can execute the touch probe in the PDO entry setting in the slave parameter.	After setting the object that supports touch probe in the PDO entry in the slave parameter of XG5000, transmit it to the controller.	Warning
10E1	4321	TriggerInput input setting value is out of range.	Please set the TriggerInput setting value to 0 (Touch Probe1) or 1 (Touch Probe2).	Warning
10E2	4322	TriggerInput input setting value of expansion touch probe command is out of range.	Set the TriggerInput setting value of the expansion touch probe command to a value between 0 (Rising edge of TouchProbe1) and 5 (Index pulse of TouchProbe2).	Warning
10E3	4323	TriggerMode input setting value of expansion touch probe command is out of range.	Set the TriggerMode setting value of the expansion touch probe command to 0 (single trigger) or 1 (continuous trigger).	Warning
10F0	4336	Parameter number setting value of parameter read/write command is out of range.	Execute the command after setting the parameter number setting value of the parameter read/write command between 0~28, 100~108, and 200~208.	Warning
10F1	4337	The data set value of the parameter set in the parameter write command is out of range.	Check the data setting range of the parameter you want to set.	Warning
10F2	4338	When changing the corresponding encoder parameter, the parameter cannot be changed because the encoder 1 maximum value is out of the pulse unit expression value.	Change the relevant parameter in advance so that no error occurs when converting the encoder 1 maximum value into pulse units.	Warning
10F3	4339	When changing the corresponding encoder parameter, the parameter cannot be changed because the encoder 1 minimum value is out of the pulse unit expression value.	Change the relevant parameter in advance so that no error occurs when converting the encoder 1 minimum value into pulse units.	Warning
10F4	4340	When changing the corresponding encoder parameter, the parameter cannot be changed because the encoder 2 maximum value is out of the pulse unit expression value.	Change the relevant parameter in advance so that no error occurs when converting the encoder 2 maximum value into pulse units.	Warning
10F5	4341	When changing the corresponding encoder parameter, the parameter cannot be changed because the encoder 2 minimum value is out of the pulse unit expression value.	Change the relevant parameter in advance so that no error occurs when converting the encoder 2 minimum value into pulse units.	Warning
1100	4352	Jog operation command cannot be executed while the axis is in operation.	Execute the jog command while the axis is stopped.	Warning
1101	4353	If the tool retract command is being executed in the NC channel, the jog operation command cannot be executed in more than one axis at the same time.	When canceling the tool retract command of the NC channel or using the tool retract command, execute the jog operation command on only one axis at a time.	Warning
1110	4368	There is an error in the cam operation MasterScaling input value.	A value of 0 cannot be entered in the MasterScaling input value.	Warning
1111	4369	There is an error in the cam operation MasterStartDistance input value.	Set the MasterStartDistance input values to a value greater than or equal to 0 and execute the command.	Warning
1112	4370	There is an error in the cam operation MasterSyncPosition input value.	Set the MasterSyncPosition input values to a value greater than or equal to 0 and execute the command.	Warning

Error code		Error Description	Action	Alarm/ Warning
Hex	Dec			
1113	4371	Cam operation StartMode input value exceeded the range.	Set the StartMode input values to 0~1 and execute the command.	Warning
1114	4372	Cam operation MasterValueSource input value exceeded the range.	Set the MasterValueSource input values to 0~1 and execute the command.	Warning
1115	4373	The specified cam table does not exist.	Adjust the cam table number to a valid cam table number and execute the command.	Warning
1116	4374	Cam operation main axis setting is out of range.	Set axis and encoder numbers with a range by product. If the main axis is a variable, check that the VarOffset value does not exceed the memory area.	Warning
1117	4375	The cam operation main axis setting is the same as that of the sub axis.	Set the main axis as an axis different from the sub axis (command axis) and execute the command.	Warning
1118	4376	Cam operation main axis are not ready to operating	Execute the command when the main axis is ready for operation.	Warning
1119	4377	If the cam operation main axis is set as an encoder, the command cannot be executed due to the occurrence of a common parameter error.	Check the encoder related items of the encoder parameter to see if it is set to a value within the range, and set the encoder parameter to a normal value using XG5000.	Warning
111A	4378	The speed of the cam operation sub-axis has exceeded the speed limit.	Reduce the speed of the main axis or adjust cam table so that the speed of the sub-axis during cam operation does not exceed the speed limit set for the sub-axis.	Warning
111B	4379	If it is not in cam operation, the cam release command cannot be executed.	The cam release command can only be used during cam operation.	Warning
111C	4380	The cam data count setting value of the cam data write command exceeded the range.	Set the cam data number setting value of write cam data command between 1 and 100.	Warning
111D	4381	The specified cam table data of the cam data read command is abnormal.	Set the cam data again, and if it occurs repeatedly after re-executing, request A/S.	Warning
111E	4382	CAM skip command cannot be executed when it is not CAM operation.	Execute the CAM skip command when operating CAM.	Warning
111F	4383	The number of cam cycles to skip in the cam skip command is set to 0.	Set the number of CAM cycles to skip in the CAM skip command to be greater than 0.	Warning
1121	4385	The skip mode setting value of the cam skip command exceeded the range.	Set the skip mode setting value of the cam skip command to a value between 0 and 2 and execute the command.	Warning
1122	4386	Cam table not registered.	Register the cam table or reset the data and execute the command.	Warning
1123	4387	The cam data of the cam data write command is abnormal.	Set the data of the cam data write command correctly.	Warning
1124	4388	Cam main axis value does not exist within the specified range	Check the MasterStartPos and MasterEndPos values, and execute the command again.	Warning
1130	4400	The phase compensation command cannot be executed if the command axis is not in InSync or InGear state of synchronous control (cam, gear operation) operation.	If the command axis of the phase compensation command is in synchronous control operation, execute the phase compensation command in InSync or InGear status.	Warning
1131	4401	There is an error in the main axis setting of the phase compensation command.	Execute the command after setting the main axis setting of the phase compensation command to be the same as the actual main axis of the current	Warning

Error code		Error Description	Action	Alarm/ Warning
Hex	Dec			
			synchronous operation.	
1132	4402	The phase compensation amount of the phase compensation command is out of the position expression range.	Execute the command after setting the phase compensation amount so that it falls within the range of -2147483648 to 2147483647 or less in pulse units.	Warning
1133	4403	The speed setting values of the phase compensation command is out of the range.	Execute the command after setting the speed value of the phase compensation command to be greater than 0 and less than the speed limit of the main axis.	Warning
1140	4416	The connected slave device does not support speed control mode.	Use a slave device that supports the Velocity mode of EtherCAT CoE to execute speed control.	Warning
1150	4432	The connected slave device does not support position control mode.	Use a slave device that supports the CSP mode of EtherCAT CoE to execute position control.	Warning
1151	4433	The operation mode of the connected slave device (0x6060, Mode Of Operation) SDO read service failed.	Check if the slave is in a state that can execute SDO service. If an error occurs repeatedly by re-executing the command, reconnect EtherCAT with “EtherCAT Disconnect – Connect Command” before use.	Warning
1152	4434	Operation mode (0x6060, Mode Of Operation) of the connected slave device Timeout for SDO read request occurred.	Check if the slave is in a state that can execute SDO service. If an error occurs repeatedly by re-executing the command, reconnect EtherCAT with “EtherCAT Disconnect – Connect Command” before use.	Warning
1160	4448	The connected slave device does not support synchronous speed control (CSV) mode.	Use a slave device that supports the synchronous speed control (CSV) mode of EtherCAT CoE to execute speed control.	Warning
1161	4449	There is no target velocity object (0x60FF) setting that can execute synchronous speed control CSV of the RxPDO entry setting in the slave parameter.	After setting the target speed object (0x60FF) supporting synchronous speed control (CSV) in the RxPDO entry of EtherCAT parameter slave data in XG5000, transmit it to the controller.	Warning
1162	4450	The CmdPosMode setting value of synchronous speed control (CSV) operation exceeds the input range.	CmdPosMode only supports a value of 0 (applies current position as command position). Execute the command again after setting CmdPosMode to 0.	Warning
1170	4464	Part length value is not valid	Part length value cannot be set less than 0.	Warning
1171	4465	The perimeter value is not valid.	The perimeter cannot be set to a value less than 0.	Warning
1172	4466	The cutting start position value is not valid.	The cutting start position cannot be set to a value smaller than 1/4 of the circumference value.	Warning
1173	4467	The cutting end position value is not valid.	The cutting end position cannot be set smaller than the cutting start position or larger than 3/4 of the perimeter.	Warning
1174	4468	The synchronous speed ratio value is not valid.	The value of the speed ratio of the cutting section must be set between 50 and 200.	Warning
1175	4469	The 0 speed section ratio value is not valid.	The value of 0 speed section must be set between 0 and 50.	Warning
1176	4470	The cam profile type value is not valid.	Change the camp profile type value and run the command.	Warning
1177	4471	The cam point count value is invalid.	Change the camp point count and run the command.	Warning
1178	4472	The cam curve type value is not valid.	Change the cam curve type value and run the command.	Warning
1179	4473	The cutting area is too wide.	Change the length of the cutting section or change the speed ratio value of the cutting section and execute	Warning

Error code		Error Description	Action	Alarm/ Warning
Hex	Dec			
			the command.	
1200	4608	A hardware upper limit error has occurred.	Clear the error by executing the error reset command after out of the external upper limit signal range using the reverse jog command.	Alarm
1201	4609	A hardware lower limit error has occurred.	Clear the error by executing the error reset command after out of the external lower limit signal range using the reverse direction jog command.	Alarm
1203	4611	The command cannot be executed due to an error in the servo driver during operation.	After removing the cause of the servo error, clear the servo error with the error reset command.	Alarm
1204	4612	The command cannot be executed because the servo is off during operation.	Execute the command again after making the command axis into the servo-on state with the servo-on command.	Alarm
1205	4613	A software upper limit error has occurred.	Clear the error by executing the error reset command after out of the software upper limit range using the reverse jog command.	Alarm
1206	4614	A software lower limit error has occurred.	Clear the error by executing the error reset command after out of the software lower limit range using the forward direction jog command.	Alarm
1210	4624	Motion commands related to traversing cannot be executed if the spindle axis is automatically controlled by the NC function module.	After checking the motion commands that can be executed for the axis assigned as the NC spindle axis, execute the motion commands allowed for the NC spindle axis.	Warning
1220	4640	Parameter number setting value of motion information read/write command is out of range.	Execute the command after setting the parameter number value within of the motion information read command between 0 and 5.	Warning
1230	4656	Master position loop control cannot be enabled/disabled based operation.	Execute/release master position loop control when the corresponding axis is not in operation.	Warning
1231	4657	The P gain input value of the master position loop control function block is negative.	Enter a value of 0 or more in the P gain of the master position loop control function block.	Warning
1232	4658	The I gain input value of the master position loop control function block is negative.	Enter a value of 0 or more in I gain of the master position loop control function block.	Warning
1233	4659	The control output limit value of the master position loop control function block is negative.	Enter a value of 0 or more in the control output limit value of the master position loop control function block.	Warning
1234	4660	Cross- coupled control cannot be enabled/disabled during operation.	Execute/release Cross-coupled control when the corresponding axis is not in operation.	Warning
1235	4661	The command axis and the connection axis of the cross-coupled control function block must be different axes.	Set different axes for the command axis and the connection axis of the cross-coupled control function block.	Warning
1236	4662	In order to operate cross-coupled control, the master position loop controller must be running.	In order to drive cross-coupled control, execute master position loop control for the relevant axis first.	Warning

Error code		Error Description	Action	Alarm/ Warning
Hex	Dec			
1237	4663	The P gain input value of the Cross-coupled control function block is negative.	Enter a value of 0 or more in the P gain of the Cross-coupled control function block.	Warning
1238	4664	The command was re-executed on an axis for which cross-coupled control is not running.	When re-executing the cross-coupled control function block, set the command axis and connection axis to the previously executed values. The value of P gain can be changed without re-executing the function block.	Warning
1F00	7936	A constant cycle communication error has occurred. (A communication error exceeding the number of master parameter constant cycle communication timeouts occurred.)	Check that the servo power is not turned off during communication, the communication cable is installed normally, or the communication cable is not exposed to noise.	Alarm
1F10	7952	SDO command can no longer be executed due to previously executed slave device SDO processing failure	Please reset the connection after checking that the status of the slave device is normal	Warning
1F11	7953	SDO parameter write command cannot be executed while its operating.	Execute SDO parameter write command while the relevant axis is not in operation.	Warning
1F12	7954	The data range of the SDO parameter Index, SubIndex, etc. is out of the range.	SDO parameter Index is set to 0x0000~0x9FFF, SubIndex is 0x00 to 0xFF, and data size is set within 4 byte and read/write SDO parameter.	Alarm
1F13	7955	Abort occurred during SDO parameter write command.	Slave device stopped writing operation while writing SDO parameter. Check the write setting data and the status of the slave device.	Alarm
1F14	7956	There is no response from the slave device the SDO parameter write command.	Slave device no response while writing SDO parameter. Check the slave device status.	Alarm
1F16	7958	Abort occurred during SDO parameter EEPROM save.	Slave device canceled while saving SDO parameter EEPROM. Check the slave device status.	Alarm
1F17	7959	There is no response from the slave device the SDO parameter EEPROM save command.	Slave no response while saving SDO parameter EEPROM. Check the slave device status.	Alarm
1F19	7961	Other commands cannot be executed while writing SDO parameters or saving SDO parameters to EEPROM.	Execute another command after all SDO parameter EEPROM saving is completed.	Alarm
1F20	7968	Abort occurred during SDO parameter read command.	Slave device stopped reading operation while reading SDO parameter. Check the read setting data and the status of the slave device.	Alarm
1F21	7969	There is no response from the slave device the SDO parameter read command.	Slave device no response while reading SDO parameter. Check the slave device status.	Alarm
1F22	7970	SDO parameter read/write command cannot be executed while SDO parameter read/write command is being executed.	Execute the command after all currently executing SDO parameter read/write is completed.	Warning
1F33	7987	Failed to change the operation mode of the servo drive to position control (CSP) mode.	Check whether the corresponding servo drive supports EtherCAT CoE CSP mode, and check the status of the servo drive.	Alarm
1F34	7988	Failed to change the operation mode of the servo drive to home return (Homing) mode.	Check whether the corresponding servo drive supports EtherCAT CoE Homing mode, and check the status of the servo drive.	Alarm
1F35	7989	Failed to change the operation mode of the servo	Check whether the corresponding servo drive	Alarm

Error code		Error Description	Action	Alarm/ Warning
Hex	Dec			
		drive to torque control (CST) mode.	supports EtherCAT CoE CST mode, and check the status of the servo drive.	
1F36	7990	Failed to change the operation mode of the servo drive to speed control (CST) mode.	Check whether the corresponding servo drive supports EtherCAT CoE CSV mode, and check the status of the servo drive.	Alarm
1F50	8016	XG5000 manual tuning function cannot be executed when the controller is in RUN status.	After changing the controller to STOP status, execute manual tuning of XG5000	Warning
1F60	8032	This command is not available in the slave's current EtherCAT state.	Set the slave's EtherCAT status to Boot status and execute the command.	Warning
1F61	8033	Send timeout occurred during file transfer	Check the status of the transfer line or slave and execute the command.	Warning
1F62	8034	Send timeout occurred during file transfer	Check the status of the transfer line or slave and execute the command.	Warning
1F63	8035	A packet error occurred during file transfer.	Check the status of the transfer line or slave and execute the command.	Warning
1F64	8036	Slave is out of memory.	Check the transferred file and execute the command.	Warning
1F65	8037	The device does not exist.	Execute the command after checking whether the FOE function is available as a slave.	Warning
1F66	8038	Access denied to slave.	Execute the command after checking whether the FOE function is available as a slave.	Warning
1F67	8039	Passwords do not match.	Confirm the password and execute the command.	Warning
1F68	8040	The data to be downloaded with the FoE function was not sent to the controller.	Check the communication cable connection status and control operation status.	Warning
1F6F	8047	Slave error occurred during file transfer	Remove the error of the slave and execute the command	Warning
2000	8192	The axis group is not ready for operation.	Execute the command while the axis group is ready for operation.	Warning
2001	8193	Axis group cannot be executed in "Disabled" state.	Check the operable axis group status of the command, and execute the command in the status where the command can be operated.	Warning
2002	8194	Axis group cannot be executed in "Standby" state.	Check the operable axis group status of the command, and execute the command in the status where the command can be operated.	Warning
2003	8195	Axis group cannot be executed in "Moving" state.	Check the operable axis group status of the command, and execute the command in the status where the command can be operated.	Warning
2004	8196	Axis group cannot be executed in "Homing" state.	Check the operable axis group status of the command, and execute the command in the status where the command can be operated.	Warning
2005	8197	Axis group cannot be executed in "Stopping" state.	Check the operable axis group status of the command, and execute the command in the status where the command can be operated.	Warning
2006	8198	Axis group cannot be executed in "Errorstop" state.	Check the operable axis group status of the command, and execute the command in the status where the command can be operated.	Warning

Error code		Error Description	Action	Alarm/ Warning
Hex	Dec			
2007	8199	The configuration axis of the axis group is not servo-on	Check the operable axis group status of the command, and execute the command in the status where the command can be operated.	Warning
200F	8207	The controller is stopped by the ESTOP command and the axis group operation cannot be continued.	Check that the controller is not stopped by ESTOP command during axis group operation.	Alarm
2010	8208	The controller has changed to 'Stop' or 'Error' status and operation cannot be continued.	Check that the controller has not changed to 'Stop' or 'Error' state while the axis group is running.	Alarm
2011	8209	Since the network connection was disconnected, operation cannot be continued.	Check if the network connection is not disconnected due to slave power error, network cable error, or noise inflow on the network cable while the axis group is running.	Alarm
2012	8210	The absolute target position where the position setting value of the command is reflected is out of the pulse unit position expression range.	Set the command position so that it does not exceed the 32-bit INT range (-2147483648 ~ 2147483647) when the absolute target position value reflecting the position setting value of the command is converted into pulse units. (When using the 'Position Control Range Expansion' function, 48-bit INT range can be set.)	Warning
2013	8211	The operating speed value is less than 0 or exceeds the maximum speed value.	Set the operation speed value greater than 0 and less than the maximum speed value set for the axis group.	Warning
2014	8212	Acceleration is set to negative.	Set the acceleration value to a value greater than or equal to 0.	Warning
2015	8213	Deceleration is set to negative.	Set the deceleration value to a value greater than or equal to 0.	Warning
2016	8214	Jerk is set to negative.	Set the Jerk value to a value greater than or equal to 0.	Warning
201A	8218	The Buffer Mode setting value is out of range.	Set a settable value in the buffer mode.	Warning
201B	8219	The Execution Mode setting value is out of input range.	Set a settable value (0 to 1) in the Execution Mode.	Warning
201C	8220	The Transition Mode setting value is out of range.	Set a settable value in the Transition Mode.	Warning
201D	8221	The Transition Parameter setting value is out of range.	Set a settable value in the Transition Parameter.	Warning
201E	8222	The axis group operation was stopped due to an axis group configuration error.	After removing the error cause, clear the error with the axis or axis group error reset command and execute the command.	Warning
201F	8223	The command position value transmitted to the servo drive is out of the pulse unit expression value.	When the command position value was converted into pulse units, it was out of the 32-bit area When the command position values is converted to a pulse, it set within the range of -2147483648 to 2147483647) (When using the 'Position Control Range Expansion' function, 48-bit INT range can be set.)	Warning
2020	8224	It is an undefined axis group command.	The command is not executed in the current version of the controller. Check the version in which the motion function block can be executed.	Warning
2021	8225	The same command was executed, thus canceling the previously executed command.	Make sure that the command is not executed again during the execution of the same command.	Warning

Error code		Error Description	Action	Alarm/ Warning
Hex	Dec			
2022	8226	The number of Buffered commands that can be executed has been exceeded.	The command cannot be executed because the command buffer of the corresponding axis group is full. The number of Buffered Command that can be executed is 100. Adjust the command execution timing.	Warning
2030	8240	Axis group parameters cannot be written while the axis group is running.	Write the axis group parameter when the axis group is not in operation.	Warning
2040	8256	Axis group parameter data is abnormal.	Download the data from XG5000 again, and if it occurs repeatedly after re-executing, request A/S.	Alarm
2041	8257	Operation cannot be executed due to the axis group parameters error.	Check the axis group parameters, and set again.	Alarm
2042	8258	The speed limit of the axis group parameter cannot be set to a value less than 0.	Set the speed limit value to greater than zero.	Warning
2043	8259	The configuration axis number setting value of the axis group parameter exceeded the range.	Set the configuration axis of axis group (axis and encoder numbers) in the range by product.	Warning
2051	8273	The axis to be added is already registered in the axis group.	Please set another axis after checking whether the axis group has the same axis number.	Warning
2052	8274	The current axis group is active, and the axis you want to add is already included in another active axis group.	Execute the command after changing the active axis group including the relevant axis to the GroupDisabled state.	Warning
2053	8275	IdentInGroup setting value of axis group add/remove command exceeded the range.	Set the IdentInGroup setting value from 1 to 10.	Warning
2060	8288	There is no axis setting in the specified axis group of the enable/disable axis group command.	Set one or more axes in the relevant axis group and execute the corresponding command.	Warning
2061	8289	The axis group cannot be activated because there is an operating axis among the axes of the current axis group.	Execute the command in the state that all axes belonging to the relevant axis group are not in operation.	Warning
2062	8290	The designated axis group cannot be activated because the current axis group's configuration axis is a configuration axis of another active axis group.	Check that the axis belonging to the relevant axis group does not belong to another active axis group.	Warning
2063	8291	Axis group operation cannot be executed because the units of axis group configuration axes are different.	To execute the corresponding operation, set the unit of the configuration axes belonging to the axis group to be the same.	Warning
2064	8292	The axis group cannot be activated because parameter of axis group configuration occur error.	Set the parameters of the configuration axes belonging to the axis group within the normal range.	Warning
2065	8293	The axis group cannot be activated because the speed command unit of axis group configuration axes are different.	Set the speed command unit of the configuration axes belonging to the axis group to be the same.	Warning
2066	8294	Axis group cannot be activated because there is an axis whose speed command unit is rpm among the axes of the axis group.	The speed command unit of the configuration axes belonging to the axis group cannot be set as rpm. Set it to a value other than rpm.	Warning
2067	8295	Coordinates operation cannot be executed because the units of axis group configuration axes are different to coordinate system type.	To execute the corresponding operation, set the unit of the configuration axes belonging to the axis group to match with coordinate system type.	Warning

Error code		Error Description	Action	Alarm/ Warning
Hex	Dec			
206F	8303	If the axis group configuration axis is in NC control operation, the axis group cannot be activated.	Execute the command while the configuration axes belonging to the axis group are not in NC control operation.	Warning
2070	8304	The servo drive of the configuration axis does not support the homing mode.	Check whether the corresponding servo drive supports EtherCAT CoE Homing mode, and check the status of the servo drive.	Warning
2071	8305	There is an axis for which homing has not been completed normally among the configuration axes.	Check the component axis error code, remove the cause of the error of the relevant axis, and then execute the command again.	Warning
2072	8306	The axis group homing command cannot be executed while the axis group is running.	After the axis group stops operating, execute the axis group homing command again in the GroupStandby state.	Warning
2080	8320	There is an axis where an error occurred while setting the current position among the configuration axes.	Check the component axis error code, remove the cause of the error of the relevant axis, and then execute the command again.	Warning
2090	8336	Absolute coordinate linear interpolation command cannot be executed in undecided homing status.	Execute the command after making the homing state with the home return command or the current position setting command.	Warning
2091	8337	The speed limit of the linear interpolation axis has been exceeded.	Execute the command by lowering the command speed so as not to exceed the speed limit of the component axis.	Warning
2092	8338	In the case of a transition with a designated corner distance, the transition operation cannot be executed because the designated corner distance is greater than the moving distance to the target position.	Set the corner distance value specified in the transition parameter to be smaller than the moving distance to the target position.	Warning
2093	8339	In the case of a transition with a specified corner distance, transition operation cannot be executed because the radius of the arc to be inserted exceeds 2147483647pulse.	Execute linear interpolation by resetting the target position or changing the transition mode so that the two lines are not nearly on a straight line.	Warning
2094	8340	Linear interpolation operation cannot be executed when the main axis or sub axis is “allowed” for infinite running repeat.	Execute the command after changing the setting of infinite running repeat of main axis or sub axis to "0: Prohibited".	Warning
20A0	8352	Circular Interpolation command cannot be executed in undecided homing status.	Execute the command after making the homing state with the home return command or the current position setting command.	Warning
20A1	8353	Circular interpolation mode setting value exceeded the range.	Set the circular interpolation mode to a value between 0 and 2 (0: auxiliary point, 1: center point, 2: radius).	Warning
20A2	8354	The circular interpolation pass selection setting value exceeds the range.	Set the circular interpolation pass selection setting value between 0 and 1 (0: CW, 1: CCW).	Warning
20A3	8355	The radius setting exceeded the range in the circular interpolation radius method.	Set radius setting from circular interpolation main axis operating data for 80% bigger than its half distance of beginning point to end point.	Warning
20A4	8356	Cannot be operated if circular interpolation start point =center point (middle point) or center point (middle point)= end point.	Execute circular interpolation after setting the center point (or middle point) to a position different from the starting point (or end point) in circular interpolation.	Warning

Error code		Error Description	Action	Alarm/ Warning
Hex	Dec			
20A5	8357	The start point and end point is not possible to be same in the middle point (radius) mode of circular interpolation.	If the circular interpolation method is set to the middle point (or radius), execute the circular interpolation after setting the position of the start point and the position of the end point differently.	Warning
20A6	8358	Radius setting error in circular interpolation.	The radius of the circle to carry out circular interpolation operation is from 0 to 2,147,483,2147483647pulse. Execute the command after setting the input value so that the radius can be calculated within the setting range. (When using the 'Position Control Range Expansion' function, 48-bit INT range can be set.)	Warning
20A7	8359	Not possible to carry out the operation as linear profile comes out of circular interpolation.	In case of circular interpolation middle point, execute circular interpolation after changing the middle point so that it is not located on a straight line between the start point and the endi point.	Warning
20A8	8360	Circular interpolation operation cannot be executed when the main axis or sub axis is “allowed” for infinite running repeat.	Execute the command after changing the setting of infinite running repeat of main axis or sub axis to "0: Prohibited".	Warning
20A9	8361	Circular interpolation cannot be executed when the number of axis group configuration axes is 4 or more.	In the case of circular interpolation, set the axis group to 2 axes, and in the case of helical interpolation, set the axis group to 3 axes.	Warning
20AA	8362	Circular interpolation cannot be executed if the axis configuration of the axis group is not configured in order.	In case of circular interpolation, set the axis group in order from the first.	Warning
20AB	8363	The speed limit of the circular interpolation axis has been exceeded.	Execute the command by lowering the command speed so as not to exceed the speed limit of the component axis.	Warning
20AC	8364	In circular interpolation, in the middle point (or radius) method, the middle point (center point) must exist in the same XY plane as the start point.	Execute circular interpolation after setting the center point (or middle point) to the same XY plane as the start point (or end point) in circular interpolation.	Warning
20AD	8365	In arc Interpolation for Coordinate System, if start point, center point (middle point), end point locates the same line, operation can be not executed.	Executes command after changing the circular interpolation center point (or midpoint) so that it is not located on a straight line between the start and end points.	Warning
20C0	8384	Coordinates operation command cannot be executed in undecided homing status.	Execute the command after making the homing state with the home return command or the current position setting command.	Warning
20C1	8385	The PCS setting parameter data of the coordinate system parameter is abnormal.	Check the PCS setting parameters, and set again.	Warning
20C2	8386	The coordinate system type parameter data of the coordinate system parameter is abnormal.	Check the coordinate system type parameters, and set again.	Warning
20C3	8387	The coordinate system parameter data of the coordinate system parameter is abnormal.	Check the machine parameters, and set again.	Warning
20C4	8388	Workspace type data of coordinate system parameter is abnormal.	Check the workspace type parameters, and set again.	Warning
20C5	8389	The workspace parameter data of the coordinate system parameter is abnormal.	Check the workspace parameters, and set again.	Warning

Error code		Error Description	Action	Alarm/ Warning
Hex	Dec			
20C6	8390	It is a position where coordinate system operation cannot be started.	Move to the position where you can start the coordinate system operation and execute the command.	Warning
20C7	8391	It is a position where coordinate system operation cannot be reach.	Check if there are any abnormalities in the target position or coordinate system parameters and set again.	Warning
20C8	8392	Operating outside the workspace.	Check if there are any abnormalities in workspace parameter position at and set again.	Warning
20C9	8393	The axis group cannot be activated because the unit of axis group configuration is different to coordinate system type.	Set the unit of the configuration axes belonging to the axis group to match the coordinate system type.	Warning
20CA	8394	Coordinate system operation exceeded the maximum interpolation speed.	Execute the command by lowering the command speed so as not to exceed the maximum interpolation speed.	Warning
20CB	8395	Coordinate system operation cannot be executed when the configuration axis is “allowed” for infinite running repeat.	Execute the command after changing the setting of infinite running repeat of configuration axis to "0: Prohibited".	Warning
20CC	8396	Unsupported CoordSystem.	Set up a supported CoordSystem and execute the command.	Warning
20CD	8397	Unsupported TrajType.	Set up a supported TrajType and execute the command.	Warning
20CE	8398	Unsupported ArmConfig	Set supporting ArmConfig and then execute command.	Warning
20CA	8399	TrajTime value exceeds the range.	Set TrajTime value to a larger value that will not excceds maximum interpolation speed and the execute command.	Warning
20D0	8400	The conveyor axis setting value is out of range.	Set conveyor axis (axis and encoder numbers) in the range by product.	Warning
20D1	8401	The axis set as the conveyor axis is set as the axis group configuration axis.	Execute the command with the conveyor axis set to another axis.	Warning
20D2	8402	There is an error in the unit setting of the conveyor axis.	Set the unit of the conveyor shaft to mm/inch.	Warning
20D3	8403	Conveyor axis are not ready to operating	Execute the command when the conveyor axis is ready for operation.	Warning
20D4	8404	Conveyor synchronous command cannot be executed when the main axis is in homing operation.	Execute the command when the conveyor axis is not in operation by homing.	Warning
20D5	8405	Conveyor synchronous command cannot be executed when the main axis is in torque control.	Execute the command when the conveyor axis is not in operation by torque control.	Warning
20D6	8406	Conveyor synchronous function conveyor axis cannot be executed when the configuration axis is “disabled” for infinite running repeat.	Execute the command after changing the setting of infinite running repeat of conveyor axis to "1: enabled".	Warning
20D7	8407	Conveyor synchronous start position and end position shoul be set to different value.	Chagne Conveyor synchronous start position or end position and executes command.	Warning
20D8	8408	Conveyor synchronous start distance value should use “0” or higher.	Set Conveyor synchronous start distance value to 0 or higher, and executes command.	Warning

Error code		Error Description	Action	Alarm/ Warning
Hex	Dec			
20D9	8409	Conveyor synchronous end distance value should use “0” or higher.	Set Conveyor synchronous end distance value to 0 or higher, and executes command.	Warning
20E0	8416	The step value of the coordinate system path operation exceeds the range.	Set the step value of the coordinate’s path operation to a value from 0 to 99 and execute the command.	Warning
20E1	8417	The commandType value of the coordinate system path operation exceeds the range.	Set thecommandType of the coordinates path operation to a value from 0 to 4 and execute the command.	Warning
20E2	8418	The Mode value of the coordinate system path operation exceeds the range.	Set the Mode value of the coordinates path operation to a value from 0 to 2 and execute the command.	Warning
20E3	8419	Coordinate system path operation exceeded the number of possible paths.	Set the step value of the coordinates path operation to a value from 0 to 99 and execute the command.	Warning
20F0	8432	Jog operation direction value excceds the range.	Set the direction value to supported value and then execute the command.	Warning
20F1	8433	Jog operation mode value excceds the range.	Set the mode value to supported value and then execute the command.	Warning
20F2	8434	Jog operation position value excceds the range.	Set the position value to “0” or higher and then execute the command.	Warning
2110	8464	Axis group override command cannot be executed if it is not operating with axis group interpolation or jog operation.	Execute the axis group override command during axis group interpolation or jog operation.	Warning
2111	8465	The override factor of the axis group override command exceeded the range.	Set the VelFactor, AccFactor, and JerkFactor values of the axis group override command to 0 or more and execute the command.	Warning
2112	8466	After reflecting factor of the axis group override command, the axis group operation speed value exceeded the maximum speed value.	Execute override within the range that does not exceed the maximum speed value of the relevant axis.	Warning
2113	8467	After reflecting factor of the axis group override command, the operation speed values belonging to axis group exceeded the maximum speed value.	Execute override within the range that does not exceed the axis included maximum speed value of the axis included in the relevant axis group.	Warning
2120	8480	Parameter number set value of read/write command of axis group parameter is out of range.	Execute the command after the parameter number setting value of the axis group parameters read parameter/write command set between 1 and 41.	Warning
2121	8481	Max interpolation speed setting values of axis group parameters write command is out of range.	Set the maximum speed interpolation value of the axis group as a positive number.	Warning
2122	8482	Coordinate system parameter setting values of axis group parameters write command is out of range.	Check the setting value of the parameter in the 'Coordinate system setting' item among the axis group parameters.	Warning
2123	8483	Workspace setting values of axis group parameter write command is out of range.	Check the setting value of the parameter in the 'workspace setting' item among the axis group parameters.	Warning
2124	8484	PCS setting values of axis group parameter write command is out of range.	Check the setting value of the parameter in the 'PCS setting' item among the axis group parameters.	Warning
2125	8485	The jog speed setting values of axis group parameters write command is out of range.	The jog speed setting value is a positive number, and please set it to less than the maximum interpolation speed setting value.	Warning
2126	8486	Configuration axis setting value of axis group	Execute the command after setting the configuration	Warning

Error code		Error Description	Action	Alarm/ Warning
Hex	Dec			
		parameter write command is out of range.	axis setting value of the axis group parameter to a value between 1 and 36.	
2130	8496	Among the configuration axes, there is an axis that has not completed servo on/off normally.	Check the component axis error code, remove the cause of the error of the relevant axis, and then execute the command again.	Warning
2140	8512	Error occurs during executing the group positioning operation (CSP mode.	Execute command again after checking whether the configuration axis of axis group is the executing status.	Warning
2141	8513	There is axis that the homing does not completed among the configuration axe of axis group to be execute the group positioning operation (CSP Mode).	After homing the configuration axis that does not homing completion executes the command again.	Warning

(6) NC error information

Error code		Error Description	Action	Alarm/ Warning
Hex	Dec			
3000	12288	The NC channel is not ready for operation.	Check if the NC channel is ready for operation. To use NC channel, NC channel must be registered in NC parameter in XG5000.	Warning
3001	12289	NC Program data is abnormal.	Download the data from XG5000 again, and if it occurs repeatedly after re-executing, request A/S.	Alarm
3002	12290	If the NC channel is in automatic operation, the program cannot be written.	If the NC channel is in automatic operation, write the program while automatic operation is stopped.	Warning
3003	12291	NC program writing was not completed normally. (File processing (DELETE) failure when writing NC program)	Download the data from XG5000 again, and if it occurs repeatedly after re-executing, request A/S.	Alarm
3004	12292	NC program writing was not completed normally. (File processing (OPEN) failure when writing NC program)	Download the data from XG5000 again, and if it occurs repeatedly after re-executing, request A/S.	Alarm
3005	12293	NC program writing was not completed normally. (File processing (WRITE) failure when writing NC program)	Download the data from XG5000 again, and if it occurs repeatedly after re-executing, request A/S.	Alarm
3006	12294	NC program writing was not completed normally. (File processing (CLOSE) failure when writing NC program)	Download the data from XG5000 again, and if it occurs repeatedly after re-executing, request A/S.	Alarm
3007	12295	If the NC channel is in automatic operation, the parameter cannot be written.	If the NC channel is in automatic operation, write the parameter while automatic operation is stopped.	Warning
3008	12296	During NC channel automatic operation, the mode of the controller is changed to STOP or ERROR status, so automatic operation cannot be continued.	Check if the mode of the controller is not changed to STOP or ERROR status while the NC channel is in automatic operation.	Alarm
3009	12297	Automatic operation cannot be continued because the EtherCAT network connection is cut during NC channel automatic operation.	Check if the network connection is not disconnected due to slave power error, network cable error, or noise inflow on the network cable while the NC channel is automatic running.	Alarm
300A	12298	During NC channel automatic operation, the controller is stopped by the ESTOP command and automatic operation cannot be continued.	Check that the controller is not stopped by the ESTOP command while the NC channel is in automatic operation.	Alarm
3011	12305	Automatic operation cannot be continued because the NC X axis is not ready for operation.	Check whether the NC X axis is in servo-off status or drive alarm has occurred. NC channel automatic operation can be executed when the configuration axis is servo-on and drive alarm does not occur.	Warning

Error code		Error Description	Action	Alarm/ Warning
Hex	Dec			
3012	12306	Automatic operation cannot be continued because the NC Y axis is not ready for operation.	Check whether the NC Y axis is in servo-off status or drive alarm has occurred. NC channel automatic operation can be executed when the configuration axis is servo-on and drive alarm does not occur.	Warning
3013	12307	Automatic operation cannot be continued because the NC Z axis is not ready for operation.	Check whether the NC Z axis is in servo-off status or drive alarm has occurred. NC channel automatic operation can be executed when the configuration axis is servo-on and drive alarm does not occur.	Warning
3014	12308	Automatic operation cannot be continued because the NC A axis is not ready for operation.	Check whether the NC A axis is in servo-off status or drive alarm has occurred. NC channel automatic operation can be executed when the configuration axis is servo-on and drive alarm does not occur.	Warning
3015	12309	Automatic operation cannot be continued because the NC B axis is not ready for operation.	Check whether the NC B axis is in servo-off status or drive alarm has occurred. NC channel automatic operation can be executed when the configuration axis is servo-on and drive alarm does not occur.	Warning
3016	12310	Automatic operation cannot be continued because the NC C axis is not ready for operation.	Check whether the NC C axis is in servo-off status or drive alarm has occurred. NC channel automatic operation can be executed when the configuration axis is servo-on and drive alarm does not occur.	Warning
3017	12311	Automatic operation cannot be continued because the NC U axis is not ready for operation.	Check whether the NC U axis is in servo-off status or drive alarm has occurred. NC channel automatic operation can be executed when the configuration axis is servo-on and drive alarm does not occur.	Warning
3018	12312	Automatic operation cannot be continued because the NC V axis is not ready for operation.	Check whether the NC V axis is in servo-off status or drive alarm has occurred. NC channel automatic operation can be executed when the configuration axis is servo-on and drive alarm does not occur.	Warning
3019	12313	Automatic operation cannot be continued because the NC W axis is not ready for operation.	Check whether the NC W axis is in servo-off status or drive alarm has occurred. NC channel automatic operation can be executed when the configuration axis is servo-on and drive alarm does not occur.	Warning
301A	12314	Automatic operation cannot be continued because the NC S axis is not ready for operation.	Check whether the NC S axis is in servo-off status or drive alarm has occurred. NC channel automatic operation can be executed when the configuration axis is servo-on and drive alarm does not occur.	Warning
3020	12320	It is an undefined NC channel command.	The NC command is not supported in the current version of the controller. Check the version in which the command can be executed.	Warning
3021	12321	The same NC Channel command was executed, thus canceling the previously executed command.	The corresponding NC channel command can only be executed once per scan. Change the operating condition of the program so that one NC channel command is executed per scan.	Warning
3030	12336	Interpreter (IPR) alarm occurred during NC channel automatic operation, so	Check the interpreter (IPR) error code among the NC channel flags. After resetting the NC channel with the NC	Alarm

Error code		Error Description	Action	Alarm/ Warning
Hex	Dec			
		automatic operation cannot be continued.	reset command (NC_Reset), execute the automatic operation start command (NC_CycleStart) again.	
3031	12337	Program processor (PA) alarm occurred during NC channel automatic operation, so automatic operation cannot be continued.	Check the iProgram processor (PA) error code among the NC channel flags. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation start command (NC_CycleStart) again.	Alarm
3040	12352	During NC channel automatic operation, the command position setting value is out of the pulse unit expression value.	When the command position value was converted into pulse units, it was out of the 32-bit area When the command position value is converted to a pulse, it check for within the range of -2147483648 to 2147483647. (When using the 'Position Control Range Expansion' function, 48-bit INT range can be set.)	Alarm
3041	12353	The command position setting value of NC X axis is out of the pulse unit expression value.	When the command position value of NC X axis was converted into pulse units, it was out of the 32-bit area When the command position value is converted to a pulse, it check for within the range of -2147483648 to 2147483647. (When using the 'Position Control Range Expansion' function, 48-bit INT range can be set.)	Alarm
3042	12354	The command position setting value of NC Y axis is out of the pulse unit expression value.	When the command position value of NC Y axis was converted into pulse units, it was out of the 32-bit area When the command position value is converted to a pulse, it check for within the range of -2147483648 to 2147483647. (When using the 'Position Control Range Expansion' function, 48-bit INT range can be set.)	Alarm
3043	12355	The command position setting value of NC Z axis is out of the pulse unit expression value.	When the command position value of NC Z axis was converted into pulse units, it was out of the 32-bit area When the command position value is converted to a pulse, it check for within the range of -2147483648 to 2147483647. (When using the 'Position Control Range Expansion' function, 48-bit INT range can be set.)	Alarm
3044	12356	The command position setting value of NC A axis is out of the pulse unit expression value.	When the command position value of NC A axis was converted into pulse units, it was out of the 32-bit area When the command position value is converted to a pulse, it check for within the range of -2147483648 to 2147483647. (When using the 'Position Control Range Expansion' function, 48-bit INT range can be set.)	Alarm
3045	12357	The command position setting value of NC B axis is out of the pulse unit expression value.	When the command position value of NC B axis was converted into pulse units, it was out of the 32-bit area When the command position value is converted to a pulse, it check for within the range of -2147483648 to 2147483647. (When using the 'Position Control Range Expansion' function, 48-bit INT range can be set.)	Alarm
3046	12358	The command position setting value of NC C axis is out of the pulse unit expression value.	When the command position value of NC C axis was converted into pulse units, it was out of the 32-bit area When the command position value is converted to a pulse, it check for within the range of -2147483648 to 2147483647. (When using the 'Position Control Range Expansion' function, 48-bit INT range can be set.)	Alarm

Error code		Error Description	Action	Alarm/ Warning
Hex	Dec			
3047	12359	The command position setting value of NC U axis is out of the pulse unit expression value.	When the command position value of NC U axis was converted into pulse units, it was out of the 32-bit area When the command position value is converted to a pulse, it check for within the range of -2147483648 to 2147483647. (When using the 'Position Control Range Expansion' function, 48-bit INT range can be set.)	Alarm
3048	12360	The command position setting value of NC V axis is out of the pulse unit expression value.	When the command position value of NC V axis was converted into pulse units, it was out of the 32-bit area When the command position value is converted to a pulse, it check for within the range of -2147483648 to 2147483647. (When using the 'Position Control Range Expansion' function, 48-bit INT range can be set.)	Alarm
3049	12361	The command position setting value of NC W axis is out of the pulse unit expression value.	When the command position value of NC W axis was converted into pulse units, it was out of the 32-bit area When the command position value is converted to a pulse, it check for within the range of -2147483648 to 2147483647. (When using the 'Position Control Range Expansion' function, 48-bit INT range can be set.)	Alarm
304A	12362	The command position setting value of NC S axis is out of the pulse unit expression value.	When the command position value of NC S axis was converted into pulse units, it was out of the 32-bit area When the command position value is converted to a pulse, it check for within the range of -2147483648 to 2147483647. (When using the 'Position Control Range Expansion' function, 48-bit INT range can be set.)	Alarm
3050	12368	The command position of the NC channel configuration axis is out of the software upper limit position.	Clear the error by executing the error reset command after out of the software upper limit by using the reverse jog command on the NC configuration axis where the error occurred.	Alarm
3051	12369	The command position of the NC X axis is out of the software upper limit position.	Clear the error by executing the error reset command after out of the software upper limit by using the reverse jog command on the NC X.	Alarm
3052	12370	The command position of the NC Y axis is out of the software upper limit position.	Clear the error by executing the error reset command after out of the software upper limit by using the reverse jog command on the NC Y.	Alarm
3053	12371	The command position of the NC Z axis is out of the software upper limit position.	Clear the error by executing the error reset command after out of the software upper limit by using the reverse jog command on the NC Z	Alarm
3054	12372	The command position of the NC A axis is out of the software upper limit position.	Clear the error by executing the error reset command after out of the software upper limit by using the reverse jog command on the NC A.	Alarm
3055	12373	The command position of the NC B axis is out of the software upper limit position.	Clear the error by executing the error reset command after out of the software upper limit by using the reverse jog command on the NC B.	Alarm
3056	12374	The command position of the NC C axis is out of the software upper limit position.	Clear the error by executing the error reset command after out of the software upper limit by using the reverse jog command on the NC C.	Alarm

Error code		Error Description	Action	Alarm/ Warning
Hex	Dec			
3057	12375	The command position of the NC U axis is out of the software upper limit position.	Clear the error by executing the error reset command after out of the software upper limit by using the reverse jog command on the NC U.	Alarm
3058	12376	The command position of the NC V axis is out of the software upper limit position.	Clear the error by executing the error reset command after out of the software upper limit by using the reverse jog command on the NC V.	Alarm
3059	12377	The command position of the NC W axis is out of the software upper limit position.	Clear the error by executing the error reset command after out of the software upper limit by using the reverse jog command on the NC W.	Alarm
305A	12378	The command position of the NC S axis is out of the software upper limit position.	Clear the error by executing the error reset command after out of the software upper limit by using the reverse jog command on the NC S.	Alarm
305B	12379	The command position of the NC channel/axis is out of the inner range of the G22 no-feed zone.	Clear the error by executing the error reset command after out of G22 No Traveling Area by using the jog command on the NC configuration axis where the error occurred.	Alarm
305C	12380	The command position of the NC channel/axis is out of the outer range of the G22 no-feed zone.	Clear the error by executing the error reset command after out of G22 No Traveling Area by using the jog command on the NC configuration axis where the error occurred.	Alarm
305D	12381	The command position of the NC channel/axis is out of range of the 3rd no-feed zone.	Clear the error by executing the error reset command after out of the 3rd no-feed zone by using the jog command on the NC configuration axis where the error occurred.	Alarm
3060	12384	The command position of the NC channel configuration axis is out of the software lower limit position.	Clear the error by executing the error reset command after out of the software lower limit by using the forward direction jog command on the NC configuration axis where the error occurred.	Alarm
3061	12385	The command position of the NC X axis is out of the software lower limit position.	Clear the error by executing the error reset command after out of the software lower limit by using the forward direction jog command on the NC X.	Alarm
3062	12386	The command position of the NC Y axis is out of the software lower limit position.	Clear the error by executing the error reset command after out of the software lower limit by using the forward direction jog command on the NC Y.	Alarm
3063	12387	The command position of the NC Z axis is out of the software lower limit position.	Clear the error by executing the error reset command after out of the software lower limit by using the forward direction jog command on the NC Z.	Alarm
3064	12388	The command position of the NC A axis is out of the software lower limit position.	Clear the error by executing the error reset command after out of the software lower limit by using the forward direction jog command on the NC A.	Alarm
3065	12389	The command position of the NC B axis is out of the software lower limit position.	Clear the error by executing the error reset command after out of the software lower limit by using the forward direction jog command on the NC B.	Alarm
3066	12390	The command position of the NC C axis is out of the software lower limit position.	Clear the error by executing the error reset command after out of the software lower limit by using the forward direction jog command on the NC C.	Alarm
3067	12391	The command position of the NC U axis is out of the software lower limit position.	Clear the error by executing the error reset command after out of the software lower limit by using the forward direction jog command on the NC U.	Alarm

Error code		Error Description	Action	Alarm/ Warning
Hex	Dec			
3068	12392	The command position of the NC V axis is out of the software lower limit position.	Clear the error by executing the error reset command after out of the software lower limit by using the forward direction jog command on the NC V.	Alarm
3069	12393	The command position of the NC W axis is out of the software lower limit position.	Clear the error by executing the error reset command after out of the software lower limit by using the forward direction jog command on the NC W.	Alarm
306A	12394	The command position of the NC S axis is out of the software lower limit position.	Clear the error by executing the error reset command after out of the software lower limit by using the forward direction jog command on the NC S.	Alarm
3071	12401	Automatic operation cannot be continued because the NC X axis is not completion for homing.	Check if the NC X axis is not completion for homing. Use the homing command (MC_Home, NC_Home) to change the axis to the homing completion state.	Warning
3072	12402	Automatic operation cannot be continued because the NC Y axis is not completion for homing.	Check if the NC Y axis is not completion for homing. Use the homing command (MC_Home, NC_Home) to change the axis to the homing completion state.	Warning
3073	12403	Automatic operation cannot be continued because the NC Z axis is not completion for homing.	Check if the NC Z axis is not completion for homing. Use the homing command (MC_Home, NC_Home) to change the axis to the homing completion state.	Warning
3074	12404	Automatic operation cannot be continued because the NC A axis is not completion for homing.	Check if the NC A axis is not completion for homing. Use the homing command (MC_Home, NC_Home) to change the axis to the homing completion state.	Warning
3075	12405	Automatic operation cannot be continued because the NC B axis is not completion for homing.	Check if the NC B axis is not completion for homing. Use the homing command (MC_Home, NC_Home) to change the axis to the homing completion state.	Warning
3076	12406	Automatic operation cannot be continued because the NC C axis is not completion for homing.	Check if the NC C axis is not completion for homing. Use the homing command (MC_Home, NC_Home) to change the axis to the homing completion state.	Warning
3077	12407	Automatic operation cannot be continued because the NC U axis is not completion for homing.	Check if the NC U axis is not completion for homing. Use the homing command (MC_Home, NC_Home) to change the axis to the homing completion state.	Warning
3078	12408	Automatic operation cannot be continued because the NC V axis is not completion for homing.	Check if the NC V axis is not completion for homing. Use the homing command (MC_Home, NC_Home) to change the axis to the homing completion state.	Warning
3079	12409	Automatic operation cannot be continued because the NC W axis is not completion for homing.	Check if the NC W axis is not completion for homing. Use the homing command (MC_Home, NC_Home) to change the axis to the homing completion state.	Warning
307A	12410	Automatic operation cannot be continued because the NC S axis is not completion for homing.	Check if the NC S axis is not completion for homing. Use the homing command (MC_Home, NC_Home) to change the axis to the homing completion state.	Warning
3080	12416	During NC channel automatic operation, automatic operation cannot be continued due to an abnormal drive condition (high/low limit, alarm, servo-off) of the NC component axis.	Remove the cause of the abnormal state after checking that the drive status of the NC configuration axis is not changed to high/low limit, alarm or servo-off status during automatic operation of NC channel.	Warning
3081	12417	During NC channel automatic operation, automatic operation cannot be continued due to an abnormal drive condition	Remove the cause of the abnormal state after checking that the drive status of the NC X is not changed to high/low limit, alarm or servo-off status during automatic operation	Warning

Error code		Error Description	Action	Alarm/ Warning
Hex	Dec			
		(high/low limit, alarm, servo-off) of the NC X.	of NC channel.	
3082	12418	During NC channel automatic operation, automatic operation cannot be continued due to an abnormal drive condition (high/low limit, alarm, servo-off) of the NC Y.	Remove the cause of the abnormal state after checking that the drive status of the NC Y. is not changed to high/low limit, alarm or servo-off status during automatic operation of NC channel.	Warning
3083	12419	During NC channel automatic operation, automatic operation cannot be continued due to an abnormal drive condition (high/low limit, alarm, servo-off) of the NC Z.	Remove the cause of the abnormal state after checking that the drive status of the NC Z is not changed to high/low limit, alarm or servo-off status during automatic operation of NC channel.	Warning
3084	12420	During NC channel automatic operation, automatic operation cannot be continued due to an abnormal drive condition (high/low limit, alarm, servo-off) of the NC A.	Remove the cause of the abnormal state after checking that the drive status of the NC A is not changed to high/low limit, alarm or servo-off status during automatic operation of NC channel.	Warning
3085	12421	During NC channel automatic operation, automatic operation cannot be continued due to an abnormal drive condition (high/low limit, alarm, servo-off) of the NC B.	Remove the cause of the abnormal state after checking that the drive status of the NC B is not changed to high/low limit, alarm or servo-off status during automatic operation of NC channel.	Warning
3086	12422	During NC channel automatic operation, automatic operation cannot be continued due to an abnormal drive condition (high/low limit, alarm, servo-off) of the NC C.	Remove the cause of the abnormal state after checking that the drive status of the NC C is not changed to high/low limit, alarm or servo-off status during automatic operation of NC channel.	Warning
3087	12423	During NC channel automatic operation, automatic operation cannot be continued due to an abnormal drive condition (high/low limit, alarm, servo-off) of the NC U.	Remove the cause of the abnormal state after checking that the drive status of the NC U is not changed to high/low limit, alarm or servo-off status during automatic operation of NC channel.	Warning
3088	12424	During NC channel automatic operation, automatic operation cannot be continued due to an abnormal drive condition (high/low limit, alarm, servo-off) of the NC V.	Remove the cause of the abnormal state after checking that the drive status of the NC V is not changed to high/low limit, alarm or servo-off status during automatic operation of NC channel.	Warning
3089	12425	During NC channel automatic operation, automatic operation cannot be continued due to an abnormal drive condition (high/low limit, alarm, servo-off) of the NC W.	Remove the cause of the abnormal state after checking that the drive status of the NC W is not changed to high/low limit, alarm or servo-off status during automatic operation of NC channel.	Warning
308A	12426	During NC channel automatic operation, automatic operation cannot be continued due to an abnormal drive condition (high/low limit, alarm, servo-off) of the NC S.	Remove the cause of the abnormal state after checking that the drive status of the NC S is not changed to high/low limit, alarm or servo-off status during automatic operation of NC channel.	Warning

Error code		Error Description	Action	Alarm/ Warning
Hex	Dec			
3100	12544	NC Channel parameter data is abnormal.	Download the data from XG5000 again, and if it occurs repeatedly after re-executing, request A/S.	Alarm
3101	12545	Operation cannot be executed due to the NC parameters error.	Check the NC parameters, and if the settings such as the data range are abnormal, set them again.	Alarm
3102	12546	Upper/Lower speed limit of the cutting feed of NC channel parameter exceeded the range.	Upper/Lower speed limit of the cutting feed of NC channel parameter set to a values greater than 0. Set the Upper speed limit of the cutting feed higher than the lower speed limit of the cutting feed.	Alarm
3103	12547	The upper/lower limit cutting speed setting value of the circular milling speed limit in the NC channel parameter exceeded the range.exceeded the range.	Set the upper/lower limit cutting speed values of the circular milling speed limit in the NC channel parameter to a value greater than 0. Set the Upper speed limit of circular interpolation higher than the lower speed limit of circular interpolation.	Alarm
3104	12548	Max/Min Speed of Spindle at Constant Speed Control setting of NC channel parameter exceeded the range.	Set Max. Speed of Spindle at Constant Speed Control setting of NC channel parameter higher than Min Speed of Spindle at Constant Speed Control setting of NC channel parameter.	Alarm
3200	12800	NC Channel /axis parameter data is abnormal.	Download the data from XG5000 again, and if it occurs repeatedly after re-executing, request A/S.	Warning
3300	13056	As Executing reset by NC_Reset or NC_Emergency command, Block Control function is canceled	If needs Blcok control function, NC_Reset or NV_Emergency command is completed and then executes command again.	
3301	13057	When processing reset by NC_Reset or NC_Emergency command, NC_Block Control command can be executed.	After NC_Reset or NC_Emergency command is complete d, executes NC command.	
3310	13072	The NC Feed Hold command was executed in a state other than automatic operation, or the currently executing program block is in a state where NC Feed Hold cannot be executed.	Execute the NC Feed Hold command (NC_FeedHold) while the NC channel is in automatic operation. If automatic operation is currently in progress, please check whether the currently executing program block is in a state where Feed Hold is possible.	Warning
3311	13073	Feed Hold function was canceled because reset by NC_Reset or NC_Emergency command was executed.	If the Feed Hold function is required, execute the command again in the automatic operating status after the NC_Reset or NC_Emergency command is finished.	Warning
3312	13074	NC_FeedHold command cannot be executed while reset by NC_Reset or NC_Emergency command is in progress.	Execute the NC command after the NC_Reset or NC_Emergency command is finished.	Warning
3320	13088	The override factor of the NC rapid traverse override command exceeded the range.	Set the VelFactor, AccFactor, and JerkFactor values of the override command to 0 or more and execute the override command.	Warning

Error code		Error Description	Action	Alarm/ Warning
Hex	Dec			
3321	13089	After the override factor of the NC rapid traverse override command is reflected, the operating speed value of the NC X axis exceeds the speed limit.	After checking the speed limit value of the axis connected to the NC X axis, override the speed limit value within the range.	Warning
3322	13090	After the override factor of the NC rapid traverse override command is reflected, the operating speed value of the NC Y axis exceeds the speed limit.	After checking the speed limit value of the axis connected to the NC Y axis, override the speed limit value within the range.	Warning
3323	13091	After the override factor of the NC rapid traverse override command is reflected, the operating speed value of the NC Z axis exceeds the speed limit.	After checking the speed limit value of the axis connected to the NC Z axis, override the speed limit value within the range.	Warning
3324	13092	After the override factor of the NC rapid traverse override command is reflected, the operating speed value of the NC A axis exceeds the speed limit.	After checking the speed limit value of the axis connected to the NC A axis, override the speed limit value within the range.	Warning
3325	13093	After the override factor of the NC rapid traverse override command is reflected, the operating speed value of the NC B axis exceeds the speed limit.	After checking the speed limit value of the axis connected to the NC B axis, override the speed limit value within the range.	Warning
3326	13094	After the override factor of the NC rapid traverse override command is reflected, the operating speed value of the NC C axis exceeds the speed limit.	After checking the speed limit value of the axis connected to the NC C axis, override the speed limit value within the range.	Warning
3327	13095	After the override factor of the NC rapid traverse override command is reflected, the operating speed value of the NC U axis exceeds the speed limit.	After checking the speed limit value of the axis connected to the NC U axis, override the speed limit value within the range.	Warning
3328	13096	After the override factor of the NC rapid traverse override command is reflected, the operating speed value of the NC V axis exceeds the speed limit.	After checking the speed limit value of the axis connected to the NC V axis, override the speed limit value within the range.	Warning
3329	13097	After the override factor of the NC rapid traverse override command is reflected, the operating speed value of the NC W axis exceeds the speed limit.	After checking the speed limit value of the axis connected to the NC W axis, override the speed limit value within the range.	Warning
332A	13098	After the override factor of the NC rapid traverse override command is reflected, the operating speed value of the NC S axis exceeds the speed limit.	After checking the speed limit value of the axis connected to the NC S axis, override the speed limit value within the range.	Warning
332B	13099	Rapid Traverse override function was canceled because reset by NC_Reset or NC_Emergency command was executed.	If the Rapid Traverse override function is required, execute the command again after the NC_Reset or NC_Emergency command is finished.	Warning

Error code		Error Description	Action	Alarm/ Warning
Hex	Dec			
332C	13100	NC_RapidTraverseOverride command cannot be executed while reset by NC_Reset or NC_Emergency command is in progress.	Execute the NC command after the NC_Reset or NC_Emergency command is finished.	Warning
3330	13104	The override factor of the NC cutting feed override command exceeded the range.	Set the VelFactor, AccFactor, and JerkFactor values of the override command to 0 or more and execute the override command.	Warning
3331	13105	After the override factor of the NC cutting feed override command was reflected, the operating speed value exceeded the cutting feed upper limit speed value.	After checking the cutting feed upper limit speed value of the NC channel parameter, override it within the range that does not exceed the cutting feed upper limit speed value.	Warning
3332	13106	Cutting feed override function was canceled because reset by NC_Reset or NC_Emergency command was executed.	If the cutting feed override function is required, execute the command again after the NC_Reset or NC_Emergency command is finished.	Warning
3333	13107	NC_CuttingFeedOverride command cannot be executed while reset by NC_Reset or NC_Emergency command is in progress.	Execute the NC command after the NC_Reset or NC_Emergency command is finished.	Warning
3340	13120	The override factor of the NC Spindle override command exceeded the range.	Set the VelFactor, AccFactor, and JerkFactor values of the override command to 0 or more and execute the override command.	Warning
3341	13121	After the override factor of the NC Spindle override command is reflected, the operating speed value of Spindle exceeds the speed limit.	After checking the speed limit value of the axis connected to spindle axis, override the speed limit value within the range.	Warning
3342	13122	Spindle override function was canceled because reset by NC_Reset or NC_Emergency command was executed.	If the Spindle override function is required, execute the command again after the NC_Reset or NC_Emergency command is finished.	Warning
3343	13123	NC_SpindleOverride command cannot be executed while reset by NC_Reset or NC_Emergency command is in progress.	Execute the NC command after the NC_Reset or NC_Emergency command is finished.	Warning
3344	13124	The spindle override command cannot be executed because the spindle axis (NC S axis) is not ready for operation.	The spindle axis (NC S axis) is in servo off state or a drive alarm occurs	Warning
3350	13136	The setting axis of the NC parameter read command is not activated as an NC axis.	Check if the setting axis of the NC parameter read command is registered as the NC channel/axis parameter. NC channel/ Axis can be registered in NC Channel parameter among motion data items of XG5000.	Warning
3351	13137	The axis setting value of the NC parameter read command exceeded the allowable range.	Set axis number to a value between 1 and 10. If the axis value is 0, read channel parameters, and if 1 to 10, read NC axis X to NC axis S.	Warning
3352	13138	The Parameter group setting value of the NC parameter read command exceeded the allowable range.	The setting range of the parameter group is 1 to 17 for channel parameters and 1 to 5 for channel/axis parameters. After checking the group number to which the parameter you want to read belongs, execute the parameter read command (NC_ReadParameter).	Warning

Error code		Error Description	Action	Alarm/ Warning
Hex	Dec			
3353	13139	The parameter number set in the parameter group of the NC parameter read command is not supported.	Check whether the parameter number set in the channel parameter or channel/axis parameter group is supported. After checking the group number ans parameter number to which the parameter you want to read belongs, execute the parameter read command (NC_ReadParameter).	Warning
3362	13154	The NC_MirrorImage command was executed in a non-SingleBlock state.	To execute the mirror image (NC_MirrorImage) command during automatic operation, execute it after creating a single block state with the block operation designation (NC_BlockControl) command.	Warning
3370	13168	If the reverse operation buffer is 0, it cannot be executed.	Set the reverse operation buffer value of the NC channel parameter to a value between 1 and 50.	Warning
3380	13184	Block skip function was canceled because reset by NC_Reset or NC_Emergency command was executed.	If the block skip function is required, execute the command again after the NC_Reset or NC_Emergency command is finished.	Warning
3381	13185	The tool position measurable position is out of range. (G37)	Execute the automatic tool measurement (NC_BlockSkip) command after checking the position where the tool position can be measured.	Warning
3500	13568	If the NC channel is in automatic operation, the automatic operation start command cannot be executed.	Check if the NC channel is in automatic operation. Start automatic operation again after the automatic operation is finished,	Warning
3501	13569	If the NC eed Hold command is enabled status, the automatic operation start command cannot be executed.	After releasing the Enable input of the NC Feed Hold command (NC_FeedHold), execute the automatic operation start command (NC_CycleStart) again.	Warning
3502	13570	If the NC emergency stop command is enabled status, the automatic operation start command cannot be executed.	After releasing the Enable input of the NC emergency stop command (NC_Emergency), execute the automatic operation start command (NC_CycleStart) again.	Warning
3503	13571	The automatic operation start command cannot be executed without the NC Interpreter (IPR) normally end.	After resetting the NC channel with the NC reset command (NC_Reset), and run automatic operation start command (NC_CycleStart) again.	Warning
3504	13572	The NC Interpreter (IPR) or Program Processor (PA) cannot execute the automatic operation start command in an error state.	Check the Interpreter (IPR) or Program Processor (PA) error code among the NC channel flags. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation start command (NC_CycleStart) again.	Warning
3505	13573	If the NC channel is not setting to execute program, the automatic operation start command cannot be executed.	After designating the program to be executed with the NC program designation command (NC_LoadProgram), execute the automatic operation start command (NC_CycleStart) again.	Warning
3506	13574	The automatic operation start command cannot be executed because the NC channel has reached the target machining quantity or the target machining quantity in M99 repeat machining.	After checking the machining quantity of the NC channel flag or the machining quantity in case of repeated machining of M99, check whether the target machining quantity has been reached. Execute the automatic operation start command (NC_CycleStart) again after resetting the machining quantity flag or M99 machining quantity flag in case of repeated machining.	Warning
3507	13575	When the NC M/S/T-code output strobe signal is On, the automatic operation start	After completing the NC M/S/T-code output strobe signal, execute the automatic operation start command	Warning

Error code		Error Description	Action	Alarm/ Warning
Hex	Dec			
		command cannot be executed.	(NC_CycleStart).	
3508	13576	The NC Channel Interpreter (IPR) did not run normally.	After resetting the NC channel with the NC reset command (NC_Reset), and run automatic operation start command (NC_CycleStart) again.	Warning
3509	13577	If entering the NC feed prohibited zone, the automatic operation start command cannot be executed.	Clear the error by executing the error reset command after out of feed prohibited zone by using the jog command on the NC configuration axis where the error occurred.	Warning
350A	13578	NC_CycleStart command cannot be executed while reset by NC_Reset or NC_Emergency command is in progress.	Execute the NC command after the NC_Reset or NC_Emergency command is finished.	Warning
350B	13579	The automatic operation start command cannot be executed because the NC spindle device does not support csv or vl mode.	Check whether the slave connected to the spindle axis supports csv or vl operation mode. It cannot be operated as a spindle axis because the spindle device does not support csv or vl mode.	Warning
350C	13580	The automatic operation start command cannot be executed because there is no required object required for spindle operation in the PDO setting of the EtherCAT slave connected to the spindle axis.	Execute the automatic operation start command (NC_CycleStart) after resuming the EtherCAT connection by setting the necessary objects required for spindle operation in the PDO setting of the EtherCAT slave connected to the spindle axis. (Refer to the manual spindle device required PDO setting)	Warning
350D	13581	The automatic operation start command cannot be executed because the reverse operation buffer is all used during reverse operation.	After releasing the enable input of NC_Reset or NC_TetraceMove commands, execute the automatic operation start command (NC_CycleStart) again.	Warning
350F	13583	The automatic operation start command cannot be executed while writing the downloaded NC program during run.	Execute the automatic operation start command after the NC program write operation downloaded during run is completed.	Warning
3510	13584	If the NC channel is not ready for configuration axis, the automatic operation start command cannot be executed.	Execute the automatic operation start command (NC_CycleStart) when all NC channel configuration axes are ready. To start automatic operation, the NC channel configuration axis must be connected to the network or set as a virtual axis.	Warning
3511	13585	If the NC X axis is not ready, the automatic operation start command cannot be executed.	Execute the automatic operation start command (NC_CycleStart) when all NC channel configuration axes are ready. To start automatic operation, the NC channel configuration axis must be connected to the network or set as a virtual axis.	Warning
3512	13586	If the NC Y axis is not ready, the automatic operation start command cannot be executed.	Execute the automatic operation start command (NC_CycleStart) when all NC channel configuration axes are ready. To start automatic operation, the NC channel configuration axis must be connected to the network or set as a virtual axis.	Warning
3513	13587	If the NC Z axis is not ready, the automatic operation start command cannot be executed.	Execute the automatic operation start command (NC_CycleStart) when all NC channel configuration axes are ready. To start automatic operation, the NC channel configuration axis must be connected to the network or set as a virtual axis.	Warning

Error code		Error Description	Action	Alarm/ Warning
Hex	Dec			
3514	13588	If the NC A axis is not ready, the automatic operation start command cannot be executed.	Execute the automatic operation start command (NC_CycleStart) when all NC channel configuration axes are ready. To start automatic operation, the NC channel configuration axis must be connected to the network or set as a virtual axis.	Warning
3515	13589	If the NC B axis is not ready, the automatic operation start command cannot be executed.	Execute the automatic operation start command (NC_CycleStart) when all NC channel configuration axes are ready. To start automatic operation, the NC channel configuration axis must be connected to the network or set as a virtual axis.	Warning
3516	13590	If the NC C axis is not ready, the automatic operation start command cannot be executed.	Execute the automatic operation start command (NC_CycleStart) when all NC channel configuration axes are ready. To start automatic operation, the NC channel configuration axis must be connected to the network or set as a virtual axis.	Warning
3517	13591	If the NC U axis is not ready, the automatic operation start command cannot be executed.	Execute the automatic operation start command (NC_CycleStart) when all NC channel configuration axes are ready. To start automatic operation, the NC channel configuration axis must be connected to the network or set as a virtual axis.	Warning
3518	13592	If the NC V axis is not ready, the automatic operation start command cannot be executed.	Execute the automatic operation start command (NC_CycleStart) when all NC channel configuration axes are ready. To start automatic operation, the NC channel configuration axis must be connected to the network or set as a virtual axis.	Warning
3519	13593	If the NC W axis is not ready, the automatic operation start command cannot be executed.	Execute the automatic operation start command (NC_CycleStart) when all NC channel configuration axes are ready. To start automatic operation, the NC channel configuration axis must be connected to the network or set as a virtual axis.	Warning
351A	13594	If the NC S axis is not ready, the automatic operation start command cannot be executed.	Execute the automatic operation start command (NC_CycleStart) when all NC channel configuration axes are ready. To start automatic operation, the NC channel configuration axis must be connected to the network or set as a virtual axis.	Warning
3520	13600	If the NC channel configuration axis is in operation, the automatic operation start command cannot be executed.	Execute the automatic operation start command (NC_CycleStart) when NC channel configuration axes are stop.	Warning
3521	13601	If the NC X axis is in operation, the automatic operation start command cannot be executed.	Execute the automatic operation start command (NC_CycleStart) when NC channel configuration axes are stop.	Warning
3522	13602	If the NC Y axis is in operation, the automatic operation start command cannot be executed.	Execute the automatic operation start command (NC_CycleStart) when NC channel configuration axes are stop.	Warning
3523	13603	If the NC Z axis is in operation, the automatic operation start command cannot be executed.	Execute the automatic operation start command (NC_CycleStart) when NC channel configuration axes are stop.	Warning
3524	13604	If the NC A axis is in operation, the	Execute the automatic operation start command	Warning

Error code		Error Description	Action	Alarm/ Warning
Hex	Dec			
		automatic operation start command cannot be executed.	(NC_CycleStart) when NC channel configuration axes are stop.	
3525	13605	If the NC B axis is in operation, the automatic operation start command cannot be executed.	Execute the automatic operation start command (NC_CycleStart) when NC channel configuration axes are stop.	Warning
3526	13606	If the NC C axis is in operation, the automatic operation start command cannot be executed.	Execute the automatic operation start command (NC_CycleStart) when NC channel configuration axes are stop.	Warning
3527	13607	If the NC U axis is in operation, the automatic operation start command cannot be executed.	Execute the automatic operation start command (NC_CycleStart) when NC channel configuration axes are stop.	Warning
3528	13608	If the NC V axis is in operation, the automatic operation start command cannot be executed.	Execute the automatic operation start command (NC_CycleStart) when NC channel configuration axes are stop.	Warning
3529	13609	If the NC W axis is in operation, the automatic operation start command cannot be executed.	Execute the automatic operation start command (NC_CycleStart) when NC channel configuration axes are stop.	Warning
352A	13610	If the NC S axis is in operation, the automatic operation start command cannot be executed.	Execute the automatic operation start command (NC_CycleStart) when NC channel configuration axes are stop.	Warning
3530	13616	The automatic operation start command cannot be executed because the NC channel configuration axis is activated as a motion axis group configuration axis.	Execute the automatic operation start command (NC_CycleStart) while the NC channel configuration axis is inactive as a motion axis group configuration axis.	Warning
3531	13617	The automatic operation start command cannot be executed because the NC X axis is activated as a motion axis group configuration axis.	Execute the automatic operation start command (NC_CycleStart) while the NC channel configuration axis is inactive as a motion axis group configuration axis.	Warning
3532	13618	The automatic operation start command cannot be executed because the NC Y axis is activated as a motion axis group configuration axis.	Execute the automatic operation start command (NC_CycleStart) while the NC channel configuration axis is inactive as a motion axis group configuration axis.	Warning
3533	13619	The automatic operation start command cannot be executed because the NC Z axis is activated as a motion axis group configuration axis.	Execute the automatic operation start command (NC_CycleStart) while the NC channel configuration axis is inactive as a motion axis group configuration axis.	Warning
3534	13620	The automatic operation start command cannot be executed because the NC A axis is activated as a motion axis group configuration axis.	Execute the automatic operation start command (NC_CycleStart) while the NC channel configuration axis is inactive as a motion axis group configuration axis.	Warning
3535	13621	The automatic operation start command cannot be executed because the NC B axis is activated as a motion axis group configuration axis.	Execute the automatic operation start command (NC_CycleStart) while the NC channel configuration axis is inactive as a motion axis group configuration axis.	Warning
3536	13622	The automatic operation start command cannot be executed because the NC C axis is activated as a motion axis group	Execute the automatic operation start command (NC_CycleStart) while the NC channel configuration axis is inactive as a motion axis group configuration axis.	Warning

Error code		Error Description	Action	Alarm/ Warning
Hex	Dec			
		configuration axis.		
3537	13623	The automatic operation start command cannot be executed because the NC U axis is activated as a motion axis group configuration axis.	Execute the automatic operation start command (NC_CycleStart) while the NC channel configuration axis is inactive as a motion axis group configuration axis.	Warning
3538	13624	The automatic operation start command cannot be executed because the NC V axis is activated as a motion axis group configuration axis.	Execute the automatic operation start command (NC_CycleStart) while the NC channel configuration axis is inactive as a motion axis group configuration axis.	Warning
3539	13625	The automatic operation start command cannot be executed because the NC W axis is activated as a motion axis group configuration axis.	Execute the automatic operation start command (NC_CycleStart) while the NC channel configuration axis is inactive as a motion axis group configuration axis.	Warning
353A	13626	The automatic operation start command cannot be executed because the NC S axis is activated as a motion axis group configuration axis.	Execute the automatic operation start command (NC_CycleStart) while the NC channel configuration axis is inactive as a motion axis group configuration axis.	Warning
3540	13632	The position unit or velocity unit setting of the NC channel configuration axis is incorrect.	For NC operation, set the unit of NC channel configuration axis (except spindle) to mm or deg. Set the speed unit in RPM for the spindle axis (S axis) and unit/min for other axes (X, Y, Z, A, B, C, U, V, W).	Warning
3541	13633	The position unit or velocity unit setting of the NC X axis is incorrect.	For NC operation, set the unit of NC channel configuration axis to mm or deg. Set the speed unit as unit/min.	Warning
3542	13634	The position unit or velocity unit setting of the NC Y axis is incorrect.	For NC operation, set the unit of NC channel configuration axis to mm or deg. Set the speed unit as unit/min.	Warning
3543	13635	The position unit or velocity unit setting of the NC Z axis is incorrect.	For NC operation, set the unit of NC channel configuration axis to mm or deg. Set the speed unit as unit/min.	Warning
3544	13636	The position unit or velocity unit setting of the NC A axis is incorrect.	For NC operation, set the unit of NC channel configuration axis to mm or deg. Set the speed unit as unit/min.	Warning
3545	13637	The position unit or velocity unit setting of the NC B axis is incorrect.	For NC operation, set the unit of NC channel configuration axis to mm or deg. Set the speed unit as unit/min.	Warning
3546	13638	The position unit or velocity unit setting of the NC C axis is incorrect.	For NC operation, set the unit of NC channel configuration axis to mm or deg. Set the speed unit as unit/min.	Warning
3547	13639	The position unit or velocity unit setting of the NC U axis is incorrect.	For NC operation, set the unit of NC channel configuration axis to mm or deg. Set the speed unit as unit/min.	Warning
3548	13640	The position unit or velocity unit setting of the NC V axis is incorrect.	For NC operation, set the unit of NC channel configuration axis to mm or deg. Set the speed unit as unit/min.	Warning
3549	13641	The position unit or velocity unit setting of the NC W axis is incorrect.	For NC operation, set the unit of NC channel configuration axis to mm or deg. Set the speed unit as unit/min.	Warning
354A	13642	The position unit or velocity unit setting of	For NC operation, set the speed unit of spindle axis to	Warning

Error code		Error Description	Action	Alarm/ Warning
Hex	Dec			
		the NC S axis is incorrect.	RPM.	
3600	13824	Cannot load program because the program set in the NC program designation command does not exist in the controller.	After writing the NC program in XG5000, write it to the controller and then execute the program designation command (NC_LoadProgram) again.	Warning
3601	13825	If the NC channel is in automatic operation, the program designation command cannot be executed.	Check if the NC channel is in automatic operation. Designate a new program after the automatic operation is finished,	Warning
3602	13826	If the NC program data is error, the program designation command cannot be executed.	After checking whether an abnormal NC program data error (0x3001) has occurred, download the data again from XG5000, and if it occurs repeatedly after re-executing, request A/S.	Warning
3603	13827	Invalid LoadMode of NC programming instruction.	After entering a value of 0 in LoadMode of the NC program designation command (NC_LoadProgram), execute the program designation command (NC_LoadProgram) again.	Warning
3604	13828	NC_LoadProgram command cannot be executed while reset by NC_Reset or NC_Emergency command is in progress.	Execute the NC command after the NC_Reset or NC_Emergency command is finished.	Warning
3605	13829	While writing the downloaded NC program during run, the program designation command could not be executed.	Execute the program designation command after the NC program write operation downloaded during run is completed.	Warning
3610	13840	The NC Channel Interpreter (IPR) did not normally reset.	Reset the NC channel again with the NC reset command (NC_Reset). If it occurs repeatedly after retry, please request A/S.	Warning
3620	13856	The NC_Emergency command was executed without automatic operation.	Execute the NC emergency stop command (NC_Emergency) while the NC channel is in automatic operation.	Warning
3630	13872	It is not the range of homing that can be specified in NC homing operation.	The range of the home (ReferenceNum) is from the 1st home to the 4th home. Specify a value between 1 and 4.	Warning
3631	13873	The Nc_home return command cannot be executed when the channel is in automatic operation.	After the automatic operation is finished, execute the origin return command.	Warning
3632	13874	If the NC emergency stop command is enabled status, the home return command cannot be executed.	After releasing the Enable input of the NC emergency stop command (NC_Emergency), execute the home return command (NC_Home) again.	Warning
3633	13875	If the NC channel is not ready for configuration axis, the home return command cannot be executed.	Execute the home return command (NC_Home) when all NC channel configuration axes are ready. To execute home return command, the NC channel configuration axis must be connected to the network or set as a virtual axis.	Warning
3634	13876	The home return command cannot be executed because the NC channel configuration axis is activated as a motion axis group configuration axis.	Execute the home return command (NC_Home) while the NC channel configuration axis is inactive as a motion axis group configuration axis.	Warning
3635	13877	An error occurred during home return in the the servo drive.	Check the error factor of the servo drive, remove the servo drive error, and then execute homing.	Warning

Error code		Error Description	Action	Alarm/ Warning
Hex	Dec			
3636	13878	NC_Home command cannot be executed while reset by NC_Reset or NC_Emergency command is in progress.	Execute the NC command after the NC_Reset or NC_Emergency command is finished.	Warning
3637	13879	When the spindle axis is automatically controlled by the NC function module, if the 'Spindle encoder selection' item of the axis parameter is '0: Disabled', the home operation cannot be executed.	After correctly homing operation encoder connection method connected to the spindle axis in the 'Spindle encoder selection' item of the axis parameter, execute home operation command.	Warning
3638	13880	If the 'Spindle encoder selection' item of the axis parameter is '1: Motor ENC', the 'Position actual value (0x6064)' object does not exist in the TxPDO setting of the EtherCAT slave connected to the spindle axis, so the home operation cannot be executed.	If the 'Spindle encoder selection' item of the axis parameter is '1: Motor ENC', add the 'Position actual value (0x6064)' object to the TxPDO setting of the EtherCAT slave connected to the spindle axis, resume EtherCAT connection, and then execute home operation command.	Warning
3639	13881	If the 'Spindle encoder selection' item of the axis parameter is '2: Built-in ENC1', the setting of the encoder 1 parameter is incorrect and the home operation cannot be executed.	If the 'Spindle encoder selection' item of the axis parameter is '2: Built-in ENC1', set the encoder 1 unit = pulse, Encoder 1 maximum value = 2147483647, Encoder 1 minimum value = -2147483648 and then execute the home operation command.	Warning
363A	13882	If the 'Spindle encoder selection' item of the axis parameter is '3: Built-in ENC2', the setting of the encoder 2 parameter is incorrect and the home operation cannot be executed.	If the 'Spindle encoder selection' item of the axis parameter is '3: Built-in ENC2', set the encoder 2 unit = pulse, Encoder 2 maximum value = 2147483647, Encoder 2 minimum value = -2147483648 and then execute the home operation command.	Warning
3640	13888	NC M-code operation completion command cannot be executed when M-code output strobe signal is Off.	After checking the status of the M-code output strobe signal among the NC channel flags, execute the M-code operation completion command (NC_McodeComplete) when the M-code output strobe signal is On.	Warning
3641	13889	NC_McodeComplete command cannot be executed while reset by NC_Reset or NC_Emergency command is in progress.	Execute the NC command after the NC_Reset or NC_Emergency command is finished.	Warning
3650	13904	NC S-code operation completion command cannot be executed when S-code output strobe signal is Off.	After checking the status of the S-code output strobe signal among the NC channel flags, execute the S-code operation completion command (NC_ScodeComplete) when the S-code output strobe signal is On.	Warning
3651	13905	NC_ScodeComplete command cannot be executed while reset by NC_Reset or NC_Emergency command is in progress.	Execute the NC command after the NC_Reset or NC_Emergency command is finished.	Warning
3660	13920	NC T-code operation completion command cannot be executed when T-code output strobe signal is Off.	After checking the status of the T-code output strobe signal among the NC channel flags, execute the T-code operation completion command (NC_TcodeComplete) when the S-code output strobe signal is On.	Warning
3661	13921	NC_TcodeComplete command cannot be executed while reset by NC_Reset or NC_Emergency command is in progress.	Execute the NC command after the NC_Reset or NC_Emergency command is finished.	Warning
3670	13936	The parameter write command cannot be	Check if the NC channel is in automatic operation.	Warning

Error code		Error Description	Action	Alarm/ Warning
Hex	Dec			
		executed when the channel is in automatic operation.	Execute the NC parameter write command (NC_WriteParameter) in the stop state after automatic operation is finished.	
3671	13937	The setting axis of the NC parameter write command is not activated as an NC axis.	Check if the setting axis of the NC parameter write command is registered as the NC channel/axis parameter. NC channel/ Axis can be registered in NC Channel parameter among motion data items of XG5000.	Warning
3672	13938	The axis setting value of the NC parameter write command exceeded the allowable range.	Set axis number to a value between 1 and 10. If the axis value is 0, write channel parameters, and if 1 to 10, write NC axis X to NC axis S.	Warning
3673	13939	The Parameter group setting value of the NC parameter write command exceeded the allowable range.	The setting range of the parameter group is 1 to 17 for channel parameters and 1 to 5 for channel/axis parameters. After checking the group number to which the parameter you want to write belongs, execute the parameter write command (NC_WriteParameter).	Warning
3674	13940	The parameter number set in the parameter group of the NC parameter write command is not supported.	Check whether the parameter number set in the channel parameter or channel/axis parameter group is supported. After checking the group number and parameter number to which the parameter you want to write belongs, execute the parameter write command (NC_WriteParameter).	Warning
3675	13941	NC The data set value of the parameter set in the NC parameter write command is out of range.	After checking the data setting range of the parameter you want to set, execute the parameter write command (NC_WriteParameter) with a value within the range.	Warning
3676	13942	When NC axis is operating, writing to “axis setting” group parameter item of NC channel/axis parameter can be executed.	When NC axis is not operating, executes writing to to “axis setting” group parameter item of NC channel/axis parameter.	Warning
3690	13968	Only the Cancel Tool Retract/Recover Mode (0), Tool Retract Mode (1) or Tool Recover Mode (2) commands can be commanded.	After specifying Cancel Tool Retract/Recover Mode (0), Tool Retract Mode (1), or Tool Recover Mode (2) in the ToolMode input, execute the Tool Retract/Recover (NC_ToolMode) command.	Warning
3691	13969	Tool Retract cannot be commanded if it is not in tool Recover mode.	Execute the Tool Recover Mode (2) command after tool Retract is completed in Tool Retract Mode (1).	Warning
3692	13970	If the tool Retract operation is not performed in the tool Retract mode, the tool Recover command cannot be commanded.	Execute tool Recover mode (2) command after executing tool Retract operation with JOG operation in tool Retract mode.	Warning
36A0	13984	Block Optional Skip number is out of range. (0~9)	Set the value to be set for SkipNum from 0 to 9 and execute the command.	Warning
36B0	14000	The axis setting value of the manual tool correction command exceeded the allowable range. (X, Y, Z)	Set one of the X~Z (1~3) axes in NcAxis and execute the command.	Warning
36B1	14001	The command cannot be executed because the axis of the manual tool correction command is not prepared as a configuration axis.	Set the axis set in NcAxis as a configuration axis and execute the command.	Warning
36B2	14002	The axis of the manual tool correction	Assign the axis to the NC axis set in NcAxis and execute	Warning

Error code		Error Description	Action	Alarm/ Warning
Hex	Dec			
		command is not activated as an NC axis.	the command.	
36B3	14003	The manual tool correction cannot be executed when the channel is in automatic operation.	After the automatic operation is finished, execute the origin return command.	Warning
36C0	14016	Spindle shaft gear change speed setting value exceeds axis speed limit.	Set the speed to be set in ChgVelocity of the NC_ChgSpindleGear command within the speed limit of the relevant axis and execute the command.	Warning
36C1	14017	Spindle axis maximum speed value is less than or equal to zero.	Set the MaxVelocity value of the NC_ChgSpindleGear command to a value greater than 0 and execute the command.	Warning
36C2	14018	The gear ratio setting value on the motor side is less than or equal to 0.	Set the GearOfMotor value of the NC_ChgSpindleGear command to a value greater than 0 and execute the command.	Warning
36C3	14019	The gear ratio setting value on the mechanical side is less than or equal to 0.	Set the GearOfMachine value of the NC_ChgSpindleGear command to a value greater than 0 and execute the command.	Warning
36C4	14020	Backlash setting is less than 0.	Set the Backlash value of the NC_ChgSpindleGear command to a value greater than or equal to 0 and execute the command.	Warning
36C5	14021	The P Gain setting is less than 0 or greater than 500.	Set the P gain value of the NC_ChgSpindleGear command to a value in the range 0 to 500 and execute the command.	Warning
36C6	14022	The FF Gain setting is less than 0 or greater than 100.	Set the FF gain value of the NC_ChgSpindleGear command to a value in the range 0 to 100 and execute the command.	Warning
36C7	14023	Spindle axis is not in operation with NC automatic operation.	Operate the spindle axis with NC automatic operation and perform NC_ChgSpindleGear operation.	Warning
36C8	14024	When processing rest by NC_Reset or NC_Emergency command, the NC function block can be executed.	After reset process is compleed, execute ChgSpindleGear function block.	Warning
36D0	14032	The spindle control command cannot be executed because the main spindle axis is not set to be automatically controlled by the NC function module.	After setting the spindle axis (S axis) in the NC channel/axis of the NC parameter, set the motor axis number connected to the spindle axis (S axis) in the 'Main spindle axis number' of the NC channel parameter.	Warning
36D1	14033	The spindle control command cannot be executed because the spindle axis is not ready for operation.	The motor axis connected to the spindle axis is not currently ready for operation. After making the axis ready for operation with the LS_Connect command, execute the spindle control command.	Warning
36E0	14048	The SD card is not installed properly.	Please check whether the SD card is properly installed and ready for use.	Warning
36E1	14049	SrcProgramName program does not exist.	Check if the program of SrcProgramName is saved, and enter the program name of the command correctly.	Warning
36E2	14050	DstProgramName program is set to overwrite protection.	To modify the program in DstProgramName, set it to allow overwriting when executing the command.	Warning

Error code		Error Description	Action	Alarm/ Warning
Hex	Dec			
36E3	14051	An error occurred while reading the file.	Please check if the status of the saved file is normal.	Warning
3800	14336	In the NC rapid traverse command, the operating speed value of the configuration axis exceeded the speed limit.	After checking the speed limit value of the axis connected to the NC configuration axis in which the error occurred, set the rapid traverse speed within the range that does not exceed the speed limit.	Warning
3801	14337	In the NC rapid traverse command, the operating speed value of the NC X axis exceeded the speed limit.	After checking the speed limit value of the axis connected to the NC X axis, set the rapid traverse speed within the range that does not exceed the speed limit.	Warning
3802	14338	In the NC rapid traverse command, the operating speed value of the NC Y exceeded the speed limit.	After checking the speed limit value of the axis connected to the NC Y axis, set the rapid traverse speed within the range that does not exceed the speed limit.	Warning
3803	14339	In the NC rapid traverse command, the operating speed value of the NC Z exceeded the speed limit.	After checking the speed limit value of the axis connected to the NC Z axis, set the rapid traverse speed within the range that does not exceed the speed limit.	Warning
3804	14340	In the NC rapid traverse command, the operating speed value of the NC A exceeded the speed limit.	After checking the speed limit value of the axis connected to the NC A axis, set the rapid traverse speed within the range that does not exceed the speed limit.	Warning
3805	14341	In the NC rapid traverse command, the operating speed value of the NC B exceeded the speed limit.	After checking the speed limit value of the axis connected to the NC B axis, set the rapid traverse speed within the range that does not exceed the speed limit.	Warning
3806	14342	In the NC rapid traverse command, the operating speed value of the NC C exceeded the speed limit.	After checking the speed limit value of the axis connected to the NC C axis, set the rapid traverse speed within the range that does not exceed the speed limit.	Warning
3807	14343	In the NC rapid traverse command, the operating speed value of the NC U exceeded the speed limit.	After checking the speed limit value of the axis connected to the NC U axis, set the rapid traverse speed within the range that does not exceed the speed limit.	Warning
3808	14344	In the NC rapid traverse command, the operating speed value of the NC V exceeded the speed limit.	After checking the speed limit value of the axis connected to the NC V axis, set the rapid traverse speed within the range that does not exceed the speed limit.	Warning
3809	14345	In the NC rapid traverse command, the operating speed value of the NC W exceeded the speed limit.	After checking the speed limit value of the axis connected to the NC W axis, set the rapid traverse speed within the range that does not exceed the speed limit.	Warning
380A	14346	In the NC rapid traverse command, the operating speed value of the NC S exceeded the speed limit.	After checking the speed limit value of the axis connected to the NC S axis, set the rapid traverse speed within the range that does not exceed the speed limit.	Warning
380B	14347	During NC rapid traverse operation, the in position of the axes of rapid traverse was not completed within the in position completion monitoring time.	Check the command in position width and the in position completion monitoring time of the NC channel parameters among the parameters of the axes connected to the NC axis.	Warning
380C	14348	During NC rapid traverse operation, automatic operation cannot be continued because there is an error among the NC configuration axes.	Check the axis in which the error occurred among the NC configuration axes. Check the error occurred in the NC axis from the axis error code number of the NC channel/axis flag.	Alarm
3810	14352	In NC feed-per-revolution mode, the	Set the speed of cutting feed operation to a non-zero value	Warning

Error code		Error Description	Action	Alarm/ Warning
Hex	Dec			
		speed of the cutting feed operation is specified as zero.	in NC feed-per-revolution mode.	
3811	14353	The operating speed of the NC cutting feed command exceeded the cutting feed upper limit speed value of the NC channel parameter.	After checking the cutting feed upper limit value of the NC channel parameter, set the cutting feed rate value (F) within the range that does not exceed the parameter value.	Warning
3812	14354	During NC cutting traverse operation, the in position of the axes of cutting traverse was not completed within the in position completion monitoring time.	Check the command in position width and the in position completion monitoring time of the NC channel parameters among the parameters of the axes connected to the NC axis.	Warning
3820	14368	In the NC cutting feed command, the operating speed value of the configuration axis exceeded the speed limit.	After checking the speed limit value of the axis connected to the NC configuration axis in which the error occurred, set the cutting traverse speed within the range that does not exceed the speed limit.	Warning
3821	14369	In the NC cutting traverse command, the operating speed value of the NC X axis exceeded the speed limit.	After checking the speed limit value of the axis connected to the NC X axis, set the cutting traverse speed within the range that does not exceed the speed limit.	Warning
3822	14370	In the NC cutting traverse command, the operating speed value of the NC Y axis exceeded the speed limit.	After checking the speed limit value of the axis connected to the NC Y axis, set the cutting traverse speed within the range that does not exceed the speed limit.	Warning
3823	14371	In the NC cutting traverse command, the operating speed value of the NC Z axis exceeded the speed limit.	After checking the speed limit value of the axis connected to the NC Z axis, set the cutting traverse speed within the range that does not exceed the speed limit.	Warning
3824	14372	In the NC cutting traverse command, the operating speed value of the NC A axis exceeded the speed limit.	After checking the speed limit value of the axis connected to the NC A axis, set the cutting traverse speed within the range that does not exceed the speed limit.	Warning
3825	14373	In the NC cutting traverse command, the operating speed value of the NC B axis exceeded the speed limit.	After checking the speed limit value of the axis connected to the NC B axis, set the cutting traverse speed within the range that does not exceed the speed limit.	Warning
3826	14374	In the NC cutting traverse command, the operating speed value of the NC C axis exceeded the speed limit.	After checking the speed limit value of the axis connected to the NC C axis, set the cutting traverse speed within the range that does not exceed the speed limit.	Warning
3827	14375	In the NC cutting traverse command, the operating speed value of the NC U axis exceeded the speed limit.	After checking the speed limit value of the axis connected to the NC U axis, set the cutting traverse speed within the range that does not exceed the speed limit.	Warning
3828	14376	In the NC cutting traverse command, the operating speed value of the NC V axis exceeded the speed limit.	After checking the speed limit value of the axis connected to the NC V axis, set the cutting traverse speed within the range that does not exceed the speed limit.	Warning
3829	14377	In the NC cutting traverse command, the operating speed value of the NC W axis exceeded the speed limit.	After checking the speed limit value of the axis connected to the NC W axis, set the cutting traverse speed within the range that does not exceed the speed limit.	Warning
382A	14378	In the NC cutting traverse command, the operating speed value of the NC S axis exceeded the speed limit.	After checking the speed limit value of the axis connected to the NC S axis, set the cutting traverse speed within the range that does not exceed the speed limit.	Warning

Error code		Error Description	Action	Alarm/ Warning
Hex	Dec			
3840	14400	NC cannot be operated if NC circular interpolation start point =center point or center point= end point.	In NC circular interpolation, set the position of the center point to a position different from the start point (or end point).	Warning
3841	14401	Invalid radius setting in NC circular interpolation.	The circular radius value in which NC circular interpolation operation can be executed is greater than 0 and less than 2147483647pulse based on the pulse unit. Set the center point or radius input value so that the radius can be calculated within the setting range. (When using the 'Position Control Range Expansion' function, 48-bit INT range can be set.)	Warning
3850	14416	Invalid axis designation in NC cylindrical interpolation.	When performing circular interpolation operation in NC cylindrical interpolation, the Y axis in the XY plane, the Z axis in the YZ plane, and the Z axis in the ZX plane must be designated.	Warning
3860	14432	The Dwell method is specified as the number of revolutions, but the number of revolutions is zero.	Drive the S axis with MC_MoveVelocity in the NC program.	Warning
3870	14448	An error occurred in the spindle axis during NC channel automatic operation, so automatic operation cannot be continued.	Check the error code that occurred in the spindle axis, change the spindle axis to an operable state, and then execute the automatic operation start command (NC_CycleStart).	Warning
3871	14449	The operation mode of the slave connected to the spindle axis cannot be changed.	Check whether the slave connected to the spindle axis supports csv or vl operation mode. Check if the operation status of the slave is normal.	Alarm
3872	14450	Orientation operation cannot be executed because the spindle axis is not home completion.	Execute the NC_Home or MC_SetPosition command on the spindle axis to make the home completion state and then execute the spindle orientation operation.	Warning
3873	14451	When the spindle axis is automatically controlled by the NC function module, if the 'Spindle encoder selection' item of the axis parameter is '0: Disabled', the orientation operation (M19) cannot be executed.	After correctly homing operation encoder connection method connected to the spindle axis in the 'Spindle encoder selection' item of the axis parameter, execute orientation operation command (M19).	Warning
3874	14452	If the 'Spindle encoder selection' item of the axis parameter is '1: Motor ENC', the 'Position actual value (0x6064)' object does not exist in the TxPDO setting of the EtherCAT slave connected to the spindle axis, so orientation operation (M19) cannot be executed.	Add 'Position actual value (0x6064)' object to the TxPDO setting of the EtherCAT slave connected to the spindle axis, resume EtherCAT connection, and then execute Orientation operation (M19).	Warning
3875	14453	If the 'Spindle encoder selection' item of the axis parameter is '2: Built-in ENC1', the setting of the encoder ENC1 is incorrect and Orientation operation (M19) cannot be executed.	If the 'Spindle encoder selection' item of the axis parameter is '2: Built-in ENC1', set the encoder 1 unit = pulse, Encoder 1 maximum value = 2147483647, Encoder 1 minimum value = -2147483648 and then execute the orientation operation command (M19), it.	Warning

Error code		Error Description	Action	Alarm/ Warning
Hex	Dec			
3876	14454	If the 'Spindle encoder selection' item of the axis parameter is '3: Built-in ENC2', the setting of the encoder ENC2 is incorrect and Orientation operation (M19) cannot be executed.	If the 'Spindle encoder selection' item of the axis parameter is '3: Built-in ENC2', set the encoder 2 unit = pulse, Encoder 2 maximum value = 2147483647, Encoder 2 minimum value = -2147483648 and then execute the orientation operation command (M19), it.	Warning
3880	14464	The reference axis of constant speed control is not set. Cannot switch to constant speed control (G96) mode	Execute the command after setting the 'reference axis for constant speed control' setting value of the NC channel parameter to a value within the range of '1: X~9: W'.	Warning
3F00	16128	Interpreter (IPR) Parsing Error - Invalid LEX MAIN TABLE configuration	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F01	16129	Interpreter (IPR) Parsing Error - Undefined character exists.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F02	16130	Interpreter (IPR) parsing error - the number has exceeded the maximum buffer.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F03	16131	Interpreter (IPR) parsing error - tThe number of LEX tokens exceeds the maximum buffer.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F04	16132	Interpreter (IPR) parsing error - more than one decimal point.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F05	16133	Interpreter (IPR) parsing error - the number of parentheses in the formula does not match.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F06	16134	Interpreter (IPR) Parsing Error - an invalid character exists in the formula.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F07	16135	Interpreter (IPR) parsing error - the grammar of the formula is not correct.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the	Alarm

Error code		Error Description	Action	Alarm/ Warning
Hex	Dec			
			automatic operation again.	
3F08	16136	Interpreter (IPR) parsing error - this is not an allowed macro variable.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F09	16137	Interpreter (IPR) parsing error - TANGENT operation error.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F0A	16138	Interpreter (IPR) parsing error - SQUARE ROOT operation error.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F0B	16139	Interpreter (IPR) parsing error - the denominator of division cannot be zero	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F0C	16140	Interpreter (IPR) parsing error - the grammar is not correct.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F0D	16141	Interpreter (IPR) Parsing Error - Invalid YACC MAIN TABLE configuration	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F0E	16142	Interpreter (IPR) parsing error - tThe number of YACC tokens exceeds the maximum buffer.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F0F	16143	Interpreter (IPR) parsing error - Unable to open IPR semaphore.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F10	16144	Interpreter (IPR) parsing error - exited without M02 or M30.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F11	16145	Interpreter (IPR) parsing error - command	After checking the 'error block number' among the NC	Alarm

Error code		Error Description	Action	Alarm/ Warning
Hex	Dec			
		can be given only at the beginning of the block.	channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	
3F12	16146	Interpreter (IPR) parsing error - the same progress block exists.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F13	16147	Interpreter (IPR) parsing error - The number of statements exceeds the maximum buffer	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F14	16148	Interpreter (IPR) parsing error - Could not find next block to proceed.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F15	16149	Interpreter (IPR) parsing error - The subprogram call syntax is incorrect.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F16	16150	Interpreter (IPR) parsing error - The maximum subprogram calls have been exceeded.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F17	16151	Interpreter (IPR) parsing error - The program has already been called.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F18	16152	Interpreter (IPR) parsing error - there is no M99 in the subprogram.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset),execute the automatic operation again.	Alarm
3F19	16153	Interpreter (IPR) parsing error - the grammar of the M99 is not correct.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F1A	16154	Interpreter (IPR) parsing error - the number of loops is too large.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the	Alarm

Error code		Error Description	Action	Alarm/ Warning
Hex	Dec			
			automatic operation again.	
3F1B	16155	Interpreter (IPR) parsing error - there is no start of the loop.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F1C	16156	Interpreter (IPR) parsing error - invalid connection of loop.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F1D	16157	Interpreter (IPR) parsing error - exceeded the M command limit within one block	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F1E	16158	Interpreter (IPR) parsing error - this is an unused G code.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F1F	16159	Interpreter (IPR) parsing error - Simultaneous commands cannot be given to one block.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F20	16160	Interpreter (IPR) parsing error - The center point of the arc could not be found.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F21	16161	Interpreter (IPR) parsing error - It is not possible to create a path for cycle code.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F22	16162	Interpreter (IPR) parsing error - the taper amount of the cycle code is too large.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F23	16163	Interpreter (IPR) parsing error - It cannot be commanded within the cycle shape block.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F24	16164	Interpreter (IPR) parsing error - there is a	After checking the 'error block number' among the NC	Alarm

Error code		Error Description	Action	Alarm/ Warning
Hex	Dec			
		problem with the cycle shape block command	channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	
3F26	16166	Interpreter (IPR) parsing error - the tool offset number is not valid.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F27	16167	Interpreter (IPR) parsing error - the position of the center point of the arc does not match	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F28	16168	Interpreter (IPR) parsing error - It cannot make subprogram calls in MDI mode.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F29	16169	Interpreter (IPR) parsing error - Chamfering and rounding are only applied to the cutting feed command.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F2A	16170	Interpreter (IPR) parsing error - Chamfering and rounding have been duplicated.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F2B	16171	Interpreter (IPR) parsing error - For chamfering and rounding, only single axis command is available.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F2C	16172	Interpreter (IPR) parsing error - the command value for chamfering and rounding is greater than the feed amount.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F2D	16173	Interpreter (IPR) parsing error - When chamfering and rounding, the following block information cannot be obtained.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F2E	16174	Interpreter (IPR) parsing error - When chamfering and rounding, an arc cannot come to the next block.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the	Alarm

Error code		Error Description	Action	Alarm/ Warning
Hex	Dec			
			automatic operation again.	
3F2F	16175	Interpreter (IPR) parsing error - Rounding cannot be performed in the same straight line feed.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F30	16176	Interpreter (IPR) parsing error - compensation start and end can only be traversed in a straight line.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F31	16177	Interpreter (IPR) parsing error - there is no feed command in the cycle shape end block.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F32	16178	Interpreter (IPR) parsing error - there is an axis command independent of the plane during chamfering and rounding.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F33	16179	Interpreter (IPR) parsing error - IJK command limit was exceeded within one block when calling macro.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F34	16180	Interpreter (IPR) parsing error - Modal macros cannot be called from subprograms.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F35	16181	Interpreter (IPR) parsing error - there is exceeded the limit of multiple calls to a modal macro.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F36	16182	Interpreter (IPR) parsing error -This is an unused M code.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F37	16183	Interpreter (IPR) parsing error -Pitch cannot be calculated for rigid tapping	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F38	16184	Interpreter (IPR) parsing error - string	After checking the 'error block number' among the NC	Alarm

Error code		Error Description	Action	Alarm/ Warning
Hex	Dec			
		exceeded max buffer.	channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	
3F39	16185	Interpreter (IPR) parsing error - string construction syntax is incorrect.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F3A	16186	Interpreter (IPR) parsing error - target processing quantity has been reached.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F3B	16187	Interpreter (IPR) parsing error - user stop of macro program.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F3C	16188	Interpreter (IPR) parsing error - It is not possible to create a path for a compound screw cycle.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F3D	16189	Interpreter (IPR) parsing error - It cannot be commanded during polar coordinate interpolation.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F3E	16190	Interpreter (IPR) parsing error - it cannot feed to 0 during polar coordinate interpolation.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F3F	16191	Interpreter (IPR) parsing error - it is a grammatical error when commanding a circular interpolation.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F40	16192	Interpreter (IPR) parsing error - It cannot be commanded during circular interpolation.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm

Error code		Error Description	Action	Alarm/ Warning
Hex	Dec			
3F41	16193	Interpreter (IPR) parsing error - It is a constant peripheral speed control mode in polar coordinates and cylindrical interpolation.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F42	16194	Interpreter (IPR) parsing error - It is not the homing.	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3F43	16195	Interpreter (IPR) parsing error - tool interference has occurred	After checking the 'error block number' among the NC channel flags, check that there is no program error in the corresponding block. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3FE0	16352	Program processor (PA) error - The corresponding pointer location in the program file does not exist.	After resetting the NC channel with the NC reset command (NC_Reset), and run automatic operation again.	Alarm
3FE1	16353	Program processor (PA) error - unable to read from program file.	After resetting the NC channel with the NC reset command (NC_Reset), and run automatic operation again. Alarm	Alarm
3FE2	16354	Program processor (PA) error - there is no program file selected.	Check if the specified program is saved in the controller. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm
3FE3	16355	Program processor (PA) error - Unable to open NcAccess semaphore.	After resetting the NC channel with the NC reset command (NC_Reset), and run automatic operation again.	Alarm
3FE4	16356	Program processor (PA) error - The number of characters in one block is limited to 300.	Check that the number of characters in one block of the specified program does not exceed 300. After resetting the NC channel with the NC reset command (NC_Reset), execute the automatic operation again.	Alarm

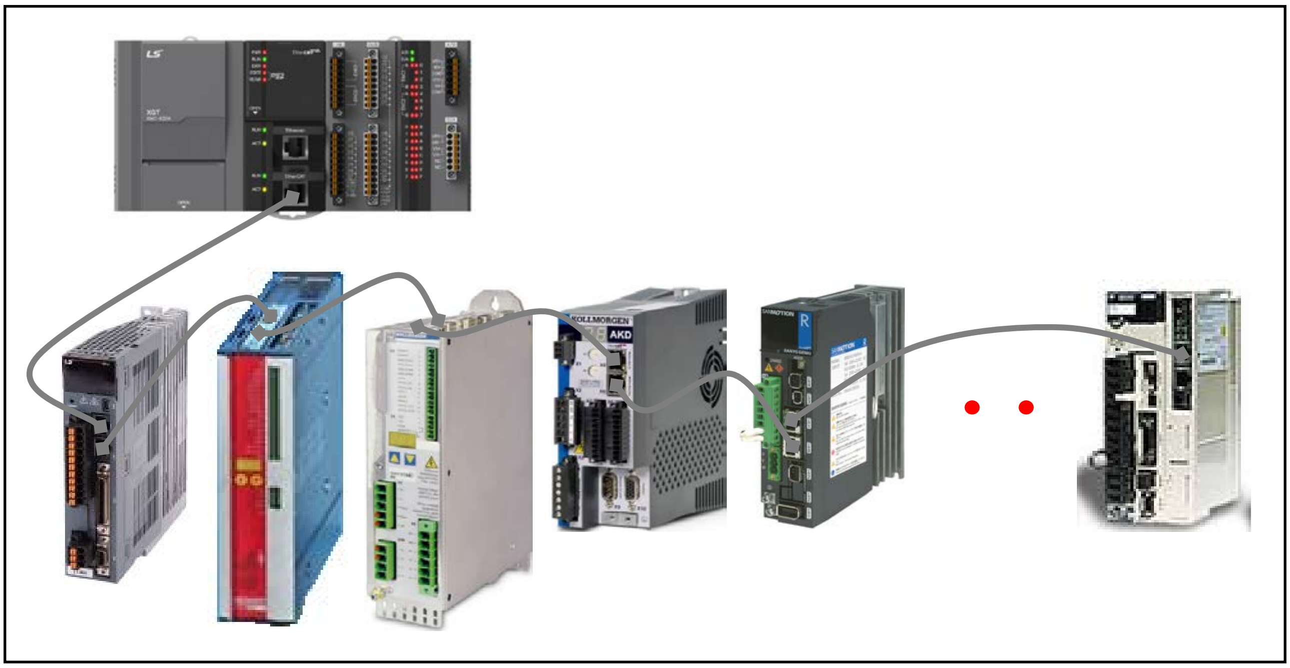
Appendix 3 Setting Example

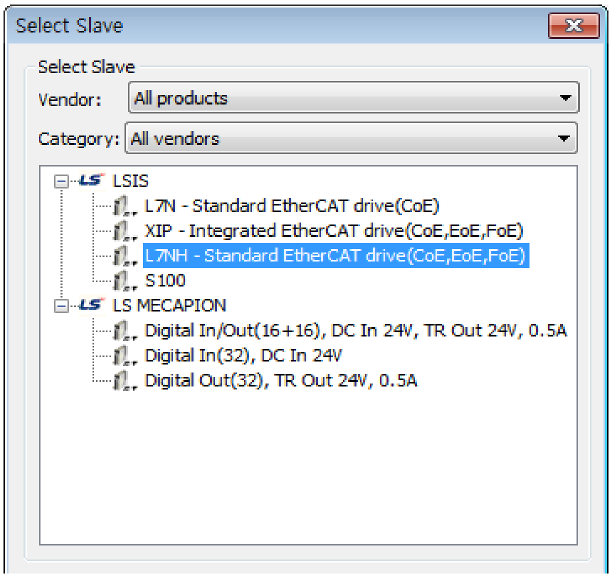
It describes how to set when using the motion controller at the beginning.

(1) Install the servo driver.
Connect the power and motor to the servo driver and connect external signal as necessary.

(2) Install motion controller.
Install motion controller. And at the beginning of test-run, for safety's sake, make sure motion controller is Stop mode.

(3) Connect the motion controller and servo driver.
Connect the motion controller and first servo driver by using Ethernet cable. And connect other servo driver.
At this time, check the I/O direction of communication port of the servo driver distinctly. Below is a list of servo drive which fundamentally has network setting information in the connection and module when servo drive and EtherCAT I/O are connected to motion controller.





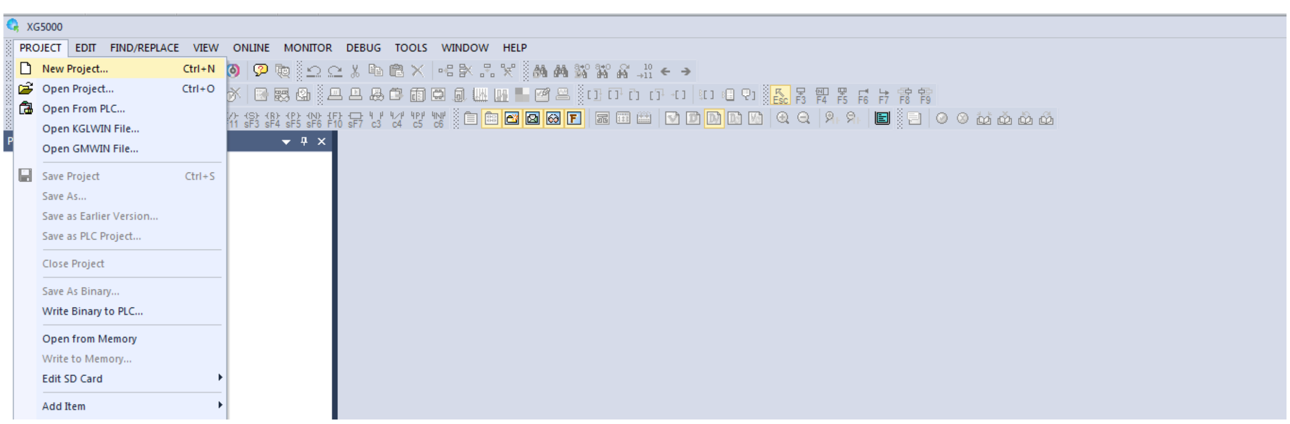
Notes

When the installation of servo drive has completed, make sure to check the following points by using dedicated setting TOOL provided by the servo drive manufacturer; failure to meet the standards requires reset to meet the actual user condition.

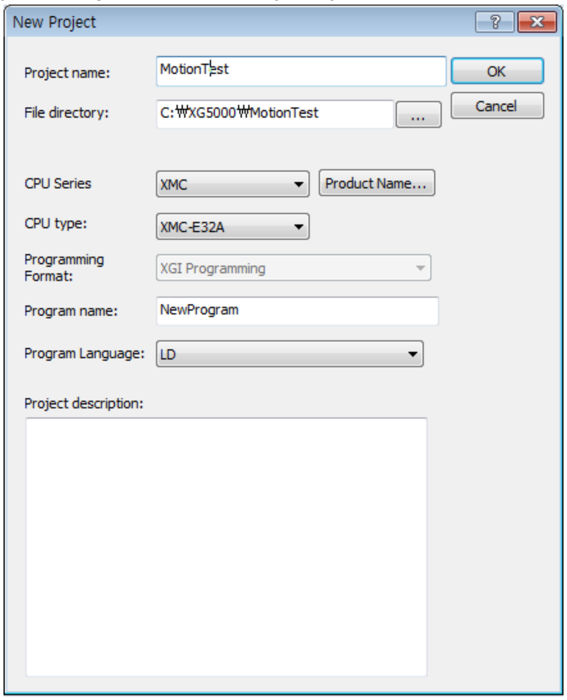
1. Power supply
   Check if the power connected to servo drive and the allowable power conditions are the same.
   (There are instances where no power setting is in parameter depending on the type of servo drive.)
2. The type of motor and encoder(feedback)
   Set the parameter according to the type of encoder and motor connected to actual servo drive.)
3. Command position unit setting
   If it is possible to set the command position unit by servo drive parameter, make sure to set it by pulse unit (Inc. or Counts), and set the encoder resolution value per motor rotation according to the bit number of encoder used.
   (There are instances where no separate setting item exists depending on the type of servo drive.)

(4) Install XG5000 at the PC.

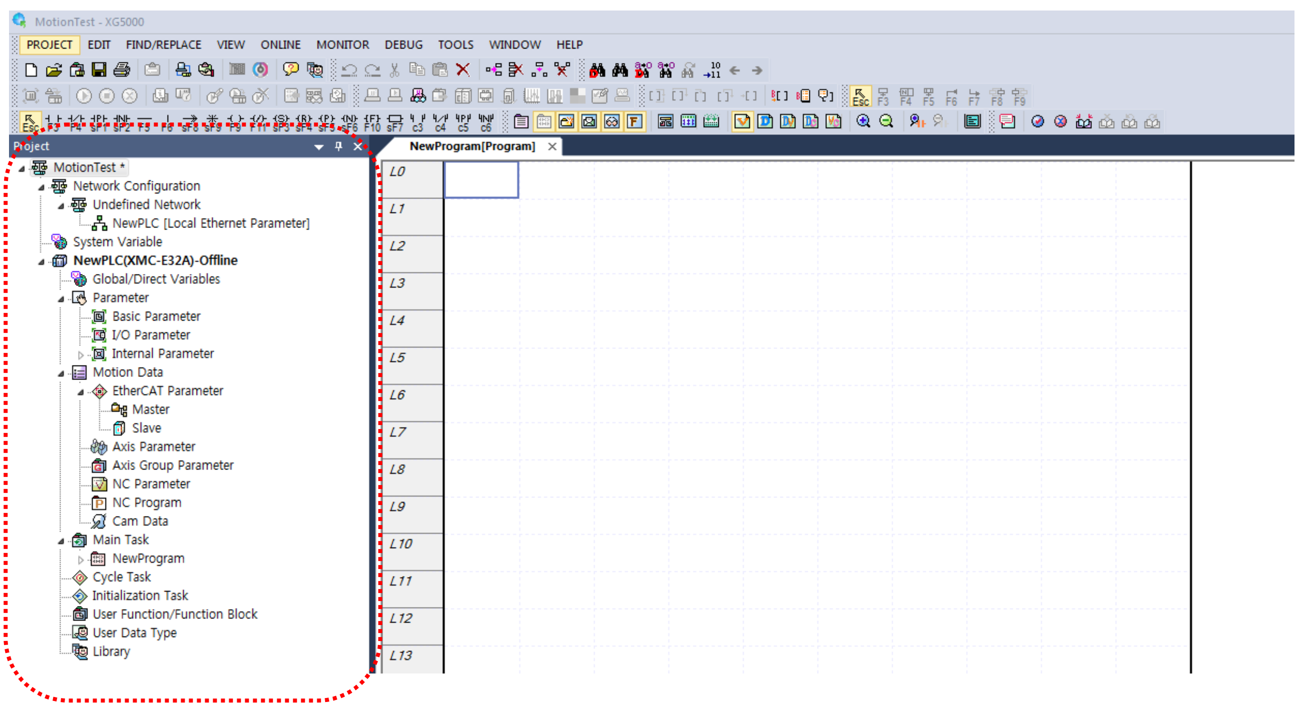
(5) Execute XG5000 and create motion control project by selecting “Project(P) – New Project(N)”.



(6) In the figure below, set up Project name, CPU series, CPU type, Program name to create new project.

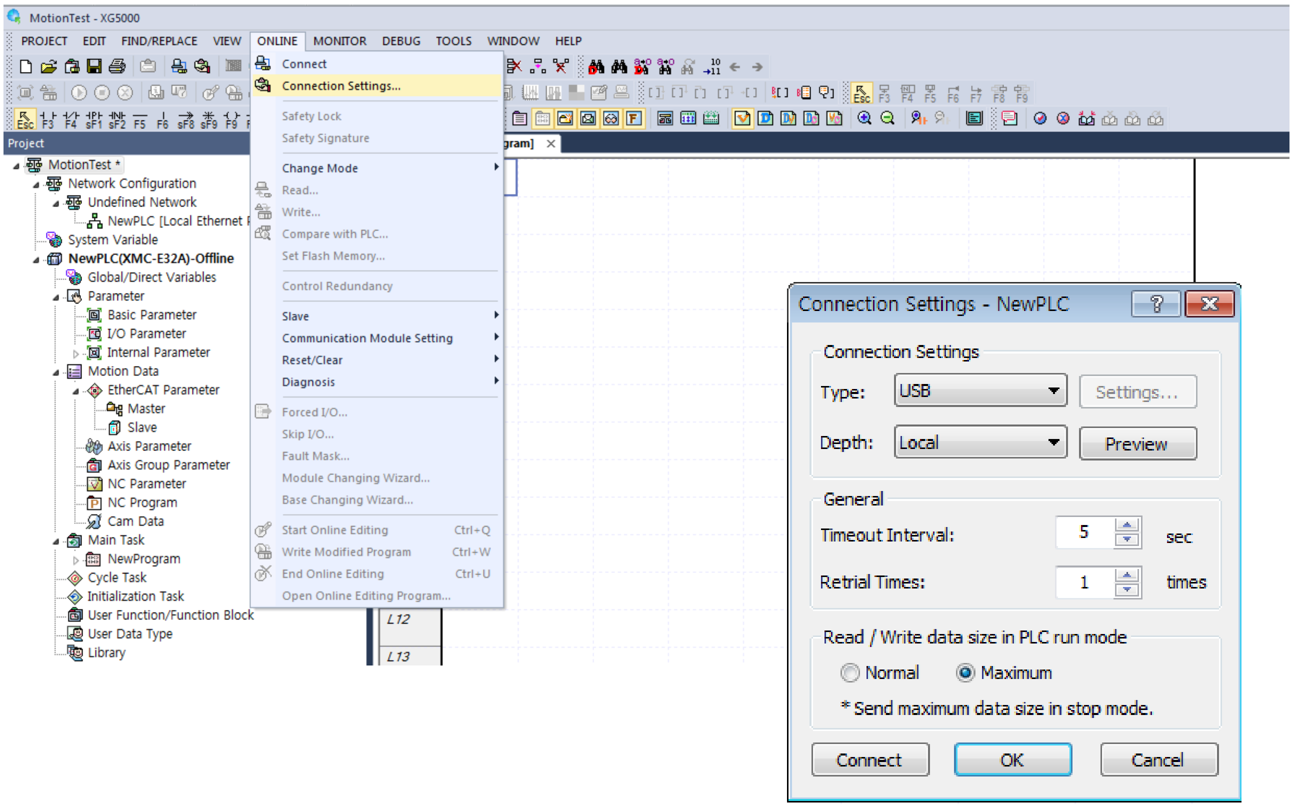


(7) If you set up as the figure above, the project will be created as follows.



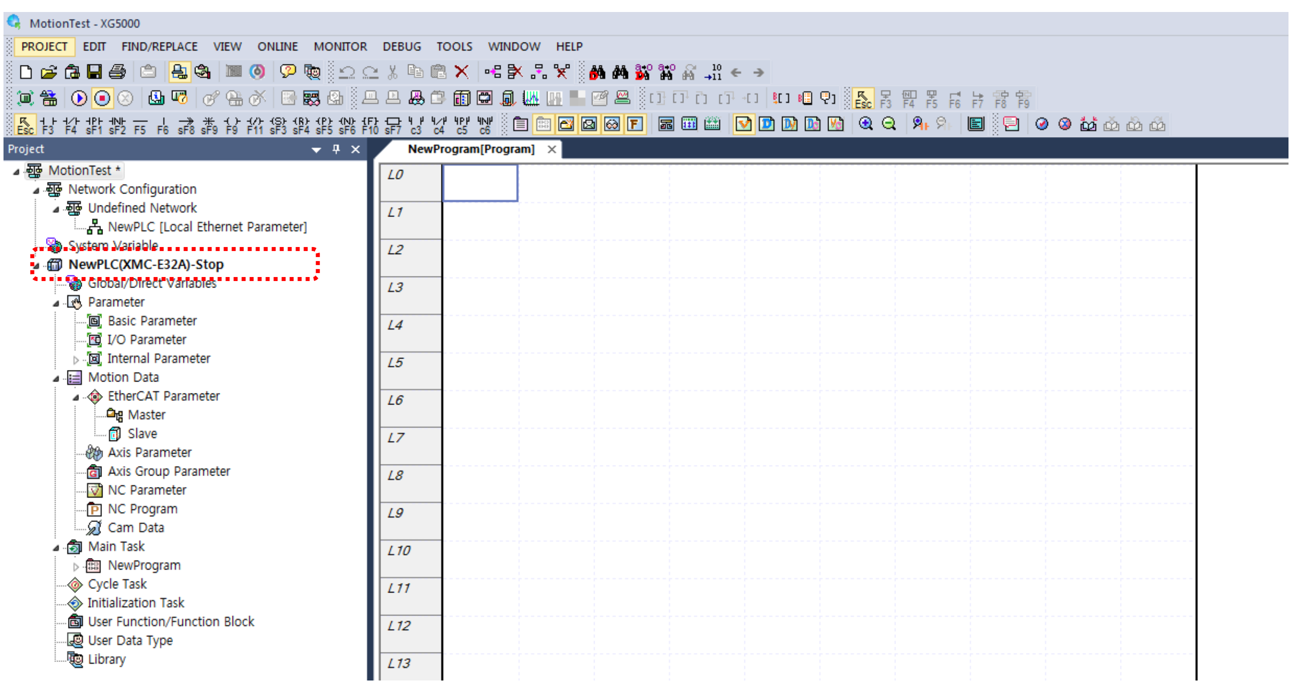
(8) Turn on motion controller and servo driver and connect PC with motion controller through USB or Ethernet cable.

(9) Select “Online(O)- Connection Setting(O)” and set up connection settings.



(10) Select “Online(O)-Connect(N)” to connect PC with motion controller.

(11) If connection is complete, the controller will be shown in ‘Run’ or ‘Stop’ as follows.

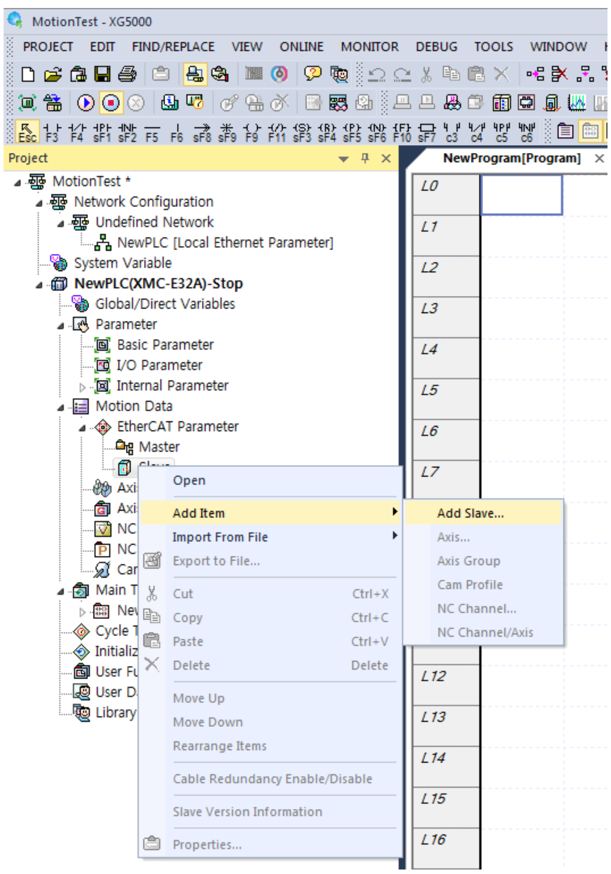


(12) If the controller doesn’t become “Online” and keeps “Offline”, check whether the controller is connected cable, is turned on.

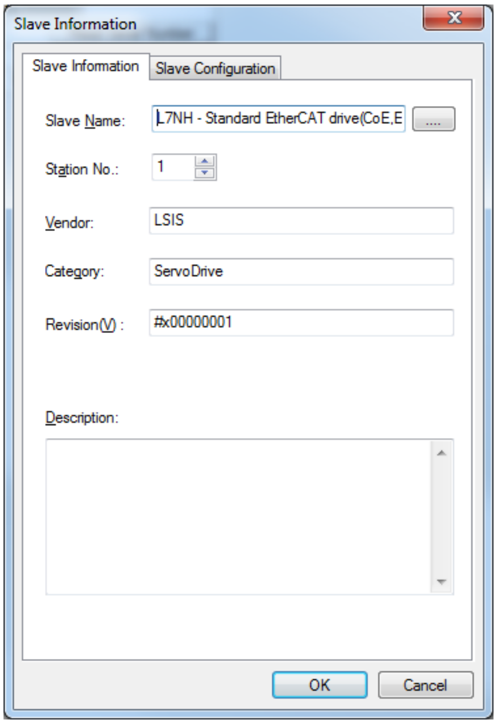
(13) Check if motion controller is in Stop state. If motion controller is in Run state, change it to Stop state and execute the next steps.

(14) Writing must be executed in the motion controller after setting the servo drive actually connected to the network parameter in order to execute the connection with servo drive. First, check if the relevant controller is in off-line state to set network parameter. If it is in on-line state, execute "Online -Disconnect" to change it to off-line state.

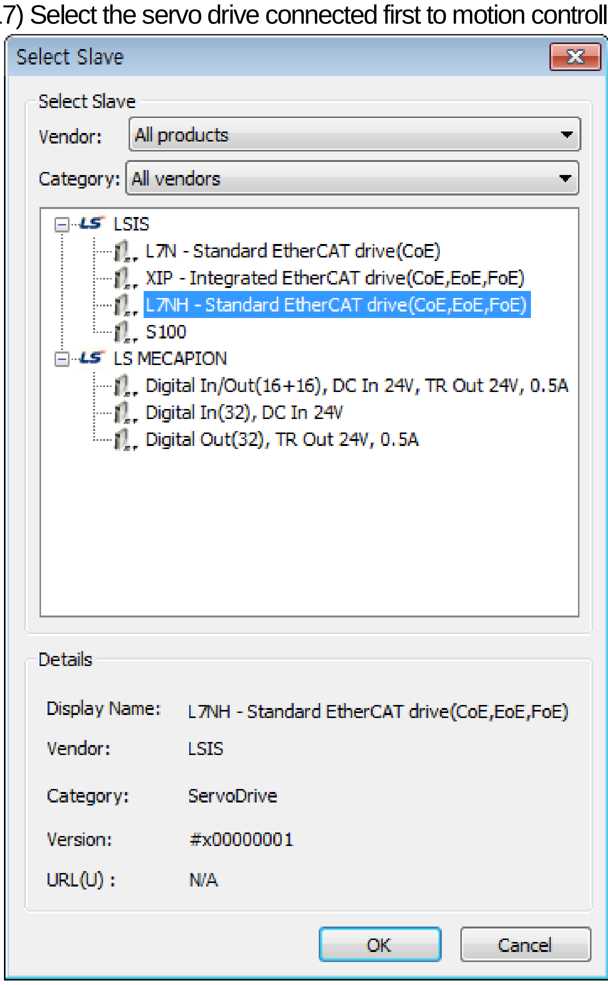
(15) Right click on a mouse in the slave parameter of the project tree and select "Add item – Slave-servo drive" in order to add servo drive to network parameter.



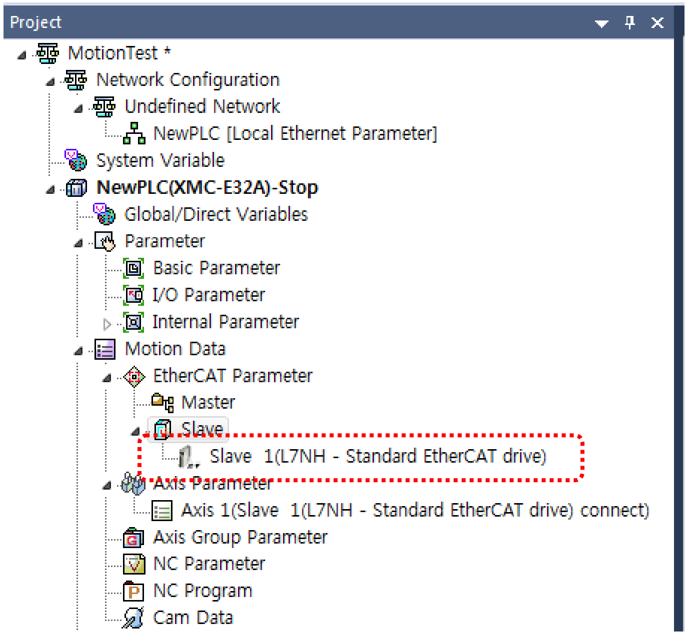
(16) When the slave information window comes up, click the"…" button next to the slave name.



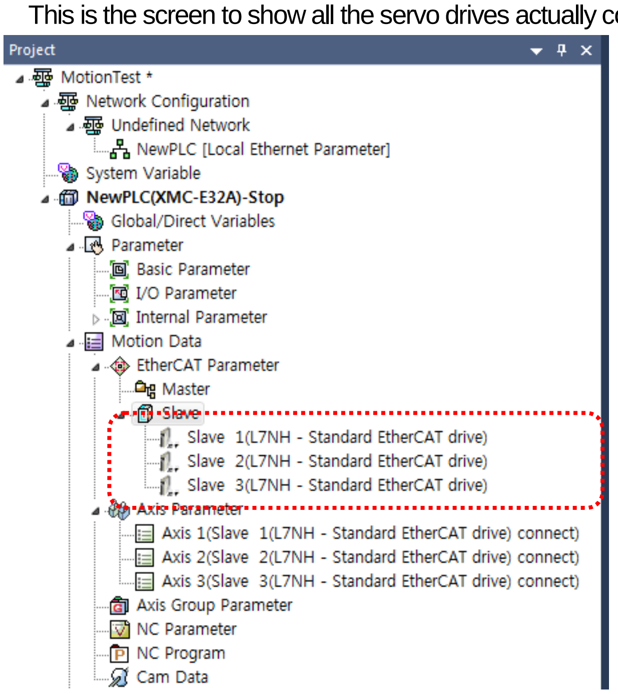
(17) Select the servo drive connected first to motion controller in the servo drive selection window and click OK.



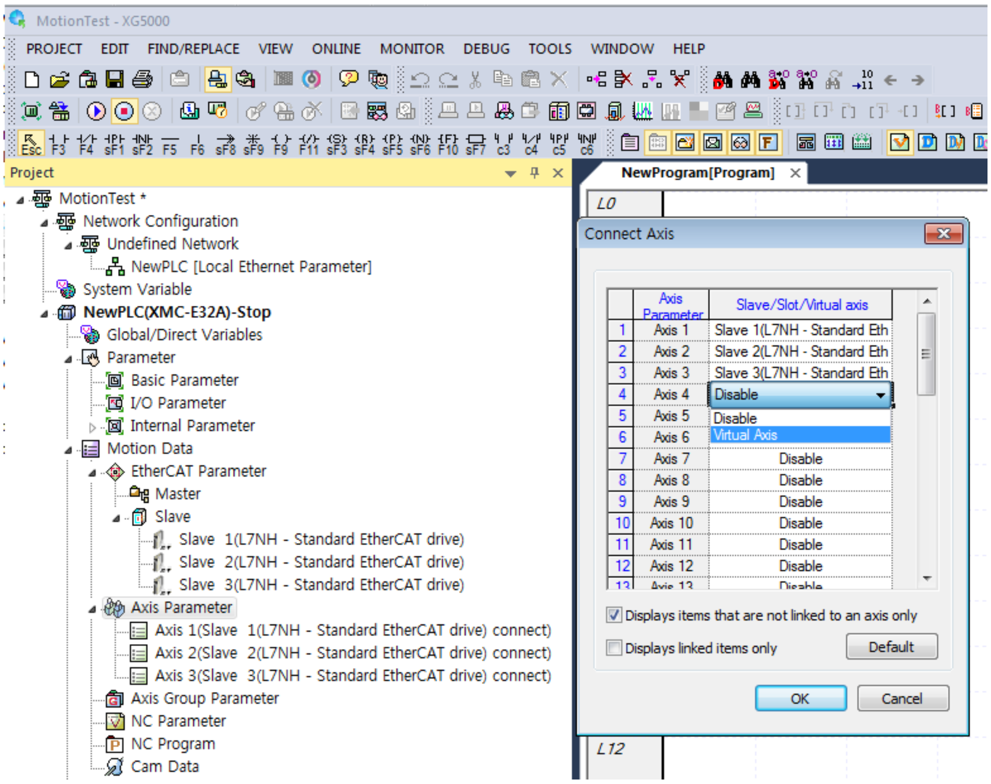
(18) When the axis number setting is completed, the servo drive added earlier is indicated in slave of EtherCAT parameter.



(19) Execute the servo drive addition in the same way for the other servo drives.
This is the screen to show all the servo drives actually connected to slave parameter are added.

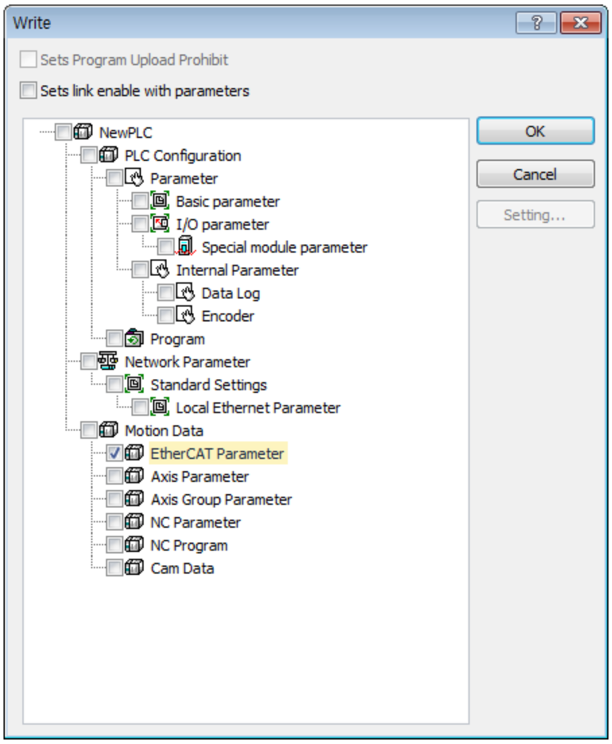


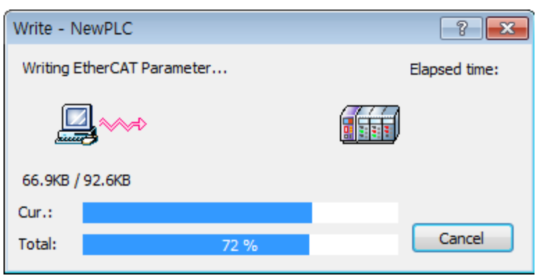
(20) After setting the slave, connect the set slave and the axis to be controlled by the motion controller.
The axes are set in the order that the slave is set but the user can arbitrarily assign slaves to the axes.
Select the axis parameters in the project tree, right-click and select "Axis / slave connection" to create the following window. Here you can assign a slave to the axis. The axis can be set to the set slave and virtual axis

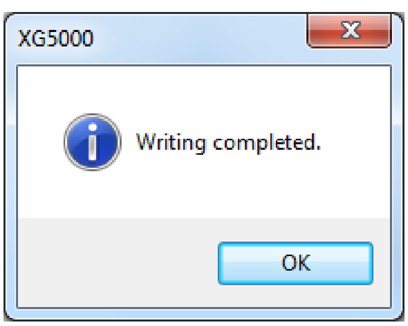


(21) After adding all the EtherCAT salves connected to EtherCAT parameter, execute "Online-Connection" first and execute "Online-Write" to write EtherCAT parameter in motion controller.

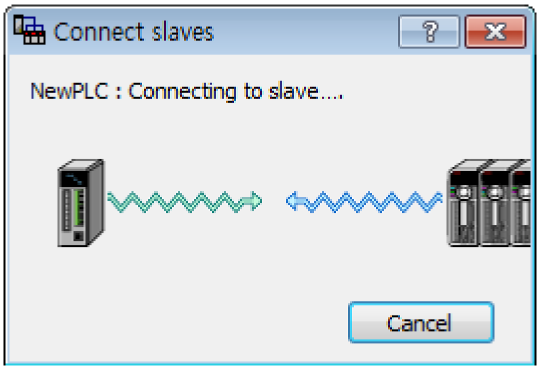
(22) When the project writing window comes up, check in the EtherCAT parameter and check OK to execute writing. This is the screen to show the whole execution process of project writing.



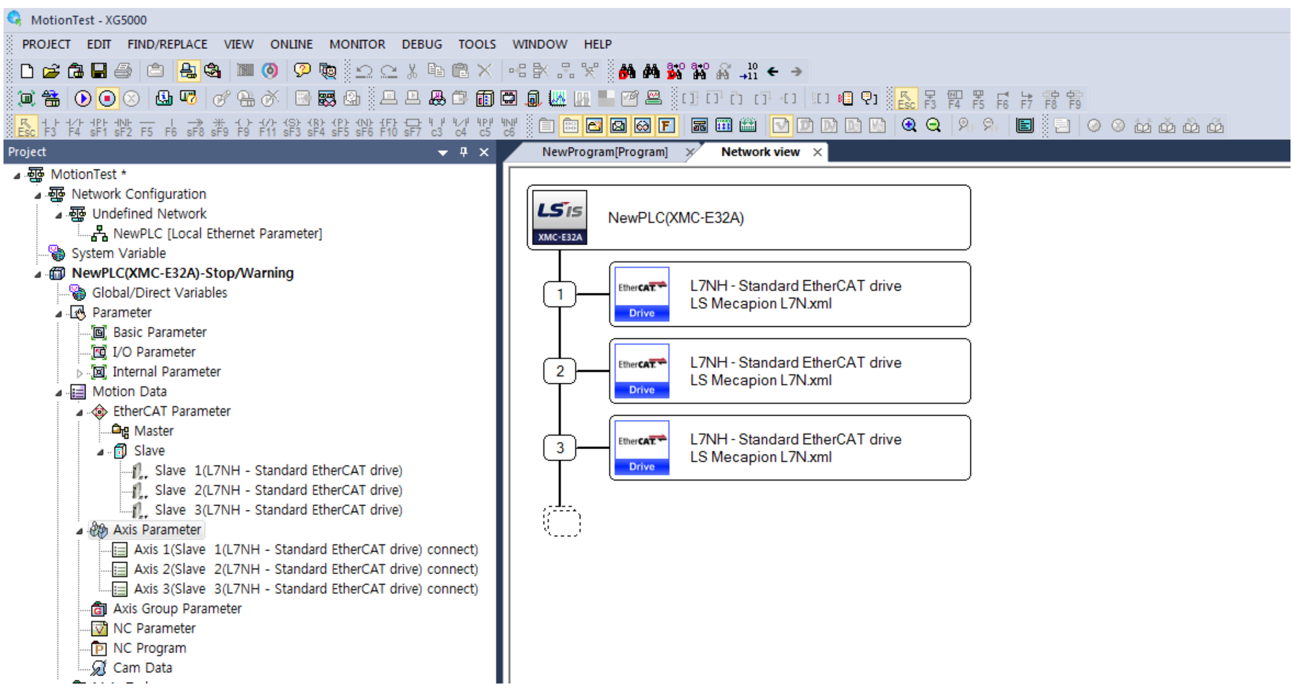




(23) Select "Online – Slave - Connect" to execute communication link between motion controller and servo drive.



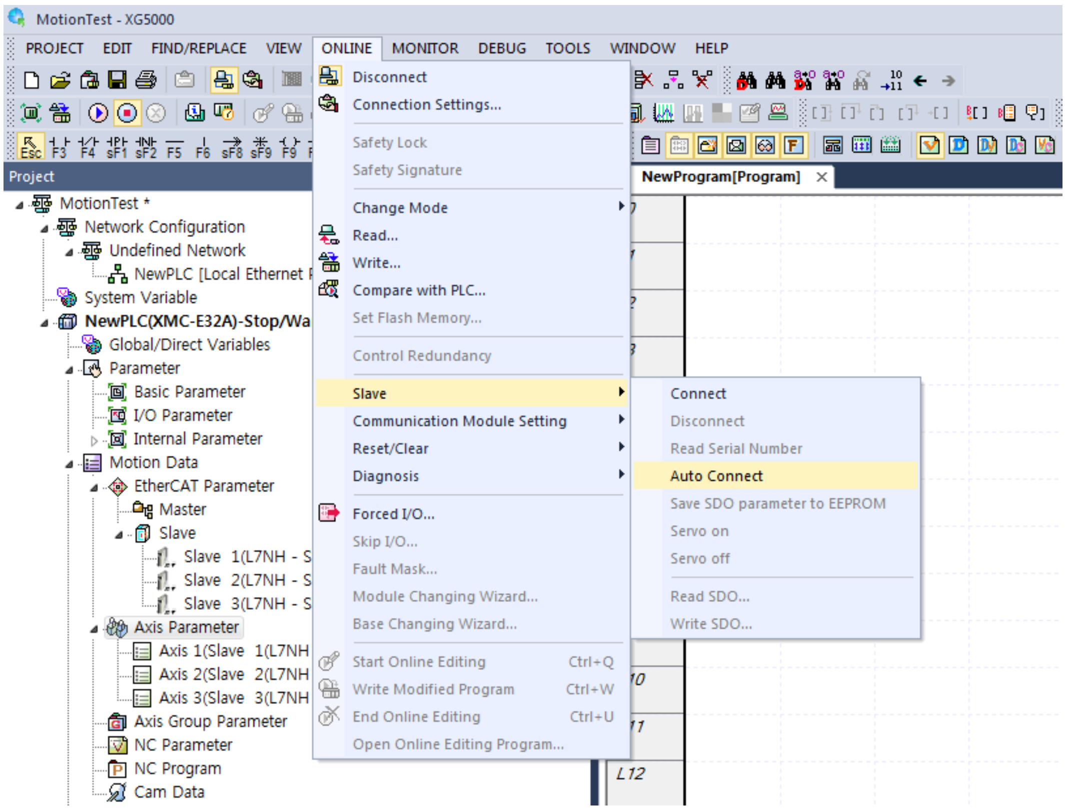
(24) When the link is completed, the servo drive name of slave parameter is activated to black from gray.
Execute the "View – Display EtherCAT Network" in the menu to check the servo drive connection.



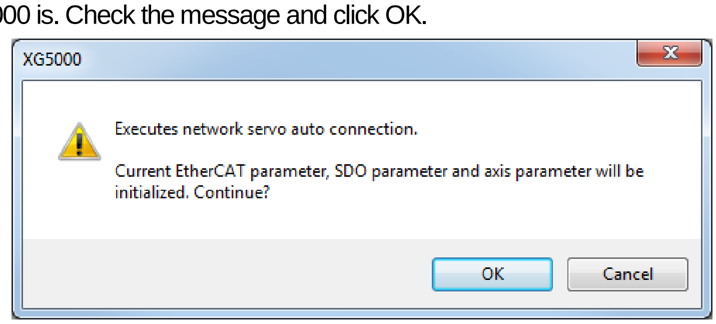
Notes

When connecting the network for the first time after the system configuration using motion controller, use "slave auto connection" to conveniently execute connection to servo drive without setting the EtherCAT slaver.

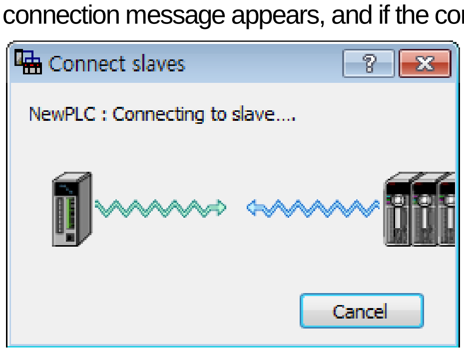
1. Execute the “Online - slave - auto connection” menu.

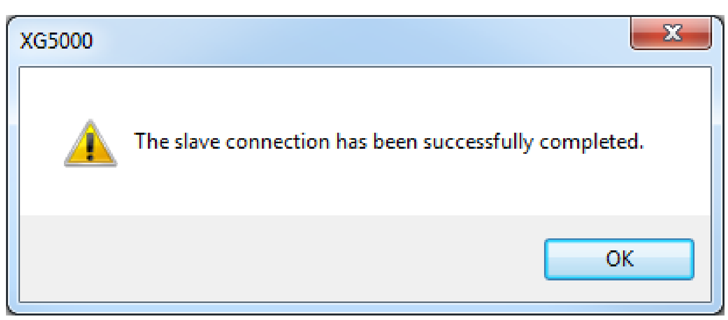


2. Popup notification message appears as follows. This is an alert message notifying when executing slave auto connection, the network parameter set in the current XG5000 and motion controller is initialized and so the servo parameter(SDO parameter) in XG5000 is. Check the message and click OK.



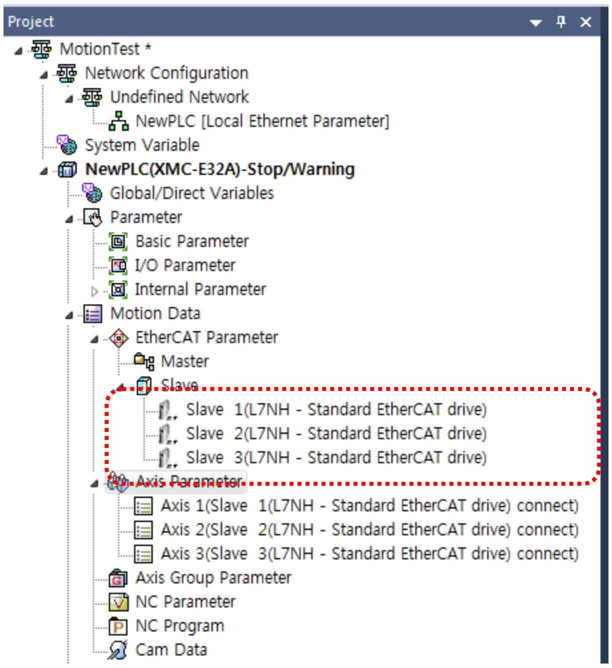
3. Slave connection message appears, and if the connection is completed normally, completion message is indicated.



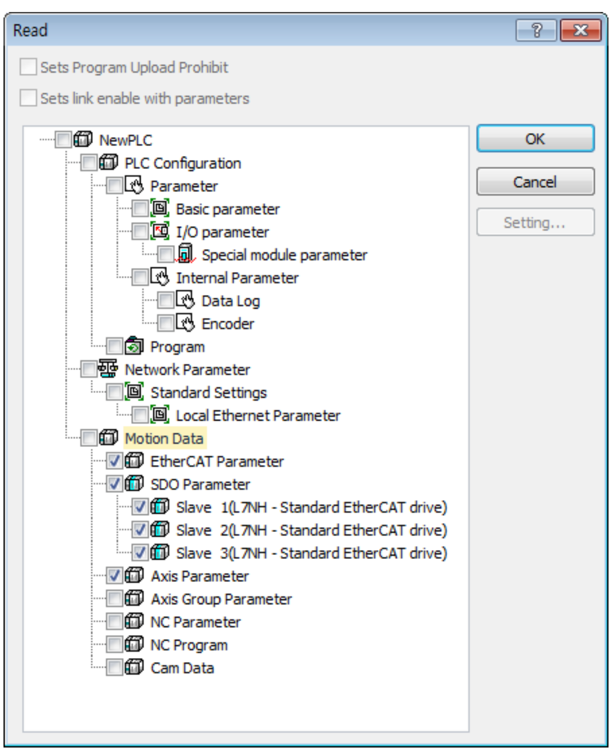


Notes

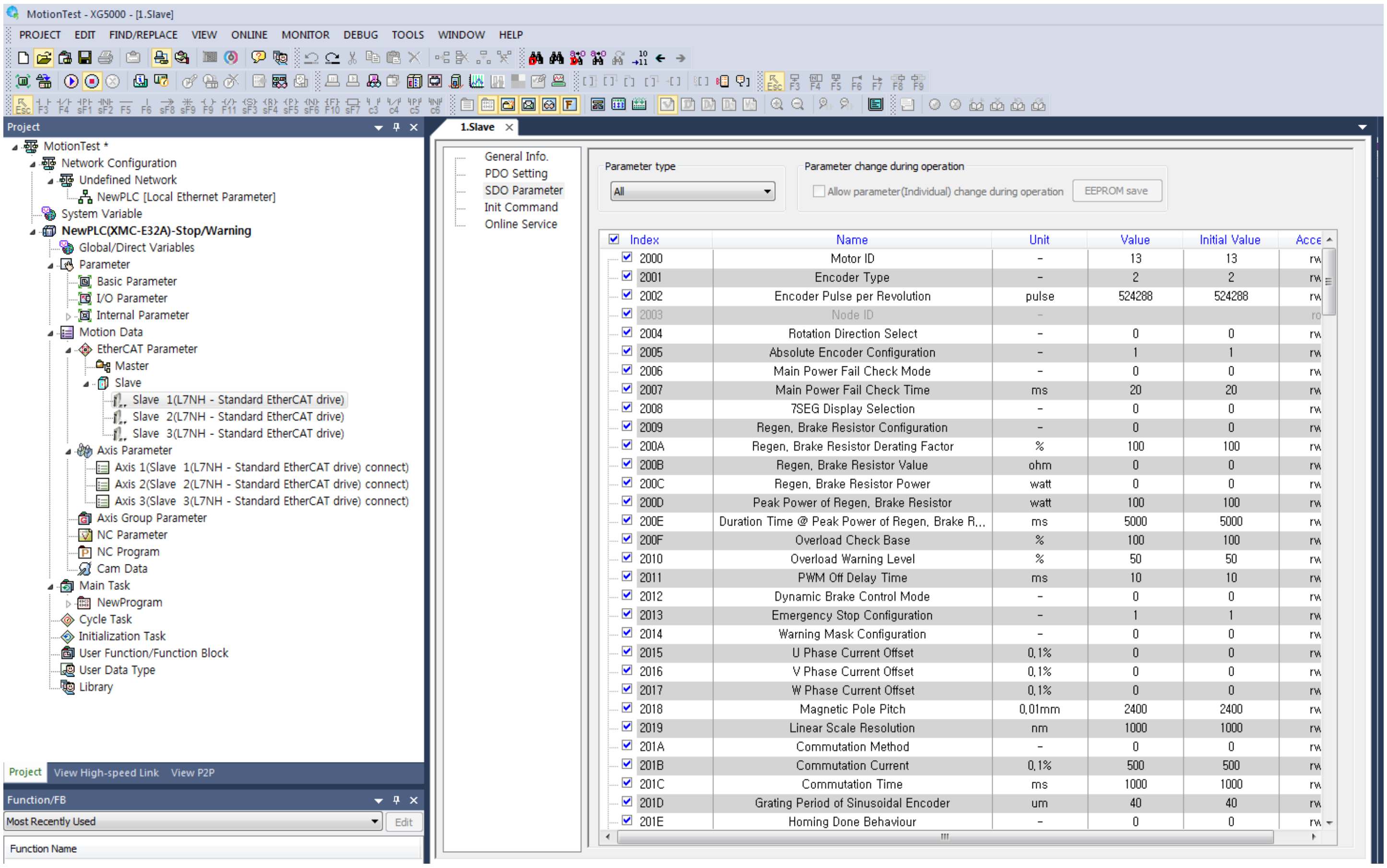
4. When executing the "slave auto connection" command, the EtherCAT slave information currently connected to the EtherCAT parameter slave parameter of XG5000 is automatically registered if the connection command is completed normally.



(25) Read SDO parameter to set operation parameter and SDO parameter of EtherCAT slave.
Select "Online -Read"in the menu and select the item to be read.

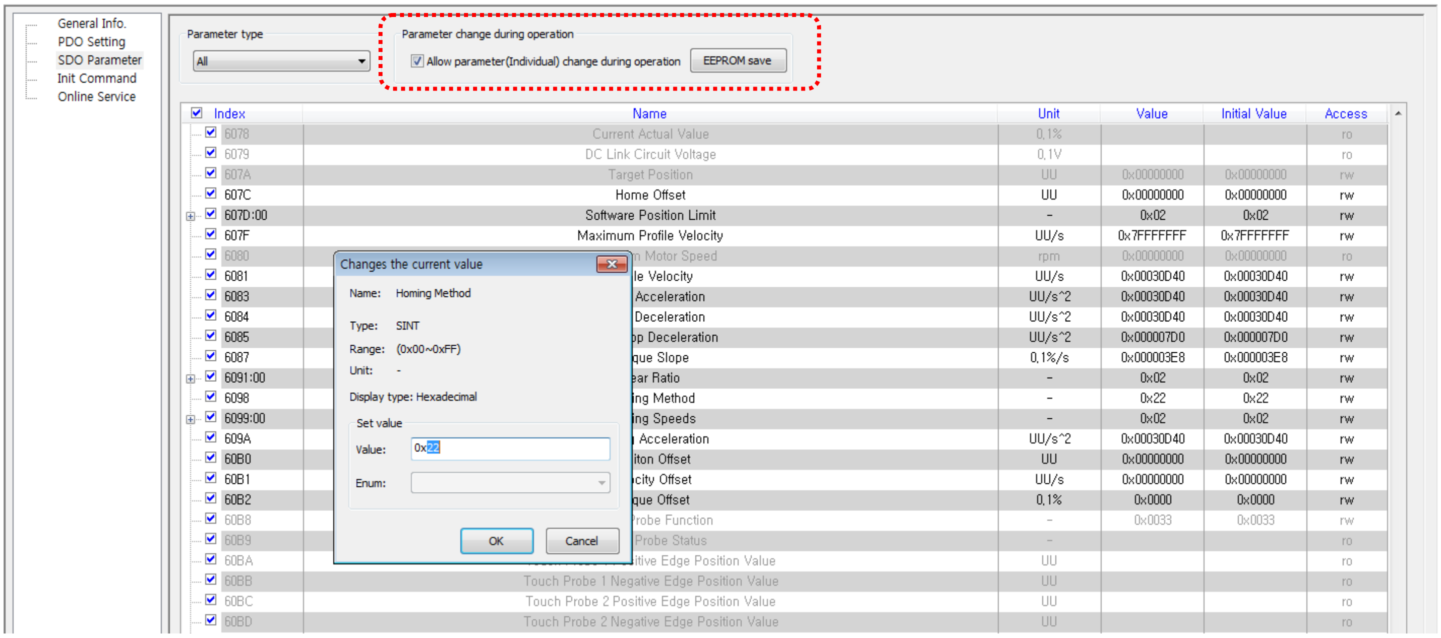


(26) Following is the reading of servo parameter content of L7NH servo drive. The content of servo parameter can differ depending on the types of servo drive. Refer to the instruction manuals of each servo drive for details.



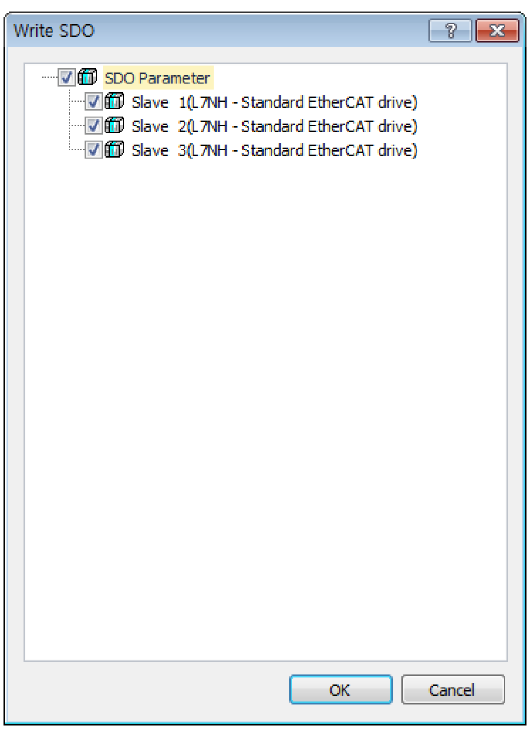
(27) SDO parameter can be set in two ways.

First method is only to change the value of one item of SDO parameter; select the 'Allow SDO Parameter(Individual) Change during Operation' checkbox and set the SDO parameter value that you want to change, then the set value is applied to slave(servo drive) immediately. Reflection of the modified value to the 'current value' column of SDO parameter means the value is transmitted normally.



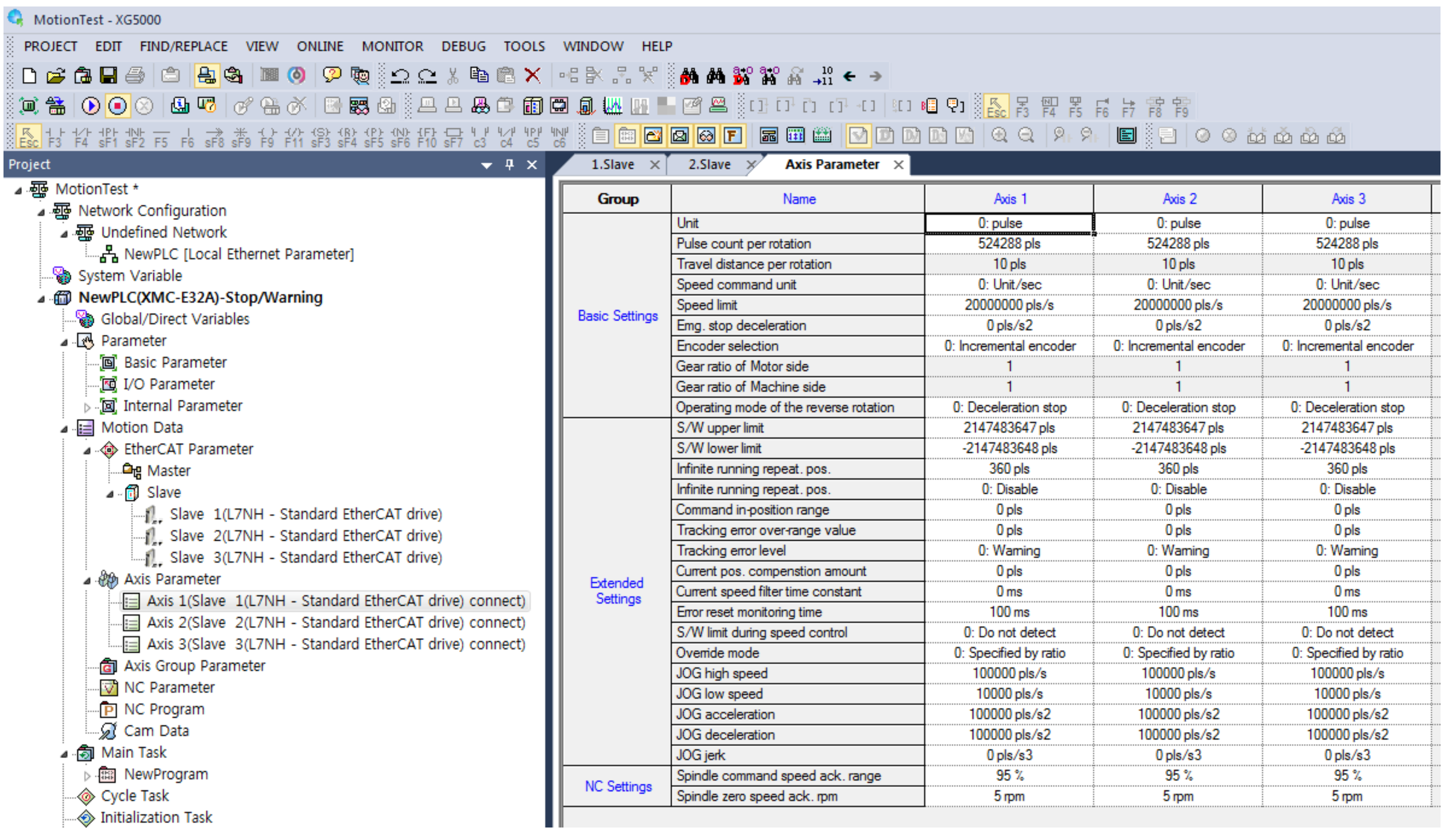
In order to keep the data after turn on/off the power of slave(servo drive), execute the "Online-Save slave parameter to EEPROM" command because modifying the parameter in operation of SDO parameter (individual) is only valid when the power is currently on.

Second method is to set all the SDO parameter you want to modify and execute 'Online -Write ' to write the whole SDO parameter in slave(servo drive) at a time.



When writing the whole SDO parameter, "Save SDO parameter to EEPROM" command is automatically executed. Therefore, you do not need to execute "SDO parameter to EEPROM" separately. Refer to the instruction manual of the relevant slave(servo drive) because sometimes modified set value is applied after the power is on/off depending on the item of SDO parameter.

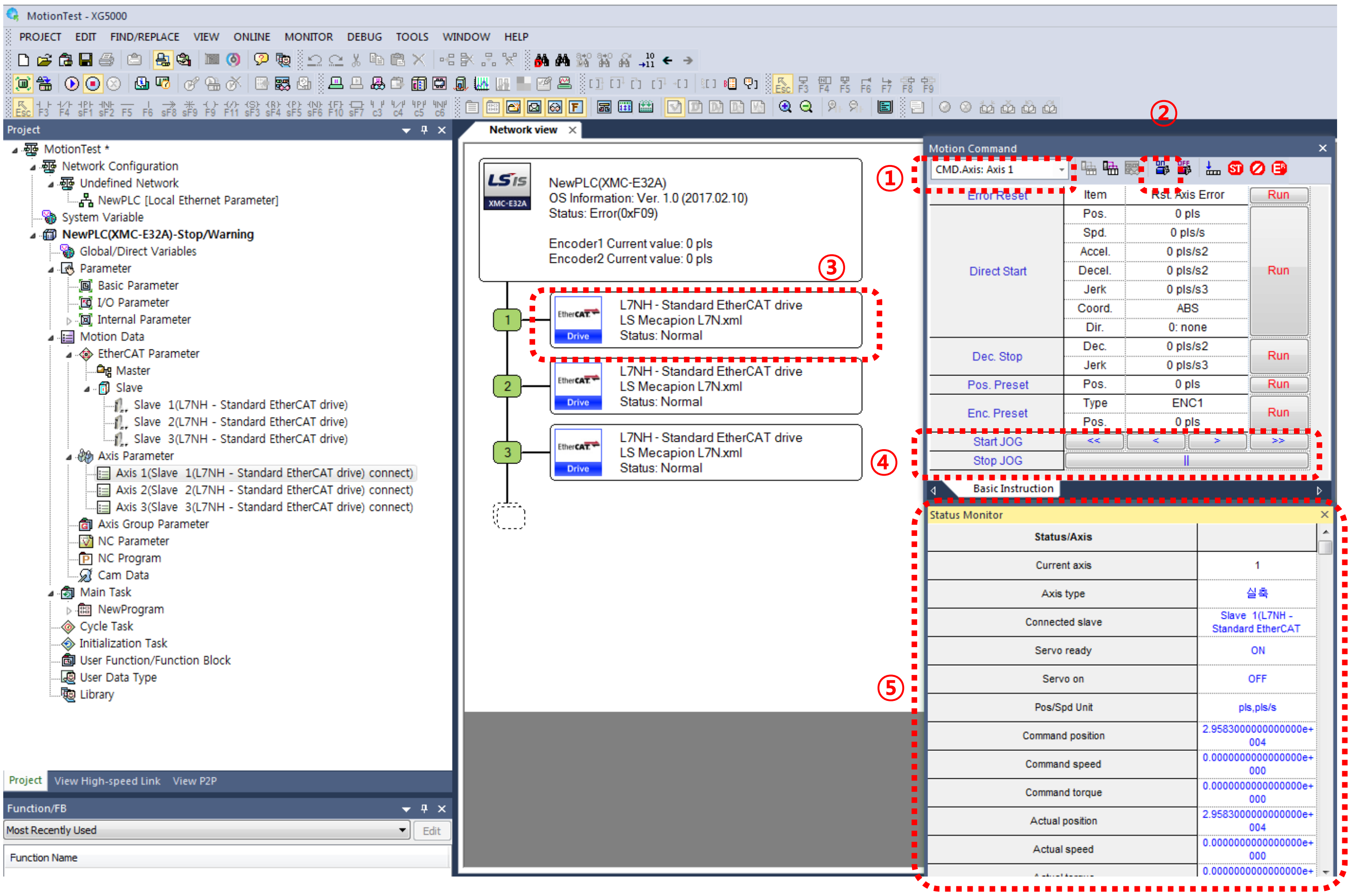
(28) When finishing the SDO parameter setting, set the operation parameter of each axis and select the operation parameter of the relevant axis in "Online-Write " to write in controller.



Group	Name	Axis 1	Axis 2	Axis 3
Basic Settings	Unit	0: pulse	0: pulse	0: pulse
	Pulse count per rotation	524288 pls	524288 pls	524288 pls
	Travel distance per rotation	10 pls	10 pls	10 pls
	Speed command unit	0: Unit/sec	0: Unit/sec	0: Unit/sec
	Speed limit	20000000 pls/s	20000000 pls/s	20000000 pls/s
	Emg. stop deceleration	0 pls/s2	0 pls/s2	0 pls/s2
	Encoder selection	0: Incremental encoder	0: Incremental encoder	0: Incremental encoder
	Gear ratio of Motor side	1	1	1
	Gear ratio of Machine side	1	1	1
	Operating mode of the reverse rotation	0: Deceleration stop	0: Deceleration stop	0: Deceleration stop
Extended Settings	S/W upper limit	2147483647 pls	2147483647 pls	2147483647 pls
	S/W lower limit	-2147483648 pls	-2147483648 pls	-2147483648 pls
	Infinite running repeat. pos.	360 pls	360 pls	360 pls
	Infinite running repeat. pos.	0: Disable	0: Disable	0: Disable
	Command in-position range	0 pls	0 pls	0 pls
	Tracking error over-range value	0 pls	0 pls	0 pls
	Tracking error level	0: Warning	0: Warning	0: Warning
	Current pos. compenstion amount	0 pls	0 pls	0 pls
	Current speed filter time constant	0 ms	0 ms	0 ms
	Error reset monitoring time	100 ms	100 ms	100 ms
	S/W limit during speed control	0: Do not detect	0: Do not detect	0: Do not detect
	Override mode	0: Specified by ratio	0: Specified by ratio	0: Specified by ratio
	JOG high speed	100000 pls/s	100000 pls/s	100000 pls/s
	JOG low speed	10000 pls/s	10000 pls/s	10000 pls/s
	JOG acceleration	100000 pls/s2	100000 pls/s2	100000 pls/s2
	JOG deceleration	100000 pls/s2	100000 pls/s2	100000 pls/s2
	JOG jerk	0 pls/s3	0 pls/s3	0 pls/s3
NC Settings	Spindle command speed ack. range	95 %	95 %	95 %
	Spindle zero speed ack. rpm	5 rpm	5 rpm	5 rpm

(29) If you turned off the power of slave(servo drive) and turned it on again in the step (28), execute "Online – Slave - Connect " again to connect module and slave(servo drive).

(30) After selecting the command axis and turning on the servo of the relevant axis, check if the relevant axis is in servo on state and check the motor operation by operating the motor using jog or others.

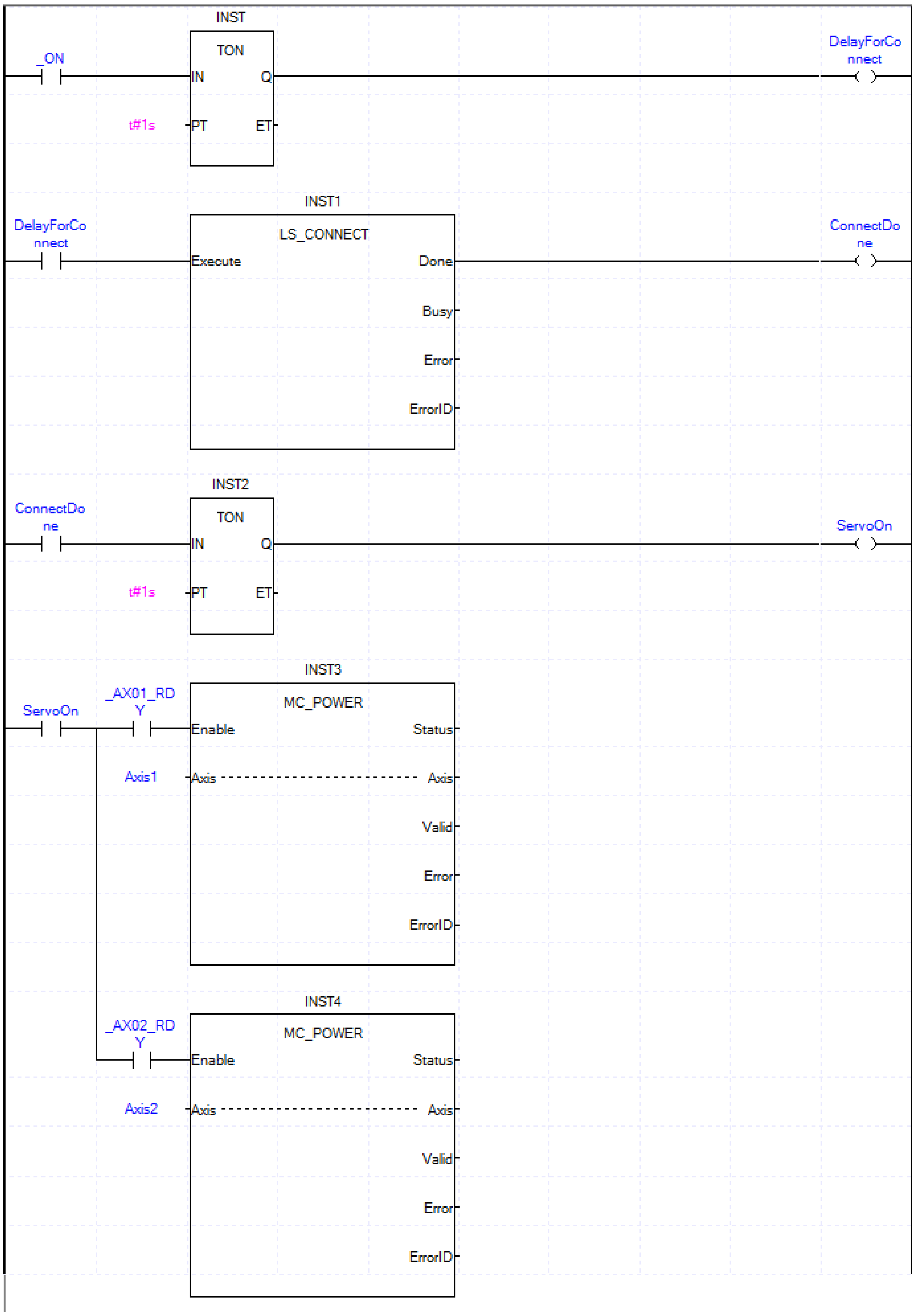


(31) If vibration or noise is generated when motor is operating, adjust the responsibility, inertia ratio, and gain values of servo parameter and transmit them to servo drive. Use the dedicated setting tool of servo drive for detailed setting such as auto tuning.

(32) Create motion program.

The exercise below is for the case that 2 servos are set to 1 & 2 axes using XGK CPU, and LS_CONNECT is used for connection and the connected axis is servo on by using MC_Power. The rest of the exercise can be added as user's need.

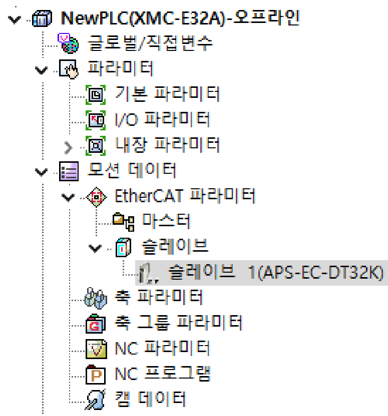
Motion task can be divided into main task, periodic task, and initialization task. You can add program to the relevant task of the project tree depending on the character of the program.



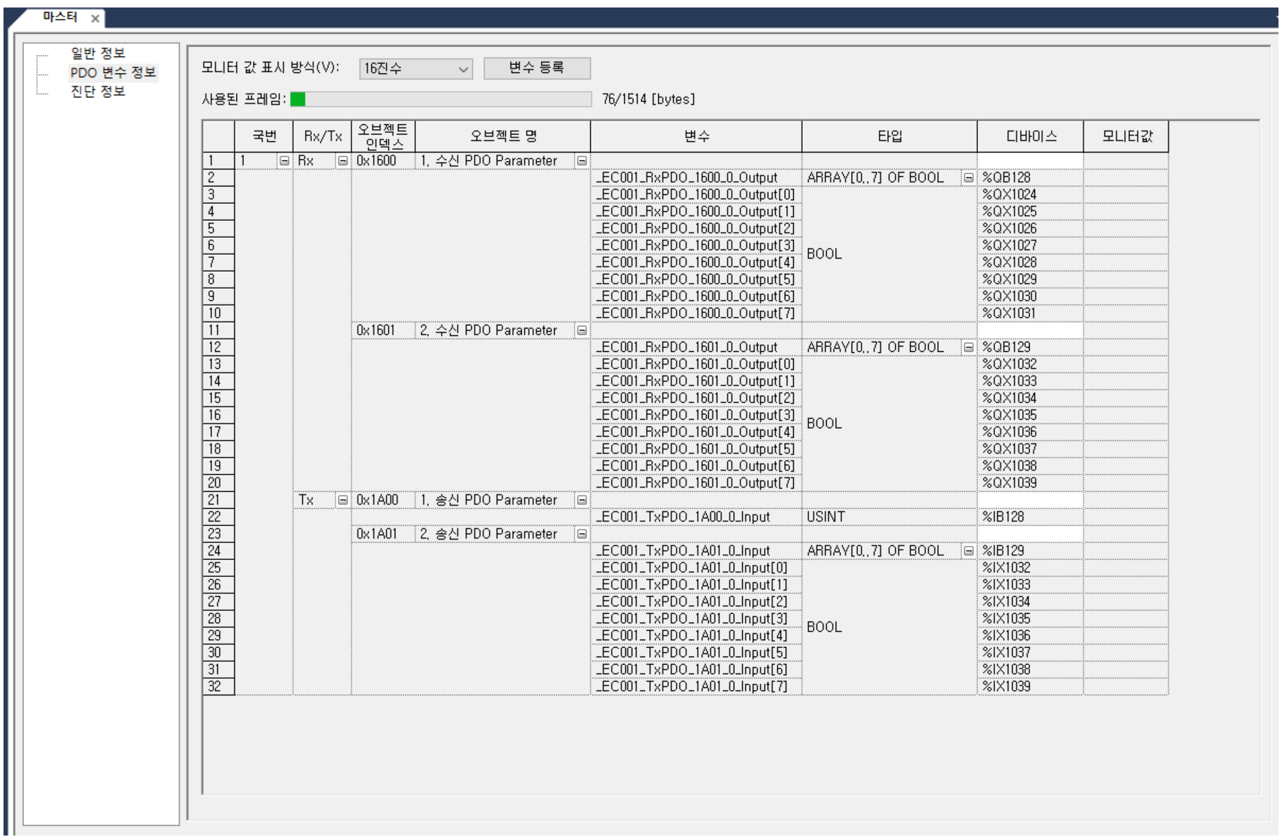
Notes

1. In the case of EtherCAT I/O products, input and output variables can be checked in “Master”-“PDO variable information”.

   1) After adding EtherCAT I/O (APS-EC-DT32K) to slave parameters, execute “Online (O)-Slave-Connection”.



   2) Access input/output variables from “Master”-“PDO variable information”. The output of EtherCAT I/O is assigned to the output variable (Q) and the input to the input variable (I) and can be checked in the “Device” column. The value of each variable can be checked or changed in “Monitor Value”.

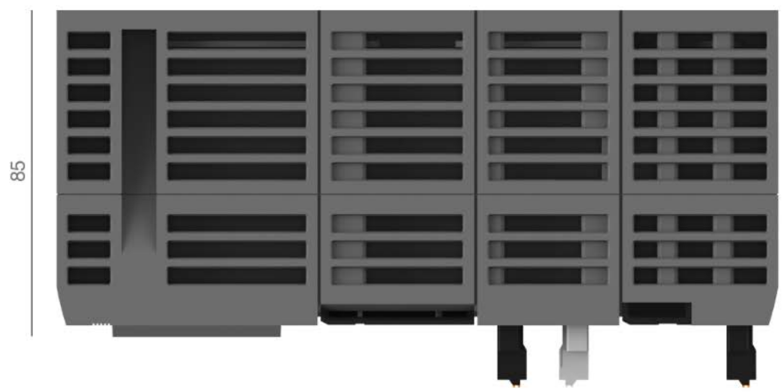


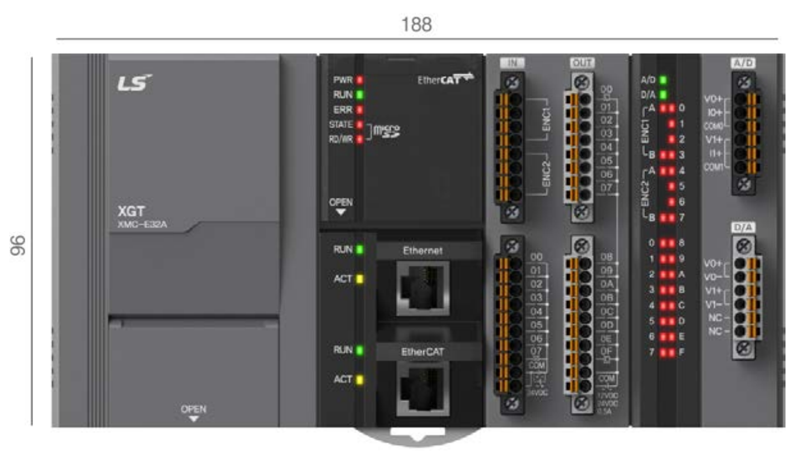
	국번	Rx/Tx	오브젝트 인덱스	오브젝트 명	변수	타입	디바이스	모니터값
1	1	Rx	0x1600	1. 수신 PDO Parameter				
2					_EC001_RxPDO_1600_0_Output	ARRAY[0..7] OF BOOL	%QB128	
3					_EC001_RxPDO_1600_0_Output[0]		%QX1024	
4					_EC001_RxPDO_1600_0_Output[1]		%QX1025	
5					_EC001_RxPDO_1600_0_Output[2]		%QX1026	
6					_EC001_RxPDO_1600_0_Output[3]	BOOL	%QX1027	
7					_EC001_RxPDO_1600_0_Output[4]		%QX1028	
8					_EC001_RxPDO_1600_0_Output[5]		%QX1029	
9					_EC001_RxPDO_1600_0_Output[6]		%QX1030	
10					_EC001_RxPDO_1600_0_Output[7]		%QX1031	
11			0x1601	2. 수신 PDO Parameter				
12					_EC001_RxPDO_1601_0_Output	ARRAY[0..7] OF BOOL	%QB129	
13					_EC001_RxPDO_1601_0_Output[0]		%QX1032	
14					_EC001_RxPDO_1601_0_Output[1]		%QX1033	
15					_EC001_RxPDO_1601_0_Output[2]		%QX1034	
16					_EC001_RxPDO_1601_0_Output[3]	BOOL	%QX1035	
17					_EC001_RxPDO_1601_0_Output[4]		%QX1036	
18					_EC001_RxPDO_1601_0_Output[5]		%QX1037	
19					_EC001_RxPDO_1601_0_Output[6]		%QX1038	
20					_EC001_RxPDO_1601_0_Output[7]		%QX1039	
21		Tx	0x1A00	1. 송신 PDO Parameter				
22					_EC001_TxPDO_1A00_0_Input	USINT	%IB128	
23			0x1A01	2. 송신 PDO Parameter				
24					_EC001_TxPDO_1A01_0_Input	ARRAY[0..7] OF BOOL	%IB129	
25					_EC001_TxPDO_1A01_0_Input[0]		%IX1032	
26					_EC001_TxPDO_1A01_0_Input[1]		%IX1033	
27					_EC001_TxPDO_1A01_0_Input[2]		%IX1034	
28					_EC001_TxPDO_1A01_0_Input[3]	BOOL	%IX1035	
29					_EC001_TxPDO_1A01_0_Input[4]		%IX1036	
30					_EC001_TxPDO_1A01_0_Input[5]		%IX1037	
31					_EC001_TxPDO_1A01_0_Input[6]		%IX1038	
32					_EC001_TxPDO_1A01_0_Input[7]		%IX1039	

Appendix 4 Dimension

XMC-E32A/XMC-E32C

-This figure shows the XMC-E32A. XMC-E32C has same Dimension.





※ The external size of the XMC-E16A, XMC-E08A, XMC-E32C is same as that of the XMC-E32A.

Appendix 5 ESC(EtherCAT Slave Controller) Register

The following table is the information ESC(EtherCAT Slave Controller) Register. For information on the all area, refer to the EtherCAT Registers(SectionⅡ) datasheet on the BECKHOFF website below.
http://www.beckhoff.com/english.asp?download/ethercat_development_products.htm

1. ESC DL Status (0x0110:0x0111)

Bit	Description	ECAT	PDI	Reset Value
0	PDI operational/EEPROM loaded correctly: 0: EEPROM not loaded, PDI not operational (no access to Process Data RAM) 1: EEPROM loaded correctly, PDI operational (access to Process Data RAM)	r*/-	r/-	0
1	PDI Watchdog Status: 0: Watchdog expired 1: Watchdog reloaded	r*/-	r/-	0
2	Enhanced Link detection: 0: Deactivated for all ports 1: Activated for at least one port NOTE: EEPROM value is only taken over at first EEPROM load after power-on or reset	r*/-	r/-	ET1100/ET1200: 1 until first EEPROM load, then EEPROM ADR 0x0000.9 IP Core with feature: 1 until first EEPROM load, then EEPROM ADR 0x0000.9 or 0x0000[15:12] Others: 0
3	Reserved	r*/-	r/-	0
4	Physical link on Port 0: 0: No link 1: Link detected	r*/-	r/-	0
5	Physical link on Port 1: 0: No link 1: Link detected	r*/-	r/-	0
6	Physical link on Port 2: 0: No link 1: Link detected	r*/-	r/-	0
7	Physical link on Port 3: 0: No link 1: Link detected	r*/-	r/-	0
8	Loop Port 0: 0: Open 1: Closed	r*/-	r/-	0
9	Communication on Port 0: 0: No stable communication 1: Communication established	r*/-	r/-	0
10	Loop Port 1: 0: Open 1: Closed	r*/-	r/-	0

Bit	Description	ECAT	PDI	Reset Value
11	Communication on Port 1: 0: No stable communication 1: Communication established	r*/-	r/-	0
12	Loop Port 2: 0: Open 1: Closed	r*/-	r/-	0
13	Communication on Port 2: 0: No stable communication 1: Communication established	r*/-	r/-	0
14	Loop Port 3: 0: Open 1: Closed	r*/-	r/-	0
15	Communication on Port 3: 0: No stable communication 1: Communication established	r*/-	r/-	0

Table 1-1: Register ESC DL Status (0x0110:0x0111)

Register 0x0111	Port 3	Port 2	Port 1	Port 0
0x55	No link, closed	No link, closed	No link, closed	No link, closed
0x56	No link, closed	No link, closed	No link, closed	Link, open
0x59	No link, closed	No link, closed	Link, open	No link, closed
0x5A	No link, closed	No link, closed	Link, open	Link, open
0x65	No link, closed	Link, open	No link, closed	No link, closed
0x66	No link, closed	Link, open	No link, closed	Link, open
0x69	No link, closed	Link, open	Link, open	No link, closed
0x6A	No link, closed	Link, open	Link, open	Link, open
0x95	Link, open	No link, closed	No link, closed	No link, closed
0x96	Link, open	No link, closed	No link, closed	Link, open
0x99	Link, open	No link, closed	Link, open	No link, closed
0x9A	Link, open	No link, closed	Link, open	Link, open
0xA5	Link, open	Link, open	No link, closed	No link, closed
0xA6	Link, open	Link, open	No link, closed	Link, open
0xA9	Link, open	Link, open	Link, open	No link, closed
0xAA	Link, open	Link, open	Link, open	Link, open
0xD5	Link, closed	No link, closed	No link, closed	No link, closed
0xD6	Link, closed	No link, closed	No link, closed	Link, open
0xD9	Link, closed	No link, closed	Link, open	No link, closed
0xDA	Link, closed	No link, closed	Link, open	Link, open

Table 1-2: Decoding port state in ESC DL Status register 0x0111 (typical modes only)

2. RX Error Counter (0x0300:0x0307)

Errors are only counted if the corresponding port is enabled.

Bit	Description	ECAT	PDI	Reset Value
7:0	Invalid frame counter of Port y (counting is stopped when 0xFF is reached).	r/- w(clr)	r/-	0
15:8	RX Error counter of Port y (counting is stopped when 0xFF is reached). This is coupled directly to RX ERR of MII interface/EBUS interface.	r/- w(clr)	r/-	0

Table 2: Register RX Error Counter Port y (0x0300+y*2:0x0301+y*2)

3. Forwarded RX Error Counter (0x0308:0x030B)

Bit	Description	ECAT	PDI	Reset Value
7:0	Forwarded error counter of Port y (counting is stopped when 0xFF is reached).	r/- w(clr)	r/-	0

Table 3: Register Forwarded RX Error Counter Port y (0x0308+y)

NOTE: Error Counters 0x0300-0x030B are cleared if one of the RX Error counters 0x0300-0x030B is written. Write value is ignored (write 0).

4. ECAT Processing Unit Error Counter (0x030C)

Bit	Description	ECAT	PDI	Reset Value
7:0	ECAT Processing Unit error counter (counting is stopped when 0xFF is reached). Counts errors of frames passing the Processing Unit (e.g., FCS is wrong or datagram structure is wrong).	r/- w(clr)	r/-	0

Table 4: Register ECAT Processing Unit Error Counter (0x030C)

NOTE: Error Counter 0x030C is cleared if error counter 0x030C is written. Write value is ignored (write 0).

5. Lost Link Counter (0x0310:0x0313)

Bit	Description	ECAT	PDI	Reset Value
7:0	Lost Link counter of Port y (counting is stopped when 0xff is reached). Counts only if port loop is Auto.	r/ w(clr)	r/-	0

Table 5: Register Lost Link Counter Port y (0x0310+y)

NOTE: Only lost links at open ports are counted. Lost Link Counters 0x0310-0x0313 are cleared if one of the Lost Link Counters 0x0310-0x0313 is written. Write value is ignored (write 0).

6. AL Status (0x0130:0x0131)

Bit	Description	ECAT	PDI	Reset Value
3:0	Actual State of the Device State Machine: 1: Init State 3: Request Bootstrap State 2: Pre-Operational State 4: Safe-Operational State 8: Operational State	r*/-	r/(w)	1
4	Error Ind: 0: Device is in State as requested or Flag cleared by command 1: Device has not entered requested State or changed State as result of a local action	r*/-	r/(w)	0
5	Device Identification: 0: Device Identification not valid 1: Device Identification loaded	r*/-	r/(w)	0
15:6	Reserved, write 0	r*/-	r/(w)	0

Table 6: Register AL Status (0x0130:0x0131)

NOTE: AL Status register is only writable from PDI if Device Emulation is off (0x0140.8=0), otherwise AL Status register will reflect AL Control register values.
* Reading AL Status from ECAT clears ECAT Event Request 0x0210[3].

7. AL Status Code (0x0134:0x0135)

Bit	Description	ECAT	PDI	Reset Value
15:0	AL Status Code	r/-	r/w	0

Table 7: Register AL Status Code (0x0134:0x0135)

8. ECAT Event Request (0x0210:0x0211)

Bit	Description	ECAT	PDI	Reset Value
0	DC Latch event: 0: No change on DC Latch Inputs 1: At least one change on DC Latch Inputs (Bit is cleared by reading DC Latch event times from ECAT for ECAT controlled Latch Units, so that Latch 0/1 Status 0x09AE:0x09AF indicates no event)	r/-	r/-	0
1	Reserved	r/-	r/-	0
2	DL Status event: 0: No change in DL Status 1: DL Status change (Bit is cleared by reading out DL Status 0x0110:0x0111 from ECAT)	r/-	r/-	0
3	AL Status event: 0: No change in AL Status 1: AL Status change (Bit is cleared by reading out AL Status 0x0130:0x0131 from ECAT)	r/-	r/-	0
4 5	Mirrors values of each SyncManager Status: 0: No Sync Channel 0 event 1: Sync Channel 0 event pending 0: No Sync Channel 1 event 1: Sync Channel 1 event pending	r/-	r/-	0
11	0: No Sync Channel 7 event 1: Sync Channel 7 event pending	r/-	r/-	0
15:12	Reserved	r/-	r/-	0

Table 8: Register ECAT Event Request (0x0210:0x0211)

Appendix 6 Using EtherCAT slaves form other companies

Describes how to use the EtherCAT slaves from other companies that is not existed ESI file in XG5000, to XMC-E32A.

(1) EtherCAT slave information file (ESI)

The information of the EtherCAT slave is defined by the ESI (EtherCAT Slave Information) file, which is supplied by the manufacturer of slave product. Based on the ESI file information, the XG5000 configures communication settings with the EtherCAT slave and downloads it to the XMC-E32A. ESI file is required for connection and operation of XMC-E32A and slaves. For normal operation of slaves, the ESI file must be a version supports the slaves.

For the latest version of ESI file, please contact the manufacturer or distributor of the slaves.

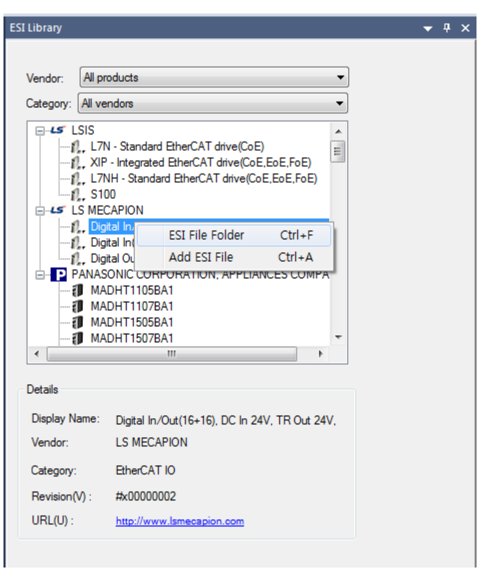
(2) Adding ESI file of EtherCAT slave from other companies

The XG5000 provides functions for adding ESI file of EtherCAT slave from other companies, and searching the installation folder of ESI file.

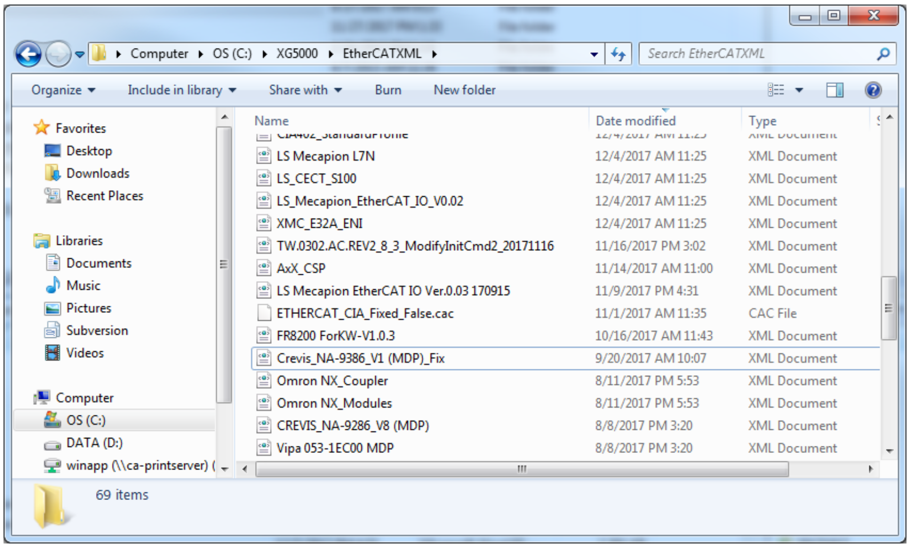
(a) Searching function of EtherCAT slave information (ESI) file (provided by XG5000 4.22 or later version)

After adding the ESI file to the EtherCATXML directory folder (XG5000\EtherCATXML) and restarting the XG5000, it will be reflected in the ESI library window.
For user's convenience, XG5000 provides the function to open the folder. Perform the ESI file folder search function in the following order.

1) Execute XG5000
2) Click 'View' – 'ESI library window' menu to activate the ESI library window.
3) Right-click on a mouse on ESI library window, and click on 'ESI file folder' menu.



4) Window Explorer of 'EtherCATXML' folder is activated as shown below.



(b) Adding function of EtherCAT slave information (ESI) file (provided by XG5000 4.22 or later version)

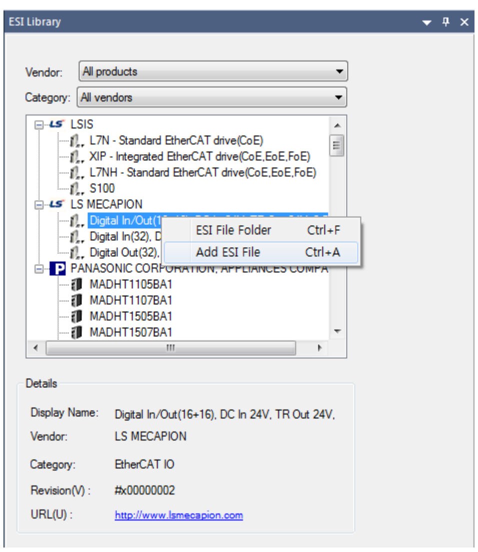
EtherCAT slave information file In addition to bulk addition of ESI files by folder search, it provides individual ESI file addition function. ESI files can be added to the ESI library window without having to restart the XG5000.

Perform the ESI file adding function in the following order.

1) Execute XG5000.

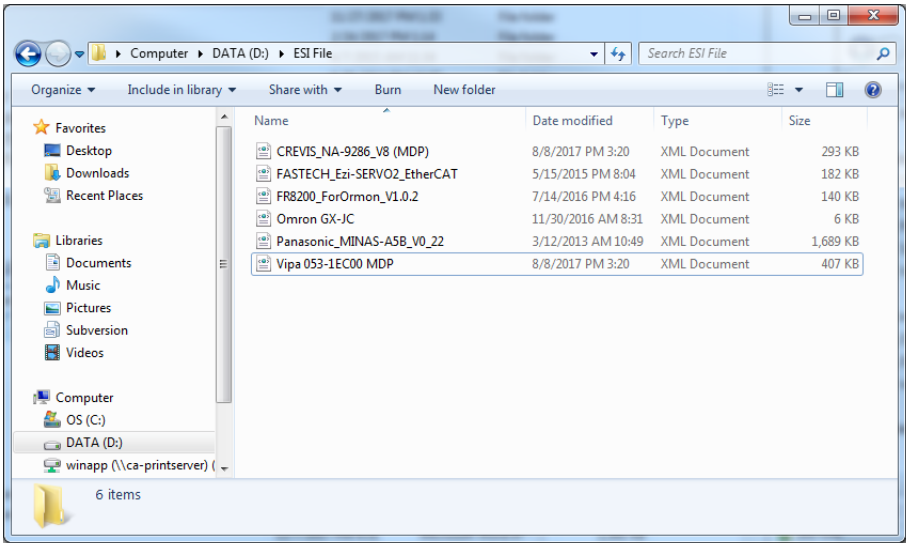
2) Click 'View' – 'ESI library window' menu to activate the ESI library window.

3) Right-click on a mouse on ESI library window, and click on 'ESI file adding' menu.

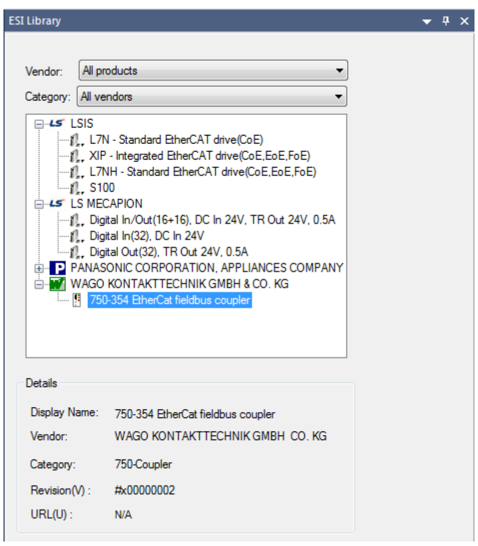


4) Window Explorer of 'EtherCATXML' folder is activated as shown below.

5) Navigate to the folder and select ESI file. (In the example below, there is ESI files in 'E:\ ESIFiles' folder)



6) Slave information the selected file is added to the ESI library window.



(3) Setting of slave supporting MDP (provided by XG5000 4.22 and XMC-E32A OS 1.1 or later version)

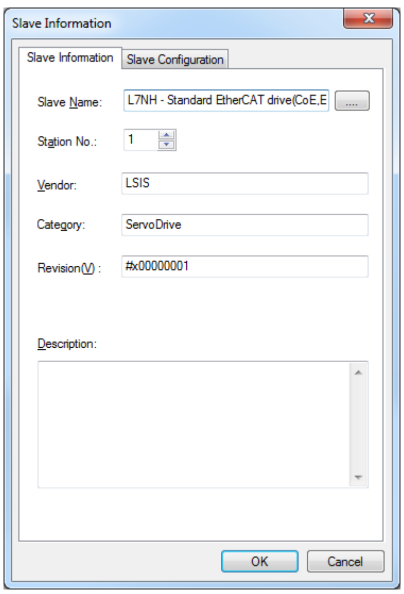
The MDP (Modular Device Profile) is the EtherCAT standard (ETG50001) that defines the configuration data structure of EtherCAT slave. Slave supporting MDP can be set in 'EtherCAT parameter – Slave configuration edit window' of XG5000.
In order to set slave supporting MDP, slave information and module information mounted on slave must be configured

(a) Addition of slave supporting MDP

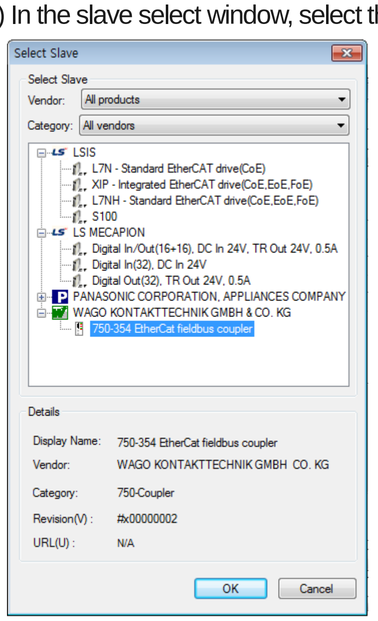
How to add slave supporting MDP and configure module.

1) In 'EtherCAT parameter – Slave' menu, Right-click on, and add slave via 'Add item - Add slave'.

2) In the slave information window, click on the .... button.

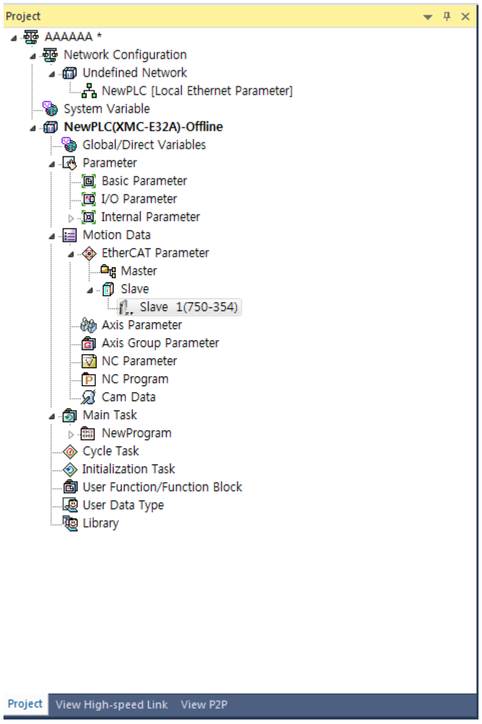


3) In the slave select window, select the slave supporting MDP and click on the OK button.



4) In the slave information window, click on the OK button

5) The Slave is added as the sub item on Project tree (Motion data – EtherCAT parameter – Slave)



(b) Editing Slave supporting MDP

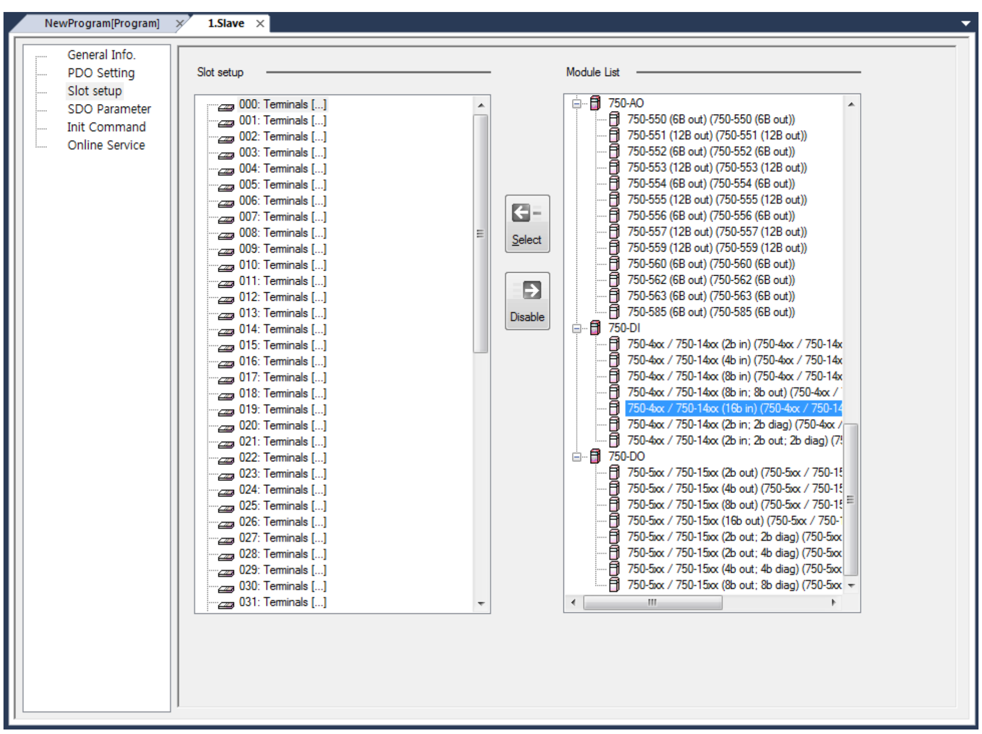
1) Double-click on 'Slave – Slave1(750-354)' on project tree

2) Click on 'Slot setup'.

3) In the module list, select the module you want to set.

4) Select the slot you want to assign the module

5) Click on the Select button.



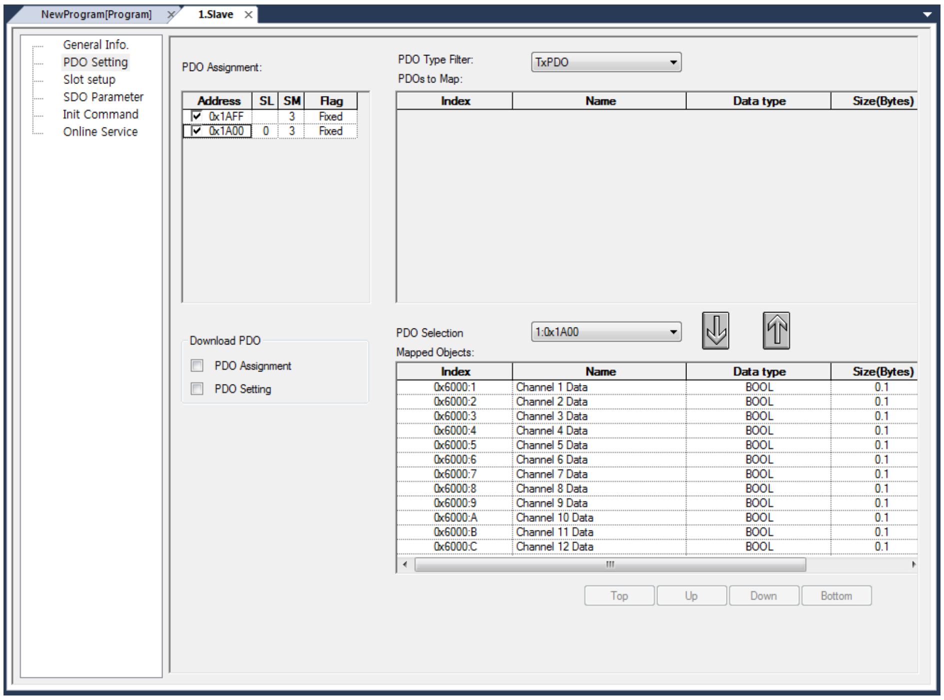
(c) Check of PDO allocation information

1) Click on ‘PDO Setting’

2) Check the PDO assignment window.

3) Check the slot number of current module in ‘SL number window’

4) Check the object mapped to current slot address in ‘PDO window’

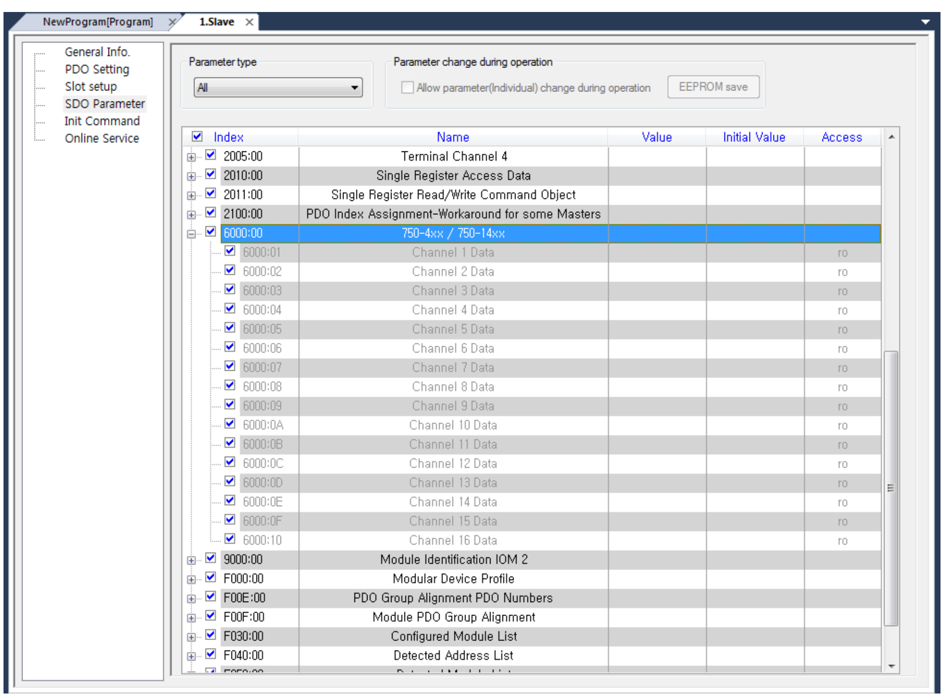


(d) Check of SDO parameter

1) Click on ‘SDO parameter’.

2) Check the added object in SDO parameter.

(However, depending on the slave type, PDO may not be added in the SDO parameter.)



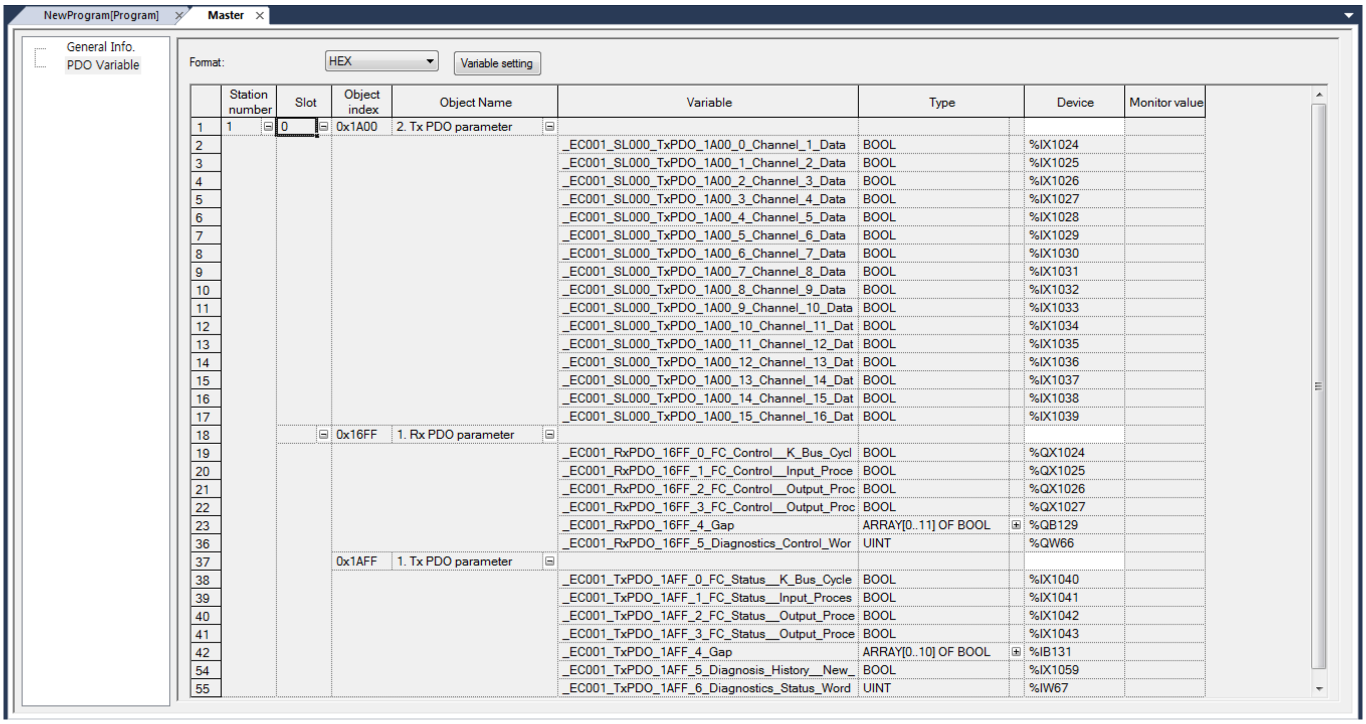
(e) Check of PDO variable information

1) Double-click on ‘EtherCAT parameter – Master’ on project tree.

2) Click on ‘PDO variable’.

3) Check the added object with slot information.

(The added PDO variable can be used as a variable specified and device in the program, after registering it in the global variable through ‘Register variable’.)



	Station number	Slot	Object index	Object Name	Variable	Type	Device	Monitor value
1	1	0	0x1A00	2. Tx PDO parameter				
2					_EC001_SL000_TxPDO_1A00_0_Channel_1_Data	BOOL	%IX1024	
3					_EC001_SL000_TxPDO_1A00_1_Channel_2_Data	BOOL	%IX1025	
4					_EC001_SL000_TxPDO_1A00_2_Channel_3_Data	BOOL	%IX1026	
5					_EC001_SL000_TxPDO_1A00_3_Channel_4_Data	BOOL	%IX1027	
6					_EC001_SL000_TxPDO_1A00_4_Channel_5_Data	BOOL	%IX1028	
7					_EC001_SL000_TxPDO_1A00_5_Channel_6_Data	BOOL	%IX1029	
8					_EC001_SL000_TxPDO_1A00_6_Channel_7_Data	BOOL	%IX1030	
9					_EC001_SL000_TxPDO_1A00_7_Channel_8_Data	BOOL	%IX1031	
10					_EC001_SL000_TxPDO_1A00_8_Channel_9_Data	BOOL	%IX1032	
11					_EC001_SL000_TxPDO_1A00_9_Channel_10_Data	BOOL	%IX1033	
12					_EC001_SL000_TxPDO_1A00_10_Channel_11_Dat	BOOL	%IX1034	
13					_EC001_SL000_TxPDO_1A00_11_Channel_12_Dat	BOOL	%IX1035	
14					_EC001_SL000_TxPDO_1A00_12_Channel_13_Dat	BOOL	%IX1036	
15					_EC001_SL000_TxPDO_1A00_13_Channel_14_Dat	BOOL	%IX1037	
16					_EC001_SL000_TxPDO_1A00_14_Channel_15_Dat	BOOL	%IX1038	
17					_EC001_SL000_TxPDO_1A00_15_Channel_16_Dat	BOOL	%IX1039	
18			0x16FF	1. Rx PDO parameter				
19					_EC001_RxPDO_16FF_0_FC_Control__K_Bus_Cycl	BOOL	%QX1024	
20					_EC001_RxPDO_16FF_1_FC_Control__Input_Proce	BOOL	%QX1025	
21					_EC001_RxPDO_16FF_2_FC_Control__Output_Proc	BOOL	%QX1026	
22					_EC001_RxPDO_16FF_3_FC_Control__Output_Proc	BOOL	%QX1027	
23					_EC001_RxPDO_16FF_4_Gap	ARRAY[0..11] OF BOOL	%QB129	
36					_EC001_RxPDO_16FF_5_Diagnostics_Control_Wor	UINT	%QW66	
37			0x1AFF	1. Tx PDO parameter				
38					_EC001_TxPDO_1AFF_0_FC_Status__K_Bus_Cycle	BOOL	%IX1040	
39					_EC001_TxPDO_1AFF_1_FC_Status__Input_Proces	BOOL	%IX1041	
40					_EC001_TxPDO_1AFF_2_FC_Status__Output_Proce	BOOL	%IX1042	
41					_EC001_TxPDO_1AFF_3_FC_Status__Output_Proce	BOOL	%IX1043	
42					_EC001_TxPDO_1AFF_4_Gap	ARRAY[0..10] OF BOOL	%IB131	
54					_EC001_TxPDO_1AFF_5_Diagnosis_History__New_	BOOL	%IX1059	
55					_EC001_TxPDO_1AFF_6_Diagnostics_Status_Word	UINT	%IW67	

Note

In addition to manual configuration through user editing, Automatic configuration is provided for connection of slave supporting MDP and generation of EtherCAT parameter.

Add ESI file of slave supporting MDP on ESI library window, connect XMC-E32A and slave as network, and perform the function 'Online - EtherCAT Slave - Auto Connect'.

If the automatic configuration does not work normally, please upgrade the product OS. (XMC-E32A OS 1.1 or later version)

Appendix 7 Terms of Cnet Communications

1) Communication Methods

(1) Simplex communication

It refers to a communication method in which the flow of information is always transmitted in one direction. The information cannot be sent in the reverse direction.

(2) Half-duplex communication

It is a method that can transmit information in both directions at intervals, although it cannot send the flow of information in both directions at the same time with the use of a 1-wire cable.

(3) Full-duplex communication

It is a method in which the flow of information can be transmitted and received simultaneously with the use of a 2-wire cable.

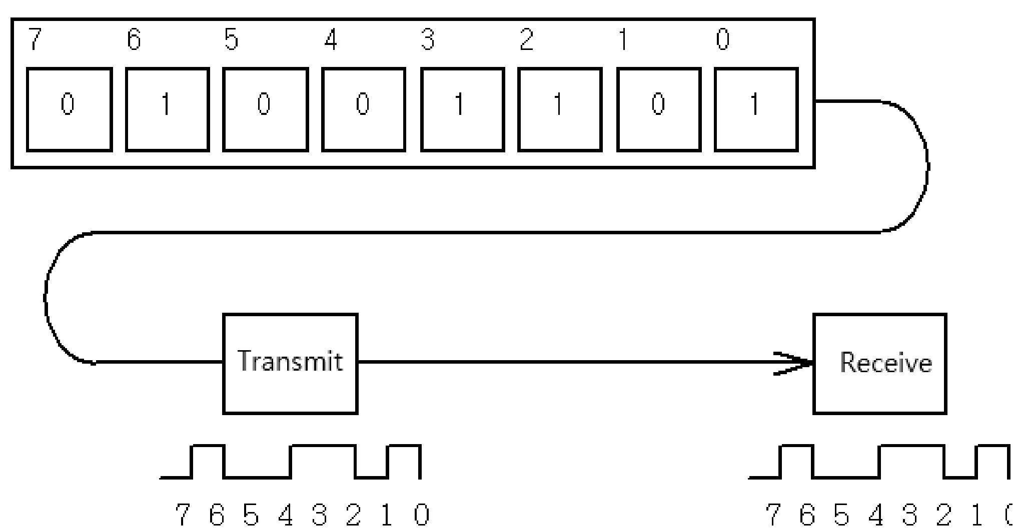
2) Transmission Methods

In consideration of the speed, stability and economical efficiency in binary (0 and 1) data transmission, transmission methods are divided into the following two methods.

(1) Serial communication

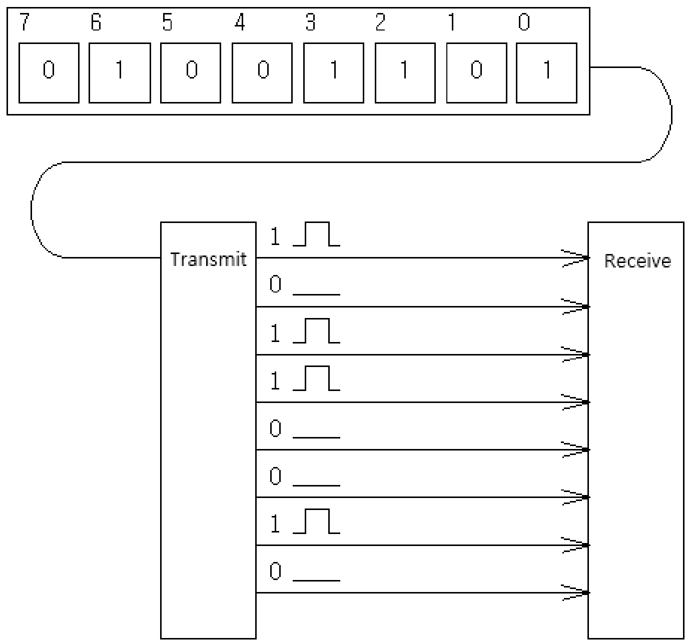
It is the process of transmitting data one bit at a time through one cable. It results in a lower transmission speed, but has advantages of low installation cost and simple software.

RS-232C, RS-422 and RS-485 are applicable to serial communication.



(2) Parallel transmission

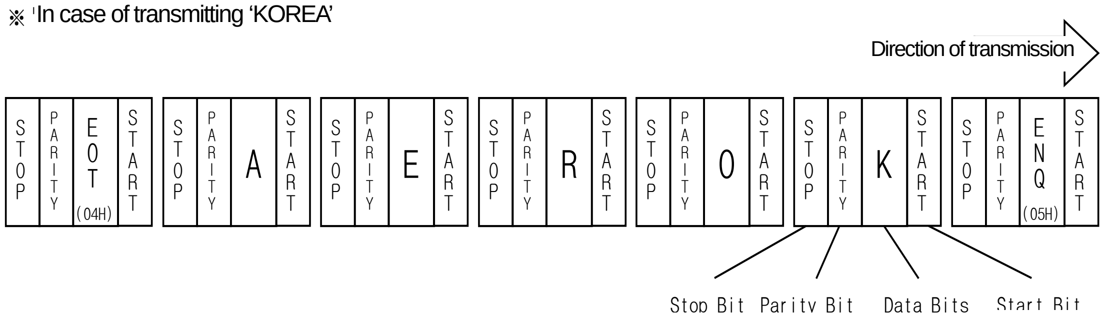
This method is used in printers, and the like. It sends one byte of data at a time, which ensures high speed and superior data accuracy, but has a disadvantage in that the installation costs increase exponentially as the transmission distance increases.



3) Asynchronous Communication

This method transmits only 1 character at a time in serial transmission, when the synchronous signal (clock, etc.) is not transmitted. It sends a character code with a start bit at the beginning of 1 character and end with a stop bit at the end.

※ 'In case of transmitting 'KOREA'



4) Protocol

It refers to a communication rule defined between the transmission and reception sides of information in advance to send and receive efficient and reliable information between more than two computers and terminals without errors. In general, it defines call establishment, connection, structure of message exchange format, retransmission of error message, circuit inversion procedures, and character synchronization between terminals.

5) BPS (Bits Per Second) and CPS (Characters Per Second)

BPS refers to the unit of transmission rate that indicates how many bits are sent per second in data transmission. CPS refers to the number of characters transmitted per second.
Normally, 1 byte (8 bits) holds 1 character, so CPS is the number of bytes that can be transmitted per second.

6) Node

Node refers to the connection point of data in a network tree structure. In general, a network consists of many nodes. It is also referred to as station number.

7) Packet

It is a term used in packet switching, which is a method of grouping data that is transmitted over a network into packets. The packet is a combination of the words “package” and “bucket” and attaches a header that indicates a destination address (station number, etc.) by separating the transmission data into a predetermined length.

8) Port

It refers to a part of the data processing device that receives or sends data from a remote terminal in data communication, and it means RS-232C or RS-422 port in Cnet serial communication.

9) RS-232C

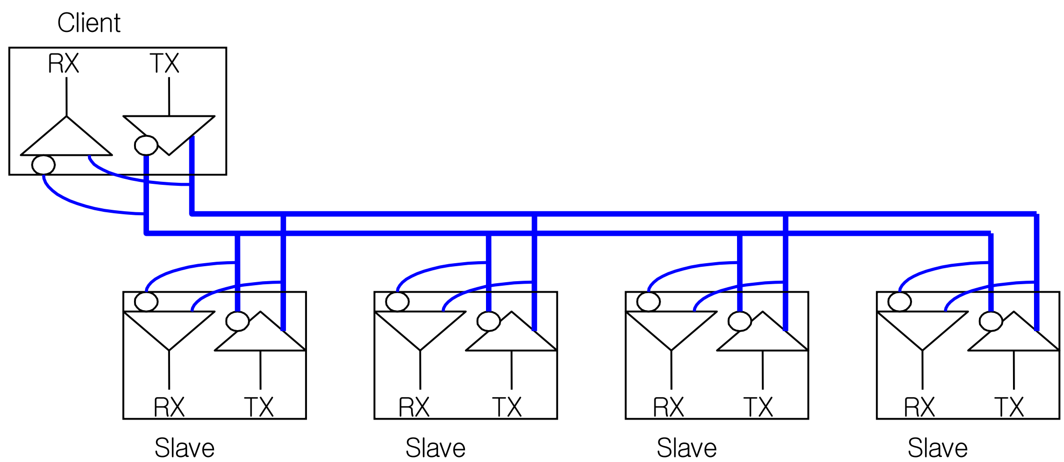
It refers to a serial communication standard established by EIA according to the recommendations of CCITT as an interface for connecting a modem to a terminal or a computer. It is used not only for modem connection but also for direct connection to null modem. The disadvantages are that the transmission distance is short, and only 1:1 communication is possible, and a standard that overcomes these disadvantages is RS-422, RS-485.

10) RS-422/RS-485

It is one of the serial transmission standards and enables 1:N connection and ensures longer transmission distance than RS-232C. The difference between the two standards is that RS-422 uses four signal lines: TX(+), TX(-), RX(+) and RX(-), whereas RS-485 has two signal lines (+) (-) and performs transmission and reception trough the same signal line. Accordingly, RS-422 implements full-duplex communication and RS-485 implements half-duplex communication.

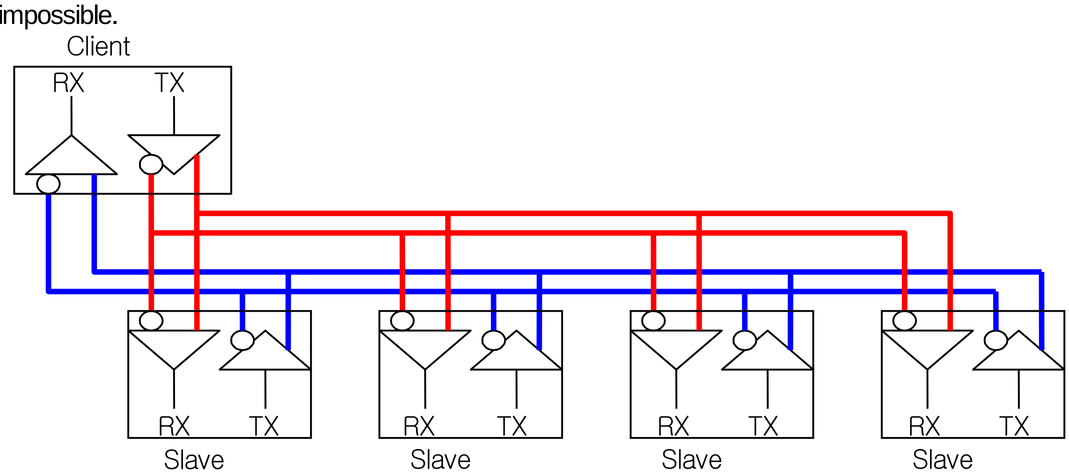
11) Half-Duplex Communication

It refers to a communication method that allows communication in both directions, but not at the same time, and RS-485 communication method is applicable to this method. Since transmission and reception are made through a single communication line, it is mainly used in multi-drop communication systems where several stations communicate over a single signal line. Because it uses one signal line, data may be lost due to data collision when several stations transmit data simultaneously, thereby ensuring that only one station transmits at a time. The flowing figure shows an example of the configuration by half-duplex communication. Since each communication station is connected to each other so that it can transmit and receive through one line, it is possible to communicate between all stations and thus to run multiple servers



12) Full-Duplex Communication

It refers to a communication method that can transmit and receive data in both directions at the same time, and RS-232C and RS-422 communication methods are applicable to this method. Since the transmission and reception lines are separated, it can transmit and receive data simultaneously without data collision. The figure shows an example of the configuration of RS-422 full-duplex communication. Since the transmitting end of the server station and the receiving end of the client stations are connected to one line, and the transmitting end of the client stations is connected to the receiving end of the server station, the multi-server function is limited because communication between the client stations is impossible.



13) BCC (Block Check Character)

Since serial communication has the potential to transmit a distorted signal due to the effects of unwanted noise in the transmission line, BCC is required to enable a receiver side to determine whether the signal is a normal signal or a distorted signal. The receiver side can calculate BCC for itself by using the data that came in the front-end of the BCC, compare it with the received BCC, and determine whether the signal is normal or abnormal.

Appendix 8 EtherCAT Diagnostics

This appendix explains the diagnosis functions provided by the EtherCAT Master and XG5000.

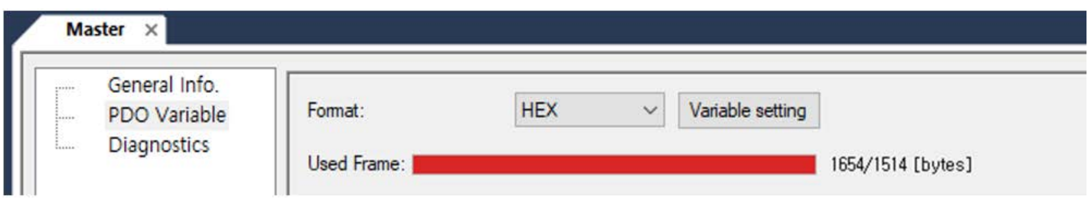
(1) Offline diagnostic function

XG5000 can examine effectiveness of a slave that participates in EtherCAT communication through ESI (EtherCAT Slave Information) file inspection used by users during an EtherCAT connection.

(a) EtherCAT frame size examination

XG5000 can confirm frame sizes used by the EtherCAT that the current user added through the following procedure.

1) Select [Motion Data]-[EtherCAT Parameter]-[Master] items on the XG5000 project window.
2) Activate the master window.
3) Activate the PDO variable information item window on the master window.
4) Confirm the used frame on the screen.
   The used frame is displayed as the frame usage that is currently set versus the maximum tolerated frame size.
5) If the frame size in current use is greater than the maximum tolerated frame size by adding slaves or PDO items, the progress bar of the used frame is displayed in red.
6) When the relevant error occurs, the frame is prepared within the maximum tolerated frame size by deleting slaves or PDO items.



(b) Effectiveness assessment for each slave operation mode

XG5000 performs the essential PDO (Process Data Object) examination for each operation mode of drive slaves that is automatically allocated by connecting with an axis when adding slaves.

Operation modes supported by XMC products are as follows:

CiA 402 Operation Mode	Support
Profile position mode	-
Velocity mode	O
Profile velocity mode	-
Torque profile mode	-
Homing mode	O
Interpolated position mode	-
Cyclic sync position mode	O
Cyclic sync velocity mode	O
Cyclic sync torque mode	O

Essential PDO configuration for each operation mode is as follows:

Velocity(vl) mode PDO Item

Index	Name	Category
6040h	Control word	Necessary
6041h	Status word	Necessary
6042h	vl target velocity	Necessary
6044h	vl velocity actual value	Necessary

Homing(hm) mode PDO Item

Index	Name	Category
6040h	Control word	Necessary
6041h	Status word	Necessary
6098h	Homing method	Necessary

Cyclic sync velocity(csv) mode PDO Item

Index	Name	Category
6040h	Control word	Necessary
6041h	Status word	Necessary
6064h	Position actual value	Necessary
60FFh	Target velocity	Necessary

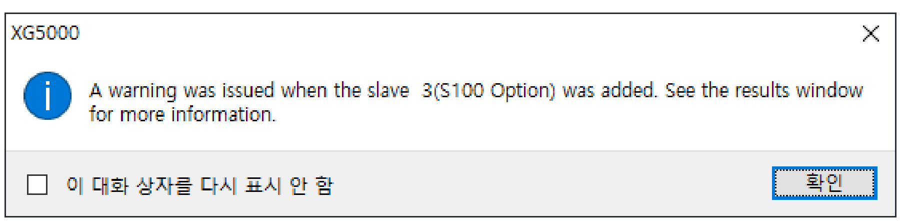
Cyclic sync position(csp) mode PDO Item

Index	Name	Category
6040h	Control word	Necessary
6041h	Status word	Necessary
607Ah	Target position	Necessary
6064h	Position actual value	Necessary

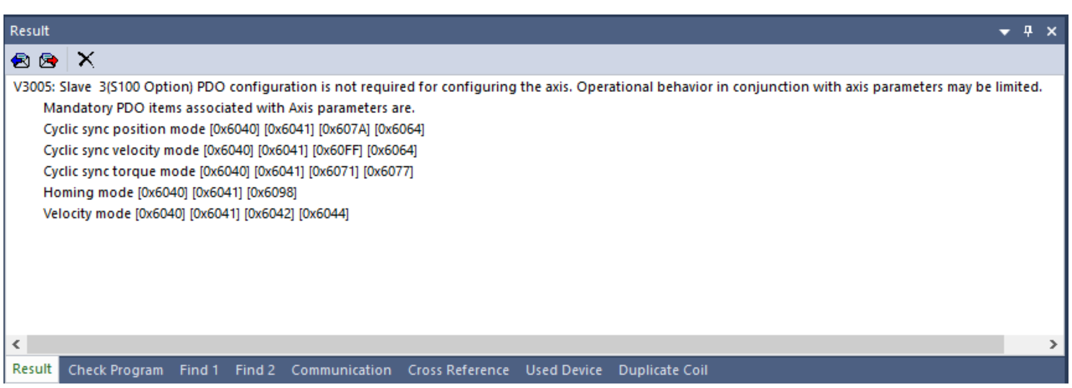
Cyclic sync torque(cst) mode PDO Item

Index	Name	Category
6040h	Control word	Necessary
6041h	Status word	Necessary
6071h	Target torque	Necessary
6077h	Torque actual value	Necessary

If there is no PDO item corresponding to the operation mode set in the PDO setting based on the above-mentioned contents, the following warning message is output on the screen when adding slaves.



The following warning message is output in the result window.



Note

The following is the contents of PDO configuration for each operation mode suggested by the Implementation Directive for CiA402 Drive Profile
(ETG.6010 D(R) V1.1.0). For PDO configuration for each operation mode, see the contents.

Table 80: Recommended PDO Mapping according to supported modes of operation

Index	Name	csp only	csv only	cst only	csp + csv	csp + cst	csv + cst	csp + csv + cst	pp + hm
RxPDOs									
6040_h	Controlword	+	+	+	+	+	+	+	+
6060_h	Modes of operation	-	-	-	+	+	+	+	o
$607A_h$	Target Position	+	-	-	+	+	-	+	+
$60FF_h$	Target velocity	-	+	-	+	-	+	+	-
6071_h	Target Torque	-	-	+	-	+	+	+	-
TxPDOs									
6041_h	Statusword	+	+	+	+	+	+	+	+
6061_h	Modes of operation display	-	-	-	+	+	+	+	o
6064_h	Position actual value	+	+	+	+	+	+	+	+
6077_h	Torque actual value	-	-	+	-	+	+	+	-

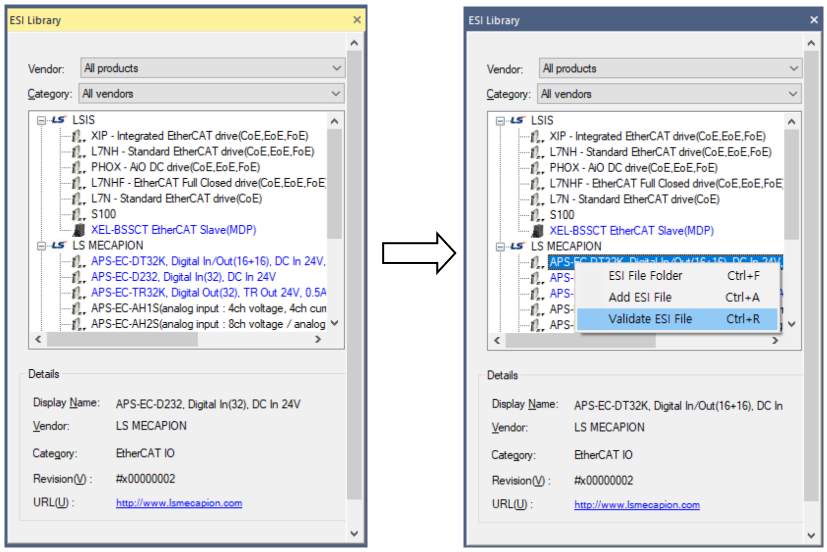
+ = should = recommended
0 = can = possible
- = not used

(c) Effectiveness assessment of ESI files

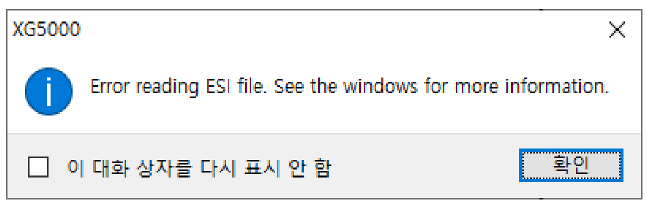
XG5000 has a function to assess ESI files through XSD (XML Schema Definition) files, or structure-defined files of the ESI file. An assessing item is the relationship between data in a file and consistency of the ESI file structure.

If an error is expected to occur in the ESI file of the assessed device, blue text is displayed on each slave of the ESI library window.

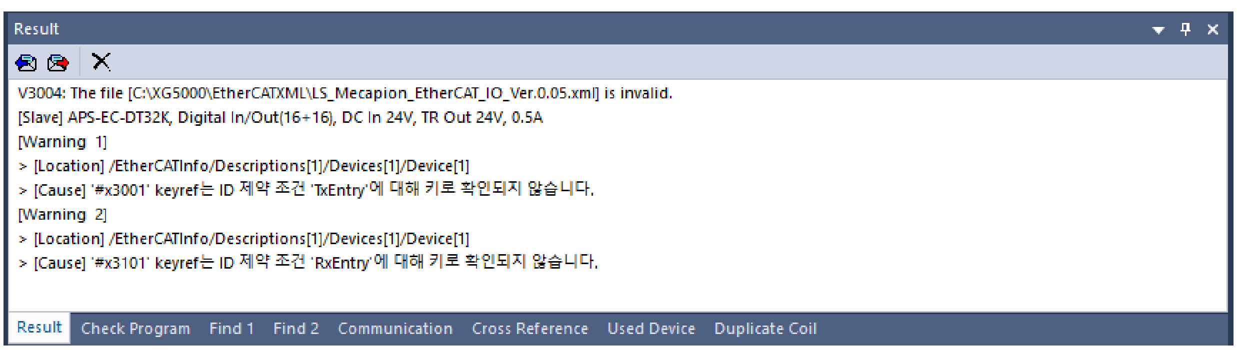
For assessment of the relevant ESI, select the slave on which the blue text is displayed, right click on the menu and execute the [ESI File Inspection] menu.



When executing it, the following warning dialog box is displayed.



In the result window, you can confirm errors of the ESI file that the device currently assessed belongs to.



(2) Online diagnostic function

XMC provides diagnostic functions of the EtherCAT network that is currently connected through flags and diagnostic functions. Diagnostic flags provide a function to read the ESC Register through a flag without using the existing ESC Register read command (LS_READESC).

The operation of diagnostic flags is conducted after the EtherCAT connection is completed through a connection behavior after manually writing the EtherCAT parameter or conducting EtherCAT slave>>Automatic connection through XG5000.

(a) Diagnostic Flag

Flag Name	Flag Description
_SLVxx_ALStatus	Shows the AL status of slave applications.
_SLVxx_ALStatusCode	Shows the error code of slave applications.
_SLVxx_DLStatus	Shows the link status information of slaves.
_SLVxx_LinkLostCounterA/B/C/D	Shows the link stop event counter for each port of slaves.
_SLVxx_InvalidFrameCounterA/B/C/D	The count increases if there are errors in frame formats such as Preamble, SFD and CRC. The whole bit sequence corresponds to the damaged frame. Errors can occur in frames.
_SLVxx_RxErrorCounterA/B/C/D	The count increases if individual symbols are not valid. Errors can occur both in and out of frames.
_SLVxx_ForwardedRXErrCounterA/B/C/D	Abnormal frames detected through the previous slaves show the received count.

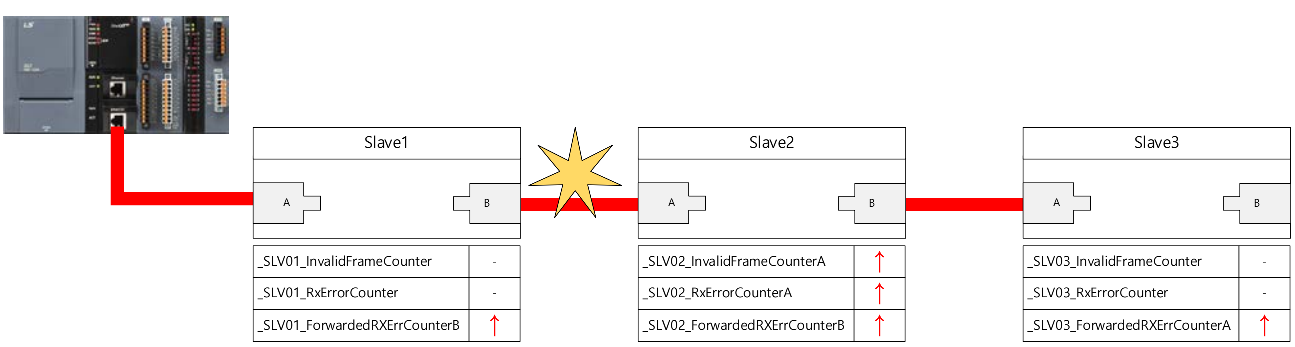
Note

For more information about a diagnostic flag value, see Appendix 5 ECS Register.

(b) Slave Diagnosis using a Diagnostic Flag

Problems can occur in EtherCAT communication due to various causes such as device failure during EtherCAT communication and the occurrence of external EMC failure. In such a case, a slave location where a problem occurs can be estimated through a diagnostic flag.

The following figure shows the case that communication failure occurs because external noise flows into a cable between slave 1 and slave 2.



As communication failure occurs due to noise that flows into a cable between slave 1 and slave 2, the value of

SLV02_InvalidFrameCounterA or SLV02_RxErrorCounterA of slave 2 increases. As the frame that an error is detected in slave 2 is conveyed to slave 3, the value of SLV03_ForwardedRxErrCounterA of slave 3 increases. As frames in communication loopbacked at slave 3 are conveyed to port B of slave 1 and slave 2, each value of SLV01_ForwardedRxErrCounterB and SLV01_ForwardedRxErrCounterB increases.

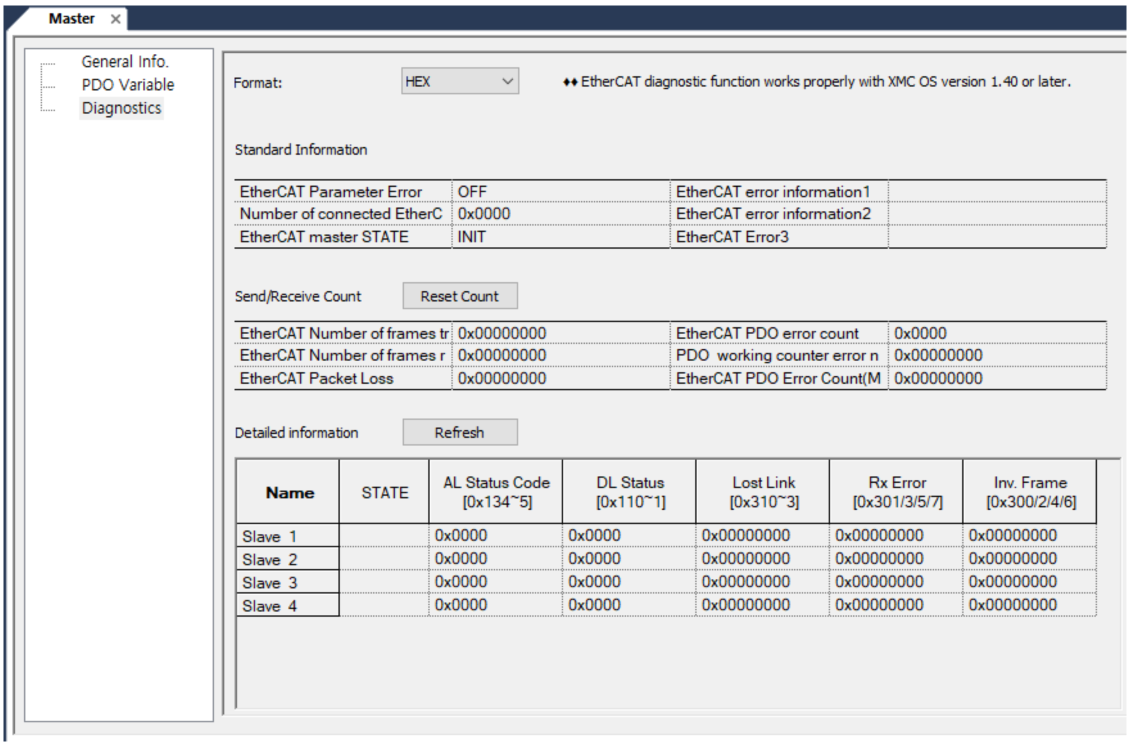
Note

If a problem occurs during EtherCAT communication, diagnosis is performed by using a diagnostic flag as follows:

Sequence	Diagnostic activities
1	Find a slave where the value of InvalidFrameCounter or RxErrorCounter has increased. - The location where many problems have occurred is a slave where the value of InvalidFrameCounter or RxErrorCounter is not 0.
2	Confirm the detected slave and a cable of the slave connected to the front end. - Confirm if the EtherCAT cable is located close to a power cable or noise sources - Confirm if our manufactured cable is bad. - Confirm if the cable shields are proper.
3	Confirm the detected slave and the slave device connected to the front end. - Confirm if the power is proper. - Confirm the ground of slaves.
4	Confirm if a problem is related with a specific device by moving the position or exchanging a device of the detected slave.

(c) Diagnostic information of the master

XMC provides a monitoring function that can confirm flags to confirm the EtherCAT communication status on the screen through [Master]-[Diagnostic information].



[Description of the diagnostic information window]

a. Methods to display monitors: Decide the types of data displayed. Decimal or hexadecimal can be chosen.

b. Basic information: Monitor common information related to EtherCAT. Display basic information related to the EtherCAT connection.

   Confirm the number of EtherCAT slave connections and the error information flag when an EtherCAT error occur.

c. Count information for sending and receiving: Users can confirm the number of sending and receiving frames while EtherCAT communication operates and the value to count PDO errors. Users can reset the current count through the [Reset count] function.

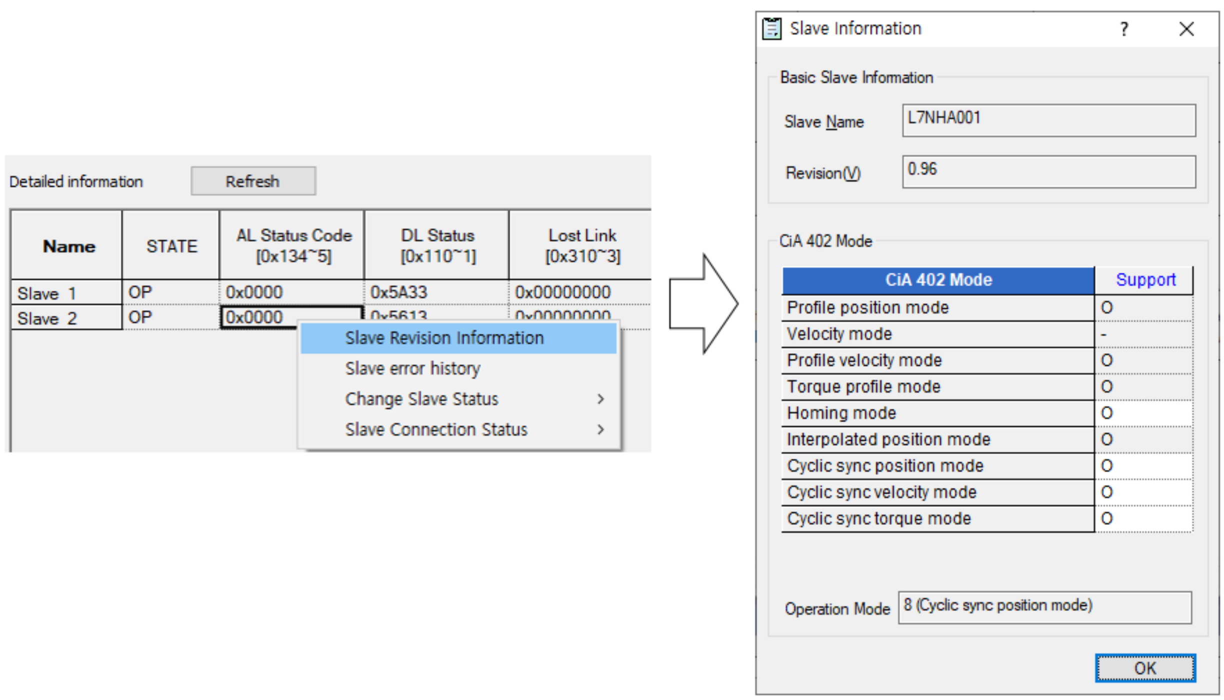
d. Detailed information: Diagnostic flags of slaves are updated during monitoring. Users can read the current flag values when they want to read during monitoring through the [Update the current value] function.

(d) Confirm the CiA 402 Drive Profile support mode

Users can confirm the CiA 402 Drive Profile mode supported by a slave on the screen of diagnostic information. After selecting a slave on the screen of diagnostic information, right click and select [Basic information of a slave].

Users can confirm information of the CiA 402 support mode supported by the relevant slave along with Slave name/Revision information on the slave information window. Users should previously confirm if the PDO configuration is appropriate for an operation mode to use the supported operation mode.

[Whether to support] displays support modes supported by the current slave and
a cell marked with a white color shows the CiA 402 support mode provided by the EtherCAT master of the current XMC product family. The [Current operation mode] displays the current value of an operation mode where a slave is operating.

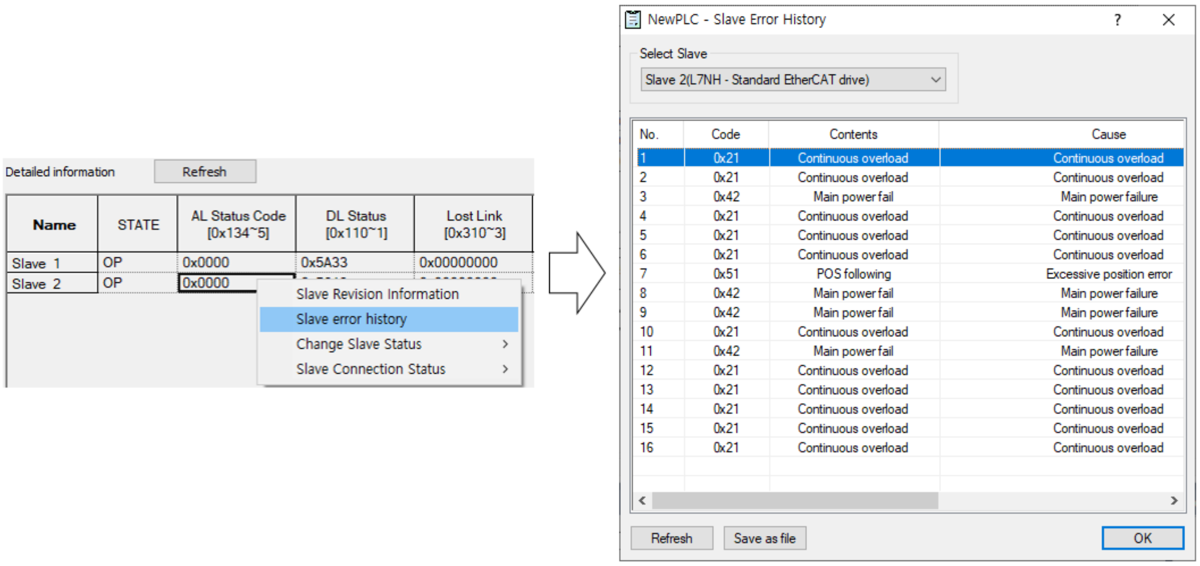


Note

The EtherCAT master of the XMC product family supports VL (Velocity mode), HM (Homing Mode), CSP (Cyclic sync position mode), CSV (Cyclic sync velocity mode) and CST (Cyclic sync torque mode) among the CiA 402 drive profile modes.
The EtherCAT slave that supports modes other than them can be used by allocating it to the operating axes.

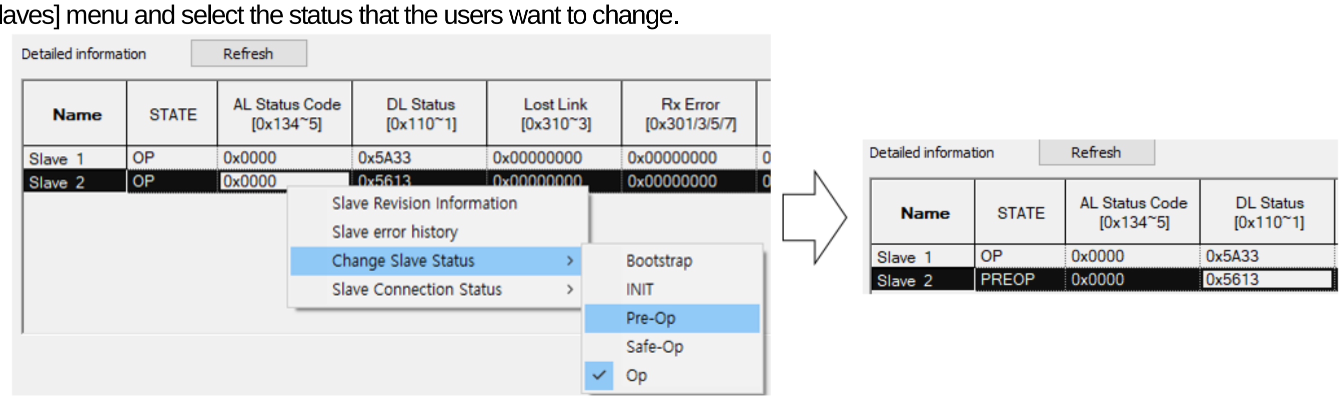
(e) History of slave errors

The function to view the history of EtherCAT slave errors on the screen of diagnostic information is provided. On the screen of diagnostic information place mouse cursor to the detailed information of slaves, and right click and select the [History of slave errors] menu. The selected history of slaves is displayed on the screen.



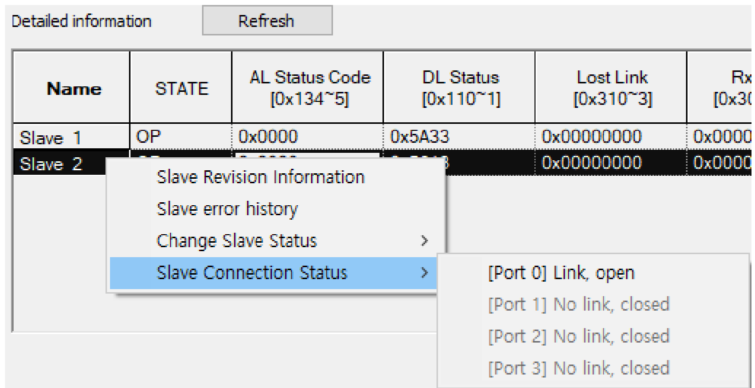
(f) Change the status of slaves

A function to change the status of EtherCAT slaves on the screen of diagnostic information is supported. On the screen of diagnostic information, place the mouse cursor on the detailed information of slaves, right click and select the [Change the status of slaves] menu and select the status that the users want to change.



(g) Status of slave connection

A function to confirm the status of an EtherCAT slave connection on the screen of diagnostic information is supported. On the screen of diagnostic information, place the mouse cursor on the detailed information of slaves, right click and select the [Status of slave connection] menu and then, users can confirm the current status of slave connections.



Note

Online diagnostic flags and diagnostic functions are supported on XMC-E32A(C) OS 1.4 and XG5000 4.28 or later.

Warranty

1. Warranty Period
The product you purchased will be guaranteed for 36 months from the date of manufacturing.

2. Scope of Warranty
(1) The initial diagnosis of faults is basically conducted by your company. However, upon your request, our company or our service network can undertake this task for a fee. If the cause of the fault lies with our company, this service will be provided free of charge.
(2) This warranty only applies if the product is used under normal conditions according to the specifications and precautions described in the handling instructions, user manuals, catalogs, and caution labels.
(3) Even within the free warranty period, the following cases will be subject to paid repairs:
1) Replacement of consumable and life-limited parts (e.g., relays, fuses, electrolytic capacitors, fans, LCDs, batteries, etc.)
2) Failures or damages caused by improper storage, handling, negligence, or accidents by the customer
3) Failures resulting from the customer's hardware or software design
4) Failures due to modifications without our consent
(Repairs will be refused, even for a fee, if recognized as modified or repaired outside our company)
5) Failures that could have been avoided if the customer's equipment, in which our product is incorporated, had safety devices required by legal regulations or common industry standards
6) Failures that could have been prevented if maintenance and replacement of consumable parts were performed normally according to the handling instructions or user manuals
7) Failures and damages to the product caused by using connected equipment or inappropriate consumables
8) Failures caused by external factors such as fire, abnormal voltage, force majeure, and natural disasters such as earthquakes, lightning,salt damage, wind, and flood damage
9) Failures due to reasons that could not be predicted with the scientific and technical standards at the time of our shipment
10) Other failures, damages, or defects recognized as the responsibility of your company

Environmental Policy

LS ELECTRIC Co., Ltd supports and observes the environmental policy as below.

Environmental Management

LS ELECTRIC considers the environmental preservation as the preferential management subject and every staff of LS ELECTRIC use the reasonable endeavors for the pleasurably environmental preservation of the earth.

About Disposal

LS ELECTRIC' PLC unit is designed to protect the environment. For the disposal, separate aluminum, iron and synthetic resin (cover) from the product as they are reusable.

www.ls-electric.com

LS ELECTRIC Co., Ltd.

■ **Headquarter**
LS-ro 127(Hogye-dong) Dongan-gu, Anyang-si, Gyeonggi-Do, 14119, Korea
■ **Seoul Office**
LS Yongsan Tower, 92, Hangang-daero, Yongsan-gu, Seoul, 04386, Korea
Tel: 82-2-2034-4033, 4888, 4703 Fax: 82-2-2034-4588
E-mail: automation@ls-electric.com

■ **Overseas Subsidiaries**
- **LS ELECTRIC Japan Co., Ltd. (Tokyo, Japan)**
  Tel: 81-3-6268-8241 E-Mail: japan@ls-electric.com
- **LS ELECTRIC (Dalian) Co., Ltd. (Dalian, China)**
  Tel: 86-411-8730-6495 E-Mail: china.dalian@lselectric.com.cn
- **LS ELECTRIC (Wuxi) Co., Ltd. (Wuxi, China)**
  Tel: 86-510-6851-6666 E-Mail: china.wuxi@lselectric.com.cn
- **LS ELECTRIC Middle East FZE (Dubai, U.A.E.)**
  Tel: 971-4-886-5360 E-Mail: middleeast@ls-electric.com
- **LS ELECTRIC Europe B.V. (Hoofddorp, Netherlands)**
  Tel: 31-20-654-1424 E-Mail: europartner@ls-electric.com
- **LS ELECTRIC America Inc. (Chicago, USA)**
  Tel: 1-800-891-2941 E-Mail: sales.us@lselectricamerica.com
- **LS ELECTRIC Turkey Co., Ltd.**
  Tel: 90-212-806-1225 E-Mail: turkey@ls-electric.com

■ **Overseas Branches**
- **LS ELECTRIC Tokyo Office (Japan)**
  Tel: 81-3-6268-8241 E-Mail: tokyo@ls-electric.com
- **LS ELECTRIC Beijing Office (China)**
  Tel: 86-10-5095-1631 E-Mail: china.auto@lselectric.com.cn
- **LS ELECTRIC Shanghai Office (China)**
  Tel: 86-21-5237-9977 E-Mail: china.auto@lselectric.com.cn
- **LS ELECTRIC Guangzhou Office (China)**
  Tel: 86-20-3818-2883 E-Mail: china.auto@lselectric.com.cn
- **LS ELECTRIC Chengdu Office (China)**
  Tel: 86-28-8670-3201 E-Mail: china.auto@lselectric.com.cn
- **LS ELECTRIC Qingdao Office (China)**
  Tel: 86-532-8501-2065 E-Mail: china.auto@lselectric.com.cn
- **LS ELECTRIC Nanjing Office (China)**
  Tel: 86-25-8467-0005 E-Mail: china.auto@lselectric.com.cn
- **LS ELECTRIC Bangkok Office (Thailand)**
  Tel: 66-90-950-9683 E-Mail: thailand@ls-electric.com
- **LS ELECTRIC Jakarta Office (Indonesia)**
  Tel: 62-21-2933-7614 E-Mail: indonesia@ls-electric.com
- **LS ELECTRIC Moscow Office (Russia)**
  Tel: 7-499-682-6130 E-Mail: info@lselectric-ru.com
- **LS ELECTRIC America Western Office (Irvine, USA)**
  Tel: 1-949-333-3140 E-Mail: america@ls-electric.com

2024.06