Right choice for ultimate yield

LSIS strives to maximize customers' profit in gratitude of choosing us for your partner.

Programmable Logic Controller

Standalone Motion Controller

XGT Series

User's Manual

XMC-E32A

Safety Instructions

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

LSIS

www.lsis.com

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

⚠ Warning

- **Please install a protection circuit on the exterior of PLC so that the whole system may operate safely regardless of failures from external power or PLC.** Any abnormal output or operation from PLC may cause serious problems to safety in whole system.
  - Install protection units on the exterior of PLC like an interlock circuit that deals with opposite operations such as emergency stop, protection circuit, and forward/reverse rotation or install an interlock circuit that deals with high/low limit under its position controls.
  - If any system error (watch-dog timer error, module installation error, etc.) is detected during CPU operation in PLC, all output signals are designed to be turned off and stopped for safety. However, there are cases when output signals remain active due to device failures in Relay and TR which can't be detected. Thus, you are recommended to install an addition circuit to monitor the output status for those critical outputs which may cause significant problems.
- **Never overload more than rated current of output module nor allow to have a short circuit**. Over current for a long period time may cause a fire .
- **Never let the external power of the output circuit to be on earlier than PLC power**, which may cause accidents from abnormal output operation.
- **Please install interlock circuits in the sequence program for safe operations in the system when exchange data with PLC or modify operation modes using a computer or other external equipments** Read specific instructions thoroughly when conducting control

Safety Instructions for Design Process

⚠ Caution

- **I/O signal or communication line shall be wired at least 100mm away from a high-voltage cable or power line.** Fail to follow this.

Safety Instructions on Installation Process

⚠ Caution

- **Use PLC only in the environment specified in PLC manual or general standard of data sheet.** 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.
- **Be sure that screws get tighten securely under vibrating environments.** Fail to do so will put the product under direct vibrations which will cause electric shock, fire and abnormal operation.
- **Do not come in contact with conducting parts in each module**, which may cause electric

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.
- **After wiring process is done, make sure that terminal covers are installed properly before its use.** Fail to install the cover may cause electric shocks.

⚠ 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.
- **Make sure that pressed terminals get tighten following the specified torque. External connector type shall be pressed or soldered using proper equipments.**

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.
- **Don't let the battery recharged, disassembled, heated, short or soldered**. Heat, explosion or ignition may cause injuries or fire.

⚠ 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.
- **When making a modification on programs or using run to modify functions under PLC operations, read and comprehend all contents in the manual fully.** Mismanagement will cause damages to products and accidents.
- **Avoid any physical impact to the battery and prevent it from dropping as well.** Damages to battery may cause leakage from its fluid. When battery was dropped or exposed under strong impact, never reuse the battery again. Moreover skilled workers are needed when exchanging batteries.

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	'17.4	First Edition	-
V 1.1	'17.11	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	'18.6	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

※ The number of User's manual is indicated right part of the back cover.

Thank you for purchasing PLC of LSIS 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.lsis.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/CPUE/CPUH/CPUS/CPUU)	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 Motion Control Module.

Current XMC-E32A manual is written based on the following version.

Related OS version list

XMC-E32A	XG5000
V1.00	V4.20
V1.10	V4.23
V1.20	V4.25

◎ 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.

(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 and 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) Speed control
3) Synchronous control
4) Multi-axis simultaneous start

(d) Various multi-axis group operations are available.
1) Circular arc 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.
It is available to create up to 32 kinds of cam data with various cam profile of XG5000.

(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.

(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

Note

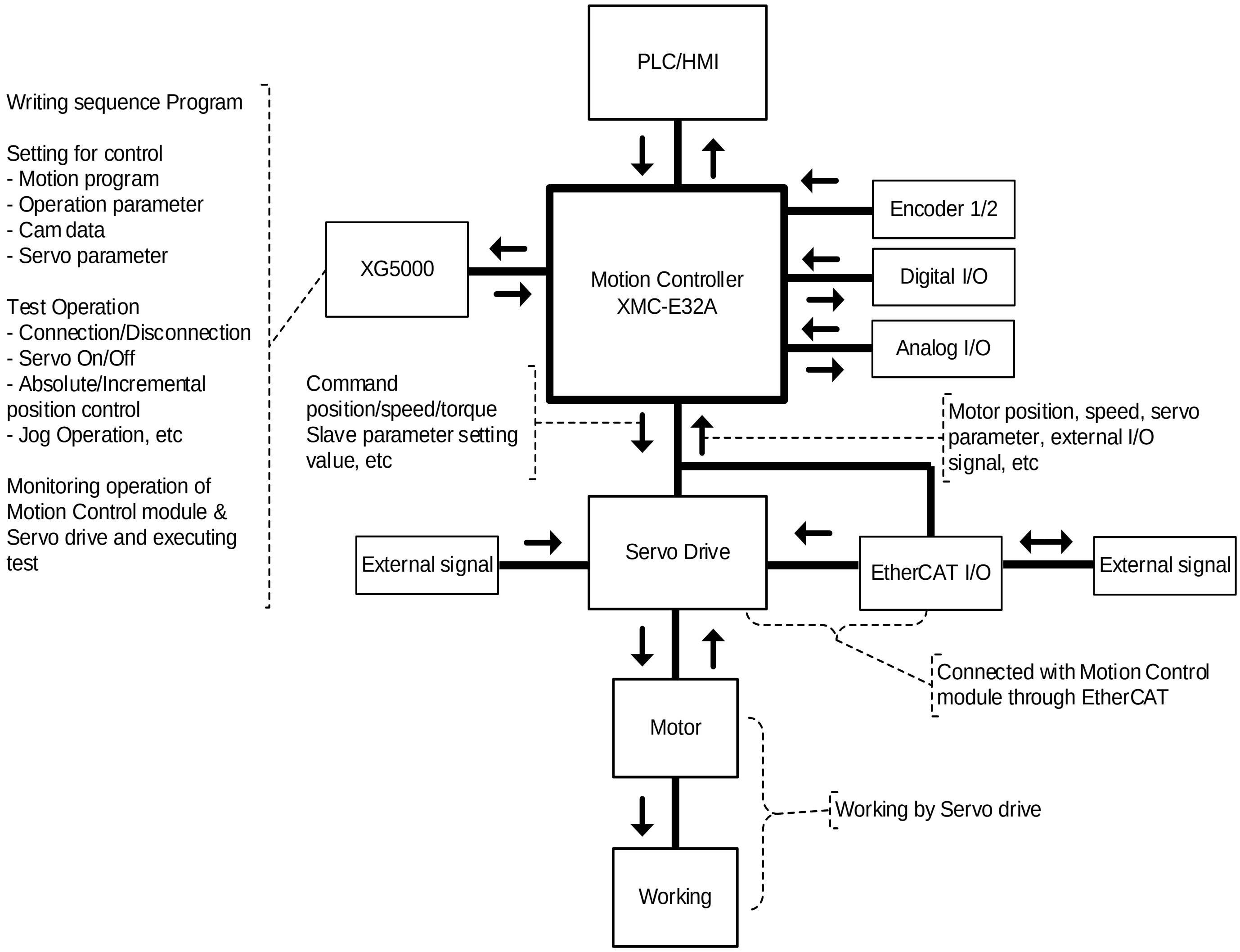
What is EtherCAT?

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.

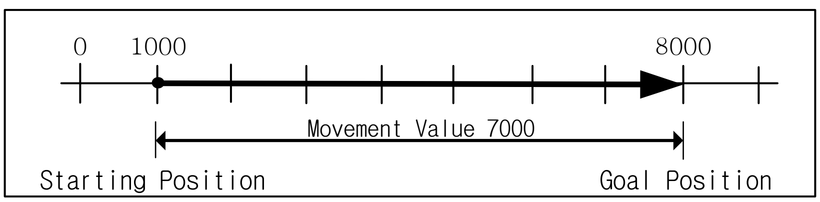
1.3.1 Position Control

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

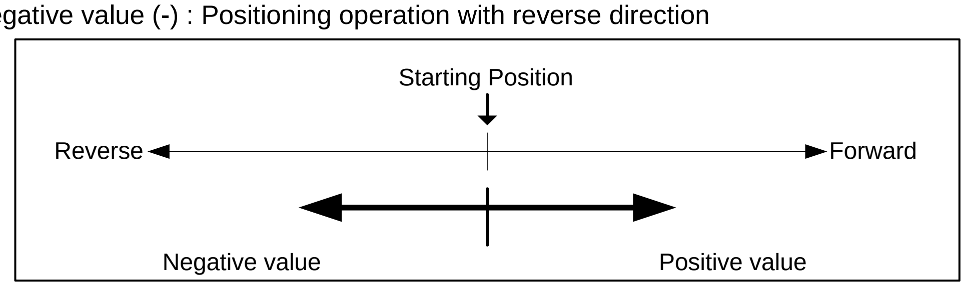
(a) Execute positioning control from starting position to goal position designated in motion function block.
(b) Positioning control is executed based on home position designated in homing.
(c) Moving direction is decided by starting position and goal position.
  1) Starting Position < Goal Position : Forward Positioning Operation
  2) Starting Position > Goal Position : Reverse Positioning Operation

[Example]
1) Starting Position : 1000
2) Goal Position : 8000
Value of Forward movement is 7000 (7000=8000-1000)

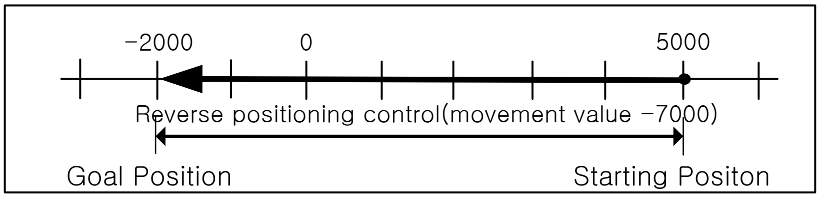


(2) Control by Incremental Coordinates

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



[Example]
1) Starting Position : 5000
2) Goal Position : -7000
In this condition, it moves reversely and stops at -2000.

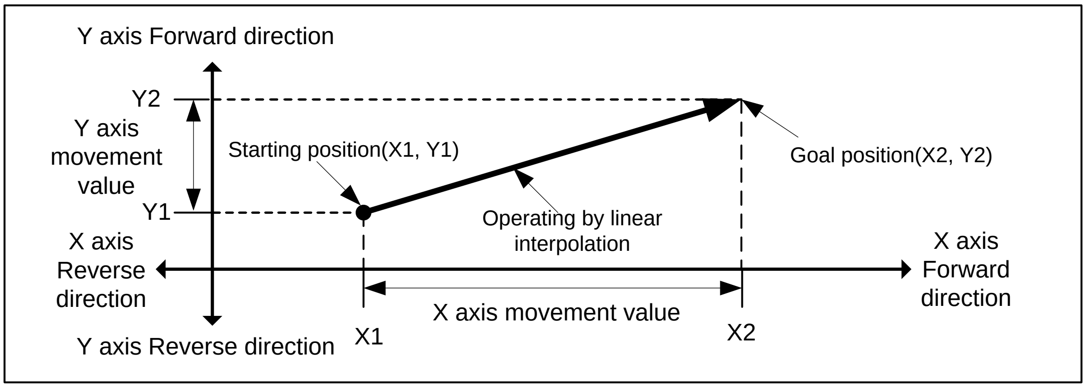


1.3.2 Interpolation Control

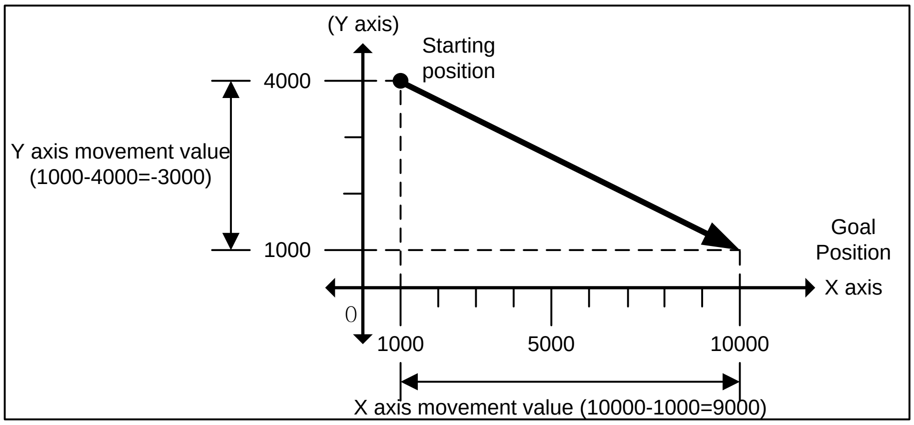
(1) Linear Interpolation Control

Execute Linear interpolation control with designated axis at start position. (Current position)
Combination of interpolation axis is unlimited and it is available to execute max. 4 axis Linear interpolation control.

(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.
a)Starting position < Goal position : Positioning operation with forward direction
b)Starting position > Goal position : Positioning operation with reverse direction



[Example]
a)Starting Position (1000, 4000)
b)Goal Position (10000, 1000)
In this condition, operation is as follows.



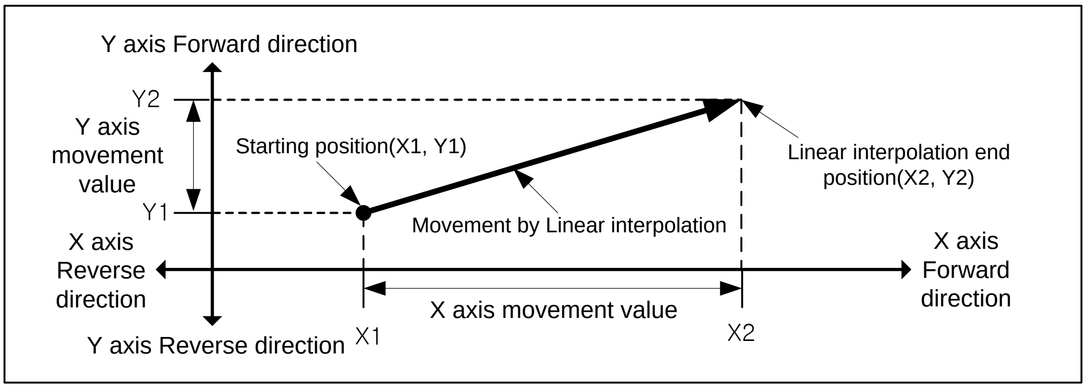
(b) Linear Interpolation by incremental coordinates

1) Goal value becomes movement value.

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

a)Positive value (+ or 0) : Positioning operation with forward direction

b)Negative value (-) : Positioning operation with reverse direction

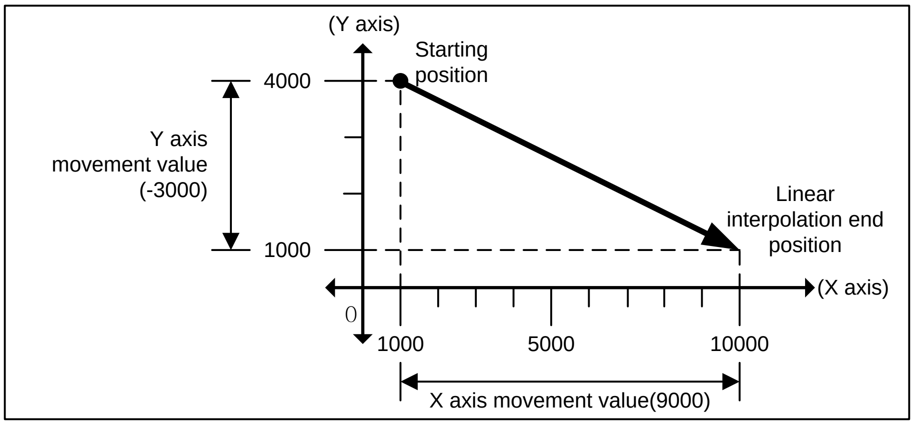


[Example]

a)Starting position (1000, 4000)

b)Goal position (9000, -3000)

In this condition, operation is as follows.



(2) Circular Interpolation Control

Execute interpolation operation along the trace of circle with 2 axes in forward direction that already designated for each axis.

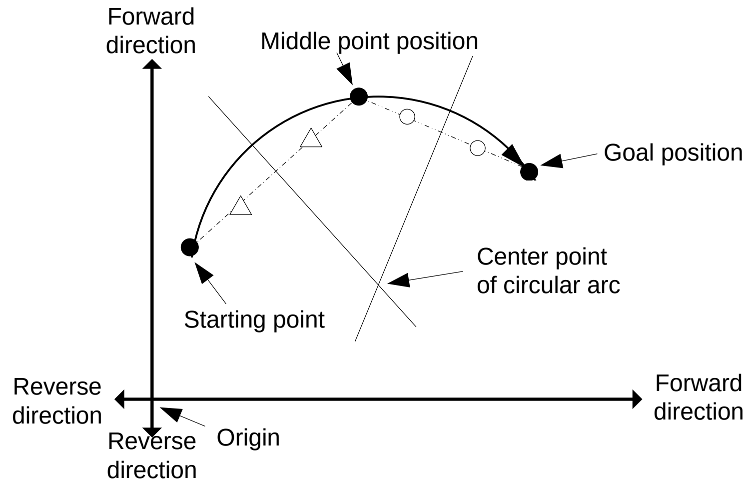
Circular interpolation has 3 types according to auxiliary point, Middle point method passing auxiliary point, Center point method using auxiliary point as center of circle and Radius method using auxiliary point as radius of circle.

The combination of 2 axes that used in circular interpolation is unlimited. Any of the two axes from the actual axes (1-axis to 32-axis) or virtual axes (37-axis to 40-axis) can be used.)

(a) Middle Point Specified Circular interpolation

1) Starts operating at starting position and executes circular interpolation through the designated middle point.

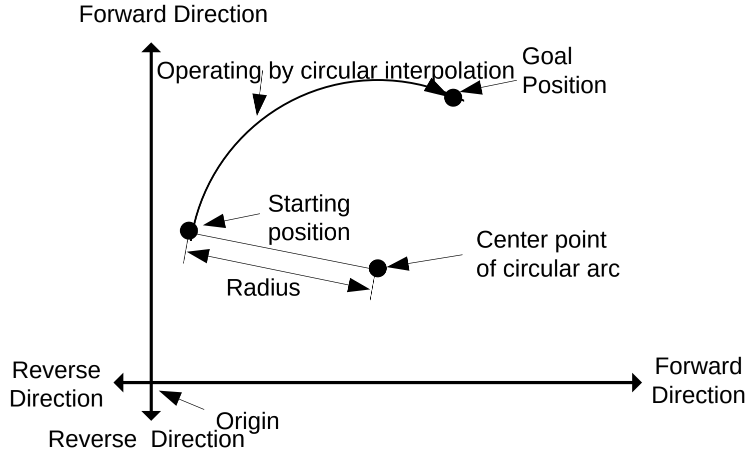
2) There will be a circular arc whose center point is crossing point of perpendicular bisection between starting position and middle point or middle point and goal position.



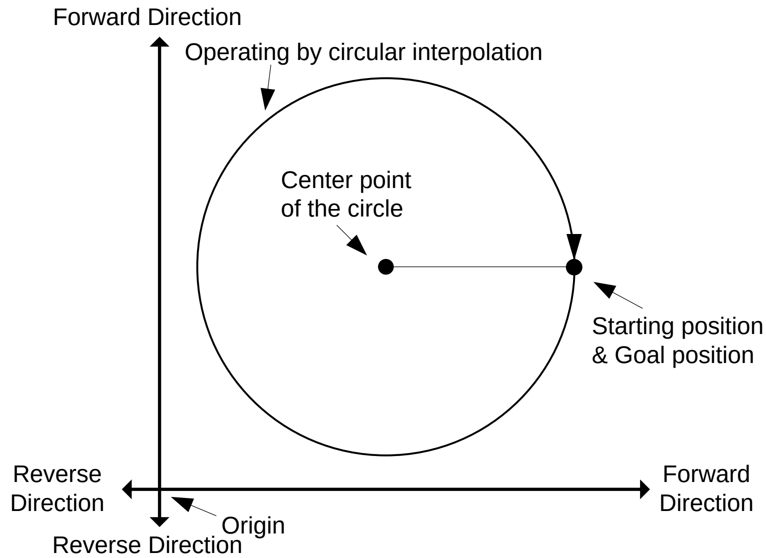
3) Movement direction is automatically designated by goal position and auxiliary point of circular interpolation.

(b) Center Point Specified Circular interpolation

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.



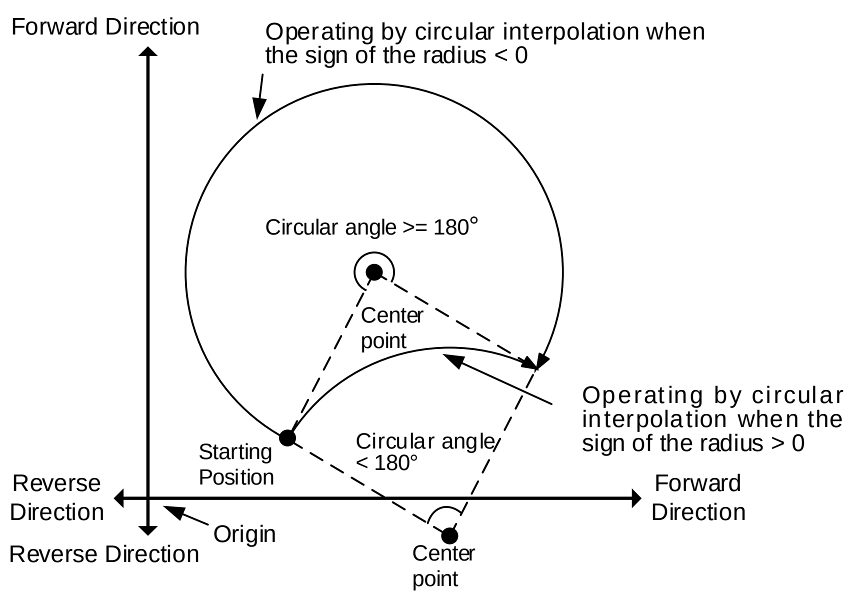
2) If the goal position is same as starting position, it is available to have an operation like a circle that has distance from starting point to auxiliary point as its radius.



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.

(3) Radius Specified Circular interpolation

(a) Starts operating from starting position and execute circular interpolation along trace of circular arc thathas 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.



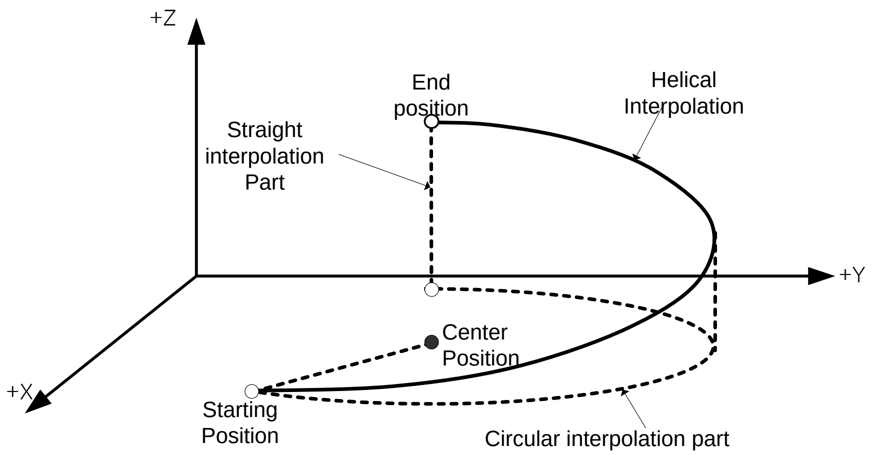
(b) In radius designation form, goal position can not be set the same as starting position.

(c) 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.

(4) Helical Interpolation

(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 (37 axes to 40 axes) are used.



1.3.3 Speed Control

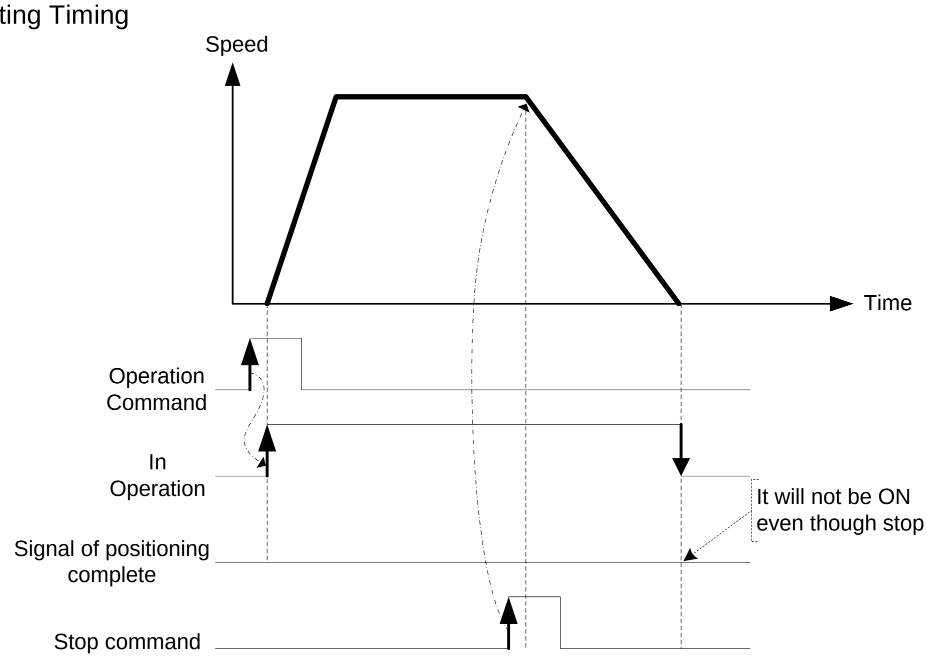
(1) Execution is made by speed control commands, and the operation proceeds at the established rate until buffer commands are executed, or stop commands are entered.

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

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

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

(3) Operating 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 command or stop command is entered.

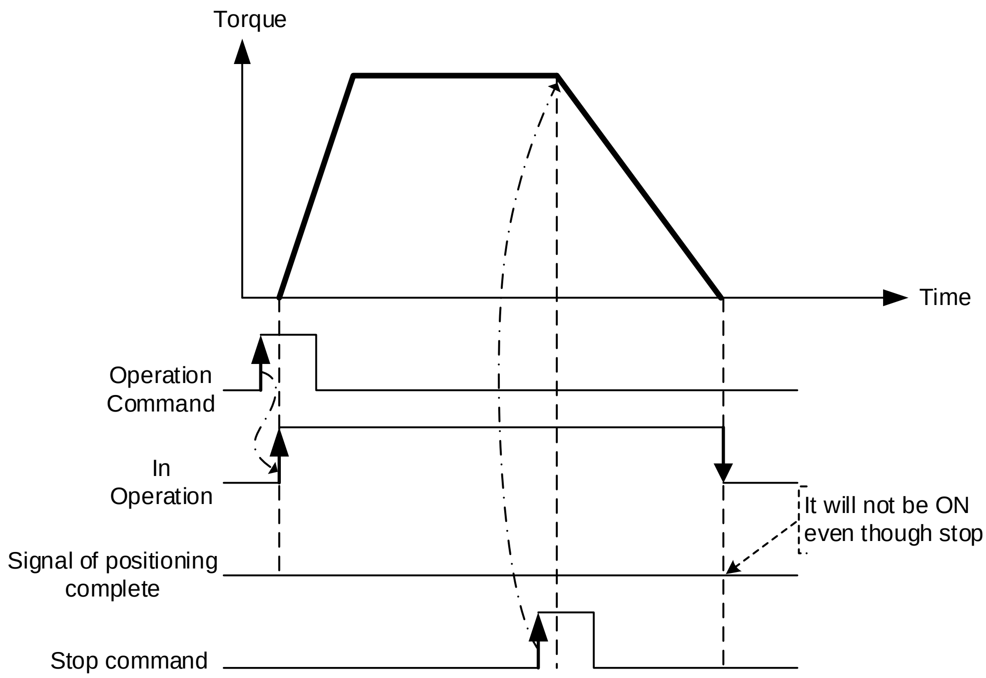
(2) Torque control includes forward operation and a reverse operation..

(a) Forward operation: When direction input setting is ‘1-forward’ direction

(b) Reverse operation: When direction input setting is 2-reverse’ direction

.

(3) Operating Timing



Chapter 2 Specification

2.1 General Specification

The following table shows the general specification of XGT series.

<table>
<tr><th>No.</th><th>Item</th><th colspan="4">Specification</th><th>Related specifications</th></tr>
<tr><td>1</td><td>Ambient temperature</td><td colspan="4">0 ~ 55 °C</td><td>-</td></tr>
<tr><td>2</td><td>Storage temperature</td><td colspan="4">−25 ~ +70 °C</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="9">5</td><td rowspan="9">Vibration resistance</td><td colspan="3">Occasional vibration</td><td>-</td><td>-</td></tr>
<tr><td>Frequency</td><td>Acceleration</td><td>Amplitude</td><td>How many times</td><td rowspan="8">IEC61131-2</td></tr>
<tr><td>5 ≤ f < 8.4Hz</td><td>-</td><td>3.5mm</td><td rowspan="7">10 times each directions
(X, Y and Z)</td></tr>
<tr><td>8.4 ≤ f ≤ 150Hz</td><td>9.8 ㎨ (1G)</td><td>-</td></tr>
<tr><td colspan="3">For continuous vibration</td></tr>
<tr><td>Frequency</td><td>Acceleration</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>4.9 ㎨ (0.5G)</td><td>-</td></tr>
<tr><td colspan="3"></td></tr>
<tr><td>6</td><td>Shock resistance</td><td colspan="4">• Peak acceleration: 147 m/s²(15G)
• Duration: 11ms
• Half-sine, 3 times each direction per each axis</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: ± 1,500V
DC: ± 900V</td><td>LSIS standard</td></tr>
<tr><td>Electrostatic discharge</td><td colspan="3">Voltage : 4kV (contact discharging)</td><td>IEC 61131-2,
IEC 61000-4-2</td></tr>
<tr><td>Radiated electromagnetic field noise</td><td colspan="3">80 ~ 1,000 MHz, 10V/m</td><td>IEC 61131-2,
IEC 61000-4-3</td></tr>
<tr><td rowspan="2">Fast transient /bust noise</td><td>Segment</td><td>Power supply module</td><td>Digital/analog input/output communication interface</td><td rowspan="2">IEC 61131-2,
IEC 61000-4-4</td></tr>
<tr><td>Voltage</td><td>2kV</td><td>1kV</td></tr>
<tr><td>8</td><td>Environment</td><td colspan="4">Free from corrosive gasses 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>

Note

1. 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="2">Items</th><th colspan="3">Specification</th><th>Remark</th></tr>
<tr><td rowspan="8">Input</td><td>Rated input voltage</td><td colspan="3">AC100V~AC240V</td><td></td></tr>
<tr><td>Input frequency</td><td colspan="3">50/60Hz</td><td></td></tr>
<tr><td rowspan="2">Input current</td><td colspan="3">0.7A or less</td><td>AC100V</td></tr>
<tr><td colspan="3">0.4A or less</td><td>AC240V</td></tr>
<tr><td>Inrush current</td><td colspan="3">120Apeak or less</td><td>AC240V, Phase 90 degree</td></tr>
<tr><td>Leakage current</td><td colspan="3">3mA or less</td><td></td></tr>
<tr><td>Efficiency</td><td colspan="3">65% or more</td><td></td></tr>
<tr><td>Permitted momentary power failure</td><td colspan="3">10ms or less</td><td></td></tr>
<tr><td rowspan="9">Output</td><td rowspan="3">Output voltage</td><td>Voltage</td><td>Output voltage ripple range</td><td>current</td><td></td></tr>
<tr><td>+5V</td><td>4.90V~5.20V</td><td>4A</td><td></td></tr>
<tr><td>+24V</td><td>21.1V~26.9V</td><td>0.4A</td><td></td></tr>
<tr><td rowspan="3">Ripple & Noise</td><td>Voltage</td><td>Ripple</td><td>Noise</td><td></td></tr>
<tr><td>+5V</td><td>100mVpp or less</td><td>200mVpp or less</td><td></td></tr>
<tr><td>+24V</td><td colspan="2">400mVpp or less</td><td></td></tr>
<tr><td rowspan="3">Protecting overcurrent</td><td>Voltage</td><td colspan="2">Current</td><td></td></tr>
<tr><td>+5V</td><td colspan="2">4.4A or more</td><td></td></tr>
<tr><td>+24V</td><td colspan="2">0.44A or more</td><td></td></tr>
</table>

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

Note

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. Overcurrent protection
   (a) When the voltage exceeding the standard is applied to the circuit of DC5V,DC24V, overcurrent protection device interrupts the circuit and stops the system.
   (b) If overcurrent occurs, after removing the causes such as shortage of current capacity, short circuit, etc., restart the system.
3. Overvoltage protection
   When the voltage exceeding the standard is applied to the circuit of DC5V, overvoltage protection device interrupts the circuit and stops the system.

2.3 Performance Specification

The following table shows the Performance specifications of motion controller.

2.3.1 Function Specification

Items			Specification
Operation method			Main task/Periodic task: Fixed cyclic operation, reiterative operation Initial task: Only once at the time of entering the RUN
Control cyclic			Main task cyclic time: 0.5ms, 1ms, 2ms, 4ms Periodic task cyclic 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
Processing speed	Basic		6.25ns or more (General point/coil)
	Move		5ns or more (Word type)
	Arithmetic		30ns or more (Word type)
Program	number		Max. 256
	Capacity		10MB(Motion program,), 10MB(NC program)
Data area	Symbolic 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
Timer			No limit in points, Time range: 0.001~ 4,294,967.295sec(1,193hour)
Counter			No limit in points, Counter range: 64 bit range
Program			Initial program, Main task program, Periodic task program, NC program
Operation mode			RUN, STOP
Restart mode			Cold, Warm
Self-diagnosis function			Cyclic error monitoring, time share over detection of task program, memory abnormal, power abnormal, etc.
Back-up method			Retain area setting in basic parameter or of variables

Items		Specification
Number of control axis		32 axes(Real/Virtual axis), 4 axes(Virtual axis), 64 Slaves(Included real/virtual axis)
Communication		EtherCAT (CoE: CANopen over EtherCAT, FoE: File Access over EtherCAT)
Communication/Control period		0.5ms, 1ms, 2ms, 4ms (Same with main task period)
Servo drive		Servo drive to support EtherCAT CoE
Control unit		pulse, mm, inch, degree
Control method		Position, Velocity, Torque(Servo drive support), Synchronous, Interpolation
Range of position		±LREAL, 0
Range of velocity		±LREAL, 0
Torque unit		Rated torque % designation
Acc./Dec. process		Trapezoid type, S-type (Setting to specify the Jerk at function block)
Rage of Acc./Dec.		±LREAL, 0
Manual operation		JOG operation
CAM operation		Max. 32 CAM profile (32,768 points/32 CAM profile)
Absolute System		Available (When using absolute encoder type servo drive)
Encoder input	Channel	2 channels
	Max. input	500 kpps
	Input method	Line drive input (RS-422A IEC specification), Available open collector output type encoder
	Input type	CW/CCW, Pulse/Dir, Phase A/B
Digital input/output	Input	8 points , 24V, 4mA
	Output	16 points (Transistor output), 24V, 0.5A/1 point
Analog input/output	Input	2 channels Voltage input range: -10~10V, 0~10V, 1~5V, 0~5V Current input range: 4~20mA, 0~20mA Resolution: 14 bit (1/16,000) Accuracy: ±0.2%(25℃), ±0.3%(0~55℃) Conversion speed: 0.5ms/channel Absolute maximum input: Voltage ±15VDC, Current ±30mADC
	Output	2 channels Voltage output range: -10~10V, 0~10V, 1~5V, 0~5V Resolution: 14 bit (1/16,000) Accuracy: ±0.2%(25℃), ±0.3%(0~55℃) Conversion speed: 0.5ms/channel

Items		Specification
Remote I/O		Maximum 64 slaves
Coordinate system		Cartesian, Delta
SD memory	Type	Micro SD/SDHC
	File system	FAT 32
	Capacity	Max. 32GB installation (Memory over 8GB can use only 8GB of overall area)
	Service	Program back-up/Restoration, Booting operation, Data log
Ethernet	Speed	Auto/10Mbps/100Mbps, 1 port
	Distance	100m
	Service	Loader service (XG5000) LSIS protocol supported (XGT, Modbus TCP) FTP server: Reading/Writing function a file of SD memory from other device SNTP client
USB	Performance	USB 2.0, 1 port
	Service	Loader service (XG5000)
Error indication		Indicated by LED
Weight		790g

Note

1. LREAL range: 2.2250738585072^{e-308} ~ 1.79769313486232^{e+308}
2. LREAL(positive) range: 0 ~1.79769313486232^{e+308} (Excluded 0)

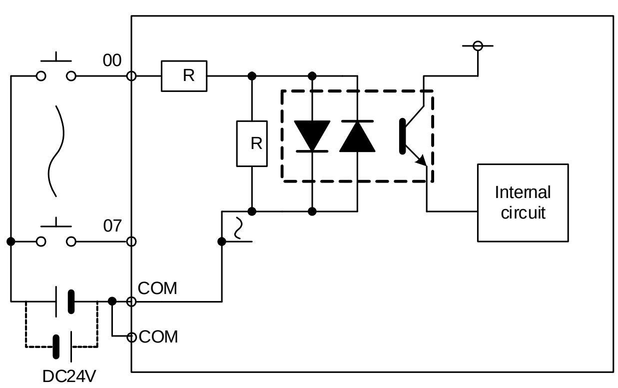
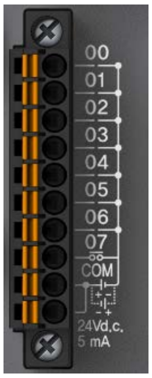
2.3.2 Communication Specification

Item	Specification
Communication protocol	EtherCAT
Support specification	CoE(CANopen over EtherCAT)
Physical layer	100BASE-TX
Communication speed	100Mbps
Topology	Daisy Chain
Communication cable	Over 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	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

2.3.3 Internal input/Output Specification

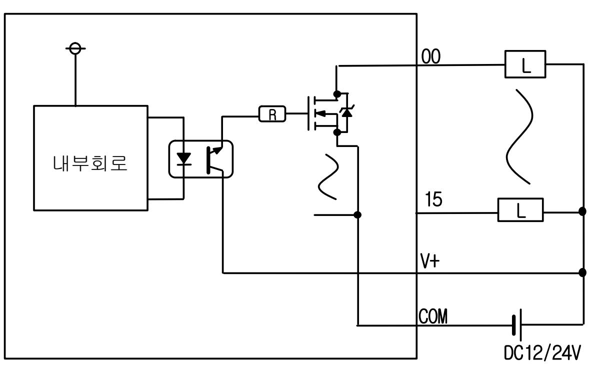
1. **Input specifications (source/sink type)**

Item		Specification
Input point		8 point
Insulation method		Photo-coupler insulation
Rated input voltage		24V
Rated output voltage		About 6mA
Used 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	Initial: 1ms(0.5/1/3/5/10/20/70/100ms: I/O Parameter setting)
	On → Off	
Insulation voltage		AC560Vrms/3 Cycle (Altitude 2,000m)
Insulation resistance		Insulation resistance 10 ㏁ or more
Common method		8 points/COM

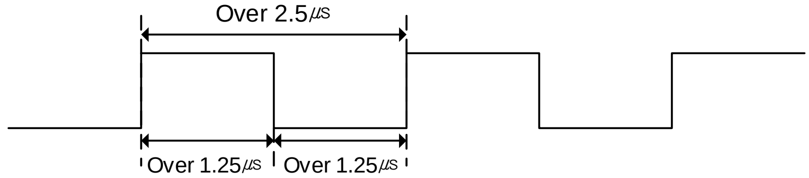
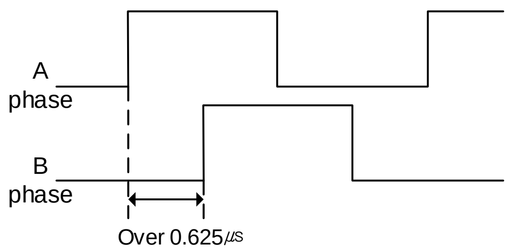
Circuit	No.	Point	External
	00	%IX0.0.0	
	01	%IX0.0.1	
	02	%IX0.0.2	
	03	%IX0.0.3	
	04	%IX0.0.4	
	05	%IX0.0.5	
	06	%IX0.0.6	
	07	%IX0.0.7	
	COM	-	
	COM	-	

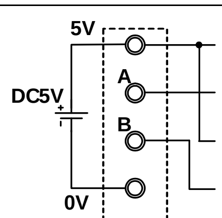
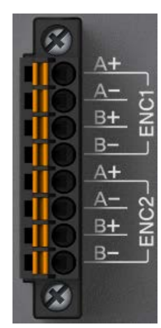
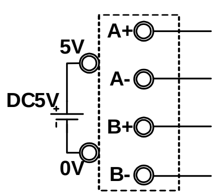
2. Output specifications (sink type)

Item		Specification
Output point		16 point
Insulation method		Photo-coupler insulation
Rated load voltage		DC 12V / 24V
Used load voltage range		DC10.2V~26.4V
Maximum load current		0.5A /point, 2A/COM
Off leakage current		0.1mA or less
Maximum inrush current		4A / 10ms or less
Maximum voltage drop(On)		DC 0.3V or less
Surge absorber		Zener diode
Response time	Off→On	1ms or less
	On→Off	1ms or less(Rated load, resistive load)
Common method		8 points/COM
External power	Voltage	DC12/24V±10% (Ripple voltage 4Vp-p or less)
	Current	10mA or less (DC24V connection)

Circuit	No.	Point	External
	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 Encoder Input Specification

Item	Specification	
Input voltage	5V (3V ~ 6V)	In accordance with RS-422A Line Driver Level
Input current	2 ㎃~7.5 ㎃	
Min. On guarantee voltage	2.5V	
Max. Off guarantee voltage	1.7V	
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	

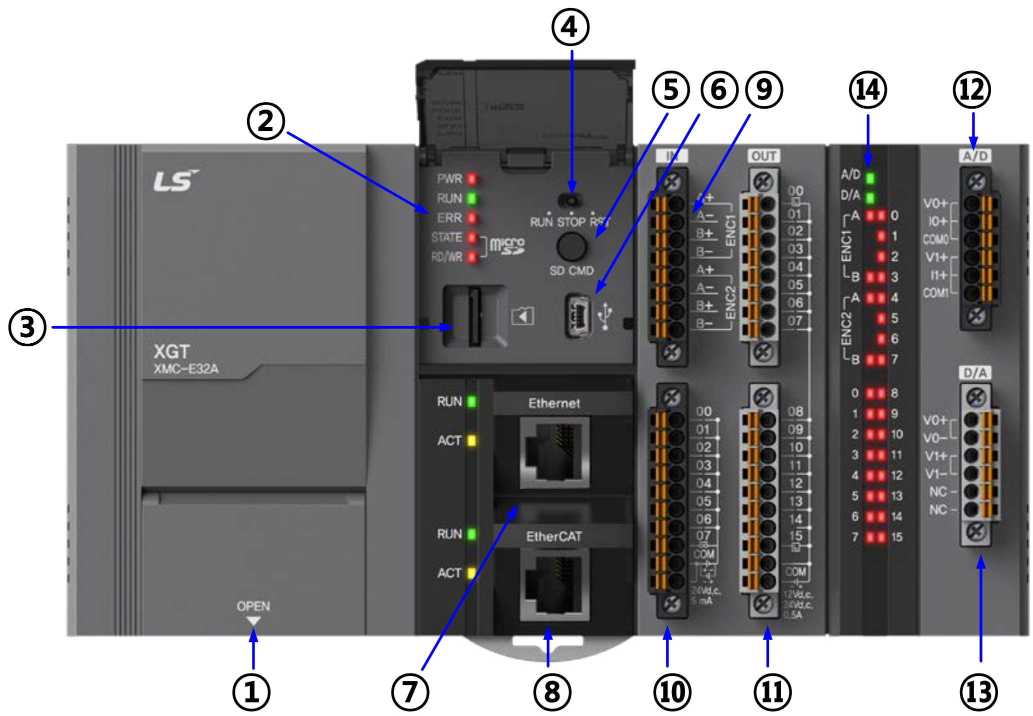
Circuit	No.	Point	External
Note1 	ENC1 A+	Encoder 1 A+ input	
	ENC1 A-	Encoder 1 A- input	
	ENC1 B+	Encoder 1 B+ input	
	ENC1 B-	Encoder 1 B- input	
Note2 	ENC2A+	Encoder 2 A+ input	
	ENC2 A-	Encoder 2 A- input	
	ENC2 B+	Encoder 2 B+ input	
	ENC2 B-	Encoder 2 B- input	

Note

Note 1 : Encoder of 5V voltage output type(Open collector)
Note 2 : Encoder of 5V voltage output type(Line driver)

2.4 The Name of Each Part

2.4.1 The Name of Each Part

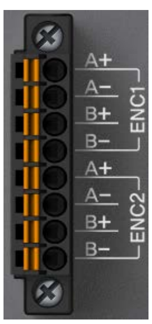
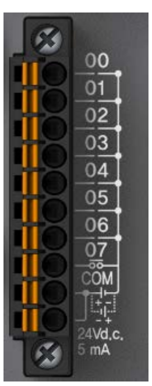


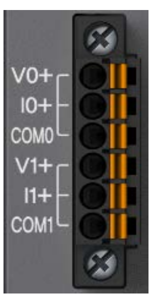
No.	Name	Description
①	Power terminal	AC 110/220V power input, LG terminal, DC24V output
②	Staus display LED	Displays the motion controller's operation mode. • PWR(Red light on): The power is supplied • RUN(Green light on): During RUN mode • ERR(Flickering Red light): Occurrence of errors during operation • 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's operation is executed. • STOP: Program's operation is stopped. • 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

No.	Name	Description
⑥	USB port	Port to access to XG5000
⑦	Ethernet port	Port to communicate Ethernet
⑧	EtherCAT port	Port to communicate EtherCAT
⑨	Encoder input connector	-
⑩	Digital input connector	-
⑪	Digital output connector	-
⑫	Analoog input connecotr	-
⑬	Analog output connector	-
⑭	Display input/output operaiton LED	Digital input/oupt, Analog input/output, Encoder input

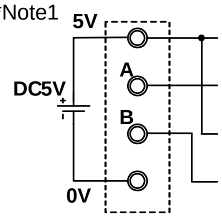
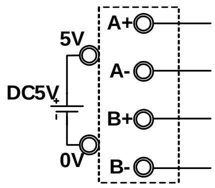
2.4.2 Specification of Interface with External Device

1. Pin arrangement of connector

<table>
<tr><th>External</th><th colspan="2">Signal name</th><th>Signal direction</th></tr>
<tr><td rowspan="8">
</td><td>ENC1 A+</td><td>Encoder 1 A+</td><td rowspan="8">Input</td></tr>
<tr><td>ENC1 A-</td><td>Encoder 1 A-</td></tr>
<tr><td>ENC1 B+</td><td>Encoder 1 B+</td></tr>
<tr><td>ENC1 B-</td><td>Encoder 1 B-</td></tr>
<tr><td>ENC2 A+</td><td>Encoder 2 A+</td></tr>
<tr><td>ENC2 A-</td><td>Encoder 2 A-</td></tr>
<tr><td>ENC2 B+</td><td>Encoder 2 B+</td></tr>
<tr><td>ENC2 B-</td><td>Encoder 2 B-</td></tr>
<tr><td rowspan="9">
</td><td>IN0</td><td>Input signal 0</td><td rowspan="8">Input</td></tr>
<tr><td>IN1</td><td>Input signal 1</td></tr>
<tr><td>IN2</td><td>Input signal 2</td></tr>
<tr><td>IN3</td><td>Input signal 3</td></tr>
<tr><td>IN4</td><td>Input signal 4</td></tr>
<tr><td>IN5</td><td>Input signal 5</td></tr>
<tr><td>IN6</td><td>Input signal 6</td></tr>
<tr><td>IN7</td><td>Input signal 7</td></tr>
<tr><td>COM</td><td>Input signal Common</td><td>Input</td></tr>
</table>

<table>
<tr><th>External</th><th colspan="2">Signal name</th><th>Signal direction</th></tr>
<tr><td rowspan="18"></td><td>OUT0</td><td>Output signal 0</td><td rowspan="16">Output</td></tr>
<tr><td>OUT1</td><td>Output signal 1</td></tr>
<tr><td>OUT2</td><td>Output signal 2</td></tr>
<tr><td>OUT3</td><td>Output signal 3</td></tr>
<tr><td>OUT4</td><td>Output signal 4</td></tr>
<tr><td>OUT5</td><td>Output signal 5</td></tr>
<tr><td>OUT6</td><td>Output signal 6</td></tr>
<tr><td>OUT7</td><td>Output signal 7</td></tr>
<tr><td>OUT8</td><td>Output signal 8</td></tr>
<tr><td>OUT9</td><td>Output signal 9</td></tr>
<tr><td>OUT10</td><td>Output signal 10</td></tr>
<tr><td>OUT11</td><td>Output signal 11</td></tr>
<tr><td>OUT12</td><td>Output signal 12</td></tr>
<tr><td>OUT13</td><td>Output signal 13</td></tr>
<tr><td>OUT14</td><td>Output signal 14</td></tr>
<tr><td>OUT15</td><td>Output signal 15</td></tr>
<tr><td>24V</td><td>DC24V</td><td rowspan="2">Input</td></tr>
<tr><td>GND</td><td>DC24V GND</td></tr>
<tr><td rowspan="6"></td><td>V0+</td><td>Analog voltage input 0</td><td rowspan="6">Input</td></tr>
<tr><td>I0+</td><td>Analog current input 0</td></tr>
<tr><td>COM0</td><td>Analog input 0 common</td></tr>
<tr><td>V1+</td><td>Analog voltage input 1</td></tr>
<tr><td>I1+</td><td>Analog current input 1</td></tr>
<tr><td>COM1</td><td>Analog input 1 common</td></tr>
<tr><td rowspan="6"></td><td>V0+</td><td>Analog voltage output 0+</td><td rowspan="6">Output</td></tr>
<tr><td>V0-</td><td>Analog voltage output 0-</td></tr>
<tr><td>V1+</td><td>Analog current output 0+</td></tr>
<tr><td>V1-</td><td>Analog current output 0-</td></tr>
<tr><td>NC</td><td>No Connection</td></tr>
<tr><td>NC</td><td>No Connection</td></tr>
</table>

2. Encoder internal circuit

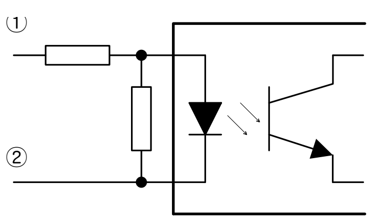
Item	Pin No.	Signal	
*Note1 	ENC1A+	ENC1A+	Encoder 1A+ input
	ENC1A-	ENC1A-	Encoder 1 A- input
	ENC1B+	ENC1B+	Encoder 1 B+ input
	ENC1B-	ENC1B-	Encoder 1 B- input
*Note2 	ENC2A+	ENC2A+	Encoder 2 A+ input
	ENC2A-	ENC2A-	Encoder 2 A- input
	ENC2B+	ENC2B+	Encoder 2 B+ input
	ENC2B-	ENC2B-	Encoder 2 B- input

Note

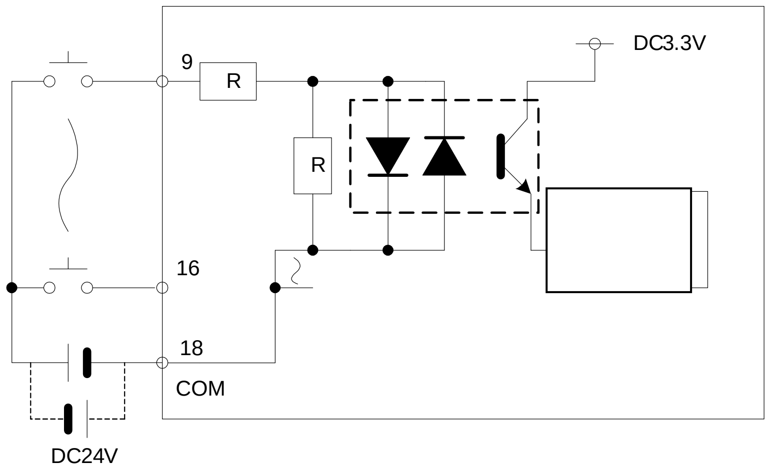
* Note1
Wiring of encoder 1 is example about 5V voltage output type (open collector). When using 12V, 24V type MPG, change the input voltage from 5V to 12V or 24V and in case of 12V, connect 910Ω resistor to ENC1 A+(pin 1), ENC1 B+ (pin3), in case of 24V, 2.4㏀ resistor, before connecting the power source (adding PULL-UP resistor is needed)

* Note2
Wiring of encoder 2 is example about 5V voltage output type (line driver)

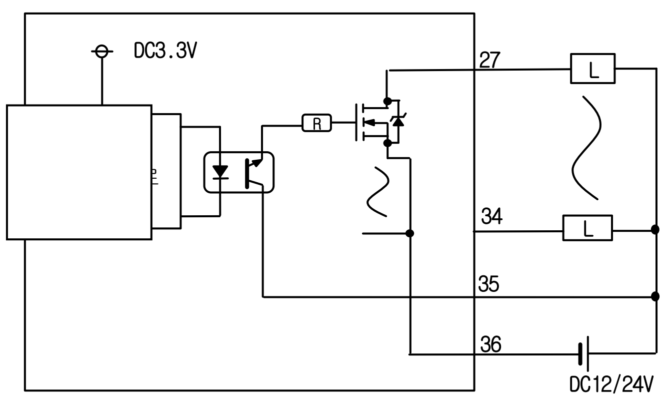
This describes the internal circuit of the module when connecting the encoder.

Item	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 internal circuit**



4. **Output internal circuit**



Chapter 3 Operation Order and Installation

3.1 Operation Order

Here describes the Operation order of motion controller.

Start

↓

Specify motion control operation method and control unit

↓

Specify the number of axis to be connected

↓

Specify the servo type and capacity

↓

Install the XG5000 on the PC

↓

Mount the motion controller on the base

↓

Connect the motor and external signal to the servo ← External emergency stop signal, External upper limit signal, External lower limit signal, Home signal, DOG signal

↓

Connect Ethernet communication cable between positioning module and servo, and between servos ← Max. Communication distance: 100m, Communication cable: STP cable

↓

Turn the Servo on

↓

Turn the motion controller on ← Check whether RDY LED is ON or not

↓

Connect the communication cable between USB(Ethernet) ports of PC and motion controller

↓

Write the motion program and parameter of motoin controller in the XG5000 and download it to the Controller ← Basic parameter, axis group parameter, master parameter, common parameter, motion program

↓

Set up the servo axis and establish communication between Servo and motion controller ← Allocate the axis number to each servo by using XG5000

↓

Write the servo parameter in the XG5000 and download it to servo ← Reading/Writing servo parameter and servo tuning are available by XG5000

↓

Execute a test run by using XG5000 ← Check the operation status and modify the module parameter and servo parameter

↓

Write the program for motion control operation through XG5000

↓

Start the motion control operation

↓

End

3.2 Installation

3.2.1 Safety Precautions

⚠ Danger

- Please design protection circuit at the external of Contrller 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. Refer to 4.1.1 Fail Safe circuit.
  (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.
- 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. 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

(1) Install the control panel available for water-proof, anti-vibration.
(2) The place free from continuous impact or vibration.
(3) The place not exposed to direct rays.
(4) The place with no dew phenomena by rapid temperature change.
(5) The place where surrounding temperature maintains 0-55℃.

2. Installation Construction

(1) In case of processing the screw hole or wiring, cares should be taken not to put the wiring remnants to Controller inside.
(2) Install on the good place to operate.
(3) Do not install the high voltage machine on the same Panel.
(4) The distance from duct or surrounding module shall be more than 50mm.
(5) Ground to the place where surrounding noise environment is good enough.

3.2.3 Notices in 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. It may cause the failure.
(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

⚠ Remark

▸ 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.

▸ Do not separate PCB from case.

Caution in handling

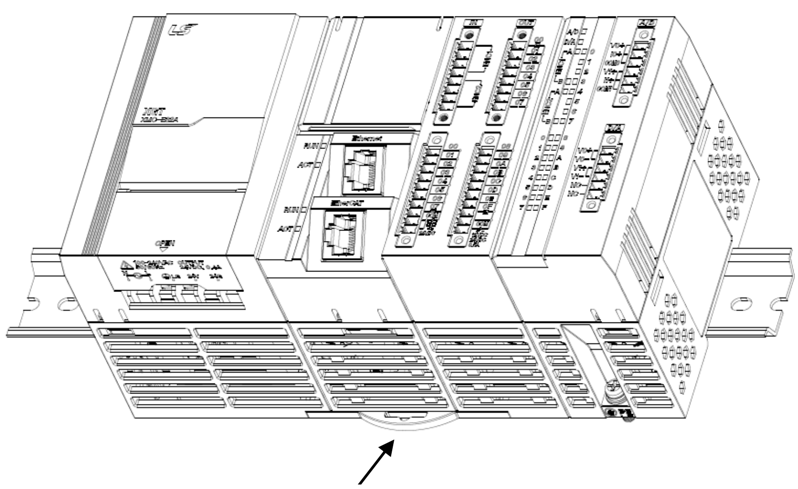
1. Use motion controller in the range of general specification specified by manual.
2. In case of usage out of range, it may cause electric shock, fire, malfunction, damage of product.

(1) Installation of motion controller

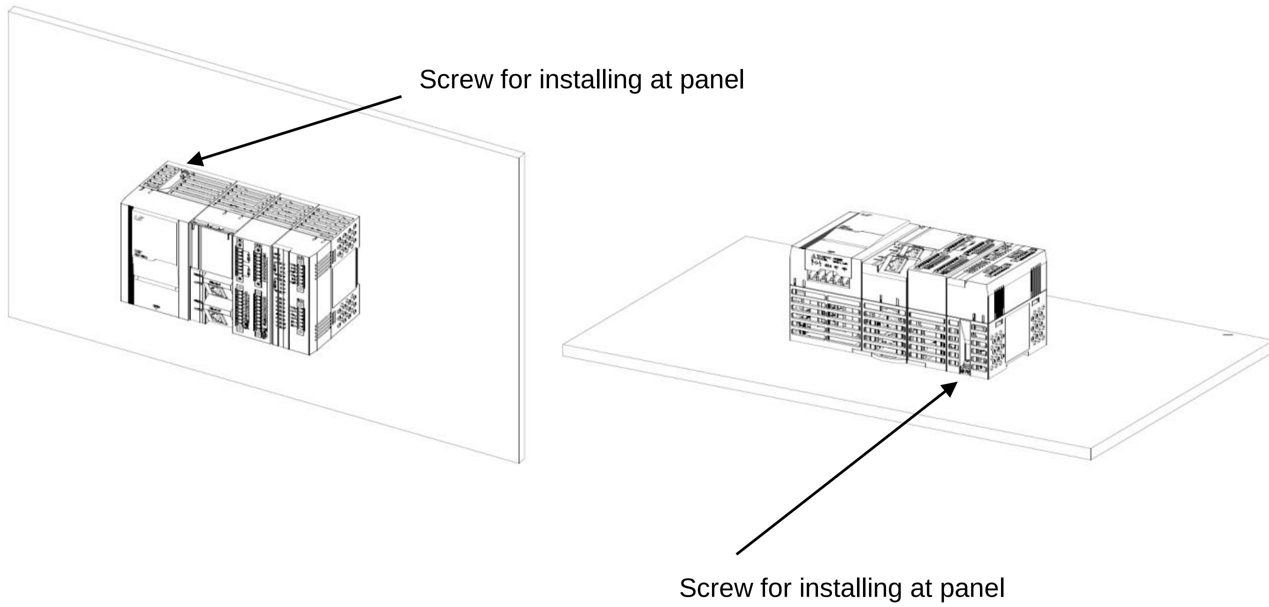
Motion controller has a hook for DIN rail (rail width: 35mm) so that cab be installed at DIN rail.

(a) In case of installing at DIN rail

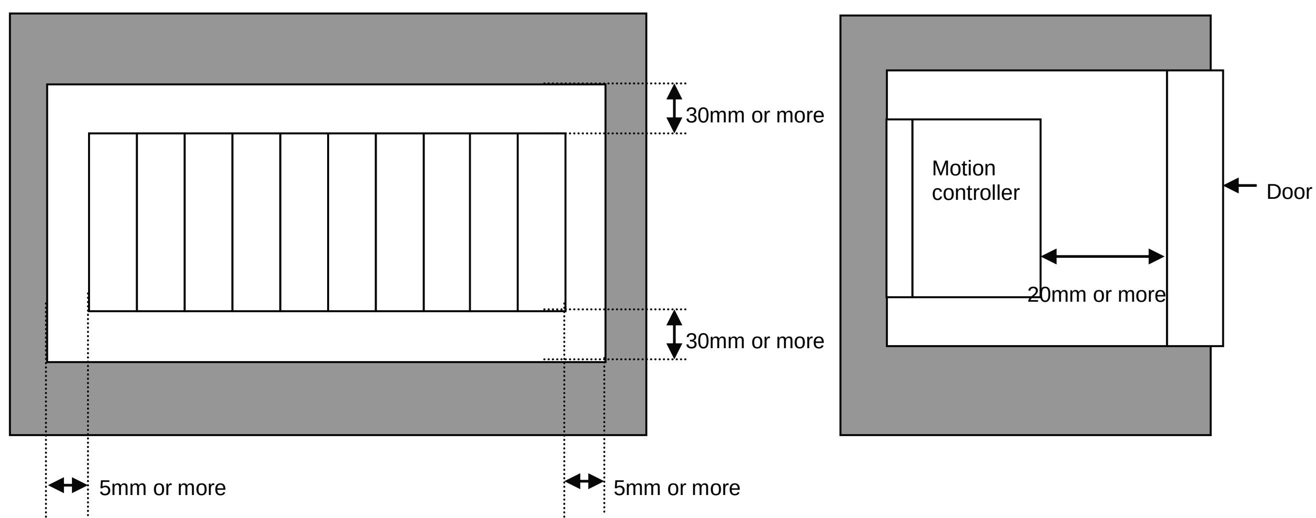
- Pull the hook as shown below for DIN rail at the bottom of motion controller and install it at DIN rail
- Push the hook to fix the module at DIN rail after installing motion controller at DIN rail



(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.

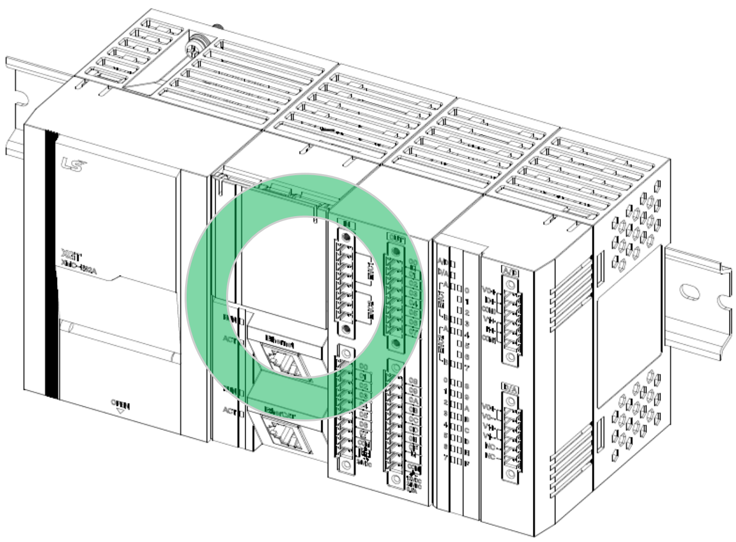


(2) Controller equipment locaiton
Keep the following distance between module and structure or part for ventilation, easy detachment and attachment.

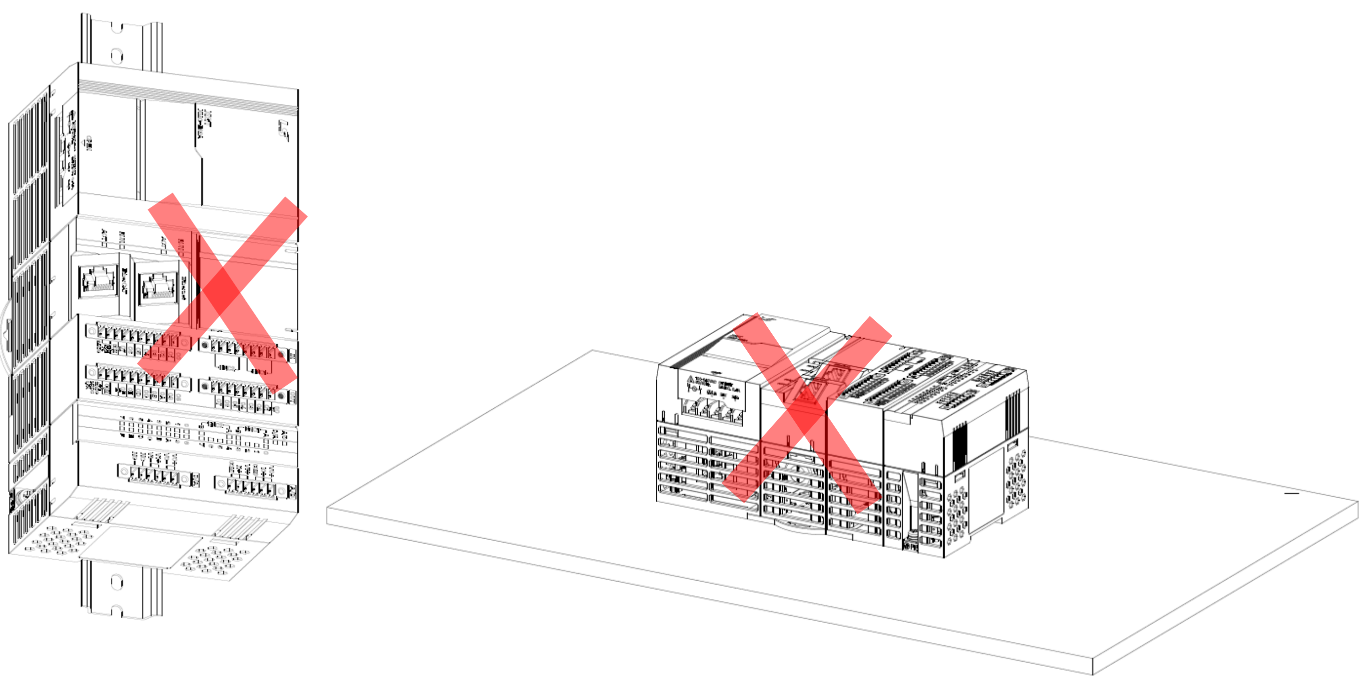


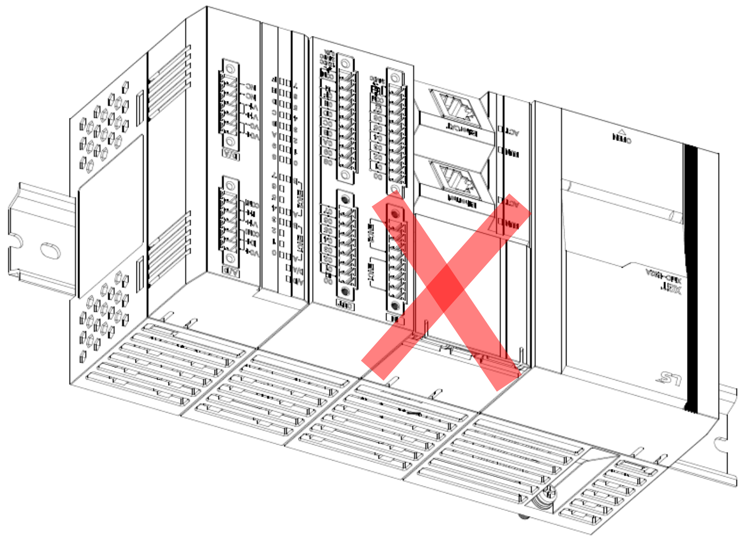
(3) Controller equipment direction

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



(b) Don’t install as shown below.

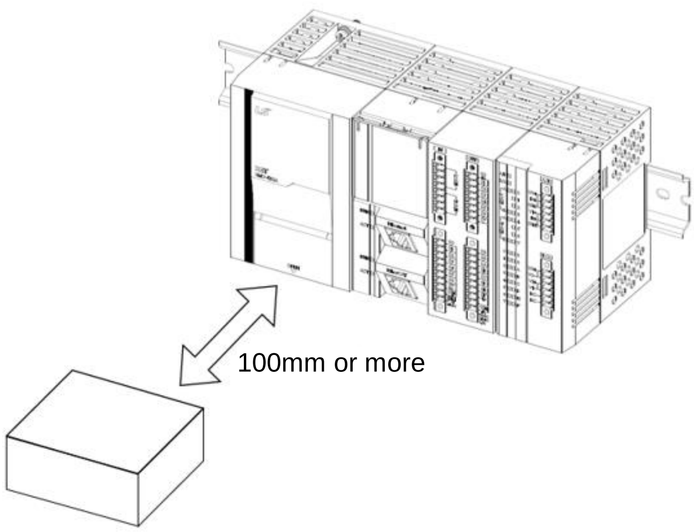


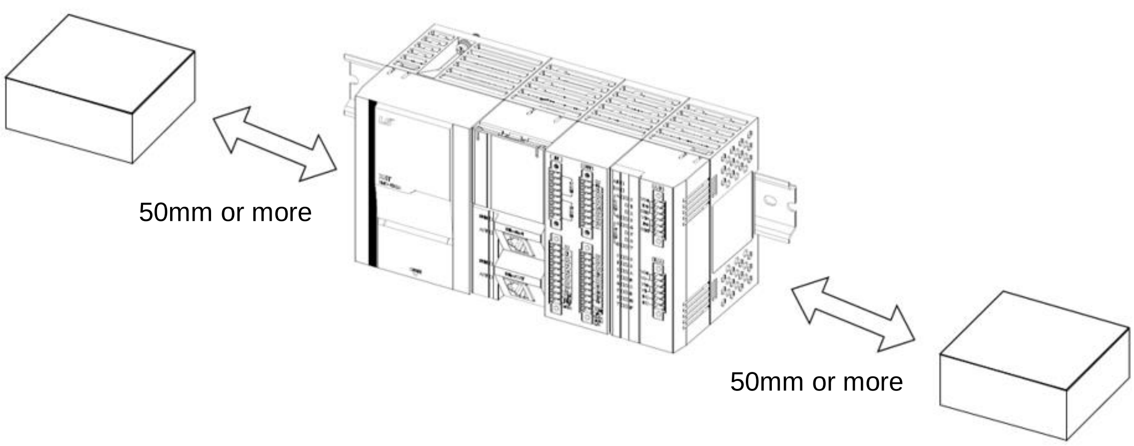


(4) Distance with other device

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

Device installed in front of motion controller: 100mm or more
Device installed beside motion controller: 50mm or more





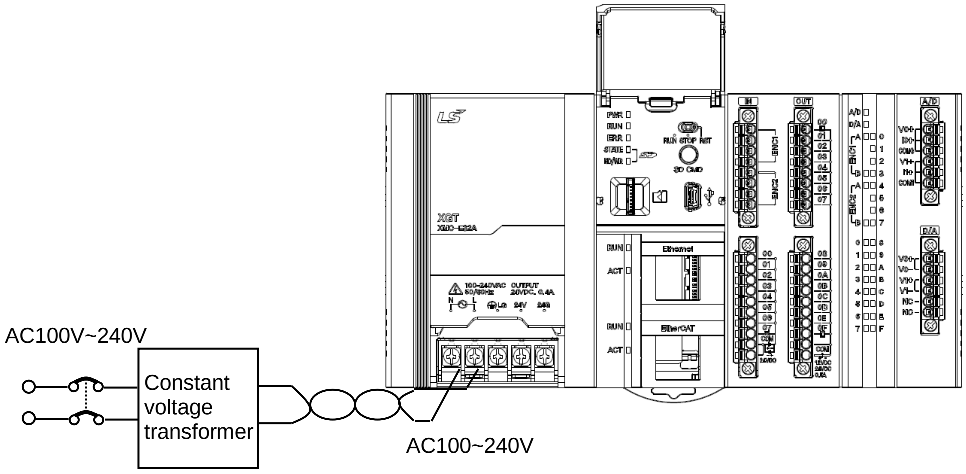
3.3 Notices in Wiring

3.3.1 Notices in Wiring

(1) The length of connecting cable between controller and drive machine shall be as short as possible. (Max. length: 2m and 10m).
(2) For alternating current and external I/O signal of controller, it is required to use the separate cables to avoid the surge or induction noise generated from the alternating current.
(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 controllerand 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.

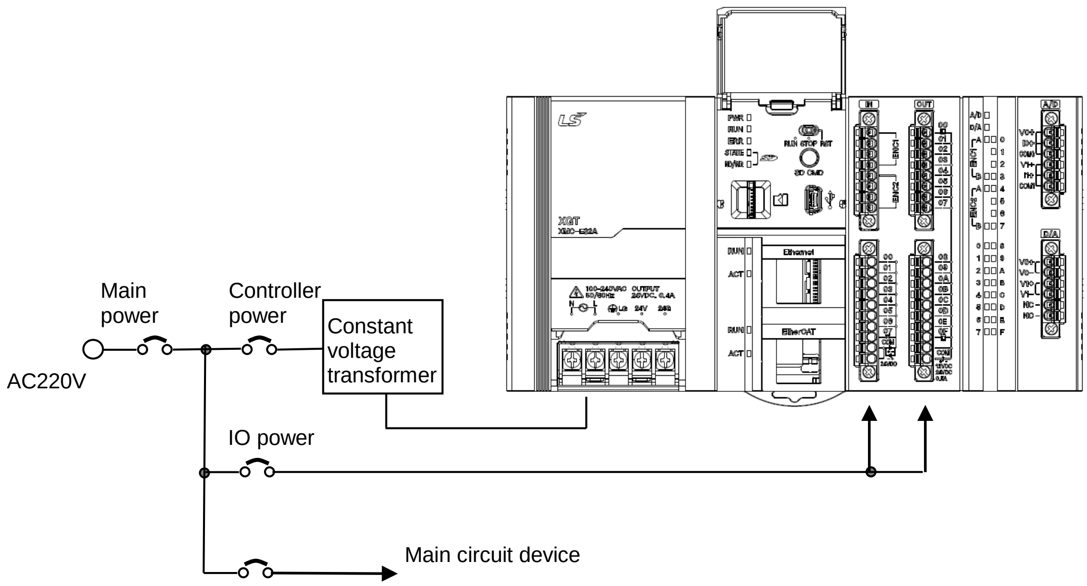
3.3.2 Power Wiring

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



(2) Connect noise that includes small noise between line and earth.
(When there is much noise, connect insulated transformer.)

(3) Isolate the contorller 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.

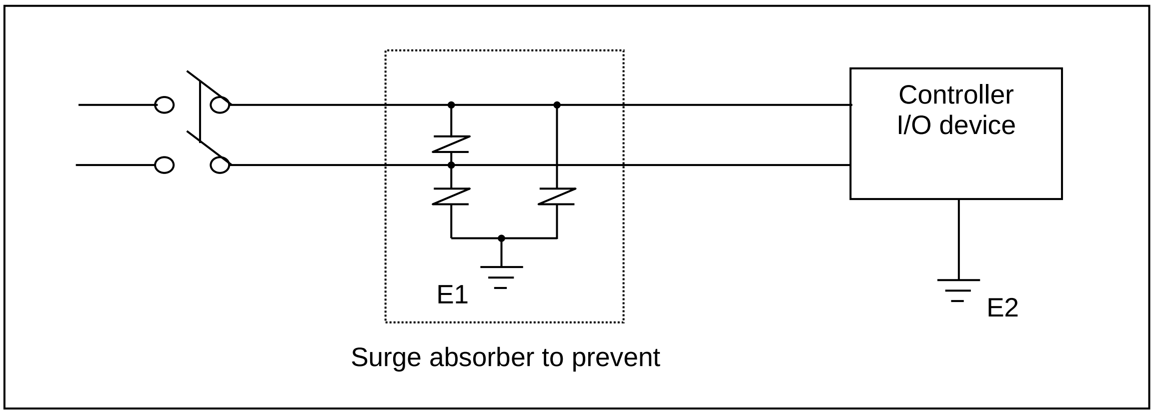
(6) AC power cables should be as thick as possible($2mm^2$) to reduce voltage drop.

(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) When noise penetration coure use an insulated shielding transformer or noise filter.

(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.



Note

Isolate the grounding(E1) of lightning surge absorber from the grounding(E2) of the controller.
Select a lightning surge absorber type so that the max. voltage may not the specified allowable voltage of the absorber.

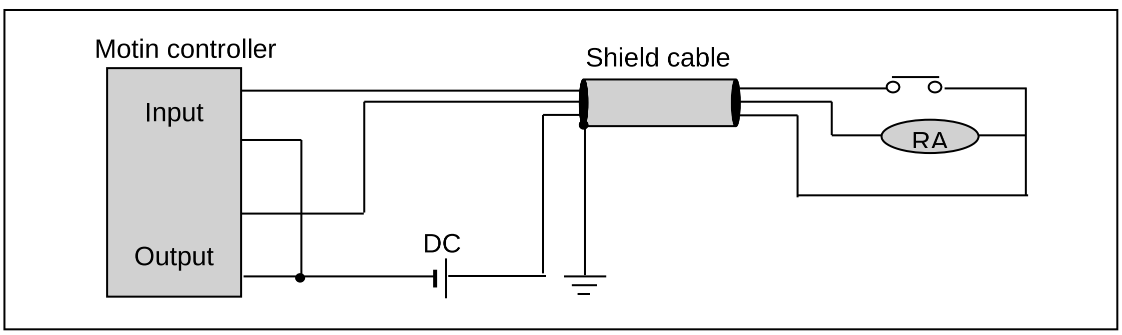
3.3.3 I/O Device Wiring

(1) The size of I/O device cable is limited to 0.3~2 mm^2 but it is recommended to select a size(0.3 mm^2) to use conveniently.

(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 contoller side should be grounded unless the main circuit cable and power cable can not be isolated.



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

3.3.4 Grounding(LG) Wiring

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

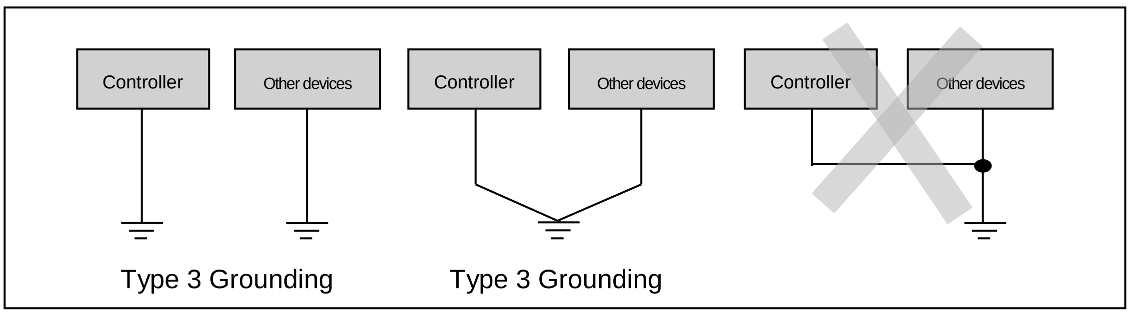
(2) LG ( ) is grounding for a power filter and used as a noise countermeasure. This controller performs sufficient noise countermeasures, but it is recommended to use LG if there is no specific reason. For the location of LG, please refer to the names of each part of 2.4.1.

(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 100 Ω)

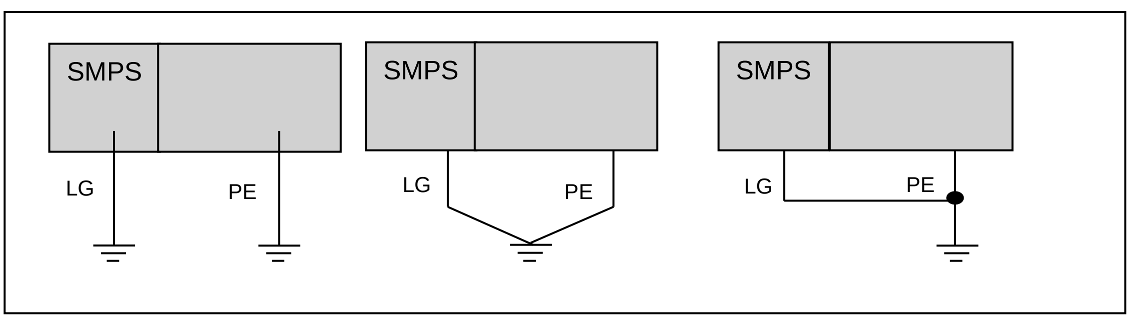
(b) If the exclusive grounding is not possible, use the common grounding as presented in B) of the figure below.



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

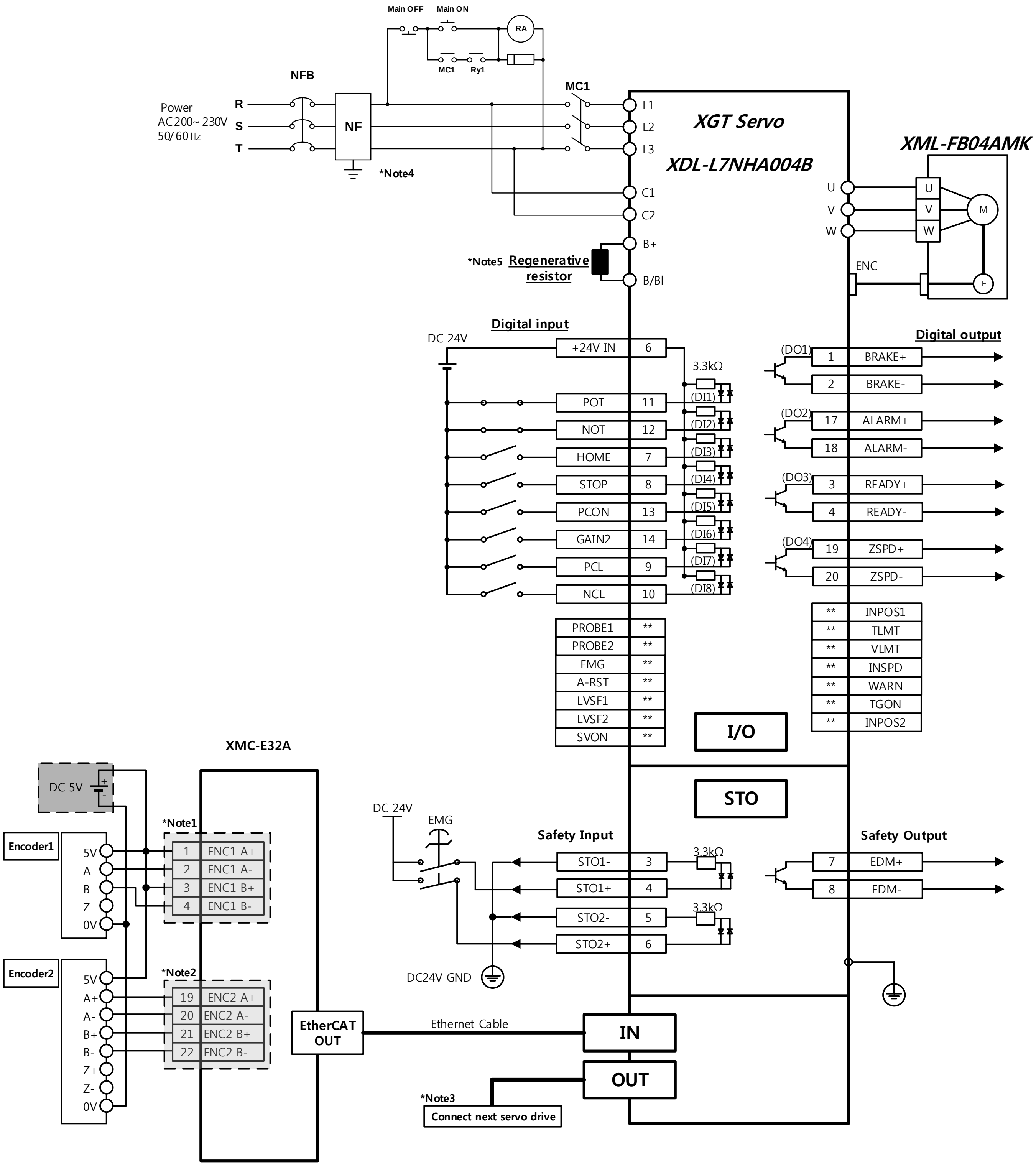
3.3.5 Specifications of Wiring Cable

The specifications of cable used for wiring are as follows.

Types of external connection	Cable specification (mm^2)	
	Lower limit	Upper limit
Digital input	0.18 (AWG24)	1.5 (AWG16)
Digital output	0.18 (AWG24)	1.5 (AWG16)
Analogue I/O	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)

3.3.6 Connection Example of Servo Drive

(1) This is an example of wiring which connects EtherCAT servo drive/motor, XDL-L7NH Model of XGT Servo, in motion cotroller. Refer to manual of each drive for details on installation and wiring.



Note

*Note1

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

*Note2

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

*Note3

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. And connection order is not related with axis order.

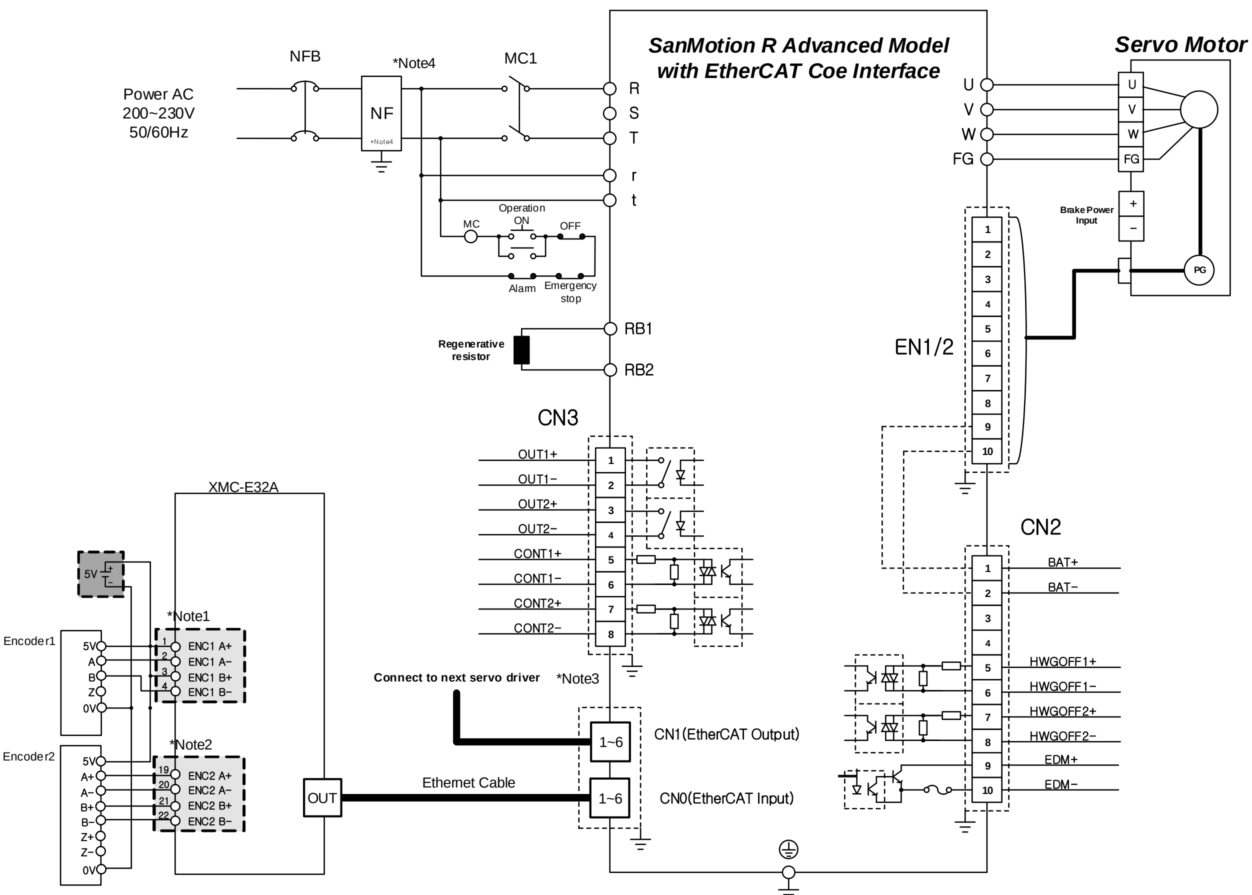
*Note4

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.



Note

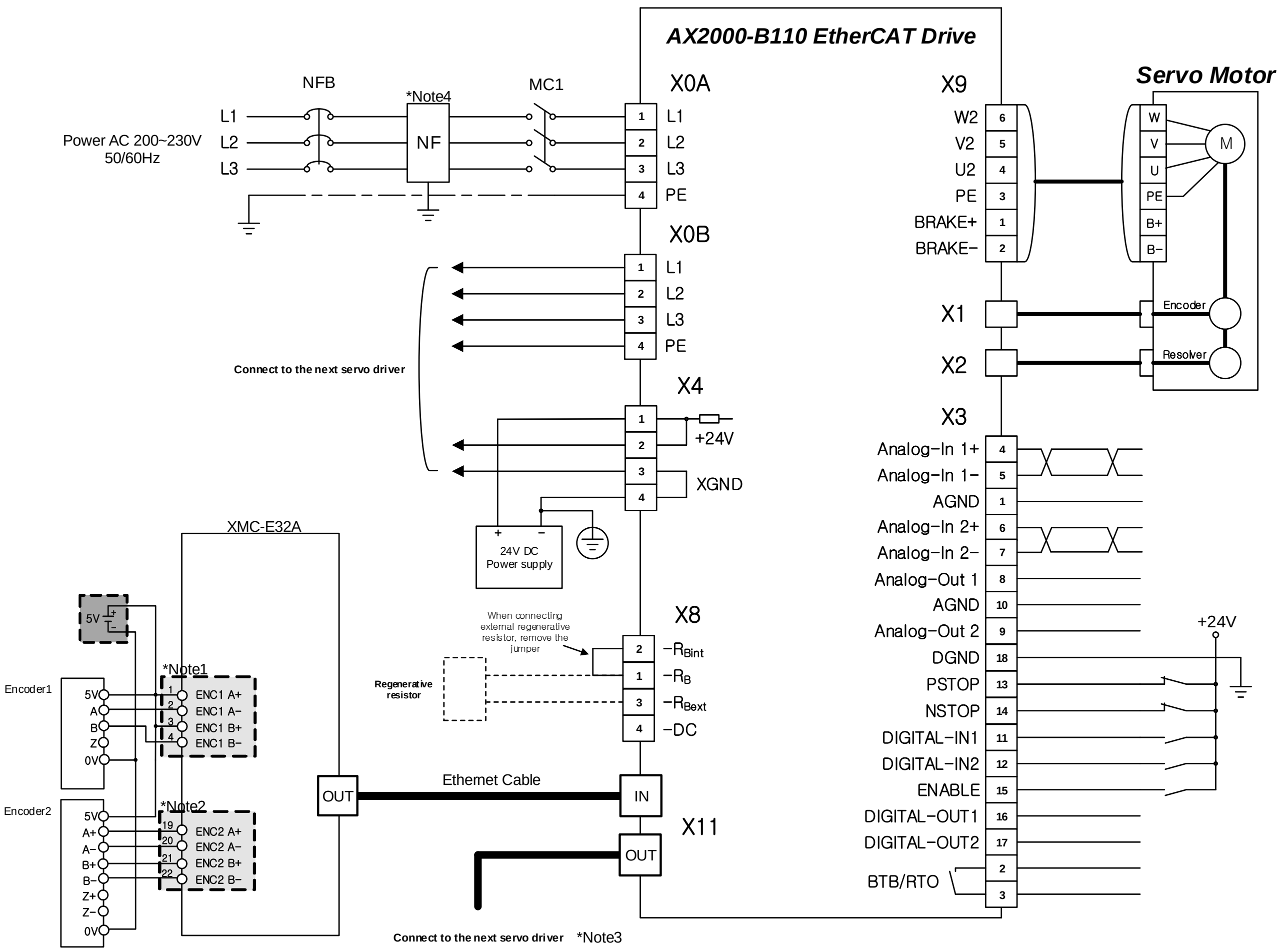
*Note1
Wiring of encoder 1 is an example about 5V voltage output (open collector) type.

*Note2
Wiring of encoder 2 is an example about 5V voltage output (line driver) type.

*Note3
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. And connection order is not related with axis order.

*Note4
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.



Note

*Note1
Wiring of encoder 1 is an example about 5V voltage output (open collector) type.

*Note2
Wiring of encoder 2 is an example about 5V voltage output (line driver) type.

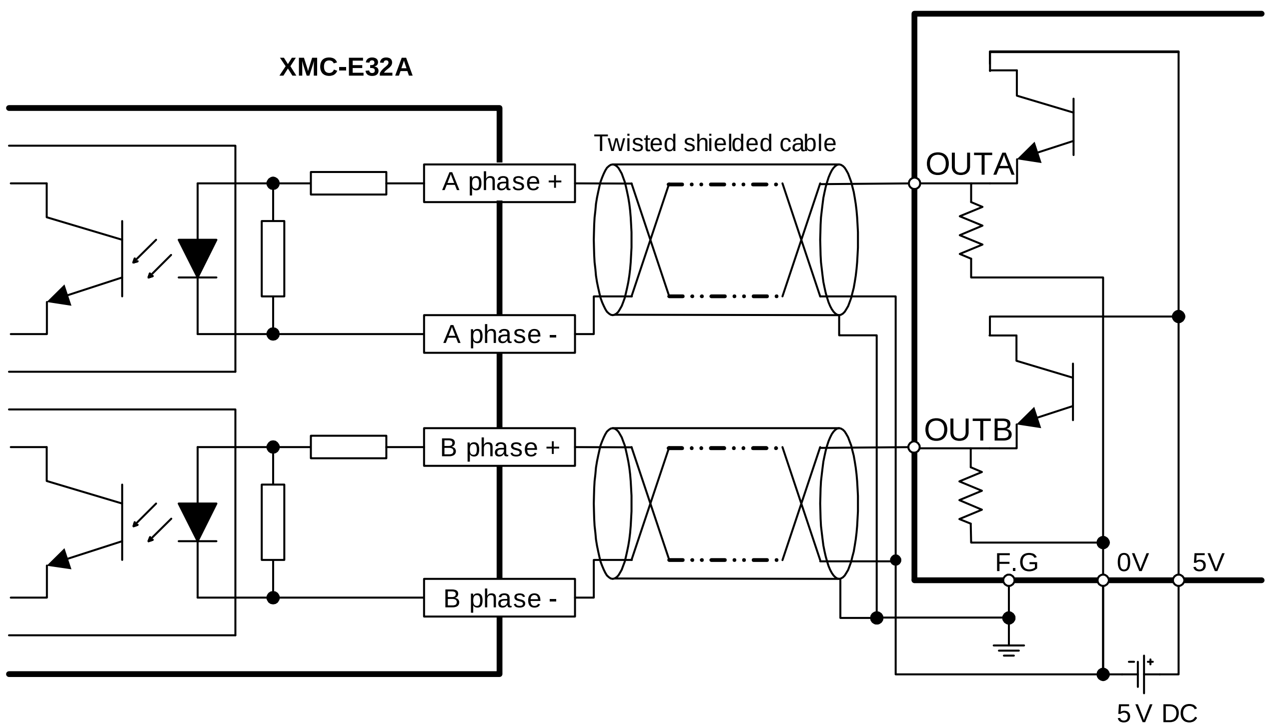
*Note3
When connecting more than 2 servo drivers, connect first servo driver’s IN to the positioning module’s OUT and for other servo drivers, connect previous servo driver’s OUT to next servo driver’s IN. Last servo driver’s OUT doesn’t need to be connected. And connection order is not related with axis order.

*Note4
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

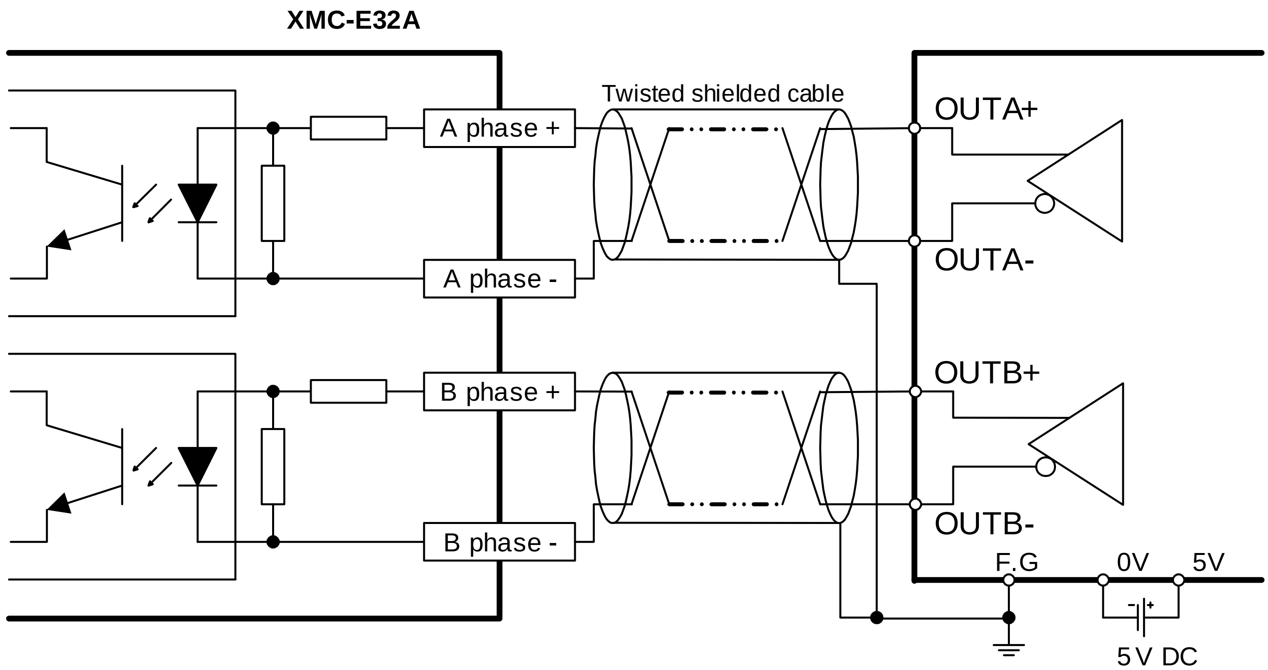
When pulseulse generator is a voltage output type, wiring example of motion controller and encoder input part is as follows.

In case pulse generator is totem-pole output and used as voltage output style, wiring is equal.



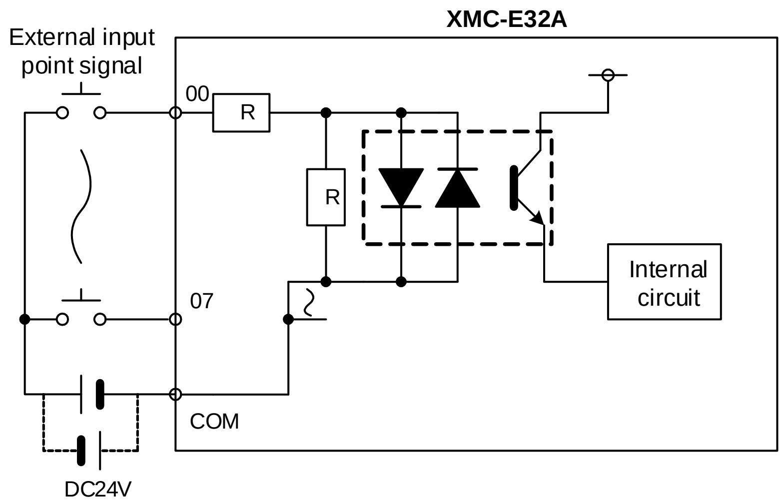
Note
Before Wiring, please consider maximum output distance of pulse generator.

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

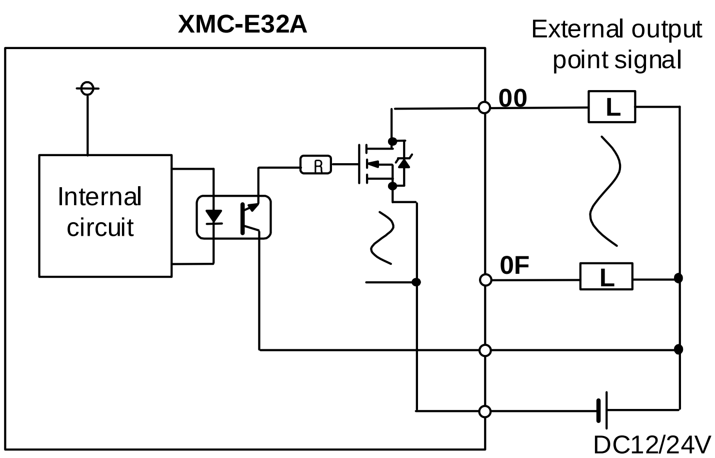


Note
Before Wiring, please consider maximum output distance of pulse generator.

3.3.9 External Input Signal Wiring Example



3.3.10 External Output Signal Wiring Example



3.4 EMC

3.4.1 EMC Standard

(1) Reauirements for comformance to EMC directive

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 controller. 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 the 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.

Specification	Test item	Test details	Standard value
EN50081-2	EN55011 Radiated noise ＊2	Electromagnetic emissions from the product are measured	30~230 ㎒ QP : 50 ㏈㎶/m ＊1 230~1000 ㎒ QP : 57 ㏈㎶/m
	EN55011 Conducted noise	Electromagnetic emissions from the product to the power line is measured	150~500 ㎑ QP : 79 ㏈ Mean: 66 ㏈ 500~230 ㎒ QP : 73 ㏈ Mean: 60 ㏈
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. The tests for the corresponding items were performed while the motio controller was installed inside a 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),

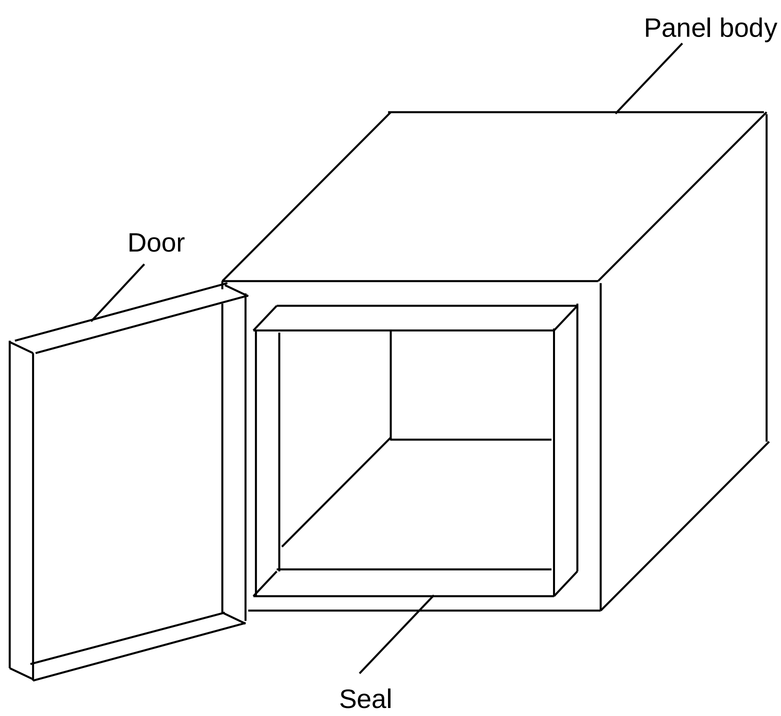
The specifications for the control panel are as follows:

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 do not leak out.

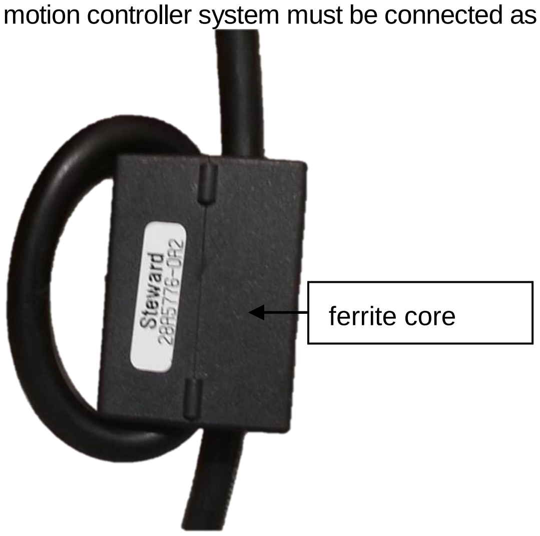
For example, make the door as a box-structure so that the panel body and the door are overlapped each other. This structure reduces the surge voltage generate by motion controller.



(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

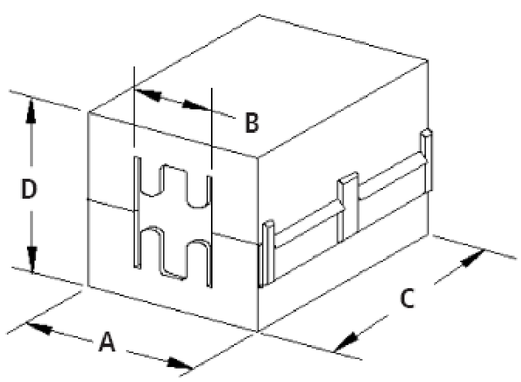
Earthing and power supply wires for the motion controller system must be connected as described below.



(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 FG (Frame Ground) terminals is to pass the noise generated in the motion controller system to the ground, so 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.

[Ferrite core]

Manufacture	Name	External Dimension (mm)				Maximum cable diameter(mm)	Address
		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 eh 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 the details conforms to the above regulation. 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.

1) Standard applied for motion contorller

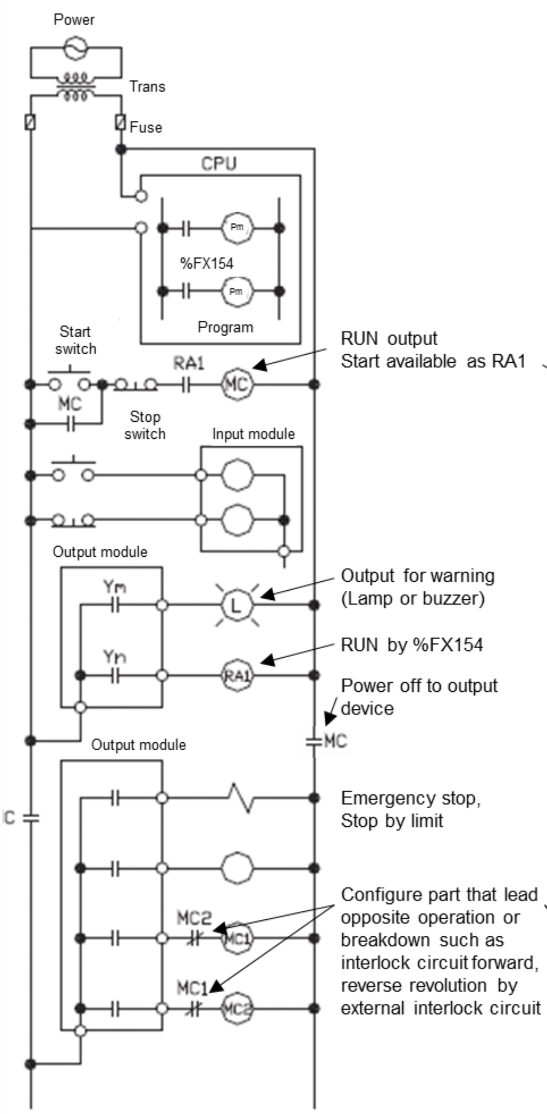
The motion controller follow EN6100-1 (safety of devices used in measurement rooms, control rooms, or laboratories). And the motion controller modules which operate at the rated voltage of AC50V/DC75V or above are also developed to conform the above standard.

3.5 Fail Safe

3.5.1 Fail Safe Circuit

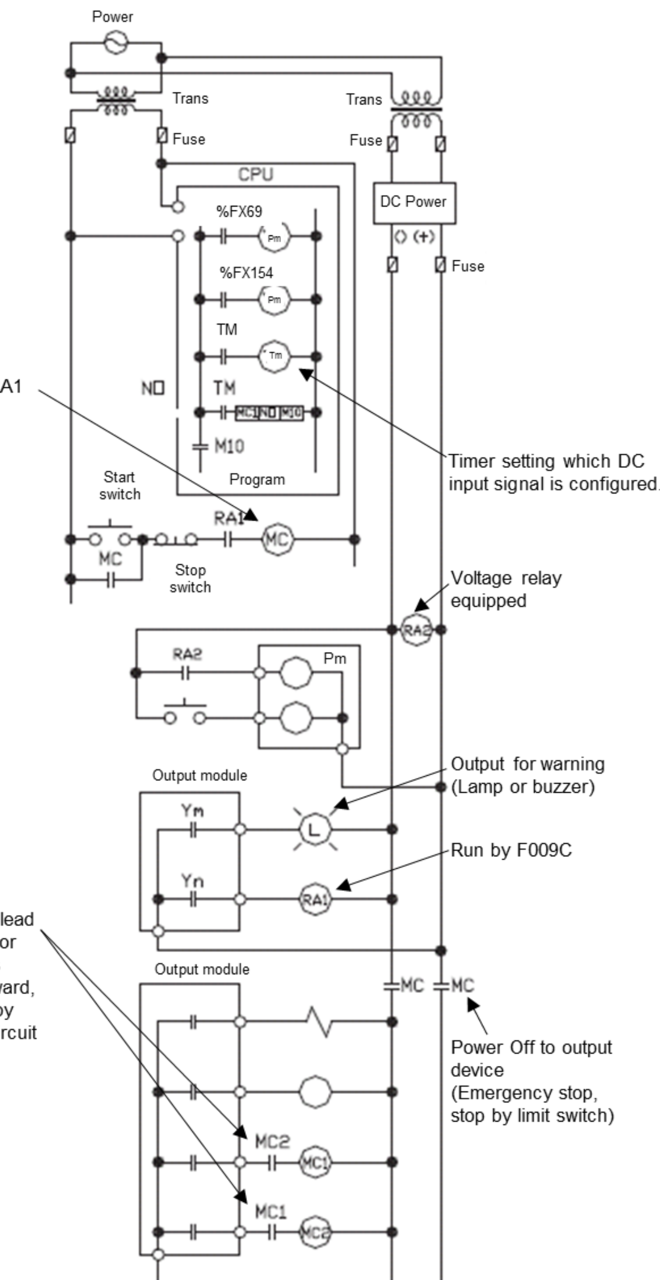
(1) Example of system design

In case of AC



Start sequence of power. In case of AC
(1) Turn on power
(2) Run CPU.
(3) Turn on start switch
(4) Output device runs by program through magnetic contactor (MC) [On]

In case of AC, DC



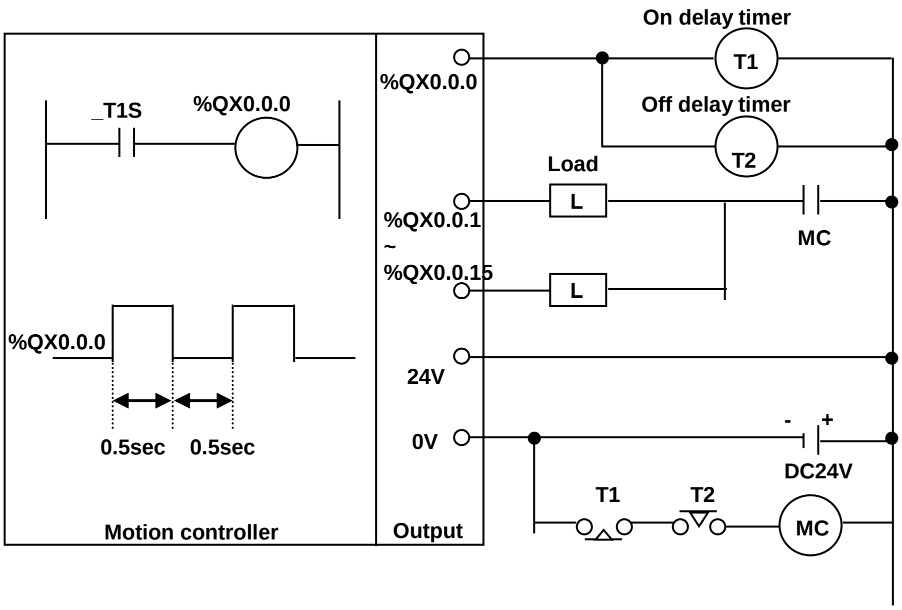
Start sequence of power. In case of AD, DC
(1) Run CPU after power is on
(2) Turn on RA2 as DC power on
(3) Turn on timer after DC power is stable.
(4) Turn on start switch
(5) Output device runs by program through magnetic contactor (MC) [On]

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

Failures of the motio 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 faile safe 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
Change rate of input voltage		Within change rate of input voltage	d 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.
Ambient environment	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.
Play of modules		No play allowed	Securely enrage the hook.
Connecting conditions of terminal screws		No loose allowed	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	Corrective Actions
Connection conditions of base		Check the screws.	Screws should not be loose.	Retighten Screws.
Connecting conditions of terminal block or extension cable		Check for loose mounting screws.	Screws should not be loose.	Retighten Screws.
		Check the distance between solderless terminals.	Proper clearance should be provided.	Correct.
		Connecting of expansion cable.	Connector should not be loose.	Correct.
LED indicator	PWR LED	Check that the LED is On.	On (Off indicates an error)	
	Run LED	Check that the LED is On during Run.	On (flickering or On indicates an error)	
	ERR LED	Check that the LED is Off during Run.	Flickering indicates an error	
	Input LED	Check that the LED turns On and Off.	On when input is On, Off when input is off.	
	Output LED	Check that the 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.

Check Items		Checking Methods	Judgment	Corrective Actions
Ambient environment	Ambient temperature	- Measure with thermometer and hygrometer - measure corrosive gas	0 ~ 55 °C	Adjust to general standard (Internal environmental standard of control section)
	Ambient Humidity		5 ~ 95%RH	
	Ambient pollution level		There should be no corrosive gases	
Controller conditions	Looseness, Ingress	The module should be move the unit	The module should be mounted securely	Retighten screws or hook
	Dust or foreign material	Visual check	No dust or foreign material	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.
Connecting conditions	Loose terminal screws	Re-tighten screws	Screws should not be loose	Retighten
	Distance between terminals	Visual check	Proper clearance	Correct
	Loose connectors	Visual check	Connectors should not be loose.	Retighten connector mounting screws
Line voltage check		Measure voltage between input terminals	3.3 Power specifications	Change supply power

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 checks
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) Trouble 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) Narrow down the possible causes of the trouble where the fault lies, i.e.:
- Inside or outside of the motion controller ?
- I/O module or another module ?
- Motion program ?

3.7.2 Troubleshooting

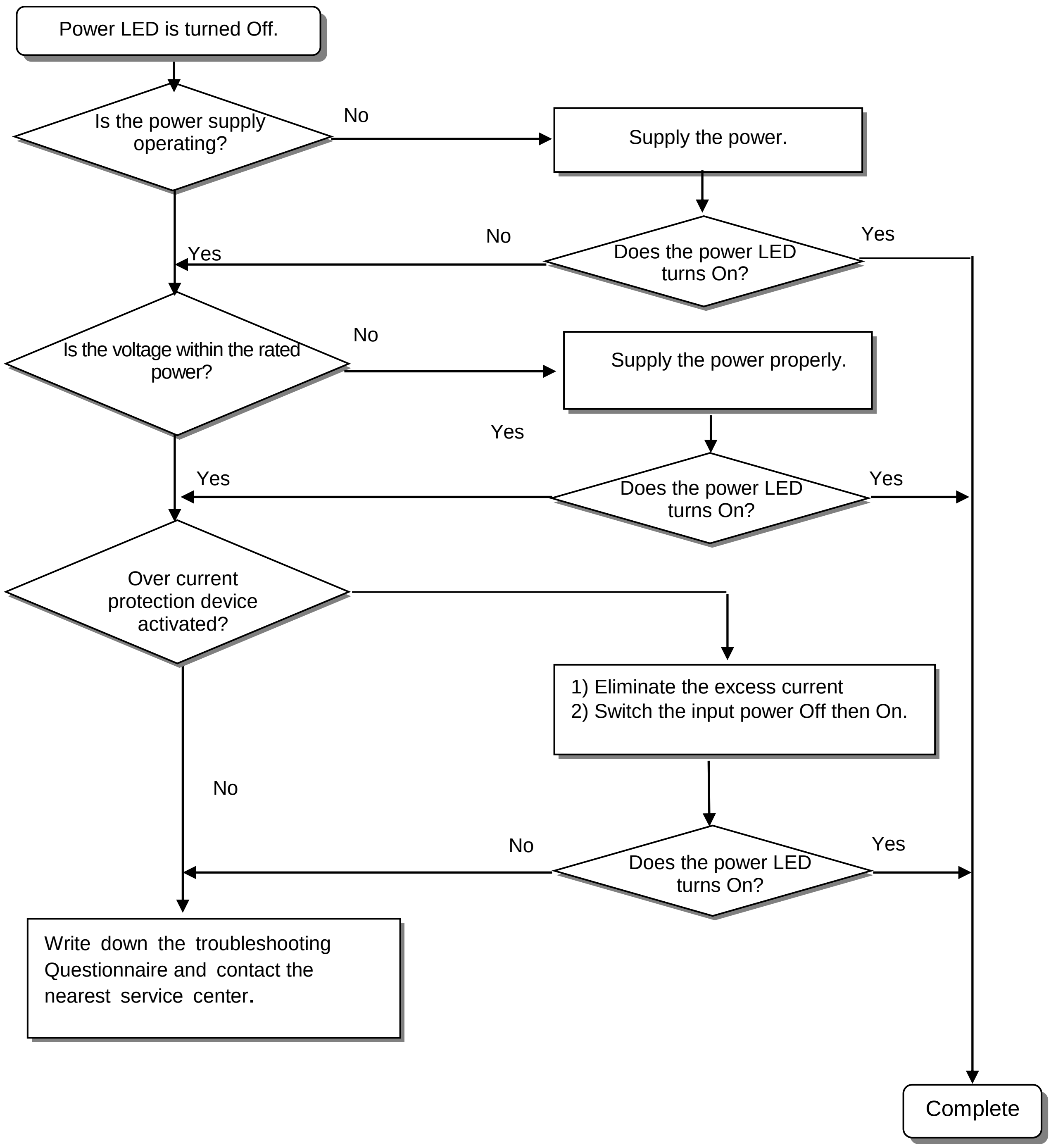
This section explains the procedure for determining the cause of troubles as well as the errors and corrective actions.

Symptoms

Is the power LED turned	⇒ Flowchart used when the POWER LED is turned Off.
Is the ERR LED flickering?	⇒ Flowchart used when the ERR LED is flickering.
Are the RUN LED turn ed	⇒ Flowchart used when the RUN turned Off.
I/O module doesn’t operate properly.	⇒ Flowchart used when the output load of the output module
Program cannot be written.	⇒ Flowchart used when a program can’t be written to the motion controller.

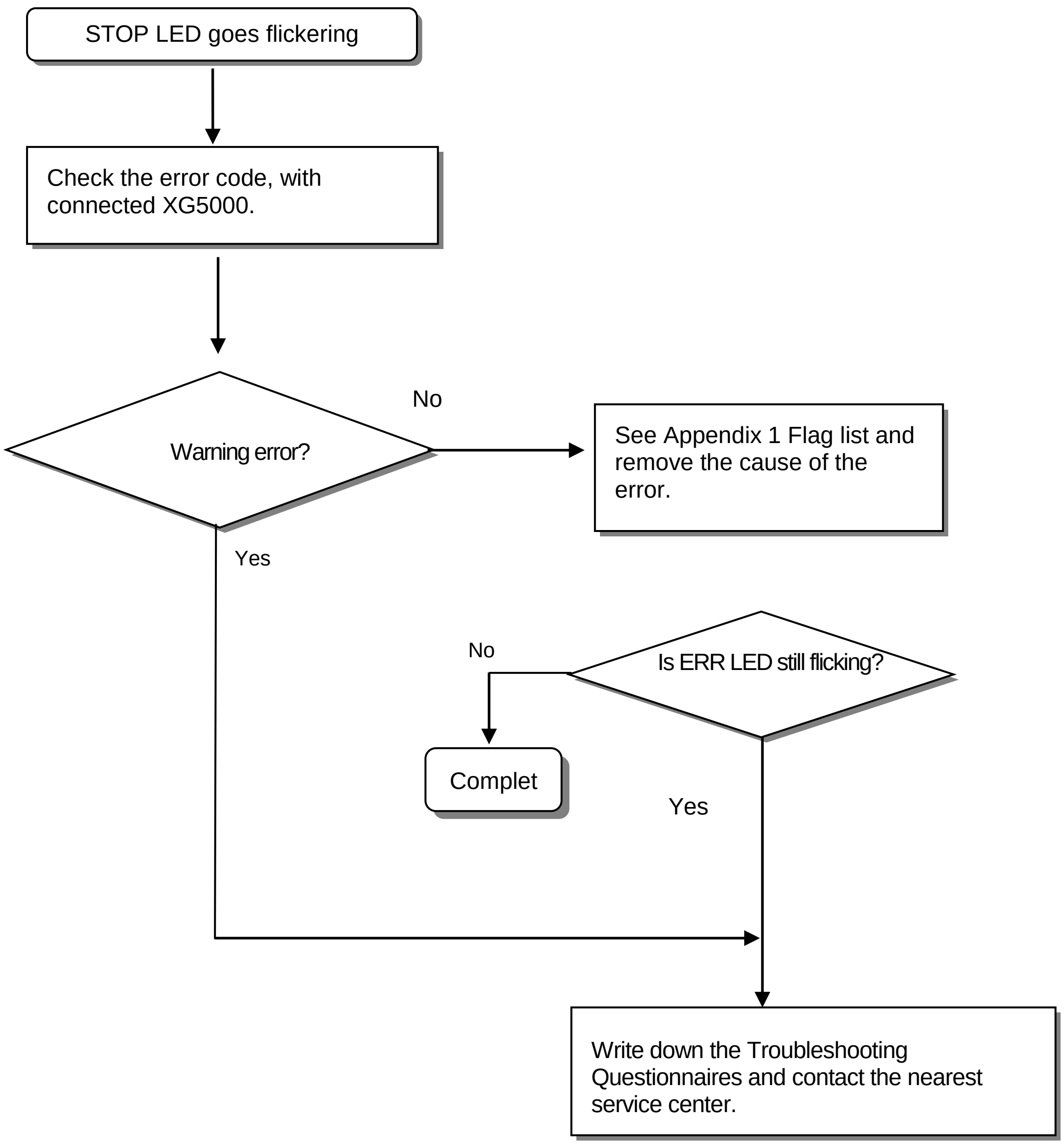
(1) Troubleshooting flowchart used when the PWR (Power) LED turns Off

The following flowchart explains corrective action procedure used when the power is supplied or the power LED turns Off during operation.



(2) Troubleshooting flowchart used with when the ERR (Error) LED is flickering

The following flowchart explains corrective action procedure used when the power is supplied starts or the ERR LED is flickering during operation.The following flowchart explains corrective action procedure used when the power is supplied or the power LED turns

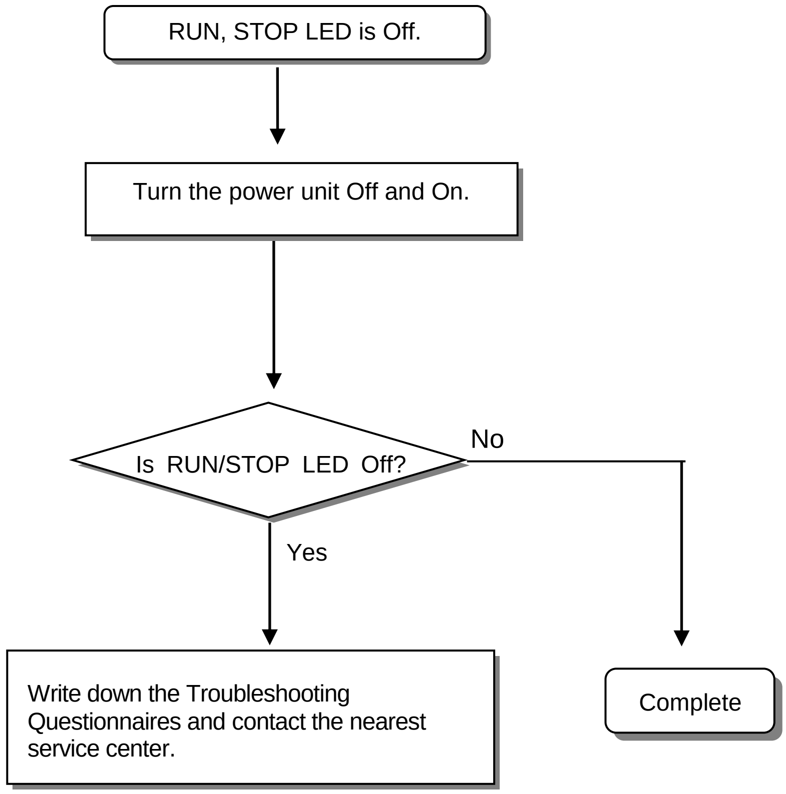


Note

Though warning error appears, motion controller system doesn't stop but corrective action is needed promptly. If not, it may cause the system failure.

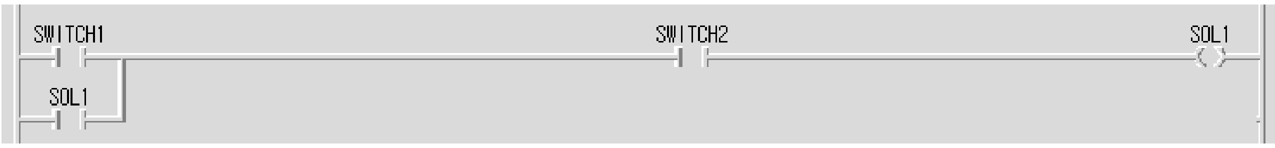
(3) Troubleshooting flowchart used with when the RUN, STOP LED turns Off.

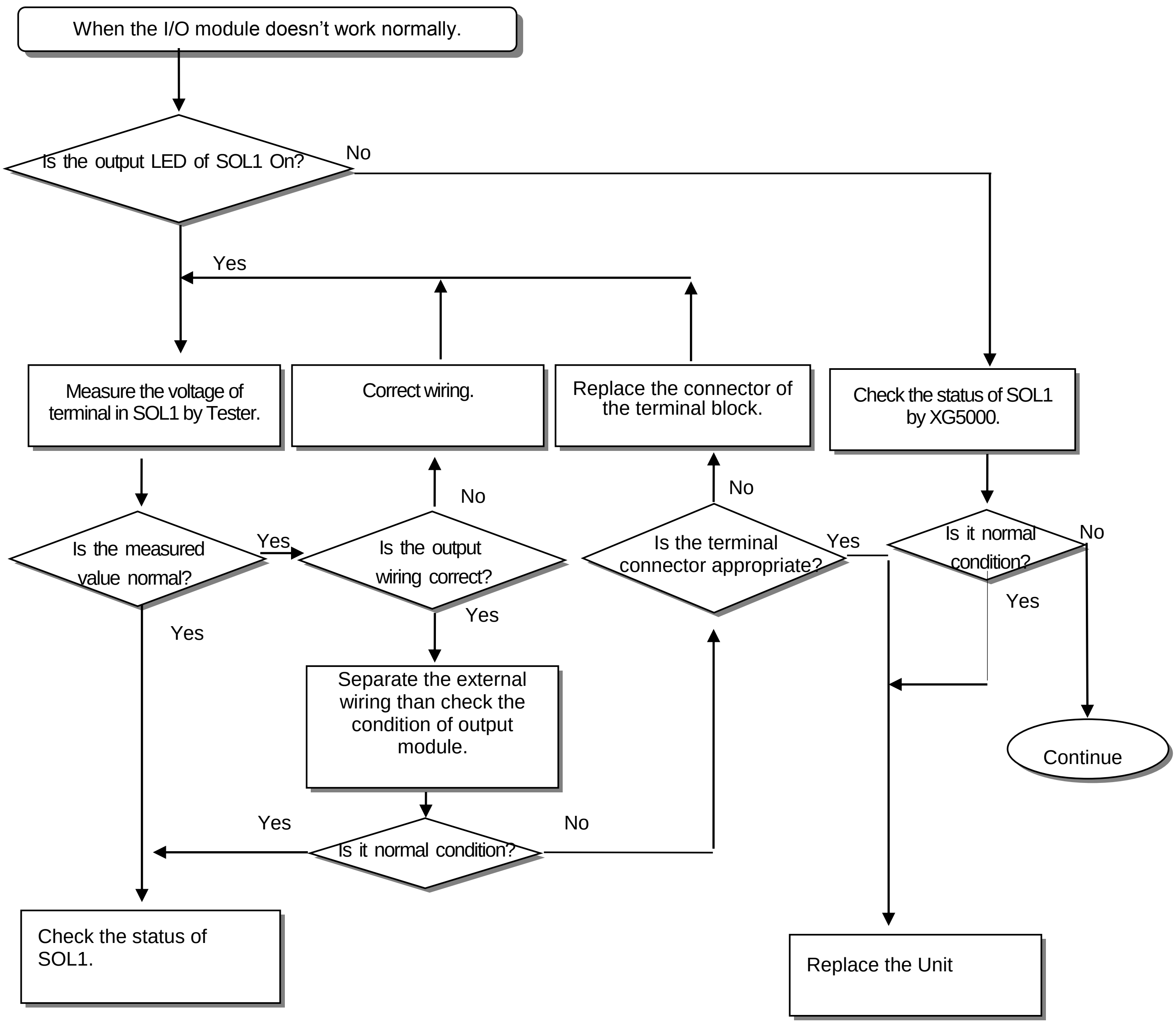
The following flowchart explains corrective action procedure to treat the lights-out of RUN LED when the power is supplied, operation starts or is in the process.

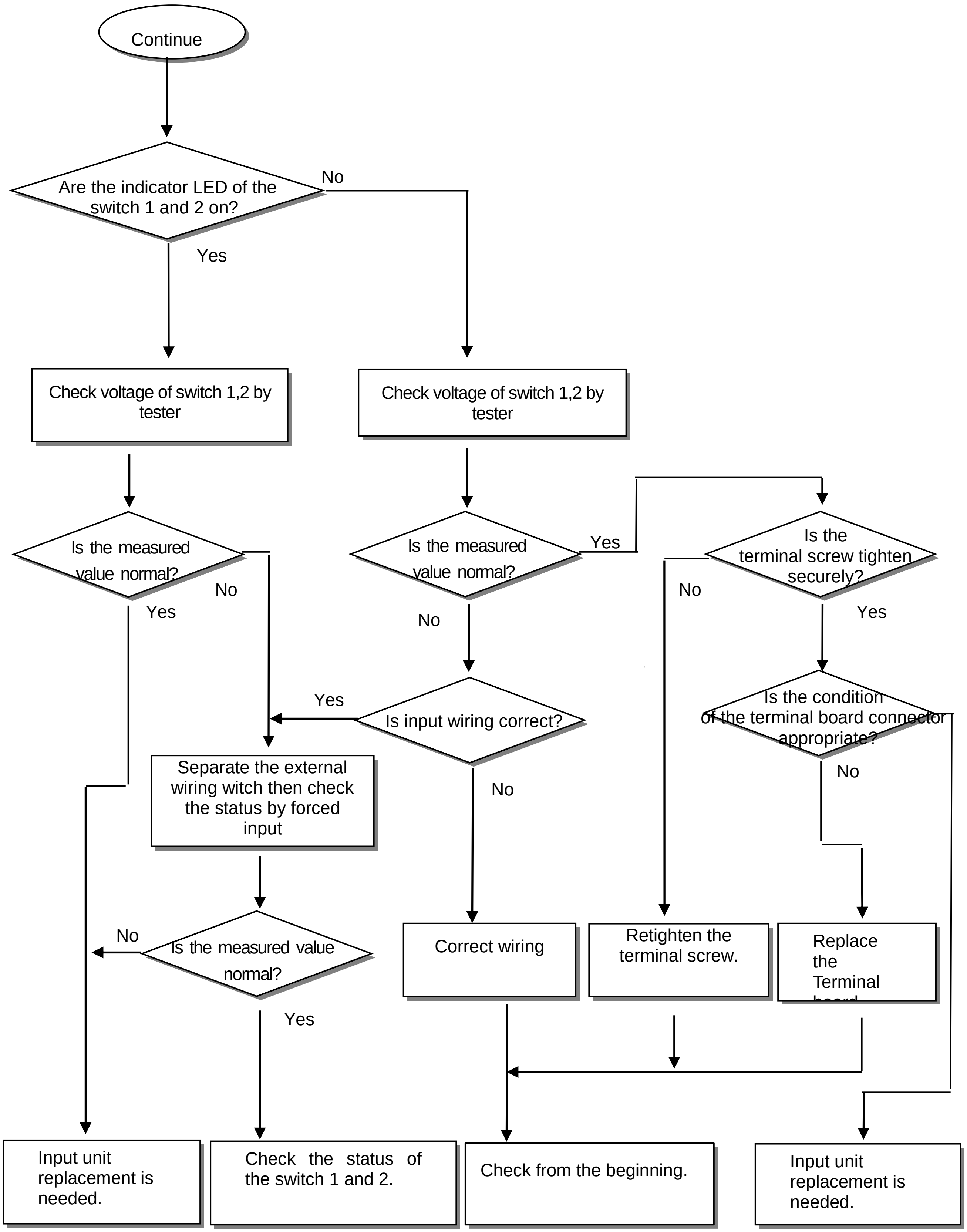


(4) Troubleshooting flowchart used when the I/O part doesn't operate normally

The following flowchart explains corrective action procedure used when the I/O module doesn't operate normally.






Continue
Are the indicator LED of the switch 1 and 2 on?
No
Yes
Check voltage of switch 1,2 by tester
Check voltage of switch 1,2 by tester
Is the measured value normal?
No
Yes
Is the measured value normal?
Yes
No
Is the terminal screw tighten securely?
No
Yes
Yes
Is input wiring correct?
Is the condition of the terminal board connector appropriate?
No
No
Separate the external wiring witch then check the status by forced input
No
Is the measured value normal?
Yes
Correct wiring
Retighten the terminal screw.
Replace the Terminal
Input unit replacement is needed.
Check the status of the switch 1 and 2.
Check from the beginning.
Input unit replacement is needed.

3.7.3 Troubleshooting Questionnaire

If any problem occurs during the operation of motin controller, please write down this Questionnaire and contact the service center via telephone or facsimile.

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

1. Telephone & FAX No
Tell) FAX)

2. Using equipment model:

3. Details of using equipment
Controller model ( ) OS version No. ( ) Serial No. ( )
XG5000 (for program compile) version No. ( )

4. General description of the device or system used as the control object:

5. The kind of the motion controller
– Operation by the mode setting switch ( )
– Operation by the XG5000 or communications ( )

6. Is the ERR. LED of the motion controller turned On ? Yes ( ), No ( )

7. XG5000 error message:

8. History of corrective actions for the error message in the article 7:

9. Other tried corrective actions:

10. Characteristics of the error
• Repetitive ( ): Periodic ( ), Related to a particular sequence ( ), Related to environment ( )
• Sometimes ( ): General error interval:

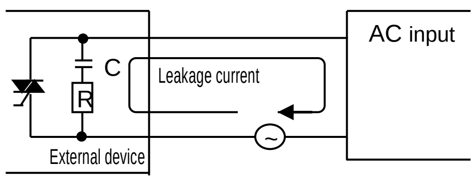
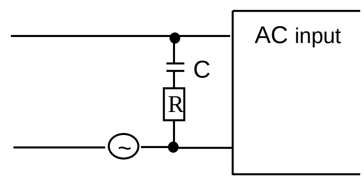
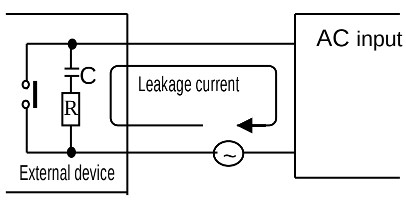
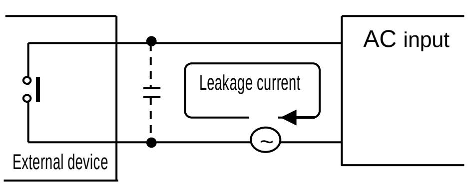
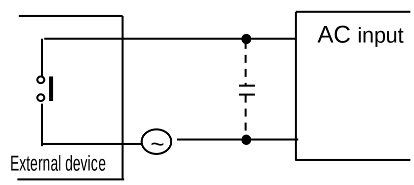
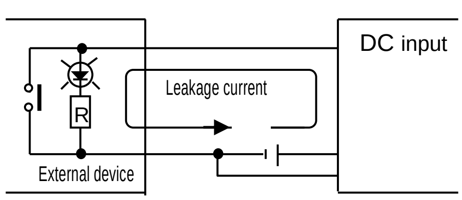
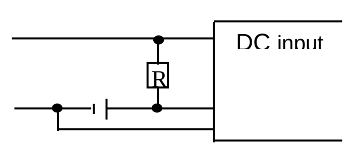
11. Detailed Description of error contents:

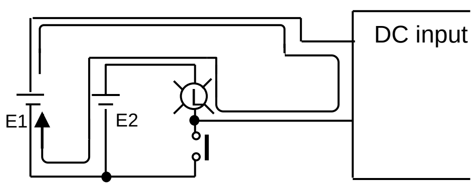
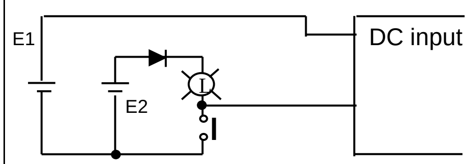
12. Configuration diagram for the applied system:

3.7.4 Troubleshooting Example

Possible troubles with various circuits and their corrective actions are explained.

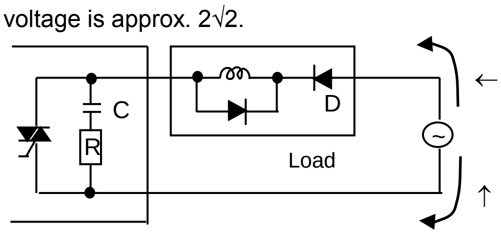
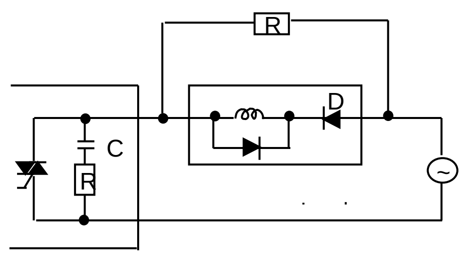
(1) Input circuit troubles and corrective actions
The followings describe possible troubles with input circuits, as well as corrective actions.

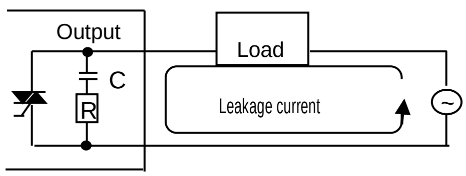
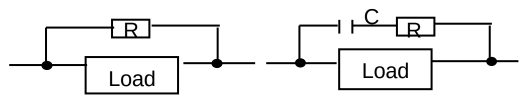
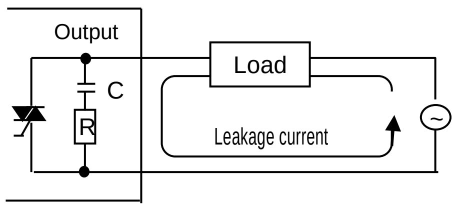
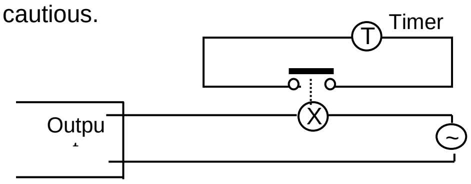
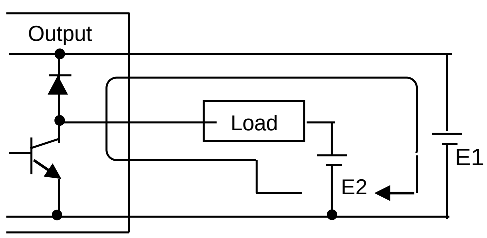
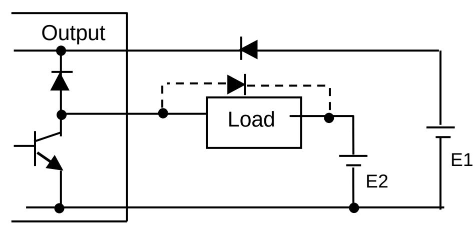
Condition	Cause	Corrective Actions
Input signal doesn't turn off.	Leakage current of external device (Such as a drive by non-contact switch)  	• Connect an appropriate register and capacity, which will make the voltage lower across the terminals of the input module.  • CR values are determined by the 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 (Neon lamp may be still on)	Leakage current of external device (Drive by a limit switch with neon lamp)  	
Input signal doesn't turn off.	Leakage current due to line capacity of wiring cable.  	• Locate the power supply on the external device side as shown 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. 

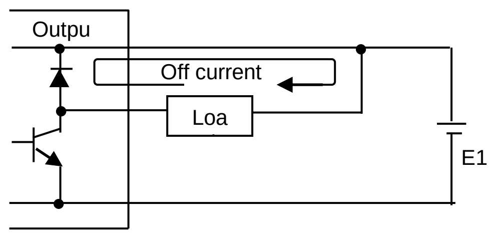
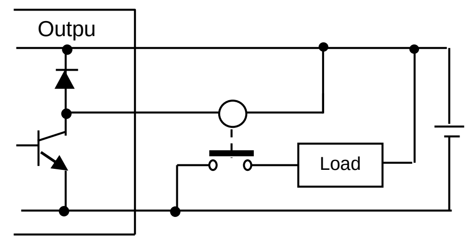
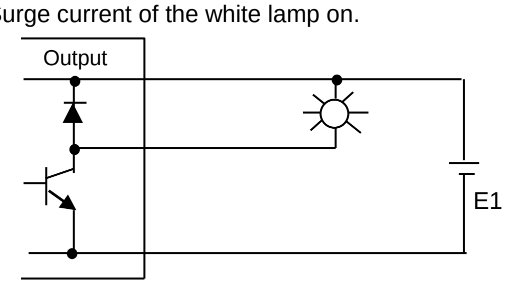
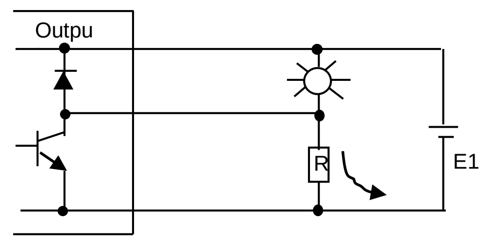
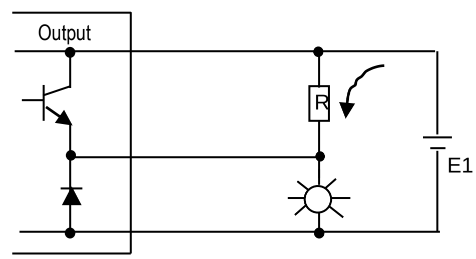
Condition	Cause	Corrective Actions
Input signal doesn't turn off	• Sneak current due to the use of two different power supplies.  • E1 > E2, sneaked.	• Use only one power supply. • Connect a sneak current prevention diode. 

3.7.5 Output Circuit and Corrective Actions

The following describes possible troubles with output circuits, as well as their corrective actions.

Condition	Cause	Corrective Action
When the output is off, excessive voltage is applied to the load.	•Load is half-wave rectified inside (in some cases, it is true of a solenoid) •When the polarity of the power supply is as shown in ①, C is charged. When the polarity is as shown in ②, the voltage charged in C plus the line voltage are applied across D. Max. voltage is approx. $2\sqrt{2}$.  *) If a resistor is used in this way, it does not pose a problem to the output element. But it may make the performance of the diode (D), which is built in the load, drop to cause problems.	• Connect registers of tens to hundreds KΩ across the load in parallel. 

Condition	Cause	Corrective Action
The load doesn't turn off	• Leakage current by surge absorbing circuit, which is connected to output element in parallel.  	• 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. 
When the load is C-R type timer, time constant fluctuates	• Leakage current by surge absorbing circuit, which is connected to output element in parallel.  	• Drive the relay using a contact and drive the C-R type timer using the since contact. • Use other timer than the C–R contact some timers have half-ware rectified internal circuits therefore, be cautious. 
The load does not turn off	• Sneak current due to the use of two different power supplies.  E1<E2, sneaks. E1 is off (E2 is on), sneaks.	• 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.

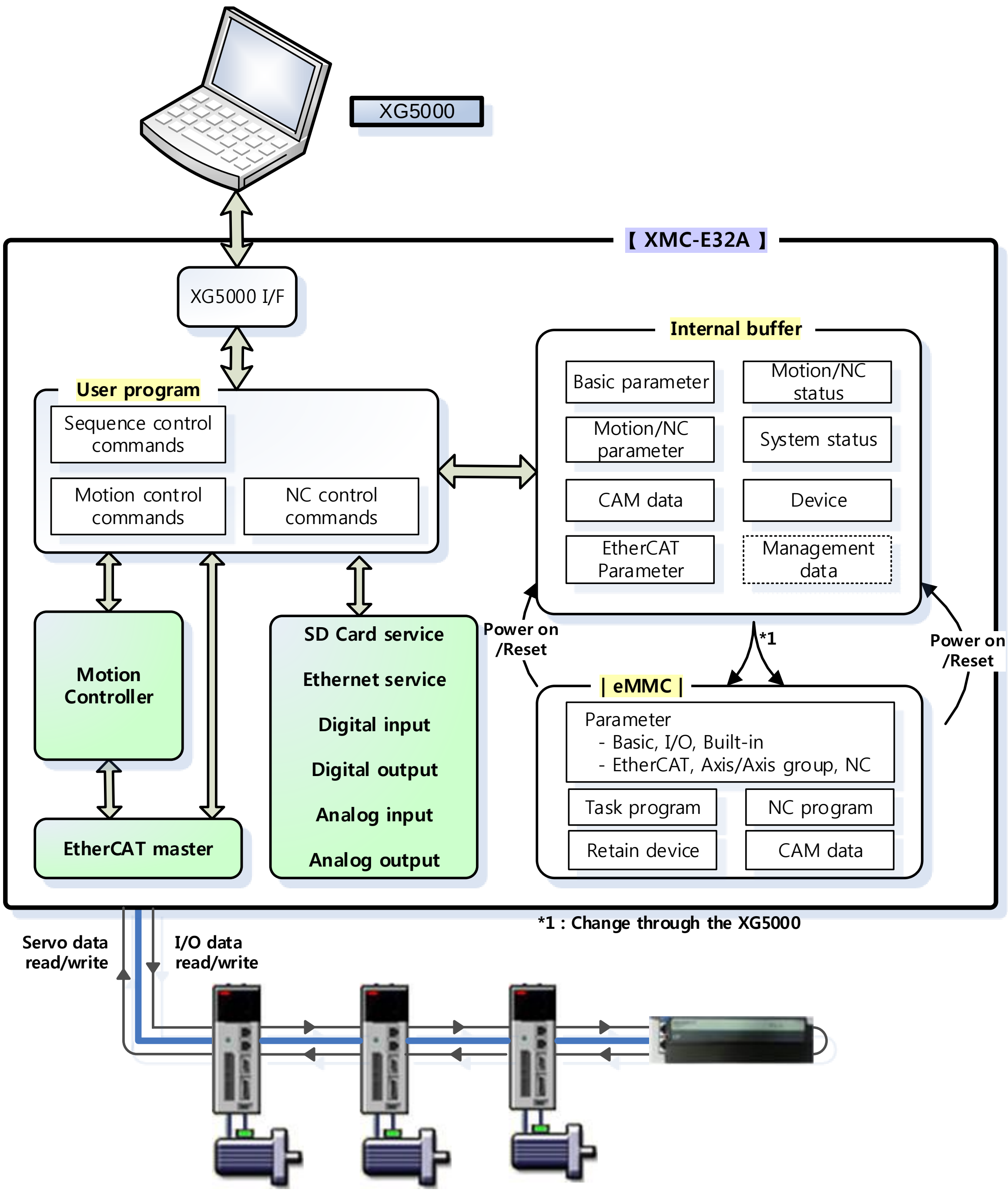
Condition	Cause	Corrective actions
The load off response time is long	• Over current at off state [The large solenoid current fluidic load (L/R is large) such as is directly driven with the transistor output.  • The off response time can be delayed by one or more second as some loads make the current 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.

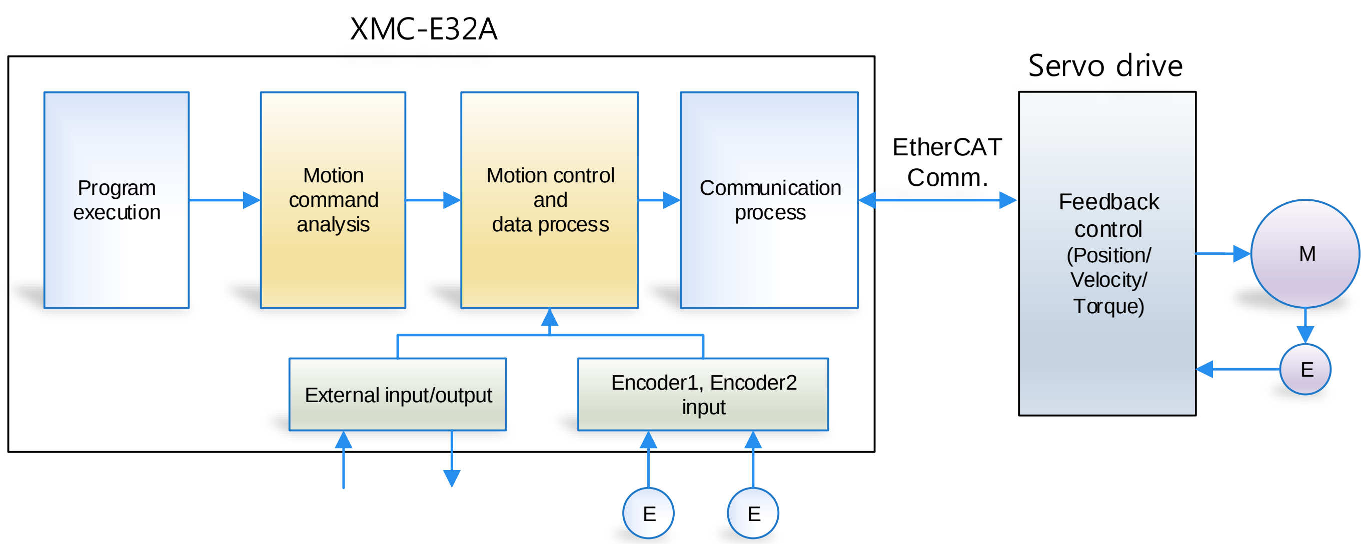
4.1 Structure of Motion Controller

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



4.2 Configuration of Motion Control

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. In addition to the built-in 8-point input and 16-point output, up to 64 EtherCAT I/O(including the axes) can be controlled. 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 Types of Tasks

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 1/2/4ms, and it can be set in the basic parameter of the motion controller.

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.

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.

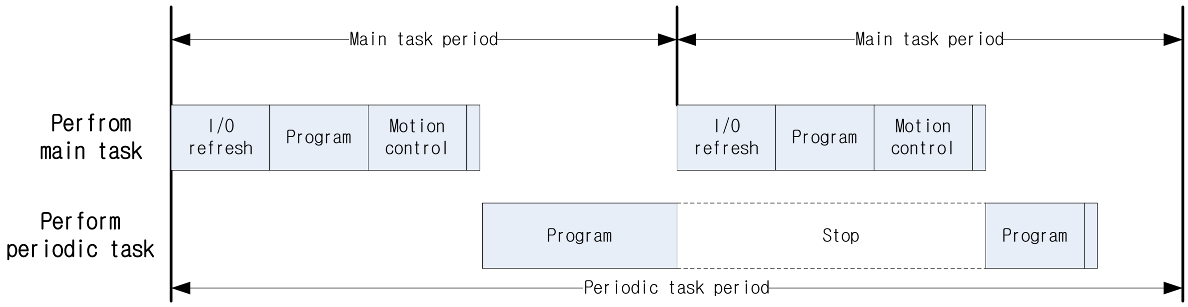
Note

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.

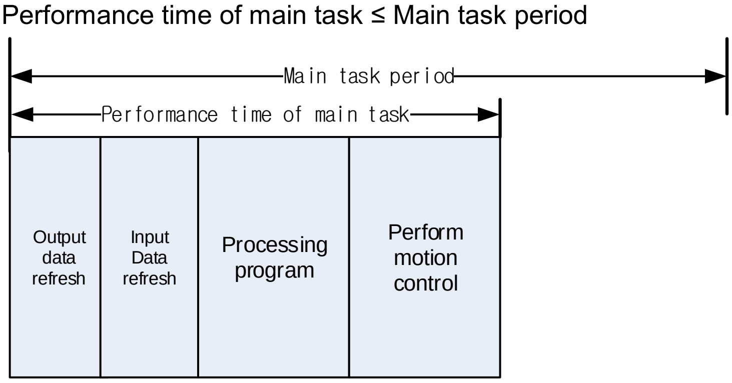


2. Main task operation

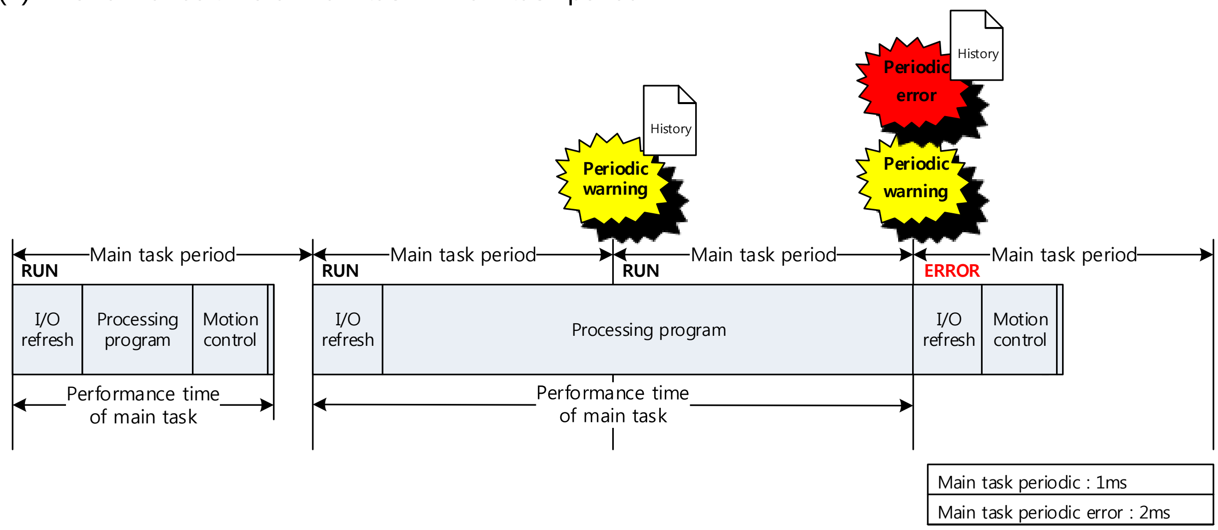
The main task must be performed in the set task period, and if the performance of the main task exceeds the set main task period, an error occurs and if motion controller is in RUN state, it is changed to STOP state.

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

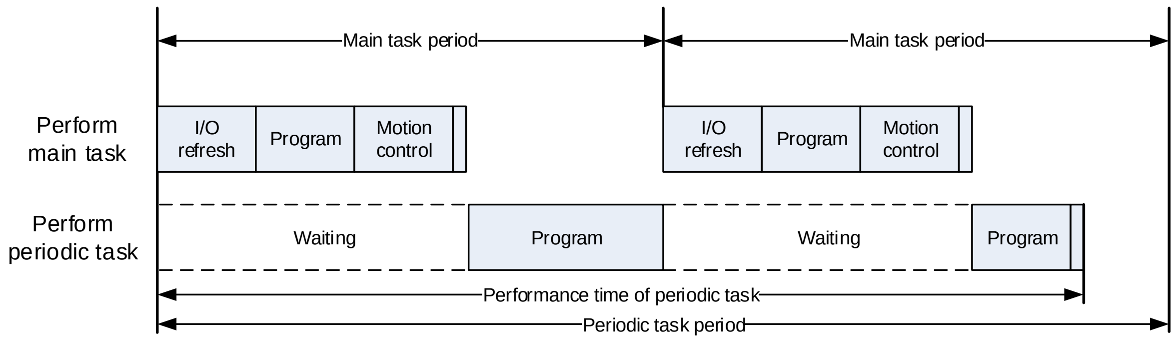


3. Periodic task operation

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

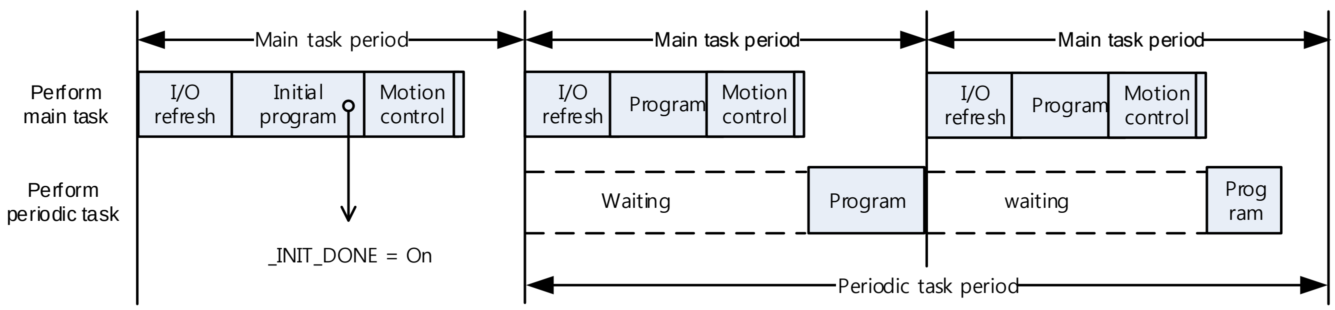


4. Initialization task operation

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 must be also performed in the set task period like the main task, and an error will occur if the performance of the initialization task exceeds the set period of the main task, and it is changed to stop state.

When using the basic function block and motion function block in the initialization task program, the function of the relevant function block may be limited. This is because it is only performed once when it enters the RUN mode due to the characteristic of the initialization task, and in the case of function block, the output parameter is not updated. Therefore, when using the basic function block and motion function block in the initialization task program, the output of the relevant function block may be different to its real function, so please take caution when in use.

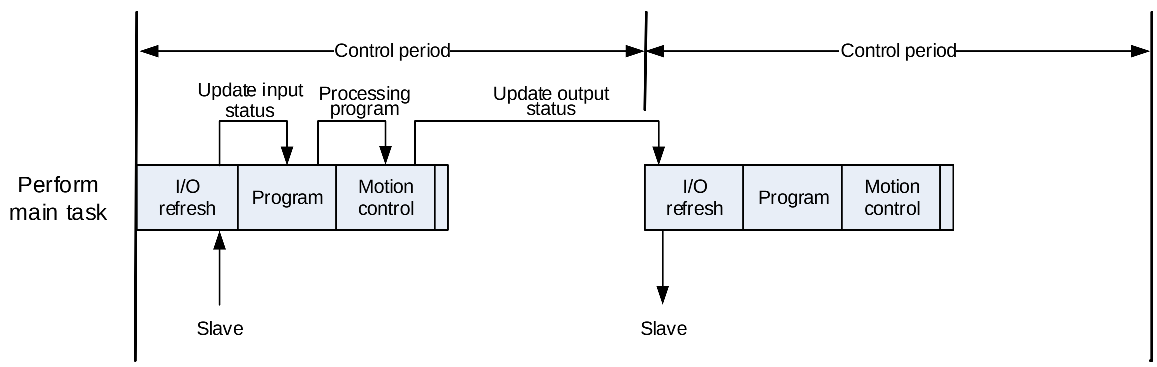
P



4.3.3 Execution of Motion Commands

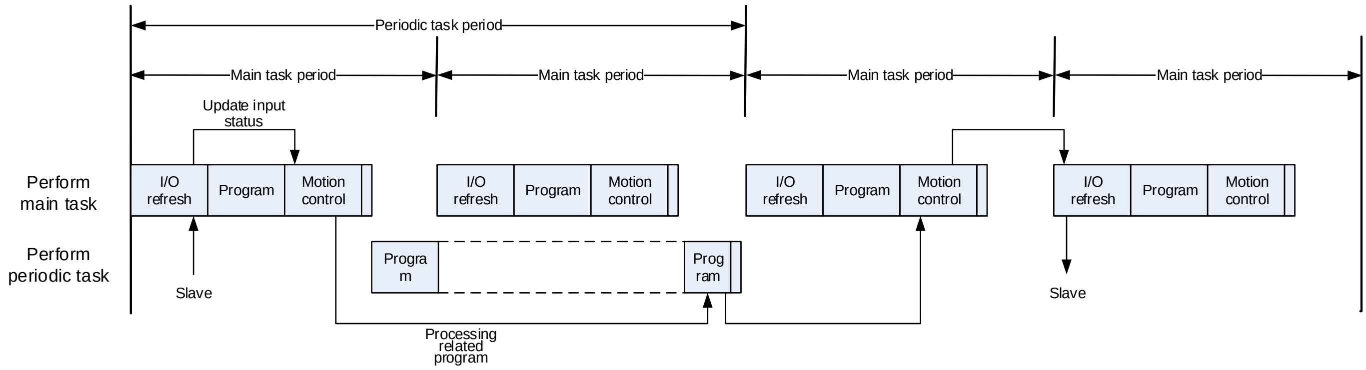
1. Execution of motion commands in the main task

Execution of motion instruction of the main task is shown in the figure below. The input value of the slave and the system parameters are updated by the I/O refresh motion of the main task, and based on this information, the program is processed and motion control motion is performed. 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 and motion control is performed in the main task based on this information. 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

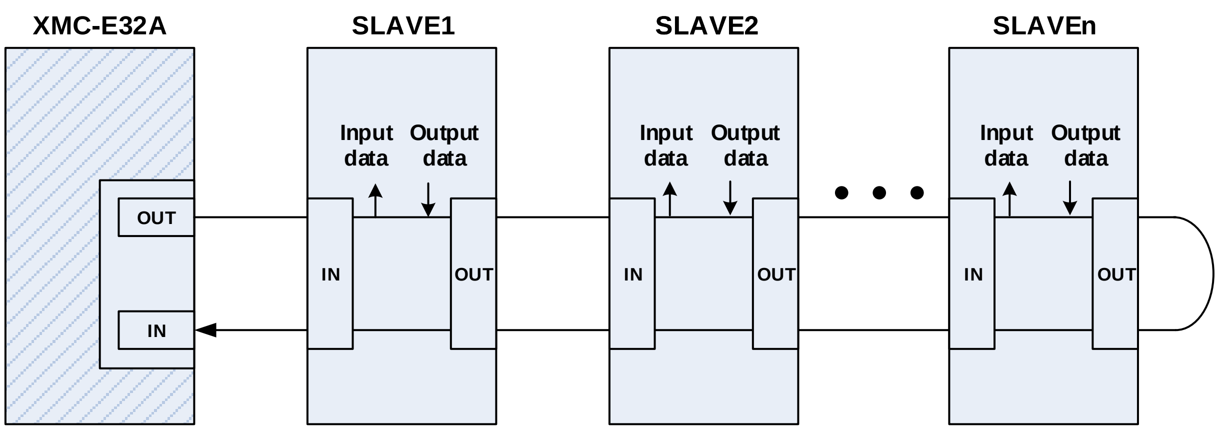


4.4 EtherCAT Communication

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 does not transmit data to each slave nodes of the network but passes one communication Frame to every slave in order, and each slave reads and writes Data in its relevant area in the Frame when the communication Frame passes through each slave. The communication Frame performs high speed data transmission with a structure where after going through the last slave, it turns back and passes through every slave and is transmitted to the master.



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.

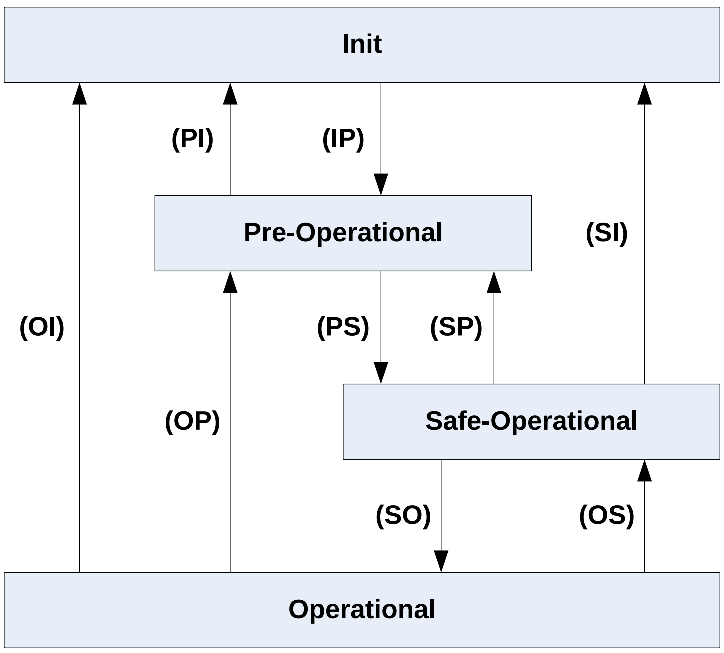
In CoE, parameter and data information of the slave are composed of Object Dictionary. Object Dictionary contains the information used in the configuration of the device and communication, and it is a group of the object (parameter) which can be accessed through the network. In the communication between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

Types of communication	Communication time	Contents
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). There are two types of process data: 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 Specification of Motion Controller EtherCAT Communication

Item	Specification
Communication protocol	EtherCAT
Support specification	CoE(CANopen over EtherCAT)
Physical layer	100BASE-TX
Communication speed	100Mbps
Topology	Daisy Chain
Communication cable	Over 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	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

4.5 Motion Control Program

4.5.1 Program Execution

1. Configuration of the program

Motion control program is composed of functional elements needed in performing certain controls and it is performed in the internal RAM of motion controller. The program is backed up in the flash memory.

Programs with these functional elements are classified as follows.

Program	Processing information
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'.
Initialization task program	Execute the command which is executed once in case of motion controller RUN.

Note

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 Modes

1. RUN mode

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

(1) Processing when changing the mode

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 calculation 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.

3. Change in operation modes

Operation mode of motion controller can be changed as follows.

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

Chapter 5 Memory and Parameter

5.1 Memory

5.1.1 Program and Data Memory

1. Memory for the program

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

Type	Size	Description
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. Memory for the data

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

Type		Size	Description
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 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
	Others	456KB	UDF/B Internal purposes and NC Local variables
History		88KB	User history (error / mode / system / power / motion error)
System operation-		43,008KB	Internal operating area of the system and other functions

5.1.2 Device

1. Types of devices

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

Type	Size	Description
Automatic variable (A)	4,096KB	Automatic variable area (able to set 2,408KB of retain)
Direct variable (G)	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 (U)	18KB	SD memory, data log and encoder flag area

(1) Automatic variable

(a) 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.

(b) The automatic variables that do not have Retain Settings are initialized to 0 when power is applied again or at Stop-to-Run.

(2) 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

(3) Input variable

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

(b) Built-in digital input is 8 points.

(c) Input variable is expressed as follows.

%I[size prefix]n

Number	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

Number	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 assigned. n3 data based on [size prefix]

Example) %IW64 = %IB128 = %IW1.0.0 = %IB1.0.0, %IW1 = %IB2 = %IW0.0.1 = %IB0.0.2

(d) 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

(4) Output variable

(a) This is a variable assigned to built-in digital output and RxPDO of EtherCAT slaves.

(b) Built-in digital output is 16 points.

(c) Input variable is expressed as follows.

%Q[size prefix]n

Number	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

Number	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 assigned. n3 data based on [size prefix]

Example) %QW64 = %QB128 = %QW1.0.0 = %QB1.0.0, %QW1 = %QB2 = %QW0.0.1 = %QB0.0.2

(d) 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

(5) Special variable

(a) This is a variable assigned to built-in analog input and output.

(b) Built-in analog input is 2 channels, and built-in analog output is 2 channels.

(c) Built-in analog variable is expressed as follows.

%I[size prefix]n1.n2.n3

Number	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

(d) 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	Channel 0 Active
_01_AD0_AVGTYPE	BYTE	%UB0.1.34	Channel 0 Average type
_01_AD0_AVGVAL	WORD	%UW0.1.18	Channel 0 Average value
_01_AD0_DATA	WORD	%UW0.1.5	Channel 0 Output data
_01_AD0_DATATYPE	BYTE	%UB0.1.26	Channel 0 Output data type setting
_01_AD0_ERR	BOOL	%UX0.1.32	Channel 0 Error

Variable	Type	Device	Description
_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 Alarm (Upper limit)
_01_AD0_IDD	BOOL	%UX0.1.72	Channel 0 Input disconnection flag
_01_AD0_LOOR	BOOL	%UX0.1.56	Channel 0 Alarm (Lower limit)
_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 setting
_01_AD1_DATA	WORD	%UW0.1.6	Channel 1 Output data
_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 Alarm (Upper limit)
_01_AD1_IDD	BOOL	%UX0.1.73	Channel 1 Input disconnection flag
_01_AD1_LOOR	BOOL	%UX0.1.57	Channel 1 Alarm (Lower limit)
_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 per channel (Array)
_01_AD_AVGTYPE_ARY	ARRAY[0..1] OF BYTE	%UB0.1.32	Average type per channel (Array)
_01_AD_AVGVAL_ARY	ARRAY[0..1] OF WORD	%UW0.1.18	Average value per channel (Array)
_01_AD_DATATYPE_ARY	ARRAY[0..1] OF BYTE	%UB0.1.26	Data type setting per channel (Array)
_01_AD_DATA_ARY	ARRAY[0..1] OF WORD	%UW0.1.5	Output data per channel (Array)
_01_AD_ERR_ARY	ARRAY[0..1] OF BOOL	%UX0.1.32	Error per channel (Array)
_01_AD_FILTCONST_ARY	ARRAY[0..1] OF WORD	%UW0.1.15	Filter constant per channel (Array)
_01_AD_HOLDVAL_ARY	ARRAY[0..1] OF BOOL	%UX0.1.320	Hold effective conversion value per channel (Array) setting
_01_AD_HOOR_ARY	ARRAY[0..1] OF BOOL	%UX0.1.48	Alarm (Upper Limit) per channel (Array)
_01_AD_IDD_ARY	ARRAY[0..1] OF BOOL	%UX0.1.72	Input Disconnection flag per channel (Array)
_01_AD_LOOR_ARY	ARRAY[0..1] OF BOOL	%UX0.1.56	Alarm (Lower Limit) per channel (Array)
_01_AD_RANGE_ARY	ARRAY[0..1] OF BYTE	%UB0.1.22	Range setting per channel (Array)
_01_AD_RUN_ARY	ARRAY[0..1] OF BOOL	%UX0.1.160	Operation setting per channel (Array)

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(Voltage) Input data type
_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(Voltage) Output status setting
_01_DA0_RANGE	BYTE	%UB0.1.24	Channel 0(Voltage) Range setting
_01_DA0_RUN	BOOL	%UX0.1.168	Channel 0(Voltage) Operation setting
_01_DA1_ACT	BOOL	%UX0.1.25	Channel 1(Voltage) Active
_01_DA1_DATA	WORD	%UW0.1.9	Channel 1(Voltage) Input
_01_DA1_DATATYPE	BYTE	%UB0.1.29	Channel 1(Voltage) Input 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(Voltage) Output status setting
_01_DA1_RANGE	BYTE	%UB0.1.25	Channel 1(Voltage) Range setting
_01_DA1_RUN	BOOL	%UX0.1.169	Channel 1(Voltage) Operation setting
_01_DA_ACT_ARY	ARRAY[0..1] OF BOOL	%UX0.1.24	Active stats per channel (Array)
_01_DA_DATATYPE_ARY	ARRAY[0..1] OF BYTE	%UB0.1.28	Input data type per channel (Array)
_01_DA_DATA_ARY	ARRAY[0..1] OF WORD	%UW0.1.8	Input data per channel (Array)
_01_DA_ERR_ARY	ARRAY[0..1] OF BOOL	%UX0.1.40	Error per channel (Array)
_01_DA_INTPMTHD_ARY	ARRAY[0..1] OF BYTE	%UB0.1.46	Interpolation method setting per channel (Array)
_01_DA_INTPTIME_ARY	ARRAY[0..1] OF BYTE	%UB0.1.48	Interpolation time setting per channel (Array)
_01_DA_INTPVAL_ARY	ARRAY[0..1] OF WORD	%UW0.1.25	Interpolation value per channel (Array)
_01_DA_INTP_ARY	ARRAY[0..1] OF BOOL	%UX0.1.64	Interpolation enabled per channel (Array)
_01_DA_OUTEN_ARY	ARRAY[0..1] OF BOOL	%UX0.1.112	Output enable setting per channel (Array)
_01_DA_OUTSTAT_ARY	ARRAY[0..1] OF WORD	%UW0.1.21	Output status setting per channel (Array)
_01_DA_RANGE_ARY	ARRAY[0..1] OF BYTE	%UB0.1.24	Range setting per channel (Array)
_01_DA_RUN_ARY	ARRAY[0..1] OF BOOL	%UX0.1.168	Operation setting per channel (Array)

Built-in analog common

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

(6) Special variable

(a) These variables are assigned to the SD memory, data log and embedded encoder flag area.

(b) The built-in encoder input is 2 channels.

(c) For the memory allocated to the Special Variable, please refer to the Appendix 1 Flag List 7) SD memory Flag ~ 9) Encoder Flag.

(7) System variable

(a) These variables are assigned to the status variable of motion control status and system statues.

(b) 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, and a certain part of the device in M area can be used as retain area by setting the basic parameter.

Characteristic table of the device which is available to set retain is shown below.

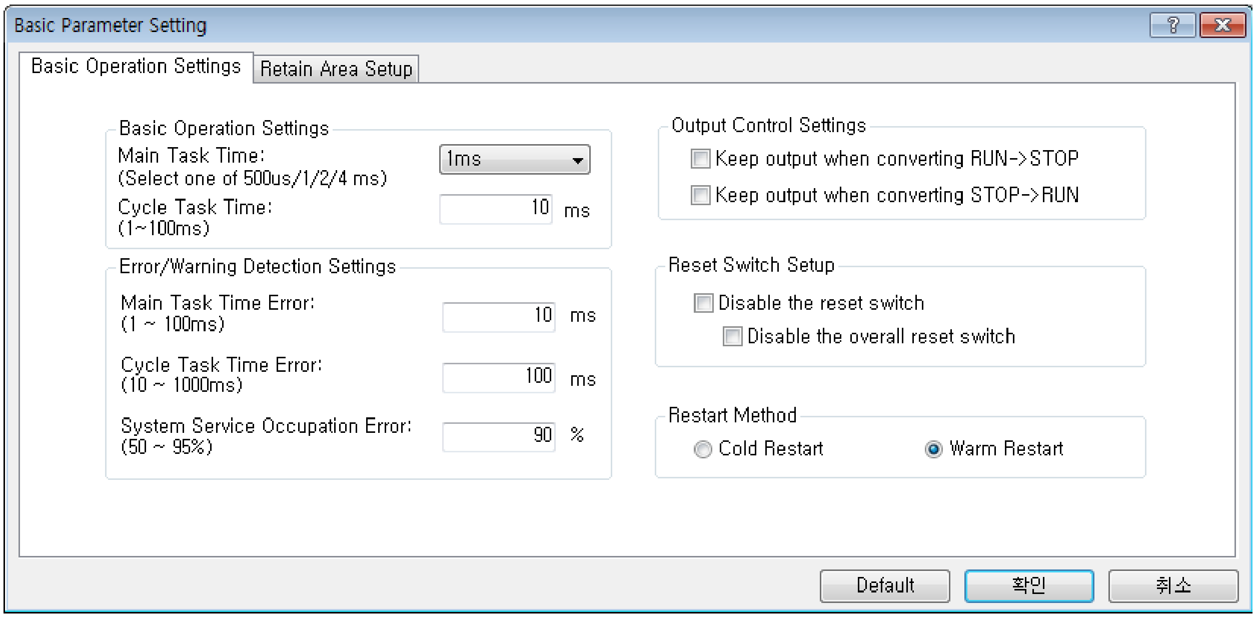
Device	Retain setting	Characteristic
Default	O	Enable retain setting when adding variable to automatical variable area
M	O	It is built-in contact area and enable retain setting at parameter
I	X	Built-in digital input, TxPDO data of EtherCAT slaves
Q	X	Built-in digital output, RxPDO data of EtherCAT slaves

5.1.3 Parameter

1. Basic parameter

Explain Basic parameter of the motion control module.

(1) Basic motion setting



(a) Main task cycle

- 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.

(b) Periodic task cycle

- 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.

(c) Detecting cycle errors of the main task

- 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.

(d) Detecting cycle errors of the periodic task

- It sets the run time of the periodic task that causes errors when the task runs beyond the set time. The setting range is 10~100ms.

(e) 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.

(f) Output control setting

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

Selection	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.

(g) Setting whether or not to turn off the Reset switch

- It sets whether or not to ‘Reset’ with the switch on the front panel of the product. The item can be set to ‘Allowed’ or ‘Prohibited’.

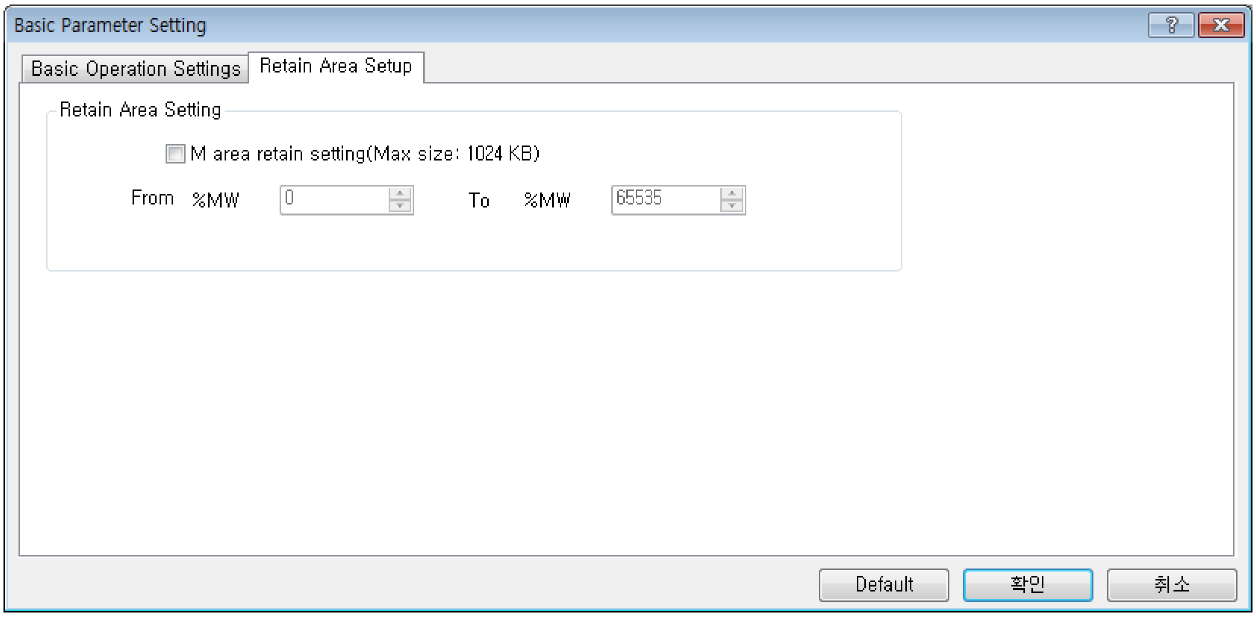
(h) Restart mode

- 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

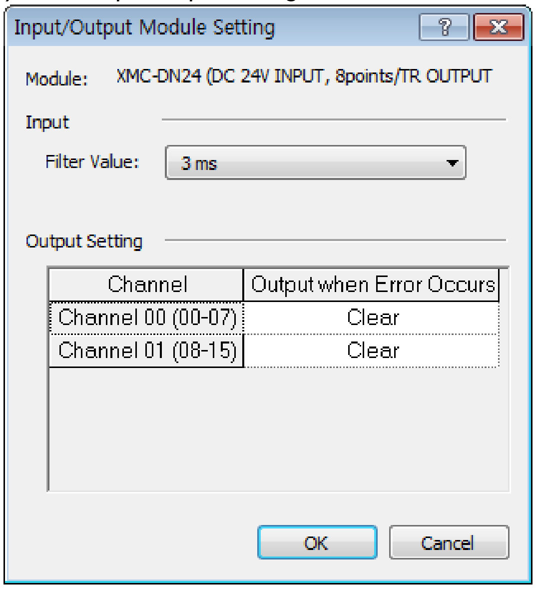
(2) Memory area setting

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

(1) Built-in input/output setting

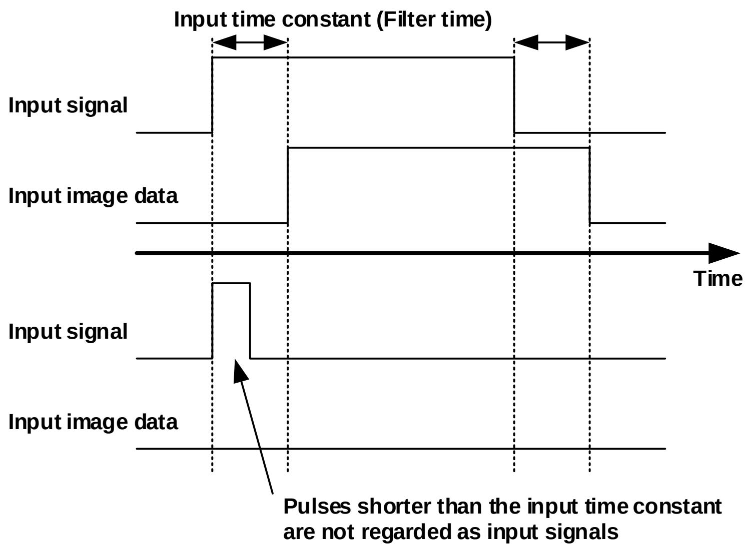


(a) Input filter function

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.



(b) Emergency output function

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 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'.

(c) Built-in analog setting

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

3. Internal parameter

(1) Data log

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

(2) Encoder

Encoder parameter is explained as follows.

Item	Description	Settings	Initialize value
Encoder1 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

Item	Description	Settings	Initialize value
Encoder1 Pulse input	Set the input mode in accordance with the output shape of encoder.	0: CW/CCW (x1) 1: PULSE/DIR (x1) 2: PULSE/DIR (x2) 3: PHASE A/B (x1) 4: PHASE A/B (x2) 5: PHASE A/B (x4)	3: PHASE A/B (x1)
Encoder1 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: No use 1: 500kPPS 2: 200kPPS 3. 100kPPS 4: 10kPPS 5: 1kPPS 6: 0.2kPPS	0: No use
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
Encoder2 Unit	Set display unit of encoder position.	0: pulse 1: mm 2: inch 3:degree	0: pulse
Encoder2 Pulses per rotation	Set Encoder2 pulses per rotation	1 ~ 4294967295	8192 pls
Encoder2 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 (x1) 1: PULSE/DIR (x1) 2: PULSE/DIR (x2) 3: PHASE A/B (x1) 4: PHASE A/B (x2) 5: PHASE A/B (x4)	3: PHASE A/B (x1)
Encoder2 Max. value	Set position display range of encoder.	Long Real(LREAL)	2147483647 pls
Encoder2 Min. value			-2147483648 pls
Encoder2 speed unit	Set the encoder speed display unit.	0: Unit/sec 1: Unit/min 2: rpm	0: Unit/sec

Item	Description	Settings	Initialize value
Encoder2 input filter value	Limit the frequency of pulse input to encoder.	0: No use 1: 500kPPS 2: 200kPPS 3. 100kPPS 4: 10kPPS 5: 1kPPS 6: 0.2kPPS	0: No use
Encoder2 position filter time constant	Set the time constant (in hours) of the filter to calculate the encoder's position average.	0 ~ 1000	0 ms

(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.

Remark

When the encoder unit is different from the synchronous operation axis, it operates by the synchronous ratio regardless of the unit.
[Setting example]
- Encoder unit: pulse
- Encoder resolution: 4096 pulse
- Unit of Synchronous operation axis: mm
- Master axis : Slave axis = 2 : 1

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

(b) Encoder Pulses per rotation

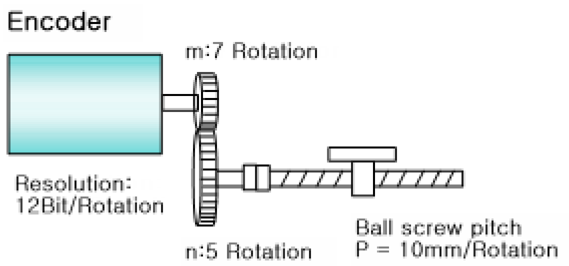
When using mm, inch, and degree for the encoder unit, set the number of purses per encoder rotation.

(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.

[Setting 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
  = 28672 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

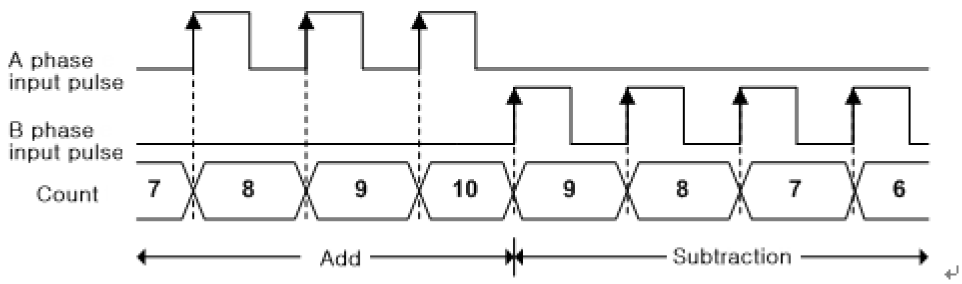
When wanting to use the encoder signal of servo drive or manual pulse generator as an input, the signal, which is right to the output form of the encoder or manual pulse generator, can be selected to be used.

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 (x1)

Count operation is performed when A phase input pulse increases or B phase input pulse increases; and adding operation is performed when A phase input pulse increases in the Low input of B phase input pulse; and subtraction operation is performed when B phase input pulse increases in the Low input of A phase input pulse.

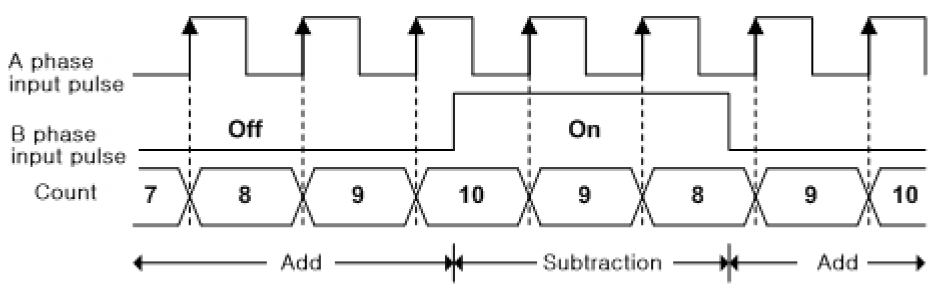
Add/Subtraction	A phase input pulse High	A phase input pulse Low
B phase input pulse High	-	Subtraction count
B phase input pulse Low	Add count	-



2) PULSE/DIR (x1)

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

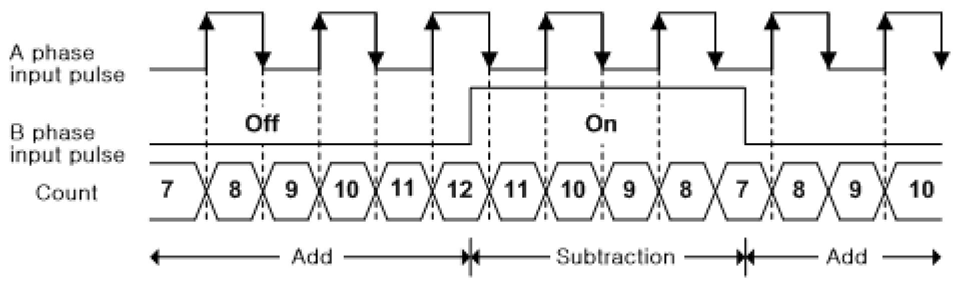
Add/Subtraction	A phase input pulse High	A phase input pulse Low
B phase input pulse Off	Add count	-
B phase input pulse On	Subtraction count	-



3) PULSE/ DIR (x2)

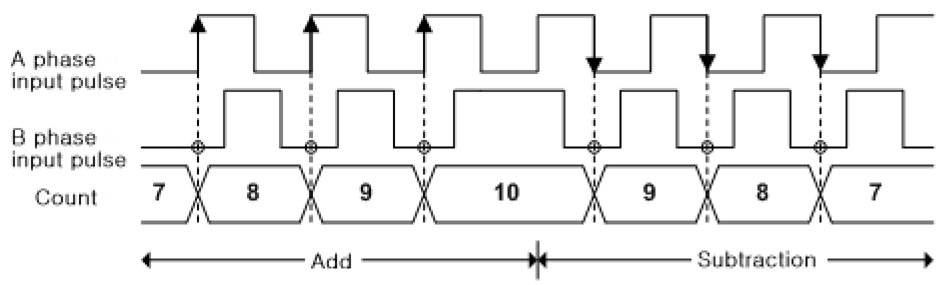
Count operation is performed when A phase input pulse increases and decreases, and whether to be added or subtracted is decided by B phase.

Add/Subtraction	A phase input pulse High	A phase input pulse Low
B phase input pulse Off	Add count	Add count
B phase input pulse On	Subtraction count	Subtraction count



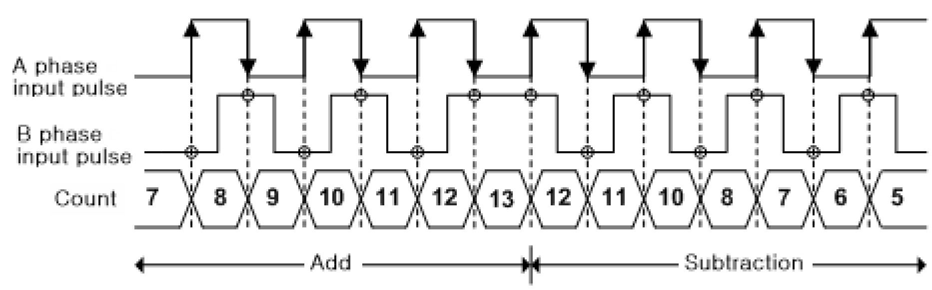
4) PHASE A/B (x1)

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.



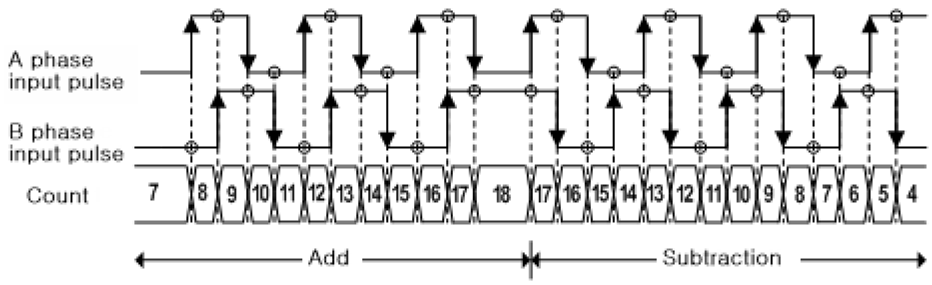
5) PHASE A/B (x2)

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 (x4)

Count operation is performed in case of the increase/decrease in A phase input pulse and the increase/decrease in B phase; and 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.

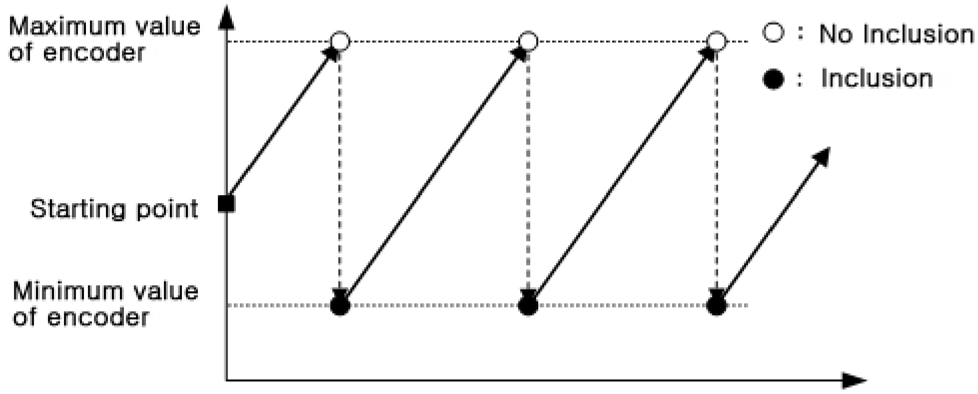


(e) Maximum and minimum values of encoder

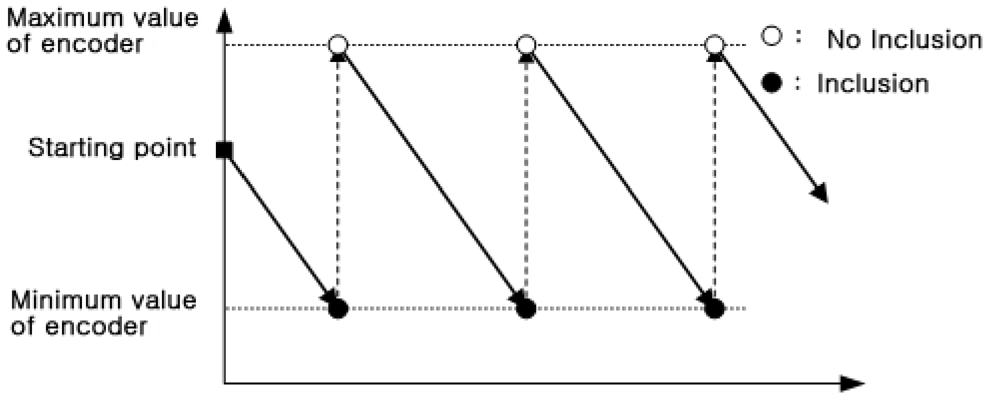
1) The range of the encoder value is set to the maximum and minimum values of encoder when counting the input pulse from the encoder signal of servo drive or manual pulse generator and indicating it to encoder value.

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.

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

When set to '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 500 kPPS.

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.

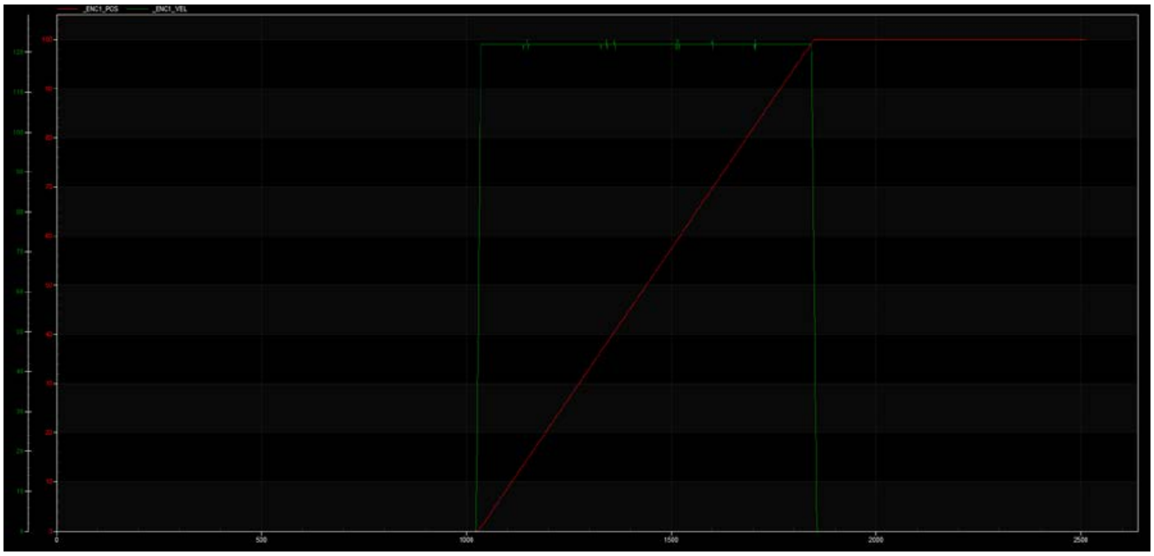
(h) Encoder position filter time constant

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.

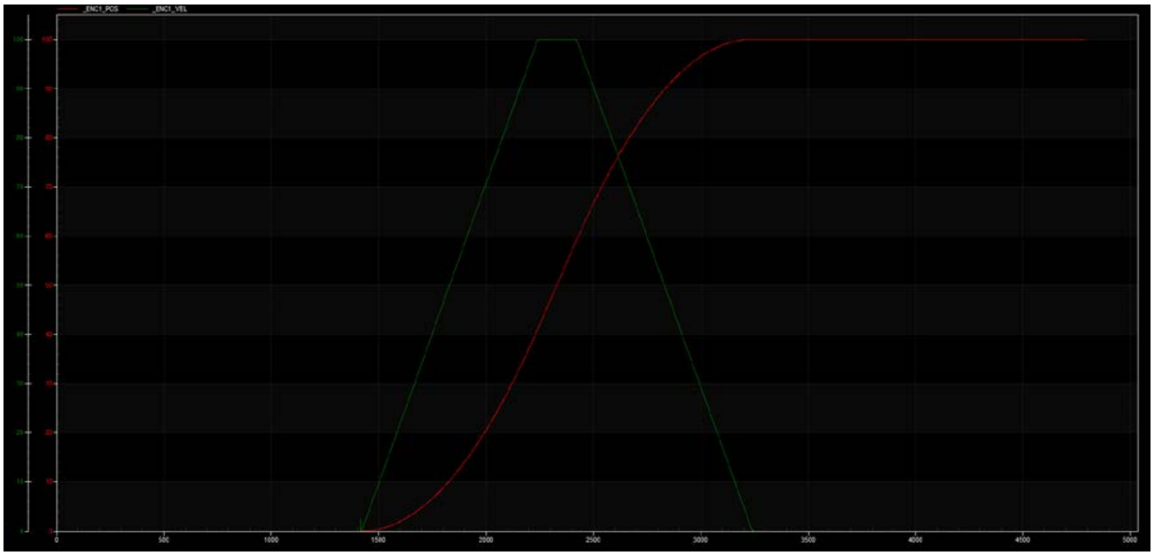
If the deviation of the current position is severe, such as when the ‘Unit’ setting of the encoder is ‘0: pulse’, a stable position can be obtained by applying the position average to the current position.

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.

(1) 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.

(a) Registration Information

Item	Description	Setting range	Initial value
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

1) 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.

2) 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.

3) 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.

(b) PDO variable information

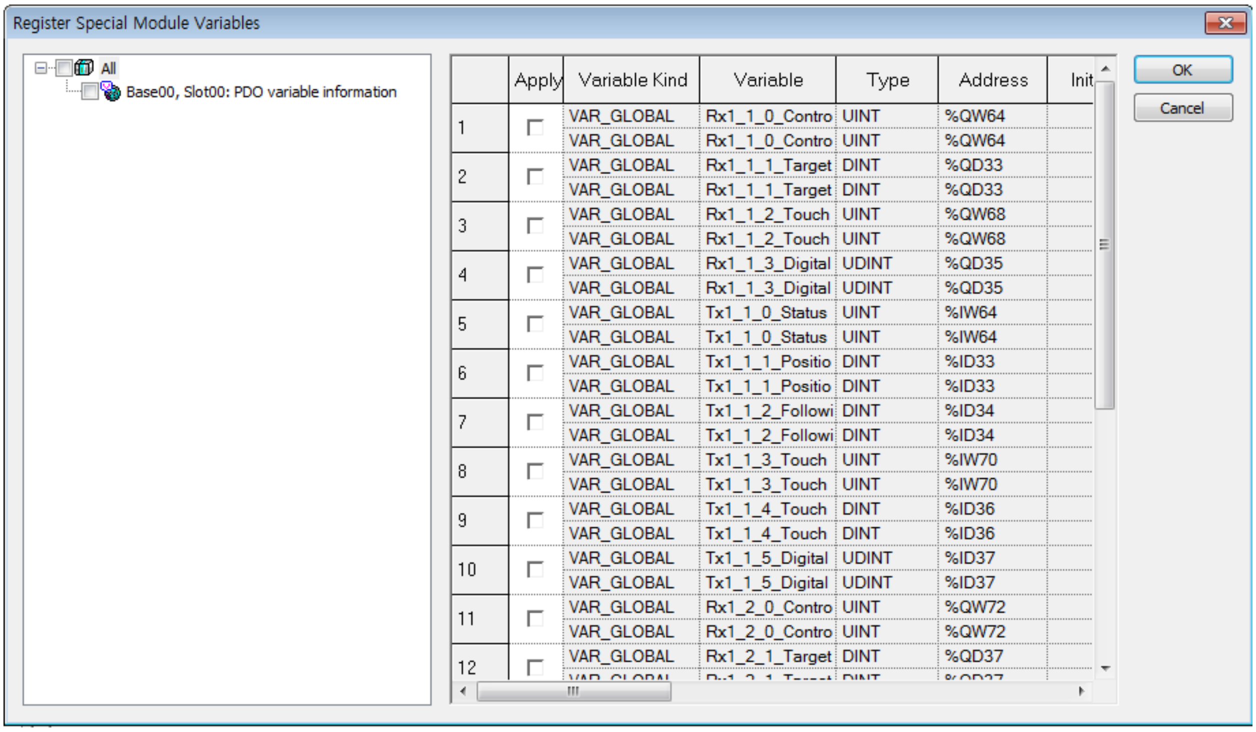
It shows the information on the memory allocation of PDO data of the registered slave.

General Info.
PDO Variable

Format: HEX
Variable setting

	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".



(1) Slave

(a) 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	Description	Setting range	Initial value
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 name of the selected slave supplier is automatically displayed.	Not configurable	-
Version	The revision of the selected slave is automatically displayed.	Not configurable	-
Serial number	The serial number of the selected slave is displayed.	Not configurable	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

1) 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.

2) 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.

3) Vendor

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

4) Version

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

5) 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.

6) 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.

7) Replacement function during connection

If the slave which has been stopped due to network disconnection or abnormal operation using the cable redundancy function is recovered from the error and connected to the network, it will detect the connection. 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’.

(b) PDO settings

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	Controlword
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	Statusword
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).

(c) SDO parameters

- 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.

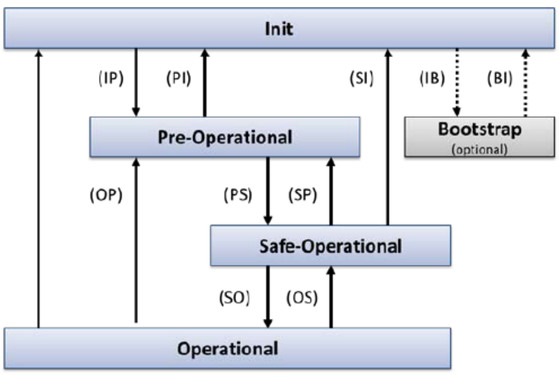
(d) 'Start' command

- 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	Description	Setting range	Initial value
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	-
Statement	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.



(e) Online service

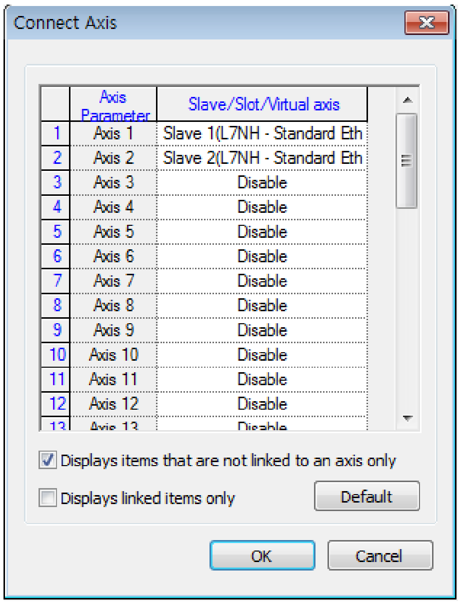
- For more information on the online service, refer to Chapter 08 Motion Control Functions -8.5.FoE Functions.

5. Axis Parameter

(1) 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.

(2) Axis parameter

(a) Basic setting

Basic parameter among basic settings is explained as follows.

Item	Description	Settings	Initial value
Unit	Set the command position unit of the axis.	0: pulse 1: mm 2: inch 3:degree	0: pulse
Pulses per rotation	Set the number of pulses per rotation of motor which corresponds encoder resolution.	1 ~ 4294967295	524288 pls
Travel per 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/Time 1: rpm	0: Unit/Time
Speed limit	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 the gear ratio between motor and 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: E.Stop 1: Stop	0: E.Stop

1) 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.

2) Pulses per rotation

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.

3) 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.

4) 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/time', 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 setting of the 「unit」 is in mm, the unit of the speed command is 'mm/s'.

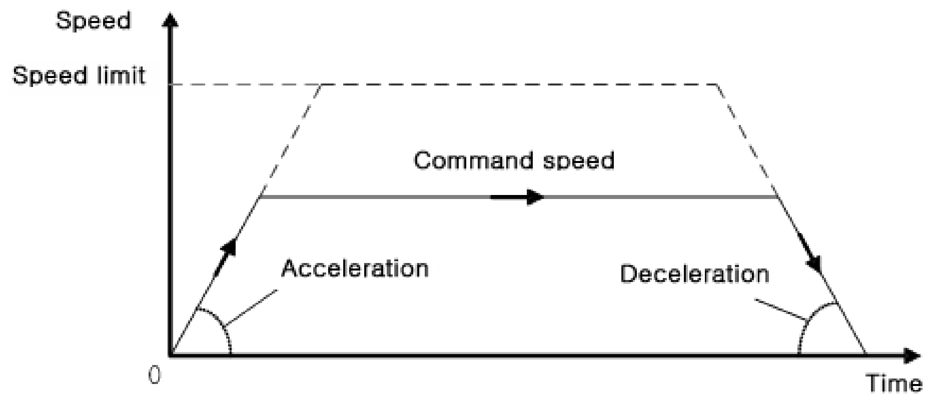
If it is set to '1: rpm', rpm is applied to the speed command unit. If the speed command unit is rpm and it is internally changed to the unit speed, values set in the 「Pulses per rotation」 and 「Travel per rotation」 parameters are used.

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.

5) Speed limit

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.



6) 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 a 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 XG-PM
- 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.)

7) Encoder select

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

The following shows the setting of "Encoder select"

Item	Settings	Description
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.

8) 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.

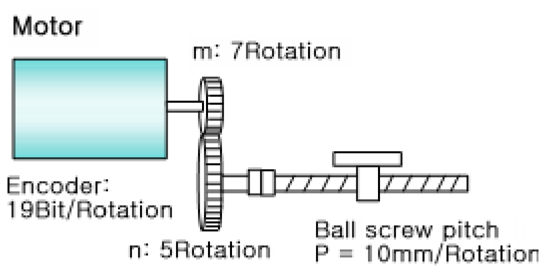
• Motor side gear ratio = m

• Machine side gear ratio = n

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

[Setting Example]

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

· Gear ratio(Machine) = 5

Note

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.

9) Operation mode of the reverse rotation

Specify the method of motion when the operation direction is reversed in the input conditions of newly executed commands.

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.

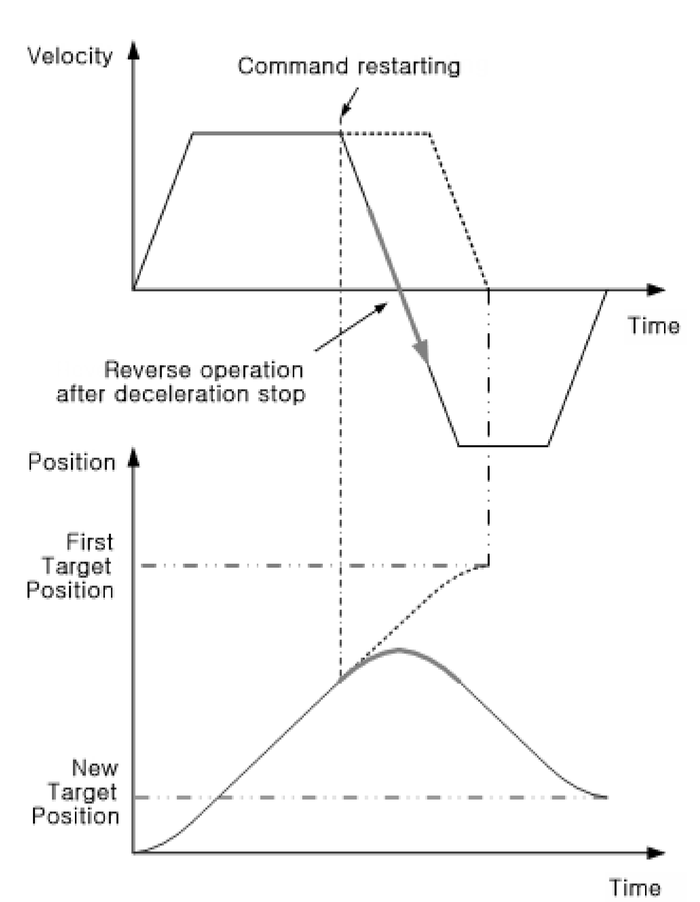
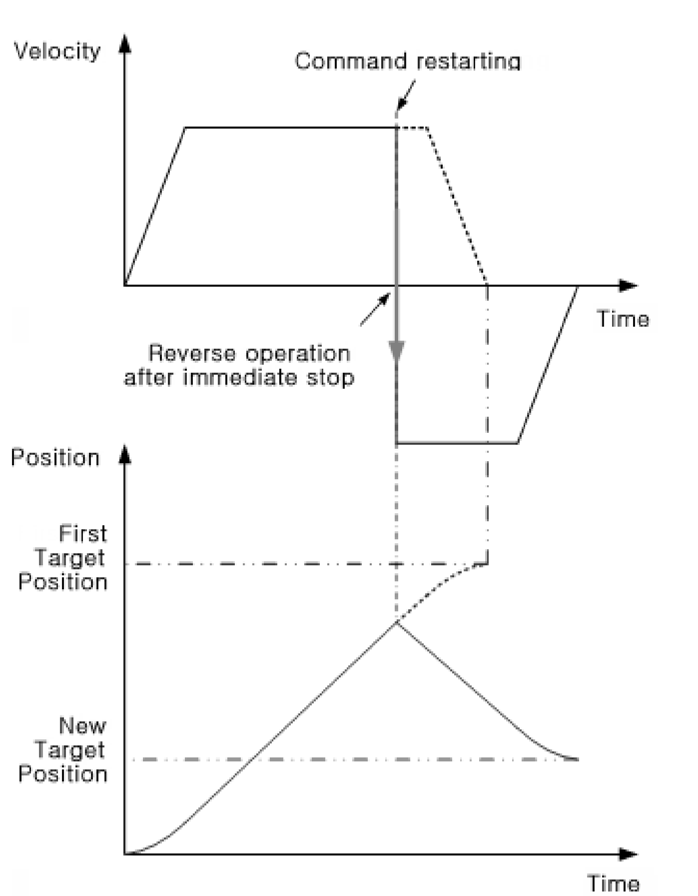
If the BufferMode is not Aborting, it is run in the specified continuous running method in the BufferMode rather than the method set in the parameter.

- '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: Immediate 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
	

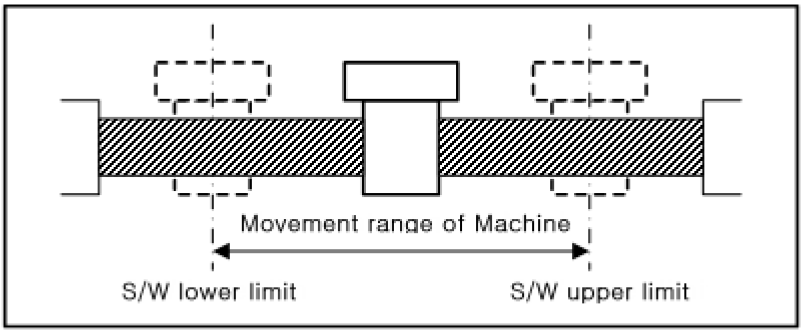
(b) Extended parameter

The following explains extended parameter of operation parameter

Item	Description	Settings	Initial value
S/W upper limit	Set the range of the software limit functions.	Long real(LREAL)	2147483647 pls
S/W lower limit			-2147483648 pls
Infinite running repeat position	Set the value of the repetitive position range in case infinite running repeat mode is used.	Long real(LREAL) Positive number	360 pls
Infinite running repeat	Set the allowable status of infinite length repetitive operation functions.	0: Disable 1: Enable	0: Disable
Command inposition range	Set the range where inposition 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
Tracking error level	Set the error level more than deviation.	0: Warning 1: Alarm	0: Warning
Current position 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
Current speed filter time constant	Set the time to calculate movement average of the current speed.	0 ~ 100	0
Error reset monitoring time	Set the monitoring time when resetting error occurred in servo drive.	1 ~ 1000	100
Software limit during speed control	Set whether the soft limit is detected during the speed control.	0: Don't detection 1: Detect	0: Don't detection
Override mode	Set the method of applying the input value, when override command is executed.	0: Ratio 1: Unit value	0: 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

1) Software upper limit / Software lower limit

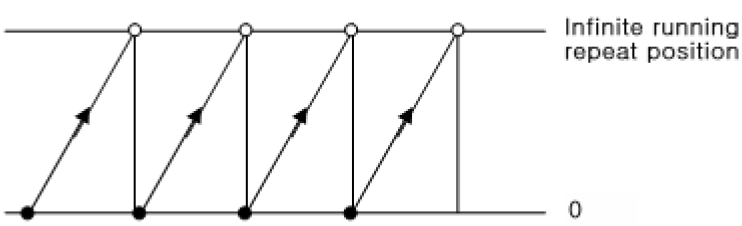
This is a function which sets the available range of the movement of the machine in the way of software by setting the upper limit & lower limit and allows the machine not to be operated beyond the set range. In other words, this is used to prevent a breakaway due to an error from setting the operation position and false operation that occurs from the user program error.
Set the external input upper limit and lower limit beyond the range of the software upper limit and the software 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 you set the software upper limit and lower limit to be the initial value (upper limit: 2,147,483,647, lower limit: -2,147,483,648) or the same value, the soft upper limit and lower limit is not be detected.

2) 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)



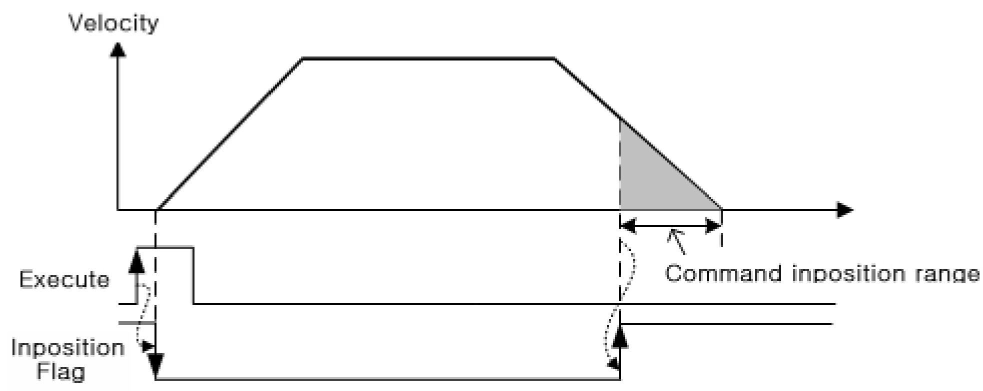
3) Infinite running repeat

Set the function availability of infinite running repeat operation.
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.

4) Command inposition range

This item sets the distance to the target position where inposition 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 inposition range」from the target position. In-position flag can be used as a trigger when executing other assistant work before completing the position control.



5) Exceeding value of tracking error

Set the value which will detect the value over 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.

6) Tracking error level

Set whether to make it a warning or an alarm when the value over deviation is detected.

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 does not stop and keeps operation.

- '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

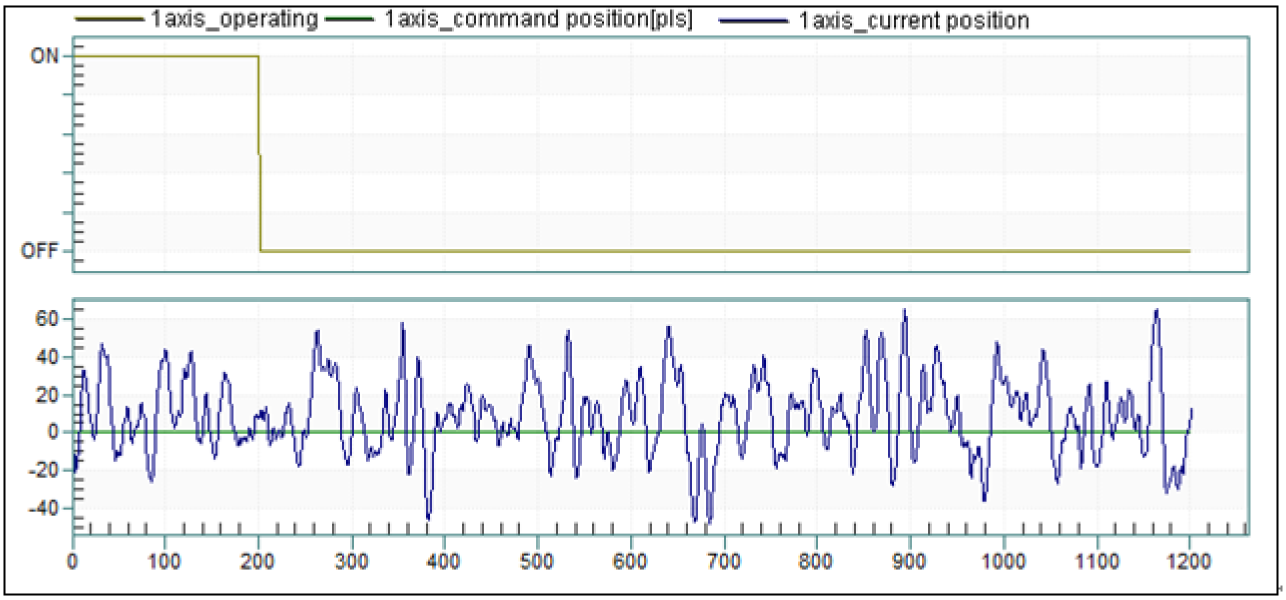
7) Current position compensation amount

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’.

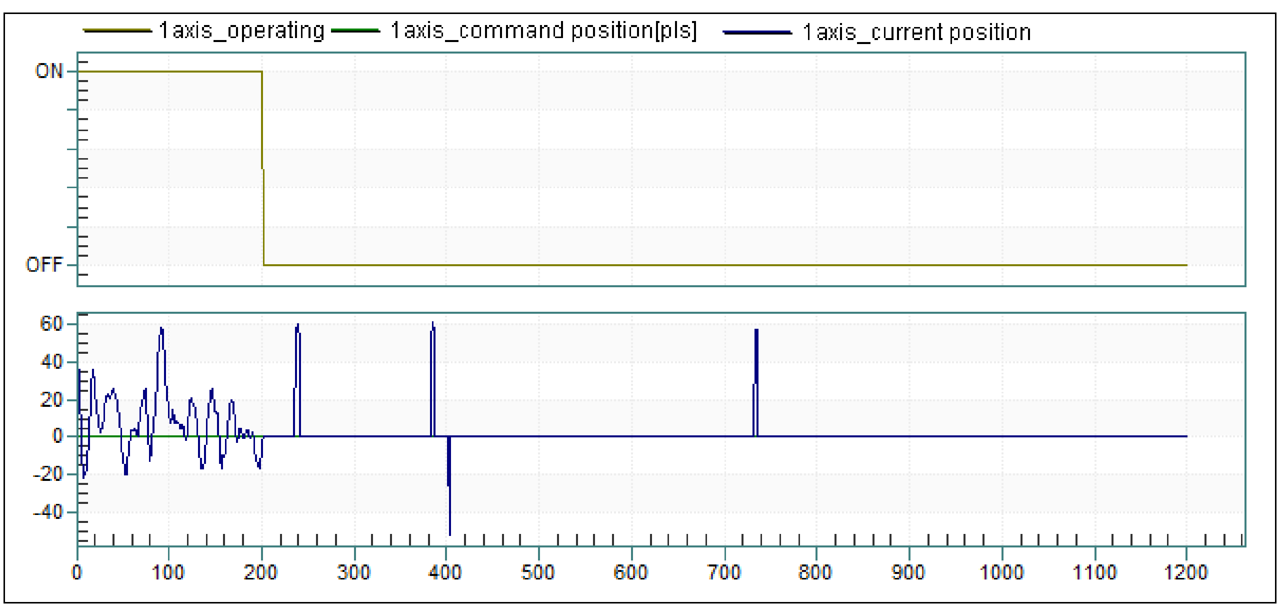
① Current position compensation amount = 0 pls

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



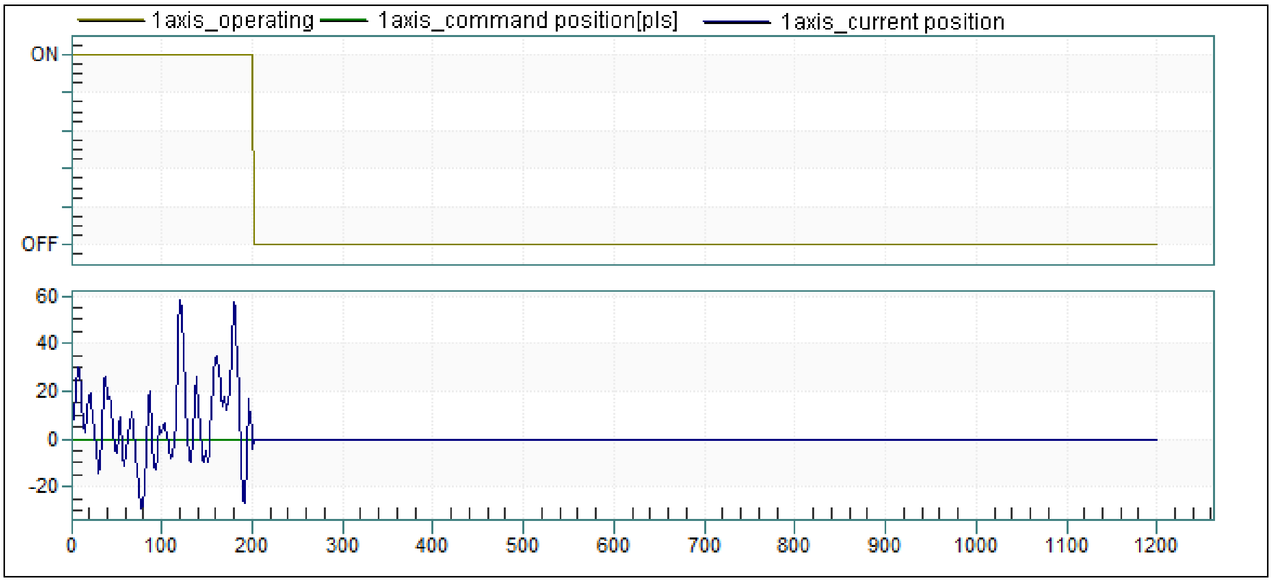
② Current position compensation amount = 50 pls

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.



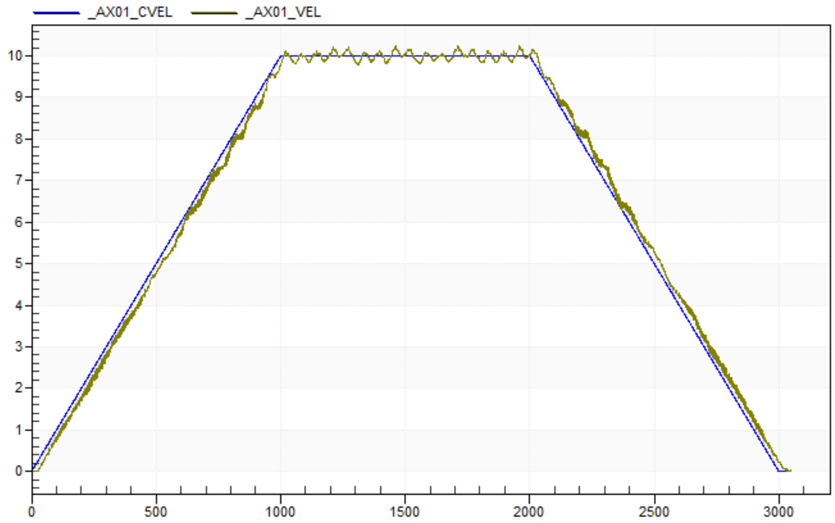
8) Current speed filter time constant

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

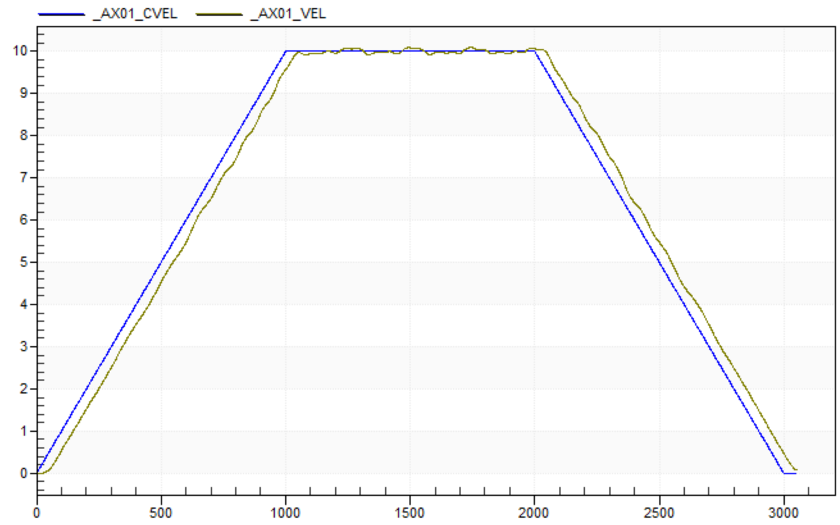
When the speed of axis is slow or there are wide variations in current speed (ex. 「unit」 setting is '0: pulse', stable speed can be achieved by applying the average of movement to the current speed.

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.

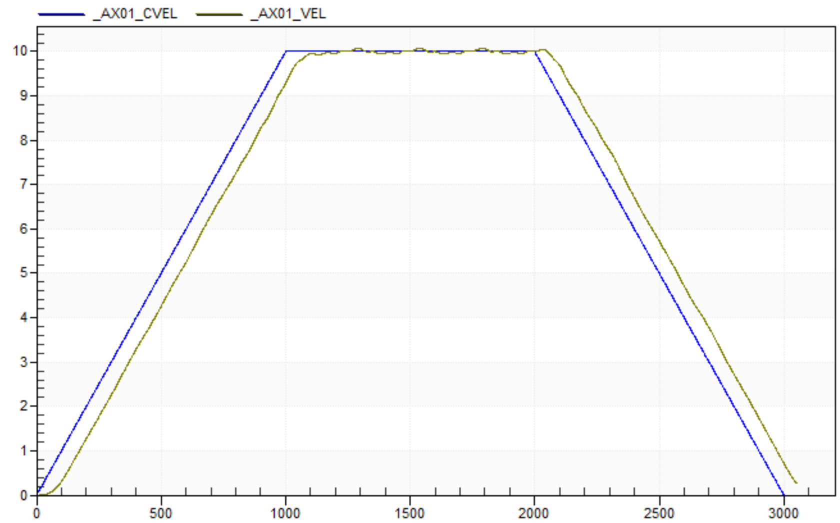
① Current speed filter time constant = 0 ms



② Current speed filter time constant = 50 ms



③ Current speed filter time constant = 100 ms



9) Error reset monitoring time

Set the monitoring time in the event of error reset occurred in the 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.

10) S/W 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. Operations according to the set values are as follows.

- '0: Don't 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.

11) JOG high speed / JOG low speed

Jog speed is related to the speed when operating jog which is a type of manual operation. Jog operation is divided into JOG high speed and JOG low speed.

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.

12) 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

(c) NC parameters

It describes the NC parameters of the axis parameters.

Item	Description	Setting range	Initial value
Identifying range to reach the spindle rotation command speed	Determine whether or not the command speed of the spindle axis is reached by the set value.	0~100%	95%
Identifying RPM to reach the spindle rotation zero speed	Determine whether or not the zero speed of the spindle axis is reached by the set value.	0~100rpm	5rpm

6. Axis group parameter

(1) Basic setting

Basic setting item is explained as follows.

Item	Description	Settings	Initial value
Configuration Axis1~10	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

(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.

(2) Coordinate system setting

Coordinate system setting item is explained as follows.

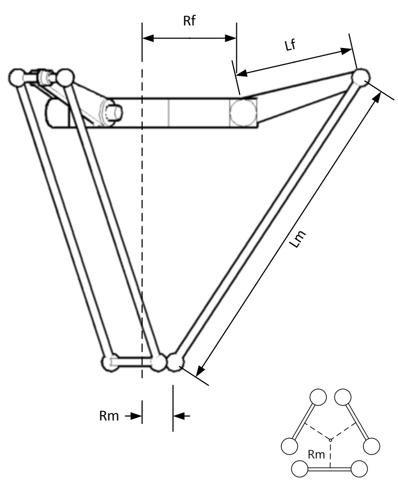
Item	Description	Settings	Initial value
Coordinate system	Set the type of robot that is applied in the operation of coordinate system.	0: None, 1: XYZ 2: Delta3 3: Delta3R 4: LinearDelta3 5: LinearDelta3R	0: None
Coordinate system parameter	Set the parameters of the machine depending on the type of coordinate 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

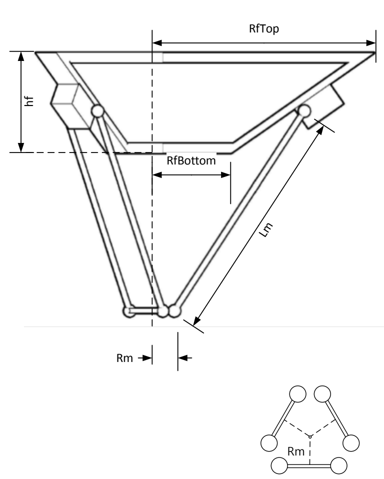
Delta is the delta robot consisting of three rotation axes. If you set the coordinate system type as Delta, you need to set the parameters of the five coordinate systems; Rf / Rm / Lf / Lm



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 the 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)
Fixed frame height (Hf)	Fixed frame height (mm)
Fixed frame radius (RfTop)	Fixed frame radius (mm)
Fixed frame radius (RfBottom)	Fixed frame radius (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)

(3) Tool setting

Tool setting item is explained as follows.

Item	Description	Settings	Initial value
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

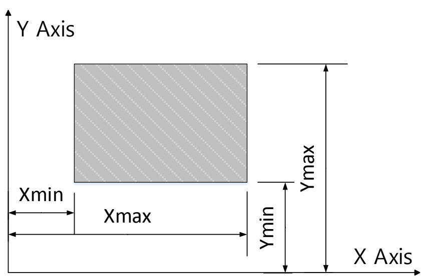
The tool setting parameter enables the position of the tool to be set as an offset so that the end of the tool can be controlled when using a separate tool at the end of the robot.

(4) Work space setting

Work space setting item is explained as follows.

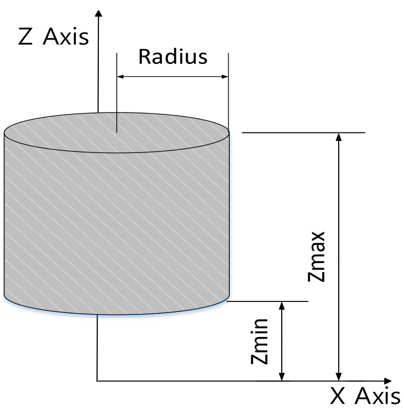
Item	Description	Settings	Initial value
Work space type	Set the type of work space.	0: No use 1: Rectangle 2: Cylinder 3: Delta 4: Sector	0
Work space error check	Set whether or not an error occurs if it deviates from a work space	0: Prohibit 1: Allow	0
Work space parameter	Set the parameter in according to work space type.	Long real(LREAL)	0

(a) Rectangle



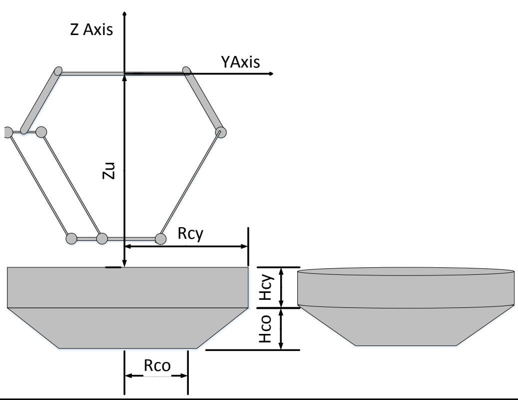
Parameter	Description
Work space parameter1	X min(mm)
Work space parameter2	X max(mm)
Work space parameter3	Y min(mm)
Work space parameter4	Y max(mm)
Work space parameter5	Z min(mm)
Work space parameter6	Z max(mm)

(b) Cylinder



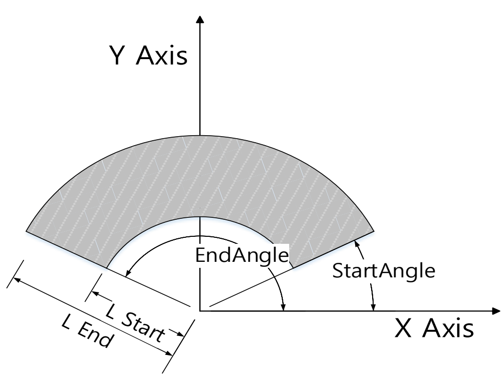
Parameter	Description
Work space parameter1	Radius(mm)
Work space parameter2	Z min(mm)
Work space parameter3	Z max(mm)

(c) Delta



Parameter	Description
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	Description
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	Description	Settings	Initial value
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	Description	Setting range	Initial value
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 parameters

- 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.

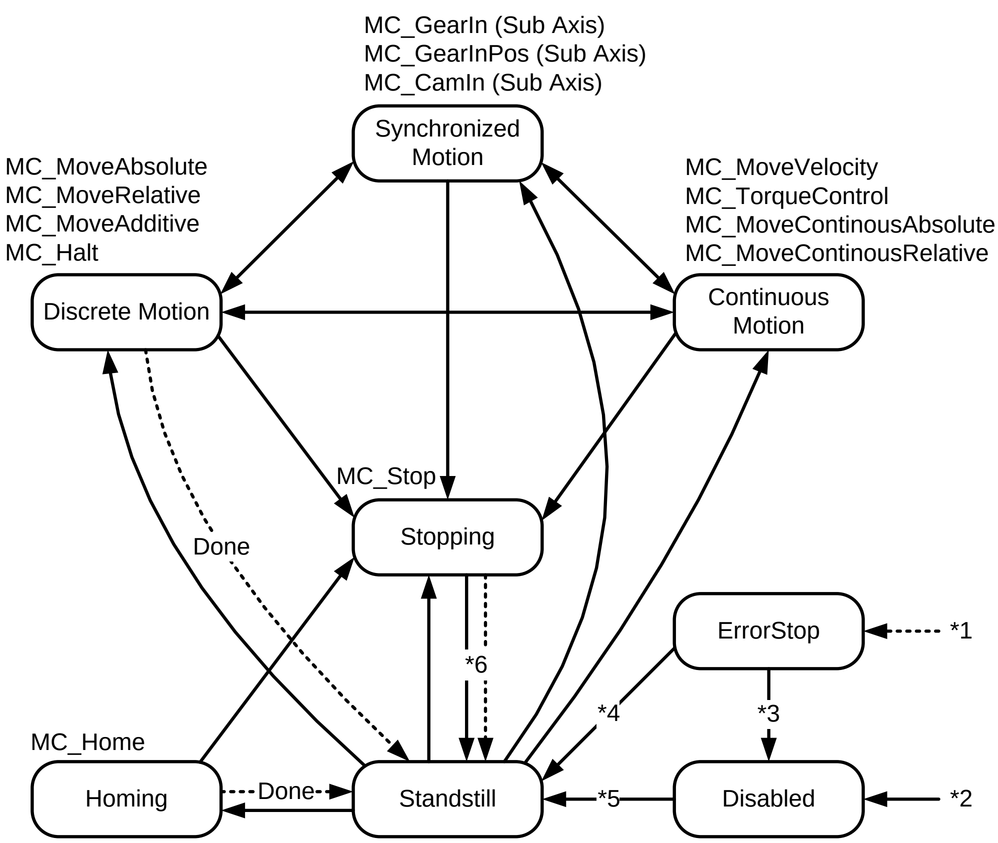
Chapter 6 Motion Function Block

This chapter describes the basic function block library mentioned in the previous chapter and other application function block library.

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

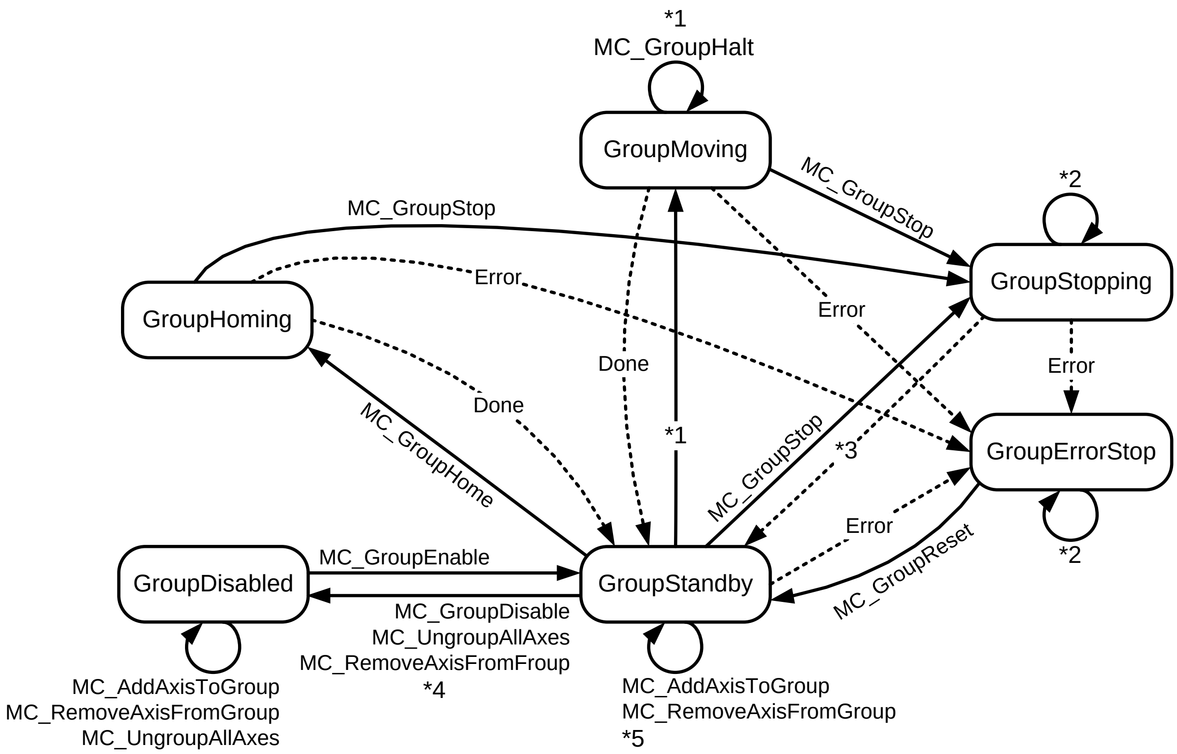
*5 Disabled → Standstill: in case of turning On MC_Power.Enable input when MC_Power.Status output is On

*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 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.(The CommandAborted output of the motion block function is On)
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	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 state indicates the axis is in homing operation.
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	It indicates state where operation continues until the current axis becomes operation stop status.
Discrete Motion	It indicates reduced operating status with target position.
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 state 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)

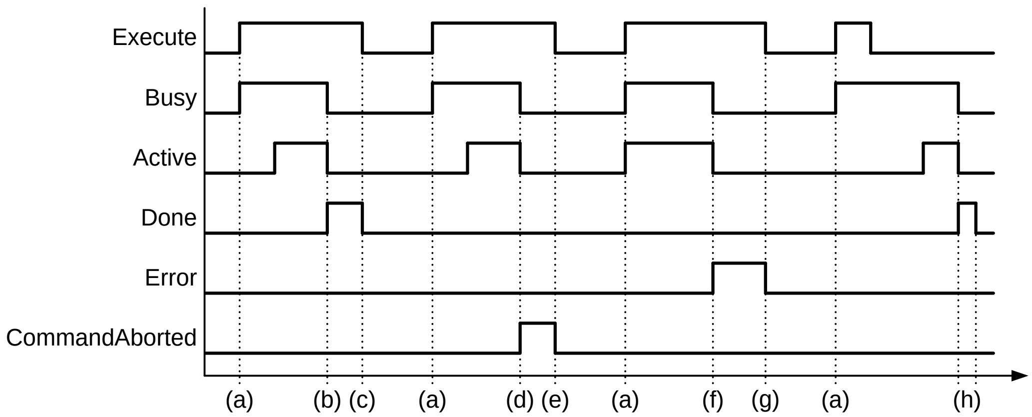
*6 GroupDisabled

: 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

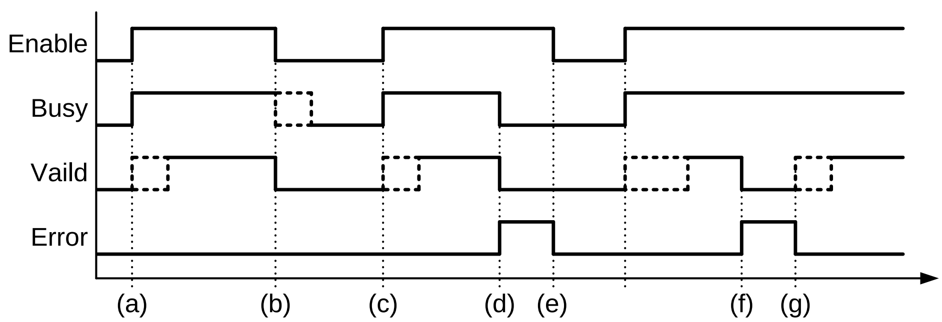
Relationships of the basic I/O parameter in the Edge operation motion function block are as below.



Variable	Description
Execute	This is an input to run the relevant function block in Edge operation function block. Function block is executed in the rising Edge. (Figure a state)
Busy	This is an output to indicate the relevant motion function block is currently running (= not completed), and this 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).

Variable	Description
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 is an input to run function block for level operation motion. This runs motion function block in the rising Edge (Figure a state), and stops it in the falling Edge(Figure b 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 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).

Variable	Description
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.	

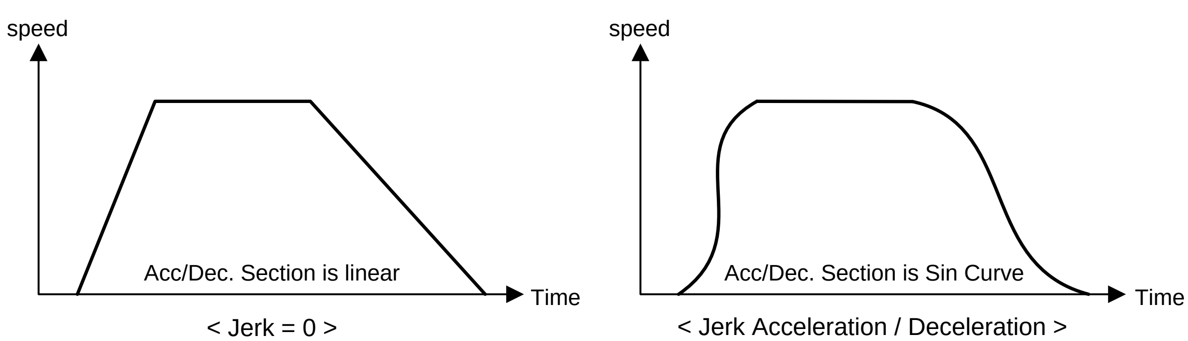
Note

1. Axis input

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

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.



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.

Number	Buffer Mode	Explanation
0	Aborting	Execute the command immediately. The existing command in operation is interrupted.
1	Buffered	Execute the command after the existing command in operation is completed.
2	BlendingLow	Do combined operation to combine the speeds of the existing command and command issuing to the low speed by comparing.
3	BlendingPrevious	Do combined operation to combine the speeds of the existing command.
4	BlendingNext	Do combined operation to combine the speeds of the command issuing.
5	BlendingHigh	Do combined operation to combine the speeds of the existing command and command giving to the high speed by comparing.

6.1.5 Changes in Parametes 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.

(1) 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.

(2) 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.)

(3) 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.

(4) 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.

(5) For MC_CAMIN function block, only the following inputs can be updated: MasterOffset, SlaveOffset, MasterScaling, SlaveScaling, MasterStartDistance, and MasterSyncPosition(If InSync=On, only MasterOffset, SlaveOffset, MasterScaling, and SlaveScaling are updated.)

(6) For MC_GEARIN function block, only the following inputs can be updated: RatioNumerator, RatioDenominator, Acceleration, and Deceleration (If InGear=On, only RatioNumerator and RatioDenominator are updated.)

6.1.6 Group Operation Route Change Settings

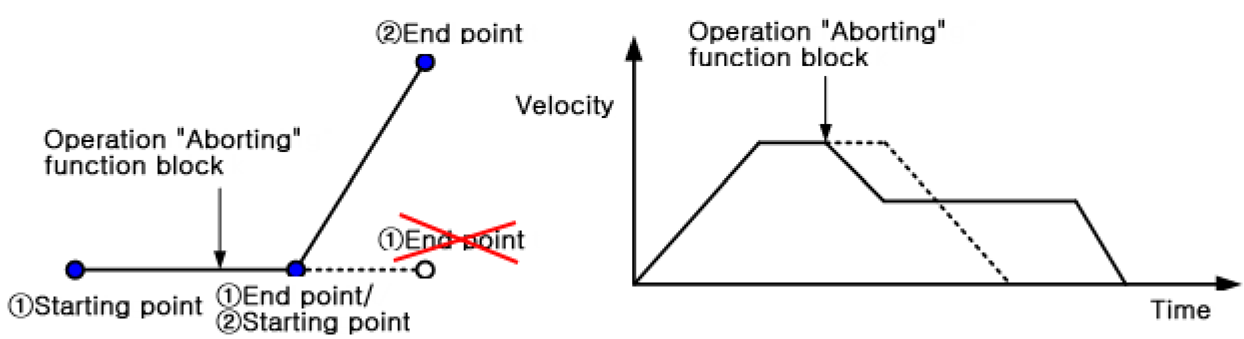
When the axis group of the current motion controller is executing a command, other command can be issued to the relevant axis group. At this point, the path, which the next command will achieve, can specify how the existing command will be connected to the existing path. The parameter of connection track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

Number	TRANSITION Mode	Explanation
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.

1. TransitionMode "TMNone"

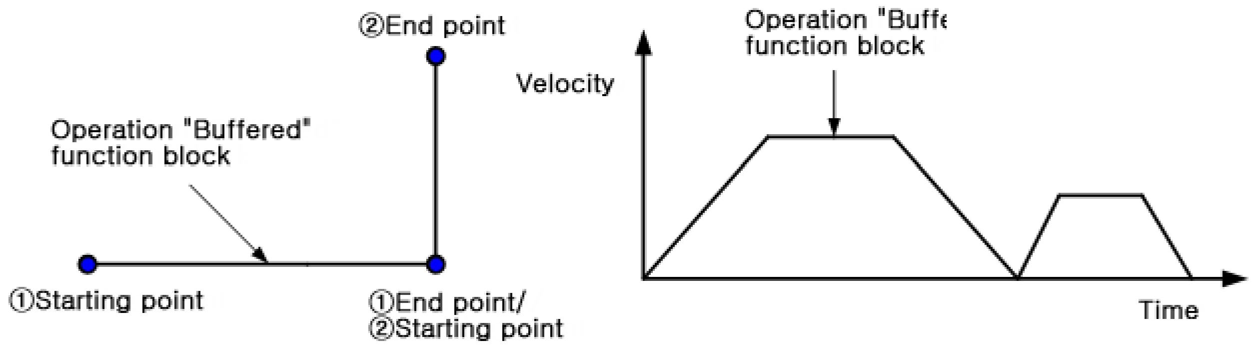
Connection track is not generated. TransitionMode input is available only to "TMNone" in case BufferMode input of motion function block is "Aborting" or "Buffered".

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 "Buffered">

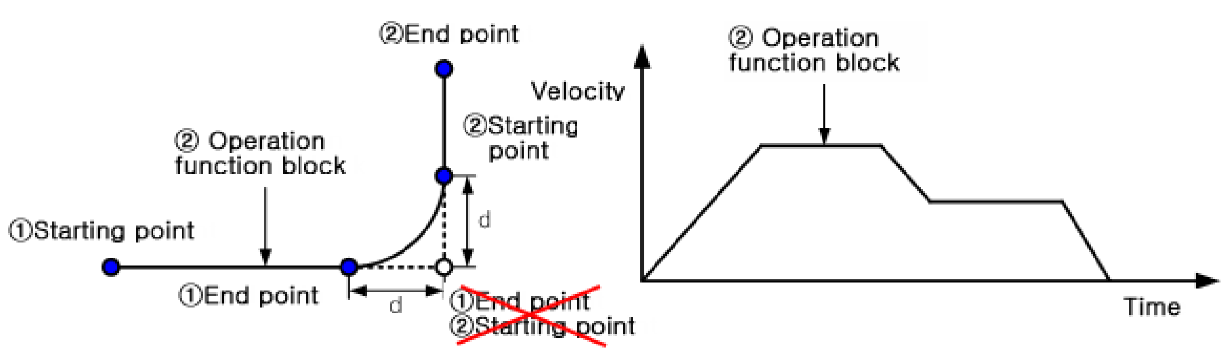
2. TransitionMode "TMCornerDistance"

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	Detailed 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 number to the areas of 1~36 or encoder number to areas of 1001~1002.
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.
0x0013	Motion response error occurred during the execution of motion function block.	Check the version of XG5000 and XMC-E32A.
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

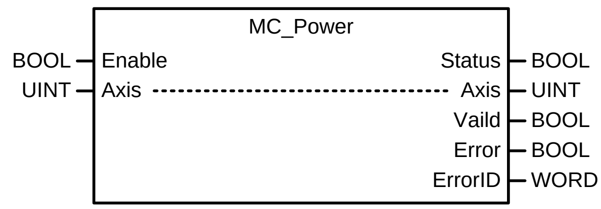
NO.	Name	Description	Operation condition
Single-axis command			
1	MC_Power	Servo On/Off	Level
2	MC_Home	Perform the search home	Edge
3	MC_Stop	Stop immediately	Edge
4	MC_Halt	Stop	Edge
5	MC_MoveAbsolute	Absolute positioning operation	Edge
6	MC_MoveRelative	Relative positioning operation	Edge
7	MC_MoveAdditive	Additive positioning operation	Edge
8	MC_MoveVelocity	Specified velocity operation	Edge
9	MC_MoveContinuousAbsolute	Absolute position operation ending with specified velocity operation	Edge
10	MC_MoveContinuousRelative	Relative position operation ending with specified velocity operation	Edge
11	MC_TorqueControl	Torque control	Edge
12	MC_SetPosition	Setting the current position	Edge
13	MC_SetOverride	Velocity/Acceleration override	Level
14	MC_ReadParameter	Read Parameter	Level
15	MC_WriteParameter	Write Parameter	Edge
16	MC_Reset	Reset axis error	Edge
17	MC_TouchProbe	Touch probe	Edge
18	MC_AbortTrigger	Abort trigger events	Edge
19	MC_MoveSuperImposed	SuperImposed operation	Edge
20	MC_HaltSuperImposed	SuperImposed operation halt	Edge
Multi-axis command			
21	MC_CamIn	Camming run	Edge
22	MC_CamOut	Camming stop	Edge
23	MC_GearIn	Electrical gearing run	Edge
24	MC_GearOut	Electrical gearing disengage	Edge
25	MC_GearInPos	Electrical gearing by specifying the position	Edge
26	MC_Phasing	Phase compensation	Edge

NO.	Name	Description	Operation condition
Group command			
27	MC_AddAxisToGroup	Adds one axis to the group	Edge
28	MC_RemoveAxisFromGroup	Removes one axis from the group	Edge
29	MC_UngroupAllAxes	Removes all axes from the group	Edge
30	MC_GroupEnable	Changes the state for group from GroupDisable to GroupEnable	Edge
31	MC_GroupDisable	Changes the state for group from GroupEnable to GroupDisable	Edge
32	MC_GroupHome	Performs the search home of all axes in the group	Edge
33	MC_GroupSetPosition	Sets the position of all axes in the group without moving	Edge
34	MC_GroupStop	Stop the group immediately	Edge
35	MC_GroupHalt	Stop the group	Edge
36	MC_GroupReset	Reset the group error	Edge
37	MC_MoveLinearAbsolute	Absolute positioning linear interpolation operation	Edge
38	MC_MoveLinearRelative	Relative positioning linear interpolation operation	Edge
39	MC_MoveCircularAbsolute	Absolute positioning circular interpolation operation	Edge
40	MC_MoveCircularRelative	Relative positioning circular interpolation operation	Edge
LS command			
41	LS_Connect	Connect servo drives	Edge
42	LS_Disconnect	Disconnect servo drives	Edge
43	LS_ReadSDO	Read SDO	Edge
44	LS_WriteSDO	Write SDO	Edge
45	LS_SaveSDO	Save SDO	Edge
46	LS_EncoderPreset	Encoder preset	Edge
47	LS_Jog	JOG operation	Level
48	LS_ReadCamData	Read CAM data	Edge
49	LS_WriteCamData	Write CAM data	Edge
50	LS_ReadEsc	Read ESC	Edge
51	LS_WriteEsc	Write ESC	Edge
52	LS_CamSkip	Skip CAM	Edge
53	LS_VarCamIn	Variable CAM operation	Edge
54	LS_VarGearIn	Variable gear operation	Edge
55	LS_VarGearInPos	Variable positioning gear operation	Edge
56	LS_ReadCAM tableSlavePos	Read the slave location of the CAM table	Edge
57	LS_InverterWriteVel	Write inverter speed	Edge
58	LS_InverterReadVel	Read inverter speed	Level
59	LS_InverterControl	Write inverter control word	Edge
60	LS_InverterStatus1	Read inverter status 1	Level
61	LS_InverterStatus2	Read inverter status 1	Level
62	LS_SyncMoveVelocity	Speed control operation (csv mode)	Edge
63	LS_ReadCamTableMasterPos	Read CAM table master position	Edge

NO.	Name	Description	Operation condition
Coordinate system command			
64	MC_SetKinTransform	Machine information setting	Edge
65	MC_SetCartesianTransform	PCS setting	Edge
66	LS_SetWorkSpaceTransform	Work space setting	Edge
67	LS_MoveLinearTimeAbsolute	Time- linear interpolation operation for absolute position of coordinate system	Edge
68	LS_MoveLinearTimeRelative	Time- linear interpolation operation for relative position of coordinate system	Edge
69	MC_MoveCircularAbsolute2D	Circular interpolation operation for absolute position of coordinate system	Edge
70	MC_MoveCircularRelative2D	Circular interpolation operation for relative position of coordinate system	Edge
71	MC_TrackConveyorBelt	Synchronization setting of the conveyor belt	Edge
72	MC_TrackRotary table	Synchronization setting of the rotary table	Edge
73	LS_RobotJOG	JOG operation of the coordinate system	Level
74	LS_SetMovePath	Set path operation data	Edge
75	LS_ResetMovePath	Delete path operation data	Edge
76	LS_GetMovePath	Read path operation data	Edge
77	LS_RunMovePath	Perform path operation	Edge
NC control command			
78	NC_LoadProgram	Specify NC program	Edge
79	NC_BlockControl	Specify Block operation	Level
80	NC_Reset	Reset	Edge
81	NC_Emergency	Emergency stop	Level
82	NC_CycleStart	Start automatic operation	Edge
83	NC_FeedHold	Feed Hold	Level
84	NC_Home	Homing	Edge
85	NC_RapidTraverseOverride	Rapid traverse override	Level
86	NC_CuttingFeedOverride	Cutting feed override	Level
87	NC_SpindleOverride	Spindle override	Level
88	NC_M codeComplete	M Code operation completed	Edge
89	NC_ScodeComplete	S Code operation completed	Edge
90	NC_TcodeComplete	T Code operation completed	Edge
91	NC_ReadParameter	Read NC parameters	Level
92	NC_WriteParameter	Write NC parameters	Edge

6.3 Single-Axis Motion Function Block

6.3.1 Servo On/Off (MC_Power)

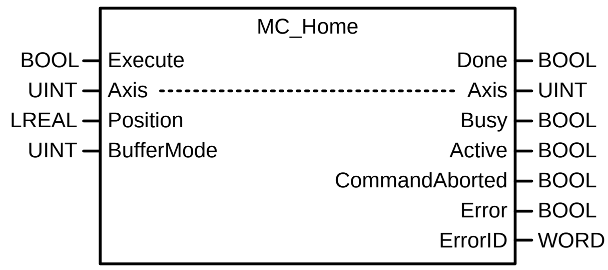
Motion Function Block		
		
Input-Output		
UINT	Axis	Specify the axis to be commanded (1~32: real/virtual axis, 33~36: virtual axis)
Input		
BOOL	Enable	Servo motor of the relevant axis is servo On while input is activated.
Output		
BOOL	Status	Indicate the power permission status of the relevant axis.
BOOL	Valid	Indicate the validity of motion function block output. (same with Status output here)
BOOL	Error	Indicate 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 is On, Servo On command is given to the relevant axis, and when it is Off, servo Off command is given.

(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 (1~32: real/virtual axis, 33~36: virtual axis)
Input		
BOOL	Execute	Start the homing operation in rising Edge.
LREAL	Position	Specify the absolute position of axis when reference signal is detected.
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 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	Indicate 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.

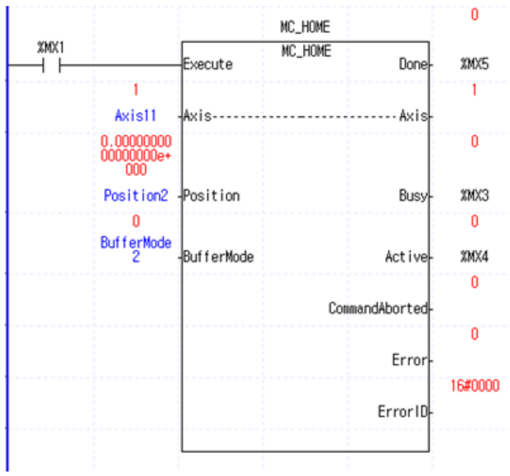
(4) While this motion function block is running, the axis is 'Homing' state, and when the command is completed, it is switched to 'Standstill'.

(5) The changed parameters can be applied by re-executing the function block (Execute input is On) before the command is completed. 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

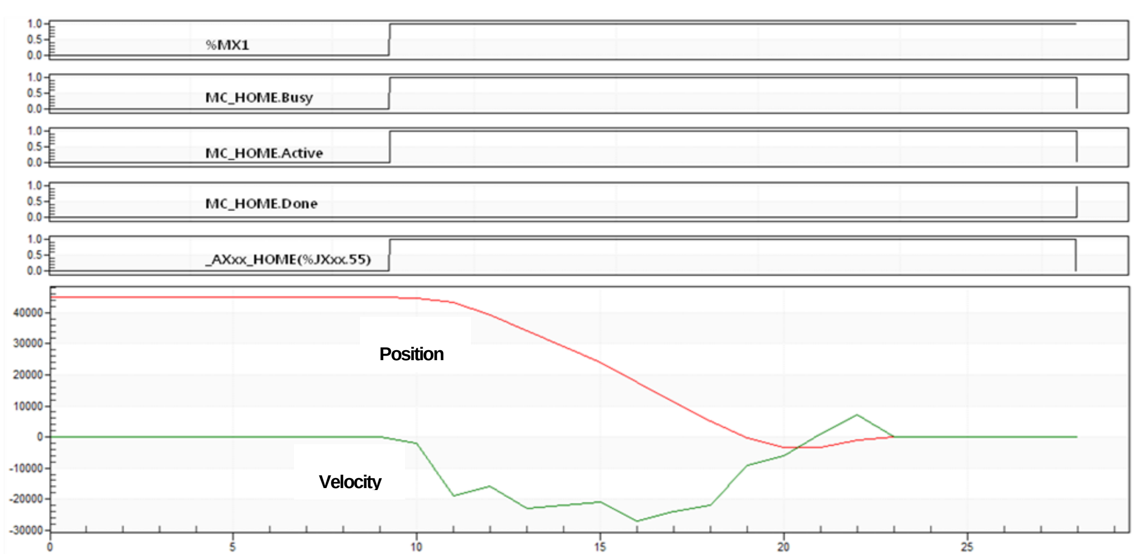


(b) Parameter setting

- Set the Homing method in SDO parameters to 33.

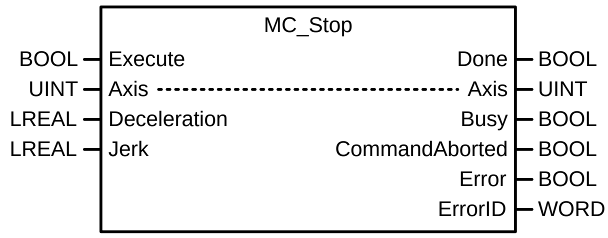
Index	Name	Unit	Current Value	Initial Value	Access
607A	Target Position	UU	0	0	rw
607C	Home Offset	UU	0	0	rw
607D:00	Software Position Limit	-	2	2	rw
607F	Maximum Profile Velocity	UU/s	2147483647	2147483647	rw
6080	Maximum Motor Speed	rpm	0	0	ro
6081	Profile Velocity	UU/s	200000	200000	rw
6083	Profile 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 (1~32: real/virtual axis, 33~36: virtual axis)
Input		
BOOL	Execute	Give immediate 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]
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	Indicate 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.

(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 is running. 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) The changed parameters can be applied by re-executing the function block (Execute input is On) before the command is completed. Only Deceleration and Jerk input can be updated

6.3.4 Stop (MC_Halt)

Motion Function Block

```
                  MC_Halt
BOOL  ─ Execute                         Done ─ BOOL
UINT  ─ Axis  -------------------------- Axis ─ UINT
LREAL ─ Deceleration                    Busy ─ BOOL
LREAL ─ Jerk                          Active ─ BOOL
UINT  ─ BufferMode          CommandAborted ─ BOOL
                                       Error ─ BOOL
                                     ErrorID ─ WORD
```

Input-Output		
UINT	Axis	Specify the axis to be commanded (1~32: real/virtual axis, 33~36: virtual axis)
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	Indicate whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

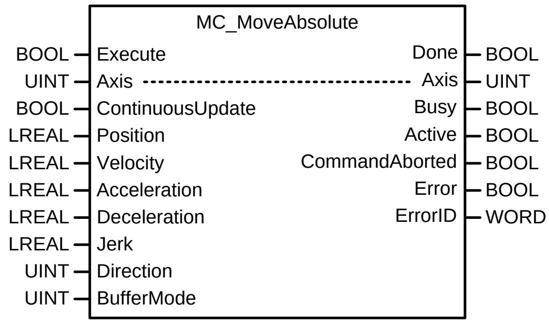
(1) This motion function block is to give a stop command to the relevant axis.

(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) The changed parameters can be applied by re-executing the function block (Execute input is On) before the command is completed.

6.3.5 Absolute positioning operation (MC_MoveAbsolute)

Motion Function Block		
		
Input-Output		
UINT	Axis	Specify the axis to be commanded (1~32: real/virtual axis, 33~36: virtual 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. (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 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	Indicate 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) Operation direction of the axis in Infinite length repetition operation is set 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) The changed parameters can be applied by re-executing the function block (Execute input is On) before the command is completed. Only Position, Velocity, Acceleration, Deceleration, Jerk, Direction input can be updated.

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

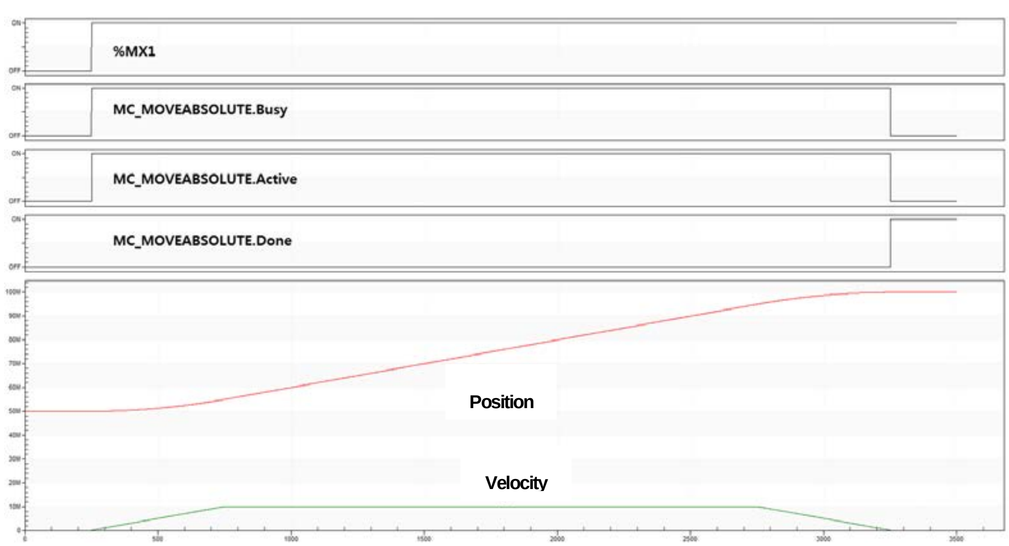
(7) During the deceleration operation, even if the Velocity and Acceleration inputs are changed by using the ContinuousUpdate function or the command re-execution function, the deceleration operation is not affected and the previous deceleration operation continues.

(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

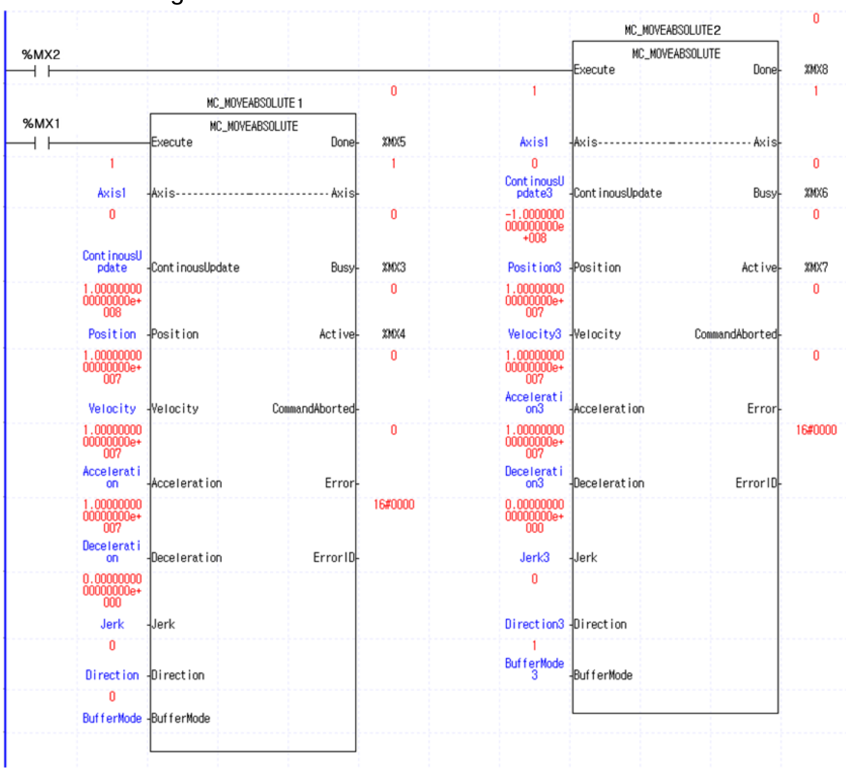
(b) Timing diagram



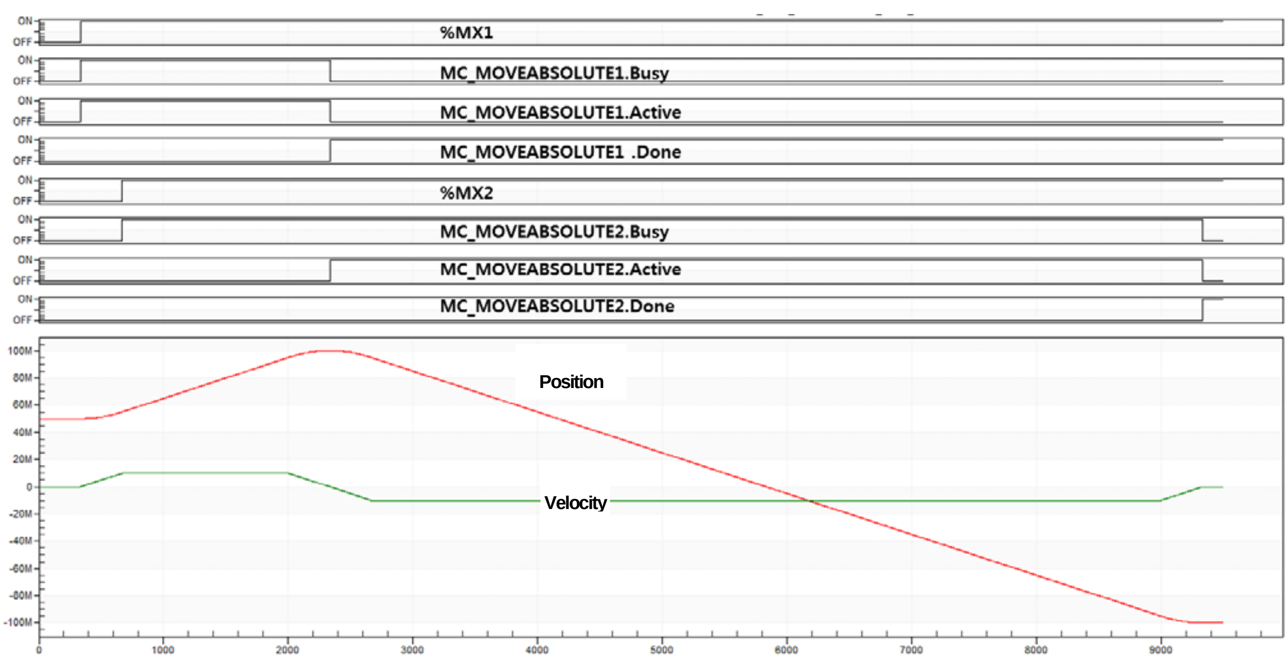
(9) Application example program

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

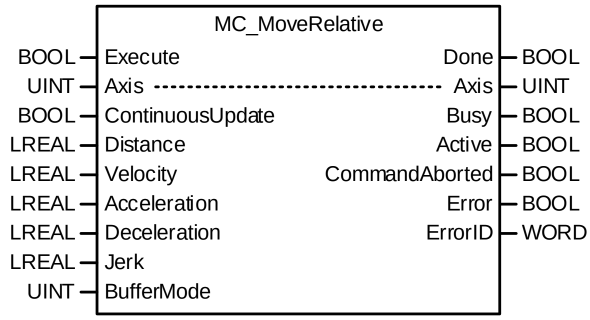


(b) Timing diagram



6.3.6 Relative positioning operation (MC_MoveRelative)

Motion Function Block



Input-Output		
UINT	Axis	Specify the axis to be commanded (1~32: real/virtual axis, 33~36: virtual 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. (Refer to 6.1.5.Changes in Parameters during Execution of Motion Function Block)
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	Indicate 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 relative position operation command to the relevant axis.

(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) The changed parameters can be applied by re-executing the function block (Execute input is On) before the command is completed. Only Distance, Velocity, Acceleration, Deceleration, Jerk input can be updated.

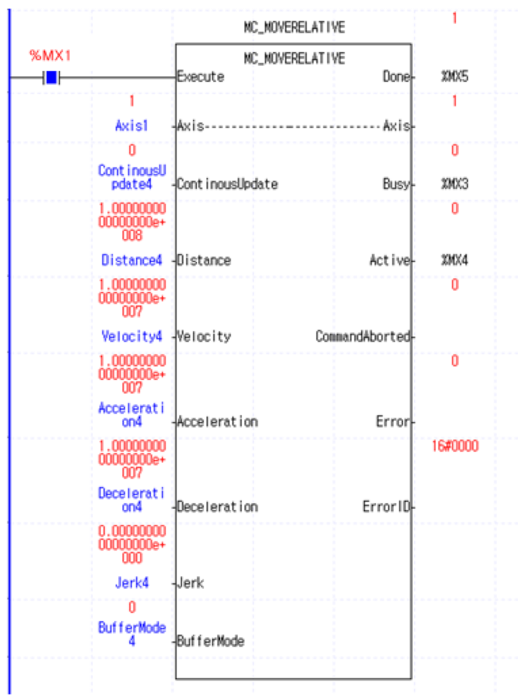
(7) Velocity input can be set to 0 or changed.

(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.

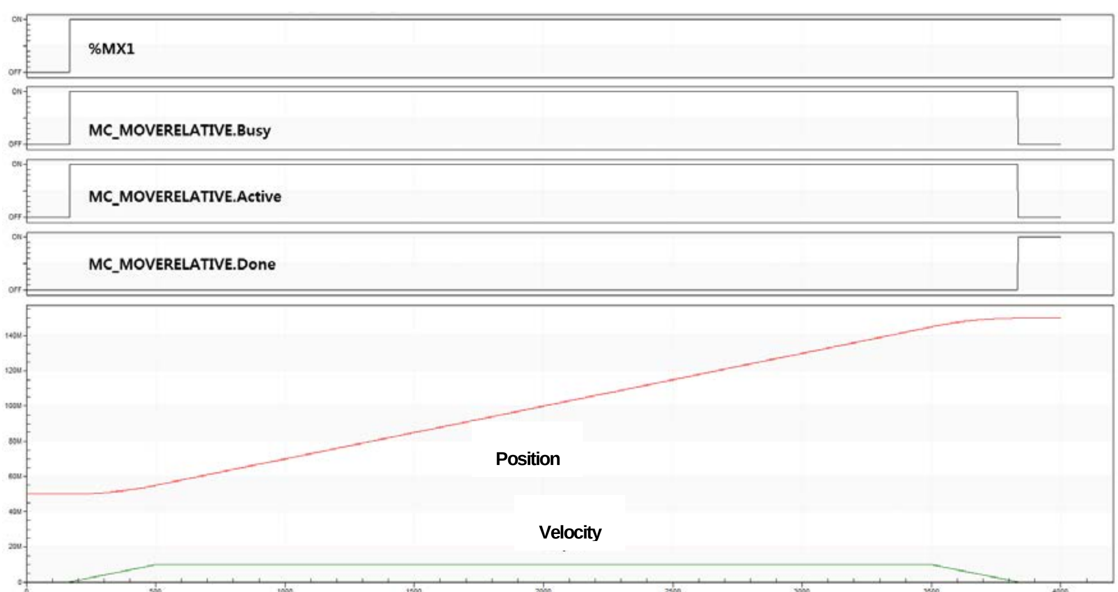
(9) Example program

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



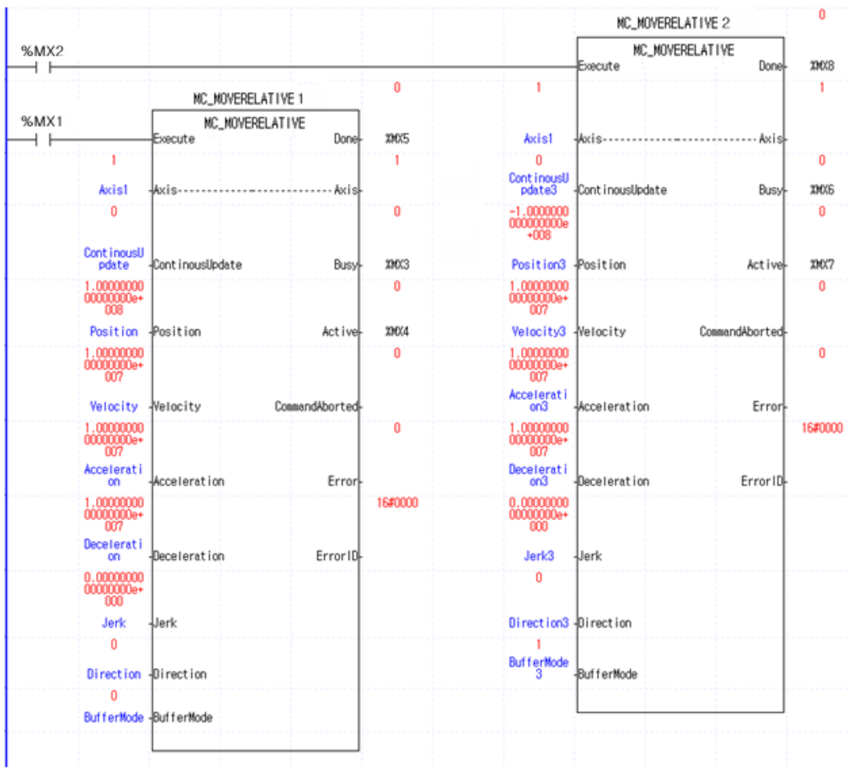
(b) Timing diagram



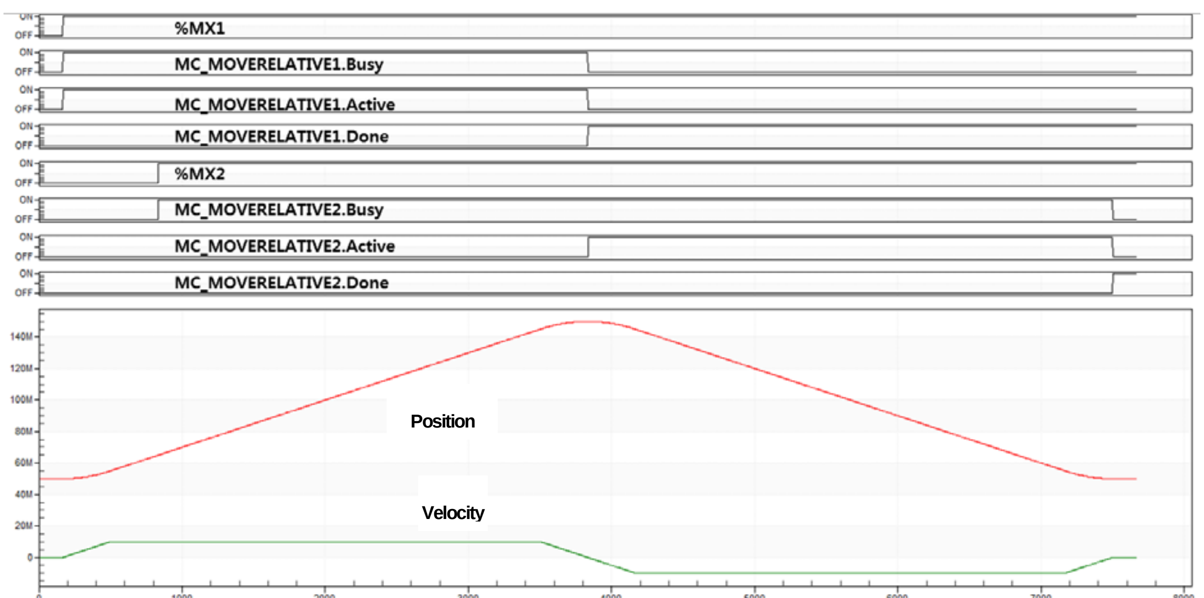
(10) Application example program

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.

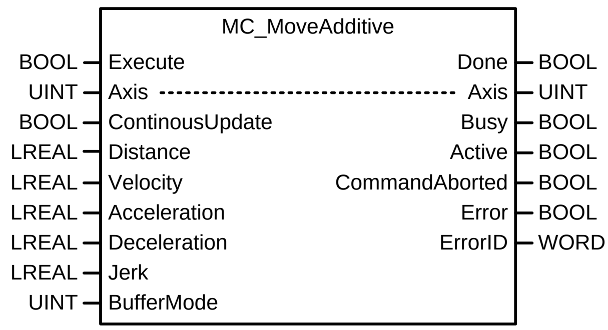
(a) Function block setting



(b) Timing diagram



6.3.7 Additive positioning operation (MC_MoveAdditive)

Motion Function Block		
		
Input-Output		
UINT	Axis	Specify the axis to be commanded (1~32: real/virtual axis, 33~36: virtual 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. (Refer to 6.1.5.Changes in Parameters during Execution of Motion Function Block)
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	Indicate whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

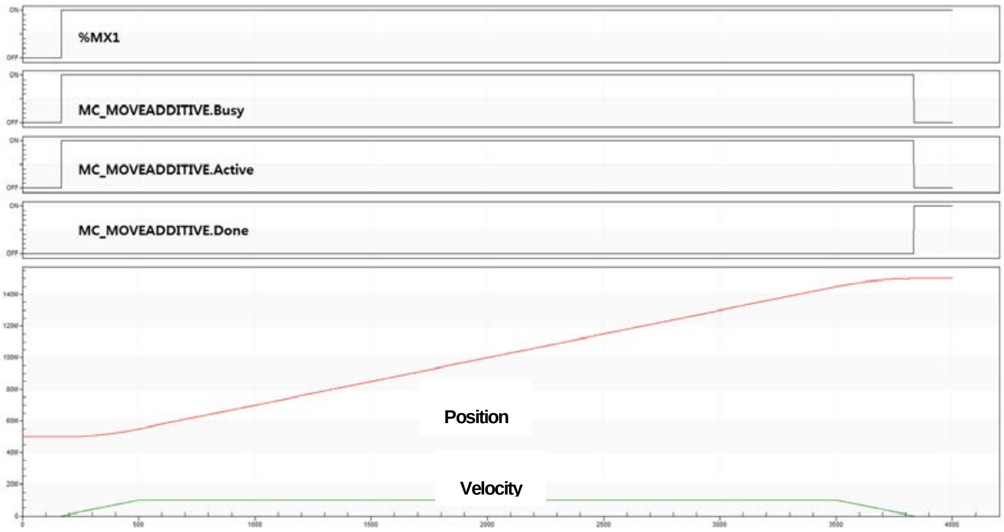
(1) This motion function block is to give the relevant additive position operation commands.
(2) Additive position motion (MC_MoveAdditive) is the motion function block which additionally moves as far as the position specified in Distance input from the final target position of the currently running 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) Moving direction is decided depending on the sign of the specified target distance in Distance input, and positive (+ or No sign) moving direction leads to forward direction, and negative (-) moving direction leads to reverse direction.
(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) The changed parameters can be applied by re-executing the function block (Execute input is On) before the command is completed. 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

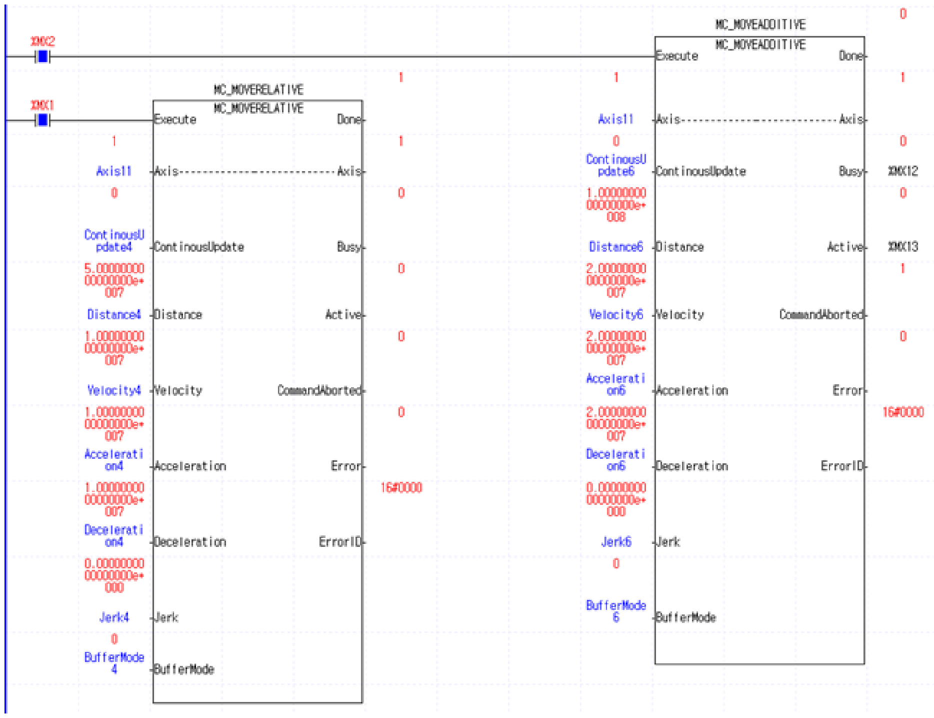
(b) Timing diagram



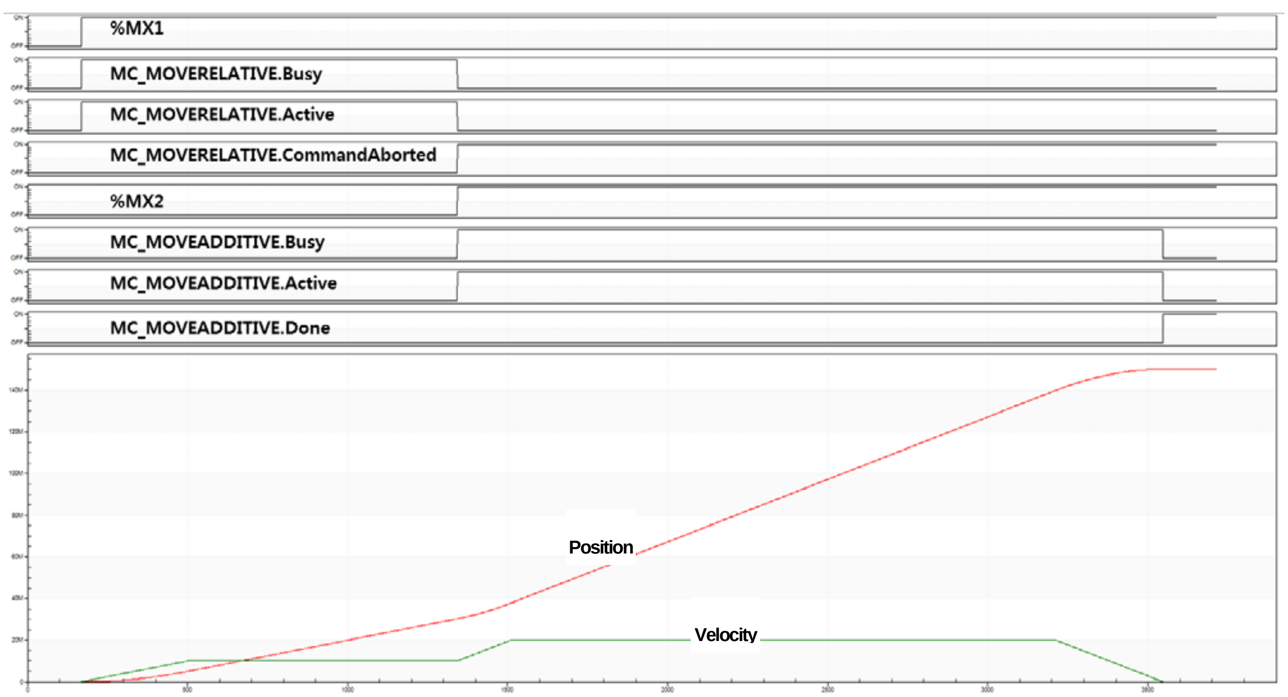
(10) Application example program

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		

```
                 MC_MoveVelocity
BOOL  ─ Execute                        InVelocity ─ BOOL
UINT  ─ Axis ----------------------------- Axis ─ UINT
BOOL  ─ ContinuousUpdate                     Busy ─ BOOL
LREAL ─ Velocity                           Active ─ BOOL
LREAL ─ Acceleration             CommandAborted ─ BOOL
LREAL ─ Deceleration                        Error ─ BOOL
LREAL ─ Jerk                              ErrorID ─ WORD
UINT  ─ Direction
UINT  ─ BufferMode
```

Input-Output		
UINT	Axis	Specify the axis to be commanded (1~32: real/virtual axis, 33~36: virtual 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. (Refer to 6.1.5.Changes in Parameters during Execution of Motion Function Block)
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 speed. (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	Indicate 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) Giving a stop command or execution of other motion function block allow to interrupt specified velocity motion.
(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) Output InVelocity is On when the relevant axis reaches the specified speed, and it is Off when the specified speed operation is interrupted.

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

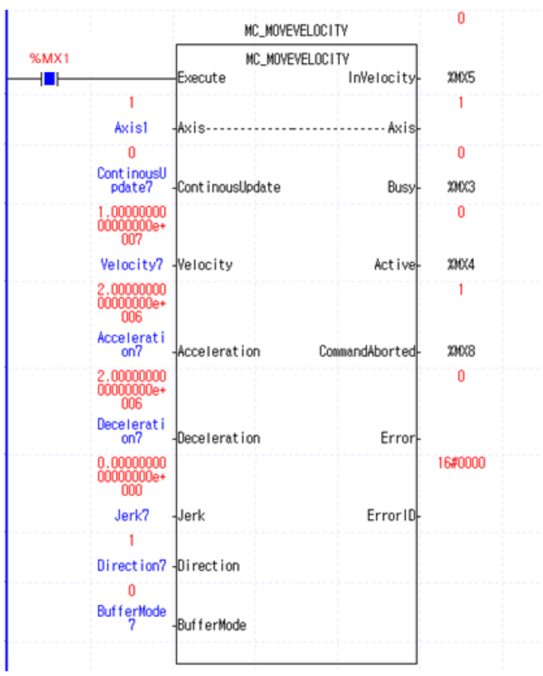
(7) The changed parameters can be applied by re-executing the function block (Execute input is On) before the command is completed. Only Distance, Velocity, Acceleration, Deceleration, Jerk, Direction input can be updated.

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

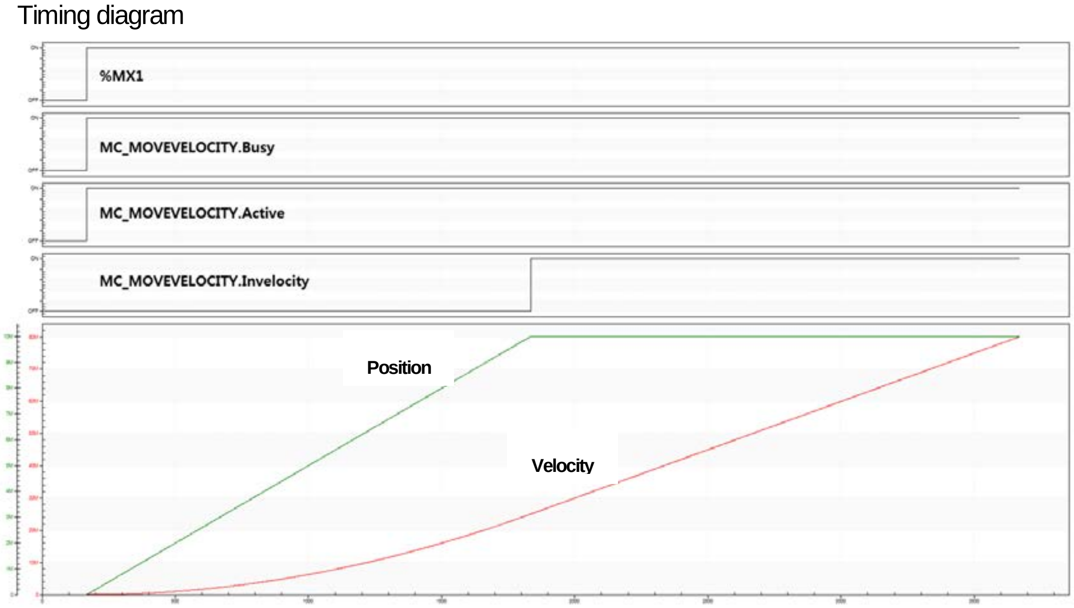
(9) Example program

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



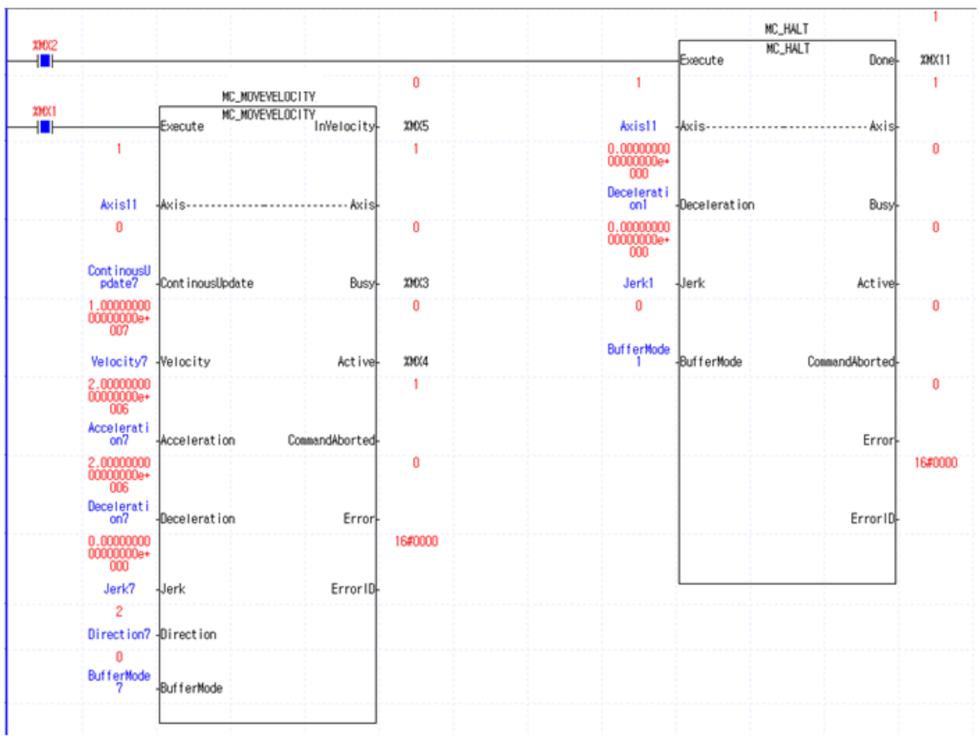
(b) Timing diagram



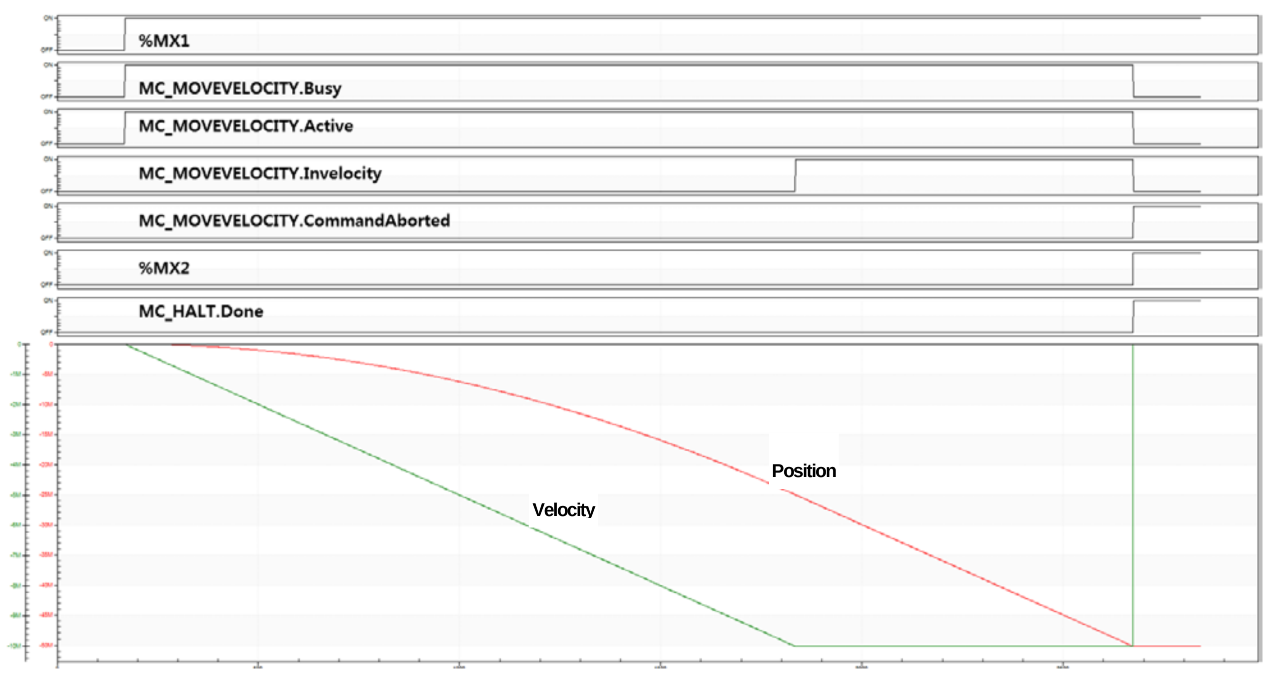
(10) Application example program

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.

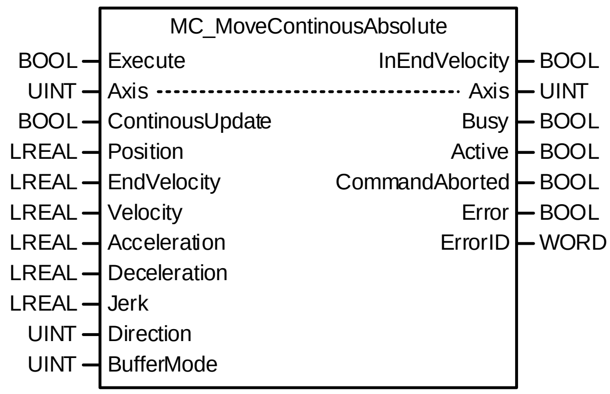
(a) Function block setting



(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 (1~32: real/virtual axis, 33~36: virtual 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. (Refer to 6.1.5.Changes in Parameters during Execution of Motion Function Block)
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. (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	Indicate whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

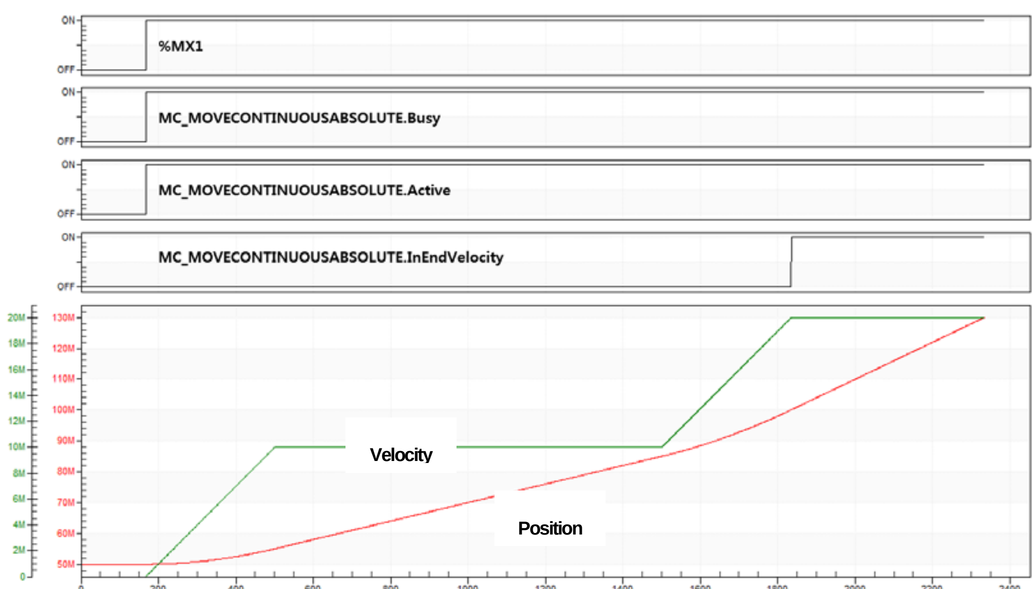
(1) This motion function block is to give Specified velocity operation after relative position operation command to the relevant axis.
(2) When executing MC_MoveContinuousAbsolute, the relevant axis moves to the position specified in Position and operates at the specified speed in EndVelocity if there 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) The changed parameters can be applied by re-executing the function block (Execute input is On) before the command is completed. 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, InEndVelocityoutput is on.

(a) Function block setting

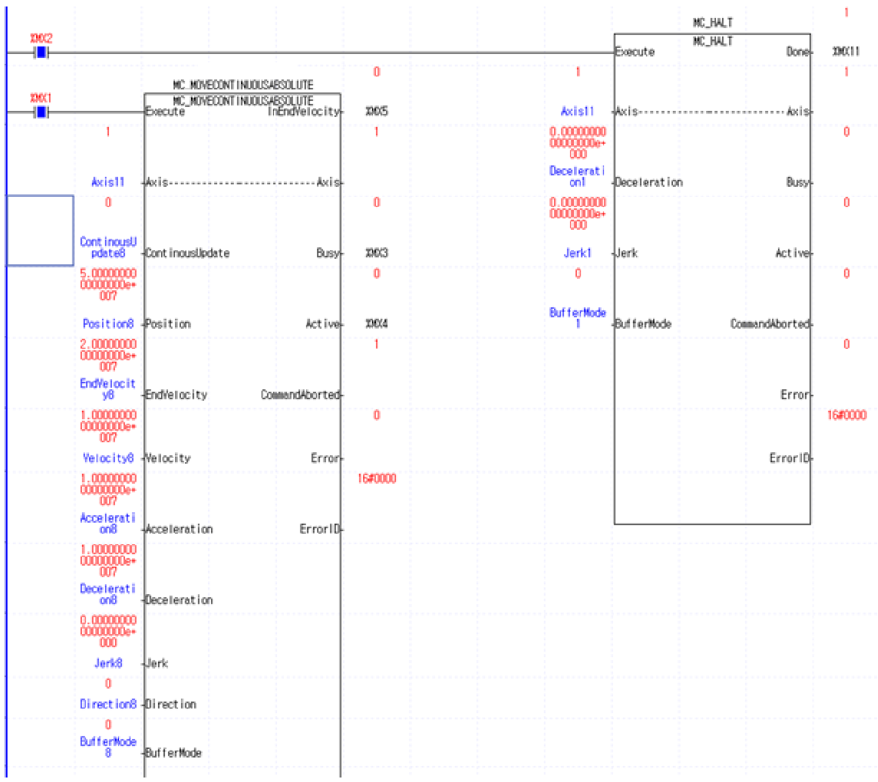
(b) Timing diagram



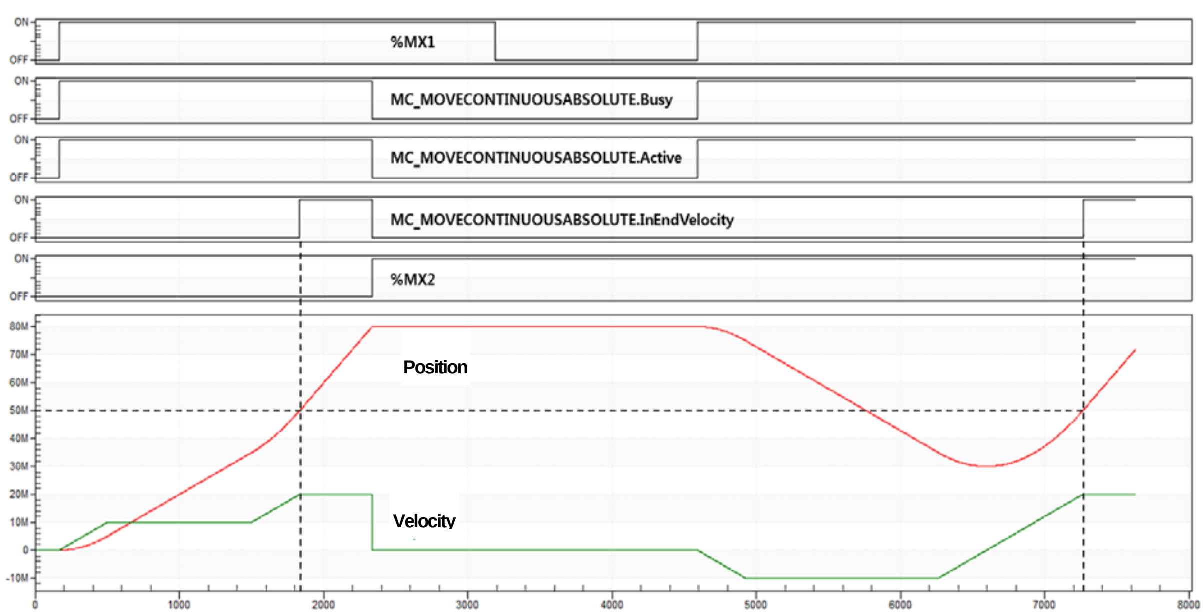
(10) Application example program

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

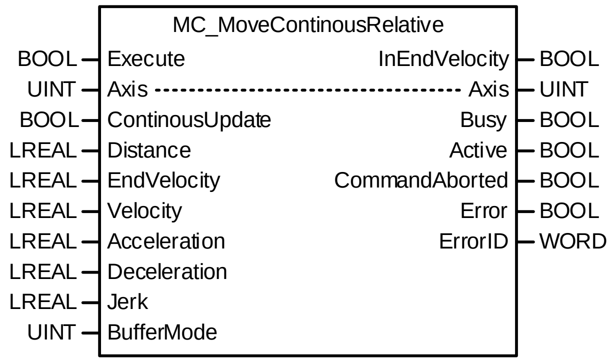


(b) Timing diagram



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 (1~32: real/virtual axis, 33~36: virtual axis)
Input		
BOOL	Execute	Give an absolute position motion 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.
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)
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	Indicate 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) The changed parameters can be applied by re-executing the function block (Execute input is On) before the command is completed. Only Distance, EndVelocity, Velocity, Acceleration, Deceleration, Jerk input can be updated. (However, in case of InEndVelocity=On, it is reflected only EndVelocity inputs.

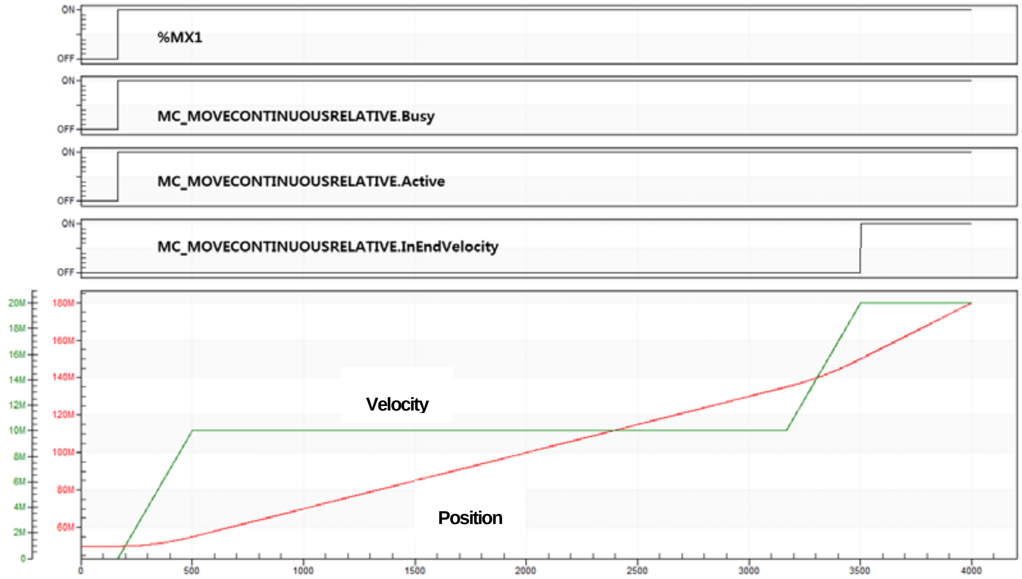
(7) Velocity and EndVelocityy input can be set to 0 or changed

(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 is on.

(a) Function block setting

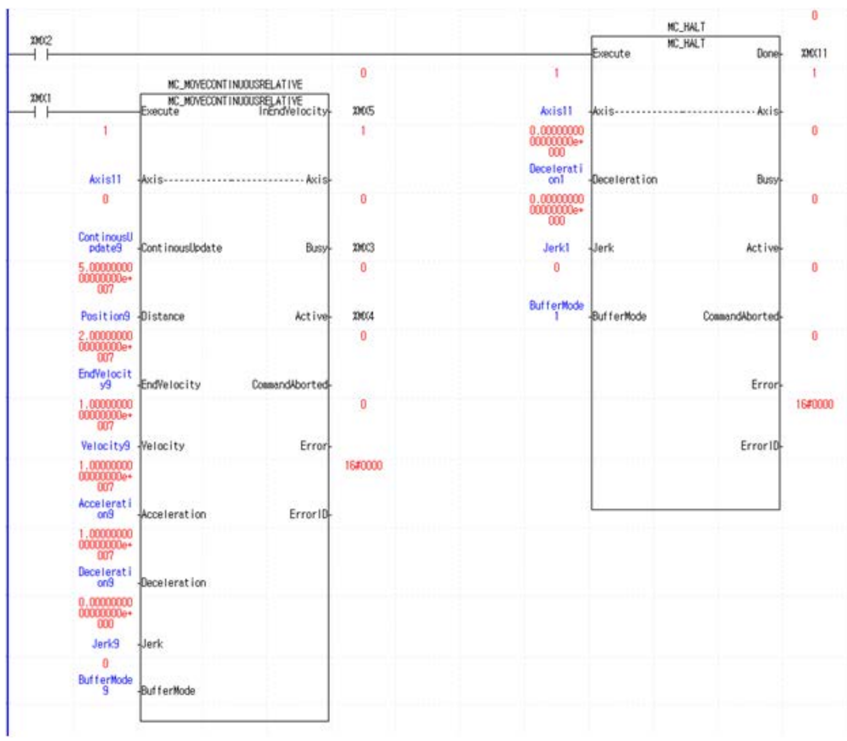
(b) Timing diagram



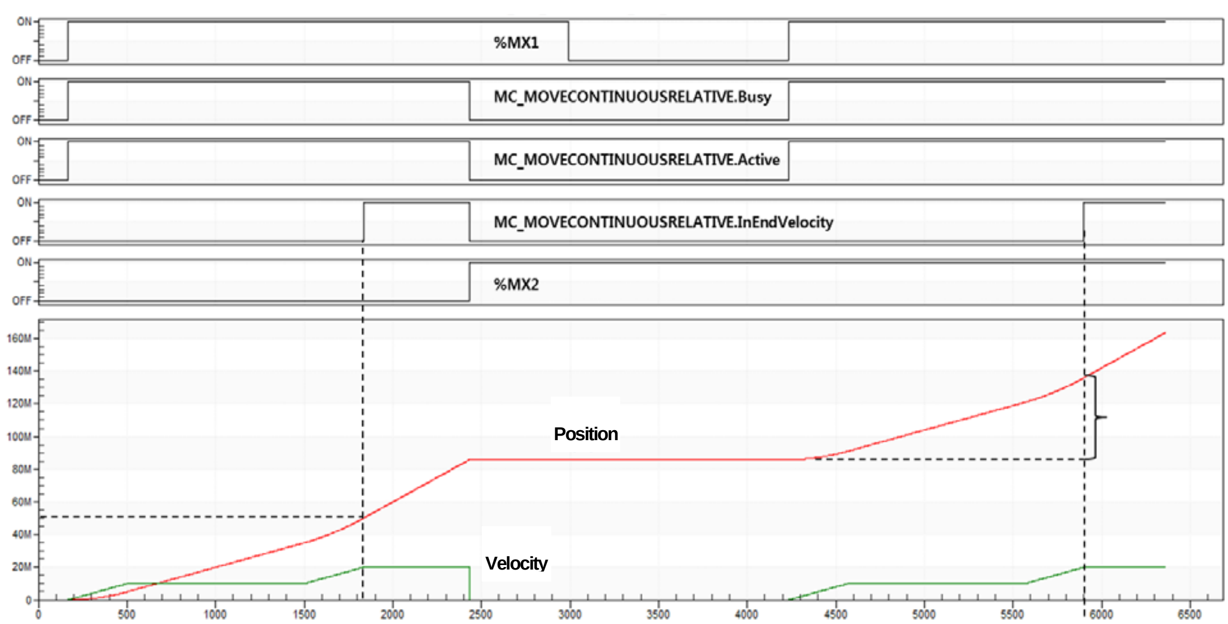
(9) Application example program

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

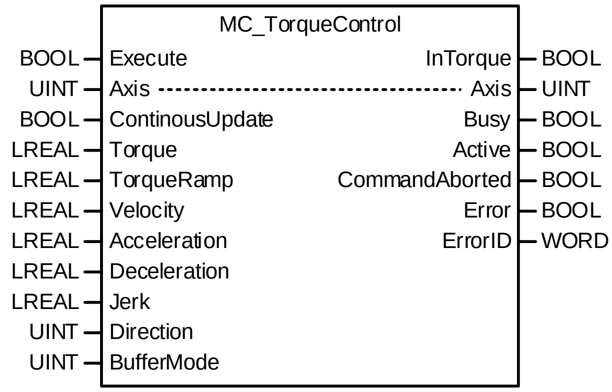


(b) Timing diagram



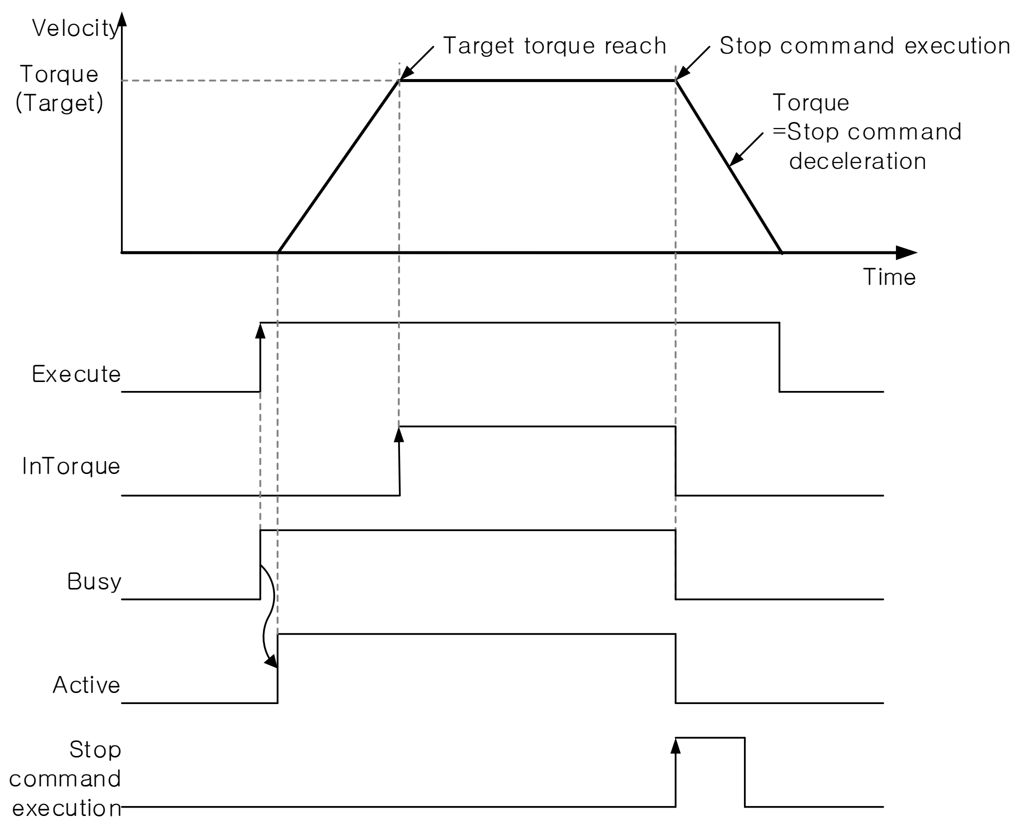
6.3.11 Torque control (MC_TorqueControl)

Motion Function Block		

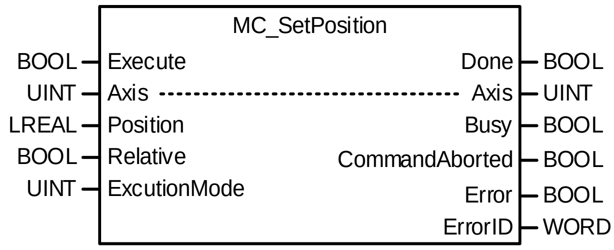


Input-Output		
UINT	Axis	Specify the axis to be commanded (1~32: real axis)
Input		
BOOL	Execute	Give an absolute position operation command to the relevant axis in the rising Edge.
BOOL	ContinuousUpdate	Specify the update setting of input value. (Refer to 6.1.5.Changes in Parameters during Execution of Motion Function Block)
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)
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	Indicate 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 when this motion function block is running.
(8) The changed parameters can be applied by re-executing the function block (Execute input is On) before the command is completed. 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 (1~32: real/virtual axis, 33~36: virtual axis)
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	ExecuteMode	0: Immediately applied the position value, 1: Applied at the same point with ‘Buffered’ of Buffermode
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	CommandAborted	Indicate that the current motion function block is interrupted while it is running.
BOOL	Error	Indicate 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) ExcutionMode 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”.

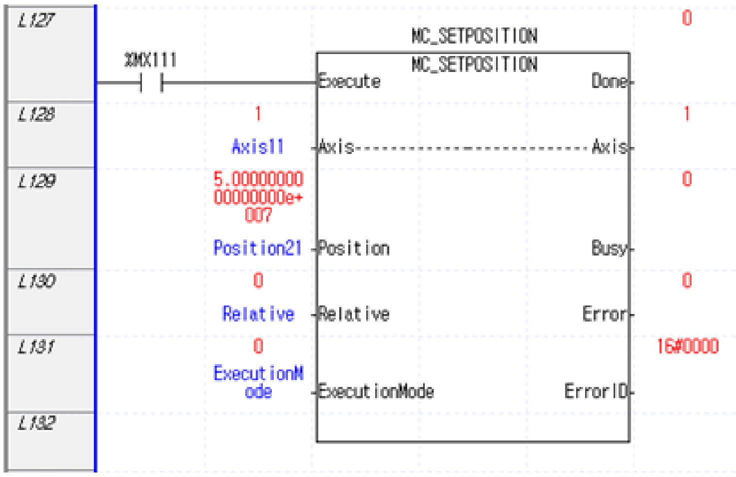
0 (mcImmediately): Change the parameter value immediately after executing function block (rising Edge in Execute input). If the relevant axis is in running, operation can be affected.

1 (mcQueued): Changed at the same point with ‘Buffered’ in Buffermode. (Refer to 6.1.4 Buffermode input)

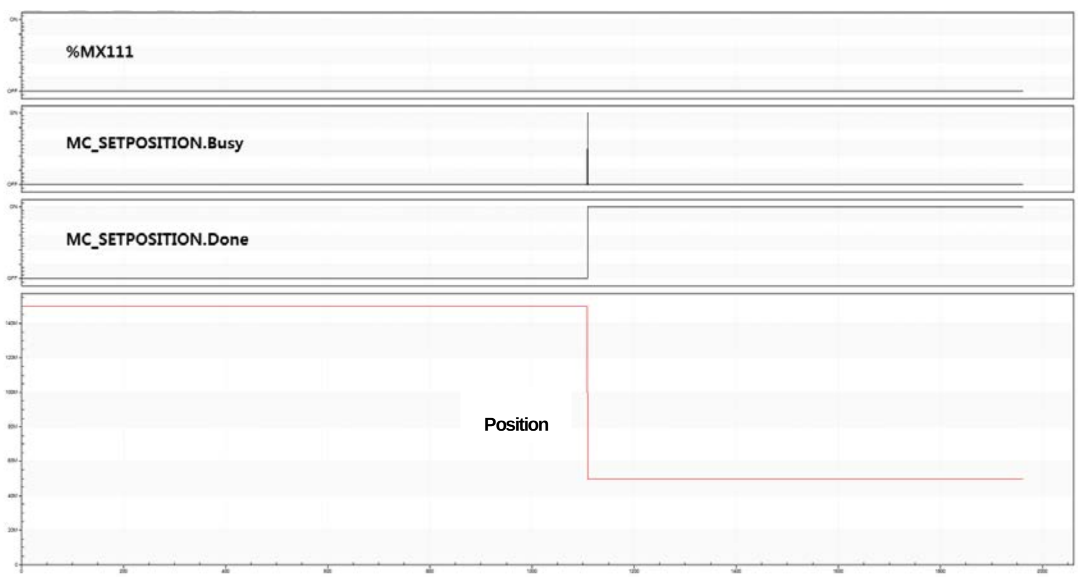
(4) Example program

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.

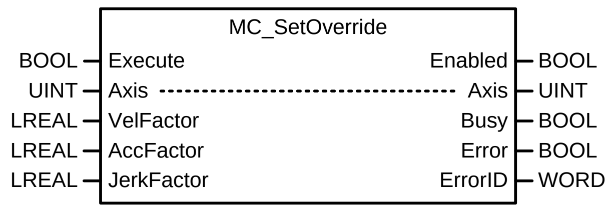
(a) Function block setting



(b) Timing diagram

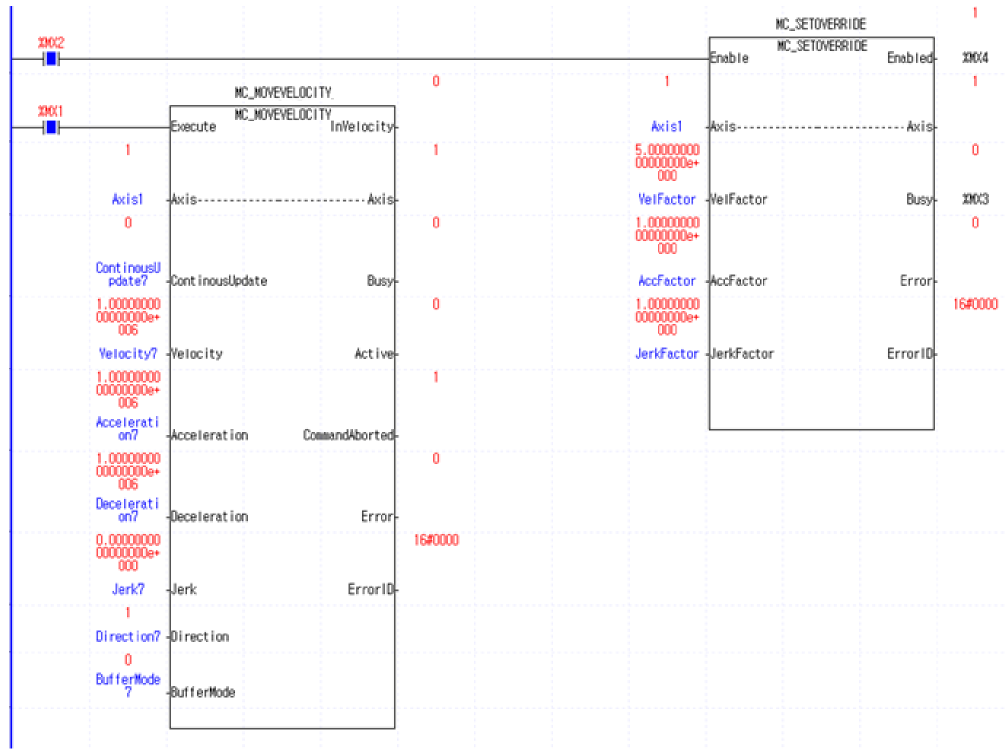


6.3.13 Velocity/Acceleration override (MC_SetOverride)

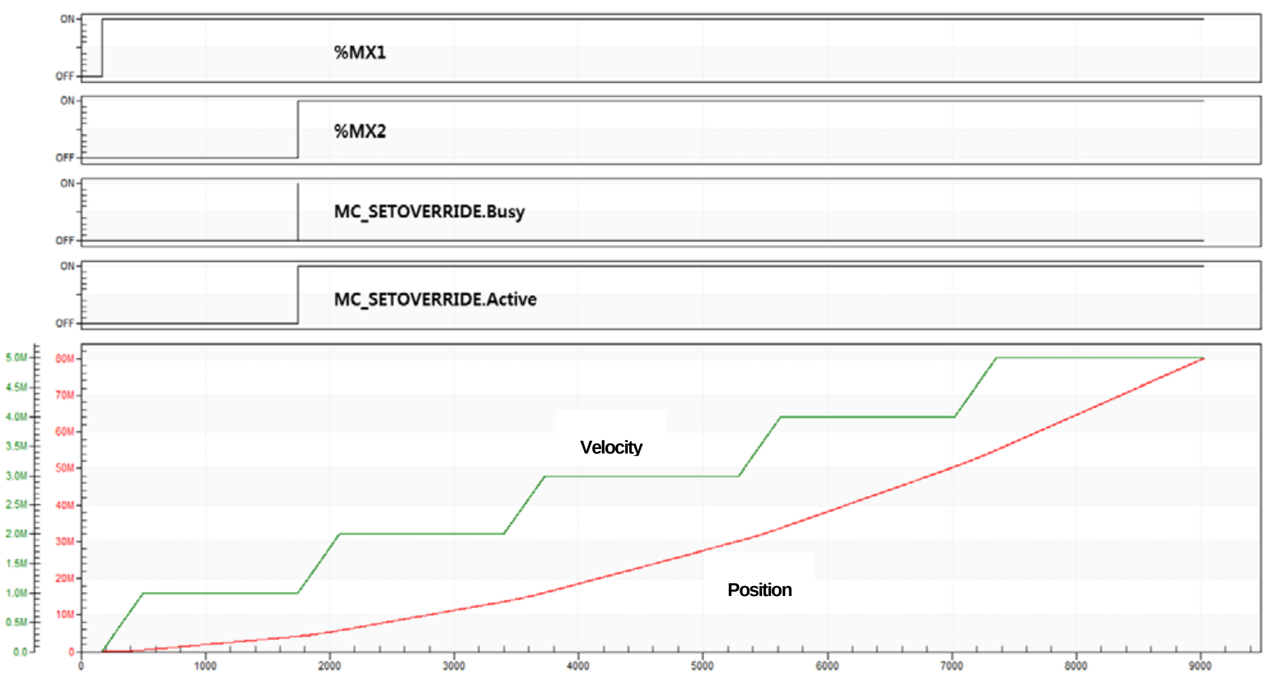
Motion Function Block		
		
Input-Output		
UINT	Axis	Specify the axis to be commanded (1~32: real/virtual axis, 33~36: virtual axis)
Input		
BOOL	Enable	Execute override operation in the relevant axis while input is activated.
LREAL	VelFactor	Specify the override rate of speed.
LREAL	AccFactor	Specify the override rate of acceleration/deceleration.
LREAL	JerkFactor	Specify the override rate of the change rate of acceleration.
Output		
BOOL	Enabled	Indicate that override rate is successfully applied.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicate 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) Speed override rate is specified in 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.

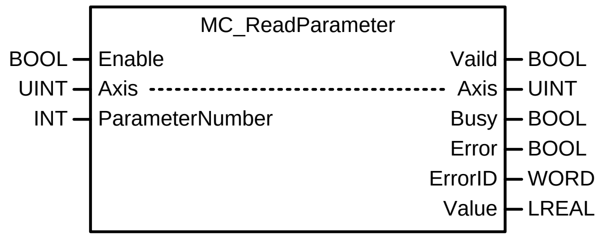
(a) Function block setting



(b) Timing diagram



6.3.14 Read parameter (MC_ReadParameter)

Motion Function Block		
		
Input-Output		
UINT	Axis	Specify the axis to be commanded (1~32: real/virtual axis, 33~36: virtual axis)
Input		
BOOL	Enable	Execute override operation in the relevant axis while input is activated.
INT	ParameterNumber	Specify the number of parameter to read. (0 ~ 25)
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	Indicate whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.
LREAL	Value	Output the value of parameter.

(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	Description
0	Basic Parameter	Unit	0:pulse,1:mm,2:inch,3:degree
1		Purses per rotation	1 ~ 4,294,967,295 [pulse]
2		Travel per rotation	0.000000001 ~ 4,294,967,295 [Unit]
3		Speed command unit	0:Unit/Time, 1:rpm
4		Speed limit	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		Operating mode of the reverse rotation	0:E.Stop, 1:Stop
10	Extended Parameter	S/W upper limit	LREAL [Unit]
11		S/W lower limit	LREAL [Unit]
12		Infinite running repeat position	LREAL Positive number [Unit]
13		Infinite running repeat	0:Disable, 1:Enable
14		Command Inposition range	0 or LREAL Positive number [Unit]
15		Tracking error over-range value	0 or LREAL Positive number [Unit]
16		Current position compensation amount	0 or LREAL Positive number [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:Don't detect, 1:Detect
20		Tracking error level	0:Warning, 1:Alarm
21		JOG high Speed	LREAL Positive number [Unit] (Jog low speed ~speed limit) [Unit/s]
22		JOG low Speed	LREAL Positive number [Unit] (< 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^2]
26		Override mode	0: Specified by ratio, 1: Specified by unit
27	NC Parameter	Identifying range to reach the spindle rotation command speed	0~100%
28		Identifying RPM to reach the spindle rotation zero speed	0~100rpm

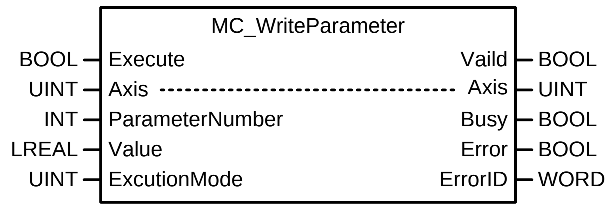
No	Parameter	Item	Description
100	Encoder Parameter	Encorder1 unit	0: pulse, 1: mm, 2: inch, 3:degree
101		Encorder1 pulse per rotation	1 ~ 4294967295
102		Encorder1 travel per rotation	0.000000001 ~ 4294967295
103		Encorder1 pulse input	0:CW/CCW 1 multiplier, 1:PULSE/DIR 1 multiplier 2:PULSE/DIR 2 multiplier, 3:PHASE A/B 1 multiplier 4:PHASE A/B 2 multiplier, 5: PHASE A/B 4multiplier
104		Encorder1 max. value	(Encoder1 min. value + 1) ~ 2147483647
105		Encorder1 min. value	-2147483648 ~ (Encoder1 max. value - 1)
106		Encoder1 input filter value	0: No use, 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		Encorder2 unit	0: pulse, 1: mm, 2: inch, 3:degree
201		Encorder2 pulse per rotation	1 ~ 4294967295
202		Encorder2 travel per rotation	0.000000001 ~ 4294967295
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 4multiplier
204		Encorder2 max. value	(Encoder2 min. value + 1) ~ 2147483647
205		Encorder2 min. value	-2147483648 ~ (Encoder2 max. value - 1)
206		Encoder2 input filter value	0: No use, 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

*Remak1) LREAL range: 2.2250738585072e-308 ~ 1.79769313486232e+308

LREAL positive range: 0 ~ 1.79769313486232e+308 (Excluded 0)

6.3.15 Write parameter (MC_WriteParameter)

Motion Function Block



Input-Output		
UINT	Axis	Specify the axis to be commanded (1~32: real/virtual axis, 33~36: virtual axis)
Input		
BOOL	Execute	Rising Edge corresponding parameters of input is written. .
INT	ParameterNumber	Specify the number of parameter to write. (0 ~ 25)
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	Indicate 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) Parameter is written in the rising Edge of 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 the chapter 6.1.4 BufferMode input)

(6) The numbers of parameter are as below.

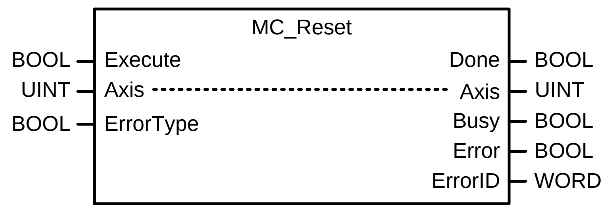
No	Parameter	Item	Description
0	Basic Parameter	Unit	0:pulse,1:mm,2:inch,3:degree
1		Purses per rotation	1 ~ 4,294,967,295 [pulse]
2		Travel per rotation	0.000000001 ~ 4,294,967,295 [Unit]
3		Speed command unit	0:Unit/Time, 1:rpm
4		Speed limit	LREAL Positive number [Unit/s, rpm] (Change according to 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		Operating mode of the reverse rotation	0:E.Stop, 1:Stop
10	Extended Parameter	S/W upper limit	LREAL [Unit]
11		S/W lower limit	LREAL [Unit]
12		Infinite running repeat position	LREAL Positive number [Unit]
13		Infinite running repeat	0:Disable, 1:Enable
14		Command Inposition range	0 or LREAL Positive number [Unit]
15		Tracking error over-range value	0 or LREAL Positive number [Unit]
16		Current position compensation amount	0 or LREAL Positive number [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:Don't detect, 1:Detect
20		Tracking error level	0:Warning, 1:Alarm
21		JOG high Speed	LREAL Positive number [Unit] (Jog low speed ~speed limit) [Unit/s]
22		JOG low Speed	LREAL Positive number [Unit] (< 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^2]
26		Override mode	0: Specified by ratio, 1: Specified by unit
27	NC Parameter	Identifying range to reach the spindle rotation command speed	0~100%
28		Identifying RPM to reach the spindle rotation zero speed	0~100rpm

No	Parameter	Item	Description
100	Encoder Parameter	Encorder1 unit	0: pulse, 1: mm, 2: inch, 3:degree
101		Encorder1 pulse per rotation	1 ~ 4294967295
102		Encorder1 travel per rotation	0.000000001 ~ 4294967295
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 4multiplier
104		Encorder1 max. value	(Encoder1 min. value + 1) ~ 2147483647
105		Encorder1 min. value	-2147483648 ~ (Encoder1 max. value - 1)
106		Encorder1 input filter value	0: No use, 1: 500kPPS 2: 200kPPS, 3: 100kPPS 4: 10kPPS, 5: 1kPPS 6: 0.1kPPS
200		Encorder2 unit	0: pulse, 1: mm, 2: inch, 3:degree
201		Encorder2 pulse per rotation	1 ~ 4294967295
202		Encorder2 travel per rotation	0.000000001 ~ 4294967295
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 4multiplier
204		Encorder2 max. value	(Encoder2 min. value + 1) ~ 2147483647
205		Encorder2 min. value	-2147483648 ~ (Encoder2 max. value - 1)
206		Encorder2 input filter value	0: No use, 1: 500kPPS 2: 200kPPS, 3: 100kPPS 4: 10kPPS, 5: 1kPPS 6: 0.1kPPS

*Remak1) LREAL range: 2.2250738585072e-308 ~ 1.79769313486232e+308

LREAL positive range: 0 ~ 1.79769313486232e+308 (Excluded 0)

6.3.16 Reset axis error (MC_Reset)

Motion Function Block		
		
Input-Output		
UINT	Axis	Specify the axis to be commanded (1~32: real/virtual axis, 33~36: virtual axis)
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	Indicate whether the axis error is successfully reset.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicate whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

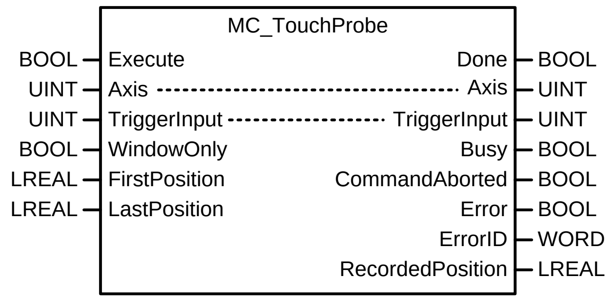
(1) This motion function block is to reset the error of the relevant axis. When setting ErrorType to '0' and executing motion function block in case the relevant axis is in ' ErrorStop' state, every axis error is reset and the axis state is switched to 'StandStill' or 'Disabled' state.

(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)

Motion Function Block



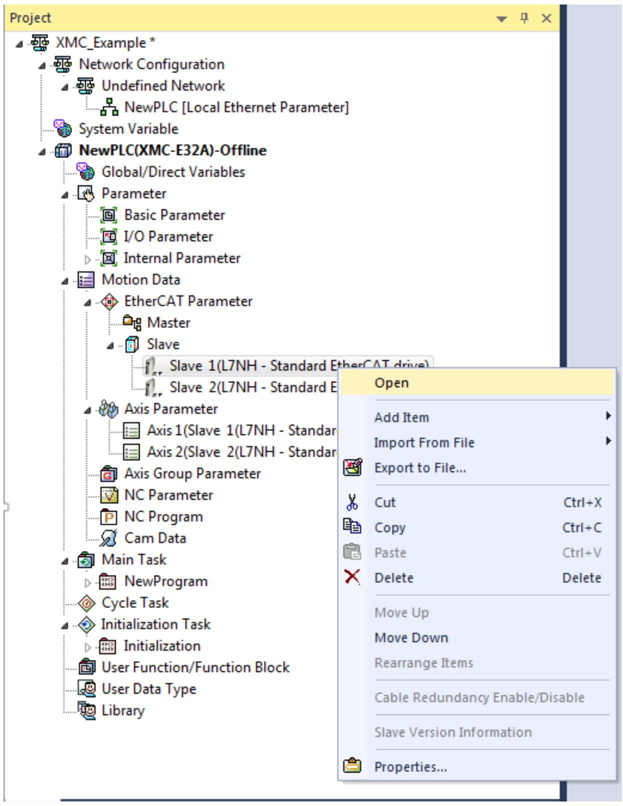
Input-Output		
UINT	Axis	Specify the axis to be commanded (1~32: real axis)
UINT	TriggerInput	Specify the signal to be used as a trigger. (0: TouchProbe 1, 1: TouchProbe 2)
Input		
BOOL	Execute	TouchProbe function starts at the rising Edge of input.
BOOL	WindowOnly	Activate the window mode.
LREAL	FirstPosition	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 by other command.
BOOL	Error	Indicate 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 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) 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.

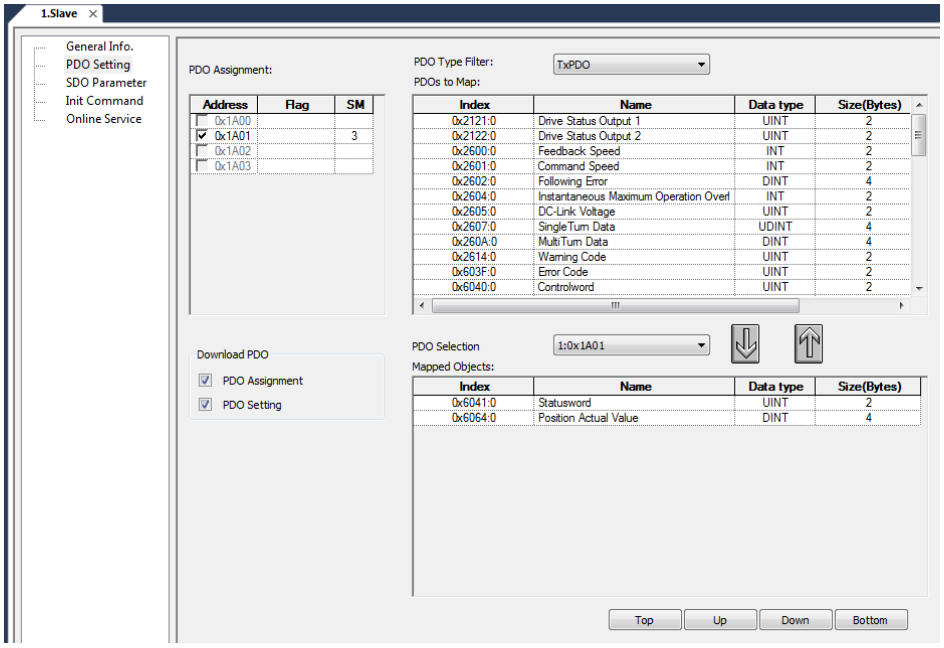
Note

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 function (0x60B8)

      Touch Probe status (0x60B9)

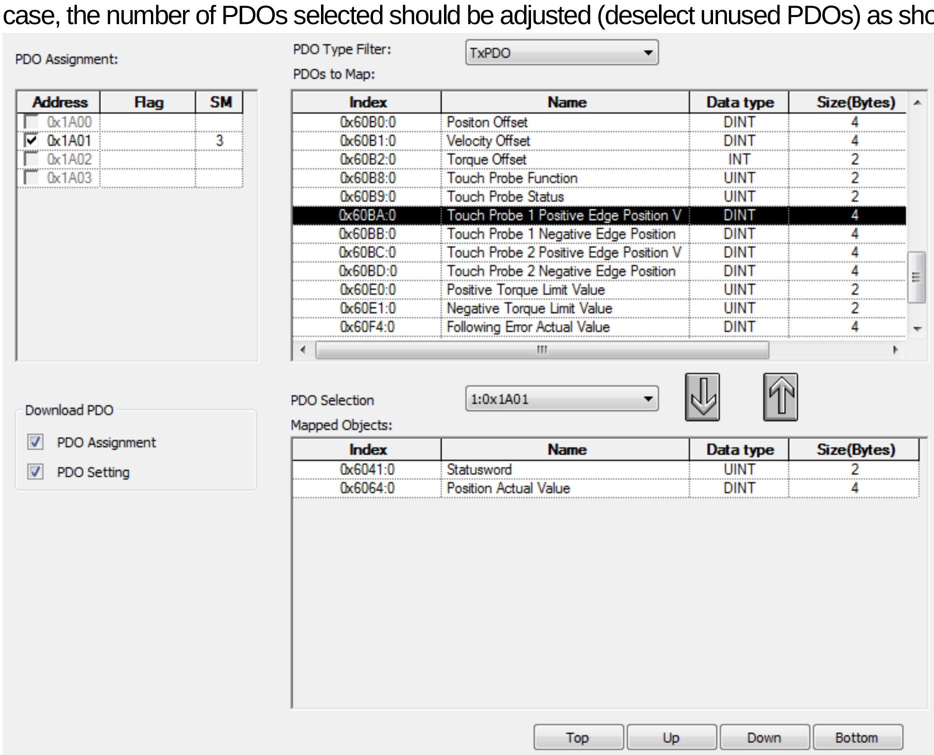
      Touch Probe 1 forward direction position value (0x60BA)

      Touch Probe 1 backward direction position value (0x60BB)

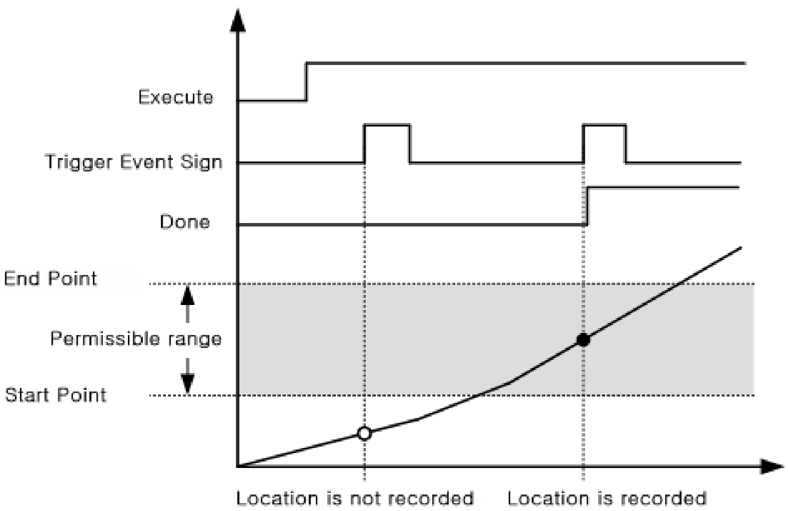
      Touch Probe 2 forward direction position value (0x60BC)

      Touch Probe 2 backward(0x60BD)

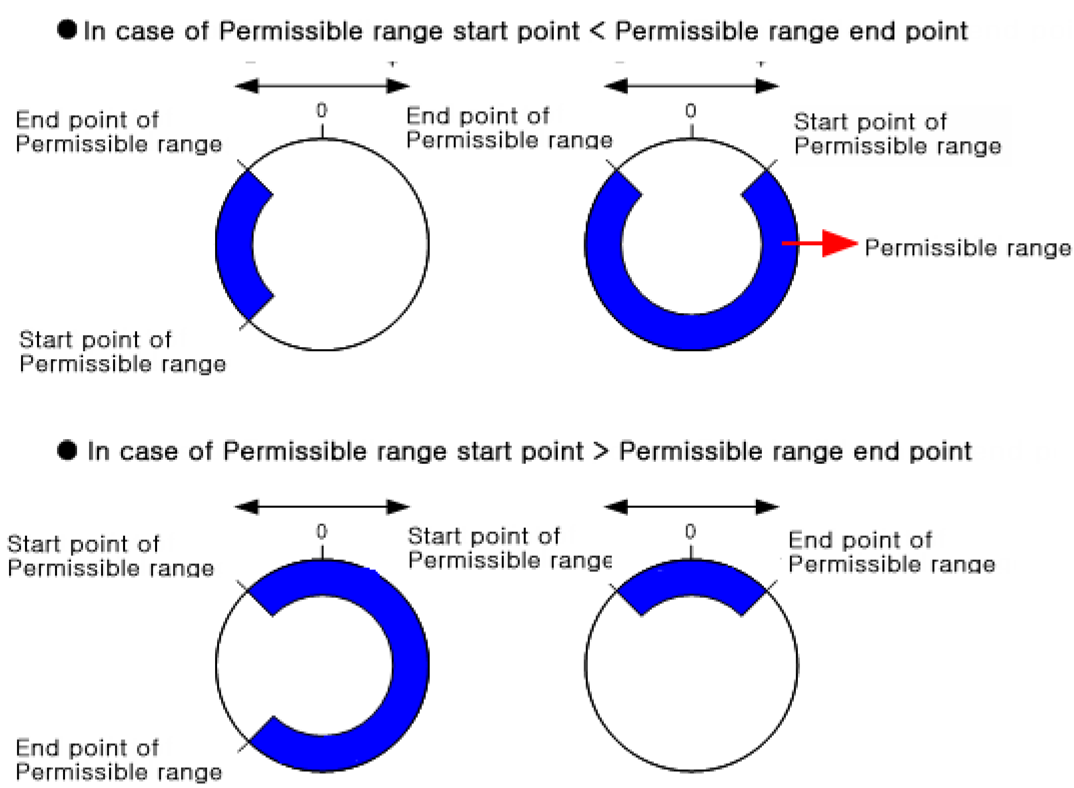
   At the PDO edit window, select the forward direction position value for Touch Probe 2, and select the down arrow. For some servo drive, a PDO setting error (0xF22) may occur, preventing connection to the servo drive. In such a case, the number of PDOs selected should be adjusted (deselect unused PDOs) as shown on the right.



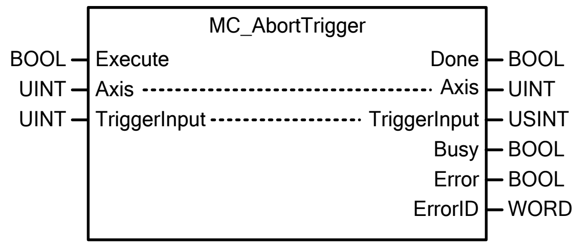
4. After PDO item is edited, must write ‘EtherCAT parameter’ in motion controller.



< In case Touch Probe function is the window mode, Operation timing >



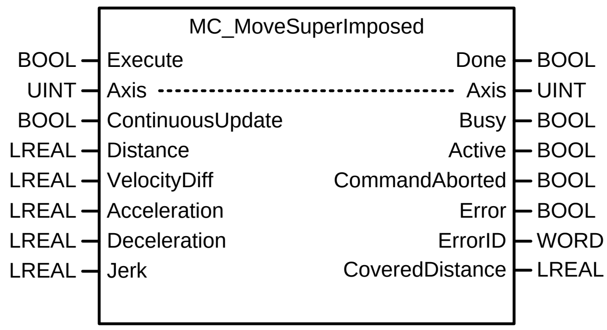
6.3.18 Abort trigger events (MC_AbortTrigger)

Motion Function Block		
		
Input-Output		
UINT	Axis	Specify the axis to be commanded (1~32: real axis)
UINT	TriggerInput	Specify the trigger signal to be disengaged. (0: TouchProbe 1, 1: TouchProbe 2)
Input		
BOOL	Execute	The trigger on standby in the relevant axis in the rising Edge is disengaged.
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	Error	Indicate 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		



Input-Output		
UINT	Axis	Specify the axis to be commanded (1~32: real/virtual axis, 33~36: virtual axis)
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 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	Indicate whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion 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.

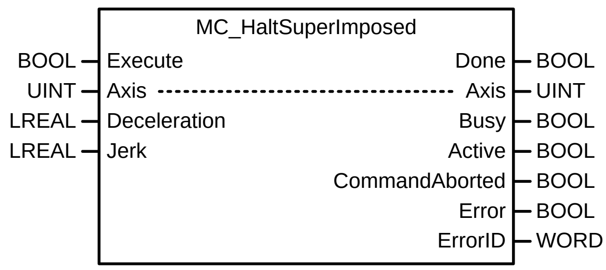
(2) SuperImposed is a command ordering to move from the current position at the time of the command to the target distance set by Distance input.

(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 halt (MC_HaltSuperImposed)

Motion Function Block		



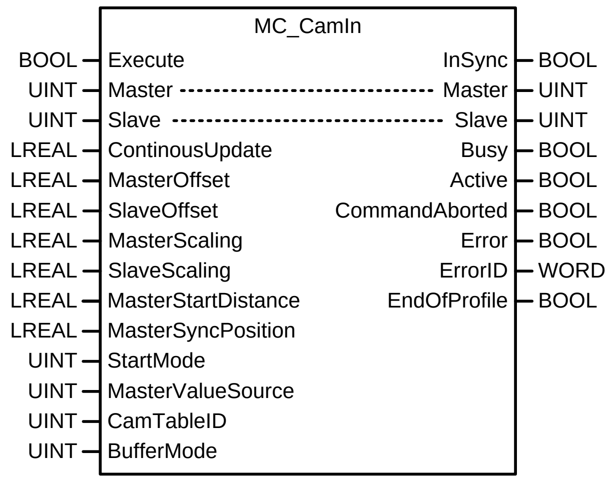
Input-Output		
UINT	Axis	Specify the axis to be commanded (1~32: real/virtual axis, 33~36: virtual axis
Input		
BOOL	Execute	Give a SuperImposed operation halt command to the relevant axis in the rising Edge.
LREAL	Deceleration	Specify deceleration in time of stop. [u/s^2]
LREAL	Jerk	Specify the change rate of acceleration/deceleration. [u/s^3]
Output		
BOOL	Done	Indicate that the speed of the relevant axis reaches 0.
BOOL	Busy	Indicate that the execution of 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	Output the number of error occurred while motion function block is running.
WORD	ErrorID	Indicate the distance moved with SuperImposed operation after SuperImposed command.

(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.

6.4 Multi-Axis Motion Function Block

6.4.1 Camming run (MC_CamIn)

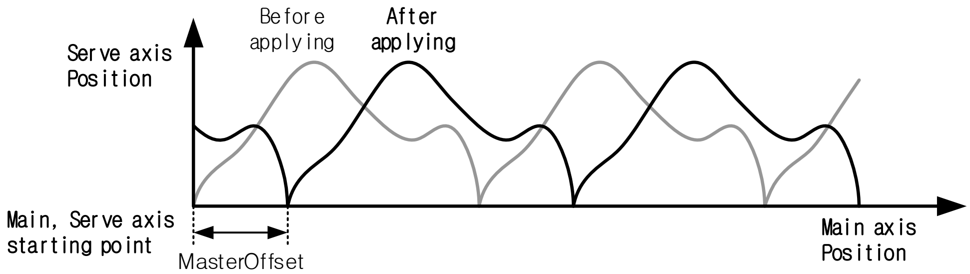
Motion Function Block

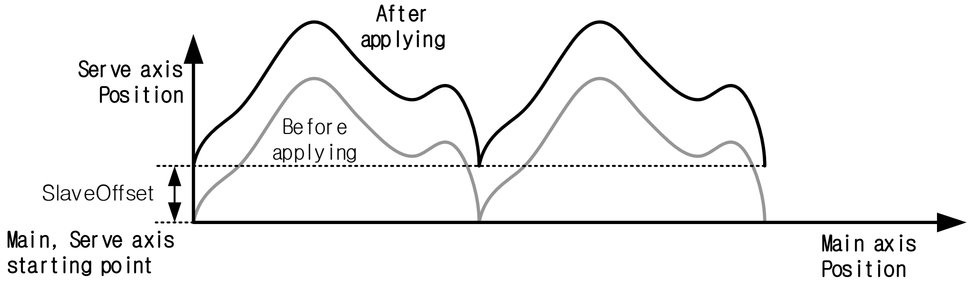


Input-Output		
UINT	Master	Set the main axis. (1~32: real/virtual axis, 33~36: virtual axis1~32: real/virtuall axis, 33~36: virtual axis, 1001~1002: encoder)
UINT	Slave	Set the serve axis. (1~32: real/virtual axis, 33~36: virtual axis)
Input		
BOOL	Execute	Give cam operation command to the relevant axis 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 serve axis cam table.
LREAL	MasterScaling	Specify the magnification of the main axis.
LREAL	SlaveScaling	Specify the magnification of the serve axis cam table.
LREAL	MasterStartDistance	Specify the position of the main axis where cam operation of the slave.
LREAL	MasterSyncPosition	Specify the starting point at cam table when cam operation starts.

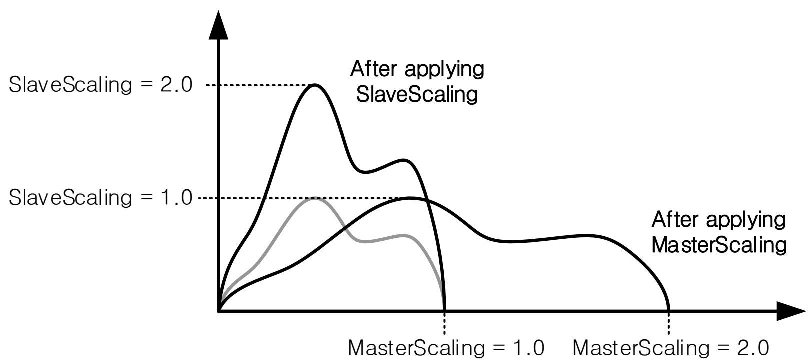
UINT	StartMode	Set the cam operation mode. 0 : Cam table is applied as an absolute value (mcAbsolute) 1: Cam table is applied as a relative value based on the command starting point (mcRelative)
UINT	MasterValueSource	Select the source of the main axis for cam operation. 0 : Synchronized in the target value of the main axis. 1 : Synchronized in the current value of the serve axis.
UINT	CamTableID	Specify the cam table to operate.
UINT	BufferMode	Specify the sequential operation setting of motion function block. (Refer to 6.1.4.BufferMode)
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	Indicate 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 operate the serve axis cam depending on the main axis.
(2) Cam operation command can be given to the serve axis even if the main axis is in stop state.
(3) You must give cam operation abort (MC_CamOut) command to the serve axis or operate other motion function block to stop cam operation.
(4) 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.
(5) 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.
(6) The axis is in 'Synchronized Motion' while this motion function block is running.
(7) Set the offset of cam table to be applied in MasterOffset and SlaveOffset. MasterOffset sets the offset with the starting point of the main axis, and SlaveOffset sets the offset with the starting point of the serve axis. Refer to the Figure below.

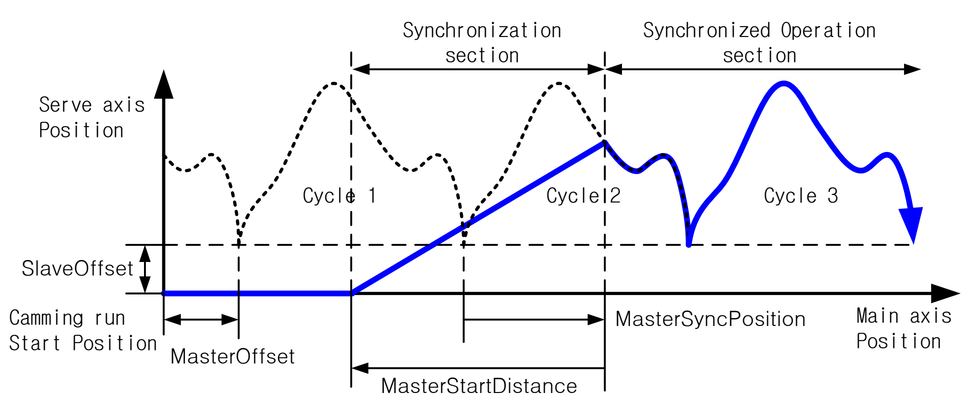




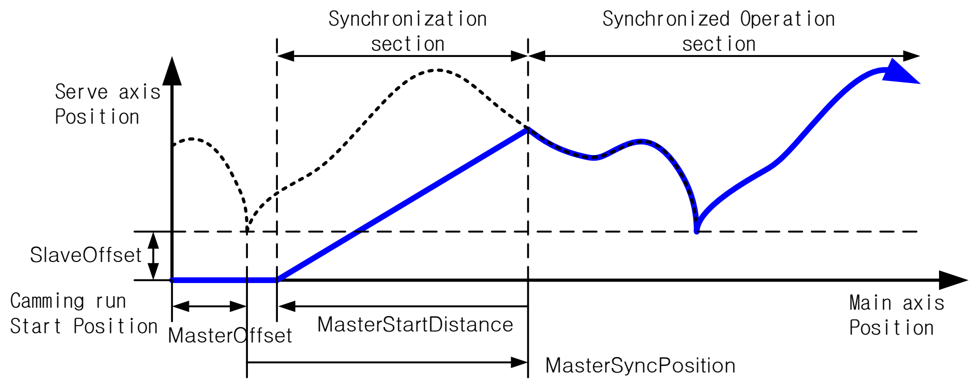
(8) Set the magnification of cam data to be applied in MasterScaling and SlaveScaling. Set the magnification of the main axis data in MasterScaling, and set the magnification of the serve axis data. Refer to the Figure below.



(9) MasterSyncPosition input specifies the position of the main axis within the table where the synchronization of actual cam operation is completed, and MasterStartDistance input specifies the relative position of the main axis where the synchronization starts. 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.



< In case MasterScaling is 1.0 >



< In case MasterScaling is 2.0 >

MasterSyncPosition position is based on the position within the cam table, and actual synchronization position is decided by considering MasterOffset and MasterScale parameters.

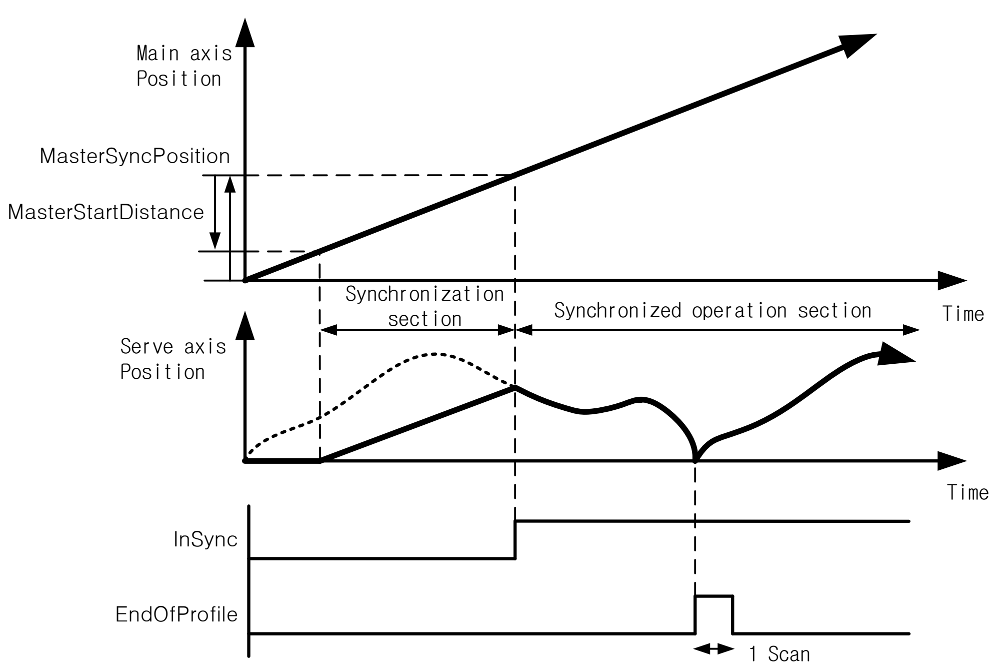
The serve axis starts moving to the synchronization position from the distance of the input value away based on the position where MasterSyncPosition is actually applied. If it is before starting moving, the serve axis waits at the relevant position in stop state, and if the serve axis is already in the section to move to the synchronization position at the beginning of the command, takes back the position of the synchronization starting point by the length of a table until it escapes the MasterStartDistance range.

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.

(10) If the ContiunuousUpdate input is On, the changed parameter can be applied.

Only MasterOffset, SlaveOffset, MasterScaling, SlaveScaling, MasterStartDistance, MasterSyncPosition can be updated (However, In InSync=On case, MasterOffset, SlaveOffset, MasterScaling, SlaveScaling can be updated.

(11) Once cam operation starts normally, InSync output is On, and EndOfProfile output is 1 scan On every time one cam table operation is completed.



(12) Cam operation mode is set in StartMode. Setting range is 0 or 1, and the input value outside the setting range causes an error.

(13) MasterValueSource selects the source of the main axis to be synchronized. If it is set to 0, the serve axis performs cam operation based on the command position of the main axis which is calculated in motion controller, and if it is set to 1, the serve axis performs cam operation based on the current position which is received by communication in servo drive of main axis.

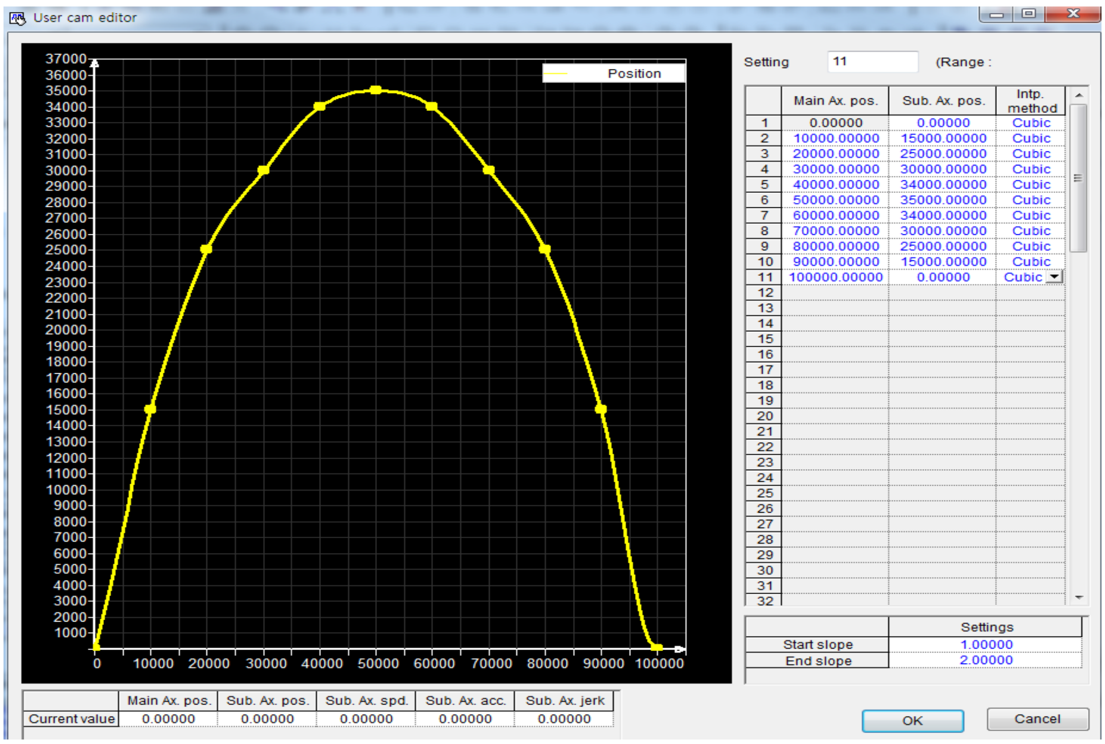
(14) CamTableID sets the number of cam table to be applied to cam operation. Setting range is 1~32, and the input value outside the setting range causes error "0x1115" in motion function block.

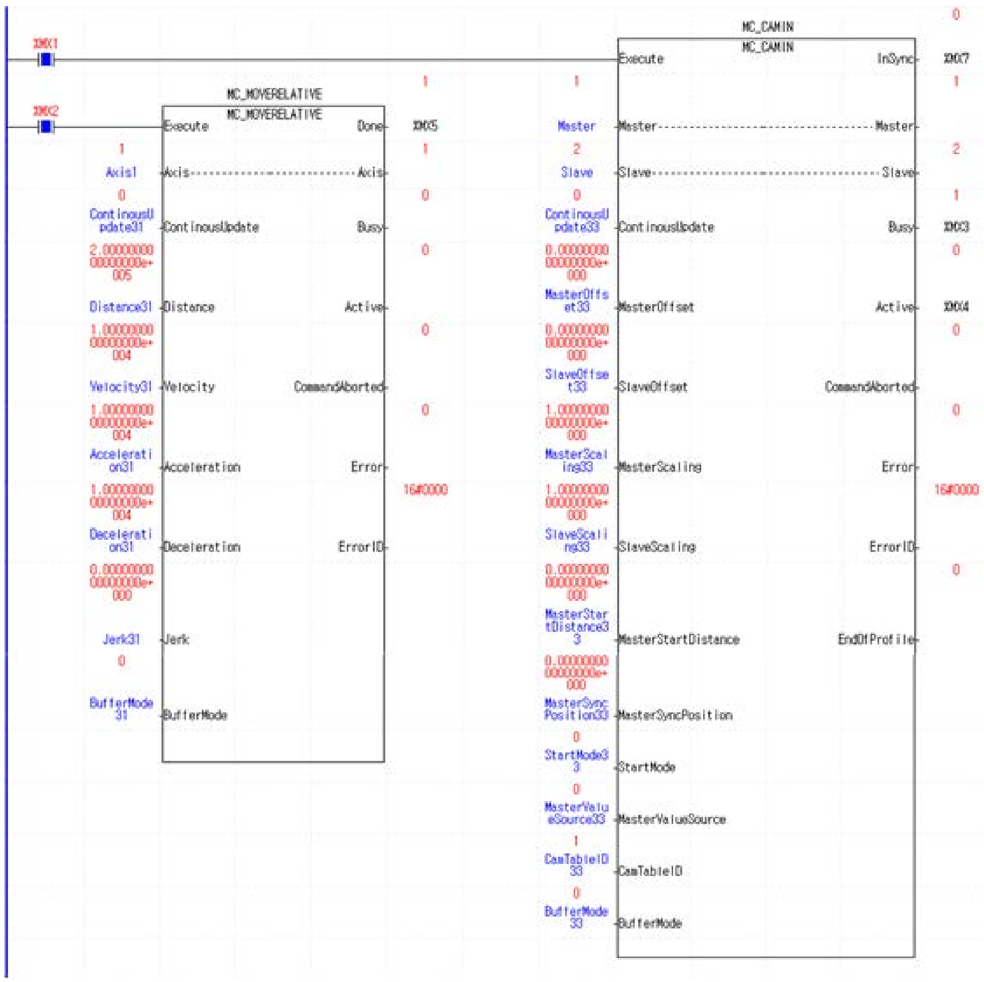
(15) The relevant axis is in "SynchronizedMotion" state while this motion function block is running.

(16) Example program

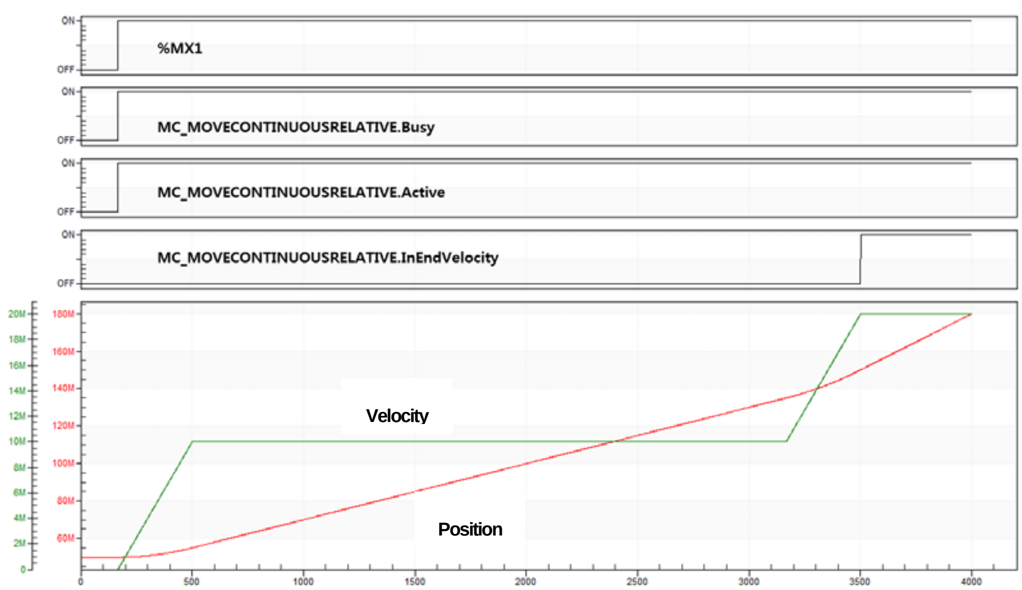
This example shows the movement of the main-axis from 0 to 200,000 positions after generating a cam profile and then executing MC_CAMIN command on the sub-axis.

(a) Function block setting





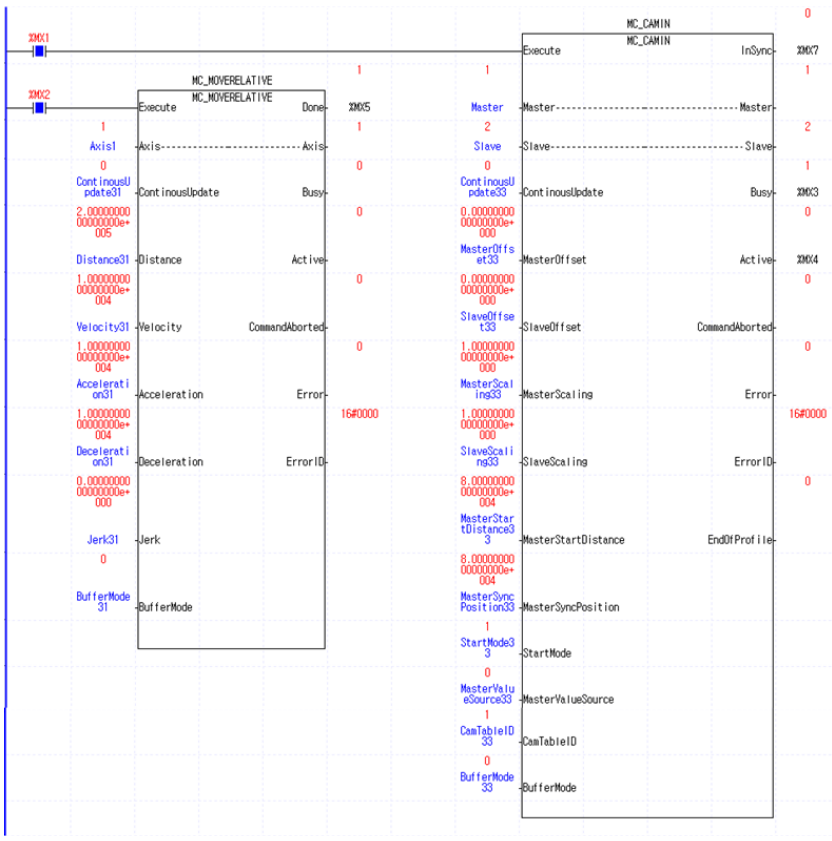
(b) Timing diagram



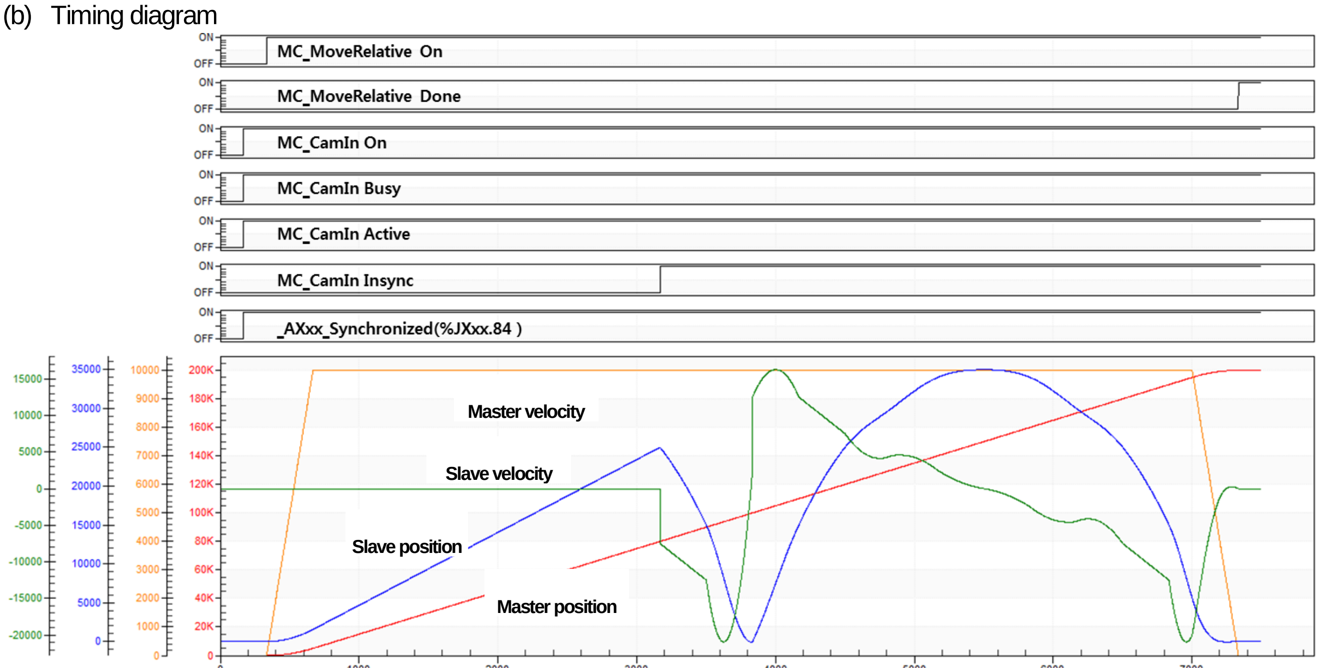
(17) Application example program

This example shows the movement of the main-axis from 0 to 200,000 positions after generating the same profile and then executing C_CAMIN command where MasterSyncPosition and MasterSyncDistance are set to 80,000 in sub-axis.

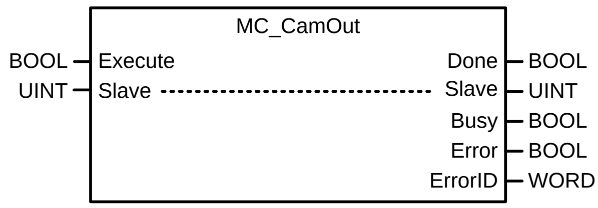
(a) Function block setting



(b) Timing diagram



6.4.2 Camming stop (MC_CamOut)

Motion Function Block		
		
Input-Output		
UINT	Slave	Set the serve axis. (1~32: real/virtual axis, 33~36: virtual axis)
Input		
BOOL	Execute	Give cam operation stop command to the relevant axis in the rising Edge.
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	Error	Indicate whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

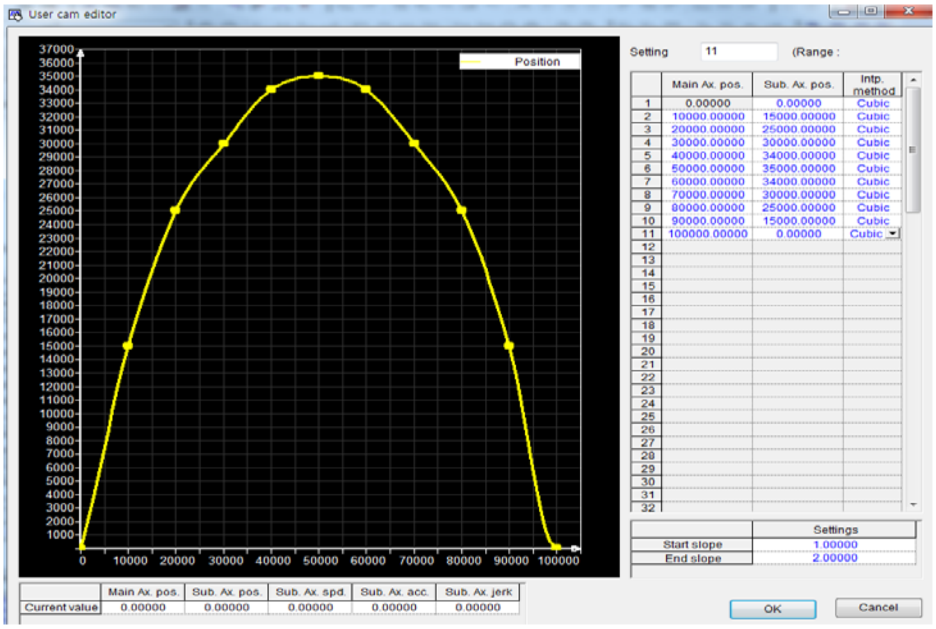
(1) This motion function block immediately disengages cam operation running in the serve axis.

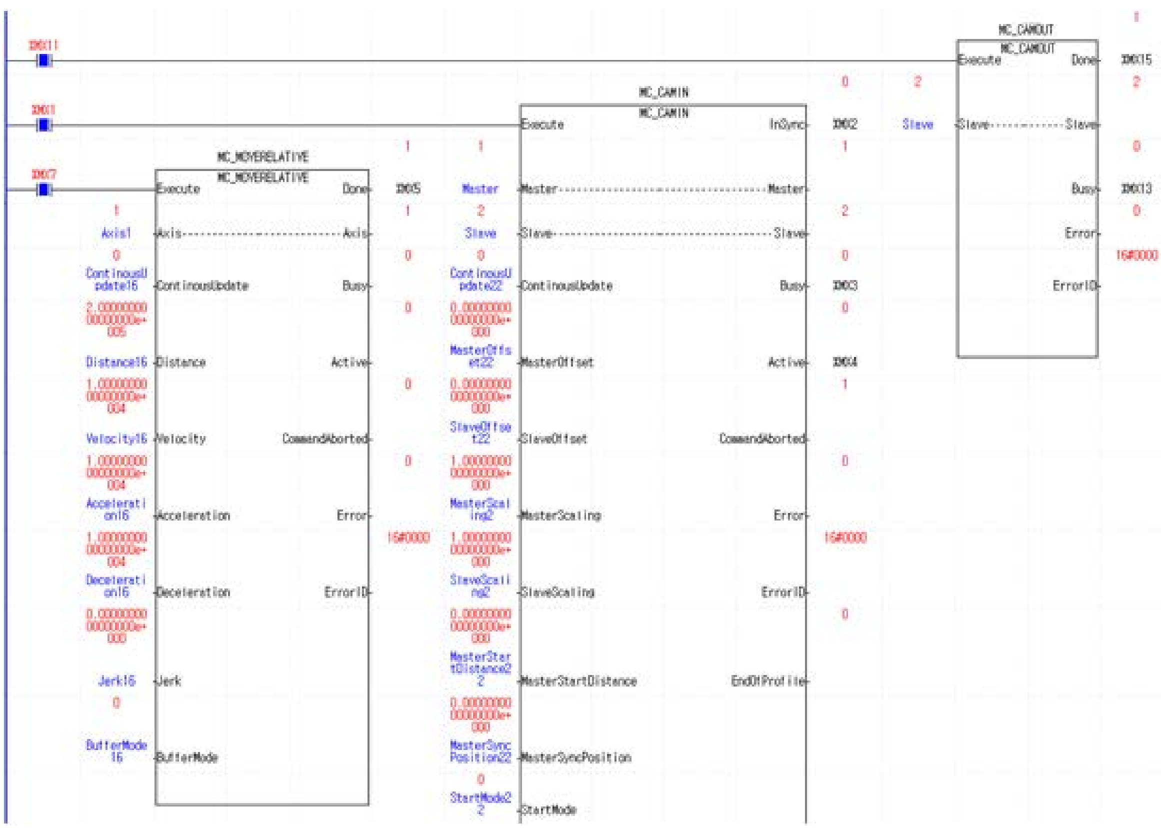
(2) If motion function block of which BufferMode is Aborting in the serve axis where cam operation is running, cam operation is automatically disengaged and the relevant motion function block is executed. To execute cam operation abort (MC_CamOut) motion function block, the relevant axis do operation 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.

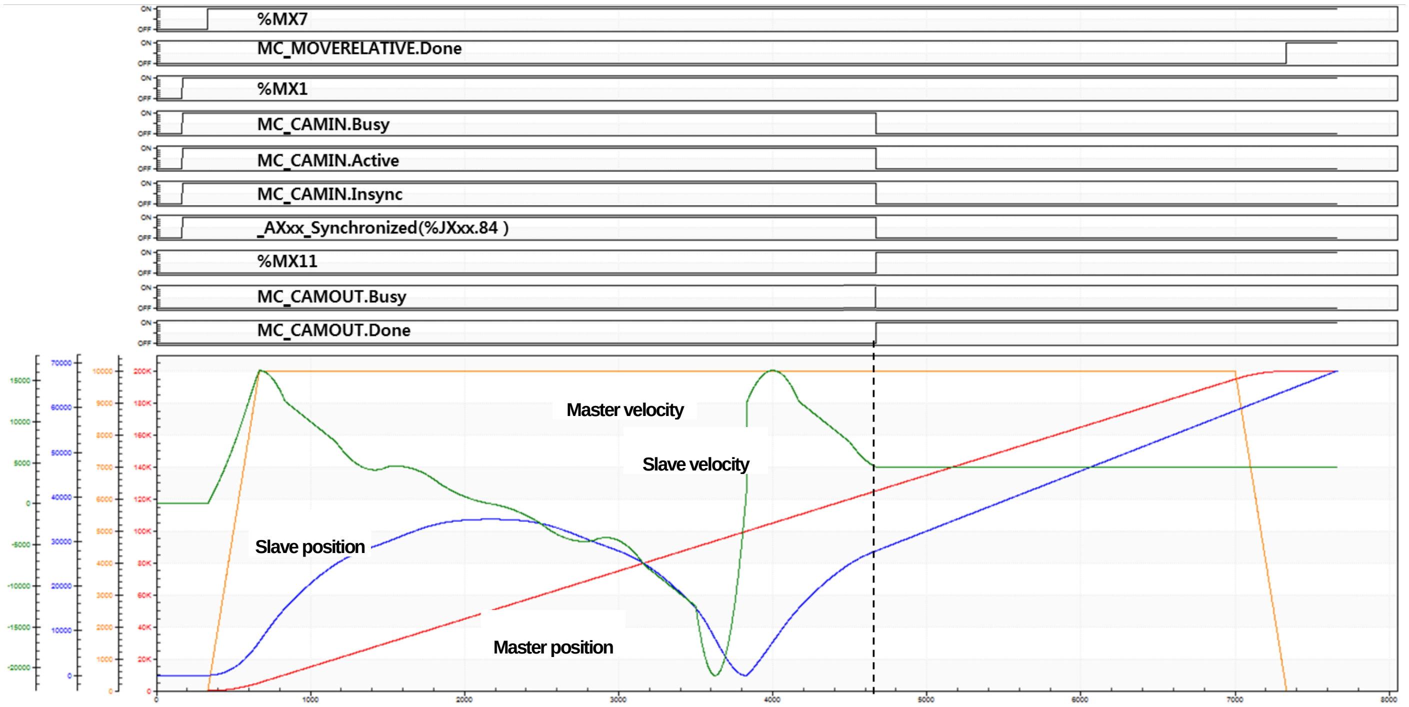
(4) Example program
This example shows generating a cam profile, executing MC_CAMIN command on the sub-axis, moving the main-axis to the 200,000 position, and then executing MC_CAMOUT. The sub-axis maintains the velocity at the time when the cam operation is terminated.

(a) Function block setting



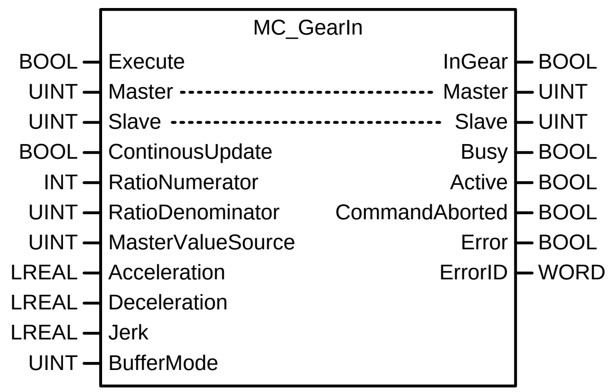


(b) Timing diagram



6.4.3 Electrical gearing run (MC_GearIn)

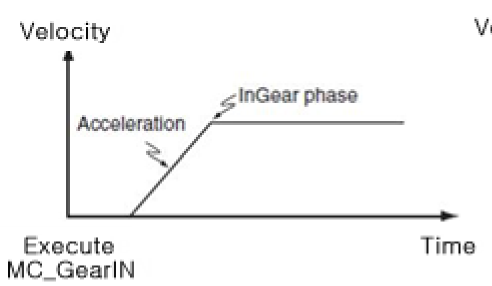
Motion Function Block

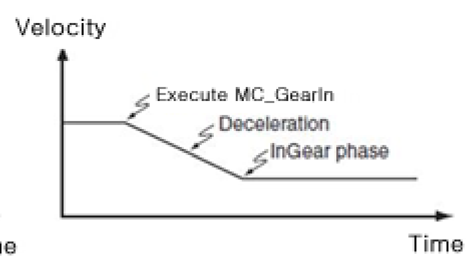


Input-Output		
UINT	Master	Set the main axis. (1~32: real/virtual axis, 33~36: virtual axis1~32: real/virtuall axis, 33~36: virtual axis, 1001~1002: encoder)
UINT	Slave	Set the serve axis. (1~32: real/virtual axis, 33~36: virtual axis)
Input		
BOOL	Execute	Give gear operation command to the relevant 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)

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	Indicate whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

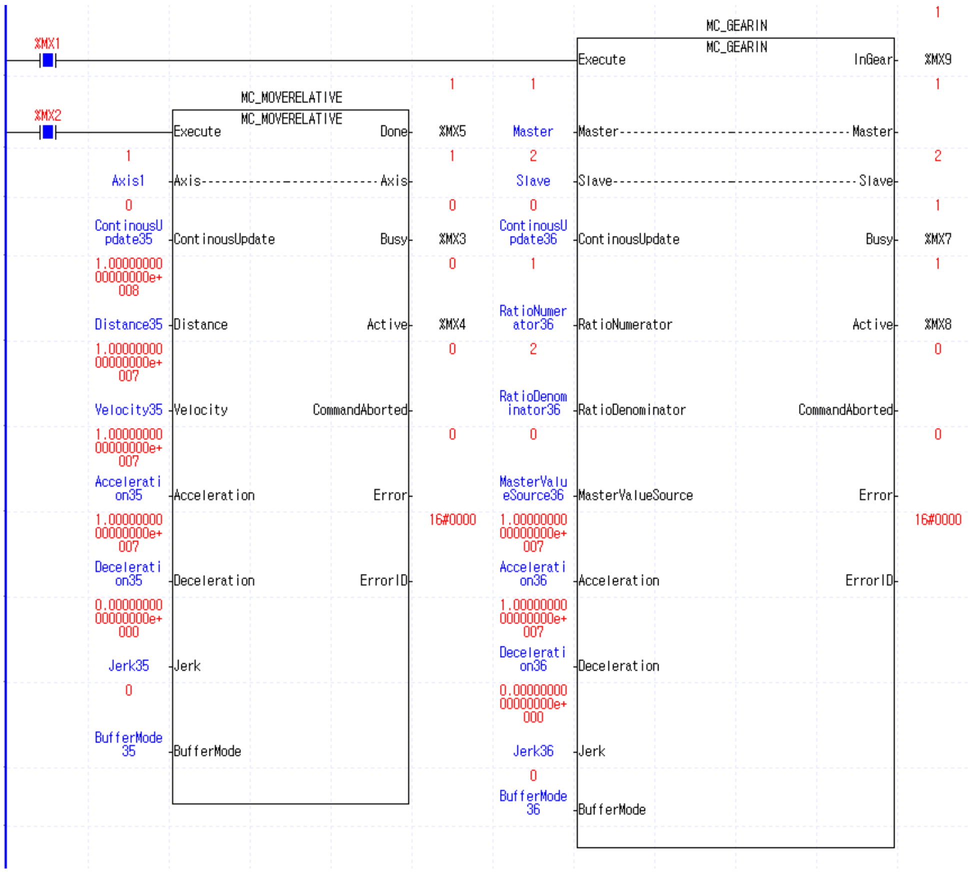
(1) This motion function block is an operation to synchronize the speed of the main axis and the serve axis depending on gear ratio which is set.
(2) Giving gear operation abort (MC_GearOut) commands to the relevant axis or execution of other motion function block allow disengaging gear operation.
(3) 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) 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.
(6) 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 the 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.
(7) When this motion function block is executed, the serve axis is synchronized with the main axis through acceleration/deceleration at the speed in synch with the relevant gear ratio.
(8) The serve axis is in 'SynchronizedMotion' while this motion function block is running.



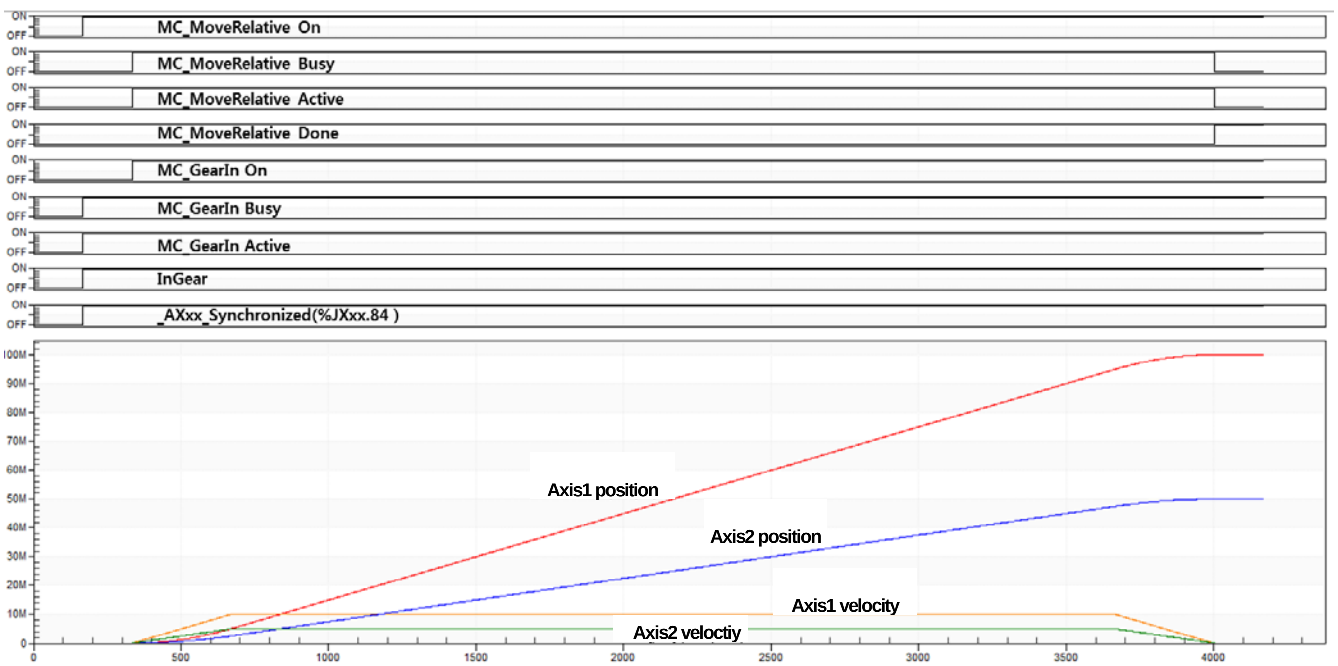


(9) The changed parameters can be applied when ContinuousUpdate input is On. Only RatioNumerator, RatioDenominator, Acceleration, Deceleration input can be updated. (However, in case of InGear=On case, RatioNumerator, RatioDenominator input can be updated)
(10) Example program
This example shows the 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

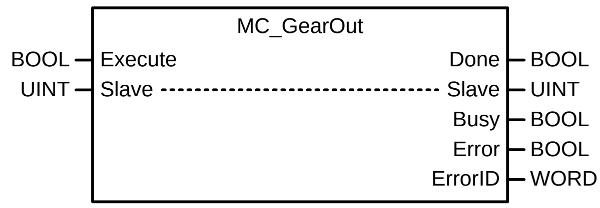


(b) Timing diagram



6.4.4 Electrical gearing disengage (MC_GearOut)

Motion Function Block



Input-Output		
UINT	Slave	Set the serve axis. (1~32: real/virtual axis, 33~36: virtual axis)
Input		
UINT	BufferMode	Specify the sequential operation setting of motion function block. (Refer to 6.1.4.BufferMode)
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	Error	Indicate whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

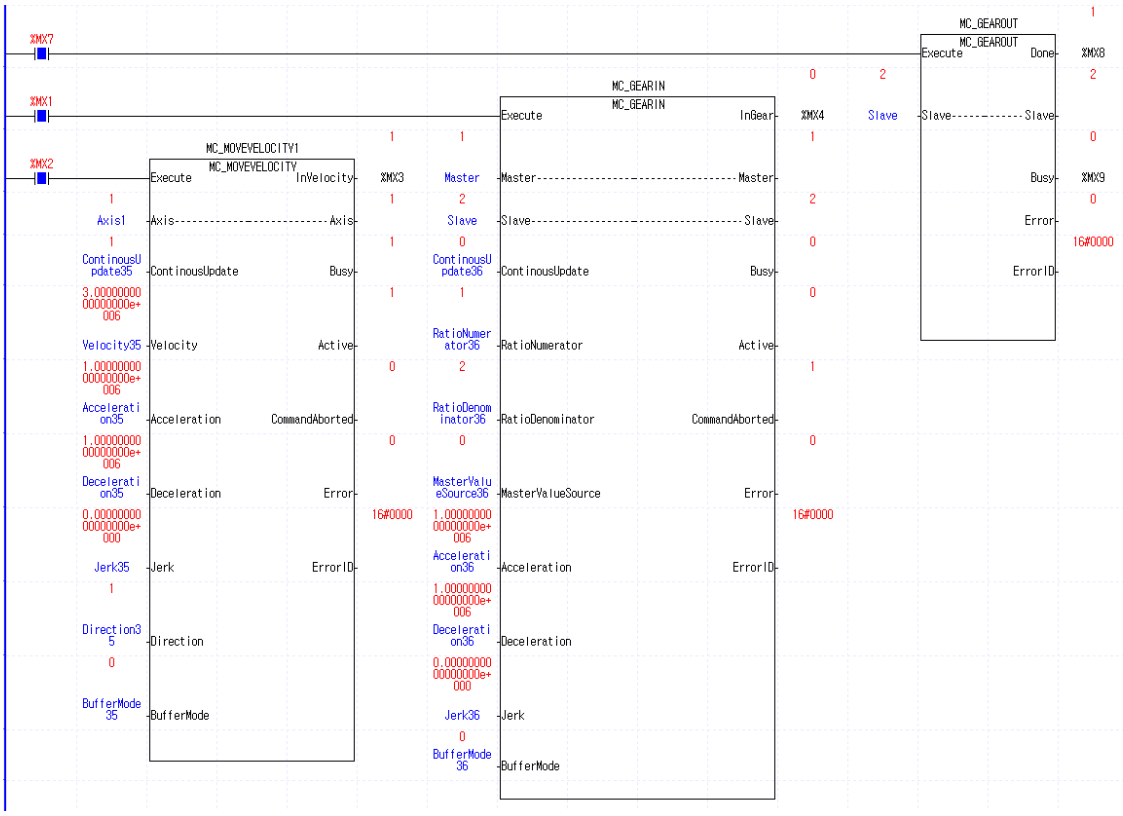
(1) This motion function block immediately disengages gear operation running in the spindle.

(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. To completely stop the spindle, use stop (MC_Halt) or immediate stop (MC_Stop) motion function block.

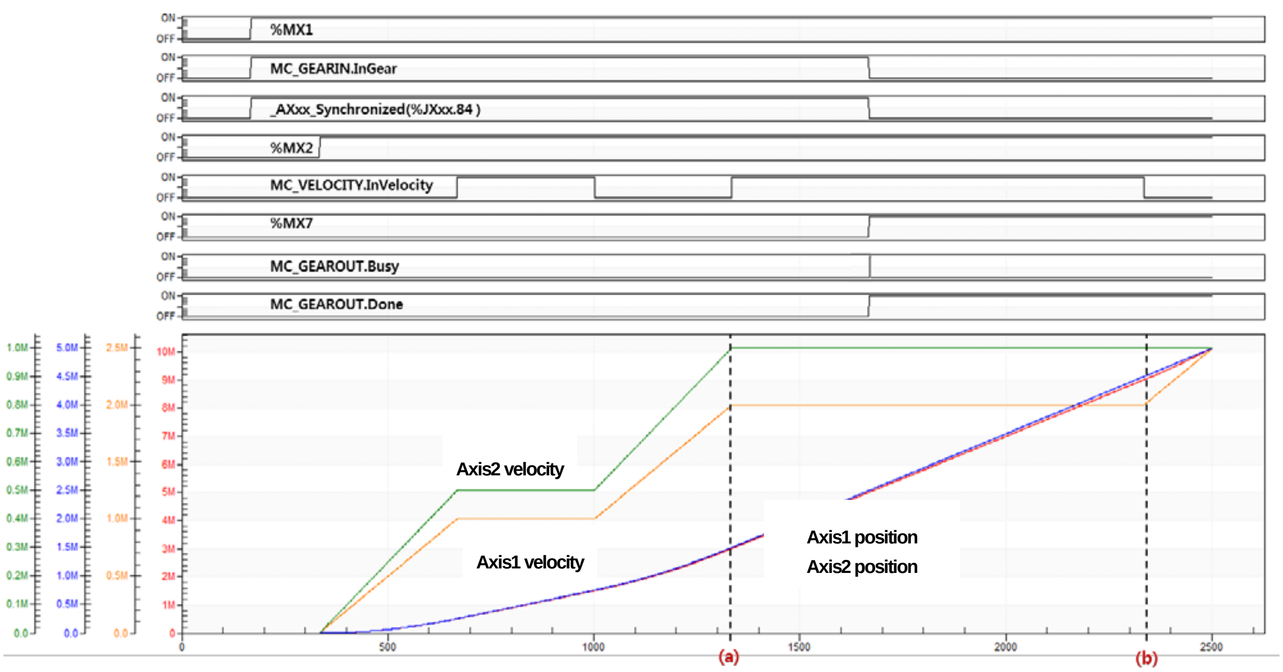
(3) Example program

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.

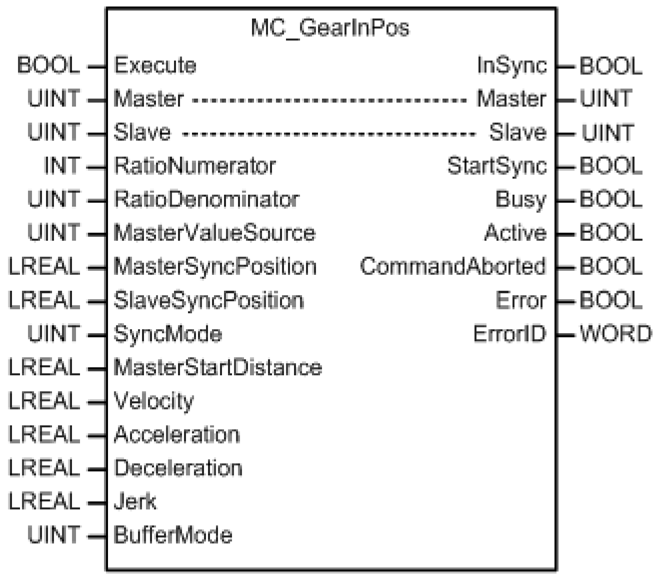
(a) Function block setting



(b) Timing diagram

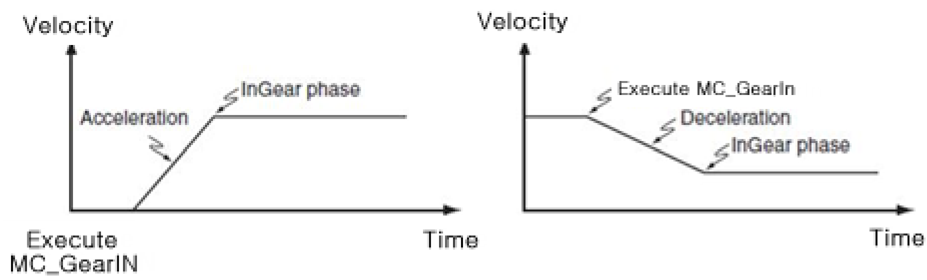


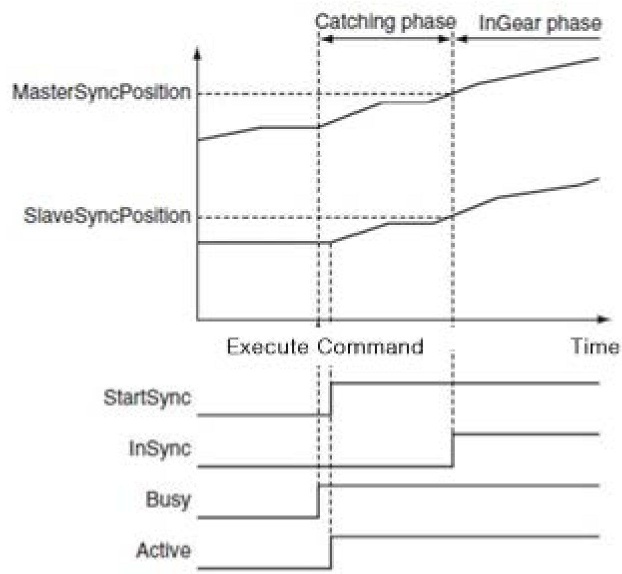
6.4.5 Electrical gearing by specifying the position (MC_GearInPos)

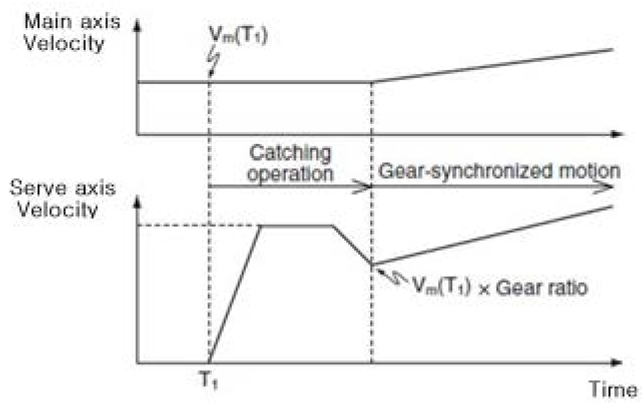
Motion Function Block		
		
Input-Output		
UINT	Master	Set the main axis. (1~32: real/virtual axis, 33~36: virtual axis1~32: real/virtuall axis, 33~36: virtual axis, 1001~1002: encoder)
UINT	Slave	Set the serve axis. (1~32: real/virtual axis, 33~36: virtual axis)
Input		
BOOL	Execute	Give a gear operation 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.
UINT	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	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	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	Indicate 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 spindle is in 'SynchronizedMotion' while this motion function block is running.





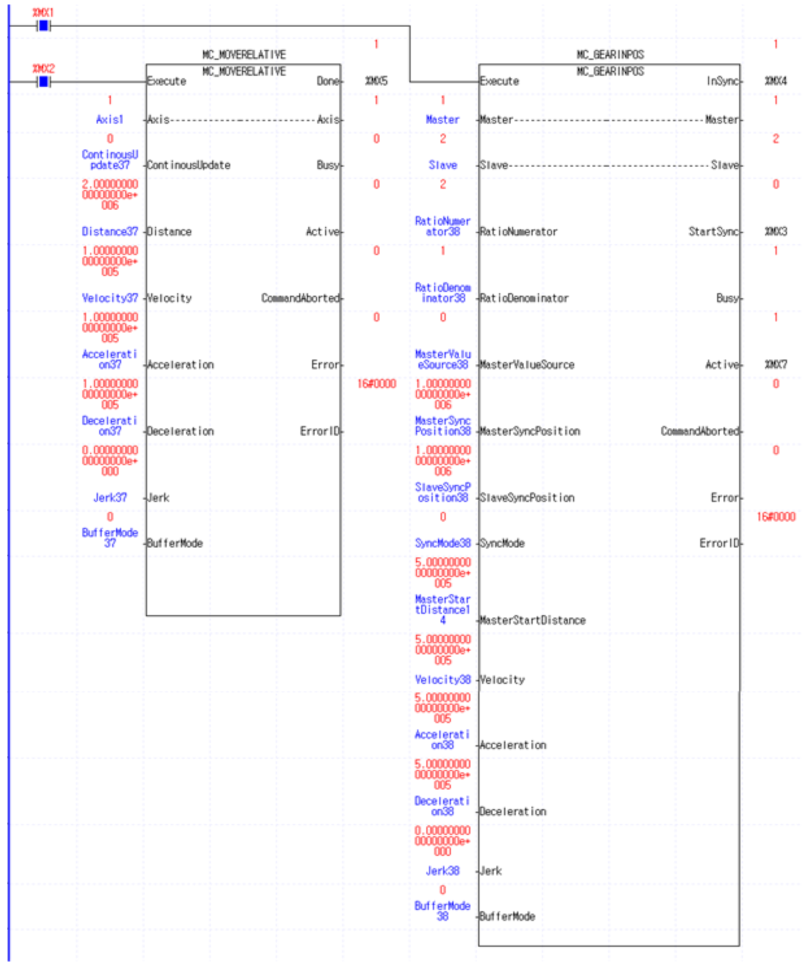


(10) The changed parameters can be applied by re-executing the function block (Execute input is On) before the command is completed. Only RatioNumerator, RatioDenominator, MasterSyncPosition, SlaveSyncPosition, MasterStartDistance, Velocity, Acceleration, Deceleration input can be updated. (However, in case of InGear=On, RatioNumerator, RatioDenominator input can be updated.

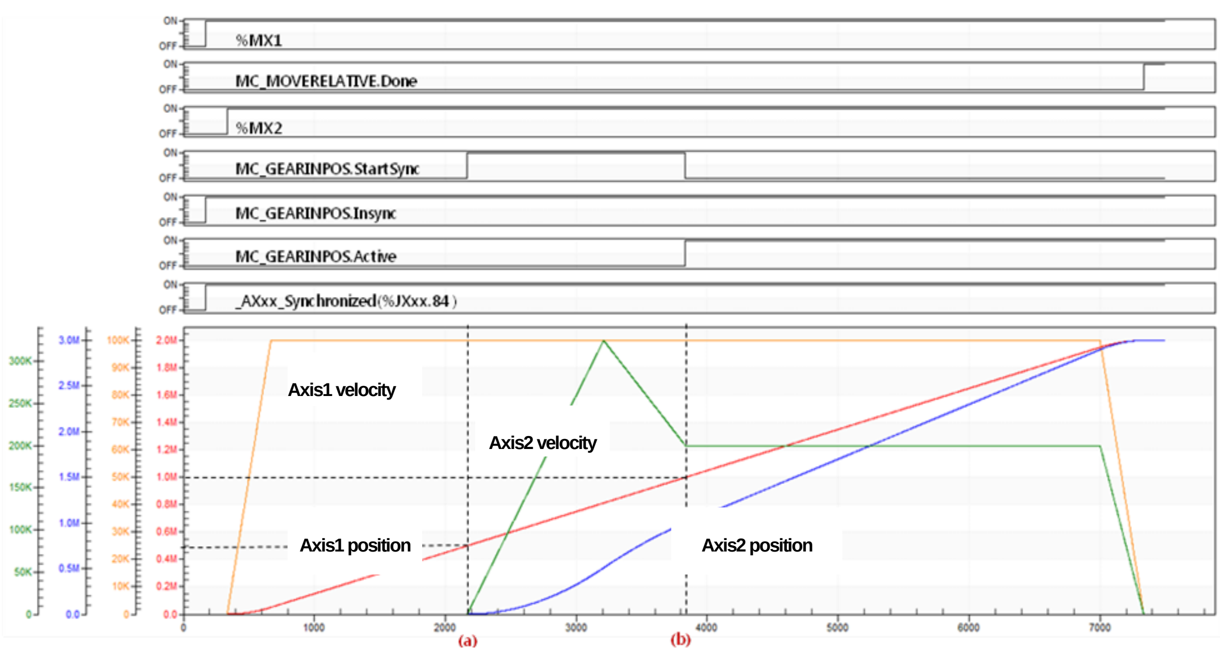
(11) Example program

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. (b)

(a) Function block setting



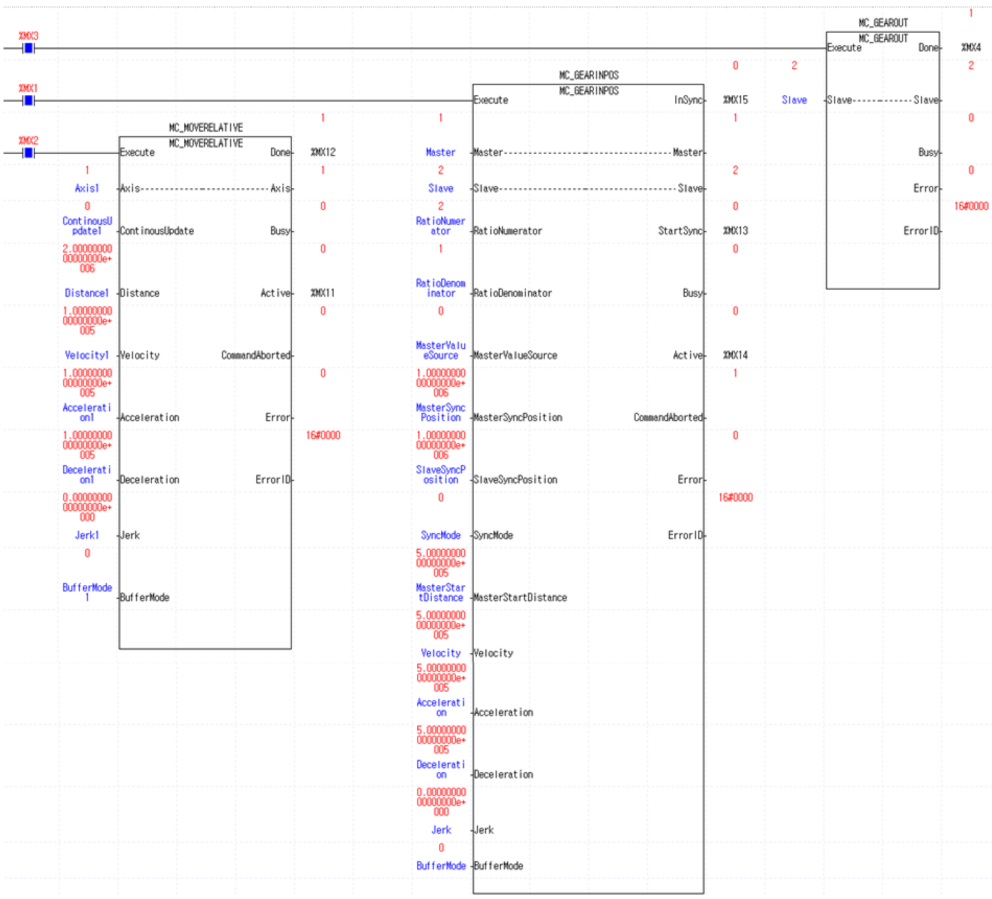
(b) Timing diagram



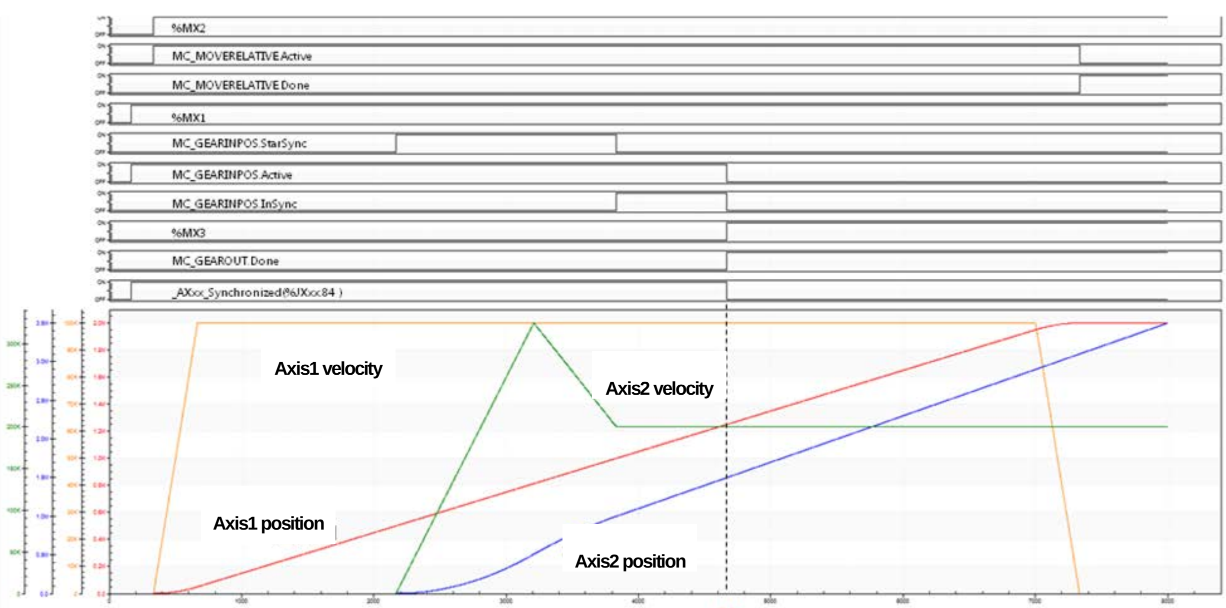
(12) Application example program

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

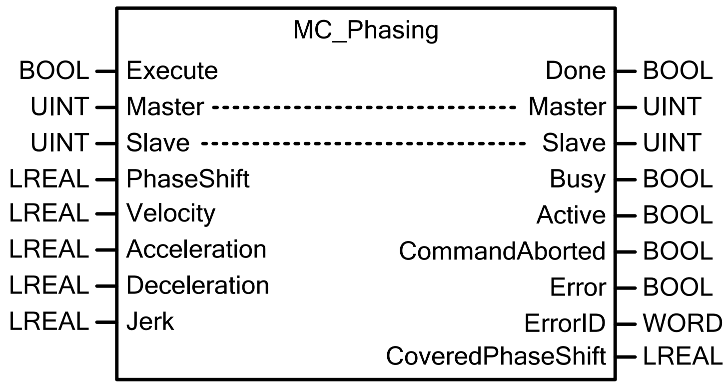


(b) Timing diagram



6.4.6 Phase compensation (MC_Phasing)

Motion Function Block

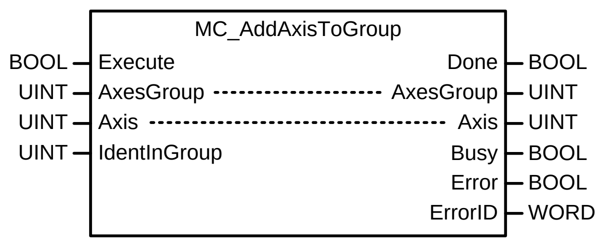


Input-Output		
UINT	Master	Set the main axis. (1~32: real/virtual axis, 33~36: virtual axis, 1001~1002: encoder)
UINT	Slave	Set the serve axis. (1~32: real/virtual axis, 33~36: virtual axis)
Input		
BOOL	Execute	Give a phase compensation command to the relevant axis in the rising Edge
LREAL	PhaseShift	Specify the main axis compensation amount.
LREAL	Velocity	Specify the phase compensation velocity. [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	Indicate 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.

6.5 Group Motion Function Blocks

6.5.1 Adds one axis to the group (MC_AddAxisToGroup)

Motion Function Block		
		
Input-Output		
UINT	AxesGroup	Set the group where the relevant axis is added. (1 ~ 16 : Group 1 ~ Group 16)
UINT	Axis	Set the axis to be added to the relevant group. (1~32: real/virtual axis, 33~36: virtual axis)
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 state of motion function block completion.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicate 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.

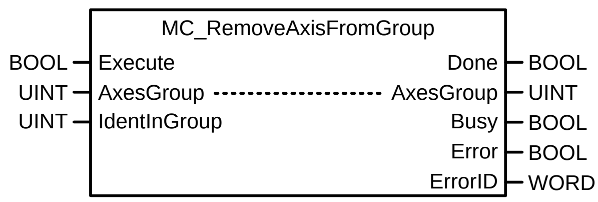
(2) ID in the axis group specified to IdentInGroup must have unique value for each axis. (ID of each axis must be different.) Maximum 10 axes can be included in each axis group, axis ID can be specified in the range of 1~10. If the specified axis number is outside the range, “error 0x0006” occurs, and 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 Removes one axis from the group (MC_RemoveAxisFromGroup)

Motion Function Block

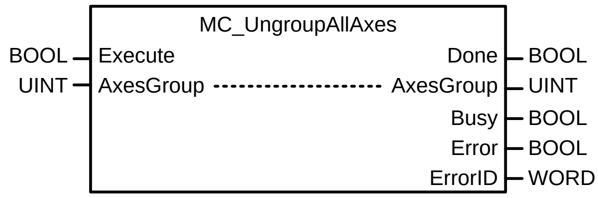


Input-Output		
UINT	AxesGroup	Set the group where the relevant axis is removed. (1 ~ 16 : Group1 ~ Group 16)
Input		
BOOL	Execute	Give group axis exclusion command to the relevant group in the rising Edge.
UINT	IdentInGroup	Set the axis number in the relevant group to be removed from the relevant group. (1~10)
Output		
BOOL	Done	Indicate the state of motion function block completion.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicate 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 : Group 1 ~ Group 16)
Input		
BOOL	Execute	Give MC_UngroupAllAxes 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.
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	Error	Indicate whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

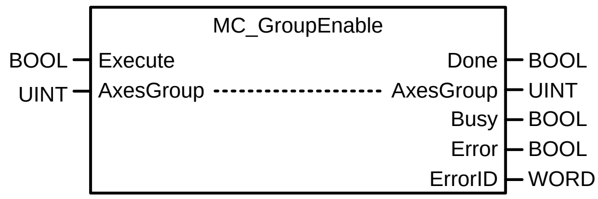
(1) This motion function block removes every axis which belongs to the axis group specified in AxesGroup input.

(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 Changes the state for group from GroupDisable to GroupEnable (MC_GroupEnable)

Motion Function Block



Input-Output		
UINT	AxesGroup	Set the group to be activated. (1 ~ 16 : Group 1 ~ Group 16)
Input		
BOOL	Execute	Give group activation command to the relevant group in the rising Edge.
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	Error	Indicate whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

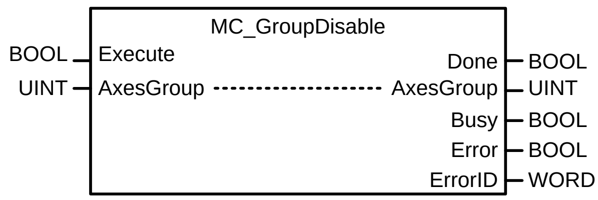
(1) This motion function block is to activate the axis group specified in AxesGroup input.

(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 Changes the state for group from GroupEnable to GroupDisable (MC_GroupDisable)

Motion Function Block



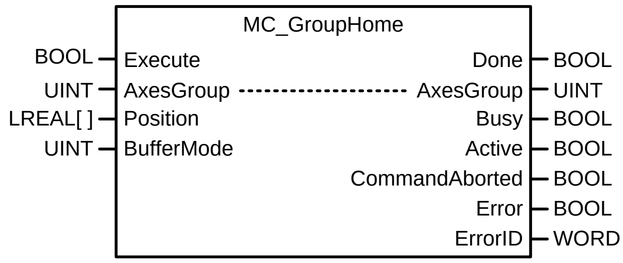
Input-Output		
UINT	AxesGroup	Set the group to be deactivated. (1 ~ 16 : Group 1 ~ Group 16)
Input		
BOOL	Execute	Give group disablement command to the relevant group in the rising Edge.
Output		
BOOL	Done	Indicate the state of motion function block completion.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicate 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 : Group 1 ~ Group 16)
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)
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	Indicate 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.

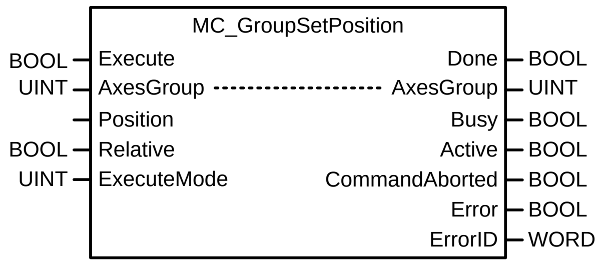
(2) Homing method is operated as specified in servo parameter of the relevant axis in advance.

(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.

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 : Group 1 ~ Group 16)
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=Relative position
UINT	ExecuteMode	0: Immediately applied the position value, 1: Applied at the same point with 'Buffered' of Buffermode
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	Indicate 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) ExcutionMode input specifies the setting point. If it is 0, it is set immediately after the execution of a command, If it is 1, it is set at the same point with ‘Buffered’ of sequential operation setting. The value unable to be set causes "error 0x201B”.

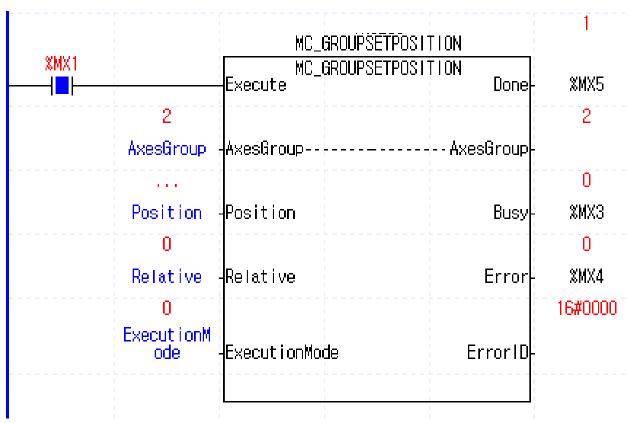
0 (mcImmediately): Change the value of parameter immediately after the execution of motion function block (rising Edge in Execute input). If the relevant axis is running, the operation can be affected.

1 (mcQueued): Changed at the same point of ‘Buffered’ of Buffermode (Refer to 6.1.4.BufferMode).

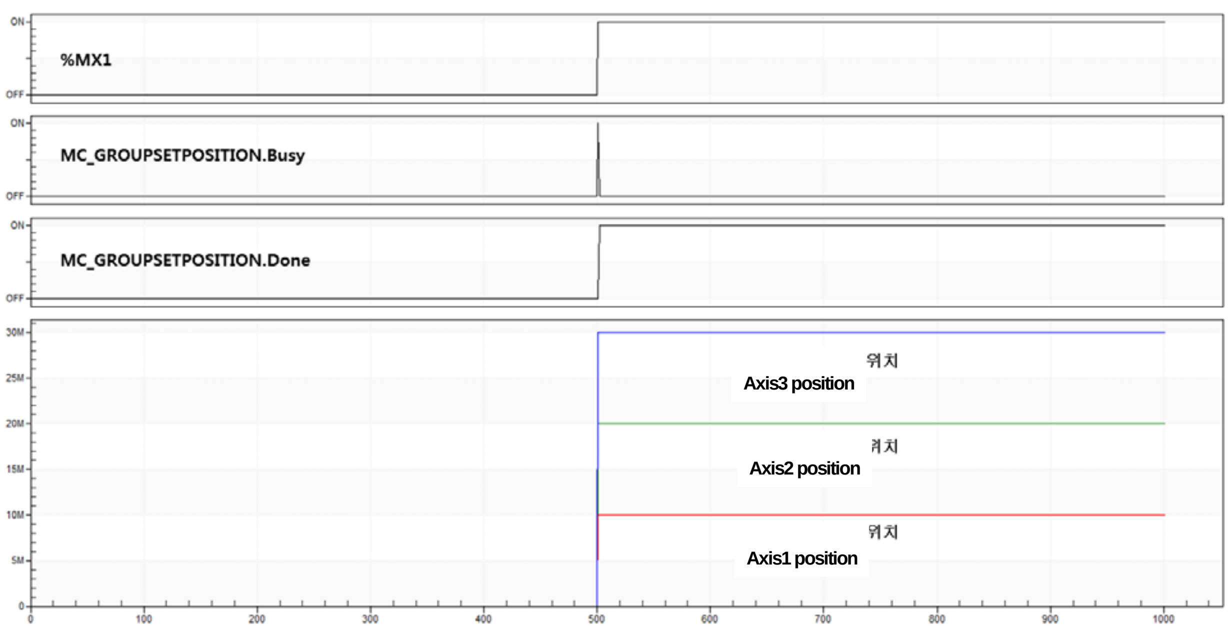
(4) Example program

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

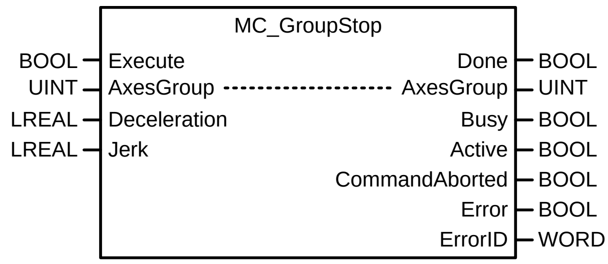


(b) Timing diagram



6.5.8 Stop the group immediately (MC_GroupStop)

Motion Function Block



Input-Output		
UINT	AxesGroup	Set the group to stop immediately. (1 ~ 16 : Group 1 ~ Group 16)
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 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	Indicate 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.

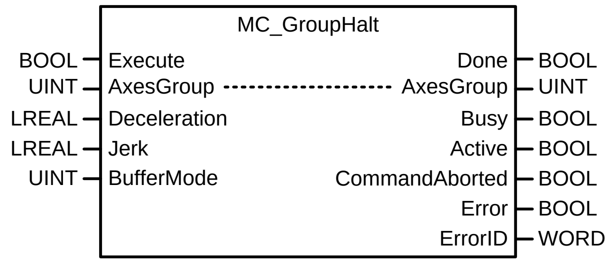
(2) The relevant axis group moves on the route which it was following until it completely stops.

(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.

6.5.9 Stop the group (MC_GroupHalt)

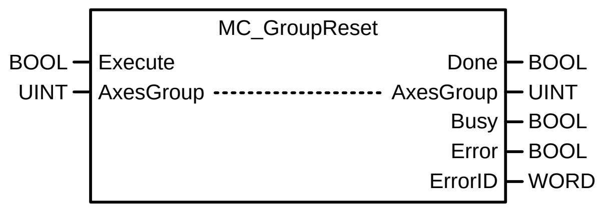
Motion Function Block		
		
Input-Output		
UINT	AxesGroup	Set the group to stop. (1 ~ 16 : Group 1 ~ Group 16)
Input		
BOOL	Execute	Give group stop command to the relevant group in the rising Edge.
LREAL	Deceleration	Specify the deceleration in the time of stop. [u/s^2]
LREAL	Jerk	Specify the change rate 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 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	Indicate 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.

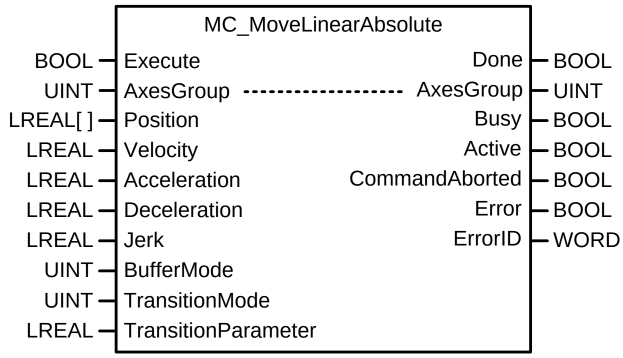
6.5.10 Reset the group error (MC_GroupReset)

Motion Function Block		
		
Input-Output		
UINT	AxesGroup	Set the group to do error reset. (1 ~ 16 : Group 1 ~ Group 16)
Input		
BOOL	Execute	Give group error reset command to the relevant group in the rising Edge.
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	Error	Indicate 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 positioning linear interpolation operation (MC_MoveLinearAbsolute)

Motion Function Block		
		
Input-Output		
UINT	AxesGroup	Set the group to perform absolute position linear interpolation operation. (1 ~ 16: Group 1 ~ Group 16)
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)
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 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	Indicate 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 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.
Operation speeds of each configuration axis are 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.

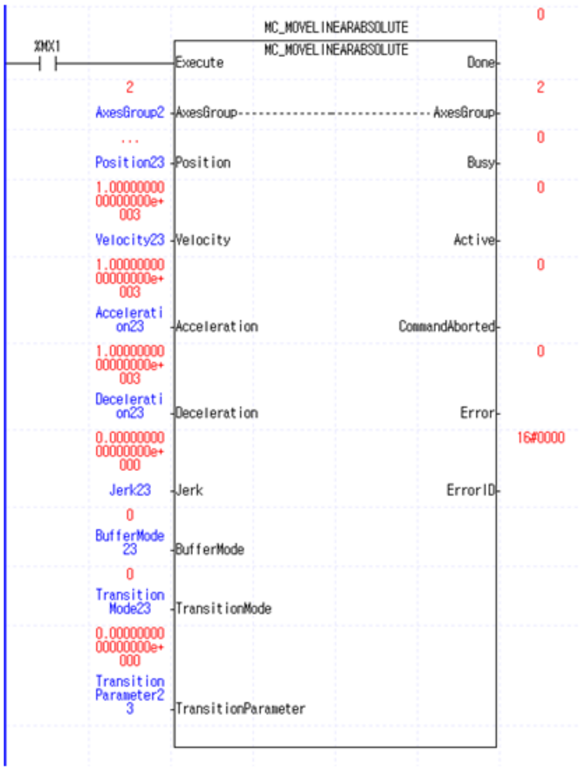
(7) The changed parameters can be applied by re-executing the function block (Execute input is On) before the command is completed. Only Velocity, Acceleration, Deceleration, Jerk, Position input can be updated.

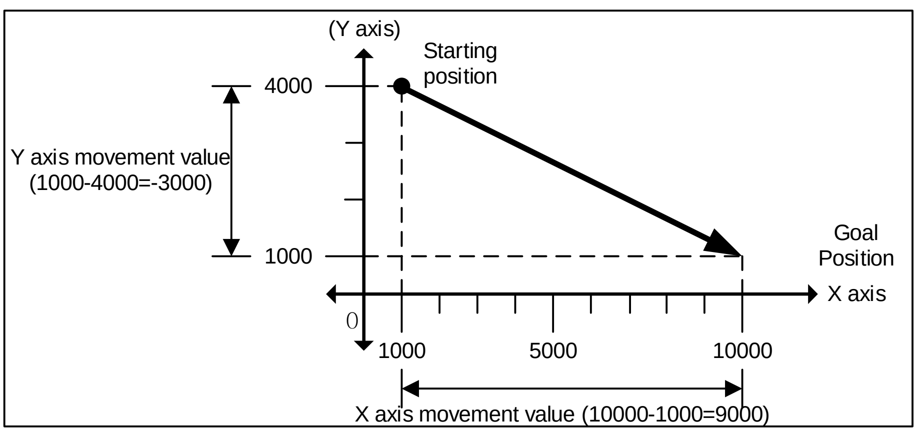
(8) Velocity input can be set to 0 or changed.

(9) Example program

This example shows the linear interpolation to the target position (10000, 1000) when the current command position is (1000, 4000).

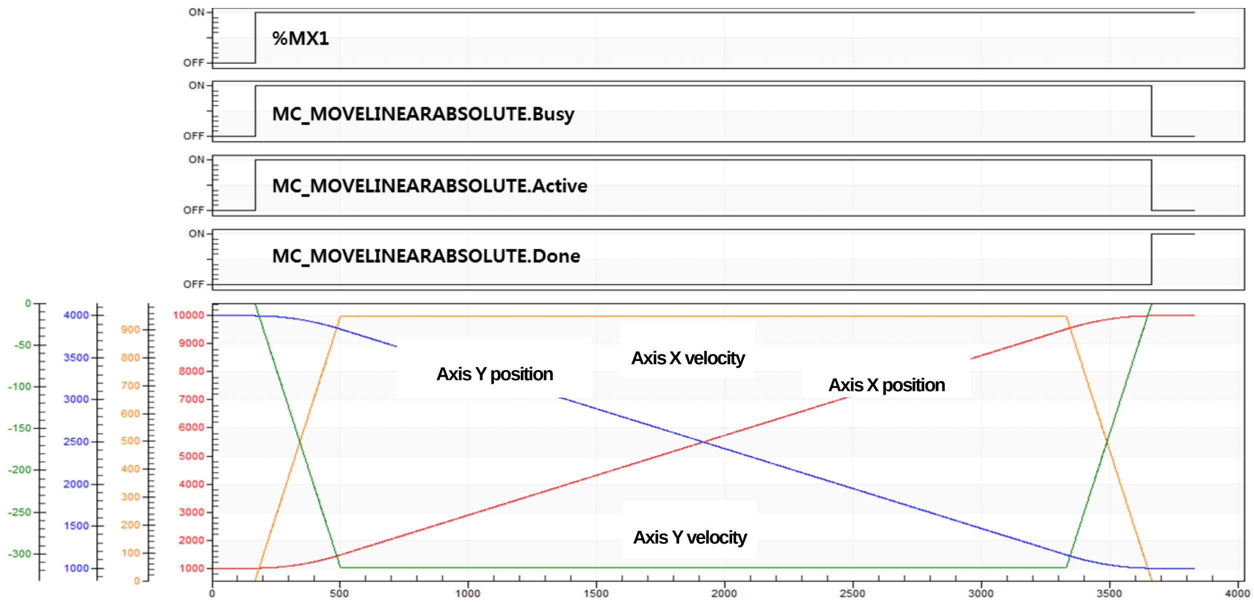
(a) Function block setting



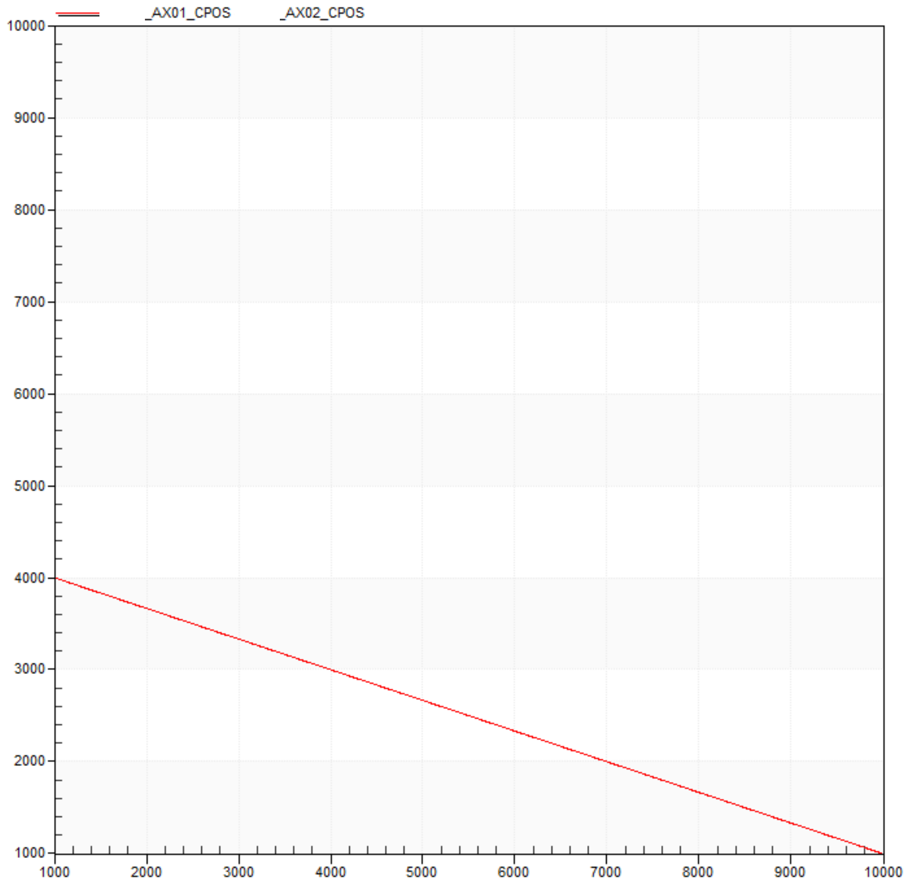


1	<GLOBAL>	%JL1.3	1.0000000000000000e+003	LREAL	_AX01_CPOS
2	<GLOBAL>	%JL2.3	4.0000000000000000e+003	LREAL	_AX02_CPOS
3	<GLOBAL>	%JL1.4	0.0000000000000000e+000	LREAL	_AX01_CVEL
4	<GLOBAL>	%JL2.4	0.0000000000000000e+000	LREAL	_AX02_CVEL
5	Group	GroupHome	On	BOOL	
6	Group	GourpHalt	Off	BOOL	
7	Group	⊟ 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

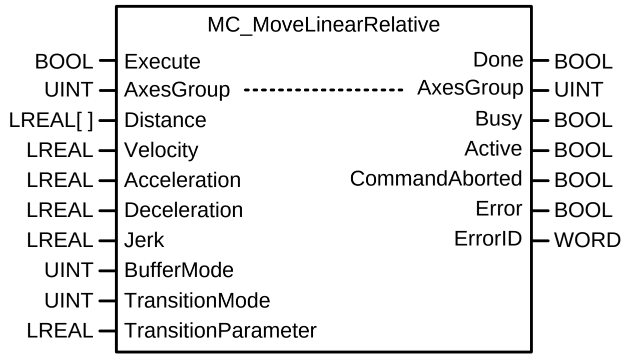


(c) XY graph



6.5.12 Relative positioning linear interpolation operation (MC_MoveLinearRelative)

Motion Function Block		



Input-Output		
UINT	AxesGroup	Set the group to do relative position linear interpolation operation. (1 ~ 16: Group 1 ~ Group 16)
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)
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 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	Indicate 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.
Operation speeds of each configuration axis are 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.

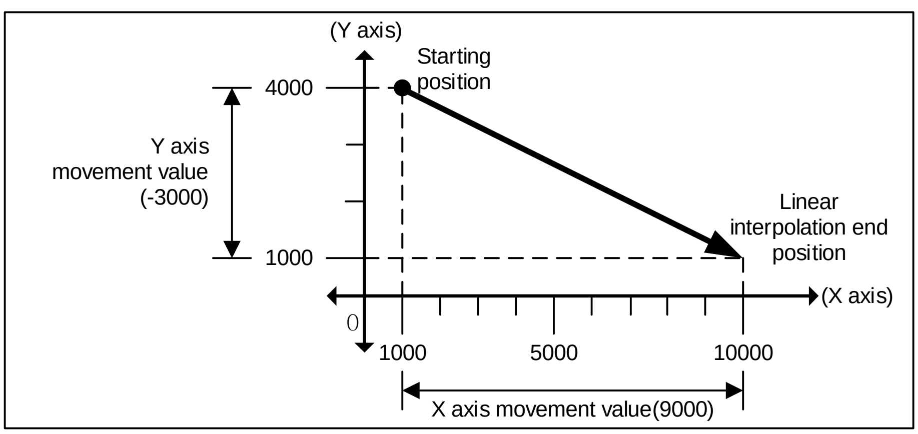
(7) The changed parameters can be applied by re-executing the function block (Execute input is On) before the command is completed. Only Velocity, Acceleration, Deceleration, Jerk, Position input can be updated.

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

(9) Example program

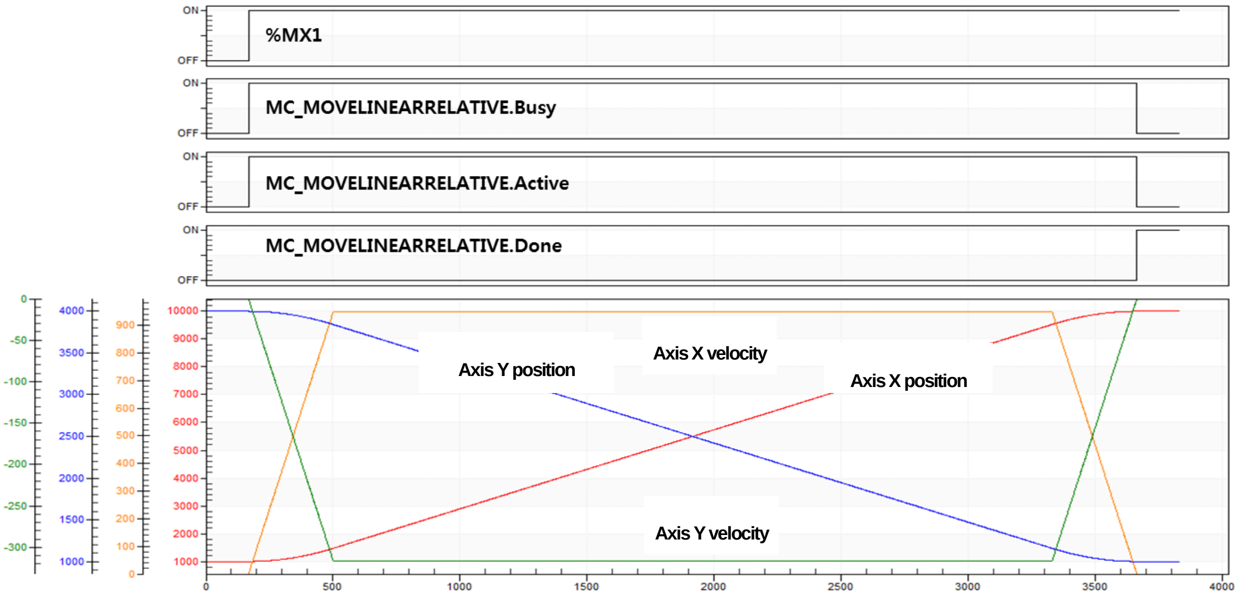
This example shows the linear interpolation to the target position (10000, 1000) by moving the target distance (X-axis: 9000, Y-axis: -3000) when the current command position is (1000, 4000).

(a) Function block setting

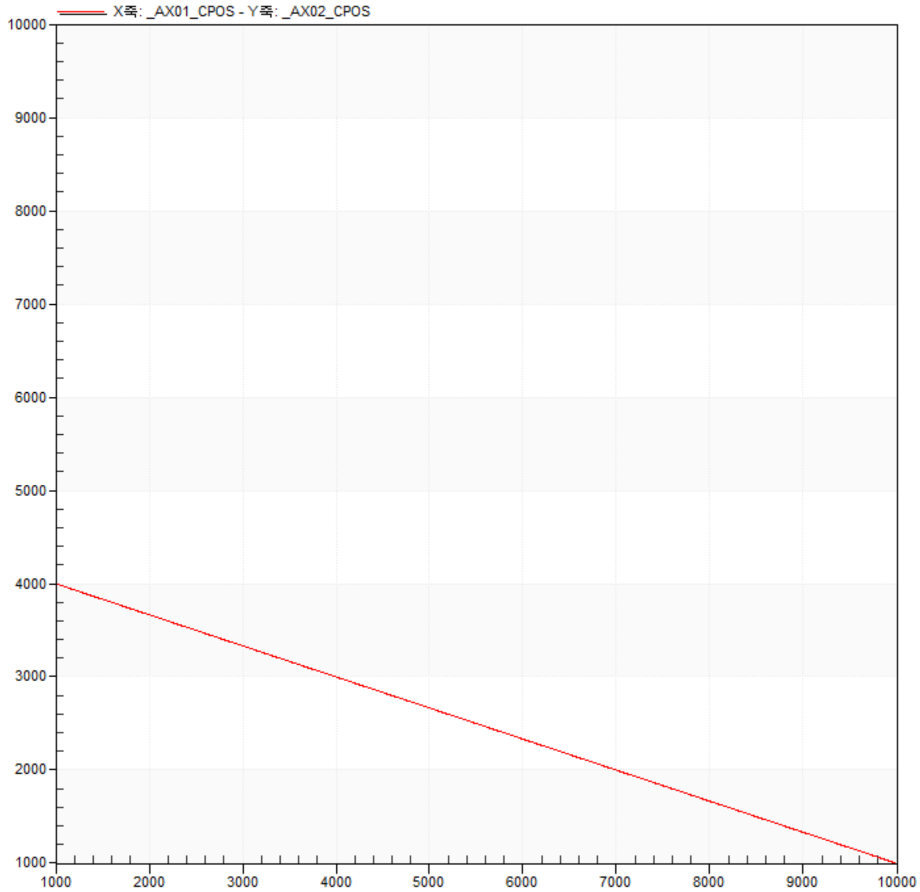


1	<GLOBAL>	%JL1.3	1.0000000000000000e+004	LREAL	_AX01_CPOS
2	<GLOBAL>	%JL2.3	1.0000000000000000e+003	LREAL	_AX02_CPOS
3	<GLOBAL>	%JL1.4	0.0000000000000000e+000	LREAL	_AX01_CVEL
4	<GLOBAL>	%JL2.4	0.0000000000000000e+000	LREAL	_AX02_CVEL
5	Group	GroupHome	On	BOOL	
6	Group	GourpHalt	Off	BOOL	
7	Group	Distance24		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

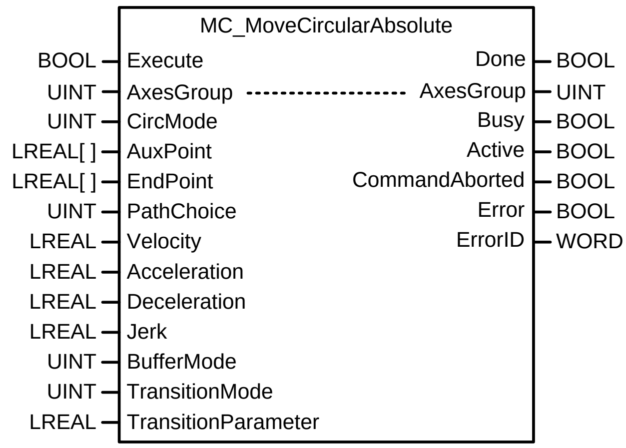


(c) XY graph



6.5.13 Absolute positioning circular interpolation operation (MC_MoveCircularAbsolute)

Motion Function Block



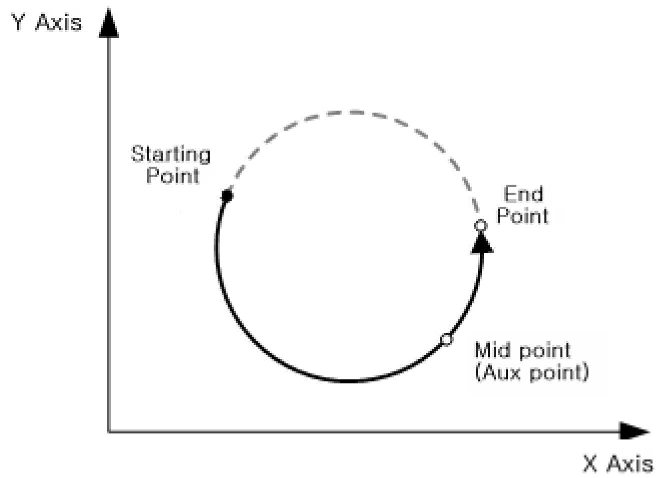
Input-Output		
UINT	AxesGroup	Set the group to do absolute position circular interpolation operation. (1 ~ 16: Group 1 ~ Group 16)
Input		
BOOL	Execute	Give absolute position circular interpolation operation command to the relevant group in the rising Edge.
UINT	CircMode	Circular interpolation method 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	Specify the end point of circular arc in an absolute coordinate.
BOOL	PathChoice	Circular route selection 0: Clockwise, 1: Counterclockwise
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)
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 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	Indicate 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 path interpolation control which refers to the set auxiliary point, and the movement direction is decided by PathChoice input. When setting PathChoice input to 0, circular interpolation operation is done clockwise, and when setting it to 1, circular interpolation operation is done counterclockwise.
(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 entered array and the axis in the group correspond in the order of the specified axis ID [ID1, ID2, ID3, ···]. (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) Specify the speed, acceleration, deceleration, and the change rate of acceleration of interpolation route in Velocity, Acceleration, Deceleration, and Jerk inputs respectively.
(5) Set the circular interpolation method in CircMode input. The circular interpolation methods which are different from the value specified in CircMode are as below.

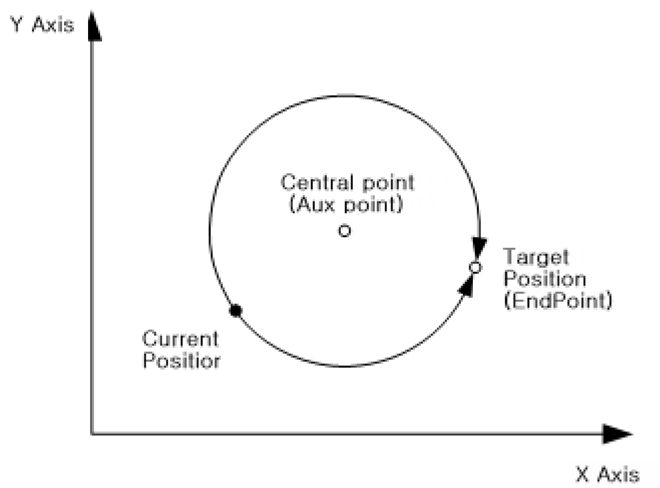
(a) Circular interpolation of midpoint specifying method (BORDER, CircMode = 0)

In this method, operation starts at the starting point and it does circular interpolation through the specified position of the central point to the target position. The Figure below shows that the coordinate of the axis group at the beginning of a command corresponds to the starting point, the coordinate entered in AuxPoint corresponds to the central point, and the coordinate entered in EndPoint corresponds to the target position in an absolute value.



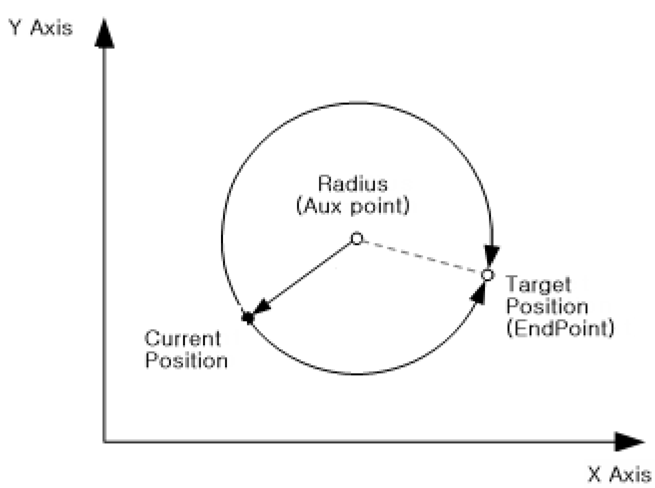
(b) Circular interpolation of central point specifying method

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. The Figure below shows that the coordinate of the axis group at the beginning of a command corresponds to the current position, the coordinate entered in AuxPoint corresponds to the central point, and the coordinate entered in EndPoint corresponds to the target point as an absolute value.



(c) Circular interpolation using the radius 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 value specified in the radius. The Figure below shows that the coordinate of the axis group at the beginning of a command corresponds to the current position, the value entered in X-axis of AuxPoint corresponds to the radius, and the coordinate entered in EndPoint corresponds to the target point in an absolute value.



(6) Refer to chapter 8.2.7 linear interpolation control part in motion controller's manual for more details.

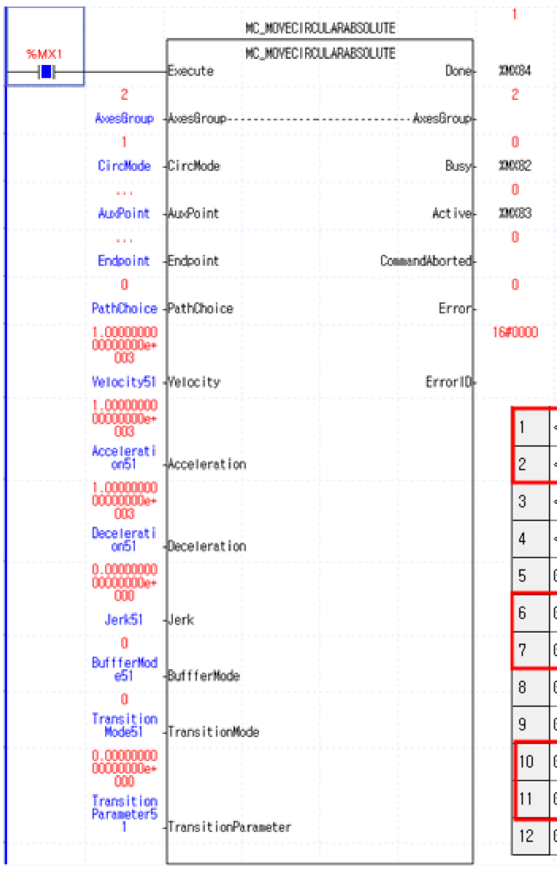
(7) The changed parameters can be applied by re-executing the function block (Execute input is On) before the command is completed. Only Velocity, Acceleration, Deceleration, Jerk, AuxPoint, EndPoint input can be updated.

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

(9) Example program

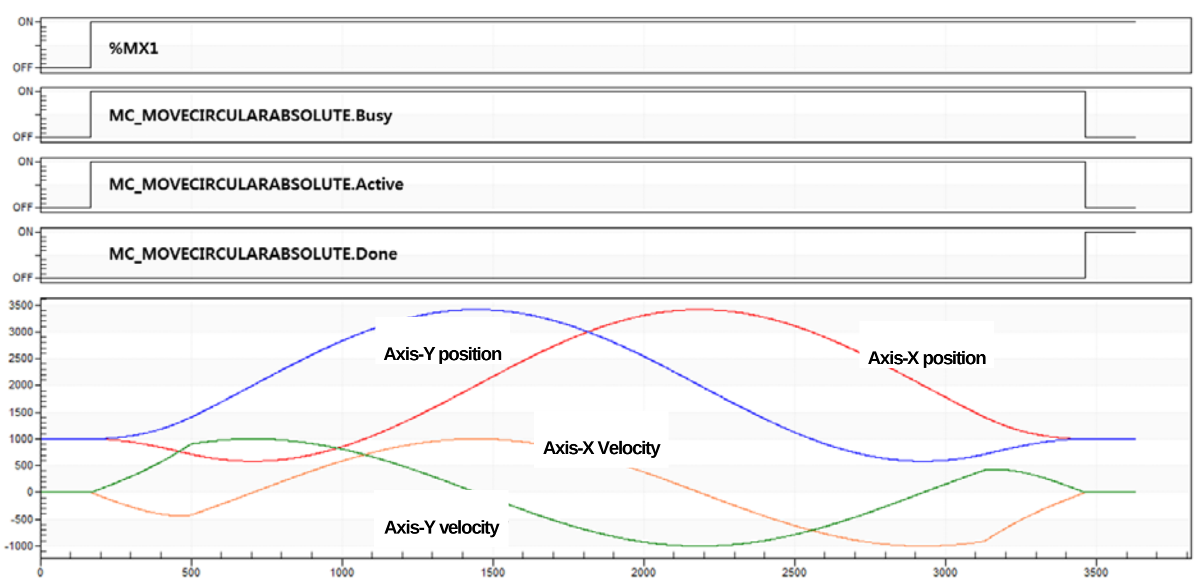
This example shows the circular interpolation to the target position (1000, 1000) by moving clock-wise after setting the center point (2000,2000) specification method when the current command position is (1000, 1000).

(a) Function block setting

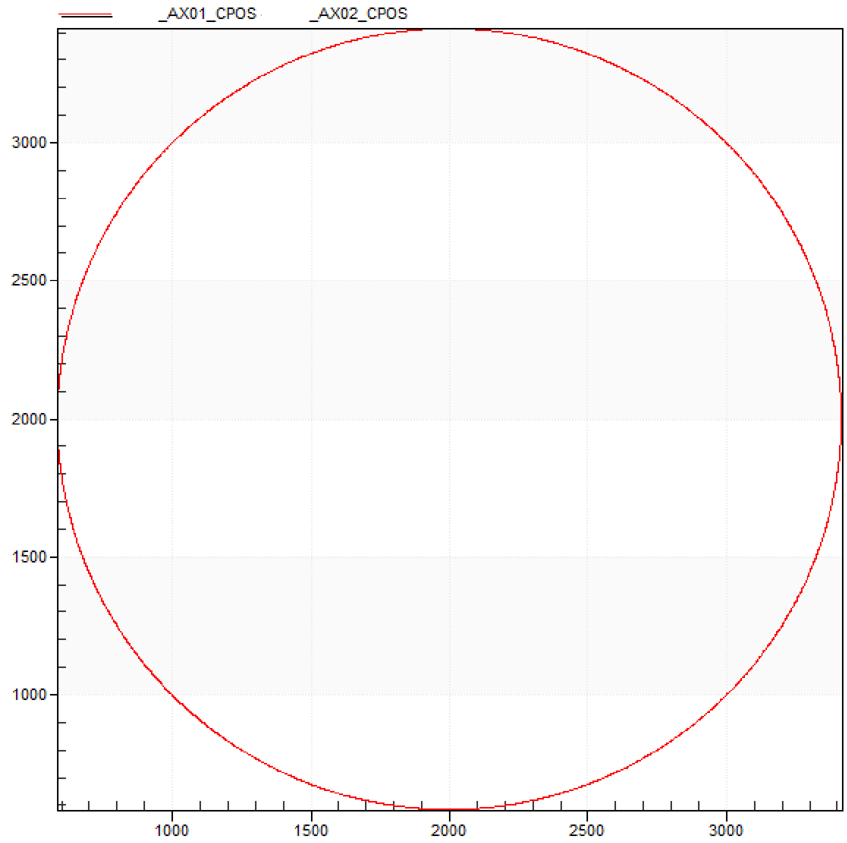


1	<GLOBAL>	%JL1.3		1.0000000000000000e+003	LREAL	_AX01_CPOS
2	<GLOBAL>	%JL2.3		1.0000000000000000e+003	LREAL	_AX02_CPOS
3	<GLOBAL>	%JL1.4		0.0000000000000000e+000	LREAL	_AX01_CVEL
4	<GLOBAL>	%JL2.4		0.0000000000000000e+000	LREAL	_AX02_CVEL
5	Group	⊟ AuxPoint			ARRAY[0..2] OF LREAL	
6	Group	AuxPoint[0]		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

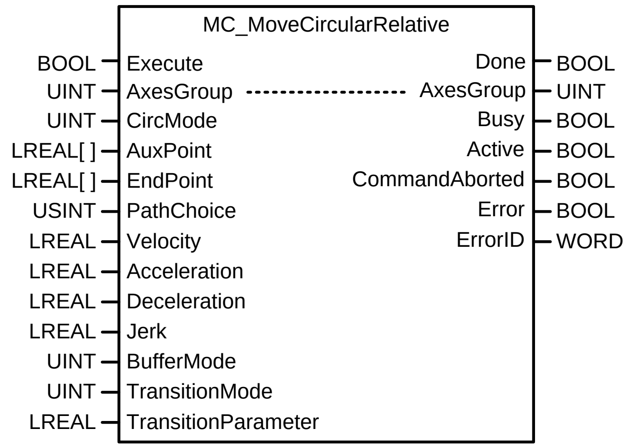


(c) XY graph


_AX01_CPOS
_AX02_CPOS
3000
2500
2000
1500
1000
1000
1500
2000
2500
3000

6.5.14 Relative positioning circular interpolation operation (MC_MoveCircularRelative)

Motion Function Block



Input-Output		
UINT	AxesGroup	Set the group to do absolute position circular interpolation operation. (1 ~ 16: Group 1 ~ Group 16)
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, 1: Counterclockwise
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)
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 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	Indicate 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 circular interpolation command to the axis group specified in AxesGroup input.

(2) When this motion function block starts, each axis performs circular path interpolation control which refers to the set auxiliary point, and the movement direction is decided by PathChoice input. When setting PathChoice input to 0, circular interpolation operation is done clockwise, and when setting it to 1, circular interpolation operation is done counterclockwise.

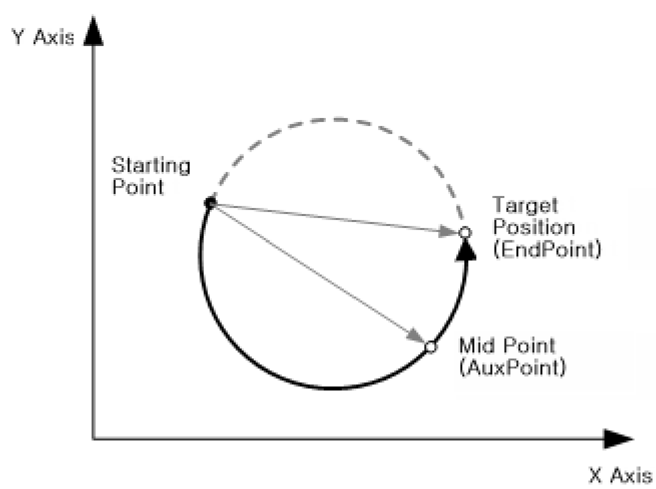
(3) Specify the relative position of the auxiliary point to refer when doing circular interpolation of each axis in AuxPoint and EndPoint inputs as array. The entered array and the axis in the group correspond in the order of the specified axis ID [ID1, ID2, ID3, ···]. (The 3 LEAL type sized array should be entered in Position input as there are 3 axes which comprise the group to give a circular interpolation operation command.)

(4) Specify the speed, acceleration, deceleration, and the change rate of acceleration of interpolation route in Velocity, Acceleration, Deceleration, and Jerk inputs respectively.

(5) Set the circular interpolation method in CircMode input. The circular interpolation methods which are different from the value specified in CircMode are as below.

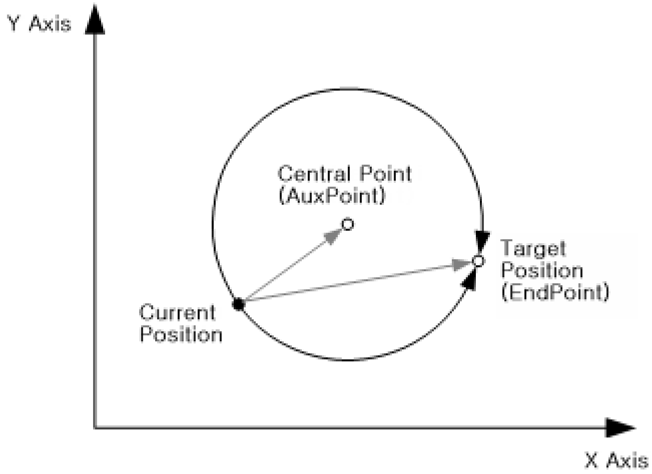
(a) Circular interpolation of midpoint specifying method (BORDER, CircMode = 0)

In this method, operation starts at the current position and it does circular interpolation through the specified position of the central point to the target position.
The Figure below shows that the coordinate of the axis group at the beginning of a command corresponds to the current position, the coordinate entered in AuxPoint corresponds to the central point, and the coordinate entered in EndPoint corresponds to the target position in a relative value.



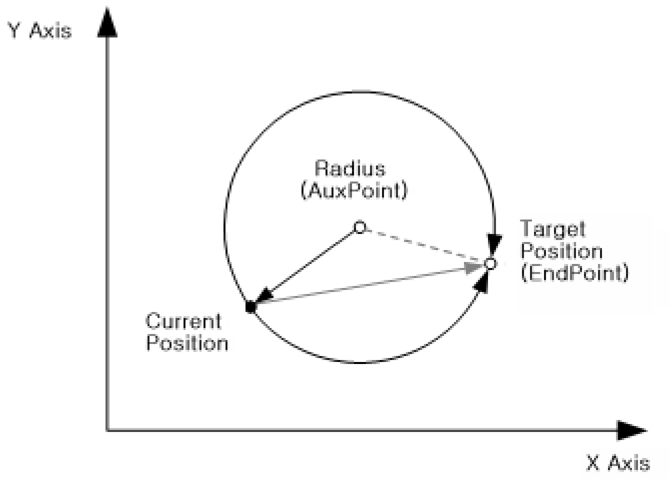
(b) Circular interpolation of central point specifying method

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. The Figure below shows that the coordinate of the axis group at the beginning of a command corresponds to the current position, the coordinate entered in AuxPoint corresponds to the central point, and the coordinate entered in EndPoint corresponds to the target point as a relative value.



(c) Circular interpolation using the radius 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 value specified in the radius. The Figure below shows that the coordinate of the axis group at the beginning of a command corresponds to the current position, the value entered in X-axis of AuxPoint corresponds to the radius, and the coordinate entered in EndPoint corresponds to the target point in a relative value.



(6) Refer to 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, and Endpoint inputs can be updated.

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

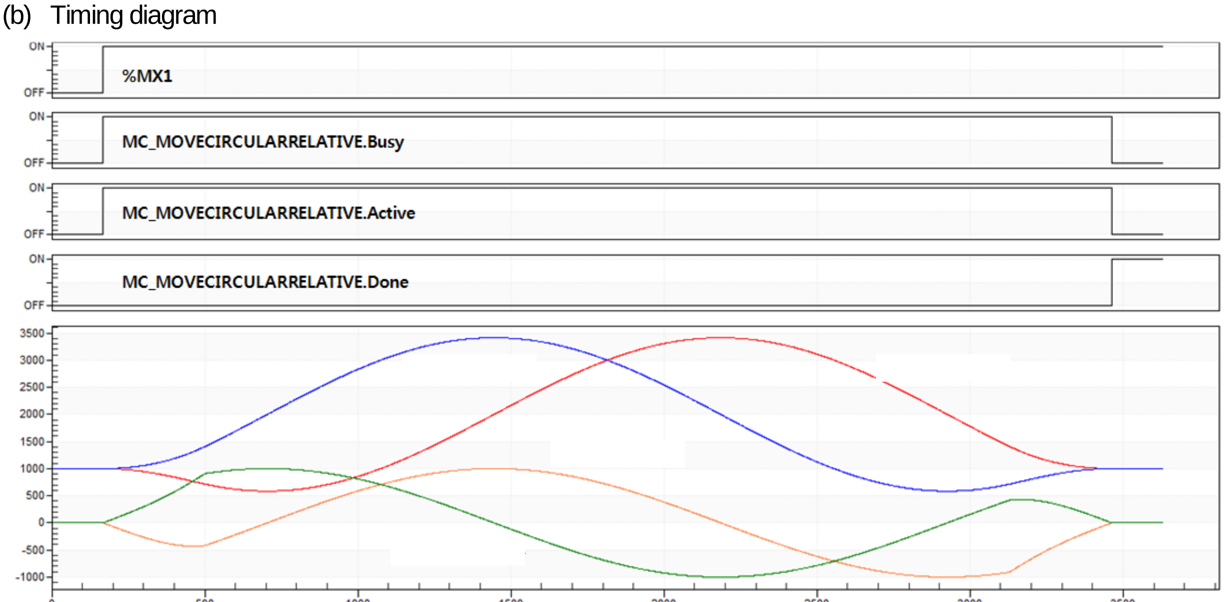
(9) Example program

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).

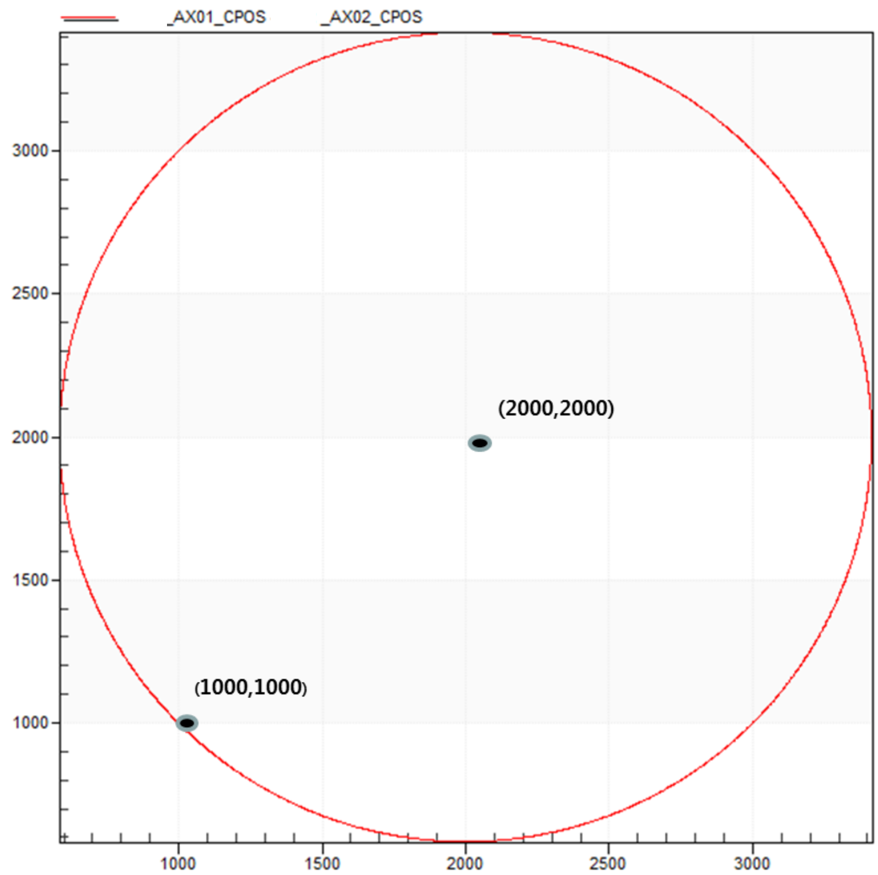
(a) Function block setting

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) Timing diagram

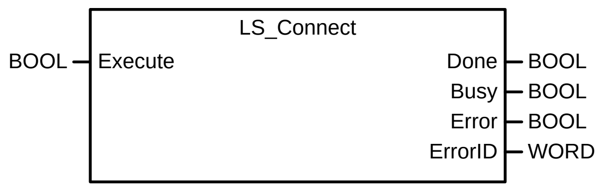


(c) XY graph


_AX01_CPOS
_AX02_CPOS
(2000,2000)
(1000,1000)
3000
2500
2000
1500
1000
1000
1500
2000
2500
3000

6.6 Exclusive Function Blocks

6.6.1 Connect servo drives (LS_Connect)

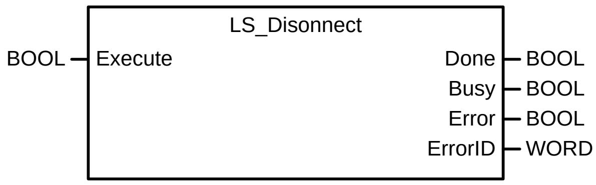
Motion Function Block		
		
Input		
BOOL	Execute	Give communication connection command to the relevant motion controller in the rising Edge.
Output		
BOOL	Done	Indicate whether to complete communication connection.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicate 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 Disconnect servo drives (LS_Disconnect)

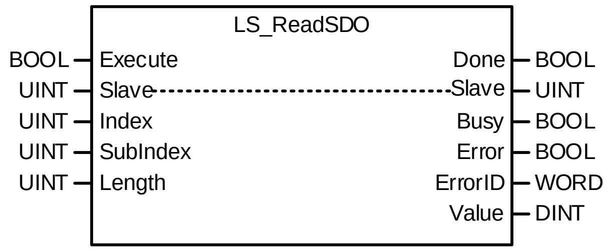
Motion Function Block		
		
Input		
BOOL	Execute	Give communication disconnection command to the relevant motion controller in the rising Edge.
Output		
BOOL	Done	Indicate whether to complete communication disconnection.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicate 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.

(3) 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		
		
Input-Output		
UINT	Slave	Set the slave to be given a command. (1~64: slave 1~slave 64)
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	Indicate whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.
LREAL	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).

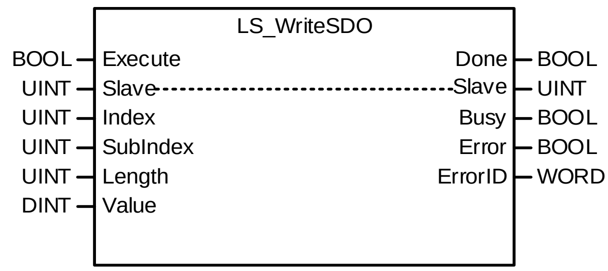
(3) Index input can be set as below. If the value is set outside the range, "error 0x1F12" occurs.

Variable	Description
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) The value between 1~4 can be set in Length, which means 1~4 Byte. If the value is set outside the range, “error 0x1F12” occurs.

6.6.4 Write SDO (LS_SDO)

Motion Function Block		
		
Input-Output		
UINT	Slave	Set the slave to be given a command. (1~64: slave 1~slave 64)
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	Indicate 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.

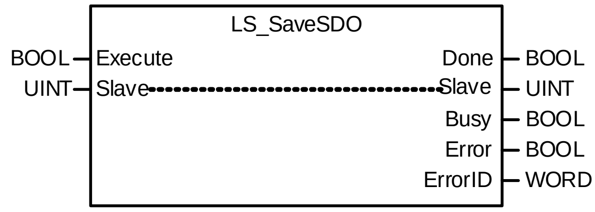
(2) Index input can be set as below. When it is set to the value besides the set value, “error 0x1F12” occurs.

Value	Description
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 the range of 0~255 can be entered in SubIndex, and if the value outside the range is set, “error 0x1F12” occurs.

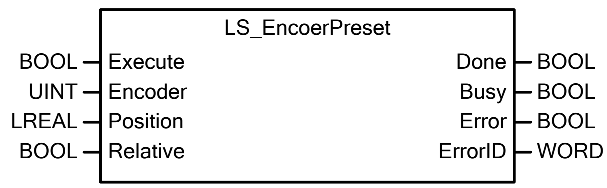
(4) The value between the range of 1~4 can be entered in Length, which means 1~4 Byte. If the value outside the range is set, "error 0x1F12” occurs.

6.6.5 Save SDO (LS_SaveSDO)

Motion Function Block		
		
Input-Output		
UINT	Slave	Set the slave to be given a command. (1~64: slave 1~slave 64)
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 the execution of motion function block is not completed.
BOOL	Error	Indicate 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 a command to save SDO of the designated slave to the memory of the slave.

6.6.6 Encoder preset (LS_EncoderPreset)

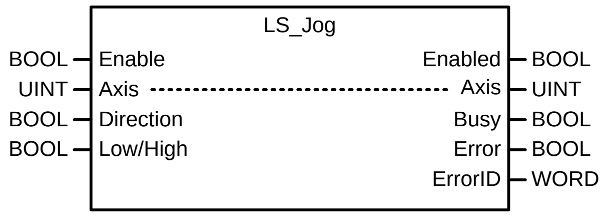
Motion Function Block		
		
Input		
BOOL	Execute	Specify the position of the relevant encoder in the rising Edge.
UINT	Encoder	Set the encoder to set the position. (1~2: Encoder 1~Encoder 2)
LREAL	Position	Specify the position to set. [u]
BOOL	Relative	0: Absolute coordinate position 1: Relative coordinate position
Output		
BOOL	Done	Indicate the state of motion function block completion.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicate 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.

6.6.7 JOG operation (LS_Jog)

Motion Function Block



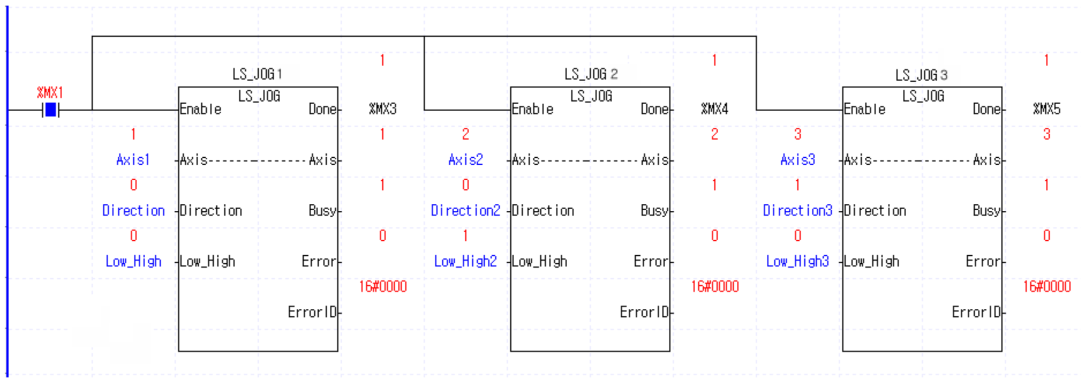
Input-Output		
UINT	Axis	Set the axis to be given a command. (1~32: real/virtual axis, 33~36: virtual axis)
Input		
BOOL	Enable	Give jog command to the relevant axis while input is On.
BOOL	Direction	Set the rotation direction in jog (0: Forward direction, 1: Reverse direction)
BOOL	Low/High	Set the jog speed in jog. (0: Jog low speed operation, 1: 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	Indicate 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.

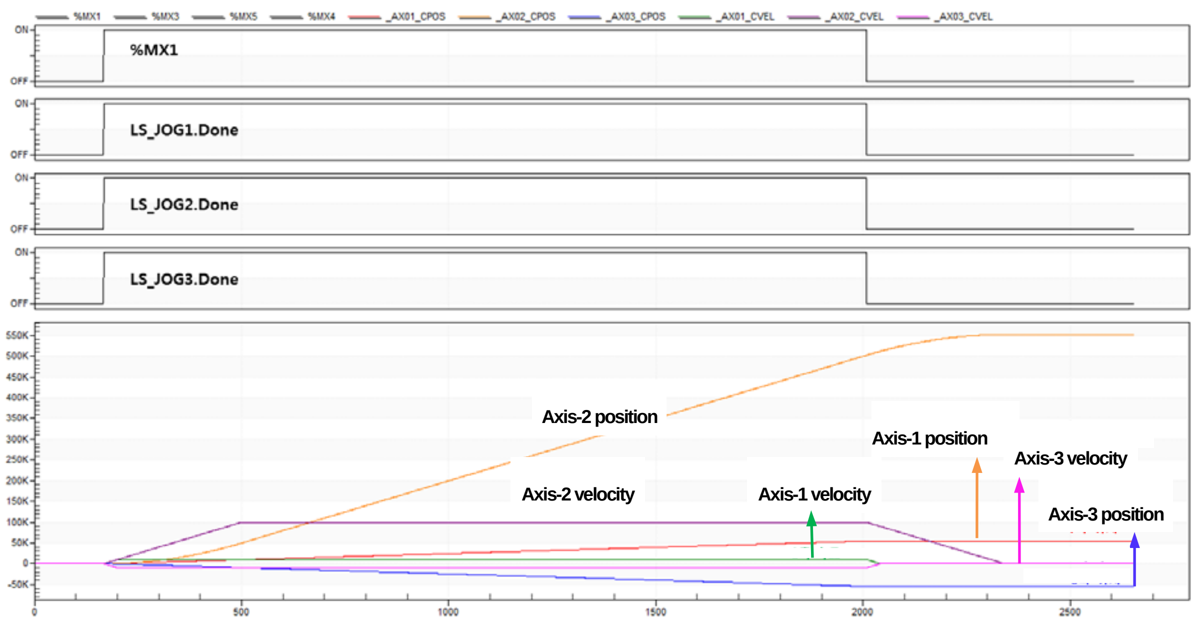
(4) Example program

This example shows jog operation under the following settings when the current command position is 0.

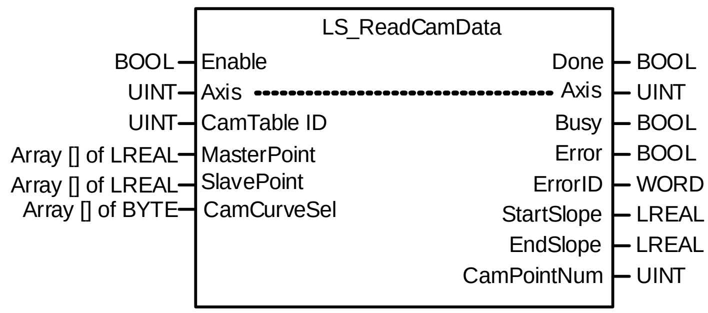
(a) Function block setting



(b) Timing diagram



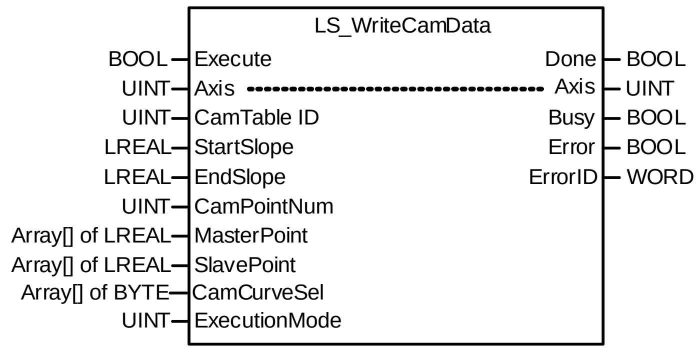
6.6.8 Read Cam data (LS_ReadCamData)

Motion Function Block		
		
Input-Output		
UINT	Axis	Specify the axis to be commanded (1~32: real/virtual axis, 33~36: virtual axis)
Input		
BOOL	Enable	Read the relevant cam data while input is On.
UINT	CamTableID	Specify the cam table to read. (1~32)
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 motion function block output.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicate whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion 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.

6.6.9 Write Cam data (LS_WriteCamData)

Motion Function Block		
Input-Output		
UINT	Axis	Specify the axis to be commanded (1~32: real/virtual axis, 33~36: virtual axis)
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. (1~32)
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 the execution of motion function block is not completed.
BOOL	Error	Indicate 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 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.

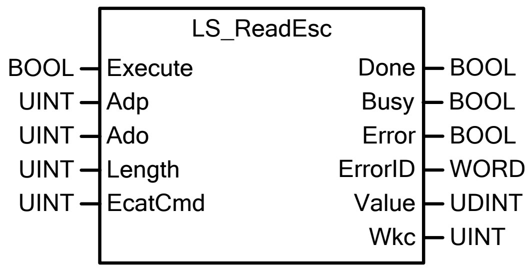
(4) If the size of MasterPoint / SlavePoint / CamCurveSel array is set to a value smaller than CamPointNum, an “error 16#000B” occurs.

(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. Setting an incorrect value will cause "Error 16#000B".

0(mcImmediately) : Itchanges the (Upward Edge of Execute input) parameter value upon executing the function block. If the axis is in operation, the motion may be affected.

1(mcQueued) : It is changed at the same point of time as in "Buffered" of Buffermode.

6.6.10 Read ESC (LS_ReadEsc)

Motion Function Block		
		
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	Indicate 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(Working Counter)	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(Address position) is designating the address of the EtherCAT slave device. The following values can be set depending on the EcatCmd setting. 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: 64th slave connected
4 (FPRD)	1 ~ 64: slave 1~slave 64
7 (BRD)	-

(4) (Length can be set to between 1 and 4, which means 1-4 bytes. Setting a value outside the above range will cause "Error 0x0F61. "

(5) At EcatCmd, set the type of command to use when reading ESC (EtherCAT Slave Controller). One of the following commands can be used: Setting a value outside the above range at EcatCmd will cause "Error 0x0F62.”

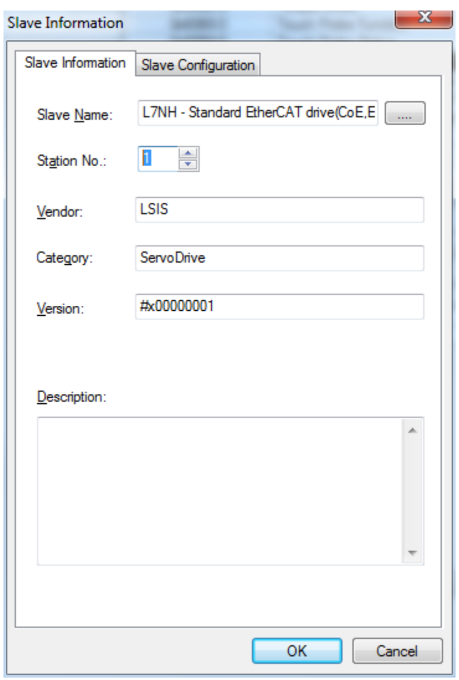
1) 1 - APRD (Auto Increment Physical Read)

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, read motion 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, read motion 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 controller	Setting value
The first slave connected	0
The second slave connected	0xFFFF
:	:
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 – BRD (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 (OR operation of each bit). 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.

6.6.11 Write ESC (LS_WriteEsc)

Motion Function Block

```
                 LS_WriteEsc
BOOL  ─ Execute              Done ─ BOOL
UINT  ─ Adp                  Busy ─ BOOL
UINT  ─ Ado                 Error ─ BOOL
UINT  ─ Length            ErrorID ─ WORD
UINT  ─ EcatCmd               Wkc ─ UINT
UDINT ─ Value
```

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	Indicate 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 designates the EtherCAT slave device address. The following values can be set depending on EcatCmd setting. 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: 64th slave connected
5 (FPWR)	1~64: slave 1~slave 64
8 (BWR)	-

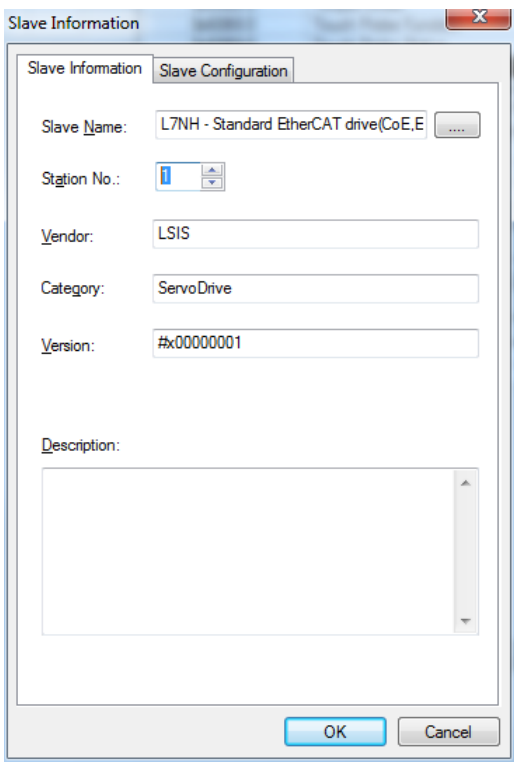
(4) (Length can be set to between 1 and 4, which means 1-4 bytes. Setting a value outside the above range will cause "Error 0x0F71".

(5) At EcatCmd, set the type of command to use when reading ESC (EtherCAT Slave Controller). The following write commands can be used. Setting a value outside the range at EcatCmd will cause "Error 0x0F72”.

1) 2 - APW (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 values depending on the slave device connection order are as follows.

Slave controller	Setting value
The first slave connected	0
The second slave connected	0xFFFF
:	:
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.



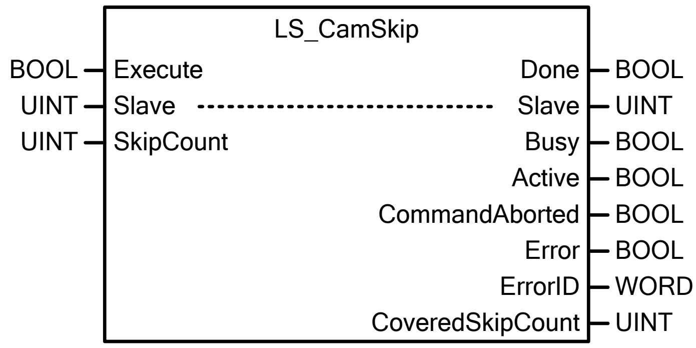
3) 8 –BWR, Broadcast Write
All connected slave devices write data of the size set by Length in the Ado area, and saves the result after Bitwise-OR (OR operation of each bit). The designated address value at Adp is ignored, and Wkc increase by 1 due to all slaves that performed normal write operation.

(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 due to all slaves that performed normal write operation.

(7) After the execution of ESC writes command, if normal data write operation is executed in the designated slave device, Done output is on.

6.6.12 Skip Cam (LS_CamSkip)

Motion Function Block

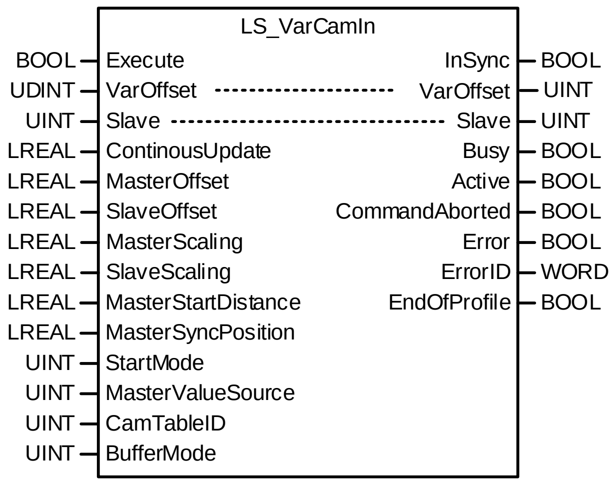


Input - Output		
UINT	Slave	Set the serve axis. (1~32: real/virtual axis, 33~36: virtual axis)
Input		
BOOL	Execute	Give cam skip command on the axis in the rising Edge.
UINT	SkipCount	Set the number of cam cycles to skip.
Output		
BOOL	Done	Indicate the completion state of cam skip operation.
BOOL	Busy	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	Indicate 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 during cam operation, the skip motion starts when the current cam cycle is completed. 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.

6.6.13 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 serve axis. (1~32: real/virtual axis, 33~36: virtual axis)
Input		
BOOL	Execute	Give cam operation command on the 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 serve axis cam table.
LREAL	MasterScaling	Specify the magnification of the main axis.
LREAL	SlaveScaling	Specify the magnification of the serve axis cam table.
LREAL	MasterStartDistance	Specify the position of the main axis where cam operation of the slave.
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 starting point (mcRelative)

UINT	MasterValueSource	Select the source of the main axis for cam operation. 0 : Synchronized in the target value of the main axis. 1 : Synchronized in the current value of the serve axis.
UINT	CamTableID	Specify the cam table to operate.
UINT	BufferMode	Specify the sequential operation setting of motion function block. (Refer to 6.1.4.BufferMode)
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	Indicate 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.

6.6.14 Variable gear operation (LS_VarGearIn)

Motion Function Block

LS_VarGearIn

Type	Input	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		

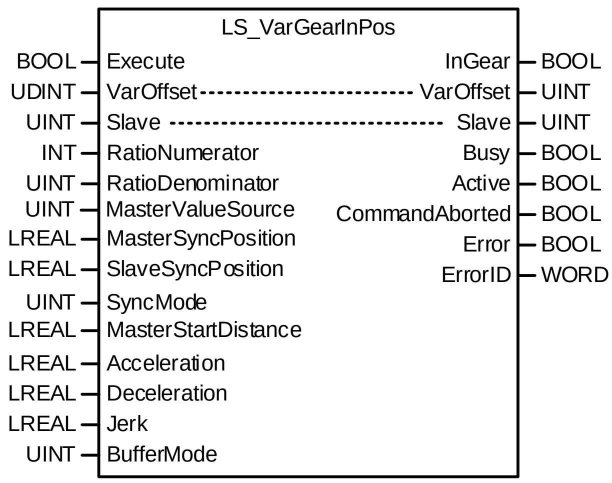
Input - Output		
UDINT	VarOffset	Set the offset value of the M device where the variable to be used as the main axis is located.
UINT	Slave	Set the serve axis. (1~32: real/virtual axis, 33~36: virtual axis)
Input		
BOOL	Execute	Give gear operation command to the relevant axis in the rising Edge.
BOOL	ContinuousUpdate	Specify the update setting of input value. (Refer to 6.1.5.Changes in Parameters during Execution of Motion Function Block)
LREAL	RatioNumerator	Specify the numerator of gear ratio. (-32768 ~ 32767)
LREAL	RatioDenominator	Specify the denominator of gear ratio. (0 ~ 65535)
LREAL	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)

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	Indicate 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.

6.6.15 Variable positioning gear operation (LS_VarGearInPos)

Motion Function Block



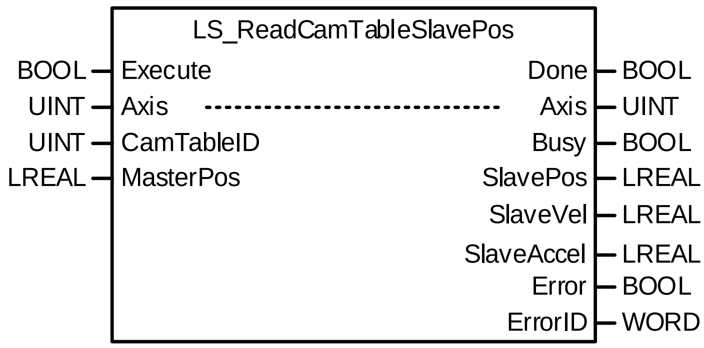
Input - Output		
UDINT	Master	Set the main axis. (1~32: real/virtual axis, 33~36: virtual axis)
UINT	Slave	Set the serve axis. (1~32: real/virtual axis, 33~36: virtual axis)
Input		
BOOL	Execute	Give gear operation command to the relevant axis in the rising Edge.
INT	RatioNumerator	Specify the numerator 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.
UINT	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	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	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	Indicate 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.

6.6.16 Read the slave location of the CAM table (LS_ReadCamTableSlavePos)

Motion Function Block

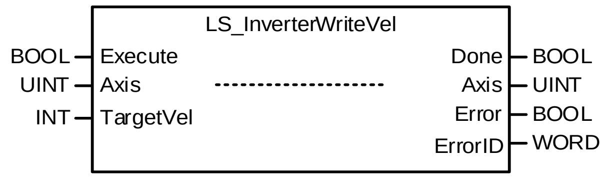


Input - Output		
UINT	Axis	Specify the axis to be commanded (1~32: real/virtual axis, 33~36: virtual axis)
Input		
BOOL	Execute	Give ReadCamTableSlavePos operation command to the relevant axis in the rising Edge.
UINT	CAM tableID	Specify the number of the CAM table to read (1~32)
LREAL	MasterPos	Specify 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	Indicate 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.

6.6.17 Write inverter speed (LS_InverterWriteVel)

Motion Function Block



Input - Output		
UINT	Axis	Specify the axis to be commanded (1~32: 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	Indicate whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

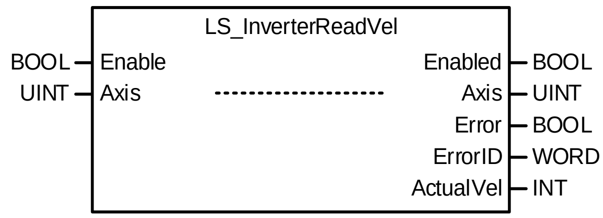
(1) This motion function block is the function block that sets the speed of the inverter to operate when controlling the inverter by the axis

(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.

6.6.18 Read inverter speed (LS_InverterReadVel)

Motion Function Block



Input - Output		
UINT	Axis	Specify the axis to be commanded (1~32: 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	Indicate 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

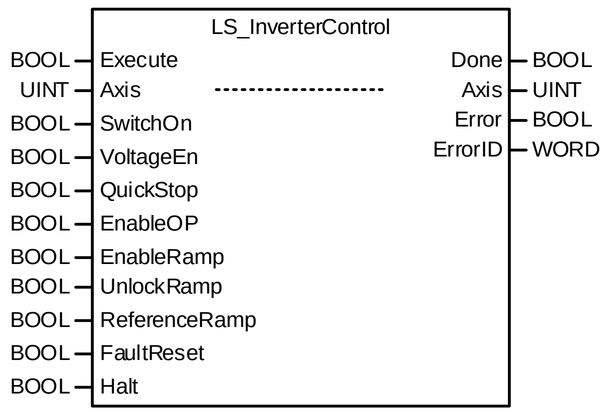
(1) This motion function block is the function block that reads the speed of the connected inverter when controlling the inverter by the axis.

(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.

6.6.19 Write inverter control word (LS_InverterControl)

Motion Function Block



Input - Output		
UINT	Axis	Specify the axis to be commanded (1~32: 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	Indicate 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.

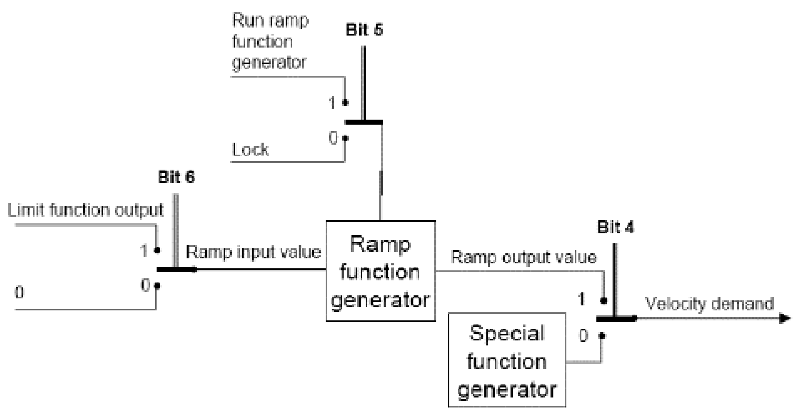
(2) In order to operate the inverter, the control word must be set to enable operation.

(3) Please refer to the following.

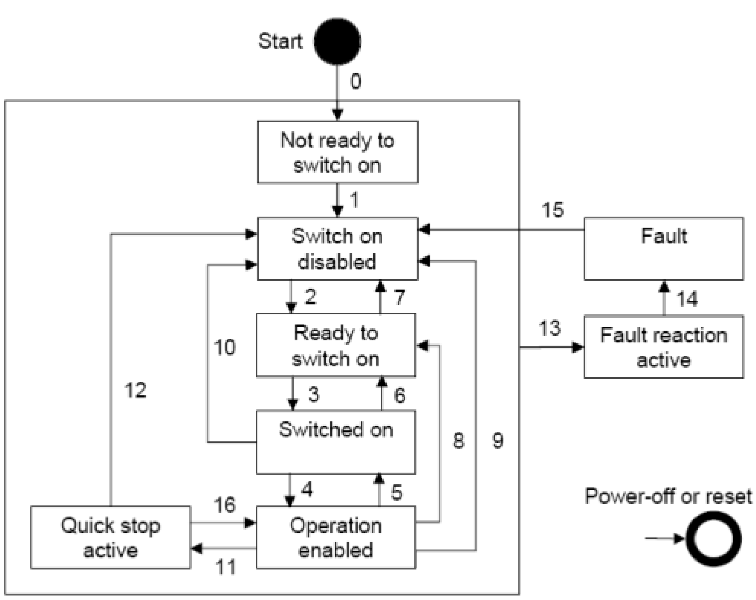
Command bit used in Enable Operation

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

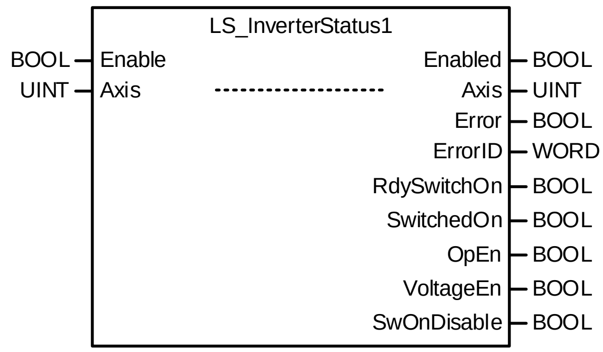


Change the inverter status according to the bit setting of the control word



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.						

6.6.20 Read inverter status 1 (LS_InverterStatus1)

Motion Function Block		
		
Input - Output		
UINT	Axis	Specify the axis to be commanded (1~32: 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	Indicate whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion 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.

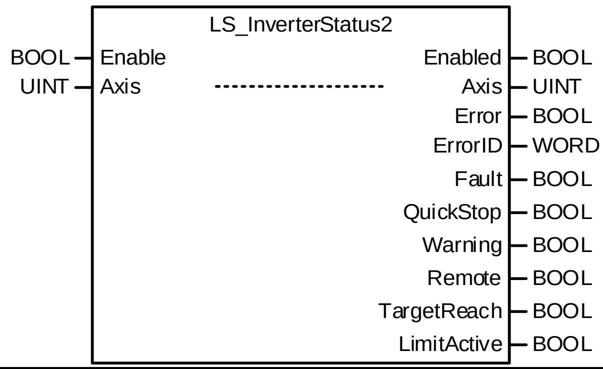
(2) RdySwitchOn, SwitchedOn, OpEn, VoltageEn, SwOnDisable are respectively the lower bit values of the Status Word among the inverter PDO Data.

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

6.6.21 Read inverter status 2 (LS_InverterStatus2)

Motion Function Block



Input - Output		
UINT	Axis	Specify the axis to be commanded (1~32: 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	Indicate whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.
BOOL	Fault	Fault(trip)
BOOL	QuickStop	Quick stop
BOOL	Warning	Warning
BOOL	Remote	Remote
BOOL	TargetReach	Target Reached
BOOL	LimitActive	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.

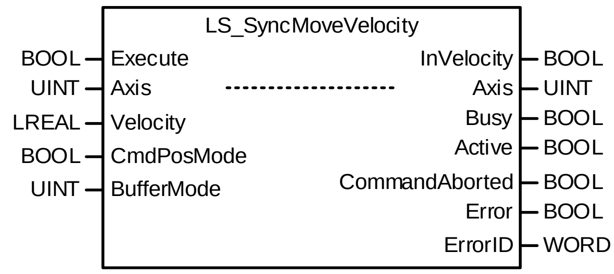
(2) Fault, QuickStop, Warning, Remote, TagetReach, LimiActive are respectively the lower bit values of the Status Word among the inverter PDO Data.

Fault	Bit 3
QuickStop	Bit 5
Warning	Bit 7
Remote	Bit 6
TargetReach	Bit 10
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

6.6.22 Speed control operation (CSV mode) (LS_SyncMoveVelocity)

Motion Function Block



Input - Output		
UINT	Axis	Specify the axis to be commanded (1~32: real axis)
Input		
BOOL	Execute	In the rising Edge, it performs speed control operation through the CSV mode.
BOOL	CmdPosMode	0: Apply the current position to the command position.
UINT	BufferMode	Specify the sequential operation setting of motion function block. (0: Aboritng, 1: Buffered, 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	Indicate 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 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. When the sign of the operation speed value is positive (+ or no sign), it moves in the forward direction and when it is negative (-), it moves in the reverse direction.

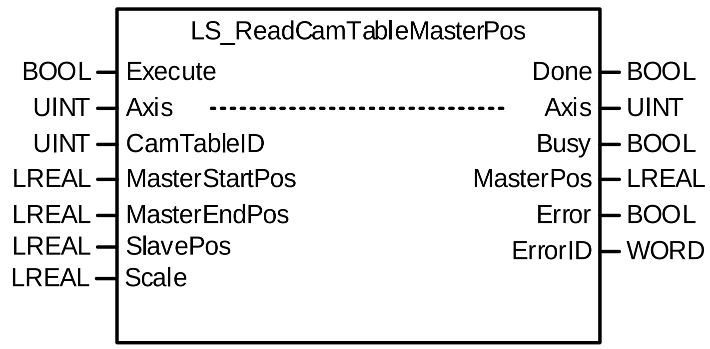
(4) CmdPosMode is used to set the update methods of the current position at the time of command. Only the initial value of 0 is available and the current position of the command is updated using the feedback current position.

(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'.

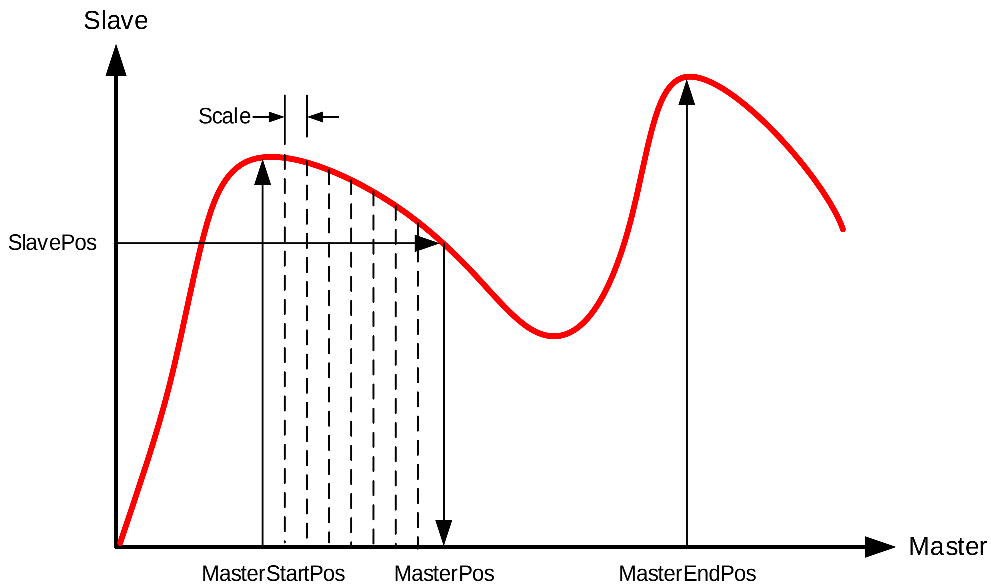
6.6.23 Read CAM table master position (LS_ReadCamTableMasterPos)

Motion Function Block



Input - Output		
UINT	Axis	Set the command axis. (1~32: real axis/virtual axis, 33~36: virtual axis)
Input		
BOOL	Execute	Give cam table master position reading command to the relevant axis in the rising Edge.
UINT	CamTableID	Set the number of 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	Indicated that the execution of motion function is not completed.
LREAL	MasterPos	Output the position of the slave
BOOL	Error	Indicate 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.

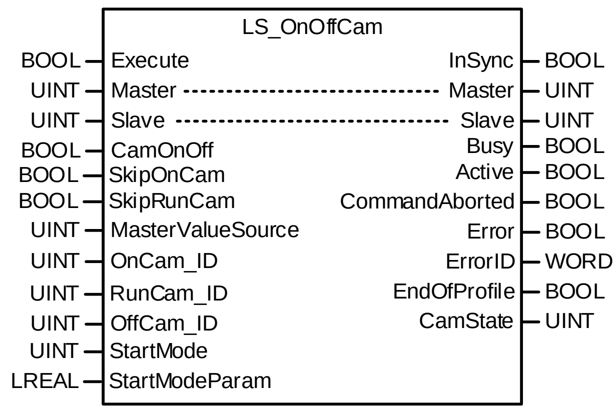


(2) Set the position of serve axis to read in the cam table as SlavePos value. Offset/Gear ratio/Phase correction operation applied to the command axis is not reflected in the MasterPos output.
(3) When the cam table master position reading operation is completed, the Done output turns on.
(4) The 'Scale', 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.

Category / Product	Module O/S	XG5000
XMC-E32A	V1.10	V4.23

6.6.24 OnOff CAM Operation (LS_OnOffCam)

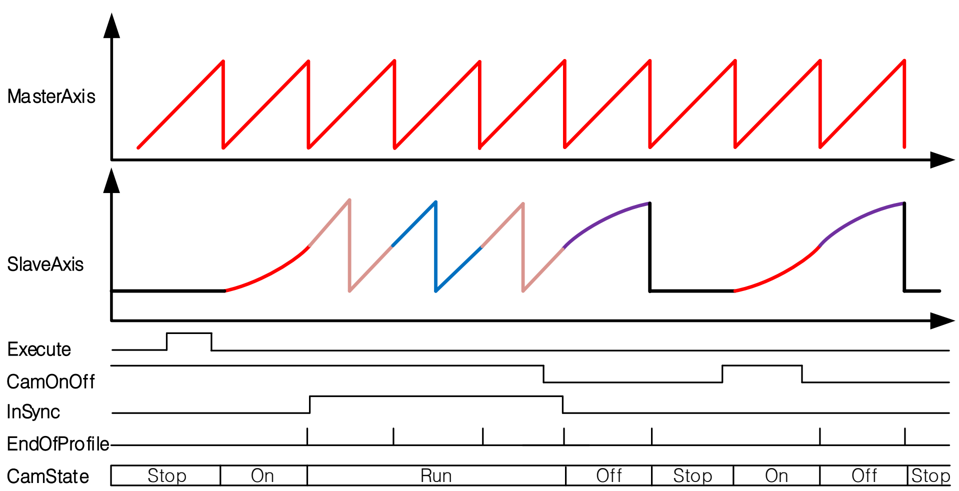
Motion Function Block



Input-Output		
UINT	Master	Set the main axis. (1-32: real/virtual axis, 33-36: virtual axis, 1001-1002: Encoder)
UINT	Slave	Set the serve axis. (1-32: real/virtual axis, 33-36: virtual axis)
Input		
BOOL	Execute	Give the OnOff cam operation command to the relevant axis on the rising Edge.
BOOL	CamOnOff	Set the on/off state of the cam operation. 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.
UINT	StartMode	Specify the method for starting 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	Indicates that the execution of the motion function block is not completed.
BOOL	Active	Indicates that the current motion function block is controlling the relevant axis.
BOOL	CommandAborted	Indicates that the current motion function block is interrupted by another command.
BOOL	Error	Indicates whether an error occurs or not.
WORD	ErrorID	Outputs the error ID that occurred while the motion function block is running.
BOOL	EndOfProfile	Indicates the end of the current cam operation.
UINT	CamState	0: Stop state 1: Executing OnCam 2: Executing RunCam 3: Executing OffCam

(1) This motion function block uses three cam tables to carry out the cam operation that is switched to a Stop state->OnCam->RunCam or a RunCam->OffCam->Stop state depending on the CamOnOff input.



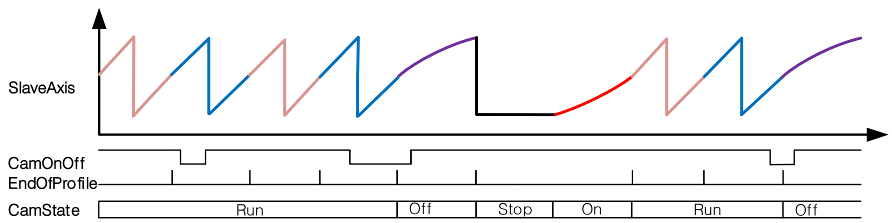
(2) The cam operation runs under a state where Execute is the rising Edge. The cam operation does not stop even if Execute is changed to Off during the operation. To stop the OnOffCam operation, you must give the MC_CamOut command or run another motion function block.

(3) If StartMode is set to 0, OnCam runs as soon as 1 is input in CamOnOff. If StartMode is set to 1, OnCam does not run as soon as 1 is input in CamOnOff, but 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 the main axis then moves in 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) EndOfProfile outputs On when passing the end of a profile during the operation of each OnCam/OffCam/RunCam cam profile.

(6) If the CamOnOff signal is Off, the operation to switch to RunCam->OffCam->Stop state is performed. 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 an 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 executing RunCam after executing OnCam. 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 outputs On.

(12) MasterValueSource selects the source of the main axis for synchronization. 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. Before executing RunCam in a Stop state, set the cam profile to run as 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 corresponding axis is in a "SynchronizedMotion" state when this motion function block is running.

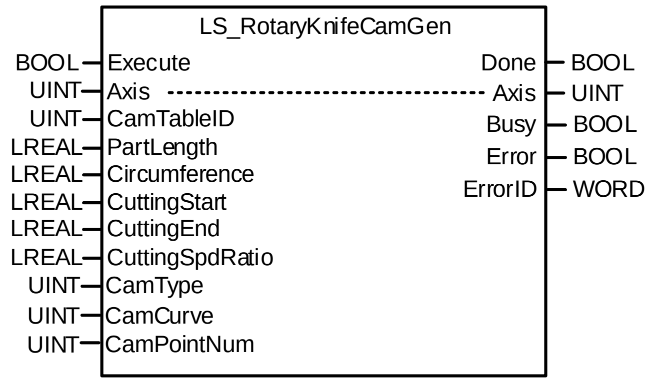
(16) For more information, see Chapter 8.6 RotaryKnife Operation under Chapter 8 Motion Control Function.

(17) This motion function block is supported in the following versions:

Category / Product	Module O/S	XG5000
XMC-E32A	V1.20	V4.25

6.6.25 RotaryKnife cam profile generation (LS_RotaryKnifeCamGen)

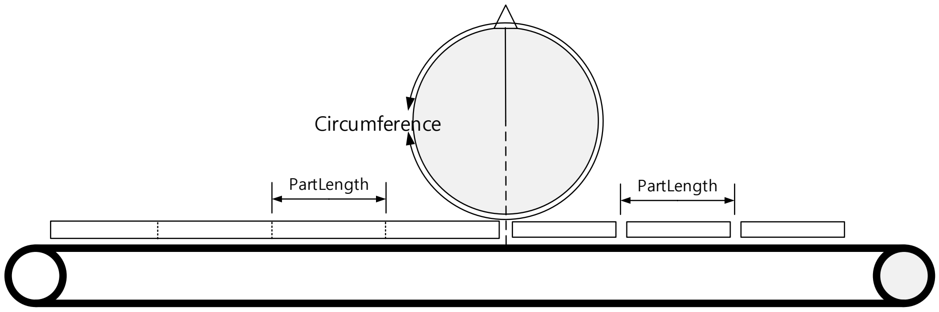
Motion Function Block



Input-Output		
UINT	Axis	Specify the axis to give the command. (1-32: real/virtual axis, 33-36: virtual axis)
Input		
BOOL	Execute	Performs cam profile generation in the rising Edge.
UINT	CamTableID	Set the cam table ID where the profile is stored.
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.)

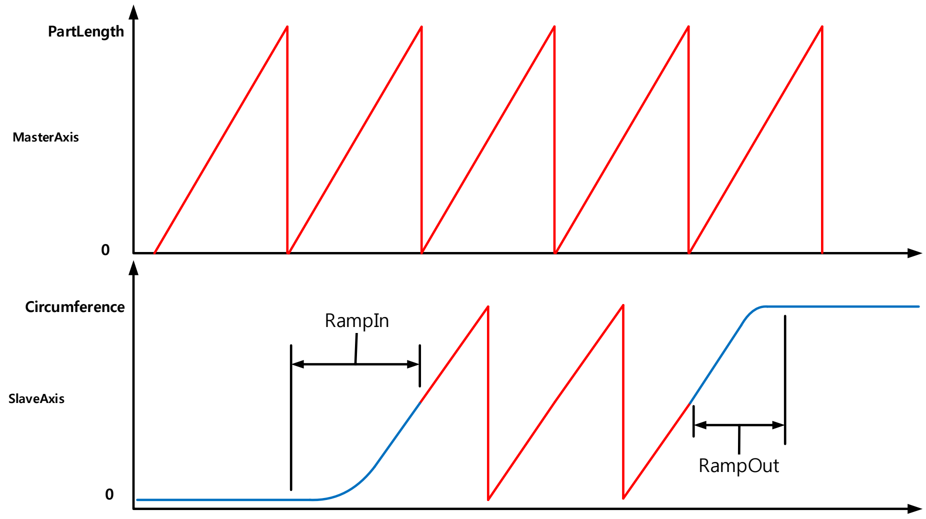
(6) On the generated cam profile, the movement amount of the main axis is 360Degree in ratio to PartLength. This means that you must set the gear ratio of the motor and the machine in the parameter so that 1 rotation of the main axis equals PartLength.

(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.

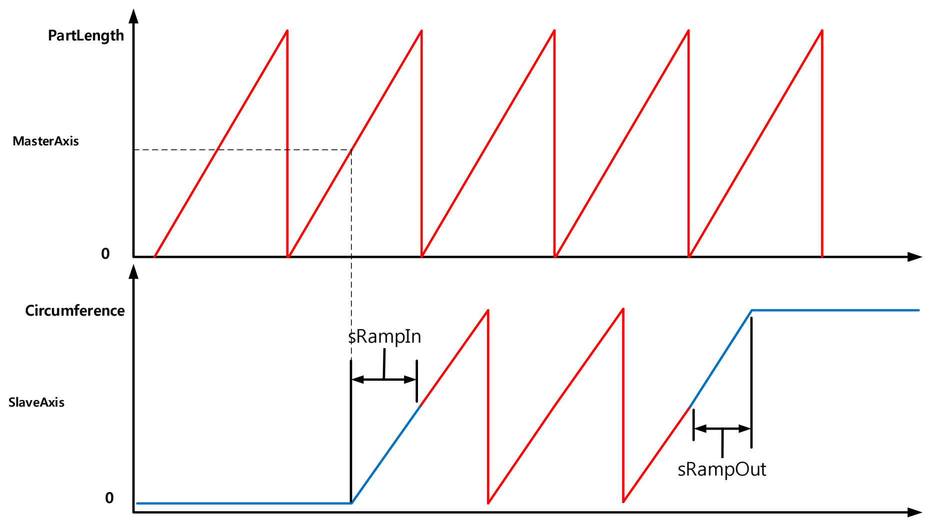
(8) For CuttingStart, you cannot enter a value that is less than 1/8 of the Circumference or greater than CuttingEnd. A "0x1172" error occurs if there is an error in the CuttingStart value.

(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.

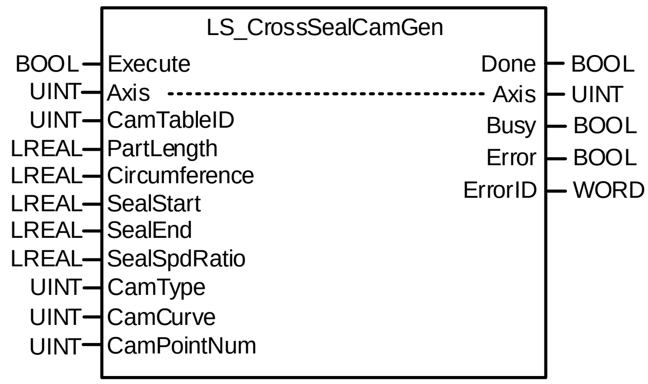
(14) The minimum number of cam points required for CamPointNum is 10 and a "0x1177" error occurs if there is an error in the CamPointNum value.

(15) This motion function block is supported in the following versions:

Category / Product	Module O/S	XG5000
XMC-E32A	V1.20	V4.25

6.6.26 Cross sealer cam profile generation (LS_CrossSealCamGen)

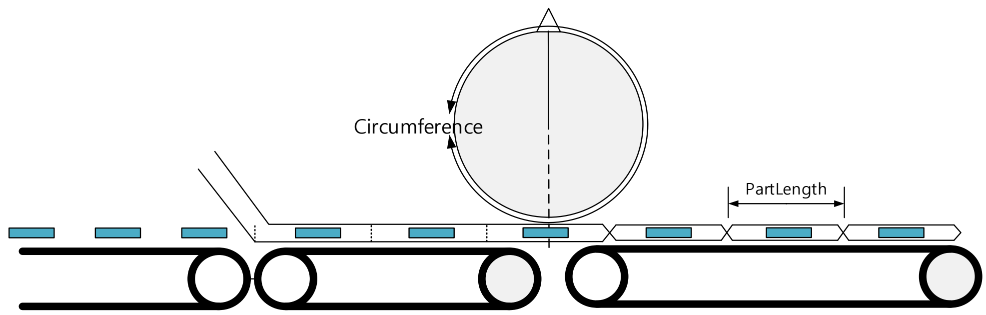
Motion Function Block



Input-Output		
UINT	Axis	Specify the axis to give the command. (1-32: real/virtual axis, 33-36: virtual axis)
Input		
BOOL	Execute	Performs cam profile generation in the rising Edge.
UINT	CamTableID	Set the cam table ID to store the cam profile.
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 SealStart input, enter the starting position for the cross sealer to end 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.)

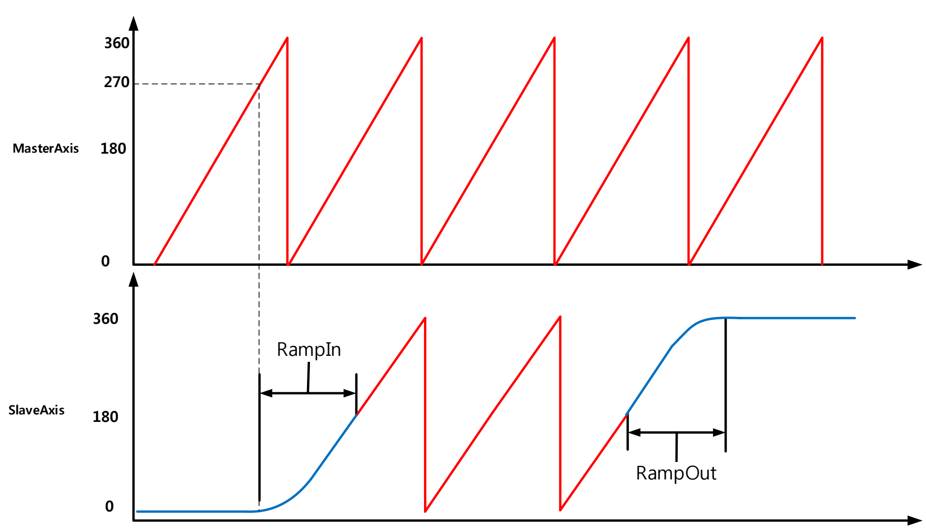
(6) On the generated cam profile, the movement amount of the main axis is 360 in ratio to PartLength. This means that you must set the gear ratio of the motor and the machine in the parameter so that when the main axis moves 360, the real distance equals PartLength.

(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.

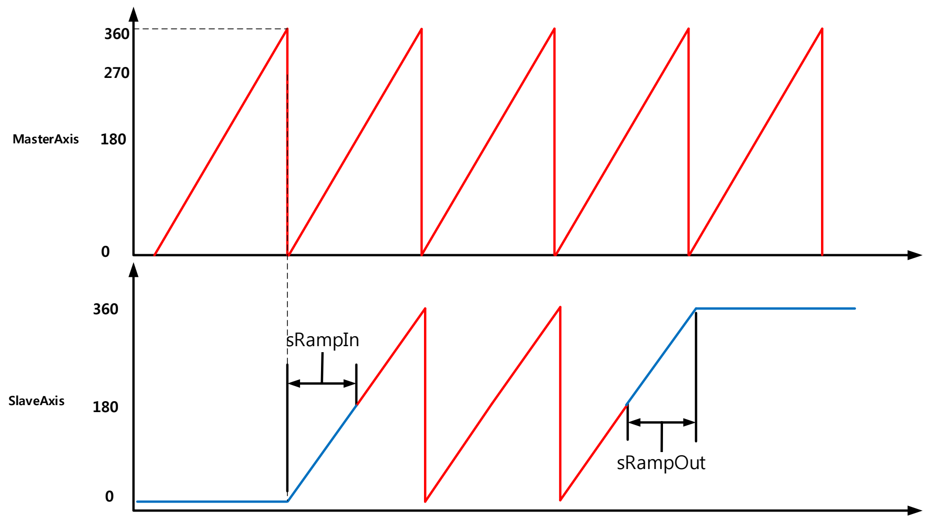
(9) For SealEnd, you cannot enter a value that is greater than 7/8 of the Circumference or smaller than SealEnd. A "0x1172" error occurs if there is an error in the SealEnd value. To set the sealing region to the minimum, set SealEnd and SealStart as equal values.

(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. 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.

(15) The minimum number of cam points required for CamPointNum is 10 and a "0x1177" error occurs if there is an error in the CamPointNum value.

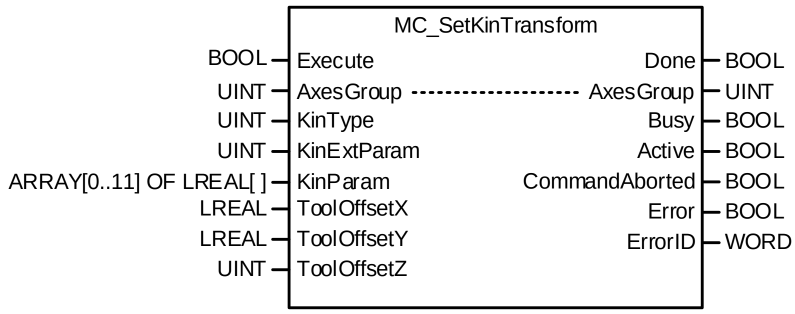
(16) This motion function block is supported in the following versions:

Category / Product	Module O/S	XG5000
XMC-E32A	V1.20	V4.25

6.7 Coordinate System Operation Function Block

6.7.1 Machine information setting(MC_SetKinTransform)

Motion Function Block

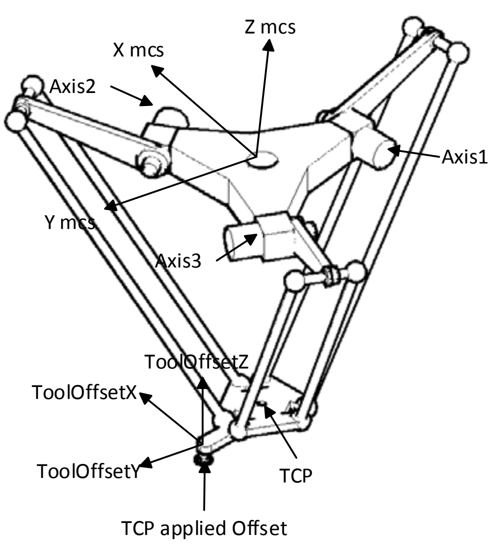


Input-Output		
UINT	AxesGroup	Set the axes group to set the machine information.(1 ~ 16 : Group 1 ~ Group 16)
Input		
BOOL	Execute	Give the machine information setting command on the axis in the rising Edge.
UINT	KinType	Set the machine type.(0:XYZ, 1:Delta3)
UINT	KinExtParam	Unused
LREAL[]	KinParam	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	Indicate whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

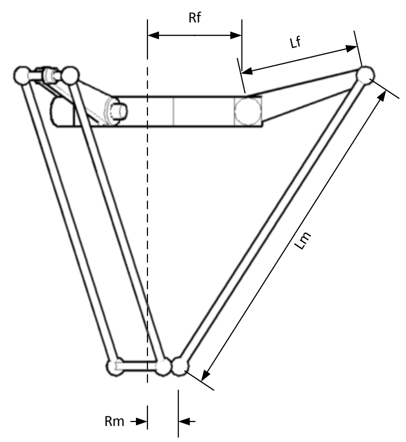
(1) This motion function block sets the ACS and MCS conversion based on the machine model defined in advance at AxesGroup.
(2) The same setting can be applied to the XG5000 group parameter settings.

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
(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.

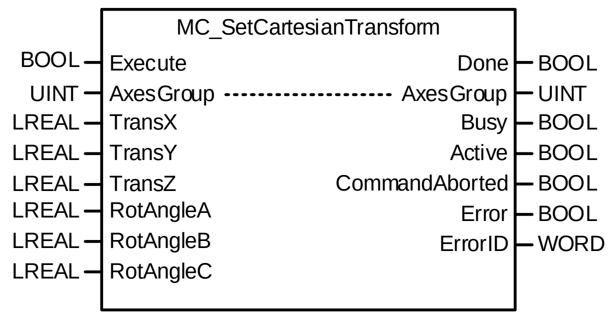


(6) When using Delta3, the device setting information is as follows. For more information, refer to 8.4.4 Machine information setting.

	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



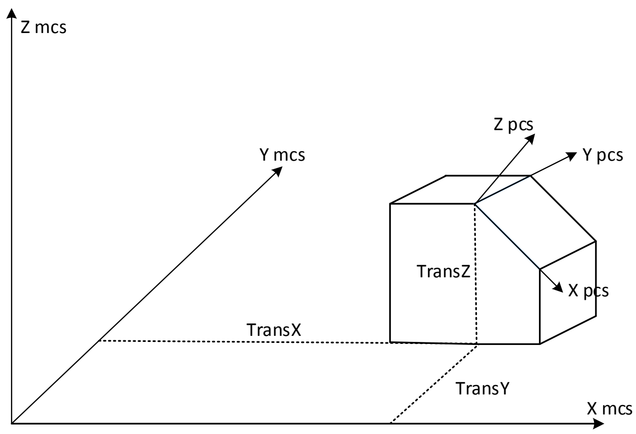
Input-Output		
UINT	AxesGroup	Set the axes group to set the PCS.(1 ~ 16 : Group 1 ~ Group 16)
Input		
BOOL	Execute	Give the PCS setting command on the axes group in the rising Edge.
LREAL	TransX	Movement from MCS to X Axis(mm)
LREAL	TransY	Movement from MCS to Y Axis(mm)
LREAL	TransZ	Movement from MCS 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	Indicate whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

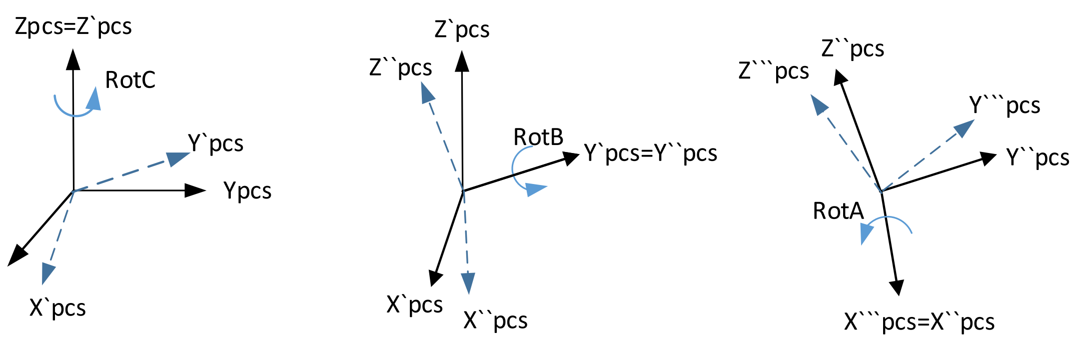
(1) This motion function block sets the perpendicular coordinate conversion between MCS and PCS at AxesGroup.

(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

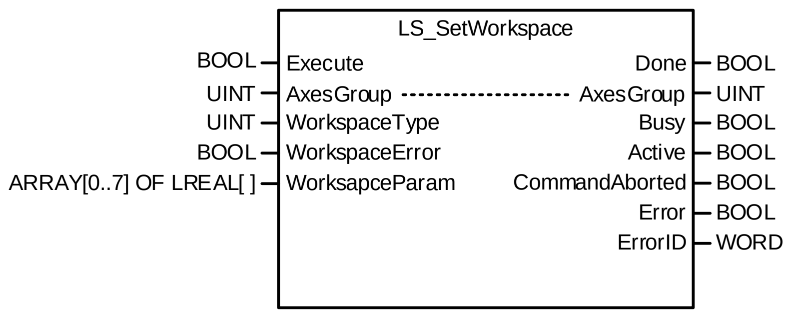
(3) TransX/TransY/TransZ is the move distance from MCS origin to PCS origin. RotA/RotB/RotC is the rotation value of PCS, RotA is the value that rotates PCS on the X-axis of PCS, RotB is the value that rotates PCS on the Y-axis of PCS, RotC is the value that rotates PCS on the Z-axis of PCS. The rotation 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 work space.(1 ~ 16 : Group 1 ~ Group 16)
Input		
BOOL	Execute	Give the work space setting command on the axes group in the rising Edge.
UINT	WorkspaceType	Set the work space type (1:Rectangle 2:Cylinder 3:Delta3 4: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	Indicate whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

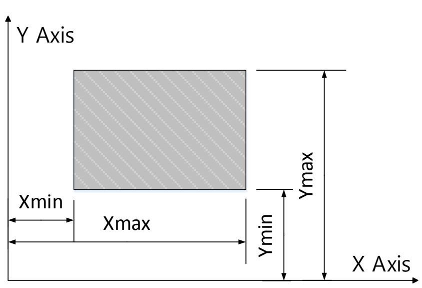
(1) This motion function block sets the work space based on the coordinate system at the axes group designated by AxesGroup input.

(2) The same setting can be performed in XG5000 group parameter setting.

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

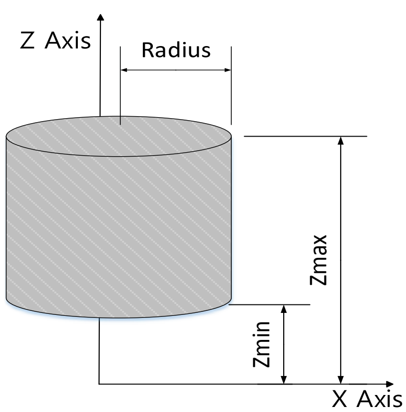
(3) WorkspaceType can be selected from 4 types (1: Rectangle 2: Cylinder 3: Delta3 4: Sector).
(4) WorkspaceError input determines whether an error occurs when a coordinate system operation exceeds the work space.
(5) WorkspaceParam input sets the parameters depending on the work space type.
(6) The 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



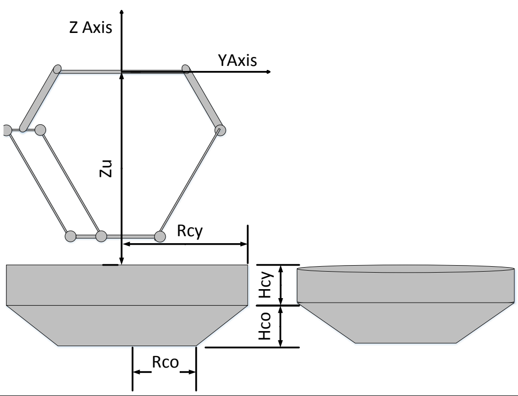
Parameter	value
WorkspaceParam[0]	X max(mm)
WorkspaceParam[1]	X min(mm)
WorkspaceParam[2]	Y max(mm)
WorkspaceParam[3]	Y min(mm)
WorkspaceParam[4]	Z max(mm)
WorkspaceParam[5]	Z min(mm)

2) Cylinder



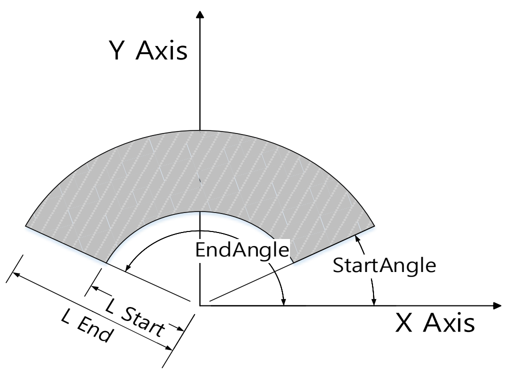
Parameter	value
WorkspaceParam[0]	Radius(mm)
WorkspaceParam[1]	Z max(mm)
WorkspaceParam[2]	Z min(mm)

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]	-

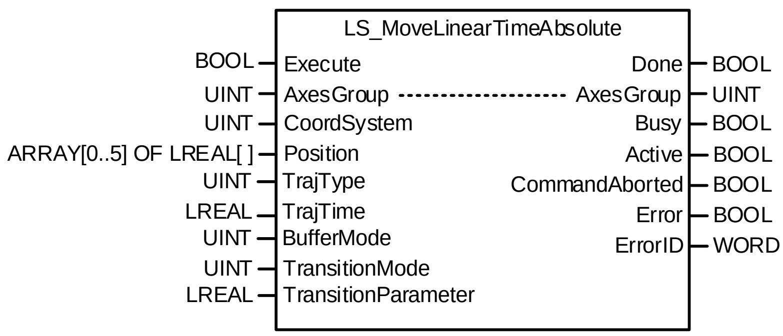
4) Sector



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 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 : Group 1 ~ Group 16)
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)
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	Indicate 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.

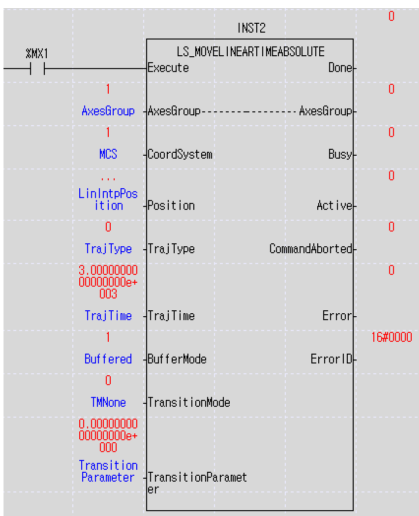
(4) TrajTime sets the time taken to reach the target position.

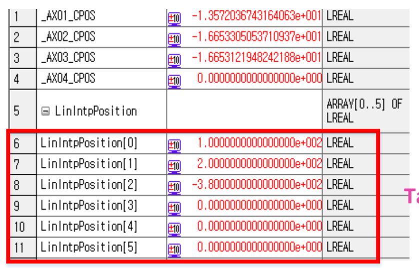
(5) Please refer to 8. 4. 6 Coordinate System Absolute Position/Time Linear Interpolation Control for further details.

(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

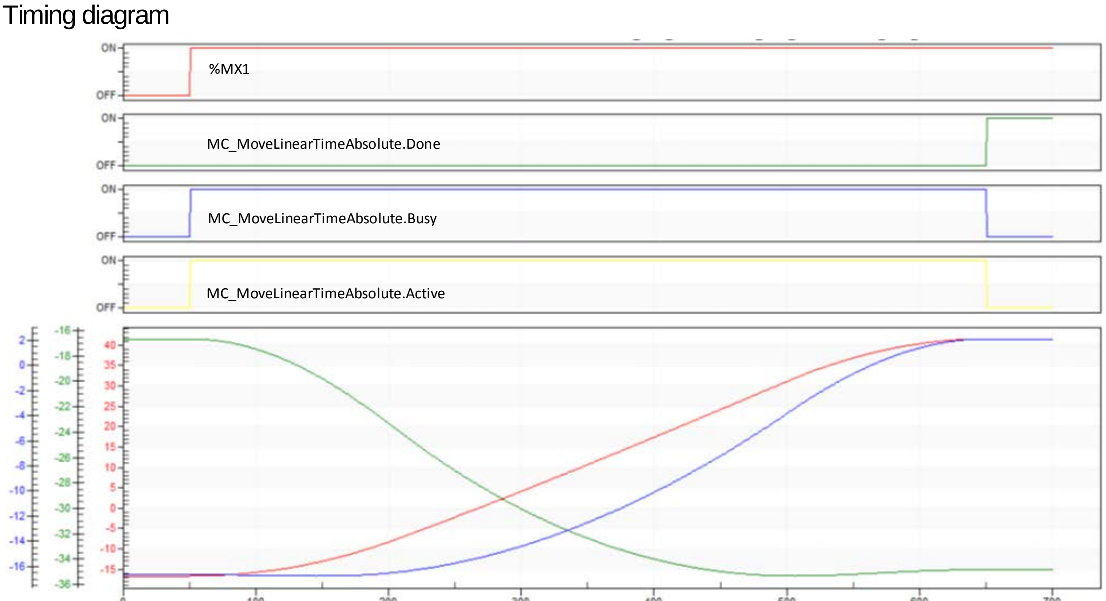




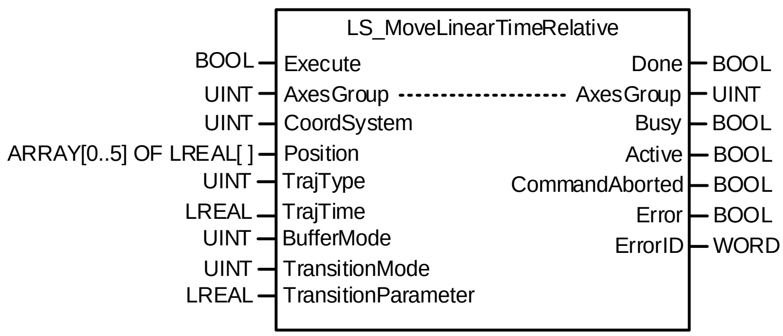
1	_AX01_CPOS	-1.3572036743164063e+001	LREAL
2	_AX02_CPOS	-1.6653305053710937e+001	LREAL
3	_AX03_CPOS	-1.6653121948242188e+001	LREAL
4	_AX04_CPOS	0.0000000000000000e+000	LREAL
5	⊟ LinIntpPosition		ARRAY[0..5] OF LREAL
6	LinIntpPosition[0]	1.0000000000000000e+002	LREAL
7	LinIntpPosition[1]	2.0000000000000000e+002	LREAL
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

Target position

(b) Timing diagram



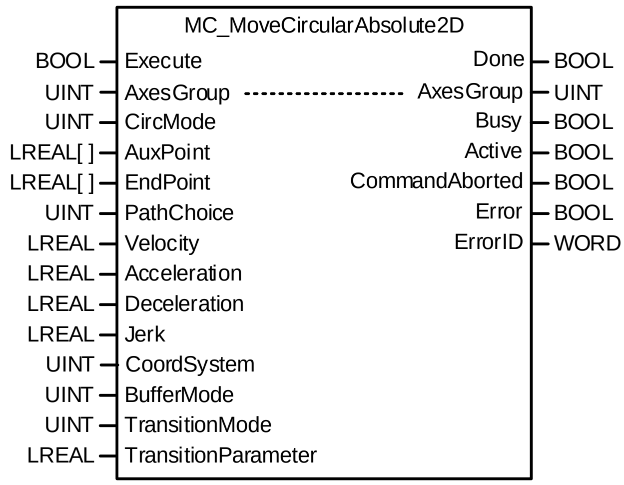
6.7.5 Time-linear interpolation operation for relative position of coordinate system (LS_MoveLinearTimeRelative)

Motion Function Block		
		
Input-Output		
UINT	AxesGroup	Set the axes group to set the relative position time linear interpolation.(1 ~ 16 : Group 1 ~ Group 16)
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	Give the sequential operation of the motion function block. (Refer to the 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	Indicate 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 inputs set 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 Relative Position/Time Linear Interpolation Control for further details.

6.7.6 Circular interpolation operation for absolute position of coordinate system (MC_MoveCircularAbsolute2D)

Motion Function Block



Input-Output

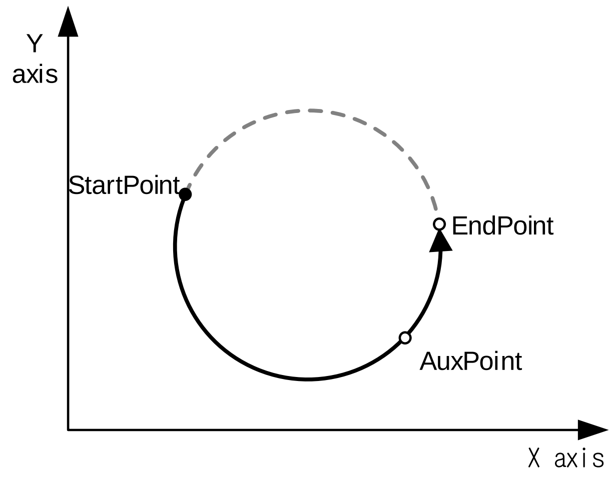
UINT	AxesGroup	Set the axes group to set the absolute position circular interpolation.(1 ~ 16 : Group 1 ~ Group 16)

Input

BOOL	Execute	Give the circular interpolation command on the axes group in the rising Edge.
UINT	CircMode	The way to set the circular interpolation [0: Middle point Aux point, 1: Center point, 2: Radius]
LREAL[ ]	AuxPoint	The auxiliary point position for circular interpolation is designated as an absolute coordinate.
LREAL[ ]	EndPoint	Set the circular end point as an absolute coordinate.
BOOL	PathChoice	Set the circular path. 0: clockwise direction, 1: counter-clockwise direction
LREAL	Velocity	Set the maximum velocity of the path.. [u/s]
LREAL	Acceleration	Set the maximum acceleration. [u/s^2]
LREAL	Deceleration	Set the minimum deceleration. [u/s^2]
LREAL	Jerk	Set the maximum acc/dec jerk. [u/s^3]
UINT	CoordSystem	Set the coordinate system's type. (1:MCS 2:PCS)
UINT	BufferMode	Specify the sequential operation setting of motion function block. (Refer to 6.1.4.BufferMode)
UINT	TransitionMode	Unused
LREAL	TransitionParameter	Unused

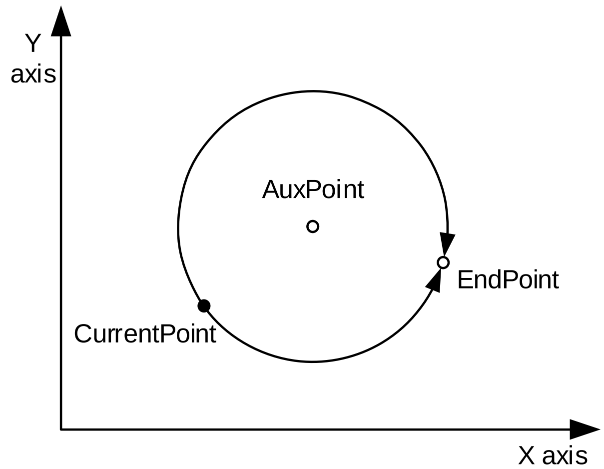
Output		
BOOL	Done	Indicate whether to reach the specified point.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Active	Indicate that whether or not motion function block is controlling the group.
BOOL	CommandAborted	Indicate that the current motion function block is interrupted while it is running.
BOOL	Error	Indicate 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) At AuxPoint and EndPoint input, designate the arrangement of the absolute position of auxiliary points to refer to for circular interpolation of each axis. The input corresponds in the order of X, Y, Z, unlike MC_MoveCircularAbsolute.
(4) Velocity, Acceleration, Deceleration, Jerk input sets the velocity, acceleration, deceleration, and acceleration/deceleration rate change of the interpolation path, respectively.
(5) CircMode input sets the circular interpolation method. 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.



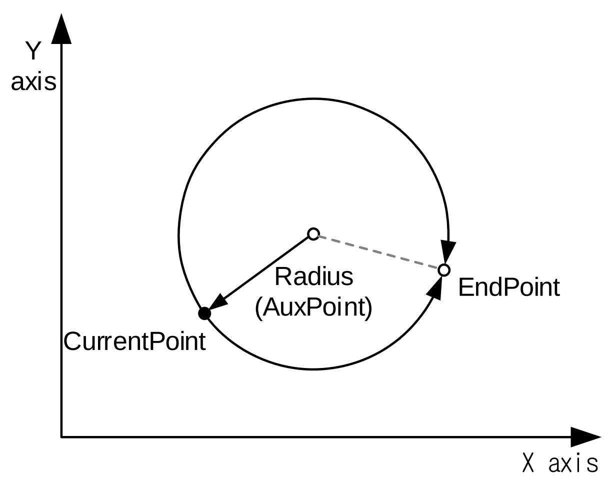
(b) Circular Interpolation Using Center Point Specification (CircMode = 1)

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.

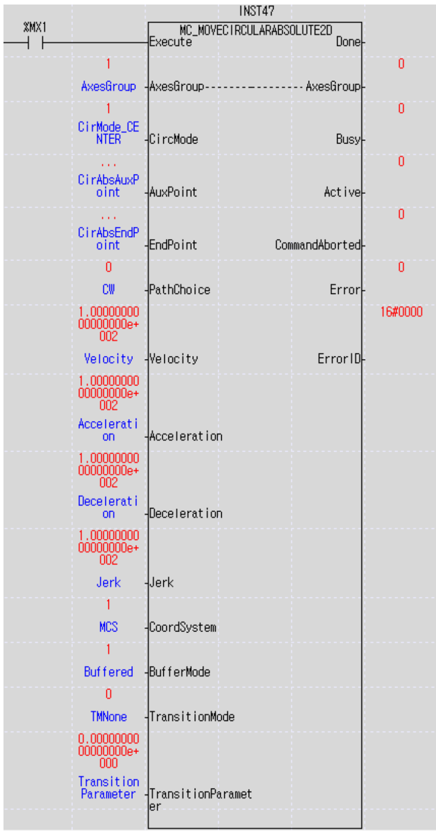
(7) The changed parameters are applied by re-executing the function block (Execute input is On) before the command is completed. Only, Velocity, Acceleration, Deceleration, Jerk, AuxPoint, Endpoint input can be updated.

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

(9) Example program

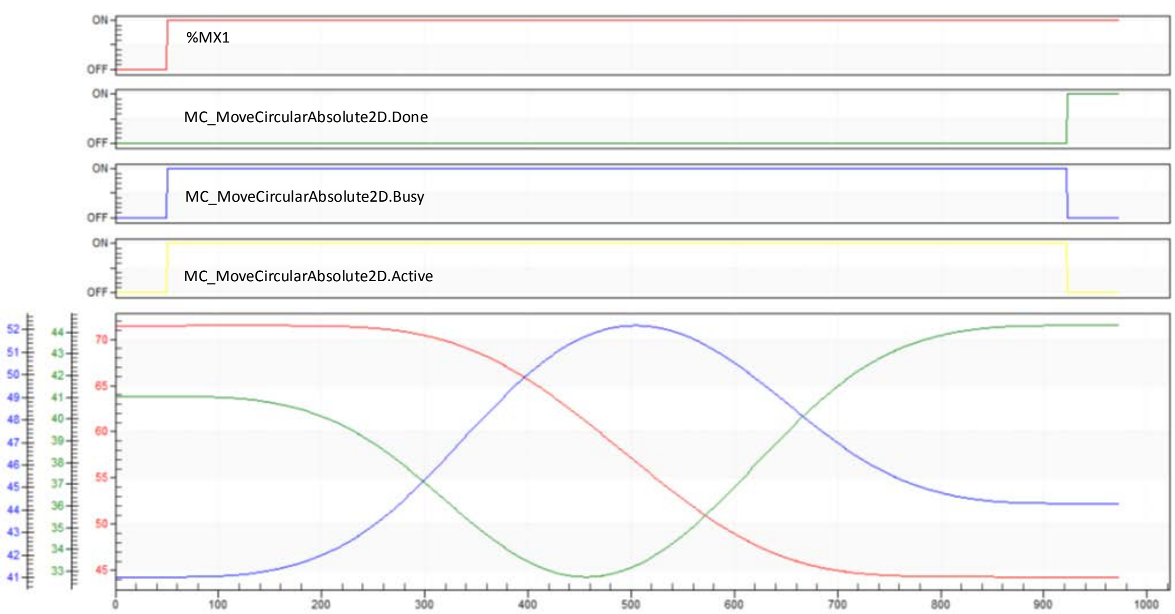
This example is to set the center point at (0, 75, -580) when the current command position is MCS (0,150,-580), and perform circular interpolation to the target position MCS(0,0,-580) by moving in a clockwise direction.

(a) Function block setting



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	CenterPoint
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	EndPoint
10	CirAbsEndPoint[1]	0.0000000000000000e+000	LREAL	
11	CirAbsEndPoint[2]	-5.8000000000000000e+002	LREAL	

(b) Timing diagram



6.7.7 Circular interpolation operation for relative position of coordinate system (MC_MoveCircularRelative2D)

Motion Function Block

MC_MoveCircularRelative2D

Input type	Input	Output	Output type
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 relative position circular interpolation operation. (1 ~ 16: Group 1 ~ Group 16)

Input

BOOL	Execute	Give relative position circular interpolation operation command on the 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 a relative coordinate.
LREAL[ ]	EndPoint	Specify the end point of the circular trajectory as a relative coordinate from the start point.
BOOL	PathChoice	Set the circular path. 0: clockwise direction, 1: counter-clockwise direction
LREAL	Velocity	Set the maximum velocity of the path. [u/s]
LREAL	Acceleration	Set the maximum acceleration. [u/s^2]
LREAL	Deceleration	Set the minimum deceleration. [u/s^2]
LREAL	Jerk	Set the maximum acc/dec jerk. [u/s^3]
UINT	CoordSystem	Set the coordinate system's type. (1:MCS 2:PCS)
UINT	BufferMode	Specify the sequential operation setting of motion function block. (Refer to 6.1.4.BufferMode)

UINT	TransitionMode	Unused
LREAL	TransitionParameter	Unused
Output		
BOOL	Done	Indicate whether to reach the specified point.
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	Indicate 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.

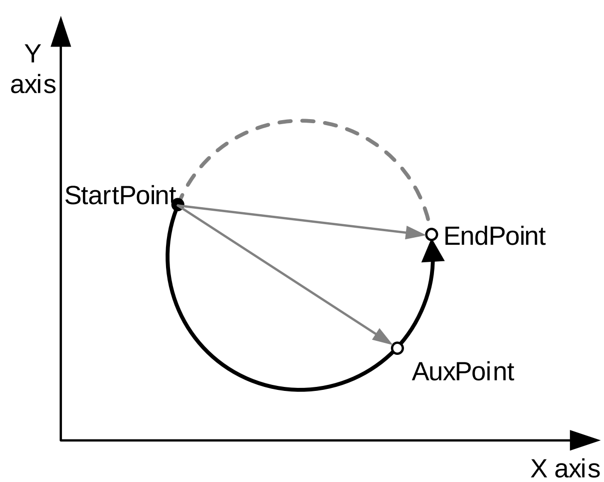
(2) When this motion function block is executed, each axis performs circular interpolation control referring to the auxiliary point input, and the direction is determined by Path Choice input. If PathChoiceinput is set to 0, circular interpolation is operated in a clockwise direction, and if it is set 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. (Since the number of axes comprising a group to issue circular interpolation command is 3, arrangements of three sizes should be input for the Position input.)

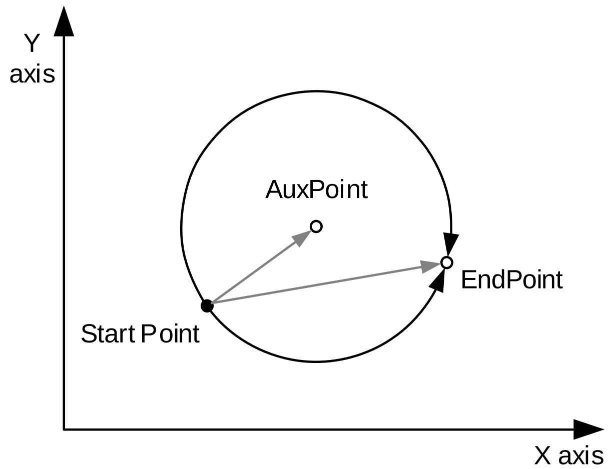
(4) In Velocity, Acceleration, Deceleration, Jerk inputs, the acceleration, deceleration, change rate of acceleration, velocity of the interpolation path are specified, respectively.

(5) CircMode input sets the circular interpolation method. The circular interpolation methods corresponding to CircMode values are as follows.

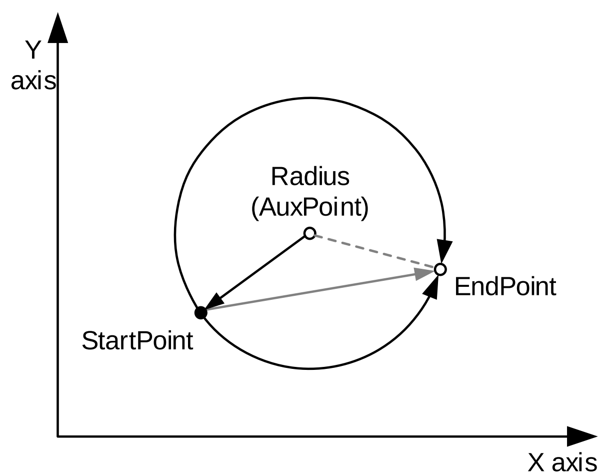
(a) Circular Interpolation Using Midpoint Specification (BORDER, CircMode = 0)
This method is to perform the circular interpolation to the target position through the midpoint position after starting operation at the current position. In the figure below, the current position corresponds to the axes 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 Using Center Point Specification (CircMode = 1)
This method is to perform the circular interpolation to the target position by starting operation at the start 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 relative coordinate input for the EndPoint.

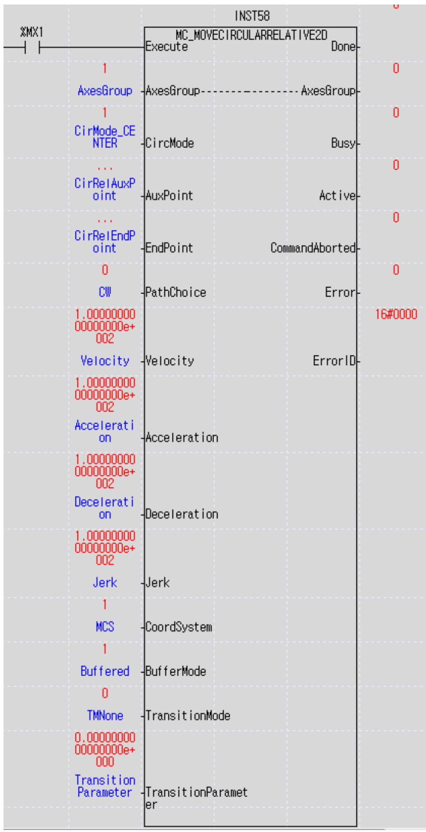


(c) Circular Interpolation using Radius Speciation (CircMode = 2)
This method is to perform the circular interpolation to the target position by starting operation at the current position, passing the designated center point, and reaching the target position. In the figure below, the current position 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 8.4.7 circular interpolation control in motion controller's manual for more details.
(7) The changed parameters are applied by re-executing the function block (Execute input is On) before the command is completed. 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 when the current command position is 1000, 1000 (set the relative position from the center point to set from the current position), and move clock-wise to perform circular interpolation to the target position (set the relative position to the target position from the current position: 0, 0).

(a) Function block setting

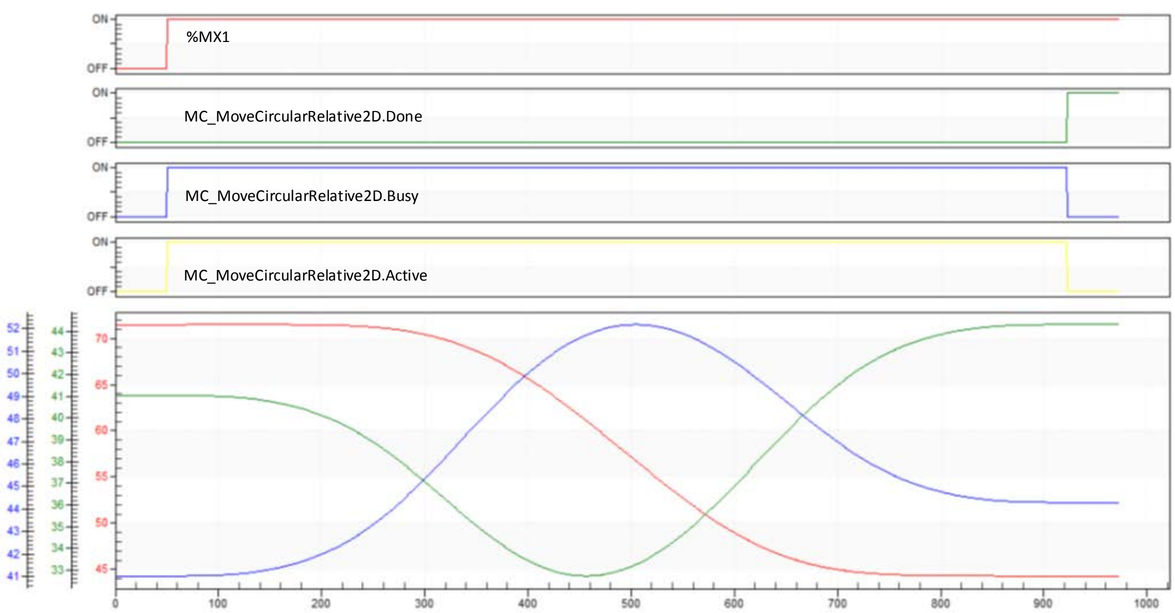


1	_AX01_CPOS	7.1485269050584748e+001	LREAL
2	_AX02_CPOS	4.1026455810822632e+001	LREAL
3	_AX03_CPOS	4.1026455810822632e+001	LREAL
4	⊟ CirRelAuxPoint		ARRAY[0..2] OF LREAL
5	CirRelAuxPoint[0]	0.0000000000000000e+000	LREAL
6	CirRelAuxPoint[1]	-7.5000000000000000e+001	LREAL
7	CirRelAuxPoint[2]	0.0000000000000000e+000	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

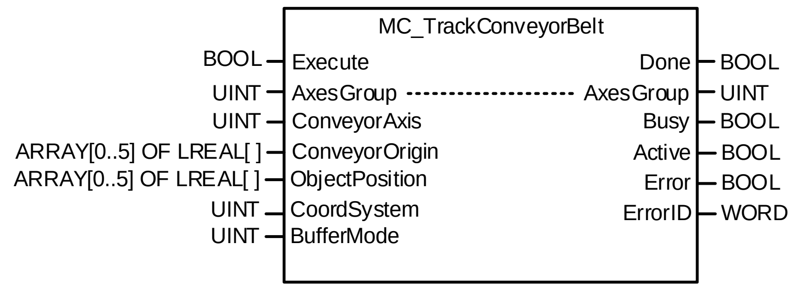
End point

(b) Timing diagram



6.7.8 Synchronization setting of conveyor belt (MC_TrackConveyorBelt)

Motion Function Block



Input-Output		
UINT	AxesGroup	Set the group to do conveyor belt synchronized setting.(1 ~ 16: Group 1 ~ Group 16)
Input		
BOOL	Execute	Give the conveyor belt synchronized setting command on the axes group in the rising Edge.
UINT	ConveyorAixs	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)
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	Indicate whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

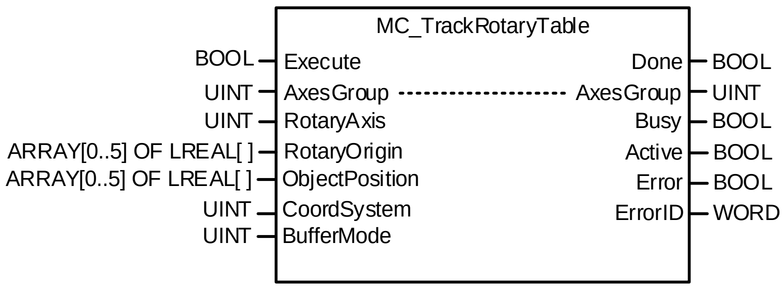
(1) This motion function block sets conveyor belt synchronized operation for the axes group designated by AxesGroup input.

(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

6.7.9 Synchronization setting of the rotary table (MC_TrackRotaryTable)

Motion Function Block

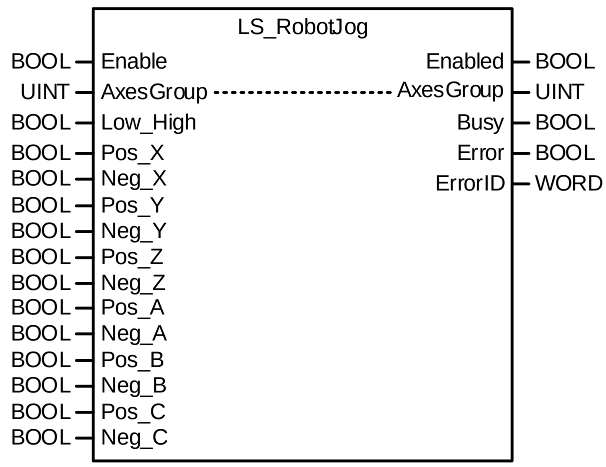


Input-Output		
UINT	AxesGroup	Set the group to do rotary table synchronized setting.(1 ~ 16: Group 1 ~ Group 16)
Input		
BOOL	Execute	Give the rotary table synchronized setting command on the axes group in the rising Edge.
UINT	RotaryAixs	Set the rotary table axis.(1 ~ 32 : Axis 1~Axis 32)
LREAL[]	RotaryOrigin	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)
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	Indicate 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 rotary tablet 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

6.7.10 JOG operation of the coordinate system (MC_RobotJog)

Motion Function Block



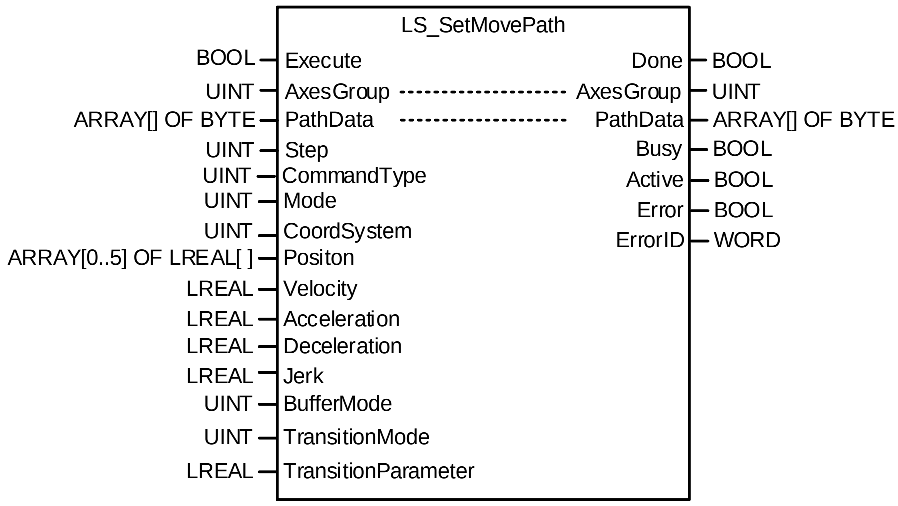
Input-Output		
UINT	AxesGroup	Set the axis group to make the command.(1 ~ 16 : Group 1 ~ Group 16)
Input		
BOOL	Enable	While the input is ON, the JOG operation command is sent to the relevant axis group.
BOOL	Low_High	Set the JOG speed in JOG operation. (0: JOG low-speed operation, 1: JOG high-speed operation)
BOOL	Pos_X	Set the linear operation direction at JOG operation.(X-axis + direction)
BOOL	Neg_X	Set the linear operation direction at JOG operation.(X-axis –direction)
BOOL	Pos_Y	Set the linear operation direction at JOG operation.(Y-axis + direction)
BOOL	Neg_Y	Set the linear operation direction at JOG operation.(Y-axis –direction)
BOOL	Pos_Z	Set the linear operation direction at JOG operation.(Z-axis + direction)
BOOL	Neg_Z	Set the linear operation direction at JOG operation.(Z-axis –direction)
BOOL	Pos_A	Set the rotary operation direction at JOG operation. (X-axis counter-clockwise 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 counter-clockwise 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 counter-clockwise 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	Indicate 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) The JOG operation is a manual operation function for testing. It is used for checking system operations, wiring status, and position address for teaching. It can be respectively applied to both high speed and low speed.
(3) 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.

6.7.11 Set path operation data (MC_SetMovePath)

Motion Function Block



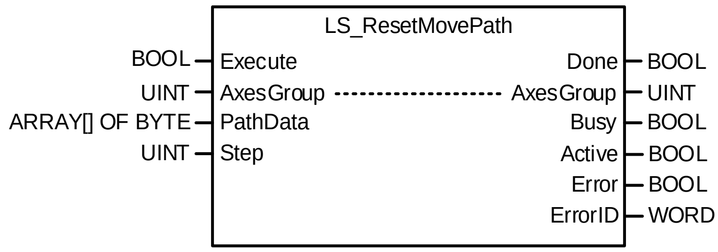
Input-Output		
UINT	AxesGroup	Set the group to set the path operation data (1 ~ 16: Group 1 ~ Group 16)
BYTE[]	PathData	Set the location where the path data is stored.
Input		
BOOL	Execute	In the rising Edge, it sends the command for setting the path operation data to the corresponding axis group.
UINT	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	Select 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 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 path. [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)
UINIT	TransitionMode	Unused
UREAL	TransitionParameter	Unused
Output		
BOOL	Done	Indicate that the path data setting 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	Indicate 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) For more details, refer to Section 8.4.11, "Path Operation of the Coordinate System ".

6.7.12 Delete path operation data (MC_RestMovePath)

Motion Function Block

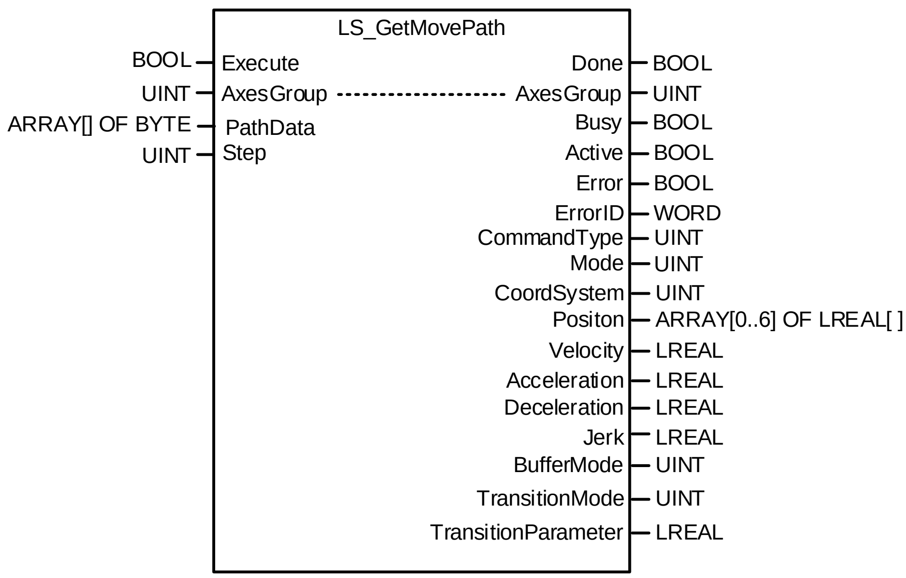


Input-Output		
UINT	AxesGroup	Set the group to set the path operation data (1 ~ 16: Group 1 ~ Group 16)
Input		
BOOL	Execute	In the rising Edge, the command for deleting the path operation data is sent to the corresponding axis group.
BYTE[]	PathData	Set the location where the path data is stored.
UINT	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	Indicate 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 8.4.11, "Path Operation of the Coordinate System ".

6.7.13 Read path operation data (MC_GetMovePath)

Motion Function Block



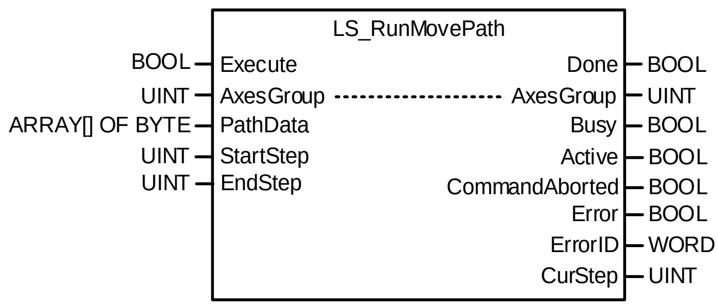
Input-Output		
UINT	AxesGroup	Set the group to set the path operation data(1 ~ 16: Group 1 ~ Group 16).
Input		
BOOL	Execute	In the rising Edge, the command for setting the path operation data is sent to the corresponding axis group.
BYTE[]	PathData	Set the location where the path data is stored.
UINT	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 that the path data setting 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	Indicate 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)
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 8.4.11, ''Path Operation of the Coordinate System ''.

6.7.14 Perform path operation (MC_RunMovePath)

Motion Function Block



Input-Output		
UINT	AxesGroup	Set the group to execute the path operation data. (1 ~ 16 : Group 1 ~ Group 16)
Input		
BOOL	Execute	In the rising Edge, the command for setting the path operation data is sent 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	Indicate 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.

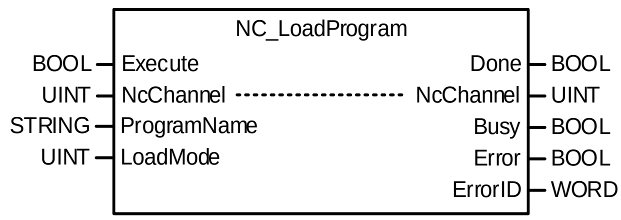
(4) The difference between StartStep and EndStep cannot be set to 100 or more. (Up to 100 step operations can be executed at one time.)

(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 8.4.11, "Path Operation of the Coordinate System ".

6.8 NC Control Function Block

6.8.1 Specify NC program (NC_LoadProgram)

Motion Function Block



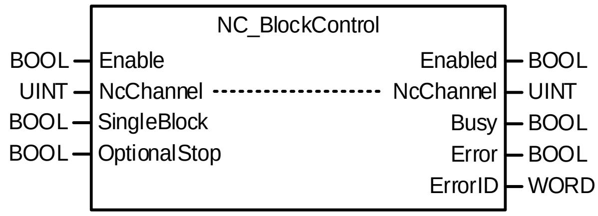
Input-Output		
UINT	NC channel	Set the NC channel to make the command.
Input		
BOOL	Execute	Set the program to be executed in the rising Edge.
STRING	ProgramName	Set the name of the program to be executed.
UINT	LoadMode	Unused (Only'0'is settable.)
Output		
BOOL	Done	Indicate the state of motion function block completion.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicate 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 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	NC channel	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	Done	Indicate the state of Block Control completion.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicate 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 determines the method to execute the program under the NC control.

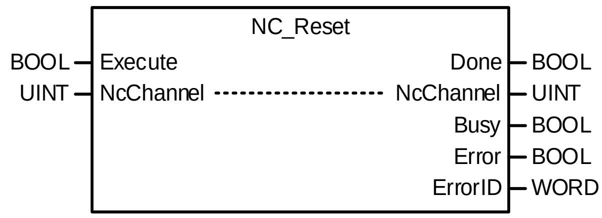
(2) If SingleBlock is set to '1', NC_CycleStart executes one block at a time and stops after execution. If SingleBlock becomes '1' during the automatic operation and NC_BlockControl function block is executed, it will be stopped after terminating the currently executing block.

(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 (NC_Reset)

Motion Function Block



Input-Output		
UINT	NC channel	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 the execution of motion function block is not completed.
BOOL	Error	Indicate whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

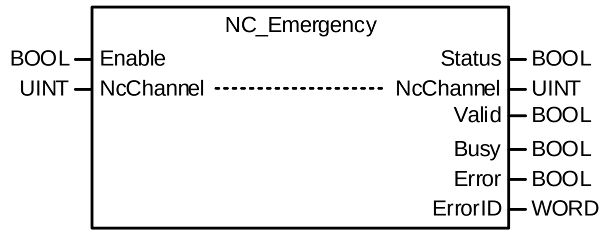
(1) This motion function block is to make the NC reset state under the NC control.
(2) If NC_Reset is executed during the automatic operation, it stops the automatic operation and changes into the reset state.
(3) The Reset state is as follows.

Contents		Status
Setting Data	Offset Value	Hold
	Parameter	Hold
Various Data	Program in Memory	Hold
	Contents in the buffer storage	MDI: Hold Other cancel
	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

Contents		Status
Action in operation	Movement	Cancel
	Dwell	Cancel
	Issuance of M, S, T code	Cancel
	Tool Length compensation	MDI: Hold Other cancel
	Cutter compensation	MDI: Hold Other cancel
	Storing called subprogram number	MDI: Hold Other cancel
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

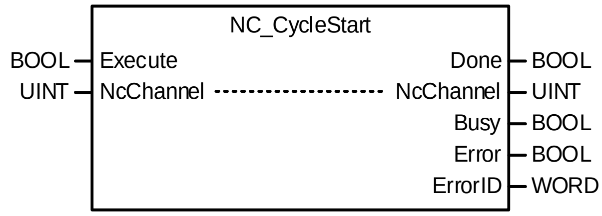


Input-Output		
UINT	NC channel	Set the NC channel to make the command.
Input		
BOOL	Enable	The emergency stop is executed while the input is '1'.
Output		
BOOL	Status	Indicate the status of the emergency stop.
BOOL	Valid	Indicate the validity of the function block output. (Same as the Status output).
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicate 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 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)

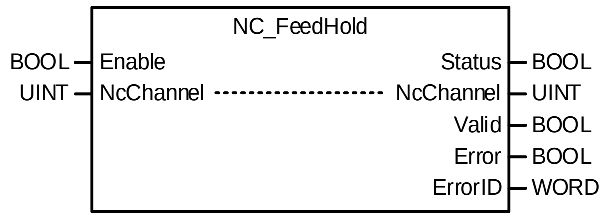
Motion Function Block		



Input-Output		
UINT	NC channel	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 the execution of motion function block is not completed.
BOOL	Error	Indicate 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 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.

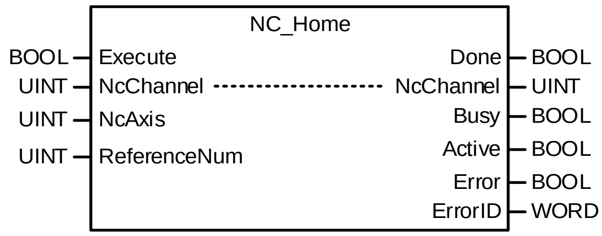
6.8.6 Feed hold (NC_FeedHold)

Motion Function Block		
		
Input-Output		
UINT	NC channel	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	Indicate 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 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 Homing (NC_Home)

Motion Function Block



Input-Output		
UINT	NC channel	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 the execution of motion function block is not completed.
BOOL	Active	Indicate that the current Function Block is controlling the axis.
BOOL	Error	Indicate whether an error occurs or not.
WORD	ErrorID	Output the number of error occurred while motion function block is running.

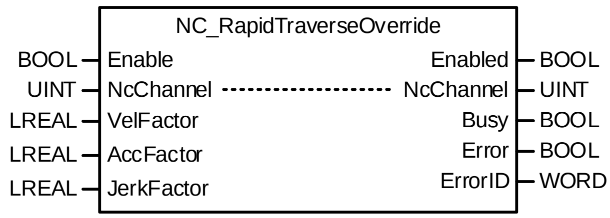
(1) This motion function block performs homing to the corresponding NC channel under the NC control.

(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

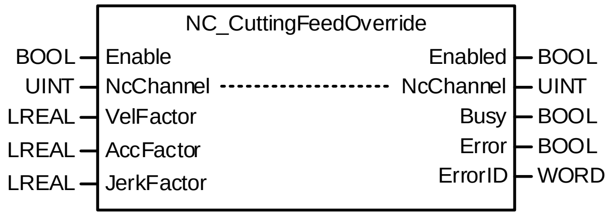


Input-Output		
UINT	NC channel	Set the NC channel to make the command.
Input		
BOOL	Enable	Execute the Rapid Traverse Override operation on the channel while the input is enabled.
LREAL	VelFactor	Specify the override rate of the speed. (0 ~ 1.0, 1.0=100%)
LREAL	AccFactor	Specify the override rate of 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 the execution of motion function block is not completed.
BOOL	Error	Indicate 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 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

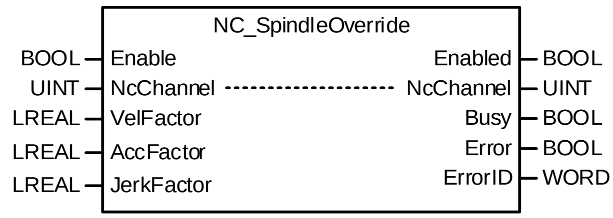


Input-Output		
UINT	NC channel	Set the NC channel to make the command.
Input		
BOOL	Enable	Execute the Cutting Feed Override operation on the channel while the input is enabled.
LREAL	VelFactor	Specify the override rate of the speed. (0 ~ 1.0, 1.0=100%)
LREAL	AccFactor	Specify the override rate of acceleration / 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 the execution of motion function block is not completed.
BOOL	Error	Indicate 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 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

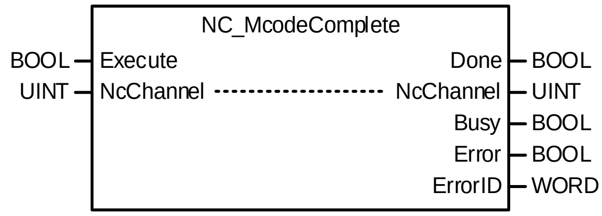


Input-Output		
UINT	NC channel	Set the NC channel to make the command.
Input		
BOOL	Enable	Execute the Spindle Override operation on the channel while the input is enabled.
LREAL	VelFactor	Specify the override rate of the speed. (0 ~ 1.0, 1.0=100%)
LREAL	AccFactor	Specify the override rate of acceleration / 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 the execution of motion function block is not completed.
BOOL	Error	Indicate 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 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	NC channel	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 the execution of motion function block is not completed.
BOOL	Error	Indicate 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 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

```
              NC_ScodeComplete
BOOL — Execute                        Done — BOOL
UINT — NcChannel ------------- NcChannel — UINT
                                      Busy — BOOL
                                     Error — BOOL
                                   ErrorID — WORD
```

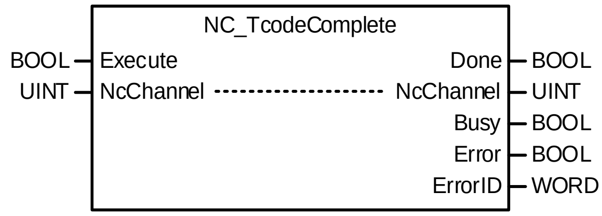
Input-Output		
UINT	NC channel	Set the NC channel to make the command.
Input		
BOOL	Execute	Set the completion of the S Code operation on the corresponding the channel.
Output		
BOOL	Done	Indicate the state of motion function block completion.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicate 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 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



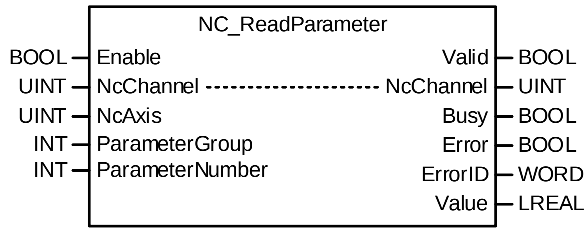
Input-Output		
UINT	NC channel	Set the NC channel to make the command.
Input		
BOOL	Execute	Set the completion of the T Code operation on the corresponding the channel.
Output		
BOOL	Done	Indicate the state of motion function block completion.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicate 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 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 parameters (NC_ReadParameter)

Motion Function Block



Input-Output		
UINT	NC channel	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	Valid	Indicate the validity of the function block output.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicate 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.

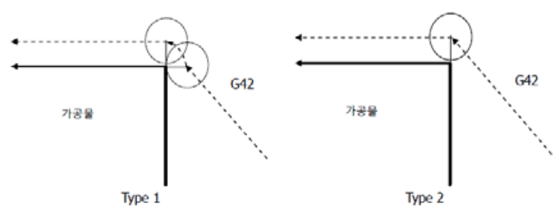
(1) This motion function block is to read and output the parameters of the channel and channel / axis of the corresponding channel.
(2) While the Enable input is active, the values of the relevant parameters are output continuously.
(3) ParameterGroup input specifies the parameter group number to read.
(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.

Parameters	Group	No.	Item	Description
1. Channel parameters	1. Basic setting	1	Target machining quantity	Set the target machining quantity. (0 ~ 2,147,483,647)
		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)
		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: Keep 1: Do not keep
		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: Time 1: Number of revolutions
		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

Parameters	Group	No.	Item	Description
1. Channel parameters	1. Basic setting	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.
		19	Set the inner/outer side of G22 No Travelling Area	0: Inner side 1: Outer side
		20	Whether to use the 3rd 'No Travelling Area'	0: 'No Travelling Area' is valid. 1: 'No Travelling Area' is invalid.
		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

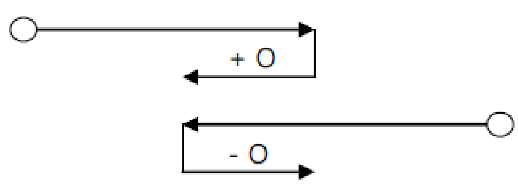
Parameters	Group	No.	Item	Description
1. Channel parameters	1. Basic setting	33	Monitoring time for in-position completion	0 ~ 65,535ms
	2. Circular milling setting	1	Regenerate the circular center when the circular alarm occurs	Set whether to recreate the central point of the arc without generating 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
		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)
		5	Circular radius with the speed-limiting function for the arc machining	(0 ~ 10,000 unit, real number)
		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 " . (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)
		9	Circular milling acceleration	Set the acceleration at the circular milling.
		10	Circular milling deceleration	Set the deceleration at the circular milling.
		11	Circular milling jerk	Set the jerk at the circular milling.
	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)
		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)
		4	Acceleration / deceleration method of the interpolation operation	1: Acceleration / deceleration before interpolation

Parameters	Group	No.	Item	Description
1. Channel parameters	3. Cutting feed setting	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
		10	Deceleration at the time of cutting feed (before interpolation)	Deceleration at the time of cutting feed
		11	Jerk at the time of cutting feed (before interpolation)	Jerk at the time of cutting feed
	8.Tool diameter 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

Parameters	Group	No.	Item	Description
1. Channel parameters	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. Workpiece coordinate system	1	Whether to use the workpiece coordinate system shift amount.	Set whether to use the workpiece coordinate system shift amount. 0: Unused 1: 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 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 values for the W axis.
		61	G56 workpiece coordinate system value 1	Sets the G56 workpiece coordinate system values 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 values for the W axis.
		71	G57 workpiece coordinate system value 1	Set the G57 workpiece coordinate system values for the X axis.
		……	……	Sets 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 values for the W axis.
		81	G58 workpiece coordinate system value 1	Set the G58 workpiece coordinate system values 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 values for the W axis.

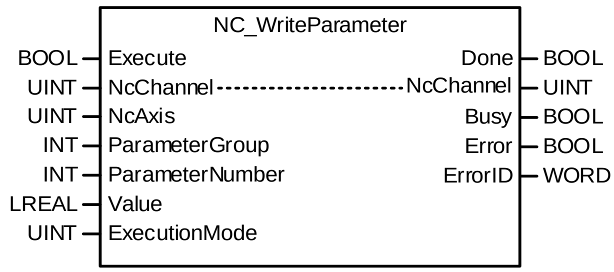
Parameters	Group	No.	Item	Description
1. Channel parameters	10. Workpiece coordinate system	91	G59 workpiece coordinate system value 1	Set the G59 workpiece coordinate system values for the X axis.
		……	……	Set the G59 workpiece coordinate system values for the 7 axes; Y, Z, A, B, C, U, V
		99	G59 workpiece coordinate system value 9	Set the G59 workpiece coordinate system values 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
		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: Do not display
		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)
		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)
		……	……	
		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)

Parameters	Group	No.	Item	Description
1. Channel parameters	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 Traverse(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)
		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)
	16. Relative coordinate 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.
2. Channel /Axis parameters	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

Parameters	Group	No.	Item	Description
2. Channel /Axis parameters	2. Origin	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 traverse	2	Rapid traverse acceleration	The set value is used as the acceleration of the G00 block.
		3	Rapid traverse deceleration	The set value is used as the deceleration of the G00 block.
		4	Rapid traverse jerk	The set value is used as the jerk of the G00 block.
		5	Rapid traverse speed	The set value is used as the traverse speed of the G00 block. (0~100000 unit/min, real number)
	4. Traverse area	1	Minimum value of the G22 Traverse-Prohibited Area range for the X, Y, and Z axis.	Set the minimum value of the G22 Traverse-Prohibited Area range for the X, Y, and Z axis. (-100,000~100,000 unit, real number)
		2	Maximum value of the G22 Traverse-Prohibited Area range for the X, Y, and Z axis.	Set the maximum value of the G22 Traverse-Prohibited Area range for the X, Y, and Z axis. (-100,000~100,000 unit, real number)
		3	Minimum value of the 3rd Traverse-Prohibited Area range for the X, Y, and Z axis.	Set the minimum value of the 3rd Traverse-Prohibited Area range for the X, Y, and Z axis. (-100,000~100,000 unit, real number)
		4	Maximum value of the 3rd Traverse-Prohibited Area range for the X, Y, and Z axis.	Set the maximum value of the 3rd Traverse-Prohibited Area range for the X, Y, and Z axis. (-100,000~100,000 unit, real number)
	5. Sub setting	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)

6.8.15 Write NC parameters (NC_WriteParameter)

Motion Function Block



Input-Output		
UINT	NC channel	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 the parameter to be written.
UNIT	ExecutionMode	Reserved
Output		
BOOL	Valid	Indicate the validity of the function block output.
BOOL	Busy	Indicate that the execution of motion function block is not completed.
BOOL	Error	Indicate 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 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 6.7.14 Reading NC parameters.

Chapter 7 Program

7.1 Program Configuration

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 Program Configuration

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.

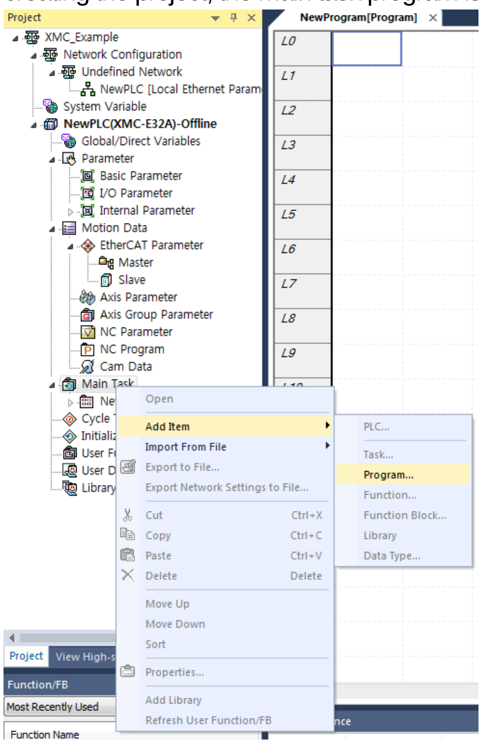
Item	Description
Initialization task program	• The initialization task program is the first task program executed after the motion controller completes its own initialization required for operation when power is applied. It runs until the INIT_DONE command is executed. • When the initialization program 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

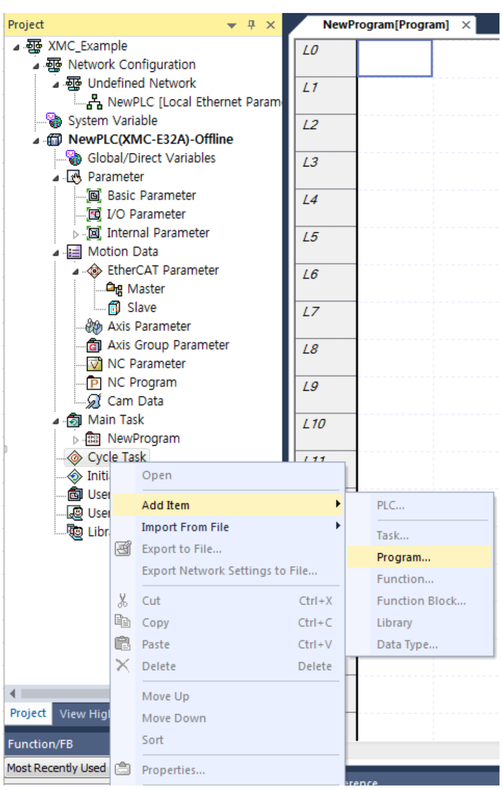
1) How to set the main task program

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.)



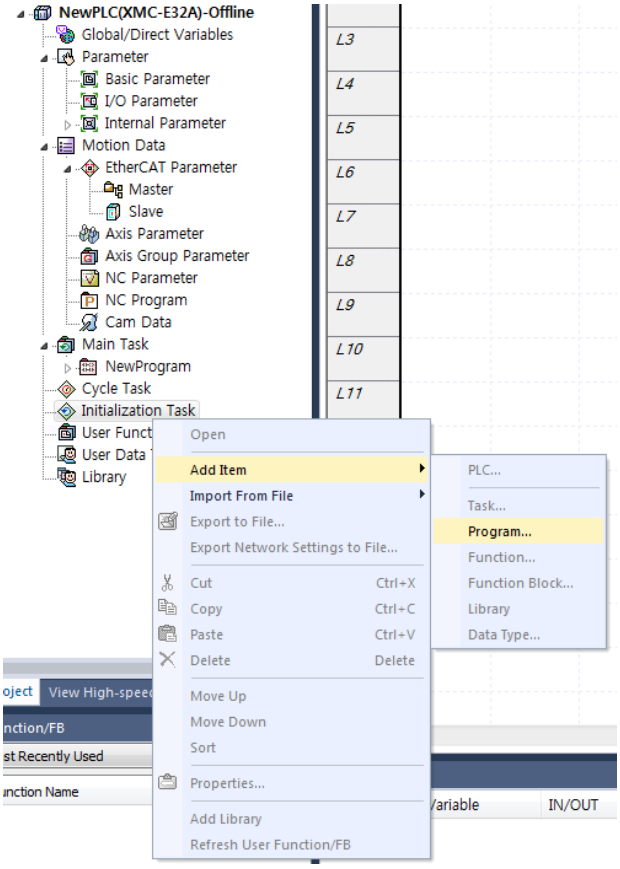
2) How to set the cycle task program

In the cycle task location, click the right mouse button and click 『Add item』-『Program』.

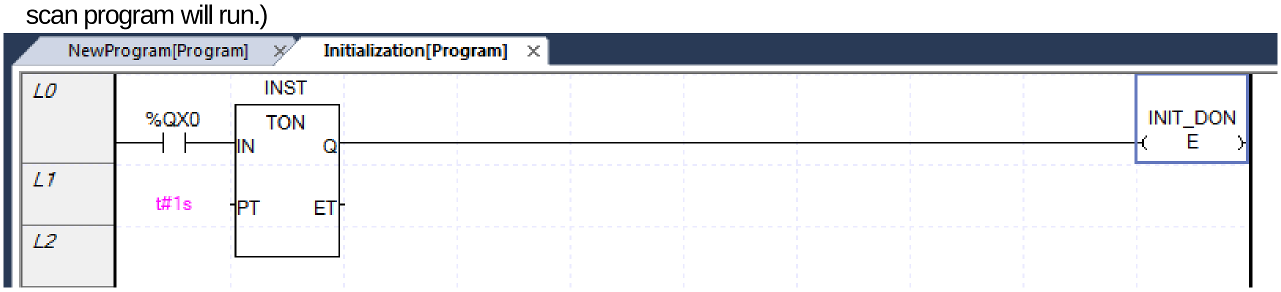


3) How to set the initialization task program

(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.)

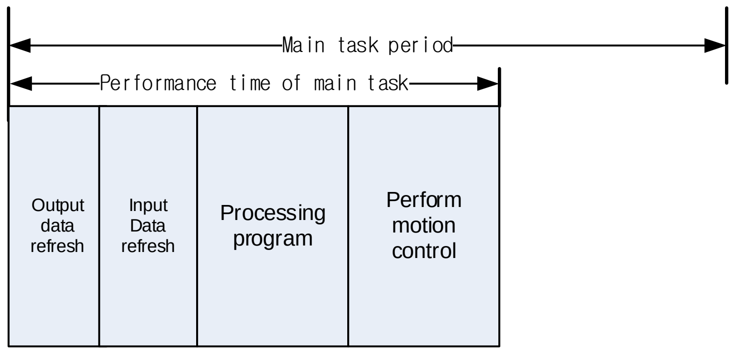


7.1.3 Run Time of the Program

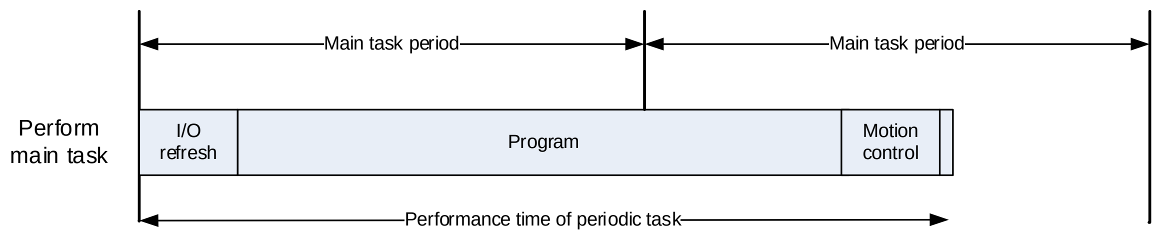
The execution time (scan time) of each task program is calculated as follows. It means the time required from the start of each control cycle to the time when the motion control execution is done, that is, the time required to complete the task.

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.

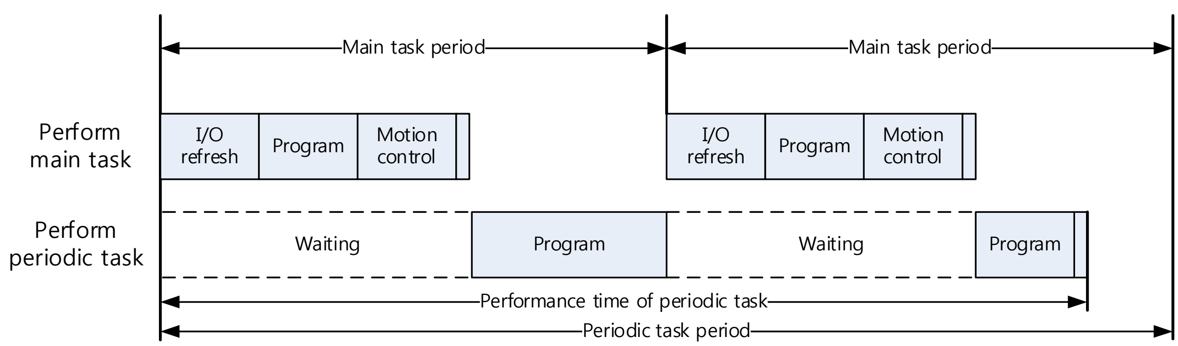


If the main task is not completed for one cycle, just measure the time from the start of the main task cycle to the time when the main task is completed as shown below.



2) Run time of the periodic task

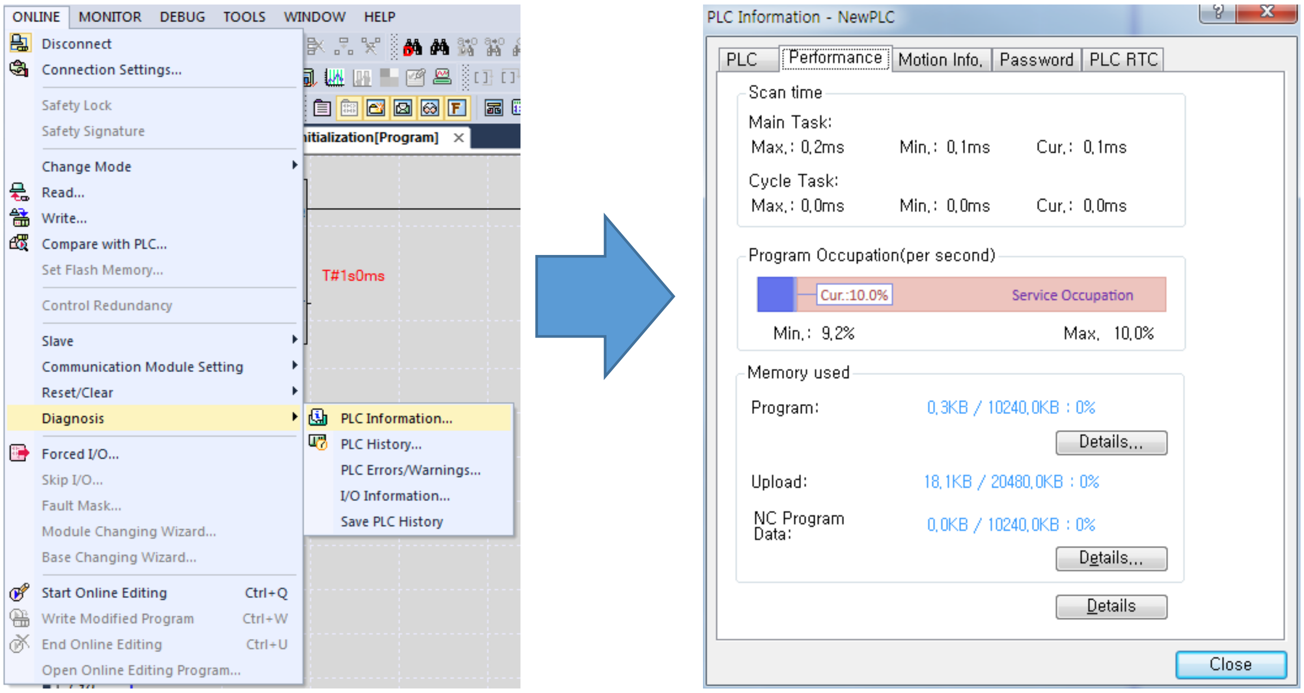
It means the time from the start of the periodic task cycle to the time when the periodic task program is done. The periodic task can be executed in several main task cycles because it runs after the main task is completed.



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	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)

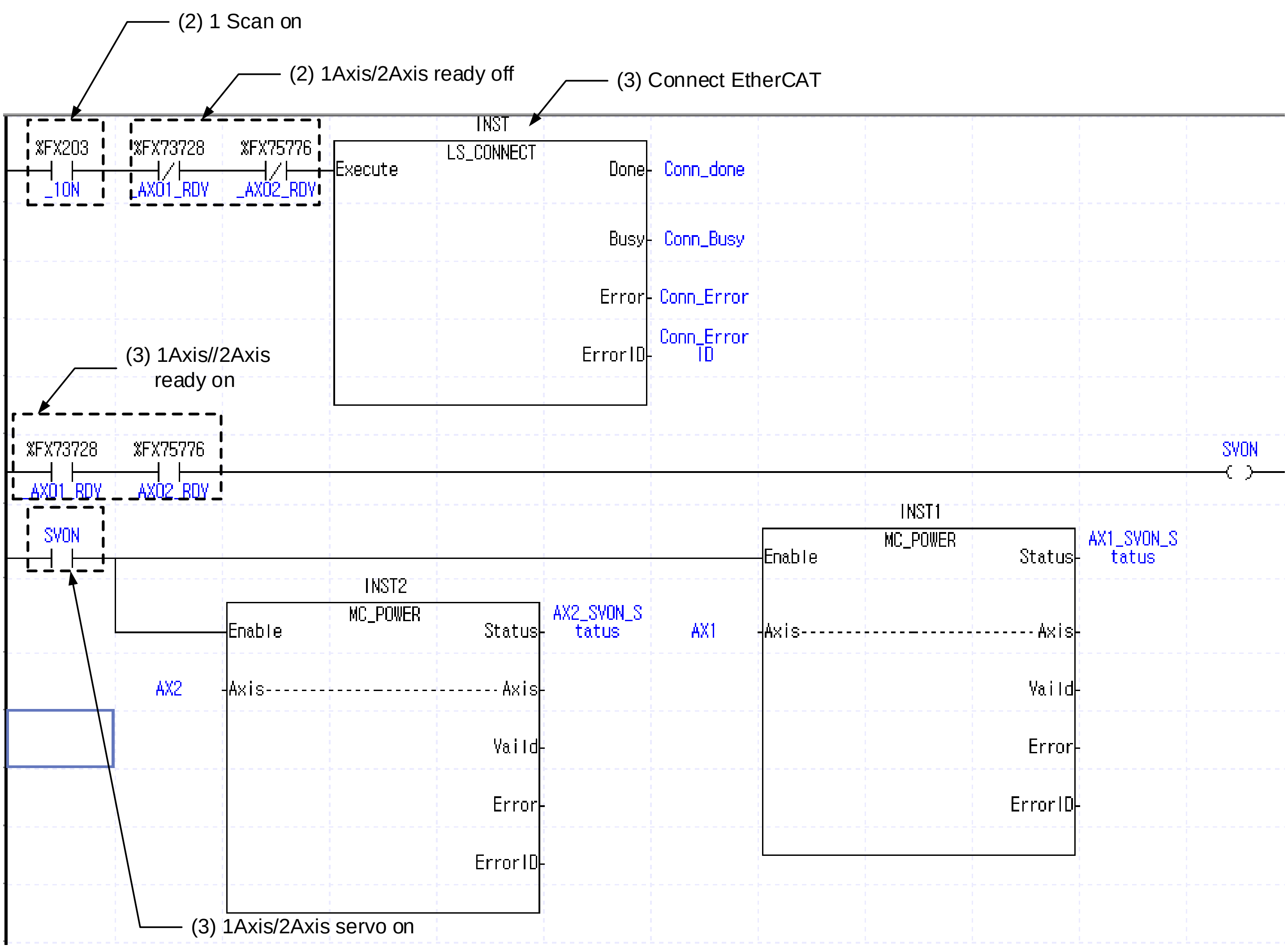
7.2 Status Information Reading

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, and can be delivered to PLC CPU by being assigned to a shared device of the motion control module.
For more information on the types and functions of flags, refer to “Appendix 1. Flag list”.

7.3 Discrete Motion Program

7.3.1 Preparation for Operation

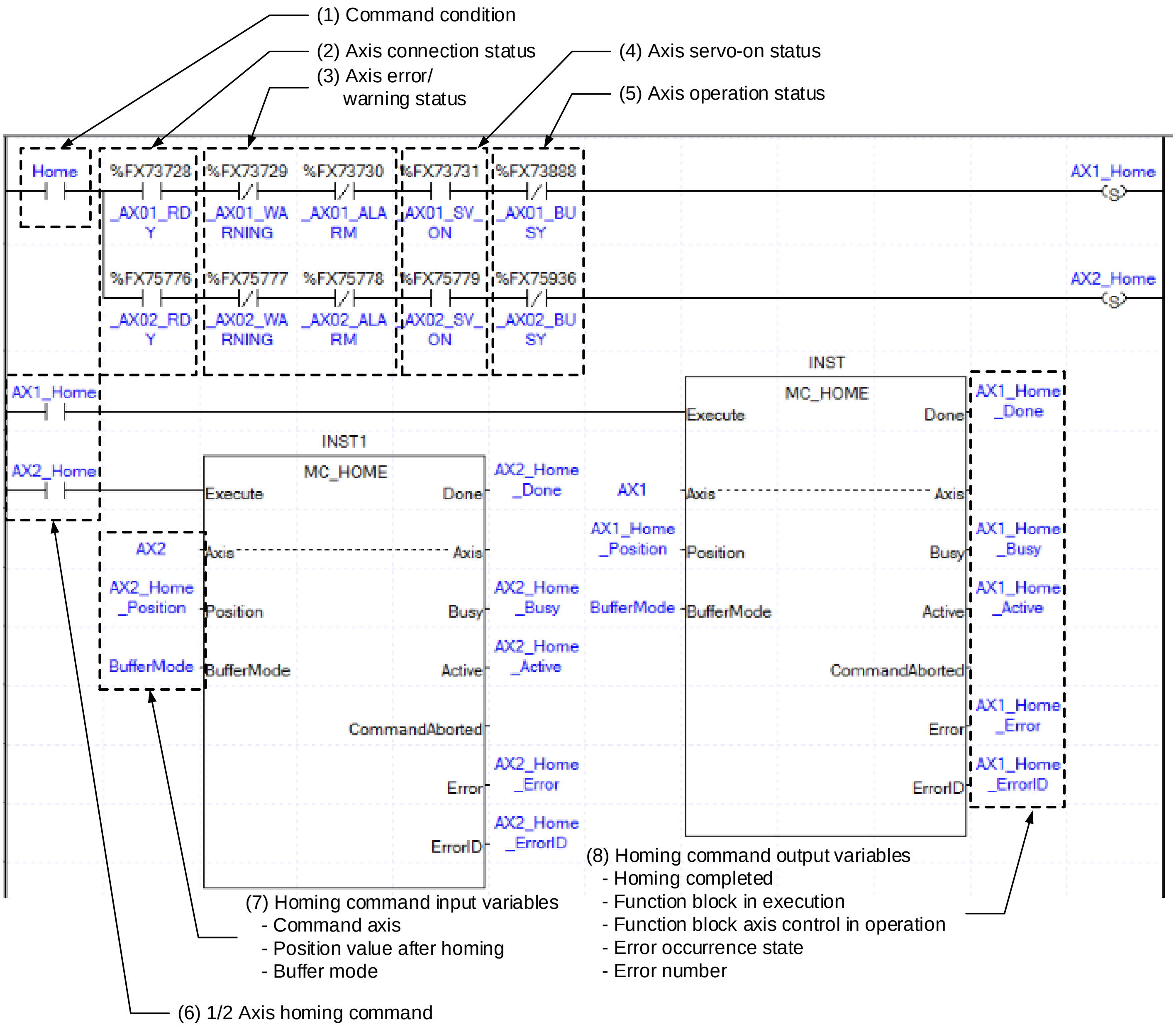
These are example programs that make access to servo drive connected with Ethernet cable and get the connected servo drive to be On to operate EtherCAT servo drive.



(1) The above examples assume situation in which two axes of 1-axis and 2-axis are connected to the motion control module.
(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 Homing 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 state flag

: In case 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

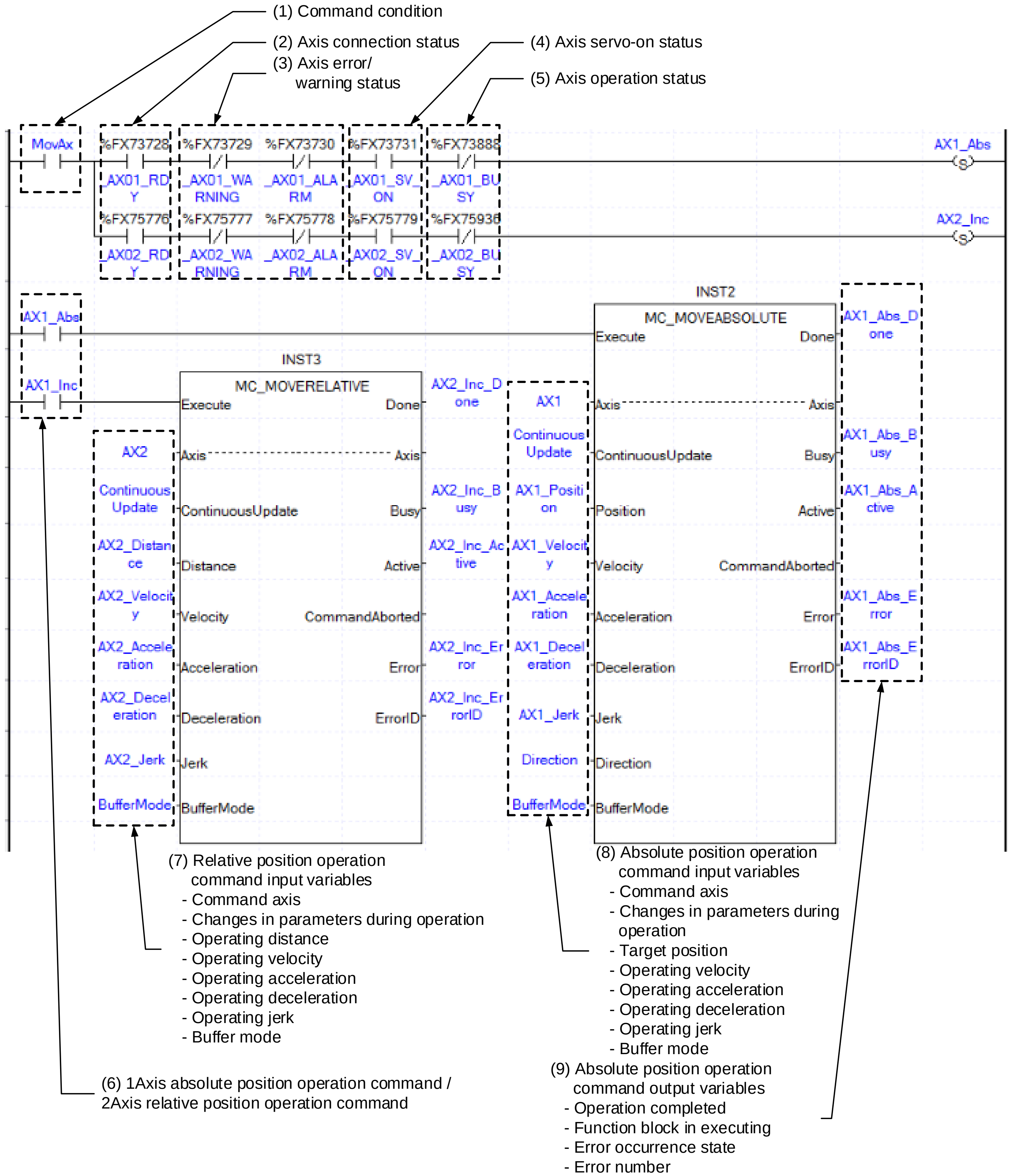
Conditions to perform motion function block may vary depending on systems.

(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 performed.
- 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 controls 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 Position/Relative Position 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 state flag
: If 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 perform 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 performed.
- 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 relative coordinate 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 performed.
- 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, “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”.

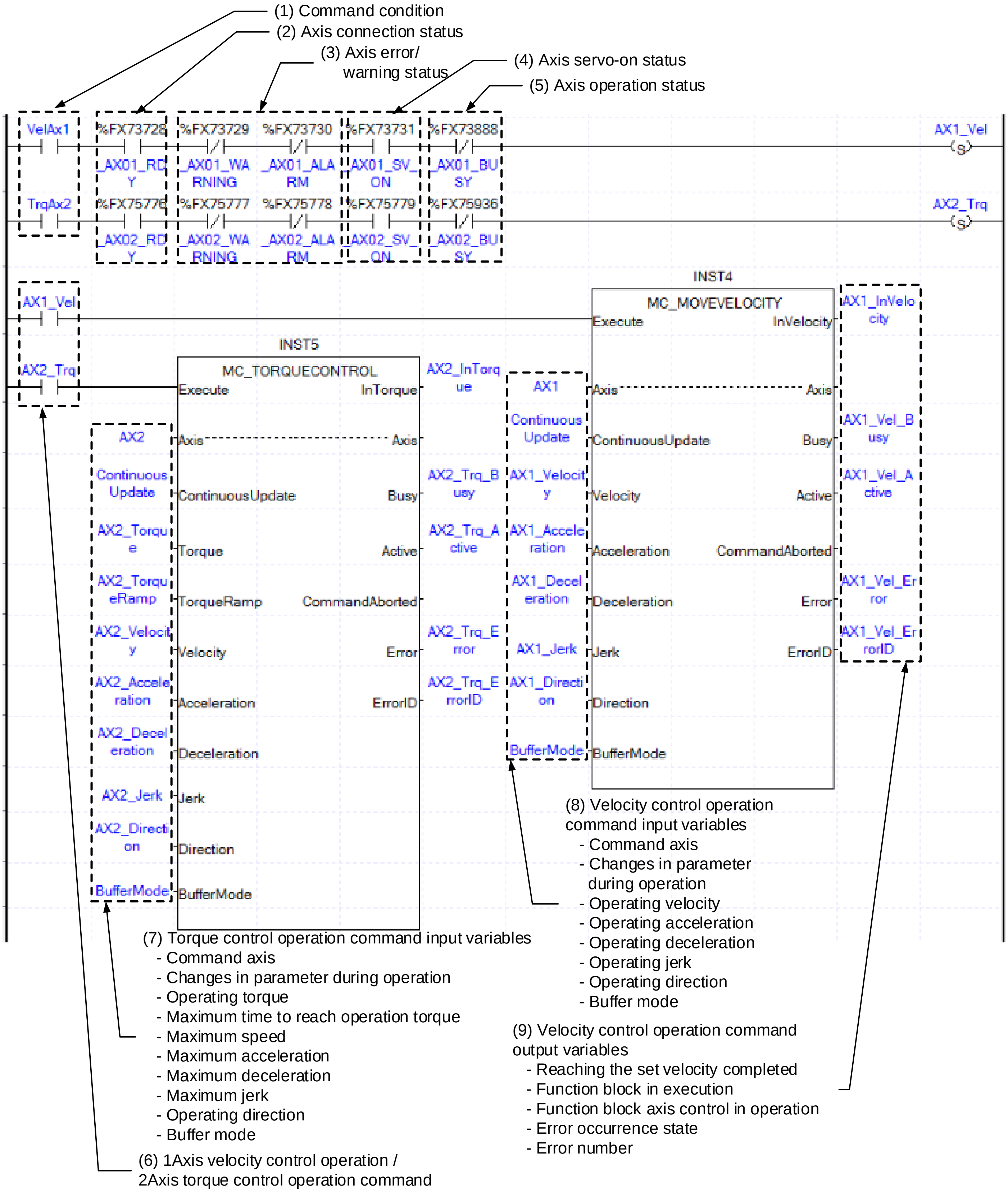
(9) Absolute position operation command output variable

: 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 if operation completed 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 when motion function block is 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, 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 state flag
: In case 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 perform 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 details, refer to "6.1.5 Changes in parameters during the 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, 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 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 details, 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 details, refer to "6.1.4 Buffer Mode input".

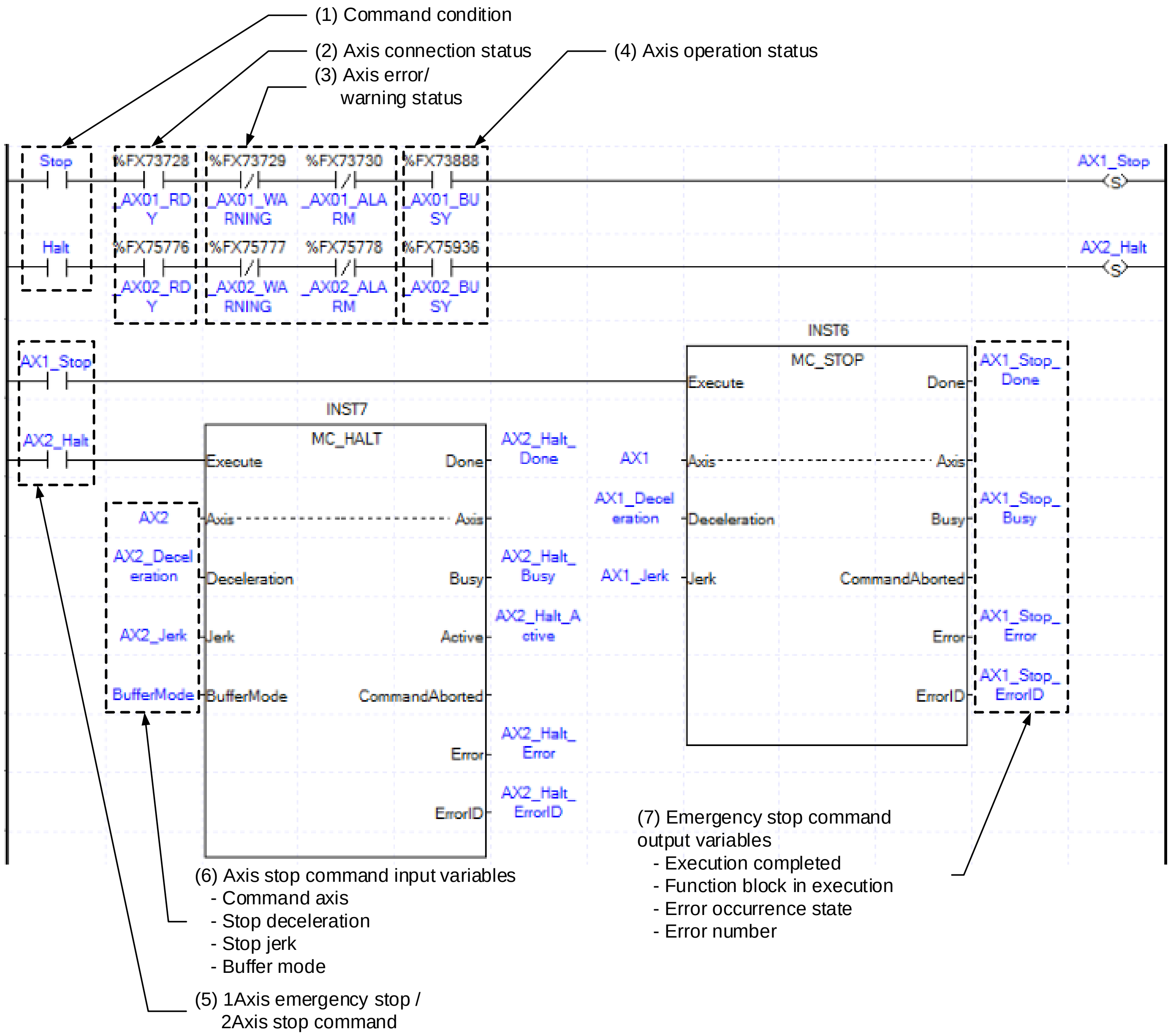
(9) Speed control operation command output variable

: 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: If motion function block is being performed, it is On, and operation is completed, it becomes Off.
- Function Block axis control in operation: In case motion function block controls 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 implement emergency stop of the axis, "Immediate Stop (MC_Stop)" performs "Immediate Stop (MC_Stop)", and other motion function blocks cannot be executed during the stop. 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".



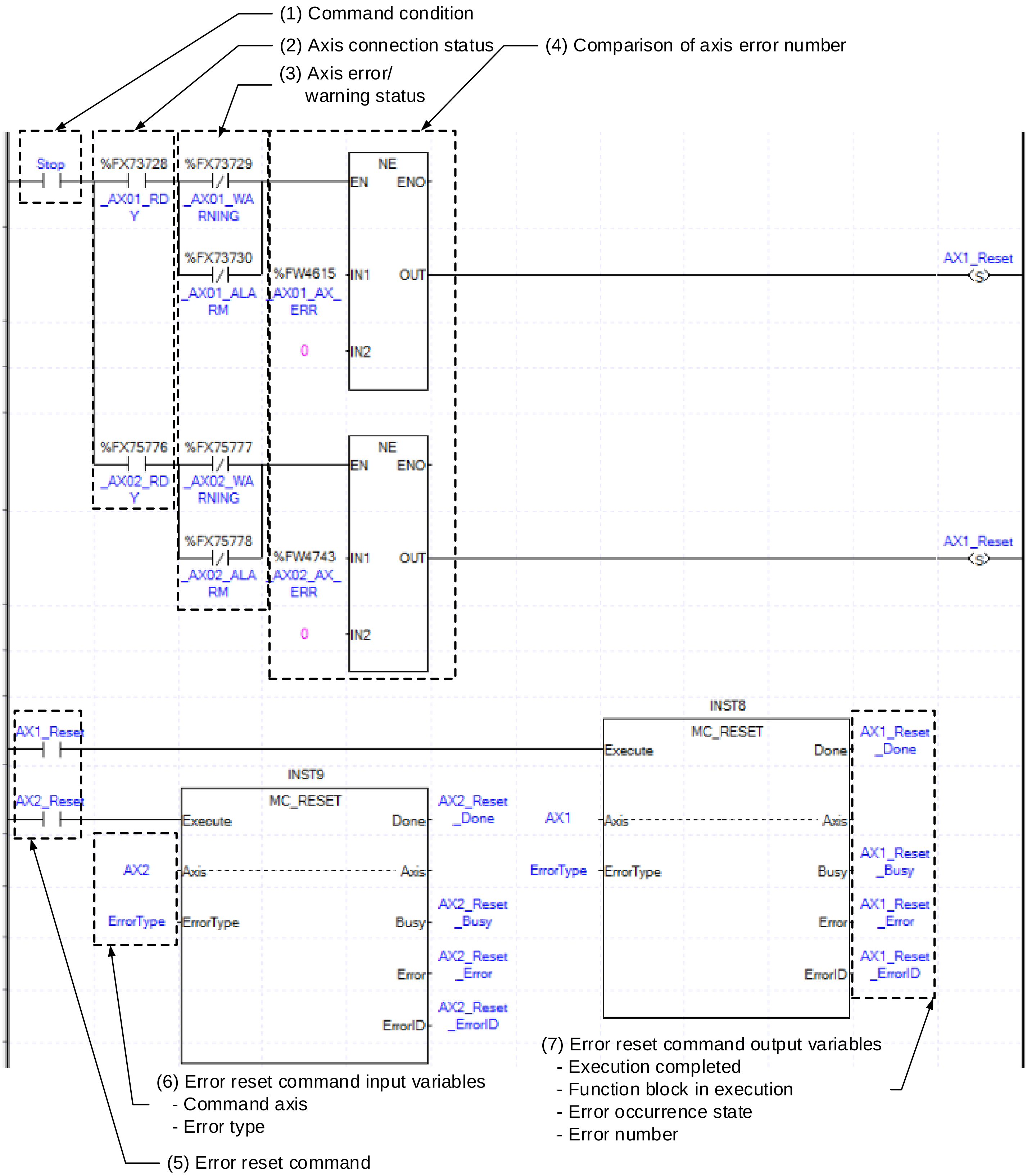
(1) Command condition
: It is a condition to give emergency stop/axis stop commands to the axis.
(2) Axis connection state flag
: In case the axis to be operated is connected to motion control module, and EtherCAT communication with motion control module is normally performed, it is On.
(3) Axis error/warning 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 perform 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 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: 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 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 Processing

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
: In case 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.

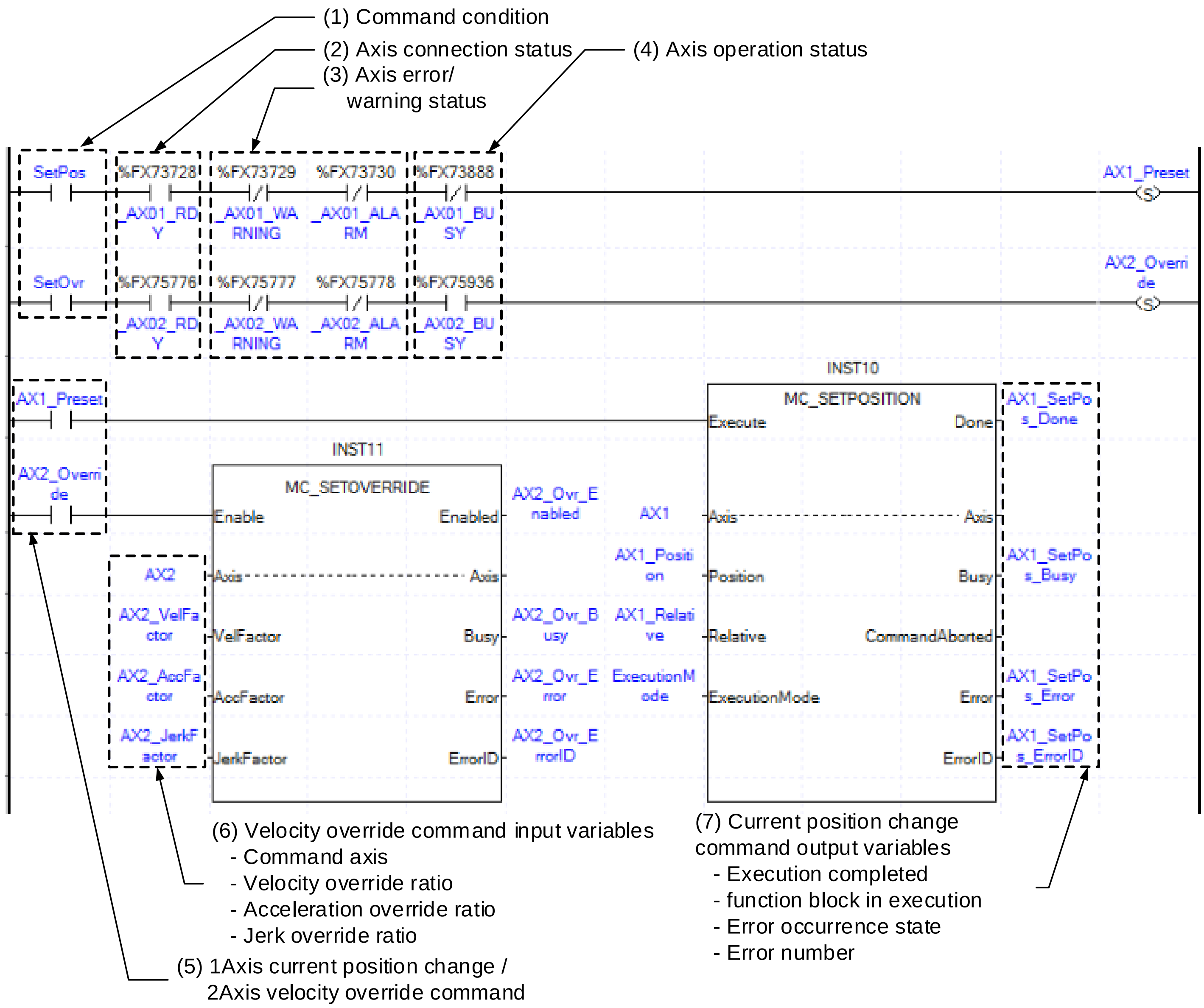
Conditions to perform motion function block may vary depending on systems.

(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: 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 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 operation motion function block” of “6.1.3 Basic I/O Variable”.

7.3.7 Change in Operation

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 state flag

: In case the axis is 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 set 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.

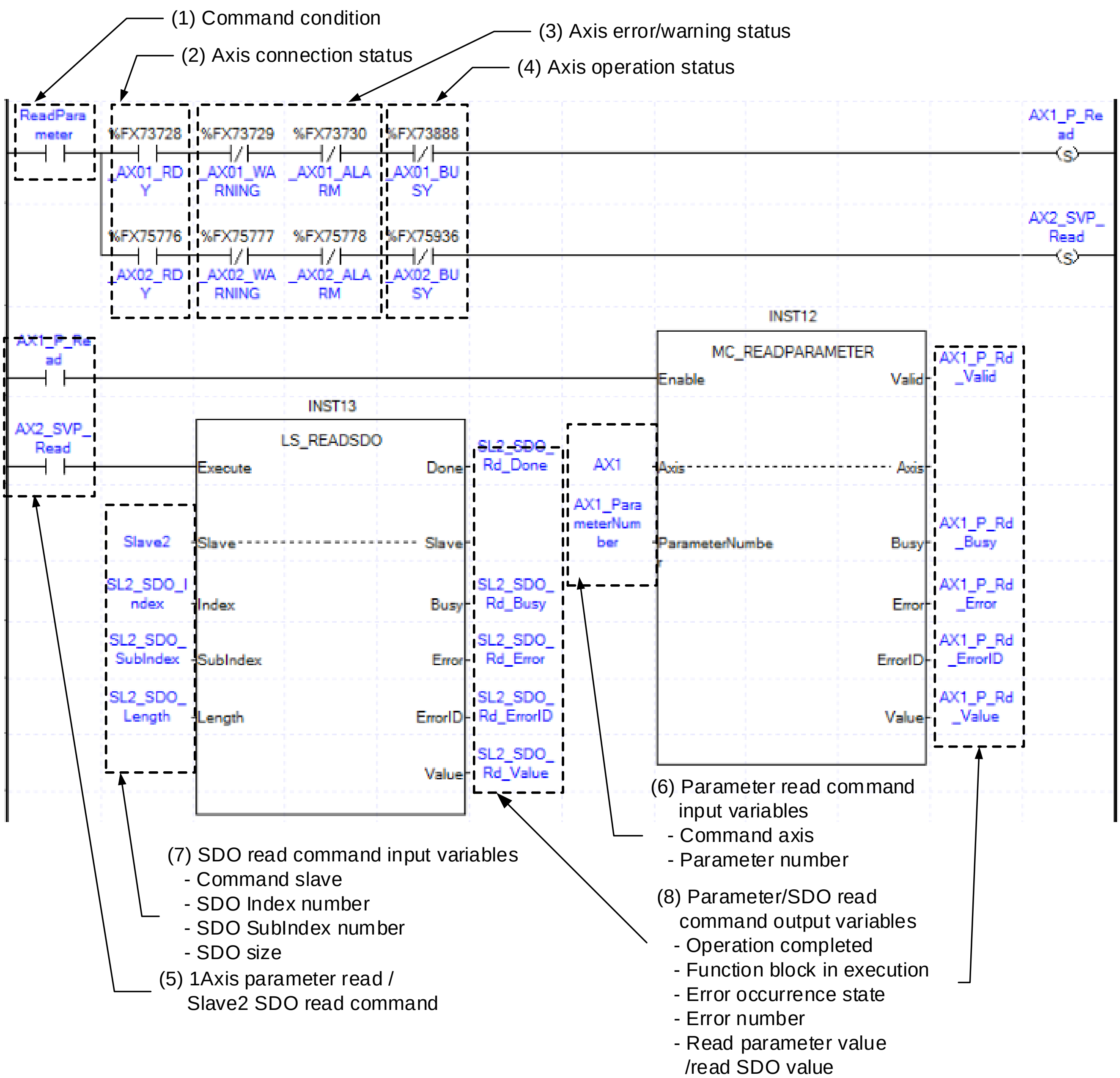
- Execution completed: If the execution of motion function block is completed, it is On.
- Function Block in execution: When motion function block is executed, it is On, and the execution is completed, It becomes Off.
- Error occurrence state: In case error occurs while the motion function block is being executed, it is O.
- 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 operation motion function block” of “6.1.3 Basic input and output variables.

7.3.8 Parameter Write/Read

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 state flag
: In case 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.

Number	Parameter	Item	Settings
0	Basic Parameter	Unit	0:pulse, 1:mm, 2:inch, 3:degree
1		Pulses per rotation	1~4,294,967,295 [pulse]
2		Travel per rotation	0.000000001 ~ 4,294,967,295 [Unit]
3		Speed command unit	0:Unit/Time, 1:rpm
4		Speed limit	LREAL (positive) [Unit/s, rpm]
5		Emergency stop deceleration	0, LREAL (positive) [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:Disable, 1:Enable
10	Extended Parameter	S/W upper limit	LREAL [Unit]
11		S/W lower limit	LREAL [Unit]
12		Infinite running repeat position	LREAL (positive) [Unit]
13		Infinite running repeat	0:Disable, 1:Enable
14		Command inposition range	0, LREAL (positive) [Unit]
15		Tracking error over-range value	0, LREAL (positive) [Unit]
16		Current position compensation amount	0, LREAL (positive)
17		Current speed filter time constant	0 ~ 100
18		Error reset monitoring time	0 ~ 1,000 [ms]
19		S/W limit during speed control	0:Don’t detect, 1:Detect
20		Tracking error level	0:Warning, 1:Alarm

Number	Parameter	Item	Settings
21	Extended Parameter	JOG high speed	LREAL (positive) [Unit/s] (JOG low speed ~ Speed limit)
22		JOG low speed	LREAL (positive) [Unit/s] (< JOG high speed)
23		JOG Acceleration	LREAL (positive) [Unit/s^2]
24		JOG Deceleration	LREAL (positive) [Unit/s^2]
25		JOG Jerk	LREAL (positive) [Unit/s^3]
26		Override Mode	0: Percent ,1: Set value
27	NC Parameter	Identifying range to reach the spindle rotation command speed	0~100%
28		Identifying RPM to reach the spindle rotation zero speed	0~100rpm
100	Encoder Parameter	Encoder1 Unit	0: pulse, 1: mm, 2: inch, 3:degree
101		Encoder1 Pulses per rotation	1~4,294,967,295 [pulse]
102		Encoder1 Travel per rotation	0.000000001 ~ 1~4,294,967,295[Unit]
103		Encoder1 Pulse input	0: CW/CCW (x1), 1: Pulse/Dir (x1) 2: Phase/DIR (x2), 3: Phase A/B (x1) 4: Phase A/B (x2), 5: Phase A/B (x4)
104		Encoder1 Max. value	(Encoder1 Min. value+1) ~ 2,147,483,647
105		Encoder1 Min. value	-2,147,483,648 ~ (Encoder1 Max. value-1)
106		Encoder1 speed unit	0: Unit/sec, 1: Unit/min, 2: rpm
107		Encoder1 input filter value	0: No use, 1: 500kPPS 2: 200kPPS, 3. 100kPPS 4: 10kPPS, 5: 1kPPS 6: 0.2kPPS
108		Encoder1 position filter time constant	0 ~ 1,000 ms
200		Encoder2 Unit	0: pulse, 1: mm, 2: inch, 3:degree
201		Encoder2 Pulses per rotation	1 ~ 4,294,967,295 [pulse]
202		Encoder2 Travel per rotation	0.000000001 ~ 4,294,967,295 [Unit]
203		Encoder2 Pulse input	0: CW/CCW (x1), 1: Pulse/Dir (x1) 2: Phase/DIR (x2), 3: Phase A/B (x1) 4: Phase A/B (x2), 5: Phase A/B (x4)
204		Encoder2 Max. value	(Encoder2 Min. value+1) ~ 2,147,483,647
205		Encoder2 Min. value	-2,147,483,648 ~ (Encoder2 Max. value-1)
206		Encoder2 speed unit	0: Unit/sec, 1: Unit/min, 2: rpm
207		Encoder2 input filter value	0: No use, 1: 500kPPS 2: 200kPPS, 3. 100kPPS 4: 10kPPS, 5: 1kPPS 6: 0.2kPPS
208		Encoder2 position filter time constant	0 ~ 1,000 ms

(7) Servo parameter read command read input variables

: These are input variables to execute Servo Parameter Read (LS_ReadSDO) motion function block.

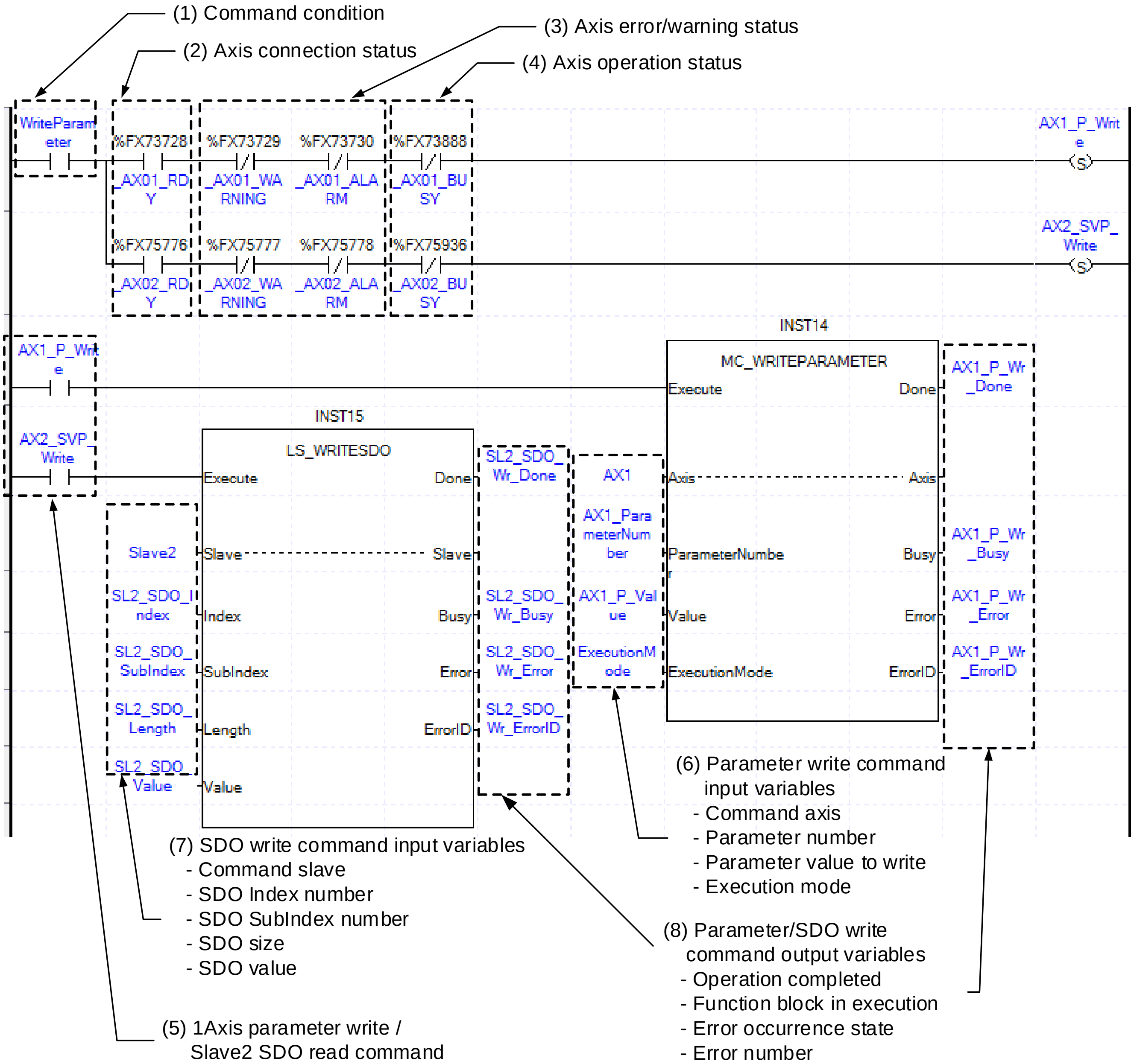
- Command axis: It sets the axis in which motion function block is executed.
- Servo parameter index number, SubIndex number, size: Each value is set in servo parameters to read. Refer to the instruction manual of the servo drive for index number, subindex number and size of servo parameters.

(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 the 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 state flag
: In case 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) Parameter 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)

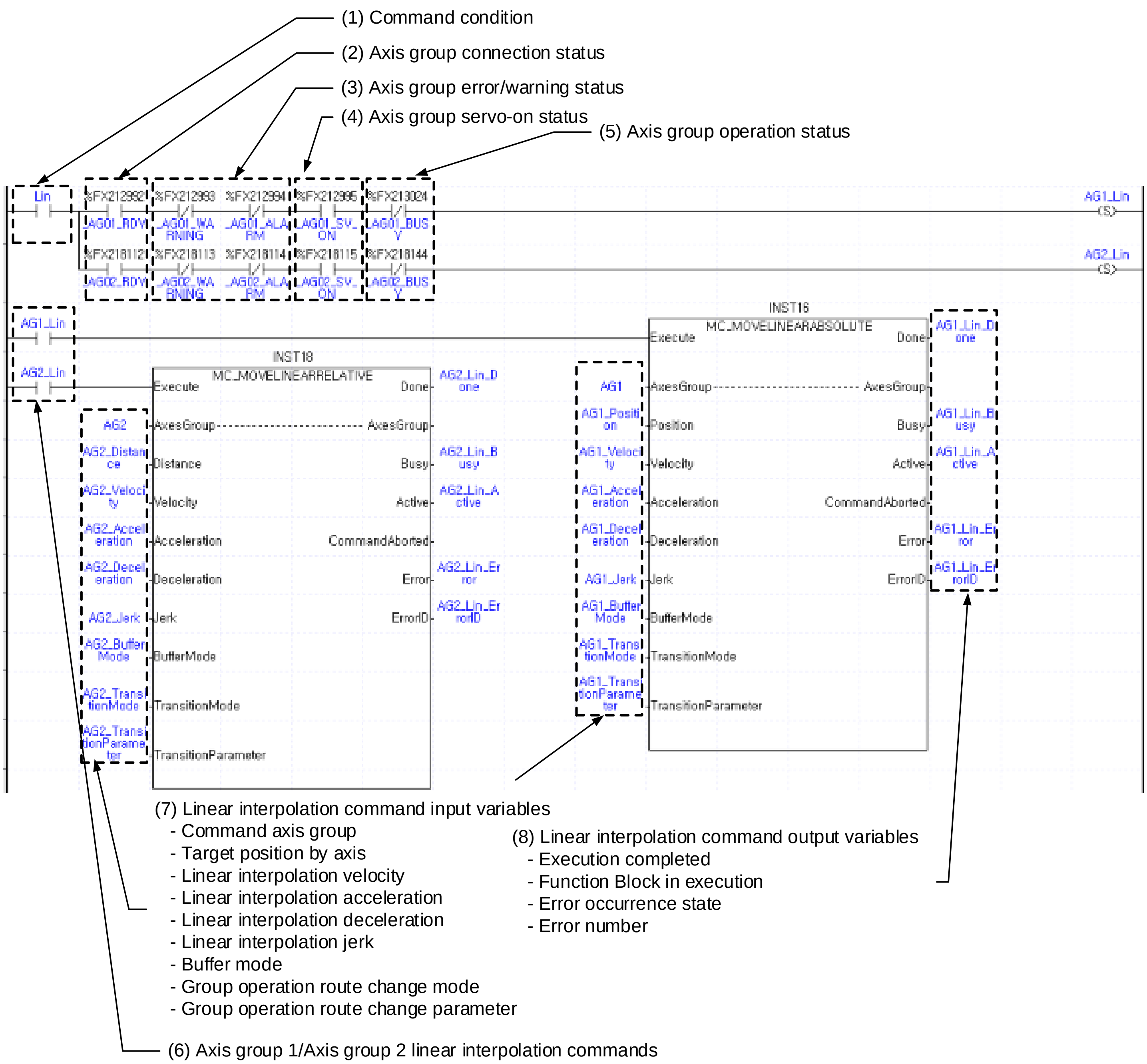
(7) SDO write command input variables
: These are input variables to execute SDO write (LS_WriteSDO) motion function block.
- Command axis: It sets the axis in which motion function block is executed.
- Servo parameter index number, subIndex number, size
  : Each value is set according to servo parameters 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 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 as the motion function block is 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 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 “7.4.5 Axis group processing” to include an axis in axis group or remove axis from axis group.

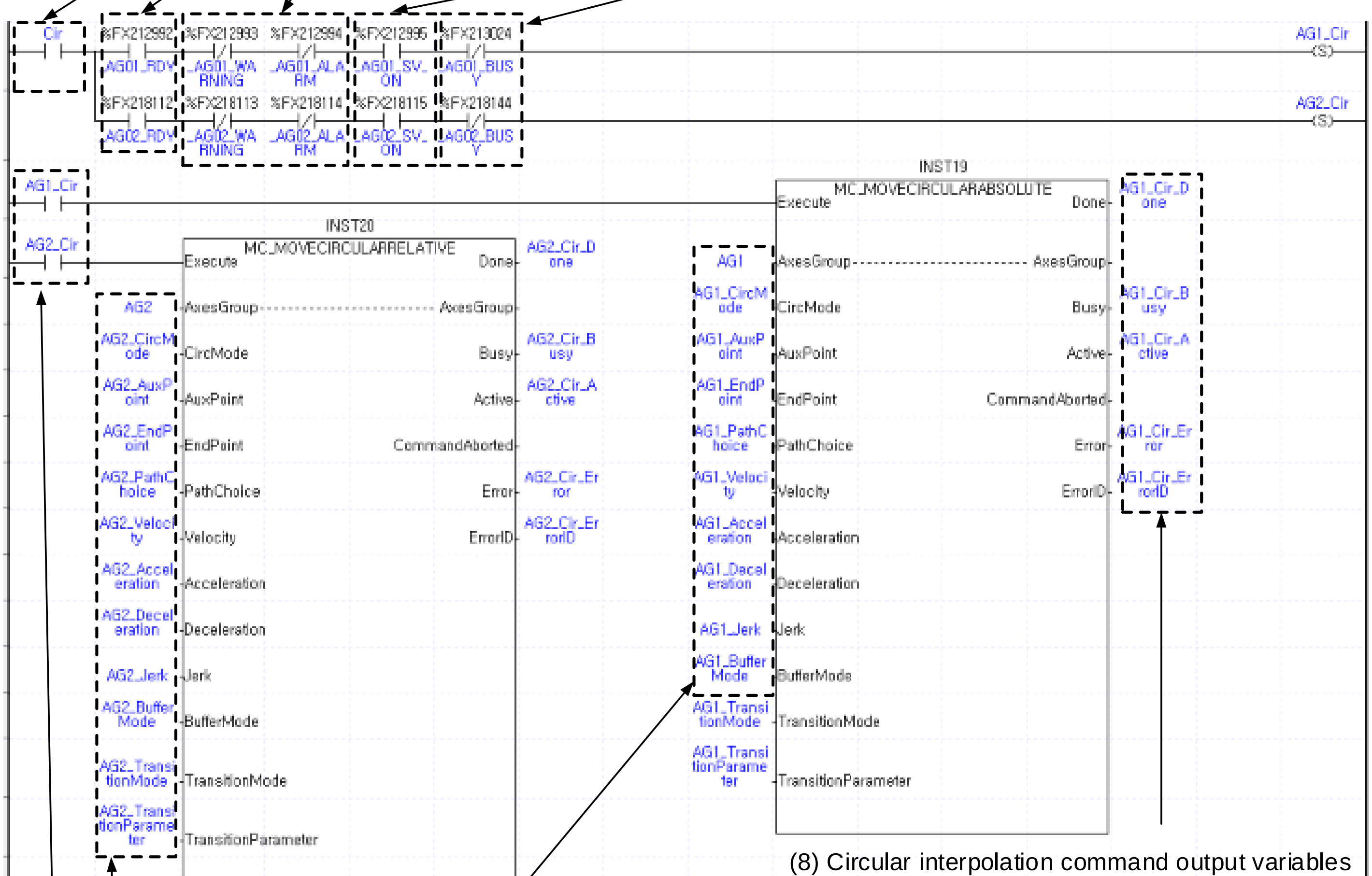
(1) Command condition

(2) Axis group connection status

(3) Axis group error/warning status

(4) Axis group servo-on status

(5) Axis group operation status



(8) Circular interpolation command output variables
- Execution completed
- Function block in execution
- The axis group being controlled by function block
- Error occurrence state
- Error number

(7) Circular interpolation command input variables
- Command axis group
- Circular interpolation method
  (mid point, center point, radius)
- Axis-specific circular interpolation auxiliary point(Array)
- Axis-specific circular interpolation target point(Array)
- Circular interpolation path(CW, CCW)
- Circular interpolation velocity
- Circular interpolation acceleration
- Circular interpolation deceleration
- Circular interpolation jerk
- Buffer mode

(6) Axis group 1/Axis group2 circular interpolation commands

(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 is 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 set whether to execute immediately or execute after the completion of commands which are currently being performed. For details, refer to “6.1.4 Buffer Mode Input”.

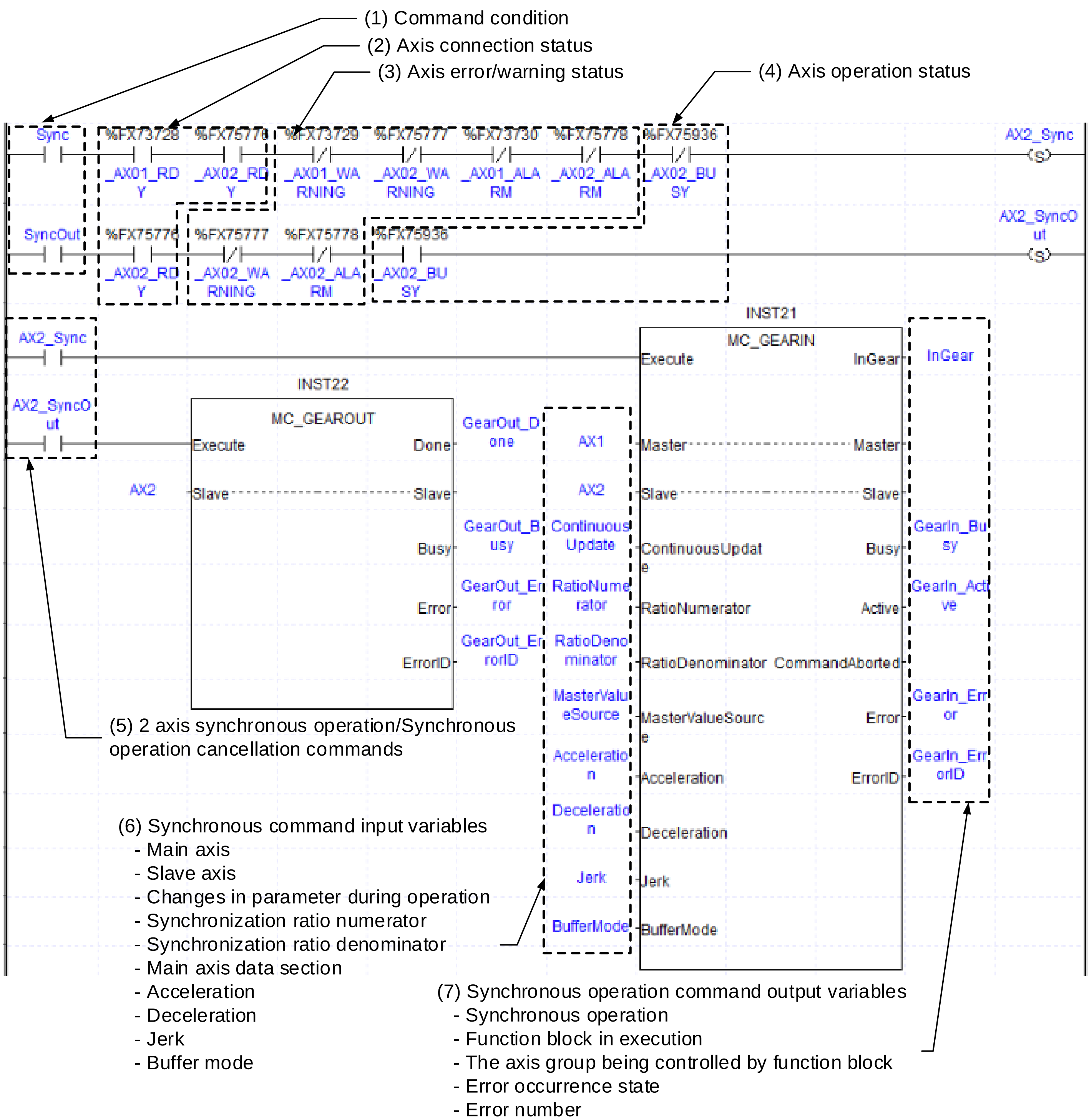
(8) Circular interpolation command output variable

: 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 as the motion function block is 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 state flag
: When 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 state 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) 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.
- Changes in parameters during operation: It sets whether to apply to the operation by changing input variable values of the function block. For details, refer to “6.1.5 Changes in parameters during execution of motion function block”.
- 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 details, refer to “6.1.4 Buffer Mode Input”.

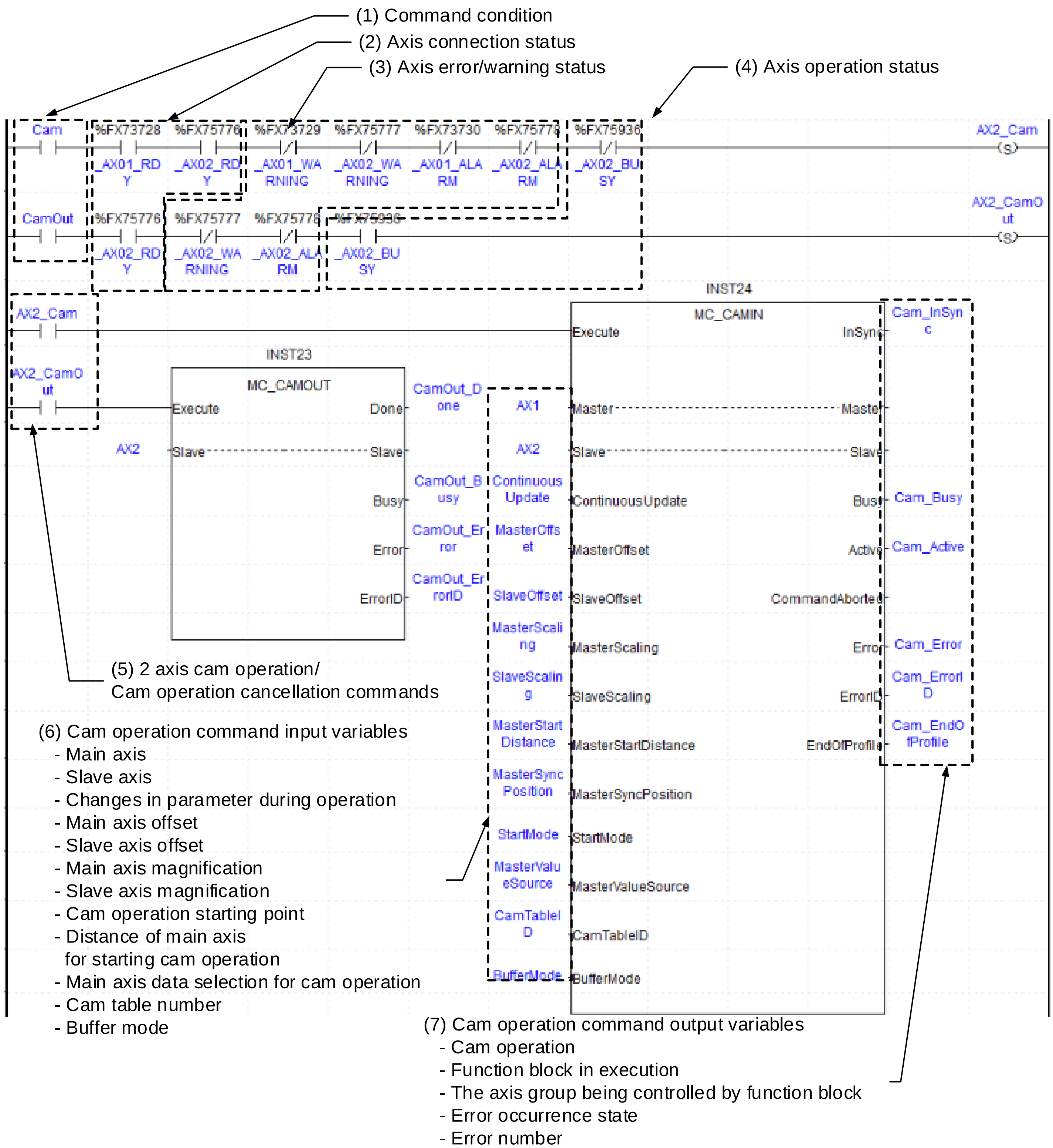
(7) Synchronous operation command output variable

: 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.
- 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 as the motion function block is 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 state 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.

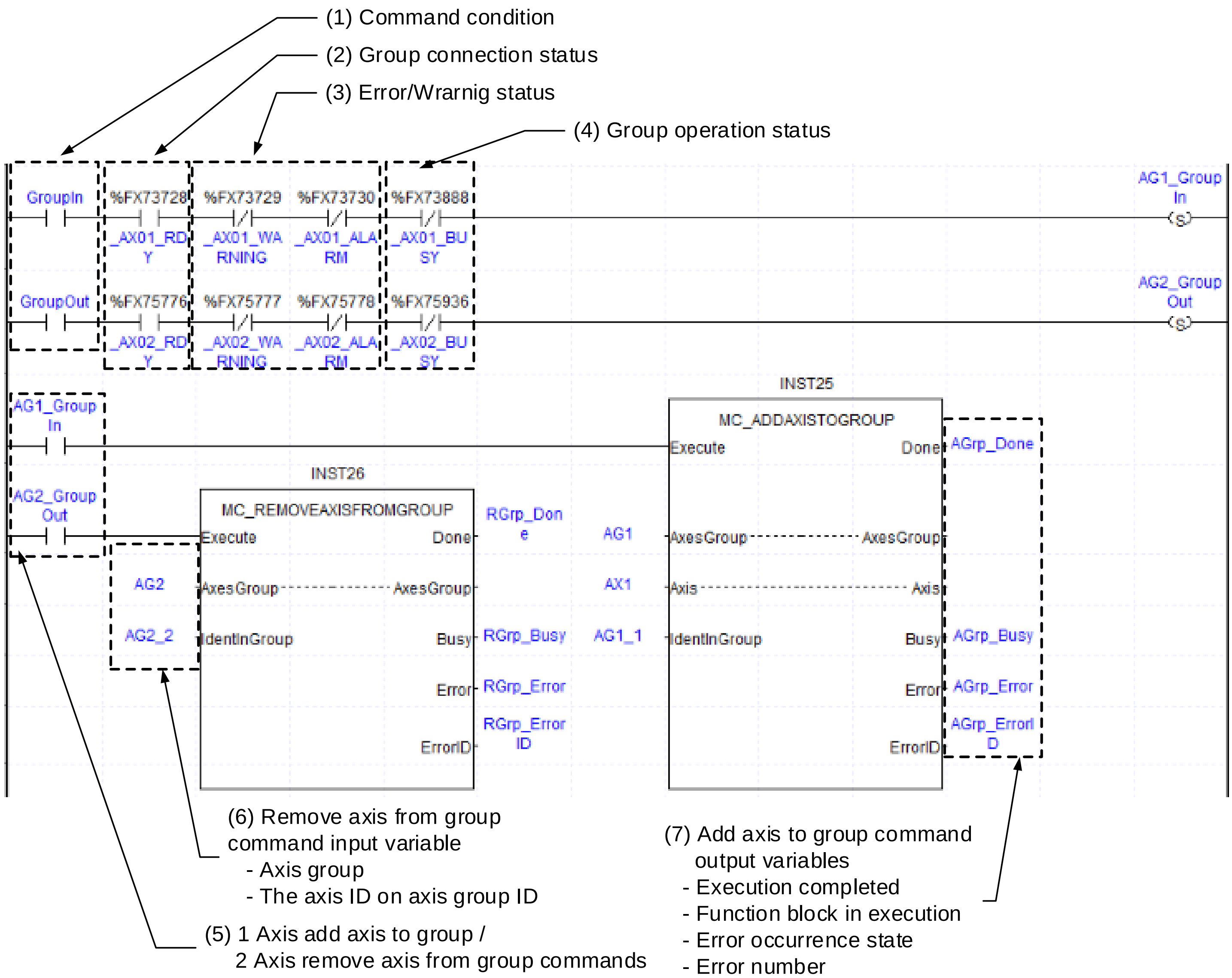
In addition, cam operation cancellation (MC_CamOut) motion function block is executed under the following conditions.
- Cam operation cancellation condition is On.
- The axis is normally connected.
- There should be no errors and warnings.
- The axis is in operation.

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.
- Changes in parameters during operation: It sets whether to apply to the operation by changing input variable values of the function block. For details, refer to “6.1.5 Changes in parameters during execution of motion function block”.
- 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 details, 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 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.
- Function Block axis control in operation: In case motion function block controls the axis, it is On.
- Error occurrence state: In case error occurs as the motion function block is 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
: In case 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 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.

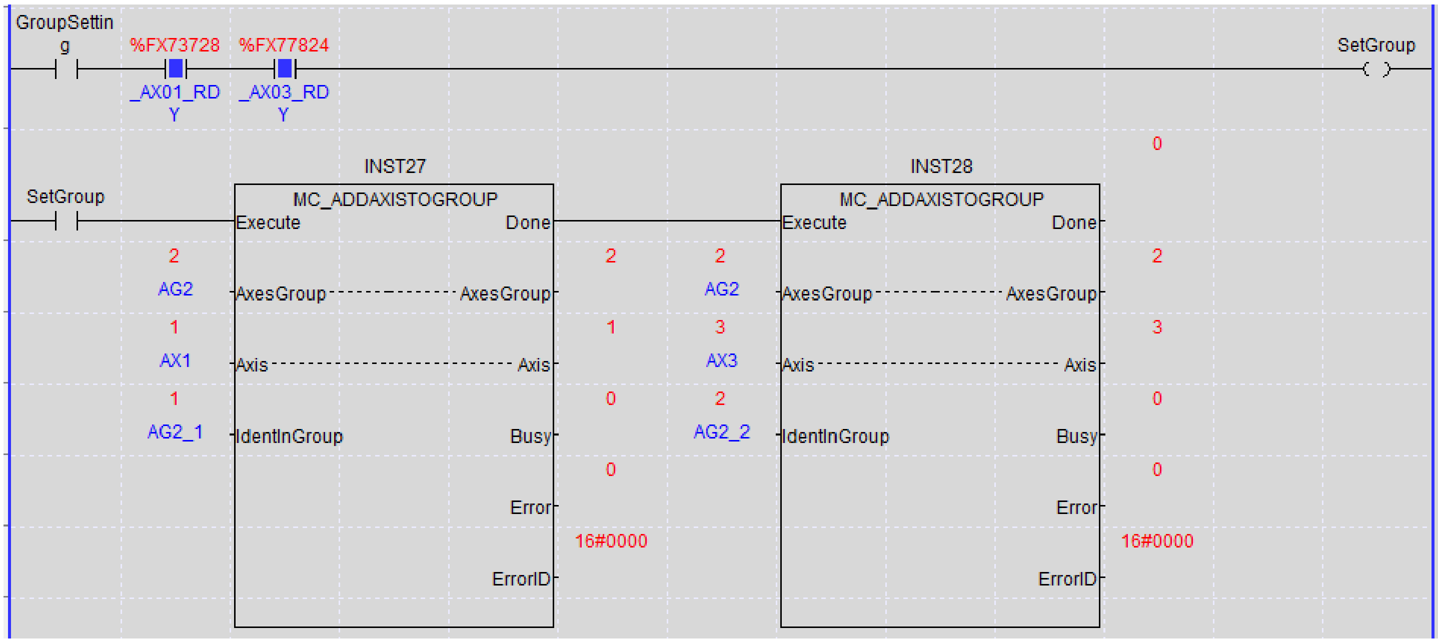
In addition, group axis exclusion (MC_RemoveAxisFromGroup) motion function block is executed under the following conditions.
- Remove axis from group condition is On.
- The axis is normally connected.
- There should be no errors and warnings.
- The axis is not in operation.

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.
- The axis ID on axis group ID: It sets ID values granted when the axis is included in axis group.

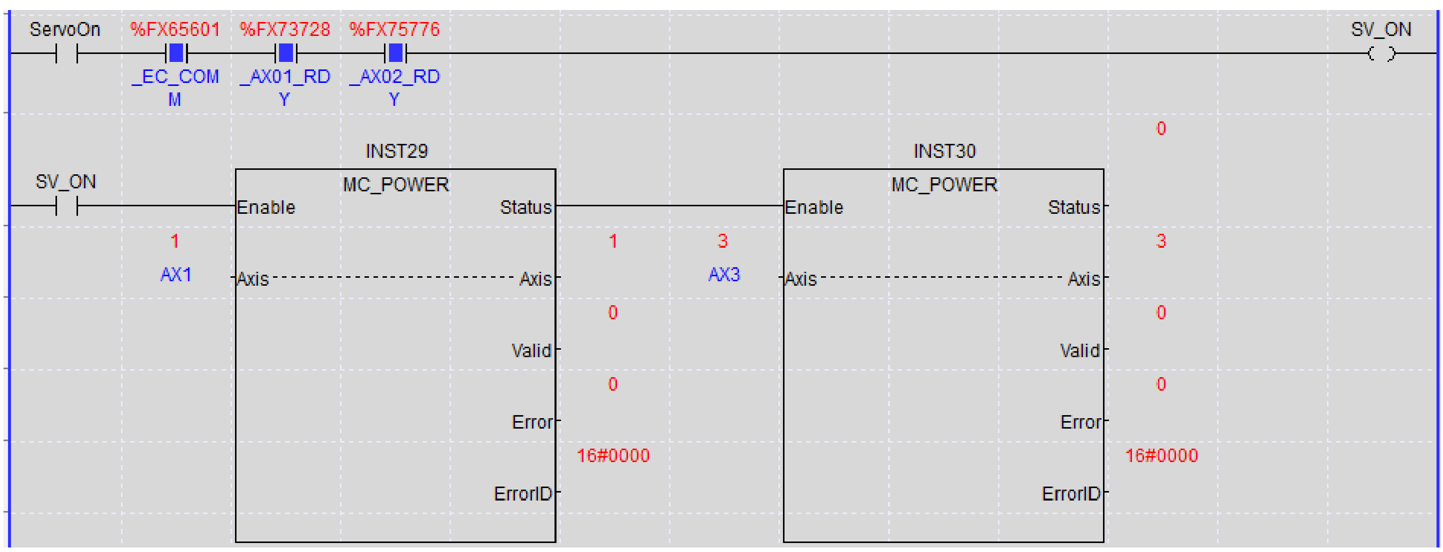
(7) Add axis to group command output variable
: It is a variable to store output values generated when add axis to group (MC_AddAxisToGroup) motion function block is executed.
- Execution completed: When motion function block is normally executed, it is On.
- Function Block in execution: When motion function block is executed, it is On, and the execution is completed, it becomes Off.
- Error occurrence state: In case error occurs as the motion function block is 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 Operation Example of Axis Group

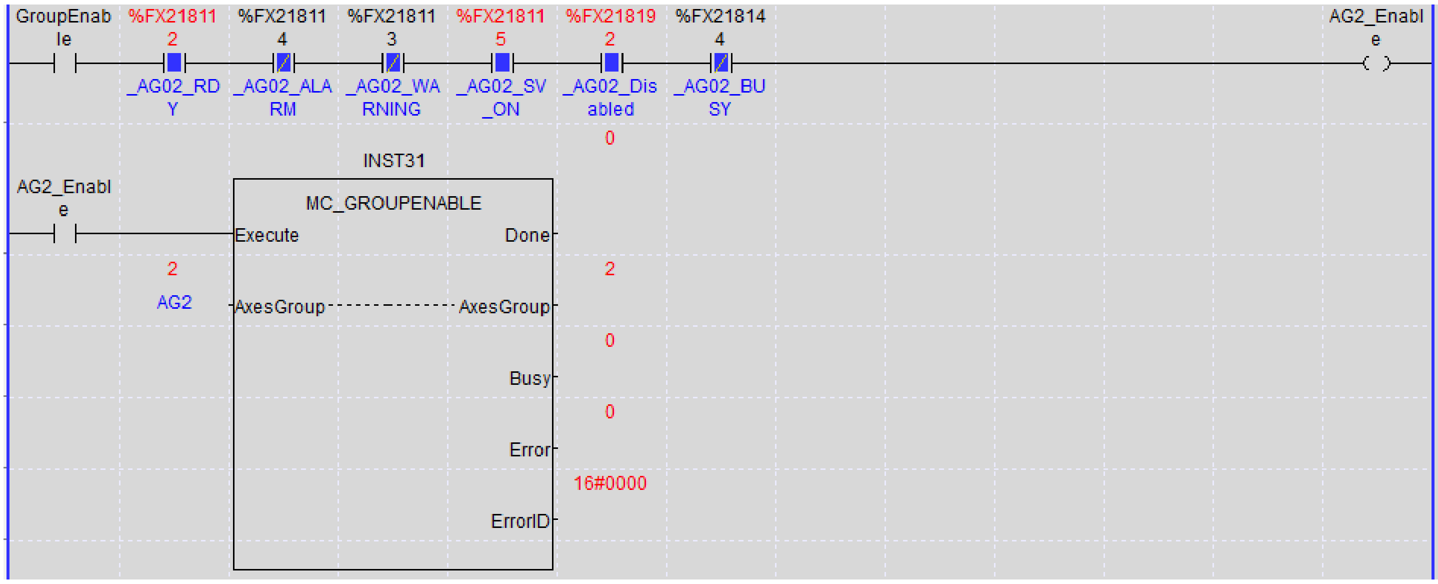
1. Group Setting



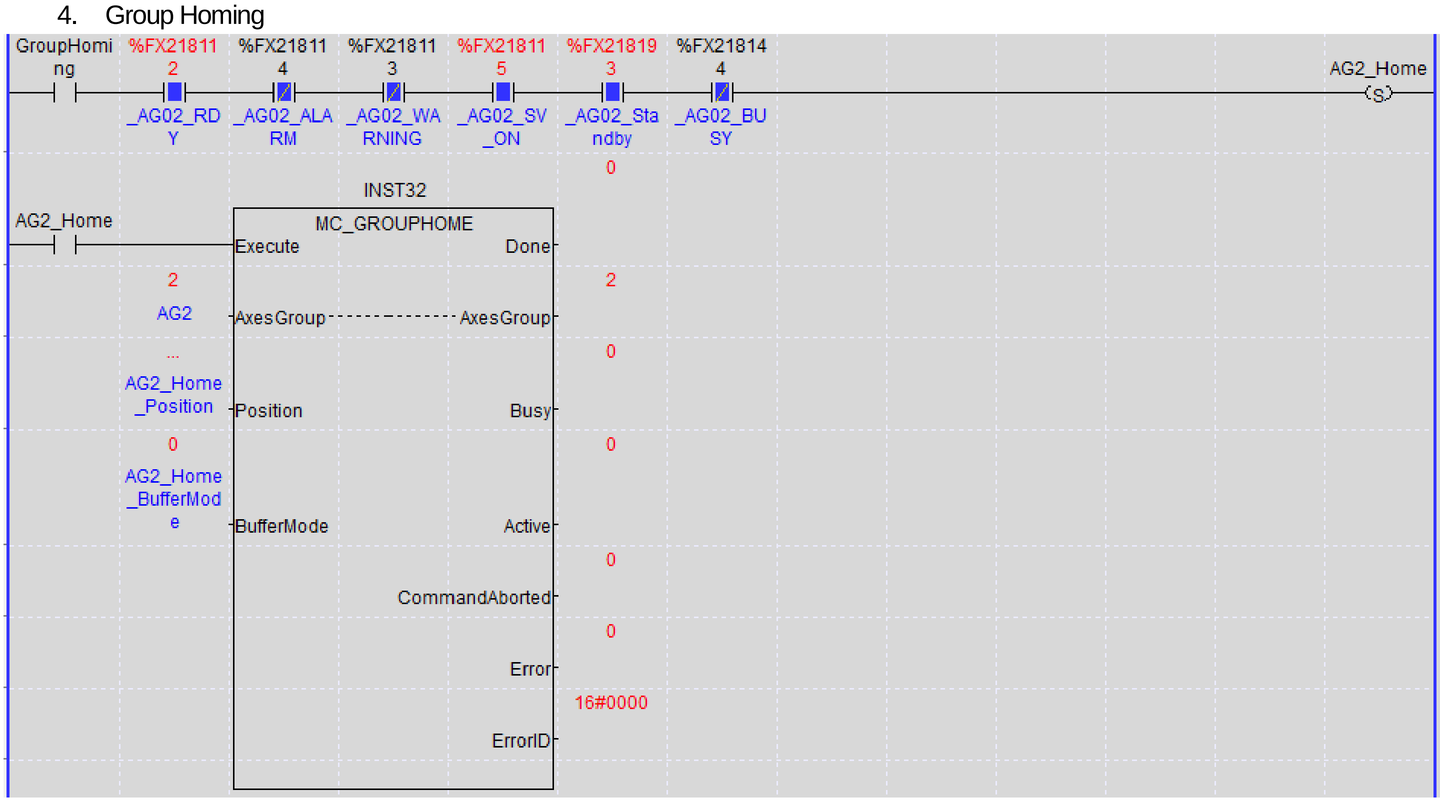
2. Servo On



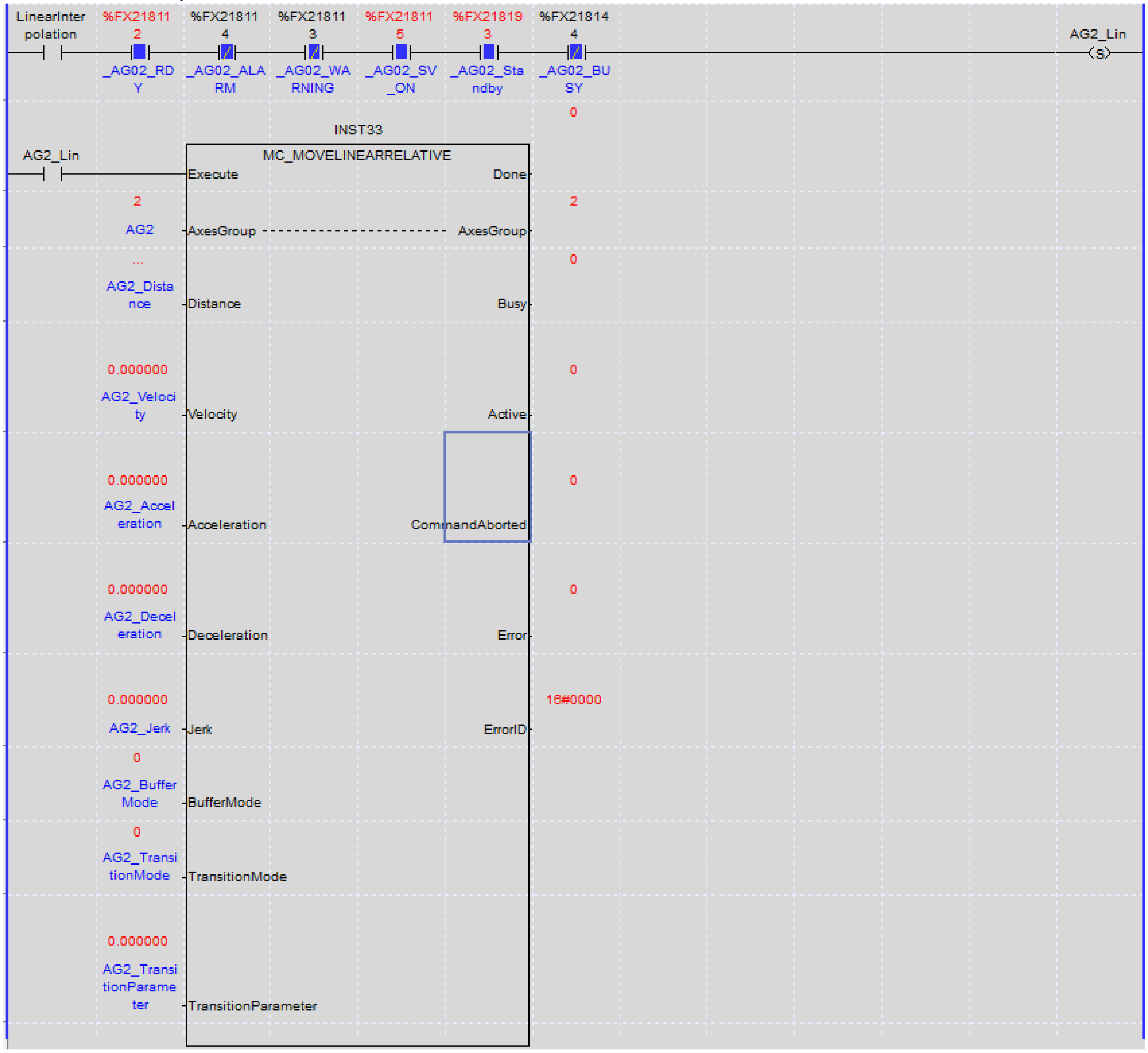
3. Group Enable



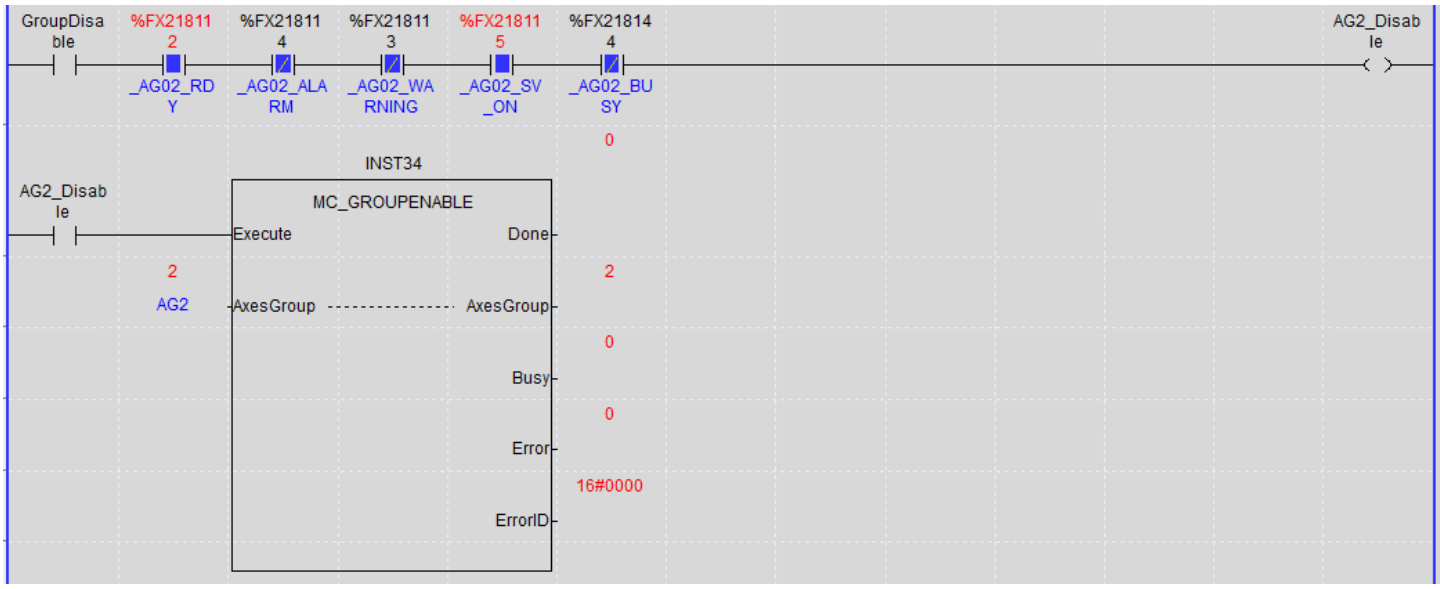
4. Group Homing



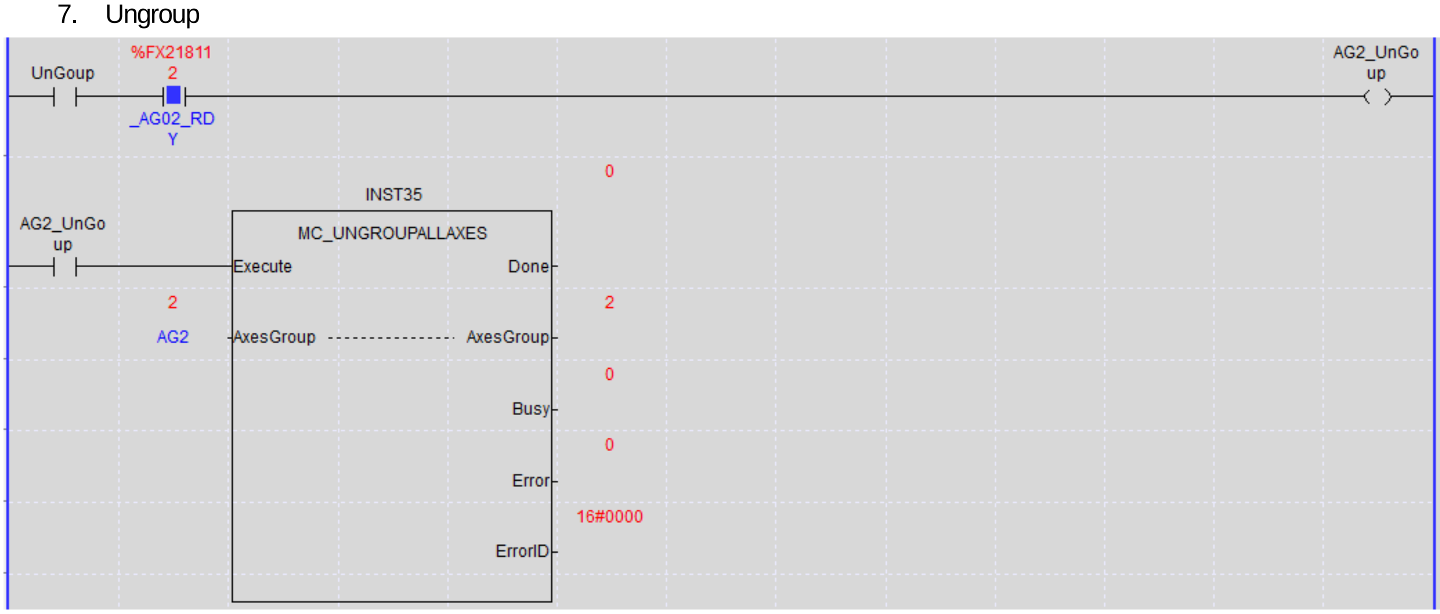
5. Linear Interpolation



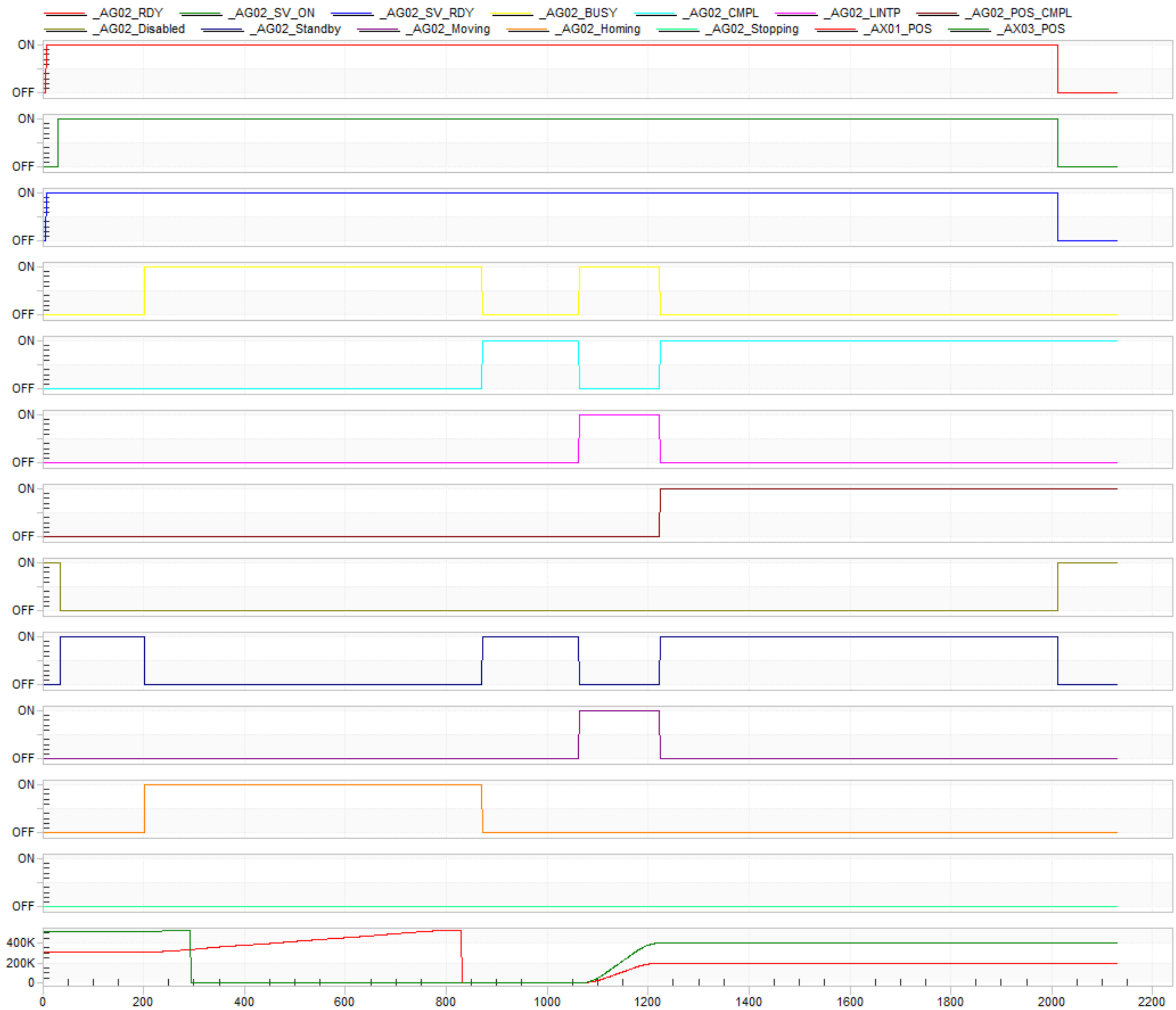
6. Group Disable

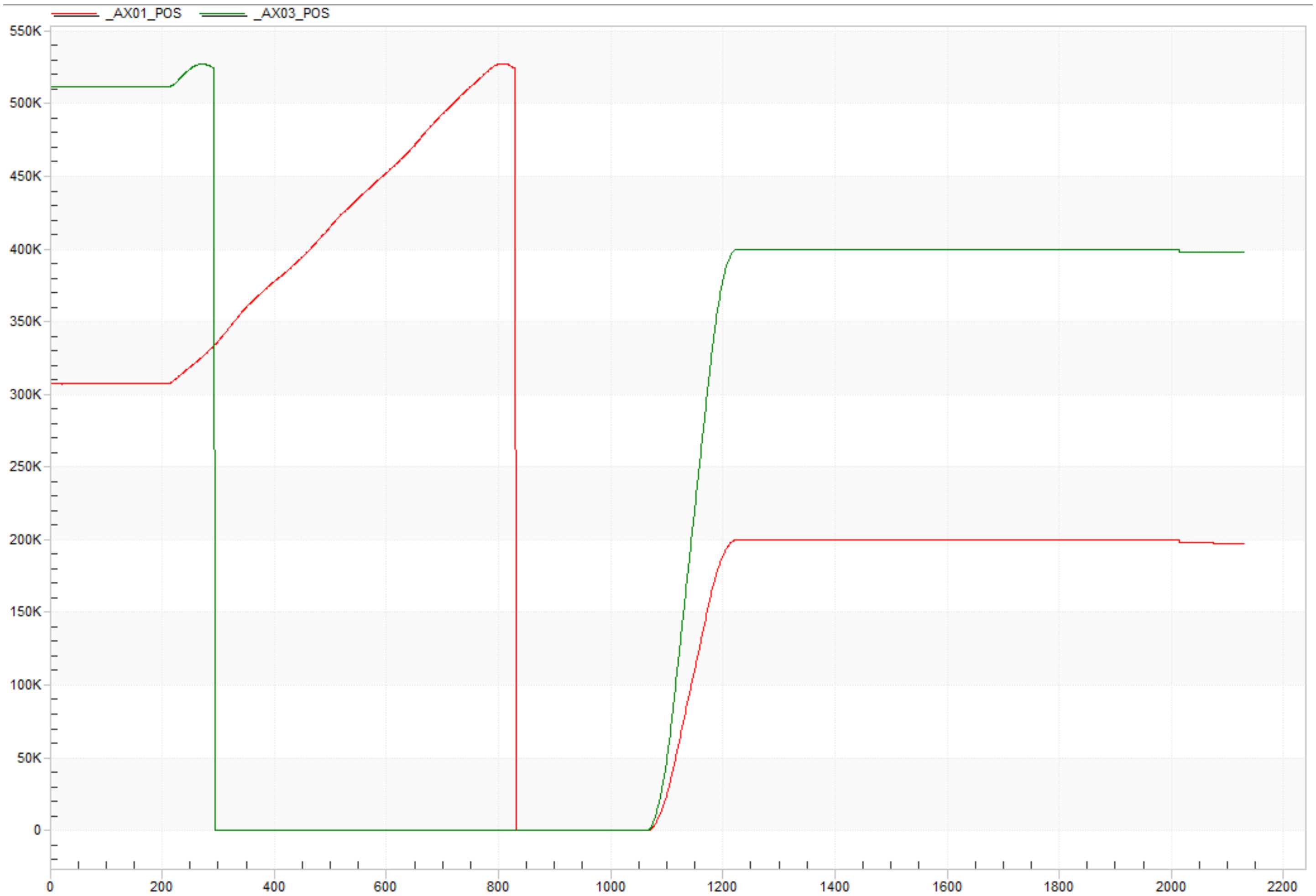


7. Ungroup



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 Input Signal Processing

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 details on the kinds and functions of flags, refer to “Appendix1. Flag”.

7.5.2 Output Signal Processing

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 details on the kinds and functions of flags, refer to “Appendix1. Flag”.

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.

(1) 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.

(2) 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.

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.

(1) In case of re-connection after servo drive power off when using an incremental encoder

(2) In case of re-connection after PLC power off/on when using an incremental encoder

(3) In case homing is not normally completed after the execution of homing command

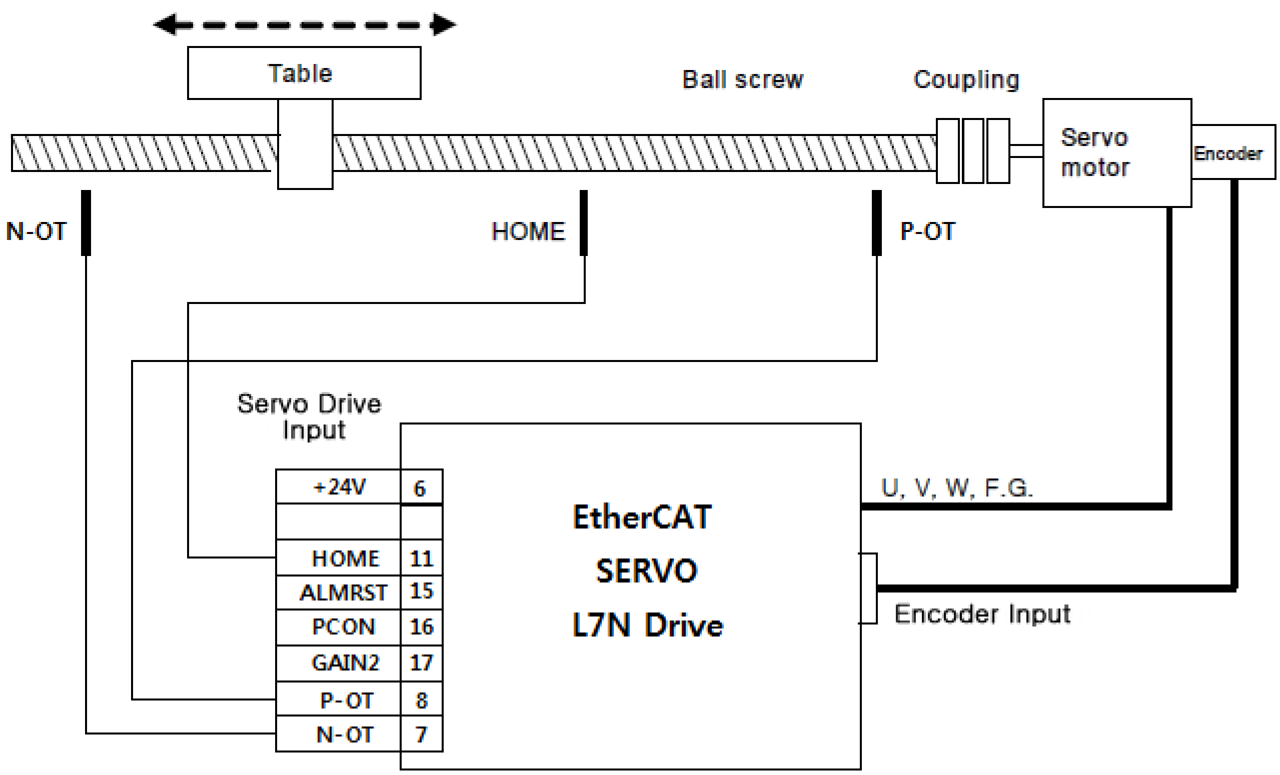
In case of the origin indetermination status as above, the origin determination should be executed for absolute coordinate position control operation.

8.1.2 Homing

1. Operation

Homing is performed to establish the origin of the machine after the power is applied. 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.

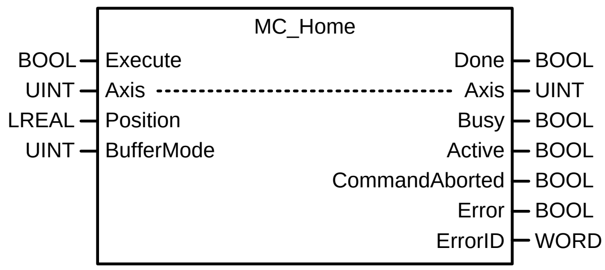
In motion control module, actual operation after starting homing is performed in servo drive, and homing method to support complies with servo drive. Before setting the homing, homing-related parameters are to be set in servo parameters of the axis.

■ Example of setting homing parameters

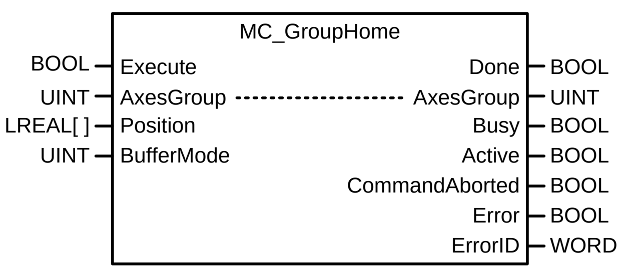
Index	Name	Unit	Current Value	Initial Value	Access
6098	Homing Method	-	0x22	0x22	rw
6099:00	Homing Speeds	-	0x02	0x02	rw
6099:01	Speed during search for switch	Vel.Unit	0x000000A0	0x000000A0	rw
6099:02	Speed during search for zero	Vel.Unit	0x00000020	0x00000020	rw
609A	Homing Acceleration	Acc.Unit	0x0000C350	0x0000C350	rw

■ Relevant motion function block

Name	Description	Operation Condition
MC_Home	Perform homing	Edge

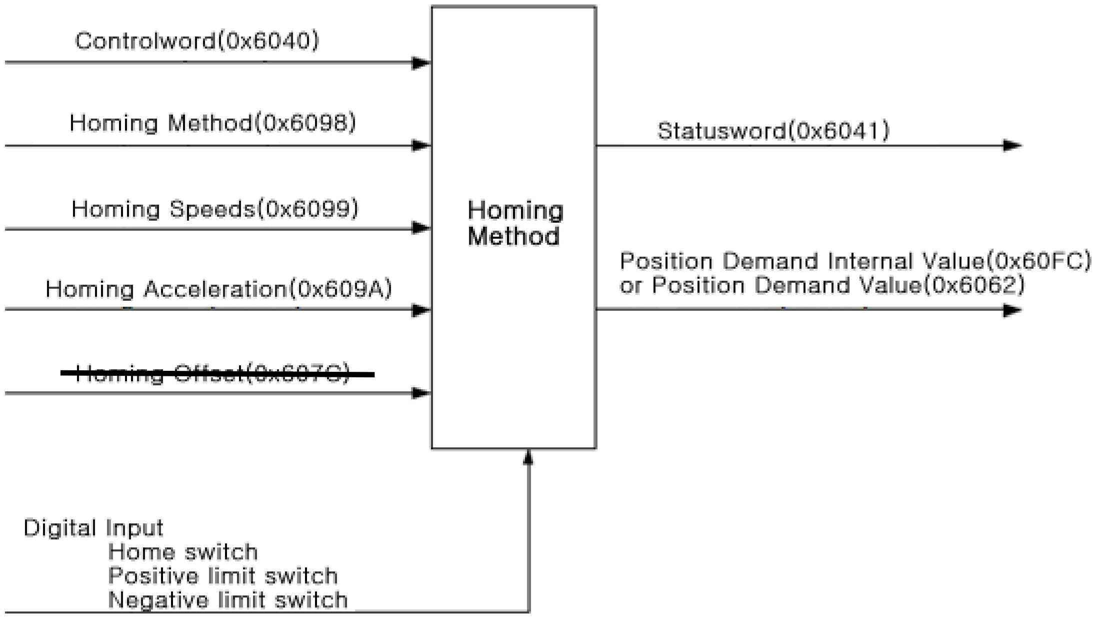


Name	Description	Operation Condition
MC_GroupHome	Perform group homing	Edge



2. XDL- N Series servo drive homing parameters and operation

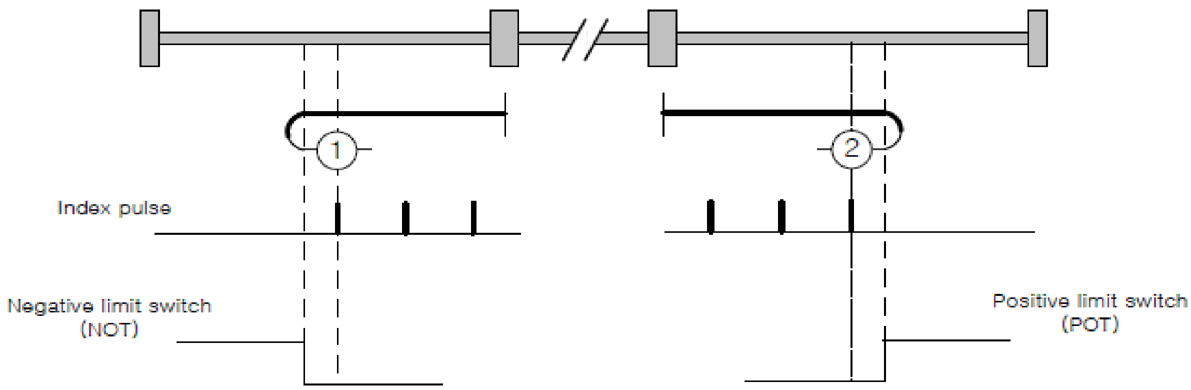
The following figure shows input and output definitions of homing-related XDL N series servo drive parameters. The velocity, acceleration and homing methods can be specified. Here, the origin (Home) offset gets the origin of user coordinate system applied as the origin. However, Home offset can't be used. Set by Position input variable of MC_Home, MC_GroupHome.

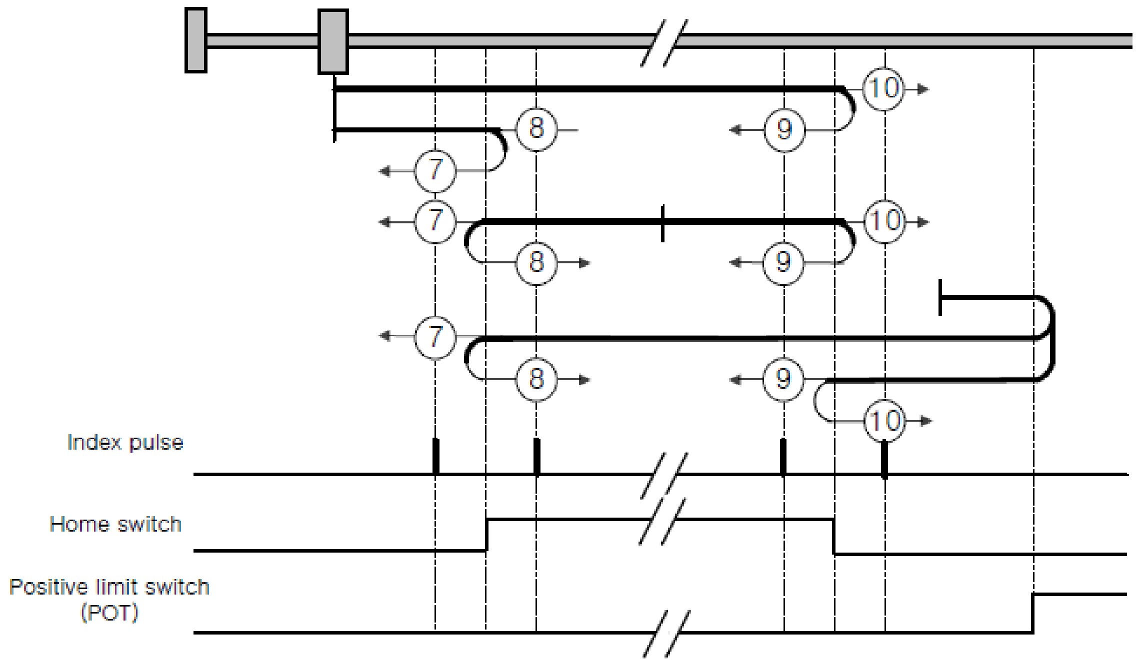


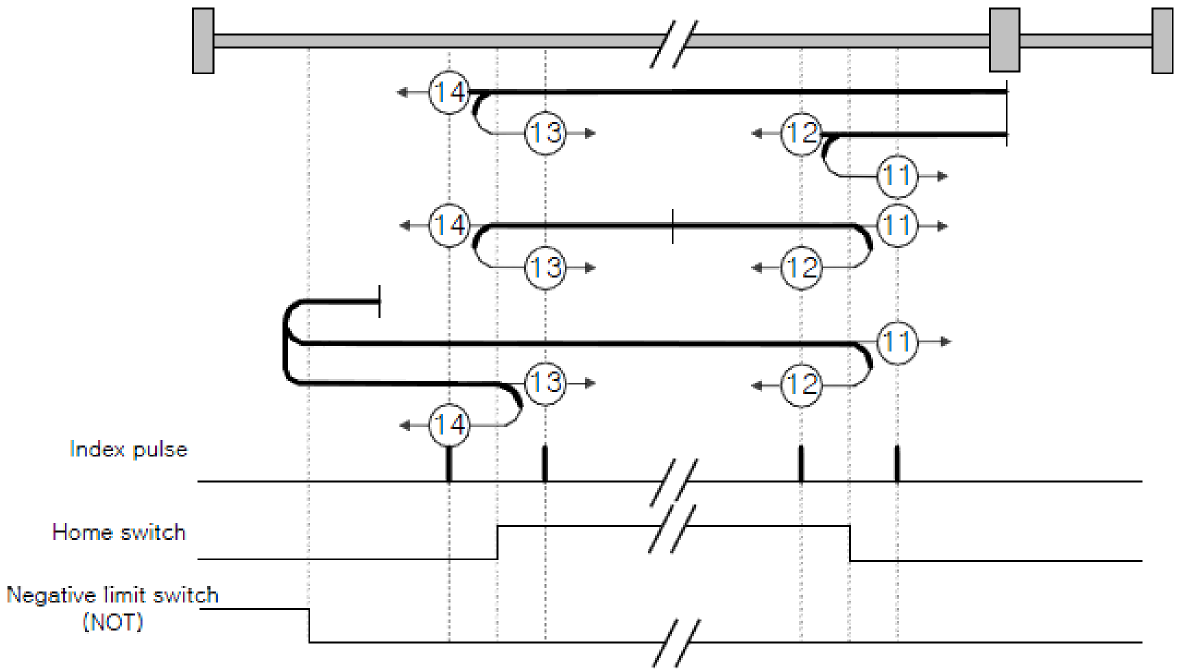
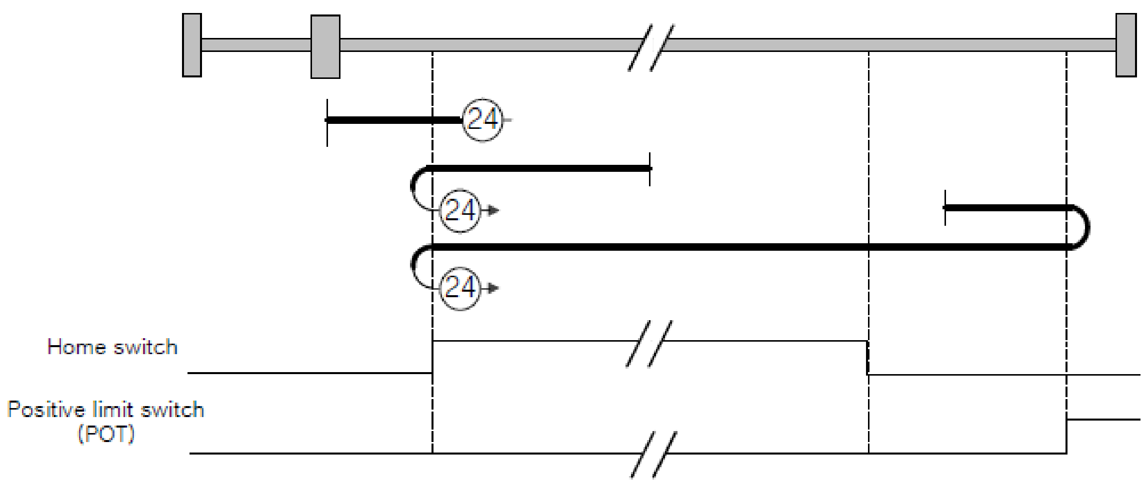
(1) Parameter related to homing

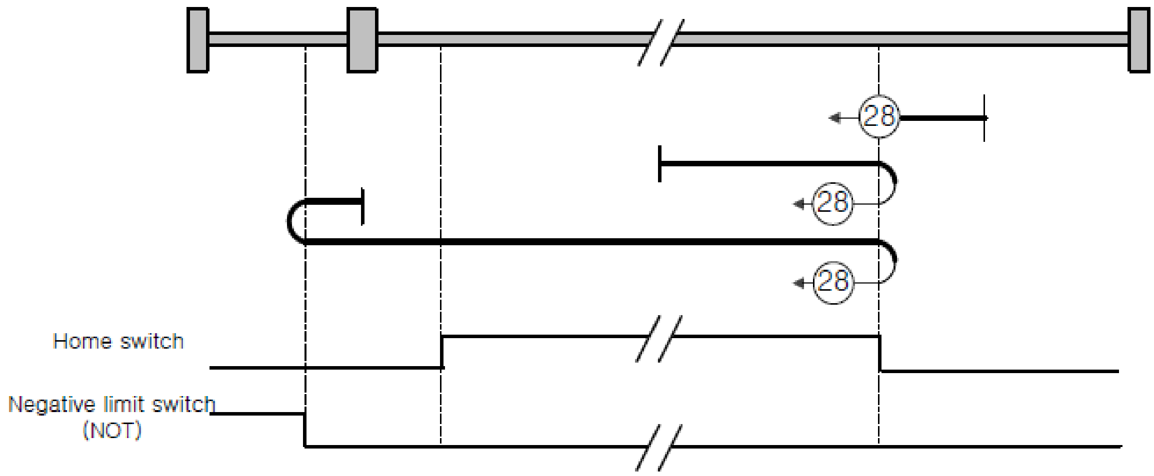
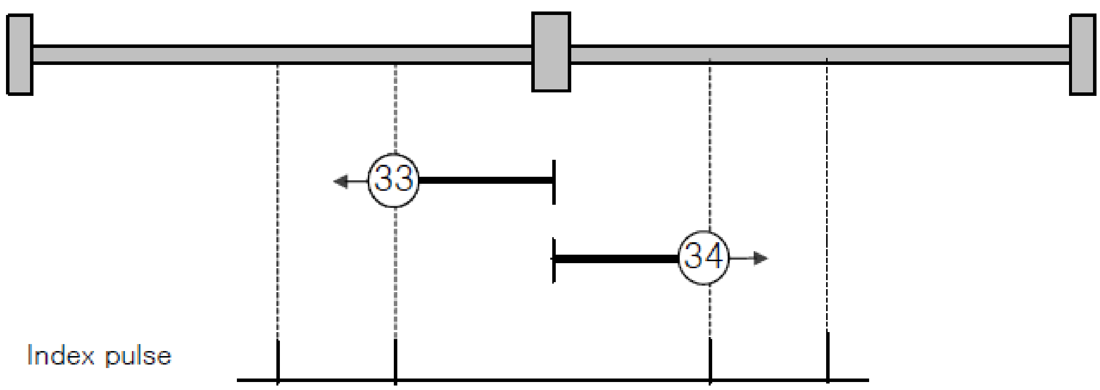
Index	Sub	Name	Data Type	Unit
0x6040	-	Control word	UINT	-
0x6041	-	Status word	UINT	-
0x607C	-	Homing Offset	DINT	[pls]
0x6098	-	Homing Method	SINT	-
0x6099	-	Homing Speeds	-	-
	0	Item Number	USINT	-
	1	Speed during search for switch	UDINT	[pls/s]
	2	Speed during search for zero	UDINT	[pls/s]
0x607D	-	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]

(2) Homing Method(0x6098)

Value	Description
0	No Homing
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 forward direction CW after POT switch is On becomes the Home position. 
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) Lower figure: If POT switch is Off, and the Home switch is On, operation is made at switch search speed, and the initial movement direction becomes reverse 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.

Value	Description
7~10	
11~14	The methods described for 11 to 14 determines the Home position using the Home switch and the NOT switch. (11) Upper figure: If the NOT switch is Off, then the drive operates at switch search speed and rotates CW. If the Home switch is turned on at this time, it changes the direction of rotation, and the position that the first index pulse encounters while driving CCW at zero search speed becomes the Home position. (11) Middle figure: If the NOT switch is Off and the Home switch is On, then the drive operates at switch search speed and rotates CCW. If the Home switch is turned off at this time, it transfers to zero search speed. After the Home switch is turned off, the position that the frist index pulse encounters while driving in the CCW direction becomes the Home position. (11) Lower figure: If the NOT switch is Off, then the drive operates at switch search speed and rotates CW. If the NOT switch is turned on at this time, it changes the direction and continues to drive CCW at switch search speed. If the Home switch is changed from On to Off, then it transfers to zero search speed, and the position that the first index pulse encounters becomes the Home position. The methods from 12 to 14 are identical to the methods for 11 in terms of how they determine the Home position. The only differecnces are the initial driving direction and Home switch polarity. Refer to the following figure.

Value	Description
	
24	(8) The Home position is determined as in (8) method, but index pulse is not used. In addition, the point where the Home switch is On/Off becomes the Home position. 

<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>28</td><td>(12) The Home position is determined as in (12) method, but index pulse is not used. In addition, the point where the Home switch is On/Off becomes the Home position.
</td></tr>
<tr><td>33, 34</td><td>The location that meets index pulse first during movement in the reverse direction CCW/forward direction CW becomes the Home position.
</td></tr>
<tr><td>35</td><td>Homing operation starting point becomes the Home position.</td></tr>
</table>

Note) ▬ : Speed during search for switch (0x6099:01), → : Speed during search for zero (0x6099:02)

8.2 Type of Control Operation

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)」)

(1) It conducts position control from starting position to target position (location specified in 'Position' of absolute position operation command).

(2) The position control is carried out based on the position (the origin position) specified in the homing.

(3) In direction (Direction) input, the direction to be operated is specified. It is valid only if operation parameter 「Infinite running repetition」 setting is '1: Enable'.

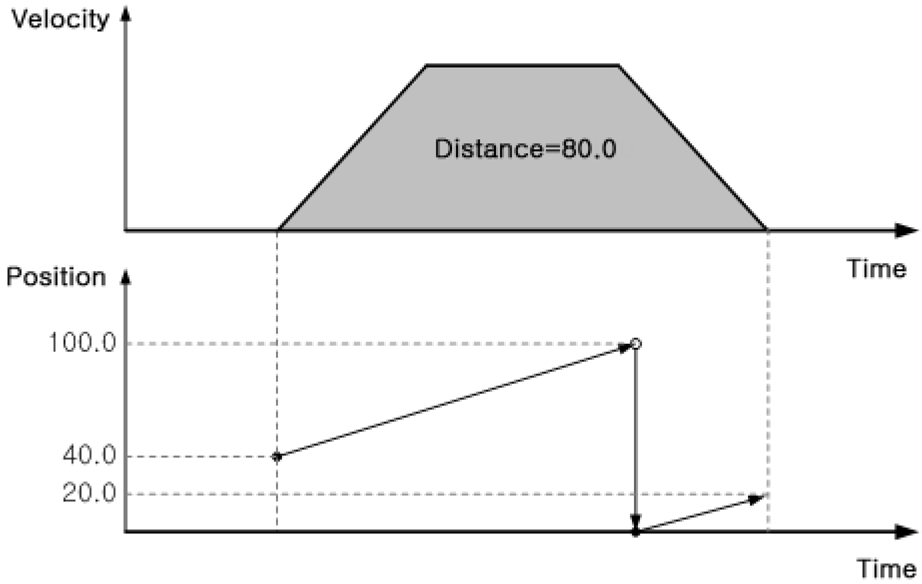
- Setting value: 0-Not specified, 1–Forward direction, 2–Shortest distance direction, 3–Reverse direction, 4–Current direction
- When the shortest direction distance is specified, the operation is made by selecting the direction that can go to the shortest direction automatically depending on the form of the axis.
- Motions according to the direction (Direction) input are as follows.

(a) 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
- Starting position: 40.0
- Target position: 120.0
- Command position after the absolute position operation = 120.0 – (100.0 x 1) = 20.0

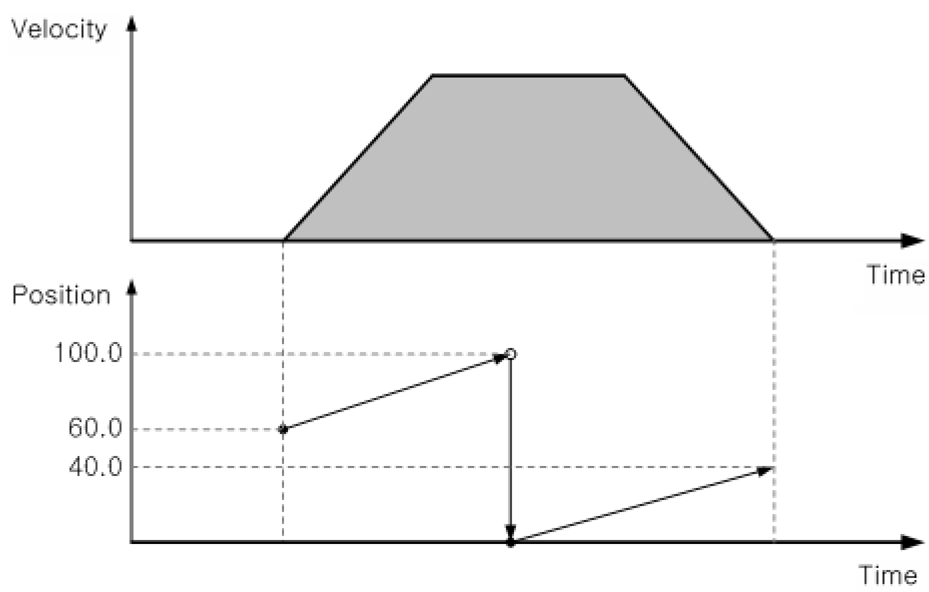


(b) 1-Forward direction

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
- Starting position: 60.0
- Target position: 40.0



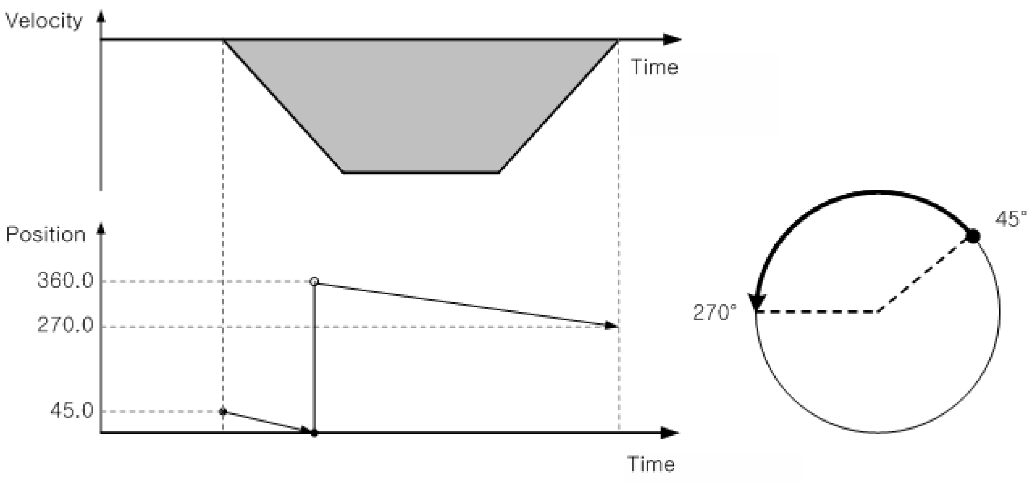
(c) 2-Shortest distance direction

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
- Starting position: 45.0
- Target 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.

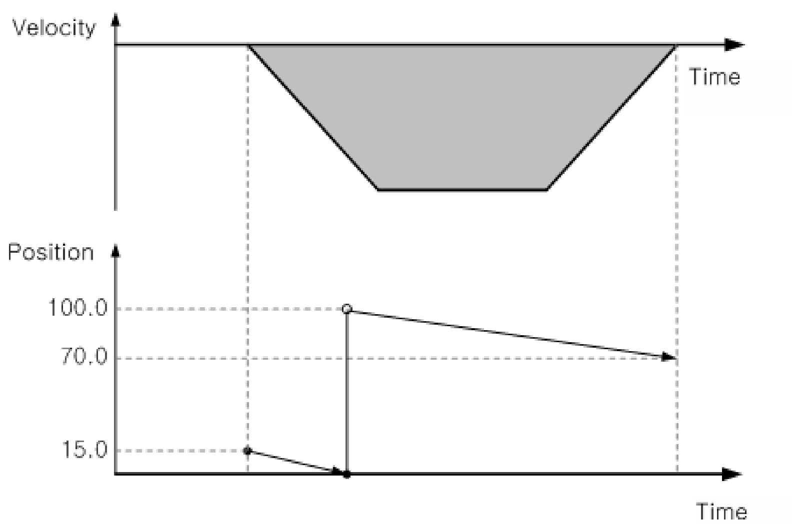


(d) 3-Reverse direction

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
- Starting position: 15.0
- Target position: 70.0



(e) 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.

(4) In case operation parameter 「Infinite running repetition」 setting is '0: disable', operating direction is determined as follows.

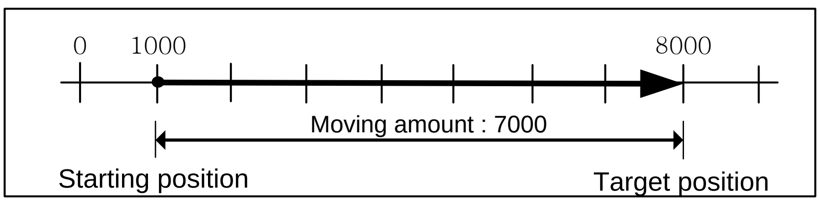
- Starting position < target position: Positioning operation in forward direction
- Starting position > target position: Positioning operation in reverse direction

[Example] Executes Absolute coordinate, single-axis position control with the following setting

▷Start position: 1000,

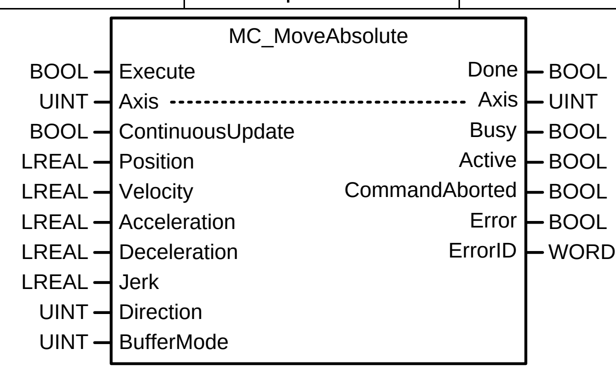
▷Target position: 8000

The moving amount to forward direction is 7000 (7000=8000-1000).



■ Relevant motion function block

Name	Description	Operation Condition
MC_MoveAbsolute	Absolute positioning operation	Edge



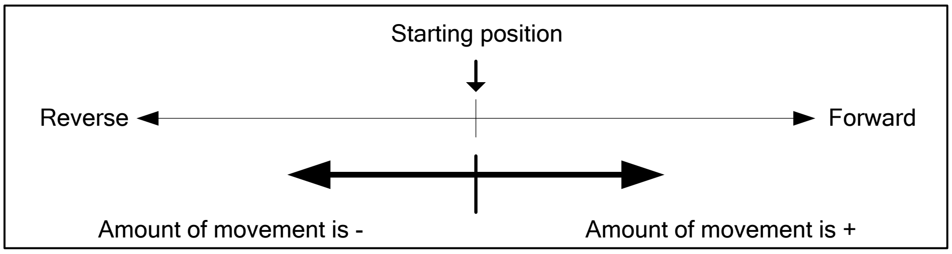
2. Control by Incremental method (「Relative positioning operation(MC_MoveRelative)」)

(1) 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.

(2) Transfer direction is determined by the sign of moving amount.

▷ Transfer direction (+) or no sign: forward direction positioning (starting position increases)

▷ Transfer direction (-) : reverse direction positioning (starting position decreases)

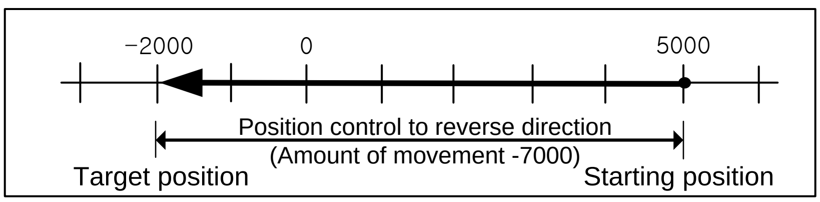


[Example] Executes Absolute coordinate, single-axis position control with the following setting

▷ Start position: 5000,

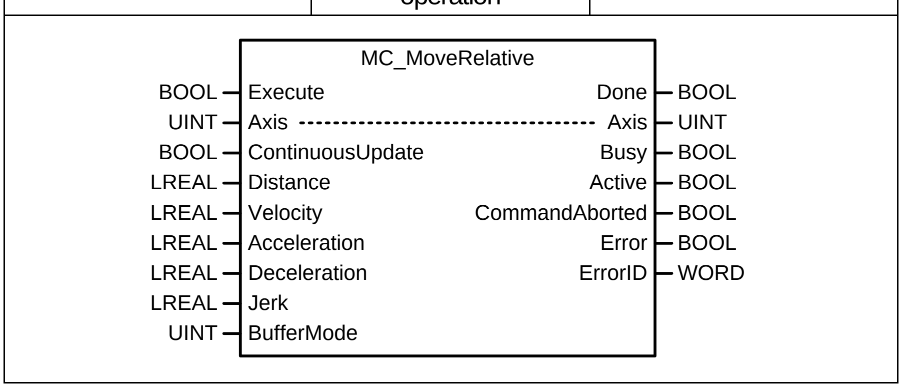
▷ Target position: -7000

It goes to reverse direction and stops at the –2000.



■ Relevant motion function block

Name	Description	Operation Condition
MC_MoveRelative	Relative positioning operation	Edge



8.2.2 Single-axis Speed 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

(1) 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.

(2) 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

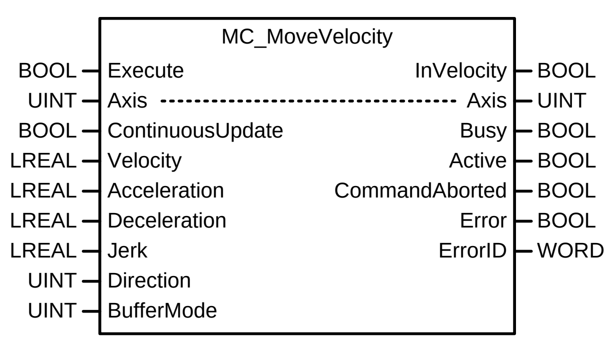
(3) Negative number can be set for target velocity (Velocity) input value. In case the target velocity setting value is negative number, operating direction becomes the opposite direction of the previously specified direction.
■ Forward operation
- Velocity > 0, Direction=1: Forward
- Velocity < 0, Direction=2: Reverse
■ Reverse operation

(4) 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.

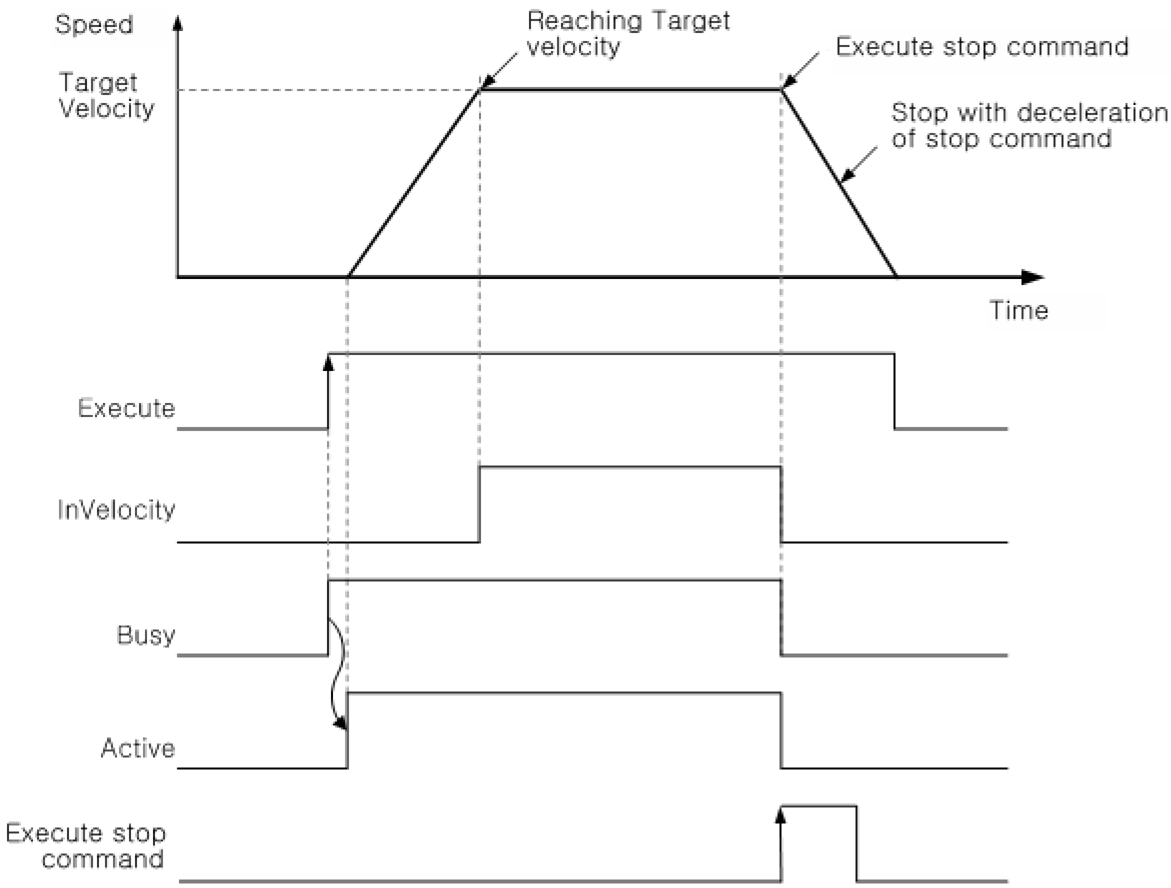
(5) The speed control which is currently being executed is stopped with halt (MC_Halt) or immediate stop (MC_Stop) motion function block.

2. Relevant 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

(1) Torque control of the specified axis is made using target torque value and torque rising slope.

(2) 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)

(3) Torque control mode is executed using torque control mode of servo drive.

(4) Target torque values are rounded to two decimals and reflected in [0.1%] unit.

(5) In Direction input, the direction to be operated is specified.

(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-Current direction

(6) 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: Forward
- Torque < 0, Direction=2: Reverse

■ Reverse operation

- Torque > 0, Direction=2: Reverse
- Torque < 0, Direction=1: Forward

(7) The setting range of the torque values are as follows.

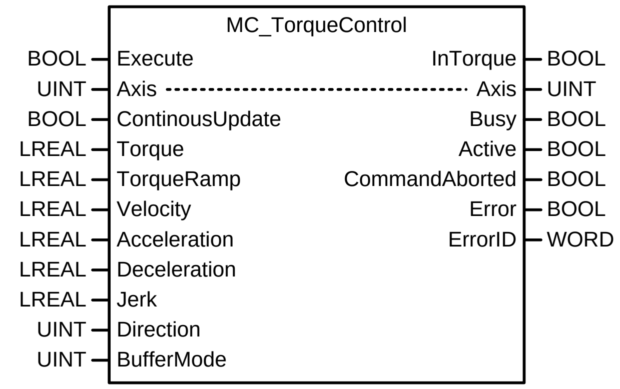
-1000.0 % ~ 1000 %

(8) 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.

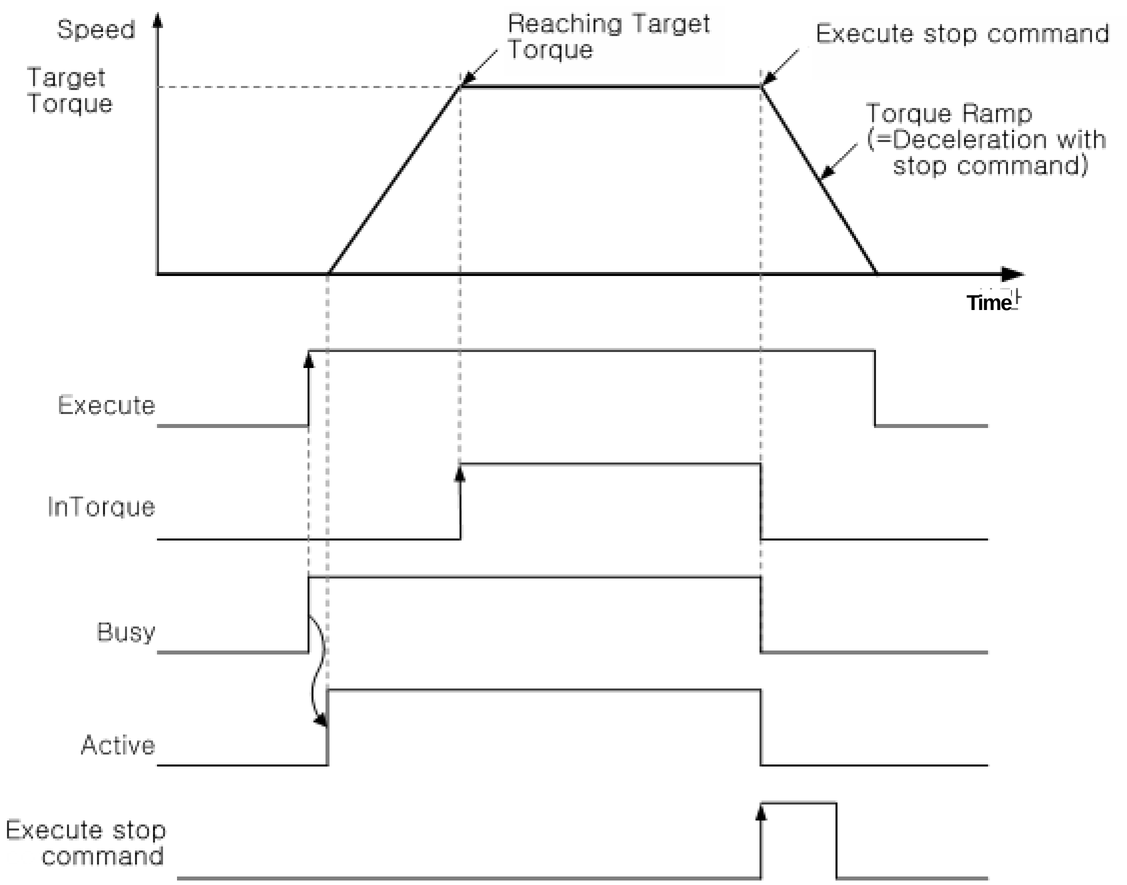
(9) Torque control which is currently being executed is stopped with halt(MC_Halt) or immediate stop (MC_Stop) motion function block.

2. Relevant 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

(1) Position control that ends with end rate specified from staring position to target position is carried out. .

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

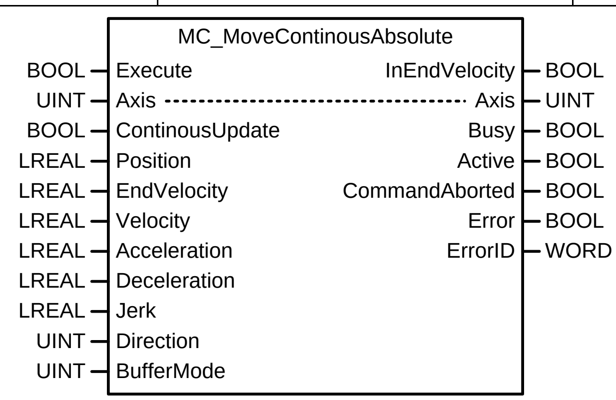
(3) In case of 「Specified speed operation after the absolute position operation (MC_MoveContinuousAbsolute)」, the direction to operate is specified in Direction input, which is valid only if operation parameter 「Infinite running repetition」 is set to '1: Enable'.

■ Setting value: 0-Not specified, 1–Forward, 2–Shortest distance direction, 3–Reverse, 4–Current direction

(4) The end rate is reached after the completion of position control operation to target position, InEndVelocity outp ut of function block is On. If there is a pending command, the pending command is executed after InEndVelocity output is On.

2. Relevant motion function block

Name	Description	Operation Condition
MC_ MoveContinuousAbsolute	Specified velocity operation after Absolute position operation	Edge

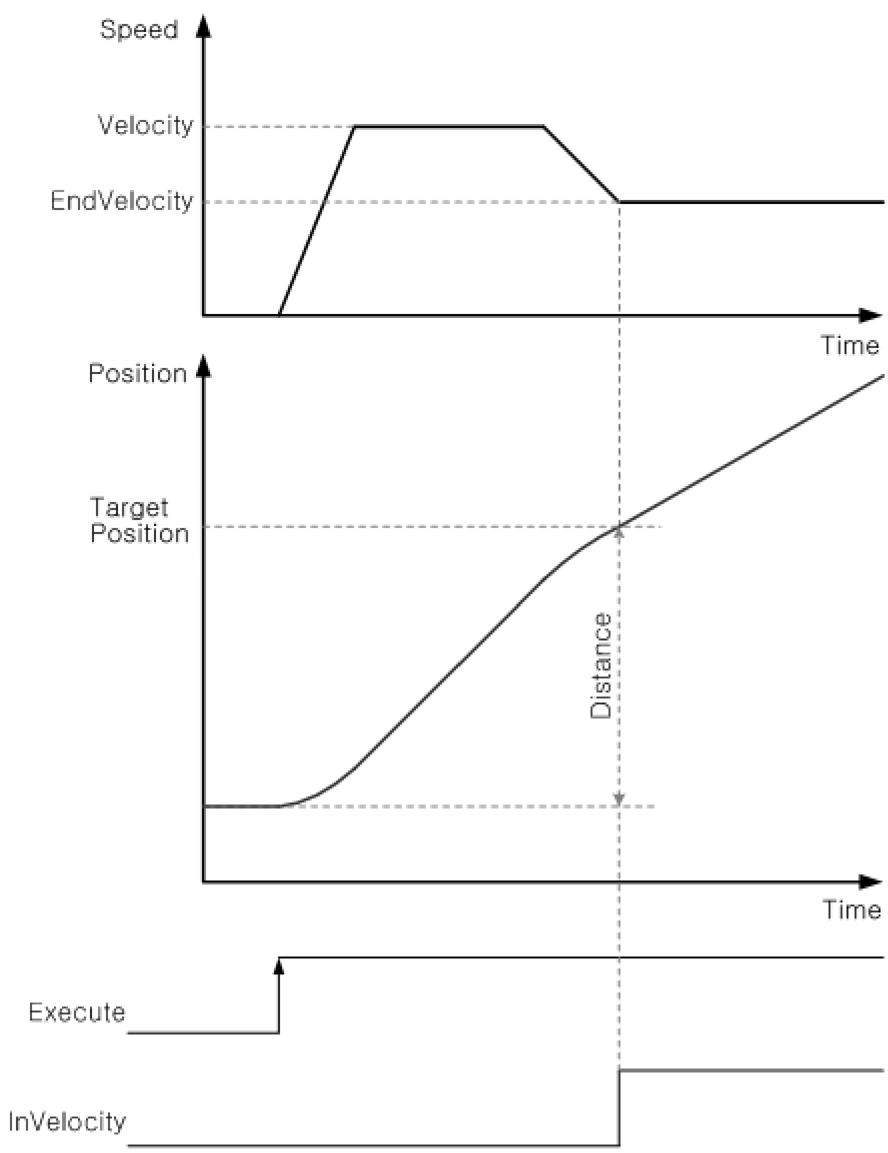


Name	Description	Operation Condition
MC_ MoveContinuousRelative	Specified velocity operation after Relative position operation	Edge

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-velocity 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. Velocity-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 torque control operation, the torque control is switched to position control, when torque value is reduced to 0, and position control continues to operate after a stop. 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.

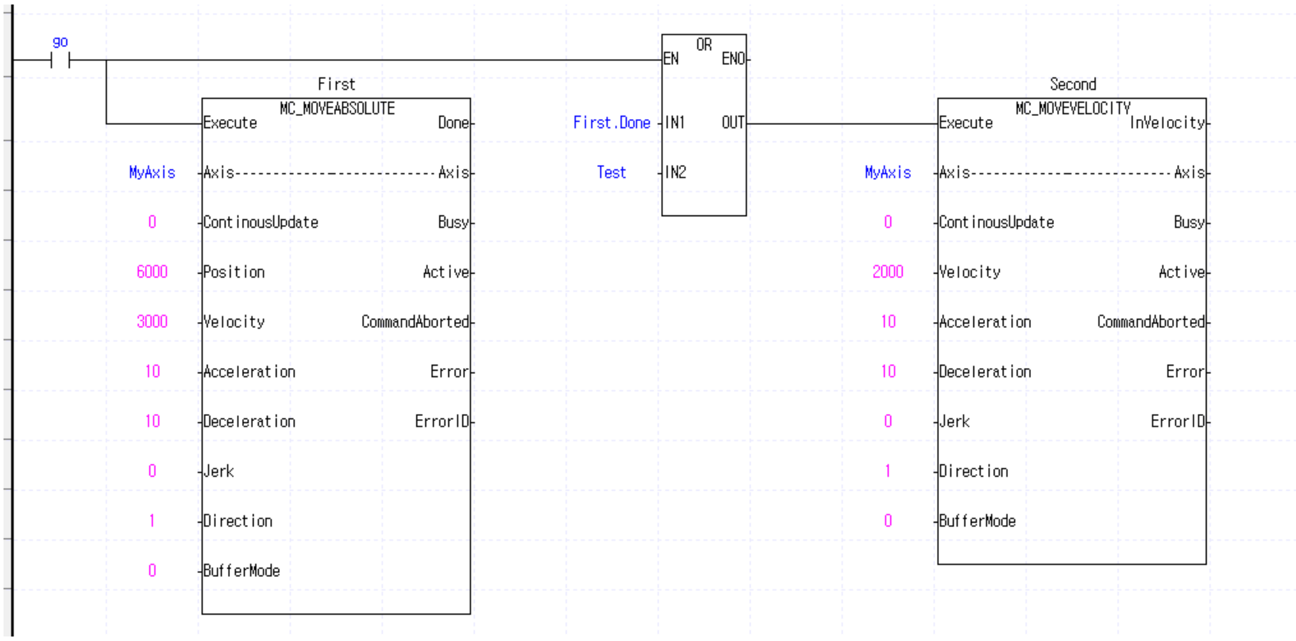
5. Velocity –torque switching control

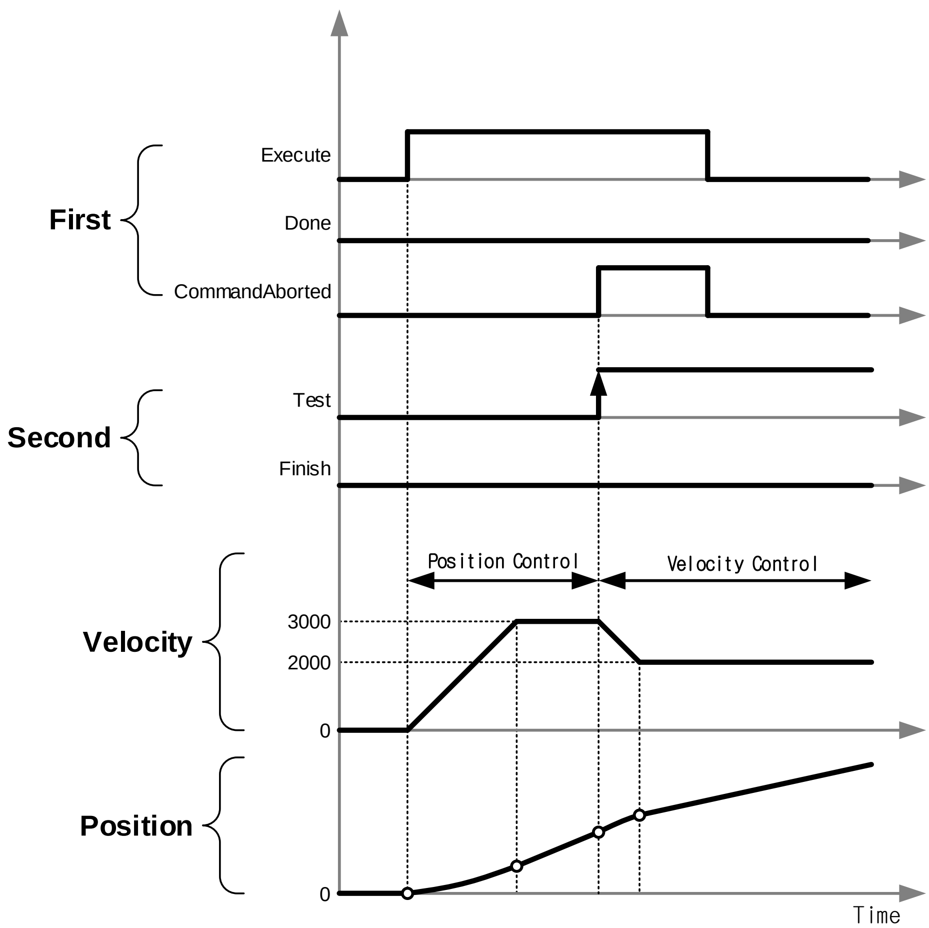
When torque control (MC_TorqueControl) motion function block is executed in the axis in specified speed operation during velocity control, the velocity control is switched to torque control. The torque at the time of being changed to torque control is operated continuously from the current torque value 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 specified speed operation (MC_MoveVelocity) motion function block is executed in the axis in torque control operation during speed control, the torque control is switched to velocity control, when torque value is reduced to 0, and velocity control continues to operate after a stop. 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.

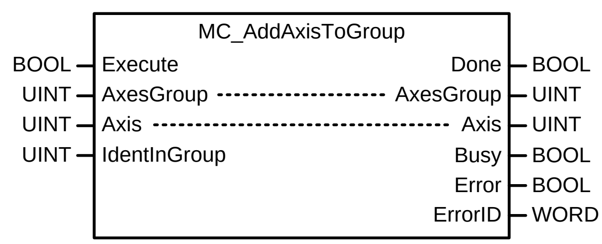
When axis group is configured, axis group operation can be executed after enabling the axis group.

(1) Add axis to group

It means adding an axis to the axis group. The configuration axis specified into IdentInGroup 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	Add axis to group	Edge



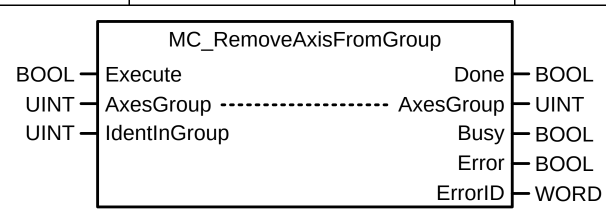
(2) Remove axis from group

It means removing an axis from the axis group. The configuration axis specified into IdentInGroup 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.

In case there are no remaining axes in the axis group, the axis group is changed to disabled state.

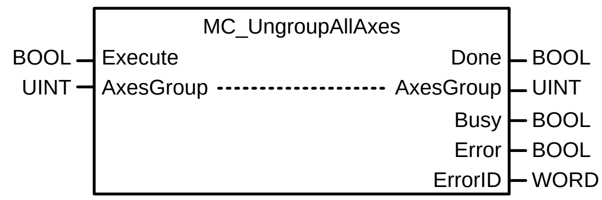
Name	Description	Operation Condition
MC_RemoveAxisToGroup	Remove axis from group	Edge



(3) Remove all axes from group

It means removing all axes from the axis group.

Name	Description	Operation Condition
MC_UngroupAllAxes	Remove all axes from group	Edge



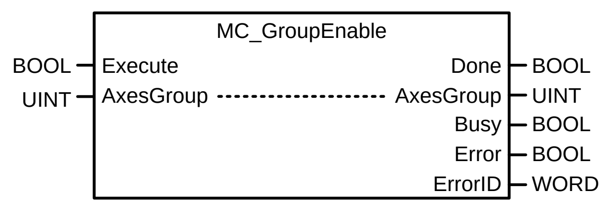
(4) Enable Group

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	Enable group	Edge

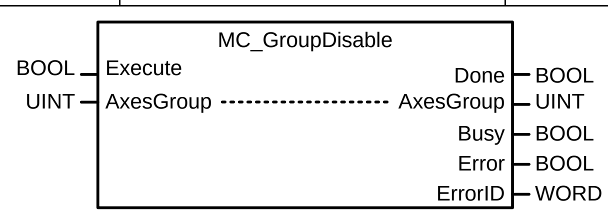


(5) Disable Group

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	Disable group	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

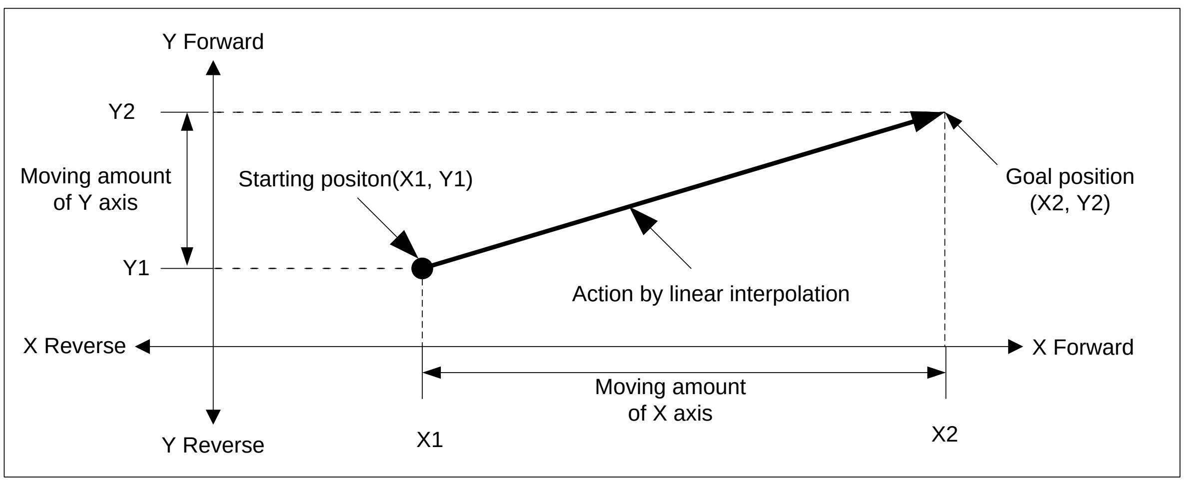
(「Absolute positioning linear interpolation operation(MC_MoveLinearAbsolute)」)

(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) The direction of movement depends on the starting position and the target position for each axis.

■ Starting position < target position: Positioning operation in forward

■ Starting position > target position: Positioning operation in reverse



(3) Interpolation that is currently being executed is stopped with group halt (MC_GroupHalt) or group immediate stop (MC_GroupStop) motion function block.

(4) 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.

Speed of each-axis and operating speed are as follows.

$$Interpolatinspeed\ (F) = Operationspeedsetinpositiondata$$

$$Interpolatingmovingamount(S) = \sqrt{S_1^2 + S_2^2 + \ldots + S_{10}^2}$$

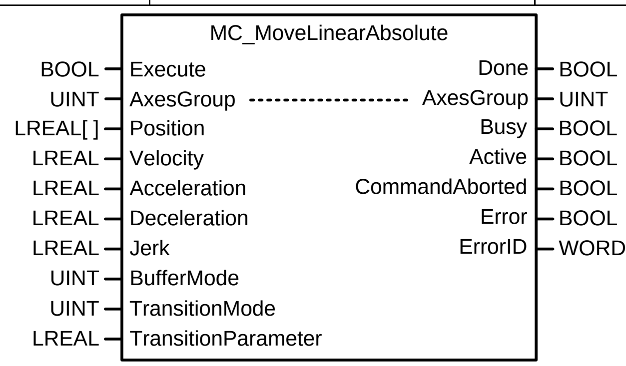
$$Axis1speed(V_1) = Interpolatingspeed\ (F) \times \frac{Main\,axismovingamount(S_1)}{Interpolatingmovingamount\ (S)}$$

$$Axis2\ \mathrm{speed}(V_2) = Interpolatingspeed\ (F) \times \frac{Sub - axis1movingamount\ (S_2)}{Interpolatingmovingamount(S)}$$

$$Axis10\ \mathrm{speed}(V_{10}) = Interpolatingspeed\ (F) \times \frac{Sub - axis2\ movingamount\ (S_{10})}{Interpolatingmovingamount\ (S)}$$

(5) Relevant motion function block

Name	Description	Operation Condition
MC_MoveLinearAbsolute	Absolute positioning linear interpolation operation	Edge



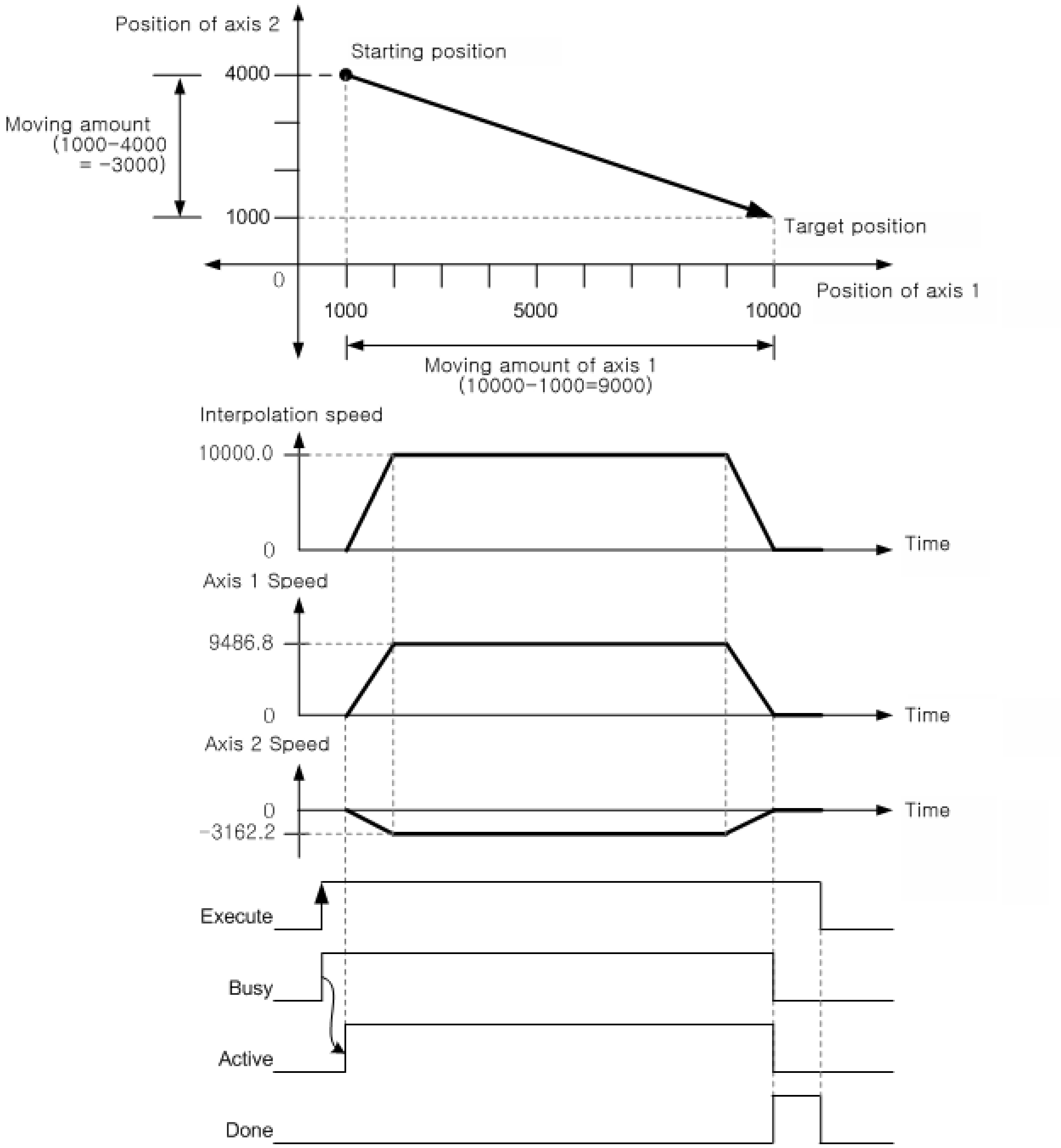
(6) Restrictions

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)

(7) Operation Timing

- Starting position: (1000.0, 4000.0)
- Target position: (10000.0, 1000.0)
- Target velocity: 10000.0



※ Velocity of each configuration axis is approximate estimate.

2. Linear interpolation control with relative coordinates

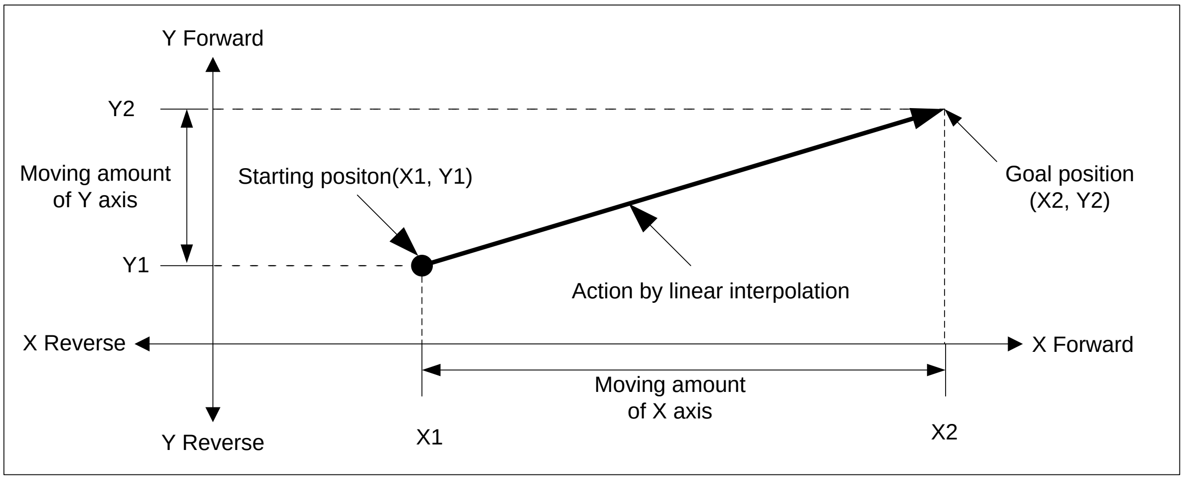
(「Relative positioning interpolation operation (MC_MoveLinearRelative)」)

(1) Linear interpolation is executed from starting position to movement direction targeted by each axis and position that includes movement direction. Positioning control is based on the current stop position.

(2) Movement direction is determined by the sign set in the target position (movement distance) of each axis.

■ 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)



(3) Interpolation that is currently being executed is stopped with group halt (MC_GroupHalt) or group immediate stop (MC_GroupStop) motion function block.

(4) The speed value set in relative position liner interpolation operation (MC_MoveLinearRelative) motion function block means interpolation speed.

The operation speed of each configuration axis is calculated as follows.

$$Interpolatinspeed\ (F) = Operationspeedsetinpositiondata$$

$$Interpolatingmovingamount(S) = \sqrt{S_1^2 + S_2^2 + \ldots + S_{10}^2}$$

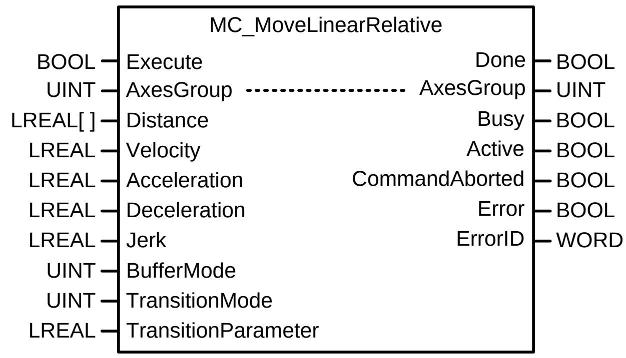
$$Axis1speed(V_1) = Interpolatingspeed\ (F) \times \frac{Main\,axismovingamount(S_1)}{Interpolatingmovingamount\ (S)}$$

$$Axis2\ \mathrm{speed}(V_2) = Interpolatingspeed\ (F) \times \frac{Sub-axis1movingamount\ (S_2)}{Interpolatingmovingamount(S)}$$

$$Axis10\ \mathrm{speed}(V_{10}) = Interpolatingspeed\ (F) \times \frac{Sub-axis2\ movingamount\ (S_{10})}{Interpolatingmovingamount\ (S)}$$

(5) Relevant motion function block

Name	Description	Operation Condition
MC_MoveLinearRelative	Relative positioning linear interpolation operation	Edge



(6) 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.

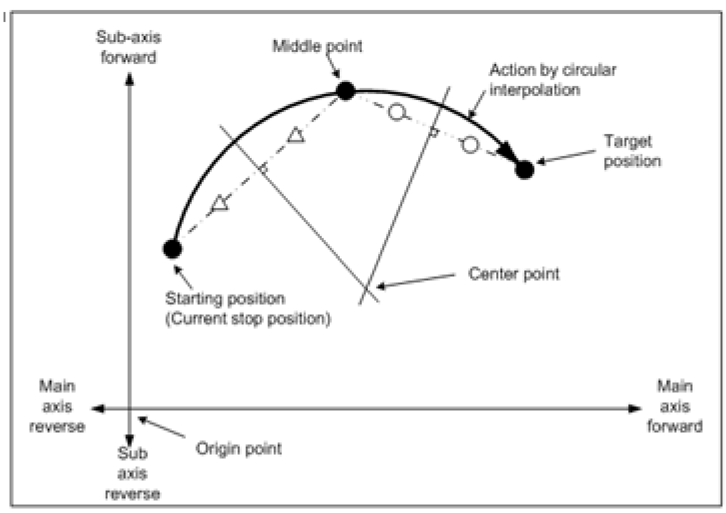
The interpolation that is currently being executed is stopped with group halt (MC_GroupHalt) or group immediate stop (MC_GroupStop) motion function block.

1. Circular interpolation using midpoint specification method

(1) Circular interpolation is executed from starting position to target position through midpoint position set in auxiliary point.

(2) The trajectory of the arc that takes an intersecting point caused by the vertical bisection of starting position and midpoint position, and midpoint position and target position is created.

(3) Movement direction is automatically determined in accordance with the set target position and auxiliary point of circular interpolation.



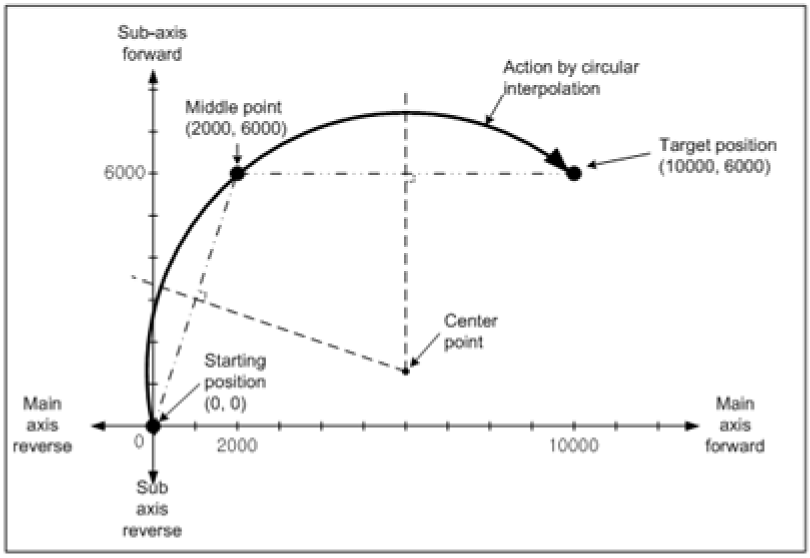
(4) Restrictions

Circular interpolation by midpoint specification method cannot be executed in the following cases.

- 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 starting 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)

(5) Operation pattern

- Starting positon: (0.0, 0.0)
- Target position: (10000.0, 6000.0)
- Middle point: (2000.0, 6000.0)
- Method(CircMode): Middle point(0)
- Direction(PathChoice): - (Ignored in middle point method)



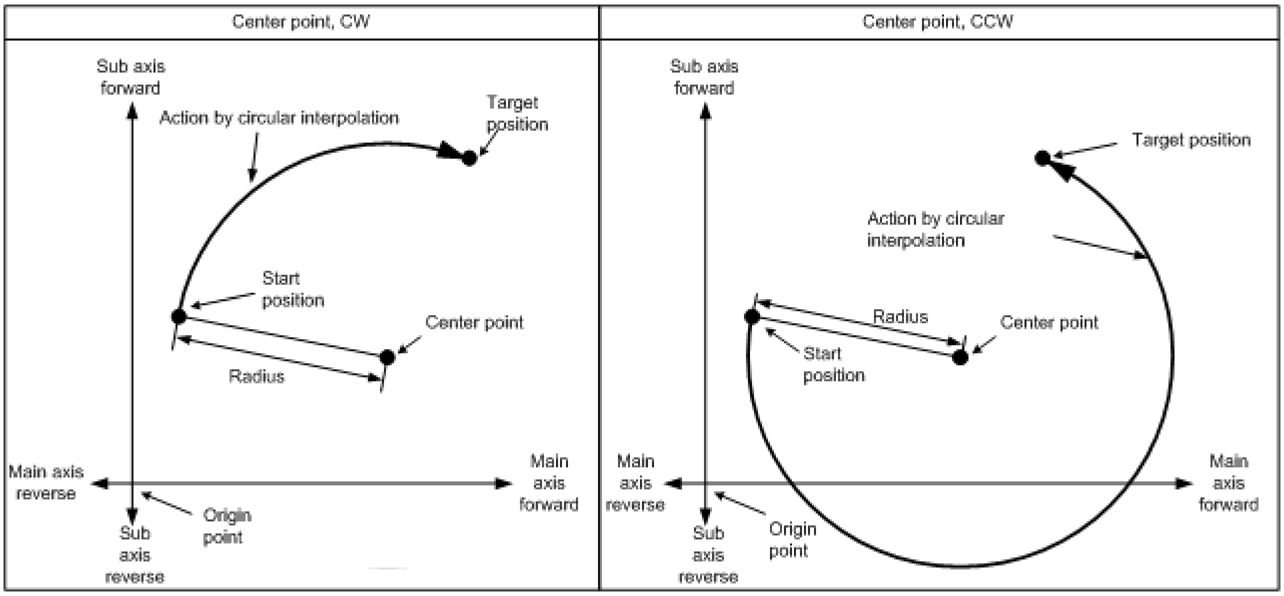
2. Circular interpolation using center point specification method

(1) Circular interpolation is performed from starting position to target position along the trajectory of the arc that takes the distance to the specified center point position as radius.

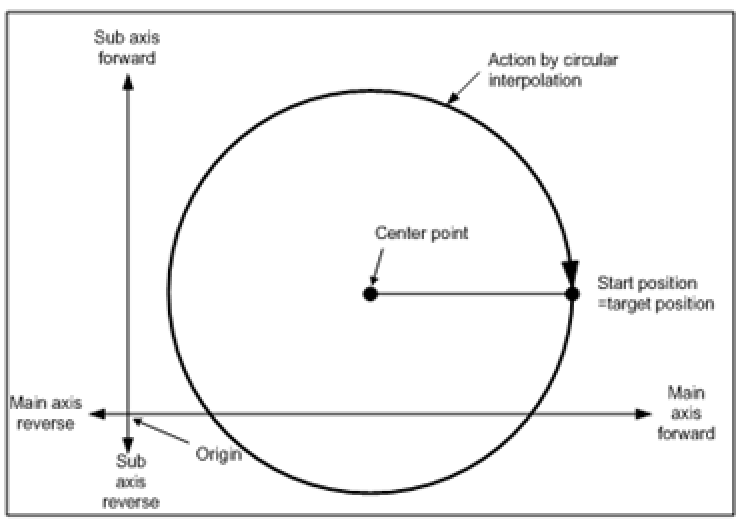
(2) Movement direction is determined by the direction set in “PathChoice” of absolute position circular interpolation operation (MC_MoveCircularAbsolute) or relative position circular interpolation operation (MC_MoveCircularRelative) motion function block.

▪ 0: 「CW」 - Circular interpolation is executed from the starting position in a clockwise direction.

▪ 1: 「CCW」 - Circular interpolation is executed from the starting position in a counterclockwise direction.



(3) If target position is same as start position, you can execute circular interpolation whose circle radius is distance from center point to starting position.



(4) Restrictions

Circular interpolation by center point specification method cannot be executed in the following cases.

• 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 starting position, auxiliary point position and target position are in a straight line (error code: 0x20A7)

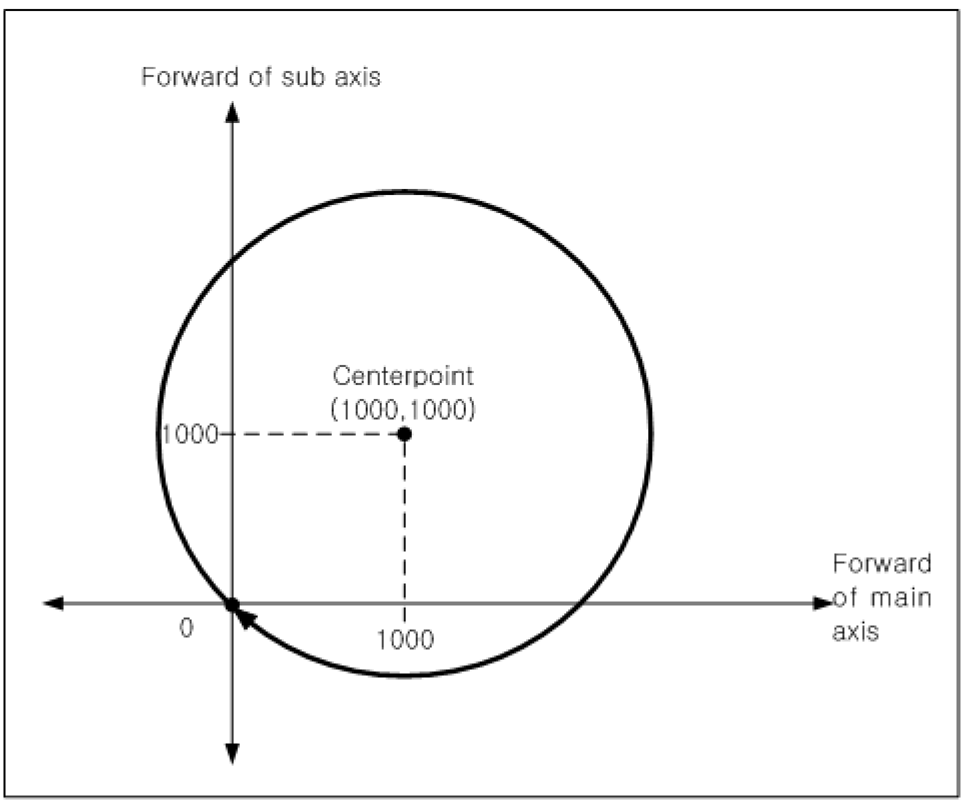
• In case there is an axis in infinite running repetition operation among configuration axes (error code: 0x20A8)

• 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)

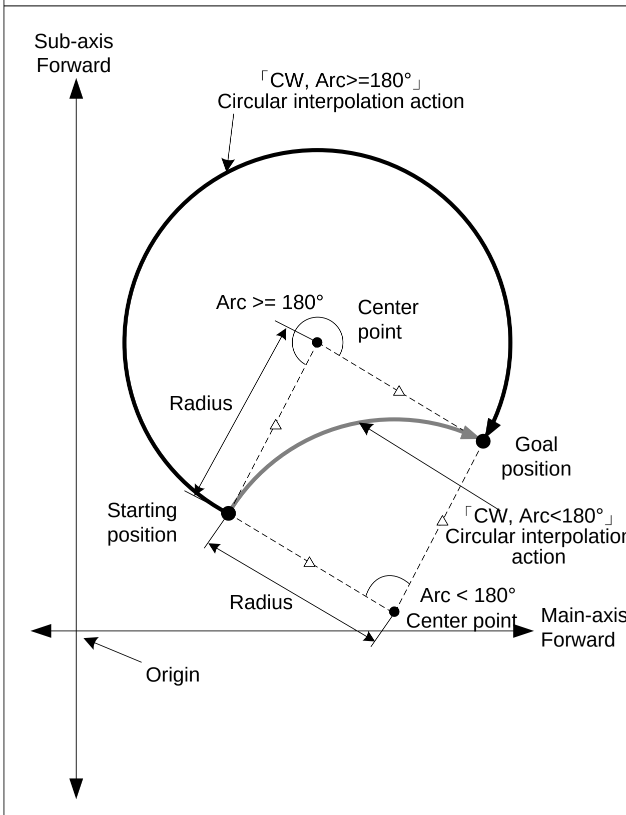
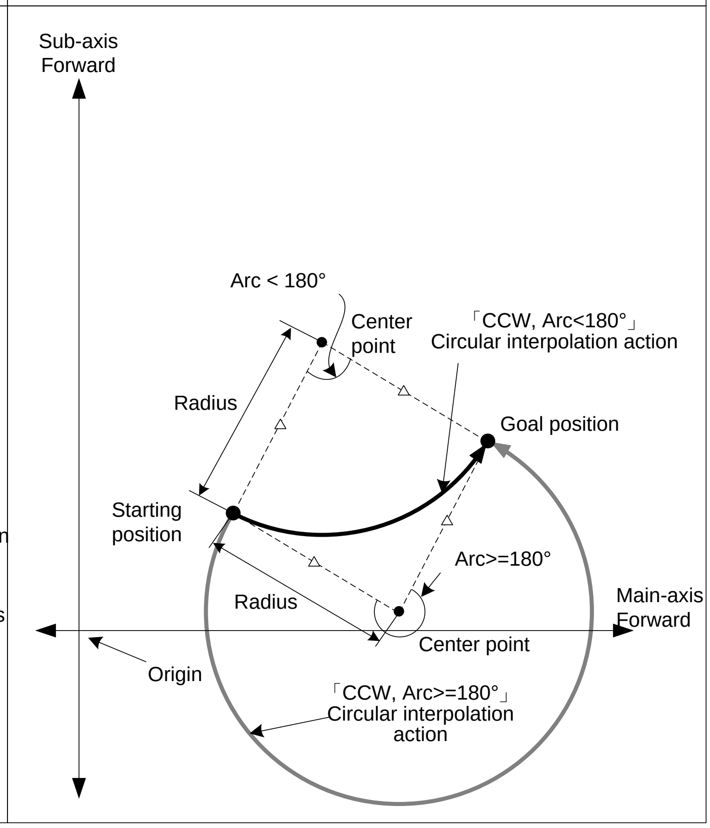
(5) Operation pattern

- Starting position: (0.0, 0.0)
- Target position: (0.0, 0.0)
- Serve position: (1000.0, 1000.0)
- Method(CircMode): Center point(1)
- Direction(PathChoice): CW(0)



3. Circular interpolation using radius specification method

(1) 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.

CW, Arc<180° CW, Arc>=180°	CCW, Arc<180° CCW, Arc>=180°
	

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

(3) Movement direction and the size of the arc is determined by the sign of auxiliary point and directions (CW, C CW) set in "PathChoice" of absolute position circular interpolation operation (MC_MoveCircularAbsolute) or relativ e position interpolation operation (MC_MoveCircularRelative) motion function block

(4) Restrictions

Circular interpolation by radius specification method cannot be executed in the following cases.

• 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 starting position, auxiliary point position and target position are in a straight line (error code: 0x20A7)

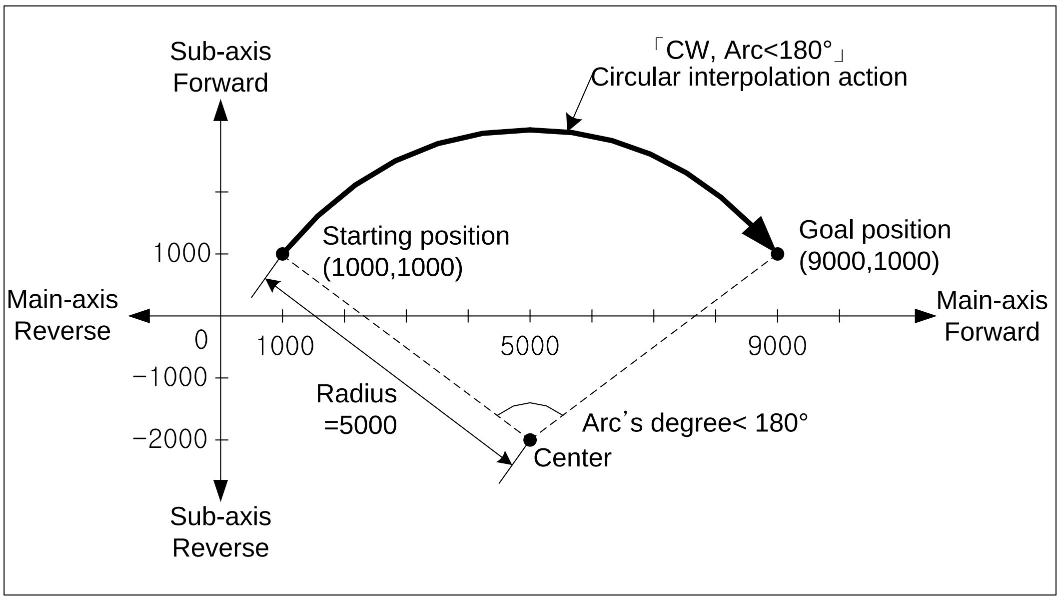
• In case there is an axis in infinite running repetition operation among configuration axes (error code: 0x20A8)

• 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)

(5) Operation patterns

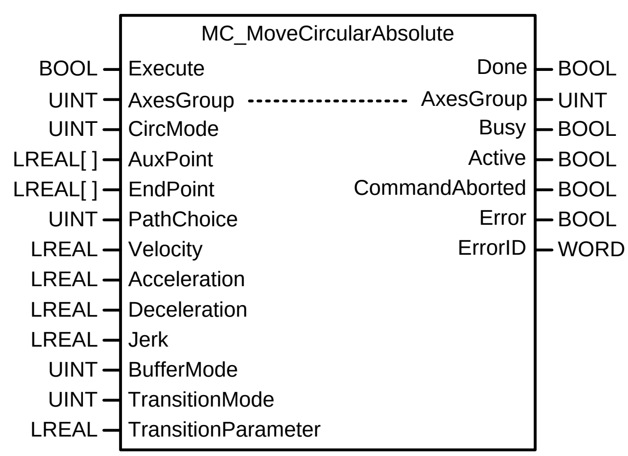
- Starting position: (1000.0, 1000.0)
- Target position: (9000.0, 1000.0)
- Serve position: (5000.0, 0.0)
- Method(CircMode): Radius(2)
- Direction(PathChoice): CW(0)



4. Relevant motion function block

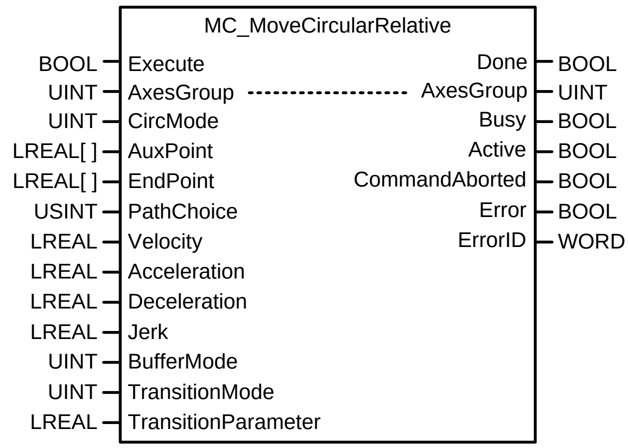
(1) Absolute positioning circular interpolation operation

Name	Description	Operation Condition
MC_MoveCircularAbsolute	Absolute positioning circular interpolation operation	Edge



(2) Relative positioning circular interpolation operation

Name	Description	Operation Condition
MC_MoveCircularRelative	Relative positioning circular interpolation operation	Edge

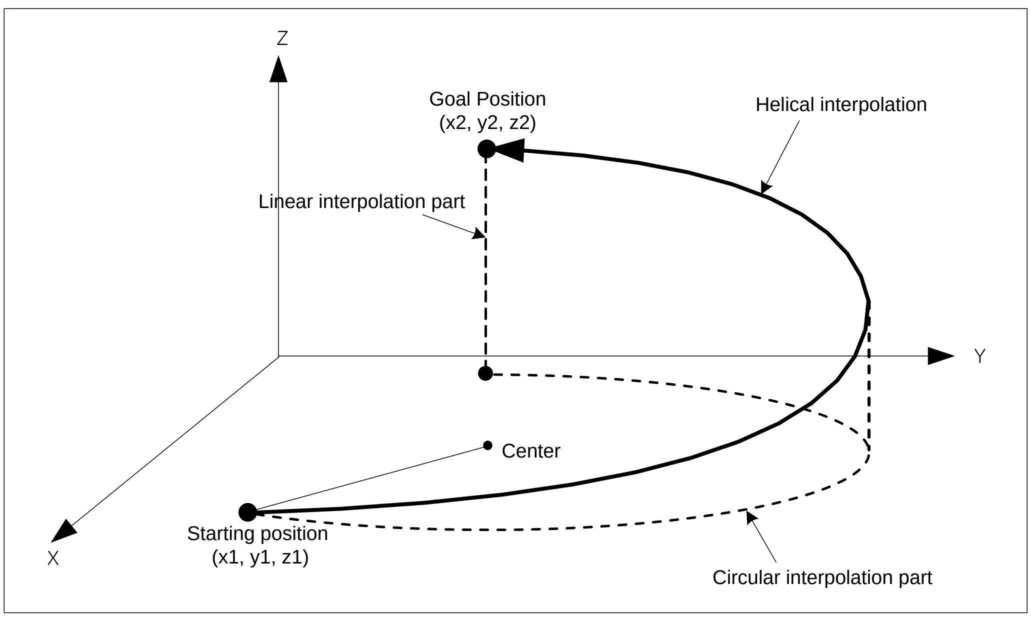


5. Helical interpolation

(1) 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.

(2) Linear axis is the third axis of the circular interpolation axis group.

(3) 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'.

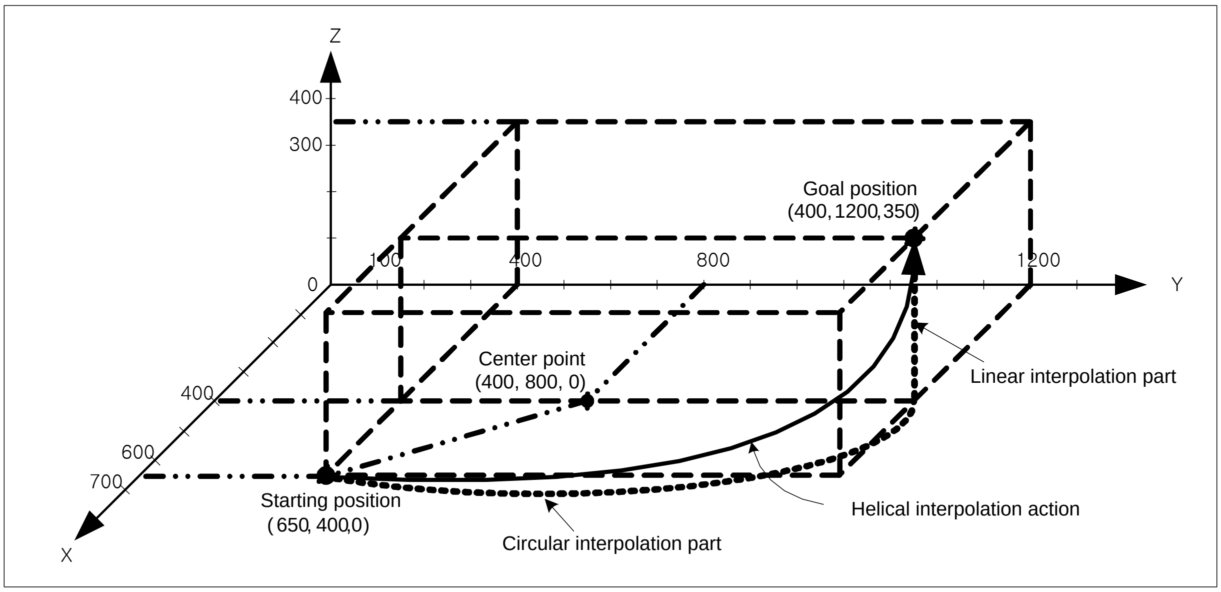


(4) Restrictions

The restrictions of helical interpolation are the same as those of circular interpolation according to the set circular interpolation modes.

(5) Operation pattern

- Starting position: (650.0, 400.0, 0)
- Target position: (400.0, 1200.350)
- Center position: (400.0, 800.0, 0)
- Method(CircMode): Center point(1)
- Direction(PathChoice): CCW(1)



8.2.9 Axis Control Buffer mode

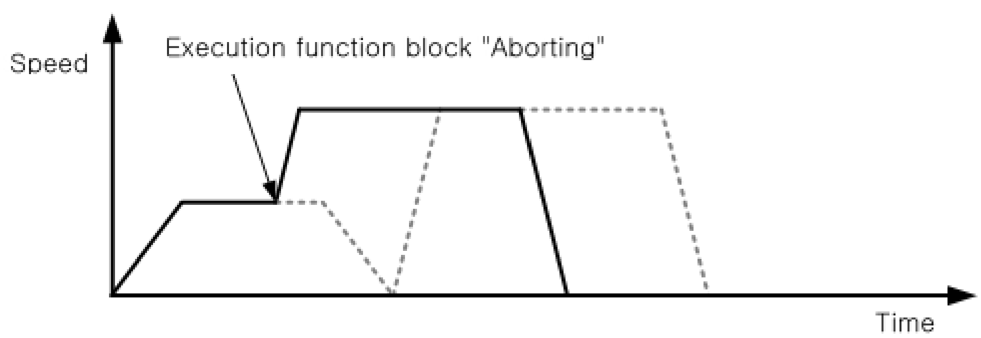
Cancellation of the existing axis motions and continued or continuous operation of them can be carried out by executing other motion function block while the axis is in operation. The motions are specified by entering buffer mode (BufferMode) in motion function block. In axis control the maximum number of runs that can be queued in the buffer is 10. In case of executing commands with buffer mode which has more than that, error (error code: 0x1022) occurs.

Values that can be set in Buffer Mode are as follows.

Buffer Mode	Description
Aborting	It executes commands immediately. The existing commands in operation are aborted.
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 higher velocity by comparing the velocity of the existing command and the command to make.

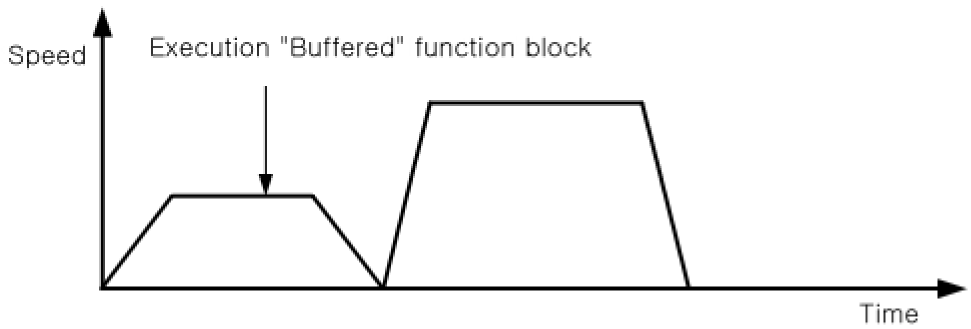
1. **Buffer Mode "Aborting"**

It aborts the existing commands in execution immediately and executes the next command. CommandAborted output of the existing motion function blocks is On.



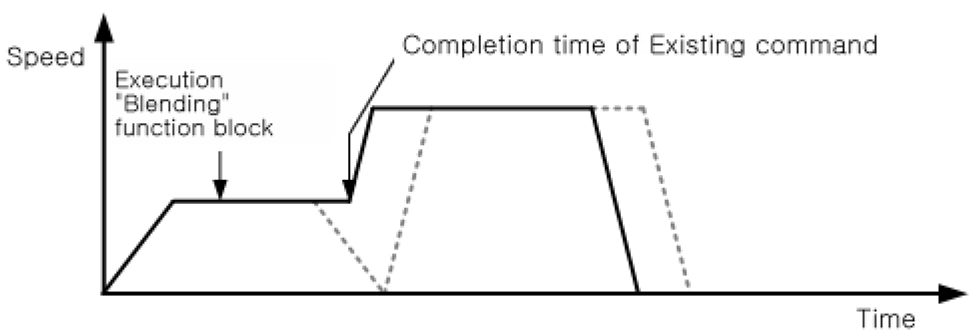
2. **Buffer Mode "Buffered"**

It execute the next command after the completion of the existing commands in execution (Done output is On).



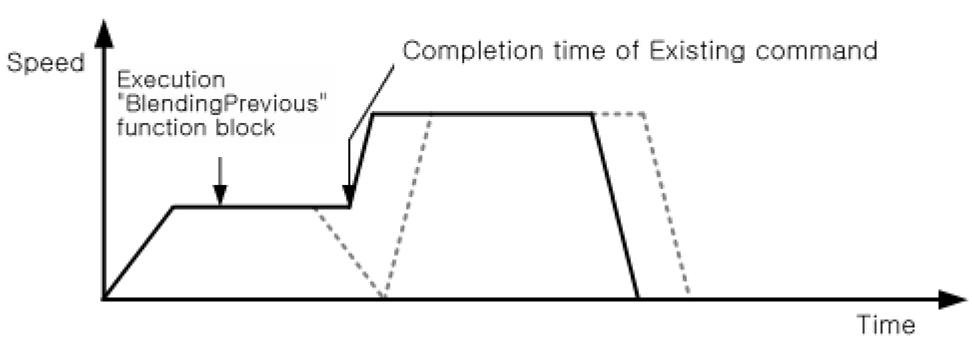
3. **Buffer Mode "BlendingLow"**

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.



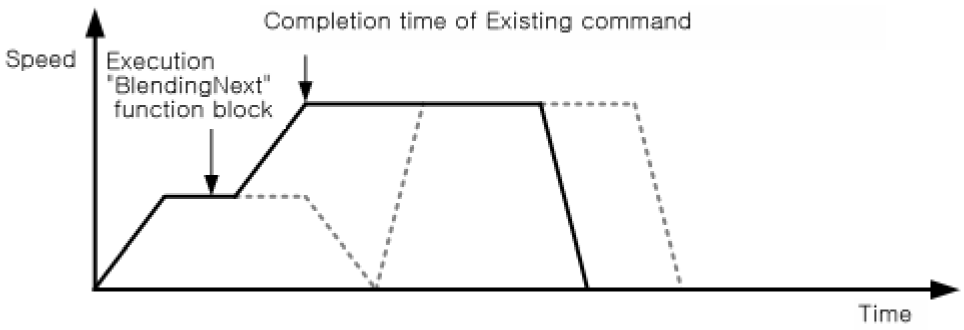
4. **Buffer Mode "BlendingPrevious"**

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.



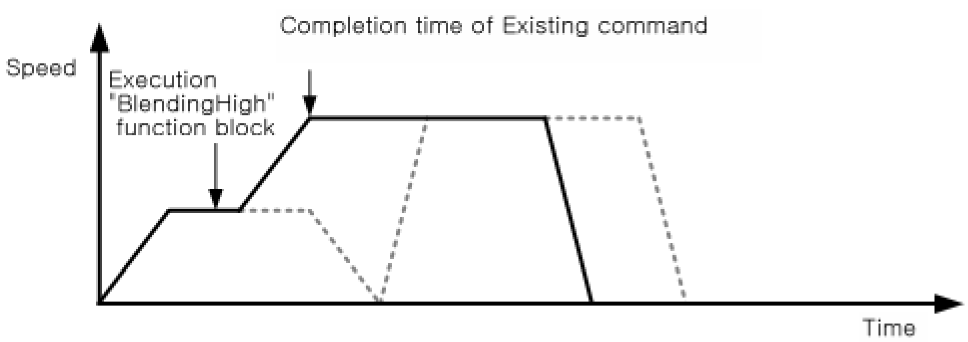
5. **Buffer Mode "BlendingNext"**

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

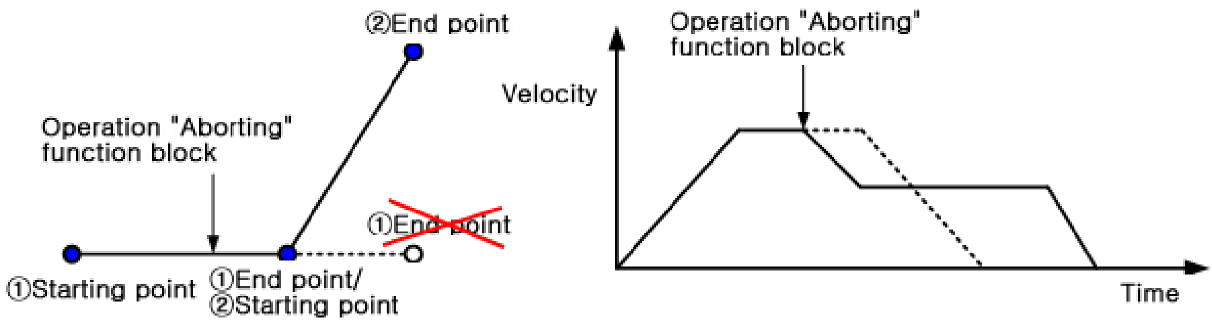
In axis group control as in speed control, motion commands can be executed continuously by using buffer mode, and the maximum number of runs that can be queued in the buffer is 10. In case of executing commands with buffer mode which is more than that, error (error code: 0x2022) occurs.

In addition, operation is possible by inserting curve between the two linear trajectories using transition mode.

1. 'BufferMode'

(1) Aborting

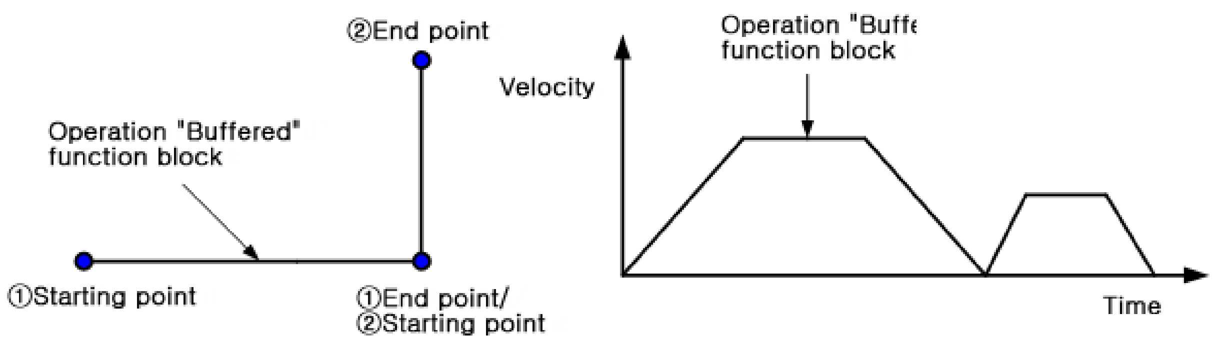
It aborts the motion that is currently running, and executes a new motion immediately.



(2) Buffered

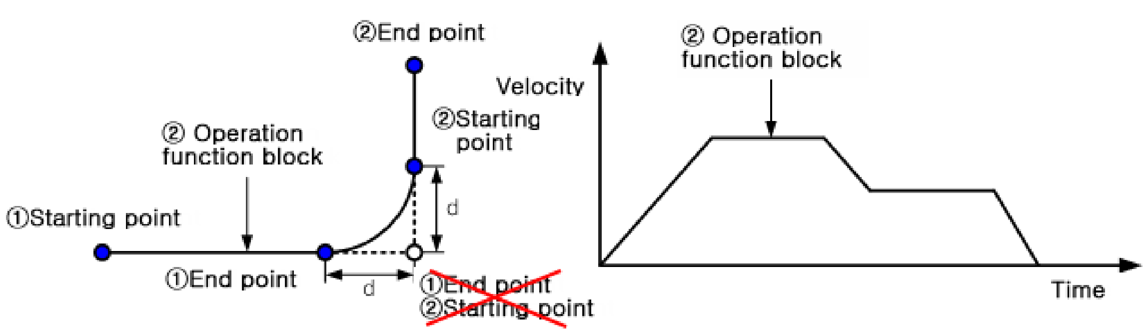
It executes the next command after completing motion operation that is being currently executed.

'TransitionMode' is not reflected.



(3) 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

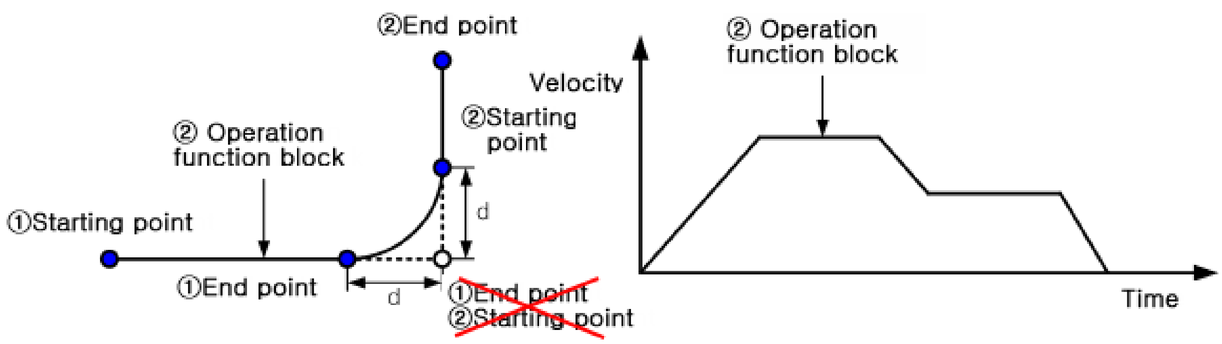
2. ‘TransitionMode’

(1) 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.

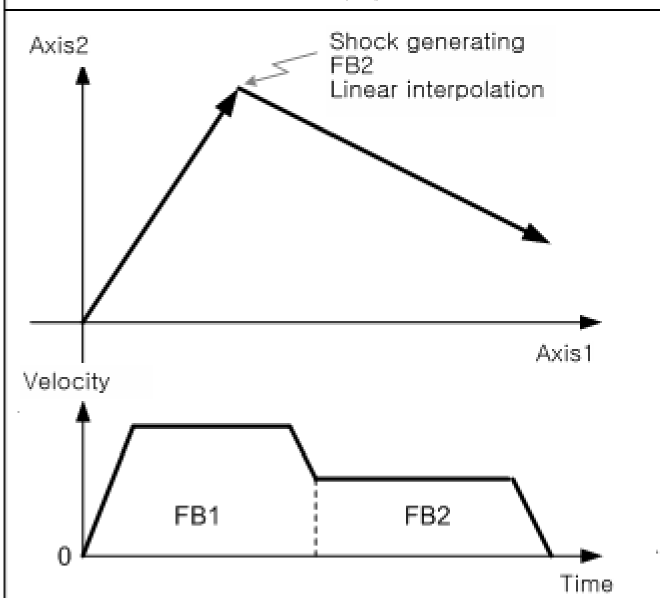
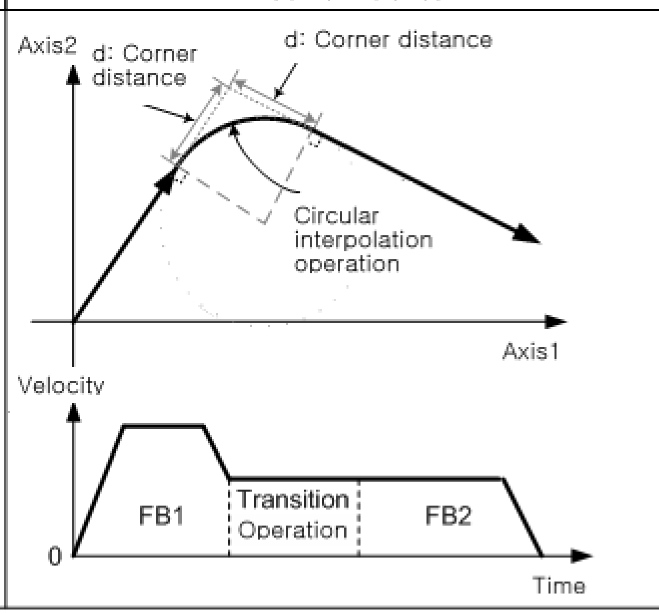
(2) TMCornerDistance

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

(3) TransitionMode Comparison

TMNone	TMCornerDistance
	

8.2.11 Synchronous Control

1. Gear operation

(1) Gear operation makes speed synchronization of main axis (or encoder) and serve axis depending on the set ratio.

(2) Gear operation can be aborted with gear operation cancellation command.

(3) Gear ratio (=velocity synchronization ratio) is calculated as follows.

Gear ratio = Main axis ratio/Serve axis ratio

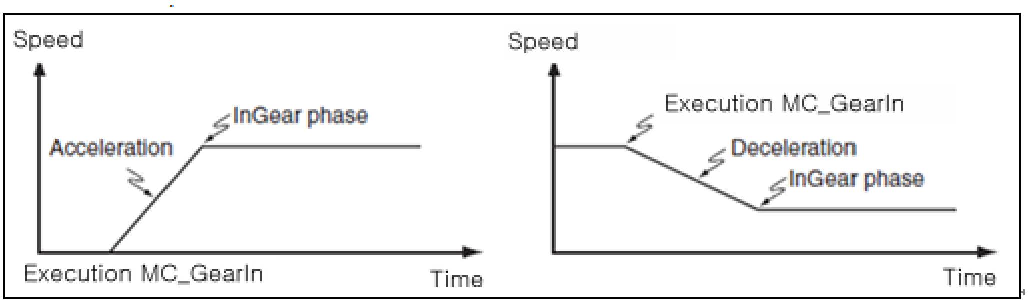
※ Main axis ratio < serve axis ratio can also be set.

(4) Rotation direction of serve axis is based on the forward direction of the main axis. In case gear ratio is positive (>0), rotation is made in forward direction, and that is negative (< 0), in reverse direction.

(5) The final operating velocity of serve axis is calculated as follows.

Operation speed of serve axis = Operation speed of main axis X Gear ratio = Operation speed of main axis X Main axis ratio/Serve axis ratio

(6) Acceleration/deceleration from the start of gear operation to target velocity can be set by using Acceleration and Deceleration input.

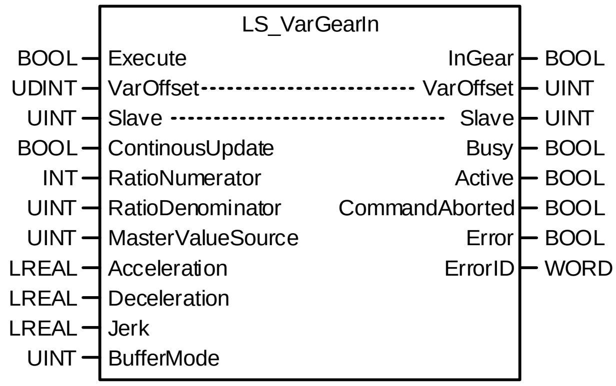


(7) Relevant motion function block

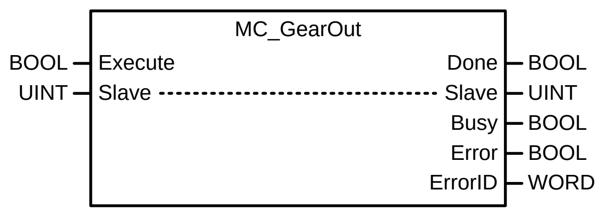
Name	Description	Operation Condition
MC_GearIn	Gearing run	Edge

```
                 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



Name	Description	Operation Condition
MC_GearOut	Gearing disengage	Edge

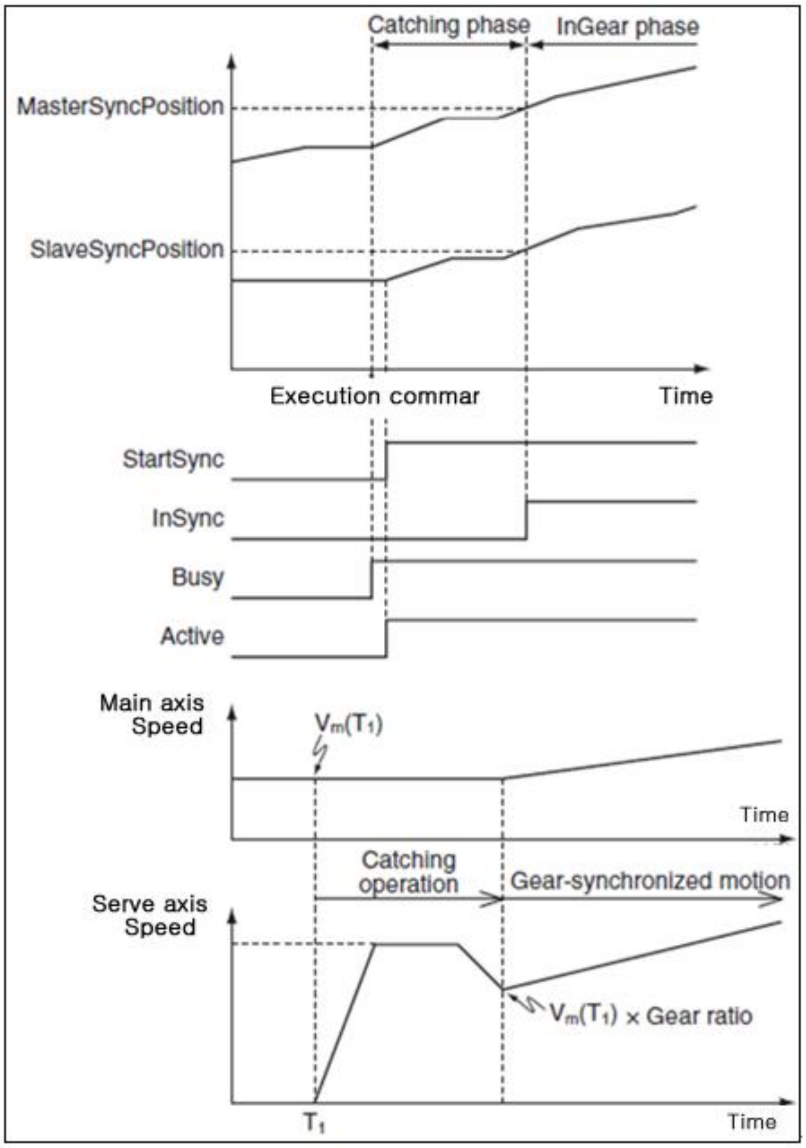


2. Positioning gear operation

(1) Positioning gear operation makes speed synchronization of main axis (or encoder) and serve axis depending on the ratio set the same as in gear operation basically.

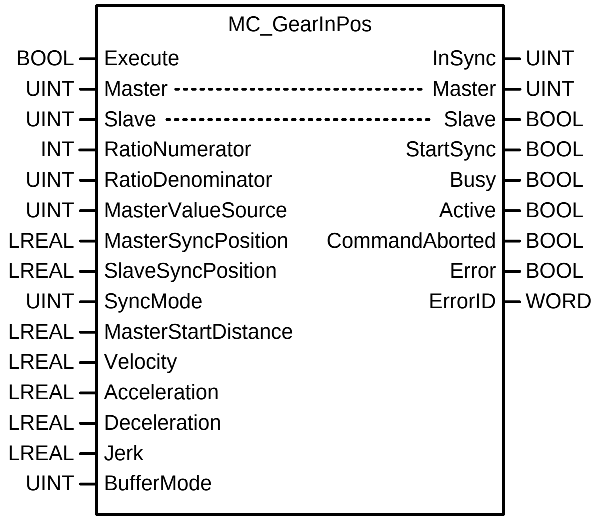
(2) The starting position in which main axis and serve axis are synchronized can be specified.

(3) Methods for operation are as follows.

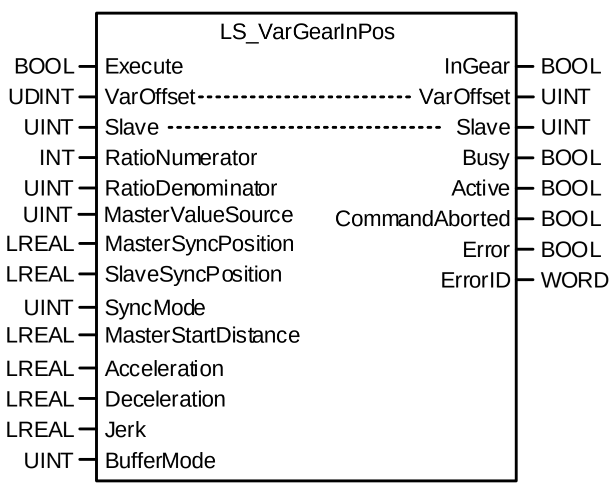

Catching phase
InGear phase
MasterSyncPosition
SlaveSyncPosition
Execution commar
Time
StartSync
InSync
Busy
Active
Main axis
Speed
$V_m(T_1)$
Time
Catching
operation
Gear-synchronized motion
Serve axis
Speed
$V_m(T_1)$ × Gear ratio
T_1
Time

(4) Relevant motion function block

Name	Description	Operation Condition
MC_GearInPos	Gearing by specifying the position	Edge



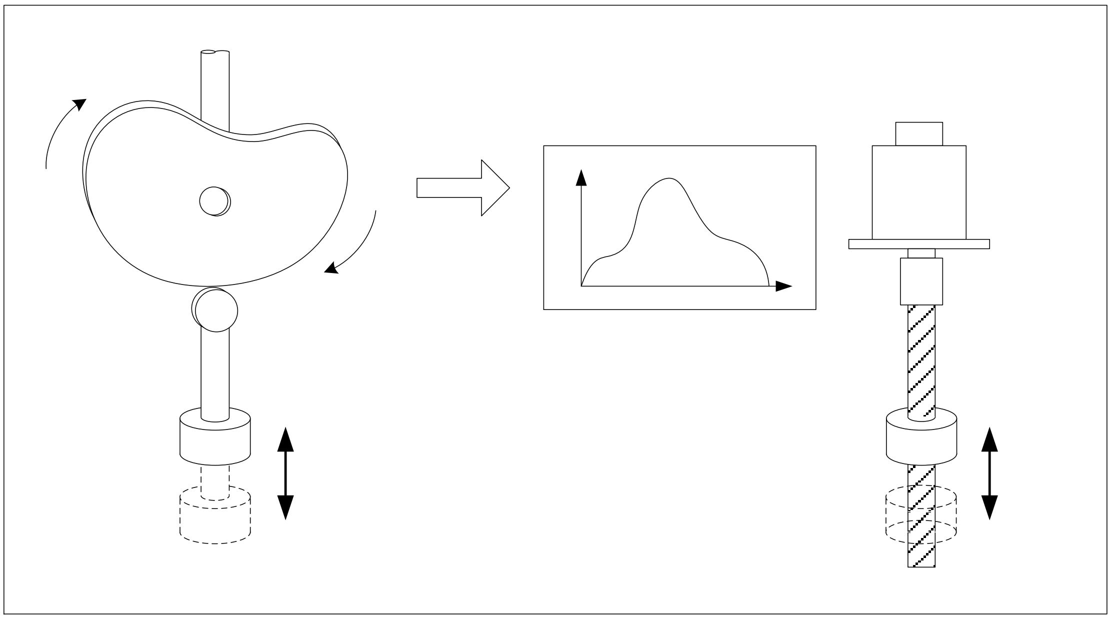
Name	Description	Operation Condition
LS_VarGearInPos	Variable Gearing by specifying the position	Edge



3. Cam operation

(1) 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.

(2) Mechanical cam operation in the past can be replaced with software cam motion using the cam data at the cam profiles.



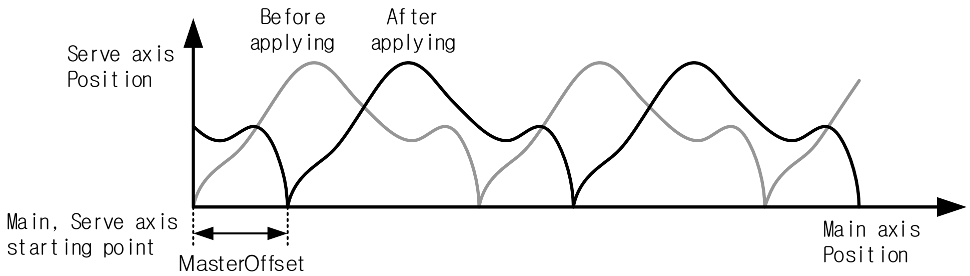
(3) A total of 32 cam profiles can be generated, each of which can be applied to each axis regardless of their order.
(4) Each cam profile consists of 100 cam data.
(5) To halt cam operation, MC_CamOut command should be issued on the sub-axis, or another motion function block should be operated (in case of Aborting).

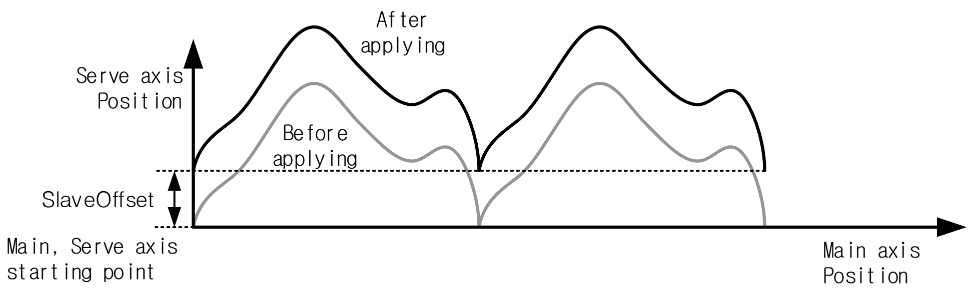
(6) Cam operation command's secondary data

Variable	Description
Master	Set the master Axis (1~32: Real Axis, 37~40: Virtual Axis, 41~42: Encoder),
MasterOffset	Set the master Axis offset value.
SlaveOffset	Set the offset value of the slave cam table.
MasterScaling	Specify the magnification of the main axis.
SlaveScaling	Specify the magnification of the serve axis cam table.
MasterStartDistance	Specify the position of the main-axis where the cam operation of the main-axis starts.
MasterSyncPosition	When cam operation starts, specify the start position at the cam table.
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 : Synchronized in the target value of the main axis. 1 : Synchronized in the current value of the serve axis.
CamTableID	Specify the cam table to operate.

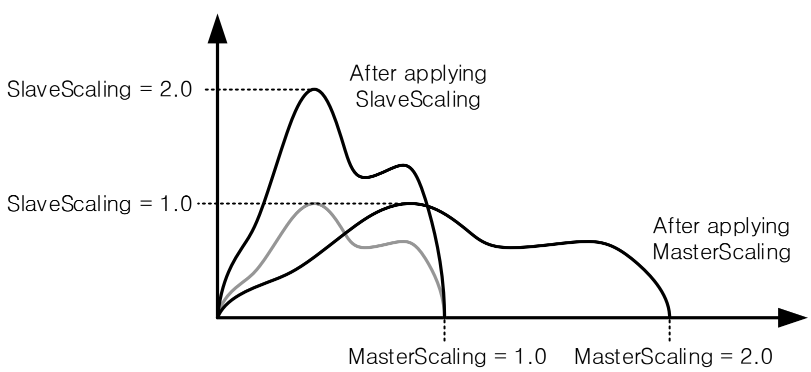
(a) At MasterOffset and SlaveOffset, set the cam table offset to apply. MasterOffset determines the offset from the main-axis start point, and Slaveoffset determines the offset from the sub-axis start point. Please refer to the figure below.

Using offset may change the start position for cam operation, causing an abrupt operation. In such a case, MasterSyncPosition, MasterStartDistance should be used.

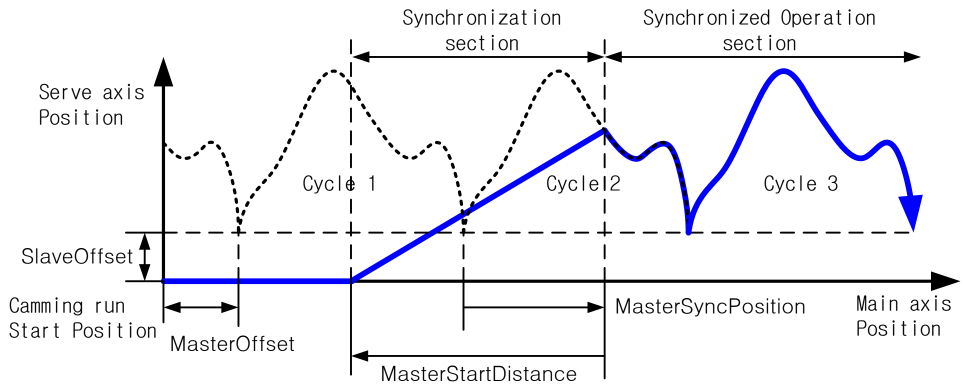




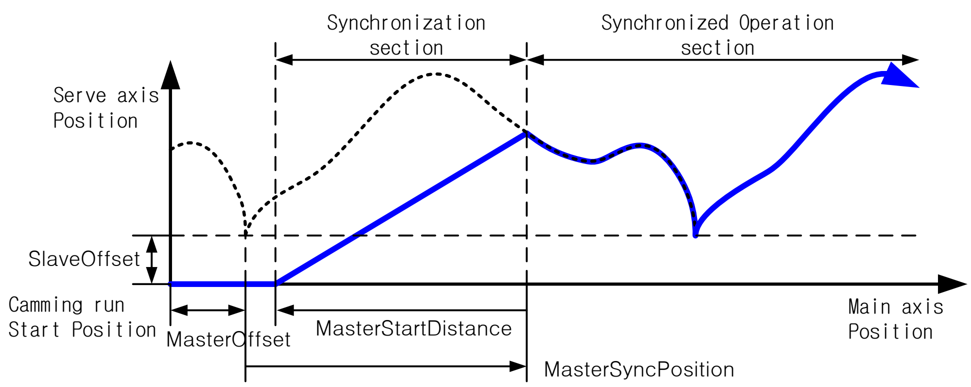
(b) At MasterScaling and SlaveScaling, set the scale rate of the cam data to apply. MasterScaling determines the scale rate of the main-axis data, and SlaveScaling determines the scale rate of the sub-axis data. Please refer to the figure below.



(c) MasterSyncPosition input specifies the position of the main axis within the table where the synchronization of actual cam operation is completed, and MasterStartDistance input specifies the relative position of the main axis where the synchronization starts. 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.



In case MasterScaling is 1.0



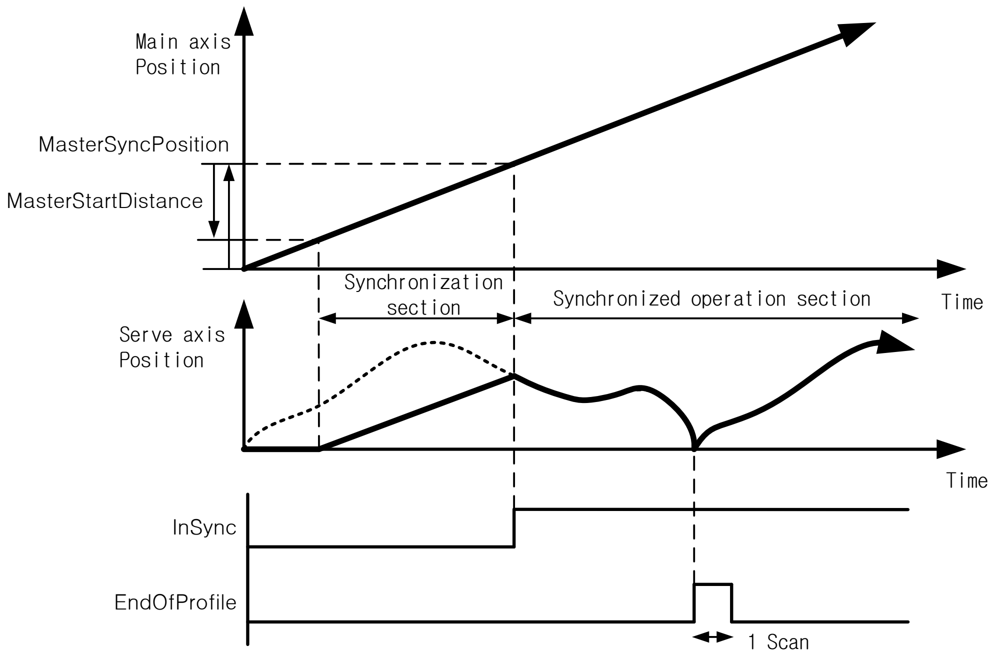
In case MasterScaling is 2.0

MasterSyncPosition position is based on the position within the cam table, and actual synchronization position is decided by considering MasterOffset and MasterScale parameters.

The serve axis starts moving to the synchronization position from the distance of the input value away based on the position where MasterSyncPosition is actually applied. If it is before starting moving, the serve axis waits at the relevant position in stop state, and if the serve axis is already in the section to move to the synchronization position at the beginning of the command, takes back the position of the synchronization starting point by the length of a table until it escapes the MasterStartDistance range.

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.

(d) InSyncoutput is on when cam operation starts the synchronization. 1 scan of EndOfProfileoutput is on whenever a single cam table operation is completed.



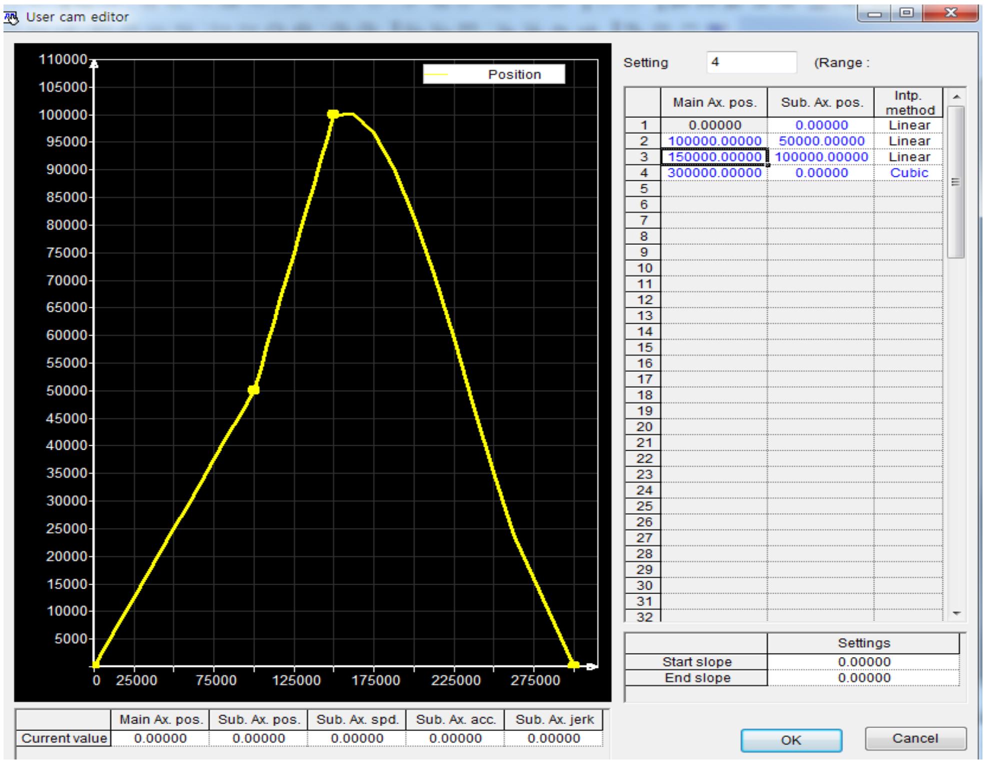
(e) At StartMode, the cam operation mode is set. The setting range is either 0 or 1. If the input value exceeds the setting range, an error occurs. If it is set to 0, the cam table start position is set to the main-axis position of 0. If it is set to 1, the cam table start position is set to the current position of the main-axis.

(f) MasterValueSource selects the source for the main-axis to synchronize. If set to 0, sub-axis performs the cam operation based on the main-axis command position calculated by the motion control module, and it set to 1, it performs the cam operation based on the current position received through the communication from from the main-axis servo drive.

(g) CamTableID sets the number of the cam table to be applied to the 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.

(7) Cam profile

<table>
<tr><th colspan="2">Parts</th><th>Description</th></tr>
<tr><td rowspan="3">CAM data</td><td>Main axis position</td><td rowspan="2">Set the sub-axis cam position corresponding to the main-axis</td></tr>
<tr><td>Slave axis position</td></tr>
<tr><td>Interpolation type</td><td>Set the characteristic curve between the cam data.
(Linear, Cubic)</td></tr>
<tr><td colspan="2">Start slope</td><td rowspan="2">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.</td></tr>
<tr><td colspan="2">End slope</td></tr>
</table>



(8) Motion function block

Name	Description	Operation condition
MC_CamIn	Cam operation	Edge

```
              MC_CamIn
BOOL  ─ Execute                      InSync ─ BOOL
UINT  ─ Master -------------------- Master ─ UINT
UINT  ─ Slave  --------------------- Slave ─ UINT
LREAL ─ ContinousUpdate                Busy ─ BOOL
LREAL ─ MasterOffset                 Active ─ BOOL
LREAL ─ SlaveOffset         CommandAborted ─ BOOL
LREAL ─ MasterScaling                 Error ─ BOOL
LREAL ─ SlaveScaling                ErrorID ─ WORD
LREAL ─ MasterStartDistance    EndOfProfile ─ BOOL
LREAL ─ MasterSyncPosition
UINT  ─ StartMode
UINT  ─ MasterValueSource
UINT  ─ CamTableID
UINT  ─ BufferMode
```

Name	Description	Operation condition
LS_VarCamIn	Variable Cam operation	Edge

LS_VarCamIn

Input type	Input	Output	Output type
BOOL	Execute	InSync	BOOL
UDINT	VarOffset	VarOffset	UINT
UINT	Slave	Slave	UINT
LREAL	ContinousUpdate	Busy	BOOL
LREAL	MasterOffset	Active	BOOL
LREAL	SlaveOffset	CommandAborted	BOOL
LREAL	MasterScaling	Error	BOOL
LREAL	SlaveScaling	ErrorID	WORD
LREAL	MasterStartDistance	EndOfProfile	BOOL
LREAL	MasterSyncPosition		
UINT	StartMode		
UINT	MasterValueSource		
UINT	CamTableID		
UINT	BufferMode		

Name	Description	Operation condition
MC_CamOut	Cam operation out	Edge

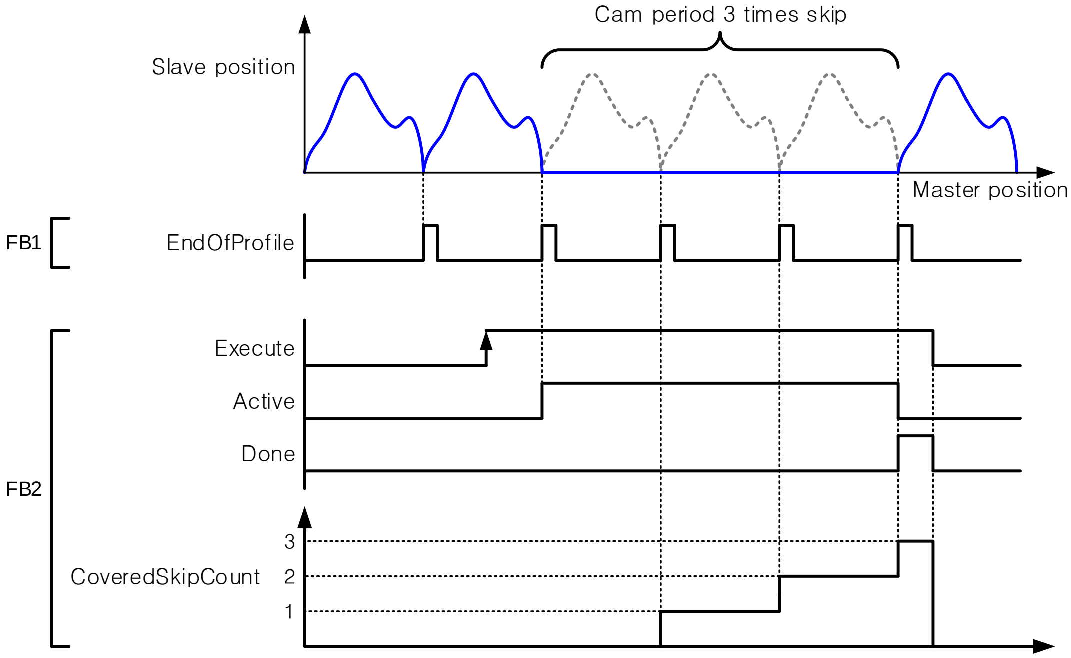
MC_CamOut

Input type	Input	Output	Output type
BOOL	Execute	Done	BOOL
UINT	Slave	Slave	UINT
		Busy	BOOL
		Error	BOOL
		ErrorID	WORD

4. Cam skip

(1) 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.

(2) 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.



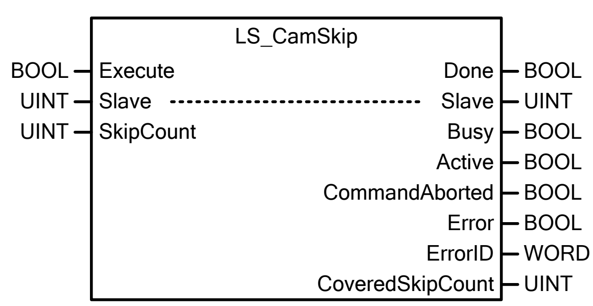
(3) After the execution of cam motion by MC_CamIn command (FB1), if three cycles are skipped using LS_CamSkip command (FB2), the output of each function block FB1 and FB2and the motion of the cam sub-axis are as displayed as shown in the figure below

(4) 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.

(5) Even if the execute contact point is off before the end of the executed cap skip motion, the active contact point is on until the operation is completed. Done and CoveredKipCount are only applied for the scans performed after completing the operation.

(6) Motion function block

Name	Description	Operation condition
LS_CamSkip	Cam skip	Edge



8.2.12 Manual Control

1. Jog operation

(1) Jog operation makes positioning control by manual jog commands of users.

(2) Jog operation is possible even in the state in which the origin of the axis is not determined.

(3) 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.

(4) Acceleration/deceleration processing 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.

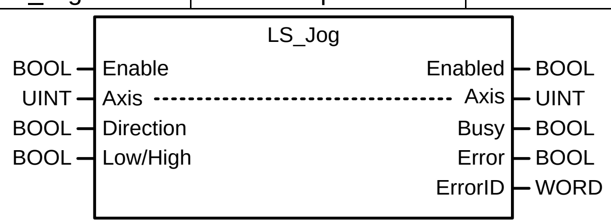
(5) Jog high-speed should be set to at the speed limit or less or at least Jog low-speed among [Operation parameter – basic parameter] setting items.

(6) Parameter setting

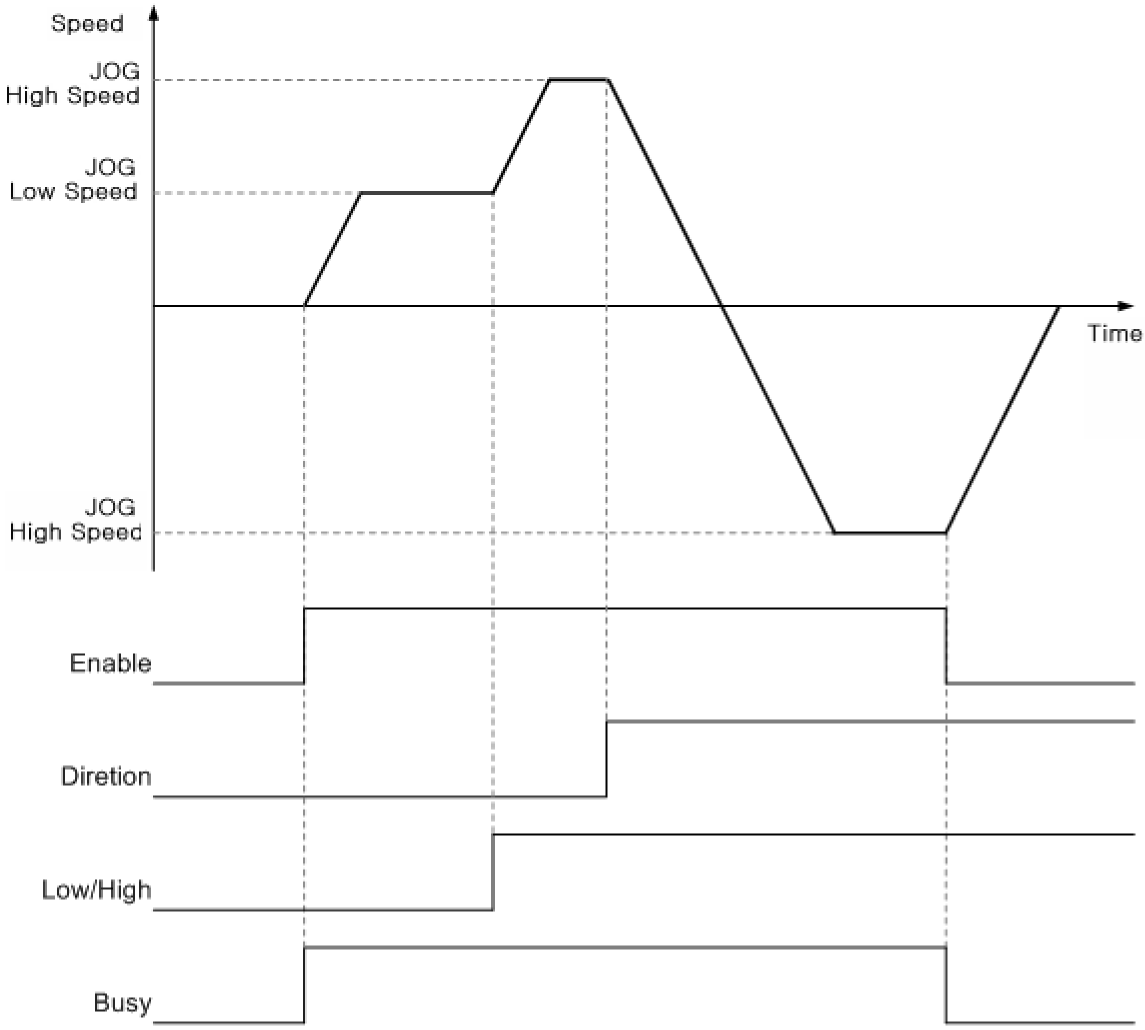
Item	Settings	Initial Value
JOG High Speed	Long Real(LREAL) Positive number	100000 pls/s
JOG Low Speed		10000 pls/s
JOG Acceleration	0 or Long Real(LREAL) Positive number	100000 pls/s^2
JOG Deceleration		100000 pls/s^2
JOG jerk		0 pls/s^3

(7) Motion function block

Name	Description	Operation Condition
MC_Jog	JOG operation	Level



(8) 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

(1) 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.

(2) The moving distance is determined depending on the signs of the target distance specified in the Distance input, and if the moving direction is positive (+ or no sign), it means forward movement, and if it is negative (-), it means reverse movement.

(3) The existing motion is not canceled, but its operation overlaps with SuperImposed operation.

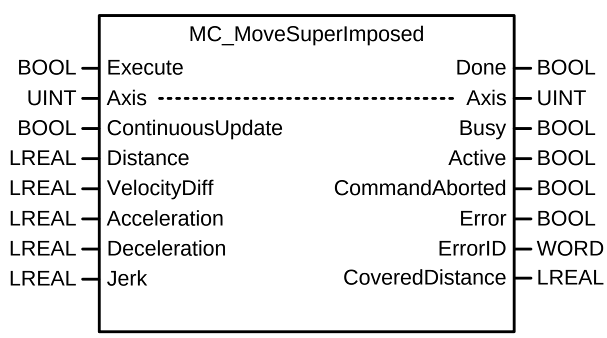
(4) Even when the existing motion is completed, SuperImposed operation continues unlessthe amount of movement does not reach the one specified in the SuperImposedoperation.

(5) If the axis is not in operation, but in "StandStill" status, SuperImposed operation works the same way as MC_MoveRelative operation.

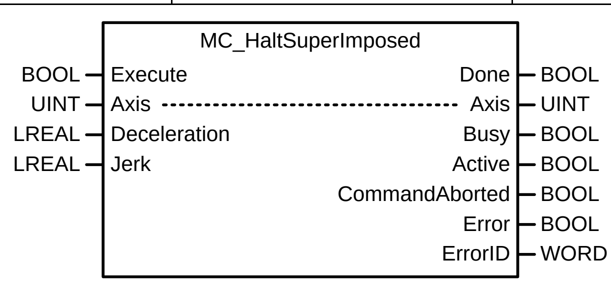
(6) The current SuperImposed operation can be halted with the MC_HaltSuperImposed command. After executing the command, SuperImposed operation decelerates and stops at the given acceleration and jerk. The existing motion which is currently being executed is not affected.

2. Motion function block

Name	Description	Operation Condition
MC_MoveSuperImposed	SuperImposed operation	Edge



Name	Description	Operation Condition
MC_HaltSuperImposed	SuperImposed operation halt	Edge



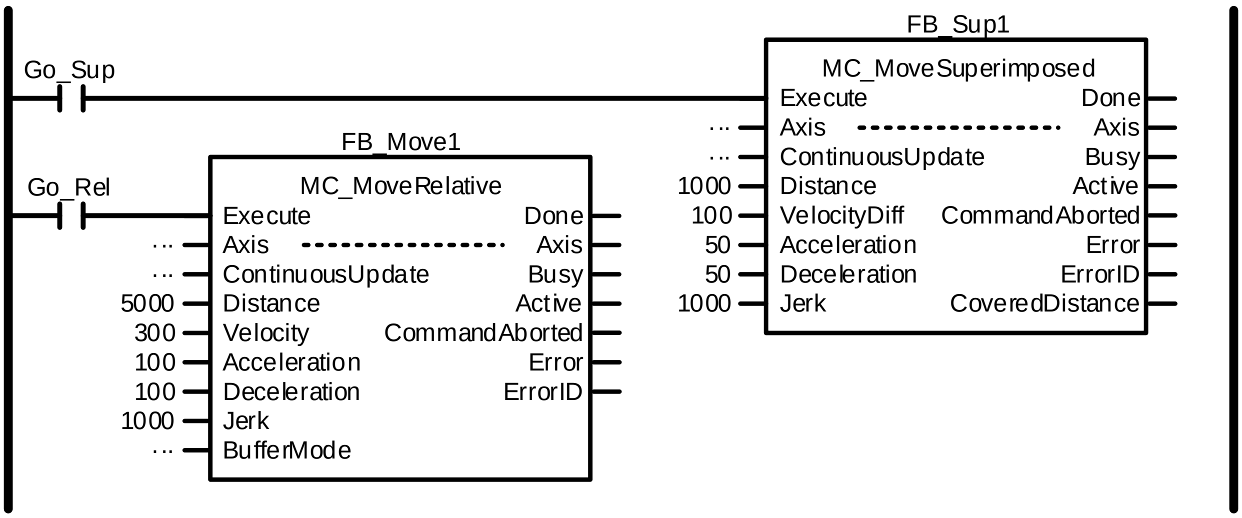
3. Limitation

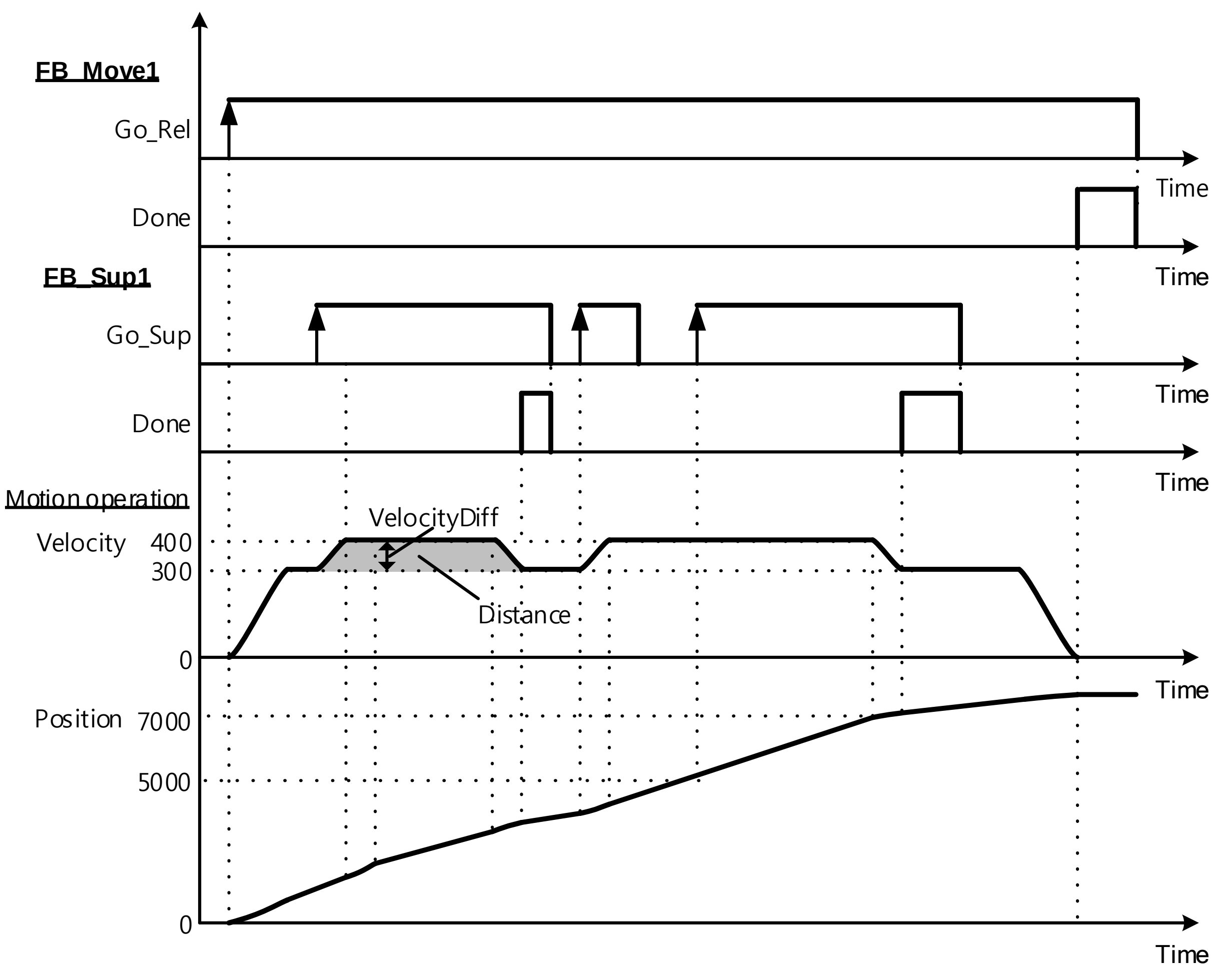
In the following cases, SuperImposed operation cannot be performed due to errors.

(1) SuperImposed command is executed during velocity control or torque control operation (Error Code: 0x1082)

(2) MC_HaltSuperImposed command is executed when SuperImposed operation is not being performed (Error Code: 0x1083)

4. Operation timing





8.2.14 Phase Correction 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. Control features

(1) Phase correction order can be executed with respect to the axes where synchronized operation is underway as in gear operation or cam operation.

(2) 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.

(3) If the main-axis is encoder, when phase correction control is executed, the operation uses the velocity limit of the sub-axis.

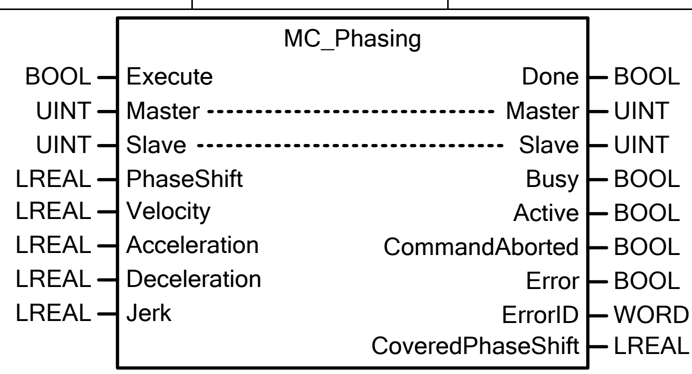
(4) The main-axis position which the sub-axis refers to during synchronized operation is "actual main-axis position + phase correction control position.”

(5) 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.

(6) If re-executing phase correction by setting PhaseShiftto 0 during the phase correction operation, the existing phase correction operation stops immediately.

2. Motion function block

Name	Description	Operation Condition
MC_Phasing	Phase correction	Edge



3. Limitation

In the following cases, Phase correction cannot be performed due to errors.

(1) Sub-axis is not performing synchronization control operation (Error Code: 0x1130)

(2) The designated main-axis is the main-axis of the actual synchronized operation (Error Code: 0x1131)

(3) PhaseShift is outside the pulse unit position expression range (INT) (Error Code: 0x1132)

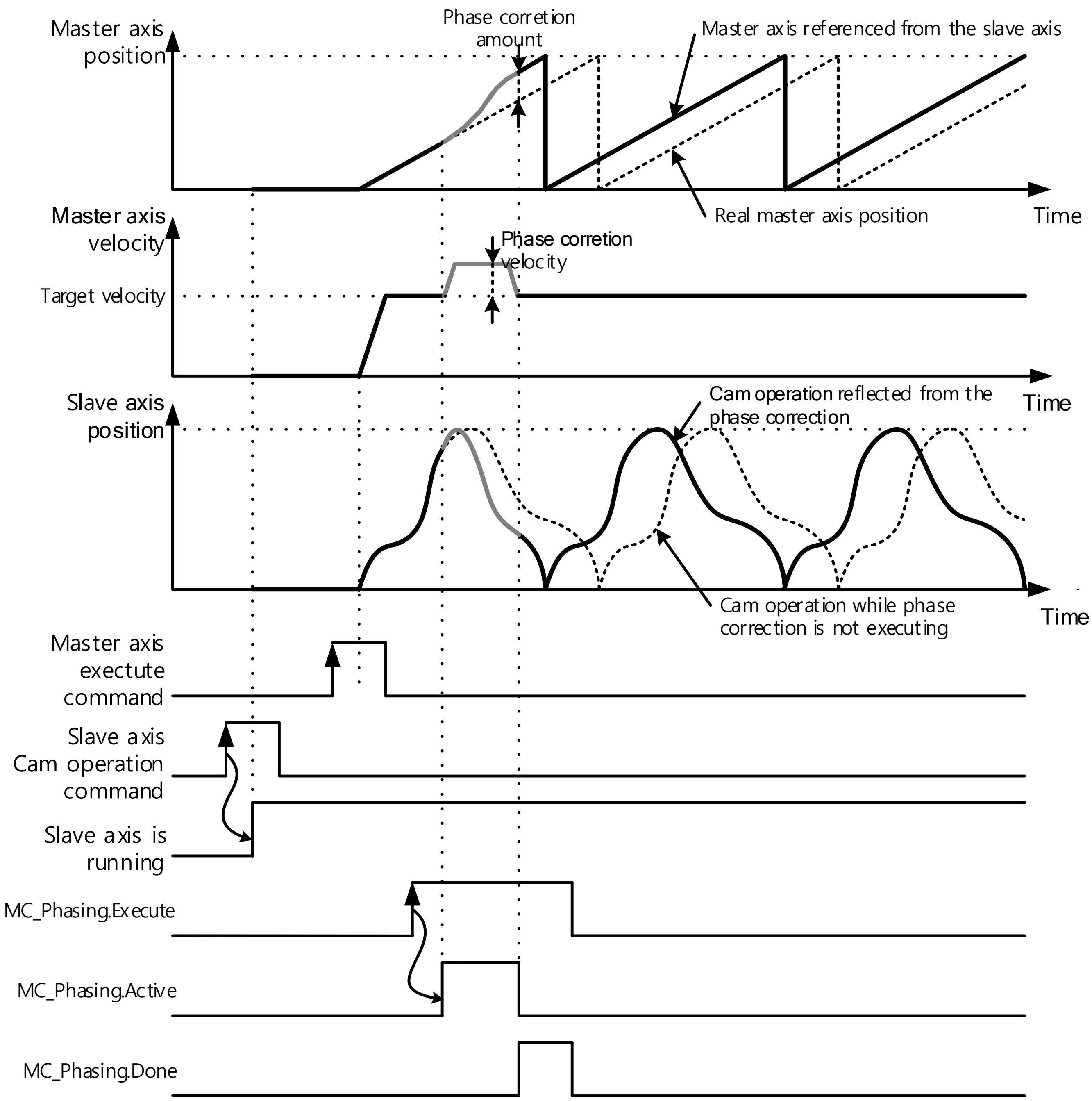
(4) Velocity setting is less than0, or exceeds the velocity limit for the main-axis (Error Code: 0x1133)

(5) Acceleration setting is less than 0 (Error Code: 0x1014)

(6) Deceleration setting is less than 0 (Error Code: 0x1015)

(7) Jerk setting is less than 0 (Error Code: 0x1016)

4. Operation timing



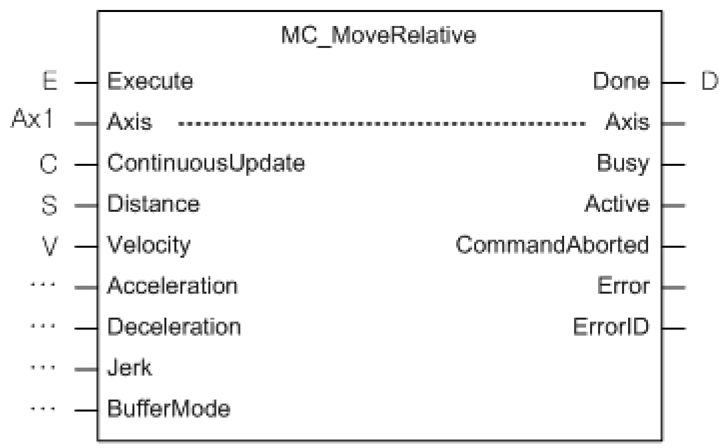
8.3 Other Functions

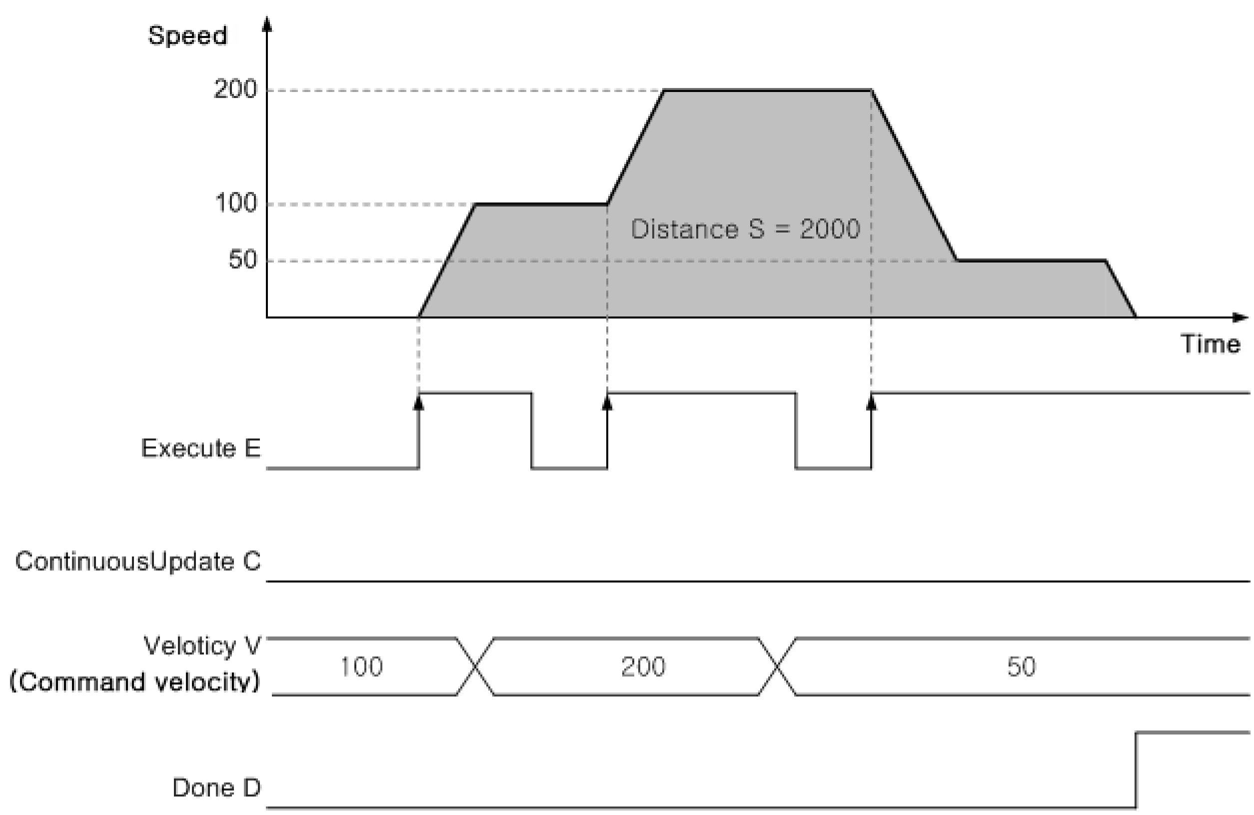
8.3.1 Functions to Change Control

1. Changes in input variables of motion function block in execution

(1) 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.

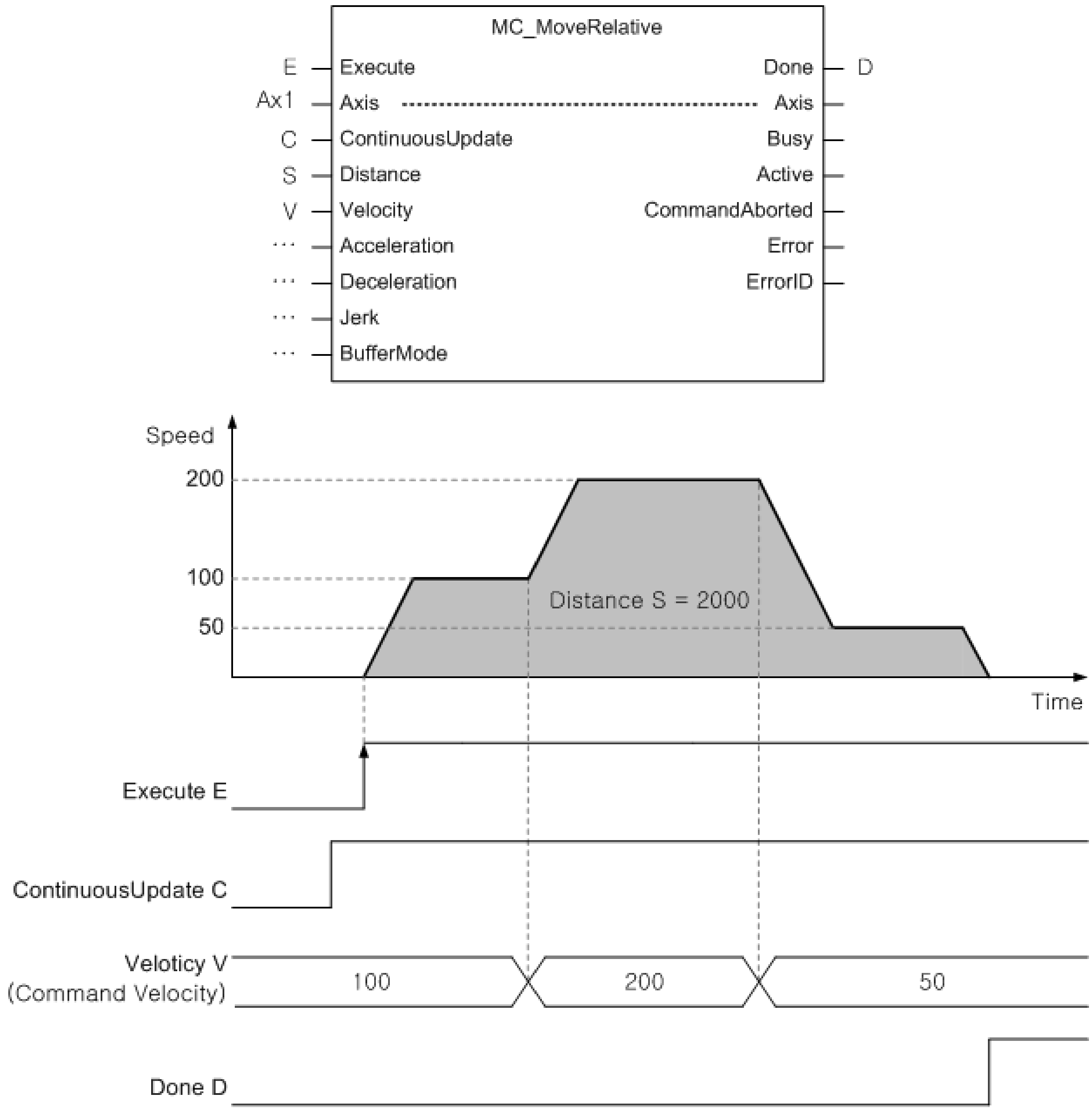
[Example] Input variable update of motion control command by re-execution of Execute





(2) 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, and 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



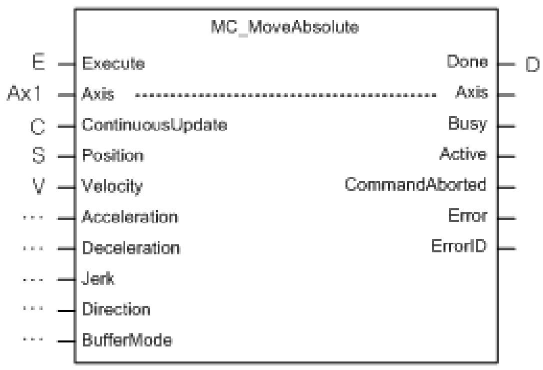
(3) In case of the level operation motion function block, input variables at the time when Enable input is On (rising Edge) are applied to operate, and input variables can be changed continuously while Enable input is On.

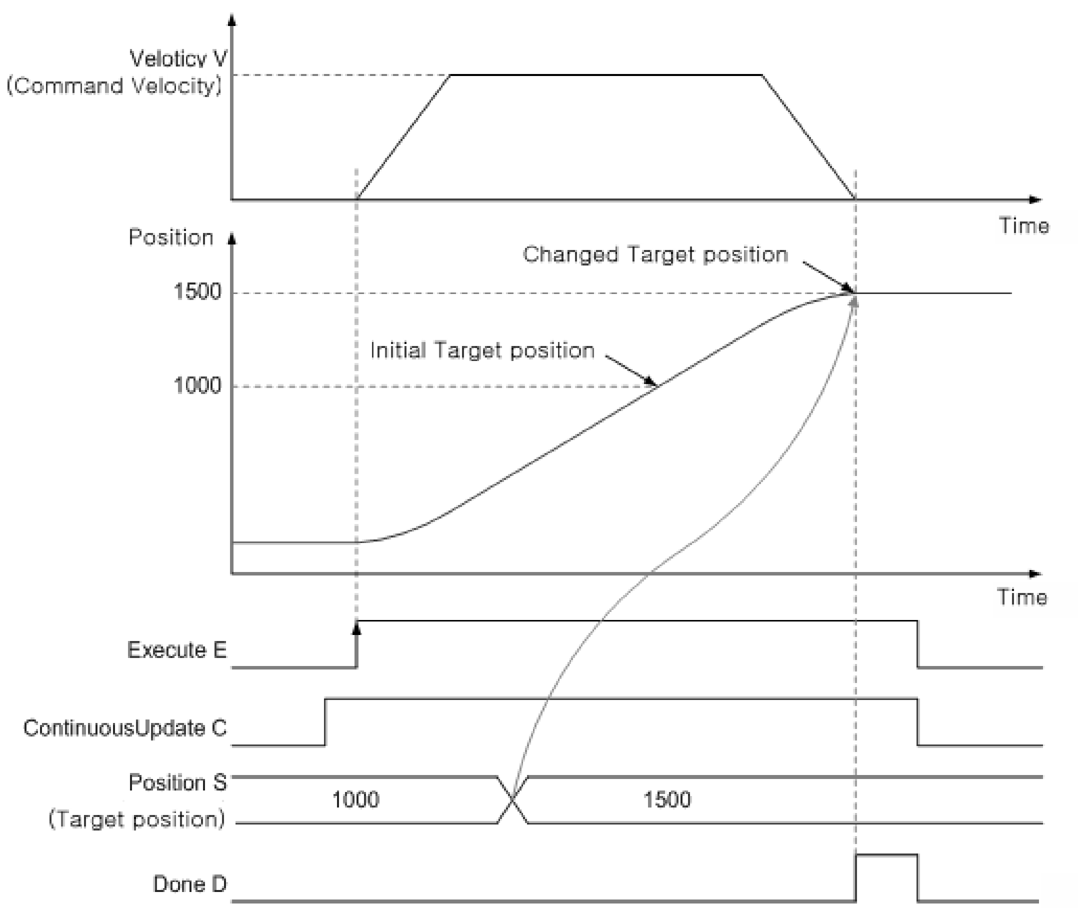
2. Position override

(1) 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.

(2) 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

(1) It is a function to conduct velocity, acceleration/deceleration and jerk override of the specified axis

(2) 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.

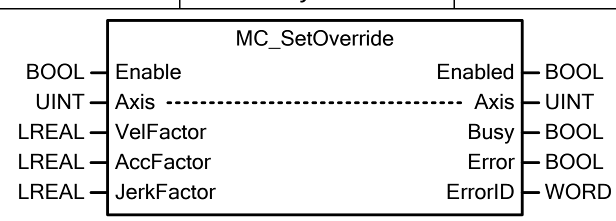
(3) For the execution of speed override operation at the rate on the current command speed, not an absolute value override (MC_SetOverride) motion function block is used for the override.

- 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.

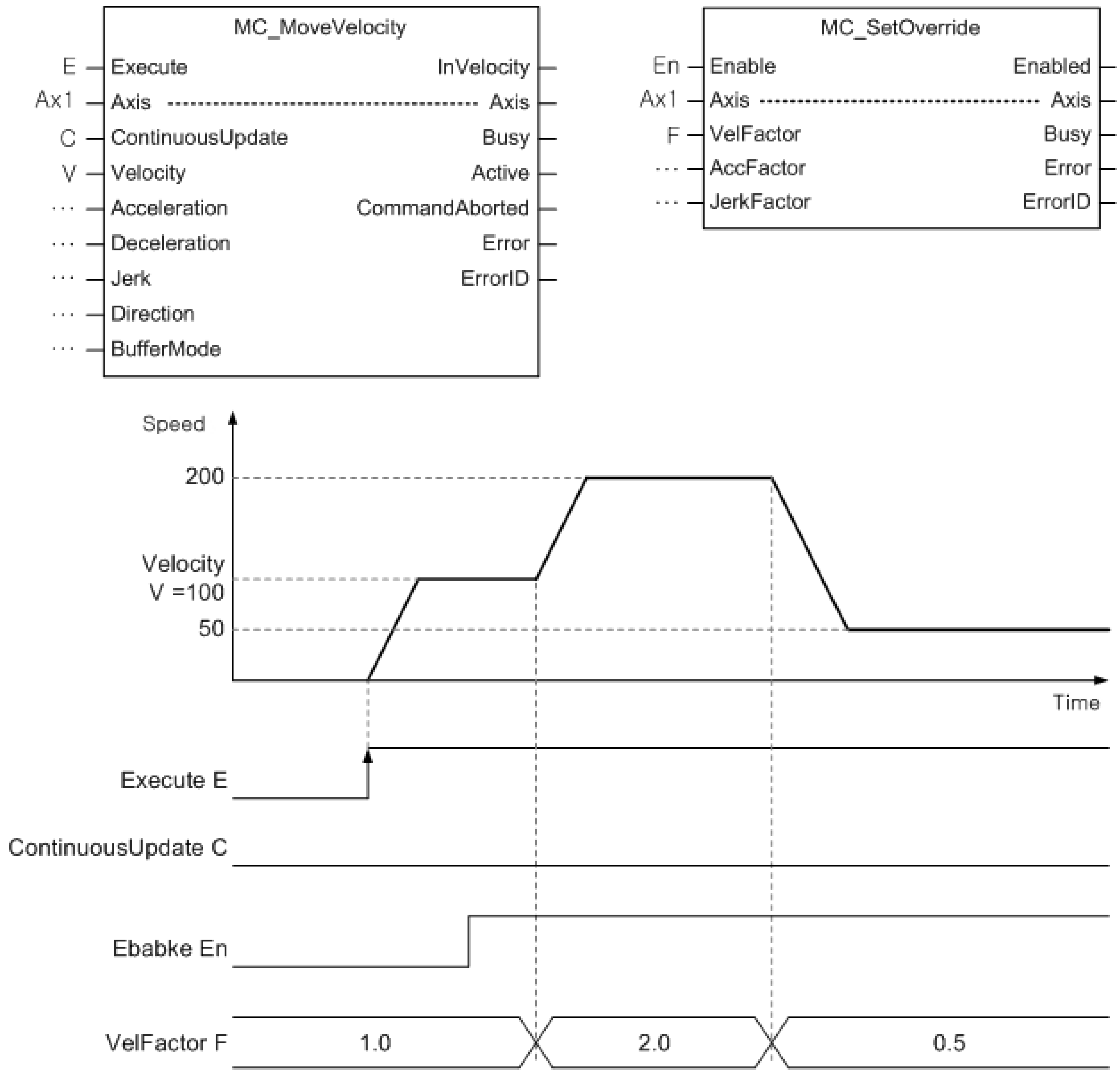
  0: percentage specified – Factor value operates at the rate on the current command speed
  1: unit value specified – Factor value is an absolute unit specified value of the set item

(4) Motion function block

Name	Description	Operation Condition
MC_SetOverride	Velocity override	level



[Example] Changes in velocity using override (MC_SetOverride) motion function block



4. Changes in the current position

(1) It is a function to change the current position of the axis to the value specified by users.

(2) In Position input, the position is specified. 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.

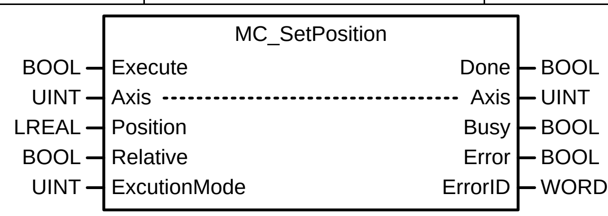
- 0: Absolute coordinate position
- 1: Relative coordinate position

(3) 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 in the same time with 'Buffered' of Buffermode

(4) Relevant motion function block

Name	Description	Operation Condition
MC_SetPosition	Setting the current position	Edge



5. Encoder preset

(1) It is a function to change the current encoder position value to any position value specified by users.

(2) In Encoder input, encoder to be changed is specified.

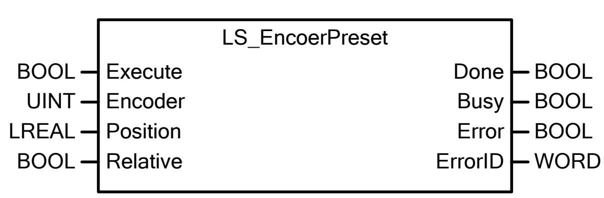
- 1: Encoder 1
- 2: Encoder 2

(3) 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 positionEncoder
-

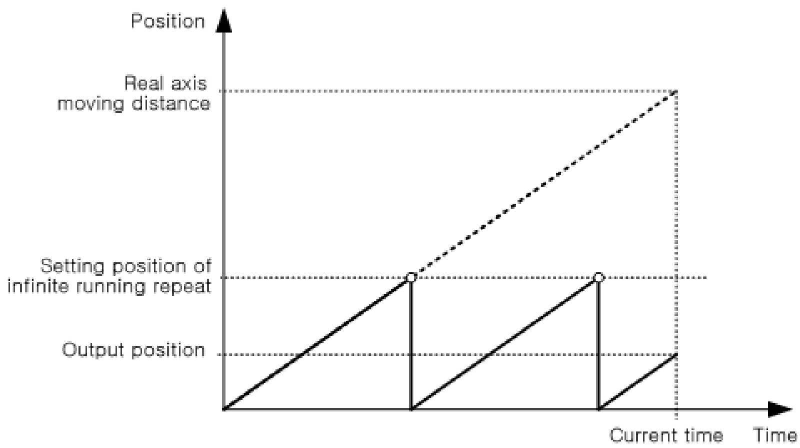
(4) Relevant motion function block

Name	Description	Operation Condition
LS_EncoderPreset	Encoder preset	Edge



6. Infinite running operation

(1) 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.



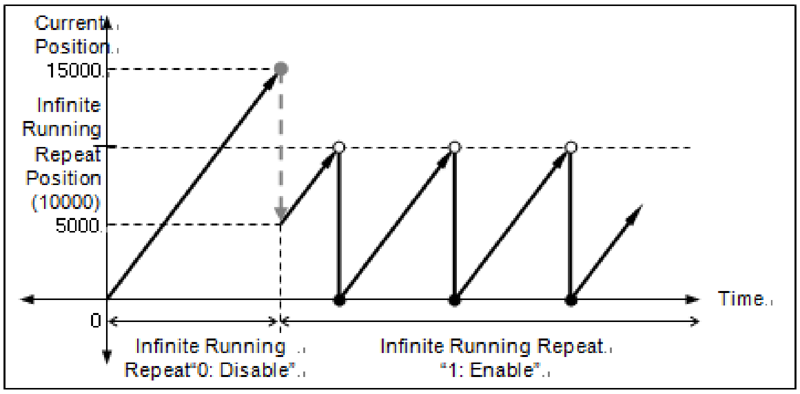
(2) The instant 'infinite running repetition' parameter among expansion parameters of operating parameters is set to Allow, the current position is automatically changed to value within the infinite running repletion position in case it is the value other than the range of infinite running repletion 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.



(3) Infinite running repetition setting of driving axis can be made by using software package or axis parameter change function.

(4) Relevant parameter setting

Item	Description	Settings	Initial Value
Infinite running repetition position	Set repeated position range value in case of being used as infinite running repetition mode	Long Real (LREAL) Positive	360 pls
Infinite running repetition	Set whether to allow infinite running repetition operation function	0: Disable 1: Enable	0: prohibited

8.3.2 Auxiliary Function of Control

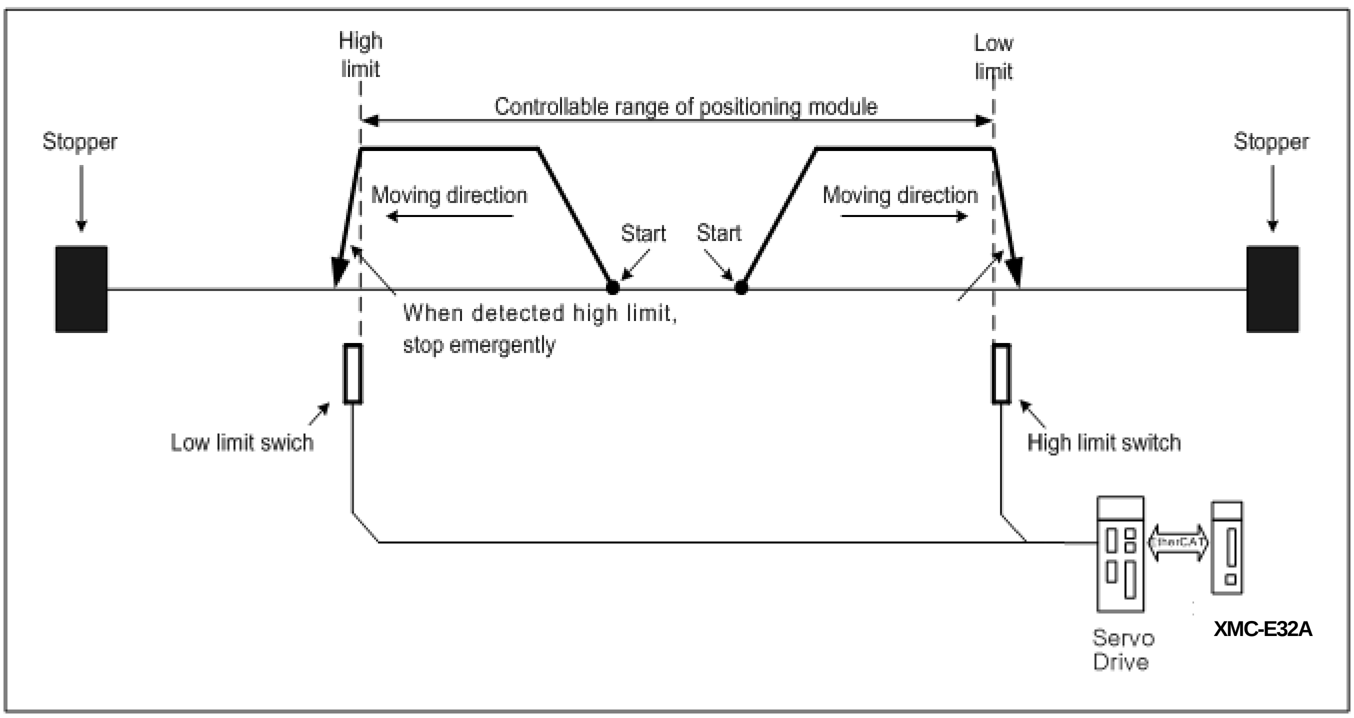
1. Hardware high/low limit

(1) 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’.

(2) 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.

(3) 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.

(4) Hardware high/low limit motions are as follows.



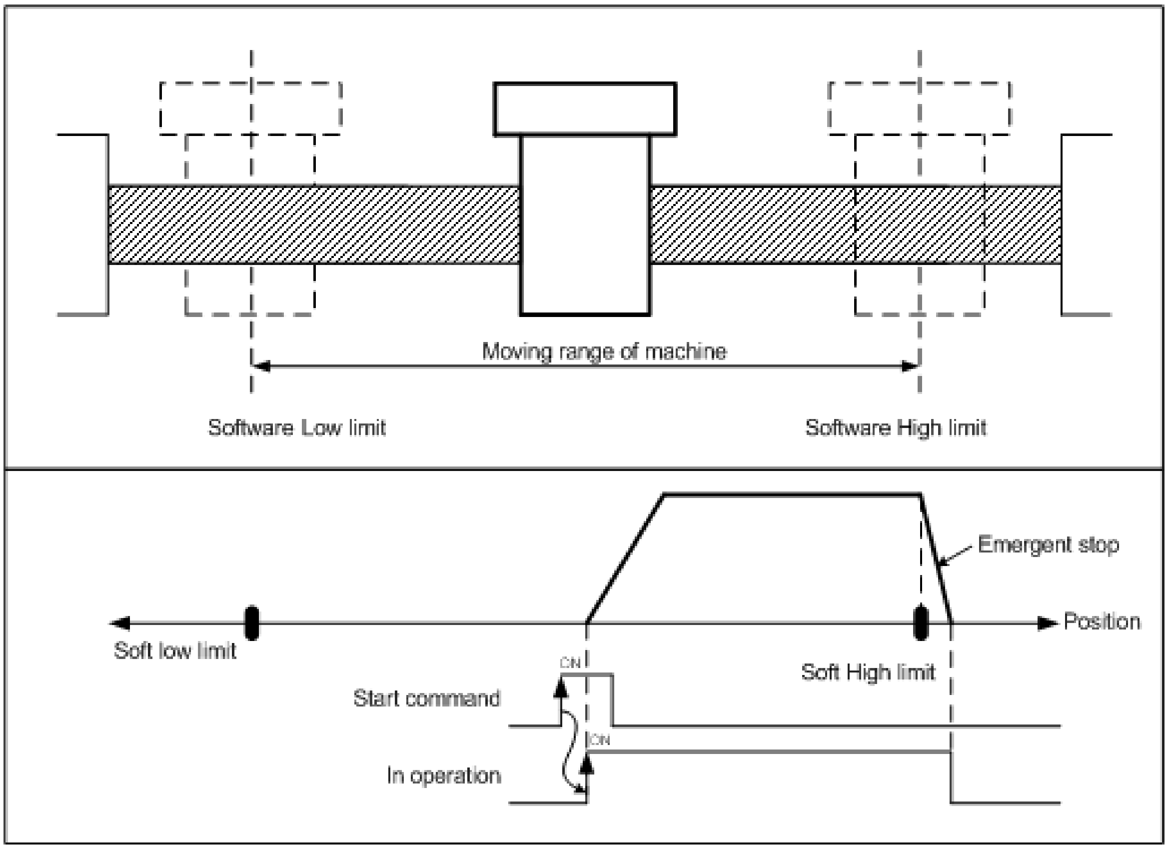
2. **Soft high/low limit**

(1) 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.

(2) Software stroke high/low limit of each driving axis can be set by using software package or axis parameter change function.

(3) If the axis is outside the range of stroke, axis error occurs.

(4) 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.



(5) Software high/low limit are don't detect in the following cases.

- In case soft upper limit value and lower limit value are set to the initial value (upper limit: 2147483647, lower limit: -2147483648)
- Software upper limit value and lower limit value are set to the same value (software upper limit = software 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"

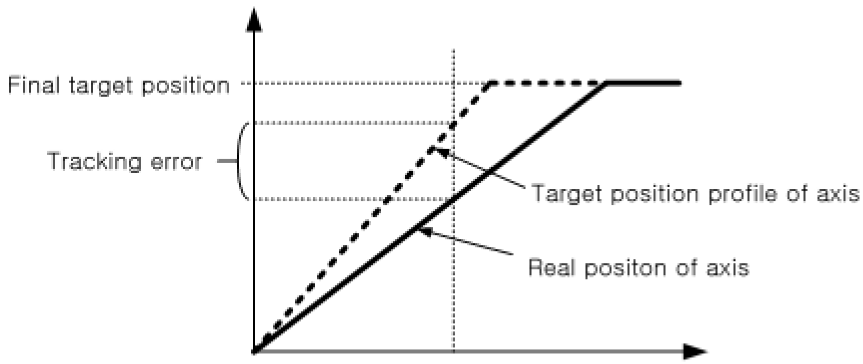
(6) Relevant parameter setting

Item	Description	Settings	Initial Value
S/W upper limit	Set the range of software limit function	Long Real(LREAL)	2147483647 pls
S/W lower limit			-2147483648 pls

3. Position tracking error

(1) 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.

(2) Position tracking tolerance of each driving axis can be set by using software package or axis parameter change function



(3) 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.

Motions according to the set value are as follows.

- ‘0: warning’
  When tracking error occurs, 「Above deviation alarm(_AXxx_DEV_WARN)」flag becomes On, and tracking error warning error (error code: 0x101D) occurs. The axis continues to operate without stopping.
- ‘1: alarm’
  When tracking error occurs, 「Above deviation alarm(_AXxx_DEV_ERR)」flag becomes On, and tracking error alarm error (error code: 0x101C) occurs. The axis makes a sudden stop at 「Sudden stop deceleration」.

(4) Inspection on tracking error is not performed in the following cases.

- In case 「Tracking error exceeding value」is 0
- In case of operation with homing or torque control

(5) Relevant parameter setting

Item	Description	Settings	Initial Value
Tracking error exceeding value	Set the value to detect more than the position deviation	0 or Long Real(LREAL) Positive	0
Tracking error level	Set the above deviation error level	0: warning 1: alarm	0: warning

4. Latch(Touch Probe)

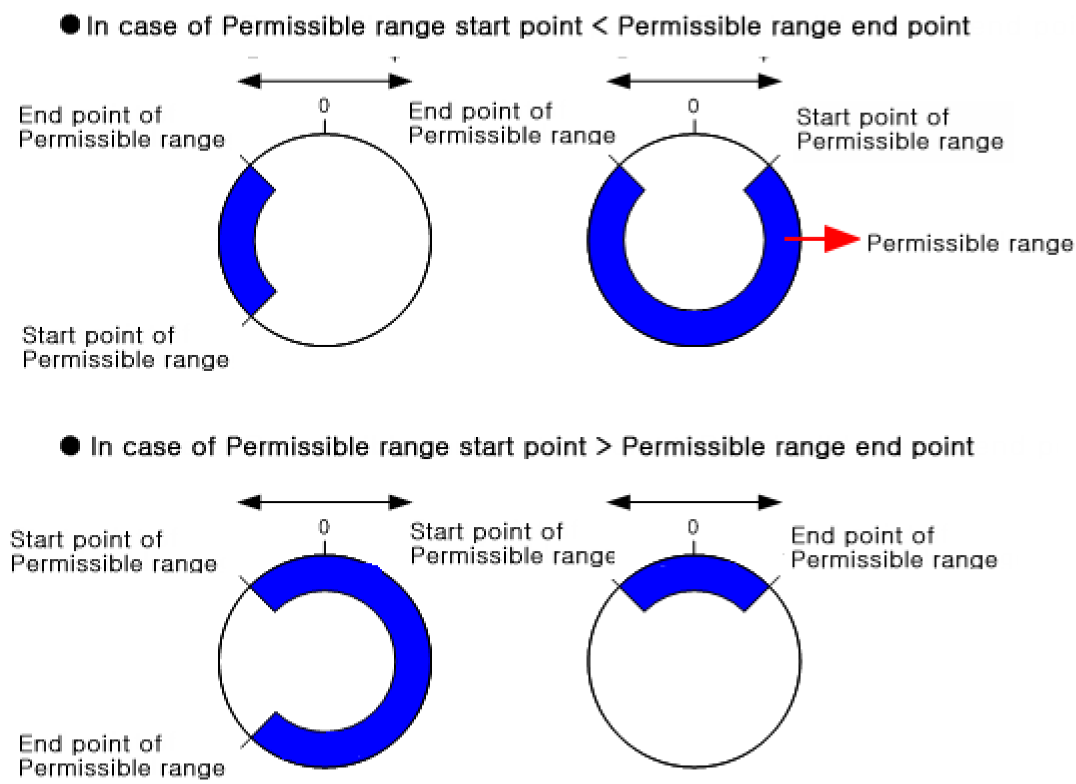
(1) It is a function to record the position of the axis when specific situation (Trigger event) occurs in the axis.

(2) 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 .

(3) The area in which latch (touch probe) function is operated can be specified.

- When permitted area is specified, operation can be made only within the specified area.
- In case of the infinite running repetition operation (rotary axis), the relationship of latch (touch probe) operating area according to the beginning and end of the permitted area is as follows.

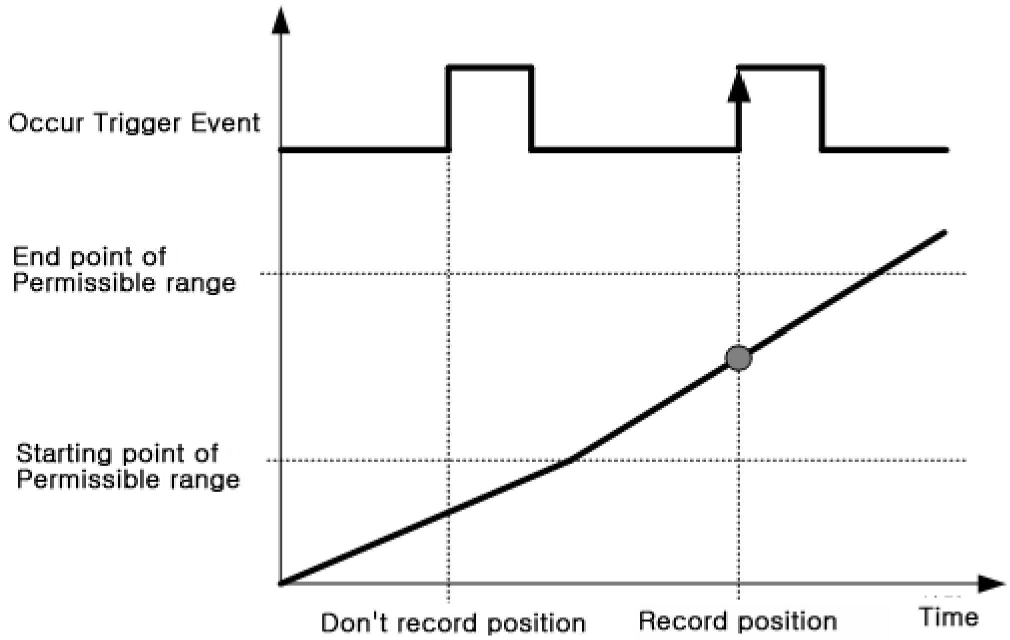


(4) For the use of latch (touch probe) function, the following objects should be included in PDO setting of slave parameter.

Trigger input	RxPDO	TxPDO
Touch Probe1	0x60B8:0 Touch probe function	0x60B9:0 Touch probe status 0x60BA:0 Forward direction positon value of touch probe1
Touch Probe2	0x60B8:0 Touch probe function	0x60B9:0 Touch probe status 0x60BC:0 Forward direction positon value of touch probe1

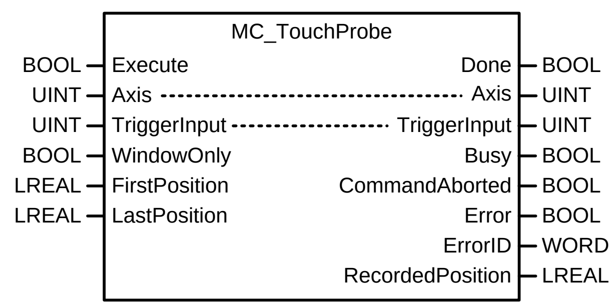
In case there are not above objects, an error (error code: 0x10E0) occurs when latch (touch probe) command is used.

(5) Operation timing

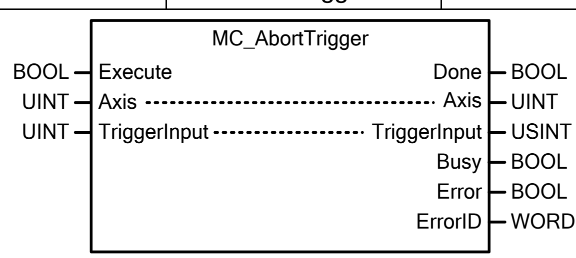


(6) Relevant motion function block

Name	Description	Operation Condition
MC_TouchProbe	Touch probe	Edge



Name	Description	Operation Condition
MC_AbortTrigger	Abort trigger	Edge



5. Error reset monitoring

(1) 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.

(2) If monitoring time is exceeded, error reset is not executed any more even if the error of the drive is not reset..

(3) Error reset monitoring time of each driving axis can be set by using software package or axis parameter change function.

(4) Relevant parameter setting

Item	Description	Settings	Initial Value
Error reset monitoring time	Set the monitoring time in case of resetting error that occurs in servo drive	1 ~ 1000 ms	100 ms

8.3.3 Data Management Function

1. Parameter management

(1) It is a function to read or change axis parameters stored in the module.

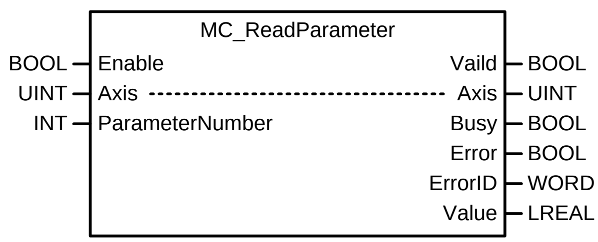
(2) It can change desired parameter values by specifying axis number and corresponding parameter number.

(3) Parameter value modified with parameter-write function is automatically stored in backup.ram in case there is no error.

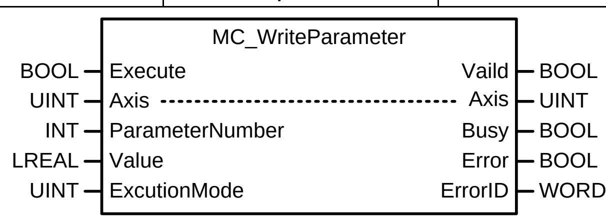
(4) For parameters to be set in “ParameterNumber”, refer to the motion function block item.

(5) Relevant motion function block

Name	Description	Operation Condition
MC_ReadParameter	Read parameter	Level



Name	Description	Operation Condition
MC_WriteParameter	Write parameter	Edge



2. Cam data management

It is able to read and change the cam data in program by the cam data Cam data reading/writing command

(1) Cam data reading

■ 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.

■ 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. For example, to save the "Main-axis Position" to the array vairable called MainAxPos[100], and "Sub-axis Position" value to SubAsPos[100] array variable, MainAxPos[0] should be set at MasterPoint of the function block, and SubAsPos[0] should be set at SlavePoint.

■ At CamCurveSel[4], the interpolation typesof the applicable points for each bit are stored.

CamCurveSel[0]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Point 8	Point 7	Point 6	Point 5	Point 4	Point 3	Point 2	Point 1
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	Point 16	Point 15	Point 14	Point 13	Point 12	Point 11	Point 10	Point 9
	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
	Point 24	Point 23	Point 22	Point 21	Point 20	Point 19	Point 18	Point 17
	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
	Point 32	Point 31	Point 30	Point 29	Point 28	Point 27	Point 26	Point 25
CamCurveSel[1]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Point 40	Point 39	Point 38	Point 37	Point 36	Point 35	Point 34	Point 33
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	Point 48	Point 47	Point 46	Point 45	Point 44	Point 43	Point 42	Point 41
	Bit 23	Bit 22	Bit 21	Bit 20	Bit 1…09	Bit 18	Bit 17	Bit 16
	Point 56	Point 55	Point 54	Point 53	Point 52	Point 51	Point 50	Point 49
	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
	Point 64	Point 63	Point 62	Point 61	Point 60	Point 59	Point 58	Point 57
CamCurveSel[2]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Point 72	Point 71	Point 70	Point 69	Point 68	Point 67	Point 66	Point 65
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	Point 80	Point 79	Point 78	Point 77	Point 76	Point 75	Point 74	Point 73
	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
	Point 88	Point 87	Point 86	Point 85	Point 84	Point 83	Point 82	Point 81
	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
	Point 96	Point 95	Point 94	Point 93	Point 92	Point 91	Point 90	Point 89
CamCurveSel[3]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Not used				Point 100	Point 99	Point 98	Point 97
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	Not used							
	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
	Not used							
	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
	Not used							

(2) Cam data writing

■ CamDataWrite command writes the value set in StartSlope and EndSlope of the cam profile designated by CamTable ID and the device value set in MasterPoint and SlavePoint in the number designated by CamPointNum as "Main-axis Position" and "Sub-axis Position" when Execute input is on.

■ At CamCurveSel[4], the interpolation types of the applicable points for each bit can be set.

CamCurveSel[0]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Point 8	Point 7	Point 6	Point 5	Point 4	Point 3	Point 2	Point 1
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	Point 16	Point 15	Point 14	Point 13	Point 12	Point 11	Point 10	Point 9
	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
	Point 24	Point 23	Point 22	Point 21	Point 20	Point 19	Point 18	Point 17
	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
	Point 32	Point 31	Point 30	Point 29	Point 28	Point 27	Point 26	Point 25
CamCurveSel[1]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Point 40	Point 39	Point 38	Point 37	Point 36	Point 35	Point 34	Point 33
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	Point 48	Point 47	Point 46	Point 45	Point 44	Point 43	Point 42	Point 41
	Bit 23	Bit 22	Bit 21	Bit 20	Bit 1…09	Bit 18	Bit 17	Bit 16
	Point 56	Point 55	Point 54	Point 53	Point 52	Point 51	Point 50	Point 49
	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
	Point 64	Point 63	Point 62	Point 61	Point 60	Point 59	Point 58	Point 57
CamCurveSel[2]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Point 72	Point 71	Point 70	Point 69	Point 68	Point 67	Point 66	Point 65
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	Point 80	Point 79	Point 78	Point 77	Point 76	Point 75	Point 74	Point 73
	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
	Point 88	Point 87	Point 86	Point 85	Point 84	Point 83	Point 82	Point 81
	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
	Point 96	Point 95	Point 94	Point 93	Point 92	Point 91	Point 90	Point 89
CamCurveSel[3]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Not used				Point 100	Point 99	Point 98	Point 97
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	Not used							
	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
	Not used							
	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
	Not used							

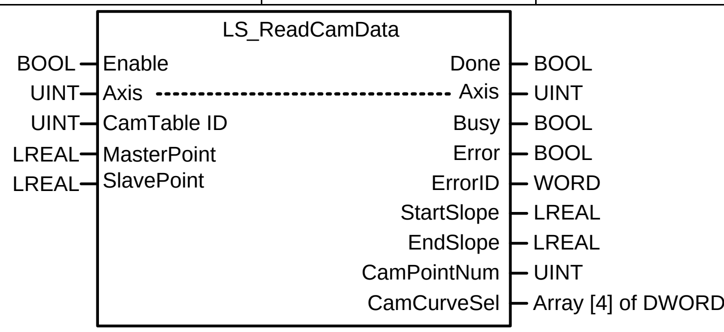
■ CamTableID input can set the range of 1~32. The setting value outside the setting range causes error "16#000B"

■ CamPointNum can set the range of 1~100. The setting value outside the setting range causes error "16#000B"

(3) Motion function block

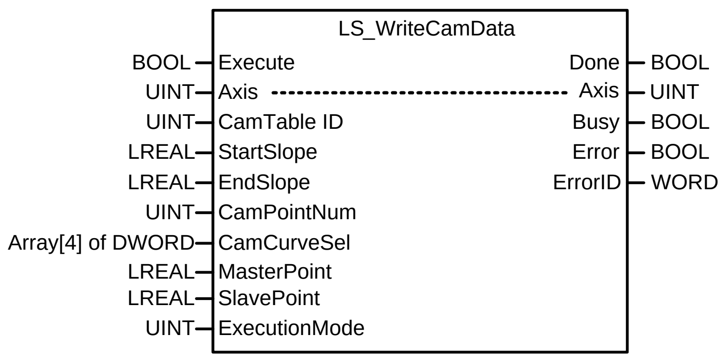
■ Cam data reading

Name	Description	Operation Condition
LS_ReadCamData	Cam data reading	Level



■ Cam data writing

Name	Description	Operation Condition
LS_WriteCamData	Cam data writing	Edge



3. SDO parameter management

(1) This function reads or changes SDO parameters of slave devices connected via network.

(2) 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

(3) Index input can be set as follows. If it is not set as the setting value, "Error 0x1F12" occurs.

설정값	내용
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) In Subindex, values ranging from 0 to 255 can be entered, and if set outside the range, "Error 0x1F12. " occurs.

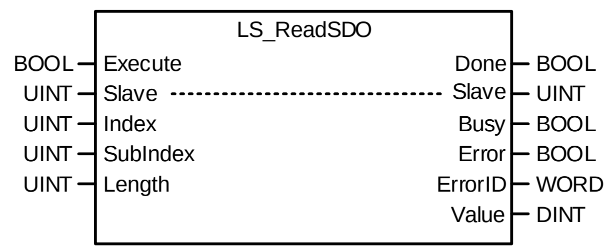
(5) In Length, values ranging from 1 to 4 can be entered, which mean 1 to 4 bytes. Setting the value outside the above range will cause "Error 0x1F12."

(6) 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.

(7) Motion function block

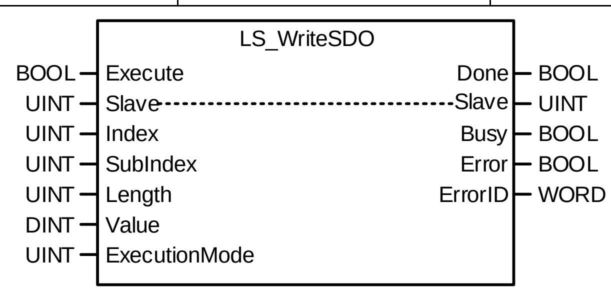
■ SDO reading

Name	Description	Operation Condition
MC_ReadSDO	SDO reading	Level



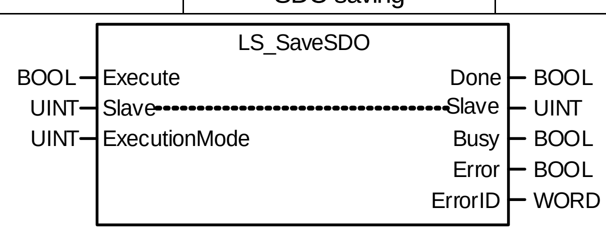
■ SDO writing

Name	Description	Operation Condition
MC_WriteSDO	SDO writing	Edge



■ SDO saving

Name	Description	Operation Condition
MC_SaveSDO	SDO saving	Edge

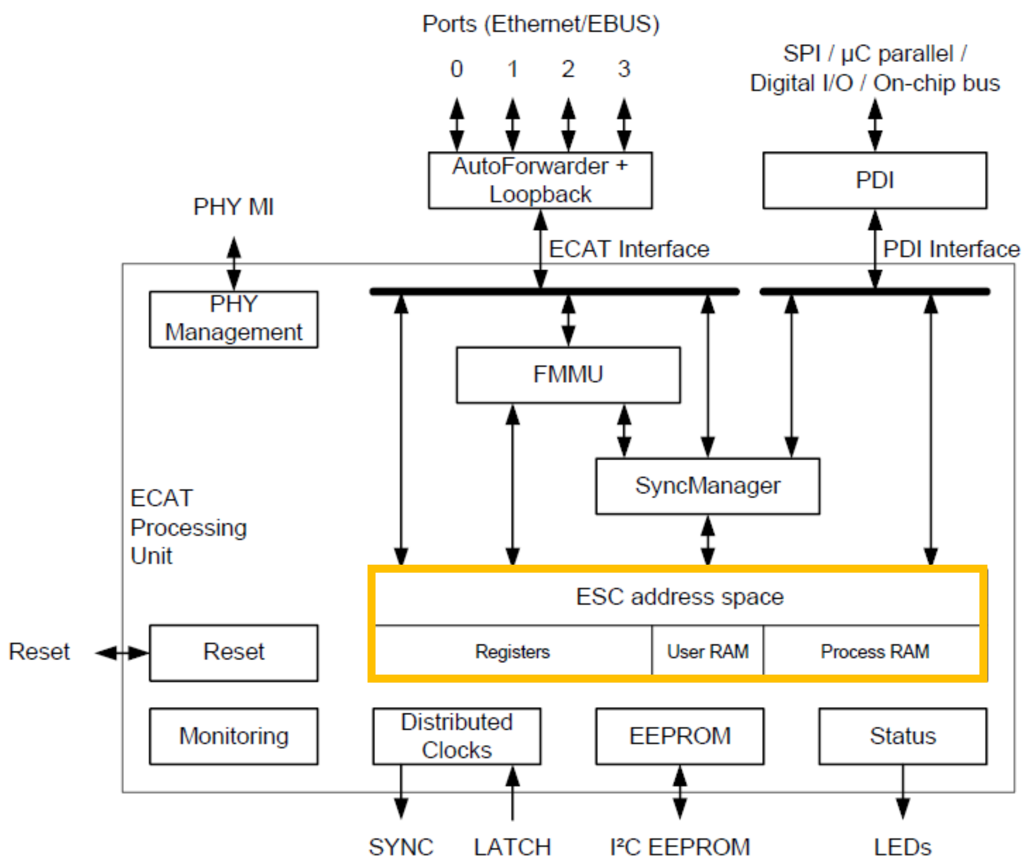


8.3.4 EtherCAT Communication Diagnosis Function

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)

(1) 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.



(2) ESC address space is configured as follows.

Address	Define	Remarks
0x0000	ESC Register	ESC Information, FMMU, SyncManager, Distributed Clocks(DC).
:		
0x0FFF		
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. ESC reading

(1) This function reads data in ESC of the slave devices connected via network.

(2) Adp(Address position) is designating the address of the EtherCAT slave device. The following values can be set depending on the EcatCmd setting. 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 : 0xFFDD: 36th slave connected
4 (FPRD)	1001 ~ 1032: 1 Axis ~ 32 Axis 1033 ~ 1036: 33 Axis IO ~ 36 Axis IO
7 (BRD)	-

(3) In Length, values ranging from 1 to 4 can be entered, which means 1-4 bytes.

(4) At EcatCmd, the type of command to use when reading ESC (EtherCAT Slave Controller) is specified. The following three commands can be used:

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. The slave device receiving Adp with 0 value will read the 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 values depending on the slave device connection order are as follows

Slave controller	Setting value
The first slave connected	0
The second slave connected	0xFFFF
:	:
36th slave connected	0xFFDD

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 the master are as follows.

Slave controller	Setting value
1 Axis	1001
2 Axis	1002
:	:
32 Axis	1032
33 Axis IO	1033
:	:
36 Axis IO	1036

3) 7 – BRD (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

(5) Value and Wkc 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.

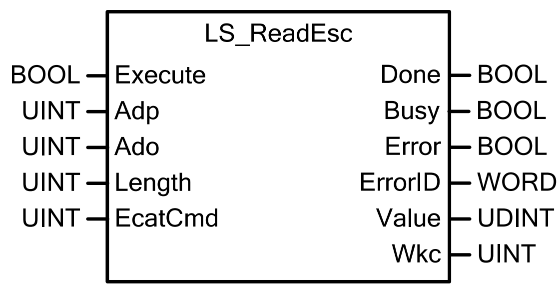
(6) Wkc stands for Working Counter. If data is successfully written at the designated slave device, it increases by 1. If EcatCmd is 7(BRD), it increases by 1 due to all slaves that performed normal reading operation.

(7) After the execution of ESC read command, if normal data read operation is executed from the designated slave device, Doneoutput is on.

(8) 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.

(9) Function block

Name	Description	Operation Condition
LS_ReadEsc	ESC reading	Edge



(10) In the following cases, ESC reading cannot be performed due to errors, properly.

1) No slave device is connected to module (Error Code: 0x0F09)

2) Adp setting value is outside the range (Error Code: 0x0F60)

3) Length setting value is outside the range (Error Code: 0x0F61)

4) EcatCmd setting value is outside the range (Error Code: 0x0F62)

5) No response to ESC read command (Error Code: 0x0F63)

3. ESC writing

(1) This function writes data in ESC of the slave devices connected via network.

(2) 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 : 0xFFDD: 36th slave connected
5 (FPWR)	1001 ~ 1032: 1 Axis ~ 32 Axis 1033 ~ 1036: 33 Axis IO ~ 36 Axis IO
8 (BWR)	-

(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.

(4) At EcatCmd, the type of command to use when readingESC (EtherCAT Slave Controller) is specified. The following three read commands can be used:

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 values depending on the slave device connection order are as follows.

Slave controller	Setting value
The first slave connected	0
The second slave connected	0xFFFF
:	:
36th slave connected	0xFFDD

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 the master is as follows.

Slave controller	Setting value
1 Axis	1001
2 Axis	1002
:	:
32 Axis	1032
33 Axis IO	1033
:	:
36 Axis IO	1036

3) 8 – 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.

(5) 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 1through each slave device specified in EcatCmd and Adp.

(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.

(8) 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)

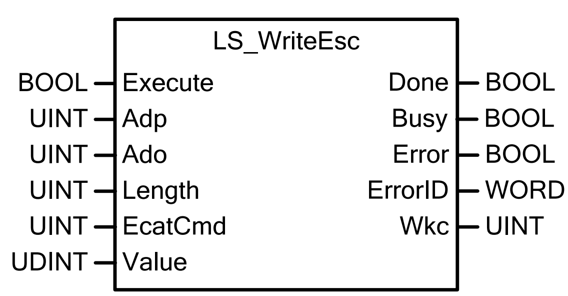
Ado range	Define
0x0010 ~ 0x0011	Configured Station Address
0x0020 ~ 0x0021	Write Protection
0x0030 ~ 0x0031	
0x0040	ESC Reset ECAT
0x0100 ~ 0x0103	DL Control
0x0120 ~ 0x0121	AL Control
0x0600 ~ 0x06FF	FMMU
0x0800 ~ 0x087F	SyncManager
0x0900 ~ 0x09FF	Distributed Clocks

※ 0x0120 (AL Control) register can be written after the connection of normal communication, not the execution of connection/disconnection command.

(9) 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 for the preceding command

(10) Motion function block

Name	Description	Operation Condition
LS_WriteEsc	ESC writing	Edge

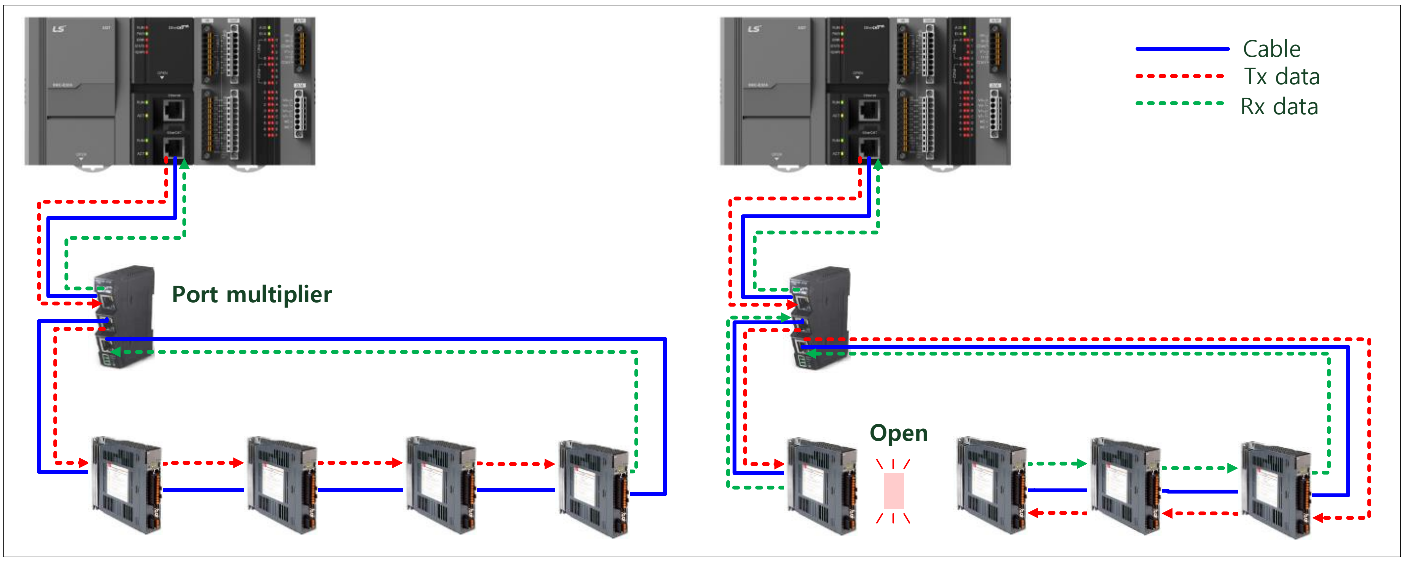


(11) In the following cases, ESC writing cannot be performed due to errors, properly

1) No slave device is connected to module (Error Code: 0x0F09)
2) Adp setting value is outside the range (Error Code: 0x0F70)
3) Length setting value is outside the range (Error Code: 0x0F71)
4) EcatCmd setting value is outside the range (Error Code: 0x0F72)
5) No response to ESC read command (Error Code: 0x0F73)
6) 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.



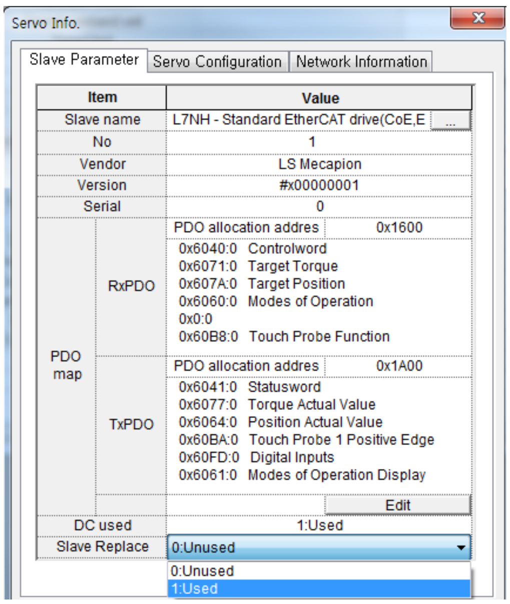
※ Port multiplier of up to 1 can be used. In case of using a port multiplier, it occupies an IO slave number. Therefore, a caution is required when using since the use of the port multiplier reduces the number of IO slaves available.

8.3.6 Replace 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.

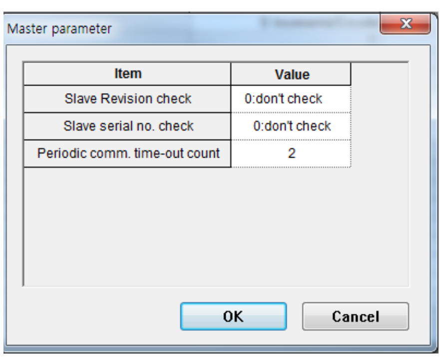
1. Replace function during connection setting

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.



2.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.



3.The way of slave exchange during connection

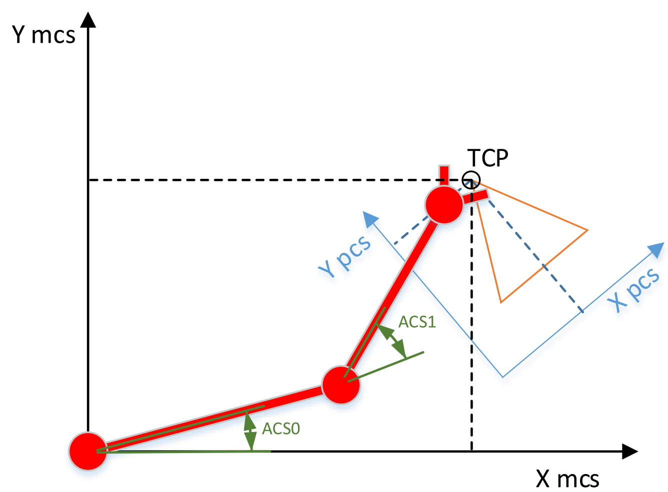
(1) Remove the input/output cable of the slave to be replaced during the network connection.
(2) Shut off the power of the slave.
(3) Apply the power to the slave to be replaced.
(4) Connect the cable of one side of the port.
(5) Connect the cable of one side of the port.
(The simultaneous connection of input/output cables may prevent normal replacement.)
(6) When the slave communication is restored, connect the cable of the other side.

8.4 Coordinate Systems Operation Function

8.4.1 Summary of the 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.

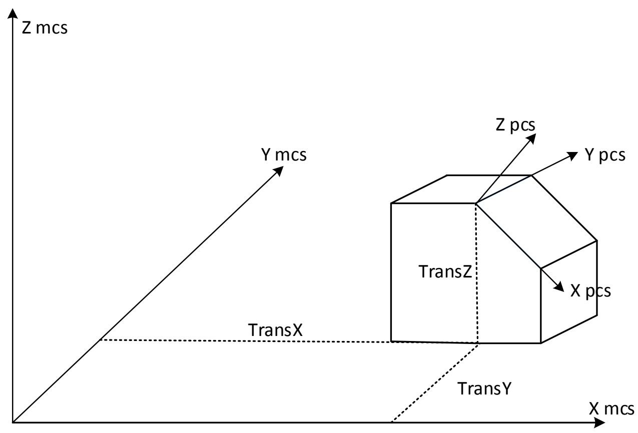
MCS: Machine Coordinate System (MCS) is related with machines (robots). It is the fixed home position of the mechanical system represented as the Cartesian coordinate system.

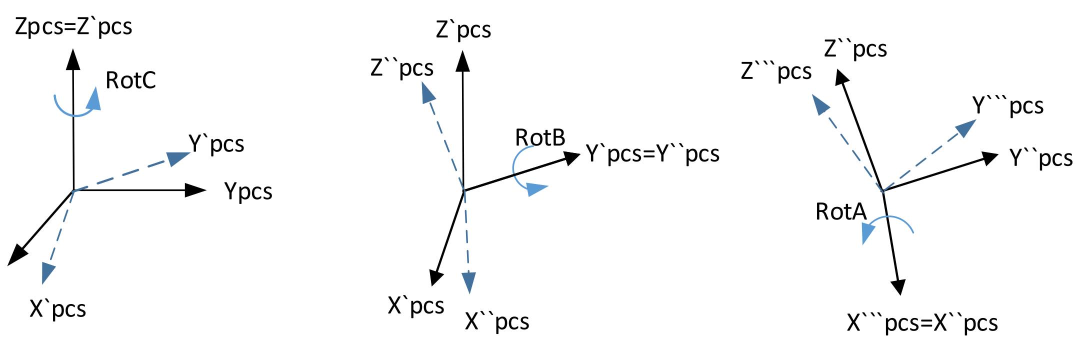
PCS: Product Coordinate System based on MCS represents the position of products being moved or rotated. PCS is linked to the products through a program, and a user can be changed.

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 PCS Setting

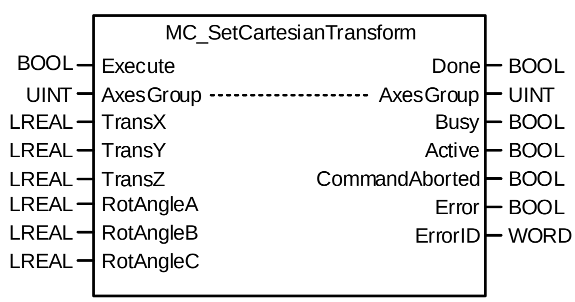
PCS represents TCP on the work stand. TCP is defined by rotation and movement from the origin point. The parameter to convert PCS into MCS can be set using MC_SetCartesianTransform function block or setting axes group parameter. In MC_SetCartesianTransform, TransX/TransY/TransZ represents the distance 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-axisPCS rotation is performed in the following order: RotC->RotB->RotA





1. 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. Either XYZ or Delta3 can be selected as the robot type.

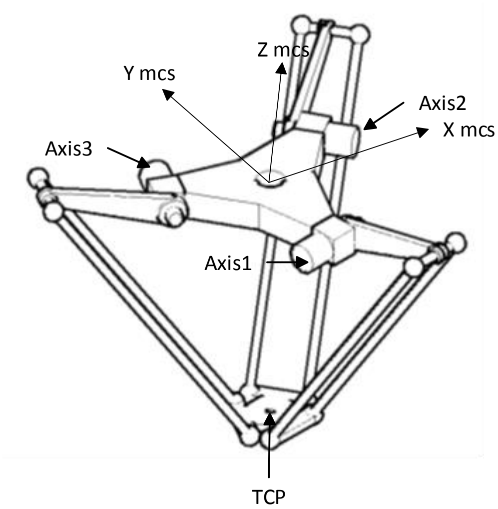
   (1) XYZ(Cartesian coordinates) robot

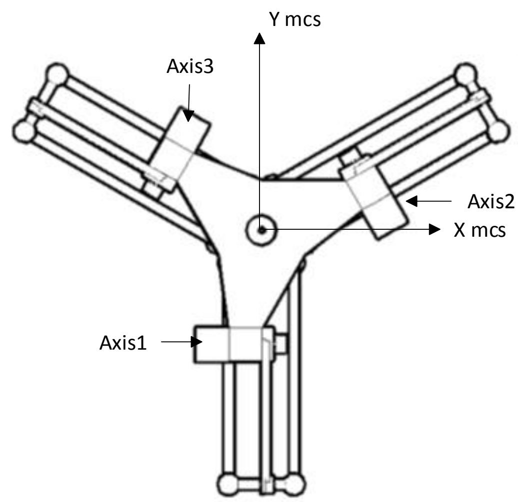
   XYZ is a robot type with servo motors connected to X/Y/Z axes, which can perform the operation in Cartesian coordinates, and it does not require additional kinematic analysis between ACS and MCS.

   (2) Delta3

   MCS of the Delta robot

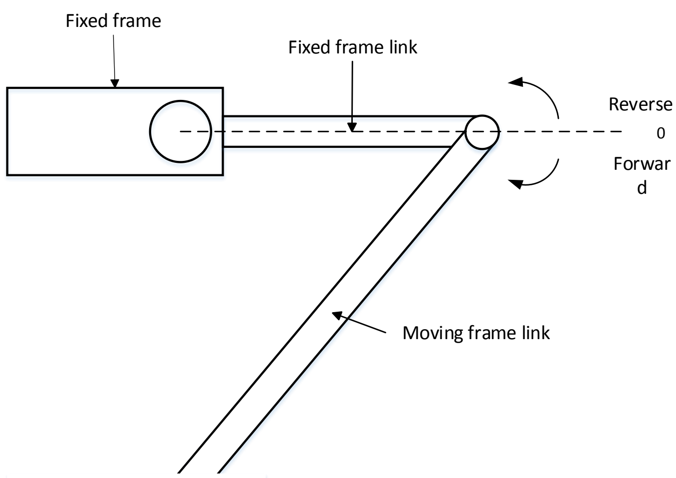
   In case of a Delta robot, the center of Fixed Frame is defined as MCS, and the relationship between each axis connected to Delta and MCS are as shown below.





   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.

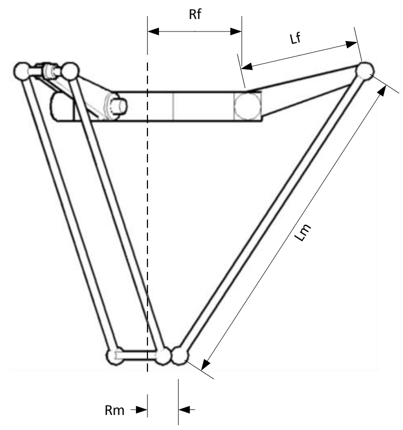


2. Machine information, machine parameter setting

(1) XYZ

XYZ robot does not require separate machine parameters, as the position of each axis matches the XYZ coordinates of TCP.

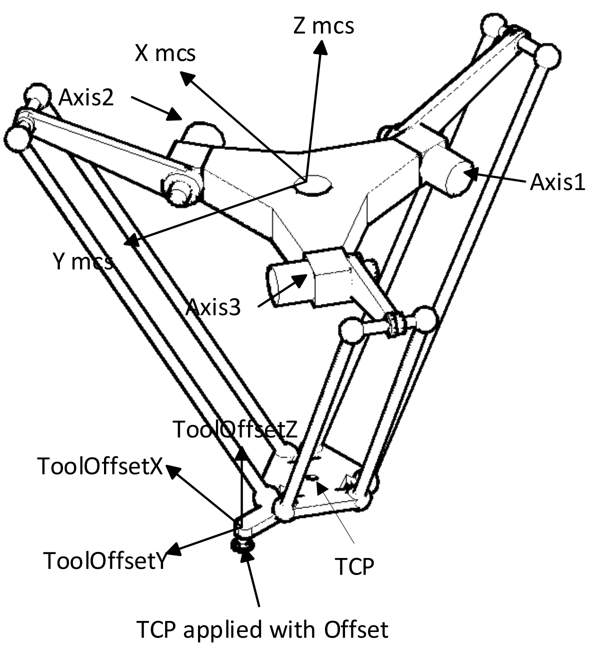
(2) Delta3



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)

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.



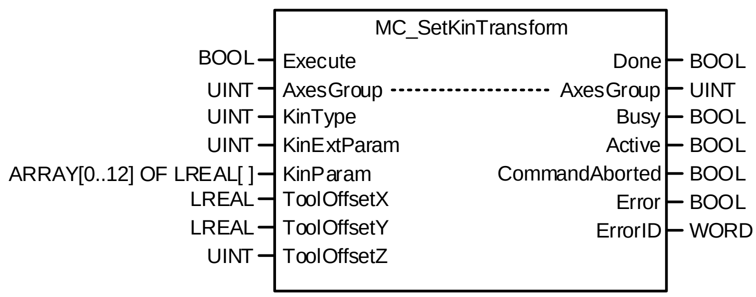
4. Axis group, axis configuration setting

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	Axis number	EA			
		1 Axis	2 Axis	3 Axis	4 Axis
XYZ	3 Axes	mm	mm	mm	-
Delta3	3 Axes	degree	degree	degree	-

5. Function block

Name	Description	Operation Condition
MC_SetKinTransform	Machine information setting	Edge



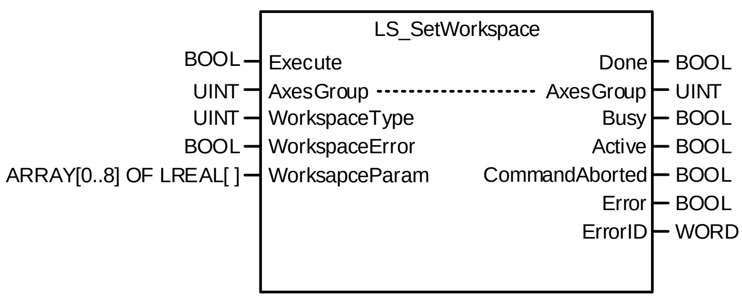
8.4.5 Work Space Setting

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 inputIf 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.

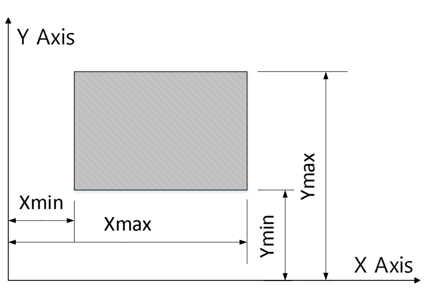
Name	Description	Operation Condition
LS_SetWorkspace	Work space setting	Edge



2. Work space type and parameter

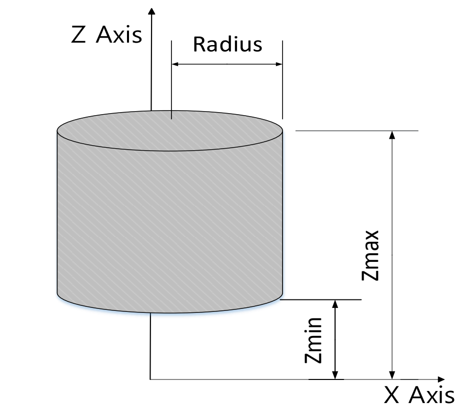
The work space type supports 4 types of Rectangle/Cylinder/Delta/Sector.

Rectangle



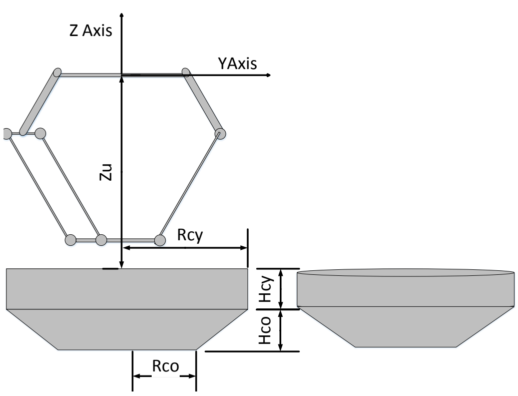
Parameter	Value
WorkspaceParam[0]	X max(mm)
WorkspaceParam[1]	X min(mm)
WorkspaceParam[2]	Y max(mm)
WorkspaceParam[3]	Y min(mm)
WorkspaceParam[4]	Z max(mm)
WorkspaceParam[5]	Z min(mm)

Cylinder



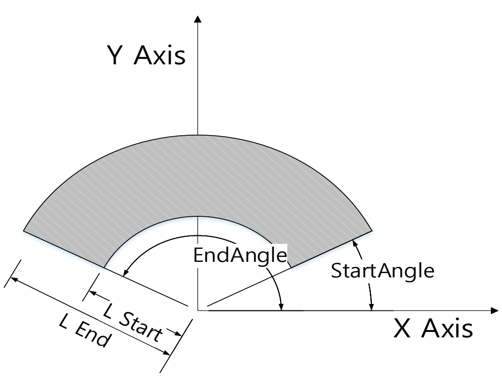
Parameter	Value
WorkspaceParam[0]	Radius(mm)
WorkspaceParam[1]	Z max(mm)
WorkspaceParam[2]	Z min(mm)

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. Perform linear interpolation from the start position to the target position (position designated by the positioning data). Positioning control is based on the position designated at return to origin point.
2. Set Position[] to define the TCP target position.

Variable	Define	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

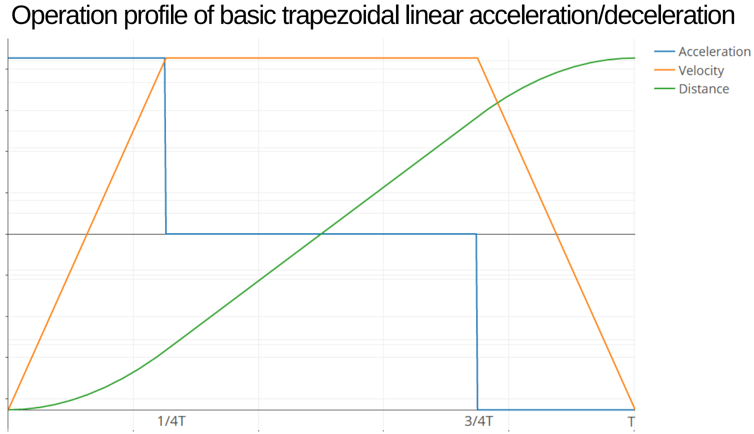
3. Depending on the robot type, some Position variable areas may not be applied. Data input in the unapplied areas is not reflected in coordinate system operation.

Variable	Define	
	XYZ	Delta3
Position[0]	Applied	Applied
Position[1]	Applied	Applied
Position[2]	Applied	Applied
Position[3]	Not applied	Not applied
Position[4]	Not applied	Not applied
Position[5]	Not applied	Not applied

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.

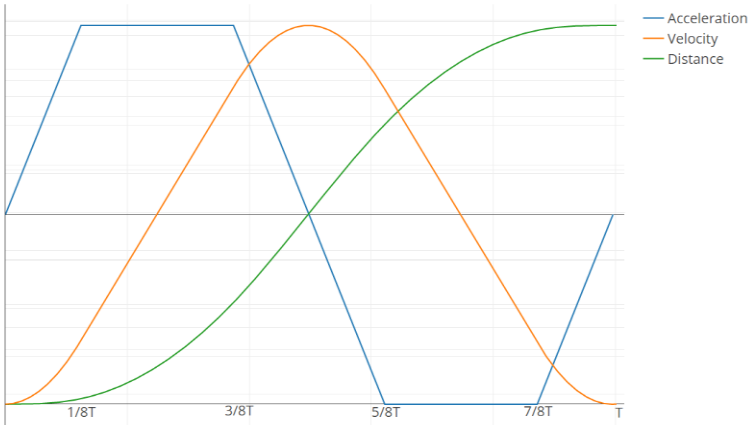
Trapezoid

Operation profile of basic trapezoidal linear acceleration/deceleration



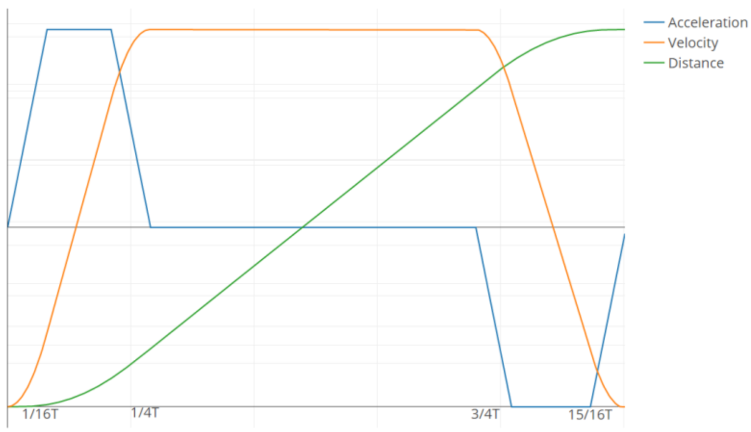
Sine1

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.



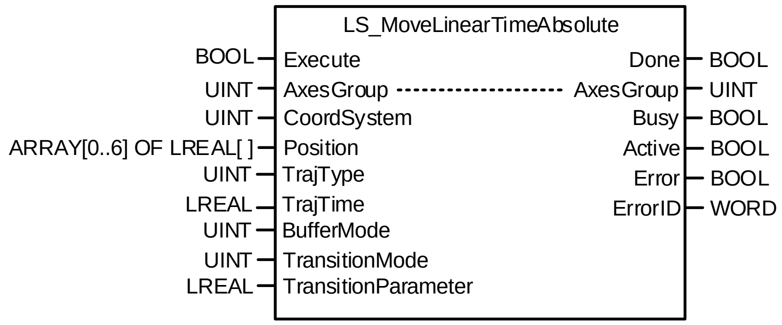
Sine2(Sine With Constant)

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. Function block

Name	Description	Operation Condition
MC_MoveLinearTimeAbsolute	Coordinate system absolute position time linear interpolation operation	Edge



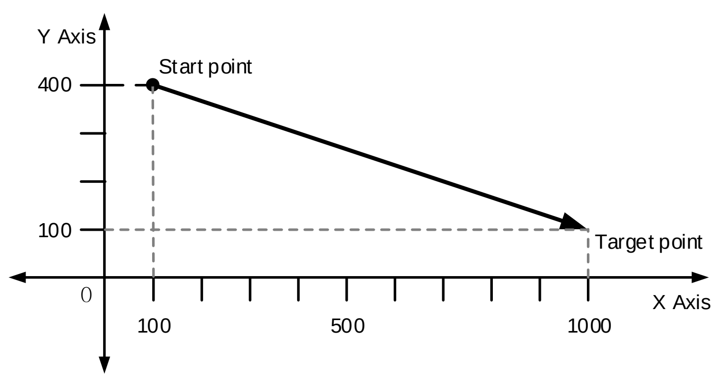
9. Limitation

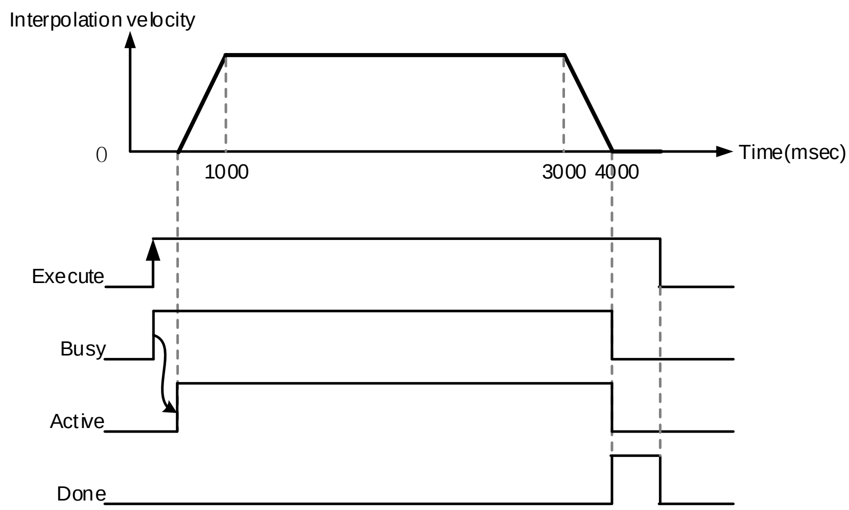
Coordinate system absolute position time linear interpolation control cannot be performed in case of the following errors

- 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)
- Of the component axes, an axis where the home position is not determined exists (Error Code: 0x20B0)
- The operation velocity of the component axes exceeds the velocity limit of each axis (Error Code: 0x20B9)
- Of the component axes, there is an axis being executed with the infinite running repeat operation (Error Code: 0x20BA)

10. Operation timing

- Start point: (100.0, 400.0, 0.0)
- Target point: (1000.0, 100.0, 0.0)
- Target time: 4000msec
- Operation type : 0



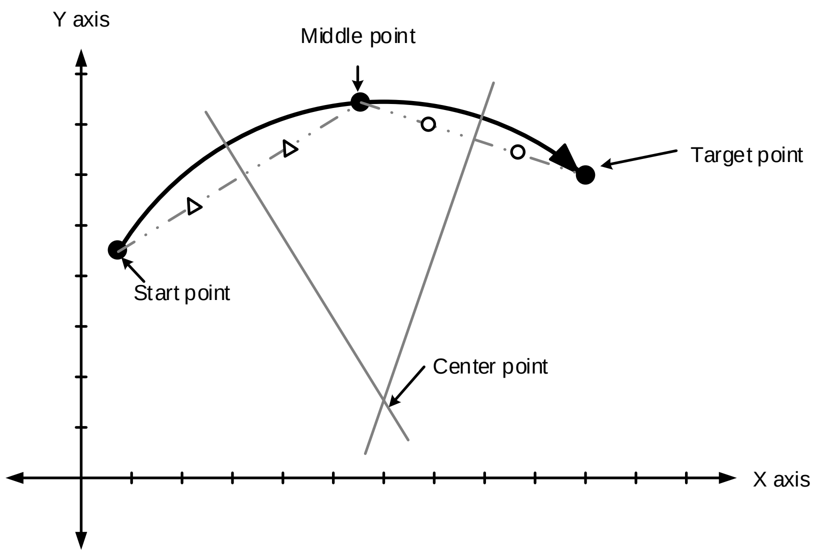


8.4.7 Circular Interpolation Operation for Coordinate System

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 types of circular interpolation: center point method, where the TCP passes the position designated by the auxiliary point following the CircMode setting and the auxiliary point; center point method where the position designated by the auxiliary point is the center point; and diameter method where the value set as the auxiliary point is the diameter of the arc. 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.

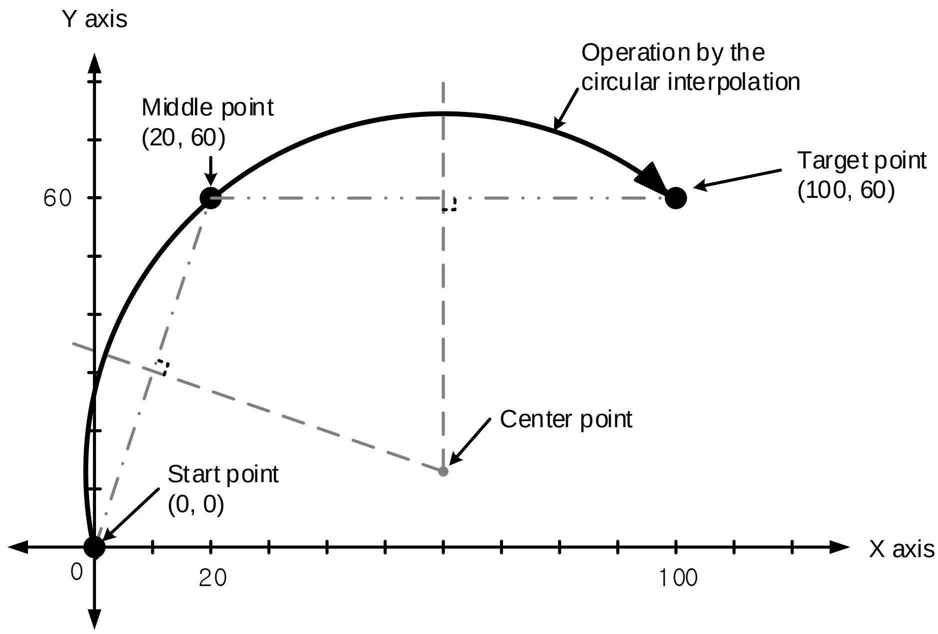
2. Circular interpolation using midpoint specification
   (3) Circular interpolation is performed by starting at the start position, passing the center point set as the auxiliary point, and moving to the target position
   (4) A circular trajectory is created of which the center point is the crossing point of the perpendicular bisectors bet ween the start position and the mid position, and the mid position and the target position.
   (5) The movement direction is automatically determined by the set target position and the auxiliary point for circular interpolation



   (6) Limitation
   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)
   - The midpoint specified as the auxiliary point is the same as the start or target position (Error Code: 0x20A4)
   - The start point is the same as the target point (Error Code: 0x20A5)
   - The calculated radius of the arc exceeds 2147483647pls (Error Code: 0x20A6)
   - The start point, the auxiliary point, and the target point are on the same straight line (Error Code: 0x20A7)
   - One or more of the component axes is performing the infinite running repeat operation (Error Code: 0x20A8)

(7) Operation pattern

- Start point: (0.0, 0.0,0.0)
- Target point: (100.0, 60.0,0.0)
- Middle point: (20.0, 60.0)
- CircMode: Middle point(0)
- PathChoice: - (Ignore in the circular Interpolation using midpoint)

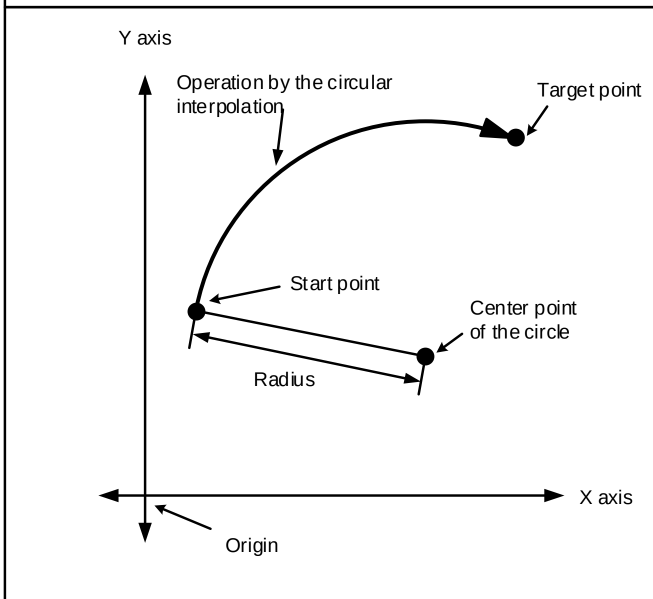
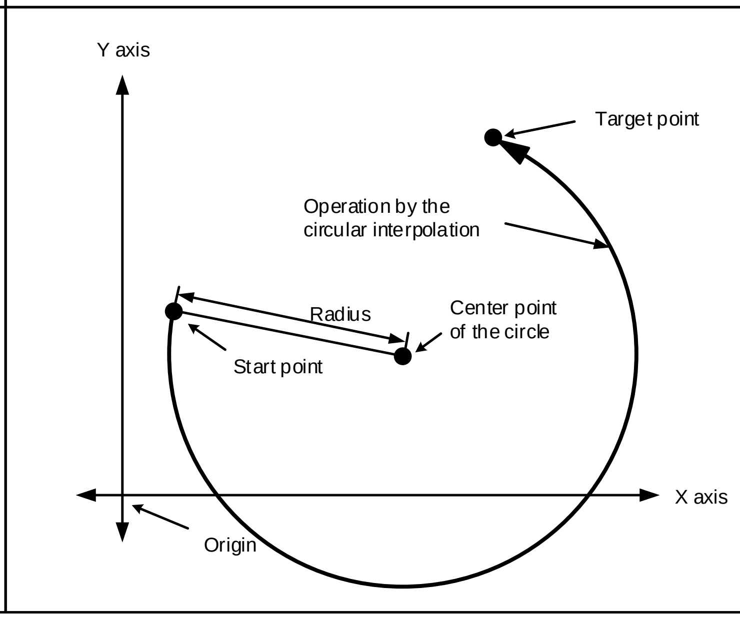


3. Circular interpolation using center point specification

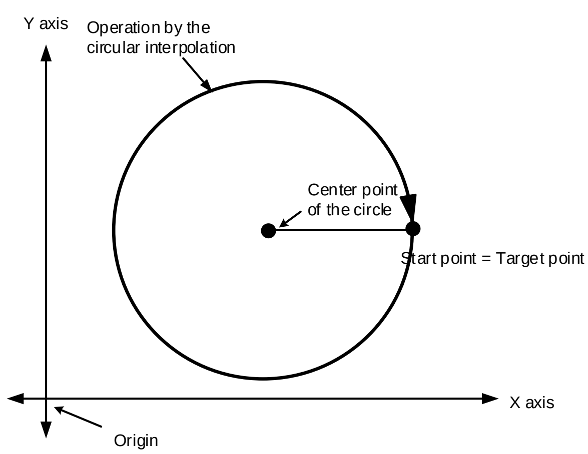
(1) Circular interpolation is performed by starting at the start position, and reaching the target position in a circular trajectory of which the diameter is the distance to the designated center point.

(2) The movement direction is determined as the direction set in the absolute position circular interpolation operation (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
	

(3) Setting the target position to be the same as the start position creates a proper circle trajectory for the circular interpolation, of which the diameter is the distance between the start position and the center point of the circle.



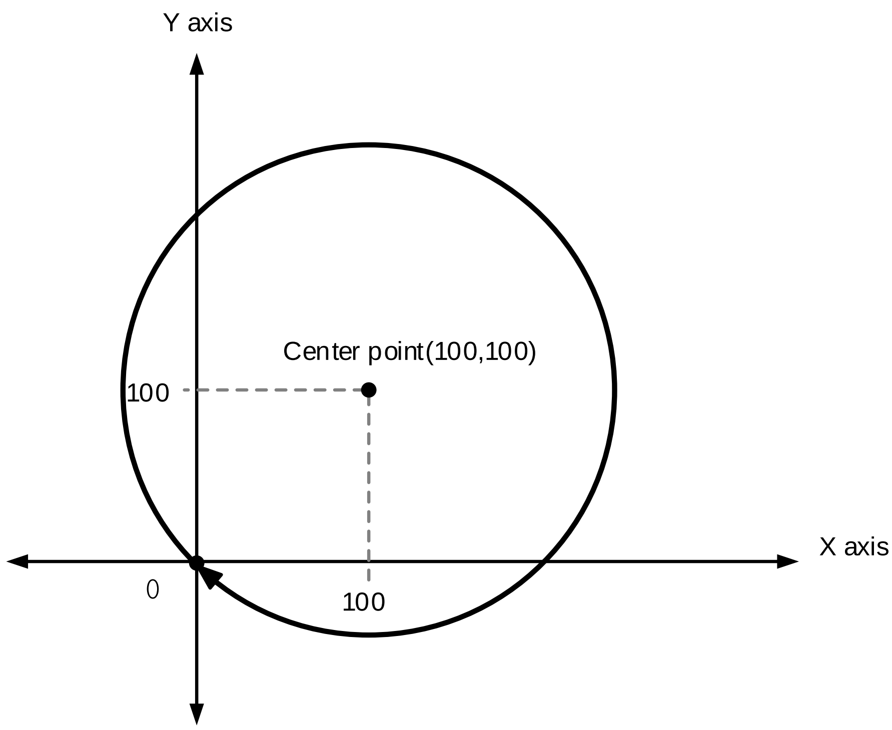
(4) Limitation

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)
- The center point set as the auxiliary point is the same as the start or target position (Error Code: 0x20A4)
- The calculated radius of the arc exceeds 2147483647pls (Error Code: 0x20A6)
- The start position, the auxiliary point, and the target position are on the same straight line (Error Code: 0x20A7)
- One or more of the component axes is performing infinite running repeat operation (Error Code: 0x20A8)

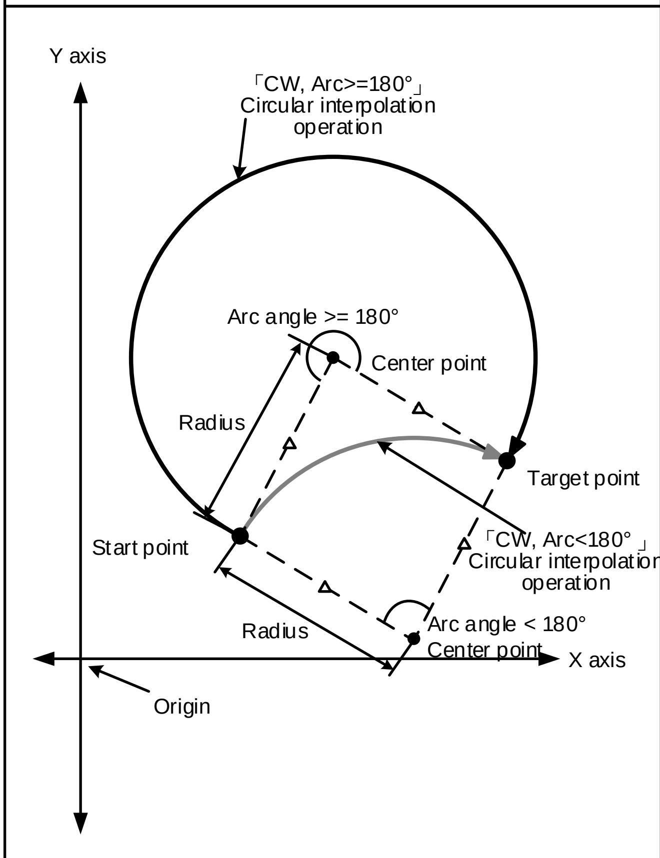
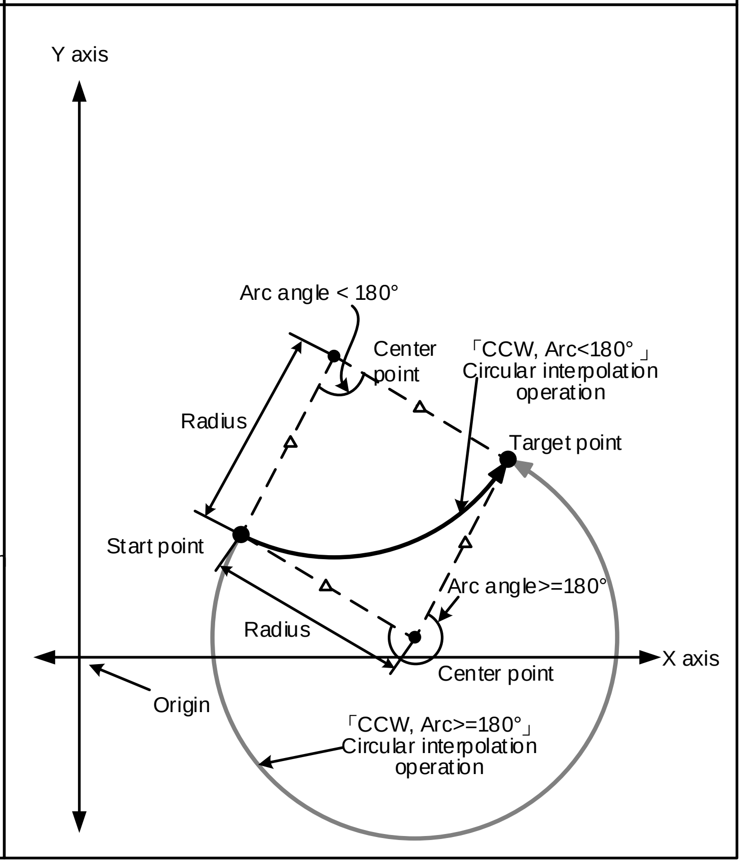
(5) Operation pattern

- Start point: (0.0, 0.0,0.0)
- Target point: (0.0, 0.0, 0.0)
- Aux point: (100.0, 100.0, 0.0)
- CircMode: Center point(1)
- PathChoice: - CW(0)



4. Circular interpolation using radius specification

(1) Circular interpolation is performed by starting at the start point, and reaching the target point in a circular traject ory of which the diameter is the distance set by the auxiliary point for the circular interpolation. The center point arc varies depending on the positivity/negativity of radius ((+): arc angle <180°, (-): arc angle>=180°).

CW, Arc<180° CW, Arc>=180°	CCW, Arc<180° CCW, Arc>=180°
	

(2) In case of designating the diameter, the target position cannot be identical to the start position.

(3) 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.

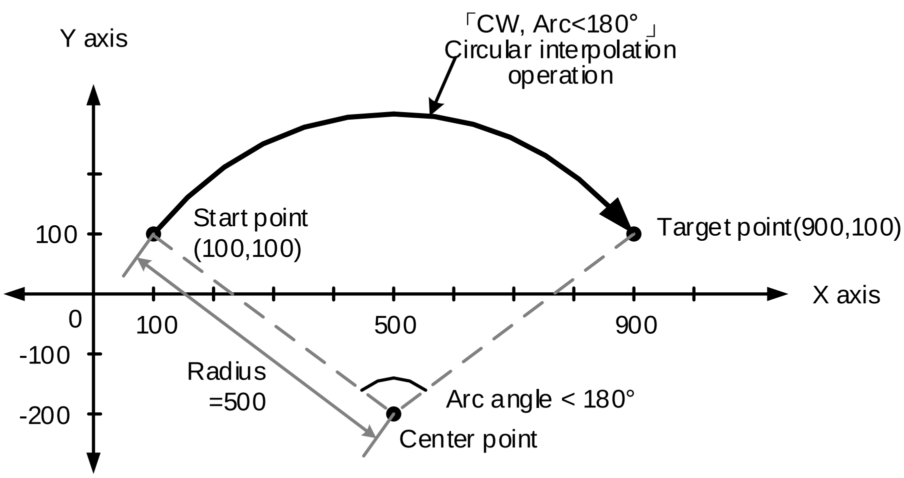
(4) Limitation

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)
- The start position is the same as the target position (Error Code: 0x20A5)
- The calculated radius of the arc exceeds 2147483647pls (Error Code: 0x20A6)
- The start position, the auxiliary point, and the target position are on the same straight line (Error Code: 0x20A7)
- One or more of the component axes is performing infinite running repeat operation (Error Code: 0x20A8)

(5) Operation pattern

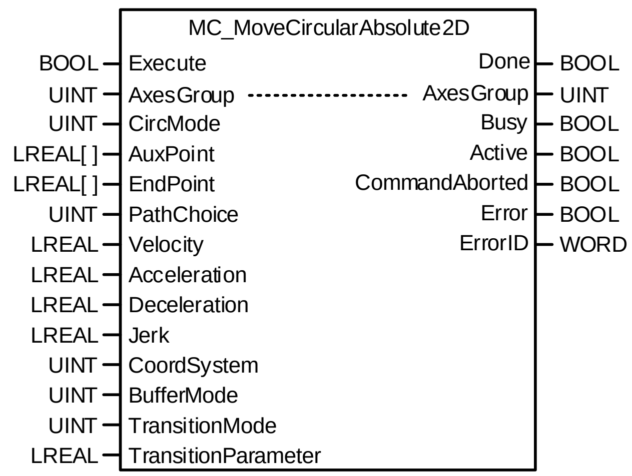
- Start point: (100.0, 100.0, 0.0)
- Target point: (900.0, 100.0)
- Aux point: (500.0, 0.0)
- CircMode: Radius(2)
- PathChoice: - CW(0)



5. Function block

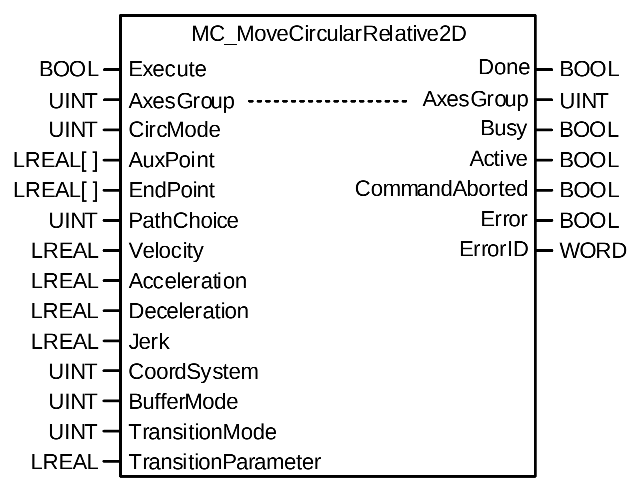
(1) Absolute position coordinate system circular interpolation operation

Name	Description	Operation Condition
MC_MoveCircularAbsolute2D	Absolute position circular interpolation operation	Edge



(2) Relative position coordinate system circular interpolation operation

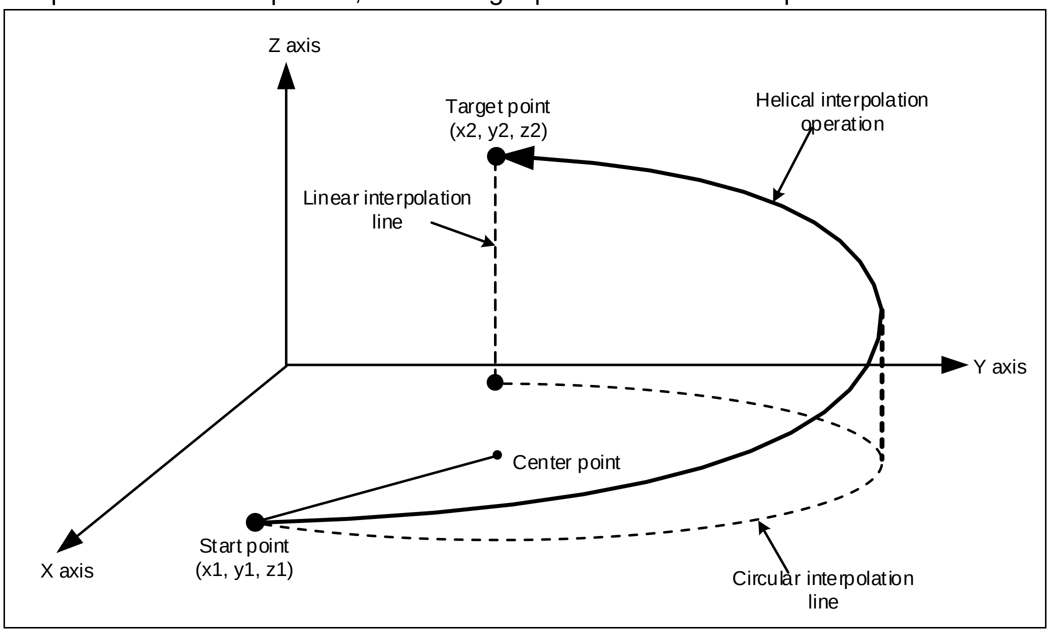
Name	Description	Operation Condition
MC_MoveCircularRelative2D	Relative position circular interpolation operation	Edge



6. Helical interpolation

(1) When circular interpolation commands (absolute position coordinate system circular interpolation operation (MC_MoveCircularAbsolute2D), relative position coordinate system circular interpolation operation (MC_MoveCircularRelative2D)) are executed, circular interpolation is performed by moving in a circular trajectory on the XY plane, while linear interpolation synchronized to the circular interpolation motion is performed with respect to Z-axis

(2) To perform helical interpolation, set the target position for linear interpolation at Pz of 'EndPoint'.

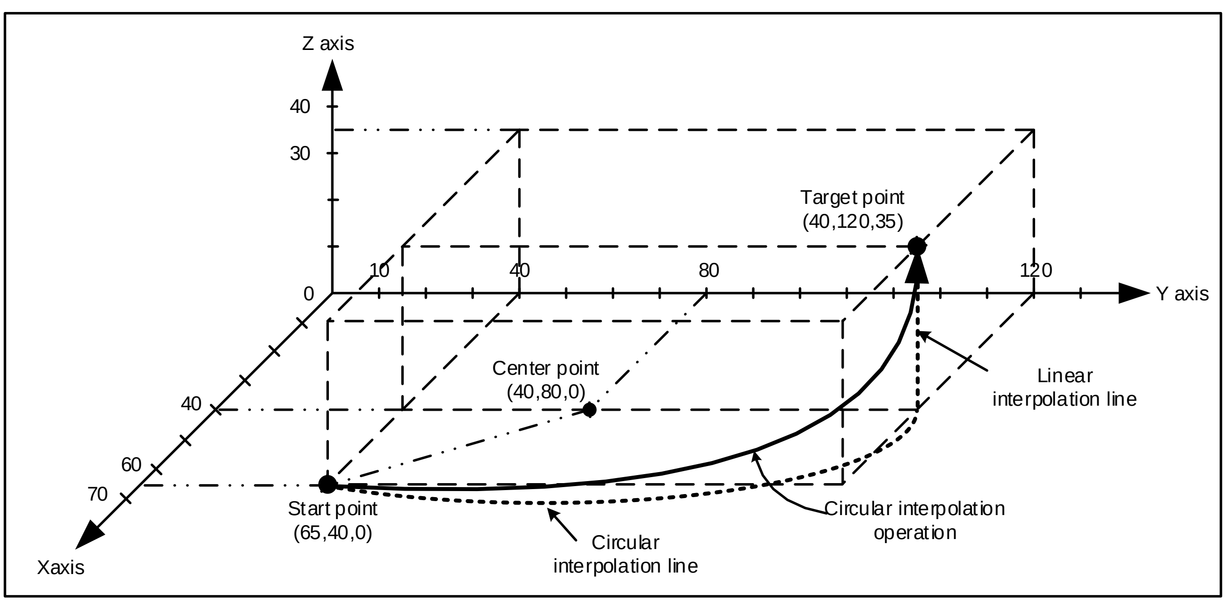


(3) Limitation

The restrictions for the circular interpolation mode designated for helical interpolation also apply to the helical interpolation.

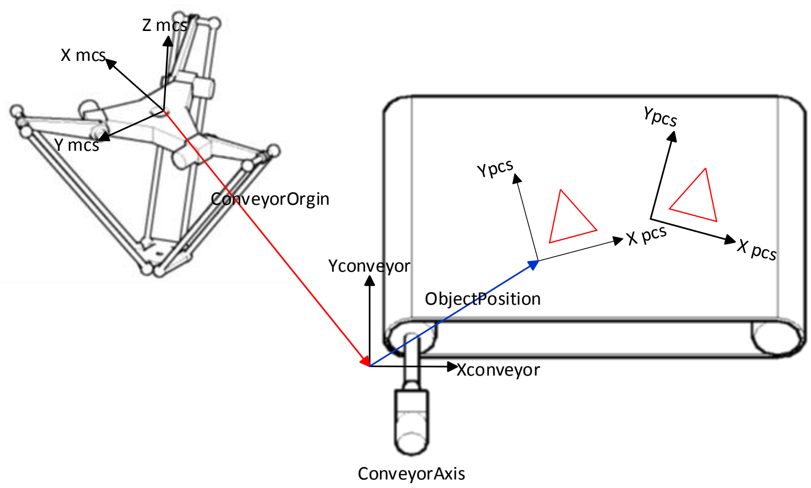
(4) Operation pattern

- Start point: (65.0, 40.0, 0.0)
- Target point: (40.0, 120.0, 35.0)
- Center point: (40.0, 80.0, 0.0)
- CircMode: Center point(1)
- PathChoice: - CCW(1)



8.4.8 Synchronized Operation for Conveyor Belt

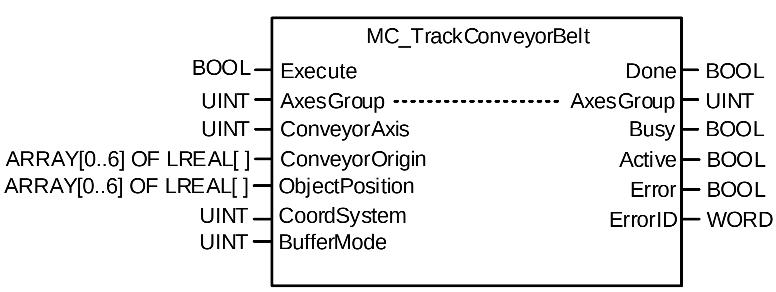
In a coordinate-based operation, one of the axes group is designated as the conveyor axis, and the objects moving on the conveyor in a straight line are tracked.



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.

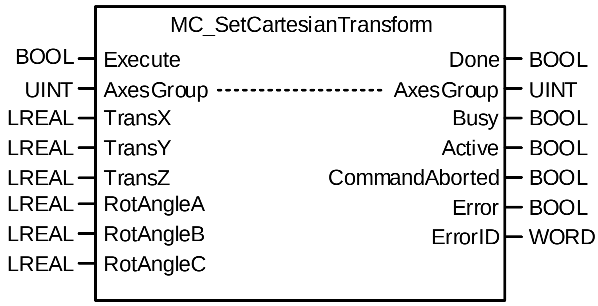
2. Function block
   (1) Conveyor belt synchronized setting

Name	Description	Operation Condition
MC_TrackConveyorBelt	Conveyor belt synchronized setting	Edge



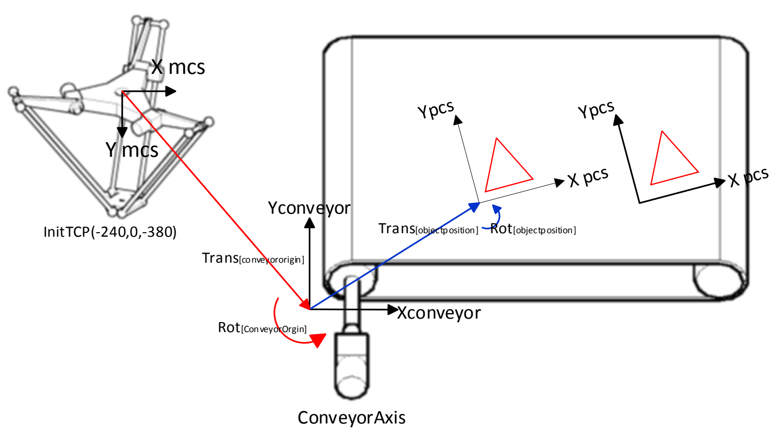
(2) Conveyor belt synchronized setting disable(PCS setting)

Name	Description	Operation Condition
MC_SetCartesianTransform	PCS setting	Edge

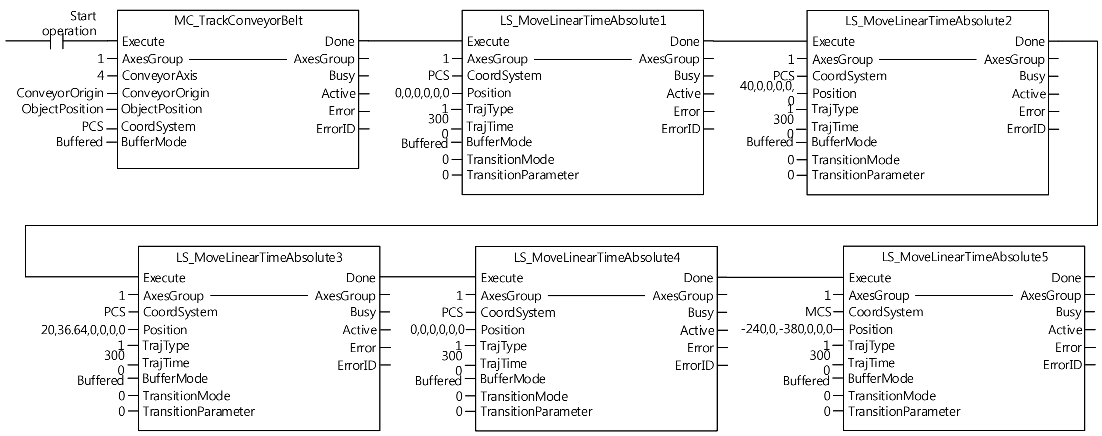


(3) Conveyor belt synchronized function operation example

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 conveyor belt position from the robot's origin point at the ConveyorAxis input, and enter the position of the product origin point from the conveyor origin point at the ConveyorOrigin input. 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.



Function Block	CoordSystem	Position[]	Description
MoveLinearTimeAbsolute1	PCS	0,0,0	Move to ConveyorOrigin
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 its initial position



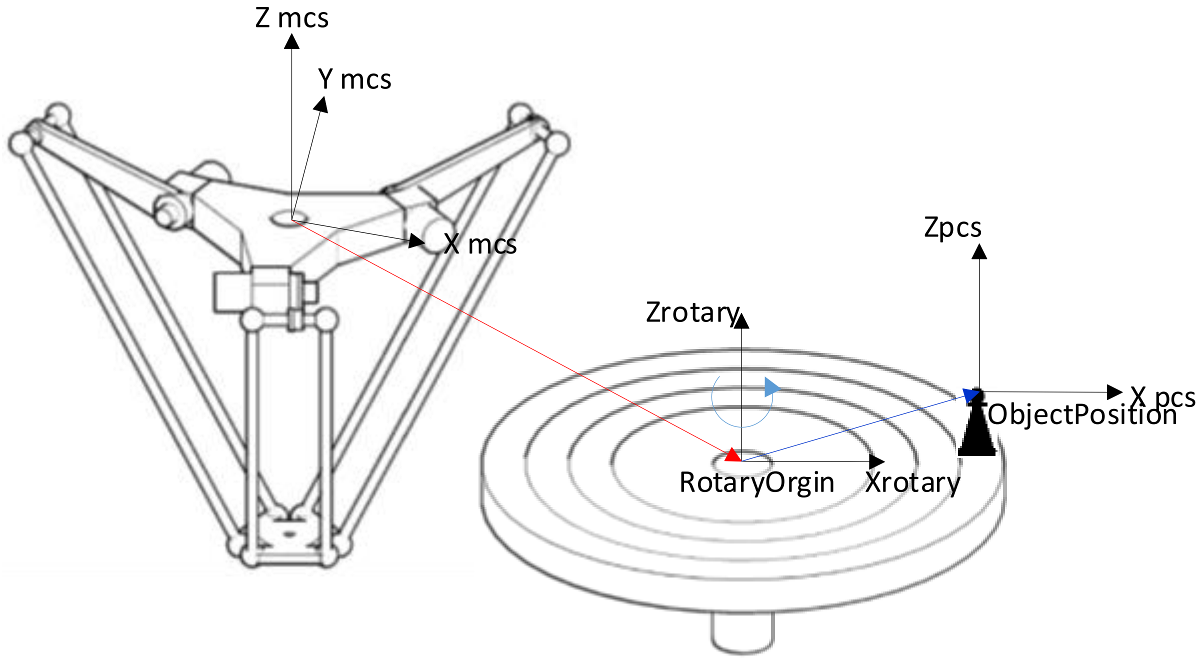
(4) Limitation

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)
- Of component axes, an axis where home position is not determined exists (Error Code 0x20B0)
- Of component axes, there is an axis being executed in infinite running repeat operation (Error Code: 0x20BA)

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.

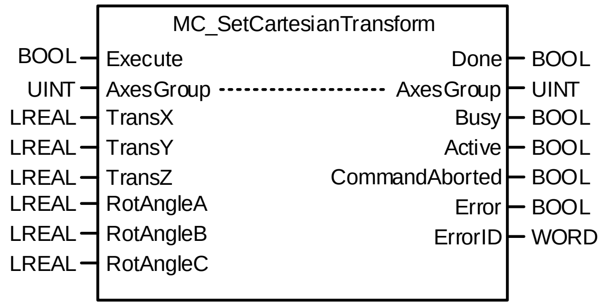
2. Function block
   (1) Rotary table synchronized setting

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..6] OF LREAL[ ] ─ RotaryOrigin                          Active ─ BOOL
  ARRAY[0..6] OF LREAL[ ] ─ ObjectPosition                         Error ─ BOOL
                         UINT ─ CoordSystem                      ErrorID ─ WORD
                         UINT ─ BufferMode
```

   (2) Rotary table synchronized setting disable (PCS setting)

Name	Description	Operation Condition
MC_SetCartesianTransform	PCS setting	Edge



(2) Rotary table synchronized function operation example

The rotary table synchronization function begins with setting rotary synchronization using MC_TrackRotaryTable function block.

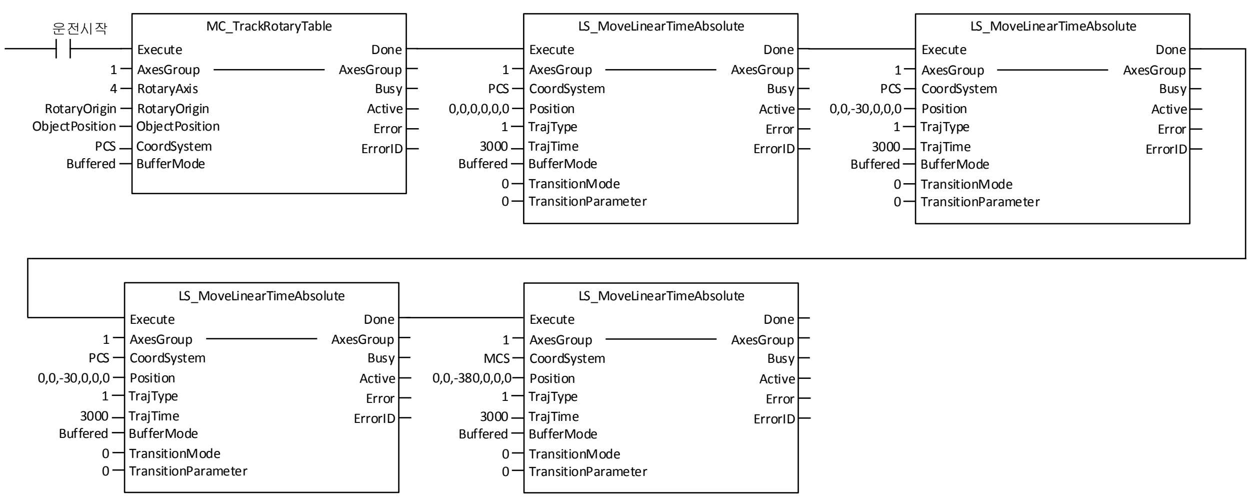
For MC_TrackRotaryTable function block, enter rotary axis value at the RotaryAxis input, enter the rotary table center position from the robot's origin point at the RotaryOrigin, enter the position of the product origin point from the rotary table center point at the ObjectPosition input.

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.

Function Block	CoordSystem	Position[]	Description
MoveLinearTimeAbsolute1	PCS	0,0,0	Move to Rotary 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



(3) Limitation

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)
- Of component axes, an axis where home position is not determined exists (Error Code 0x20B0)
- Of component axes, there is an axis being executed in infinite running repeat operation (Error Code: 0x20BA)

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

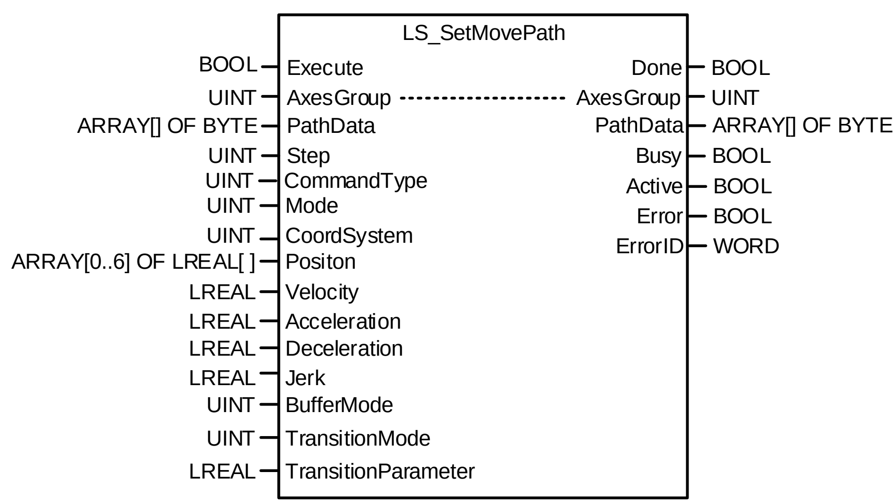
The data of the coordinate system path operation is set using the LS_SetMovePath function block. The path data set in the LS_SetMovePath is stored in the array variable specified as the PathData input. The array variable specified as input of PathData should use an array that is large enough to store 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 the CommandType, the type of command to execute the operation (0: None 1: 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) are specified. 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) Motion function block

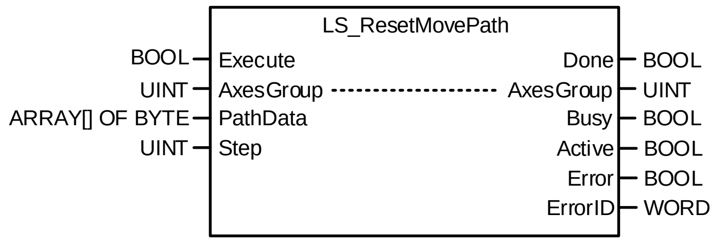
(1) Coordinate system path data setting

Name	Description	Operation condition
LS_SetMovePath	Coordinate system path data setting	Edge



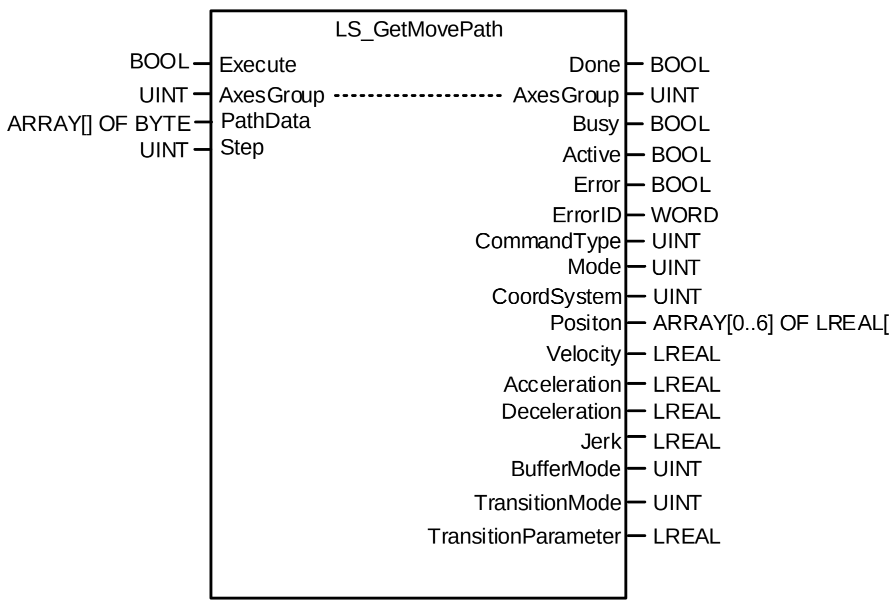
(2) Coordinate system path data remove

Name	Description	Operation condition
LS_ResetMovePath	Coordinate system path data remove	Edge



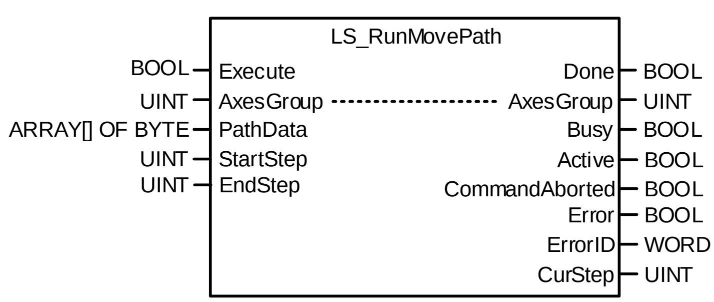
(3) Coordinate system path data read

Name	Description	Operation condition
LS_GetMovePath	Coordinate system path data read	Edge



(4) Coordinate system path operation

Name	Description	Operation condition
LS_RunMovePath	Coordinate system path operaton	Edge



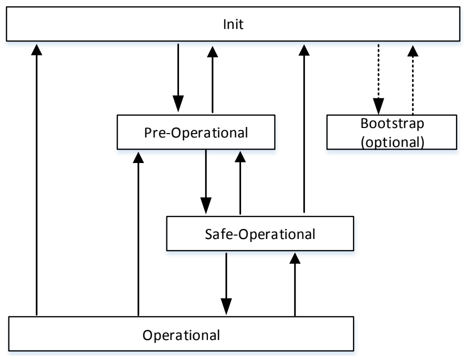
8.5 FoE(File Access over EtherCAT) Function

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.

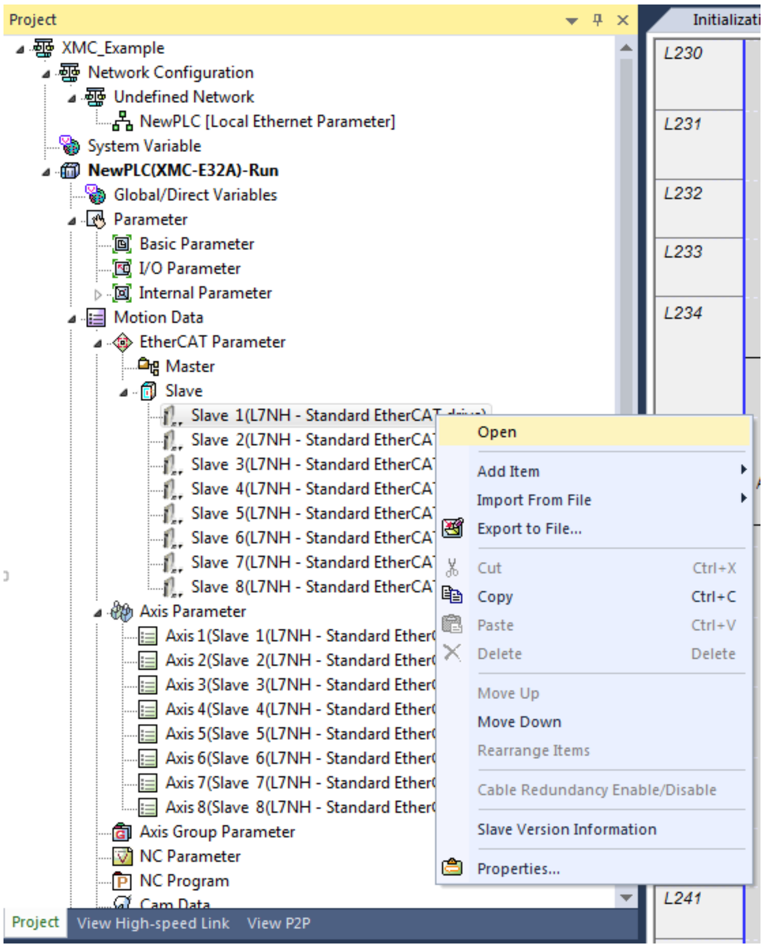
8.5.2 FoE Download

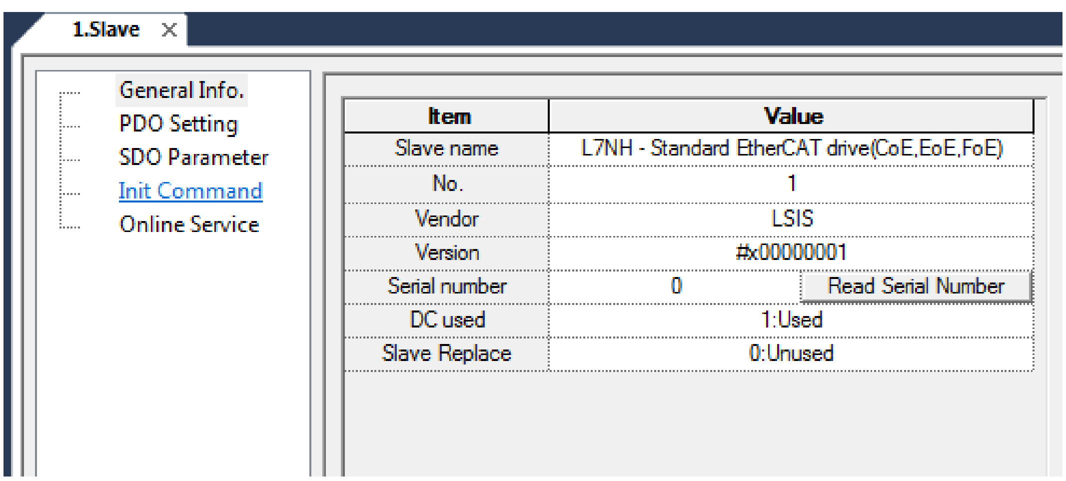
The slave operates as a state machine depending on the functions it actually supports. Since the FoE function is supported only in Boot (Bootstrap) mode, which is the bootloader state, the salve state should be converted to the Boot mode in order to use the FoE function. If the motion controller performs 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.

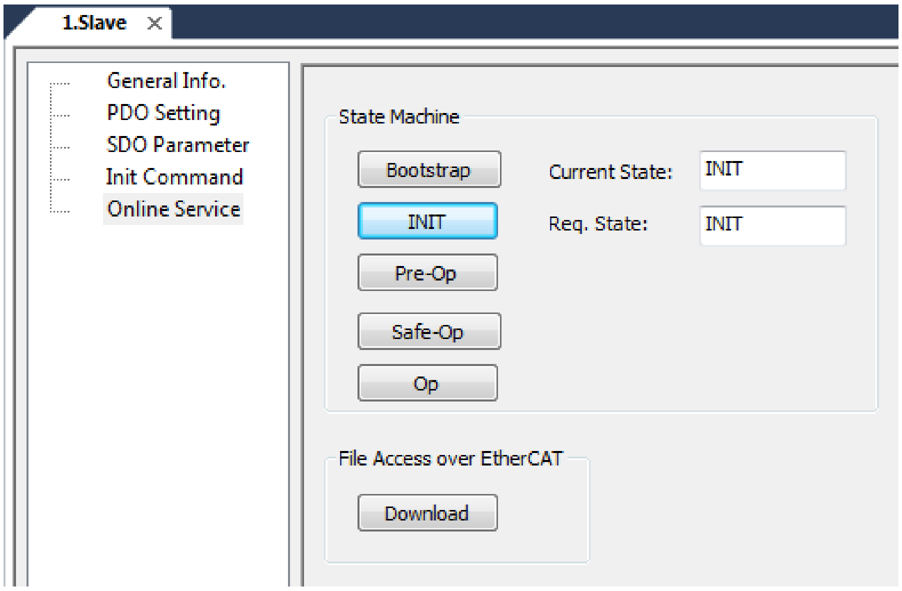

Project
XMC_Example
Network Configuration
Undefined Network
NewPLC [Local Ethernet Parameter]
System Variable
NewPLC(XMC-E32A)-Run
Global/Direct Variables
Parameter
Basic Parameter
I/O Parameter
Internal Parameter
Motion Data
EtherCAT Parameter
Master
Slave
Slave 1(L7NH - Standard EtherCAT drive)
Slave 2(L7NH - Standard EtherCAT
Slave 3(L7NH - Standard EtherCAT
Slave 4(L7NH - Standard EtherCAT
Slave 5(L7NH - Standard EtherCAT
Slave 6(L7NH - Standard EtherCAT
Slave 7(L7NH - Standard EtherCAT
Slave 8(L7NH - Standard EtherCAT
Axis Parameter
Axis 1(Slave 1(L7NH - Standard Ether
Axis 2(Slave 2(L7NH - Standard Ether
Axis 3(Slave 3(L7NH - Standard Ether
Axis 4(Slave 4(L7NH - Standard Ether
Axis 5(Slave 5(L7NH - Standard Ether
Axis 6(Slave 6(L7NH - Standard Ether
Axis 7(Slave 7(L7NH - Standard Ether
Axis 8(Slave 8(L7NH - Standard Ether
Axis Group Parameter
NC Parameter
NC Program
Cam Data
Project
View High-speed Link
View P2P
Open
Add Item
Import From File
Export to File...
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Delete Delete
Move Up
Move Down
Rearrange Items
Cable Redundancy Enable/Disable
Slave Version Information
Properties...
Initializati
L230
L231
L232
L233
L234
L241


1.Slave
General Info.
PDO Setting
SDO Parameter
Init Command
Online Service
Item
Value
Slave name
L7NH - Standard EtherCAT drive(CoE,EoE,FoE)
No.
1
Vendor
LSIS
Version
#x00000001
Serial number
0
Read Serial Number
DC used
1:Used
Slave Replace
0:Unused

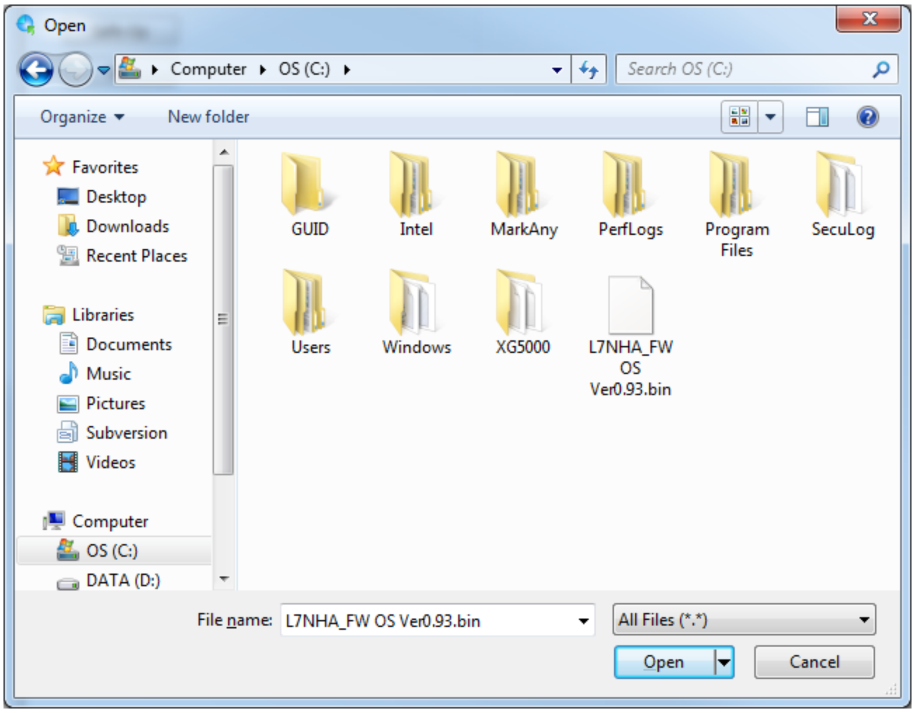
(2) Downloading files

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. Please refer to the salve instruction manual.

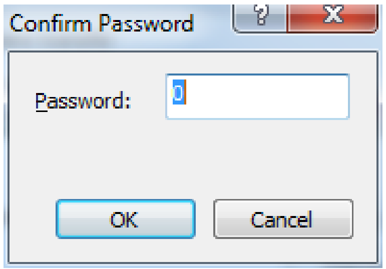
1) Change OpMode to Boot and click the Download button in the FoE to start the download.



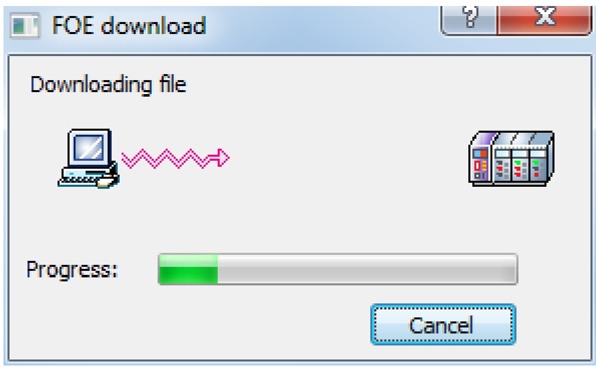
2) Select the file you want to download from the open dialog box.



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.

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 Command

9.1.1 Definition of the NC Command

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.

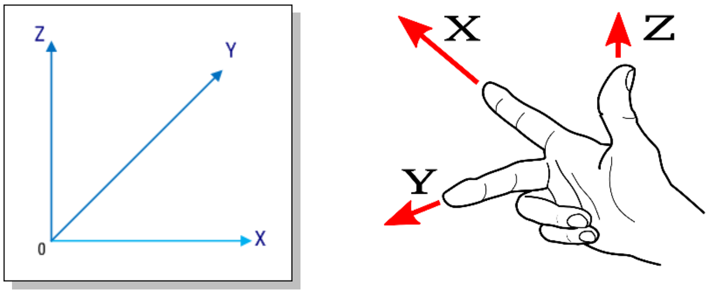
Reserved character	Description
Character set	
(	Start comment
)	End comment
[	Left parenthesis
]	Right parenthesis
+,-,*,/	Four fundamental arithmetic operations
A(AND), O(OR)	Comparison operation
=	Assignment operation
0~9	Numeric data
;	Block end
#	Variable
Address character	
X	X axis of the XYZ rectangular 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)

Reserved character	Description
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)
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
%, ;, ( )	Comment
LE,GE,EQ,LT,GT,NE	Compare instruction

Reserved character	Description
SIN, COS, TAN, ATAN, SQRT, ABS, ROUND, AND, OR, RIX, 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.

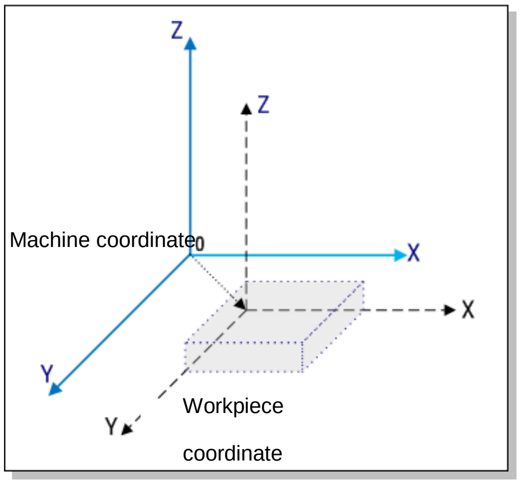
For more details on setting the machine coordinate system, refer to the G53 command description.

(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]

9.2 Configuration of the Program

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) Subprogram

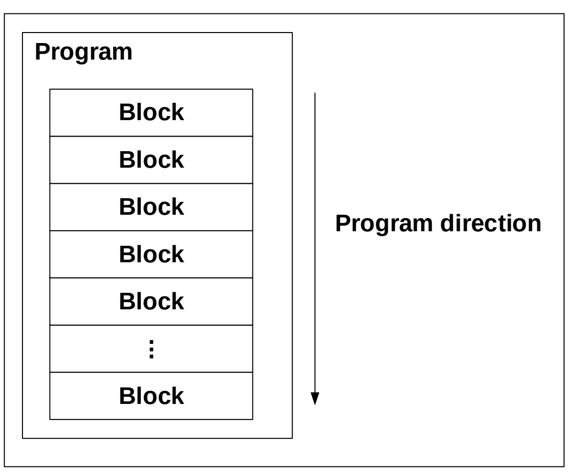
The subprogram is executed by the main program’s call, and the only 4-digit numbers can be applied to the program file name. (0000 to 9999) The extension is the "nc" which is the same as the main program. The subprogram can be called directly from the main program so it must be written in numeric names only and distinguished by them.

9.2.2 Configuration of the NC Program

(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	Preparatory 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.

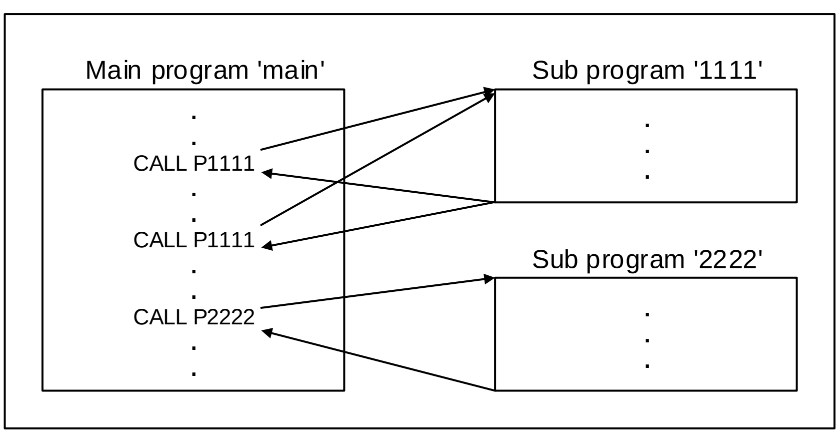
At the head of each block, there is a Statement Number which contains the motion control's sequence information. The Statement Number is often referred in the GOTO and CALL instructions, and does not have to be used unless it is necessary. The Statement Number can also be written alone in a block.

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

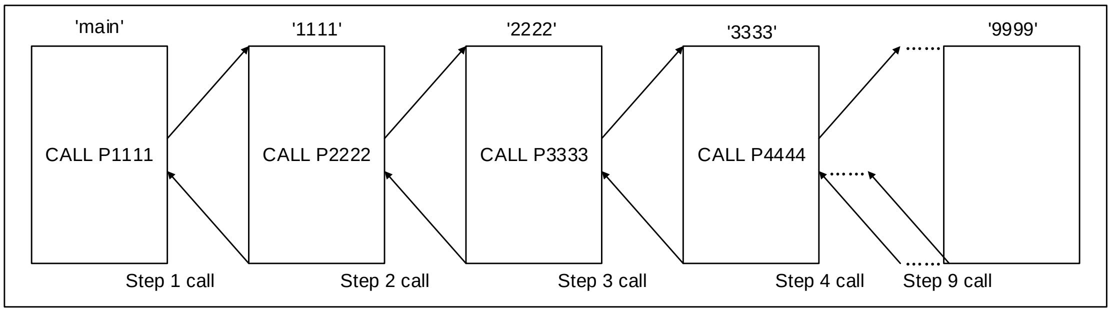
Basically, you can configure and control the NC program with the main program only. However, if a series of identical commands or control intervals are repeated, you can make such parts into the subprogram and call it in case of necessity. When the subprogram is called, the details of the subprogram 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) System of units

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
```

9.3 NC Command

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 positioning (X, Y, Z, A, B, C, U, V, W, S)

```
X_ Y_ Z_ A_ B_ C_ U_ V_ W_ S_
```

X_ Y_ Z_ ~ S_: Commanding the positions of axes

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) Specifying the central point of an 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 List of the NC Commands

The NC commands (G / M code and other commands) used for the motion controller are as follows.

Category	Program instruction	Function
G code instruction	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
	G17	Select the circular interpolation plane (XY plane)
	G18	Select the circular interpolation plane (ZX plane)
	G19	Select the circular interpolation plane (YZ plane)
	G21	Metric unit 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 and 3rd homing
	G40	Cancel compensation of tool diameter

Category	Program instruction	Function
G code instruction	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 +
	G49	Cancel compensation of the tool length
	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	Selecting the workpiece coordinate system 4
	G58	Selecting the workpiece coordinate system 5
	G59	Selecting the workpiece coordinate system 6
	G60	Single direction positioning
	G90	Absolute command
	G91	Incremental command
	G92	Set the workpiece coordinate system, the max. speed of the main axis
	G94	Feed mode command per minute
	G95	Feed mode command per revolution
	G107	Cylindrical interpolation mode setting
	G112	Interpolation mode of the polar coordinate ON
	G113	Interpolation mode of the polar coordinates OFF
M code	M00	Program stop
	M01	Optional stop
	M02	PROGRAM END
	M03	Forward rotation of the main axis
	M04	Reverse rotation of the main axis
	M05	Main axis stop
	M06	Tool change
	M08	Coolant ON
	M09	Coolant OFF
	M30	End of the program
	M98	Auxiliary program call
	M99	End of the auxiliary program

Category	Program instruction	Function
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	Specify the dwelling 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	End of loop
	%, ;	Comment processing command
	LE, GE, EQ, LT, GT, NE	Compare instruction
	AND, OR, XOR, +, -, *, /	Operation instruction
	=	Assignment operator
	SIN, COS, TAN, ATAN, SQRT, ABS, ROUND, AND, OR, RIX, 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.

Category	Description
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 instruction	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.

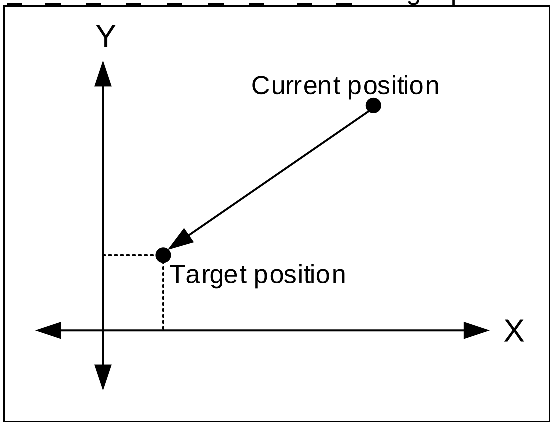
1) Rapid traverse (G00)

```
(G90, G91) G00 X_ Y_ Z_ A_ B_ C_ U_ V_ W_ S_
```

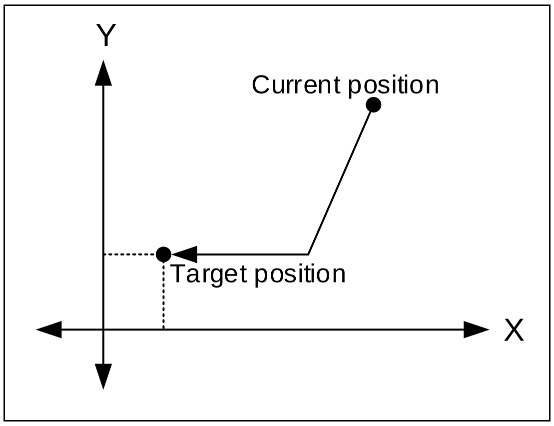
G90, G91: Absolute/Incremental command

G00: Rapid positioning control command

X_ Y_ Z_ A_ B_ C_ U_ V_ W_ S_: Target position to traverse



[Current position and the target point to traverse]



[Shape of the traverse 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 traverse	Rapid traverse acceleration
		Rapid traverse deceleration
		Rapid traverse jerk
		Rapid traverse speed

<table>
<tr><th>NC parameter</th><th>Group and parameter name</th></tr>
<tr><td>NC channel/axis parameter</td><td>
<table>
<tr><th>Group</th><th>Name</th><th>X Axis</th></tr>
<tr><td rowspan="4">Rapid Traverse Settings</td><td>Rapid traverse acceleration</td><td>500 mm/s2</td></tr>
<tr><td>Rapid traverse deceleration</td><td>500 mm/s2</td></tr>
<tr><td>Rapid traverse jerk</td><td>0 mm/s3</td></tr>
<tr><td>Rapid traverse speed</td><td>10000 mm/m</td></tr>
</table>
</td></tr>
</table>

2) Linear interpolation (G01)

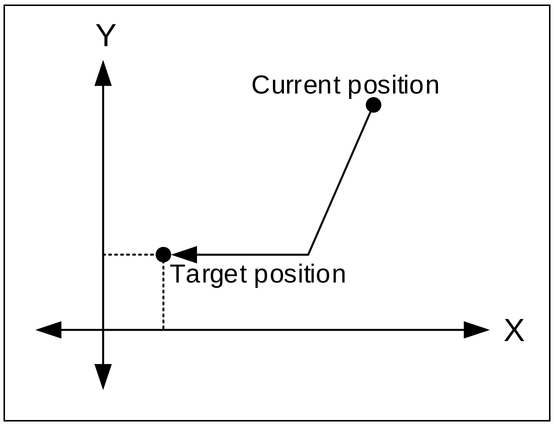
```
(G90, G91) G01 X_ Y_ Z_ A_ B_ C_ U_ V_ W_ S_ (F_)
```

G90, G91: Absolute/Incremental command

G01: Interpolation feed control command

X_ Y_ Z_ A_ B_ C_ U_ V_ W_ S_: Target position to traverse

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.

For "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 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 circular interpolation

G02, G03: Clockwise, counter clockwise circular interpolation

X_ Y_ Z_: Target position to traverse

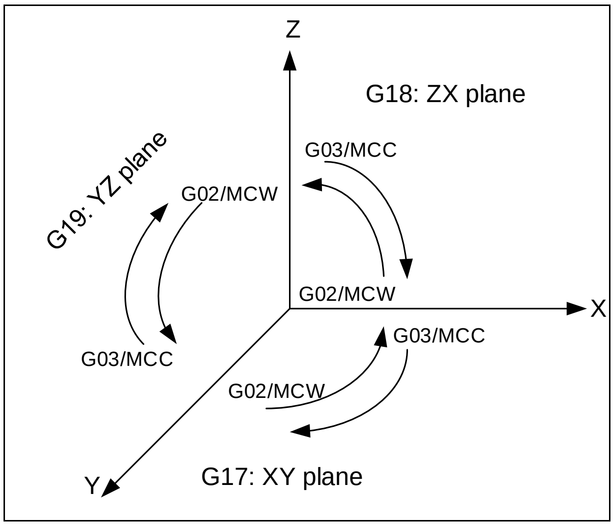
I_ J_ K_/R_: Reference point or radius of an arc

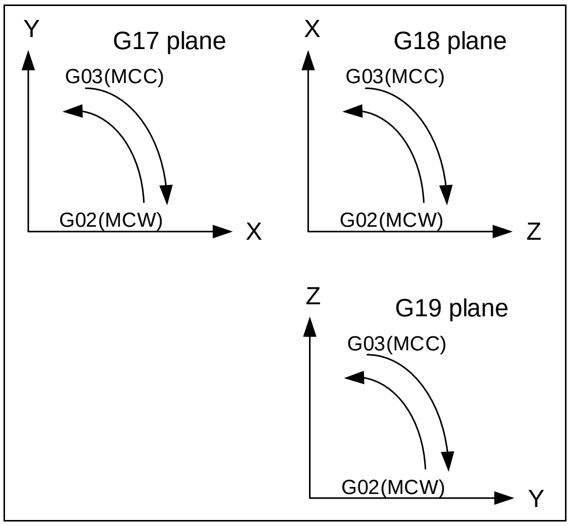
F_: Feed rate

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.

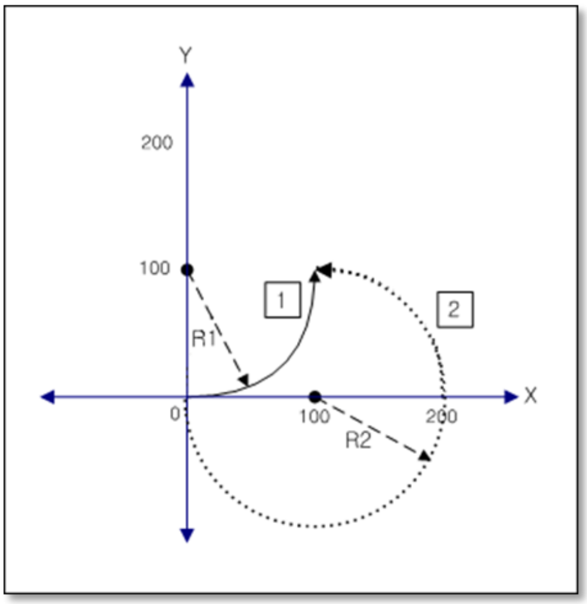
When the current position and the target position to traverse are the same, the circular interpolation with specifying the central point can command a 360-degree perfect circle.

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.

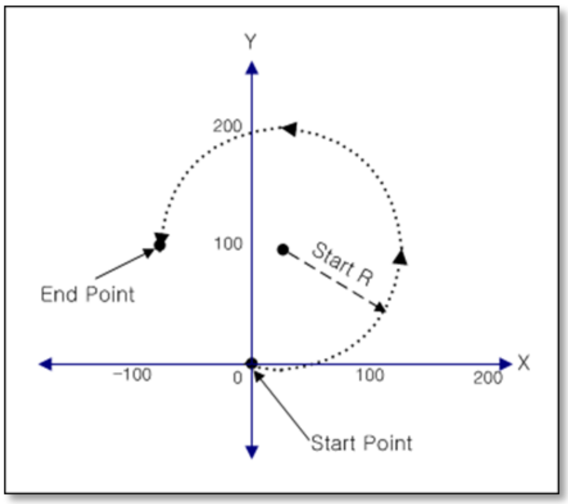
Unlike the circular interpolation with specifying the central point, the circular interpolation with the R (radius) designation cannot command a 360-degree perfect circle.

\



[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

% 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 name 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

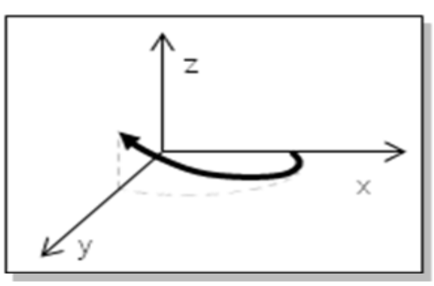
G02, G03: Clockwise, counter clockwise circular interpolation

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)

```
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.

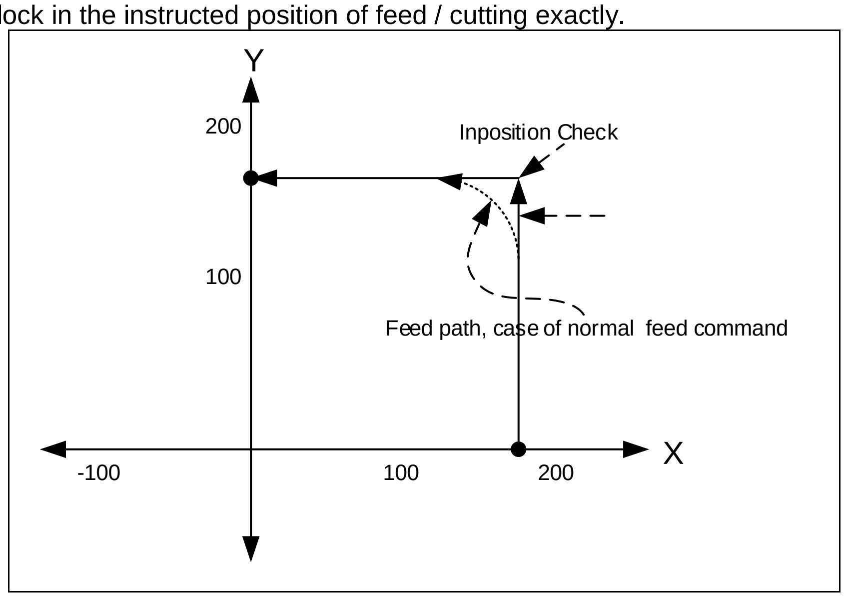
NC parameter	Group name and parameter name		
	Group	Name	Channel 1
NC channel parameter	Basic Settings	Target Machining Quantity	0
		Target machining quantity at M99 repetition	0
		Check of decimal point	1: Unused
		Keep workpiece coordinate system	0: Keep
		Macro call on T-code command	0: Do not call
		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

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) 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

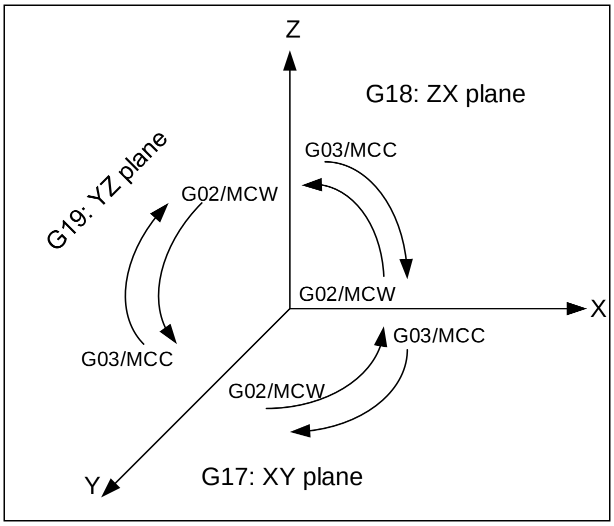
G02, G03: clockwise, counter clockwise circular interpolation

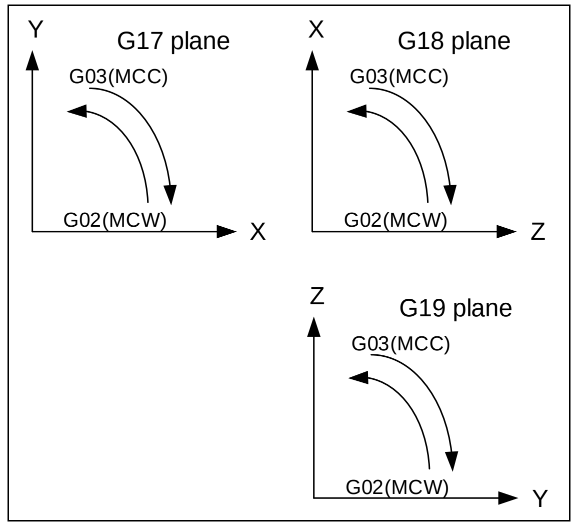
X_ Y_ Z_: Target position to traverse

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 name and parameter name		
	Group	Name	Channel 1
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 G-code for Limit check	0: G22

8) Metric unit input (G21)

G21

G21: Metric unit input

This command sets the unit of position to be entered in the future as a meter.

Since the motion controller currently support only the metric unit, the default position unit is the metric unit if the G21 is not used.

9) Enable/Disable stroke function (G22, G23)

G22 (X_ Y_ Z_) (I_ J_ K_) G23

G22: Stroke check function On

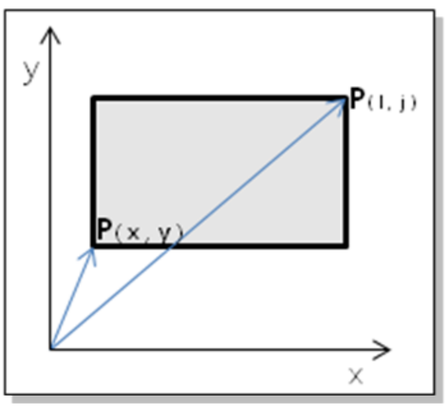
G23: Stroke check function Off

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 1 to use the G22 command.



The parameters related to Enable /Disable stroke are as follows.

NC parameters	Group name and parameter name		
	Group	Name	Channel 1
NC channel parameters	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

10) Homing check (G27)

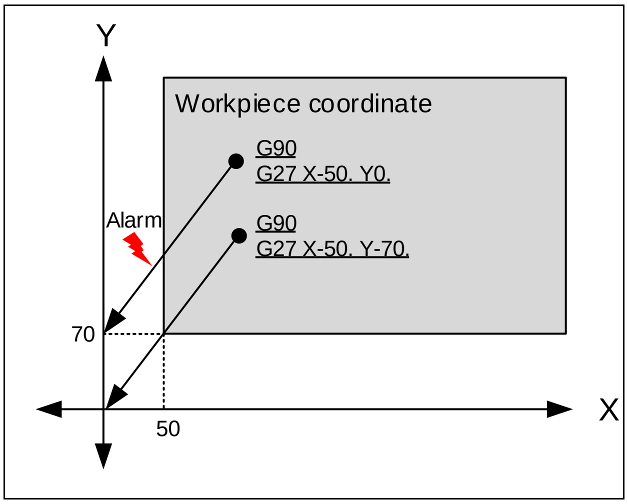
(G90, G91) G27 X_ Y_ Z_ A_ B_ C_ U_ V_ W_ S_

G90, G91: Absolute/Incremental command

G27: Homing check

X_ Y_ Z_ A_ B_ C_ U_ V_ W_ S_: Target coordinates

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.



11) Auto-homing (G28)

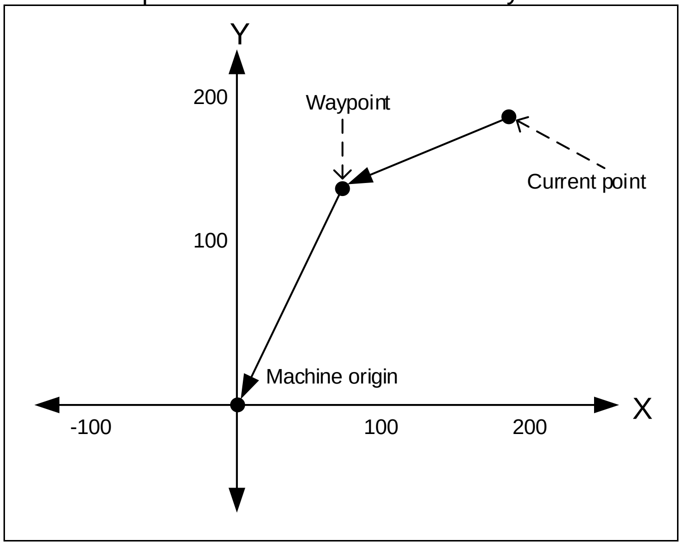
(G90, G91) G28 X_ Y_ Z_ A_ B_ C_ U_ V_ W_ S_

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.

12) Return from the auto-origin (G29)

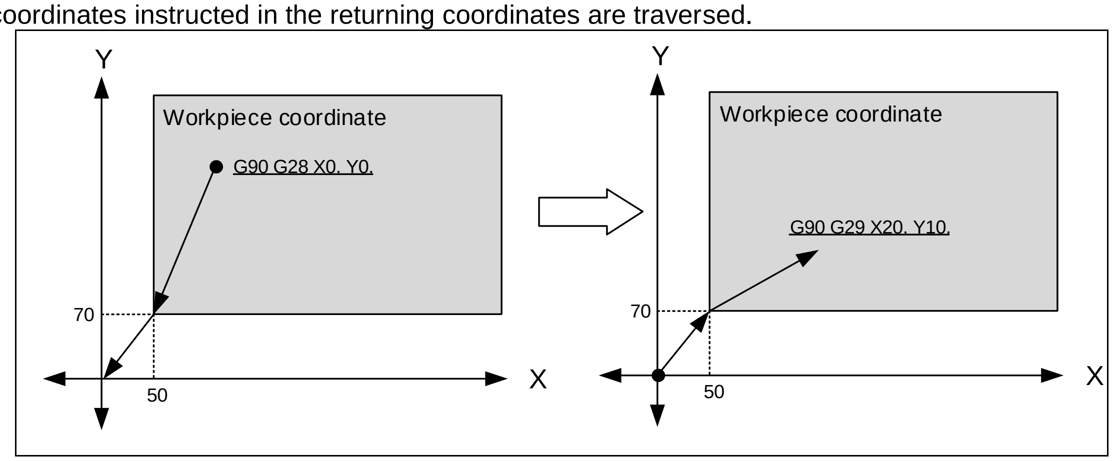
```
(G90, G91) G29 X_ Y_ Z_ A_ B_ C_ U_ V_ W_ S_
```

G90, G91: Absolute/Incremental command
G29: Return command from the origin
X_ Y_ Z_ A_ B_ C_ U_ V_ W_ S_: Returning 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.

Up to the midpoint, all axes that were previously commanded during homing are transferred, and after that, only the coordinates instructed in the returning coordinates are traversed.

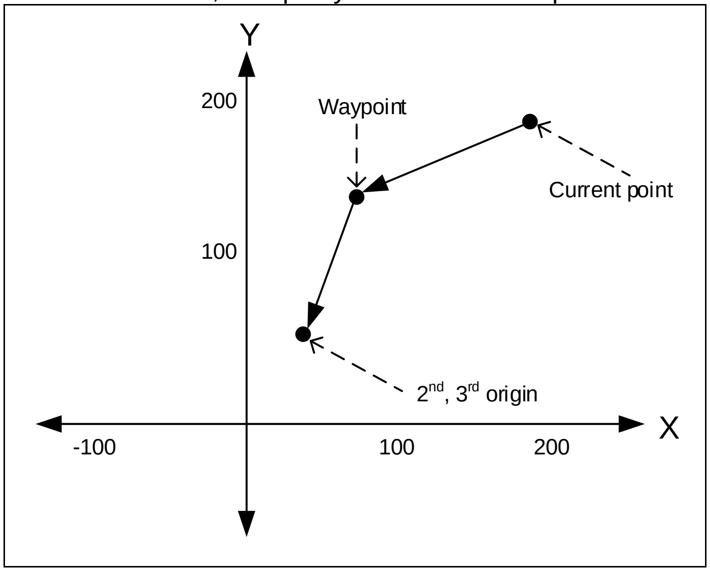


13) 2nd, 3rd, 4th homing (G30)

(G90, G91) G30 (P2, P3, P4) X_ Y_ Z_ U_

G90, G91: Absolute/Incremental command
G30: Auto-homing command
P2: 2nd origin
P3: 3rd origin
P4: 4th origin
X_ Y_ Z_ A_ B_ C_ U_ V_ W_ S_: Coordinate of the 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 parameters	Group name and parameter name		
	Group	Name	X Axis
NC channel/axis parameters	Home Settings	Position of 2nd home	0 mm
		Position of 3rd home	0 mm
		Position of 4rd home	0 mm

14) 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_

15) 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

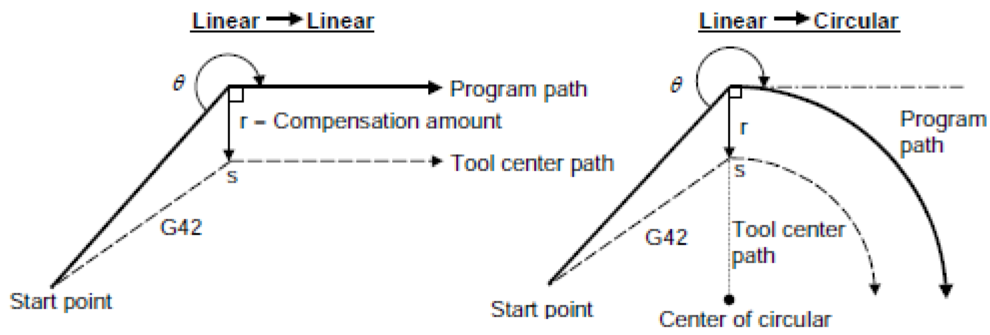
X_ Y_ Z_ U_: Instruct the vector of the next command block

(1) Start – Up mode

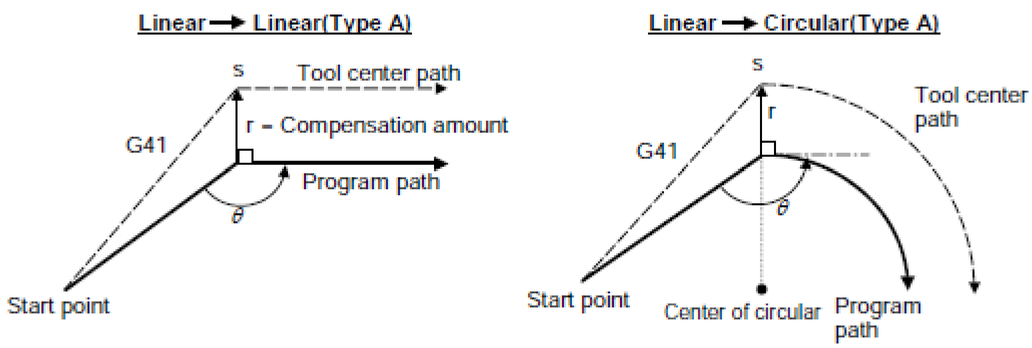
When the tool diameter compensation is started by commanding G41 / G42 in the status of Disable Tool Diameter Compensation, it is called the Start-Up mode.

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.

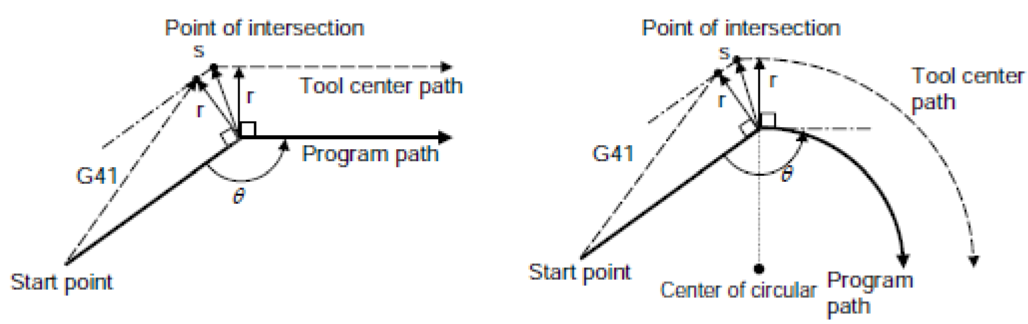
In the Start-Up block, the axis motion command must be greater than the tool radius. In the start-up or cancellation mode, the arc command [G02 / G03] is not executed. When such commands are made, an alarm occurs.



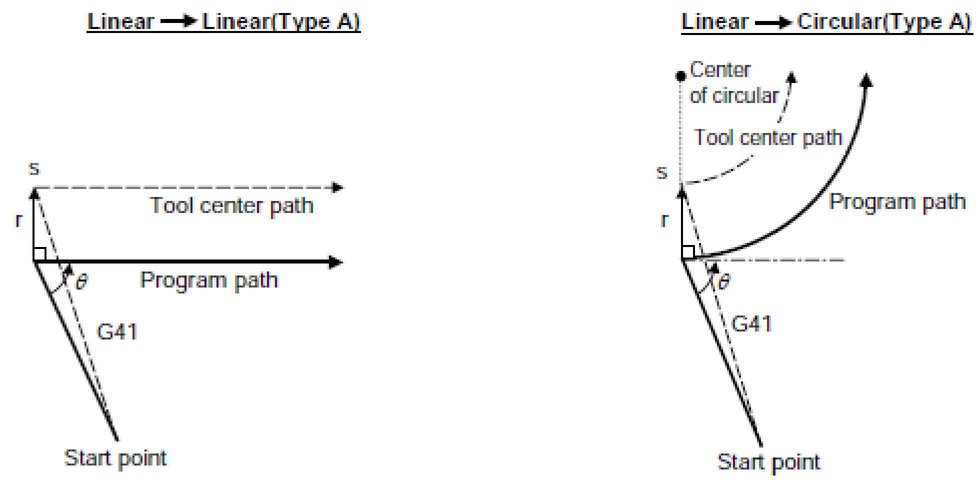
[Tool path of the inner corner]



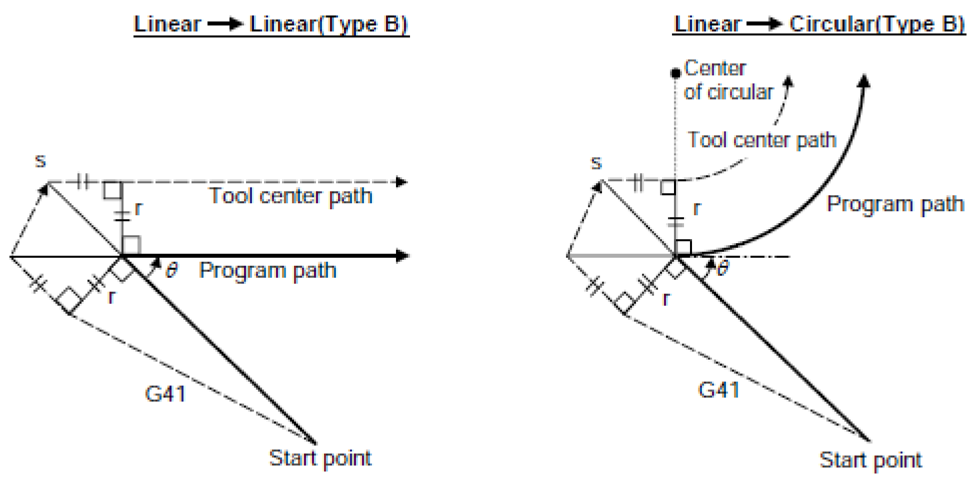
[Tool path of the outer corner (obtuse angle)] (Type A)



[Tool path of the outer corner (obtuse angle)] (Type B)

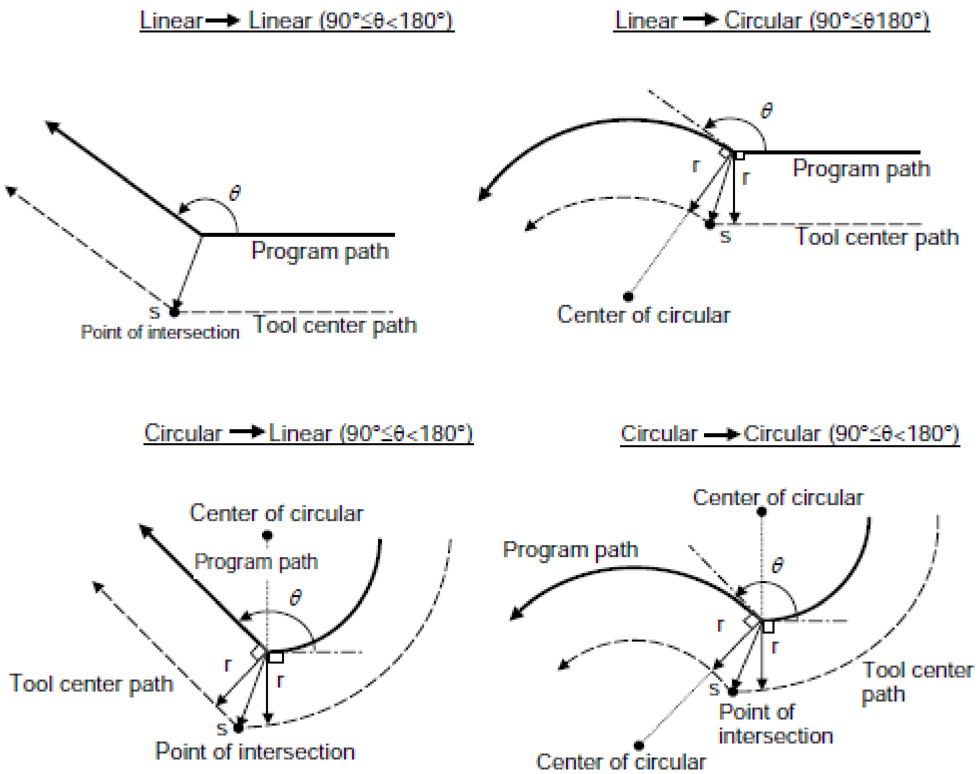


[Tool path of the outer corner (acute angle)] (Type A)

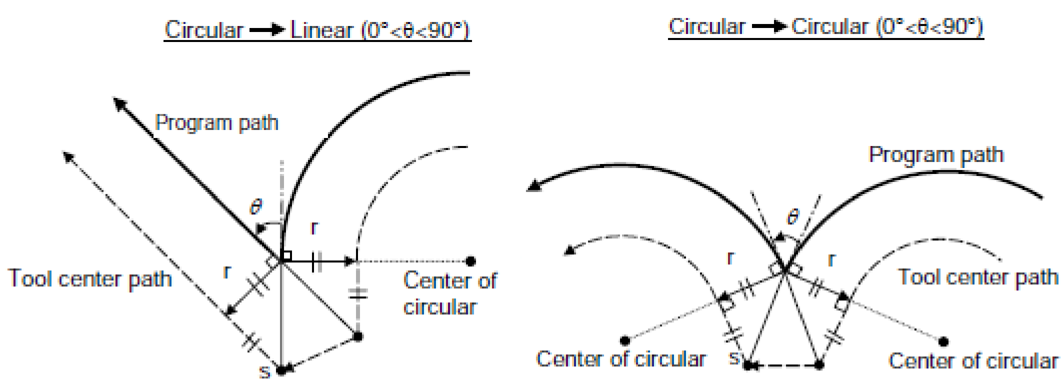


[Tool path of the outer corner (acute angle)] (Type B)

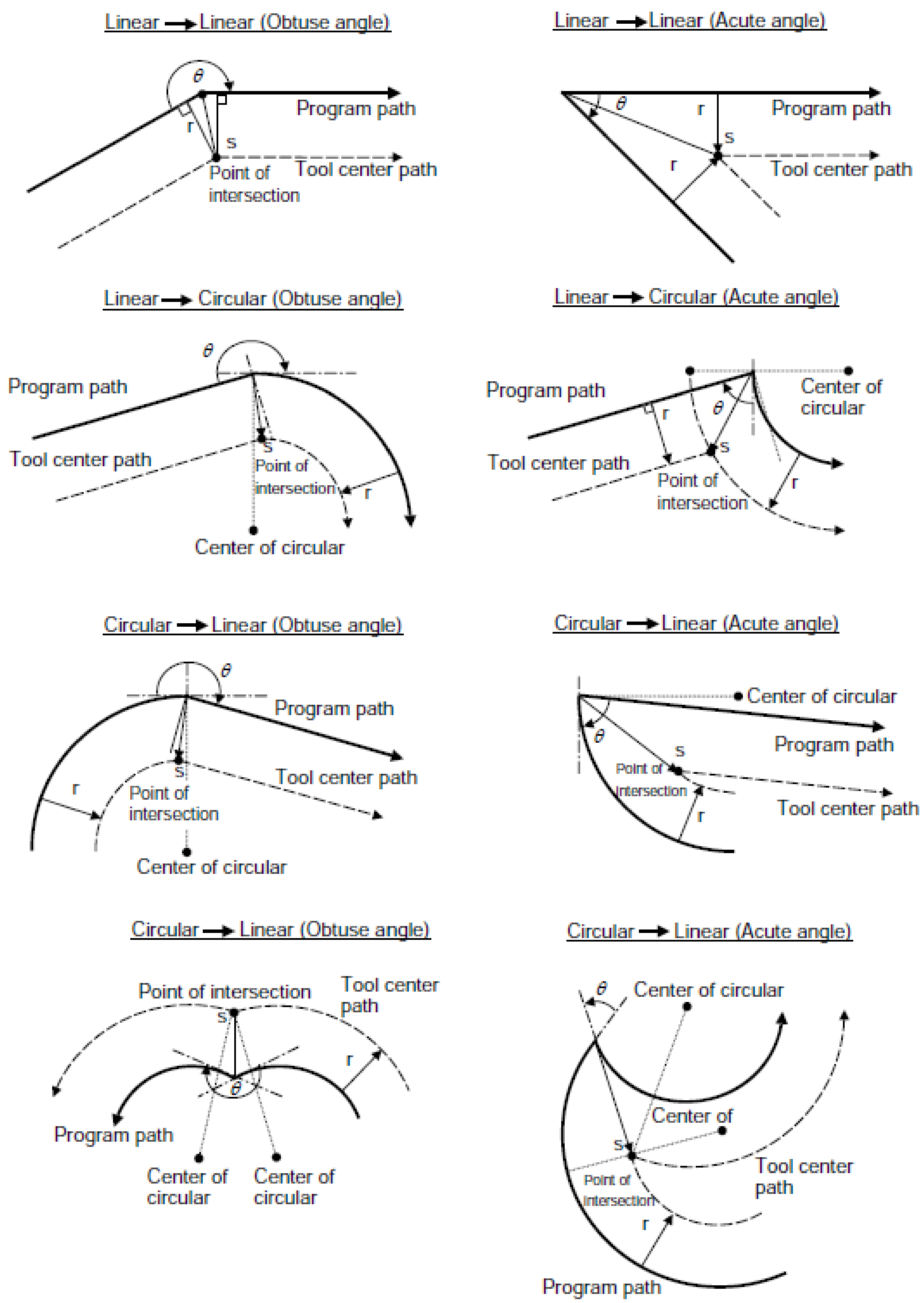
(2) Compensation mode



[Outer wall machining (obtuse angle)]

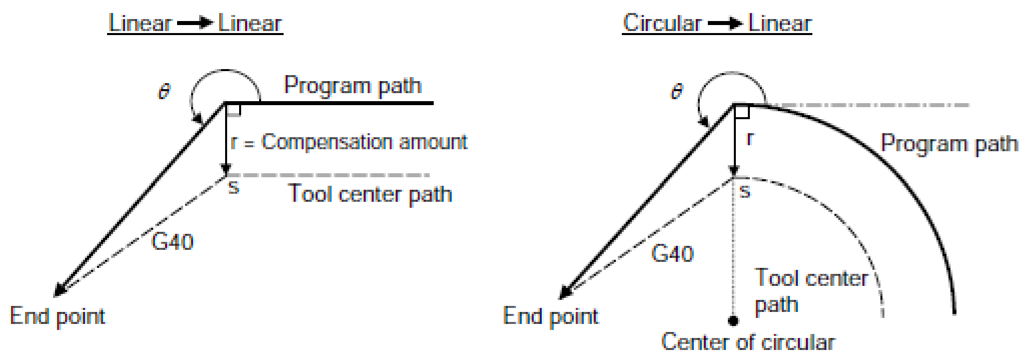


[Outer wall machining (acute angle)]

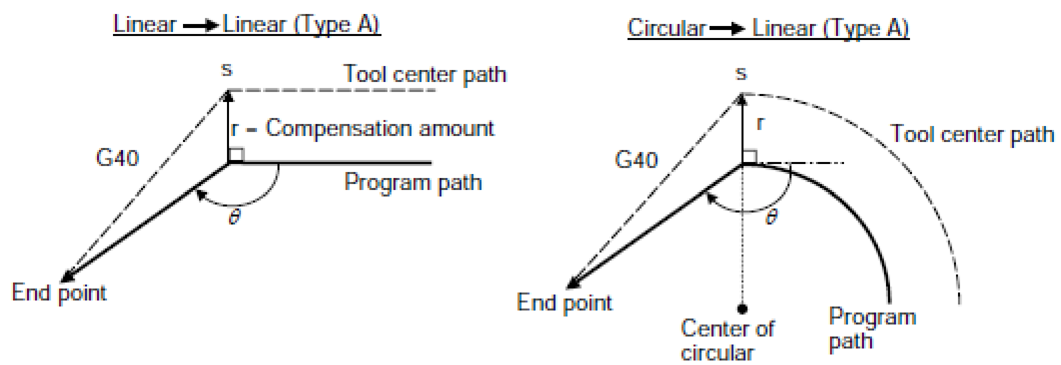


[Inner wall machining]

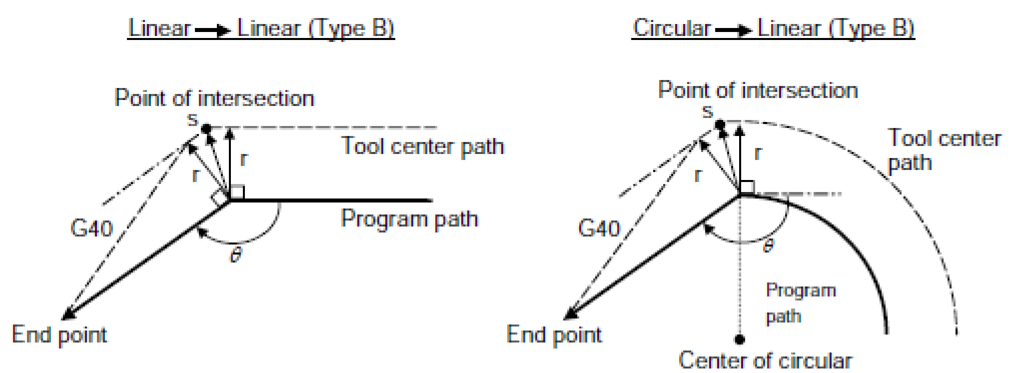
(3) Cancel mode



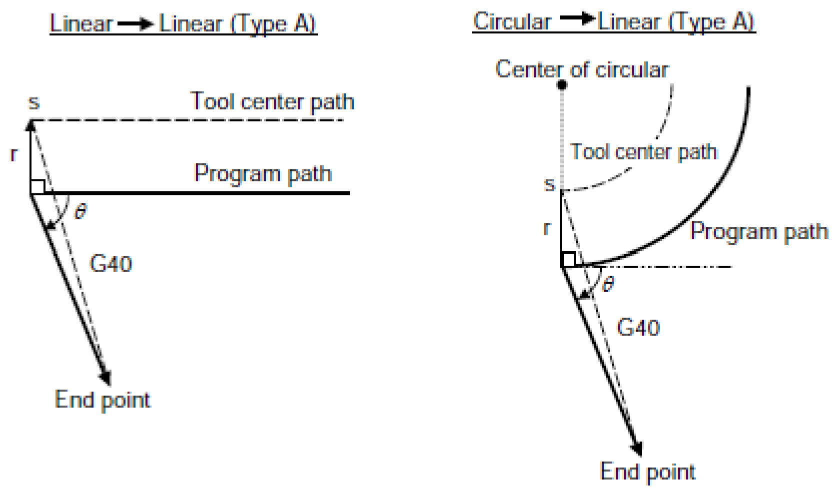
[Tool path of the inner corner]



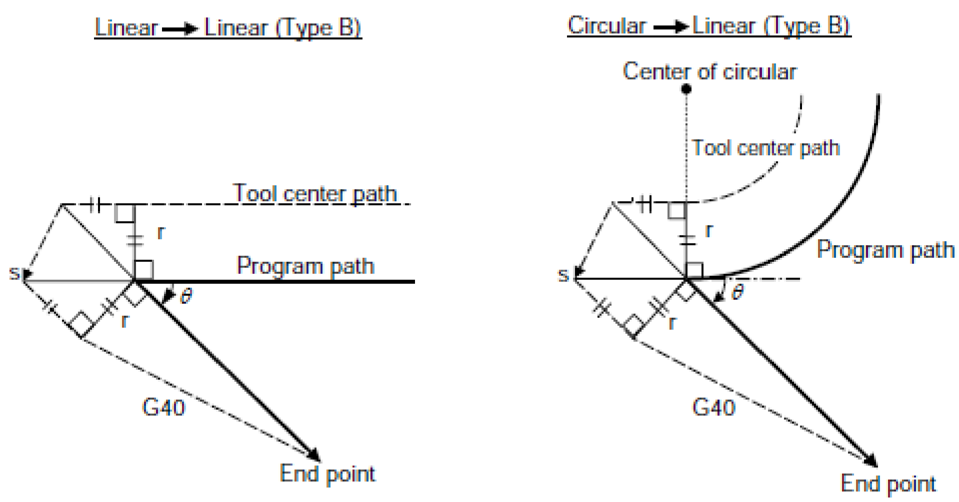
[Tool path of the outer corner (obtuse angle)] (Type A)



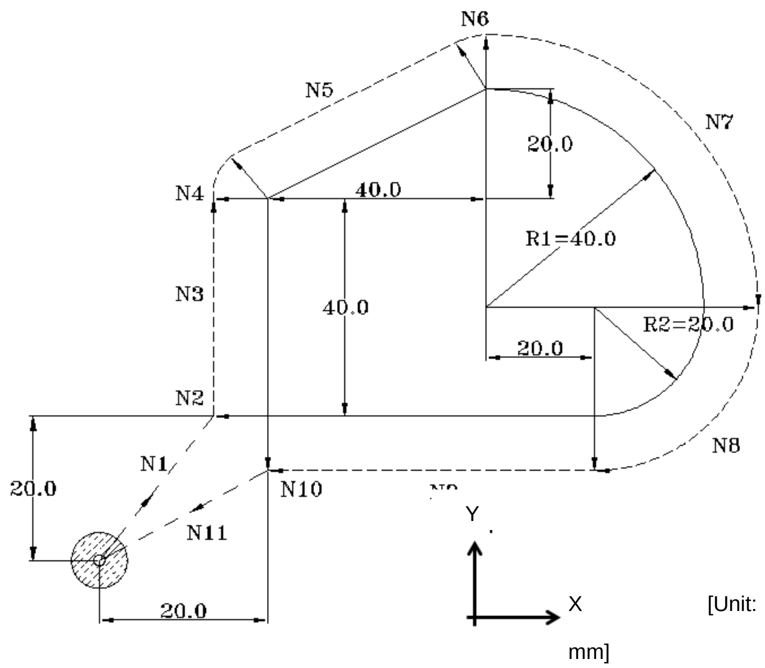
[Tool path of the outer corner (obtuse angle)] (Type B)



[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.	(Compensation path of the arc type)
N5 X40 Y20.	
N6 G39 X40.	(Compensation path of the arc type)
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.

<table>
<tr><th>NC parameters</th><th>Group name and parameter name</th></tr>
<tr><td>NC channel parameters</td><td>
<table>
<tr><th>Group</th><th>Name</th><th>Channel 1</th></tr>
<tr><td></td><td>How to Apply the Compensation Value of th</td><td>0: Apply the diameter value</td></tr>
<tr><td></td><td>Compensation Type of the Tool Diameter</td><td>0: Bypass Traverse</td></tr>
<tr><td></td><td>Whether to check the tool interference duri</td><td>0: Do not check</td></tr>
<tr><td></td><td>Compensation amount of the tool diameter</td><td>0 mm</td></tr>
<tr><td></td><td>Compensation amount of the tool diameter</td><td>0 mm</td></tr>
<tr><td></td><td>Compensation amount of the tool diameter</td><td>0 mm</td></tr>
<tr><td></td><td>Compensation amount of the tool diameter</td><td>0 mm</td></tr>
<tr><td></td><td>Compensation amount of the tool diameter</td><td>0 mm</td></tr>
<tr><td></td><td>Compensation amount of the tool diameter</td><td>0 mm</td></tr>
<tr><td></td><td>Compensation amount of the tool diameter</td><td>0 mm</td></tr>
<tr><td></td><td>Compensation amount of the tool diameter</td><td>0 mm</td></tr>
<tr><td></td><td>Compensation amount of the tool diameter</td><td>0 mm</td></tr>
<tr><td></td><td>Compensation amount of the tool diameter</td><td>0 mm</td></tr>
<tr><td></td><td>Compensation amount of the tool diameter</td><td>0 mm</td></tr>
<tr><td></td><td>Compensation amount of the tool diameter</td><td>0 mm</td></tr>
<tr><td></td><td>Compensation amount of the tool diameter</td><td>0 mm</td></tr>
</table>
* The parameter related to the tool diameter compensation amount is "tool diameter compensation amount 1 ~ tool diameter compensation amount 128".</td></tr>
</table>

16) Tool length compensation (G43, G49)

```
G43 Z_ H_
G49 Z_
```

G43: Tool length + length compensation

G49: Cancel the tool length compensation

Z_: Z-axis movement command (in the case of the G17 plane)

The absolute command and incremental command are available.

H_: Offset number storing the tool length compensation value

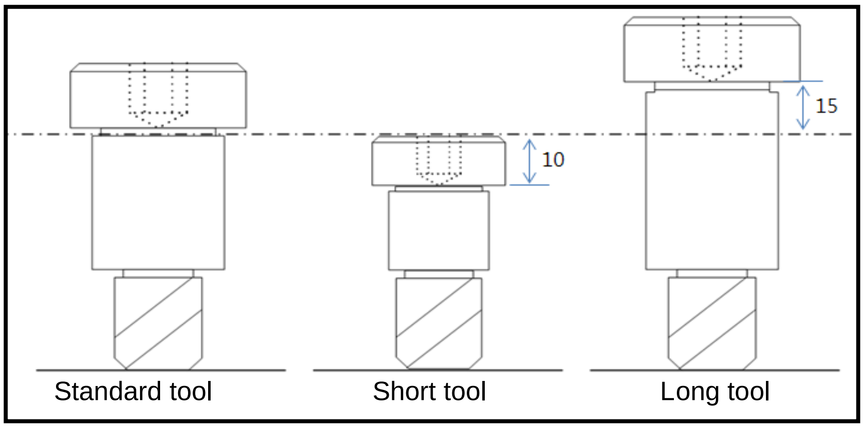
When instructed by the offset number, the H code which stores the tool length compensation value at the end point coordinate value of the Z-axis movement command programmed by the incremental and the absolute command, for the G43, the coordinate value of the added result becomes the end point. The length value can have a +, - sign.

If the Z-axis movement command is omitted, G43 applies the length offset in the block with the next Z-axis command in the + direction.

The following methods are applied to measure the tool length first.

(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.



The G43 and G49 functions are modal commands so they remain valid once they are commended. Accordingly, if the G43 is commanded once immediately after tool change and then, the G49 is commanded before tool change after the tool task is completed, the 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.

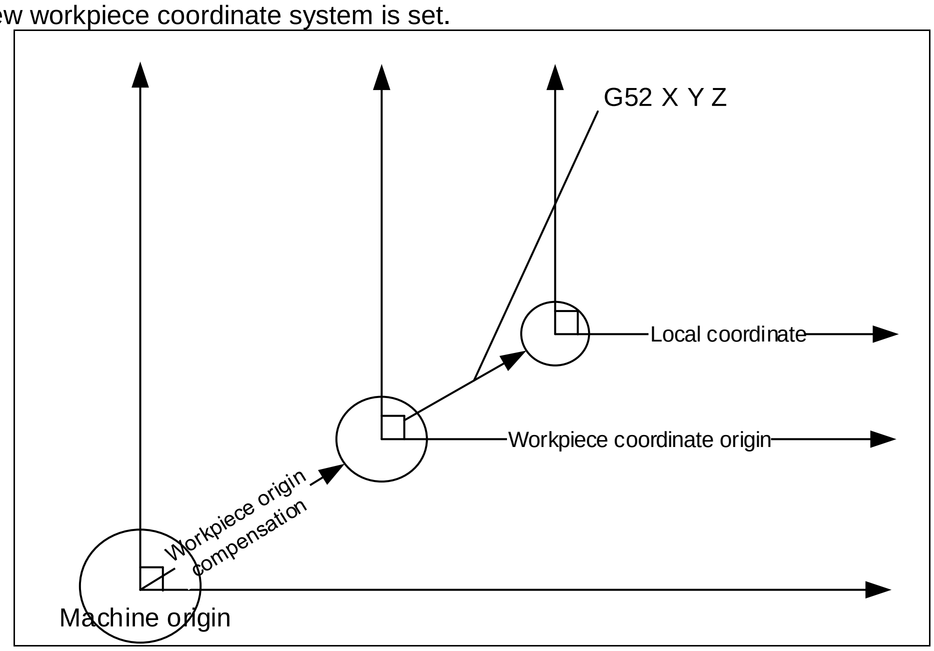
NC parameters	Group name and parameter name		
NC channel parameters	**Group**	**Name**	**Channel 1**
	Tool Length Compensation	Compensation amount of the tool length 62	0 mm
		Compensation amount of the tool length 63	0 mm
		Compensation amount of the tool length 64	0 mm
		Compensation amount of the tool length 65	0 mm
		Compensation amount of the tool length 66	0 mm
	* The parameter related to the tool length compensation amount is "tool length compensation amount 1 ~ length compensation amount 128".		

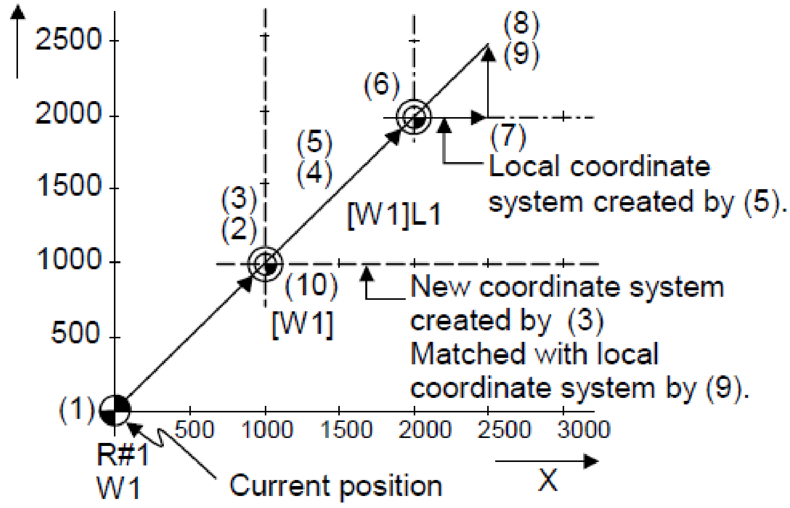
17) Local coordinate system setting (G52)

G52 X_ Y_ Z_

G52: Local coordinate system setting

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
```

18) Selecting 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      (Moving to the X-140 Y-120 Z0 position of the machine coordinate system)
G92 X0 Y0 Z150              (Rest by changing the workpiece coordinate system)
G30 G91 Z0
G54 G00 G90 X0 Y0
M30
```

19) Selecting the workpiece coordinate system 1~6 (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. That is, G54
                        % Traverse to the origin of the coordinate system rapidly
M30
```

20) 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.

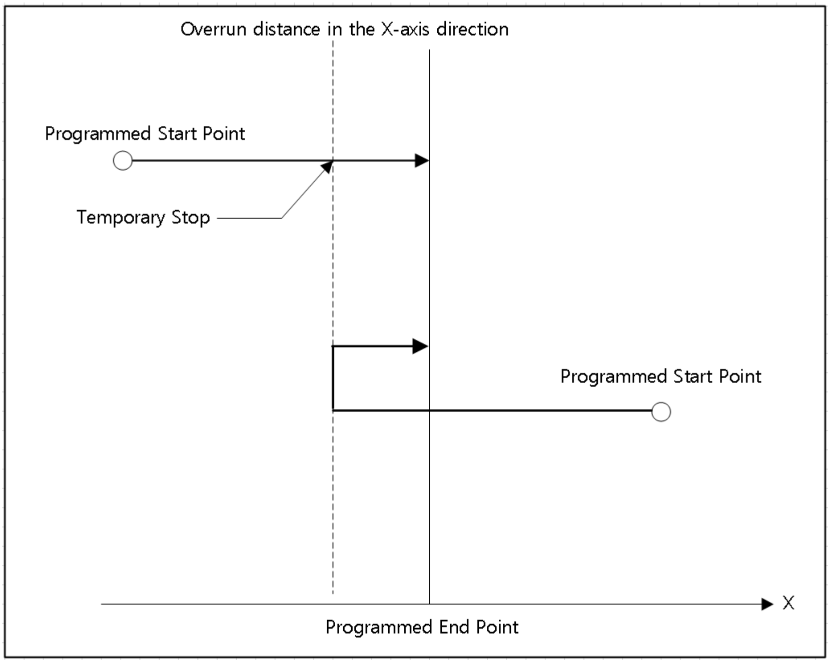
Therefore, the G60 command is applied, it always moves from the same direction to the target position.

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 parameters	Group name and parameter name		
NC channel/axis parameters	Group	Name	X Axis
	Auxiliary Function	Overrun distance in single dir. positioning	0 mm

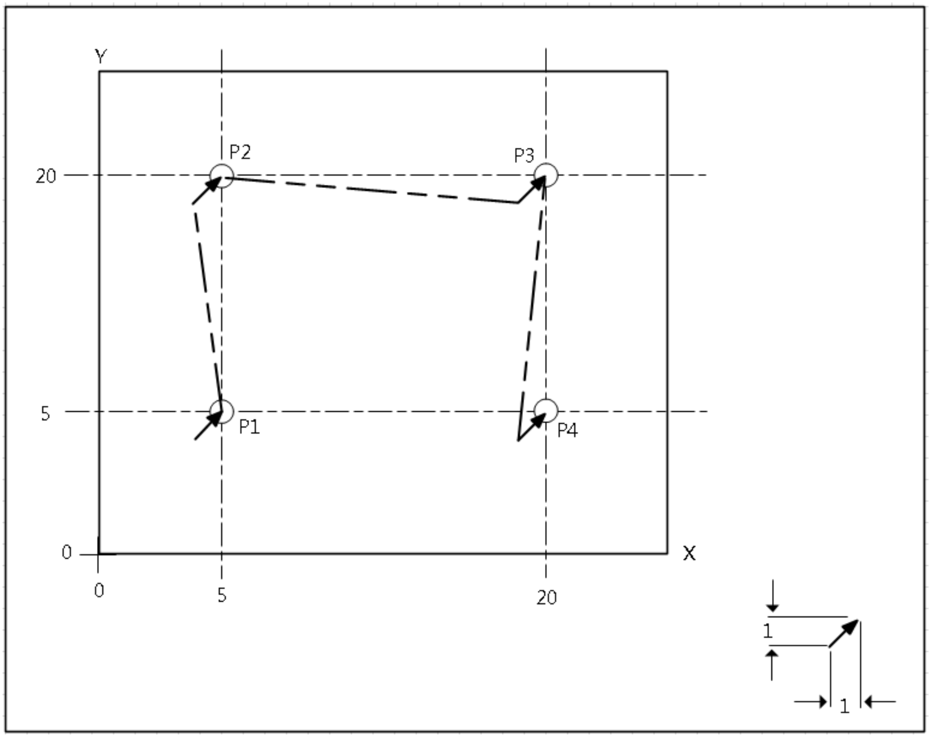
Note that the single direction positioning command does not eliminate the backlash physically.



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
```

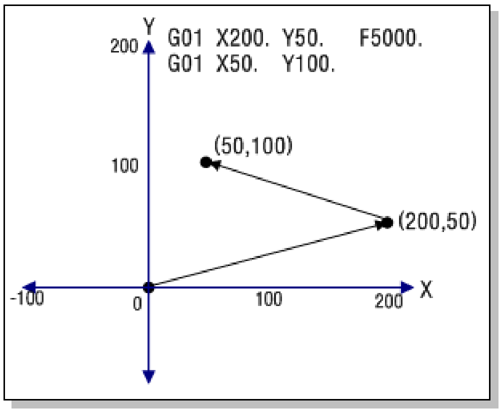
21) Absolute command (G90)

```
G90 G01 X_ Y_ Z_ A_ B_ C_ U_ V_ W_
```

G90: Absolute command
G01/G00: Linear interpolation /Positioning
X_ Y_ Z_ A_ B_ C_ U_ V_ W_: Target position to traverse

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.

22) Incremental command (G91)

G91 G01 X_ Y_ Z_ A_ B_ C_ U_ V_ W_

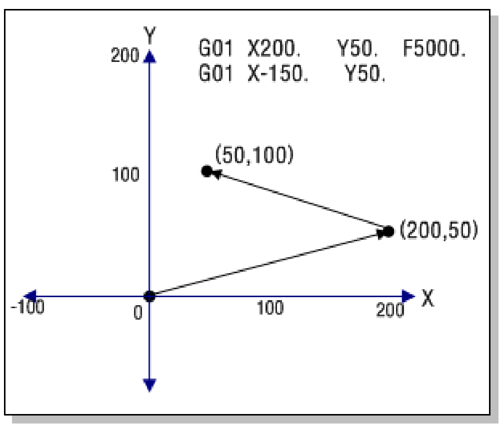
G91: Incremental command

G01/G00: Linear interpolation /Positioning

X_ Y_ Z_ A_ B_ C_ U_ V_ W_: Target position to traverse

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.

23) Setting the workpiece coordinate system, maximum speed of the main axis (G92)

When setting the workpiece coordinate system

G92 X_ Y_ Z_ A_ B_ C_ U_ V_ W_

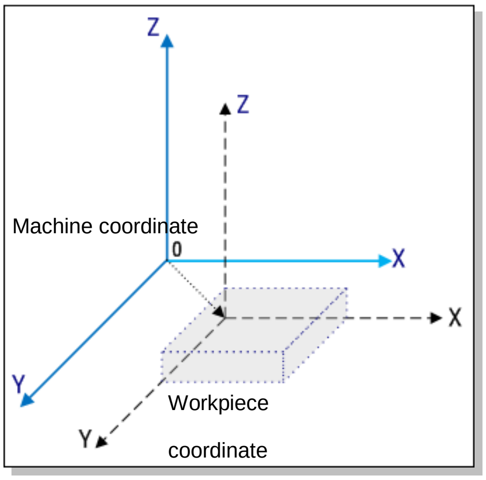
G92: workpiece coordinate system command

X_ Y_ Z_ A_ B_ C_ U_ V_ W_: Enter the offset value of each axis

The workpiece coordinate system setting is the command to shift the current coordinate system by the entered offset.

It is used to reset the coordinate system based on the reference point of the workpiece. Until the workpiece coordinate system is selected after the applicable command, it operates based on the shifted coordinate system(G54 ~ G59).

* Caution: It should be applied after homing.



% Apply after homing	
G90 X10 Y10	% Move to the position of the workpiece coordinate system X=10 Y=10
G92 X100 Y100	% Shift the workpiece coordinate system X=10,Y10 to X=100,Y=100
G90 X10 Y10	% Move to the position of the workpiece coordinate system -80,-80

24) 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.

25) Feed mode command per revolution (G95)

G95 G01 X_ F_

G95: Feed rate per revolution of the main 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 main 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.

26) Cylindrical interpolation mode setting (G107)

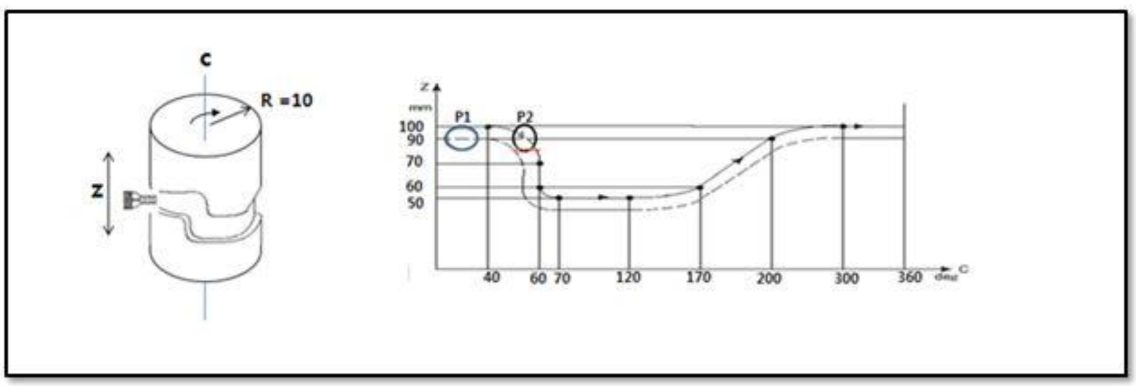
G107 C_

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.

27) 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

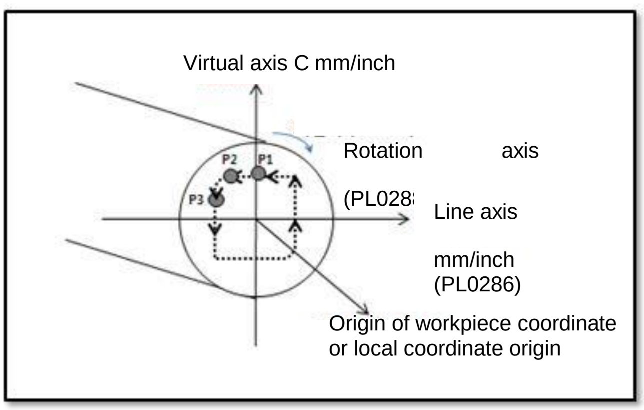
The polar coordinate interpolation converts the command entered into the rectangular coordinate system into the movement of the linear axes (X, Y, Z: tool movement) and the rotary axes (A, B, C: workpiece). This is executed on the polar coordinate interpolation plane created by the linear axis and the 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.

NC parameters	Group name and parameter name		
	Group	Name	Channel 1
NC channel parameters	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

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 feed
G01 X-8	% P2: X -8 position feed
G03 X-10. C8. R2.	% P3: X -10 Y 8
G113	% Cancel 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	Description
M00	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	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	End of Program	This is the command indicating the end of the program. After the operation of the block is completed, the main 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 main axis [CW]	Forward rotation of the main axis Before this command, you must adjust the gear shift and the number of revolutions of the main axis in advance.
M04	Reverse rotation of the main axis [CCW]	Reverse rotation of the main axis. Before this command, you must adjust the gear shift and the number of revolutions of the main axis in advance.
M05	Main axis stop	Main 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	End of Tap	At the end of the program, the program returns to its beginning again and 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          ; False 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.

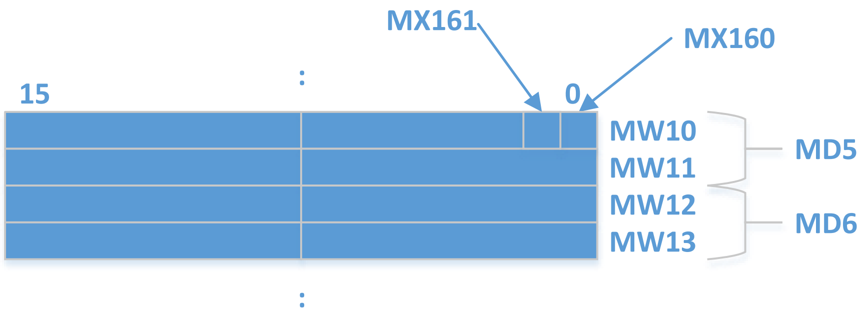
2) Usage according to the range of variables

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	Remarks
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
16bit (W)	Addressing the value in word units (16bit) after the 'W' indicator
32bit (D)	Addressing the value in double word units(32bit) after the 'D' indicator
64bit (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
16bit	#W0 ~ #W2047	#MW0 ~ #MW1048575	#FW0 ~ #FW65535
32bit	#D0 ~ #D1024	#MD0 ~ #MD524287	#FD0 ~ #FD32767
64bit	#L0 ~ #L511	#ML0 ~ #ML262143	#FL0 ~ #FL16383

3) Specifying 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 statement (IF)

```
IF [……] GOTO N_
```

IF: IF statement

[……]: IF statement’s conditions

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) Repetitive statement (DO, WHILE)

```
WHILE [<Conditional expression>] DO n
(n = 1, 2, 3, …)
~
END n
```

WHILE: Conditional repetitive statement

DO n: Repeat until n declaration statement

[……]: Conditional statement

END n: End of the block to be repeated

When the <conditional expression> is met, it repeats from the block following DO n to the END n block. If the <conditional expression> is not satisfied, it jumps to the block following the END n. 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.

Category	Syntax	Remarks
Substitution	#Li = #Lj	
Addition	#Li = #Lj + #Lk	
Subtraction	#Li = #Lj - #Lk	
OR	#Li = #Lj OR #Lk	
XOR	#Li = #Lj XOR #Lk	
Multiplication	#Li = #Lj * #Lk	
Division	#Li = #Lj / #Lk	
AND	#Li = #Lj AND #Lk	
Sin	#Li = SIN[#Lj]	
Cos	#Li = COS[#Lj]	
Tan	#Li = TAN[#Lj]	
Atan	#Li = ATAN[#Lj]	

Category	Syntax	Remarks
Sqrt	#Li = SQRT[#Lj]	
Abs	#Li = ABS[#Lj]	
Round	#Li = ROUND[#Lj]	Round-off operation
And	#Li = AND[#Lj]	
Or	#Li = OR[#Lj]	
Fix	#Li = FIX[#Lj]	Round-down operation
Fup	#Li = FUP[#Lj]	Round-up operation

8) Comment (;, %, ())

```
%, ;, ()
```

% : Comment statement

; : Comment statement

( ) : Comment statement

Description of commands

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) Example of using program operation instructions

```
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)  % Equation

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.

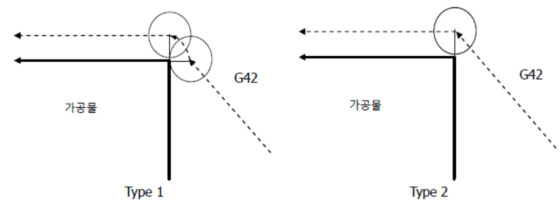
Parameters	Group	No.	Item	Description
1. Channel parameters	1. Basic setting	1	Target machining quantity	Set the target machining quantity. (0 ~ 2,147,483,647)
		2	Target machining quantity at M99 repeated machining	Set the target machining quantity for repeated machining with M99. If the set value matches the current machining quantity, the cycle automatically stops. (0 ~ 2,147,483,647)
		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: Keep 1: Do not keep
		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: Time 1: Number of revolutions

Parameters	Group	No.	Item	Description
1. Channel parameters	1. Basic setting	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
		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.
		19	Set the inner/outer side of G22 No Travelling Area	0: Inner side 1: Outer side

Parameters	Group	No.	Item	Description
1. Channel parameters	1. Basic setting	20	Whether to use the 3rd 'No Travelling Area'	0: 'No Travelling Area' is valid. 1: 'No Travelling Area' is invalid.
		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
		33	Monitoring time for in-position completion	0 ~ 65,535ms

Parameters	Group	No.	Item	Description
1. Channel parameters	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
		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)
		5	Circular radius with the speed-limiting function for the arc machining	(0 ~ 10,000 unit, real number)
		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 " . (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)
		9	Circular milling acceleration	Set the acceleration at the circular milling.
		10	Circular milling deceleration	Set the deceleration at the circular milling.
		11	Circular milling jerk	Set the jerk at the circular milling.

Parameters	Group	No.	Item	Description
1. Channel parameters	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)
		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)
		4	Acceleration / deceleration method of the interpolation operation	1: Acceleration / deceleration before interpolation
		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
		10	Deceleration at the time of cutting feed (before interpolation)	Deceleration at the time of cutting feed
		11	Jerk at the time of cutting feed (before interpolation)	Jerk at the time of cutting feed

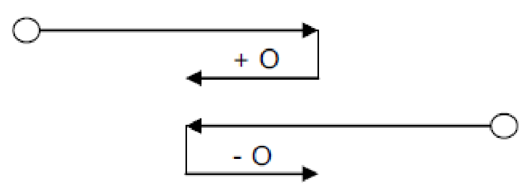
Parameters	Group	No.	Item	Description
1. Channel parameters	8.Tool diameter compensation	129	How to apply the compensation value of the tool diameter	Set the method of applying the compensation amount of the tool diameter when compensating the tool diameter. 0: Apply the diameter value 1: Apply the radius 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

Parameters	Group	No.	Item	Description
1. Channel parameters	10. Workpiece coordinate system	1	Whether to use the workpiece coordinate system shift amount.	Set whether to use the workpiece coordinate system shift amount. 0: Unused 1: 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 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 G55workpiece 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 values for the W axis.
		61	G56 workpiece coordinate system value 1	Sets the G56 workpiece coordinate system values 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 values for the W axis.
		71	G57 workpiece coordinate system value 1	Set the G57 workpiece coordinate system values for the X axis.
		……	……	Sets the G57 workpiece coordinate system values for the 7 axes; Y, Z, A, B, C, U, V

Parameters	Group	No.	Item	Description
1. Channel parameters	10. Workpiece coordinate system	79	G57 workpiece coordinate system value 9	Set the G57 workpiece coordinate system values for the W axis.
		81	G58 workpiece coordinate system value 1	Set the G58 workpiece coordinate system values 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 values for the W axis.
		91	G59 workpiece coordinate system value 1	Set the G59 workpiece coordinate system values 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 values 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
		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: Do not display 1: Display
		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)
		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)
		……	……	

Parameters	Group	No.	Item	Description
1. Channel parameters	11. Macro program	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.
	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 Traverse(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)
		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)

Parameters	Group	No.	Item	Description
1. Channel parameters	16. Relative coordinate 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.
2.Channel/Axis parameters	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. Origin	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 traverse	2	Rapid traverse acceleration	The set value is used as the acceleration of the G00 block.
		3	Rapid traverse deceleration	The set value is used as the deceleration of the G00 block.
		4	Rapid traverse jerk	The set value is used as the jerk of the G00 block.

Parameters	Group	No.	Item	Description
2.Channel/Axis parameters	3. Rapid traverse	5	Rapid traverse speed	The set value is used as the traverse speed of the G00 block. (0~100000 unit/min, real number)
	4. Traverse area	1	Minimum value of the G22 Traverse-Prohibited Area range for the X, Y, and Z axis.	Set the minimum value of the G22 Traverse-Prohibited Area range for the X, Y, and Z axis. (-100,000~100,000 unit, real number)
		2	Maximum value of the G22 Traverse-Prohibited Area range for the X, Y, and Z axis.	Set the maximum value of the G22 Traverse-Prohibited Area range for the X, Y, and Z axis. (-100,000~100,000 unit, real number)
		3	Minimum value of the 3rd Traverse-Prohibited Area range for the X, Y, and Z axis.	Set the minimum value of the 3rd Traverse-Prohibited Area range for the X, Y, and Z axis. (-100,000~100,000 unit, real number)
		4	Maximum value of the 3rd Traverse-Prohibited Area range for the X, Y, and Z axis.	Set the maximum value of the 3rd Traverse-Prohibited Area range for the X, Y, and Z axis. (-100,000~100,000 unit, real number)
	5. Sub setting	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)

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

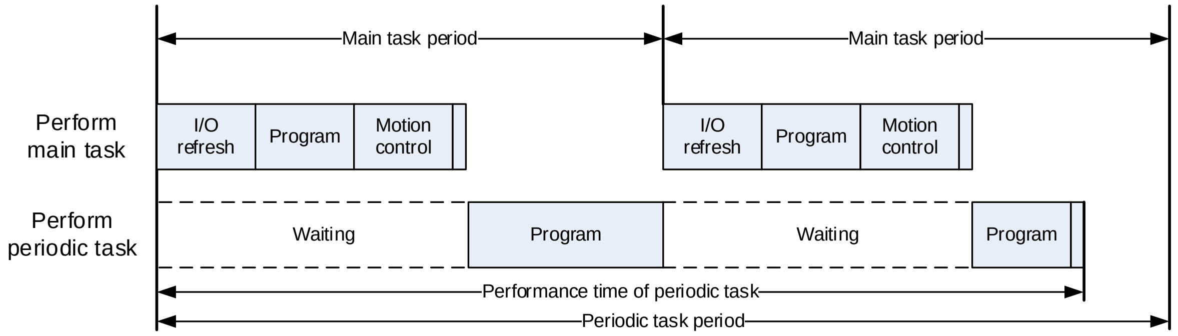
Main task cycle	Periodic task cycle that can be set
500 ㎲	1㎳, 2㎳, 3㎳, 4㎳, 5㎳, … 95㎳, 96㎳, 97㎳, 98㎳, 99㎳, 100㎳
1 ㎳	1㎳, 2㎳, 3㎳, 4㎳, 5㎳, … 95㎳, 96㎳, 97㎳, 98㎳, 99㎳, 100㎳
2 ㎳	2㎳, 4㎳, 6㎳, 8㎳, 10㎳, … 92㎳, 94㎳, 96㎳, 98㎳, 100㎳
4 ㎳	4㎳, 8㎳, 12㎳, 16㎳, 20㎳, … 84㎳, 88㎳, 92㎳, 96㎳, 100㎳

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.

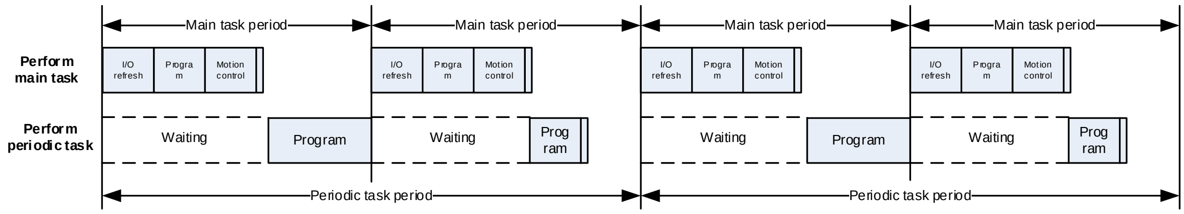


In the main task execution, the double line display after the program execution the motion control or periodic task execution indicates that the task execution is completed.

(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.



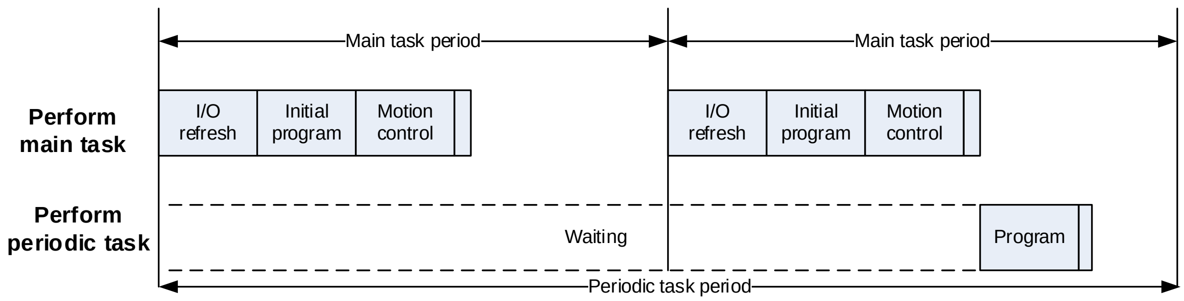
(2) Initialization task

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.

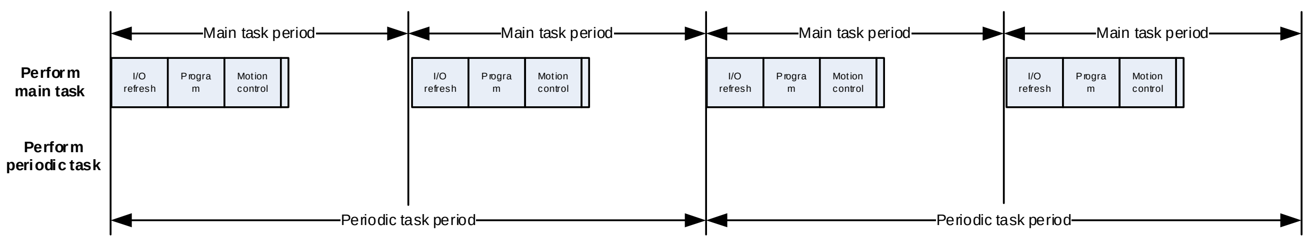
When the initialization program execution is completed, and the initialization task execution is terminated as shown below, the main task program and periodic task program are executed.



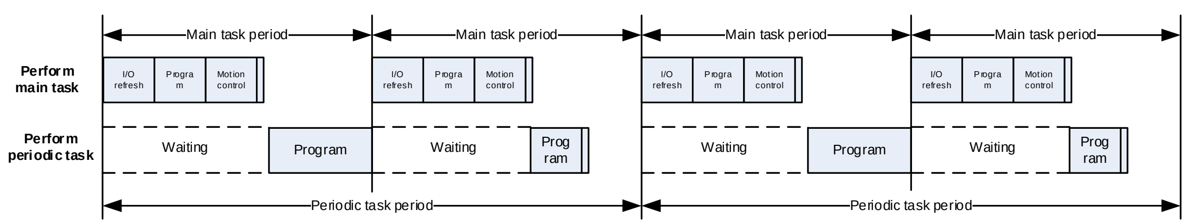
10.1.4 Examples of Task Execution Sequence

Below are descriptions of the execution sequence for the main task and periodic task.

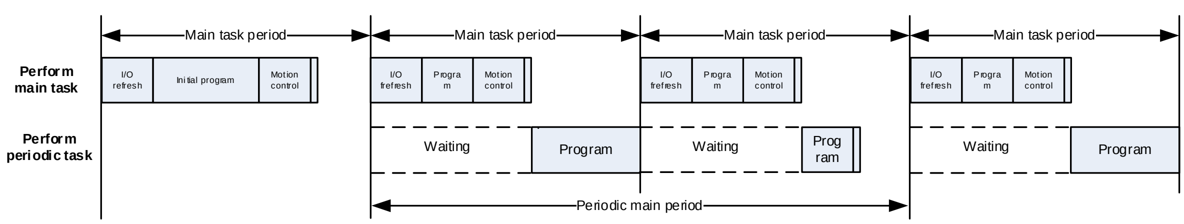
(1) If there is only main task program



(2) If there main/periodic task programs



(3) If there is an initialization task program

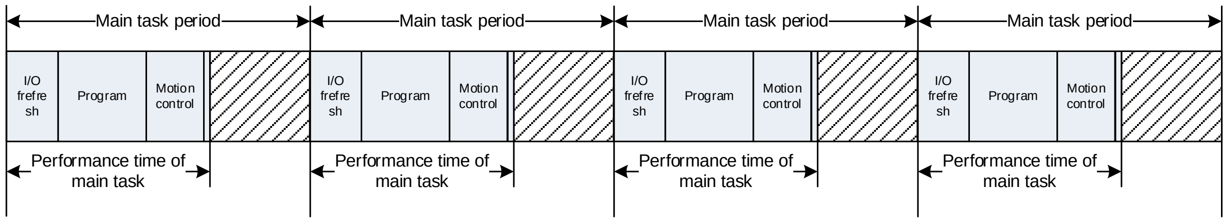


10.1.5 System Service Processing

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.

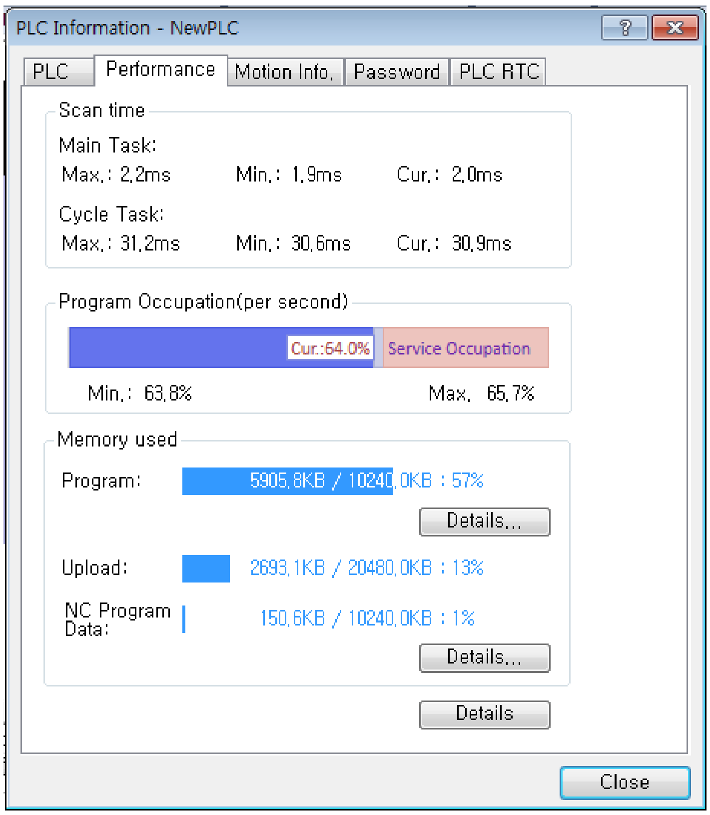


The order of priority of the system service and each task is main task > periodic task > system service, and the main task has the highest priority. When the main task execution cycle is reached while the system service is running, the system service is paused, and the main task is executed. In addition, 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.

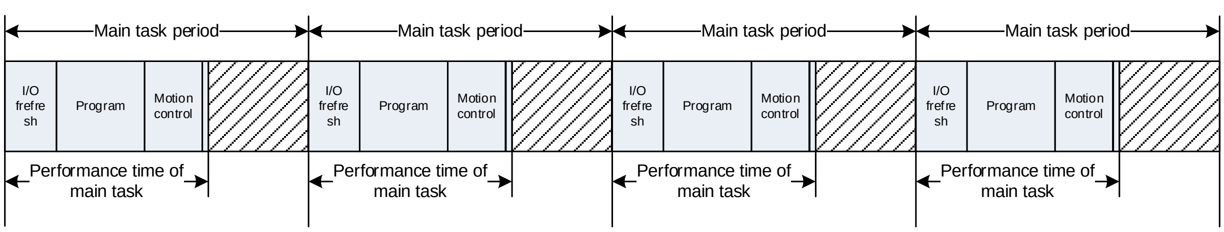
10.1.6 Program Occupancy Rate Operation

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.

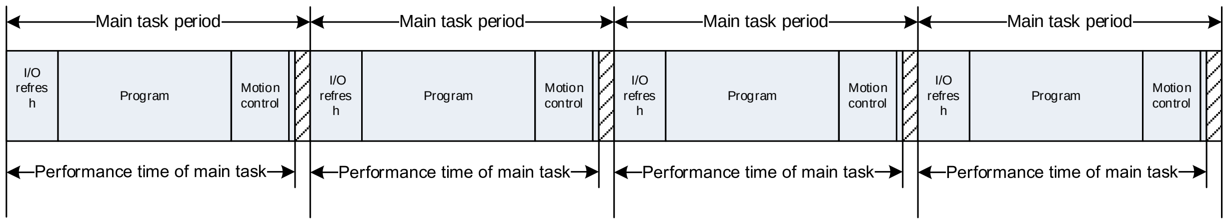


If there is no periodic task as shown below, the system service can be executed in the shaded section if the main task execution is completed.

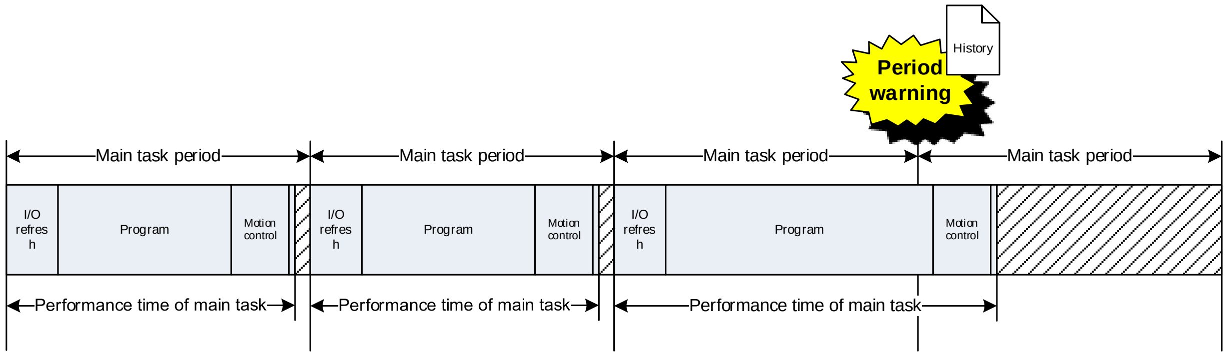


If the program occupancy rate is high (system service occupancy rate is low) as shown below, the system service may not be performed normally. In the basic parameter, a user can set the value ranging from 50 to 95%, and if the set value is exceeded, the task program occupancy rate warning is generated. If the program occupancy rate exceeds 100%, the system enters the ERROR state.

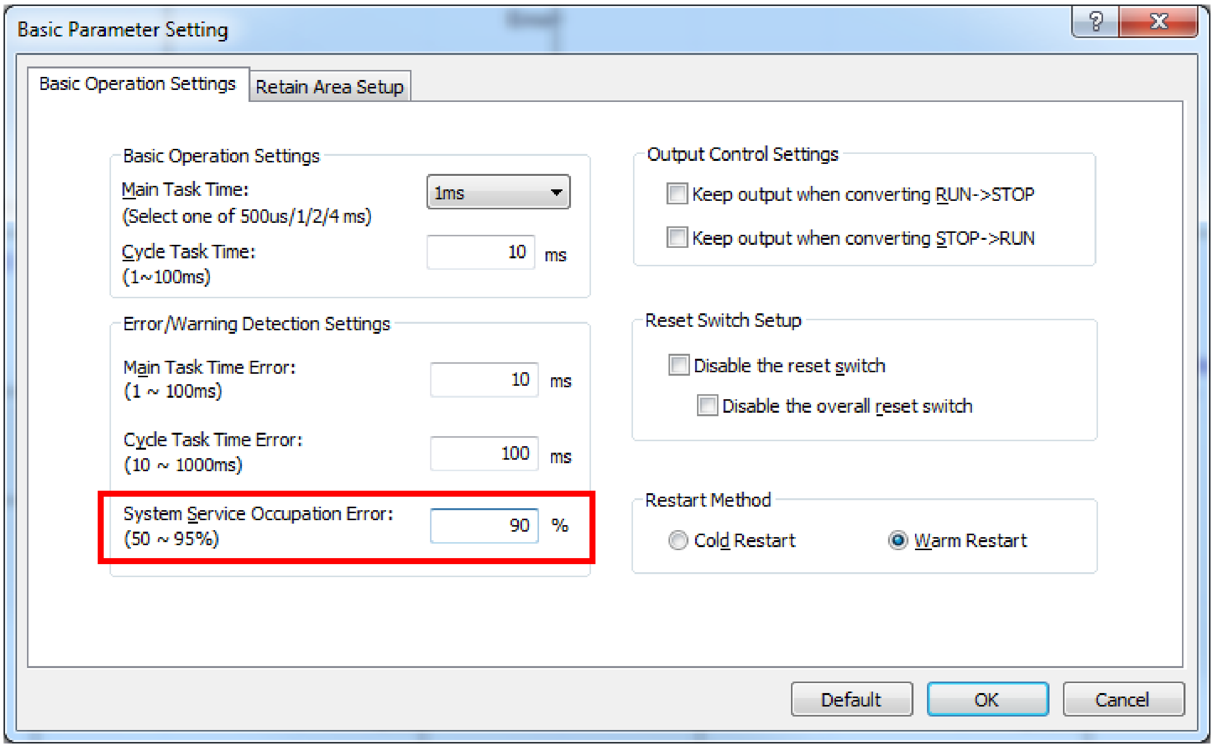
※ Adjust the main task cycle so that the program occupancy rate does not exceed 90%, if possible.



If the cycle warning is generated as shown below, the program occupancy rate may increase.

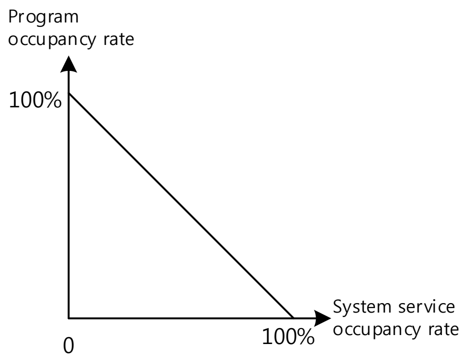


The value for setting the task program occupancy rate excess warning detection can be set from 50% to 95% in the basic parameter settings.



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.

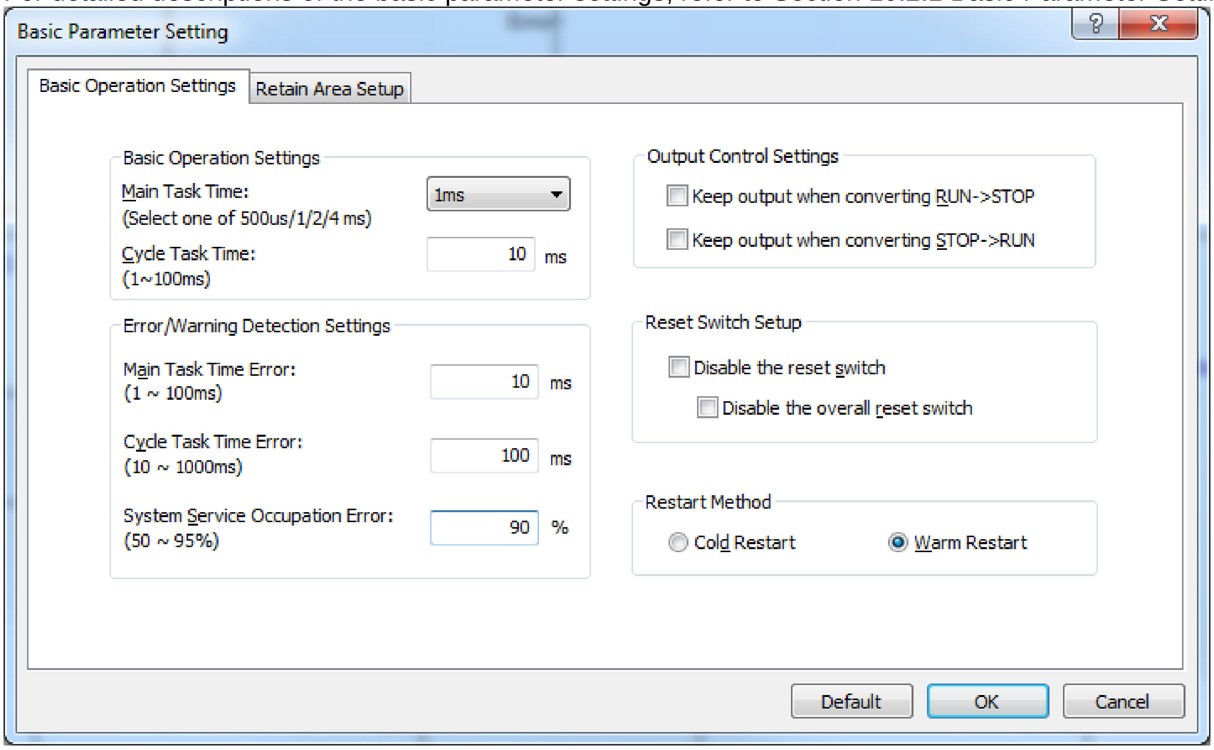
The program occupancy rate is inversely proportional to the system occupancy rate. If the program occupancy rate is 20%, the system occupancy rate is 80%. But if the program occupancy rate is 80%, the system occupancy rate is 20%.



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.



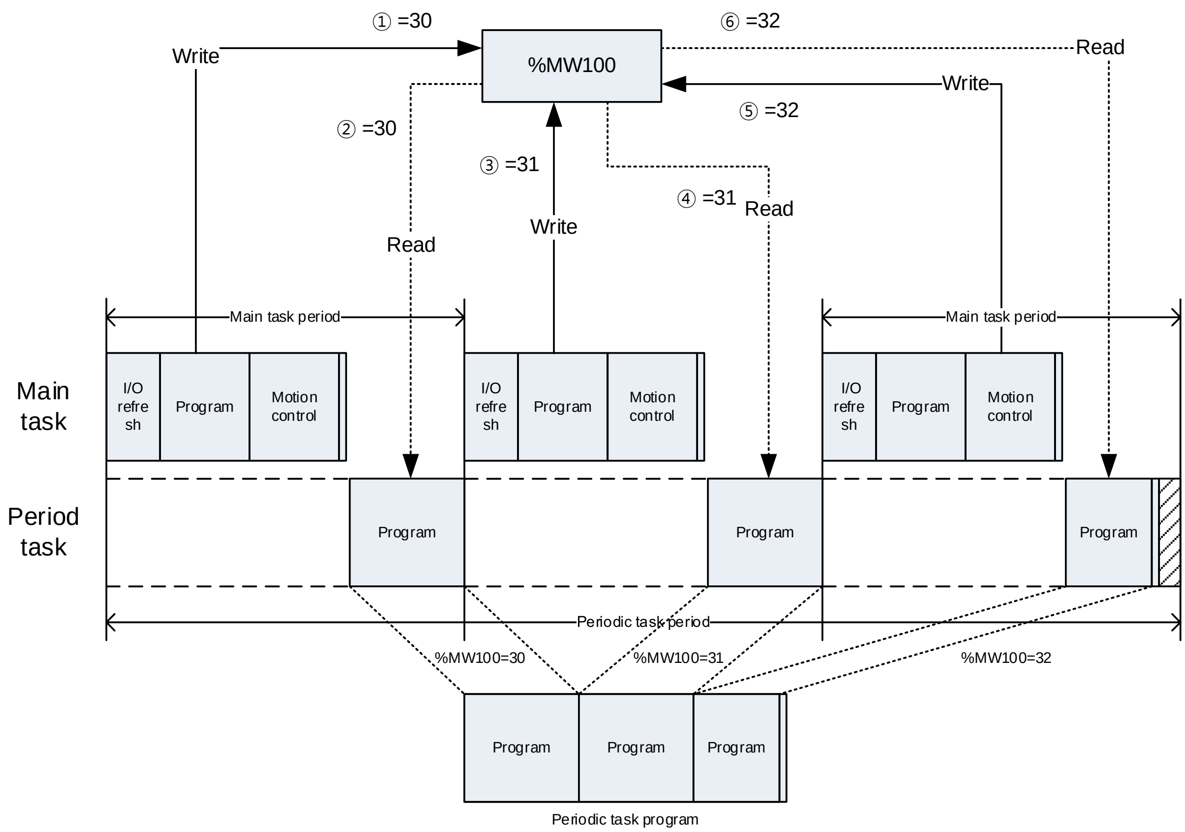
Items	Descriptions	Setting Values	Default
Main task cycle	Sets the time for the main task	500 ㎲, 1 ㎳, 2 ㎳, 4 ㎳	1 ㎳
Periodic task cycle	Sets the time of the periodic task as the multiple of the main task cycle	1~100 ㎳	10 ㎳
Main task cycle error	Sets the main task execution time when the main task is executed 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

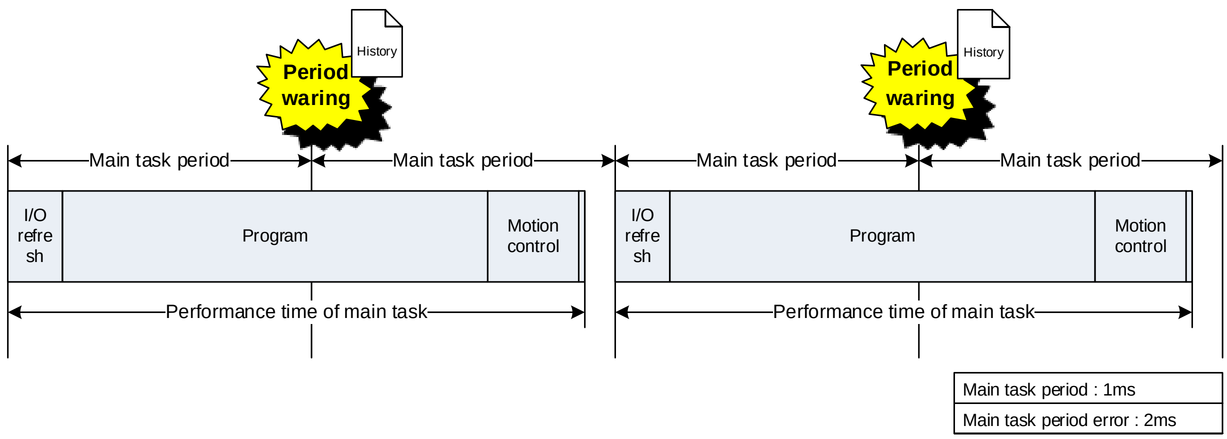
Below are descriptions of the task flags.

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

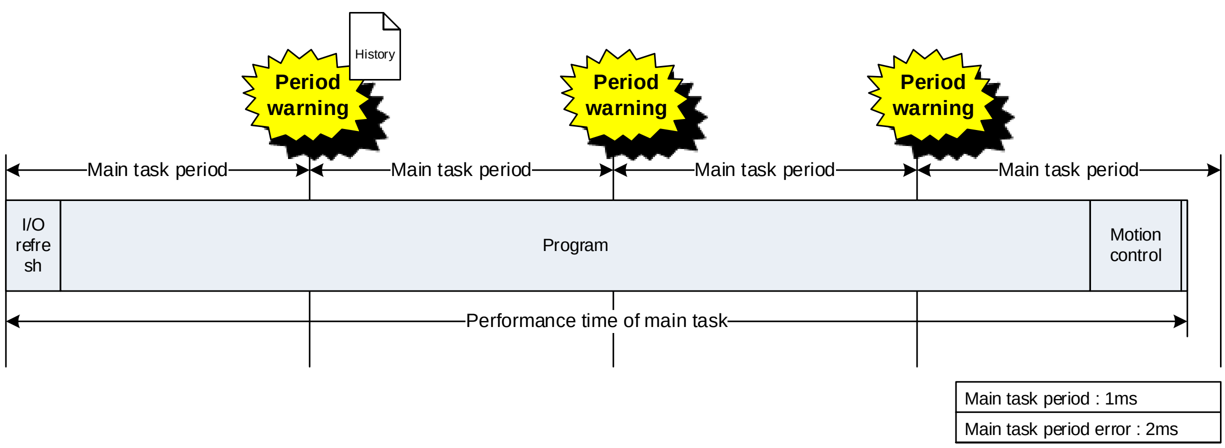
10.1.10 Task-Related Warning/Error

(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.

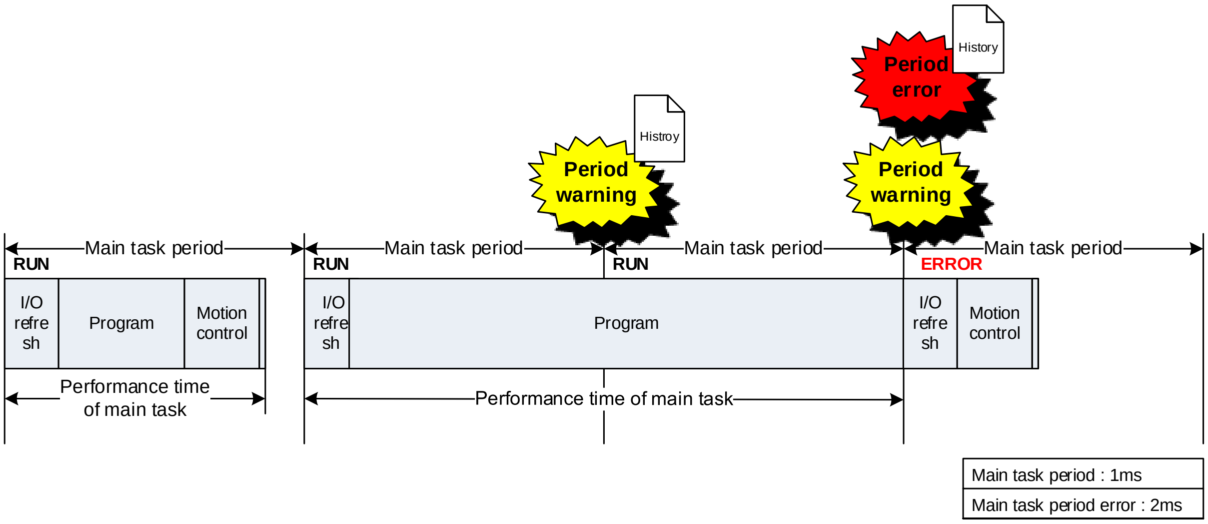


If the task execution is completed in the previous cycle as shown in the figure below, the history is stored. If the cycle over warning is continuously generated, the cycle over warning history is saved only for the first occurrence of the warning. The saved history can be checked in the error history.



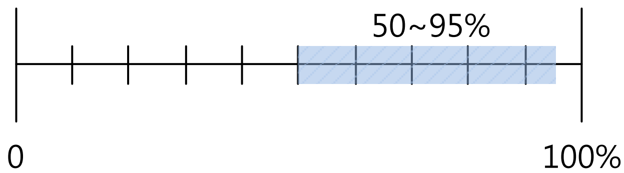
(2) Task cycle error

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.

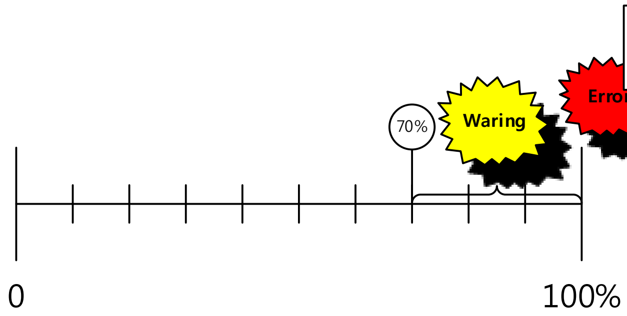


(3) Program occupancy rate excess warning/error

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%.

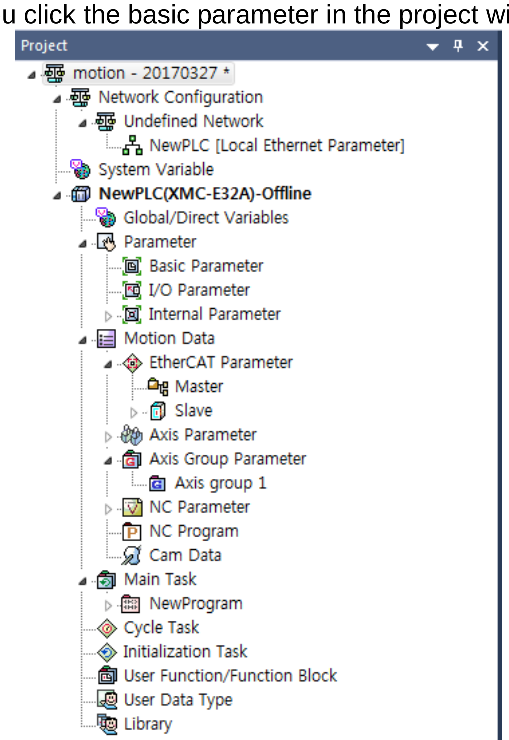


10.2 Parameter Setting

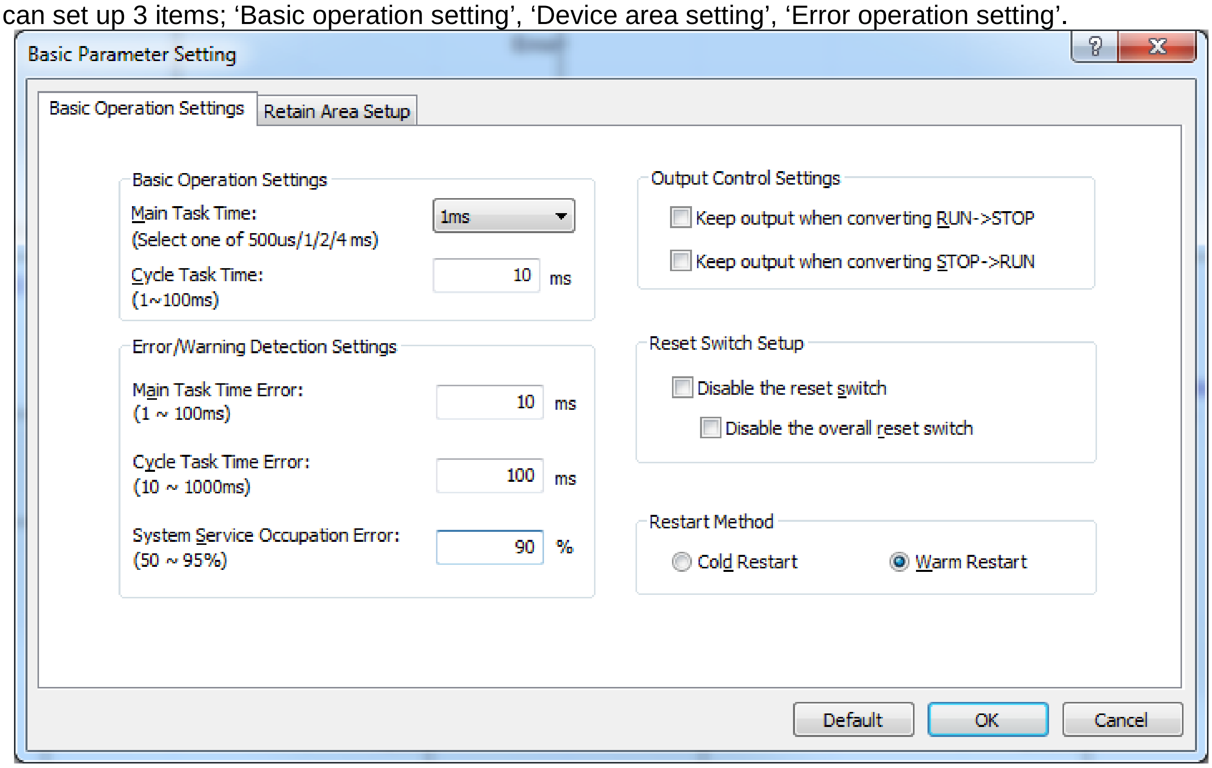
This section describes motion controller's parameter setting.

10.2.1 Basic Parameter Setting

If you click the basic parameter in the project window, the below screen will be displayed.



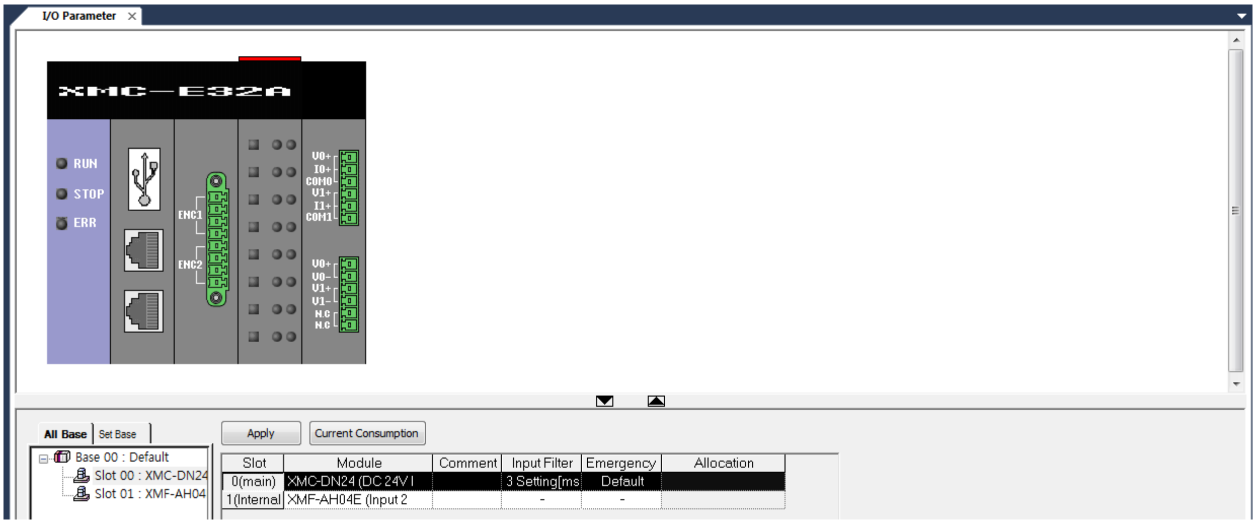
You can set up 3 items; 'Basic operation setting', 'Device area setting', 'Error operation setting'.



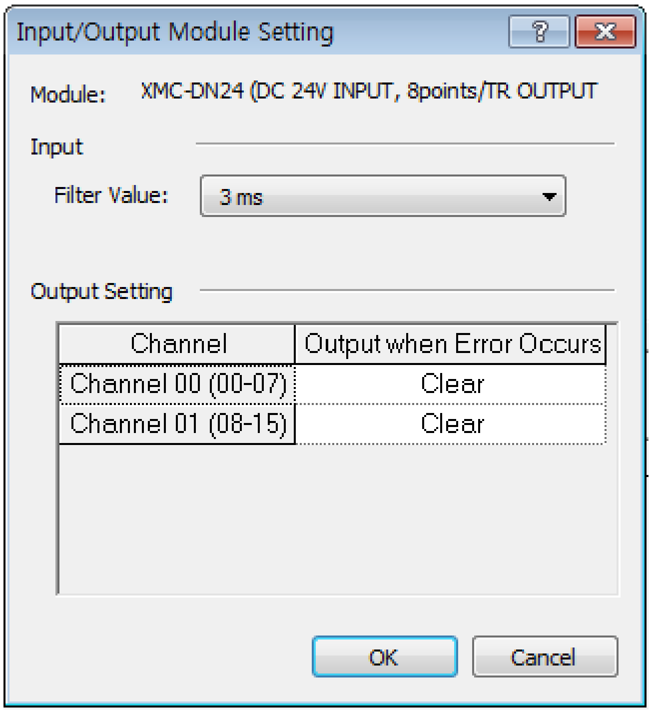
Classification	Items	Descriptions	Setting Values
Basic operations	Main task cycle	Sets the time of the main task	500us, 1ms, 2ms, 4ms
	Periodic task cycle	Sets the time of the periodic task	1~100ms(a multiple of the main task)
	Main task cycle error	Sets the main task execution time that causes an error 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

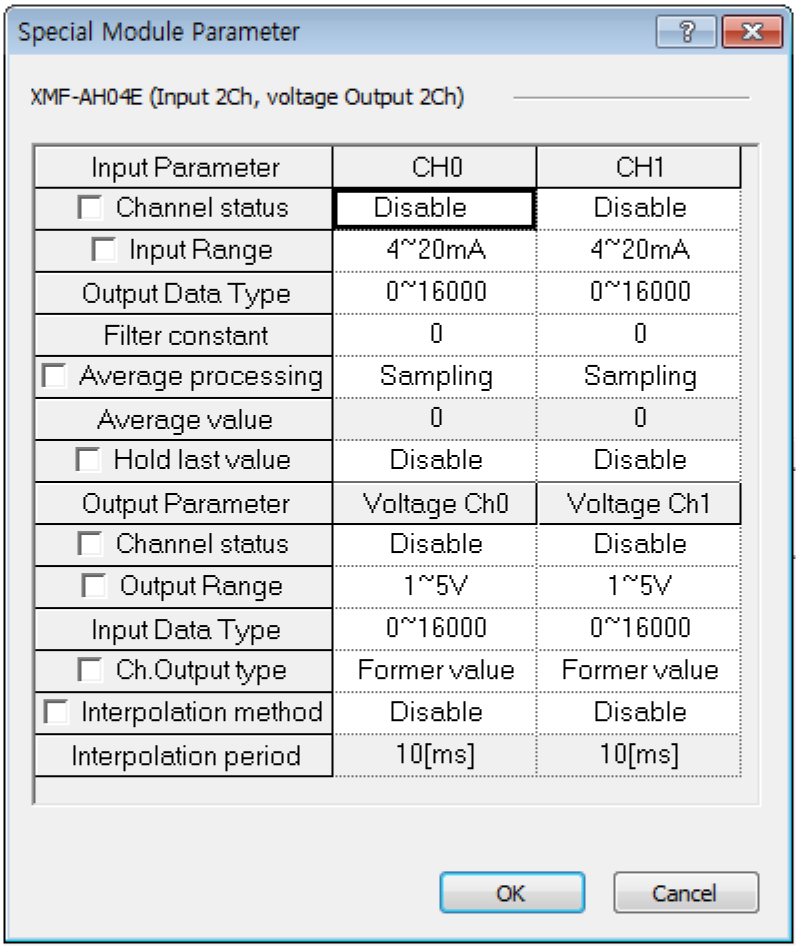
10.2.2 I/O Parameter Setting

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.





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]

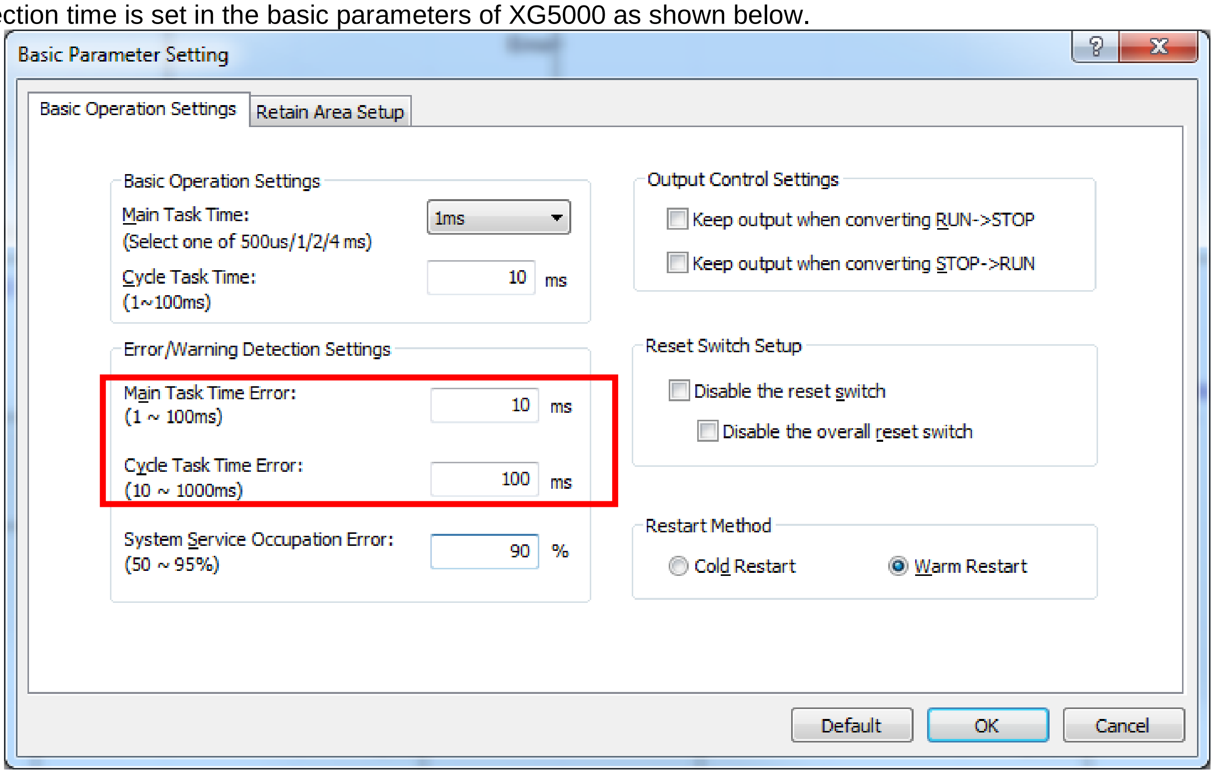
10.3 Self-Diagnosis Function

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.



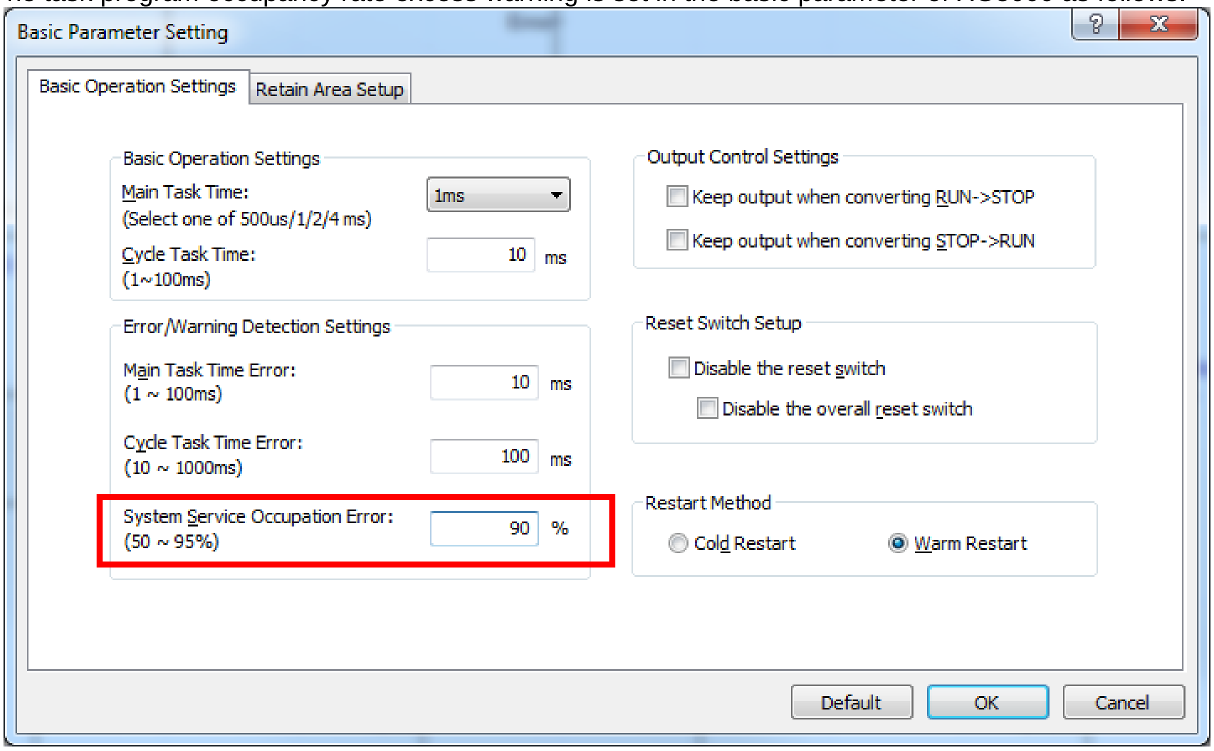
(2) While the program is running, the elapsed scan time is monitored, and if the set detection time is exceeded, the operation of the motion controller is stopped immediately, and an error is generated.

(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.

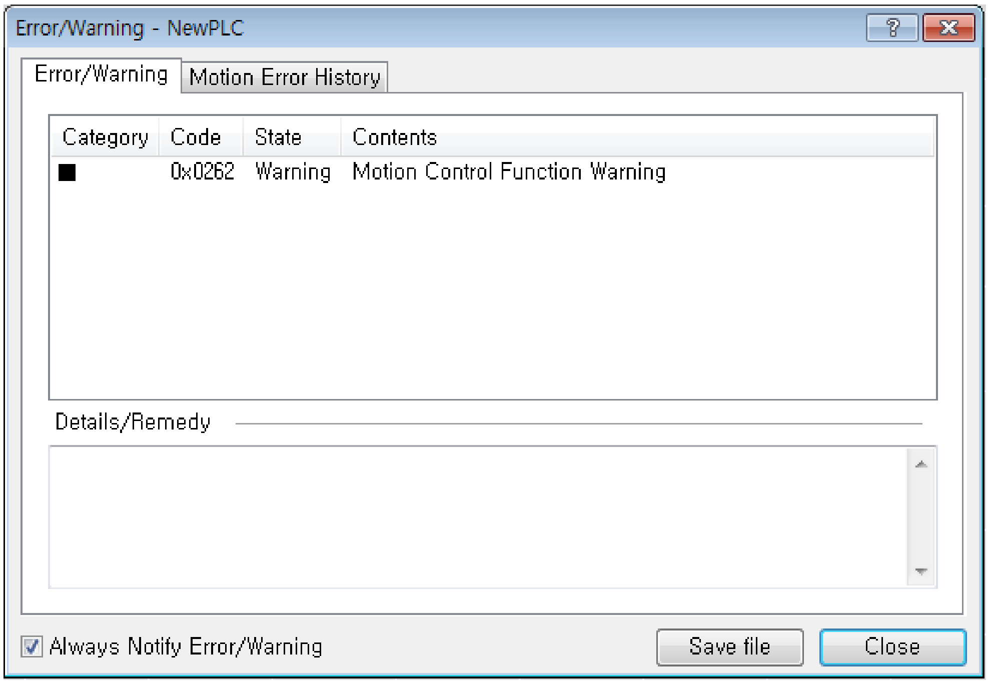


(2) If the set occupancy rate is exceeded while monitoring the task program occupancy rate during the execution of program, the task program occupancy rate excess warning is generated. If the task program occupancy rate is 100% in the state where the warning occurs, the task program occupancy rate error 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.

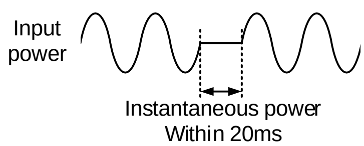
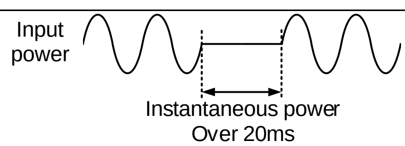
(e) When minor failures occur, through XG5000, a user can figure out the causes of failures by monitoring %FW20483(_ANC_WAR)flag.

(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.

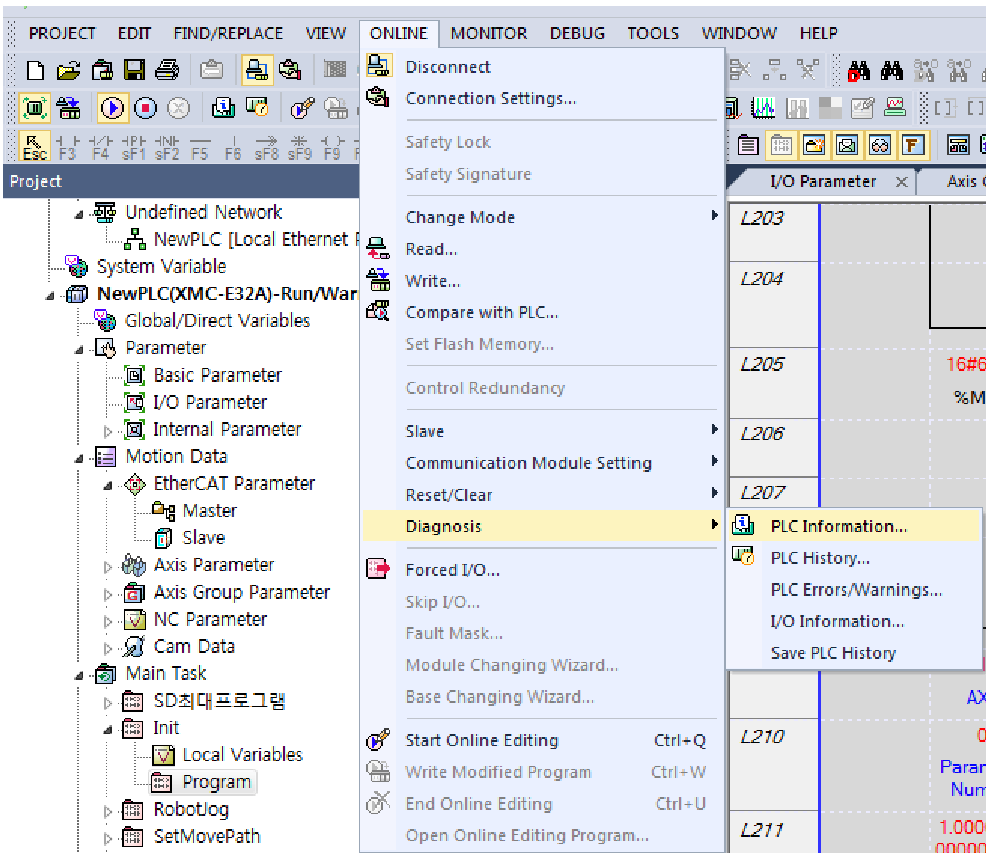
10.4.1 How to Use the RTC

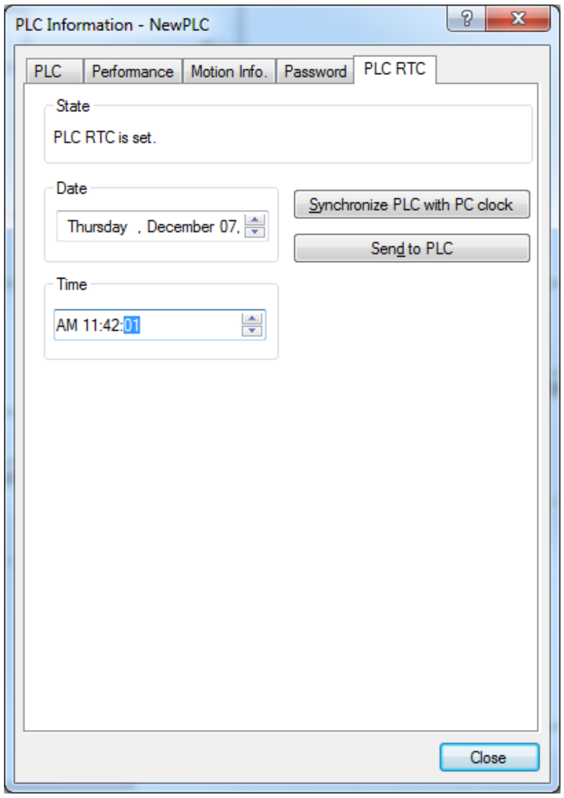
(1) Reading/Setting clock data

(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』.





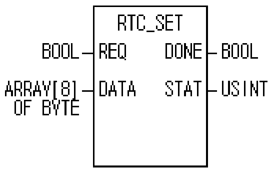
3) If you want to send the time of the PC to the motion controller, click ‘Synchronization with PC clock’ button.
4) If you want to set up the user defined time, after changing set values of the data and time box, click ‘Send to motion controller’.

(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.

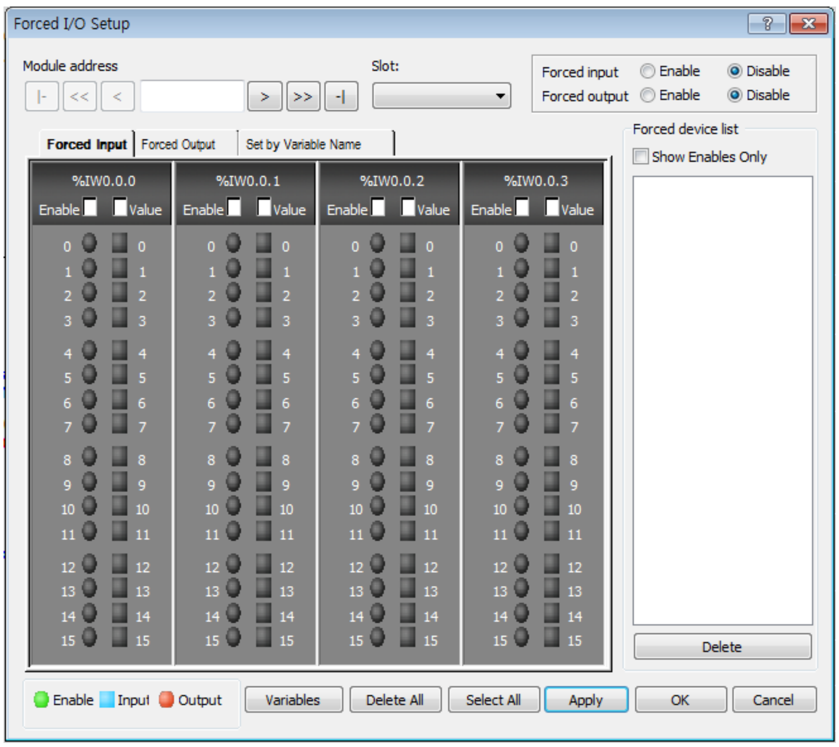
(3) Remote reset
- It is the function to reset the CPU module by remote control when errors occur.
- 'Reset' and ‘Overall Reset’ are available.

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

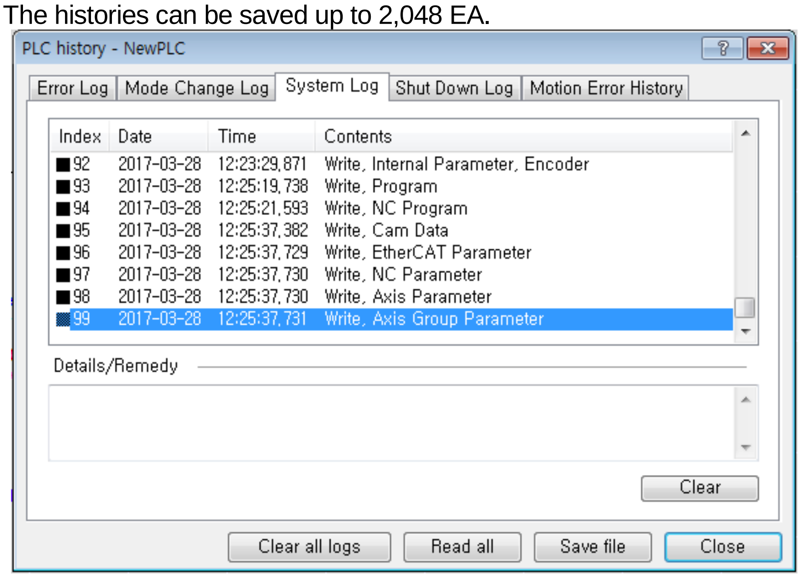
It saves the operation history of the system occurred during operation.

- 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.



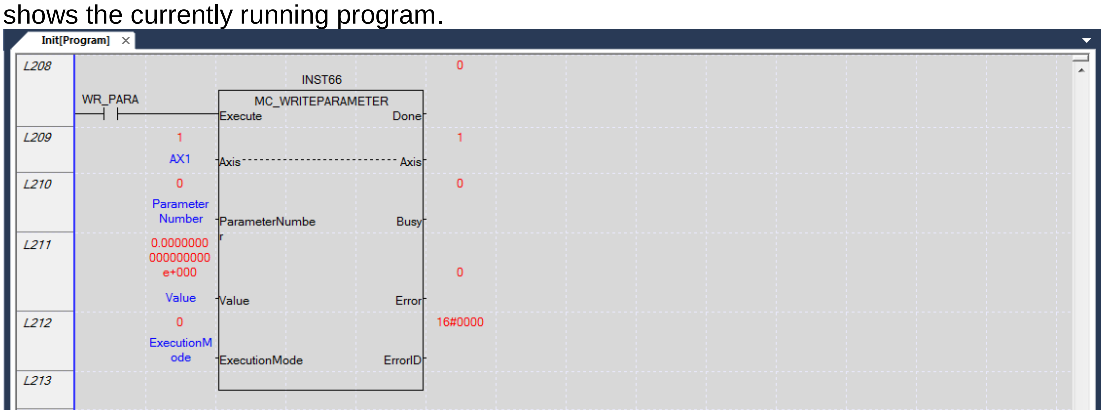
10.8 Program Modification during Operation(Modification during RUN)

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.

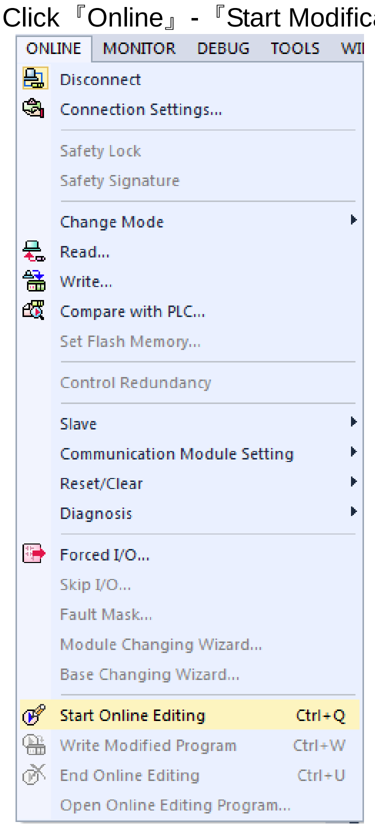
The items that can be modified during RUN are limited to programs, network parameters.
You cannot modify adding tasks, deletion, parameters, etc. during RUN.

10.8.1 Modification Procedures during RUN

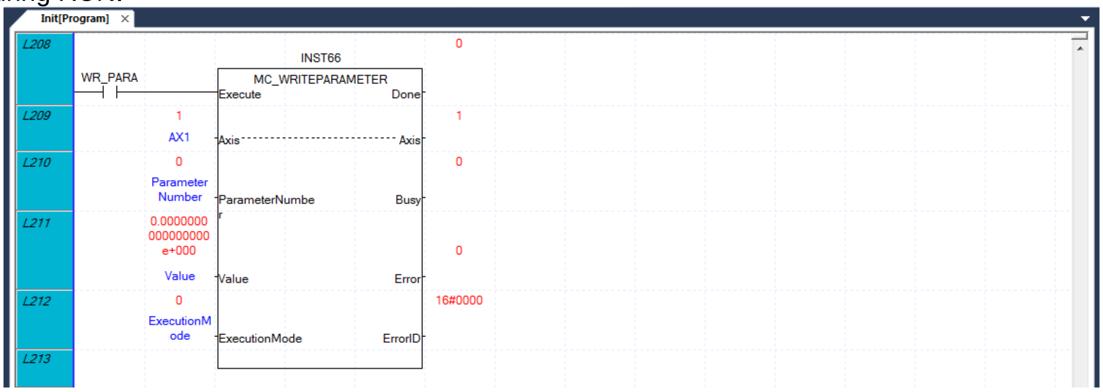
(1) It shows the currently running program.



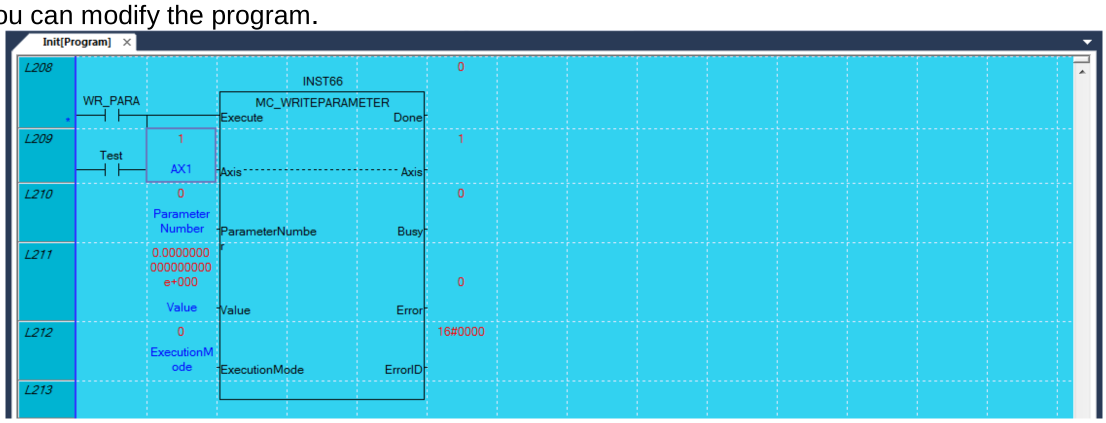
(2) Click『Online』-『Start Modification During RUN』.



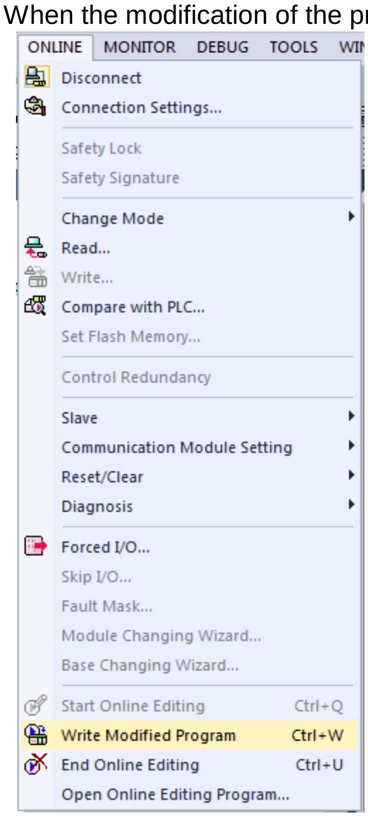
(3) Then, the background color of the program window changes and it is converted into the mode of modification during RUN.

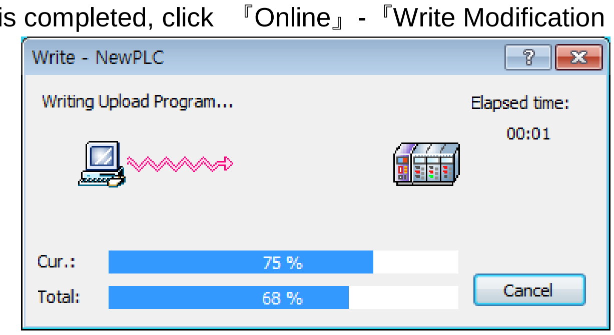


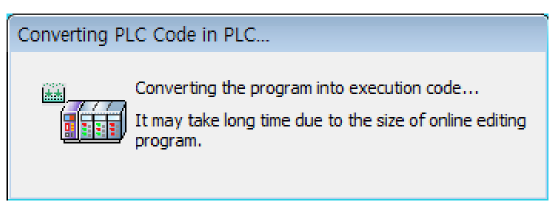
(4) You can modify the program.

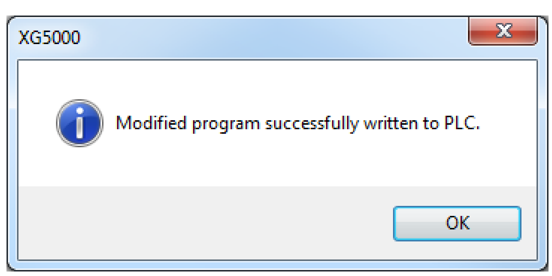


(5) When the modification of the program is completed, click 『Online』 - 『Write Modification During RUN』

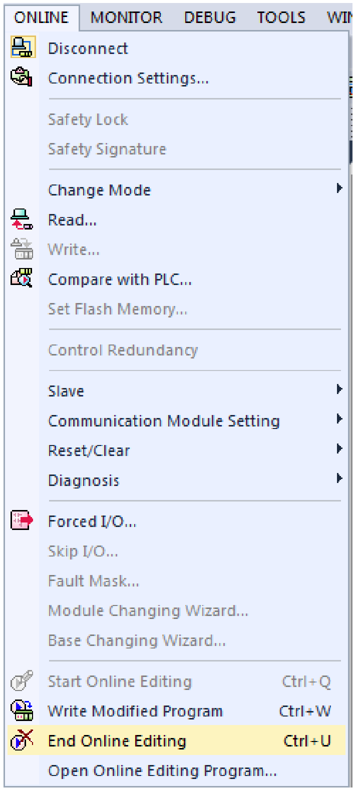


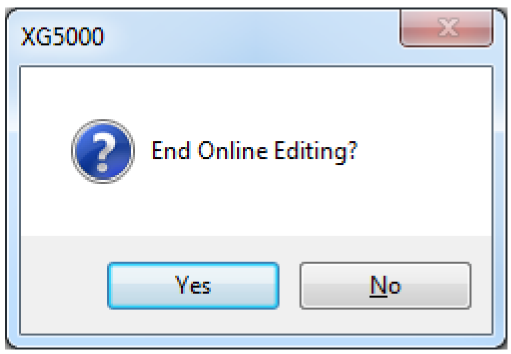




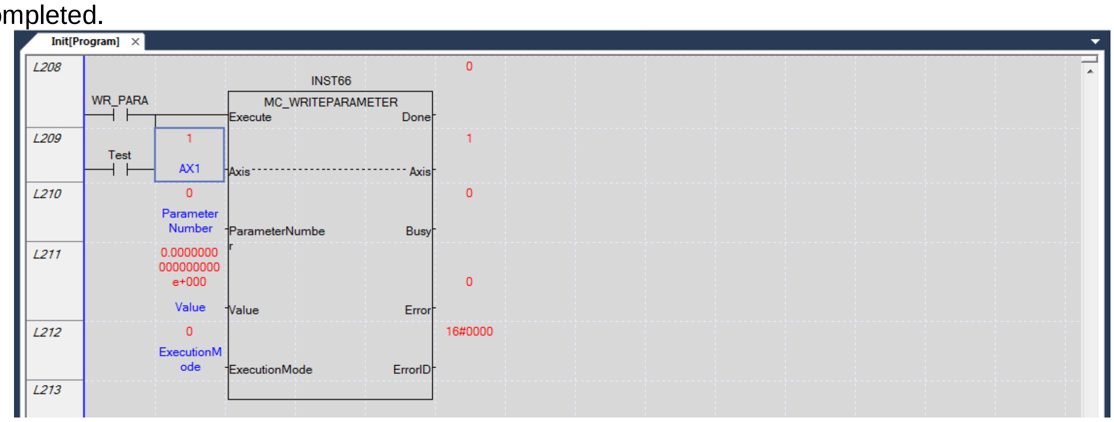


(6) When Write Program is completed, click 『Online』 - 『End Modification During RUN』.





(7) The background color of the program window changes into the original one and modification during RUN is completed.



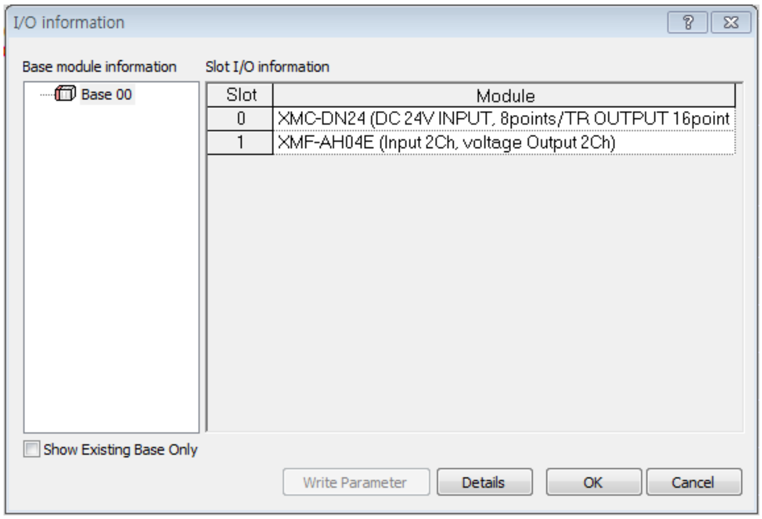
Notice

• 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.

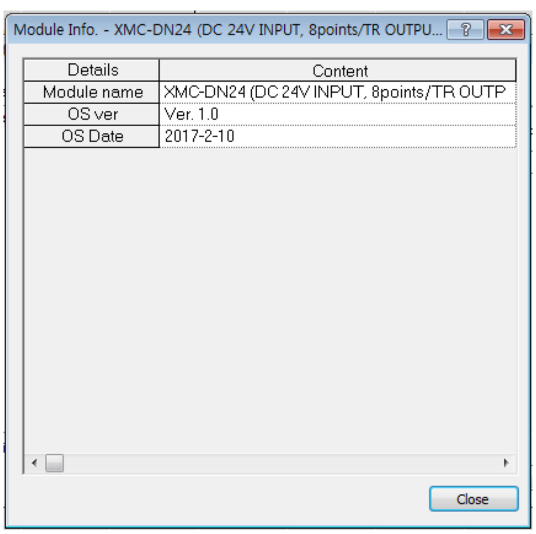
10.9 Read I/O Information

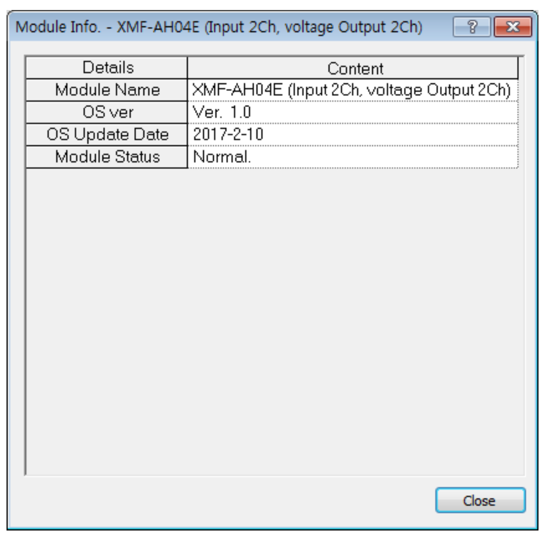
It is the function to monitor each module's information comprising the motion controller system.

(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.

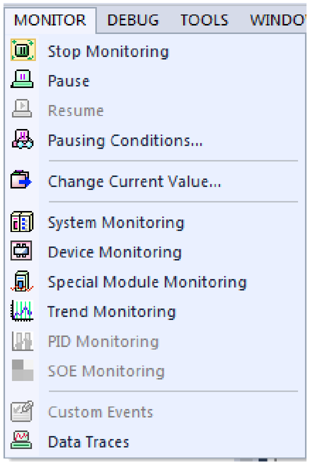




10.10 Monitoring Functions

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.

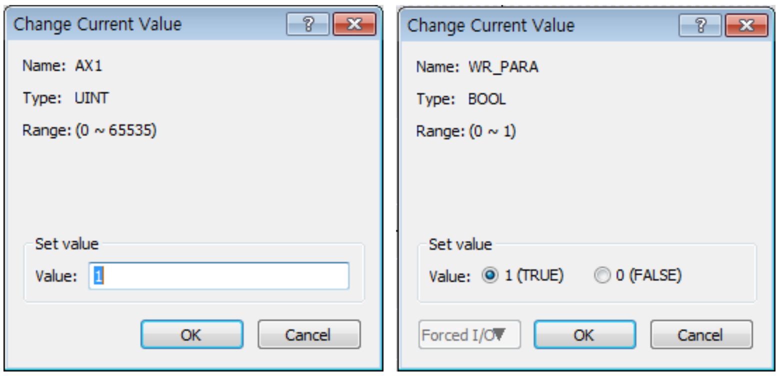


(2) The below table provides the descriptions on each item.

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.	

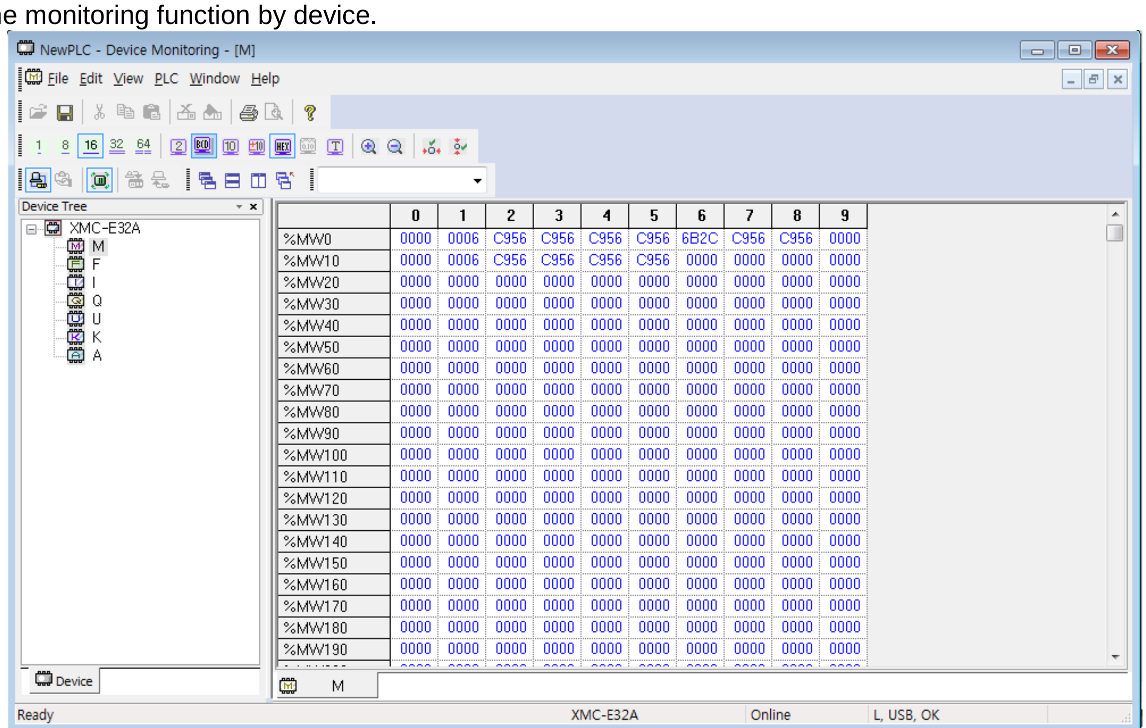
(a) Changing the current value

It is the function to change the current value of each selected device in the program window.



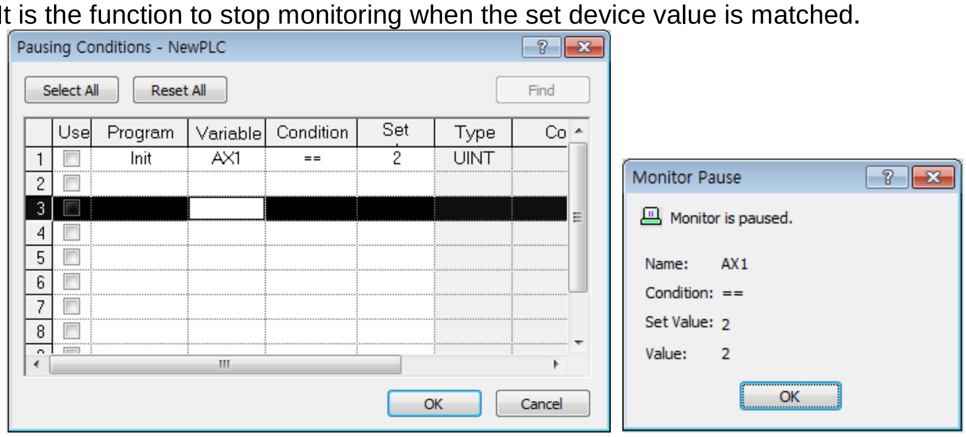
(b) Device monitor

It is the monitoring function by device.



(c) Monitor suspension setting

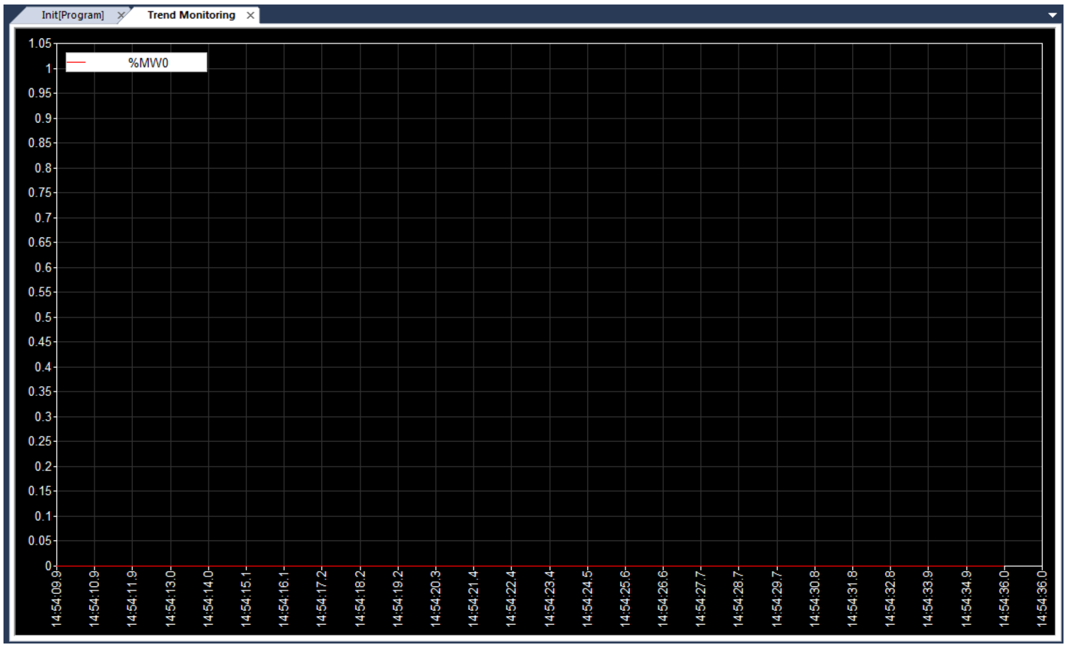
It is the function to stop monitoring when the set device value is matched.



(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

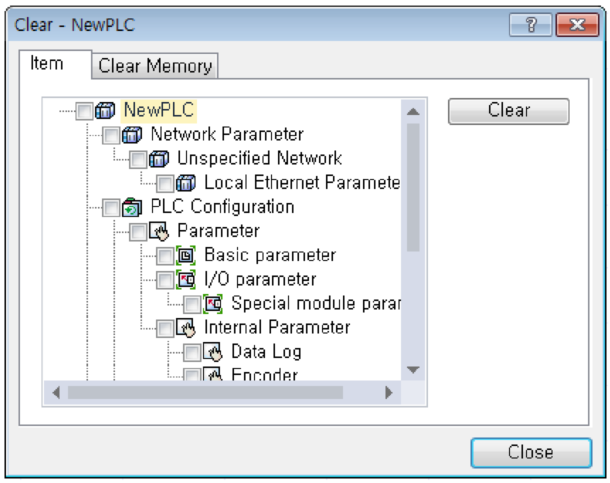
For more details on the monitoring function, refer to XG5000 manual.

10.11 Function to Delete All of the Motion Controller

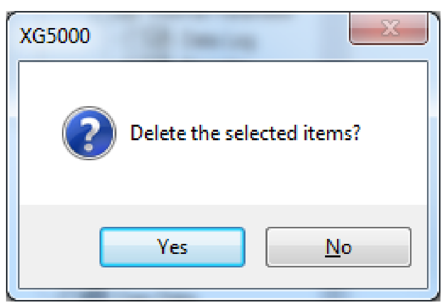
The function to delete all of motion controller is the initialization function to delete all programs, parameters, passwords, data stored in the motion controller.

(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

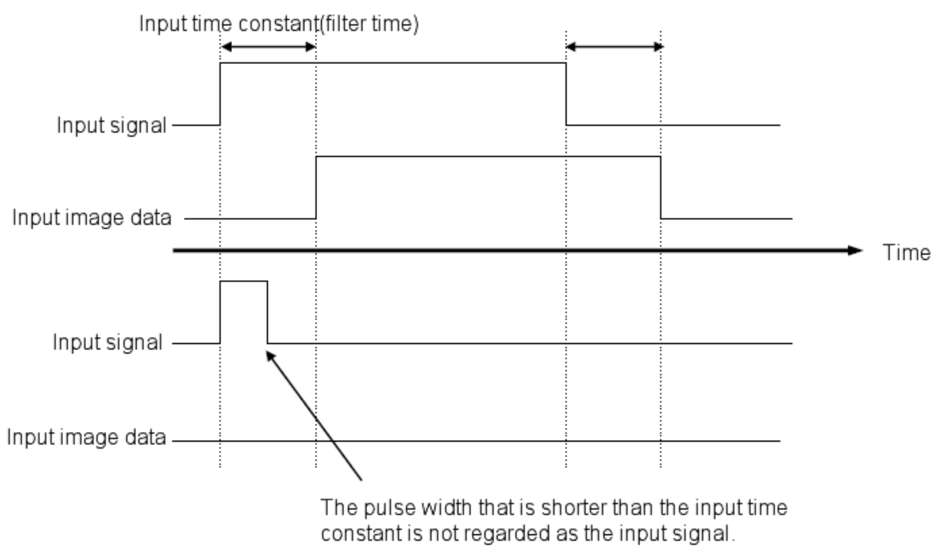
10.12.1 Input Filter Function

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

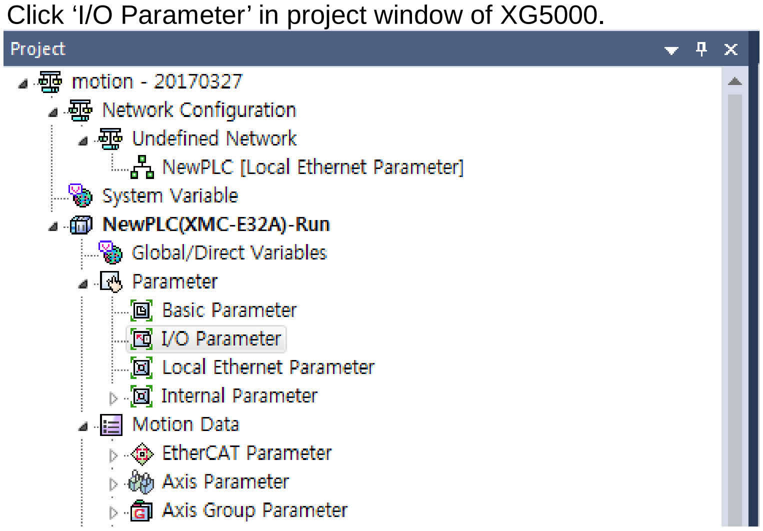
Under the environment with serious noise or in the case of the equipment that is greatly affected by the input signal's pulse width, the system may receive incorrect input depending on the input signal status. To prevent such incorrect input, the input filter function does not regard the signal that 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.

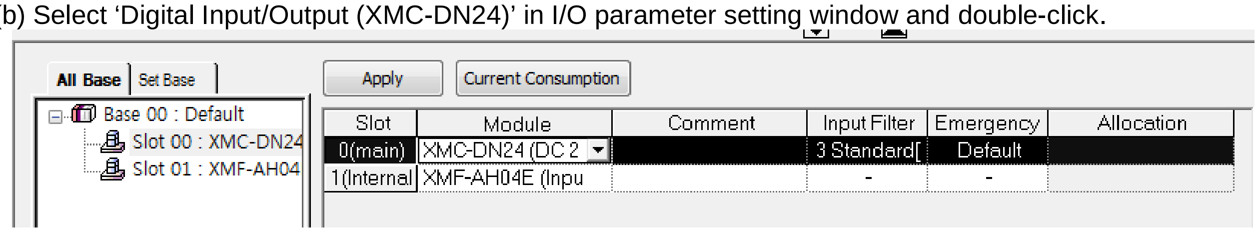


(2) Input filter setting method

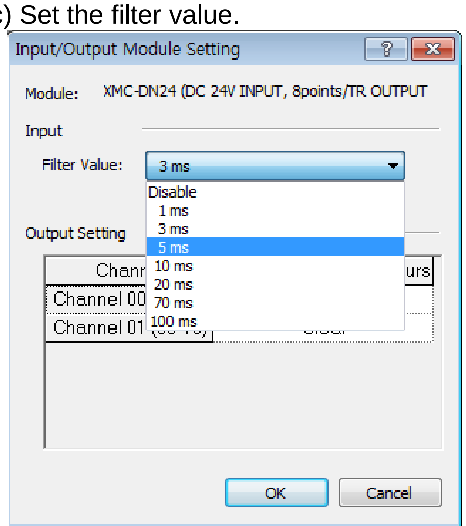
(a) Click 'I/O Parameter' in project window of XG5000.



(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.

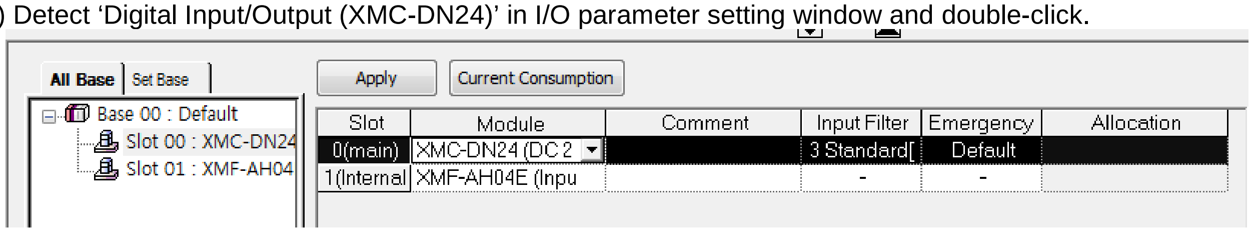
For example, if the filter value is set to 3ms, the cycle of the main task should be set to 1ms or 2ms.

10.12.2 Emergency Output Function

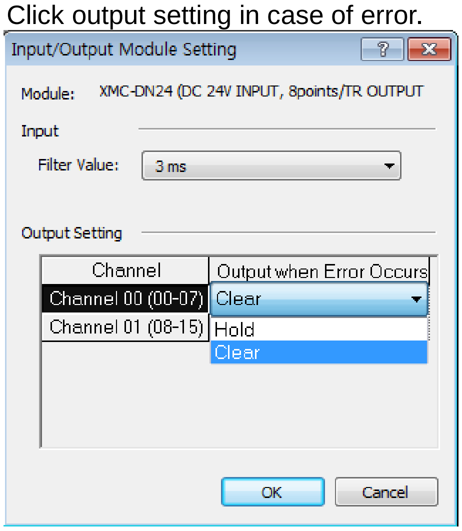
The XMC's output module supports the emergency output function to determine whether maintaining the output status of the output module or clearing it when the motion controller is stopped due to errors.
You can set the emergency output by 8 points. For more details on how to set the emergency output, 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.

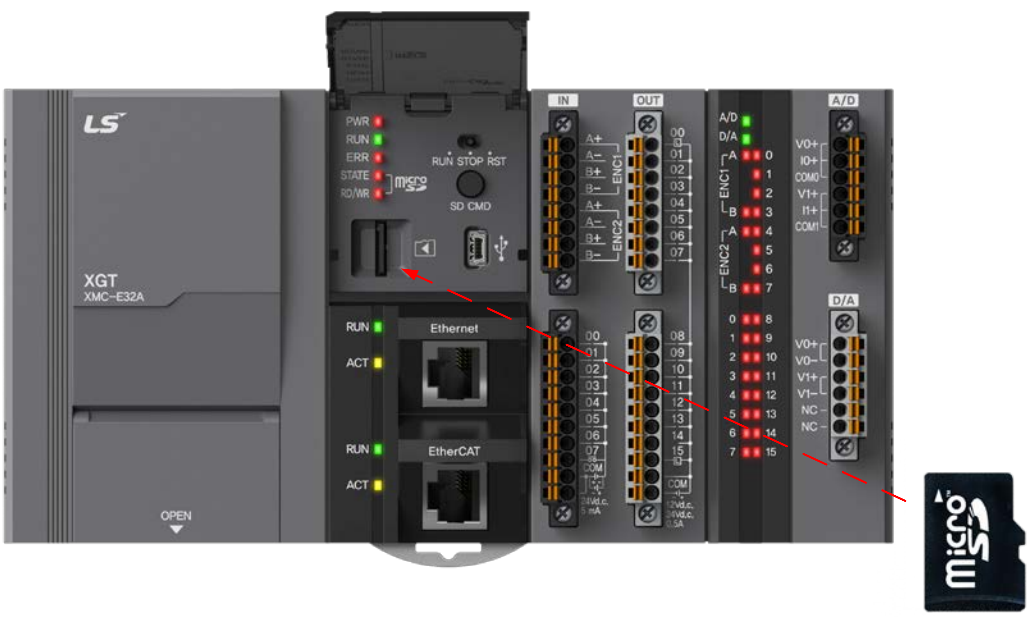
11.1.1 Features

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

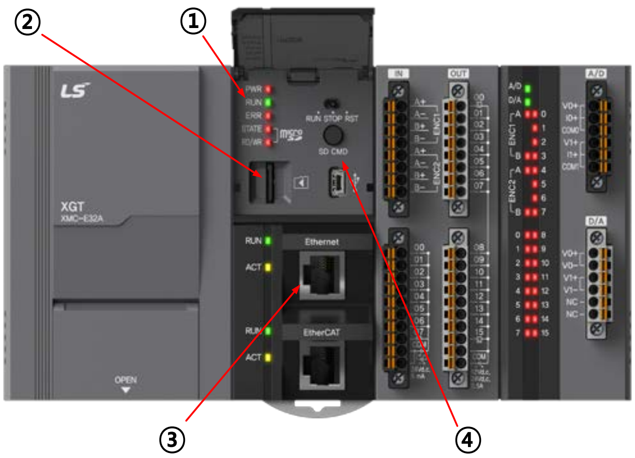
(4) FTP Interface

Files saved in the datalog can be read remotely using FTP, making it easier to verify data fluctuations.

11.1.2 Part Names

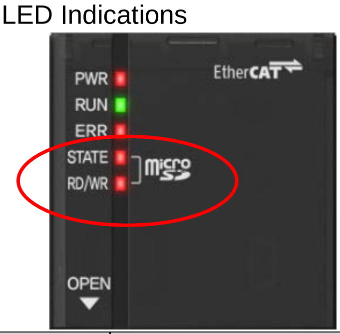
The names of pars related to datalog function are as follows.

(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

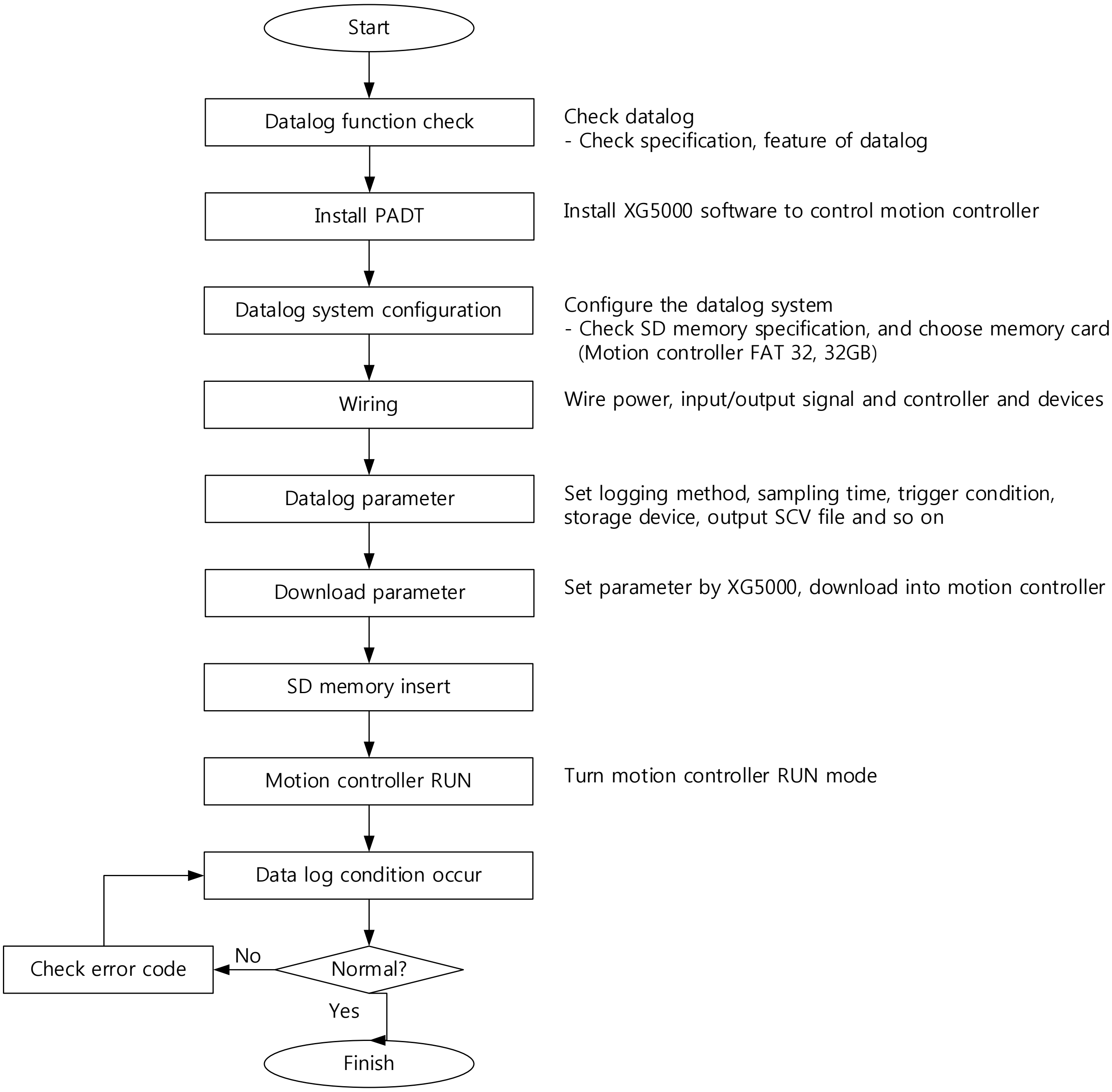
(2) LED Indications



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.

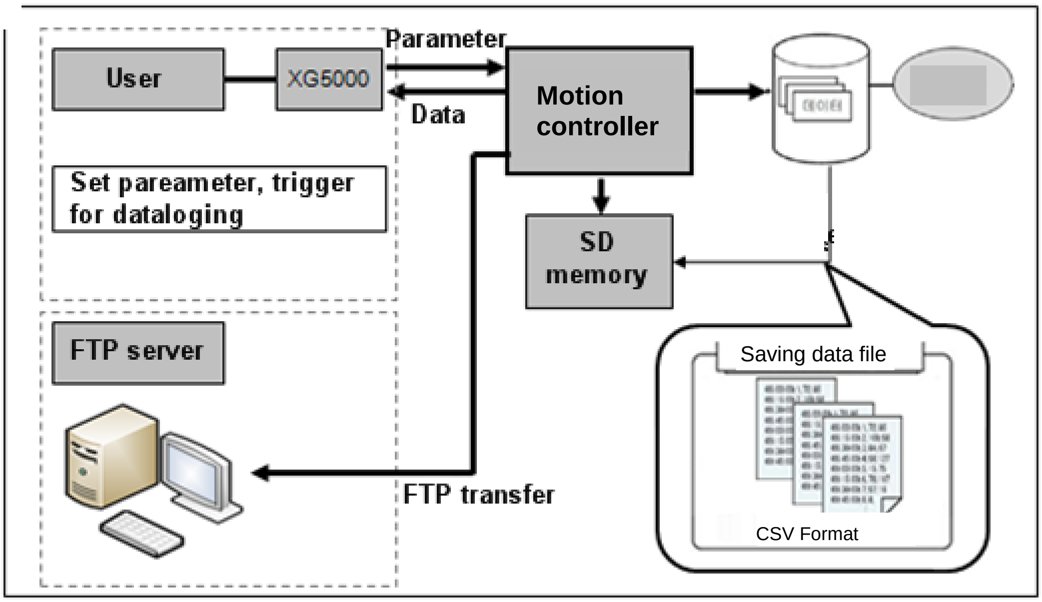


Note

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

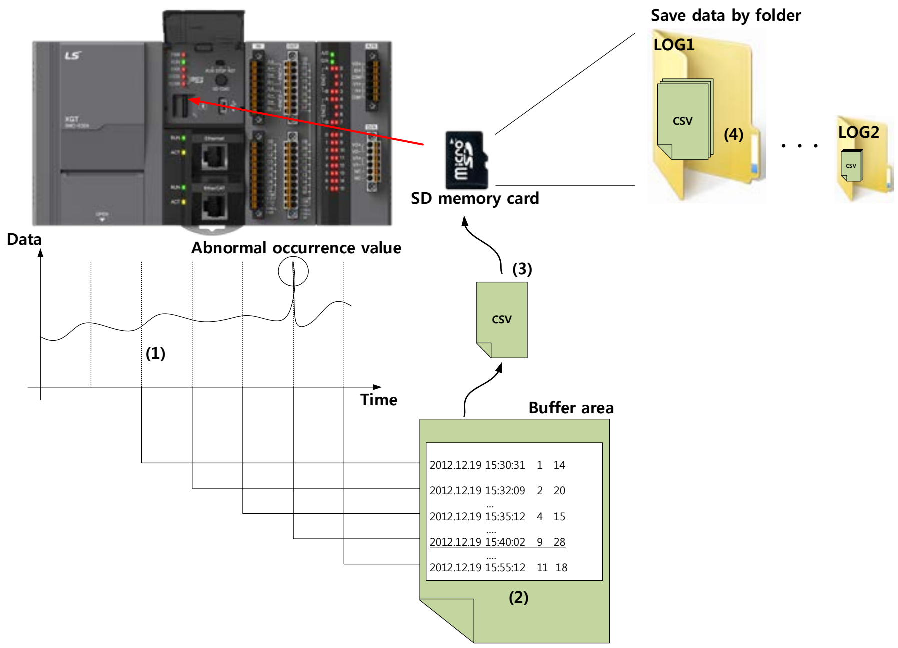
Items			Specifications	Note
Function Configuration	Group Configuration		Up to 16 groups	
	Configuration Data		Up to 64 per group	
	Data Collection Type		regular / trigger / event	
	File Format		CSV	
	File Size		Up to 16MByte	
	Data Type		BIT, BYTE, WORD, DWORD, LWORD, SINT, INT, DINT, LINT USINT, UINT, 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

ASCII Code	Indication	ASCII Code	Indication	ASCII Code	Indication	ASCII Code	Indication
0x20	SP	0x2A	*	0x3E	>	0x7B	{
0x21	!	0x2B	+	0x3F	?	0x7C	|
0x22	“	0x2D	-	0x41 ~ 0x5A	English (upper case)	0x7D	}
0x23	#	0x2E	.	0x5B	[	0x7E	~
0x24	$	0x2F	/	0x5C	\		
0x25	%	0x30 ~ 0x39	Number	0x5D	]		
0x26	&	0x3A	:	0x5E	^		
0x27	‘	0x3B	;	0x5F	_		
0x28	(	0x3C	<	0x60	`		
0x29	)	0x3D	=	0x61 ~ 0x7A	English (lower case)		

(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

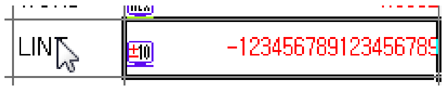
- When converting character strings, characters which do not correspond with ASCII (see 11.3.1) are all converted to Null.

Sequence	#16	#15	#14	…	#1
Device	%MW14	%MW13	…	%MW0	%MW14

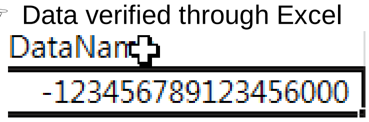
Note

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

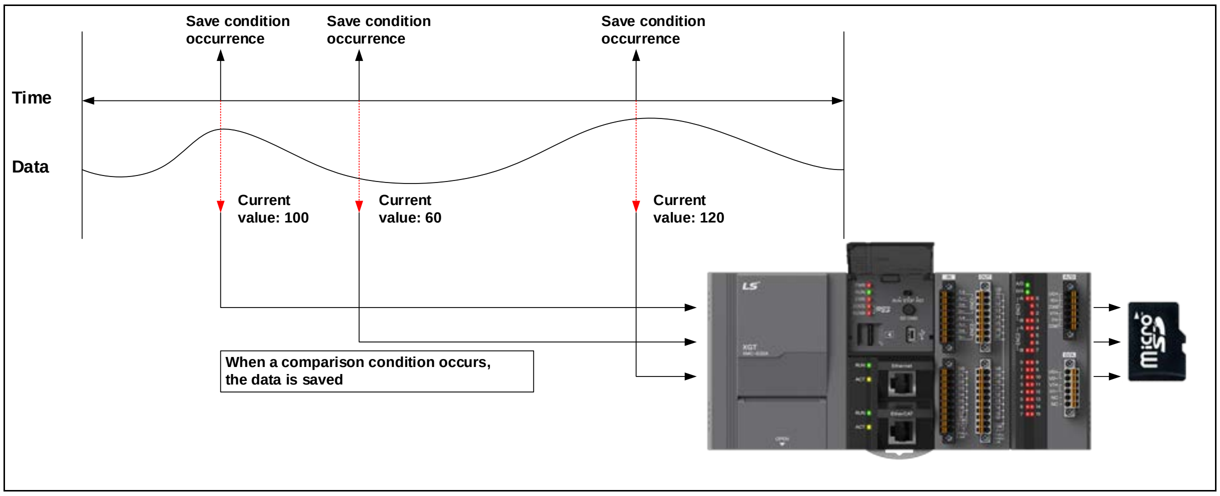
In case of REAL, LREAL types, –NaN, +NaN are saved for undefined data, and -INF, +INF character strings are saved for data with infinite range.
Please verify the data save range before use.

11.3.2 Data Save Method

The datalog function saves data using one of the three methods that follows.

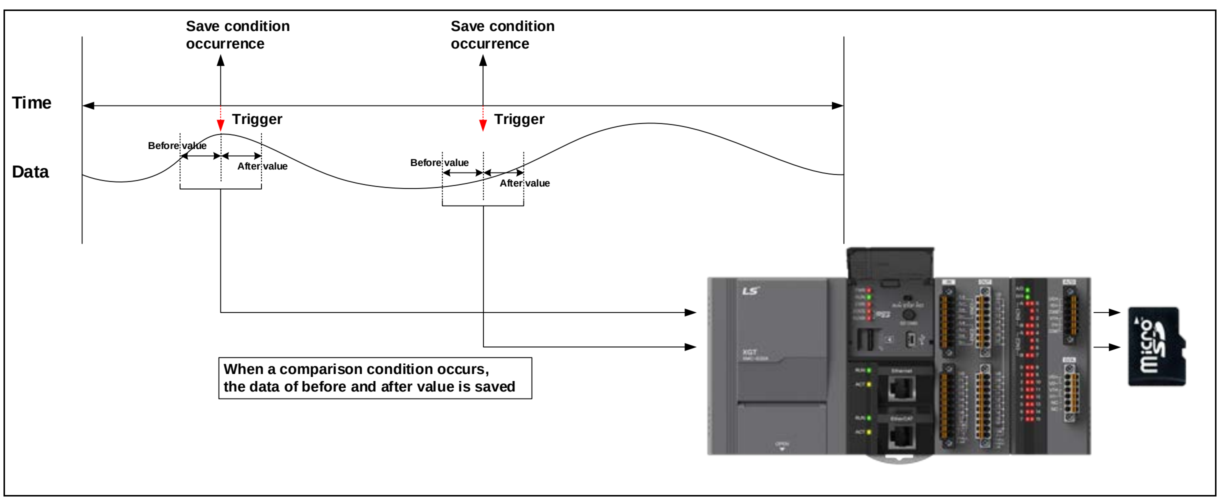
(1) Regular Save

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.

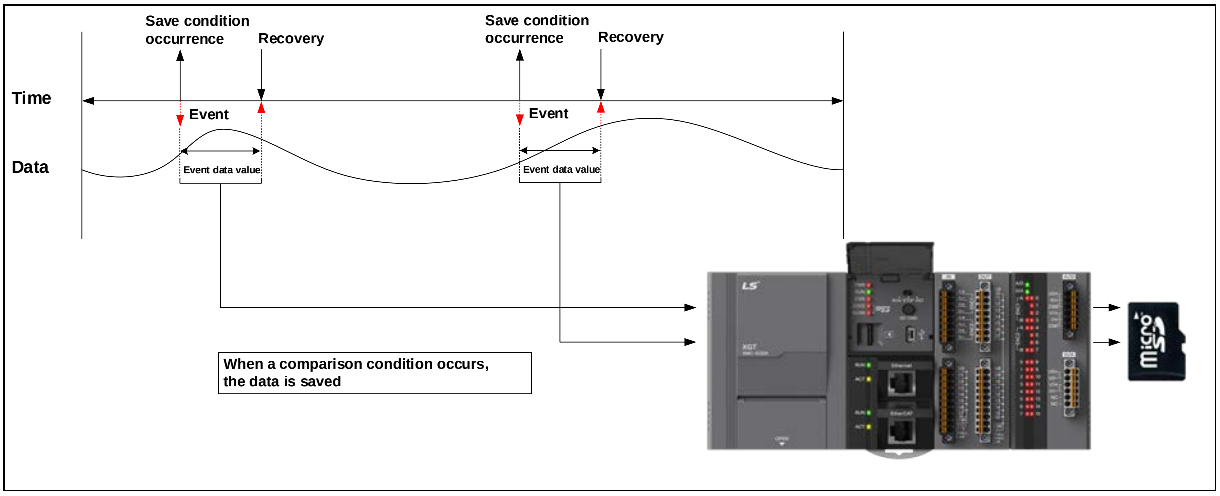


(2) Trigger Save

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.



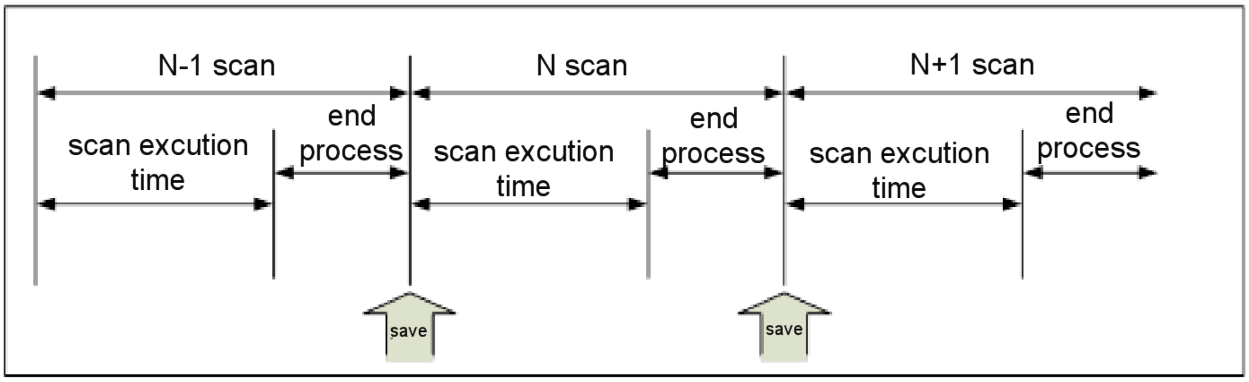
11.3.3 Data Sampling Condition

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.



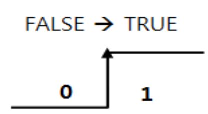
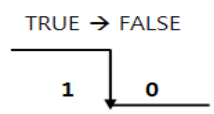
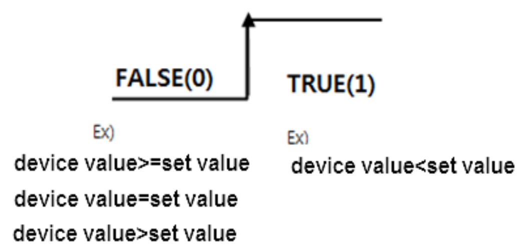
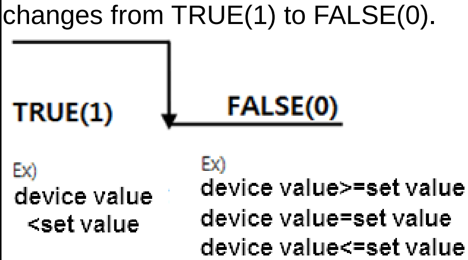
(b) Designation Cycle Save
It samples data when a set interval arrives.

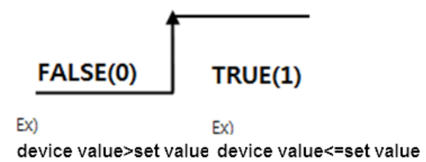
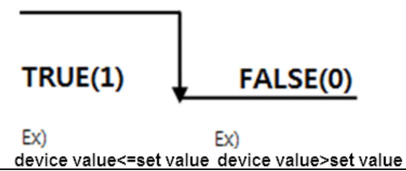
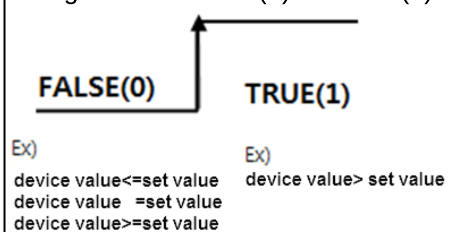
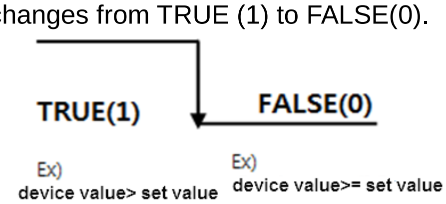
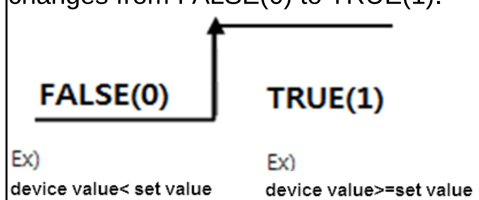
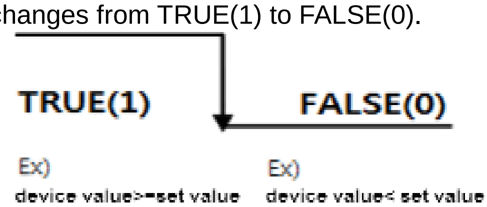
(c) Designation Time Save
It samples data when a set interval arrives.

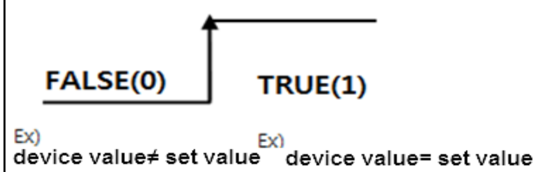
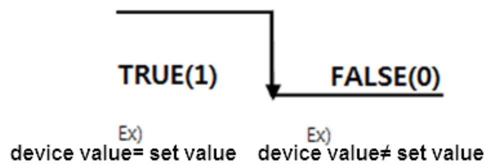
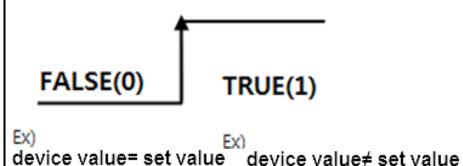
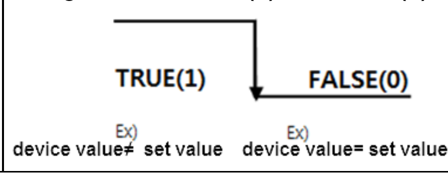
Note

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.

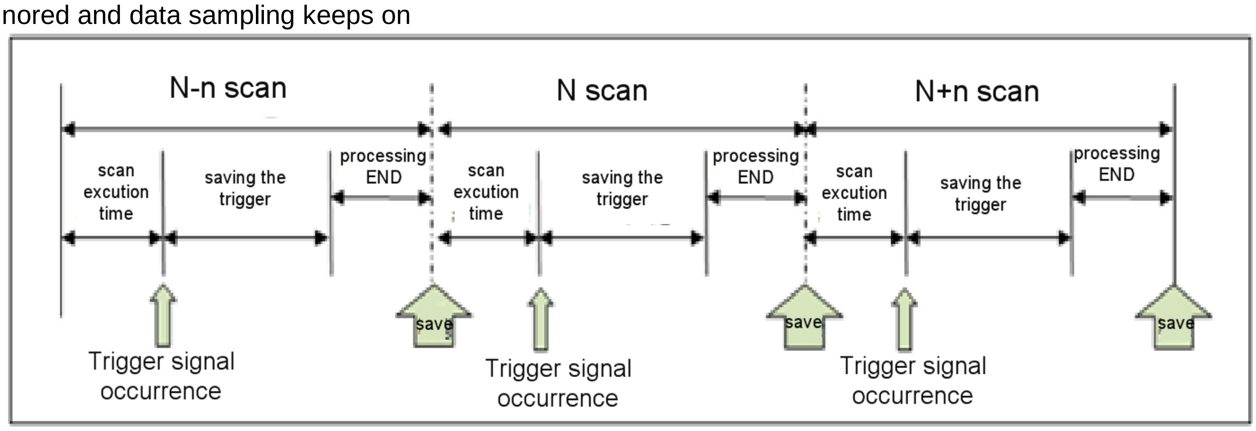
(2) Trigger Save
Save data in the preset number of collection data. The following are condition setting items for Regular Save.

<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).  	

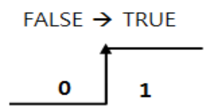
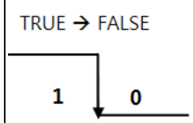
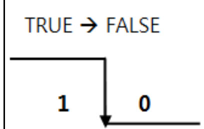
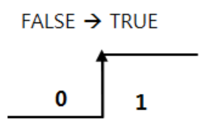
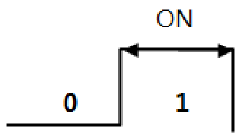
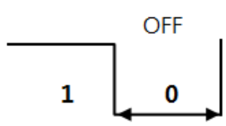
	Trigger Occurrence Condition	Device Set Condition	Operation	Note
Word Condition	Elevation	Same	Samples data at the elevation edge when Device Set Condition changes from FALSE(0) to TRUE(1). 	
	Descent		Samples data at the descent edge when Device Set Condition changes from TRUE(1) to FALSE(0). 	
	Elevation	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). 	

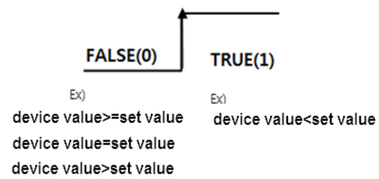
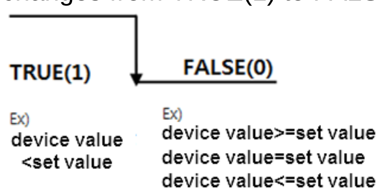
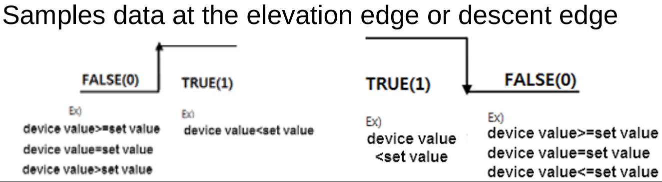
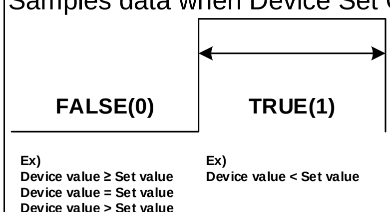
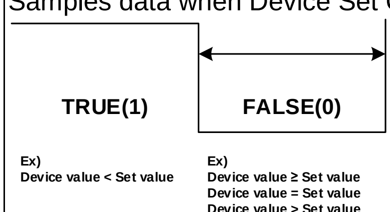
Trigger occurrence condition is decided by main task END. If trigger occurs again when data sampling, the trigger is ignored and data sampling keeps on

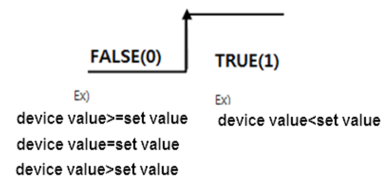
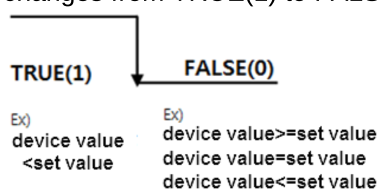
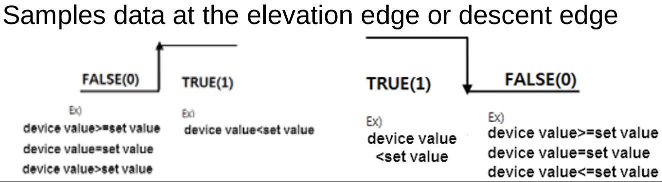
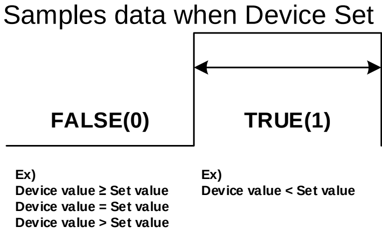
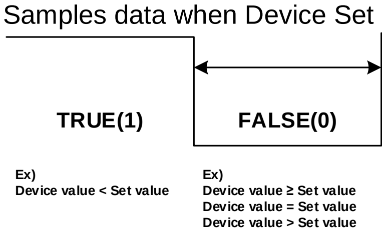


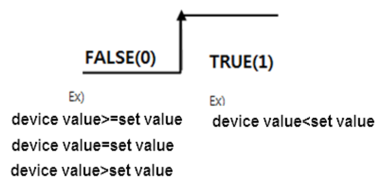
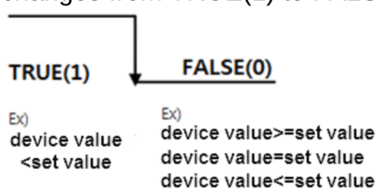
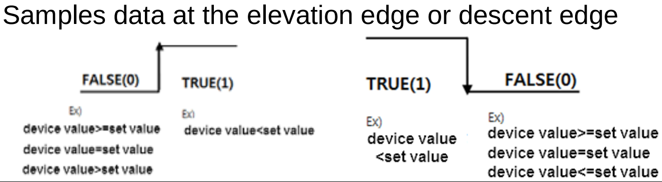
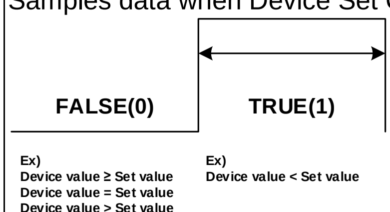
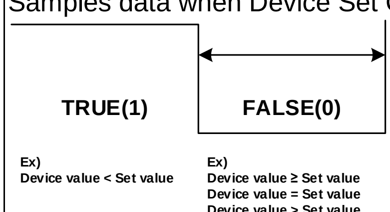
(3) Event Save

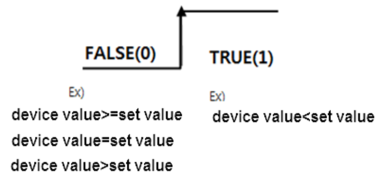
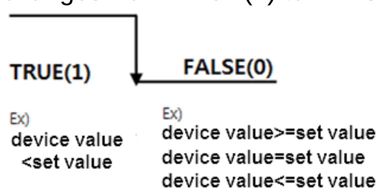
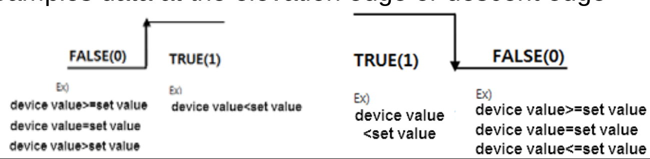
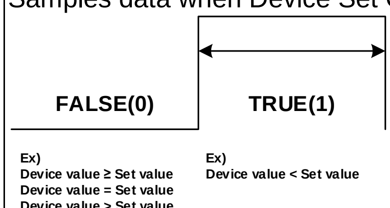
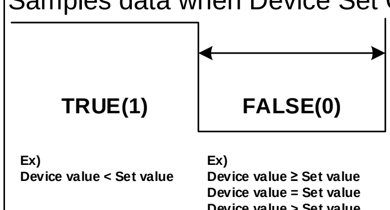
Event Save runs with similar conditions to Trigger Save. Event Save refers to saving data when the event occurs, until the conditions are not satisfied.

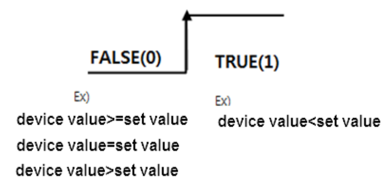
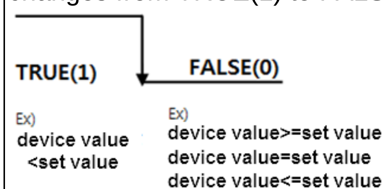
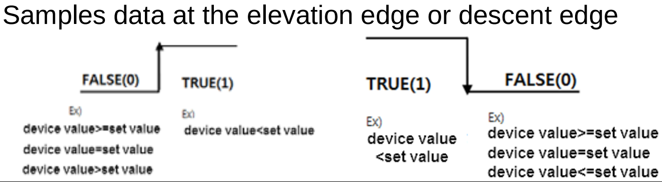
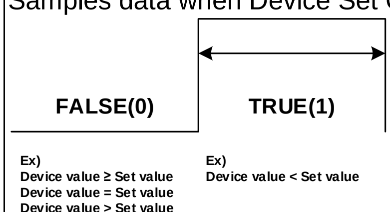
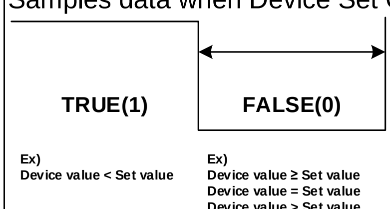
	Trigger Occurrence Condition	Device Set Condition	Operation	Release Value Setting
Bit Condition	Elevation		Saves data at elevation edge of set device bit value 	
	Descent		Saves data at descent edge of set device bit value 	
	Transfer		Saves data when set device bit value is transferred  or 	
	ON		Saves data when set device bit value is ON 	
	OFF		Saves data when set device bit value is OFF 	

	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) 	

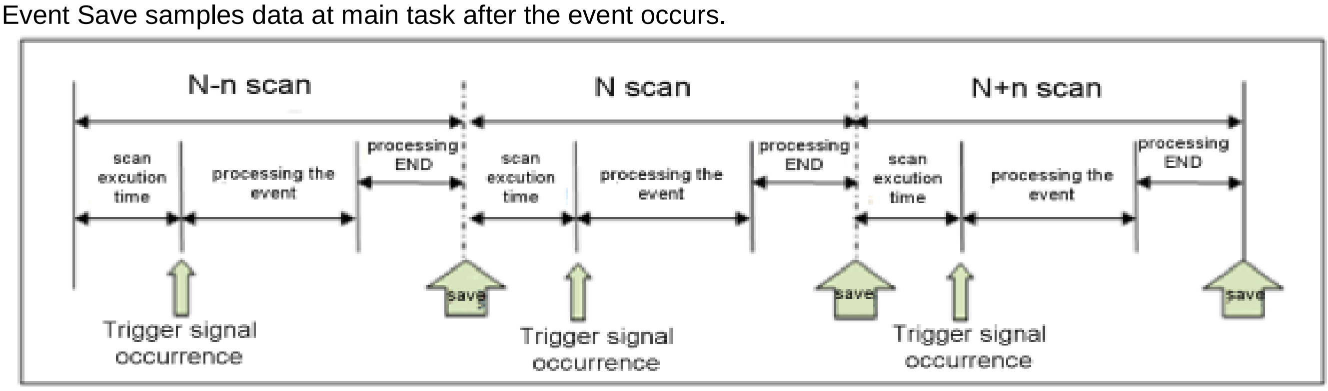
	Trigger Occurrence Condition	Device Set Condition	Operation	Release Value Setting
Word Condition	Elevation	Large	Samples data at the elevation edge when Device Set Condition changes from FALSE (0) to TRUE(1). 	Setting Available
	Descent		Samples data at the descent edge when Device Set Condition changes 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) 	

<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>

<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">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>

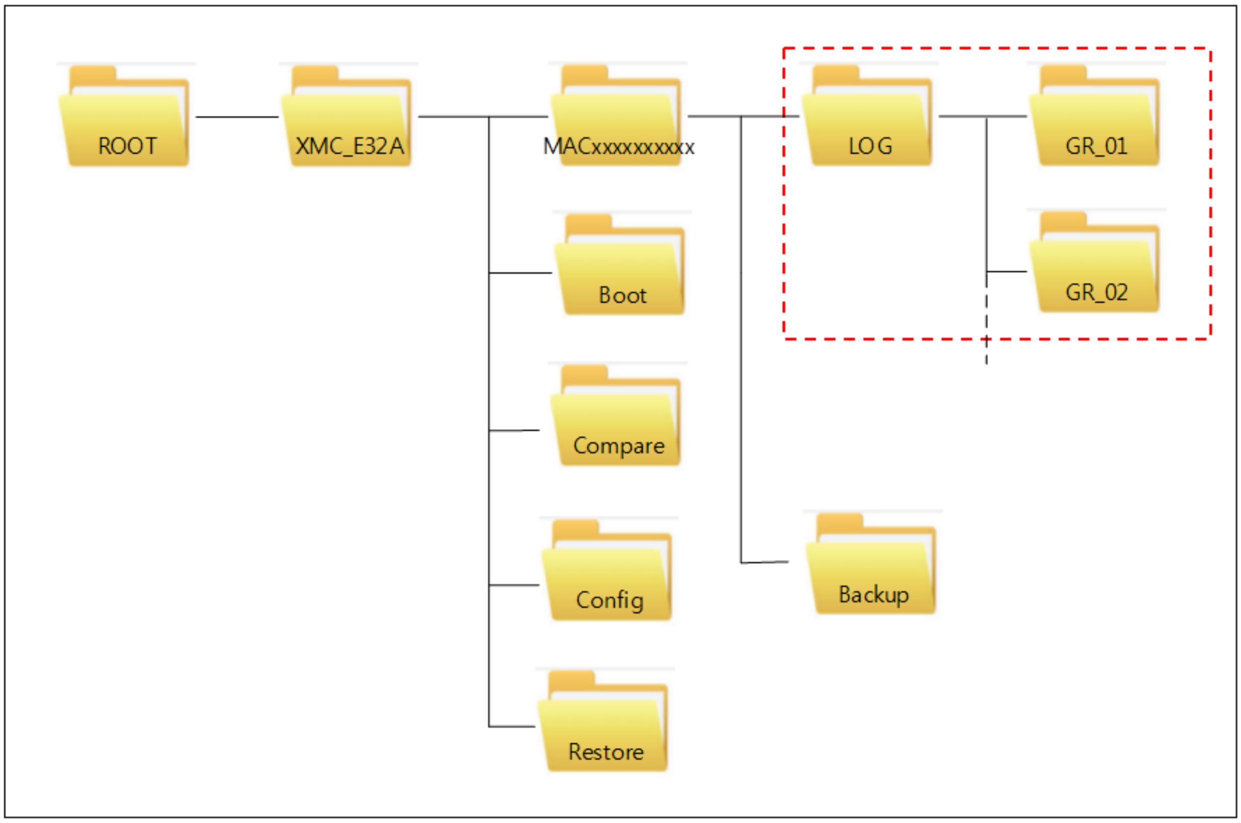
	Trigger Occurrence Condition	Device Set Condition	Operation	Release Value Setting
Word Condition	Elevation	Not Same	Samples data at the elevation edge when Device Set Condition changes from FALSE (0) to TRUE(1). FALSE(0) TRUE(1) Ex) device value>=set value, device value=set value, device value>set value 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 (rows 1–7)

Data File (rows 9–11)

	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

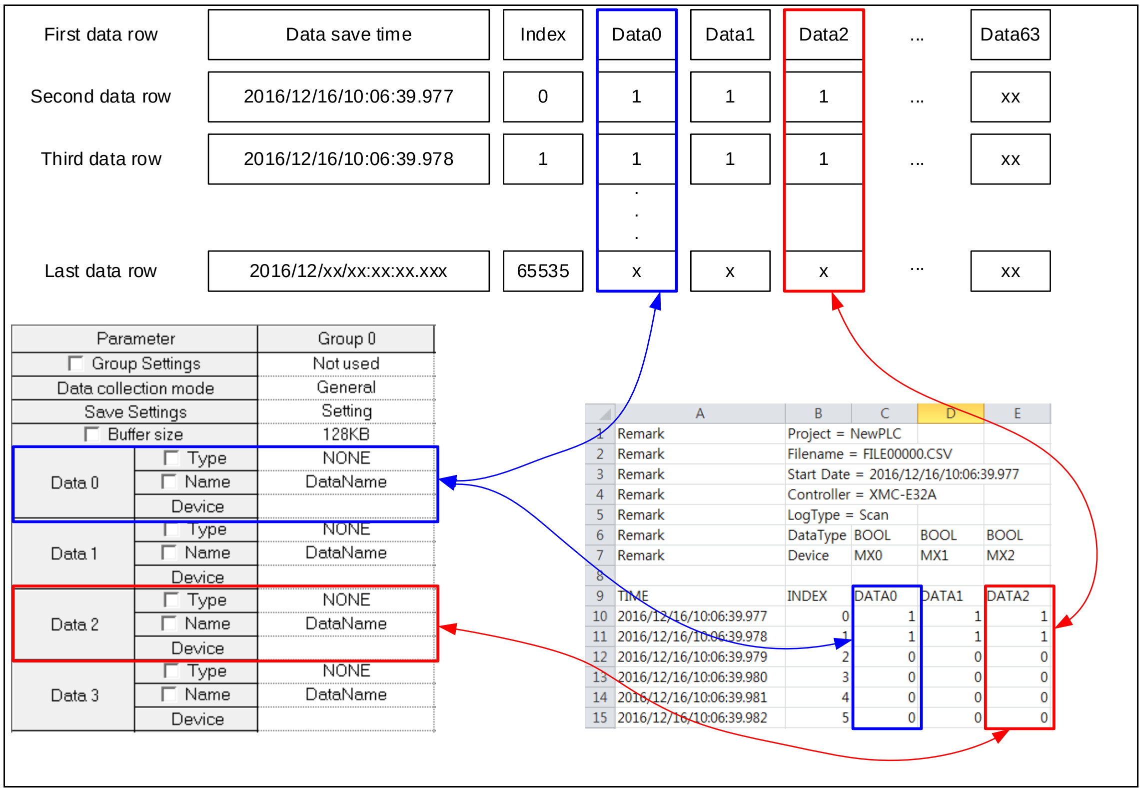
(1) Header File Structure

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

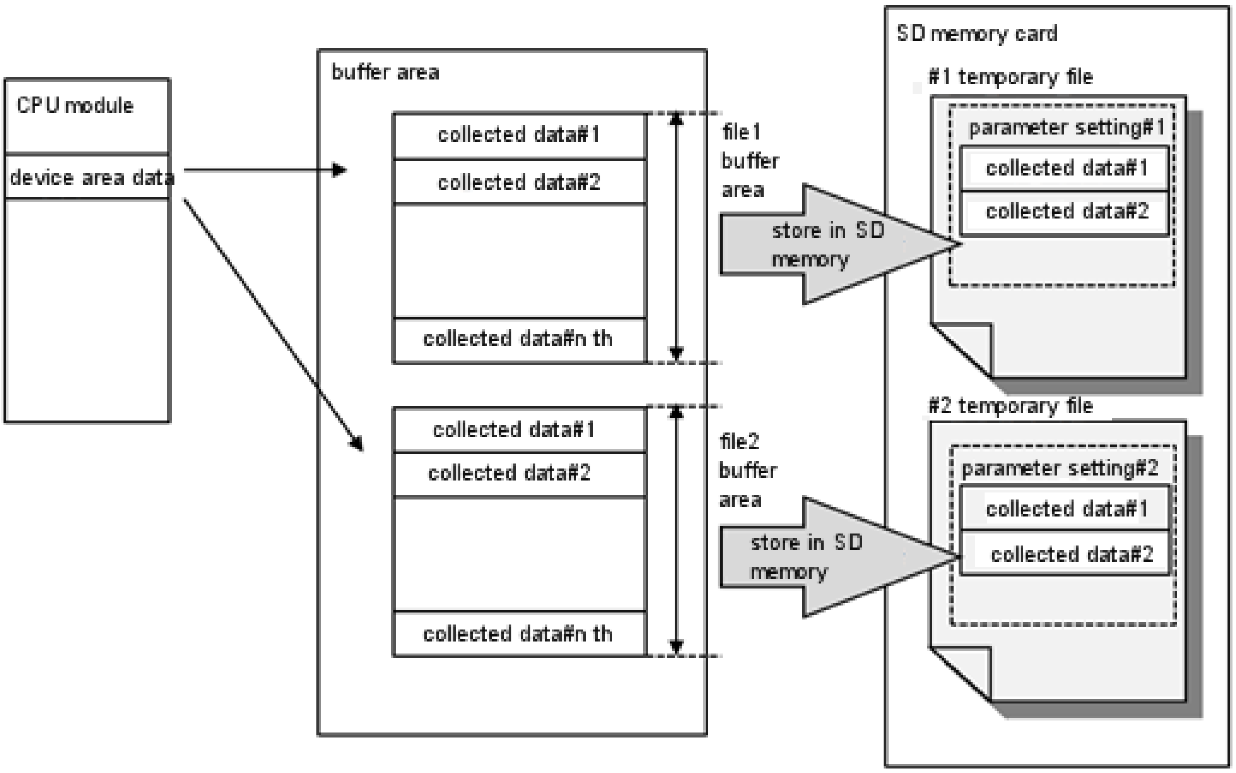
String Name	Output	Size (Word)
Temporary String	Indicates date and time with fixed characters	5
Index String	Indicates index name	2
Data String	Outputs the data name designated at data setting	1-64 (Depends on parameter setting)

(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.

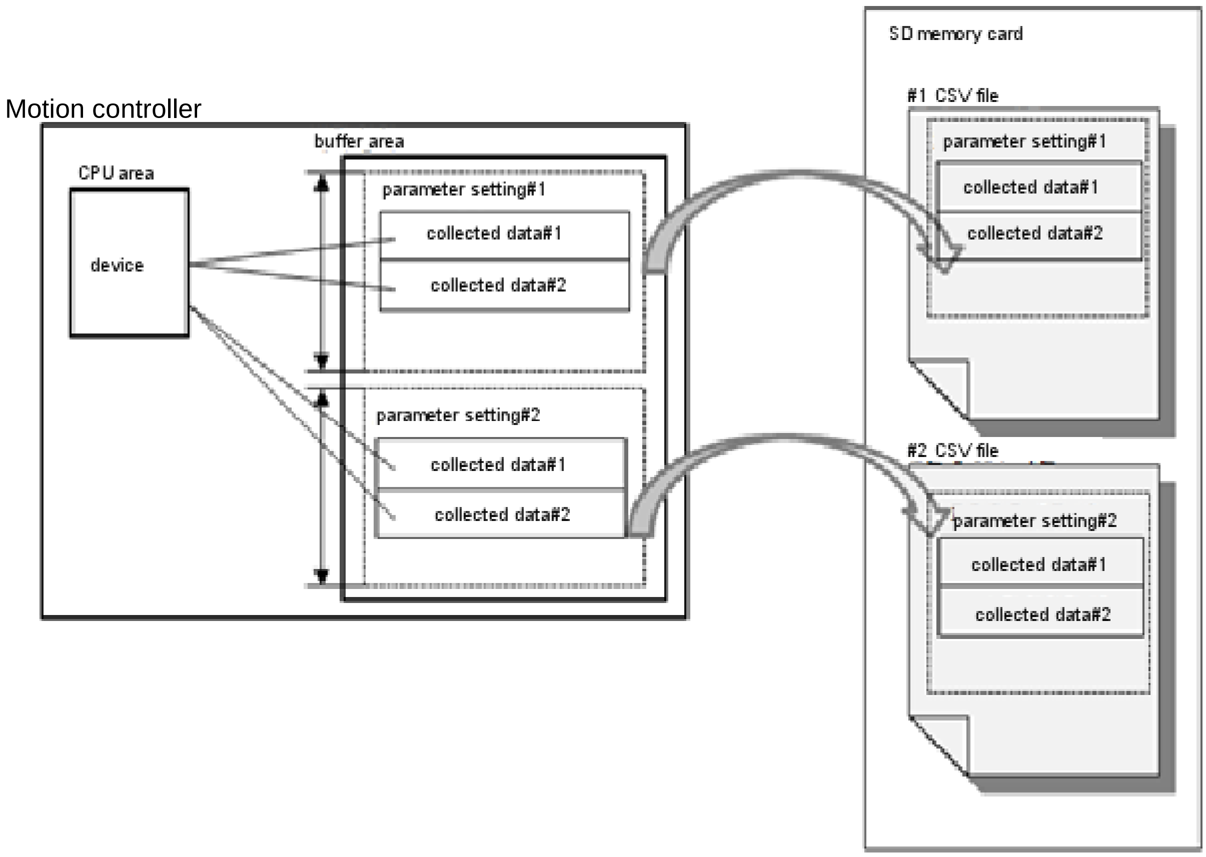
Operation Setting in Case of Excess	Operation	Note
Maintains the latest history	Overwrites and saves new data over the oldest file	
Maintains the initial history	Performs no more file saving	

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

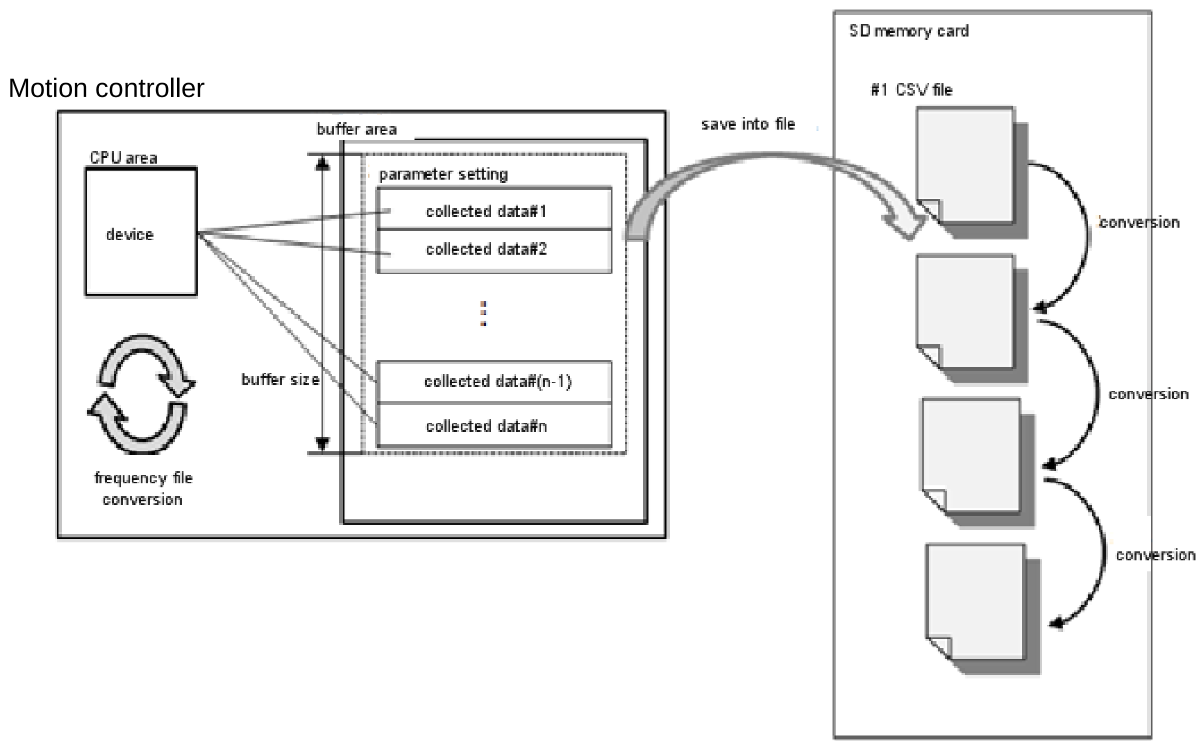
Data omission refers to situation where normal data collection is not possible. If data collection interval is set too short, data sampling might not be performed at every set interval, which in turn might cause data omission. Cases include the following.

(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.

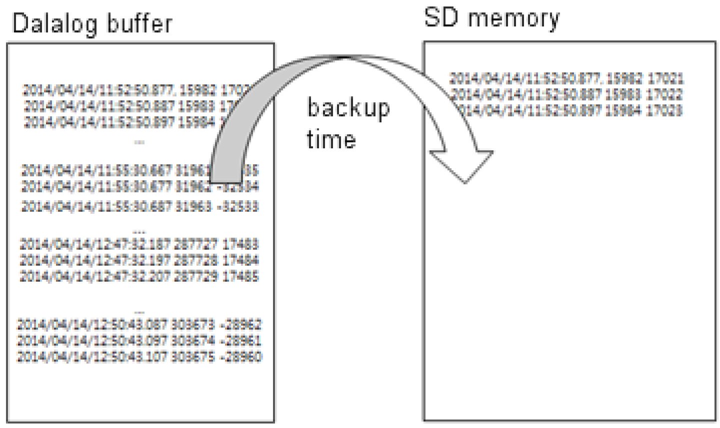
(2) Too Frequent File Conversion

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.

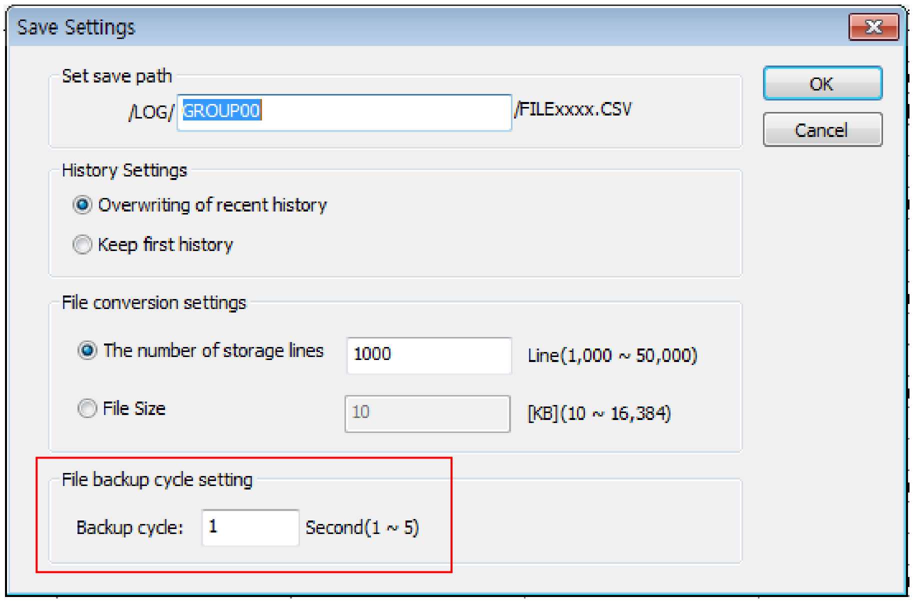


11.3.9 Files Backup Cycle

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.

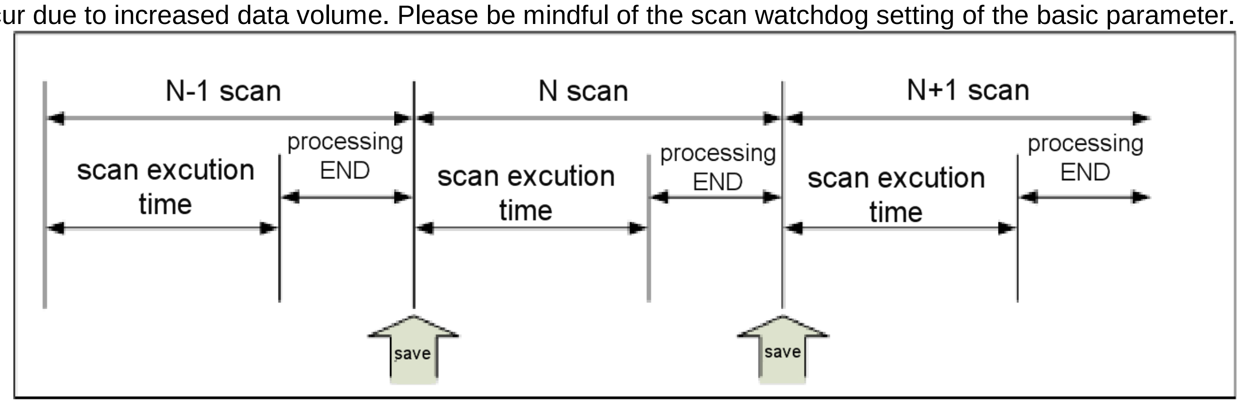


11.4 Regular Save

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.

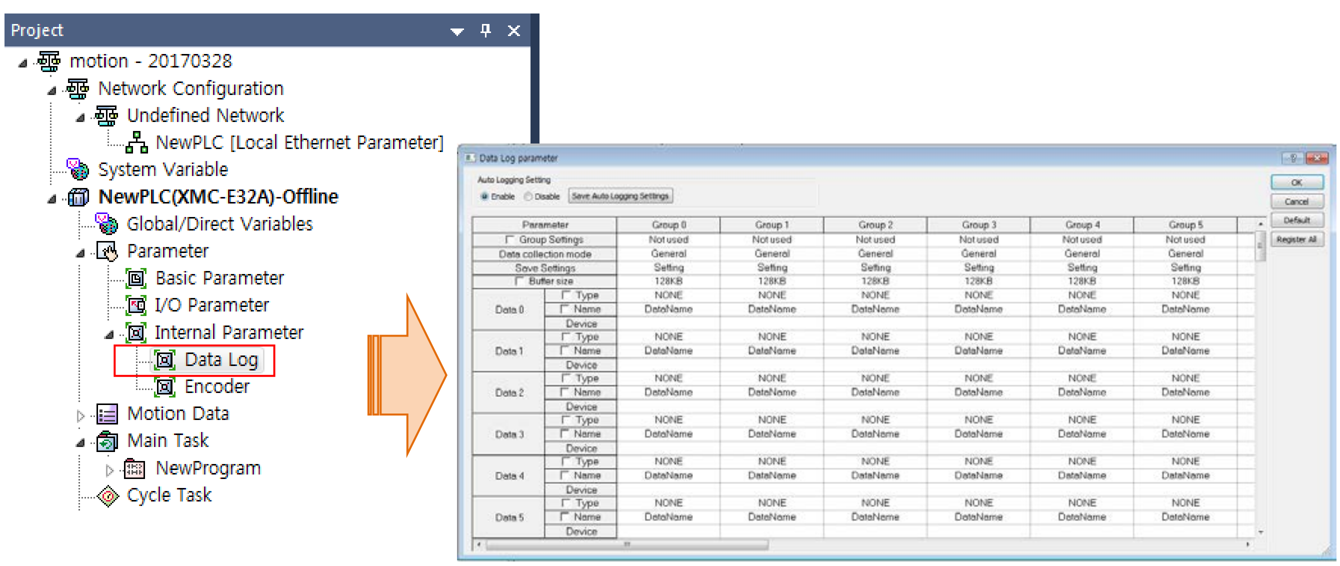
11.4.1 Save Method

(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.

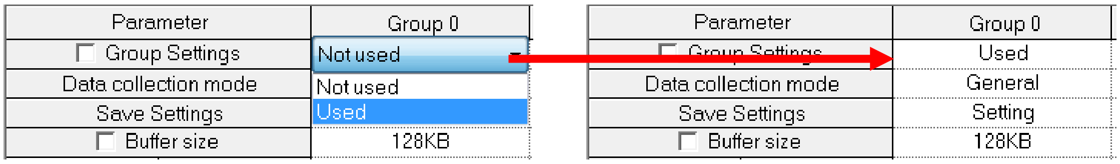


(2) Setting Method

(a) Choose XG5000 – [Project Window] - [internal parameter] - [datalog]
This activates the datalog parameter setting window.



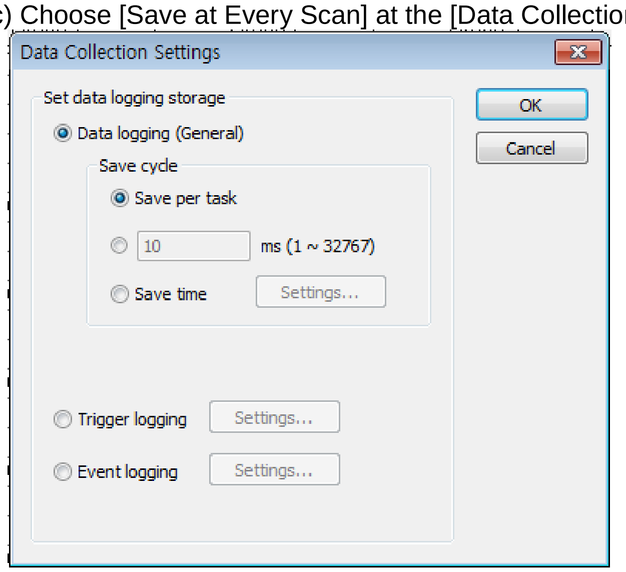
(b) Set the group to use on the datalog parameter window.



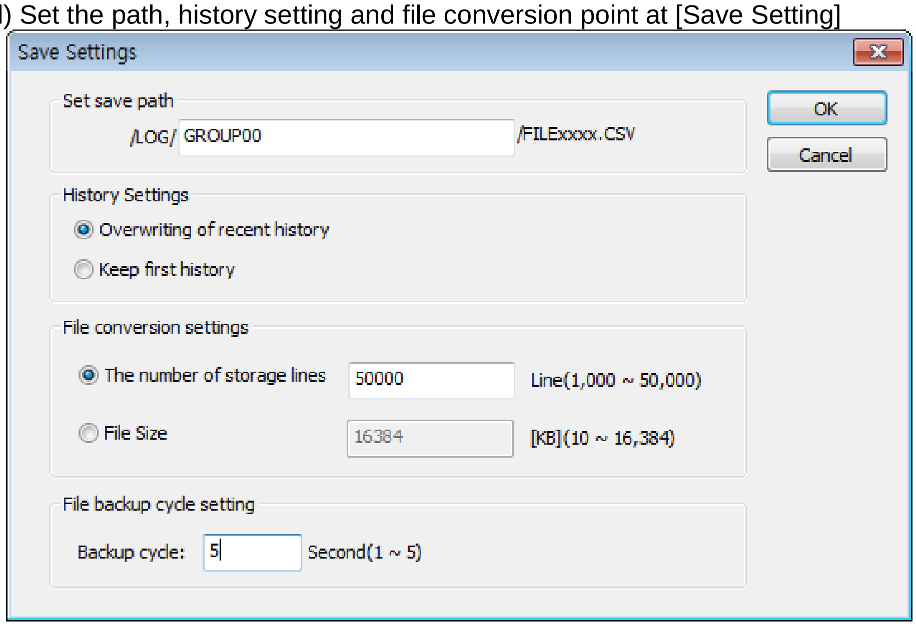
Note

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)

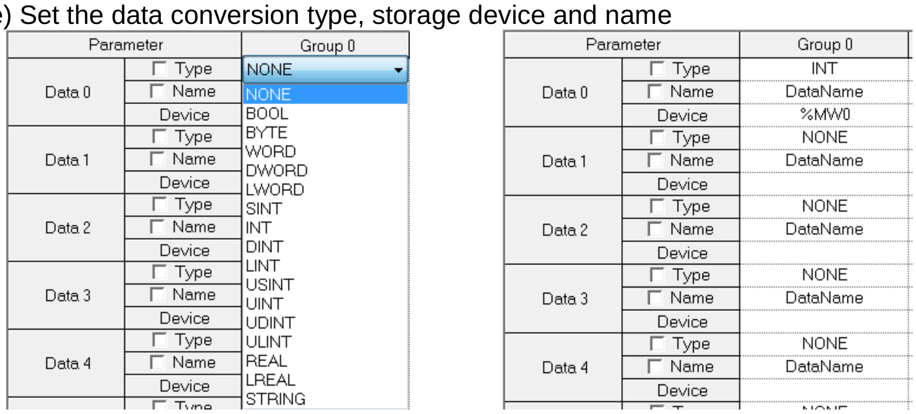
(c) Choose [Save at Every Scan] at the [Data Collection Method]



(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.

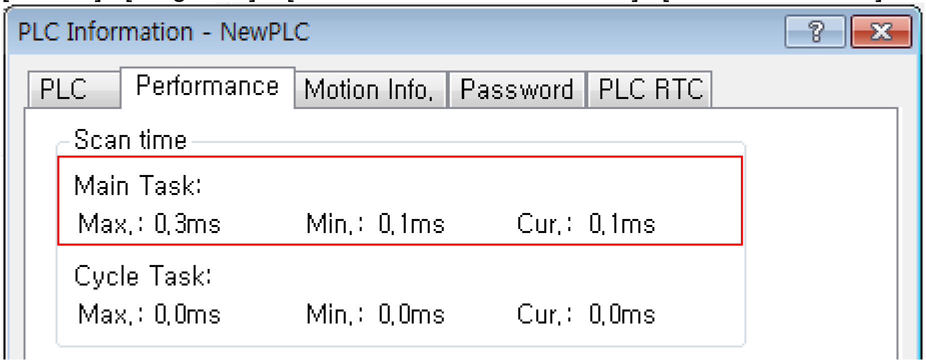
The following are Enable Flags for each datalog group

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.

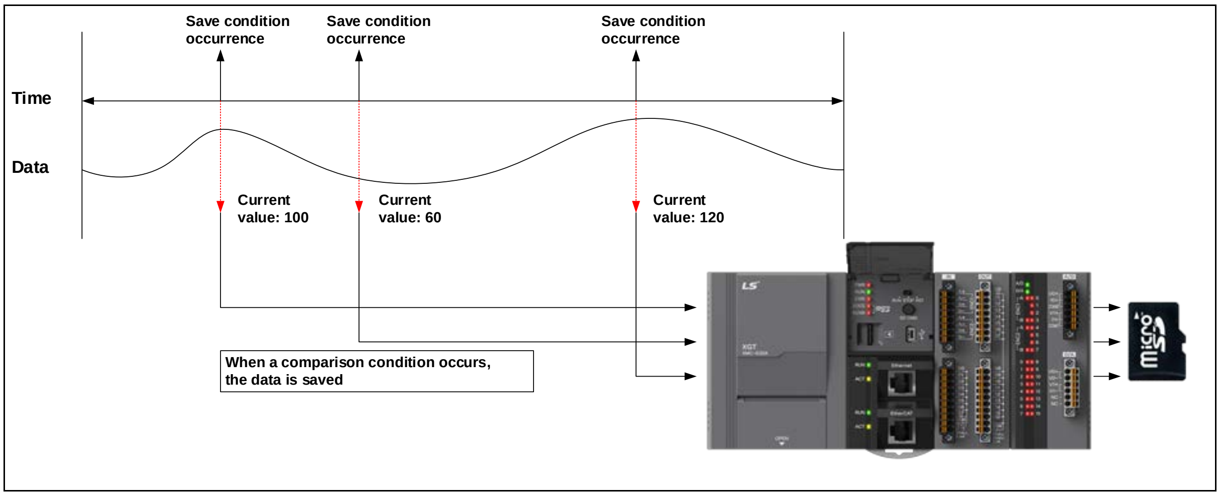
Note

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]



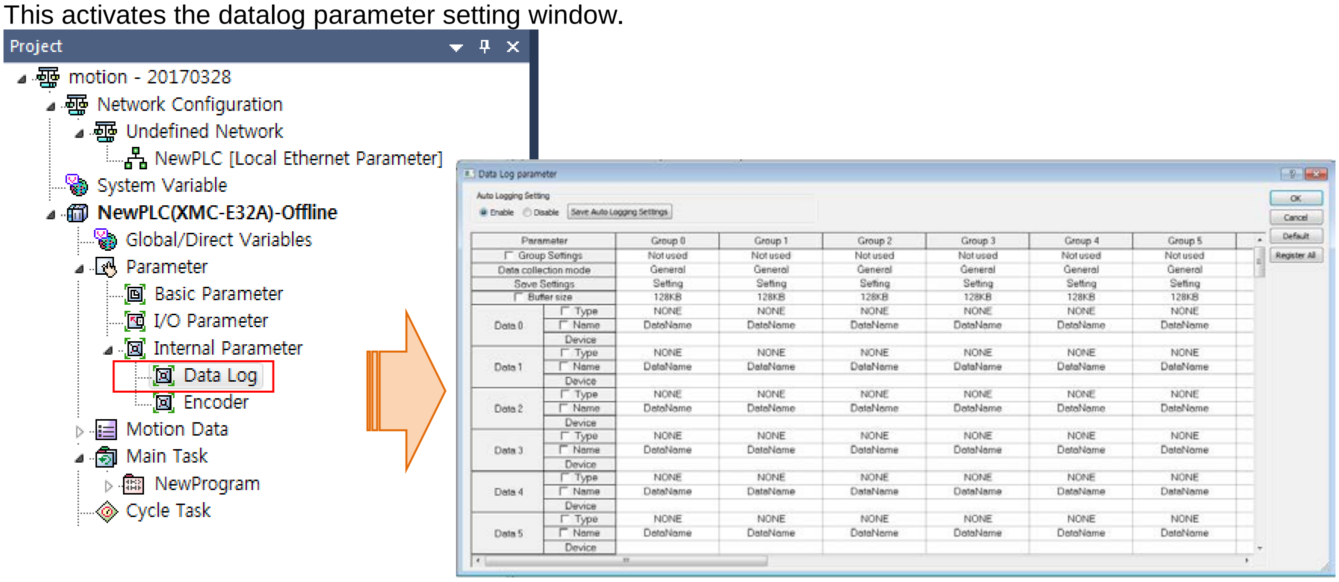
11.4.2 Save at Designated Interval

(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.

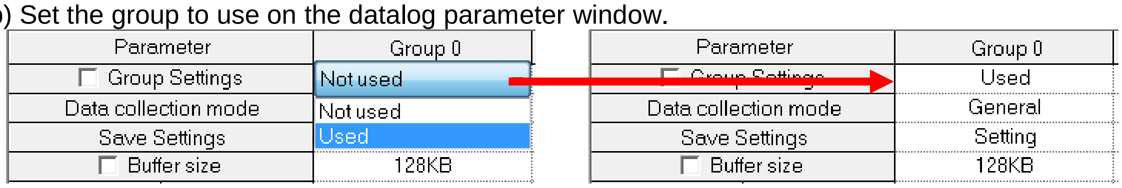


(2) Setting Method

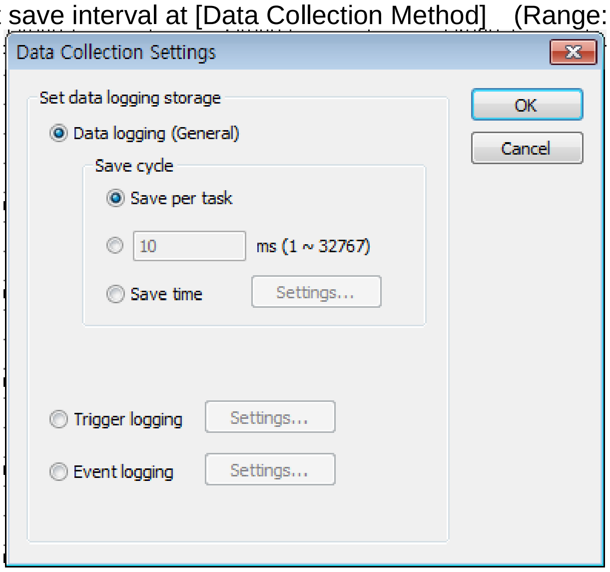
(a) Choose XG5000 – [Project Window] - [internal parameter] - [datalog]
This activates the datalog parameter setting window.



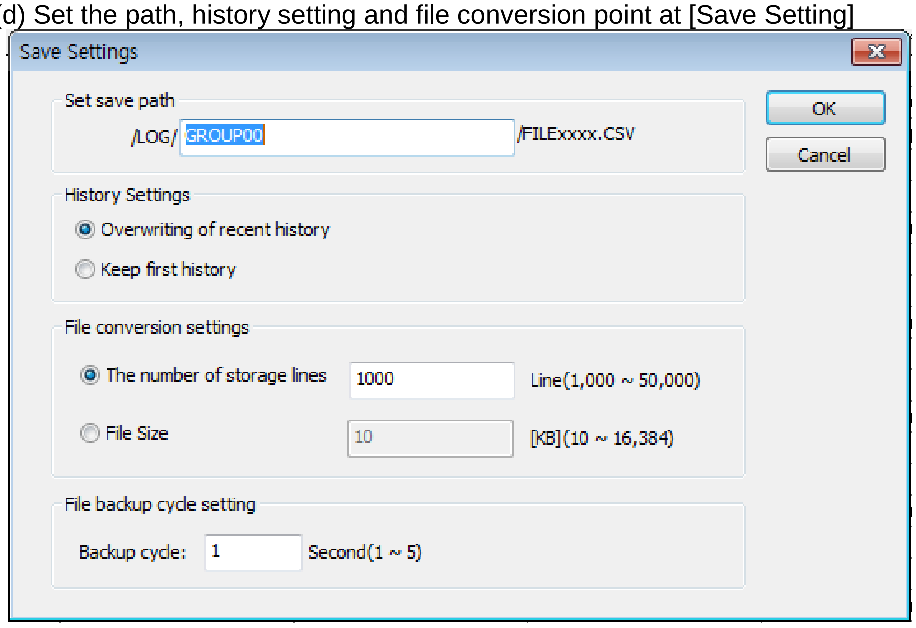
(b) Set the group to use on the datalog parameter window.



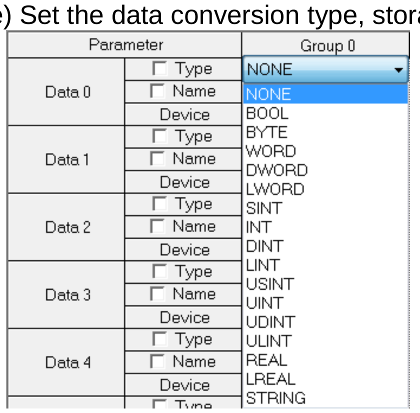
(c) Set save interval at [Data Collection Method] (Range: 1~32,767ms)

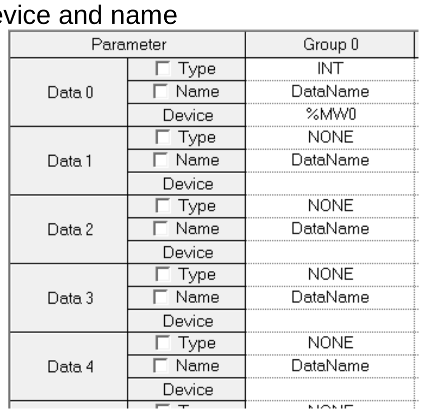


(d) Set the path, history setting and file conversion point at [Save Setting]



(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

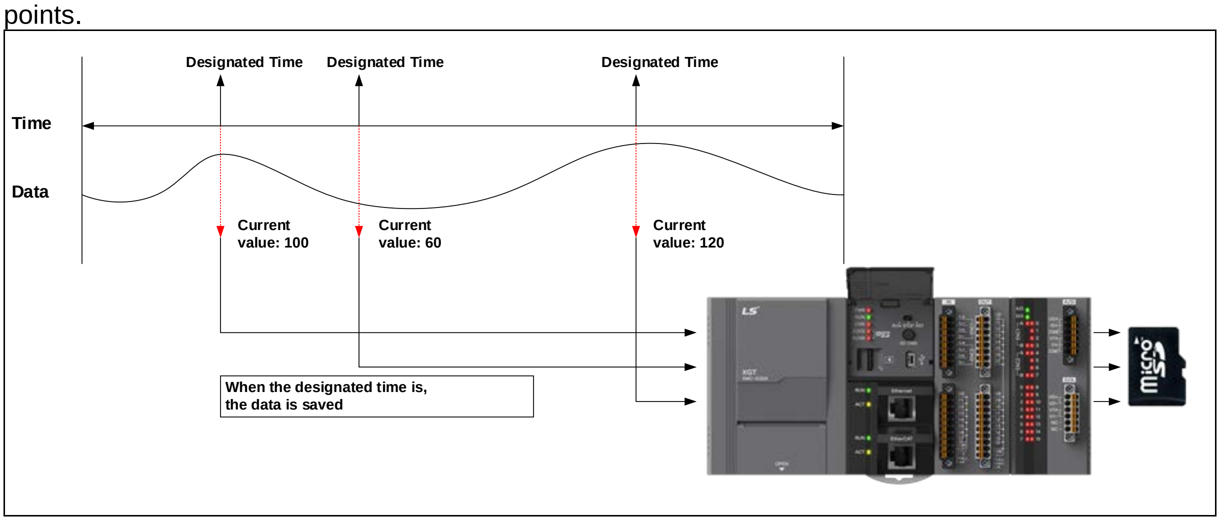
Item	Memory allocation	Type	Description
-	%KW514	WORD	Datalog Enable Flags
_DL00_Enable	%KX8224	BOOL	Group 00 Enable Flag 1: Operation, 0: Stop
_DL01_Enable	%KX8225	BOOL	Group 01 Enable Flag 1: Operation, 0: Stop
_DL02_Enable	%KX8226	BOOL	Group 02 Enable Flag 1: Operation, 0: Stop
_DL03_Enable	%KX8227	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

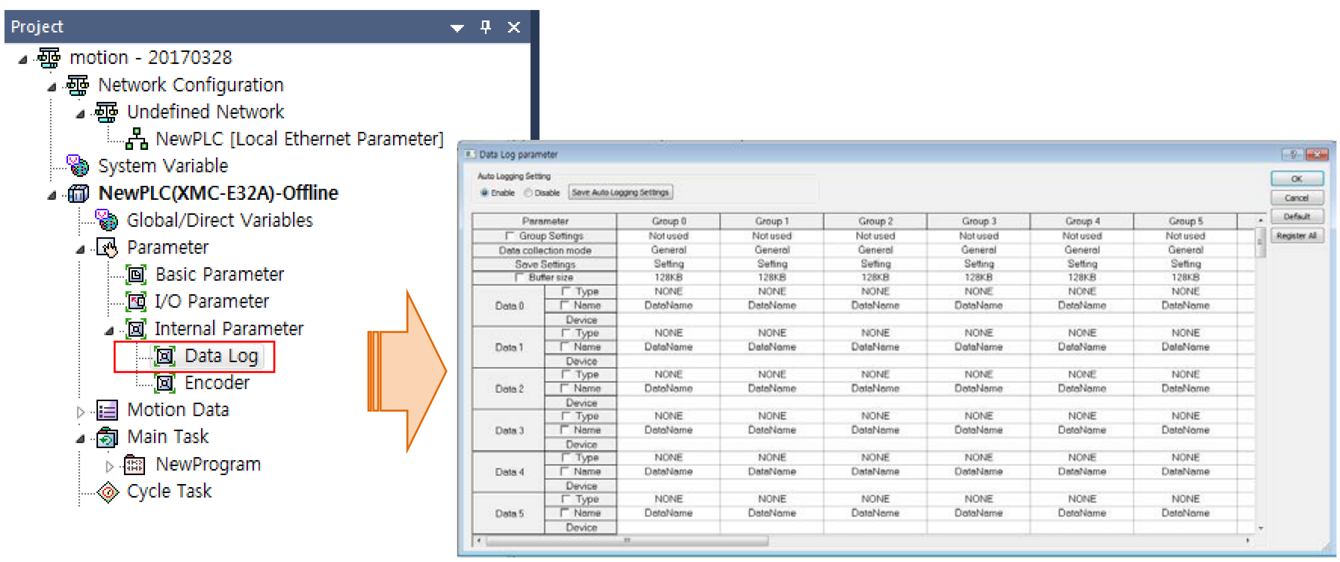
Save at Designated Interval refers to saving data at Designated Time set by the user. It is different from Designed Interval Save in that the former collects data at certain intervals, and is capable of saving data at more accurate points.



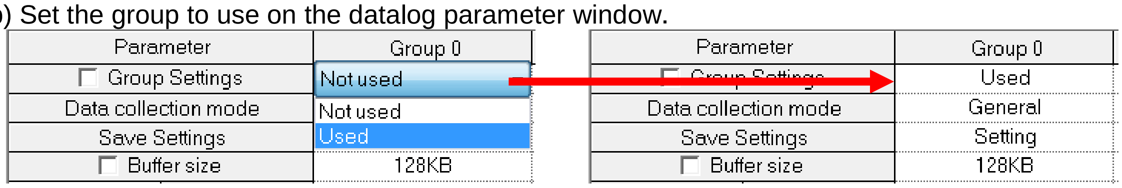
(2) Setting Method

(a) Choose XG5000 – [Project Window] - [internal parameter] - [datalog]

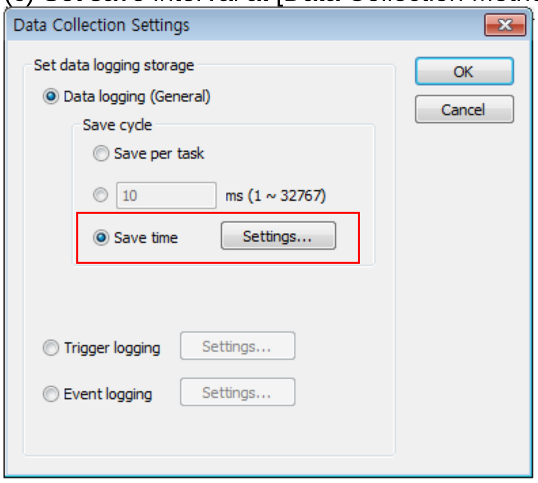
This activates the datalog parameter setting window.



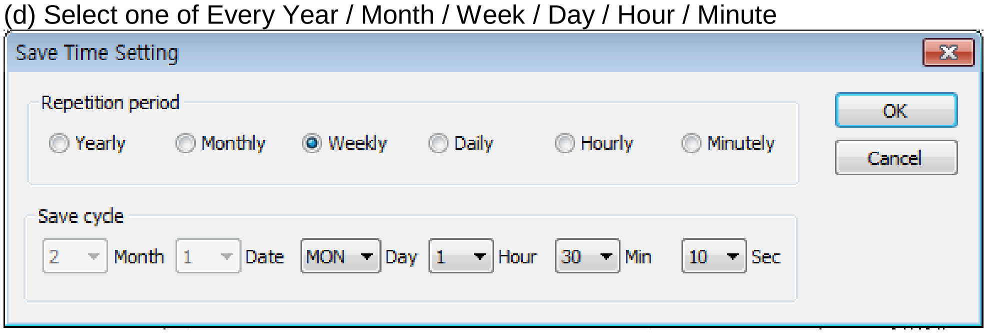
(b) Set the group to use on the datalog parameter window.



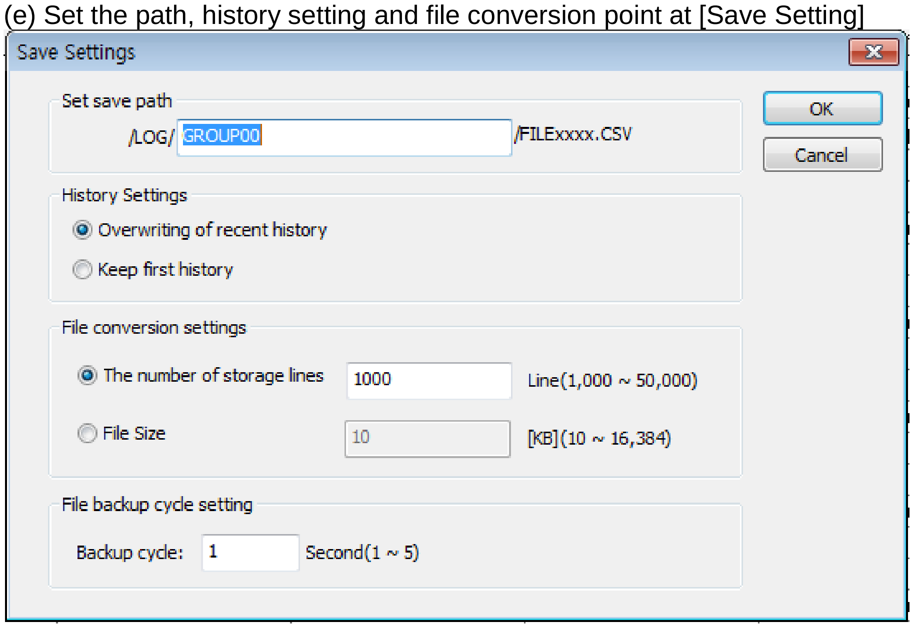
(c) Set save interval at [Data Collection Method] (Range: 1~32,767ms)



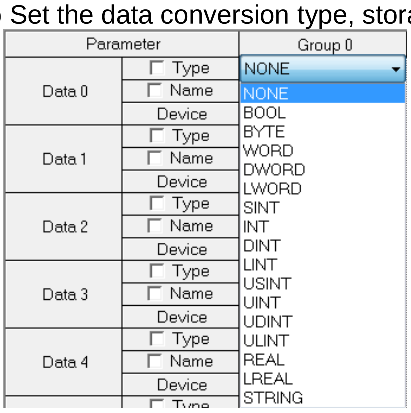
(d) Select one of Every Year / Month / Week / Day / Hour / Minute

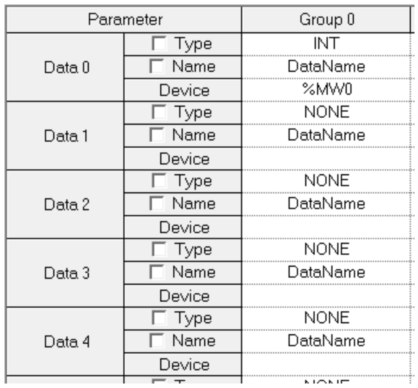


(e) Set the path, history setting and file conversion point at [Save Setting]



(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

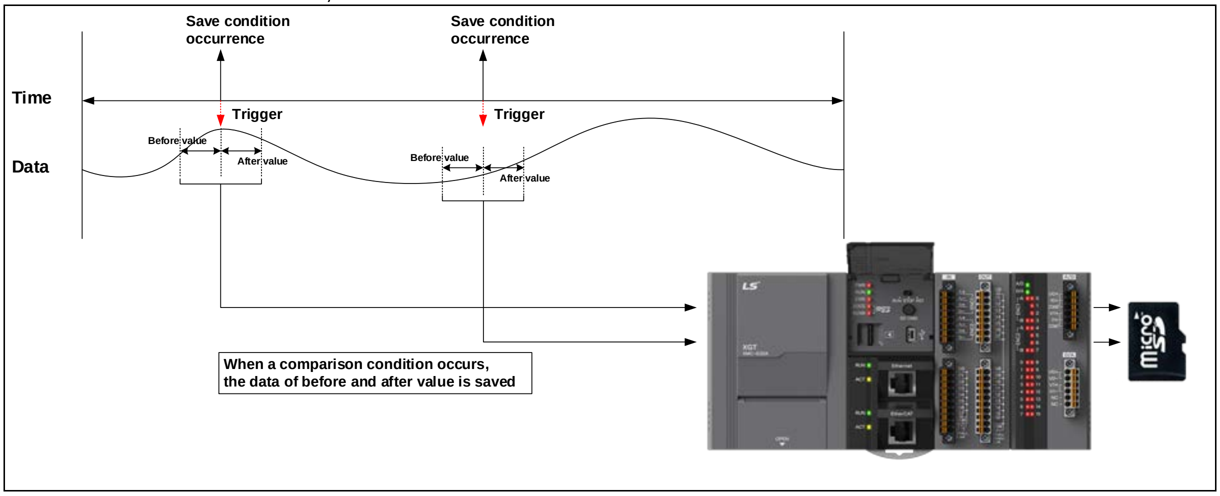
Item	Memory allocation	Type	Description
-	%KW514	WORD	Datalog Enable Flags
_DL00_Enable	%KX8224	BOOL	Group 00 Enable Flag 1: Operation, 0: Stop
_DL01_Enable	%KX8225	BOOL	Group 01 Enable Flag 1: Operation, 0: Stop
_DL02_Enable	%KX8226	BOOL	Group 02 Enable Flag 1: Operation, 0: Stop
_DL03_Enable	%KX8227	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

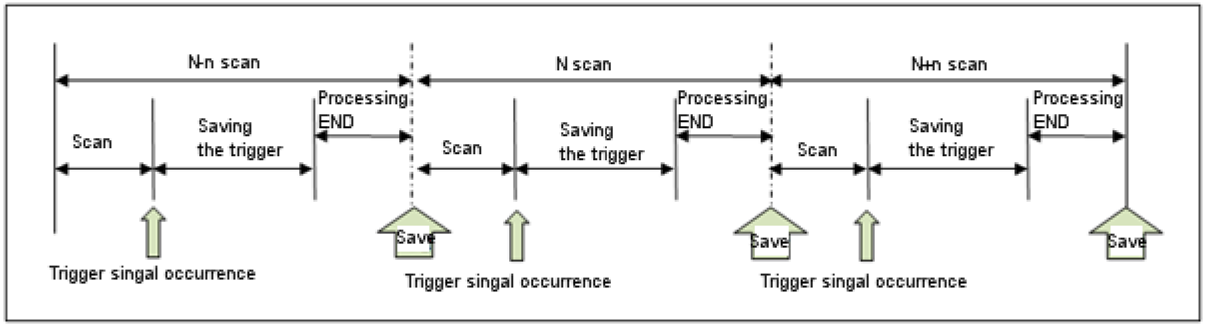
OFF the datalog Enable Flag to stop data saving.

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

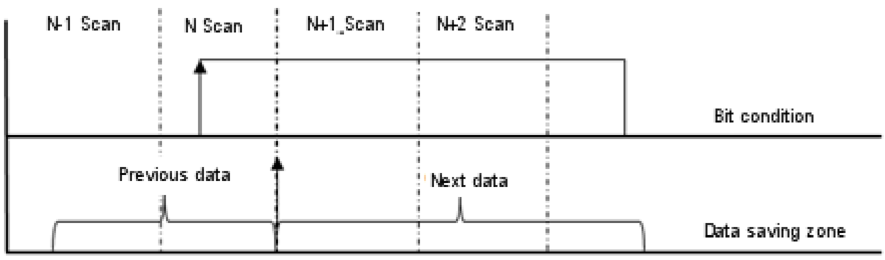
If a trigger occurs after collecting one less than the number of sample blocks prior to the trigger, only the number of blocks collected until then is stored, and the sample is started after the trigger. In this case, the total number of blocks collected can be less than the total number of trigger blocks

11.5.1 Trigger Condition

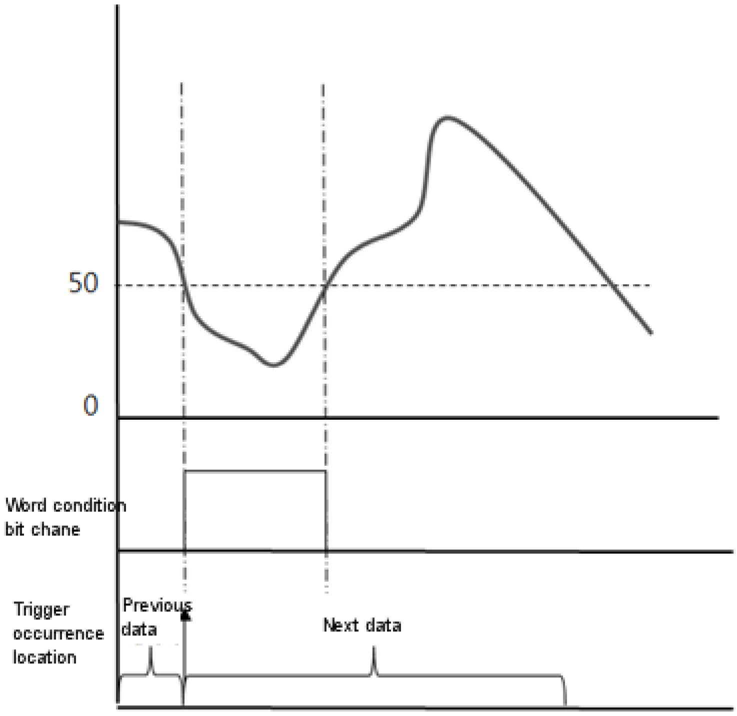
Trigger Save function runs under Single Condition, Multiple Condition. The setting item for single/multiple conditions are as follows. Multiple Condition runs by connecting Single Condition using AND, OR. Up to 4 Single Conditions can be set to form a condition. When the Trigger Condition occurs and 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].

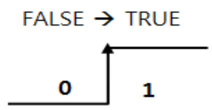
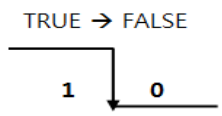
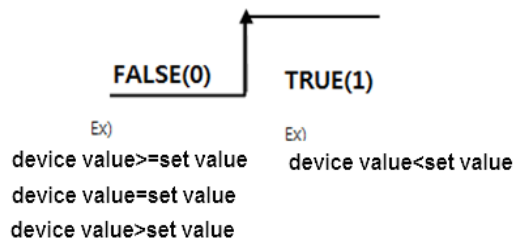
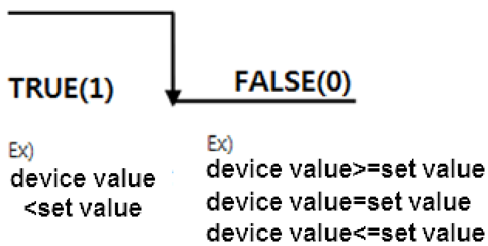
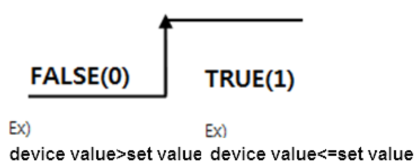
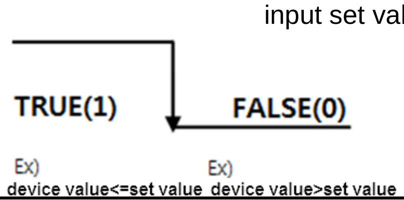
(a) BIT Condition
BIT condition checks the set device BIT value, and collects data by detecting trigger when the value is either [elevation] or [descent].

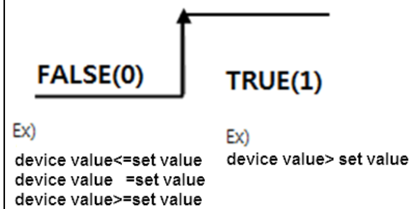
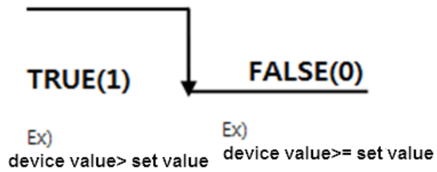
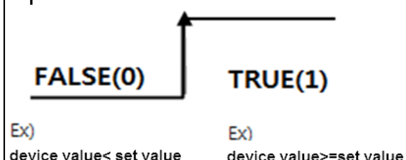
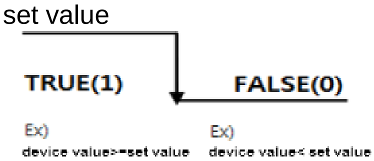
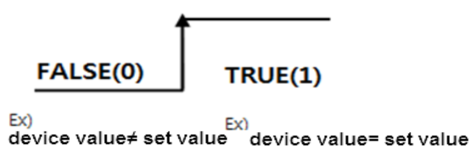
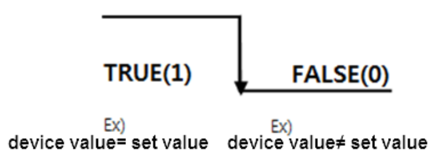


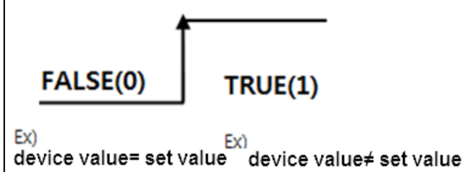
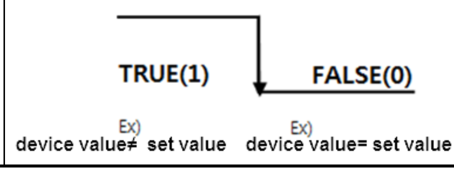
(b) WORD Condition
Word Condition compares the set device with the input value, and converts them into TRUE or FALSE. If the set device value satisfies the input condition, data are collected when the value is either [elevation] or [descent].
Ex) If set value is <50, elevation condition



(c) Condition Description

<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="4">Word Condition</td><td>Elevation</td><td rowspan="2">Small</td><td>Saves data at the elevation edge of the relevant bit, when the set word device value is smaller than the input set value
</td><td></td></tr>
<tr><td>Descent</td><td>Saves data at the descent edge of the relevant bit, when the set word device value is smaller than the input set value
</td><td></td></tr>
<tr><td>Elevation</td><td rowspan="2">Small or Same</td><td>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
</td><td></td></tr>
<tr><td>Descent</td><td>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
</td><td></td></tr>
</table>

	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 	

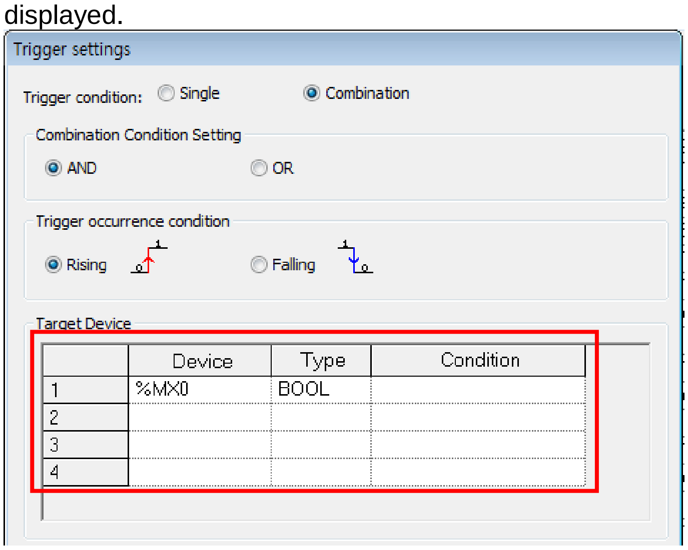
	Trigger Occurrence Condition	Device Set Condition	Operation	Note
Word Condition	Elevation	Different	Saves data at the elevation edge of the relevant bit, when the set word device value is different from the input set value 	
	Descent		Saves data at the descent edge of the relevant bit, when the set word device value is different from the input set value 	

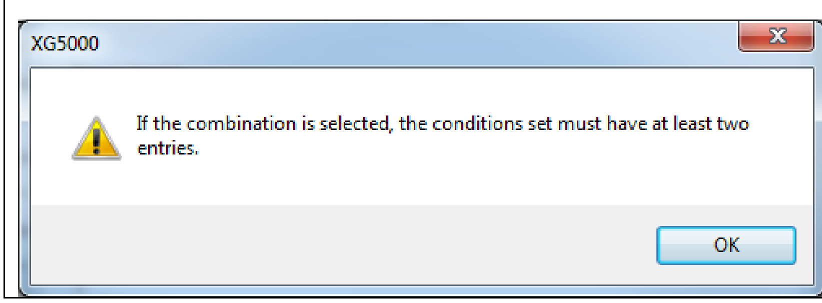
(2) Multiple Condition

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.





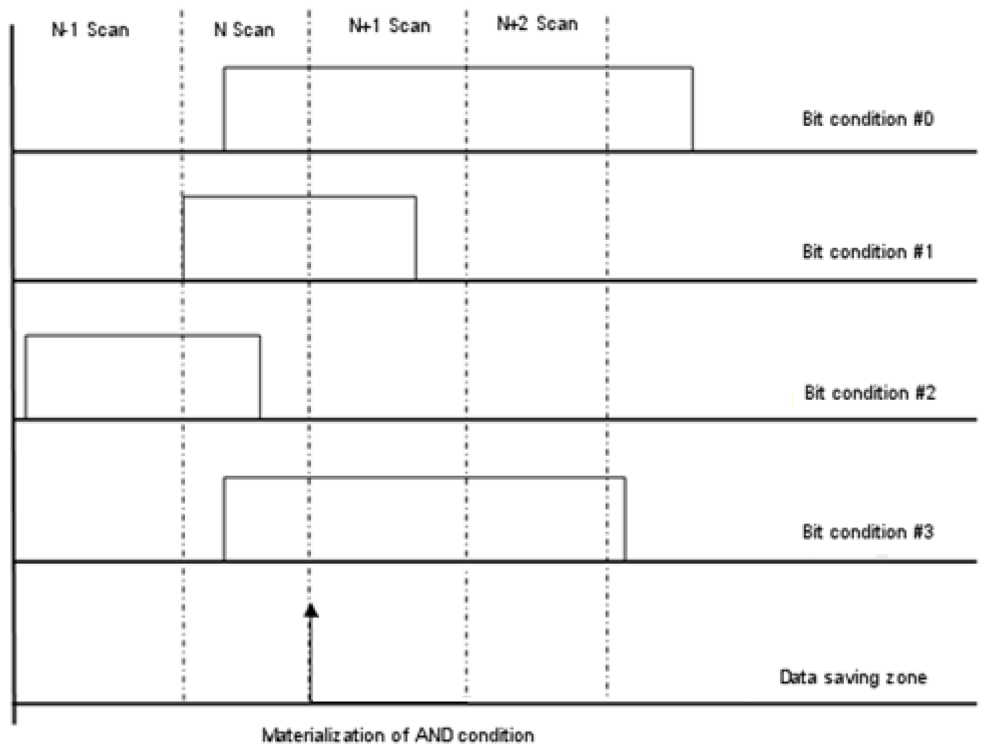
(a) AND Calculation

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.

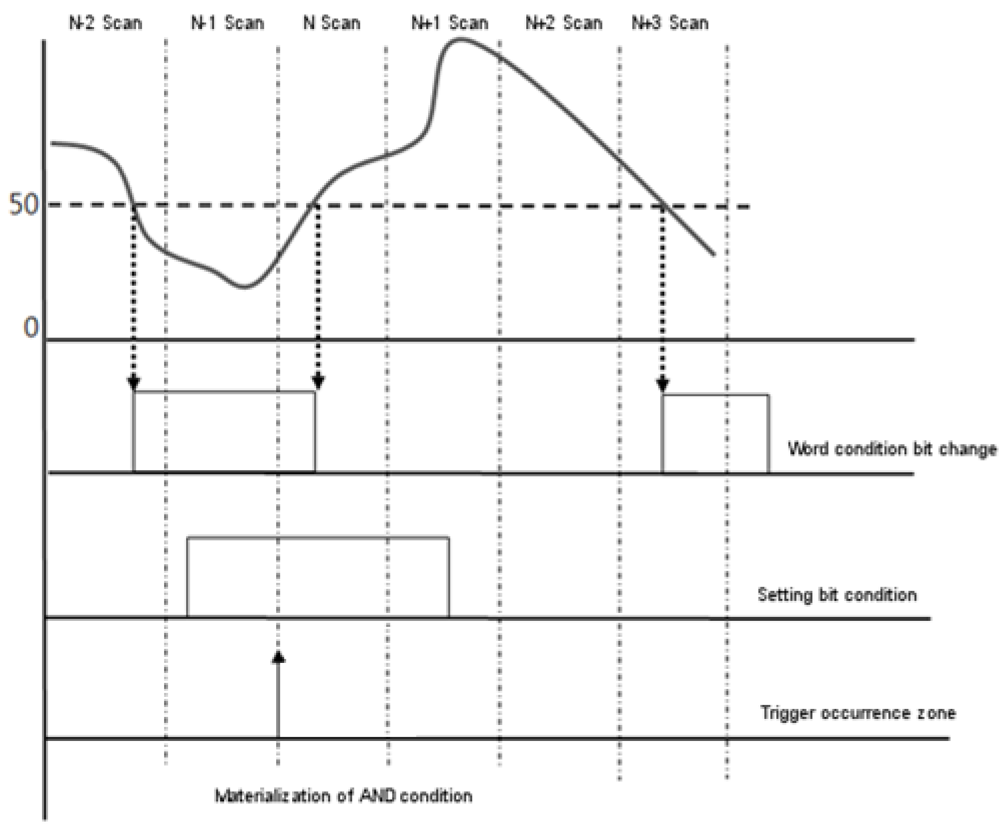
☞ When setting only with BIT condition

	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	

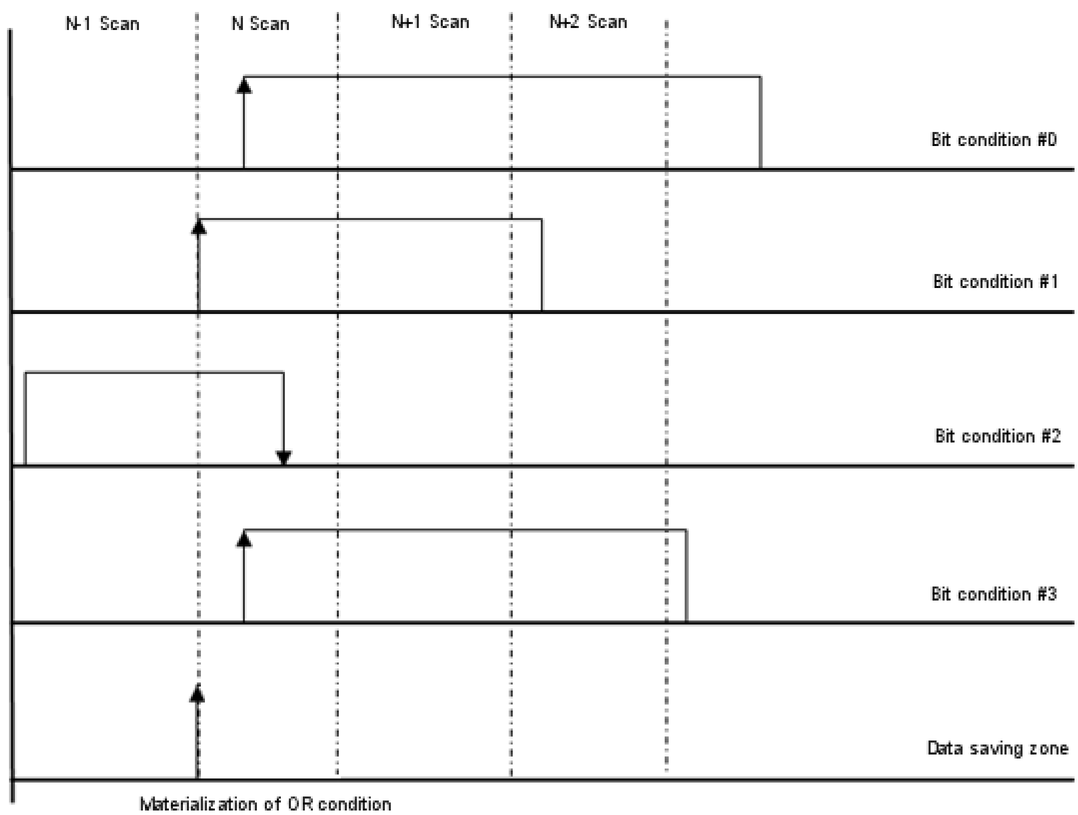


(b) OR Calculation

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.

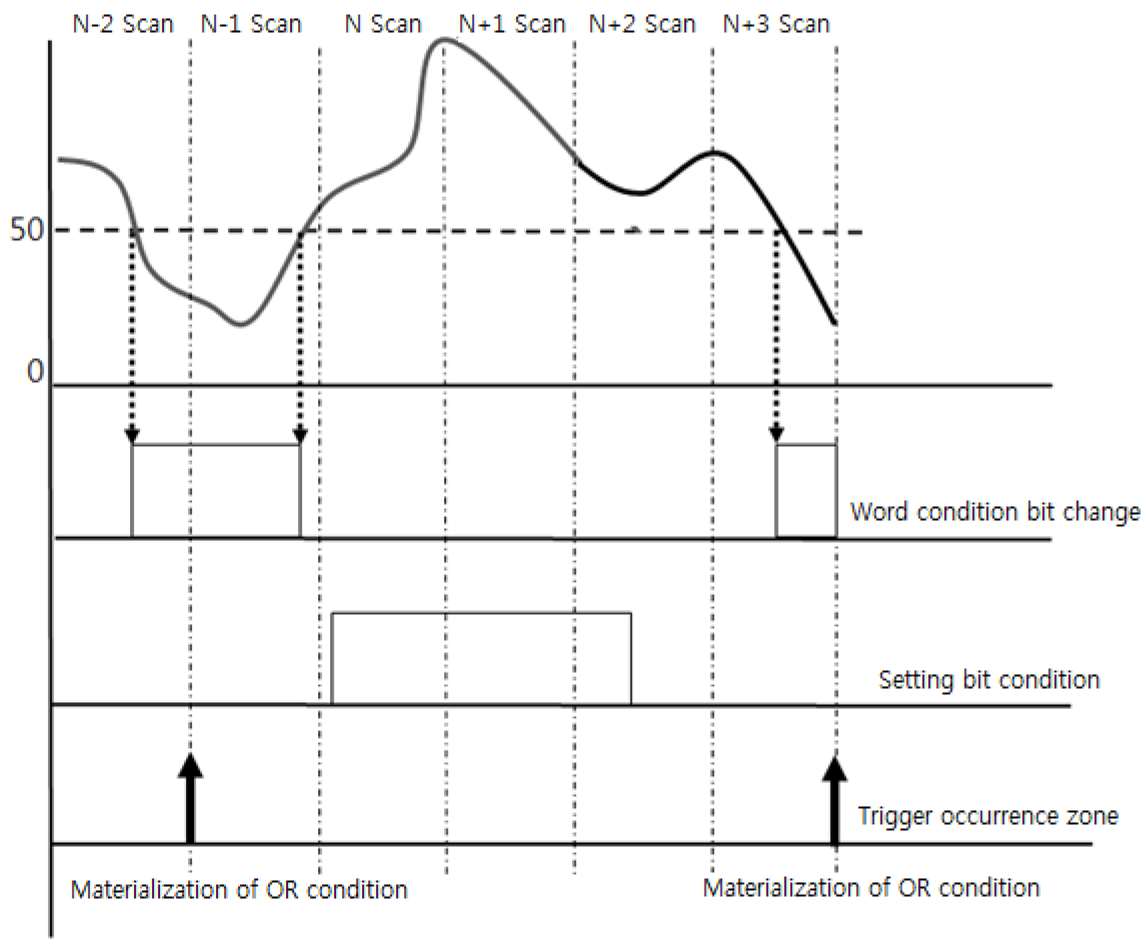
☞ When setting only with BOOL condition

	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

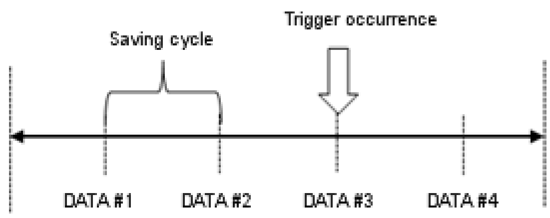
The item that can be set at the parameter is the total number of trigger sample blocks and the number of sample blocks before trigger condition. The number of sample blocks after trigger is determined by the two input values

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**

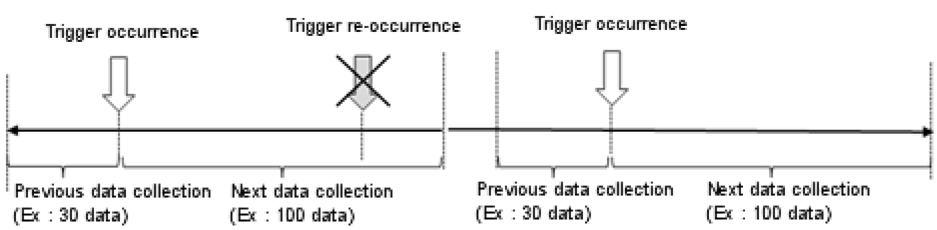


Caution

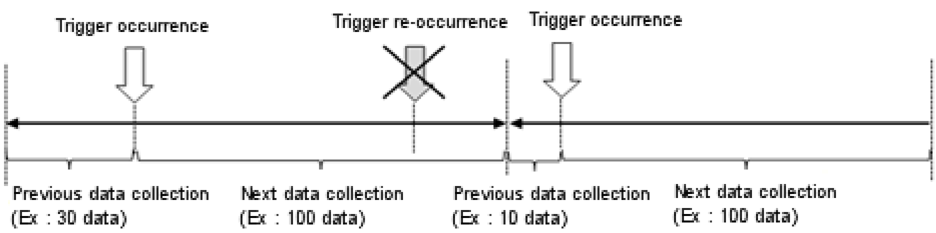
After selecting Trigger Save, if the Trigger Condition is again satisfied before data saving is complete, the new trigger is ignored and the trigger collisions number flag value increases. Trigger Condition is checked after saving the set number of trigger sample blocks, and then the data are saved.

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



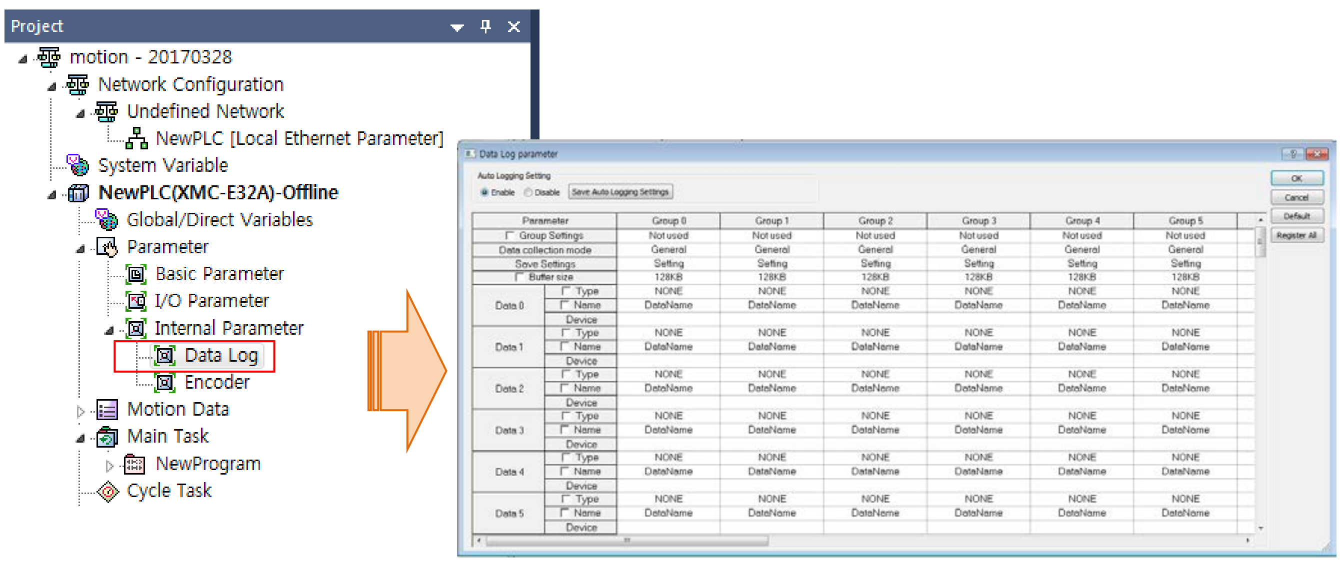
(2) If Trigger occurs before the number of previous data set by the parameter
→ Saves data in the number of transfer data collected, and then collects subsequent data
(Saves less number of data than the number set by the parameter)



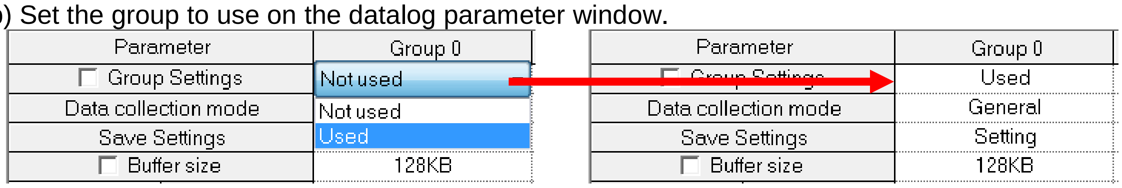
11.5.6 Setting Method

(1) Single BIT Condition

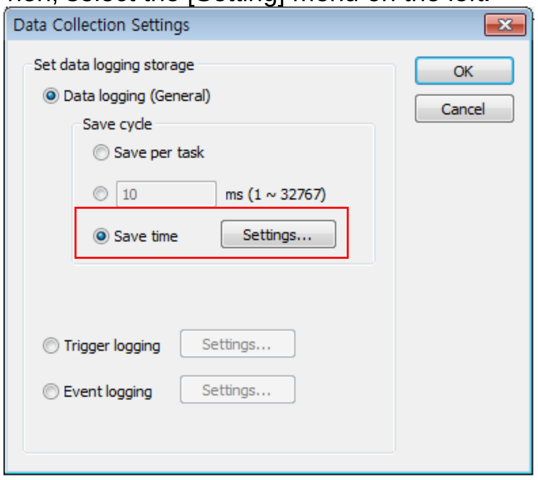
(a) Choose XG5000 – [Project Window] - [internal parameter] - [datalog]
This activates the datalog parameter setting window.



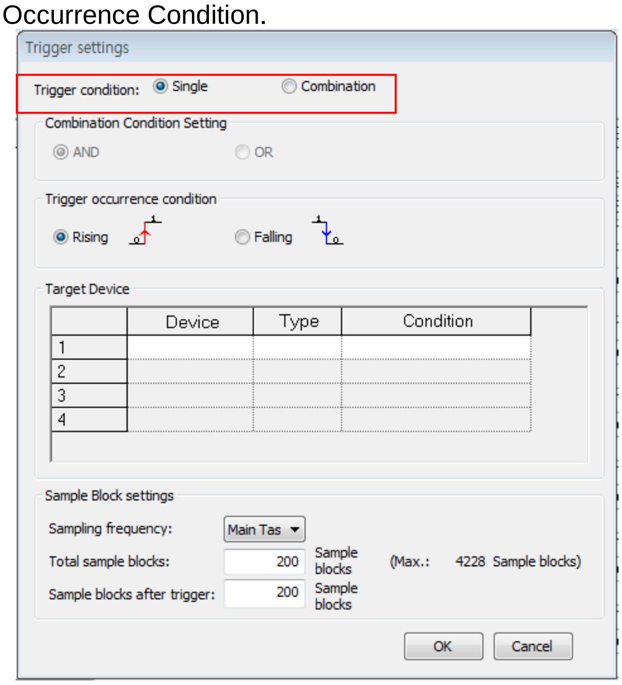
(b) Set the group to use on the datalog parameter window.



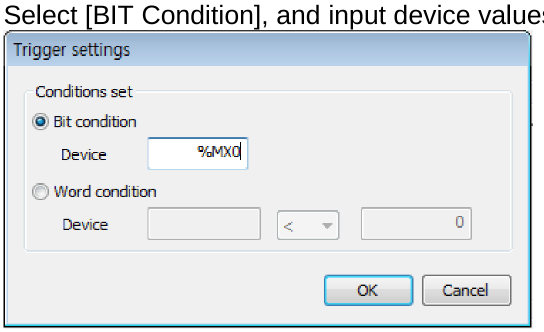
(c) Select [Trigger Logging] at [Data Collection Method] to activate [Setting] menu on the left.
Then, select the [Setting] menu on the left.



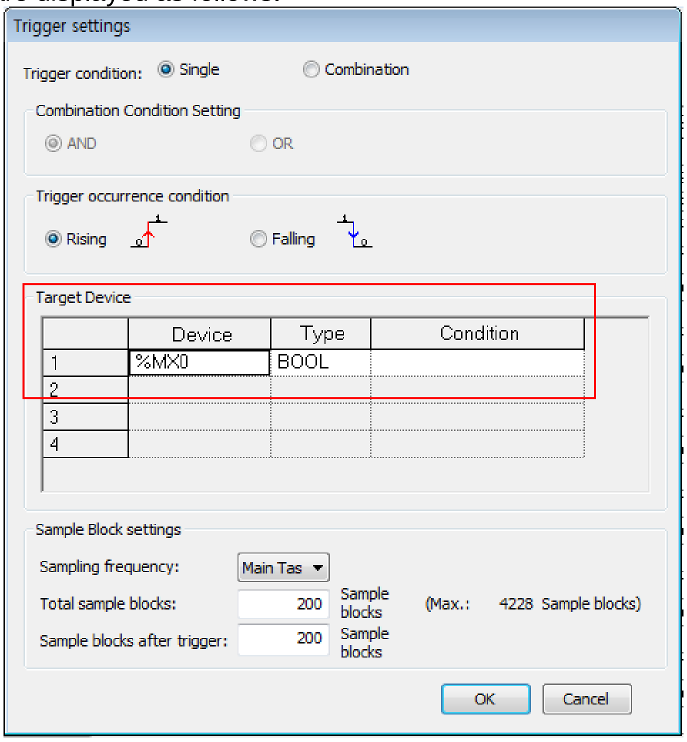
(d) Upon selection, the following window is activated for trigger setting.
Select [Single Condition] as the Trigger Condition. Select either [Elevation] or [Descent] as the Trigger Occurrence Condition.



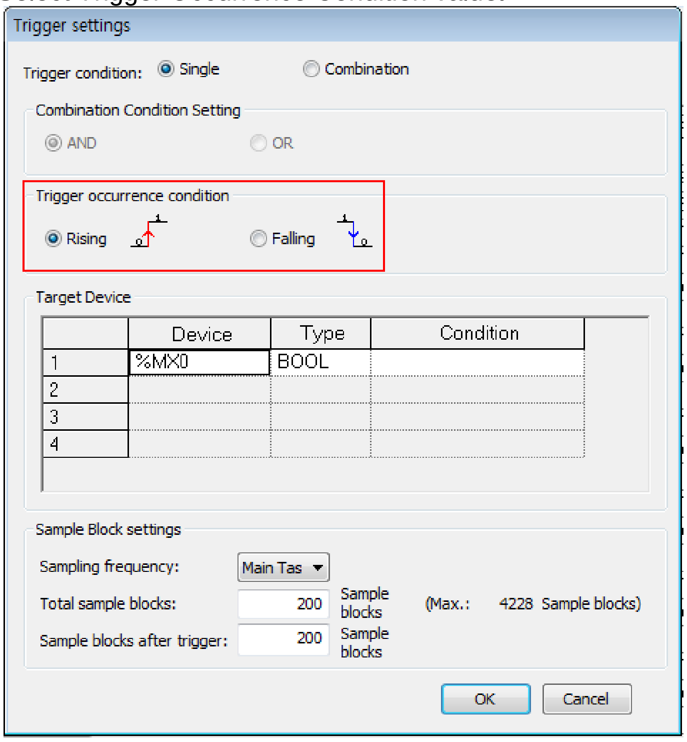
(e) Select the condition setting menu to activate the following setting window.
Select [BIT Condition], and input device values into the device window in BIT types.



When setting is complete, the window closes and the conditions initially set at the Trigger Setting Condition menu are displayed as follows.

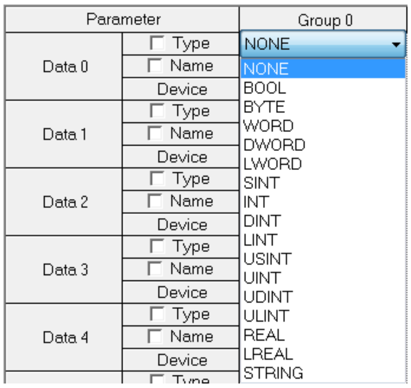


(f) Select Trigger Occurrence Condition value.



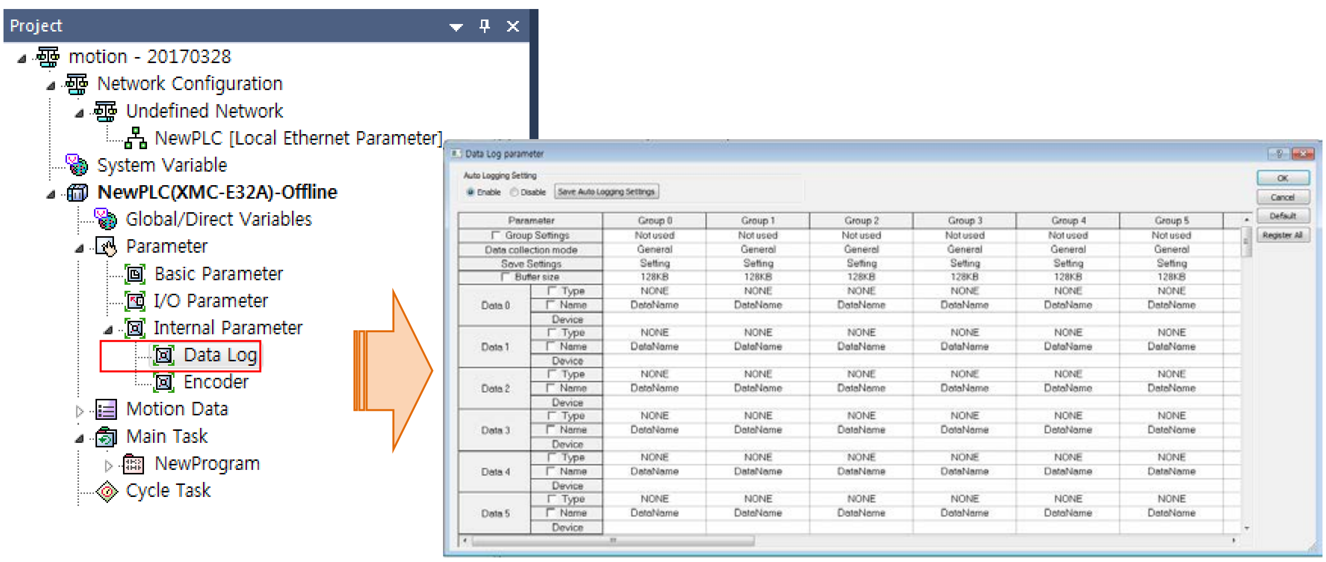
(g) Input sampling interval, total number of samples and number of samples after trigger, then finish Trigger setting. See [11.5.2 Trigger Sample Block Calculation] for operation of number of sample blocks.

(h) Device values set at the Datalog Basic Setting window are collected, and saved into the SD memory after type conversion.

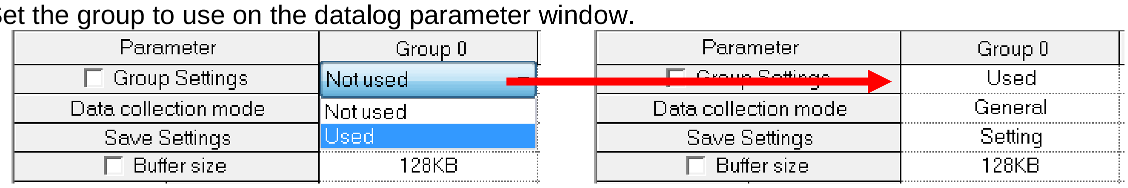


(2) Single WORD Condition

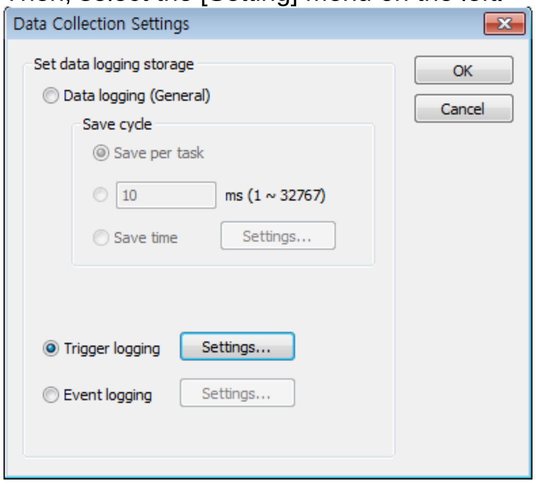
(a) Choose XG5000 – [Project Window] - [internal parameter] - [datalog]
This activates the datalog parameter setting window.



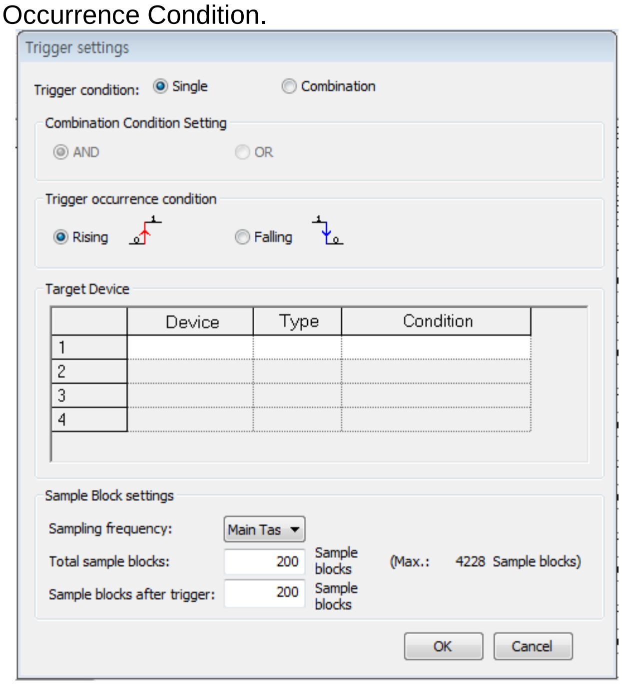
(b) Set the group to use on the datalog parameter window.



(c) Select [Trigger Logging] at [Data Collection Method] to activate [Setting] menu on the left.
Then, select the [Setting] menu on the left.

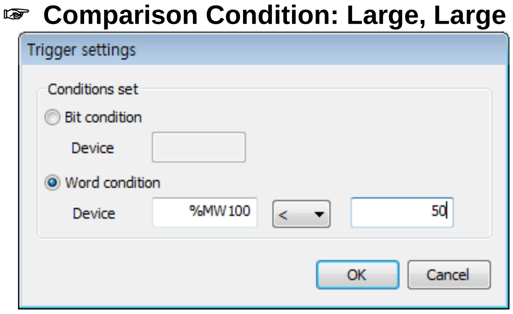


(d) Upon selection, the following window is activated for trigger setting.
Select [Single Condition] as the Trigger Condition. Select either [Elevation] or [Descent] as the Trigger Occurrence Condition.

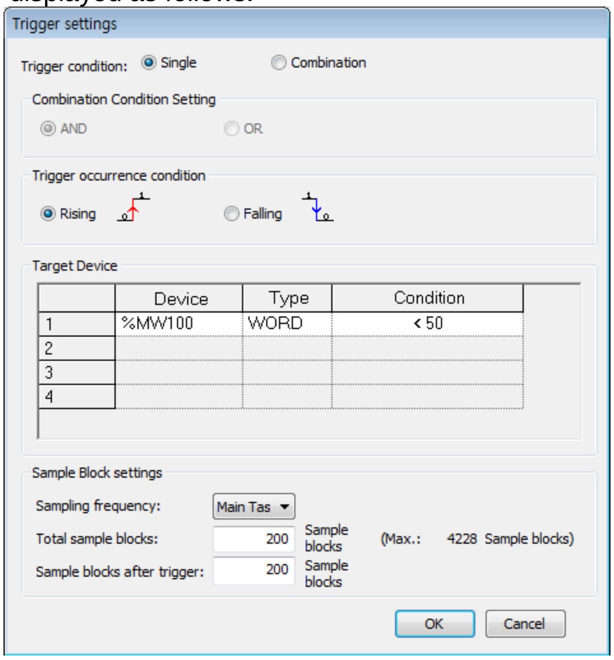


(e) Select the condition setting menu to activate the following setting window.
Select [Word Condition], and input device values into the device window in BIT types, and input comparison condition and comparison values

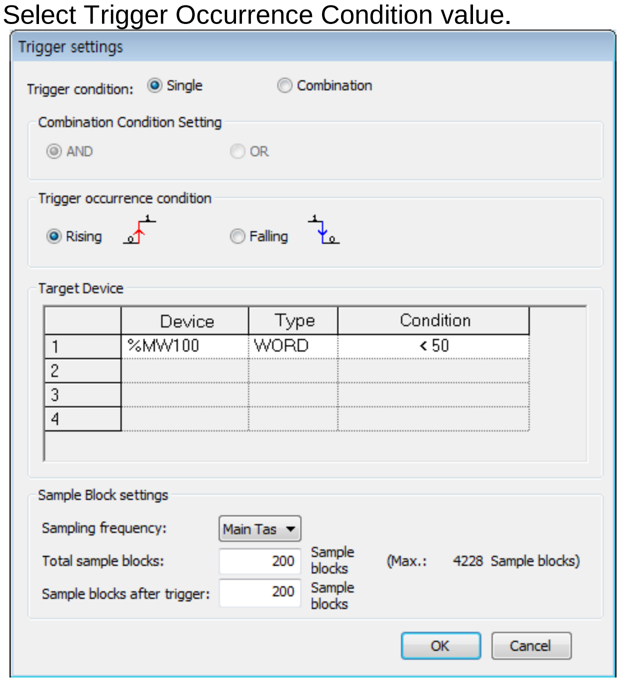
☞ **Comparison Condition: Large, Large or Same, Same, Small, Small or Same, Not Same.**



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.

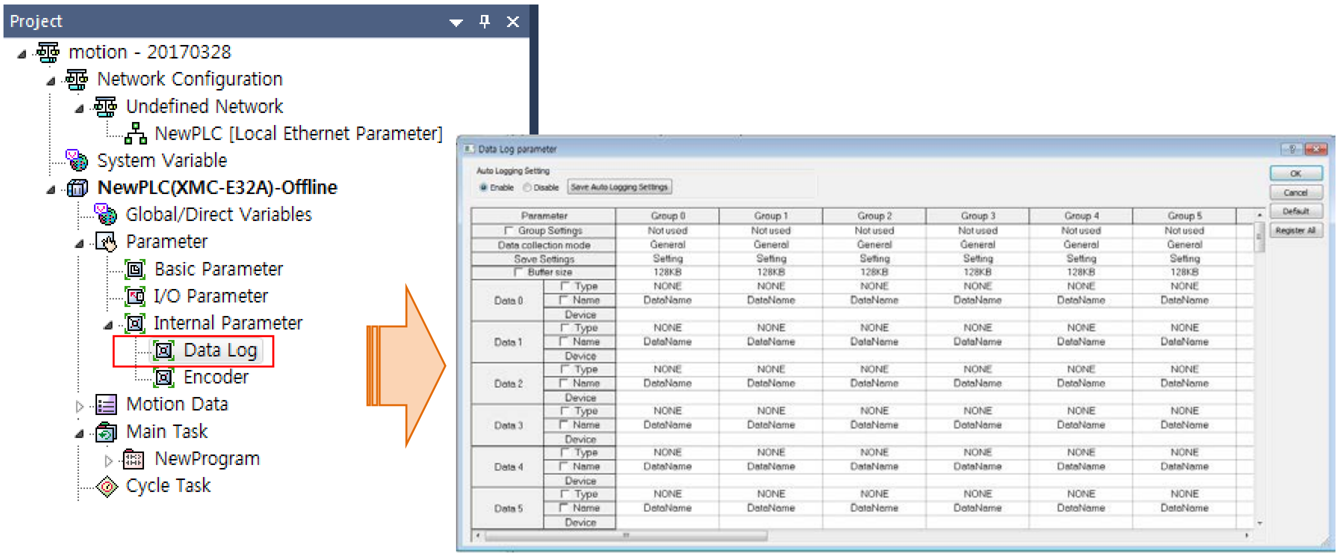
(h) Device values set at the Datalog Basic Setting window are collected, and saved after type conversion.

Caution

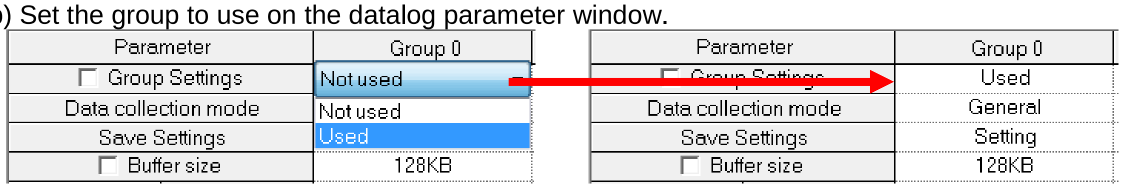
When inputting single, word condition set values, set device type as [BIT] and [WORD], respectively.

(3) Multiple AND Condition

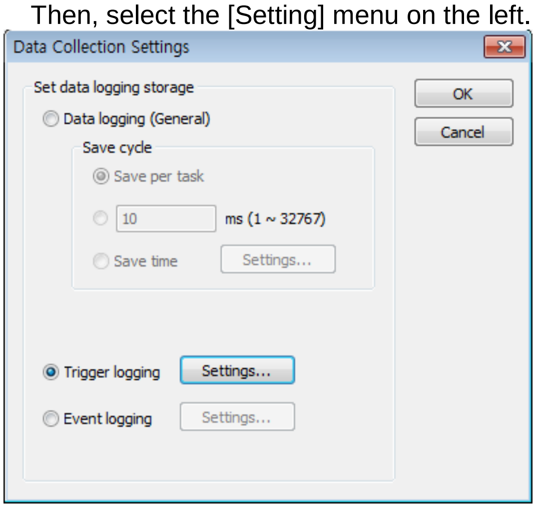
(a) Choose XG5000 – [Project Window] - [internal parameter] - [datalog]
This activates the datalog parameter setting window.



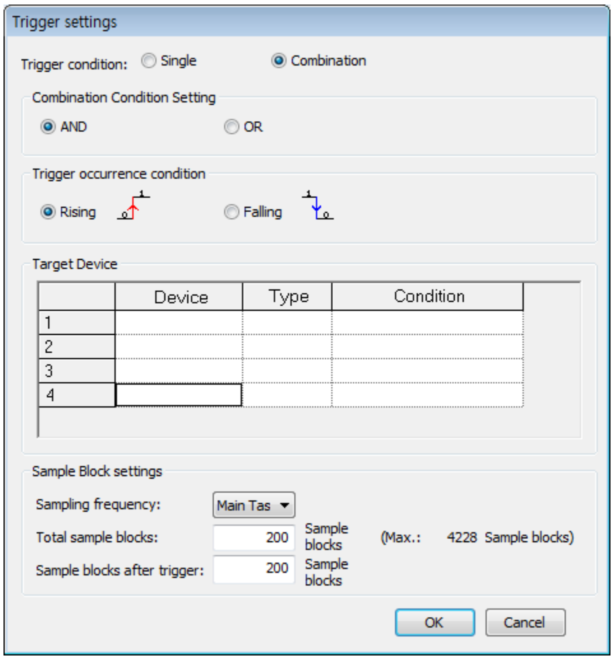
(b) Set the group to use on the datalog parameter window.



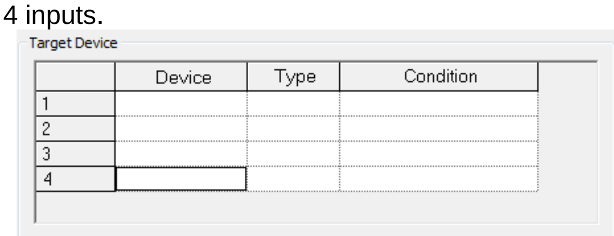
(c) Select [Trigger Logging] at [Data Collection Method] to activate [Setting] menu on the left.
Then, select the [Setting] menu on the left.



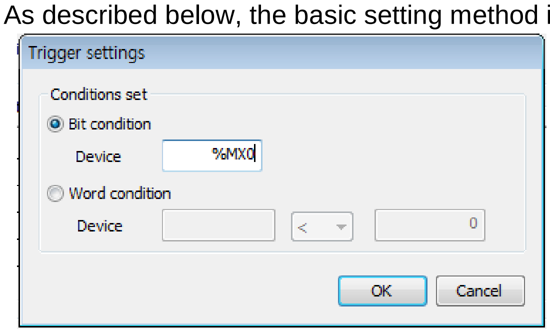
(d) Upon selection, the following window is activated for trigger setting.
Select [Multiple Condition] as Trigger Condition, Select either [Elevation] or [Descent] as the Trigger Occurrence Condition.

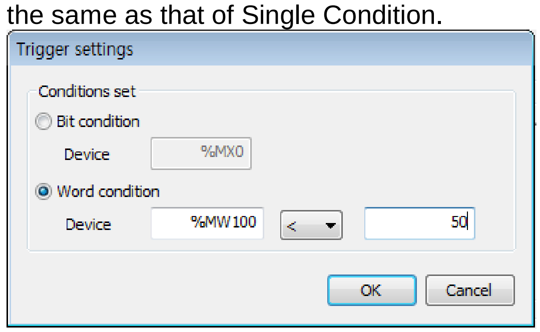


(e) Select [Trigger Condition] and [Multiple Condition] to activate the condition setting window which allows for up to 4 inputs.

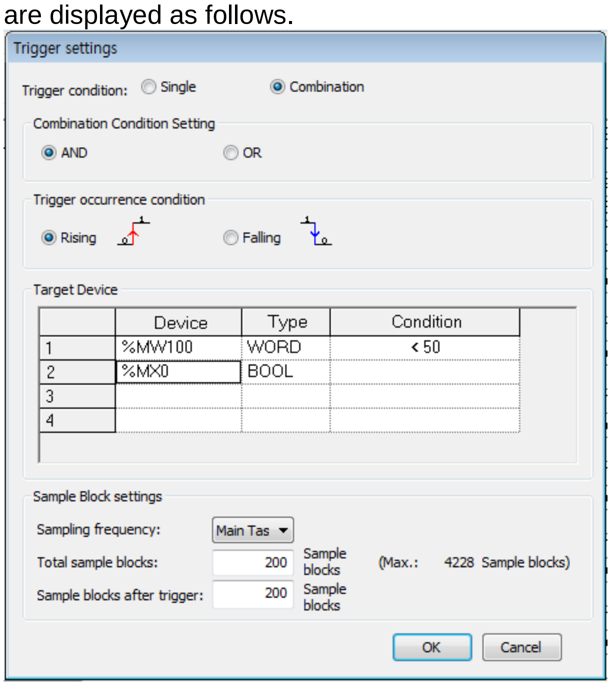


(f) Select each condition setting menu one by one, inputting specific set values.
[Multiple Condition] activates Trigger Condition by combining [Single Conditions] through operation to save data. As described below, the basic setting method is the same as that of Single Condition.

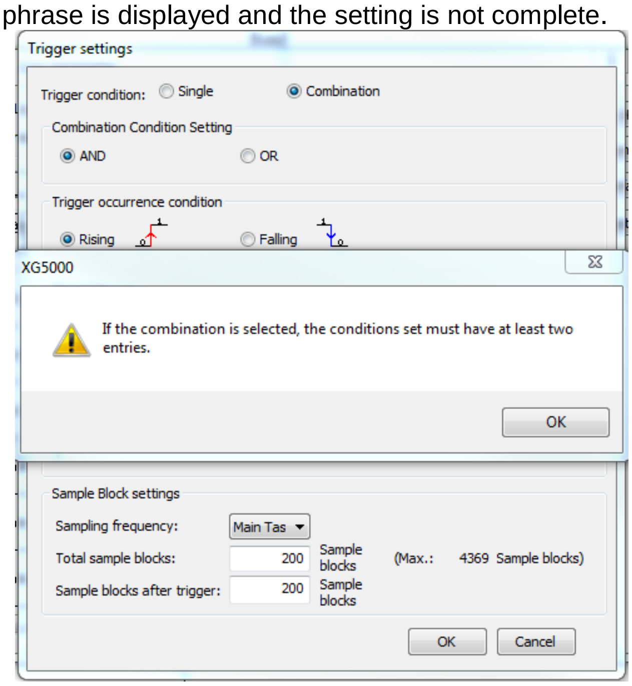




When setting is complete, the window closes and the conditions initially set at the Trigger Setting Condition menu are displayed as follows.

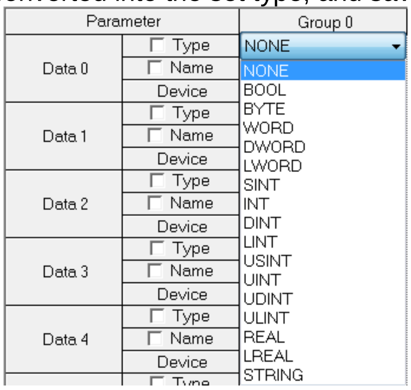


If only one [Condition Setting] is input after selecting Calculation Condition before finishing the setting, the following phrase is displayed and the setting is not complete.



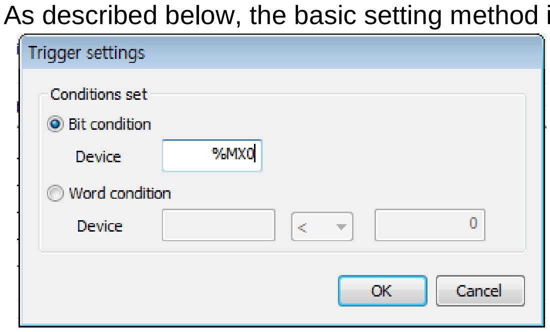
(g) Input sampling interval, total number of samples and number of samples after trigger, then finish Trigger setting.

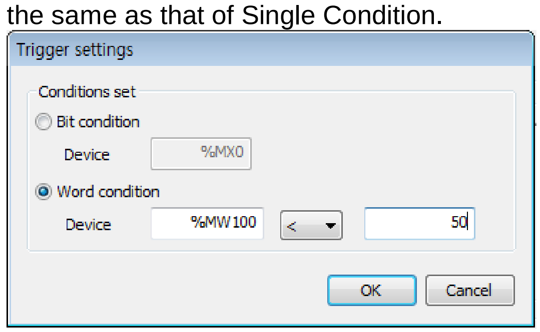
(h) Device values set at the Datalog Basic Setting window are collected when the Trigger Condition occurs, converted into the set type, and saved into the SD memory.ion.



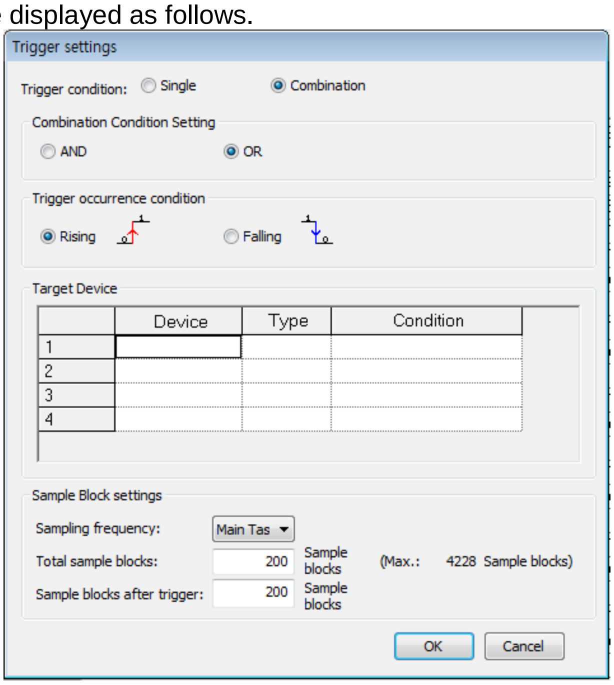
(4) Multiple OR Condition
[Trigger Setting] is identical to the [Multiple AND Calculation] above.

(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.

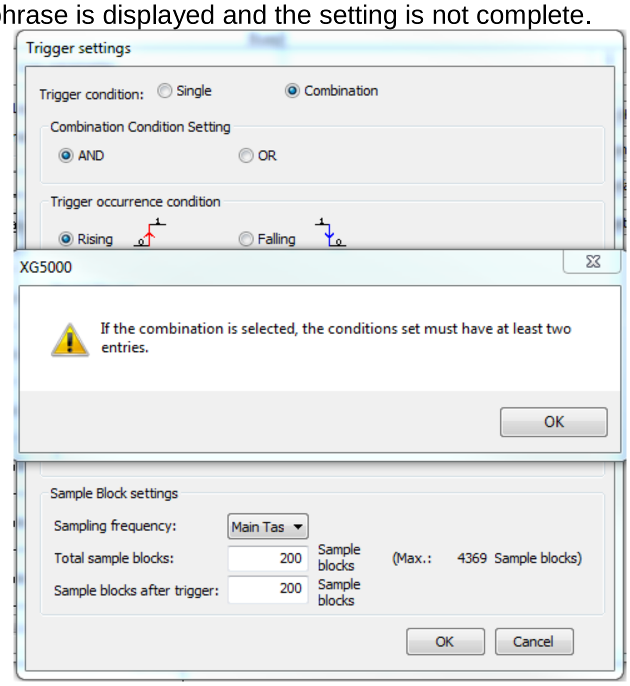




When setting is complete, the window closes and the conditions initially set at the Trigger Setting Condition menu are displayed as follows.

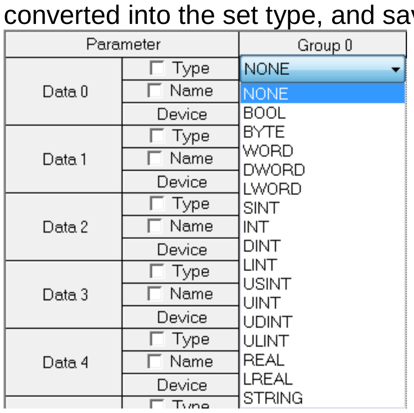


If only one [Condition Setting] is input after selecting Calculation Condition before finishing the setting, the following phrase is displayed and the setting is not complete.



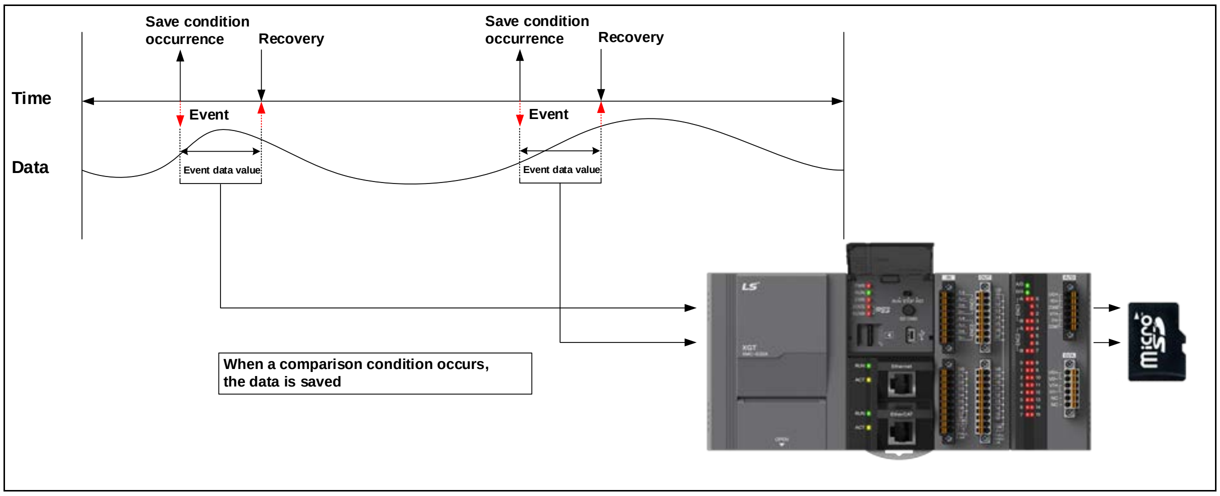
(b) Input sampling interval, total number of samples and number of samples after trigger, then finish Trigger setting.

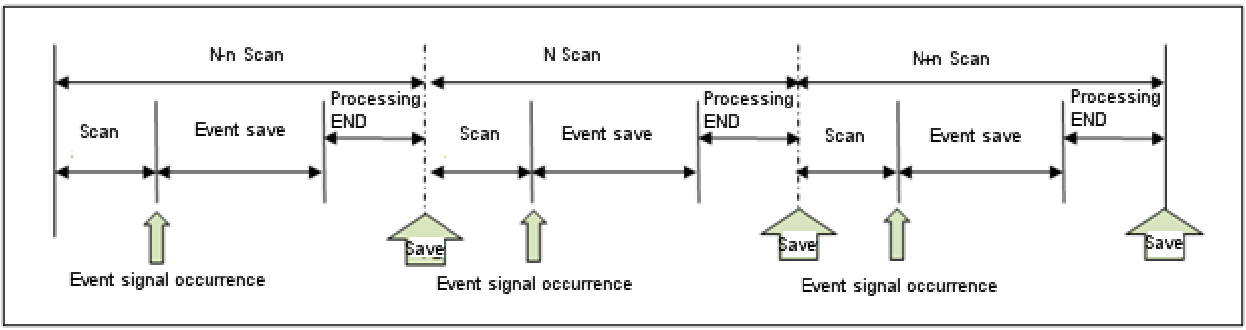
(c) Device values set at the Datalog Basic Setting window are collected when the Trigger Condition occurs, converted into the set type, and saved into the SD memory.



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

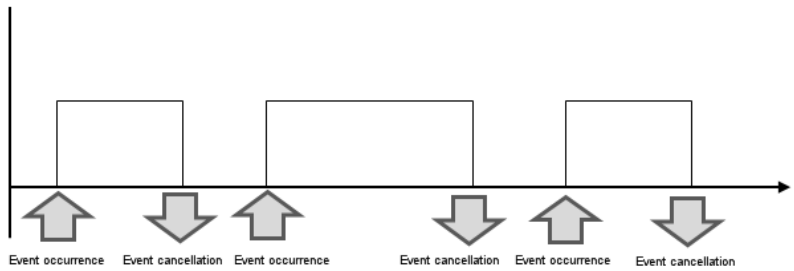
After selecting Trigger Save, if the first trigger condition occurs and another trigger condition occurs while collecting data, the new trigger is ignored.

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.

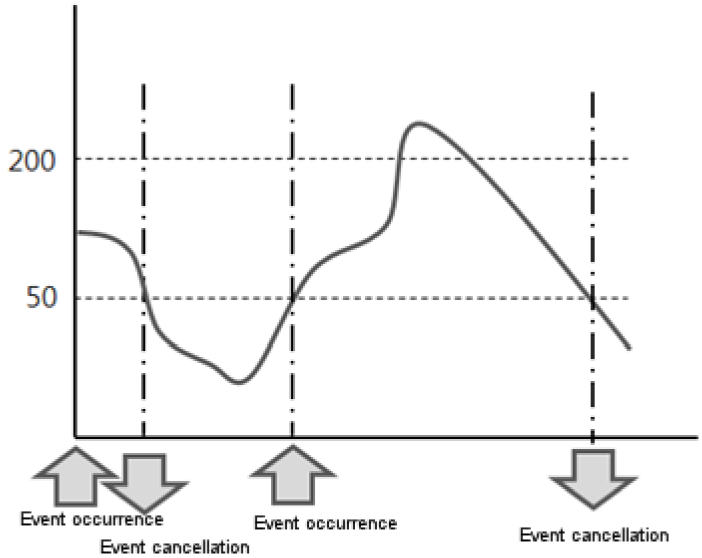
(1) Single Condition
Single Condition runs under BIT Condition, WORD Condition.

(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

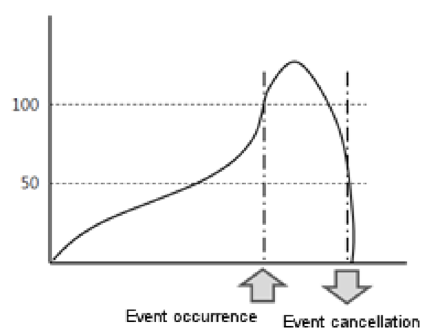
Release value can be set as follows. If the following is not complied with, an error window will appear and data input will not work. Check it when setting the parameter.

☞ 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.

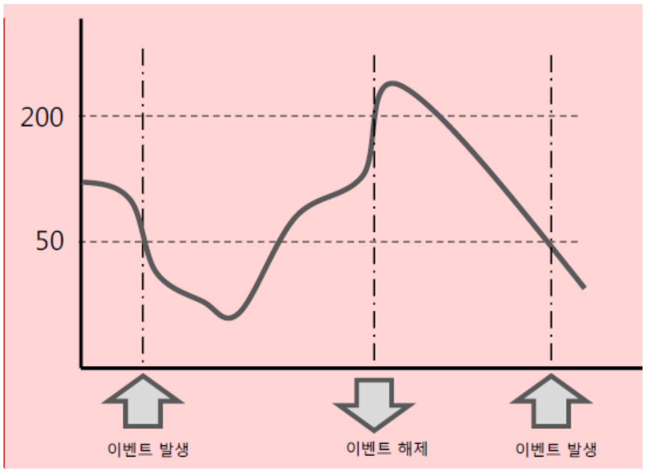
☞ If %MW0 exceeds 100, an event occurs, and the data is saved. However, since the cancelation value is set to 50, the data storage is performed until it reaches 50.



☞ 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.

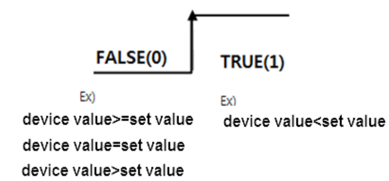
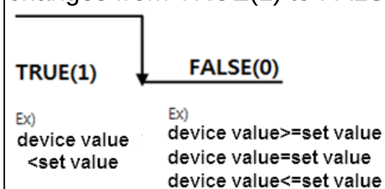
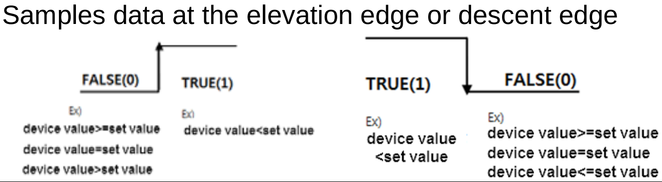
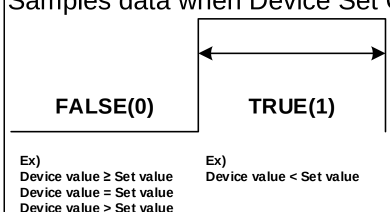
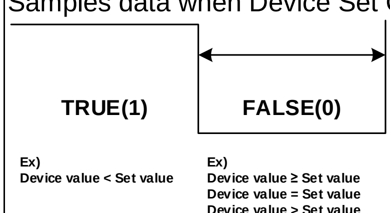
Example 2) In the word condition, if the value is set to %MW0<50, and the cancelation value is set to 200

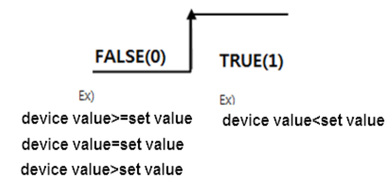
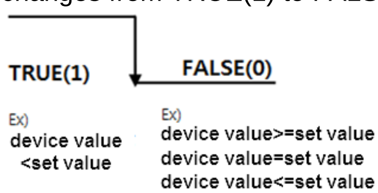
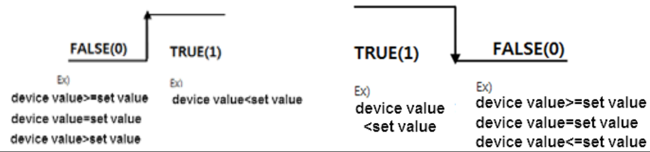
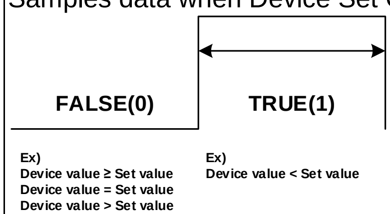
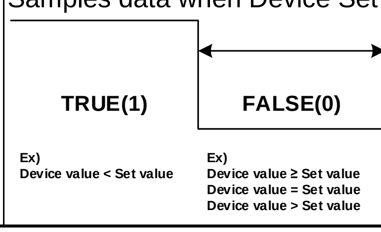
☞ If %MW0 is less than 50, an event occurs, and the data is saved. However, since the cancelation value is set to 200, the data is saved until it reaches 200.

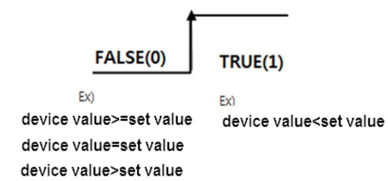
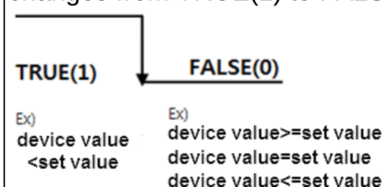
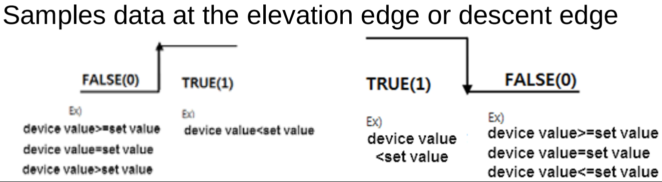
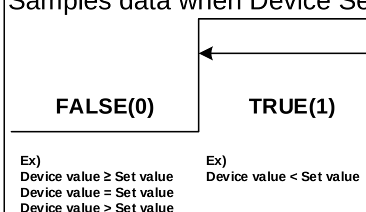
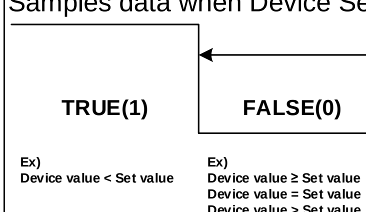


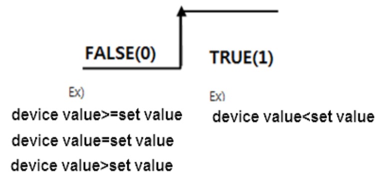
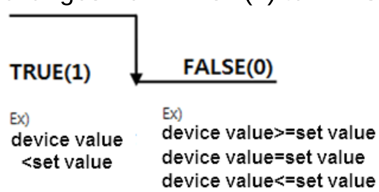
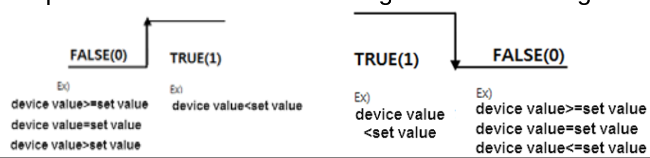
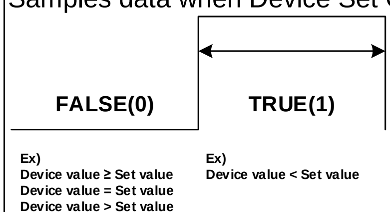
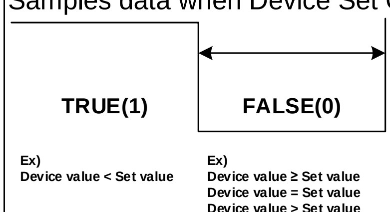
(d) Condition Description

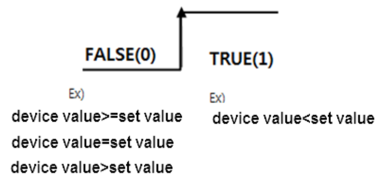
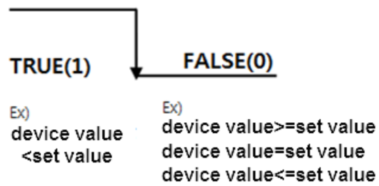
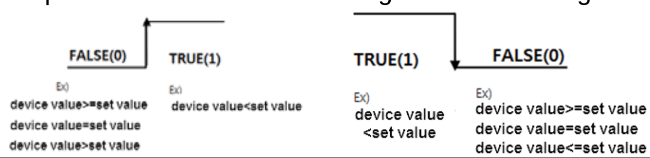
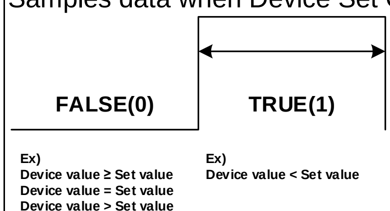
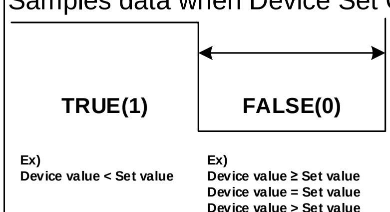
	Trigger Occurrence Condition	Device Set Condition	Operation	Release Value Setting
Bit Condition	Elevation		Saves data at elevation edge of set device bit value FALSE → TRUE 0 1	
	Descent		Saves data at descent edge of set device bit value TRUE → FALSE 1 0	
	Transfer		Saves data when set device bit value 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	

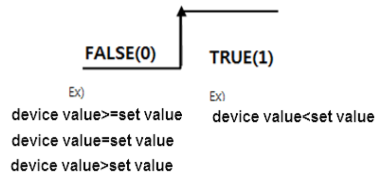
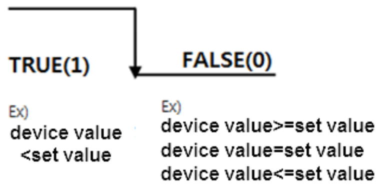
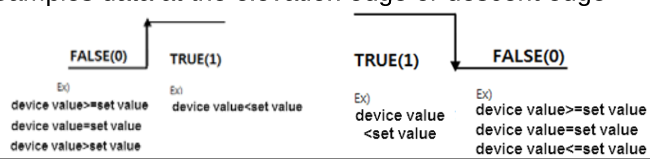
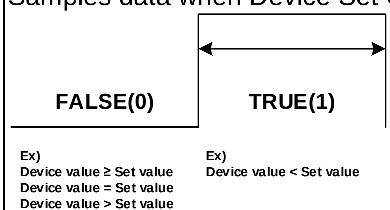
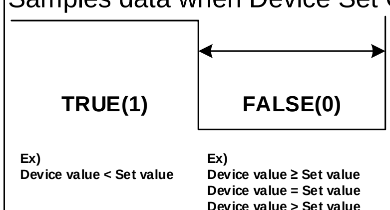
	Trigger Occurrence Condition	Device Set Condition	Operation	Release Value Setting
Word Condition	Elevation	Small	Samples data at the elevation edge when Device Set Condition changes from FALSE (0) to TRUE(1). 	Setting Available
	Descent		Samples data at the descent edge when Device Set Condition changes 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) 	

	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) 	

<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>

<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">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>

<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">Not 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>

(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

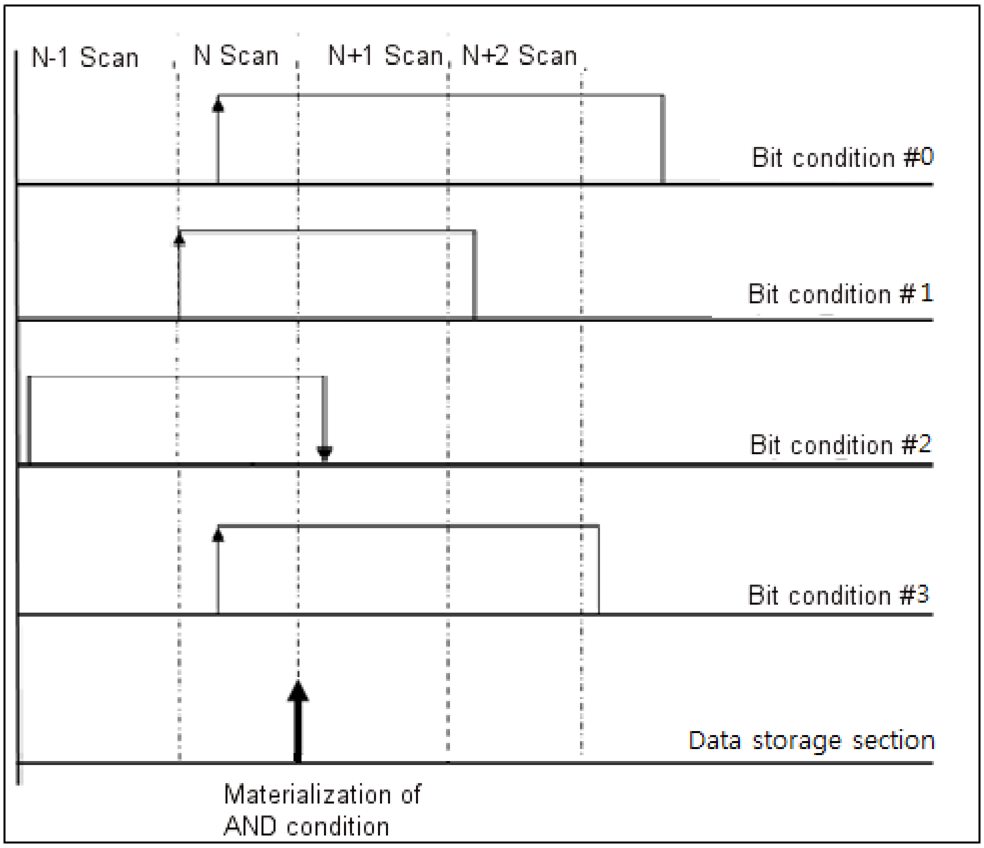
Setting	Operation	Note
AND Condition	Performs AND run with the set conditions, and saves data when the result is 1.	
OR Condition	Performs OR run with the set conditions, and saves data when the result is 1.	

(a) AND Calculation

Event occurs when all relevant conditions are satisfied at a single scan. The following is an example of activating Event Save.

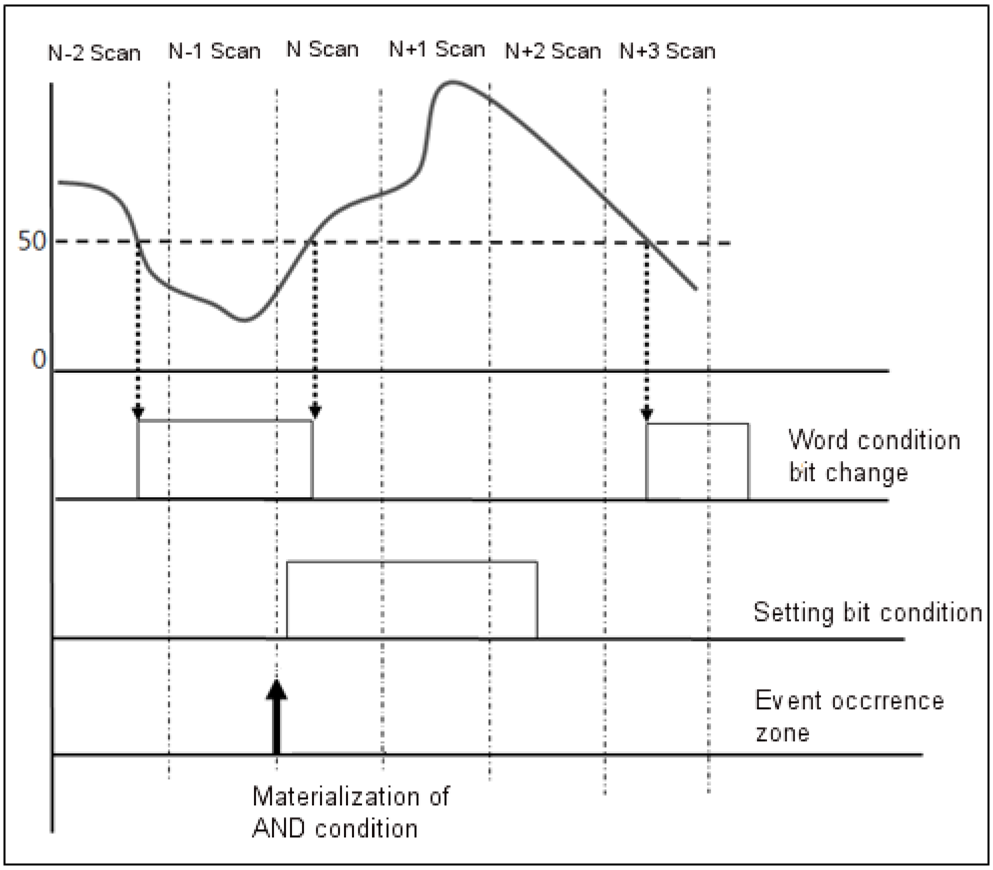
☞ When setting only with BIT condition

	Condition	Set Device	Event Occurrence Condition
Condition 0	BOOL	%MX1010	Elevation
Condition 1	BOOL	%IX1	
Condition 2	BOOL	%MX2010	
Condition 3	BOOL	%QX130	



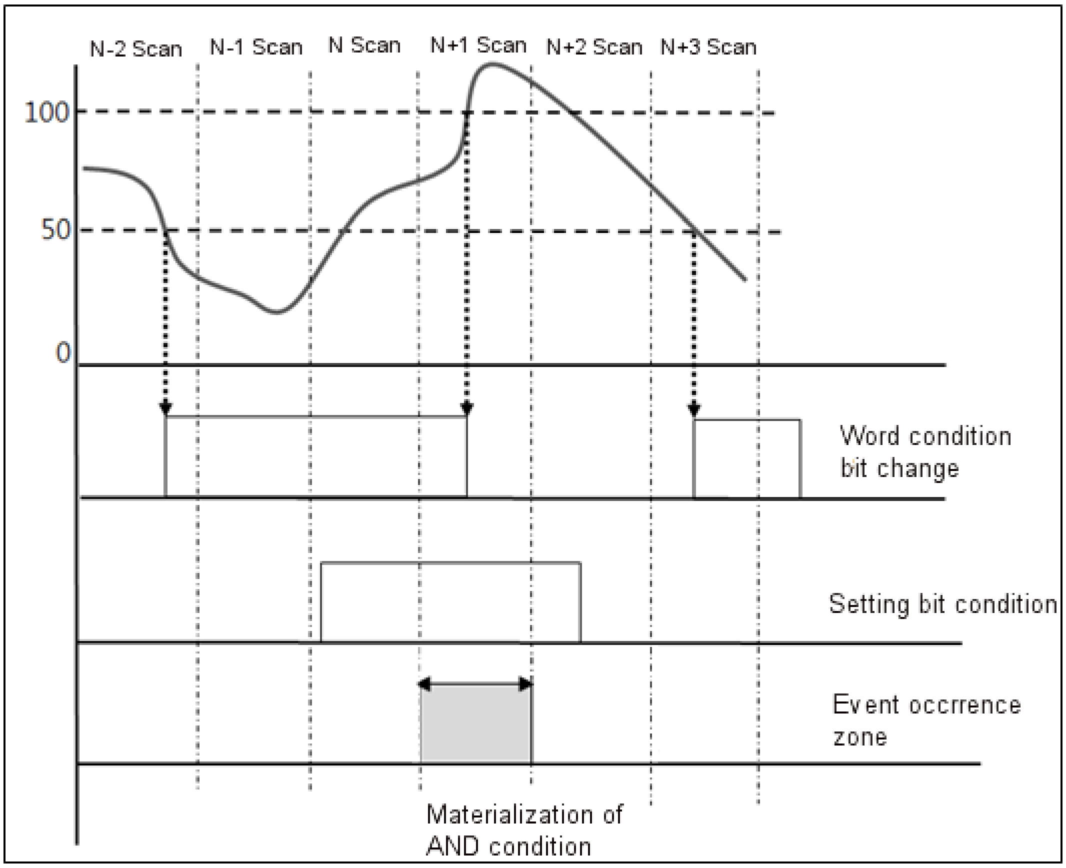
☞ When setting with combination of BIT and WORD conditions (no release value set)

	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	

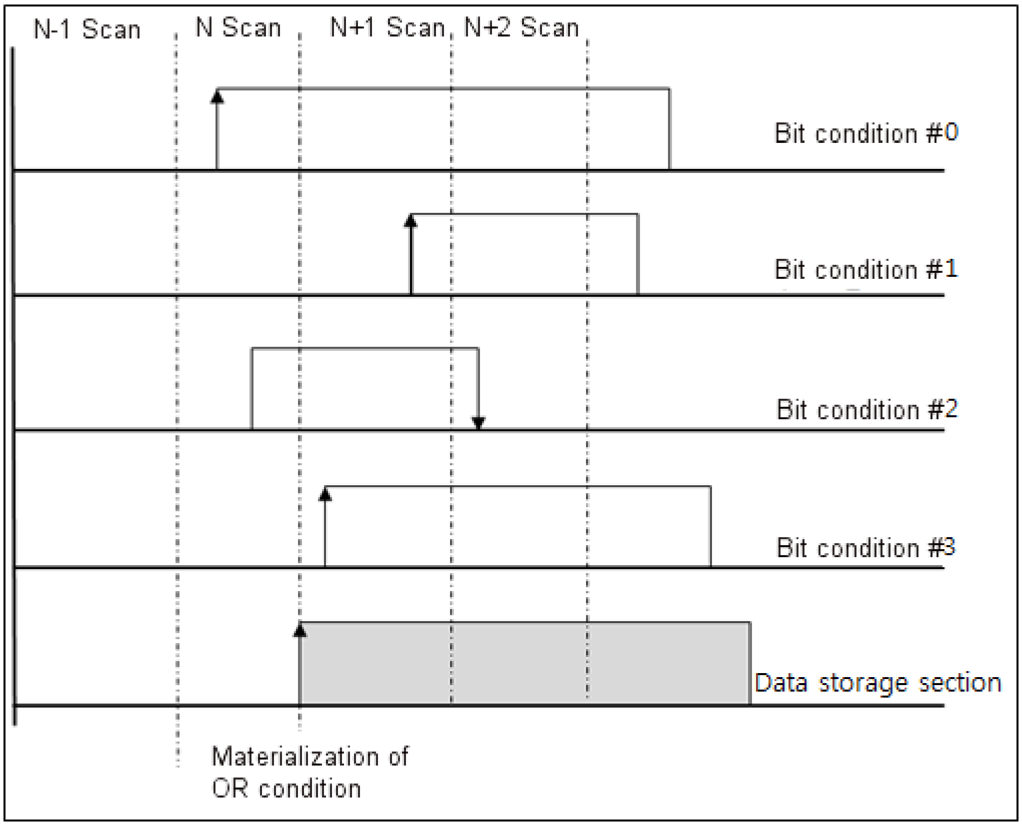


(b) OR Calculation

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.

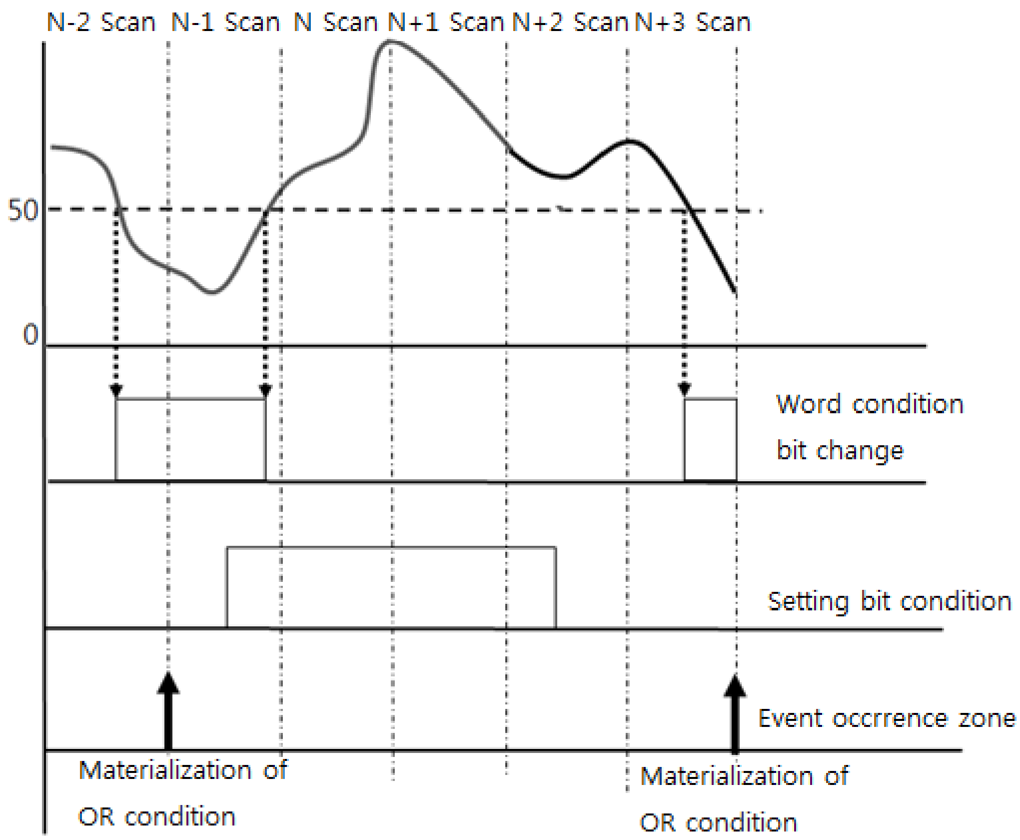
☞ When setting only with BIT condition

	Condition	Set Device	Event Occurrence Condition
Condition 0	BOOL	%MX1010	ON
Condition 1	BOOL	%IX1	
Condition 2	BOOL	%MX2010	
Condition 3	BOOL	%QX130	



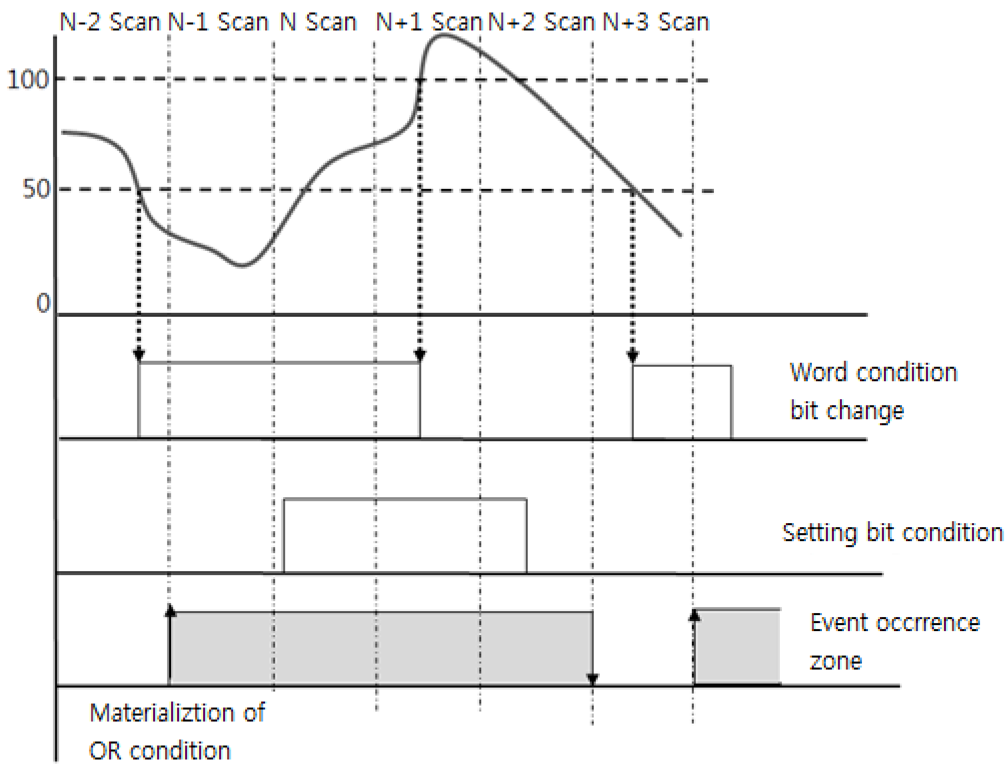
☞ When setting with combination of BIT and WORD conditions (no release value set)

	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)

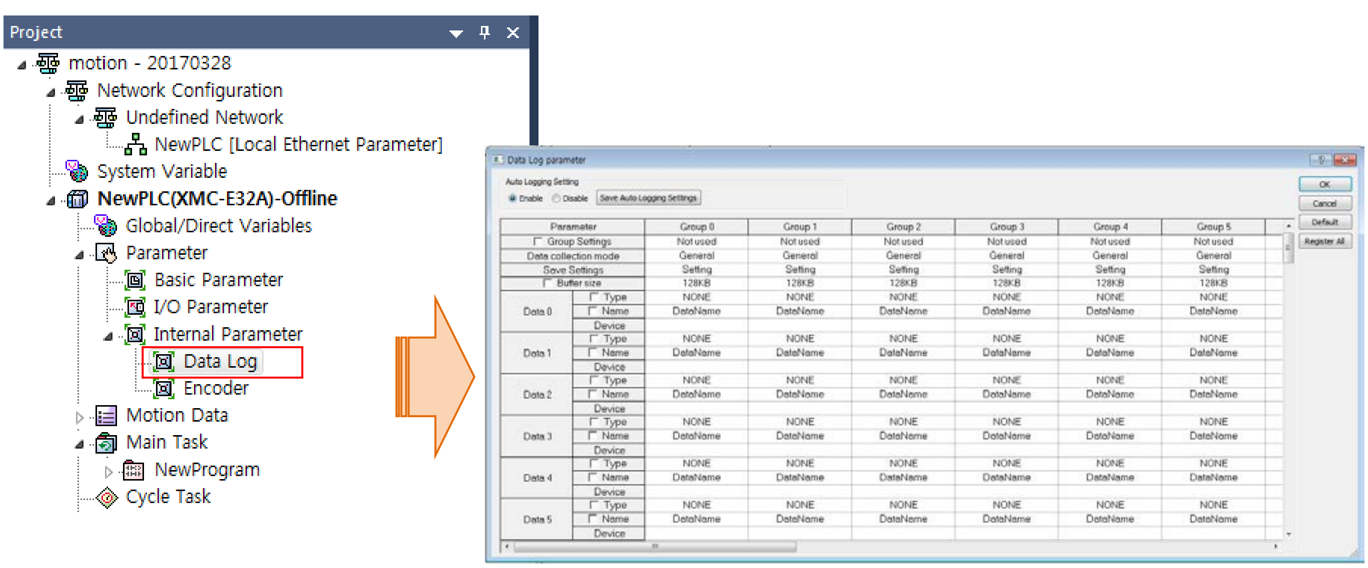
	Condition	Comparison Condition	Set Value	Release Value	Set Device	Event Occurrence Condition
Condition 0	Word	<	50	100	%MW10	ON
Condition 1	BOOL				%MX15	



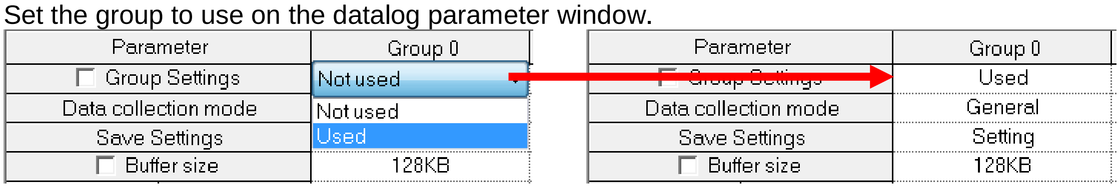
11.6.2 Setting Method

(1) Single BIT Condition

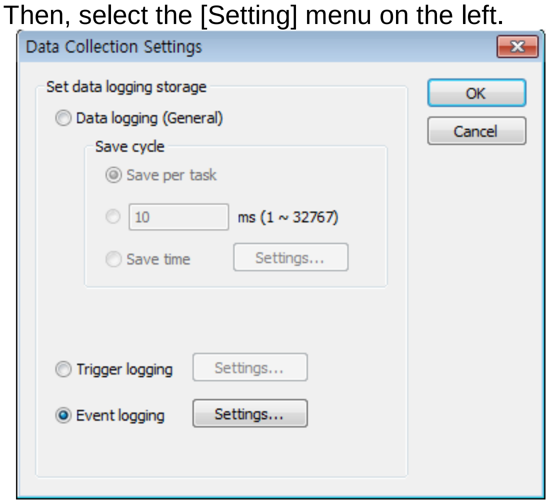
(a) Choose XG5000 – [Project Window] - [internal parameter] - [datalog]
This activates the datalog parameter setting window.



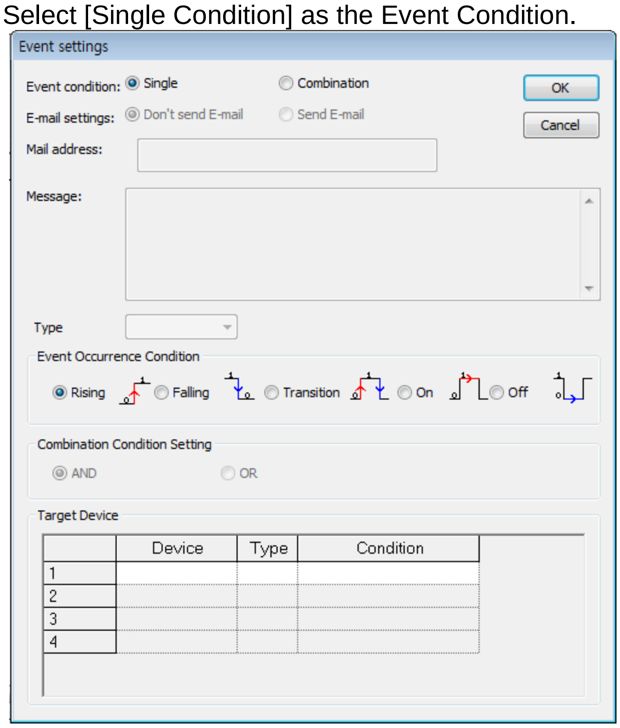
(b) Set the group to use on the datalog parameter window.



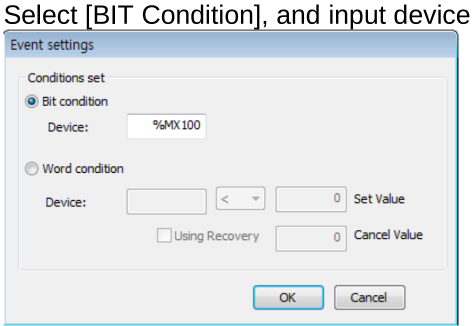
(c) Select [Event Logging] at [Data Collection Method] to activate [Setting] menu on the left.
Then, select the [Setting] menu on the left.



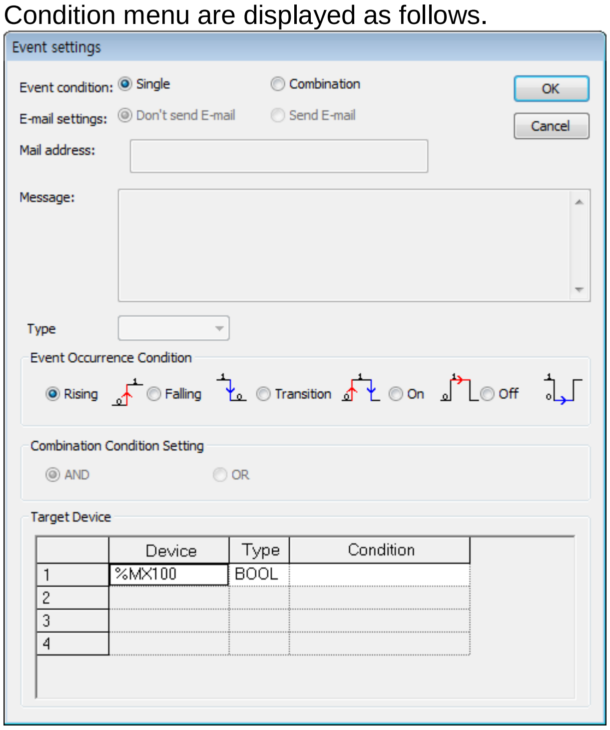
(d) Upon selection, the following window is activated for event setting.
Select [Single Condition] as the Event Condition.



(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.



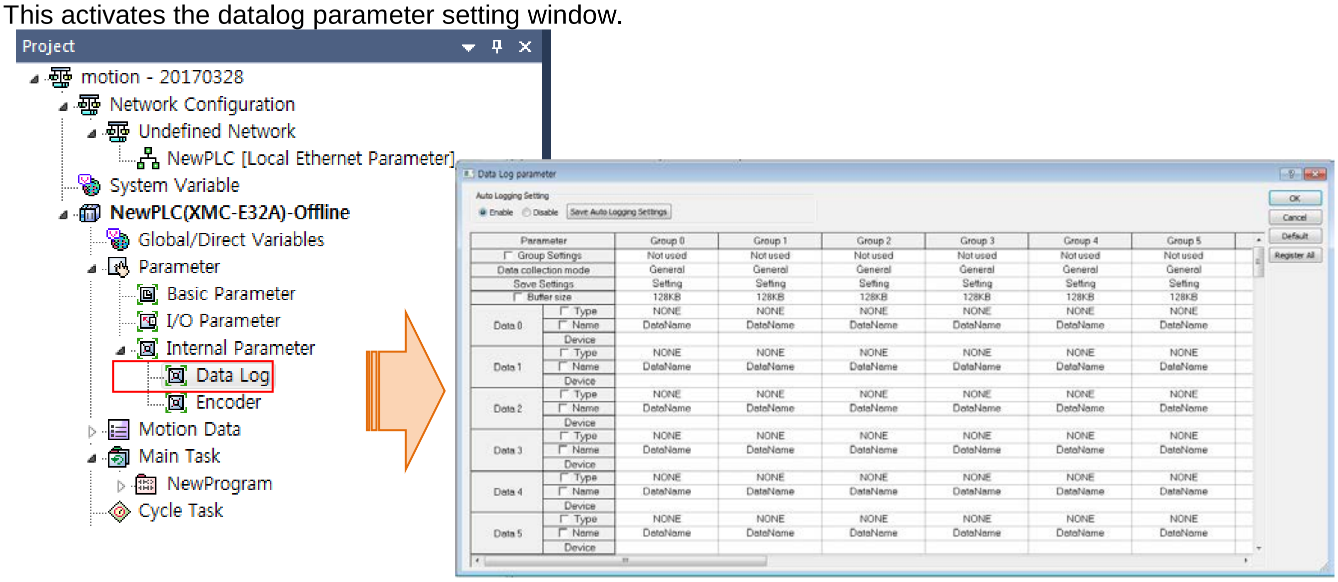
When setting is complete, the window closes and the conditions initially set at the Event Setting Condition menu are displayed as follows.



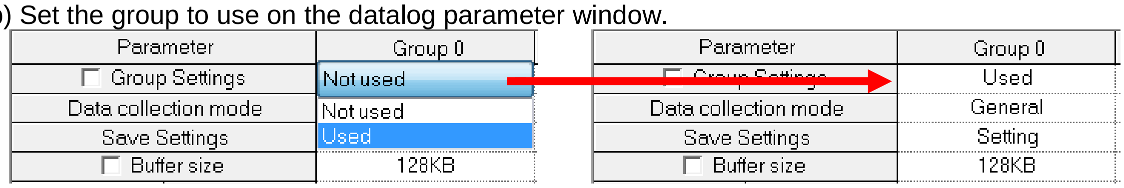
(f) Select the timing of data saving at the Event Occurrence Condition. The number and timing of data change depending on the set value.

(2) Single WORD Condition

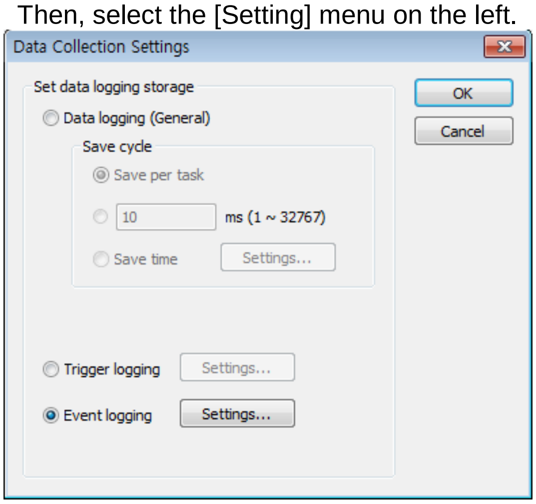
(a) Choose XG5000 – [Project Window] - [internal parameter] - [datalog]
This activates the datalog parameter setting window.



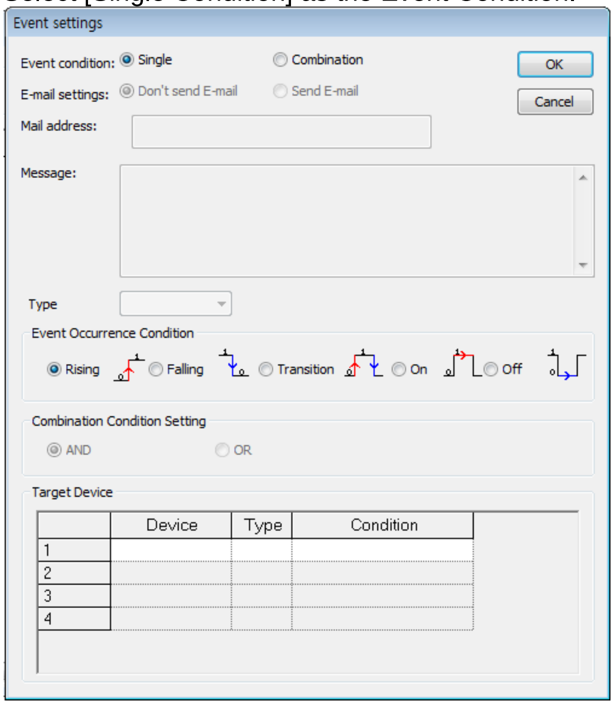
(b) Set the group to use on the datalog parameter window.



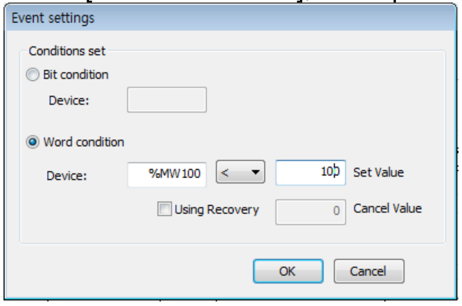
(c) Select [Event Logging] at [Data Collection Method] to activate [Setting] menu on the left.
Then, select the [Setting] menu on the left.



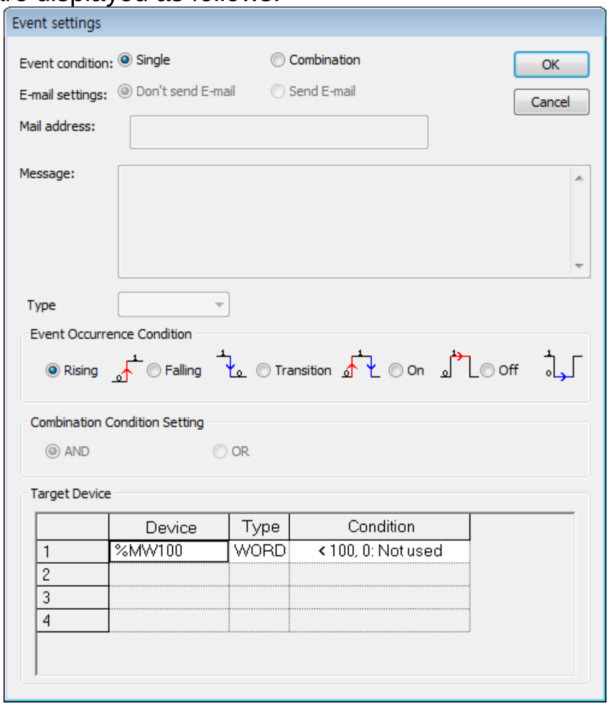
(d) Upon selection, the following window is activated for event setting.
Select [Single Condition] as the Event Condition.



(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.



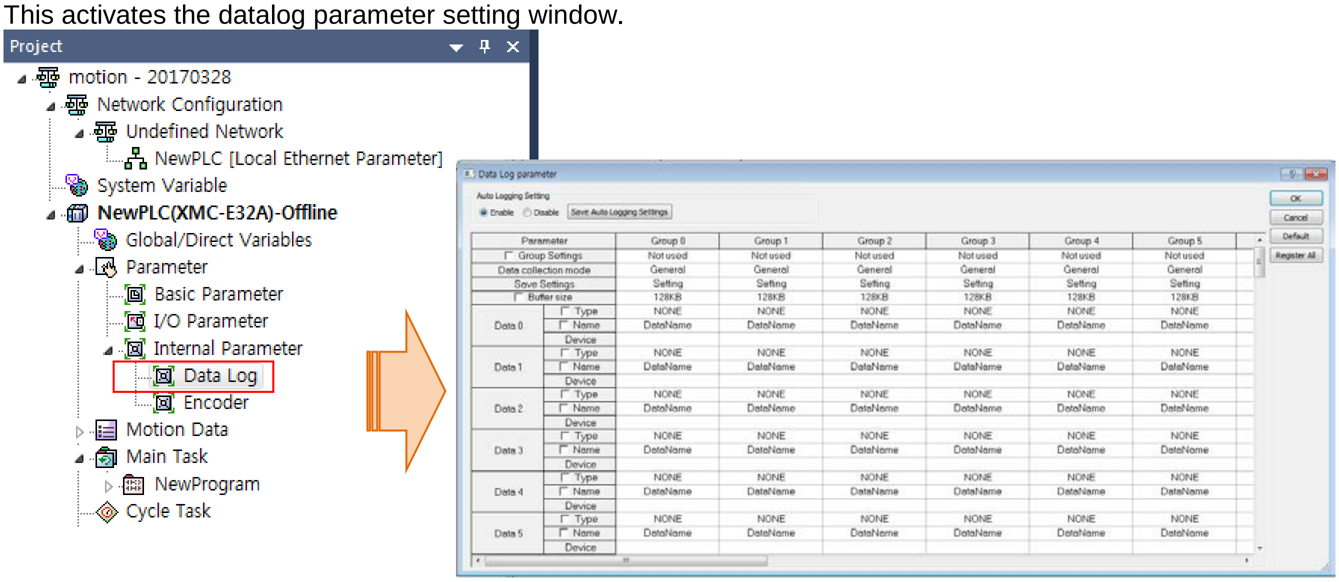
When setting is complete, the window closes and the conditions initially set at the Event Setting Condition menu are displayed as follows.



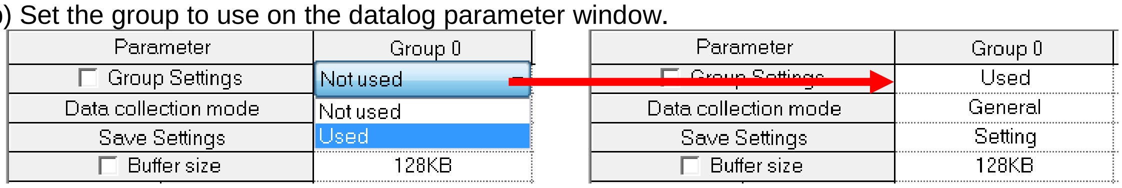
(f) Select the timing of data saving at the Event Occurrence Condition. The number and timing of data change depending on the set value.

(3) Multiple AND Condition

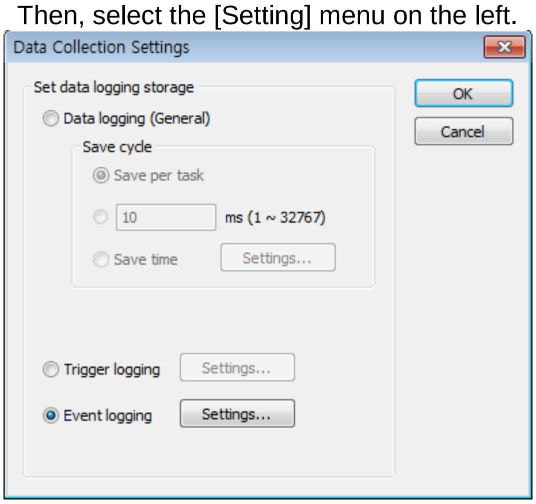
(a) Choose XG5000 – [Project Window] - [internal parameter] - [datalog]
This activates the datalog parameter setting window.



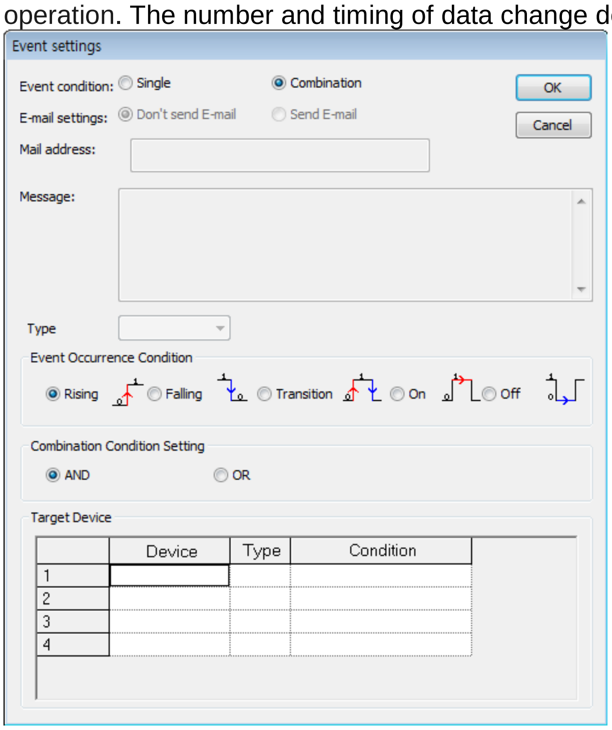
(b) Set the group to use on the datalog parameter window.



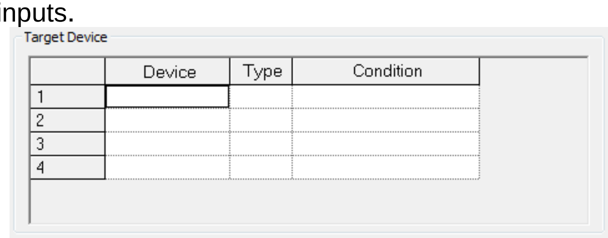
(c) Select [Event Logging] at [Data Collection Method] to activate [Setting] menu on the left.
Then, select the [Setting] menu on the left.



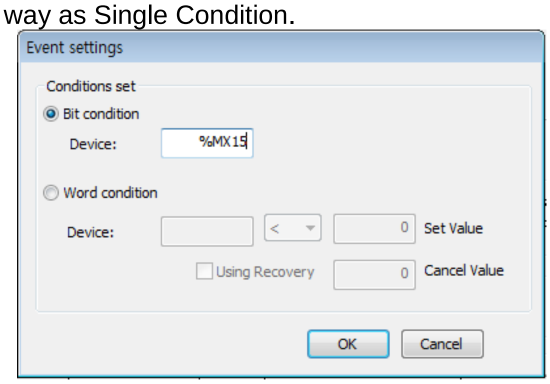
(d) Select the timing of data saving at the Event Occurrence Condition and set the operation condition to AND operation. The number and timing of data change depending on the set value and Time.

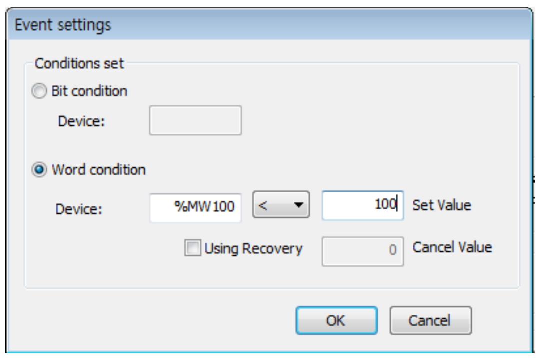


(e) Select [Event Condition] and [Multiple Condition] to activate the condition setting window which allows for up to 4 inputs.

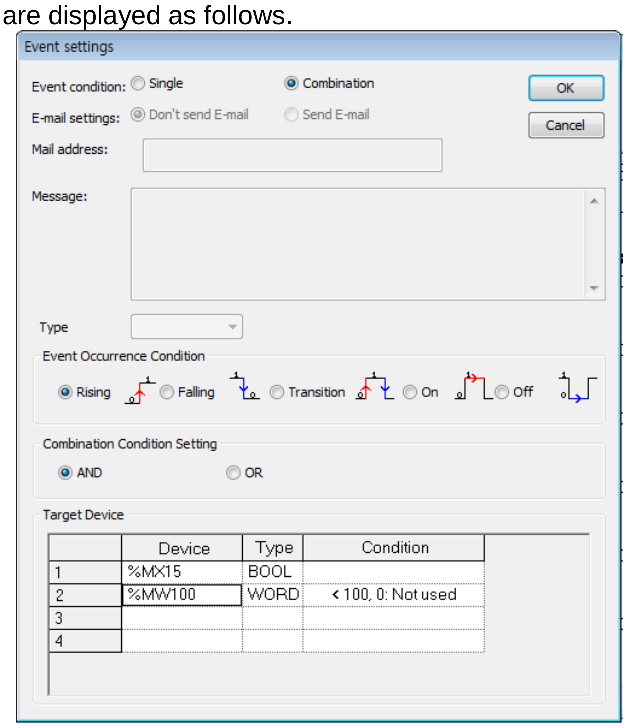


(f) Select each condition setting menu one by one, inputting specific set values. [Multiple Condition] activates Event Condition by calculating [Single Conditions] using the set run method. The basic setting is performed in the same way as Single Condition.

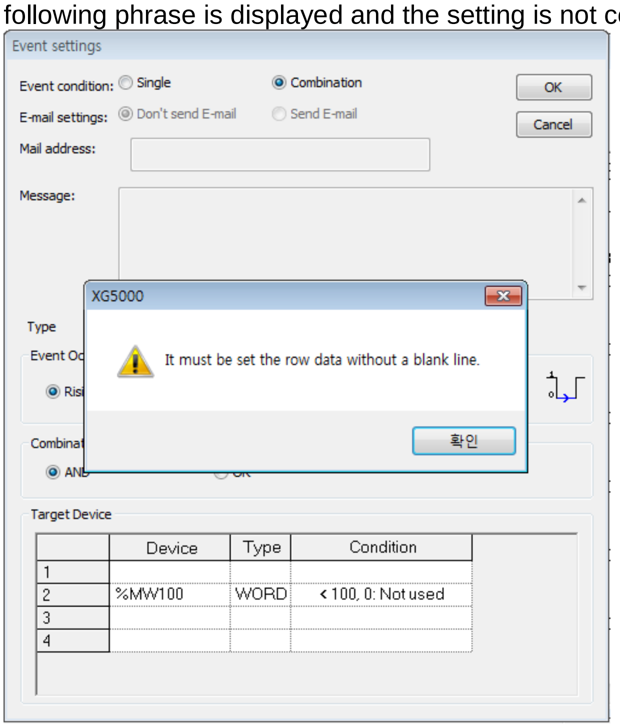




(g) When setting is complete, the window closes and the conditions initially set at the Event Setting Condition menu are displayed as follows.

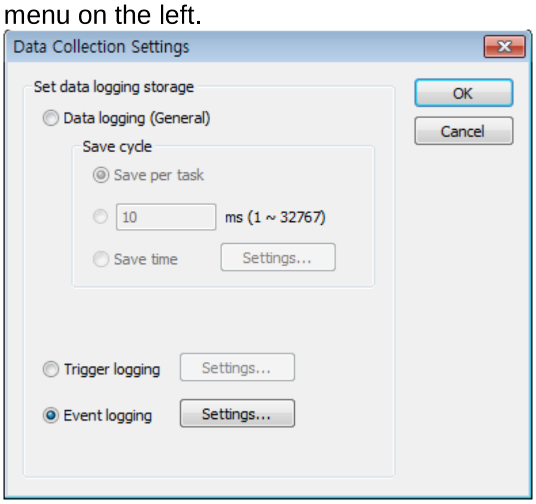


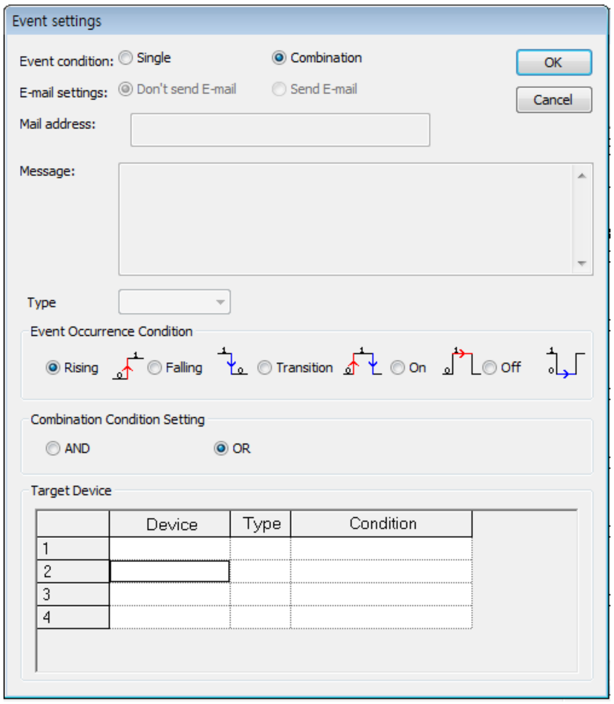
If only one [Condition Setting] is input after selecting Calculation Condition before finishing the setting, the following phrase is displayed and the setting is not complete.



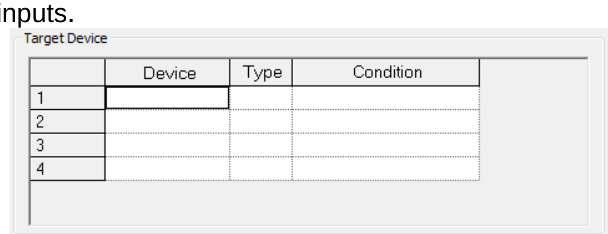
(4) Multiple OR Condition

(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.



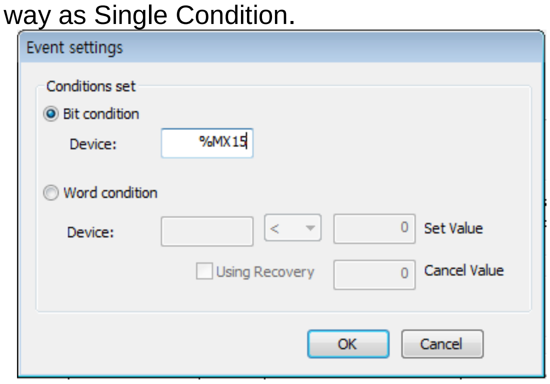


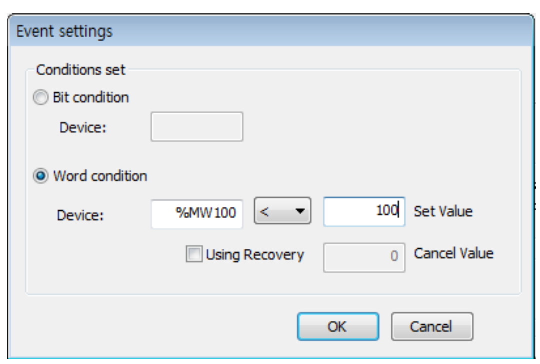
(c) Select [Event Condition] and [Multiple Condition] to activate the condition setting window which allows for up to 4 inputs.



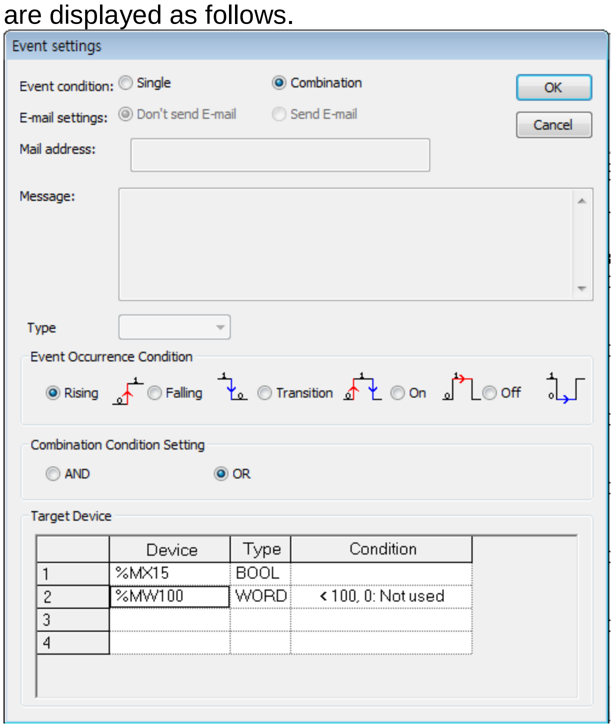
(d) Select the timing of data saving at the Event Occurrence Condition and set the operation condition to OR operation. The number and timing of data change depending on the set value.

(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.

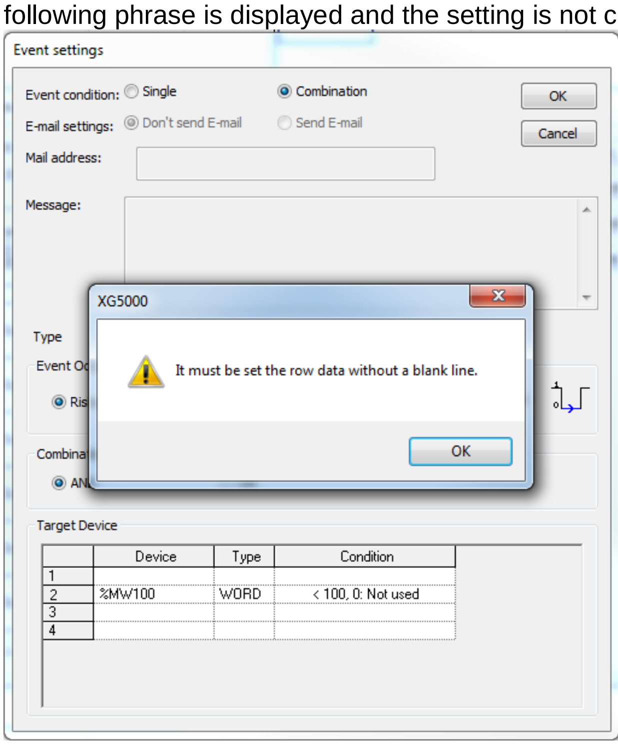




(f) When setting is complete, the window closes and the conditions initially set at the Event Setting Condition menu are displayed as follows.



(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.



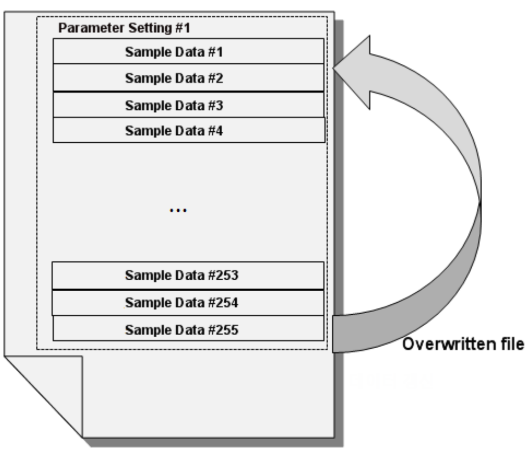
11.7 Additional Functions

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



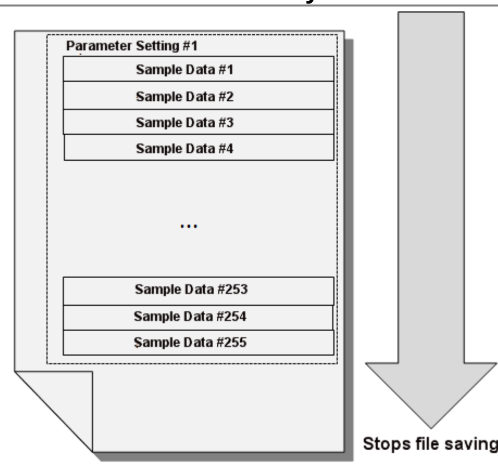
☞ Saves data in the maximum number of saved files (256 files/folder), and then goes back to the beginning to delete old files, and save the latest history.

☞ 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.

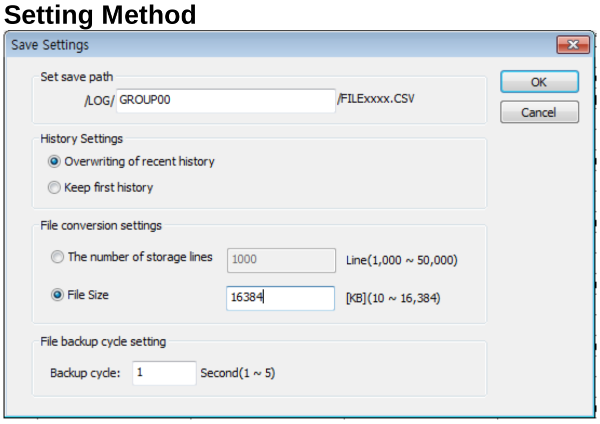
Maintains the initial history

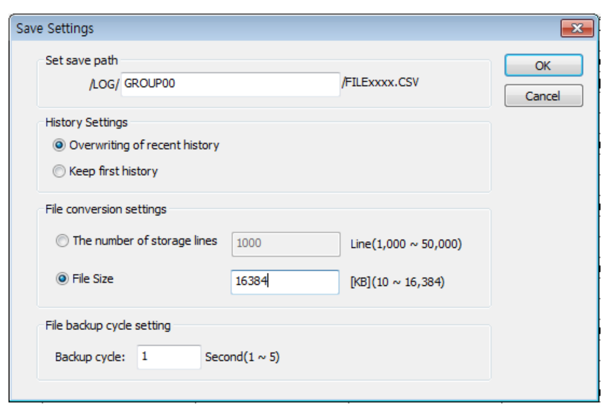


☞ Saves data in the maximum number of saved files (256 files/folder), and then stops file saving.

☞ 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 System[1)]	FAT32
Supported SD memory Capacity[2)]	2GByte ~ 32GByte
Allotted Cluster Size[3)]	32KByte (512 Sector[4)] x 8)
Volume Label[5)]	LSIS (fixed)
Motion controller Operation Mode[6)]	STOP (REMOTE available)
Formatting Mode[7)]	Fast Formatting

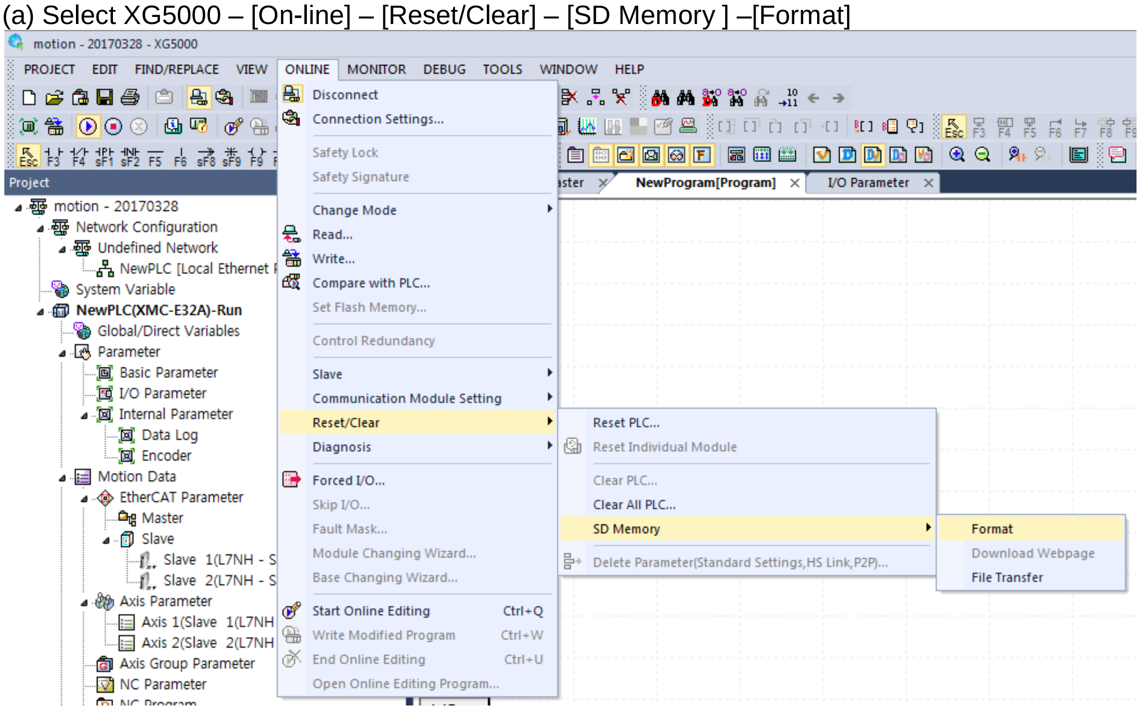
(a) File System: Rules of Saving Files into Disk
(b) Supported SD memory Capacity: MMC card not supported, 2GByte~ 32GByte SD memory supported
(Micro SD, SDHC supported)
(c) Allotted Cluster Size: 32kByte
(d) Sector: Minimum Unit for Data Saving (Default: 512 Byte)
(e) Volume Label: SD memory Card Name
(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

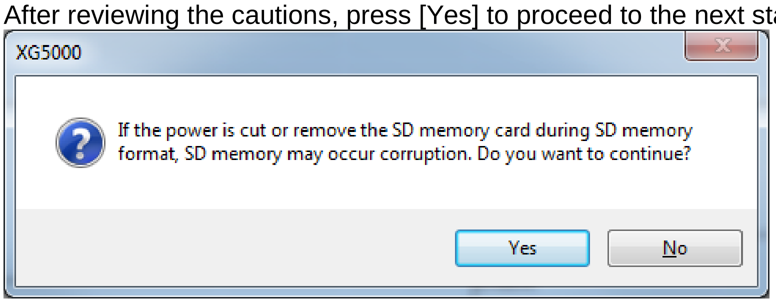
1. When performing [Formatting Function] at motion controller, all contents within the SD memory are deleted, followed by creation of a folder with the name set by the parameter.
2. If the SD card file system is not FAT32, the format function is not supported in the XG5000. In an attempt to format the 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.

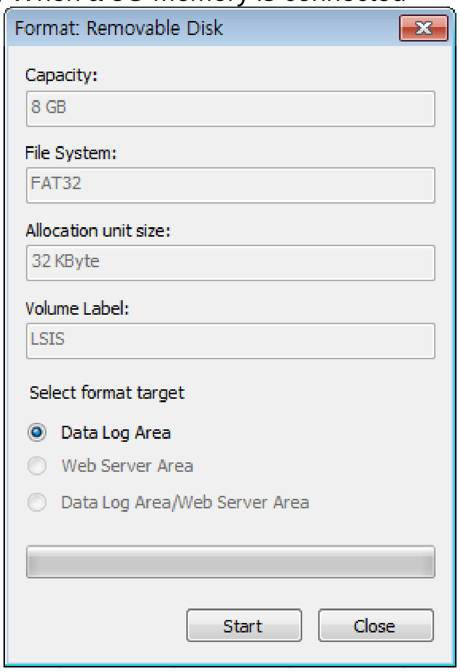


Caution

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

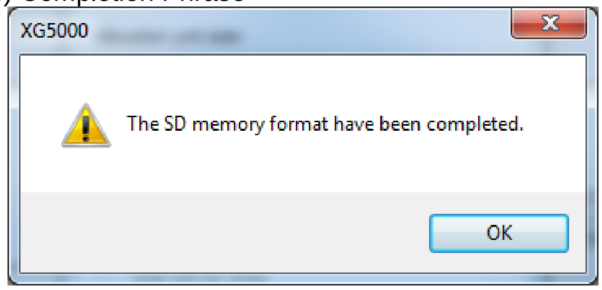
(a) Status Information

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

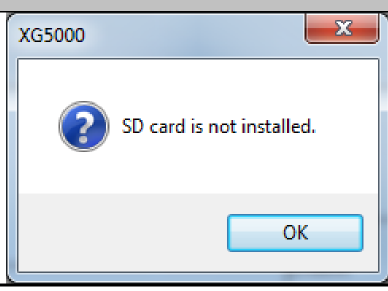
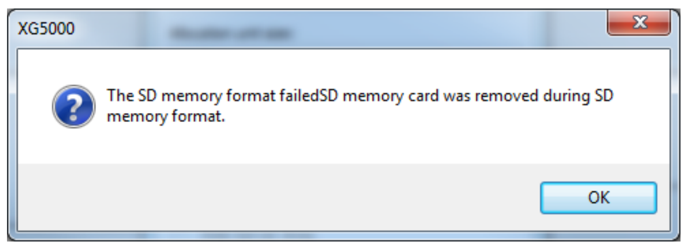
Error Code	Error Name	Error Description
0x0001	SD Detachment	When the SD memory card is forcibly removed during SD memory formatting
0x0002	File System Damage	When the file system is damaged during SD memory formatting

(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.

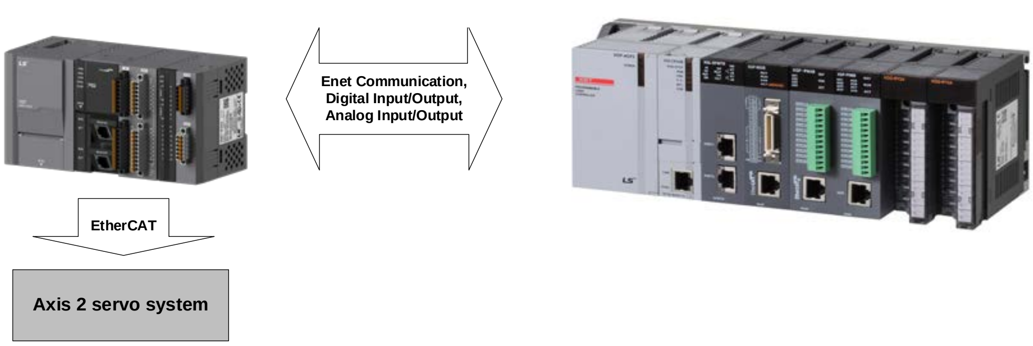
11.11.2 Save Performance by Main Task Interval

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

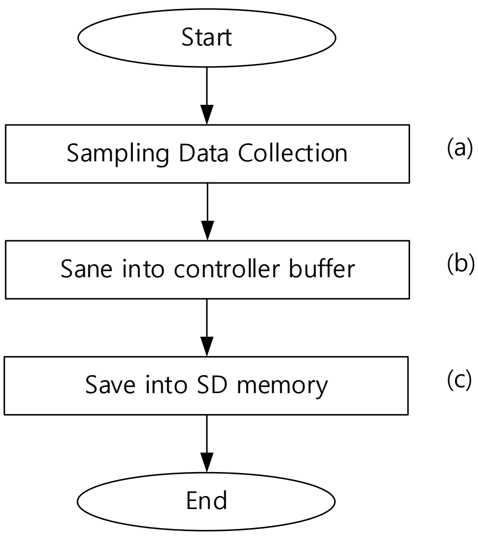
11.11.3 Save Process Time Verification

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

(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.

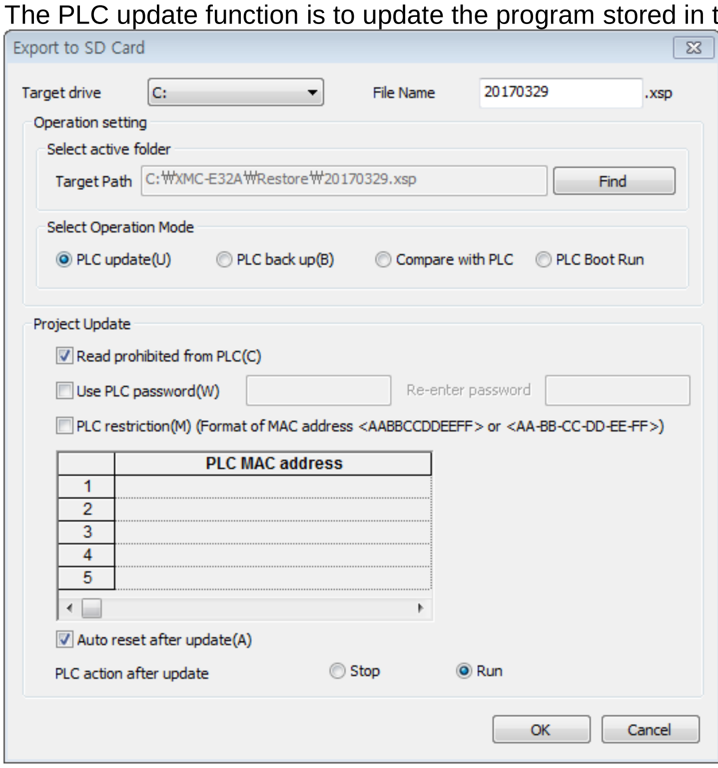
(5) Motion controller's boot operation using the SD card
- Motion controller program can be protected by password setting.
- Limiting the boot operation using the motion controller's MAC address

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.

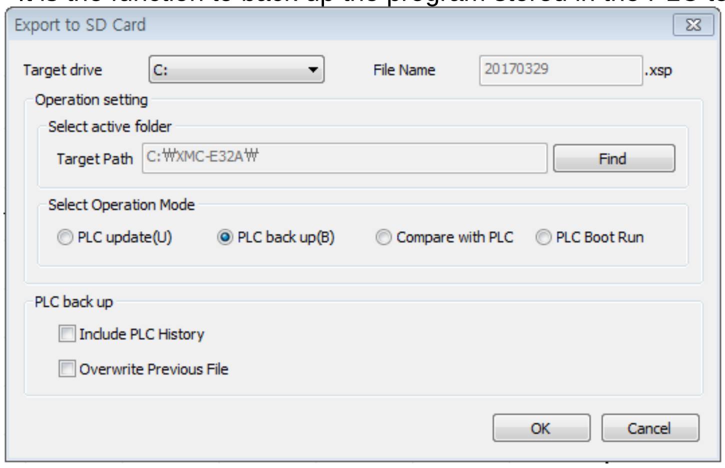


The description of each item in the PLC update mode is as follows.

Item	Description
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
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.

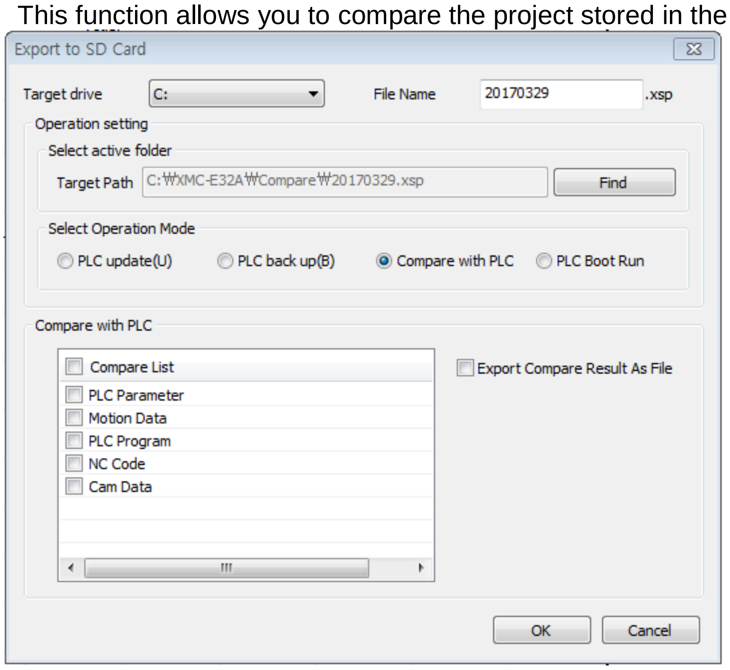


In the PLC backup mode, the description of each item is as follows.

Item	Description
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 the PLC

This function allows you to compare the project stored in the PLC with the project stored in the SD card.

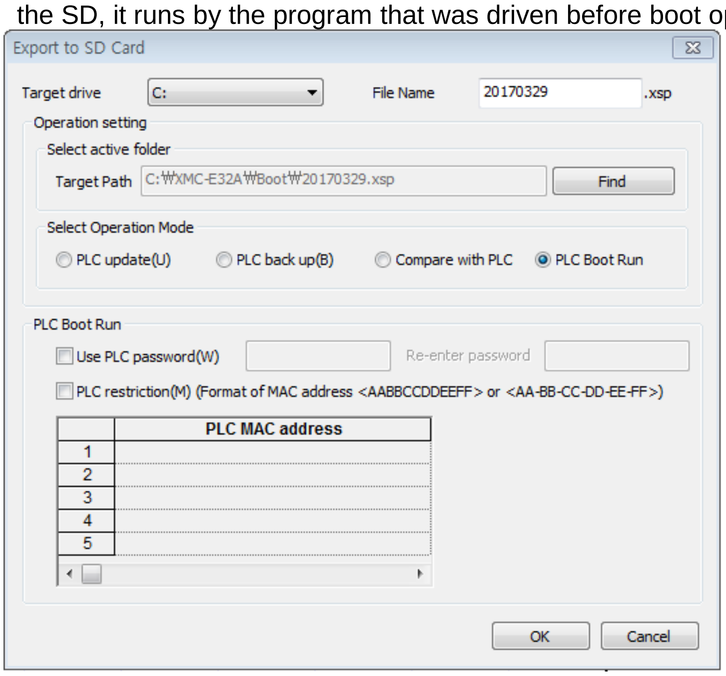


In the comparison mode with the PLC, the description of each item is as follows.

Item	Description
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 Comparison with the PLC: Compare the projects stored in the PLC and SD card
Comparison 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.



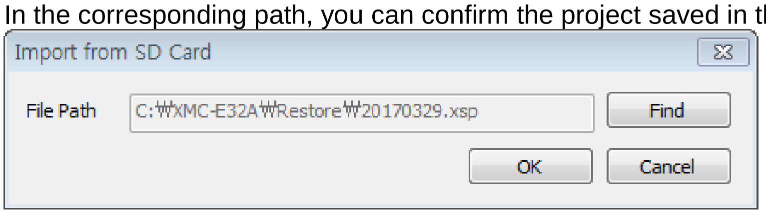
The description of each item in the PLC boot operation mode is as follows.

Item	Description
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 boot operation: Operating 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.

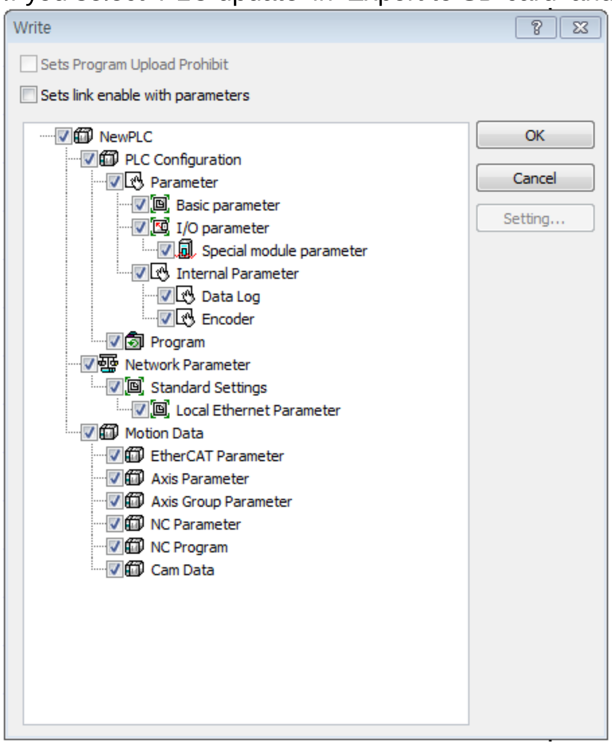
In the corresponding path, you can confirm the project saved in the SD of XG5000 is opened.



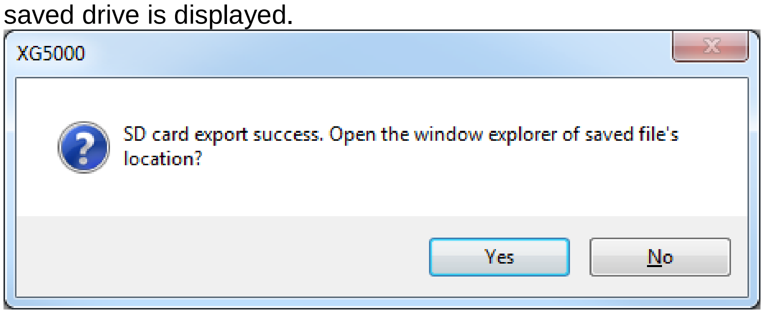
12.1.4 PLC Update Function

The PLC update function is available only when the PLC is in the STOP mode.

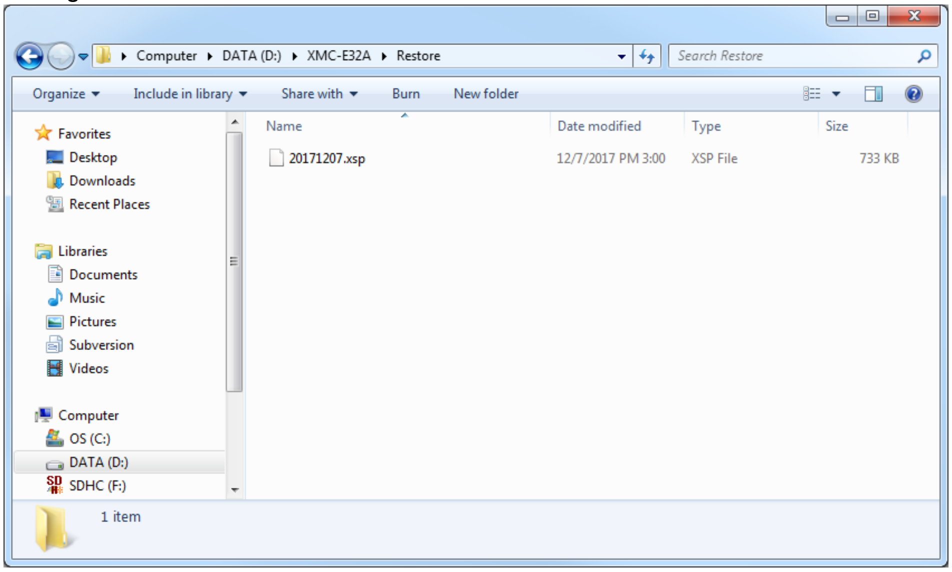
If you select 'PLC update' in 'Export to SD card' and click OK, the writing window will be created as below.



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 Config 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 features mode
0	Additional functions X
1	PLC backup function
2	PLC update function
3	Comparison with the PLC
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 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 performing the project update using the SD, the update will be executed only when the password set as the option matches the password of the PLC.

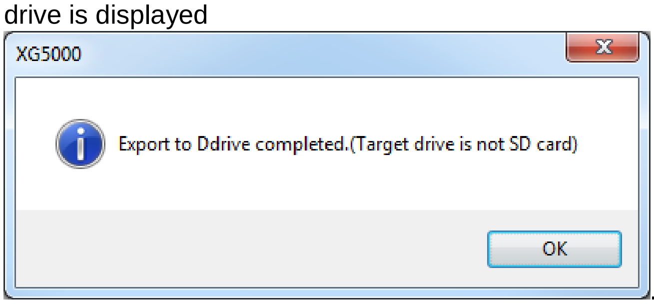
(2) If ‘Limit PLC usage’ option is set, the PLC update is performed only when the MAC address stored as the option matches the MAC address of the PLC.

(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 Backup 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 'Export to SD card' is done successfully, the window indicating successful completion is created and the saved drive is displayed



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 features mode
0	Additional functions X
1	PLC backup function
2	PLC update function
3	Comparison with the PLC
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.

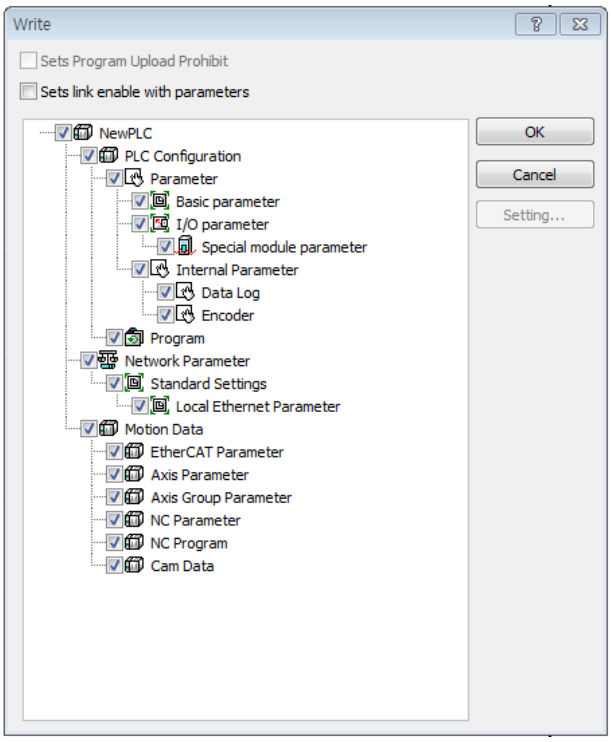
If an error occurs during operation, the SD additional function error flag (%KX8641) is turned On and the error value is displayed in the SD additional function error code (%KW542).

After removing the SD card, you can see the project will be saved under the Backup folder in the product's corresponding path and the saved project will be opened when executing 'Import from SD' in XG5000.

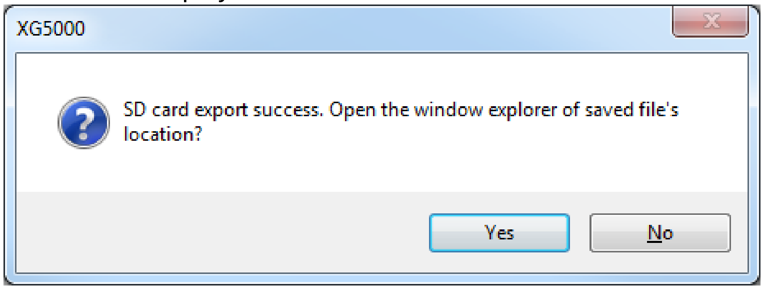
12.1.6 Comparison with the PLC

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.

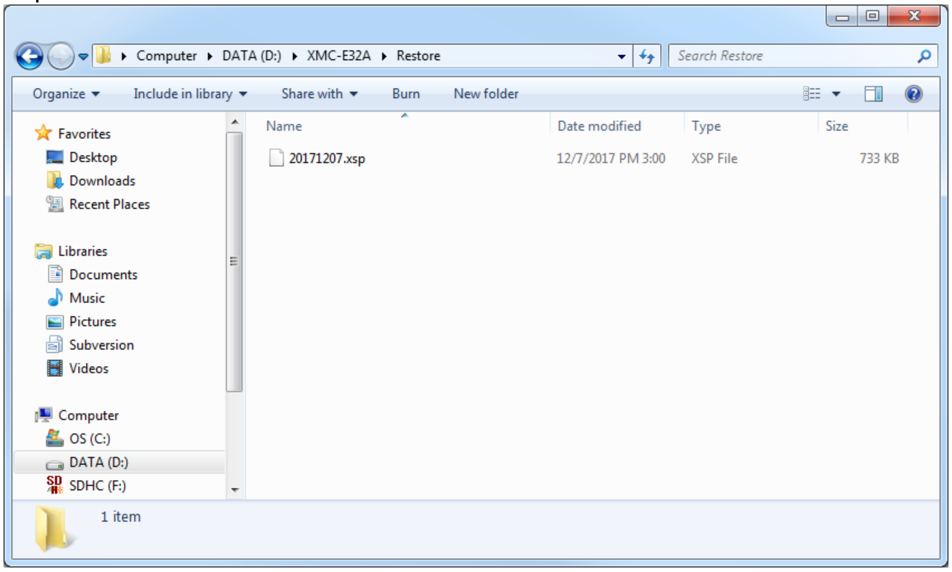
If you select 'Comparison with the PLC' in 'Export to SD card' and click OK, the writing window will be created as below.



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 features mode
0	Additional functions X
1	PLC backup function
2	PLC update function
3	Comparison with the PLC
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).

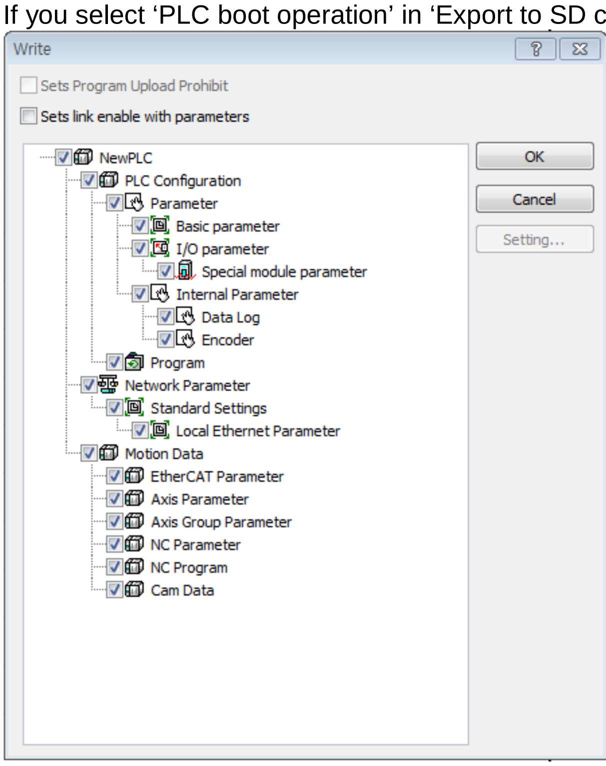
(1) When there is a discrepancy of the comparison, the SD comparison result flag (% KX8643) is turned Off and if it is the same, the SD comparison flag is turned On.

(2) If you check the item, 'Save comparison result to file' when executing 'Export to SD card', the result file (CmpResult.csv) is created in the 'Compare' folder and the comparison result is saved.

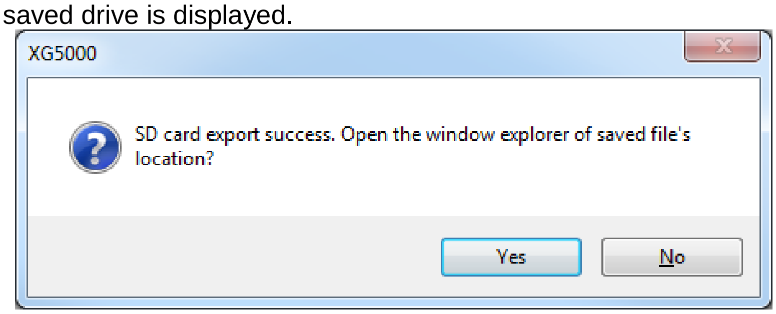
12.1.7 PLC Boot Operation

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.

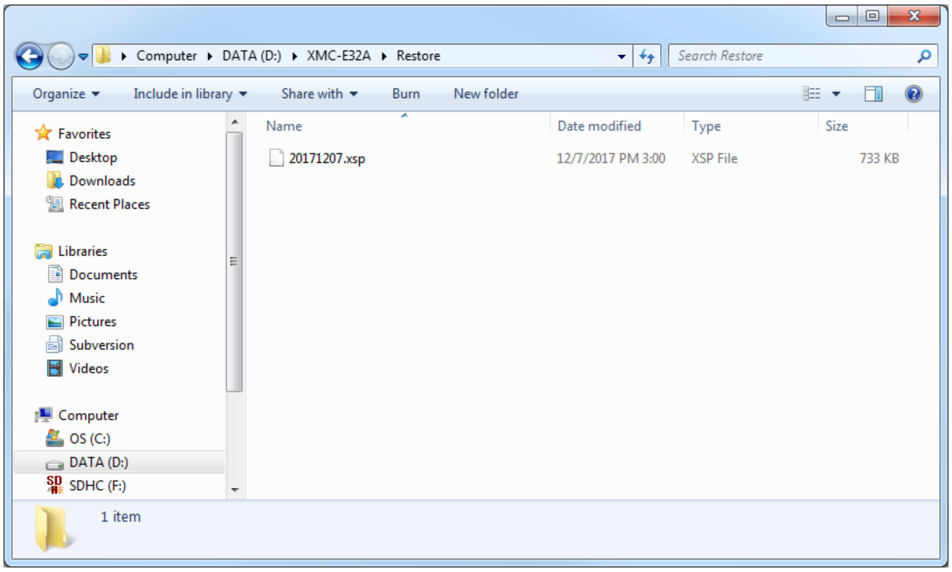
If you select ‘PLC boot operation’ in ‘Export to SD card’ and click OK, the writing window will be created as below.



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.

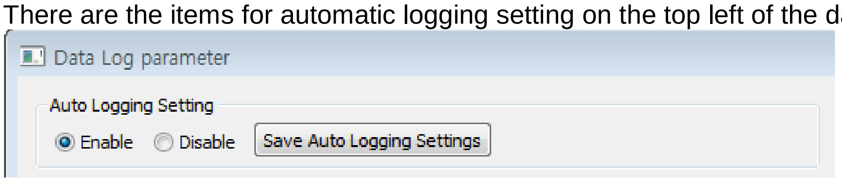
When the boot operation mode is completed normally, the flag(%KX8640) is turned Off and the completion flag of SD additional features (%KX8642) is turned On while the SD RD/WR LED and SD additional features are running.

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).

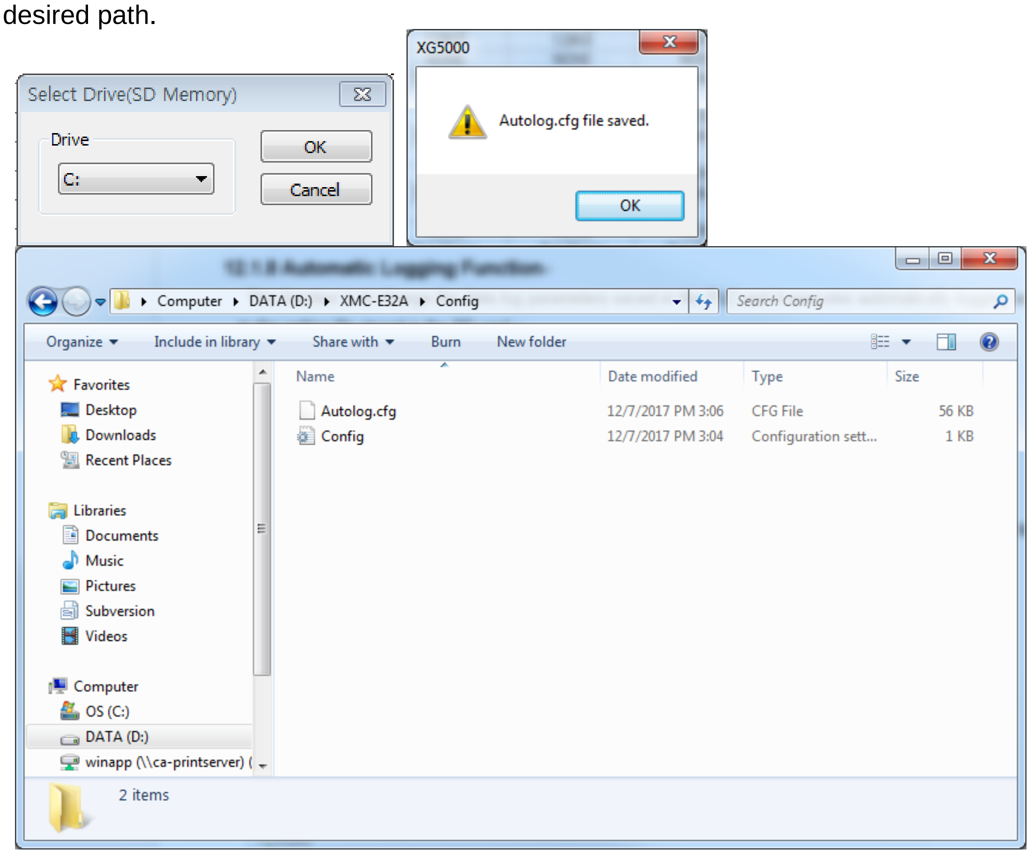
12.1.8 Automatic Logging Function

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.

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.

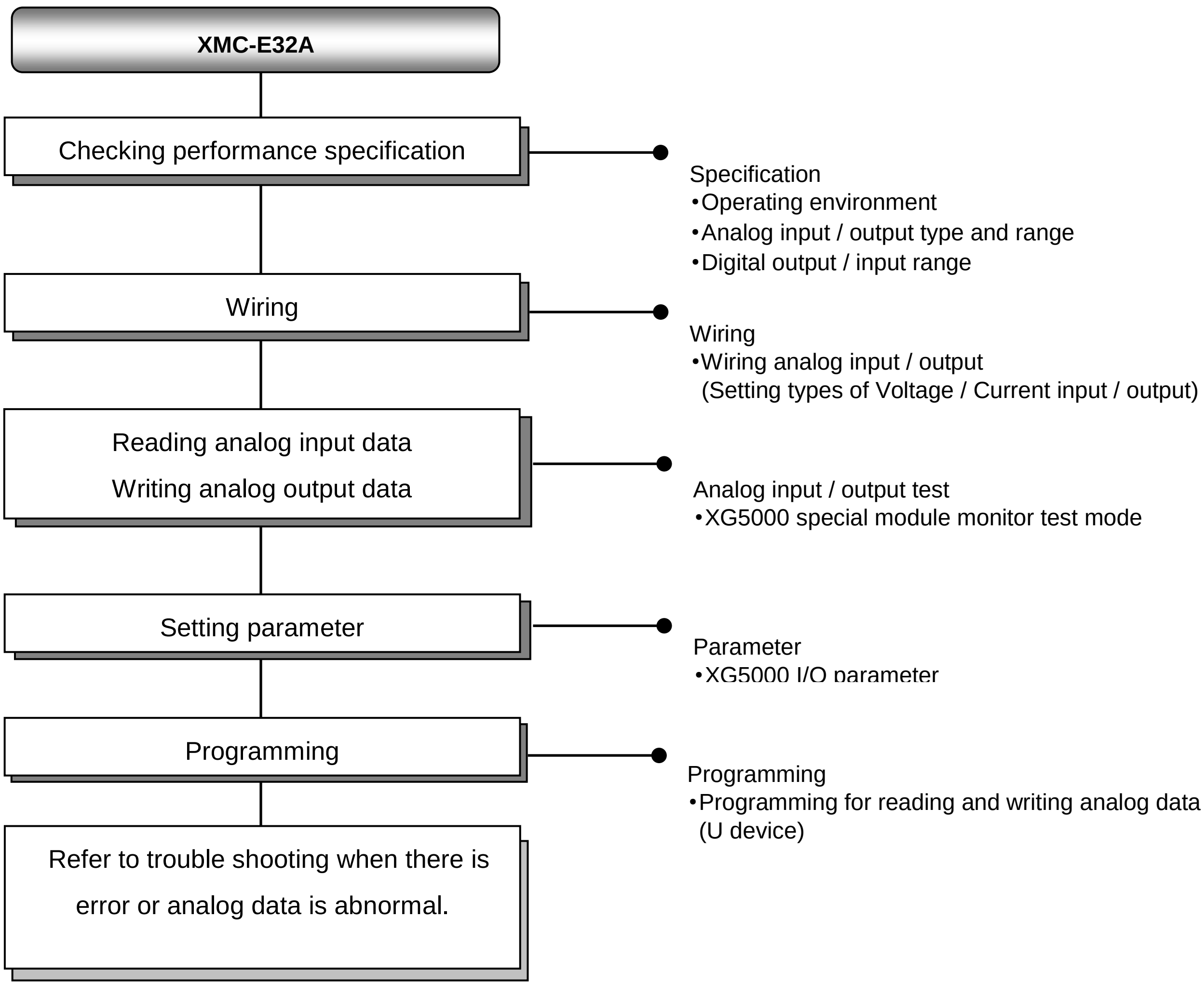
(For example, when there is no file password among PLC update functions (0x2005) → 2: additional function mode, 5: error operation)

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 Function

13.1 Overview

Before using the analog input and output function, follow steps below.



Performance specifications are as follows

(1) Input performance specification

Items		Performance specification	
Number of channels		2 channels	
Analog input	Type	Voltage	Current
	Range	1 to 5 V DC 0 to 5 V DC 0 to 10 V DC -10 to 10 V DC (Input resistance: 1 ㏁)	4 to 20 ㎃ DC 0 to 20 ㎃ DC (Input resistance 250 Ω)
		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.	
Digital range	Unsigned value	0 to 16,000	
	Signed value	-8,000 to 8,000	
	Precise value	1,000 to 5,000 (1 to 5 V) 0 to 5,000 (0 to 5 V) 0 to 10,000 (0 to 10 V) -10,000 to 10,000 (±10 V)	4,000 to 20,000 (4 to 20 ㎃) 0 to 20,000 (0 to 20 ㎃)
	Percentile value	0 to 10,000	
Max. resolution		1/16,000	
		0.250 ㎷ (1 to 5 V) 0.3125 ㎷ (0 to 5 V) 0.625 ㎷ (0 to 10 V) 1.250 ㎷ (±10V)	1.0 ㎂ (4 to 20 ㎃) 1.25 ㎂ (0 to 20 ㎃)
Accuracy		±0.2% or less (When ambient temperature 25±5℃) ±0.3% or less (When ambient temperature 0 to 55℃)	
Max. conversion speed		0.5 ㎳/channel	
Absolute max. input		±15 V DC	±30 ㎃ DC
Additional function	Filter	Digital filter (4 to 64,000 ㎳)	
	Average	Time average (4 to 16,000 ㎳)	
		Count average (2 to 64,000)	
		Moving average (2 to 100)	
		Weighted average (1 to 99%)	
	Detection alarm	Disconnection (1 to 5 V DC, 4 to 20 ㎃ DC)	
	Hold last value	When input signal exceeds the effective range, holds the last effective value.	
	Alarm function	When input signal exceeds the effective range, relevant flag turns on.	
Input terminal		6 point terminal	

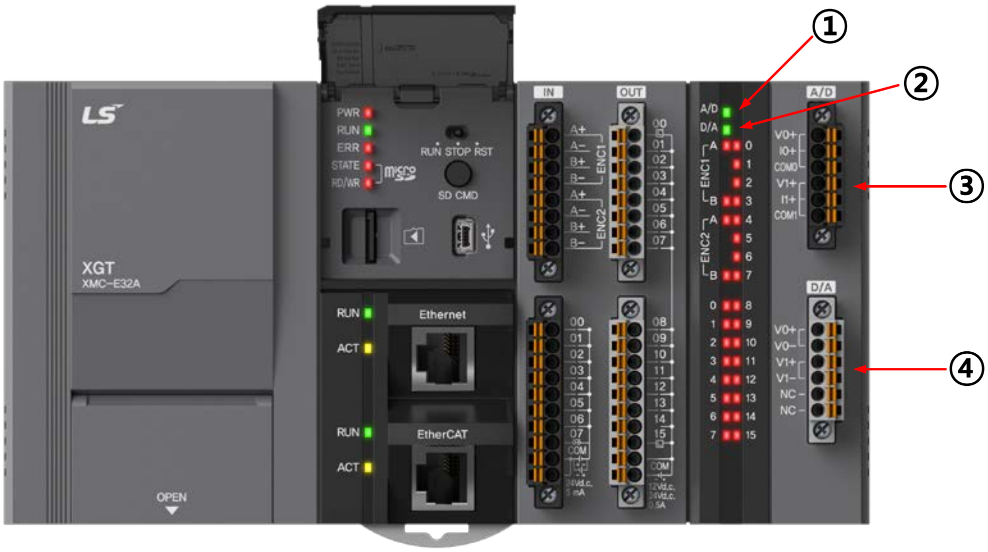
(2) Output performance specification

Items		Performance specification
Number of channels		2 channels
Analog output	Range	1 to 5 V DC 0 to 5 V DC 0 to 10 V DC -10 to 10 V DC (Load resistance: 1 ㏀ or more)
		Output ranges are set in user program or I/O parameter per each channel.
Digital range	Unsigned value	0 to 16,000
	Signed value	-8,000 to 8,000
	Precise value	1,000 to 5,000 (1 to 5 V) 0 to 5,000 (0 to 5 V) 0 to 10,000 (0 to 10 V) -10,000 to 10,000 (±10 V)
	Percentile value	0 to 10,000
Max. resolution		1/16,000
		0.250 ㎷ (1 to 5 V) 0.3125 ㎷ (0 to 5 V) 0.625 ㎷ (0 to 10 V) 1.250 ㎷ (±10 V)
Accuracy		±0.2% or less (When ambient temperature is 25±5℃) ±0.3% or less (When ambient temperature is 0 to 55℃)
Max. conversion speed		0.5 ㎳/ channel
Additional function		• 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

(3) Common performance specification

Items	Performance specification
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 point assignment: 64 points

13.2 Name of Analog Part and Functions



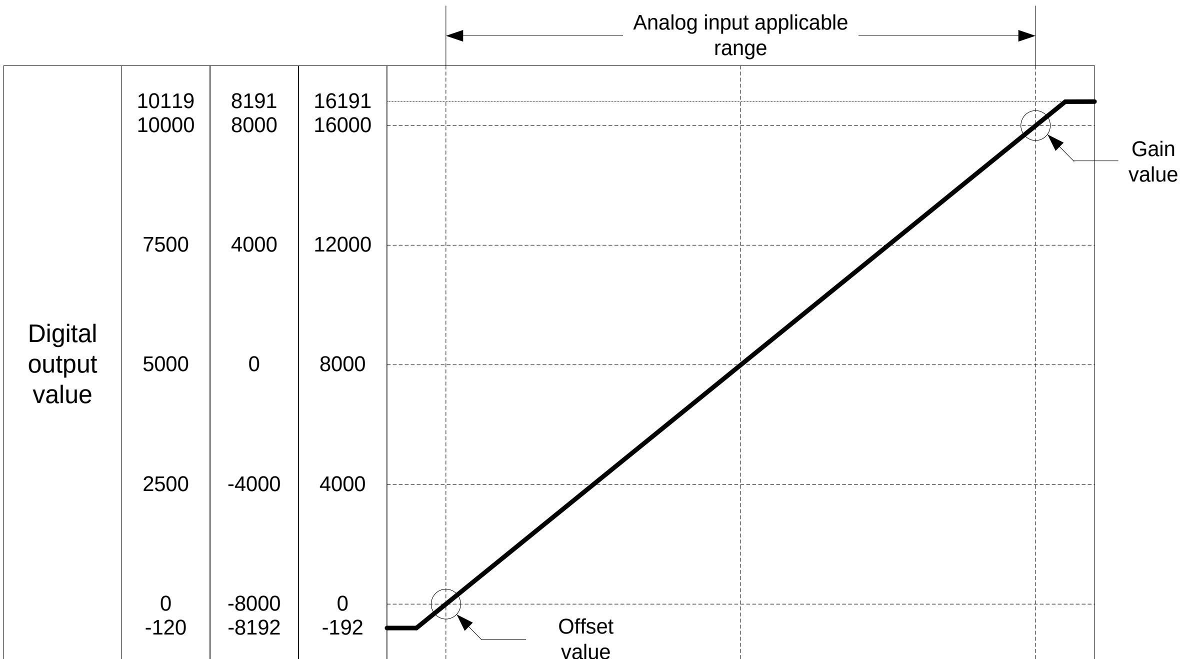
No.	Name	Description
①	A/D LED	Displays the operation status of analog input part • On: Normal operation • Blinks: Error occurs (Flickering 1s intervals) • Off: Power off or module error
②	D/A LED	Displays the operation status of analog output part • On: Operation normal • Blinks: Error occurs (Flickering 1s intervals) • Off: Power off or module error
③	Input terminal	Wiring input terminal block to connect with external device
④	Output terminal	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 shown below.

(1) Unsigned Value
(2) Signed Value
(3) Precise Value
(4) Percentile Value

13.3.1 Input Characteristic



Analog input value				
	DC 4 ~ 20 mA	4 mA	12 mA	20 mA
	DC 0 ~ 20 mA	0 mA	10 mA	20 mA
	DC 1 ~ 5 V	1 V	3 V	5 V
	DC 0 ~ 5 V	0 V	2.5 V	5 V
	DC 0 ~ 10 V	0 V	5 V	10 V
	DC -10 ~ 10 V	-10 V	0 V	10 V

(1) 4 to 20mA DC Input range

Digital output range	Analog input current (㎃)						
	3.808	4	8	12	16	20	20.191
Unsigned value (-192 to 16,191)	-192	0	4,000	8,000	12,000	16,000	16,191
Signed value (-8,192 to 8,191)	-8,192	-8,000	-4,000	0	4,000	8,000	8,191
Precise value (3,808 to 20,191)	3,808	4,000	8,000	12,000	16,000	20,000	20,191
Percentile value (-120 to 10,119)	-120	0	2,500	5,000	7,500	10,000	10,119

(2) 0 to 20 mA DC Input range

Digital output range	Analog input current (㎃)						
	-0.24	0	5	10	15	20	20.239
Unsigned value (-192 to 16191)	-192	0	4,000	8,000	12,000	16,000	16,191
Signed value (-8192 to 8191)	-8,192	-8,000	-4,000	0	4,000	8,000	8,191

Precise value (-240 to 20239)	-240	0	5,000	10,000	15,000	20,000	20,239
Percentile value (-120 to 10119)	-120	0	2,500	5,000	7,500	10,000	10,119

(3) 1 to 5 V DC Input range

Digital output range	Analog input voltage (V)						
	0.952	1	2	3	4	5	5.047
Unsigned Value (-192 to 16,191)	-192	0	4,000	8,000	12,000	16,000	16,191
Signed Value (-8,192 to 8,191)	-8,192	-8,000	-4,000	0	4,000	8,000	8,191
Precise Value (952 to 5,047)	952	1,000	2,000	3,000	4,000	5,000	5,047
Percentile Value (-120 to 10,119)	-120	0	2,500	5,000	7,500	10,000	10,119

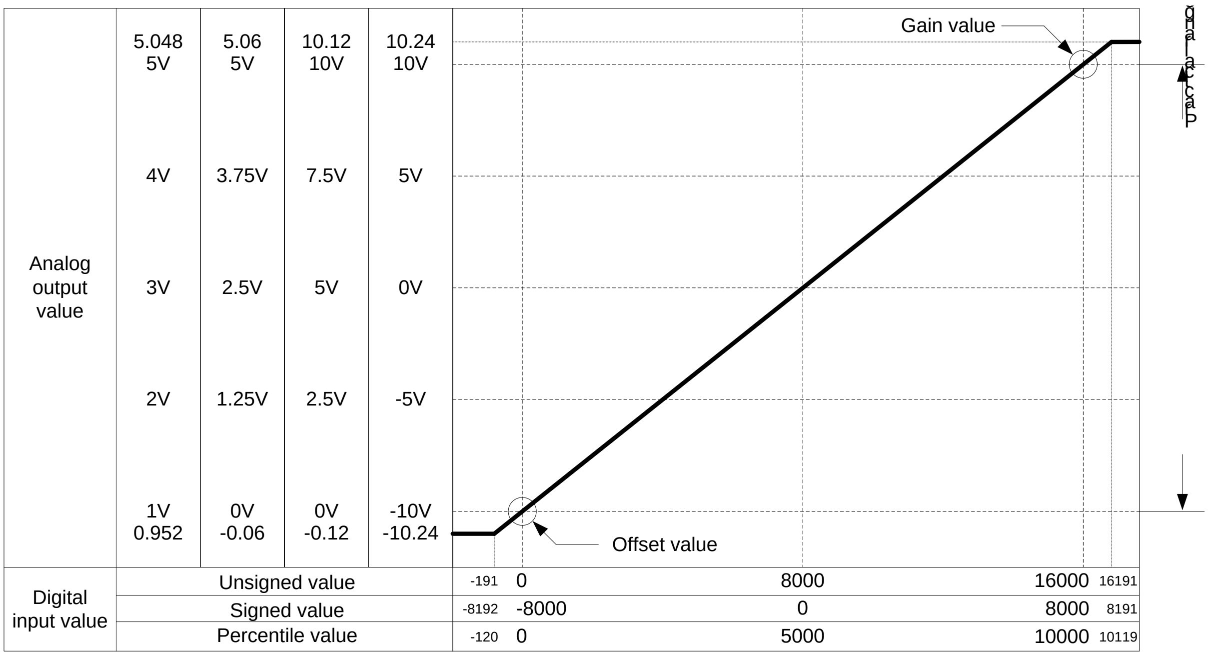
(4) 0 to 5 V DC Input range

Digital output range	Analog input voltage (V)						
	-0.06	0	1.25	2.5	3.75	5	5.059
Unsigned Value (-192 to 16,191)	-192	0	4,000	8,000	12,000	16,000	16,191
Signed Value (-8,192 to 8,191)	-8,192	-8,000	-4,000	0	4,000	8,000	8,191
Precise Value (-60 to 5,059)	-60	0	1,250	2,500	3,750	5,000	5,059
Percentile Value (-120 to 10,119)	-120	0	2,500	5,000	7,500	10,000	10,119

(5) 0 to 10 V DC Input range

Digital output range	Analog input voltage (V)						
	-0.12	0	2.5	5	7.5	10	10.119
Unsigned Value (-192 to 16,191)	-192	0	4,000	8,000	12,000	16,000	16,191
Signed Value (-8,192 to 8,191)	-8,192	-8,000	-4,000	0	4,000	8,000	8,191
Precise Value (-120 to 10,119)	-120	0	2,500	5,000	7,500	10,000	10,119
Percentile Value (-120 to 10,119)	-120	0	2,500	5,000	7,500	10,000	10,119

13.3.2 Output Characteristic



(1) 1 to 5 V DC Output range

Digital input	Analog output voltage (V)						
	0.952	**1**	**2**	**3**	**4**	**5**	**5.047**
Unsigned value (-192 to 16,191)	-192	0	4,000	8,000	12,000	16,000	16,191
Signed value (-8,192 to 8,191)	-8,192	-8,000	-4,000	0	4,000	8,000	8,191
Precise value (952 to 5,047)	952	1,000	2,000	3,000	4,000	5,000	5,047
Percentile value (-120 to 10,119)	-120	0	2,500	5,000	7,500	10,000	10,119

(2) 0 to 5 V DC Output range

Digital value	Analog output voltage (V)						
	-0.06	**0**	**1.25**	**2.5**	**3.75**	**5**	**5.059**
Unsigned value (-192 to 16,191)	-192	0	4,000	8,000	1,2000	16,000	16,191
Signed value (-8,192 to 8,191)	-8,192	-8,000	-4,000	0	4,000	8,000	8,191
Precise value (-60 to 5,059)	-60	0	1,250	2,500	3,750	5,000	5,059
Percentile value (-120 to 10,119)	-120	0	2,500	5,000	7,500	10,000	10,119

(3) 0 to 10 V DC Output range

Digital input	Analog output voltage (V)						
	-0.12	**0**	**2.5**	**5**	**7.5**	**10**	**10.119**
Unsigned value (-192 to 16,191)	-192	0	4,000	8,000	12,000	16,000	16,191
Signed value (-8,192 to 8,191)	-8,192	-8,000	-4,000	0	4,000	8,000	8,191
Precise value (-120 to 10,119)	-120	0	2,500	5,000	7,500	10,000	10,119
Percentile value (-120 to 10,119)	-120	0	2,500	5,000	7,500	10,000	10,119

(4) -10 to 10 V DC Output range

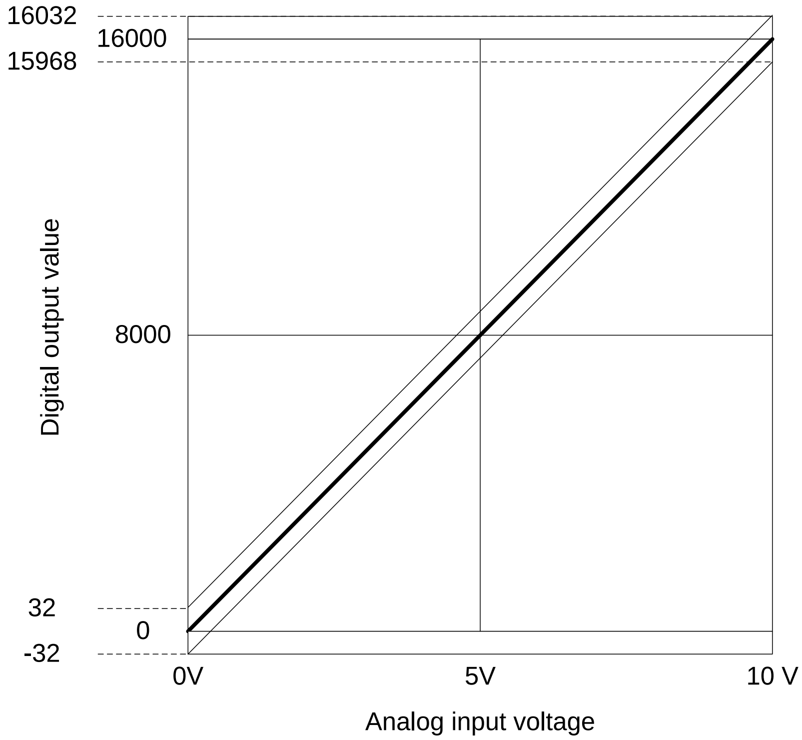
	Analog output voltage (V)						
	-10.24	**-10**	**-5**	**0**	**5**	**10**	**10.239**
Unsigned value (-192 to 16,191)	-192	0	4,000	8,000	12,000	16,000	16,191
Signed value (-8,192 to 8,191)	-8,192	-8,000	-4,000	0	4,000	8,000	8,191
Precise value (-10,240 to 10,239)	-10,240	-10,000	-5,000	0	5,000	10,000	10,239
Percentile value (-120 to 10,119)	-120	0	2,500	5,000	7,500	10,000	10,119

13.4 Accuracy

13.4.1 Input Accuracy

Accuracy of digital output value does not changed 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

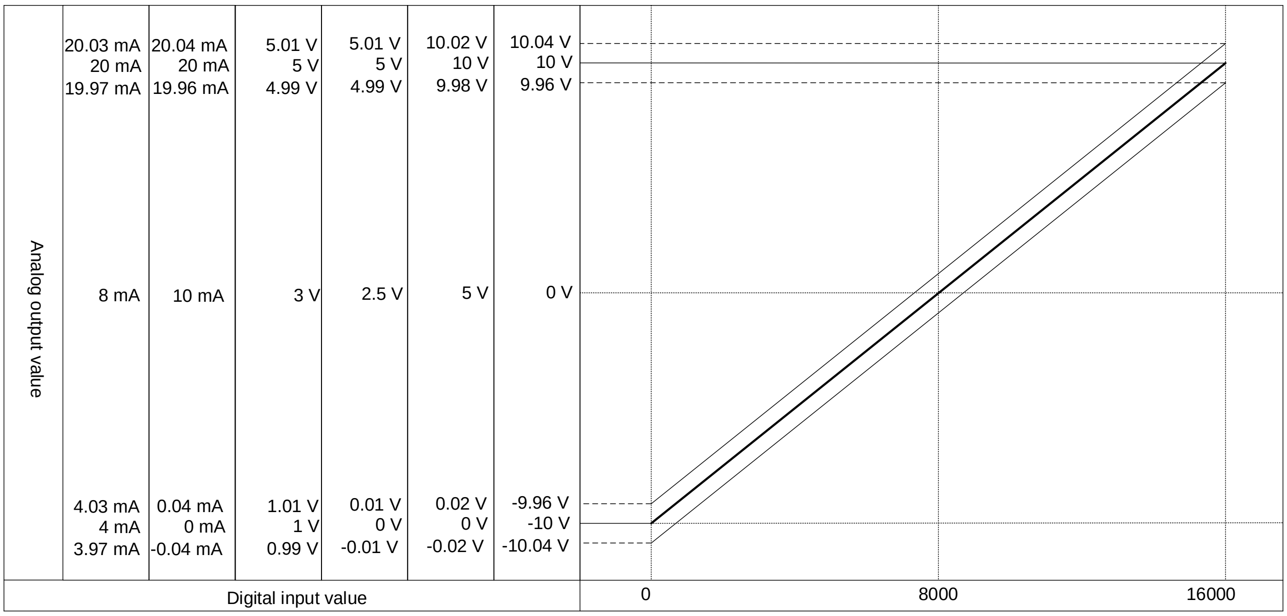
Therefore the range of the accuracy will become (8,000-32) to (8,000+32) = 7,968 to 8,032 when using 5 V input.

(2) Accuracy when using 10 V 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 when using 10 V input.

13.4.2 Output Accuracy

Accuracy of digital output value does not changed even if input range is changed. When digital input range is selected with unsigned value, accuracy is ±0.2% (Ambient temperature of 25 ±5℃).



(1) Accuracy when using -10 to 10 V 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	Description
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 format 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 whether disconnection of the input circuit, the alarm is displayed by a single flag.(Input signal range: 4 to 20 ㎃, 1 to 5 V) ● 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 (Previous, Min, Mid, Max value) are provided.
Interpolation method setting	● Set linear interpolation, S-type interpolation method.

13.5.1 Sampling Processing

It 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 ㎳

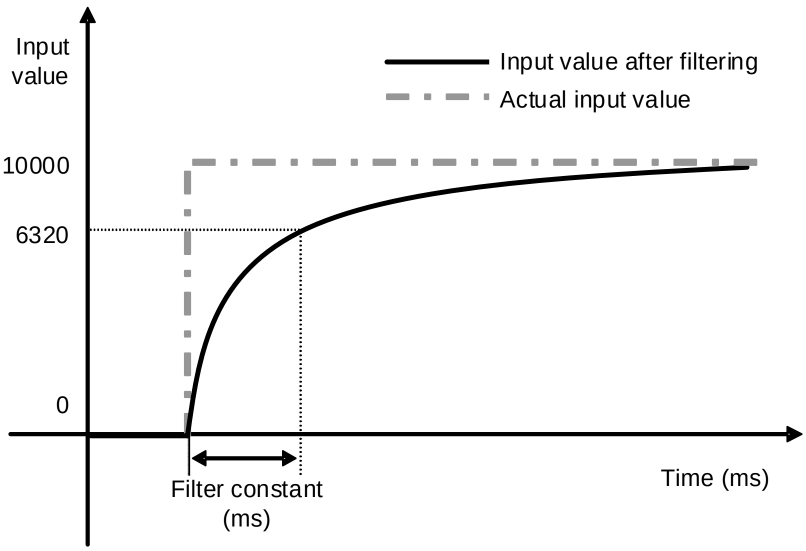
Sampling is to calculate the sampling value of continuous analog sign at a specific interval.

13.5.2 Filter Processing

Pre-filter input value and specified channel are calculated as below.

$$\text{Filtered Value} = \frac{(\text{Pre} - \text{Filtered Input Value} \times \text{Filter Constant}) + (\text{Current Input Value} \times 0.5\,ms \times \text{Number of used channels})}{\text{Filter Constant} + (0.5\,ms \times \text{Number of used channels})}$$

Setting range of Filter constant = 4 to 64,000 [ms]

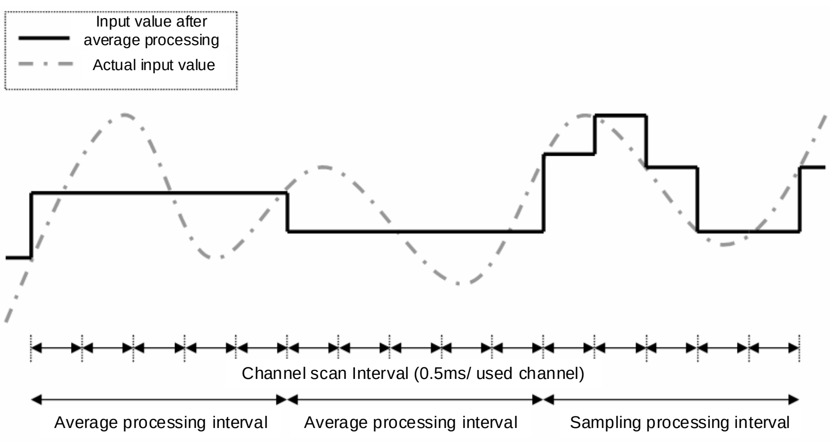


As the above graph, if the input value rapidly decreases from 0 to 10,000, the input value will be filtered. Specified time with filter constant is that the input value is the time to change by 63.2% of actual time constant.

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.

$$\text{Average processing count} = \frac{\text{Average time}}{\text{Number of used channels x 0.5 ms}}$$

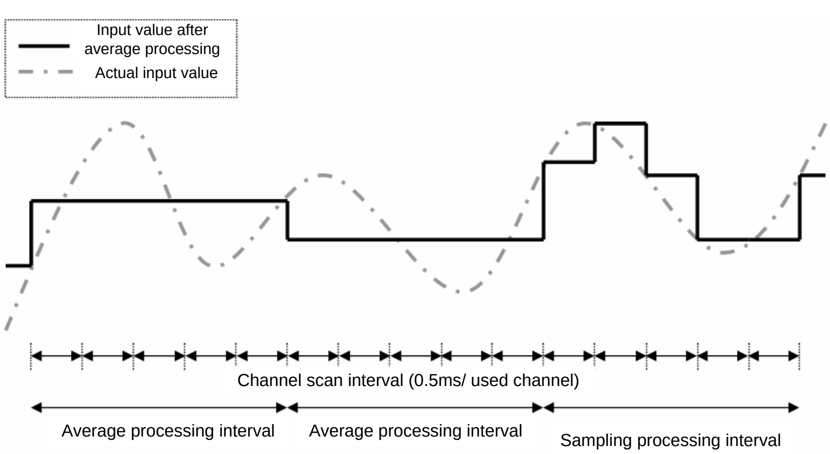
Time average is converted to count average in A/D conversion module internally, and then processed. In this case, remainder can be generated when dividing average time by (number of used channels X 0.5 ms). The remainder is rounded down.

(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



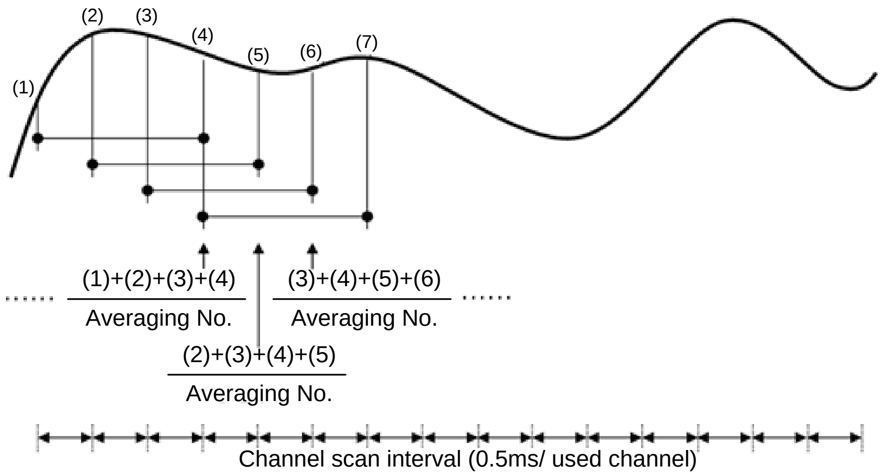
Setting range = 2 to 64,000 [times]

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, and its average is calculated and outputted in 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
	-	Scan 1	Scan 2	Scan 3	
No Setting	0	8000	8000	8000	Not process weighted average
1	0	7920	7999	7999	Apply 1% of former value
50	0	4000	6000	7000	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. And then the chosen average function is adapted. Finally, digital dat a is outputted. 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)

In case that Input voltage (1 to 5 VDC) or Input current (4 to 20 mADC) is chosen with analog input range, the analog input module has diagnostic function by checking disconnection and showing. 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 to 20 ㎃	0.8 ㎃ or less
1 to 5 V	0.2 V 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 operation	AD LED condition	Disconnection flag
Normal	Operation	On	Off
	Stop	On	Off
Input wiring is disconnected or Input is not connected.	Operation	**Flickering (1s intervals)**	**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	Description	Condition
%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.

(a) 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 to 16,191	-8,192 to 8,191	(2) Reference	-120 to 10,119
Function enabled	0 to 16,000	-8,000 to 8,000		0 to 10,000

(b) Digital output value depending on input range (Precise value)

Analog input range	Classification	Precise value
4 to 20 mA	Function disabled	3,808 to 2,191
	Function enabled	4,000 to 20,000
0 to 20 mA	Function disabled	-240 to 20,239
	Function enabled	0 to 20,000
1 to 5 V	Function disabled	952 to 5,047
	Function enabled	1,000 to 5,000
0 to 5 V	Function disabled	-60 to 5,059
	Function enabled	0 to 5,000
0 to 10 V	Function disabled	-120 to 10,119
	Function enabled	0 to 10,000
-10 to 10 V	Function disabled	-10,240 to 10,239
	Function enabled	-10,000 to 10,000

(2) Operation

When operating with 4 to 20 mA while being enabled this function, output value for input value change of the moment is as follows (Output data type: In case of 0 to 16,000).

Input current (mA)	12 mA	3 mA	4 mA	12 mA	21 mA	20 mA
Digital output value	8,000	8,000	0	8,000	8,000	16,000
Remarks	-	**Hold last value**	-	-	**Hold last value**	-

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

Detection condition for each input signal range is as follows.

Analog input range	Signal difference	Permission range	Lower limit	Upper limit
4 to 20 ㎃	16 ㎃	1.2%	3.808 ㎃	20.192 ㎃
0 to 20 ㎃	20 ㎃		-0.24 ㎃	20.24 ㎃
1 to 5 V	4 V		0.952 V	5.048 V
0 to 5 V	5 V		-0.06 V	5.06 V
0 to 10 V	10 V		-0.12 V	10.12 V
-10 to 10 V	20 V		-10.24 V	10.24 V

(2) Alarm indication for each channel

Alarm detection signal is shown on U01.08 and U01.09. If input signal returns to the within of effective range, alarm detection signal also returns to the normal status automatically.

(a) Upper limit alarm

Device Name	Device assignment	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	

(b) Lower limit alarm

Device Name	Device assignment	Description	Status description
_01_AD0_LOOR	%UX0.1.56	CH0 upper limit alarm	Off: Normal On: Maximum alarm occurrence
_01_AD1_LOOR	%UX0.1.57	CH1 upper 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 to 20 ㎃	On	3.808 ㎃ to 4 ㎃ 20 ㎃ to 20.192 ㎃	On	0
0 to 20 ㎃	On	-0.24 ㎃ to 0 ㎃ 20 ㎃ to 20.24 ㎃	On	0
1 to 5 V	On	0.952 V to 1 V 5 V to 5.048 V	On	0
0 to 5 V	On	-0.06 V to 0 V 5 V to 5.06 V	On	0
0 to 10 V	On	-0.12 V to 0 V 10 V to 10.12 V	On	0
-10 to 10 V	On	-10.24 V to -10 V 10 V to 10.24 V	On	0

13.5.7 Setting Function of Channel Output Status

Set the output against stop and abnormal condition of PLC.

(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.

(a) Previous value: The last output operated normally is retained.
(b) Min: The Min value of each range is outputted.
(c) Mid: The Mid value of each range is outputted.
(d) Max: The Max value of each range is outputted.

(3) Example
When the range of output channel is set by 4 to 20 mA and the output is 10 mA, and then if the system is changed from 'Run' to 'Stop', the output will be as follows depending on setting data of channel output status.

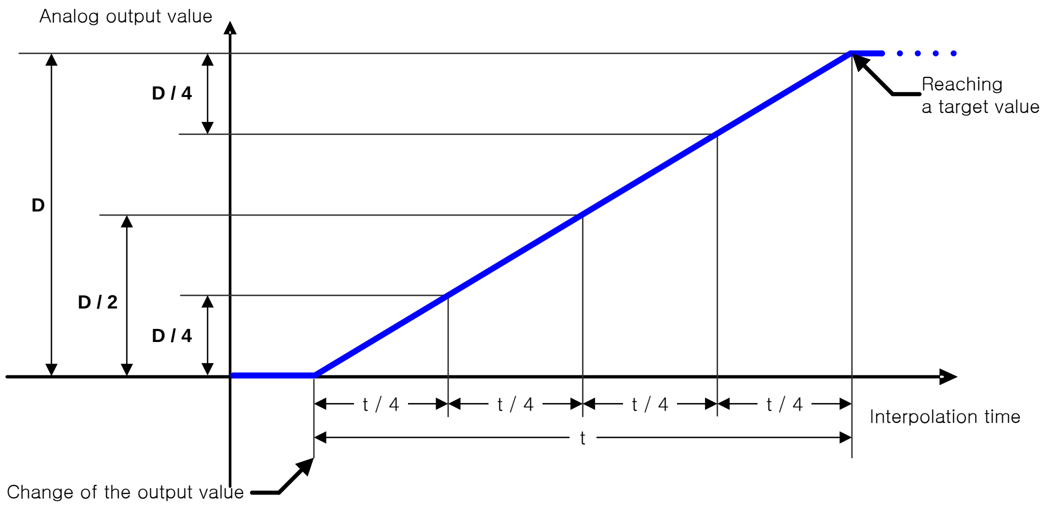
(a) Previous value: 10 mA which is previous output value is retained.
(b) Min value: 4 mA which is min value of relevant range is outputted.
(c) Mid value: 12 mA which is mid value of relevant range is outputted
(d) Max value: 20 mA which is max value of relevant range is outputted.

13.5.8 Interpolation Method Setting

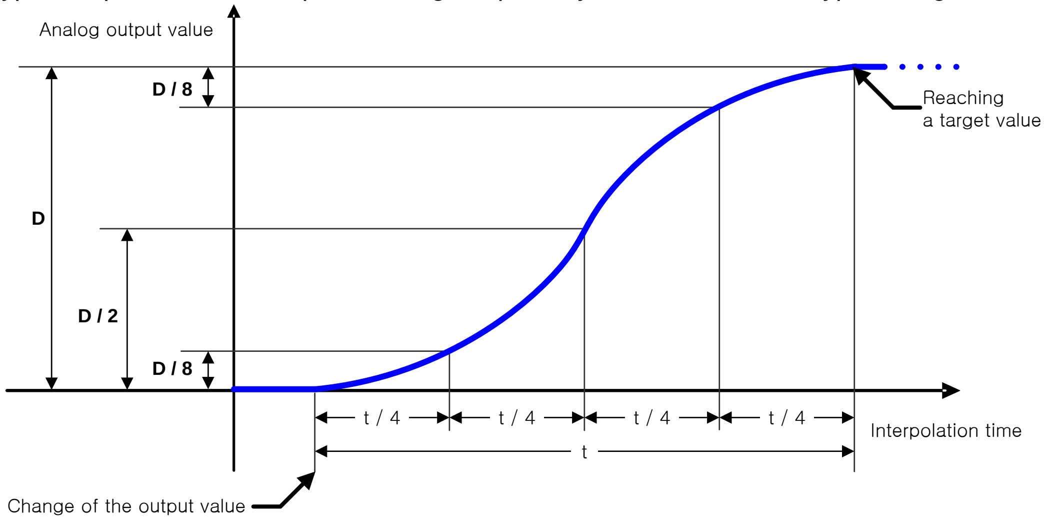
(1) Functions
The output signal of module is used in order to execute interpolation output depending on set interpolation time. When the voltage and current is outputted, it can be used to prevent transient response of load system as a suddenly changed output.

(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	Address of interpolation output value	Details
_01_DA0_INTPVAL	%UW0.1.25	Channel 0 interpolation operation value
_01_DA1_INTPVAL	%UW0.1.26	Channel 1 interpolation operation value

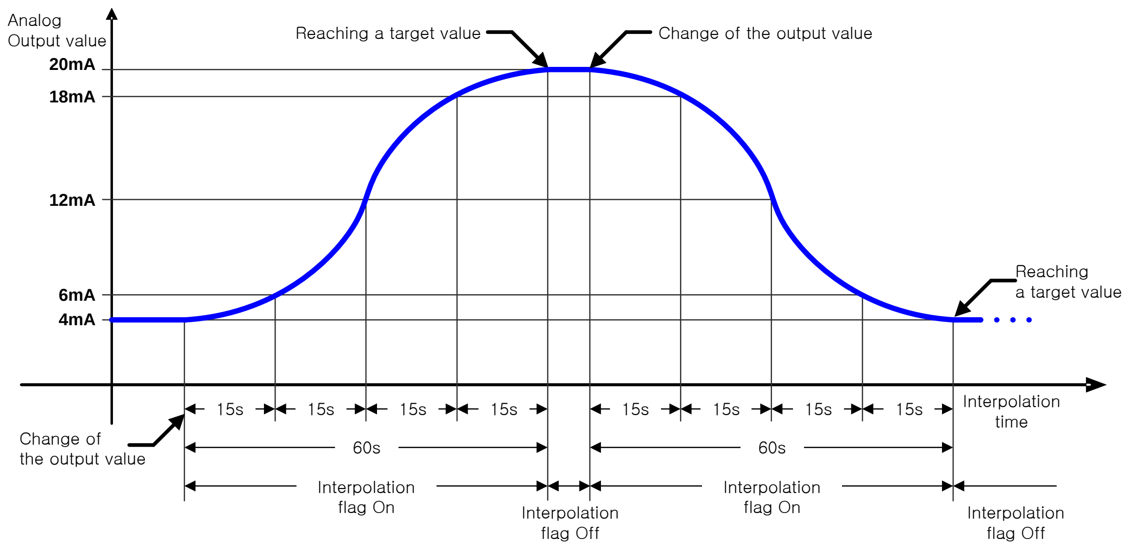
(5) Interpolation flag turns on while the interpolation is outputted. And when the interpolation output value is reached at objective value, it will turn off.

Variable Name	Interpolation flag	Details
_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.

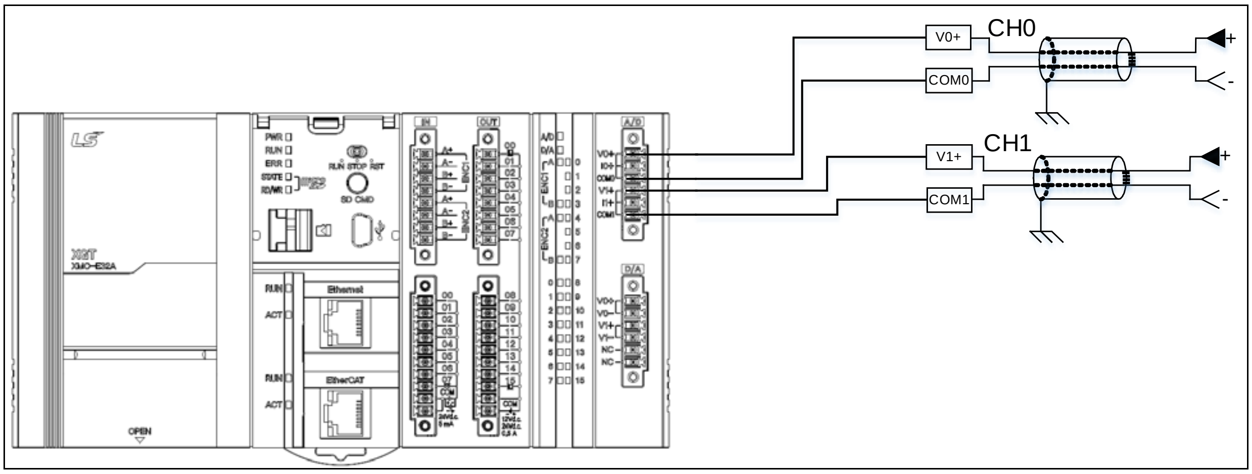
13.6 Wiring

13.6.1 Example for Wiring Analog Input

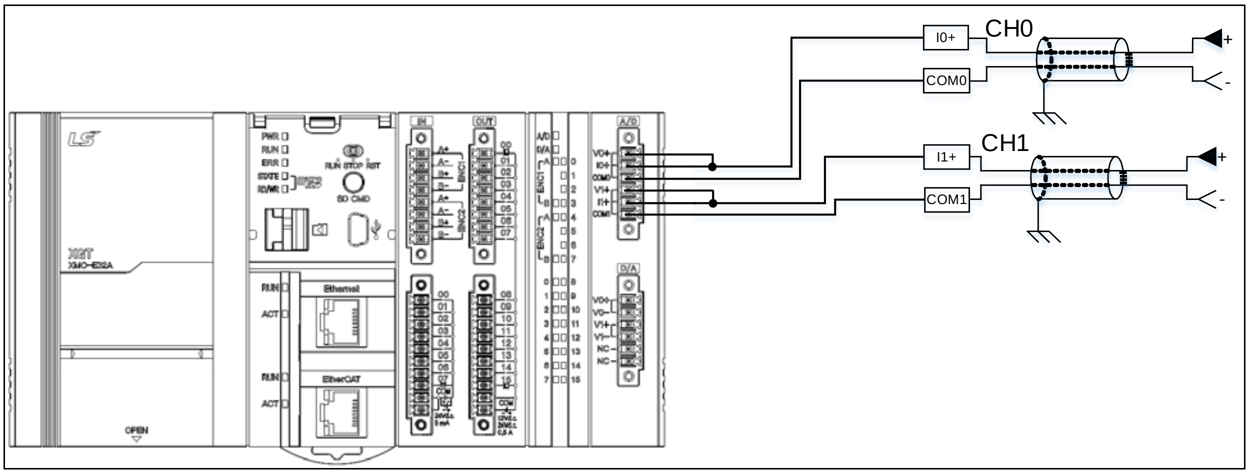
(1) The input resistance of current input circuit is 250 Ω (typ.).
(2) The input resistance of voltage input circuit is 1 MΩ or more.
(3) Set the operation mode only if you want to use channels.
(4) Example for analog input wiring

When inputting the voltage, relevant channel V+ and COM terminal is used. When inputting the current, relevant channel V+ and COM terminal is used after connecting between V+ and I+ terminal.

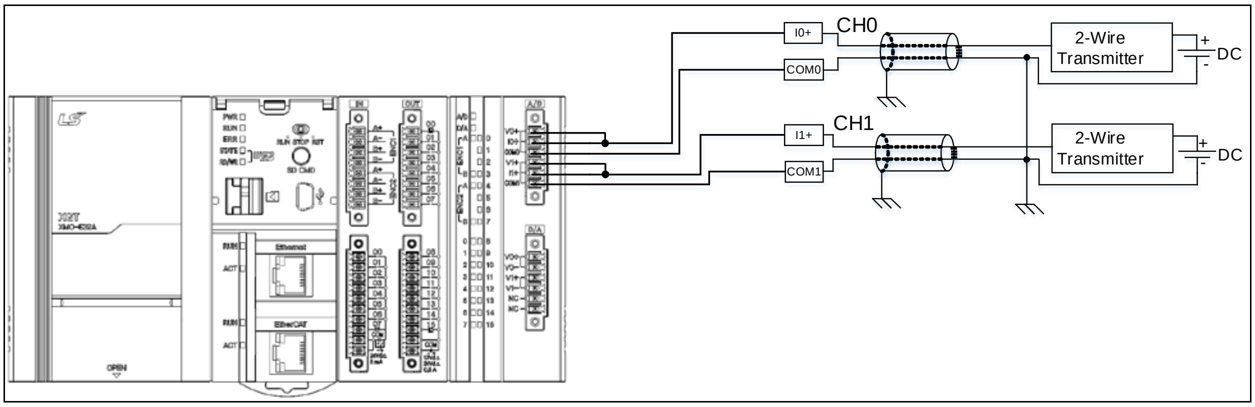
(a) Voltage wiring



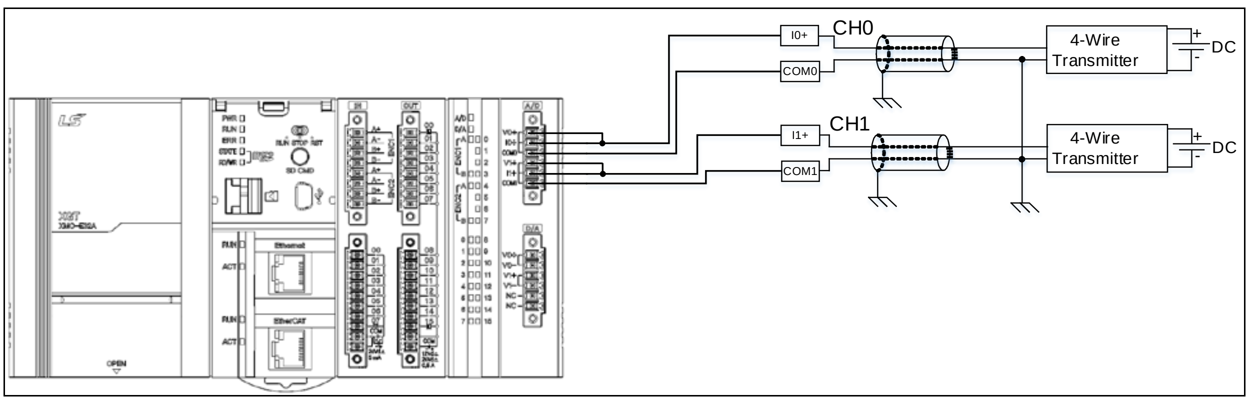
(b) Current wiring



(5) The example of analog input 2-Wire sensor/transmitter wiring (The current input)
Use the I+ and COM terminal after connecting V+ with I+ terminal.

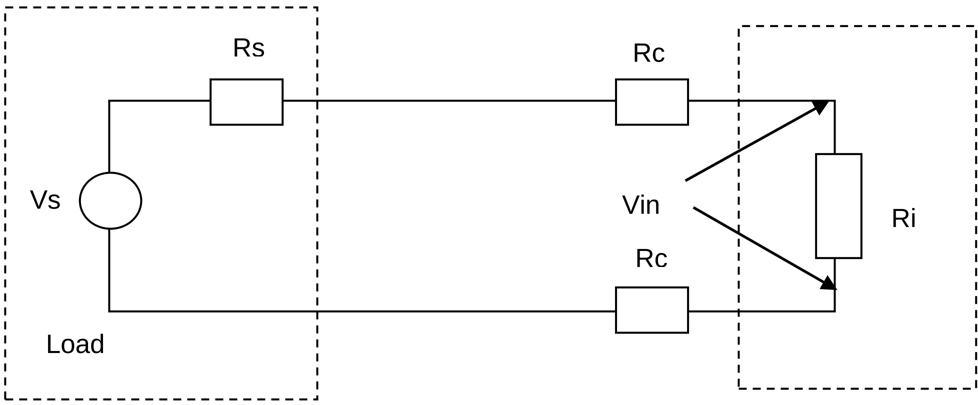


(6) The example of analog input 4-Wire sensor/transmitter wiring (The current input)
Use the I+ and COM terminal after connecting V+ with I+ terminal.



(7) Relationship between voltage input accuracy and wiring length

In voltage input, the wiring (cable) length between transmitter or sensor and module has an effect on digital-converted values of the module as specified below;



Analog input(Voltage)

Where,

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

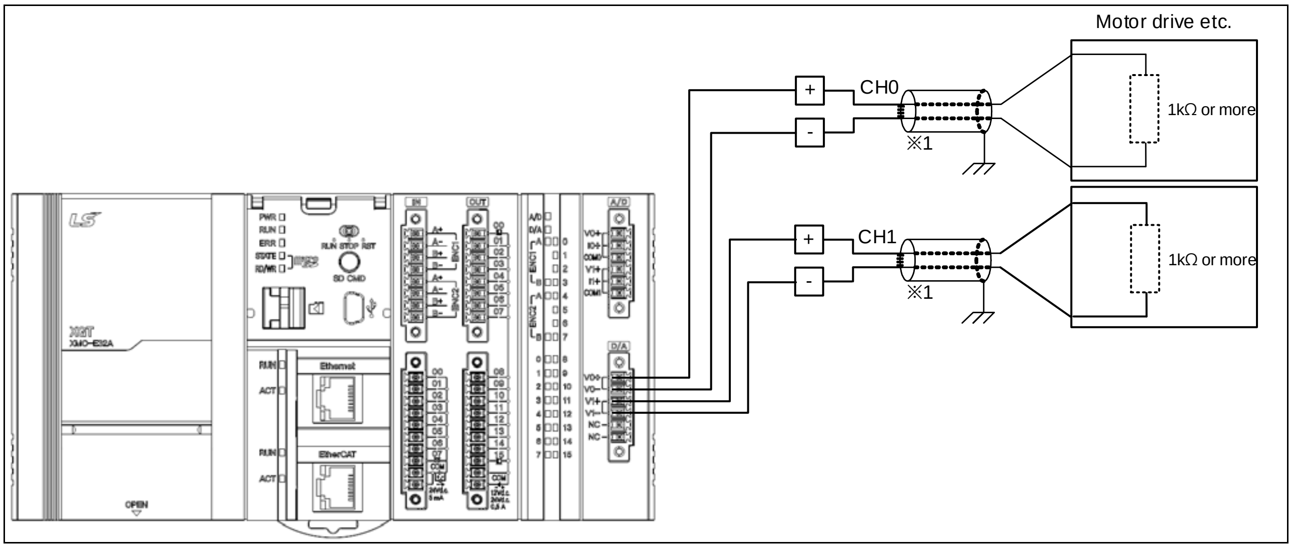
$$Vin = \frac{Ri \times Vs}{[Rs + (2 \times Rc) + Ri]}$$

$$\%Vi = \left(1 - \frac{Vin}{Vs}\right) \times 100\%$$

Notes
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].

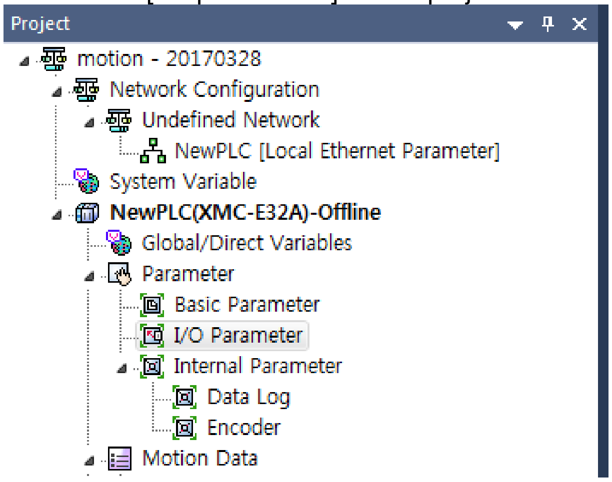
(1) Settings

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.

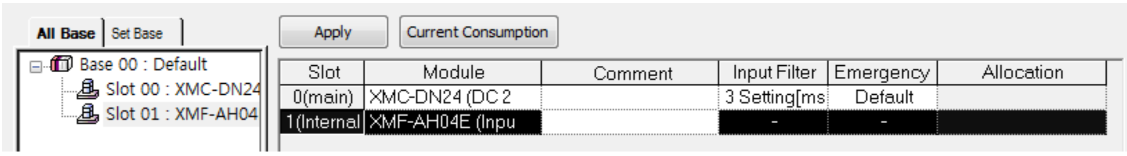
Item	Details
[I/O parameter]	(a) Input parameter setting Specify the following setting items necessary for the module operation. 1) Channel Enable/Disable setting 2) Input voltage (current) range 3) Output data format setting 4) Filter constant setting 5) Average processing method setting 6) Average value setting 7) Hold last value setting (b) Output parameter setting Specify the following setting items necessary for the module operation. 1) Channel Enable/Disable setting 2) Output (voltage· current) range 3) Input data format setting 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

(a) Run XG5000 to create a project.
(Refer to XG5000 program manual for details on how to create the project)
(b) Double-click [I/O parameters] on the project window.

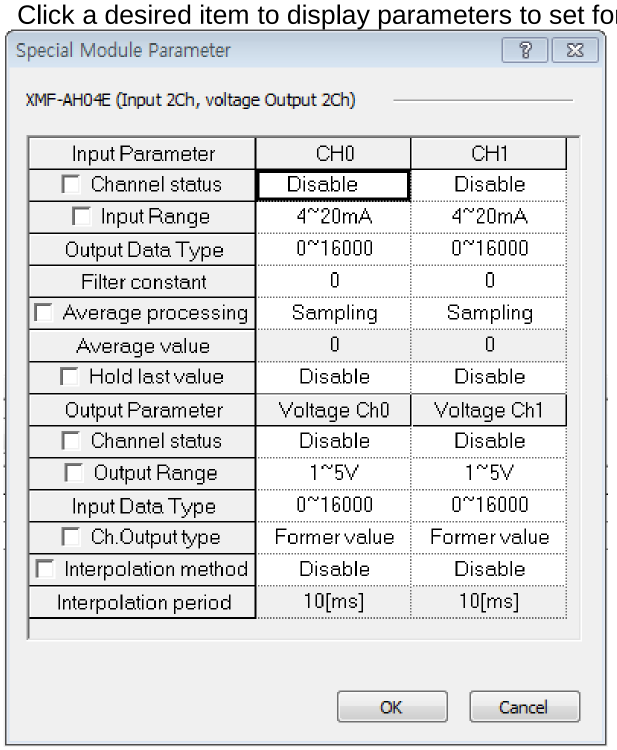


(c) [I/O Parameter setting] On the ‘I/O Parameter setting’ screen, find and clink the slot 1 (internal) which has embedded function.



(d) Click the arrow button on the screen above to display the screen where an applicable module can be selected. Search and select the embedded analog input/output module to select.

(e) 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.



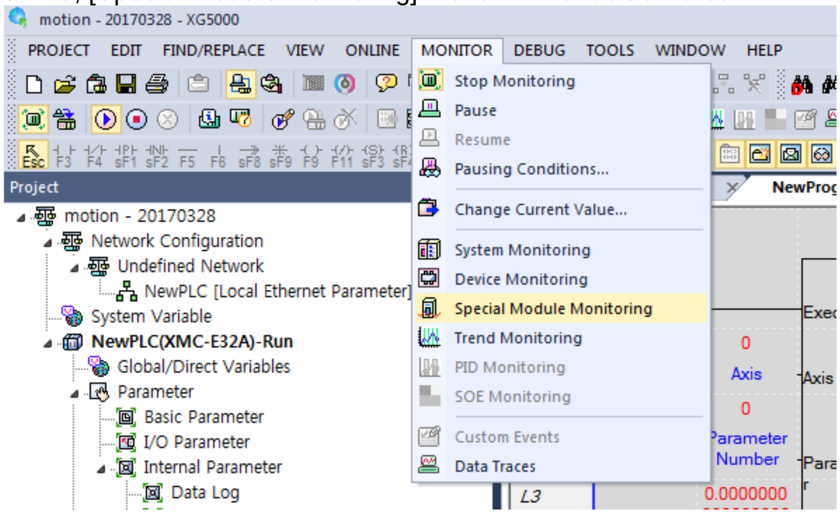
Input Parameter	CH0	CH1
Channel status	Disable	Disable
Input Range	4~20mA	4~20mA
Output Data Type	0~16000	0~16000
Filter constant	0	0
Average processing	Sampling	Sampling
Average value	0	0
Hold last value	Disable	Disable
Output Parameter	Voltage Ch0	Voltage Ch1
Channel status	Disable	Disable
Output Range	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]

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.

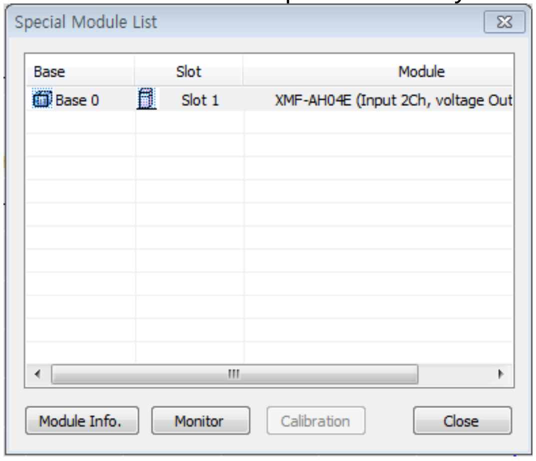


Notes

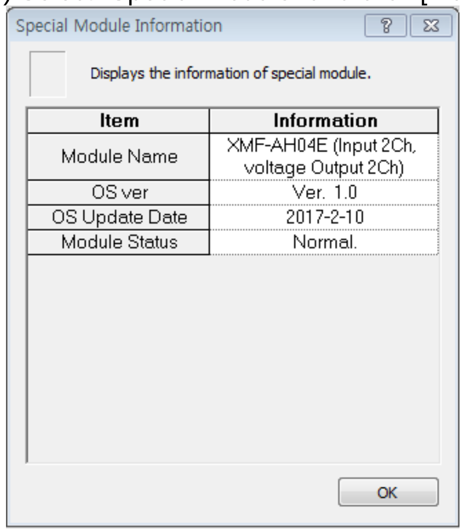
1. The screen may not normally be shown due to the lack of system resource. In this case, terminate all applications and try to start XG5000 again.
2. I/O parameter set in status of [Special Module Monitor] is temporally set to implement the test.
   So, If status of [Special Module Monitor] is ended, I/O parameter which is set becomes extinct.
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

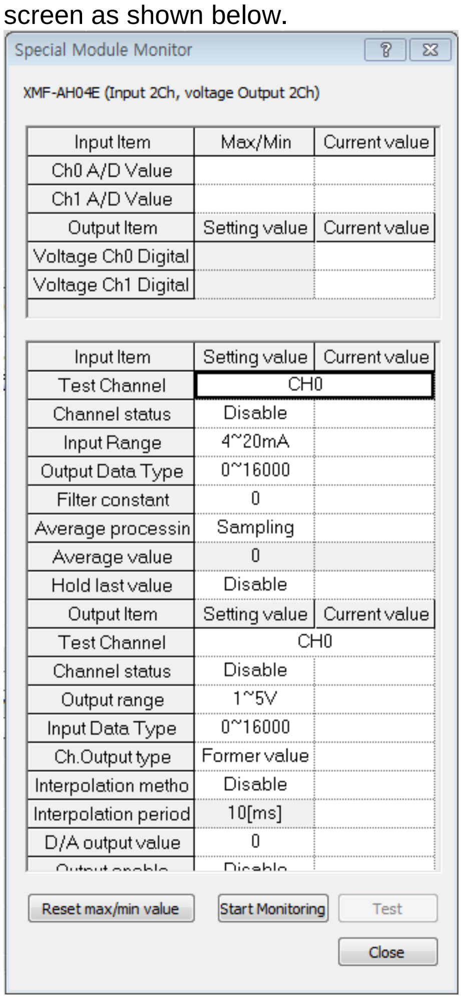
(a) With XG5000 connected to PLC CPU (on-line status), click [Monitor] -> [Special Module Monitoring] to display 'Special Module Select' screen as below showing base/slot information in addition to special module type. The module installed on the present PLC system will be displayed on the list dialog box.



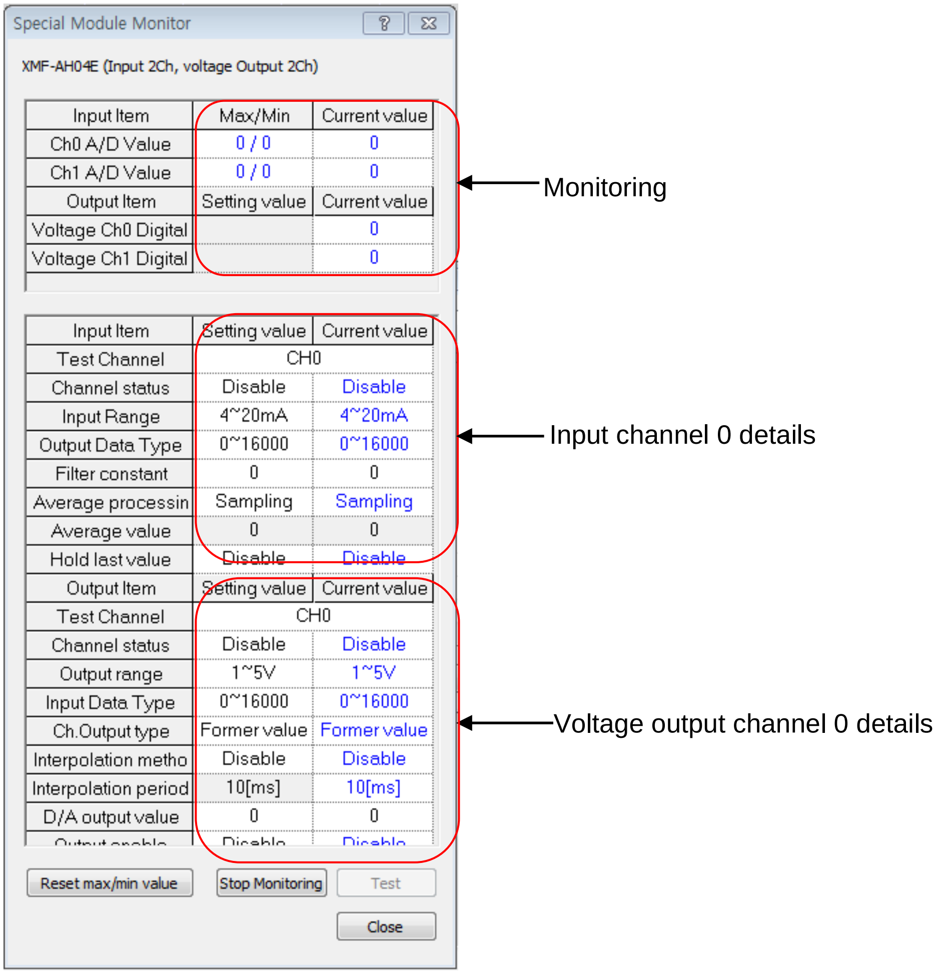
(b) Select “Special Module” and click [Module information] to display the information as shown below.



(c) Click [Monitor] on the “Special Module” screen in [Special Module List] to display [Special Module Monitoring] screen as shown below.

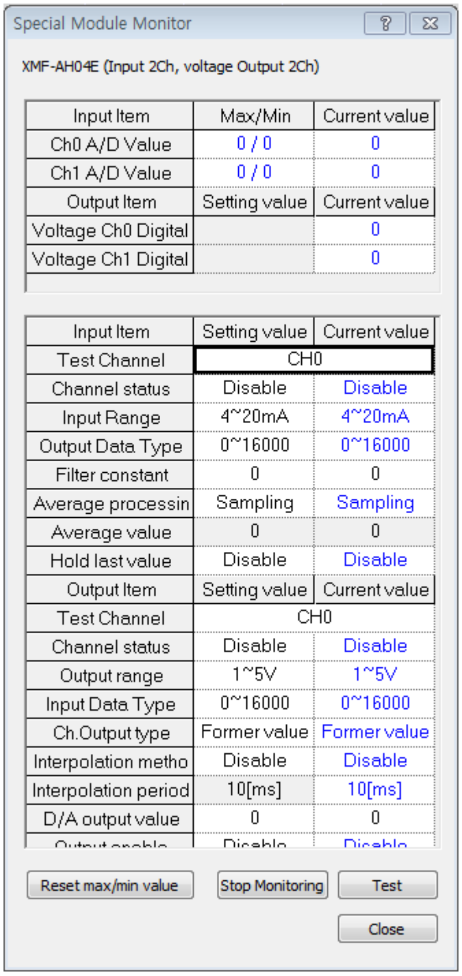


(d) Start Monitoring: Click [Start Monitoring] to show digital input / output data of current operated channel.



Execution screen of [Start Monitoring]

(e) 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.



Input Item	Max/Min	Current value
Ch0 A/D Value	0 / 0	0
Ch1 A/D Value	0 / 0	0
Output Item	Setting value	Current value
Voltage Ch0 Digital		0
Voltage Ch1 Digital		0

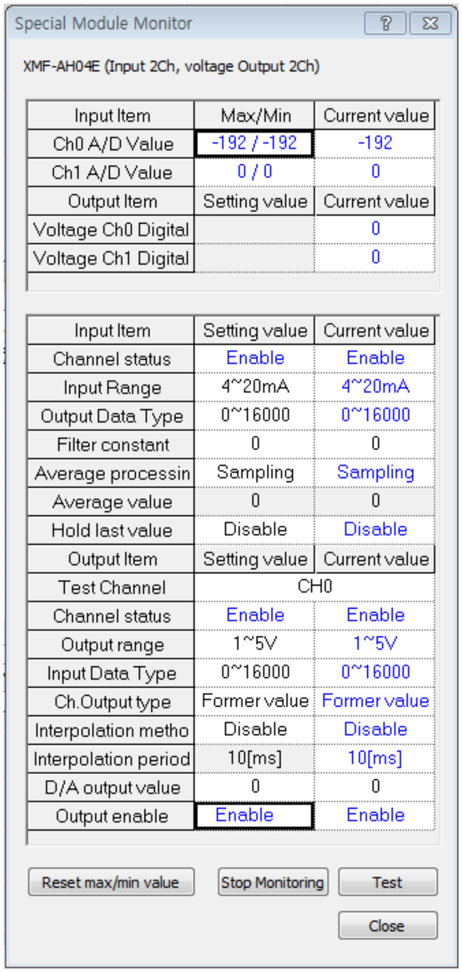
Input Item	Setting value	Current value
Test Channel	CH0	
Channel status	Disable	Disable
Input Range	4~20mA	4~20mA
Output Data Type	0~16000	0~16000
Filter constant	0	0
Average processin	Sampling	Sampling
Average value	0	0
Hold last value	Disable	Disable
Output Item	Setting value	Current value
Test Channel	CH0	
Channel status	Disable	Disable
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

Execution screen of [Test]

(f) 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.



[Max/Min Value Monitor] execution screen

(g) Close

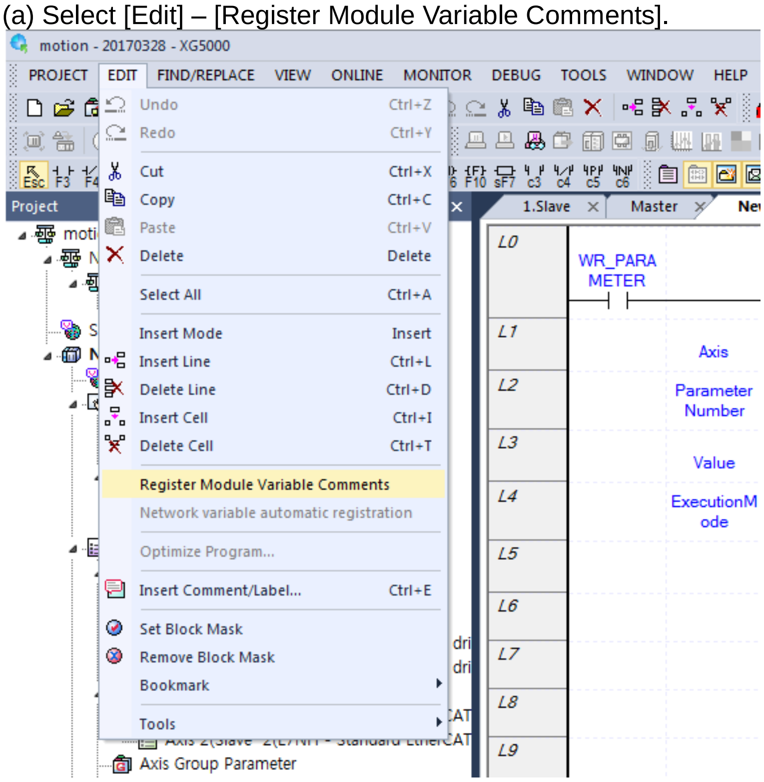
[Close]: [Close] is used to escape from the monitoring/test screen. When the monitoring/test screen is closed, the max value, the min. value and the present value will not be saved any more.

13.9 Automatic Register U Devices

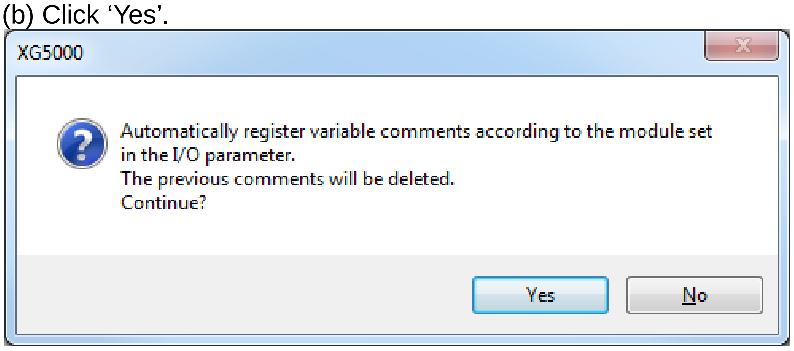
Register the variables for each module referring to the special module information that is set in the I/O parameter. The user can modify the variables and comments.

(1) Procedure

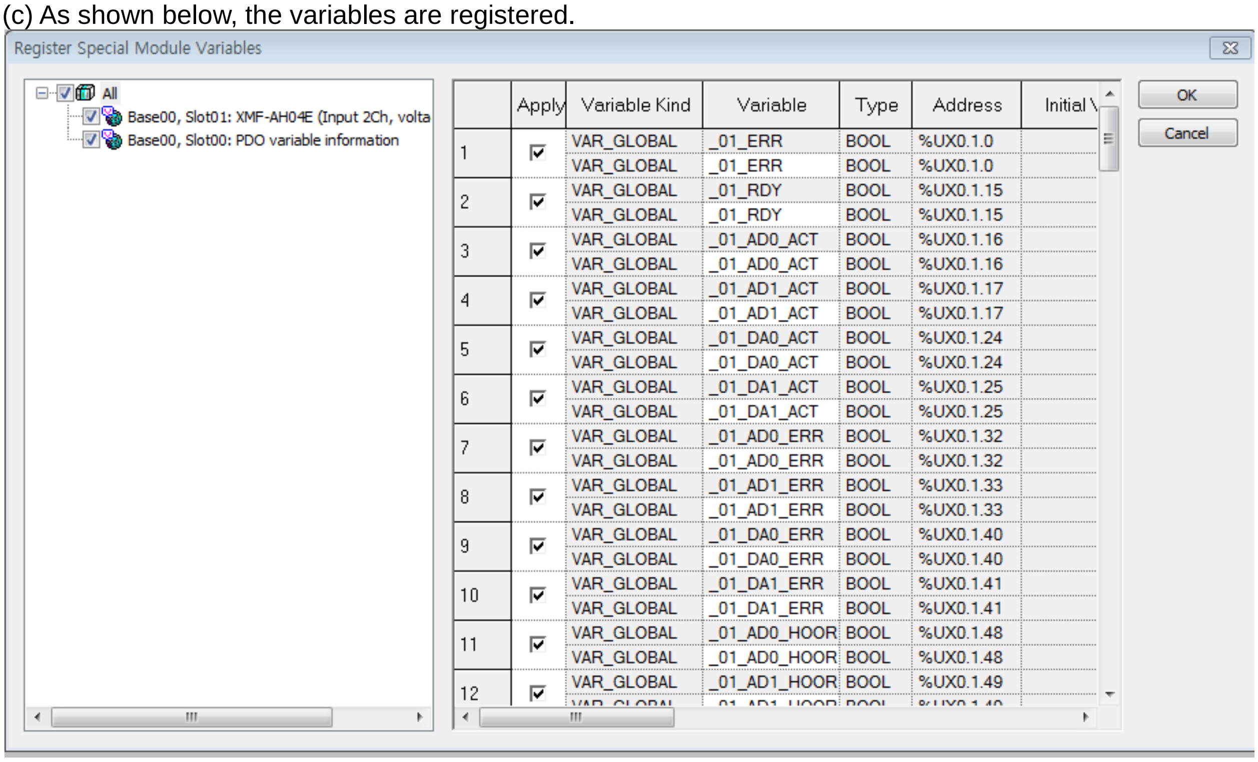
(a) Select [Edit] – [Register Module Variable Comments].



(b) Click 'Yes'.

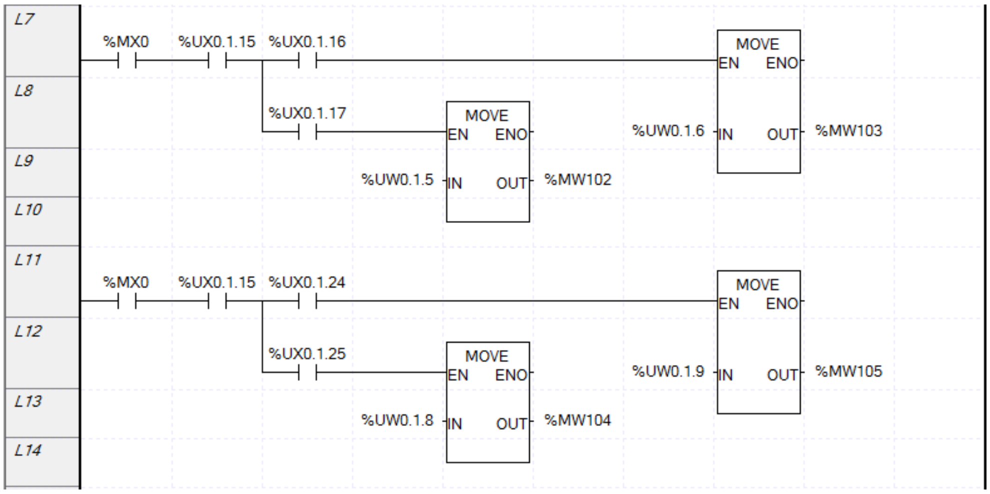


(c) As shown below, the variables are registered.

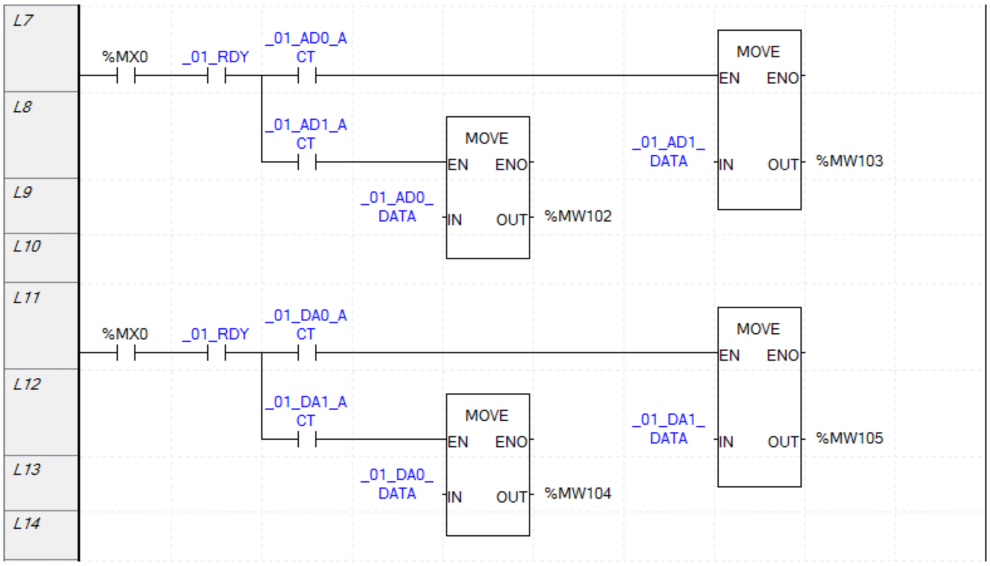


(2) Save variables
(a) The contents of 'View Variable' can be saved as a text file.
(b) Select [Edit] → [Export to File].
(c) The contents of 'View variable' are saved as a text file.

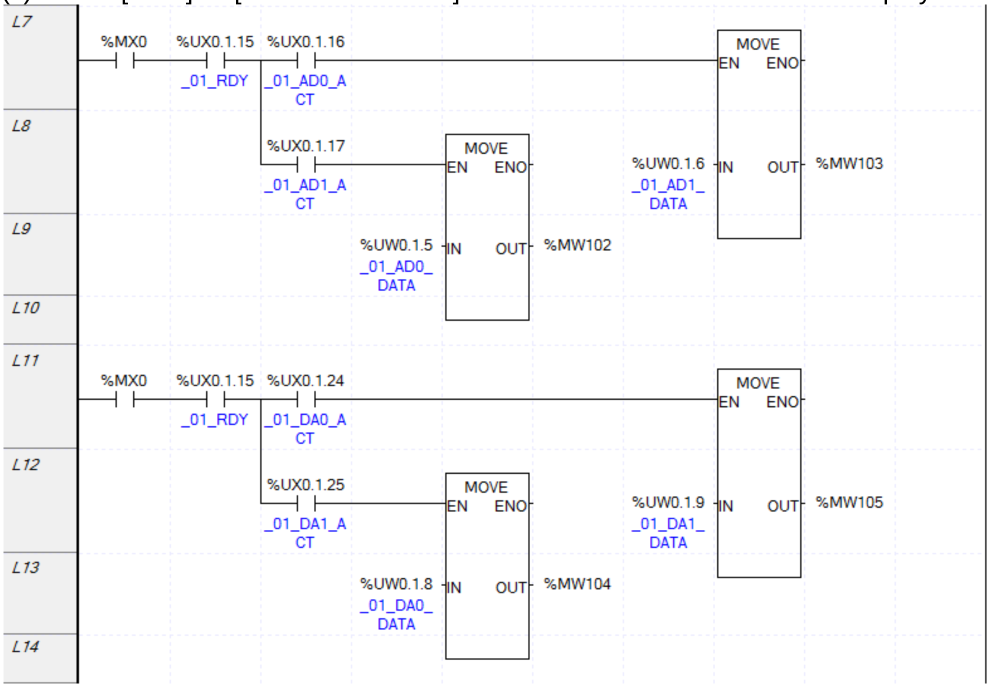
(3) View variables in program
(a) The example program of XG5000 is as shown below.



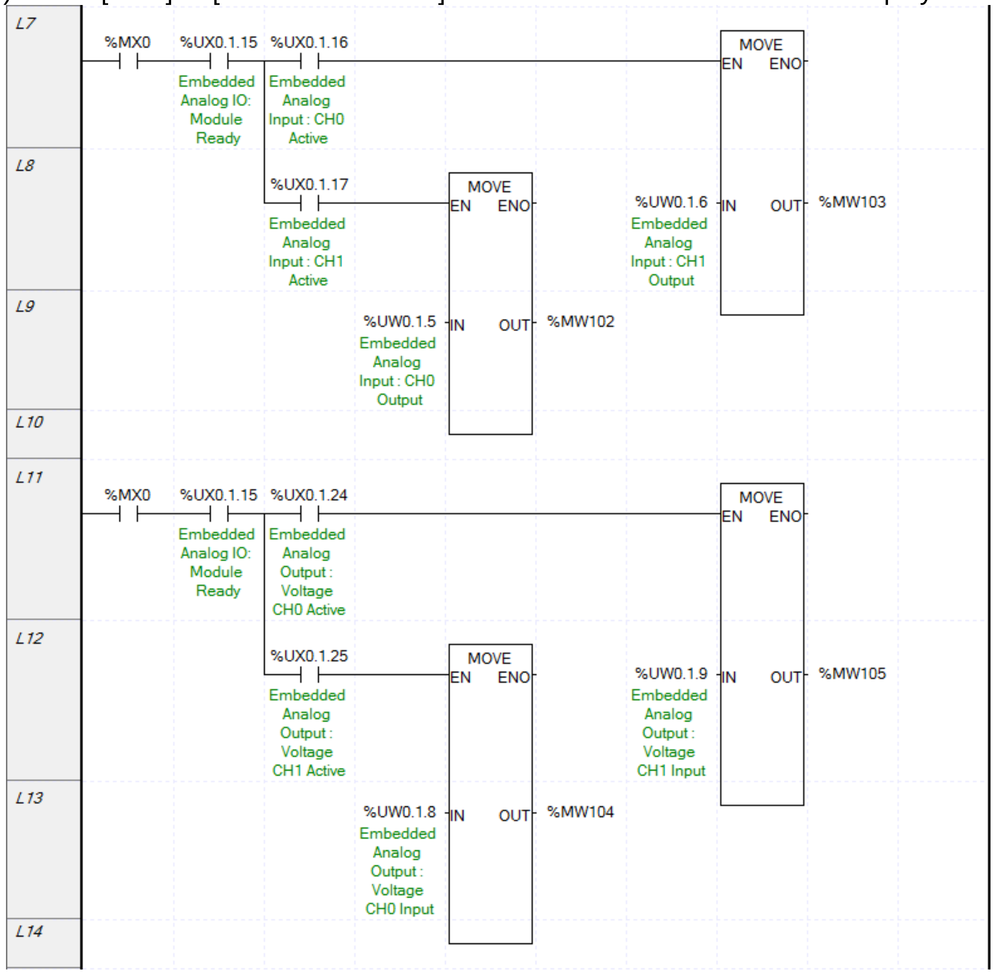
(b) Select [View] → [Variables]. The devices are changed into variables.



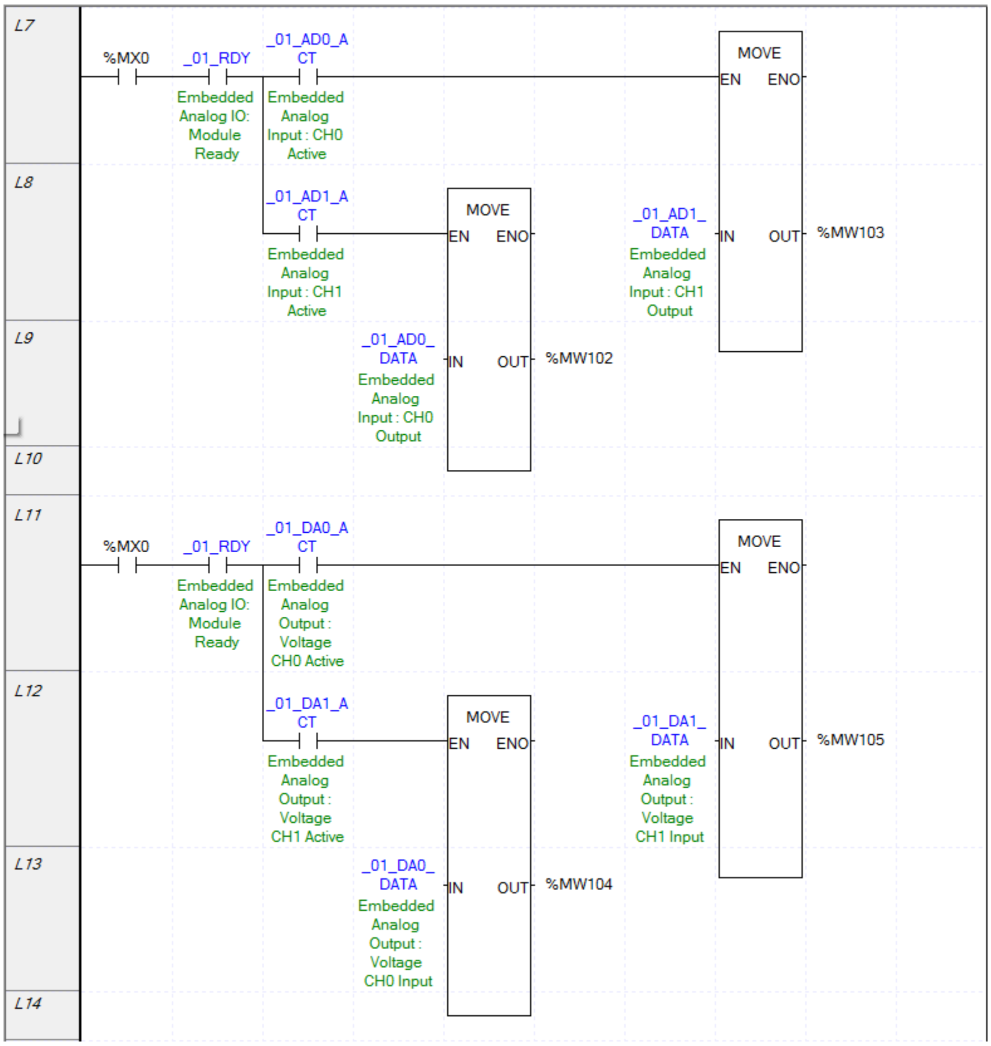
(c) Select [View] → [Devices/Variables]. Devices and variables are both displayed.



(d) Select [View] -> [Device/Comments]. Devices and comments are both displayed.



(e) Select [View] → [Variables/Comments]. Variables and comments are both displayed.


L7
%MX0
_01_RDY
Embedded Analog IO: Module Ready
_01_AD0_ACT
Embedded Analog Input : CH0 Active
MOVE
EN ENO
L8
_01_AD1_ACT
Embedded Analog Input : CH1 Active
MOVE
EN ENO
_01_AD1_DATA
Embedded Analog Input : CH1 Output
IN OUT
%MW103
L9
_01_AD0_DATA
Embedded Analog Input : CH0 Output
IN OUT
%MW102
L10
L11
%MX0
_01_RDY
Embedded Analog IO: Module Ready
_01_DA0_ACT
Embedded Analog Output : Voltage CH0 Active
MOVE
EN ENO
L12
_01_DA1_ACT
Embedded Analog Output : Voltage CH1 Active
MOVE
EN ENO
_01_DA1_DATA
Embedded Analog Output : Voltage CH1 Input
IN OUT
%MW105
L13
_01_DA0_DATA
Embedded Analog Output : Voltage CH0 Input
IN OUT
%MW104
L14

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	Device	Comment
_01_AD0_ACT	BOOL	%UX0.1.16	Channel 0 Active
_01_AD0_AVGTYPE	BYTE	%UB0.1.34	Channel 0 Average type
_01_AD0_AVGVAL	WORD	%UW0.1.18	Channel 0 Average value
_01_AD0_DATA	WORD	%UW0.1.5	Channel 0 Output data
_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 Alarm (Upper Limit)
_01_AD0_IDD	BOOL	%UX0.1.72	Channel 0 Input disconnection flag
_01_AD0_LOOR	BOOL	%UX0.1.56	Channel 0 Alarm (Lower Limit)
_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 setting
_01_AD1_DATA	WORD	%UW0.1.6	Channel 1 Output data
_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 Alarm (Upper Limit)
_01_AD1_IDD	BOOL	%UX0.1.73	Channel 1 Input disconnection flag
_01_AD1_LOOR	BOOL	%UX0.1.57	Channel 1 Alarm (Lower Limit)
_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 per channel (Array)
_01_AD_AVGTYPE_ARY	ARRAY[0..1] OF BYTE	%UB0.1.32	Average type per channel (Array)
_01_AD_AVGVAL_ARY	ARRAY[0..1] OF WORD	%UW0.1.18	Average value per channel (Array)
_01_AD_DATATYPE_ARY	ARRAY[0..1] OF BYTE	%UB0.1.26	data type setting per channel (Array)
_01_AD_ERR_ARY	ARRAY[0..1] OF BOOL	%UX0.1.32	Error per channel (Array)
_01_AD_FILTCONST_ARY	ARRAY[0..1] OF WORD	%UW0.1.15	Filter constant per channel (Array)
_01_AD_HOLDVAL_ARY	ARRAY[0..1] OF BOOL	%UX0.1.320	Hold effective conversion value per channel (Array) setting
_01_AD_HOOR_ARY	ARRAY[0..1] OF BOOL	%UX0.1.48	Alarm (Upper Limit) per channel (Array)
_01_AD_IDD_ARY	ARRAY[0..1] OF BOOL	%UX0.1.72	Input Disconnection Flag per channel (Array)
_01_AD_LOOR_ARY	ARRAY[0..1] OF BOOL	%UX0.1.56	Alarm (Lower Limit) per channel (Array)
_01_AD_RANGE_ARY	ARRAY[0..1] OF BYTE	%UB0.1.22	Range setting per channel (Array)
_01_AD_RUN_ARY	ARRAY[0..1] OF BOOL	%UX0.1.160	Operation setting per channel (Array)

Built-in analog output

Variable name	Type	Device assigned	Comment
_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(Voltage) Input data type
_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(Voltage) Output status setting
_01_DA0_RANGE	BYTE	%UB0.1.24	Channel 0(Voltage) Range setting
_01_DA0_RUN	BOOL	%UX0.1.168	Channel 0(Voltage) Operation setting
_01_DA1_ACT	BOOL	%UX0.1.25	Channel 1(Voltage) Active
_01_DA1_DATA	WORD	%UW0.1.9	Channel 1(Voltage) Input
_01_DA1_DATATYPE	BYTE	%UB0.1.29	Channel 1(Voltage) Input 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(Voltage) Output status setting
_01_DA1_RANGE	BYTE	%UB0.1.25	Channel 1(Voltage) Range setting
_01_DA1_RUN	BOOL	%UX0.1.169	Channel 1(Voltage) Operation setting
_01_DA_ACT_ARY	ARRAY[0..1] OF BOOL	%UX0.1.24	Active stats per channel (Array)
_01_DA_DATATYPE_ARY	ARRAY[0..1] OF BYTE	%UB0.1.28	Input data type per channel (Array)
_01_DA_DATA_ARY	ARRAY[0..1] OF WORD	%UW0.1.8	Input data per channel (Array)
_01_DA_ERR_ARY	ARRAY[0..1] OF BOOL	%UX0.1.40	Error per channel (Array)
_01_DA_INTPMTHD_ARY	ARRAY[0..1] OF BYTE	%UB0.1.46	Interpolation method setting per channel (Array)
_01_DA_INTPTIME_ARY	ARRAY[0..1] OF BYTE	%UB0.1.48	Interpolation time setting per channel (Array)
_01_DA_INTPVAL_ARY	ARRAY[0..1] OF WORD	%UW0.1.25	Interpolation value per channel (Array)
_01_DA_INTP_ARY	ARRAY[0..1] OF BOOL	%UX0.1.64	Interpolation enabled per channel (Array)
_01_DA_OUTEN_ARY	ARRAY[0..1] OF BOOL	%UX0.1.112	Output enable setting per channel (Array)
_01_DA_OUTSTAT_ARY	ARRAY[0..1] OF WORD	%UW0.1.21	Output status setting per channel (Array)
_01_DA_RANGE_ARY	ARRAY[0..1] OF BYTE	%UB0.1.24	Range setting per channel (Array)
_01_DA_RUN_ARY	ARRAY[0..1] OF BOOL	%UX0.1.168	Operation setting per channel (Array)

Built-in analog common

Variable name	Type	Device	Comment
_01_ERR	BOOL	%UX0.1.0	Motion controller error
_01_RDY	BOOL	%UX0.1.15	Motion controller ready
_01_SETTINGERR	WORD	%UW0.1.27	Setting error information

(1) Built-in analog module ready/error flag (_01_RDY/_01_ERR)

(a) %UX0.1.15 : It will be ON when module is powered or reset with D/A conversion ready to process A/D conversion.
(b) %UX0.1.0 : It is a flag to display the error status of built-in analog module

	Bit15	Bit4	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
%UW0.1.0	Ready	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Error

Module ready
Bit On (1) : Normal
Bit Off (0) : Error

Module error
Bit On (1) : Error
Bit Off (0) : Normal

(2) Channel active information

(a) This area shows the channel being used.
(b) _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_DA1_ACT(%UX0.1.25) : Output channel 1 active

	Bit15	Bit4	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
%UW0.1.1	-	-	-	-	-	-	Output channel 0	Output channel 1	-	-	-	-	-	-	Input channel 0	Input channel 1

Active channel information
Bit On (1) : Active
Bit Off (0) : Inactive

Active channel information
Bit On (1) : Active
Bit Off (0) : Inactive

(3) Channel error information

(a) This area shows the channel error status.
(b) _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_DA1_ERR(%UX0.1.41) : Output channel 1 error

	Bit15	Bit4	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
%UW0.1.2	-	-	-	-	-	-	Output channel 0	Output channel 1	-	-	-	-	-	-	Input channel 0	Input channel 1

Channel error information
Bit On (1) : Error
Bit Off (0) : Normal

Channel error information
Bit On (1) : Error
Bit Off (0) : Normal

(4) Input alarm (upper/lower) flag

(a) This area shows upper/lower alarm per channel status.
(b) _01_AD0_HOOR(%UX0.1.48) : Input channel 0 upper alarm
_01_AD1_HOOR(%UX0.1.49) : Input channel 1 upper alarm
_01_AD0_LOOR(%UX0.1.56) : Input channel 0 lower alarm
_01_AD1_LOOR(%UX0.1.57) : Input channel 1 lower 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) Input disconnection / interpolation output status

(a) This area shows the channel detecting input disconnection and being outputting interpolation.
(b) _01_DA0_INTP(%UX0.1.64) : Output channel 0 outputting interpolation
_01_DA1_INTP(%UX0.1.65) : Output channel 1 outputting interpolation.
_01_AD0_IDD(%UX0.1.72) : Detecting Input channel 0 disconnection
_01_AD1_IDD(%UX0.1.73) : Detecting Input channel 1 disconnection

	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.

(a) This area shows converted(A/D) digital output value by channel in buffer memory (%UW0.1.5 ~%UW0.1.6)
(b) Digital output value is stored in 16-bit binary number.
(c) _01_AD0_DATA(%UW0.1.5) : Input channel 0 conversion value.
_01_AD1_DATA(%UW0.1.6) : Input channel 1 conversion value.

	Bit15 … Bit0
%UW0.1.5	Input channel 0 conversion value
%UW0.1.6	Input channel 1 conversion value

(7) Output permission setting

(a) The output enable / disable for each channel can be set.
(b) When the output permission is not set, the output of all channels will be prohibited
(c) _01_DA0_OUTEN(%UX0.1.112) : Output channel 0 output enable
_01_DA1_OUTEN(%UX0.1.113) : Output channel 1 output enable

	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) Digital input value

(a) 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.
(b) If the digital input value is not set, it will be handled as '0'.

(c) _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) Operating channel setting.

(a) If the operating channel is not set, overall channel status is STOP.
(b) _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

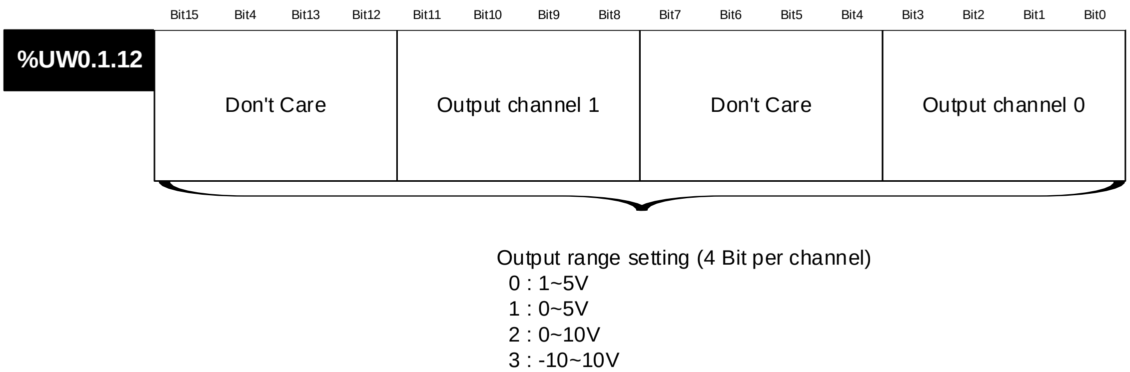
(a) The ranges of analog input voltage are DC 1~5V, DC 0~5V, DC 0~10V, DC -10~10V, the ranges of analog current input are DC 4~20mA, DC 0~20mA.
(b) _01_AD0_RANGE(%UB0.1.22) : Input channel 0 range setting.
_01_AD1_RANGE(%UB0.1.23) : Input channel 1 range setting.

	Bit15	Bit4	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
%UW0.1.11	Don't Care				Input channel 1				Don't Care				Input channel 0			

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

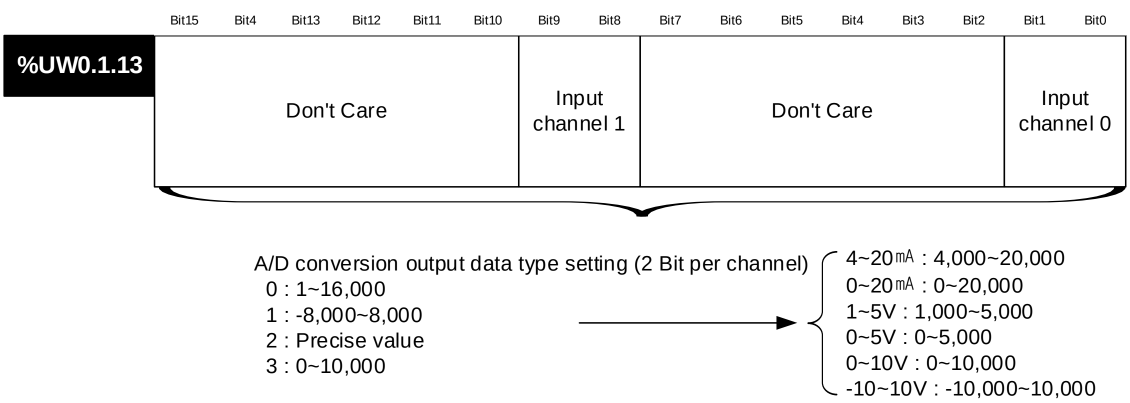
(11) Output range setting

(a) The ranges of analog output voltage are DC 1~5V, DC 0~5V, DC 0~10V, DC -10~10V.
(b) When the input range is not set or it is entered out of setting values, it is handled as range of DC 1~5V.
(c) _01_DA0_RANGE(%UB0.1.24) : Output channel 0 range setting.
_01_DA1_RANGE(%UB0.1.25) : Output channel 1 range setting.



(12) Built-in analog Input data type setting

(a) The range of digital input data(A/D conversion) can be specified for respective channels.
(b) If the input data range is not specified, the range of all the channels will be set to 0 ~ 16000.
(c) _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.



(13) Built-in analog Output data type setting

(a) The range of digital input data(D/A conversion) can be specified for respective channels.
(b) If the input data range is not specified, the range of all the channels will be set to 0 ~ 16000.
(c) _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.

<table>
<tr><th></th><th>Bit15</th><th>Bit4</th><th>Bit13</th><th>Bit12</th><th>Bit11</th><th>Bit10</th><th>Bit9</th><th>Bit8</th><th>Bit7</th><th>Bit6</th><th>Bit5</th><th>Bit4</th><th>Bit3</th><th>Bit2</th><th>Bit1</th><th>Bit0</th></tr>
<tr><td>%UW0.1.14</td><td colspan="6">Don't Care</td><td colspan="2">Output Channel 1</td><td colspan="6">Don't Care</td><td colspan="2">Output Channel 0</td></tr>
</table>

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

(a) When the filter constant is specified with 0, the filter will not be operated.
(b) _01_AD0_FILCONST(%UW0.1.15) : Input channel 0 Filter constant
_01_AD1_FILCONST(%UW0.1.16) : Input channel 1 Filter constant

<table>
<tr><th></th><th>Bit15</th><th>Bit4</th><th>Bit13</th><th>Bit12</th><th>Bit11</th><th>Bit10</th><th>Bit9</th><th>Bit8</th><th>Bit7</th><th>Bit6</th><th>Bit5</th><th>Bit4</th><th>Bit3</th><th>Bit2</th><th>Bit1</th><th>Bit0</th></tr>
<tr><td>%UW0.1.15</td><td colspan="16">Input channel 0 filter constant(0 or 64,000ms)</td></tr>
<tr><td>%UW0.1.16</td><td colspan="16">Input channel 1 filter constant(0 or 64,000ms)</td></tr>
</table>

(15) Average process method setting

(a) When setting average process, the average process method is selected among time average, count average, moving average, or weighted average.
(b) If setting average process is not specified, all channels will not handle the average process.
(c) _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.

<table>
<tr><th></th><th>Bit15</th><th>Bit4</th><th>Bit13</th><th>Bit12</th><th>Bit11</th><th>Bit10</th><th>Bit9</th><th>Bit8</th><th>Bit7</th><th>Bit6</th><th>Bit5</th><th>Bit4</th><th>Bit3</th><th>Bit2</th><th>Bit1</th><th>Bit0</th></tr>
<tr><td>%UW0.1.17</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>

Average process method setting (4 Bit per channel)
0 : Sampling
1 : Time average
2 : Count average
3 : Moving average
4 : Weighted average

(16) Average value setting

(a) Set to range of 4 ~ 16,000 as time average value.
(b) Set to range of 2 ~ 64,000 as count average value.
(c) Set to range of 2 ~ 100 as moving average value.
(d) Set to range of 1~ 99 as weighted average value.
(e) 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.
(f) _01_AD0_AVGVAL(%UW0.1.18) : Input channel 0 Average value.
_01_AD1_AVGVAL(%UW0.1.19) : Input channel 1 Average value.

Bit15 Bit4 Bit13 Bit12 Bit11 Bit10 Bit9 Bit8 Bit7 Bit6 Bit5 Bit4 Bit3 Bit2 Bit1 Bit0

%UW0.1.18	Input channel 0 average value
%UW0.1.19	Input channel 1 average value

(17) Hold last value setting

(a) 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.
(b) 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".
(c) For the detailed usage, refer to chapter 13.5.5 Hold Last Value Function.
(d) 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

(a) When the motion controller is stopped, set the analog output status
(b) When the output status setting is not specified, output the previous value.

	Bit15 Bit4 Bit13 Bit12 Bit11 Bit10 Bit9 Bit8 Bit7 Bit6 Bit5 Bit4 Bit3 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	Device	Comment	Setting
_01_DA0_OUTSTAT	%UW0.1.21	Channel 0 output status setting	Input data type setting (bit) - 00 : Previous value - 01 : Min value - 10 : Mid value - 11 : Max value
_01_DA1_OUTSTAT	%UW0.1.22	Channel 1 output status setting	

(19) Interpolation method setting

(a) Shows the setting of the interpolation method of each channel.
(b) _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 period setting

(a) Shows the setting of interpolation time of each channel.
(b) _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

(a) Shows the interpolation operation value of each channel.
(b) _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) Error code

(a) Shows the error code of each channel.
(b)If it is normal, the error code is 0.
(c)_01_SETTINGERR(%UW0.1.27) : Error information.

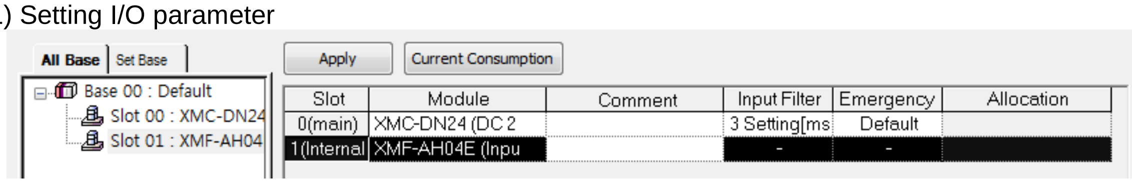
	Bit15	Bit4	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
%UW0.1.27	Setting error information															

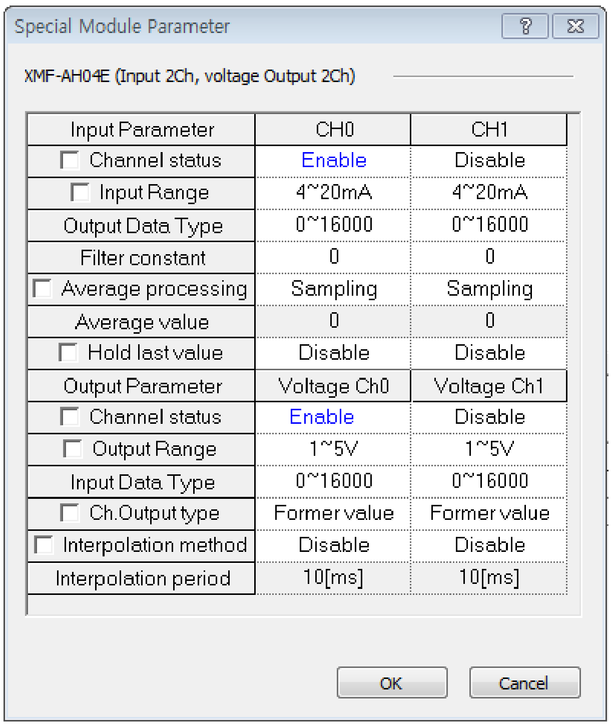
Type	Error code (Decimal)	LED sign	Details	Priority of error code	Remarks for reference
Error	10#	AD LED Flickering 1s intervals	Setting error of input channel range	1	#:Channel number (Channel0~1)
	20#		Setting error of input channel filter value	2	
	30#		Setting error of input channel average value	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#		Setting error of output channel interpolation method range	6	

(d) 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. In case of that the errors are occurred at the same time in voltage output channel and current output channel, the error code of voltage output channel is saved preferentially.

13.11 Example Program

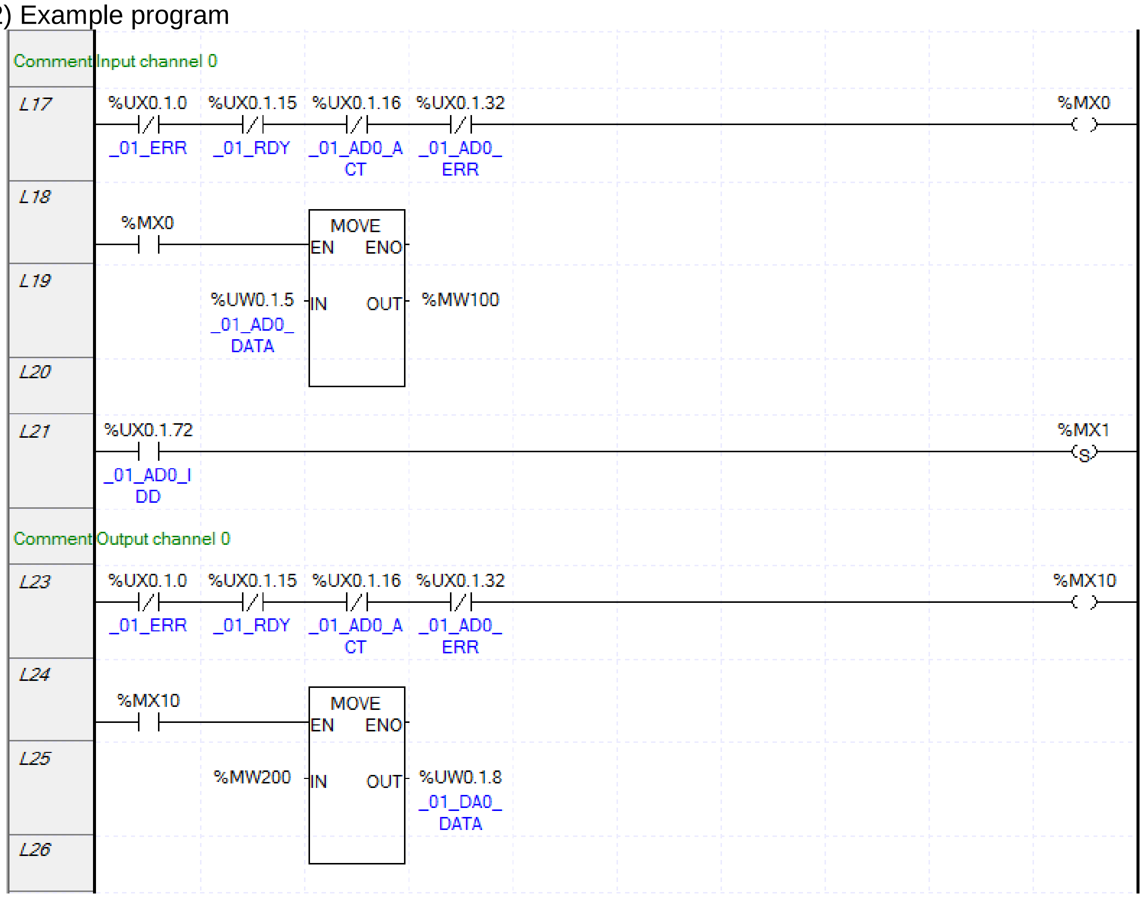
(1) Setting I/O parameter





(a) The input channel 0 is set with operation channel and the range is set with 4~20mA.
(b) The voltage output channel 0 is set with operation channel and the range is set with 1~5V.

(2) Example program



(a) Example of input program

1) 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
2) When the '%MX0' is on, conversion value (%UW0.1.5) of CH0 is moved to the '%MW100'.
3) If the error is caused on CH0, %UX0.1.72 (CH0 disconnection) will be on and the ‘%MX1’ will be on.

(b) Example of output program

1) 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
2) When the '%MX10' is on, voltage channel 0 output status (%UX0.1.112) is on, and the output is permitted.
3) If ‘%MX10’ is on, ‘%MW200’ data is moved to voltage channel 0 output value (%UW0.1.8) and then it is output.

13.12 Troubleshooting

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 channel 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 range 4 to 20mA or 0 to 20mA)	Flickering 1s interval (Output parameter setting error)
Operation	•Normal operation •Operation of all functions	•Operation of all functions •Shows minimum input value	Operation of all functions with default parameter
Measure	-	Check wiring	Check parameter

13.12.2 Check the Built-in Analog Module

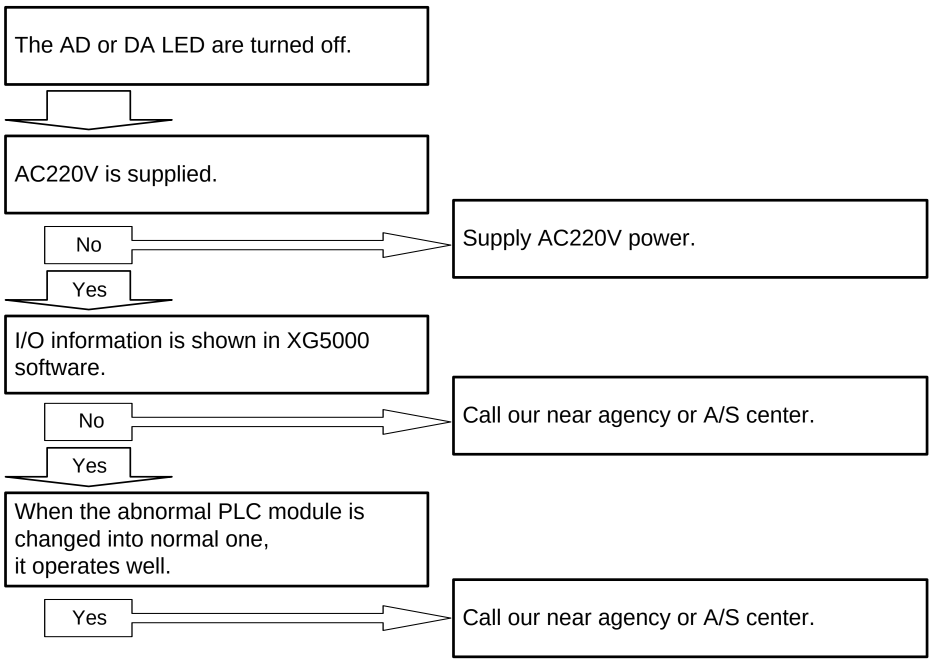
The status of built-in analog module can be checked through the system monitor of XG5000.

(1) The order of execution
It can be implemented through one of the methods among next items.
(a) [Monitor] → [System Monitor] → Click the right button of mouse on the painting of module.
→ [Module Information]
(b) [Monitor] → [System Monitor] → Double click the painting of module
(c) [Monitor] → [Special Module Monitor] → Built-in Analog Module Selection
→ Click the module information
(d) [Online] → [I/O Information] → Built-in Analog Module Selection → Click the details
(e)[Online] → [I/O Information] → Built-in Analog Module Double click

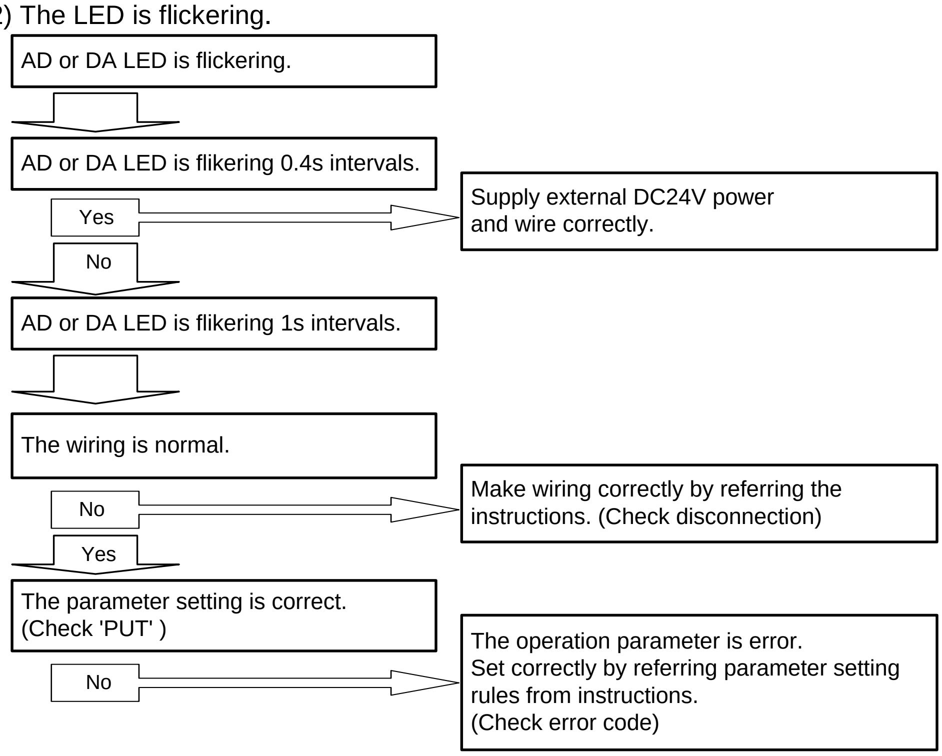
(2) Module information
(a) OS Version: OS version of module is shown.
(b) OS Update Date: The OS prepared date of module is shown.
(c) Module status: The present error code is shown.

13.12.3 Troubleshooting

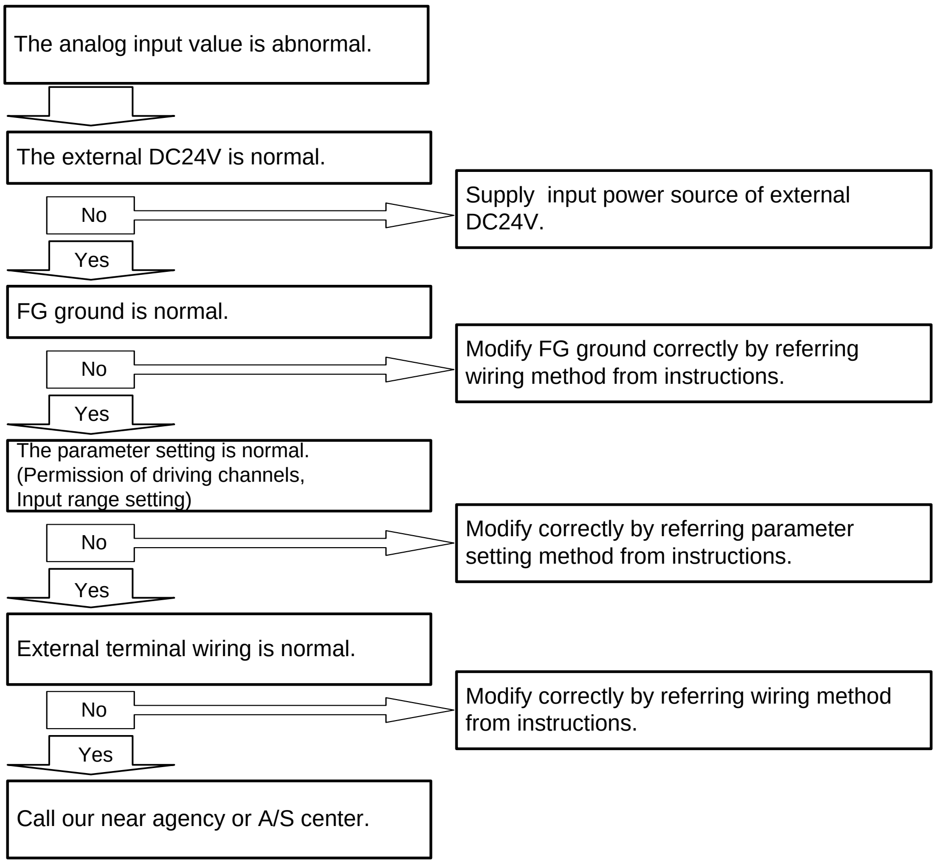
(1) The AD or DA LED is turned off.



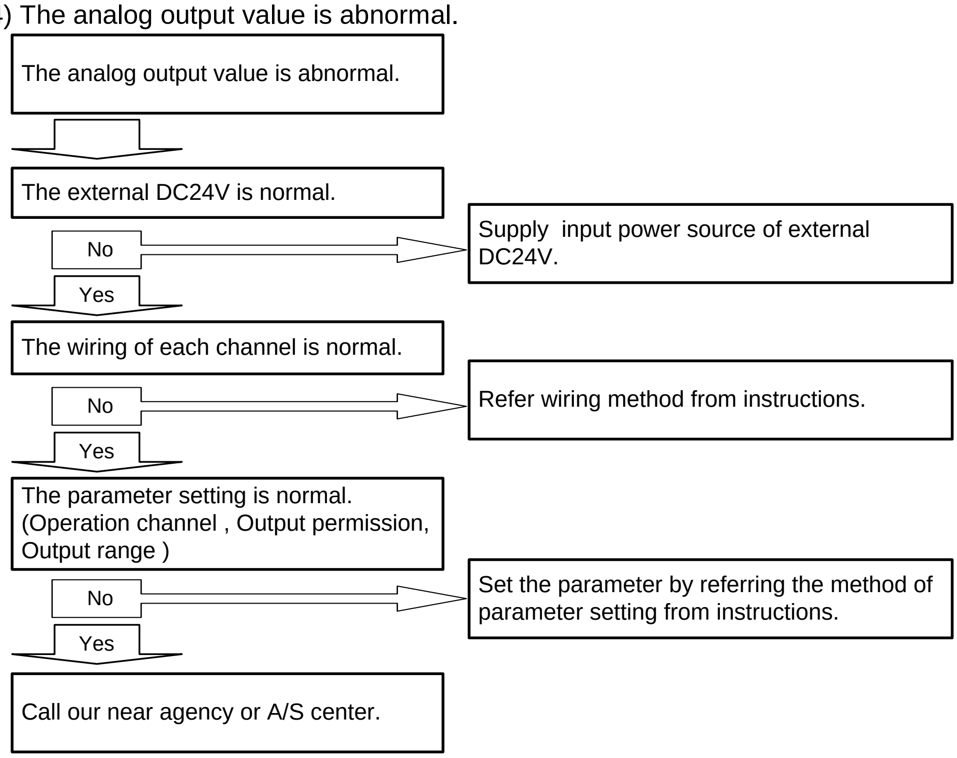
(2) The LED is flickering.



(3) The analog input value is abnormal.



(4) The analog output value is abnormal.



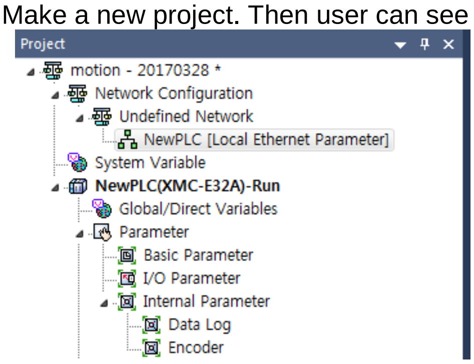
Chapter 14 Local Ethernet Function

14.1 Local Ethernet Function

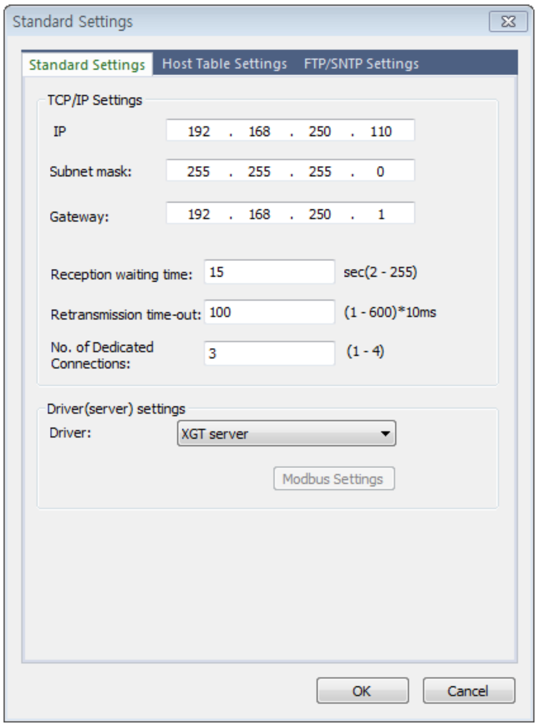
Motion controller can carry out the functions of Ethernet server using internal local Ethernet function.

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.

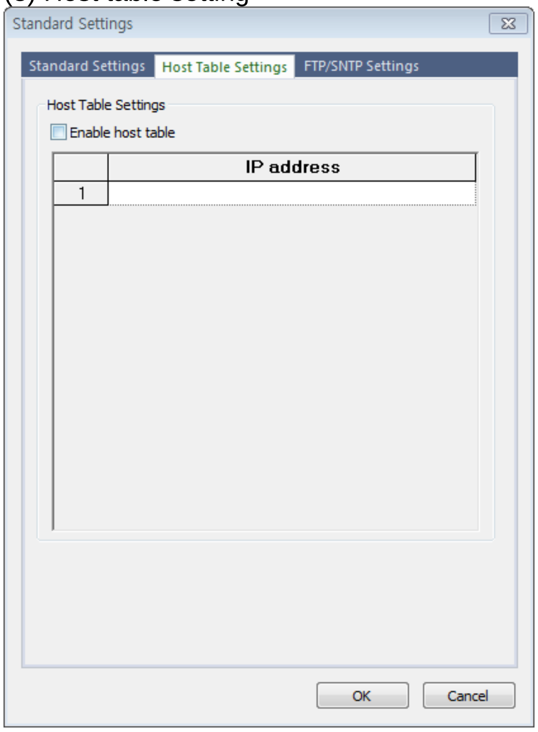
(1) TCP/IP Setting

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

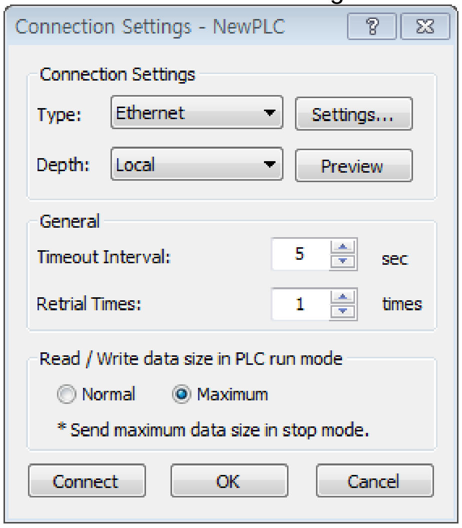
14.1.2 Local Ethernet Connection with XG5000

After finishing Local Ethernet Parameter settings, download the settings to the motion controller, then user can connect to XG5000.

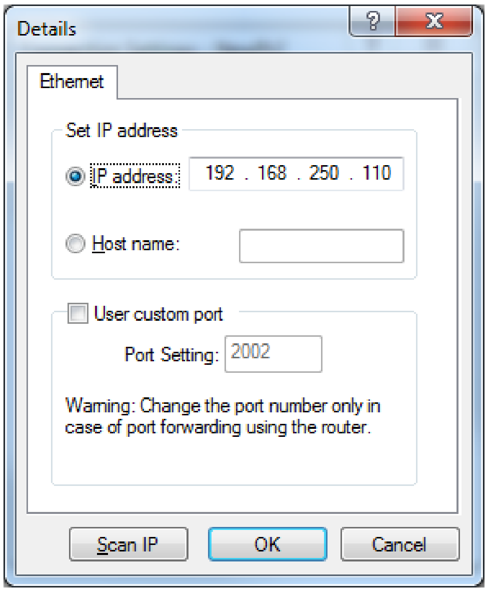
Select Online Settings and set the options as shown below figure.

(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.

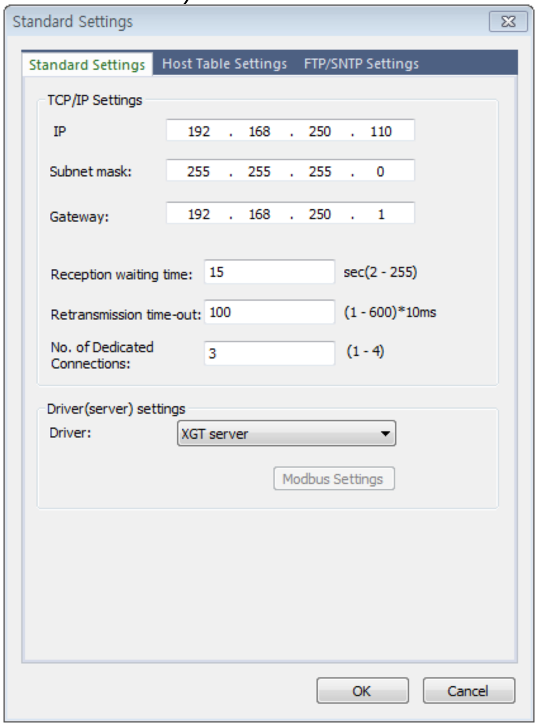


Click the setting button to specify Ethernet IP. Click OK after specify the Ethernet IP set before.
User can find the IP information available now.



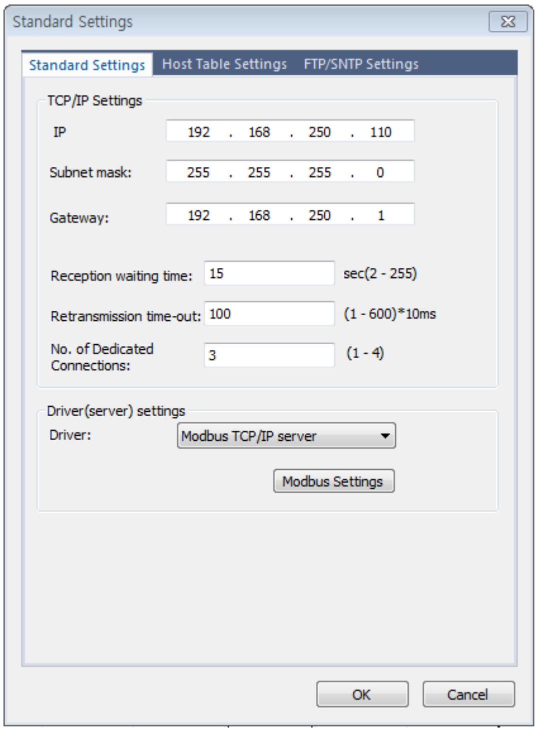
14.1.3 Local Ethernet Connection with XGT Server

Set the Local Ethernet Parameters as shown below figure. User can use it as a XGT Server (LSIS dedicated Protocol Communication).

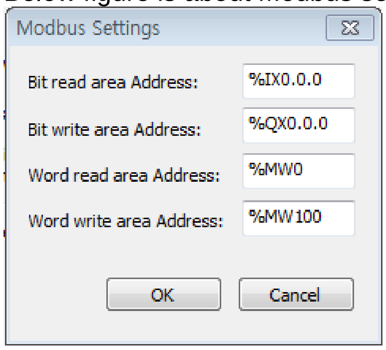


14.1.4 Local Ethernet Connection with TCP/IP Server

Set the Local Ethernet Parameters as shown below figure. User can use it as a Modbus server.



Below figure is about Modbus settings. .



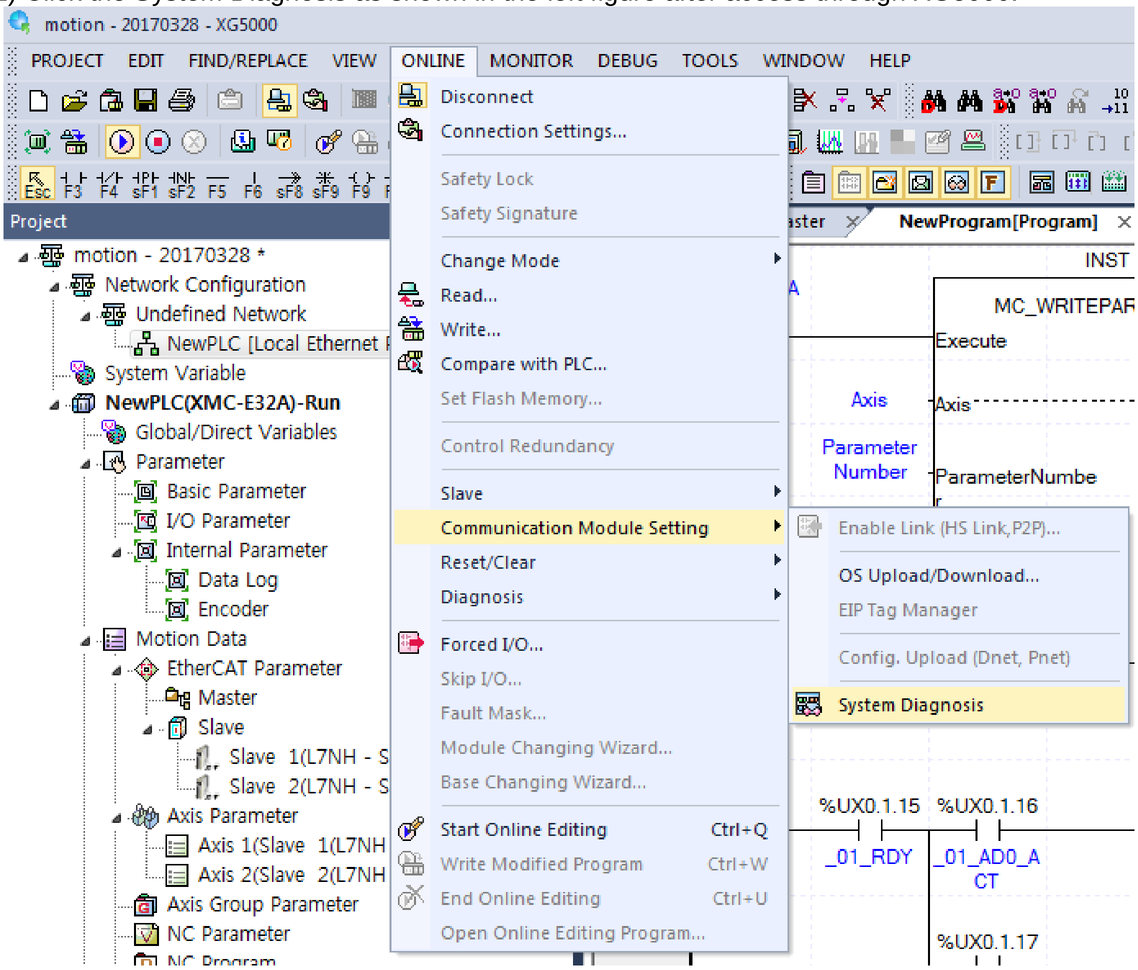
Note

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.

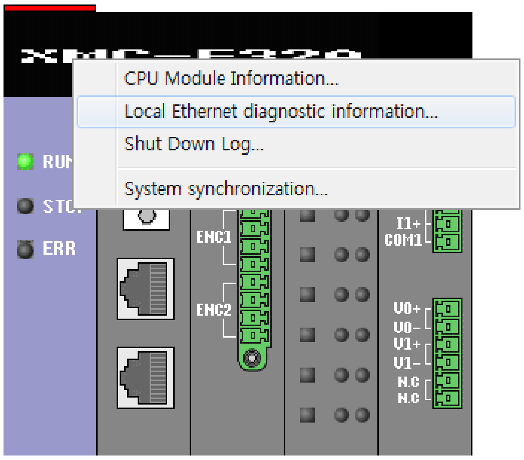
14.1.5 Local Ethernet Diagnosis Information Function

Motion controller provides local Ethernet diagnosis information function to monitor the status of local Ethernet.

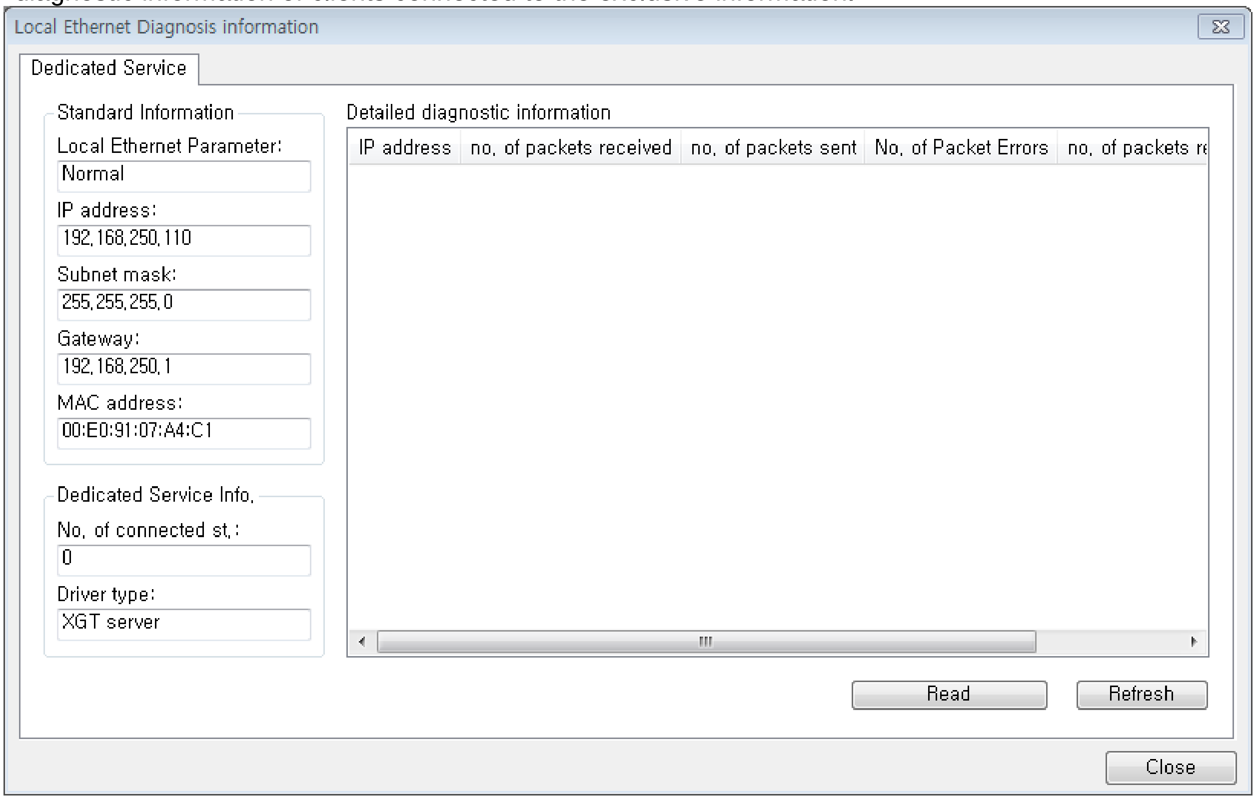
(1) Click the System Diagnosis as shown in the left figure after access through XG5000.



(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.



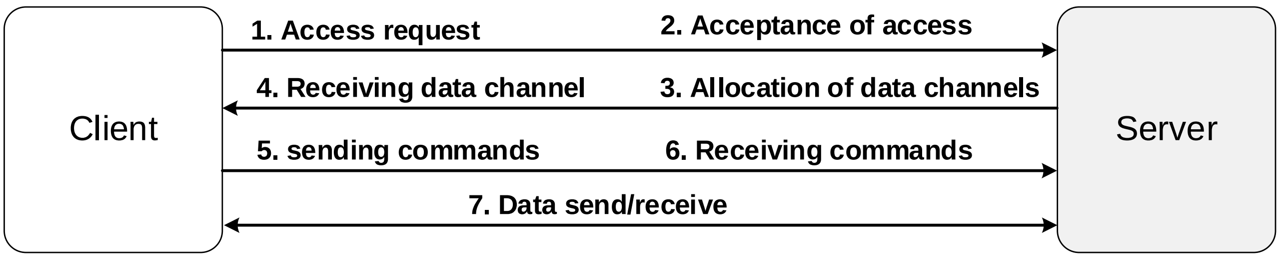
(3) If you click the loca1l Etnernet diagnostic information among the menus that occur when clicking the right mouse button, the local Etnernet diagnostic information window will be created as shown below. Through the local Ethernet information window, you can monitor the Ethernet basic information, 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.

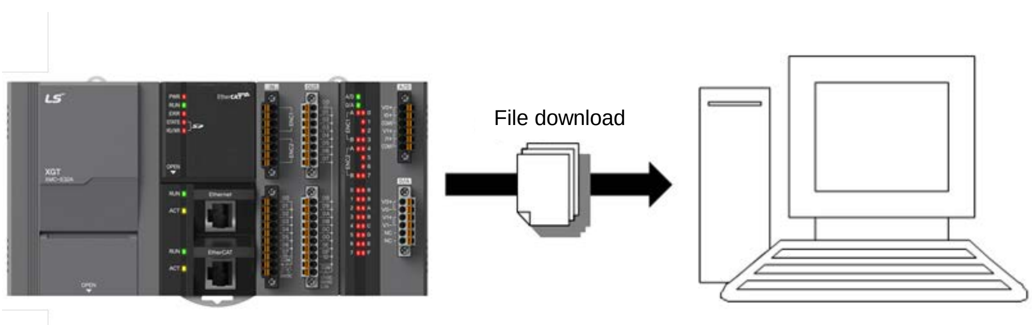


14.2.2 Support Functions

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)

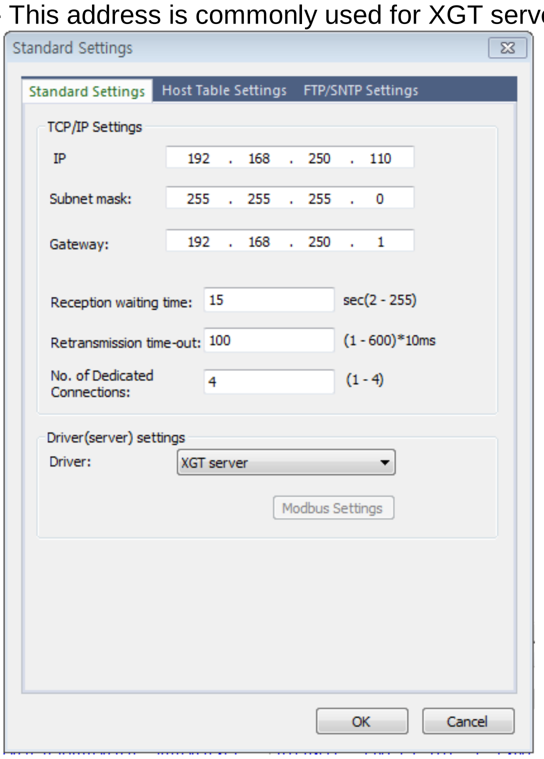
It is the function to import and save files to the devices with FTP client or PC (Personal Computer) from motion controller's FTP server. The files are saved to the designated directory path of the devices with FTP client or PC.



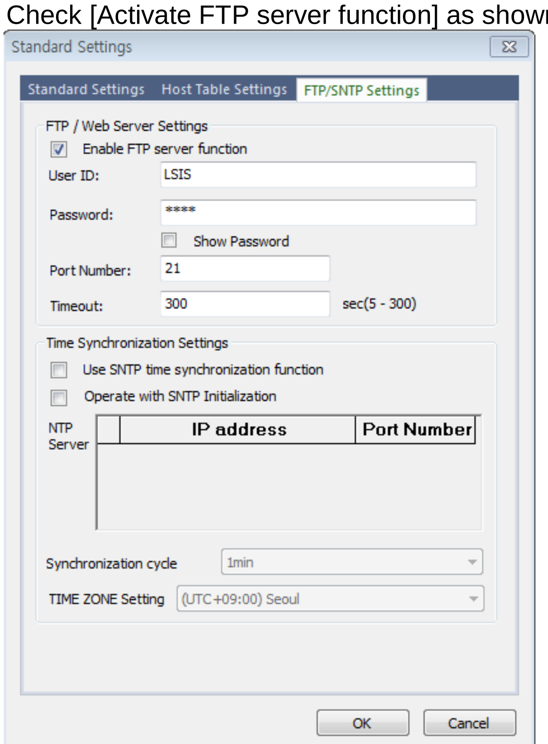
14.2.3 Setting FTP Server Parameters

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.

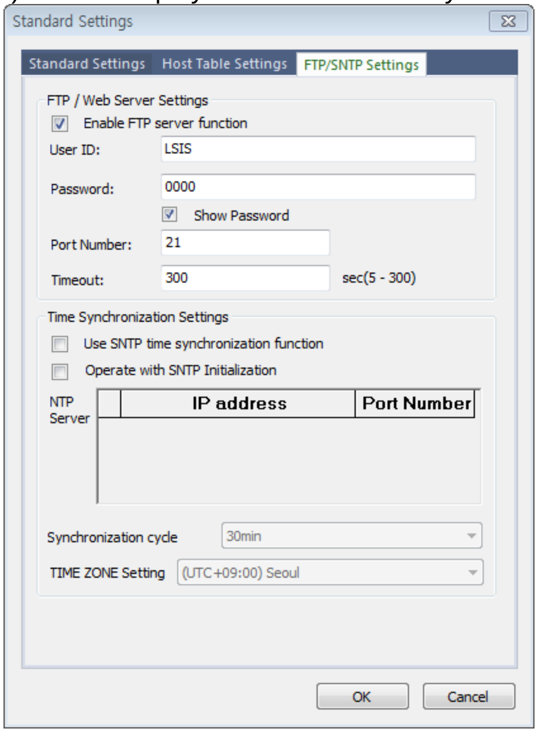


(2) Check [Activate FTP server function] as shown below figure.



(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

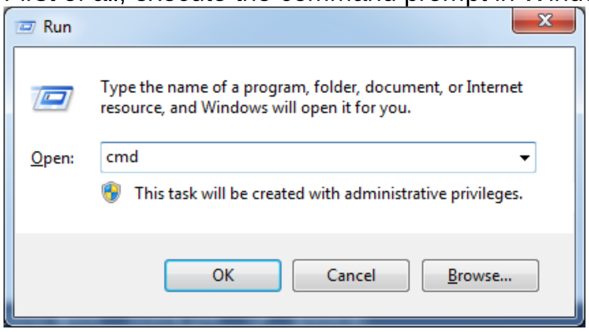
1. Unless you set the user ID and password, basic ID and password will be set initially
   - Default setting ID: LSIS
   - Default password: 0000
2. Rules for applying the user ID and password
   - You can enter the user ID and password that are composed of alphabetical characters and numbers but special characters are not available.
   - They must be case-sensitive and must not exceed the maximum of 8 digits.

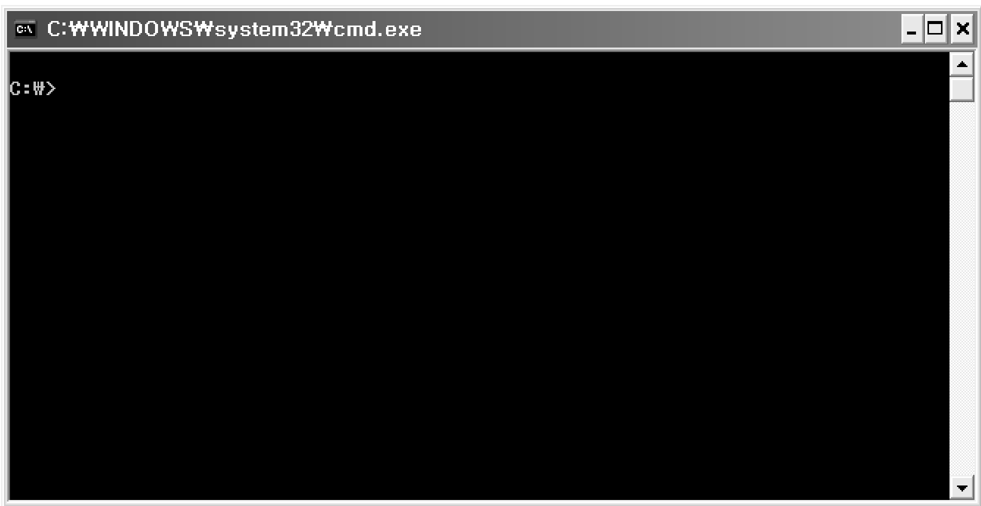
14.2.4 How to Access to the FTP Server

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

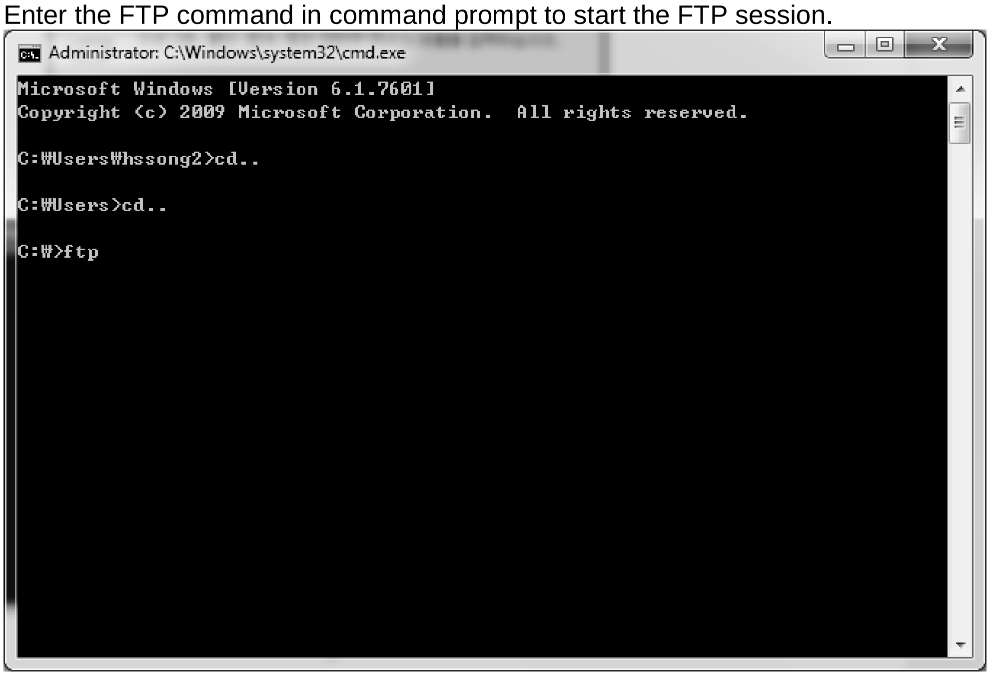




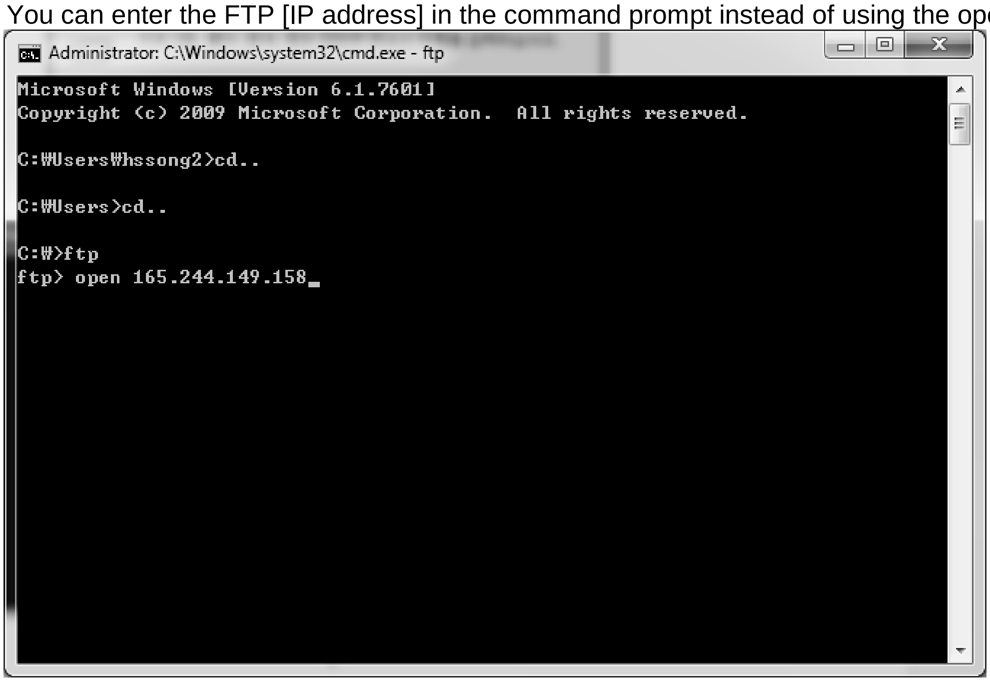
Note

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.

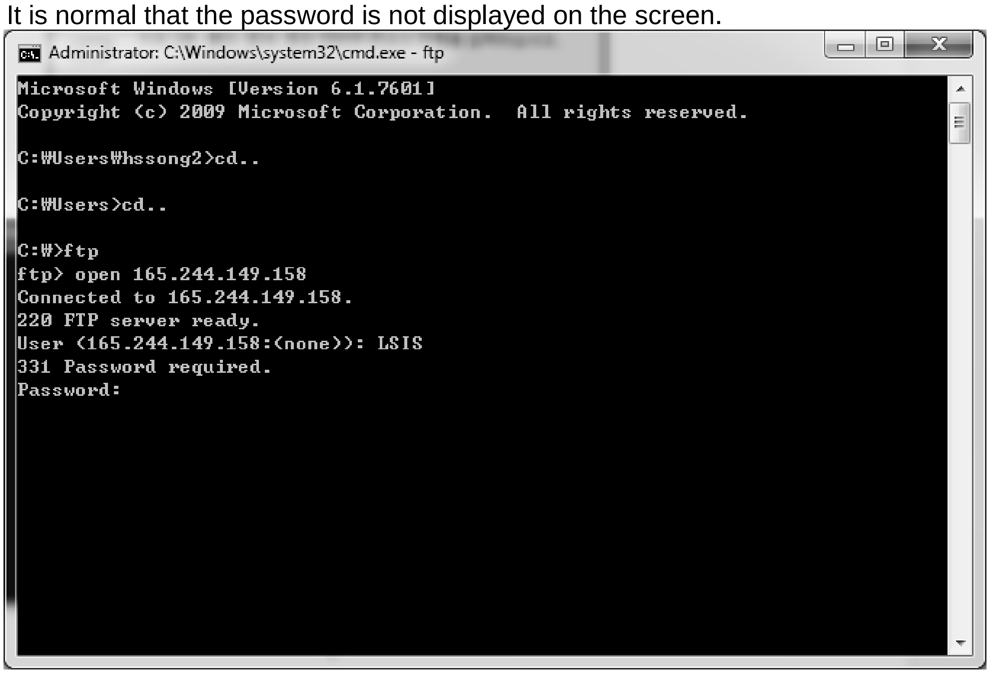
(b) Enter the FTP command in command prompt to start the FTP session.



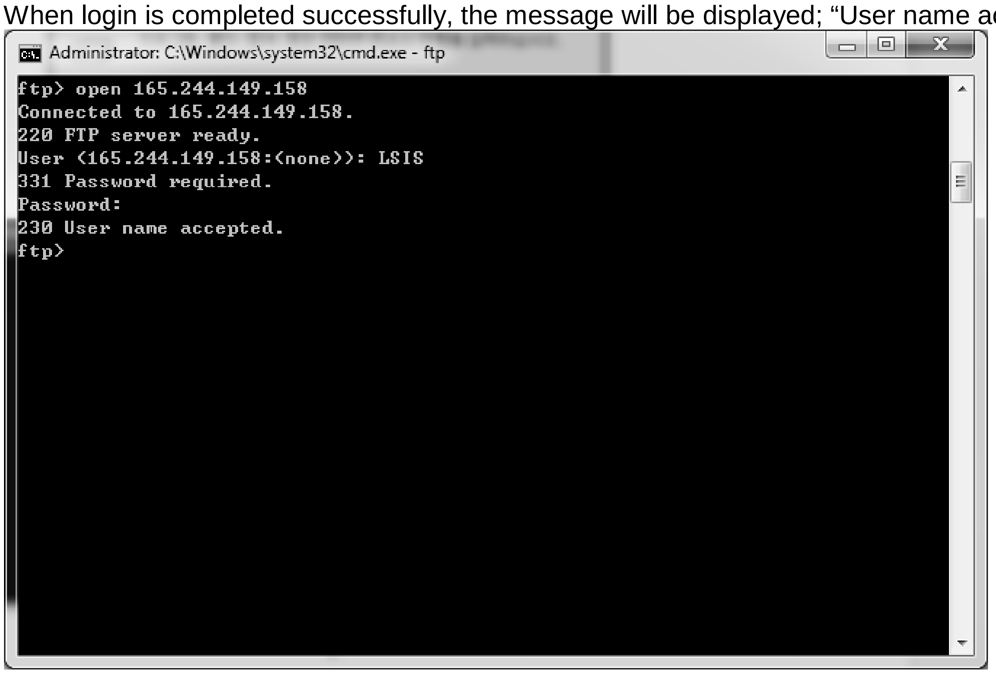
(c) Enter the ‘open [IP address]’to access to the FTP server
- You can enter the FTP [IP address] in the command prompt instead of using the open command.



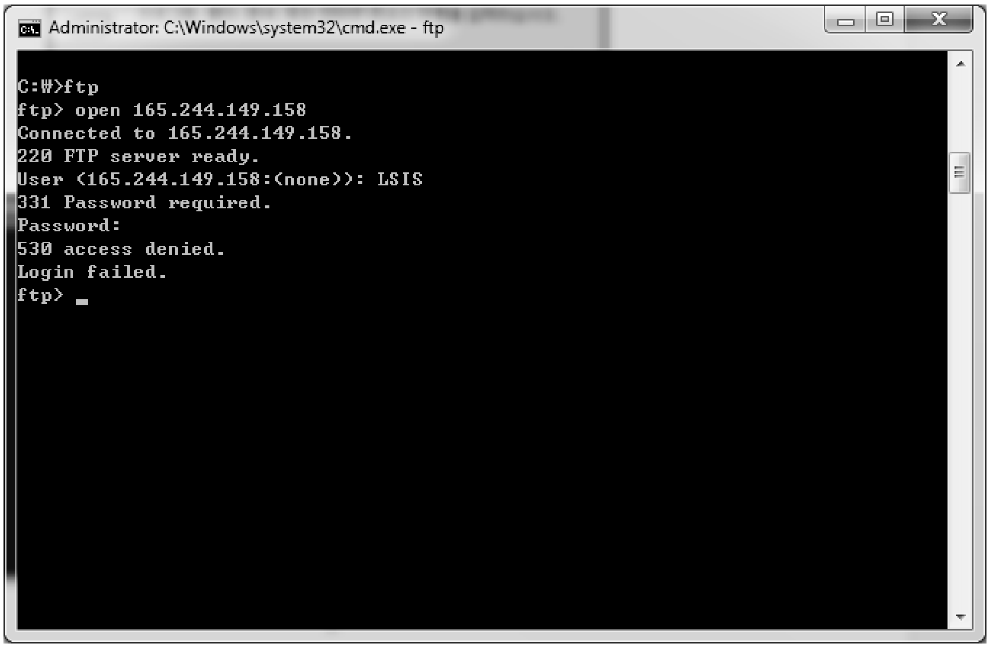
(d) Enter the user ID and password to access to the FTP server.
- It is normal that the password is not displayed on the screen.



(e) When login is completed successfully, the message will be displayed; “User name accepted.”

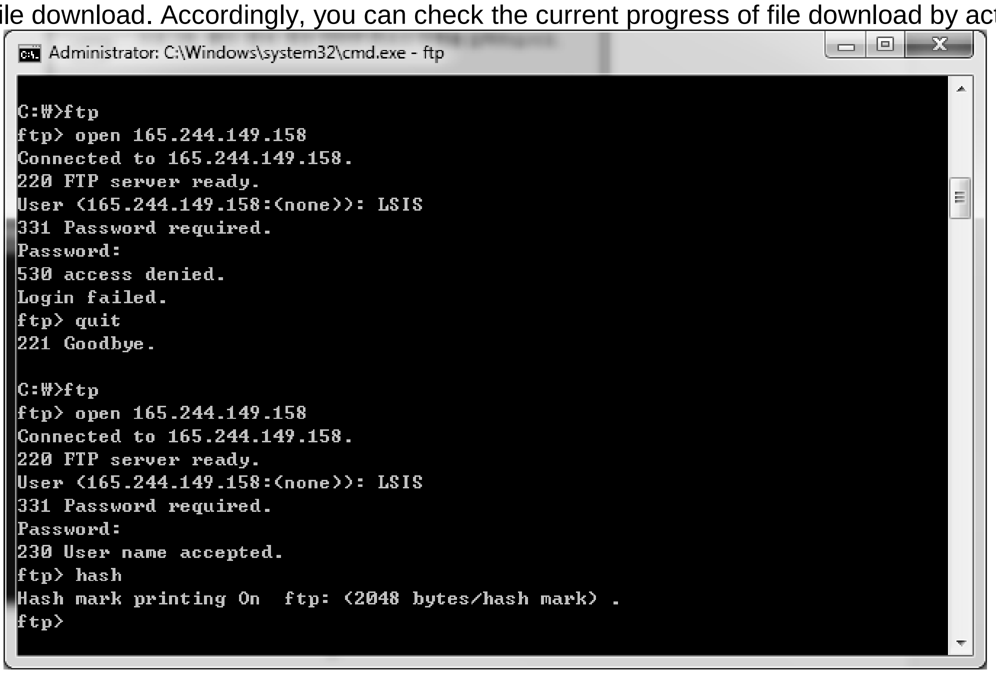


[Completion of FTP server access and login]

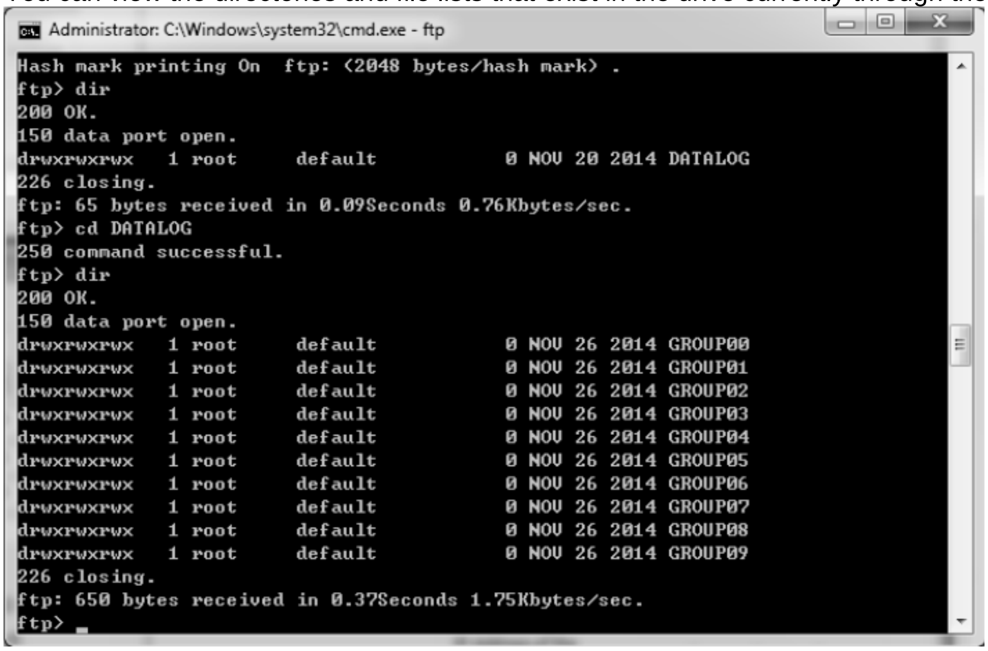


[Failure of FTP server login]

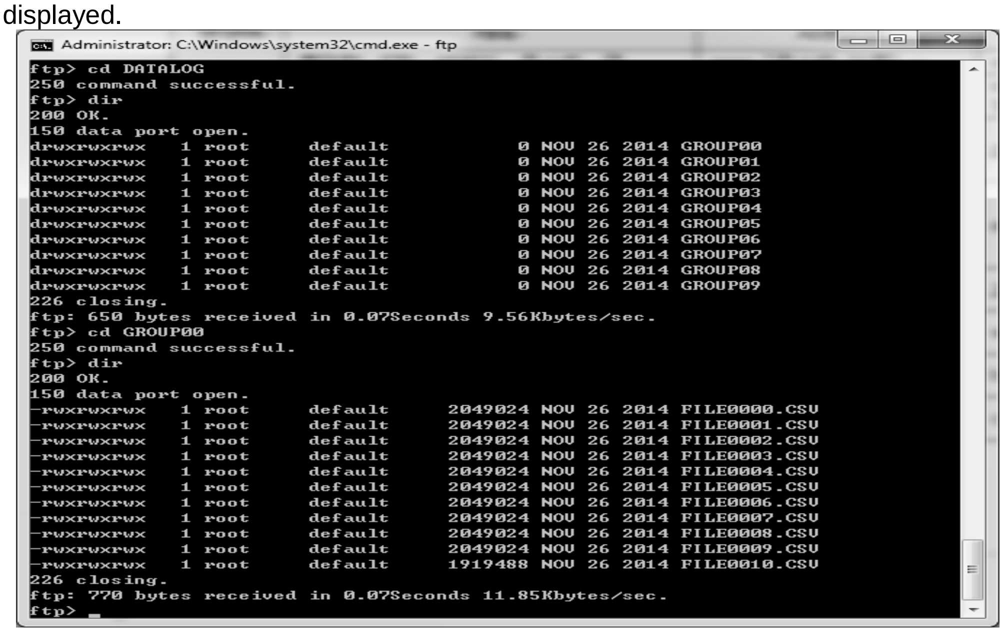
(f) In case you access to the FTP server through windows command prompts, you cannot see the progress status of file download. Accordingly, you can check the current progress of file download by activating the HASH function.



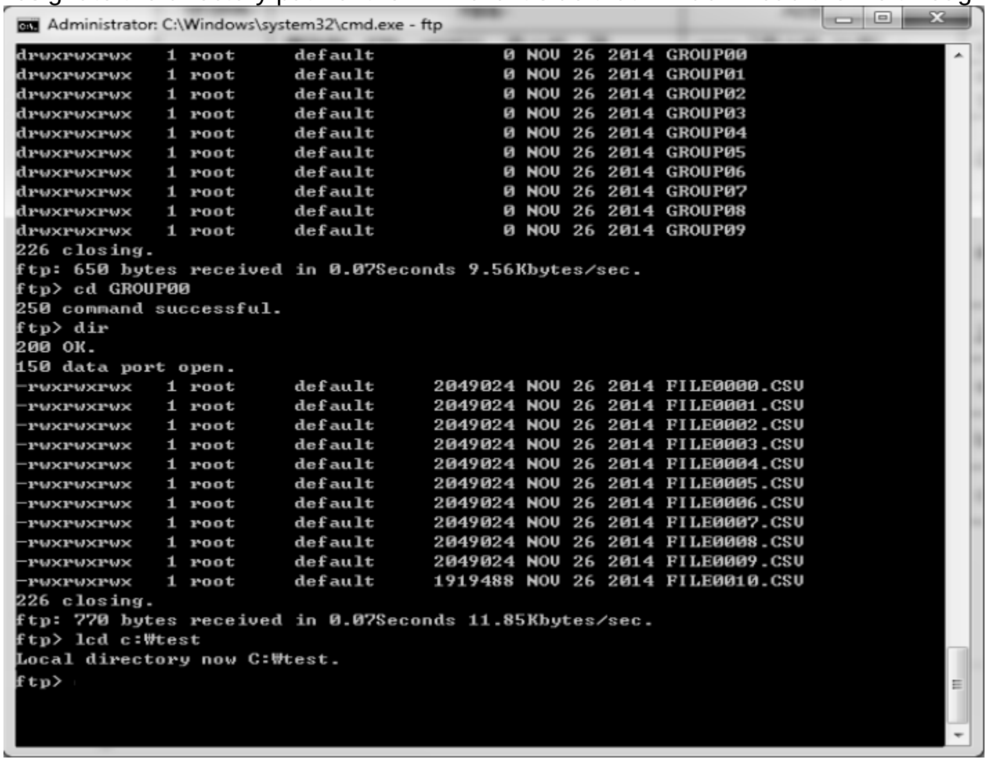
(g) You can view the directories and file lists that exist in the drive currently through the ‘DIR’ command.



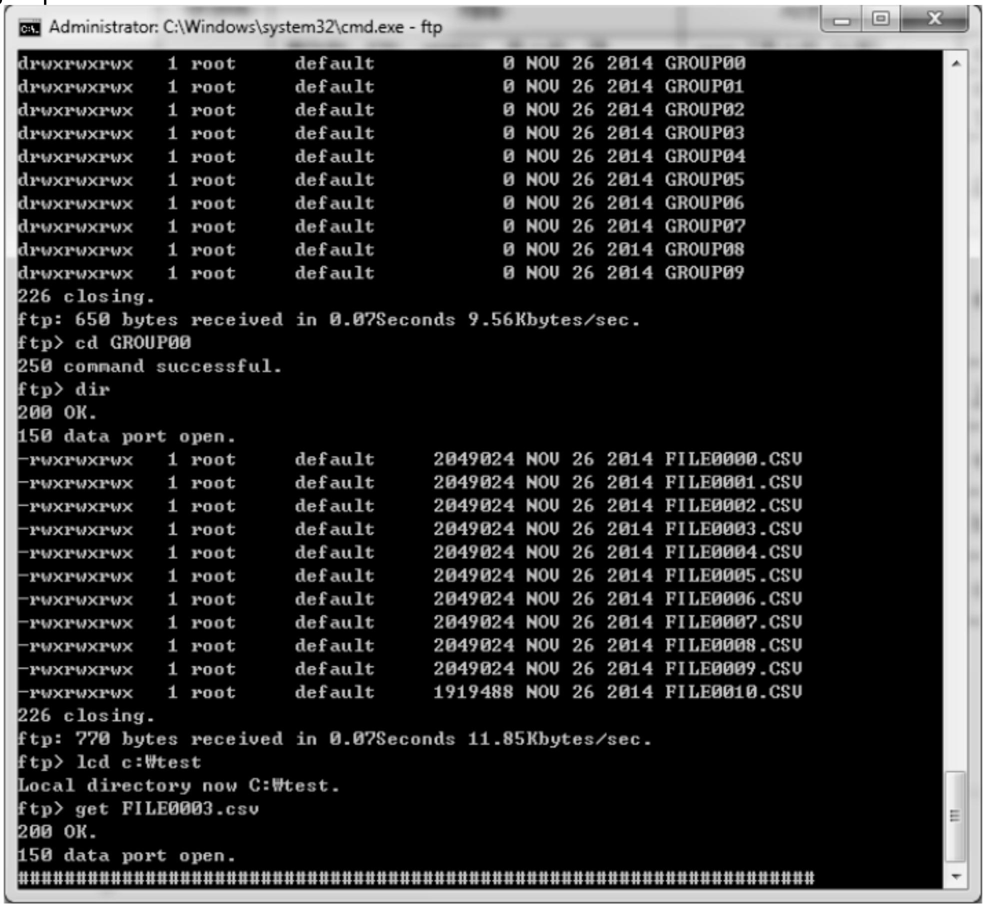
(h) You can go into the lower folder through the ‘cd [Folder name]’command. If you execute the ‘DIR’ command again after going into the lower folder, only the files ahasnd folder lists that exist in the lower folder will be displayed.



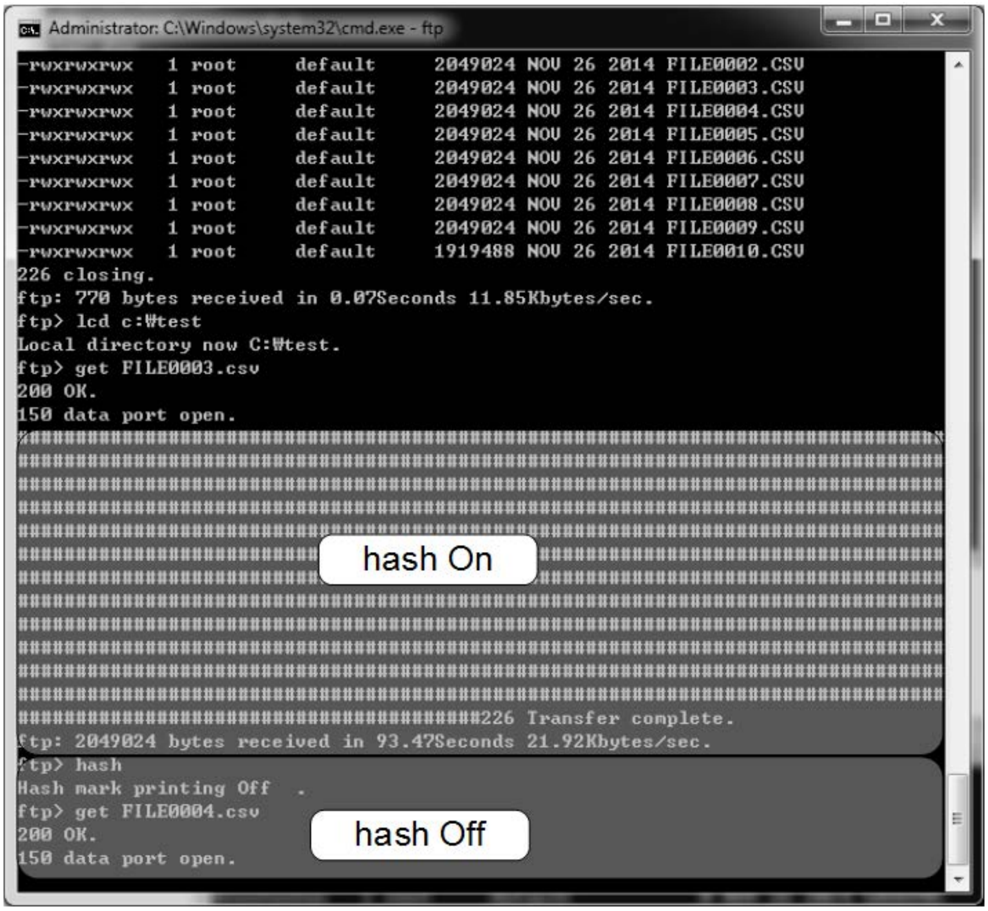
(i) Designate the directory path of the FTP client side that will download the file through the 'lcd' command.



(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

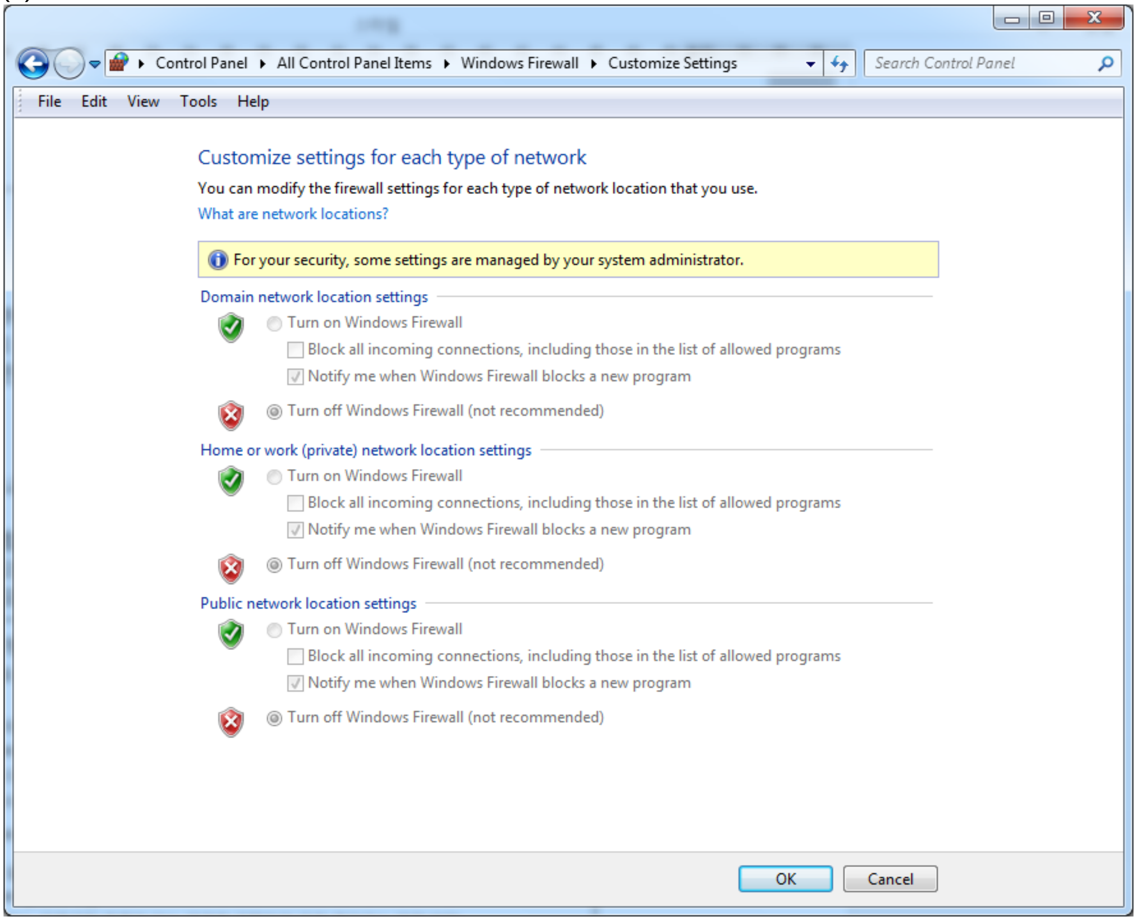
Response codes	Description
150	File status okay; about to open data connection
200	Command okay
202	Command not implemented, superfluous at this site
211	Type: ASCII, Structure: FILE, Mode: Stream
215	UNIX Type: L8 Version: Nucleus-ftpd
220	Nucleus FTP Server (Version 1.7) ready
221	Service closing control 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.



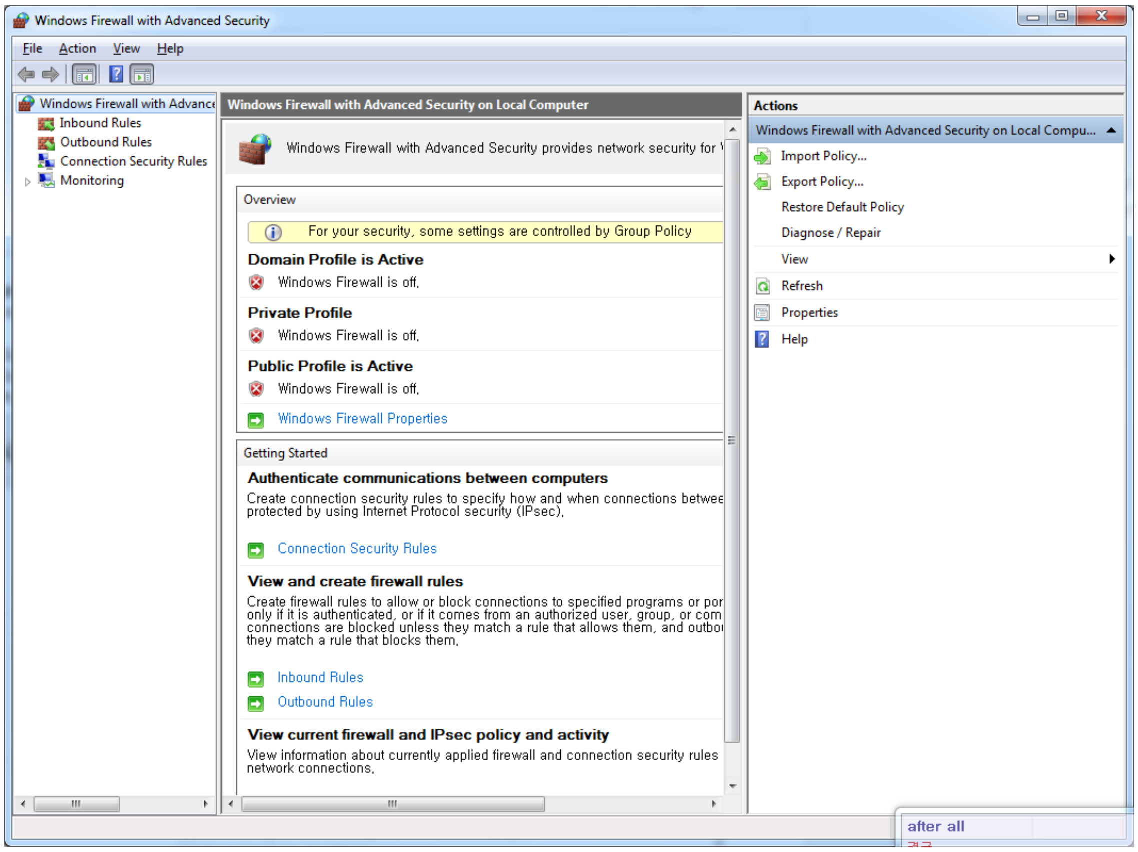
Note

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.

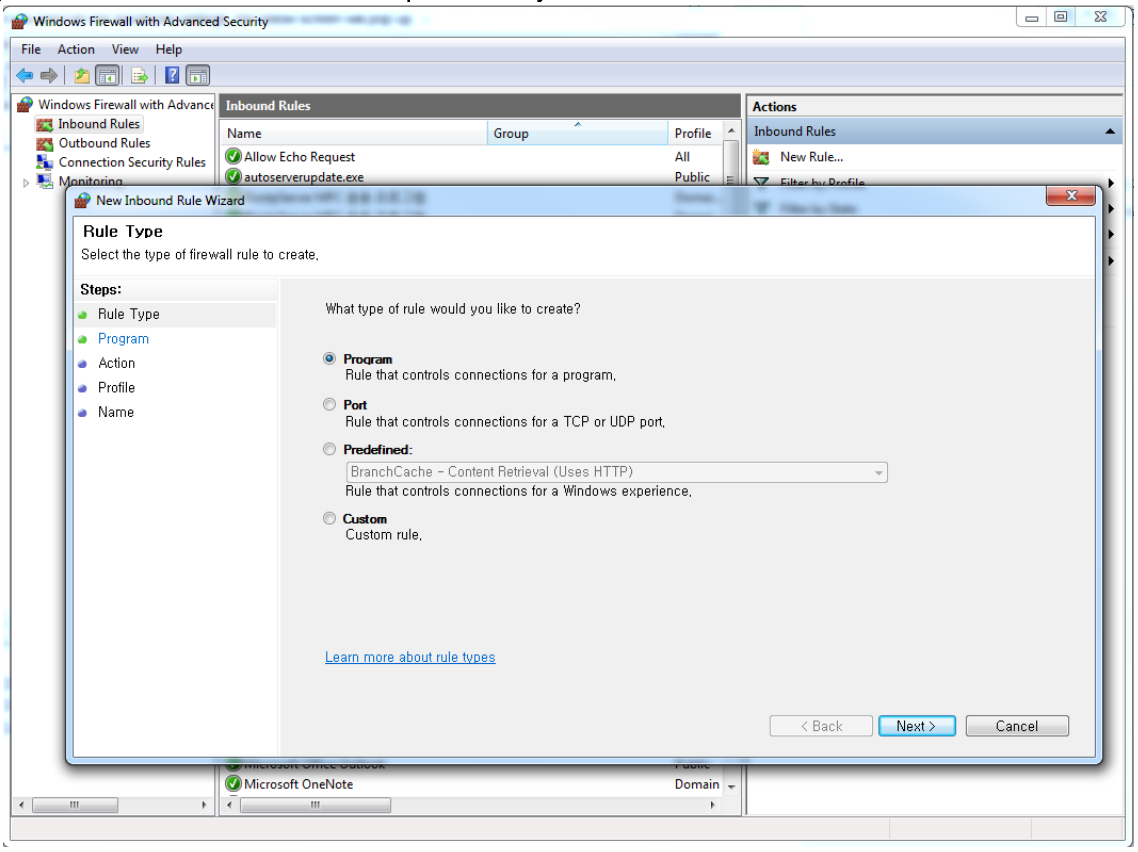
(2) Registration of exceptional rules

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

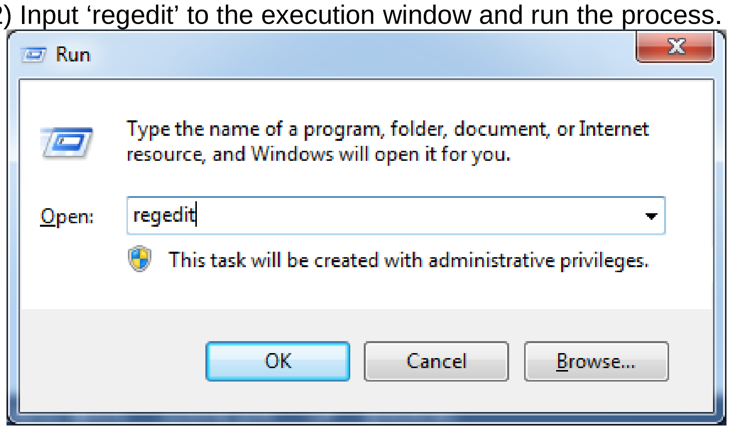
For registering exceptional rules, refer to the window manual.

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.

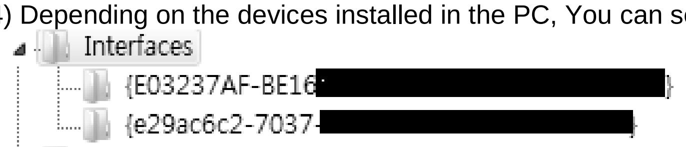
(1) Select the [Start] button of Windows for execution.(Shortcut key /Windows key + R)

(2) Input 'regedit' to the execution window and run the process.

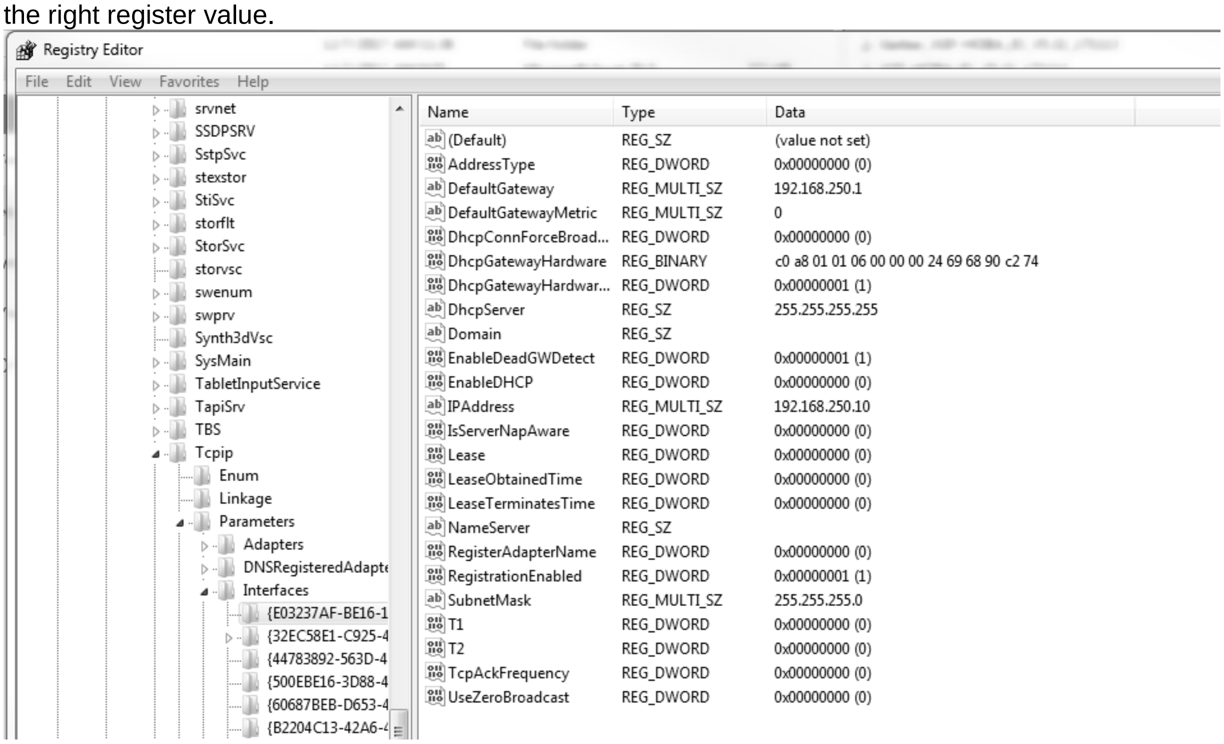


(3) Check the below path.
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\Tcpip\Parameters\Interfaces

(4) Depending on the devices installed in the PC, You can see the folders are created



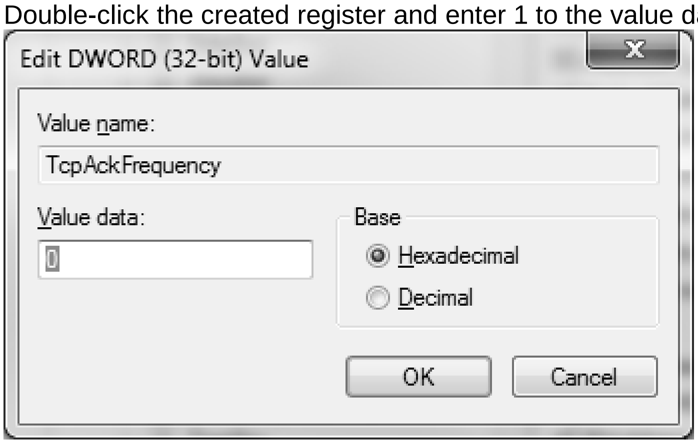
(5) If there are several register folders, select one by one and find the folder where the current PC's IP address is set in the right register value.



(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.

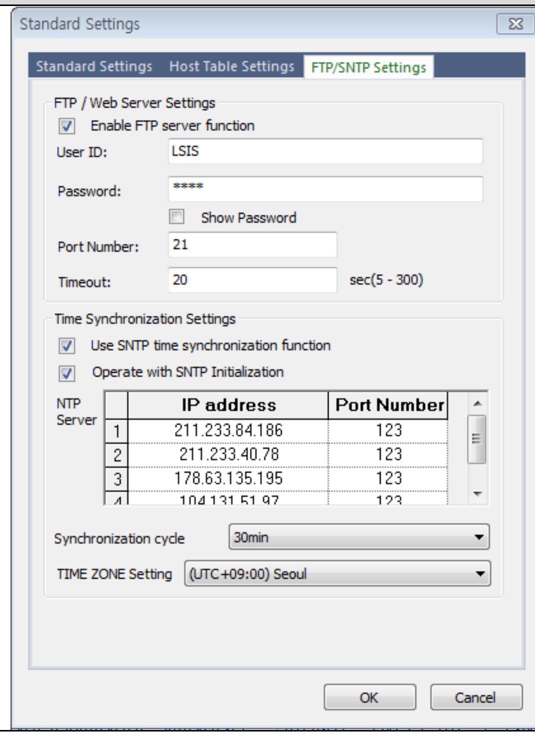
14.3 SNTP Client Functions

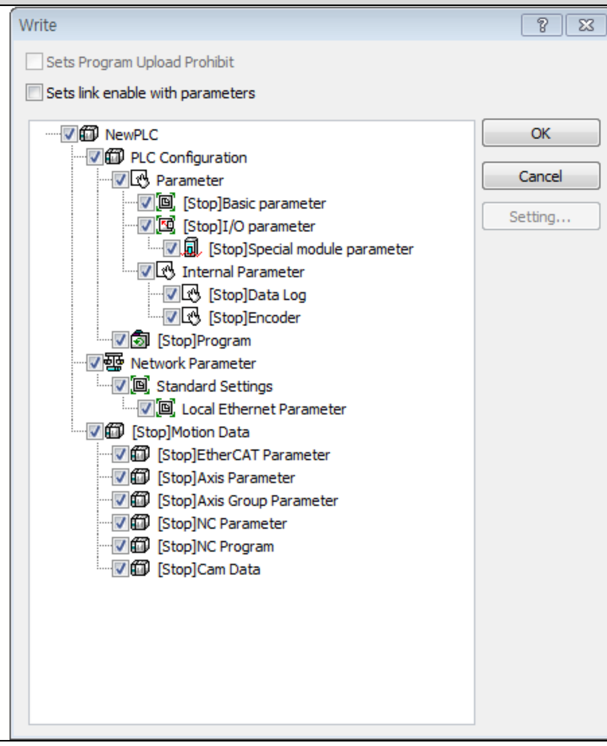
14.3.1 Outline of the Time Synchronization Protocol

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)

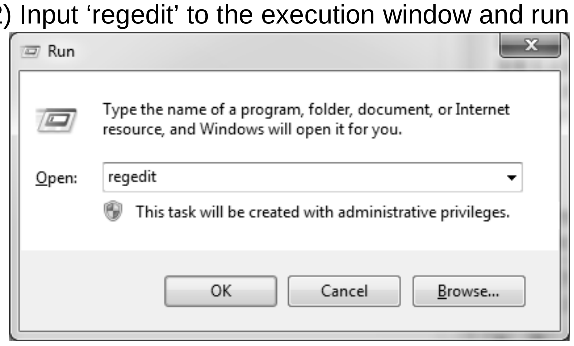
(4) If you cannot use a public NTP server, Please setup a local NTP server refer to **'14.3.3 How to setup a local NTP server'.**

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)

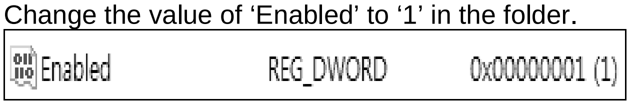
(2) Input 'regedit' to the execution window and run the process.



(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.

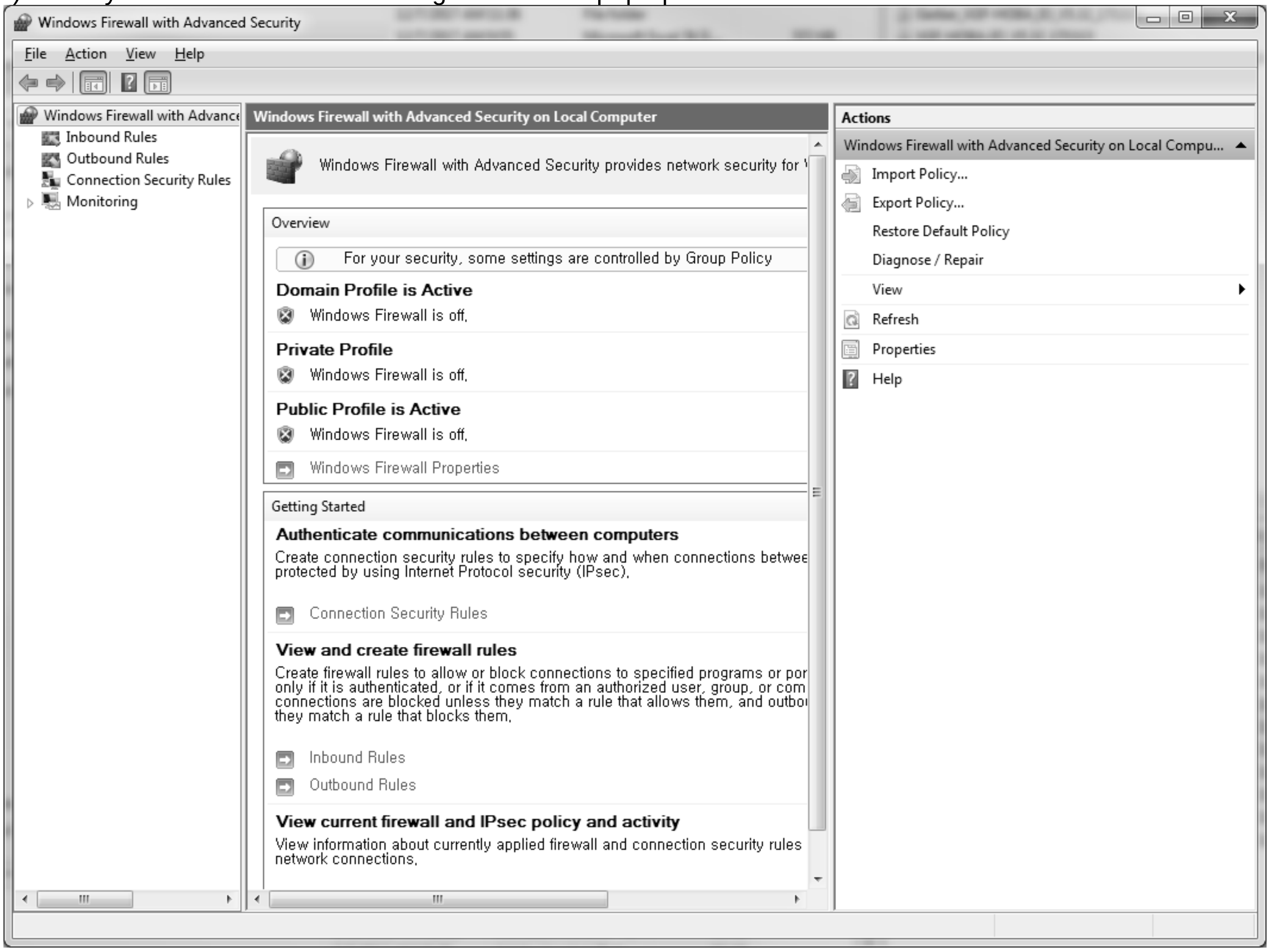


(5) Check the below path.
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W32Time\Config

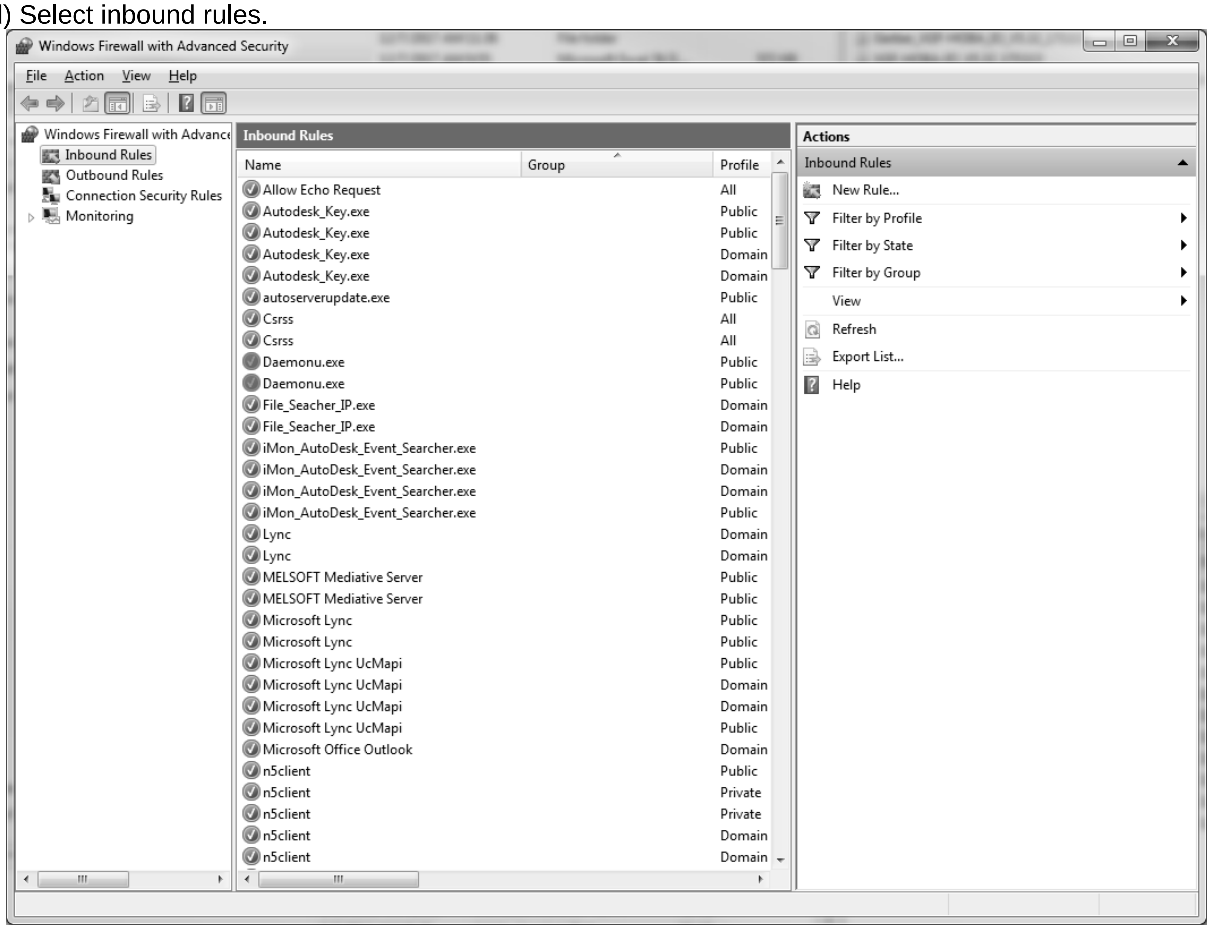
(6) Change the value of ‘AnnounceFlags’ to ‘5’ in the folder.

(7) Reboot the computer.

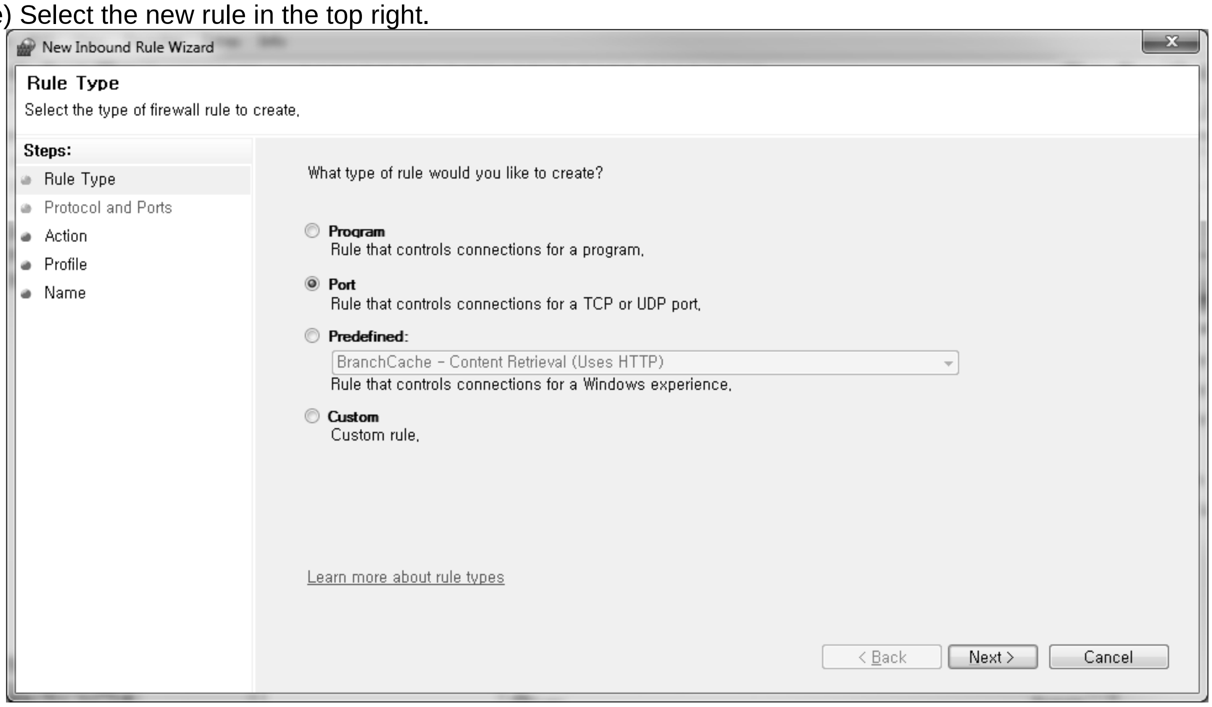
(8) Setup inbound firewall rules.
(a) Run the Control Panel.
(b) Run the Window Firewall
(c) When you run the Advanced Settings screen will pop up as shown below.



(d) Select inbound rules.

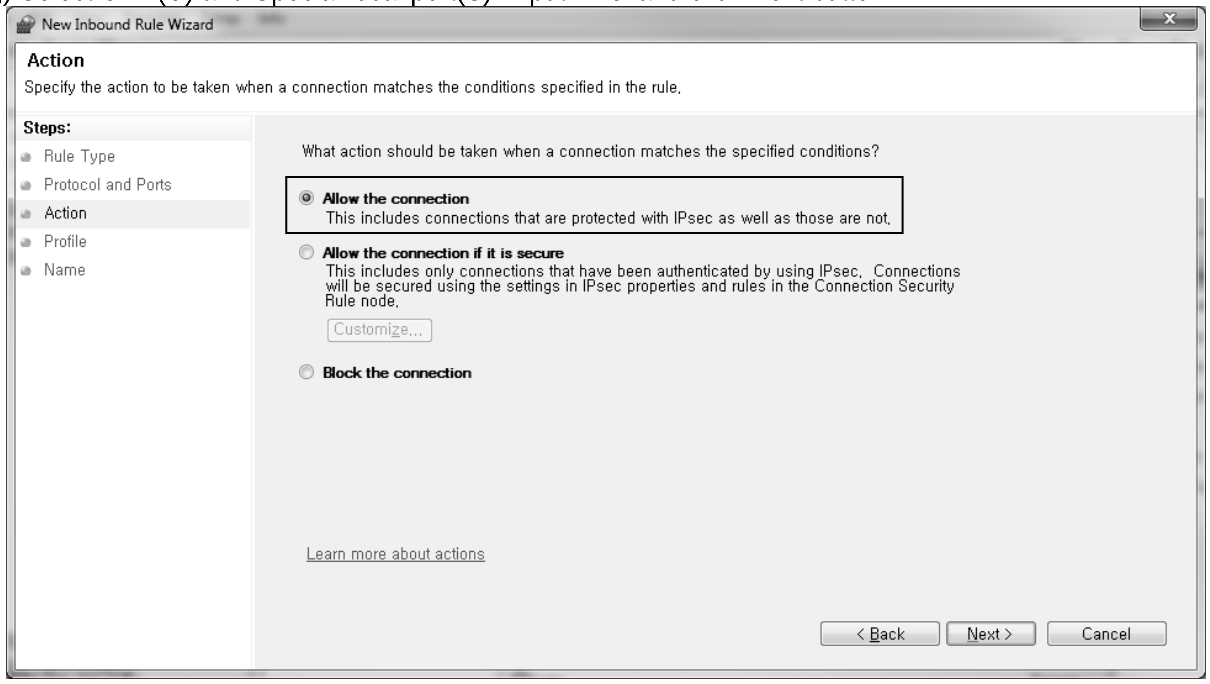


(e) Select the new rule in the top right.

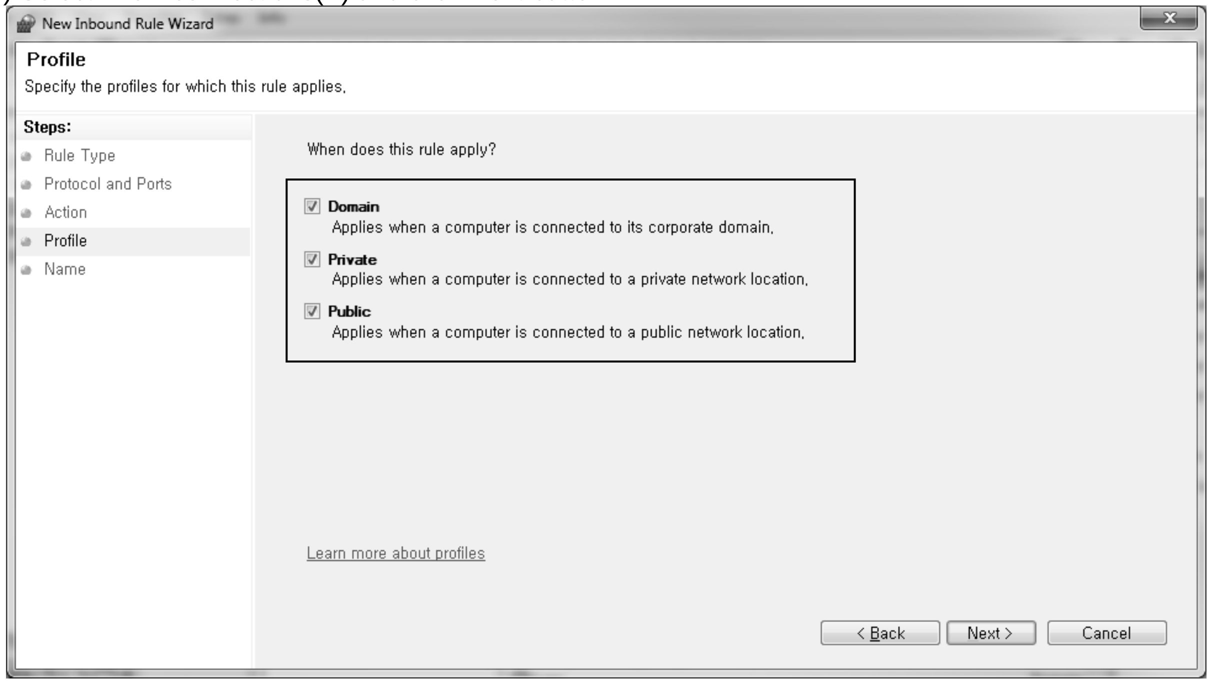


(f) Select the port and click Next button.

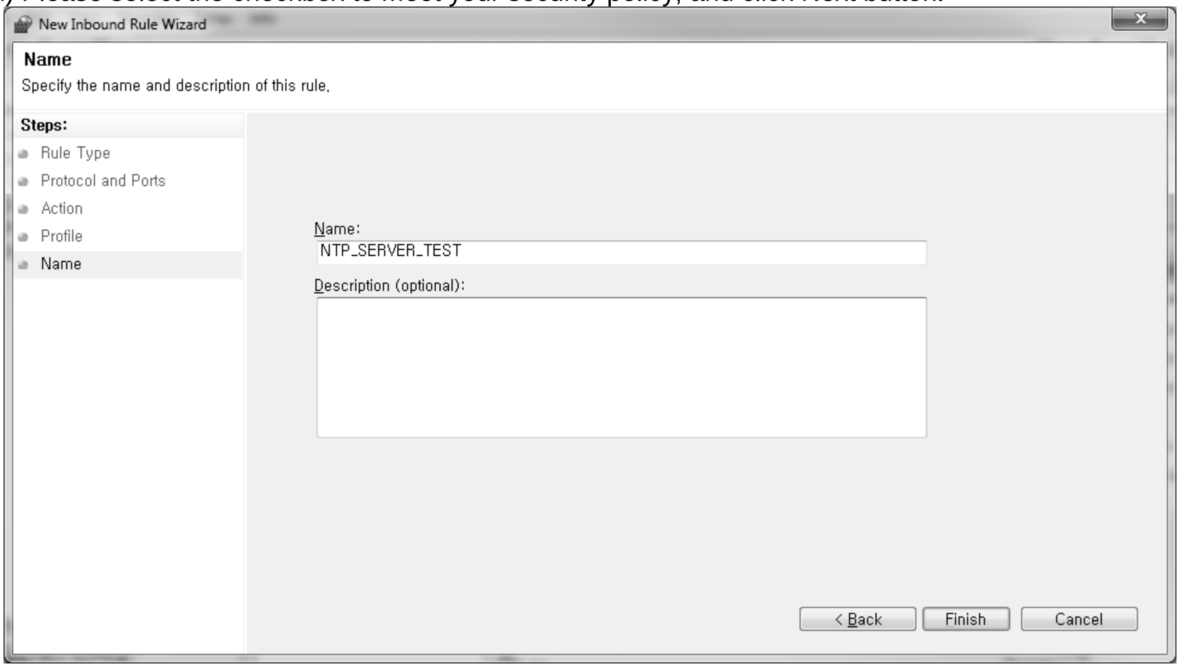
(g) Select UDP(U) and Special local port(S). Input ‘123’ and click Next button.



(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’)

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

Variable	Type	Address	Description
_NCPRM_ER	BOOL	%FX89	NC Parameter Error
_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
_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
_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
_EC_LINKUP_INFO	BOOL	%FX65600	EtherCAT Link Up/Down Information
_EC_COMM	BOOL	%FX65601	EtherCAT Communication connection state
_EC_COMM_ERR	BOOL	%FX65602	EtherCAT Communication timeout error
_EC_PDO_ERR_CNT	UINT	%FW4102	EtherCAT PDO error count
_EC_SLAVE_RDY	ARRAY[0..63] OF BOOL	%FX65664	EtherCAT Slave ready
_EC_SDO_BUSY	ARRAY[0..63] OF BOOL	%FX65792	EtherCAT Slave SDO processing busy
_EC_SDO_ERR	ARRAY[0..63] OF BOOL	%FX65920	EtherCAT Slave SDO processing error
_EC_LINE_FAIL	ARRAY[0..63] OF BOOL	%FX66048	EtherCAT Cable disconnection state
_EC_MASTER_STATE	BYTE	%FB8264	EtherCAT master STATE
_EC_SLAVE_NUM	WORD	%FW4133	Number of connected EtherCAT Slave
_EC_ERR_INFO1	STRING	%FB8272	EtherCAT error information1
_EC_ERR_INFO2	STRING	%FB8304	EtherCAT error information2
_EC_TRANSMITTED_OK	UDINT	%FD2084	EtherCAT Number of frames transmitted
_EC_RECEIVED_OK	UDINT	%FD2085	EtherCAT Number of frames received
_EC_CRCERR_CNT	UDINT	%FD2086	EtherCAT Receive CRC error frame
_EC_COLLISION_CNT	UDINT	%FD2087	EtherCAT Number of collision frames
_EC_CARRIER_SENSE_ERR	UDINT	%FD2088	EtherCAT Carrier sense error
_EC_LINKOFF_CNT	UDINT	%FD2089	EtherCAT Number of Link Off
_EC_OVERSIZE_FRAME	UDINT	%FD2090	EtherCAT Receive oversize frames
_EC_UNDERSIZE_FRAME	UDINT	%FD2091	EtherCAT Receive undersize frames
_EC_JABBER_FRAME	UDINT	%FD2092	EtherCAT Receive jabber frame
_EC_PDO_CUR_TRANSCYCLE	UDINT	%FD2093	EtherCAT PDO transfer cycle ns
_EC_PDO_MAX_TRANSCYCLE	UDINT	%FD2094	EtherCAT Maximum PDO transfer cycle ns
_EC_PDO_MIN_TRANSCYCLE	UDINT	%FD2095	EtherCAT Minimum PDO transfer cycle ns
_EC_PDO_TRANS_JITTER	UDINT	%FD2096	EtherCAT PDO frame transfer jitter ns

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.

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
_AXxx_COMM_ERR	BOOL	%FX73882	Axis xx communication alarm
_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

Variable	Type	Address	Description
_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
_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

Variable	Type	Address	Description
_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

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

Variable	Type	Address	Description
_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_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) Slave flag

Variable	Type	Address	Description
_SLVxx_EC_STATE	SINT	%FB47104	EtherCAT Slave xx STATE
_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

5) 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_RapidTrvsOpr	BOOL	%FX524736	NC Ch. yy rapid traverse operation
_NCyy_CuttingFeedOpr	BOOL	%FX524737	NC Ch. yy cutting feed operation
_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
_NCyy_RapidOverride	LREAL	%FL8207	NC Ch. yy rapid override

Variable	Type	Address	Description
_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_SpindleCVelAgr	BOOL	%FX525380	NC Ch. yy spindle command velocity reached signal
_NCyy_SpindleZeroVel	BOOL	%FX525381	NC Ch. yy spindle zero velocity reached signal
_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
_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

Variable	Type	Address	Description
_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_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_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_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_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

6) 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

Variable	Type	Address	Description
_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_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
_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
_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_AxErr	WORD	%FW33372	NC Ch. 01 axis X error code
_NC01X_DrvErr	WORD	%FW33373	NC Ch. 01 axis X drive error code

7) 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

8) 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

9) 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 speed
_ENC2_VEL	LREAL	%KL4	Encoder 2 speed

Appendix 2 Error Information & Solution

Here describes the information error types and its solutions.

(1) Function block error information

Error code		Error Description	Solutions
Hex	Dec		
0005	5	The current motion controller does not support the function block.	This command is not performed in the current version of the controller. Please contact customer support team of our company after checking the version in which the command can be executed.
0006	6	The axis number (Axis input) or encoder number (Encoder input) of the function block exceeded the allowable range.	Set the axis number to be 1~36. (Encode number: 1001~1002)
0007	7	The axis group number (AxesGroup input) of the function block exceeded the allowable range.	Set the group axis number to be between 1 and 16.
0008	8	The NC channel of the function block exceeded the allowable range.	Check the range of the NC channel and set it again.
0009	9	The slave number (Slave input) of the function block exceeded the allowable range.	Check the range of the slave number and set it again.
000B	11	The input of the function block exceeded the allowable range.	Check the input range of the function block and set it again.
000C	12	The array input of the function block exceeded the allowable range.	Check the size of the array input of the function block and set it again.
0012	18	Error block internal execution error occurred during execution of the function block.	The problem can arise in the current controller version. Please check the support version of XG5000 and controller.
0013	19	Motion response error occurred during execution of the function block.	The problem can arise in the current controller version. Please check the support version of XG5000 and controller.
0014	20	It exceeded the allowable range of the cam ID(CamTableID input) of the function block.	Check the range of the cam ID and set it again.

(2) System error information

Error code		Error Description	Solutions
Hex	Dec		
000E	14	System Error	Request for A/S if it occurs repeatedly even when the power is supplied again.
0017	23	Program Error	Start the program after modifying and re-loading the program
0018	24	IO Configuration Parameter Abnormality	Check the preservation status after uploading I/O parameter. If it is broken, modify and re-download it to check the operation. If there is still abnormality, exchange it.
0019	25	Basic Parameter Abnormality	Check the preservation status after uploading basic parameter. If it is broken, modify and re-download it to check the operation. If there is still abnormality, exchange it.
001D	29	Special Module Parameter Abnormality	Check the preservation status after uploading special module parameter. If it is broken, modify and re-download it to check the operation. If there is still abnormality, exchange it.
0027	39	CPU Abnormal Termination or Failure	System has been terminated abnormally due to noise and hardware abnormalities. 1) Request for A/S if it occurs repeatedly even when the power is supplied again. 2)Implement noise countermeasures
002B	43	Built-in Parameter-Encoder Abnormality	Check the preservation status after uploading built-in parameter. If it is broken, modify and re-download it to check the operation. If there is still abnormality, exchange it.
002C	44	Axis Parameter Abnormality	Modify the parameter and re-download it
002D	45	Axis Group Parameter Abnormality	Modify the parameter and re-download it
002E	46	EtherCAT Parameter Abnormality	Modify the parameter and re-download it
002F	47	NC Parameter Abnormality	Modify the parameter and re-download it
0030	48	NC Program Inspection Error	Check the program and re-download it
0032	50	Major Failure Detection Error of External Equipment	Repair the wrong equipment and restart it by referring to major failure detection error flag of the external equipment(depending on the parameter)
0038	56	Main Task Cycle Error	Check the main task cycle flag, re-download it after modifying the main task program or download it by increasing the main task cycle of the basic parameter
0039	57	Periodic Task Cycle Error	Check the periodic task cycle flag, re-download it after modifying the periodic task program or download it by increasing the periodic task cycle of the basic parameter
003A	58	Task Program Occupancy Rate Excess Error	1) Secure the time for the system internal service to operate by reducing the amount of user program execution in the main task/periodic task. 2) Secure the time for the system internal service to operate by setting the execution cycle of the main task/periodic task of the basic parameter to be higher
003B	59	Local Ethernet Parameter Inspection Error	Modify the parameter and re-download it
01F5	501	RTC Data Abnormality	Reset it using RTC clock function. If it occurs repeatedly, place requests for A/S

(3) Data log, SD additional function error information

Error code		Error Description	Solutions
Hex	Dec		
Overall Error Code			
0001	1	SD Card Recognition Error	Format it to FAT32 and connect to SD memory
0002	2	Partition Information Error	Format it to FAT32 and connect to SD memory
0003	3	File System Error	Format it to FAT32 and connect to SD memory
0004	4	Unsupported SD Card	Connect SD card with a capacity of 2GB to 32GB
0005	5	SD Card Capacity Check Error	The SD memory capacity test failed, and thus DS cannot be used. Replace SD memory or re-connect it after formatting
0006	6	SD Card Capacity Excess	SD memory capacity is used up, and data storage is not possible. Replace SD memory or re-connect it after formatting. If the available capacity is less than 20%
0007	7	Folder Creation Failed	Failed to create data log folder in SD. Replace SD memory or re-connect it after formatting
Error Code by Data Log Group			
1000	4096	Group x Folder Creation Error	Format it to FAT32 and connect to SD memory
2000	8192	Group x File Open Error	Format it to FAT32 and connect to SD memory
4000	16384	Group x File Write Error	Format it to FAT32 and connect to SD memory
SD Additional Function Error Code			
0001	1	File Error(File Open Failure, CRC Error)	Operate it after creating the file again
0002	2	Damaged File (Damages to Head and Tail, etc.)	Operate it after creating the file again
0005	5	No Password in File	The password is set in the PLC, but there is no password in the file stored in the SD card. Set the password and create a file.
0006	6	Password Mismatch	The password set in the PLC does not match the password of the file stored in the SD card. Confirm it again after checking the password.
0007	7	MAC Address Mismatch	The set MAC address does not match the MAC address of the PLC. Check the MAC address and reset it.
000A	10	No Saved File	There is no file saved in the SD card. Operate it after creating a file.
000B	11	PLC Mode Is RUN State	Check it after switching PLC mode to STOP.

(4) Analog error information

Error code		Error Description	Solutions
Hex	Dec		
0064	0100	Range setting error of input channel 0	Set the range that can be set
0065	0101	Range setting error of input channel 0	Set the range that can be set
00C8	0200	Filter value setting error of input channel 0	Set the filter value that can be set
00C9	0201	Filter value setting error of input channel 0	Set the filter value that can be set
012C	0300	Average value setting error of input channel 0	Set the average value that can be set
012D	0301	Average value setting error of input channel 0	Set the average value that can be set
0190	0400	Range setting error of output channel 0	Set the output range that can be set
0191	0401	Range setting error of output channel 0	Set the output range that can be set
01F4	0500	Input value setting error of output channel 0	Set the input value that can be set
01F5	0501	Input value setting error of output channel 0	Set the input value that can be set
0258	0600	Interpolation method range setting error of output channel 0	Set the interpolation method range that can be set
0259	0601	Interpolation method range setting error of output channel 0	Set the interpolation method range that can be set

(5) Motion error information

Error code		Error Description	Solutions
Hex	Dec		
0E00	3584	Command data range transmitted from XG5000 was out of the allowed value.	The problem can arise in the current controller version. Please check the support version of XG5000 and controller.
0E01	3585	The XG5000 test operation function cannot be executed if the controller is in the RUN state.	Execute the test operation of XG5000 after changing controller to STOP state.
0E02	3586	Cam data cannot be written if there is an axis in operation.	Write the cam data while all axes are not in operation.
0E03	3587	Encoder parameters cannot be written if there is an axis in operation.	Write the encoder parameters while all axes are not in operation.
0E04	3588	EtherCAT parameters cannot be written while EtherCAT communication is being connected.	Rewrite the EtherCAT parameters after disconnecting the EtherCAT communication.
0E10	3600	Encoder parameter data is abnormal.	Download the data again from XG5000 and place requests for A/S if the error occurs repeatedly after re-execution.
0E11	3601	Encoder 1 pulse input type of encoder parameter exceeded the setting range.	Set the encoder 1 pulse input of the encoder parameter to be between 0 and 5.
0E12	3602	Encoder 1 maximum value of encoder parameter was out of the range of pulse unit expression value.	Set the encoder 1 maximum value of the encoder parameter in the range of -2,147,483,648 to 2,147,483,647 when converted in pulse unit.
0E13	3603	Encoder 1 minimum value of encoder parameter was out of the range of pulse unit expression value.	Set the encoder 1 minimum value of the encoder parameter in the range of -2,147,483,648 to 2,147,483,647 when converted in pulse unit.
0E14	3604	Encoder 1 maximum value and minimum value of encoder parameters exceeded the range.	Set the encoder 1 minimum value of the encoder parameter to be smaller than the maximum value.
0E15	3605	Encoder 2 pulse input type of encoder parameter exceeded the range.	Set the encoder 2 pulse input of the encoder parameter to be between 0 and 5.
0E16	3606	Encoder 2 maximum value of encoder parameter was out of the range of pulse unit expression value.	Set the encoder 2 maximum value of the encoder parameter in the range of -2,147,483,648 to 2,147,483,647 when converted in pulse unit.
0E17	3607	Encoder 2 minimum value of encoder parameter was out of the range of pulse unit value.	Set the encoder 2 minimum value of the encoder parameter in the range of -2,147,483,648 to 2,147,483,647 when converted in pulse unit.
0E18	3608	Encoder 2 maximum value and minimum value of encoder parameters exceeded the range.	Set the encoder 2 minimum value of the encoder parameter to be smaller than the maximum value.
0E19	3609	Encoder input settings cannot be made above encoder settings of encoder parameter.	Confirm the encoder-related items of the encoder parameter and set values within the range.
0E1A	3610	The number of pulses per rotation of encoder 1 of encoder parameter exceeded the setting range.	Set the number of pulses per rotation of encoder 1 of encoder parameter to be greater than 0 and less than or equal to 4294967295.
0E1B	3611	The transfer distance per rotation of encoder 1 of encoder parameter exceeded the setting range.	Set the transfer distance per rotation of encoder 1 of encoder parameter to be more than 0.000000001 and less than or equal to 4294967295.
0E1C	3612	The number of pulses per rotation of encoder 2 of encoder parameter exceeded the setting range.	Set the number of pulses per rotation of encoder 2 of encoder parameter to be greater than 0 and less than or equal to 4294967295.
0E1D	3613	The transfer distance per rotation of encoder 2 of encoder parameter exceeded the setting range.	Set the transfer distance per rotation of encoder 2 of encoder parameter to be more than 0.000000001 and less than or equal to 4294967295.

Error code		Error Description	Solutions
Hex	Dec		
0E1E	3614	Encoder 1 input filter value of encoder parameter exceeded the setting range.	Set the encoder 1 input filter value of encoder parameter to a value between 0 and 6.
0E1F	3615	Encoder 2 input filter value of encoder parameter exceeded the setting range.	Set the encoder 2 input filter value of encoder parameter to a value between 0 and 6.
0E20	3616	Encoder 1 maximum value and minimum value of encoder parameters are set not to include the current position of encoder 1.	Set the range of encoder 1 minimum and maximum values of encoder parameters to include the current position of encoder 1. Or, in order to operate with the set parameters, change the current position of encoder to the value within the parameter by using the encoder preset command.
0E21	3617	Encoder 2 maximum value and minimum value of encoder parameters are set not to include the current position of encoder 2.	Set the range of encoder 2 minimum and maximum values of encoder parameters to include the current position of encoder 2. Or, in order to operate with the set parameters, change the current position of encoder to the value within the parameter by using the encoder preset command.
0E30	3632	EtherCAT parameter data is abnormal.	Download the data from XG5000 again and place requests for A/S if the error occurs repeatedly after re-execution.
0E31	3633	The periodic communication timeout count of the EtherCAT parameter exceeded the range.	Set the periodic communication timeout count of the EtherCAT parameter to be between 1 and 8.
0E32	3634	Error occurred during the EtherCAT parameter parsing	Check the EtherCAT parameters and set them again.
0E40	3648	The connection command cannot be executed beyond the EtherCAT parameter.	Check the EtherCAT parameter and set it again.
0E41	3649	The EtherCAT slave connect command is running.	Check if the EtherCAT slave connect command has been entered again while the EtherCAT slave connect command is running.
0E42	3650	The EtherCAT slave disconnect command is running.	Check whether the EtherCAT disconnect command has been entered again while the EtherCAT slave disconnect command is running.
0E43	3651	The connection/disconnection command cannot be executed due to mode switch.	Check whether the mode switch has been executed during the EtherCAT slave connection/disconnection command operation.
0E44	3652	The connection/disconnection command cannot be executed with the ESTOP command.	Check whether the ESTOP command was executed during the EtherCAT slave connection/disconnection command operation.
0E50	3664	The encoder preset command cannot be executed due to abnormal encoder parameter.	Confirm the encoder-related items of the encoder parameter, heck if they are set to values within the range and set the encoder parameter to normal values by using XG5000.
0E51	3665	The preset command cannot be executed because there is an axis that operates with the encoder as the main axis.	Check if the encoder preset command has been entered when there is an axis that operates with the encoder as the main axis.
0E52	3666	The encoder preset position was out of the range of maximum value or minimum value of the encoder.	Set the encoder preset position to a range that is greater than or equal to the minimum value, and smaller than or equal to the maximum value of the encoder.
0E53	3667	The encoder selection of encoder preset command exceeded the range.	Set the encoder selection to be between 0 and 1 (0: Encoder 1, 1: Encoder 2).
0E60	3680	The command cannot be executed due to the basic parameter data abnormality.	Download the basic parameter again from XG5000 and place requests for A/S if 'the basic parameter error' occurs repeatedly after re-execution.

Error code		Error Description	Solutions
Hex	Dec		
0E61	3681	Cam data is abnormal.	Download the data again from XG5000 and place requests for A/S it is occurs repeatedly after re-execution.
0F00	3840	It failed to change to EtherCAT INIT state.	Check the communication cable status and slave operation status (power-on and error occurrence). And check whether the communication cable is exposed to noise.
0F06	3846	It is EtherCAT INIT state initialization (DC_INIT) error.	Check the communication cable connection status and slave operation status(power-on and error occurrence). And check whether the communication cable is exposed to noise.
0F09	3849	There is no EtherCAT slave connected to the controller.	Check whether the communication cable between the controller and the EtherCAT slave is properly installed, the power is normally applied to the EtherCAT slave, or the communication cable is exposed to noise if there is a slave connected to the controller.
0F0A	3850	It exceeded the maximum number of connected slaves.	Make sure that there are not more than 64 EtherCAT slaves connected to the controller.
0F0E	3854	There is a difference in node ID and EtherCAT parameter settings of the EtherCAT slave.	Check whether the order of the network cable connection between the controller and the EtherCAT slaves matches the EtherCAT parameter settings.
0F0F	3855	There is an error in setting node ID of the EtherCAT slave.	Check whether there are duplicate node IDs or errors in settings.
0F10	3856	It failed to change to EtherCAT PREOP stage.	Check the communication cable connection status and slave operation status (power-on and error occurrence). And check whether the communication cable is exposed to noise.
0F1E	3870	There is no slave setting data of EtherCAT parameter.	Set the slave of the EtherCAT parameter using XG5000, and then send the EtherCAT parameter to the controller.
0F1F	3871	The slave setting data of the EtherCAT parameter and the connected slave are different.	Set the slave setting of the EtherCAT parameter to match the actually connected slave information. The slave of the EtherCAT parameter can be set automatically using the 'EtherCAT Slave Auto Connection' function in XG5000.
0F20	3872	It failed to change to EtherCAT SAFEOP state.	Check the communication cable connection status and slave operation status (power-on and error occurrence). And check whether the communication cable is exposed to noise.
0F30	3888	It failed to change to EtherCAT OP state.	Check the communication cable connection status and slave operation status (power-on and error occurrence). And check whether the communication cable is exposed to noise.
0F40	3904	It failed to change from EtherCAT OP state to INIT state.	Check the communication cable connection status and slave operation status (power-on and error occurrence). And check whether the communication cable is exposed to noise.
0F50	3920	There is no response in a communication connection state.	Check the communication cable connection status and slave operation status (power-on and error occurrence). And check whether the communication cable is exposed to noise.
0F51	3921	Periodic communication error occurred. (Communication error that exceeds the periodic communication timeout count of master parameter occurs)	Check whether servo power is off during communication, communication cable is properly installed, or communication cable is exposed to noise.

Error code		Error Description	Solutions
Hex	Dec		
0F60	3936	The slave device address (Adp) setting value of ESC read command exceeded the range.	Check the slave device address (Adp) range according to the EtherCAT command code(EcatCmd) setting value to set it.
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 ~ 4 (BYTE).
0F62	3938	EtherCAT command code(EcatCmd) setting value of the ESC read command was incorrect.	Set the EtherCAT command code to one among 1(APRD), 4(FPRD) and 7(BRD).
0F63	3939	There was no response from the slave device for the ESC read command.	Check whether the salve device designated as Adp is installed properly, or the Ado address value is in the read-permission area.
0F70	3952	The slave device address (Adp) setting value of ESC write command exceeded the range.	Check the slave device address (Adp) range according to the EtherCAT command code(EcatCmd) setting value to set it.
0F71	3953	The data size setting value of the ESC write command exceeded the range.	Set the data size setting value of the ESC read command to 1 ~ 4 (BYTE).
0F72	3954	EtherCAT command code(EcatCmd) setting value of the ESC write command was incorrect.	Set the EtherCAT command code to one among 2(APWR), 5(FPWR) and 8(BWR).
0F73	3955	There was no response from the slave device for the ESC write command.	Check whether the salve device designated as Adp is installed properly, or the Ado address value is in the read-permission area.
0F74	3956	You cannot write the specified area to the ESC address(Ado) during execution of the communication connection/disconnection command or in the communication connection state.	Set the ESC address (Ado) that can be written during execution of the communication connection/disconnection command or in the communication connection state.
0FF2	4082	The normal operation in relation to encoder input cannot be executed due to controller H/W abnormality.	Place requests for A/S if it occurs repeatedly when the power is supplied again.
1000	4096	The axis is not ready for driving. (It is not connected to EtherCAT network.)	Execute the command when the axis is ready for operation.
1001	4097	It cannot be executed in "Disabled" state.	Check the operable axis status of the command and execute the command when the command can be run.
1002	4098	It cannot be executed in "Standstill" state.	Check the operable axis status of the command and execute the command when the command can be run.
1003	4099	It cannot be executed in "Discrete" state.	Check the operable axis status of the command and execute the command when the command can be run.
1004	4100	It cannot be executed in "Continuous" state.	Check the operable axis status of the command and execute the command when the command can be run.
1005	4101	It cannot be executed in "Synchronized" state.	Check the operable axis status of the command and execute the command when the command can be run.
1006	4102	It cannot be executed in "Homing" state.	Check the operable axis status of the command and execute the command when the command can be run.
1007	4103	It cannot be executed in "Stopping" state.	Check the operable axis status of the command and execute the command when the command can be run.
1008	4104	It cannot be executed in "Errorstop" state.	Check the operable axis status of the command and execute the command when the command can be run.
100A	4106	Motion command cannot be executed when the belonging axis group is active.	Execute the command after changing the axis group to GroupDisabled state with the axis group disable command.
100B	4107	This command cannot be given to a virtual axis.	The command cannot be executed on a virtual axis. Check whether the command is executed on the virtual axis.

Error code		Error Description	Solutions
Hex	Dec		
100C	4108	The command cannot be executed if it is registered as an NC channel/axis and is in NC control operation.	Check the operable axis status of the command and execute the command when the command can be run.
100D	4109	The command cannot be executed because the axis is not enabled.	Check whether the setting axis of the command is registered in axis parameter. The axis can be registered in the axis parameter among the motion data items of XG5000.
100E	4110	It is changed to 'run' state during execution of motion test run command, and thus the operation cannot continue.	Check whether the controller was changed to the 'run' state while the axis is running.
100F	4111	The axis operation cannot continue because the controller is stopped by the ESTOP command.	Check whether the controller was stopped by the ESTOP command during the axis operation.
1010	4112	The controller was changed to 'Stop' or 'Error' state, and thus operation cannot continue.	Check whether the controller has been changed to 'Stop' or 'Error' state while the axis is running.
1011	4113	The EtherCAT network connection was lost, and thus operation cannot continue.	Check whether the EtherCAT network connection has been disconnected due to slave power supply error, network cable error and noise inflow on network cable while the axis is running.
1012	4114	The position setting value of the command was out of the range of pulse unit expression value.	It exceeded a 32-bit area when the command position value was converted in pulse unit. Set the value in the range of -2,147,483,648 to 2,147,483,647 when converting the command position value to pulse.
1013	4115	The operation speed value was less than 0, or it exceeded the maximum speed value.	Set the operation speed value to be larger than 0 and less than the maximum speed value set in the axis.
1014	4116	The acceleration was set as the negative number.	Set the acceleration value to a value greater than or equal to 0.
1015	4117	The deceleration was set as the negative number.	Set the deceleration value to a value greater than or equal to 0.
1016	4118	The jerk was set as the negative number.	Set the jerk value to a value greater than or equal to 0.
1017	4119	The direction specification exceeded the range.	Check the range of the direction setting value of the command and set the value within the range. (Refer to Chapter 6 Commands and Functions)
1018	4120	The torque setting value exceeded the range.	Set the torque setting value within 1000%.
1019	4121	The torque ramp setting value exceeded the range.	Set the torque ramp setting value to a value greater than or equal to 0.
101A	4122	Buffer Mode setting value exceeded the input range.	Set value (0 ~ 5) that can be set in the Buffer Mode.
101B	4123	Execution Mode setting value exceeded the input range.	Set value (0 ~ 1) that can be set in the Execution Mode.
101C	4124	Tracking error-over range occurred, and thus operation cannot continue.	Deviation between command position and current position exceeded 'Tracking error-over range value'. To prevent an alarm from occurring, tune the servo drive or set the 'Tracking error-over range value' to a larger value.
101D	4125	Tracking error-over range occurred.	Deviation between command position and current position exceeded 'Tracking error-over range value'. To prevent an alarm from occurring, tune the servo drive or set the 'Tracking error-over range value' to a larger value.

Error code		Error Description	Solutions
Hex	Dec		
101F	4127	The command position value transmitted to the servo driver was out of the range of pulse unit expression value.	It exceeded a 32-bit area when the command position value was converted in pulse unit. Set the value in the range of -2,147,483,648 to 2,147,483,647 when converting the command position value to pulse.
1020	4128	It is the undefined axis command.	This command is not performed in the current version of the controller. Please contact customer support team of our company after checking the version in which the command can be executed.
1021	4129	The same command was executed, and thus the previously executed command was canceled.	The motion command can be executed only once per scan. Change the operation condition of the program so that one motion command can be executed in one scan.
1022	4130	It exceeded the number of commands that can execute a Buffered command.	The command cannot be executed because the command buffer of the axis is full. The number of commands that can be executed with the Buffered command is 100. Adjust the timing of the command execution.
1030	4144	Axis parameters cannot be written when the axis is in operation.	Perform parameter writing when the axis is not in operation.
1040	4160	Axis parameter data is abnormal.	Download the data again from XG5000 and place requests for A/S if the error occurs repeatedly after re-execution.
1041	4161	It is not possible to execute operation due to axis parameter error.	Check the axis parameter and set it again.
1042	4162	The speed limit of the axis parameter cannot be set to the value less than 0.	Set the speed limit of the basic parameter to a value greater than 0.
1043	4163	The soft upper/lower limit value of axis parameter exceeded the range.	Set the soft upper limit value of the axis parameter to be greater than or equal to the soft lower limit value.
1044	4164	The current speed filter time constant value of axis parameter exceeded the range.	Set the parameter setting value to 0 ~ 100.
1045	4165	The error reset monitoring time of axis parameter exceeded the range.	Set the parameter setting value to 1 ~ 1000.
1046	4166	The setting value of transfer distance per rotation exceeded the range.	Set the parameter setting value to more than 0.000000001 and less than 4294967295.
1047	4167	The setting value of infinite length repeat position exceeded the range.	Set the parameter setting value to more than 0 and less than 2,147,483,647 in pulse unit.
1048	4168	The setting value of in-position width exceeded the range.	Set the parameter setting value to more than 0 and less than 2,147,483,647 in pulse unit.
1049	4169	The setting value of tracking error-over range exceeded the range.	Set the parameter setting value to more than 0 and less than 2,147,483,647 in pulse unit.
104A	4170	The setting value of current position display compensation amount exceeded the range.	Set the parameter setting value to more than 0 and less than 2,147,483,647 in pulse unit.
104B	4171	The setting value of jog high speed exceeded the range.	Set the parameter setting value to more than 0, larger than the jog low speed value and less than the speed limit.
104C	4172	The setting value of jog low speed exceeded the range.	Set the parameter setting value to more than 0, smaller than the jog high speed value and less than the speed limit.
104D	4173	The setting value of jog acceleration exceeded the range.	Set the parameter setting value to more than 0.

Error code		Error Description	Solutions
Hex	Dec		
104E	4174	The setting value of jog deceleration exceeded the range.	Set the parameter setting value to more than 0.
104F	4175	The setting value of jog jerk exceeded the range.	Set the parameter setting value to more than 0.
1050	4176	Motor-side gear ratio setting value exceeded the range.	Set the parameter setting value to 1 ~ 65535.
1051	4177	Machine-side gear ratio setting value exceeded the range.	Set the parameter setting value to 1 ~ 65535.
1052	4178	The setting value for the number of pulses per rotation exceeded the range.	Set the parameter setting value to be greater than 0 and less than or equal to 4294967295.
1053	4179	Connection device setting value exceeded the range.	Set the device number of the slave that can be supported. The node setting range is 0(no connection device), and 1 ~ 64.
1054	4180	Axis type setting value exceeded the range.	Set the parameter setting value to '0: actual axis' or '1: virtual axis'.
1055	4181	Speed command unit setting value exceeded the range.	Set the parameter setting value from '0: unit/sec', '1: unit/min', '2: rpm'.
1060	4192	Servo On cannot be executed due to occurrence of servo drive errors.	Execute Servo On after checking the error factor of the servo drive and removing the servo drive error.
1061	4193	Servo On command was executed again in the middle of processing Servo On.	Check whether the Servo On command was performed again in the middle of processing Servo On in program or XG5000.
1062	4194	It is not possible to complete Servo On because the servo drive cannot be changed to "ReadyToSwitchON" state.	Check the status of the servo drive. The Servo On command may not be executed in certain circumstances.
1063	4195	It is not possible to complete Servo On because the servo drive cannot be changed to "Switched on" stage.	Check the status of the servo drive. The Servo On command may not be executed in certain circumstances.
1064	4196	It is not possible to complete Servo On because the servo drive cannot be changed to "Operation enabled" state.	Check the status of the servo drive. The Servo On command may not be executed in certain circumstances.
1065	4197	It is not possible to complete Servo On because "Quick Stop" of servo drive is enabled.	Check the status of the servo drive. The Servo On command may not be executed in certain circumstances.
1066	4198	Servo Off command was executed again in the middle of processing Servo Off.	Check whether the Servo Off command was performed again in the middle of processing Servo Off in program or XG5000.
1067	4199	The execution of Servo Off command was not completed.	Check the status of the servo drive.
1070	4208	It exceeded the servo error reset monitoring time.	The servo drive error has not been cleared even after the error reset monitoring time set in the axis parameter went by. Execute the error reset command again after removing the servo drive error factor.
1080	4224	Commands that use absolute coordinates cannot be executed at absolute coordinates of the state of undetermined origin.	Execute the absolute coordinate operation command after making the determined origin state with homing command or current position setting command.
1081	4225	In infinite length repeat enable state, the target position was beyond the range of infinite length repeat position from the direction specification.	Set the target position within the infinite length repeat position from the direction specification.

Error code		Error Description	Solutions
Hex	Dec		
1082	4226	SuperImposed command cannot be executed during operation with speed control or torque control.	Execute the SuperImposed command when it is not in operation with speed control or torque control.
1083	4227	SuperImposed operation stop command cannot be executed when the SuperImposed operation is not working.	Execute the SuperImposed operation stop command when the SuperImposed operation is in progress.
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 to more than the soft lower limit value of the extended parameter and less than the soft upper limit value.
1091	4241	The current position change command cannot be executed in case of operation with homing, speed synchronization, cam and torque control.	Execute the current position change when the axis is not in operation of among one of homing, speed synchronization, cam and torque control.
10A0	4256	Servo drive does not support torque control mode.	Perform torque control by using serve drive that supports CST mode of EtherCAT CoE.
10A1	4257	There is no target torque object (0x6071) setting that can execute the torque control on RxPDO entry setting in slave data of the EtherCAT parameter.	Set the target torque object (0x6071) that supports torque control to the RxPDO entry of the EtherCAT parameter slave data in XG5000, and then send it to the controller.
10B0	4272	Servo drive does not support homing mode.	Perform homing by using servo drive that supports homing mode of EtherCAT CoE.
10B1	4273	An error occurred during the execution of the homing in the servo drive.	Check the error factor of the servo drive and perform homing after removing the servo drive error.
10B2	4274	The homing command cannot be executed when axis is running.	Perform the homing command again in the standstill state after the axis stops.
10C0	4288	The override command cannot be executed when the position /speed control is not in operation.	Execute override command during operation with position control or speed control.
10C1	4289	The override factor of the override command exceeded the range.	Set the VelFactor, AccFactor and JerkFactor values of the override command to be 0 or more and then execute the override command.
10C2	4290	The operation speed value exceeded the maximum speed value after reflecting the factor of the override command.	Perform override within the range that does not exceed the maximum speed value of the axis.
10D0	4304	The gear ratio denominator value cannot be zero.	Execute the command after setting the gear ratio denominator to a value other than 0.
10D1	4305	Gear operation MasterValueSource setting value exceeded the range.	Execute the command after setting the MasterValueSource input value to a value between 0 and 1.
10D2	4306	Gear operation main axis setting was out of the range.	Execute the command after setting the main axis in the area of 1~36 (axis) and 1001~1002(encoder). Check whether the VarOffset value exceeds the memory area if the main axis is a variable.
10D3	4307	The gear operation main axis setting is identical with the subordinate axis.	Execute the command after setting the main axis to the different axis from the subordinate axis(command axis).
10D4	4308	The gear operation main axis was not ready.	Execute the command when the main axis is ready.

Error code		Error Description	Solutions
Hex	Dec		
10D5	4309	In case the gear operation main axis is set as an encoder, the gear operation command cannot be executed if the encoder parameter setting error of the built-in parameter occurs.	Confirm the encoder-related items of the encoder parameters, check if they are set to values within the range and set the encoder parameters to normal values by using XG5000.
10D6	4310	The MC_GearInPos command cannot be executed when the main axis is running with torque control.	Execute the MC_GearInPos command while the main axis is not operating in torque control.
10D7	4311	The speed of the gear operation subordinate axis exceeded the speed limit.	Reduce the speed of the main axis or change the gear ratio lest the speed of the subordinate axis in gear operation should exceed the speed limit set on the subordinate axis.
10D8	4312	The gear release command cannot be executed if it is not gear operation.	The gear release command can be used only when the gear is in operation.
10D9	4313	The command cannot be executed because the target speed setting value of MC_GearInPos command is smaller than the current operation speed or the gear operation speed.	Execute the command after setting the target speed setting value of MC_GearInPos command to the current operation speed or gear operation speed or more.
10DA	4314	It will not be able to reach the subordinate axis synchronization position within the time when the main axis operates to the main axis synchronization position during the MC_GearInPos operation.	Execute the command after increasing the target speed setting value MC_GearInPos command or adjusting MasterStartDistance so that the subordinate axis moves to the subordinate axis synchronization position within the time when the main axis operates to the main axis synchronization position.
10DB	4315	The synchronous operation command (gear, cam, etc.) cannot be executed if the main axis is in homing operation.	Execute the synchronous operation (gear, cam, etc.) when the main axis is not in homing operation.
10E0	4320	There are no object settings in the slave data of the EtherCAT parameter than enable the touch probe to PDO entry settings.	Set the object that supports touch probe to the PDO entry of the EtherCAT parameter slave data in XG5000 and then send it to the controller.
10E1	4321	TriggerInput input setting value was out of the range.	Set the TriggerInput setting value to 0(Touch Probe1) or 1(Touch Probe2).
10F0	4336	The parameter number setting value of the parameter read/write command was out of the range.	Execute the command after setting the parameter number setting value of parameter read/write command to be between 0~28, 100~106 and 200~206.
10F1	4337	The data setting value of the set parameter of the parameter write command was out of the range.	Check the data setting range of the parameter to be set.
10F2	4338	The parameter cannot be changed because the maximum value of encoder 1 is out of the pulse unit expression value when the encoder parameter is changed.	Change the maximum value of encoder 1 in advance to prevent errors when converted in pulse unit, and then change the parameter.
10F3	4339	The parameter cannot be changed because the minimum value of encoder 1 is out of the pulse unit expression value when the encoder parameter is changed.	Change the minimum value of encoder 1 in advance to prevent errors when converted in pulse unit, and then change the parameter.
10F4	4340	The parameter cannot be changed because the maximum value of encoder 2 is out of the pulse unit expression value when the encoder parameter is changed.	Change the maximum value of encoder 2 in advance to prevent errors when converted in pulse unit, and then change the parameter.

Error code		Error Description	Solutions
Hex	Dec		
10F5	4341	The parameter cannot be changed because the minimum value of encoder 2 is out of the pulse unit expression value when the encoder parameter is changed.	Change the minimum value of encoder 2 in advance to prevent errors when converted in pulse unit, and then change the parameter.
1100	4352	The jog operation command cannot be executed when the axis is running.	Execute the jog command when the axis is in stop state.
1110	4368	There is an error in the cam operation MasterScaling input value.	You cannot put 0 in the MasterScaling input value.
1111	4369	There is an error in the cam operation MasterStartDistance input value.	Set the MasterStartDistance input value to a value greater than 0 and execute the command.
1112	4370	There is an error in the cam operation MasterSyncPosition input value.	Set the MasterSyncPosition input value to a value greater than 0 and execute the command.
1113	4371	The cam operation StartMode input value exceeded the range.	Set the StartMode input value to a value between 0 and 1 and execute the command.
1114	4372	The cam operation MasterValueSource input value exceeded the range.	Set the MasterValueSource input value to a value between 0 and 1 and execute the command.
1115	4373	The specified cam table does not exist.	Adjust the cam table number to a valid cam table number and execute the command.
1116	4374	The cam operation main axis setting was out of the range.	Execute the command after setting the main axis in the area of 1~36 (axis) and 1001~1002(encoder). Check whether the VarOffset value deviates from the memory area if the main axis is a variable.
1117	4375	The cam operation main axis setting is identical with the subordinate axis.	Execute the command after setting the main axis to the different axis from the subordinate axis(command axis).
1118	4376	The cam operation main axis was not ready.	Execute the command when the main axis is ready.
1119	4377	In case the cam operation main axis is set as an encoder, the command cannot be executed if the encoder parameter error of the built-in parameter occurs.	Confirm the encoder-related items of the encoder parameter check if they are set to values within the range and set the encoder parameter to normal values by using XG5000.
111A	4378	The speed of the cam operation main axis exceeded the speed limit.	Reduce the speed of the main axis or adjust the cam table lest the speed of the subordinate axis in cam operation should exceed the speed limit set on the subordinate axis.
111B	4379	The cam release command cannot be executed if it is not cam operation.	The cam release command can be used only when the cam is in operation.
111C	4380	The value for setting the number of cam data of the cam data write command exceeded the range.	Set the value for setting the number of cam data of the cam data write command to less than the number of registered cam points. Add a cam table if the cam table is not registered.
111D	4381	The specified cam table data of the cam data read command is abnormal.	Reset the cam data and place requests for A/S if it occurs again after re-execution.
111E	4382	The cam skip command cannot be executed if the cam is not in operation.	Execute the cam skip command when the cam is running.
111F	4383	The number of cam cycles to be skipped by the cam skip command was set to 0.	Set the number of cam cycles to be skipped by the cam skip command to be greater than 0.
1121	4385	The skip mode setting value of the cam skip command exceeded the range.	Execute the command after setting the skip mode setting value of the cam skip command to a value between 0 and 2.

Error code		Error Description	Solutions
Hex	Dec		
1122	4386	The cam table is not registered.	Register the cam table or set the data again to execute the command.
1123	4387	The cam data of the cam data write command is abnormal.	Set the data of the cam data write command correctly.
1124	4388	Cam main axis value does not exist within the specified range.	Check the MasterStartPos and MasterEndPos values and run the command again.
1130	4400	The phase correction command cannot be executed if the command axis is not InSync or InGear status of synchronous control(cam, gear operation)operation.	Execute the phase correction command when the command axis of the phase correction command is in synchronous control operation and InSync or InGear state.
1131	4401	There is an error in setting the main axis of the phase correction command.	Execute the command after setting the main axis of the phase correction command to the same as the actual axis of the current synchronous operation.
1132	4402	The phase correction amount of the phase correction command was out of the position expression range.	Execute the command after setting the phase correction amount so that the phase correction amount is within the range from more than -2,147,483,648 to less than 2,147,483,647 in pulse unit.
1133	4403	The speed setting value of the phase correction command was out of the range.	Execute the command after setting the speed value of the phase correction command to be larger than 0 and less than the speed limit of the main axis.
1140	4416	The connected slave device does not support the speed control mode.	Perform the speed control by using the slave device that supports the velocity mode of the EtherCAT CoE.
1150	4432	The connected slave device does not support the position control mode.	Perform the position control by using the slave device that supports the CSP mode of the EtherCAT CoE.
1200	4608	The hardware upper limit error occurred.	Remove the error by executing the error reset command after breaking away from the external upper limit signal range with the use of reverse jog command.
1201	4609	The hardware lower limit error occurred.	Remove the error by executing the error reset command after breaking away from the external lower limit signal range with the use of forward jog command.
1203	4611	The command cannot be executed due to servo drive error during operation.	Remove the servo error with the error reset command after eliminating the servo error factor.
1204	4612	The command cannot be executed due to servo-off during operation.	Re-execute the command after changing the command axis to servo-on state with the servo-on command.
1205	4613	The software upper limit error occurred.	Remove the error by executing the error reset command after breaking away from the software upper limit range with the use of reverse jog command.
1206	4614	The software lower limit error occurred.	Remove the error by executing the error reset command after breaking away from the software lower limit range with the use of forward jog command.
1F00	7936	The periodic communication error occurred. (The communication error exceeding the master parameter periodic communication timeout count occurred)	Check whether servo power is off during communication, communication cable is normally installed and communication cable is exposed to noise.

Error code		Error Description	Solutions
Hex	Dec		
1F10	7952	SDO commands can no longer be executed due to the SDO processing failure of slave device that was performed previously	Reset the connection after checking whether the status of the slave device is normal.
1F11	7953	The SDO parameter write command cannot be executed during operation.	Execute the SDO parameter write command when the axis is not in operation.
1F12	7954	The range of data such as SDO parameter Index, subIndex and etc. was out of the allowed value.	Execute the SDO parameter write command after setting the SDO parameter Index to 0x0000~0x9FFF, SubIndex to 0x00~0xFF and data size within 4 words.
1F13	7955	Abort occurred during SDO parameter write command.	Stop of writing operation was done in the middle of writing SDO parameter in slave device. Check the writting data and the status of slave device.
1F14	7956	There is no response of the slave device regarding the command to write SDO parameter.	There is no response from slave device in the middle of writing SDO parameter. Check the status of slave device.
1F16	7958	Abort occurred while saving SDO parameter EEPROM.	Cancelation was done in the middle of saving SDO parameter EEPROM in slave device. Check the status of slave device.
1F17	7959	There is no response of the slave device regarding the parameter to save SDO parameter EEPROM.	There is no response from slave device in the middle of saving SDO parameter EEPROM. Check the status of slave device.
1F19	7961	Other commands cannot be executed while writing SDO parameter or saving SDO parameter EEPROM.	Execute other commands after saving SDO parameter EEPROM.
1F20	7968	Abort occurred in the middle of the command to write SDO parameter.	Stop of writing operation was done in the middle of writing SDO parameter in slave device. Check the writting data and the status of slave device.
1F21	7969	There is no response of the slave device regarding the command to read SDO parameter.	There is no response from slave device in the middle of reading SDO parameter EEPROM. Check the status of slave device.
1F22	7970	The SDO parameter read/write commands cannot be executed while the SDO parameter read/write commands are being executed.	Execute the command after the currently executed SDO parameter read/write is completed.
1F33	7987	It failed to change the operation mode of the servo drive to the position control(CSP) mode	Confirm whether the servo drive supports EtherCAT CoE CSP mode and check the status of servo drive.
1F34	7988	It failed to change the operation mode of the servo drive to the homing mode	Confirm whether the servo drive supports EtherCAT CoE Homing mode and check the status of servo drive.
1F35	7989	It failed to change the operation mode of the servo drive to the torque control (CST) mode.	Confirm whether the servo drive supports EtherCAT CoE CST mode and check the status of servo drive.
1F36	7990	It failed to change the operation mode of the servo drive to the speed control (CSV) mode.	Confirm whether the servo drive supports the EtherCAT CoE CSV mode and check the status of the servo drive.
1F50	8016	The XG5000 manual tuning function cannot be executed in case the controller is in the RUN state.	Perform manual tuning of XG5000 after changing the controller to STOP state.
1F60	8032	It is the command that is not available in the operation mode of the current slave.	Execute the command after setting the slave to the Boot state.
1F61	8033	Transmission timeout occurred during file transfer.	Check the status of the transmission line or slave and execute the command.
1F62	8034	Receive timeout occurred during file transfer.	Check the status of the transmission line or slave and execute the command.

Error code		Error Description	Solutions
Hex	Dec		
1F63	8035	Packet error occurred during file transfer.	Check the status of the transmission line or slave and execute the command.
1F64	8036	There is a memory shortage in slave.	Check the transferred file and execute the command.
1F65	8037	The device does not exist.	Check whether the FOE function is available slave and execute the command.
1F66	8038	Access to the slave is defied.	Check whether the FOE function is available slave and execute the command.
1F67	8039	The password does not match.	Check the password and execute the command.
1F68	8040	Data to be downloaded by the FoE function was not transferred to the controller.	Check the communication cable connection status and controller operation status.
1F6F	8047	There was a slave error during file transfer.	Remove the slave error and execute the command.
1170	4464	PartLength value is invalid.	The PartLength cannot be smaller than 0.
1171	4465	Circumference value is invalid.	The Circumference cannot be smaller than 0.
1172	4466	The value for the cutting start position is invalid.	The cutting start position cannot be smaller than 1/4 of the Circumference.
1173	4467	The value for the cutting end position is invalid.	The cutting end position cannot be bigger than 1/4 of the Circumference or bigger than the cutting start position.
1174	4468	The value for the synchronization speed ratio is invalid.	The synchronization speed ratio value must be between 50-200.
1175	4469	The ratio for the 0 speed region is invalid.	The ratio for the 0 speed region must be between 0-50.
1176	4470	The value for the cam profile type is invalid.	Change the cam profile type and run the command.
1177	4471	The value for the cam point number is invalid.	Change the cam point number and run the command.
1178	4472	The value for the cam curve type is invalid.	Change the cam curve type and run the command.
1179	4473	The cutting region is too wide.	Change the length of the cutting region or the speed ratio and run the command.
2000	8192	The axis group was not ready for operation.	Execute the command when the axis group is ready for operation.
2001	8193	The axis group cannot be executed in "Disabled" state.	Check the operable axis status of the command and execute the command when the command can be run.
2002	8194	The axis group cannot be executed in "Standby" state.	Check the operable axis status of the command and execute the command when the command can be run.
2003	8195	The axis group cannot be executed in "Moving" state.	Check the operable axis status of the command and execute the command when the command can be run.
2004	8196	The axis group cannot be executed in "Homing" state.	Check the operable axis status of the command and execute the command when the command can be run.
2005	8197	The axis group cannot be executed in "Stopping" state.	Check the operable axis status of the command and execute the command when the command can be run.
2006	8198	The axis group cannot be executed in "Errorstop" state.	Check the operable axis status of the command and execute the command when the command can be run.
2007	8199	The axis configuration of the axis group is not servo-on state.	Check the operable axis status of the command and execute the command when the command can be run.
200F	8207	The axis group operation cannot continue because the controller is stopped by the ESTOP command.	Check whether the controller was stopped by the ESTOP command during the axis group operation.
2010	8208	The controller was changed to 'Stop' or 'Error' state, and thus operation cannot continue.	Check whether the controller was changed to 'Stop' or 'Error' state during operation of the axis group.

Error code		Error Description	Solutions
Hex	Dec		
2011	8209	The EtherCAT network connection was lost, and thus operation cannot continue.	Check whether the EtherCAT network connection has been disconnected due to slave power supply error, network cable error and noise inflow on network cable during operation of the axis group.
2012	8210	The position setting value of the command was out of the pulse unit expression value.	It exceeded a 32-bit area when the command position value was converted in pulse unit. Set the value in the range of -2,147,483,648 to 2,147,483,647 when converting the command position value to pulse.
2013	8211	The operation speed value was less than 0, or it exceeded the maximum speed value.	Set the operation speed value to a value that is greater than 0 and less than or equal to the maximum speed value set in the axis group.
2014	8212	The acceleration was set as the negative number.	Set the acceleration value to a value greater than or equal to 0.
2015	8213	The deceleration was set as the negative number.	Set the deceleration value to a value greater than or equal to 0.
2016	8214	The jerk was set as the negative number.	Set the jerk value to a value greater than or equal to 0.
201A	8218	Buffer Mode setting value exceeded the range.	Set the value that can be set in Buffer Mode.
201B	8219	Execution Mode setting value exceeded the input range.	Set the value that can be set (0 ~ 1) in Execution Mode.
201C	8220	Transition Mode setting value exceeded the range.	Set the value that can be set in Transition Mode in the command.
201D	8221	Transition Parameter setting value exceeded the range.	Set the value that can be set in Transition Parameter in the command.
201E	8222	The axis group operation was stopped due to the error occurrence of axis group configuration axis.	Execute the command after eliminating the error factor and removing the error with the axis or axis group reset command.
201F	8223	The command position value transmitted to the servo drive was out of the pulse unit expression value.	It exceeded the 31-bit area when the command position value was converted in pulse unit. Set the value in the range of -2,147,483,648 ~ 2,147,483,647 when converted in pulse unit.
2020	8224	It is undefined axis group command.	This command is not performed in the current version of the controller. Please contact customer support team of our company after checking the version in which the command can be executed.
2021	8225	The previously executed command was canceled because the same command was executed.	Check whether the command was executed again during execution of the same command.
2022	8226	It exceeded the number of commands that can execute Buffered command.	The command cannot be executed because the command buffer of the axis is full. The number of commands that can be executed with the Buffered command is 100. Adjust the timing of the command execution.
2030	8240	Axis group parameters cannot be written if the axis group is in operation.	Perform the axis group parameter writing when the axis group is not in operation.
2040	8256	Axis group parameter data is abnormal.	Download the data again from XG5000 and place requests for A/S if the error occurs repeatedly after re-execution.
2041	8257	Operation cannot be executed due to error of the axis group parameter.	Check the axis group parameter and set it again.
2042	8258	The speed limit of the axis group parameter cannot be set to a value less than or equal to 0.	Set the speed limit to a value greater than 0.

Error code		Error Description	Solutions
Hex	Dec		
2043	8259	The configuration axis number setting value of the axis group parameter exceeded the range.	Set the configuration axis of the axis group in the range of 1~36.
2051	8273	The axis which you are going to add is already registered in the axis group.	Check whether the same axis number exists in the axis group and then set another axis.
2052	8274	The current axis group is active, and the axis you want to add is already included in the other activated axis group.	Execute the command after changing the enabled axis group that includes the axis to GroupDisabled state.
2053	8275	The IdentInGroup setting value of axis group add/remove command exceeded the range.	Set the IdentInGroup setting value in the range of 1 to 10.
2060	8288	There is no axis setting in the specified axis group of the axis group enable/disable command.	Set one or more axes in the axis group and execute the command.
2061	8289	The axis group cannot be enabled because there is an operating axis among the configuration axes of the current axis group.	Execute the command while all axes belonging to the axis group are not in operation.
2062	8290	The specified axis group cannot be enabled because the configuration axis of the current axis group is the configuration axis of another enabled axis group.	Check whether the axis belonging to the axis group is included in another enabled axis group.
2063	8291	The axis group operation cannot be executed because the configuration axes of the axis group have different units.	Set the unit of the configuration axis belonging to the axis group to the same value in order to execute the operation.
2064	8292	The axis group cannot be enabled due to the parameter error of the axis group configuration axis.	Set the parameter of the configuration axis belonging to the axis group within the normal range.
2065	8293	The axis group cannot be enabled because the speed command units of the axis group configuration axes are different from each other.	Set the same speed command unit for the configuration axes belonging to the axis group.
2066	8294	The axis group cannot be enabled because there is an axis whose speed command unit is rpm among the axis group configuration axes.	The speed command unit of the configuration axes belonging to the axis group cannot be set to rpm. Set it to a value other than rpm.
2067	8295	The coordinate system operation cannot be executed because the unit of the axis group configuration axes is different from the coordinate type.	Set the unit of the configuration axis belonging to the axis group to match the coordinate system type in order to execute the command.
206F	8303	The axis group cannot be activated if the axis group configuration axis is in NC control operation.	Execute the command when configuration axis belonging to the axis group is not in NC control operation.
2070	8304	The servo drive of the configuration axis does not support homing mode.	Confirm whether the servo drive supports EtherCAT CoE Homing mode and check the status of servo drive.
2071	8305	There is an axis where homing is not completed normally among configuration axes.	Execute the command again after eliminating the error factor by checking the configuration axis error code.
2072	8306	The axis group homing command cannot be executed when the axis group is in operation.	Execute the axis group homing command again in GroupStandby state after axis group operation stops.
2080	8320	There is an axis that has an error during the setting of the current position among the configuration axes.	Execute the command again after eliminating the error factor by checking the configuration axis error code.

Error code		Error Description	Solutions
Hex	Dec		
2090	8336	The absolute coordinate linear interpolation command cannot be executed if the configuration axis is in the undetermined origin state.	Execute the command after making origin determination state with the homing command or the current position setting command.
2091	8337	It exceeded the speed limit of the linear interpolation configuration axis.	Execute the command at a lower command speed so as not to exceed the speed limit of the configuration axis.
2092	8338	In the case of a specified corner distance transition, transition operation cannot be executed because the corner distance specification value is larger 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.
2093	8339	In the case of a specified corner distance transition, transition operation cannot be executed because the radius of an arc to be inserted exceeds 2,147,483,647 pulse.	Execute the linear interpolation by resetting the target position or changing the transition mode so that the two straight lines cannot be located on a straight line.
2094	8340	Linear interpolation operation cannot be executed when the main axis or subordinate axis is in infinite length repeat "allowable" state	Execute the command after changing the infinite length repeat setting of the main axis or subordinate axis to "0: Disable".
20A0	8352	Absolute coordinate circular interpolation command cannot be executed when the configuration axis is in undetermined origin state.	Execute the command after making the determined origin state with homing command or current position setting command.
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).
20A2	8354	Circular interpolation pass selection setting value exceeded the range.	Set the circular interpolation pass selection to a value between 0 and 1 (0: CW, 1: CCW).
20A3	8355	The radius setting exceeded the range in the circular interpolation radius method.	Set the radius setting value of the circular interpolation main axis operation data to be more than 80% of the half of the length from the start point to the end point.
20A4	8356	In circular interpolation, operation cannot be executed if start point = center point(midpoint) or center point(midpoint) = end point.	Execute the circular interpolation after setting the center point(or midpoint) to a different position from the start point(or end point).
20A5	8357	In circular interpolation, the start point and the end point cannot be the same in midpoint(or radius) method.	If you set the circular interpolation to midpoint (or radius), set the position of the start point differently from that of the end point and then execute the circular interpolation.
20A6	8358	It is radius setting error in circular interpolation.	The radius of a circle where the circular interpolation operation can be executed is greater than 0 and less than 2,147,483,647 pulse. Execute the command after setting the input value so that the radius can be calculated within the setting range.
20A7	8359	The operation cannot be executed because linear profile appears in circular interpolation.	Execute the circular interpolation changing the midpoint so that it cannot be located on the straight line of the start point and the end point in the case of a circular interpolation midpoint method.
20A8	8360	Linear interpolation operation cannot be executed when the main axis or subordinate axis is in infinite length repeat "allowable" state.	Execute the command after changing the infinite repeat setting of the main axis or subordinate axis to "0: Disable".
20A9	8361	Circular interpolation cannot be executed if there are more than four axes constituting the axis group.	Set the axis group to 2-axis for circular interpolation and 3-axis for helical interpolation.

Error code		Error Description	Solutions
Hex	Dec		
20AA	8362	Circular interpolation cannot be executed if the axis configuration of the axis group is not configured in regular sequence.	For circular interpolation, set the configuration axis for the axis group in regular sequence.
20AB	8363	It exceeded the speed limit of the circular interpolation configuration axis.	Execute the command with the command speed lowered so as not to exceed the speed limit of the configuration axis.
20AC	8364	In circular interpolation, the midpoint(center point) must be in the same XY plane as the start point in the midpoint(or radius) method.	Execute the circular interpolation after setting the center point (or midpoint) to the position in the same the XY plane as the start point (end point) in circular interpolation.
20C0	8384	The coordinate system operation command cannot be executed when the constituent axis is in the undetermined origin state.	Execute the command after making the determined origin state with the homing command or the current position setting command.
20C1	8385	The PCS setting parameter data of the coordinate system parameter is abnormal.	Check the PCS setting parameter and set it again.
20C2	8386	Coordinate system parameter, coordinate system-type parameter data is abnormal.	Check the coordinate system-type parameter and set it again.
20C3	8387	Coordinate system parameter, coordinate system-type parameter data is abnormal.	Check the instrument parameter and set it again.
20C4	8388	Coordinate system parameter, workspace-type data is abnormal.	Check the workspace-type parameter and set it again.
20C5	8389	Coordinate system parameter, workspace parameter data is abnormal.	Check the workspace parameter and set it again.
20C6	8390	It is the position where you cannot start the coordinate system operation.	Move to the position where you can start the coordinate system operation and execute the command.
20C7	8391	It is the target position that cannot be reached by the coordinate system operation.	Check whether there is abnormality in the target position or coordinate system parameter and set it again.
20C8	8392	It is the operation out of the workspace.	Check whether there is abnormality in the workspace parameter or target position and set it again.
20C9	8393	The axis group cannot be activated because the unit of the axis group configuration axis is different from the coordinate system type.	Make sure that the unit of the configuration axis belonging to the axis group matches the coordinate system time.
20CA	8394	The 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.
20CB	8395	The coordinate system operation cannot be executed when the configuration axis is in infinite length repeat "allowable" state.	Execute the command after changing the infinite length repeat setting of the configuration axis to "0: Disable".
20CC	8396	It is the CoordSystem that is not supported.	Execute the command after setting the supported CoordSystem.
20CD	8397	It is the TrajType that is not supported.	Execute the command after setting the supported TrajType.
20D0	8400	The conveyor axis setting value exceeded the range.	Set the conveyor axis in the range of 1~36.
20D1	8401	The axis set as the conveyor axis was set as the axis group configuration axis.	Execute the command when the conveyor axis is set to another axis.
20D2	8402	There is an error in setting the conveyor axis unit.	Set the unit of the conveyor axis to mm/inch.
20D3	8403	The conveyor axis was not ready.	Execute the command when the conveyor axis is ready for operation.

Error code		Error Description	Solutions
Hex	Dec		
20D4	8404	The conveyor synchronization command cannot be executed if the main axis is in homing operation.	Execute the command when the conveyor axis is not in homing operation.
20D5	8405	The conveyor synchronization command cannot be executed if the main axis is operating with torque control.	Execute the command when the conveyor axis is not in operation with torque control.
20D6	8406	Conveyor synchronization function cannot be executed if the conveyor axis is in infinite length repeat "Disable" state.	Execute the command after changing the infinite length repeat setting of the conveyor axis to "1: Permit".
20E0	8416	The SETP value of the coordinate system path operation exceeded the range.	Set the STEP value of the coordinate system path operation to a value between 0 and 99 and execute the command.
20E1	8417	The CommandType value of the coordinate system path operation exceeded the range.	Set the CommandType value of the coordinate system path operation to a value between 0 and 4 and execute the command.
20E2	8418	The Mode value of the coordinate system path operation exceeded the range.	Set the Mode value of the coordinate system path operation to a value between 0 and 2 and execute the command.
20E3	8419	It exceeded the number of paths that can be performed by the coordinate system path operation.	Set the STEP value of the coordinate system path operation to a value between 0 and 99 and execute the command.
3000	12288	The NC channel was not ready for operation.	Check whether the NC channel is ready for operation. In order to use the NC channel, the NC channel should be registered in NC parameter in XG5000.
3001	12289	NC program data is abnormal.	Download the data again from XG5000 and place requests for A/S if it occurs repeatedly after re-execution.
3002	12290	Program cannot be written when NC channel is running automatically.	Execute the program writing when the automatic operation stops if the NC channel is in automatic operation.
3003	12291	NC program writing was not completed normally. (File processing (DELETE) failure in NC program writing)	Download the data again from XG5000 and place requests for A/S if it occurs repeatedly after re-execution.
3004	12292	NC program writing was not completed normally. (File processing(OPEN) failure in writing NC program)	Download the data again from XG5000 and place requests for A/S if it occurs repeatedly after re-execution.
3005	12293	NC program writing was not completed normally. (File processing(WRITE) failure in writing NC program)	Download the data again from XG5000 and place requests for A/S if it occurs repeatedly after re-execution.
3006	12294	NC program writing was not completed normally. (File processing(CLOSE) failure in writing NC program)	Download the data again from XG5000 and place requests for A/S if it occurs repeatedly after re-execution.
3007	12295	Parameters cannot be written when NC channel is running automatically.	Execute the program writing when the automatic operation stops if the NC channel is in automatic operation.
3008	12296	The automatic operation cannot continue because the controller's mode is changed to STOP or ERROR state during the NC channel automatic operation.	Check whether the controller's mode is changed to STOP or ERROR state during the NC channel automatic operation.

Error code		Error Description	Solutions
Hex	Dec		
3009	12297	The automatic operation cannot continue because EtherCAT network connection is disconnected during the NC channel automatic operation.	Check whether the EtherCAT network connection has been disconnected due to slave power supply error, network cable error and noise inflow on network cable during the NC channel automatic operation.
300A	12298	The automatic operation cannot continue because the controller is stopped by the ESTOP command during the NC channel automatic operation.	Check whether the controller was stopped by the ESTOP command during the NC channel automatic operation.
3011	12305	The automatic operation cannot continue because NC X axis is not ready for operation.	Check whether the NC X axis is in servo-off state or drive alarm state. The NC channel automatic operation can be executed when the configuration axis is in servo-on state, or drive alarm does not occur.
3012	12306	The automatic operation cannot continue because NC Y axis is not ready for operation.	Check whether the NC Y axis is in servo-off state or drive alarm state. The NC channel automatic operation can be executed when the configuration axis is in servo-on state, or drive alarm does not occur.
3013	12307	The automatic operation cannot continue because NC Z axis is not ready for operation.	Check whether the NC Z axis is in servo-off state or drive alarm state. The NC channel automatic operation can be executed when the configuration axis is in servo-on state, or drive alarm does not occur.
3014	12308	The automatic operation cannot continue because NC A axis is not ready for operation.	Check whether the NC A axis is in servo-off state or drive alarm state. The NC channel automatic operation can be executed when the configuration axis is in servo-on state, or drive alarm does not occur.
3015	12309	The automatic operation cannot continue because NC B axis is not ready for operation.	Check whether the NC B axis is in servo-off state or drive alarm state. The NC channel automatic operation can be executed when the configuration axis is in servo-on state, or drive alarm does not occur.
3016	12310	The automatic operation cannot continue because NC C axis is not ready for operation.	Check whether the NC C axis is in servo-off state or drive alarm state. The NC channel automatic operation can be executed when the configuration axis is in servo-on state, or drive alarm does not occur.
3017	12311	The automatic operation cannot continue because NC U axis is not ready for operation.	Check whether the NC U axis is in servo-off state or drive alarm state. The NC channel automatic operation can be executed when the configuration axis is in servo-on state, or drive alarm does not occur.
3018	12312	The automatic operation cannot continue because NC V axis is not ready for operation.	Check whether the NC V axis is in servo-off state or drive alarm state. The NC channel automatic operation can be executed when the configuration axis is in servo-on state, or drive alarm does not occur.
3019	12313	The automatic operation cannot continue because NC W axis is not ready for operation.	Check whether the NC W axis is in servo-off state or drive alarm state. The NC channel automatic operation can be executed when the configuration axis is in servo-on state, or drive alarm does not occur.

Error code		Error Description	Solutions
Hex	Dec		
301A	12314	The automatic operation cannot continue because NC S axis is not ready for operation.	Check whether the NC S axis is in servo-off state or drive alarm state. The NC channel automatic operation can be executed when the configuration axis is in servo-on state, or drive alarm does not occur.
3020	12320	It is undefined NC channel command.	The NC command cannot be supported in the current controller version. Check the version in which the command can be executed and contact the customer support team of our company.
3021	12321	The previously executed command was canceled because the same NC channel command was executed.	The NC channel command can be executed only once per scan. Change the operating condition of the program so that one NC channel command can be executed in one scan.
3030	12336	The automatic operation cannot continue because interpreter (IPR) alarm occurs during the NC channel automatic operation.	Check the interpreter (IPR) error code among the NC channel flags. You can execute the automatic operation start command (NC_CycleStart) again after resetting the NC channel with the NC reset command(NC_Reset).
3031	12337	The automatic operation cannot continue because program processor(PA) alarm occurs during the NC channel automatic operation.	Check the program processor (PA) error code among the NC channel flags. You can execute the automatic operation start command (NC_CycleStart) again after resetting the NC channel with the NC reset command(NC_Reset).
3040	12352	The command position setting value was out of the pulse unit expression value during NC channel automatic operation.	It exceeded a 32-bit area when the command position value was converted in pulse unit. Set the value in the range of -2,147,483,648 to 2,147,483,647 when converting the command position value to pulse.
3041	12353	The command position setting value of NC X axis was out of the pulse unit expression value.	It exceeded a 32-bit area when the command position value of NC X axis was converted in pulse unit. Set the value in the range of -2,147,483,648 to 2,147,483,647 when converting the command position value to pulse.
3042	12354	The command position setting value of NC Y axis was out of the pulse unit expression value.	It exceeded a 32-bit area when the command position value of NC Y axis was converted in pulse unit. Set the value in the range of -2,147,483,648 to 2,147,483,647 when converting the command position value to pulse.
3043	12355	The command position setting value of NC Z axis was out of the pulse unit expression value.	It exceeded a 32-bit area when the command position value of NC Z axis was converted in pulse unit. Set the value in the range of -2,147,483,648 to 2,147,483,647 when converting the command position value to pulse.
3044	12356	The command position setting value of NC A axis was out of the pulse unit expression value.	It exceeded a 32-bit area when the command position value of NC A axis was converted in pulse unit. Set the value in the range of -2,147,483,648 to 2,147,483,647 when converting the command position value to pulse.
3045	12357	The command position setting value of NC B axis was out of the pulse unit expression value.	It exceeded a 32-bit area when the command position value of NC B axis was converted in pulse unit. Set the value in the range of -2,147,483,648 to 2,147,483,647 when converting the command position value to pulse.
3046	12358	The command position setting value of NC C axis was out of the pulse unit expression value.	It exceeded a 32-bit area when the command position value of NC C axis was converted in pulse unit. Set the value in the range of -2,147,483,648 to 2,147,483,647 when converting the command position value to pulse.

Error code		Error Description	Solutions
Hex	Dec		
3047	12359	The command position setting value of NC U axis was out of the pulse unit expression value.	It exceeded a 32-bit area when the command position value of NC U axis was converted in pulse unit. Set the value in the range of -2,147,483,648 to 2,147,483,647 when converting the command position value to pulse.
3048	12360	The command position setting value of NC V axis was out of the pulse unit expression value.	It exceeded a 32-bit area when the command position value of NC V axis was converted in pulse unit. Set the value in the range of -2,147,483,648 to 2,147,483,647 when converting the command position value to pulse.
3049	12361	The command position setting value of NC W axis was out of the pulse unit expression value.	It exceeded a 32-bit area when the command position value of NC W axis was converted in pulse unit. Set the value in the range of -2,147,483,648 to 2,147,483,647 when converting the command position value to pulse.
304A	12362	The command position setting value of NC S axis was out of the pulse unit expression value.	It exceeded a 32-bit area when the command position value of NC S axis was converted in pulse unit. Set the value in the range of -2,147,483,648 to 2,147,483,647 when converting the command position value to pulse.
3050	12368	The command position of the NC channel configuration axis was out of the software upper limit position.	Deviate from the software upper limit range by using the reverse jog command in the NC configuration axis where the error occurred, and then remove the error by executing the error reset command.
3051	12369	The command position of NC X axis was out of the software upper limit position.	Deviate from the software upper limit range by using the reverse jog command in NC X axis, and then remove the error by executing the error reset command.
3052	12370	The command position of NC Y axis was out of the software upper limit position.	Deviate from the software upper limit range by using the reverse jog command in NC Y axis, and then remove the error by executing the error reset command.
3053	12371	The command position of NC Z axis was out of the software upper limit position.	Deviate from the software upper limit range by using the reverse jog command in NC Z axis, and then remove the error by executing the error reset command.
3054	12372	The command position of NC A axis was out of the software upper limit position.	Deviate from the software upper limit range by using the reverse jog command in NC A axis, and then remove the error by executing the error reset command.
3055	12373	The command position of NC B axis was out of the software upper limit position.	Deviate from the software upper limit range by using the reverse jog command in NC B axis, and then remove the error by executing the error reset command.
3056	12374	The command position of NC C axis was out of the software upper limit position.	Deviate from the software upper limit range by using the reverse jog command in NC C axis, and then remove the error by executing the error reset command.
3057	12375	The command position of NC U axis was out of the software upper limit position.	Deviate from the software upper limit range by using the reverse jog command in NC U axis, and then remove the error by executing the error reset command.
3058	12376	The command position of NC V axis was out of the software upper limit position.	Deviate from the software upper limit range by using the reverse jog command in NC V axis, and then remove the error by executing the error reset command.

Error code		Error Description	Solutions
Hex	Dec		
3059	12377	The command position of NC W axis was out of the software upper limit position.	Deviate from the software upper limit range by using the reverse jog command in NC W axis, and then remove the error by executing the error reset command.
305A	12378	The command position of NC S axis was out of the software upper limit position.	Deviate from the software upper limit range by using the reverse jog command in NC S axis, and then remove the error by executing the error reset command.
305B	12379	The command position of the NC channel/axis was out of the inner range of G22 traverse prohibited area.	Deviate from the range of G22 traverse prohibited area by using the jog command in the NC configuration axis where the error occurred, and then remove the error by executing the error reset command.
305C	12380	The command position of the NC channel/axis was out of the outer range of G22 traverse prohibited area.	Deviate from the range of G22 traverse prohibited area by using the jog command in the NC configuration axis where the error occurred, and then remove the error by executing the error reset command.
305D	12381	The command position of the NC channel/axis was outside the range of the third traverse prohibited area.	Deviate from the range of the third traverse prohibited area by using the jog command in the NC configuration axis where the error occurred, and then remove the error by executing the error reset command.
3060	12384	The command position of the NC channel configuration axis was out of the software lower limit position.	Deviate from the software lower limit range by using the forward jog command in the NC configuration axis where the error occurred, and then remove the error by executing the error reset command.
3061	12385	The command position of NC X axis was out of the software lower limit position.	Deviate from the software lower limit range by using the forward jog command in NC X axis, and then remove the error by executing the error reset command.
3062	12386	The command position of NC Y axis was out of the software lower limit position.	Deviate from the software lower limit range by using the forward jog command in NC Y axis, and then remove the error by executing the error reset command.
3063	12387	The command position of NC Z axis was out of the software lower limit position.	Deviate from the software lower limit range by using the forward jog command in NC Z axis, and then remove the error by executing the error reset command.
3064	12388	The command position of NC A axis was out of the software lower limit position.	Deviate from the software lower limit range by using the forward jog command in NC A axis, and then remove the error by executing the error reset command.
3065	12389	The command position of NC B axis was out of the software lower limit position.	Deviate from the software lower limit range by using the forward jog command in NC B axis, and then remove the error by executing the error reset command.
3066	12390	The command position of NC C axis was out of the software lower limit position.	Deviate from the software lower limit range by using the forward jog command in NC C axis, and then remove the error by executing the error reset command.
3067	12391	The command position of NC U axis was out of the software lower limit position.	Deviate from the software lower limit range by using the forward jog command in NC U axis, and then remove the error by executing the error reset command.
3068	12392	The command position of NC V axis was out of the software lower limit position.	Deviate from the software lower limit range by using the forward jog command in NC V axis, and then remove the error by executing the error reset command.

Error code		Error Description	Solutions
Hex	Dec		
3069	12393	The command position of NC W axis was out of the software lower limit position.	Deviate from the software lower limit range by using the forward jog command in NC W axis, and then remove the error by executing the error reset command.
306A	12394	The command position of NC S axis was out of the software lower limit position.	Deviate from the software lower limit range by using the forward jog command in NC S axis, and then remove the error by executing the error reset command.
3071	12401	The automatic operation cannot continue because NC X axis is not in the origin determination complete state.	Check whether NC X axis is in the undetermined origin state. You can change the axis to the determined origin state by using the homing command(MC_Home, NC_Home).
3072	12402	The automatic operation cannot continue because NC Y axis is not in the origin determination complete state.	Check whether NC Y axis is in the undetermined origin state. You can change the axis to the determined origin state by using the homing command(MC_Home, NC_Home).
3073	12403	The automatic operation cannot continue because NC Z axis is not in the origin determination complete state.	Check whether NC Z axis is in the undetermined origin state. You can change the axis to the determined origin state by using the homing command(MC_Home, NC_Home).
3074	12404	The automatic operation cannot continue because NC A axis is not in the origin determination complete state.	Check whether NC A axis is in the undetermined origin state. You can change the axis to the determined origin state by using the homing command(MC_Home, NC_Home).
3075	12405	The automatic operation cannot continue because NC B axis is not in the origin determination complete state.	Check whether NC B axis is in the undetermined origin state. You can change the axis to the determined origin state by using the homing command(MC_Home, NC_Home).
3076	12406	The automatic operation cannot continue because NC C axis is not in the origin determination complete state.	Check whether NC C axis is in the undetermined origin state. You can change the axis to the determined origin state by using the homing command(MC_Home, NC_Home).
3077	12407	The automatic operation cannot continue because NC U axis is not in the origin determination complete state.	Check whether NC U axis is in the undetermined origin state. You can change the axis to the determined origin state by using the homing command(MC_Home, NC_Home).
3078	12408	The automatic operation cannot continue because NC V axis is not in the origin determination complete state.	Check whether NC V axis is in the undetermined origin state. You can change the axis to the determined origin state by using the homing command(MC_Home, NC_Home).
3079	12409	The automatic operation cannot continue because NC W axis is not in the origin determination complete state.	Check whether NC W axis is in the undetermined origin state. You can change the axis to the determined origin state by using the homing command(MC_Home, NC_Home).
307A	12410	The automatic operation cannot continue because NC S axis is not in the origin determination complete state.	Check whether NC S axis is in the undetermined origin state. You can change the axis to the determined origin state by using the homing command(MC_Home, NC_Home).
3080	12416	The automatic operation cannot continue because drive abnormal condition (upper/lower limit, alarm, servo off) of the NC configuration axis occurs during the NC channel automatic operation.	Remove the cause for abnormal condition after checking whether the drive status of the NC configuration axis was changed to the upper limit/lower limit, or alarm occurrence or servo-off state during the NC channel automatic operation.
3081	12417	The automatic operation cannot continue because drive abnormal condition (upper/lower limit, alarm, servo off) of NC X axis occurs during the NC channel automatic operation.	Remove the cause for abnormal condition after checking whether the drive status of NC X axis was changed to the upper limit/lower limit, or alarm occurrence or servo-off state during the NC channel automatic operation.

Error code		Error Description	Solutions
Hex	Dec		
3082	12418	The automatic operation cannot continue because drive abnormal condition (upper/lower limit, alarm, servo off) of NC Y axis occurs during the NC channel automatic operation.	Remove the cause for abnormal condition after checking whether the drive status of NC Y axis was changed to the upper limit/lower limit, or alarm occurrence or servo-off state during the NC channel automatic operation.
3083	12419	The automatic operation cannot continue because drive abnormal condition (upper/lower limit, alarm, servo off) of NC Z axis occurs during the NC channel automatic operation.	Remove the cause for abnormal condition after checking whether the drive status of NC Z axis was changed to the upper limit/lower limit, or alarm occurrence or servo-off state during the NC channel automatic operation.
3084	12420	The automatic operation cannot continue because drive abnormal condition (upper/lower limit, alarm, servo off) of NC A axis occurs during the NC channel automatic operation.	Remove the cause for abnormal condition after checking whether the drive status of NC A axis was changed to the upper limit/lower limit, or alarm occurrence or servo-off state during the NC channel automatic operation.
3085	12421	The automatic operation cannot continue because drive abnormal condition (upper/lower limit, alarm, servo off) of NC B axis occurs during the NC channel automatic operation.	Remove the cause for abnormal condition after checking whether the drive status of NC B axis was changed to the upper limit/lower limit, or alarm occurrence or servo-off state during the NC channel automatic operation.
3086	12422	The automatic operation cannot continue because drive abnormal condition (upper/lower limit, alarm, servo off) of NC C axis occurs during the NC channel automatic operation.	Remove the cause for abnormal condition after checking whether the drive status of NC C axis was changed to the upper limit/lower limit, or alarm occurrence or servo-off state during the NC channel automatic operation.
3087	12423	The automatic operation cannot continue because drive abnormal condition (upper/lower limit, alarm, servo off) of NC U axis occurs during the NC channel automatic operation.	Remove the cause for abnormal condition after checking whether the drive status of NC U axis was changed to the upper limit/lower limit, or alarm occurrence or servo-off state during the NC channel automatic operation.
3088	12424	The automatic operation cannot continue because drive abnormal condition (upper/lower limit, alarm, servo off) of NC V axis occurs during the NC channel automatic operation.	Remove the cause for abnormal condition after checking whether the drive status of NC V axis was changed to the upper limit/lower limit, or alarm occurrence or servo-off state during the NC channel automatic operation.
3089	12425	The automatic operation cannot continue because drive abnormal condition (upper/lower limit, alarm, servo off) of NC W axis occurs during the NC channel automatic operation.	Remove the cause for abnormal condition after checking whether the drive status of NC W axis was changed to the upper limit/lower limit, or alarm occurrence or servo-off state during the NC channel automatic operation.
308A	12426	The automatic operation cannot continue because drive abnormal condition (upper/lower limit, alarm, servo off) of NC S axis occurs during the NC channel automatic operation.	Remove the cause for abnormal condition after checking whether the drive status of NC S axis was changed to the upper limit/lower limit, or alarm occurrence or servo-off state during the NC channel automatic operation.
3100	12544	NC channel parameter data is abnormal.	Download the data again from XG5000 and place requests for A/S if it occurs repeatedly after re-execution.
3101	12545	Operation cannot be executed due to the NC parameter abnormality.	Check the NC parameter and reset it if the settings such as the data range are not correct.
3102	12546	The cutting feed upper/lower limit rate setting value of NC channel parameter exceeded the range.	Set the cutting feed upper/lower limit rate value of the NC channel parameter to a value greater than 0. Set the cutting feed upper limit rate value to be larger than the cutting feed lower limit rate value.

Error code		Error Description	Solutions
Hex	Dec		
3103	12547	The circular processing speed limit upper/lower limit cutting feed rate setting value of NC channel parameter exceeded the range.	Set the arc processing speed limit upper/lower limit cutting speed value of the NC channel parameter to a value greater than 0. Set the arc processing speed limit upper limit speed value to be larger than the arc processing speed limit lower limit speed value.
3200	12800	NC channel/axis parameter data is abnormal.	Download the data again from XG5000 and place requests for A/S if it occurs repeatedly after re-execution.
3310	13072	NC Feed Hold command was executed in a state other than automatic operation, or the currently executed program block cannot execute NC Feed Hold.	Execute NC Feed Hold command (NC_FeedHold) when the NC channel is in automatic operation. Check whether the currently executed program block is ready for Feed Hold if it is in automatic operation.
3320	13088	The override factor of the NC rapid traverse override command exceeded the range.	Execute the override command after setting the VelFactor, AccFactor and JerkFactor value of the override command to be greater than 0.
3321	13089	The operation speed value of NC X axis exceeded the speed limit value after the override factor of NC rapid traverse override command was reflected.	Perform the override within the range that does not exceed the speed limit value after checking the speed limit value of the axis connected to NC X axis.
3322	13090	The operation speed value of NC Y axis exceeded the speed limit value after the override factor of the NC rapid traverse override command was reflected.	Perform the override within the range that does not exceed the speed limit after checking the speed limit value of the axis connected to NC Y axis.
3323	13091	The operation speed value of NC Z axis exceeded the speed limit value after the override factor of the NC rapid traverse override command was reflected.	Perform the override within the range that does not exceed the speed limit after checking the speed limit value of the axis connected to NC Z axis.
3324	13092	The operation speed value of NC A axis exceeded the speed limit value after the override factor of the NC rapid traverse override command was reflected.	Perform the override within the range that does not exceed the speed limit after checking the speed limit value of the axis connected to NC A axis.
3325	13093	The operation speed value of NC B axis exceeded the speed limit value after the override factor of the NC rapid traverse override command was reflected.	Perform the override within the range that does not exceed the speed limit after checking the speed limit value of the axis connected to NC B axis.
3326	13094	The operation speed value of NC C axis exceeded the speed limit value after the override factor of the NC rapid traverse override command was reflected.	Perform the override within the range that does not exceed the speed limit after checking the speed limit value of the axis connected to NC C axis.
3327	13095	The operation speed value of NC U axis exceeded the speed limit value after the override factor of the NC rapid traverse override command was reflected.	Perform the override within the range that does not exceed the speed limit after checking the speed limit value of the axis connected to NC U axis.
3328	13096	The operation speed value of NC V axis exceeded the speed limit value after the override factor of the NC rapid traverse override command was reflected.	Perform the override within the range that does not exceed the speed limit after checking the speed limit value of the axis connected to NC V axis.

Error code		Error Description	Solutions
Hex	Dec		
3329	13097	The operation speed value of NC W axis exceeded the speed limit value after the override factor of the NC rapid traverse override command was reflected.	Perform the override within the range that does not exceed the speed limit after checking the speed limit value of the axis connected to NC W axis.
332A	13098	The operation speed value of NC S axis exceeded the speed limit value after the override factor of the NC rapid traverse override command was reflected.	Perform the override within the range that does not exceed the speed limit after checking the speed limit value of the axis connected to NC S axis.
332B	13099	The reset by NC_Reset or NC_Emergency command was executed, and thus the NC rapid traverse override command was canceled.	Execute the command again after NC_Reset or NC_Emergency command ends, if the NC rapid traverse override function was needed.
332C	13100	The NC_RapidTraverseOverride command cannot be executed when it is resetting by NC_Reset or NC_Emergency command.	Execute the NC command after NC_Reset or NC_Emergency command ends.
3330	13104	The override factor of NC cutting feed override command exceeded the range.	Execute the override command after setting the VelFactor, AccFactor and JerkFactor value of the override command to be greater than 0.
3331	13105	The operation speed value exceeded the cutting feed upper limit rate value after the override factor of NC cutting feed override command was reflected.	Perform the override within the range that does not exceed the cutting feed upper limit rate value after checking the cutting feed upper limit rate value of the NC channel parameter.
3332	13106	The reset by NC_Reset or NC_Emergency command was executed, and thus the NC cutting feed override command was canceled.	Execute the command again after NC_Reset or NC_Emergency command ends, if the NC cutting feed override function was needed.
3333	13107	The NC_CuttingFeedOverride command cannot be executed when it is resetting by NC_Reset or NC_Emergency command.	Execute the NC command after NC_Reset or NC_Emergency command ends.
350A	13578	The NC_CycleStart command cannot be executed when it is resetting by NC_Reset or NC_Emergency command.	Execute the NC command after NC_Reset or NC_Emergency command ends.
3340	13120	The override factor of NC spindle override command exceeded the range.	Execute the override command after setting the VelFactor, AccFactor and JerkFactor value of the override command to be greater than 0.
3341	13121	The operation speed value of the spindle exceeded the speed limit value after the override factor of the NC spindle override command was reflected.	Perform the override within the range that does not exceed the speed limit after checking the speed limit value of the axis connected to the spindle axis.
3342	13122	The NC_SpindleOverride command cannot be executed when it is resetting by NC_Reset or NC_Emergency command.	Execute the command again after NC_Reset or NC_Emergency command ends, if the NC spindle override function was needed.
3343	13123	The NC_SpindleOverride command cannot be executed when it is resetting by NC_Reset or NC_Emergency command.	Execute the NC command after NC_Reset or NC_Emergency command ends.

Error code		Error Description	Solutions
Hex	Dec		
3350	13136	The setting axis of the NC parameter read command was not enabled as the NC axis.	Check whether the setting axis of the NC parameter read command was registered as NC channel/axis parameter. The NC channel/axis can be registered in the NC channel parameter among the motion data items of XG5000.
3351	13137	The axis setting value of the NC parameter read command exceeded the allowable range.	Set the axis number to a value between 0 and 10. Perform channel parameter reading if the axis value is 0 and NC axis X ~ NC axis S reading if 1 ~ 10.
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 ~ 17 for the channel parameter, and 1 ~ 5 for the channel/axis parameter. Check the group number to which the parameter you want to read belongs, and then execute the parameter read command(NC_ReadParameter).
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 channel parameter or channel/axis parameter group is supported. Check the group number to which the parameter you want to read belongs and the parameter number, and then execute the parameter read command (NC_ReadParameter).
3500	13568	The automatic operation start operation cannot be executed if the NC channel is in automatic operation.	Check whether NC channel is currently running automatically. The automatic operation can be restarted after the automatic operation is completed.
3501	13569	The automatic operation start operation cannot be executed because NC Feed Hold command is in Enable status.	Execute the automatic operation start command (NC_CycleStart) again after releasing the Enable input of the NC Feed Hold command (NC_FeedHold).
3502	13570	The automatic operation start operation cannot be executed because NC emergency stop command is in Enable status.	Execute the automatic operation start command (NC_CycleStart) again after releasing the Enable input of the NC emergency stop command (NC_Emergency).
3503	13571	The operation start command cannot be executed when NC interpreter(IPR) is not terminated normally.	Execute the automatic operation start command (NC_CycleStart) again after resetting the NC channel with the NC reset command(NC_Reset).
3504	13572	The automatic operation start command cannot be executed because NC interpreter(IPR) or program processor(PA) is in error state.	Check the interpreter(IPR) and program processor(PA) error code in the NC channel flags. The automatic operation start command (NC_CycleStart) can be executed again after the NC channel is reset with the NC reset command (NC_Reset).
3505	13573	The automatic operation start command cannot be executed because the program to be executed on the NC channel is not set.	Execute the automatic operation start command (NC_CycleStart) again after designating the program to be executed with the NC program designation command(NC_LoadProgram).
3506	13574	The automatic operation start command cannot be executed because the NC channel reached the target processing quantity or the target processing quantity in M99 repeat processing.	Confirm the processing quantity of the NC channel flag or the processing quantity in M99 repeat processing, and then check whether it has reached the target processing quantity. Execute the automatic operation start command (NC_CycleStart) again after resetting the processing quantity or the processing quantity flag in M99 repeat processing
3507	13575	The automatic operation start command cannot be executed if NC M/S/T-code output strobe signal is on.	Execute the automatic operation start command (NC_CycleStart) after completing the NC M/S/T-code output strobe signal.

Error code		Error Description	Solutions
Hex	Dec		
3508	13576	The NC channel interpreter (IPR) was not executed normally.	Execute the automatic operation start command (NC_CycleStart) again after resetting the NC channel with the NC reset command (NC_Reset).
3509	13577	The automatic operation start command cannot be executed when entered into the NC traverse prohibited area.	Deviate from the range of the traverse prohibited area by using the jog command in the NC configuration axis where an error occurred, and then remove the error by executing the error reset command.
350A	13578	The NC_CycleStart command cannot be executed when it is resetting by NC_Reset or NC_Emergency command.	Execute the NC command after NC_Reset or NC_Emergency command ends.
3510	13584	The automatic operation start command cannot be executed because NC channel configuration axis is not ready.	Execute the automatic operation start command (NC_CycleStart) when the NC channel configuration axes are all ready. In order to start the automatic operation, the NC channel configuration axes should be connected to the EtherCAT network or set as virtual axes.
3511	13585	The automatic operation start command cannot be executed because NC X axis is not ready.	Execute the automatic operation start command (NC_CycleStart) when the NC channel configuration axes are all ready. In order to start the automatic operation, the NC channel configuration axes should be connected to the EtherCAT network or set as virtual axes.
3512	13586	The automatic operation start command cannot be executed because NC Y axis is not ready.	Execute the automatic operation start command (NC_CycleStart) when the NC channel configuration axes are all ready. In order to start the automatic operation, the NC channel configuration axes should be connected to the EtherCAT network or set as virtual axes.
3513	13587	The automatic operation start command cannot be executed because NC Z axis is not ready.	Execute the automatic operation start command (NC_CycleStart) when the NC channel configuration axes are all ready. In order to start the automatic operation, the NC channel configuration axes should be connected to the EtherCAT network or set as virtual axes.
3514	13588	The automatic operation start command cannot be executed because NC A axis is not ready.	Execute the automatic operation start command (NC_CycleStart) when the NC channel configuration axes are all ready. In order to start the automatic operation, the NC channel configuration axes should be connected to the EtherCAT network or set as virtual axes.
3515	13589	The automatic operation start command cannot be executed because NC B axis is not ready.	Execute the automatic operation start command (NC_CycleStart) when the NC channel configuration axes are all ready. In order to start the automatic operation, the NC channel configuration axes should be connected to the EtherCAT network or set as virtual axes.
3516	13590	The automatic operation start command cannot be executed because NC C axis is not ready.	Execute the automatic operation start command (NC_CycleStart) when the NC channel configuration axes are all ready. In order to start the automatic operation, the NC channel configuration axes should be connected to the EtherCAT network or set as virtual axes.

Error code		Error Description	Solutions
Hex	Dec		
3517	13591	The automatic operation start command cannot be executed because NC U axis is not ready.	Execute the automatic operation start command (NC_CycleStart) when the NC channel configuration axes are all ready. In order to start the automatic operation, the NC channel configuration axes should be connected to the EtherCAT network or set as virtual axes.
3518	13592	The automatic operation start command cannot be executed because NC V axis is not ready.	Execute the automatic operation start command (NC_CycleStart) when the NC channel configuration axes are all ready. In order to start the automatic operation, the NC channel configuration axes should be connected to the EtherCAT network or set as virtual axes.
3519	13593	The automatic operation start command cannot be executed because NC W axis is not ready.	Execute the automatic operation start command (NC_CycleStart) when the NC channel configuration axes are all ready. In order to start the automatic operation, the NC channel configuration axes should be connected to the EtherCAT network or set as virtual axes.
351A	13594	The automatic operation start command cannot be executed because NC S axis is not ready.	Execute the automatic operation start command (NC_CycleStart) when the NC channel configuration axes are all ready. In order to start the automatic operation, the NC channel configuration axes should be connected to the EtherCAT network or set as virtual axes.
3520	13600	The automatic operation start command cannot be executed because NC channel configuration axis is in operation.	Execute the automatic operation start command (NC_CycleStart) when the NC channel configuration axis stops.
3521	13601	The automatic operation start command cannot be executed because NC X axis is in operation.	Execute the automatic operation start command (NC_CycleStart) when the NC channel configuration axis stops.
3522	13602	The automatic operation start command cannot be executed because NC Y axis is in operation.	Execute the automatic operation start command (NC_CycleStart) when the NC channel configuration axis stops.
3523	13603	The automatic operation start command cannot be executed because NC Z axis is in operation.	Execute the automatic operation start command (NC_CycleStart) when the NC channel configuration axis stops.
3524	13604	The automatic operation start command cannot be executed because NC A axis is in operation.	Execute the automatic operation start command (NC_CycleStart) when the NC channel configuration axis stops.
3525	13605	The automatic operation start command cannot be executed because NC B axis is in operation.	Execute the automatic operation start command (NC_CycleStart) when the NC channel configuration axis stops.
3526	13606	The automatic operation start command cannot be executed because NC C axis is in operation.	Execute the automatic operation start command (NC_CycleStart) when the NC channel configuration axis stops.
3527	13607	The automatic operation start command cannot be executed because NC U axis is in operation.	Execute the automatic operation start command (NC_CycleStart) when the NC channel configuration axis stops.
3528	13608	The automatic operation start command cannot be executed because NC V axis is in operation.	Execute the automatic operation start command (NC_CycleStart) when the NC channel configuration axis stops.
3529	13609	The automatic operation start command cannot be executed because NC W axis is in operation.	Execute the automatic operation start command (NC_CycleStart) when the NC channel configuration axis stops.
3530	13616	The automatic operation start command cannot be executed because NC channel configuration axis is enabled as a motion axis group configuration axis.	Execute the automatic operation start command (NC_CycleStart) with the NC channel configuration axis disabled as a motion axis group configuration axis.

Error code		Error Description	Solutions
Hex	Dec		
3531	13617	The automatic operation start command cannot be executed because NC X axis is enabled as a motion axis group configuration axis.	Execute the automatic operation start command (NC_CycleStart) with the NC channel configuration axis disabled as a motion axis group configuration axis.
3532	13618	The automatic operation start command cannot be executed because NC Y axis is enabled as a motion axis group configuration axis.	Execute the automatic operation start command (NC_CycleStart) with the NC channel configuration axis disabled as a motion axis group configuration axis.
3533	13619	The automatic operation start command cannot be executed because NC Z axis is enabled as a motion axis group configuration axis.	Execute the automatic operation start command (NC_CycleStart) with the NC channel configuration axis disabled as a motion axis group configuration axis.
3534	13620	The automatic operation start command cannot be executed because NC A axis is enabled as a motion axis group configuration axis.	Execute the automatic operation start command (NC_CycleStart) with the NC channel configuration axis disabled as a motion axis group configuration axis.
3535	13621	The automatic operation start command cannot be executed because NC B axis is enabled as a motion axis group configuration axis.	Execute the automatic operation start command (NC_CycleStart) with the NC channel configuration axis disabled as a motion axis group configuration axis.
3536	13622	The automatic operation start command cannot be executed because NC C axis is enabled as a motion axis group configuration axis.	Execute the automatic operation start command (NC_CycleStart) with the NC channel configuration axis disabled as a motion axis group configuration axis.
3537	13623	The automatic operation start command cannot be executed because NC U axis is enabled as a motion axis group configuration axis.	Execute the automatic operation start command (NC_CycleStart) with the NC channel configuration axis disabled as a motion axis group configuration axis.
3538	13624	The automatic operation start command cannot be executed because NC V axis is enabled as a motion axis group configuration axis.	Execute the automatic operation start command (NC_CycleStart) with the NC channel configuration axis disabled as a motion axis group configuration axis.
3539	13625	The automatic operation start command cannot be executed because NC W axis is enabled as a motion axis group configuration axis.	Execute the automatic operation start command (NC_CycleStart) with the NC channel configuration axis disabled as a motion axis group configuration axis.
353A	13626	The automatic operation start command cannot be executed because NC S axis is enabled as a motion axis group configuration axis.	Execute the automatic operation start command (NC_CycleStart) with the NC channel configuration axis disabled as a motion axis group configuration axis.
3540	13632	The position unit or speed unit setting of NC channel configuration axis is invalid.	For NC operation, set the unit of the NC channel configuration axis (except for spindle) to mm or deg. Set the unit of speed to RPM for spindle axis(S axis) and unit/min for other axes(X, Y, Z, A, B, C, U, V, W).
3541	13633	The position unit or speed unit setting of NC X axis is invalid.	For NC operation, set the unit of the NC channel configuration axis to mm or deg. Set the unit of speed to unit/min.
3542	13634	The position unit or speed unit setting of NC Y axis is invalid.	For NC operation, set the unit of the NC channel configuration axis to mm or deg. Set the unit of speed to unit/min.
3543	13635	The position unit or speed unit setting of NC Z axis is invalid.	For NC operation, set the unit of the NC channel configuration axis to mm or deg. Set the unit of speed to unit/min.
3544	13636	The position unit or speed unit setting of NC A axis is invalid.	For NC operation, set the unit of the NC channel configuration axis to mm or deg. Set the unit of speed to unit/min.
3545	13637	The position unit or speed unit setting of NC B axis is invalid.	For NC operation, set the unit of the NC channel configuration axis to mm or deg. Set the unit of speed to unit/min.
3546	13638	The position unit or speed unit setting of NC C axis is invalid.	For NC operation, set the unit of the NC channel configuration axis to mm or deg. Set the unit of speed to unit/min.

Error code		Error Description	Solutions
Hex	Dec		
3547	13639	The position unit or speed unit setting of NC U axis is invalid.	For NC operation, set the unit of the NC channel configuration axis to mm or deg. Set the unit of speed to unit/min.
3548	13640	The position unit or speed unit setting of NC V axis is invalid.	For NC operation, set the unit of the NC channel configuration axis to mm or deg. Set the unit of speed to unit/min.
3549	13641	The position unit or speed unit setting of NC W axis is invalid.	For NC operation, set the unit of the NC channel configuration axis to mm or deg. Set the unit of speed to unit/min.
354A	13642	The position unit or speed unit setting of NC S axis is invalid.	For NC operation, set the speed unit of the spindle axis to RPM.
3600	13824	The program cannot be loaded because there is no program set in NC program designation command in the controller.	Execute the program designation command (NC_LoadProgram) again after writing the NC program created in XG5000 in the controller.
3601	13825	The program designation command cannot be executed if NC channel is in automatic operation.	Check whether the NC channel is currently running automatically. You can designate a new program after the automatic operation ends.
3602	13826	The program designation command cannot be executed due to the NC program data abnormality.	Download the data again from XG5000 after checking whether NC program data abnormal error (0x3001) occurs and place requests for A/S if it occurs repeatedly after re-execution.
3603	13827	The LoadMode of the NC program designation command was invalid.	Execute the program designation command (NC_LoadProgram) again after entering a value of 0 in the LoadMode of the NC program designation command(NC_LoadProgram).
3604	13828	The NC_LoadProgram command cannot be executed when it is resetting by NC_Reset or NC_Emergency command.	Execute the NC command after NC_Reset or NC_Emergency command ends.
3610	13840	The NC channel interpreter(IPR) was not reset normally.	Rest the NC channel with the NC reset command (NC_Reset). Place requests for A/S if it occurs repeatedly after re-execution.
3620	13856	The NC_Emergency command was executed in a state other than automatic operation.	Execute the NC emergency stop command (NC_Emergency) when the NC channel is in automatic operation.
3630	13872	It is not the range of origin that can be specified in NC homing operation.	The range of the origin(ReferenceNum) is from the first origin to the fourth origin. Specify it as a value between 1 and 4.
3631	13873	The NC homing command cannot be executed when the channel is in automatic operation.	Execute the homing command after the automatic operation is completed.
3632	13874	The homing command cannot be executed because the NC emergency stop command is in Enable state.	Execute the homing command (NC_Home) again after releasing the Enable input of the NC emergency stop command (NC_Emergency)
3633	13875	The homing command cannot be executed because the NC channel configuration axis is not ready.	Execute the homing command(NC_Home) with all the NC channel configuration axes ready. In order to perform the homing command, the NC channel configuration axis should be connected to the EtherCAT network or set as a virtual axis.
3634	13876	The homing command cannot be executed because the NC channel configuration axis is enabled as a motion axis group configuration axis.	Execute the homing command (NC_Home) with the NC channel configuration axis disabled as a motion axis group configuration axis.
3635	13877	Error occurred during homing operation in the servo drive.	Check the error factor of the servo drive and perform homing after removing the servo drive error.

Error code		Error Description	Solutions
Hex	Dec		
3636	13878	The NC_Home command cannot be executed when it is resetting by NC_Reset or NC_Emergency command.	Execute the NC command after NC_Reset or NC_Emergency command ends.
3640	13888	NC M-code operation completion command cannot be executed when M-code output Strobe signal is off.	Execute the M-code operation completion command (NC_McodeComplete) with the M-code output Strobe signal on after checking the status of the M-code output Strobe signal among the NC channel flags.
3641	13889	The NC_McodeComplete command cannot be executed when it is resetting by NC_Reset or NC_Emergency command.	Execute the NC command after NC_Reset or NC_Emergency command ends.
3650	13904	NC S-code operation completion command cannot be executed when S-code output Strobe signal is off.	Execute the S-code operation completion command (NC_McodeComplete) with the M-code output Strobe signal on after checking the status of the M-code output Strobe signal among the NC channel flags.
3651	13905	The NC_ScodeComplete command cannot be executed when it is resetting by NC_Reset or NC_Emergency command.	Execute the NC command after NC_Reset or NC_Emergency command ends.
3660	13920	NC T-code operation completion command cannot be executed when T-code output Strobe signal is off.	Execute the T-code operation completion command (NC_McodeComplete) with the T-code output Strobe signal on after checking the status of the T-code output Strobe signal among the NC channel flags.
3661	13921	The NC_TcodeComplete command cannot be executed when it is resetting by NC_Reset or NC_Emergency command.	Execute the NC command after NC_Reset or NC_Emergency command ends.
3670	13936	The NC parameter write command cannot be executed when the channel is in automatic operation.	Check whether NC channel is currently in automatic operation. Execute the NC parameter write command (NC_WriteParameter) in stop state after the automatic operation ends.
3671	13937	The setting axis of the NC parameter write command was not enabled as the NC axis.	Check whether the setting axis of the NC parameter write command was registered as NC channel/axis parameter. The NC channel/axis can be registered in the NC channel parameter among the motion data items of XG5000.
3672	13938	The axis setting value of the NC parameter write command exceeded the allowable range.	Set the axis number to a value between 0 and 10. Perform channel parameter writing if the axis value is 0 and NC axis X ~ NC axis S writing if 1 ~ 10.
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 ~ 17 for the channel parameter, and 1 ~ 5 for the channel/axis parameter. Check the group number to which the parameter you want to write belongs, and then execute the parameter write command(NC_WriteParameter).
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 channel parameter or channel/axis parameter group is supported. Check the group number to which the parameter you want to write belongs and the parameter number, and then execute the parameter write command(NC_WriteParameter).

Error code		Error Description	Solutions
Hex	Dec		
3675	13941	The data setting value of the parameter set in the NC parameter write command exceeded the range.	Check the data setting range of the parameter to be set, and then execute the parameter write command(NC_WriteParameter) with a value within the range.
3800	14336	In the NC rapid traverse command, the operation speed value of the configuration axis exceeded the speed limit value.	Set the rapid traverse rate within the range that does not exceed the speed limit value after checking the speed limit value of the axis connected to the NC configuration axis where error occurred.
3801	14337	In the NC rapid traverse command, the operation speed value of NC X axis exceeded the speed limit value.	Set the rapid traverse rate within the range that does not exceed the speed limit value after checking the speed limit value of the axis connected to NC X axis where error occurred.
3802	14338	In the NC rapid traverse command, the operation speed value of NC Y axis exceeded the speed limit value.	Set the rapid traverse rate within the range that does not exceed the speed limit value after checking the speed limit value of the axis connected to NC Y axis where error occurred.
3803	14339	In the NC rapid traverse command, the operation speed value of NC Z axis exceeded the speed limit value.	Set the rapid traverse rate within the range that does not exceed the speed limit after checking the speed limit value of the axis connected to NC Z axis.
3804	14340	In the NC rapid traverse command, the operation speed value of NC A axis exceeded the speed limit value.	Set the rapid traverse rate within the range that does not exceed the speed limit after checking the speed limit value of the axis connected to NC A axis.
3805	14341	In the NC rapid traverse command, the operation speed value of NC B axis exceeded the speed limit value.	Set the rapid traverse rate within the range that does not exceed the speed limit after checking the speed limit value of the axis connected to NC B axis.
3806	14342	In the NC rapid traverse command, the operation speed value of NC C axis exceeded the speed limit value.	Set the rapid traverse rate within the range that does not exceed the speed limit after checking the speed limit value of the axis connected to NC C axis.
3807	14343	In the NC rapid traverse command, the operation speed value of NC U axis exceeded the speed limit value.	Set the rapid traverse rate within the range that does not exceed the speed limit after checking the speed limit value of the axis connected to NC U axis.
3808	14344	In the NC rapid traverse command, the operation speed value of NC V axis exceeded the speed limit value.	Set the rapid traverse rate within the range that does not exceed the speed limit after checking the speed limit value of the axis connected to NC V axis.
3809	14345	In the NC rapid traverse command, the operation speed value of NC W axis exceeded the speed limit value.	Set the rapid traverse rate within the range that does not exceed the speed limit after checking the speed limit value of the axis connected to NC W axis.
380A	14346	In the NC rapid traverse command, the operation speed value of NC S axis exceeded the speed limit value.	Set the rapid traverse rate within the range that does not exceed the speed limit after checking the speed limit value of the axis connected to NC S axis.
380B	14347	In-position of rapid traverse component axes was not completed within in-position completion monitoring time during the NC rapid traverse operation.	Check the command in-position width and the in-position completion monitoring time of the NC channel parameter among the parameters connected to the NC axis.
380C	14348	The automatic operation cannot continue because there is an axis that has an error among NC configuration axes during the NC rapid traverse operation.	Check the axis where the error occurred among the NC configuration axes. You can check the error that occurred on the NC axis in the axis error code number of the NC channel/axis flag.

Error code		Error Description	Solutions
Hex	Dec		
3810	14352	In NC feed per rotation mode, the cutting feed operation rate was set to 0.	Set the rate of the cutting feed operation to a value other than 0 in NC feed per rotation mode.
3811	14353	The operation speed of NC cutting feed command exceeded the cutting feed upper limit rate value of the NC channel parameter.	Set the cutting feed rate value(F) within the range that does not exceed the parameter value after checking the cutting feed upper limit rate value of the NC channel parameter.
3812	14354	The in-position of the cutting feed configuration axes was not completed within the in-position completion monitoring time during the NC cutting feed operation.	Check the command in-position width and the in-position completion monitoring time of the NC channel parameter among the parameters connected to the NC axis.
3820	14368	In the NC cutting feed command, the operation speed value of the configuration axis exceeded the speed limit value.	Set the cutting feed rate within the range that does not exceed the speed limit value after checking the speed limit value of the axis connected to the NC configuration axis where error occurred.
3821	14369	In the NC cutting feed command, the operation speed value of NC X axis exceeded the speed limit value.	Set the cutting feed rate within the range that does not exceed the speed limit value after checking the speed limit value of the axis connected to NC X axis.
3822	14370	In the NC cutting feed command, the operation speed value of NC Y axis exceeded the speed limit value.	Set the cutting feed rate within the range that does not exceed the speed limit value after checking the speed limit value of the axis connected to NC Y axis.
3823	14371	In the NC cutting feed command, the operation speed value of NC Z axis exceeded the speed limit value.	Set the cutting feed rate within the range that does not exceed the speed limit value after checking the speed limit value of the axis connected to NC Z axis.
3824	14372	In the NC cutting feed command, the operation speed value of NC A axis exceeded the speed limit value.	Set the cutting feed rate within the range that does not exceed the speed limit value after checking the speed limit value of the axis connected to NC A axis.
3825	14373	In the NC cutting feed command, the operation speed value of NC B axis exceeded the speed limit value.	Set the cutting feed rate within the range that does not exceed the speed limit value after checking the speed limit value of the axis connected to NC B axis.
3826	14374	In the NC cutting feed command, the operation speed value of NC C axis exceeded the speed limit value.	Set the cutting feed rate within the range that does not exceed the speed limit value after checking the speed limit value of the axis connected to NC C axis.
3827	14375	In the NC cutting feed command, the operation speed value of NC U axis exceeded the speed limit value.	Set the cutting feed rate within the range that does not exceed the speed limit value after checking the speed limit value of the axis connected to NC U axis.
3828	14376	In the NC cutting feed command, the operation speed value of NC V axis exceeded the speed limit value.	Set the cutting feed rate within the range that does not exceed the speed limit value after checking the speed limit value of the axis connected to NC V axis.
3829	14377	In the NC cutting feed command, the operation speed value of NC W axis exceeded the speed limit value.	Set the cutting feed rate within the range that does not exceed the speed limit value after checking the speed limit value of the axis connected to NC W axis.
382A	14378	In the NC cutting feed command, the operation speed value of NC S axis exceeded the speed limit value.	Set the cutting feed rate within the range that does not exceed the speed limit value after checking the speed limit value of the axis connected to NC S axis.
3840	14400	In NC circular interpolation, operation cannot be executed in case of start point = center point or center point = end point.	In NC circular interpolation, set the position of the center point to a different position from the start point(or end point).

Error code		Error Description	Solutions
Hex	Dec		
3841	14401	The radius setting was incorrect in NC circular interpolation.	The radius value of the circle where the NC circular interpolation operation can be executed is greater than 0 and less than or equal to 2,147,483,647pulse based on the pulse unit. Set the center point or radius input value so that the radius can be calculated within the setting range.
3850	14416	Axis designation was incorrect in NC cylindrical interpolation.	In performing circular interpolation operation in NC cylindrical interpolation, Y-axis should be designated in the XY plane, Z-axis in the YZ plane, and Z-axis in the ZX plane.
3860	14432	The rest method is specified as the number of rotations, but the number of rotations is 0.	Run the S-axis with MC_MoveVelocity in the NC program.
3F00	16128	Interpreter (IPR) parsing error - LEX MAIN TABLE configuration is invalid.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command (NC_Reset).
3F01	16129	Interpreter(IPR) parsing error - Undefined character exists.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F02	16130	Interpreter (IPR) parsing error - Number exceeded the maximum buffer.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F03	16131	Interpreter(IPR) parsing error - The number of LEX tokens exceeded the maximum buffer.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F04	16132	Interpreter(IPR) parsing error - There are one or more decimal points.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F05	16133	Interpreter (IPR) parsing error - The number of brackets in the formula is incorrect.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F06	16134	Interpreter(IPR) parsing error - There exist characters that cannot be used in the formula.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F07	16135	Interpreter(IPR) parsing error - The syntax of the formula is incorrect.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F08	16136	Interpreter (IPR) parsing error - It is not a permitted macro variable.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).

Error code		Error Description	Solutions
Hex	Dec		
3F09	16137	Interpreter (IPR) parsing error - It is a TANGENT operation error.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F0A	16138	Interpreter(IPR) parsing error - It is a SQUARE ROOT operation error.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F0B	16139	Interpreter(IPR) parsing error - The denominator of division cannot be 0.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F0C	16140	Interpreter(IPR) parsing error - Syntax is incorrect.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F0D	16141	Interpreter(IPR) parsing error - YACC MAIN TABLE configuration is invalid.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F0E	16142	Interpreter(IPR) parsing error - The number of YACC tokens exceeded the maximum buffer.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F0F	16143	Interpreter (IPR) parsing error - It is not possible to open IPR semaphore.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F10	16144	Interpreter(IPR) parsing error - It was terminated without M02 or M30.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F11	16145	Interpreter(IPR) parsing error - It can be commanded only at the head of the block.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F12	16146	Interpreter(IPR) parsing error - The same progress block exists.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F13	16147	Interpreter(IPR) parsing error - The number of statement numbers exceeded the maximum buffer.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).

Error code		Error Description	Solutions
Hex	Dec		
3F14	16148	Interpreter (IPR) parsing error - It is not possible to find the next block to proceed with.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F15	16149	Interpreter(IPR) parsing error - Subprogram call syntax is incorrect.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F16	16150	Interpreter(IPR) parsing error - It exceeded the maximum subprogram call.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F17	16151	Interpreter(IPR) parsing error - It is a program that was already called.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F18	16152	Interpreter (IPR) parsing error - There is no M99 in the subprogram.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F19	16153	Interpreter (IPR) parsing error - M99 syntax is incorrect.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F1A	16154	Interpreter (IPR) parsing error - There are a large number of loops.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F1B	16155	Interpreter(IPR) parsing error - There is no start of loop.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F1C	16156	Interpreter(IPR) parsing error - The connection of loops is invalid.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F1D	16157	Interpreter(IPR) parsing error - It exceeded M command limit in one block.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F1E	16158	Interpreter (IPR) parsing error - It is an unused G code.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).

Error code		Error Description	Solutions
Hex	Dec		
3F1F	16159	Interpreter (IPR) parsing error - It is not possible to be commanded simultaneously in one block.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F20	16160	Interpreter(IPR) parsing error - The center point of the arc cannot be found.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F21	16161	Interpreter(IPR) parsing error - It is not possible to create a path of cycle code.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F22	16162	Interpreter(IPR) parsing error - The taper amount of the cycle code is too large.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F23	16163	Interpreter(IPR) parsing error - It is not possible to be commanded within a cycle shape block.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F24	16164	Interpreter(IPR) parsing error - There is a problem with the cycle shape block command.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F26	16166	Interpreter(IPR) parsing error - The tool offset number is not valid.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F27	16167	Interpreter(IPR) parsing error - The center point of the arc is not correct.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F28	16168	Interpreter(IPR) parsing error - It is not possible to call the subprogram in MDI mode.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F29	16169	Interpreter(IPR) parsing error - Chamfering and rounding can applied only to the cutting feed command.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F2A	16170	Interpreter(IPR) parsing error - Chamfering and rounding are duplicated in instructions.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).

Error code		Error Description	Solutions
Hex	Dec		
3F2B	16171	Interpreter(IPR) parsing error - Only a single axis command is available in chamfering and rounding.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F2C	16172	Interpreter(IPR) parsing error - The chamfering and rounding reference value is greater than the feed amount.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F2D	16173	Interpreter(IPR) parsing error - It is not possible to obtain the next block information in chamfering and rounding.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F2E	16174	Interpreter(IPR) parsing error - An arc in the next block is not allowed in chamfering and rounding.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F2F	16175	Interpreter(IPR) parsing error - Rounding cannot be performed in the same straight line feed.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F30	16176	Interpreter(IPR) parsing error - Correction start and end can be made only in linear feed.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F31	16177	Interpreter(IPR) parsing error - There is no feed command in cycle shape end block.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F32	16178	Interpreter(IPR) parsing error - There is an axis command that is irrelevant to the plane in chamfering and rounding.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F33	16179	Interpreter(IPR) parsing error - It exceeded IJK command limit within one block in calling macro.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F34	16180	Interpreter(IPR) parsing error - Modal macro cannot be called from the subprogram.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F35	16181	Interpreter(IPR) parsing error - It exceeded the multiple call limits of modal macro.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).

Error code		Error Description	Solutions
Hex	Dec		
3F36	16182	Interpreter(IPR) parsing error - It is unused M code.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F37	16183	Interpreter(IPR) parsing error - Pitch cannot be calculated in rigid tapping.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F38	16184	Interpreter(IPR) parsing error - String exceeded the maximum buffer.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F39	16185	Interpreter(IPR) parsing error - String syntax is incorrect.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F3A	16186	Interpreter(IPR) parsing error - It reached the target processing quantity.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F3B	16187	Interpreter(IPR) parsing error - It is user stop of macro program.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F3C	16188	Interpreter(IPR) parsing error - It is not possible to create a path for compound thread cycle.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F3D	16189	Interpreter(IPR) parsing error - It cannot be commanded at polar coordinate interpolation.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F3E	16190	Interpreter(IPR) parsing error - It cannot be traversed to 0 at polar coordinate interpolation.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F3F	16191	Interpreter(IPR) parsing error - It is a syntax error in cylindrical interpolation command.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F40	16192	Interpreter(IPR) parsing error - It cannot be commanded during cylindrical interpolation.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).

Error code		Error Description	Solutions
Hex	Dec		
3F41	16193	Interpreter(IPR) parsing error - It is a constant surface speed control mode in polar coordinate and cylindrical interpolations.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F42	16194	Interpreter(IPR) parsing error - It is not the origin.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3F43	16195	Interpreter(IPR) parsing error - Tool interference occurred.	Check whether there is a program error in the block after confirming the 'error block number' among the NC channel flags. The automatic operation can be executed again after the NC channel is reset with the NC reset command NC_Reset).
3FE0	16352	Program processor(PA) error - There is no corresponding pointer location of the program file.	Execute the automatic operation again after resetting the NC channel with the NC reset command(NC_Reset).
3FE1	16353	Program processor(PA) error - It is not possible to read from the program file.	Execute the automatic operation again after resetting the NC channel with the NC reset command(NC_Reset).
3FE2	16354	Program processor(PA) error - The selected program file does not exist.	Check whether the specified program is stored in the controller. The automatic operation can be executed again after the NC channel is reset with the NC reset command (NC_Reset).
3FE3	16355	Program processor(PA) error - It is not possible to open NcAccess semaphore.	Execute the automatic operation again after resetting the NC channel with the NC reset command(NC_Reset).
3FE4	16356	Program processor(PA) error - The number of characters per block is limited to 300.	Check whether the number of characters per block of the specified program exceeds 300. The automatic operation can be executed again after the NC channel is reset with the NC reset command (NC_Reset).

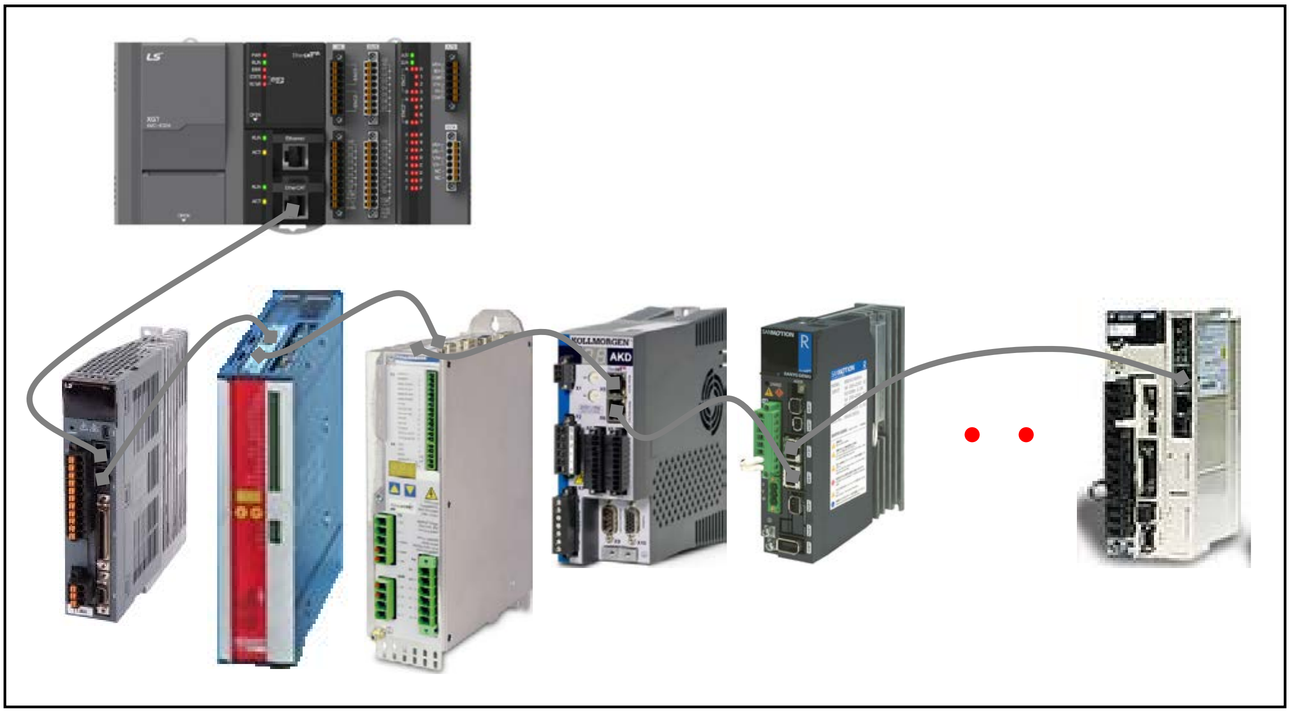
Appendix 3 Setting Example

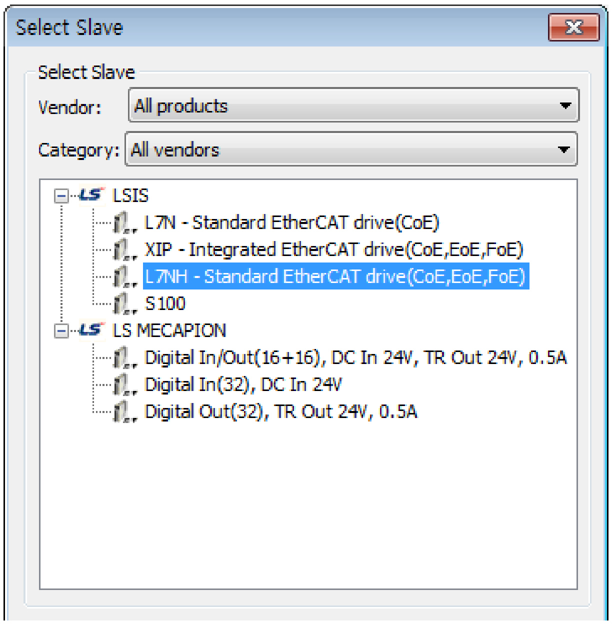
It describes how to set when using the motion controller at the beginning.

(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.





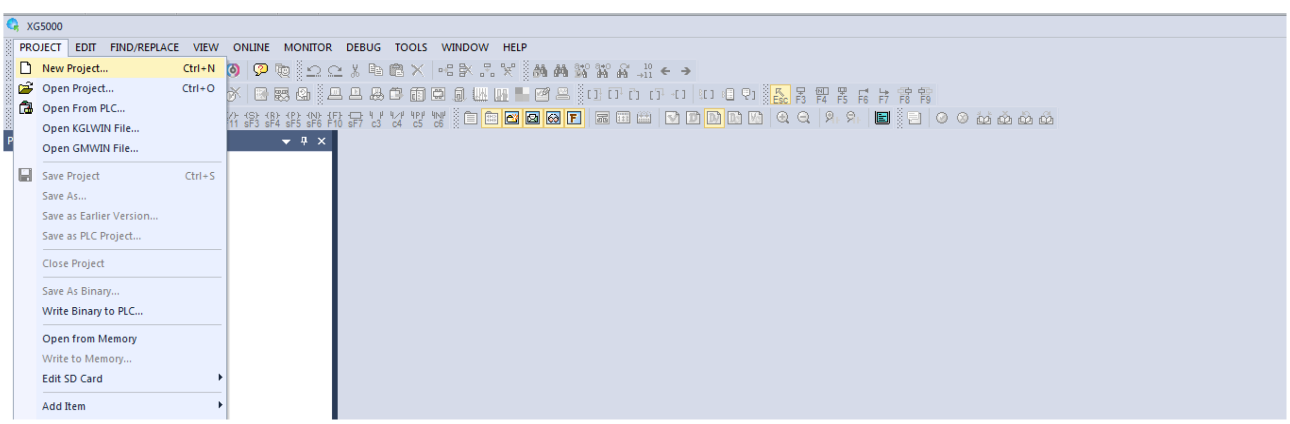
Notes

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.

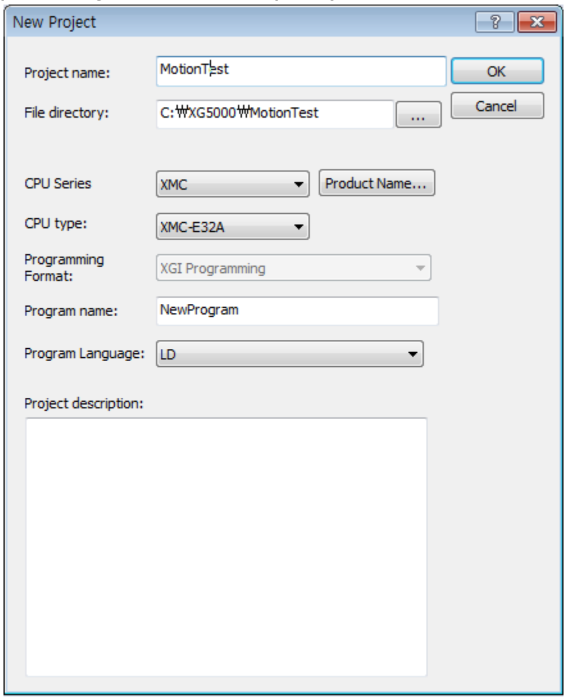
1. Power supply
   Check if the power connected to servo drive and the allowable power conditions are the same.
   (There are instances where no power setting is in parameter depending on the type of servo drive.)
2. The type of motor and encoder(feedback)
   Set the parameter according to the type of 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.

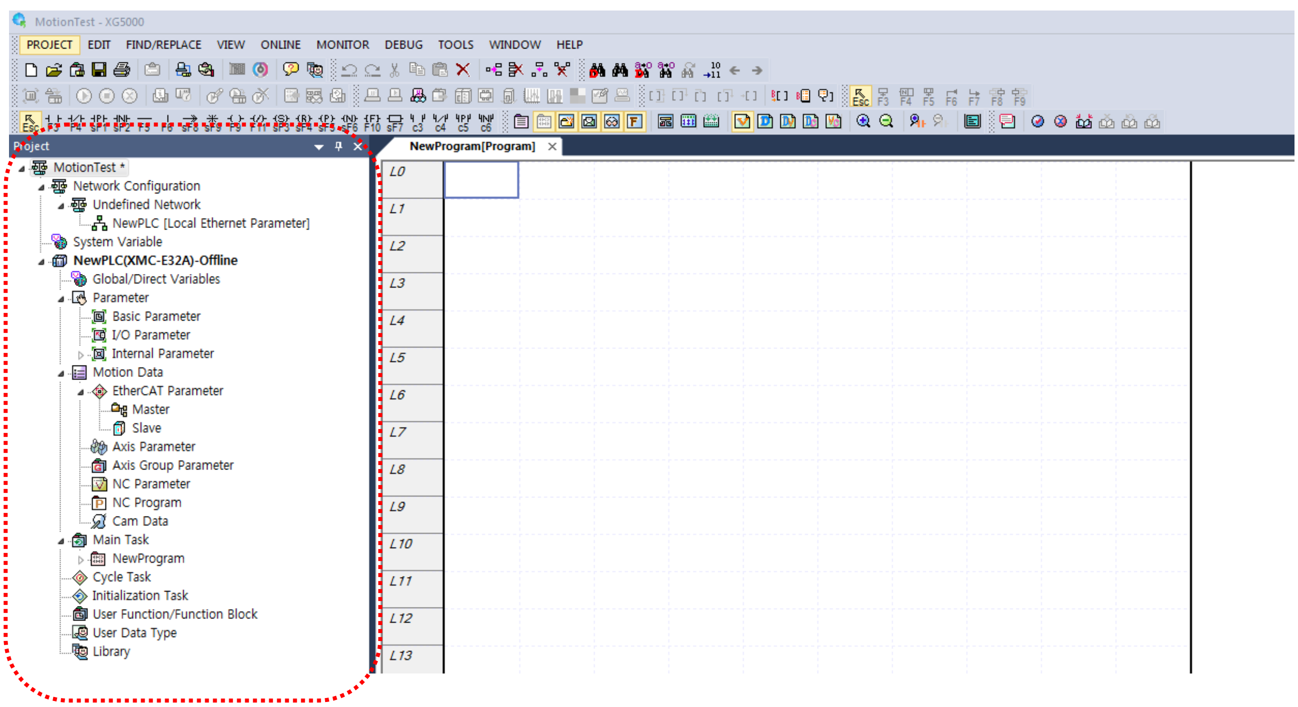
(5) Execute XG5000 and create motion control project by selecting “Project(P) – New Project(N)”.



(6) In the figure below, set up Project name, CPU series, CPU type, Program name to create new project.

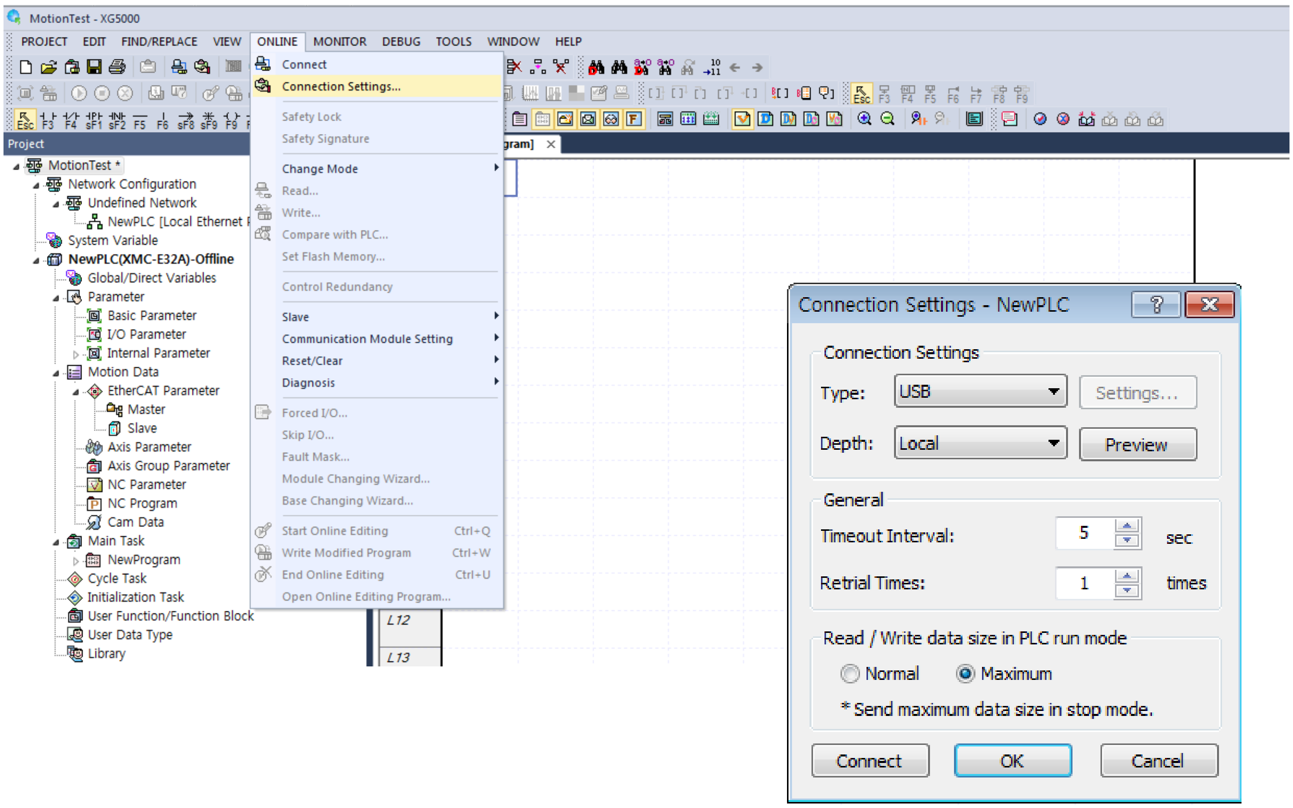


(7) If you set up as the figure above, the project will be created as follows.



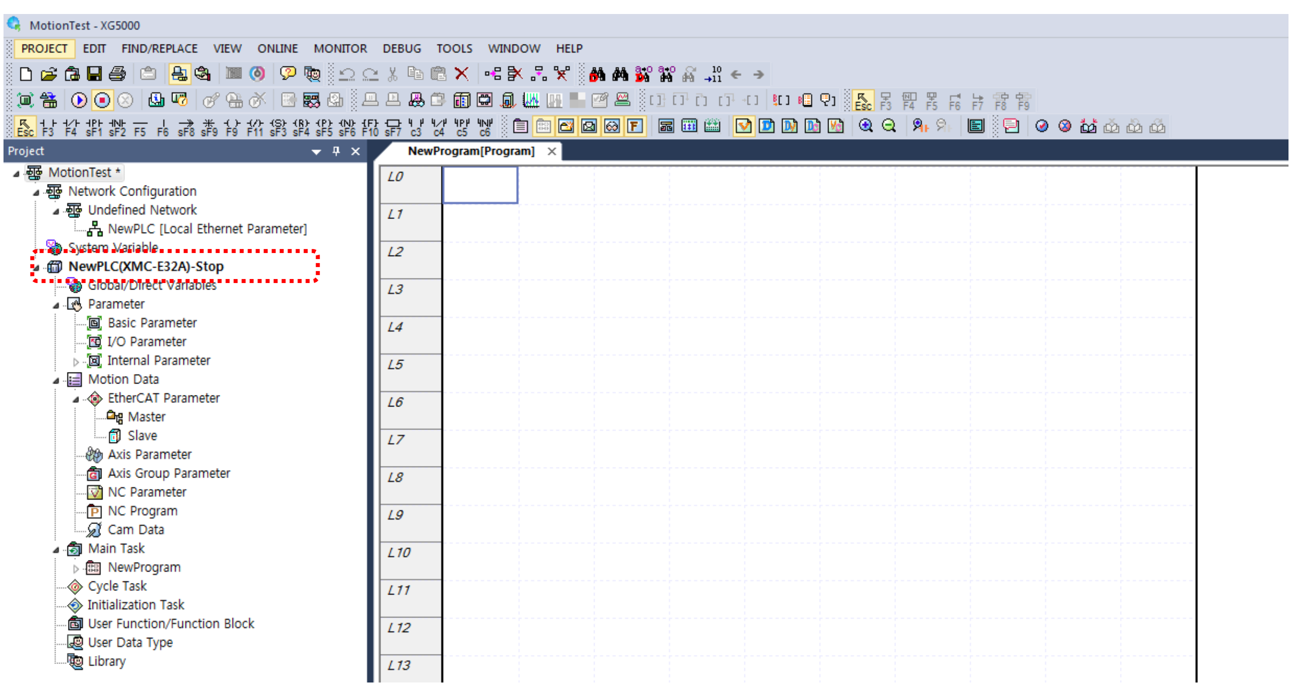
(8) Turn on motion controller and servo driver and connect PC with motion controller through USB or Ethernet cable.

(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.

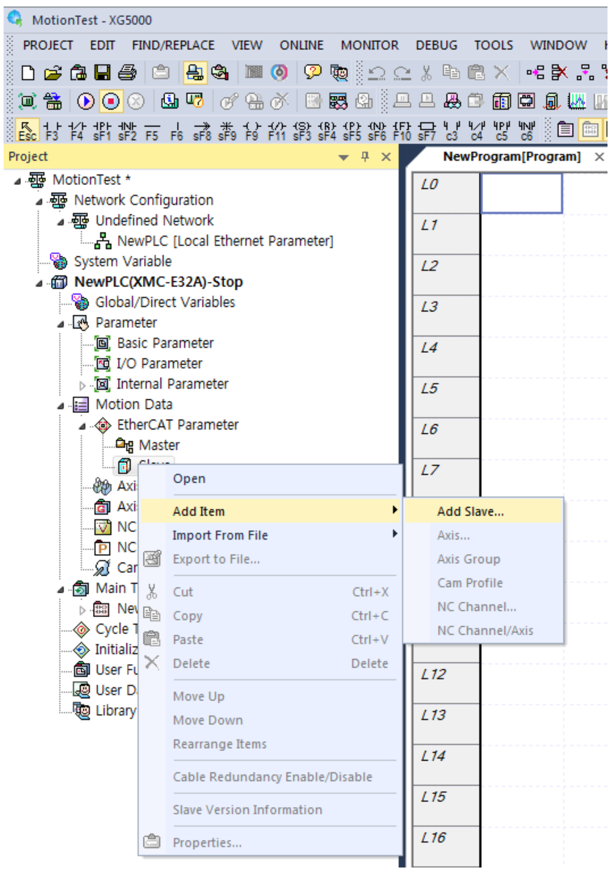


(12) If the controller doesn’t become “Online” and keeps “Offline”, check whether the controller is connected cable, is turned on.

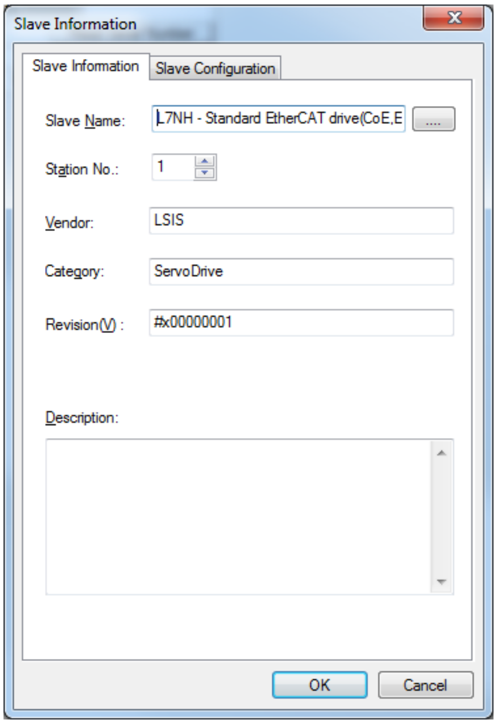
(13) Check if motion controller is in Stop state. If motion controller is in Run state, change it to Stop state and execute the next steps.

(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.

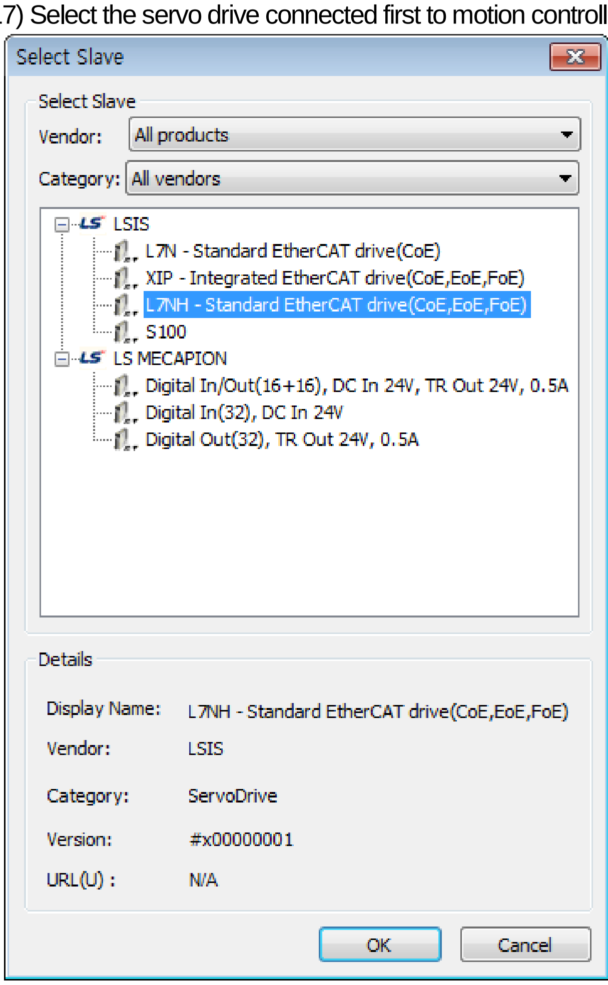
(15) Right click on a mouse in the slave parameter of the project tree and select "Add item – Slave-servo drive" in order to add servo drive to network parameter.



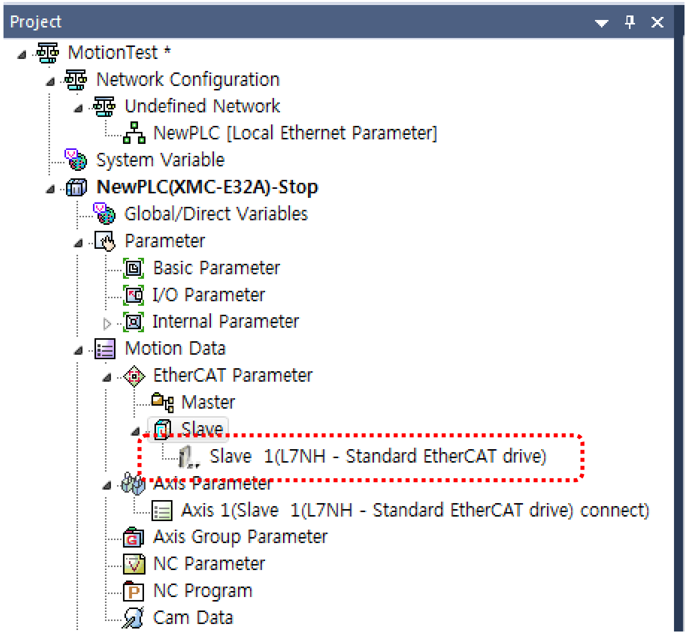
(16) When the slave information window comes up, click the"…" button next to the slave name.



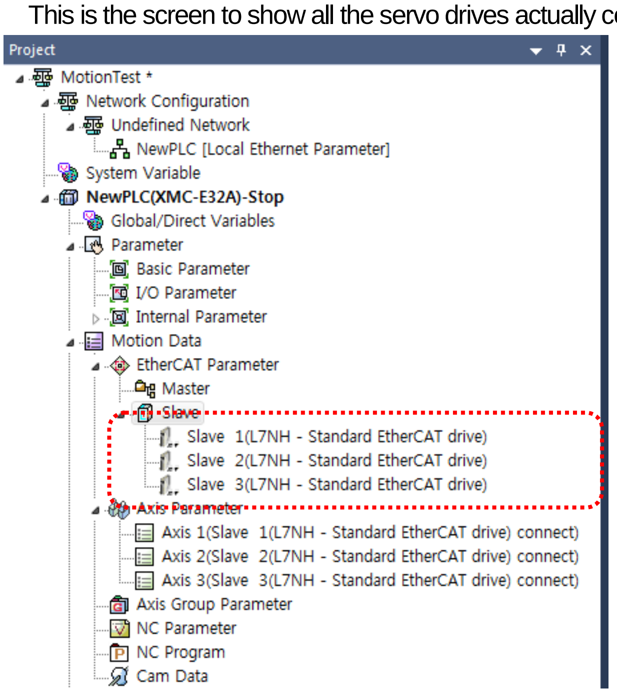
(17) Select the servo drive connected first to motion controller in the servo drive selection window and click OK.



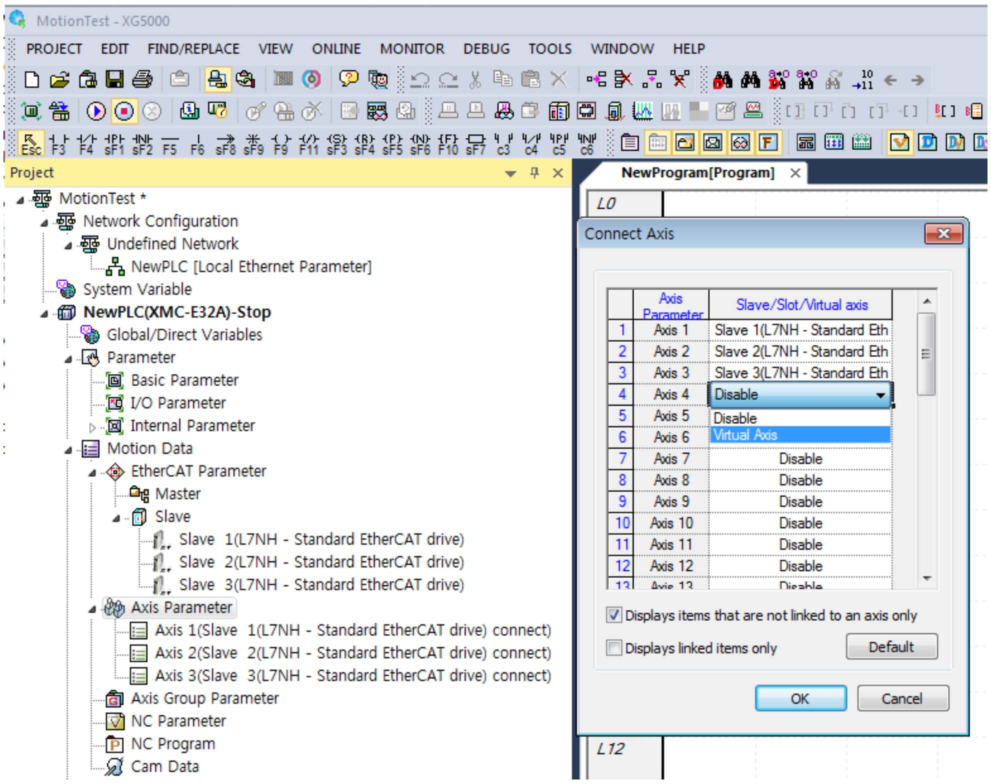
(18) When the axis number setting is completed, the servo drive added earlier is indicated in slave of EtherCAT parameter.



(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.

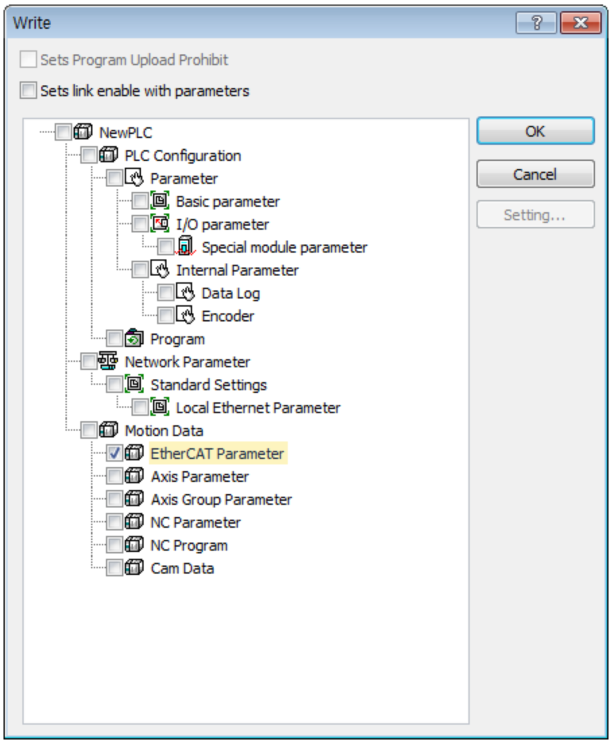


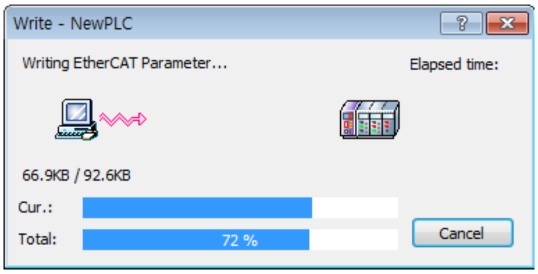
(20) After setting the slave, connect the set slave and the axis to be controlled by the motion controller.
The axes are set in the order that the slave is set but the user can arbitrarily assign slaves to the axes.
Select the axis parameters in the project tree, right-click and select "Axis / slave 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

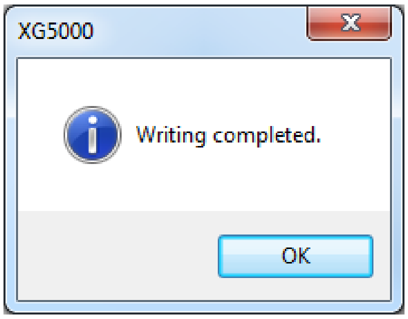


(21) After adding all the EtherCAT salves connected to EtherCAT parameter, execute "Online-Connection" first and execute "Online-Write" to write EtherCAT parameter in motion controller.

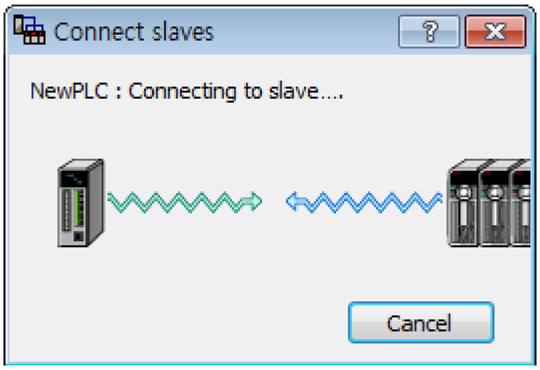
(22) When the project writing window comes up, check in the EtherCAT parameter and check OK to execute writing. This is the screen to show the whole execution process of project writing.



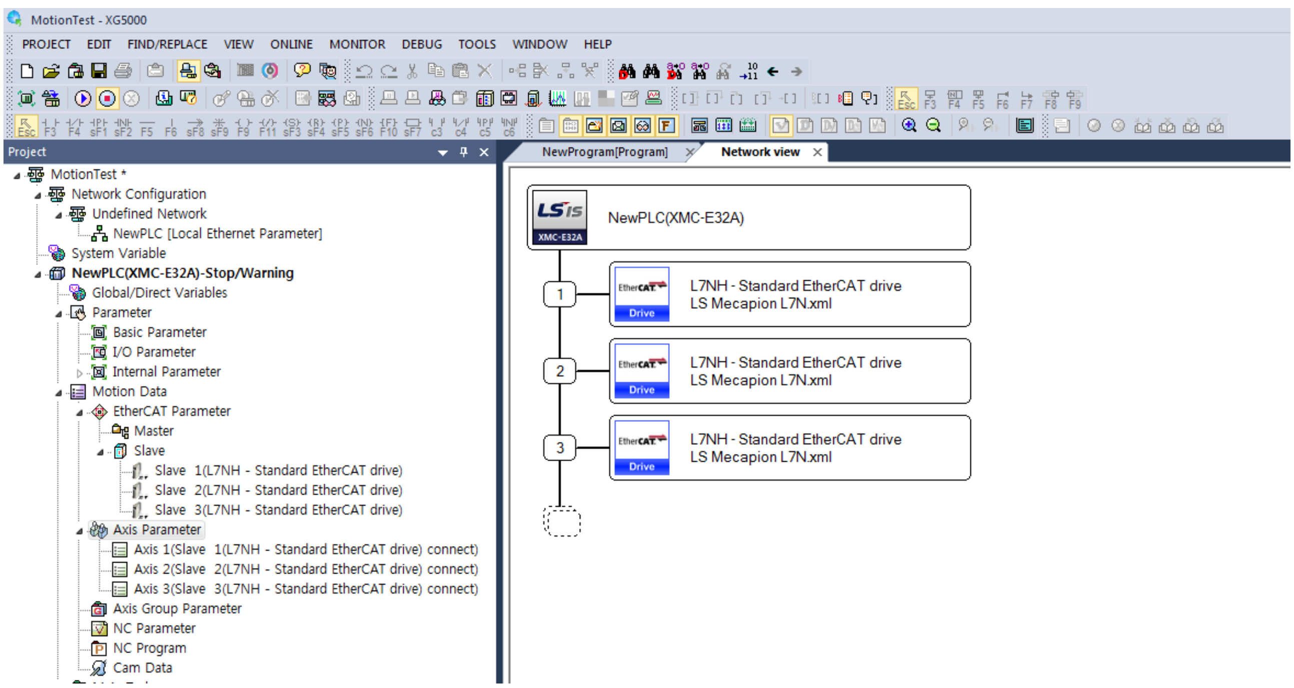




(23) Select "Online – Slave - Connect" to execute communication link between motion controller and servo drive.



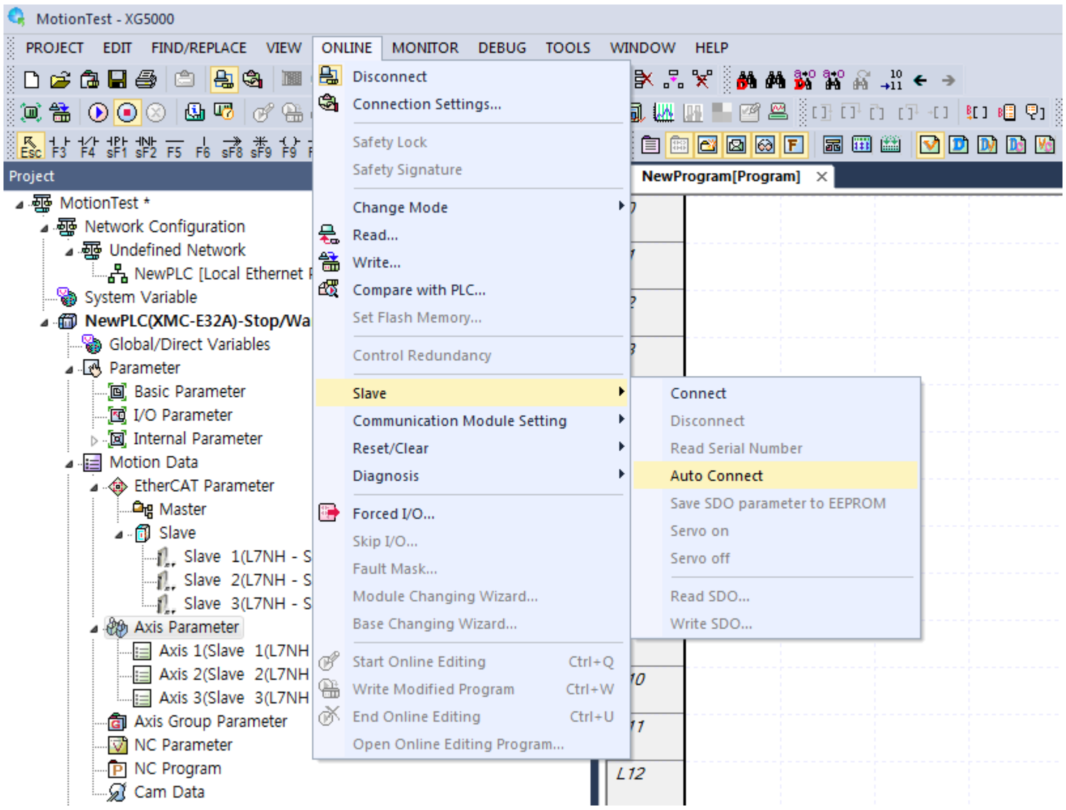
(24) When the link is completed, the servo drive name of slave parameter is activated to black from gray.
Execute the "View – Display EtherCAT Network" in the menu to check the servo drive connection.



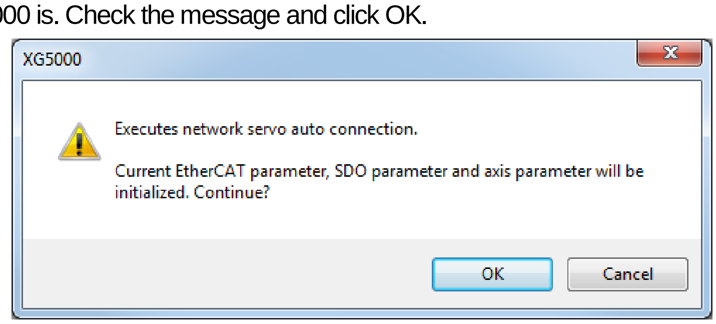
Notes

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.

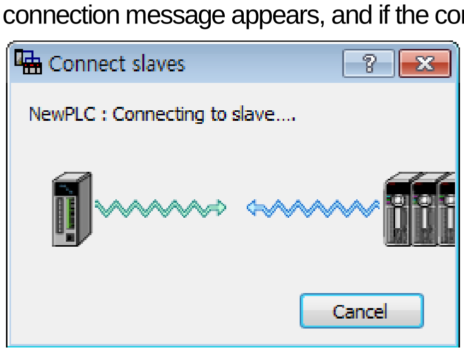
1. Execute the “Online - slave - auto connection” menu.

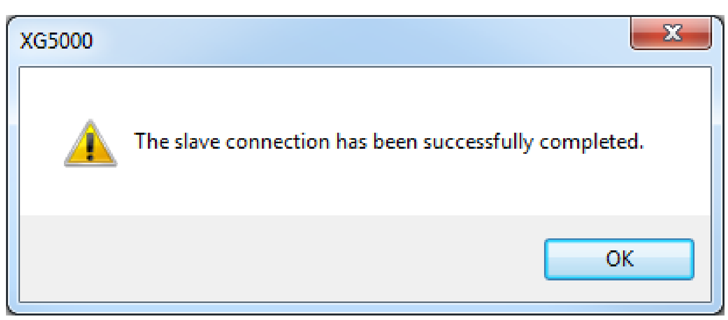


2. Popup notification message appears as follows. This is an alert message notifying when executing slave auto connection, the network parameter set in the current XG5000 and motion controller is initialized and so the servo parameter(SDO parameter) in XG5000 is. Check the message and click OK.



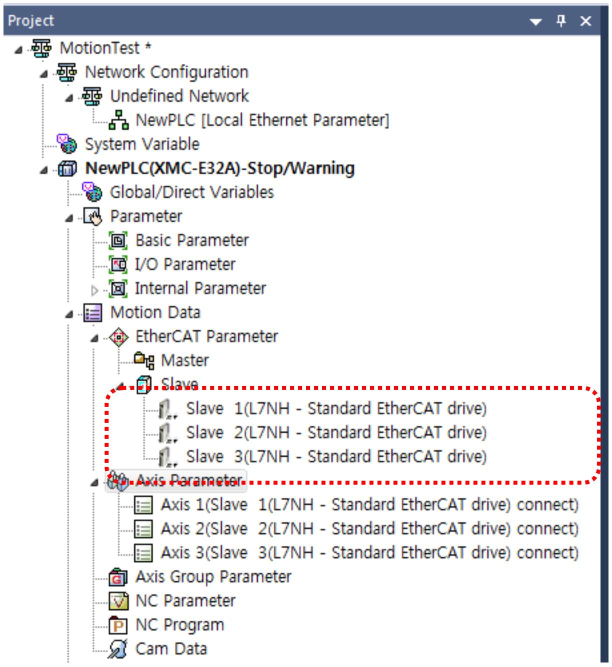
3. Slave connection message appears, and if the connection is completed normally, completion message is indicated.



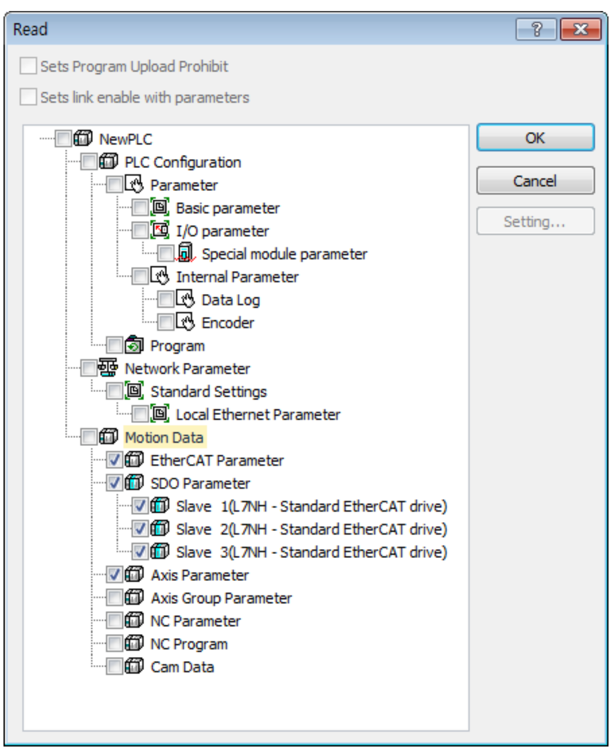


Notes

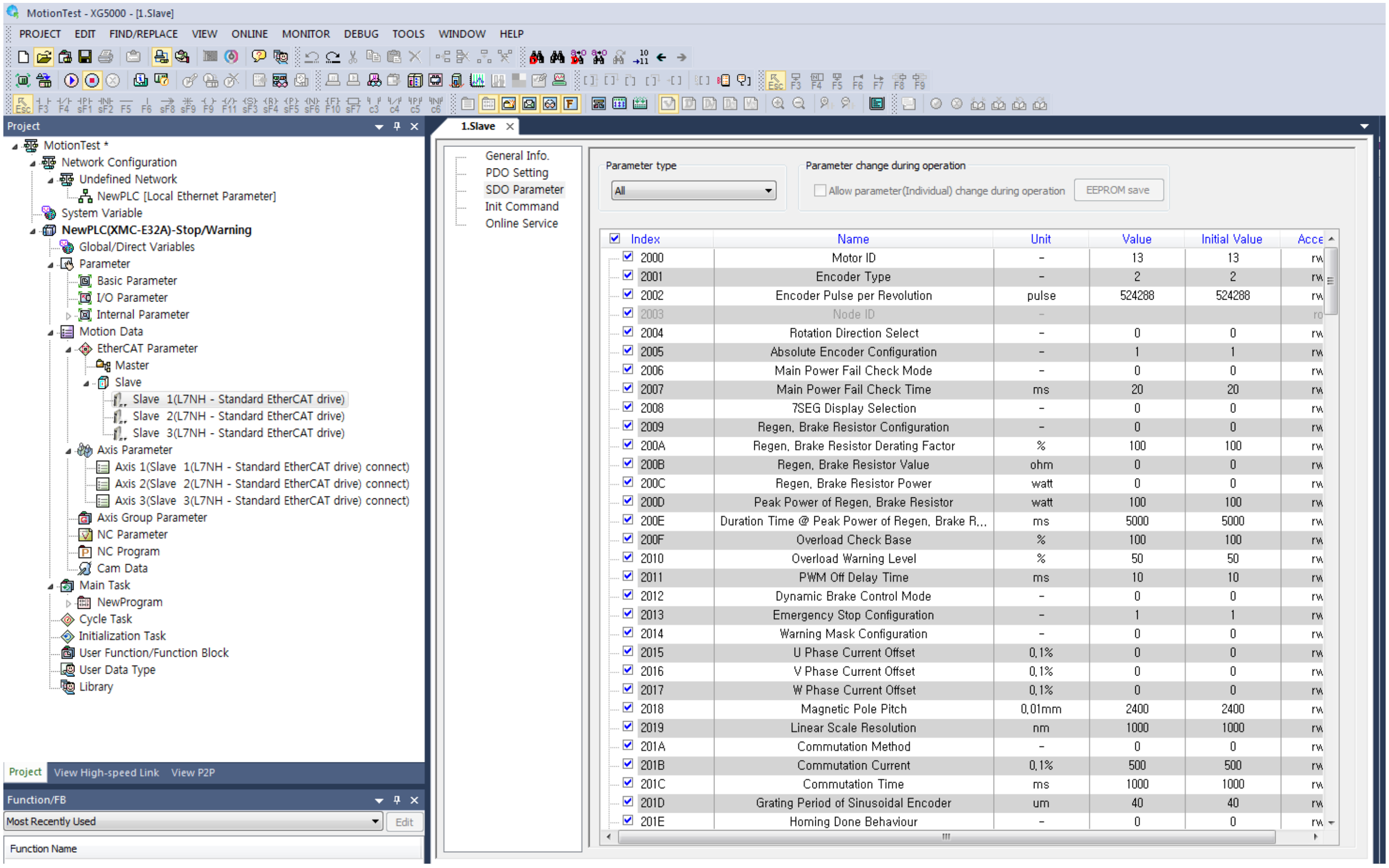
4. When executing the "slave auto connection" command, the EtherCAT slave information currently connected to the EtherCAT parameter slave parameter of XG5000 is automatically registered if the connection command is completed normally.



(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.

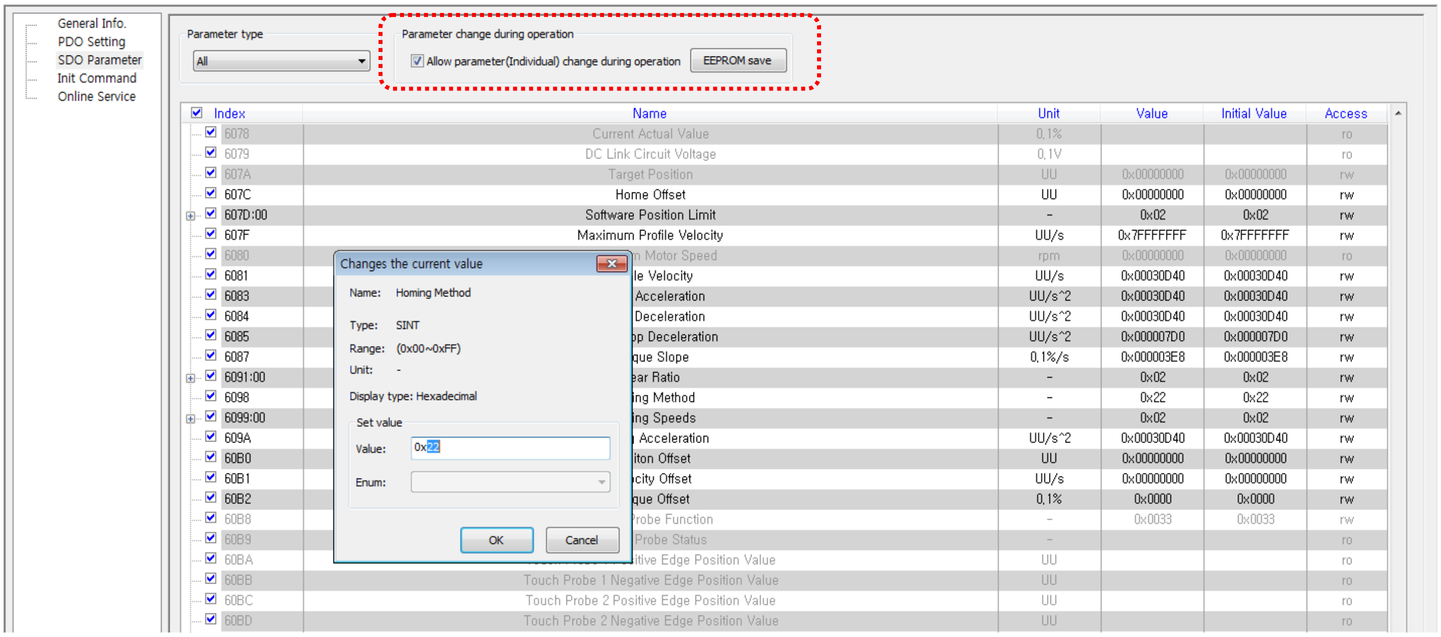


(26) Following is the reading of servo parameter content of L7NH servo drive. The content of servo parameter can differ depending on the types of servo drive. Refer to the instruction manuals of each servo drive for details.



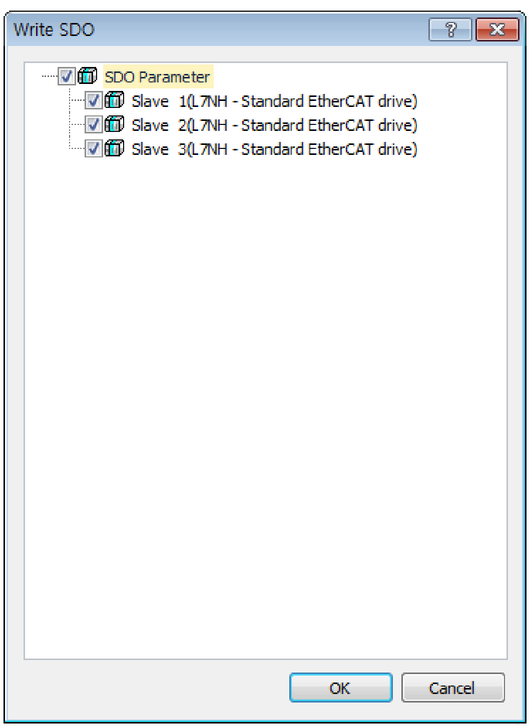
(27) SDO parameter can be set in two ways.

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.



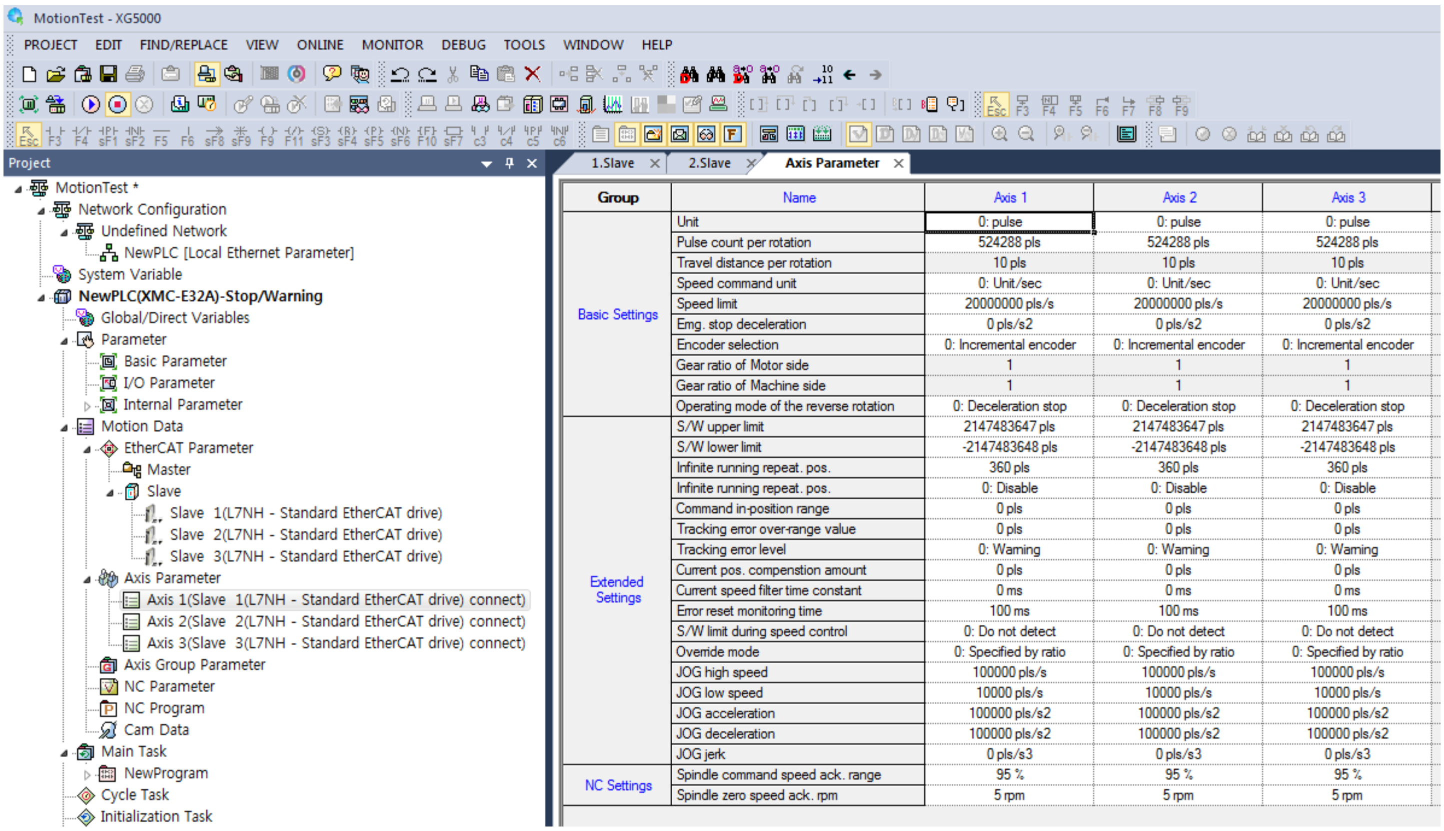
In order to keep the data after turn on/off the power of slave(servo drive), execute the "Online-Save slave parameter to EEPROM" command because modifying the parameter in operation of SDO parameter (individual) is only valid when the power is currently on.

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.

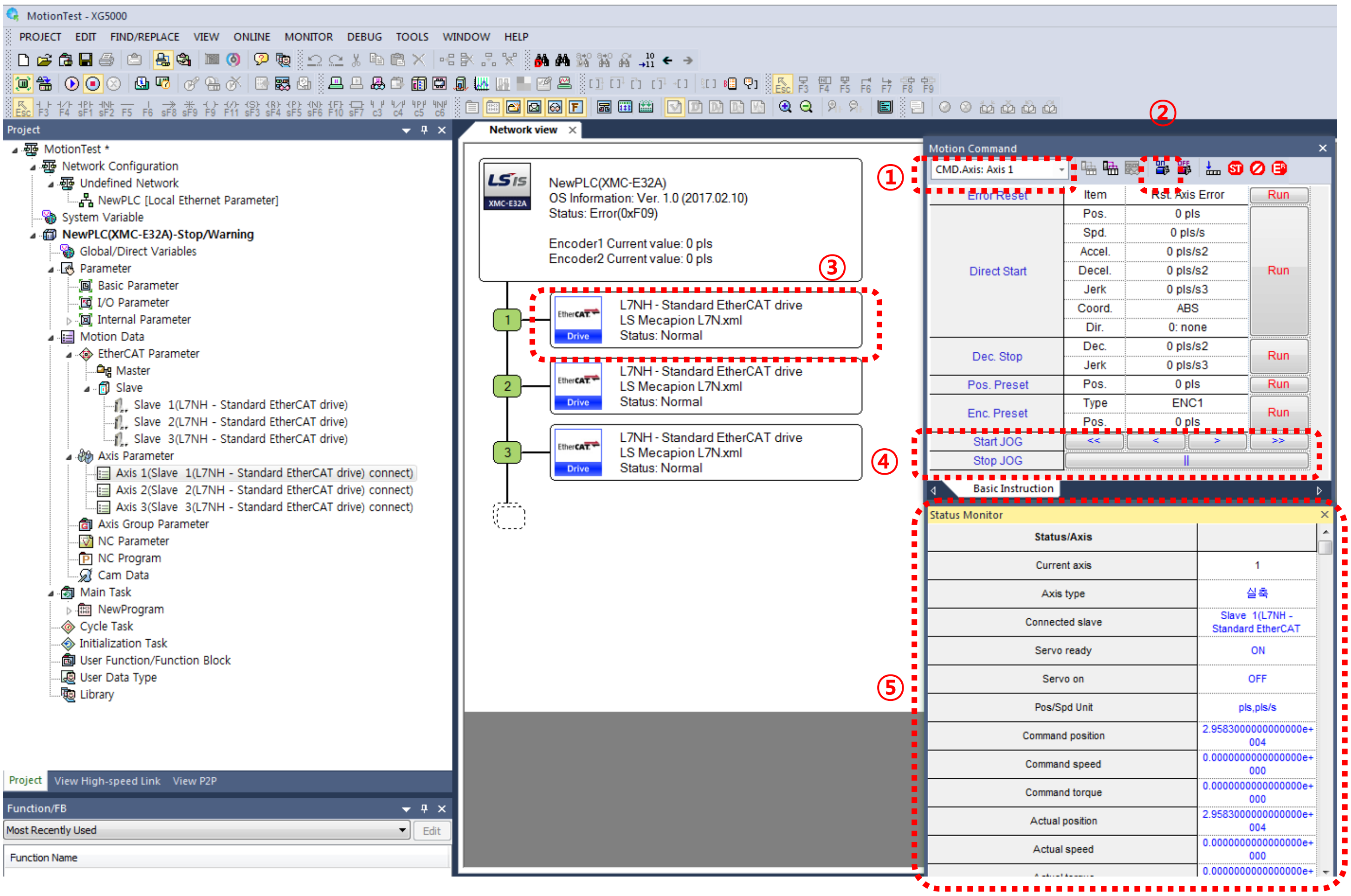
(28) When finishing the SDO parameter setting, set the operation parameter of each axis and select the operation parameter of the relevant axis in "Online-Write " to write in controller.



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.

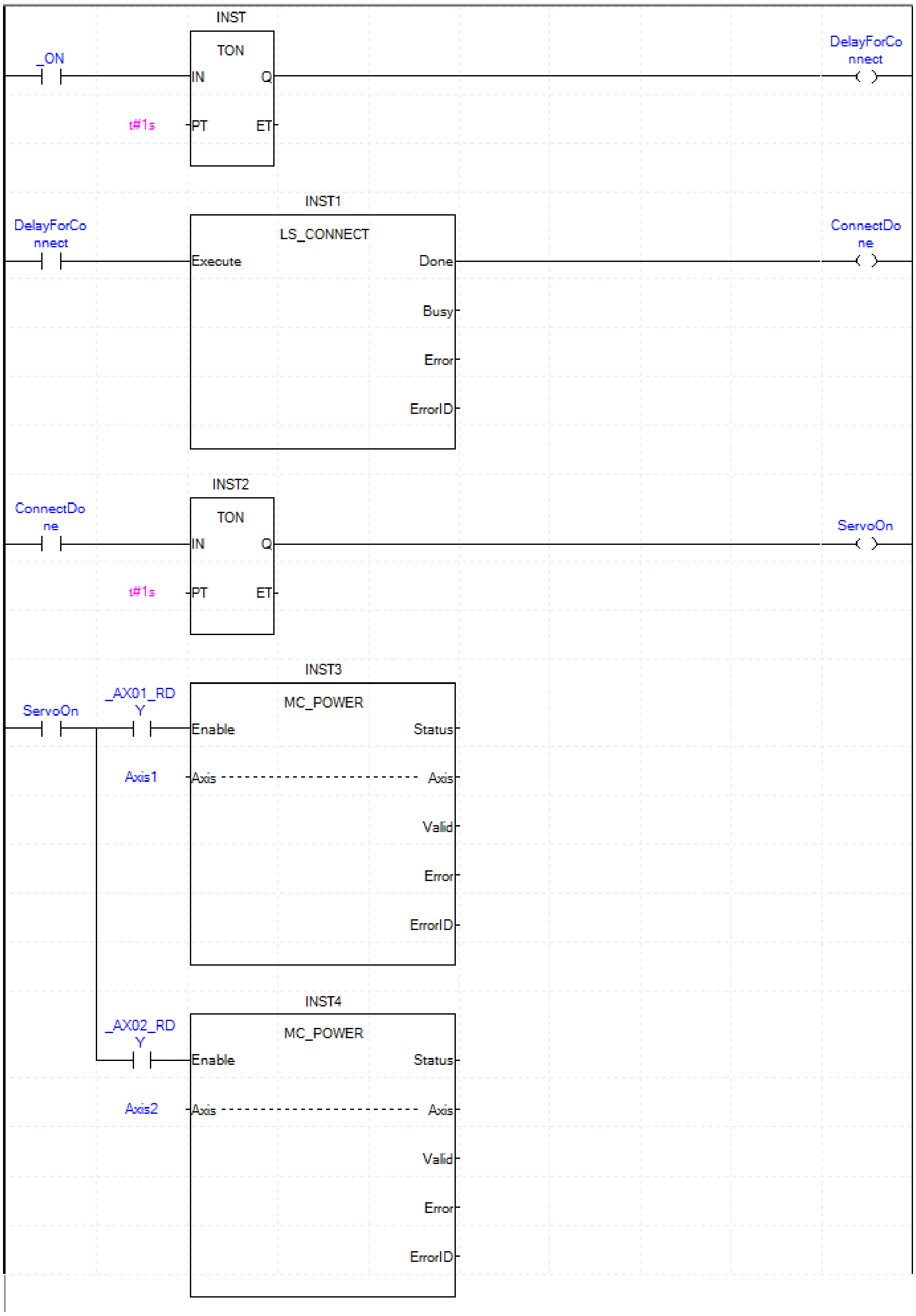


(31) If vibration or noise is generated when motor is operating, adjust the responsibility, inertia ratio, and gain values of servo parameter and transmit them to servo drive. Use the dedicated setting tool of servo drive for detailed setting such as auto tuning.

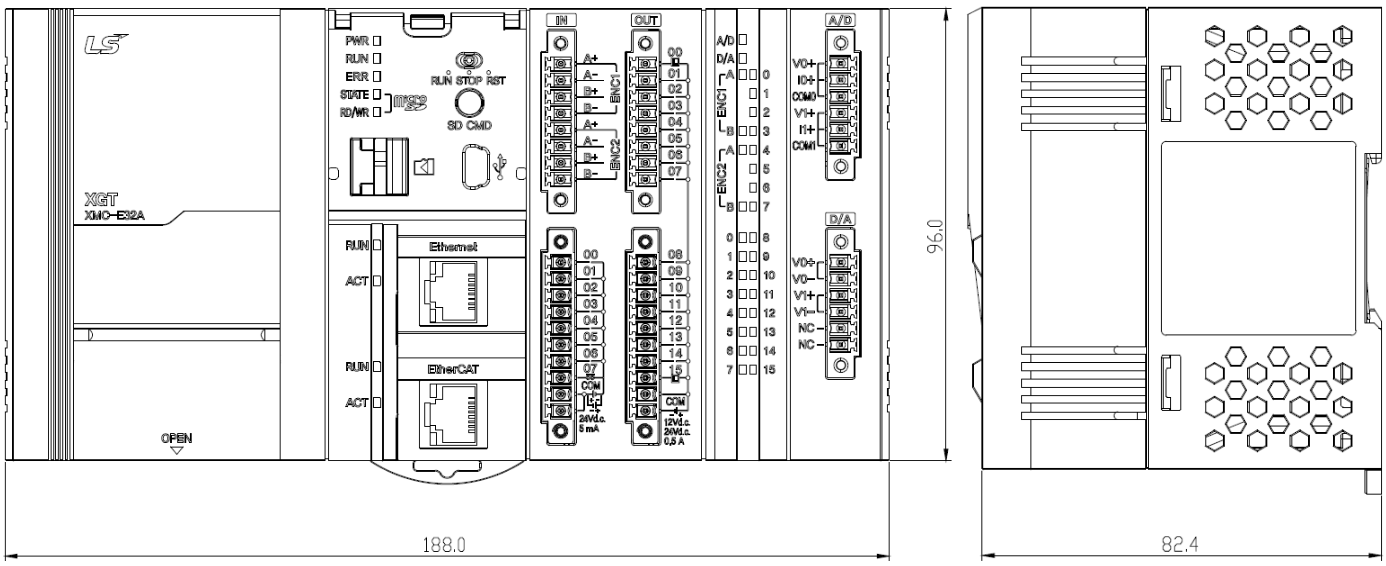
(32) Create motion program.

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.



Appendix 4 Dimension



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.

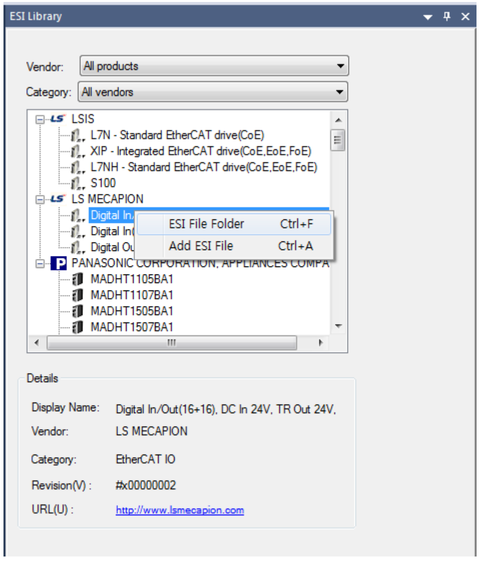
(2) Adding ESI file of EtherCAT slave from other companies

The XG5000 provides functions for adding ESI file of EtherCAT slave from other companies, and searching the installation folder of ESI file.

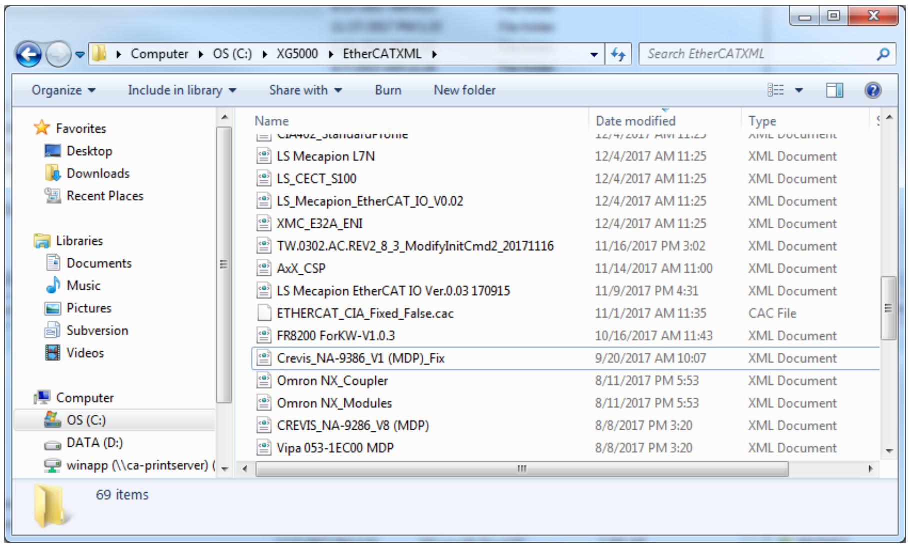
(a) Searching function of EtherCAT slave information (ESI) file (provided by XG5000 4.22 or later version)

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)

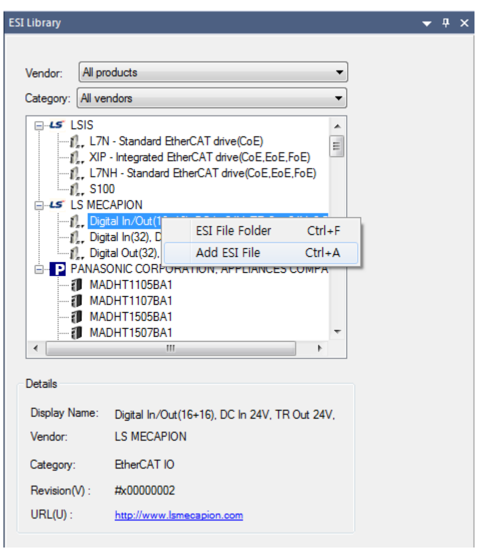
EtherCAT slave information file In addition to bulk addition of ESI files by folder search, it provides individual ESI file addition function. ESI files can be added to the ESI library window without having to restart the XG5000.

Perform the ESI file adding function in the following order.

1) Execute XG5000.

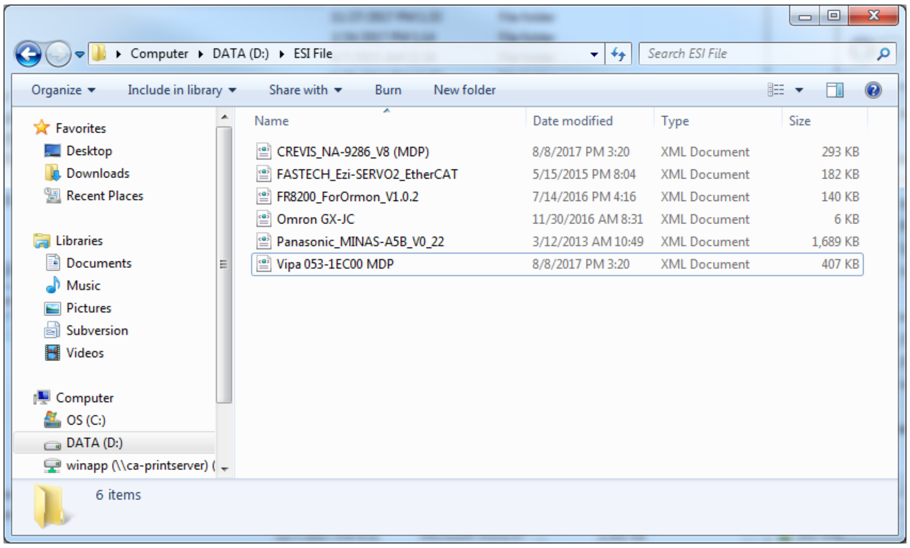
2) Click 'View' – 'ESI library window' menu to activate the ESI library window.

3) Right-click on a mouse on ESI library window, and click on 'ESI file adding' menu.

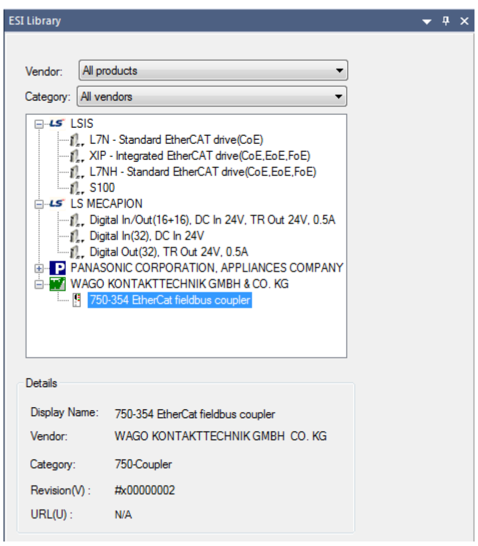


4) Window Explorer of ‘EtherCATXML’ folder is activated as shown below.

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)

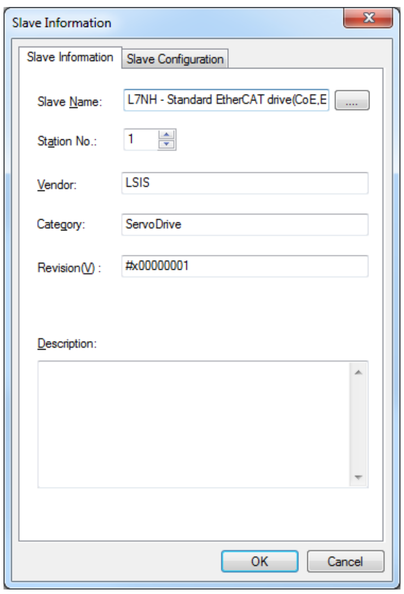
The MDP (Modular Device Profile) is the EtherCAT standard(ETG50001) that defines the configuration data structure of EtherCAT slave. Slave supporting MDP can be set in 'EtherCAT parameter – Slave configuration edit window' of XG5000.
In order to set slave supporting MDP, slave information and module information mounted on slave must be configured

(a) Addition of slave supporting MDP

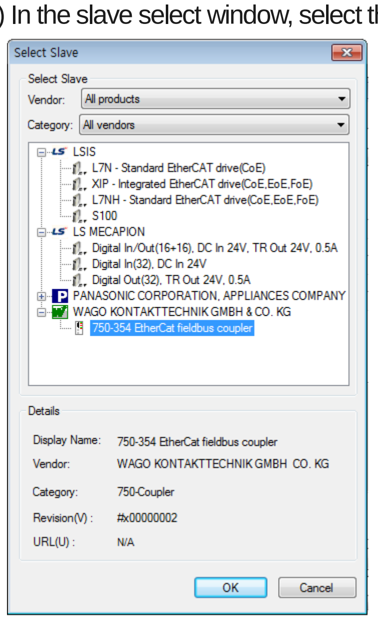
How to add slave supporting MDP and configure module.

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.

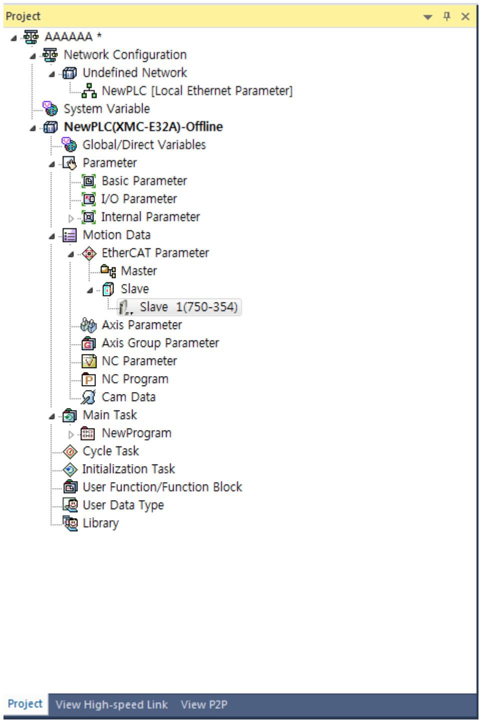


3) In the slave select window, select the slave supporting MDP and click on the OK button.



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

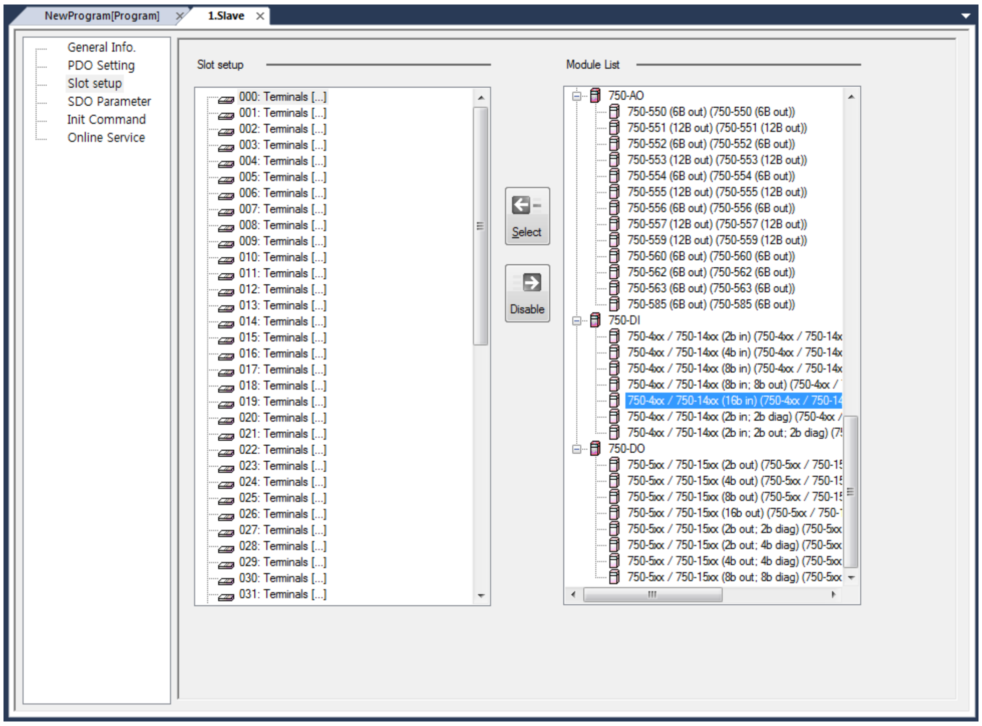
1) Double-click on 'Slave – Slave1(750-354)' on project tree

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.



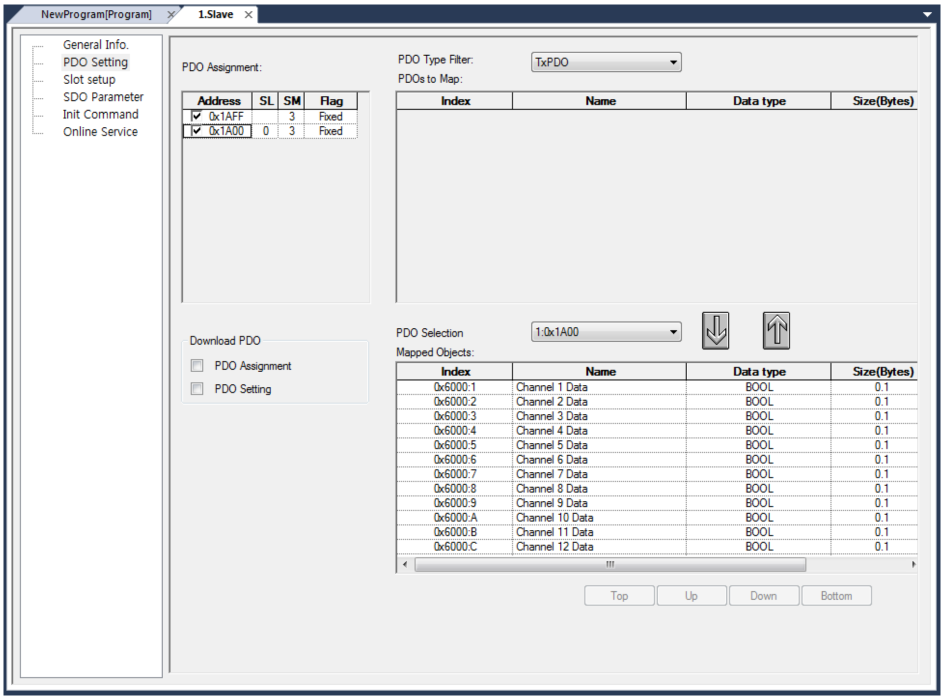
(c) Check of PDO allocation information

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’

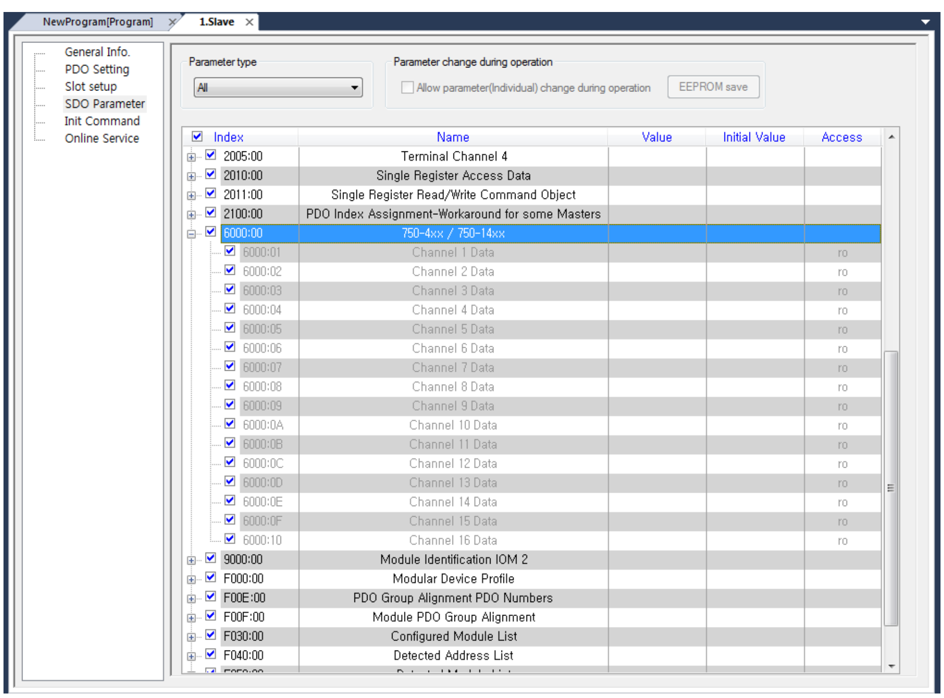


(d) Check of SDO parameter

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.)



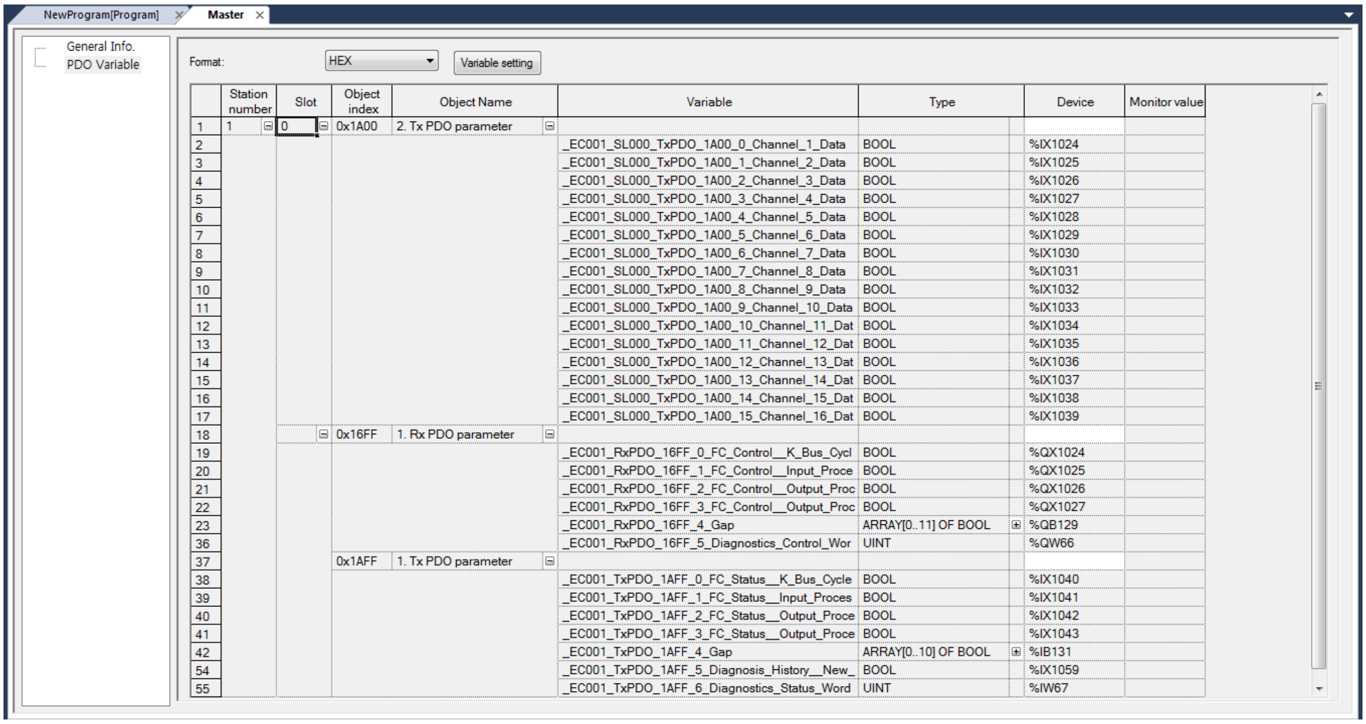
(e) Check of PDO variable information

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)

Warranty

1. Warranty Period
The product you purchased will be guaranteed for 18 months from the date of manufacturing.

2. Scope of Warranty
Any trouble or defect occurring for the above-mentioned period will be partially replaced or repaired. However, please note the following cases will be excluded from the scope of warranty.

(1) Any trouble attributable to unreasonable condition, environment or handling otherwise specified in the manual,
(2) Any trouble attributable to others' products,
(3) If the product is modified or repaired in any other place not designated by the company,
(4) Due to unintended purposes
(5) Owing to the reasons unexpected at the level of the contemporary science and technology when delivered.
(6) Not attributable to the company; for instance, natural disasters or fire

3. Since the above warranty is limited to PLC unit only, make sure to use the product considering the safety for system configuration or applications.

Environmental Policy

LSIS Co., Ltd supports and observes the environmental policy as below.

Environmental Management

LSIS considers the environmental preservation as the preferential management subject and every staff of LSIS use the reasonable endeavors for the pleasurably environmental preservation of the earth.

About Disposal

LSIS' 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.

LS values every single customers.

Quality and service come first at LSIS.

Always at your service, standing for our customers.

www.lsis.com

LSIS

■ HEAD OFFICE
LS Tower, 127, LS-ro, Dongan-gu, Anyang-si, Gyeonggi-Do, 431-848, Korea
Tel : 82-2-2034-4286 Fax : 82-2-2034-4648 E-mail : PLCSales@lsis.com

■ Factory
Samseong 4-gil, Mokcheon-eup, Dongnam-gu, Cheonan-si, Chungcheongnam-do, 31226, Korea

■ LSIS(Shanghai) Co., Ltd. /CHINA
32nd Room 1~4, 32/F, Great Wall Building, No.3000 North Zhongshan Road, Putuo District, Shanghai, P.R. China
Tel : 86-21-5237-9977(609) Fax : 86-21-5237-7189

■ LSIS(Dalian) Co., Ltd. /CHINA
No. 15, Liaohexi 3-Road, Economic and Technical Development zone, Dalian, P.R. China
Tel : 86-411-8731-7542 Fax : 86-411-8730-7560

■ LSIS(Wuxi) Co., Ltd./CHINA
102-A, National High & New Tech Industrial Development Area, Wuxi, Jiangsu, P.R. China
Tel : 86-510-8534-6666 Fax : 86-510-8534-4078

■ LS Hukai Electric(Hubei) Co., Ltd./CHINA
No. 100, Tanjiahe Road, Dianjun District, Yichang City, Hubei Province, P.R. China
Tel : 86-717-667-7536 Fax : 86-717-667-7222

■ LS-VINA Industrial Systems Co., Ltd./VIETNAM
Room 1311, 13th, M3-M4 Building 91 Nguyen Chi Thanh street, Hanoi, Vietnam
Tel : 84-4-6275-8055 Fax : 86-21-5237-7189

■ LSIS(ME) FZE/U.A.E.
LOB 19-205, JAFZA View Tower, Jebel Ali Free Zone, Dubai, United Arab Emirates
Tel : 971-4-886-5360 Fax : 971-4-886-536

■ LSIS Europe B.V./NETHERLANDS
1st. Floor, Tupolevlaan 48, 1119NZ,Schiphol-Rijk, The Netherlands
Tel : 31-20-654-1420 Fax : 31-20-654-1429

■ LSIS Japan Co., Ltd./JAPAN
16th, Higashi-Kan, Akasaka Twin Tower, 2-17-22, Akasaka, Minato-ku, Tokyo, Japan
Tel : 81-3-3582-9128 Fax : 81-3-3582-2667

■ LSIS USA Inc./U.S.A
2000 Millbrook Drive, Lincolnshire, Chicago, IL 60069, United States of America
Tel : 847-941-8240 Fax : 847-941-8259

※ LSIS constantly endeavors to improve its product so that information in this manual is subject to change without notice.

2018. 6