MELSOFT

MITSUBISHI ELECTRIC

Logic Simulator

MELSOFT Mirror Operating Manual

-SW1DND-LCS

SAFETY PRECAUTIONS

(Read these precautions before using this product.)

Before using this product, please read this manual carefully, and pay full attention to safety to handle the product correctly.

If the equipment is used in a manner not specified by the manufacturer, the protection provided by the equipment may be impaired.

The precautions given in this manual are concerned with this product only.

For the safety precautions of the programmable controller system, refer to the user's manual for the module used.

In this manual, the safety precautions are classified into two levels: "⚠WARNING" and "⚠CAUTION".

⚠ **WARNING**	Indicates that incorrect handling may cause hazardous conditions, resulting in death or severe injury.
⚠ **CAUTION**	Indicates that incorrect handling may cause hazardous conditions, resulting in minor or moderate injury or property damage.

Under some circumstances, failure to observe the precautions given under "⚠CAUTION" may lead to serious consequences.

Observe the precautions of both levels because they are important for personal and system safety.

Make sure that the end users read this manual and then keep the manual in a safe place for future reference.

[Security Precautions]

⚠ WARNING

- To maintain the MELSOFT Mirror security (confidentiality, integrity, and availability) against unauthorized access, denial-of-service (DoS) attacks, computer viruses, and other cyberattacks from external devices via the network, take appropriate measures such as firewalls, virtual private networks (VPNs), and antivirus solutions.

INTRODUCTION

Thank you for purchasing the logic simulator MELSOFT Mirror (hereinafter referred to as "MELSOFT Mirror").
This manual is intended to help you understand how to perform simulations using MELSOFT Mirror.
Before using this product, please read this manual carefully and develop familiarity with the functions and performance of MELSOFT Mirror to handle the product correctly.
Note that the menu names and operating procedures may differ depending on an operating system in use and its version. When reading this manual, replace the names and procedures with the applicable ones as necessary.
When applying the simulation results of MELSOFT Mirror to an actual system, ensure the applicability and confirm that it will not cause system control problems.
Please make sure that the end users read this manual.

CONTENTS

RELEVANT MANUALS

Manual name [manual number]	Description
MELSOFT Mirror Operating Manual [SH-082701ENG] (this manual)	System configuration, start-up procedure, and programmable controller system simulation method of MELSOFT Mirror
GX Works3 Operating Manual [SH-081215ENG]	System configuration, parameter settings, and online operations of GX Works3

TERMS

Unless otherwise specified, this manual uses the following terms.

Term	Description
Buffer memory	Memory in an intelligent function module to store data such as setting values and monitored values. For CPU modules, it refers to memory used to store data such as setting values and monitored values of the Ethernet function and data used for data communications of the multiple CPU system function.
Control CPU	A CPU module that controls connected I/O modules and intelligent function modules. In a multiple CPU system, a control CPU can be set for each module.
Dedicated instruction	An instruction that simplifies programming for using functions of intelligent function modules
Device	Memory in a CPU module to store data. Devices such as X, Y, M, D, and others are provided depending on the intended use.
Engineering tool	A tool used for setting up programmable controllers, programming, debugging, and maintenance
FMU	An abbreviation for Functional Mock-up Unit. A FMU is a file that conforms to the FMI standard (an open standard for connecting simulation models). For definitions of FMU and FMI, refer to the following website. https://fmi-standard.org/
Global label	A label that is valid for all the program data when multiple program data are created in the project. There are two types of global label: a module specific label (module label), which is generated automatically by GX Works3, and an optional label, which can be created for any specified device.
Intelligent function module	A module that has functions other than input and output, such as an A/D converter module and D/A converter module
Label	A variable consisting of a specified string used in I/O data or internal processing
Module label	A label that represents one of memory areas (I/O signals and buffer memory areas) specific to each module in a given character string. For the module used, GX Works3 automatically generates this label, which can be used as a global label.
POU	A unit that configures a program. Units are categorized and provided in accordance with functions. Use of POUs enables dividing the lower-layer processing in a hierarchical program into some units in accordance with processing or functions, and creating programs for each unit.
Program block	A group of POUs that configure a program
Simulation server	A personal computer on which MELSOFT Mirror is installed. By connecting a personal computer to the simulation server, the operation of Mitsubishi Electric FA devices such as programmable controllers can be verified on this server.

GENERIC TERMS AND ABBREVIATIONS

Unless otherwise specified, this manual uses the following generic terms and abbreviations.

Generic term and abbreviation	Description
MELSOFT Mirror	An abbreviation for logic simulator MELSOFT Mirror

1 MELSOFT Mirror

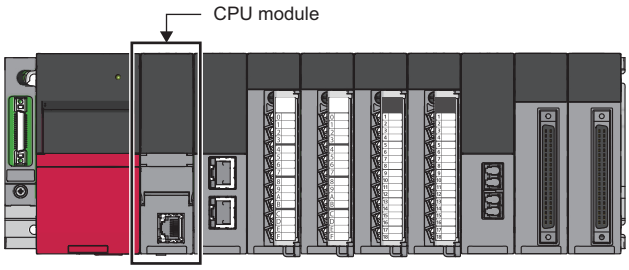
MELSOFT Mirror is simulation software for verifying the operation of Mitsubishi Electric FA devices such as programmable controllers on a personal computer.
MELSOFT Mirror can verify the operation of up to 30 CPU modules. Verifications performed at the production site can only be performed with MELSOFT Mirror.
Verifications with MELSOFT Mirror can detect and correct device errors that could only be detected at the production site.

1.1 Systems That Can Be Simulated

The following systems can be simulated with MELSOFT Mirror.

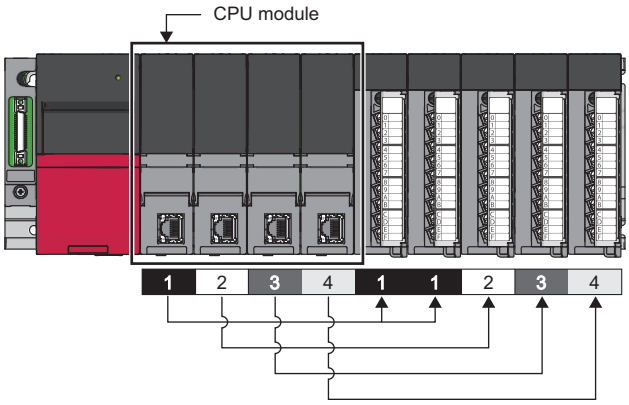
Single CPU system

This system controls I/O modules and intelligent function modules with a single CPU module mounted on a main base unit.



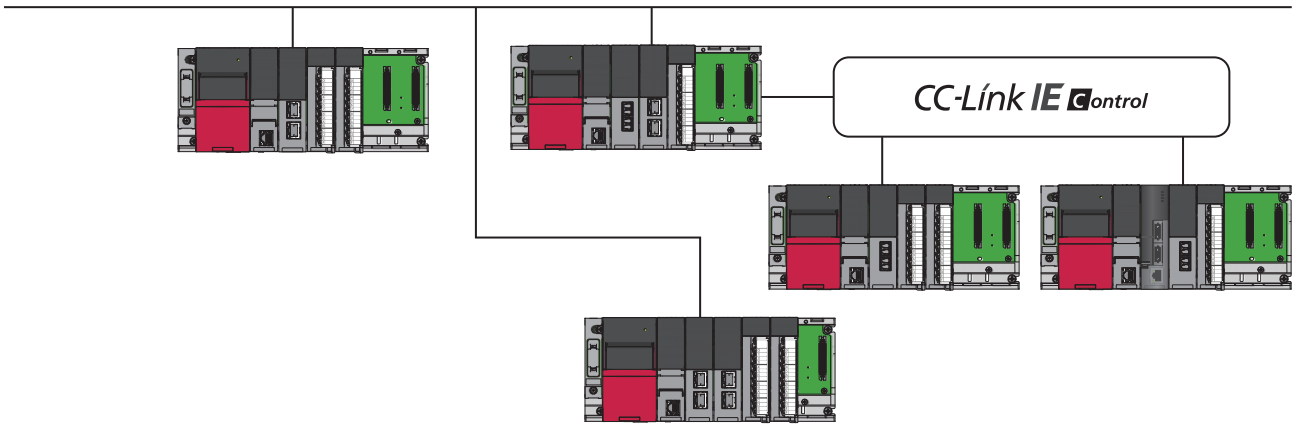
Multiple CPU system

This system requires multiple CPU modules, and each CPU module controls I/O modules and intelligent function modules.



Network-connected system

This system connects single CPU systems or multiple CPU systems over a network such as CC-Link IE Control.

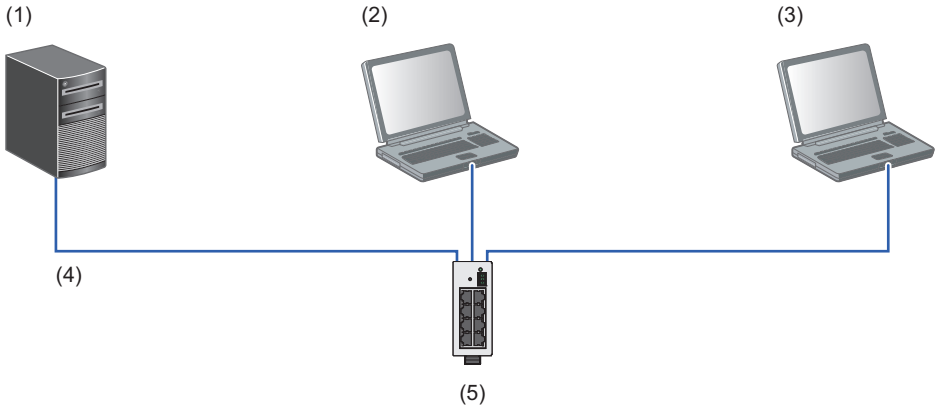


2 SYSTEM CONFIGURATION

This chapter describes the system required to use MELSOFT Mirror and supported modules.

2.1 System Required to Use MELSOFT Mirror

The MELSOFT Mirror system consists of personal computers, Ethernet cables, and a switching HUB. MELSOFT Mirror is installed on the personal computer (1).
Software that can be connected to MELSOFT Mirror is installed on the personal computers (2) and (3). Without using the personal computers (2) and (3), both MELSOFT Mirror and the software to be connected to MELSOFT Mirror can be installed on the personal computer (1).



No.	Name	Description
(1)	Personal computer on which MELSOFT Mirror is installed (simulation server)	Install MELSOFT Mirror. (☞ Page 11 MELSOFT Mirror operating environment) When other personal computers (2) and (3) are connected to MELSOFT Mirror, use Ethernet.
(2)	Personal computer connected to MELSOFT Mirror	Install the software that can be used together with MELSOFT Mirror and connect to MELSOFT Mirror to perform simulations. (☞ Page 12 MELSOFT Mirror display environment, Page 12 Software that can be connected to MELSOFT Mirror)
(3)		
(4)	Ethernet cable	For the network operating environment, refer to the following. (☞ Page 11 MELSOFT Mirror operating environment)
(5)	Switching hub	

Point

If MELSOFT Mirror is operated at the same time in (2) and (3), the operations are shared.

MELSOFT Mirror operating environment

The following table lists the operating environment requirements for the personal computer on which MELSOFT Mirror is installed.

Item	Description
Compatible OS	• Microsoft® Windows® 11 (Pro, Enterprise) • Microsoft® Windows® Server 2025 • Microsoft® Windows® Server 2022
CPU	• Intel Core i7 10th generation or later for the simulation of a system with up to 16 CPU modules • Intel Core i9 10th generation or later for the simulation of a system with 17 or more CPU modules
Required memory	16GB or more
Storage	SSD (128GB or more free space)*1
File system	NTFS
Display resolution	800 × 600 or higher
Network	Ethernet 1Gbps or faster

*1 A free space of 1GB or more is required on the system drive separately.

MELSOFT Mirror display environment

The MELSOFT Mirror windows are displayed by accessing the simulation server with a web browser.
The following table lists the MELSOFT Mirror display environment requirements.

Item	Description
Compatible OS	Microsoft® Windows® 11 (Pro, Enterprise)
Web browser	Microsoft Edge (version 110 or later, IE mode not supported)
CPU	• Intel Core i5 10th generation or later • Intel Core i7 10th generation or later • Intel Core i9 10th generation or later
Required memory	8GB or more
Storage	20GB or more free space
Display resolution	1600 × 1200 or higher
Network	Ethernet 100Mbps or faster

Software that can be connected to MELSOFT Mirror

By connecting software to MELSOFT Mirror, the personal computer on which MELSOFT Mirror is installed can be used as a programmable controller for simulation.

Software that can be connected	Condition
Microsoft Edge	Version 110 or later
GX Works3	Version 1.100E or later

Reading/writing virtual SD memory card files

To read/write virtual SD memory card files from/to Explorer, MELSOFT Mirror Tools needs to be installed additionally.
(☞ Page 27 MELSOFT Mirror Tools Download)

Supported languages

The languages supported for the window display of MELSOFT Mirror are English, Japanese, and Chinese (Simplified).

2.2 Supported Modules

MELSOFT Mirror can simulate MELSEC iQ-R series modules.
The supported modules are listed below.
○: Supported, —: Not supported
Hardware configuration setting: Modules that can be placed in the MELSOFT Mirror system configuration
Simulation target selection: Modules that can be placed and simulated in the MELSOFT Mirror system configuration

Point

To simulate modules, the parameters need to be set for the modules. The parameters of the RD77MS and network modules are written via the control CPU. Enable the module operation setting of the control CPU. Or, disable the module operation setting of the control CPU, which controls these modules but is not supported by MELSOFT Mirror.

MELSEC iQ-R series

Main base unit

Item	Model	Hardware configuration setting	Simulation target selection
Main base unit	R33B, R35B, R38B, R312B	○	—
Extended temperature range main base unit	R310B-HT	○	—
Redundant power supply main base unit	R310RB	○	—
Extended temperature range redundant power supply main base unit	R38RB-HT	○	—

Extension base unit

Item	Model	Hardware configuration setting	Simulation target selection
Extension base unit	R65B, R68B, R612B	○	—
Extended temperature range extension base unit	R610B-HT	○	—
Redundant power supply extension base unit	R610RB	○	—
Extended temperature range redundant power supply extension base unit	R68RB-HT	○	—
Redundant extension base unit	R68WRB	○	—
Extended temperature range redundant extension base unit	R66WRB-HT	○	—
RQ extension base unit	RQ65B, RQ68B, RQ612B	○	—

Power supply module

Item	Model	Hardware configuration setting	Simulation target selection
Power supply module	R61P, R63P, R64P	○	—
	R62P	○	—
Redundant power supply module	R63RP, R64RP	○	—

CPU module

Item	Model	Hardware configuration setting	Simulation target selection
Programmable controller CPU	R04CPU, R08CPU, R16CPU, R32CPU, R120CPU	○	○
	R00CPU, R01CPU, R02CPU, R04ENCPU, R08ENCPU, R16ENCPU, R32ENCPU, R120ENCPU	○	—
Process CPU	R08PCPU, R16PCPU, R32PCPU, R120PCPU	○	○
C Controller module	R12CCPU-V	○	—
SIL2 Process CPU	R08PSFCPU, R16PSFCPU, R32PSFCPU, R120PSFCPU	○	—
Safety CPU	R08SFCPU, R16SFCPU, R32SFCPU, R120SFCPU	○	—

CPU module extension

Item	Model	Hardware configuration setting	Simulation target selection
Redundant function module	R6RFM	○	—
SIL2 function module	R6PSFM	○	—
Safety function module	R6SFM	○	—

Motion CPU

Item	Model	Hardware configuration setting	Simulation target selection
Motion CPU	R16MTCPU	○	—
	R32MTCPU	○	—
	R64MTCPU	○	—

NCCPU

Item	Model	Hardware configuration setting	Simulation target selection
NCCPU	R16NCCPU	○	—

Robot CPU

Item	Model	Hardware configuration setting	Simulation target selection
Robot CPU	R16RTCPU	○	—

Remote head module

Item	Model	Hardware configuration setting	Simulation target selection
CC-Link IE Field Network	RJ72GF15-T2	○	—

I/O module

Item	Model	Hardware configuration setting	Simulation target selection
AC input module	RX10	○	○
	RX10-TS	○	○
	RX28	○	○
DC input module	RX40C7, RX41C4, RX42C4	○	○
	RX40C7-TS, RX41C4-TS, RX70C4, RX71C4, RX72C4	○	○
DC high-speed input module	RX40PC6H, RX40NC6H	○	○
	RX41C6HS, RX61C6HS	○	○
Input module with diagnostic functions	RX40NC6B	○	○
Contact output module	RY10R2	○	○
	RY10R2-TS	○	○
	RY18R2A	○	○
Triac output module	RY20S6	○	○
Transistor output module	RY40NT5P, RY41NT2P, RY42NT2P, RY40PT5P, RY41PT1P, RY42PT1P	○	○
	RY40NT5P-TS, RY41NT2P-TS, RY40PT5P-TS, RY41PT1P-TS	○	○
Transistor high-speed output module	RY41NT2H, RY41PT2H	○	○
Output module with diagnostic functions	RY40PT5B	○	—
SIL2 analog control output module	RY40PT5B-AS	○	—
I/O combined module	RH42C4NT2P	○	—
Dual signal module	R173SXY	○	—

Intelligent function module

Item	Model	Hardware configuration setting	Simulation target selection
Analog-digital converter module	R60AD4, R60ADI8, R60ADV8	○	—
Channel isolated analog-digital converter module	R60AD8-G	○	—
	R60AD16-G	○	—
Channel isolated analog-digital converter module (with signal conditioning function)	R60AD6-DG	○	—
High speed analog-digital converter module	R60ADH4	○	—
HART-enabled analog-digital converter module	R60ADI8-HA	○	—
Digital-analog converter module	R60DA4, R60DAI8, R60DAV8	○	—
Channel isolated digital-analog converter module	R60DA8-G	○	—
	R60DA16-G	○	—
High speed digital-analog converter module	R60DAH4	○	—
Simple Motion module	RD77MS2, RD77MS4, RD77MS8, RD77MS16	○	○
	RD77GF4, RD77GF8, RD77GF16, RD77GF32	○	—
Channel isolated RTD input module	R60RD8-G	○	—
Channel isolated thermocouple input module	R60TD8-G	○	—
Temperature control module	R60TCTRT2TT2, R60TCTRT2TT2BW, R60TCRT4, R60TCRT4BW	○	—
	R60TCTRT2TT2-TS, R60TCRT4-TS	○	—
High-speed counter module	RD62P2, RD62D2, RD62P2E	○	—
Channel isolated pulse input module	RD60P8-G	○	—
Flexible high-speed I/O control module	RD40PD01	○	—
MES interface module	RD81MES96	○	—
	RD81MES96N	○	—

Item	Model	Hardware configuration setting	Simulation target selection
High speed data logger module	RD81DL96	○	—
OPC UA server module	RD81OPC96	○	—
Recorder module	RD81RC96	○	—
Camera recorder module	RD81RC96-CA	○	—
High speed data communication module	RD81DC96	○	—
C intelligent function module	RD55UP06-V	○	—
Positioning module	RD75P2, RD75P4, RD75D2, RD75D4	○	—
Ethernet	RJ71EN71	○	○ (except RJ71EN71(E+E))
CC-Link IE TSN	RJ71GN11-T2	○	—
Motion module	RD78G4, RD78G8, RD78G16, RD78G32, RD78G64, RD78GHV, RD78GHW	○	—
CC-Link IE Controller Network	RJ71GP21-SX	○	○
	RJ71GP21S-SX	○	○
	RJ71EN71	○	○
CC-Link IE Field Network	RJ71GF11-T2	○	○
	RJ71EN71	○	○
CC-Link	RJ61BT11	○	—
MELSECNET/H network module	RJ71LP21-25	○	○
Serial communication	RJ71C24, RJ71C24-R2, RJ71C24-R4	○	—
GP-IB interface module	RJ71GB91	○	—
MODBUS	RJ71C24, RJ71C24-R2, RJ71C24-R4	○	—
AnyWireASLINK master module	RJ51AW12AL	○	—
BACnet module	RJ71BAC96	○	—
CANopen module	RJ71CN91	○	—
DeviceNet master/slave module	RJ71DN91	○	—
PROFIBUS-DP module	RJ71PB91V	○	—
FL-net(OPCN-2) interface module	ER-1FL2-T	○	—
Laser displacement sensor control module	R60MH112, R60MH112NA	○	—
EtherNet/IP	RJ71EIP91	○	—

Energy measuring

Item	Model	Hardware configuration setting	Simulation target selection
Energy measuring module	RE81WH	○	—

Blank cover

Item	Model	Hardware configuration setting	Simulation target selection
Blank cover module	RG60	○	—

MELSEC-Q series

In MELSOFT Mirror, the MELSEC-Q series modules mounted on an RQ extension base unit and a Q extension base unit can be set in a hardware configuration, but cannot be selected as simulation targets.

Item		Model	Hardware configuration setting	Simulation target selection
Base unit	Extension base unit	Q52B, Q55B, Q63B, Q65B, Q68B, Q612B	○	—
	Q series large type extension base unit	Q55BL, Q65BL, Q68BL	○	—
	Q series large type extension base unit (AnS series size)	Q55BLS, Q65BLS, Q68BLS, Q55BLS-D, Q65BLS-D, Q68BLS-D	○	—
Power supply module		Q61P, Q61P-A1, Q61P-A2, Q62P, Q63P, Q64P, Q64PN, Q61P-D	○	—
I/O module	AC input module	QX10, QX10-TS, QX28	○	—
	DC input module	QX40, QX40-TS, QX40-S1, QX41, QX41-S1, QX41-S2, QX42, QX42-S1, QX70, QX71, QX72, QX80, QX80-TS, QX81, QX81-S2, QX82, QX82-S1	○	—
	DC high-speed input module	QX40H, QX70H, QX80H, QX90H	○	—
	AC/DC input module	QX50	○	—
	Contact output module	QY10, QY10-TS, QY18A	○	—
	Triac output module	QY22	○	—
	Transistor output module	QY40P, QY40P-TS, QY41P, QY42P, QY50, QY68A, QY70, QY71, QY80, QY80-TS, QY81P, QY82P	○	—
	Transistor high-speed output module	QY41H	○	—
	I/O combined module	QH42P, QX48Y57, QX41Y41P	○	—
	Interrupt module	QI60	○	—
	Large type AC input module	QX11L, QX21L	○	—
	Large type contact output module	QY11AL, QY13L	○	—
	Large type triac output module	QY23L	○	—
	Large type transistor output module	QY51PL	○	—

Item		Model	Hardware configuration setting	Simulation target selection
Intelligent function module	Analog-digital converter module	Q64AD, Q68ADV, Q68ADI	○	—
	Channel isolated high resolution analog-digital converter module	Q64AD-GH	○	—
	Channel isolated high resolution analog-digital converter module (with signal conditioning function)	Q62AD-DGH	○	—
	Channel isolated analog-digital converter module	Q68AD-G	○	—
	Channel isolated analog-digital converter module (with signal conditioning function)	Q66AD-DG	○	—
	High speed analog-digital converter module	Q64ADH	○	—
	Digital-analog converter module	Q62DA, Q62DAN, Q64DA, Q64DAN, Q68DAV, Q68DAVN, Q68DAI, Q68DAIN	○	—
	Channel isolated digital-analog converter module	Q62DA-FG, Q66DA-G	○	—
	High speed digital-analog converter module	Q64DAH	○	—
	Analog input/output module	Q64AD2DA	○	—
	Load cell input module	Q61LD	○	—
	Current transformer input module	Q68CT	○	—
	RTD input module	Q64RD	○	—
	Channel isolated RTD input module	Q64RD-G, Q68RD3-G	○	—
	Thermocouple input module	Q64TD	○	—
	Channel isolated thermocouple/micro voltage input module	Q64TDV-GH	○	—
	Channel isolated thermocouple input module	Q68TD-G-H01, Q68TD-G-H02	○	—
	Temperature control module	Q64TCTTN, Q64TCRTN, Q64TCTTBWN, Q64TCRTBWN	○	—
	Loop control module	Q62HLC	○	—
	Multichannel high-speed counter module	QD63P6	○	—
	4Mpps capable high-speed counter module	QD64D2	○	—
	Channel isolated pulse input module	QD60P8-G	○	—
	Multi function counter/timer module	QD65PD2	○	—
	Positioning module	QD70P4, QD70P8, QD70D4, QD70D8, QD73A1	○	—
	Positioning module with built-in counter function	QD72P3C3	○	—
	MES interface module	QJ71MES96, QJ71MES96N	○	—
	Web server module	QJ71WS96	○	—
	CC-Link/LT master module	QJ61CL12	○	—
	MELSECNET/H network module	QJ71LP21-25, QJ71LP21S-25, QJ71LP21G, QJ71BR11, QJ71NT11B	○	—
	MODBUS/TCP interface module	QJ71MT91	○	—
	MODBUS interface module	QJ71MB91	○	—
	FL-net(OPCN-2) interface module	QJ71FL71, QJ71FL71-T, QJ71FL71-B2, QJ71FL71-B5, QJ71FL71-F01, QJ71FL71-T-F01, QJ71FL71-B2-F01, QJ71FL71-B5-F01	○	—
	AS-i master module	QJ71AS92	○	—
	Intelligent communication module	QD51, QD51-R24	○	—
	DeviceNet master-slave module	QJ71DN91	○	—
	AnyWireASLINK master module	QJ51AW12AL	○	—
	Energy measuring module	QE81WH, QE84WH, QE81WH4W, QE83WH4W	○	—
	Insulation monitoring module	QE82LG	○	—
Blank cover module	Blank cover	QG60	○	—

3 MELSOFT Mirror INSTALLATION

This chapter describes how to install MELSOFT Mirror.

3.1 Installation Procedure

The following selections are required during installation.

Restriction

- Do not specify a removable disk as the installation destination folder. If a removable disk is specified as the installation destination, the operation of "MirrorSetup.exe" and MELSOFT Mirror cannot be guaranteed.
- MELSOFT Mirror does not run if software that uses TCP and UDP port numbers 5000 to 5010 is installed on the same personal computer. An example of such software is LabVIEW. MELSOFT Mirror does not run on a personal computer with LabVIEW installed.

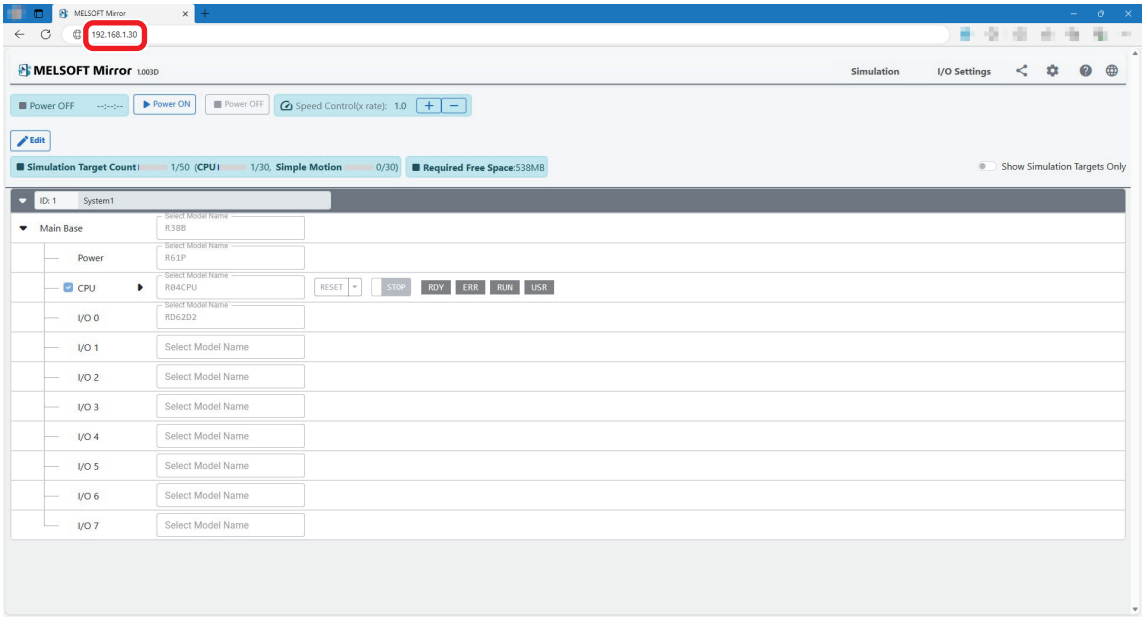
1. Double-click "MirrorSetup.exe" in the downloaded installer.
2. Follow the instructions on the window.
3. Specify the installation destination folder and click the [Next] button.
4. On the confirmation window, click the [Next] button. (The installation will start.)
5. The installation completion window is displayed.

3.2 Starting MELSOFT Mirror

1. From the Windows Start menu, open the MELSOFT folder and start MELSOFT Mirror Management Tool. (☞ Page 39 MELSOFT Mirror Management Tool)
2. Configure the IP address and port number and click the [Apply] button.
3. Click the [Start] button to start MELSOFT Mirror.

3.3 Connection to MELSOFT Mirror

1. Start MELSOFT Mirror on the server computer. (☞ Page 20 Starting MELSOFT Mirror)
2. Open a web browser on the personal computer (client) to be connected to MELSOFT Mirror.
3. Enter the IP address of the personal computer (simulation server) on which MELSOFT Mirror is installed in the address bar of the web browser. (The following is an example of accessing the IP address 192.168.1.30.)



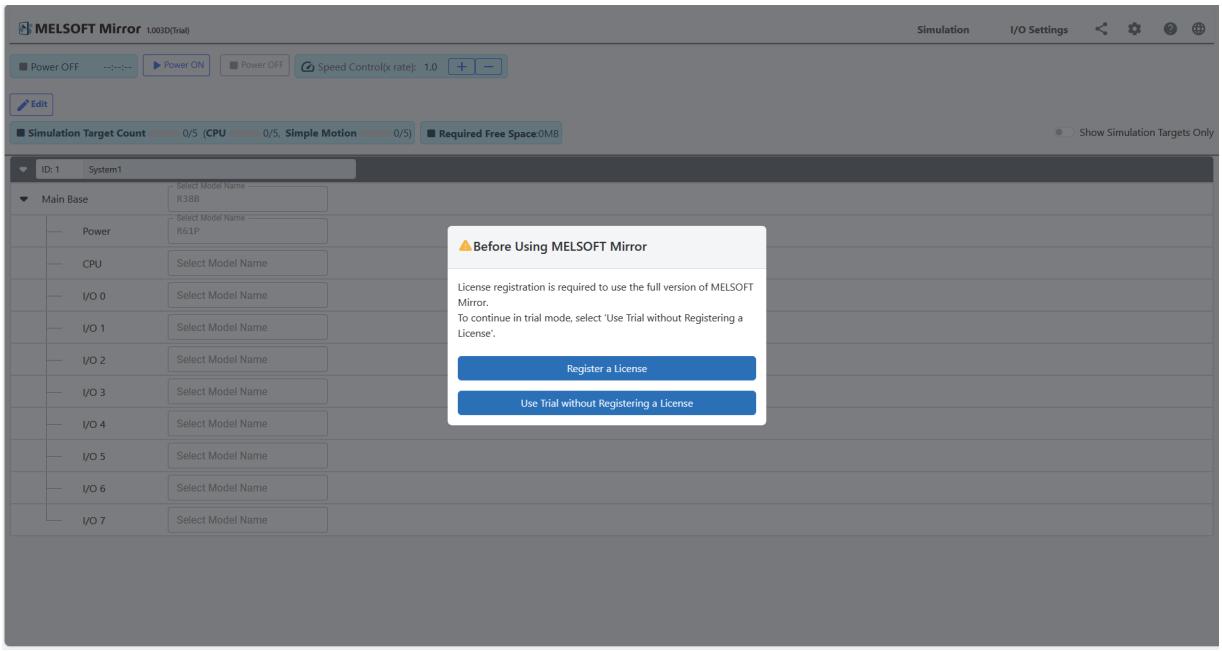
Point

If the port number is set to a number other than 80, add a colon (:) after the IP address, and enter the port number.
(The following is an example where the port number is set to 8080. (192.168.1.30:8080))

4. The top window of MELSOFT Mirror is displayed in the web browser. (☞ Page 41 Simulation)

3.4 MELSOFT Mirror License Registration

When MELSOFT Mirror is connected for the first time, a confirmation window for license registration is displayed.



When registering a license

Follow the procedure below to register a license.

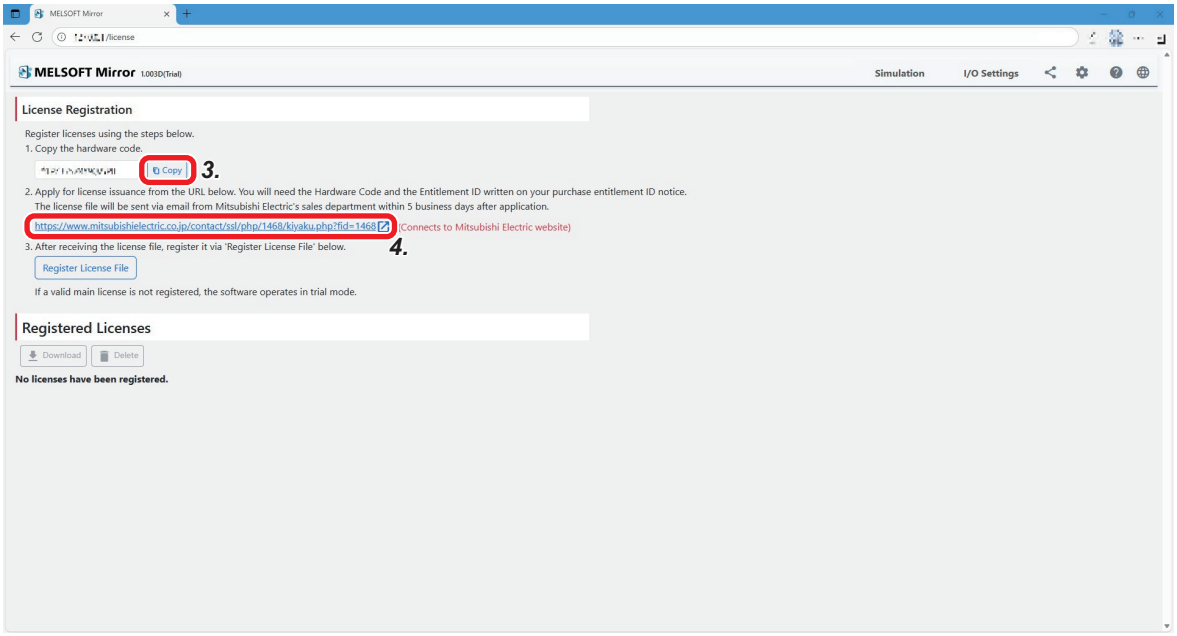
License registration requires two steps: applying for a license file and registering the received license file.

For details, refer to the following.

📖MELSOFT Mirror License Application Instructions [BCN-P5999-1719]

Applying for a license file

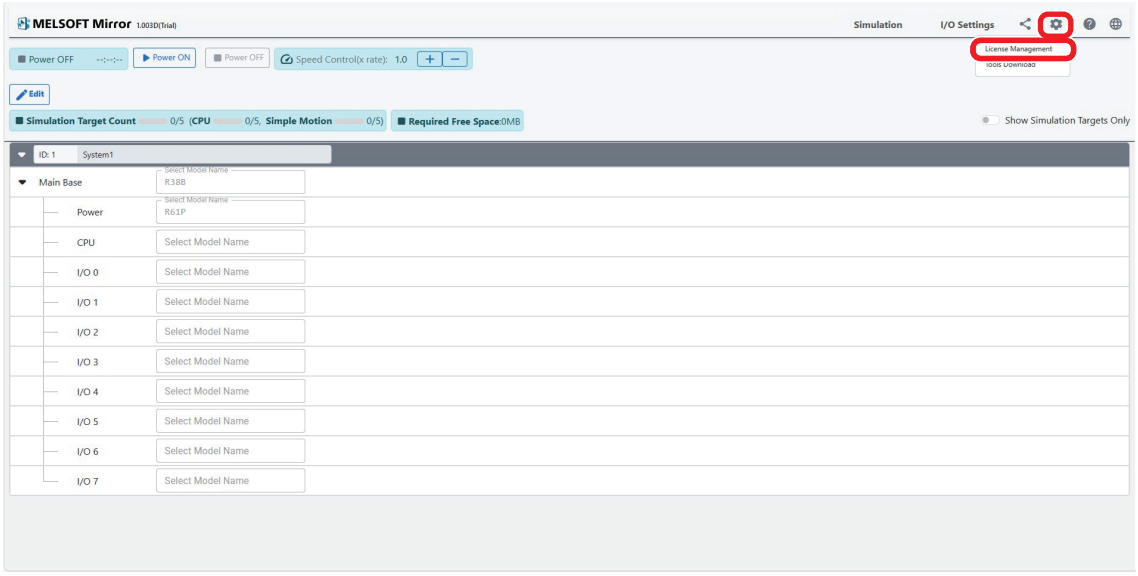
1. Click [Register a license.].
2. The License Management window opens.



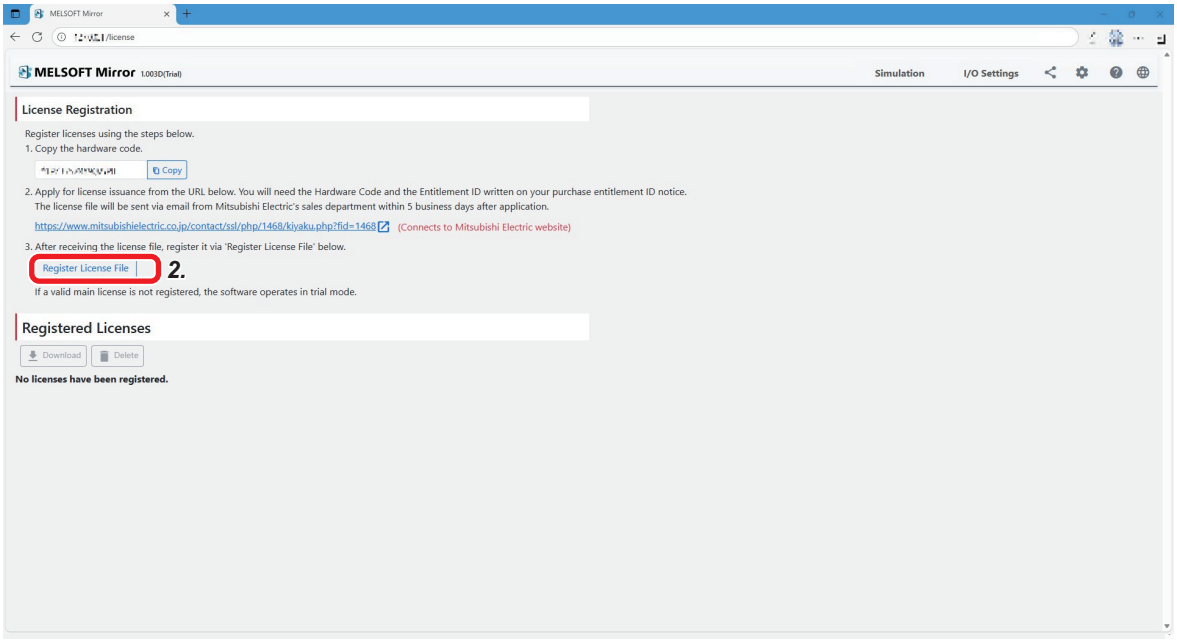
3. Click the [Copy] button of the hardware code. The hardware code is copied to the clipboard.
4. Click the URL of the license application page. The license application page opens in a new tab.
5. On the license application page, enter the hardware code and other required information.
6. A license file is delivered to the registered e-mail address within five business days after the application is accepted.

Registering the received license file

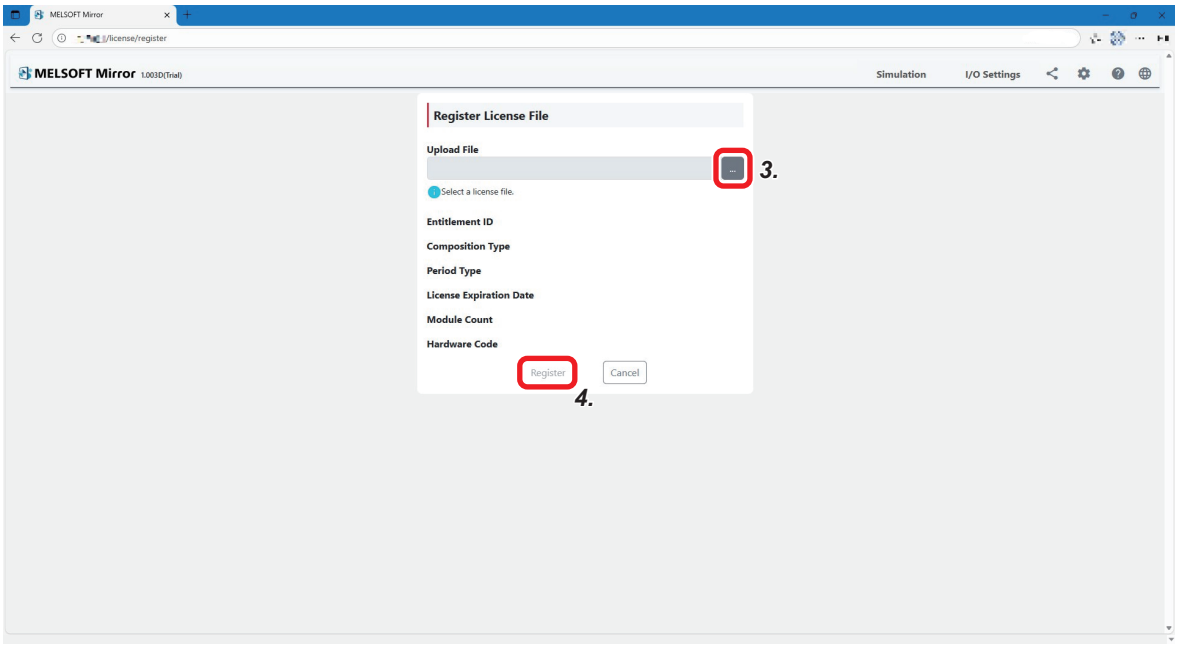
1. Open the License Management window.



2. Click the [Register License File] button. The license file upload window is displayed.



3. Select the license file received.



4. Check the content of the license. If there are no problems, click the [Register] button.

5. Check that the license is registered in [Registered Licenses].

When not registering a license

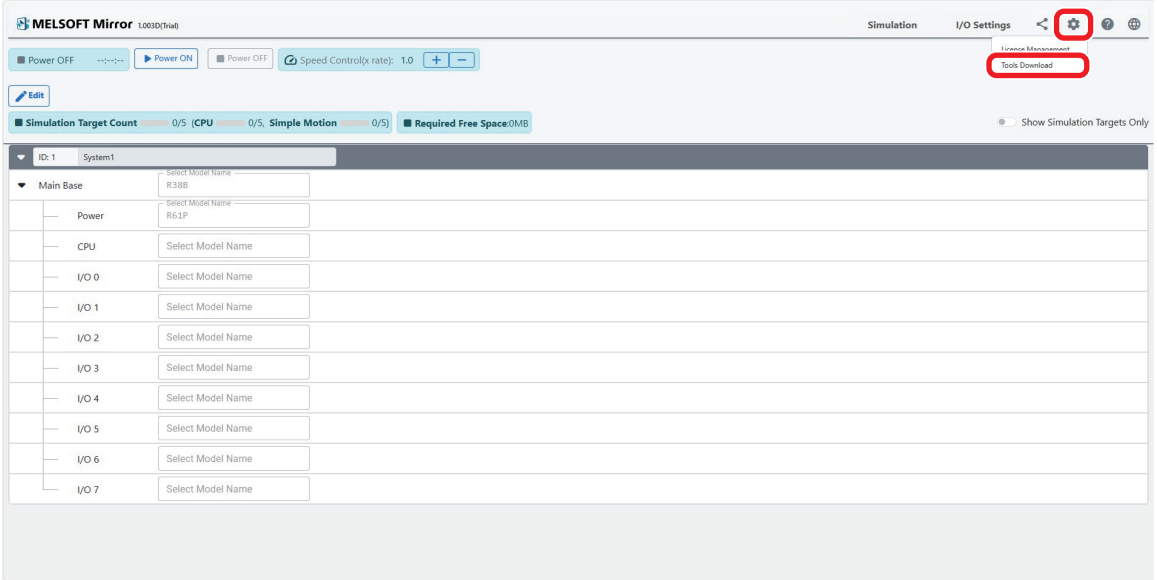
Click [Use Trial without Registering a License]. MELSOFT Mirror starts as a trial version.

The trial version has the following restrictions.

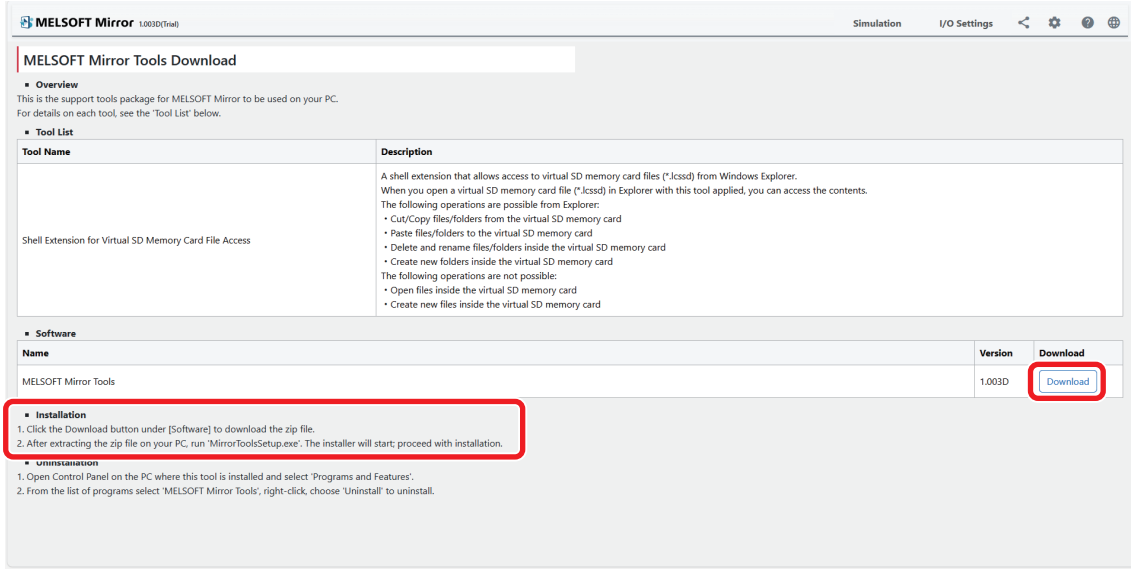
Restriction target	Description
Number of modules to be simulated	5 or less
Duration of one simulation	Within 1 hour

3.5 MELSOFT Mirror Tools Download

From the top window of MELSOFT Mirror, click [Tools Download] to display the MELSOFT Mirror Tools download window.



Click the [Download] button to download MELSOFT Mirror Tools.
The installation procedure is described below the [Download] button.



Restriction

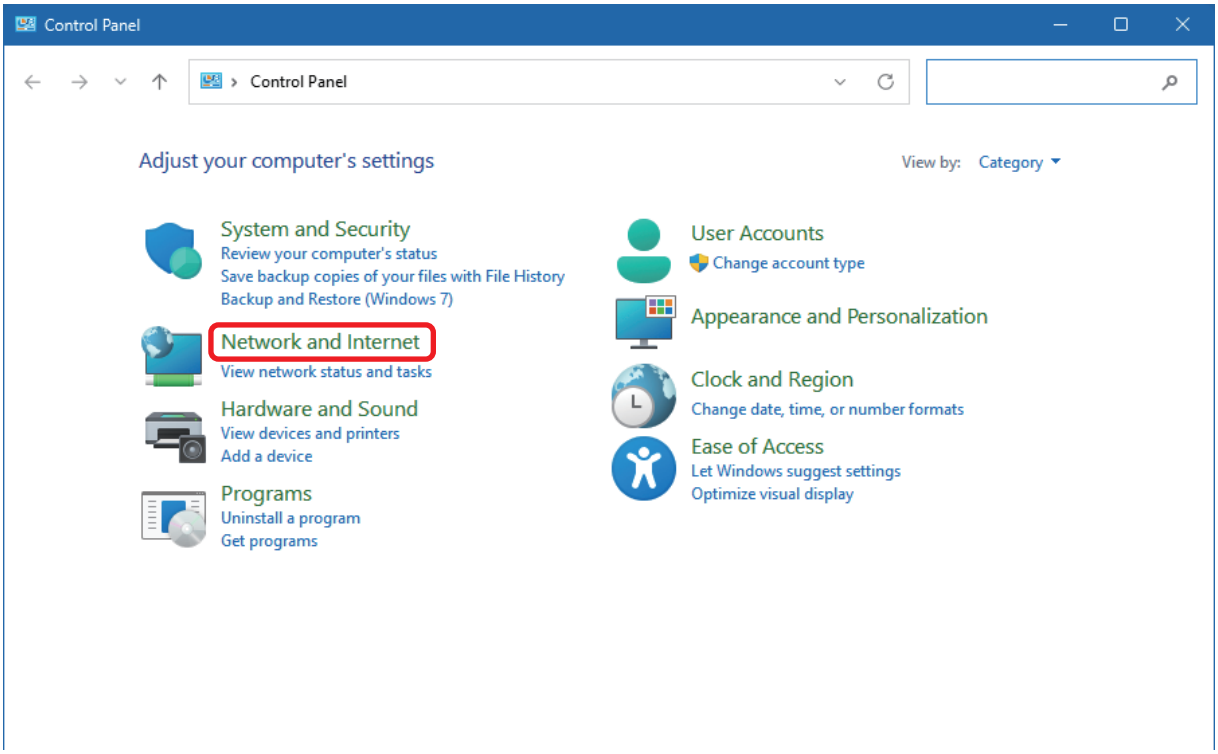
Do not specify a removable disk as the installation destination for MELSOFT Mirror Tools. If it is installed on a removable disk, its operation cannot be guaranteed.

3.6 Setting Multiple IP Addresses on MELSOFT Mirror

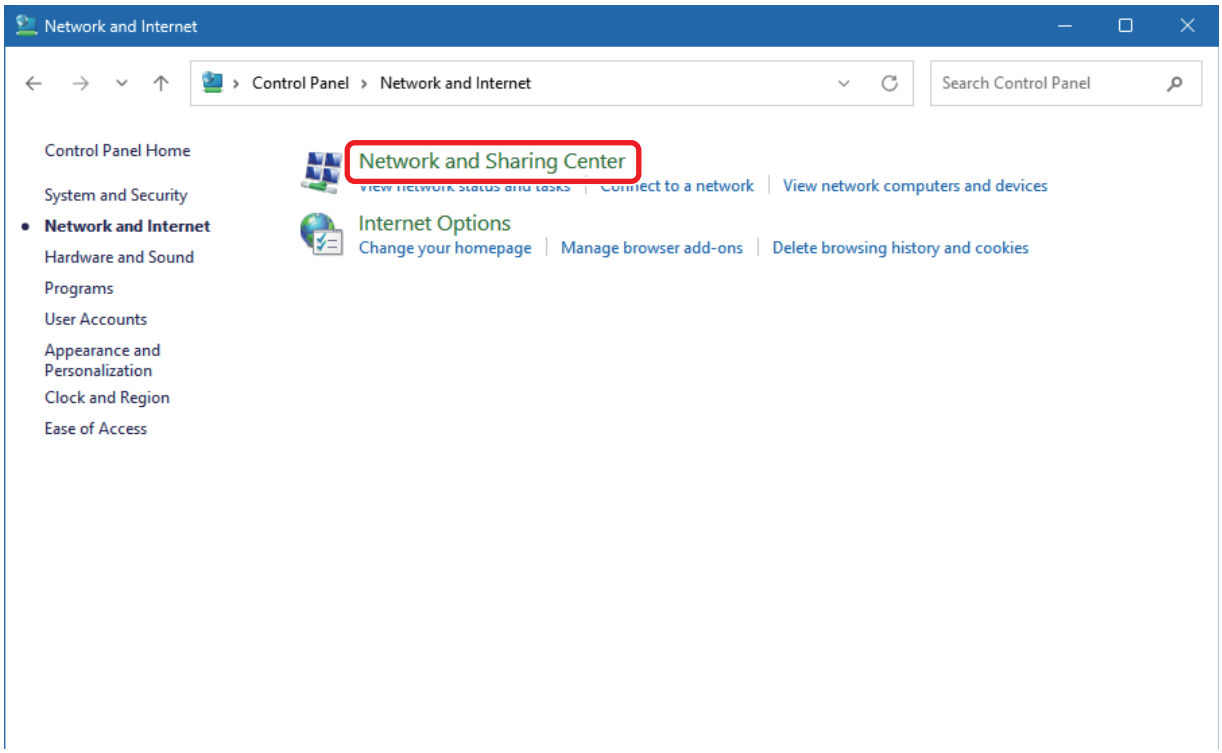
This section describes the procedure for assigning different IP addresses to multiple CPU modules on MELSOFT Mirror.

Settings on a personal computer (Setting multiple IP addresses)

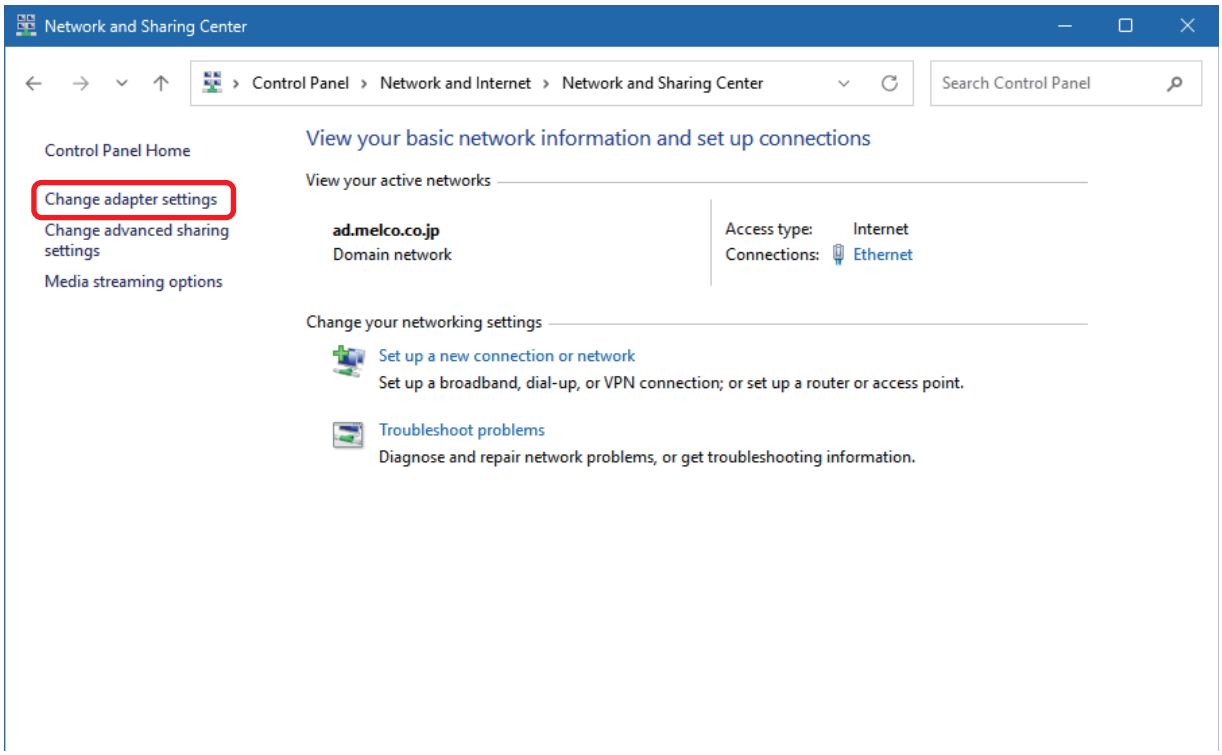
1. Open [Control Panel] from the Windows Start menu. Select [Network and Internet].



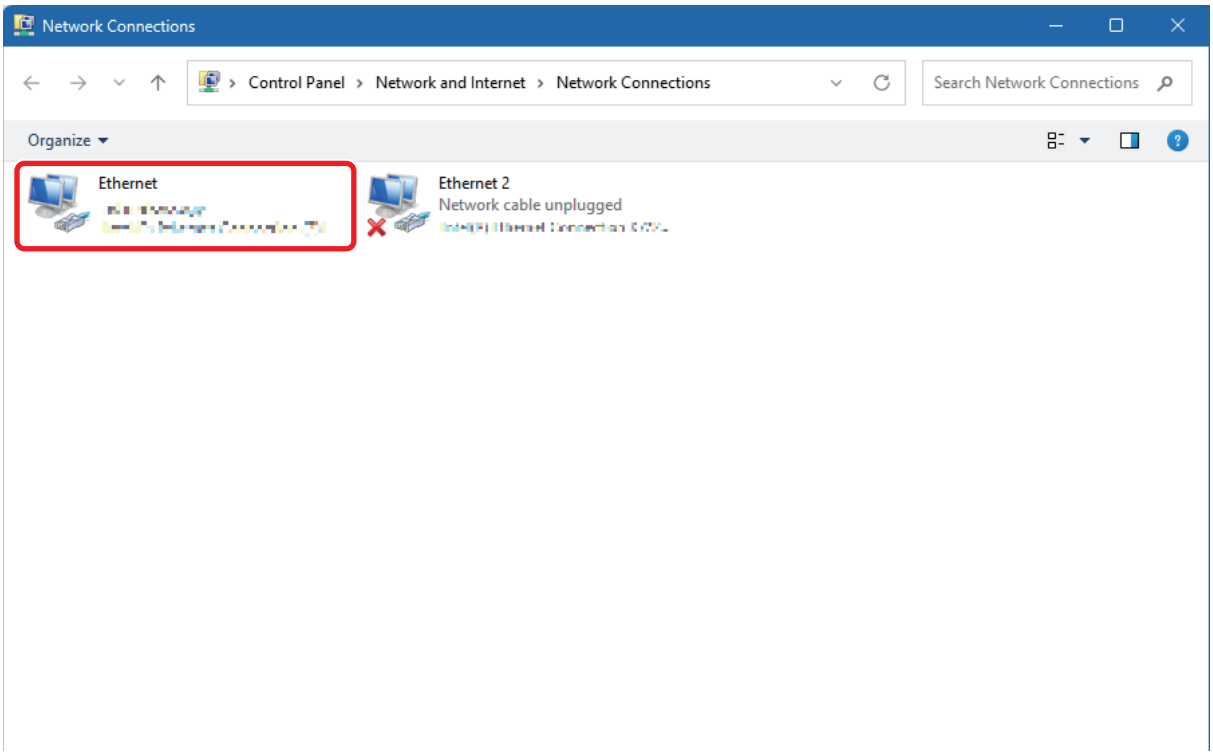
2. Click [Network and Sharing Center].



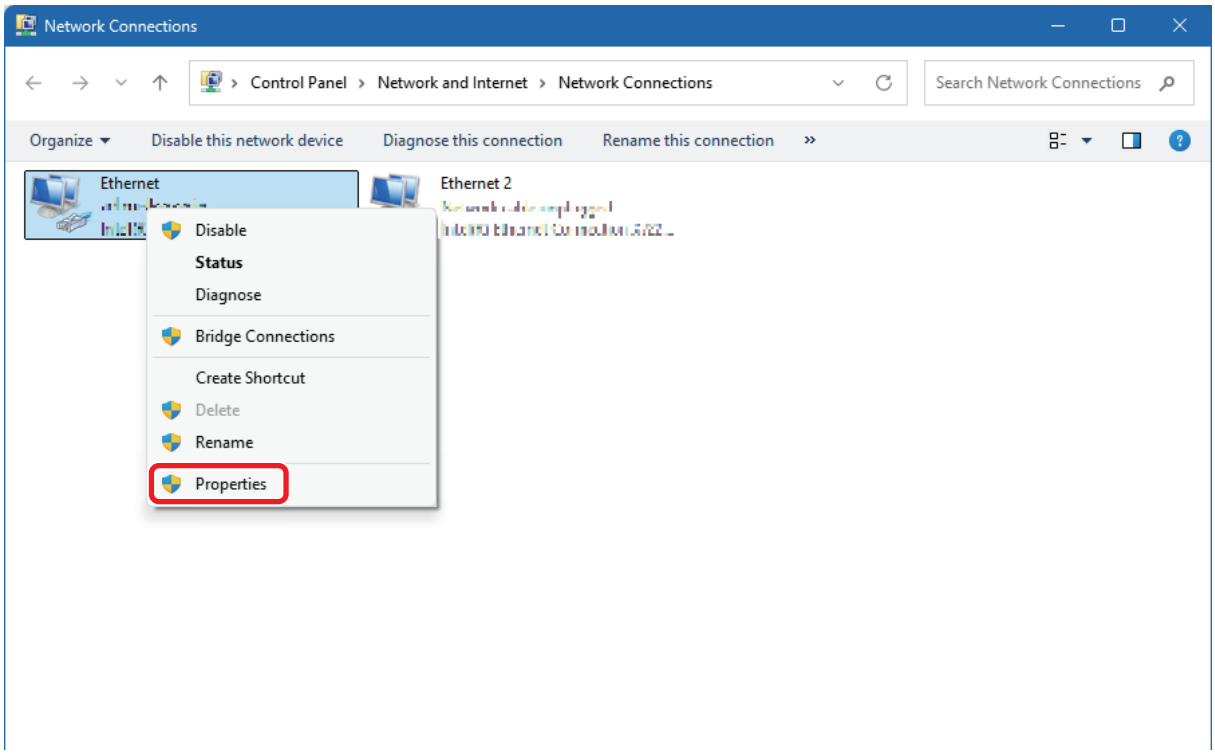
3. Click [Change adapter settings] in the left pane.



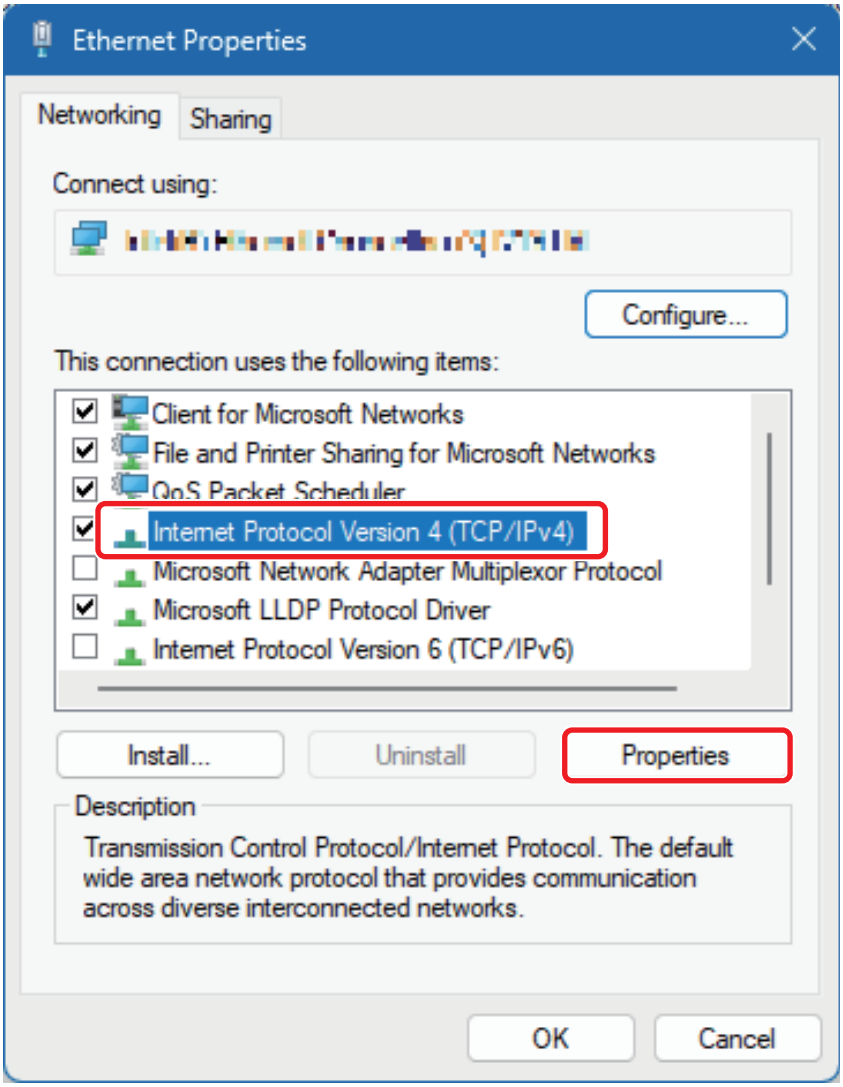
4. In the list of Ethernet adapters connected to the personal computer, right-click on the Ethernet adapter to set an IP address.



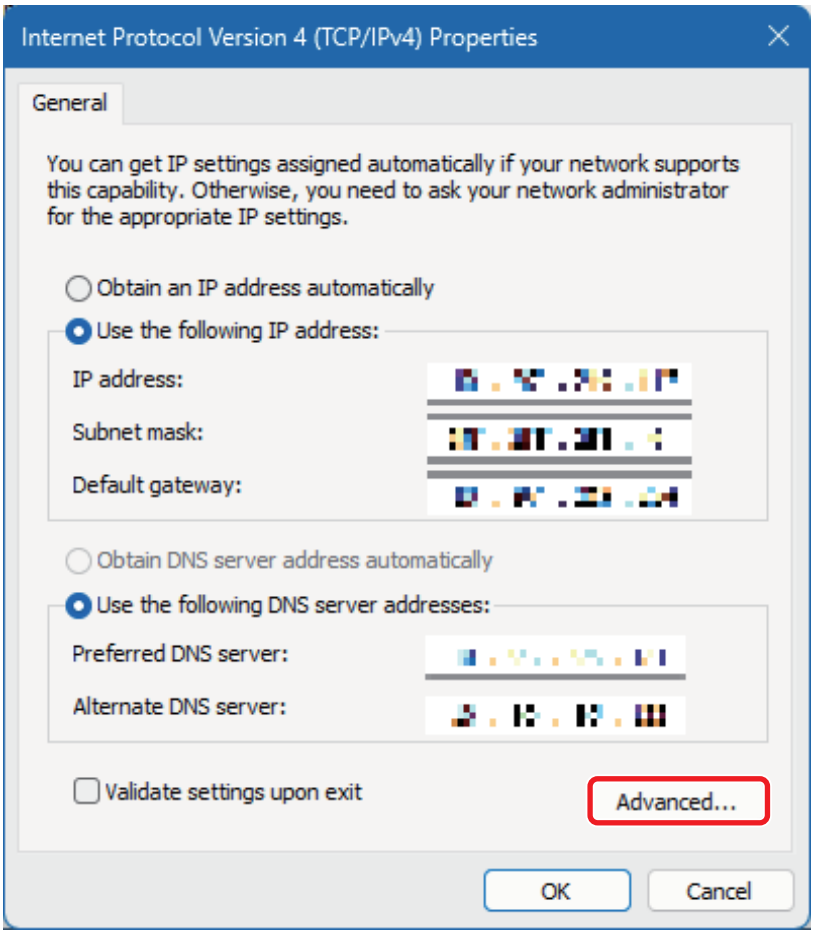
5. Select "Properties" in the drop-down list.



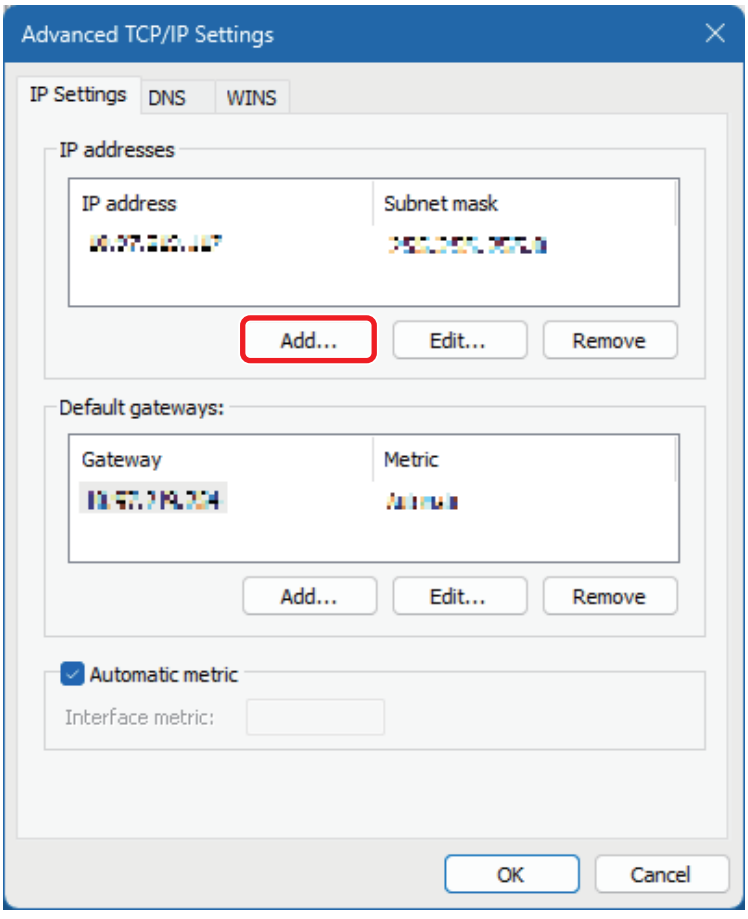
6. Select "Internet Protocol Version 4 (TCP/IPv4)", and click the [Properties] button.



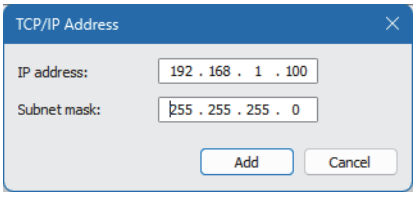
7. Click the [Advanced] button.



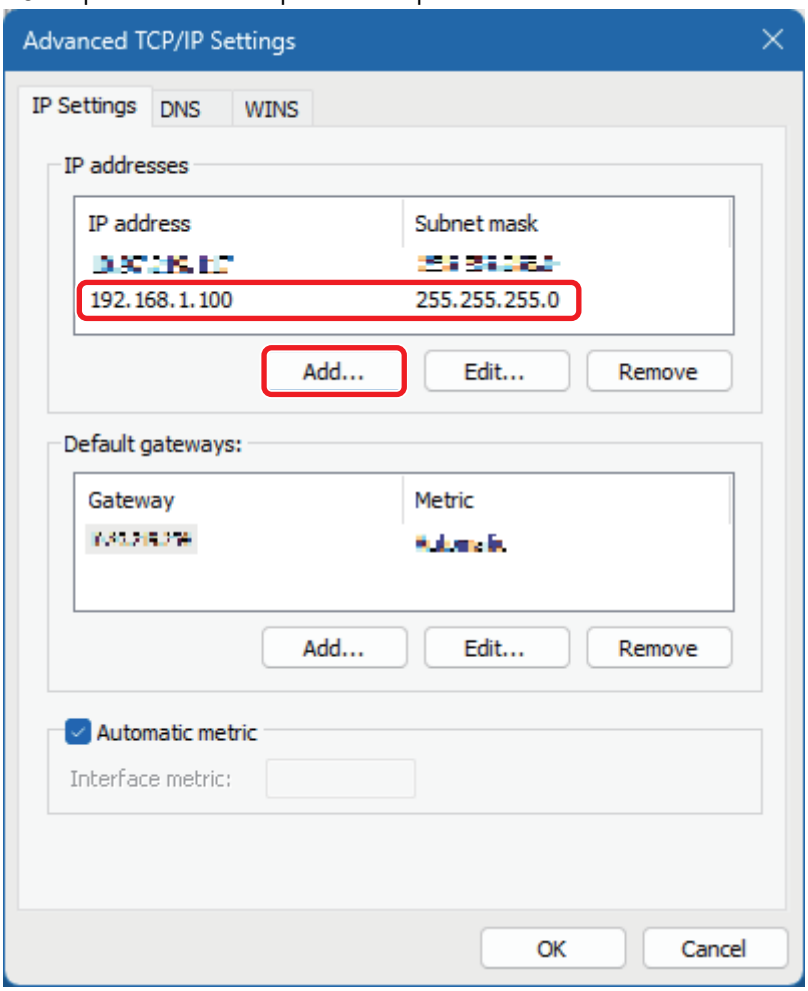
8. Click the [Add] button of "IP addresses".



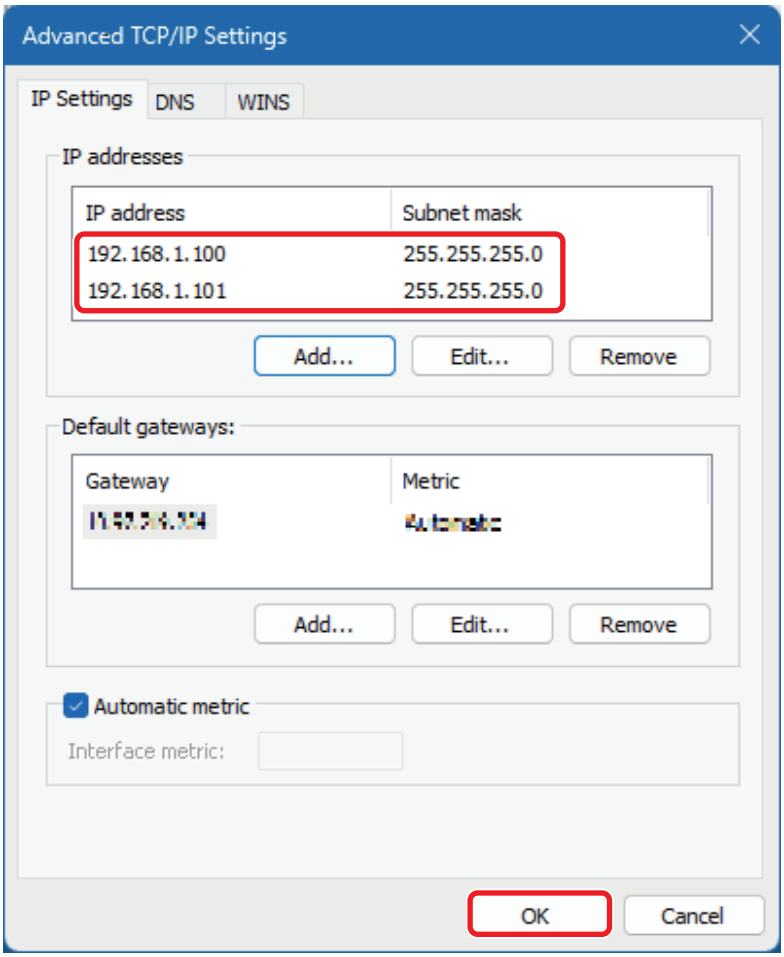
9. Enter the IP address and subnet mask. Example) IP address: 192.168.1.101, subnet mask: 255.255.255.0



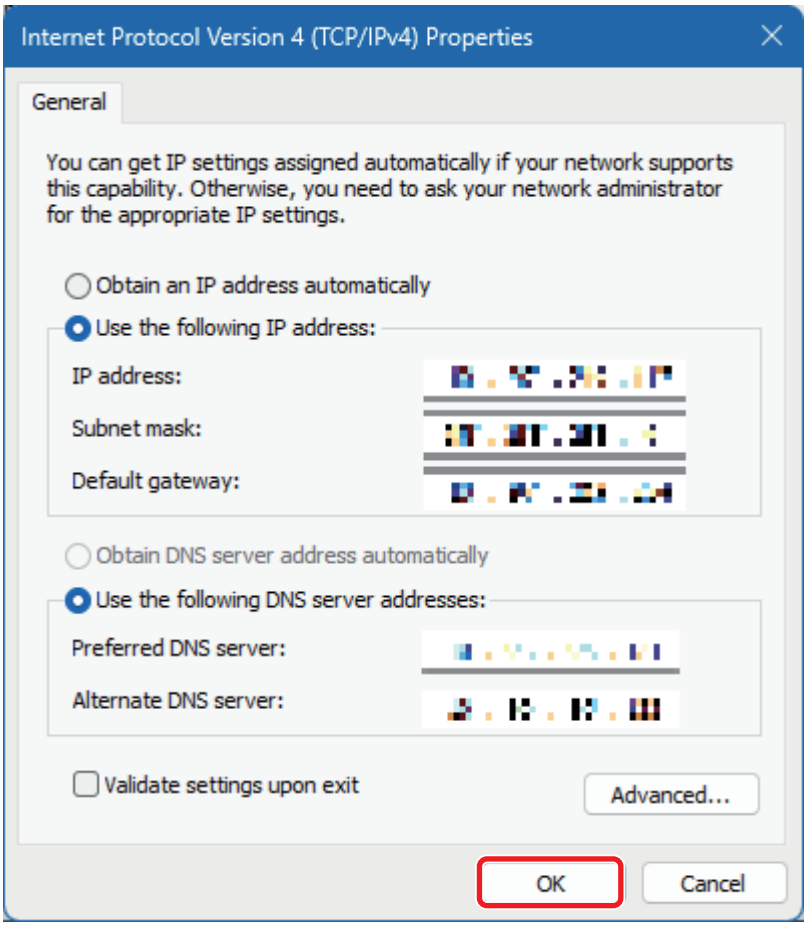
10. Repeat the above steps for the required number of IP addresses.



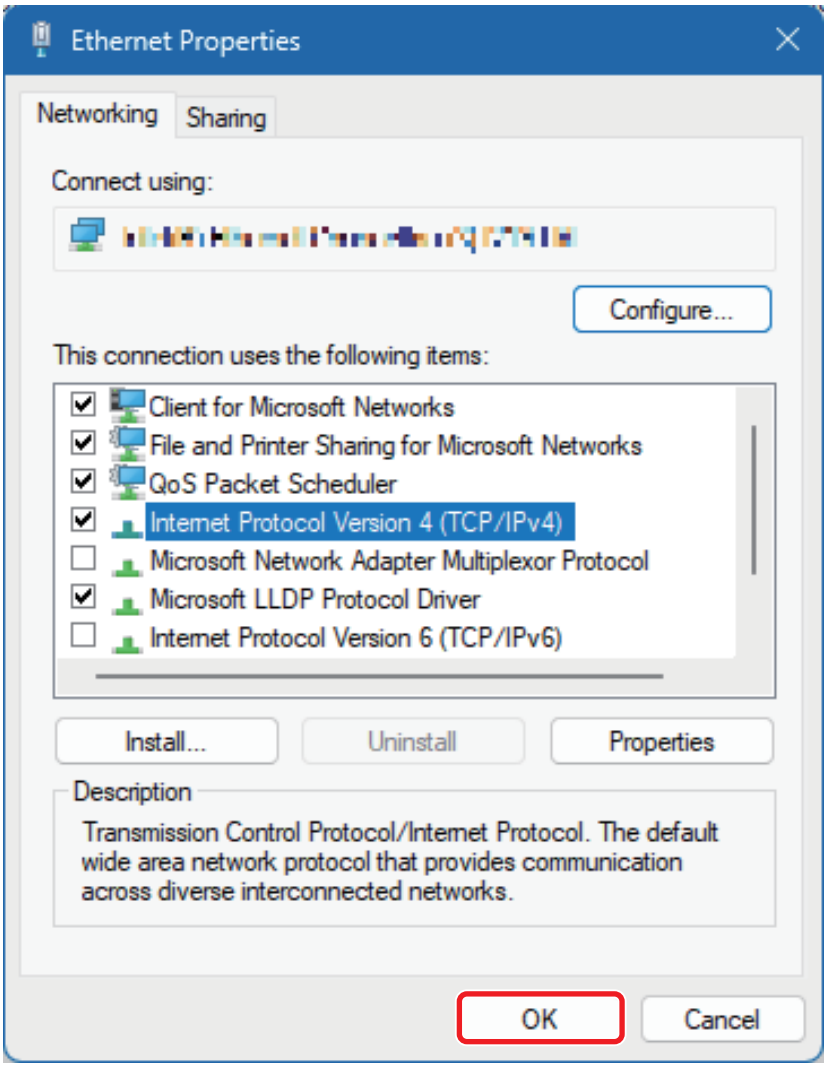
11. When the settings are complete, click the [OK] button.



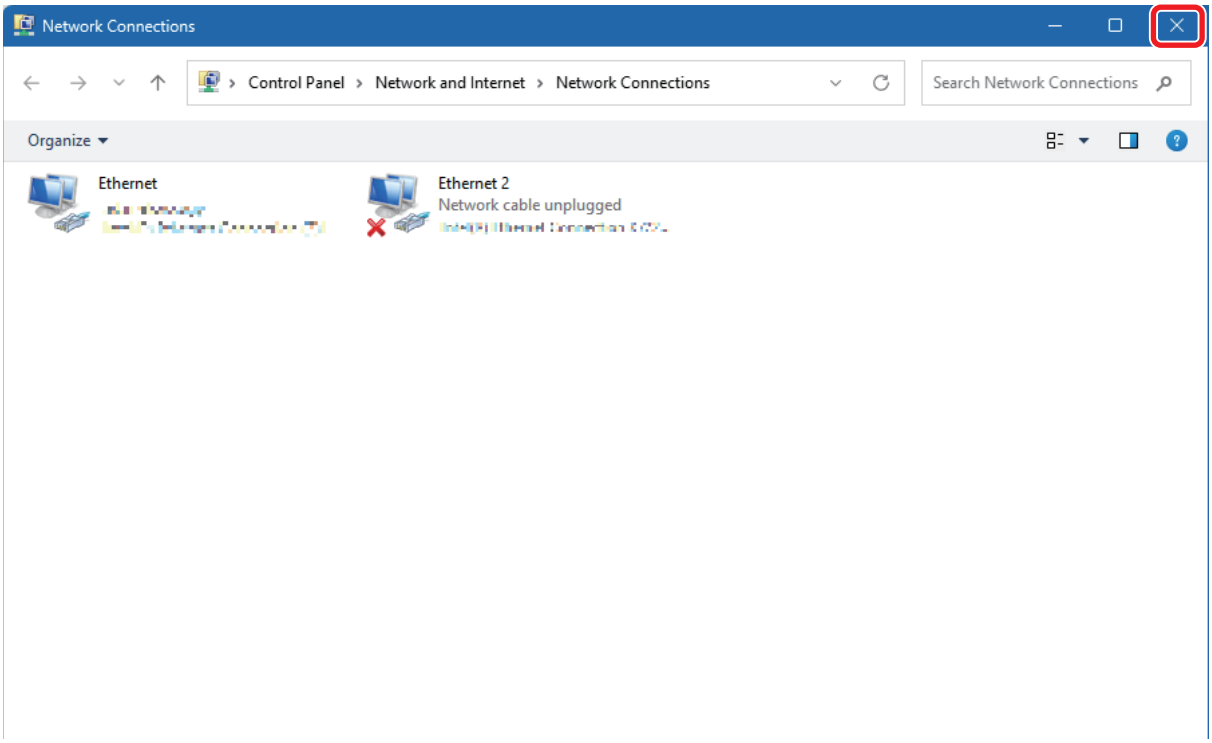
12. Click the [OK] button.



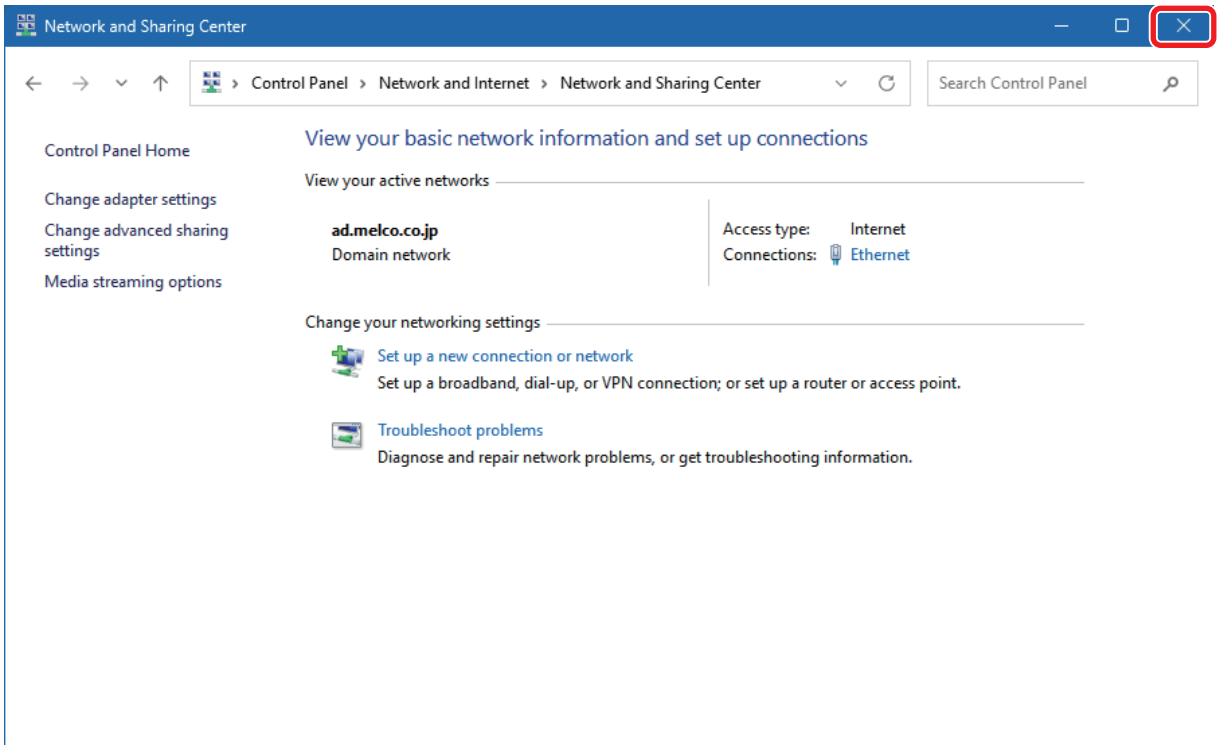
13. Click the [OK] button.



14. Close the "Network Connections" window.



15. Close [Control Panel].



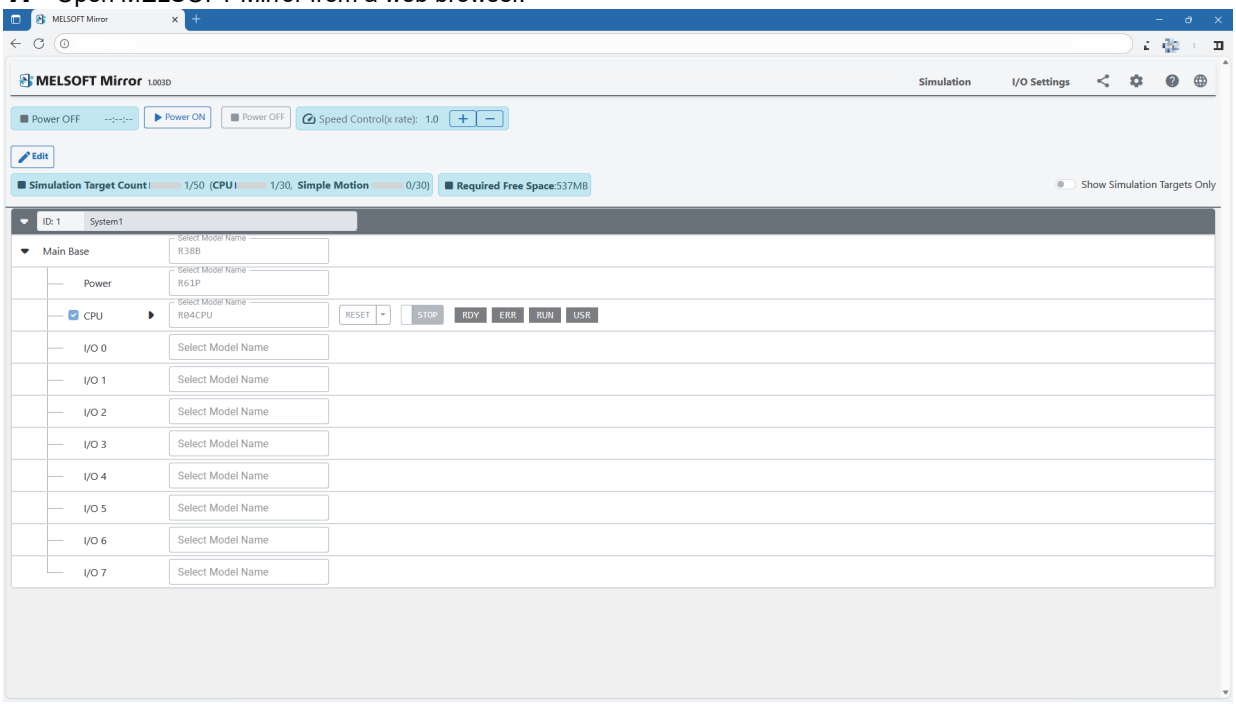
Precautions

The set IP address is added to the Ethernet adapter that can communicate.

When adding an IP address, check with the network administrator to make sure that the IP address to be added does not duplicate with the IP address of another Ethernet device.

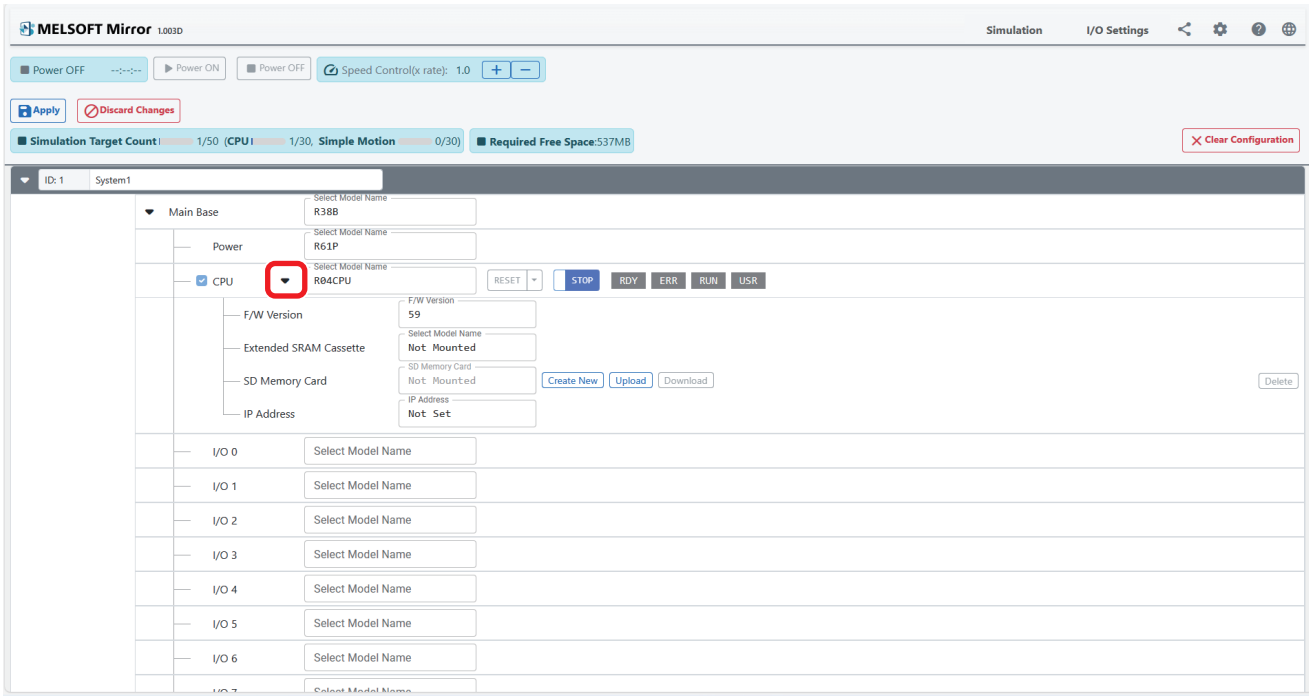
Settings on MELSOFT Mirror

1. Open MELSOFT Mirror from a web browser.

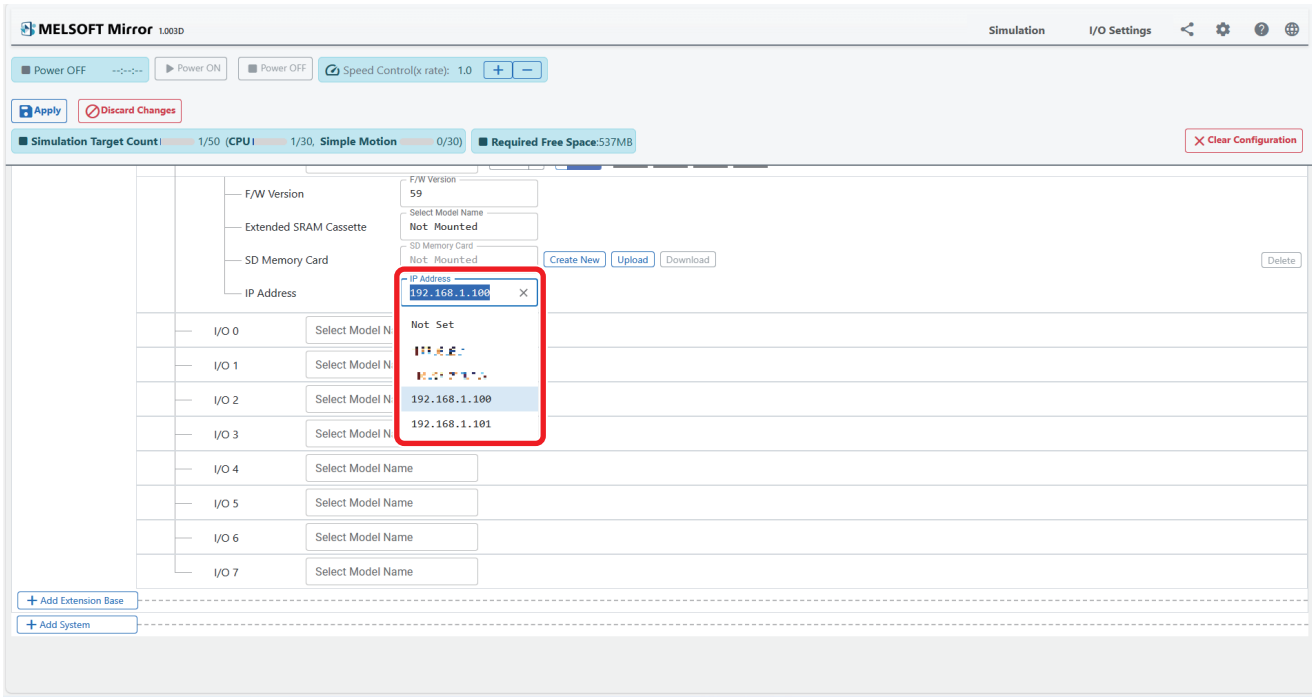


2. Click the [Edit] button.

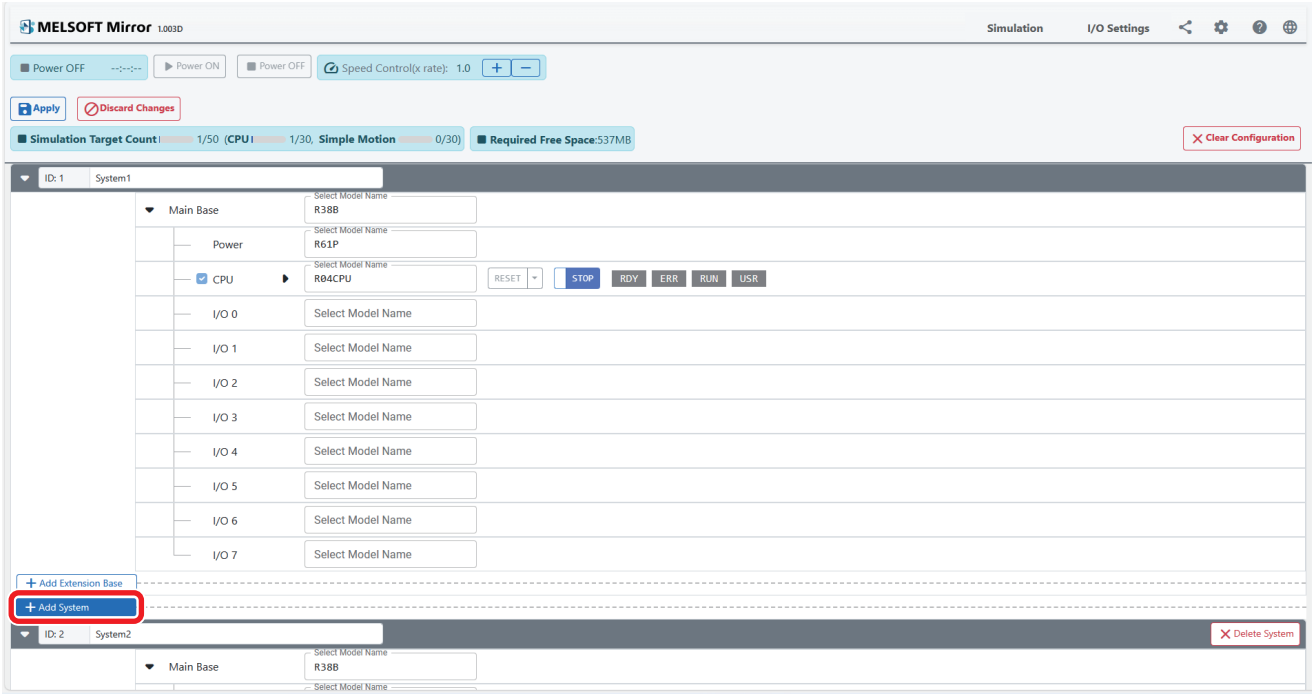
3. Select a CPU module.



4. Set the IP address of the CPU module. Multiple IP addresses set on the personal computer are displayed. Select an IP address from them.

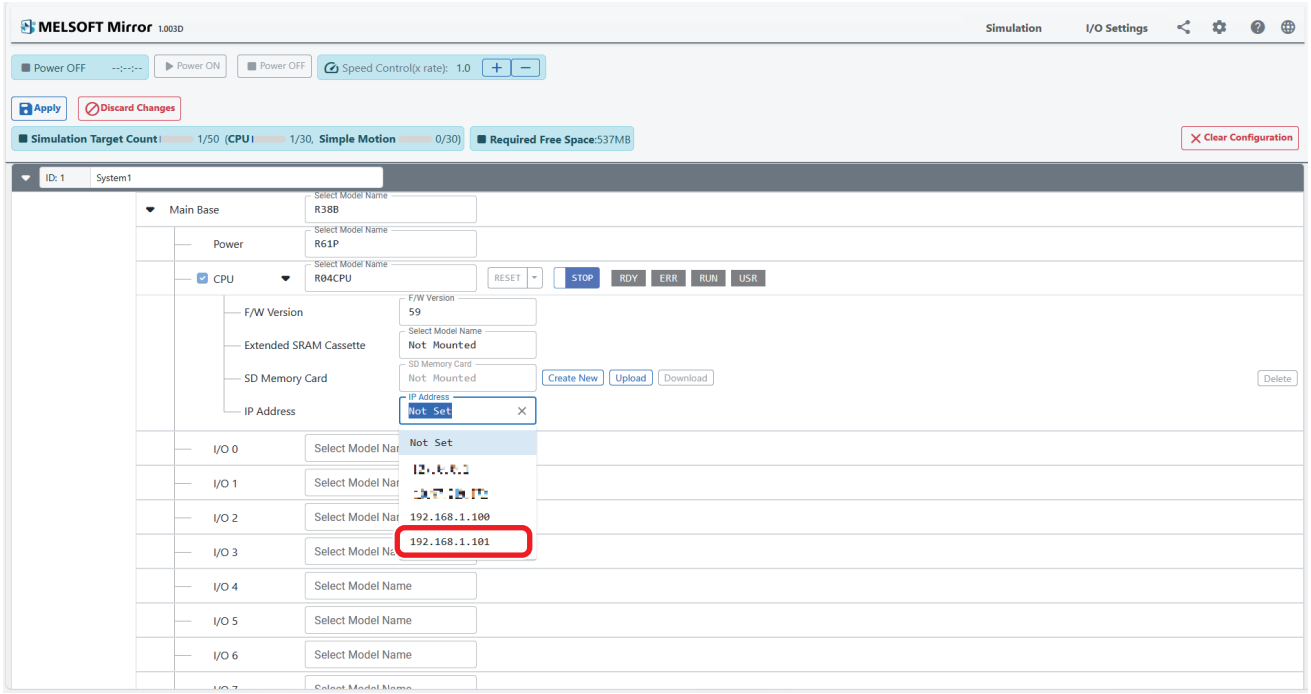


5. Click the [Add System] button.

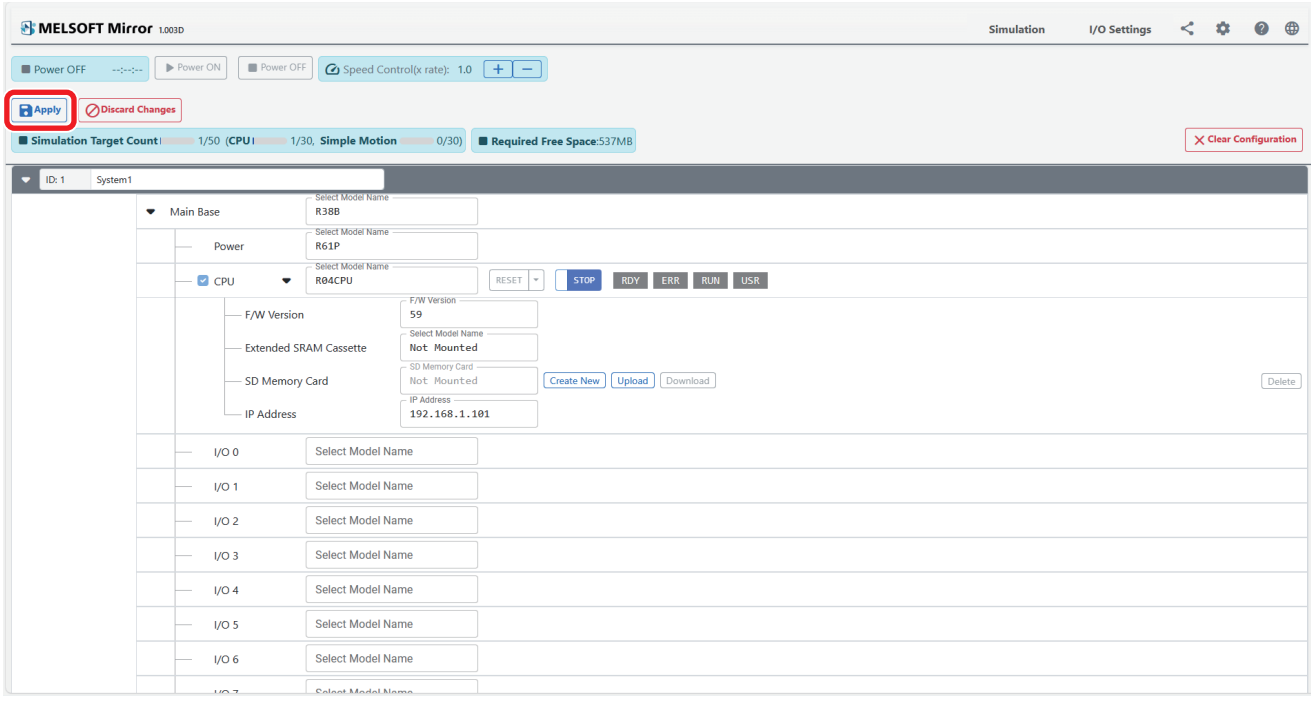


6. A new main base unit is added. Select a CPU module.

7. Select the IP address of the CPU module. At this time, select an IP address different from that set for the first CPU module.



8. Click the [Apply] button to complete the settings.



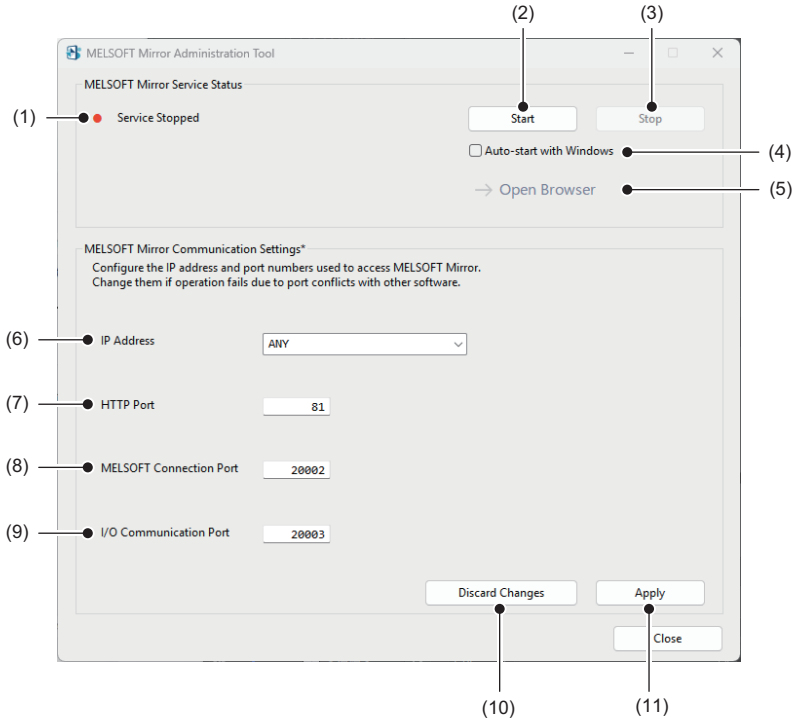
Point

After setting multiple IP addresses to the personal computer, if the set IP addresses are not displayed in MELSOFT Mirror, stop and start the MELSOFT Mirror service. (☞ Page 250 Checking the Start of Service)

4 SCREEN CONFIGURATION AND BASIC OPERATIONS

4.1 MELSOFT Mirror Management Tool

This section describes the screen configuration of MELSOFT Mirror Management Tool.



Name	Description
(1) Service status display	Displays the service status.
(2) Start	Starts the service.
(3) Stop	Stops the service.
(4) Auto-start with Windows	When selected, the service automatically starts when Windows starts.
(5) Open Browser	Opens the MELSOFT Mirror screen in the browser.
(6) IP Address	Specifies the IP address to access MELSOFT Mirror.
(7) HTTP Port	Specifies the HTTP port to access MELSOFT Mirror.
(8) MELSOFT Connection Port	Specifies the port for connecting from the engineering tool to MELSOFT Mirror.
(9) I/O Communication Port	Specifies the port for connecting from another simulator to MELSOFT Mirror via a FMU.
(10) Discard Changes	Restores the changes.
(11) Apply	Applies the changes.

The following table lists the IP addresses that can be selected and the URLs that can be connected to MELSOFT Mirror.

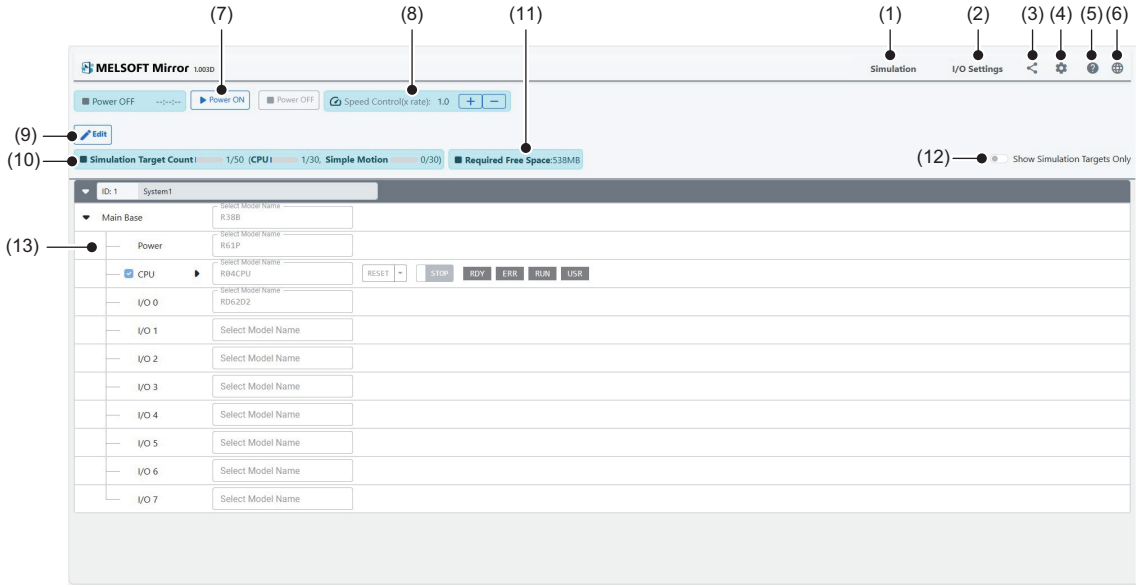
IP address that can be selected	URL that can be connected to MELSOFT Mirror	Remarks
ANY	127.0.0.1 localhost All IP addresses registered on the personal computer	Select this option to use any IP address registered on the personal computer.
127.0.0.1	127.0.0.1 localhost	Select this option to block access from an external personal computer.
IP address set on the personal computer	An IP address registered on the personal computer	All IP addresses registered on the personal computer are displayed, and any one of them can be selected.

Point

- Set the firewall on the personal computer on which MELSOFT Mirror is installed to allow data reception at the HTTP, MELSOFT connection, and I/O connection ports listed above, or disable the firewall on the personal computer on which MELSOFT Mirror is installed.
- MELSOFT Mirror Management Tool cannot be launched more than once at a time.

4.2 Simulation

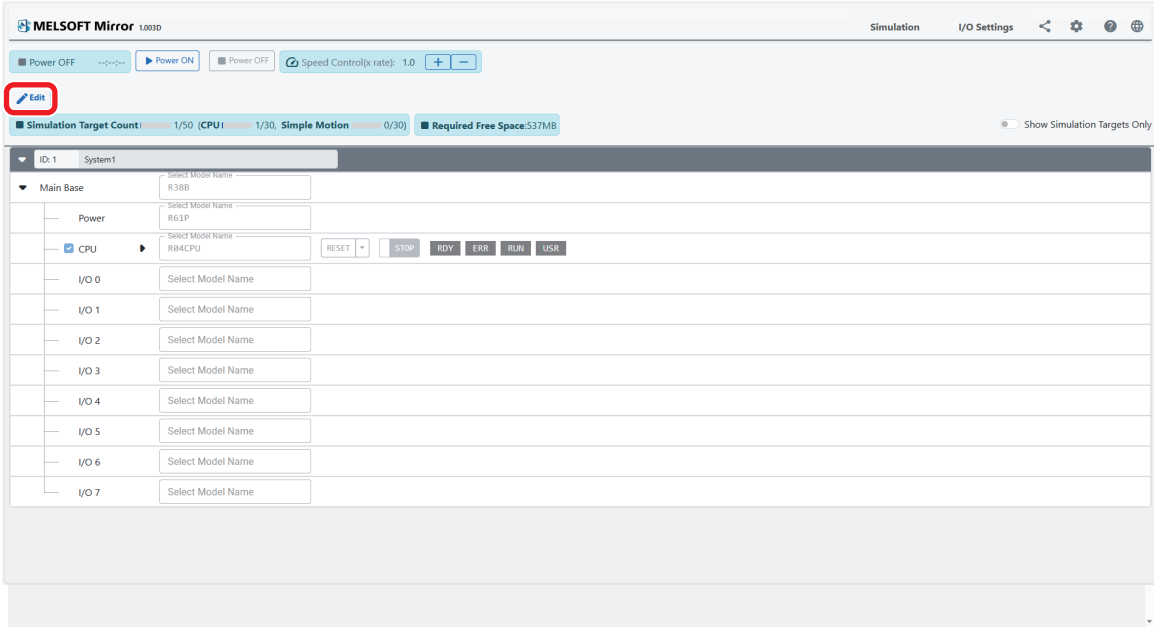
Enter the IP address of the personal computer on which MELSOFT Mirror is installed in the address bar of the web browser. The following top window is displayed.



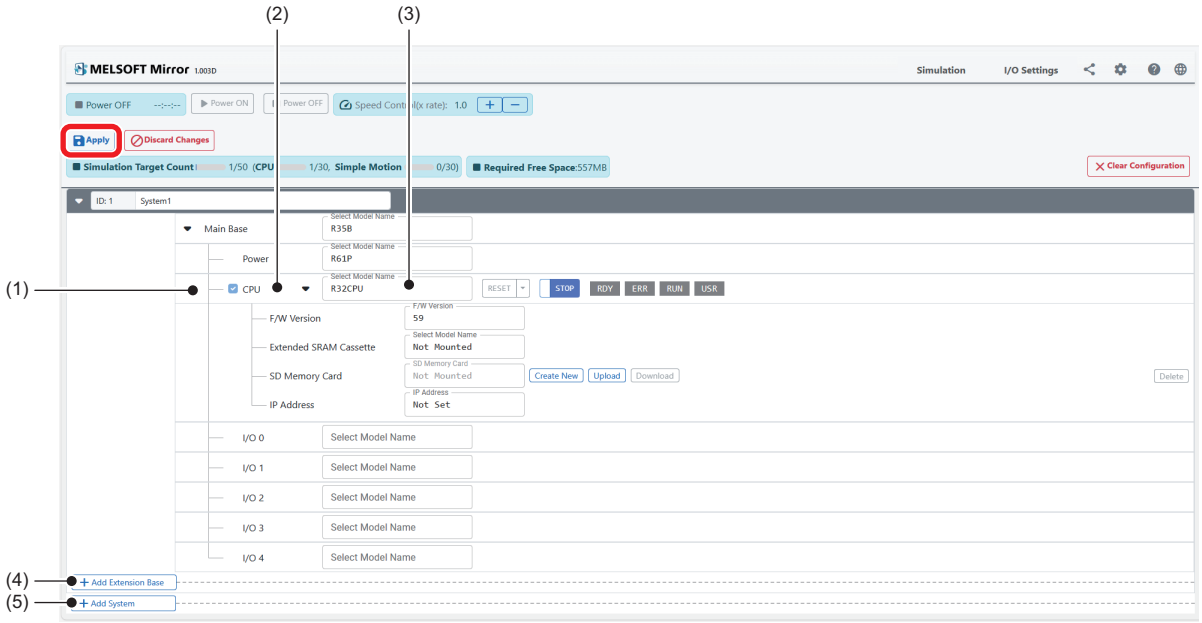
Name	Description
(1) Link to the Simulation window	A link to this window
(2) Link to the I/O Settings window	A link to the I/O Settings window
(3) Share	The URL of this window can be copied to the clipboard.
(4) Settings	Displays links to the License Management window and the Tools Download window.
(5) Help	Displays this manual.
(6) Display language	Changes the display language.
(7) Simulator power	Displays the simulator power status. The status is controlled by the [Power ON] and [Power OFF] buttons.
(8) Speed Control (x rate)	Displays the rate of the simulation execution speed. Use the [+] and [-] buttons to control the rate.
(9) Edit	Edits the simulation system configuration.
(10) Simulation Target Count	Displays the number of modules that can be simulated.
(11) Required Free Space	Displays the disk space used when the simulator is powered on.
(12) Show Simulation Targets Only	Displays only the modules that can be simulated in the system configuration.
(13) Simulation system configuration	Displays the current configuration of the simulation system.

Simulation Settings

Click the [Edit] button to edit the simulation system configuration.



Click the [Apply] button to save the simulation settings and finish editing.



Name	Description
(1) Module operation setting	Enables/disables the operation of a module during simulation using the checkbox. • Checkbox selected: Module operation is enabled. The module can be simulated. • Checkbox unselected: Module operation is disabled. The module cannot be simulated.
(2) Module setting details	Shows/hides setting details of a module. When displayed, the settings can be changed. For the CPU module, the following settings can be changed. • [F/W Version] • [Extended SRAM Cassette] • [IP Address]*1 • [SD Memory Card]

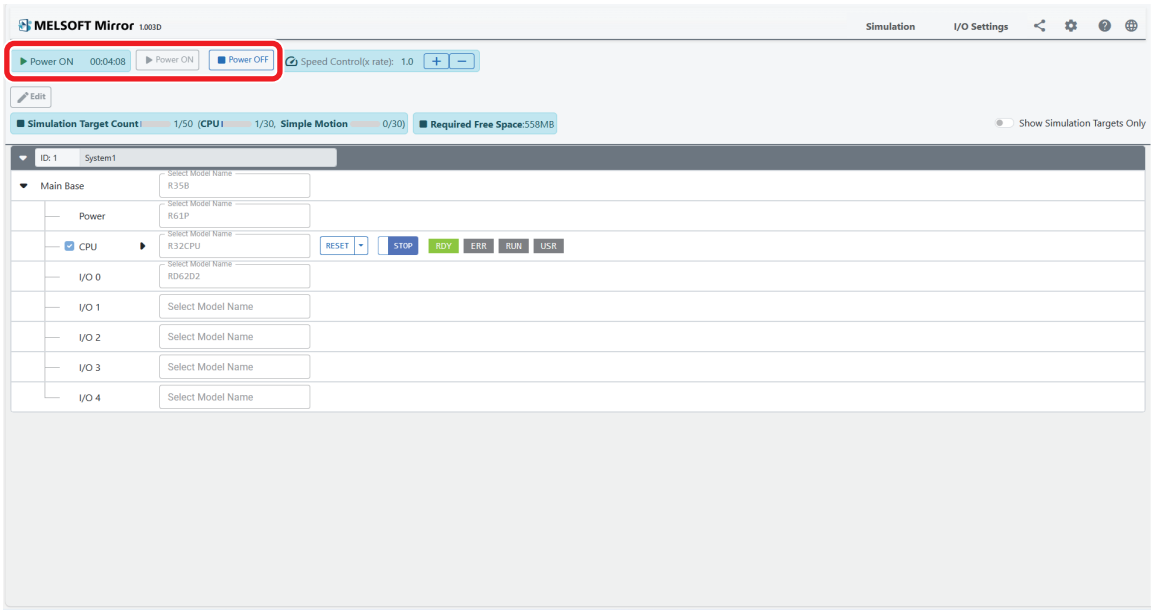
Name	Description
(3) Module selection	Selects a module to be placed on the main base unit from the drop-down list. A module with "(2Slot)" at the end of the model name occupies two slots. For the R12CCPU-V and RnENCPU, the next slot of the slot on which the module is mounted must be empty. For other 2-slot modules, the previous slot of the slot on which the module is mounted must be empty.
(4) Add Extension Base	Adds an extension base unit to the main base unit.
(5) Add System	Adds a new main base unit to the simulation system. A new ID is assigned to the new main base unit.

*1 The IP address of the CPU module can be set. Depending on the application, the engineering tool may use the IP address that was set in the module parameter of the CPU module.
For details, refer to the following. (☞ Page 280 IP address setting of the CPU module)

Simulation Operation

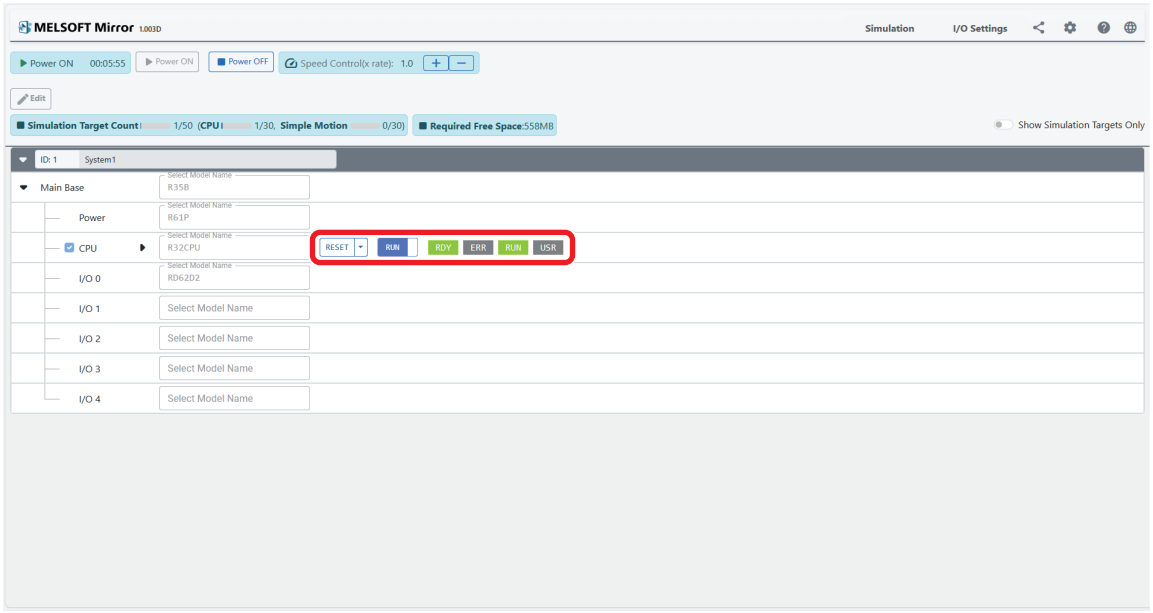
To run the simulation, click the [Power ON] button.

To stop the running simulation, click the [Power OFF] button.



Module Display and Operation

While the simulation is running, the LEDs of the module running the simulation flash or turn on.



Precautions

Flashing at intervals of 2 seconds or less is not supported because the LEDs refresh at one second intervals.

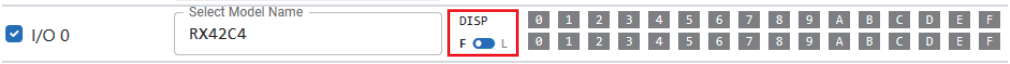
Switch operation

■CPU module

- RESET: Resets the CPU module.
- RUN/STOP: Changes the operating status of the CPU module between RUN and STOP.

■I/O module

For input or output modules that support simulation and have 64 I/O points, the following LED display selector switch is displayed.



Use this switch to change which 32-point block is displayed on the LEDs:

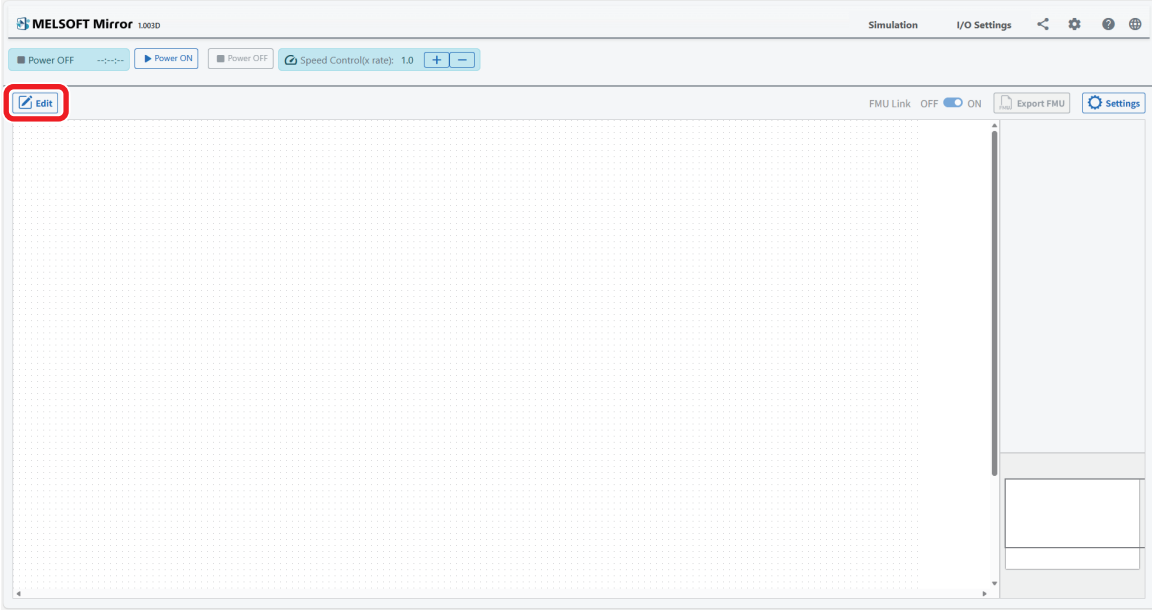
- F: Display the state of the first 32 points on the LEDs.
- L: Display the state of the last 32 points on the LEDs.

4.3 I/O Setting

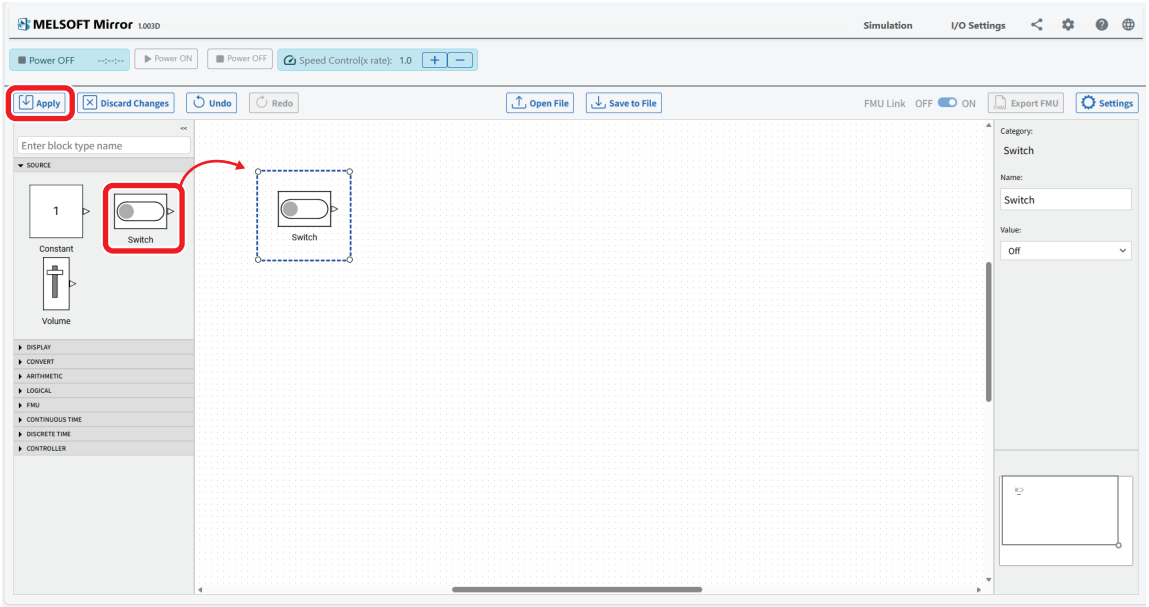
Click [I/O Settings] to display the I/O Settings window.

Editing the I/O settings

External devices can be edited by clicking the [Edit] button on the I/O Settings window.
To return to the Simulation window, click [Simulation].
For details on the setting method, refer to the following. (☞ Page 118 I/O SETTING)



Drag and drop the symbol of an external device from the list to the configuration area. Then, connect the external device to the programmable controller.
To finish editing, click the [Apply] button.



Precautions

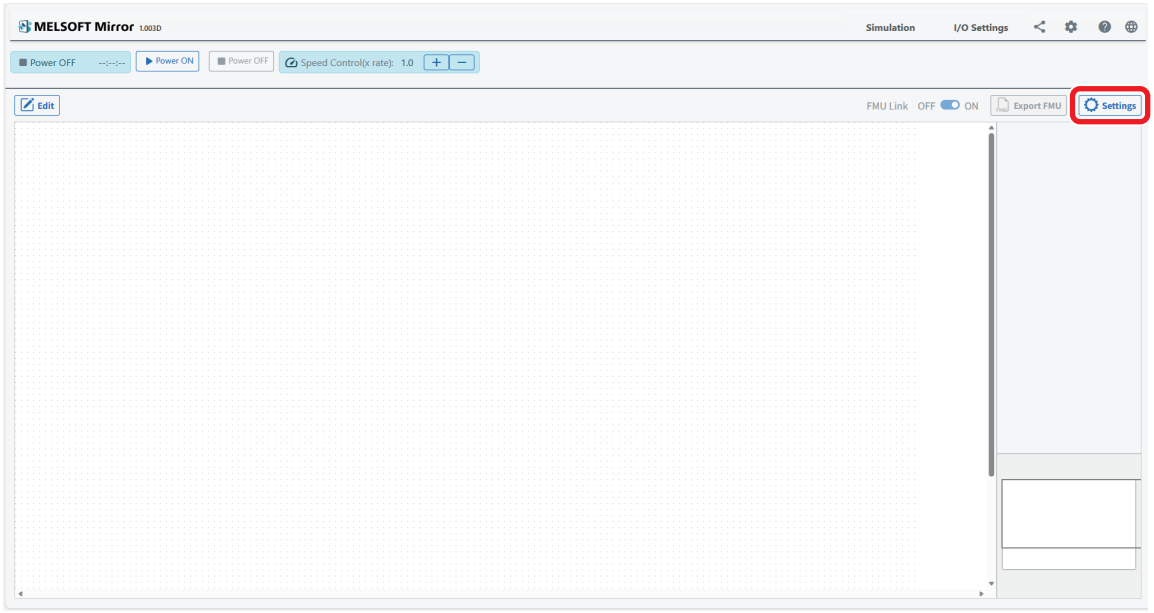
When clicking the [Power ON] button to power on the simulator, all of the following conditions must be met.

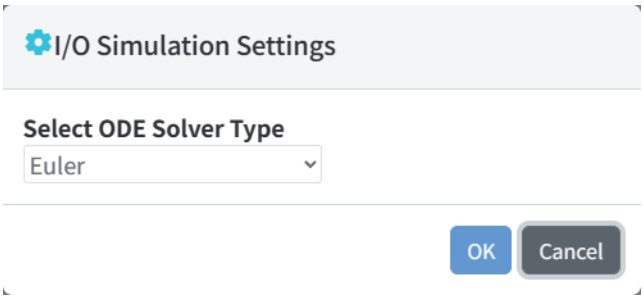
- Modules corresponding to the simulation system configuration are placed.
- Modules that are not in the simulation system configuration are not placed.
- Two or more blocks are placed, and one or more wires are connected between the output terminal and the input terminal.
- All input terminals are connected. Note that controller blocks are excluded.
- When the block output is looped and input to the block, the loop includes either of the following blocks: an integral block, a primary delay block, a secondary delay block, a primary hold block, or a delay block with a delay time other than 0.

If the above conditions are not met, an error message will be displayed and the simulator will not power on.
In this case, edit the I/O setting again so that all the conditions are met.

I/O simulation setting

Click the [Settings] button to display "I/O Simulation Settings".





The following table lists the setting item.

Name	Description
ODE Solver Type Selection	Determines how to solve ordinary differential equations for simulation in I/O setting.

The following table lists the characteristics of each option.

ODE solver	Execution speed	Calculation accuracy
Euler	Fast	Medium
Fourth-Order Runge Kutta	Slow	High

The ODE solver selection impacts the execution speed and calculation accuracy of the following blocks.

- Integral block (☞ Page 194 Integral block)
- Primary delay block (☞ Page 199 Primary delay block)
- Secondary delay block (☞ Page 201 Secondary delay block)

Change the ODE solver selection to adjust the execution speed and calculation accuracy of these blocks.

FMU link function

This function connects MELSOFT Mirror to another simulator via a FMU.

Restriction

When importing a FMU file exported from MELSOFT Mirror into Simulink and running a co-simulation, the following restrictions on the operating procedure apply.

- To start the simulation, click the [Power ON] button on MELSOFT Mirror first. After the "Waiting for FMU Communication" window appears, start the Simulink simulation.
- To stop the simulation, there is no required order: you may power off MELSOFT Mirror or stop the Simulink simulation in either order.
- In Simulink co-simulation, the step forward, pause, and variable-step cannot be used.

These restrictions also apply when MELSOFT Mirror is used together with FMI master software other than Simulink.

FMU export

On the "I/O Settings" window (view mode), click the [Export FMU] button to export a FMU file.

Importing the exported FMU into other simulation software enables co-simulation with MELSOFT Mirror.

Point

If LC_Simulator.fmu already exists in the download folder, the exported file will be saved under an alternate name (for example, LC_Simulator (1).fmu).

FMU link

On the "I/O Settings" window (view mode), turn on [FMU Link] to enable the FMU link function.
The FMU link function starts when you click the [Power ON] button.

Point

To enable the FMU link function, place one or more FMU output blocks or FMU input blocks in the block diagram.

4.4 License Management

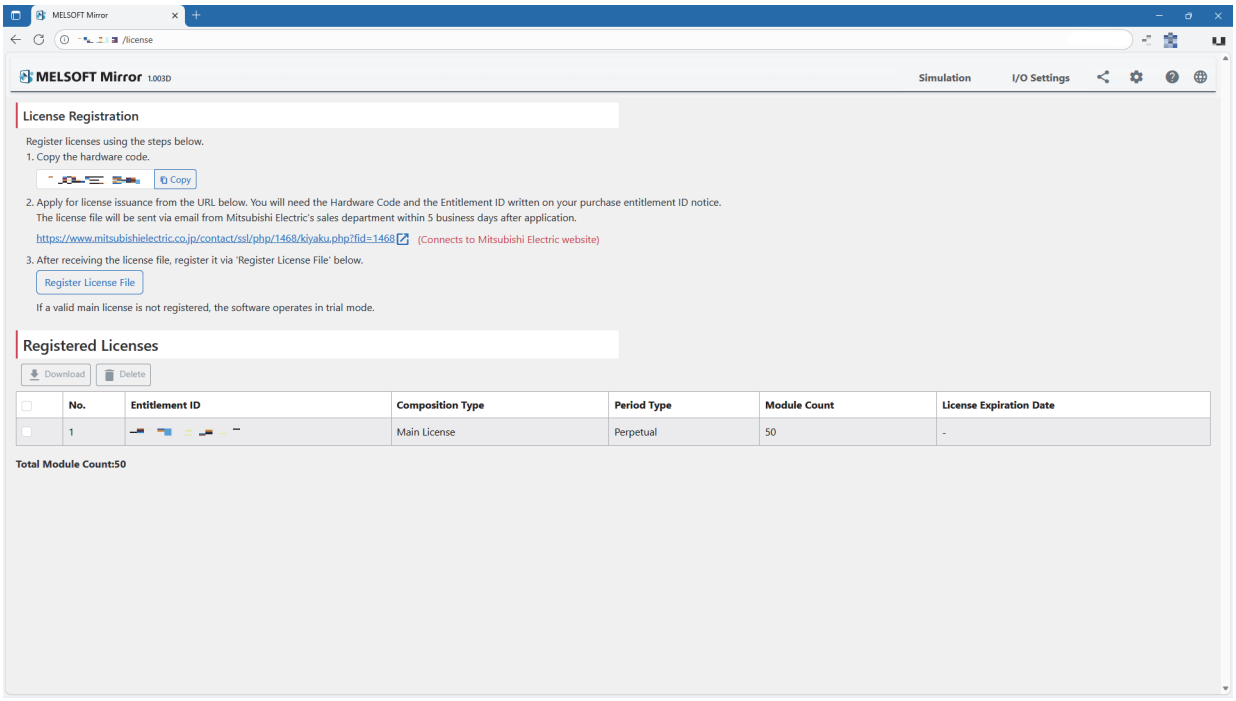
For license registration, refer to the following. (☞ Page 22 MELSOFT Mirror License Registration)

The License Management window displays "Registered Licenses".

When the number of days remaining before the license expires is as follows, a warning message is displayed.

- One week (daily warning message until expired)

The license expiration time is 23:59 of the license expiration date at UTC-12. The expiration time in each time zone varies depending on the time difference from UTC-12.



5 START-UP PROCEDURE

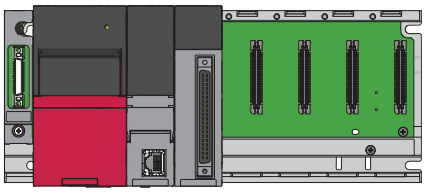
This section describes the procedure for simulating a system with MELSOFT Mirror.
After installing MELSOFT Mirror, connect to MELSOFT Mirror from a web browser.

5.1 System Configuration Example

The following system configuration is used for explanation.

System configuration

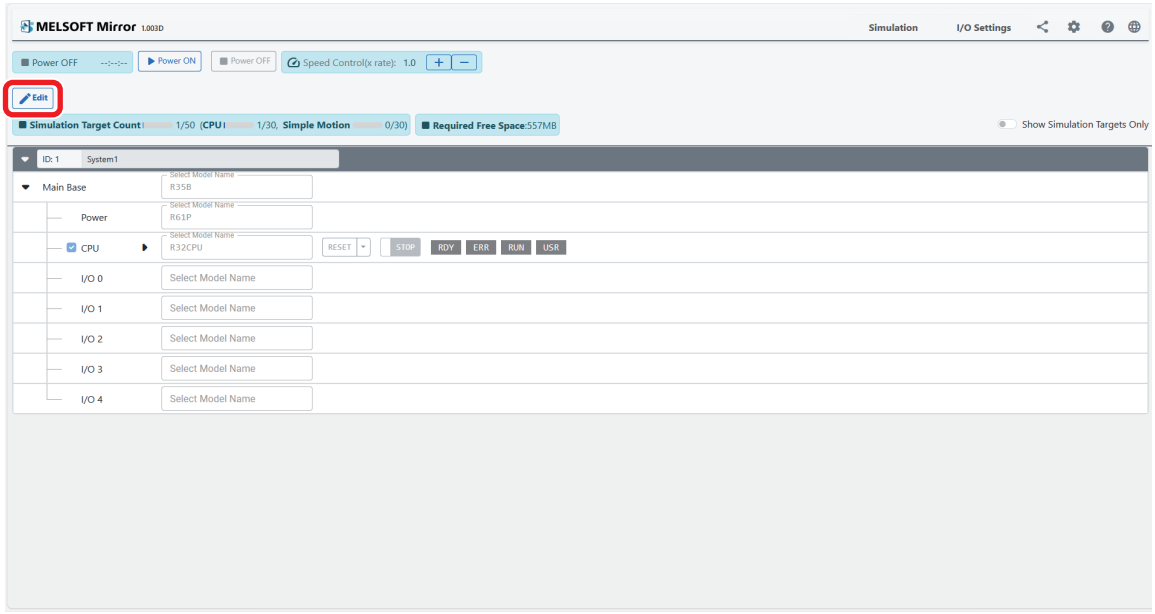
- Main base unit: R35B
- Power supply module: R61P
- CPU module: R32CPU
- High-speed counter module: RD62D2 (start I/O number: 0000H to 000FH)



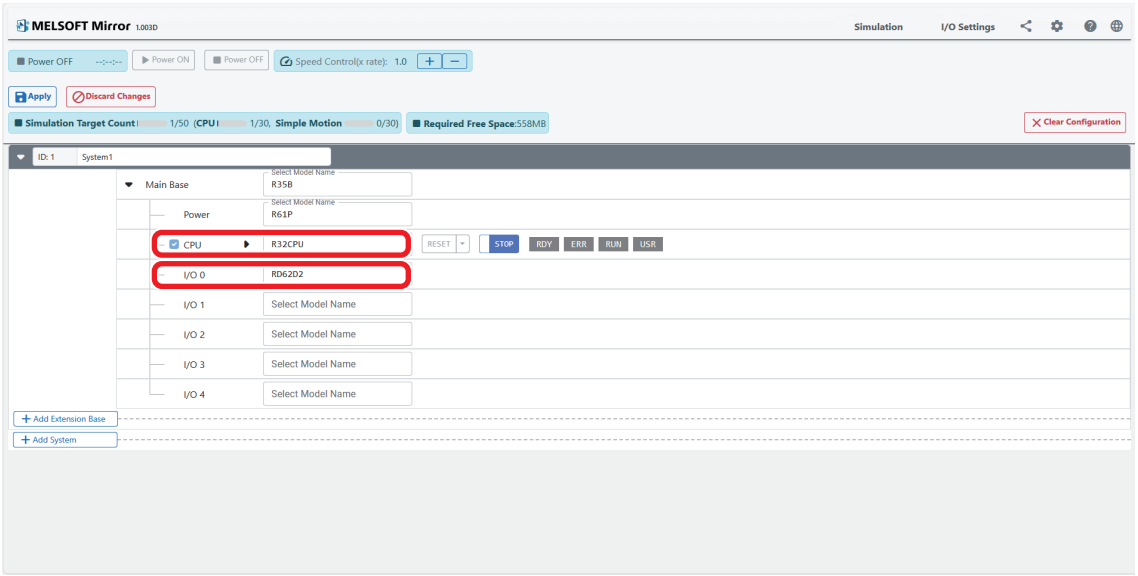
5.2 Setting System Configuration

Set the programmable controller system configuration on MELSOFT Mirror.

1. Click the [Edit] button.



2. Edit the configuration of the modules.



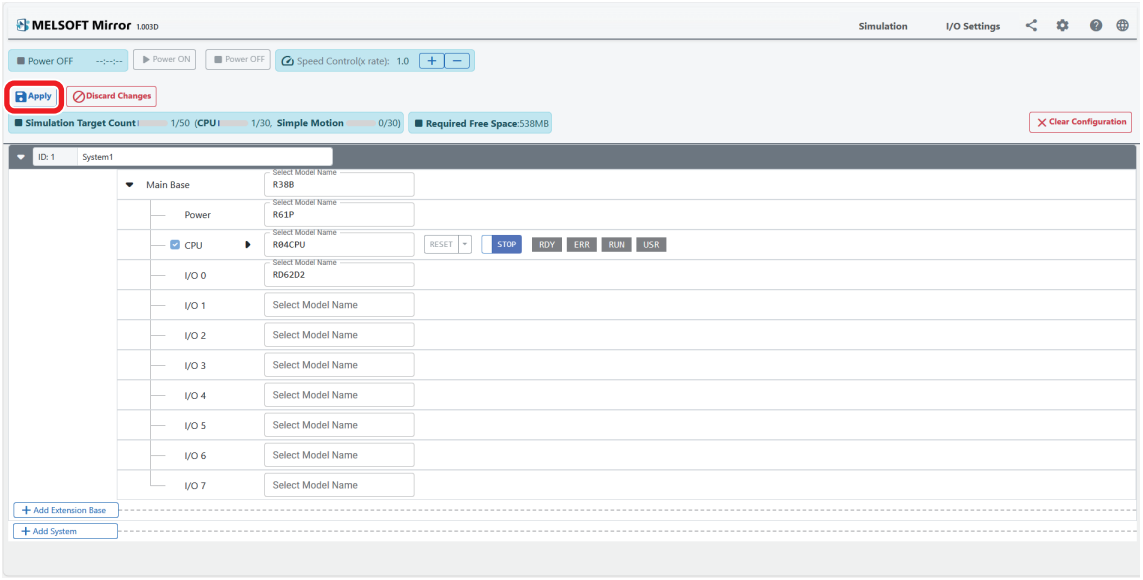
Point

To add an extension base unit, click the [Add Extension Base] button.
To add a main base unit, click the [Add System] button.

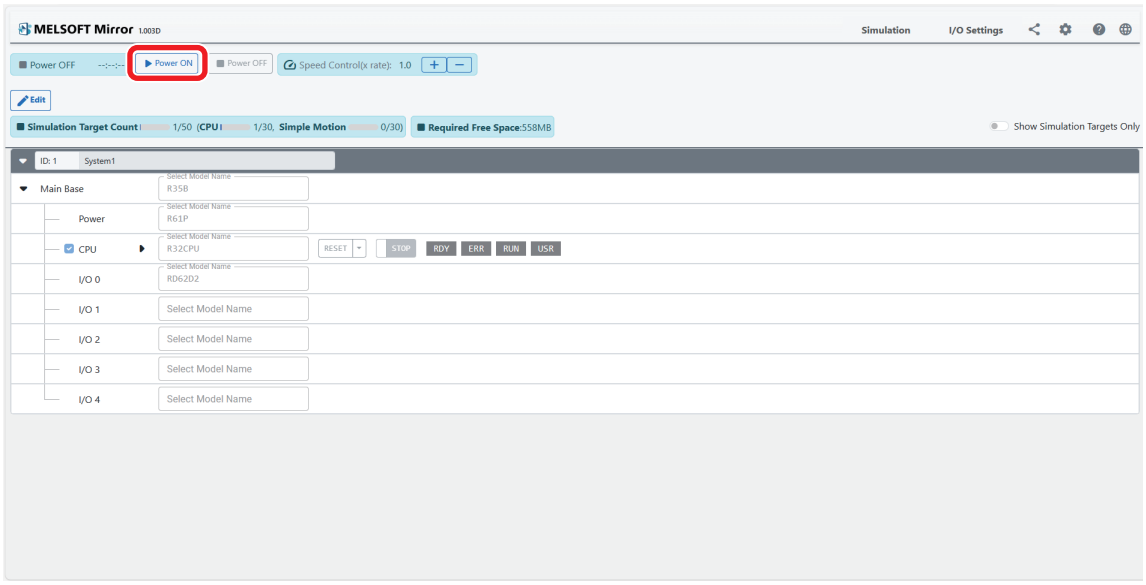
3. Select [R32CPU] from the "CPU" combo box.

4. Select [RD62D2] from the "I/O0" combo box.

5. Click the [Apply] button.



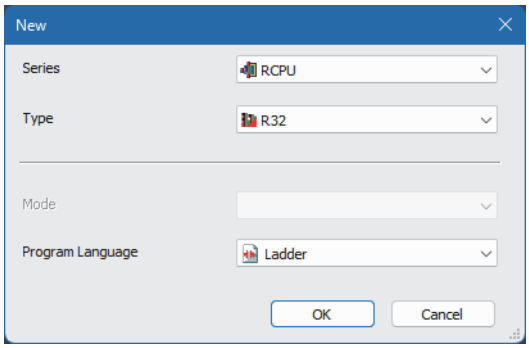
6. Click the [Power ON] button to start the simulation.



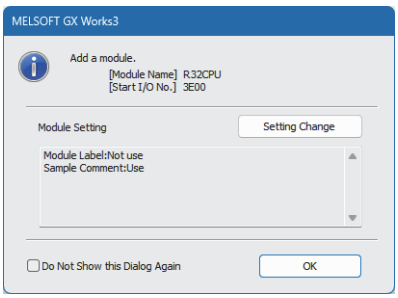
5.3 Creating a Project with the Engineering Tool

Procedure

Create a program, following the procedure below. This example uses ladder diagram as programming language.



1. Select "RCPU" for "Series" and the CPU module model to be used for "Type". Then, select a programming language to be used in the project for "Program Language". In this example, select "Ladder", and click the [OK] button.



2. When the window for adding a module appears, click the [OK] button.

Point

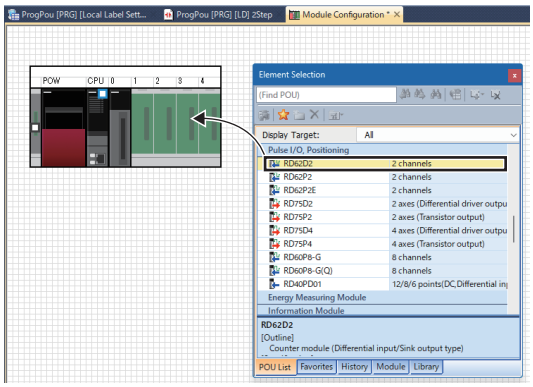
- A module label is a label where the I/O signals and buffer memory areas of a module have already been defined. The use of module labels allows programming without being aware of module internal addresses.
- A sample comment is a comment where system devices (special relay/special register) and buffer memory areas of a module have already been defined, The use of sample comments saves time for entering comments.

5.4 Setting Parameters

This section describes how to set system parameters and parameters for each module.

Setting parameters on the Module Configuration window

Set parameters by configuring a system manually on the Module Configuration window of the engineering tool.



1. Select a base unit on the Element Selection window, and drag and drop it to the Module Configuration window.
2. Drag and drop modules to be used on the base unit placed.
3. Select [Edit] ⇨ [Parameter] ⇨ [Fix] on the menu bar.
4. Click the [Yes] button when the window for adding module labels of the configured modules appears.
5. Open the parameter editor of each module by double-clicking the module.
6. Set parameters, and click the [Apply] button to close the window.

Point

- Match the settings on the Module Configuration window of the engineering tool with the module configuration set in MELSOFT Mirror.
- To use the RnENCPU, the modules can be found from different categories. Select "PLC CPU" for the RnENCPU (CPU part) and "CPU Extension" for the RnENCPU (network part).

Setting parameters from the Navigation window

Set the following parameters from the Navigation window.

■System parameters

These parameters need to be set from the Navigation window in the following cases: to change the number of slots on the base unit or the number of occupied points of the module; for a multiple CPU system; and for module synchronization operations.

[Navigation window] ⇨ [Parameter] ⇨ [System Parameter]

■Module parameters of the CPU module

These parameters are required to execute the Ethernet function of the CPU module.

[Navigation window] ⇨ [Parameter] ⇨ [Module model name] ⇨ [Module Parameter]

■Memory card parameters

These parameters are required to execute functions that access the SD memory card.

[Navigation window] ⇨ [Parameter] ⇨ [Module model name] ⇨ [Memory Card Parameter]

■Multiple module parameters and module extension parameters

Some intelligent function modules require multiple module parameters and module extension parameters.

[Navigation window] ⇨ [Parameter] ⇨ [Module Information] ⇨ [Intelligent function module] ⇨ [Module Parameter] or [Module Extended Parameter]

Point

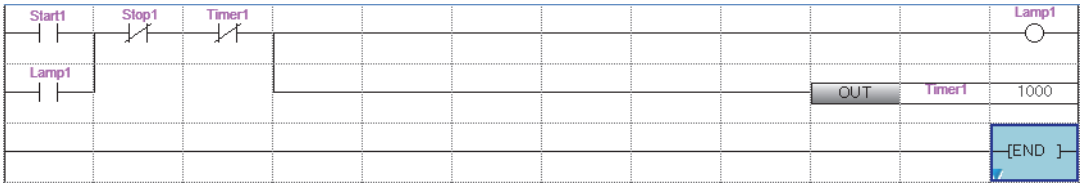
System parameters can be set on the following windows. Select the window depending on their application purposes.

- Module Configuration window: Use this window to use the module specific data (such as number of occupied points) as is.
- System Parameter window: Use this window to change the number of slots or the number of occupied points.

5.5 Create Programs

Create programs. This section describes how to create a program using the following program example.

Program example



- When Start1 turns on, Timer1 starts counting, and Lamp1 turns on.
- When the current value of Timer1 reaches 1000, Lamp1 turns off.
- When Stop1 turns on, Lamp1 turns off.

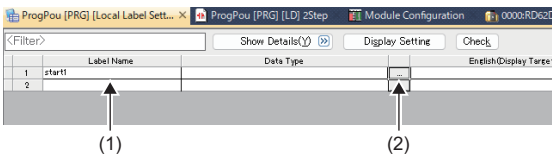
Registering labels

A label is a variable whose name and data type can be declared by a user.
The use of labels allows programming without being aware of devices and buffer memory addresses. For this reason, programs using labels can be used in other systems where the module configuration is different.
Labels can be registered on the label editor.

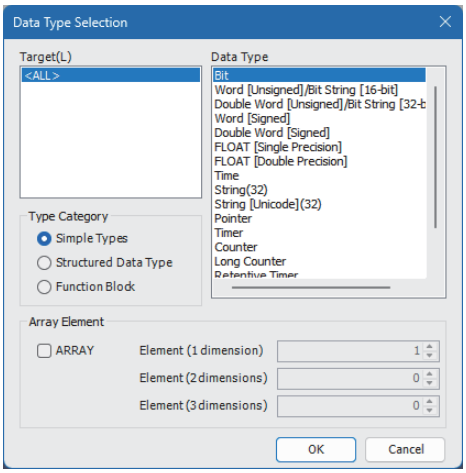
[Navigation window] ⇨ [Program] ⇨ [Scan] ⇨ [MAIN] ⇨ [ProgPou] ⇨ [Local Label]

Procedure

Register the label "Start1" in the program example, following the procedure below.



1. Enter the name, "Start1", in the "Label Name" field (1).
2. Click the button (2) on the right of the "Data Type" field to open the "Data Type Selection" window.



3. Specify the data type. Select "Bit", and click the [OK] button.

Register other labels in the program example in the same way.

Point

- The class, initial value, and constant of labels can be set as needed by clicking the [Show Details] button on the label editor.
- Labels can also be registered while programming without opening the label editor.
- Devices can be assigned to global labels. Open the global label editor, and enter a device in the "Assign (Device/Label)" field.

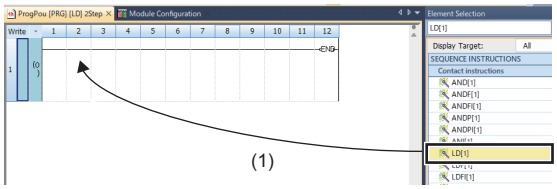
Inserting program elements

Drag and drop required program elements to the ladder editor.

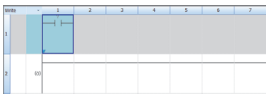
[Navigation window] ⇨ [Program] ⇨ [Scan] ⇨ [MAIN] ⇨ [ProgPou] ⇨ [ProgramBody]

Procedure

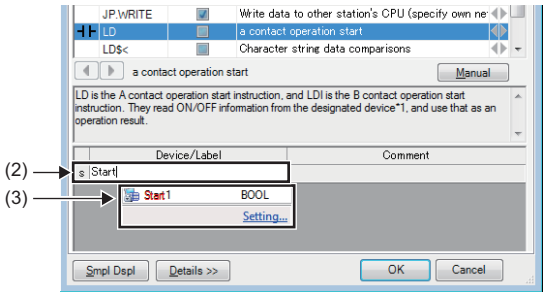
Insert a normally open contact of "Start1" in the program example, following the procedure below.



1. Select a program element from the Element Selection window, and drag and drop (1) it to the desired position on the ladder editor. In this example, drag and drop "LD[1]".

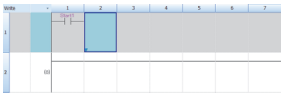


2. Double-click the inserted program element, and click the [Extd Dspl] button to open the ladder entry window.



3. Specify the operand. In this example, enter "Start" for "s" (2) in the "Device/Label" field.

4. Select an item from the displayed list (3). In this example, select "Start1".

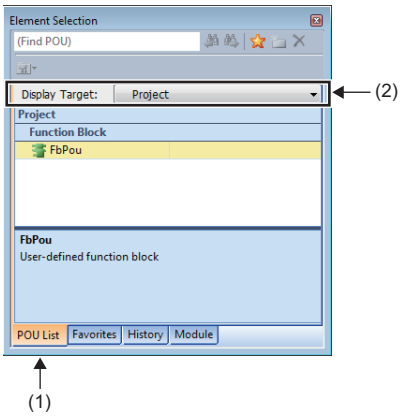


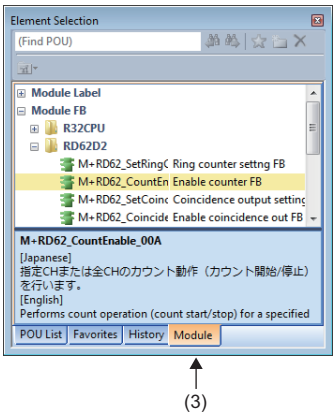
5. The normally open contact of "Start1" is inserted to the program.

Insert other program elements in the program example in the same way.

Point

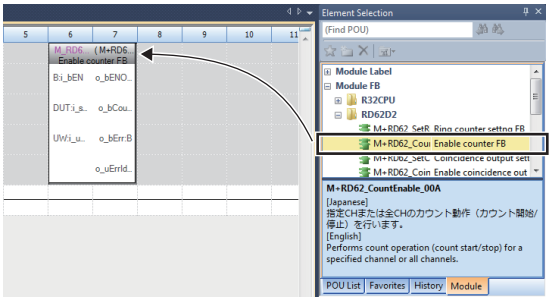
- Common instructions, such as contacts and coils, standard functions/function blocks, and user-defined function blocks/functions can be found in the [POU List] tab (1) of the Element Selection window. Narrow the list using the drop-down menu (2) for "Display Target", and select program elements.
- Module labels and module function blocks can be found in the [Module] tab (3). Programs are efficiently created only by dragging and dropping the module labels and module function blocks on the editor.



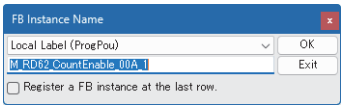


■Inserting function blocks

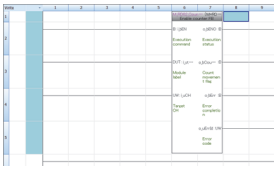
Insert function blocks, following the procedure below.



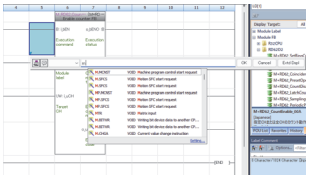
1. Select a function block from the Element Selection window, and drag and drop it to the desired position on the ladder editor.



2. The "FB Instance Name" window opens. Select the target label (global label or local label), and enter an instance name.



3. Select [Convert] ⇨ [Convert] on the menu bar. The ladder is converted, and the rungs are connected to the input and output labels of the FB instance.

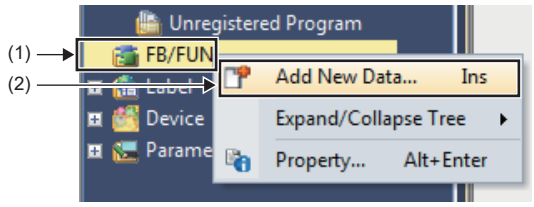


4. Add the input and output parts of the inserted function block to complete the program. For details on the input and output parts, refer to the following.

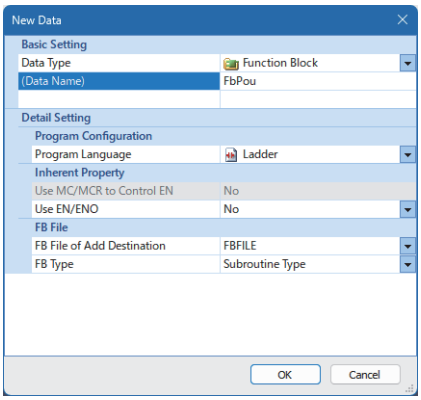
Function Block Reference for the module used

■Creating function blocks/functions

User-defined function blocks/functions must be created before inserted to the program.



1. Right-click "FB/FUN" (1) on the Navigation window, and select "Add New Data" (2) to open the setting window.



2. To create a function block, set the data type to "Function Block" using the drop-down list. To create a function, set the data type to "Function". Then, click the [OK] button.

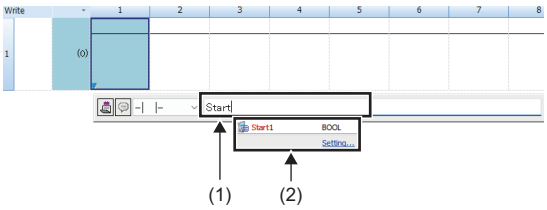
3. Program the operation of the function block or function in "ProgramBody".

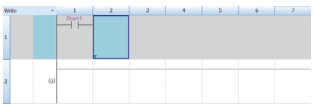
Inserting POUs by key input

POUs can be inserted by key input.

Procedure

Inserting a normally open contact of "Start1" in the program example, following the procedure below.

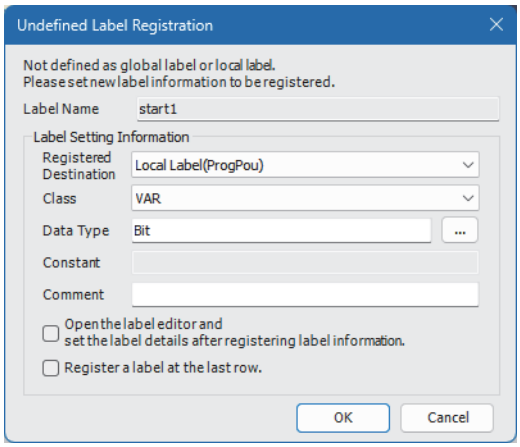




1. Click the insertion position on the ladder editor, and press F5.
2. Enter the name, "Start", in the entry field (1). Select "Start1" from the displayed list (2).
3. The normally open contact of "Start1" is inserted to the program.

Point

A new label can be registered during insertion. Enter the name of a new label on the ladder entry window, and click the [OK] button. Then, specify the registered destination, class, and data type of the label on the "Undefined Label Registration" window, and click the [OK] button.



5.6 Converting the Program

Determine the input ladder blocks.

Procedure

1. Select [Convert] ⇨ [Convert] on the menu bar.
2. When the conversion processing completes and the input ladder blocks are determined, the color of those ladder blocks changes from gray to white.

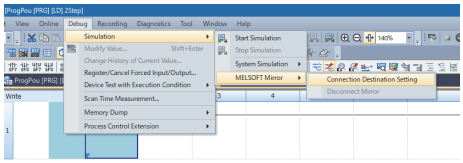
5.7 Saving the Project

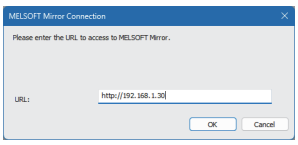
Save the created project.

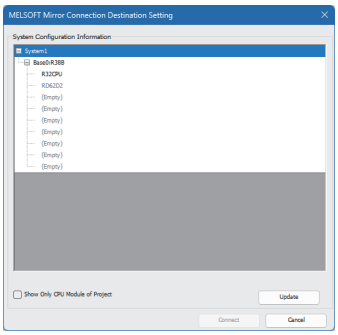
[Project] ⇨ [Save as]

5.8 Connecting the Engineering Tool to MELSOFT Mirror

Connect the engineering tool to MELSOFT Mirror.







1. Connect the personal computer on which the engineering tool is installed and the personal computer on which MELSOFT Mirror is installed over Ethernet.
2. Select [Debug] ⇨ [Simulation] ⇨ [MELSOFT Mirror] ⇨ [Connect Simulation] on the menu bar of the engineering tool.
3. Enter the URL on the top window of MELSOFT Mirror (example: http://192.168.1.30). The URL can be copied using the [Share] button on the top window of MELSOFT Mirror. (☞ Page 41 Simulation)
4. The system configuration set in MELSOFT Mirror is displayed.
5. Select the CPU module, and click the [Connect] button. The window is closed and the engineering tool is connected to MELSOFT Mirror.

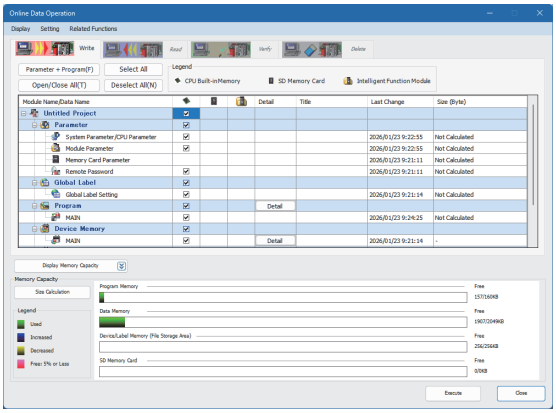
5.9 Writing Data to the Programmable Controller

Write the set parameters and the created program to the CPU module.

At this time, check that MELSOFT Mirror is powered on.

[Online] ⇨ [Write to PLC]

Procedure



1. Select the system parameter file, CPU parameter file, module parameter file, and program file on the "Online Data Operation" window. If any function block is used, select the corresponding FB/FUN file as well.
2. Click the [Execute] button.
3. After the write processing completes, click the [Close] button.

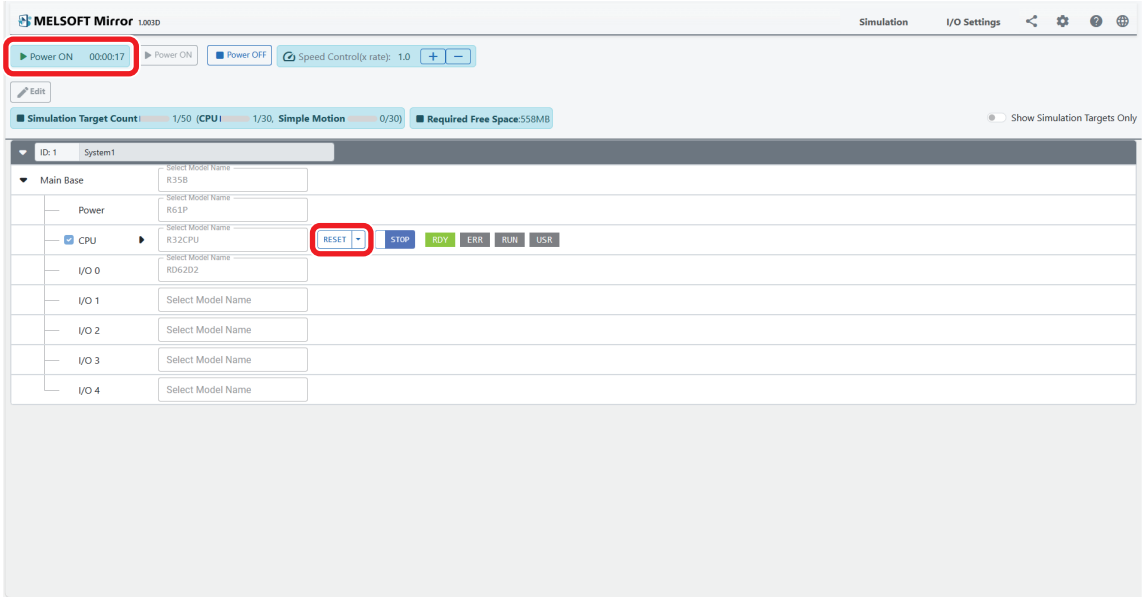
Point

- For the operation of the CPU module, the system parameter file, CPU parameter file, and program file must be written. For the operation of the I/O modules and intelligent function modules, the module parameter files and module extension parameter file must be written as well.
- When a parameter is newly set or changed, reset the CPU module in MELSOFT Mirror.
- Use of the [Select Favorites] button enables users to easily select frequently-used files, such as the system parameter file, CPU parameter file, and program file. Register items as favorites on the window opened by selecting [Setting] ⇨ [Register Favorites Selection] on the menu bar.

5.10 Resetting the Modules

Reset the modules on MELSOFT Mirror.

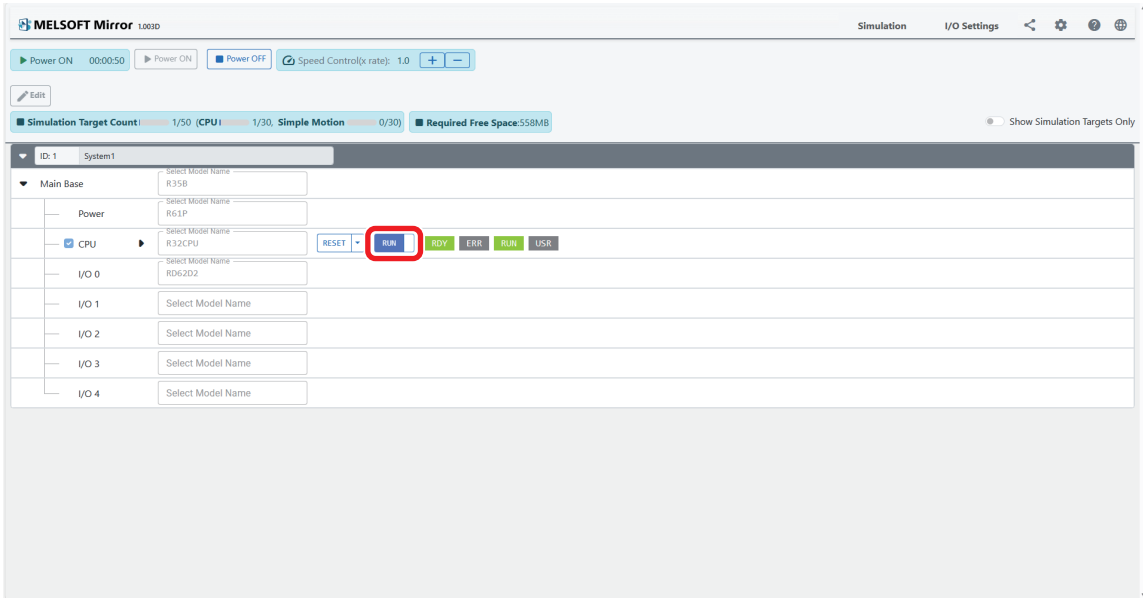
Procedure



1. Check that MELSOFT Mirror is powered on, and click the [RESET] button for the CPU module.

5.11 Executing the Program

Execute the program written to the programmable controller.



1. Set the switch of the CPU module to [RUN] on MELSOFT Mirror.

5.12 Monitoring the Program

Monitor the program operation using the engineering tool.

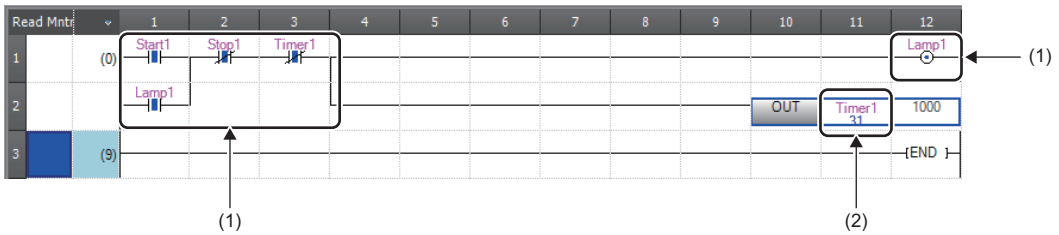
Monitoring on the monitor status bar

For the monitor status bar, refer to the following.

GX Works3 Operating Manual

Monitoring on the ladder editor

The on/off states of contacts and coils and the current values of word devices and labels can be monitored on the ladder editor.

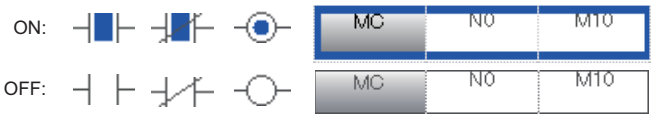


(1) The on/off states of the contacts and coils are displayed.
(2) The current value of the word/double word type data is displayed.

1. Select [Online] ⇨ [Monitor] ⇨ [Start Monitoring] on the menu bar.
2. Monitor the on/off states of contacts and coils and the current values of word devices and labels.

■On/off state display

The on/off states are displayed on the editor as follows.



■Changing the current value

To change the current value, select the cell on the ladder editor, and press Shift + double-click the cell or press Shift + Enter while the program is being monitored.

Point

The program can also be monitored on the Device/Buffer Memory Batch window or the Watch window. (GX Works3 Operating Manual)

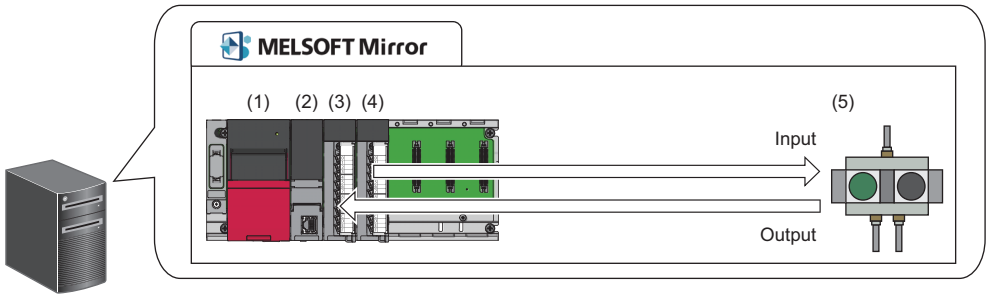
6 I/O MODULE SIMULATION

6.1 System Configuration Example

This section describes an example system configuration when input and output modules are used.

System configuration

- Main base unit: R35B
- Power supply module: R61P
- CPU module: R04CPU
- Input module: RX40C7
- Output module: RY40NT5P
- Valve with sensor: It is assumed that the sensor output is produced immediately when the valve opens in response to an input signal. (Valve dynamics are not considered.)

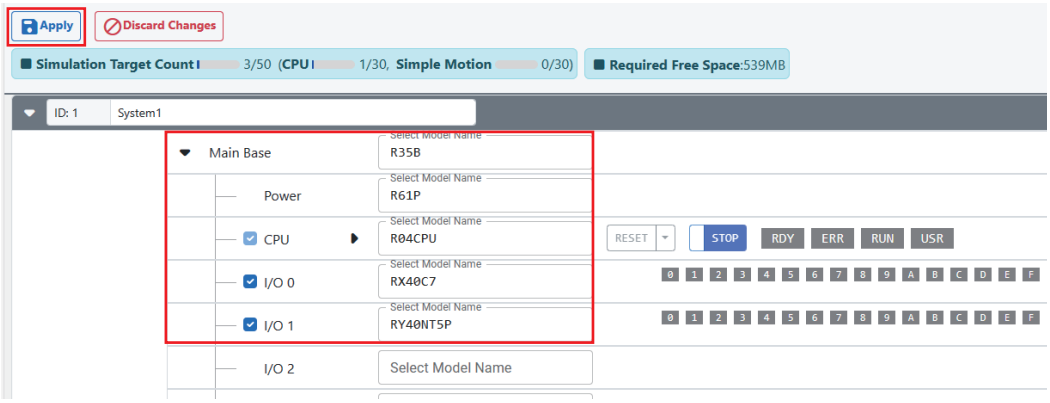


(1) Power supply module
(2) CPU module
(3) Input module
(4) Output module
(5) Value with sensor

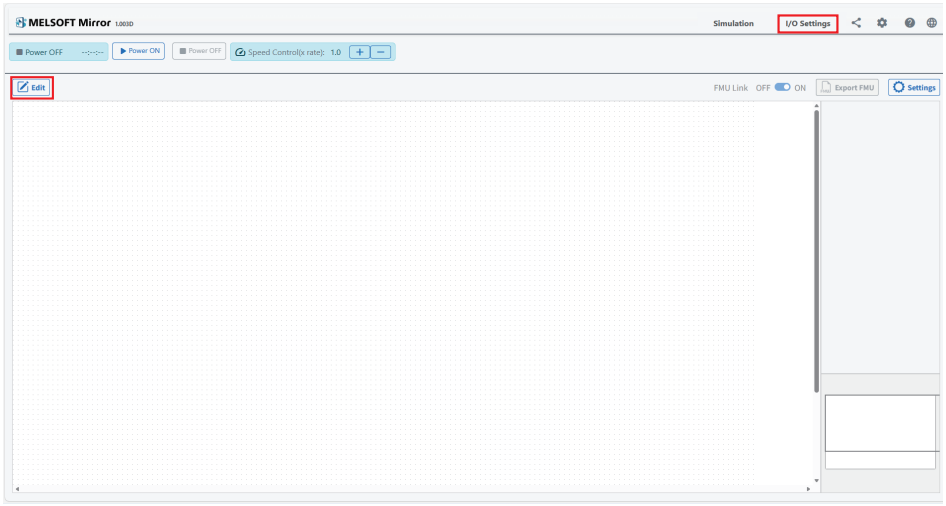
6.2 Setting System Configuration

Set the programmable controller system configuration on MELSOFT Mirror.

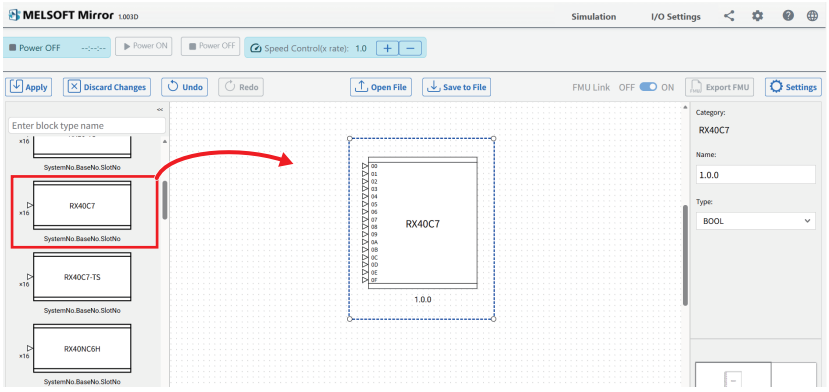
1. Click the [Edit] button.
2. Configure the following settings in the simulation setting window, and click the [Apply] button.



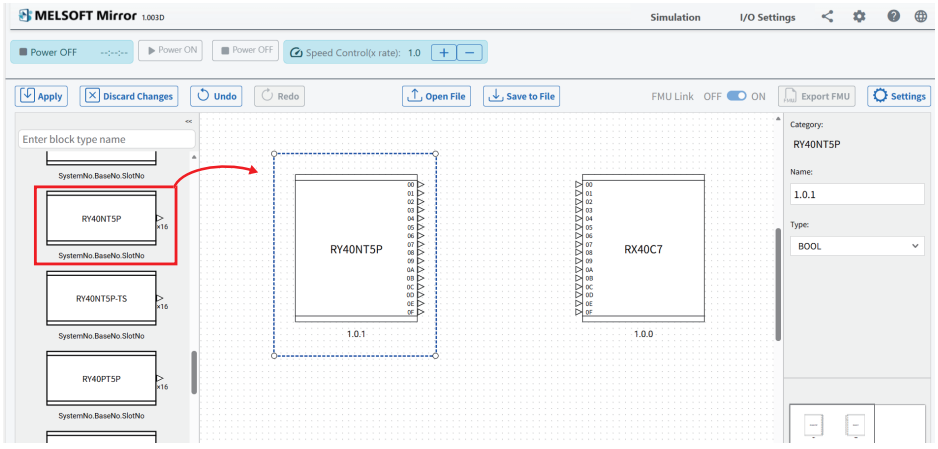
3. Click "I/O Settings" to open the "I/O Settings" window, and click the [Edit] button.



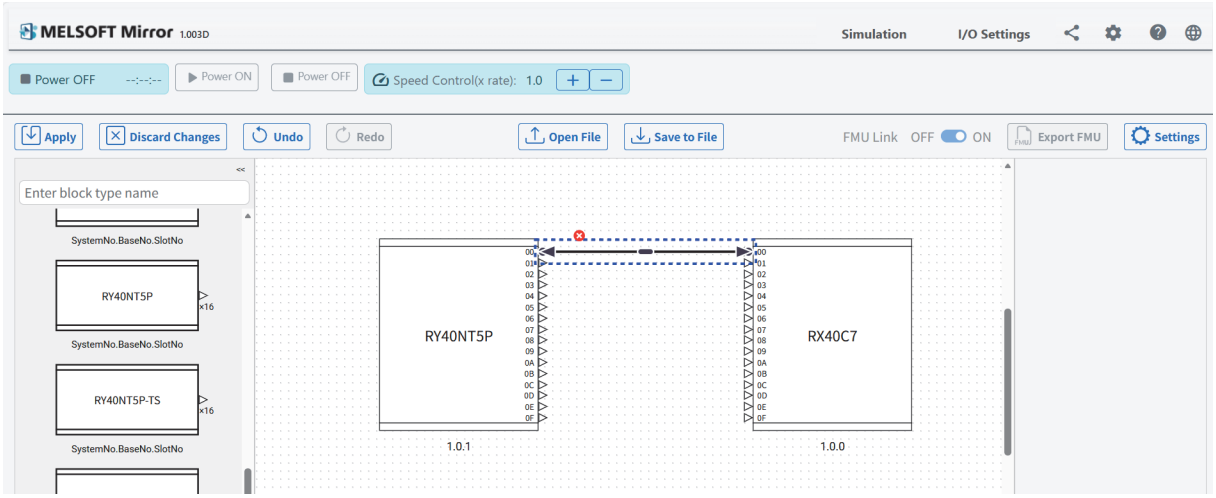
4. Select "RX40C7" from the block list on the left, and drag and drop it to the block diagram in the center. Set the block name to "1.0.0".



5. Select "RY40NT5P" from the block list on the left, and drag and drop it to the block diagram in the center. Set the block name to "1.0.1".



6. Use drag and drop to draw a line from the terminal 00 of the RY40NT5P to the terminal 00 of the RX40C7. When a signal is output from the terminal 00 (Y10) of the RY40NT5P, the signal is input to the terminal 00 (X0) of the RX40C7 after 222μs (execution cycle configured in the I/O settings).

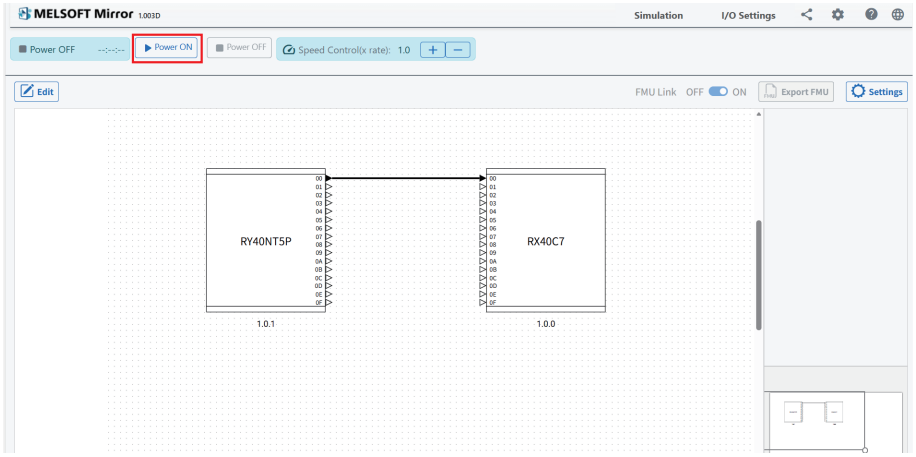


Point

When the CPU module outputs a signal to the RY40NT5P, pay attention to operational differences between Y and DY devices. (The same caution applies to X and DX devices.) For details, refer to the following.
MELSEC iQ-R Programmable Controller CPU Module User's Manual

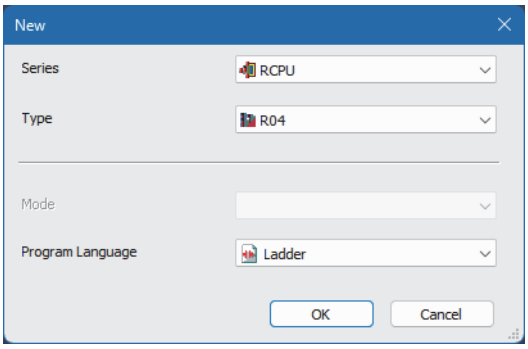
7. Click the [Apply] button.

8. Click the [Power ON] button.

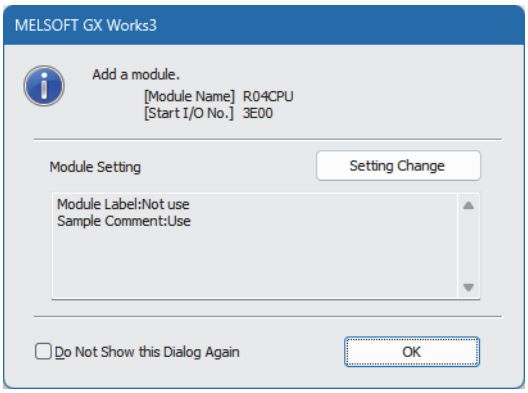


6.3 Creating a Project with the Engineering Tool and Writing Data to MELSOFT Mirror

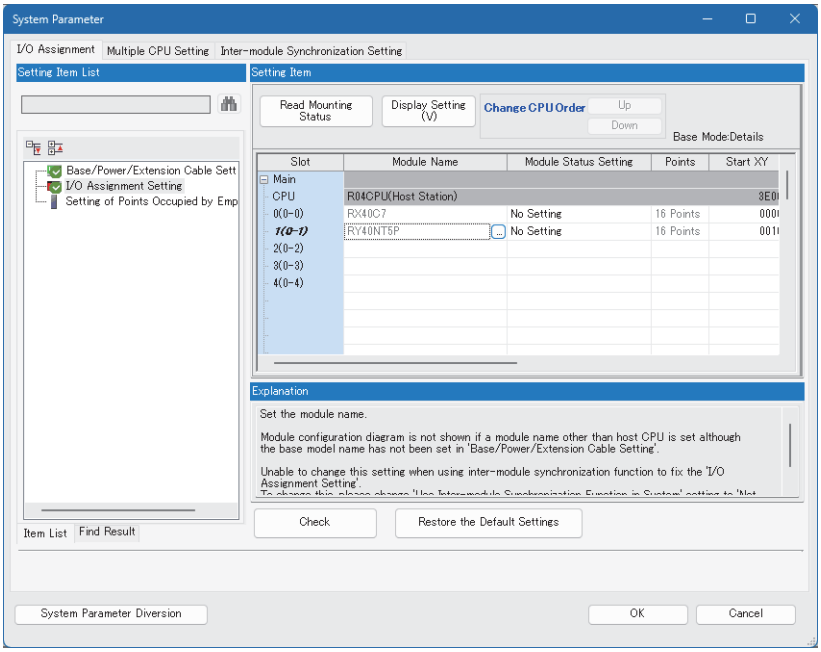
1. Create a project using the engineering tool. Set the following in the window for creating a new project.



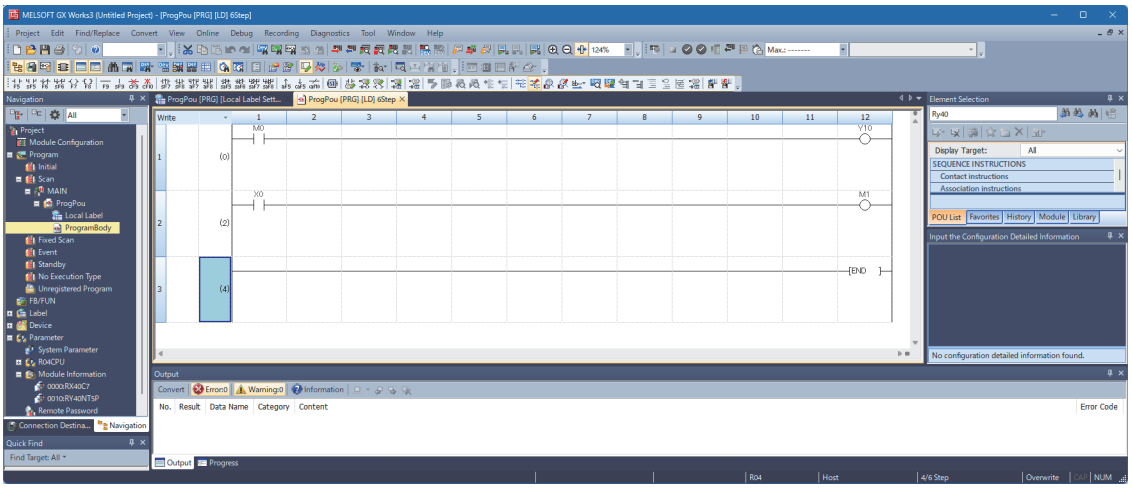
2. When the window for adding a module appears, click the [OK] button.



3. Place the following modules in the "System Parameter" window.



4. Write contacts and coils, as shown below, on the ladder program.



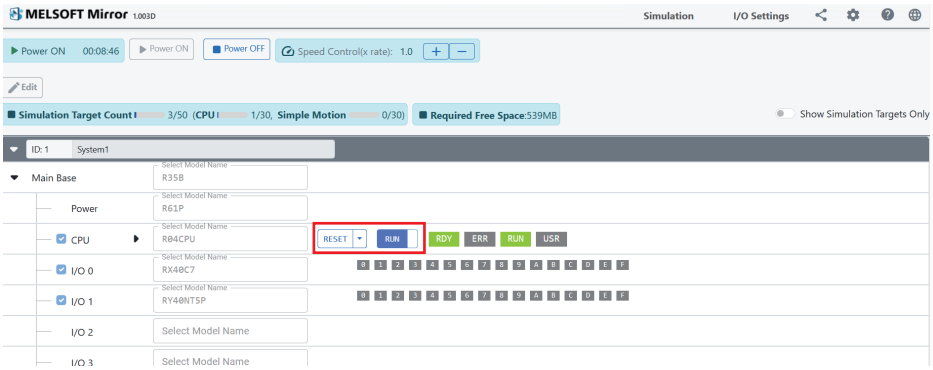
5. Select [Convert] ⇨ [Convert] on the menu bar to determine the ladder blocks.

6. Connect the engineering tool to MELSOFT Mirror. ☞ Page 61 Connecting the Engineering Tool to MELSOFT Mirror

7. Write the parameters and program to the R04CPU. ☞ Page 62 Writing Data to the Programmable Controller

6.4 Executing the Program

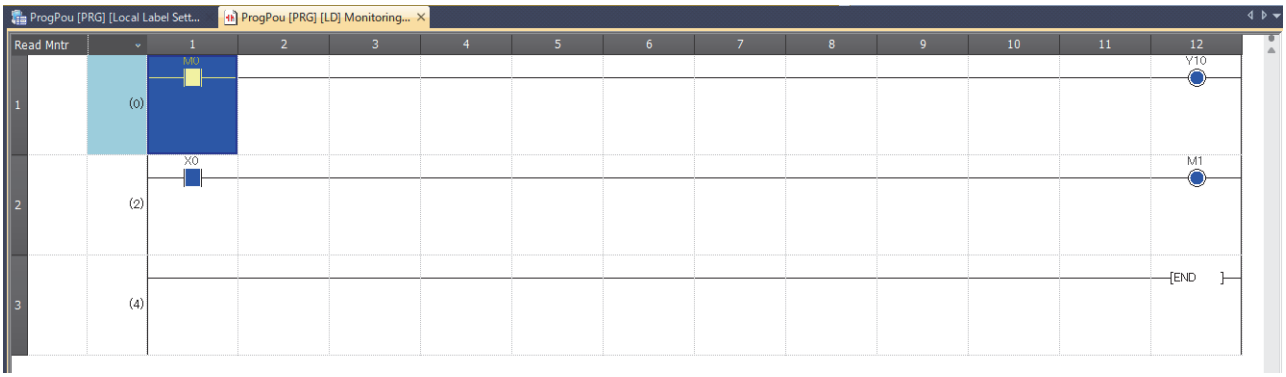
1. Reset the R04CPU, and set the switch to RUN on MELSOFT Mirror.



2. The engineering tool starts the program monitor. ☞ Page 64 Monitoring the Program

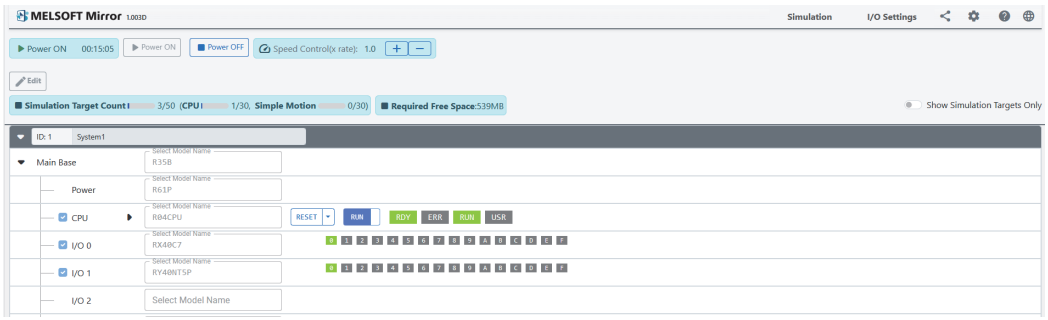
3. The program is executed as follows according to the ladder logic:

- When M0 turns on, Y10 turns on.
- When Y10 turns on and connection was made in the I/O settings, X0 turns on.
- When X0 turns on, M1 turns on.



Point

The states of X0 (terminal 00 of the RX40C7) and Y10 (terminal 00 of the RY40NT5P) can be checked using the LEDs in the MELSOFT Mirror simulation setting window.



7 MULTIPLE CPU SYSTEM SIMULATION

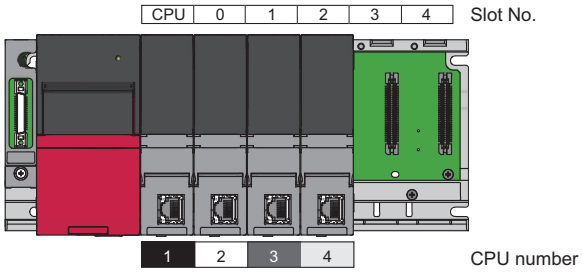
This chapter describes the procedure for simulating a multiple CPU system with MELSOFT Mirror.

7.1 System Configuration Example

The following system configuration is used for explanation.

System configuration

- Main base unit: R35B
- Power supply module: R61P
- CPU module No.1: R32CPU
- CPU module No.2: R32CPU
- CPU module No.3: R32CPU
- CPU module No.4: R32CPU



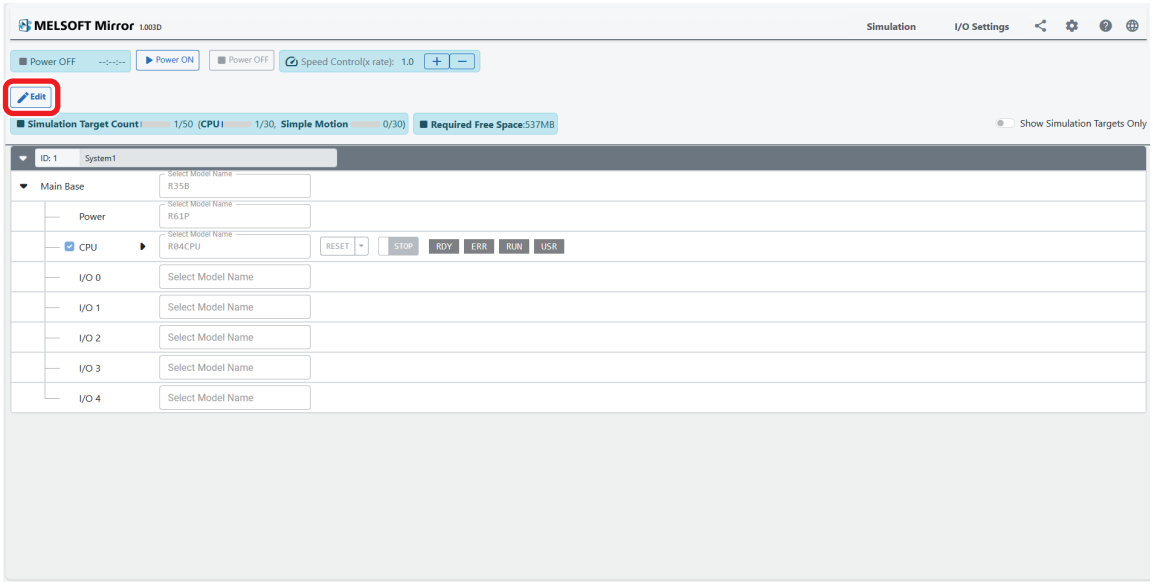
7.2 Setting System Configuration

Set the programmable controller system configuration on MELSOFT Mirror.

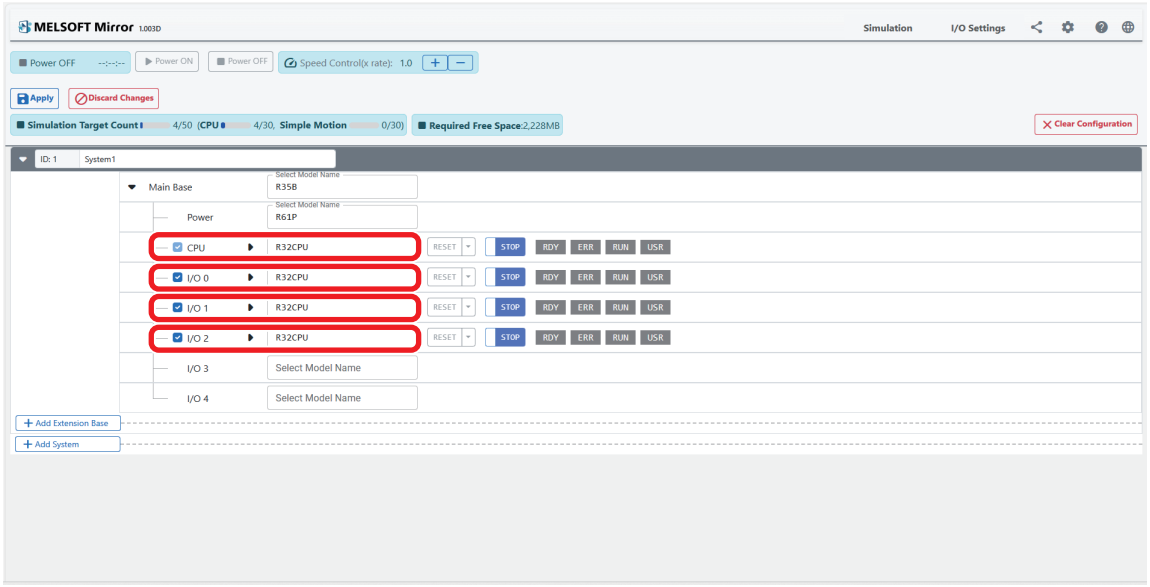
Point

To simulate modules, the parameters need to be set for the modules. The parameters of the RD77MS and network modules are written via the control CPU. Enable the module operation setting of the control CPU. Or, disable the module operation setting of the control CPU, which controls these modules but is not supported by MELSOFT Mirror.

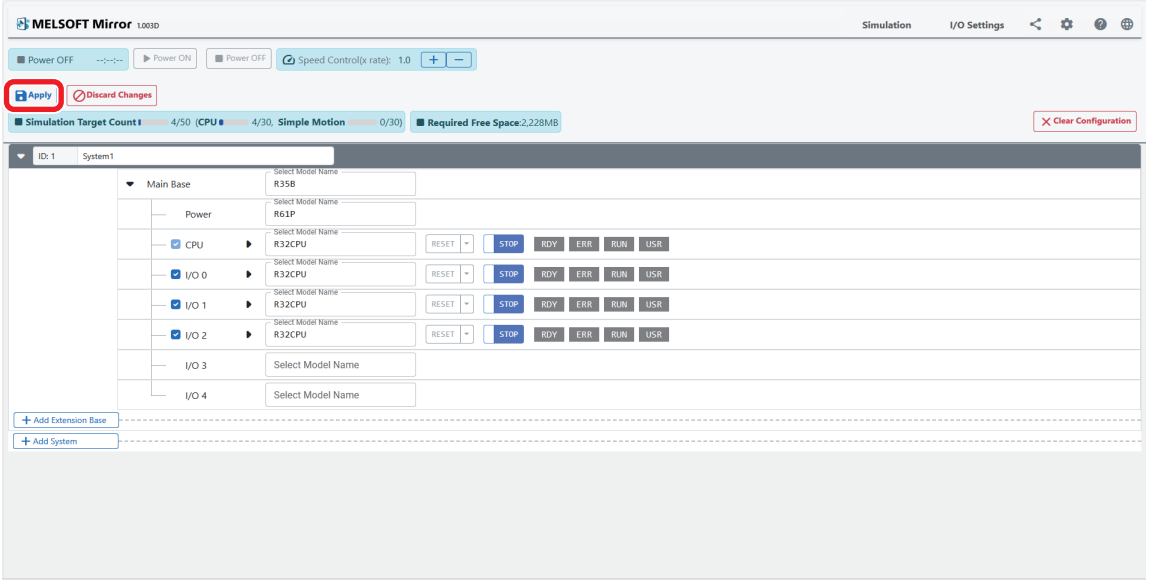
1. Click the [Edit] button.



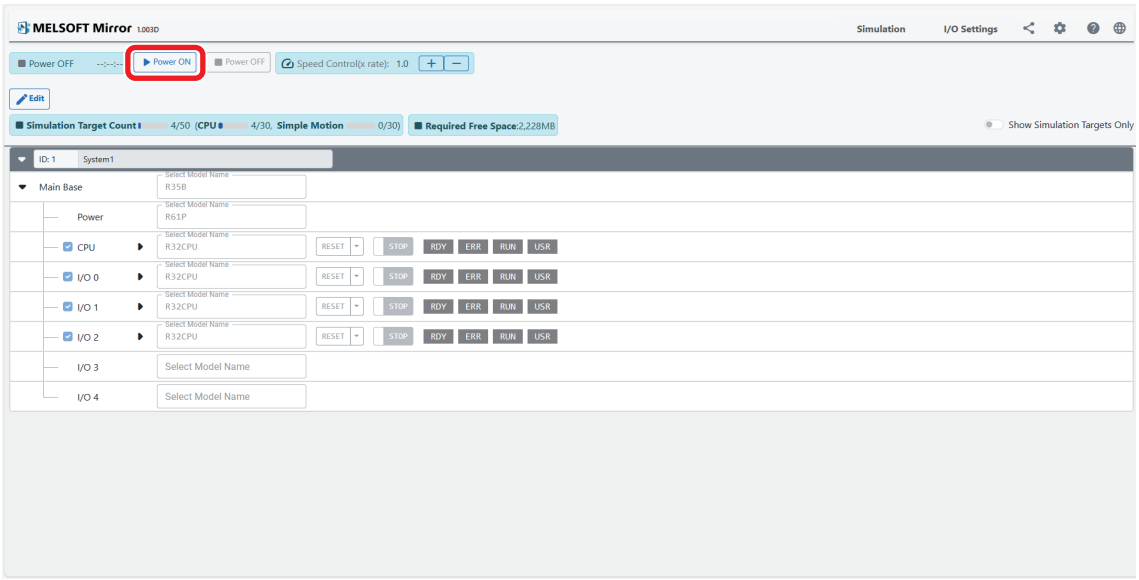
2. Edit the configuration of the modules.



3. Select [R32CPU] from the "CPU" combo box.
4. Select [R32CPU] from the "I/O0" combo box, and check that the "I/O0" checkbox is selected.
5. Select [R32CPU] from the "I/O1" combo box, and check that the "I/O1" checkbox is selected.
6. Select [R32CPU] from the "I/O2" combo box, and check that the "I/O2" checkbox is selected.
7. Click the [Apply] button.



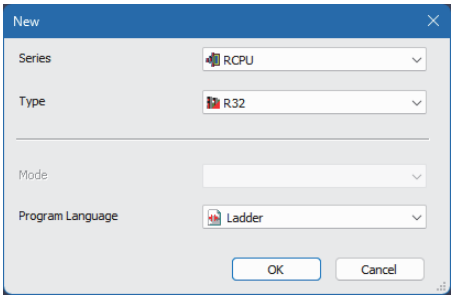
8. Click the [Power ON] button to start the simulation.



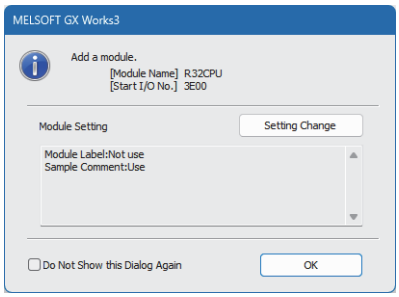
7.3 Creating a Project with the Engineering Tool and Writing Data to MELSOFT Mirror

In a multiple CPU system, create projects for the number of CPU modules to be used, and write a project to each CPU module.

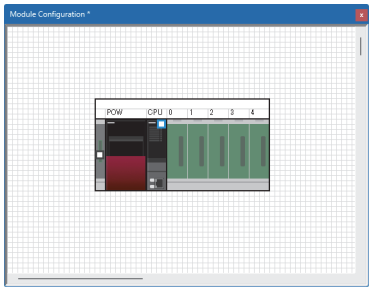
Procedure for creating a project and writing it to the CPU No.1



1. Create a new project. Select "RCPU" for "Series" and the CPU module model to be used for "Type". Select "Ladder" for "Program Language", and click the [OK] button.

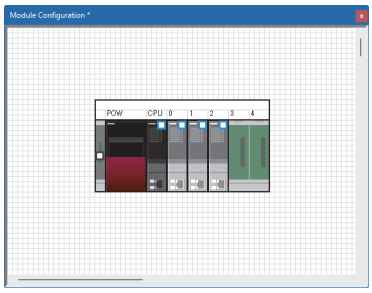


2. When the window for adding a module appears, click the [OK] button.

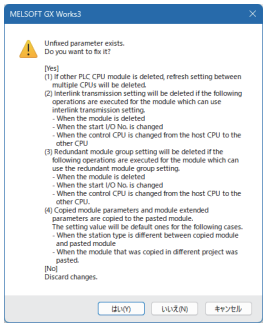


3. In the Module Configuration window, place the CPU No.1 in the CPU slot.

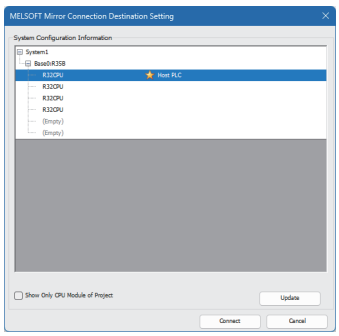
[Navigation window] ⇨ [Module Configuration]



4. From the Element Selection window, select the CPU No.2, and place it in the I/O0 slot.
5. From the Element Selection window, select the CPU No.3, and place it in the I/O1 slot.
6. From the Element Selection window, select the CPU No.4, and place it in the I/O2 slot.



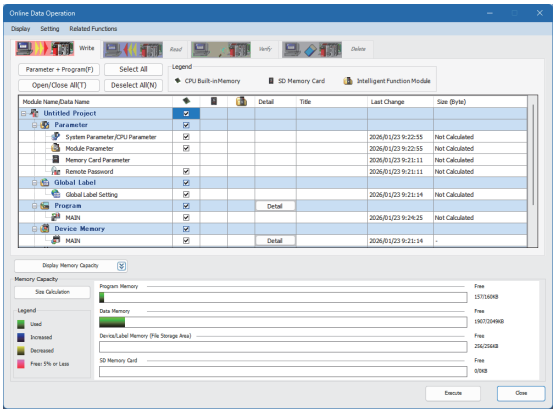
7. Close the Module Configuration window. When a warning window appears, click the [Yes] button to confirm the parameters.



8. Select [Debug] ⇨ [Simulation] ⇨ [MELSOFT Mirror] ⇨ [Connect Simulation] on the menu bar of the engineering tool.

9. Enter the URL on the top window of MELSOFT Mirror.

10. Select the CPU module displayed as "Host Station", and click the [Connect] button.



11. Write the set parameters and the created program to the CPU No.1.

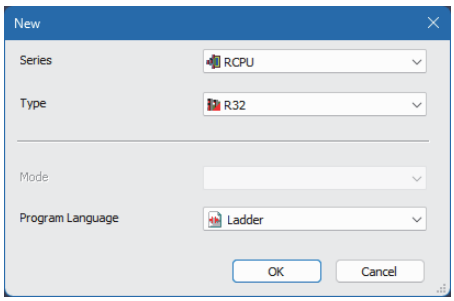
[Online] ⇨ [Write to PLC]

12. Select the system parameter file, CPU parameter file, module parameter file, and program file on the "Online Data Operation" window.

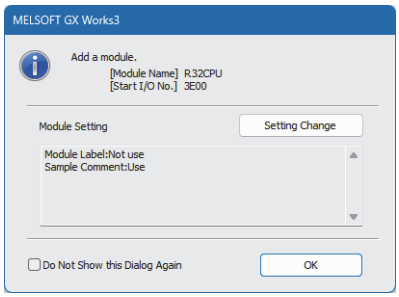
13. Click the [Execute] button.

14. After the write processing completes, click the [Close] button.

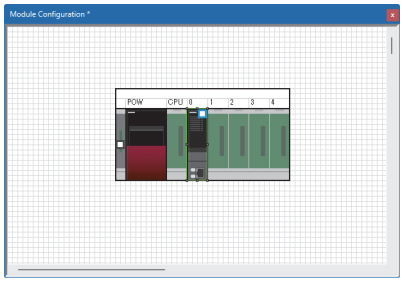
Procedure for creating a project and writing it to the CPU No.2



1. Create a new project. Select "RCPU" for "Series" and the CPU module model to be used for "Type". Select "Ladder" for "Program Language", and click the [OK] button.

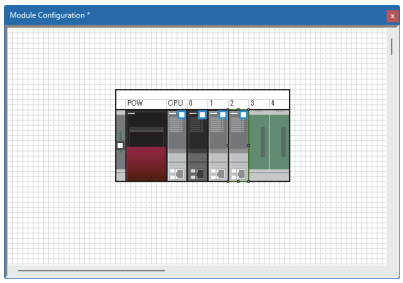


2. When the window for adding a module appears, click the [OK] button.



3. In the Module Configuration window, place the CPU No.2 in the I/O0 slot.

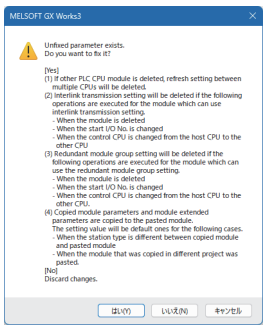
[Navigation window] ⇨ [Module Configuration]



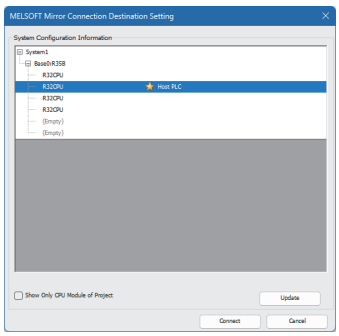
4. From the Element Selection window, select the CPU No.1, and place it in the CPU slot.

5. From the Element Selection window, select the CPU No.3, and place it in the I/O1 slot.

6. From the Element Selection window, select the CPU No.4, and place it in the I/O2 slot.



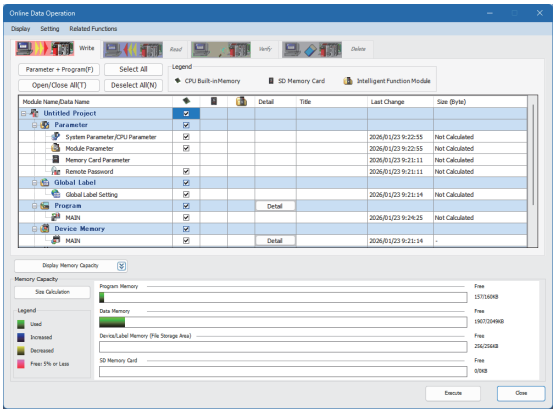
7. Close the Module Configuration window. When a warning window appears, click the [Yes] button to confirm the parameters.



8. Select [Debug] ⇨ [Simulation] ⇨ [MELSOFT Mirror] ⇨ [Connect Simulation] on the menu bar of the engineering tool.

9. Enter the URL on the top window of MELSOFT Mirror.

10. Select the CPU module displayed as "Host Station", and click the [Connect] button.



11. Write the set parameters and the created program to the CPU No.2.

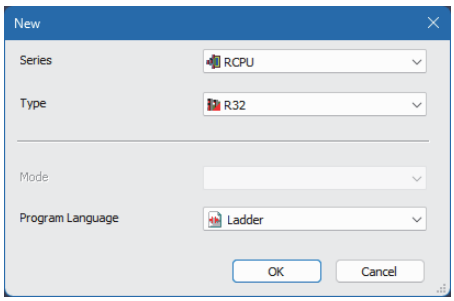
[Online] ⇨ [Write to PLC]

12. Select the system parameter file, CPU parameter file, module parameter file, and program file on the "Online Data Operation" window.

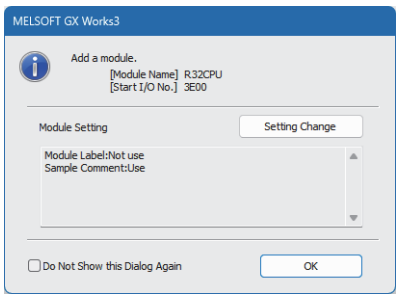
13. Click the [Execute] button.

14. After the write processing completes, click the [Close] button.

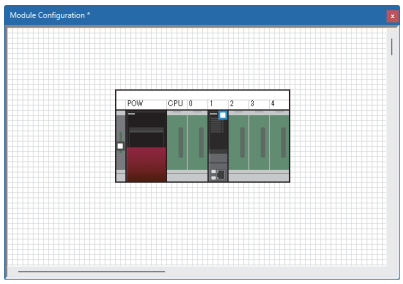
Procedure for creating a project and writing it to the CPU No.3



1. Create a new project. Select "RCPU" for "Series" and the CPU module model to be used for "Type". Select "Ladder" for "Program Language", and click the [OK] button.

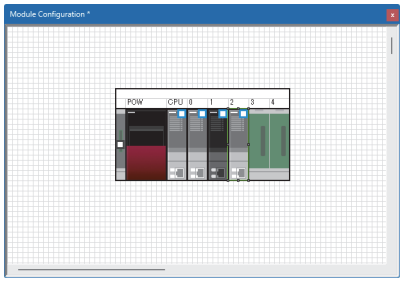


2. When the window for adding a module appears, click the [OK] button.



3. In the Module Configuration window, place the CPU No.3 in the I/O1 slot.

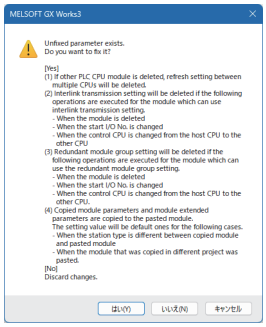
[Navigation window] ⇨ [Module Configuration]



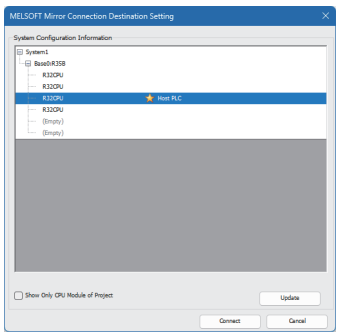
4. From the Element Selection window, select the CPU No.1, and place it in the CPU slot.

5. From the Element Selection window, select the CPU No.2, and place it in the I/O0 slot.

6. From the Element Selection window, select the CPU No.4, and place it in the I/O2 slot.



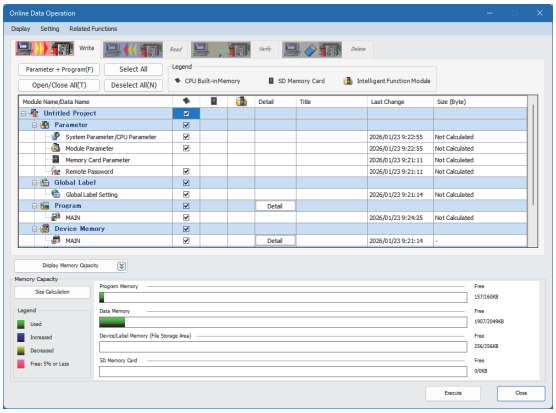
7. Close the Module Configuration window. When a warning window appears, click the [Yes] button to confirm the parameters.



8. Select [Debug] ⇨ [Simulation] ⇨ [MELSOFT Mirror] ⇨ [Connect Simulation] on the menu bar of the engineering tool.

9. Enter the URL on the top window of MELSOFT Mirror.

10. Select the CPU module displayed as "Host Station", and click the [Connect] button.



11. Write the set parameters and the created program to the CPU No.3.

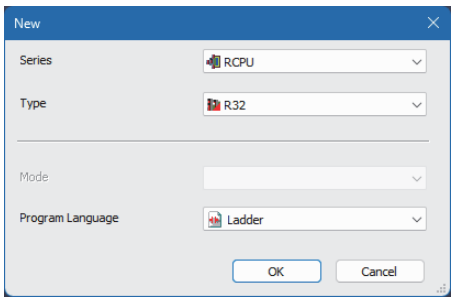
[Online] ⇨ [Write to PLC]

12. Select the system parameter file, CPU parameter file, module parameter file, and program file on the "Online Data Operation" window.

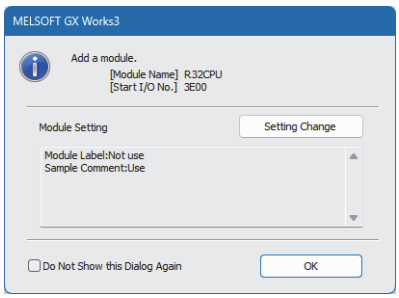
13. Click the [Execute] button.

14. After the write processing completes, click the [Close] button.

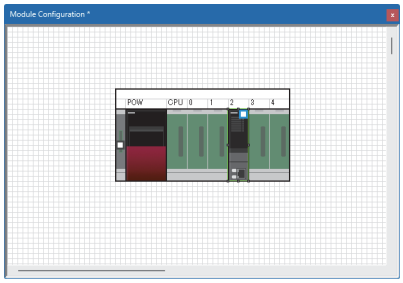
Procedure for creating a project and writing it to the CPU No.4



1. Create a new project. Select "RCPU" for "Series" and the CPU module model to be used for "Type". Select "Ladder" for "Program Language", and click the [OK] button.

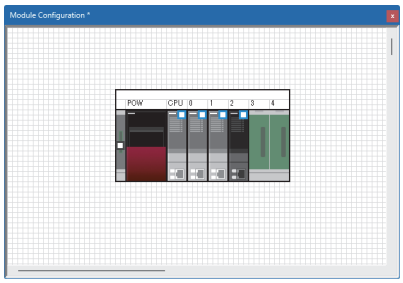


2. When the window for adding a module appears, click the [OK] button.



3. In the Module Configuration window, place the CPU No.4 in the I/O2 slot.

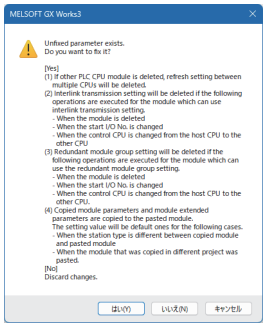
[Navigation window] ⇨ [Module Configuration]



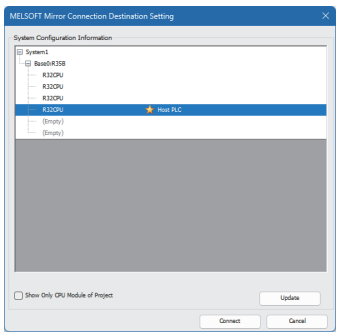
4. From the Element Selection window, select the CPU No.1, and place it in the CPU slot.

5. From the Element Selection window, select the CPU No.2, and place it in the I/O0 slot.

6. From the Element Selection window, select the CPU No.3, and place it in the I/O1 slot.



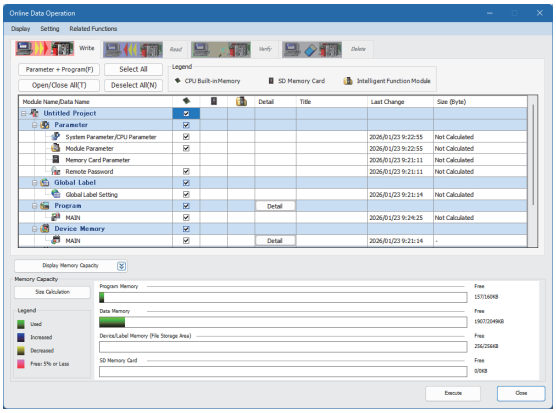
7. Close the Module Configuration window. When a warning window appears, click the [Yes] button to confirm the parameters.



8. Select [Debug] ⇨ [Simulation] ⇨ [MELSOFT Mirror] ⇨ [Connect Simulation] on the menu bar of the engineering tool.

9. Enter the URL on the top window of MELSOFT Mirror.

10. Select the CPU module displayed as "Host Station", and click the [Connect] button.



11. Write the set parameters and the created program to the CPU No.4.

[Online] ⇨ [Write to PLC]

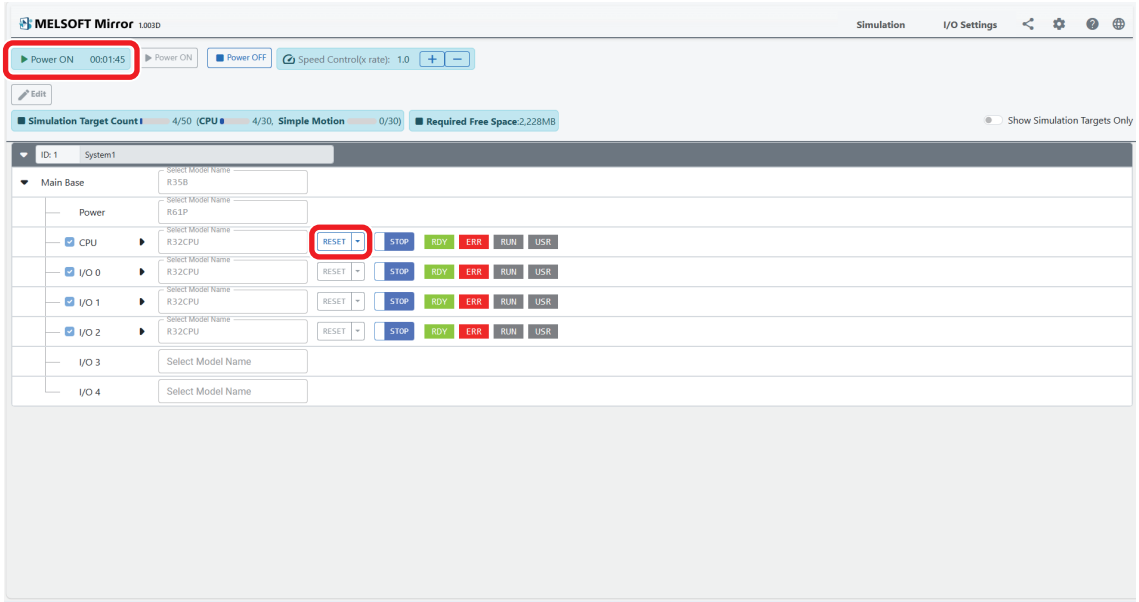
12. Select the system parameter file, CPU parameter file, module parameter file, and program file on the "Online Data Operation" window.

13. Click the [Execute] button.

14. After the write processing completes, click the [Close] button.

Resetting the modules

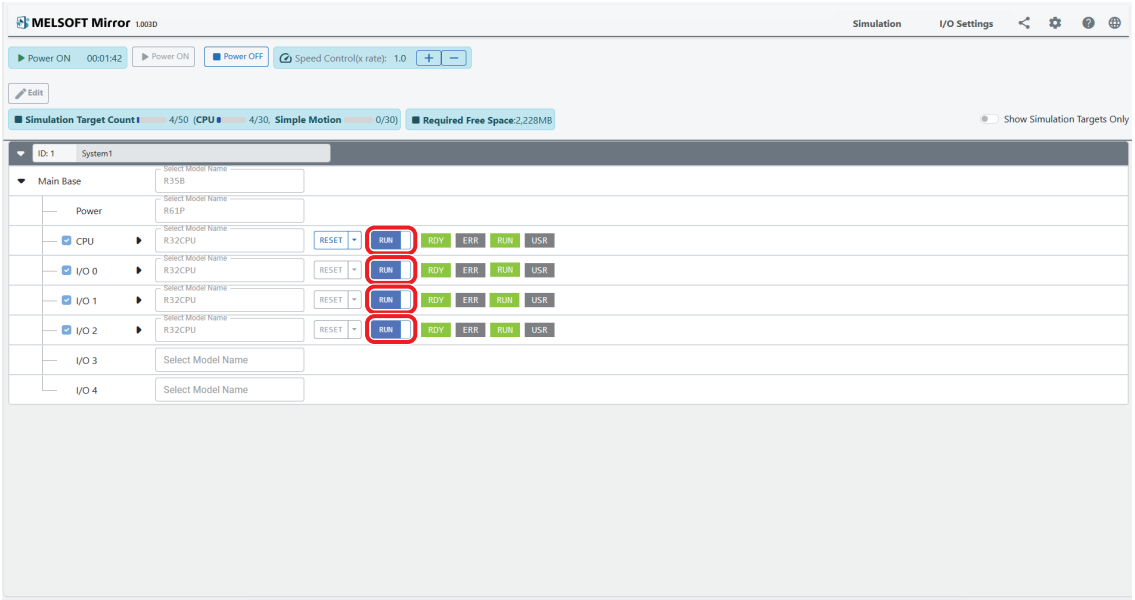
Reset the modules on MELSOFT Mirror.



1. Check that MELSOFT Mirror is powered on, and click the [RESET] button for the CPU module.

Executing the program

Execute the program written to the programmable controller.



1. Set the switch of the CPU module to [RUN] on MELSOFT Mirror.

7.4 Setting System Parameters

For details on setting the system parameters of a multiple CPU system, refer to the following.

MELSEC iQ-R CPU Module User's Manual (Application)

8 NETWORK SIMULATION

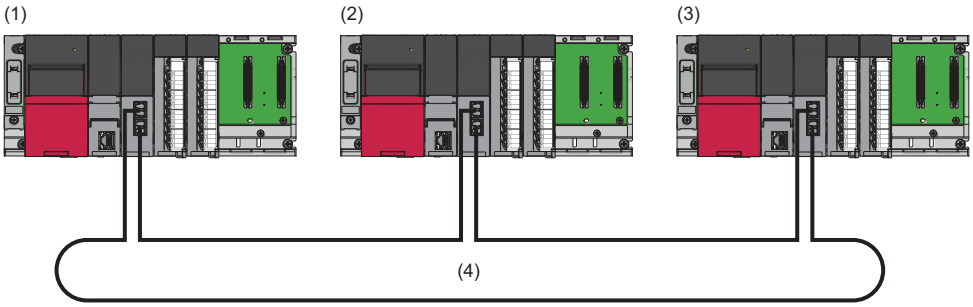
This chapter describes the procedure for simulating cyclic transmission in a system with one control station and two normal stations.

8.1 System Configuration Example

The following system configuration is used for explanation.

System configuration

- Main base unit: R35B
- Power supply module: R61P
- CPU module: R04CPU
- RJ71GP21-SX CC-Link IE Controller Network module: RJ71GP21-SX (start I/O number: 0000H to 001FH)
- Input module: RX10 (start I/O number: 0020H to 002FH)
- Output module: RY10R2 (start I/O number: 0030H to 003FH)



(1) Control station (No.1)
(2) Normal station (No.2)
(3) Normal station (No.3)
(4) Network number 1

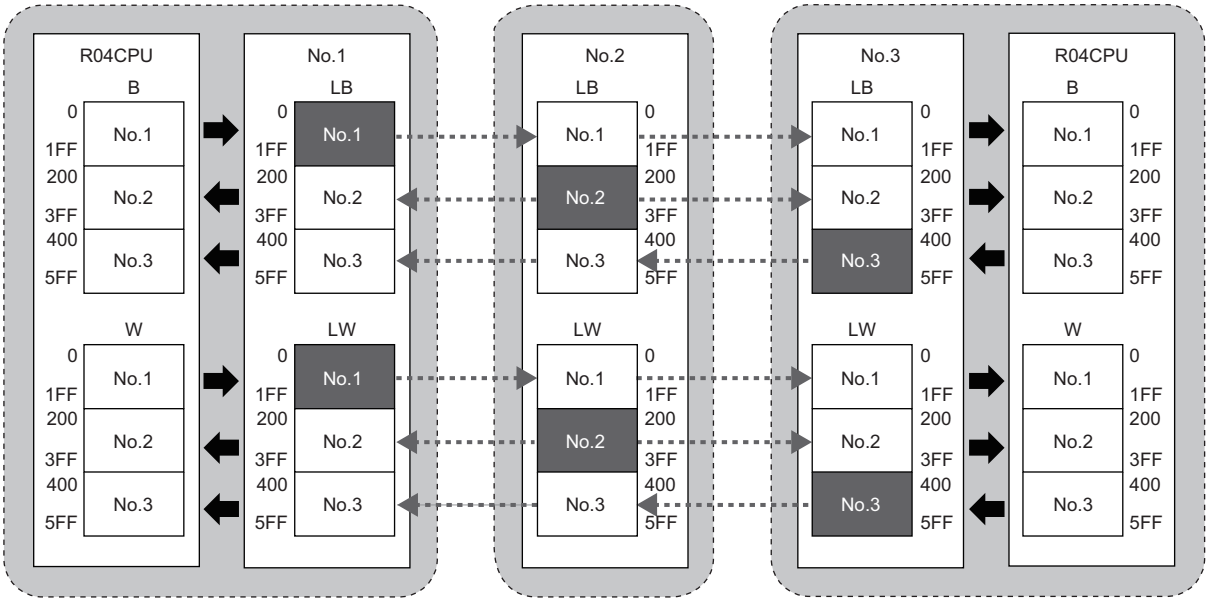
Link device assignment

In this example, 512 points are assigned to each station.

■LB/LW assignment

The numbers in the figure (No.1 to No.3) indicate station numbers.

No.1 is a control station, and No.2 and No.3 are normal stations.

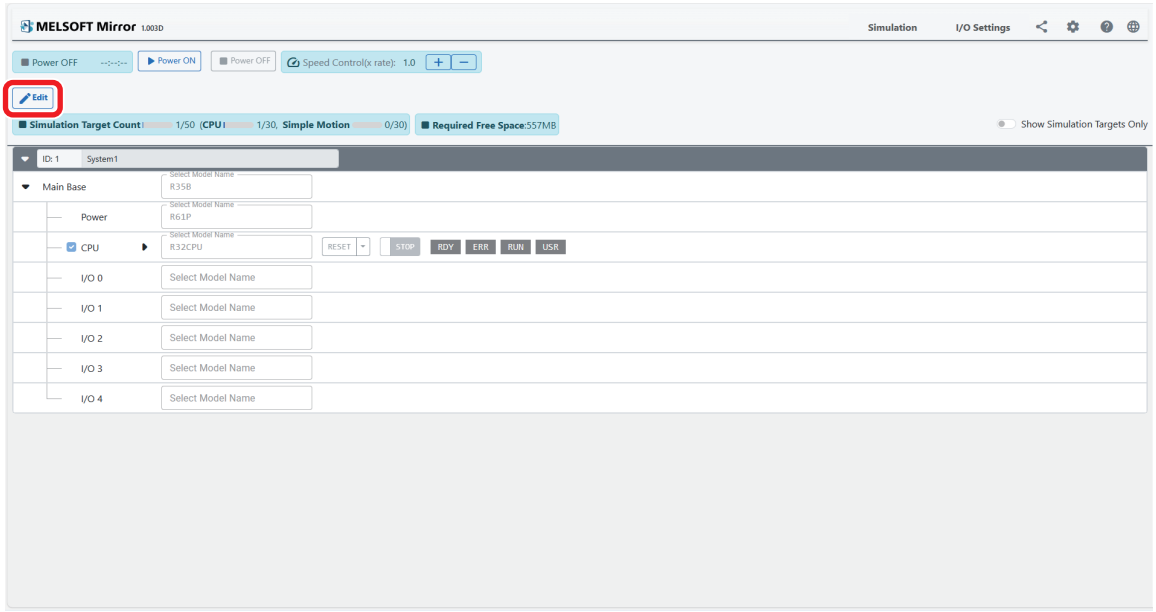


8

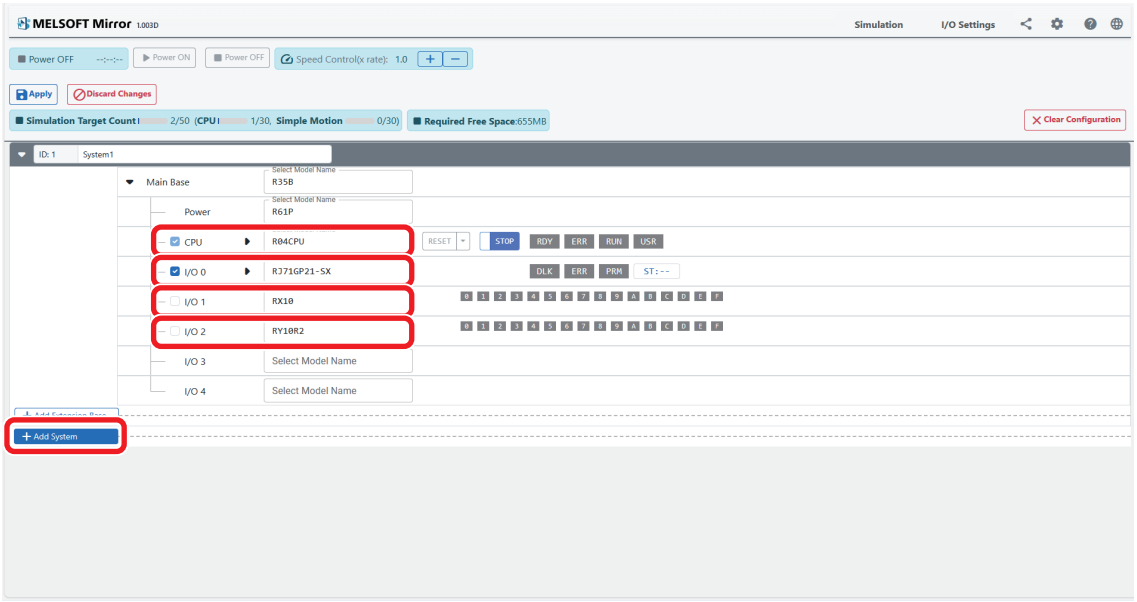
8.2 Setting System Configuration

Set the programmable controller system configuration on MELSOFT Mirror.

1. Click the [Edit] button.



2. Edit the configuration of the modules.



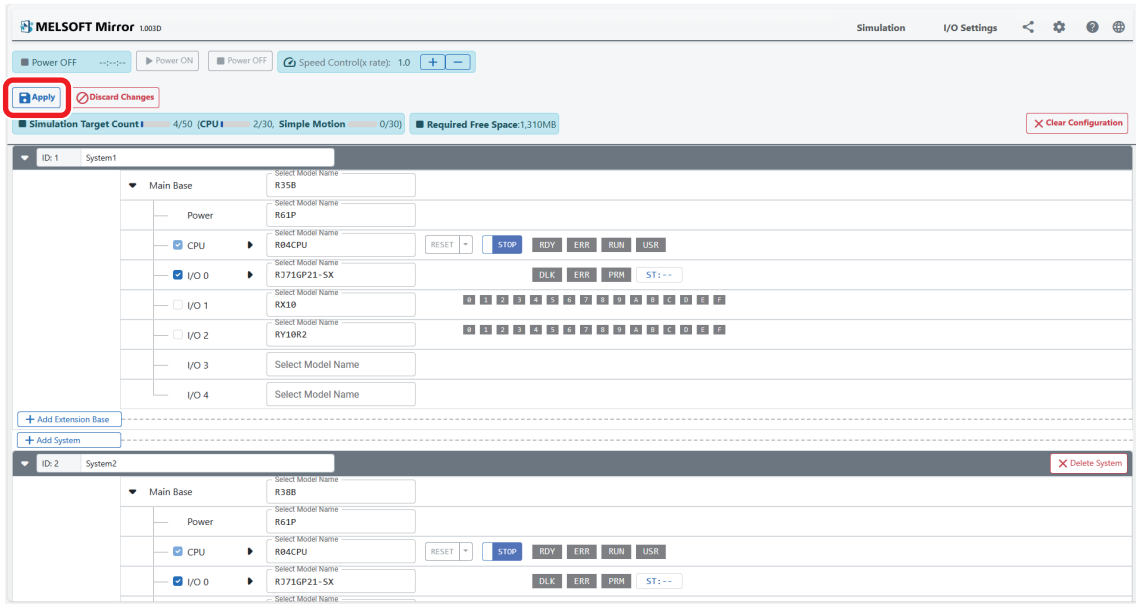
3. Select [R04CPU] from the "CPU" combo box.
4. Select [RJ71GP21-SX] from the "I/O0" combo box, and check that the "I/O0" checkbox is selected.
5. Select [RX10] from the "I/O1" combo box.
6. Select [RY10R2] from the "I/O2" combo box.
7. Click the [Add System] button to add a main base unit.
8. Repeat the steps for the main base unit of [System1].
9. Click the [Add System] button to add a main base unit.
10. Repeat the steps for the main base unit of [System1].

Point

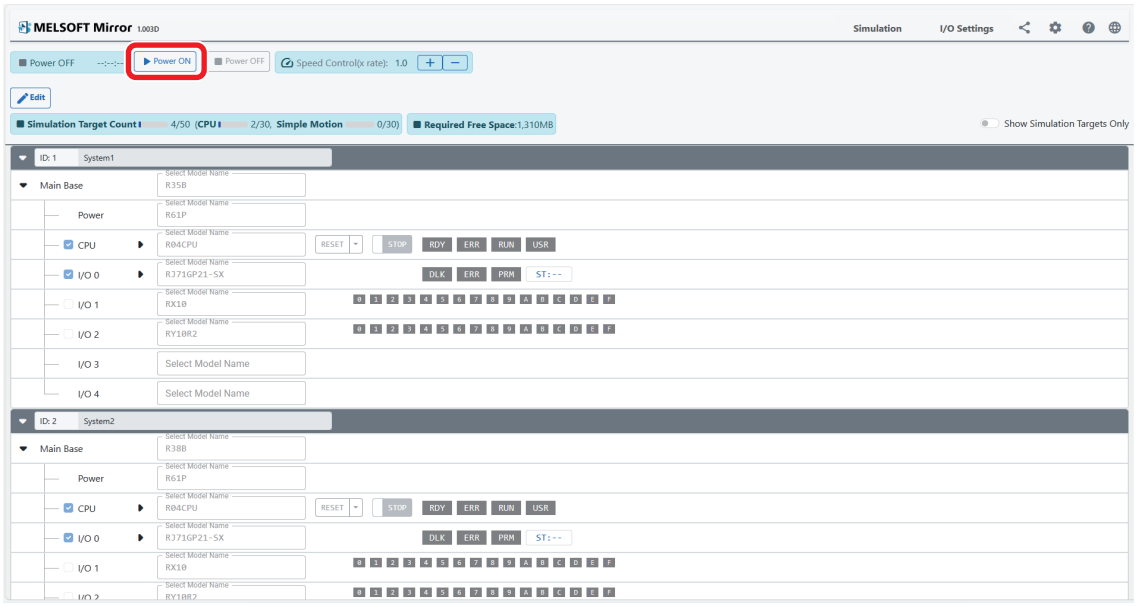
To add a main base unit, click the [Add System] button.

This system configuration uses three main base units, so click the [Add System] button twice.

11. Click the [Apply] button.

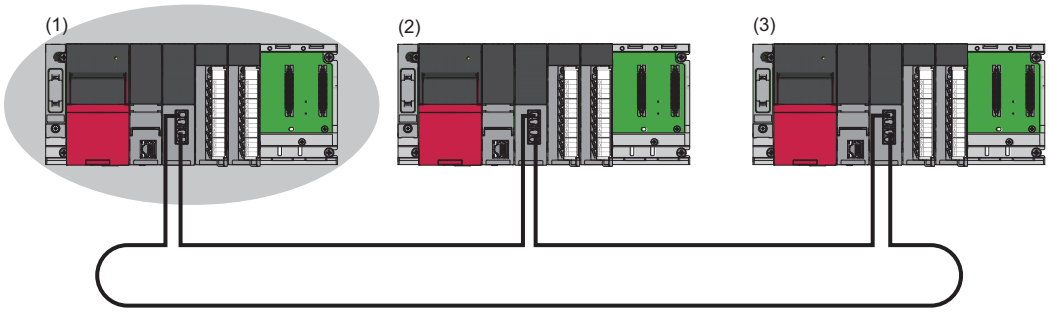


12. Click the [Power ON] button to start the simulation.



8.3 Setting the Control Station

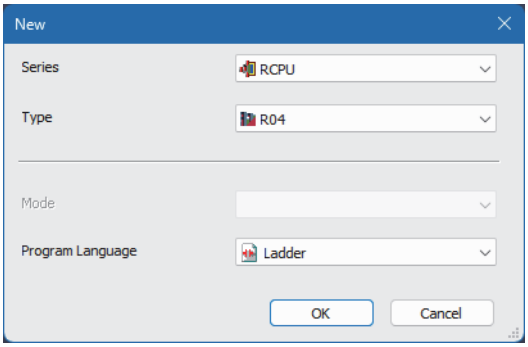
Connect the engineering tool to the CPU module on the control station, and set parameters.



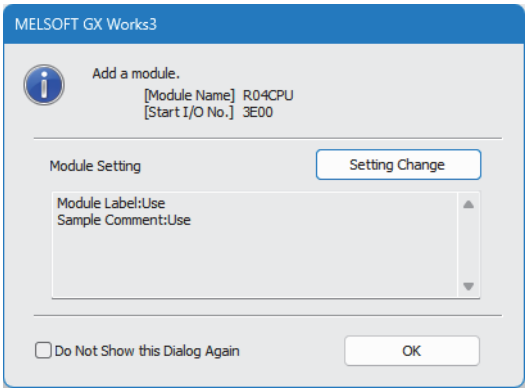
(1) Control station (No.1)
(2) Normal station (No.2)
(3) Normal station (No.3)

1. Set the CPU module as follows.

[Project] ⇨ [New]



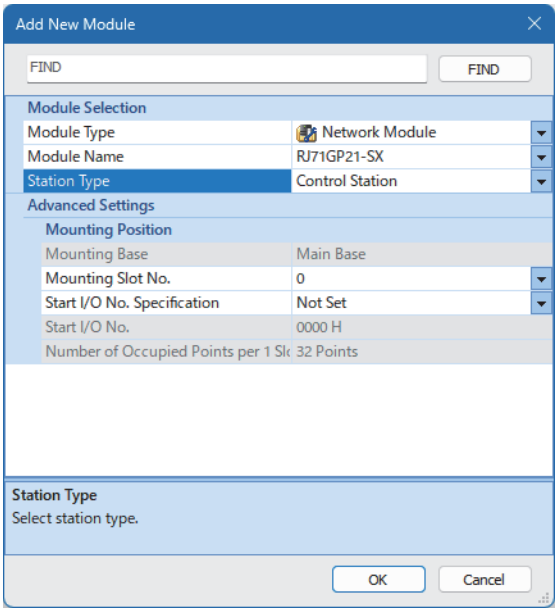
2. Click the [Setting Change] button in the following window, and set the module labels to be used.



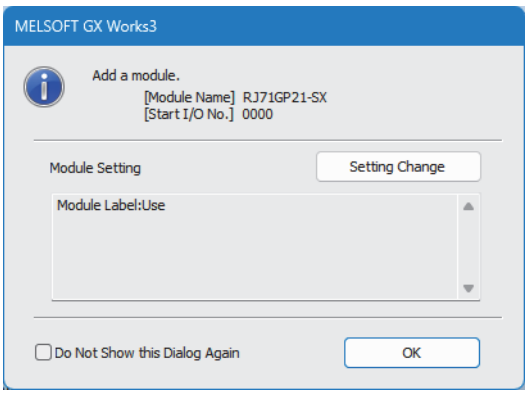
3. Click the [OK] button to add the module labels of the CPU module.

4. Set the CC-Link IE Controller Network-equipped module as follows.

[Navigation window] ⇨ [Parameter] ⇨ [Module Information] ⇨ Right-click ⇨ [Add New Module]

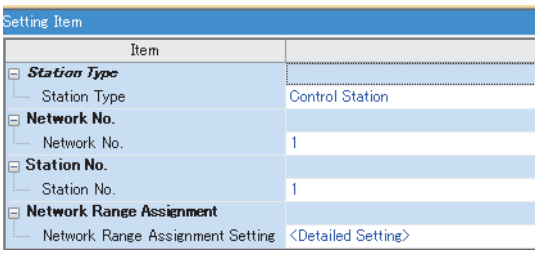


5. Click the [OK] button to add the module labels of the CC-Link IE Controller Network-equipped module.



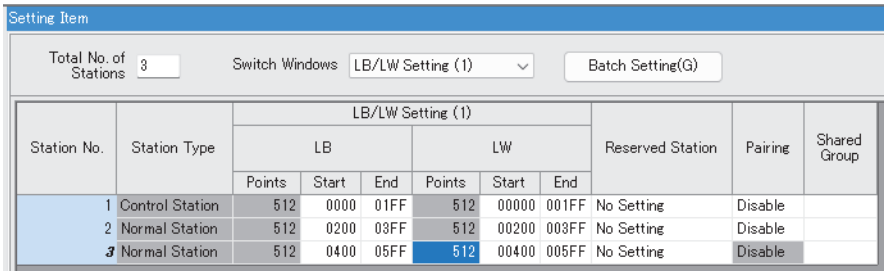
6. Set the parameters of "Required Settings" as follows.

[Navigation window] ⇨ [Parameter] ⇨ [Module Information] ⇨ [RJ71GP21-SX] ⇨ [Required Settings]



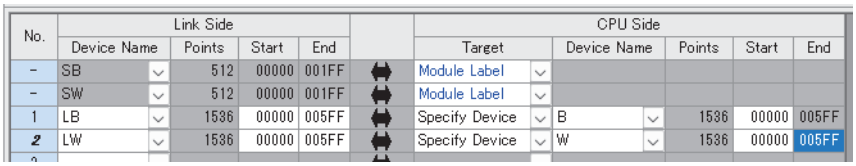
7. Set the network range assignment as follows.

[Navigation window] ⇨ [Parameter] ⇨ [Module Information] ⇨ [RJ71GP21-SX] ⇨ [Required Settings] ⇨ [Network Range Assignment]



8. Set the refresh settings as follows.

[Navigation window] ⇨ [Parameter] ⇨ [Module Information] ⇨ [RJ71GP21-SX] ⇨ [Basic Settings] ⇨ [Refresh Setting]



9. Connect the engineering tool to MELSOFT Mirror.

[Debug] ⇨ [Simulation] ⇨ [MELSOFT Mirror] ⇨ [Connect Simulation]

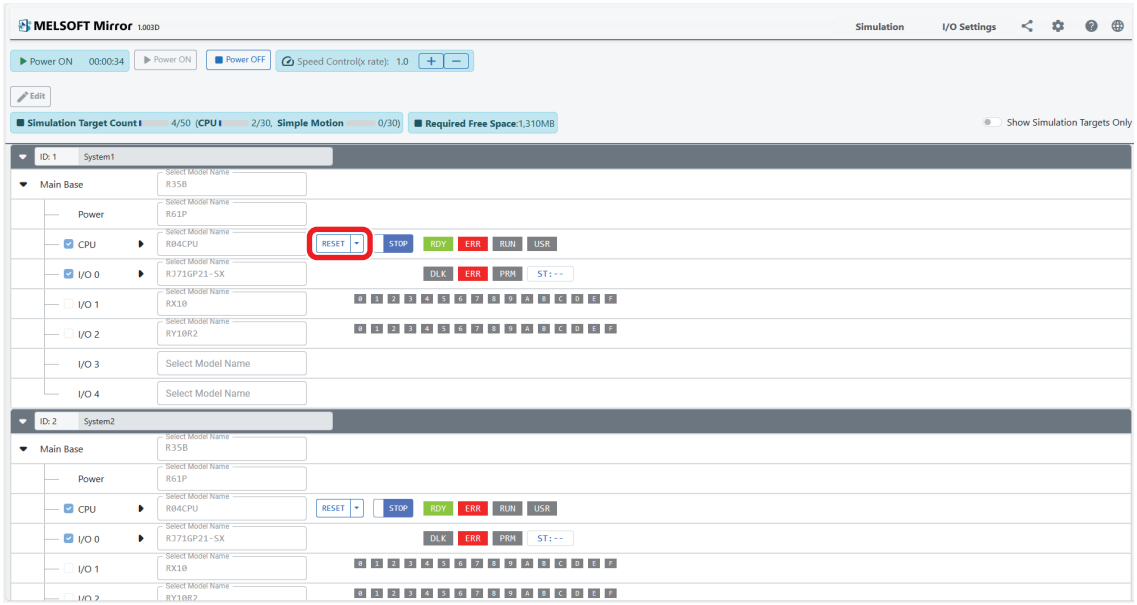
Double-click [System1] ⇨ [Base0:R35B] ⇨ [R04CPU] displayed in "System Configuration Information".

10. Write the set parameters and the created program to the CPU module on the control station.

[Online] ⇨ [Write to PLC]

At this time, check that MELSOFT Mirror is powered on.

11. Click the [RESET] button for the CPU module on MELSOFT Mirror.



Point

In this example, default values are used for parameters that are not shown above. For the parameters, refer to the following.

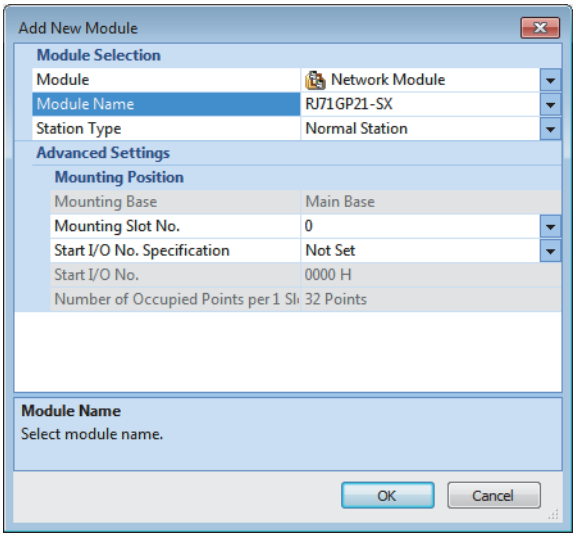
MELSEC iQ-R CC-Link IE Controller Network User's Manual (Application)

8.4 Setting the Normal Stations

Connect the engineering tool to the CPU module on the normal station, and set parameters. Configure the same settings to the station number 2 and 3.

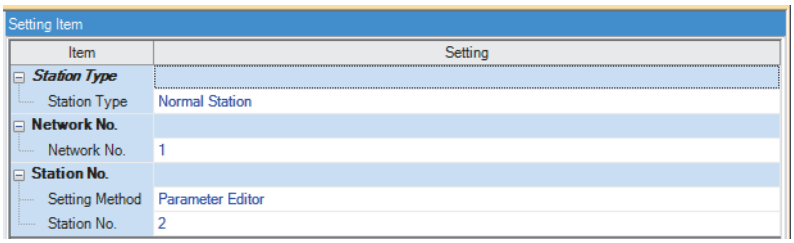
1. Set the CPU module, and add the module labels of the CPU module. The setting and adding methods are the same as those of the control station. (☞ Page 88 Setting the Control Station)
2. Set the CC-Link IE Controller Network-equipped module as follows.

[Navigation window] ⇨ [Parameter] ⇨ [Module Information] ⇨ Right-click ⇨ [Add New Module]



3. Add the module labels of the CC-Link IE Controller Network-equipped module. The adding method is the same as that of the control station. (☞ Page 88 Setting the Control Station)
4. Set the parameters of "Required Settings" as follows. For the station number 3, set "3" to "Station No.".

[Navigation window] ⇨ [Parameter] ⇨ [Module Information] ⇨ [RJ71GP21-SX] ⇨ [Required Settings]



5. Set the refresh settings as follows. Configure the same refresh settings to the station number 2 and 3 of the normal stations.

[Navigation window] ⇨ [Parameter] ⇨ [Module Information] ⇨ [RJ71GP21-SX] ⇨ [Basic Settings] ⇨ [Refresh Setting]

No.	Link Side					CPU Side				
	Device Name	Points	Start	End		Target	Device Name	Points	Start	End
-	SB	512	00000	001FF	⇔	Module Label				
-	SW	512	00000	001FF	⇔	Module Label				
1	LB	1536	00000	005FF	⇔	Specify Device	B	1536	00000	005FF
2	LW	1536	00000	005FF	⇔	Specify Device	W	1536	00000	005FF

6. Connect the engineering tool to the normal station on MELSOFT Mirror.

[Debug] ⇨ [Simulation] ⇨ [MELSOFT Mirror] ⇨ [Connect Simulation]

Double-click the following CPU module displayed in "System Configuration Information".

Station number 2: [System2] ⇨ [Base0:R35B] ⇨ [R04CPU]

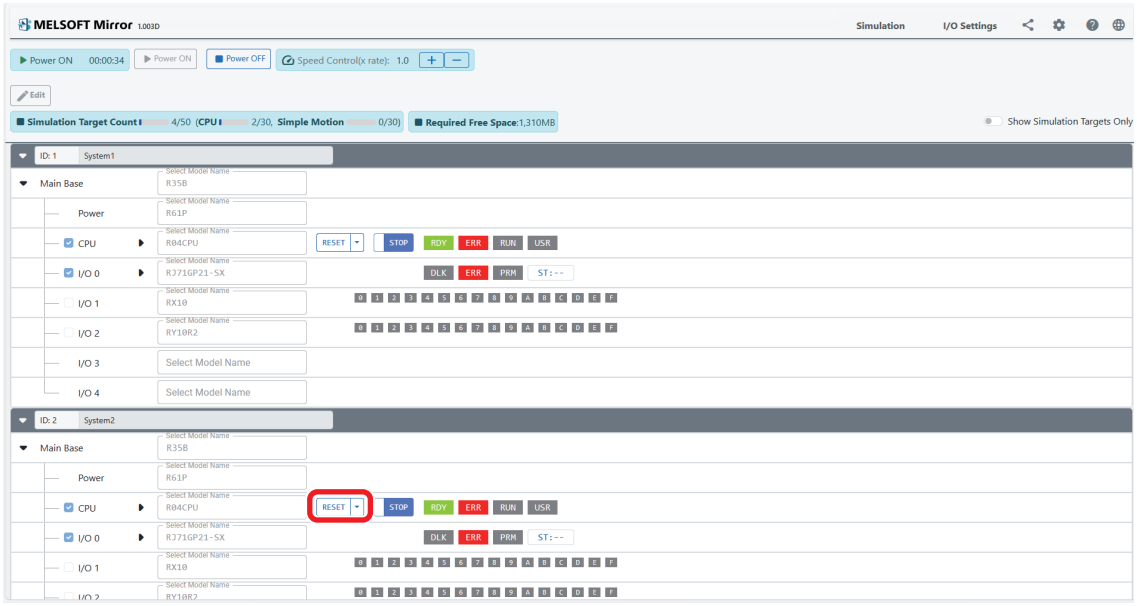
Station number 3: [System3] ⇨ [Base0:R35B] ⇨ [R04CPU]

7. Write the set parameters and the created program to the CPU module on the normal station.

[Online] ⇨ [Write to PLC]

At this time, check that MELSOFT Mirror is powered on.

8. Click the [RESET] button for the CPU module on MELSOFT Mirror.



Point

In this example, default values are used for parameters that are not shown above. For the parameters, refer to the following.

MELSEC iQ-R CC-Link IE Controller Network User's Manual (Application)

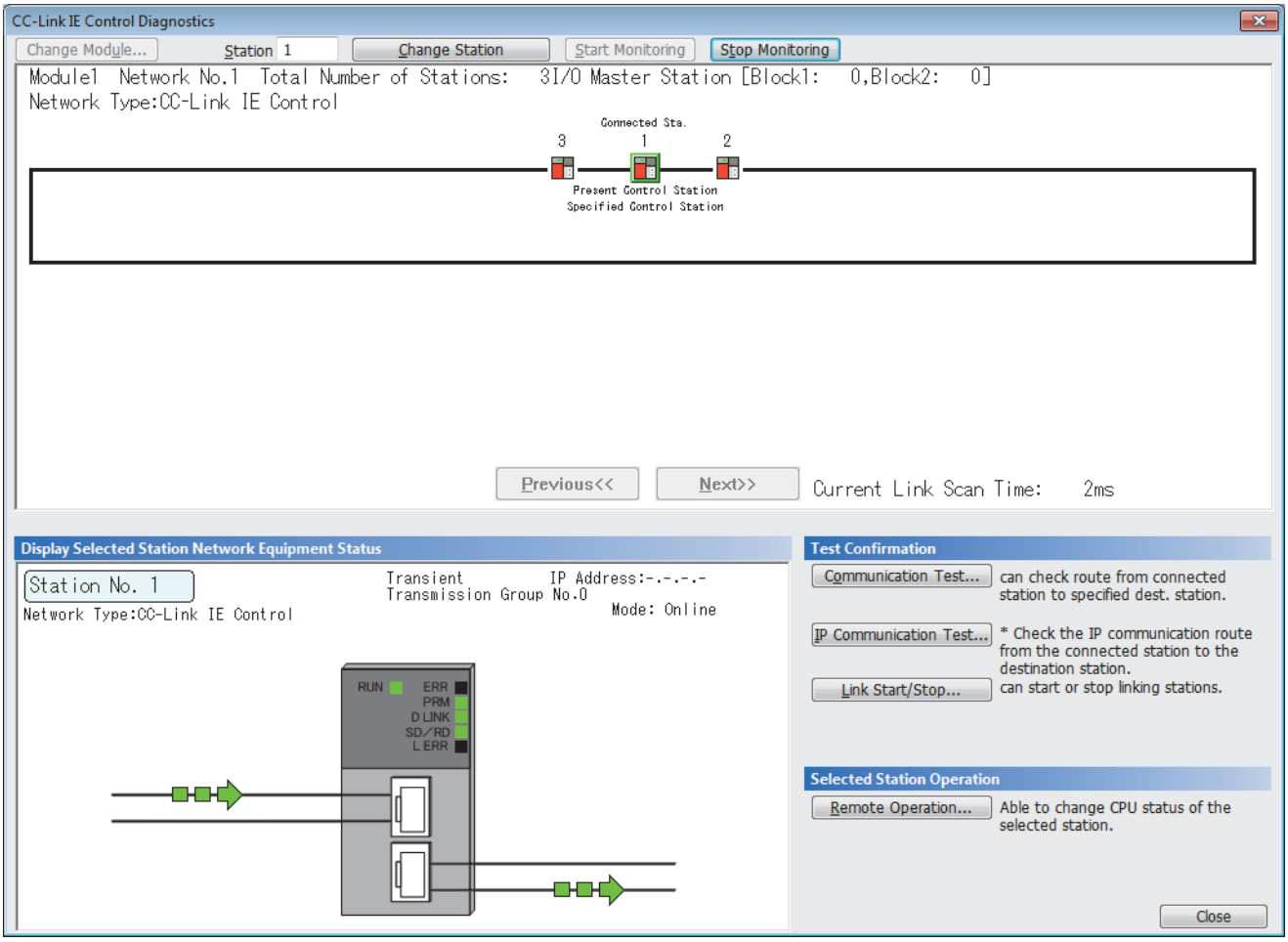
8.5 Checking the Network Status

After setting the parameters of the control station and normal stations, check whether data link can be normally performed. Use the CC-Link IE Controller Network diagnostics of the engineering tool.

1. Connect the engineering tool to the CPU module on the control station.
2. Start the CC-Link IE Controller Network diagnostics.

[Diagnostics] ⇨ [CC-Link IE Control Diagnostics (Optical Cable)]

If the following display appears, data link is normal.



When an error icon is displayed in "Network information display" on the "CC-Link IE Control Diagnostics" window, identify the error cause using the CC-Link IE Controller Network diagnostics, and take actions. (MELSEC iQ-R CC-Link IE Controller Network User's Manual (Application))

8.6 Program Examples

This section provides program examples.

Program example for cyclic transmission

The following is a program example of cyclic transmission.

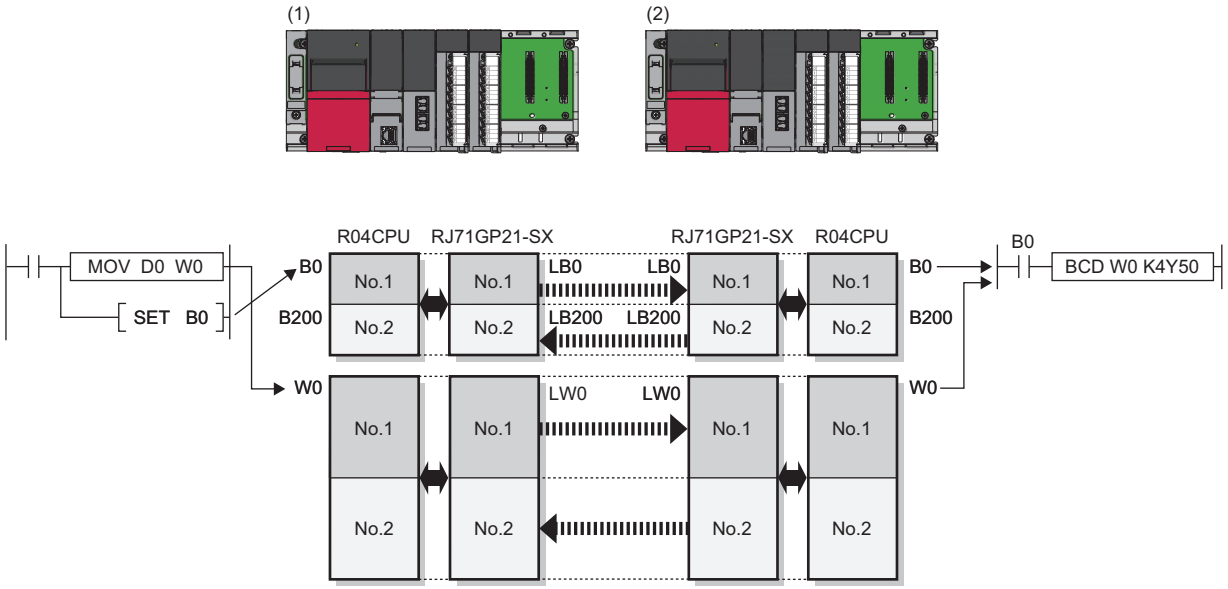
■Overview of the program example

Data in D0 of the control station (No.1) is sent to Y50 to Y5F of the normal station (No.2) by cyclic transmission.

In this example, an interlock is configured with the link relay (LB) and cyclic transmission is performed.

The numbers in the figure (No.1 and No.2) indicate station numbers.

No.1 is a control station, and No.2 is a normal station.



Cyclic transmission (range set in the network range assignment)

Link refresh (range set in the refresh settings)

(1) Control station (No.1)

(2) Normal station (No.2)

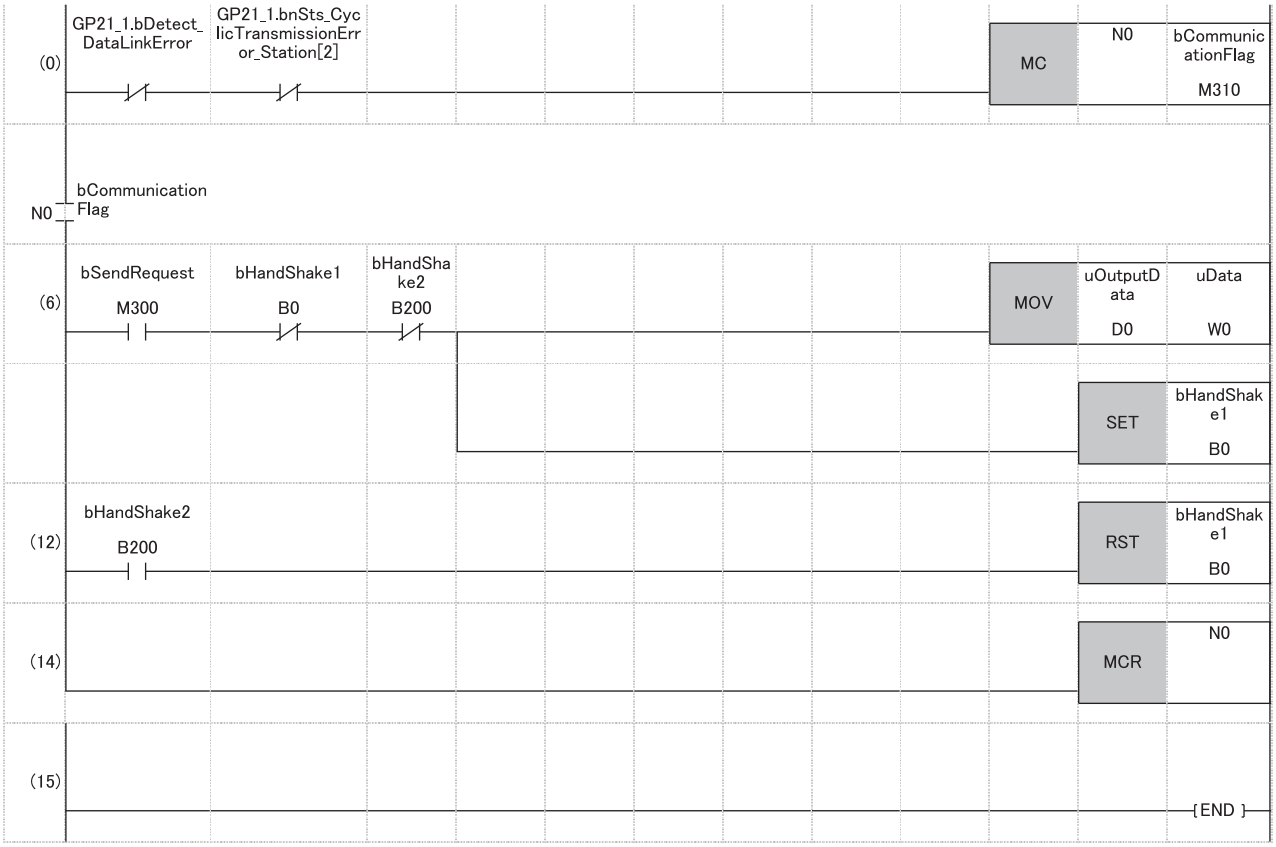
■Program examples

When creating a cyclic transmission program, configure an interlock with 'Data link error status of own station' (SB0049) and 'Data link status of each station' (SW00B0 to SW00B7).

• Control station (No.1)

Classification	Label name	Description	Device
Module label	GP21_1.bDetect_DataLinkError	Data link error status of own station	SB0049
	GP21_1.bnSts_CyclicTransmissionError_Station[2]	Data link status of each station (station No.2)	SW00B0.1
Label to be defined	Define global labels as shown below:		

	Label Name	Data Type		Class	Assign (Device/Label)
1	bSendRequest	Bit	...	VAR_GLOBAL	M300
2	bCommunicationFlag	Bit	...	VAR_GLOBAL	M310
3	bHandShake1	Bit	...	VAR_GLOBAL	B0
4	bHandShake2	Bit	...	VAR_GLOBAL	B200
5	uData	Word [Unsigned]/Bit String [16-bit]	...	VAR_GLOBAL	W0
6	uOutputData	Word [Unsigned]/Bit String [16-bit]	...	VAR_GLOBAL	D0



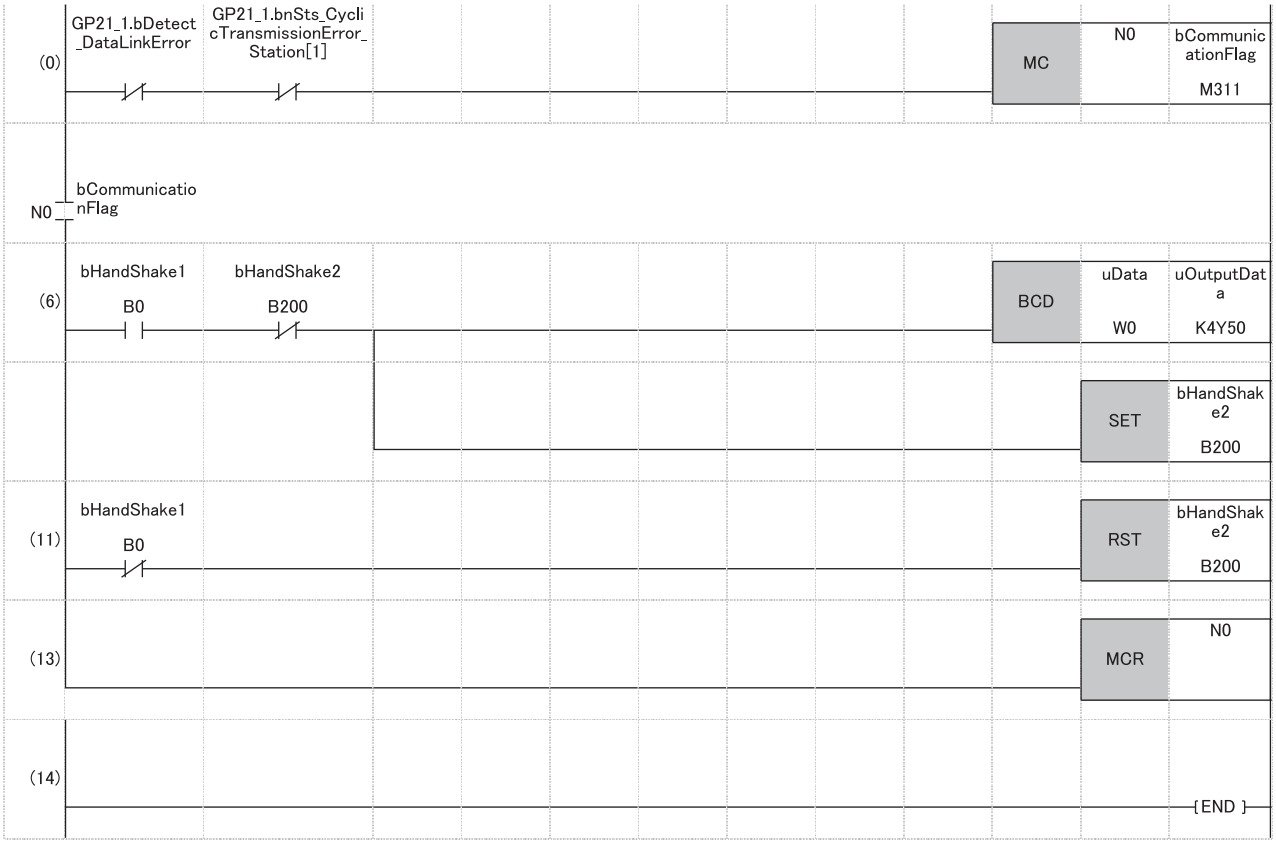
(6) When 'bSendRequest' (M300) is turned on, the contents of 'uOutputData' (D0) is stored in 'uData' (W0).
When data is stored, 'bHandShake1' (B0) turns on.
Data in the link relay (LB) is sent first, and then data in the link register (LW) is sent by cyclic transmission.

(12) When data is sent to the receiving station and 'bHandShake2' (B200) turns on, 'bHandShake1' (B0) turns off.

• Normal station (No.2)

Classification	Label name	Description	Device
Module label	GP21_1.bDetect_DataLinkError	Data link error status of own station	SB0049
	GP21_1.bnSts_CyclicTransmissionError_Station[1]	Data link status of each station (station No.1)	SW00B0.0
Label to be defined	Define global labels as shown below:		

	Label Name	Data Type		Class	Assign (Device/Label)
1	bCommunicationFlag	Bit	...	VAR_GLOBAL	M311
2	bHandShake1	Bit	...	VAR_GLOBAL	B0
3	bHandShake2	Bit	...	VAR_GLOBAL	B200
4	uData	Word [Unsigned]/Bit String [16-bit]	...	VAR_GLOBAL	W0
5	uOutputData	Word [Unsigned]/Bit String [16-bit]	...	VAR_GLOBAL	K4Y50



(6) When 'bHandShake1' (B0) turns on, data in 'uData' (W0) is stored in 'uOutputData' (Y50 to Y5F).
When data is stored, 'bHandShake2' (B200) turns on.

Program example for error detection in other stations

■Data flow

In this example, the program monitors the baton pass status and cyclic transmission status of each station in a three-station system and detects a network error in other stations.

Point

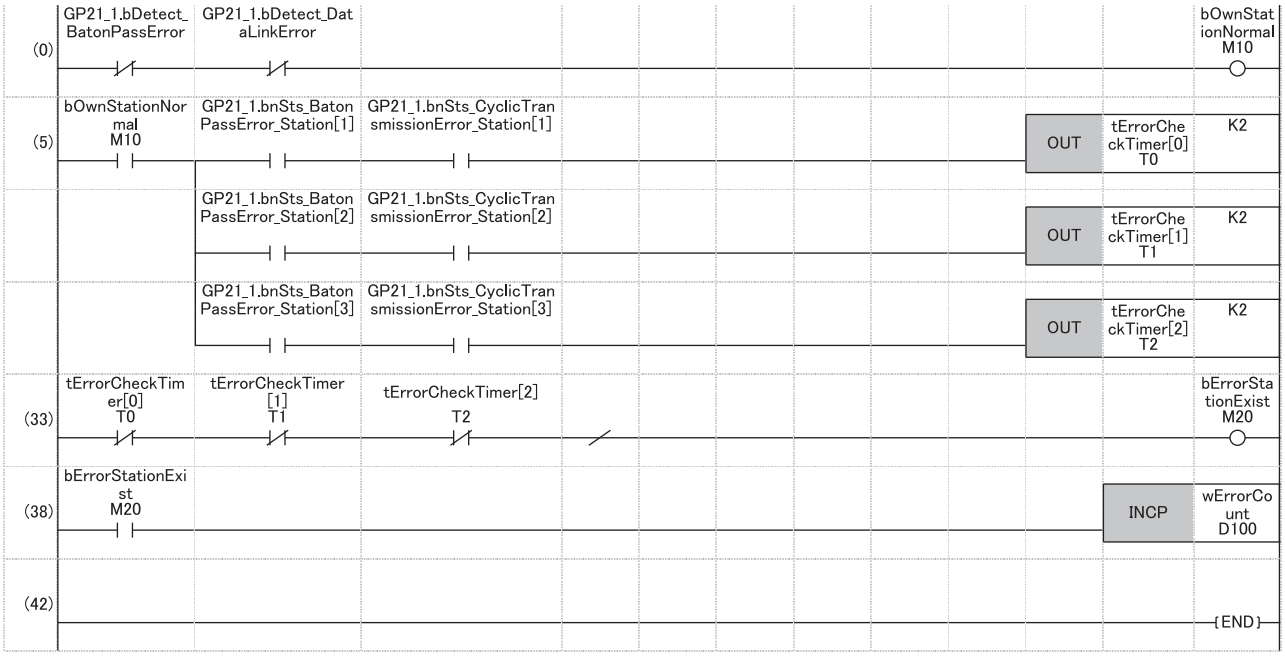
- When a disconnection or a reconnection of stations is detected, the CC-Link IE Controller Network performs line control in all stations to reconstruct the loop.
- During line control, 'Baton pass status of each station' (SW00A0 to SW00A7) and 'Data link status of each station' (SW00B0 to SW00B7) turn on for up to 100ms on all stations (normally 50ms or less). Set an error monitoring timer value to greater than 100ms within the range allowed by the system in use.

■Program example

In this program example, the error monitoring timer is set to 200ms.

Classification	Label name	Description	Device
Module label	GP21_1.bDetect_BatonPassError	Baton pass error status of own station	SB0047
	GP21_1.bDetect_DataLinkError	Data link error status of own station	SB0049
	GP21_1.bnSts_BatonPassError_Station[1]	Baton pass status of each station (station No.1)	SW00A0.0
	GP21_1.bnSts_BatonPassError_Station[2]	Baton pass status of each station (station No.2)	SW00A0.1
	GP21_1.bnSts_BatonPassError_Station[3]	Baton pass status of each station (station No.3)	SW00A0.2
	GP21_1.bnSts_CyclicTransmissionError_Station[1]	Data link status of each station (station No.1)	SW00B0.0
	GP21_1.bnSts_CyclicTransmissionError_Station[2]	Data link status of each station (station No.2)	SW00B0.1
	GP21_1.bnSts_CyclicTransmissionError_Station[3]	Data link status of each station (station No.3)	SW00B0.2
Label to be defined	Define global labels as shown below:		

	Label Name	Data Type		Class	Assign (Device/Label)
1	bOwnStationNormal	Bit	...	VAR_GLOBAL	M10
2	bErrorStationExist	Bit	...	VAR_GLOBAL	M20
3	tErrorCheckTimer	Timer(0..2)	...	VAR_GLOBAL	T0
4	wErrorCount	Word [Signed]	...	VAR_GLOBAL	D100



(38) Processing performed when an error is detected

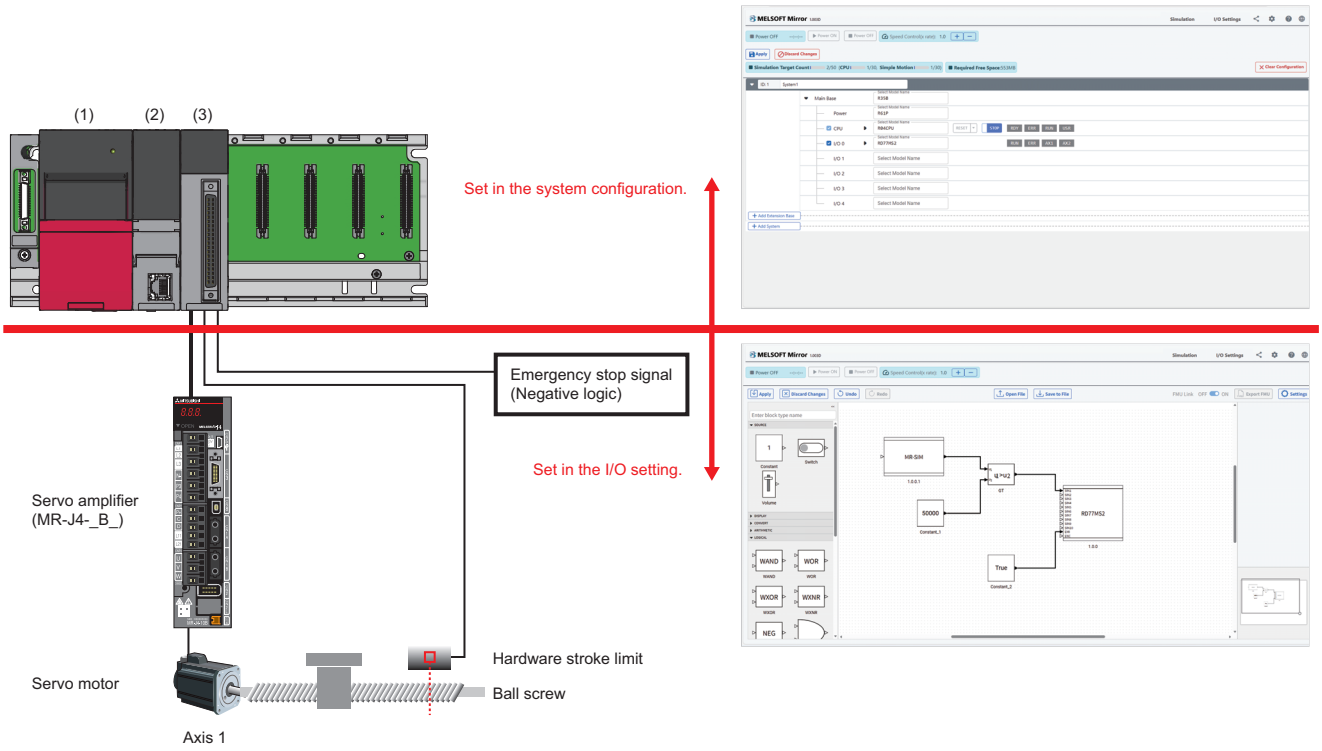
9 SIMPLE MOTION SIMULATION

9.1 System Configuration Example

A system where a servo amplifier (MR-J4) is connected to the MELSEC iQ-R series Simple Motion module (RD77MS2) and a feed machine is connected as an external device is used for explanation.

System configuration

- Main base unit: R35B
- Power supply module: R61P
- CPU module: R04CPU
- Simple Motion module: RD77MS2 (connected servo amplifier: MR-J4)

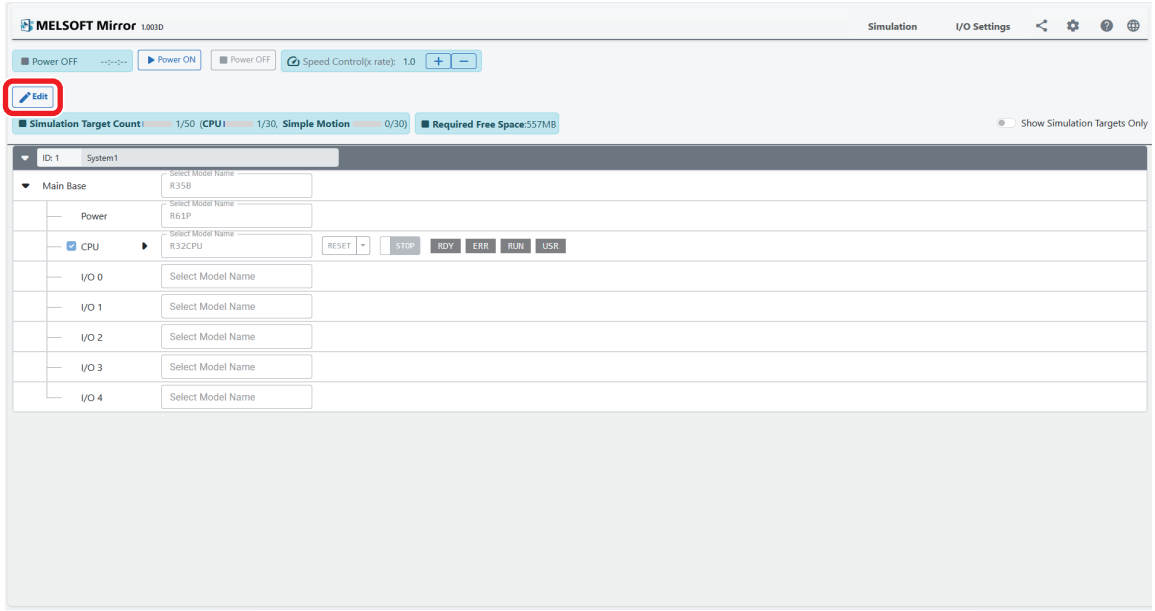


(1) Power supply module
(2) CPU module
(3) Simple Motion module

9.2 Setting System Configuration

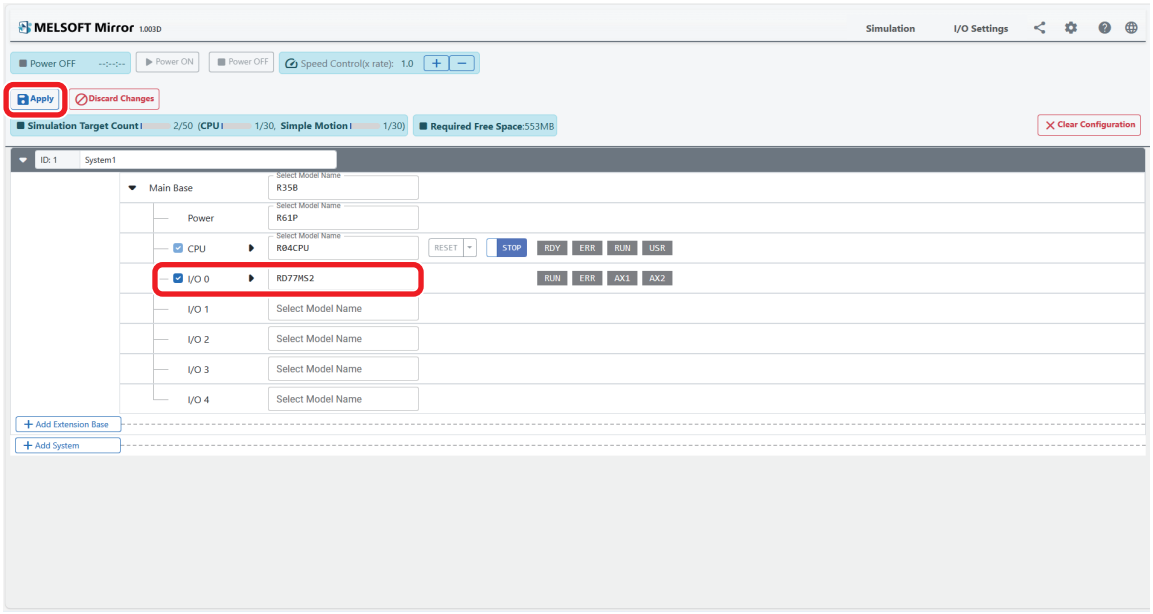
Set the Simple Motion system configuration using I/O Setting on MELSOFT Mirror.

1. Set the system configuration from the top window of MELSOFT Mirror. Click the [Edit] button to change the system configuration.

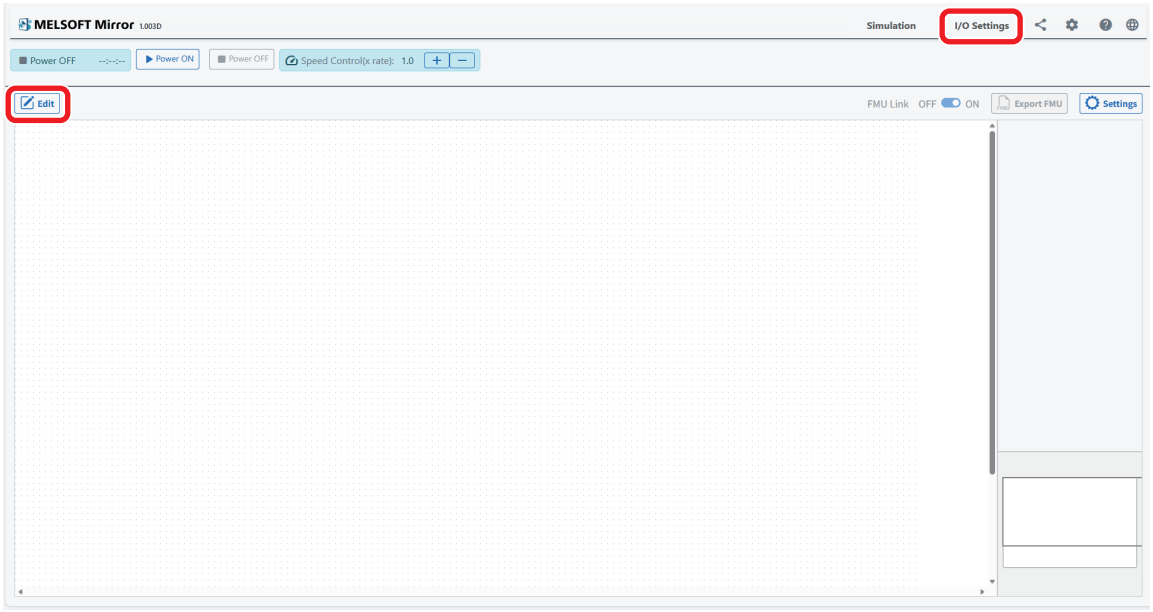


2. Select [R04CPU] from the "CPU" combo box.
3. Select [RD77MS2] from the "I/O0" combo box, and check that the "I/O0" checkbox is selected.

4. Click the [Apply] button to complete the system configuration.



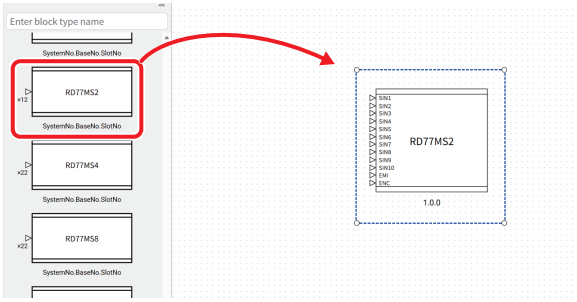
5. Configure settings for simulating an external device. Click "I/O Settings" to open the "I/O Settings" window, and click the [Edit] button.



Point

For details on the I/O setting, refer to the following. (☞ Page 118 I/O SETTING)
This section describes the operation using the following program example. When the machine feed value (MR-SIM output) reaches the machine end (50.00mm), the upper limit switch of the hardware stroke limit turns on, and simple motion stops due to an error. After the error is cleared, the simple motion stops and the machine feed value returns to 0.

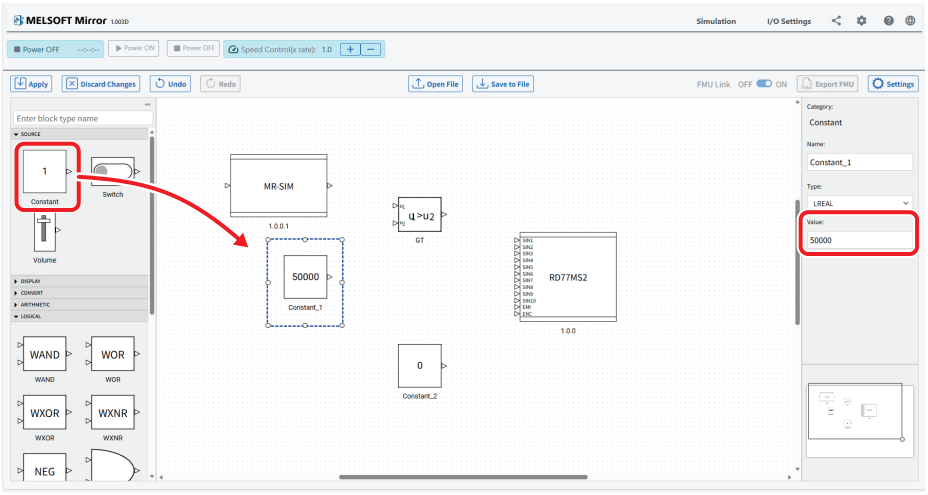
6. Select "Controller" from the element menu, and drag and drop "RD77MS2" to the center of the window.



7. Select "MR-SIM" from the element menu "Controller", and drag and drop it to the window.

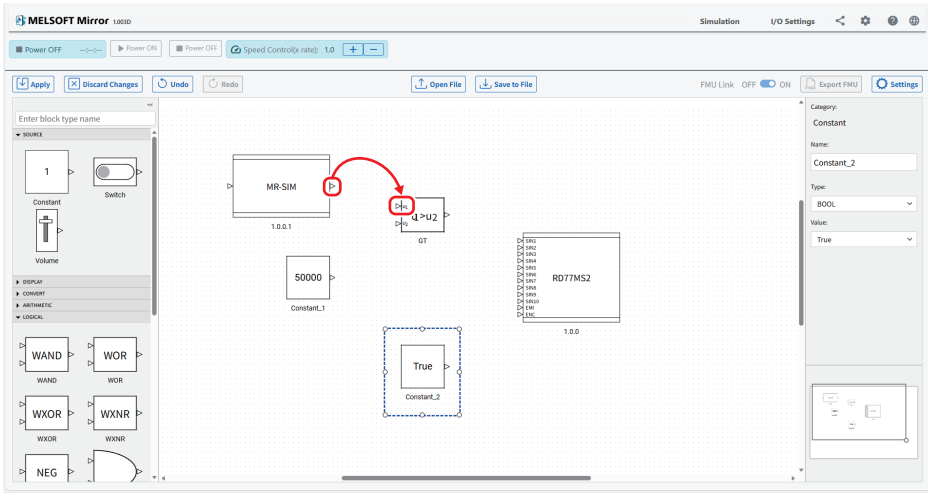
8. Select "GT" from the element menu "Logical Operation", and drag and drop it to the window.

9. Select "Constant" from the element menu "Source", and drag and drop it to the window. Set "50000" to "Value" in the menu displayed on the right side of the window.

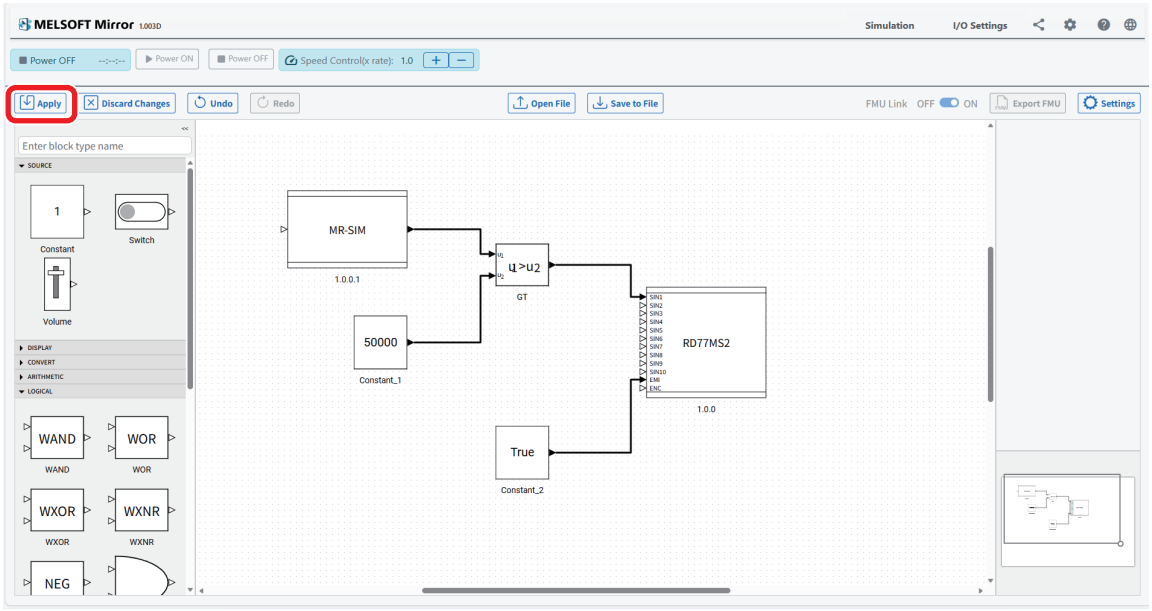


10. Select "Constant" from the element menu "Source", and drag and drop it to the window. Set "BOOL" to "Type" and "TRUE" to "Value" in the menu displayed on the right side of the window.

11. Connect blocks with a line. Use drag and drop to draw a line from the output of one block to the input of another.



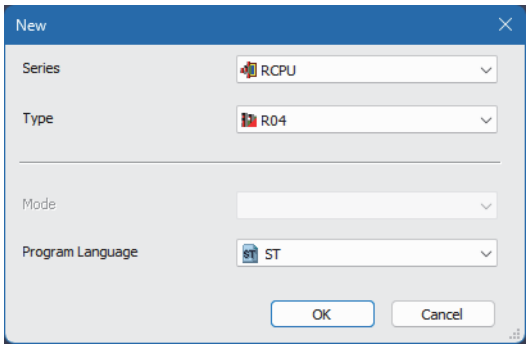
12. After drawing the lines as shown below, click [Apply]. In particular, check that the output of the "GT" block is connected to "SIN1" (general-purpose input signal) of the "RD77MS2" block, and the output of the "Constant_1" block is connected to "EMI" (forced stop input signal) of the "RD77MS2" block.



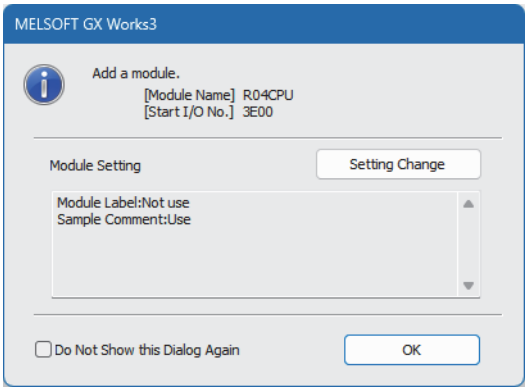
13. Click the [Power ON] button to start the simulation.

9.3 Creating a Project with the Engineering Tool

Procedure



1. Select "RCPU" for "Series" and the CPU module model to be used for "Type". Then, select a programming language to be used in the project for "Program Language". In this example, select "ST", and click the [OK] button.



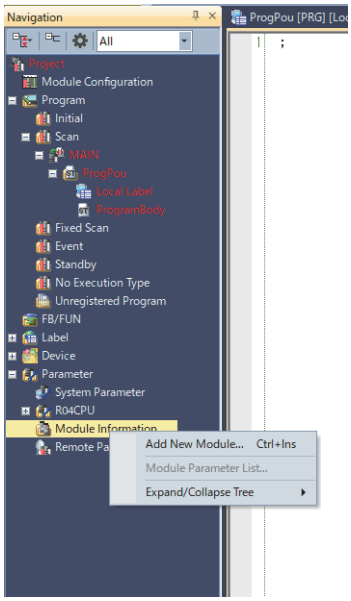
2. When the window for adding a module appears, click the [OK] button.

9.4 Setting Parameters

This section describes how to set system parameters and parameters for each module.

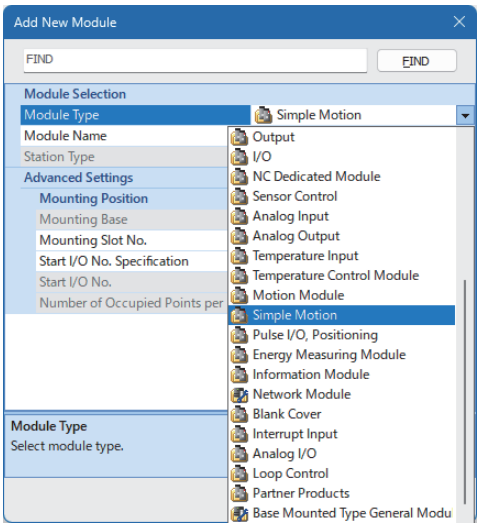
Setting parameters from the Navigation window

Parameters are set from the Navigation window.



1. Right-click "Module Information" from "Navigation window", and click "Add New Module".

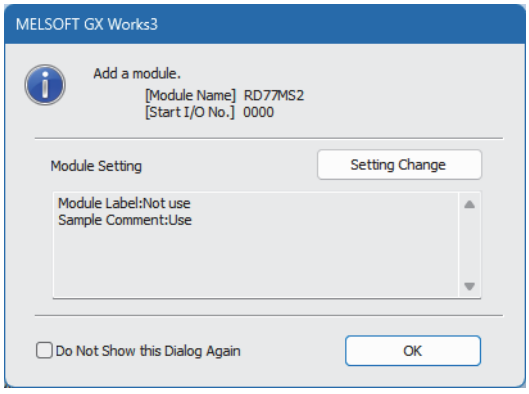
[Navigation window] ⇨ [Parameter] ⇨ [Module Information]



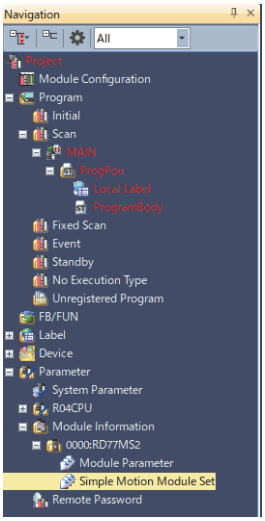
2. Select "Simple Motion" for "Module Type".

3. Select "RD77MS2" for "Module Name".

4. Click the [OK] button.

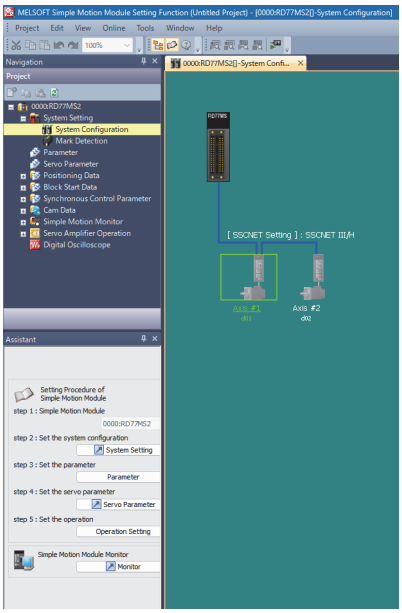


5. Click the [OK] button.



6. Double-click "Simple Motion Module Setting (Module Extended Parameter)".

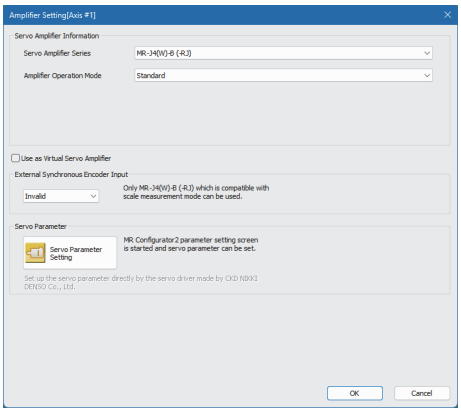
[Navigation window] ⇨ [Parameter] ⇨ [Module Information] ⇨ [0000: RD77MS2] ⇨ [Simple Motion Module Setting (Module Extended Parameter)]



7. Double-click "System Configuration".

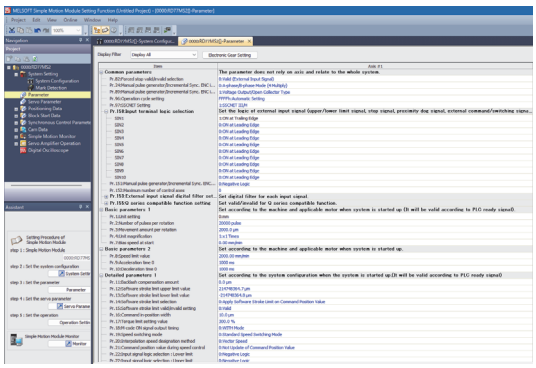
[Navigation window] ⇨ [Project] ⇨ [0000: RD77MS2] ⇨ [System Configuration] ⇨ [System Configuration]

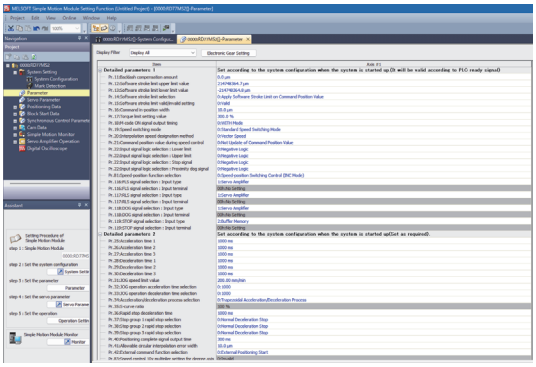
8. Double-click "Axis 1" in the "System Configuration" window.

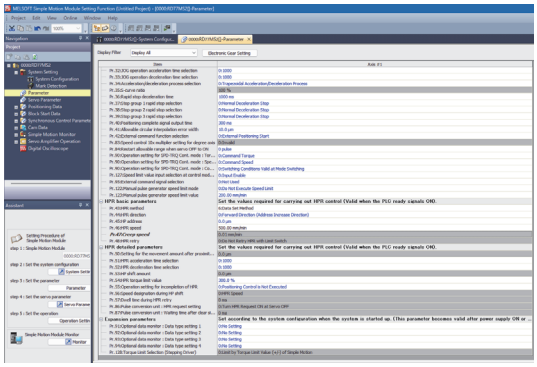


9. Change "Servo Amplifier Series" to "MR-J4(W)-B (-RJ)".

10. Click the [OK] button.





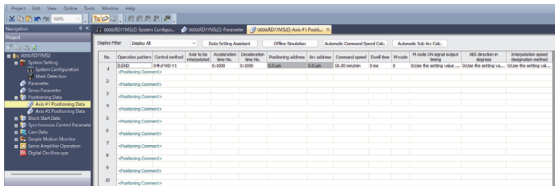


11. Double-click "Parameter".

[Navigation window] ⇨ [Project] ⇨ [0000: RD77MS2] ⇨ [Parameter]

12. Change the following parameters.

- [Common parameters] ⇨ [PR.150: Input terminal logic selection] ⇨ [SIN1]: "1: ON at trailing edge"
- [Basic parameters 1] ⇨ [PR.1: Unit setting]: "0: mm"
- [Detailed parameters1] ⇨ [PR.116: FLS signal selection: Input type]: "0: Simple Motion module"
- [Detailed parameters1] ⇨ [PR.116: FLS signal selection: Input terminal]: "01h: SIN1"
- [Home position return basic parameters] ⇨ [PR.43: Home position return method]: "6: Data set method"
- [Home position return basic parameters] ⇨ [PR.46: Home position return speed]: "500.00mm/min"

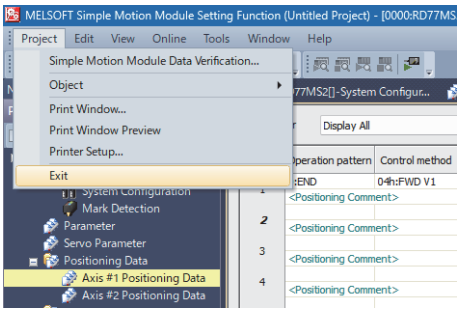


13. Double-click "Axis 1 positioning data No.".

[Navigation window] ⇨ [Project] ⇨ [0000: RD77MS2] ⇨ [Positioning data] ⇨ [Axis 1 positioning data No.]

14. Set the following parameters for No.1.

- [Operation pattern]: "0: Positioning complete"
- [Control method]: "04h: Forward run speed 1"
- [Axis to be interpolated]: "-"
- [Acceleration time No.]: "0: 1000"
- [Deceleration time No.]: "0: 1000"
- [Positioning address]: "0.0μm"
- [Arc address]: "0.0μm"
- [Command speed]: "50.00mm/min"
- [Dwell time]: "0ms"
- [M code]: "0"
- [M code ON signal output timing]: "0: Uses the setting value of "[Pr.18] M code ON signal output timing"."
- [ABS direction in degrees]: "0: Uses the setting value of "[Cd.40] ABS direction in degrees"."
- [Interpolation speed designation method]: "0: Uses the setting value of "[Pr.20] Interpolation speed designation method"."



15. From the menu, click [Project] ⇨ [Exit].

9.5 Creating Programs

After setting the parameters, create programs.

Classification	Label
Label to be defined	Define local labels as shown below.

	Label Name	Data Type		Class	Initial Value	Constant
1	Machine_Origin	Bit	...	VAR		
2	Fast_Origin	Bit	...	VAR		
3	Move_pos	Bit	...	VAR		
4	ErrorReset	Bit	...	VAR		
5	TON_1	TON	...	VAR		
6			...			

Create the following five programs in the structured text language.

- Ready_ServoON
- Origin
- MovePos
- ErrorReset
- Timer

1. Ready_ServoON

ST

```
//If the M0 device does not match TRUE, turn on the programmable controller ready and servo signals.
IF(M0 <> TRUE) THEN
    Y0 := TRUE;
    Y1 := TRUE;
END_IF;
```

2. Origin

ST

```
//Start machine home position return.
IF(Machine_Origin = TRUE) THEN
//Axis 1 is not busy.
  IF(X10 <> TRUE) THEN
//Positioning start data No.9001 (Machine home position return)
    U0\G4300 := 9001;//Cd.3 Positioning start No.
    Y10 := TRUE;//Positioning start

//Return to origin after home position return is completed.
    IF(U0\G2417.4 = TRUE) THEN//Md.31 Status: Home position return complete flag
      U0\G4300 := 0;//Cd.3 Positioning start number
      Y10 := FALSE;//Positioning start

//Set the execution label back to FALSE after 1 second.
        TON_1.IN := TRUE;
        IF(TON_1.Q = TRUE) THEN
          TON_1.IN := FALSE;
          Machine_Origin := FALSE;
        END_IF;
    END_IF;
  END_IF;
END_IF;

//Start fast home position return.
IF(Fast_Origin = TRUE) THEN
//Axis 1 is not busy.
  IF(X10 <> TRUE) THEN
//Positioning start data No.9002 (Fast home position return)
    U0\G4300 := 9002;//Cd.3 Positioning start number
    Y10 := TRUE;//Positioning start
  END_IF;

//Return to origin after home position return is started.
  IF(U0\G2417.E = TRUE) THEN//Md.31 Status: Start complete
    U0\G4300 := 0;//Cd.3 Positioning start number
    Y10 := FALSE;//Positioning start
    Fast_Origin := FALSE;
 END_IF;
END_IF;
```

3. MovePos

ST

```
//Start positioning.
IF(Move_pos = TRUE) THEN
  //Axis 1 is not busy.
  IF(X10 <> TRUE) THEN
    U0\G4300 := 1;//Cd.3 Positioning start number
   Y10 := TRUE;//Positioning start
  END_IF;

 //Return to origin after positioning is started.
  IF(U0\G2417.E = TRUE) THEN//Md.31 Status: Start complete
     U0\G4300 := 0;//Cd.3 Positioning start number
     Y10 := FALSE;//Positioning start
     Move_pos := FALSE;
  END_IF;
END_IF;
```

4. ErrorReset

ST

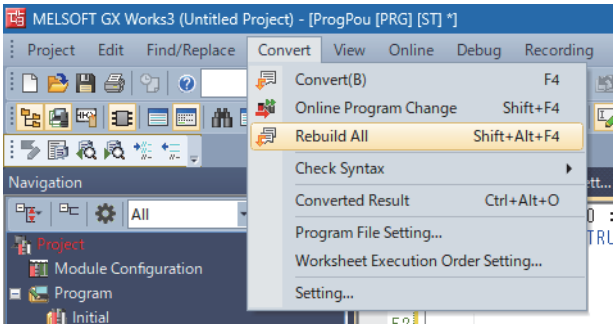
```
//Recover from a hardware stroke limit (+) error.
IF(ErrorReset = TRUE) THEN
  //The error was reset.
  IF(M10 = TRUE) THEN
    //Error is cleared.
    IF(U0\G2417.D = FALSE) THEN//Md.31 Status: Error detection
      //Use JOG operation to move the workpiece outside the range of the hardware stroke limit.
      U0\G4318:D := 100;//Cd.17 JOG speed
      U0\G30102:= 1;//Cd.182 Reverse run JOG start
      M10 := FALSE;
    END_IF;
  END_IF;

  //Error being detected
  IF(U0\G2417.D = TRUE) THEN//Md.31 Status: Error detection
    //Reset the error.
    U0\G4302 := 1;//Cd.5 Axis error reset
    M10 := TRUE;
  END_IF;

  //JOG operation
  IF(U0\G2409 = 3) THEN//Md.26 Axis operation status is 3 (JOG operation).
    //Stop JOG operation because the workpiece has moved within the range of the hardware stroke limit.
    IF(U0\G2402:D < 500000) THEN//Md.21 Machine feed value
      U0\G30102:= 0;//Cd.182 Reverse run JOG start
      ErrorReset := FALSE;
    END_IF;
  END_IF;
END_IF;
```

5. Timer

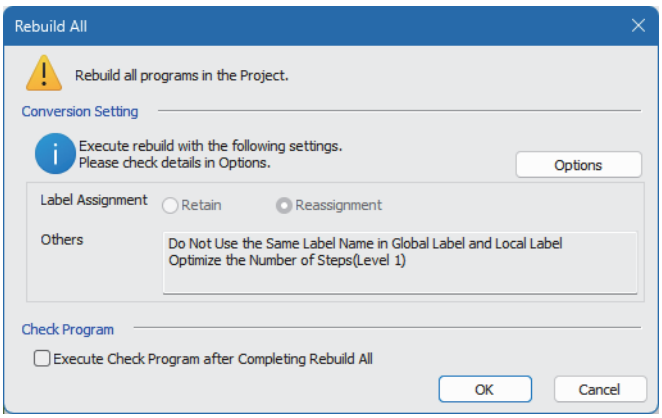
ST

```
//1 second timer
TON_1(PT:= T#1s);
```

6. After the programs are created, click [Convert] ⇨ [Rebuild All] on the menu bar of the engineering tool.

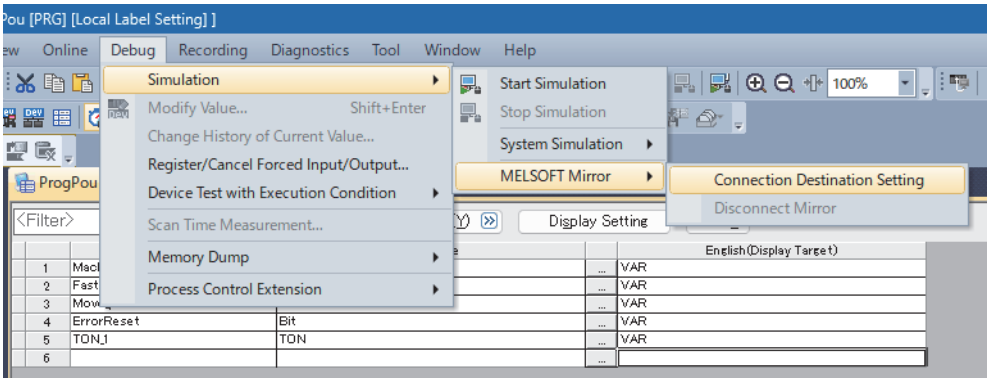


7. When the following window is displayed, click the [OK] button.

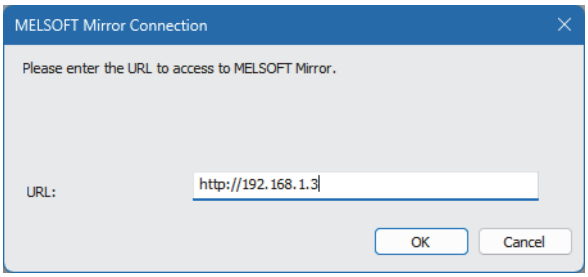


9.6 Connecting the Engineering Tool to MELSOFT Mirror

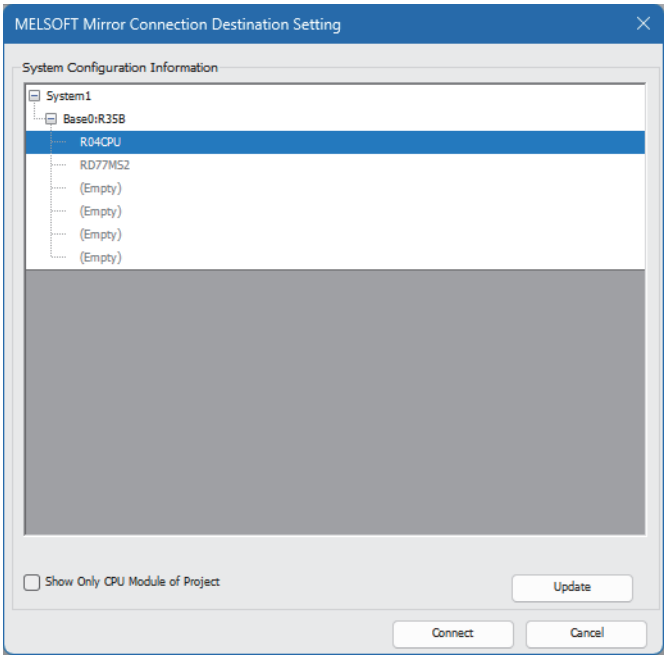
1. After the project is created, click [Debug] ⇨ [Simulation] ⇨ [MELSOFT Mirror] ⇨ [Connect Simulation] on the menu bar of the engineering tool.



2. Enter the URL on the top window of MELSOFT Mirror.



3. Select the CPU module, and click the [Connect] button.

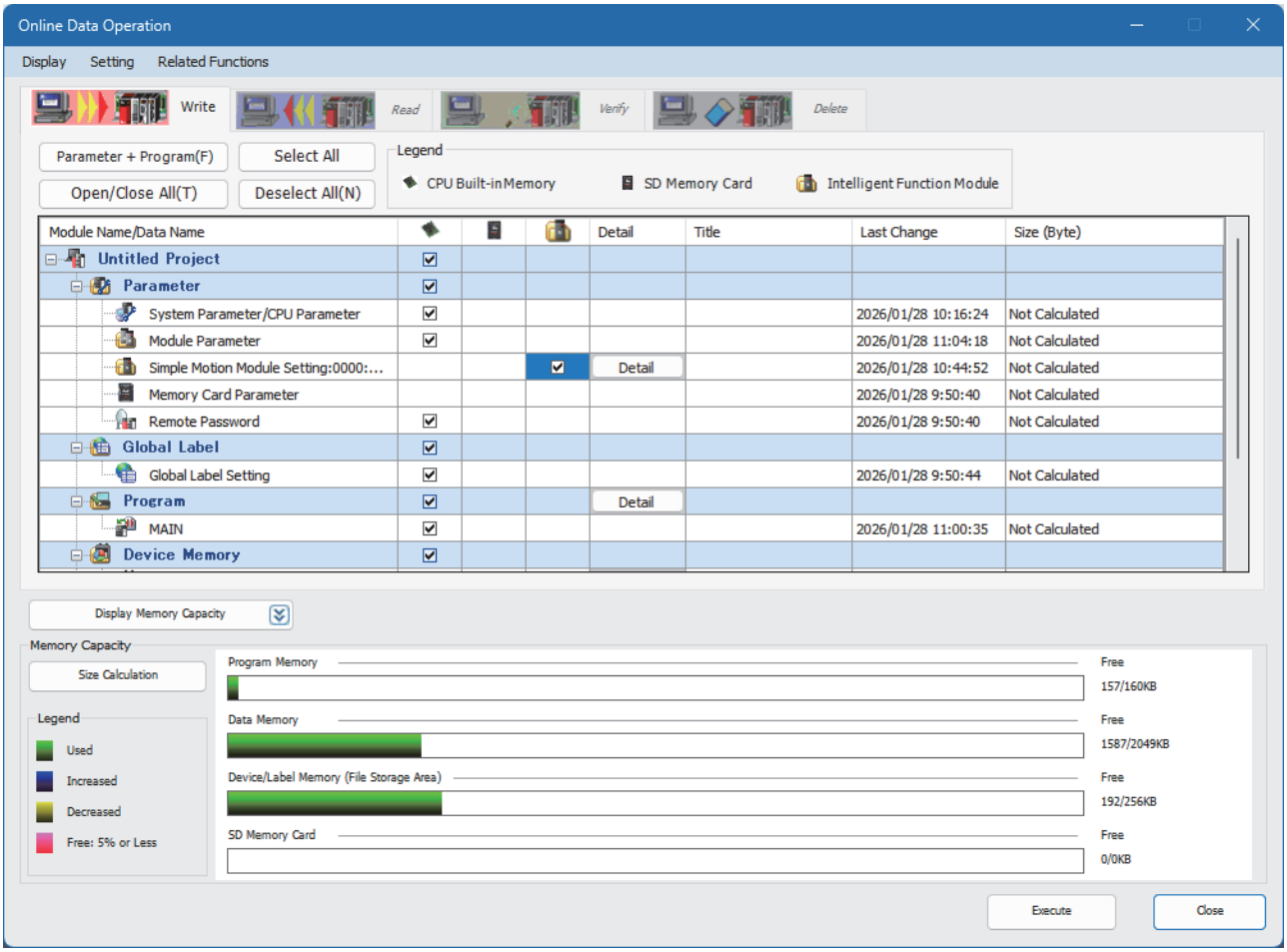


9.7 Writing Data to the Programmable Controller

1. Write the set parameters and the created programs to the CPU module.

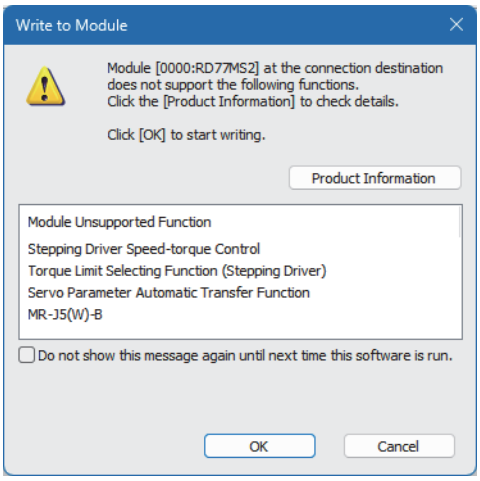
[Online] ⇨ [Write to PLC]

2. Select the system parameter file, CPU parameter file, module parameter file, program file, and Simple Motion module setting file on the "Online Data Operation" window.



3. Click the [Execute] button.

4. When the following window is displayed, click the [OK] button.

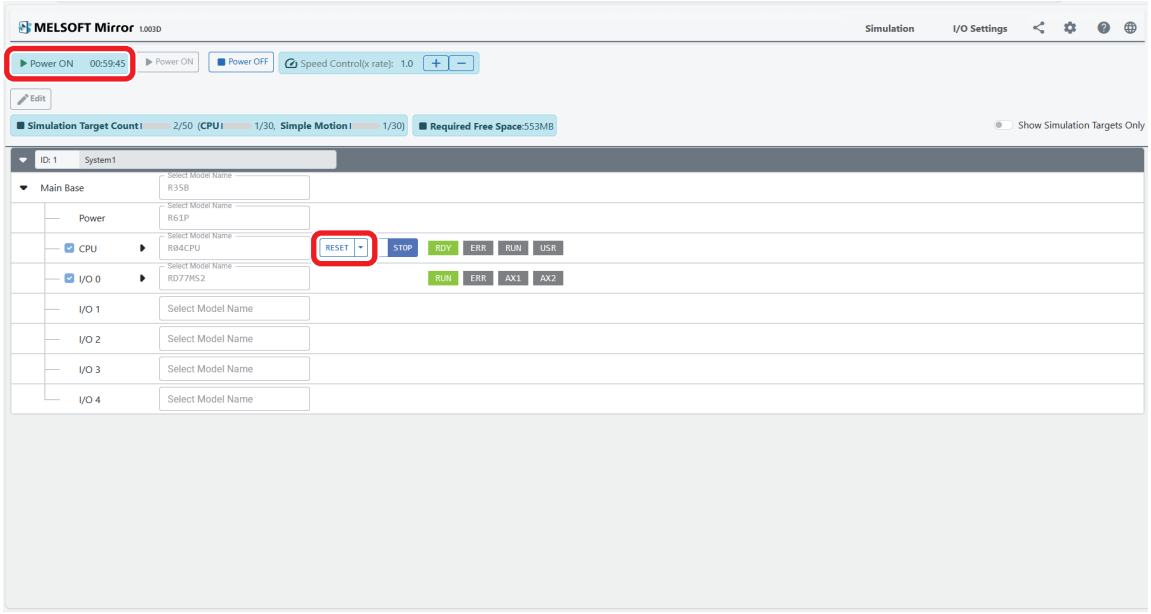


5. After the write processing completes, click the [Close] button.

9.8 Resetting the Modules

Resetting the modules

Reset the modules on MELSOFT Mirror.

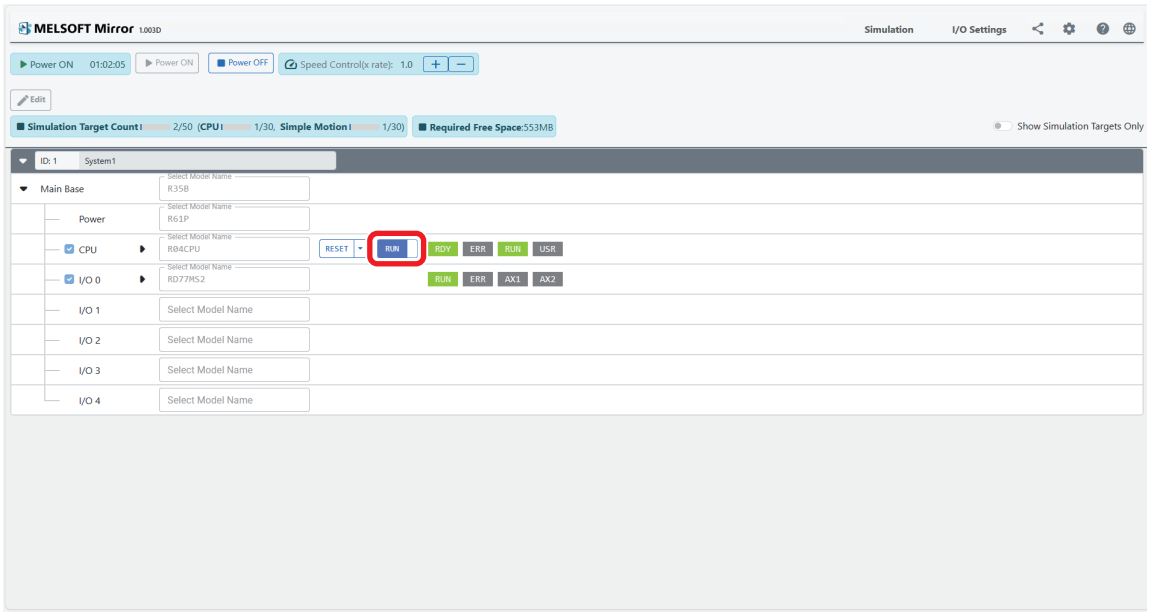


1. Check that MELSOFT Mirror is powered on, and click the [RESET] button for the CPU module.

9.9 Executing the Programs

Executing the programs

Execute the programs written to the programmable controller.



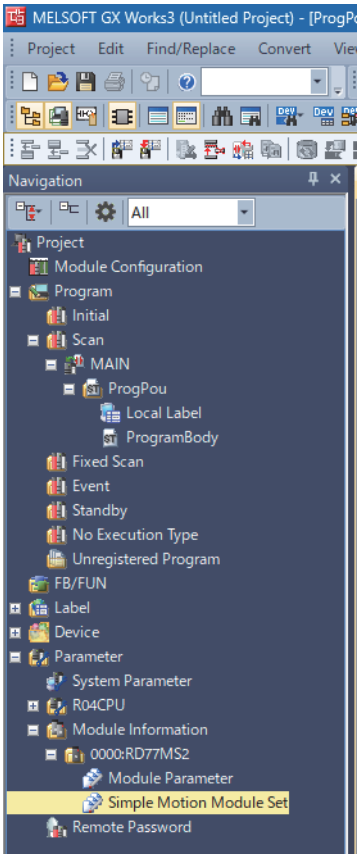
1. Set the switch of the CPU module to [RUN] on MELSOFT Mirror.

9.10 Checking Operating Status Using Axis Monitor

Check the operating status of the servo motor using the axis monitor.

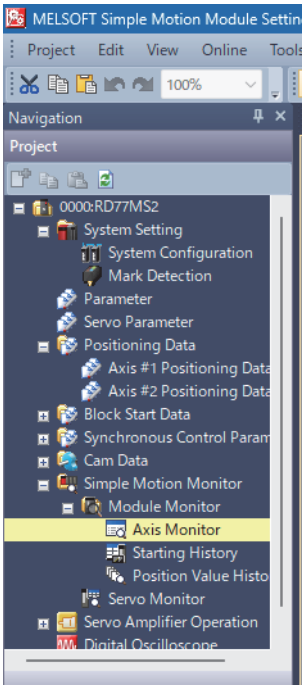
1. Double-click "Simple Motion Module Setting (Module Extended Parameter)".

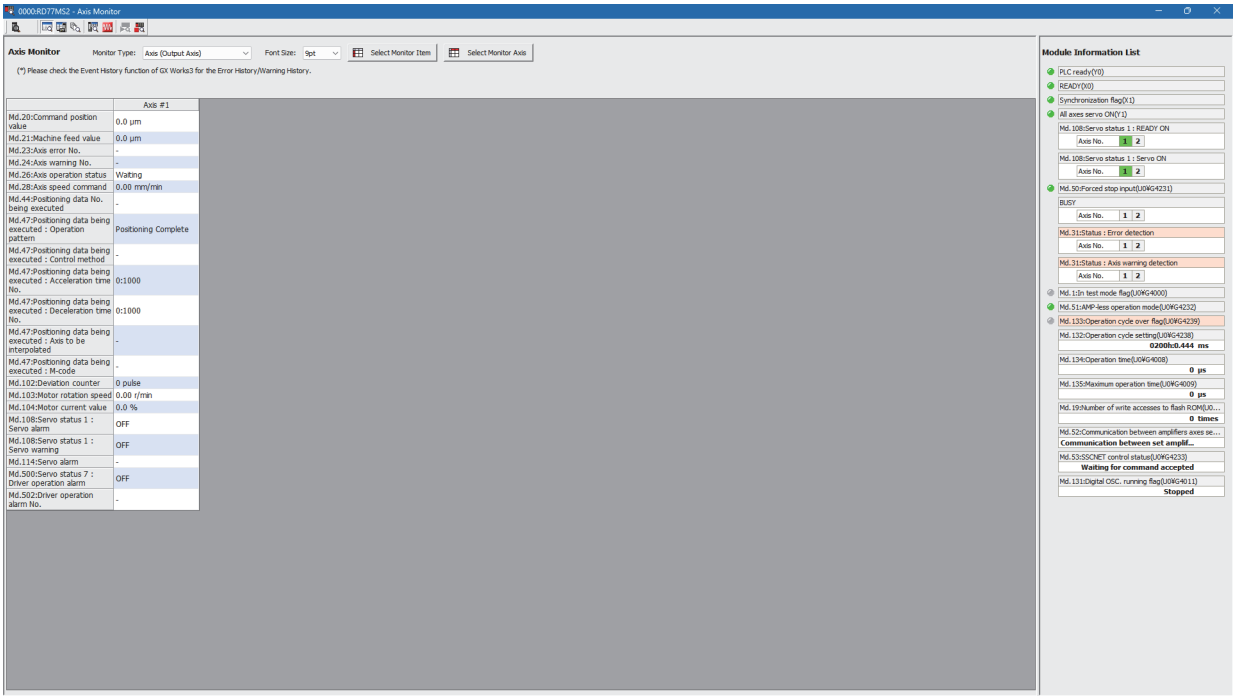
[Navigation window] ⇨ [Parameter] ⇨ [Module Information] ⇨ [0000: RD77MS2] ⇨ [Simple Motion Module Setting (Module Extended Parameter)]



2. Double-click "Axis Monitor" from "Navigation Window" to display the axis monitor.

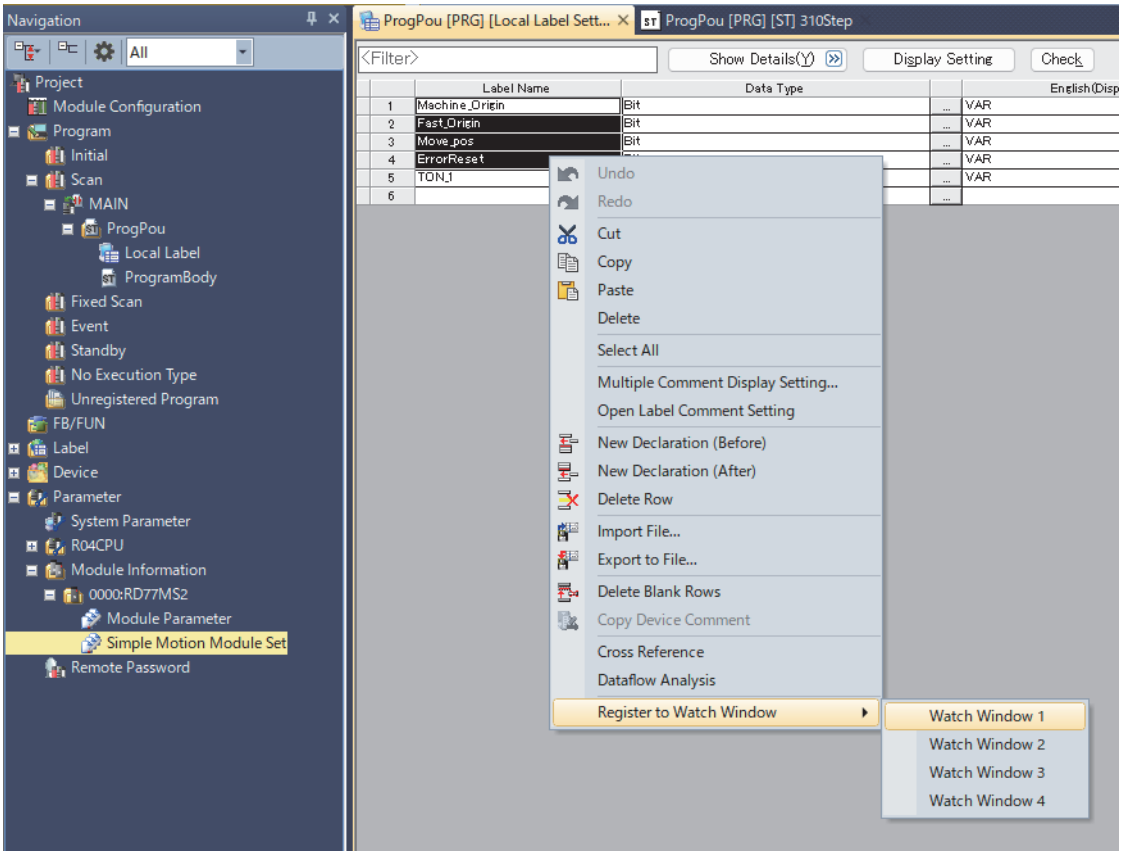
[Navigation window] ⇨ [Project] ⇨ [0000: RD77MS2] ⇨ [Simple Motion Monitor] ⇨ [Module Monitor] ⇨ [Axis Monitor]



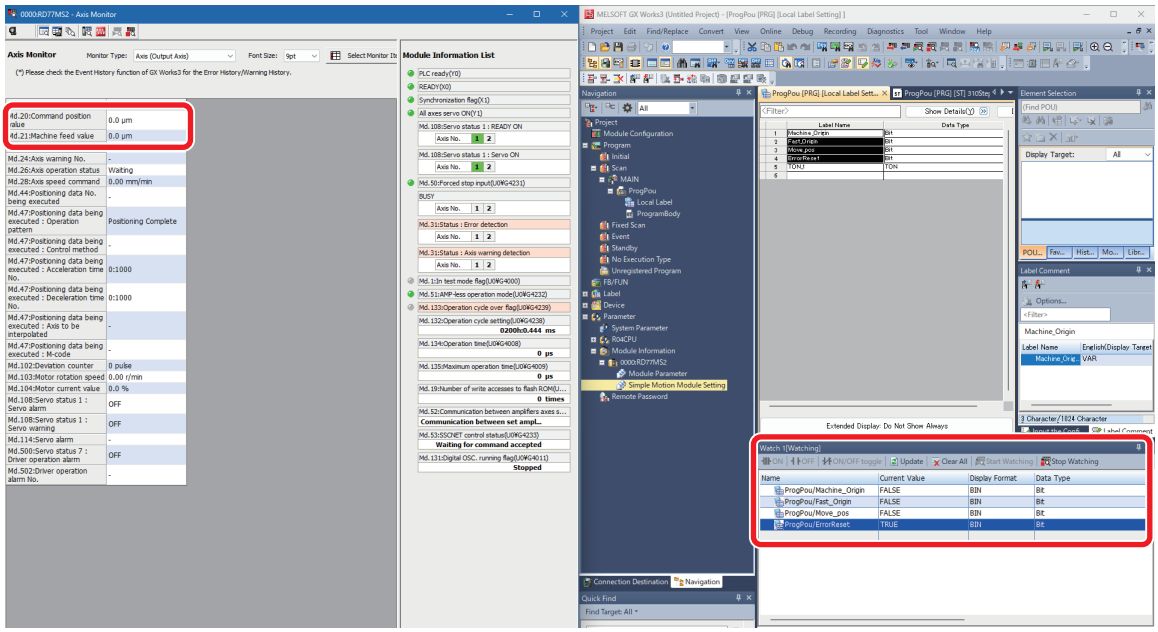


3. Return to the engineering tool, right-click the following label, and register it to "Watch 1".

- Machine_Origin
- Fast_Origin
- Move_Pos
- ErrorReset



4. Control axis 1 by turning on/off the label registered in the Watch 1 window. Display the axis monitor and the Watch 1 window side by side to easily operate and check the operation result.



Point

The operations of labels registered in the Watch 1 window are as follows.

- Machine_Origin: Original position registration
- Fast_Origin: Home position return
- Move_Pos: Positioning
- ErrorReset: Error clear

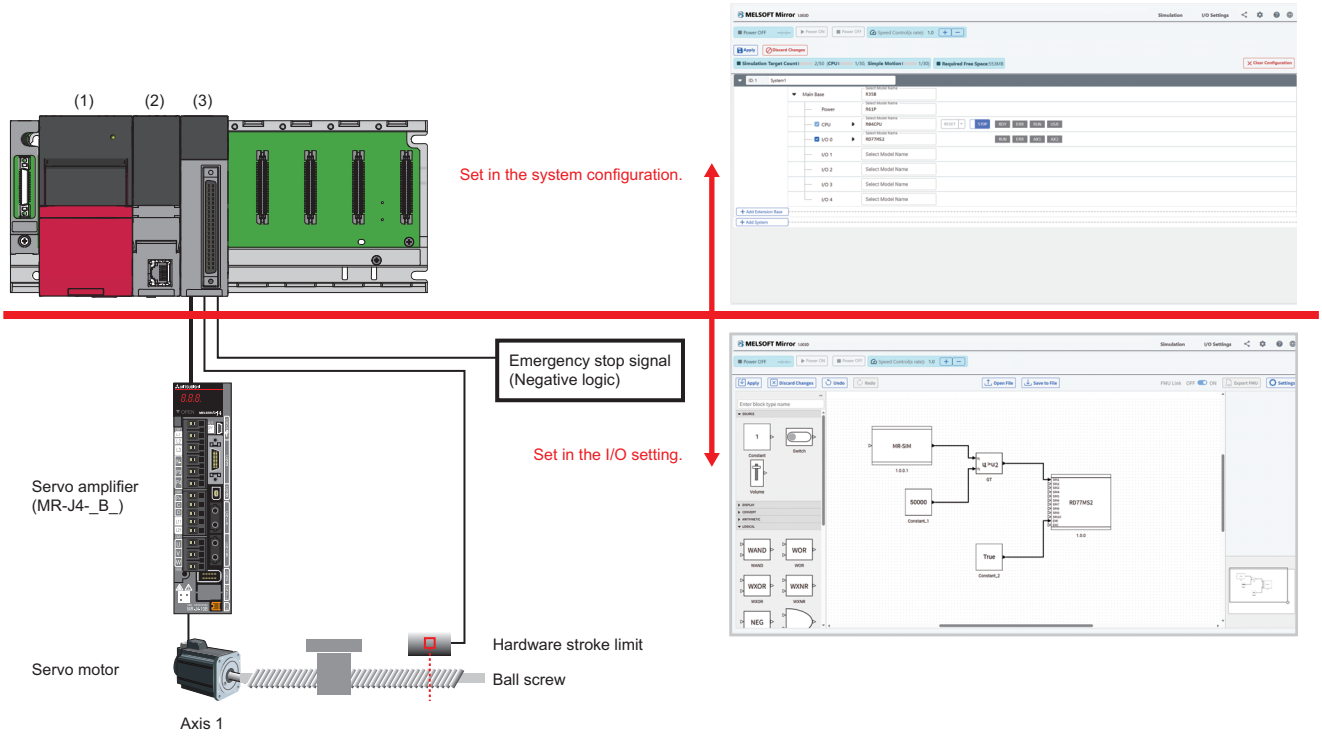
The operating procedure is described below.

- Turn on Machine_Origin to register the origin.
- Positioning starts when Move_Pos is turned on with the origin registered.
- If the operation stops due to an error "Hardware stroke limit (+)", turn on ErrorReset to clear the error.
- Turn on Fast_Origin to return to the home position.

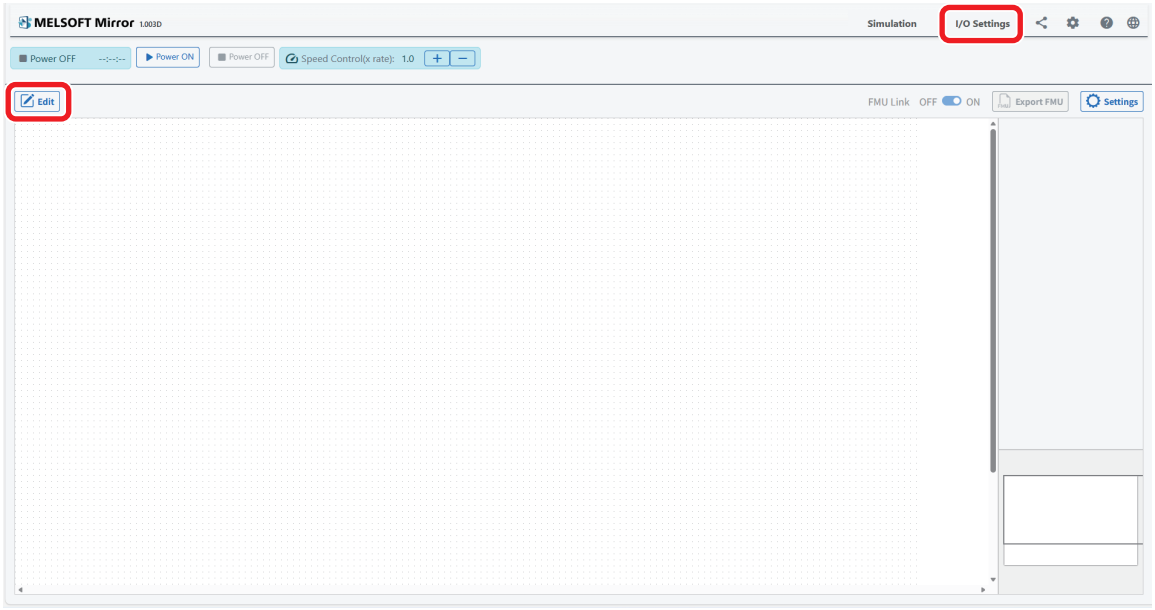
At this time, users can check that the Simple Motion module is operating by monitoring the changes of the values in the "Md.20: Command position value" and "Md.21: Machine feed value" on the axis monitor.

10 I/O SETTING

The I/O setting enables simulation of a system where an external device (such as a switch) is connected to the programmable controller.



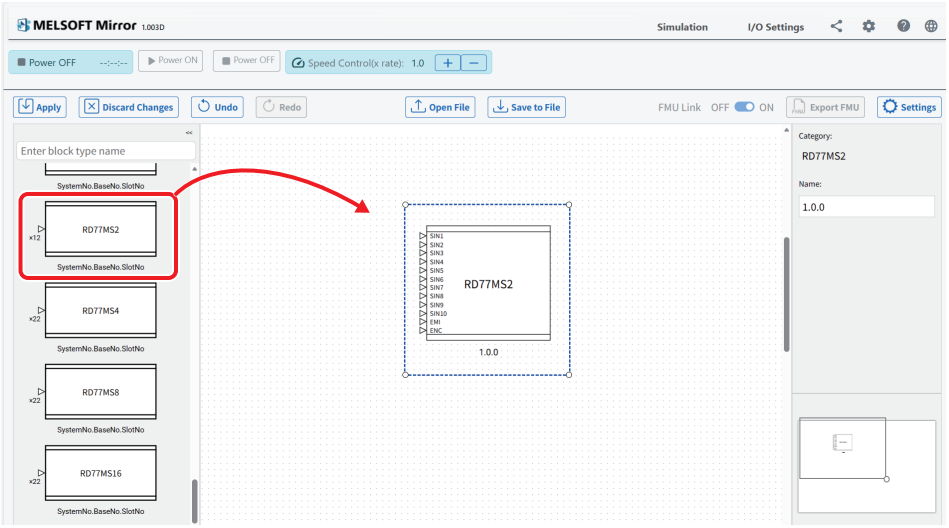
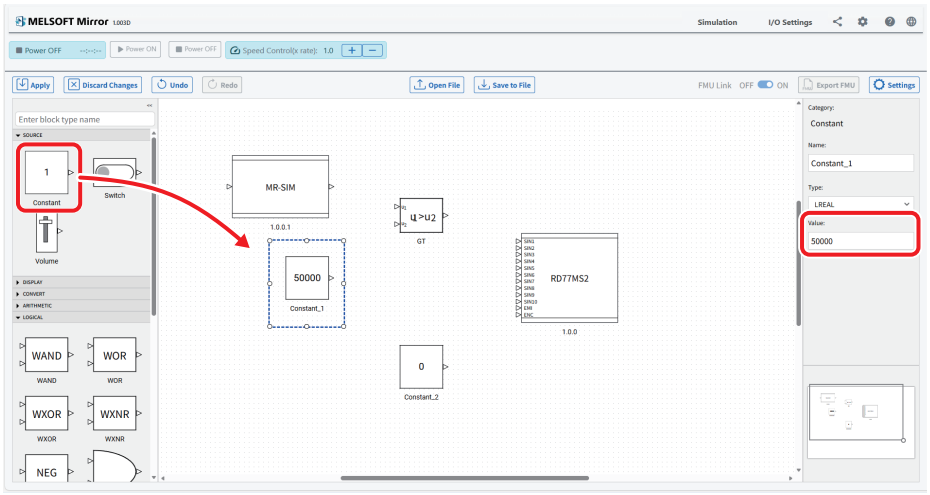
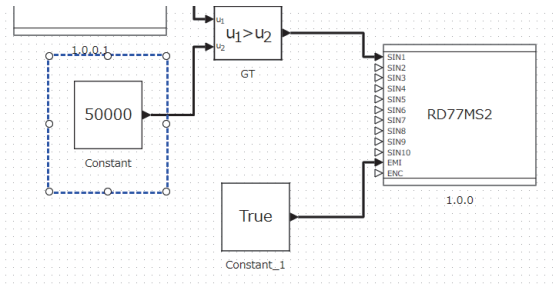
Click the [I/O Settings] button at the top right of the window to open the "I/O Settings" window.

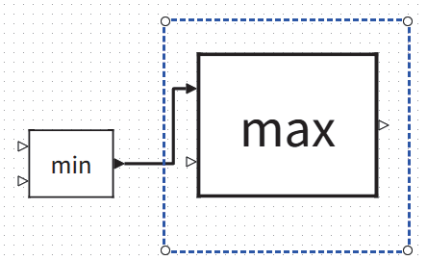
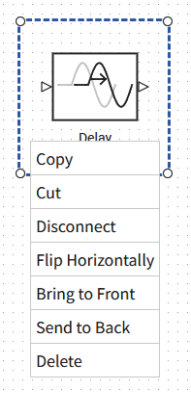
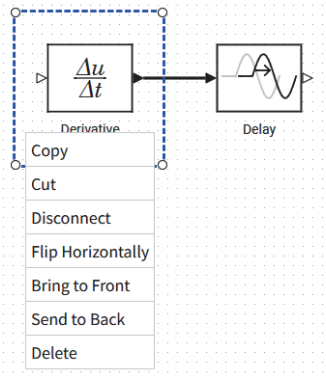


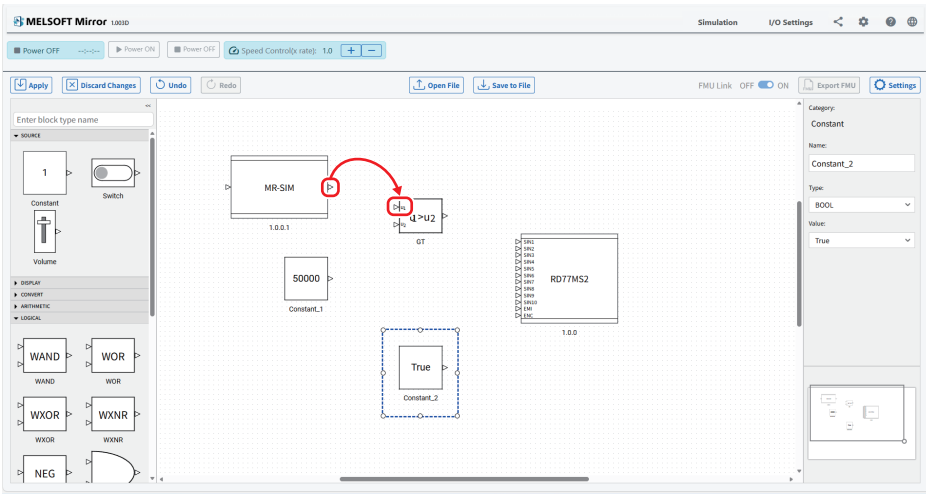
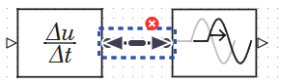
10.1 Block Diagram Explanation

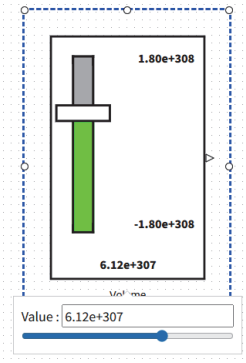
Block operation explanation

The following table lists the block operations.

Operation	Operation details	Remarks
Adding a block	Select a block from the block catalog, and drag and drop it into the block diagram area. 	If a block of the same type already exists, the default block name is automatically suffixed with "_[number]". The [number] increments. If there is a missing number in the existing suffixes, the missing number is used. Example: When Delay_1 and Delay_3 already exist, the newly added block is named Delay_2.
Editing a block (select)	Select a block on the block diagram to highlight it. The properties of the selected block are displayed on the right side of the window. Properties display block setting items such as the setting of block constants. With the block selected, edit the properties. Press Enter to reflect the changes. If the block in the block diagram area is deselected without pressing Enter, the changes are not reflected.  If a block is dragged while it is not selected, it becomes selected. Multiple blocks can also be selected. 	—
Editing a block (move)	Select and drag a block on the block diagram.	—

Operation	Operation details	Remarks
Editing a block (resize)	Select a block on the block diagram, and drag its bottom right corner to enlarge or reduce the block. Multiple blocks can be selected. 	• The terminals of a block are not enlarged or reduced. • The default size of constant block, volume block, gauge block, and display block is the minimum and cannot be reduced further. • The aspect ratio of constant block, volume block, gauge block, and display block can be changed when a block is resized.
Editing a block (invert left and right)	Select a block on the block diagram, right-click the block to display the context menu, and select "Invert Left and Right" from the menu. A selected block can also be inverted with a keyboard shortcut. 	Keyboard shortcut: Ctrl + I
Cutting a block	Select a block on the block diagram, right-click the block to display the context menu, and select [Cut] from the menu. A selected block can also be cut with a keyboard shortcut. 	Keyboard shortcut: Ctrl + X
Copying a block	Select a block on the block diagram, right-click the block to display the context menu, and select [Copy] from the menu. The copied block is temporarily stored in the clipboard. A selected block can also be copied with a keyboard shortcut.	Keyboard shortcut: Ctrl + C

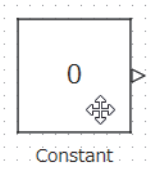
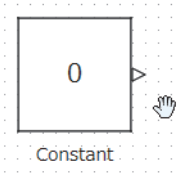
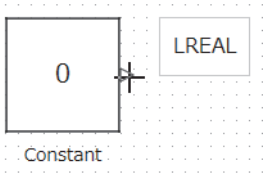
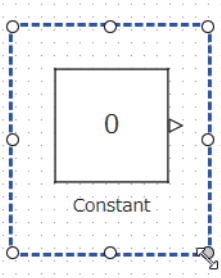
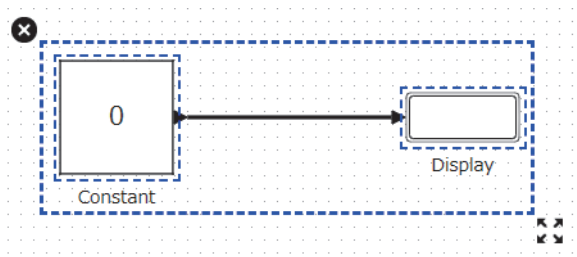
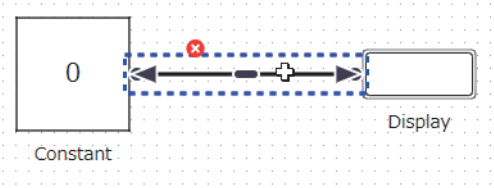
Operation	Operation details	Remarks
Pasting a block	After copying a block, select any position on the block diagram, right-click the block to display the context menu, and select [Paste] from the menu. Blocks that have been cut or copied can be repeatedly pasted. With any position on the block diagram selected, blocks can be pasted with a keyboard shortcut.	Keyboard shortcut: Ctrl + V • When pasting multiple selected blocks, their relative positions are preserved. • If the total number of the blocks already placed on the block diagram and the blocks to be newly pasted exceeds 100, an error window appears and the paste operation is canceled.
Deleting a block	Select a block on the block diagram, right-click the block to display the context menu, and select [Delete] from the menu. A selected block can also be deleted with Delete or Back space.	—
Connecting blocks	Select and drag a block terminal on the block diagram to display a connecting line. In this state, dragging the line to another block terminal highlights the connectable terminal*1 and connects the blocks. Blocks of the same type can be connected. Changing the type of one of the connected blocks disconnects the blocks. 	—
Editing the block connection (replace)	Select and drag the end of a connected line on the block diagram to change the destination terminal or disconnect the blocks.	—
Disconnecting blocks	Select a block on the block diagram, right-click the block to display the context menu, and select [Disconnect] from the menu. All lines connected to the terminals of the selected block are disconnected.	—
Deleting the block connection	Select a connected line on the block diagram, right-click the line to display the context menu, and select [Delete] from that menu. A selected line can be deleted by pressing the × button or by pressing Delete or Back space. 	—
Moving a block to the front	Select a block, right-click the block to display the context menu, and select [Bring to Front] from that menu. When multiple blocks are stacked on top of each other, the block for which [Bring to Front] was selected is displayed at the top of the stack.	—
Moving a block to the back	Select a block, right-click the block to display the context menu, and select [Send to Back] from the menu. When multiple blocks are stacked on top of each other, the block for which [Send to Back] was selected is displayed at the bottom of the stack.	—
Changing a switch	Double-click a switch block on the block diagram, and change the on/off state. The property values of the selected switch block can also be changed.	Valid in view mode/ edit mode

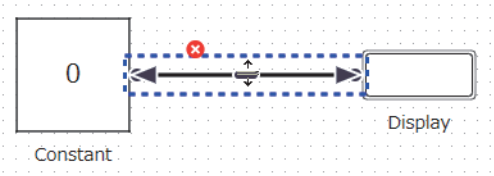
Operation	Operation details	Remarks
Changing a volume	Double-click a volume block on the block diagram, and change the value using the pop-up slider or direct input. The property values of the selected volume block can also be changed. 	Valid in view mode/ edit mode
Displaying a block (other than above)	Select a block on the block diagram to display its properties.	Valid in view mode/ edit mode
Changing a block (other than above)	Select a block on the block diagram to change the property values.	Valid in edit mode
Enlarging or reducing the block diagram	Enlarge or reduce the block diagram display. The operation can be performed by scrolling while holding down Alt, keyboard shortcuts, or selecting [Zoom In/ Zoom Out/100%] from the context menu.	Keyboard shortcuts: • Zoom in: Alt + ↑ • Zoom out: Alt + ↓ • Reset zoom to 100%: Alt + 1
Displaying the entire block diagram	Display all blocks in the center of the block diagram. The operation can be performed by using Space or selecting [Show Entire] from the context menu.	—
Changing the display area of the block diagram	Change the display area of the block diagram. Hover the mouse over the block diagram and use the hand tool, or use the scroll wheel to move up and down, and the horizontal scroll wheel to move left and right. The operation can also be performed by using ↑ ↓ ← →. Selecting [Move to Center] from the context menu adjusts the display area so that all added blocks are centered on the block diagram.	Valid in view mode/ edit mode
Enlarging or reducing the block diagram	Enlarge or reduce the block diagram display by dragging the lower right corner of the area block displayed on the mini map.	Valid in view mode/ edit mode
Changing the display area of the block diagram	Change the display area of the block diagram by dragging the area block displayed on the mini map.	Valid in view mode/ edit mode

*1 When connecting a line from an input terminal, this indicates the output terminal of another block of the same type. When connecting a line from an output terminal, this indicates the input terminal of another block of the same type.

Mouse cursor explanation

The following table lists operations of the mouse cursor.

Mouse cursor operation	Corresponding operation
 Mouse over a block.	• Moving a block • Pointing a connection point (▲) of the connecting line • Moving a bend point (●) of the connecting line
 Mouse outside a block.	• Moving the display range of the block diagram area • Moving a connection point (▲) of the connecting line
 Mouse over a block terminal.	• Displaying the block terminal type • Connecting the block terminal • Selecting multiple blocks (Shift) • Pointing the connected output terminal of the block
 After clicking a block once, mouse over the circle at corner.	Changing the size of a block (The mouse cursor may change while enlarging or reducing the block.)
 Shift + drag and drop to select a range.	• When multiple blocks are selected: Changing the size of multiple blocks • When multiple blocks are selected: Deleting the selected blocks • Selecting a connecting line • When a connecting line is selected: Deleting the selected connecting line
 After clicking a connecting line once, mouse over any area other than the center of the line.	Adding a bend point (●) of the connecting line

Mouse cursor operation	Corresponding operation
 After clicking a connecting line once, mouse over the center of the line.	Moving the line connecting the two bend points (●) of the line

Block naming rules

This section describes the block naming rules.

The following table lists the block names and the values that can be entered for names.

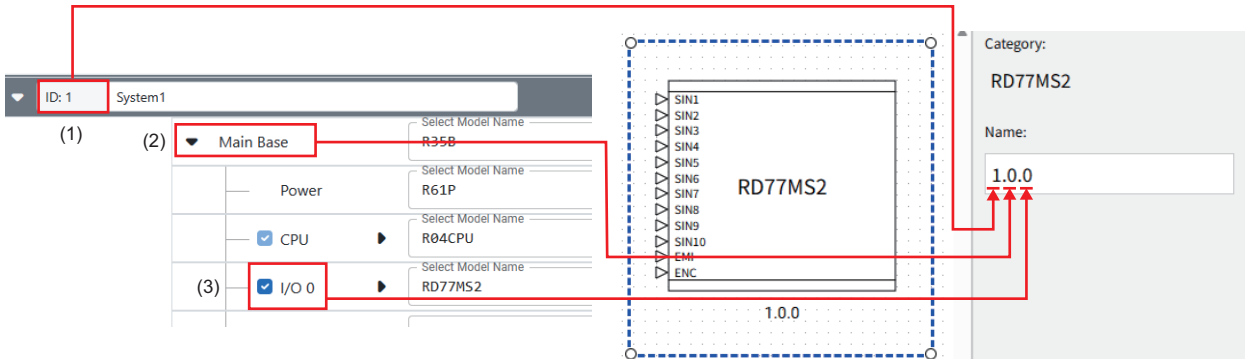
Block name	Possible input value	Default
Simple Motion block Input block Output block	One-byte numbers and dot (.)	1.0.0 For how to name a block, refer to the following. ☞ Page 125 Simple Motion, input, and output blocks
Servo amplifier block	One-byte numbers and dot (.)	1.0.0.1 For how to name a block, refer to the following. ☞ Page 126 Servo amplifier block
FMU input block	• Numbers (0 to 9) • Alphabetic characters (A to Z, a to z) • Symbols (-, ., _) Note that the dot (.) is prohibited.	FmuIn If a block name is already in use, follow the naming rules for adding blocks. ☞ Page 119 Block operation explanation
FMU output block	• Numbers (0 to 9) • Alphabetic characters (A to Z, a to z) • Symbols (-, ., _) Note that the dot (.) is prohibited.	FmuOut If a block name is already in use, follow the naming rules for adding blocks. ☞ Page 119 Block operation explanation
Blocks other than the above	• Numbers (0 to 9) • Alphabetic characters (A to Z, a to z) • Symbols (-, ., _)	Follow the display name on the block catalog.

Precautions

If the name of a controller block does not match the name set in the simulation setting window, an error will occur at power-on.

Simple Motion, input, and output blocks

Set the block name to match the simulation setting window using the format: "SystemNo.BaseNo.SlotNo".



No.	Item	Corresponding number	Setting range	Maximum number of digits to be displayed*1	Description
(1)	System number	Number of system ID	1 to 30	3	—
(2)	Base number	Number of main base or extension base	0 to 7	1	Set 0 for the main base. Set 1 to 7 for extension bases.
(3)	Slot number	Number that follows I/O	0 to 11	2	—

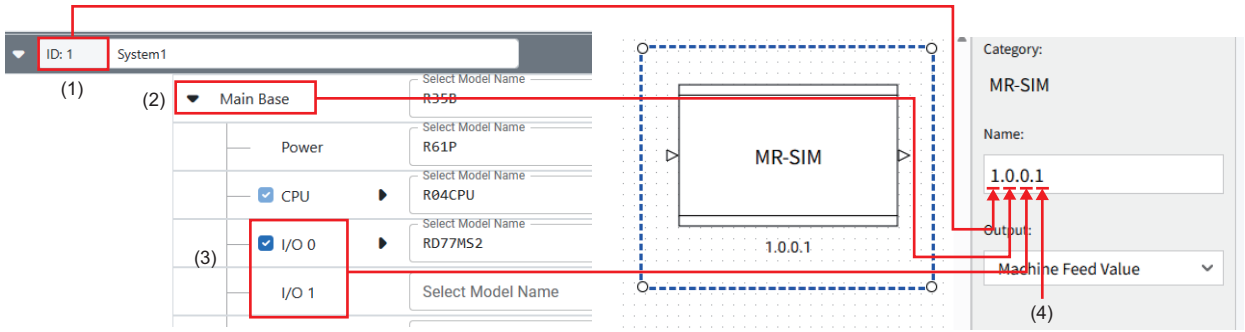
*1 Values must be less than or equal to the maximum number of digits to be displayed. For example, if the maximum number of digits to be displayed is 3, you can enter "1", "01", or "001" (pad with leading zeros up to the maximum).

Point

When adding a block, if the same block name already exists, the name for the new block is determined by incrementing values in the following priority order: slot number → base number → system number.

Servo amplifier block

Set the block name using the format: "SystemNo.BaseNo.SlotNo.AxisNo".



No.	Item	Corresponding number	Setting range	Maximum number of digits to be displayed*1	Description
(1)	System number	Number of system ID	1 to 30	3	Same number as the Simple Motion block
(2)	Base number	Number of main base or extension base	0 to 7	1	Same number as the Simple Motion block
(3)	Slot number	Number that follows I/O	0 to 11	2	Same number as the Simple Motion block
(4)	Axis number	—	1 to 16	3	—

*1 Values must be less than or equal to the maximum number of digits to be displayed. For example, if the maximum number of digits to be displayed is 3, you can enter "1", "01", or "001" (pad with leading zeros up to the maximum).

Point

When adding a block, if the same block name already exists, the name for the new block is determined by incrementing values in the following priority order: axis number → slot number → base number → system number.

Notation that can be used for properties

The following table lists the notation that can be used for properties.

Notation	Supported type	Example	Remarks
Decimal	LREAL, INT, WORD, DINT, DWORD	• 123.456 • 123	Only LREAL data type can enter values after the decimal point.
Hexadecimal	INT, WORD, DINT, DWORD	• 0x123 • 0x0a • 0x0A	When values are entered as a to f (lower-case character), they are converted to A to F (upper-case character).
Octal	INT, WORD, DINT, DWORD	• 0o123	—
Exponential	LREAL	• 123.4E+5 • 123.4e+5 • 1.E-5 • 1.e-5 • 1E5 • 1e5	—

10.2 Block List

The following tables list blocks.

Source

Block name	Description	Reference
Constant block	Always outputs the constant value set in the property.	☞ Page 132 Constant block
Switch block	Switches the on/off state like a switch.	☞ Page 133 Switch block
Volume block	Specifies a numerical value like in volume control.	☞ Page 134 Volume block

Display

Block name	Description	Reference
Lamp block	Turns on when a signal is input.	☞ Page 136 Lamp block
Gauge block	Outputs the input step by step.	☞ Page 137 Gauge block
Display block	Used to display the input value in the browser.	☞ Page 139 Display block
Text block	Used to enter text in the I/O Setting window.	☞ Page 140 Text block

Conversion

Block name	Description	Reference
TO_LREAL block	Converts the input value to LREAL data type and outputs the converted value.	☞ Page 141 TO_LREAL block
TO_INT block	Converts the input value to INT data type and outputs the converted value.	☞ Page 142 TO_INT block
TO_WORD block	Converts the input value to WORD data type and outputs the converted value.	☞ Page 143 TO_WORD block
TO_DINT block	Converts the input value to DINT data type and outputs the converted value.	☞ Page 144 TO_DINT block
TO_DWORD block	Converts the input value to DWORD data type and outputs the converted value.	☞ Page 145 TO_DWORD block
TO_BOOL block	Outputs FALSE if the input value is 0, and otherwise outputs TRUE.	☞ Page 146 TO_BOOL block
BCD_TO_WORD block	Converts the input value to a decimal value and outputs the converted value.	☞ Page 147 BCD_TO_WORD block
WORD_TO_BCD block	Converts the input value to BCD and outputs the converted value.	☞ Page 148 WORD_TO_BCD block
Digit-specified K4 block	Combines 16 BOOL data type input values and outputs the value converted to WORD data type.	☞ Page 149 Digit-specified K4 block
Bit decomposition (WORD) block	Converts the WORD data type input value to 16 BOOL data type input values and outputs the converted values.	☞ Page 150 Bit decomposition (WORD) block
Digit-specified K8 block	Combines 32 BOOL data type input values and outputs the value converted to DWORD data type.	☞ Page 151 Digit-specified K8 block
Bit decomposition (DWORD) block	Converts the DWORD data type input value to 32 BOOL data type input values and outputs the converted values.	☞ Page 152 Bit decomposition (DWORD) block

Arithmetic operation

Block name	Description	Reference
Add block	Adds two input values and outputs the result.	☞ Page 153 Add block
Sub block	Subtracts the second input value from the first input value and outputs the result.	☞ Page 154 Sub block
Mul block	Multiplies two input values and outputs the result.	☞ Page 155 Mul block
Div block	Divides the first input value by the second input value and outputs the result.	☞ Page 156 Div block
Abs block	Outputs the absolute value of the input value.	☞ Page 157 Abs block
Min block	Compares two input values and outputs the smaller value.	☞ Page 158 Min block
Max block	Compares two input values and outputs the larger value.	☞ Page 159 Max block
Round block	Rounds off the input value and outputs the result.	☞ Page 160 Round block
Floor block	Rounds down the input value and outputs the result.	☞ Page 161 Floor block
Ceil block	Rounds up the input value and outputs the result.	☞ Page 162 Ceil block
Sqrt block	Outputs the square root of the input value.	☞ Page 163 Sqrt block
Sin block	Outputs the sine of the input value.	☞ Page 164 Sin block
Cos block	Outputs the cosine of the input value.	☞ Page 165 Cos block
Tan block	Outputs the tangent of the input value.	☞ Page 166 Tan block
Asin block	Outputs the arcsine of the input value.	☞ Page 167 Asin block
Acos block	Outputs the arccosine of the input value.	☞ Page 168 Acos block
Atan block	Outputs the arctangent of the input value.	☞ Page 169 Atan block
Atan2 block	Outputs the arctangent of the division of two input values.	☞ Page 170 Atan2 block
Sinh block	Outputs the hyperbolic sine of the input value.	☞ Page 171 Sinh block
Cosh block	Outputs the hyperbolic cosine of the input value.	☞ Page 172 Cosh block
Tanh block	Outputs the hyperbolic tangent of the input value.	☞ Page 173 Tanh block
Exp block	Outputs the natural exponential function of the input value.	☞ Page 174 Exp block
Log block	Outputs the natural logarithm of the input value.	☞ Page 175 Log block
Log10 block	Outputs the common logarithm of the input value.	☞ Page 176 Log10 block

Logical operation

Block name	Description	Reference
WAND block	Outputs the logical AND of two input values in units of bits.	☞ Page 177 WAND block
WOR block	Outputs the logical OR of two input values in units of bits.	☞ Page 177 WOR block
WXOR block	Outputs the exclusive OR of two input values in units of bits.	☞ Page 178 WXOR block
WXNR block	Outputs the exclusive NOR of two input values in units of bits.	☞ Page 178 WXNR block
NEG block	Outputs the logical NOT of the input value in units of bits.	☞ Page 179 NEG block
AND block	Outputs the logical AND of two input values.	☞ Page 179 AND block
OR block	Outputs the logical OR of two input values.	☞ Page 180 OR block
NOT block	Outputs the logical NOT of the input value.	☞ Page 180 NOT block
EQ block	Compares two input values and outputs TRUE if they are equal, and FALSE if they are not equal.	☞ Page 181 EQ block
NE block	Compares two input values and outputs FALSE if they are equal, and TRUE if they are not equal.	☞ Page 182 NE block
LT block	Compares two input values and outputs TRUE if the first input value is less than the second input value, and otherwise outputs FALSE.	☞ Page 183 LT block
GT block	Compares two input values and outputs TRUE if the first input value is greater than the second input value, and otherwise outputs FALSE.	☞ Page 184 GT block
LE block	Compares two input values and outputs TRUE if the first input value is less than or equal to the second input value, and otherwise outputs FALSE.	☞ Page 185 LE block
GE block	Compares two input values and outputs TRUE if the first input value is greater than or equal to the second input value, and otherwise outputs FALSE.	☞ Page 186 GE block

FMU

Block name	Description	Reference
FMU input block	Defines the data to be input to MELSOFT Mirror as an input terminal in a FMU exported using the FMU export function.	☞ Page 187 FMU input block
FMU output block	Defines the data to be output from MELSOFT Mirror as an output terminal in a FMU exported using the FMU export function.	☞ Page 191 FMU output block

Continuous time

Block name	Description	Reference
Integral block	Outputs the integral value calculated using the specified formula.	☞ Page 194 Integral block
Differential block	Outputs the derivative value calculated using the specified formula.	☞ Page 196 Differential block
Delay block	Outputs values input before the delay time after the delay time set in the property is exceeded.	☞ Page 197 Delay block
Primary delay block	Calculates the specified differential equation and outputs the result.	☞ Page 199 Primary delay block
Secondary delay block	Calculates the specified differential equation and outputs the result.	☞ Page 201 Secondary delay block
Backlash block	Outputs the value calculated using the specified formula.	☞ Page 203 Backlash block
Dead band block	Outputs the value calculated using the specified formula.	☞ Page 205 Dead band block
Quantization block	Outputs the value calculated using the specified formula.	☞ Page 207 Quantization block
Rate limiter block	Outputs the value calculated using the specified formula.	☞ Page 209 Rate limiter block
Saturation block	Outputs the value calculated using the specified formula.	☞ Page 211 Saturation block

Discrete time

Block name	Description	Reference
Primary hold block	Interpolates inputs and outputs between blocks with different sample times.	☞ Page 213 Primary hold block

Controller

Block name	Description	Reference
Input block	Inputs signals to the input module.	☞ Page 215 Input block
Output block	Outputs signals to the output module.	☞ Page 218 Output block
Simple Motion block	Sets the external inputs of the Simple Motion module and reflects them in the simulation.	☞ Page 221 Simple Motion block (RD77MS2)
Servo amplifier	Inputs the synchronous encoder of the servo amplifier and monitors the current value of the servo amplifier input axis.	☞ Page 225 Servo amplifier block

10.3 Block Details

This section describes details of each block.

The following table is used to describe details of each block.

Signal name	Classification	Input order	Type	Range	Remarks
A signal name	A signal type (input or output)	The processing order of input values in a block that requires two or more input signals	A data type of the input/output value	An acceptable range of the input/output value. An error occurs if the range is exceeded.	Additional information, such as block behavior

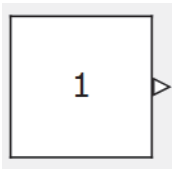
The following are the range of each data type.

Data type	Minimum value	Maximum value
LREAL	-1.7e+308	1.7e+308
INT	-32768	32767
WORD	0	65535
DINT	-2147483648	2147483647
DWORD	0	4294967295
BOOL	FALSE	TRUE

Up to 100 blocks can be configured.

Source

Constant block



■Function details

This block always outputs the constant value set in the property. (☞ Page 132 Properties)

■I/O signals

The following table lists the I/O signals of the constant block.

Signal name	Classification	Input/output order	Type	Range	Remarks
output	Output	1	LREAL WORD INT DWORD DINT BOOL	Minimum value to maximum value of each data type	—

■Properties

The following table lists the properties of the constant block.

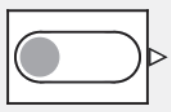
Property name	Attribute	Range/option	Default	Possible input value	Remarks
Type	Drop-down list	• LREAL • WORD • INT • DWORD • DINT • BOOL	LREAL	—	—
Value	• Text box • Drop-down list	Minimum value to maximum value of each data type	FALSE for BOOL data type, and otherwise 0	■LREAL data type • One-byte numbers (0 to 9) • One-byte alphabetic characters (E, e) • One-byte symbols (plus (+), hyphen (-), period (.)) ■BOOL data type • FALSE, TRUE ■Other data types • One-byte numbers (0 to 9) • One-byte alphabetic characters (A to F, a to f, o, x) • One-byte symbols (plus (+), hyphen (-))	• Up to 13 characters can be entered. • For BOOL data type, a drop-down list is used. For other data types, a text box is used. • For the notation that can be used, refer to the following. ☞ Page 127 Notation that can be used for properties

■Error

The following table lists errors that occur in the constant block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	—	—	—
0 division	—	—	—

Switch block



■Function details

This block switches the on/off state like a switch.

The on/off state can be switched by double-clicking the block on the browser.

The block outputs TRUE when set to on. The block outputs FALSE when set to off.

■I/O signals

The following table lists the I/O signals of the switch block.

Signal name	Classification	Input/output order	Type	Range	Remarks
output	Output	1	BOOL	FALSE, TRUE	The block outputs TRUE when set to on. The block outputs FALSE when set to off.

■Properties

The following table lists the properties of the switch block.

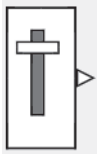
Property name	Attribute	Range/option	Default	Possible input value	Remarks
Value	Drop-down list	• Off • On	Off	—	—

■Error

The following table lists errors that occur in the switch block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	—	—	—
0 division	—	—	—

Volume block



■Function details

This block specifies a numerical value like in volume control.

A numerical value can be specified by dragging the mouse on the browser.

A numerical value can also be entered directly.

■I/O signals

The following table lists the I/O signals of the volume block.

Signal name	Classification	Input/output order	Type	Range	Remarks
output	Output	1	LREAL WORD INT DWORD DINT	Minimum value set and maximum value set	—

■Properties

The following table lists the properties of the volume block.

Property name	Attribute	Range/option	Default	Possible input value	Remarks
Type	Drop-down list	• LREAL • WORD • INT • DWORD • DINT	LREAL	—	—
Minimum	Text box	Minimum value of each data type to Maximum	0	■LREAL data type • One-byte numbers (0 to 9) • One-byte alphabetic characters (E, e) • One-byte symbols (plus (+), hyphen (-), period (.)) ■Other data types • One-byte numbers (0 to 9) • One-byte alphabetic characters (A to F, a to f, o, x) • One-byte symbols (plus (+), hyphen (-))	• Up to 13 characters can be entered. • When the data type is changed, the Minimum field is set to the minimum value of each data type. • For the notation that can be used, refer to the following. ☞ Page 127 Notation that can be used for properties
Maximum	Text box	Minimum to maximum value of each data type	Maximum value of each data type	■LREAL data type • One-byte numbers (0 to 9) • One-byte alphabetic characters (E, e) • One-byte symbols (plus (+), hyphen (-), period (.)) ■Other data types • One-byte numbers (0 to 9) • One-byte alphabetic characters (A to F, a to f, o, x) • One-byte symbols (plus (+), hyphen (-))	• Up to 13 characters can be entered. • For the notation that can be used, refer to the following. ☞ Page 127 Notation that can be used for properties

Property name	Attribute	Range/option	Default	Possible input value	Remarks
Value	Text box	Minimum to Maximum	0	■LREAL data type • One-byte numbers (0 to 9) • One-byte alphabetic characters (E, e) • One-byte symbols (plus (+), hyphen (-), period (.)) ■Other data types • One-byte numbers (0 to 9) • One-byte alphabetic characters (A to F, a to f, o, x) • One-byte symbols (plus (+), hyphen (-))	• Up to 13 characters can be entered. • If the value is less than the minimum after changing the minimum, the value will become the minimum after the change. • If the value exceeds the maximum after changing the maximum, the value will become the maximum after the change. • If the value is changed with the slider, it will be displayed as a decimal number. • For the notation that can be used, refer to the following. ☞ Page 127 Notation that can be used for properties
Number of digits to be displayed after the decimal point	Text box	0 to 3	2 for LREAL data type, and otherwise 0	One-byte numbers	• Up to 13 characters can be entered. • The setting value is displayed within the block. • Only LREAL data type can display values after the decimal point. Other data types are integer values and do not have values after the decimal point.

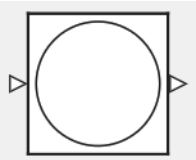
■Error

The following table lists errors that occur in the volume block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	—	—	—
0 division	—	—	—

Display

Lamp block



■Function details

This block turns on when a signal is input.

If the input is TRUE, the block is displayed (turns on) in the color set in the property. (☞ Page 136 Properties)

If the input is FALSE, the block is grayed out (turns off).

■I/O signals

The following table lists the I/O signals of the lamp block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input	Input	1	BOOL	FALSE, TRUE	—
output	Output	1	BOOL	FALSE, TRUE	—

■Properties

The following table lists the properties of the lamp block.

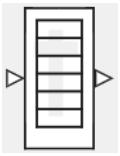
Property name	Attribute	Range/option	Default	Possible input value	Remarks
Value	Drop-down list	• Red • Yellow • Green	Green	—	The off color is gray.

■Error

The following table lists errors that occur in the lamp block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	—	—	—
0 division	—	—	—

Gauge block



■Function details

This block outputs the input step by step.

The input value is displayed step by step in accordance with the number of steps set in the property. (☞ Page 137 Properties)

If the input value is less than the minimum value set in the property, the number of steps will be 0. If the input value exceeds the maximum value set in the property, the set number of steps will be applied.

■I/O signals

The following table lists the I/O signals of the gauge block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input	Input	1	LREAL WORD INT DWORD DINT	Minimum value to maximum value of each data type	—
output	Output	1	LREAL WORD INT DWORD DINT	Minimum value to maximum value of each data type	—

■Properties

The following table lists the properties of the gauge block.

Property name	Attribute	Range/option	Default	Possible input value	Remarks
Type	Drop-down list	• LREAL • WORD • INT • DWORD • DINT	LREAL	—	—
Minimum	Text box	Minimum value of each data type to Maximum	0	■LREAL data type • One-byte numbers (0 to 9) • One-byte alphabetic characters (E, e) • One-byte symbols (plus (+), hyphen (-), period (.)) ■Other data types • One-byte numbers (0 to 9) • One-byte alphabetic characters (A to F, a to f, o, x) • One-byte symbols (plus (+), hyphen (-))	• Up to 13 characters can be entered. • For the notation that can be used, refer to the following. ☞ Page 127 Notation that can be used for properties

Property name	Attribute	Range/option	Default	Possible input value	Remarks
Maximum	Text box	Minimum to maximum value of each data type	Maximum value of each data type	■LREAL data type • One-byte numbers (0 to 9) • One-byte alphabetic characters (E, e) • One-byte symbols (plus (+), hyphen (-), period (.)) ■Other data types • One-byte numbers (0 to 9) • One-byte alphabetic characters (A to F, a to f, o, x) • One-byte symbols (plus (+), hyphen (-))	• Up to 13 characters can be entered. • For the notation that can be used, refer to the following. ☞ Page 127 Notation that can be used for properties
Number of steps	Numerical input	1 to 16	10	One-byte numbers	Up to 13 characters can be entered.
Number of digits to be displayed after the decimal point	Text box	0 to 3	2 for LREAL data type, and otherwise 0	One-byte numbers	• Up to 13 characters can be entered. • The setting value is displayed within the block. • Only LREAL data type can display values after the decimal point. Other data types are integer values and do not have values after the decimal point.
Display orientation	Drop-down list	• Vertical • Horizontal	Vertical	—	—

■Error

The following table lists errors that occur in the gauge block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	—	—	—
0 division	—	—	—

Display block

■Function details

This block is used to display the input value in the browser. (The block does not output the value.)

For LREAL data type, the input value is displayed in accordance with the number of digits to be displayed after the decimal point set in the property. (☞ Page 139 Properties)

If the value has less than the specified number of digits, 0 will be added. If the value has more than the specified number of digits, the value will be rounded off.

Ex.

When the number of digits to be displayed after the decimal point is 2

- Input value: 0.3 → 0.30
- Input value: 0.329 → 0.33

■I/O signals

The following table lists the I/O signals of the display block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input	Input	1	LREAL WORD INT DWORD DINT	Minimum value to maximum value of each data type	—

■Properties

The following table lists the properties of the display block.

Property name	Attribute	Range/option	Default	Possible input value	Remarks
Type	Drop-down list	• LREAL • WORD • INT • DWORD • DINT	LREAL	—	—
Number of digits to be displayed after the decimal point	Text box	0 to 5	2 for LREAL data type, and otherwise 0	One-byte numbers	• Up to 13 characters can be entered. • The setting value is displayed within the block. • Only LREAL data type can display values after the decimal point. Other data types are integer values and do not have values after the decimal point.

■Error

The following table lists errors that occur in the display block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	—	—	—
0 division	—	—	—

Text block

■Function details

This block is used to enter text in the I/O Settings window.

Up to 100 characters can be entered. Press Shift + Enter to start a new line while entering text. Press Enter to complete the text entry.

The text set in the property is displayed in the browser. (☞ Page 140 Properties)

■I/O signals

There are no I/O signals in the text block.

■Properties

The following table lists the properties of the text block.

Property name	Attribute	Range/option	Default	Possible input value	Remarks
Text	Text box	—	—	• One-byte numbers (0 to 9) • One-byte alphabetic characters (A to Z, a to z) • One-byte symbols (!, ", #, $, %, &, (,), *, +, ,, -, ., /, :, ;, <, =, >, ?, @, [, \,], ^, _, ', {, |, }, ~) • Two-byte symbols (excluding vertical tabs (\v))	• Up to 100 characters can be entered. • Press Shift + Enter to start a new line while entering text. • Press Enter to complete the text entry.

■Error

The following table lists errors that occur in the text block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	—	—	—
0 division	—	—	—

Conversion

TO_LREAL block

■Function details

This block converts the input value to LREAL data type and outputs the converted value.

The data type of input values is set in the property. (☞ Page 141 Properties)

■I/O signals

The following table lists the I/O signals of the TO_LREAL block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input	Input	1	WORD INT DWORD DINT	Minimum value to maximum value of each data type	—
output	Output	1	LREAL	Minimum value to maximum value of LREAL data type	—

■Properties

The following table lists the properties of the TO_LREAL block.

Property name	Attribute	Range/option	Default	Possible input value	Remarks
Type	Drop-down list	• WORD • INT • DWORD • DINT	WORD	—	—

■Error

The following table lists errors that occur in the TO_LREAL block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	—	—	—
0 division	—	—	—

TO_INT block

■Function details

This block converts the input value to INT data type and outputs the converted value.

The data type of input values is set in the property. (☞ Page 142 Properties)

If the input value is LREAL data type, values below the decimal point of the input value are rounded down. If the input value exceeds the maximum value of INT data type, the maximum value of INT data type is output. If the input value is less than the minimum value of INT data type, the minimum value of INT data type is output.

■I/O signals

The following table lists the I/O signals of the TO_INT block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input	Input	1	LREAL WORD DWORD DINT BOOL	Minimum value to maximum value of each data type	—
output	Output	1	INT	Minimum value to maximum value of INT data type	—

■Properties

The following table lists the properties of the TO_INT block.

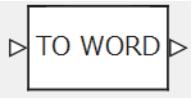
Property name	Attribute	Range/option	Default	Possible input value	Remarks
Type	Drop-down list	• LREAL • WORD • DWORD • DINT • BOOL	LREAL	—	—

■Error

The following table lists errors that occur in the TO_INT block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	—	—	—
0 division	—	—	—

TO_WORD block



■Function details

This block converts the input value to WORD data type and outputs the converted value.

The data type of input values is set in the property. (☞ Page 143 Properties)

If the input value is LREAL data type, values below the decimal point of the input value are rounded down. If the input value exceeds the maximum value of WORD data type, the maximum value of WORD data type is output. If the input value is less than the minimum value of WORD data type, the minimum value of WORD data type is output.

■I/O signals

The following table lists the I/O signals of the TO_WORD block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input	Input	1	LREAL INT DWORD DINT BOOL	Minimum value to maximum value of each data type	—
output	Output	1	WORD	Minimum value to maximum value of WORD data type	—

■Properties

The following table lists the properties of the TO_WORD block.

Property name	Attribute	Range/option	Default	Possible input value	Remarks
Type	Drop-down list	• LREAL • INT • DWORD • DINT • BOOL	LREAL	—	—

■Error

The following table lists errors that occur in the TO_WORD block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	—	—	—
0 division	—	—	—

TO_DINT block

■Function details

This block converts the input value to DINT data type and outputs the converted value.

The data type of input values is set in the property. (☞ Page 144 Properties)

If the input value is LREAL data type, values below the decimal point of the input value are rounded down. If the input value exceeds the maximum value of DINT data type, the maximum value of DINT data type is output. If the input value is less than the minimum value of DINT data type, the minimum value of DINT data type is output.

■I/O signals

The following table lists the I/O signals of the TO_DINT block.

Signal name	Classification	Input order	Type	Range	Remarks
input	Input	1	LREAL WORD INT DWORD BOOL	Minimum value to maximum value of each data type	—
output	Output	1	DINT	Minimum value to maximum value of DINT data type	—

■Properties

The following table lists the properties of the TO_DINT block.

Property name	Attribute	Range/option	Default	Possible input value	Remarks
Type	Drop-down list	• LREAL • WORD • INT • DWORD • BOOL	LREAL	—	—

■Error

The following table lists errors that occur in the TO_DINT block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	—	—	—
0 division	—	—	—

TO_DWORD block

■Function details

This block converts the input value to DWORD data type and outputs the converted value.

The data type of input values is set in the property. (☞ Page 145 Properties)

If the input value is LREAL data type, values below the decimal point of the input value are rounded down. If the input value exceeds the maximum value of DWORD data type, the maximum value of DWORD data type is output. If the input value is less than the minimum value of DWORD data type, the minimum value of DWORD data type is output.

■I/O signals

The following table lists the I/O signals of the TO_DWORD block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input	Input	1	LREAL WORD INT DINT BOOL	Minimum value to maximum value of each data type	—
output	Output	1	DWORD	Minimum value to maximum value of DWORD data type	—

■Properties

The following table lists the properties of the TO_DWORD block.

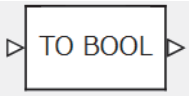
Property name	Attribute	Range/option	Default	Possible input value	Remarks
Type	Drop-down list	• LREAL • WORD • INT • DINT • BOOL	LREAL	—	—

■Error

The following table lists errors that occur in the TO_DWORD block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	—	—	—
0 division	—	—	—

TO_BOOL block



■Function details

This block outputs FALSE if the input value is 0, and otherwise outputs TRUE.

The data type of input values is set in the property. (☞ Page 146 Properties)

■I/O signals

The following table lists the I/O signals of the TO_BOOL block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input	Input	1	WORD DWORD	Minimum value to maximum value of each data type	—
output	Output	1	BOOL	FALSE, TRUE	—

■Properties

The following table lists the properties of the TO_BOOL block.

Property name	Attribute	Range/option	Default	Possible input value	Remarks
Type	Drop-down list	• WORD • DWORD	WORD	—	—

■Error

The following table lists errors that occur in the TO_BOOL block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	—	—	—
0 division	—	—	—

BCD_TO_WORD block

■Function details

This block recognizes the input value as BCD and outputs it as a WORD value.

■I/O signals

The following table lists the I/O signals of the BCD_TO_WORD block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input	Input	1	WORD	A value from 0 to 39321 that can be recognized as BCD	—
output	Output	1	WORD	0 to 9999	—

■Properties

The BCD_TO_WORD block has no properties.

■Error

The following table lists errors that occur in the BCD_TO_WORD block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	○	When a value that cannot be recognized as BCD is input	—
Output overflow	—	—	—
0 division	—	—	—

WORD_TO_BCD block

■Function details

This block converts the input value to BCD and outputs the converted value.

■I/O signals

The following table lists the I/O signals of the WORD_TO_BCD block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input	Input	1	WORD	0 to 9999	—
output	Output	1	WORD	0 to 39321	—

■Properties

The WORD_TO_BCD block has no properties.

■Error

The following table lists errors that occur in the WORD_TO_BCD block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	○	—	—
Output overflow	—	—	—
0 division	—	—	—

Digit-specified K4 block

■Function details

This block combines 16 BOOL data type input values and outputs the value converted to WORD data type. The highest input terminal value is assigned to the least significant bit. Then, the next terminal value is assigned to the next bit in sequence.

■I/O signals

The following table lists the I/O signals of the digit-specified K4 block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input	Input	1	BOOL×16	FALSE, TRUE	—
output	Output	1	WORD	Minimum value to maximum value of WORD data type	—

■Properties

The digit-specified K4 block has no properties.

■Error

The following table lists errors that occur in the digit-specified K4 block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	—	—	—
0 division	—	—	—

Bit decomposition (WORD) block

■Function details

This block decomposes a WORD data type input value into 16 units and outputs the value converted to BOOL data type. The highest input terminal value is assigned to the least significant bit. Then, the next terminal value is assigned to the next bit in sequence.

■I/O signals

The following table lists the I/O signals of the bit decomposition (WORD) block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input	Input	1	WORD	Minimum value to maximum value of WORD data type	—
output	Output	1	BOOL×16	FALSE, TRUE	—

■Properties

The bit decomposition (WORD) block has no properties.

■Error

The following table lists errors that occur in the bit decomposition (WORD) block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	—	—	—
0 division	—	—	—

Digit-specified K8 block

■Function details

This block combines 32 BOOL data type input values and outputs the value converted to DWORD type. The highest input terminal value is assigned to the least significant bit. Then, the next terminal value is assigned to the next bit in sequence.

■I/O signals

The following table lists the I/O signals of the digit-specified K8 block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input	Input	1	BOOL×32	FALSE, TRUE	—
output	Output	1	DWORD	Minimum value to maximum value of DWORD data type	—

■Properties

The digit-specified K8 block has no properties.

■Error

The following table lists errors that occur in the digit-specified K8 block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	—	—	—
0 division	—	—	—

Bit decomposition (DWORD) block

■Function details

This block decomposes a DWORD data type input value into 32 units and outputs the value converted to BOOL data type. The highest input terminal value is assigned to the least significant bit. Then, the next terminal value is assigned to the next bit in sequence.

■I/O signals

The following table lists the I/O signals of the bit decomposition (DWORD) block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input	Input	1	DWORD	Minimum value to maximum value of DWORD data type	—
output	Output	1	BOOL×32	FALSE, TRUE	—

■Properties

The bit decomposition (DWORD) block has no properties.

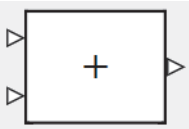
■Error

The following table lists errors that occur in the bit decomposition (DWORD) block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	—	—	—
0 division	—	—	—

Arithmetic operation

Add block



■Function details

This block adds two input values by using the formula below and outputs the result.

$$output = input1 + input2$$

- output: Output value
- input1: First input value
- input2: Second input value

The data type of input/ output values is set in the property. (☞ Page 153 Properties)

■I/O signals

The following table lists the I/O signals of the Add block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input1	Input	1	LREAL WORD INT DWORD DINT	Minimum value to maximum value of each data type	—
input2	Input	2	LREAL WORD INT DWORD DINT	Minimum value to maximum value of each data type	—
output	Output	1	LREAL WORD INT DWORD DINT	Minimum value to maximum value of each data type	—

■Properties

The following table lists the properties of the Add block.

Property name	Attribute	Range/option	Default	Possible input value	Remarks
Type	Drop-down list	• LREAL • WORD • INT • DWORD • DINT	LREAL	—	—

■Error

The following table lists errors that occur in the Add block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	○	When the data type is LREAL and output is outside the following range $\|output\| < 2^{1024}$	—
0 division	—	—	—

Sub block

■Function details

This block subtracts two input values by using the formula below and outputs the result.

$$output = input1 - input2$$

- output: Output value
- input1: First input value
- input2: Second input value

The data type of input/ output values is set in the property. (☞ Page 154 Properties)

■I/O signals

The following table lists the I/O signals of the Sub block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input1	Input	1	LREAL WORD INT DWORD DINT	Minimum value to maximum value of each data type	—
input2	Input	2	LREAL WORD INT DWORD DINT	Minimum value to maximum value of each data type	—
output	Output	1	LREAL WORD INT DWORD DINT	Minimum value to maximum value of each data type	—

■Properties

The following table lists the properties of the Sub block.

Property name	Attribute	Range/option	Default	Possible input value	Remarks
Type	Drop-down list	• LREAL • WORD • INT • DWORD • DINT	LREAL	—	—

■Error

The following table lists errors that occur in the Sub block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	○	When the data type is LREAL and output is outside the following range $\|output\| < 2^{1024}$	—
0 division	—	—	—

Mul block

■Function details

This block multiplies two input values by using the formula below and outputs the result.

$$output = input1 \times input2$$

- output: Output value
- input1: First input value
- input2: Second input value

The data type of input/ output values is set in the property. (☞ Page 155 Properties)

■I/O signals

The following table lists the I/O signals of the Mul block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input1	Input	1	LREAL WORD INT DWORD DINT	Minimum value to maximum value of each data type	—
input2	Input	2	LREAL WORD INT DWORD DINT	Minimum value to maximum value of each data type	—
output	Output	1	LREAL WORD INT DWORD DINT	Minimum value to maximum value of each data type	—

■Properties

The following table lists the properties of the Mul block.

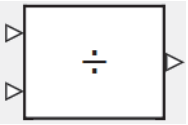
Property name	Attribute	Range/option	Default	Possible input value	Remarks
Type	Drop-down list	• LREAL • WORD • INT • DWORD • DINT	LREAL	—	—

■Error

The following table lists errors that occur in the Mul block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	○	When the data type is LREAL and output is outside the following range $\|output\| < 2^{1024}$	—
0 division	—	—	—

Div block



■Function details

This block divides two input values by using the formula below and outputs the result.

$$output = input1 / input2$$

- output: Output value
- input1: First input value
- input2: Second input value

The data type of input/ output values is set in the property. (☞ Page 156 Properties)

■I/O signals

The following table lists the I/O signals of the Div block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input1	Input	1	LREAL WORD INT DWORD DINT	Minimum value to maximum value of each data type	—
input2	Input	2	LREAL WORD INT DWORD DINT	Minimum value to maximum value of each data type	—
output	Output	1	LREAL WORD INT DWORD DINT	Minimum value to maximum value of each data type	—

■Properties

The following table lists the properties of the Div block.

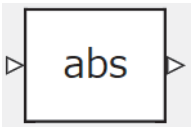
Property name	Attribute	Range/option	Default	Possible input value	Remarks
Type	Drop-down list	• LREAL • WORD • INT • DWORD • DINT	LREAL	—	—

■Error

The following table lists errors that occur in the Div block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	○	When the data type is LREAL and output is outside the following range $\|output\| < 2^{1024}$	—
0 division	○	When the following is satisfied input2 = 0	—

Abs block



■Function details

This block calculates and outputs the absolute value of the input value.

The data type of input/ output values is set in the property. (☞ Page 157 Properties)

For INT and DINT data types, if the minimum value of each data type (INT type: -32768, DINT type: -2147483648) is input, the same value as the input value is output.

■I/O signals

The following table lists the I/O signals of the Abs block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input	Input	1	LREAL WORD INT DWORD DINT	Minimum value to maximum value of each data type	—
output	Output	1	LREAL WORD INT DWORD DINT	Minimum value to maximum value of each data type	—

■Properties

The following table lists the properties of the Abs block.

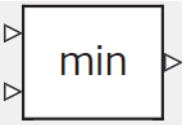
Property name	Attribute	Range/option	Default	Possible input value	Remarks
Type	Drop-down list	• LREAL • WORD • INT • DWORD • DINT	LREAL	—	—

■Error

The following table lists errors that occur in the Abs block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	—	—	—
0 division	—	—	—

Min block



■Function details

This block compares two input values and outputs the smaller value.

The data type of input/ output values is set in the property. (☞ Page 158 Properties)

■I/O signals

The following table lists the I/O signals of the Min block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input1	Input	1	LREAL WORD INT DWORD DINT	Minimum value to maximum value of each data type	—
input2	Input	2	LREAL WORD INT DWORD DINT	Minimum value to maximum value of each data type	—
output	Output	1	LREAL WORD INT DWORD DINT	Minimum value to maximum value of each data type	—

■Properties

The following table lists the properties of the Min block.

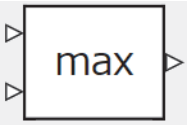
Property name	Attribute	Range/option	Default	Possible input value	Remarks
Type	Drop-down list	• LREAL • WORD • INT • DWORD • DINT	LREAL	—	—

■Error

The following table lists errors that occur in the Min block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	—	—	—
0 division	—	—	—

Max block



■Function details

This block compares two input values and outputs the larger value.

The data type of input/ output values is set in the property. (☞ Page 159 Properties)

■I/O signals

The following table lists the I/O signals of the Max block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input1	Input	1	LREAL WORD INT DWORD DINT	Minimum value to maximum value of each data type	—
input2	Input	2	LREAL WORD INT DWORD DINT	Minimum value to maximum value of each data type	—
output	Output	1	LREAL WORD INT DWORD DINT	Minimum value to maximum value of each data type	—

■Properties

The following table lists the properties of the Max block.

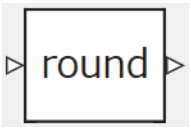
Property name	Attribute	Range/option	Default	Possible input value	Remarks
Type	Drop-down list	• LREAL • WORD • INT • DWORD • DINT	LREAL	—	—

■Error

The following table lists errors that occur in the Max block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	—	—	—
0 division	—	—	—

Round block



■Function details

This block outputs an integer value obtained by rounding off the input value to the first decimal place in LREAL data type.

■I/O signals

The following table lists the I/O signals of the Round block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input	Input	1	LREAL	Minimum value to maximum value of LREAL data type	—
output	Output	1	LREAL	Minimum value to maximum value of LREAL data type	—

■Properties

The Round block bas no properties.

■Error

The following table lists errors that occur in the Round block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	—	—	—
0 division	—	—	—

Floor block

■Function details

This block outputs an integer value obtained by rounding down the input value to the decimal point in LREAL data type.

■I/O signals

The following table lists the I/O signals of the Floor block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input	Input	1	LREAL	Minimum value to maximum value of LREAL data type	—
output	Output	1	LREAL	Minimum value to maximum value of LREAL data type	—

■Properties

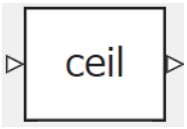
The Floor block has no properties.

■Error

The following table lists errors that occur in the Floor block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	—	—	—
0 division	—	—	—

Ceil block



■Function details

This block outputs an integer value obtained by rounding up the input value to the first decimal point in LREAL data type.

■I/O signals

The following table lists the I/O signals of the Ceil block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input	Input	1	LREAL	Minimum value to maximum value of LREAL data type	—
output	Output	1	LREAL	Minimum value to maximum value of LREAL data type	—

■Properties

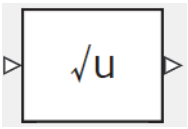
The Ceil block has no properties.

■Error

The following table lists errors that occur in the Ceil block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	—	—	—
0 division	—	—	—

Sqrt block



■Function details

This block calculates the square root of the input value by using the formula below and outputs the result.

$$output = \sqrt{input}$$

- output: Output value
- input: Input value

■I/O signals

The following table lists the I/O signals of the Sqrt block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input	Input	1	LREAL	0 to maximum value of LREAL data type	—
output	Output	1	LREAL	0 to square root of maximum value of LREAL data type	—

■Properties

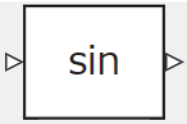
The Sqrt block has no properties.

■Error

The following table lists errors that occur in the Sqrt block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	○	When the following is satisfied input < 0	—
Output overflow	—	—	—
0 division	—	—	—

Sin block



■Function details

This block calculates the sine of the input value by using the formula below and outputs the result. The unit of the input value is radian.

$$output = \sin(input)$$

- output: Output value
- input: Input value

■I/O signals

The following table lists the I/O signals of the Sin block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input	Input	1	LREAL	Minimum value to maximum value of LREAL data type	—
output	Output	1	LREAL	-1 ≤ output ≤ 1	—

■Properties

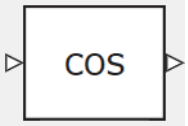
The Sin block has no properties.

■Error

The following table lists errors that occur in the Sin block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	—	—	—
0 division	—	—	—

Cos block



■Function details

This block calculates the cosine of the input value by using the formula below and outputs the result. The unit of the input value is radian.

$$output = \cos(input)$$

- output: Output value
- input: Input value

■I/O signals

The following table lists the I/O signals of the Cos block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input	Input	1	LREAL	Minimum value to maximum value of LREAL data type	—
output	Output	1	LREAL	-1 ≤ output ≤ 1	—

■Properties

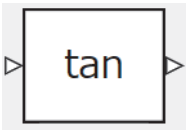
The Cos block has no properties.

■Error

The following table lists errors that occur in the Cos block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	—	—	—
0 division	—	—	—

Tan block



■Function details

This block calculates the tangent of the input value by using the formula below and outputs the result. The unit of the input value is radian.

$$output = \tan(input)$$

- output: Output value
- input: Input value

■I/O signals

The following table lists the I/O signals of the Tan block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input	Input	1	LREAL	Minimum value to maximum value of LREAL data type	—
output	Output	1	LREAL	Minimum value to maximum value of LREAL data type	—

■Properties

The Tan block has no properties.

■Error

The following table lists errors that occur in the Tan block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	—	—	—
0 division	—	—	—

Asin block

■Function details

This block calculates the arcsine of the input value by using the formula below and outputs the result. The unit of the input value is radian.

$$output = \sin^{-1}(input)$$

- output: Output value
- input: Input value

■I/O signals

The following table lists the I/O signals of the Asin block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input	Input	1	LREAL	-1 ≤ input ≤ 1	—
output	Output	1	LREAL	-π/2 ≤ output ≤ π/2	—

■Properties

The Asin block has no properties.

■Error

The following table lists errors that occur in the Asin block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	○	When either of the following is satisfied • input < -1 • 1 < input	—
Output overflow	—	—	—
0 division	—	—	—

Acos block

■Function details

This block calculates the arccosine of the input value by using the formula below and outputs the result. The unit of the input value is radian.

$$output = \cos^{-1}(input)$$

- output: Output value
- input: Input value

■I/O signals

The following table lists the I/O signals of the Acos block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input	Input	1	LREAL	-1 ≤ input ≤ 1	—
output	Output	1	LREAL	0 ≤ output ≤ π	—

■Properties

The Acos block has no properties.

■Error

The following table lists errors that occur in the Acos block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	○	When either of the following is satisfied • input < -1 • 1 < input	—
Output overflow	—	—	—
0 division	—	—	—

Atan block

■Function details

This block calculates the arctangent of the input value by using the formula below and outputs the result. The unit of the input value is radian.

$$output = \tan^{-1}(input)$$

- output: Output value
- input: Input value

■I/O signals

The following table lists the I/O signals of the Atan block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input	Input	1	LREAL	Minimum value to maximum value of LREAL data type	—
output	Output	1	LREAL	-π/2 < output < π/2	—

■Properties

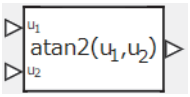
The Atan block has no properties.

■Error

The following table lists errors that occur in the Atan block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	—	—	—
0 division	—	—	—

Atan2 block



■Function details

This block calculates the arctangent of the division of two input values by using the formula below and outputs the result. The unit of the input value is radian.

$$output = \tan^{-1}(u1/u2)$$

- output: Output value
- u1: First input value
- u2: Second input value

■I/O signals

The following table lists the I/O signals of the Atan2 block.

Signal name	Classification	Input/output order	Type	Range	Remarks
u1	Input	1	LREAL	Minimum value to maximum value of LREAL data type	—
u2	Input	2	LREAL	Minimum value to maximum value of LREAL data type	—
output	Output	1	LREAL	$-\pi < output \leq \pi$	When input2 is 0, output becomes 0 regardless of the value of input1.

■Properties

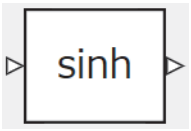
The Atan2 block has no properties.

■Error

The following table lists errors that occur in the Atan2 block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	○	When both of the following are satisfied • input1 = 0 • input2 = 0	—
Output overflow	—	—	—
0 division	—	—	—

Sinh block



■Function details

This block calculates the hyperbolic sine by using the formula below and outputs the result.

$$output = \sinh(input)$$

- output: Output value
- input: Input value

■I/O signals

The following table lists the I/O signals of the Sinh block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input	Input	1	LREAL	Minimum value to maximum value of LREAL data type	—
output	Output	1	LREAL	Minimum value to maximum value of LREAL data type	—

■Properties

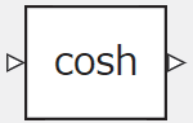
The Sinh block has no properties.

■Error

The following table lists errors that occur in the Sinh block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	○	When the data type is LREAL and output is outside the following range $\|output\| < 2^{1024}$	—
0 division	—	—	—

Cosh block



■Function details

This block calculates the hyperbolic cosine by using the formula below and outputs the result.

$$output = \cosh(input)$$

- output: Output value
- input: Input value

■I/O signals

The following table lists the I/O signals of the Cosh block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input	Input	1	LREAL	Minimum value to maximum value of LREAL data type	—
output	Output	1	LREAL	1 to maximum value of LREAL data type	—

■Properties

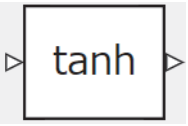
The Cosh block has no properties.

■Error

The following table lists errors that occur in the Cosh block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	○	When output exceeds the following range output < 2^{1024}	—
0 division	—	—	—

Tanh block



■Function details

This block calculates the hyperbolic tangent by using the formula below and outputs the result.

$$output = \tanh(input)$$

- output: Output value
- input: Input value

■I/O signals

The following table lists the I/O signals of the Tanh block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input	Input	1	LREAL	Minimum value to maximum value of LREAL data type	—
output	Output	1	LREAL	-1 ≤ output ≤ 1	—

■Properties

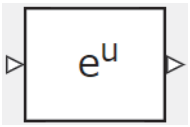
The Tanh block has no properties.

■Error

The following table lists errors that occur in the Tanh block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	—	—	—
0 division	—	—	—

Exp block



■Function details

This block calculates the natural exponential function by using the formula below and outputs the result.

$$output = e^{input}$$

- output: Output value
- input: Input value

■I/O signals

The following table lists the I/O signals of the Exp block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input	Input	1	LREAL	Minimum value to maximum value of LREAL data type	—
output	Output	1	LREAL	Minimum value to maximum value of LREAL data type	—

■Properties

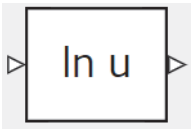
The Exp block has no properties.

■Error

The following table lists errors that occur in the Exp block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	○	When output exceeds the following range output < 2^{1024}	—
0 division	—	—	—

Log block



■Function details

This block calculates the natural logarithm by using the formula below and outputs the result.

$$output = \log(input)$$

- output: Output value
- input: Input value

■I/O signals

The following table lists the I/O signals of the Log block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input	Input	1	LREAL	0 < input ≤ Maximum value of LREAL data type	—
output	Output	1	LREAL	Minimum value to maximum value of LREAL data type	—

■Properties

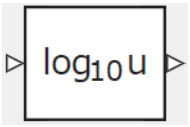
The Log block has no properties.

■Error

The following table lists errors that occur in the Log block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	○	When the following is satisfied input ≤ 0	—
Output overflow	—	—	—
0 division	—	—	—

Log10 block



■Function details

This block calculates the common logarithm by using the formula below and outputs the result.

$$output = \log_{10}(input)$$

- output: Output value
- input: Input value

■I/O signals

The following table lists the I/O signals of the Log10 block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input	Input	1	LREAL	0 < input ≤ Maximum value of LREAL data type	—
output	Output	1	LREAL	Minimum value to maximum value of LREAL data type	—

■Properties

The Log10 block has no properties.

■Error

The following table lists errors that occur in the Log10 block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	○	When the following is satisfied input ≤ 0	—
Output overflow	—	—	—
0 division	—	—	—

Logical operation

WAND block

■Function details

This block calculates the logical AND of two input values in units of bits and outputs the result.

■I/O signals

The following table lists the I/O signals of the WAND block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input1	Input	1	WORD	Minimum value to maximum value of WORD data type	—
input2	Input	2	WORD	Minimum value to maximum value of WORD data type	—
output	Output	1	WORD	Minimum value to maximum value of WORD data type	—

■Properties

The WAND block has no properties.

■Error

The following table lists errors that occur in the WAND block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	—	—	—
0 division	—	—	—

WOR block

■Function details

This block calculates the logical OR of two input values in units of bits and outputs the result.

■I/O signals

The following table lists the I/O signals of the WOR block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input1	Input	1	WORD	Minimum value to maximum value of WORD data type	—
input2	Input	2	WORD	Minimum value to maximum value of WORD data type	—
output	Output	1	WORD	Minimum value to maximum value of WORD data type	—

■Properties

The WOR block has no properties.

■Error

The following table lists errors that occur in the WOR block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	—	—	—
0 division	—	—	—

WXOR block

■Function details

This block calculates the exclusive OR of two input values in units of bits and outputs the result.

■I/O signals

The following table lists the I/O signals of the WXOR block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input1	Input	1	WORD	Minimum value to maximum value of WORD data type	—
input2	Input	2	WORD	Minimum value to maximum value of WORD data type	—
output	Output	1	WORD	Minimum value to maximum value of WORD data type	—

■Properties

The WXOR block has no properties.

■Error

The following table lists errors that occur in the WXOR block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	—	—	—
0 division	—	—	—

WXNR block

■Function details

This block calculates the exclusive NOR of two input values in units of bits and outputs the result.

■I/O signals

The following table lists the I/O signals of the WXNR block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input1	Input	1	WORD	Minimum value to maximum value of WORD data type	—
input2	Input	2	WORD	Minimum value to maximum value of WORD data type	—
output	Output	1	WORD	Minimum value to maximum value of WORD data type	—

■Properties

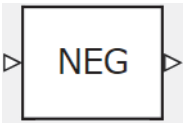
The WXNR block has no properties.

■Error

The following table lists errors that occur in the WXNR block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	—	—	—
0 division	—	—	—

NEG block



■Function details

This block calculates the logical NOT of the input value in units of bits and outputs the result.

■I/O signals

The following table lists the I/O signals of the NEG block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input	Input	1	WORD	Minimum value to maximum value of WORD data type	—
output	Output	1	WORD	Minimum value to maximum value of WORD data type	—

■Properties

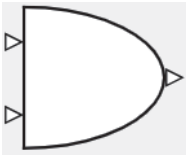
The NEG block has no properties.

■Error

The following table lists errors that occur in the NEG block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	—	—	—
0 division	—	—	—

AND block



■Function details

This block calculates the logical AND of two input values and outputs the result.

■I/O signals

The following table lists the I/O signals of the AND block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input1	Input	1	BOOL	FALSE, TRUE	—
input2	Input	2	BOOL	FALSE, TRUE	—
output	Output	1	BOOL	FALSE, TRUE	—

■Properties

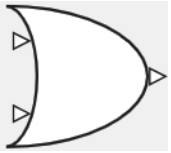
The AND block has no properties.

■Error

The following table lists errors that occur in the AND block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	—	—	—
0 division	—	—	—

OR block



■Function details

This block calculates the logical OR of two input values and outputs the result.

■I/O signals

The following table lists the I/O signals of the OR block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input1	Input	1	BOOL	FALSE, TRUE	—
input2	Input	2	BOOL	FALSE, TRUE	—
output	Output	1	BOOL	FALSE, TRUE	—

■Properties

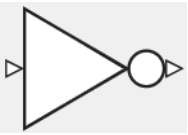
The OR block has no properties.

■Error

The following table lists errors that occur in the OR block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	—	—	—
0 division	—	—	—

NOT block



■Function details

This block calculates the logical NOT of the input value and outputs the result.

■I/O signals

The following table lists the I/O signals of the NOT block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input	Input	1	BOOL	FALSE, TRUE	—
output	Output	1	BOOL	FALSE, TRUE	—

■Properties

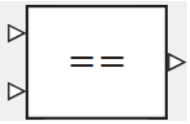
The NOT block has no properties.

■Error

The following table lists errors that occur in the NOT block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	—	—	—
0 division	—	—	—

EQ block



■Function details

This block compares two input values and outputs TRUE if they are equal, and FALSE if they are not equal.

The data type of input/ output values is set in the property. (☞ Page 181 Properties)

■I/O signals

The following table lists the I/O signals of the EQ block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input1	Input	1	LREAL WORD INT BOOL	Minimum value to maximum value of each data type	—
input2	Input	2	LREAL WORD INT BOOL	Minimum value to maximum value of each data type	—
output	Output	1	BOOL	FALSE, TRUE	—

■Properties

The following table lists the properties of the EQ block.

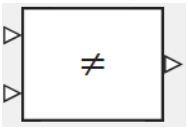
Property name	Attribute	Range/option	Default	Possible input value	Remarks
Type	Drop-down list	LREAL WORD INT BOOL	LREAL	—	—

■Error

The following table lists errors that occur in the EQ block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	—	—	—
0 division	—	—	—

NE block



■Function details

This block compares two input values and outputs FALSE if they are equal, and TRUE if they are not equal.

The data type of input/ output values is set in the property. (☞ Page 182 Properties)

■I/O signals

The following table lists the I/O signals of the NE block.

Signal name	Classification	Input/output order	Type	Range	Remarks
u1	Input	1	LREAL WORD INT BOOL	Minimum value to maximum value of each data type	—
u2	Input	2	LREAL WORD INT BOOL	Minimum value to maximum value of each data type	—
output	Output	1	BOOL	FALSE, TRUE	—

■Properties

The following table lists the properties of the NE block.

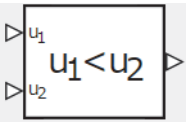
Property name	Attribute	Range/option	Default	Possible input value	Remarks
Type	Drop-down list	LREAL WORD INT BOOL	LREAL	—	—

■Error

The following table lists errors that occur in the NE block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	—	—	—
0 division	—	—	—

LT block



■Function details

This block compares two input values and outputs TRUE if the first input value is less than the second input value, and otherwise outputs FALSE.

The data type of input/ output values is set in the property. (☞ Page 183 Properties)

■I/O signals

The following table lists the I/O signals of the LT block.

Signal name	Classification	Input/output order	Type	Range	Remarks
u1	Input	1	LREAL WORD INT	Minimum value to maximum value of each data type	—
u2	Input	2	LREAL WORD INT	Minimum value to maximum value of each data type	—
output	Output	1	BOOL	FALSE, TRUE	—

■Properties

The following table lists the properties of the LT block.

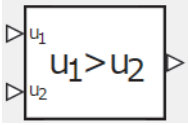
Property name	Attribute	Range/option	Default	Possible input value	Remarks
Type	Drop-down list	LREAL WORD INT	LREAL	—	—

■Error

The following table lists errors that occur in the LT block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	—	—	—
0 division	—	—	—

GT block



■Function details

This block compares two input values and outputs TRUE if the first input value is greater than the second input value, and otherwise outputs FALSE.

The data type of input/ output values is set in the property. (☞ Page 184 Properties)

■I/O signals

The following table lists the I/O signals of the GT block.

Signal name	Classification	Input/output order	Type	Range	Remarks
u1	Input	1	LREAL WORD INT	Minimum value to maximum value of each data type	—
u2	Input	2	LREAL WORD INT	Minimum value to maximum value of each data type	—
output	Output	1	BOOL	FALSE, TRUE	—

■Properties

The following table lists the properties of the GT block.

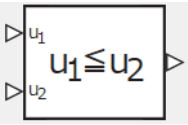
Property name	Attribute	Range/option	Default	Possible input value	Remarks
Type	Drop-down list	LREAL WORD INT	LREAL	—	—

■Error

The following table lists errors that occur in the GT block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	—	—	—
0 division	—	—	—

LE block



■Function details

This block compares two input values and outputs TRUE if the first input value is less than or equal to the second input value, and otherwise outputs FALSE.

The data type of input/ output values is set in the property. (☞ Page 185 Properties)

■I/O signals

The following table lists the I/O signals of the LE block.

Signal name	Classification	Input/output order	Type	Range	Remarks
u1	Input	1	LREAL WORD INT	Minimum value to maximum value of each data type	—
u2	Input	2	LREAL WORD INT	Minimum value to maximum value of each data type	—
output	Output	1	BOOL	FALSE, TRUE	—

■Properties

The following table lists the properties of the LE block.

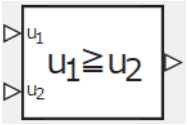
Property name	Attribute	Range/option	Default	Possible input value	Remarks
Type	Drop-down list	LREAL WORD INT	LREAL	—	—

■Error

The following table lists errors that occur in the LE block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	—	—	—
0 division	—	—	—

GE block



■Function details

This block compares two input values and outputs TRUE if the first input value is greater than or equal to the second input value, and otherwise outputs FALSE.

The data type of input/ output values is set in the property. (☞ Page 186 Properties)

■I/O signals

The following table lists the I/O signals of the GE block.

Signal name	Classification	Input/output order	Type	Range	Remarks
u1	Input	1	LREAL WORD INT	Minimum value to maximum value of each data type	—
u2	Input	2	LREAL WORD INT	Minimum value to maximum value of each data type	—
output	Output	1	BOOL	FALSE, TRUE	—

■Properties

The following table lists the properties of the GE block.

Property name	Attribute	Range/option	Default	Possible input value	Remarks
Type	Drop-down list	LREAL WORD INT	LREAL	—	—

■Error

The following table lists errors that occur in the GE block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	—	—	—
0 division	—	—	—

FMU

FMU input block

■Function details

This block functions as an input terminal in a FMU exported using the FMU export function.

By placing and connecting the FMU block, the FMU input block defines the data input to MELSOFT Mirror by another simulation software when the MELSOFT Mirror FMU is embedded as a component within that software.

When FMU Link is on, the block outputs the input value to the FMU of the FMI master software. When FMU Link is off, the block outputs the value configured in the "Value when FMU Link OFF" setting.

The name assigned to the FMU input block is used as the name of the corresponding FMU input terminal in the generated FMU. The order of input terminals in the generated FMU is determined by sorting the input terminal names of all FMU input blocks in ASCII order.

■I/O signals

The following table lists the I/O signals of the FMU input block.

Signal name	Classification	Input/output order	Type	Range	Remarks
output	Output	1	LREAL WORD DWORD INT DINT BOOL	Minimum value to maximum value of each data type	—

■Properties

The following table lists the properties of the FMU input block.

Property name	Attribute	Range/option	Default	Possible input value	Remarks
Type	Drop-down list	LREAL WORD DWORD INT DINT BOOL	LREAL	—	—
Input as Bit Array	Drop-down list	Off On	Off	—	• For WORD and DWORD data types, this property is enabled. For other data types, the property is disabled and set to Off. • When the property is set to On and the data type is WORD, MELSOFT Mirror receives 16 FMI Boolean data type arrays from the FMI master software as a single WORD value. The FMU input block outputs the received WORD value. • When the data type is DWORD, MELSOFT Mirror receives 32 FMI Boolean data type arrays. • In the FMI master software, signal names use the format "BlockName#BitNumber(Hexadecimal)". • The bit number is one digit (0 to F) for WORD and two digits (00 to 1F) for DWORD. • When the property is set to Off, WORD and DWORD values are input as FMI Integer data type.
Unit	Drop-down list	No Unit m m/s m/s2 deg deg/s deg/s2 rad rad/s rad/s2	No Unit	—	• For LREAL data type, this property is enabled. For other data types, the property is disabled and set to No Unit. • The unit set here is displayed in the FMU Import block properties of the FMI master software.
Value when FMU Link OFF	• Text box • Drop-down list	Minimum value to maximum value of each data type	FALSE for BOOL data type, and otherwise 0	■LREAL data type • One-byte numbers (0 to 9) • One-byte alphabetic characters (E, e) • One-byte symbols (plus (+), hyphen (-), period (.)) ■BOOL data type • FALSE, TRUE ■Other data types • One-byte numbers (0 to 9) • One-byte alphabetic characters (A to F, a to f, o, x) • One-byte symbols (plus (+), hyphen (-))	• Up to 13 characters can be entered. • For BOOL data type, a drop-down list is used. For other data types, a text box is used. • When FMU Link is off, this property value is output as a substitute for FMU input. • For the notation that can be used, refer to the following. ☞ Page 127 Notation that can be used for properties

■FMU data types

The type of a FMU input signal, which is input to the FMU input block, is determined by the block properties "Type" and "Input as Bit Array".

FMU input signals are converted as follows: Real to LREAL, Integer to DINT, and Boolean to BOOL. For DWORD data type, the block outputs a DWORD value whose Integer value matches its binary representation.

Point

Set the input/output data type and the value to use when FMU Link is off in the properties.

The following table lists the settings and their corresponding FMU input signal types.

Type	Input as Bit Array	FMU input signal type	Remarks
LREAL	Disabled (OFF)	Real	—
WORD	OFF	Integer	Converts the Integer value, which is input from the FMI master software, to its bit representation and outputs the lower 16 bits as a WORD value.
WORD	ON	Boolean × 16	• The FMU input block outputs the 16 Boolean signals, which are input from the FMI master software, as a single WORD value. • FMI variable names are suffixed with #0 through #F. Bit 0 of the WORD value corresponds to #0, and the remaining bits follow the same pattern.
DWORD	OFF	Integer	Converts the Integer value, which is input from the FMI master software, to its bit representation and outputs it as a DWORD value.
DWORD	ON	Boolean × 32	• The FMU input block outputs the 32 Boolean signals, which are input from the FMI master software, as a single DWORD value. • FMI variable names are suffixed with #00 through #1F. Bit 0 of the DWORD value corresponds to #0, and the remaining bits follow the same pattern.
INT	Disabled (OFF)	Integer	Converts the Integer value, which is input from the FMI master software, to its bit representation, and outputs the lower 16 bits as an INT value.
DINT	Disabled (OFF)	Integer	—
BOOL	Disabled (OFF)	Boolean	—

The following table lists example correspondences between Integer values and DWORD, WORD, DINT, and INT values.

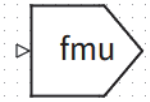
Bit representation (hexadecimal)	Integer value	DWORD value	WORD value	DINT value	INT value
0x00000000	0	0	0	0	0
0x00000001	1	1	1	1	1
0x00007FFF	32767	32767	32767	32767	32767
0x00008000	32768	32768	32768	32768	-32768
0x0000FFFF	65535	65535	65535	65535	-1
0x00010000	65536	65536	0	65536	0
0x7FFFFFFF	2147483647	2147483647	65535	2147483647	-1
0x80000000	-2147483648	2147483648	0	-2147483648	0
0x80000001	-2147483647	2147483649	1	-2147483647	1
0xFFFFFFFF	-1	4294967295	65535	-1	-1

■Error

The following table lists errors that occur in the FMU input block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	○	When output exceeds the following range $\|output\| < 2^{1024}$	—
0 division	—	—	—

FMU output block



■Function details

This block functions as an output terminal in a FMU exported using the FMU export function.

By placing and connecting the FMU block, the FMU output block defines the data output from MELSOFT Mirror to another simulation software when the MELSOFT Mirror FMU is embedded as a component within that software.

When FMU Link is on, the block outputs the input value as the FMU output value of the FMI master software. When FMU Link is off, the input value is ignored.

The name assigned to the FMU output block is used as the name of the corresponding FMU output terminal in the generated FMU. The order of output terminals in the generated FMU is determined by sorting the output terminal names of all FMU output blocks in ASCII order.

■I/O signals

The following table lists the I/O signals of the FMU output block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input	Input	1	LREAL WORD INT DWORD DINT BOOL	Minimum value to maximum value of each data type	—

■Properties

The following table lists the properties of the FMU output block.

Property name	Attribute	Range/option	Default	Possible input value	Remarks
Type	Drop-down list	LREAL WORD DWORD INT DINT BOOL	LREAL	—	—
Output as Bit Array	Drop-down list	Off On	Off	—	• For WORD and DWORD data types, this property is enabled. For other data types, the property is disabled and set to Off. • When the property is set to On and the data type is WORD, the FMU output block outputs a single WORD value, which is input to the FMU output block, to the FMI master software as 16 FMI Boolean data type arrays. (The FMI master software receives the single WORD value, which is input to the FMU output block of MELSOFT Mirror, as 16 FMI Boolean data type arrays.) • When the data type is DWORD, the FMI master software receives 32 FMI Boolean data type arrays. • In the FMI master software, signal names use the format "BlockName#BitNumber(Hexadecimal)". • The bit number is one digit (0 to F) for WORD and two digits (00 to 1F) for DWORD. • When the property is set to Off, WORD and DWORD values are output as FMI Integer data type.
Unit	Drop-down list	No Unit m m/s m/s2 deg deg/s deg/s2 rad rad/s rad/s2	No Unit	—	• For LREAL data type, this property is enabled. For other data types, the property is disabled and set to No Unit. • The unit set here is displayed in the FMU Import block properties of the FMI master software.

■FMU data types

The type of a FMU output signal, which is output from the FMU output block, is determined by the block properties "Type" and "Output as Bit Array".

FMU output signals are converted as follows: Real to FMU REAL, WORD/INT to FMU Integer (equivalent to DINT), BOOL to FMU Boolean. DWORD data type is treated as the Integer data type with the same bit representation.

The following table lists the settings and their corresponding FMU output signal types.

Type	Output as Bit Array	FMU output signal type	Remarks
LREAL	Disabled (OFF)	Real	—
WORD	OFF	Integer	—
WORD	ON	Boolean × 16	• Converts the WORD value (16 bits) to 16 Boolean signals and outputs the converted value. • FMI variable names are suffixed with #0 through #F. Bit 0 of the WORD value corresponds to #0, and the remaining bits follow the same pattern.
DWORD	OFF	Integer	The bit representation of DWORD is interpreted as the bit representation of Integer. Consequently, the values 2147483648 to 4294967295 are converted to the values -2147483648 to -1.
DWORD	ON	Boolean × 32	• Converts the DWORD value (32 bits) to 32 Boolean signals and outputs the converted value. • FMI variable names are suffixed with #00 through #1F. Bit 0 of the DWORD value corresponds to #0, and the remaining bits follow the same pattern.
INT	Disabled (OFF)	Integer	—
DINT	Disabled (OFF)	Integer	—
BOOL	Disabled (OFF)	Boolean	—

The following table lists example correspondences between DWORD values and Integer values.

Bit representation (hexadecimal)	DWORD value	Integer value
0x00000000	0	0
0x00000001	1	1
0x7FFFFFFF	2147483647	2147483647
0x80000000	2147483648	-2147483648
0x80000001	2147483649	-2147483647
0xFFFFFFFF	4294967295	-1

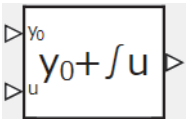
■Error

The following table lists errors that occur in the FMU output block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	—	—	—
0 division	—	—	—

Continuous time

Integral block



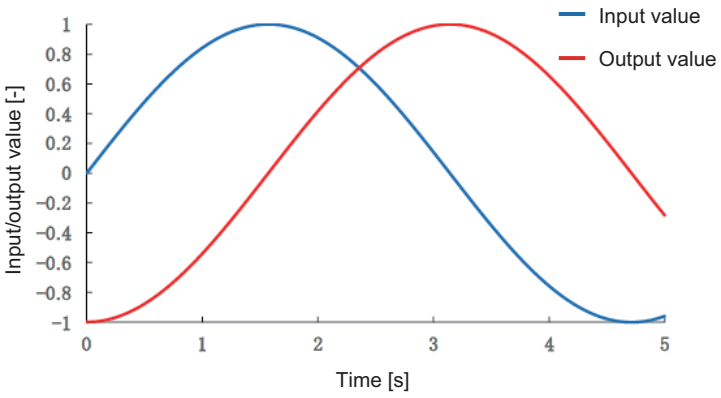
■Function details

This block calculates the integral value by using the formula below and outputs the result.
The integral calculation algorithm can be selected from "Fourth-Order Runge Kutta" and "Euler" and set in the I/O Settings window. (☞ Page 46 I/O Setting)

$$output = y0 + \int_{t_0}^{t} u\, dt$$

- output: Output value
- y0: Initial value
- t: Current step time
- t_0: Initial step time
- u: Input value

Ex.
The following is a graph of the output values when sine wave signals are input for 5 seconds at 0.1 second intervals.



■I/O signals

The following table lists the I/O signals of the integral block.

Signal name	Classification	Input/output order	Type	Range	Remarks
y0	Input	1	LREAL	Minimum value to maximum value of LREAL data type	—
u	Input	2	LREAL	Minimum value to maximum value of LREAL data type	—
output	Output	1	LREAL	Minimum value to maximum value of LREAL data type	—

■Properties

The integral block has no properties.

■Error

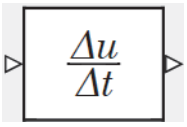
The following table lists errors that occur in the integral block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	○	When output is outside the following range $\|output\| < 2^{1024}$	—
0 division	—	—	—

Point

Examples of systems that use an integral block include an emergency braking control system for automobiles and a volume control system for water storage tanks.

Differential block



■Function details

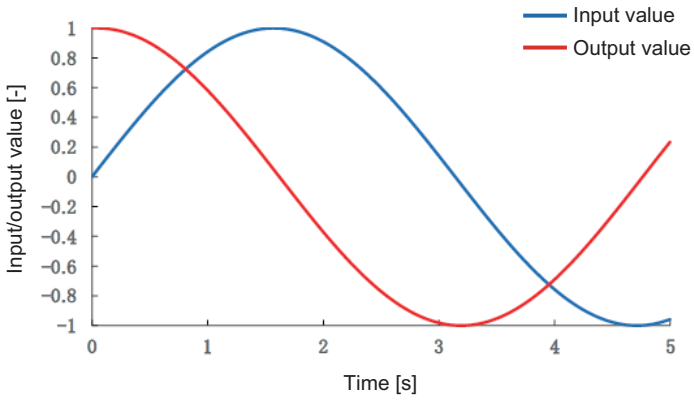
This block calculates the differential value by using the formula below and outputs the result.

$$output = \frac{\Delta input}{\Delta t}$$

- output: Output value
- Δinput: Difference with the calculated value of the previous step (calculated value of the initial step: 0)
- Δt: Integral step size in I/O simulator (222μ) [seconds]

Ex.

The following is a graph of the output values when sine wave signals are input for 5 seconds at 0.1 second intervals.



■I/O signals

The following table lists the I/O signals of the differential block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input	Input	1	LREAL	Minimum value to maximum value of LREAL data type	—
output	Output	1	LREAL	Minimum value to maximum value of LREAL data type	—

■Properties

The differential block has no properties.

■Error

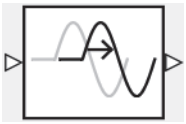
The following table lists errors that occur in the differential block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	○	When output is outside the following range |output| < 2^{1024}	—
0 division	—	—	—

Point

An example of a system that uses a differential block is a temperature control system using PID control.

Delay block



■Function details

This block outputs the initial output until the delay time set in the property. After the delay time is exceeded, the block outputs values input before the delay time. (☞ Page 198 Properties)

If the delay time is 0, the input values are output as they are.

When the "Delay Time Type" property is set to "Integration Step Size Multiplier", the block outputs the product of the specified value and the integration step size as the delay time.

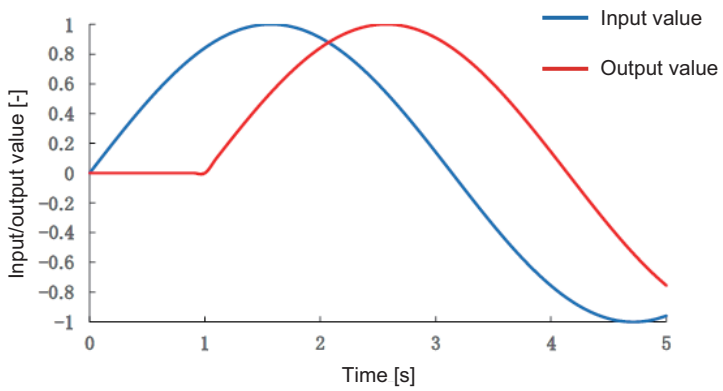
When the "Delay Time Type" property is set to "Milliseconds", the block outputs the largest multiple of the integration step size that is less than or equal to the specified value.

Ex.

The following is a graph of the output values when sine wave signals are input for 5 seconds at 0.1 second intervals.

Initial output: 0.0

Delay time: 1.0



■I/O signals

The following table lists the I/O signals of the delay block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input	Input	1	LREAL WORD INT DWORD DINT BOOL	Minimum value to maximum value of each data type	—
output	Output	1	LREAL WORD INT DWORD DINT BOOL	Minimum value to maximum value of each data type	—

■Properties

The following table lists the properties of the delay block.

Property name	Attribute	Range/option	Default	Possible input value	Remarks
Type	Drop-down list	LREAL WORD INT DWORD DINT BOOL	LREAL	—	—
Delay Time Type	Drop-down list	Integration Step Size Multiplier Milliseconds	Integration Step Size Multiplier	—	—
Delay Time [Integration Step Size Multiplier]	Text box	0 to 270001 (1 minute / Integration Step Size)	0	One-byte numbers	• Input field of Integration Step Size Multiplier • 13 characters maximum • This property is displayed only when "Delay Time Type" is set to "Integration Step Size Multiplier". • When "Delay Time Type" is set to "Milliseconds", the default value is set.
Delay Time [ms]	Text box	0 to 60000	0	One-byte numbers	• Input field of Milliseconds • 13 characters maximum • This property is displayed only when "Delay Time Type" is set to "Milliseconds". • When "Delay Time Type" is set to "Integration Step Size Multiplier", the default value is set.
Initial output	Text box	Minimum value to maximum value of each data type	• BOOL data type: False • Others: 0	• LREAL data type: One-byte numbers, one-byte alphabetic characters (E, e), and symbols (+, -) • BOOL data type: True, False • Others: One-byte numbers	• 13 characters maximum • For the notation that can be used, refer to the following. ☞Page 127 Notation that can be used for properties

■Error

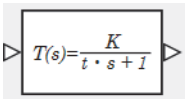
The following table lists errors that occur in the delay block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	—	—	—
0 division	—	—	—

Point

An example of a system that uses a delay block is a control system that includes variable time delays.

Primary delay block



■Function details

This block calculates the differential equation below and outputs the result.

$$u(t) = \frac{T}{K}y'(t) + \frac{1}{K}y(t)$$

- u(t): Input value
- T: Time constant
- K: Steady-state gain
- y(t): Output value
- y'(t): First derivative of output value

The transfer function of the differential equation above is expressed by the formula below.

$$G(s) = \frac{K}{Ts + 1}$$

- G(s): Transfer function (function that expresses the relationship between input and output of the system)
- s: Complex variable

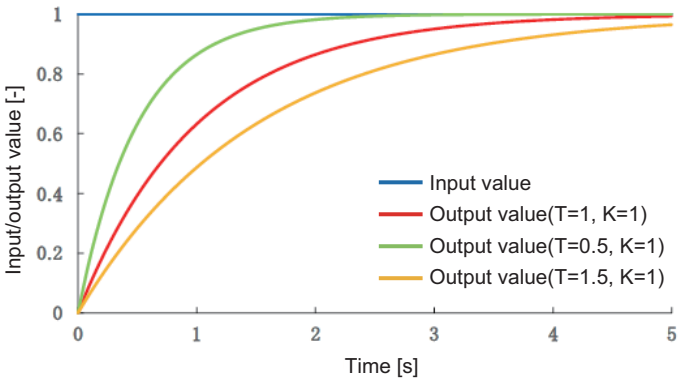
T (time constant) and K (steady-state gain) are set in the properties. (☞ Page 200 Properties)

Ex.

The following is a graph of the output values when step inputs (inputs that are always 1 after time 0) are input.

Steady-state gain: 1.0

Time constant: 0.5, 1.0, 1.5



■I/O signals

The following table lists the I/O signals of the primary delay block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input	Input	1	LREAL	Minimum value to maximum value of LREAL data type	—
output	Output	1	LREAL	Minimum value to maximum value of LREAL data type	—

■Properties

The following table lists the properties of the primary delay block.

Property name	Attribute	Range/option	Default	Possible input value	Remarks
Time constant [-]	Text box	Minimum value to maximum value of LREAL data type	1	One-byte numbers, one-byte alphabetic characters (E, e), symbols (+, -, .)	• 13 characters maximum • Exponential or decimal point notation can be used.
Steady-state gain [-]	Text box	Minimum value to maximum value of LREAL data type	1	One-byte numbers, one-byte alphabetic characters (E, e), symbols (+, -, .)	• 13 characters maximum • Exponential or decimal point notation can be used.

■Error

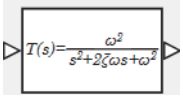
The following table lists errors that occur in the primary delay block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	○	When output is outside the following range $\|output\| < 2^{1024}$	—
0 division	—	—	—

Point

Examples of systems that use a primary delay block include a system that controls the rotation speed of motors and a system that controls RC circuits.

Secondary delay block



■Function details

This block calculates the differential equation below and outputs the result.

$$u(t) = \frac{1}{\omega^2} y''(t) + \frac{2\zeta}{\omega} y'(t) + y(t)$$

- u(t): Input value
- ω: Natural angular frequency
- ζ: Damping ratio
- y(t): Output value
- y'(t): First derivative of output value
- y"(t): Second derivative of output value

The transfer function of the differential equation above is expressed by the formula below.

$$G(s) = \frac{\omega^2}{s^2 + 2\zeta\omega s + \omega^2}$$

- G(s): Transfer function (function that expresses the relationship between input and output of the system)
- s: Complex variable

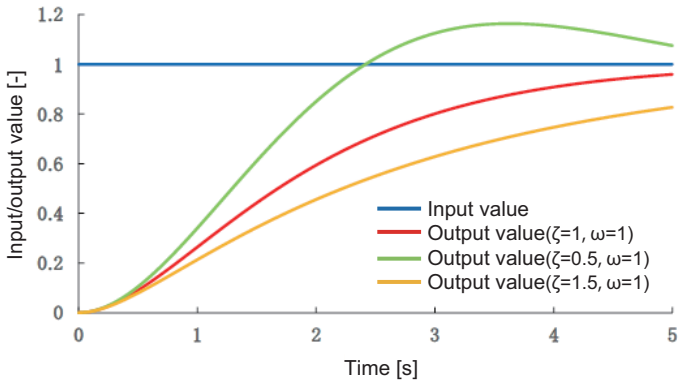
ω (natural angular frequency) and ζ (damping ratio) are set in the properties. (☞ Page 202 Properties)

Ex.

The following is a graph of the output values when step inputs (inputs that are always 1 after time 0) are input.

Natural angular frequency: 1.0

Damping ratio: 0.5, 1.0, 1.5



■I/O signals

The following table lists the I/O signals of the secondary delay block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input	Input	1	LREAL	Minimum value to maximum value of LREAL data type	—
output	Output	1	LREAL	Minimum value to maximum value of LREAL data type	—

■Properties

The following table lists the properties of the secondary delay block.

Property name	Attribute	Range/option	Default	Possible input value	Remarks
Natural angular frequency [-]	Text box	Minimum value to maximum value of LREAL data type	0	One-byte numbers, one-byte alphabetic characters (E, e), symbols (+, -, .)	• 13 characters maximum • Exponential or decimal point notation can be used.
Damping ratio [-]	Text box	Minimum value to maximum value of LREAL data type	0	One-byte numbers, one-byte alphabetic characters (E, e), symbols (+, -, .)	• 13 characters maximum • Exponential or decimal point notation can be used.

■Error

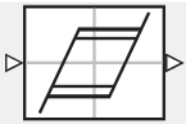
The following table lists errors that occur in the secondary delay block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	○	When output is outside the following range $\|output\| < 2^{1024}$	—
0 division	—	—	—

Point

Examples of systems that use a secondary delay block include a system that controls the rotation angle during pendulum rotation and a system that controls RLC circuits.

Backlash block



■Function details

This block outputs the value calculated using the formula below.

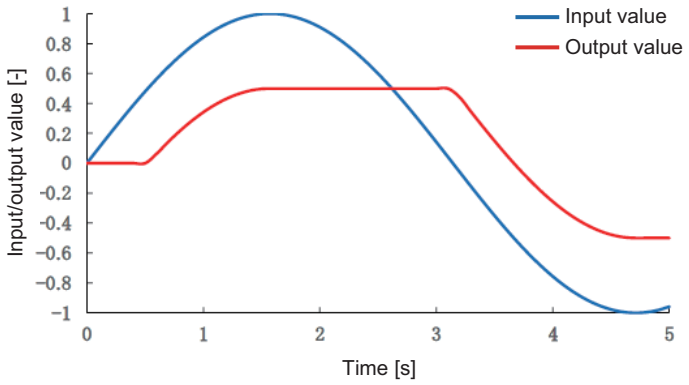
$$output = \begin{cases} preOut, & preOut - \frac{w}{2} \leq input \leq preOut + \frac{w}{2} \\ input + \frac{w}{2}, & input < preOut - \frac{w}{2} \\ input - \frac{w}{2}, & input > preOut + \frac{w}{2} \end{cases}$$

- output: Output value
- input: Input value
- w: Dead band width
- preOut: Output value of previous step (The initial step outputs the initial output.)

Initial output and w (dead band width) are set in the properties. (☞ Page 203 Properties)

Ex.

The following is a graph of the output values when sine wave signals are input for 5 seconds at 0.1 second intervals.
Initial output: 0.0, Dead band width: 1.0



■I/O signals

The following table lists the I/O signals of the backlash block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input	Input	1	LREAL	Minimum value to maximum value of LREAL data type	—
output	Output	1	LREAL	Minimum value to maximum value of LREAL data type	—

■Properties

The following table lists the properties of the backlash block.

Property name	Attribute	Range/option	Default	Possible input value	Remarks
Initial output	Text box	Minimum value to maximum value of LREAL data type	0	One-byte numbers, one-byte alphabetic characters (E, e), symbols (+, -, .)	• 13 characters maximum • Exponential or decimal point notation can be used.
Dead band width [-]	Text box	0 to maximum value of LREAL data type	0	One-byte numbers, one-byte alphabetic characters (E, e), symbols (+, -, .)	• 13 characters maximum • Exponential or decimal point notation can be used.

■Error

The following table lists errors that occur in the backlash block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	—	—	—
0 division	—	—	—

Point

An example of a system that uses a backlash block is a system that controls the engagement of two gears.

Dead band block

■Function details

This block outputs the value calculated using the formula below.

$$output = \begin{cases} 0, & LL \leq input \leq UL \\ input - LL, & input < LL \\ input - UL, & input > UL \end{cases}$$

- output: Output value
- input: Input value
- UL: Dead band upper limit
- LL: Dead band lower limit

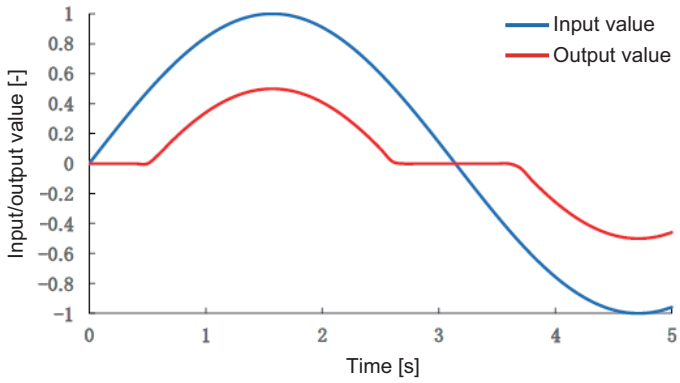
UL (dead band upper limit) and LL (dead band lower limit) are set in the properties. (☞ Page 205 Properties)

Ex.

The following is a graph of the output values when sine wave signals are input for 5 seconds at 0.1 second intervals.

Dead band upper limit: 0.5

Dead band lower limit: -0.5



■I/O signals

The following table lists the I/O signals of the dead band block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input	Input	1	LREAL	Minimum value to maximum value of LREAL data type	—
output	Output	1	LREAL	Minimum value to maximum value of LREAL data type	—

■Properties

The following table lists the properties of the dead band block.

Property name	Attribute	Range/option	Default	Possible input value	Remarks
Dead band upper limit UL [-]	Text box	0 to maximum value of LREAL data type	0	One-byte numbers, one-byte alphabetic characters (E, e), symbols (+, -, .)	• 13 characters maximum • Exponential or decimal point notation can be used.
Dead band lower limit LL [-]	Text box	Minimum value of LREAL data type to 0	0	One-byte numbers, one-byte alphabetic characters (E, e), symbols (+, -, .)	• 13 characters maximum • Exponential or decimal point notation can be used.

■Error

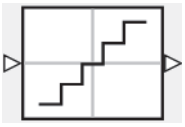
The following table lists errors that occur in the dead band block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	—	—	—
0 division	—	—	—

Point

An example of a system that uses a dead band block is a system that has a wide range of sensitivity thresholds for sensors.

Quantization block



■Function details

This block outputs the value calculated using the formula below.

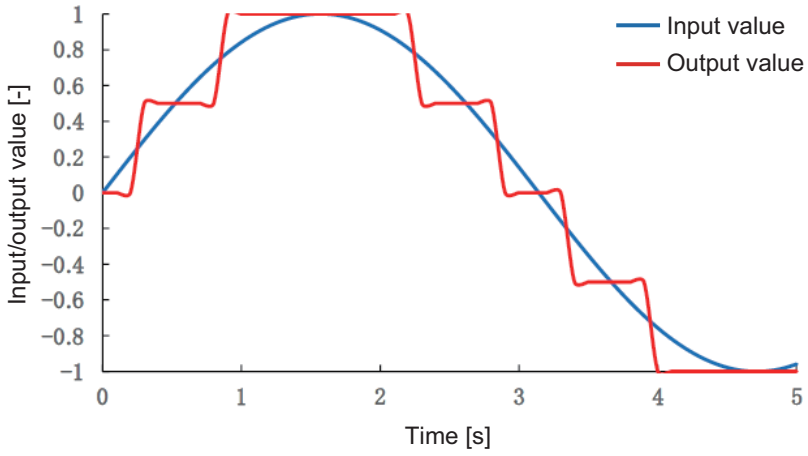
$$output = q \times round(\frac{input}{q})$$

- output: Output value
- input: Input value
- q: Quantization interval
- round: Round-off function (conversion to integer)

q (quantization interval) is set in the property. (☞ Page 207 Properties)

Ex.

The following is a graph of the output values when sine wave signals are input for 5 seconds at 0.1 second intervals.
Quantization interval: 0.5



■I/O signals

The following table lists the I/O signals of the quantization block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input	Input	1	LREAL	Minimum value to maximum value of LREAL data type	—
output	Output	1	LREAL	Minimum value to maximum value of LREAL data type	—

■Properties

The following table lists the properties of the quantization block.

Property name	Attribute	Range/option	Default	Possible input value	Remarks
Quantization interval [-]	Text box	2.22045E-016 to maximum value of LREAL data type	0.5	One-byte numbers, one-byte alphabetic characters (E, e), symbols (+, -, .)	• 13 characters maximum • Exponential or decimal point notation can be used.

■Error

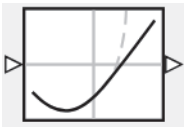
The following table lists errors that occur in the quantization block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	○	When output is outside the following range $\|output\| < 2^{1024}$	—
0 division	—	—	—

Point

An example of a system that uses a quantization block is a system that digitizes analog voltages.

Rate limiter block



■Function details

This block outputs the value calculated using the formula below.

$$output = \begin{cases} input, & F \leq \dfrac{input - preOut}{\Delta t} \leq R \\ \Delta t \cdot F + preOut, & \dfrac{input - preOut}{\Delta t} < F \\ \Delta t \cdot R + preOut, & \dfrac{input - preOut}{\Delta t} > R \end{cases}$$

- output: Output value
- input: Input value
- Δt: Integral step size in I/O simulator (222μ) [seconds]
- preOut: Output value of previous step (The initial step outputs the initial output.)
- F: Falling edge limit value
- R: Rising edge limit value

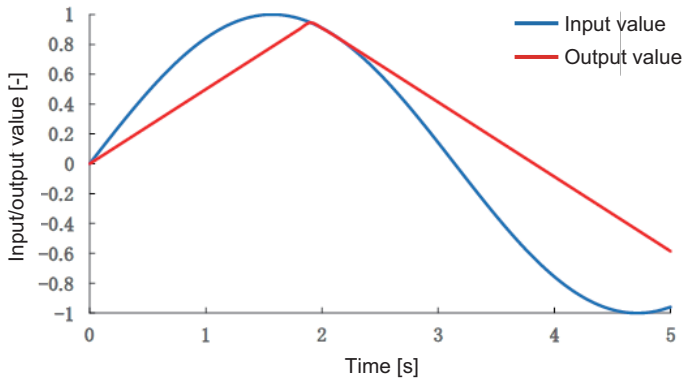
F (falling edge limit value), R (rising edge limit value), and initial output are set in the properties. (☞ Page 210 Properties)

Ex.

The following is a graph of the output values when sine wave signals are input for 5 seconds at 0.1 second intervals.

Rising edge limit value: 0.5

Falling edge limit value: -0.5



■I/O signals

The following table lists the I/O signals of the rate limiter block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input	Input	1	LREAL	Minimum value to maximum value of LREAL data type	—
output	Output	1	LREAL	Minimum value to maximum value of LREAL data type	—

■Properties

The following table lists the properties of the rate limiter block.

Property name	Attribute	Range/option	Default	Possible input value	Remarks
Rising edge limit value [-]	Text box	0 to maximum value of LREAL data type	0	One-byte numbers, one-byte alphabetic characters (E, e), symbols (+, -, .)	• 13 characters maximum • Exponential or decimal point notation can be used.
Falling edge limit value [-]	Text box	Minimum value of LREAL data type to 0	0	One-byte numbers, one-byte alphabetic characters (E, e), symbols (+, -, .)	• 13 characters maximum • Exponential or decimal point notation can be used.
Initial output	Text box	Minimum value to maximum value of LREAL data type	0	One-byte numbers, one-byte alphabetic characters (E, e), symbols (+, -, .)	• 13 characters maximum • Exponential or decimal point notation can be used.

■Error

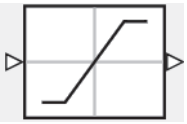
The following table lists errors that occur in the rate limiter block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	○	When output is outside the following range $\|output\| < 2^{1024}$	—
0 division	—	—	—

Point

An example of a system that uses a rate limiter block is a system that controls the number of connections to a network.

Saturation block



■Function details

This block outputs the value calculated using the formula below.

$$output = \begin{cases} input, & lower \leq input \leq upper \\ lower, & input < lower \\ upper, & input > upper \end{cases}$$

- output: Output value
- input: Input value
- lower: Lower limit value
- upper: Upper limit value

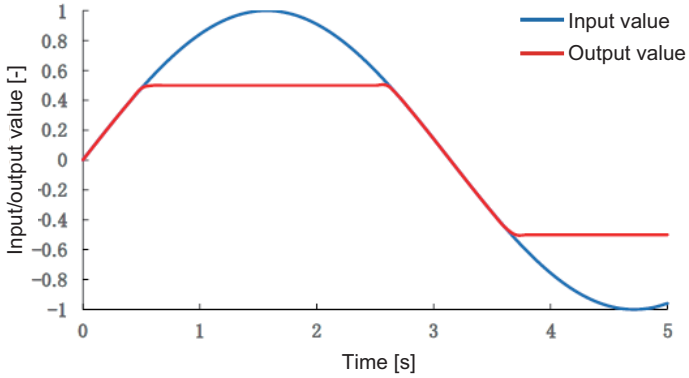
lower (lower limit value) and upper (upper limit value) are set in the properties. (☞ Page 211 Properties)

Ex.

The following is a graph of the output values when sine wave signals are input for 5 seconds at 0.1 second intervals.

Upper limit value: 0.5

Lower limit value: -0.5



■I/O signals

The following table lists the I/O signals of the saturation block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input	Input	1	LREAL	Minimum value to maximum value of LREAL data type	—
output	Output	1	LREAL	Minimum value to maximum value of LREAL data type	—

■Properties

The following table lists the properties of the saturation block.

Property name	Attribute	Range/option	Default	Possible input value	Remarks
Upper limit value [-]	Text box	Lower limit value [-] to maximum value of LREAL data type	0	One-byte numbers, one-byte alphabetic characters (E, e), symbols (+, -, .)	• 13 characters maximum • Exponential or decimal point notation can be used.
Lower limit value [-]	Text box	Minimum value of LREAL data type to Upper limit value [-]	0	One-byte numbers, one-byte alphabetic characters (E, e), symbols (+, -, .)	• 13 characters maximum • Exponential or decimal point notation can be used.

■Error

The following table lists errors that occur in the saturation block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	—	—	—
0 division	—	—	—

Point

An example of a system that uses a saturation block is a system that defines the output upper and lower limits.

Discrete time

Primary hold block

■Function details

This block interpolates inputs and outputs between blocks with different sample times.
The block outputs the value calculated using the formula below. Note that the initial step outputs the initial output.

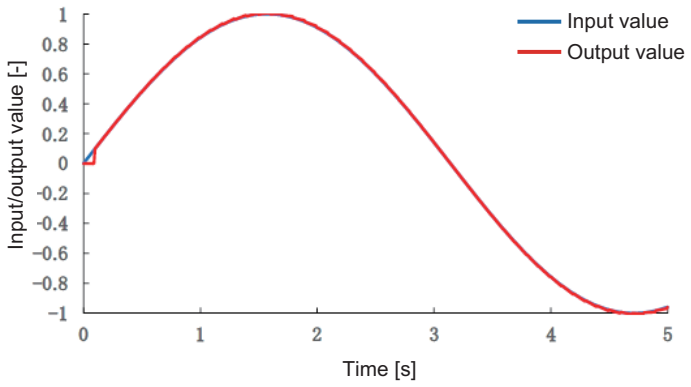
$$o_t = i_k + \frac{i_k - i_{k-1}}{T_k - T_{k-1}}(t - T_k)$$

- o_t: Output value at time t
- k: Any positive number representing the step number
- t: Any time that satisfies $T_k \leq t \leq T_{k+1}$
- T_k: Time at k step
- T_{k-1}: Time at k-1 step
- i_k: Input value at k step
- i_{k-1}: Input value at k-1 step

The input/output data type and initial output are set in the properties. (☞ Page 214 Properties)

Ex.

The following is a graph of the output values when a block that outputs sine wave signals at 0.1 second intervals and a block that outputs sine wave signals at 0.01 second intervals are interpolated using this block.



■I/O signals

The following table lists the I/O signals of the primary hold block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input	Input	1	LREAL WORD INT DWORD DINT	Minimum value to maximum value of each data type	—
output	Output	1	LREAL WORD INT DWORD DINT	Minimum value to maximum value of each data type	—

■Properties

The following table lists the properties of the primary hold block.

Property name	Attribute	Range/option	Default	Possible input value	Remarks
Type	Drop-down list	LREAL WORD INT DWORD DINT	LREAL	—	—
Initial output	Text box	Minimum value to maximum value of each data type	0	■LREAL data type • One-byte numbers (0 to 9) • One-byte alphabetic characters (E, e) • One-byte symbols (plus (+), hyphen (-), period (.)) ■Other data types • One-byte numbers (0 to 9) • One-byte alphabetic characters (A to F, a to f, o, x) • One-byte symbols (plus (+), hyphen (-))	• 13 characters maximum • For the notation that can be used, refer to the following. ☞ Page 127 Notation that can be used for properties

■Error

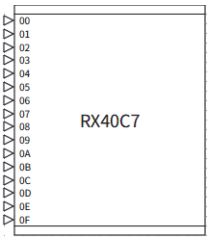
The following table lists errors that occur in the primary hold block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	○	When output is outside the following range $\|output\| < 2^{1024}$	—
0 division	—	—	—

Point

An example of a system that uses a primary hold block is a system that interpolates systems with different sample times.

Input block



■Function details

This block inputs signals to the input module.

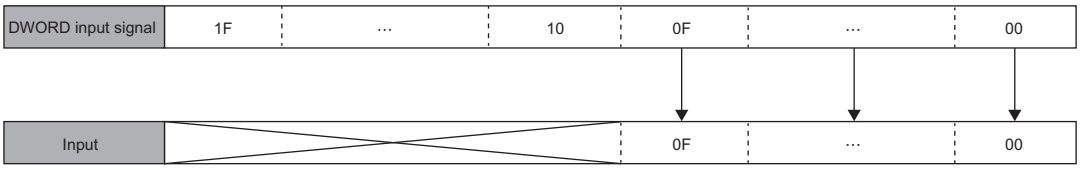
For details on each terminal of the input module, refer to the following.

📖MELSEC iQ-R I/O Module User's Manual

When the input module has 32 or fewer terminals and the Type property is set to DWORD, the lower bits of the input DWORD value are input to the input module terminals corresponding the input block and the other bits are discarded.

The following figure shows an example for a module with 16 terminals.

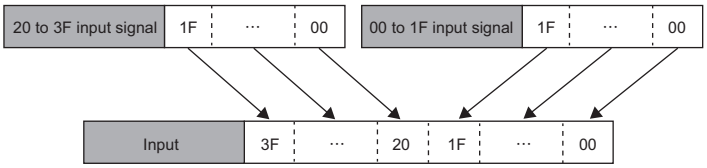
When assigning lower-order input signals sequentially starting from 00



When the input module has 32 or more terminals and the Type property is set to DWORD, the input signals for terminals 00 to 1F are treated as the lower bits and terminals 20 to 3F as the upper bits. These bits are combined into an unsigned 64-bit numerical value and input to the terminals corresponding to the input block. Unconnected inputs are treated as 0.

The following figure shows an example for a module with 64 terminals.

When assigning lower-order input signals sequentially starting from 00



■Applicable module

The following input modules are applicable. The model and the number of input signals shown at the center of the block diagram vary depending on the module.

RX28, RX40PC6H, RX40NC6H, RX40C7, RX40C7-TS, RX70C4, RX10, RX10-TS, RX41C4, RX41C4-TS, RX41C6HS, RX71C4, RX61C6HS, RX42C4, RX72C4

■I/O signals

The following table lists the I/O signals of the input block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input	Input	1	DWORD BOOL	DWORD data type • 0 to 4294967295 BOOL data type • FALSE, TRUE	The number of input signals depends on the module.*1

*1 The following table lists the number of input signals for the BOOL and DWORD data types.

Module	Number of signals (DWORD data type)	Number of signals (BOOL data type)
RX28	1	8
RX40PC6H, RX40NC6H, RX40C7, RX40C7-TS, RX70C4, RX10, RX10-TS	1	16
RX41C4, RX41C4-TS, RX41C6HS, RX71C4, RX61C6HS	1	32
RX42C4, RX72C4	2	64

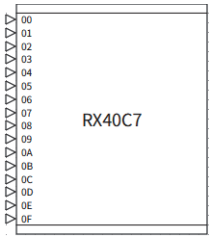
■Block diagram display

The block diagram display varies depending on the Type property setting and the number of module input signals.

Ex.

When the Type property is set to BOOL:

The terminals for the number of module input signals are displayed.



Ex.

When the Type property is set to DWORD:

The range of input signal numbers are displayed on the input signal block terminal.

For modules that have 32 or fewer input signals in BOOL, the number of input signals becomes 1 when the Type property is set to DWORD.

For modules that have 64 input signals in BOOL, the number of input signals becomes 2 when the Type property is set to DWORD.

■Properties

The following table lists the properties of the input block.

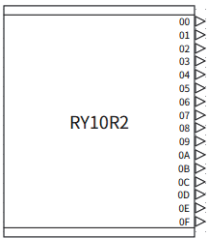
Property name	Attribute	Range/option	Default	Possible input value	Remarks
Type	Drop-down list	DWORD BOOL	BOOL	—	When the Type property is set to DWORD, memory data is copied rather than the value.

■Error

The following table lists errors that occur in the input block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	—	—	—
0 division	—	—	—

Output block



■Function details

This block outputs signals to the output module.

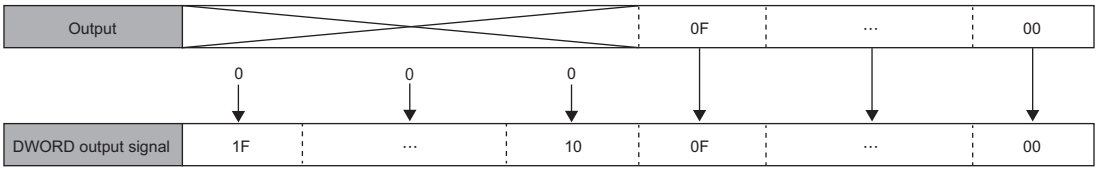
For details on each terminal of the output module, refer to the following.

📖MELSEC iQ-R I/O Module User's Manual

When the output module has 32 or fewer terminals and the Type property is set to DWORD, the value output from the output block is input to the DWORD bits sequentially, starting from the least significant bit. Any bits that do not correspond to a terminal are padded with zeros in DWORD.

The following figure shows an example for a module with 16 terminals.

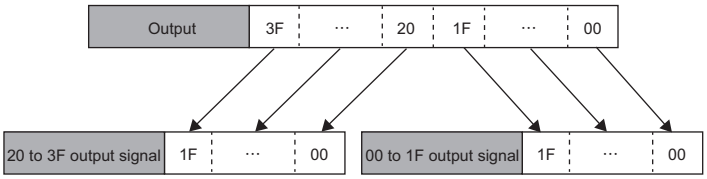
When assigning lower-order output signals sequentially starting from 00



When the output module has 32 or more terminals and the Type property is set to DWORD, the values output from the output block are treated as follows: 00 to 1F corresponds to the lower 32 bits and 20 to 3F corresponds to the upper 32 bits in DWORD.

The following figure shows an example for a module with 64 terminals.

When assigning lower-order output signals sequentially starting from 00



■Applicable module

The following output modules are applicable. The model and the number of output signals shown at the center of the block diagram vary depending on the module.

RY18R2A, RY40NT5P, RY40NT5P-TS, RY40PT5P, RY40PT5P-TS, RY10R2, RY10R2-TS, RY20S6, RY41NT2P, RY41NT2P-TS, RY41NT2H, RY41PT1P, RY41PT1P-TS, RY41PT2H, RY42NT2P, RY42PT1P

■I/O signals

The following table lists the I/O signals of the output block.

Signal name	Classification	Input/output order	Type	Range	Remarks
output	Output	1	DWORD BOOL	DWORD data type • 0 to 4294967295 BOOL data type • FALSE, TRUE	The number of output signals depends on the module.

*1 The following table lists the number of output signals for the BOOL and DWORD data types.

Module	Number of signals (DWORD data type)	Number of signals (BOOL data type)
RY18R2A	1	8
RY40NT5P, RY40NT5P-TS, RY40PT5P, RY40PT5P-TS, RY10R2, RY10R2-TS, RY20S6	1	16
RY41NT2P, RY41NT2P-TS, RY41NT2H, RY41PT1P, RY41PT1P-TS, RY41PT2H	1	32
RY42NT2P, RY42PT1P	2	64

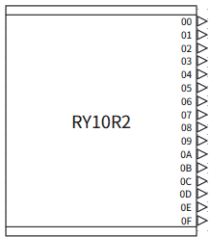
■Block diagram display

The block diagram display varies depending on the Type property setting and the number of module output signals.

Ex.

When the Type property is set to BOOL:

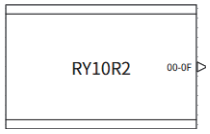
The terminals for the number of module output signals are displayed.



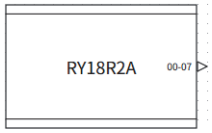
Ex.

When the Type property is set to DWORD:

The range of output signal numbers are displayed on the output signal block terminal.



For modules that have 32 or fewer output signals in BOOL, the number of output signals becomes 1 when the Type property is set to DWORD.



For modules that have 64 output signals in BOOL, the number of output signals becomes 2 when the Type property is set to DWORD.

■Properties

The following table lists the properties of the output block.

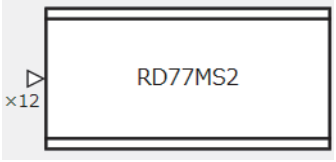
Property name	Attribute	Range/option	Default	Possible output value	Remarks
Type	Drop-down list	DWORD BOOL	BOOL	—	When the Type property is set to DWORD, memory data is copied rather than the value.

■Error

The following table lists errors that occur in the output block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	—	—	—
0 division	—	—	—

Simple Motion block (RD77MS2)



■Function details

This block sets the external inputs of the Simple Motion module and reflects them in the simulation.

For details on each terminal of the RD77MS2, refer to the following.

MELSEC iQ-R Simple Motion Module User's Manual (Startup)

■I/O signals

The following table lists the I/O signals of the Simple Motion block (RD77MS2).

Signal name	Classification	Input/output order	Type	Range	Remarks
SIN1 to SIN10	Input	1 to 10	BOOL	FALSE, TRUE	The following external input signals are input according to the Simple Motion module settings. • Upper/lower limit signal (FLS/RLS) • Near-point dog signal (DOG) • Stop signal (STOP) Input logic selection can be set to positive logic or negative logic.
EMI	Input	11	BOOL	FALSE, TRUE	An emergency stop signal is input according to the Simple Motion module settings. Input logic selection can be set only to negative logic.
ENC	Input	12	DINT	Minimum value to maximum value of DINT data type	A pulse signal generated from a manual pulse generator or INC synchronous encoder is input according to the Simple Motion module settings. (Unit: pulse/s) The operating range can be set to one of the following in the Simple Motion module. • -200000 to 200000 • -400000 to 400000 • -800000 to 800000 • -1000000 to 1000000 If the value exceeds the above operating range, it will be treated as the upper limit value. If the value falls below the range, it will be treated as the lower limit value.

■Properties

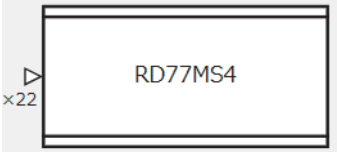
The Simple Motion block (RD77MS2) has no properties.

■Error

The following table lists errors that occur in the Simple Motion block (RD77MS2).

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	—	—	—
0 division	—	—	—

Simple Motion block (RD77MS4)



■Function details

This block sets the external inputs of the Simple Motion module and reflects them in the simulation.

For details on each terminal of the RD77MS4, refer to the following.

MELSEC iQ-R Simple Motion Module User's Manual (Startup)

■I/O signals

The following table lists the I/O signals of the Simple Motion block (RD77MS4).

Signal name	Classification	Input/output order	Type	Range	Remarks
SIN1 to SIN20	Input	1 to 20	BOOL	FALSE, TRUE	The following external input signals are input according to the Simple Motion module settings. • Upper/lower limit signal (FLS/RLS) • Near-point dog signal (DOG) • Stop signal (STOP) Input logic selection can be set to positive logic or negative logic.
EMI	Input	21	BOOL	FALSE, TRUE	An emergency stop signal is input according to the Simple Motion module settings. Input logic selection can be set only to negative logic.
ENC	Input	22	DINT	Minimum value to maximum value of DINT data type	A pulse signal generated from a manual pulse generator or INC synchronous encoder is input according to the Simple Motion module settings. (Unit: pulse/s) The operating range can be set to one of the following in the Simple Motion module. • -200000 to 200000 • -400000 to 400000 • -800000 to 800000 • -1000000 to 1000000 If the value exceeds the above operating range, it will be treated as the upper limit value. If the value falls below the range, it will be treated as the lower limit value.

■Properties

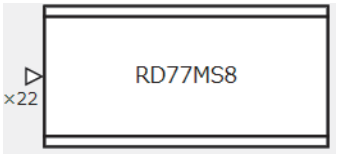
The Simple Motion block (RD77MS4) has no properties.

■Error

The following table lists errors that occur in the Simple Motion block (RD77MS4).

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	—	—	—
0 division	—	—	—

Simple Motion block (RD77MS8)



■Function details

This block sets the external inputs of the Simple Motion module and reflects them in the simulation.

For details on each terminal of the RD77MS8, refer to the following.

MELSEC iQ-R Simple Motion Module User's Manual (Startup)

■I/O signals

The following table lists the I/O signals of the Simple Motion block (RD77MS8).

Signal name	Classification	Input/output order	Type	Range	Remarks
SIN1 to SIN20	Input	1 to 20	BOOL	FALSE, TRUE	The following external input signals are input according to the Simple Motion module settings. • Upper/lower limit signal (FLS/RLS) • Near-point dog signal (DOG) • Stop signal (STOP) Input logic selection can be set to positive logic or negative logic.
EMI	Input	21	BOOL	FALSE, TRUE	An emergency stop signal is input according to the Simple Motion module settings. Input logic selection can be set only to negative logic.
ENC	Input	22	DINT	Minimum value to maximum value of DINT data type	A pulse signal generated from a manual pulse generator or INC synchronous encoder is input according to the Simple Motion module settings. (Unit: pulse/s) The operating range can be set to one of the following in the Simple Motion module. • -200000 to 200000 • -400000 to 400000 • -800000 to 800000 • -1000000 to 1000000 If the value exceeds the above operating range, it will be treated as the upper limit value. If the value falls below the range, it will be treated as the lower limit value.

■Properties

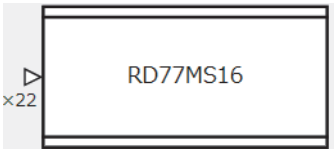
The Simple Motion block (RD77MS8) has no properties.

■Error

The following table lists errors that occur in the Simple Motion block (RD77MS8).

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	—	—	—
0 division	—	—	—

Simple Motion block (RD77MS16)



■Function details

This block sets the external inputs of the Simple Motion module and reflects them in the simulation.

For details on each terminal of the RD77MS16, refer to the following.

📖MELSEC iQ-R Simple Motion Module User's Manual (Startup)

■I/O signals

The following table lists the I/O signals of the Simple Motion block (RD77MS16).

Signal name	Classification	Input/output order	Type	Range	Remarks
SIN1 to SIN20	Input	1 to 20	BOOL	FALSE, TRUE	The following external input signals are input according to the Simple Motion module settings. • Upper/lower limit signal (FLS/RLS) • Near-point dog signal (DOG) • Stop signal (STOP) Input logic selection can be set to positive logic or negative logic.
EMI	Input	21	BOOL	FALSE, TRUE	An emergency stop signal is input according to the Simple Motion module settings. Input logic selection can be set only to negative logic.
ENC	Input	22	DINT	Minimum value to maximum value of DINT data type	A pulse signal generated from a manual pulse generator or INC synchronous encoder is input according to the Simple Motion module settings. (Unit: pulse/s) The operating range can be set to one of the following in the Simple Motion module. • -200000 to 200000 • -400000 to 400000 • -800000 to 800000 • -1000000 to 1000000 If the value exceeds the above operating range, it will be treated as the upper limit value. If the value falls below the range, it will be treated as the lower limit value.

■Properties

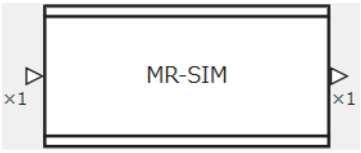
The Simple Motion block (RD77MS16) has no properties.

■Error

The following table lists errors that occur in the Simple Motion block (RD77MS16).

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	—	—	—
0 division	—	—	—

Servo amplifier block



■Function details

This block inputs the synchronous encoder of the servo amplifier and monitors the current value of the servo amplifier input axis.

■I/O signals

The following table lists the I/O signals of the servo amplifier block.

Signal name	Classification	Input/output order	Type	Range	Remarks
input	Input	1	DINT	Minimum value to maximum value of DINT data type	A pulse signal generated from a synchronous encoder is input. (Unit: pulse/s) Input values are limited to the following range. • -1000000 to 1000000 If the value exceeds the above range, it will be treated as the upper limit value. If the value falls below the range, it will be treated as the lower limit value.
output	Output	1	LREAL	Minimum value to maximum value of LREAL data type	Either of the following value set in the property is output. • Machine feed value • Command speed

■Properties

The following table lists the properties of the servo amplifier block.

Property name	Attribute	Range/option	Default	Possible input value	Remarks
Output	Drop-down list	• Machine feed value • Command speed	Machine feed value	—	—

■Error

The following table lists errors that occur in the servo amplifier block.

Error type	Error occurrence	Occurrence condition	Remarks
Out of range	—	—	—
Output overflow	—	—	—
0 division	—	—	—

10.4 Example Procedure for Using Blocks

This section describes an example procedure for using blocks.

FMU input block and FMU output block

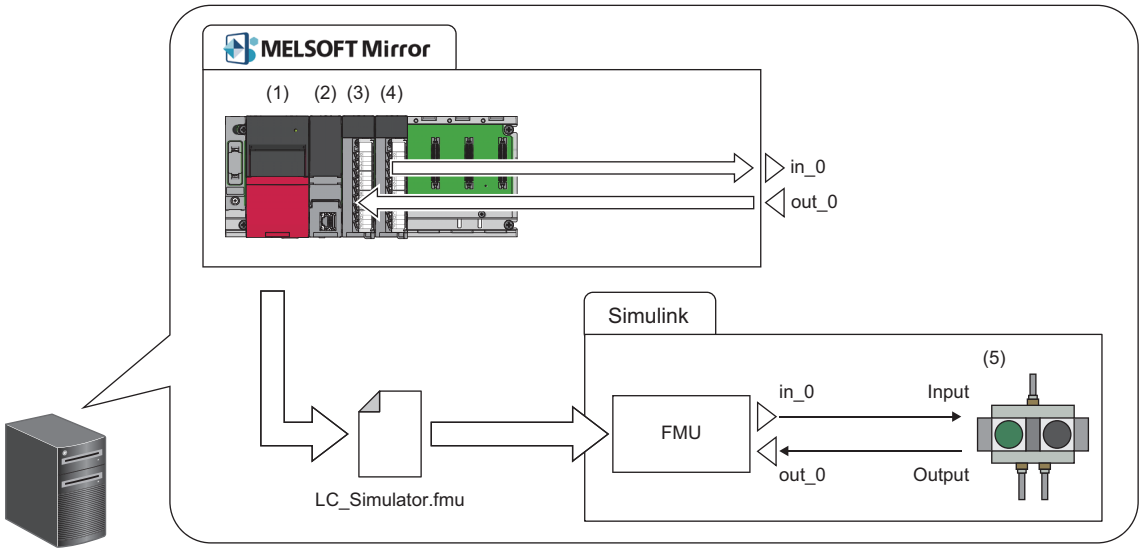
The following shows the procedure for using the FMU input block and the FMU output block.

System configuration example

Use Simulink manufactured by MathWorks as the FMI master.
Generate a FMU that processes the MELSOFT Mirror input and output (in_0 and out_0) using the FMU export function, then import the FMU into Simulink using the Simulink FMU import function. This embeds MELSOFT Mirror as one block inside Simulink and enables co-simulation between MELSOFT Mirror and Simulink.

■System configuration

- Main base unit: R35B
- Power supply module: R61P
- CPU module: R04CPU
- Input module: RX40C7
- Output module: RY40NT5P
- Valve with sensor: It is assumed that the sensor output is produced immediately when the valve opens in response to an input signal. (Valve dynamics are not considered.)

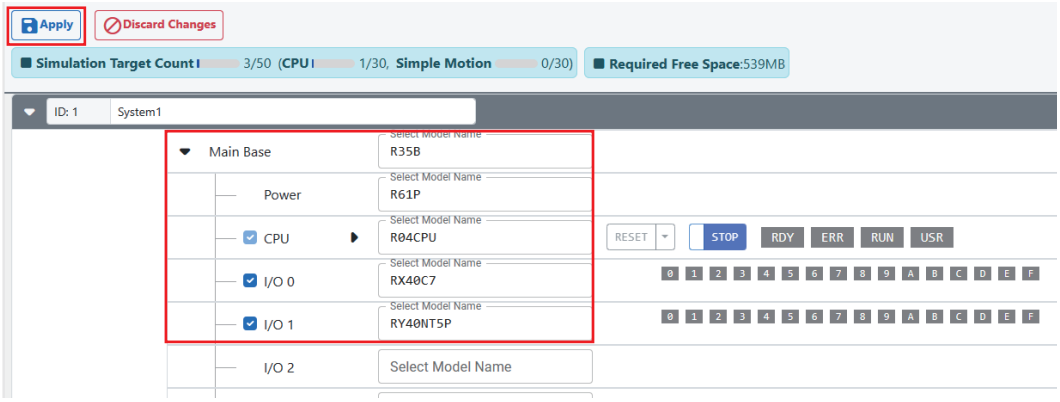


(1) Power supply module
(2) CPU module
(3) Input module
(4) Output module
(5) Value with sensor

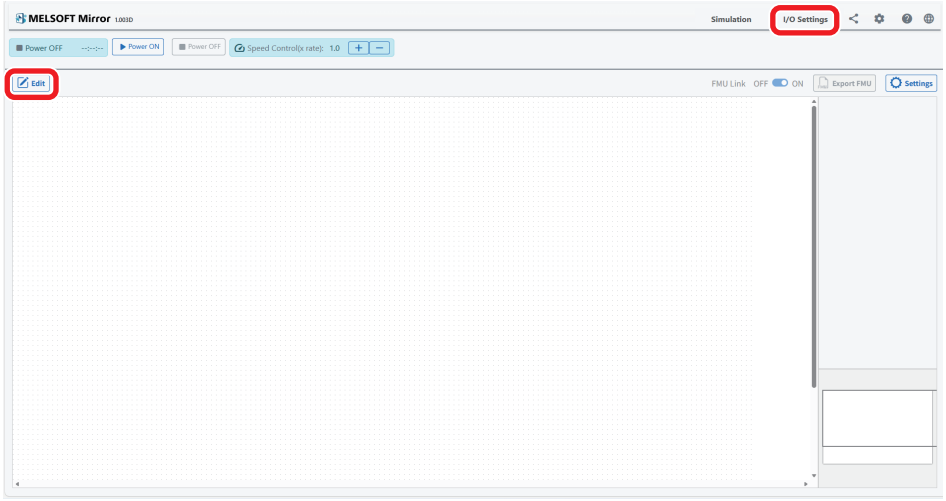
Setting system configuration

Set the programmable controller system configuration on MELSOFT Mirror.

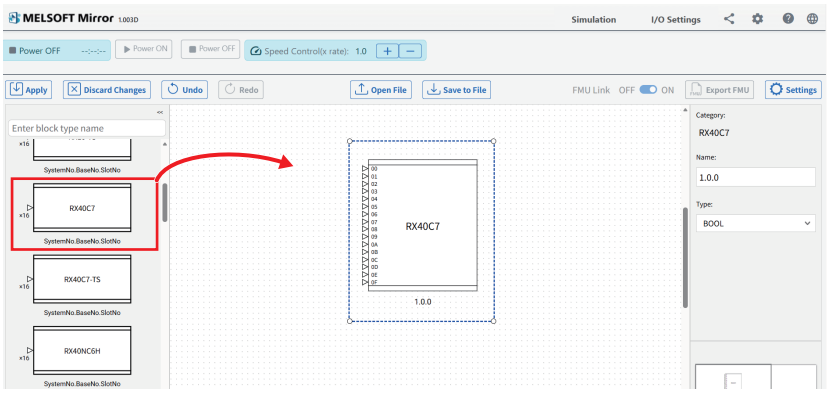
1. Click the [Edit] button.
2. Configure the following settings in the simulation setting window, and click the [Apply] button.



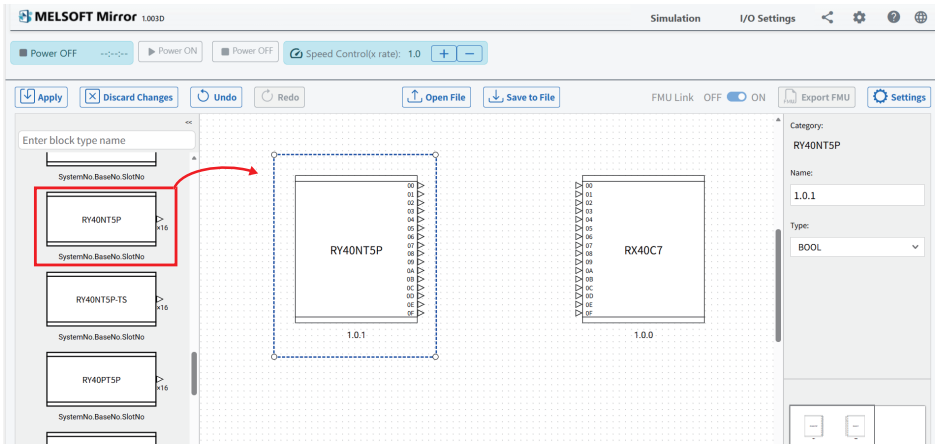
3. Click "I/O Settings" to open the "I/O Settings" window, and click the [Edit] button.



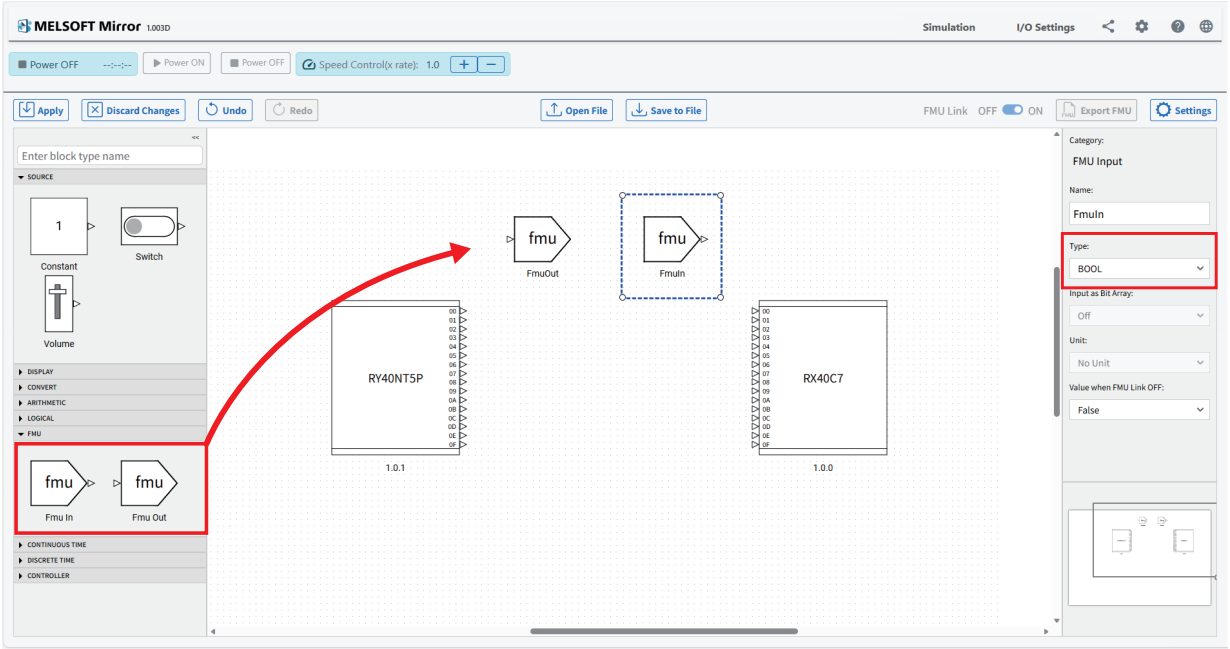
4. Select "RX40C7" from the block list on the left, and drag and drop it to the block diagram in the center. Set the block name to "1.0.0".



5. Select "RY40NT5P" from the block list on the left, and drag and drop it to the block diagram in the center. Set the block name to "1.0.1".



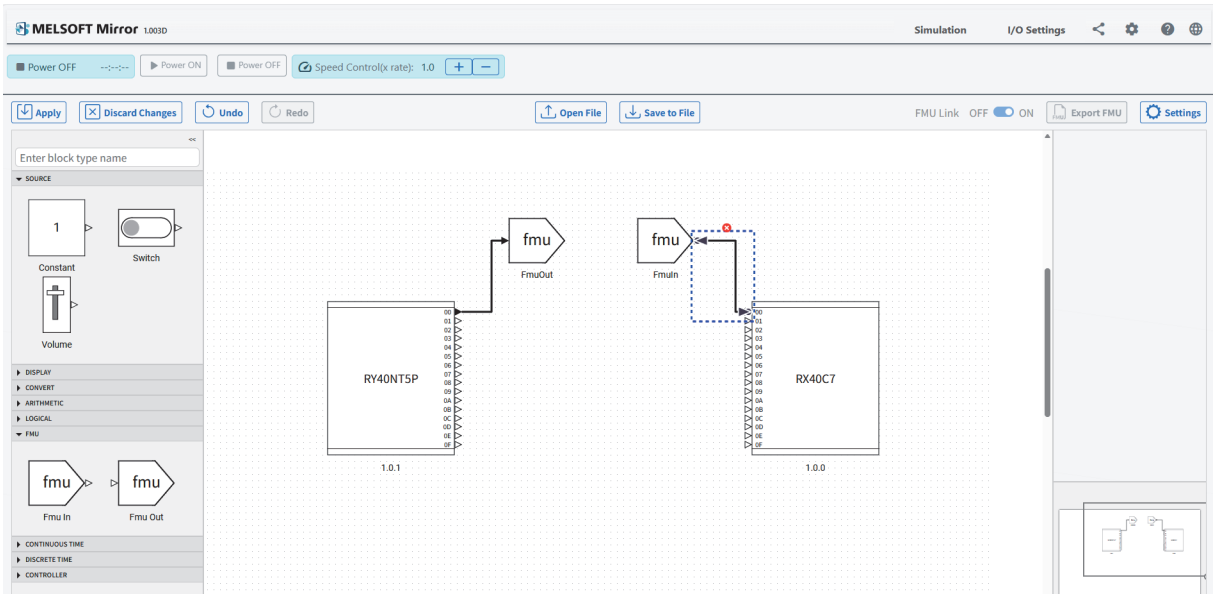
6. Select "Fmu In" and "Fmu Out" from the block list on the left, and drag and drop them to the block diagram area. Change the Type properties of the FmuOut and FmuIn blocks to BOOL.



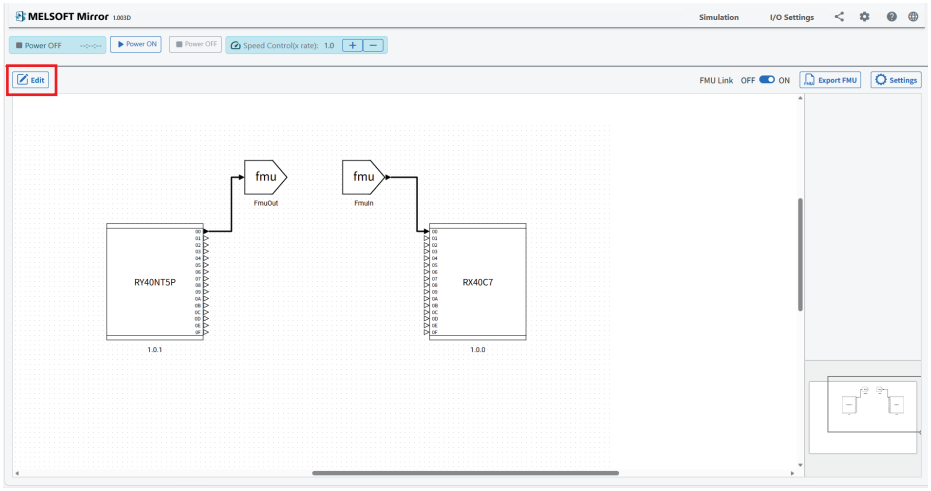
7. Connect the terminals by the drag and drop as shown below.

- Terminal 00 (Y10) of RY40NT5P and FmuOut terminal
- Terminal 00 (X0) of RX40C7 and FmuIn terminal

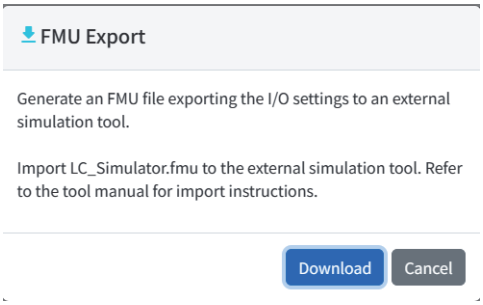
The signal from the terminal 00 (Y10) of the RY40NT5P is used as the FMU output and the signal from the terminal 00 (X0) of the RX40C7 is used as the FMU input.



8. Click the [Apply] button.

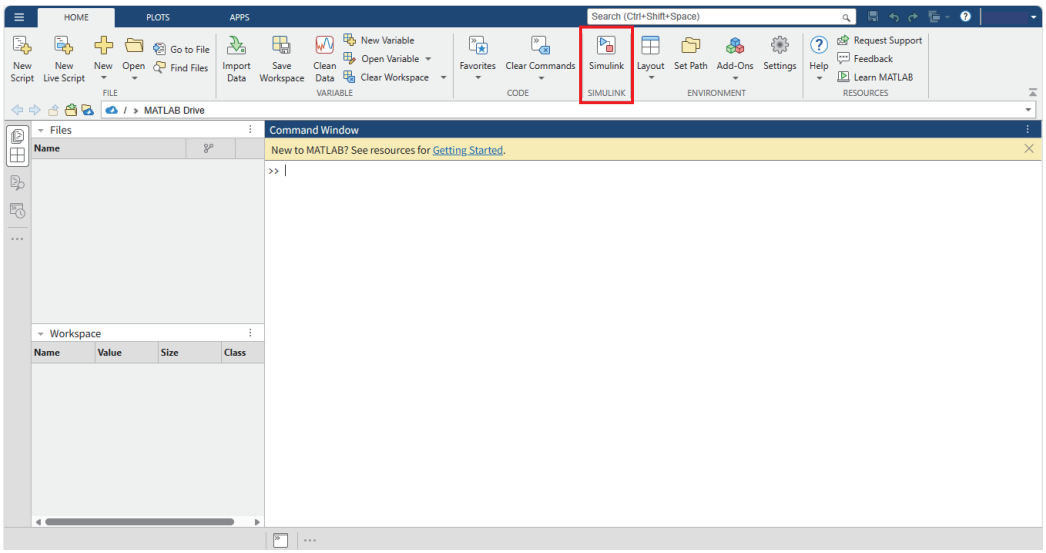


9. Click the [FMU Export] button, and then click the [Download] button. LC_Simulator.fmu is downloaded according to your browser settings.

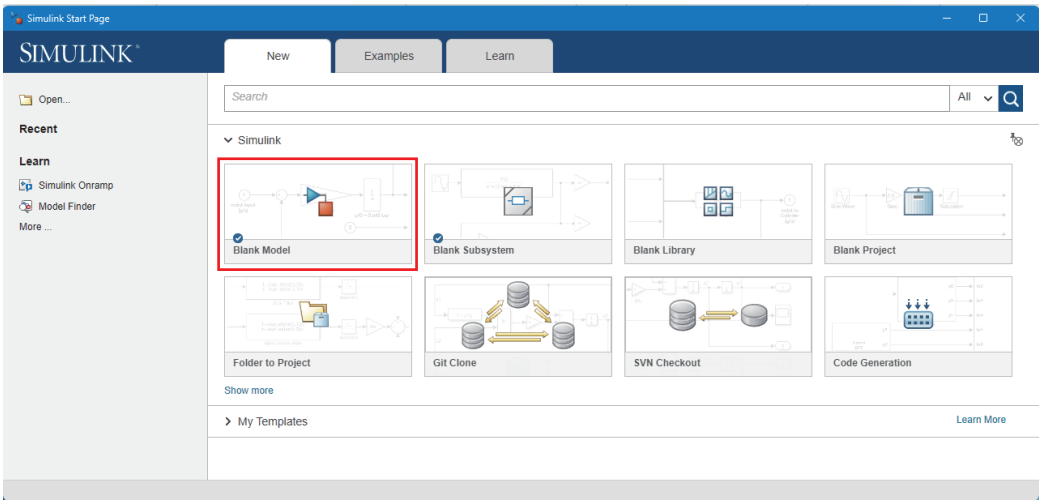


Setting Simulink

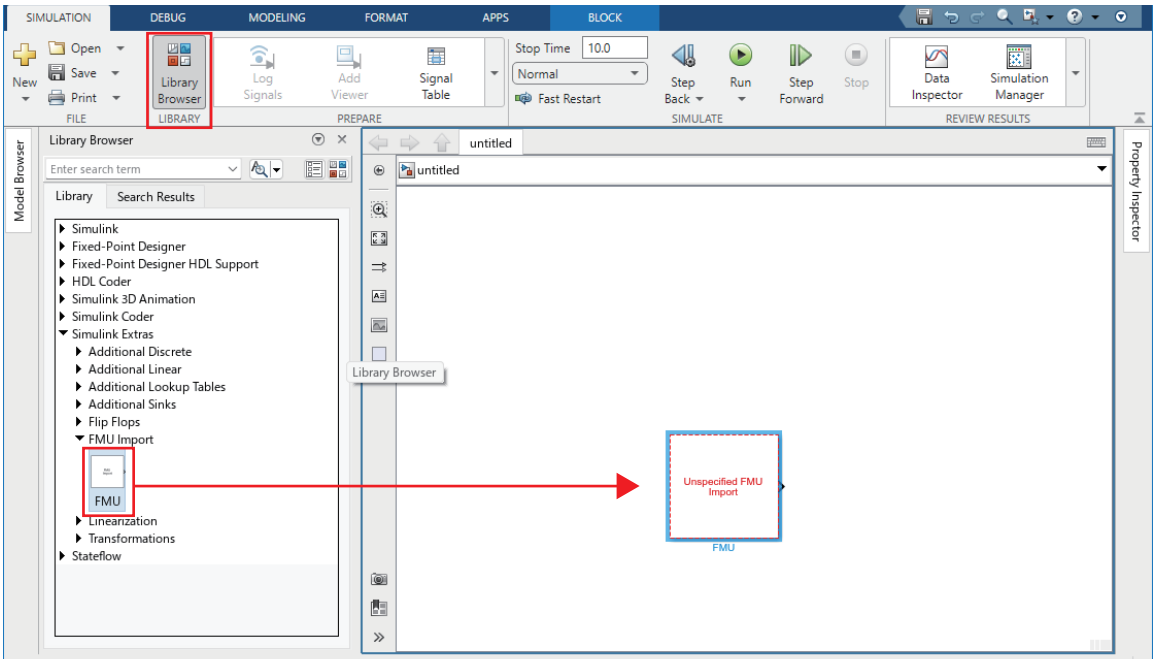
1. Start MATLAB and click [Simulink] in the Toolstrip.



2. In the "Simulink Start Page", click [Blank Model].

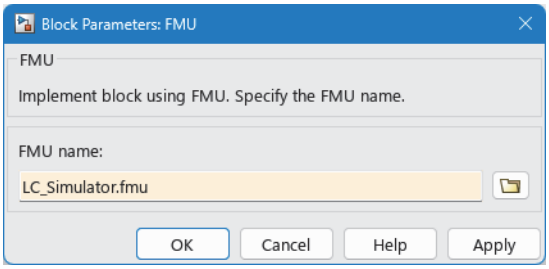


3. In the "Library Browser" panel, click [Simulink Extras] and open [FMU Import]. Drag and drop "FMU" to the canvas.



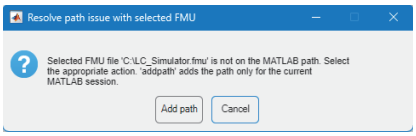
4. Right-click the placed FMU and select [Block Parameters].

5. In the "Block Parameters" dialog, specify the FMU exported from MELSOFT Mirror and click the [OK] button.

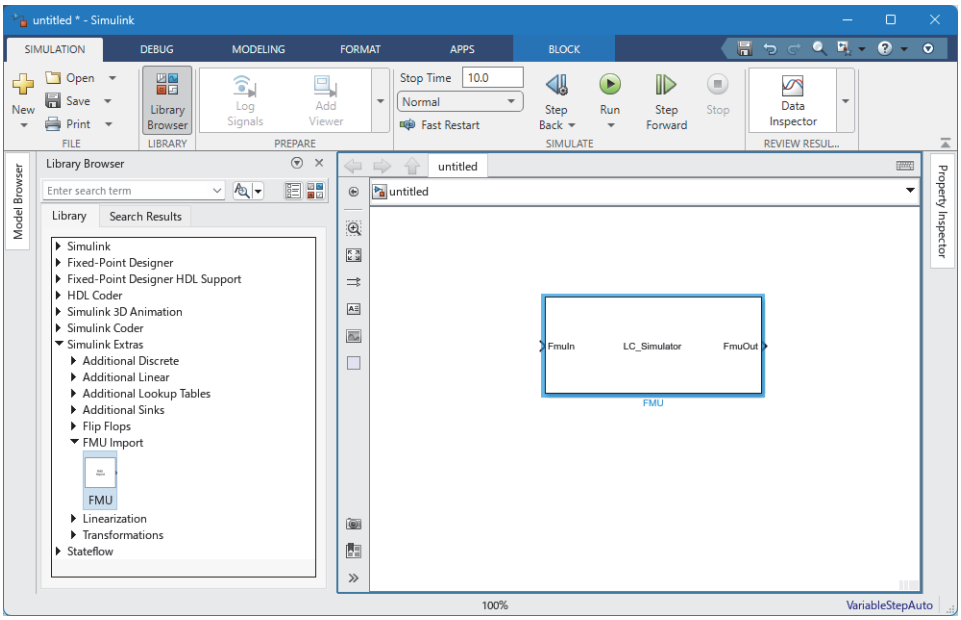


Point

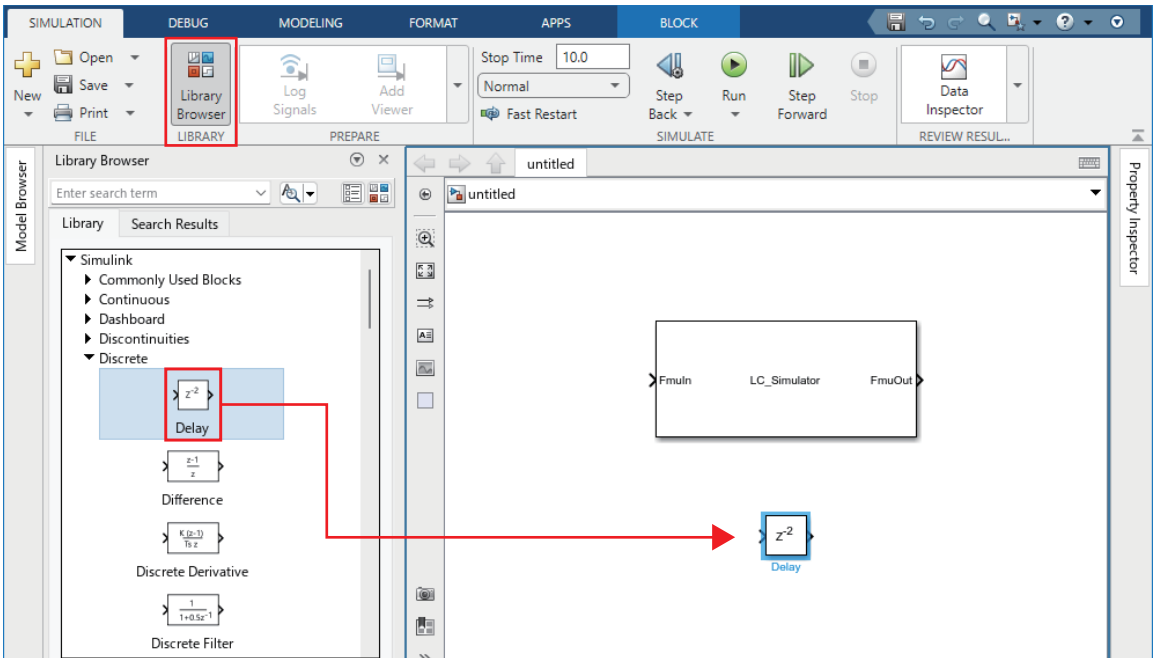
If a confirmation dialog appears asking to add a file to the MATLAB path, click the [Add path] button.



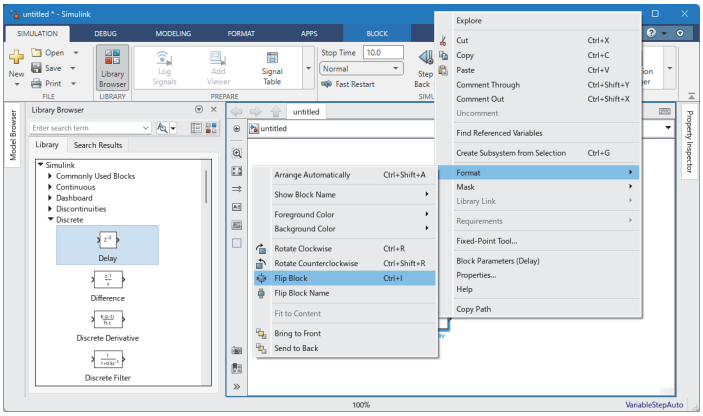
6. The block label changes to "LC_Simulator" and the input/output terminals defined in the MELSOFT Mirror FMU are set.



7. In the "Library Browser" panel, click [Simulink] and open [Discrete]. Drag and drop "Delay" to the canvas. You can flip the Delay block horizontally.

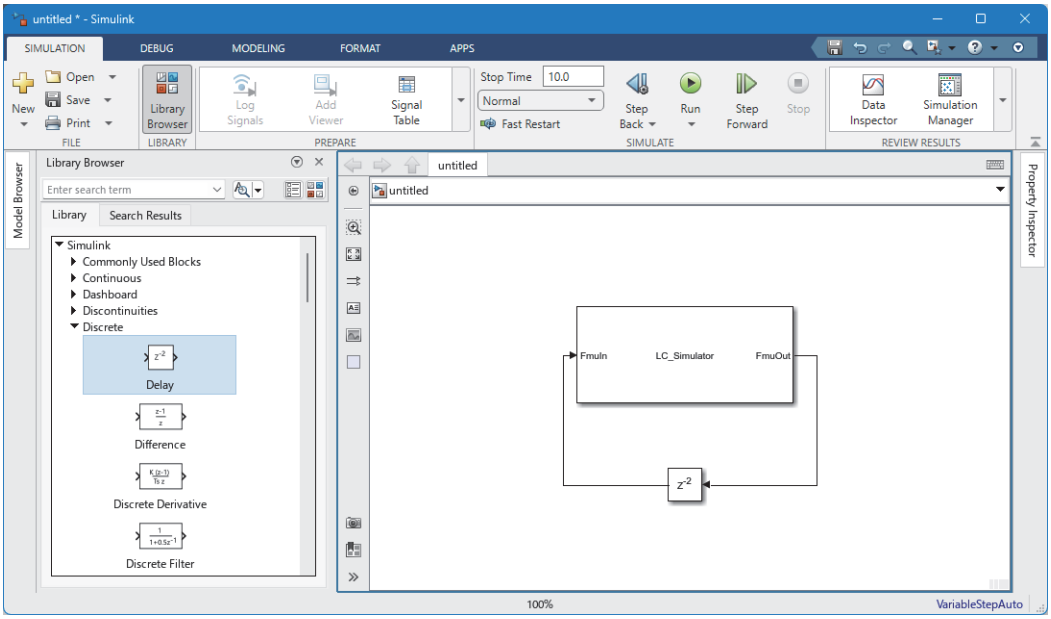


8. To flip it, right-click the Delay block → "Format" → "Flip Block".



9. Connect the terminals of the LC_Simulator block and the Delay block by the drag and drop as shown below.

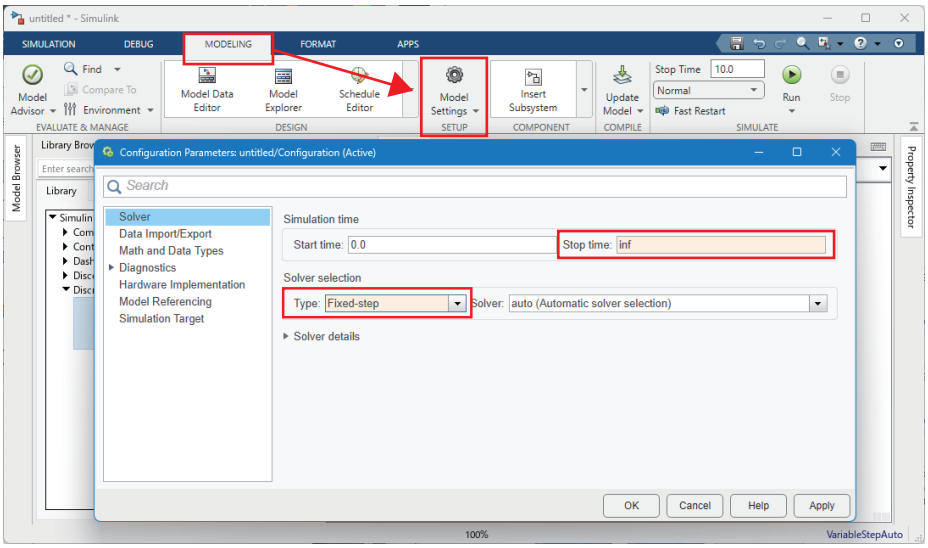
- FmuOut terminal of LC_Simulator block and input terminal of Delay block
- FmuIn terminal of LC_Simulator block and output terminal of Delay block



10. In the "Configuration Parameters" window, configure the settings as follows and click the [Apply] button.

Click the [MODELING] tab ⇨ [Model Settings] in the Simulink Toolstrip.

Setting item	Setting value
Stop time	inf
Type	Fixed-step

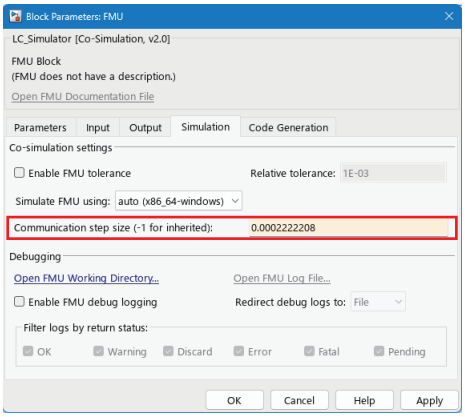


11. Double-click the FMU block and open the "Block Parameters" window.

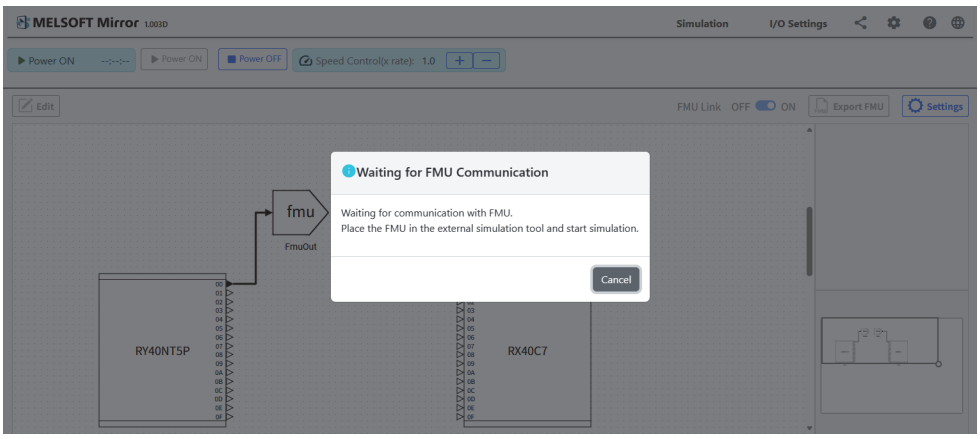
12. In the [Simulation] tab, configure the setting as follows.

Setting item	Setting value
Communication step size	0.0002222208*1

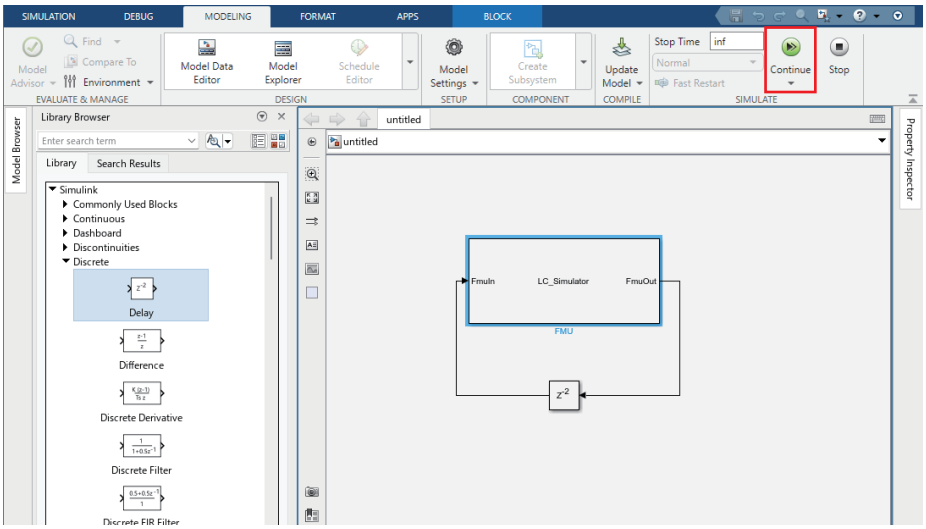
*1 Values equal to or larger than 0.000000000001 can be set, however, a value other than 0.0002222208 may produce communication steps that MELSOFT Mirror does not handle. Thus, 0.0002222208 is recommended.



13. Click the [Power ON] button on MELSOFT Mirror. (The "Waiting for FMU Communication" dialog appears. This dialog will automatically close when the Simulink simulation runs.)

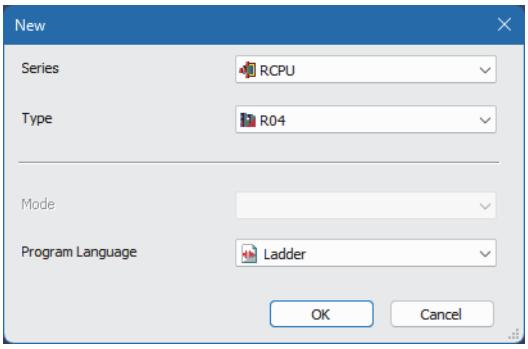


14. In Simulink, click [Continue] on the Simulink Toolstrip. The simulation will run simultaneously in both MELSOFT Mirror and Simulink.

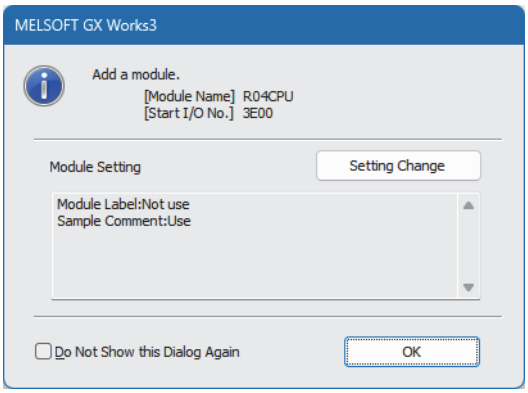


Creating a project with the engineering tool and writing data to MELSOFT Mirror

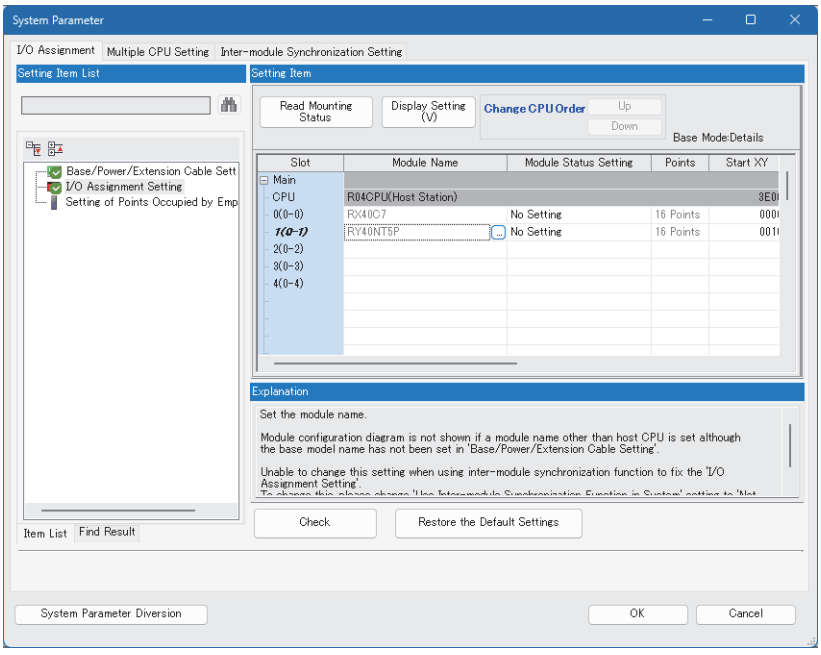
1. Create a project using the engineering tool. Set the following in the window for creating a new project.



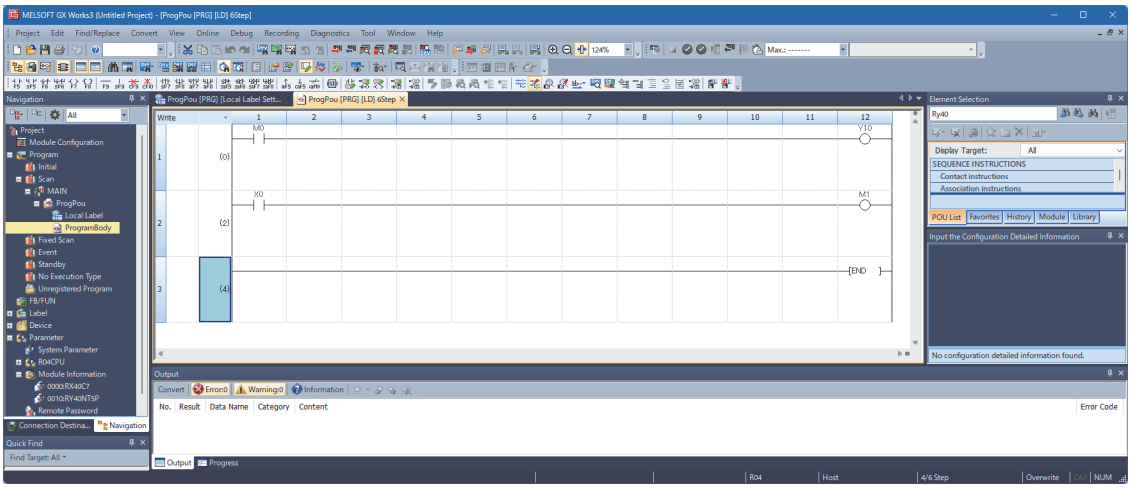
2. When the window for adding a module appears, click the [OK] button.



3. Place the following modules in the "System Parameter" window.



4. Write contacts and coils, as shown below, on the ladder program.



5. Select [Convert] ⇨ [Convert] on the menu bar to determine the ladder blocks.

6. Connect the engineering tool to MELSOFT Mirror. ☞ Page 61 Connecting the Engineering Tool to MELSOFT Mirror

7. Write the parameters and program to the R04CPU. ☞ Page 62 Writing Data to the Programmable Controller

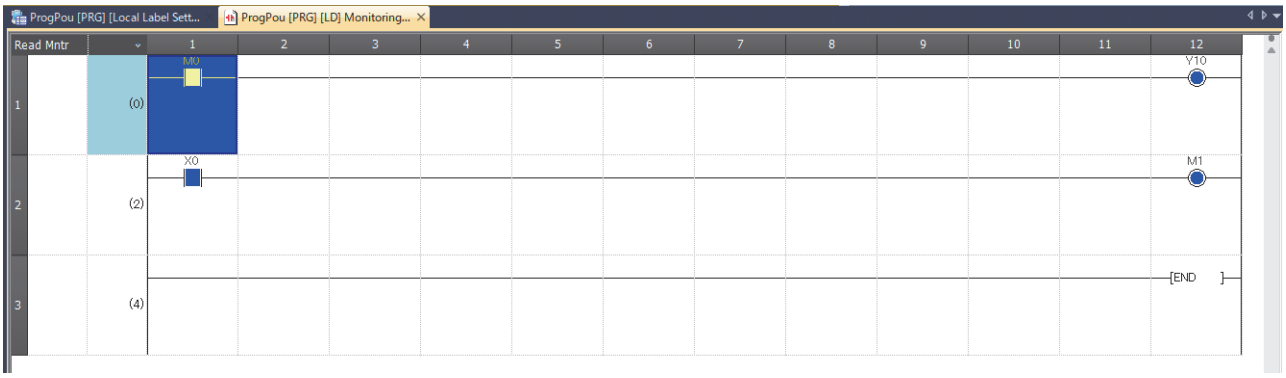
Executing the program

1. Reset the R04CPU, and set the switch to RUN on MELSOFT Mirror.

2. The engineering tool starts the program monitor. ☞ Page 64 Monitoring the Program

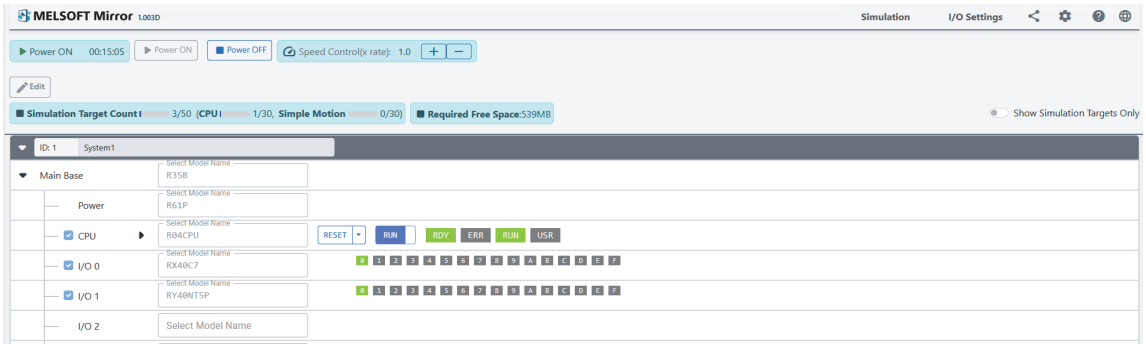
3. The program is executed as follows according to the ladder logic:

- When M0 turns on, Y10 turns on.
- When Y10 turns on and connection was made in the I/O settings, FmuOut in MELSOFT Mirror turns on.
- When FmuOut in MELSOFT Mirror turns on and connection was made with Simulink, FmuIn turns on.
- When FmuIn in MELSOFT Mirror turns on and connection was made in the I/O settings, X0 turns on.
- When X0 turns on, M1 turns on.



Point

The states of X0 (terminal 00 of the RX40C7) and Y10 (terminal 00 of the RY40NT5P) can be checked using the LEDs in the MELSOFT Mirror simulation setting window.



11 SLMP CONNECTION

SLMP connection allows SLMP communications using the built-in Ethernet function of a CPU module.

11.1 SLMP-Compatible CPU Module

The following table lists the SLMP-compatible CPU modules.

The SLMP-compatible CPU modules are the CPU modules that can be simulated and have the SLMP function.

Model name
R04CPU
R08CPU
R16CPU
R32CPU
R120CPU
R08PCPU
R16PCPU
R32PCPU
R120PCPU

11.2 SLMP Client Specifications

This section describes the specifications of external devices (SLMP clients).

Number of connectable SLMP clients

A maximum of 10 SLMP clients can be connected simultaneously to MELSOFT Mirror.

If 11 or more SLMP clients are connected simultaneously, the communication speed may decrease or other operations may not be performed properly.

Protocols

The following table lists the protocols that can be used for SLMP communications.

Protocol
TCP
UDP

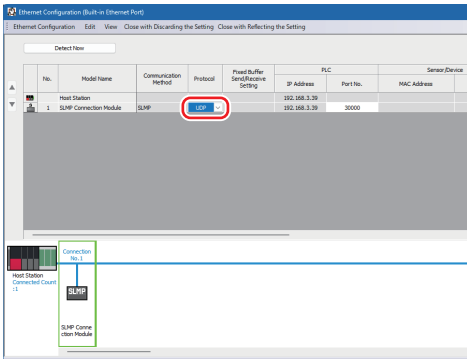
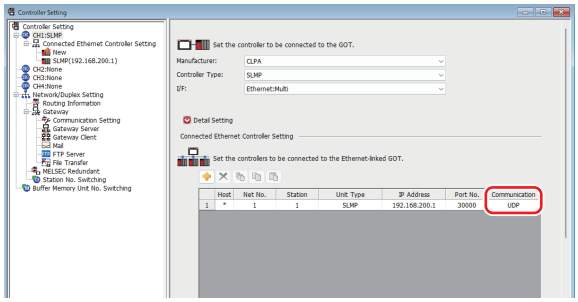
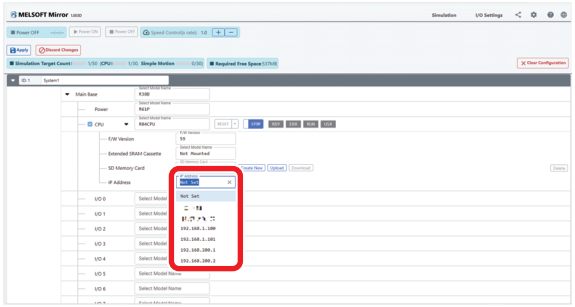
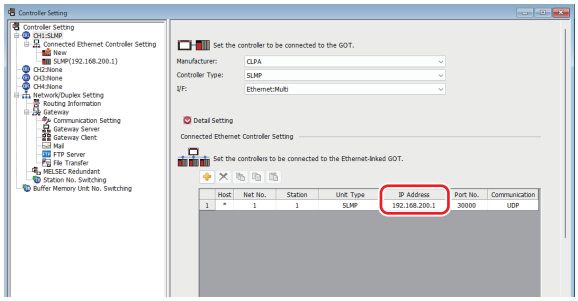
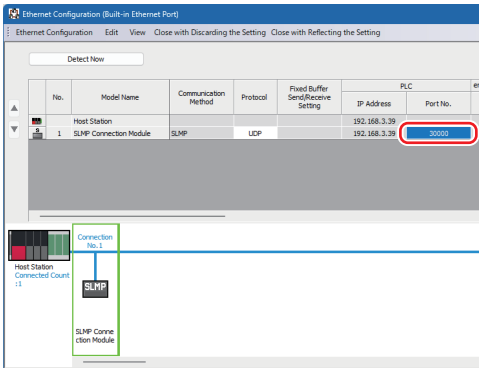
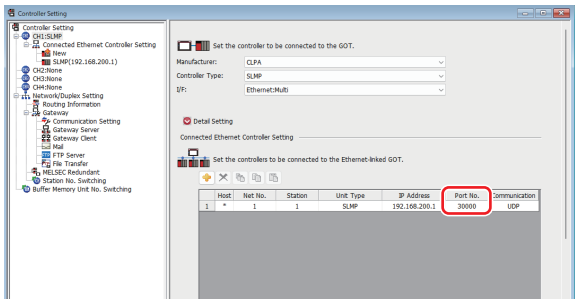
Frame types and communication data codes

The following table lists the frame types and communication data codes that can be used for SLMP communications.

Frame type	Communication data code
3E	Binary
	ASCII
4E	Binary
	ASCII

11.3 Connection Method

This section describes the settings required to perform SLMP communications between the CPU module and an external device (SLMP client). Set the same protocol, IP address, and port number to the CPU module and the external device (SLMP client).

Setting item	CPU module	External device (SLMP client) (The GOT settings are shown below as an example.)
Protocol	Set a protocol used for SLMP communications. Set the protocol of the CPU module in the following window. Select "Module Parameter" in the Navigation window of GX Works3. Select "External Device Configuration" to open the configuration window. Add a "SLMP Connection Module", and set "Protocol". 	Set a protocol used for SLMP communications. Set the same protocol as the CPU module shown on the left. 
IP address	Set an IP address of the CPU module on MELSOFT Mirror. The IP address set here will be the destination IP address of the external device (SLMP client). SLMP communications cannot be performed without this setting. Set the IP address of the CPU module in the following window. Access MELSOFT Mirror from a web browser, and click the [Edit] button. Select a CPU module, and select an IP address from the drop-down list. The IP addresses set in the simulation server are displayed in the list. 	Set a destination IP address for SLMP communications. Set the same IP address as the CPU module shown on the left. 
Port number	Set a port number used for SLMP communications. Set the port number in the following window. Select "Module Parameter" in the Navigation window of GX Works3. Select "External Device Configuration" to open the configuration window, and set "Port No." of "SLMP Connection Module". 	Set a port number used for SLMP communications. Set the same port number as the CPU module shown on the left. 

Precautions

Set an IP address for each CPU module.

When multiple CPU modules are used, set multiple IP addresses to the personal computer on which MELSOFT Mirror has been installed.

For the procedure, refer to the following.

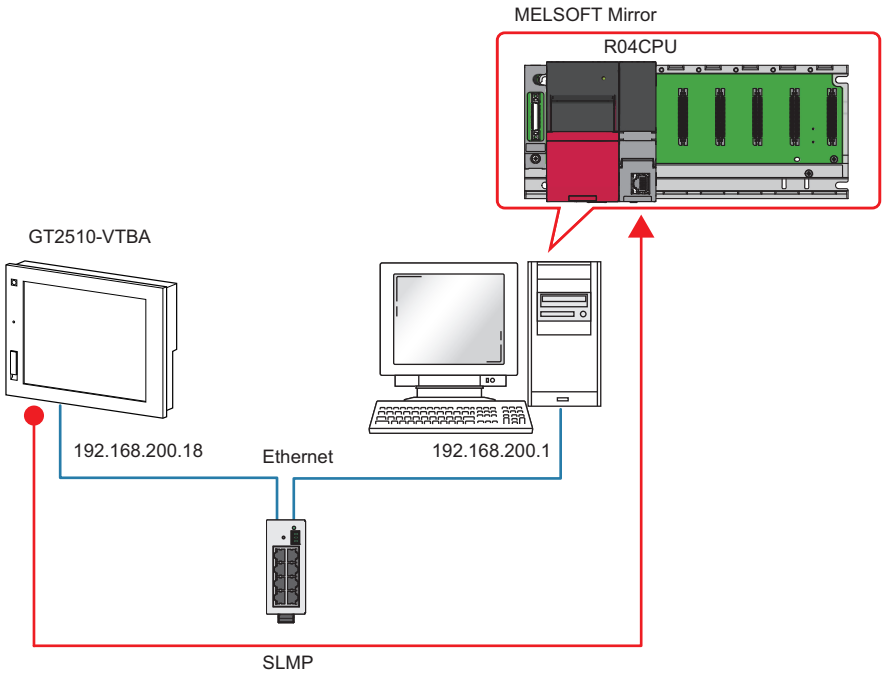
☞ Page 28 Setting Multiple IP Addresses on MELSOFT Mirror

Connection example

The following is an example of the SLMP connection. In this example, GOT performs SLMP communications with the CPU module simulated on MELSOFT Mirror.

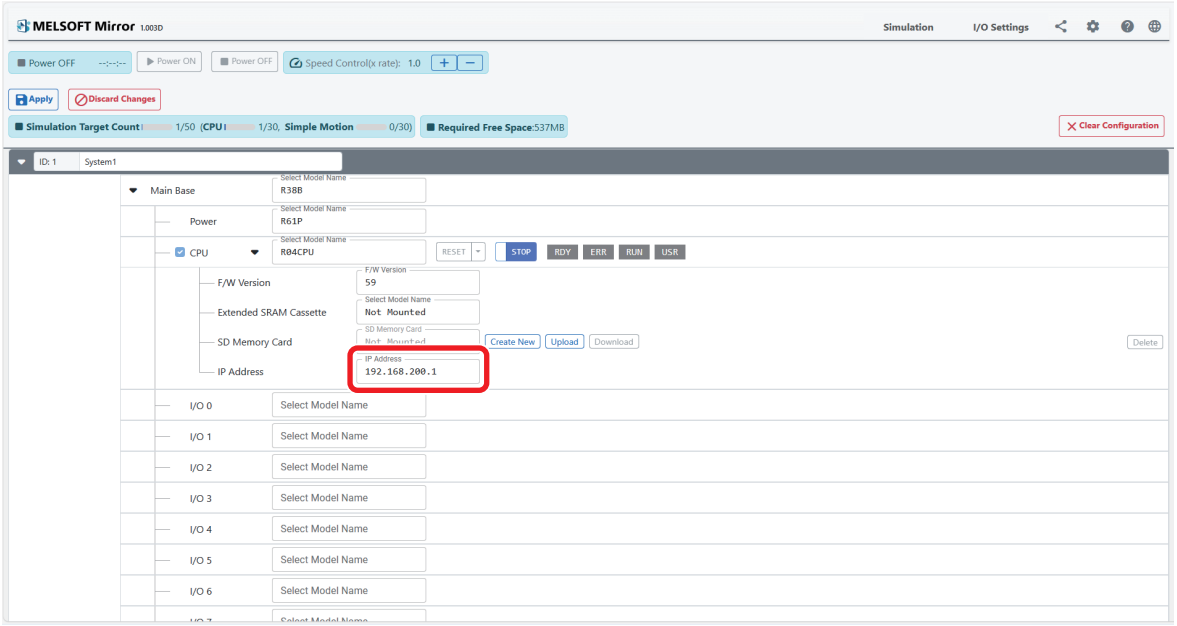
System configuration

The following is the system configuration for the SLMP communications.



Connection procedure

1. Select "R04CPU" on the following window of MELSOFT Mirror Management Tool, and set "192.168.200.1" for the IP address.

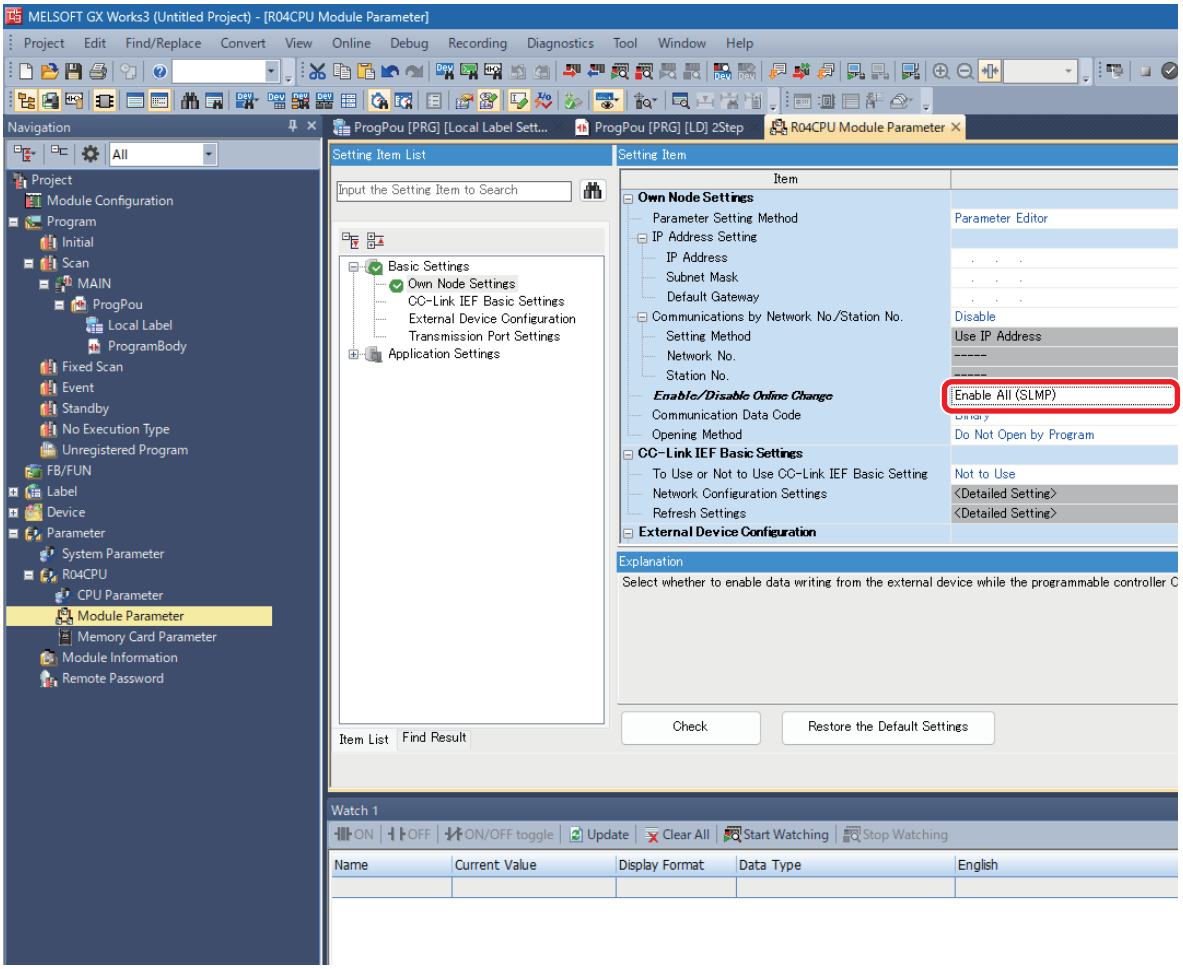


2. Click the [Apply] button. Then, click the [Power ON] button.

3. Create a project using GX Works3. Set the following in the window for creating a new project.

- Type: R04
- Program Language: Ladder

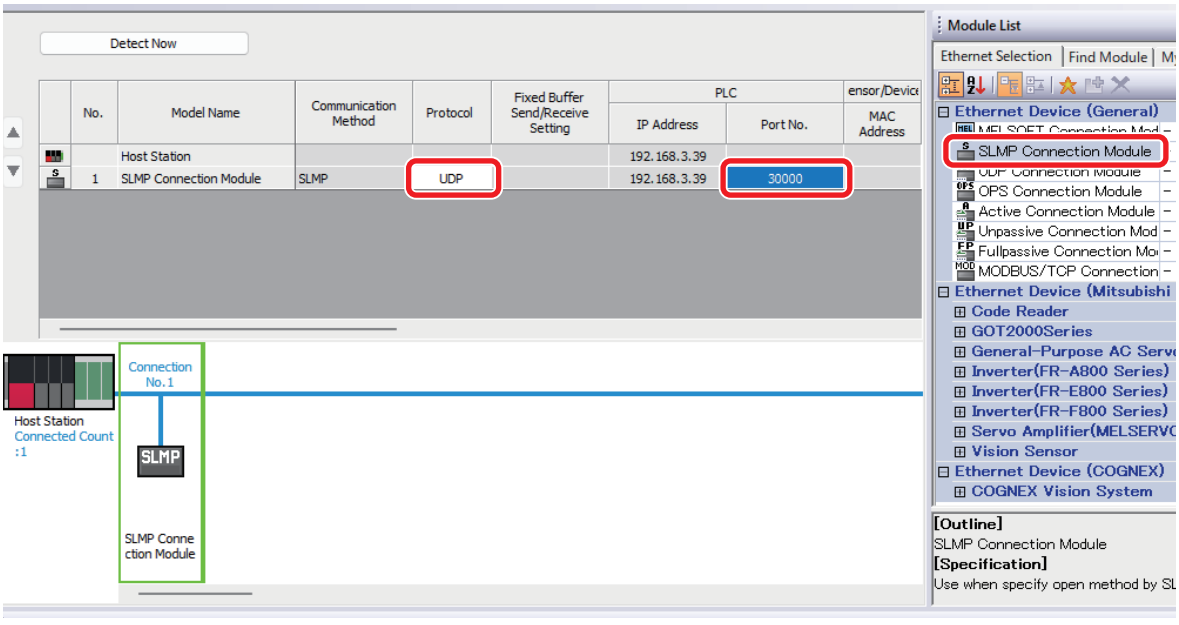
4. Select "Module Parameter" in the Navigation window. Change "External Device Online Change" to "Enable All (SLMP)".



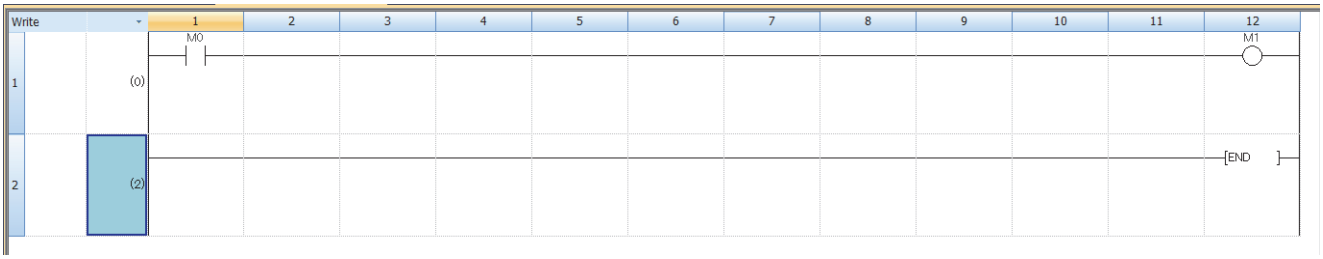
5. Double-click "Detailed Setting" of "External Device Configuration" to display the "Ethernet Configuration" window.

Add a "SLMP Connection Module", and set the following.

- Protocol: UDP
- Port No.: 30000



6. Write a contact (M0) and a coil (M1), as shown below, on the ladder program.



7. Connect to the R04CPU configured on MELSOFT Mirror, and write the parameters and program.

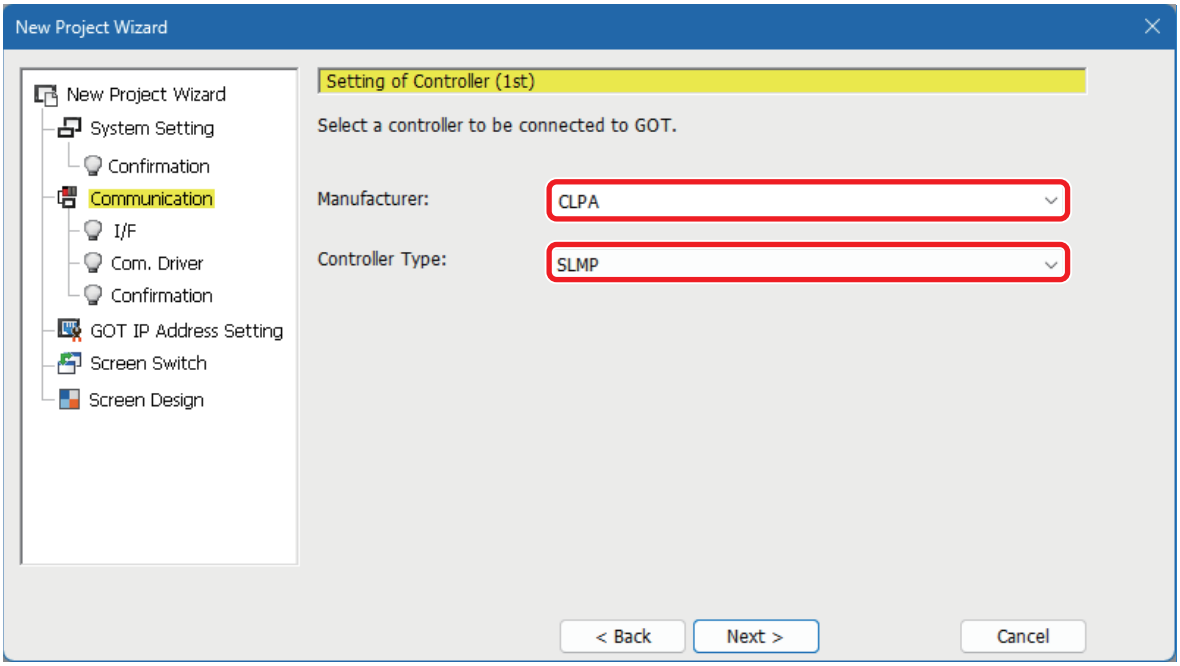
8. Reset the R04CPU, and set the switch to RUN on MELSOFT Mirror.

9. Create a new project of "GT25**-V" using GT Designer3.

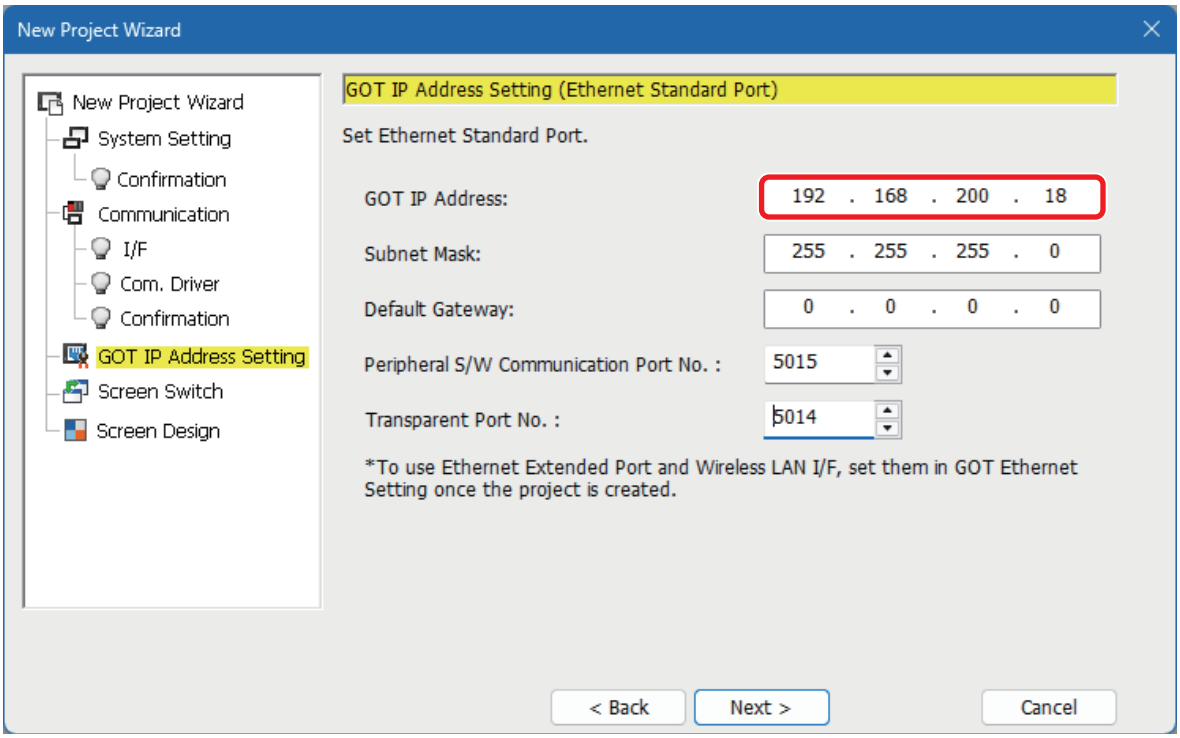
10. Click "Communication", and set the following.

- Manufacturer: CLPA
- Controller Type: SLMP

Use default settings for other items.



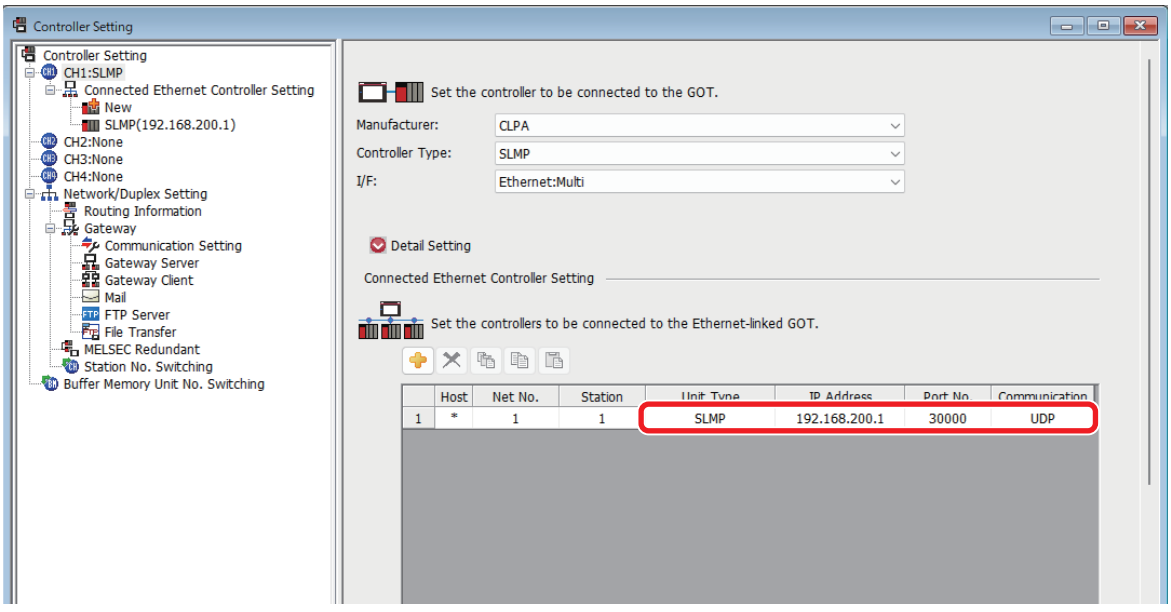
11. Select "GOT IP Address Setting", and set "192.168.200.18" to "GOT IP Address". Use default settings for other items.



12. Select [Common] ⇨ [Controller Setting] from the menu, and set the following for SLMP.

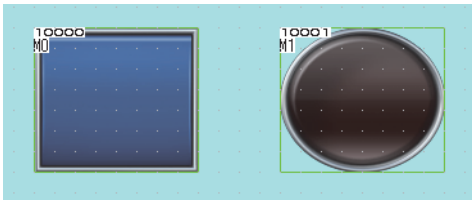
- IP Address: 192.168.200.1*1
- Port No.: 30000
- Communication method: UDP

*1 Set the same IP address set in Step 1 with MELSOFT Mirror. Set the same port number and communication method set in Step 5 with GX Works3.



13. Place a button and a lamp on the drawing screen, and set the following.

- Button: M0 (Bit alternate)
- Lamp: M1



14. Power on the GT2510-VTBA.

15. Connect GT Designer3 to the GT2510-VTBA, write the project, and reset the GT2510-VTBA.

16. SLMP communications are now performed between the GT2510-VTBA and the R04CPU configured on MELSOFT Mirror. The lamp (M1) turns on and off in response to the button (M0) operation.

11.4 Available Commands

The following table lists the commands that can be executed from an SLMP client to the CPU module.

For details on each command, refer to the following.

SLMP Reference Manual

Type	Operation	Command	Subcommand	Description
Device	Read	0401	0001	Reads value from the bit devices (consecutive device No.) in 1-point units.
			0081	
			0000	• Reads value from the bit devices (consecutive device No.) in 16-point units. • Reads value from the word devices (consecutive device No.) in one-word units.
			0080	
			0003	Reads value from the bit devices (consecutive device No.) in 1-point units.
			0083	
			0002	• Reads value from the bit devices (consecutive device No.) in 16-point units. • Reads value from the word devices (consecutive device No.) in one-word units.
			0082	
	Write	1401	0001	Writes value to the bit devices (consecutive device No.) in 1-point units.
			0081	
			0000	• Writes value to the bit devices (consecutive device No.) in 16-point units. • Writes value to the word devices (consecutive device No.) in one-word units.
			0080	
			0003	Writes value to the bit devices (consecutive device No.) in 1-point units.
			0083	
			0002	• Writes value to the bit devices (consecutive device No.) in 16-point units. • Writes value to the word devices (consecutive device No.) in one-word units.
			0082	
	Read Random	0403	0000	Specifies the device No. and reads value from the word devices in one-word units or two-word units. This can be specified with inconsecutive device No.
			0080	
			0002	Specifies the device No. and reads value from the word devices in one-word units or two-word units. This can be specified with inconsecutive device No.
			0082	
	Write Random	1402	0001	Specifies the device No. to bit device in 1-point units and writes value. This can be specified with inconsecutive device No.
			0081	
			0000	• Specifies the device No. to bit device in 16-point units and writes value. This can be specified with inconsecutive device No. • Specifies the device No. to word device in one-word units or two-word units and writes value. This can be specified with inconsecutive device No.
			0080	
			0003	Specifies the device No. to bit device in 1-point units and writes value. This can be specified with inconsecutive device No.
			0083	
			0002	• Specifies the device No. to bit device in 16-point units and writes value. This can be specified with inconsecutive device No. • Specifies the device No. to word device in one-word units or two-word units and writes value. This can be specified with inconsecutive device No.
			0082	
	Entry Monitor Device	0801	0000	Registers a device to be read by Execute Monitor (command: 0802).
			0080	
			0002	
			0082	
	Execute Monitor	0802	0000	Reads the value of device registered by Entry Monitor Device (command: 0801).

Type	Operation	Command	Subcommand	Description
Device	Read Block	0406	0000	Reads data by treating n points of word devices or bit devices (one point is equivalent to 16 bits) as one block and specifying multiple blocks. This can be specified with inconsecutive device No.
			0080	
			0002	
			0082	
	Write Block	1406	0000	Writes data by treating n points of word devices or bit devices (one point is equivalent to 16 bits) as one block and specifying multiple blocks. This can be specified with inconsecutive device No.
			0080	
			0002	
			0082	
Label	Array Label Read	041A	0000	Reads data from array type labels or labels whose structure members are the array.
	Array Label Write	141A	0000	Writes data to array type labels or labels whose structure members are the array.
	Read Random	041C	0000	Specifies labels and reads data.
	Write Random	141B	0000	Specifies labels and writes data.
Memory	Read	0613	0000	Reads the buffer memory data of own station (SLMP-compatible device).
	Write	1613	0000	Writes the data in the buffer memory of own station (SLMP-compatible device).
Remote Control	Remote Run	1001	0000	Executes the remote RUN to the access destination module.
	Remote Stop	1002	0000	Executes the remote STOP to the access destination module.
	Remote Pause	1003	0000	Executes the remote PAUSE to the access destination module.
	Remote Latch Clear	1005	0000	Executes the remote latch clear to the access destination module.
	Remote Reset	1006	0000	Executes the remote RESET to the access destination module.
	Remote Type Name	0101	0000	Reads the model name and model code of the access destination module.
File	Read Directory/File	1810	0040	Reads file list information.
	Search Directory/File	1811	0040	Reads the presence of the specified file, file No., and file size.
	New File	1820	0040	Reserves storage area for the specified file.
	Delete File	1822	0040	Deletes a file.
	Copy File	1824	0040	Copies the specified file.
	Change File State	1825	0040	Changes file attributes.
	Change File Date	1826	0040	Changes the file creation date.
	Open File	1827	0040	Locks a file so that the content of the file is not changed by other devices.
	Read File	1828	0000	Reads the data of a file.
	Write File	1829	0000	Writes the data to a file.
	Close File	182A	0000	Cancels the file lock by open processing.
Self Test		0619	0000	Tests whether the communication with external devices is normally executed or not.

11.5 Error Codes

For details on SLMP-related error codes, refer to the following.

MELSEC iQ-R CPU Module User's Manual (Application)

12 TROUBLESHOOTING

If an error occurs, an error message is displayed in the MELSOFT Mirror window. If troubleshooting is required, check the message displayed in the MELSOFT Mirror window.
This chapter describes the troubleshooting not displayed in the MELSOFT Mirror window.

12.1 Checking the Start of Service

This section describes how to check the start-up of MELSOFT Mirror.

1. Press Ctrl + Shift + Esc to open Task Manager, and select the "Services" tab. In the simple view, Task Manager does not display the "Services" tab. In this case, change to the detailed view.
2. Check whether "MELSOFT Mirror" is displayed in the "Name" column of the service list. If "MELSOFT Mirror" is displayed, MELSOFT Mirror is installed.
3. The running status of MELSOFT Mirror can be checked in the "Status" column of the "MELSOFT Mirror" row. The "Status" column shows whether the service is "Running" or "Stopped". The operating status, "Start" or "Stop", of the service can be set by right-clicking the "MELSOFT Mirror" row.

12.2 Troubleshooting by Symptom

Troubleshooting browser errors

Unable to connect to simulation server by using the browser

Check item	Action
Are the Ethernet cables connected correctly?	Reconnect the Ethernet cables.
Is the switching HUB being used operating normally?	• Check whether the switching HUB meets the standards. (☞ Page 11 System Required to Use MELSOFT Mirror) • Power off and on the switching HUB.
Does the network have any overlapping IP addresses?	Set unique IP addresses within the network.
Is an out-of-range IP address specified?	Note the following, and specify the correct IP address. • The IP address must be either a private IP address that is usable on the network to which the computer running MELSOFT Mirror is connected (10.0.0.1 to 10.255.255.254, 172.16.0.1 to 172.31.255.254, or 192.168.0.1 to 192.168.255.254), the loopback address (127.0.0.1), or any address (0.0.0.0). • There must be no spaces in the IP address.
Is the network heavily loaded?	• Reduce the communication load within the network. • Check that a large amount of unrelated data is not received (for example, broadcasting), and if applicable, take measures such as network isolation.
Are the data communications blocked by the firewall on the personal computer?	Check the state of the firewall on the personal computer. If the firewall is on, turn it off. Ensure that no security issues are present before changing the state.
Is the MELSOFT Mirror service running?	After powering on the personal computer, wait for a while, and access MELSOFT Mirror. If the service does not start even after waiting a while, refer to the following. (☞ Page 250 Checking the Start of Service)
Is a proxy server used?	• When multiple network adapters are enabled, set an IP address that can be connected to the proxy server using MELSOFT Mirror Management Tool. When the proxy server is not used, enter the IP address set in MELSOFT Mirror Management Tool in the "Proxy Settings" window on the computer. Open [Control Panel] in Windows. Select [Network and Internet] ⇨ [Internet Options] ⇨ [Connections] tab ⇨ [LAN settings] button ⇨ [Advanced] button, and enter the IP address set in MELSOFT Mirror Management Tool in the "Exceptions" box.

Troubleshooting MELSOFT Mirror errors

Clicking the [Power ON] button results in an error; MELSOFT Mirror does not start

Check item	Action
Is the operating environment of MELSOFT Mirror fulfilled?	Check the operating environment of the personal computer on which MELSOFT Mirror is installed. ☞ Page 11 MELSOFT Mirror operating environment
Is the I/O setting configured so that no errors occur?	• Review the configuration of the I/O setting. • Check that the module mounted on the slot indicated by the name of the controller block corresponds to the controller block, and that the module can be simulated. For example, when the name of the RD77MS2 block is "1.0.2", the RD77MS2 must be mounted on I/O2 slot of the main base unit (base number 0) in system 1, and the module operation setting of the RD77MS2 must be set to "Enabled".

A value cannot be changed in the Volume block pop-up window in the block diagram

Check item	Action
Are functions other than the pop-up window working?	Update the web browser connected to MELSOFT Mirror to the latest version.

When the ERR LED of the module is flashing, but the cause of the problem is unknown

Check item	Action
Is the module whose ERR LED is flashing a CPU module?	Select [Diagnostics] ⇨ [Module Diagnostics (CPU Diagnostics)] from the menu of the engineering tool, identify the error on the Module Diagnostics window, and eliminate the error cause.
Is the module whose ERR LED is flashing a network module?	• If the network module is a control station or master station, an error occurs when there is no normal station or local station that can be simulated within the same network. Enable the module operation setting of the normal station or local station connected. Or, disable the module operation setting of the network module. If the network module is a normal station or local station, an error occurs when there is no control station or master station that can be simulated within the same network. Enable the module operation setting of the control station or master station connected. Or, disable the module operation setting of the network module. If the module operation setting of some stations on the network is disabled, a disconnection error (minor error) may occur. Ignore the error, and continue the simulation. Or, enable the module operation setting of all the stations on the network. Refer to the following, and eliminate the error cause. ☞ Page 281 Simulation of network modules
Is the module whose ERR LED is flashing a Simple Motion module?	Refer to the following, and eliminate the error cause. ☞ Page 282 Troubleshooting of Simple Motion module

Troubleshooting engineering tool errors

When the engineering tool cannot connect to the simulation server

Check item	Action
Is MELSOFT Mirror powered on?	Power on MELSOFT Mirror.

Module configuration error (2000H-2021H)/CPU configuration error (2040H-2050H)

Refer to the following manual as well.

MELSEC iQ-R Programmable Controller CPU Module User's Manual

Check item	Action
Are the system parameters of each CPU module and the MELSOFT Mirror module configuration the same?	Configure the MELSOFT Mirror system again in accordance with the system parameters ("I/O Assignment Setting"). Or, set the system parameters ("I/O Assignment Setting") again in accordance with the MELSOFT Mirror system configuration.
In the MELSOFT Mirror system configuration, is the simulation of the control CPU that controls the simulation-supported RD77MS or network module enabled?	In the MELSOFT Mirror system configuration, enable the simulation of the control CPU that controls the simulation-supported RD77MS or network module. Or, disable the simulation of the control CPU that controls the simulation-unsupported RD77MS or network module.
In the MELSOFT Mirror system configuration, is the simulation of the control CPU that controls the simulation-supported I/O module enabled?	In the MELSOFT Mirror system configuration, enable the simulation of the control CPU that controls the I/O module. Or, disable the simulation of the control CPU that controls the simulation-unsupported I/O module.

"Cannot communicate with the specified CPU or station." is displayed

Refer to the following manual as well.

MELSEC iQ-R Programmable Controller CPU Module User's Manual

Check item	Action
Are the system parameters of each CPU module and the MELSOFT Mirror module configuration the same?	Configure the MELSOFT Mirror system again in accordance with the system parameters ("I/O Assignment Setting"). Or, set the system parameters ("I/O Assignment Setting") again in accordance with the MELSOFT Mirror system configuration.
In the MELSOFT Mirror system configuration, is the simulation of the control CPU that controls the simulation-supported RD77MS or network module enabled?	In the MELSOFT Mirror system configuration, enable the simulation of the control CPU that controls the simulation-supported RD77MS or network module. Or, disable the simulation of the control CPU that controls the simulation-unsupported RD77MS or network module.
In the MELSOFT Mirror system configuration, is the simulation of the control CPU that controls the simulation-supported I/O module enabled?	In the MELSOFT Mirror system configuration, enable the simulation of the control CPU that controls the I/O module. Or, disable the simulation of the control CPU that controls the simulation-unsupported I/O module.

Troubleshooting SLMP-compatible device errors

When communications with the CPU module cannot be performed

Check item	Action
Is a connection with the CPU module established when using TCP/IP?	Issue a connection request from the external device to the CPU module. (Active open)
Is a request message sent from the external device?	• Send a request message from the external device to the CPU module. • Check that the command in the request message is supported.
Is the network load high?	• Decrease the frequency of request message transmission from the external device. • Reduce the network load.
Is the IP address correct?	• Match the network part of the external device IP address with the network part of the simulation-target CPU module IP address set in the Simulation window. • Within the network where a personal computer on which MELSOFT Mirror has been installed is connected, set a unique IP address to each personal computer and Ethernet device. • Set a unique IP address to each module simulated with MELSOFT Mirror in the module parameter. • Set the IP address set to the simulation-target CPU module in the Simulation window to the destination IP address of the request message.
Is the protocol (TCP/IP or UDP/IP) correct?	Match the external device protocol with the protocol set in the module parameter of the CPU module.
Is the port number correct?	• Match the destination port number of the request message with the own station port number set in the module parameter of the CPU module. • Check that no software that uses the own station port number set in the module parameter of the CPU module is running on the personal computer on which MELSOFT Mirror has been installed. • Set 49151 or less to the own station port number in the module parameter of the CPU module. When the dynamic port range of Windows has been changed, set the port number outside the dynamic port range.
Is the communication data code (ASCII or binary code) the same between the external device and the CPU module?	Match the communication data code of the request message (ASCII or binary code) with the communication data code set in the module parameter of the CPU module.
Is the request message format correct?	Send the request message in the message format described in the SLMP Reference Manual. (SLMP Reference Manual)
Is the storing order and the value range of the specified data within the request message correct?	Set the specified data within the request message in the storing order and the value range described in the SLMP Reference Manual. (SLMP Reference Manual)
Is the "end code" of the response message 0?	If the "end code" is not 0, an error has occurred in the CPU module. Check the meaning of the end code in the manual of the CPU module used, and take a corrective action.
When using TCP/IP, is the length of the response message that the external device actually received the same as the one expected?	• If the response message is shorter than expected, take action to receive the remaining data. • If the response message is longer than expected, check the corresponding request message. • Reset the CPU module.
Is the firewall set?	Change the firewall settings to allow communications using the specified IP address, protocol, and port number.
When the label access is used, is "Access from External Device" enabled in the global label setting editor of GX Works3?	Enable "Access from External Device" in the global label setting editor of GX Works3.
Is the number of SLMP clients being connected 10 or less?	Reduce the number of SLMP clients being connected to 10 or less to MELSOFT Mirror.

When the module does not operate during SLMP communications

Check item	Action
When accessing the buffer memory areas of a module, is the access destination module set as a simulation-target module?	To set the module or obtain the module status by accessing the buffer memory areas, the module must be set as a simulation-target module. If the module is supported by MELSOFT Mirror, set it as a simulation-target module. When the module cannot be set as a simulation-target module, the default values in all buffer memory areas are 0. The functions of the module do not operate although external devices can read/write the values from/to the buffer memory areas. The values in all buffer memory areas return to 0 by powering off and on or resetting the module. Set the external devices so that they operate normally even if the read value is 0 or the last write value, or the module functions do not operate.

APPENDICES

Appendix 1 Function List

This section describes the functions of the modules supported by MELSOFT Mirror and the differences from the functions of the modules actually operated in a system.

Supported modules

For modules supported by MELSOFT Mirror, refer to the following. (☞ Page 13 Supported Modules)

Supported CPU module functions

The functions of the CPU modules supported by MELSOFT Mirror are listed below.
For details on each function, refer to the following.
📖MELSEC iQ-R CPU Module User's Manual (Application)

CPU module (RnCPU, RnPCPU)

○: Supported, △: Partly restricted, ×: Not supported, RnPCPU(P): RnPCPU process mode, RnPCPU(R): RnPCPU redundant mode

Function					Availability		
					RnCPU	RnPCPU(P)	RnPCPU(R)
Basic	Program	Program execution (ladder, ST, FBD/LD)			○	○	○
		Program execution (SFC)			○	○	○
		Scan time monitoring (WDT setting)			△*1	△*1	△*1
	Clock	Clock	Clock		△*1	△*1	△*1
			Time zone setting		○	○	○
			Daylight saving time		○	×	×
		System clock			○	○	○
	LED indication				○	○	○
	Write to PLC, Read to PLC	Writing data to the programmable controller (including online change)			○	○	○
		Reading data from the programmable controller (including online change)			○	○	△*1
	Diagnostics	Self-diagnostics	Error detection		△*1	△*1	△*1
			Module diagnostics		△*1	△*1	△*1
	Monitor and test	Monitor	Program monitor (circuit monitor)		○	○	○
		Test	Device/label data test (changing values)	Device/label data test (changing values)	○	○	○
				Local device/label data test	○	○	○
	Debug	Program/parameter online change	Program online change (ladder block)		○	○	○
		Program restoration information write selection			○	○	×
	Program file storage destination change				○	○	○
	System operation setting	LED control setting for program restoration information write status			○	○	×
		File batch online change operation setting			○	○	○

A

Function				Availability		
				RnCPU	RnPCPU(P)	RnPCPU(R)
Operation	Program	Program execution setting	Program execution type setting	○	○	○
			Device/file usage setting	○	○	○
			Refresh group setting	△*1	△*1	△*1
		Interrupt	Fixed scan interval setting	○	○	○
			File register (R) block number saving/ recovering setting	○	○	○
			Interrupt priority setting from module	○	○	○
			Interrupt enable setting during instruction execution	○	○	○
		PID control		○	○	○
		Process control		×	○	○
		Constant scan		○	○	○
	Database			△*1	×	×
	Operation	Remote operation	Operation change	○	○	○
			Latch clear	○	○	○
			RUN-PAUSE contact setting	○	○	○
	Device/data	Device/label memory area setting	Device/label memory area setting	○	○	○
			Local device setting	○	○	○
		Link direct device setting		○	○	○
		Refresh memory setting		○	×	×
		Internal buffer capacity setting		○	○	○
		Index register setting		○	○	○
		Device/label initial value setting	Device/label initial value setting	○	○	○
		Label initial value reflection setting		○	○	○
		Label initialization when all data is rebuilt and written to the programmable controller		×	○	○
		Device comment		○	○	○
		File register setting		○	○	○
		Timer device limit setting		○	○	○
		STOP to RUN operation setting of output (Y)		○	○	○
		Label access setting from external devices		○	○	○
		System clock (SM400 to SM403) every end update setting		○	×	×
	Latch	Latch		○	○	○
	Memory card	Reading/writing data from/to an SD memory card		○	○	○
		SD memory card forcibly disabled		○	○	○
	Communication setting	Device/label access service processing setting	Device/label access service processing setting	○	○	○
			Device/label access service processing constant wait	×	○	○
RAS	Diagnostics	Self-diagnostics	Error clear	○	○	○
			FB hierarchy information display	○	○	○
	System monitoring			△*1	×	×
	History	Event history function (error history/module error history collection)		△*1	△*1	△*1

Function					Availability		
					RnCPU	RnPCPU(P)	RnPCPU(R)
Maintenance	Monitor	(Interrupt) Program list monitor	(Interrupt) Program list monitor		△*1	△*1	△*1
			Specified program start/stop		○	○	○
		Scan time monitor			△*1	△*1	△*1
		Execution time measurement (Scan time measurement)			×	×	×
		SFC program (diagram) monitor			○	○	○
		(Local) Device/label batch/registration monitor			○	○	○
		Buffer memory monitor			○	○	○
		Real-time monitor			×	×	×
		Specified program monitor/current value change			○	×	×
	Test	Program test	Device test with execution conditions		○	○	○
		I/O control	External input/output forced on/off		○	○	○
		Device/label/file register clear			○	○	○
		Latch device/label clear			○	○	○
	Logging	Logging			×	×	×
		Data storage destination memory	SD memory card		×	×	×
			CPU built-in memory (function memory)		×	×	×
		Logging target	Global device		×	×	×
			Local device		×	×	×
			Global label		×	×	×
			Local label		×	×	×
		File format	Unicode text file		×	×	×
			Binary file		×	×	×
			CSV file		×	×	×
	Debug	Program/parameter online change	Timer/counter setting value change		○	○	○
			SFC program online change	Inactive block	○	○	○
				Active block	○	○	○
		Memory dump			○	×	×
Multiple CPU system	Operation settings	Other PLC control module setting	Control CPU setting		○	○	×
			Out-of-group I/O fetch		△*1	△*1	×
		Operation mode setting	Clock data synchronization		△*1	△*1	×
			Operation setting at stop error		△*1	△*1	×
			Multiple CPU synchronous startup setting		△*1	△*1	×
	Data communications between CPU modules	Communications	Fixed scan communication interval setting		△*1	△*1	×
			Fixed scan communication send data setting		△*1	△*1	×
			Module-by-module data guarantee		△*1	△*1	×
			Communication through refresh		△*1	△*1	×
			Communication through direct access		△*1	△*1	×
	Interrupt	Multiple CPU synchronous interrupt (I45)			△*1	△*1	×
System configuration	System configuration settings	I/O number/points variable setting			○	○	○
		MELSEC iQ-R series 2-slot modules			○	○	○

A

Function				Availability		
				RnCPU	RnPCPU(P)	RnPCPU(R)
Programming common	Operation settings	MELSEC iQ-R series module setting (all modules)	Initial value setting	△*1	△*1	△*1
			Network parameter setting (Ethernet)	△*1	△*1	△*1
			Network parameter setting (CC-Link IE Control)	△*1	△*1	△*1
			Network parameter setting (CC-Link IE Field)	△*1	△*1	△*1
			Network parameter setting (CC-Link)	×	×	×
			Network parameter setting (MELSECNET/H)	△*1	△*1	△*1
			Operation setting at H/W error	○	○	○
			Input response time setting	×	×	×
	Data communications	Refresh functions (I/O refresh, network refresh, buffer memory (intelligent function module) refresh)		○	○	○
		Direct access functions (U□\G, J□\X, J□\Y, J□\B, J□\SB, J□\W, J□SW)		○	○	○
	Program	Interrupt from module		△*1	△*1	△*1
		Dedicated instructions (MELSEC Q and MELSEC iQ-R series intelligent function modules)		△*1	△*1	△*1
	Program creation support	Module label/module FB (device comment) auto creation		○	○	○
		Sample program display (template display)		○	○	○
		Parameter auto-creation (initial setting/refresh setting)		○	○	○
General-purpose communications	SLMP communications			○	○	○
	MC protocol communication	QnA-compatible 2/3/4(C/E) frame		×	×	×
Label communications	Non-fixed scan communications	Label communications from GOT		×	×	×
		Label communications for SLMP and middleware for partners		×	×	×
Inter-module synchronization	Inter-module synchronization	Fixed scan synchronous control		△*1	△*1	×
	Interrupt	Inter-module synchronous interrupt (I44)		△*1	△*1	×
Security	File read/write restriction	File password		○	○	○
	Access control/program execution control by security key authentication			×	×	×
	Access control by communication route	Remote password		×	×	×
RAS	History	Event history	Event history	△*1	△*1	△*1
			Error history/system error history	△*1	△*1	△*1
			Network event history	△*1	△*1	△*1
	Diagnostics	Module diagnostics		△*1	△*1	△*1
		Network diagnostics	CC-Link IE Control diagnostics	△*1	△*1	△*1
			CC-Link IE Field diagnostics	△*1	△*1	△*1
			MELSECNET diagnostics	△*1	△*1	△*1

Function				Availability		
				RnCPU	RnPCPU(P)	RnPCPU(R)
Ethernet	Communications with devices (maintenance)	Connection with MELSOFT products and a GOT	Connection with MELSOFT products and a GOT	×	×	×
			Network number and station number specification	×	×	×
	Communications with devices	Communications using SLMP (MC protocol)		○	○	○
		Socket communications	Socket communications	×	×	×
			Connection close wait time setting	×	×	×
	Communications with devices (upper)	Communications using SLMP client instructions		×	×	×
		Communications using the web function (web server)	Detailed module information display	○	×	×
			CPU diagnostics	○	×	×
			Event history	△*1	×	×
			Device batch monitor	○	×	×
			Watch	○	×	×
			Device test	○	×	×
			Language switcher	○	×	×
			User Web page	○	×	×
			Access log	○	×	×
	Communications with devices (field)	Predefined protocol support		×	×	×
		SP.ECPRTCL instruction completion condition setting		×	×	×
	Communications with devices (communications between programmable controllers)	Simple CPU communication		△*1	×	×
	Others	Discard received data at CPU STOP setting		×	×	×
Network	Routing			△*1	△*1	△*1
	Interlink transmission (between MELSEC iQ-R series modules)			△*1	△*1	△*1
Redundant function	System determination	System A/B determination		×	×	△*1
	Operation mode change	Separate mode		×	×	△*1
	Redundant extension base unit control	Redundant extension base unit diagnostic function		×	×	△*1

*1 For details, refer to the following. (☞ Page 260 Restrictions)

■Restrictions

×: Not supported

Function				Restriction		
				RnCPU	RnPCPU(P)	RnPCPU(R)
Basic	Program	Scan time monitoring (WDT setting)		Instruction processing speeds differ between MELSOFT Mirror and the actual CPU module. (☞ Page 280 Simulation time)		
	Clock			The clock data is written to the internal time of the simulator. It differs from the time on the personal computer.		
	Diagnostics	Self-diagnostics	Error detection	☞ Page 282 Self-diagnostic function errors		
			Module diagnostics	Only the host CPU module can use the function.		
Operation	Program	Program execution setting	Refresh group setting	Only modules that can be simulated with MELSOFT Mirror can perform data refresh.		
	Database			☞ Page 283 Database	×	
RAS	System monitoring			MAC address and firmware hash value cannot be used.	×	
	History	Event history function (error history/ module error history collection)		Only modules that can be simulated with MELSOFT Mirror can use this function.		
Maintenance	Monitor	(Interrupt) Program list monitor		☞ Page 280 Simulation time		
		Scan time monitor				
Multiple CPU system	Operation settings	Other PLC control module setting	Out-of-group I/O fetch	Only modules that can be simulated with MELSOFT Mirror can use these functions.		×
		Operation mode setting	Clock data synchronization			
			Operation setting at stop error			
			Multiple CPU synchronous startup setting			
	Data communications between CPU modules	Communications	Fixed scan communication interval setting			
			Fixed scan communication send data setting			
			Module-by-module data guarantee			
			Communication through refresh			
			Communication through direct access			
	Interrupt	Multiple CPU synchronous interrupt (I45)				

Function				Restriction		
				RnCPU	RnPCPU(P)	RnPCPU(R)
Programming common	Operation settings	MELSEC iQ-R series module setting (all modules)	Initial value setting	Only modules that can be simulated with MELSOFT Mirror can use these functions.		
			Network parameter setting (Ethernet)			
			Network parameter setting (CC-Link IE Control)			
			Network parameter setting (CC-Link IE Field)			
			Network parameter setting (MELSECNET/H)			
	Program	Interrupt from module		Only modules that can be simulated with MELSOFT Mirror can use these functions.		
		Dedicated instructions (MELSEC Q and MELSEC iQ-R series intelligent function modules)				
Inter-module synchronization	Inter-module synchronization	Fixed scan synchronous control		Only modules that can be simulated with MELSOFT Mirror can use these functions.		×
	Interrupt	Inter-module synchronous interrupt (I44)				
RAS	History	Event history	Event history	Only modules that can be simulated with MELSOFT Mirror can use these functions.		
			Error history/ system error history			
			Network event history			
	Diagnostics	Module diagnostics		Only the host CPU module can use the function.		
		Network diagnostics	CC-Link IE Control diagnostics	Only modules that can be simulated with MELSOFT Mirror can use these functions. Operation tests cannot be used.		
			CC-Link IE Field diagnostics			
			MELSECNET diagnostics			
Ethernet	Communications with devices (upper)	Communications using the web function (web server)	Event history	Only modules that can be simulated with MELSOFT Mirror can use these functions.	×	
	Communications with devices (communication between programmable controllers)	Simple CPU communication				
Network	Routing			Only modules that can be simulated with MELSOFT Mirror can use these functions.		
	Interlink transmission (between MELSEC iQ-R series modules)					
Redundant function	System determination			×		The CPU module on system A can use the function.
	Operation mode change					The CPU module operates only in separate mode.
	Redundant extension base unit control					Redundant extension base units are supported, but system switching is not supported.

A

Supported devices

MELSOFT Mirror supports devices of the CPU module.

In addition, MELSOFT Mirror supports the special relay and special register used in supported functions.

For details on devices, refer to the following.

MELSEC iQ-R CPU Module User's Manual (Application)

Supported instructions

The following tables list the instructions supported by MELSOFT Mirror.

For details on each instruction, refer to the following.

MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks)

MELSEC iQ-R Programming Manual (Process Control Function Blocks/Instructions)

Note that no processing is performed when unsupported instructions are executed.

RCPU

■Sequence instructions

Classification	Instruction symbol
Contact instructions	AND, ANDF, ANDFI, ANDP, ANDPI, ANI, LD, LDF, LDFI, LDI, LDP, LDPI, OR, ORF, ORFI, ORI, ORP, ORPI
Association instructions	ANB, EGF, EGP, INV, MEF, MEP, MPP, MPS, MRD, ORB
Output instructions	DELTA(P)*1, FF, OUT, OUT C, OUT F, OUT LC, OUT LT/LST, OUT T/ST, OUTH T/ST, PLF, PLS, RST, RST F, SET, SET F
Shift instructions	SFT(P)
Master control instructions	MC, MCR
Phase processing instructions	PHASE*2, PHASECHG*2, PHASEEND*2, PHASERST*2
Termination instructions	END, FEND
Stop instruction	STOP
No operation instructions	NOP, NOPLF

*1 The instruction is used for the intelligent function modules supported by MELSOFT Mirror.

*2 The RnCPU supports these instructions.

■Basic instructions

Classification	Instruction symbol*1
Comparison operation instructions	AND□(_U), ANDD□(_U), BKCMP□(P)(_U), CMP(P)(_U)*2, DBKCMP□(P)(_U), DCMP(P)(_U)*2, DZCP(P)(_U)*2, LD□(_U), LDD□(_U), OR□(_U), ORD□(_U), ZCP(P)(_U)*2
Arithmetic operation instructions	*(P)(_U), +(P)(_U), -(P)(_U), /(P)(_U), B*(P), B+(P), B-(P), B/(P), BK+(P)(_U), BK-(P)(_U), D*(P)(_U), D+(P)(_U), D-(P)(_U), D/(P)(_U), DB*(P), DB+(P), DB-(P), DB/(P), DBK+(P)(_U), DBK-(P)(_U), DDEC(P)(_U), DEC(P)(_U), DINC(P)(_U), INC(P)(_U)
Logical operation instructions	BKAND(P), BKOR(P), BKXNR(P), BKXOR(P), DAND(P), DOR(P), DXNR(P), DXOR(P), WAND(P), WOR(P), WXNR(P), WXOR(P)
Bit processing instructions	BKRST(P), BRST(P), BSET(P), DTEST(P), TEST(P), ZRST(P)*2
Data conversion instructions	ASC2INT(P), BCD(P), BIN(P), BKBCD(P), BKBIN(P), BTOW(P), DABCD(P), DABIN(P)(_U), DBCD(P), DBIN(P), DBL2DINT(P), DBL2INT(P), DBL2UDINT(P), DBL2UINT(P), DDABCD(P), DDABIN(P)(_U), DECO(P), DGBIN(P)(_U), DGRY(P)(_U), DHABIN(P), DINT2INT(P), DINT2UDINT(P), DINT2UINT(P), DIS(P), DNEG(P), DVAL(P)(_U), EMOD(P), ENCO(P), FLT2DINT(P), FLT2INT(P), FLT2UDINT(P), FLT2UINT(P), GBIN(P)(_U), GRY(P)(_U), HABIN(P), INT2DINT(P), INT2UDINT(P), INT2UINT(P), NDIS(P), NEG(P), NUNI(P), SEG(P), UDINT2DINT(P), UDINT2INT(P), UDINT2UINT(P), UINT2DINT(P), UINT2INT(P), UINT2UDINT(P), UNI(P), VAL(P)(_U), WTOB(P)
Shift instructions	BSFL(P), BSFR(P), EDSFTL(P)*2, EDSFTR(P)*2, ESFTL(P)*2, ESFTR(P)*2, DDSFL(P)*2, DDSFR(P)*2, DSFL(P), DSFR(P), DWSFTL(P)*2, DWSFTR(P)*2, EDSFL(P)*2, EDSFR(P)*2, ESFL(P)*2, ESFR(P)*2, ESFTL(P)*2, ESFTR(P)*2, SFL(P), SFR(P), SFTEL(P)*2, SFTER(P)*2, SFTDWL(P)*2, SFTBL(P), SFTBR(P), SFTEDL(P)*2, SFTEDR(P)*2, SFTEL(P)*2, SFTER(P)*2, SFTDWL(P)*2, SFTDWR(P)*2, SFTEDL(P)*2, SFTEDR(P)*2, SFTL(P)*2, SFTR(P)*2, SFTWL(P), SFTWR(P), WSFL(P)*2, WSFR(P)*2
Data transfer instructions	BLKMOVB(P), BMOV(P), BMOVL(P), BXCH(P), CML(P), CMLB(P), DCML(P), DFMOV(P), DFMOVL(P), DMOV(P), DSWAP(P), DXCH(P), FMOV(P), FMOVL(P), MOV(P), MOVB(P), SMOV(P)*2, SWAP(P), XCH(P)

*1 □ indicates an operator.

*2 The RnCPU supports these instructions.

■Application instructions

Classification	Instruction symbol[*1]
Rotation instructions	DRCL(P), DRCR(P), DROL(P), DROR(P), RCL(P), RCR(P), ROL(P), ROR(P)
Program branch instructions	CJ, GOEND, JMP, SCJ
Program execution control instructions	DI, EI, IMASK, IRET, SIMASK, WDT(P)
Structure creation instructions	BREAK(P), CALL(P), ECALL(P), EFCALL(P), FCALL(P), FOR, NEXT, RET, XCALL
Data table operation instructions	FDEL(P), FIFR(P), FIFW(P), FINS(P), FPOP(P)
Reading/writing data instructions	S(P).DEVLD, SP.DEVST
File operation instructions	SP.FDELETE[*2], SP.FCOPY[*2], SP.FMOVE[*2], SP.FREAD, SP.FRENAME[*2], SP.FSTATUS[*2], SP.FWRITE
Debugging and failure diagnostic instructions	LEDR, PALERT(P)[*3], PABORT[*3]
String processing instructions	$+(P), $MOV(P), $MOV(P)_WS, AND$□, BCDDA(P), BINDA(P)(_U), BINHA(P), DBCDDA(P), DBINDA(P)(_U), DBINHA(P), DSTR(P)(_U), ESTR(P), INSTR(P), INT2ASC(P), LD$□, LEFT(P), LEN(P), MIDR(P), MIDW(P), OR$□, RIGHT(P), SJIS2WS(P), SJIS2WSB(P), STR(P)(_U), STRDEL(P), STRINS(P), WS2SJIS(P)
Floating-point instructions	ACOS(P), ACOSD(P), ANDE□, ANDED□, ASIN(P), ASIND(P), ATAN(P), ATAND(P), BACOS(P), BASIN(P), BATAN(P), BCOS(P), BDSQRT(P), BSIN(P), BSQRT(P), BTAN(P), COS(P), COSD(P), DBL2FLT(P), DEG(P), DEGD(P), DINT2DBL(P), DINT2FLT(P), E*(P), E+(P), E-(P), E/(P), ECMP(P)[*2], ED*(P), ED+(P), ED-(P), ED/(P), EDCMP(P)[*2], EDMAX(P), EDMIN(P), EDMOV(P), EDNEG(P), EDSQRT(P), EDZCP(P)[*2], EMAX(P), EMIN(P), EMOV(P), ENEG(P), EREXP(P), ESQRT(P), EVAL(P), EXP(P), EXPD(P), EZCP(P)[*2], FLT2DBL(P), INT2DBL(P), INT2FLT(P), LDE□, LDED□, LOG(P), LOG10(P), LOG10D(P), LOGD(P), ORE□, ORED□, POW(P), POWD(P), RAD(P), RADD(P), SIN(P), SIND(P), TAN(P), TAND(P), UDINT2DBL(P), UDINT2FLT(P), UINT2DBL(P), UINT2FLT(P)
Random number instructions	RND(P), SRND(P)
Index register instructions	ZPOP(P), ZPUSH(P)
Data control instructions	BAND(P)(_U), DBAND(P)(_U), DLIMIT(P)(_U), DSCL(P)(_U), DSCL2(P)(_U), DZONE(P)(_U), LIMIT(P)(_U), SCL(P)(_U), SCL2(P)(_U), ZONE(P)(_U)
Special counter instructions	UDCNT1, UDCNT2
Special timer instructions	STMR, TTMR
Shortcut control instruction	ROTC
Ramp signal instruction	RAMPQ
Pulse related instructions	PLSY, PWM, SPD
Matrix input instruction	MTR
Check code instructions	CCD(P)[*2]
Data processing instructions	BON(P)[*2], CRC(P)[*2], DBON(P)[*2], DMAX(P)(_U), DMEAN(P)(_U), DMIN(P)(_U), DSERDATA(P), DSERMM(P)[*2], DSORTD(_U), DSORTTBL(_U)[*2], DSORTTBL2(_U)[*2], DSQRT(P)[*2], DSUM(P), DWSUM(P)(_U), MAX(P)(_U), MEAN(P)(_U), MIN(P)(_U), SERDATA(P), SERMM(P)[*2], SORTD(_U), SORTTBL(_U)[*2], SORTTBL2(_U)[*2], SQRT(P)[*2], SUM(P), WSUM(P)(_U)
Database access instructions	DBOPEN(P), DBCLOSE(P), DBINSERT(P)[*4], DBUPDATE(P)[*4], DBSELECT(P)[*2], DBDELETE(P)[*4], DBIMPORT(P)[*4], DBEXPORT(P), DBTRANS(P), DBCOMMIT(P), DBROLBAK(P)
File register operation instructions	QDRSET(P), RSET(P)
File register read/write instructions	ZRRDB(P), ZRWRB(P)
Indirect address read instructions	ADRSET(P)
Clock instructions	ANDDT□, ANDTM□, DATE+(P), DATE-(P), DATERD(P), DATEWR(P), DATE2SEC(P)(_U), LDDT□, LDTM□, ORDT□, ORTM□, SEC2DATE(P)(_U), S(P).DATE+, S(P).DATE-, S(P).DATERD, SEC2TIME(P), TCMP(P)[*2], TIME2SEC(P), TZCP(P)[*2]
Timing check instructions	DHOURM[*2], DUTY, HOURM[*2], TIMCHK
Module access instructions	COM(P), DFROM(P), DFROMD(P), DTO(P), DTOD(P), FROM(P), FROMD(P), TO(P), TOD(P), RFS(P), S(P).ZCOM, UNIINFRD(P)
Routing information instructions	S(P).RTREAD, S(P).RTWRITE
Data collection instruction	DATATRG
Built-in Ethernet function instructions	SP.ECPRTCL, SP.FTPGET, SP.FTPPUT, SP.SOCOPEN, SP.SOCCLOSE, SP.SOCRCV, S.SOCRCVS, SP.SOCSND, SP.SOCCINF, SP.SOCCSET, SP.SOCRMODE, S.SOCRDATA, SP.SOCRDATA
Program control instructions	POFF(P), PSCAN(P), PSTOP(P)
PID control instructions	PIDCONT(P), PIDINIT(P), PIDPRMW(P), PIDRUN(P), PIDSTOP(P), S(P).PIDCONT, S(P).PIDINIT, S(P).PIDPRMW, S(P).PIDRUN, S(P).PIDSTOP
SFC control instructions	AND, ANI, BMOV(P), BRSET, DMOV(P), LD, LDI, MOV(P), OR, ORI, PAUSE, RST, RSTART, SET
SFC dedicated instruction	TRAN

Classification	Instruction symbol*1
Process control instructions*3	S.2PID, S.□, S.ABS, S.ADD, S.AMR, S.AT1, S.AVE, S.BC, S.BPI, S.BUMP, S.D, S.DBND, S.DED, S.DIV, S.DUTY, S.ENG, S.FG, S.FLT, S.HS, S.I, S.IENG, S.IFG, S.IN, S.IPD, S.LIMIT, S.LLAG, S.LS, S.MID, S.MOUT, S.MUL, S.ONF2, S.ONF3, S.OUT1, S.OUT2, S.PGS, S.PHPL, S.PHPL2, S.PID, S.PIDP, S.PSUM, S.R, S.SEL, S.SPI, S.SQR, S.SUB, S.SUM, S.TPC, S.VLMT1, S.VLMT2
Multiple CPU dedicated instructions (built-in programmable controller CPU)	D(P).DDRD, D(P).DDWR, M(P).DDRD, M(P).DDWR

*1 □ indicates an operator.
*2 The RnPCPU does not support these instructions.
*3 The RnCPU does not support these instructions.
*4 Field names and indexes are not case sensitive.

■Module dedicated instructions

For the following modules, module dedicated instructions can be used.

For details on module dedicated instructions, refer to the manual for the module used.

Module type		Model
Network module	CC-Link IE Field Network module	RJ71GF11-T2, RJ71GF11-T2(MR), RJ71GF11-T2(SR), RJ71GF11-T2(LR)
	CC-Link IE Controller Network module	RJ71GP21-SX, RJ71GP21-SX(R), RJ71GP21S-SX, RJ71GP21S-SX(R)

Supported parameters

The following tables list the parameter setting items supported by MELSOFT Mirror.

RCPU

■System parameters

Item		
I/O Assignment	Base/Power/Extension Cable Setting	Base, Power Supply Module, Extension Cable
		Slots
	I/O Assignment Setting	Module, Points, Start XY
		Module Status Setting
		Module Name
		Control PLC Settings
	Setting of Points Occupied by Empty Slot	
Multiple CPU Setting*1	Setting for number of CPU modules	
	Communication Setting between CPUs	Refresh Setting
		CPU Buffer Memory Setting
		PLC Unit Data
		Fixed Scan Communication Function
		Fixed Scan Communication Area Setting
	Fixed Scan Communication Setting	Fixed Scan Interval Setting of Fixed Scan Communication
		Fixed Scan Communication Function and Inter-module Synchronization Function
	Operation Mode Setting	Operation setting at stop error
		Synchronous Startup Setting
	Other PLC Control Module Setting	I/O Setting Outside Group
Inter-module Synchronization Setting*1	Use Inter-module Synchronization Function in System	
	Select Inter-module Synchronization Target Module	
	Fixed Scan Interval Setting of Inter-module Synchronization	
	Inter-module Synchronous Master Setting	

*1 The RnPCPU does not support these parameters.

■CPU parameters

Item		
Interrupt Settings	Fixed Scan Interval Setting	
	Fixed Scan Execution Mode Setting	
	Interrupt Enable Setting in Executing Instruction	
	Block No. Save/Recovery Setting	
	Interrupt Priority Setting from Module	
Operation Related Setting	Timer Limit Setting	
	RUN-PAUSE Contact Setting	
	Remote Reset Setting	
	Output Mode Setting at STOP to RUN	
	Module Synchronous Setting	
	Clock Related Setting	
Service Processing Setting	Device/Label Access Service Processing Setting	
File Setting	File Register Setting	
	Initial Value Setting*1	
	Label Initial Value Reflection Setting*2	Label Initial Value Reflection Setting at STOP to RUN*2
	File Setting for Device Data Storage	

Item		
Memory/Device Setting	Cassette Setting	Extended SRAM Cassette Setting
	Device/Label Memory Area Setting	Device/Label Memory Area Capacity Setting
		Device Points
		Local Device
		Latch Range Setting
		Latch Type Setting of Latch Type Label
	Index Register Setting	
	Refresh Memory Setting	
	Device Latch Interval Setting	
	Pointer Setting	
	Internal Buffer Capacity Setting	
	Link Direct Device Setting	
RAS Setting	Scan Time Monitoring Time (WDT) Setting	
	Constant Scan Setting	
	Error Detections setting	
	CPU Module Operation Setting at Error Detection	
	LED Display Setting	
	Event History Setting	Save Destination, Storage Capacity Setting per File*3
		Save Device/Label Operations
Program Setting	Program Setting	Program Name
		Execution Type
		Detailed Setting Information (Fixed Scan)
		Detailed Setting Information (Event)
		Refresh Group
		Device/File Use or not
	FB/FUN File Setting	
Refresh Setting between Multiple CPUs	Refresh Setting (At the END)	
	Refresh Setting (At I45 Exe.)	
I/O Assignment Setting	CPU Module Operation Setting at Error Detection	
Routing Setting	Routing Setting	
SFC Setting	SFC Program Start Mode Setting	
	Start Condition Setting	
	Output Mode at Block Stop	

*1 "Memory Card" can be set to "Target Memory".
*2 The RnPCPU supports these parameters.
*3 "Memory Card" can be set to "Save Destination".

■Memory card parameters

Item
Setting of File/Data Use or Not in Memory Card[*1]

*1 This parameter cannot be set to a device station.

■Module parameters

Item		
CPU built-in function setting	Simple CPU Communication Setting	Simple CPU Communication Setting[*1]
	Communications by Network No./Station No.	Communications by Network No./Station No.

*1 Only "MELSEC iQ-R (built-in Ethernet)" can be selected for "Communication Destination".

■Module parameters (network)

Item	
Dedicated parameters specific to Ethernet function	Ethernet operation setting[*1]
	Connection setting
	Connection comment setting
	Station number<->IP related information setting
	Web server setting
	Data communication timer setting
Function common parameter	IP address setting[*2]

*1 Only the online change setting, communication data code, and open method are supported.

*2 This cannot be used for communication with external devices. It can only be used as a destination for simple CPU communication.

A

Module buffer memory

The number of buffer memory (U□\G) points of a module supported by MELSOFT Mirror is the same as the module actually configured.

I/O module

The functions and parameters of the I/O modules supported by MELSOFT Mirror are listed below.

Functions

Function	Remarks
Interrupt input function	—
In-error output mode setting	—
Output ON number count function	—
Inter-module synchronization function	—

Parameters

Parameter		Remarks
Interrupt setting	input/interrupt setting	—
	interrupt condition setting	—
	interrupt pointer	—
Setting of error-time output mode		—

Network

The following tables list the network types, network modules, network functions, network parameters, and dedicated instructions for network modules supported by MELSOFT Mirror.
For details on each item, refer to the manual for the network module used.

Network type

Series	Network types
MELSEC iQ-R	CC-Link IE Controller Network
	CC-Link IE Field Network
	MELSECNET/H

Network module

Series	Network modules
MELSEC iQ-R	RJ71EN71
	RJ71GP21-SX
	RJ71GP21S-SX
	RJ71GF11-T2
	RJ71LP21-25

Network functions

■CC-Link IE Controller Network

Function		Remarks
Cyclic transmission	Communications with other stations	—
	Link refresh	
	Direct access to link devices	
	Cyclic data integrity assurance	
	Interlink transmission	
	Cyclic transmission punctuality assurance	
	Group cyclic transmission	
	Cyclic transmission stop and restart	
Transient transmission	Dedicated instructions	—
	Group transient transmission	
RAS	External power supply	Always off
Others	Interrupt request to CPU module	—

■CC-Link IE Field Network

Function			Remarks
Cyclic transmission	Communications with other stations		—
	Link refresh		
	Direct access to link devices		
	Cyclic data integrity assurance		
	Interlink transmission		
	Sequence scan synchronization specification		
	Cyclic transmission punctuality assurance		
	I/O maintenance settings	Output hold/clear setting during CPU STOP	
		Data link faulty station setting	
	Cyclic transmission stop and restart		
Transient transmission	Communications within the same network		—
RAS	Station type and station number duplication detection		Stations added later are detected as duplicates.
Others	CC-Link IE Field Network synchronous communications		—
	Reserved station specification		—
	Temporary cancel of the reserved station		
	Error invalid station, temporary error invalid station setting		
	Interrupt request to CPU module		

■MELSECNET/H

Function		Remarks
Cyclic transmission	Cyclic transmission	—
	Low-speed cyclic transmission	
	Link refresh	
	Direct access to link devices	
	Constant link scan	
	Cyclic data integrity assurance	
	Interlink transmission	
	Cyclic transmission stop and restart	
Transient transmission	Dedicated instructions	
	Group transient transmission	
RAS	Checking network status	
Others	Interrupt program start	
	Number of send points extension	

Network parameters

■CC-Link IE Controller Network

Parameter				Remarks
Required Settings	Station Type	Station Type		—
	Network Number	Network Number		
	Station Number	Setting Method		
		Station No.		
	Network Range Assignment	Total No. Stations		
		LB/LW Setting (1)		
		LB/LW Setting (2)		
		LX/LY Setting (1)		
		LX/LY Setting (2)		
		I/O Master Station		
		Reserved Station		
		Pairing		The system switching function of the redundant system is not supported. Only the control system performs cyclic transmission.
		Shared Group		—
Basic Settings	Refresh Settings			
Application Settings	Supplementary Cyclic Settings	Data Link Monitoring Time		
		Constant Link Scan Time		
		Transient Setting	Punctuality Assurance	
			Maximum No. of Transients in One Station	
	Link points extended setting			
	Interrupt Settings			
	Transient Transmission Group No.			
	Parameter Name			
	Event Reception from Other Stations			
	Module Operation Mode			
	Interlink Transmission Settings			

■CC-Link IE Field Network

Parameter				Remarks
Required Settings	Station Type	Station Type		The submaster station always operates as a local station.
	Network No.	Network No.		—
	Station No.	Station No.		
		Setting Method		
	Parameter Setting Method	Setting Method of Basic/Application Settings		
Basic Settings	Network Configuration Settings	STA# (Station No.)		
		Model Name		
		Station Type		
		RX/RY Setting		
		RWr/RWw Setting		
		Reserved/Error Invalid Station/System Switching Monitoring Target Station		• When "Reserved Station" or "Error Invalid Station" is set, all stations operate as error invalid stations. • "System Switching Monitoring Target Station" is not supported.
		Pairing		The system switching function of the redundant system is not supported. Only the control system performs cyclic transmission.
		Network Synchronous Communication		—
		Alias		—
		Comment		
	Refresh Settings	Refresh Settings (SB/SW)		
Application Settings	Supplementary Cyclic Settings	Link Scan Mode		
		I/O Maintenance Settings	Output Hold/Clear Setting during CPU STOP	
			Data Link Error Station Setting	
			Output Mode upon CPU Error	
	Interrupt Settings			
	Communication Mode			
	Parameter Name			
	Event Reception from Other Stations			
	Module Operation Mode			
	Interlink Transmission Settings			
	Safety Communication Setting	To Use or Not to Use the Safety Communication Setting		
		Safety Communication Setting		

A

■MELSECNET/H

Parameter			Remarks
Required Settings	Station Type	Station Type	—
	Network No.	Network No.	
	Station No.	Station No.	
	Network Range Assignment	Total No. Stations	
		LB/LW Setting	
		LX/LY Setting (1)	
		LX/LY Setting (2)	
		I/O Master Station	
		Reserved Station	
		Pairing	The system switching function of the redundant system is not supported. Transmission to the control station is performed only with lower station numbers.
	Transmission Speed	Transmission Speed	—
Basic Settings	Link Refresh Settings	SB	
		SW	
		LB/LW/LX/LY setting (64 ranges)	
Application Settings	Supplementary Cyclic Settings	Data Link Monitoring Time	
		Constant Link Scan Time	
		Maximum No. of Transient Transmissions	
		Maximum No. of Transient in One Station	
	Low-Speed Cyclic Setting	Low-Speed Cyclic Setting	
		Fixed Interval Cycle Setting	
		System Timer Setting 1 to 8	
	Interrupt Settings		
	Transient Transmission Group No.		
	Parameter Name		
	Event Reception from Other Stations		
	Module Operation Mode		
Others	Interlink Transmission Settings		

Dedicated instructions for network modules

■CC-Link IE Controller Network

Classification	Instruction symbol
Link dedicated instruction	READ, SREAD, WRITE, SWRITE, SEND, RECV, RECVS, REQ, RRUN, RSTOP, RTMRD, RTMWR
Station number setting	UINI

■CC-Link IE Field Network

Classification	Instruction symbol
Link dedicated instruction	READ, SREAD, WRITE, SWRITE, SEND, RECV, RECVS, REQ
Parameter setting	CCPASET
Station number setting	UINI

■MELSECNET/H

Classification	Instruction symbol
Link dedicated instruction	READ, SREAD, WRITE, SWRITE, SEND, RECV, RECVS, REQ, RRUN, RSTOP, RTMRD, RTMWR

Precautions

No processing is performed when unsupported instructions are executed (NOP processing).

A

Simple Motion module

The functions of the Simple Motion module supported by MELSOFT Mirror are listed below.

For details on each function, refer to the following.

📖MELSEC iQ-R Simple Motion Module User's Manual (Application)

📖MELSEC iQ-R Simple Motion Module User's Manual (Advanced Synchronous Control)

RD77MS

○: Supported, △: Partly restricted, ×: Not supported

Function			Availability
Start and stop	Start		○
	Stop		○
	Restart		○
Home position return control	Machine home position return	Proximity dog method	○
		Count method 1	○
		Count method 2	△*1
		Data set method	○
		Scale origin signal detection method	△*1
		Driver home position return method	△*1
	Fast home position return		○
	Home position return setting condition		△*1
Major positioning control	Operation pattern	Independent positioning control	○
		Continuous positioning control	○
		Continuous path control	○
	1- to 4-axis linear interpolation control		○
	Fixed-feed control		○
	2-axis circular interpolation control	Sub point designation	○
		Center point designation	○
	3-axis helical interpolation control	Sub point designation	○
		Center point designation	○
	Speed control		○
	Speed-position switching control		△*1
	Position-speed switching control		△*1
	Current value change		○
	NOP instruction		○
	JUMP instruction		○
	LOOP		○
	LEND		○
High-level positioning control	Control method	Block start	○
		Condition start	○
		Wait start	○
		Simultaneous start	○
		Repeated start (FOR loop)	○
		Repeated start (FOR condition)	○
		Multiple axes simultaneous start	○
	Condition data	Device X	○
		Device Y	○
		Buffer memory	○
		Positioning data No.	○
Manual control	JOG operation		○
	Inching operation		○
	Manual pulse generator operation		○
Inter-module synchronization function			○

Function			Availability
Expansion control	Position control mode		△*1
	Speed control mode		△*1
	Torque control mode		△*1
	Continuous operation to torque control mode		△*1
Control sub function	Sub functions specifically for machine home position return	Home position return retry function	○
		Home position shift function	○
	Functions to compensate the control	Backlash compensation function	○
		Electronic gear function	○
		Near pass function	○
	Functions to limit the control	Speed limit function	○
		Torque limit function	○
		Software stroke limit function	○
		Hardware stroke limit function	○
		Forced stop function	○
	Functions to change the control details	Speed change function	○
		Override function	○
		Acceleration/deceleration time change function	○
		Torque change function	○
		Target position change function	○
	Functions related to start	Pre-reading start function	○
	Absolute position system		○
	Functions related to stop	Stop command processing for deceleration stop function	○
		Continuous operation interrupt function	○
		Step function	○
	Other functions	Skip function	○
		M code output function	○
		Teaching function	○
		Command in-position function	○
		Acceleration/deceleration processing function: Trapezoidal	○
		Acceleration/deceleration processing function: S-curve	○
		Deceleration start flag function	○
		Speed control 10 × multiplier setting for degree axis function	○
		Operation setting for incompletion of home position return function	○
	Servo ON/OFF	Servo ON/OFF	○
		Follow up function	×*1

A

Function				Availability
Common functions	Parameter initialization function			○
	Execution data backup function			○
	External input signal select function	External input signal setting	Simple Motion module	△*1
			Servo amplifier	△*1
			Buffer memory	○
		Signal logic switching		○
		External input connection connector		△*1
	History monitor function	Start history		○
		Current value history		○
	Amplifier-less operation function			×*1
	Virtual servo amplifier function			○
	Driver communication function			△*1
	Mark detection function			△*1
	Optional data monitor function			△*1
	Event history function			○
	Connect/disconnect function of SSCNET communication			×*1
	Online module change			×*1
	Test mode			○
	Functions related to peripheral connection	Digital oscilloscope		△*1
		Offline digital oscilloscope		○
		Communications with MR Configurator		×*1
		Backup/restoration		×*1
Advanced synchronous control	Input axis module	Servo input axis		○
		Synchronous encoder axis	Incremental synchronous encoder	○
			Synchronous encoder via servo amplifier	△*1
			Synchronous encoder via CPU	○
	Cam function	Cam data operation function		○
		Cam auto-generation function	Cam for rotary cutter	○
	Advanced synchronous control	Main shaft module	Main shaft gear	○
			Main shaft clutch	△*1
			Composite main shaft gear	○
		Auxiliary shaft module	Auxiliary shaft gear	○
			Auxiliary shaft clutch	△*1
			Composite auxiliary shaft gear	○
		Clutch	Control method for clutch	○
			Smoothing method for clutch	○
		Speed change gear module		○
		Output axis module		○
		Synchronous control change function		○
		Phase compensation function		○
	Advanced synchronous control initial position	Synchronous control initial position	Cam axis position value per cycle restoration	○
			Cam reference position restoration	○
			Cam axis command position value restoration	○
		Cam position calculation function		○

*1 For details, refer to the following. (☞ Page 277 Restrictions)

■Restrictions

Function			Restrictions
Home position return control	Machine home position return	Count method 2	The movement amount (deviation) compensation is 0 when the DOG signal is input.
		Scale origin signal detection method	The operation stops when the first zero point signal is detected.
		Driver home position return method	The operation will be the same as that in amplifier-less operation mode. When this method is set, the home position return completes without the positioning operation.
	Home position return setting condition		The operation will be the same as that in amplifier-less operation mode.
Major positioning control	Speed-position switching control		The movement amount compensation is 0 when the CHANGE signal is input.
	Position-speed switching control		The movement amount compensation is 0 when the CHANGE signal is input.
Expansion control	Position control mode		The calculation method for feedback position and motor rotation speed differs depending on the control mode. For details, refer to the following. ☞ Page 279 Calculation method for feedback position and motor rotation speed
	Speed control mode		The calculation method for feedback position and motor rotation speed differs depending on the control mode. For details, refer to the following. ☞ Page 279 Calculation method for feedback position and motor rotation speed
	Torque control mode		The calculation method for feedback position and motor rotation speed differs depending on the control mode. For details, refer to the following. ☞ Page 279 Calculation method for feedback position and motor rotation speed The speed limit signal of "[Md.119] Servo status2" turns on immediately when the torque command value is set to a value other than 0.
	Continuous operation to torque control mode		The calculation method for feedback position and motor rotation speed differs depending on the control mode. For details, refer to the following. ☞ Page 279 Calculation method for feedback position and motor rotation speed The speed limit signal of "[Md.119] Servo status2" turns on immediately when the torque command value is set to a value other than 0.
Control sub function	Servo ON/OFF	Follow up function	Not supported

Function				Restrictions
Common functions	External input signal select function	External input signal setting	Simple Motion module	When the built-in DI number is set, the signal detection position estimation compensation processing by the H/W counter is not performed.
			Servo amplifier	The operation will be the same as that in amplifier-less operation mode.
		External input connection connector		Signals are input from the I/O setting.
	Amplifier-less operation function			It always operates in amplifier-less operation mode. The parameter for "[Cd.137] Amplifier-less operation mode switching request" is not read. "[Md.51] Amplifier-less operation mode status" is always set to "1". For control switching command requests (gain/PI-PID/control loop), only responses are returned.
	Driver communication function			According to the command value to the master axis, the slave axis is emulated so that it behaves in the same way as the master axis.
	Mark detection function			The detection data compensation processing by the H/W counter latch is not performed when a signal is detected.
	Optional data monitor function			The servo amplifier data monitor items are fixed to 0 except for the following. • Servo motor speed • Encoder multiple revolution counter • Position feedback • Encoder position within one revolution
	Connect/disconnect function of SSCNET communication			The parameter for "[Cd.102] SSCNET control command" is not read. "[Md.53] SSCNET control status" is always set to "0".
	Online module change			Not supported
	Functions related to peripheral connection	Digital oscilloscope		The servo amplifier data monitor items are fixed to 0 except for the following. • Servo motor speed • Encoder multiple revolution counter • Position feedback • Encoder position within one revolution
		Communications with MR Configurator		Not supported
		Backup/restoration		Not supported
Advanced synchronous control	Input axis module	Synchronous encoder axis	Synchronous encoder via servo amplifier	The servo motor operates according to pulse commands input from the I/O setting
	Advanced synchronous control	Main shaft module	Main shaft clutch	In the main shaft clutch control using a high speed input request signal, the signal detection position estimation compensation processing by the H/W counter is not performed when a signal is detected.
		Auxiliary shaft module	Auxiliary shaft clutch	In the auxiliary shaft clutch control using a high speed input request signal, the signal detection position estimation compensation processing by the H/W counter is not performed when a signal is detected.

■Calculation method for feedback position and motor rotation speed

Data	Control mode	Description
Feedback position	Position control	The servo command value is set.
	Speed control	The Δ movement amount (pulse) for each control cycle is calculated from the set command speed, and the feedback position and command value are updated.
	Torque control/ Continuous operation to torque control	• When the torque command value is 0 Assuming that the motor is stopped, the previous feedback position is set. • When the torque command value is positive Assuming that the motor is rotating in the forward run direction at the speed control value, the Δ movement amount (pulse) for each control cycle is calculated from the set speed limit value, and the feedback position and command value are updated. • When the torque command value is negative Assuming that the motor is rotating in the reverse run direction at the speed limit value, the Δ movement amount (pulse) for each control cycle is calculated from the set speed limit value, and the feedback position and command value are updated.
Motor rotation speed	Position control	The motor rotation speed converted from the Δ movement amount for each control cycle is set.
	Speed control	The motor rotation speed converted from the Δ movement amount for each control cycle is set.
	Torque control/ Continuous operation to torque control	• When the torque command value is 0 Assuming that the motor is stopped, 0 is set. • When the torque command value is positive Assuming that the motor is rotating in the forward run direction at the speed control value, the set + speed limit value (0.01rpm) is set. • When the torque command value is negative Assuming that the motor is rotating in the reverse run direction at the speed limit value, the set - speed limit value (0.01rpm) is set.

Notes

The following points should be considered when debugging with MELSOFT Mirror.

Simulation time

Since the instruction processing speed differs between MELSOFT Mirror and a CPU module, MELSOFT Mirror operates as follows.

- A simulation time is calculated by adding an instruction processing time value each time an instruction is executed in a CPU module. MELSOFT Mirror operates behind the actual time when the simulation time is longer than the actual time due to the performance of a personal computer. If the actual time is longer than the simulation time, the simulation time is advanced according to the actual time.
- If the actual scan time is longer than a value set in the following CPU parameter, a constant scan time error does not occur even after a constant scan time is exceeded; therefore, instructions continue. ([CPU Parameter] ⇨ [RAS Setting] ⇨ [Constant Scan Setting])

For the instruction processing time, refer to the following.

📖MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks)

IP address setting of the CPU module

The IP address of the CPU module is set in the module parameter of the CPU module using the engineering tool. However, depending on the application, the IP address set using MELSOFT Mirror may be used.*1 Which IP address to use depends on the application. For details, refer to the following.

Application	IP address
Web server function*2 SLMP communications	MELSOFT Mirror Simulation setting IP address set for each CPU module
Simple CPU communication function*2	Engineering tool CPU module parameter IP address

*1 MELSOFT Mirror can use either a private IP address that is usable on the network to which the personal computer running MELSOFT Mirror is connected (10.0.0.1 to 10.255.255.254, 172.16.0.1 to 172.31.255.254, or 192.168.0.1 to 192.168.255.254) or the loopback address (127.0.0.1). When any IP address other than the above is set, operation is not guaranteed.

*2 For details on the Web server function and simple CPU communication function, refer to the following.
📖MELSEC iQ-R Ethernet User's Manual (Application)

Fixed scan execution type program

A program is executed in accordance with the simulation time.

If the load on MELSOFT Mirror is high due to the number or cycle of the fixed scan execution type programs, the response from MELSOFT Mirror may be delayed and a communication error may occur.

Handling of floating-point numbers

A rounding error may occur in the operation result of an instruction using floating-point numbers. The result will not be the same as the operation result of the CPU module.

Simulation of network modules

■Cable connections between network modules

All network modules with the same network number are considered to be connected.

■Online operation to other stations

With MELSOFT Mirror, online operations can be performed only for the own station.

■Station number duplication

If a duplicate station number is detected before data transmission is enabled, a duplication error occurs in both stations.
If a duplicate station number is detected while data transmission is enabled, a duplication error occurs in the station where the setting is configured later.

■Network topology setting

The following modules operate in a star topology regardless of the network topology setting.

- RJ71EN71 (CCIEC)
- RJ71EN71 (E+CCIEC)
- RJ71EN71 (CCIEF)
- RJ71EN71 (E+CCIEF)
- RJ71GF11-T2

■Troubleshooting of network modules

For details on troubleshooting, refer to the following.

📖MELSEC iQ-R CC-Link IE Controller Network User's Manual (Application)
📖MELSEC iQ-R CC-Link IE Field Network User's Manual (Application)

Network module simulation does not support some troubleshooting methods that are performed on network modules actually connected to the system.
The following table lists the troubleshooting methods and their support status.
○: Supported, ×: Not supported

Troubleshooting method	Support status
Checking the LED status	○
Checking the network status by executing the CC-Link IE Field Network diagnostics or CC-Link IE Controller Network diagnostics	○
Checking the error information by executing the module diagnostics for the network module	○
Checking the event history by executing the module diagnostics for the CPU module*1	○
Checking the system monitor and detailed module information	×
Monitoring the link special relay (SB) and link special register (SW)	○

*1 How to check the event history:
Select [Diagnostics] ⇨ [Module Diagnostics (CPU Diagnostics)], and click the [Event History] button on the Module Diagnostics window.

Ports of network modules

The RJ71EN71(CCIEC), RJ71EN71(CCIEF), and RJ71GF11-T2 use port 1 for network connection. Port 2 cannot be used.

Watchdog timer

The time required to execute an instruction is calculated as 2ns. The time it takes for the watchdog timer to timeout when executed on a simulator has an error compared to the actual module.

Writing parameters to intelligent function modules

When writing parameters from the "Online Data Operation" window of the engineering tool, the availability of writing differs depending on whether the target intelligent function module can be operated on MELSOFT Mirror.
○: Writable, ×: Not writable, —: Not supported

Module operation setting	Enabled	Disabled
Simple Motion module setting	○	×
Other than above	—	×

Troubleshooting of Simple Motion module

For details on troubleshooting, refer to the following.

MELSEC iQ-R Simple Motion Module User's Manual (Application)

MELSEC iQ-R Simple Motion Module User's Manual (Network)

Simple Motion module simulation does not support some troubleshooting methods that are performed on a Simple Motion module actually connected to the system.

The following table lists the troubleshooting methods and their support status.

○: Supported, ×: Not supported

Troubleshooting method	Support status
Checking the LED status	○
Checking the error information by executing the module diagnostics for the Simple Motion module	×
Checking the event history by executing the module diagnostics for the CPU module*1	○
Monitoring in the Drive Tool List window	○
Monitoring in the Device/Buffer Memory Batch Monitor window	○

*1 How to check the event history:
Select [Diagnostics] ⇨ [Module Diagnostics (CPU Diagnostics)], and click the [Event History] button on the Module Diagnostics window.

Redundant system

The system switching function is not supported and the system operates only in separate mode.

Multiple CPU system function

The multiple CPU system function is supported only by the CPU modules that can be simulated with MELSOFT Mirror (those with the module operation enable/disable setting "Enabled").

The function is not supported by the CPU modules that cannot be simulated with MELSOFT Mirror (those with the module operation enable/disable setting "Disabled").

Errors in CPU modules that cannot be simulated with MELSOFT Mirror and in modules controlled by those CPU modules are not detected.

Self-diagnostic function errors

The following self-diagnostic errors are not detected.

- 1900H: Constant scan time error
- 1A00H: Safety cycle time exceeded
- 1A01H: Safety cycle processing error

Error information during module diagnostics

If multiple errors are detected in the CPU module on MELSOFT Mirror, the order in which they are detected may differ from the CPU module actually connected to the system. Therefore, the latest error displayed on the Module Diagnostics window and the order of errors in the event history may differ from the CPU module actually connected to the system.

SD memory card

While MELSOFT Mirror is running, the setting of a virtual SD memory card file cannot be changed.

■File operation instructions

When a virtual SD memory card file is set in the MELSOFT Mirror window, the operation of the SD memory card can be simulated using the file operation instructions (SP.FREAD, SP.FWRITE, SP.FDELETE, SP.FCOPY, SP.FMOVE, SP.FRENAME, and SP.FSTATUS).

If a virtual SD memory card file has not been set, the operation of the SD memory card cannot be simulated, even if the file operation instructions are executed.

If an error (error code: 8000H) occurs during the execution of a file operation instruction, set a virtual SD memory card file in the MELSOFT Mirror window.

Timer, clock function, and system clock

The current value of a timer, clock function, and system clock operate based on the simulation time. When the system is powered on for the first time after configuration, the time on the personal computer is applied to the time of the clock function of the CPU module. When the system is powered on for the second time, the elapsed time on the personal computer while the system was off is added to the time of the clock function of the CPU module.

Database

A database created in the actual SD memory card and one created in the virtual SD memory card file are not compatible. (Even if they are created based on the same data, the size of the created databases is different.)

To use a database in a simulation, create the database in the virtual SD memory card file set in the CPU module of MELSOFT Mirror as follows.

- Database access instruction: DBIMPORT instruction
- Label communication function: Write global label assignment information to the simulator.

For information on how to create a database, refer to the following.

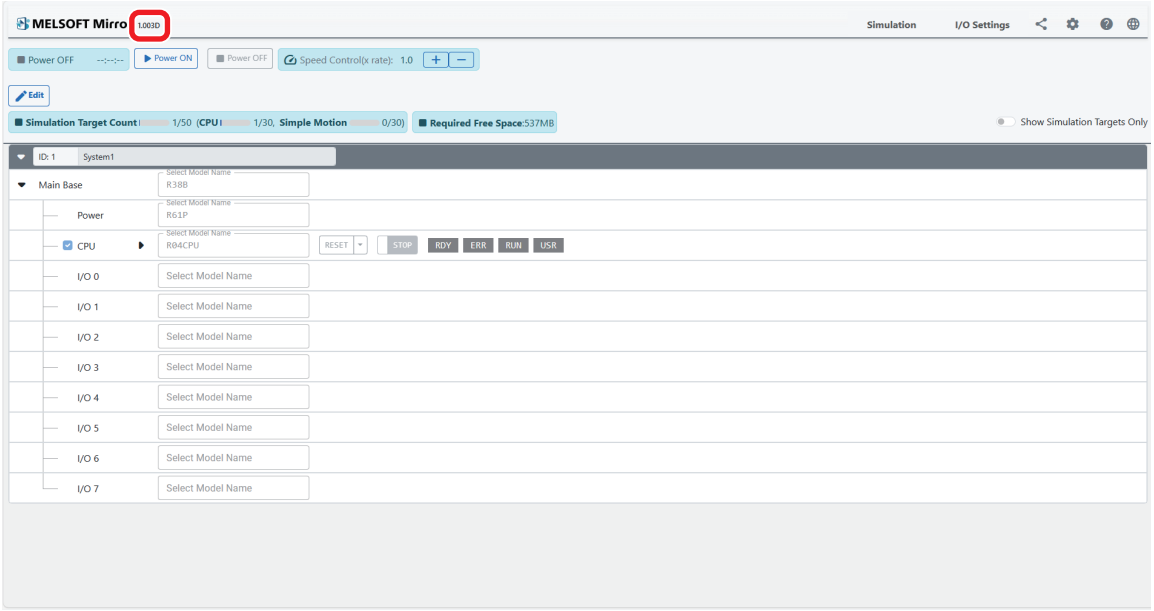
MELSEC iQ-R CPU Module User's Manual (Application)

A database created on MELSOFT Mirror is not case sensitive. If the same string with different uppercase/lowercase characters is used for any of the following three items in each instruction of the database function, an error may occur.

- Table name
- Field name
- Index name

Appendix 2 How to Check the Version

The version of MELSOFT Mirror is displayed on the top window.



Appendix 3 Software Licenses

This software consists of multiple software components. Each of those components is copyrighted by Mitsubishi Electric or a third party.
The open source software incorporated into this software and the software conditions are listed in a file in the installation folder.

Ex.
File location when using the default installation folder
C:\Program Files\MELSOFT\LCS\Doc\Info.txt
Source codes are not distributed for the software of Mitsubishi Electric or third parties.
Please refrain from inquiring about the source codes of the open source software.

A

MEMO

INDEX

T

V

W

REVISIONS

*The manual number is given on the bottom left of the back cover.

Revision date	*Manual number	Description
June 2024	SH(NA)-082701ENG-A	First edition
October 2024	SH(NA)-082701ENG-B	■Added function SLMP ■Added or modified parts Section 3.5, Chapter 10, Section 11.2, Appendix 1
March 2026	SH(NA)-082701ENG-C	■Added or modified parts Section 2.1, 2.2, 3.1, 3.2, Chapter 4, Section 4.2, Chapter 6, 10, Section 12.2, Appendix 1

Japanese manual number: SH-082663-D

INFORMATION AND SERVICES

For further information and services, please contact your local Mitsubishi Electric sales office or representative.
Visit our website to find our locations worldwide.

MITSUBISHI ELECTRIC Factory Automation Global Website
Locations Worldwide
www.MitsubishiElectric.com/fa/about-us/overseas/

TRADEMARKS

Microsoft, Microsoft Edge, Windows, and Windows Server are trademarks of the Microsoft group of companies.
The company names, system names and product names mentioned in this manual are either registered trademarks or trademarks of their respective companies.
In some cases, trademark symbols such as '™' or '®' are not specified in this manual.

COPYRIGHTS

The screens (screenshots) are used in accordance with the Microsoft Corporation guideline.
For the open source software used in this product, refer to the following.
☞ Page 285 Software Licenses

TRANSLATION OF THE ORIGINAL INSTRUCTIONS

SH(NA)-082701ENG-C(2603)MEE
MODEL: SW1DND-LCS-O-E

MITSUBISHI ELECTRIC CORPORATION

HEAD OFFICE: TOKYO BLDG., 2-7-3, MARUNOUCHI, CHIYODA-KU, TOKYO 100-8310, JAPAN
NAGOYA WORKS: 1-14, YADA-MINAMI 5-CHOME, HIGASHI-KU, NAGOYA 461-8670, JAPAN