MITSUBISHI ELECTRIC

Changes for the Better

for a greener tomorrow

eco Changes

FATEC

Mitsubishi Programmable Controllers Training Manual MELSEC iQ-R Process Control Basic Course

SAFETY PRECAUTIONS

(Read these precautions before exercise.)

When designing the system, always read the relevant manuals and give sufficient consideration to safety.

During the exercise, pay full attention to the following points and handle the product correctly.

[EXERCISE PRECAUTIONS]

⚠WARNING

● Do not touch the terminals while the power is on to prevent electric shock.

● Before opening the safety cover, turn off the power or ensure the safety.

⚠CAUTION

● Follow the instructor's direction during the exercise.

● Do not remove the module of the demonstration machine or change wirings without permission.
Doing so may cause failures, malfunctions, personal injuries and/or a fire.

● Turn off the power before mounting or removing the module.
Failure to do so may result in malfunctions of the module or electric shock.

● When the demonstration machine emits abnormal odor/sound, press the "Power switch" or
"Emergency switch" to turn off.

● When a problem occurs, notify the instructor as soon as possible.

REVISIONS

*The manual number is given on the bottom left of the back cover.

Revision date	*Manual number	Description
August 2020	SH(NA)-082349ENG-A	First edition

TRADEMARKS

Ethernet is a registered trademark of Fuji Xerox Co., Ltd. in Japan.

Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

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.

CONTENTS

CONTENTS

5

INTRODUCTION

To help users acquire the knowledge required for configuring a process control system using the MELSEC iQ-R series process control, this manual describes the functions and specifications of hardware and software, and procedures to write programs in FBD language.

RELEVANT MANUALS

Manual name [manual number]	Description	Available form
MELSEC iQ-R CPU Module User's Manual (Startup) [SH-081263ENG]	Specifications, procedures before operation, and troubleshooting of the CPU module	e-Manual PDF
MELSEC iQ-R CPU Module User's Manual (Application) [SH-081264ENG]	Memory, functions, devices, and parameters of the CPU module	e-Manual PDF
MELSEC iQ-R Ethernet/CC-Link IE User's Manual (Startup) [SH-081256ENG]	Specifications, procedures before operation, system configuration, wiring, and communication examples of Ethernet, CC-Link IE Controller Network, and CC-Link IE Field Network	e-Manual PDF
MELSEC iQ-R Programming Manual (Process Control Function Blocks/Instructions) [SH-081749ENG]	General process FBs, tag access FBs, tag FBs, and process control instructions designed for process control	e-Manual PDF
MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks) [SH-081266ENG]	Instructions for the CPU module and standard functions/function blocks	e-Manual PDF
MELSEC iQ-R Programming Manual (Module Dedicated Instructions) [SH-081976ENG]	Dedicated instructions for the intelligent function modules	e-Manual PDF
PX Developer Version 1 Operating Manual (Monitor Tool) [SH-080370E]	Explains the operating method of PX Developer monitor tool and the monitoring and controlling method of the DDC processing using the tag FB.	PDF
GX Works3 Operating Manual [SH-081215ENG]	System configuration, parameter settings, and online operations of GX Works3	e-Manual PDF

Point

e-Manual refers to the Mitsubishi Electric FA electronic book manuals that can be browsed using a dedicated tool.

e-Manual has the following features:
 • Required information can be cross-searched in multiple manuals.
 • Other manuals can be accessed from the links in the manual.
 • The hardware specifications of each part can be found from the product figures.
 • Pages that users often browse can be bookmarked.
 • Sample programs can be copied to an engineering tool.

1 MELSEC iQ-R PROCESS CONTROL

1.1 Features

Compatible with small- to large-size process control

The MELSEC iQ-R series process CPUs can execute high-speed PID control and high-speed sequence control simultaneously, with four types available ranging from those of low-capacity to high-capacity (maximum 1200K steps). Also, a redundant system can be set in combination with a redundant function module, and various network modules supporting the redundant system configuration provide flexibility in catering to customers' needs, resulting in significantly enhanced reliability.

Enables creation of highly reliable systems

The MELSEC iQ-R series redundant system provides redundant layers of monitoring (SCADA), controllers, network, and I/O to make the system highly reliable. This allows the system to be applied to special systems that, such as in dedicated systems, require high reliability.

Data visualization and collection

By linking the SCADA software with the MELSEC iQ-R series, the status of the entire factory can be monitored, helping customers increase productivity and product quality.

Simplified engineering using the integrated engineering software

GX Works3, the engineering software for the MELSEC iQ-R series programmable controller, integrates the process control program editor (FBD language) and other program editors. GX Works3 allows the users to easily create a process control system through process control tag label information sharing, use of simplified program structures, and simple programmable controller read/write operations.

Operation mode of the Process CPU

A Process CPU operates in one of the following two modes.

■Process mode
Use this mode when a redundant system is not built. When "Mode" has been set to "Process" at project creation with the engineering tool, the Process CPU is started up in the process mode.

■Redundant mode
Use this mode when a redundant system is built. When "Mode" has been set to "Redundant" at project creation with the engineering tool, the Process CPU is started up in the redundant mode.

> **Point**
>
> Set the redundant mode for both systems in a redundant system.

1.2 Process CPU and Redundant System

Redundant system

This system consists of two basic systems that have a CPU module, a power supply module, a network module, or other modules for each of them. Even if an error occurs in one system, control is continued with the other system.

A redundant configuration of the systems of main base units is available when redundant function modules are used and Process CPUs are operated in the redundant mode. To configure a redundant system, configure two basic systems that have the same configuration including a redundant function module, and connect the redundant function modules in those two systems with tracking cables. Connecting the redundant function modules with two tracking cables also makes the cables to be redundant.

(1) Process CPU (redundant mode)
(2) Redundant function module
(3) Tracking cable
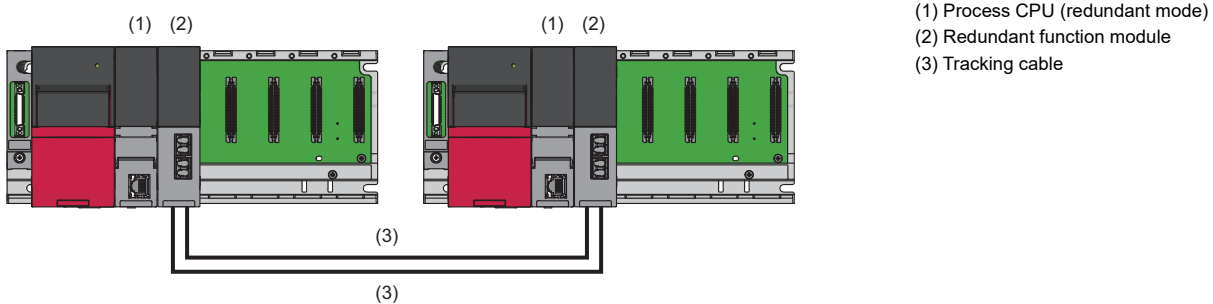
System

System A and system B

In a redundant system, one system is specified as a system A and the other is specified as a system B to distinguish between two systems connected with tracking cables. Set the system A or B with the engineering tool.

Point

When one system is set to system A, the other system is automatically set to system B, and vice versa.

Control system and standby system

In a redundant system, the CPU module in one system executes programs to perform controls. The other system is in the standby state and does not perform controls. The system that performs controls is called a control system and the system in the standby state is called a standby system.

A control system or standby system is determined when both the systems are started and ready for tracking communications.
(☞ Page 11 Determination of control system/standby system)

System switching between the control system and standby system

In a redundant system, data link is performed between the redundant function modules connected with tracking cables and data required for operation is transferred (tracking transfer) at every scan from the control system to the standby system. If an error occurs in the control system, the standby system will function as the new control system and continue the control using the data that the system has received.
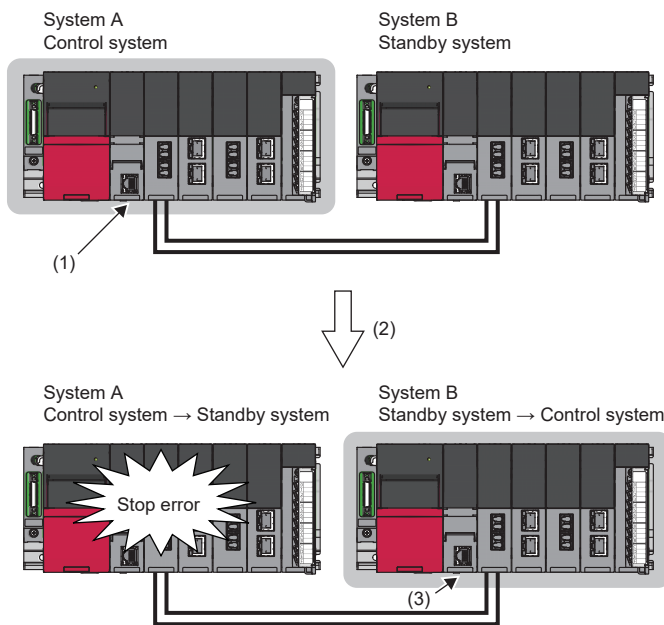
(1) The control system is operating without errors. (Data is transferred to the standby system from the control system at every scan.)
(2) A stop error occurs in the control system.
(3) The standby system now functions as the new control system and continues the control using the data that the system has received.

For details on the system switching, refer to the following.

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

System consistency check

In a redundant system, whether both systems have the same configuration is checked to switch the system and continue the operation without causing a system failure.

Operation modes of the redundant system

A redundant system operates in one of the following two modes.

Operation mode	Description
Backup mode	A mode used to normally operate the redundant system When an error or failure has occurred in the control system, the standby system is switched to the control system to continue the operation. Even if a system failure has occurred in the control system, the standby system of the redundant system continues the operation because data is transferred from the control system to the standby system at every scan.
Separate mode	A mode used to maintain the redundant system without stopping controls In separate mode, the control system CPU module and the standby system CPU module can execute different programs. The program of the standby system can be modified and the operation of the program can be checked with the data tracked from the control system while the system is being controlled in the control system.

A redundant system is started in the backup mode. To switch the operation mode, use the engineering tool.

For details on the differences between the backup mode and separate mode, refer to the following.

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

Determination of control system/standby system

This section describes how to determine which system is the control system and the other is the standby system in the following three cases.

When starting up both systems

Whether each system is the control system or standby system is determined when the both systems are started up by powering off and on or reset of the CPU module and then ready for tracking communications.

 • When both systems are started up simultaneously

The system A operates as the control system and the system B as the standby system.

"Simultaneously" here means that one system is started up within three seconds after the other system is started.

 • When one of the system A and system B is started up first

The system will wait for the start-up of the other system in three seconds. (📖 MELSEC iQ-R CPU Module User's Manual (Application))

When communications for both systems are established, system A operates as the control system and system B as the standby system.

When starting up one system first

When starting up either system A or system B, perform one of the following operations to the CPU module that is waiting for the start-up of the other system. The system of the CPU module will be started as the control system.

 • Online operation

Perform the following operation on the engineering tool.

 ✎ [Online] ⇨ [Redundant PLC Operation] ⇨ [Redundant Operation]

Select "Forced Start of Control System while Waiting for Other System to Start" and click the [Execute] button.

 • Switch operation

Set the RUN/STOP/RESET switch of the CPU module to the RUN, STOP, and then the RUN position again.

Set "Control System Start-up Setting (Switch Operation)" to "Enable" in the CPU parameter in advance. (📖 MELSEC iQ-R CPU Module User's Manual (Application))

 • Operation with the input (X)

Turn on the input (X) set with the parameters.

Set "Control System Start-up Setting (Input (X))" to "Enable" in the CPU parameter in advance. (📖 MELSEC iQ-R CPU Module User's Manual (Application))

When starting up the previous control system as the control system

In a redundant system, when both systems are started up simultaneously, the system A is always specified as the control system and the system B as the standby system.

Even though both systems are temporarily powered off due to a power failure or other causes while the system B is operating as the control system, the system A is started up as the control system when both systems are powered on again.

To start up the system B that was the previous control system as the control system again, create a program that uses SM1636 (Previous control system identification flag).

For a system having network modules, wait until the network modules of the other system start up and then execute the SP.CONTSW instruction.
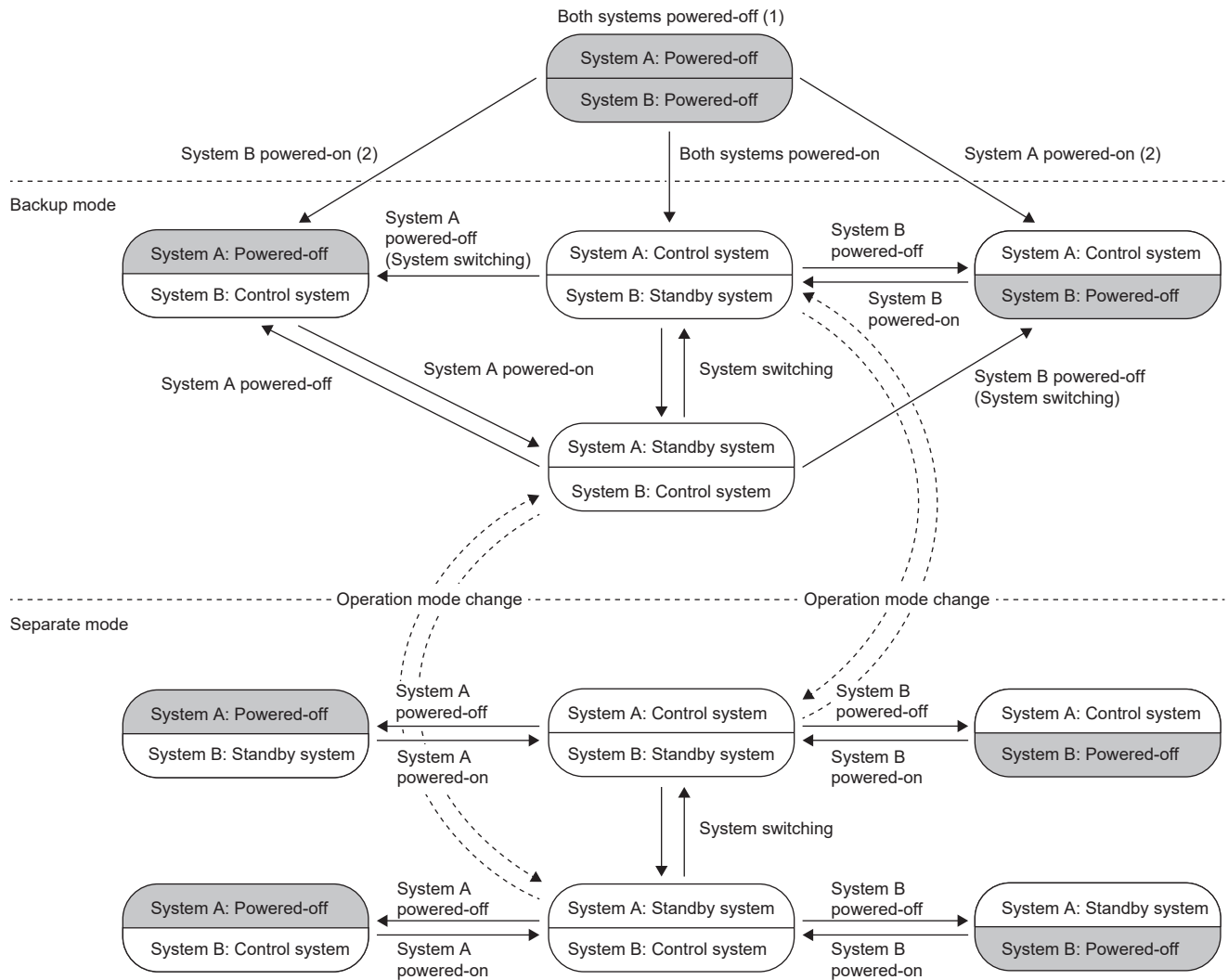
 • Program example

Executing a system switching instruction when the CPU module in the system A is set to the RUN state for the first time switches the system B to the control system.

Turn on SM1646 (System switching by a user) in advance.

State transition of the redundant system

The following figure shows the state transition due to the operation mode change and system switching for the redundant system after start-up.

Both systems powered-off (1)

System A: Powered-off
System B: Powered-off

System B powered-on (2)   Both systems powered-on   System A powered-on (2)

Backup mode

System A: Powered-off
System B: Control system

System A powered-off (System switching)

System A: Control system
System B: Standby system

System B powered-off

System B powered-on

System A: Control system
System B: Powered-off

System A powered-off   System A powered-on   System switching   System B powered-off (System switching)

System A: Standby system
System B: Control system

---- Operation mode change ---- Operation mode change ----

Separate mode

System A: Powered-off
System B: Standby system

System A powered-off / System A powered-on

System A: Control system
System B: Standby system

System B powered-off / System B powered-on

System A: Control system
System B: Powered-off

System switching

System A: Powered-off
System B: Control system

System A powered-off / System A powered-on

System A: Standby system
System B: Control system

System B powered-off / System B powered-on

System A: Standby system
System B: Powered-off

(1) When both systems are powered off, both systems enter the power-off state regardless of the operation mode or system type.
(2) When a system that has been powered on is started as the control system. (📖 MELSEC iQ-R CPU Module User's Manual (Application))

1.3 Intelligent Function Module

The following table lists the intelligent function modules that can be used with the Process CPU.

Each of the CPU module is represented by the following symbols.
- RnP(P): Process CPU (process mode)
- RnP(R): Process CPU (redundant mode)
- Rem: Remote head module
- Rem(R): Remote head module (redundant system)

○: Available, ×: Not available

Module	Model	RnP(P)	RnP(R)	Rem	Rem(R)
Analog-digital converter module	R60AD4, R60ADI8, R60ADV8	○	×	○	○
Channel isolated analog-digital converter module	R60AD8-G, R60AD16-G[*2]	○	×	○	○
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[*2]	○	×	○	○
High speed digital-analog converter module	R60DAH4	○	×	○	○
Simple motion module	RD77MS2, RD77MS4, RD77MS8, RD77MS16	○	×	○[*4]	×
	RD77GF4, RD77GF8, RD77GF16, RD77GF32	○	×	×	×
Channel isolated RTD input module	R60RD8-G	○	×	○	○
Channel isolated thermocouple input module	R60TD8-G	○	×	○	○
Temperature control module[*5]	R60TCTRT2TT2, R60TCTRT2TT2BW, R60TCRT4, R60TCRT4BW	○	×	○	○
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[*6]	RD81MES96	○	×	×	×
High speed data logger module	RD81DL96	○	×	×	×
OPC UA server module	RD81OPC96	○	×	×	×
C intelligent function module	RD55UP06-V	○	×	×	×
Positioning module[*4]	RD75P2, RD75P4, RD75D2, RD75D4	○	×	○	×
Ethernet[*3][*5]	RJ71EN71	○	○	○	○[*5]
CC-Link IE Controller Network	RJ71GP21-SX[*5]	○	○	×	×
	RJ71EN71[*1]	×	×	×	×
CC-Link IE Field Network	RJ71GF11-T2[*5]	○	○	×	×
	RJ71EN71	○	×	×	×
CC-Link[*5]	RJ61BT11	○	○	○	○
Serial communication[*5]	RJ71C24, RJ71C24-R2, RJ71C24-R4	○	×	○	○
AnyWireASLINK master module	RJ51AW12AL	○	×	○	×
BACnet module	RJ71BAC96	○	×	○	×
CANopen module	RJ71CN91	○[*7]	×	×	×
DeviceNet master-slave module	RJ71DN91	○[*9]	×	×	×
PROFIBUS-DP module[*8]	RJ71PB91V	○	○	○	○

*1 When the CC-Link IE Controller Network function of the RJ71EN71 is used, there are restrictions on the firmware versions of the RnCPU, the C Controller module, and the RJ71EN71. (📖 User's Manual (Application) for the module used)

*2 These modules occupy two slots. When these modules are used with the RnCPU or the C Controller module, there are restrictions on each firmware version. (📖 User's Manual (Application) for the CPU module used)

*3 When the RJ71EN71 is used with a remote head module, there are restrictions on network combinations. 📖 MELSEC iQ-R Ethernet/CC-Link IE User's Manual (Startup)

*4 When a redundant system is built with a master station of CC-Link IE Field Network, these modules cannot be used with a remote head module.

*5 When a redundant system is built with these modules, there are restrictions on the firmware version. (📖 Manual for each module)

*6 Depending on the CPU module used, there are restrictions on the firmware version of the MES interface module. (📖 MELSEC iQ-R MES Interface Module User's Manual (Application))

*7   Use the RnCPU with firmware version "28" or later, the RnENCPU with firmware version "17" or later, and the Process CPU with firmware version "12" or later. When the CPU module with a firmware version earlier than ones described above is used, "Target" cannot be set to "RPDO" and "TPDO" under "Refresh Setting" in "Module Parameter". Parameter error (error code: 2220H) occurs.

*8   There are restrictions on the firmware version. (Manual for each module)

*9   Use the RnCPU with firmware version "28" or later, the RnENCPU with firmware version "17" or later, the Process CPU (process mode) with firmware version "12" or later, and the Safety CPU with firmware version "14" or later. When the CPU module with a firmware version earlier than this version is used, there are restrictions on "Refresh Setting" in "Module Parameter".

Features/specifications of network module

Features

■Equipped with two Gigabit Ethernet ports

The module is equipped with two gigabit Ethernet ports to support Ethernet, CC-Link IE Controller Network (twisted pair cable), and CC-Link IE Field Network communications. These two Ethernet ports can be connected to different networks. As one module can handle two networks such as the Ethernet and CC-Link IE Field Network, network construction cost can be reduced. For Ethernet communications, up to 128 connections can be opened simultaneously, allowing for even more Ethernet-compatible modules to be connected.

■Highly reliable optical duplex loop network

The duplex loop transmission method using noise-resistant optical fiber cables provides the loopback function to maintain communications when errors such as cable disconnection and power outage occur.

■Immediate detection of line problems and module errors

The status of the entire CC-Link IE Controller Network is visible in the engineering tool, allowing for line problems and module errors to be detected easily. This allows faulty areas to be quickly identified at trouble occurrence, reducing the system downtime. Programmable controllers in other stations can also be monitored via the network.

■Construction of flexible network

 • Star topology wiring

Use the switching hub to connect each module in a star topology. Slave stations can be easily added to the network using this topology.

 • Line topology wiring

Modules are connected in a line topology.

This topology can reduce wiring cost.

 • Ring topology wiring

Modules are connected in a ring topology.

When an error occurs in a slave station, the loopback function is used to allow data link to be continued with the stations that are operating normally.

■CC-Link IE Field Network synchronous communication function

This function synchronizes control intervals between slave stations over CC-Link IE Field Network according to synchronization cycle specified in the master station.

This allows different slave stations on the same network to operate with the same timing.

Specifications

∎Ethernet

The following table lists the performance specifications of Ethernet.

Item			RJ71EN71	
			Ethernet	Q-compatible Ethernet
Transmission specifications	Data transmission speed		1Gbps/100Mbps/10Mbps	1Gbps[*1]/100Mbps/10Mbps
	Communication mode	1000BASE-T	Full-duplex	—
		100BASE-TX	Full-duplex/half-duplex	
		10BASE-T	Full-duplex/half-duplex	
	Interface		RJ45 connector (Auto MDI/MDI-X)	
	Transmission method		Base band	
	Maximum frame size		• 1518 bytes • 9022 bytes (when jumbo frames are used)	1518 bytes
	Jumbo frame		Available	Not available
	Maximum segment length		100m (length between hub and node)[*2]	
	Number of cascade connections	1000BASE-T	[*3]	
		100BASE-TX	2 levels maximum[*4]	
		10BASE-T	4 levels maximum[*4]	
	IP version		Compatible with IPv4	
Sending/receiving data storage memory	Number of simultaneous open connections		128 connections (connections usable on a program)	16 connections (connections usable on a program)
	Fixed buffer		5K words × 16	1K words × 16
	Socket communications		• 5K words × 48 (when only P1 is used) • 5K words × 112 (when P1 and P2 are used)	—
	Random access buffer		6K words × 1	
Number of occupied I/O points			32 points	
Internal current consumption (5VDC)			0.82A	
External dimensions	Height		106mm (Base unit mounting side: 98mm)	
	Width		27.8mm	
	Depth		110mm	
Weight			0.17kg	

*1 When using 1Gbps, set "Communication Speed" under "Application Settings" to "Automatic Negotiation". ("1Gbps/Full Duplex" cannot be selected.)

*2 For maximum segment length (length between hubs), consult the manufacturer of the hub used.

*3 Consult the manufacturer of the switching hub used.

*4 This applies when a repeater hub is used. For the number of levels that can be constructed when using a switching hub, consult the manufacturer of the switching hub used.

> **Point**
>
> The operation of commercial devices used for the following applications is not guaranteed. Check the operation before using the module.
> • Internet (general public line) (Internet-access service offered by an Internet service provider or a telecommunications carrier)
> • Firewall device(s)
> • Broadband router(s)
> • Wireless LAN

■CC-Link IE Controller Network

The following table lists the performance specifications of CC-Link IE Controller Network.

Item		RJ71GP21-SX
Maximum number of link points per network	LB	32K points (32768 points, 4K bytes)
	LW	128K points (131072 points, 256K bytes)
	LX	8K points (8192 points, 1K bytes)
	LY	8K points (8192 points, 1K bytes)
Maximum number of link points per station	LB	16K points (16384 points, 2K bytes), Extended mode: 32K points (32768 points, 4K bytes)
	LW	16K points (16384 points, 32K bytes), Extended mode: 128K points (131072 points, 256K bytes)
	LX	8K points (8192 points, 1K bytes)
	LY	8K points (8192 points, 1K bytes)
Transient transmission capacity		1920 bytes maximum
Communication speed		1Gbps
Network topology		Duplex loop
Communication cable		Optical fiber cable which satisfies 1000BASE-SX standard: Multi-mode optical fiber (GI)
Maximum station-to-station distance		550m
Overall cable distance		66000m (when 120 stations are connected)
Number of cascade connections		—
Maximum number of connectable stations		120 stations (control station: 1, normal station: 119)[1]
Maximum number of networks		239
Maximum number of groups		32
Communication method		Token ring
Optical fiber specifications		Standard: IEEE 802.3, IEC 60793-2-10 Types A1a.1 Outside diameter of the core/clad: $50\mu m/125\mu m$ Transmission loss: 3.5dB/km or lower [λ=850nm] Transmission band: 500MHz·km or higher (λ=850nm)
Connector specifications		Duplex LC connector Standard: IEC 61754-20 (Type LC connector) Connection loss: 0.3dB or lower Polished surface: PC (Physical Contact) polishing
Laser class (IEC 60825-1/JIS C 6802)		Class 1 laser product
Number of occupied I/O points		32 points
Internal current consumption (5VDC)		0.88A
External dimensions	Height	106mm (Base unit mounting side: 98mm)
	Width	27.8mm
	Depth	110mm
Weight		0.18kg

[1] When using a CC-Link IE Controller Network-equipped module in a normal station, maximum number of connectable stations differs depending on the CPU module used in a control station. For details, refer to the user's manual for the control station used.

■CC-Link IE Field Network

The following table lists the performance specifications of CC-Link IE Field Network.

Item			RJ71GF11-T2
Maximum number of link points per network		RX	16K points (16384 points, 2K bytes)
		RY	16K points (16384 points, 2K bytes)
		RWr	8K points (8192 points, 16K bytes)
		RWw	8K points (8192 points, 16K bytes)
Maximum number of link points per station	Master station	RX	16K points (16384 points, 2K bytes)
		RY	16K points (16384 points, 2K bytes)
		RWr	8K points (8192 points, 16K bytes)
		RWw	8K points (8192 points, 16K bytes)
	When the submaster function is used[3]	Master operating station	RX	16K points
			RY	16K points (Own station send range is 2K points.)
			RWr	8K points
			RWw	8K points (Own station send range is 1024 points.) 8K points when communication mode is "High-Speed" (Own station send range is 256 points.)
		Submaster operating station[1]	RX	2K points (assigned for the station number 0 or submaster station)
			RY	2K points (assigned for the station number 0 or submaster station)
			RWr	1024 points (assigned for the station number 0 or submaster station) 256 points when communication mode is "High-Speed"
			RWw	1024 points (assigned for the station number 0 or submaster station) 256 points when communication mode is "High-Speed"
	Local station[1][3]		RX	2K points (2048 points, 256 bytes)
			RY	2K points (2048 points, 256 bytes)
			RWr	1K points (1024 points, 2K bytes) 256 points (512 bytes) when communication mode is "High-Speed"
			RWw	1K points (1024 points, 2K bytes) 256 points (512 bytes) when communication mode is "High-Speed"
Safety communications[2]	Maximum number of safety connections per network		1814 connections
	Maximum number of safety connections per station		120 connections
	Maximum number of link points per safety connection		8 words (Input: 8 words, output: 8 words)
Transient transmission capacity			1920 bytes maximum
Communication speed			1Gbps
Network topology			Line topology, star topology (coexistence of line topology and star topology is also possible), and ring topology
Communication cable			Ethernet cable which satisfies 1000BASE-T standard: Category 5e or higher, straight cable (double shielded, STP)
Maximum station-to-station distance			100m (conforms to ANSI/TIA/EIA-568-B (Category 5e))
Overall cable distance			Line topology: 12000m (when 121 stations are connected) Star topology: Depends on the system configuration. Ring topology: 12100m (when 121 stations are connected)
Number of cascade connections			20 levels maximum
Maximum number of connectable stations			121 stations (master station: 1, slave station: 120)
Maximum number of networks			239
Communication method			Token passing
Number of occupied I/O points			32 points
Internal current consumption (5VDC)			0.82A
External dimensions	Height		106mm (Base unit mounting side: 98mm)
	Width		27.8mm
	Depth		110mm
Weight			0.17kg

*1   The maximum number of points that a master station can assign to one station. A submaster station and a local station can receive the data from other stations in addition to this number of points.

*2   This function is not available for the RJ71EN71. For the RJ71GF11-T2, availability depends on the firmware version.
     (□ MELSEC iQ-R CC-Link IE Field Network User's Manual (Application))

*3   Submaster station and local stations cannot be connected when "Communication Mode" under "Application Settings" is "High-Speed Remote Net".

Features/specifications of analog module

Features

■High-frequency noise filtering

The analog input module is equipped with a primary delay filter. By using the primary delay filter, analog input signals removed of high-frequency noise components can be obtained. Programming is not necessary in using this filter because only the time constant of the filter needs to be set by a parameter.

■Event-driven program execution using warning output

The enhanced interrupt function using warning output flags makes preventive maintenance easier. If the measured value or variation rate of analog input signals exceeds the specified upper/lower limit, the interrupt processing is executed regardless of the scan time of the program, allowing the error to be handled promptly.

■High-speed data sampling without dependence on scan time

The analog input module provides a logging function that collects analog input data highly needed for industrial applications at high speed. Data is collected in the specified cycle, and each channel can store up to 10000 points of logging data.
Data collection can be stopped via a program at any timing or by the hold trigger that occurs when the data status changes. This function allows analog input data before/after the hold trigger to be saved, allowing for easy identification of occurred phenomena and collection of test data. One example of application is a test device for a motor. While test pattern control commands are being sent from the programmable controller to the inverter or dynamometer, test data can be collected at high speed from the motor being tested.

■High-speed and seamless analog waveform output

The analog output module is equipped with a function to register any waveform data and continuously analog output it in a specified conversion cycle. The module can be applied to analog (torque) control in a pressing machine or injection molding machine by setting automatic output of preregistered control waveforms, allowing for quicker and seamless control by a program. Because analog waveform control is possible simply by registering waveform data with the module, programming is no longer necessary for creating waveforms for line control and other repeated controls, thereby reducing man-hours required for programming.

■Shift operation and scaling without programming

Shift operation and scaling can be easily set using parameters, and there is no need to create a dedicated program. This helps to reduce cost associated with program development and suppress the size of programs.

1 MELSEC iQ-R PROCESS CONTROL
1.3 Intelligent Function Module

Specifications

■Channel isolated thermocouple input module

The following table describes the performance specifications of the thermocouple input module.

Item		R60TD8-G
Number of analog input channels		8 points (8 channels) + cold junction compensation channel/1 module
Output	Temperature measured value	16-bit signed binary data (-2700 to 18200)
	Scaling value	16-bit signed binary data
Thermocouple compliance standards		IEC 60584-1(1995), IEC 60584-2(1982), JIS C1602-1995
Applicable thermocouples and conversion accuracy		Refer to the following. MELSEC iQ-R Channel Isolated Thermocouple Input Module/Chanel Isolated RTD Input Module User's Manual (Startup)
Cold junction compensation		±1.0℃
Accuracy[1]		Refer to the following. MELSEC iQ-R Channel Isolated Thermocouple Input Module/Chanel Isolated RTD Input Module User's Manual (Startup)
Resolution		B, R, S, N: 0.3℃ K, E, J, T: 0.1℃
Conversion speed[2]		30ms/channel
Isolation method[3]	Between the thermocouple input channel and programmable controller power supply	Transformer isolation
	Between the thermocouple input channels	Transformer isolation
Withstand voltage	Between the thermocouple input channel and programmable controller power supply	500VACrms for 1 minute
	Between the thermocouple input channels	1000VACrms for 1 minute
Insulation resistance	Between the thermocouple input channel and programmable controller power supply	10MΩ or more at 500VDC
	Between the thermocouple input channels	10MΩ or more at 500VDC
Disconnection detection		Enabled
Number of offset/gain settings[4]		50000 times maximum
Number of occupied I/O points		16 points (I/O assignment: intelligent 16 points)
External interface		40-pin connectors
Applicable wire size	When A6CON1 or A6CON4 is used	0.088 to 0.3㎟ (28 to 22 AWG) (stranded wire)
	A6CON2	0.088 to 0.24㎟ (28 to 24 AWG) (stranded wire)
Connectors for external devices		A6CON1, A6CON2, A6CON4 (sold separately)
Internal current consumption (5VDC)		0.36A
External dimensions	Height	106mm
	Width	27.8mm
	Depth	110mm
Weight		0.19kg

*1 Cases when affected by noise are excluded.
*2 The conversion speed is defined by the time it takes for the temperature measured value to be stored in the buffer memory in sampling processing.
*3 The section between the cold junction compensation channel and the programmable controller power supply is not isolated.
*4 If the number of settings exceeds 50000, the offset/gain setting maximum number of writes reach error (error code: 1080H) occurs.

■Channel isolated digital-analog converter module

The following table describes the performance specifications of the D/A converter module.

Item	R60DA8-G, R60DA16-G		
Number of analog output channels	• R60DA8-G: 8 points (8 channels) • R60DA16-G: 16 points (16 channels)		
Digital input	16-bit signed binary data (-32768 to 32767)		
Analog output voltage	-12 to 12VDC (external load resistance value 1KΩ or more)		
Analog output current	0 to 20mADC (external load resistance value 0Ω to 600Ω) 0 to 22mADC (external load resistance value[6])		
I/O conversion characteristic, resolution[1]	Analog output range	Digital value	Resolution
	Voltage	0 to 5V	0 to 32000	156.3μV
		1 to 5V		125.0μV
		-10 to 10V	-32000 to 32000	312.5μV
		-12 to 12V		378.4μV
		1 to 5V (extended mode)	-8000 to 36000[7]	125.0μV
		User range setting 2	-32000 to 32000	378.4μV
		User range setting 3		312.0μV
	Current	0 to 20mA	0 to 32000	625.0nA
		4 to 20mA		500.0nA
		4 to 20mA (extended mode)	-8000 to 36000[7]	500.0nA
		User range setting 1	-32000 to 32000	360.1nA
Accuracy (accuracy for the maximum analog output value)[2]	Reference accuracy: ±0.1% (voltage: ±10mV, current: ±20μA) or less[3] Temperature coefficient: ±50ppm/℃ (0.005%/℃)[4]		
Conversion speed	1ms/CH		
Number of offset/gain settings[5]	50000 times maximum		
Output short circuit protection	Enabled		
Isolation method	Between the I/O terminal and programmable controller power supply: Transformer isolation Between analog output channels: Transformer isolation Between the external power supply and analog output channel: Transformer isolation		
Withstand voltage	Between the I/O terminal and programmable controller power supply: 500VACrms for 1 minute Between analog output channels: 1000VACrms for 1 minute Between the external power supply and analog output channel: 500VACrms for 1 minute		
Insulation resistance	Between the I/O terminal and programmable controller power supply: 10MΩ or more at 500VDC Between analog output channels: 10MΩ or more at 500VDC Between the external power supply and analog output channel: 10MΩ or more at 500VDC		
Number of occupied I/O points	• R60DA8-G: 16 points 1 slots (I/O assignment: intelligent 16 points) • R60DA16-G: 48 points 2 slots (I/O assignment: empty 16 points + intelligent 32 points)		
External interface	40-pin connectors		
Applicable wire size	When A6CON1 or A6CON4 is used	0.088 to 0.3㎟ (28 to 22 AWG) (stranded wire)	
	A6CON2	0.088 to 0.24㎟ (28 to 24 AWG) (stranded wire)	
Connectors for external devices	A6CON1, A6CON2, A6CON4 (sold separately)		
External power supply	24VDC +20%, -15%		
	Ripple, spike 500mV$_{P-P}$ or less		
	Inrush current • R60DA8-G: 4.2A, 540μs or less • R60DA16-G: 4.2A, 540μs or less for 24VDC_1, 24VDC_2		
	Current consumption • R60DA8-G: 0.36A • R60DA16-G: 0.70A		
Internal current consumption (5VDC)	• R60DA8-G: 0.18A • R60DA16-G: 0.25A		
External dimensions	Height	106mm (Base unit mounting side: 98mm)	
	Width	• R60DA8-G: 27.8mm • R60DA16-G: 56mm	
	Depth	110mm	
Weight	• R60DA8-G: 0.21kg • R60DA16-G: 0.32kg		

*1 For details on the I/O conversion characteristics, refer to the following.
 📖 MELSEC iQ-R Channel Isolated Digital-Analog Converter Module User's Manual (Startup)
*2 Cases when affected by noise are excluded.
*3 Accuracy in the ambient temperature when the offset/gain is set
*4 30-Minute warm-up operation (energization) is required to satisfy the accuracy.
 Accuracy per 1℃ temperature change
*5 If the number of settings exceeds 50000, the offset/gain setting maximum number of writes reach error (error code: 1080H) occurs.
*6 For an output current of 20mA or higher, the external load resistance value is as follows.
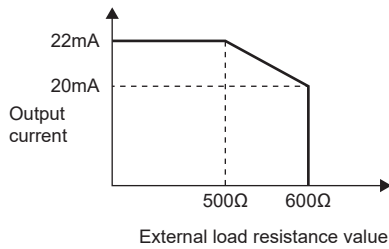
*7 The following table shows the resolution (16 bits and 32 bits) in extended mode.

Input range	16 bits		32 bits	
	Analog value	Digital value	Analog value	Digital value
1 to 5V (extended mode)	0 to 5.095V	-8000 to 32767	0 to 5.5V	-8000 to 36000
4 to 20mA (extended mode)	0 to 20.38mA	-8000 to 32767	0 to 22mA	-8000 to 36000

Precautions

Because the R60DA16-G occupies two slots, there are restrictions on the available firmware version of the RCPU. For details, refer to the following.

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

■Channel isolated analog-digital converter module

The following table describes the performance specifications of the A/D converter module.

Item	R60AD8-G, R60AD16-G			
Number of analog input channels	• R60AD8-G: 8 points (8 channels) • R60AD16-G: 16 points (16 channels)			
Analog input voltage	-10 to 10VDC (input resistance 1MΩ)			
Analog input current	0 to 20mADC (input resistance 250Ω)			
Digital output value	16-bit signed binary data (-32768 to 32767)			
I/O conversion characteristic, resolution[*1]	Analog input range		Digital output value	Resolution
	Voltage	0 to 10V	0 to 32000	312.5μV
		0 to 5V		156.3μV
		1 to 5V		125.0μV
		1 to 5V (extended mode)	-8000 to 32767 (-8000 to 36000)[*10]	125.0μV
		-10 to 10V	-32000 to 32000	312.5μV
		User range setting		29.2μV[*9]
	Current	0 to 20mA	0 to 32000	625.0nA
		4 to 20mA		500.0nA
		4 to 20mA (extended mode)	-8000 to 32767 (-8000 to 36000)[*10]	500.0nA
		User range setting	-32000 to 32000	115.5nA[*9]
Accuracy (accuracy for the maximum digital output value)[*2]	• Reference accuracy: Within ±0.1% (±32 digit)[*3] • Temperature coefficient: ±35ppm/℃ (0.0035%/℃)[*4]			
Common mode characteristic	Common mode voltage between input and common ground (input voltage 0V): 500VAC			
	Common mode voltage rejection ratio (VCM < 500V): 60Hz 107dB, 50Hz 106dB			
Conversion speed[*5]	10ms/CH			
Response time[*6]	20ms			
Absolute maximum input	Voltage: ±15V, Current: 30mA[*7]			
Number of offset/gain settings[*8]	50000 times maximum			
Isolation method	Between the I/O terminal and programmable controller power supply: Transformer isolation Between analog input channels: Transformer isolation			
Withstand voltage	Between the I/O terminal and programmable controller power supply: 500VACrms for 1 minute Between analog input channels: 1000VACrms for 1 minute			
Insulation resistance	Between the I/O terminal and programmable controller power supply: 10MΩ or more at 500VDC Between analog input channels: 10MΩ or more at 500VDC			
Number of occupied I/O points	• R60AD8-G: 16 points 1 slot (I/O assignment: intelligent 16 points) • R60AD16-G: 32 points 2 slots (I/O assignment: empty 16 points + intelligent 16 points)			
External interface	40-pin connectors			
Applicable wire size	When A6CON1 or A6CON4 is used	0.088 to 0.3㎟ (28 to 22 AWG) (stranded wire)		
	A6CON2	0.088 to 0.24㎟ (28 to 24 AWG) (stranded wire)		
Connectors for external devices	A6CON1, A6CON2, A6CON4 (sold separately)			
Internal current consumption (5VDC)	• R60AD8-G: 0.33A • R60AD16-G: 0.52A			
External dimensions	Height	106mm (Base unit mounting side: 98mm)		
	Width	• R60AD8-G: 27.8mm • R60AD16-G: 56mm		
	Depth	110mm		
Weight	• R60AD8-G: 0.19kg • R60AD16-G: 0.26kg			

*1 For details on the I/O conversion characteristics, refer to the following.
   📖 MELSEC iQ-R Channel Isolated Analog-Digital Converter Module User's Manual (Startup)
*2 Cases when affected by noise are excluded.
*3 Accuracy in the ambient temperature when the offset/gain is set
*4 Accuracy per 1℃ temperature change
*5 Cycle in which the digital output value is updated
*6 Time taken for an analog input signal to reach the AD converter in the module

*7 These voltage and current values are instantaneous values at which no breakdown occurs in the internal resistance of the module.
*8 If the number of settings exceeds 50000, the offset/gain setting maximum number of writes reach error (error code: 1080H) occurs.
*9 Maximum resolution for the user range setting
*10 Range of data stored in digital output value (32 bits)

Precautions

Because the R60AD16-G occupies two slots, there are restrictions on the available firmware version of the RCPU. For details, refer to the following.

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

1.4 Engineering Tool

GX Works3

GX Works3 manages programs and parameters for each CPU module in units of projects.

The main functions of GX Works3 are as follows.

Program creation function

The language relevant to the processing content can be used for programming.

[Ladder program]
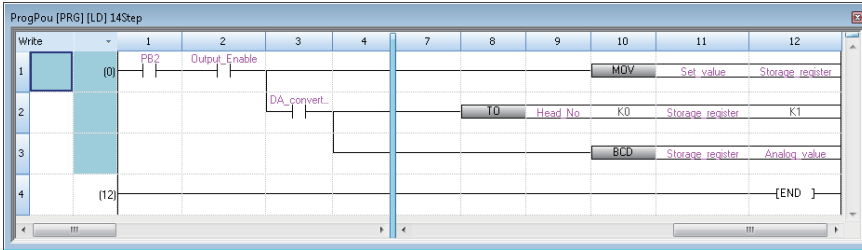
[ST program]

[FBD/LD program]
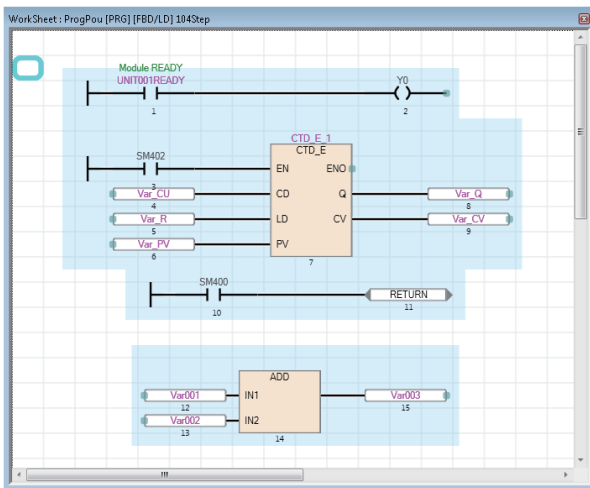
[SFC program]
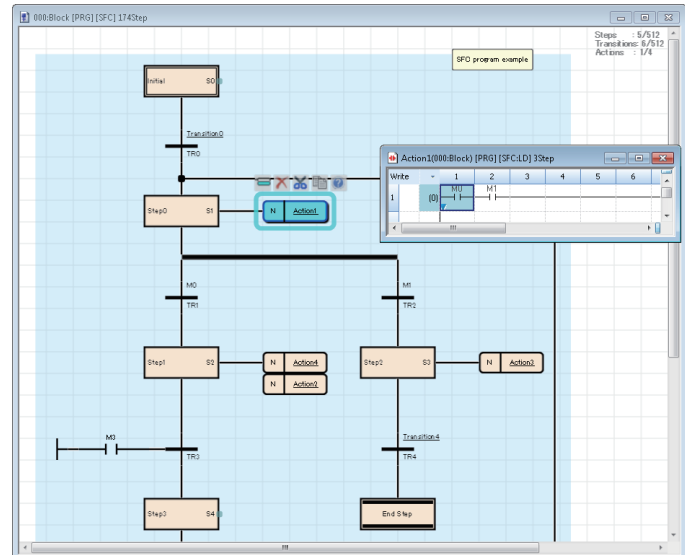
Parameter setting function

This function allows parameters of CPU modules, I/O modules, and intelligent function modules to be set.
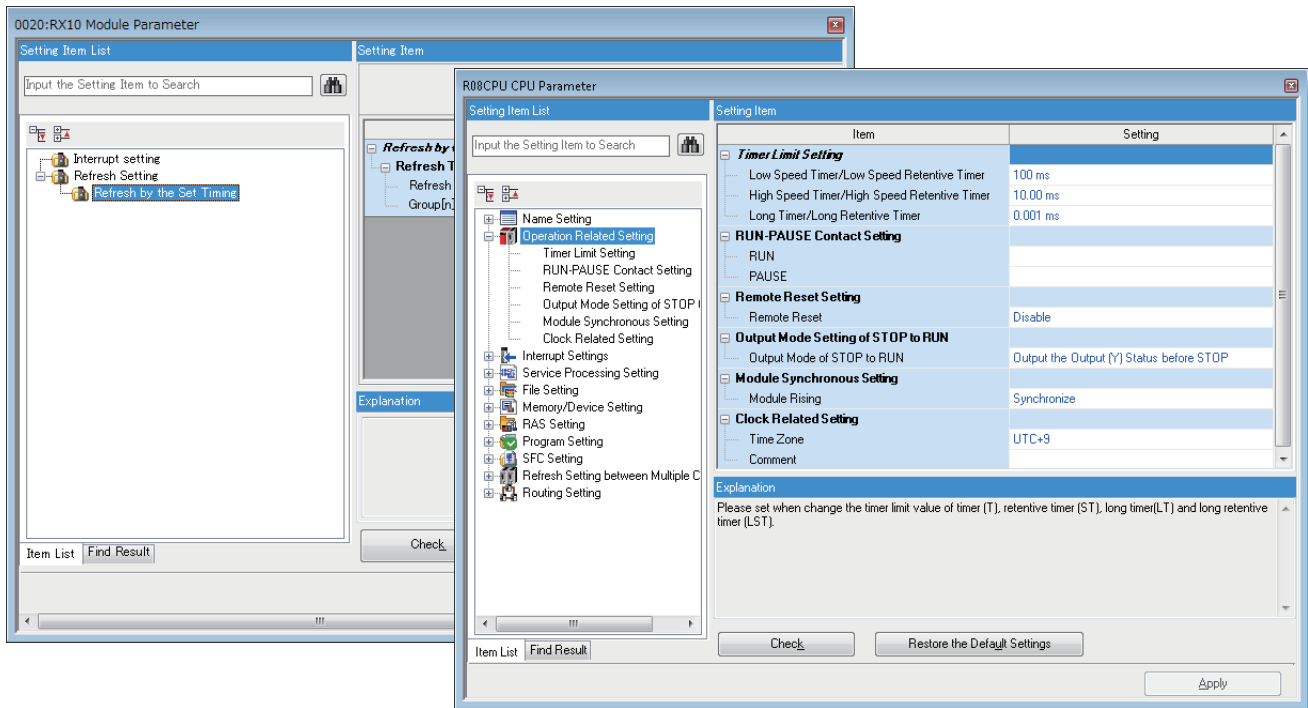
CPU module read/write function

With the "Read from PLC"/"Write to PLC" functions, a created sequence program can be read/written from/to the CPU module.

Also, with the online program change function, the sequence program can be changed even when the CPU module is in the RUN state.
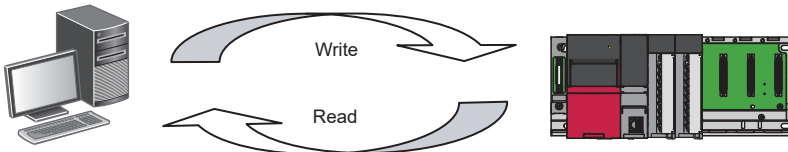
Monitor/debug function

This function monitors device values of when the created sequence program is written to the CPU module or is operated on the CPU module.

Even when the CPU module is not connected, a virtual programmable controller (simulation function) can be used to debug the program.

Shift + Enter
Turns on/off every time the above keys are pressed.
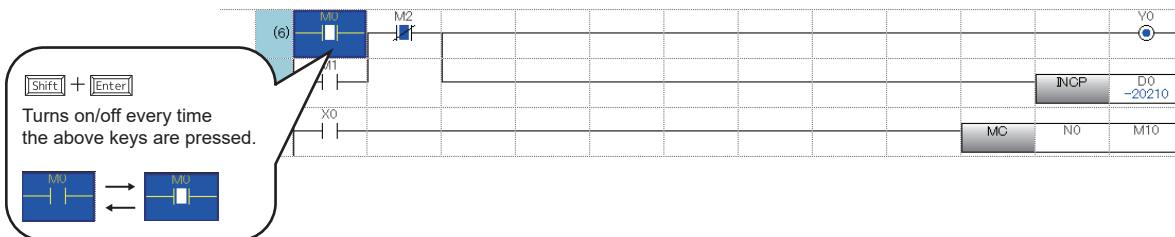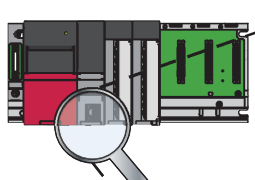
Diagnostic function

This function provides diagnostics for the current error status and error history of the CPU module and network. The function helps to shorten the recovery time.

Detailed information of intelligent function modules can also be checked via system monitoring. This feature further shortens the recovery time when an error occurs.

■CPU module diagnostics (module diagnostics window)

Process control system

Apart from the contents of this manual, refer to the following manuals.
📖 MELSEC iQ-R Programming Manual (Process Control Function Blocks/Instructions)
📖 GX Works3 Operating Manual

List of functions compatible with the process control system

GX Works3 provides functions (process control functions) for creating process control programs and adjusting and maintaining the process control system. The sequence control and process control can be integrated in a seamless manner by programming these controls with a single engineering tool.

Function	Description
Tag FB setting	Registers tag FB instances and tag data used in the FBD/LD program for process control at a time.
User-defined tag FB	Creates a function block in which tag data can be referenced. A user-defined tag FB can be created by pairing it with a function, function block, and process control function block.
FB property	Displays the initial label values of the VAR_PUBLIC class and VAR_PUBLIC_RETAIN class contained in the FB and tag FB. The initial values can be changed.
Faceplate	Displays the contents of tag data in a controller format. The value of each tag data can be changed. To use this function, the latest version of PX Developer must be installed.
FB property initial value update/management	Saves the current FB property values read from the programmable controller as the FB property initial values.
PX Developer monitor tool link	The PX Developer monitor tool can be used to adjust, monitor, and control the operation of the FBD/LD program for process control created in GX Works3.

Settings for using the process control function

The following settings need to be set to use the process control function. For other settings, use the same settings used for normal projects.

■Enabling the process control extension

Set the following to enable the process control extension for the program file.
- From the "Navigation" window, open the "Properties" window of the program file where the process control extension is to be enabled and set "Use the process control extension" to "Yes".
  ("Use the process control extension" can also be set to "Yes" by dragging and dropping the process control function block and tag FB from the "Element Selection" window.)

Process control extension can be enabled only for program files of the scan execution type and fixed scan execution type. For the scan execution type, only one program file with process control extension enabled can be created.
To divide a program with process control extension enabled, create multiple program blocks in the program file or create multiple worksheets in the program block.

■Setting process control extension options (required)

Check and set the following options.

Option	Description	Purpose of setting
[Tool] ⇨ [Options] ⇨ "Convert" ⇨ "Process Control Extension Setting"	Check and set each item according to the project to be created.	To use the process control function in the project to be created
[Tool] ⇨ [Options] ⇨ "Convert" ⇨ "Basic Setting" ⇨ "Conversion Operation" ⇨ "Enable Rebuild All (Retain)"	Specify it to "No".	If "Rebuild all" (retain) or "Convert" is enabled, labels added/changed after writing to the programmable controller cannot be initialized for the next writing to the programmable controller.[*1]
[Tool] ⇨ [Options] ⇨ "Convert" ⇨ "Basic Setting" ⇨ "Conversion Operation" ⇨ "Enable Conversion"		

*1 To initialize only the labels added/changed, execute "Convert" + "Online Program Change".

■Setting CPU parameters

Set the following items in the CPU parameter.

Parameter items	Description	Purpose of setting
"File Setting" ⇨ "File Register Setting" ⇨ "Use Or Not Setting"	Specify "Use Common File Register in All Programs".	The tag FB needs to use the file register.
"Memory/Device Setting" ⇨ "Device/Label Memory Area Setting" ⇨ "Device/Label Memory Area Detailed Setting" ⇨ "Device Setting"	Set ZR(R) in the [Latch (2)] tab of the "Latch Range Setting" window. *1,*2	The file register to be used in the tag FB needs to be latched.
"File Setting" ⇨ "Label Initial Value Reflection Setting" ⇨ "Label Initial Value Reflection Setting at STOP to RUN"	Specify it to "Disable".	To apply the value only once when the CPU module status changes from STOP to RUN after the timing of reflecting the label initial value to the current value is written to the programmable controller
"Memory/Device Setting" ⇨ "Index Register Setting" ⇨ "Points Setting" ⇨ "Total Points" ⇨ "Index Register (Z)"	Specify 7 points or more.	To use Z0 to Z6 for system data (system header and footer) for process control extension

*1 Click "Device Setting" ⇨ the [...] button of "Detailed Setting" ⇨ the [...] button of "Latch (2)" to display the "Latch Range Setting" window.
*2 The setting can also be applied by clicking the [Yes] button on the confirmation window displayed when setting the following parameter item.
   Click "File Setting" ⇨ "File Register Setting" ⇨ "Use Or Not Setting", and specify "Use Common File Register in All Programs".

■Setting the execution cycle

Set the execution cycle of a program with the process control extension enabled according to the processing.

Execution type	Description
Scan	Set the execution cycle on the "Properties" window of the program block.
Fixed scan	Set a fixed scan interval in the program setting of the CPU parameter.

■Process control extension toolbar

The following toolbar is displayed in a project with the process control function enabled.

■Multiple comment display setting
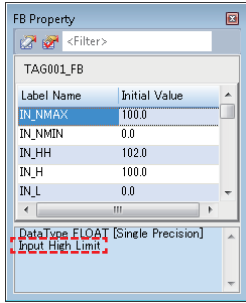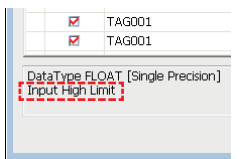
On the "Multiple Comments Display Setting" window[*1], set "Japanese" to "Target" to display tag FB labels and labels of the tag data structure members.

 *1 The "Multiple Comments Display Setting" window can be displayed by selecting the following menu items.
   [View] ⇨ [Multiple Comments] ⇨ [Display Setting]
The following table shows a display example.

Function	Display location	Example of display
FBD/LD editor	Edit box options	
	Label comment	
	Tool tip	

Function	Display location	Example of display
FB property	Description field of the "FB Property" window	
FB property initial value update/ management	Description field of the "Update the Initial Value of FB Property" window	
	Description field of the "FB Property Management" window	

Precautions

The following table lists considerations that need to be observed when the process control extension is enabled for the program.

Target project/ program	Item	Precautions
Program where the process control extension is enabled (including functions and function blocks used in that program)	Timer device	If the timer, retentive timer, and timer function block (TIMER_□_M) are used, proper measurement of time is impossible. For time measurement, use the long timer, long retentive timer, pulse timer (TP(_E)), on delay timer (TON(_E)), and off delay timer (TOF(_E)).
	Counter device and counter function block	If the count input signal of the counter, long counter, or counter function block turns on and off in a cycle shorter than the execution cycle, the rising edge of the count input signal cannot be detected and an intended value may not be obtained from the counter/long counter/counter function block. Set an execution cycle so that the count input signal turns on and off in a cycle longer than the execution cycle.
	Special relay	An intended value may not be obtained from special relay. The value obtained in the target program is applied.
Project containing a program where the process control extension is enabled	QDRSET(P) instruction	Do not use the QDRSET(P) instructions. If the file name of file register is changed by the QDRSET(P) instructions, programs do not operate correctly.
	Label class	Specify "VAR_RETAIN", "VAR_OUTPUT_RETAIN", or "VAR_PUBLIC_RETAIN" for the label class if there is a need to hold the label values of the process control program at power-off or reset of the programmable controller.
	Writing the global label setting (file register) to the programmable controller	Use the following procedure to write global labels to the programmable controller. ❶ Write the CPU parameters to the programmable controller. ❷ Power off and on or reset the programmable controller. ❸ Write the global label setting (file register)[*1].
	System resource (File register: ZR or R)	The range set in the option[*2] is used for assignment for the system area and tag data. Therefore, do not use the setting range of the file register in the program. However, it can be used for items of the public tag data. Refer to "Tag Data List" in the following manual. 📖 MELSEC iQ-R Programming Manual (Process Control Function Blocks/Instructions)
	System resource (Index register: Z)	The range (Z0 to Z6) displayed in the option[*3] is used for internal processing. Therefore, do not use Z0 to Z6 of the index register in the program where the process control extension is enabled, and in functions and function blocks used in that program.
Program where the process control extension is disabled	Function block	Do not use the following function blocks. Otherwise, the program may not work properly. • Process control function block • User-defined tag FB • User-defined FB that uses a process control function block or user-defined tag FB

*1   When the global label setting is written to the programmable controller, the data of file registers in the range set by the following procedure is written automatically.
[Tool] ⇨ [Options] ⇨ "Convert" ⇨ "Process Control Extension Setting" ⇨ "System Resource" ⇨ "File Register: ZR"
*2   [Tool] ⇨ [Options] ⇨ "Convert" ⇨ "Process Control Extension Setting" ⇨ "System Resource" ⇨ "File Register: ZR"
*3   [Tool] ⇨ [Options] ⇨ "Convert" ⇨ "Process Control Extension Setting" ⇨ "System Resource" ⇨ "index Register: Z"

Difference of terminology between GX Works3 and PX Developer

Some terminologies related to the process control function are different between PX Developer and GX Works3, both of which are the engineering tools for creating process control programs. The following table lists the PX Developer terms and corresponding GX Works3 terms.

PX Developer terminology	GX Works3 terminology
FB property current value read	FB property initial value update
General function, general FB	Standard function, standard function block
Global variable	Global label
Public variable	VAR_PUBLIC, VAR_PUBLIC_RETAIN
Cold start compile	Rebuild all (reassignment)
Compile (online change)	Convert + Online program change
Output variable	VAR_OUTPUT, VAR_OUTPUT_RETAIN
Tag FB variable	Tag FB instance and tag data
Constant	VAR_CONSTANT
Device variable	Device
Internal variable	VAR, VAR_RETAIN
I/O simulation setting	[Tool] from the GX Simulator3 window ⇨ [I/O System Setting]/[Start I/O System Setting]
Input variable	VAR_INPUT
Program executions setting	Execution type of program file
Project parameter	[Tool] ⇨ [Options] ⇨ "Convert" ⇨ "Process Control Extension Setting"
Process function, process FB	Process control function block
Variable registration monitor	Watch
Hot start compile	Rebuild all (retain), convert
Module FB	Module label and module FB
Local variable	Local label

PX Developer monitor tool

Overview

PX Developer is a software package for executing DDC processing on the CPU module.

PX Developer consists of the programming tool for programming with the FBD language and the monitor tool for monitoring DDC processing using tag data.

The monitor tool is a software tool that monitors and controls DDC processing being executed in the CPU module.

The monitor tool uses the programming tool to monitor and control DDC processing implemented in the form of tag FB.

The tag FB contains a data area called "tag data". This tag data is allocated to the programmable controller so that the status of the DDC processing logic is applied as needed when a program containing the corresponding tag FB is executed. The DDC processing logic itself is controlled by parameters contained in the tag data.

The monitor tool communicates with the CPU module to monitor and control this tag data. Various monitor screens display read tag data, and users can operate these screens to write values to tag data. Global variables and current device values can also be read and displayed.
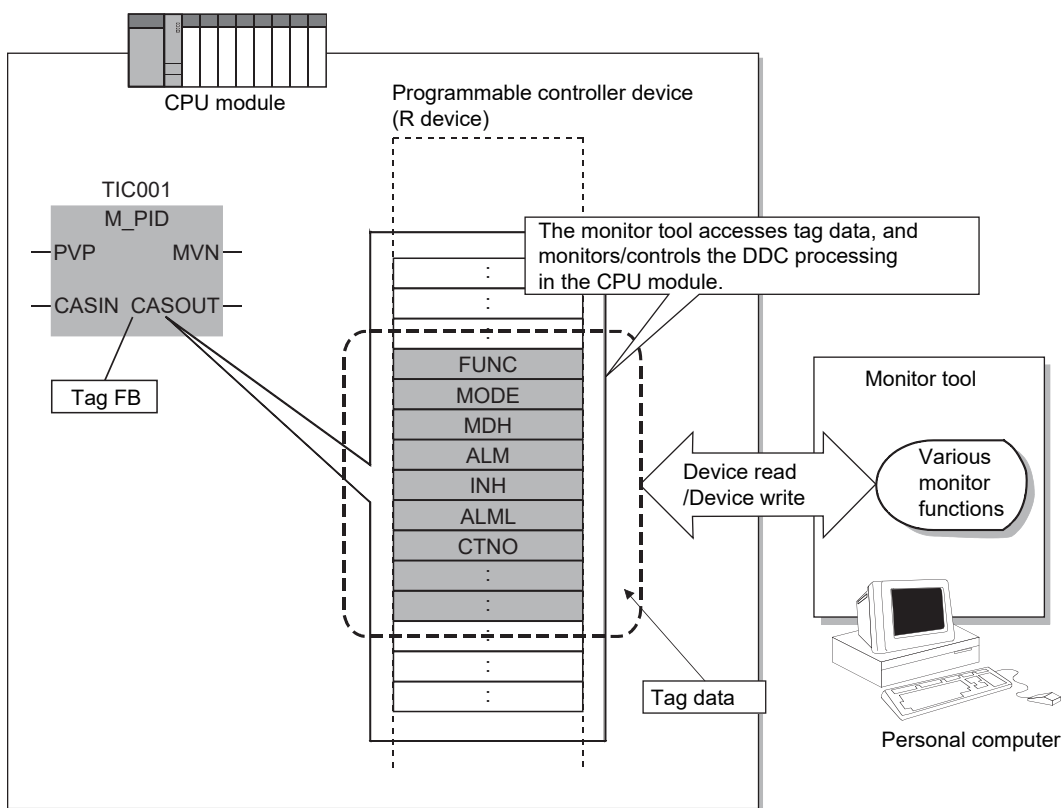
In addition, to allow users to create any user graphic screen (user monitoring screen) using Microsoft® Visual Basic®, this monitor tool provides the ActiveX controls for obtaining or setting any values in tag data and the ActiveX controls for displaying faceplates.

User graphic screens created with ActiveX controls can be easily opened via button operations on the monitor tool.

Features

■Equipped with the tag monitoring control function to reduce the time needed to operate the process control system

The monitoring tool provides various standard functions for tag monitoring control which are needed for start-up adjustment of the process control system. Only simple settings are required to monitor tags created with the programming tool, and there is no need to program communication processing for monitoring.

■Provides various methods to connect CPU modules

The monitor tool can simultaneously monitor and operate up to 16 CPU modules via the CC-Link IE Controller Network and Ethernet network.

Use the serial/USB connection for adjustment.

○: Available, ×: Not available

Communication path to the personal computer connection destination	Adjustment	Operation
Serial/USB	○	×
CC-Link IE Controller Network	○	○
Ethernet	○	○*1

　*1　Not available if the built-in Ethernet port of an Ethernet built-in CPU is directly connected

■Operation environment supporting single- and multi-window applications

Depending on their application, users can select the single-window mode to always display a single window at the maximum size or the multi-window mode to display multiple windows at the same time.

In multi-window mode, the monitor windows can be freely resized and organized with ease.

■User name and password management to prevent erroneous operations by unauthorized users

In the monitor tool, three types of modes can be set for each user authority.

Users whose user name and password are not registered (not having authority) cannot change the mode to edit tag data values and settings. When needed to leave the station, the person in charge can lock the mode to prevent unintended operations by unauthorized users.
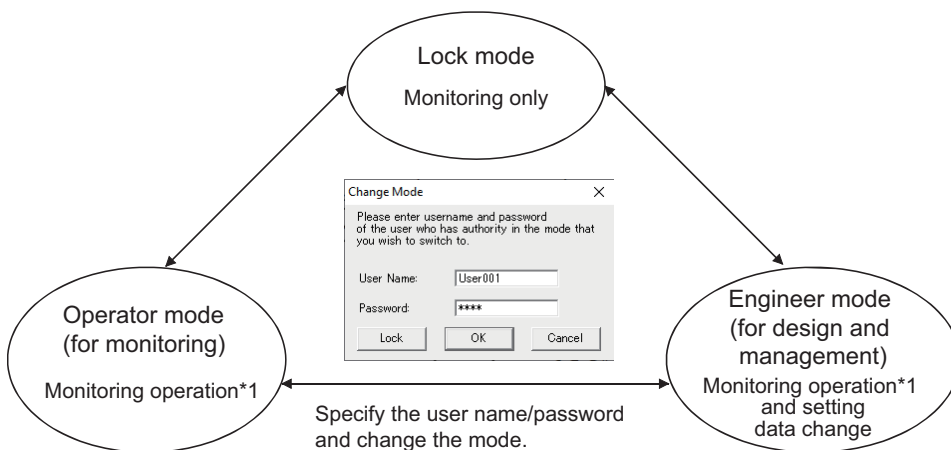
*1: The monitoring operation means monitoring/changing the tag data.

■**Link with GT SoftGOT**

Graphic screens created with GT Designer2 or GT Designer3 can be displayed by starting up GT SoftGOT from the monitor tool. Users can use existing GOT screens and create drawings using GT Designer2 and GT Designer3 whose operation they are familiar with.
The monitor tool functions such as the faceplate can be called from GT SoftGOT, reducing the time needed to create a screen.

■**Auto generation of GOT screen project**

By performing simple settings that include screen placement in the monitor tool and entering necessary information as prompted by the wizard, a GOT screen project is automatically created from the PX Developer project, reducing the time needed to create a monitor operation screen.

■**Supports Microsoft$^®$ Visual Basic$^®$ for creating user graphic screens**

The monitor tool provides the ActiveX control for obtaining or setting tag data values. This control can be used on a screen created with Microsoft$^®$ Visual Basic$^®$ to monitor tag data.

■**Redundant system**

This monitor tool can monitor and control a redundant system.
Auto-tracking is possible when system switching occurs.
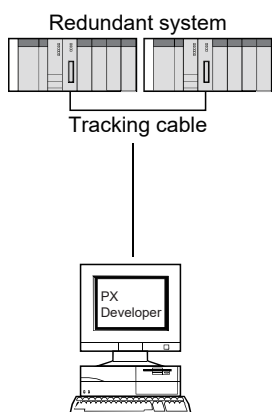
■**Monitoring with only the monitor tool**

PX Developer whose version is 1.06G or later is capable of monitoring as a standalone monitor tool even if the GX application is not installed on the computer.

■**Simulator connection**

PX Developer whose version is 1.28E or later supports offline debugging using a simulator. This function allows users to check the settings and simulate monitor operations without connecting to the CPU module.

■Support for server/client monitor system configuration

• Enhanced communication performance between the monitor tool and programmable controller

The server performs communication with programmable controllers in a batch, thereby reducing communication load on programmable controllers while increasing communication performance.

• Same monitor screen on all OPS monitor tools

Monitoring data collected and managed by the server is displayed on the client side, making it possible to display the same monitor screen for all OPS monitor tools.

• Continuous monitoring even if an error occurs on the server

Because the monitoring system consists of two servers (primary and secondary), a client can switch the server automatically to continue monitoring even if an error occurs on the primary server.

• Synchronized buzzer stoppage, warning/event check and deletion

If a buzzer is stopped or a warning/event is checked or deleted on any of the OPS in the monitoring system, the result is reflected to all OPSs in the monitoring system.
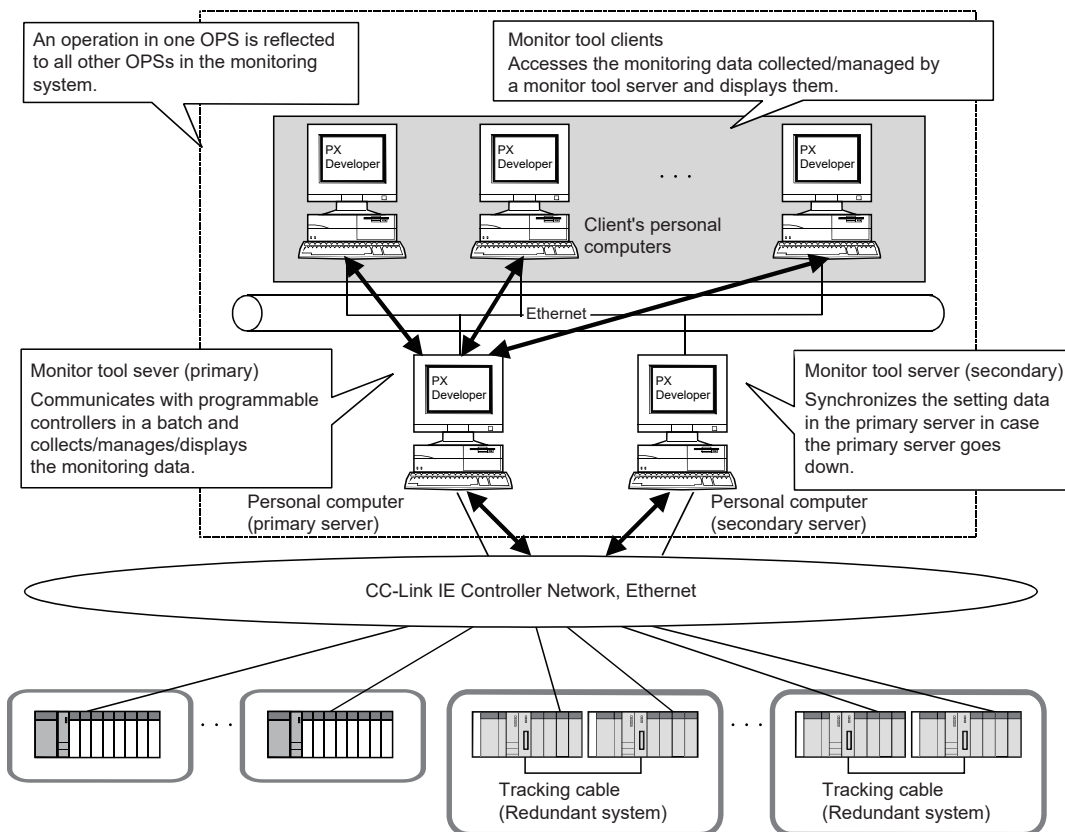
Point

• When a redundant server system is used, the settings can be changed on the primary server. The secondary server uses the setting data of the primary server to start up.

• Monitoring data is not equalized between the primary and secondary servers.
 Equalizing the monitoring data means synchronizing the server monitoring data such as trends and warnings between the primary and secondary servers.

2 CREATING FBD/LD PROGRAMS

This chapter describes how to create an FBD/LD program.

For details on the FBD/LD program specifications, refer to the following. Please read this beforehand.

📖 MELSEC iQ-R Programming Manual (Program Design)

The process control function blocks can be used in FBD/LD programs where the process control extension is enabled.

For details, refer to the following.

☞ Page 29 Settings for using the process control function

2.1 FBD/LD Editor Configuration

The FBD/LD editor is a graphical program editor for creating programs using the combination of the FBD language and ladder language.

A program can be created simply by freely placing available program elements vertically/horizontally and connecting them with lines.

Window

"Navigation" window ⇨ "Program" ⇨ "(Execution type)" ⇨ "(Program file)" ⇨ "(Program block)" ⇨ "(Work sheet)"

■ Toolbar

■ FBD/LD editor
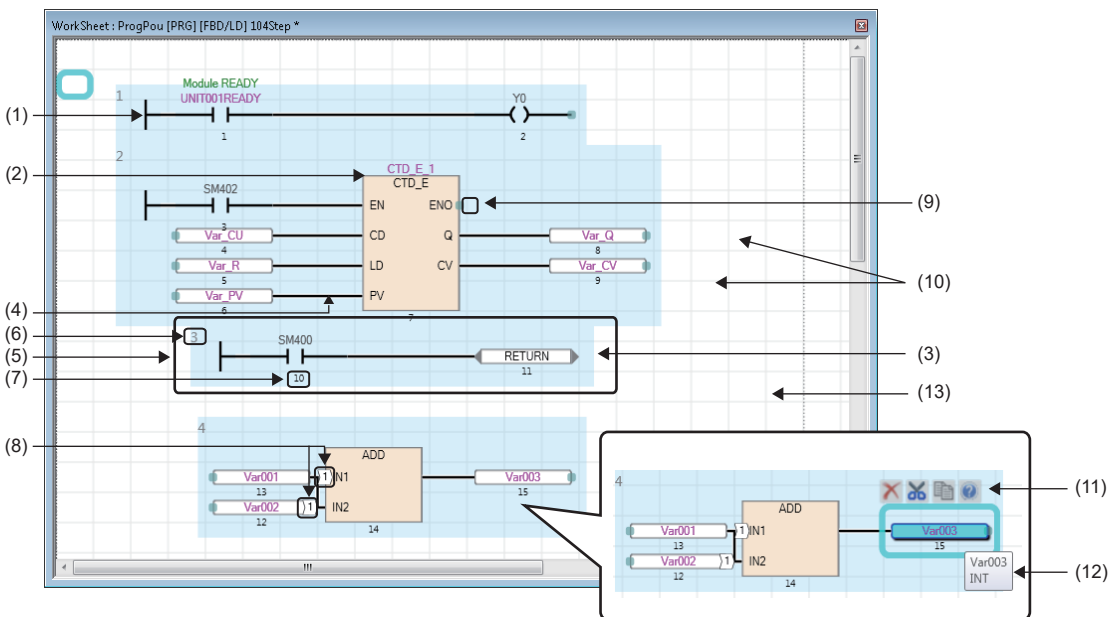
Displayed items

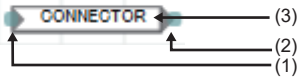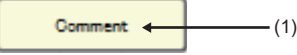| Item	Description	Edit operation
(1) LD element	A ladder program component element ⤷ Page 39 LD element	■How to edit: Page 42 Inputting a Program
(2) FBD element	An FBD program component element ⤷ Page 40 FBD element	
(3) Common element	Common program element irrelevant to program languages ⤷ Page 41 Common element	
(4) Connection line	A line connecting the connection points of program elements A connection line is displayed when a program element is moved close to the connection point of another program element.	—
(5) Ladder block	A ladder block is a group of all connected program elements or a program element (function block or jump label) that independently executes processing. Up to 4096 ladder blocks can be created within a program.	■Disconnection from the ladder block: Select a program element to be disconnected, and move the element while holding down ⌈Shift⌋. ■Layout adjustment: Page 46 Layout adjustment
(6) Ladder block No.[*1,*2]	Ladder blocks display their numbers that are assigned in order of left to right and top to bottom on the editor.	—
(7) Execute order[*1,*2]	This number indicates the execution order in the program.	—
(8) Automatic connector	Connectors are automatically displayed when the connection line cannot be displayed due to element placement. Automatic connectors with the same number indicate that they are connected.	—
(9) Connection point	An end point of a program element used to connect a line A program element can be added as connected by adding it with its connection point selected.	■Inversion: Page 42 How to switch contacts/instructions
(10) Grid[*1,*2]	A grid is displayed as a guide for placing program elements.	—
(11) Smart tag[*2]	Operation buttons displayed around the selected program element Each button executes the related function when clicked.	—
(12) Tool tip[*2]	Placing a mouse cursor over a program element displays information of the device/label and FB/FUN.	—
(13) Page break[*1]	This line indicates a page break for printing.	—

*1 Whether to display or hide this item can be set from the [View] menu.
*2 Click [Tool] ⇨ [Options] ⇨ "Program Editor" ⇨ "FBD/LD Editor", and set whether to display/hide this item.

Point
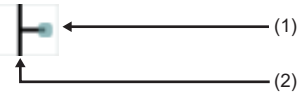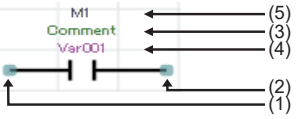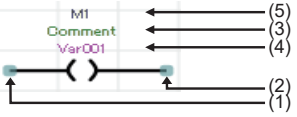
The font color, background color, and font can be changed.

Note that only the font color and background color can be changed in comment elements.

Program elements

The following describes program elements that can be used in FBD/LD programs.

■LD element
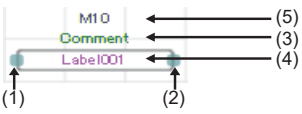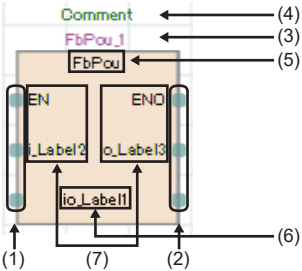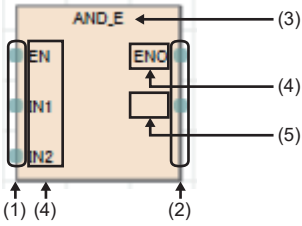
Program element	Part name	Description
Left rail element	(1) Output connection point (2) Left rail	The left rail can be positioned freely and used as the starting point for creating a ladder program.
Contact element	(1) Input connection point (2) Output connection point (3) Label comment/device comment[*1][*2][*3] (4) Device/label[*3] (5) Assigned device[*1,*2] (only global labels to which devices are assigned)	Specify the device/label. ON/OFF signals are transmitted according to the specified information.
Coil element		Specify the device/label. Information is output to the specified device/label according to received ON/OFF signal.

*1 Whether to display or hide this item can be set from the [View] menu.
*2 Click [Tool] ⇨ [Options] ⇨ "Program Editor" ⇨ "FBD/LD Editor", and set whether to display/hide this item.
*3 Click [Tool] ⇨ [Options] ⇨ "Program Editor" ⇨ "FBD/LD Editor", and set whether to wrap the comment.

■FBD element

Program element	Part name	Description
Variable element 	(1) Input connection point (2) Output connection point (3) Label comment/device comment[1][2][3] (4) Device/label (5) Assigned device[1,2] (only global labels to which devices are assigned)	Specify the device/label. Information can be acquired from or set to the specified device/label. If a constant is input, the variable element changes to a constant element.
Constant element 	(1) Output connection point (2) Constant value	Specify a constant. The specified constant can be output. If a device or label is input, the constant element changes to a variable element.
Function block element 	(1) Input connection point (2) Output connection point (3) FB instance name (label) (4) Label comment[1][2][3] (5) Data type (6) I/O label (VAR_IN_OUT) (7) I/O labels (other than VAR_IN_OUT)	Indicates a function block corresponding to the data type. Assign an FB instance name to each program element before use.
Function element 	(1) Input connection point (2) Output connection point (3) Data type (4) I/O label (argument) (5) Return value	Indicates a function block corresponding to the data type. No name is displayed for the return value.
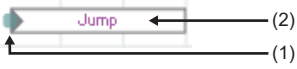
*1 Whether to display or hide this item can be set from the [View] menu.
*2 Click [Tool] ⇨ [Options] ⇨ "Program Editor" ⇨ "FBD/LD Editor", and set whether to display/hide this item.
*3 Click [Tool] ⇨ [Options] ⇨ "Program Editor" ⇨ "FBD/LD Editor", and set whether to wrap the comment.

■Common element

A jump element/return element cannot be inverted on a connection point.

Program element	Part name	Description
Jump element Jump ◄—— (2) ◄—— (1)	(1) Input connection point (2) Label[*1]	Used to skip the execution processing from the jump element to the jump label element.
Jump label element Jump ◄—— (1)	(1) Label[*1]	Used to specify the label to which to jump.
Connector element CONNECTOR ◄—— (3) ◄—— (2) ◄—— (1)	(1) Input connection point (2) Output connection point (3) Connector label	Used instead of a connection line when a circuit block is placed within the display/print area of the editor. It is connected to the connector label with the same name.
Return element RETURN ◄—— (2) ◄—— (1)	(1) Input connection point (2) "RETURN" string (cannot be edited)	Used to break the processing.
Comment element Comment ◄—— (1)	(1) Comments display area	Used to input a comment. Up to 2000 characters can be input in a comment element. Place the mouse cursor over the element and double-click on it to automatically adjust the element size according to the character string.

*1 Only local labels whose "Data type" is set to "Pointer" can be used. Note that structure members cannot be used.

Point

The smart tag is available to individually change the color setting of a comment element.
- Click the smart tag button (1) to set the font color.
- Click the smart tag button (2) to set the background color.
- Click the smart tag button (3) to clear the individually set color setting.
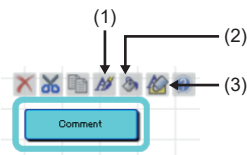
(1): Font color
(2): Background color
(3): Color setting clear

Precautions

If the following settings are changed from the menu or in the option settings, display of the label name, device, comment, and assigned device for the element may overlap with other elements. In such case, adjust the position of the element.
- Whether to display the device/label comment
- Number of lines for the device/label comment
- Whether to display the assigned device
- Number of lines for the wrapped device/label name

2.2 Inputting a Program

This section describes how to input FBD/LD programs.

Adding a program element

Operating procedure

■Input from the edit box

Select a cell to be added[*1], and directly enter a label name and FB/FUN data type.

Or, select a placed program element, press ⎣F2⎦ and directly edit the element.

The following items can be input.

- Device
- Label/assigned device
- Constant
- FB/FUN

*1 Press ⎣Ctrl⎦ + ⎣Space⎦ to display auto-suggested instructions/labels.

■Inserting from the menu/toolbar

Select a cell to be added on the FBD/LD editor, and select [Edit] ⇨ [Add Element (Ladder Symbol)] ⇨ [(Program element)].

Or, select the same from the toolbar.

■Inserting from the "Element Selection" window

Select a program element from the "Element Selection" window, and drag and drop it to the FBD/LD editor.

■How to switch contacts/instructions

Select a contact/instruction to be changed, and select [Edit] ⇨ [Easy Edit] ⇨ [Invert Contact (Open/Close)]/[Switch Pulse]/
[Switch SET and RST]. Or, use the following procedure for switching.

Switching contacts and instructions	Short cut keys	Remarks
Open/closed contact switching, rising/ falling pulse switching		—
Inverse, SET/RST instruction switching		—
FB/FUN connection point inverse switching		This switching is available only for the following connection point data types. • Bit • Word [unsigned]/bit string [16 bits] • Double word [unsigned]/bit string [32 bits] • ANY_BIT • ANY_BOOL

■Displaying the global label by inputting the assigned device

By inputting the device assigned to the program element, the global label of that device can be displayed on the program element.

Note that only global labels with primitive data types (not an array) can be displayed.

Change the input method by using either of the following procedures and then input the assigned device. The global label is displayed when the input is complete.

- [Edit] ⇨ [Edit Mode] ⇨ [Use Assigned Device for Label Input]
- [Tool] ⇨ [Options] ⇨ "Program Editor" ⇨ "FBD/LD Editor ⇨ "Use assigned device for label input"

■How to specify the device data type

On the FBD/LD editor, the data type of a word device can be specified.

To specify a data type, append a suffix indicating a data type to a device name.

For details, refer to the following.

📖 MELSEC iQ-R Programming Manual (Program Design)

📖 MELSEC iQ-F FX5 Programming Manual (Program Design)

Common operations for program elements

Operation		Operating procedure	
		Mouse	**Keyboard**
Select	Selecting a single program element	Click the program element to be selected.	Move the cursor over the program element to be selected.
	Selecting multiple program elements	• While pressing [Ctrl]/[Shift], click multiple program elements. • Click the background of the FBD/LD editor, and drag to enclose the elements to be selected.	While pressing [Shift], move the cursor and select multiple program elements.
	Selecting in units of ladder block[*1]	Click a program element, and select [Edit] ⇨ [Select FBD Network Block].	Select a program element, and press [Ctrl]+[Shift]+[A].
	Selecting all program elements	The procedure is the same as that for selecting multiple program elements.	Press [Ctrl]+[A].
Rename		Double-click a desired program element and rename it.	Select a program element to be renamed, and press [Enter][*2]/[F2], or enter a name.
Move		Drag and drop a program element. (Moving a program element while holding down [Ctrl] allows it to be moved being disconnected from the ladder block. The disconnected element will automatically be connected[*3] to the previously connected element again after it is moved.)	Select a program element to be moved, and press [Ctrl]+[Shift]+[←]/[→]/[↑]/[↓].
Copy		Drag and drop a program element to be copied while pressing [Ctrl].	Press [Ctrl]+[C], select a copy destination, and press [Ctrl]+[V]. (Copy and paste)
Delete		—	Select a program element to be deleted, and press [Delete]. (After the deletion, the previously connected program elements are connected automatically[*3].)
Connection line	Connection	Click a connection point, and drag it to the target connection point.	—
	Automatic connection[*3]	Click a program element to be connected, and drag it close to the target connection point.	—
	Changing the connection	Select the connection line while pressing [Shift], and drag and drop it to the connection point of the target program element.	—
	Inserting a program element	Click a program element and move it to a connection line while pressing [Shift]. (Only for program elements having connection points on the input and output sides at the same level.)	—

*1 After selecting multiple program elements, perform the operation that allows selection in units of ladder block to allow multiple ladder blocks to be selected. Program elements that are not connected to other program elements will be deselected.

*2 For function elements and function block elements, the operation resulting from the following procedure is applied: [Tool] ⇨ [Options] ⇨ "Program Editor" ⇨ double-click "FBD/LD Editor".

*3 This function automatically connects program elements with a connection line. The function can be enabled/disabled in [Edit] ⇨ [Edit Mode] ⇨ [Element Auto-connect].

Replacing function elements/function block elements

Operating procedure

1. Select a function element/function block element.

2. Select [Edit] ⇨ [Change FB/FUN Data], and enter a data type to be applied.

Point

An element can also be replaced by dragging and dropping it from the "Element Selection" window to the target element. Also, the data type of a function block element can be changed by selecting a function block element and directly entering a data type.

Adding/deleting arguments

Arguments can be changed/deleted only for functions where the number of arguments can be changed.

Operating procedure

1. Move the cursor over a function element.

2. Select [Edit] ⇨ [I/O Argument] ⇨ [Increment Argument]/[Delete Argument].

FB/FUN whose definition is unknown

If the definition of a pasted element is unknown because it was deleted or changed, the element is displayed in error state as shown in the following figure.
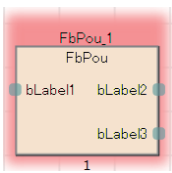
If the definition has been changed, select the element, and select [Edit] ⇨ [Update FB/FUN] to update the definition information.

If no definition exists, select the element, and select [Edit] ⇨ [Change FB/FUN Data] to update the data.

Automatic adjustment

When a program element is added or moved, its position is automatically adjusted so that it is not overlapped with other elements.

If the ladder block rectangular area (increased by one cell in lateral and longitudinal directions from the area of the element connected by a connection line) is overlapped with another area, the element position is adjusted automatically so that the entire ladder block is not overlapped.

Only the comment elements are not repositioned and can be overlapped.

Stacking order of comment elements

When a comment element is placed over another comment element, the comment later added is displayed in front.

The stacking order of comment elements can be changed by selecting [Edit] ⇨ [Order Comment] ⇨ [Bring to Front]/[Bring Forward]/[Send Backward]/[Send to Back].

Comment elements are in unconverted state after the stacking order is changed.

Layout adjustment

Operating procedure

■Inserting a row
Select a cell where a row is to be inserted and select [Edit] ⇨ [Layout] ⇨ [Insert Row].

A row is inserted above the selected cell.

■Deleting a row
Select a cell of a row to be deleted, and select [Edit] ⇨ [Layout] ⇨ [Delete Row].

The row of the selected cell is deleted. Note that the row cannot be deleted if another element exists on that row of the selected cell.

■Inserting/deleting a column
Move the cursor inside a ladder block where a column is to be inserted/deleted, and select [Edit] ⇨ [Layout] ⇨ [Insert Column (in FBD Network Block)]/[Delete Column (in FBD Network Block)].

A column is inserted into or deleted from the ladder block.

■Inserting multiple rows
Select a cell where multiple rows are to be inserted, and select [Edit] ⇨ [Layout] ⇨ [Insert Multiple Rows].

On the "Insert Multiple Rows" window, set the number of rows to be inserted.

A specified number of rows are inserted above the selected cell.

■Deleting multiple rows
Select a cell of a row to be deleted, and select [Edit] ⇨ [Layout] ⇨ [Delete Multiple Rows].

On the "Delete Multiple Rows" window, set the number of rows to be deleted.

The specified number of rows under the selected cell are deleted.

If an element exists in any of the rows to be deleted, rows up to the row containing the element are deleted.

■Deleting blank rows between ladder blocks
Select [Edit] ⇨ [Layout] ⇨ [Delete the Blank Row Between FBD Network Blocks].

Rows between ladder blocks and elements are deleted.

■Deleting blank columns inside ladder blocks
Select [Edit] ⇨ [Layout] ⇨ [Layout Correction in FBD Network Block].

Columns not containing elements are deleted from the ladder block.

■Ladder block left alignment
Select [Edit] ⇨ [Layout] ⇨ [Batch Alignment of All FBD Network Blocks to the Left].

All ladder blocks are left-aligned.

If multiple ladder blocks exist on the same row, the blank columns between these ladder blocks are maintained.

Pasting from various windows

Label/device names can be pasted by dragging and dropping from the label editor or device comment editor.

Registering undefined labels

When an undefined label is input, the "Undefined Label Registration" window is displayed from which the undefined label can be registered to the label editor.

Reuse of tag FBs

This section describes how to reuse tag FBs for an FBD/LD program for process control.

The relevant tag FBs need to be registered in advance using the tag FB editor.

Operating procedure

1. Select a tag FB instance on the "Element Selection" window and drag and drop it to the FBD/LD editor.

2. Connect input and output variables to the tag FB instance.
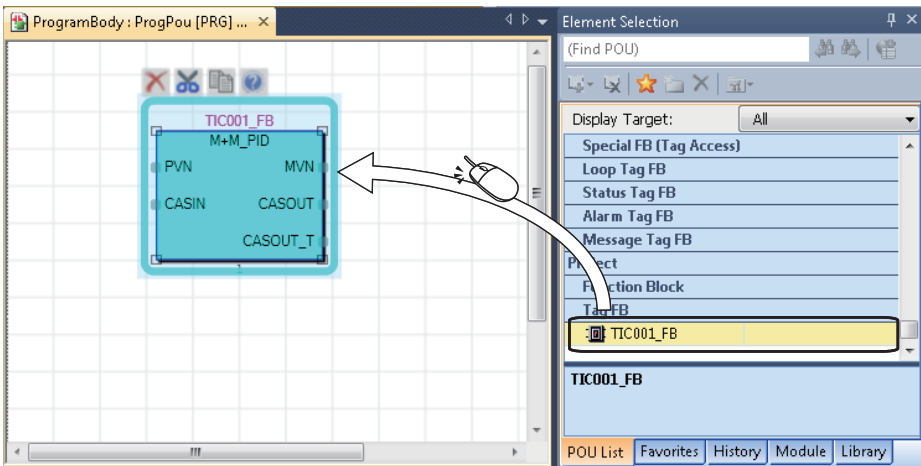
3 PROCESS CONTROL FUNCTION BLOCK

A process control function block is a function block whose function is extended for process control. It contains structure data including process conditions and process status (tag data).

A process control function block can be used with the MELSEC iQ-R series Process CPU and SIL2 Process CPU (standard program only).

Ex.

Program using the tag FB (M+M_2PIDH_) which optimizes responsive performance for a setting value and control performance to a disturbance
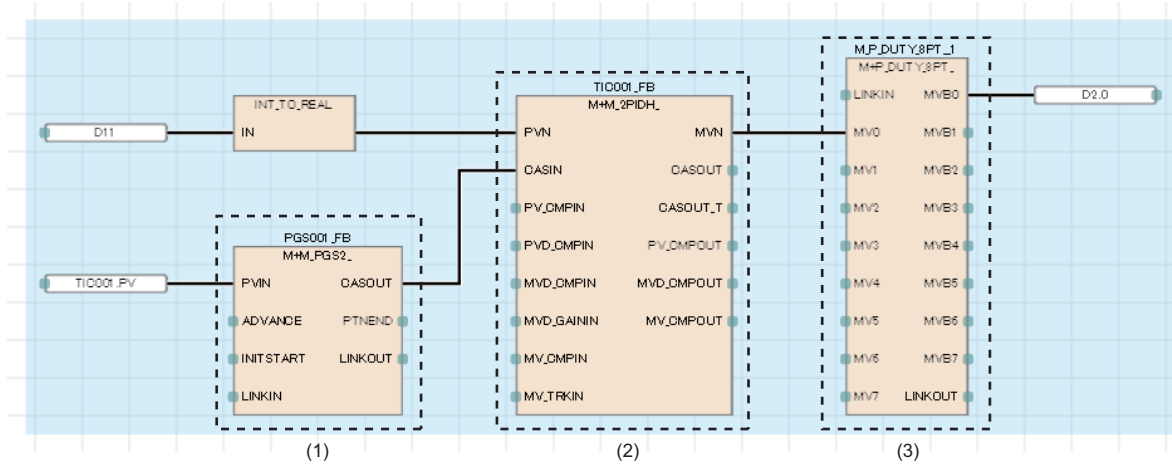
(1) Tag FB of multi-point program setter (M+M_PGS2_)
(2) Tag FB of two-degree-of-freedom advanced PID control (M+M_2PIDH_)
(3) General process FB of 8 points time proportional output (M+P_DUTY_8PT_)

3.1 Process Control Function Block Types

The process control function block has the following types.

Item	Description
General process FB	Performs process control operation, such as correction operation and control operation.
Tag access FB	Performs processing by accessing tag data of a user-defined tag FB. It can be used only in the user-defined tag FB.
Tag FB	Performs process control operation as a controller or an indicator. It performs processing by accessing tag data defined as global labels. The execution status can be checked and controlled on the faceplate of the engineering tool.
User-defined tag FB	Combines tag access FBs, standard functions, or standard function blocks to extend process control processing of a tag FB. It performs processing by accessing tag data in the same way as a tag FB. The execution status can be checked and controlled on the faceplate of the engineering tool.

3.2 Program Supporting Process Control Function Blocks

A process control function block can be used in an FBD/LD program for process control.

An FBD/LD program for process control can be created when the process control extension is enabled on the "Properties" window of the program file.

The scan execution type or fixed scan execution type can be specified as an execution type of the FBD/LD program for process control. For the scan execution type, only one program file with process control extension enabled can be created. The following table shows the availability of process control function blocks.

Item			Called function block					
			Local label			Global label		
			General process FB	Tag access FB	Tag FB, user-defined tag FB	General process FB	Tag access FB	Tag FB, user-defined tag FB
Calling source	Program	FBD/LD (process function extension enabled)	○	×	×	○	×	○*2
		FBD/LD (process function extension disabled)	×	×	×	×	×	×
		Other than FBD/LD*3	×	×	×	×	×	×
	Function block	FBD/LD (user-defined tag FB)	○	○*1	×	○	×	○*2
		FBD/LD (other than user-defined tag FB)*4	○	×	×	○	×	○*2
		Other than FBD/LD*3	×	×	×	×	×	×
	Function		×	×	×	×	×	×

*1 The tag access FB can be used only in the user-defined tag FB.

*2 The tag FB and the user-defined tag FB can be declared only as a global label. Declare them in the "Tag FB Setting" window of the engineering tool.

*3 Actions and transitions in an SFC program are included.

*4 Do not use a process control function block or a user-defined FB uses a user-defined tag FB in programs with the process control extension disabled. Doing so causes an error in the programs.

3.3 Tag FBs

Tag FBs and user-defined tag FBs have functions, such as a controller or a indicator, for process control. They perform processing by accessing tag data defined as global labels.

The tag data defines data relevant to the process control system as a structure. The operating status of a tag FB can be checked and controlled by accessing the tag data from the faceplate of an engineering tool.
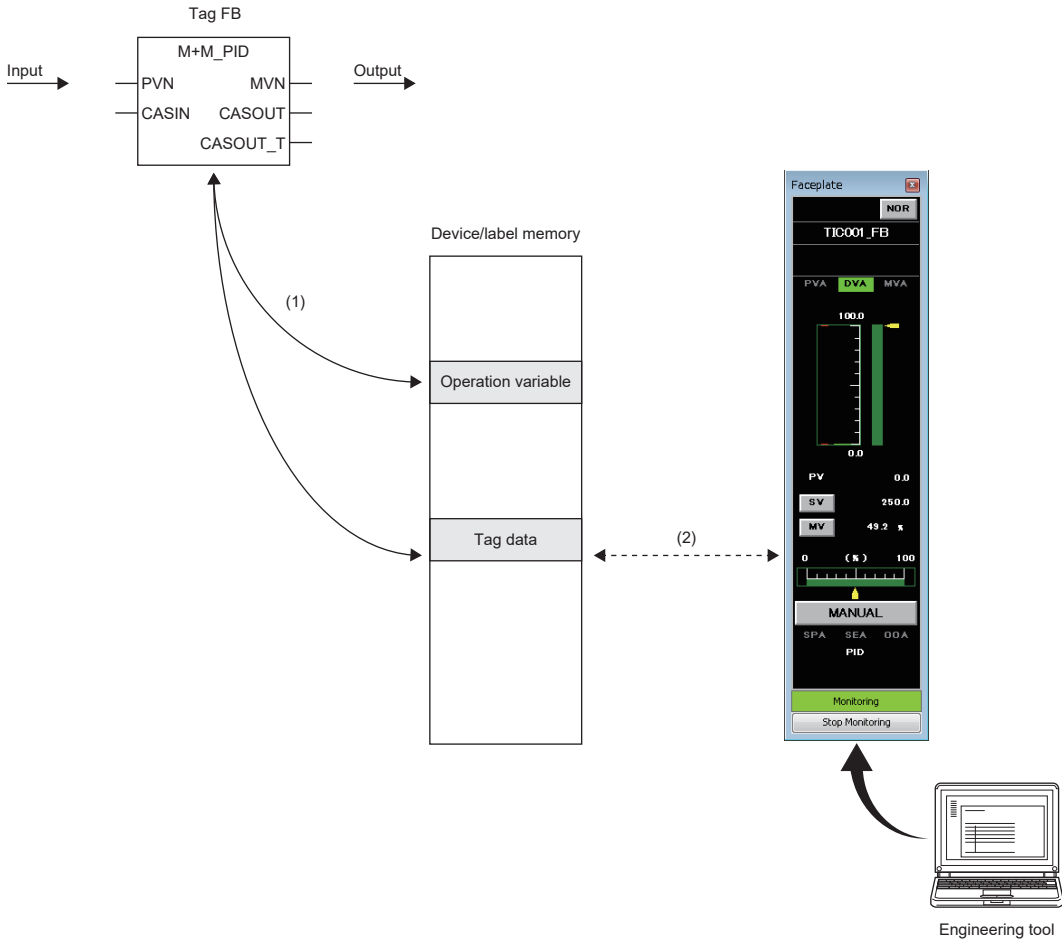
(1) The tag FB performs processing by accessing the tag data or operation constants (public variables of the FB).
(2) The execution status of the tag FB can be checked and controlled on the faceplate of the engineering tool.

Tag FB categories

The tag FB is classified into four categories according to the tag type. The following table lists the tag type classification.

Classification	Description
Loop tag	Used for loop control processing. This is equivalent to the loop tag used in process control instructions.
Status tag	Used for monitoring or control of the on/off states.
Alarm tag	Used for alarm notification.
Message tag	Used for message notification.

The tag data structure and faceplate type depend on the tag type of tag FB.

Ex.

Tag type (PID) and faceplate for the tag FB (M+M_PID)
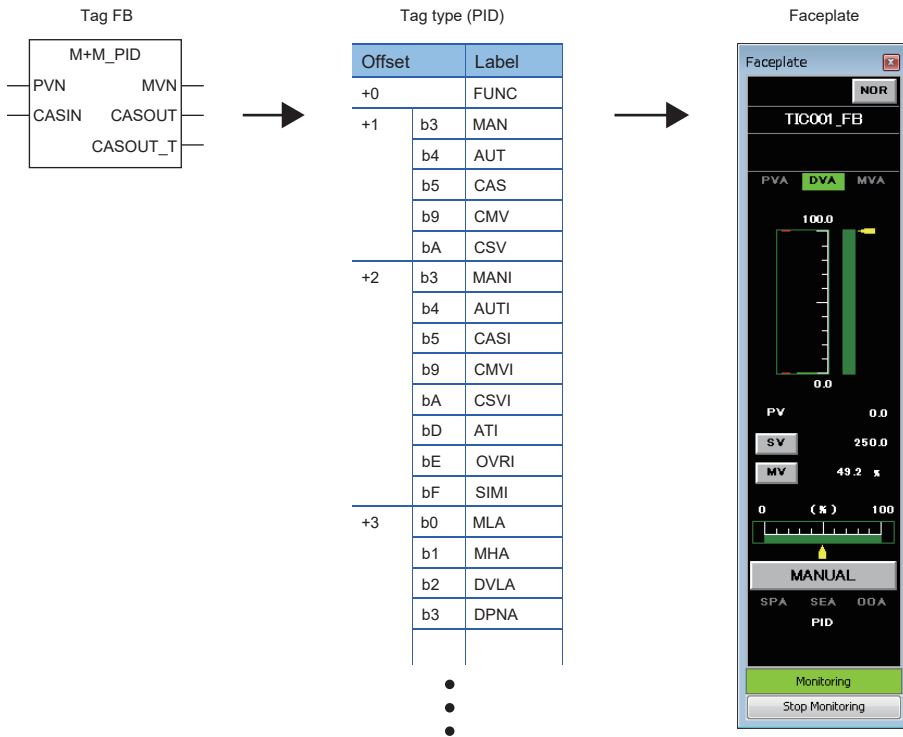
Initial values for tag data and operation constants (public variables)

The tag FB requires the initial value settings for tag data and operation constants (public variables).

The initial values for tag data and public variables of each tag FB part can be set on the "FB Property" window of the engineering tool.

Control mode

The tag FB has six control modes.

Switching the control modes switches the tag FB control to the manual operation, automatic operation, or cascade operation.

The following table lists the control modes.

Symbol	Control mode	Description
MAN	MANUAL	Performs the manual operation. The MV setting value is output.
AUT	AUTO	Performs the automatic operation. The manipulated value is controlled on the basis of the SV setting value.
CAS	CASCADE	Performs the cascade operation. The output value (MV) of primary loop is controlled as the setting value (SV).
CMV	COMPUTER MV	Performs the manual operation with the host computer. The MV setting value of the host computer is output.
CSV	COMPUTER SV	Performs the automatic operation with the host computer. The manipulated value is controlled on the basis of the SV setting value of the host computer.
CASDR	CASCADE DIRECT	Directly outputs the output value of primary loop as that of secondary loop in the cascade connection.

The control mode can be switched with the faceplate of tag FB or M+P_MCHG.

Available control modes depend on the tag type.

Switching the control mode

■When the tag type is other than 2PIDH
There are no restrictions on the control mode switching.

■When the tag type is 2PIDH
A tag FB can go into CASCADE DIRECT(CASDR) mode only from CASCADE(CAS) mode. There are no restrictions on other control mode switching.
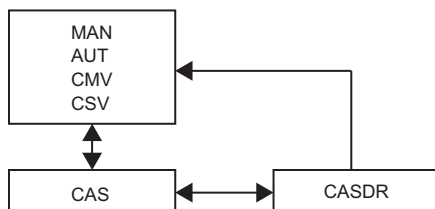
Point

When the stop alarm (SPA) is TRUE, the control mode is automatically and forcibly switched to the MANUAL mode.

I/O mode

Inputs and outputs of the tag FB is connected or disconnected with the I/O module in accordance with the I/O mode setting.
Also, the input processing and loop control operation of the tag FB can be stopped.

The tag FB has four I/O modes as follows.

Symbol	I/O mode	Description
NOR	NORMAL	Connects the signals from the I/O module (normal mode).
SIM	SIMULATION	Disconnects the signals from the I/O module and performs simulations.
OVR	OVERRIDE	Disconnects signals from the input module and enables the input of the process variable (PV) on the faceplate. This mode is used in case of the input sensor failure.
TSTP	TAG STOP	Performs no processing related to the tag. The input processing and loop control operation are stopped. This mode is set for the tag which has been defined for future use or has stopped. All alarms related to the tag are reset, and no unnecessary alarm will occur.

Available I/O modes depend on the tag type.

Switching the I/O mode

The I/O mode can be switched when the control mode is MANUAL. Use the faceplate for switching. (📖 PX Developer Version 1 Operating Manual (Monitor Tool))

3.4 Program Execution Control

A program execution cycle can be set for an FBD/LD program for process control.

There are two methods for executing a program: timer execution and interrupt execution (fixed scan). Select either of the methods in accordance with the intended use of the program.

Timer execution

An FBD/LD program for process control is executed in the execution cycle (ΔT) which is set to each program.

Among tag access FBs that are used in a loop tag FB, M+P_IN, M+P_PHPL, and M+P_OUT1 for I/O control are executed every execution cycle. Additionally, M+P_PID and M+P_2PID used in loop control operation are executed in the control cycle (CT) which is set to each tag. (☞ Page 59 Control cycle (CT))

Setting the execution cycle (ΔT)

Select "High-Speed", "Normal-Speed", or "Low-Speed" for an execution cycle of each program block.

Set the execution cycle on the "Properties" window of the program block.

⊟ Inherent Property	
Execution Interval	Normal-Speed(1000ms) ▾
Phase	1
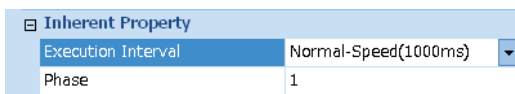
Changing the execution cycle (ΔT)

The following table shows the setting values and details on the execution cycle.

Execution cycle	Description	Default
High-Speed	"50ms", "100ms", or "200ms" can be selected for the high-speed execution cycle.	200ms
Normal-Speed	The normal-speed execution cycle is determined as follows: High-speed execution cycle × Setting value. "2", "3", "4", or "5" can be selected for the setting value.	1000ms (Setting value = 5)
Low-Speed	The low-speed execution cycle is determined as follows: High-speed execution cycle × Setting value. "5", "10", "20", "25", or "50" can be selected for the setting value.	4000ms (Setting value = 20)

The high-speed execution cycle and the setting values for normal/low-speed execution cycle can be set on the "Options" window of the engineering tool.

🗐 Convert	⊟ Program Execution	
Basic Setting	⊟ Execution Interval	
Online Program Change	High-Speed	200ms ▾
Output Result	Normal-Speed (High-Speed x Setting Value)	5 ▾
Process Control Extension Setting	Low-Speed (High-Speed x Setting Value)	20 ▾
	⊟ I/O Control	
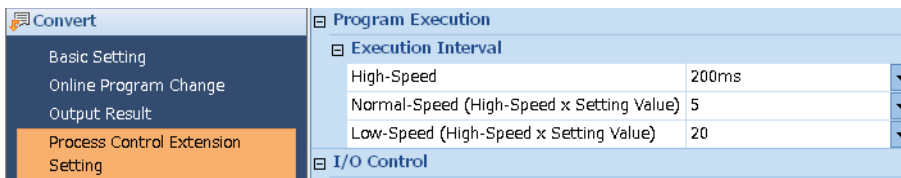
■Precautions

Set the scan time so that it is equal to or shorter than the high-speed execution cycle.

If the scan time exceeds the high-speed execution cycle, the fixed scan cycle cannot be maintained and an error of one scan time will occur at maximum in the timer execution program cycle. (☞ Page 55 Phase and execution order)

Ex.
When the high-speed execution cycle is 200ms
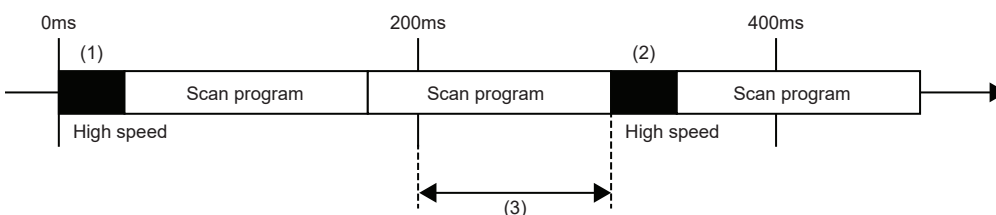
(1) When the timer exceeds the specific cycle, the program block for process control is executed.

(2) When the scan program is executed for the specific cycle or longer, the program block for process control is executed after the execution of the scan program ends.

(3) The time to the end of the last scan program is an error. An error of one scan time can be occur at maximum.

Phase and execution order

For the normal/low-speed execution cycle, set the phase.

With the phase setting, a program block can be executed at a timing shifted by a constant interval from the set execution cycle.

Set the phase on the "Properties" window of the program block.

⊟ Inherent Property	
Execution Interval	Normal-Speed(800ms) ▾
Phase	1 ⬍

Ex.

Phase for the normal-speed execution cycle of 800ms

On the "Options" window of the engineering tool, set "200ms" to the high-speed execution cycle and "4" to the setting value of normal-speed execution cycle.

⊟ Execution Interval	
High-Speed	200ms ▾
Normal-Speed (High-Speed x Setting Value)	4 ▾
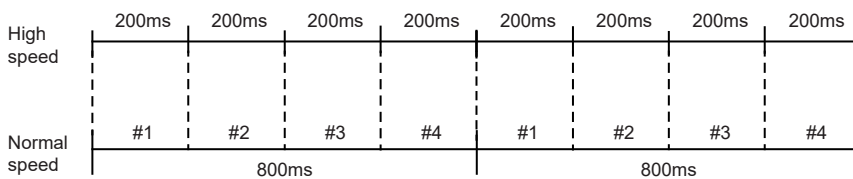
In this case, the execution cycle of 800ms is divided into four sections. The phases #1 to #4 can be set for the normal-speed program block.

Ex.

Fixed period execution of program block with phase
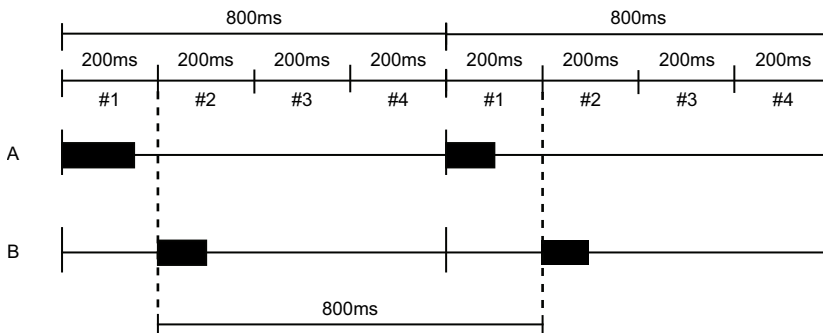
The execution cycle of 800ms is divided into four sections (200ms × 4). Set the program block A to be executed in the phase #1 and the program block B in the phase #2. Consequently, when the execution time of the program block A is 200ms or shorter, the execution cycle of the program block B is always 800ms regardless of the execution time of the program block A.

Ex.

Fixed period execution of program block without phase
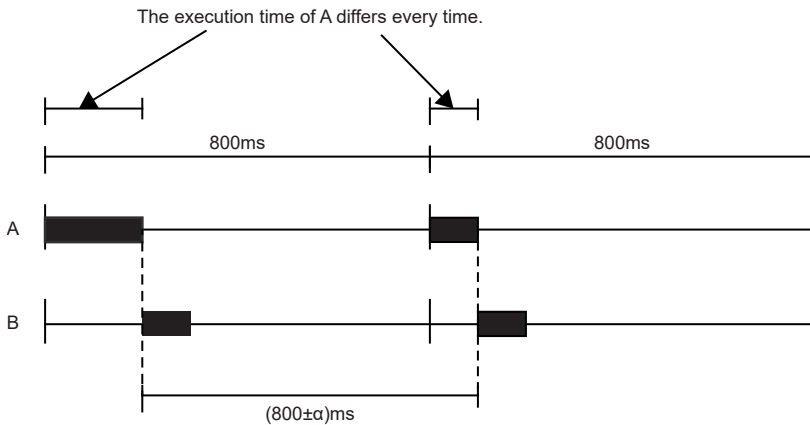
The program blocks A and B have the same execution cycle of 800ms, and the program block A is first executed. In this case, the program blocks are executed simultaneously, and the program block A is executed at an interval of 800ms and the program block B is executed after the execution of the program block A ends. Therefore, the execution time of program block A decides the execution cycle of the program block B, which will not be exactly 800ms. The on-time performance thus degrades. The undetermined factor ($\pm\alpha$) in the execution cycle of the program block B results from the fluctuation of execution time of the program block A.

■Program block with delay

If the program execution time exceeds the phase due to an interrupt execution, the next program block is not executed in the execution cycle, and the on-time performance may degrade.

Ex.

Fixed period execution of program block with delay
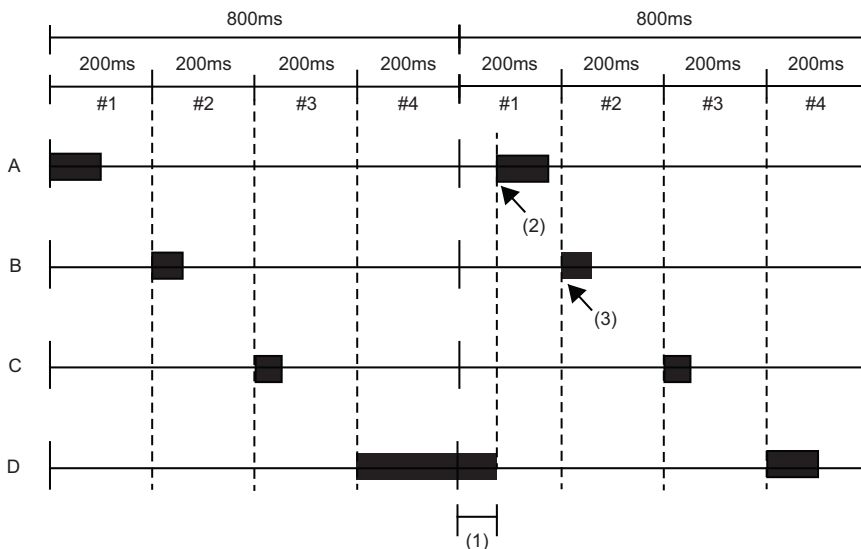
The program block A is executed in the phase #1, the program block B in the phase #2, the program block C in the phase #3, and the program block D in the phase #4.
(1) The execution time of the program block D is 200ms or longer due to an interrupt start program or other causes.
(2) The program block A cannot be executed until the execution of the program block D ends. Therefore, the execution cycle of the program block A is not 800ms, and the on-time performance will degrade.
(3) However, when the total sum of the exceeding time of the program block D and the execution time of the program block A is 200ms or shorter, the program block B is executed in the normal execution cycle.

56   3 PROCESS CONTROL FUNCTION BLOCK
3.4 Program Execution Control

■Program block with the same execution cycle and phase

When multiple program blocks are set to be executed in the same phase, the program blocks are executed in the order set on the "Program File Setting" window of the engineering tool.

Ex.
Fixed period execution of program blocks with the same execution cycle and phase
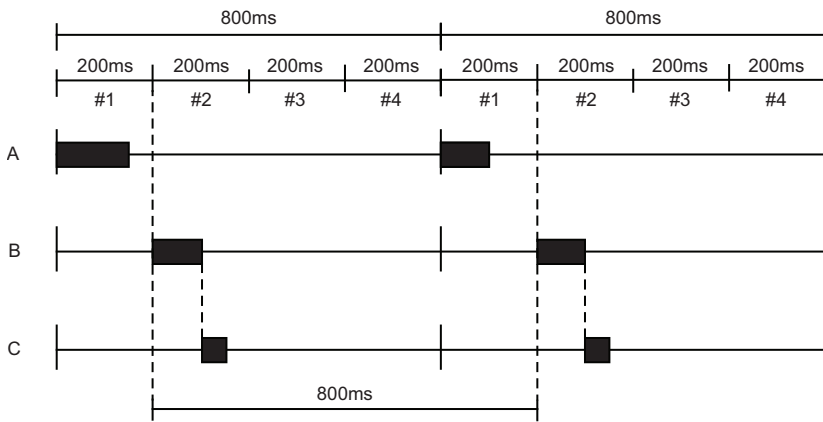
Although the program blocks B and C have the same cycle and phase, the program block B is executed first in accordance with the execution order.

■Synchronizing phases in different execution cycles

The normal-speed program block is executed once when the high-speed program block is executed for "Normal-speed execution cycle ÷ High-speed execution cycle" times. The low-speed program block is executed once when the high-speed program block is executed for "Low-speed execution cycle ÷ High-speed execution cycle" times.

For example, when the execution cycle of high-speed program block is 100ms and that of normal-speed program block is 500ms, the normal-speed program block is executed once when the high-speed program block is executed five times. Therefore, if the scan time exceeds 100ms, each program block cannot be executed in the setting cycle.

For example, when the execution cycle of normal-speed program block is 500ms and the scan time is 150ms, the high-speed program block is executed every 150ms. In this case, the normal-speed program block is executed once when the high-speed program block is executed five times. Consequently, the execution cycle of normal-speed program block is 750ms, and an error of 250ms (750ms - 500ms) will occur. Therefore, the scan time must be set to 100ms or shorter.

■Phase at operation change of the CPU module

A program is executed from the first phase when:
 • The CPU module is reset.
 • The CPU module is powered off and on.
 • The operating status of the CPU module is switched from STOP to RUN.

Control cycle (CT)

A control cycle is the cycle in which the loop control operation is performed.

Set an integral multiple of the execution cycle as the control cycle for each tag. The execution cycles are counted, and the loop control operation is performed when the specified number of control cycles is reached.

Ex.

When the control cycle of M+M_PID is 1s in the program block with the execution cycle of 200ms

Program block (ΔT = 200ms)
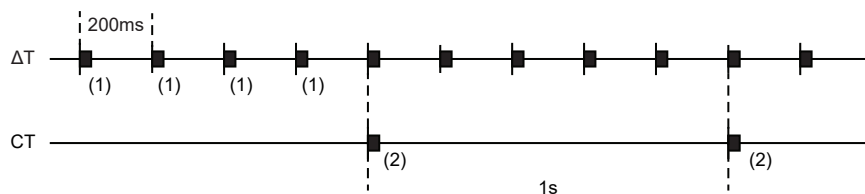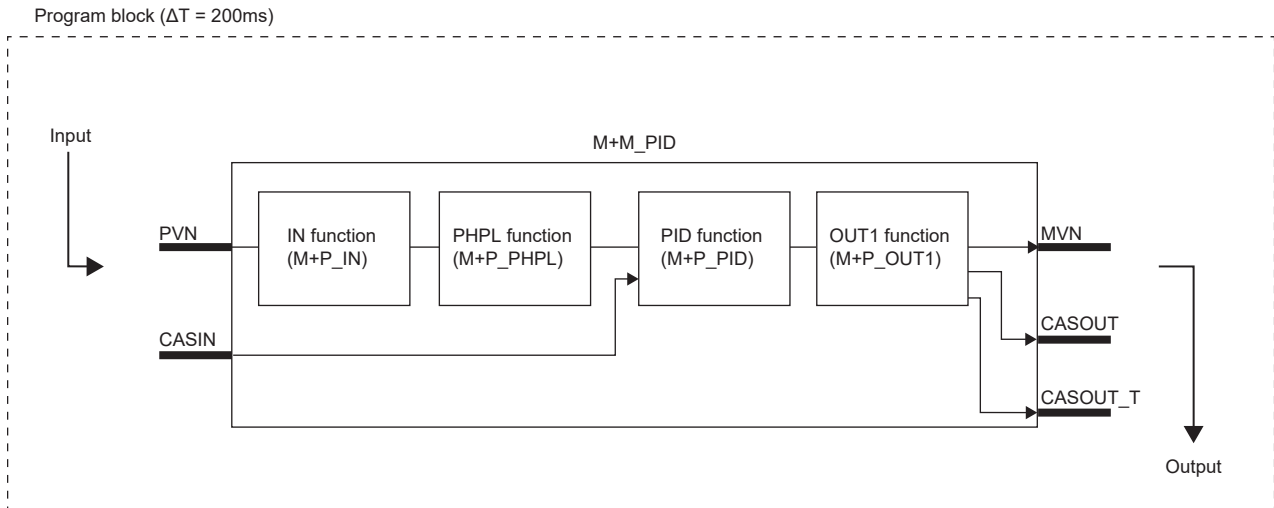
(1) The IN function, PHPL function, and OUT1 function of M+M_PID are executed every 200ms. The PID function is not executed and the last value is used for an output to OUT1.

(2) The IN function, PHPL function, PID function, and OUT1 function of M+M_PID are executed.

If the control cycle is not the integral multiple of execution cycle, round off the number after the decimal point of control cycle (CT) ÷ execution cycle (ΔT) and multiply the execution cycle to calculate the control cycle.

For example, when the execution cycle (ΔT) is 1.0s and the control cycle (CT) is 2.5s, the calculation is 2.5 ÷ 1.0 = 2.5 → 3. Therefore, the control cycle is 3s.

■**Setting the control cycle (CT)**

Set the control cycle in the "FB Property" of the engineering tool.

Item	Name	Setting/Storage range		Unit	Initial value	Data type	Number of digits after the decimal point
		Low limit	High limit				
CT	Control cycle	0	9999	s	1.00	REAL	2

The following table lists the available tag types and tag access FBs performing operations every control cycle.

Tag type	Tag access FB
PID	M+P_PID(_T)
2PID	M+P_2PID(_T)
2PIDH	M+P_2PIDH(_T)_
PIDP	M+P_PIDP(_T), M+P_PIDP_EX(_T)_
IPD	M+P_IPD(_T)
BPI	M+P_BPI(_T)
R	M+P_R(_T)
ONF2	M+P_ONF2(_T)
ONF3	M+P_ONF3(_T)
SWM	M+P_MSET_
SEL	M+P_SEL_T3_

Interrupt execution (fixed scan)

The interrupt execution interrupts the timer execution program and executes a program.

Set the program file as a fixed scan execution type program. The program file is executed at the fixed scan interval which has been set. (□ MELSEC iQ-R CPU Module User's Manual (Application))

The fixed scan interval is recommended to set 10ms or more.

When a safety program is used

The safety program is executed at the timing of the safety cycle time. The standard program (+ END processing) is executed in remaining time of the safety cycle time. Therefore, if the safety program + the safety input/output processing time is long, the execution cycle set for the FBD/LD program for process control may shift.

For the safety program and the safety cycle time, refer to the following.

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

3.5 Auto Tuning

The auto tuning function detects dynamic characteristics of a control target and automatically tunes the proportional gain (Kp), integral time (Ti), and derivative time (Td) for PID to suitable values.

The auto tuning function has two methods: step response method and limit cycle method.

Item	AT1 (Step response method)	AT2 (Limit cycle method)
Overview	In this method, a proportional gain (Kp), an integral time (Ti), and a derivative time (Td) for PID operations are calculated with the ZN method (Ziegler-Nichols' step response method) for setting their initial values.	In this method, a proportional gain (Kp), an integral time (Ti), and a derivative time (Td) for PID operations are calculated from the variation amplitude and variation cycle of a process variable obtained by repeatedly outputting the high and low limit values of a manipulated value. The limit cycle method is less influenced by the process variable noise compared to the step response method, leading a stable tuning result.
Applicable control mode	MANUAL, COMPUTER MV	AUTO, MANUAL, CASCADE, COMPUTER MV, COMPUTER SV

PID constants calculation specification method	■P control tuning
	  Execute tuning after setting Ti = 0 and Td = 0.
	■PI control tuning
	  Execute tuning after setting Ti > 0 and Td = 0.
	■PID control tuning
	  Execute tuning after setting Ti > 0 and Td > 0.
		Execute tuning by selecting the PI control or PID control with the PX Developer monitor tool.
		(Tuning only for the P control cannot be executed.)

Corresponding tag access FB	• M+P_PID(_T) • M+P_PID_DUTY(_T) • M+P_2PID(_T) • M+P_2PID_DUTY(_T) • M+P_2PIDH(_T)_	• M+P_2PIDH(_T)_
Corresponding tag FB	• M+M_PID(_T) • M+M_PID_DUTY(_T) • M+M_2PID(_T) • M+M_2PID_DUTY(_T) • M+M_2PIDH(_T)_	• M+M_2PIDH(_T)_

Point

For the execution and status check of the auto tuning, refer to the following.

☞ Page 145 Checking auto tuning operation

Step response method

■Operation method and processing details

Hold the manipulated value and stabilize the process variable, and then perform the following procedure.

1. Display the auto tuning window of the PX Developer monitor tool.

👆 [Control Panel] ⇨ [Details] button on the faceplate ⇨ [Auto Tuning] button ⇨ [Select Auto Tuning Operations] window ⇨ [Executes Auto Tuning by Step Response method] ⇨ [Next] button
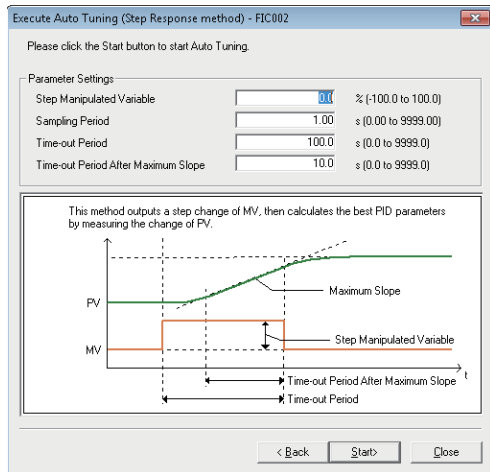
2. Set the following items and click the [Start] button.
 • Step Manipulated Variable (AT1STEPMV)
 • Sampling Period (AT1ST) (seconds)
(PV data collection period during tuning)
 • Time-out Period (AT1TOUT1) (seconds)
 • Time-out Period After Maximum Slope (AT1TOUT2) (seconds)

3. The "Step Manipulated Variable" is output from the current manipulated value in step form.

4. The manipulated value returns to its original value automatically after auto tuning is completed. P, I, and D constants generated from auto tuning are automatically saved in the tag data.

> Point🔎
> • PID constants are automatically overwritten after auto tuning. Save the previous PID constants in advance as necessary.
> • Auto tuning stops automatically when an alarm occurs.

■Operation details of step response method

The manipulated value is output in step form to the actual plant, and each constant is determined according to the maximum slope and equivalent dead time.

The following operations are automatically performed. The generated value is saved in P, I, and D areas of the tag data.
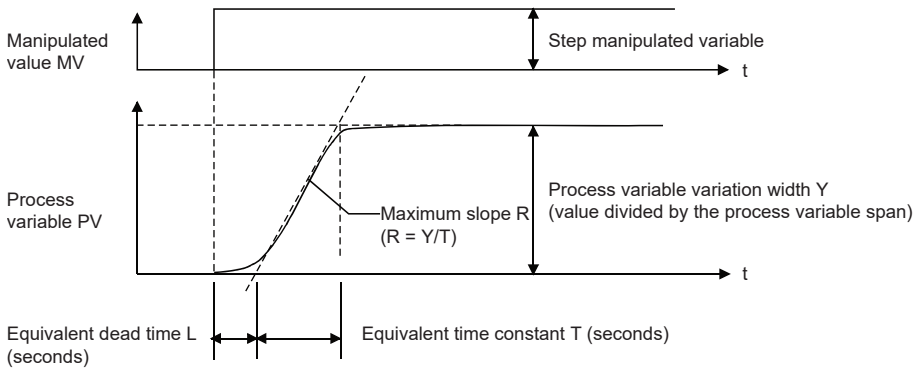
- Constants calculation with ZN method (Ziegler-Nichols' step response method)

The P/PI/PID control type is determined by I and D values of the tag data before auto tuning.

Condition		Control type	Constant		
Integral time (I)	Derivative time (D)		Proportional gain (P)	Integral time (I) (seconds)	Derivative time (D) (seconds)
I = 0 (Integral time ∞)	D = 0	P control	$\dfrac{1.0}{R \times L} \times \dfrac{\left\|\text{Step manipulated variable}\right\|}{100}$	0 (Integral time ∞)	0
I > 0	D = 0	PI control	$\dfrac{0.9}{R \times L} \times \dfrac{\left\|\text{Step manipulated variable}\right\|}{100}$	3.33L	0
I > 0	D > 0	PID control	$\dfrac{1.2}{R \times L} \times \dfrac{\left\|\text{Step manipulated variable}\right\|}{100}$	2L	0.5L

Ex.

Auto tuning is executed when the tag data before auto tuning are I > 0 and D = 0.

(For I > 0 and D = 0, the PI control calculates constants.)

Item	Value
Step manipulated variable	20%
Equivalent dead time L	8 seconds
Equivalent time constant T	16 seconds
Process variable variation width Y	0.25
Maximum slope R	0.25 / 16 = 0.016

$$\text{Proportional gain (P)} = \frac{0.9}{R \times L} \times \frac{\left|\text{Step manipulated variable}\right|}{100} = \frac{0.9}{0.016 \times 8} \times \frac{20}{100} = 1.4$$

Integral time (I) = 3.33L = 3.33 × 8 = 26.6 seconds, derivative time (D) = 0 second

■Fine tuning after auto tuning

After auto tuning is completed, the change of process variable (PV) in relation to the setting value (SV) is observed on the tuning setting execution window of the PX Developer monitor tool. The optimal values are obtained by adjusting the P, I, and D values.

The response of the process variable (PV) corresponding to the change of the setting value (SV) is observed.

- Response is quick, but oscillatory.

Fine tuning when the response is quick, but oscillatory
  - Proportional gain: Reduce (Proportional effect is reduced.)
  - Integral time: Increase (Integral effect is reduced.)

- Optimal value

- Response is slow

Fine tuning when the response is slow
  - Proportional gain: Increase (Proportional effect is increased.)
  - Integral time: Reduce (Integral effect is increased.)

When the derivative action is applied, the derivative time is adjusted with the stability and rapid response check. (Increasing the derivative time increases the derivative effect, and reducing the derivative time reduces the derivative effect.)

Limit cycle method

■Operation method and processing details

1. Display the [Execute Auto Tuning (Limit Cycle method)] window of the PX Developer monitor tool.

✎ [Control Panel] ⇨ [Details] button on the faceplate ⇨ [Auto Tuning] button ⇨ [Select Auto Tuning Operations] window ⇨ [Executes Auto Tuning by Limit Cycle method] ⇨ [Next] button
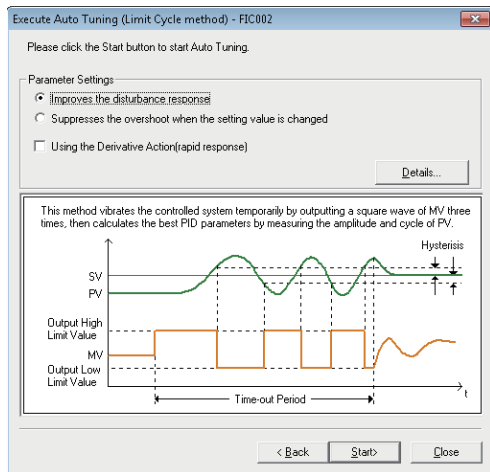
2. Set the following items and start auto tuning.

The control type is determined by selecting either of "Improves the disturbance response" or "Suppresses the overshoot when the setting value is changed", and whether to check "Using the Derivative Action (rapid response)".

○: Selected, ×: Not selected

Control type	Improves the disturbance response	Suppresses the overshoot when the setting value is changed	Using the Derivative Action (rapid response)
Constant-value PI control	○	×	×
Constant-value PID control	○	×	○
Follow-up PI control	×	○	×
Follow-up PID control	×	○	○

Set the following items on the detail setting window of the limit cycle method which is displayed by clicking the [Details] button.

- Output high limit (AT2MVH)
- Output low limit (AT2MVL)
- Hysteresis (AT2HS)
- Time-out period (AT1TOUT1) (seconds)

3. The high and low limit values of the manipulated value are repeatedly output.

Even if the value set to AT2MVH/AT2MVL exceeds MH/ML, the manipulated value is output within the range of MH to ML.

4. The manipulated value returns to its original value automatically after auto tuning is completed.

Values for proportional gain (Kp), integral time (Ti), and derivative time (Td) which are calculated by auto tuning are set automatically.

> *Point*
>
> - PID constants are automatically overwritten after auto tuning.
> - Auto tuning stops automatically when an alarm occurs.
> - MV output values return to the values at start when auto tuning is completed or interrupted.

■Operation details of the limit cycle method

● Generation and measurement of the limit cycle waveform

In the AUTO TUNING mode, the PV limit cycle waveform is generated by performing the 2-position ON/OFF operation of MV output three times.

The 2-position ON/OFF operation is performed under the following conditions.

Control operation	First MV output	2-position ON/OFF operation	Remarks
Reverse action (PN = 0)	■PV ≤ SV 　MV = Output high limit (AT2MVH) ■PV > SV 　MV = Output low limit (AT2MVL)	■PV ≥ SV + Hysteresis (AT2HS) 　MV = Output low limit (AT2MVL) ■PV ≤ SV - Hysteresis (AT2HS) 　MV = Output high limit (AT2MVH)	For operation images, refer to the following examples.
Direct action (PN = 1)	■PV ≤ SV 　MV = Output low limit (AT2MVL) ■PV > SV 　MV = Output high limit (AT2MVH)	■PV ≥ SV + Hysteresis (AT2HS) 　MV = Output high limit (AT2MVH) ■PV ≤ SV - Hysteresis (AT2HS) 　MV = Output low limit (AT2MVL)	The MV high limit and low limit values at PN = 0 are reversely output.

The PV oscillation waveform data for the limit cycle method at the first 2-position ON/OFF operation is ignored. The oscillation amplitude Xc and oscillation period Tc are measured by using the PV oscillation waveform data at the second and third 2-position ON/OFF operations.

Auto tuning ends at the apex of the third PV oscillation waveform.

The setting value is calculated by using SVC (setting value (current)) at auto tuning start.

The hysteresis (AT2HS) functions as a minimum compensation value of the amplitude. Set it in advance according to the control target so that the oscillation period and oscillation amplitude are measured properly.

Ex.

PV ≤ SV (PN = 0)
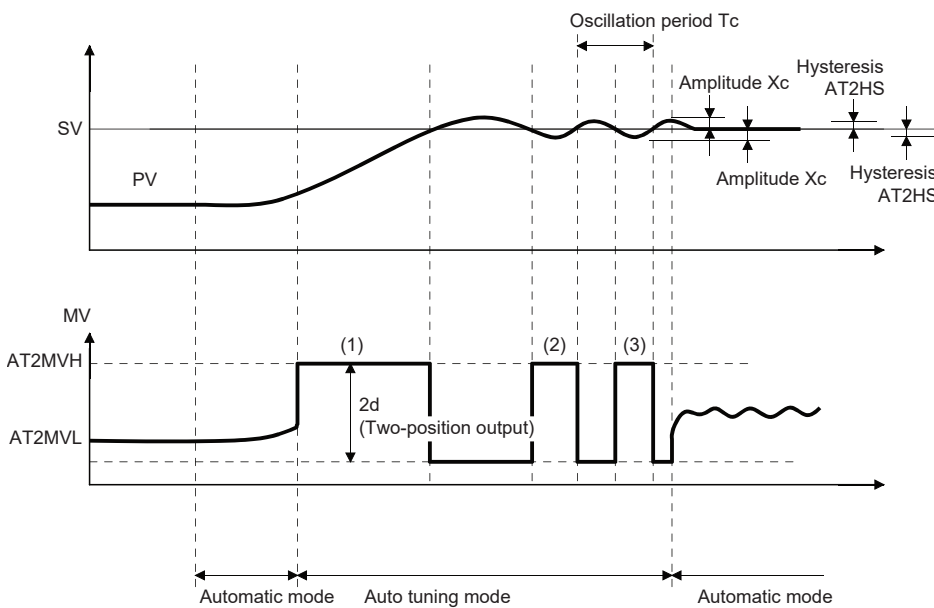
(1) 1st 2-position operation
(2) 2nd 2-position operation
(3) 3rd 2-position operation

Ex.
PV > SV (PN = 0)
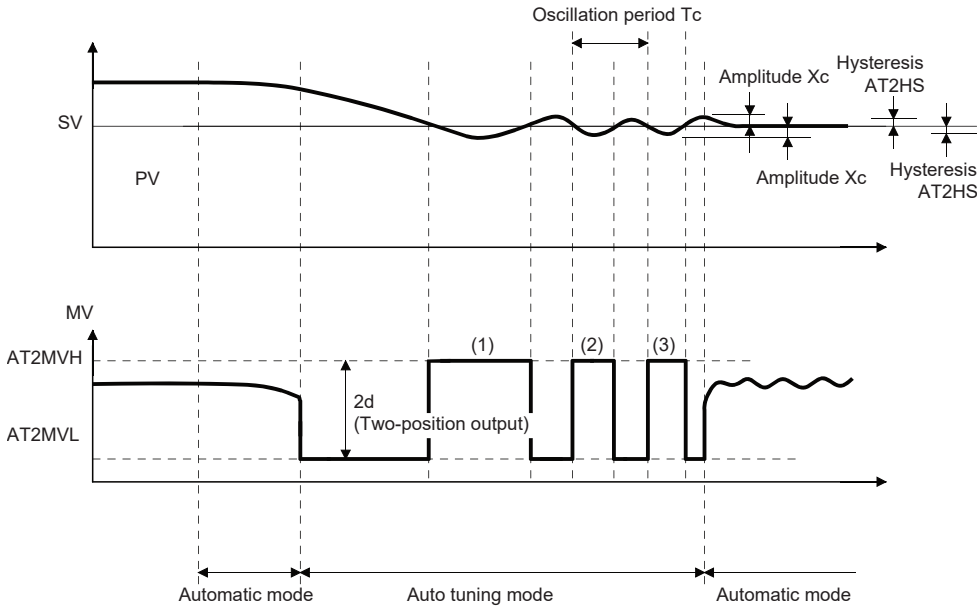
(1) 1st 2-position operation
(2) 2nd 2-position operation
(3) 3rd 2-position operation

The oscillation amplitude Xc is calculated by measuring and averaging out the plus side and minus side maximum values of |PV - SV|.

The output range d is calculated by (AT2MVH - AT2MVL) / 2.

● Calculation of threshold sensitivity and threshold period

The threshold sensitivity (Ku) and the threshold period (Tu) are calculated from the measurement result of auto tuning by the limit cycle method.

$Ku = 4d / (\pi\sqrt{(Xc^2 - AT2HS2)})$

$Tu = Tc$

● Calculation of optimal PID constant

The optimal PID constant is calculated from the threshold sensitivity (Ku) and threshold period (Tu).

Values of the proportional gain (Kp), integral time (Ti), and derivative time (Td) are calculated by using the coefficients specified for the control type (ATTYPE) shown below.

Control type	Control operation	Control type (ATTYPE)	Proportional gain (Kp)	Integral time (Ti)	Derivative time (Td)	Empirical rule
Constant-value control	PI	1	0.45Ku	0.83Tu	0	Ziegler-Nichols' method
	PID	2	0.6Ku	0.5Tu	0.125Tu	
Follow-up control	PI	3	0.3Ku	1.0Tu	0	CHR method
	PID	4	0.45Ku	0.6Tu	0.1Tu	

Ku: Process threshold sensitivity, Tu: Process threshold period

■Fine tuning after auto tuning

Fine tuning of PID constants is the same as that for the step response method. (☞ Page 64 Fine tuning after auto tuning)

4 DEMONSTRATION MACHINE

4.1 Demonstration Machine Configuration

Hardware configuration

The following describes the hardware configuration of the demonstration machine. The power supply switch is used to supply power to each power supply module of the programmable controller. The heater is equipped with a thermocouple for temperature control. The temperature control can be stopped by using the emergency stop button (push-lock and turn-reset switch).

The operation is performed from the GOT2000.

Front side of the demonstration machine

Left side

Right side
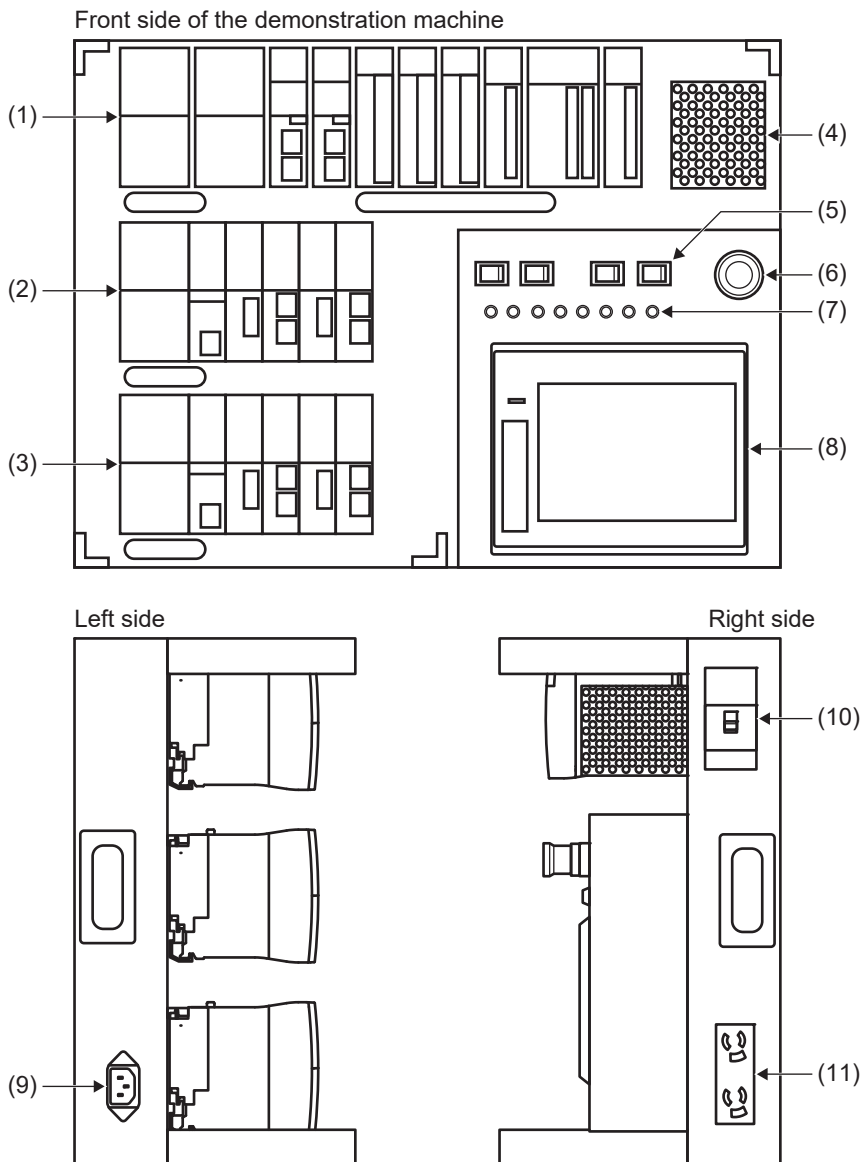
No.	Name		Product model
(1)	Slave station	Power supply	R64RP
		CC-Link IE Field Network remote head module	RJ72GF15-T2
		DC input module with diagnostic functions	RX40NC6B
		Transistor output module with diagnostics function	RY40PT5B
		Contact output module	RY10R2
		Channel isolated thermocouple input module	R60TD8-G
		Channel isolated digital-analog converter module	R60DA16-G
		Channel isolated analog-digital converter module	R60AD8-G
		Base unit	R38RB-HT
(2)(3)	Master station system A (redundant) Master station system B (redundant)	Power supply	R61P
		Process CPU[*1]	R32PCPU
		Redundant function module	R6RFM
		Ethernet module[*2]	RJ71EN71
		CC-Link IE Controller module[*2]	RJ71GF21-SX
		CC-Link IE Field module[*2]	RJ71GF11-T2
		Base unit	R35B
(4)	Heater	—	—
(5)	Power supply switch	—	—
(6)	Emergency stop (temperature control stop)	—	—
(7)	LED (lighting every 10ms)	—	—
(8)	GOT2000	GT2705-V	—
(9)	Main power supply	—	—
(10)	Main breaker	—	—
(11)	Outlet (100V, 2 ports)	—	—

*1 The redundant function is supported in the firmware version 4 or later.

*2 The firmware version 12 or later is used when using functions compatible with the redundant system.

4

System configuration

The following describes the system configuration of the demonstration machine. The redundant system (master station) is configured with the slave station and the CC-Link IE Field Network redundant lines.

The demonstration machine is operated from the personal computer (monitor tool) and GOT2000 via Ethernet. The CC-Link IE Controller Network is used to read the status of the other system.

In this exercise, GX Works3 version 1.047Z will be used for module configuration and program creation, and PX Developer version 1.50C will be used for monitoring the process control system.
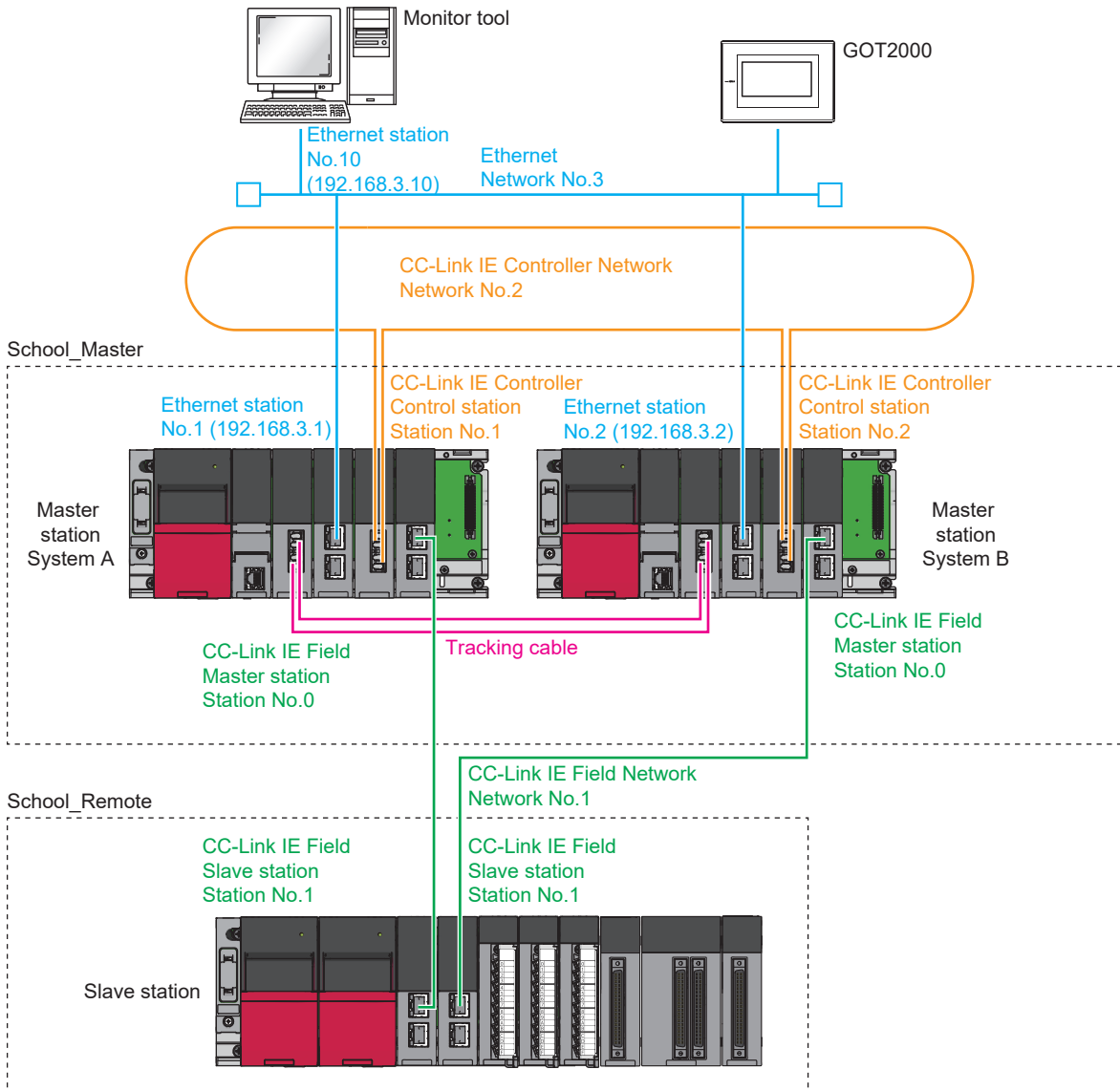
Point

Configure the TCP/IP setting to set the IP address of the personal computer to 192.168.3.10 in advance.

4.2 Wiring the Demonstration Machine

Ethernet

This section describes the wiring for Ethernet.

1. Connect an Ethernet cable to "P1" of the RJ71EN71 of the master station system A.
   This Ethernet cable is connected to the hub inside the demonstration machine, through which the master station system A and system B, GOT2000, and personal computer are connected.

Ethernet cable

Master station
System A

P1

RJ71EN71

Master station
System B
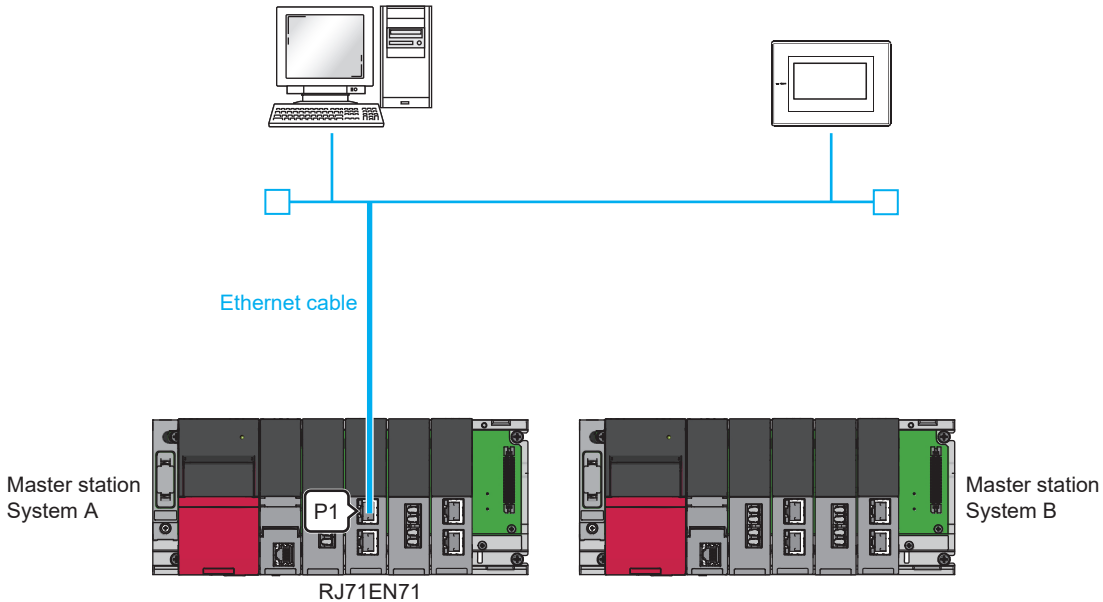
2. In the same way, connect an Ethernet cable to "P1" of the RJ71EN71 of the master station system B.

Ethernet cable

Master station
System A

Master station
System B

P1

RJ71EN71
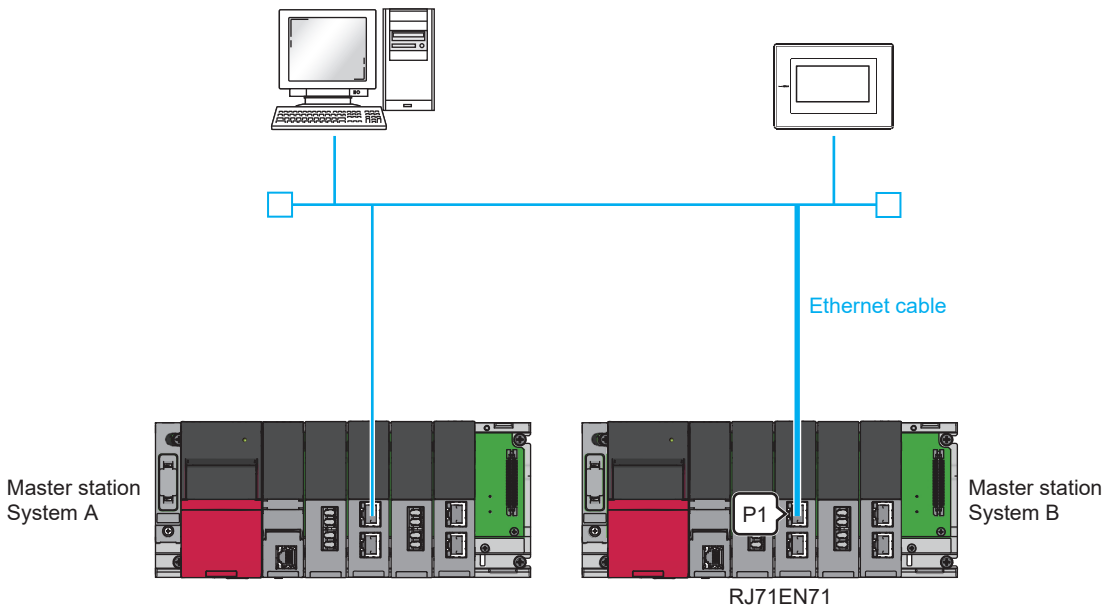
Point

A hub is installed inside the demonstration machine. Connect the Ethernet cable to the hub in advance, and route the cable through the hole located on the front of the demonstration machine and pull out the connector.

CC-Link IE Controller Network

This section describes the wiring for CC-Link IE Controller Network.

1. Connect "IN" of the RJ71GP21-SX of the master station system A and "OUT" of the RJ71GP21-SX of the master station system B with an optical fiber cable.

Optical fiber cable

Master station
System A

IN

RJ71GP21-SX

Master station
System B

OUT
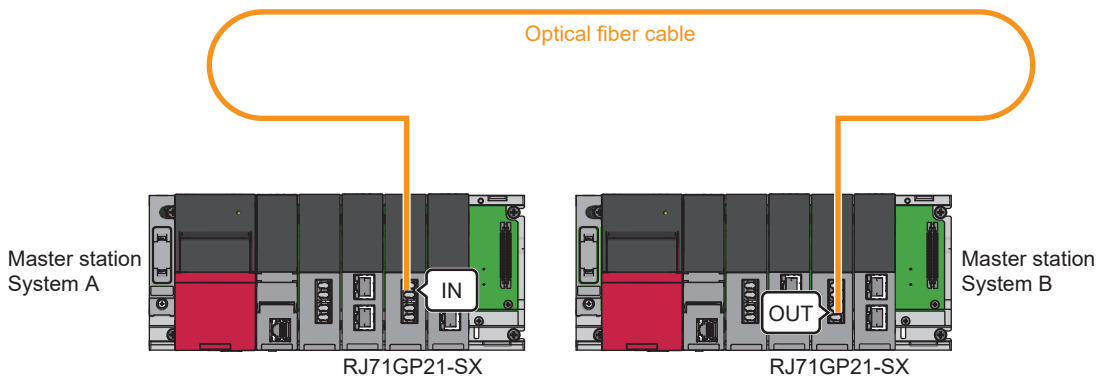
RJ71GP21-SX

2. Connect "IN" of the RJ71GP21-SX of the master station system B and "OUT" of the RJ71GP21-SX of the master station system A with an optical fiber cable.

Optical fiber cable

Master station
System A

OUT

RJ71GP21-SX

Master station
System B

IN

RJ71GP21-SX
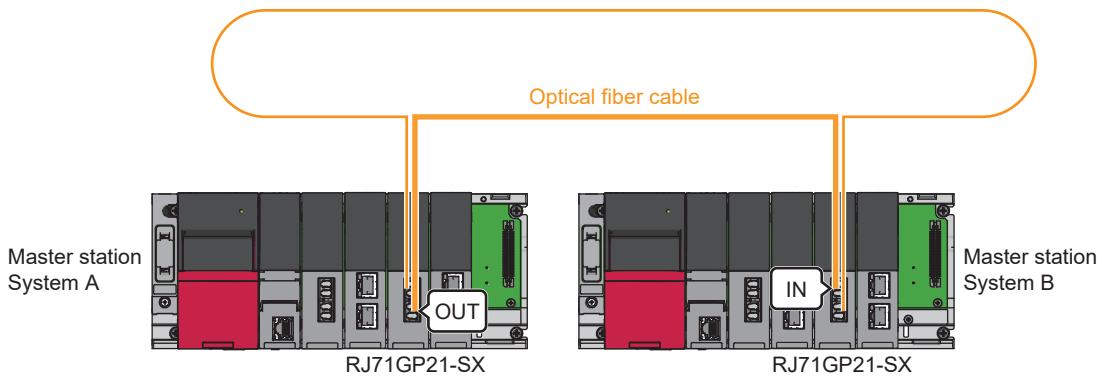
CC-Link IE Field Network

This section describes the wiring for CC-Link IE Field Network.

1. Connect "P1" of the RJ71GF11-T2 of the master station system A and "P1" of the RJ72GF15-T2 of the slave station with an Ethernet cable.
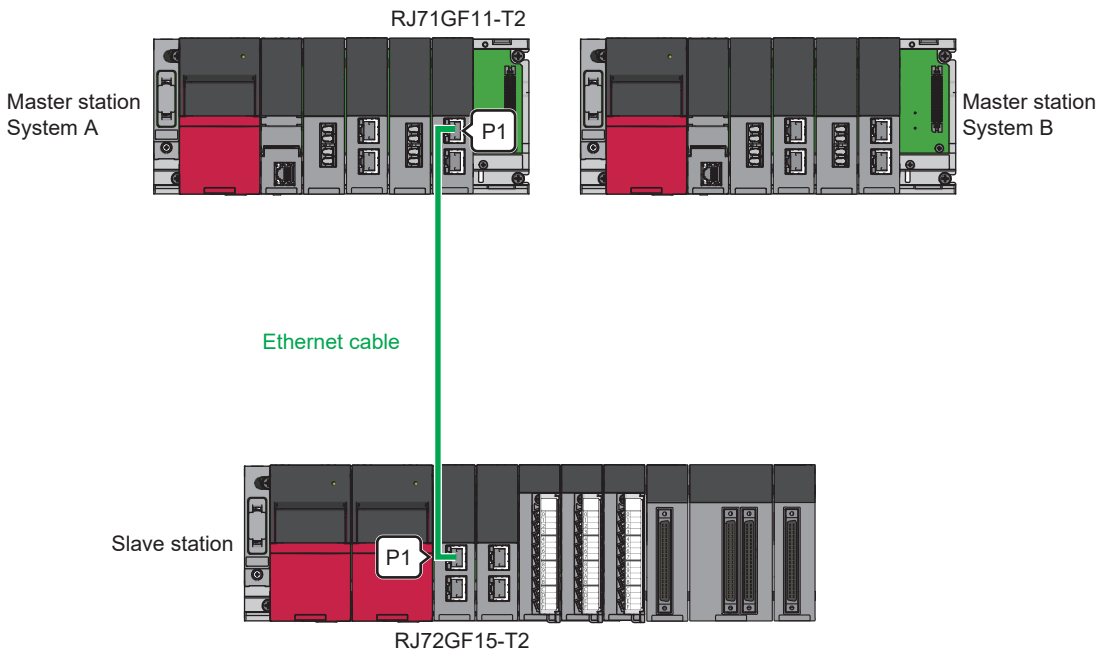
2. Then, connect "P1" of the RJ71GF11-T2 of the master station system B and "P1" of the RJ72GF15-T2 of the slave station with an Ethernet cable.
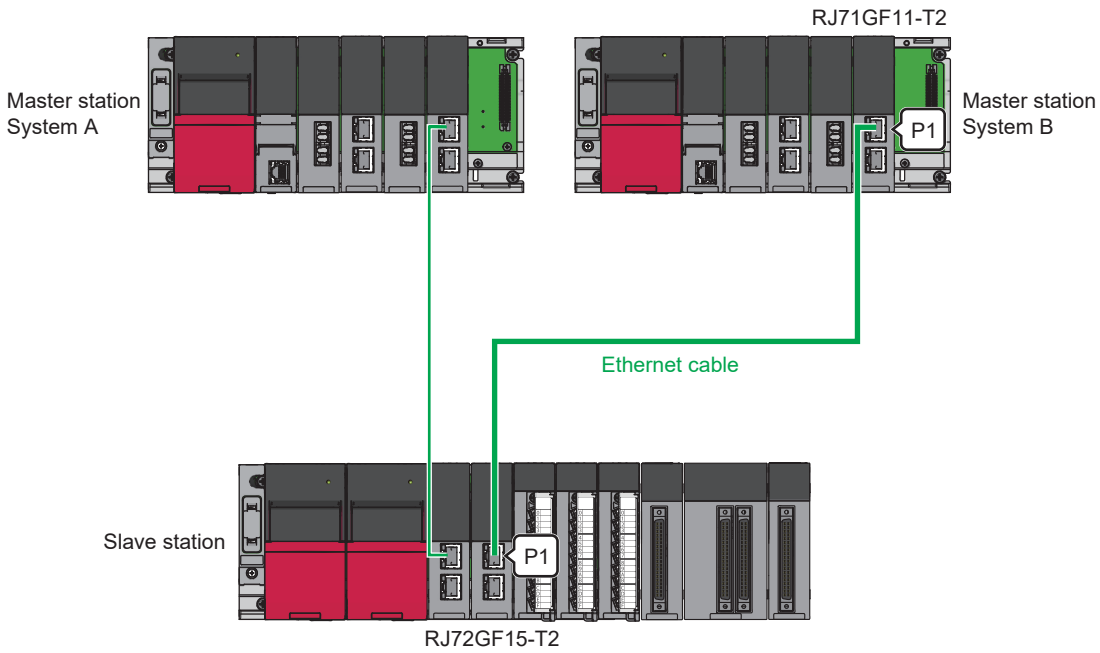
Redundant function module

This section describes how to wire the redundant function modules.

1. Connect "IN" of the R6RFM of the master station system A and "OUT" of the R6RFM of the master station system B with a tracking cable.
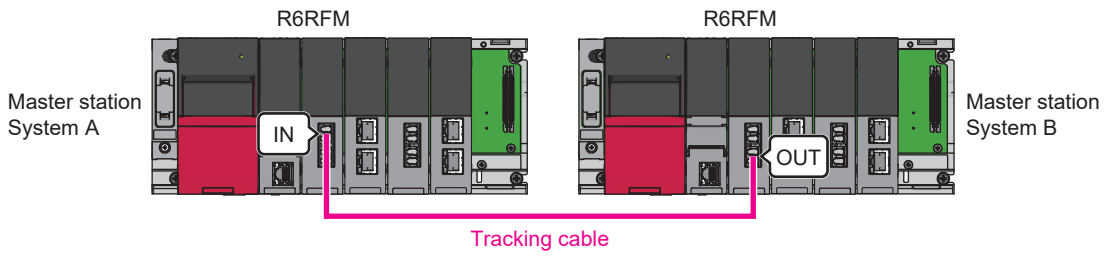
2. Connect "IN" of the R6RFM of the master station system B and "OUT" of the R6RFM of the master station system A with a tracking cable.
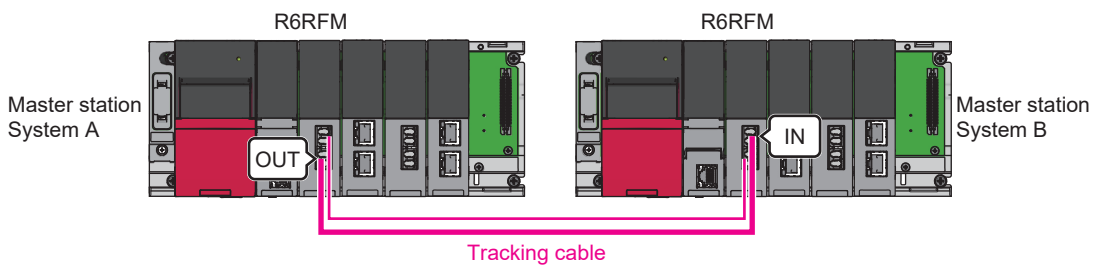
4.3 Procedure for Starting Up the Demonstration Machines

Overview

The following table lists two procedures for starting up a redundant system.

Start-up procedure	Description
Starting up both systems at a time	Prepare both systems and follow this procedure to start up both systems as a redundant system when the systems are powered on. ☞ Page 75 Starting up both systems at a time
Starting up the systems one by one	Follow this procedure to start up a control system first to perform controls, and then start up a standby system to build a redundant system. • Debugging before operation can be performed with only one system. • If one system has failed and the system is powered off before module replacement, powering on this system again starts up only this system. For details, refer to the following manual. 📖 MELSEC iQ-R CPU Module User's Manual (Application)

Starting up both systems at a time

1. Installing batteries

Install a battery to the CPU module of each system. (📖 MELSEC iQ-R CPU Module User's Manual (Startup))

2. Inserting extended SRAM cassettes and SD memory cards

Mount an extended SRAM cassette and insert an SD memory card to the CPU module of each system as needed. (📖 MELSEC iQ-R CPU Module User's Manual (Startup))

Do not power off or reset the CPU module, or remove an SD memory card during an access to the SD memory card.

3. Mounting modules and connecting cables

Mount modules on the base unit, and connect cables. Use modules of the same model, and mount them on the same slots on each base unit. (📖 MELSEC iQ-R Module Configuration Manual, ☞ Page 71 Wiring the Demonstration Machine)

4. Powering on the system

Check each system for the following and power on the systems.
• Wiring to the power supply module is correct.
• Power supply voltage is within the specified range.
• The CPU module is in the STOP state.

Check that the following LEDs turn on after the system is powered on.
• Power supply module: POWER LED
• CPU module: READY LED
• Redundant function module: RUN LED

In step 4, the ERROR LED of the CPU module flashes and the ERR LED of the redundant function module turns on. Proceed to the next step.

5. Connecting a personal computer and the CPU module

Start an engineering tool installed on a personal computer.

Connect the personal computer to the CPU module of one system.

6. Initializing the CPU module

Initialize the CPU module using the engineering tool. (📖 MELSEC iQ-R CPU Module User's Manual (Startup))

After initialing the CPU module, connect the other CPU module and the personal computer.

Initialize the CPU module in the same way.

7. Setting parameters

Set system parameters, CPU parameters, and module parameters.
 • To execute functions accessing the SD memory card, set memory card parameters as well.
 • To mount an intelligent function module, set intelligent function module parameters.

> **Point**
>
> Users can set system parameters by reading the actual system configuration to the module configuration of the engineering tool.

8. Creating a program

Create a program with the engineering tool. After creating the program, convert the program and save the project.

9. Writing the system A/B setting

Set the system A or B with the engineering tool. (☞ Page 254 System A/B settings)

10. Writing data to the programmable controller

Write the set parameters and created programs to both systems with the engineering tool. (☞ Page 108 Writing a Project)

11. Resetting the CPU module

Restart both systems with either of the following methods.
 • Powering off and on the systems
 • Resetting the CPU module

12. Checking the LEDs

Check that the LEDs of the CPU module and redundant function module are in the following states. The CARD READY LED status depends on whether an SD memory card has been installed to each CPU module or not.

• LEDs of System A
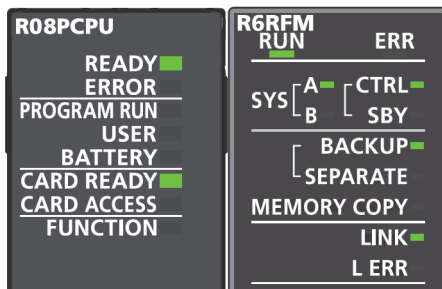
• LEDs of System B
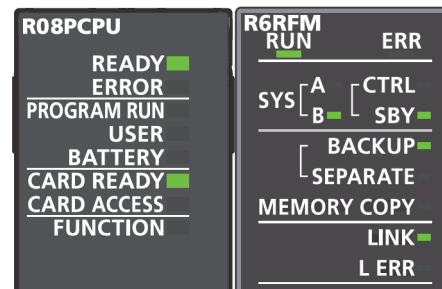
When an error has occurred, the following LEDs are on. Check details of the error with the engineering tool and eliminate the error cause.
 • CPU module: ERROR LED (📖 MELSEC iQ-R CPU Module User's Manual (Startup))
 • Redundant function module: ERR LED, L ERR LED (📖 MELSEC iQ-R CPU Module User's Manual (Application))
To start up the system when the data logging function is used, refer to the following manual.
📖 MELSEC iQ-R CPU Module User's Manual (Application)

13. Executing the program

Power off both systems. Set the RUN/STOP/RESET switch of the CPU module of each system to the RUN position and power on both systems.

Check that the PROGRAM RUN LED of the CPU module of the control system turns on.

> **Point**
>
> Individually setting the RUN/STOP/RESET switch of each CPU module to the RUN position with the CPU module powered on causes a continuation error due to the operating status mismatch and the error is detected in the standby system. Therefore, to start up both systems simultaneously, performing step 13 is recommended.

14. Monitoring the program

Check that the program operates normally using the engineering tool.

4.4 Operating the Demonstration Machine

This section describes details of the exercise performed on each screen and how to operate each screen.

The demonstration machine is operated from the GOT2000 (GT2705-V).

Write Master_forGOT.gx3 to the Process CPU, R32PCPU, and Remote_forGOT.gx3 to the remote head module, RJ72GF15-T2, in advance.

☞ Page 108 Writing a Project
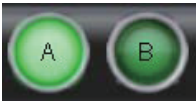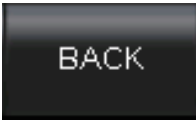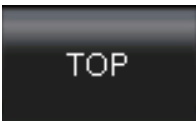
Screen transition

The following shows screen transition of the GOT2000.

[Top screen (English)]

Language switch (EN ⇔ JA)

[Top screen (Japanese)]

Same as English version

START

TOP

[MENU screen]

TOP

BACK

[To each screen]

TOP
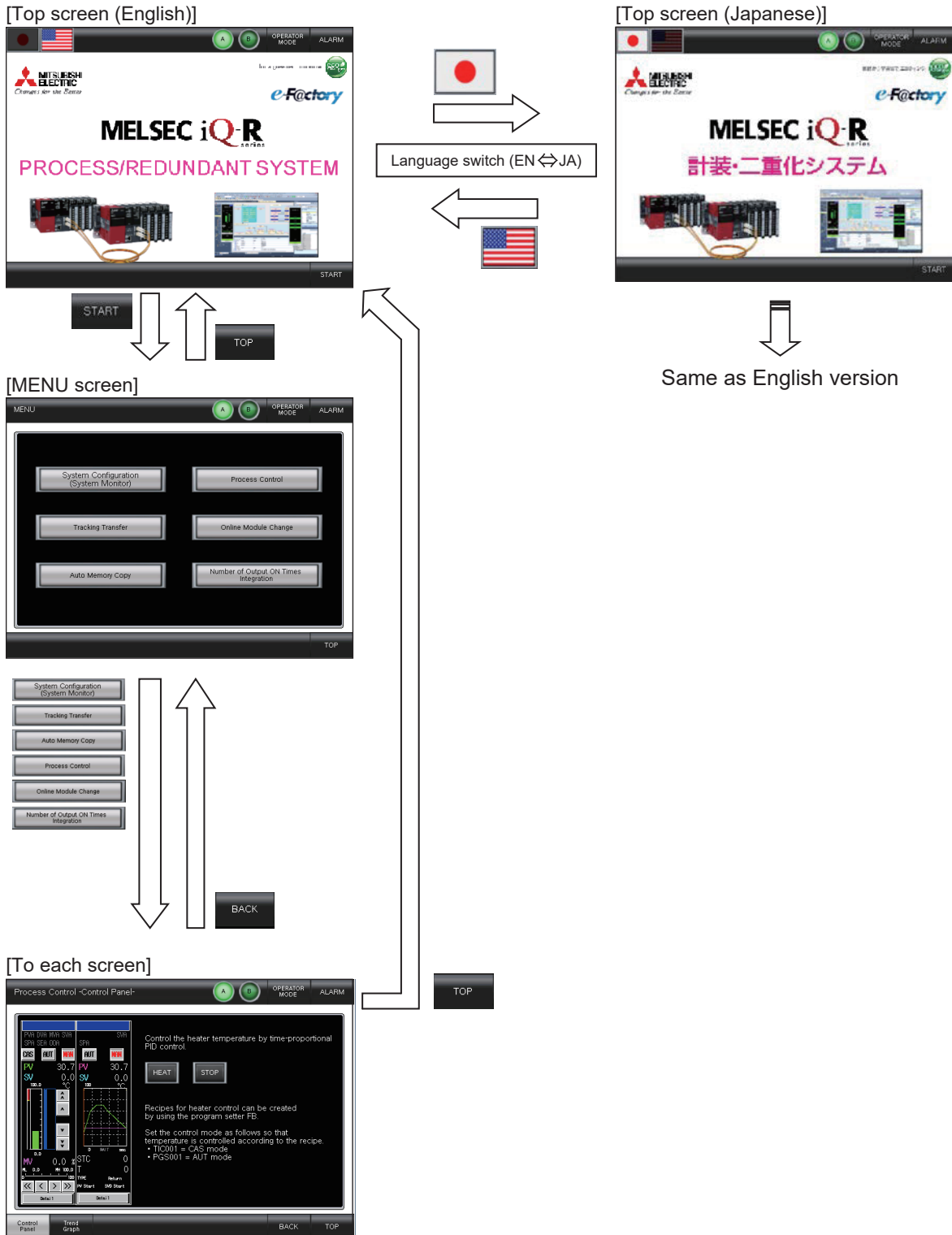
Basic operations

The following describes the basic operations of the demonstration machine.

Lamps/buttons

Item	Description
System A/B indicator lamps System A control　　System B control 	Indicates whether the current control system is system A or system B. The lamp for the current control system is on.
[ALARM] button Normal　　　　Error (Flashing)	Flashes in red if an error occurs. Touch the button to display the alarm screen. The button color changes to black when the alarm is cleared. For details on the alarm screen, refer to the following. ☞ Page 79 Alarm screen
[BACK] button 	Moves back to the previously displayed screen.
[TOP] button 	Moves to the title screen.
Language selector 	Switches the language of the GOT2000 between Japanese and English. The default language is Japanese.

Alarm screen

This section describes how to operate the alarm screen. On the alarm screen, alarms that have occurred can be checked and reset.

The alarm screen consists of the process alarm screen and system alarm screen. Alarms regarding process control (such as sensor errors) are displayed on the process alarm screen, and alarms that have occurred on the Process CPU and GOT2000 are displayed on the system alarm screen.

The process alarm screen is displayed by default when transitioning to the alarm screen.
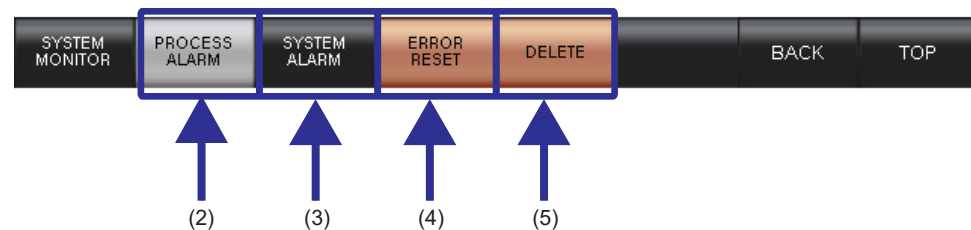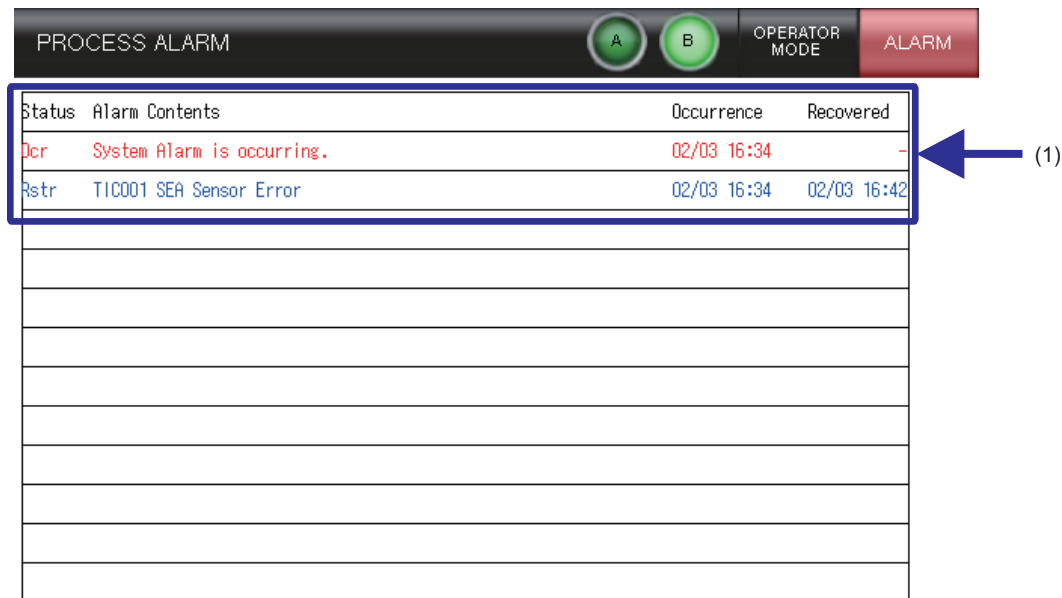
No.	Item	Description
(1)	Alarm display	Displays existing alarms. • Alarms that have not been reset are displayed in red. • Alarms that have been reset are displayed in blue.
(2)	[PROCESS ALARM] button	Moves to the process alarm screen. On the process alarm screen, alarms regarding process control (such as sensor errors) are displayed.
(3)	[SYSTEM ALARM] button	Moves to the system alarm screen. On the system alarm screen, errors that have occurred on the Process CPU and GOT2000 are displayed.
(4)	[ERROR RESET] button	Resets the alarms that has occurred. This operation resets both process alarms and system alarms. Therefore, errors that have occurred on the CPU and other modules are also cleared. The display color of the alarms that have been reset changes from red to blue.
(5)	[DELETE] button	Deletes alarms that have been reset (displayed in blue). If the [ALARM] button flashes red, it changes to black when all alarms are deleted.

Details of each screen

This section describes details of the exercise performed on each screen and how to operate each screen.

The following six screens are used in this exercise, which can be accessed by touching each button on the MENU screen.

System configuration (System monitor) screen

This screen can be displayed by touching the [System Configuration (System Monitor)] button on the MENU screen.

The configuration diagram of the redundant system is displayed on this screen. The Process CPU and remote head module LED status, and the network status are also displayed.

This screen can be projected via a projector to describe the redundant system configuration and to demonstrate system switching operation and standby system diagnostic function. (The network error detection function for the standby system is achieved through "programs executed in both systems and standby system data read via the CC-Link IE Controller".)
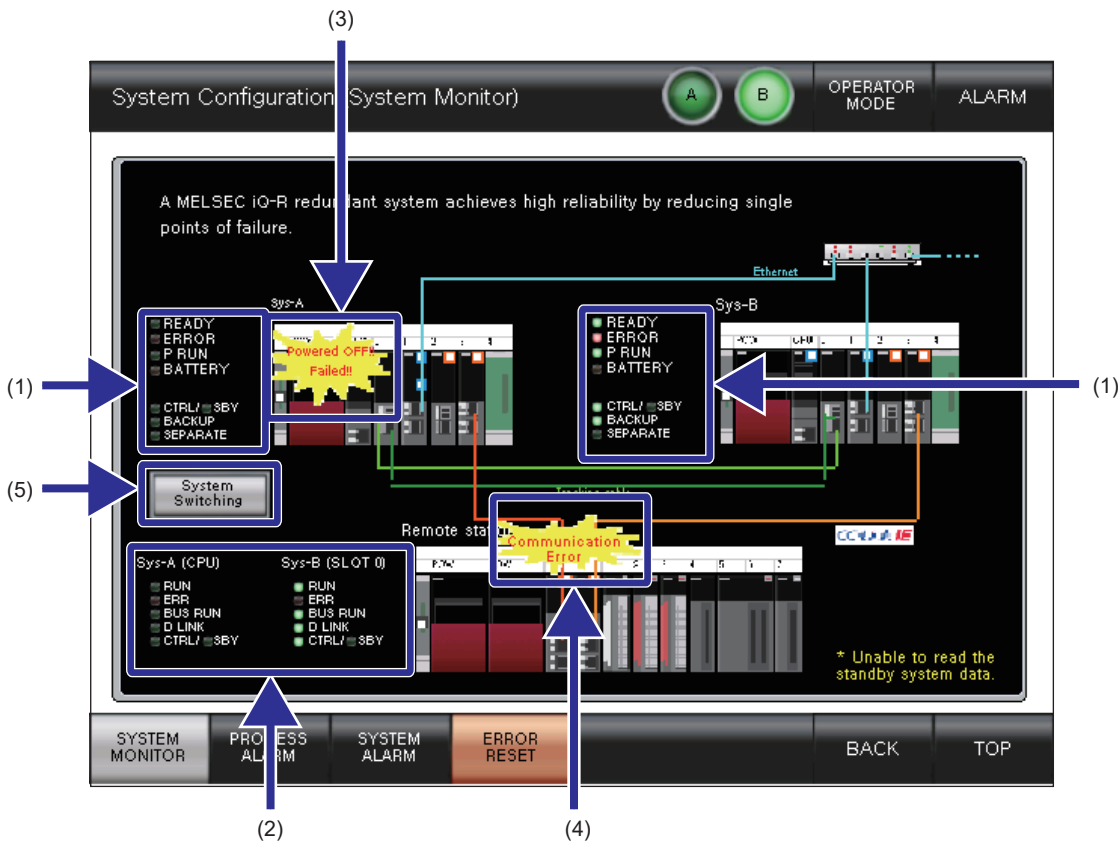
No.	Item	Description
(1)	CPU LED indication (master station)	The LED indication status of the following modules can be checked. • Process CPU • Redundant function module
(2)	CPU LED indication (slave station)	The LED indication status of the following module can be checked. • Remote head module
(3)	CPU error	Displayed when any of the following errors occur in the master station. • Process CPU stop error • Power-off/reset state
(4)	Network status	If any of the following network cannot be communicated with, this message is displayed on the network where an error was detected. • Ethernet • CC-Link IE Controller • CC-Link IE Field • Tracking
(5)	[System Switching] button	Switches the system.

Process control screen

This screen can be displayed by touching the [Process Control] button on the MENU screen.

Temperature control using a heater can be performed on this screen. The heater is equipped with a thermocouple and inputs a temperature measured value to CH1 of the R60TD8-G of the slave station and outputs a 24V signal from CH1 of the RY40PT5B to perform time proportional control.
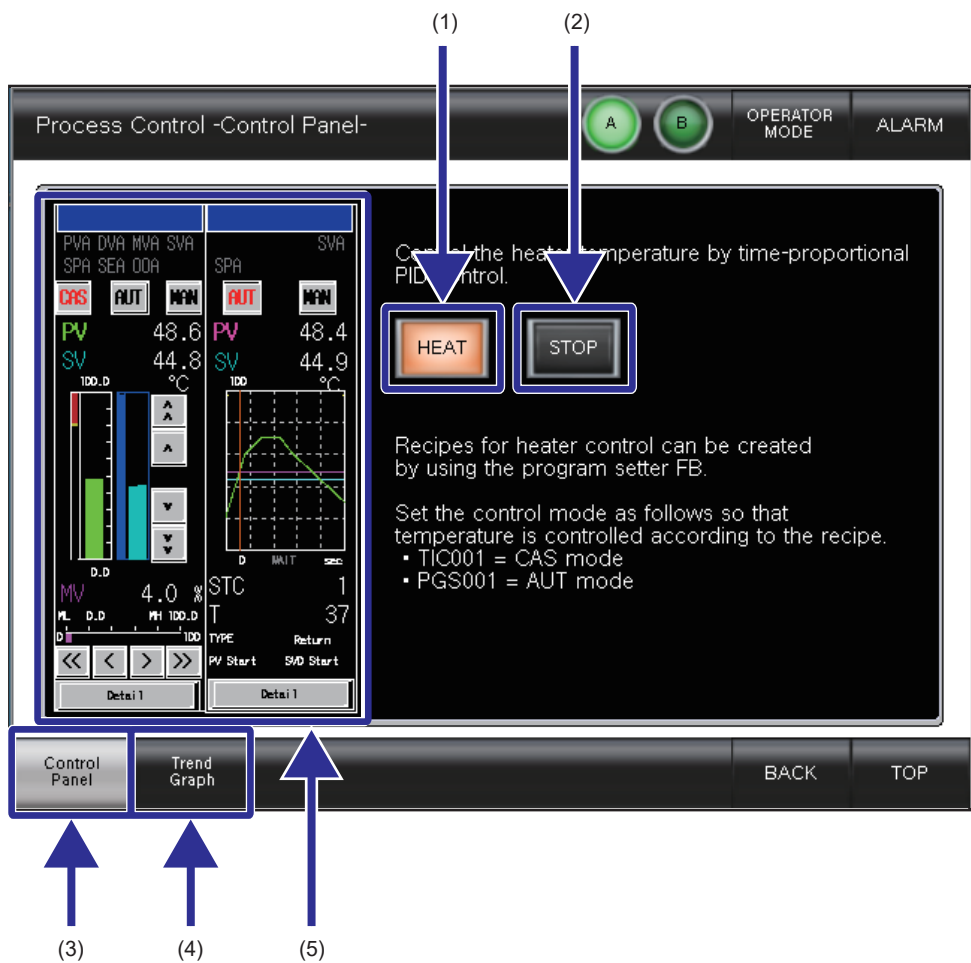
• Temperature control program

No.	Item	Description
(1)	[HEAT] button	Turns on the temperature control function. This switch must be turned on first before executing temperature control. This switch will be interlocked if any of the following events occur. • Temperature high high limit (TC temperature 80℃ or higher) • Communication error (Ethernet, CC-Link IE Field) • Thermocouple disconnected • During online module change (R60TD8-G, RY40PT5B)
(2)	[STOP] button	Turns off the temperature control function.
(3)	[Control Panel] button	Moves to the control panel screen. On the Control Panel screen, the temperature control function can be turned on/off, and the faceplate can be operated.
(4)	[Trend Graph] button	Moves to the trend graph screen. The temperature control trend graph is displayed on the trend graph screen.
(5)	Faceplate	The "TIC001" and "PGS001" faceplates are displayed. For details, refer to the following. ☞ Page 84 "TIC001" faceplate ☞ Page 85 "PGS001" faceplate

■"TIC001" faceplate

No.	Item	Description
(1)	Alarm display	Indicator of any alarm that occurs turns on
(2)	Mode selector	Switch the control mode (CASCADE, AUTO, MANUAL). The current mode is indicated in red. • CAS: CASCADE mode • AUT: AUTO mode • MAN: MANUAL mode
(3)	PV value	Displays the current PV value.
(4)	SV value	Displays the current SV value. The SV value can be changed in AUTO or MANUAL mode.
(5)	MV value	Displays the current MV value. The MV value can be changed in MANUAL mode.
(6)	[Detail] button	Moves to the TIC001 tag tuning screen.

Touch the [Detail] button on the "TIC001" faceplate to move to the tuning screen.

Touch the [SETTING] button to move to the setting screen, and touch the [TUNING] button to move to the tuning screen.

For details on the operation method, refer to the following.

📖 PX Developer Version 1 Operating Manual (GOT Screen Generator)
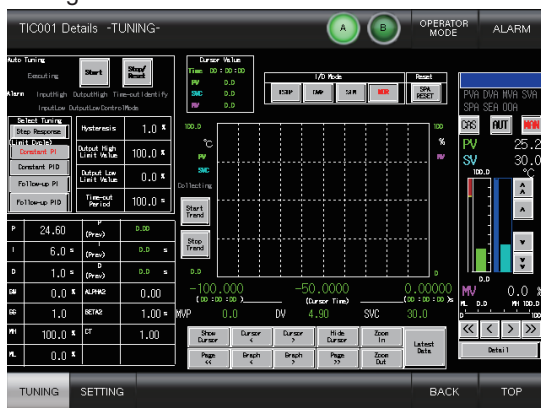
Tuning screen

Setting screen
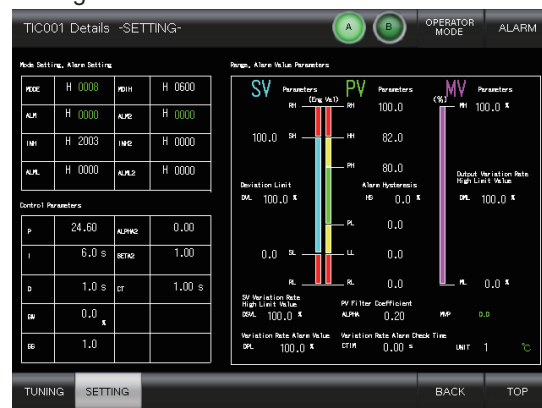
■"PGS001" faceplate
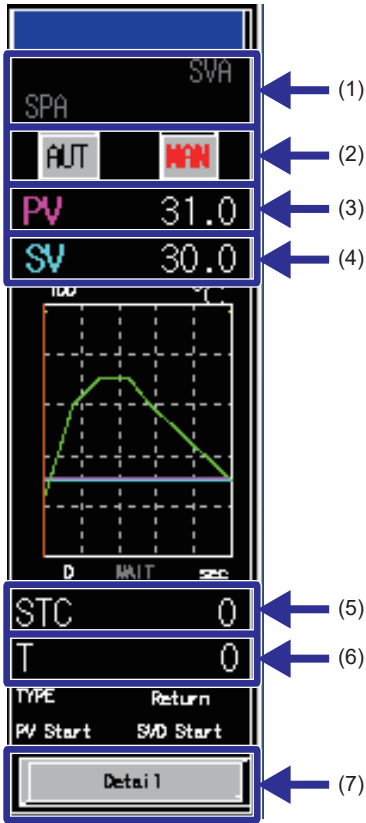
No.	Item	Description
(1)	Alarm display	Indicator of any alarm that occurs turns on
(2)	Mode selector	Switch the control mode (AUTO, MANUAL). The current mode is indicated in red. • AUT: AUTO mode • MAN: MANUAL mode
(3)	PV value	Displays the current PV value.
(4)	SV value	Displays the current SV value. The SV value can be changed in AUTO or MANUAL mode.
(5)	STC value	Displays/changes the current step.
(6)	T value	Displays/changes the current elapsed time.
(7)	[Detail] button	Moves to the PGS001 tag tuning screen.

Touch the [Detail] button on the "PGS001" faceplate to move to the tuning screen.

Touch the [SETTING] button to move to the setting screen, and touch the [TUNING] button to move to the tuning screen.

For details on the operation method, refer to the following.

📖 PX Developer Version 1 Operating Manual (GOT Screen Generator)
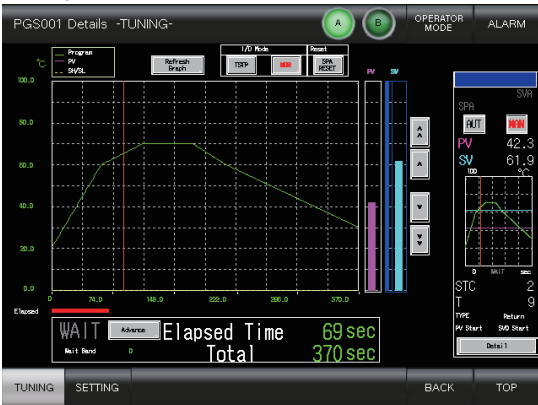
Tuning screen

Setting screen
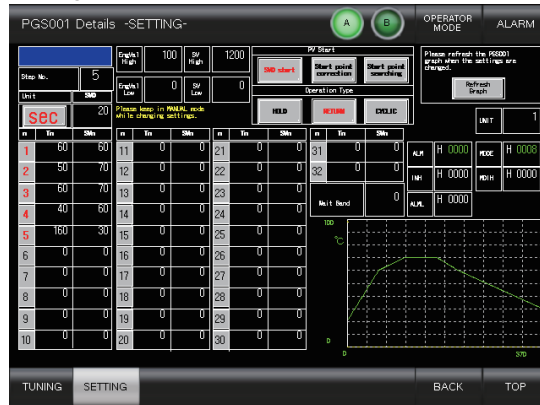
Tracking transfer screen

This screen is displayed by touching the [Tracking Transfer] button on the MENU screen.

On this screen, the tracking transfer capacity can be switched and each scan time can be measured.

The current scan time value is the tracking transfer time plus the program execution time.

No.	Item	Description
(1)	Tracking Transfer Capacity	Switches the tracking transfer capacity to the value indicated on the button. The currently selected setting is indicated in green.
(2)	Scan Time	Displays the current scan time.
(3)	In Q Series (Reference)	Displays a theoretical scan time of when transferring the current tracking capacity in the Q series.

Online module change screen

This screen is displayed by touching the [Online Module Change] button on the MENU screen.

On this screen, select the module to be replaced in the slave station and the procedure to replace that module is displayed.

No.	Item	Description
(1)	Head module LED indication	Displays the LED status of the remote head module.
(2)	Module button	Touch the module where an online module is to be replaced and the procedure to replace that module is displayed. The [NEXT] or [END] button is displayed at the bottom right of the replacement procedure screen. • Touch the [NEXT] button to display the subsequent procedure of replacement. • Touch the [END] button to finish the replacement. Touch the [CANCEL] button to return to the Online Module Change initial screen.

Auto memory copy screen

This screen is displayed by touching the [Auto Memory Copy] button on the MENU screen.

The auto memory copy operation is displayed on this screen. This function is useful when the actual module LED indicators cannot be visually checked.

During the memory copy, the MEMORY COPY LEDs of the redundant function modules flash.
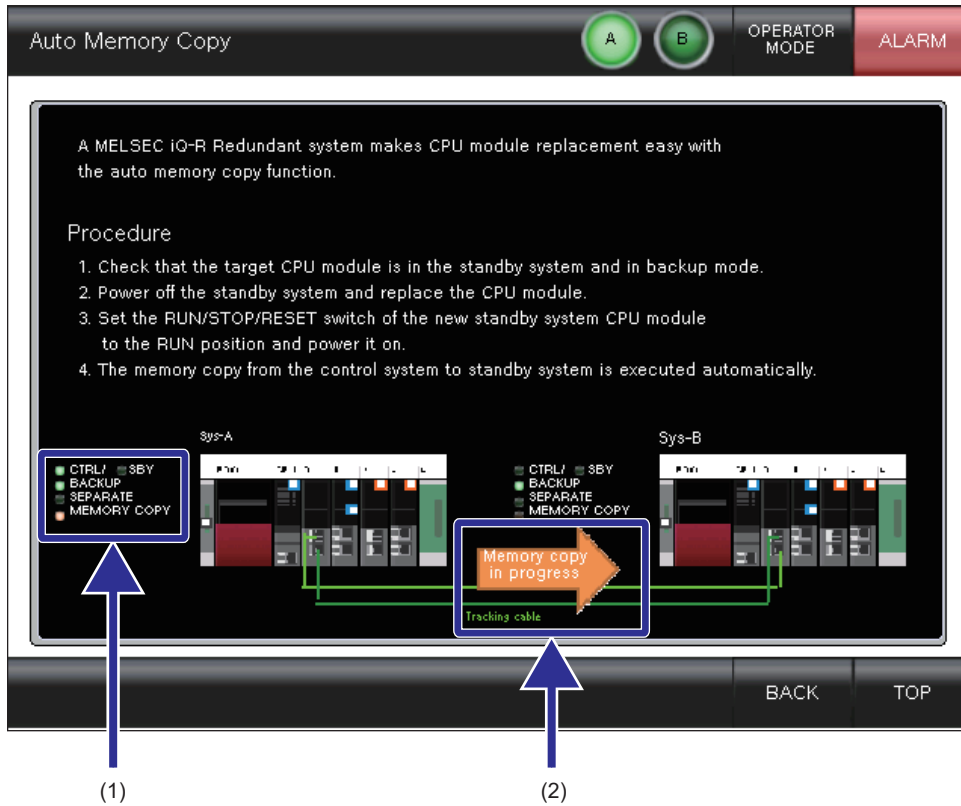
No.	Item	Description
(1)	CPU LED indication	Indicates the LED status of the CPU module.
(2)	Memory copy in progress	Indicates that memory copy is in progress.

Number of output ON times integration screen

This screen can be displayed by touching the [Number of Output ON Times Integration] button on the MENU screen.

On this screen, the number of output ON times of the output module (RY40PT5B) with diagnostic functions is counted.

For channels where the alarm is valid, an alarm is issued if the total value exceeds the threshold.
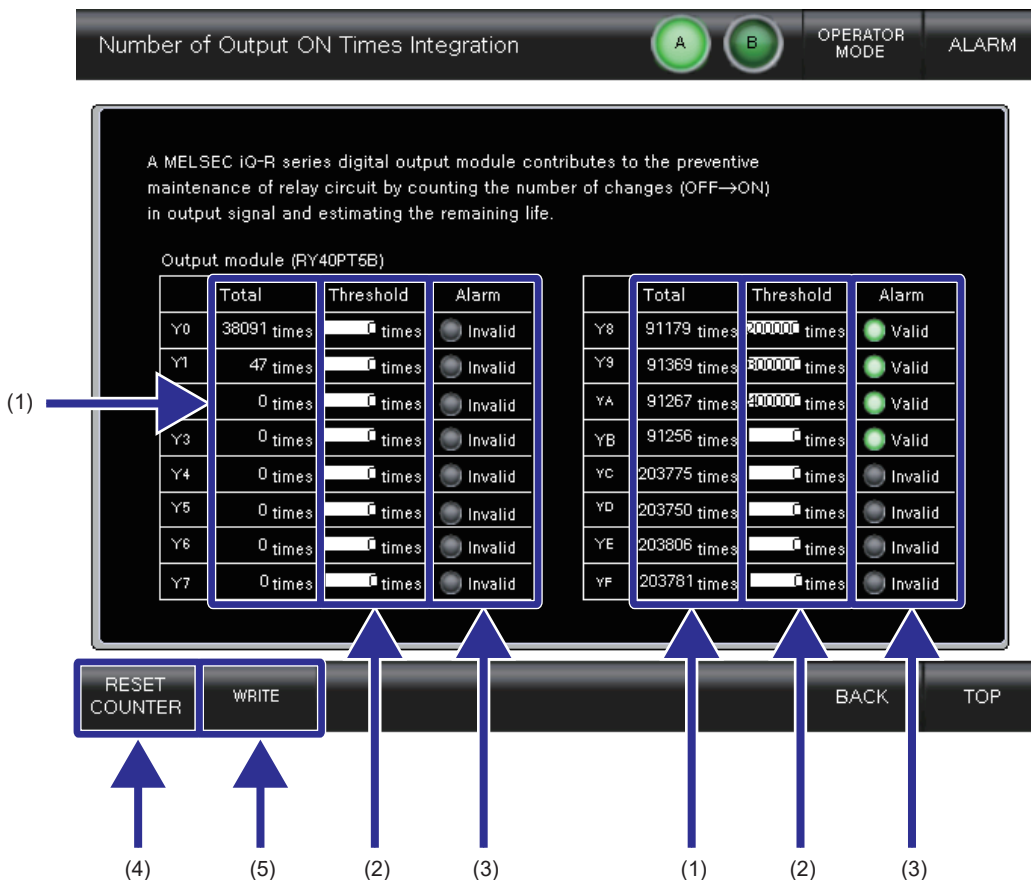
No.	Item	Description
(1)	Total	Displays the number of output ON times integration value for each channel.
(2)	Threshold	Set the threshold count for issuing an alarm for each channel. Touch the entry field to display the keypad and enter any value. * Touch the [WRITE] button to apply the changes to the programmable controller. Changes will not be applied if this button is not pressed.
(3)	Alarm	Switches the alarm status (valid/invalid) for each channel. Touch this area of each channel to switch the alarm status. * Touch the [WRITE] button to apply the changes to the programmable controller. Changes will not be applied if this button is not pressed.
(4)	[RESET COUNTER] button	Resets the total value of the selected channel.
(5)	[WRITE] button	Writes the settings of "Threshold" and "Alarm" to the programmable controller. * The settings of "Threshold" and "Alarm" will not be applied to the programmable controller until the [WRITE] button is touched.

5 CREATING A PROCESS CONTROL PROGRAM USING GX Works3

This section describes the procedure for creating a process control program for temperature control that use PID control using GX Works3 and the redundant system.

5.1 Operation

Create a process control program for temperature control that use PID control using GX Works3.

1. PID (two-degree-of-freedom advanced PID) control is used for temperature control to reach the target temperature set by the PGS (program setter). The MV value for PID control is converted to the ON/OFF signal by DUTY (time proportioning output).

2. Program operation values in the CPU module are output from the master station and input to the slave station via the CC-Link IE Field Network.

3. The heater temperature data is input to the temperature input module (R60TD8-G) in the slave station, and the ON/OFF signal is output from the output module (RY40PT5B) to the heater.

Temperature monitoring/setting/auto tuning
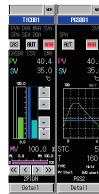
Faceplate (GX Works3, PX Developer monitor tool, GOT)

Master station

Slave station

MELSEC iQ-R redundant system

Input temperature data (PV) from the thermocouple to the R60TD8-G: CH1.

Output a manipulated value (MV) from the RY40PT5B: CH1 to the heater.

Thermocouple (demonstration machine heater)

Temperature range: 0 to 100°C
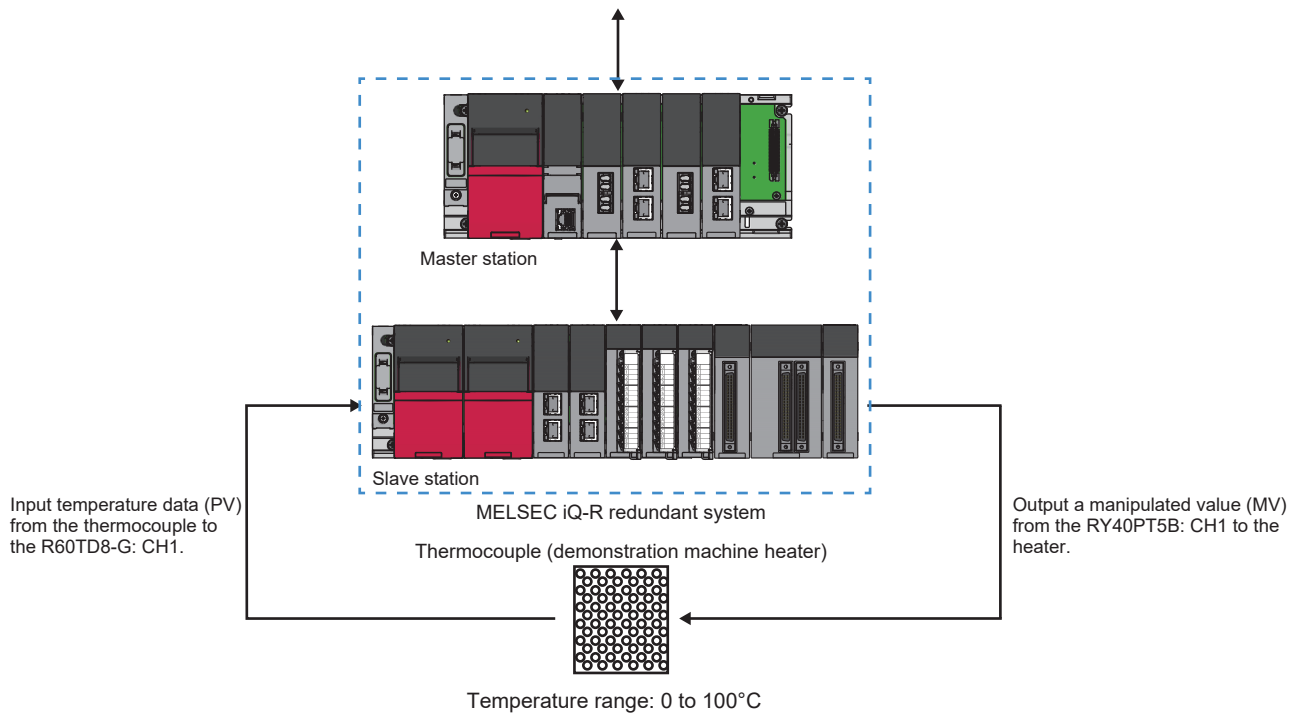
5.2 Setting Parameters

When creating a process control program, the file register to be used must be common to all programs.

Open the "School_Master_base" project, and set the file register.

> **Point**
>
> In this exercise, the "School_Master_base" project in which the redundant system parameters relevant to the demonstration machine are already set is used to create a process control project.

Operating procedure

1. Double-click "CPU Parameter" on the Navigation window.

1. Double-click

2. Select "File Setting"

3. Set the parameters
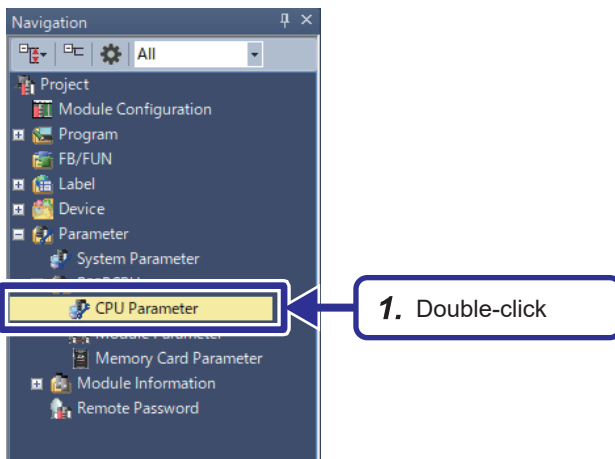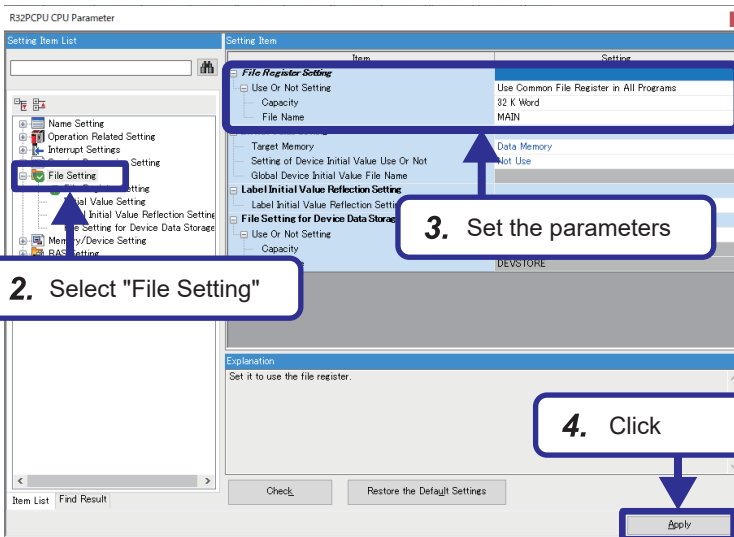
4. Click

2. Select "File Setting" on the list of setting items.

3. Configure the "File Register Setting" as follows.
[Settings]
Use Or Not Setting: Use Common File Register in All Programs
Capacity: 32 K Word
File Name: MAIN

4. Click the [Apply] button.

MELSOFT GX Works3

⚠ Are you sure you want to set the file register to the all points latch (2)?

5. Click → Yes | No
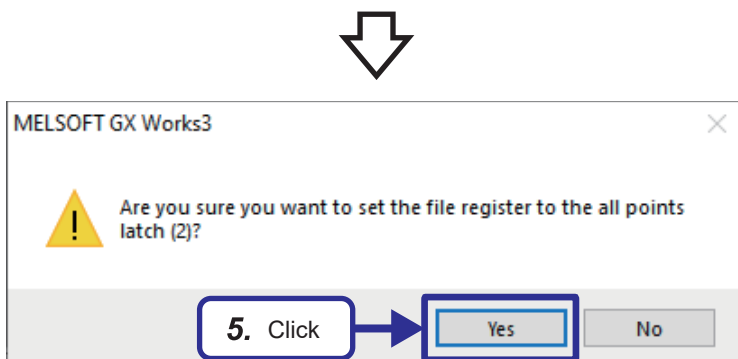
5. Click the [Yes] button.

Precautions
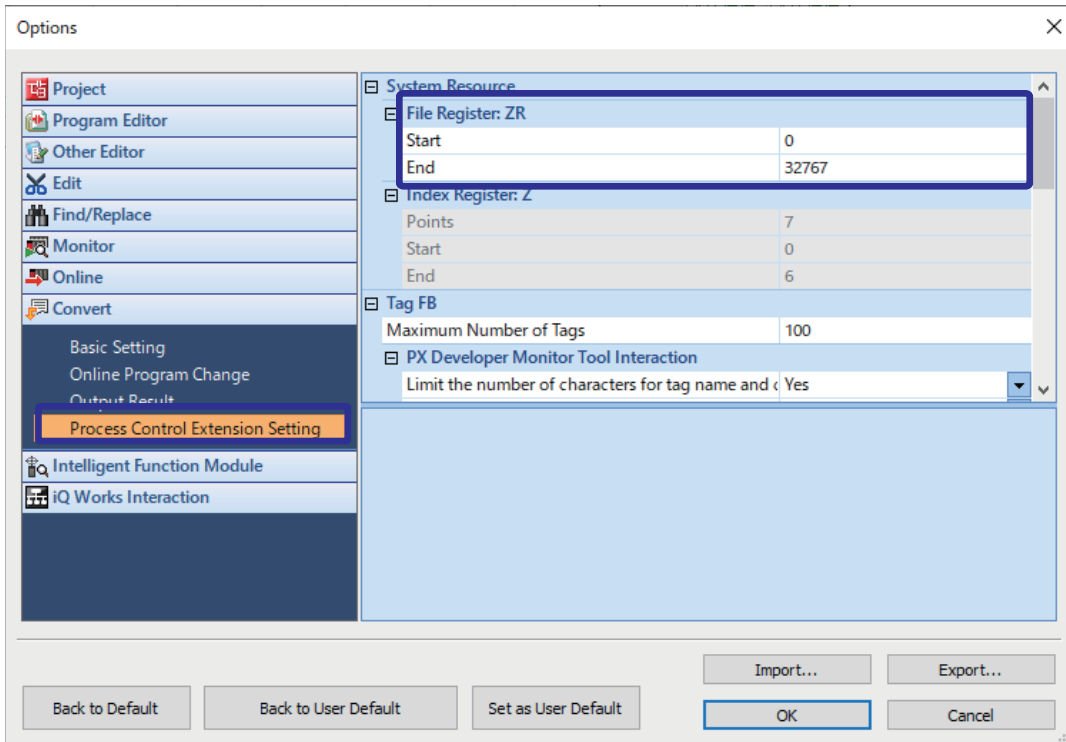
For School_Master_base, the process control program uses the 32K word file register, as shown in [Options] ⇨ [Process Control Extension Setting] ⇨ [System Resource] ⇨ [File Register: ZR].

Therefore, set the file register capacity to 32K words or more.

5.3 Registering Tag FBs

Register tag FBs to be used in the process control program.

Tag FBs are function blocks containing tag data (such as PV, SV, MV, P, I, D) for process control. Each tag data is assigned 130 words starting from the start device registered in the tag FB setting.

Operating procedure
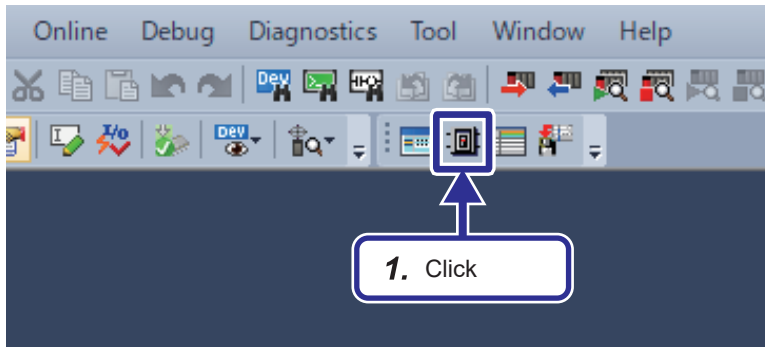
1. Click the [Tag FB Setting] icon (![icon]) on the toolbar.

> **Point**
>
> Select [View] → [Toolbar] → [Process Control Extension] to display the button.
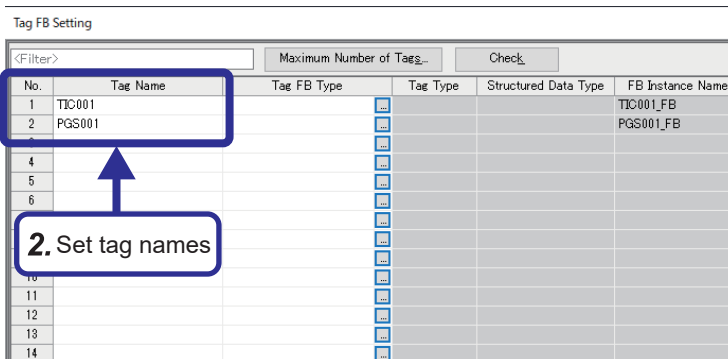
2. Set the tag names as follows.
[Settings]
No.1: TIC001
No.2: PGS001
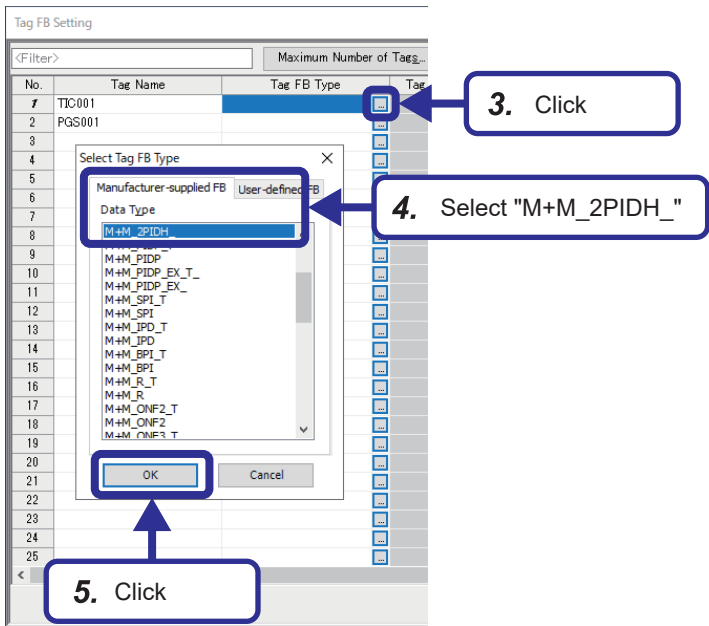
3. Click the ⊡ button in the "Tag FB Type" column for No.1.

4. When the tag FB type selection window is displayed, open the "Manufacturer-supplied FB" tab and select "M+M_2PIDH_".

5. Click the [OK] button.

6. In the same way, click the ⊡ button in the "Tag FB Type" column for No.2, and select "M+M_PGS2_".
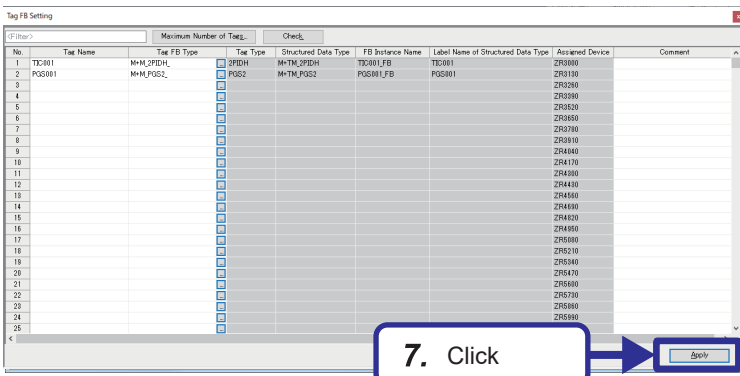
7. Click the [Apply] button.
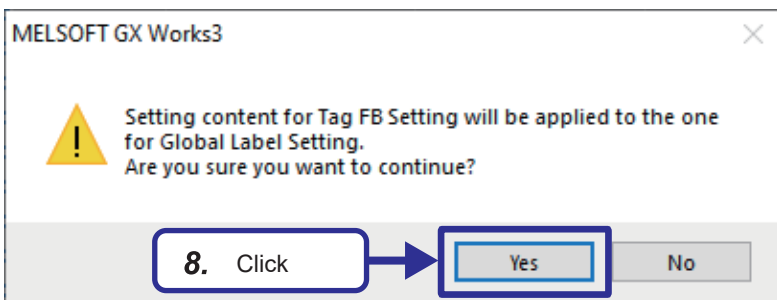
8. Click the [Yes] button.
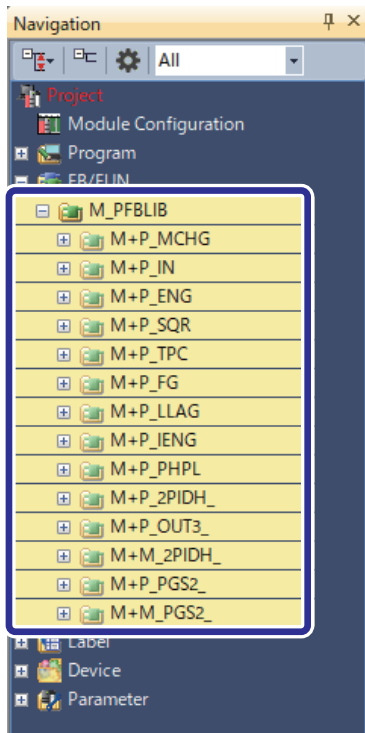
9. The tag FBs are registered on the Navigation window.

Point

When "M+M_2PIDH_" and "M+M_PGS2_" are registered, function blocks referenced by these tag FBs are also added.

5.4 Programming

Create a process control program using the FBD language.

Creating a new program file

Create a new program file. In this exercise, a scan-type FBD program file will be created.
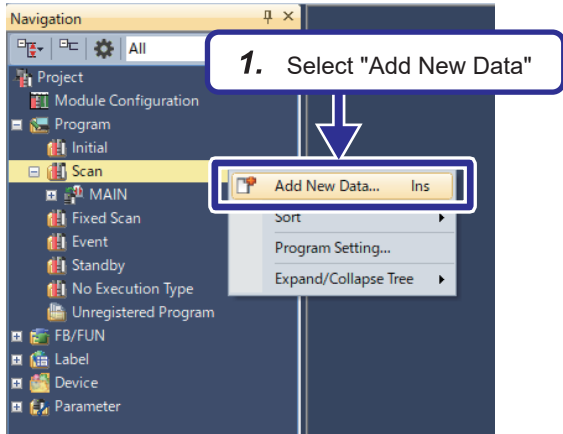
Operating procedure

1. Right-click "Scan" on the Navigation window, and select "Add New Data".
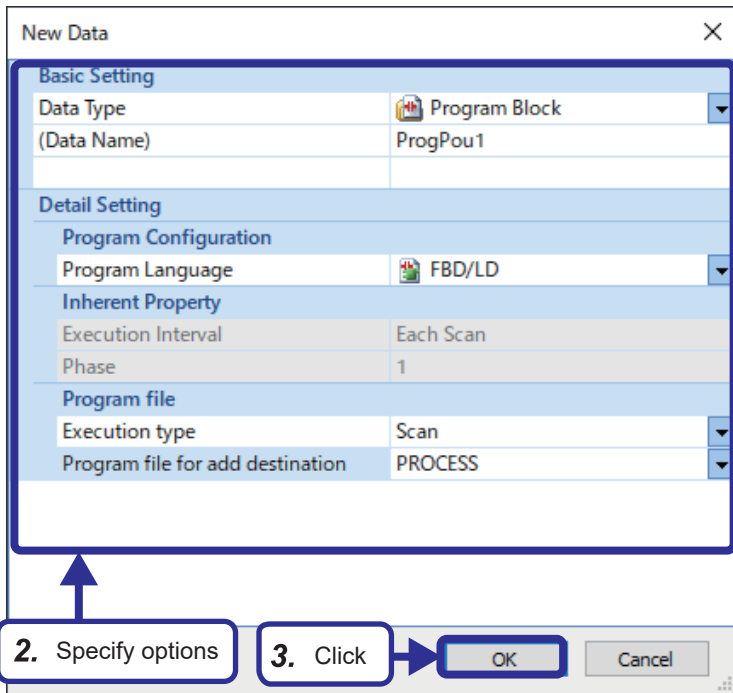
2. The "New Data" window appears. Configure the settings as follows.
   [Settings]
   Data type: Program Block
   (Data Name): Any name
   Program Language: FBD/LD
   Execution type: Scan
   Program file for add destination: New file name

3. Click the [OK] button.
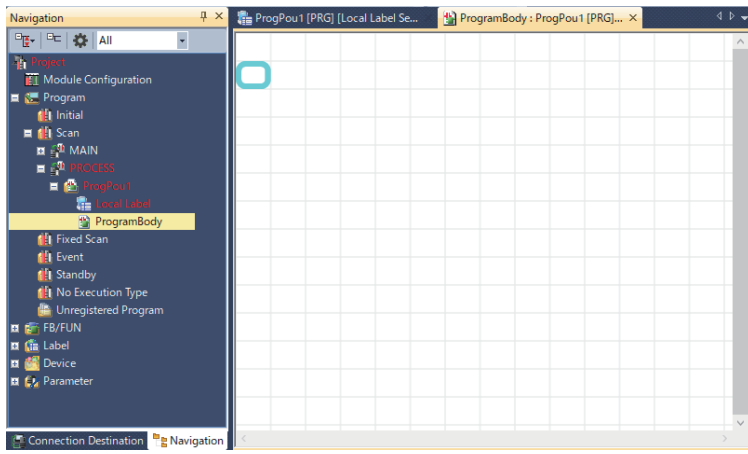
4. A new program file is created on the Navigation window, and the sheet is displayed.
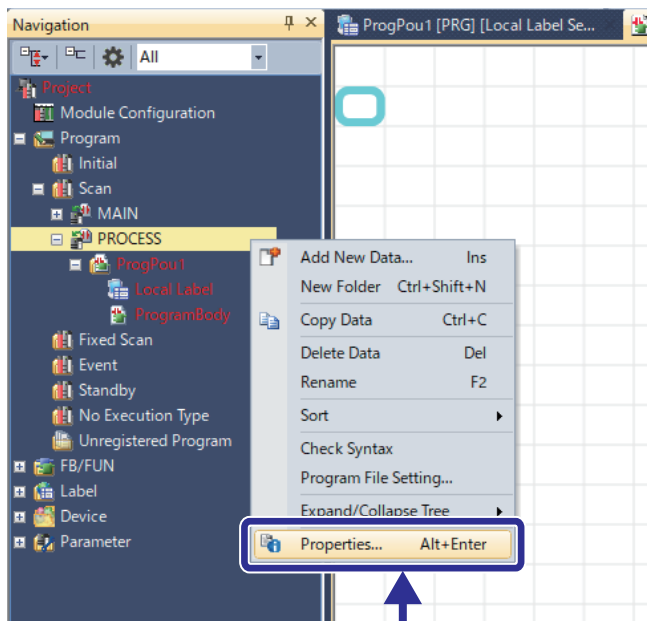
5. Right-click the created program file on the Navigation window, and select "Properties".

5. Select "Properties"
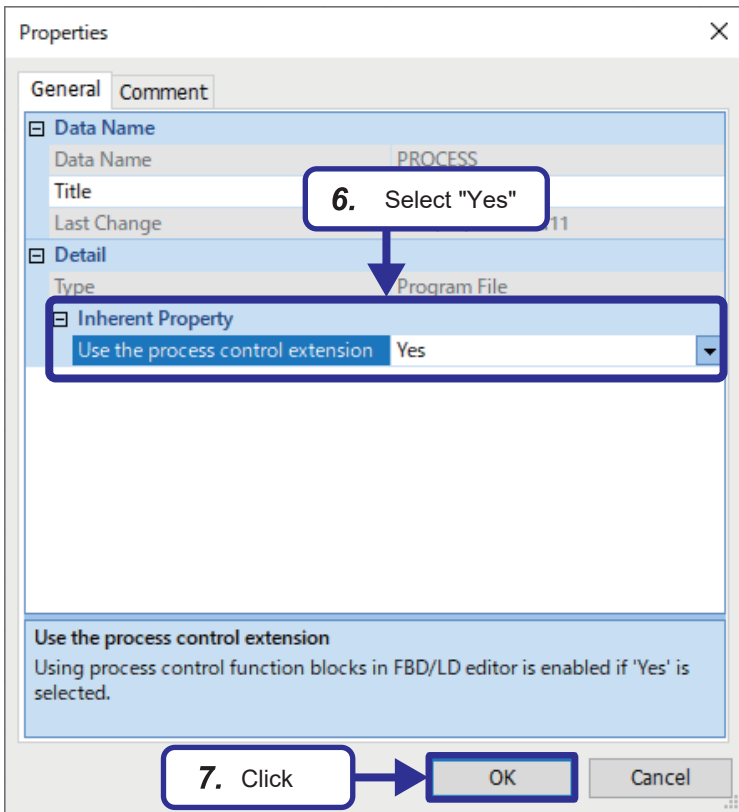
6. The "Properties" window appears. Select "Yes" for "Use the process control extension".

7. Click the [OK] button.
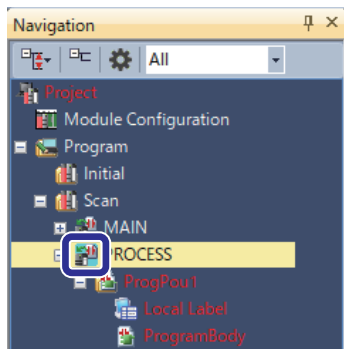
8. For a program file with the process control extension enabled, the icon background is displayed in light blue.

Point

Only FBD/LD programs can be used for process control programs.

Also, for the scan execution type, only one process control program file can be created.

Programming using the FBD language

This section describes how to create a program in the FBD language using GX Works3.
This example will describe the procedure for creating a program in the same way as creating a temperature control program that uses PID control.

Adding variable elements

Operating procedure
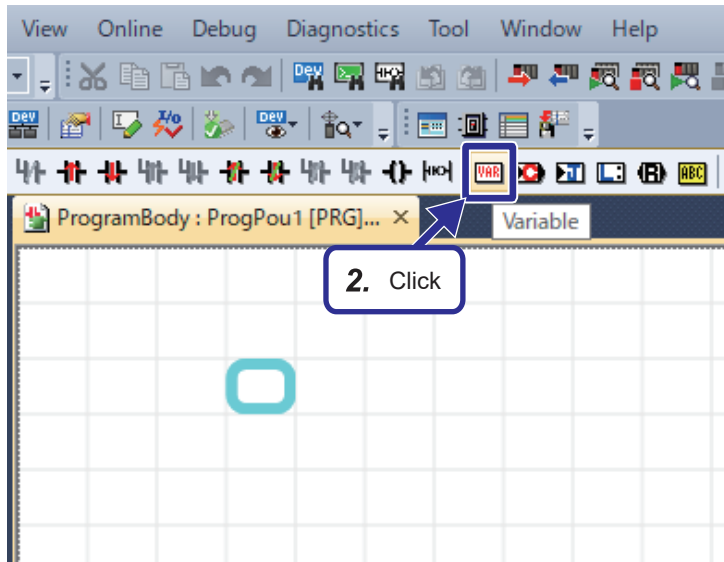
1. Move the cursor to a position where a program element is to be added.

2. Click the [Variable] icon ( ) on the toolbar.

> **Point**
>
> Or, select [Edit] ⇨ [Add Element] ⇨ [Variable] from the menu bar.
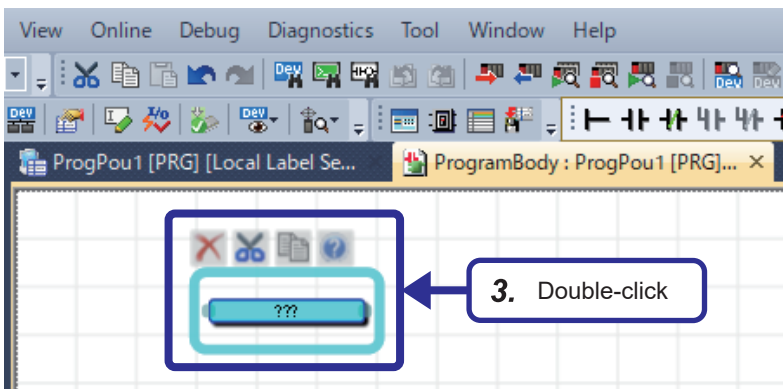
3. The program element appears at the cursor position.
Double-click the added program element.
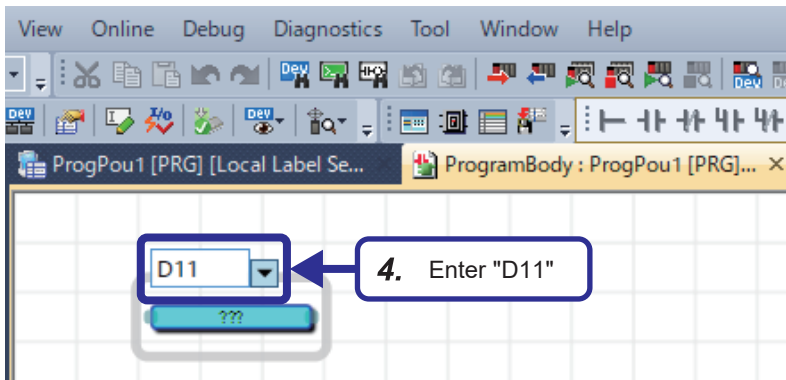
4. Enter "D11" as the device name.
"D11" will be used as the storage location of temperature data input from the thermocouple.
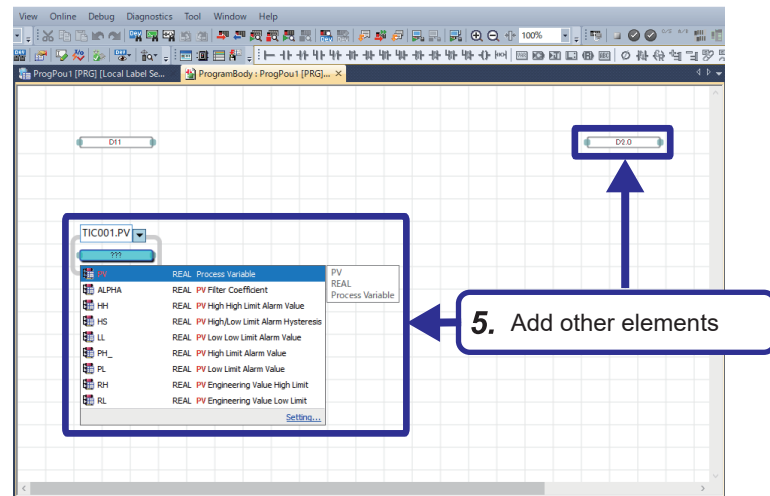
5. Repeat steps **1.** through **3.** to add variable elements "D2.0" and "TIC001.PV". "D2.0" is the output to the heater, and "TIC001.PV" is the label assigned to TIC001 measurement value (PV).

Point

Both a device name and a label name can be used for a variable element.

Adding a tag FB element

Operating procedure
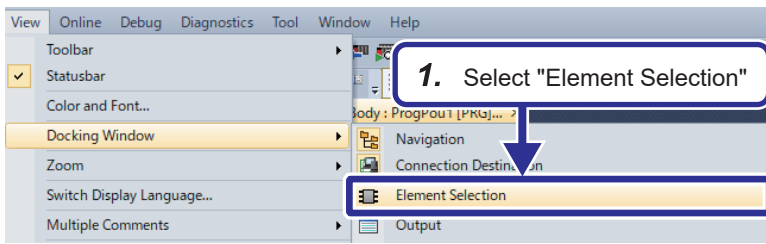
1. Select [View] ⇨ [Docking Window] ⇨ [Element Selection] on the menu of the engineering tool.
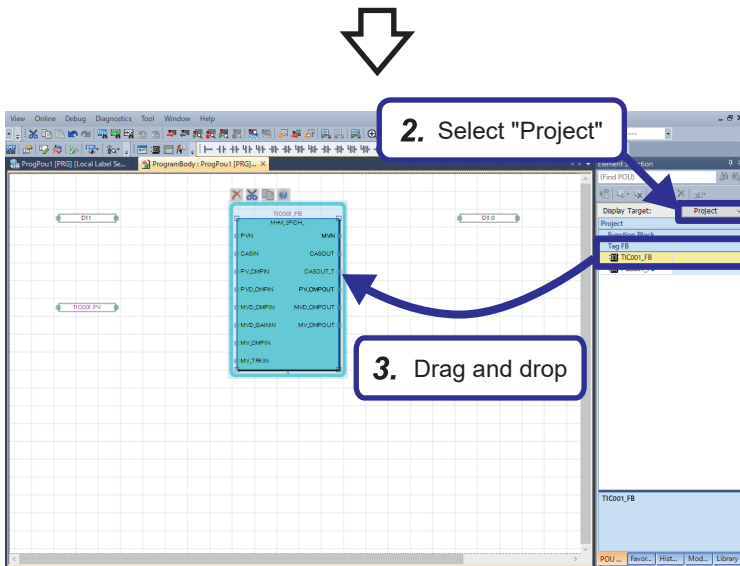
2. When the Element Selection window is displayed, select "Project" from the "Display Target" drop-down menu.

3. Select "TIC001_FB" from the "Tag FB" section, and drag and drop it to the sheet.
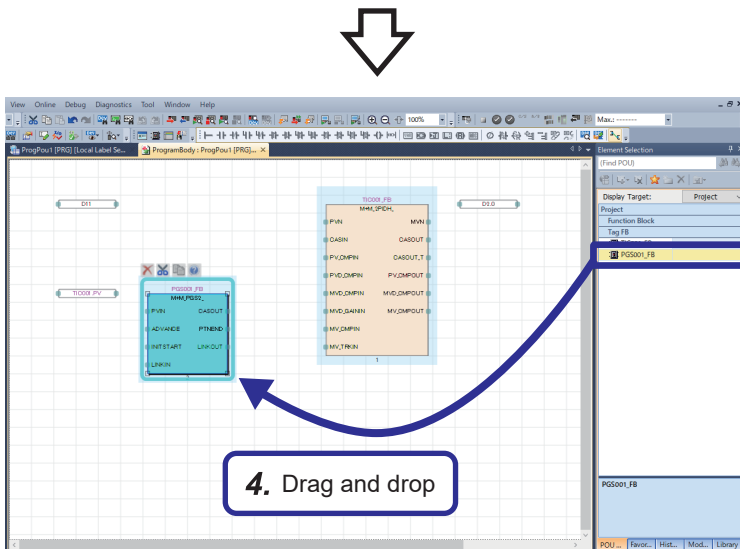
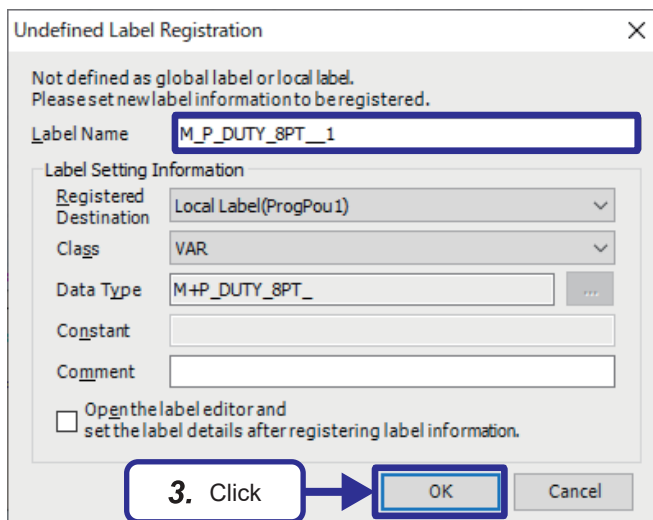4. In the same way, select "PGS001_FB" and drag and drop it to the sheet.

Preparing other FB elements

■Process control FB element

Operating procedure
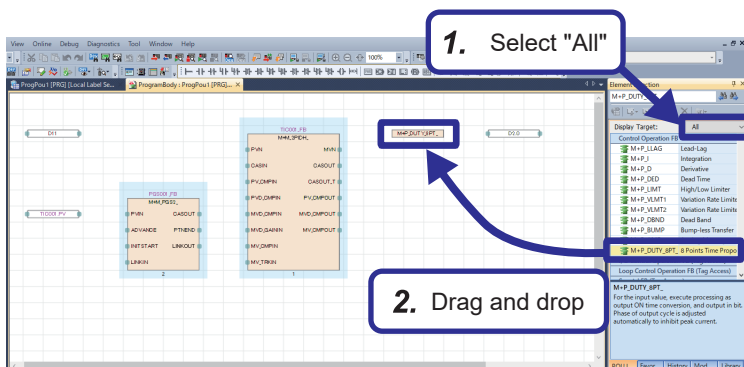
1. On the Element Selection window, select "All" from the "Display Target" drop-down menu.

2. Select "M+P_DUTY_8PT_" from the "Control Operation FB" tab under "Process Control Function Block", and drag and drop it to the sheet.

3. The "Undefined Label Registration" window appears. Enter a label name and click the [OK] button.
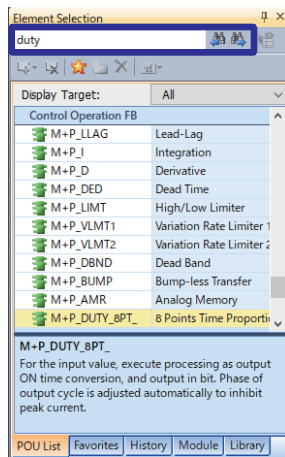
■Standard FB element

Operating procedure
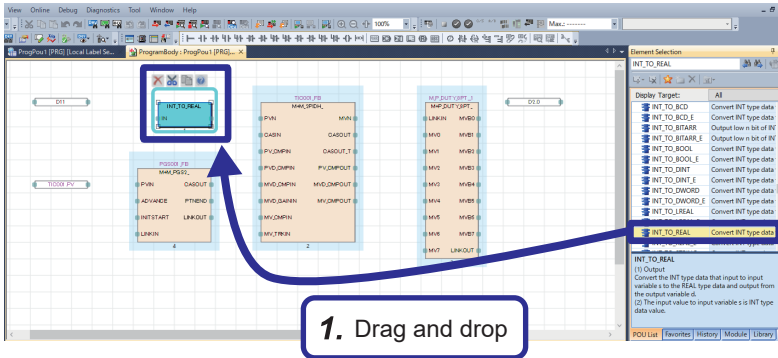
1. Drag and drop

1. Select "INT_TO_REAL" on the "Type Conversion" tab under "Standard Function/ Function Block", and drag and drop it to the sheet.
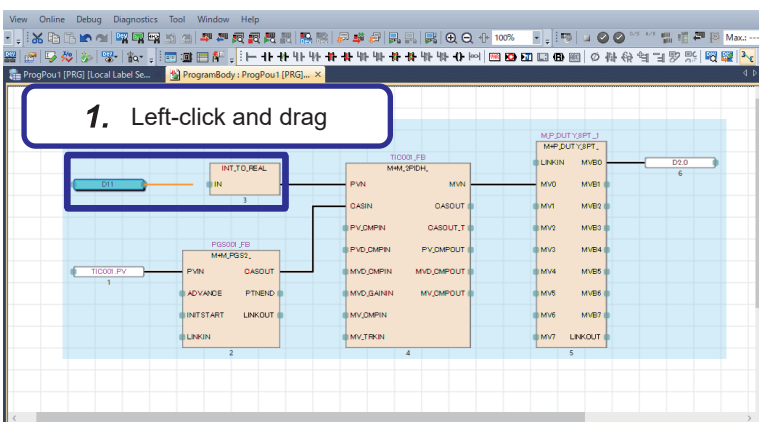

Point

Function blocks can be searched by entering any character string in the search box on the Element Selection window and clicking the [Previous]/[Next] icon.

Connecting elements

Operating procedure

1. Left-click and drag

1. Connect pins of each program element as shown in the left figure.
Left-click an area between the pins and drag to connect the pins.

Setting tag FB parameters

Set the tag FB parameters using GX Works3. Tag FB parameters are called FB properties, and the initial values can be set on the FB Property window.

FB property initial value setting example

Set the FB property initial values as shown in the following setting example. The following are the minimum parameters required for temperature control for the demonstration machine.

- FB property initial value setting example for TIC001

Label name (item)	Description		Setting value
IN_NMAX (Input high limit)	Analog input	Set the high limit for the range (0 to 8000) of the A/D conversion value to be input from the analog input module.	1000.0
IN_NMIN (Input low limit)		Set the low limit for the range (0 to 8000) of the A/D conversion value to be input from the analog input module.	0.0
IN_HH (High limit range error occurrence)		Set the value of the high limit range error for the range of the A/D conversion value to be input from the analog input module. A sensor error (SEA) occurs if the A/D conversion value exceeds this value.	1200.0
IN_H (High limit range error return)		Set the value of error reset after the high limit range error. The sensor error (SEA) is reset if the A/D conversion value falls to or below this value.	1000.0
IN_L (Low limit range error return)		Set the value of error reset after the low limit range error. The sensor error (SEA) is reset if the A/D conversion value exceeds this value.	0.0
IN_LL (Low limit range error occurrence)		Set the value of the low limit range error for the A/D conversion value to be input from the analog input module. A sensor error (SEA) occurs if the A/D conversion value drops to or below this value.	-50.0
PID2H_SVPTN_B0 (Setting value SV used)	Cascade connection	Set whether to use cascade connection secondary-side loop. TRUE: Secondary-side loop not used FALSE: Secondary-side loop used	FALSE
UNIT (Unit number)	Monitor tool display setting	Set the unit number for tag data to be displayed as engineering value on the monitor tool. Specify a number corresponding to any unit string registered on the "Unit Setting" window of the monitor tool. "1" corresponds to "℃".	1
PROPORTIONAL (Gain)	PID constant	Set the proportional gain for P operation. Specify the proportional gain, and not the proportional band. Proportional gain = 100/proportional band%	21.0
INTEGRAL (Integral time)		Set the integral time for I operation.	13.0
DERIVATIVE (Derivative time)		Set the derivative time for D operation.	0.2

- FB property initial value setting example for PGS001

Label name (item)	Description		Setting value
UNIT (Unit number)	Monitor tool display setting	Set the unit number for tag data to be displayed as engineering value on the monitor tool. Specify a number corresponding to any unit string registered on the "Unit Setting" window of the monitor tool. "1" corresponds to "℃".	1
STNO (Number of steps setting)	Break point	Set the total number of steps in a single program.	5
SV0 to SV5 (Setting value of step n)		Set the setting value of each step. SV0 is the setting value at start.	SV0: 20 SV1: 60 SV2: 70 SV3: 70 SV4: 60 SV5: 20
T1 to T5 (Time span of step n)		Set the time width of each step.	T1: 60 T2: 30 T3: 50 T4: 30 T5: 160

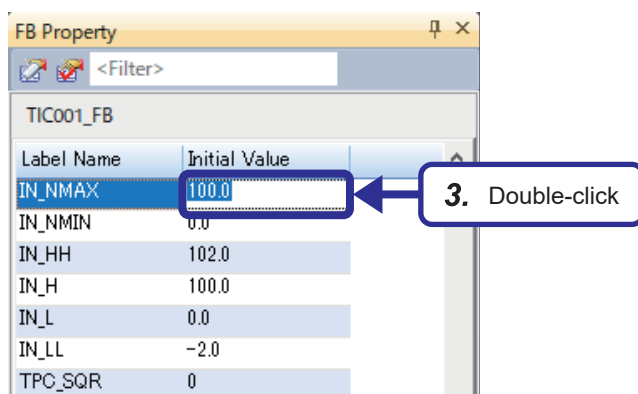## Procedure for setting the FB property initial values

Operating procedure
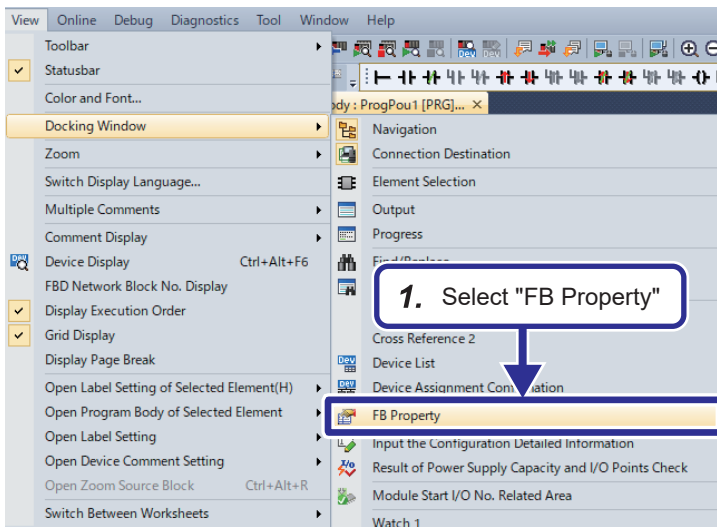
1. Select [View] ⇨ [Docking Window] ⇨ [FB Property] on the menu of the engineering tool.
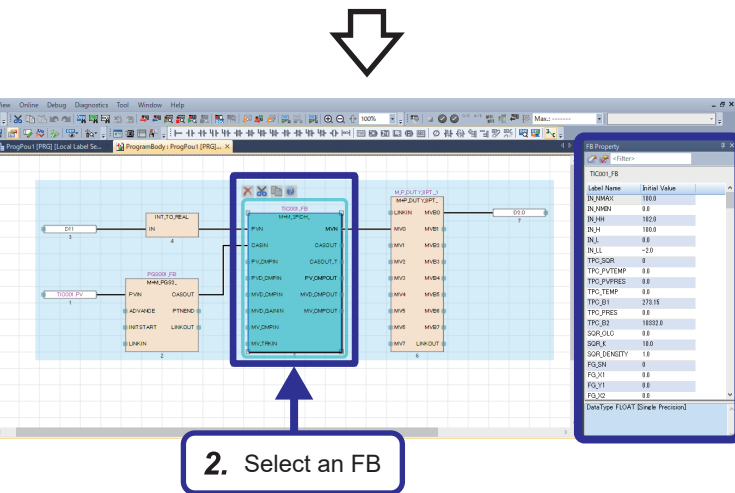
2. The FB properties window is displayed. When an FB is selected, the list of properties of the selected FB is displayed on the FB properties window.
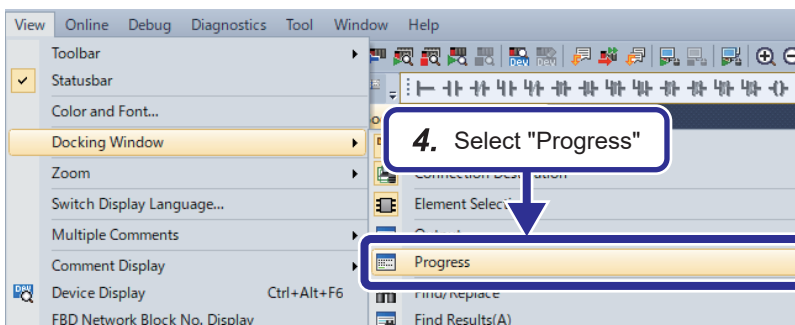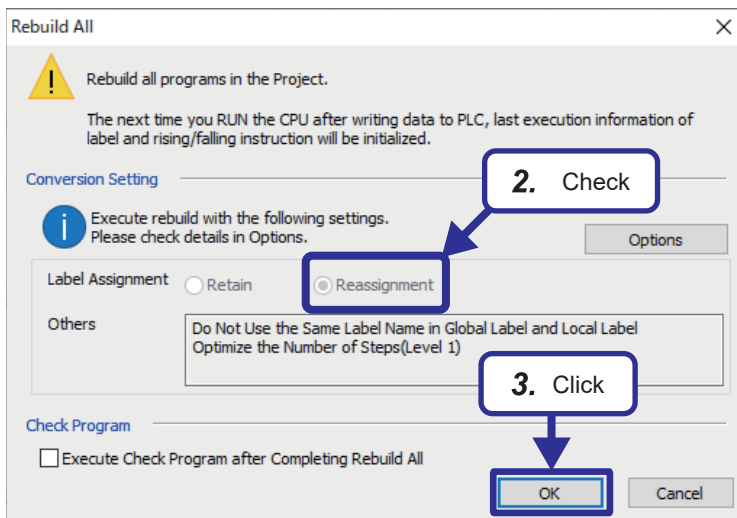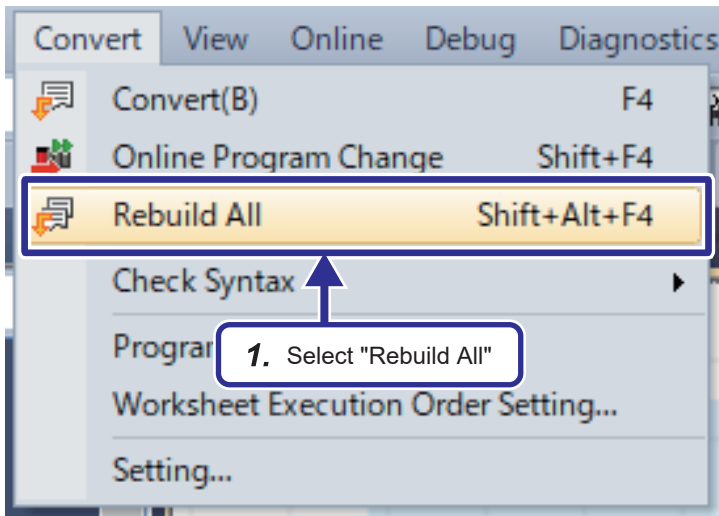

3. Double-click an initial value on the FB properties window to edit the initial value. Edit the properties as shown in "Page 104 FB property initial value setting example".

Converting the program

Convert a created process control program.

1. Select [Convert] ⇨ [Rebuild All] from the menu of the engineering tool.

1. Select "Rebuild All"

2. Check

3. Click

2. The "Rebuild All" window appears. Check that "Label Assignment" is set to "Reassignment".
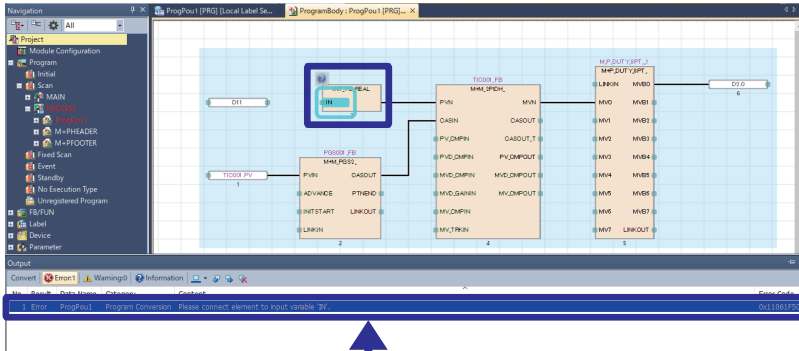
Point

If it is set to "Retain", click the [Options] button, and set "Reassign Labels in Executing Rebuild All" to "Yes".

3. Click the [OK] button.
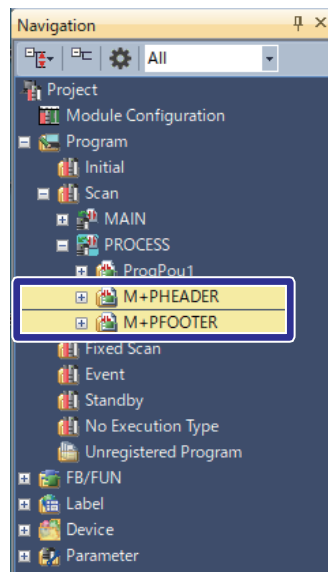

4. Select "Progress"

4. Select [View] ⇨ [Docking Window] ⇨ [Progress] from the menu of the engineering tool.

5. The "Progress" window appears. Check that all conversion processes are complete and that there are no errors.

6. If an error has occurred, check and fix the error from the output window ([View] → [Docking Window] → [Output]).
Double-click the error message to jump to the error section.
For warnings, check the content and fix only if there is any problem.

6. Double-click

7. When compilation is performed for the first time after enabling the process control extension setting, the two program blocks "M+PHEADER" and "M+PFOOTER" are created automatically.

8. Save the project with the following name.
File name: School_Master.gx3

Precautions

"M+PHEADER" and "M+PFOOTER" are system files used for process control. Do not edit or delete them, and also do not change their execution order.
Otherwise, process control may not operate normally.

Point

When a project in process CPU redundant mode is converted, the process control file register ZR is automatically registered to the tracking block No.64 of the tracking transfer setting.

5.5 Writing a Project

Write the created project "School_Master" to the process CPU.

Writing data to the programmable controller

Write the "School_Master" project from GX Works3 to each Process CPU of the master station system A and system B.
Write the "School_Remote" project in which parameters are configured to the remote head module of the slave station.

> **Point**
>
> • To delete the content of Master_forGOT.gx3 written in "☞ Page 77 Operating the Demonstration Machine", delete data in the CPU module from the "Online Data Operation" window in advance.
> • To write a file register, the file register capacity needs to be secured.
>   Write the CPU parameters in advance, and power off and on or reset the programmable controller.
>   Then, write the file register.
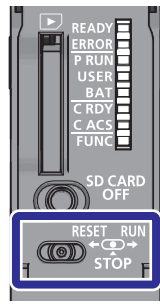
Operating procedure
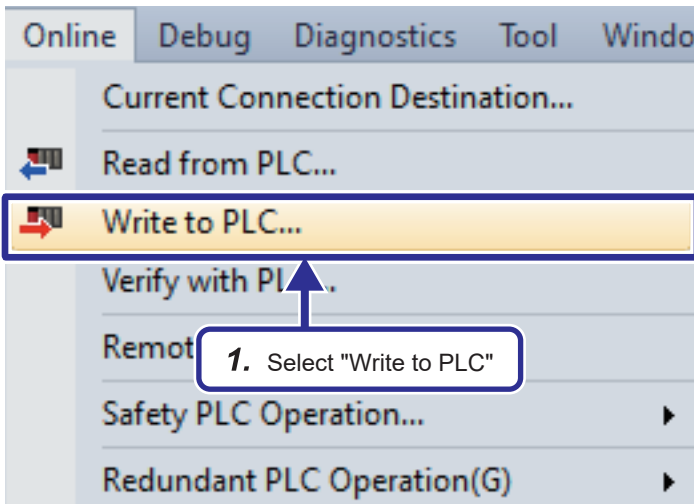
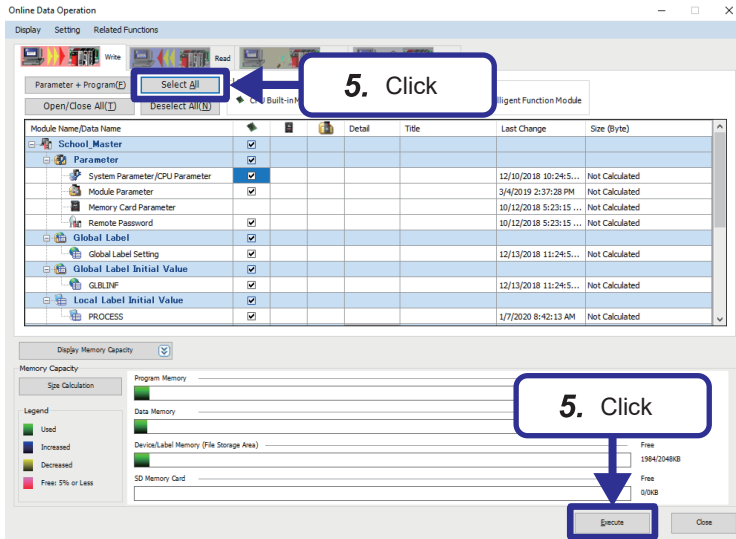**1.** Select [Online] ⇨ [Write to PLC] from the menu of the engineering tool.
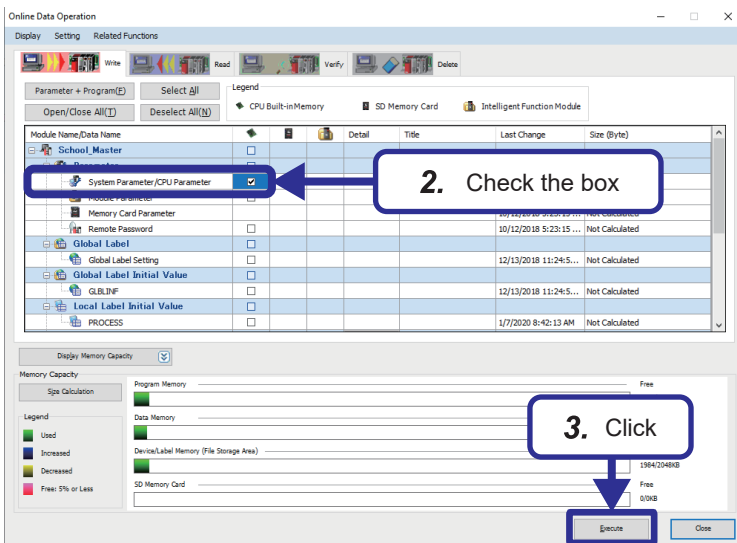
2. The "Online Data Operation" window appears. Select "System Parameter/CPU Parameter" as data to be written.

3. Click the [Execute] button.

4. Set the RUN/STOP/RESET switch of the Process CPU to the RESET position once (for approximately 1 second), and then set it to the STOP position.
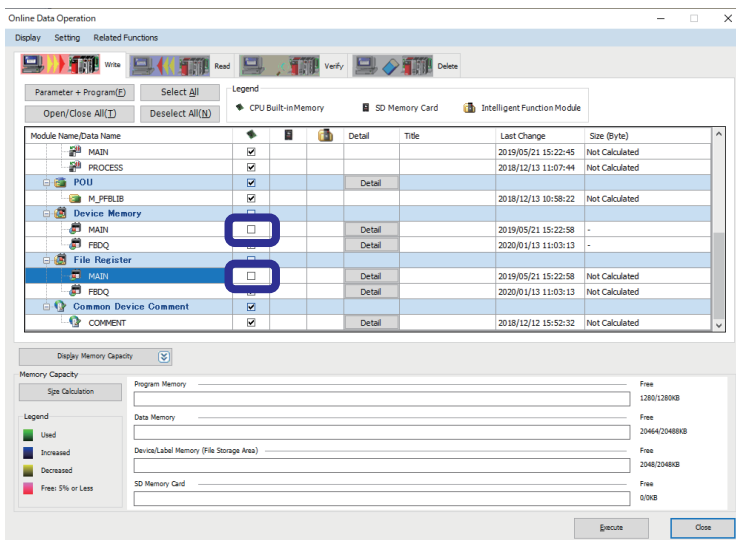

5. Click the [Select All] button to place check marks for all items to be written.

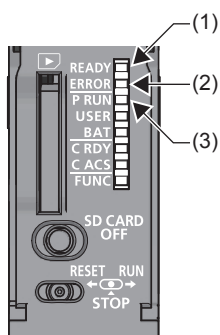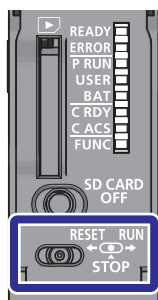**6.** Click the [Execute] button to write the data.

Precautions

When the above steps are complete, two file registers and two device memory areas may be registered.

In this case, uncheck the file register and device memory area that are not used.

Executing data link

Operating procedure

1. Set the RUN/STOP/RESET switch of the Process CPU to the RESET position for 1 second or more, and then set it to the RUN position.

2. Operate the power switch of the demonstration machine to power off both systems. Set the RUN/STOP/RESET switch of the CPU module of each system to the RUN position and power on both systems.

3. Check the operation with the LEDs of each module.
   Also check that the LED status of the Process CPU is as follows.
   (1) READY LED: On
   (2) ERROR LED: Off
   (3) P RUN LED: On (system A only)

Point

Individually setting the RUN/STOP/RESET switch of each CPU module to the RUN position with the CPU module powered on causes a continuation error due to the operating status mismatch and the error is detected in the standby system. After powering off both systems, set the RUN/STOP/RESET switch of the CPU module of each system to the RUN position and power on both systems.

6 MONITORING THE PROCESS CONTROL SYSTEM

6.1 Monitoring with GX Works3

GX Works3 provides simple monitoring using a faceplate and allows FB properties to be changed online.
This section will explain the procedure using the example where the temperature control monitor and PID are manually adjusted from the faceplate.

Monitoring using the faceplate

Start program monitoring on GX Works3 and display the faceplate.

Operating procedure

1. Select "Start Monitoring (All Windows)"
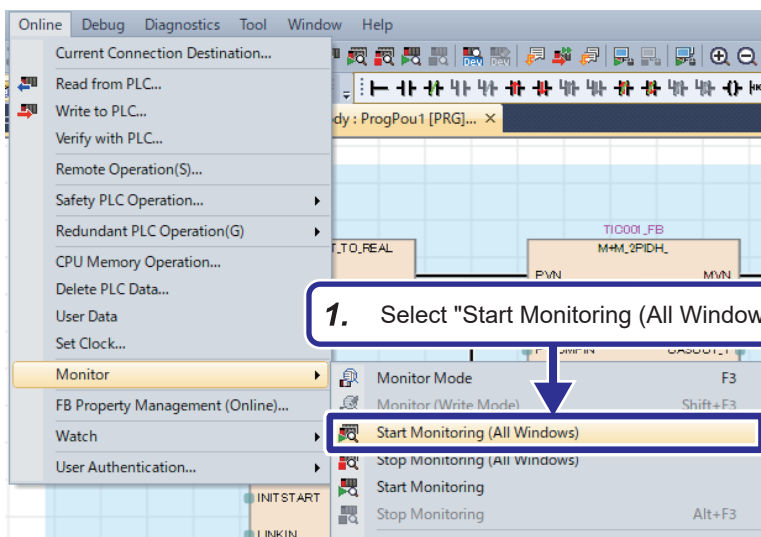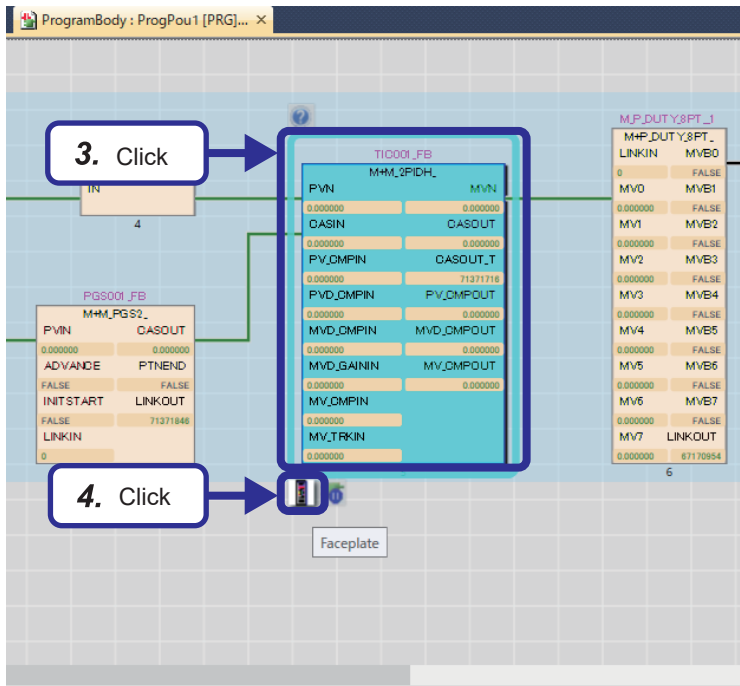
1. Select [Online] ⇨ [Monitor] ⇨ [Start Monitoring (All Windows)] from the menu of the engineering tool.
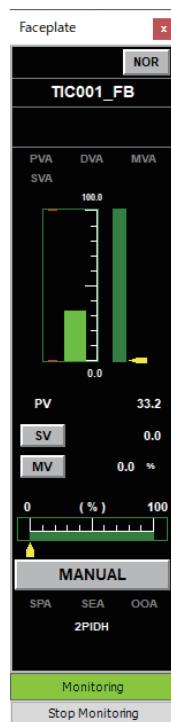

2. Program monitoring starts.
"Monitoring (Read Only)" is displayed at the top of the window.

3. When TIC001_FB is clicked, the [Faceplate] button is displayed at the bottom left.

4. Click the [Faceplate] button.
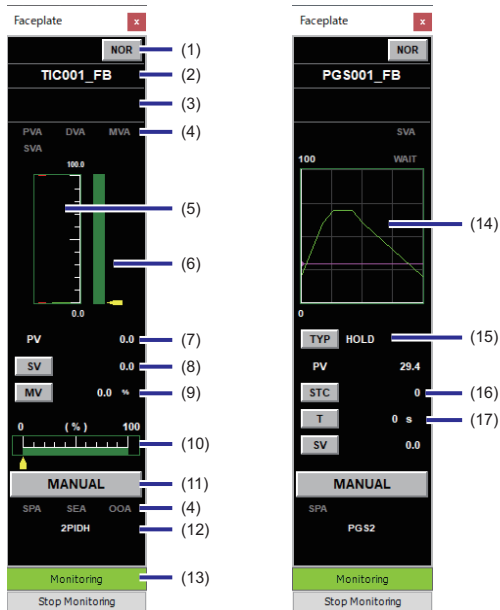
5. The faceplate is displayed.

Faceplate

The faceplate is a window where the contents of tag data in the tag FB are displayed in a controller form.

Access the tag data to monitor the status of relevant processes and set conditions.

Each item of tag data can be viewed and edited on the faceplate.

On GX Works3, up to two faceplates can be displayed simultaneously.

| No.	Name	Description
(1)	I/O mode display area	Displays the current I/O mode if the tag type is loop tag or status tag. When this button is clicked, the "Change I/O Mode" window is displayed. The following four I/O modes are available. • NORMAL (NOR) • SIMULATION (SIM) • OVERRIDE (OVR) • TAG STOP (TSTP) The I/O mode can be changed only when the control mode is MANUAL.
(2)	Tag name display area	Displays the tag name.
(3)	Tag comment display area	Displays a comment defined with the tag FB setting editor.
(4)	Alarm display area	Displays the tag alarm.
(5)	PV value bar display area	Displays the current PV value in the bar form. A graph is displayed in the range of PV engineering value upper limit to PV engineering value lower limit. An alarm bar is displayed if the PV upper upper limit alarm value, PV upper limit alarm value, PV lower limit alarm value, or PV lower lower limit alarm value is set.
(6)	SV value bar display area	Displays a green bar indicating the range of the SV upper limit value to SV lower limit value, and a yellow guide indicating the current SV value.
(7)	PV value setting/display area	Numerically displays the current PV value. This button is displayed only when the I/O mode is OVERRIDE. When this button is clicked, the PV value setting dialog is displayed.
(8)	SV value setting/display area	Numerically displays the current SV value. When this button is clicked, the SV value setting dialog is displayed.
(9)	MV value setting/display area	Numerically displays the current MV value. (Unit: %) When this button is clicked, the MV value setting dialog is displayed.
(10)	MV value bar display area	Displays a green bar indicating the range of the MV upper limit value to MV lower limit value, and a yellow guide indicating the current MV value.

| No.	Name	Description
(11)	Control mode button	Displays the current control mode of when the tag type is loop tag or status tag. When this button is clicked, the control mode setting change dialog is displayed. The following six types of control modes are available. • MANUAL (MAN) • AUTO (AUT) • CASCADE (CAS) • COMPUTER MV (CMV) • COMPUTER SV (CSV) • CASCADE DIRECT (CASDR) The control mode cannot be switched from the MANUAL mode to other modes if the I/O mode is OVERRIDE.
(12)	Tag type display area	Displays the tag type.
(13)	Monitor status display area	Displays the current monitor status.
(14)	Pattern graph display area	Displays a graph showing the setting values for the time predetermined by each step.
(15)	Operation type setting/display area	Displays a window for changing the operation type.
(16)	Execution step number setting/display area	Displays a window for setting the execution step number.
(17)	Time in the step setting/ display area	Displays a window for setting the time in the step.

Checking the operation

Monitor the temperature control operation.

Operating procedure

1. Display the TIC001_FB faceplate, and click the Control mode button.
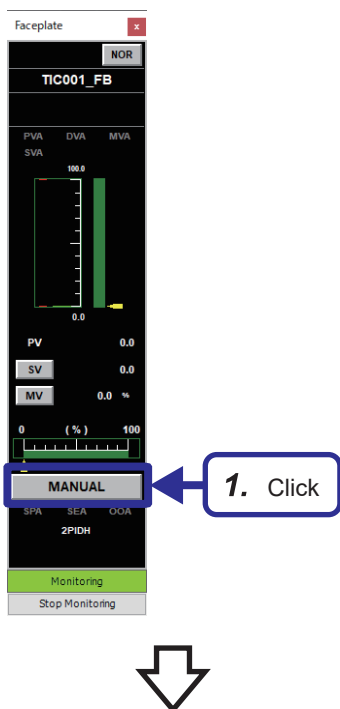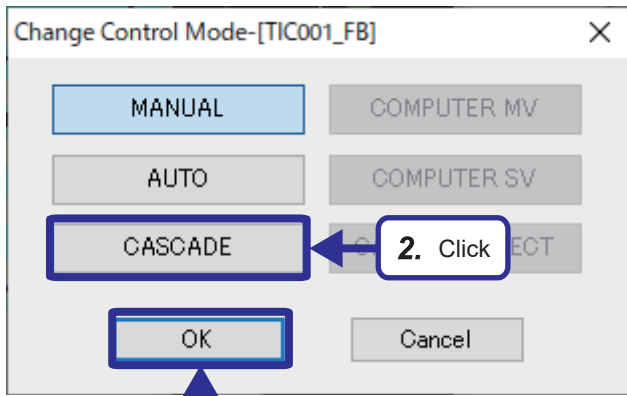
2. Click the [CASCADE] button.

3. Click the [OK] button.
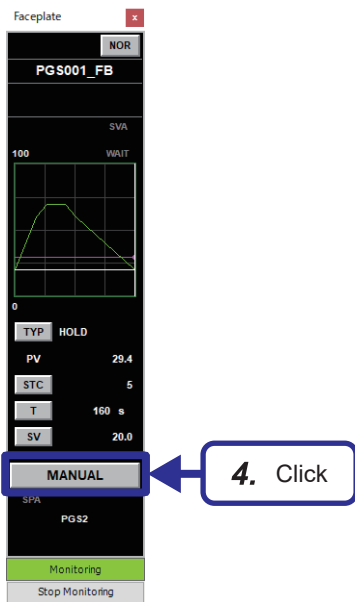
4. Display the PGS001_FB faceplate, and click the Control mode button.
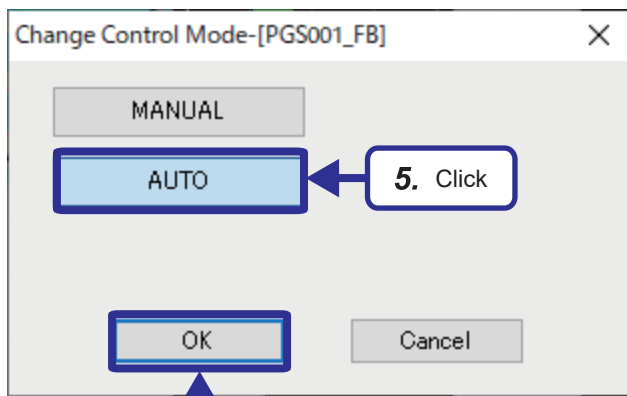
5. Click the [AUTO] button.

6. Click the [OK] button.
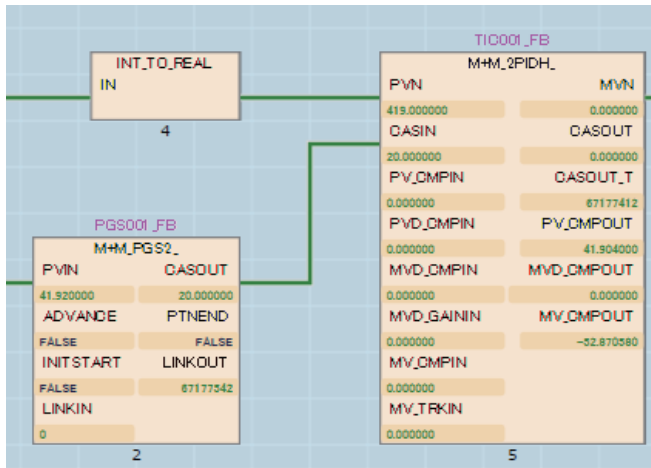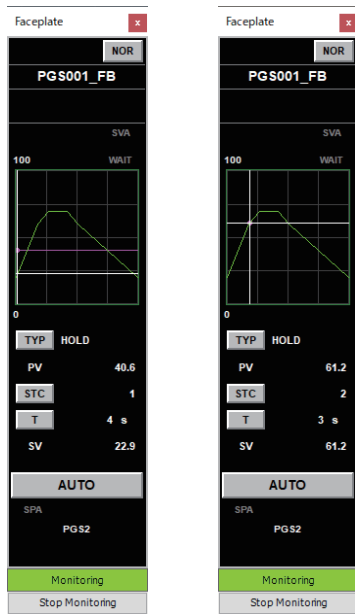
7. The executing step number (STC), time in the step (T), and SV value change according to the PGS001 values set in "☞ Page 104 FB property initial value setting example".

8. The PGS001_FB outputs the SV value to the TIC001_FB.

9. Change and adjust the MV value so that the SV value and PV value are the same in the TIC001_FB.
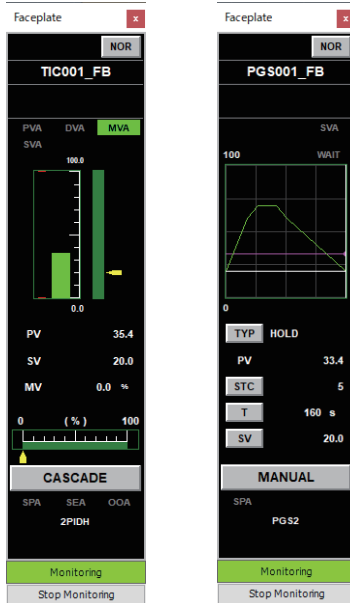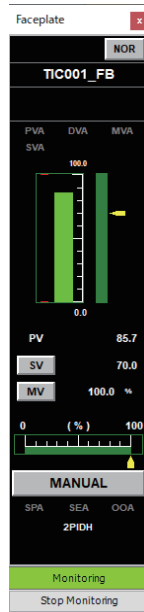
10. When 160 seconds pass at the executing step number 5 (STC), the operation stops.

Changing FB property values online

The following describes the procedure for changing FB property values in the CPU module via GX Works3 online.
In this example, TIC001 PID will be changed.

Operating procedure
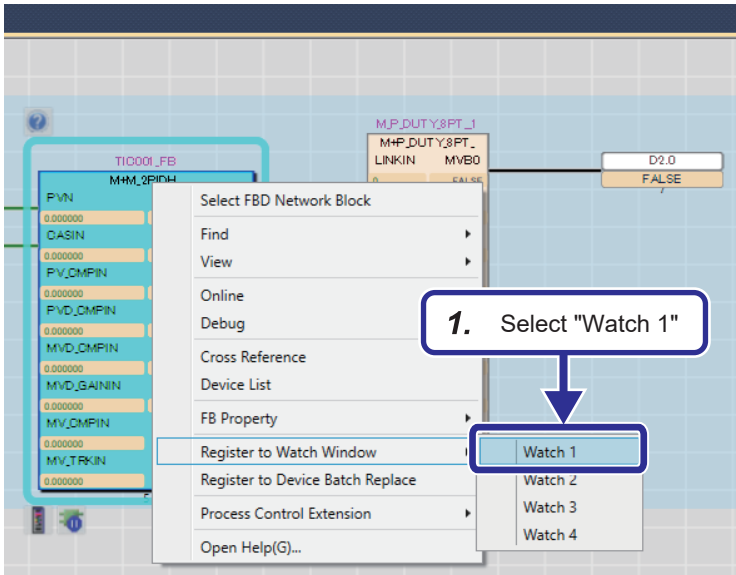
1. Select a tag FB whose FB property values are to be changed, and right-click and select [Register to Watch Window] ⇨ [Watch 1] to register the tag FB.
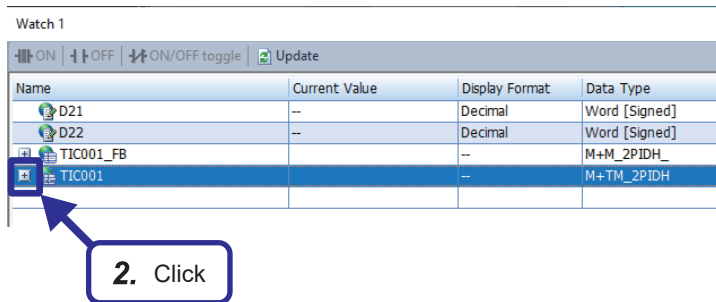
2. Open the Watch 1 window, and click [+] for "TIC001".
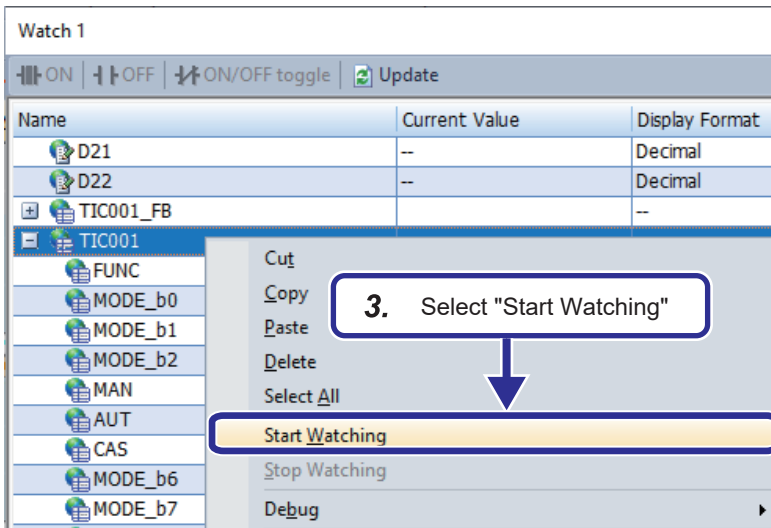
3. Right-click and select [Start Watching].

3. Select "Start Watching"
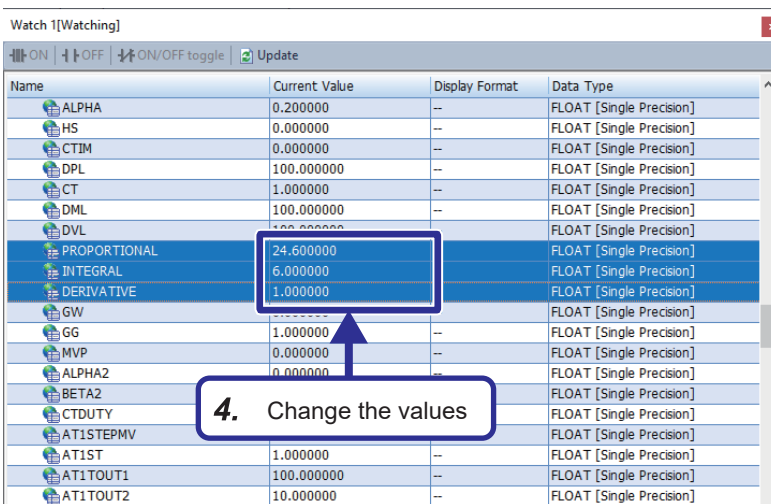
4. Double-click the FB property values in the "Current Value" column and change them as follows.

[Changes]
PROPORTIONAL: 24.6
INTEGRAL: 6.0
DERIVATIVE: 1.0

4. Change the values

- If the Watch 1 window is not displayed, select [View] ⇨ [Docking Window] ⇨ [Watch 1] from the menu of the engineering tool.
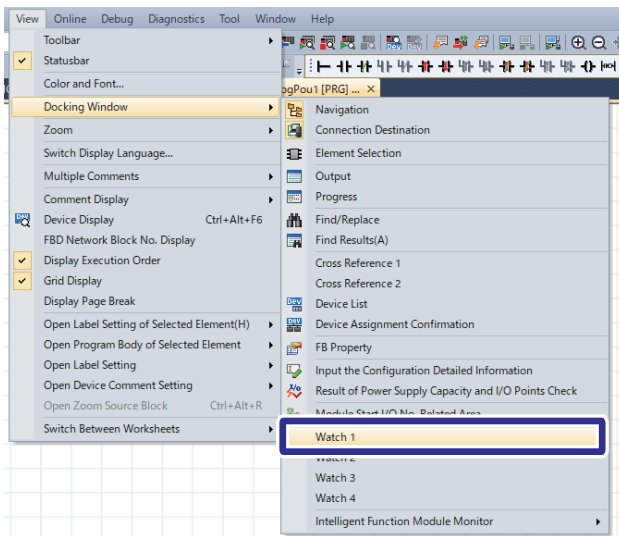
- While "[Watching]" is not displayed in the title of the Watch window, the values are not monitored. Right-click and select [Start Watching] to start monitoring of the Watch window.
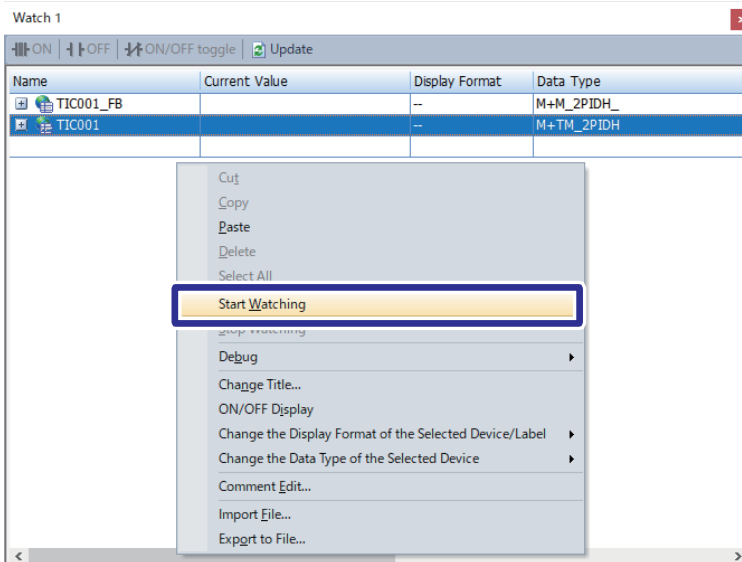
Applying FB property values to the project

The following describes the procedure for applying the FB property values currently written on the programmable controller to the project as initial values.

This section describes the procedure to apply the PID parameters of TIC001 changed online in "Page 118 Changing FB property values online".

Operating procedure

1. Select "FB Property Management (Online)"
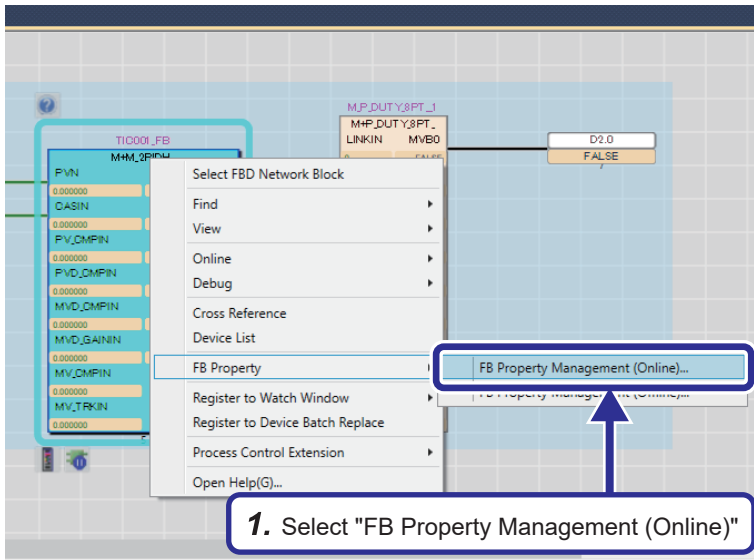
1. Select the tag FB to which the current FB property values are to be applied as the project initial values, and right-click and select [FB Property] ⇨ [FB Property Management (Online)].
To apply all tag FBs, select [Online] ⇨ [FB Property Management (Online)] from the menu of the engineering tool.

MELSOFT GX Works3

⚠ Are you sure you want to read the current value of FB property?

[Caution]
- It may take several minutes to complete, depending on the number of FB properties.
- It may not be read correctly if the contents are not matched between the PLC and the Project. Please execute Verify with PLC before proceeding to ensure the contents are matched.

2. Click → [Yes] [No]
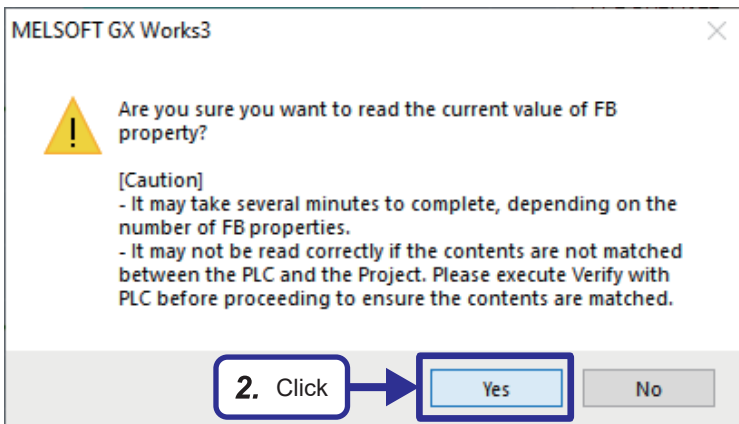
2. Click the [Yes] button.
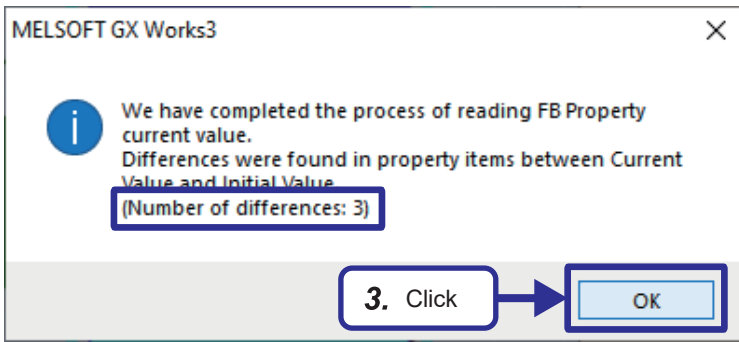
3. Click the [OK] button.
The number of items where the current value and initial value are different is displayed.
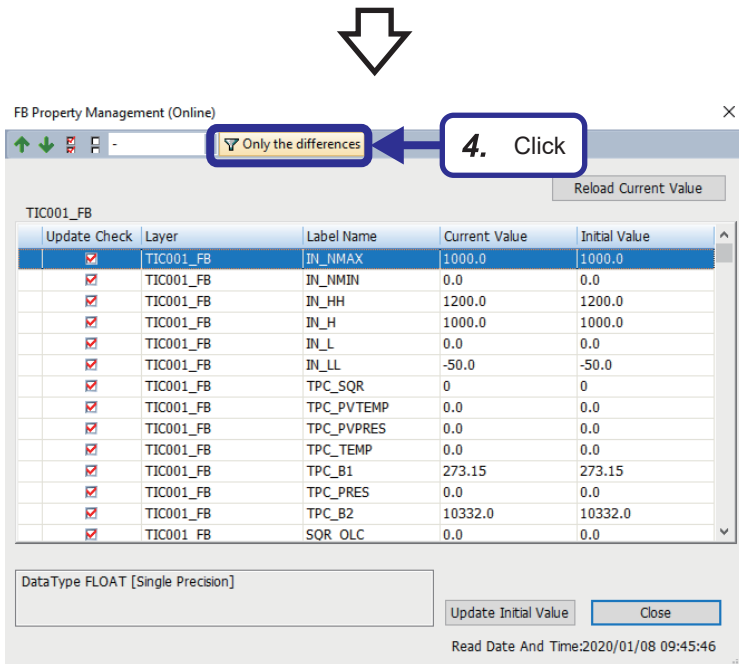
4. The "FB Property Management (Online)" window is displayed, where the current values and initial values are listed.
Selecting "Only the differences" filters the list, displaying only the items where the current value and initial value are different.
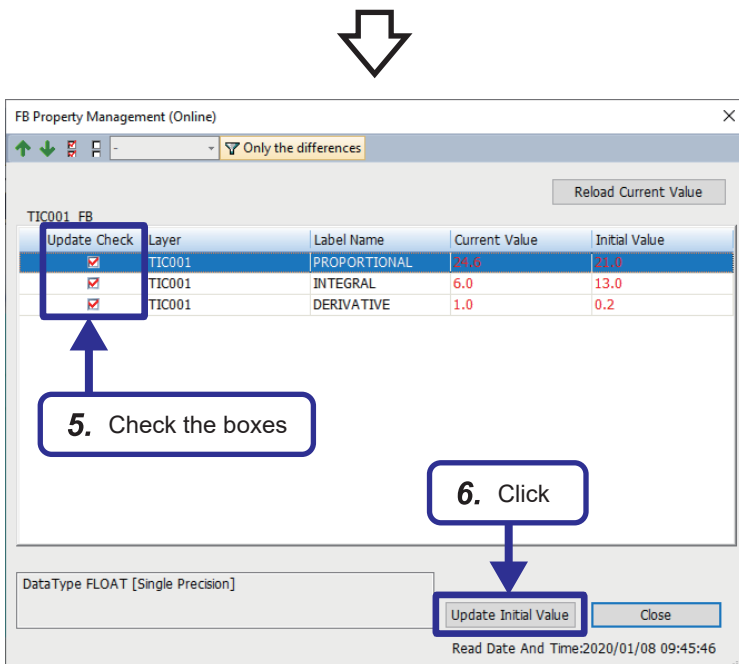
5. Select the "Update Check" check box of items to which the current value is to be applied as the initial value.

6. Click the [Update Initial Value] button to apply the current value as the initial value for the selected items.

7. Convert the program whose settings are changed, and write it to the Process CPU.
☞ Page 106 Converting the program
☞ Page 108 Writing a Project

8. For operational checks after changing the PID setting, refer to ☞ Page 172 Checking the Operation.

6.2 Monitoring with PX Developer Monitor Tool

This section describes the procedure for monitoring the redundant system via Ethernet by using the redundant system monitor system of the PX Developer monitor tool as shown in the following figure.
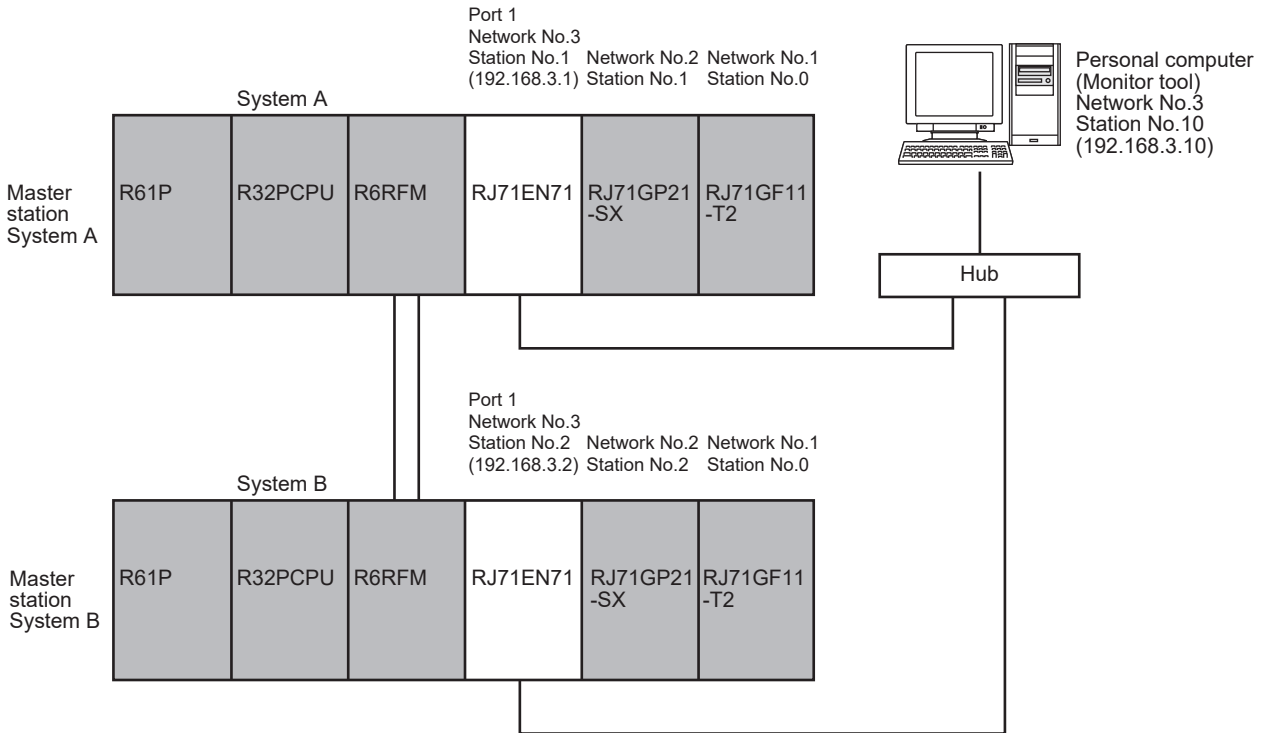
Setting the GX Works3 project

The following setting is required for monitoring a redundant system using the PX Developer monitor tool.

■Setting the process control extension

Enable the process control extension for the program file. For details, refer to the following.

☞ Page 96 Creating a new program file

■Setting the event notification function

Operating procedure
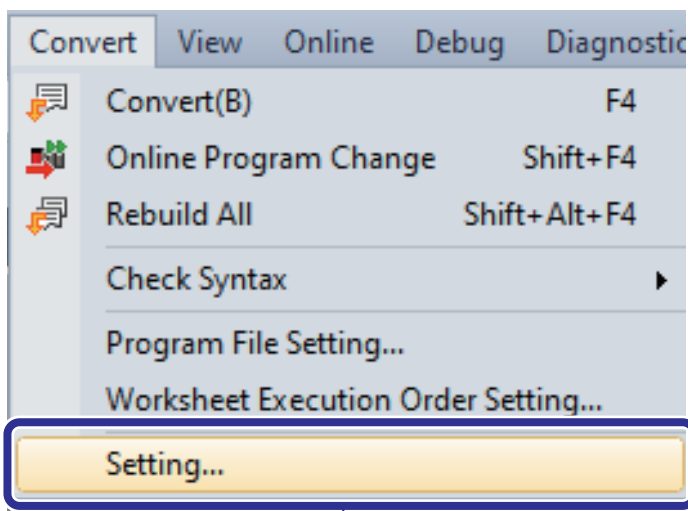
1. Select "Setting"

2. Set "Event Notification"
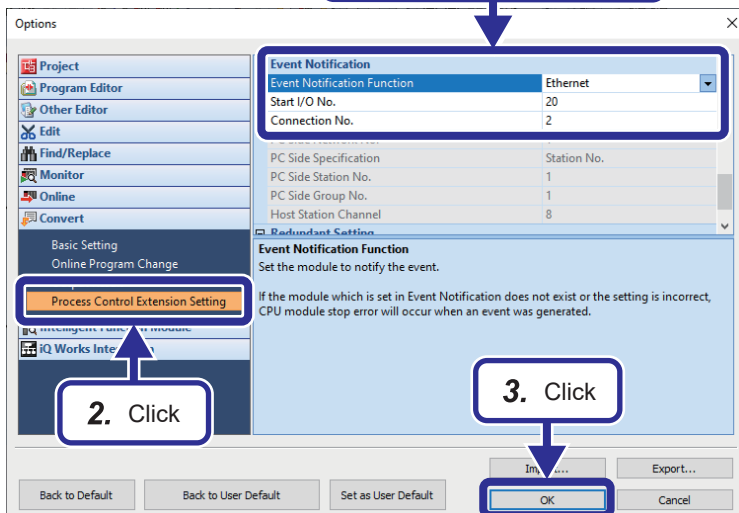
2. Click

3. Click

1. Select [Convert] ⇨ [Setting] from the menu of the engineering tool to open the "Options" window.

2. Click "Process Control Extension Setting" and configure the "Event Notification" settings as follows.
[Settings]
Event Notification Function: Ethernet
Start I/O No.: 20
Connection No.: 2

3. Click the [OK] button.

4. Convert the program.
☞ Page 106 Converting the program

Creating an assignment information database file

Export the assignment information database file (.FADB) from the project file in GX Works3. The PX Developer monitor tool uses this file to communicate with the programmable controller.

Operating procedure
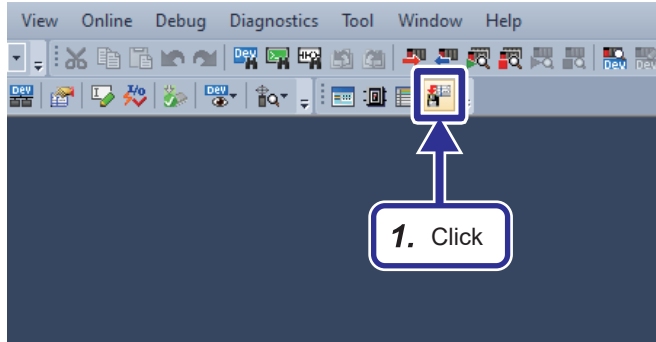
1. In the "School_Master" project, click the [Export Assignment Information Database File] icon ( ).
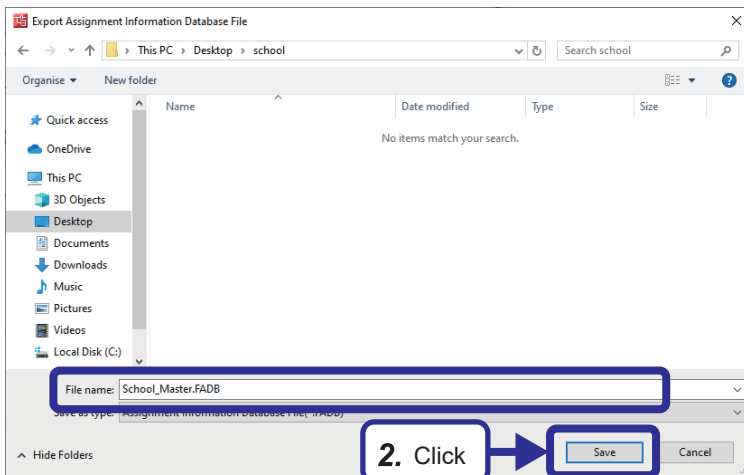
2. Save the assignment information database file (.FADB) with a name.
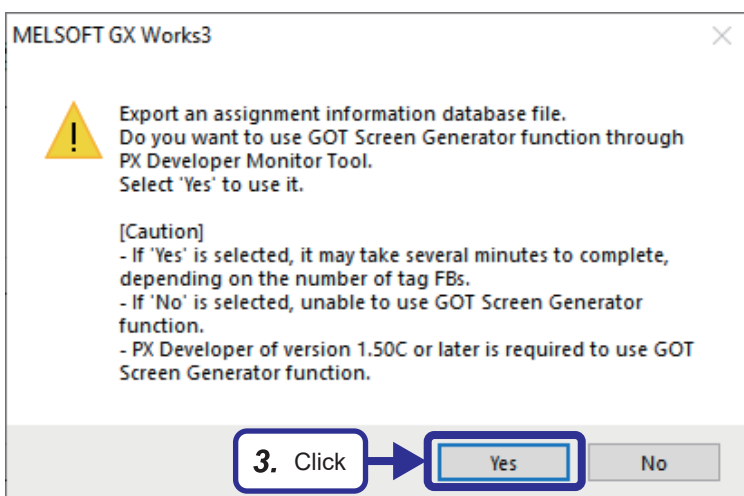
3. Click the [Yes] button.

TCP/IP setting on the personal computer

Configure the TCP/IP setting on the personal computer.

This manual describes the procedure using a personal computer running Microsoft® Windows® 10 Operating System.

Operating procedure
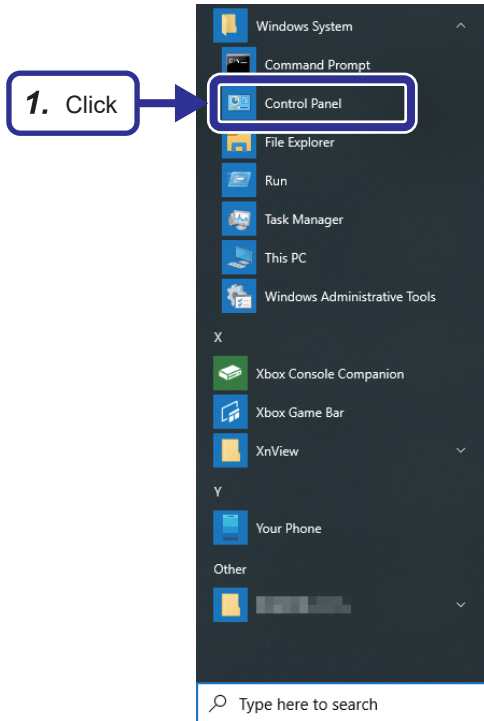
1. From the Start menu of Windows®, click [Windows System] ⇨ [Control Panel].
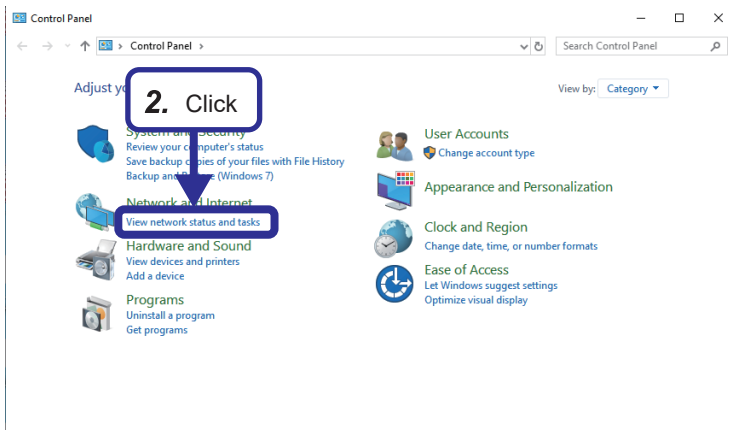
2. The Control Panel window appears. Click "View network status and tasks".
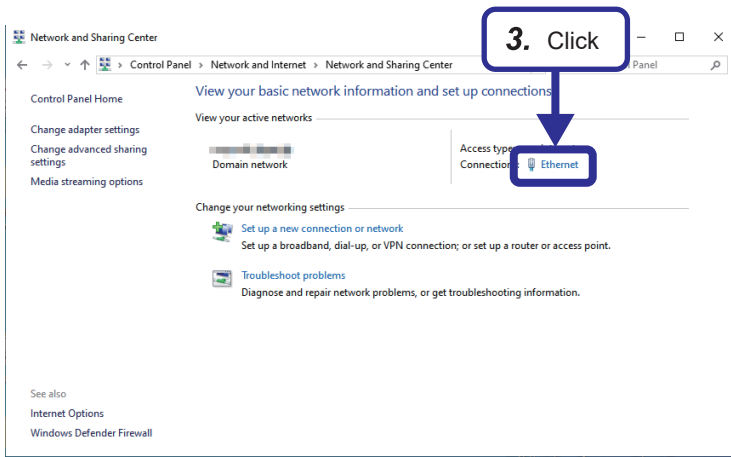
3. Click "Ethernet".
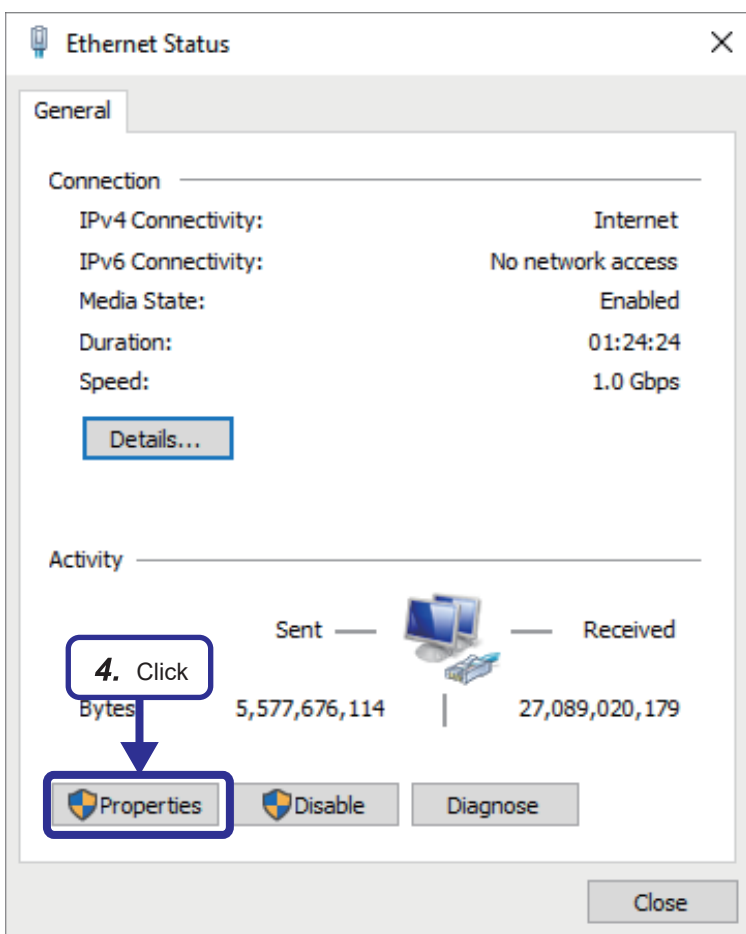
4. Click the [Properties] button.
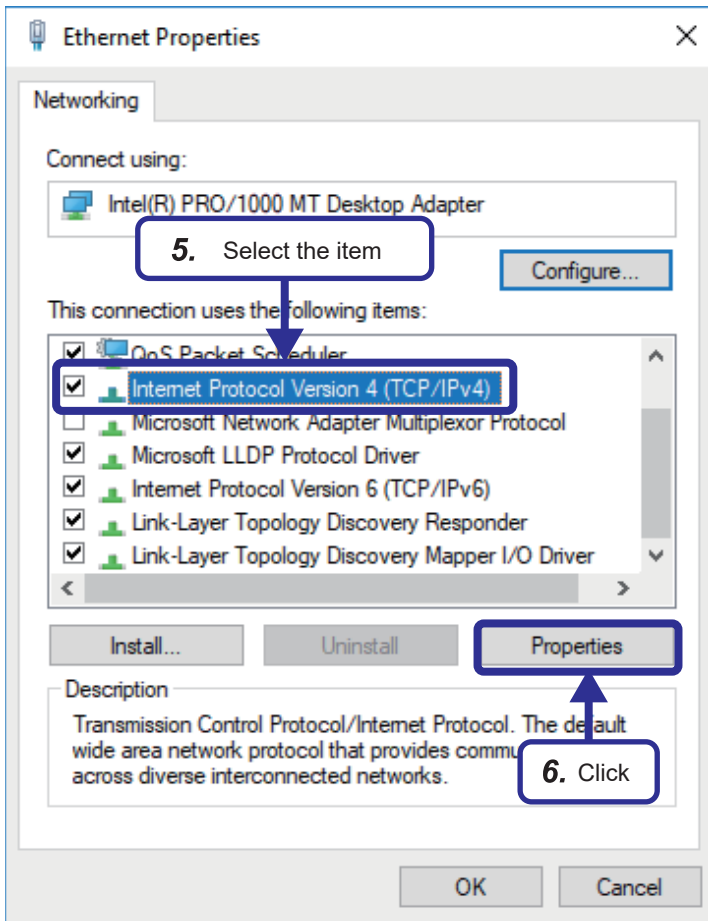
5. Select "Internet Protocol Version 4 (TCP/IPv4)".

6. Click the [Properties] button.
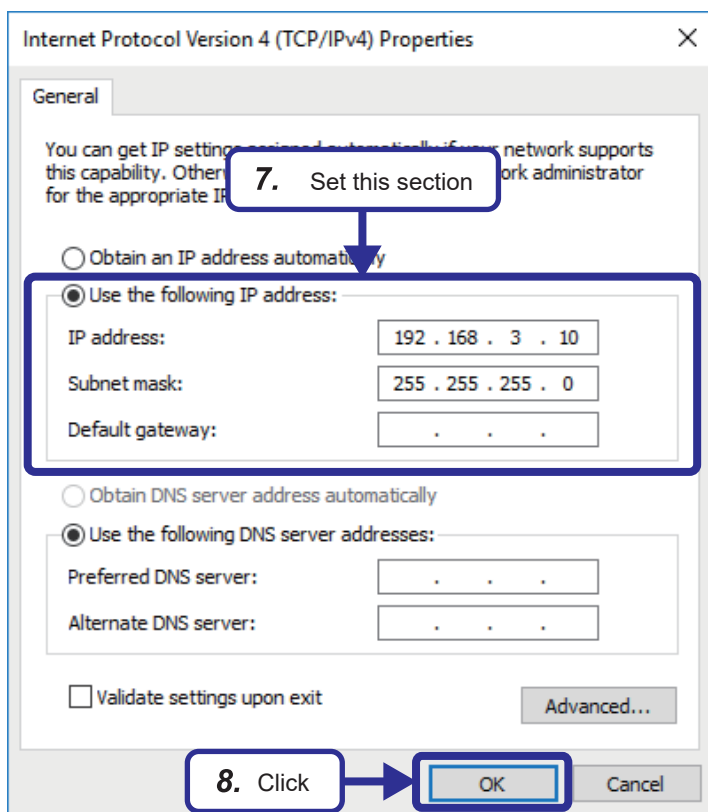
7. Select "Use the following IP address" and configure the settings as follows.
[Settings]
IP address: 192.168.3.10
Subnet mask: 255.255.255.0

8. Click the [OK] button.

9. Click the [OK] button.
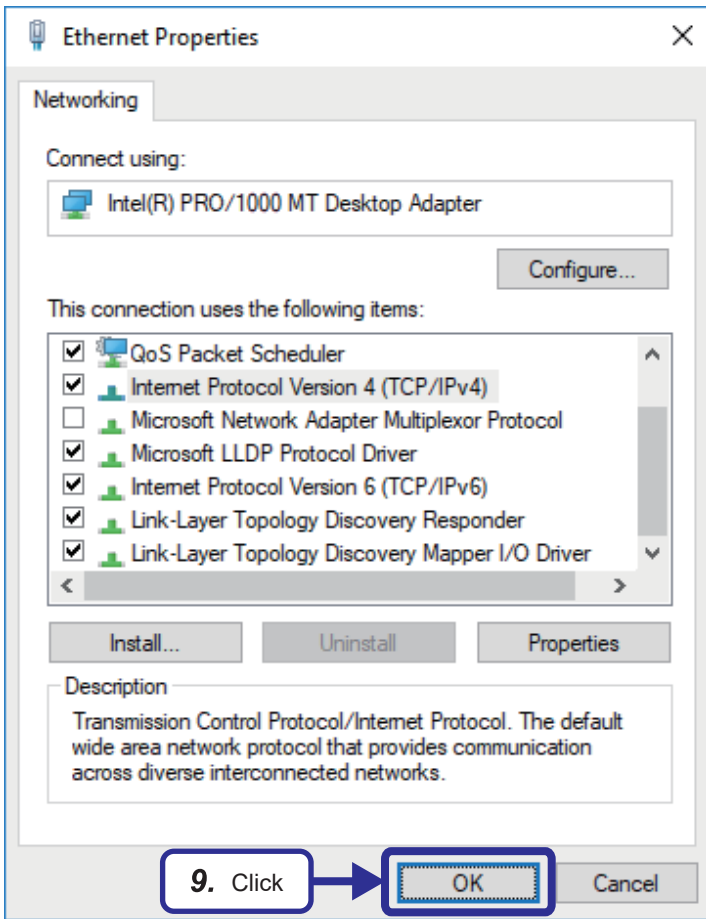
10. Click the [Close] button.
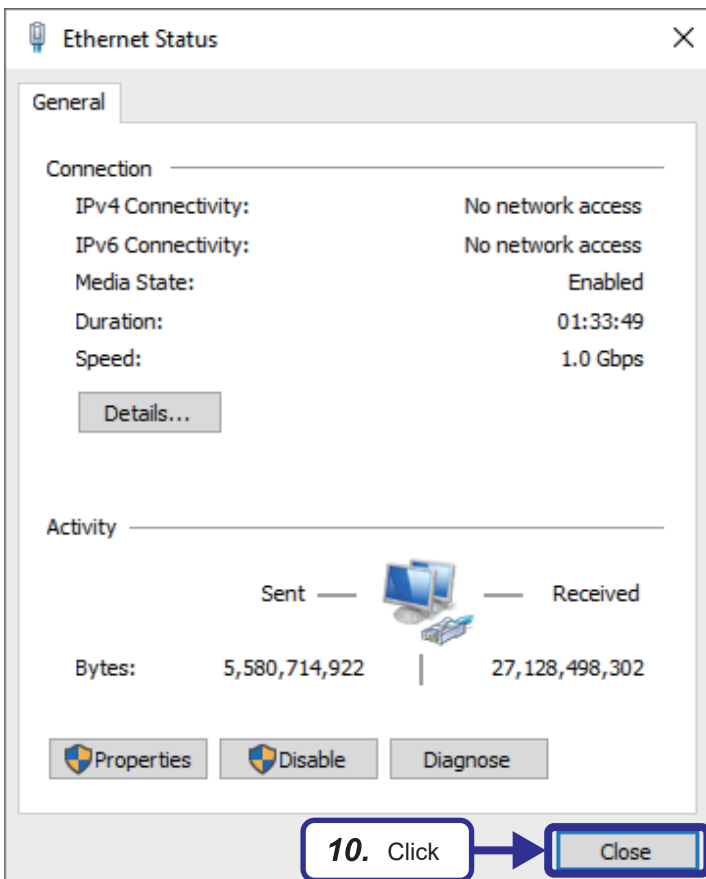
Checking Windows® firewall

If Windows® firewall is enabled and communications by GX Works3 and the monitor tool are not permitted, communications may fail and timeout may occur. Configure the following settings to permit communications.
If other security software with a firewall function is used, follow the procedure in the relevant manual and permit communications of the monitor tool.

Operating procedure
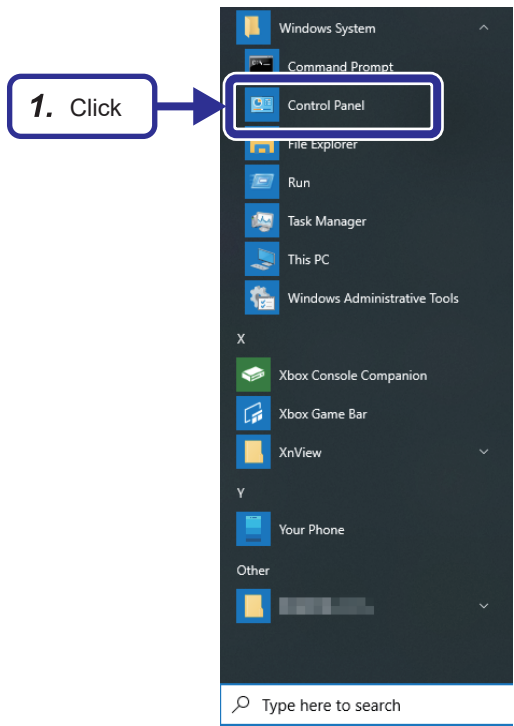
1. From the Start menu of Windows®, click [Windows System] ⇨ [Control Panel].
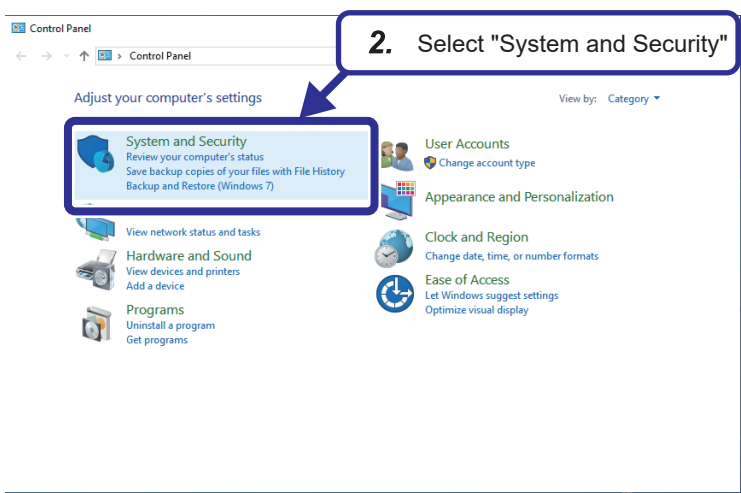
2. When the Control Panel window is displayed, select "System and Security".
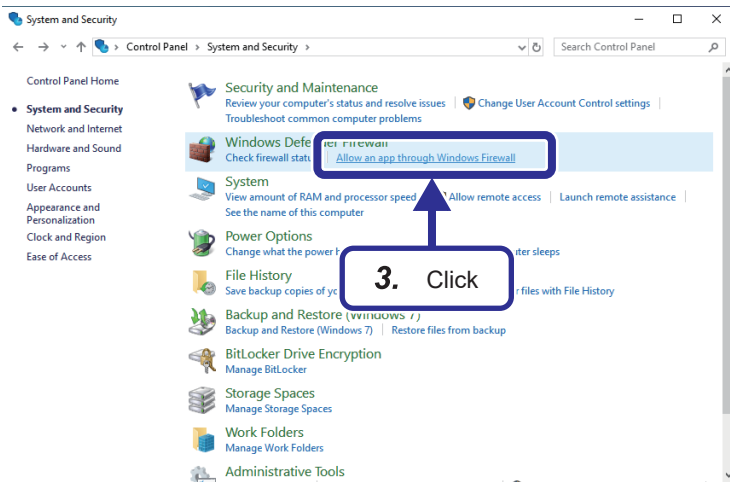
3. Click "Allow an app through Windows Firewall".
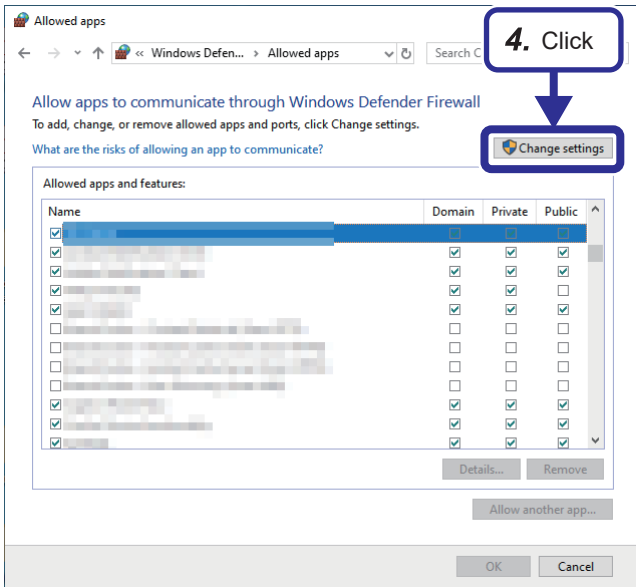
4. Click the [Change settings] button.
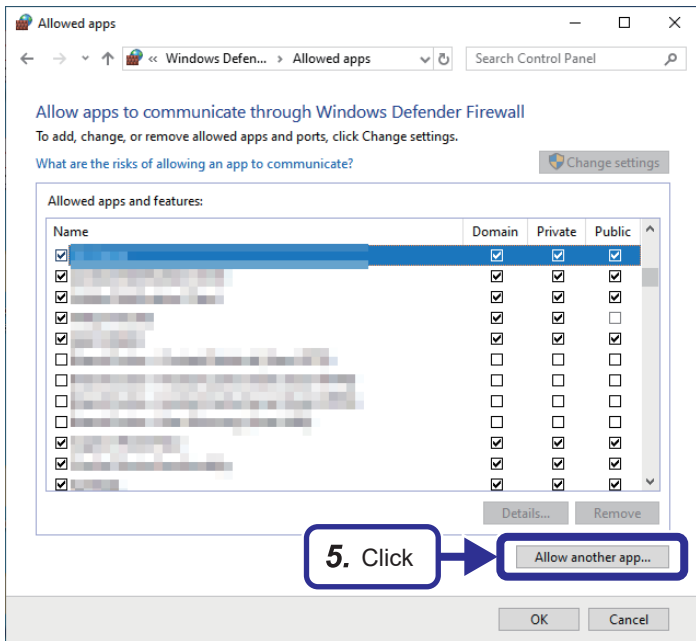
5. Click the [Allow another app] button.
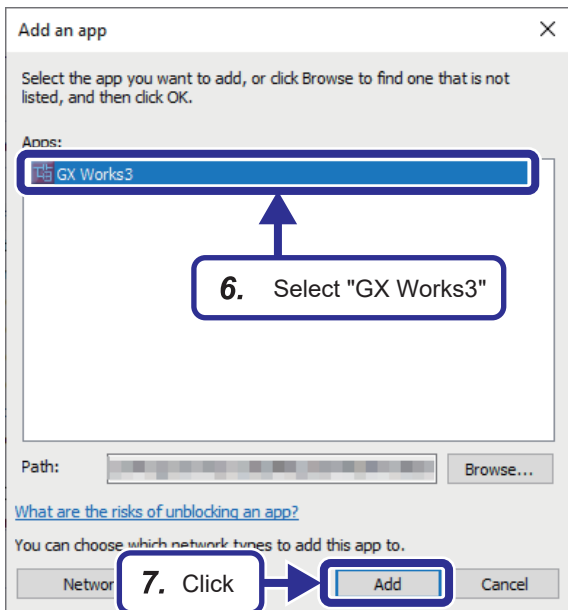
6. Select "GX Works3".

7. Click the [Add] button.
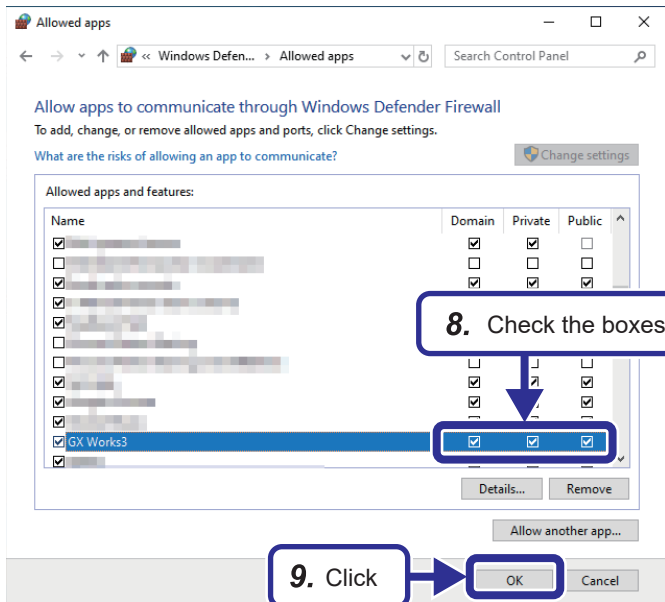
8. Select the "Private" and "Public" checkboxes for GX Works3 added on the list.
Also, select the "Domain" checkbox if it is present on the "Allowed apps and features" list.

9. Click the [OK] button.

10. Also add the monitor tool to the "Allowed programs and features" list using the same procedure.

6

Communications settings of PX Developer monitor tool

Configure the settings to allow the PX Developer monitor tool to communicate with the programmable controller.

Operating procedure
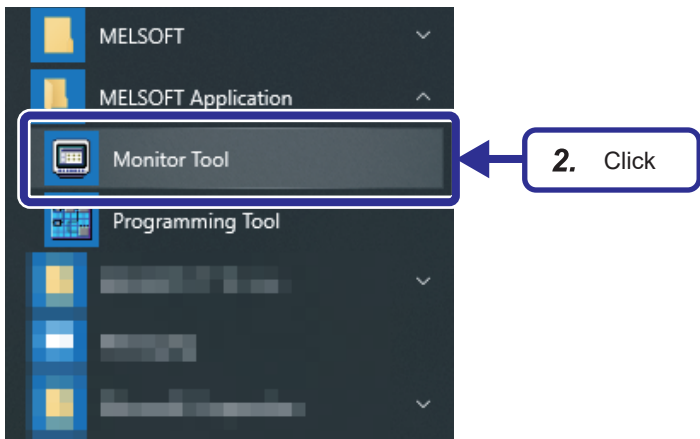
1. From the Start menu of Windows®, select [All Programs] ⇨ [MELSOFT Application] ⇨ [PX Developer]*1.

*1 It is not displayed on Windows® 8 or later.

2. Click [Monitor Tool].
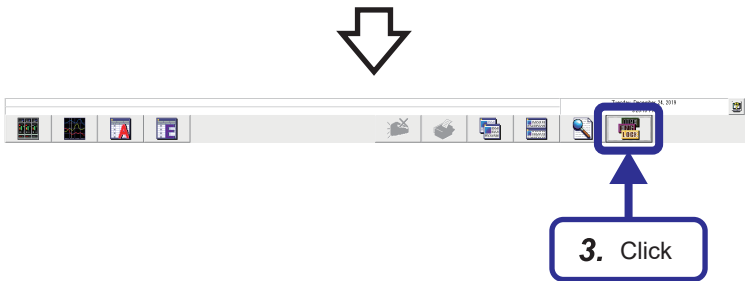
3. Click the [Change Mode] icon (🔲) of the PX Developer monitor tool.
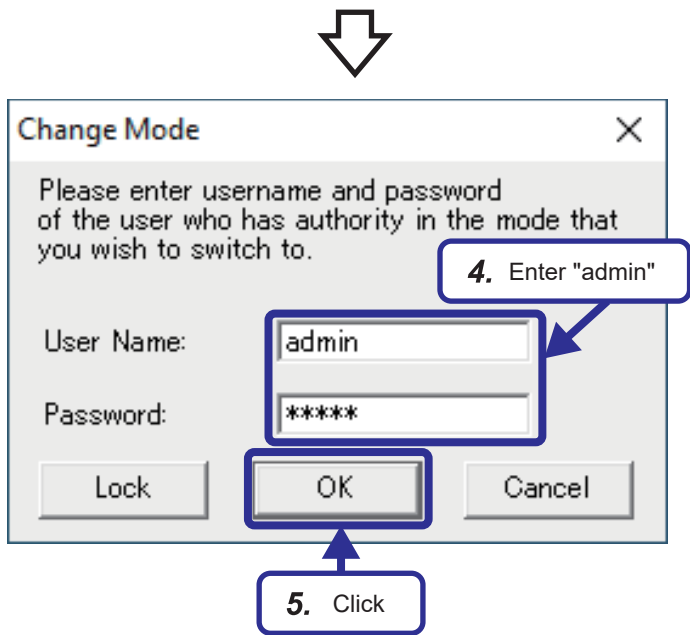
4. The "Change Mode" window appears.
Set the user name and password as follows.
[Settings]
User name: admin
Password: admin

5. Click the [OK] button.

6. The PX Developer monitor tool switches to engineering mode.
Click the [Setting Window] icon ( ).
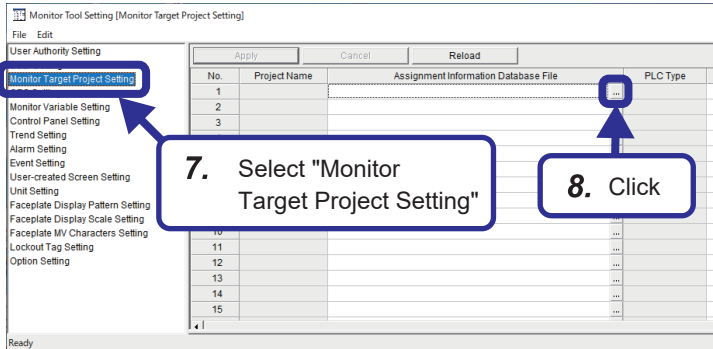
6. Click

7. The "Monitor Tool Setting" window appears.
Select "Monitor Target Project Setting".

8. Click the [...] button of "Assignment Information Database File".

7. Select "Monitor Target Project Setting"

8. Click

9. Select the assignment information database file (.FADB) to be monitored, which was created in "Page 125 Creating an assignment information database file".

10. Click the [Open] button.
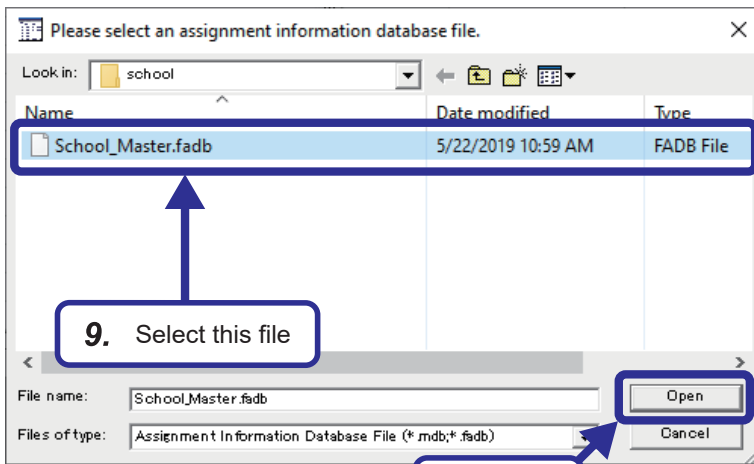
9. Select this file
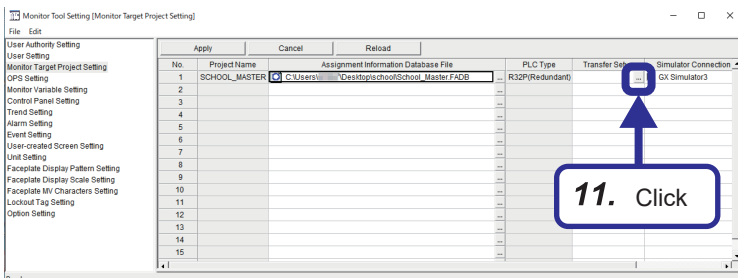
10. Click

11. Set "Transfer Setup" for the database file.
Click the [...] button in the "Transfer Setup" column.

11. Click
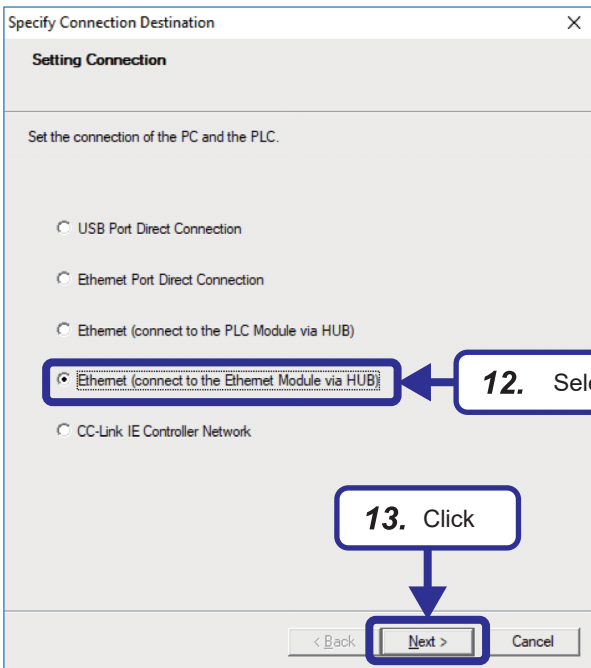
12. The "Specify Connection Destination" window appears.
Select "Ethernet (connect to the Ethernet Module via HUB)".

13. Click the [Next] button.
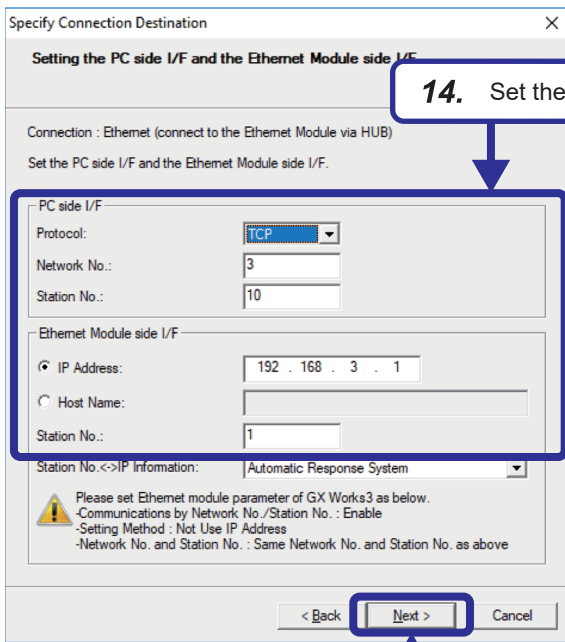
14. Configure the personal computer-side I/F and Ethernet module-side I/F as follows.
[Settings]
[PC side I/F]
Protocol: TCP
Network No.: 3
Station No.: 10
[Ethernet Module side I/F]
IP Address: 192.168.3.1
Station No.: 1

15. Click the [Next] button.
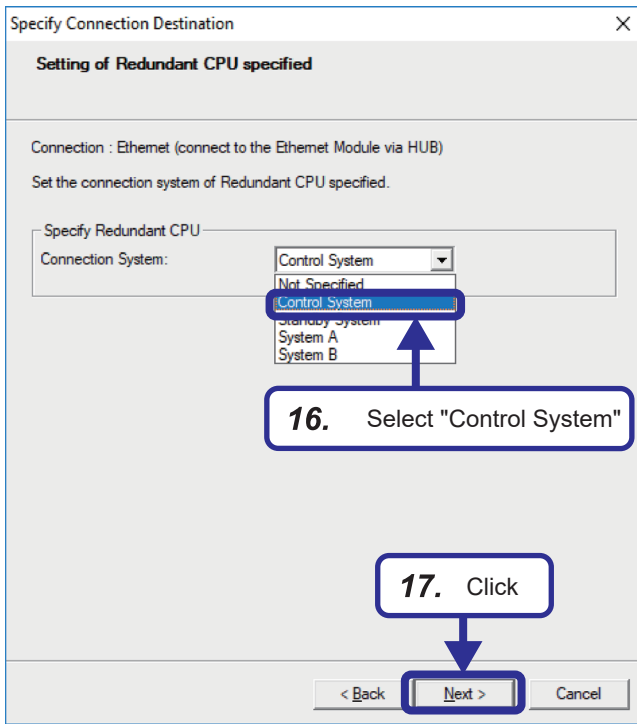
16. Specify the connection destination for the redundant system.
From the "Connection System" drop-down menu, select "Control System".

17. Click the [Next] button.
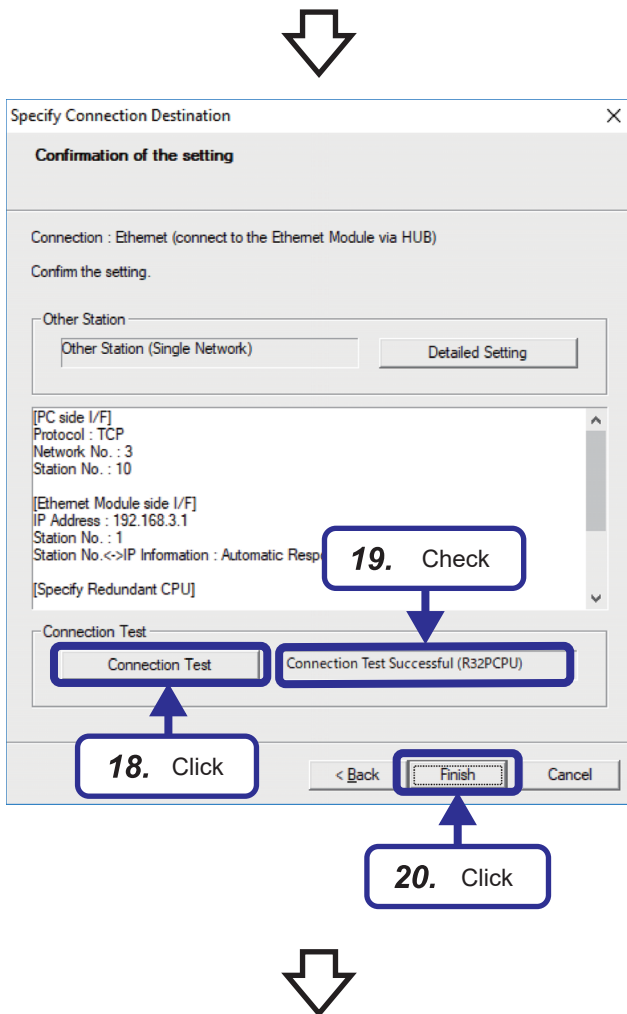
18. Perform a connection test.
Click the [Connection Test] button.

19. The connection test result is displayed in the field to the right of the [Connection Test] button.
Check that "Connection Test Successful" is displayed.

20. After checking that the connection test was successful, click the [Finish] button.
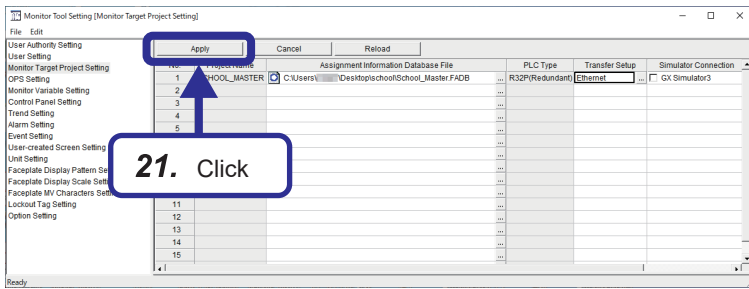
21. Click the [Apply] button to apply the setting, and close the monitor tool setting window.
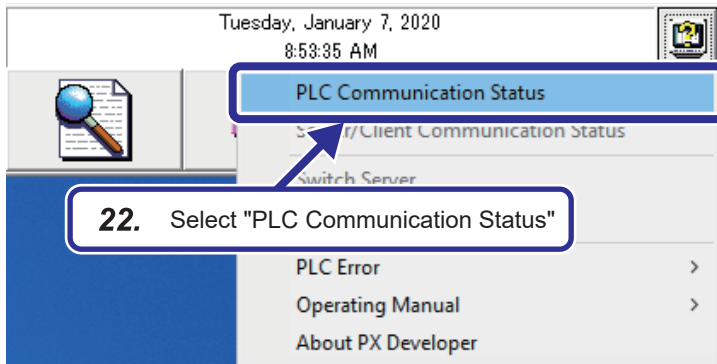
22. From the toolbar of the PX Developer monitor tool, select [Help Menu] (◻️) ⇨ [PLC Communication Status].

23. The "PLC Communication Status" window appears.
Check the communication status.

■ Normal state

■ Error state
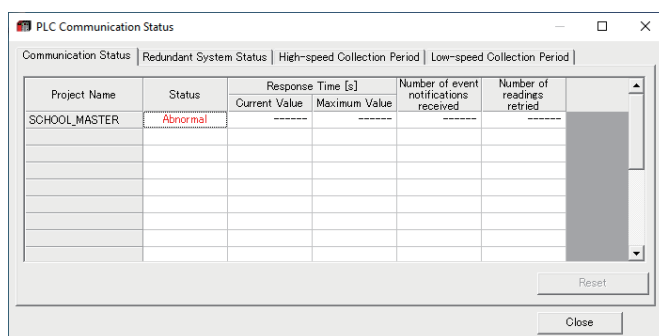
Point

The assignment information database file (.FADB) contains assignment information of variables that store tag data and global variables. When the rebuild (reassign) operation is performed for the GX Works3 project, perform the export and read operations again.

Registering a tag to PX Developer monitor tool

The PX Developer monitor tool has the control panel function for displaying up to eight faceplates on one window. One window is referred to as a control panel group, and up to 500 groups (8 x 500 = 4000 faceplates) can be monitored.
Before start monitoring, register the monitoring target tags used in the GX Works3 project to the PX Developer monitor tool.
This section describes the procedure for registering the tags (TIC001, PGS001) registered in "Page 93 Registering Tag FBs" as monitoring targets.

Control panel setting

The control panel is a function for displaying faceplates that are managed in groups. In the control panel setting, assign the tags to be displayed on the control panel window.

Operating procedure

1. Click the [Setting Window] icon ([icon]).
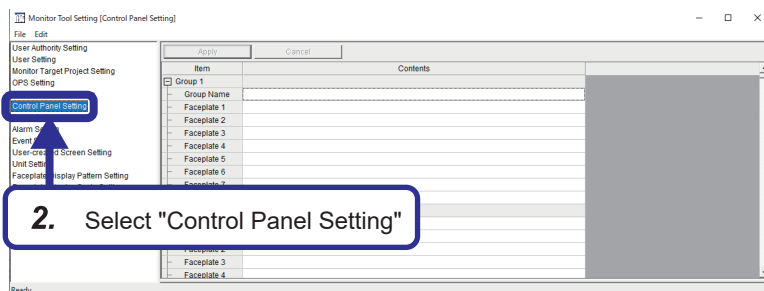
2. The "Monitor Tool Setting" window appears.
Select "Control Panel Setting".
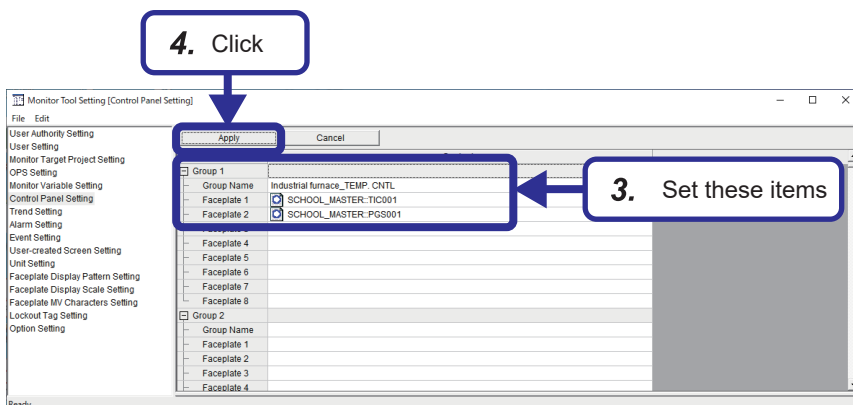
3. Set the tags whose faceplate is to be displayed as shown below.
[Settings]
Group Name: Industrial furnace_TEMP.CNTL
Faceplate 1: SCHOOL_MASTER::TIC001
Faceplate 2: SCHOOL_MASTER::PGS001

4. Click the [Apply] button.

Trend setting

The trend graph function manages chronological transitions of trend item values in groups and displays them in graphs. In the trend setting, register tag data whose trend is to be collected.

Operating procedure

1. Click the [Setting Window] icon (▥).
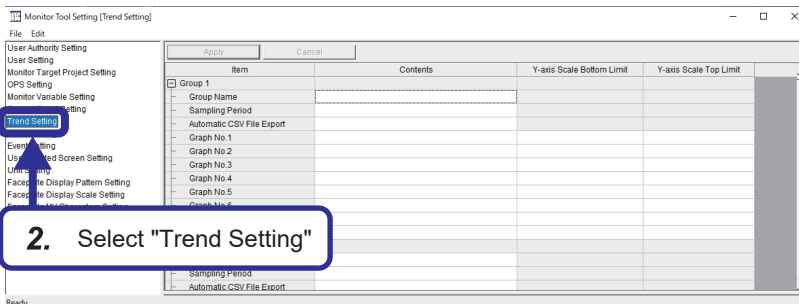
2. The "Monitor Tool Setting" window appears.
   Select "Trend Setting".
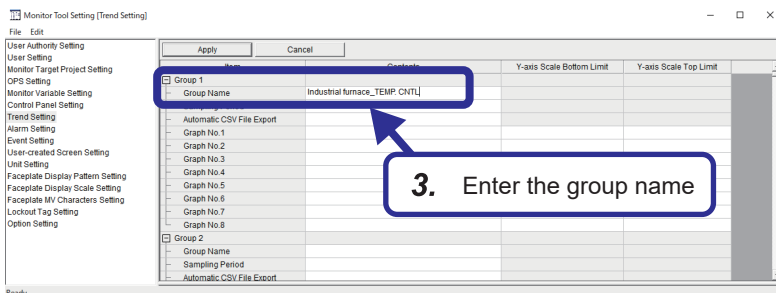
3. Register tag data whose trend is to be collected.
   Enter "Industrial furnace_TEMP.CNTL" in "Group Name" for Group 1.
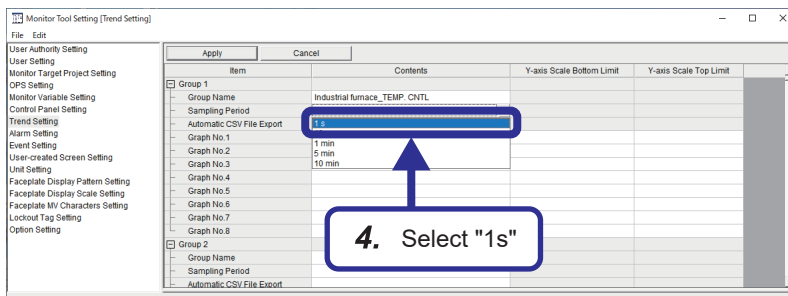
4. From the "Sampling Period" drop-down menu, select "1 s".
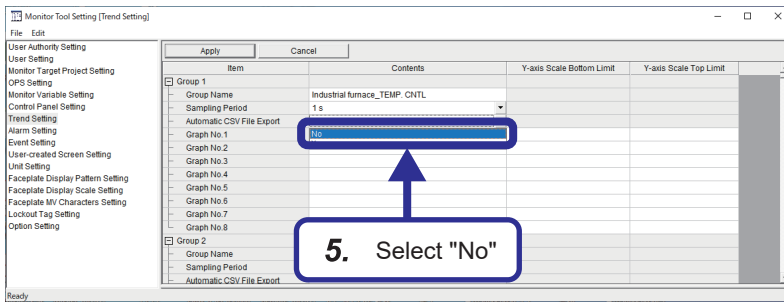
5. From the "Automatic CSV File Export" drop-down menu, select "No".
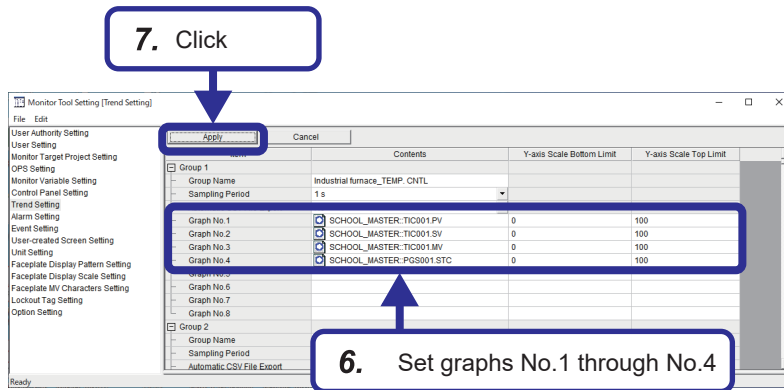
6. Configure graph No.1 to No.4 as follows.
[Settings]
[Graph No.1]
Contents:
SCHOOL_MASTER::TIC001.PV
Y-axis Scale Bottom Limit: 0
Y-axis Scale Top Limit: 100
[Graph No.2]
Contents:
SCHOOL_MASTER::TIC001.SV
Y-axis Scale Bottom Limit: 0
Y-axis Scale Top Limit: 100
[Graph No.3]
Contents:
SCHOOL_MASTER::TIC001.MV
Y-axis Scale Bottom Limit: 0
Y-axis Scale Top Limit: 100
[Graph No.4]
Contents:
SCHOOL_MASTER::PGS001.STC
Y-axis Scale Bottom Limit: 0
Y-axis Scale Top Limit: 100

7. Click the [Apply] button.

Unit setting

In the unit setting, register a unit string to be used to display tag data items. By setting corresponding numbers to the unit strings registered here, the unit strings to be used for display can be specified in FB property UNIT.

For example, when the UNIT for TIC001 FB property is set to "1" and No.1 of the unit setting of the PX Developer monitor tool is set to "℃", the display unit for TIC001 PX Developer monitor tool is "℃".

Operating procedure
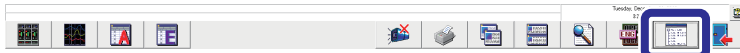
1. Click the [Setting Window] icon ( ).
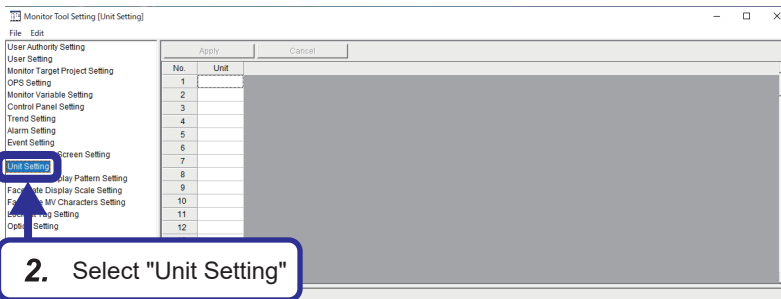
1. Click

2. The "Monitor Tool Setting" window appears.
Select "Unit Setting".

2. Select "Unit Setting"
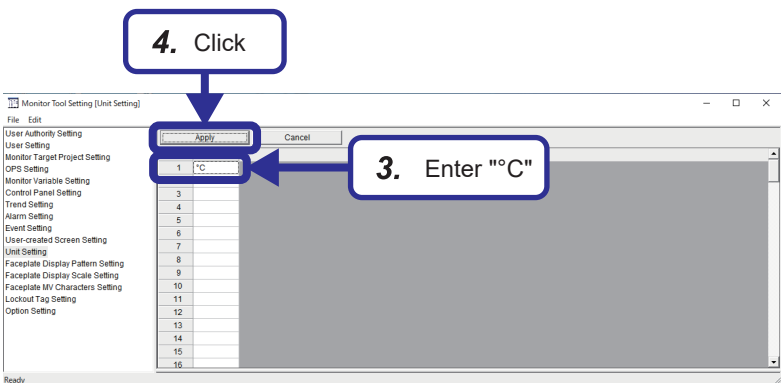
3. Register a unit string.
Enter "℃" for No.1.

4. Click

4. Click the [Apply] button.

3. Enter "°C"

Checking the displayed information

Display the control panel and trend graph to check that the settings in "Page 139 Control panel setting", "Page 140 Trend setting", and "Page 142 Unit setting" are reflected on the PX Developer monitor tool.

Operating procedure

1. Click the [Control Panel] icon ( ).
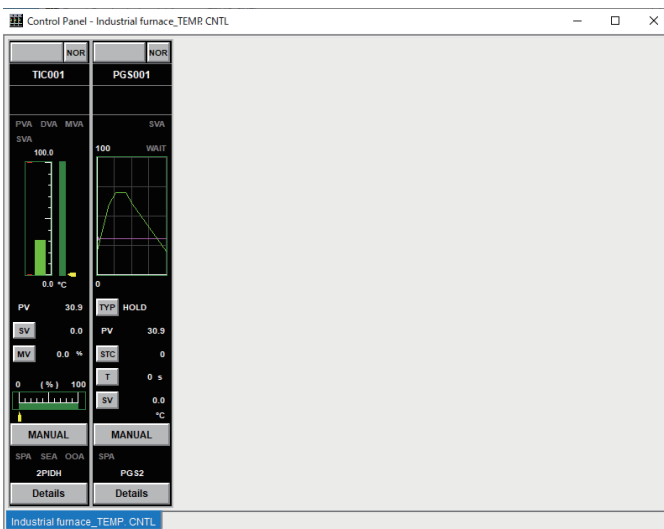
1. Click

2. Check that the faceplates of the tags registered in the control panel setting are displayed.

3. Click the [Trend Graph] icon ( ) to display the trend graph.

3. Click
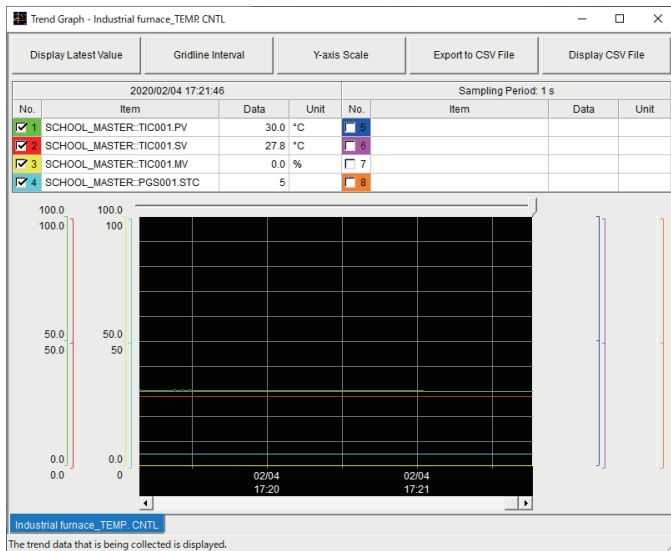
4. Check that the tag data registered in the trend setting are collected.

Checking auto tuning operation

When performing tuning on the pop-up tuning window of PID-type, 2PID-type, and 2PIDH-type tags, auto tuning can be performed for automatic rough adjustment before performing precise adjustment.

For details on the auto tuning, refer to "☞ Page 61 Auto Tuning".

Auto tuning by the limit cycle method

Because the tag FB TIC001 (M+M_2PIDH_) is used in the created project "School_Master", auto tuning by the limit cycle method is available.

This section describes how to perform auto tuning by the limit cycle method.

Operating procedure
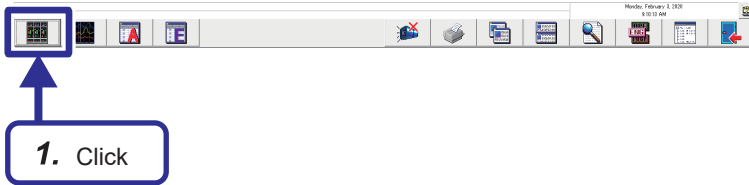
1. Double-click the [Control Panel] icon ( ) in the PX Developer monitor tool to display the control panel.
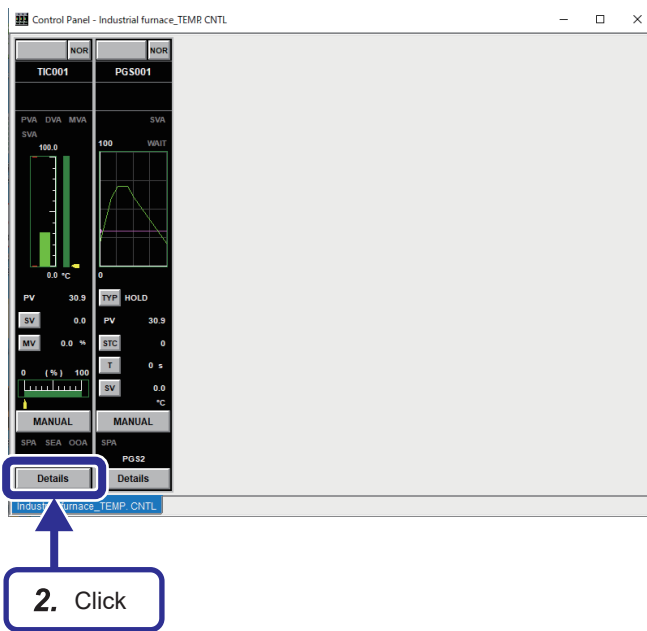
1. Click

2. Click the [Details] button of the "TIC001" faceplate.
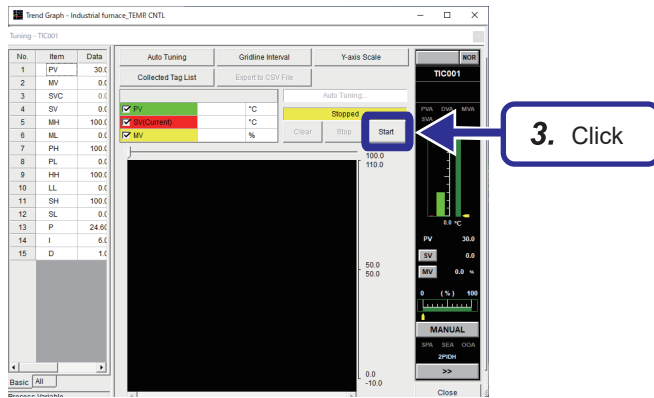
2. Click

3. Click the [Start] button.
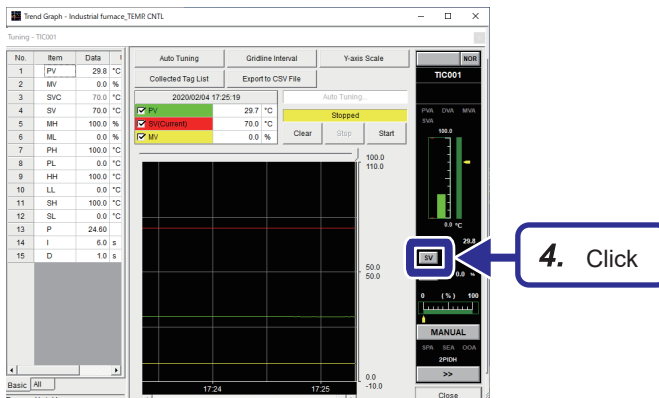
3. Click

4. Click the [SV] button of the faceplate.

4. Click

5. Set a value
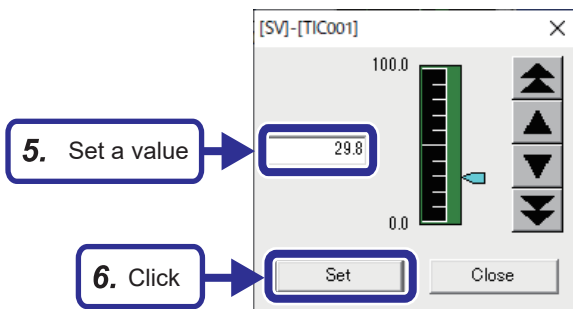
6. Click

5. When the SV window is displayed, set the SV value to the same value as the ambient temperature.

6. Click the [Set] button.

7. Click the Control mode button.

7. Click → MANUAL
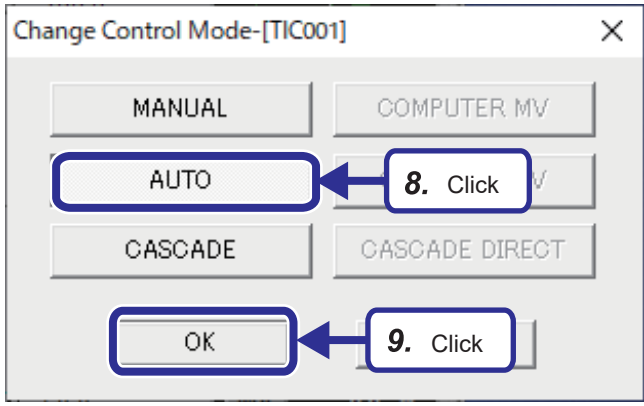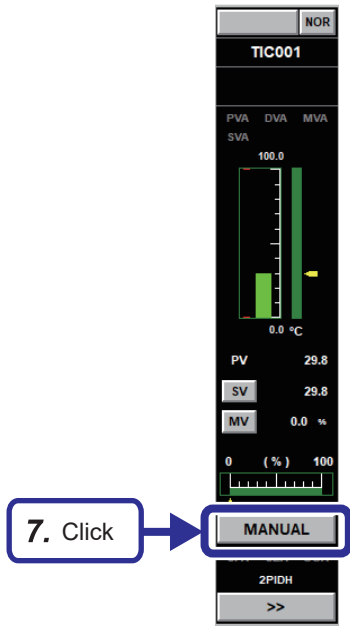
8. Click the [AUTO] button.

9. Click the [OK] button.
Check that the MV value is 0% on the faceplate.

Change Control Mode-[TIC001] ×

MANUAL   COMPUTER MV

AUTO ← **8.** Click

CASCADE   CASCADE DIRECT

OK ← **9.** Click

11. Set the SV to 50

10. Set the mode to MANUAL
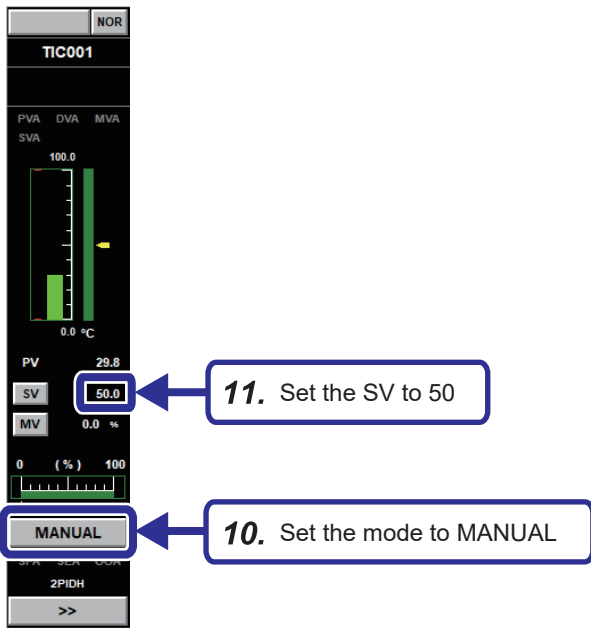
12. Click

10. Return the control mode to MANUAL on the window in step **7.** .

11. Set the SV value to 50 on the window in step **4.** .
Check that the MV value is 0% on the faceplate.

12. Click the [Auto Tuning] button.

13. Select this option

14. Click

15. Set this section

16. Click

17. Click
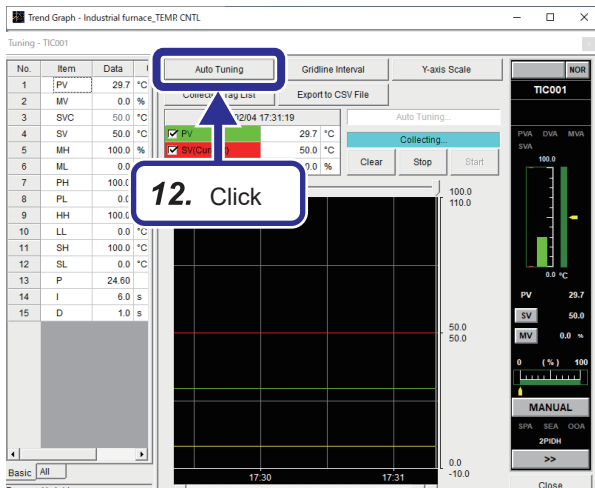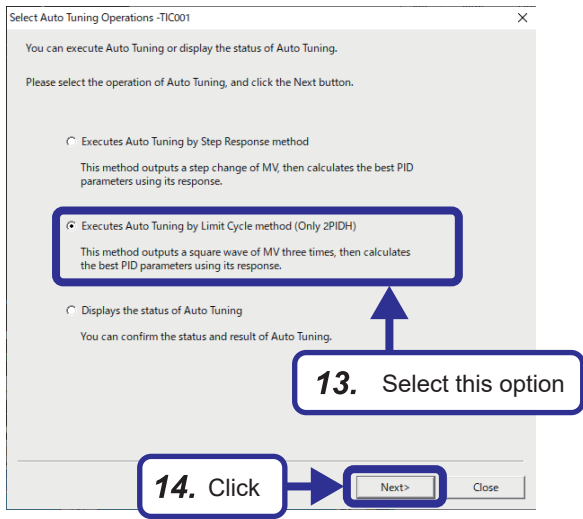
13. The "Select Auto Tuning Operations" window appears.
Select "Executes Auto Tuning by Limit Cycle method".

14. Click the [Next] button.
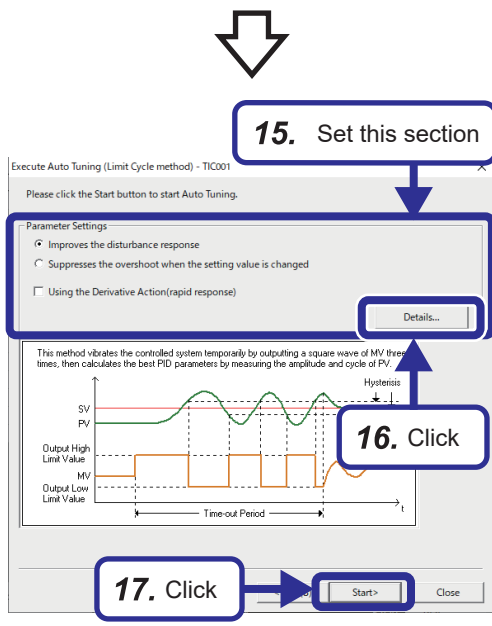
15. Configure the auto tuning related settings.
For details on the settings, refer to the following.
☞ Page 151 Auto tuning by the limit cycle method

16. Click the [Details] button.
Configure the settings as shown in the following figure.
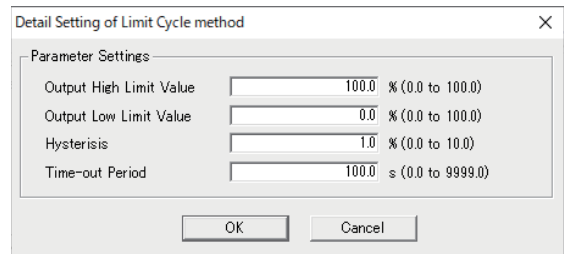
17. When the [Start] button is clicked, auto tuning begins.

6
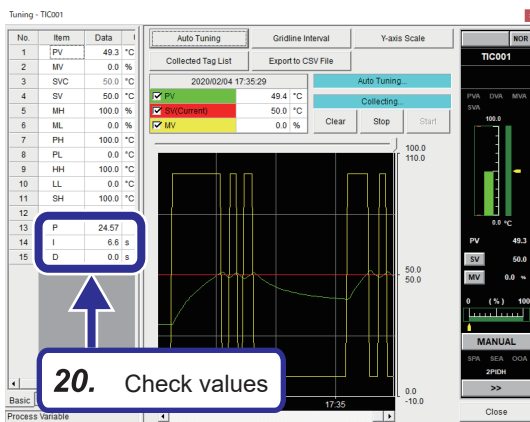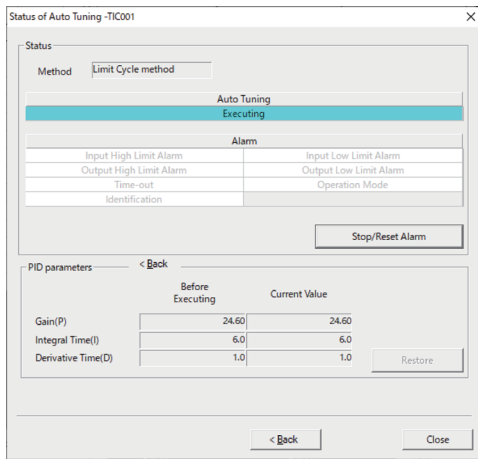
18. The "Status of Auto Tuning" window appears. The upper and lower limit values of MV are repeatedly output.
Even if the value set to AT2MVH/AT2MVL exceeds MH/ML, the MV value is output within the range of MH to ML.

19. The MV value returns to its original value automatically after auto tuning is completed. Values for proportional gain (Kp), integral time (Ti), and derivative time (Td) which are calculated by auto tuning are set automatically.

20. It can be checked that the PID values calculated by auto tuning are stored.

20. Check values

■Setting of auto tuning by the limit cycle method

The control type is determined by selecting either of "Improves the disturbance response" or "Suppresses the overshoot when the setting value is changed", and whether to check "Using the Derivative Action (rapid response)".

○: Selected, ×: Not selected

Control type	Improves the disturbance response	Suppresses the overshoot when the setting value is changed	Using the Derivative Action (rapid response)
Constant-value PI control	○	×	×
Constant-value PID control	○	×	○
Follow-up PI control	×	○	×
Follow-up PID control	×	○	○

Set the following items on the detail setting window of the Limit Cycle method which is displayed by clicking the [Details] button.
- Output upper limit (AT2MVH)
- Output lower limit (AT2MVL)
- Hysteresis (AT2HS)
- Time-out period (AT1TOUT1) (seconds)

Point

- PID constants are automatically overwritten after auto tuning.
- Auto tuning stops automatically when an alarm occurs.
- MV output values return to the values at start when auto tuning is completed or interrupted.

6

Auto tuning status display

The current auto tuning status and variation of PID constants after the auto tuning can be checked.

Operating procedure
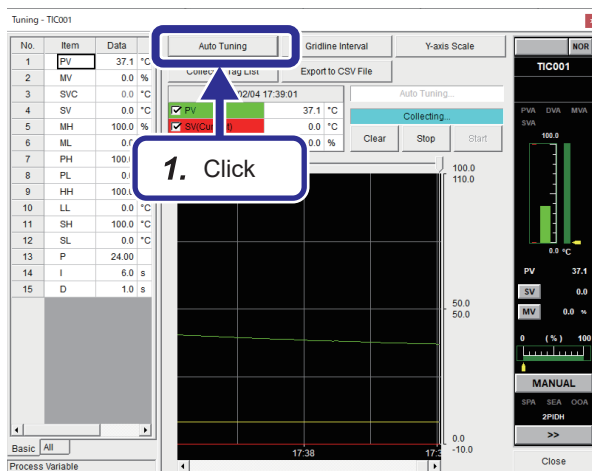
1. Click the [Auto Tuning] button.
If auto tuning is in progress, the "Status of Auto Tuning" window appears as shown in step 4.
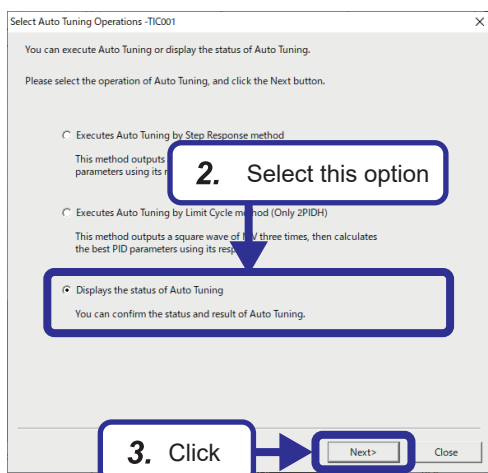
2. The "Select Auto Tuning Operations" window appears.
Select "Displays the status of Auto Tuning".

3. Click the [Next] button.
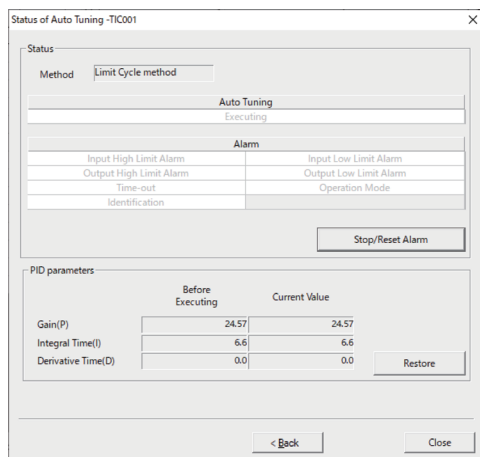
4. The "Status of Auto Tuning" window appears.

■Status of auto tuning window

No.	Item	Description
(1)	Title bar	Displays the name of the tag targeted for auto tuning.
(2)	Method	Displays the auto tuning method. • Step response method: Auto tuning is being executed by the step response method. • Limit cycle method: Auto tuning is being executed by the limit cycle method. • ------: The auto tuning method is not defined.
(3)	Auto Tuning	Indicates the auto tuning status as follows. ■During auto tuning Display: Execution in progress Background color: Light blue Text color: Black ■Auto tuning stopped Display: Execution in progress Background color: White Text color: Gray
(4)	Alarm	Indicates the status of each alarm with the corresponding background color and text color. ■Alarm issued Background color: Red Text color: Black ■No alarm Background color: White Text color: Gray
(5)	Stop/Reset Alarm	Click this button to stop auto tuning and reset the alarms.
(6)	PID constant display	Displays the PID constants before executing auto tuning and the current PID constants.
(7)	Restore	Write the PID constants before executing auto tuning to the programmable controller. The "Restore" button is not available in the following cases. (Disabled status) • In lock mode • When the lockout tag is set • During auto tuning • When PID constants are blank ("------") before executing auto tuning
(8)	Back	Displays the previously displayed window.
(9)	Close	Closes the window.

• Alarm occurrence condition

The following shows the alarms to be displayed in the alarm display field.

Alarm	Step response method	Limit cycle method
Input High Limit Alarm	When either the input upper limit alarm (PHA) or the input upper upper limit alarm (HHA) is ON in the alarm detection (ALM) of the loop tag data	
Input Low Limit Alarm	When either the input lower limit alarm (PLA) or the input lower lower limit alarm (LLA) is ON in the alarm detection (ALM) of the loop tag data	
Output High Limit Alarm	When T1 > MV upper limit value (MH) in T1=MV+AT1 step manipulated value (AT1STEPMV)	Alarms do not occur.[1] (Output value (MV) is controlled below MV upper limit value (MH).)
Output Low Limit Alarm	When T1 > MV lower limit value (ML) in T1=MV+AT1 step manipulated value (AT1STEPMV)	Alarms do not occur.[1] (Output value (MV) is controlled below MV lower limit value (ML).)
Time-out	When the time from the start of auto tuning exceeds the timeout time (AT1TOUT1)	
Operation Mode	When the control mode (MODE) is other than MANUAL	• When the control mode (MODE) is switched during auto tuning • When auto tuning is executed in CASDR mode • When ATTYPE < 0 or ATTYPE > 4 • When AT2MVH $\leq$ AT2MVL
Identification	Response speed for calculation R' = Maximum ramp value (%)/Sampling period for AT1 Response speed R = |R'|/100(s-1) When the above is calculated and the result is R $\leq$ 0	When PID constants cannot be calculated • When Xc = AT2HS • When the calculation result PID constants are: gain > 999 and integral time/derivative time > 9999.

[1] With auto tuning by the limit cycle method, MHA, MLA, and DMLA of ALM tag are cleared.

Fine tuning of PID constants

After auto tuning is completed, the change of process variable (PV) in relation to the setting value (SV) is observed on the tuning setting execution window of the PX Developer monitor tool. The optimal values are obtained by adjusting the P, I, and D values.

The response of the process variable (PV) corresponding to the change of the setting value (SV) is observed.

• Response is quick, but oscillatory.

Fine tuning when the response is quick, but oscillatory
  • Proportional gain: Reduce (Proportional effect is reduced.)
  • Integral time: Increase (Integral effect is reduced.)

• Optimal value

• Response is slow

Fine tuning when the response is slow
  • Proportional gain: Increase (Proportional effect is increased.)
  • Integral time: Reduce (Integral effect is increased.)

When the derivative action is applied, the derivative time is adjusted with the stability and rapid response check. (Increasing the derivative time increases the derivative effect, and reducing the derivative time reduces the derivative effect.)

6.3 Monitoring via GOT

This section describes the procedure for monitoring the GOT and redundant system via Ethernet by using the redundant system monitor system of the GOT in the following figure.

The PX Developer monitor tool has the function to automatically generate a GOT screen. The following describes the procedure for using the PX Developer monitor tool to automatically generate a window and using the GOT to monitor the redundant system.
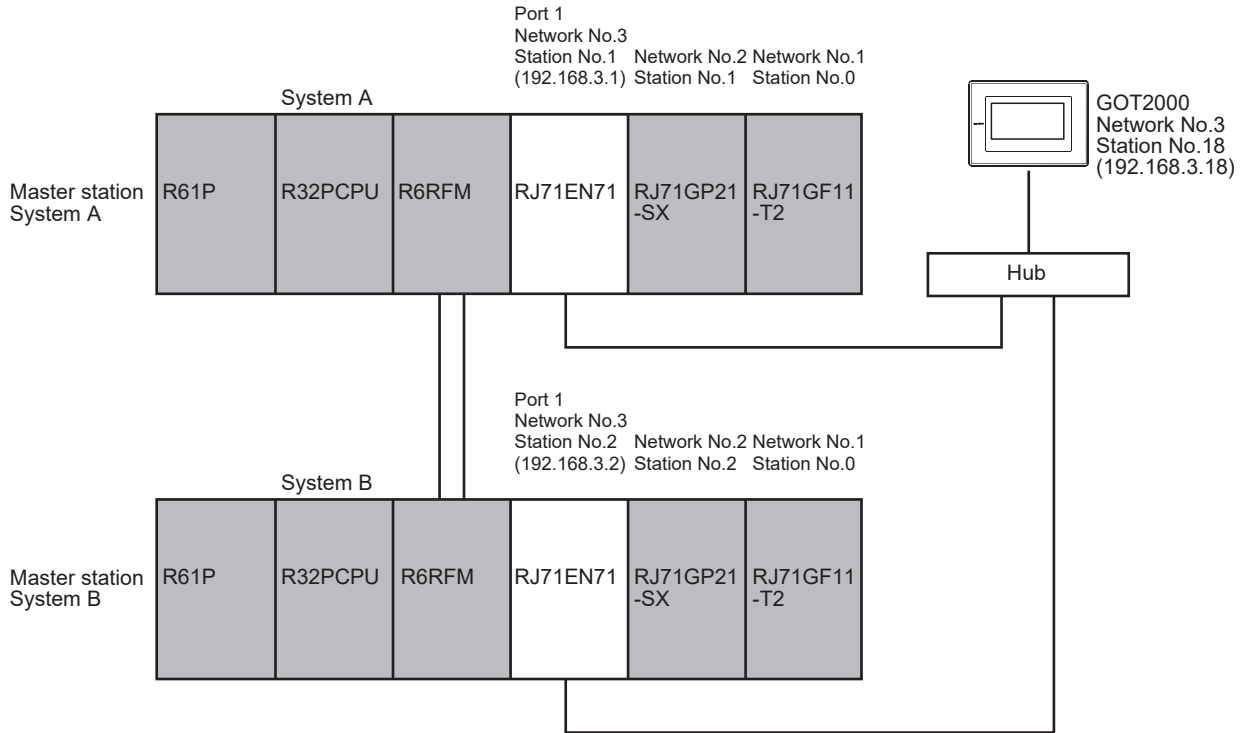
Auto generation of GOT screen

By using the PX Developer monitor tool and GT Designer3 in combination, a GOT screen with faceplates, trends, and alarms can be automatically generated.

1. Read the assignment information database created in "Page 125 Creating an assignment information database file" with the PX Developer monitor tool (read in "☞ Page 134 Communications settings of PX Developer monitor tool"), and generate a GOT screen.

2. Convert the screen project generated by GT Designer3 in accordance with the demonstration machine model.

3. Configure connection with the redundant system.

Generating a GOT screen

Generate a GOT screen for the GOT2000 series.

Operating procedure

1. Click the [Setting Window] icon ( ).
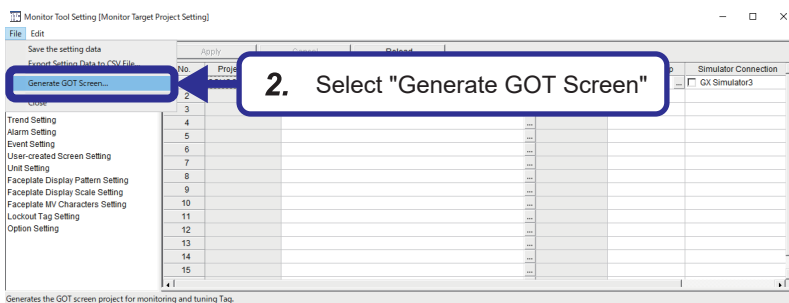
1. Click

2. Select [File] ⇨ [Generate GOT Screen] to start GOT screen generation.

2. Select "Generate GOT Screen"
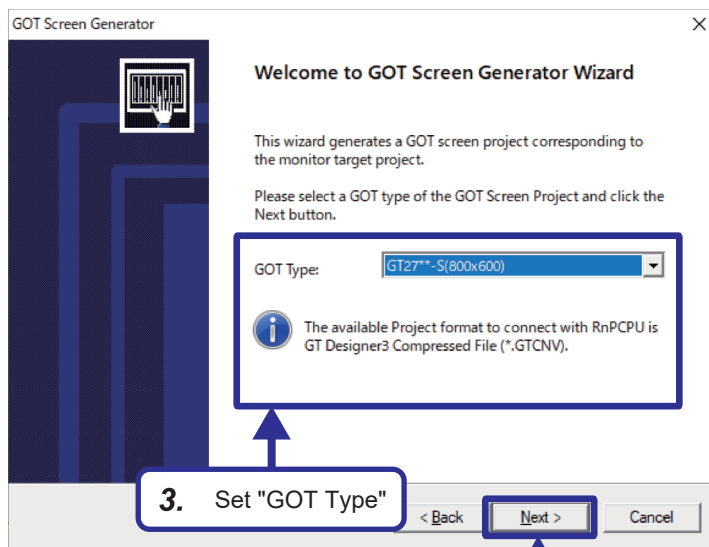
3. The GOT screen generator wizard appears.
Set "GOT Type" as follows.
[Settings]
GOT Type: GT27**-S (800 × 600 dots)

4. Click the [Next] button.

3. Set "GOT Type"
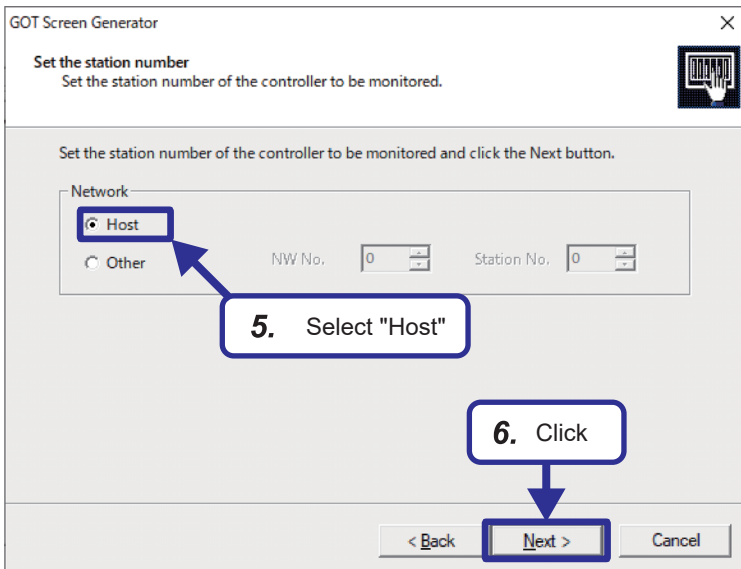
4. Click

5. Select "Host" for the station number of the connected device of the monitor destination.

6. Click the [Next] button.
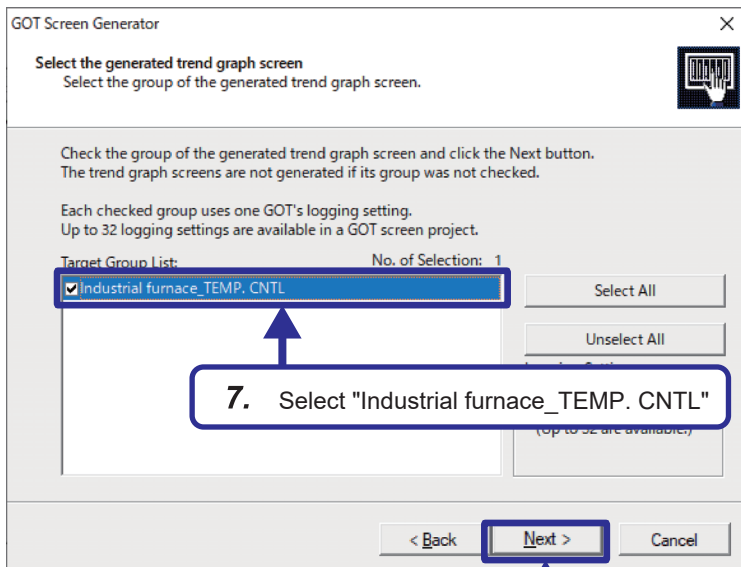
7. For selection of trend graph screens to be generated, select "Industrial furnace_TEMP.CNTL" in "Target Group List".

8. Click the [Next] button.
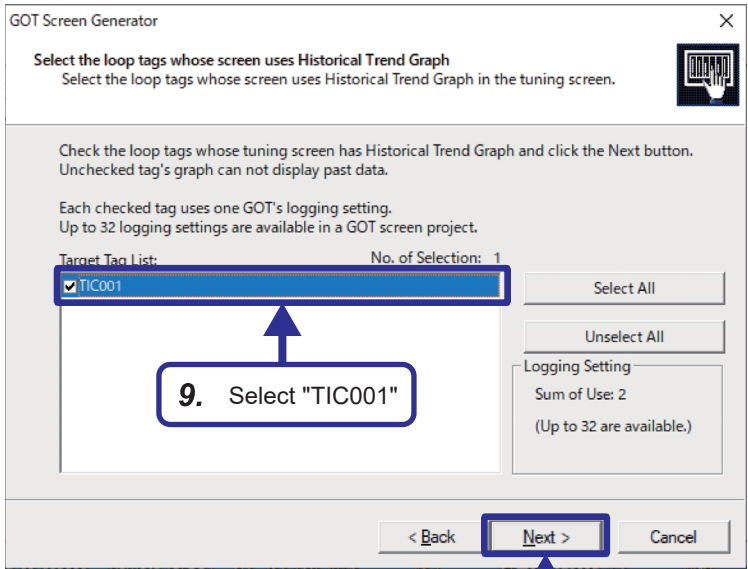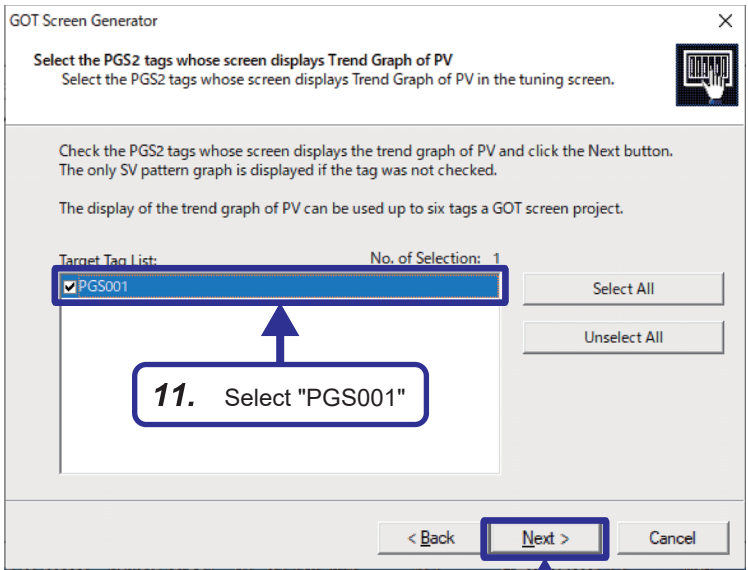
9. For selection of loop tags that uses historical trend graphs, select "TIC001" in "Target Group List".

10. Click the [Next] button.

11. For selection of PGS2-type tags for displaying the trend of PV values, select "PGS001" in "Target Group List".

12. Click the [Next] button.

6
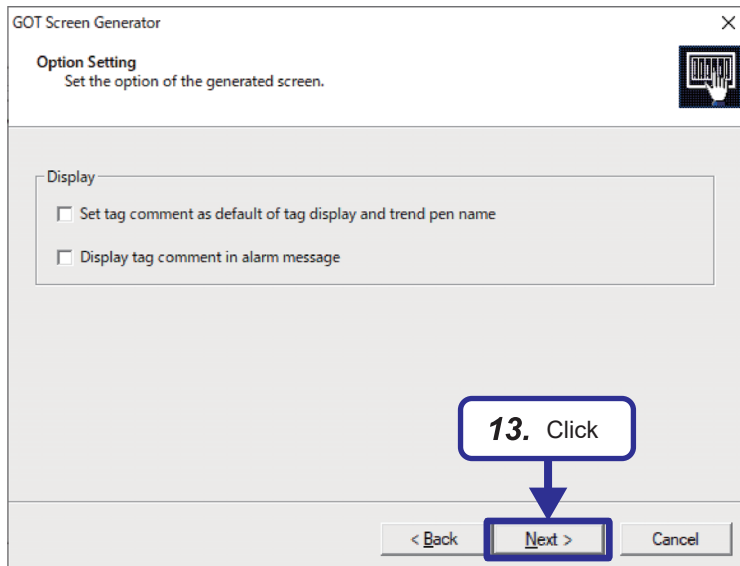
13. For the option setting, leave checkboxes unchecked and click the [Next] button.
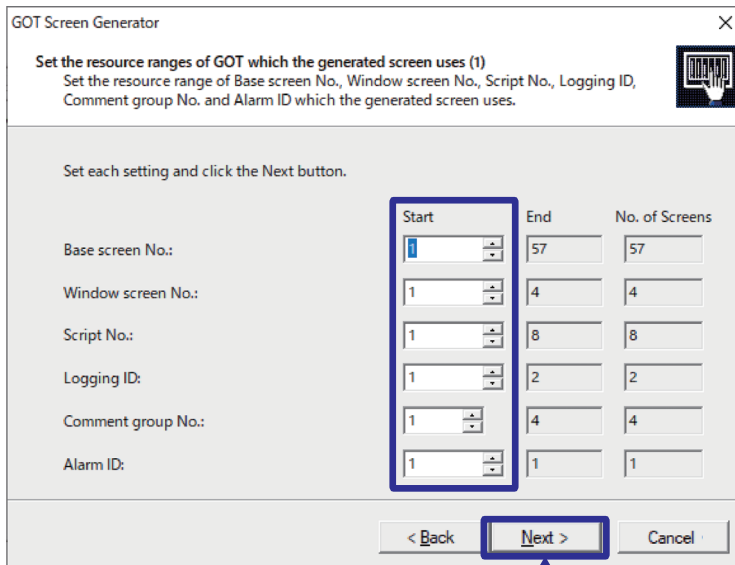
14. For all the settings in "Set the resource ranges of GOT which the generated screen uses (1)", leave the default values as it is and click the [Next] button.
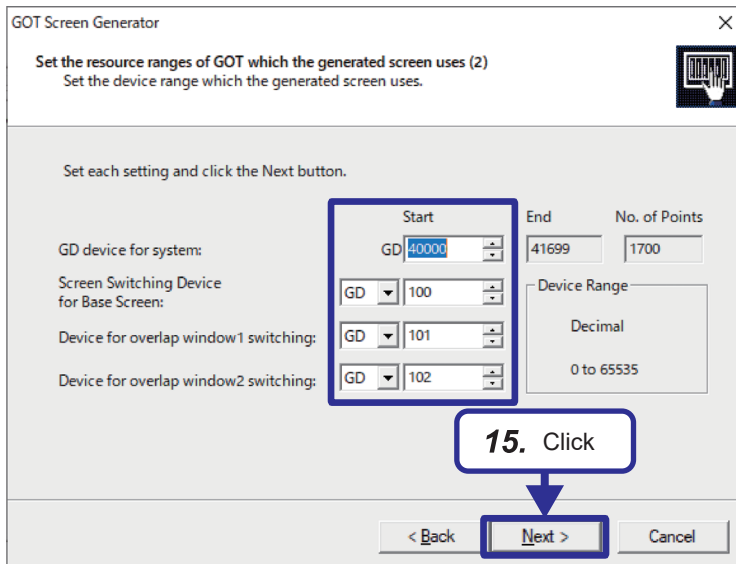
15. Also for all the settings in "Set the resource ranges of GOT which the generated screen uses (2)", leave the default values as it is and click the [Next] button.
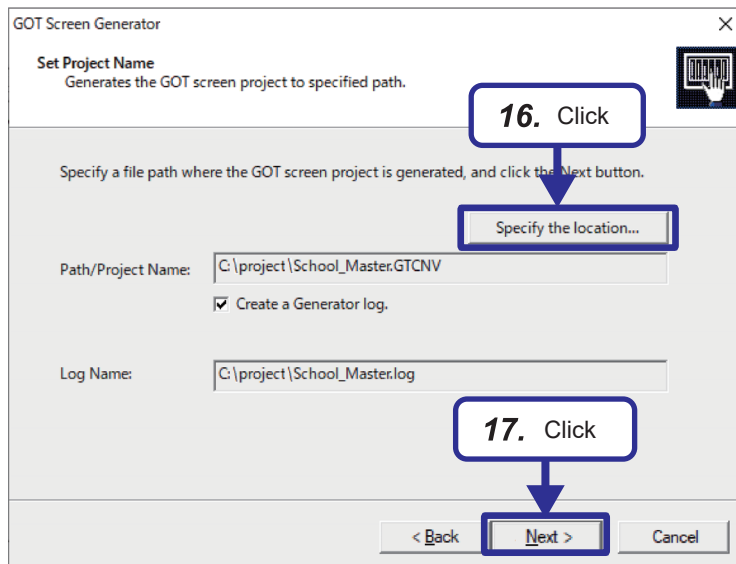
16. Click the [Specify the location] button and specify the project generation destination.

17. Click the [Next] button.
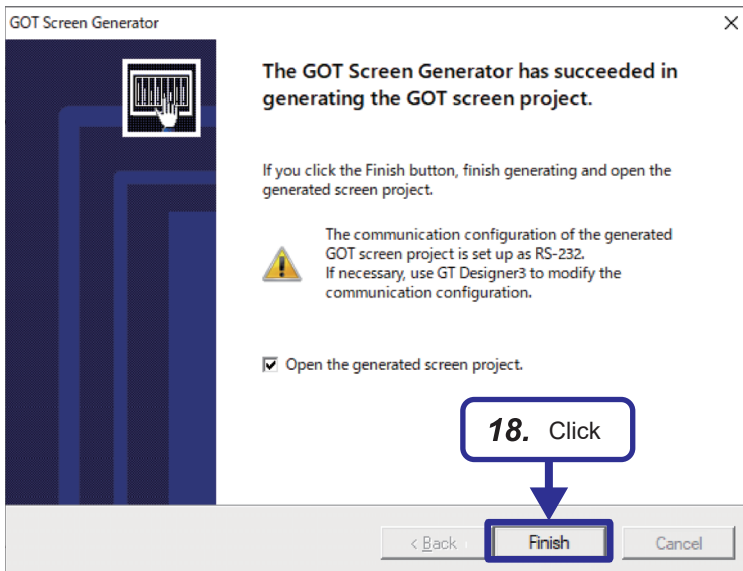
18. When GOT screen project generation is complete, the window on the left is displayed.
Click the [Finish] button to close the wizard.

Converting the generated screen to the GOT model of the demonstration machine

GOT2000 data generated by the auto-generation function is compatible with model GT27**-S (800 × 600). Use GT Designer3 to convert the data to the GOT size of the demonstration machine.

Operating procedure
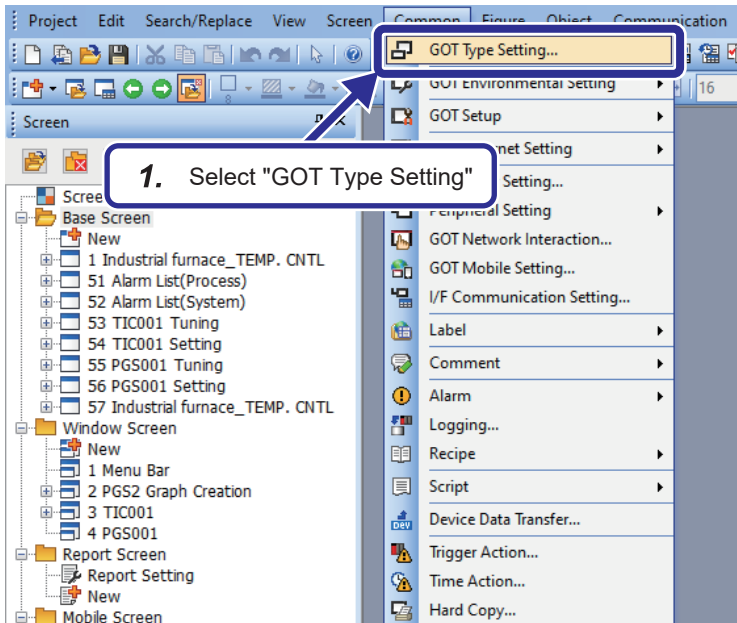
1. Select [Common] ⇨ [GOT Type Setting] from the menu bar.
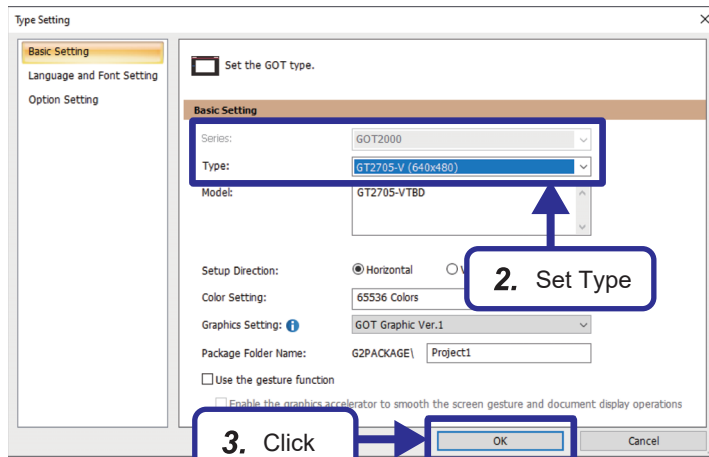
2. On the "Type Setting" window, configure the settings as follows.
   [Settings]
   Type: GT2705-V (640 × 480)

3. Click the [OK] button to execute conversion.
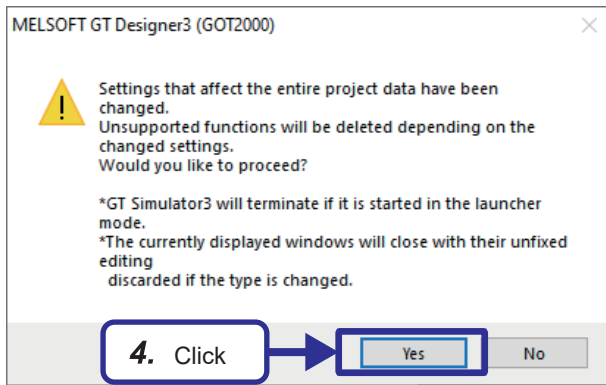
4. Click the [Yes] button.
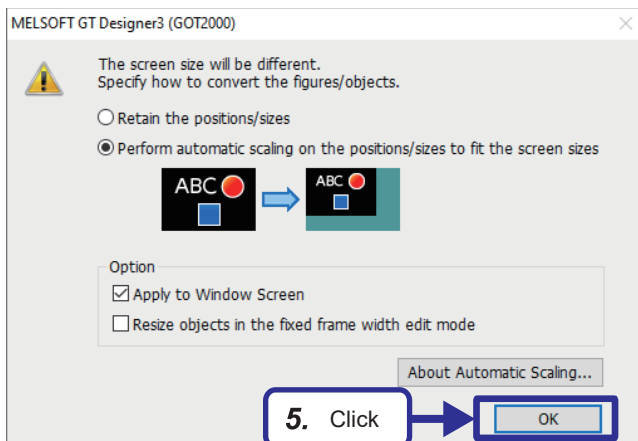
5. Select the items as shown in the figure, press the [OK] button.
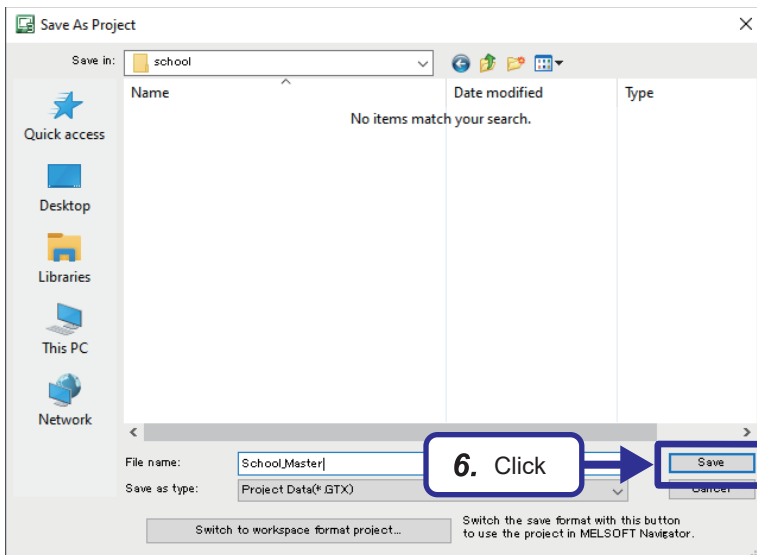
6. Save the generated screen data (.GTCNV).
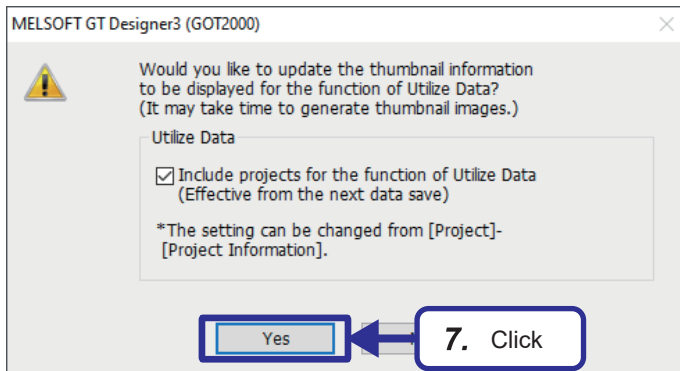
7. Click the [Yes] button.

Point

The layout may be broken when the GOT size is changed due to model conversion. In this case, edit the window in GT Designer3.

GOT connection setting

Use GT Designer3 to configure the settings for connecting to the redundant system.

Operating procedure
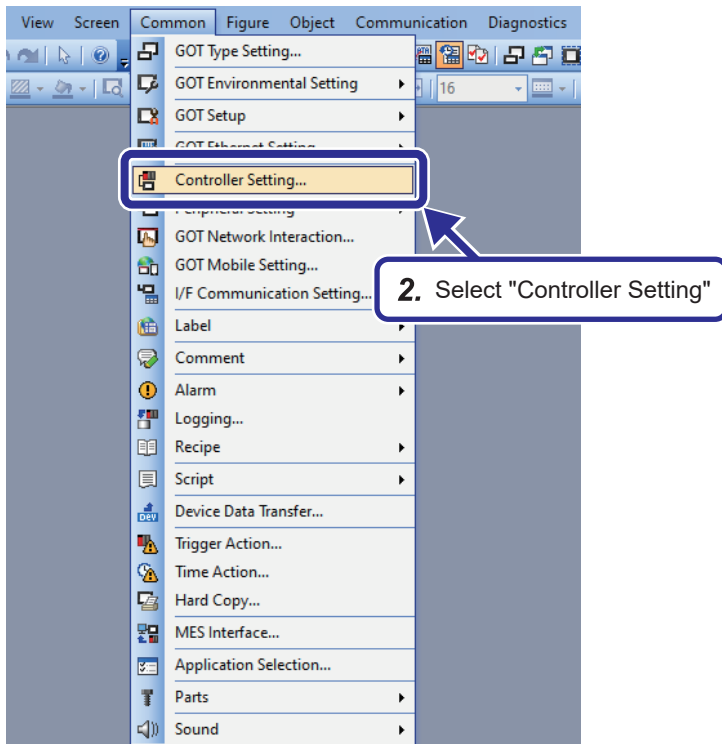
1. Start up the generated data in GT Designer3.

2. Select [Common] ⇨ [Controller Setting] from the menu bar.
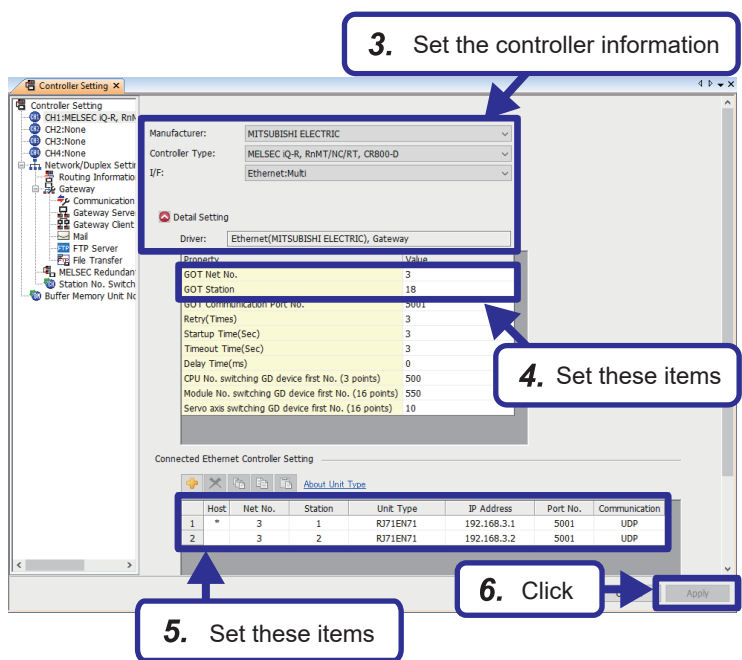
3. Set the connected device as follows.
[Settings]
Manufacturer: MITSUBISHI ELECTRIC
Controller Type: MELSEC iQ-R, RnMT/NC/RT, CR800-D
I/F: Ethernet:Multi
Driver: Ethernet(MITSUBISHI ELECTRIC), Gateway

4. Configure the detailed settings as follows.
[Settings]
GOT Net No.: 3
GOT Station: 18

5. Click the [+] button under "Connected Ethernet Controller Setting", add a setting line for the second device, and configure the settings as follows. (The "Port No." and "Communication" values are fixed.)
[Settings (first device]
Net No.: 3
Station: 1
Unit Type: RJ71EN71
IP Address: 192.168.3.1
[Settings (second device)]
Net No.: 3
Station: 2
Unit Type: RJ71EN71
IP Address: 192.168.3.2
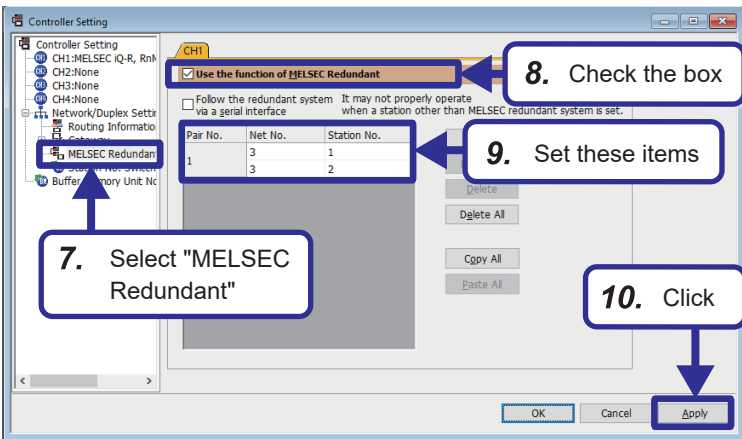
6. Click the [Apply] button.

7. Under "Controller Setting", select "MELSEC Redundant".

8. Select the "Use the function of MELSEC Redundant" checkbox.

9. For Pair No.1, set "Net No." to "3", and set "Station No." to "1" and "2".

10. Click the [Apply] button.

11. Overwrite the project.

12. Write data to the GOT.

Checking auto tuning operation

This section describes how to perform auto tuning by the limit cycle method using GOT. Check the operation after auto tuning.

Auto tuning by the limit cycle method

Operating procedure

1. Touch the [Detail] button on the "TIC001" faceplate.
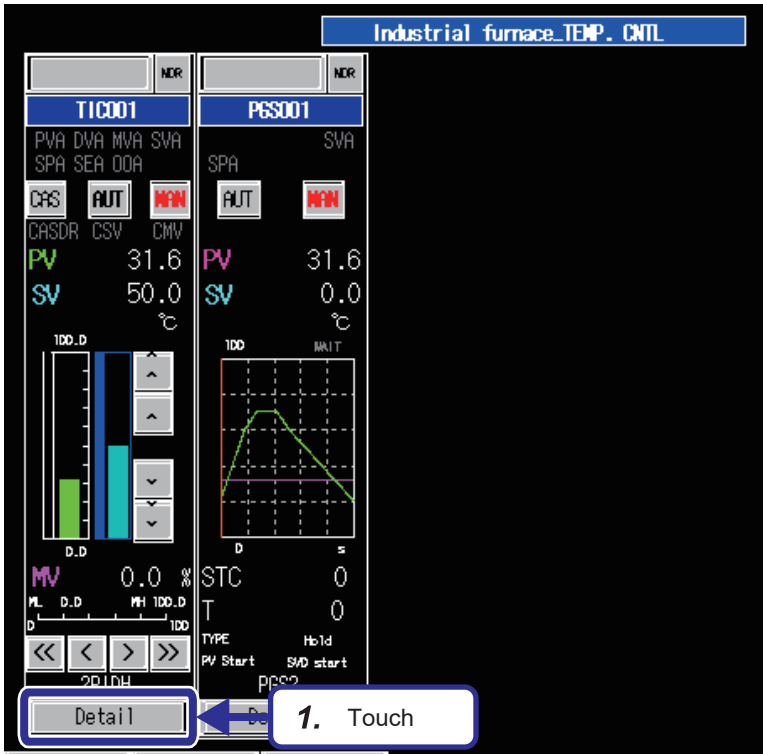
2. Touch the [Start Trend] button.
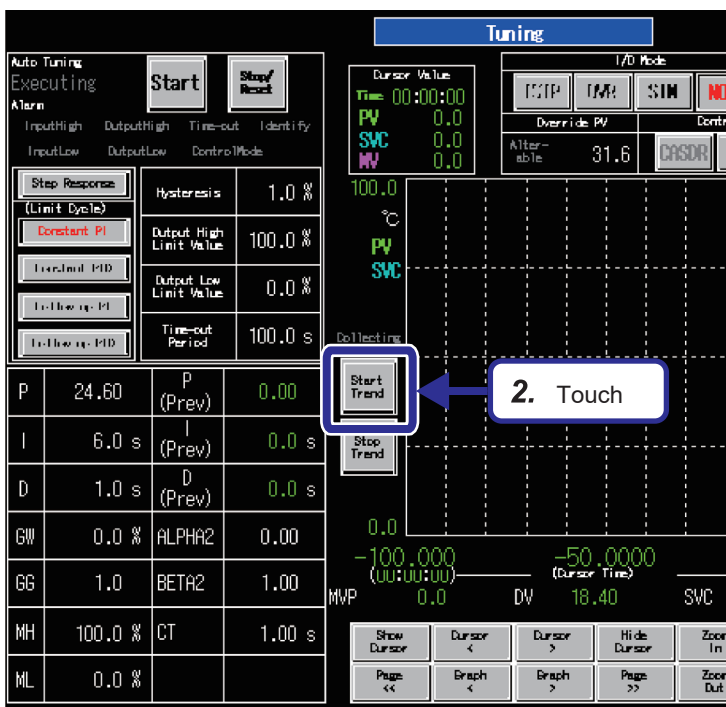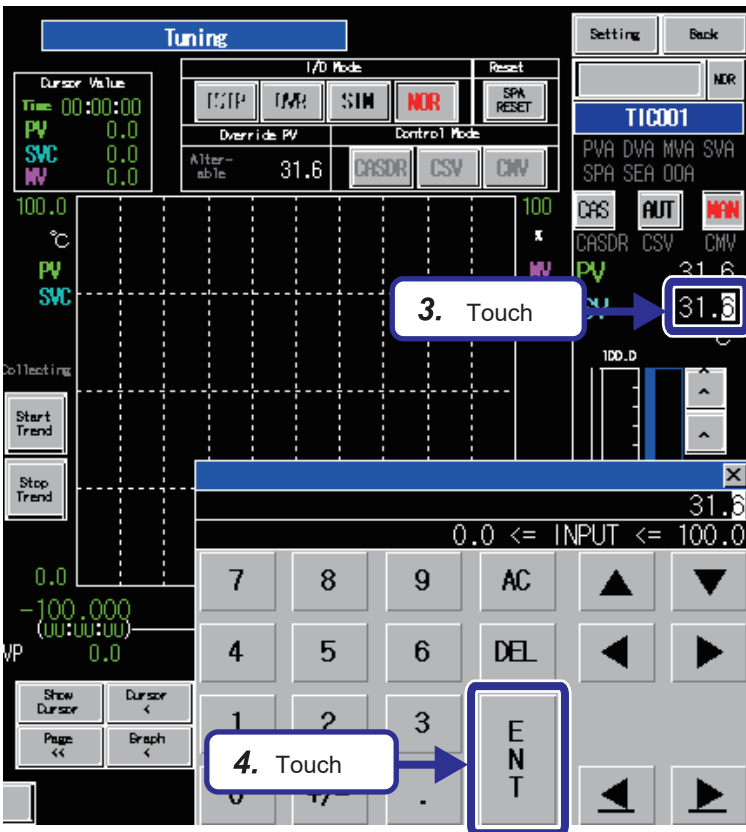
3. Touch the SV value.

4. Set the SV value to the same value as the ambient temperature, and touch the [ENT] button.
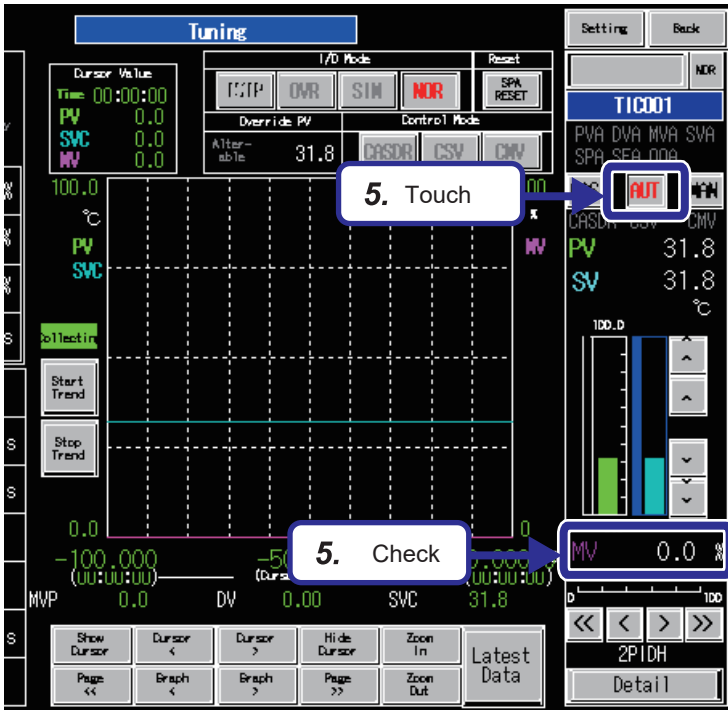
5. Touch the [AUT] button.
Check that the MV value is 0%.
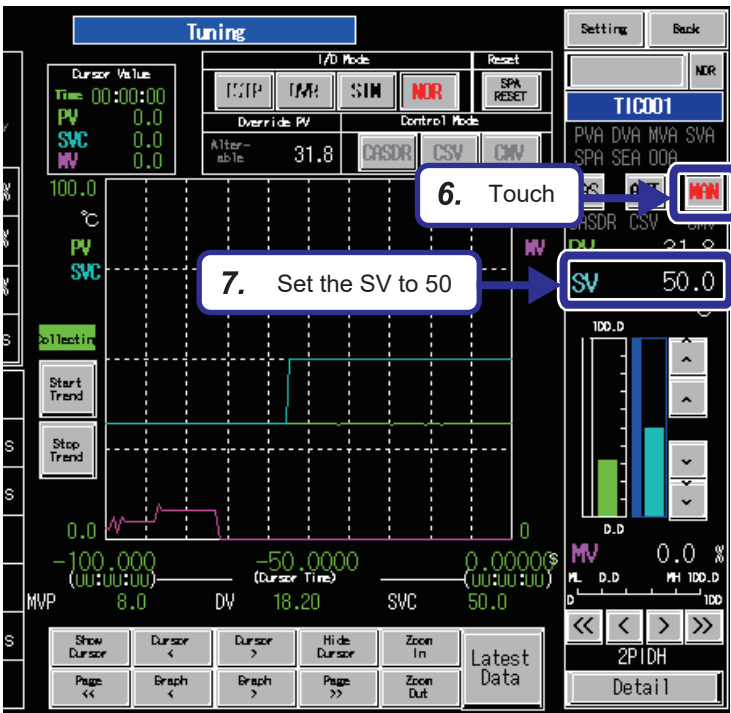
6. Touch the [MAN] button.

7. Set the SV value to 50.
Check that the MV value is 0%.
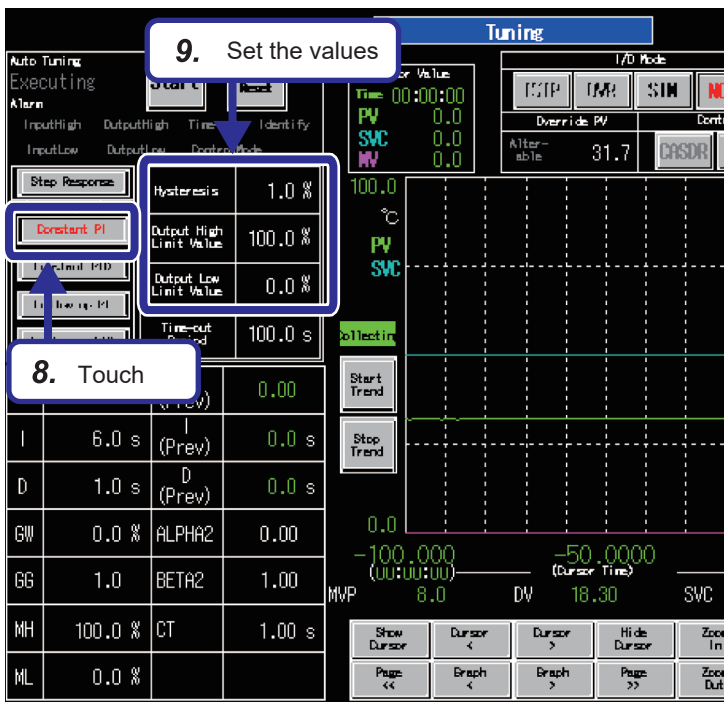
8. Touch the [Constant PI] button of the limit cycle method.

9. Configure the settings as follows.
[Settings]
Hysteresis: 1
Output High Limit Value: 100
Output Low Limit Value: 0
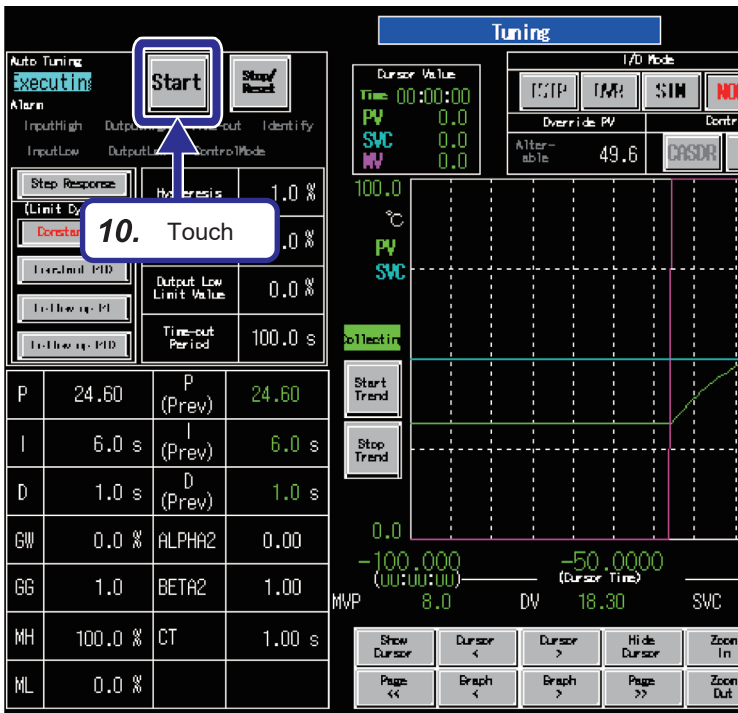
10. Touch the [Start] button of auto tuning. "Executing" is highlighted in blue during auto tuning.
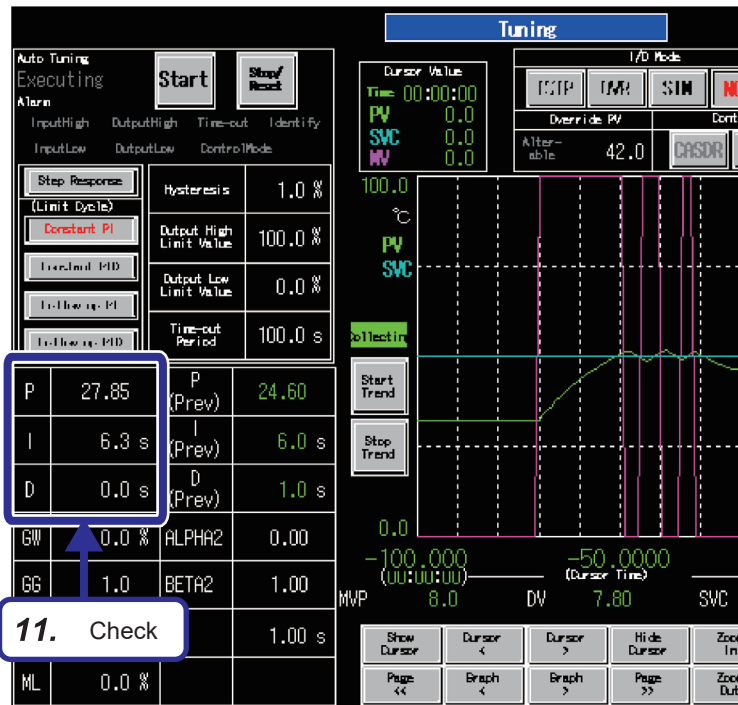
11. After auto tuning is complete, the PID values that is calculated by auto tuning are stored.

6.4 Checking the Operation

Check the difference of operation before and after the PID values are changed. In this exercise, the operation is checked using GOT.

Operating procedure

1. By using the procedures in "☞ Page 118 Changing FB property values online" and "Page 121 Applying FB property values to the project" change the FB property initial values as follows.
[Changes]
PROPORTIONAL: 1
INTEGRAL: 2
DERIVATIVE: 0
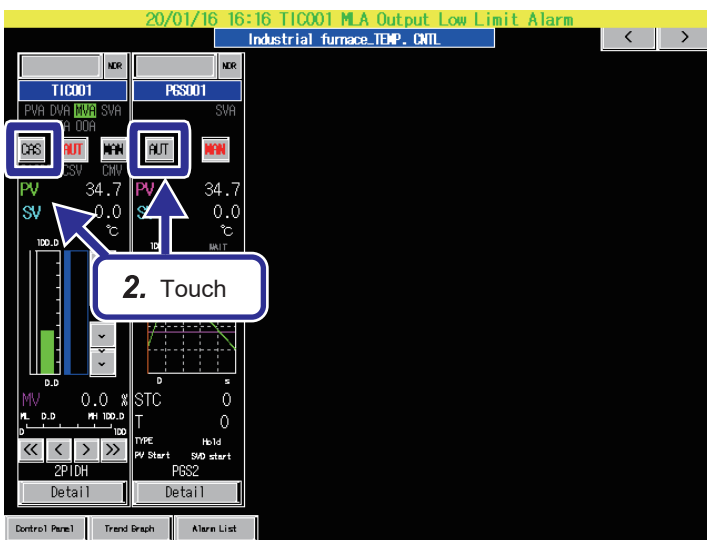
2. On the GOT screen, touch the [AUT] button of "PGS001", and touch the [CAS] button of "TIC001".
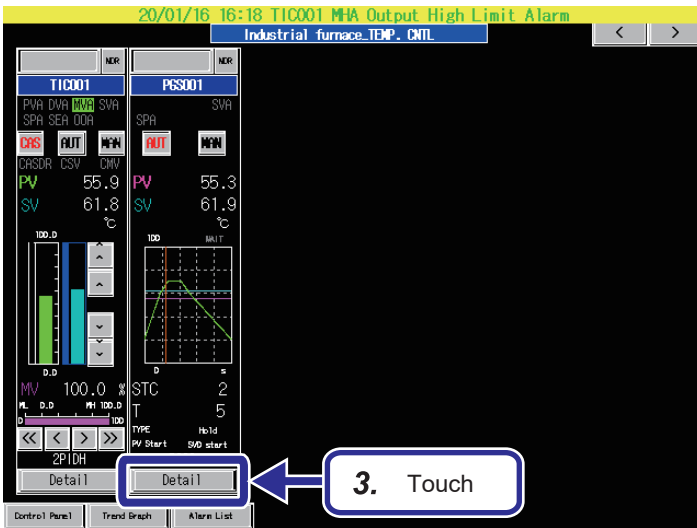
3. Touch the [Detail] button of "PGS001" and move to the tuning screen.

4. Check the graph behavior.
It takes some time for the heater to cool down. Therefore, while the temperature is decreasing, the graph may not be consistent with the setting value.
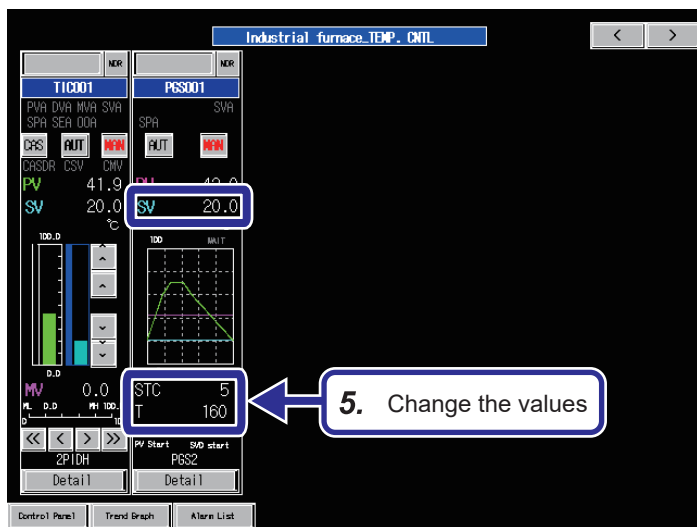
5. Set the SV, STC, and T of "PGS001" to 0.
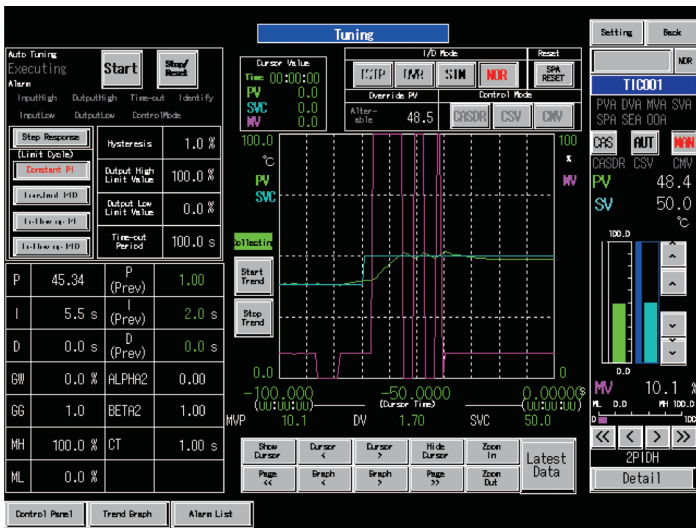
6. Perform auto tuning using the procedure in "☞ Page 168 Checking auto tuning operation".

7. After auto tuning, wait until the PV value becomes the same as the ambient temperature. Perform steps *2.* to *4.* using the calculated PID values.

8. Check the graph behavior.
   Compared to step *4.*, the graph is precisely consistent with the setting value.

APPENDICES

Appendix 1 Lists of Process Control Function Blocks

How to read the lists is shown below.

Item	Description
Function block symbol	A function block name
Processing details	An overview of the function blocks

> **Point**
>
> - For instructions for the CPU module, dedicated instructions for the intelligent function modules, and standard functions/function blocks, refer to the following.
>
>   📖 MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks)
> - For module FBs specific to each module, refer to the following.
>
>   📖 Function Block Reference for the module used

General process FBs

Analog value selection/average

The following function blocks output the maximum, minimum, middle, average, and absolute values of input values.

■High selector

Function block symbol	Processing details
M+P_HS(_E)	Outputs the maximum value of input values (array).

■Low selector

Function block symbol	Processing details
M+P_LS(_E)	Outputs the minimum value of input values (array).

■Middle value selection

Function block symbol	Processing details
M+P_MID(_E)	Outputs the middle value of input values (array).

■Average value

Function block symbol	Processing details
M+P_AVE(_E)	Outputs the average value of input values (array).

■Absolute value

Function block symbol	Processing details
M+P_ABS(_E)	Outputs the absolute value of input values.

A

Correction operation

The following function blocks perform operation processing such as broken line correction, standard filter, engineering value conversion, temperature/pressure correction, and summation.

■Function generator

Function block symbol	Processing details
M+P_FG	Outputs (OUT_) values according to the function generator pattern that consists of SN break points with regard to the input (IN).

■Inverse function generator

Function block symbol	Processing details
M+P_IFG	Outputs (OUT_) values according to the function generator pattern that consists of SN break points with regard to the input (IN).

■Standard filter (moving average)

Function block symbol	Processing details
M+P_FLT	Outputs (OUT_) the average value of SN pieces of data of the input (IN) collected at the data collection interval ST.

■Engineering value conversion

Function block symbol	Processing details
M+P_ENG	Converts a value (%) of the input (IN) in percentage into an engineering value such as temperature and pressure and outputs (OUT_) the result.

■Engineering value inverse conversion

Function block symbol	Processing details
M+P_IENG	Converts an input engineering value (IN) such as temperature and pressure into a value in percentage (%) and outputs (OUT_) the result.

■Temperature/pressure correction

Function block symbol	Processing details
M+P_TPC	Performs temperature/pressure correction (or either of temperature correction or pressure correction) to the input (IN) of the differential pressure (%) and outputs (OUT_) the result.

■Summation

Function block symbol	Processing details
M+P_SUM	Performs integration processing to the input (IN) when the integration start signal (START) is TRUE, and outputs (OUT_) the result.

■Summation (internal integer integration)

Function block symbol	Processing details
M+P_SUM2_	Performs integration processing to the input (IN) when the integration start signal (START) is TRUE, and outputs the result. Internal integration for the integral part is executed with signed 32-bit integers.

■Range conversion

Function block symbol	Processing details
M+P_RANGE_	Performs range conversion to the input (IN) and outputs (OUT_) the result.

Arithmetic operation

The following function blocks perform operation processing such as addition/subtraction, multiplication/division, and square root.

■Addition (with coefficient)

Function block symbol	Processing details
M+P_ADD	Adds values with coefficients and bias to the inputs (IN1 to IN5) and outputs (OUT_) the result.

■Subtraction (with coefficient)

Function block symbol	Processing details
M+P_SUB	Subtracts values with coefficients and bias from the values of the inputs (IN1 to IN5) and outputs (OUT_) the result.

■Multiplication (with coefficient)

Function block symbol	Processing details
M+P_MUL	Multiplies the values of the inputs (IN1 to IN5) with coefficients and bias and outputs (OUT_) the result.

■Division (with coefficient)

Function block symbol	Processing details
M+P_DIV	Divides the values of the inputs (IN1, IN2) with coefficients and bias and outputs (OUT_) the result.

■Square root (with coefficient)

Function block symbol	Processing details
M+P_SQR	Performs square root extraction with coefficients to the value of the input (IN) and outputs (OUT_) the result.

Comparison operation

The following function blocks perform comparison operation ($\geq$, $>$, $=$, $<$, $\leq$).

■Comparison (>) with setting value

Function block symbol	Processing details
M+P_GT	Compares (>) the input 1 (IN1) with the input 2 (IN2) using a setting value and hysteresis and outputs the result to the comparison output (B_). This function block always outputs values to the input 1 (IN1) from the output (OUT_).

■Comparison (<) with setting value

Function block symbol	Processing details
M+P_LT	Compares (<) the input 1 (IN1) with the input 2 (IN2) using a setting value and hysteresis and outputs the result to the comparison output (B_). This function block always outputs values to the input 1 (IN1) from the output (OUT_).

■Comparison (=) with setting value

Function block symbol	Processing details
M+P_EQ	Compares (=) the input 1 (IN1) with the input 2 (IN2) using a setting value and outputs the result to the comparison output (B_). This function block always outputs values to the input 1 (IN1) from the output (OUT_).

■Comparison (>=) with setting value

Function block symbol	Processing details
M+P_GE	Compares ($\geq$) the input 1 (IN1) with the input 2 (IN2) using a setting value and hysteresis and outputs the result to the comparison output (B_). This function block always outputs values to the input 1 (IN1) from the output (OUT_).

■Comparison (<=) with setting value

Function block symbol	Processing details
M+P_LE	Compares ($\leq$) the input 1 (IN1) with the input 2 (IN2) using a setting value and hysteresis and outputs the result to the comparison output (B_). This function block always outputs values to the input 1 (IN1) from the output (OUT_).

Control operation

The following function blocks control operation of lead-lag, integration, derivative, high/low limiter, variation rate limiter, dead band, bumpless transfer, and analog memory.

■Lead-lag

Function block symbol	Processing details
M+P_LLAG	Performs lead-lag compensation to the input (IN) when the operation signal (INVLD) is FALSE, and outputs (OUT_) the result.

■Integration

Function block symbol	Processing details
M+P_I	Performs integral operation to the input (IN) when the operation signal (INVLD) is FALSE, and outputs (OUT_) the result.

■Derivative

Function block symbol	Processing details
M+P_D	Performs derivative operation to the input (IN) when the operation signal (INVLD) is FALSE, and outputs (OUT_) the result.

■Dead time

Function block symbol	Processing details
M+P_DED	Outputs (OUT_) the result with a delay by the dead time to the input (IN) when the operation signal (INVLD) is FALSE.

■High/low limiter

Function block symbol	Processing details
M+P_LIMT	Applies high/low limiters with hysteresis to the input (IN) and outputs (OUT_) the result.

■Variation rate limiter 1

Function block symbol	Processing details
M+P_VLMT1	Limits the variation speed to the input (IN) and outputs (OUT_) the result.

■Variation rate limiter 2

Function block symbol	Processing details
M+P_VLMT2	Limits the output variation speed to the input (IN) and outputs (OUT_) the result.

■Dead band

Function block symbol	Processing details
M+P_DBND	Sets a dead band to the input (IN) and outputs (OUT_) the result.

■Bump-less transfer

Function block symbol	Processing details
M+P_BUMP	Changes the output (OUT_) from the output control value CTRLV to the output setting value SETV smoothly when the mode (MODE) is changed from FALSE (MANUAL) to TRUE (AUTO).

■Analog memory

Function block symbol	Processing details
M+P_AMR	Increases or decreases the output (OUT_) at a fixed rate.

■8 points time proportional output

Function block symbol	Processing details
M+P_DUTY_8PT_	Performs output ON time conversion for input values and outputs the result in bits. This function block also adjusts the phase of output cycles automatically to suppress overlapping of output (for example, peak current).

A

Tag access FBs

I/O control operation

The following function blocks perform I/O processing including analog input/output, pulse integration, and batch counter.

■Analog input processing

Function block symbol	Processing details
M+P_IN	Performs range check, input limiter, engineering value inverse conversion, and digital filter processing. The input limiter processing can be enabled or disabled on the "Options" window of the engineering tool.

■Output processing-1 with mode switching (with input addition)

Function block symbol	Processing details
M+P_OUT1	Performs input addition, variation rate & high/low limiter, reset windup, and output conversion processing to an input value (ΔMV) and outputs a manipulated value. (with integral and anti-reset windup processing)

■Output processing-2 with mode switching (without input addition)

Function block symbol	Processing details
M+P_OUT2	Performs variation rate & high/low limiter processing and output conversion processing to an input value (MV), and outputs a manipulated value. (without integral and anti-reset windup processing)

■Output processing-3 with mode switching (with input addition and compensation)

Function block symbol	Processing details
M+P_OUT3_	Performs input addition, MV compensation, preset MV, MV hold, MV tracking, variation rate & high/low limiter, reset windup, tight shut/full open, MV reverse, and output conversion processing to an input value (ΔMV) and outputs a manipulated value.

■Manual output

Function block symbol	Processing details
M+P_MOUT	Reads a manipulated value (MV) of tag data, performs output conversion processing, and outputs a manipulated value.

■Time proportioning output

Function block symbol	Processing details
M+P_DUTY	Performs input addition, variation rate & high/low limiter, reset windup, output ON time conversion, and output conversion processing to an input value (ΔMV) and outputs the value as bits.

■Pulse integrator

Function block symbol	Processing details
M+P_PSUM	Performs input value increment operation, integrated value calculation, and output conversion processing to a count value (CIN) and outputs the result when the integration start signal (RUN) is TRUE.

■Batch counter

Function block symbol	Processing details
M+P_BC	Compares the input (CIN) with the estimated value 1 and estimated value 2, and outputs a completion signal when the input reaches an estimated value. At this time, this function block performs the high limit check, variation rate check, and output conversion processing to the input (CIN).

■Manual setter

Function block symbol	Processing details
M+P_MSET_	Performs SV variation rate & high/low limiter processing, sets the result as the set value (current) (SVC) of tag data, and outputs the value (SVC).

Loop control operation

The following function blocks perform loop control processing including ratio control, various PID controls, two-position (on/off) control, three-position (on/off) control, program setter, and loop selector.

■Ratio control (enable tracking for primary loop)

Function block symbol	Processing details
M+P_R_T	Controls two sets of control amount at a constant ratio and outputs the result (ΔMV). The primary loop can be tracked.

■Ratio control (disable tracking for primary loop)

Function block symbol	Processing details
M+P_R	Controls two sets of control amount at a constant ratio and outputs the result (ΔMV).

■Velocity type PID control (enable tracking for primary loop)

Function block symbol	Processing details
M+P_PID_T	Performs PID operations using process variable differential, inexact differential, and velocity type, and outputs the result (ΔMV). The primary loop can be tracked.

■Velocity type PID control (disable tracking for primary loop)

Function block symbol	Processing details
M+P_PID	Performs PID operations using process variable differential, inexact differential, and velocity type, and outputs the result (ΔMV).

■2-degree-of-freedom PID control (enable tracking for primary loop)

Function block symbol	Processing details
M+P_2PID_T	Optimizes the responsive performance (tracking performance) in a setting value change and control performance to a disturbance, and outputs the result (ΔMV). The primary loop can be tracked.

■2-degree-of-freedom PID control (disable tracking for primary loop)

Function block symbol	Processing details
M+P_2PID	Optimizes the responsive performance (tracking performance) in a setting value change and control performance to a disturbance, and outputs the result (ΔMV).

■2-degree-of-freedom advanced PID control (enable tracking for primary loop)

Function block symbol	Processing details
M+P_2PIDH_T_	Optimizes the responsive performance (tracking performance) in a setting value change and control performance to a disturbance, and outputs the result (ΔMV). This function block performs two-degree-of-freedom PID operation, PV tracking, integration stop, derivative stop, and SV variation rate & high/low limiter processing. The primary loop can be tracked.

■2-degree-of-freedom advanced PID control (disable tracking for primary loop)

Function block symbol	Processing details
M+P_2PIDH_	Optimizes the responsive performance (tracking performance) in a setting value change and control performance to a disturbance, and outputs the result (ΔMV). This function block performs two-degree-of-freedom PID operation, PV tracking, integration stop, derivative stop, and SV variation rate & high/low limiter processing.

■Position type PID control (enable tracking for primary loop/disable tracking from secondary loop)

Function block symbol	Processing details
M+P_PIDP_T	Performs PID operations using process variable differential, inexact differential, and position type, and outputs the result. The primary loop can be tracked.

■Position type PID control (disable tracking for primary loop/disable tracking from secondary loop)

Function block symbol	Processing details
M+P_PIDP	Performs PID operations using process variable differential, inexact differential, and position type, and outputs the result.

■Position type PID control (enable tracking for primary loop/enable tracking from secondary loop)

Function block symbol	Processing details
M+P_PIDP_EX_T_	Performs PID operations using process variable differential, inexact differential, and position type, and outputs the result. It also allows manipulated value bumpless switching and tracking from the primary loop and secondary loop at a change of control mode.

■Position type PID control (disable tracking for primary loop/enable tracking from secondary loop)

Function block symbol	Processing details
M+P_PIDP_EX_	Performs PID operations using process variable differential, inexact differential, and position type, and outputs the result. It also allows manipulated value bumpless switching and tracking from the secondary loop at a change of control mode.

■Sample PI control (enable tracking for primary loop)

Function block symbol	Processing details
M+P_SPI_T	Performs PI control during operating time (ST_) and outputs the result (ΔMV). This function block holds the output (ΔMV = 0) during hold time (HT). The primary loop can be tracked.

■Sample PI control (disable tracking for primary loop)

Function block symbol	Processing details
M+P_SPI	Performs PI control during operating time (ST_) and outputs the result (ΔMV). This function block holds the output (ΔMV = 0) during hold time (HT).

■I-PD control (enable tracking for primary loop)

Function block symbol	Processing details
M+P_IPD_T	Controls values to slowly respond to a change of the setting value (SV) without any shocks and outputs the result (ΔMV) by using the process variable (PV) in the proportional term and derivative term. The primary loop can be tracked.

■I-PD control (disable tracking for primary loop)

Function block symbol	Processing details
M+P_IPD	Controls values to slowly respond to a change of the setting value (SV) without any shocks and outputs the result (ΔMV) by using the process variable (PV) in the proportional term and derivative term.

■Blend PI control (enable tracking for primary loop)

Function block symbol	Processing details
M+P_BPI_T	This function block is used when the control amount vibrates in a short period but is stable in a long period. The primary loop can be tracked.

■Blend PI control (disable tracking for primary loop)

Function block symbol	Processing details
M+P_BPI	This function block is used when the control amount vibrates in a short period but is stable in a long period.

■High/Low limit alarm check

Function block symbol	Processing details
M+P_PHPL	Performs the high high limit/high limit/low limit/low low limit checks and the variation rate check to the input (PVPI) and outputs the results. If a value has exceeded an allowable range, an alarm occurs.

■2-position ON/OFF control (enable tracking for primary loop)

Function block symbol	Processing details
M+P_ONF2_T	Performs two-position (on/off) control. The primary loop can be tracked.

■2-position ON/OFF control (disable tracking for primary loop)

Function block symbol	Processing details
M+P_ONF2	Performs two-position (on/off) control.

■3-position ON/OFF control (enable tracking for primary loop)

Function block symbol	Processing details
M+P_ONF3_T	Performs three-position (on/off) control. The primary loop can be tracked.

■3-position ON/OFF control (disable tracking for primary loop)

Function block symbol	Processing details
M+P_ONF3	Performs three-position (on/off) control.

■Program setter

Function block symbol	Processing details
M+P_PGS	Registers time width sets and setting value programs of up to 16 steps, and outputs a setting value corresponding to the progress time of each step with linear interpolation.

■Multi-point program setter

Function block symbol	Processing details
M+P_PGS2_	Registers time width sets and setting value programs of up to 32 steps, and outputs a setting value corresponding to the progress time of each step with linear interpolation.

■Loop selector (disable tracking for primary loop)

Function block symbol	Processing details
M+P_SEL	Selects an input value depending on the setting of the select signal and outputs the result. The input 1 and 2 are not tracked.

■Loop selector (enable tracking for primary loop only by CASIN_T)

Function block symbol	Processing details
M+P_SEL_T1	Selects an input value depending on the setting of the select signal and outputs the result. The input 1 is not tracked. The primary loop of the input 2 is tracked, but tracking from the secondary loop is not performed.

■Loop selector (enable tracking for primary loop by CASIN_T1/_T2)

Function block symbol	Processing details
M+P_SEL_T2	Selects an input value depending on the setting of the select signal and outputs the result. The primary loop of the input 1 and 2 is tracked, but tracking from the secondary loop is not performed.

■Loop selector (enable tracking from secondary loop to primary loop)

Function block symbol	Processing details
M+P_SEL_T3_	Selects two input values depending on the setting of the select signal and outputs the result. The primary loop of the input 1 and 2 is tracked from the secondary loop.

Special

The following function block changes the control mode.

■Change control mode

Function block symbol	Processing details
M+P_MCHG	Switches the control mode to one of MANUAL, AUTO, CASCADE, COMPUTER MV, COMPUTER SV, and CASCADE DIRECT according to a mode change signal.

A

Tag FBs

Loop tag

The following function blocks perform loop control processing including ratio control, various PID controls, two-position (on/off) control, three-position (on/off) control, program setter, and loop selector.

■Velocity type PID control (enable tracking for primary loop)

Function block symbol	Processing details
M+M_PID_T	Performs velocity type PID control combining the functions of M+P_IN, M+P_PHPL, M+P_PID_T, and M+P_OUT1 into a single FB. The primary loop can be tracked.

■Velocity type PID control (disable tracking for primary loop)

Function block symbol	Processing details
M+M_PID	Performs velocity type PID control combining the functions of M+P_IN, M+P_PHPL, M+P_PID, and M+P_OUT1 into a single FB.

■Velocity type PID control and DUTY output (enable tracking for primary loop)

Function block symbol	Processing details
M+M_PID_DUTY_T	Performs velocity type PID control and duty output combining the functions of M+P_IN, M+P_PHPL, M+P_PID_T, and M+P_DUTY into a single FB. The primary loop can be tracked.

■Velocity type PID control and DUTY output (disable tracking for primary loop)

Function block symbol	Processing details
M+M_PID_DUTY	Performs velocity type PID control and duty output combining the functions of M+P_IN, M+P_PHPL, M+P_PID, and M+P_DUTY into a single FB.

■2-degree-of-freedom PID control (enable tracking for primary loop)

Function block symbol	Processing details
M+M_2PID_T	Performs two-degree-of-freedom PID control combining the functions of M+P_IN, M+P_PHPL, M+P_2PID_T, and M+P_OUT1 into a single FB. The primary loop can be tracked.

■2-degree-of-freedom PID control (disable tracking for primary loop)

Function block symbol	Processing details
M+M_2PID	Performs two-degree-of-freedom PID control combining the functions of M+P_IN, M+P_PHPL, M+P_2PID, and M+P_OUT1 into a single FB.

■2-degree-of-freedom PID control and DUTY O/P (enable tracking for primary loop)

Function block symbol	Processing details
M+M_2PID_DUTY_T	Performs two-degree-of-freedom PID control and duty output combining the functions of M+P_IN, M+P_PHPL, M+P_2PID_T, and M+P_DUTY into a single FB. The primary loop can be tracked.

■2-degree-of-freedom PID control and DUTY O/P (disable tracking for primary loop)

Function block symbol	Processing details
M+M_2PID_DUTY	Performs two-degree-of-freedom PID control and duty output combining the functions of M+P_IN, M+P_PHPL, M+P_2PID, and M+P_DUTY into a single FB.

■2-degree-of-freedom advanced PID control (enable tracking for primary loop)

Function block symbol	Processing details
M+M_2PIDH_T_	Performs two-degree-of-freedom PID control combining the functions of M+P_IN, M+P_PHPL, M+P_2PIDH_T_, and M+P_OUT3_ into a single FB with PV/MV correction. The primary loop can be tracked.

■2-degree-of-freedom advanced PID control (disable tracking for primary loop)

Function block symbol	Processing details
M+M_2PIDH_	Performs two-degree-of-freedom PID control combining the functions of M+P_IN, M+P_PHPL, M+P_2PIDH_, and M+P_OUT3_ into a single FB with PV/MV correction.

■Position type PID control (enable tracking for primary loop/disable tracking from secondary loop)

Function block symbol	Processing details
M+M_PIDP_T	Performs position type PID control combining the functions of M+P_IN, M+P_PHPL, and M+P_PIDP_T into a single FB. The primary loop can be tracked.

■Position type PID control (disable tracking for primary loop/disable tracking from secondary loop)

Function block symbol	Processing details
M+M_PIDP	Performs position type PID control combining the functions of M+P_IN, M+P_PHPL, and M+P_PIDP into a single FB.

■Position type PID control (enable tracking for primary loop/enable tracking from secondary loop)

Function block symbol	Processing details
M+M_PIDP_EX_T_	Performs position type PID control combining the functions of M+P_IN, M+P_PHPL, and M+P_PIDP_EX_T_ into a single FB. It also allows manipulated value bumpless switching and tracking from the primary loop and secondary loop at a change of control mode.

■Position type PID control (disable tracking for primary loop/enable tracking from secondary loop)

Function block symbol	Processing details
M+M_PIDP_EX_	Performs position type PID control combining the functions of M+P_IN, M+P_PHPL, and M+P_PIDP_EX_ into a single FB. It also allows manipulated value bumpless switching and tracking from the secondary loop at a change of control mode.

■Sample PI control (enable tracking for primary loop)

Function block symbol	Processing details
M+M_SPI_T	Performs sample PI control combining the functions of M+P_IN, M+P_PHPL, M+P_SPI_T, and M+P_OUT1 into a single FB. The primary loop can be tracked.

■Sample PI control (disable tracking for primary loop)

Function block symbol	Processing details
M+M_SPI	Performs sample PI control combining the functions of M+P_IN, M+P_PHPL, M+P_SPI, and M+P_OUT1 into a single FB.

■I-PD control (enable tracking for primary loop)

Function block symbol	Processing details
M+M_IPD_T	Performs I-PD control combining the functions of M+P_IN, M+P_PHPL, M+P_IPD_T, and M+P_OUT1 into a single FB. The primary loop can be tracked.

■I-PD control (disable tracking for primary loop)

Function block symbol	Processing details
M+M_IPD	Performs I-PD control combining the functions of M+P_IN, M+P_PHPL, M+P_IPD, and M+P_OUT1 into a single FB.

■Blend PI control (enable tracking for primary loop)

Function block symbol	Processing details
M+M_BPI_T	Performs blend PI control combining the functions of M+P_IN, M+P_PHPL, M+P_BPI_T, and M+P_OUT1 into a single FB. The primary loop can be tracked.

■Blend PI control (disable tracking for primary loop)

Function block symbol	Processing details
M+M_BPI	Performs blend PI control combining the functions of M+P_IN, M+P_PHPL, M+P_BPI, and M+P_OUT1 into a single FB.

■Ratio control (enable tracking for primary loop)

Function block symbol	Processing details
M+M_R_T	Performs ratio control combining the functions of M+P_IN, M+P_PHPL, M+P_R_T, and M+P_OUT2 into a single FB. The primary loop can be tracked.

■Ratio control (disable tracking for primary loop)

Function block symbol	Processing details
M+M_R	Performs ratio control combining the functions of M+P_IN, M+P_PHPL, M+P_R, and M+P_OUT2 into a single FB.

■2-position ON/OFF control (enable tracking for primary loop)

Function block symbol	Processing details
M+M_ONF2_T	Performs two-position (on/off) control combining the functions of M+P_IN, M+P_PHPL, and M+P_ONF2_T into a single FB. The primary loop can be tracked.

■2-position ON/OFF control (disable tracking for primary loop)

Function block symbol	Processing details
M+M_ONF2	Performs two-position (on/off) control combining the functions of M+P_IN, M+P_PHPL, and M+P_ONF2 into a single FB.

■3-position ON/OFF control (enable tracking for primary loop)

Function block symbol	Processing details
M+M_ONF3_T	Performs three-position (on/off) control combining the functions of M+P_IN, M+P_PHPL, and M+P_ONF3_T into a single FB. The primary loop can be tracked.

■3-position ON/OFF control (disable tracking for primary loop)

Function block symbol	Processing details
M+M_ONF3	Performs three-position (on/off) control combining the functions of M+P_IN, M+P_PHPL, and M+P_ONF3 into a single FB.

■Monitor

Function block symbol	Processing details
M+M_MONI	Performs monitoring combining the functions of M+P_IN and M+P_PHPL into a single FB.

■Manual output with monitor

Function block symbol	Processing details
M+M_MWM	Performs manual output with monitor combining the functions of M+P_IN, M+P_PHPL, and M+P_MOUT into a single FB.

■Batch preparation

Function block symbol	Processing details
M+M_BC	Performs batch preparation combining the functions of M+P_PSUM and M+P_BC into a single FB.

■Pulse integrator

Function block symbol	Processing details
M+M_PSUM	Performs pulse integration combining the functions of M+P_PSUM into a single FB.

■Loop selector (disable tracking for primary loop)

Function block symbol	Processing details
M+M_SEL	Performs loop selector combining the functions of M+P_SEL into a single FB.

■Loop selector (enable tracking for primary loop only by CASIN_T)

Function block symbol	Processing details
M+M_SEL_T1	Performs loop selector combining the functions of M+P_SEL_T1 into a single FB. The input 1 is not tracked. The primary loop of the input 2 is tracked, but tracking from the secondary loop is not performed.

■Loop selector (enable tracking for primary loop by CASIN_T1/_T2)

Function block symbol	Processing details
M+M_SEL_T2	Performs loop selector combining the functions of M+P_SEL_T2 into a single FB. The primary loop of the input 1 and 2 is tracked, but tracking from the secondary loop is not performed.

■Loop selector (enable tracking from secondary loop to primary loop)

Function block symbol	Processing details
M+M_SEL_T3_	Performs loop selector combining the functions of M+P_SEL_T3_ into a single FB. The primary loop of the input 1 and 2 can be tracked from the secondary loop.

■Manual output

Function block symbol	Processing details
M+P_MOUT	Performs manual output combining the functions of M+P_MOUT into a single FB.

■Program setter

Function block symbol	Processing details
M+M_PGS	Performs program settings combining the functions of M+P_PGS into a single FB.

■Multi-point program setter

Function block symbol	Processing details
M+M_PGS2_	Performs program settings combining the functions of M+P_PGS2_ into a single FB. Multiple FBs can be connected to execute time width sets and setting value programs of 32 steps or more.

■Manual setter with monitor

Function block symbol	Processing details
M+M_SWM_	Performs manual settings with monitor combining the functions of M+P_IN, M+P_PHPL, and M+P_MSET_ into a single FB.

■Position proportional output

Function block symbol	Processing details
M+M_PVAL_T_	Outputs open/close command bits to operate the motor valve opening according to the deviation of opening between the feedback input and setting value. The primary loop can be tracked.

■Heating and cooling output

Function block symbol	Processing details
M+M_HTCL_T_	Outputs two manipulated values after split conversion and output conversion to setting values. This function block can perform temperature control by outputting the values to both the heating and cooling operation terminal. The primary loop can be tracked.

A

Status tag

The following function blocks perform reversible/irreversible operation and ON/OFF operation and operates as a timer and a counter.

■Motor irreversible (2 input/2 output)

Function block symbol	Processing details
M+M_NREV	Performs irreversible operation and controls a solenoid valve.

■Motor reversible (2 input/3 output)

Function block symbol	Processing details
M+M_REV	Performs reversible operation.

■ON/OFF operation (2 input/2 output)

Function block symbol	Processing details
M+M_MVAL1	Controls an ON/OFF motor valve and solenoid valve.

■ON/OFF operation (2 input/3 output)

Function block symbol	Processing details
M+M_MVAL2	Controls an ON/OFF motor valve (with intermediate status).

■Timer 1 (timer stops when COMPLETE flag is on)

Function block symbol	Processing details
M+M_TIMER1	This function block is a clock timer. It stops counting the time when the timer count value reaches the set value.

■Timer 2 (timer continues when COMPLETE flag is on)

Function block symbol	Processing details
M+M_TIMER2	This function block is a clock timer. It continues to count the time even if the time count value exceeds the set value, and the timer stops when the time count value reaches the timer high limit value.

■Counter 1 (counter stops when COMPLETE flag is on)

Function block symbol	Processing details
M+M_COUNTER1	This function block is a counter that counts contact signal inputs. It stops counting the inputs when the count value reaches the set value.

■Counter 2 (counter continues when COMPLETE flag is ON)

Function block symbol	Processing details
M+M_COUNTER2	This function block is a counter that counts contact signal inputs. It continues to count the inputs even if the count value exceeds the set value, and stops counting the input when the count value reaches the counter high limit value.

■Push button operation (5 input/5 output)

Function block symbol	Processing details
M+M_PB_	Performs push button operation.

Alarm tag

The following function blocks perform alarm notification.

■Alarm

Function block symbol	Processing details
M+M_ALARM	Displays the alarms for the input pins ALMIN1 to ALMIN8 to which TRUE is input on the "Alarm List" window of the PX Developer monitor tool.

■64 points alarm

Function block symbol	Processing details
M+M_ALARM_64PT_	Displays the alarms for the bits of the input pins ALMIN_W1 to ALMIN_W4 to which TRUE is input on the "Alarm List" window of the PX Developer monitor tool.

Message tag

The following function blocks perform message notification.

■Message

Function block symbol	Processing details
M+M_MESSAGE	Displays the events for the input pins MSGIN1 to MSGIN8 to which TRUE is input on the "Event List" window of the PX Developer monitor tool.

■64 points message

Function block symbol	Processing details
M+M_MESSAGE_64PT_	Displays the events for the bits of the input pins MSGIN_W1 to MSGIN_W4 to which TRUE is input on the "Event List" window of the PX Developer monitor tool.

A

Appendix 2 Tag Data List

How to read items in the list is shown below.

Item	Description
Offset	Indicates the offset word of memory data inside the tags.
Label	Indicates tag data (tag member).
Data type	Indicates the memory data structure. • BOOL: Bit • WORD: Integer data (word [unsigned]/bit string [16 bits]) • INT: Integer data (word [signed]) • DINT: Integer data (double word [signed]) • REAL: Real number data (single-precision real number)
Setting/Storage range (Low limit/ High limit)	Indicates the setting range of all items. Refer to following range for PH, PL, HH, LL setting/storage range. • PV high limit alarm value (PH_): (RL) to (RH) and (PL) < (PH_) • PV low limit alarm value (PL): (RL) to (RH) and (PL) < (PH_) • PV high high limit alarm value (HH): (RL) to (RH) and (PH_) ≤ (HH) • PV low low limit alarm value (LL): (RL) to (RH) and (LL) ≤ (PL) Set the control cycle (CT) to the integral multiple of an execution cycle. The execution cycle indicates an execution cycle set by GX Works3 and an execution cycle of a program block property. 👆 [Tool] ⇨ [Options] ⇨ [Convert] ⇨ [Process Control Extension Setting]
Initial value	Indicates default value.
Number of digits after the decimal point	Indicates the number of digits after the decimal point. N is indicated by the number of digits after the decimal point of +9[N_].
Unit	Indicates units. UNIT is set in the "Unit Setting" window of PX Developer. 👆 [Monitor Tool Setting] ⇨ [Unit Setting]
Application	Indicates the name/application of tag data (tag member).
Storage	Indicates whether it is allowed to read/write tag data using programs. ■User It is allowed to read/write tag data. However, tag data with (condition 1/2/3) can only be written only under the following conditions. • Condition 1: When changing the control mode using a program, use M+P_MCHG of the tag access FB. Transition to the control mode with TRUE as the corresponding bit item of Disable Mode Change is not allowed. • Condition 2: Stop alarm (SPA) and output open (OOA) of an alarm are written by a program. When the program sets stop alarm (SPA) as TRUE, stop alarm in the corresponding loop is processed. The bit items except stop alarm (SPA) and output open (OOA) of an alarm are written by the system. Do not write them using a program. • Condition 3: It is allowed to write tag data only when the control mode is MAN or CMV. ■System It is allowed only to read tag data. Do not write tag data. The operations are not guaranteed if it is written. This item is not displayed on the "FB Property" window on GX Works3. ■Tag data access control Tag data can be written only from the ActiveX container application that uses the tag data access control that is an ActiveX control. For details on tag data access control, refer to the following. 📖 PX Developer Version 1 Operating Manual (Monitor Tool)
Tag access FB	Indicates the tag access FB which reads/writes tag data.

The following table describes Boolean items of tag data.

Item	Description
Control mode	TRUE: Valid, FALSE: Invalid
Disable Mode Change	TRUE: Valid, FALSE: Invalid
Alarm	TRUE: Detected, FALSE: Reset
Disable alarm detection	TRUE: Valid, FALSE: Invalid
Alarm level	TRUE: Major alarm, FALSE: Minor alarm

2PIDH

Structure name: M+TM_2PIDH

Offset		Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB
				Low limit	High limit						
+0		FUNC	INT	17	17	17	—	—	Tag Function Code	System	—
+1	b3	MAN	BOOL	FALSE	TRUE	TRUE	—	—	Control Mode: MANUAL	User (condition 1)	M+P_MCHG M+P_2PIDH(_T)_ M+P_OUT3
	b4	AUT	BOOL	FALSE	TRUE	FALSE	—	—	Control Mode: AUTO	User (condition 1)	M+P_MCHG M+P_2PIDH(_T)_ M+P_OUT3
	b5	CAS	BOOL	FALSE	TRUE	FALSE	—	—	Control Mode: CASCADE	User (condition 1)	M+P_MCHG M+P_2PIDH(_T)_ M+P_OUT3
	b9	CMV	BOOL	FALSE	TRUE	FALSE	—	—	Control Mode: COMPUTER MV	User (condition 1)	M+P_MCHG M+P_2PIDH(_T)_ M+P_OUT3
	bA	CSV	BOOL	FALSE	TRUE	FALSE	—	—	Control Mode: COMPUTER SV	User (condition 1)	M+P_MCHG M+P_2PIDH(_T)_ M+P_OUT3
	bB	CASDR	BOOL	FALSE	TRUE	FALSE	—	—	Control Mode: CASCADE DIRECT	User (condition 1)	M+P_MCHG M+P_2PIDH(_T)_ M+P_OUT3

A

Offset		Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB
				Low limit	High limit						
+2	b3	MANI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Mode Change: Disable MANUAL	User	M+P_MCHG
	b4	AUTI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Mode Change: Disable AUTO	User	M+P_MCHG
	b5	CASI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Mode Change: Disable CASCADE	User	M+P_MCHG
	b9	CMVI	BOOL	FALSE	TRUE	TRUE	—	—	Disable Mode Change: Disable COMPUTER MV	User	M+P_MCHG
	bA	CSVI	BOOL	FALSE	TRUE	TRUE	—	—	Disable Mode Change: Disable COMPUTER SV	User	M+P_MCHG
	bB	CASDRI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Mode Change: Disable CASCADE DIRECT	User	M+P_MCHG
	bC	TSTPI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Mode Change: Disable TAG STOP	User	M+P_MCHG
	bD	ATI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Mode Change: Disable Auto Tuning	User	M+P_MCHG
	bE	OVRI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Mode Change: Disable OVERRIDE	User	M+P_MCHG
	bF	SIMI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Mode Change: Disable SIMULATION	User	M+P_MCHG

Offset		Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB
				Low limit	High limit						
+3	b0	MLA	BOOL	FALSE	TRUE	FALSE	—	—	Output Low Limit Alarm	User (condition 2)	Common
	b1	MHA	BOOL	FALSE	TRUE	FALSE	—	—	Output High Limit Alarm	User (condition 2)	Common
	b2	DVLA	BOOL	FALSE	TRUE	FALSE	—	—	Large Deviation Alarm	User (condition 2)	Common
	b3	DPNA	BOOL	FALSE	TRUE	FALSE	—	—	Negative Variation Rate Alarm	User (condition 2)	Common
	b4	DPPA	BOOL	FALSE	TRUE	FALSE	—	—	Positive Variation Rate Alarm	User (condition 2)	Common
	b5	PLA	BOOL	FALSE	TRUE	FALSE	—	—	Input Low Limit Alarm	User (condition 2)	Common
	b6	PHA	BOOL	FALSE	TRUE	FALSE	—	—	Input High Limit Alarm	User (condition 2)	Common
	b7	LLA	BOOL	FALSE	TRUE	FALSE	—	—	Input Low Low Limit Alarm	User (condition 2)	Common
	b8	HHA	BOOL	FALSE	TRUE	FALSE	—	—	Input High High Limit Alarm	User (condition 2)	Common
	b9	SEA	BOOL	FALSE	TRUE	FALSE	—	—	Sensor Error Alarm	User (condition 2)	Common
	bA	OOA	BOOL	FALSE	TRUE	FALSE	—	—	Output Open Alarm	User (condition 2)	Common
	bB	DMLA	BOOL	FALSE	TRUE	FALSE	—	—	Output Variation Rate Limit Alarm	User (condition 2)	Common
	bE	SPA	BOOL	FALSE	TRUE	FALSE	—	—	Stop Alarm	User (condition 2)	Common

A

Offset		Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB
				Low limit	High limit						
+4	b0	MLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Output Low Limit Alarm	User	Common
	b1	MHI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Output High Limit Alarm	User	Common
	b2	DVLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Large Deviation Alarm	User	Common
	b3	DPNI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Negative Variation Rate Alarm	User	Common
	b4	DPPI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Positive Variation Rate Alarm	User	Common
	b5	PLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input Low Limit Alarm	User	Common
	b6	PHI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input High Limit Alarm	User	Common
	b7	LLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input Low Low Limit Alarm	User	Common
	b8	HHI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input High High Limit Alarm	User	Common
	b9	SEI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Sensor Error Alarm	User	Common
	bB	DMLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Output Variation Rate Limit Alarm	User	Common
	bD	TRKF	BOOL	FALSE	TRUE	FALSE	—	—	Tracking Flag	User	Common
	bF	ERRI	BOOL	FALSE	TRUE	FALSE	—	—	Disable All Alarms	User	Common

Offset		Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB
				Low limit	High limit						
+5	b0	MLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Output Low Limit Alarm	User	—
	b1	MHL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Output High Limit Alarm	User	—
	b2	DVLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Large Deviation Alarm	User	—
	b3	DPNL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Negative Variation Rate Alarm	User	—
	b4	DPPL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Positive Variation Rate Alarm	User	—
	b5	PLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input Low Limit Alarm	User	—
	b6	PHL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input High Limit Alarm	User	—
	b7	LLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input Low Low Limit Alarm	User	—
	b8	HHL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input High High Limit Alarm	User	—
	b9	SENL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Sensor Error Alarm	User	—
	bA	OOL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Output Open Alarm	User	—
	bB	DMLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Output Variation Rate Limit Alarm	User	—
	bE	SPL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Stop Alarm	User	—
+6		CTNO	INT	0	32	0	—	—	Lockout Tag No.	System	—
+7		CTFN	WORD	0	H0002	H0000	—	—	Lockout Tag Function	System	—
+8		UNIT	INT	0	127	0	—	—	Unit	User	—
+9		N_	INT	0	4	1	—	—	Number of digits after the decimal point	User	—
+10 +11		PV	REAL	RL	RH	0.0	N	UNIT	Process Variable	System	M+P_PHPL
+12 +13		MV	REAL	-10	110	0.0	1	%	Manipulated Variable	User (condition 3)	M+P_OUT3 M+P_2PIDH(_T)_
+14 +15		SVC	REAL	RL	RH	0.0	N	UNIT	Setting Value (Current)	System	M+P_2PIDH(_T)_
+16 +17		DV	REAL	-110	110	0.0	1	%	Deviation	System	M+P_2PIDH(_T)_

Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB
			Low limit	High limit						
+18 +19	MH	REAL	-10	110	100.0	1	%	MV High Limit Value	User	M+P_OUT3 M+P_2PIDH(_T)_
+20 +21	ML	REAL	-10	110	0.0	1	%	MV Low Limit Value	User	M+P_OUT3 M+P_2PIDH(_T)_
+22 +23	RH	REAL	-999999	999999	100.0	N	UNIT	PV Engineering Value High Limit	User	M+P_PHPL
+24 +25	RL	REAL	-999999	999999	0.0	N	UNIT	PV Engineering Value Low Limit	User	M+P_PHPL
+26 +27	PH_	REAL	RL	RH	100.0	N	UNIT	PV High Limit Alarm Value	User	M+P_PHPL
+28 +29	PL	REAL	RL	RH	0.0	N	UNIT	PV Low Limit Alarm Value	User	M+P_PHPL
+30 +31	HH	REAL	RL	RH	100.0	N	UNIT	PV High High Limit Alarm Value	User	M+P_PHPL
+32 +33	LL	REAL	RL	RH	0.0	N	UNIT	PV Low Low Limit Alarm Value	User	M+P_PHPL
+34 +35	SH	REAL	RL	RH	100.0	N	UNIT	SV High Limit Value	User	M+P_2PIDH(_T)_
+36 +37	SL	REAL	RL	RH	0.0	N	UNIT	SV Low Limit Value	User	M+P_2PIDH(_T)_
+38 +39	ALPHA	REAL	0	1	0.2	2	—	PV Filter Coefficient	User	M+P_IN
+40 +41	HS	REAL	0	100	0.0	1	%	PV High/Low Limit Alarm Hysteresis	User	M+P_PHPL
+42 +43	CTIM	REAL	0	9999	0.0	2	s	Variation Rate Alarm Check Time	User	M+P_PHPL
+44 +45	DPL	REAL	0	100	100.0	1	%	Variation Rate Alarm Value	User	M+P_PHPL
+46 +47	CT	REAL	0	9999	1.0	2	s	Control Cycle	User	M+P_2PIDH(_T)_
+48 +49	DML	REAL	0	100	100.0	1	%	Output Variation Rate High Limit Value	User	M+P_OUT3
+50 +51	DVL	REAL	0	100	100.0	1	%	Deviation Limit Value	User	M+P_2PIDH(_T)_
+52 +53	PROPORTIONAL	REAL	0	999	1.0	2	—	Gain	User	M+P_2PIDH(_T)_
+54 +55	INTEGRAL	REAL	0	9999	10.0	1	s	Integral Time	User	M+P_2PIDH(_T)_
+56 +57	DERIVATIVE	REAL	0	9999	0.0	1	s	Derivative Time	User	M+P_2PIDH(_T)_
+58 +59	GW	REAL	0	100	0.0	1	—	Gap Width	User	M+P_2PIDH(_T)_
+60 +61	GG	REAL	0	99	1.0	1	—	Gap Gain	User	M+P_2PIDH(_T)_

Offset		Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB
				Low limit	High limit						
+62 +63		MVP	REAL	-999999	999999	0.0	1	—	MV Internal Operation Value	System	—
+64 +65		ALPHA2	REAL	0	1	0.0	2	—	2-degree-of-freedom Parameter Alpha	User	M+P_2PIDH(_T)_
+66 +67		BETA2	REAL	0	1	1.0	2	—	2-degree-of-freedom Parameter Beta	User	M+P_2PIDH(_T)_
+68 +69		CTDUTY	REAL	0	9999	1.0	2	s	Control Output Cycle	User	Reserved for future use
+70 +71		AT1STEPMV	REAL	-100	100	0.0	1	%	Step Manipulated Variable for AT1	User	M+P_2PIDH(_T)_
+72 +73		AT1ST	REAL	0	9999	1.0	2	s	Sampling Interval Time for AT1	User	M+P_2PIDH(_T)_
+74 +75		AT1TOUT1	REAL	0	9999	100.0	1	s	Time-out Interval for AT1/AT2	User	M+P_2PIDH(_T)_
+76 +77		AT1TOUT2	REAL	0	9999	10.0	1	s	Time-out Interval after Maximum Slope for AT1	User	M+P_2PIDH(_T)_
+78 +79		AT2HS	REAL	0	10	1.0	1	%	Hysteresis for AT2	User	M+P_2PIDH(_T)_
+80 +81		AT2MVH	REAL	0	100	100.0	1	%	Output High Limit Value for AT2	User	M+P_2PIDH(_T)_
+82 +83		AT2MVL	REAL	0	100	0.0	1	%	Output Low Limit Value for AT2	User	M+P_2PIDH(_T)_
+86		ATTYPE	INT	0	4	1	—	—	Control Type for AT	User	M+P_2PIDH(_T)_
+87	bD	SVLA	BOOL	FALSE	TRUE	FALSE	—	—	SV Low Alarm	System	Common
	bE	SVHA	BOOL	FALSE	TRUE	FALSE	—	—	SV High Alarm	System	Common
	bF	DSVLA	BOOL	FALSE	TRUE	FALSE	—	—	SV Variation Rate Limit Alarm	System	Common
+88	bD	SVLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable SV Low Alarm	User	Common
	bE	SVHI	BOOL	FALSE	TRUE	FALSE	—	—	Disable SV High Alarm	User	Common
	bF	DSVLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable SV Variation Rate Limit Alarm	User	Common
+89	bD	SVLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of SV Low Alarm	User	Common
	bE	SVHL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of SV High Alarm	User	Common
	bF	DSVLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of SV Variation Rate Limit Alarm	User	Common
+90 +91		SV	REAL	RL	RH	0.0	N	UNIT	Setting Value (Target)	User	—

A

Offset		Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB
				Low limit	High limit						
+92 +93		DSVL	REAL	0	100	100.0	1	%	SV Variation Rate High Limit Value	User	M+P_2PIDH(_T)_
+94	b0	DOM_AT_START_SET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Start Setting	Tag data access control	—
	b1	DOM_AT_STOP_SET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Stop Setting	Tag data access control	—
	b2	DOM_AT_TYPE	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Command Type	Tag data access control	—
	bD	TSTP	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: TAG STOP	Tag data access control	—
	bE	OVR	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: OVERRIDE	Tag data access control	—
	bF	SIM	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: SIMULATION	Tag data access control	—
+95	b0	DIM_AT_RUN	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Tuning	System	—
	b2	DIM_AT_PH	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Input High Limit	System	—
	b3	DIM_AT_PL	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Input Low Limit	System	—
	b4	DIM_AT_MVH	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Output High Limit	System	—
	b5	DIM_AT_MVL	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Output Low Limit	System	—
	b6	DIM_AT_TO	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Time-out	System	—
	b7	DIM_AT_MODE	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Operation Mode	System	—
	b8	DIM_AT_ID	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Identification	System	—
	b9	DIM_PREMV	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Preset MV	System	—
	bA	DIM_MVHLD	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: MV Hold	System	—
	bB	DIM_MVTRK	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: MV Tracking	System	—

PGS2

Structure name: M+TM_PGS2

Offset		Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB
				Low limit	High limit						
+0		FUNC	INT	18	18	18	—	—	Tag Function Code	System	—
+1	b3	MAN	BOOL	FALSE	TRUE	TRUE	—	—	Control Mode: MANUAL	User (condition 1)	M+P_MCHG M+P_PGS2_
	b4	AUT	BOOL	FALSE	TRUE	FALSE	—	—	Control Mode: AUTO	User (condition 1)	M+P_MCHG M+P_PGS2_
+2	b3	MANI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Mode Change: Disable MANUAL	User	M+P_MCHG
	b4	AUTI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Mode Change: Disable AUTO	User	M+P_MCHG
	bC	TSTPI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Mode Change: Disable TAG STOP	User	M+P_MCHG
+3	b0	SVLA	BOOL	FALSE	TRUE	FALSE	—	—	SV Low Alarm	User (condition 2)	Common
	b1	SVHA	BOOL	FALSE	TRUE	FALSE	—	—	SV High Alarm	User (condition 2)	Common
	bE	SPA	BOOL	FALSE	TRUE	FALSE	—	—	Stop Alarm	User (condition 2)	Common
+4	b0	SVLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable SV Low Alarm	User	Common
	b1	SVHI	BOOL	FALSE	TRUE	FALSE	—	—	Disable SV High Alarm	User	Common
	bF	ERRI	BOOL	FALSE	TRUE	FALSE	—	—	Disable All Alarms	User	Common
+5	b0	SVLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of SV Low Alarm	User	—
	b1	SVHL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of SV High Alarm	User	—
	bE	SPL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Stop Alarm	User	—
+6		CTNO	INT	0	32	0	—	—	Lockout Tag No.	System	—
+7		CTFN	WORD	0	H0002	H0000	—	—	Lockout Tag Function	System	—
+8		UNIT	INT	0	127	0	—	—	Unit	User	—
+9		N_	INT	0	4	1	—	—	Number of digits after the decimal point	User	—
+10		STNO	INT	0	32	0	—	—	Number of Steps Setting	User	M+P_PGS2_
+11		PVSTART	INT	0	2	0	—	—	PV Start Type	User	M+P_PGS2_
+12 +13		SV	REAL	RL	RH	0.0	N	UNIT	Setting value	User (condition 3)	M+P_PGS2_
+14		STC	INT	0	32	0	—	—	Executing Step No.	User	M+P_PGS2_
+15		T_	INT	0	32767	0	—	—	s (min)[*2] Time in the step	User	M+P_PGS2_

A

Offset		Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB
				Low limit	High limit						
+16		PV	REAL	RL	RH	0.0	N	UNIT	Process Variable	System	M+P_PGS2_
+17											
+18		SH	INT	-32768	32767	100	—	UNIT[*1]	SV High Limit Value	User	M+P_PGS2_
+19		SL	INT	-32768	32767	0	—	UNIT[*1]	SV Low Limit Value	User	M+P_PGS2_
+20	b0	TYP_HOLD	BOOL	FALSE	TRUE	TRUE	—	—	Operation Type: HOLD	User	M+P_PGS2_
	b1	TYP_RETURN	BOOL	FALSE	TRUE	FALSE	—	—	Operation Type: RETURN	User	M+P_PGS2_
	b2	TYP_CYCLIC	BOOL	FALSE	TRUE	FALSE	—	—	Operation Type: CYCLIC	User	M+P_PGS2_
	b8	TUNIT	BOOL	FALSE	TRUE	FALSE	—	—	Unit of time	User	M+P_PGS2_
+21		WAIT	INT	0	32767	0	—	UNIT[*1]	Wait Width	User	M+P_PGS2_
+22		RH	INT	-32768	32767	100	—	UNIT[*1]	Engineering Value High Limit	User	M+P_PGS2_
+23		RL	INT	-32768	32767	0	—	UNIT[*1]	Engineering Value Low Limit	User	M+P_PGS2_
+26		SV0	INT	-32768	32767	0	—	UNIT[*1]	Start Point	User	M+P_PGS2_
+27		SV0C	INT	-32768	32767	0	—	UNIT[*1]	Start Point (Current)	System	M+P_PGS2_
+28		T1_	INT	0	32767	0	—	s (min)[*2]	Time Span of Step 1	User	M+P_PGS2_
+29		SV1	INT	-32768	32767	0	—	UNIT[*1]	Setting Value of Step 1	User	M+P_PGS2_
+30		T2_	INT	0	32767	0	—	s (min)[*2]	Time Span of Step 2	User	M+P_PGS2_
+31		SV2	INT	-32768	32767	0	—	UNIT[*1]	Setting Value of Step 2	User	M+P_PGS2_
+32		T3_	INT	0	32767	0	—	s (min)[*2]	Time Span of Step 3	User	M+P_PGS2_
+33		SV3	INT	-32768	32767	0	—	UNIT[*1]	Setting Value of Step 3	User	M+P_PGS2_
+34		T4_	INT	0	32767	0	—	s (min)[*2]	Time Span of Step 4	User	M+P_PGS2_
+35		SV4	INT	-32768	32767	0	—	UNIT[*1]	Setting Value of Step 4	User	M+P_PGS2_
+36		T5_	INT	0	32767	0	—	s (min)[*2]	Time Span of Step 5	User	M+P_PGS2_
+37		SV5	INT	-32768	32767	0	—	UNIT[*1]	Setting Value of Step 5	User	M+P_PGS2_
+38		T6_	INT	0	32767	0	—	s (min)[*2]	Time Span of Step 6	User	M+P_PGS2_
+39		SV6	INT	-32768	32767	0	—	UNIT[*1]	Setting Value of Step 6	User	M+P_PGS2_
+40		T7_	INT	0	32767	0	—	s (min)[*2]	Time Span of Step 7	User	M+P_PGS2_
+41		SV7	INT	-32768	32767	0	—	UNIT[*1]	Setting Value of Step 7	User	M+P_PGS2_
+42		T8_	INT	0	32767	0	—	s (min)[*2]	Time Span of Step 8	User	M+P_PGS2_
+43		SV8	INT	-32768	32767	0	—	UNIT[*1]	Setting Value of Step 8	User	M+P_PGS2_
+44		T9_	INT	0	32767	0	—	s (min)[*2]	Time Span of Step 9	User	M+P_PGS2_

Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB
			Low limit	High limit						
+45	SV9	INT	-32768	32767	0	—	UNIT[*1]	Setting Value of Step 9	User	M+P_PGS2_
+46	T10_	INT	0	32767	0	—	s (min)[*2]	Time Span of Step 10	User	M+P_PGS2_
+47	SV10	INT	-32768	32767	0	—	UNIT[*1]	Setting Value of Step 10	User	M+P_PGS2_
+48	T11_	INT	0	32767	0	—	s (min)[*2]	Time Span of Step 11	User	M+P_PGS2_
+49	SV11	INT	-32768	32767	0	—	UNIT[*1]	Setting Value of Step 11	User	M+P_PGS2_
+50	T12_	INT	0	32767	0	—	s (min)[*2]	Time Span of Step 12	User	M+P_PGS2_
+51	SV12	INT	-32768	32767	0	—	UNIT[*1]	Setting Value of Step 12	User	M+P_PGS2_
+52	T13_	INT	0	32767	0	—	s (min)[*2]	Time Span of Step 13	User	M+P_PGS2_
+53	SV13	INT	-32768	32767	0	—	UNIT[*1]	Setting Value of Step 13	User	M+P_PGS2_
+54	T14_	INT	0	32767	0	—	s (min)[*2]	Time Span of Step 14	User	M+P_PGS2_
+55	SV14	INT	-32768	32767	0	—	UNIT[*1]	Setting Value of Step 14	User	M+P_PGS2_
+56	T15_	INT	0	32767	0	—	s (min)[*2]	Time Span of Step 15	User	M+P_PGS2_
+57	SV15	INT	-32768	32767	0	—	UNIT[*1]	Setting Value of Step 15	User	M+P_PGS2_
+58	T16_	INT	0	32767	0	—	s (min)[*2]	Time Span of Step 16	User	M+P_PGS2_
+59	SV16	INT	-32768	32767	0	—	UNIT[*1]	Setting Value of Step 16	User	M+P_PGS2_
+60	T17_	INT	0	32767	0	—	s (min)[*2]	Time Span of Step 17	User	M+P_PGS2_
+61	SV17	INT	-32768	32767	0	—	UNIT[*1]	Setting Value of Step 17	User	M+P_PGS2_
+62	T18_	INT	0	32767	0	—	s (min)[*2]	Time Span of Step 18	User	M+P_PGS2_
+63	SV18	INT	-32768	32767	0	—	UNIT[*1]	Setting Value of Step 18	User	M+P_PGS2_
+64	T19_	INT	0	32767	0	—	s (min)[*2]	Time Span of Step 19	User	M+P_PGS2_
+65	SV19	INT	-32768	32767	0	—	UNIT[*1]	Setting Value of Step 19	User	M+P_PGS2_
+66	T20_	INT	0	32767	0	—	s (min)[*2]	Time Span of Step 20	User	M+P_PGS2_
+67	SV20	INT	-32768	32767	0	—	UNIT[*1]	Setting Value of Step 20	User	M+P_PGS2_
+68	T21_	INT	0	32767	0	—	s (min)[*2]	Time Span of Step 21	User	M+P_PGS2_
+69	SV21	INT	-32768	32767	0	—	UNIT[*1]	Setting Value of Step 21	User	M+P_PGS2_
+70	T22_	INT	0	32767	0	—	s (min)[*2]	Time Span of Step 22	User	M+P_PGS2_
+71	SV22	INT	-32768	32767	0	—	UNIT[*1]	Setting Value of Step 22	User	M+P_PGS2_
+72	T23_	INT	0	32767	0	—	s (min)[*2]	Time Span of Step 23	User	M+P_PGS2_

A

Offset		Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB
				Low limit	High limit						
+73		SV23	INT	-32768	32767	0	—	UNIT[*1]	Setting Value of Step 23	User	M+P_PGS2_
+74		T24_	INT	0	32767	0	—	s (min)[*2]	Time Span of Step 24	User	M+P_PGS2_
+75		SV24	INT	-32768	32767	0	—	UNIT[*1]	Setting Value of Step 24	User	M+P_PGS2_
+76		T25_	INT	0	32767	0	—	s (min)[*2]	Time Span of Step 25	User	M+P_PGS2_
+77		SV25	INT	-32768	32767	0	—	UNIT[*1]	Setting Value of Step 25	User	M+P_PGS2_
+78		T26_	INT	0	32767	0	—	s (min)[*2]	Time Span of Step 26	User	M+P_PGS2_
+79		SV26	INT	-32768	32767	0	—	UNIT[*1]	Setting Value of Step 26	User	M+P_PGS2_
+80		T27_	INT	0	32767	0	—	s (min)[*2]	Time Span of Step 27	User	M+P_PGS2_
+81		SV27	INT	-32768	32767	0	—	UNIT[*1]	Setting Value of Step 27	User	M+P_PGS2_
+82		T28_	INT	0	32767	0	—	s (min)[*2]	Time Span of Step 28	User	M+P_PGS2_
+83		SV28	INT	-32768	32767	0	—	UNIT[*1]	Setting Value of Step 28	User	M+P_PGS2_
+84		T29_	INT	0	32767	0	—	s (min)[*2]	Time Span of Step 29	User	M+P_PGS2_
+85		SV29	INT	-32768	32767	0	—	UNIT[*1]	Setting Value of Step 29	User	M+P_PGS2_
+86		T30_	INT	0	32767	0	—	s (min)[*2]	Time Span of Step 30	User	M+P_PGS2_
+87		SV30	INT	-32768	32767	0	—	UNIT[*1]	Setting Value of Step 30	User	M+P_PGS2_
+88		T31_	INT	0	32767	0	—	s (min)[*2]	Time Span of Step 31	User	M+P_PGS2_
+89		SV31	INT	-32768	32767	0	—	UNIT[*1]	Setting Value of Step 31	User	M+P_PGS2_
+90		T32_	INT	0	32767	0	—	s (min)[*2]	Time Span of Step 32	User	M+P_PGS2_
+91		SV32	INT	-32768	32767	0	—	UNIT[*1]	Setting Value of Step 32	User	M+P_PGS2_
+94	b0	DOM_ADV_START	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Advance Command	Tag data access control	—
	bD	TSTP	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: TAG STOP	Tag data access control	—
+95	b0	DIM_WAIT_MODE	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Waiting	System	—

*1 This tag data is set in integer, ignoring the number of digits after the decimal point (N_).
  When the setting after the decimal point is required depending on the engineering value range, set the tag data as follows.
  (Example) When the setting value is 1.5MPa, convert its unit to 1500KPa to fit the value within the range of -32768 to 32767.
*2 When the unit of time (TUNIT) is FALSE, "s" is used and "min" is used for TRUE.
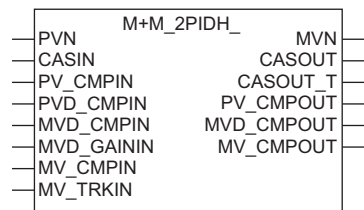
Appendix 3 Function Block Details

This section describes the details on the process control function blocks used in the programs in this manual.

2-degree-of-freedom advanced PID control (disable tracking for primary loop) (M+M_2PIDH_)

M+M_2PIDH_

This function block performs two-degree-of-freedom PID control combining the functions of M+P_IN, M+P_PHPL, M+P_2PIDH_, and M+P_OUT3_ into a single FB with PV/MV correction.
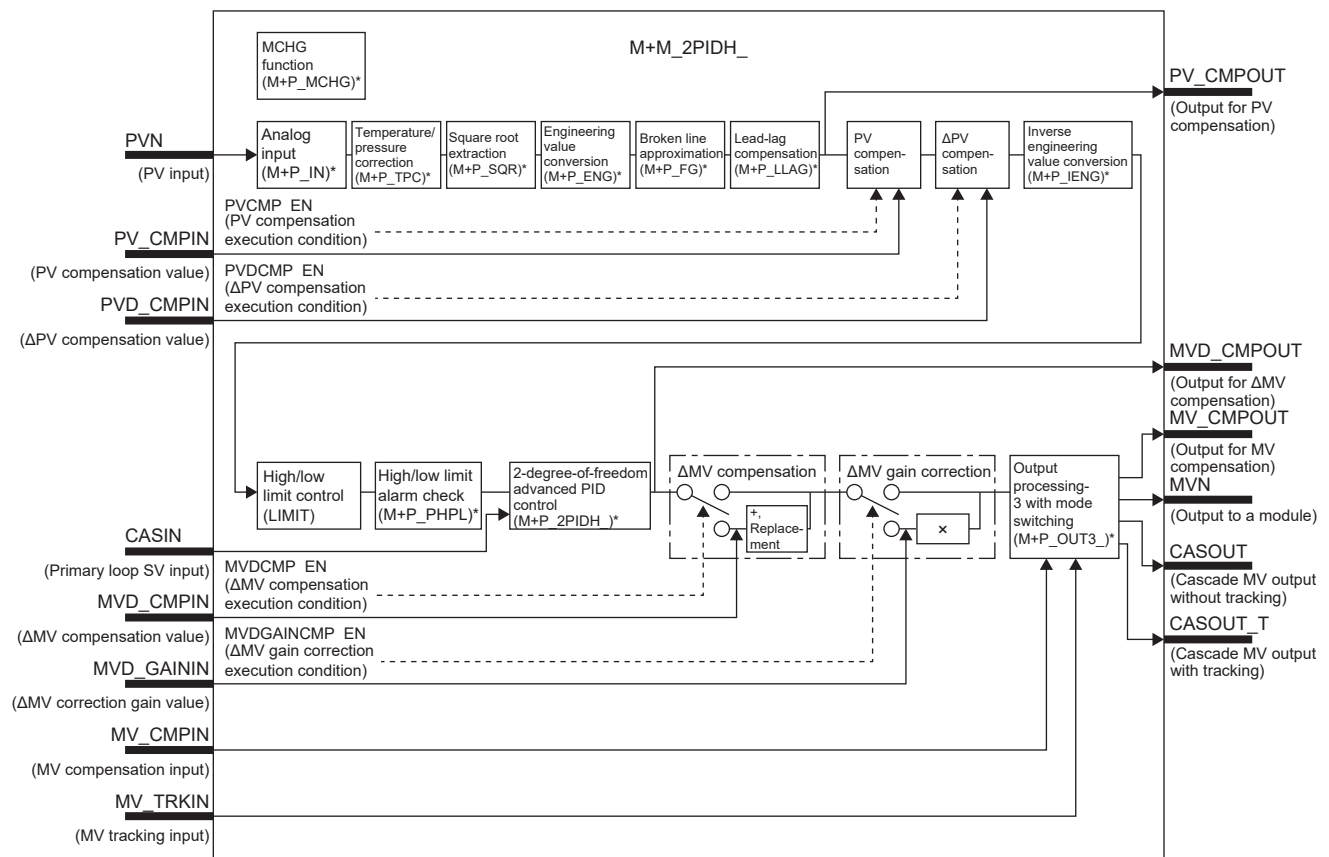
FBD/LD

```
         M+M_2PIDH_
─│PVN                    MVN│─
─│CASIN              CASOUT│─
─│PV_CMPIN        CASOUT_T│─
─│PVD_CMPIN      PV_CMPOUT│─
─│MVD_CMPIN     MVD_CMPOUT│─
─│MVD_GAININ     MV_CMPOUT│─
─│MV_CMPIN                 │
─│MV_TRKIN                 │
```

Applicable tag type

2PIDH

Control mode

MAN	AUT	CAS[*1]	CMV	CSV
○	○	○	○	○

*1 Transition to CASDR is possible.

■Block diagram

* (M+P_□) indicates the main structure member tag access FB which is the main component of each function.

Setting data

■Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVN	Input from a module	-999999 to 999999	Input variable	REAL
CASIN	Primary loop SV input (Unit: %)	0 to 100[%]	Input variable	REAL
PV_CMPIN	PV compensation value	-999999 to 999999	Input variable	REAL
PVD_CMPIN	ΔPV compensation value	-999999 to 999999	Input variable	REAL
MVD_CMPIN	ΔMV compensation value (Unit: %)	-100 to 100[%]	Input variable	REAL
MVD_GAININ	ΔMV correction gain value	-999999 to 999999	Input variable	REAL
MV_CMPIN	MV compensation value (Unit: %)	-999999 to 999999 [%]	Input variable	REAL
MV_TRKIN	MV tracking input (Unit: %)	0 to 100[%]	Input variable	REAL
MVN	Output to a module	OUT3_NMIN to OUT3_NMAX	Output variable	REAL
CASOUT	Cascade MV output (Unit: %)	0 to 100[%]	Output variable	REAL
CASOUT_T	Cascade MV output (Unit: %) (With tracking) (Indirect address)	0 to 100[%]	Output variable	DWORD
PV_CMPOUT	Output for PV compensation	-999999 to 999999	Output variable	REAL
MVD_CMPOUT	Output for ΔMV compensation (Unit: %)	-100 to 100[%]	Output variable	REAL
MV_CMPOUT	Output for MV compensation (Unit: %)	0 to 100[%]	Output variable	REAL

■Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
IN_NMAX	Input High Limit	-999999 to 999999	100.0	User	REAL
IN_NMIN	Input Low Limit	-999999 to 999999	0.0	User	REAL
IN_HH	High Limit Range Error	-999999 to 999999	102.0	User	REAL
IN_H	High Limit Range Error Reset	-999999 to 999999	100.0	User	REAL
IN_L	Low Limit Range Error Reset	-999999 to 999999	0.0	User	REAL
IN_LL	Low Limit Range Error	-999999 to 999999	-2.0	User	REAL
TPC_SQR	Temperature Pressure Correction pattern	0: None 1: Square root extraction 2: Temperature correction + Square root extraction 3: Pressure correction + Square root extraction 4: Temperature/pressure correction + Square root extraction	0	User	INT
TPC_PVTEMP	Temperature Pressure Correction: Measured Temperature (Engineering Value)	-999999 to 999999	0.0	User	REAL
TPC_PVPRES	Temperature Pressure Correction: Measured Pressure (Engineering Value)	-999999 to 999999	0.0	User	REAL
TPC_TEMP	Temperature Pressure Correction: Design Temperature	-999999 to 999999	0.0	User	REAL
TPC_B1	Temperature Pressure Correction: Bias Temperature	-999999 to 999999	273.15	User	REAL
TPC_PRES	Temperature Pressure Correction: Design Pressure	-999999 to 999999	0.0	User	REAL
TPC_B2	Temperature Pressure Correction: Bias Pressure	-999999 to 999999	10332.0	User	REAL
SQR_OLC	Square Root Extraction: Output Low Cut-off Value	0 to 999999	0.0	User	REAL
SQR_K	Square Root Extraction: Coefficient	0 to 999999	10.0	User	REAL
SQR_DENSITY	Square Root Extraction: Density Correction Value	0 to 999999	1.0	User	REAL
FG_SN	Function Generator: Number of Points	0 to 48	0	User	INT
FG_X1 to FG_X48	Function Generator: Input coordinate (x-coordinate)	-999999 to 999999	0.0	User	REAL
FG_Y1 to FG_Y48	Function Generator: Output coordinate (y-coordinate)	-999999 to 999999	0.0	User	REAL

Variable name	Description	Recommended range	Initial value	Set by	Data type
LLAG_EN	First Order Lag: Execution condition	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
LLAG_T1	First Order Lag: Delay Time (Sec)	0 to 999999 [s]	1.0	User	REAL
PVCMP_EN	PV Compensation Execution condition	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
PVCMP_MODE	PV Compensation Mode	0: Addition 1: Replacement	0	User	INT
PVDCMP_EN	ΔPV Compensation Execution condition	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
PID2H_MTD	Derivative Gain	0 to 9999	8.0	User	REAL
PID2H_DVLS	Large Deviation Alarm Hysteresis	0 to 100	2.0	User	REAL
PID2H_PN	Reverse Action/Direct Action	0: Reverse action 1: Direct action	0	User	INT
PID2H_SVPTN_B0	Setting Value (SV) Used	TRUE: Not used FALSE: Used	TRUE	User	BOOL
PID2H_PVTRK_EN	PV Tracking Execution condition	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
PID2H_ISTP	Integration Stop Signal	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
PID2H_DSTP	Derivation Stop Signal	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
PID2H_LMT_ISTP	Stop Integration, when MV variation rate limiter alarm occurred	TRUE: Stop FALSE: Not stop	FALSE	User	BOOL
PID2H_SVLMT_EN	SV High/Low Limiter	TRUE: Execute FALSE: Not execute	FALSE	User	BOOL
MVDCMP_EN	ΔMV Compensation Execution condition	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
MVDCMP_MODE	ΔMV Compensation Mode	0: Addition 1: Replacement	0	User	INT
MVDGAINCMP_EN	ΔMV Gain Correction Execution condition	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
OUT3_NMAX	Output Conversion High Limit	-999999 to 999999	100.0	User	REAL
OUT3_NMIN	Output Conversion Low Limit	-999999 to 999999	0.0	User	REAL
OUT3_MVCMP_EN	MV Compensation Execution condition	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
OUT3_MVCMP_MODE	MV Compensation Mode	0: Addition 1: Replacement	0	User	INT
OUT3_PREMV_EN	Preset MV Execution condition	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
OUT3_PREMV_V	Preset MV Value (Unit: %)	0 to 100[%]	0.0	User	REAL
OUT3_MVHLD_EN	MV Hold Execution condition	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
OUT3_MVTRK_EN	MV Tracking Execution condition	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
OUT3_STP_OTYPE	Output when loop or tag is stopped	0: Hold 1: Preset value	0	User	INT
OUT3_SEA_OTYPE	MV Output Selection when SEA is occurred	0: Hold 1: Preset MV output 2: Neither hold nor preset MV output is performed.	0	User	INT
OUT3_ARW_EX_EN	MV Value Instantaneous Pullback	TRUE: Execute FALSE: Not execute	FALSE	User	BOOL
OUT3_MVPH	MV Internal Operation High Limit Value (Unit: %)	MH to 999999[%]	100.0	User	REAL
OUT3_MVPL	MV Internal Operation Low Limit Value (Unit: %)	-999999 to ML[%]	0.0	User	REAL
OUT3_MVREV_EN	MV Reverse Execution condition	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
OUT3_FOTS_EN	Tight Shut/Full Open Execution condition	TRUE: Execute FALSE: Stop	FALSE	User	BOOL

A

Variable name	Description	Recommended range	Initial value	Set by	Data type
OUT3_MVFO	Output Value for Full Open (Unit: %)	100 to 125[%]	112.5	User	REAL
OUT3_MVTS	Output Value for Tight Shut (Unit: %)	-25 to 0[%]	-16.82	User	REAL

■Public variable (others) [*1]

• Simulation processing

Variable name	Description	Recommended range	Initial value	Set by	Data type
SIMIN	Simulation Input	NMIN to NMAX	0.0	User	REAL
SIMOUT	Simulation Output	NMIN to NMAX	0.0	System	REAL

• Control mode change processing (MCHG function)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MODEIN	Mode Change Signal	1 to 6 1: MAN 2: AUT 3: CAS 4: CMV 5: CSV 6: CASDR	0	User	INT
E_	Enable Mode Change	TRUE: Execute FALSE: Stop	FALSE	User	BOOL

*1 Read or write the variables using a program. They are not displayed on "FB Property" of the engineering tool.

■Tag data

For details on tag data that is read or written with this tag FB, refer to the following.

Processing details

This tag FB consists of the following function blocks.

Item	Function block	Reference
IN function	M+P_IN	📖 MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks)
TPC function	M+P_TPC	
SQR function	M+P_SQR	
ENG function	M+P_ENG	
FG function	M+P_FG	
LLAG function	M+P_LLAG	
IENG function	M+P_IENG	
LIMIT function	LIMIT	
PHPL function	M+P_PHPL	
2PIDH function	M+P_2PIDH_	
OUT3 function	M+P_OUT3_	
MCHG function	M+P_MCHG	

■PV compensation

This function block adds an externally-input compensation value to a process variable or replaces a process variable with an externally-input compensation value.

Condition		Processing result
PVCMP_EN = TRUE	PVCMP_MODE = 0 (addition)	IN + PV_CMPIN
	PVCMP_MODE = 1 (replacement)	PV_CMPIN
PVCMP_EN = FALSE		IN

IN: Input value (Process variable), PV_CMPIN: Compensation value, PVCMP_MODE: Compensation mode

■ΔPV compensation

This function block adds a ΔPV compensation value (PVD_CMPIN) to an internal addition value (ΣPVD_CMPIN) when PVDCMP_EN is enabled.

ΣPVD_CMPIN is added to a process variable.

■ΔMV compensation

This function block adds an externally-input compensation value to ΔMV or replaces ΔMV with an externally-input compensation value.

Condition		Processing result
MVDCMP_EN = TRUE	MVDCMP_MODE = 0 (addition)	IN + MVD_CMPIN
	MVDCMP_MODE = 1 (replacement)	MVD_CMPIN
MVDCMP_EN = FALSE	—	IN

IN: Input value (ΔManipulated value), MVD_CMPIN: Compensation value, MVDCMP_MODE: Compensation mode

■ΔMV gain correction

This function block multiplies ΔMV by a gain correction value.

Condition	Processing result
MVDGAINCMP_EN = TRUE	IN × MVD_GAININ
MVDGAINCMP_EN = FALSE	IN

IN: Input value (ΔManipulated value), MVD_GAININ: Gain correction value

Operation error

Error code (SD0)	Description
3402H	An internal operation result of a function block, input data, operation constant, or tag data is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.

An error may occur in a function block that is used in a tag FB. For details on the error, refer to corresponding function blocks.

A

Multi-point program setter (M+P_PGS2_)

M+P_PGS2_

This function block registers time width sets and setting value programs of up to 32 steps, and outputs a setting value corresponding to the progress time of each step with linear interpolation.

FBD/LD

```
        ┌─────────────────────┐
        │      M+P_PGS2_       │
      ──┤PVIN          CASOUT  ├──
      ──┤ADVANCE       PTNEND  ├──
      ──┤INITSTART     LINKOUT ├──
      ──┤LINKIN                │
        └─────────────────────┘
```
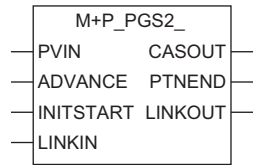
Applicable tag type

PGS2

Control mode

MAN	AUT	CAS	CMV	CSV
○	○	—	—	—

■Block diagram
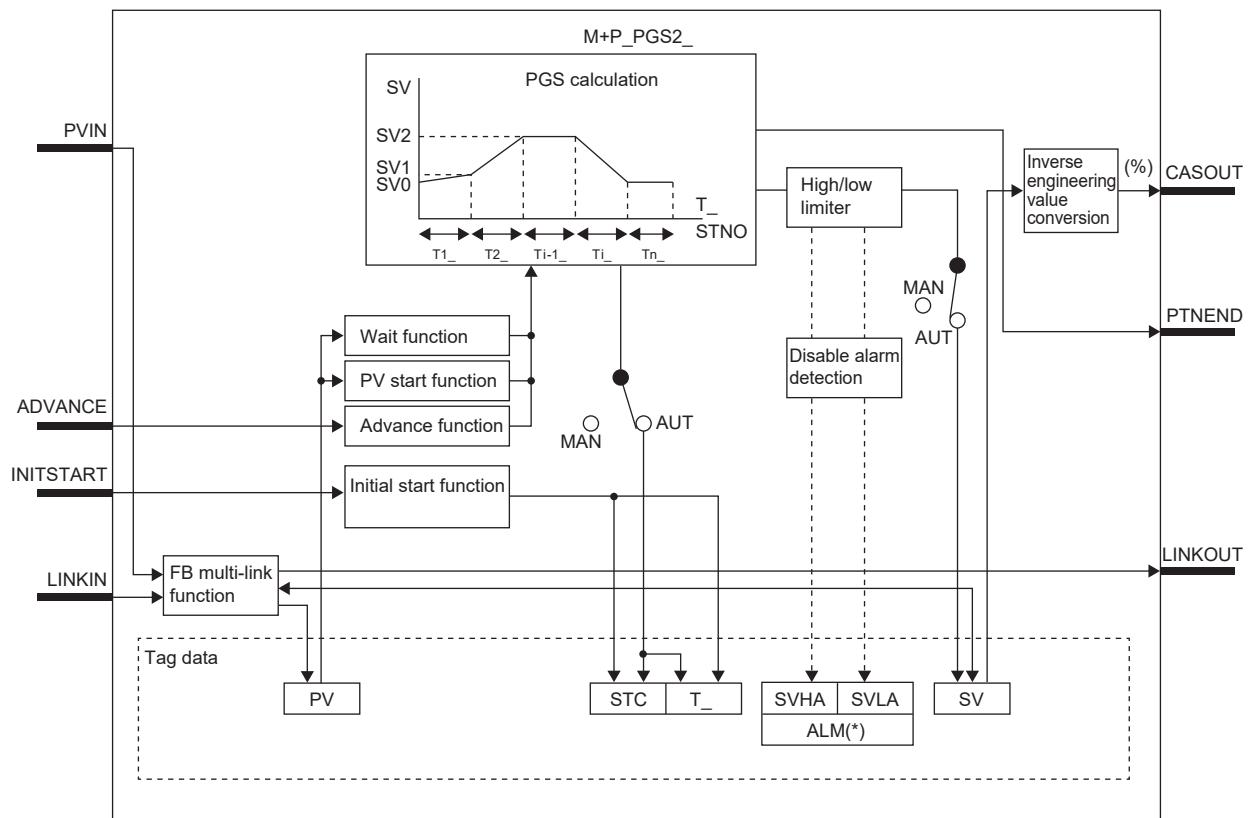
* Indicates a bit item.

Setting data

■Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVIN	Process input (Engineering value)	-32768 to 32767	Input variable	REAL
ADVANCE	Advance command	TRUE, FALSE	Input variable	BOOL
INITSTART	Initial start command	TRUE, FALSE	Input variable	BOOL
LINKIN	Link input (Indirect address)	—	Input variable	DWORD
CASOUT	Cascade output (Unit: %)	0 to 100[%]	Output variable	REAL
PTNEND	Pattern end output	TRUE, FALSE	Output variable	BOOL
LINKOUT	Link output (Indirect address)	—	Output variable	DWORD

■Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
PVSTARTNO	PV Start Search Start Step	1 to 32	1	User	INT
PVENDNO	PV Start Search End Step	1 to 32	32	User	INT
PRIMARY	Lead FB specified	TRUE: Lead FALSE: Following	TRUE	User	BOOL

■Public variable (others) [*1]

Variable name	Description	Recommended range	Initial value	Set by	Data type
TCNT	Second counter for minute mode	0 to 59	0	System	INT
TMCNT	Millisecond counter for second mode	0 to 999	0	System	INT

*1 Read or write the variables using a program. They are not displayed on "FB Property" of the engineering tool.

■Tag data

For details on tag data that is read or written with this tag access FB, refer to the following.

☞ Page 199 PGS2

A

Processing details

■PV start

At the start of a control (when the AUT mode is switched), this function block reduces a gap with process variable by referring to the process variable and adjusting the control start point. The adjusted control start point is set as the start point (current) (SV0C).

PV start type	Control start condition				
	STNO = 0	STNO ≠ 0			
		STC = 0 and T_ = 0	STC = 0 and T_ ≠ 0	STC ≠ 0	After the second cycle at cyclic
PVSTART = 0	(1)	SV0 start (2)	(5)	(6)	(7)
PVSTART = 1		PV start 1 (3)			
PVSTART = 2		PV start 2 (4)			

• (1) When the number of steps (STNO) is 0

Because the step setting is invalid, the control mode will be immediately switched to MAN mode, if the mode has been switched to AUT mode, and the pattern completion output (PTNEND) will turn on by only one cycle. The system operates in MAN mode, and the FB multi-link function and inverse engineering value conversion are activated.

• (2) SV0 start (Fixed start point: PVSTART = 0)

The control will start using the start point (SV0) as the control start point without referring to the process variable.
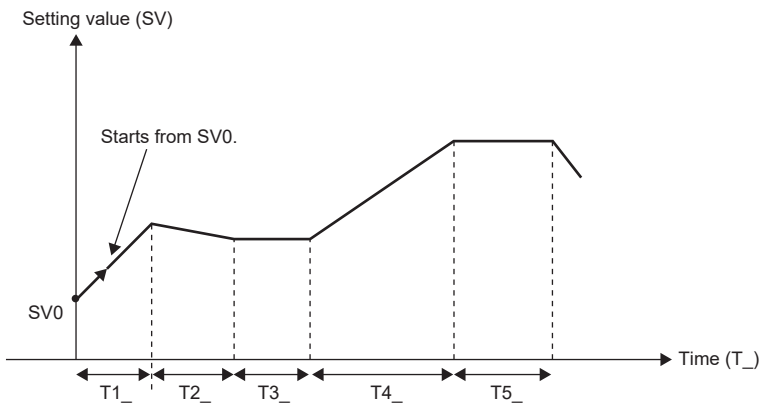
• (3) PV start 1 (Start point correction: PVSTART = 1)

The control will start using the process variable as the control start point.
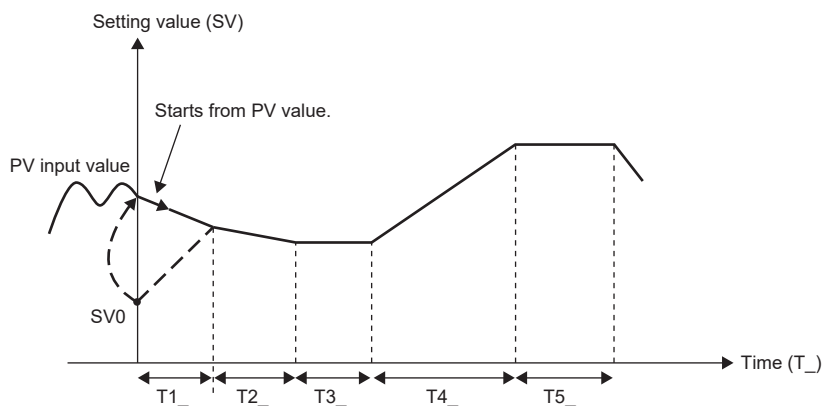
- • (4) PV start 2 (Start point searching: PVSTART = 2)

The process variable and the PV start search start step (PVSTARTNO) are used to search a match point within the range specified by the PV start search end step (PVENDNO), and the control will start with the step number or time.

When no match point is found, the closest point (maximum point or minimum point) is used.

If multiple maximum points or minimum points are found, the point that comes last is used.

When a match point is the end point of the last step, the control will be immediately completed and the setting value before the mode switching to AUT mode will be held.

Ex.
When a match point was found within the search range
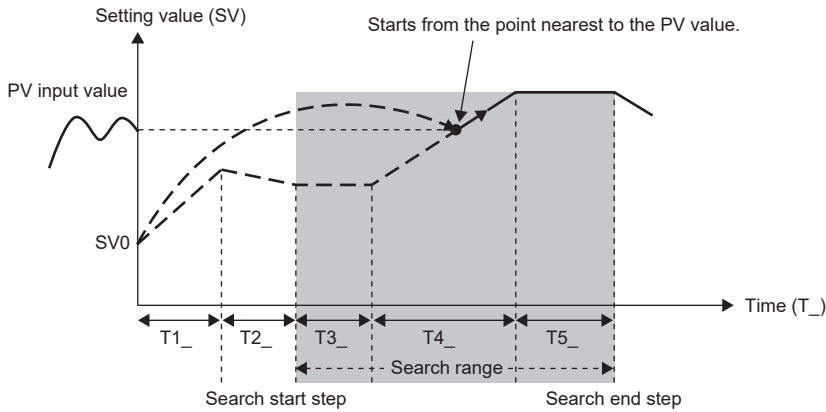
Ex.
When no match point was found within the search range
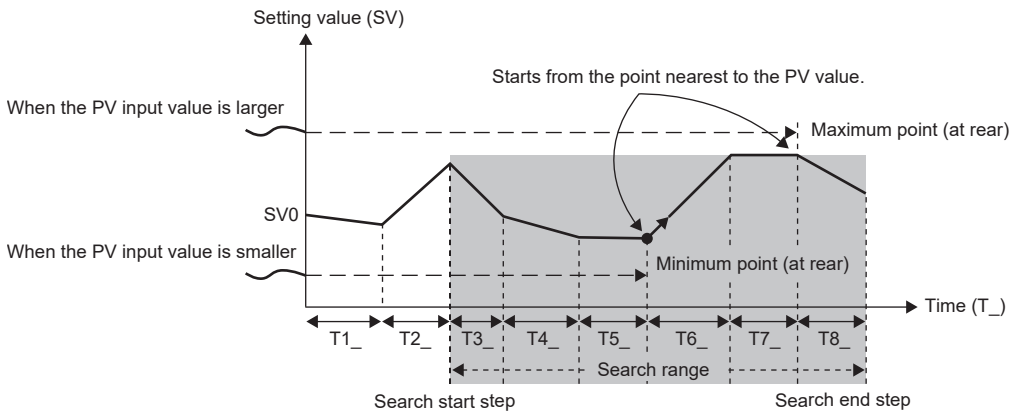
• (5) Starting a control when STC = 0 and T_ ≠ 0

The control will start from the current time (T_) at STC = 1 using the start point (SV0) as the control start point.

• (6) Starting a control when STC ≠ 0

The control will start from the current STC and T_ using the control start point (start point (current) (SV0C)) of the previous control (AUT mode).

• (7) After the second cycle at cyclic

The control will start using the last setting value (SVn) as the next control start point (start point (current) (SV0C)).

■PGS calculation

In AUT mode, this function block outputs setting values to the time predetermined by each step as time proceeds.

There are the following three operation types: hold type, return type, and cyclic type.

Time width sets and setting values of up to 32 steps can be registered in the format of $(Tn\_, SVn)$.

The control start point depends on the PV start type (PVSTART) setting.
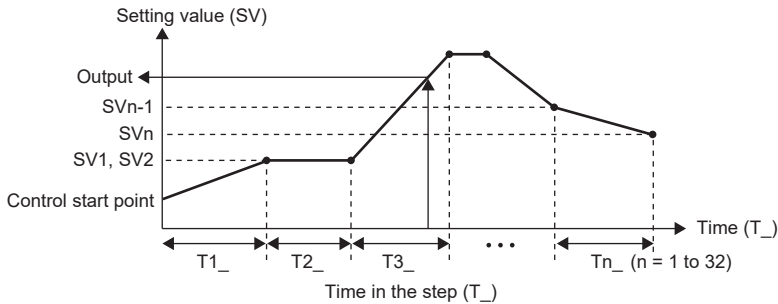
- Step specification execution

In AUT mode, changing the executing step number (STC) jumps to the start of a specified step $(T\_ = 0)$.

Changing the time in the step $(T\_)$ jumps to the time of the step.

To jump to the end in the step $(T = Tn)$ by changing the time in the step $(T\_)$ when the wait function is valid, refer to the wait function.

- Step management

Progress processing of the time in the step $(T\_)$ and the executing step number (STC) is performed in AUT mode.

Condition		Processing result	
		Time in the step $(T\_)$	Executing Step No. (STC)
$STC \leq 0$		0	1
$STC > 0$	$T\_ < 0$	0	Last value
	$MV < 0 \leq T\_ < Ti\_$	$T\_ + \Delta T$ [*1]	Last value
	$Ti\_ \leq T\_$ [*2]	0	Move to the next step (STC + 1)

ΔT: Execution cycle, i: Executing step number (STC)

*1　The execution cycle addition of the time in the step $(T\_)$ is calculated in real numbers with a resolution in units of milliseconds when the second has been specified for the unit of time (TUNIT), or with a resolution in units of seconds when the minute has been specified for the unit of time (TUNIT).

*2　To enable and use the wait function, refer to the wait function.

- SV_{PGS} calculation

In AUT mode, a setting value is calculated according to the executing step number (STC) and the time in the step $(T\_)$.

Condition	Processing result
$Ti\_ \leq 0$	SVi
$0 < T\_ \leq Ti\_$	$\dfrac{SVi - SVi - 1}{Ti\_} \times T\_ + Sv_{i-1}$ (SV_{PGS} is calculated in real numbers.)

i: Executing step number (STC)

• Processing of when the last step processing is completed

In AUT mode, the pattern end output (PTNEND) is turned on by one cycle and the following processing is performed after patterns are performed and the last step is ended.

Item	Operation Type		
	Hold	Return	Cyclic
Control mode transition	Transition to MAN	Transition to MAN	No transition
Time in the step (T_)	Last value	0	0
Executing Step No. (STC)	Last value	0	1
SV$_{PGS}$ calculation	SV output value of the last step (SVn)	SV output value of the last step (SVn)	Processing is restarted from step 1 with the SV output value of the last step (SVn) as the control start point.

Point

• Setting the time width T1_ to Tn_ in PGS calculation

The time width is set in integer number. The number of seconds or minutes is set in a loop tag.

The unit of time is common to all steps and is specified by the unit of time (TUNIT) of tag data.

The maximum value of the time width setting is 32767 seconds (approximately 9 hours) or 32767 minutes (approximately 22 days) for each step.

• Setting the setting value SV1 to SVn in PGS calculation

Set values in engineering values. The setting range is -32768 to 32767 and set the values as engineering values. A value with a decimal point cannot be specified.

When a value with a decimal point needs to be set depending on the engineering value range, when 1.5MPa is set, for example, convert the value into 1500kPa to fit the value within the range of -32768 to 32767.

■Advance function

In AUT mode, this function block proceeds to the next step by forcibly ending the step in execution.
This function is executed on the rising edge of the input variable ADVANCE command and proceeds the processing by one step. To proceed the processing by one step next time, turn off the command and then turn it on again.
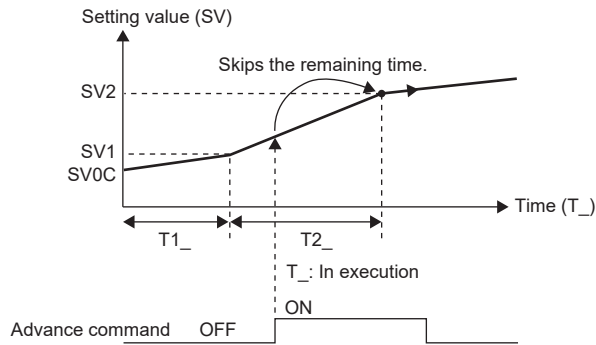
When the advance command is executed in the waiting status, the status is cleared and the processing shifts to the one of the next step.
When the advance command is executed at the last step, the control will be immediately completed and the setting value before the execution of the advance command will be held.

■Wait function

In AUT mode, this function block checks if the process variable (PV) has followed the setting value (SV), and controls the progress of steps every time each step is completed.
The wait width setting is common to all steps.

Condition			Processing result
Control mode	Wait Width	Process Variable	
MAN	—	—	The wait function is not executed.
AUT	WAIT ≤ 0	—	
	WAIT > 0	$|PV - SV| > WAIT$	Transition to the next step stops. The last value of the step is held as the setting value (SV).[1]
		$|PV - SV| ≤ WAIT$	Transition to the next step is performed.

*1 When the time in the step was changed and the step was completed (T = Tn), the setting value (SVPGS) immediately before the change of the time in the step is held instead of the last value of the step. To hold the last value of the step (SVn), change the time in the step to the time immediately before the end of the step (T = Tn - 1).
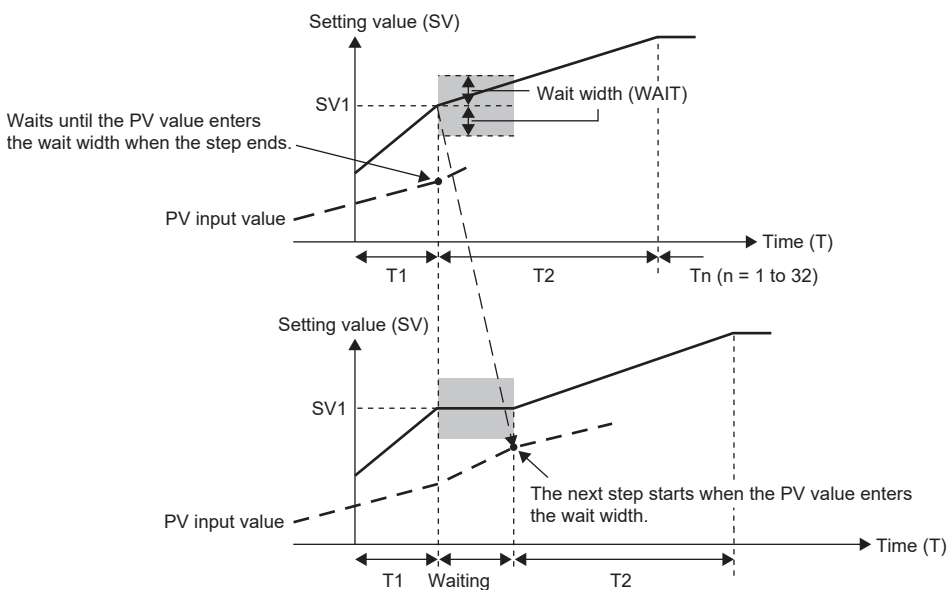
■Disable alarm detection

This function block sets whether to detect alarms (ALM) or not in the setting value high/low limiter processing.

Item	Disable alarm detection processing
Disable alarm detection by setting tag data	If the following bits in the disable alarm detection (INH) are TRUE, SVHA or SVLA of the alarm (ALM) will not be detected. • ERRI • SVHI • SVLI
Disable alarm detection by loop stop processing	☞ Page 218 Loop stop processing

■Setting value high/low limiter

In AUT mode, this function block checks the high/low limits.

Condition		Setting value (SV)	Alarm (ALM)	
			SV low limit (SVLA)	SV high limit (SVHA)
MAN		SV_{PGS}	FALSE (Reset)	FALSE (Reset)
AUT	$SV_{PGS} > SH$	SH	FALSE (Reset)	TRUE (Detected)
	$SV_{PGS} < SL$	SL	TRUE (Detected)	FALSE (Reset)
	$SL \leq SV_{PGS} \leq SH$	SV_{PGS}	FALSE (Reset)	FALSE (Reset)

Control mode	Description
MAN	
AUT	

SH: SV high limit value
SL: SV low limit value
SVHA: SV high limit alarm
SVLA: SV low limit alarm

■**Inverse engineering value conversion**

This function block converts the setting value (SV) into a value in percentage (%) within the engineering value range of RL to RH.

$$\text{CASOUT (\%)} = \frac{\text{SV} - \text{RL}}{\text{RH} - \text{RL}} \times 100$$

CASOUT: Cascade output
RH: Engineering value high limit
RL: Engineering value low limit
SV: Setting value
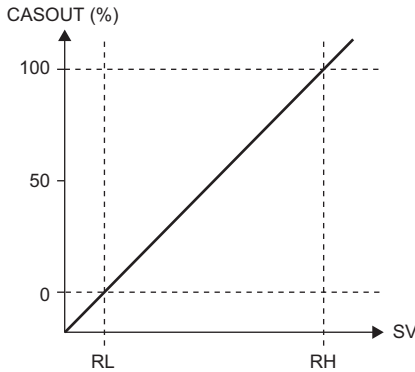
■**Initial start function**

In MAN mode, this function block initializes the executing step number (STC) and the time in the step (T_) on the rising edge of an input variable INITSTART command, and switches the control mode to the AUT mode.

Type	Variable name	Data type	Description	Value to be initialized
Loop tag	STC	INT	Executing Step No.	0
	T_	INT	Time in the step	0
Public variable	TCNT	INT	Second counter for minute mode	0
	TMCNT	INT	Millisecond counter for second mode	0

■**FB multi-link function**

To create a program having over 32 steps, link multiple tag FBs of the M+M_PGS2_ type or user-defined tag FBs of the PGS2 type so that they function as a program setter, and perform the following operations.

 • Use a common setting value for the multiple FBs so that the same setting value will be the last output value even after the setting value of an FB has changed.
 • The control mode of the following FB is preferentially managed, so that multiple FBs are not switched to AUT mode when the control mode is switched to AUT mode.
 • The process variable input to the lead FB is copied to the process variable of the following FB.

For programs having multi-linked FBs, refer to the following.

A

■**Loop stop processing**

When the stop alarm (SPA) of the alarm (ALM) or the tag stop (TSTP) of the monitor output buffer (DOM) is TRUE, this function block performs loop stop processing.

Loop stop processing result					
Input (PV)	Executing Step No. (STC)/ Time in the step (T_)	Output (SV)	Mode	Alarm reset[1]	Alarm detected[2]
Hold	Hold	Hold	MAN	Reset	Not detected

*1 When SVLA or SVHA of the alarm (ALM) has occurred, the SVLA or SVHA is reset.
*2 No alarm is detected in the setting value high/low limiter processing.

■**Processing operation**

○: Performed, ×: Not performed

Control mode	Processing operation				
	PV start	PGS calculation	Advance function	Wait function	Alarm
MAN	×	×	×	×	×
AUT	○	○	○	○	○[1]

Control mode	Processing operation			
	Setting value high/low limiter	Inverse engineering value conversion	Initial start function	FB multi-link function
MAN	×	○	○	○
AUT	○	○	×	○

*1 An alarm whose corresponding bit is TRUE (Valid) in the disable alarm detection (INH) is not detected.

Operation error

Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	An internal operation result of a function block or input data is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.

Multi-point program setter (M+M_PGS2_)

M+M_PGS2_

This function block performs program settings combining the functions of M+P_PGS2_ into a single FB. Multiple FBs can be connected to execute time width sets and setting value programs of 32 steps or more.

FBD/LD

```
        ┌──────────────────┐
        │    M+M_PGS2_     │
 ───────┤ PVIN      CASOUT ├───────
 ───────┤ ADVANCE   PTNEND ├───────
 ───────┤ INITSTART LINKOUT├───────
 ───────┤ LINKIN           │
        └──────────────────┘
```

Applicable tag type

PGS2

Control mode

MAN	AUT	CAS	CMV	CSV
○	○	—	—	—

■Block diagram

* (M+P_□) indicates the main structure member tag access FB which is the main component of each function.

Setting data

■Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVIN	Process input (Engineering value)	-32768 to 32767	Input variable	REAL
ADVANCE	Advance command	TRUE, FALSE	Input variable	BOOL
INITSTART	Initial start command	TRUE, FALSE	Input variable	BOOL
LINKIN	Link input (Indirect address)	—	Input variable	DWORD
CASOUT	Cascade output (Unit: %)	0 to 100[%]	Output variable	REAL
PTNEND	Pattern end output	TRUE, FALSE	Output variable	BOOL
LINKOUT	Link output (Indirect address)	—	Output variable	DWORD

■Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
PGS2_PVSTARTNO	PV Start Search Start Step	1 to 32	1	User	INT
PGS2_PVENDNO	PV Start Search End Step	1 to 32	32	User	INT
PGS2_PRIMARY	Lead FB specified	TRUE: Lead FALSE: Following	TRUE	User	BOOL

■Public variable (others) [*1]

• M+P_PGS2_ processing

Variable name	Description	Recommended range	Initial value	Set by	Data type
PGS2_TCNT	Second counter for minute mode	0 to 59	0	System	INT
PGS2_TMCNT	Millisecond counter for second mode	0 to 999	0	System	INT

• Control mode change processing (MCHG function)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MODEIN	Mode Change Signal	1, 2 1: MAN 2: AUT	0	User	INT
E_	Enable Mode Change	TRUE: Execute FALSE: Stop	FALSE	User	BOOL

*1 Read or write the variables using a program. They are not displayed on "FB Property" of the engineering tool.

■Tag data

For details on tag data that is read or written with this tag FB, refer to the following.

Processing details

This tag FB consists of the following function blocks.

Item	Function block	Reference
PGS2 function	M+P_PGS2_	☞ Page 208 M+P_PGS2_
MCHG function	M+P_MCHG	📖 MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks)

Operation error

An error may occur in a function block that is used in a tag FB. For details on the error, refer to corresponding function blocks.

Program example

■When the output of a single M+M_PGS2_ function block is used

Set the following items.

Type/pin	Variable name	Description	Setting/connection method
Public variable	PGS2_PRIMARY	Lead FB specified	TRUE
Input pin	LINKIN	Link input	Not connected

A
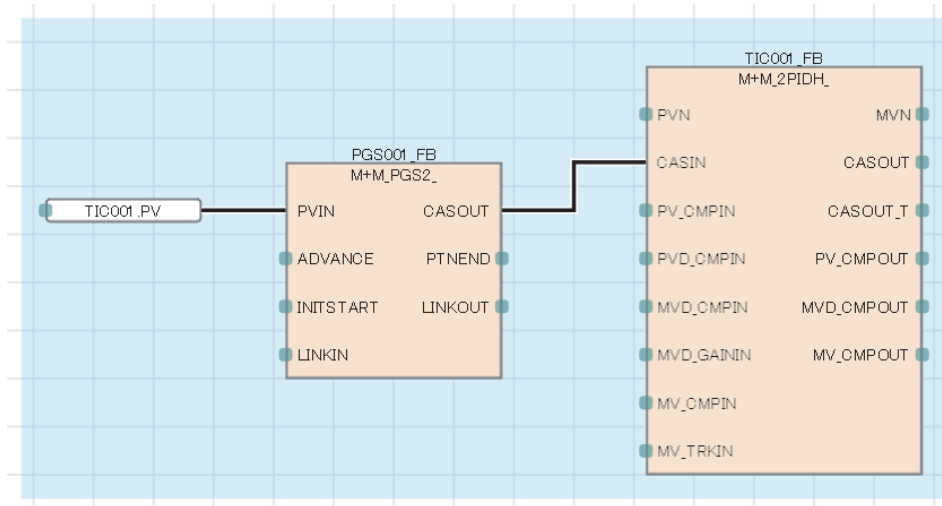
■When multiple M+M_PGS2_ function blocks are connected (When a program with 32 steps or more is used)

Set the following items.

Target FB	Type/pin	Variable name	Description	Setting/connection method
Lead FB	Public variable	PGS2_PRIMARY	Lead FB specified	TRUE
	Input pin	LINKIN	Link input	Not connected
	Output pin	PTNEND	Pattern end output	Connected with INITSTART of the following FB
		LINKOUT	Link output	Connected with LINKIN of the following FB
Following FB	Public variable	PGS2_PRIMARY	Lead FB specified	FALSE
	Input pin	INITSTART	Initial start command	Connected with PTNEND of the preceding FB
		LINKIN	Link input	Connected with LINKOUT of the preceding FB
	Output pin	PTNEND	Pattern end output	Connected with INITSTART of the following FB
		LINKOUT	Link output	Connected with LINKIN of the following FB
Last FB	Public variable	PGS2_PRIMARY	Lead FB specified	FALSE
	Input pin	INITSTART	Initial start command	Connected with PTNEND of the preceding FB
		LINKIN	Link input	Connected with LINKOUT of the preceding FB
	Output pin	CASOUT	Cascade output (Unit: %)	Connected with CASIN of the following tag FB such as PID

 • Operation description

PGS001 (first program setter FB) is started in AUT mode, and the output pin PTNEND turns on for one cycle at completion.

When PGS002 (second program setter FB) receives the PTNEND output of PGS001, the mode is changed to AUT mode by the INITSTART command and the control is switched.

For details, refer to the following.

📖 MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks)
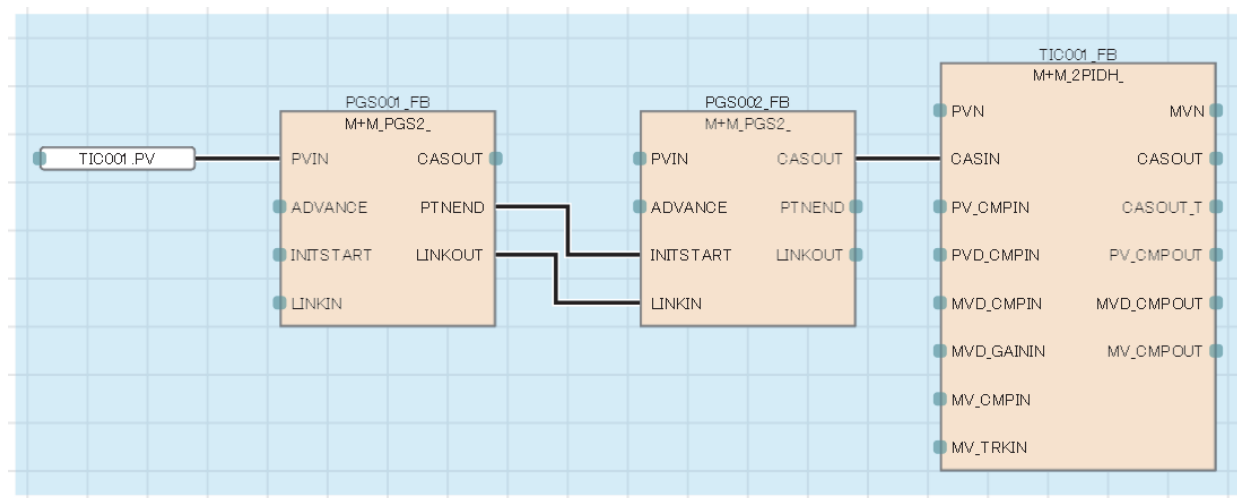
8 points time proportional output (M+P_DUTY_8PT_)

M+P_DUTY_8PT_

This function block performs output ON time conversion for input values and outputs the result in bits. This function block also adjusts the phase of output cycles automatically to suppress overlapping of output (for example, peak current).

FBD/LD
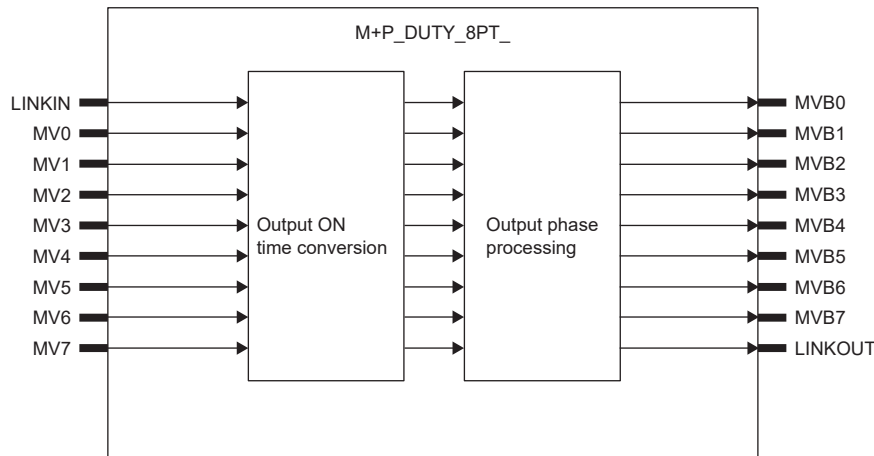
```
          M+P_DUTY_8PT_
    ─ LINKIN      MVB0 ─
    ─ MV0         MVB1 ─
    ─ MV1         MVB2 ─
    ─ MV2         MVB3 ─
    ─ MV3         MVB4 ─
    ─ MV4         MVB5 ─
    ─ MV5         MVB6 ─
    ─ MV6         MVB7 ─
    ─ MV7      LINKOUT ─
```

■Block diagram

Setting data

■Input/output variable

Variable name	Description	Recommended range	Type	Data type
LINKIN	Link input (Indirect address)	—	Input variable	DWORD
MV0 to MV7	MV input	-10 to 110[%]	Input variable	REAL
MVB0 to MVB7	Bit ON/OFF duty output	TRUE, FALSE	Output variable	BOOL
LINKOUT	Link output (Indirect address)	—	Output variable	DWORD

■Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
PRIMARY	Lead FB specified	TRUE, FALSE	TRUE	User	BOOL
CTDUTY	Control Output Cycle	0 to 9999 [s]	1.0	User	REAL
DELAY	Output ON Delay Time	0 to 9999 [s]	0.0	User	REAL

Processing details

■Output ON time conversion/Output phase processing

This function block captures an input value (MVn) every control output cycle (CTDUTY) and outputs a duty manipulated value (MVBn) to the input value.

At the same time, this function block adjusts the phase of output cycles automatically to suppress peak current.

Item	Description
Duty manipulated value (MVBn) ON time	Duty manipulated value (MVBn) ON time = ON time execution cycle count × ΔT The ON time execution cycle count is defined as the following calculation result with the first digit after the decimal point rounded off. $$\left(\frac{CTDUTY \times MVn}{\Delta T \times 100} \right)$$
Duty manipulated value (MVBn) OFF time	Duty manipulated value (MVBn) OFF time = OFF time execution cycle count × ΔT The OFF time execution cycle count is defined as (number of executions in a control output cycle) - (execution cycle count of ON time).

CTDUTY: Control output cycle
ΔT: Execution cycle
MVn: Input to the nth pin (%)
MVBn: Output from the nth pin (BOOL)

• Interval of each bit ON/OFF duty output

Output of each pin starts with a delay for the time set with DELAY after the previous pin output turns off.

When the input value from an input pin is 0%, the delay time for the output pin is ignored.

Ex.

Execution cycle: ΔT: 100ms

Control output cycle: CTDUTY = 2.0s

Input: MV0 = 25%, MV1 = 35%, MV2 = 40%, MV3 = 35%

Output ON delay time: DELAY = 0.1s

Lead FB specified: PRIMARY = TRUE
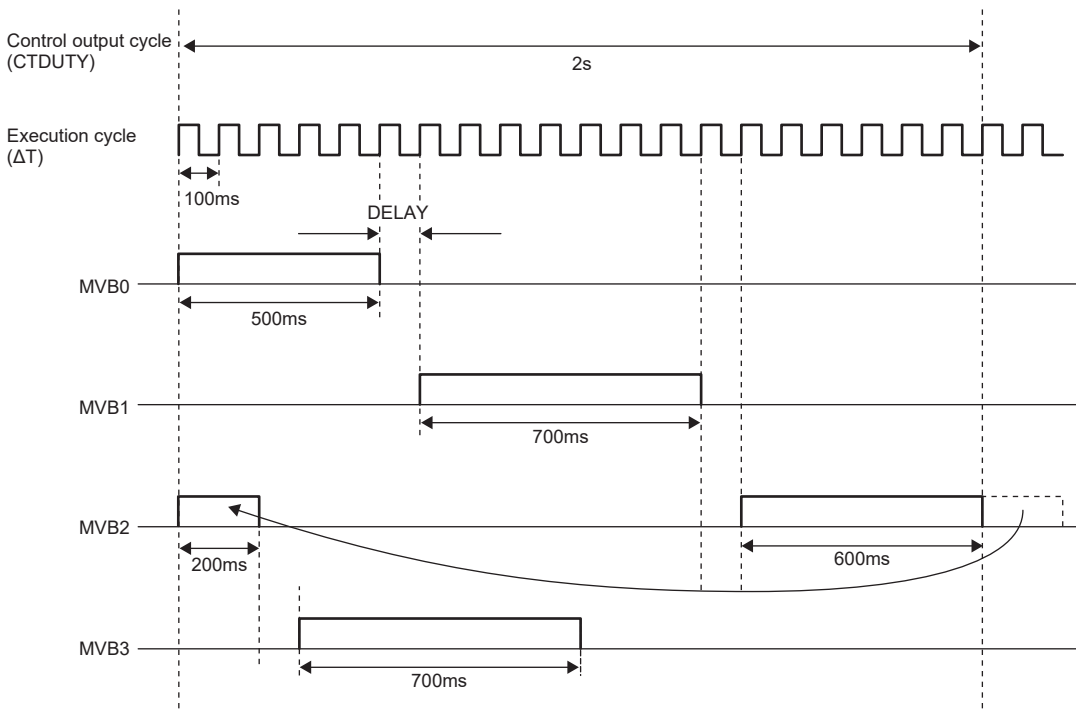
• Changing nine or more bit ON/OFF duty output phases

If multiple M+P_DUTY_8PT_ function blocks are connected, nine or more bit ON/OFF duty output phases can be changed. When connecting multiple M+P_DUTY_8PT_ function blocks, connect LINKOUT of the preceding FB with LINKIN of the following FB, set PRIMARY of the lead FB to TRUE, and PRIMARY of the following FB to FALSE. The value of CTDUTY and DELAY of the lead FB is applied to the value of CTDUTY and DELAY of the following FB.

Ex.
Execution cycle: ΔT: 100ms
- First FB
  - Control output cycle: CTDUTY = 2.0s
  - Input: MV0 = 40%, MV1 = 20%, MV2 = 40%, MV3 = 15%, MV4 = 30%, MV5 = 100%, MV6 = 0%, MV7 = 35%
  - Output ON delay time: DELAY = 0.1s
  - Lead FB specified: PRIMARY = TRUE
- Second FB
  - Control output cycle: CTDUTY = Same as CTDUTY of the preceding FB (2.0s in this example)
  - Input: MV0 = 25%
  - Output ON delay time: DELAY = Same as DELAY of the preceding FB (0.1s in this example)
  - Lead FB specified: PRIMARY = FALSE
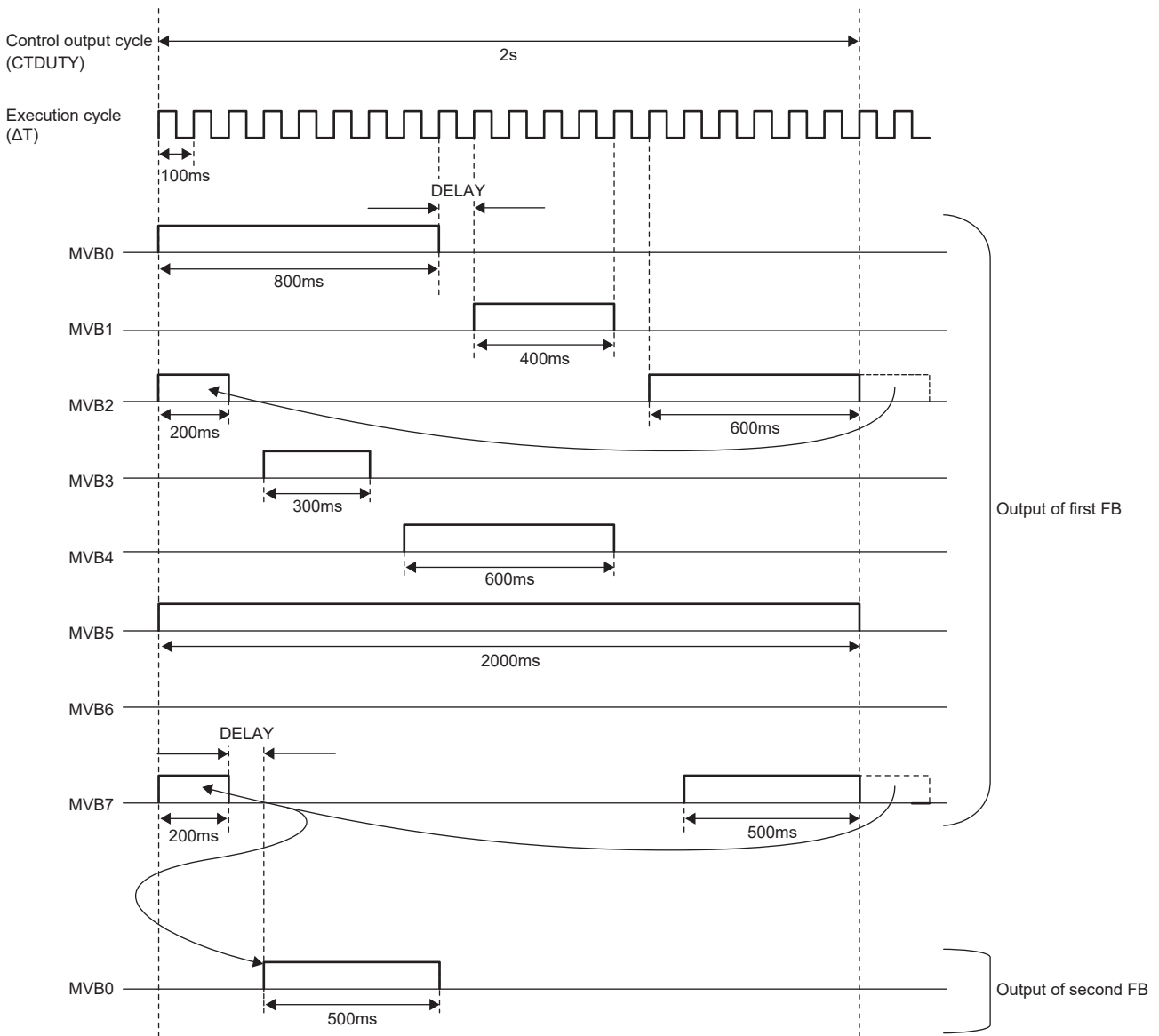
Operation error

Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Input data (MV0 to MV7) or operation constant is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.

Program example

• When one M+P_DUTY_8PT_ is used

Variable type/pin	Variable name	Description	Setting/connection method
Public variable	PRIMARY	Lead FB specified	TRUE
Input pin	LINKIN	Link input	Not connected
Output pin	LINKOUT	Link output	Not connected
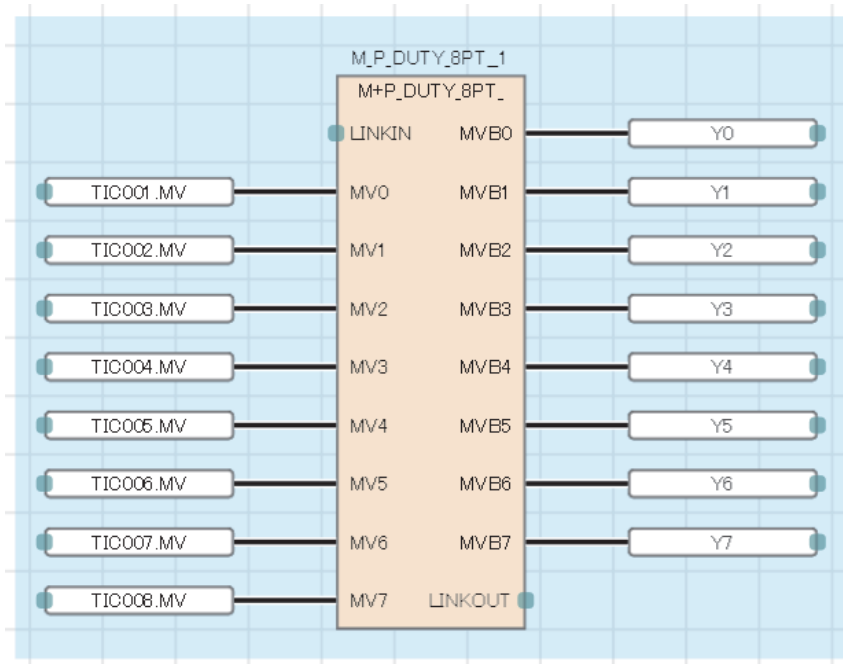
- When multiple M+P_DUTY_8PT_ function blocks are connected (when nine or more bit ON/OFF duty output phases is changed)

Target FB	Variable type/pin	Variable name	Description	Setting/connection method
Lead FB	Public variable	PRIMARY	Lead FB specified	TRUE
	Input pin	LINKIN	Link input	Not connected
	Output pin	LINKOUT	Link output	Connected with LINKIN of the following FB
Following FB	Public variable	PRIMARY	Lead FB specified	FALSE
	Input pin	LINKIN	Link input	Connected with LINKOUT of the preceding FB
	Output pin	LINKOUT	Link output	Connected with LINKIN of the following FB
Last FB	Public variable	PRIMARY	Lead FB specified	FALSE
	Input pin	LINKIN	Link input	Connected with LINKOUT of the preceding FB
	Output pin	LINKOUT	Link output	Not connected
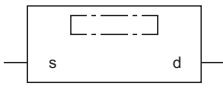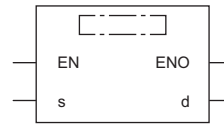
Point

When multiple M+P_DUTY_8PT_ function blocks are connected, the control output cycle and output ON delay time of the following FB used in operations can be checked with the value of the public variables CTDUTY and DELAY of the lead FB.

Converting INT to REAL

INT_TO_REAL(_E)

These functions convert a value from INT data type to REAL data type.

Ladder, FBD/LD		ST

[Without EN/ENO]
 	
─┤ s d ├─
	[With EN/ENO]
	
	─┤ EN ENO ├─
	─┤ s d ├─
		[Without EN/ENO]
		d:=INT_TO_REAL(s);
		[With EN/ENO]
		d:=INT_TO_REAL_E(EN,ENO,s);

Setting data

■Description, type, data type

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Executed, FALSE: Not executed)	Input variable	BOOL
s(IN)	Input	Input variable	INT
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Input variable	BOOL
d	Output	Input variable	REAL

Processing details

■Operation processing

• These functions convert the value input to (s) from INT data type to REAL data type, and output the converted value from (d).

(s)		(d)
1234	⟹	1234.0
INT		REAL
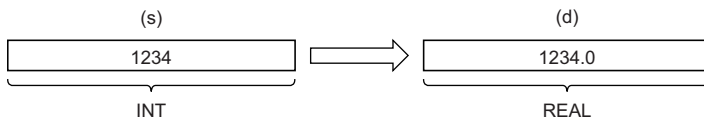
• Input an INT data type value to (s).

■Operation result

1. Function without EN/ENO

A function without EN/ENO performs operation processing and outputs the result from (d).

2. Function with EN/ENO

The execution conditions and operation results of a function with EN/ENO will be as follows.

Execution condition	Operation result	
EN	ENO	(d)
TRUE (executed)	TRUE	Operation result output value
FALSE (not executed)	FALSE[*1]	Undefined value

*1 When ENO outputs FALSE, an undefined value is output from (d). Create a program so that the undefined value will not be used in operations.

Operation error

There is no operation error.

Appendix 4 Related Functions of Process

This section describes functions related to process control.

Various controls

Cascade control

The cascade control consists of the primary loop and the secondary loop.
This control quickly detects disturbances occurred in the secondary loop and absorbs them in the secondary loop to remove the effect on the process and improve the whole control performance.
Generally, the response speed of the secondary loop should be more than three times as fast as the primary loop.

> **Point**
>
> When tracking is required, connect CASOUT_T of the primary loop with CASIN_T of the secondary loop. In addition, settings for the following operation constants of the secondary loop are necessary.
> - PID_TRK
> - PID_SVPTN_B0
> - PID_SVPTN_B1
>
> For details, refer to the following.
> ☞ Page 232 Cascade connection when tracking is required

Ex.

When controlling the furnace temperature
The fuel supply variation is absorbed by flow control of the secondary control loop, and the response characteristics of the whole temperature control are improved.
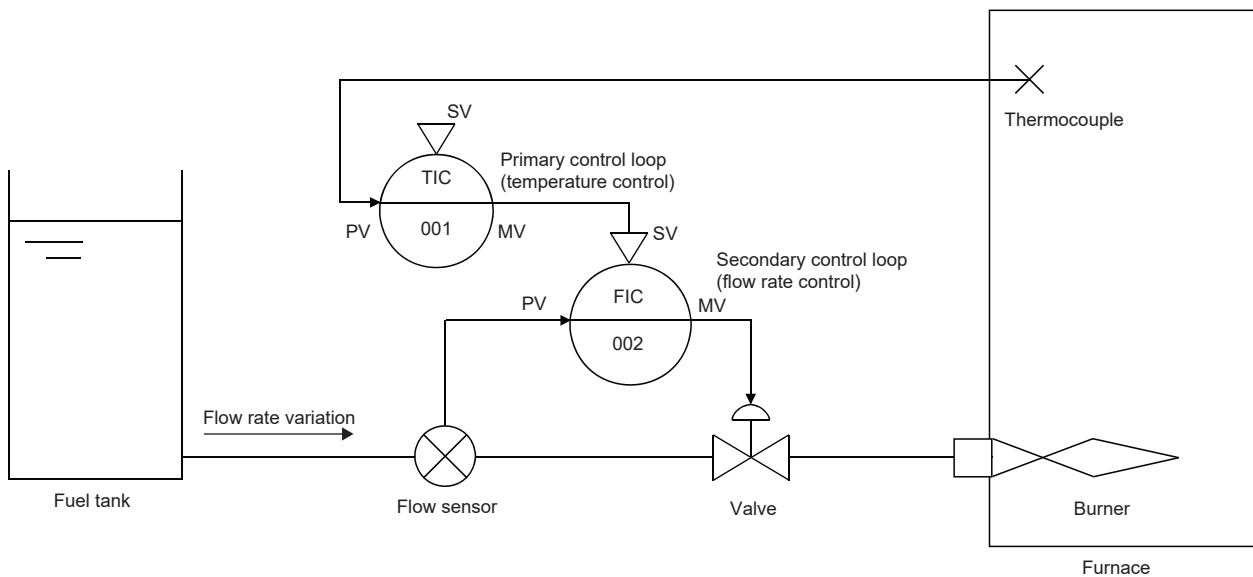
■Cascade direct

For 2PIDH tags, the CASCADE DIRECT mode can be selected as the control mode. In the cascade direct control, an output value of the primary loop is directly output as that of the secondary loop in a cascade connection.

If an input sensor has failed in the secondary loop, the PID operation result of the secondary loop will be improper. In this case, an output result of the primary loop is output as the substitute output value of the secondary loop.

The CASCADE DIRECT mode is set with the tag of the secondary loop.
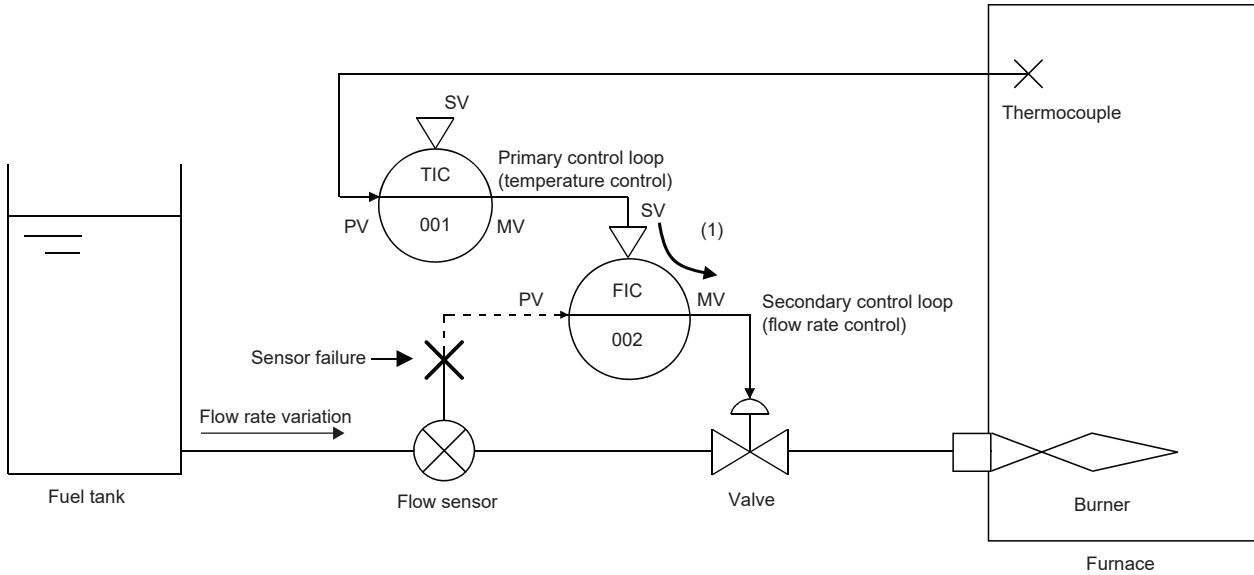
(1) MV of the primary loop is used as MV of the secondary loop.

Tracking

The tracking includes the following two functions:

Function	Description
Bumpless function	At AUTO ↔ MANUAL mode switching, this function prevents a step change caused by a sharp change of the manipulated variable (MV) output and enables a smooth change of MV.
Output limiter processing function	This function sets the upper/lower limit for the manipulated variable (MV) that is output by PID operation in AUTO mode. This function is valid only in AUTO mode. It cannot be executed in MANUAL mode. Even if the primary loop is in AUTO mode, this function is not executed while the primary loop is being tracked by the secondary loop since the tracking data is stored as the manipulated value.

> **Point**
>
> When tracking is required, connect CASOUT_T of the primary loop with CASIN_T of the secondary loop. In addition, settings for the following operation constants of the secondary loop are necessary.
> - PID_TRK
> - PID_SVPTN_B0
> - PID_SVPTN_B1
>
> For details, refer to the following.
> ☞ Page 232 Cascade connection when tracking is required

Ex.

Tracking of cascade loop

In the control loops constituting the cascade loop, SV of the secondary loop is transferred to MV of the primary loop to prevent a sharp change of SV at the control mode switching in the secondary loop.
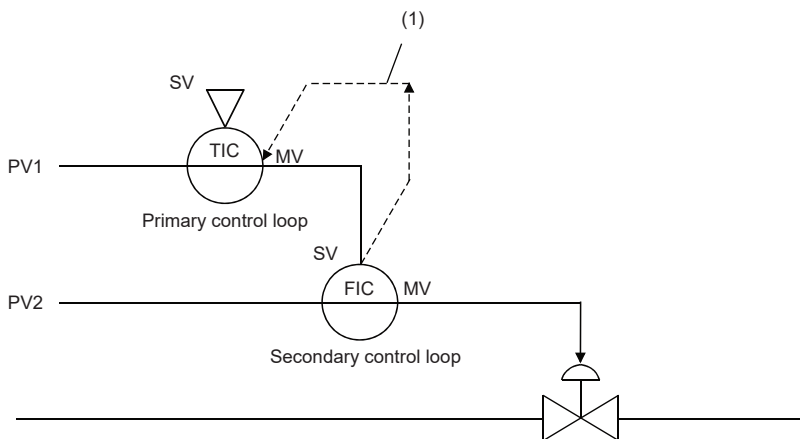
(1) If the control mode of the secondary control loop is other than CASCADE and tracking processing function is valid, SV of the secondary loop is transferred to MV of the primary loop, and tracking processing is performed. When the operation mode of the secondary loop is switched from AUTO to CASCADE, MV sharp change can be prevented since SV of the secondary loop is the same as MV of the primary loop.
The mode change of the secondary loop is: CASCADE ↔ AUTO ↔ MANUAL. The MANUAL mode is switched to CASCADE mode through AUTO mode. The CASCADE mode is switched to MANUAL mode through AUTO mode.
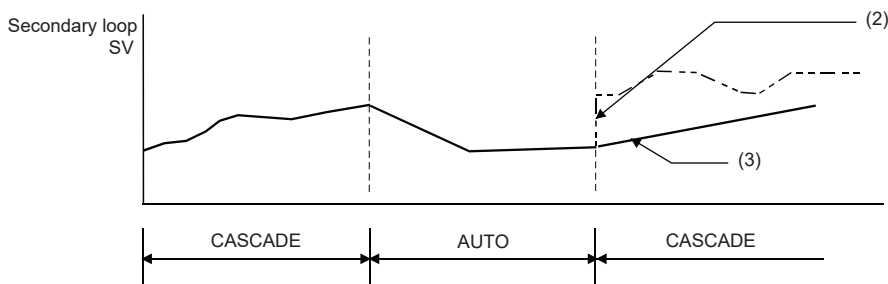
(2) Not tracked: SV of the secondary loop is sharply changed.
(3) Tracked: SV of the secondary loop is not sharply changed.

Tracking of loop selector
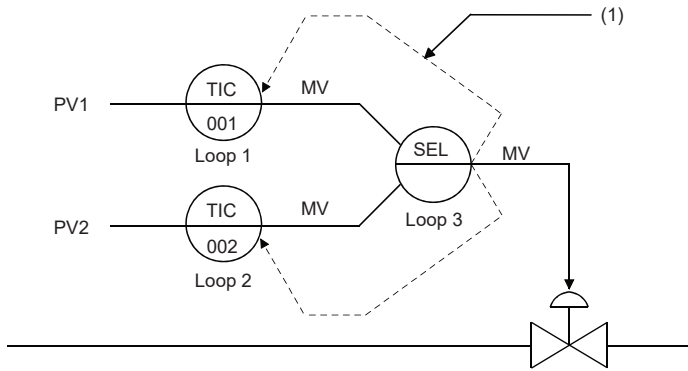
(1) MV of SEL (loop selector) is transferred to the primary loop when the control mode of SEL is manual (MANUAL, COMPUTER MV) or when the variation rate and high/low limiter alarm occurs in other than MANUAL (in AUTO, CASCADE, or COMPUTER SV) mode.
    The primary loop with tracking setting to the SEL is tracked.

■Cascade connection when tracking is required

Connect CASOUT_T of the primary loop with CASIN_T of the secondary loop. The tag FB of the secondary loop should be the one with CASIN_T (such as M+M_PID_T and M+M_2PID_T).
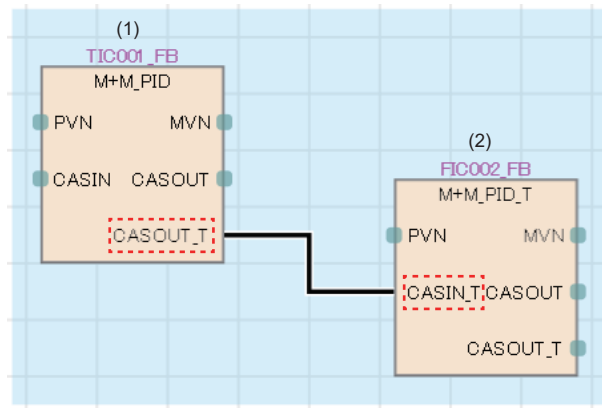
(1) Primary loop
(2) Secondary loop

When tracking is executed, tracking setting for the operation constant of the secondary loop tag FB is required.

Variable name	Description	Range	Initial value	Setting for tracking
PID_TRK	Tracking flag	0: Without tracking 1: With tracking	0	1
PID_SVPTN_B0	Setting value (SV) used	TRUE: Not used FALSE: Used	TRUE	FALSE
PID_SVPTN_B1	Setting value (SV) pattern	TRUE: Not primary MV FALSE: Primary MV	TRUE	When the primary loop is the tag FB: FALSE (typically FALSE) When the primary loop is not the tag FB: TRUE

■Cascade connection when tracking is not required

Connect CASOUT of the primary loop with CASIN of the secondary loop. The tag FB of the secondary loop should be the one with CASIN.
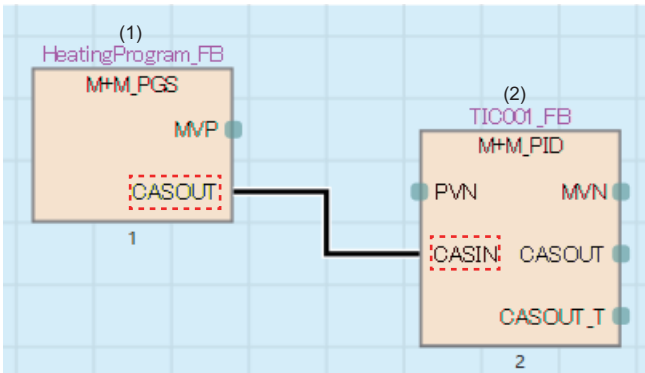
(1) Primary loop
(2) Secondary loop

When tracking is not executed, the operation constant of the secondary loop tag FB is as follows.

Variable name	Description	Range	Initial value	Setting for tracking
PID_SVPTN_B0	Setting value (SV) used	TRUE: Not used FALSE: Used	TRUE	FALSE

A

I/O mode

This section describes the simulation function (SIMULATION mode), override function (OVERRIDE mode), and tag stop function (TAG STOP mode).

Simulation function (SIMULATION mode)

This function performs simulations without actual input/output for I/O modules.

Execute the simulation function (SIMULATION mode) after changing the mode to SIMULATION with the I/O mode change in the faceplate. For details on the I/O mode change, refer to the following.

📖 PX Developer Version 1 Operating Manual (Monitor Tool)

■Function

Item	Description
For the loop tag FB	The loop control is executed by using MV output as feedback input while the PV external input and MV external output are not executed (with the input and output separated from the external). The loop test that is separated from the actual plant can be performed.
For the status tag FB	The input and output are separated from the external and the input signal is substituted a certain period after the output instruction is received. The actual valve ON/OFF commands and response of the answer signal can be simulated to check the control operation.

■Creating simulation data

Item	Description
For the loop tag FB	The simulation data of the loop tag FB can be created by returning the simulation output (SIMOUT) to the simulation input (SIMIN) as variable in the tag FB.
For the status tag FB	Creation of program which returns input to output is not required. Set the simulation answer back signal period (SIMT) of tag data.

■Sample programs of the loop tag FB simulation function (SIMULATION mode)

The following shows the sample programs for the simulation function (SIMULATION mode) when the loop tag FB is used.

Ex.

For the typical loop with the same range for PVN and MVN
- Program example for the simulation function (SIMULATION mode)
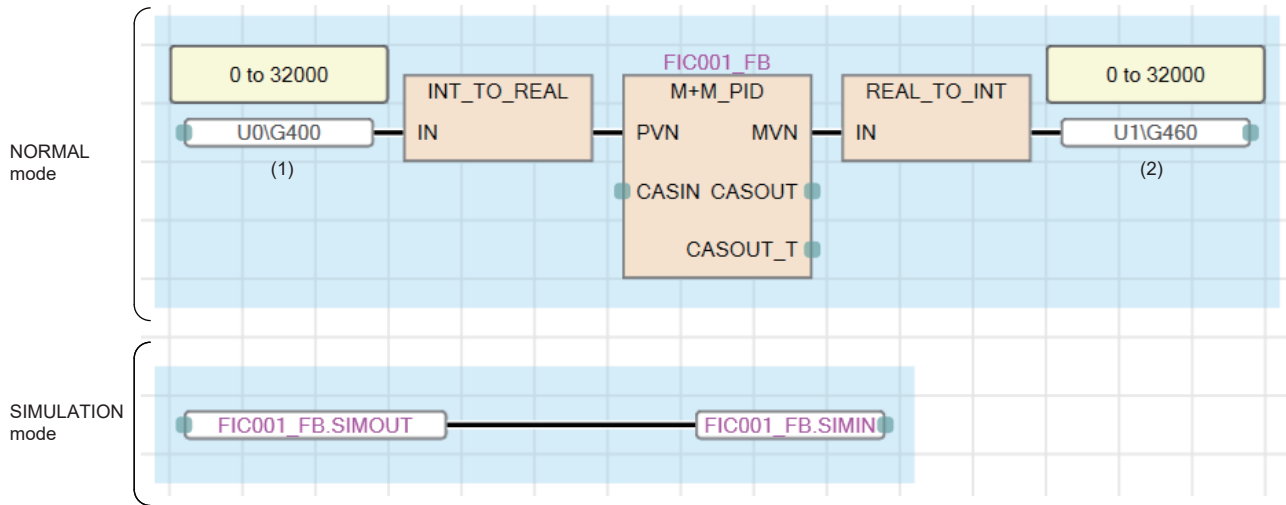
(1) Analog input module CH1 digital output value
(2) Analog output module CH1 digital value

- Program example for the simulation function (SIMULATION mode) (with Primary delay + Dead time)
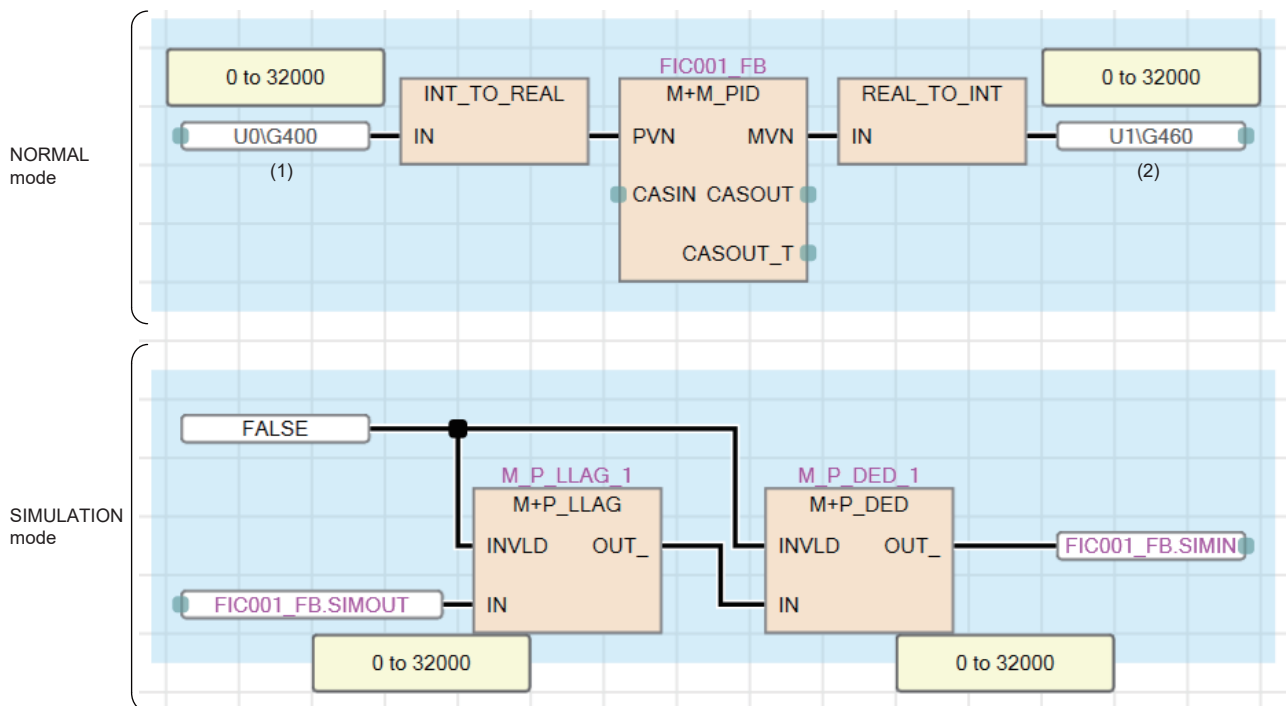
(1) Analog input module CH1 digital output value
(2) Analog output module CH1 digital value

For the cascade connection

• Program example for the simulation function (SIMULATION mode)

NORMAL mode

0 to 32000		LIC001_FB
U0\G400	INT_TO_REAL	M+M_PID
(1)	IN	PVN MVN
		CASIN CASOUT
		CASOUT_T

FIC002_FB
M+M_PID_T

U0\G600	INT_TO_REAL	PVN MVN	REAL_TO_INT	0 to 32000
(2)	IN	CASIN_T CASOUT	IN	U1\G460
		CASOUT_T		(3)

SIMULATION mode

FIC002_FB.SIMIN —————————— LIC001_FB.SIMIN

FIC002_FB.SIMOUT —————————— FIC002_FB.SIMIN

(1) Analog input module CH1 digital output value
(2) Analog input module CH2 digital output value
(3) Analog output module CH1 digital value

• Program example for the simulation function (SIMULATION mode) (with Primary delay + Dead time)

NORMAL mode

| 0 to 32000 |
| U0\G400 |
(1)

INT_TO_REAL
IN

LIC001_FB
M+M_PID
PVN   MVN
CASIN CASOUT
CASOUT_T

| U0\G600 |
(2)

INT_TO_REAL
IN

FIC002_FB
M+M_PID_T
PVN   MVN
CASIN_T   CASOUT
CASOUT_T

REAL_TO_INT
IN

| 0 to 32000 |
| U1\G460 |
(3)

SIMULATION mode

FALSE

M_P_LLAG_1
M+P_LLAG
INVLD   OUT_
IN

M_P_DED_1
M+P_DED
INVLD   OUT_
IN

LIC001_FB.SIMIN

FIC002_FB.SIMIN

FALSE

M_P_LLAG_2
M+P_LLAG
INVLD   OUT_
IN

M_P_DED_2
M+P_DED
INVLD   OUT_
IN

FIC002_FB.SIMIN

FIC002_FB.SIMOUT

| 0 to 32000 |

| 0 to 32000 |

(1) Analog input module CH1 digital output value
(2) Analog input module CH2 digital output value
(3) Analog output module CH1 digital value

Point

For the cascade connection, the simulation loopback input data of the secondary loop (FIC002.SIMIN in the example above) is used as the simulation input data of primary loop (LIC001.SIMIN in the example above).

A

Ex.

When the ranges of PVN and MVN are different (the range of A/D conversion module (AIN_4CH) is 0 to 32000, and the range of D/A conversion module (AOUT_4CH) is -32000 to 32000)

• Program example for the simulation function (SIMULATION mode)

M+P_ENG and M+P_IENG convert FIC001_FB.SIMOUT into FIC001_FB.SIMIN for connection.
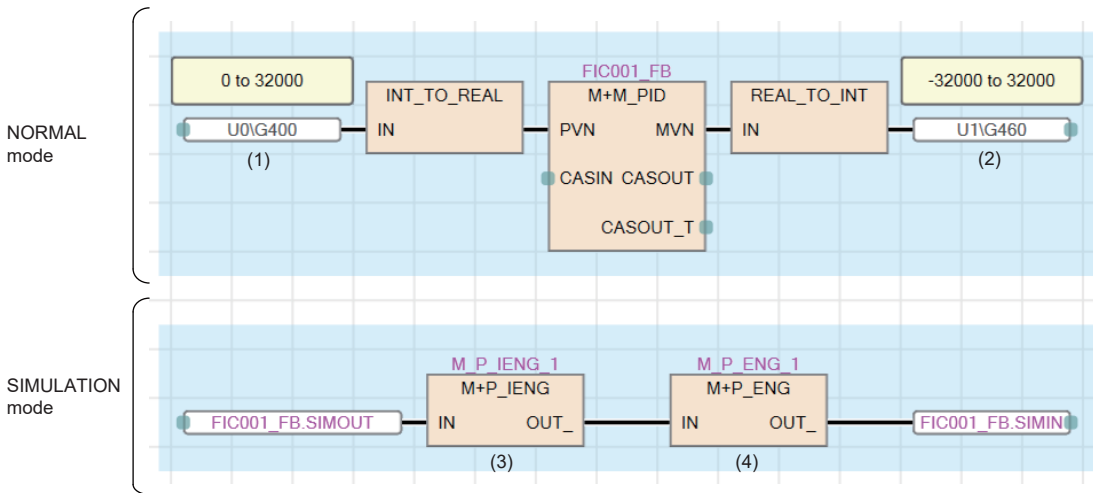
(1) Analog input module CH1 digital output value
(2) Analog output module CH1 digital value
(3) The engineering value inverse conversion of 0 to 32000/0 to 100 (%)
(4) The engineering value conversion of 0 to 100 (%)/-32000 to 32000

• Program example for the simulation function (SIMULATION mode) (with Primary delay + Dead time)

M+P_ENG and M+P_IENG convert FIC001_FB.SIMOUT into FIC001_FB.SIMIN for connection.
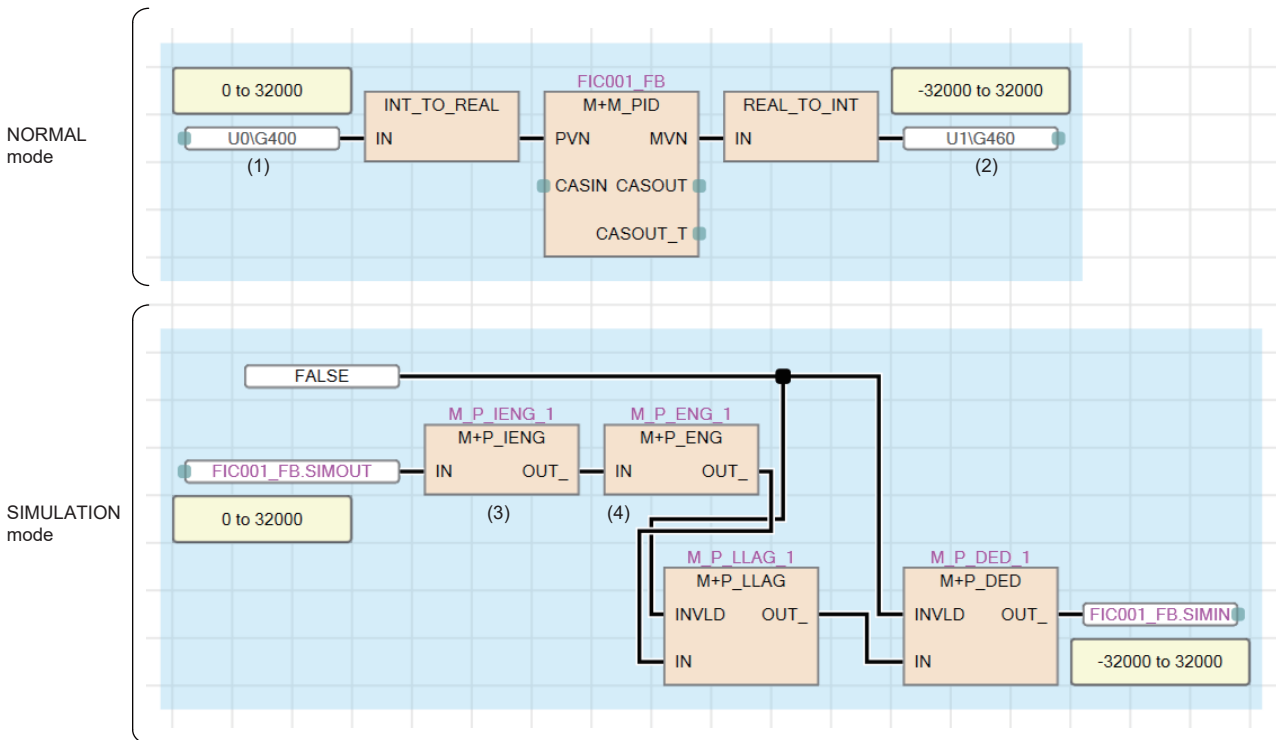
(1) Analog input module CH1 digital output value
(2) Analog output module CH1 digital value
(3) The engineering value inverse conversion of 0 to 32000/0 to 100 (%)
(4) The engineering value conversion of 0 to 100 (%)/-32000 to 32000

Override function (OVERRIDE mode)

This function enables the PV setting on the pop-up tuning window of the PX Developer monitor tool when a correct input signal cannot be acquired due to a failure of the sensor, limit switch or A/D conversion module.

External outputs of the manual manipulated value and ON/OFF signal are performed.

Execute the override function after changing the mode to OVERRIDE with the I/O mode change in the faceplate. For details on the I/O mode change, refer to the following.

📖 PX Developer Version 1 Operating Manual (Monitor Tool)

■Function

Item	Description
For the loop tag FB	This function enables the PV setting on the pop-up tuning window of the PX Developer monitor tool when a correct PV input signal cannot be acquired due to a failure of the sensor or A/D conversion module. However, external output is performed. In this case, MV is output in MANUAL mode. The function is used when the input signal is used for inter-lock conditions or transition conditions of the batch sequence. For the PV setting, input the value to the tag monitor field on the pop-up tuning window of the PX Developer monitor tool.
For the status tag FB	This function enables to set input signals on the pop-up tuning window of the PX Developer monitor tool when a correct input signal cannot be acquired due to a poor contact of the valve open/close limit switch. However, external output is performed. The function is used when the input signal is used for inter-lock conditions or transition conditions of the batch sequence.

> **Point**
>
> For the loop tag FB, the override function can be operated in MANUAL mode only.
>
> Operations in other than MANUAL mode cannot be operated.

Tag stop function (TAG STOP mode)

This function stops input processing and loop control operation. Set it for the tag reserved for future use.

The tag stop function can be set with the I/O mode change in the faceplate. For details on the I/O mode change, refer to the following.

📖 PX Developer Version 1 Operating Manual (Monitor Tool)

■Function

The tag stop function can be used for 2PIDH, PGS2, SWM, PVAL, and HTCL tags.

This function is used for the tag that has been defined for future use with the engineering tool or has stopped.

For details, refer to the description of each function block.

A

Stop alarm processing in loop control

Overview of stop alarm (SPA)

When an alarm related to the loop tag data such as a sensor error (SEA) occurs, the control mode can be changed to MANUAL forcibly by switching the stop alarm (SPA) from FALSE to TRUE.

Switch the stop alarm (SPA) from FALSE to TRUE using a user program as necessary.

When the stop alarm (SPA) is switched to TRUE, alarms which have occurred (MLA, MHA, DVLA, DPPA, PLA, PHA, LLA, HHA, SEA, and DMLA) will be automatically reset (TRUE → FALSE).

Alarms (ALM) of loop tag data

The following table lists alarms (ALM) of loop tag data.

Offset		Label	Application
+3	b0	MLA	Output Low Limit Alarm
	b1	MHA	Output High Limit Alarm
	b2	DVLA	Large Deviation Alarm
	b3	DPNA	Negative Variation Rate Alarm
	b4	DPPA	Positive Variation Rate Alarm
	b5	PLA	Input Low Limit Alarm
	b6	PHA	Input High Limit Alarm
	b7	LLA	Input Low Low Limit Alarm
	b8	HHA	Input High High Limit Alarm
	b9	SEA	Sensor Error Alarm
	bA	OOA	Output Open Alarm
	bB	DMLA	Output Variation Rate Limit Alarm
	bE	SPA	Stop Alarm

An alarm (ALM) consists of multiple Boolean variables. Boolean variables constituting an alarm (ALM) become global variables.

When using the stop alarm in a program, refer to the following example.

Ex.
When a loop sensor error (SEA) of FIC001 occurs
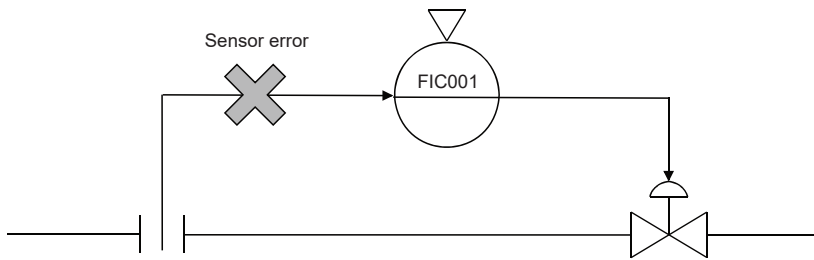
In the example above, the sensor error (SEA) can be acquired with the tag data label of the variable element.
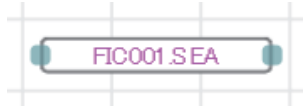
Relation between stop alarm (SPA) and other alarm processing

The following table shows the relation between the status of the stop alarm (SPA) and other loop alarm processing.

Stop alarm (SPA)	Result
TRUE	Alarms processing other than stop alarm (MLA, MHA, DVLA, DPPA, PLA, PHA, LLA, HHA, SEA, DMLA) are stopped, and then the alarm is automatically switched to FALSE.
FALSE	Alarm processing other than stop alarm (MLA, MHA, DVLA, DPPA, PLA, PHA, LLA, HHA, SEA, DMLA) is executed.
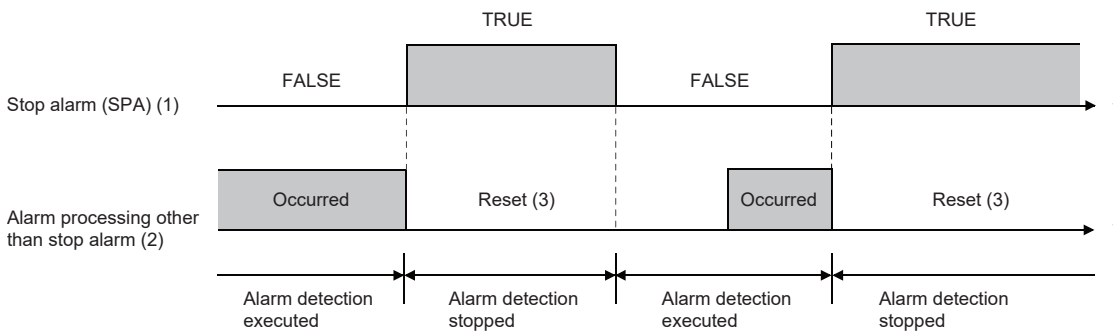
(1) Switch the stop alarm (SPA) to TRUE and FALSE using a program.
(2) The output open alarm (OOA) follows the processing in the program.
(3) When the stop alarm (SPA) is TRUE, the alarm detection processing is stopped. Alarms which have occurred are automatically reset.

How to use the output open alarm

The output open alarm (OOA) of loop tag data is designed for displaying the wire break detection signal as an alarm on the loop tag FB of the output source when a disconnection is detected on the module FB on the output side.

The following shows a program example to feed back a signal from the module FB that is to detect a disconnection to the loop tag FB that is to display it as an alarm.

Program example

Input the module access device on the output side (CH☐ disconnection detection flag) to the tag item (OOA) of the loop tag FB.

For the output open alarm (OOA), the disable alarm detection flag does not exist. Therefore, create a program separately to enable or disable this alarm.
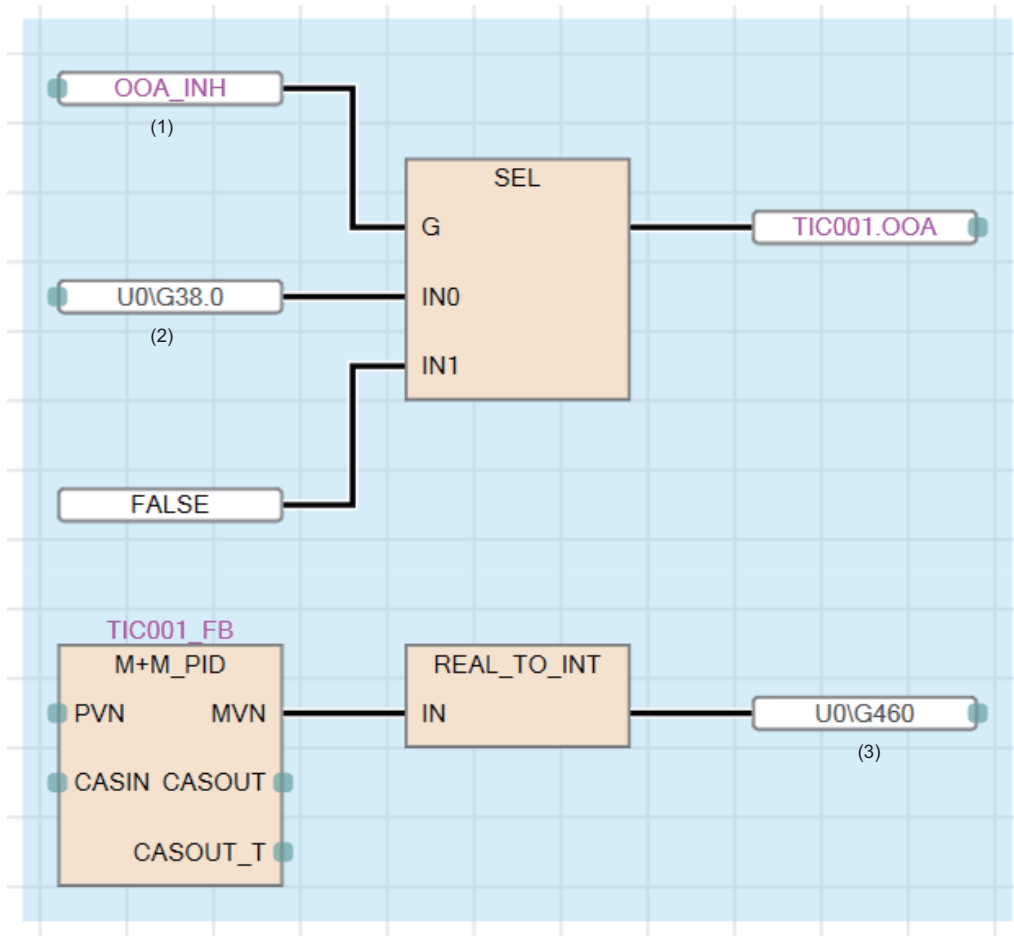
(1) User-created variable
(2) CH1 disconnection detection flag
(3) CH1 digital value

Program setter setting method

This section describes the setting method of the program setter (PGS) and multi-point program setter (PGS2).

Program setter (PGS)

■Operation method

The X-Y graph method is used. The output MV is calculated by using the X-Y graph function according to the SV (time) of the whole program.

■Relation between the program setting method and each variable

Register a program using the X-Y graph method shown below.
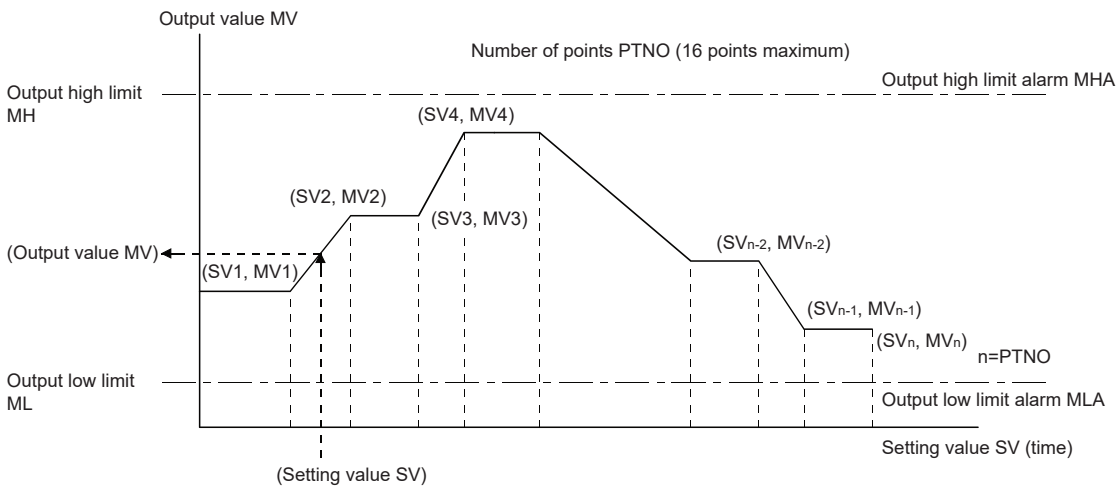
■Registration format

Register the break point data up to 16 points in real number (REAL).

■Time management

Set the time in units of seconds.

■Output high/low limit alarm

Place the output low limit alarm MLA and output high limit alarm MHA to bit 0 and bit 1 (standard locations) of the tag data +3 (ALM).

■Mode and operation type

Five control modes are available: MANUAL, COMPUTER MV, AUTO, CASCADE, and COMPUTER SV.

In CASCADE mode, the operation type is CYCLIC.

In AUTO mode, the operation type HOLD or RETURN can be selected.

Multi-point program setter (PGS2)

■Operation method

Steps (time width and setting value) are registered and the progress of each step is managed.

The setting value (SV) is calculated according to the time in the step (T) in each step.

■Relation between the program setting method and each variable

Register each step data as shown below. Set the program start point to SV0.
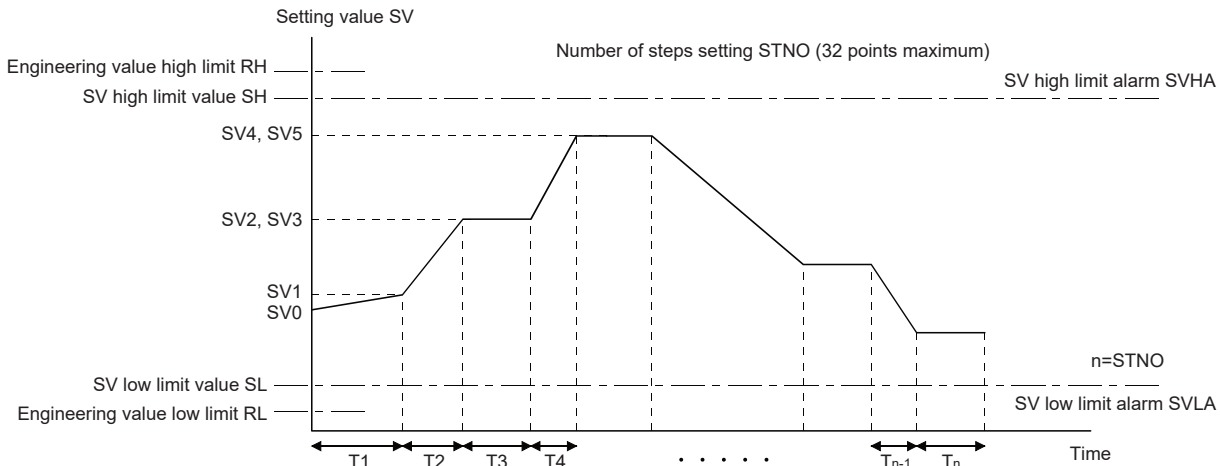
Tn: Time in the step (T)

n: Executing step No. (STC)

The following main parameters are changed from the program setter (PGS).

• MV (Output value) → SV (Setting value)

• SV (Setting value) → STC (Executing step number) + T (Time in the step)

■Registration format

Register the step data up to 32 points integer (INT).

The setting range is from -32768 to 32767.

■Time management

Set the time in units of seconds or minutes. (Set at TUNIT in the tag data.)

■Output high/low limit alarm

Place the SV low limit alarm SVLA and SV high limit alarm SVHA to bit 0 and bit 1 of the tag data +3 (ALM).

(Variable names differ from those for the program setter (PGS).)

■Mode and operation type

Two control modes are available: MANUAL and AUTO.

In AUTO mode, the operation type HOLD, RETURN, or CYCLIC can be selected.

Tight shut/full open function

To use the tight shut/full open function with a module that has the extended mode in the range setting, set the extended mode in the output range setting.

To use the tight shut/full open function with a module that does not have the extended mode in the range setting, set 0 to 20mA and 0 to 5V in the range setting and set the output conversion high/low limit values of the two-degree-of-freedom advanced PID control FB again.

> **Point**
>
> When 4 to 20mA and 1 to 5V are regarded as a standard, only the tight shut (full open when MV reverse is valid) is applied since the upper limit side can output up to 100% (20mA, 5V) with output range setting of 0 to 20mA and 0 to 5V.

The following table shows a setting example for the two-degree-of-freedom advanced PID control FB when the signal of 4 to 20mA is regarded as a standard, tight shut output value is set to -20% (0.8mA).

For the analog module, set 0 to 20mA for the output range setting and 0 to 32000 for the resolution.

Item	Setting value	Remarks
Output Conversion High Limit (OUT3_NMAX)	32000.0	Equivalent to 20mA
Output Conversion Low Limit (OUT3_NMIN)	6400.0	Equivalent to 4mA
Tight Shut/Full Open Execution condition (OUT3_FOTS_EN)	TRUE	—
Output Value for Full Open (Unit: %) (OUT3_MVFO)	100.0	100%, since 20mA or more cannot be outputted.
Output Value for Tight Shut (Unit: %) (OUT3_MVTS)	-20.0	Equivalent to 0.8mA

Appendix 5 Simulation Function

The Simulation function is a function of GX Works3 and debugs programs using a virtual programmable controller on a personal computer.

This section describes the procedure to operate the process control program "School_Master" on the simulator and check the PID control of TIC001.

• Process control program to use
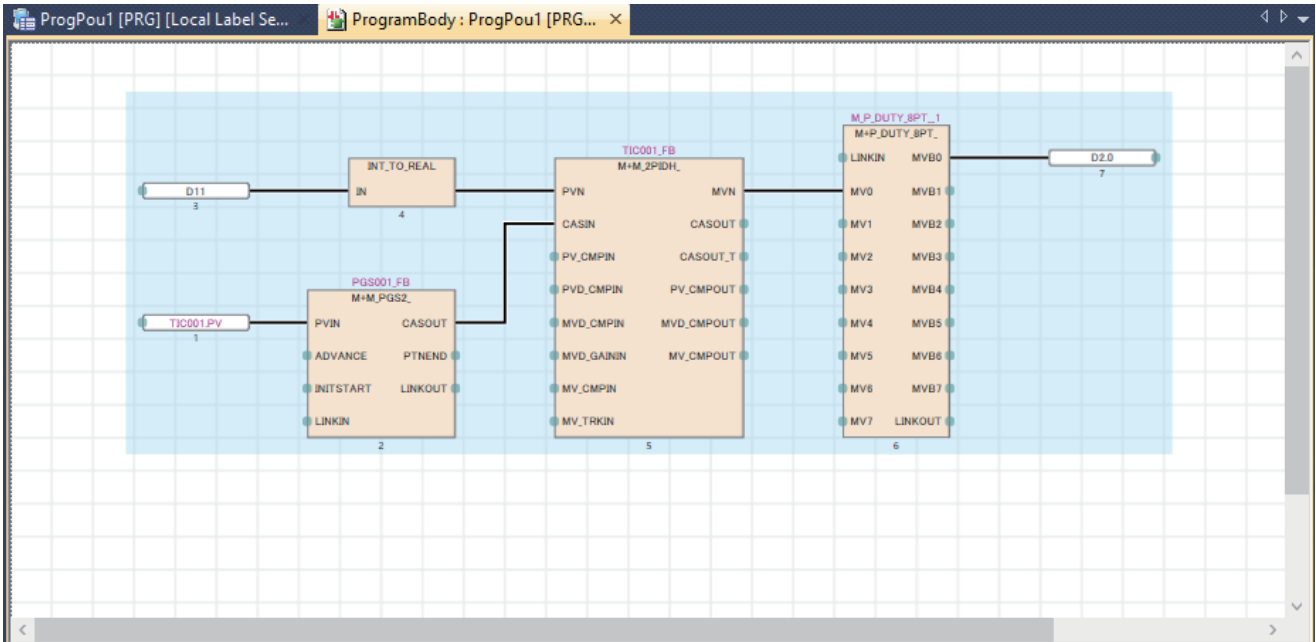
• Devices to use in the program

Device	Description
D2.0	Output: Heating output
D11	Input: Measured temperature value (INT data type)
D12	Simulation area (REAL data type)

Operating procedure

1. Select "Rebuild All"
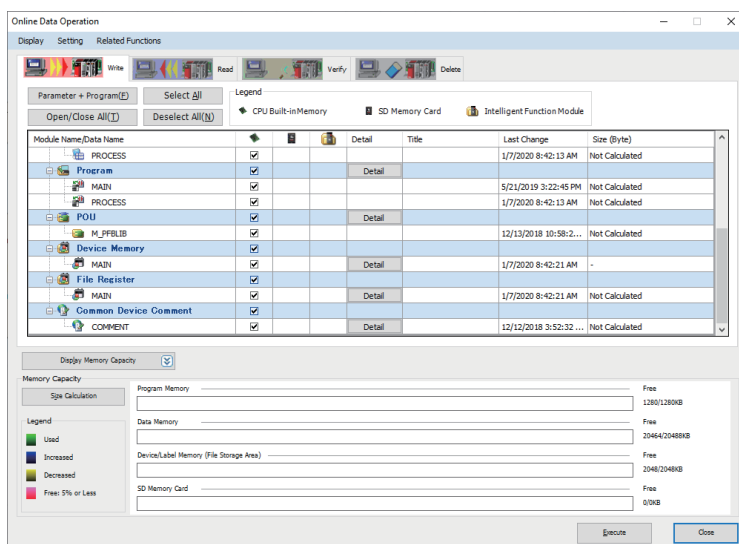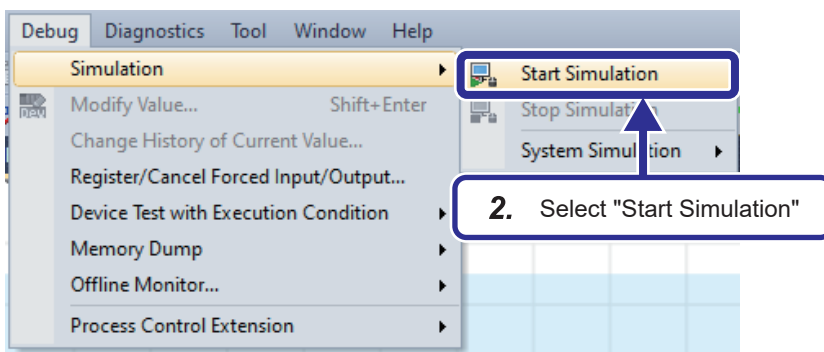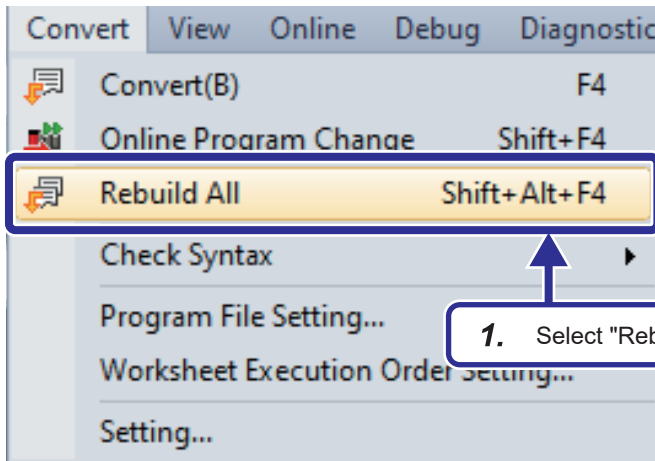
1. Open "School_Master". Click [Convert] on the menu bar of the engineering tool, select [Rebuild All], and rebuild (reassign) all programs following the procedure described in Page 106 Converting the program.

2. Select "Start Simulation"

2. Click [Debug] on the menu bar of the engineering tool and select [Simulation] ⇨ [Start Simulation] to start GX Simulator3.

3. GX Simulator3 and the "Online Data Operation" window appears. Write the project to the simulator following the procedure described in Page 108 Writing data to the programmable controller.

A
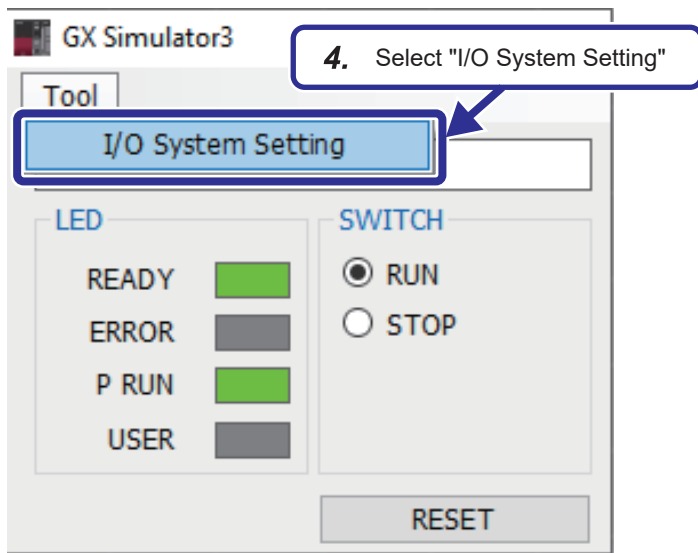
4. Check the PID control of TIC001 on the simulator by using the virtual I/O of GX Simulator3.
Click [Tool] and select [I/O System Setting].
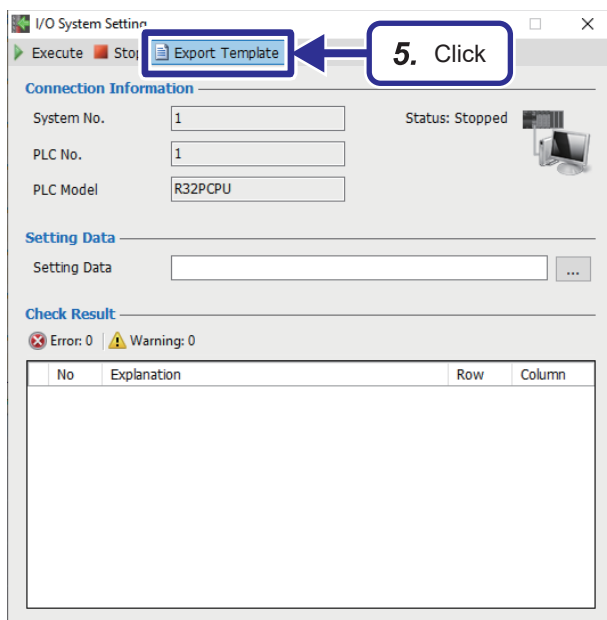
5. The "I/O System Setting" window appears. Click the [Export Template] button.
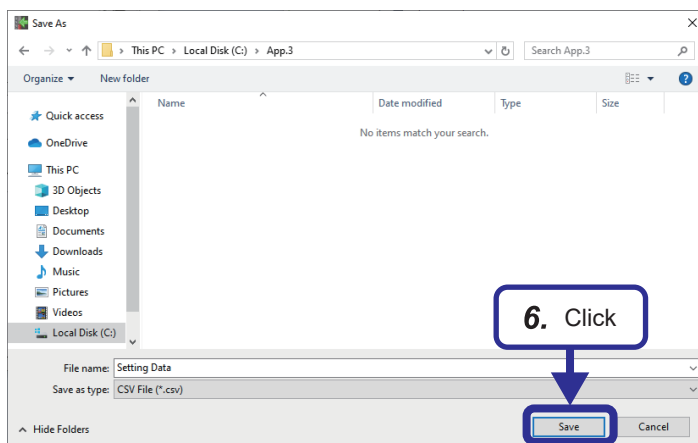
6. Specify a save destination and enter a file name.

7. Configure the I/O system settings in the exported template (CSV file format). Edit the settings as follows so that TIC001 adds the primary delay and dead time to the MV value and returns a value to D11.

[Control record settings]

Record Type	GX Works3 Project Path
1	Path to save the project

[Data record settings]

Setting No.1: Adds the delay and dead time to TIC001.MV, performs the range conversion, and writes a value to D12 (REAL data type) continuously.

Setting No.2: Writes a value in D12 (REAL data type) to D11 (INT data type) continuously.

Record Type	Setting No.	Conditions	Timer (ms)	Output
2	1	SM400=TRUE	*	D12:E:=LAG_DED(0,1000,0,100,1,0,TIC001.MV)
2	2	SM400=TRUE	0	D11:=REAL_TO_INT(D12:E)

8. Save the CSV file.

9. Click the [...] button of "Setting Data" and set the CSV file created in Step 7 of this procedure.

10. Click the [Execute] button.
If an error is detected, the error description will be displayed under "Check Result". Correct the I/O system settings and check the setting data again.
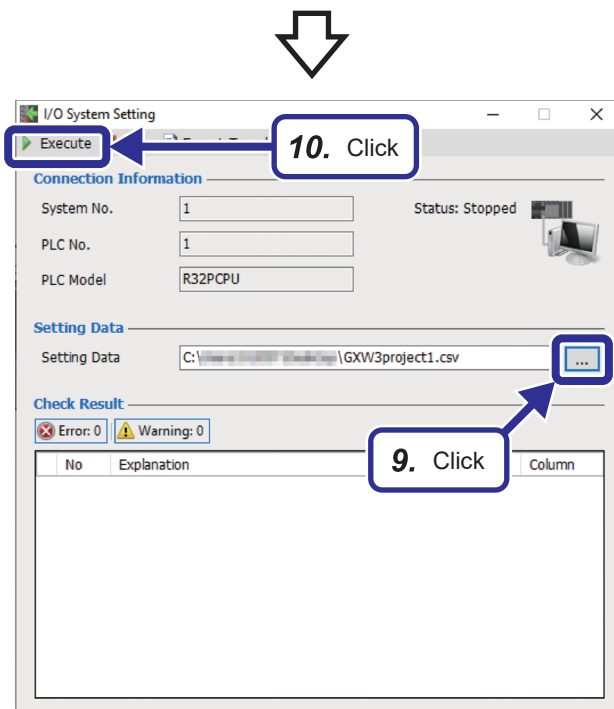
A
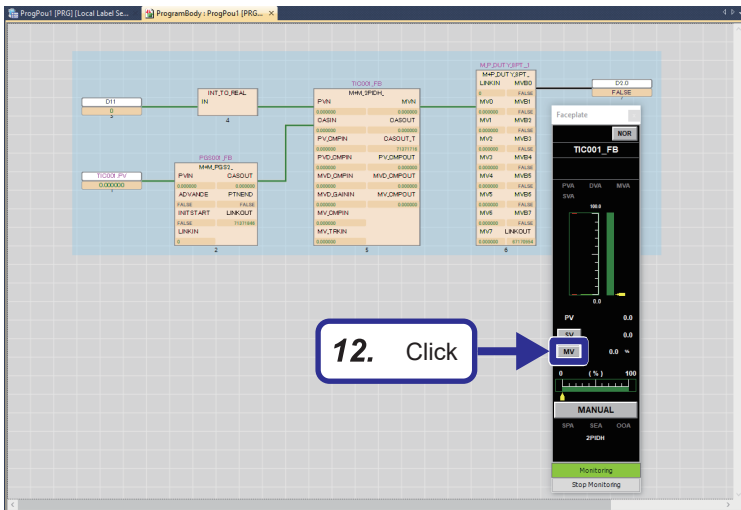
11. Check the operation of TIC001 with the I/O system settings.
Follow the procedure in Page 111 Monitoring using the faceplate to display the faceplate.

12. Click the [MV] button.
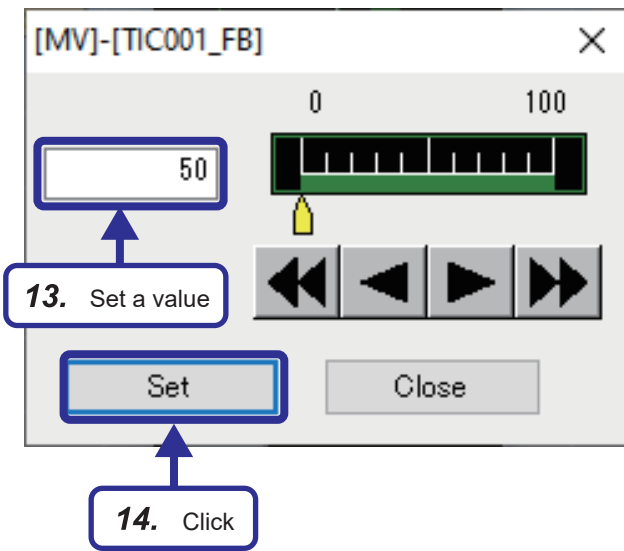
13. The MV window appears. Set a value for the MV.

14. Click the [Set] button.
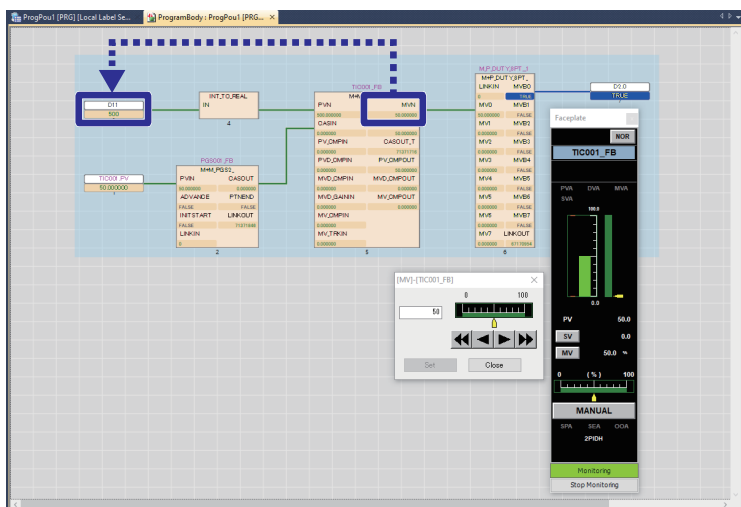
15. Check that the MV value with the primary delay and dead time of TIC001 is returned to D11.
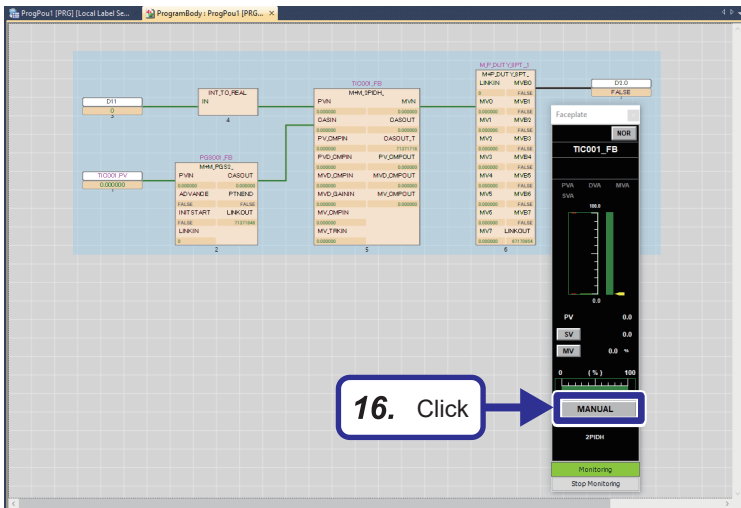
16. Check the operation of PID control of TIC001.
Click the [MANUAL] button in the faceplate.

16. Click
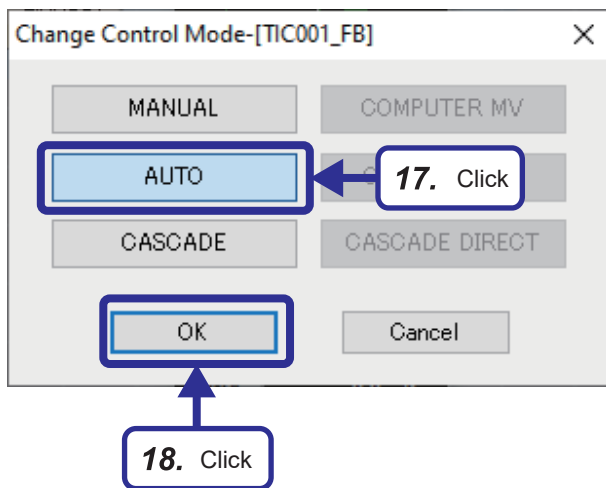
17. Click the [AUTO] button.

18. Click the [OK] button.

17. Click

18. Click
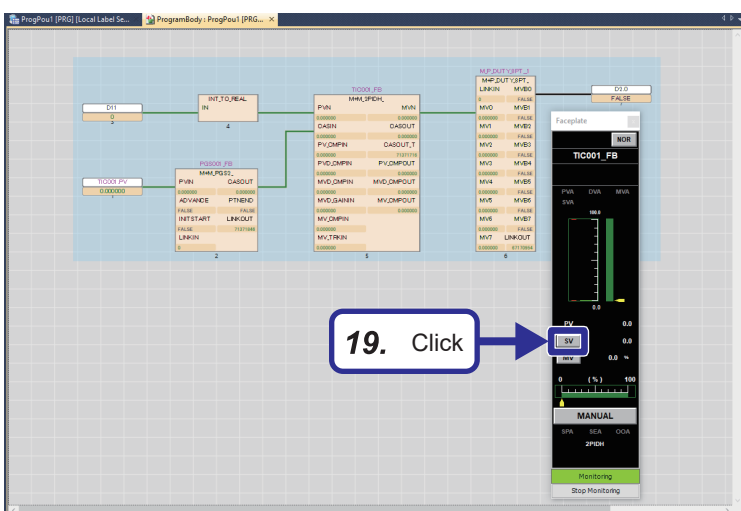
19. Click the [SV] button.

19. Click

A
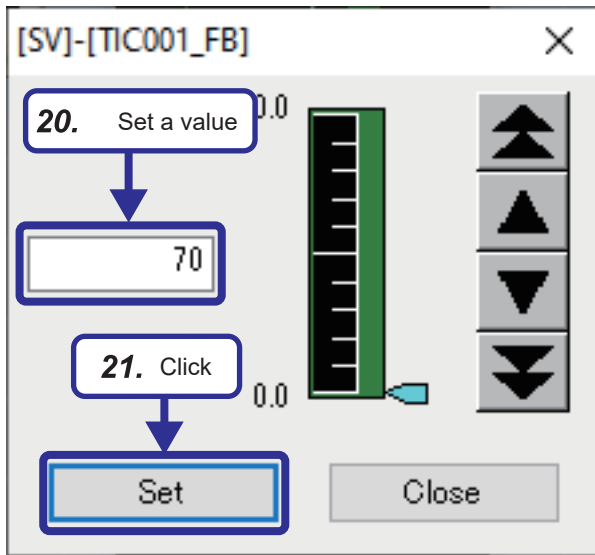
20. The SV window appears. Set a value for the SV.

21. Click the [Set] button.
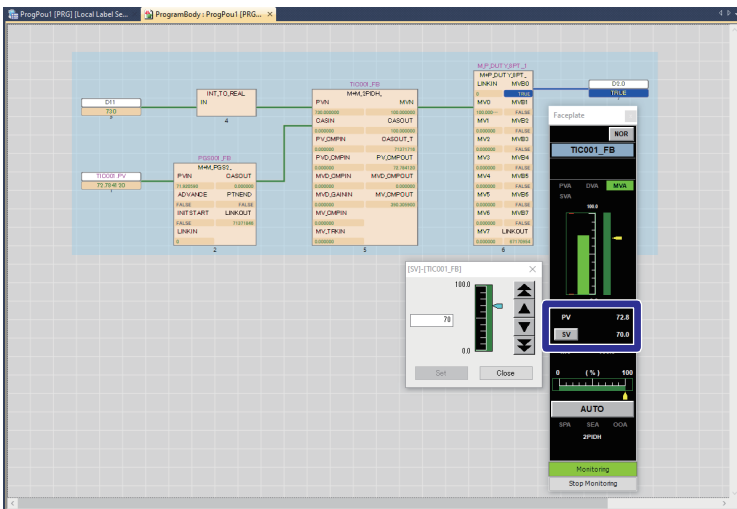
22. Check that the PV value is kept close to the SV value by PID control.

Appendix 6 Creating Redundant System Projects

This section describes the procedure for creating the projects "School_Master_base" and "School_Remote".

Creating a project (master station)

This section describes how to create a new master station project using GX Works3.

Creating a new project

Create the master station project "School_Master_base".

Operating procedure
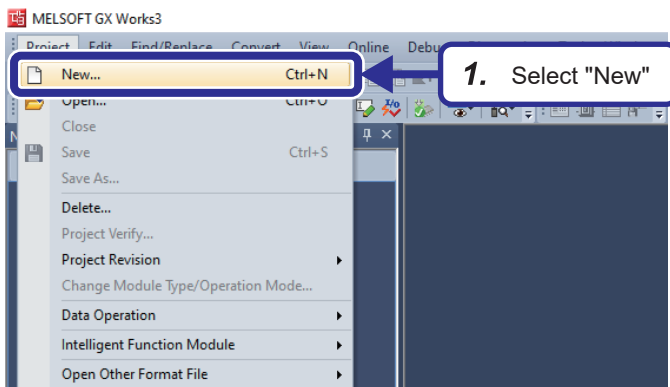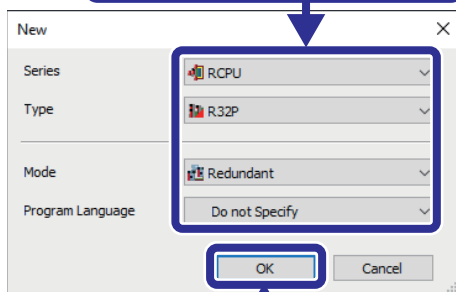
1. Select [Project] ⇨ [New].

2. When the New window is displayed, set the CPU model, operation mode, and program language as shown in the left figure.

3. Click the [OK] button.
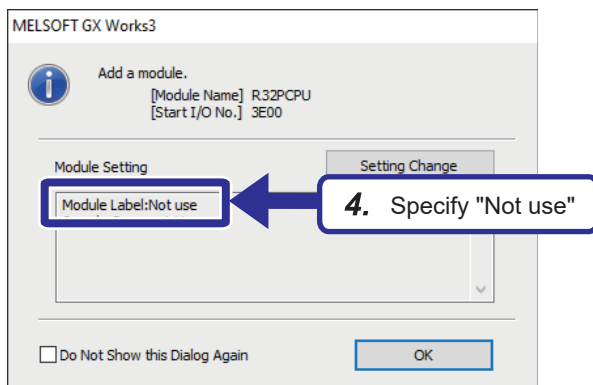
4. Set the R32PCPU module label to "Not use".

System A/B settings

Define system A and system B using GX Works3.

Operating procedure

2. Select "Current Connection Destination"

3. Set these sections

3. Set this section
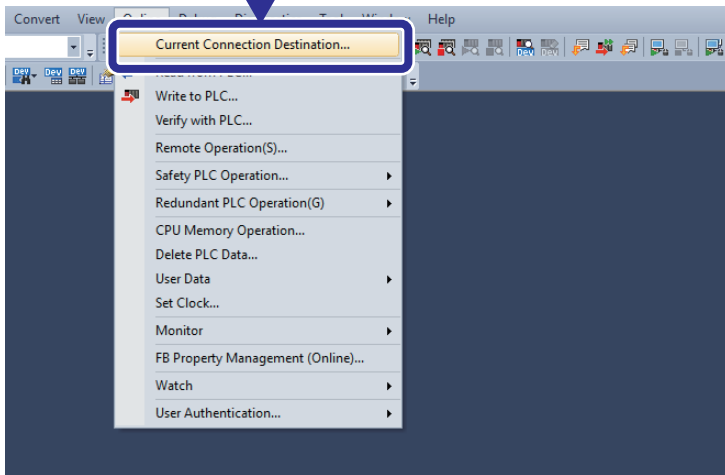
1. Connect a USB cable to the Process CPU to be set for system A.

2. Select [Online] ⇨ [Current Connection Destination] to open the "Specify Connection Destination" window.
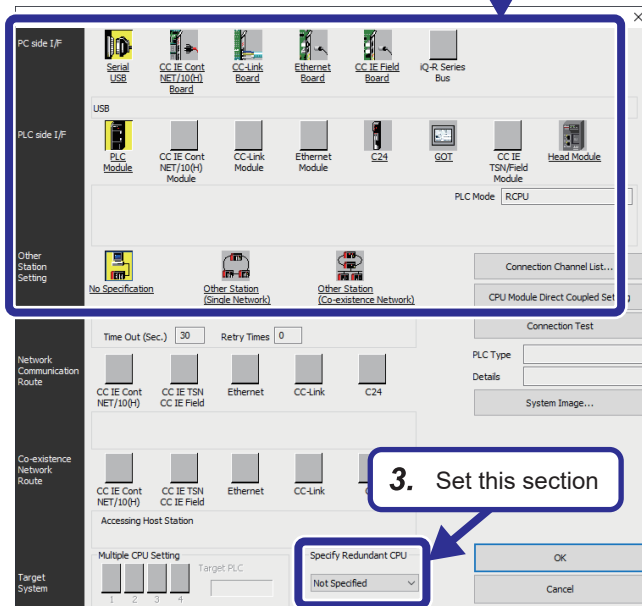
3. Set the settings as follows.
[Settings]
PC side I/F: Serial USB
PLC side I/F: PLC Module
Other Station Setting: No Specification
Specify Redundant CPU: Not Specified
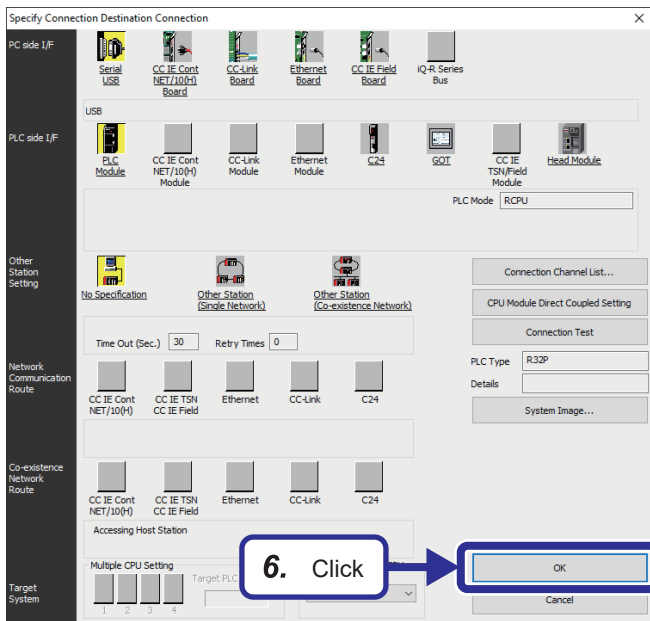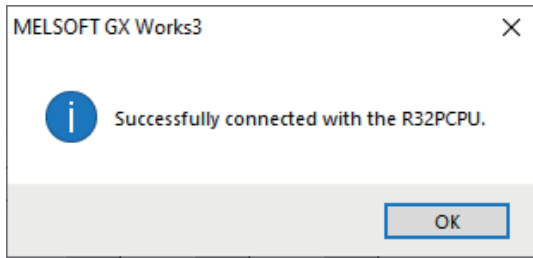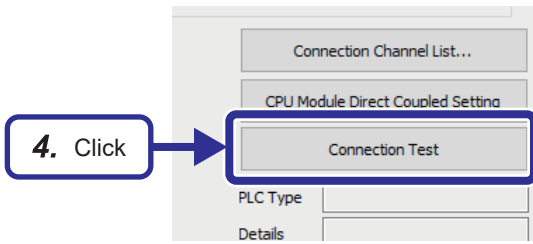
4. Click the [Connection Test] button.

5. Check that the CPU module is connected successfully.

6. Click the [OK] button.
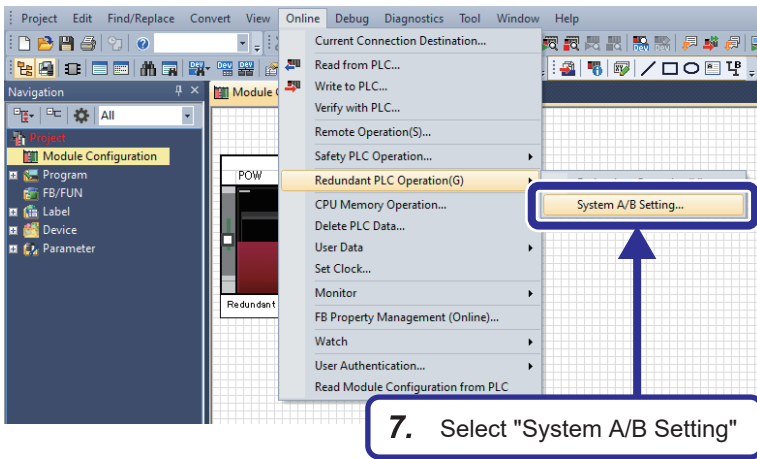
7. Select [Online] ⇨ [Redundant PLC Operation] ⇨ [System A/B Setting].

7. Select "System A/B Setting"

8. Specify "System A"
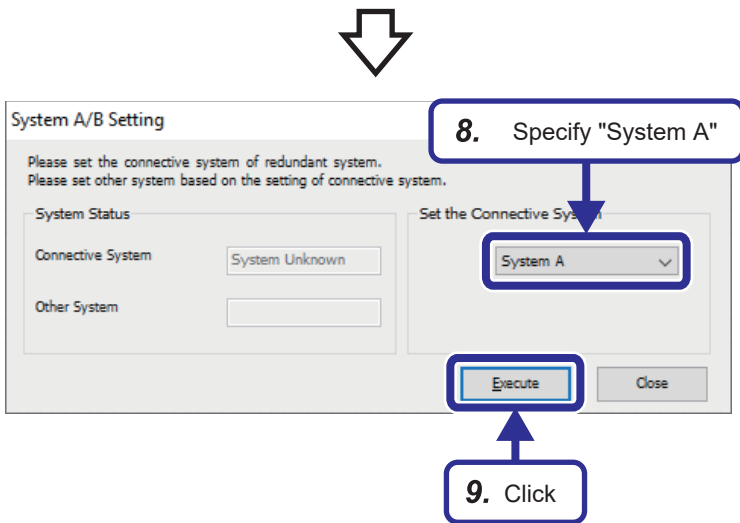
8. Select either system A or system B as the primary connection system. System A is selected in this example.
When system A is selected as the primary connection system, system B automatically becomes the secondary connection system.

9. Click the [Execute] button.

9. Click

10. Click the [Yes] button.

10. Click

APPX
Appendix 6 Creating Redundant System Projects

MELSOFT GX Works3

System A/B Setting is completed.
Please turn off and turn it back or reset the CPU module in
which System A/B Setting is changed.

OK
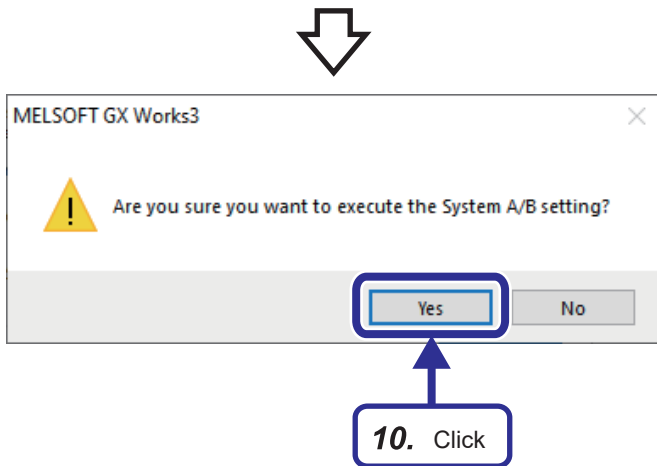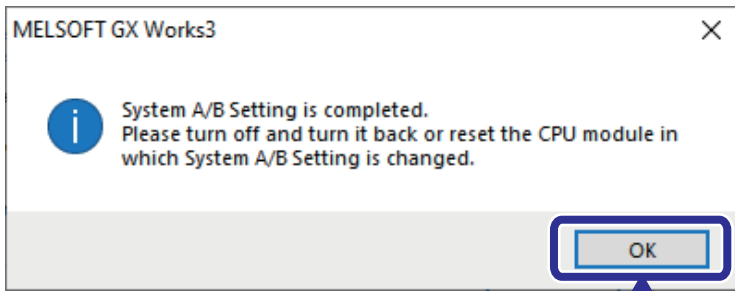
11. Click

11. When the window shown on the left is
displayed, click the [OK] button.

12. When the System A/B Setting window is
displayed again, click the [Close] button.

13. Set the RUN/STOP/RESET switch of the
CPU module to the "RESET" position once
(for approximately 1 second) to reset.[1]

*1 If the system A/B settings are set to the Process CPU that is not running in the redundant mode, powering off and on or resetting the
CPU module turns off the SYS A LED or SYS B LED of the redundant function module. It will turn on after the project is written.

A

■**Checking method**

Check the LED of each redundant function module to check the system status.

Setting on the engineering tool	LED of the redundant function module
System A	SYS [A■ / B]
System B	SYS [A / B■]
When the system setting is switched from system A to B	SYS [A■ / B] ➡ SYS [A■ / B■] ➡ SYS [A / B■] When the system setting is switched to the system B with the engineering tool, the SYS B LED will flash. Powering off and on or resetting the CPU module turns off the SYS A LED and turns on the SYS B LED.

> **Point**
>
> Users can check the system A/B setting with the engineering tool. (📖 GX Works3 Operating Manual)
> - System monitor
> - "System A/B Setting" window
> - Monitor status bar

Precautions

- Do not power off or reset the CPU module during system A/B setting. Doing so may not properly reflect the system A/B setting. If the system A/B setting has not been properly reflected, set the system again.
- For the system A/B setting, use the CPU module that has been connected with a USB cable or an Ethernet cable. Users cannot set a system if they have changed the connection destination on the engineering tool.
- The set system A/B information cannot be deleted. Only changing the set system A/B information is supported.
- When starting up both systems, if both systems have no system A/B setting yet or the same system A/B setting, a stop error will occur when tracking communications are established. Set a different system A/B setting for each system.
- If the system that has no system A/B setting yet is started first, a stop error will occur when tracking communications are established. Set a different system A/B setting for each system.
- Do not connect system A and system A or system B and system B with tracking cables while running. If connected, a continuation error occurs.

> **Point**
>
> Besides the method that uses the engineering tool for setting system A and B, system A and B can be automatically set by using the systems.
> When one system is set to system A and gets ready for tracking communications, the other system is automatically set to system B, and vice versa.
> For the system to which system A or system B has been assigned automatically, the system setting will be overwritten automatically. In that case, the system setting change can be checked in the event history. (Event code: 00700)
> - If a system that has no system A/B setting yet is connected to the system specified as system A, the system that has no system A/B setting will be automatically set to system B. (This event is not recorded in the event history.)
> - If a system specified as system A is connected to another system specified as system A, the system where the CPU module is turned off and on or reset will be automatically set to system B. (This event is recorded in the event history.)

Setting the module configuration window

Set the module configuration diagram in GX Works3 and define the master station system configuration.
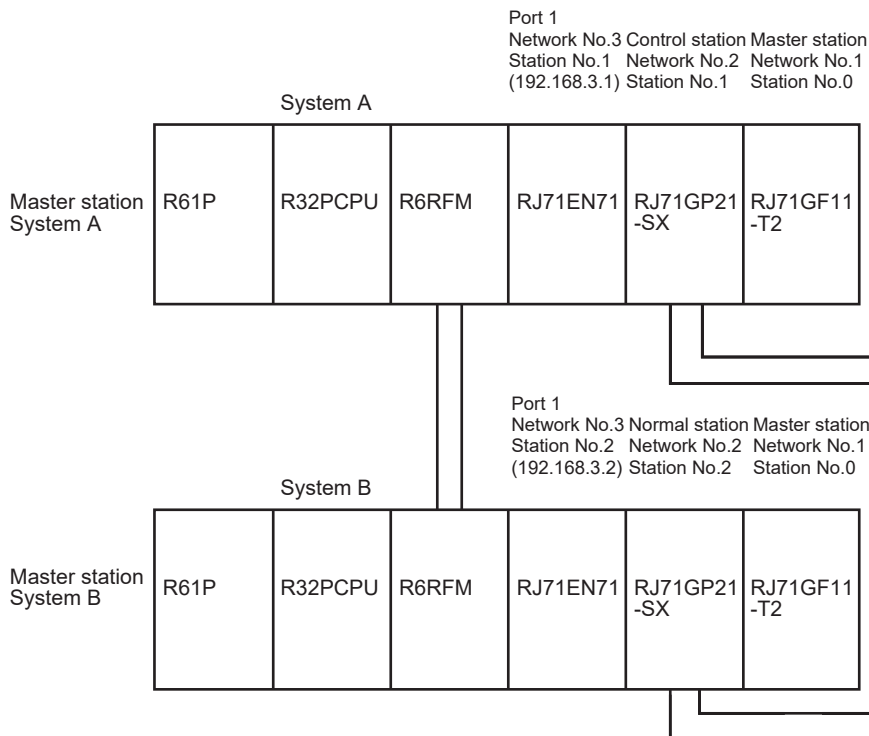
Operating procedure

1. In the Navigation window, double-click "Module Configuration".
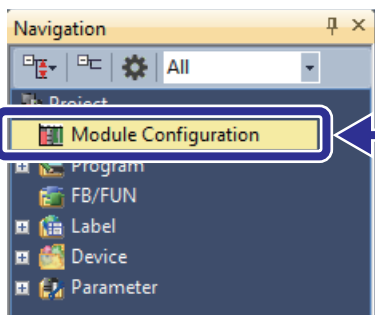
1. Double-click
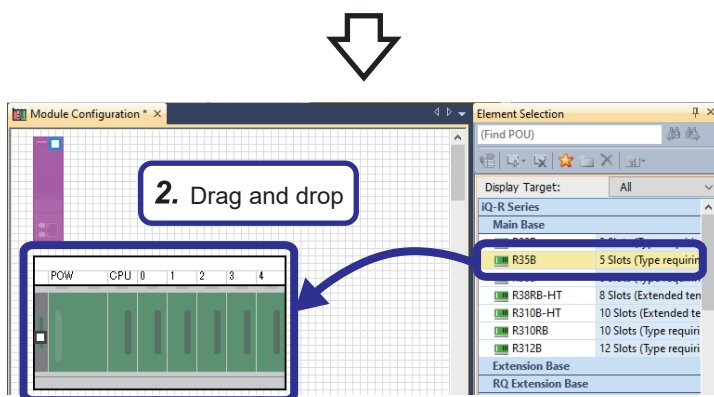
2. Drag and drop

2. Select "R35B" from the "Main Base" section on the Element Selection window, and drag and drop it to the Module Configuration window.
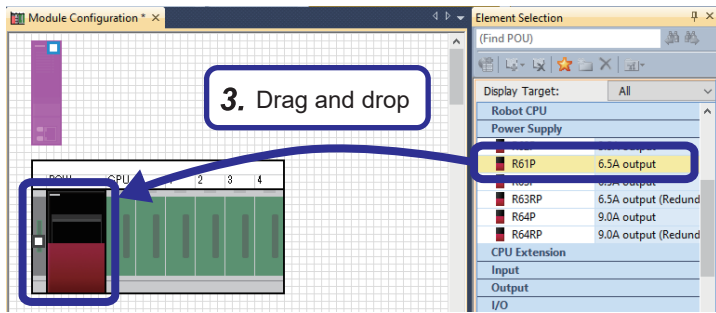
3. Select "R61P" from the "Power Supply" section on the Element Selection window, and drag and drop it to the power supply slot of the R35B that was added to the Module Configuration window.
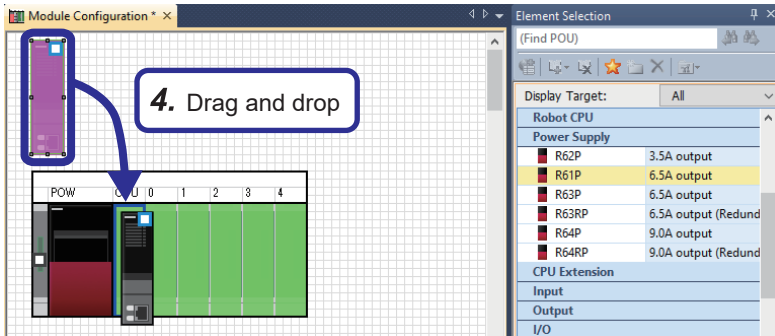
4. Drag and drop the R32PCPU displayed by default on the Module Configuration window to the CPU slot of the R35B.
Do not use CPUs on the Element Selection window because they are intended for multi-CPU configuration.
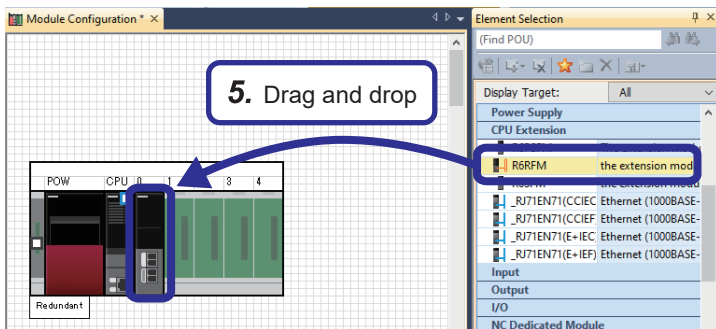
5. Select "R6RFM" from the "CPU Extension" section on the Element Selection window, and drag and drop it to slot No.0 of the R35B.
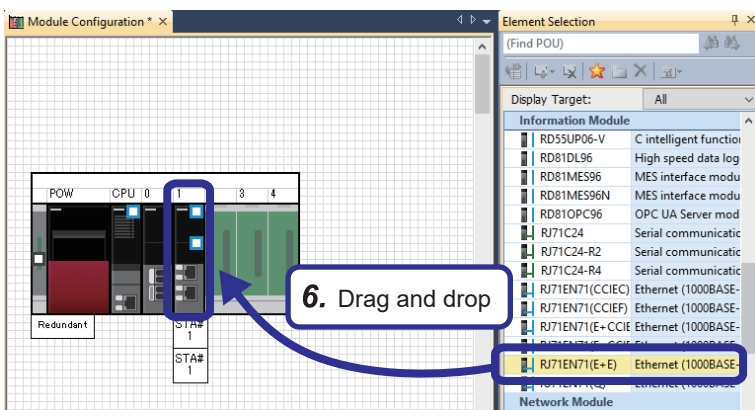
6. Select "RJ71EN71(E+E)" from the "Information Module" section on the Element Selection window, and drag and drop it to slot No.1 of the R35B.
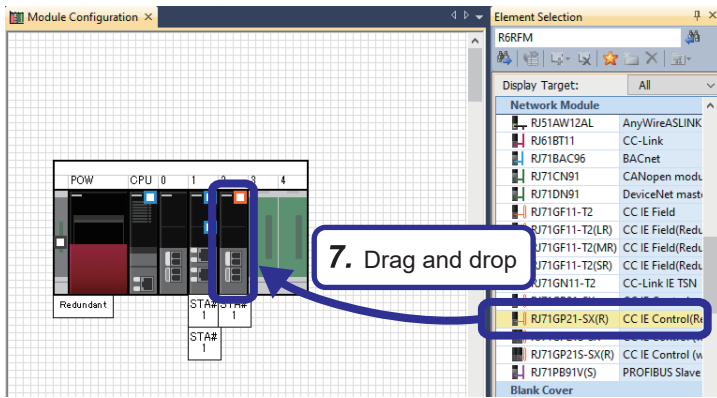
7. Select "RJ71GP21-SX(R)" from the "Network Module" section on the Element Selection window, and drag and drop it to slot No.2 of the R35B.
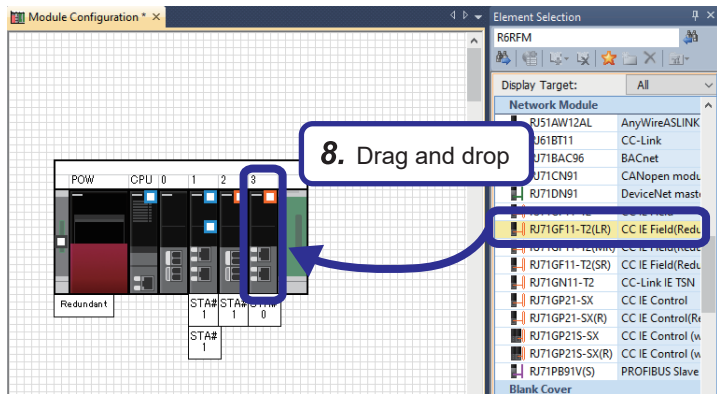
8. Select "RJ71GF11-T2(LR)" from the "Network Module" section on the Element Selection window, and drag and drop it to slot No.3 of the R35B.
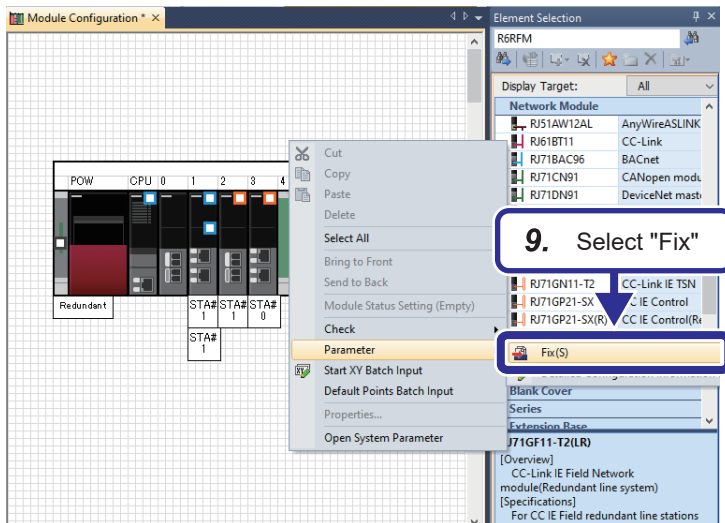
9. Right-click and select [Parameter] ⇨ [Fix] to determine the parameters.
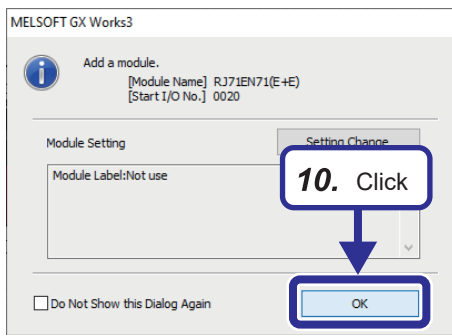
10. When the window shown on the left is displayed for each module, click the [OK] button.
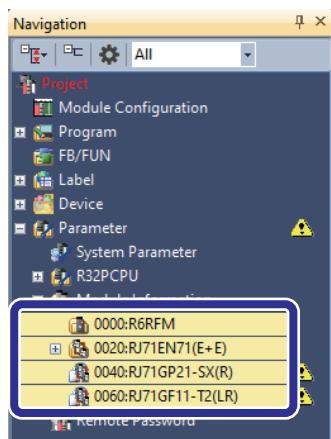
11. The module information is automatically registered on the Navigation window.

Point

Six types of elements are available for the communication module RJ71EN71 and the differences between them are as follows. Note that only (E+E) or (Q) is available for the redundant system.
 • (CCIEC): Both port 1 and port 2 are used for CC-Link IE Controller Network.
 • (CCIEF): Both port 1 and port 2 are used for CC-Link IE Field Network.
 • (E+CCIEC): Port 1 is used for Ethernet and port 2 is used for CC-Link IE Controller Network.
 • (E+CCIEF): Port 1 is used for Ethernet and port 2 is used for CC-Link IE Field Network.
 • (E+E): Both port 1 and port 2 are used for Ethernet.
 • (Q): Port 1 is used for the Q-compatible Ethernet mode, and port 2 is not used. The Q-compatible Ethernet mode ensures that module I/O signals and buffer memory addresses are compatible with the Q Series.
Four types of elements are available for the network module RJ71GF11-T2 and the differences between them are as follows.
 • (LR): Redundant line configuration
 • (MR): Redundant master station configuration
 • (SR): Redundant slave station configuration
 • None: Single-system configuration
Two types of elements are available for the network module RJ71GF21-SR and the differences between them are as follows.
 • (R): Uses a redundant system with the redundant line configuration
 • None: Single-system configuration

Setting network parameters

Set parameters necessary for configuring each network.

■Ethernet parameter setting

This section describes necessary Ethernet communication settings for the network configuration where the redundant system is connected to the personal computer via the Ethernet module RJ71EN71.
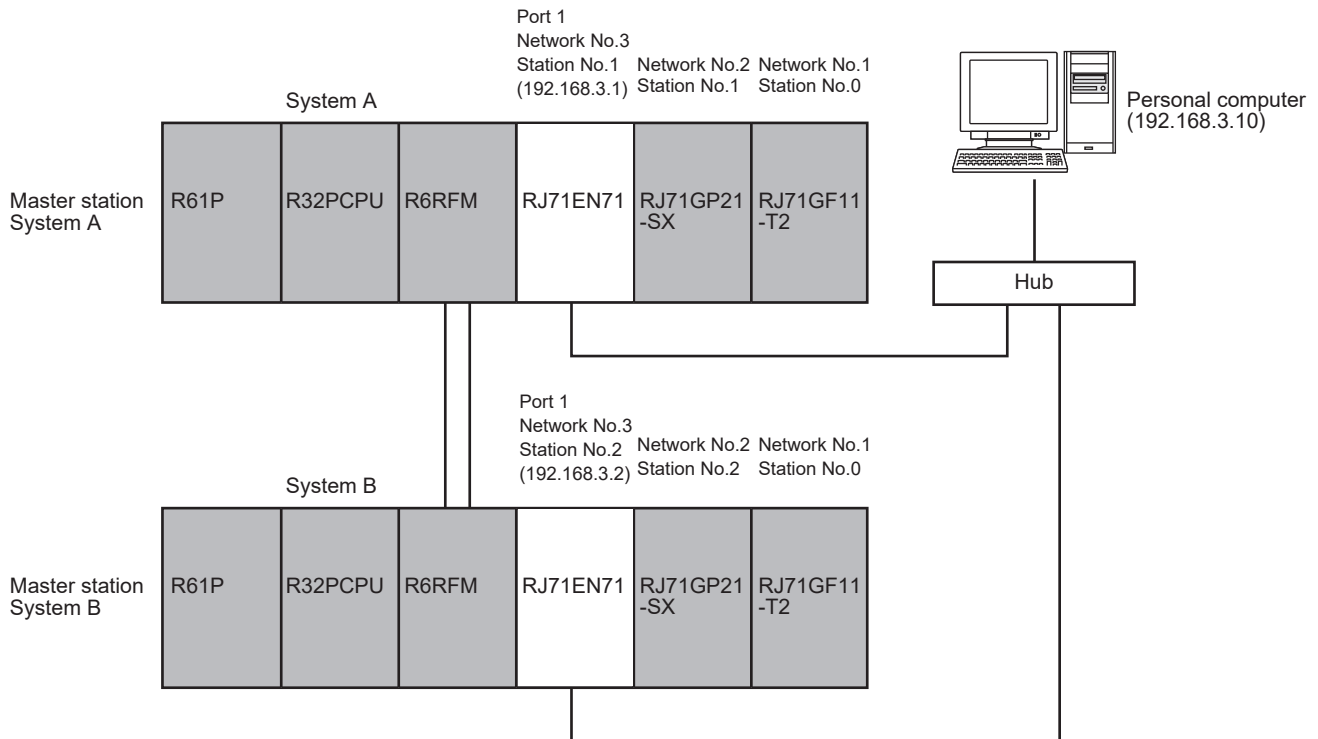
- Set the module parameters.

Set the module parameters in the RJ71EN71. Set different station numbers and IP addresses for system A and system B.

Operating procedure
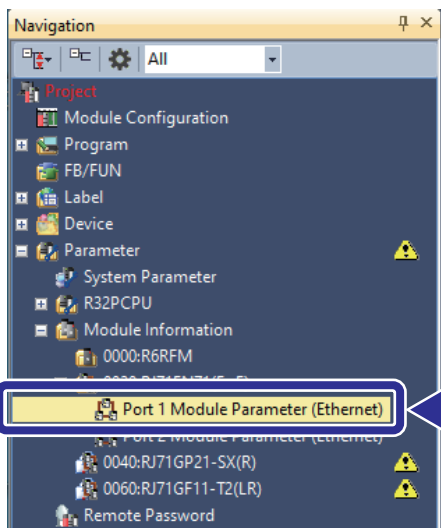
1. On the Navigation window, double-click "Port 1 Module Parameter (Ethernet)" for the Ethernet module.
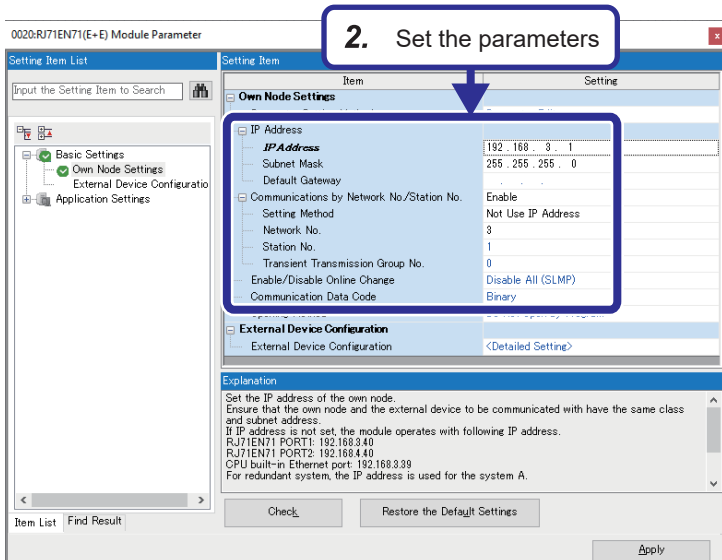
2. Set "Basic Settings".
[Settings]
IP Address: 192.168.3.1
Subnet Mask: 255.255.255.0
Communications by Network No./Station No.:
Enable
Setting Method: Not Use IP Address
Network No.: 3
Station No.: 1

> **Point**
>
> When trying to apply the settings, the message "Overlap in module (00XX:RJ71XXXXX) network No." may be displayed. In this case, open the module parameter settings of the displayed network module, and assign a different network number.
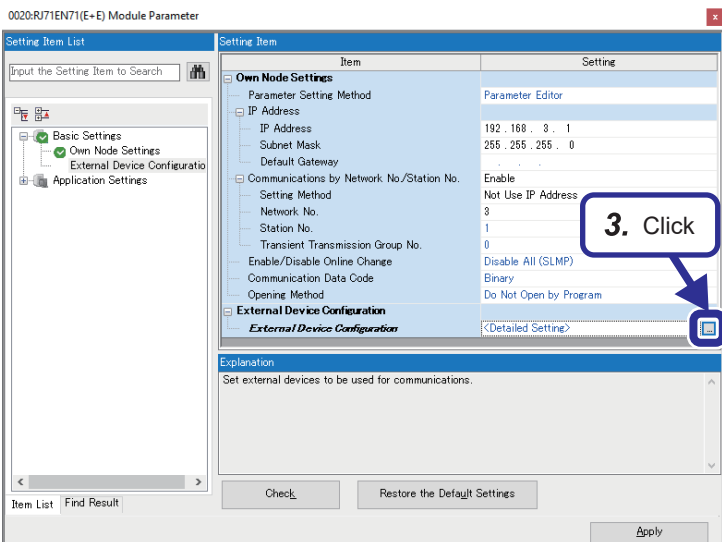
3. Select "External Device Configuration" and click the ▣ button.
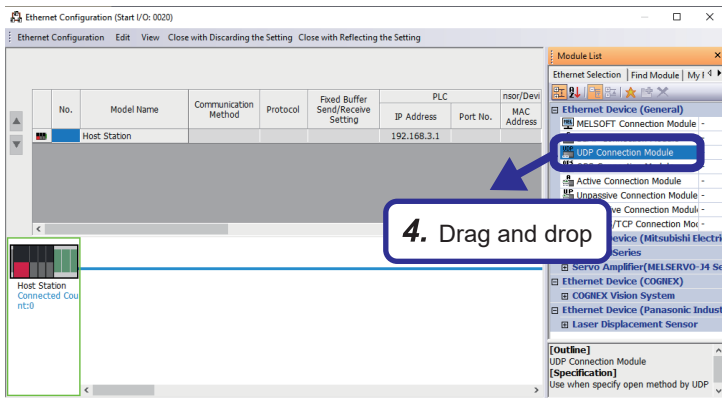
4. The "Ethernet Configuration" window appears. Select "UDP Connection Module" from the module list "Ethernet Device (General)" and drag and drop it to the device list or device map area.

4. Drag and drop
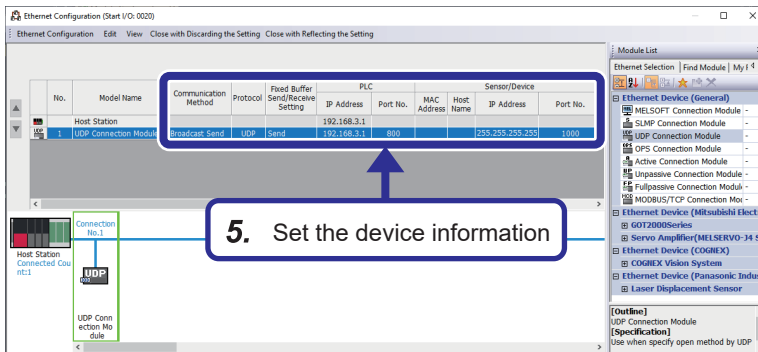
5. "UDP Connection Module" is added to the device list.
Set "UDP Connection Module" as follows.
[Settings]
Communication Method: Broadcast Send
Port No. (PLC): 800
Port No. (Sensor/Device): 1000

5. Set the device information
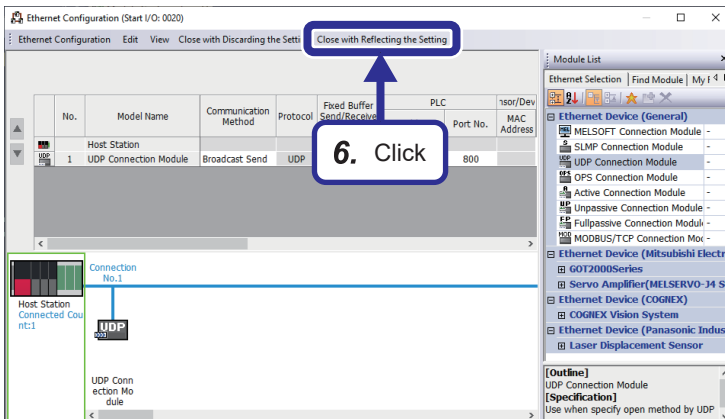
6. Then, click the [Close with Reflecting the Setting] from the menu bar to close the "Ethernet Configuration" window.

6. Click

A
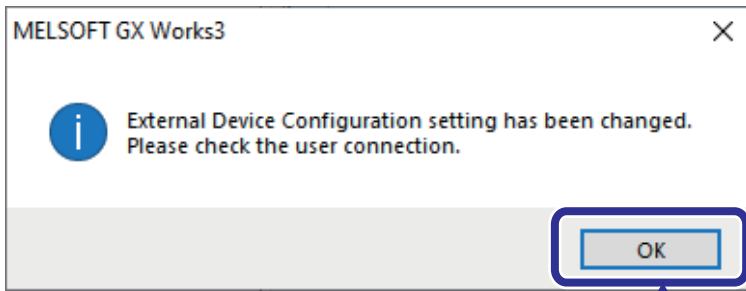
7. The user connection confirmation window appears. Click the [OK] button.
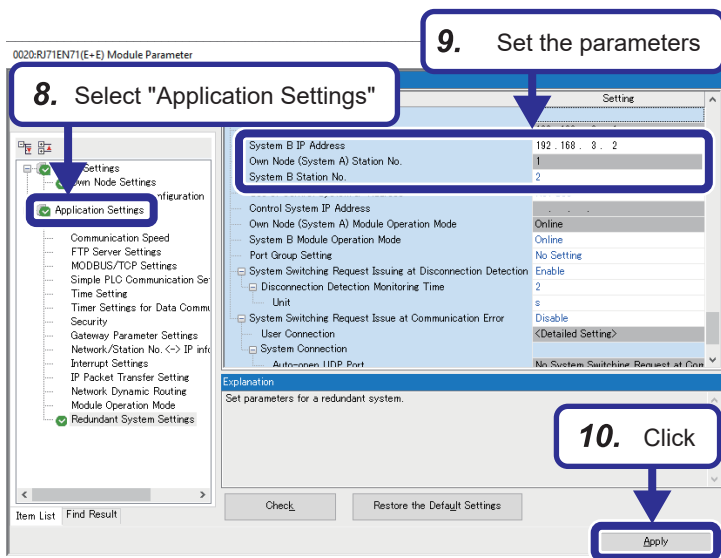
8. Select "Application Settings" from the list of setting items.

9. Set the "Redundant System Settings" as follows.
[Settings]
System B IP Address: 192.168.3.2
System B Station No.: 2

10. Click the [Apply] button.

Point

- When "Use IP Address" is selected for "Setting Method", the network number and station number are set to the third and fourth octets of the IP address. When the IP address is set to 192.168.3.1, the network number is set to 3 and the station number is set to 1. For details, refer to the following.
  📖 MELSEC iQ-R Ethernet User's Manual (Application)
- Because the network number is set to 3 for Ethernet of the demonstration machines, set this setting item to "Not Use IP Address".
- By setting the same IP address (control system IP address) for both systems, system A and system B use the common IP address (control system IP address) and the program can maintain communications without switching the connected system regardless of system switching. For details, refer to the following.
  📖 MELSEC iQ-R Ethernet User's Manual (Application) For demonstration machines, there is no need to set the control system IP address.

• Set the option setting.

Set the option setting to enable the event notification function. The event notification function is a monitor tool that monitors warnings and events and suppresses communication traffic.
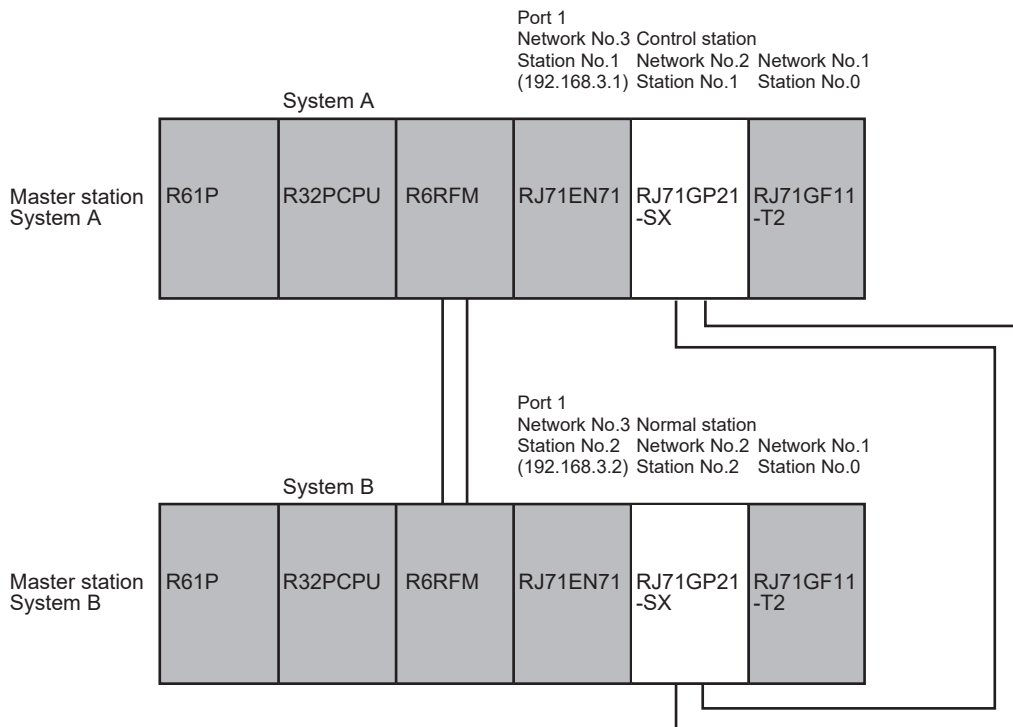
Operating procedure
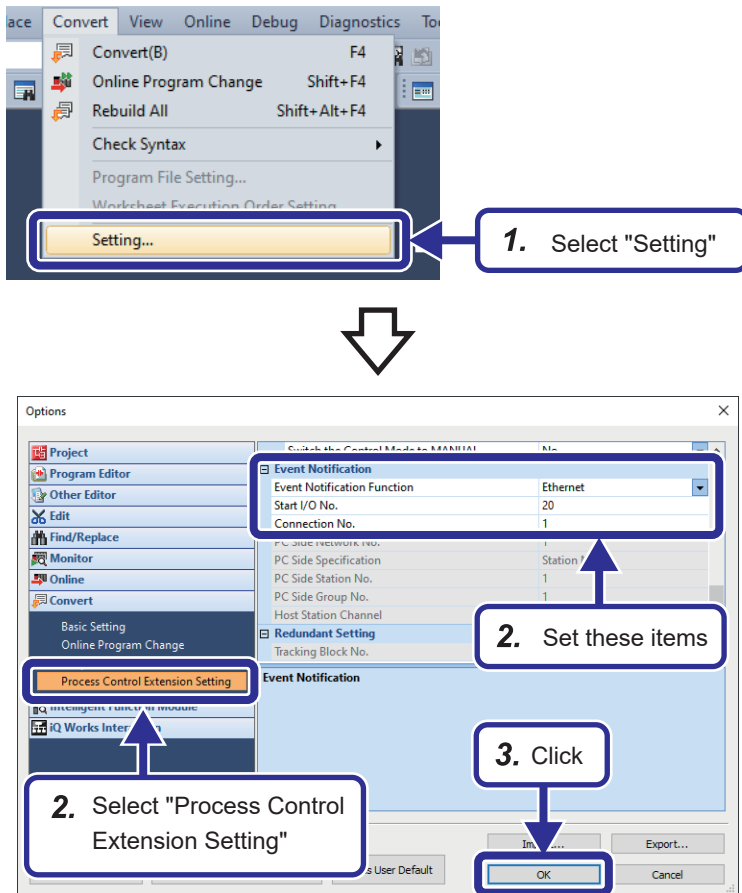
1. Select "Setting"

1. Select [Convert] ⇨ [Setting] to open the "Options" window.

2. Select "Process Control Extension Setting"

2. Set these items

3. Click

2. Select "Process Control Extension Setting" and set "Event Notification" as follows.
[Settings]
Event Notification Function: Ethernet
Start I/O No.: 20
Connection No.: 1

3. Click the [OK] button.

A

■**Setting the parameters of CC-Link IE Controller Network**

This section describes the settings necessary for CC-Link IE Controller Network communications in the network configuration where redundant system A and system B are connected via the CC-Link IE Controller Network module RJ71GP21-SX.

• Set the module parameters.

Set the module parameters in the RJ71GP21-SX.

Operating procedure
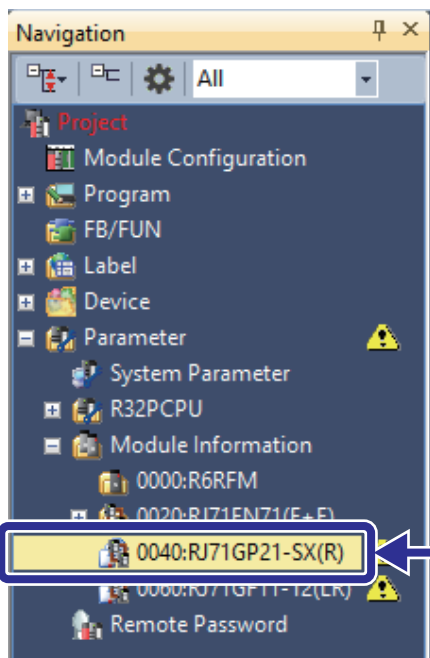
1. Double-click "0040:RJ71GP21-SX(R)" (CC-Link IE Controller Network module) on the Navigation window.
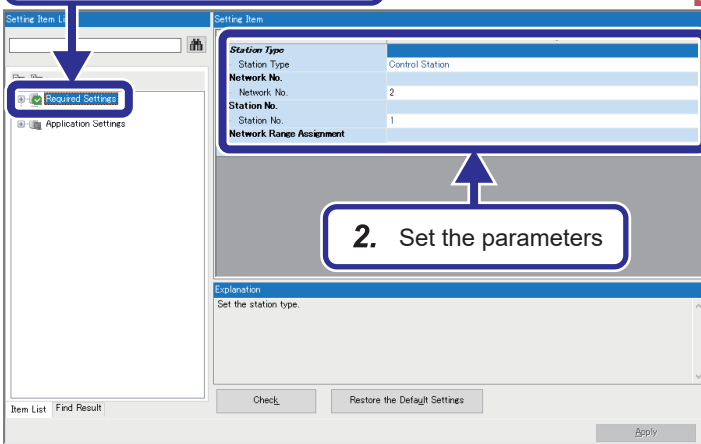
2. Set "Required Settings" as follows.
[Settings]
Station Type: Control Station
Network No.: 2
Station No: 1
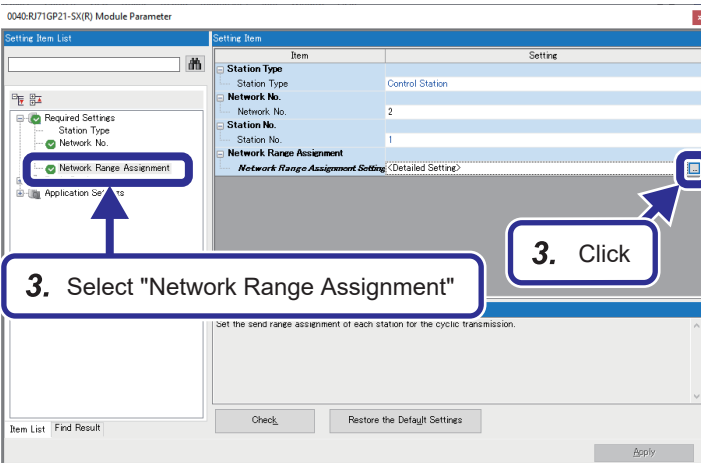
3. Select "Network Range Assignment" and click the ⬚ button.
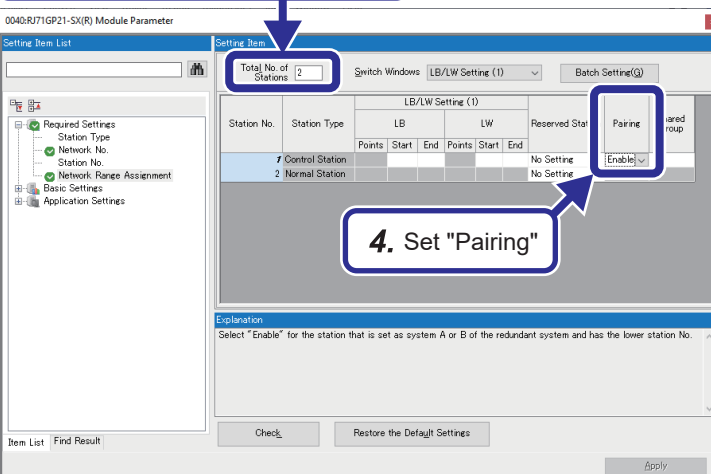
4. Set "Network Range Assignment" as follows.
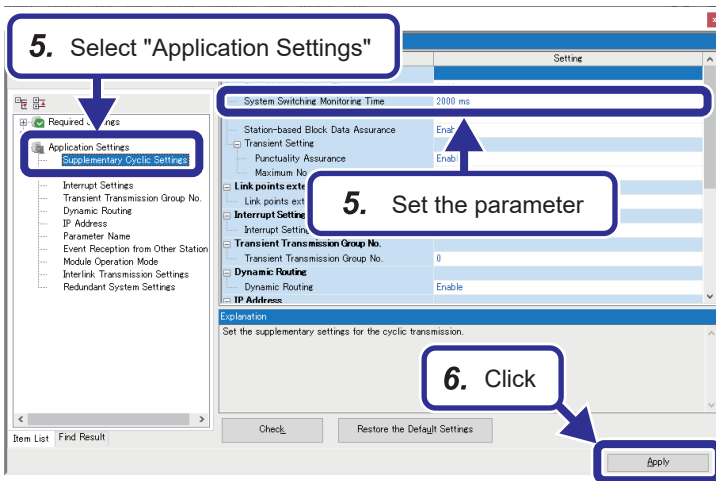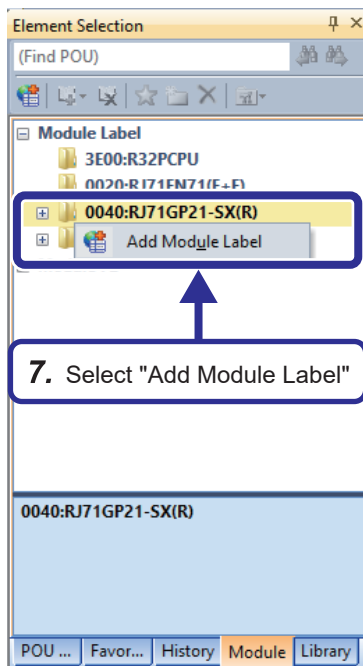[Settings]
Total No. of Stations: 2
Pairing: Enable

5. Select "Supplementary Cyclic Settings" under "Application Settings" and set "System Switching Monitoring Time" to "2000ms".

6. Click the [Apply] button.

7. Add the module label. On the "Module" tab of the "Element Selection" window, right-click "0040:RJ71GP21-SX(R)" and select [Add Module Label].

■Setting the CC-Link IE Field Network parameters

Set the parameters necessary to start CC-Link IE Field Network communications in the redundant line network configuration where the CC-Link IE Field Network module RJ71GF11-T2 is mounted to the master station or where the two CC-Link IE Field remote head modules RJ71GF15-T2 are mounted to the slave station.

This section describes the procedure for setting the parameters in the CC-Link IE Field Network module in the master station.
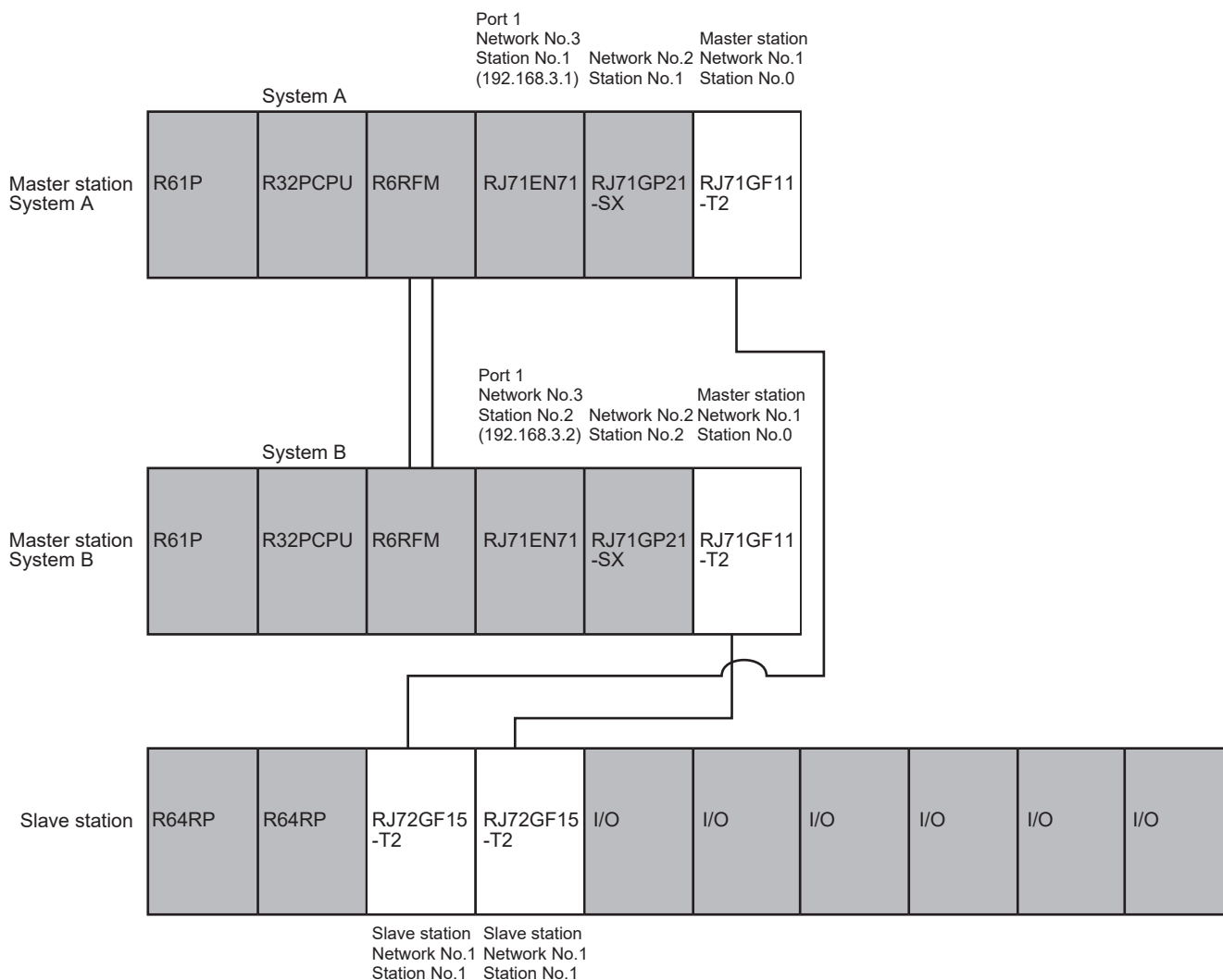
• Set the module parameters.

Set the refresh range by referring to the following signal assignment table.

Master station CPU R32P	Master station RJ71GF11-T2 Refresh device	Send direction	Slave station RJ72GF15-T2 Refresh device[*1]	Remote head module RJ72GF15-T2
X1000 to X10FF	RX0000 to RX00FF	←	RX1000 to RX10FF	X0000 to X00FF
Y1000 to Y10FF	RY0000 to RY00FF	→	RY1000 to RY10FF	Y0000 to Y00FF
W1500 to W15FF	RWr0000 to RWr00FF	←	RWr0000 to RWr00FF	W1000 to W10FF
W1000 to W10FF	RWw0000 to RWw00FF	→	RWw0000 to RWw00FF	W0000 to W00FF

*1 For the refresh devices RWw and RWr of the remote head module RJ72GF15-T2, the assignment starting devices are fixed to "RWw → W0, RWr → W1000".
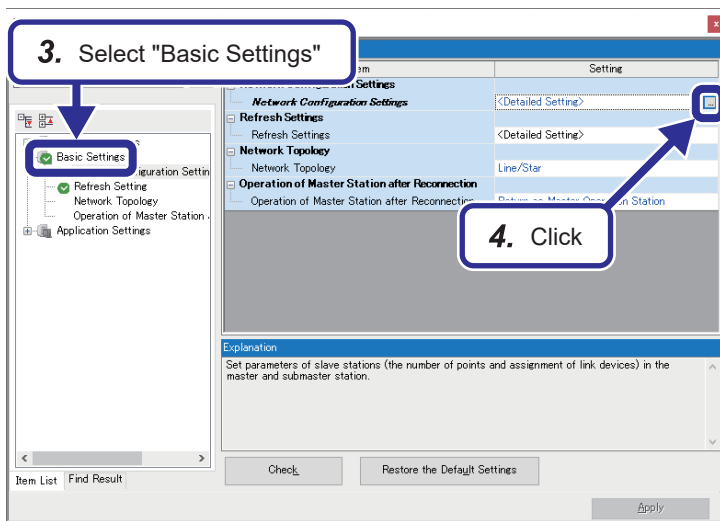
Operating procedure
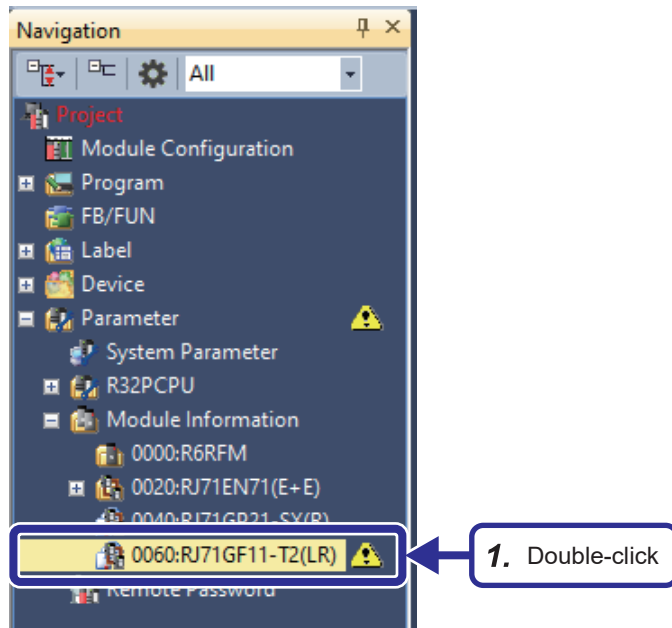
1. Double-click "0060:RJ71GF11-T2(LR)" (CC-Link IE Field Network module) on the Navigation window.
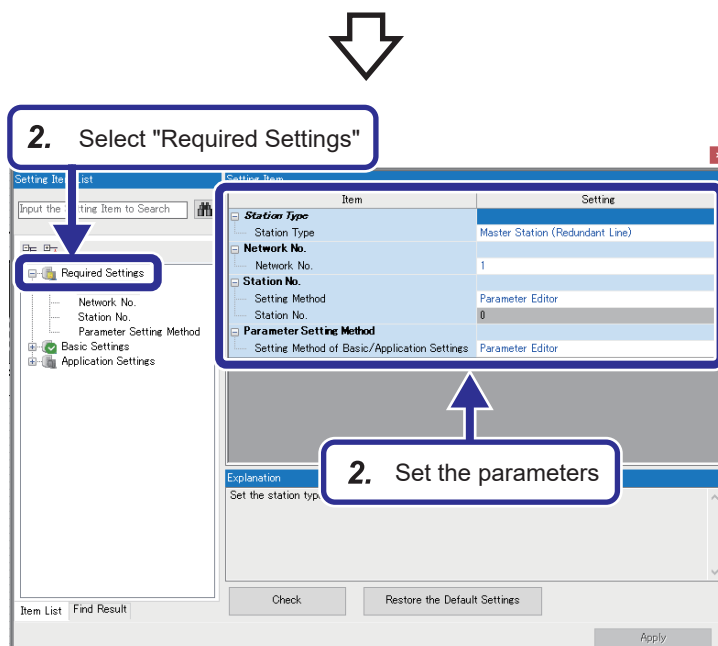

2. Set "Required Settings" as follows.
[Settings]
Station Type: Master Station (Redundant Line)
Network No.: 1
Setting Method: Parameter Editor
Setting Method of Basic/Application Settings: Parameter Editor

3. Select "Basic Settings"

4. Click

3. Select "Basic Settings" from the list of setting items.

4. Click the button for "Network Configuration Settings".
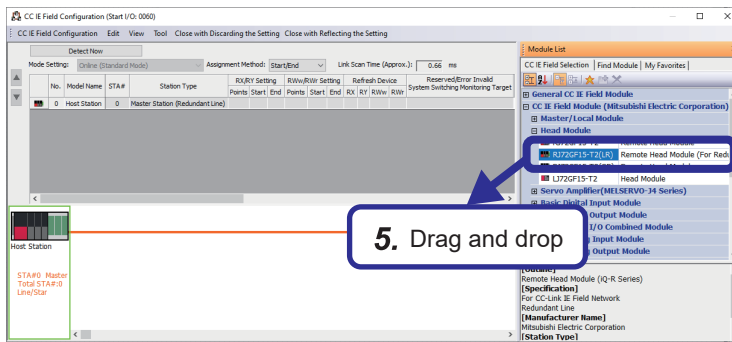
5. Drag and drop

5. The "CC IE Field Configuration" window appears. Select "RJ72GF15-T2(LR)" from "Head Module" in the module list and drag and drop it to the device list or device map area.

6. Specify "Assignment Method"

6. Set the module information
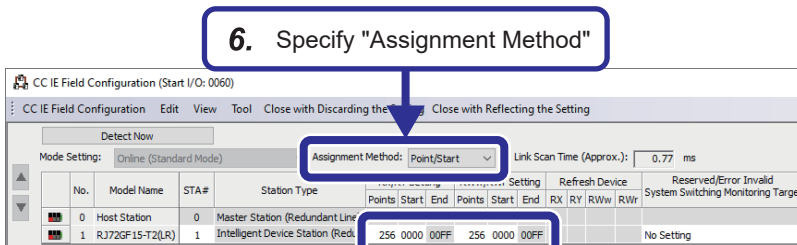
6. "RJ72GF15-T2(LR)" is added to the device list.
Change "Assignment Method" to "Point/Start", and set "RJ72GF15-T2(LR)" as follows.
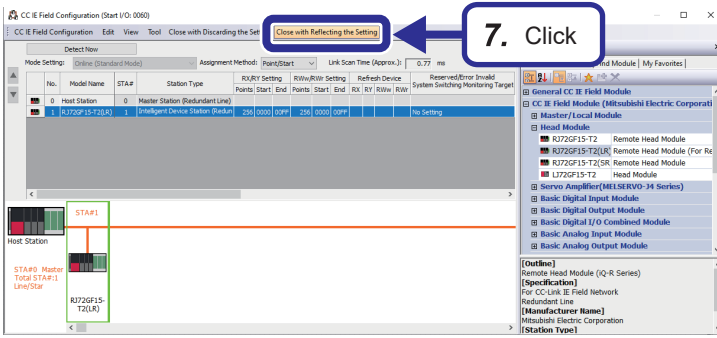[Settings]
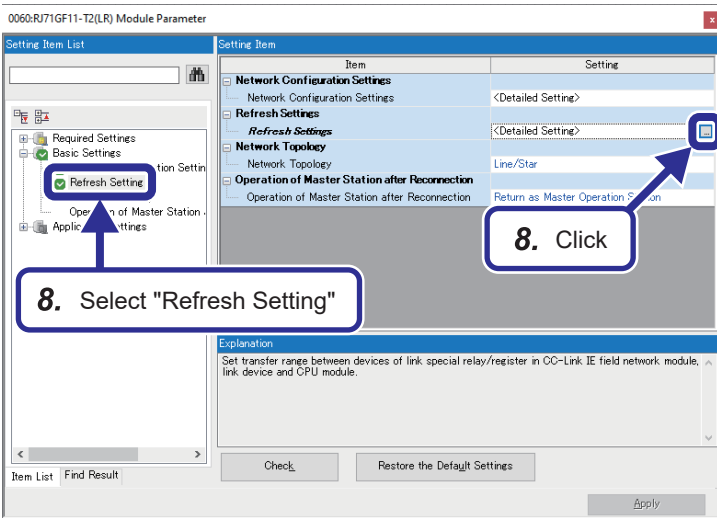Points (RX/RY Setting): 256
Start (RX/RY Setting): 0000
Points (RWw/RWr Setting): 256
Start (RWw/RWr Setting): 0000

A

7. Then, click "Close with Reflecting the Setting" from the menu bar to close the "CC IE Field Configuration" window.

8. Select "Refresh Setting" and click the ▦ button.

9. Set the link devices and their range as shown in the left figure.

10. Select "Application Settings"

11. Set the parameter
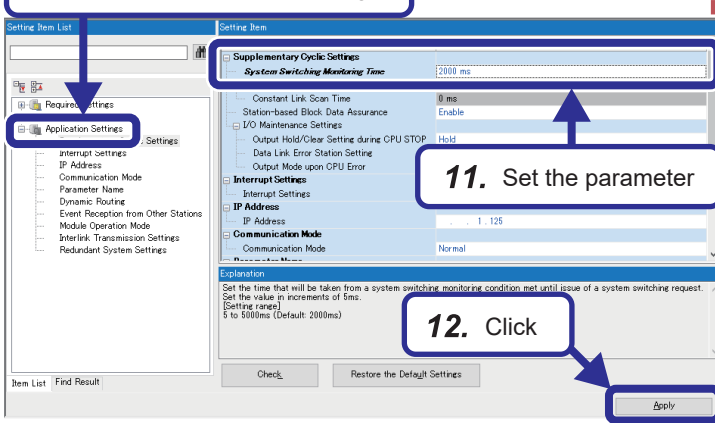
12. Click

10. Select "Application Settings" from the list of setting items.

11. Set "System Switching Monitoring Time" to "2000ms".

12. Click the [Apply] button.
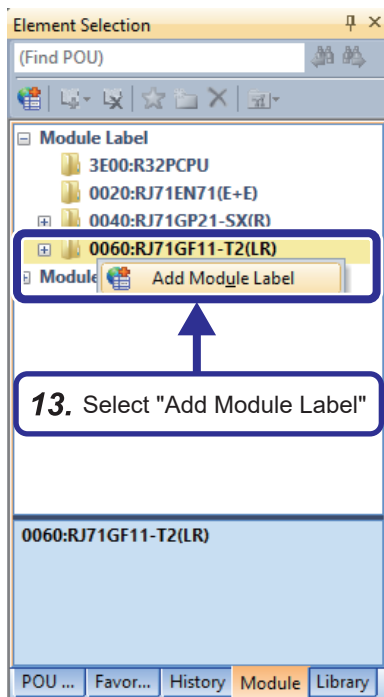
13. Select "Add Module Label"

13. Add the module label. On the "Module" tab of the "Element Selection" window, right-click "0060:RJ71GF11-T2(LR)" and select [Add Module Label].

Tracking setting

Set tracking from "CPU Parameter" of "School_Master".

Operating procedure
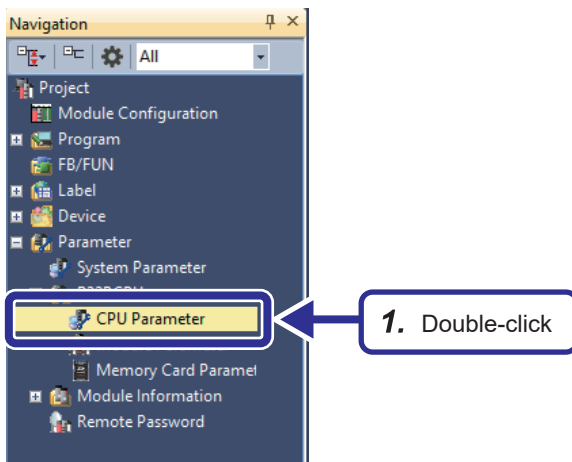
1. Double-click "CPU Parameter" on the Navigation window of "School_Master".
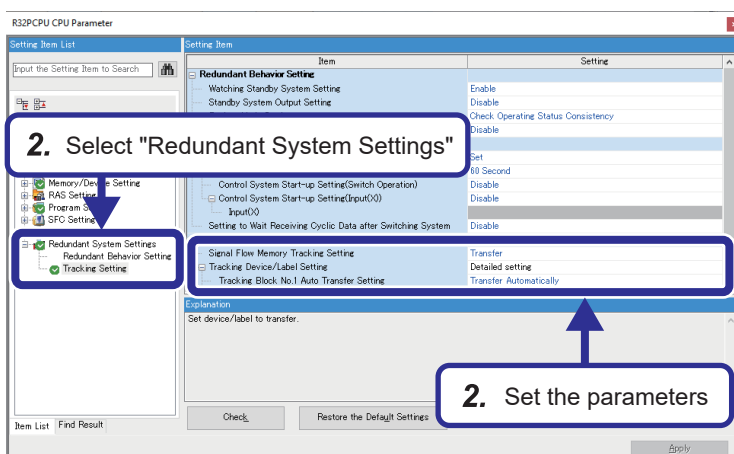
2. Select "Redundant System Settings" from the list of setting items, and set "Tracking Setting" as follows.
   [Settings]
   Signal Flow Memory Tracking Setting: Transfer
   Tracking Device/Label Setting[*1]
   Tracking Block No.1 Auto Transfer Setting: Transfer Automatically

*1 When using the process control extension, always set this item to "Detailed setting" because the file register (ZR) is not transferred when it is set to "Transfer collectively". (The file register is automatically registered to the block No.64 during conversion.)
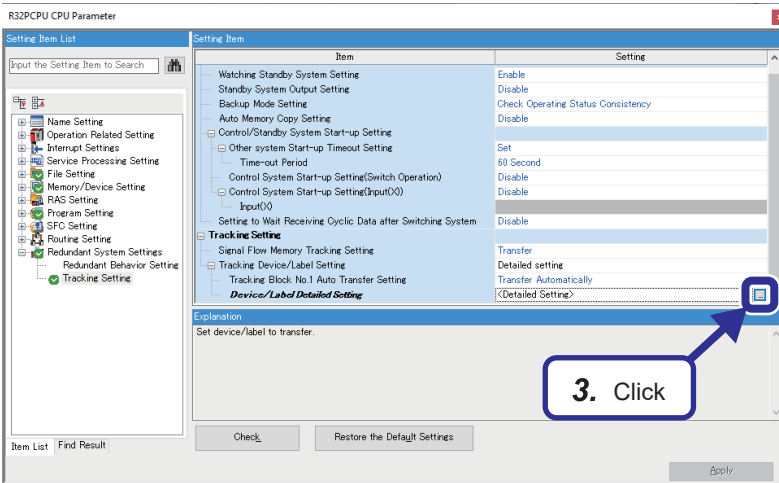
3. Select "Device/Label Detailed Setting" and click the ... button.
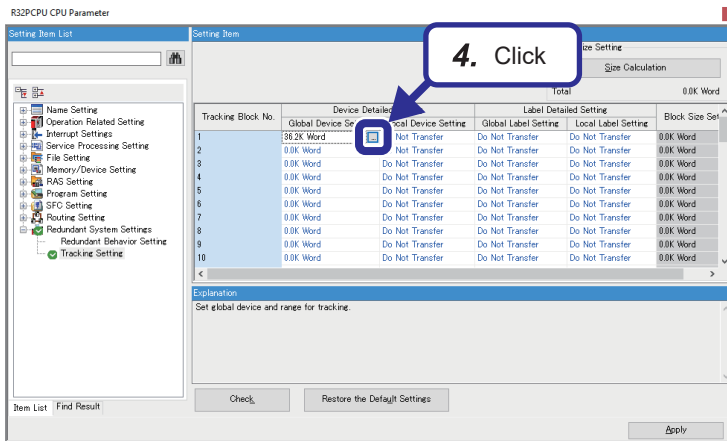
4. Click the ... button for "Global Device Setting" of the tracking block No.1.
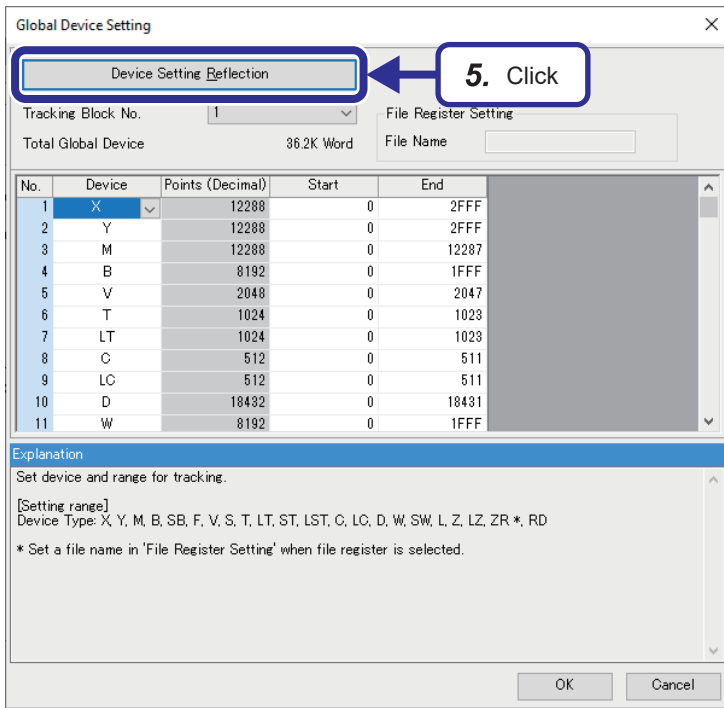
5. The "Global Device Setting" window appears. Click the [Device Setting Reflection] button.
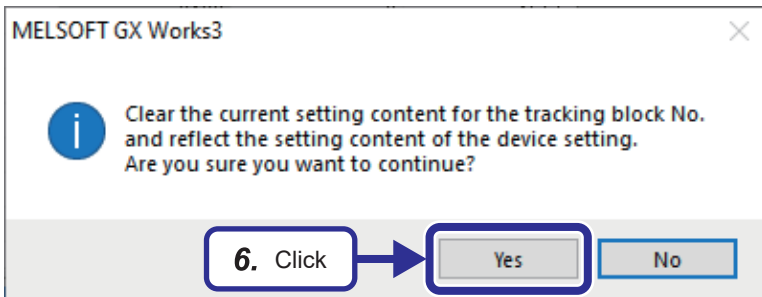
6. Click the [Yes] button to automatically apply the device range being used.

APPX
Appendix 6 Creating Redundant System Projects
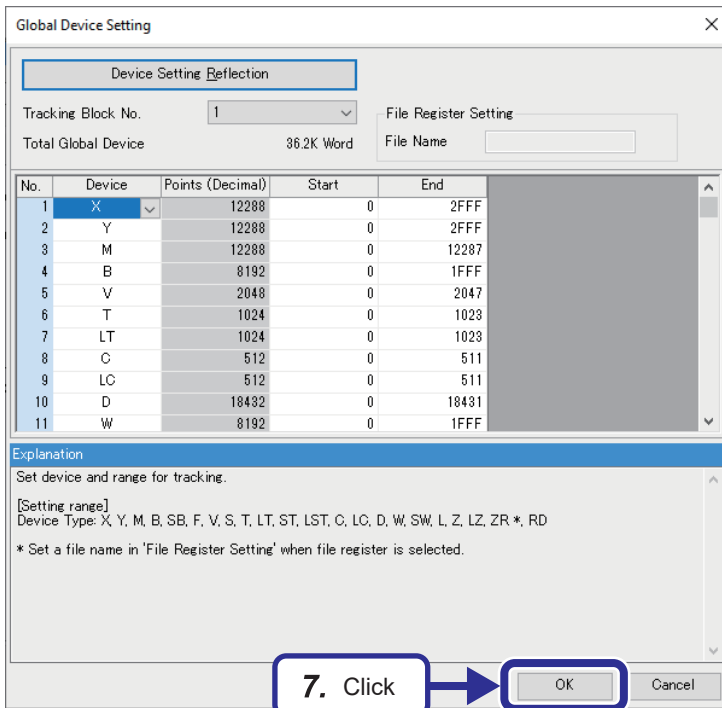
7. Click the [OK] button to close the "Global Device Setting" window.
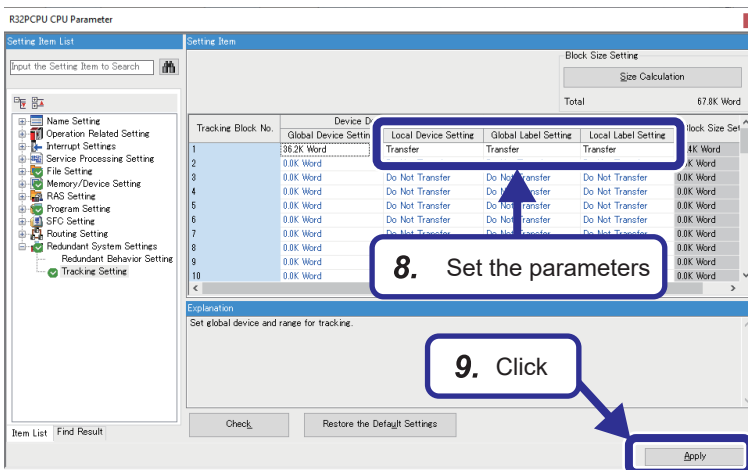
8. Set the tracking block No.1 as follows.
[Settings]
Local Device Setting: Transfer
Global Label Setting: Transfer
Local Label Setting: Transfer

9. Click the [Apply] button.

Point

Track the total range of the devices being used, except for when particular devices need not be tracked (such as when specific device values are different per system in a program running in both systems).

Programming

Creating a new program file

Create a new program file in "School_Master". In this example, a scan-type ladder program file will be created.

Operating procedure
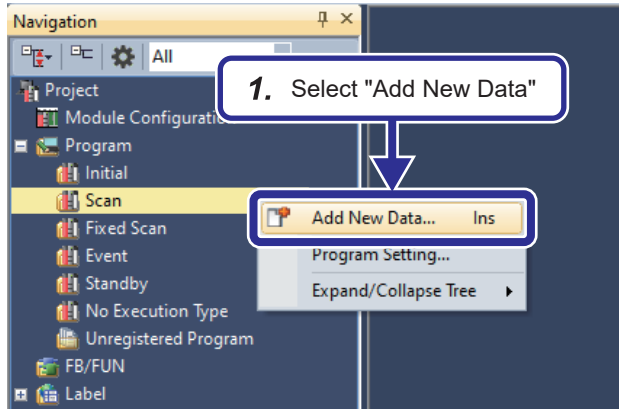
1. Right-click "Scan" on the Navigation window of "School_Master", and select "Add New Data".
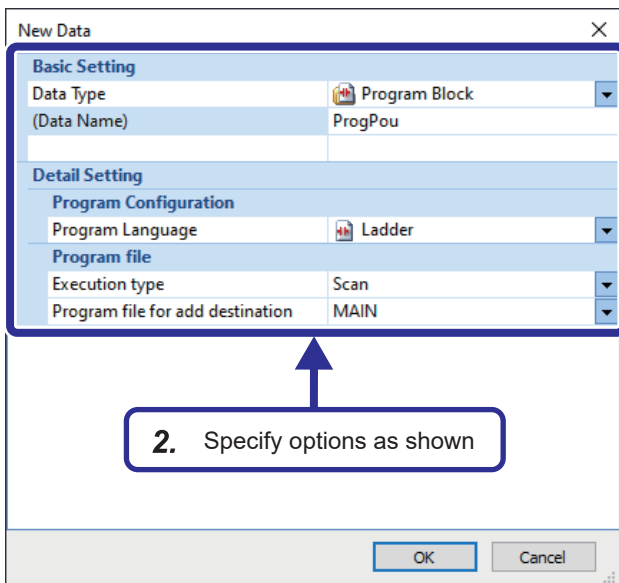
2. The "New Data" window appears. Specify the options as shown in the left figure.
Specify "(Data Name)" and "Program file for add destination" to any value.
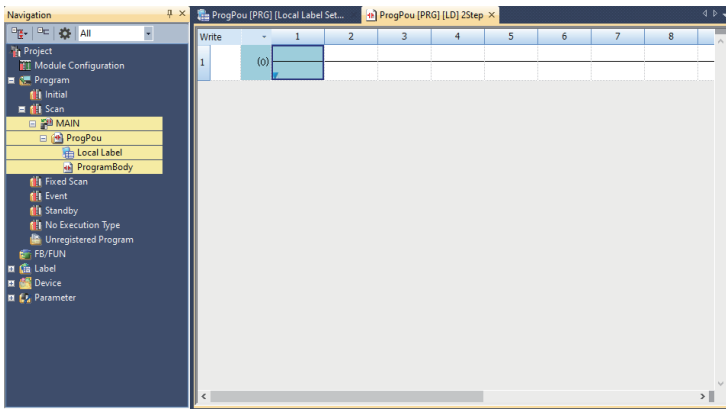
3. A new program file is created on the Navigation window, and the sheet is displayed.

Ladder programming

Create a program necessary for signal input/output in "School_Master". The program necessary for signal input/output from slave stations consists of the following five circuits.

- CC-Link IE Field Network interlock circuit
- 24VDC input/output and contact output transfer circuit (RY10R2, RX40NC6B, RY40PT5B)
- Temperature measured value read circuit (R60TD8-G)
- DA conversion execution circuit (R60DA16-G)
- AD conversion execution circuit (R60AD8-G)

Example program for inputting/outputting signals from a slave station

R60DA16-G_DA conversion execution

```
        X1070              X1077                                                                    ┌─────┬─────────┬─────────┐
(114)   ─┤ ├───────────────┤ ├──────────────────────────────────────────────────────────────────── │ MOV │   D21   │  W1071  │
        Module READY    External power supply                                                       │     │ CH1 digital │ CH1 digital │
                        READY flag                                                                   │     │ value   │ value   │
                                                                                                    │     │ temporary │         │
                                                                                                    │     │ area    │         │
                                                                                                    └─────┴─────────┴─────────┘

                                                                                                    ┌─────┬─────────────┐
                                                                                                    │ SET │   Y1071     │
                                                                                                    │     │ CH1         │
                                                                                                    │     │ Output      │
                                                                                                    │     │ enable/     │
                                                                                                    │     │ disable     │
                                                                                                    │     │ flag        │
                                                                                                    └─────┴─────────────┘

                                                                                                    ┌─────┬─────────┬─────────┐
                                                                                                    │ MOV │   D22   │  W1072  │
                                                                                                    │     │ CH2 digital │ CH2 digital │
                                                                                                    │     │ value   │ value   │
                                                                                                    │     │ temporary │         │
                                                                                                    │     │ area    │         │
                                                                                                    └─────┴─────────┴─────────┘

                                                                                                    ┌─────┬─────────────┐
                                                                                                    │ SET │   Y1072     │
                                                                                                    │     │ CH2         │
                                                                                                    │     │ Output      │
                                                                                                    │     │ enable/     │
                                                                                                    │     │ disable     │
                                                                                                    │     │ flag        │
                                                                                                    └─────┴─────────────┘

                                                                                                    ┌─────┬─────────────┐
                     ─────/─────────────────────────────────────────────────────────────────────── │ RST │   Y1071     │
                                                                                                    │     │ CH1         │
                                                                                                    │     │ Output      │
                                                                                                    │     │ enable/     │
                                                                                                    │     │ disable     │
                                                                                                    │     │ flag        │
                                                                                                    └─────┴─────────────┘

                                                                                                    ┌─────┬─────────────┐
                                                                                                    │ RST │   Y1072     │
                                                                                                    │     │ CH2         │
                                                                                                    │     │ Output      │
                                                                                                    │     │ enable      │
                                                                                                    │     │ /disable    │
                                                                                                    │     │ flag        │
                                                                                                    └─────┴─────────────┘
```

R60AD8-G_AD conversion execution

```
        X1090              X109E              Y1099        W1590.0                                   ┌─────┬─────────┬─────────┐
(143)   ─┤ ├───────────────┤ ├────────────────┤/├──────────┤ ├────────────────────────────────────── │ MOV │  W1591  │   D31   │
        Module READY    A/D conversion     Operating     CH1                                        │     │ CH1 analog │ CH1 analog │
                        completed flag      condition      Conversion                               │     │ input value │ input value │
                                            setting        completed                                │     │         │ temporary │
                                            request        flag                                     │     │         │ area    │
                                                                                                    └─────┴─────────┴─────────┘

                                                        W1590.1                                     ┌─────┬─────────┬─────────┐
                                                        ─┤ ├────────────────────────────────────── │ MOV │  W1592  │   D32   │
                                                        CH2                                         │     │ CH2 analog │ CH2 analog │
                                                        Conversion                                  │     │ input value │ input value │
                                                        completed                                   │     │         │ temporary │
                                                        flag                                        │     │         │ area    │
                                                                                                    └─────┴─────────┴─────────┘

                                                                                                    ┌─────┬─────────────┐
(171)   ──────────────────────────────────────────────────────────────────────────────────────────│ MCR │    N0       │
                                                                                                    └─────┴─────────────┘


(172)   ───────────────────────────────────────────────────────────────────────────────────────────────────────{END }
```

■CC-Link IE Field Network interlock circuit

An interlock circuit is required for the CC-Link IE Field Network to monitor data link status of the own station and other stations. When link-down occurs, establish an interlock using the following module labels so that the link refresh values are not used.[1]

- Own station data link error status (SB0049)
- Each station redundant system data link status (SW01B0 to SW01B7)

[1] When the Process CPU with firmware version "18" or later is used, the "Setting to wait cyclic data receive after system switching" can be used instead of using the CC-Link IE Field Network interlock circuit to address the issue of cyclic data being turned off at the time of system switching. For details, refer to the following.
　📖 MELSEC iQ-R CPU Module User's Manual (Application)

This program uses the MC instruction to interlock all the circuits using link refresh values. All the following circuits will be included in the MC and MCR instructions.

- List of devices to be used

Device <Module label>	Description
SB0049 <GF11_1.bSt_DataLink Error>	Data link error status of own station
SW01B0.0 <GF11_1.bnSts_RedundantSystemDataLinkError_Station[1]>	Each station redundant system data link status (station No.1)
N0	Nesting
M0	Communication ready flag (station No.1)

- Program example
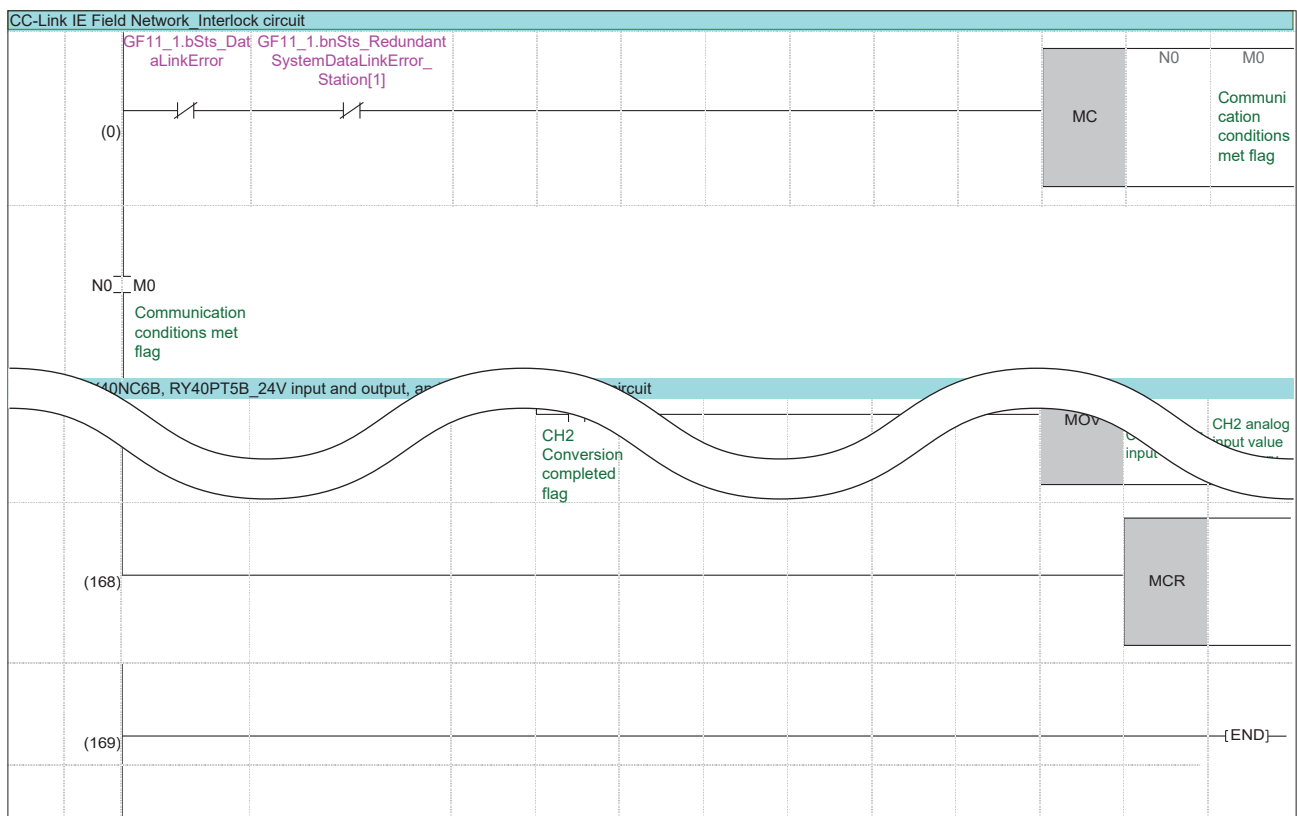
MC instructions are not executed when a data link error occurs.

If the execution command is off, the MC instruction performs operation for devices up to the MCR instruction as follows.

- Device in OUT instruction: Forced off
- Device in other instruction: Hold
- High-speed (low-speed) timer: The count value is set to 0, and the coil and contact are turned off.
- High-speed (low-speed) retentive timer, counter: The coil is turned off, and the count value and contact are maintained.

■24VDC input/output and contact output transfer circuit (RX40NC6B, RY40PT5B, RY10R2)

To perform bulk interlock processing for the CC-Link IE Field Network, create a buffer program. Devices used as temporary areas are used for input/output processing in the project. By interlocking this program, there is no need to write all the interlock circuits to the project.

Note that CH1 of the contact output module RY10R2 is used as a heating signal for the heater.

- Device list

Device	Description	
SM400	Always on	Special relay
X1000 to X100F	RX40NC6B input	Slave station X0000 to X000F
Y1040 to Y104F	RY10R2 output	Slave station Y0040 to Y004F
Y1020 to Y102F	RY40PT5B output	Slave station Y0020 to Y002F
D0	RY10R2 output temporary area	
D1	RX40NC6B input temporary area	
D2	RY40PT5B output temporary area	

- Program example

RY10R2, RX40NC6B, RY40PT5B_24V input and output, and contact output transfer circuit

I/O status is always transferred between temporary storage devices and refresh devices.

■Temperature measured value read circuit (R60TD8-G)

Create a program that inputs thermocouple signals input to the R60TD8-G.

Apply an interlock to prevent the input value from being used when signal input fails. Temporary storage devices are used to read temperature measured values in the project.

- Device list

Device	Description	
X1050	Module READY	Slave station X0050
X105E	Conversion completed flag	Slave station X005E
Y1059	Operating condition setting request	Slave station Y0059
W1550.0	[CH1] Conversion completed flag	Slave station W1050.0
W1551	[CH1] Temperature measured value	Slave station W1051
D11	[CH1] Temperature measured value temporary area	
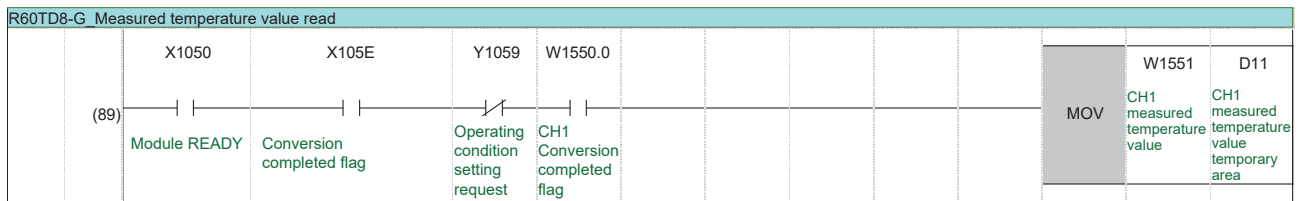
- Program example

R60TD8-G_Measured temperature value read

	X1050	X105E	Y1059	W1550.0						MOV	W1551	D11
(89)	Module READY	Conversion completed flag	Operating condition setting request	CH1 Conversion completed flag							CH1 measured temperature value	CH1 measured temperature value temporary area

Point

Apply an interlock to prevent the input value from being used when signal conversion fails.

■DA conversion execution circuit (R60DA16-G)

Create a program that outputs analog signals from the R60DA16-G.

Apply an interlock to prevent output when signal output fails. Temporary storage devices are used to read or modify output values in the project.

- Device list

Device	Description	
X1070	Module READY	Slave station X0070
X1077	External power supply READY flag	Slave station X0077
Y1071	[CH1] Output enable/disable flag	Slave station Y0071
Y1072	[CH2] Output enable/disable flag	Slave station Y0072
W1071	[CH1] Digital output value	Slave station W0071
W1072	[CH2] Digital output value	Slave station W0072
D21	[CH1] Digital output value temporary area	
D22	[CH2] Digital output value temporary area	

- Program example
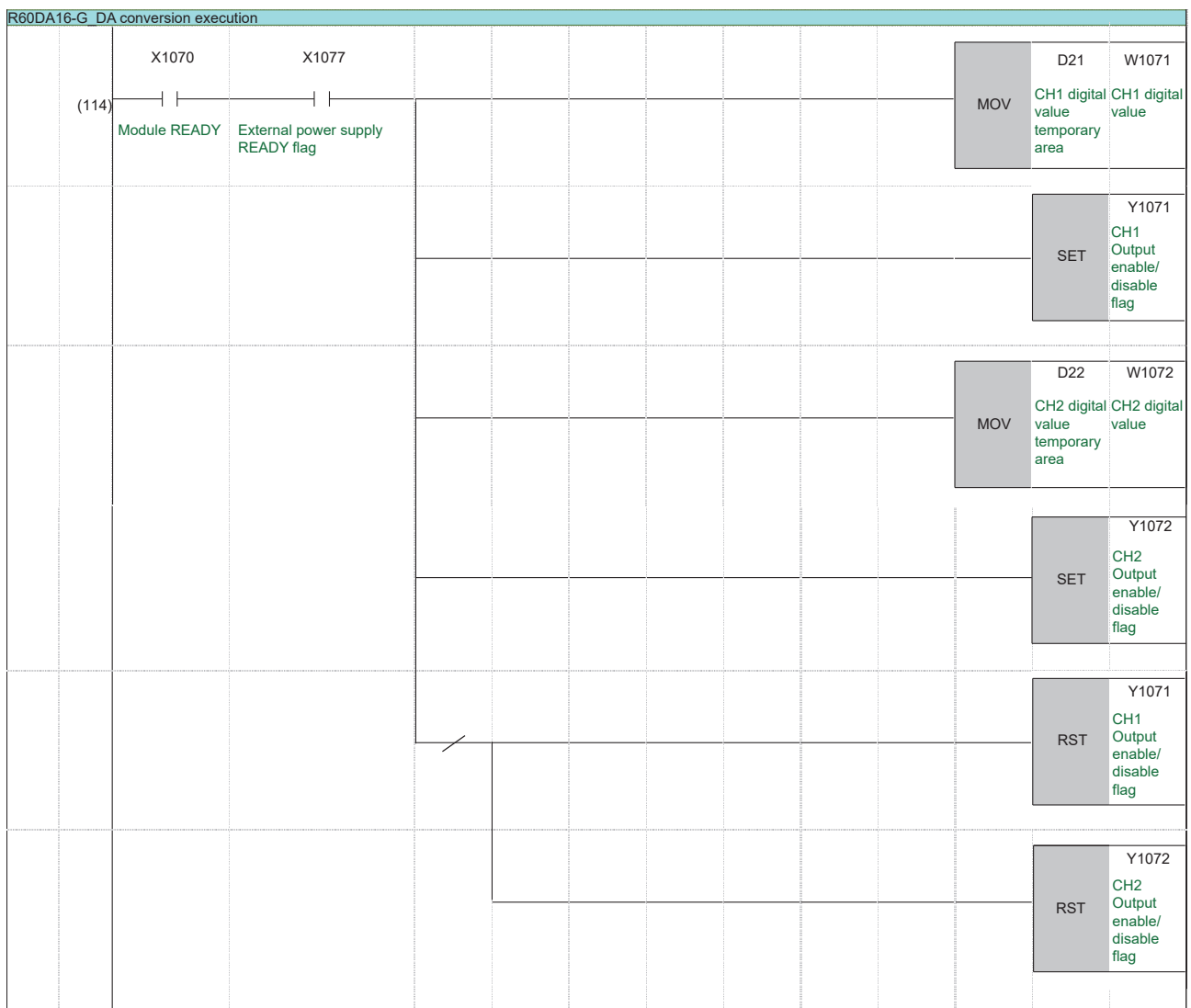
Point

Apply an interlock to prevent output when signal output is abnormal.

Analog output is stopped when Output enable flags (Y1071, Y1072) of the R60DA16-G are turned off.

■AD conversion execution circuit (R60AD8-G)

Create a program that inputs analog signals input to the R60AD8-G.

Apply an interlock to prevent input when signal input fails. Temporary storage devices are used to read input values in the project.

• Device list

Device	Description	
X1090	Module READY	Slave station X0090
X109E	A/D conversion completed flag	Slave station X009E
Y1099	Operating condition setting request	Slave station Y0099
W1590.0	[CH1] Conversion completed flag	Slave station W1090.0
W1590.1	[CH2] Conversion completed flag	Slave station W1090.1
W1591	[CH1] Analog input value	Slave station W1091
W1592	[CH2] Analog input value	Slave station W1092
D31	[CH1] Analog input value temporary area	
D32	[CH2] Analog input value temporary area	
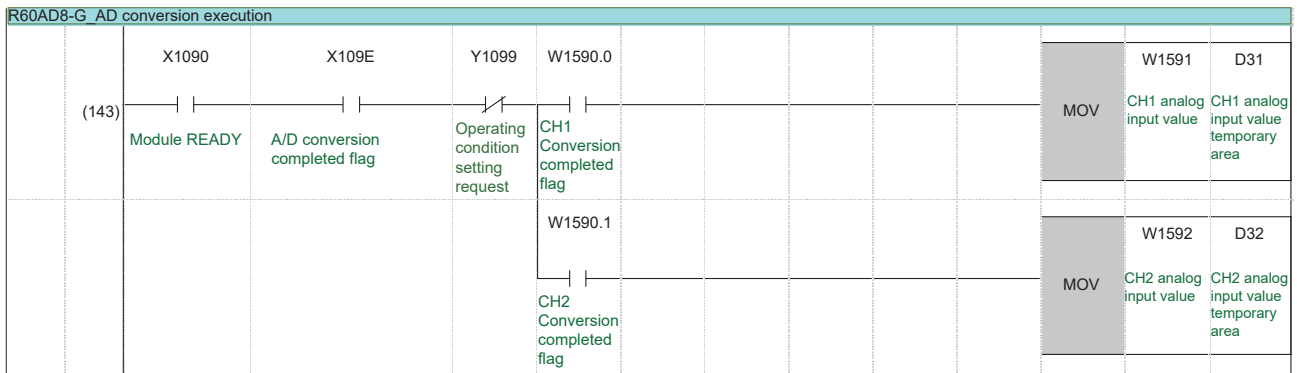
• Program example

Point

Apply an interlock to prevent the input value from being used when signal conversion fails.

Creating a project (slave station)

This section describes how to create a new slave station project using GX Works3.

Creating a new project

Create a slave station project "School_Remote".

Operating procedure
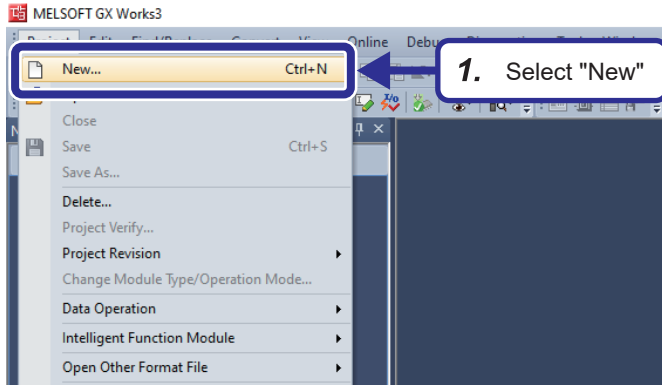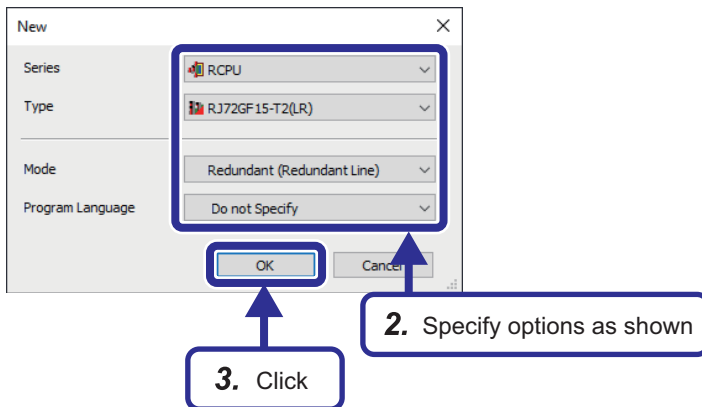
1. Select [Project] ⇨ [New].

2. When the New window is displayed, set the CPU model, operation mode, and program language as shown in the left figure.

3. Click the [OK] button.

Point

Three types of models are available for the RJ72GF15-T2 remote head module and the differences between them are as follows.
- (LR): Redundant line configuration
- (SR): Redundant slave station configuration
- None: Redundant master station configuration, single-system configuration

For details on each configuration, refer to the following. 📖 MELSEC iQ-R CC-Link IE Field Network User's Manual (Application)

Setting the module configuration window

Set the module configuration diagram in GX Works3 and define the slave station system configuration. The slave station configuration is as follows.

Slave station	R64RP	R64RP	RJ72GF15-T2	RJ72GF15-T2	RX40NC6B	RY40PT5B	RY10R2	R60TD8-G	R60DA16-G	R60AD8-G
			Slave station Network No.1 Station No.1	Slave station Network No.1 Station No.1						

Operating procedure
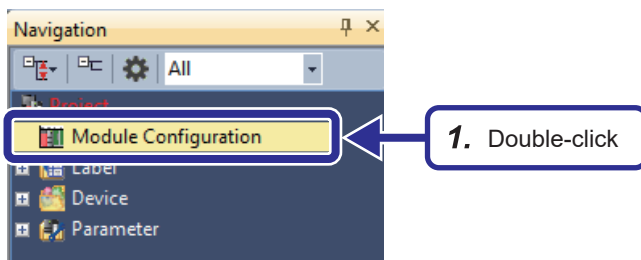
1. In the Navigation window, double-click "Module Configuration".
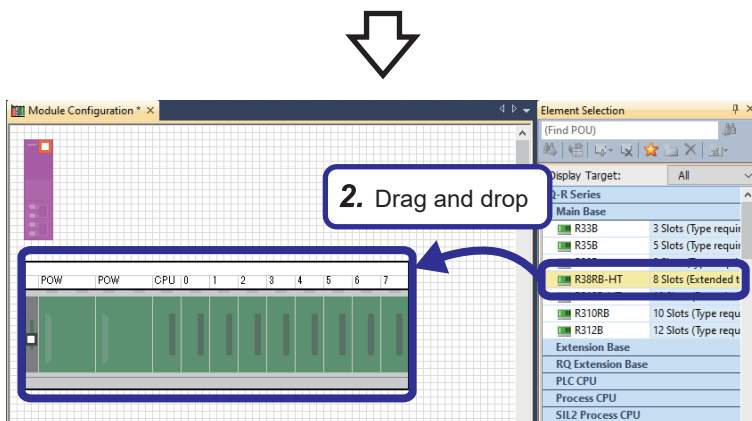
2. Select "R38RB-HT" from the "Main Base" section on the "Element Selection" window, and drag and drop it to the "Module Configuration" window.
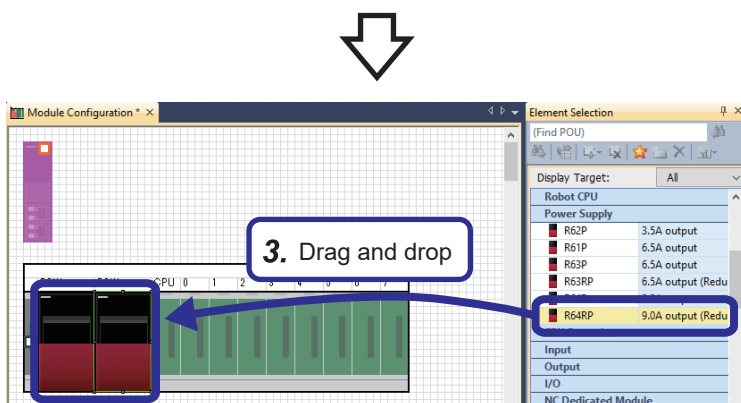
3. Select "R64RP" from the "Power Supply" section on the "Element Selection" window, and drag and drop it to two power supply slots of the R38RB-HT added to the module configuration.
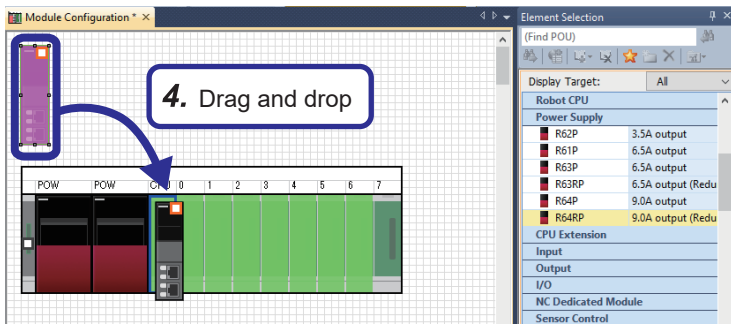
4. Drag and drop the RJ72GF15-T2(LR) displayed by default on the Module Configuration window to the CPU slot of the R38RB-HT.
Do not use remote head modules on the Element Selection window because they are intended for redundant configuration.
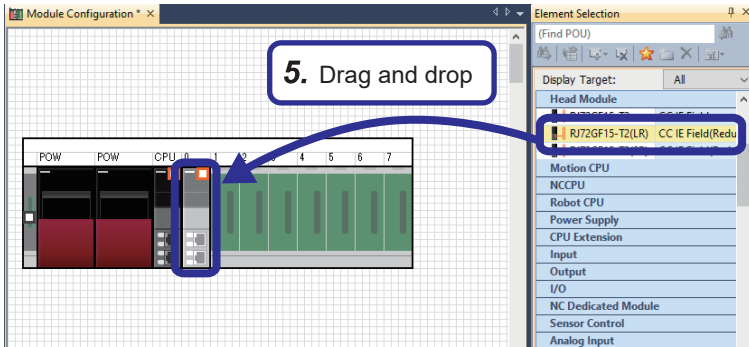
5. Select "RJ72GF15-T2(LR)" from the "Head Module" section on the "Element Selection" window, and drag and drop it to slot No.0 of the R38RB-HT.
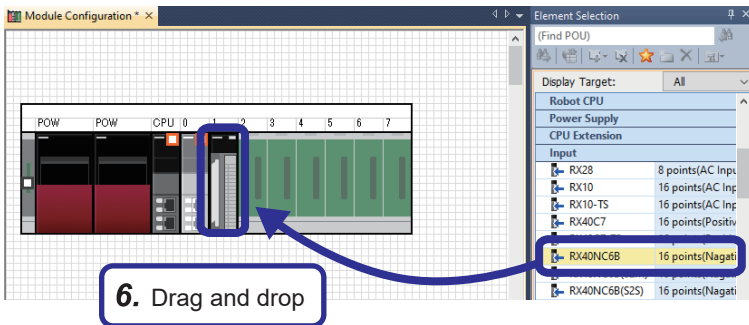
6. Select "RX40NC6B" from the "Input" section on the "Element Selection" window, and drag and drop it to slot No.1 of the R38RB-HT.
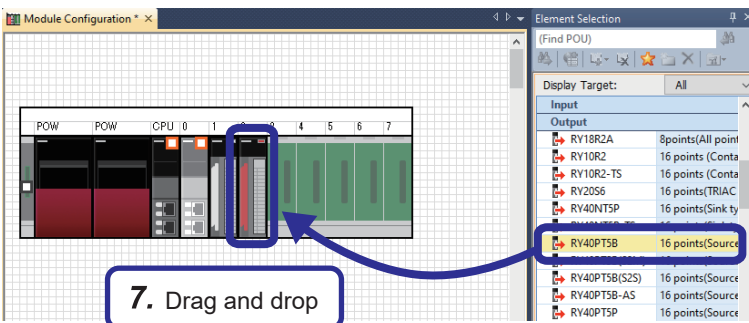
7. Select "RY40PT5B" from the "Output" section on the "Element Selection" window, and drag and drop it to slot No.2 of the R38RB-HT.

A
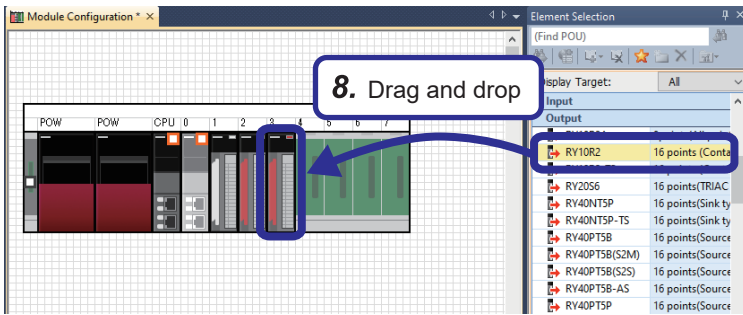
8. Select "RY10R2" from the "Output" section on the "Element Selection" window, and drag and drop it to slot No.3 of the R38RB-HT.
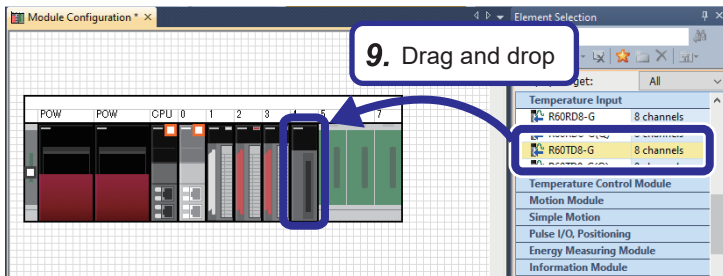
9. Select "R60TD8-G" from the "Temperature Input" section on the "Element Selection" window, and drag and drop it to slot No.4 of the R38RB-HT.
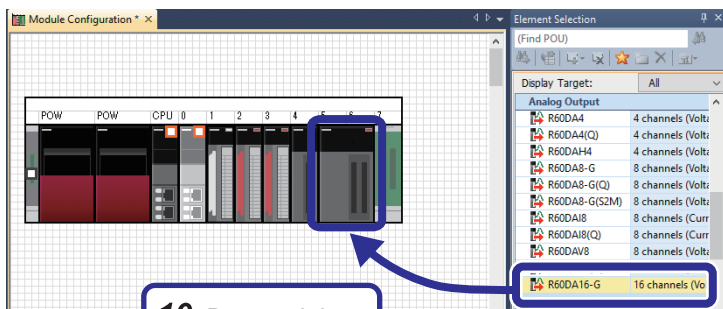
10. Select "R60DA16-G" from the "Analog Output" section on the "Element Selection" window, and drag and drop it to slot No.5 and No.6 of the R38RB-HT.
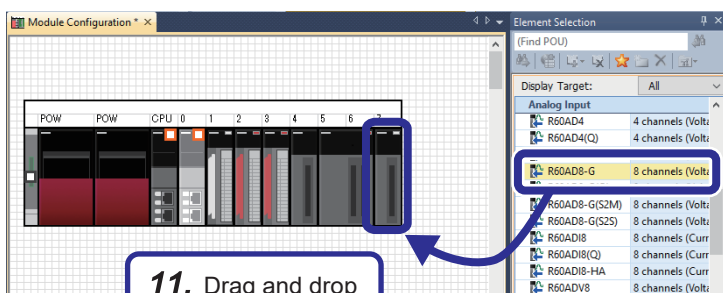
11. Select "R60AD8-G" from the "Analog Input" section on the "Element Selection" window, and drag and drop it to slot No.7 of the R38RB-HT.
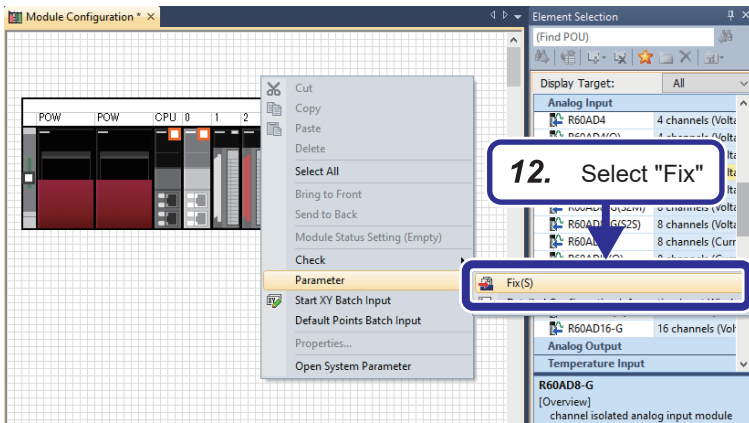
12. Right-click and select [Parameter] ⇨ [Fix] from the menu to determine the parameters.

12. Select "Fix"
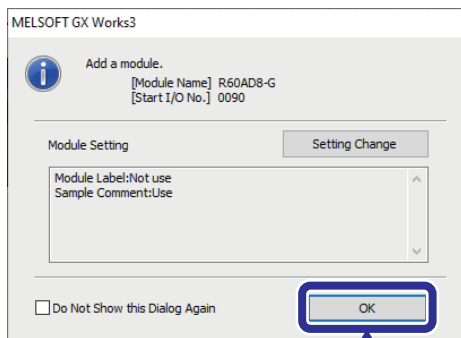
13. When the window shown on the left is displayed for each module, click the [OK] button.

13. Click
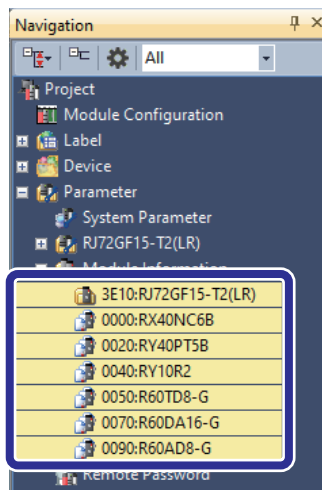
14. The module information is automatically registered on the Navigation window.

Setting network parameters

Set parameters necessary for configuring each network.

■Setting the CC-Link IE Field Network parameters

Since the parameters of the RJ71GF11-T2 for the master station are set (☞ Page 271 Setting network parameters), now set the parameters of the RJ72GF15-T2 for the slave station.
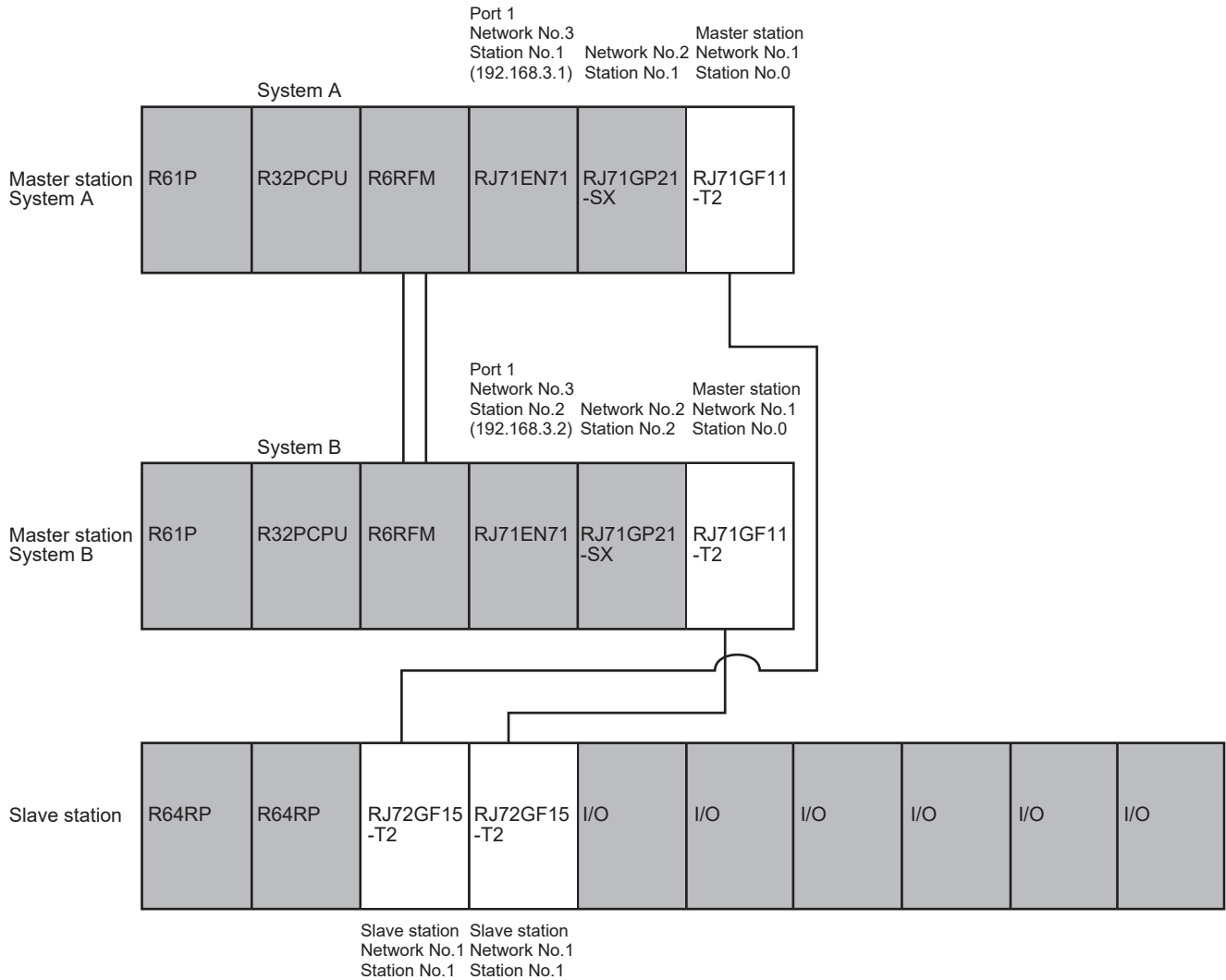
• Set the module parameters.

Set the parameters in the CC-Link IE Field remote head module.

Operating procedure
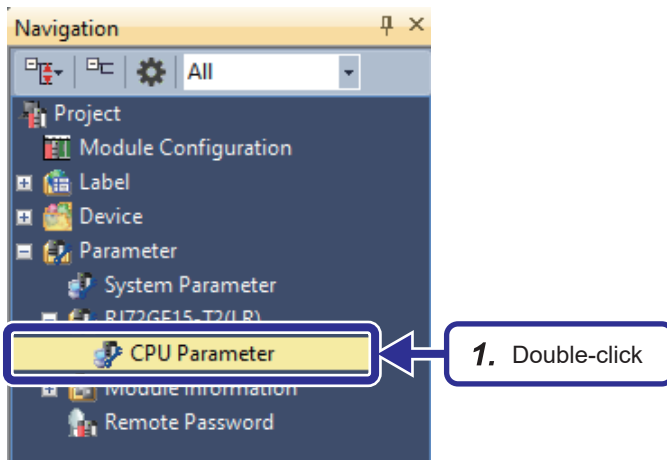
1. Double-click "CPU Parameter" of the CC-Link IE Field remote head module on the Navigation window.
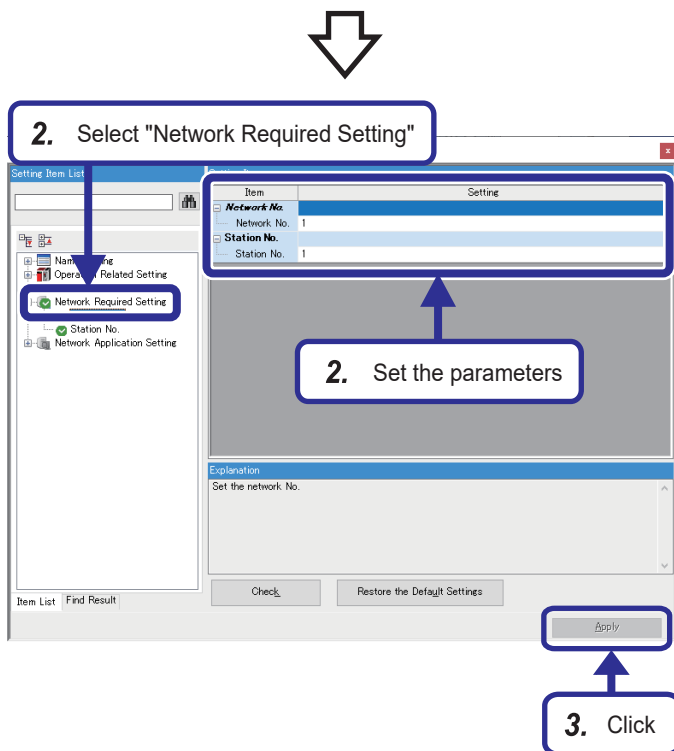
2. Set "Network Required Setting" as follows.
[Settings]
Network No.: 1
Station No.: 1

3. Click the [Apply] button.

Tracking setting

Set parameters in each input/output module using GX Works3.

In this example, the minimum settings required for signal input/output are performed based on the example settings in the following table. Modules handling analog signals require parameter setting. Modules handling 24VDC input/output and contact output operate using the default settings.

• Parameter setting examples of each module

Module name	Module type	Necessity of setting for signal input/output	Setting example
Relay output	RY10R2	Not required	—
Input with diagnostic functions	RX40NC6B		
Output with diagnostic functions	RY40PT5B		
Channel isolated thermocouple input	R60TD8-G	Required	CH1: type K thermocouple input
Channel isolated DA conversion	R60DA16-G		CH1: 4 to 20mA output CH2: 1 to 5V output
Channel isolated AD conversion	R60AD8-G		CH1: 4 to 20mA input CH2: 1 to 5V input

■Setting the thermocouple input module R60TD8-G

Set the settings necessary for signal input of the thermocouple input module R60TD8-G. In this example, a setting to input a type K thermocouple signal to CH1 will be performed.

Operating procedure
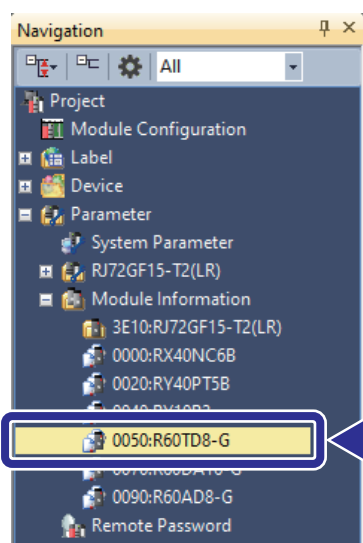
1. Double-click "0050:R60TD8-G" on the Navigation window of "School_Remote".

2. Select "Basic setting"

2. Set the parameter

2. Set the parameter
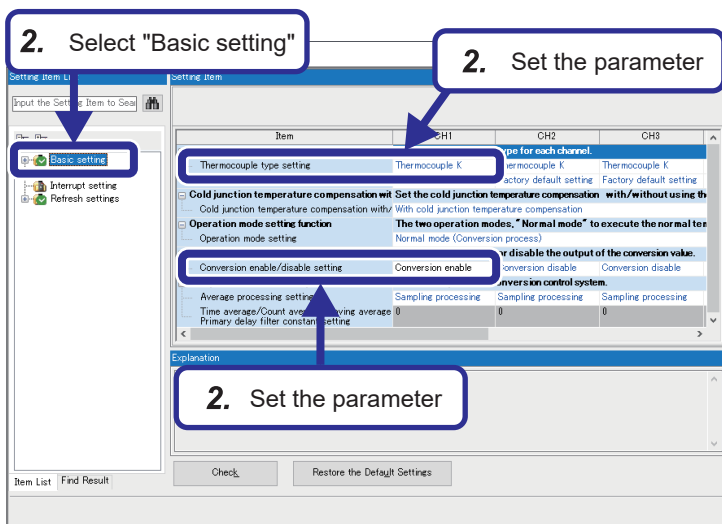
2. Set "Basic setting" as follows.
[Settings]
Thermocouple type setting: Thermocouple K
Conversion enable/disable setting: Conversion enable

3. Select "Refresh settings"

3. Set the parameters
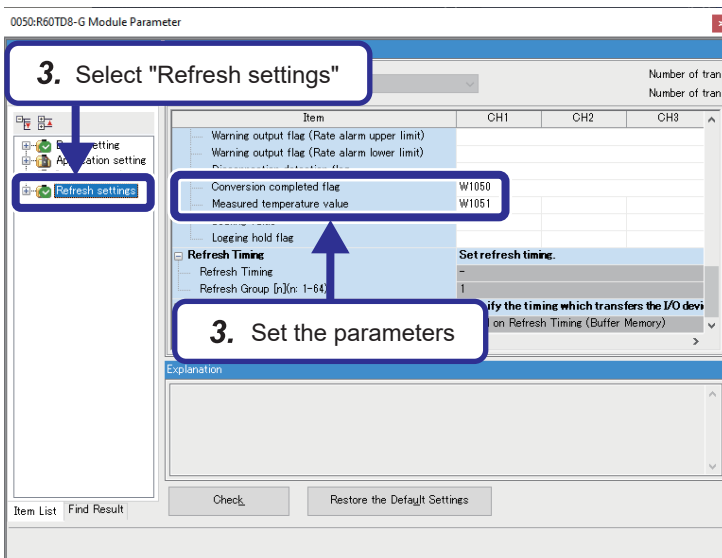
3. Set "Refresh settings" as follows.
[Settings]
Conversion completed flag: W1050
Measured temperature value: W1051

Point

For channels where "Conversion enable/disable setting" is set to "Conversion disable", input signals are not converted to digital values. Set this setting to "Conversion enable" for channels where input signals are to be handled.

Data is refreshed to the CPU module devices as follows, according to the refresh setting set in the CC-Link IE Field module.
 • Conversion complete flag (Remote head module W1050 → Process CPU W1550)
 • Temperature measured value (Remote head module W1051 → Process CPU W1551)

■Setting the DA conversion module R60DA16-G

Set the settings necessary for signal output of the DA conversion module R60DA16-G. In this example, CH1 will be set to output 4 to 20mA and CH2 will be set to output 1 to 5V. These outputs are also input to CH1 and CH2 of the AD conversion module R60AD8-G.

Operating procedure
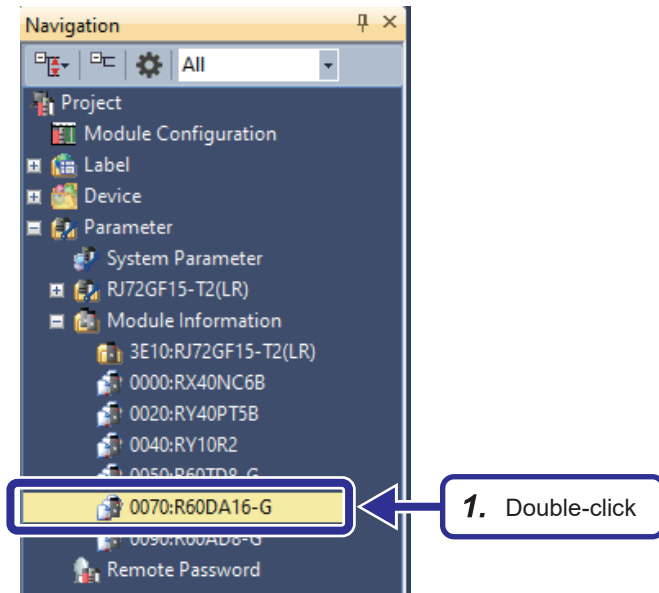
1. Double-click "0070:R60DA16-G" on the Navigation window of "School_Remote".
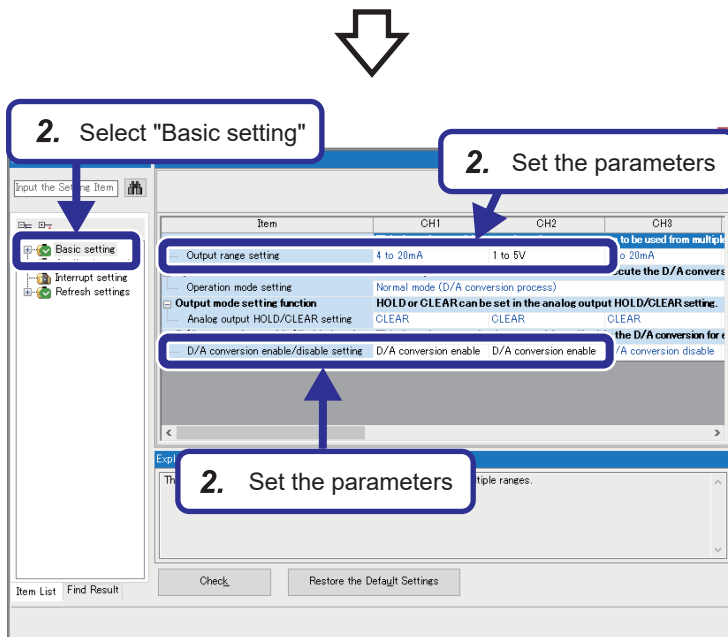
2. Set "Basic setting" as follows.
[Settings]
[CH1]
Output range setting: 4 to 20mA
D/A conversion enable/disable setting: D/A conversion enable
[CH2]
Output range setting: 1 to 5V
D/A conversion enable/disable setting: D/A conversion enable

3. Select "Refresh settings"

3. Set the parameters
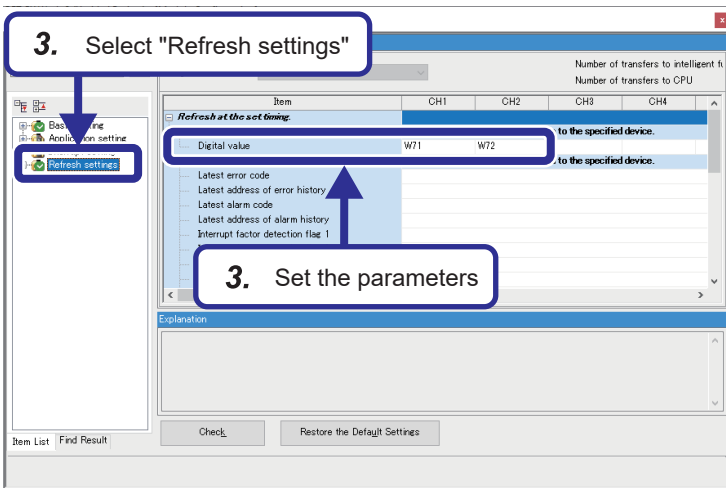
3. Set "Refresh settings" as follows.
[Settings]
[CH1]
Digital value: W71
[CH2]
Digital value: W72

Point

For channels where "D/A conversion enable/disable setting" is set to "D/A conversion disable", digital values are not converted to output signals. Set this setting to "D/A conversion enable" for channels from which signals are to be output.

Data is refreshed to the CPU module devices as follows, according to the refresh setting set in the CC-Link IE Field module.
- CH1 digital value (Remote head module W71 ← Process CPU W1071)
- CH2 digital value (Remote head module W72 ← Process CPU W1072)

■**Setting the AD conversion module R60AD8-G**

Set the settings necessary for signal input of the AD conversion module R60AD8-G. In this example, CH1 will be set to input 4 to 20mA and CH2 will be set to input 1 to 5V.

Operating procedure
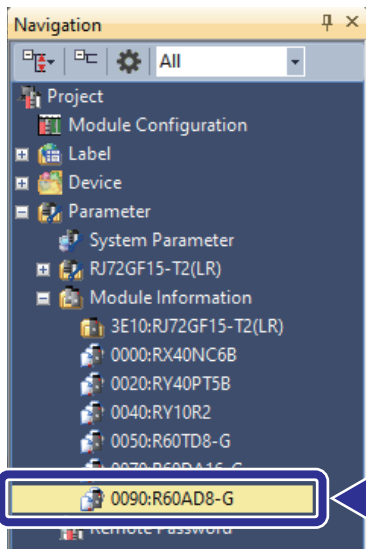
1. Double-click "0090:R60AD8-G" on the Navigation window of "School_Remote".

1. Double-click
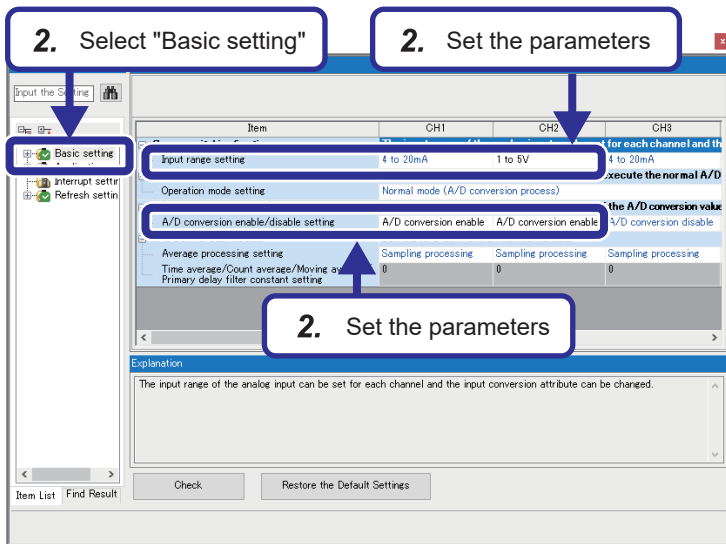
2. Set "Basic setting" as follows.

[Settings]

[CH1]

Input range setting: 4 to 20mA

A/D conversion enable/disable setting: A/D conversion enable

[CH2]

Input range setting: 1 to 5V

A/D conversion enable/disable setting: A/D conversion enable
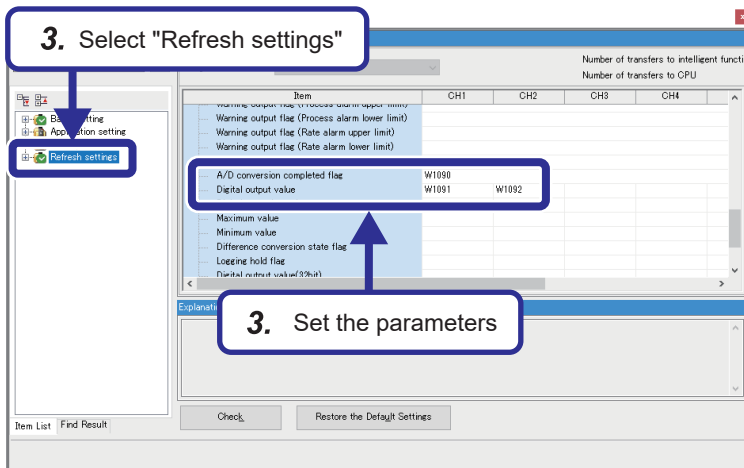
3. Set "Refresh settings" as follows.

[Settings]

A/D conversion competed flag: W1090

Digital output value (CH1): W1091

Digital output value (CH2): W1092

Point

For channels where "A/D conversion enable/disable setting" is set to "A/D conversion disable", input signals are not converted to digital values. Set this setting to "A/D conversion enable" for channels to which signals are to be input.

Data is refreshed to the CPU module devices as follows, according to the refresh setting set in the CC-Link IE Field module.

• Conversion complete flag (Remote head module W1090 → Process CPU W1590)

• CH1 digital output value (Remote head module W1091 → Process CPU W1591)

• CH2 digital output value (Remote head module W1092 → Process CPU W1592)

■Setting inter-device transfer

Set inter-device transfer using GX Works3. The inter-device transfer setting is used for transferring the following devices of the remote head module to the master station as cyclic data.

- Link special relay (SB)
- Link special register (SW)
- Special relay (SM)
- Special register (SD)
- Link register (W)

The example here will show the setting to transfer Clear error (SM50) and Operating status (SD230, SD231) of the remote head module to the master station as cyclic data. This setting will allow the CPU module in the master station to clear errors of the remote head module and to monitor the operating status of the remote head module.

- Details of the remote head module devices to be transferred

Device number	Description	Details
SM50	Error clear	• This device clears the error state when the mode transfers from off to on. • This device switches from on to off when the error clear has been completed.
SD230	Remote head module operating status (Remote head No.1) [CPU slot]	The operation information for each module is stored. b0 to b3: Operating status • 0: RUN • 2: STOP • 4: INITIALIZE • FH: RESET b4, b5: Classification • 0: Normal • 1: Minor error • 2: Medium error • 3: Major error b6: Empty
SD231	Remote head module operating status (Remote head No.2) [Slot 0]	CH1: type K thermocouple input

Operating procedure
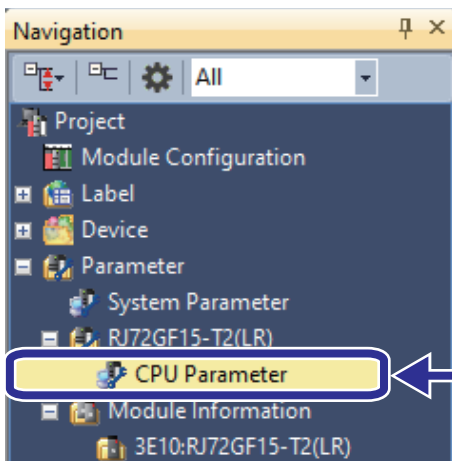
1. Double-click

1. Double-click "CPU Parameter" on the Navigation window of "School_Remote".
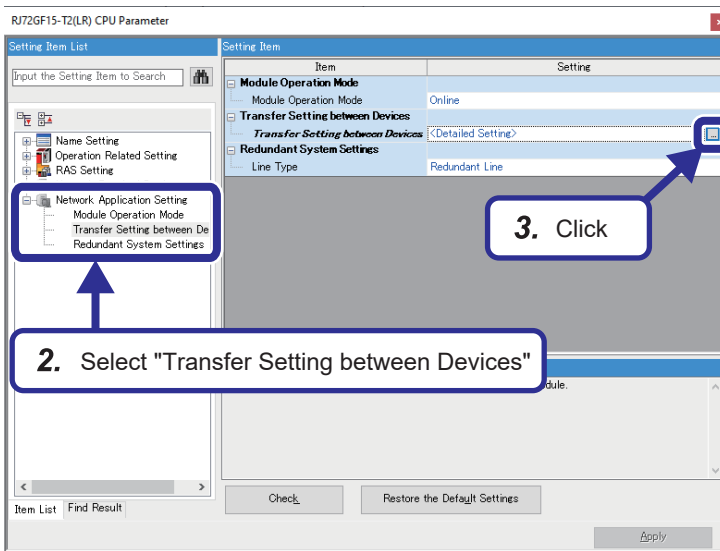
2. From the list of setting items, select "Network Application Setting" ⇨ "Transfer Setting between Devices".

3. Select "Transfer Setting between Devices" and click the ▢ button.
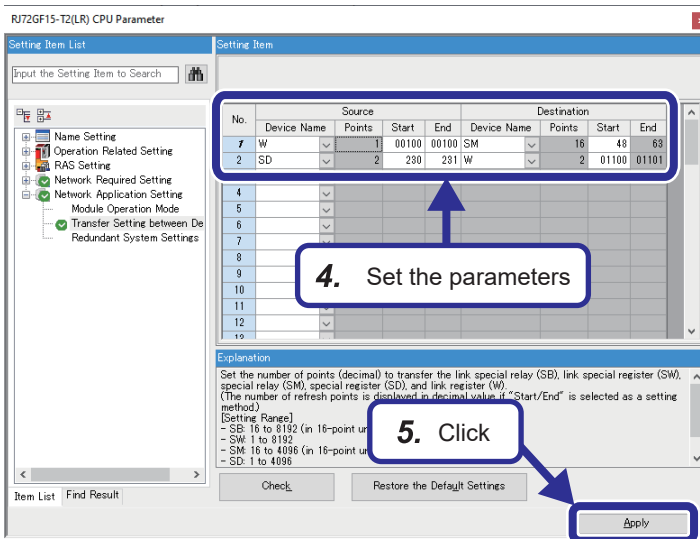
4. Set the parameters as follows.
[Settings]
[No.1]
<Source>
Device Name: W
Start: 100
End: 100
<Destination>
Device Name: SM
Start: 48
[No.2]
<Source>
Device Name: SD
Start: 230
End: 231
<Destination>
Device Name: W
Start: 1100

5. Click the [Apply] button.

Point

Data is refreshed to the CPU module devices as follows, according to the refresh setting set in the CC-Link IE Field module.
- Remote head module (W100 ← Process CPU W1100)
- Remote head module (W1100 → Process CPU W1600)
- Remote head module (W1101 → Process CPU W1601)

Appendix 7 Creating a User-Defined FB

This section describes the procedure to add a user-defined FB to the program. In this example, a user-defined FB for switching the control mode of the tag FB "TIC001" (M+M_2PIDH_) will be added to the process control program "School_Master" that was created.

Program example

Program elements to be used

Program element name	Description
R_TRIG	When the input to the input pin "Rising edge detection" (CLK) is changed from FALSE to TRUE, the output pin "Q" outputs TRUE (one scan only).
SEL	While FALSE is being input to G, the IN1 value is output from OUT. While TRUE is being input to G, the IN2 value is output from OUT.
OR	The logical sum of data input to IN1 to IN3 is output from OUT.

Program
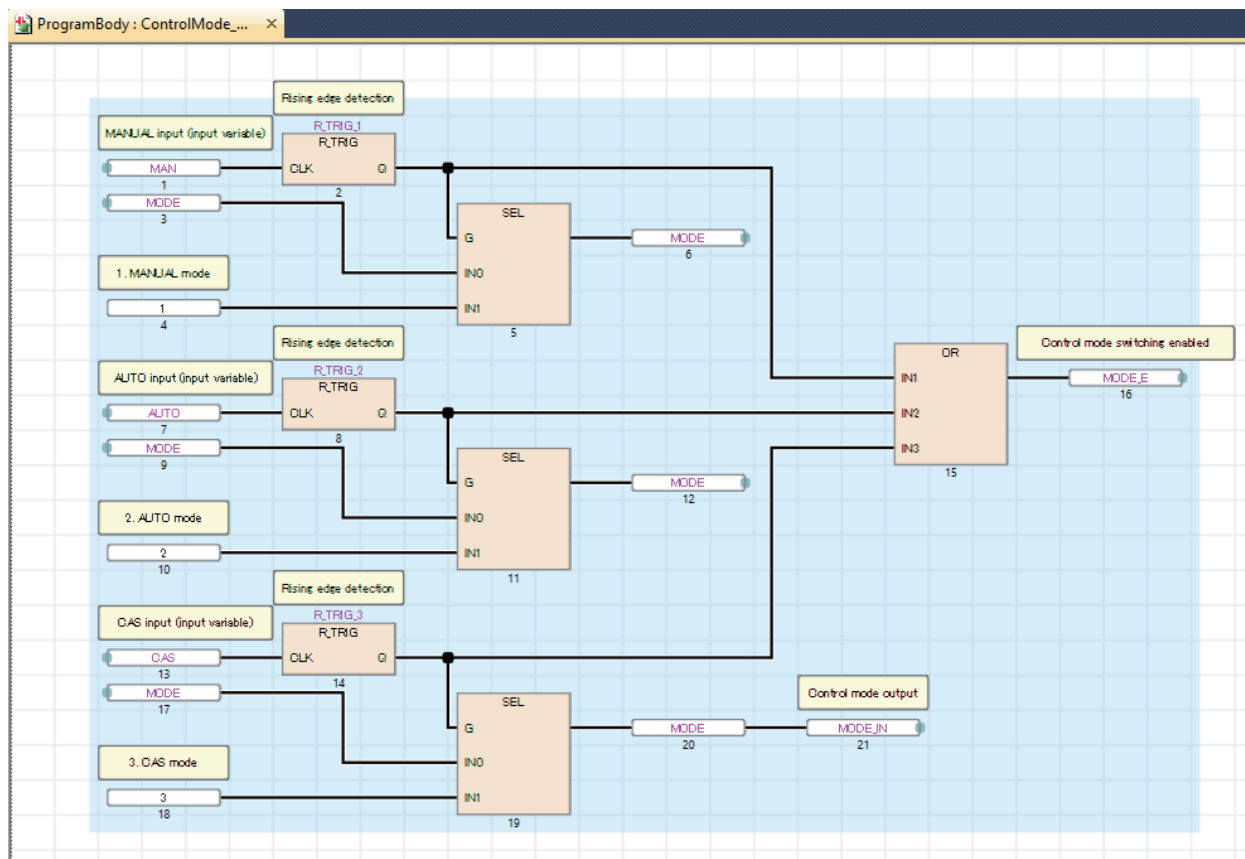
Creating a new function block

Create a new function block.

Operating procedure
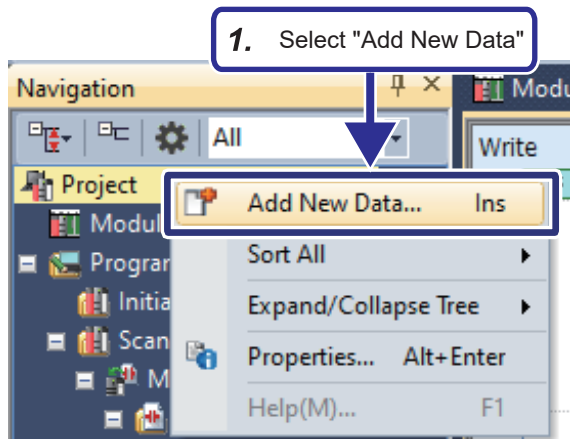
1. Right-click "Project" on the Navigation window, and select [Add New Data].
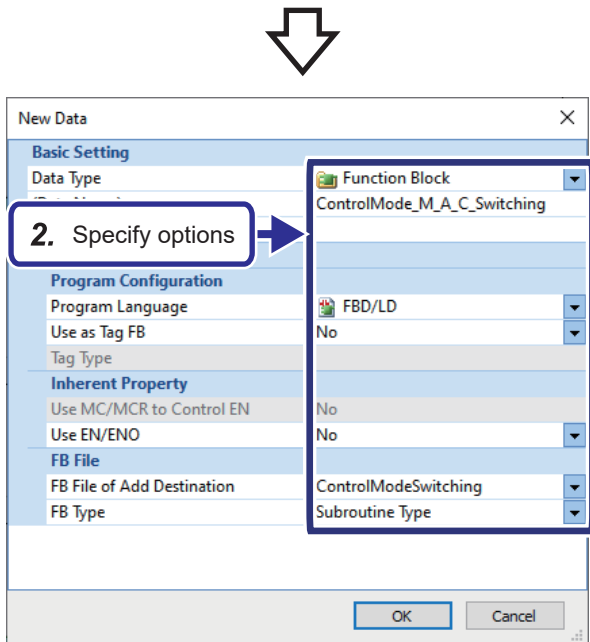
2. On the "New Data" window, specify the options as follows.
[Settings]
Data Type: Function Block
(Data Name): ControlMode_M_A_C_Switching
Program Language: FBD/LD
Use as Tag FB: No
Use EN/ENO: No
FB File of Add Destination: ControlModeSwitching
FB Type: Subroutine Type
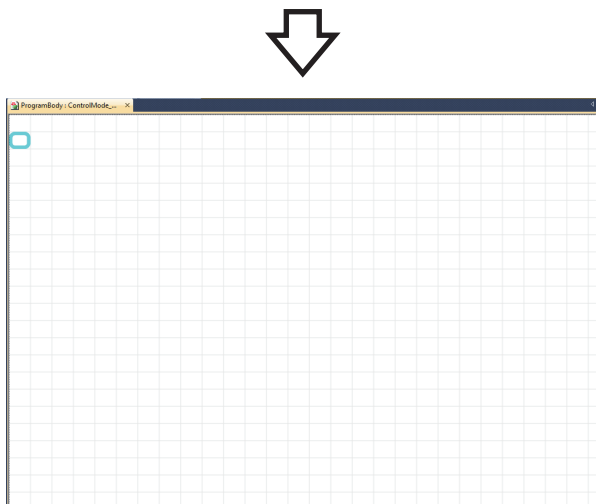
3. The program body window is displayed.

Editing a user-defined tag FB

Create a program in a user-defined FB by combining tag access FBs, standard functions, or standard FBs.
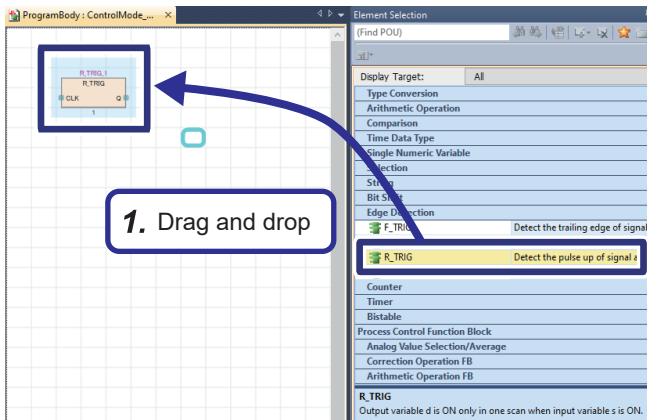
1. Select "R_TRIG" from the "Edge Detection" section on the Element Selection window, and drag and drop it to the sheet.
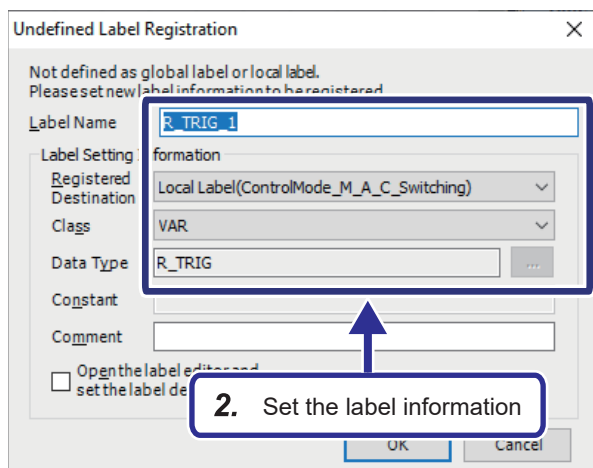
2. The "Undefined Label Registration" window appears. Configure the settings as follows.
[Settings]
Label Name: R_TRIG_1
Registered Destination: Local Label(ControlMode_M_A_C_Switching)
Class: VAR
Data Type: R_TRIG
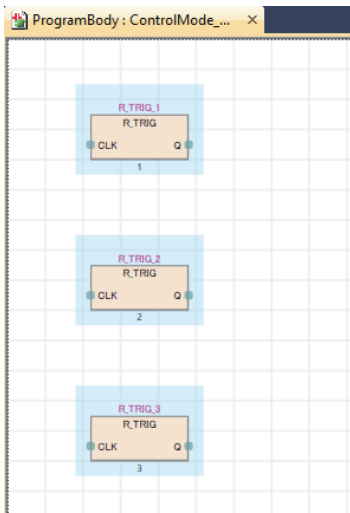
3. Repeat steps **1.** and **2.** to add three R_TRIG.
   [Settings]
   Label Name: R_TRIG_2
                R_TRIG_3
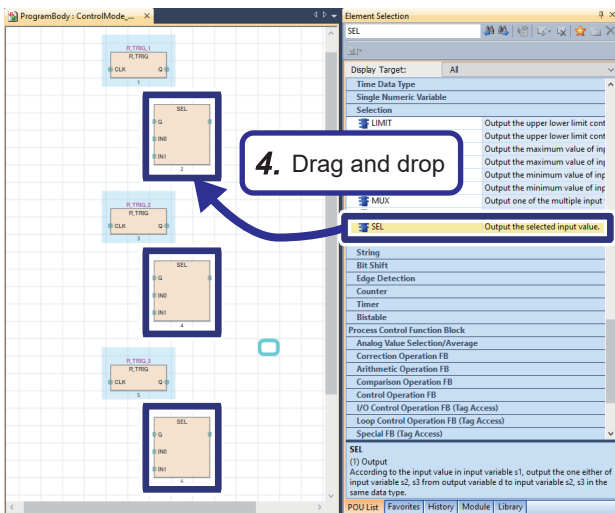
4. Select "SEL" from the "Selection" section on the Element Selection window, and drag and drop it to the three sheets as shown in the left figure.
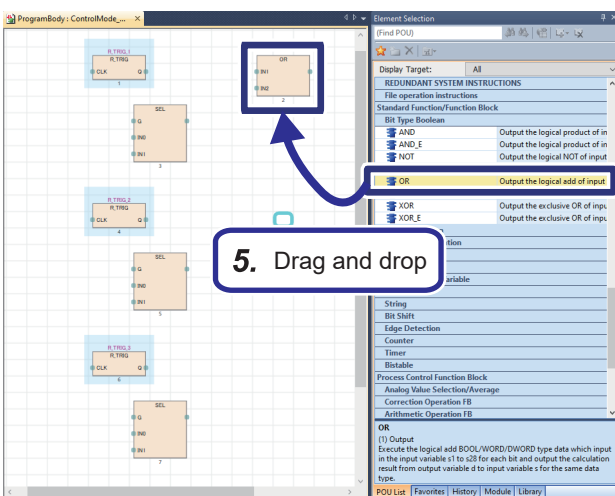
5. Select "OR" from the "Bit Type Boolean" section on the Element Selection window, and drag and drop it to the sheet.
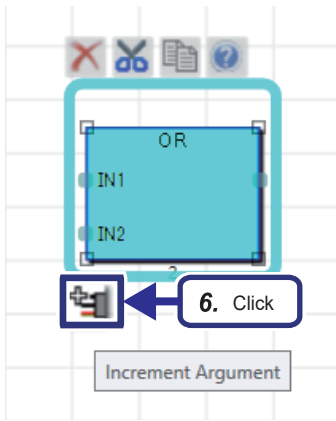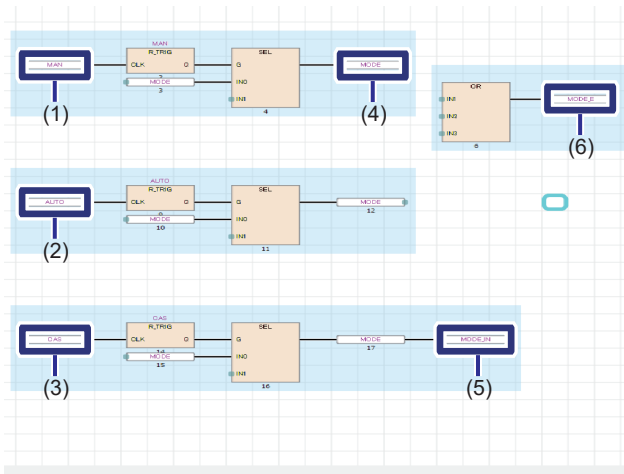
**6.** Select the pasted OR and single-click the [Increment Argument] icon.

6. Click
Increment Argument
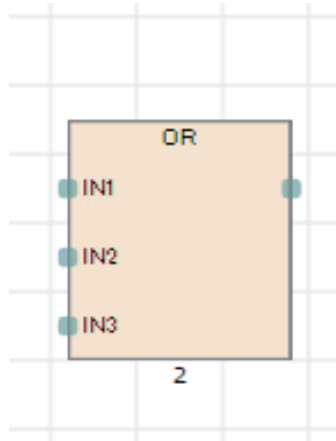
7. An input pin is added.

8. Add the following variable elements as shown in the image.
(☞ Page 99 Adding variable elements)

	Label name	Registered destination	Class	Data type
(1)	MAN	Local Label(ControlMode_M_A_C_Switching)	VAR_INPUT	Bit
(2)	AUTO	Local Label(ControlMode_M_A_C_Switching)	VAR_INPUT	Bit
(3)	CAS	Local Label(ControlMode_M_A_C_Switching)	VAR_INPUT	Bit
(4)	MODE	Local Label(ControlMode_M_A_C_Switching)	VAR	Word [signed]
(5)	MODE_IN	Local Label(ControlMode_M_A_C_Switching)	VAR_OUTPUT	Word [signed]
(6)	MODE_E	Local Label(ControlMode_M_A_C_Switching)	VAR_OUTPUT	Bit
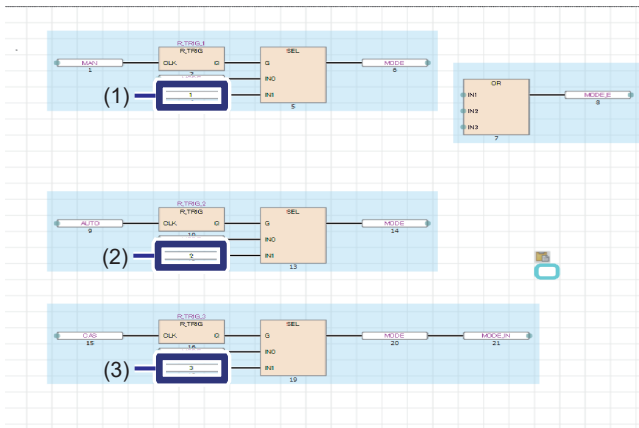
9. Paste the following constant elements as shown in the image.

(1): 1

(2): 2

(3): 3

The values set to the above constant elements indicate the modes as follows.

1: MANUAL

2: AUTO

3: CASCADE

**10.** Connect elements and add comments, referring to "☞ Page 303 Program".

Point

For constant elements, the data type of the constant value is not yet determined when the constant value is input. The data type is determined when the constant element is connected to an FBD element with a connection line. The data type of the FBD element connected with the connection line is applied as the data type of the constant value.

# Creating a program using a user-defined FB

Add a created function block "ControlMode_M_A_C_Switching" to the program created in "☞ Page 96 Programming".

Operating procedure

1. Open "ProgramBody"

1. On the Navigation window, select "Scan" ⇨ "PROCESS" ⇨ "ProgPou1" to open the program body.
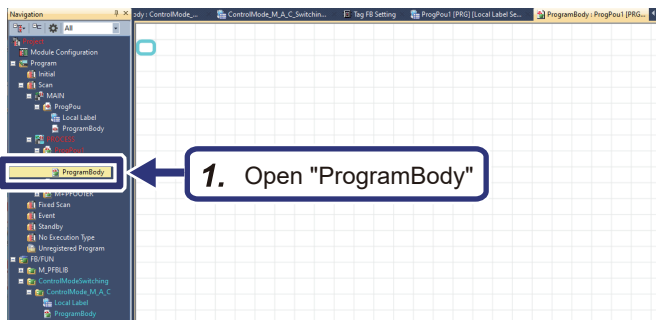

2. Drag and drop

2. On the Element Selection window, select "Project" from the "Display Target" drop-down menu. Select "ControlMode_M_A_C_Switching" from "Function Block" and drag and drop it to the sheet.
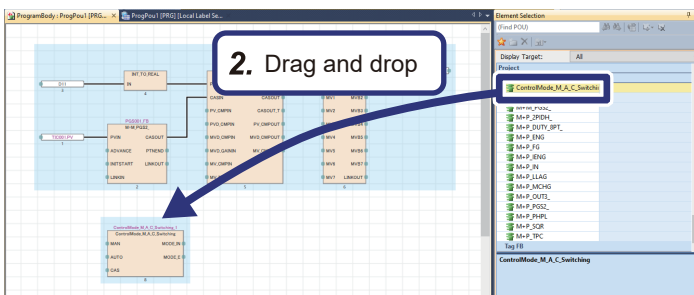
3. Set the label information

3. The "Undefined Label Registration" window appears. Configure the settings as follows.
[Settings]
Label Name: ControlMode_M_A_C_Switching_1
Registered Destination: Local Label(ProgPou1)
Class: VAR
Data Type: ControlMode_M_A_C_Switching
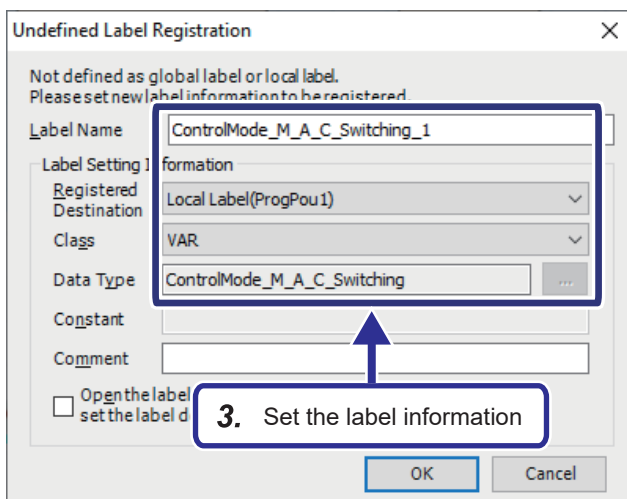
A
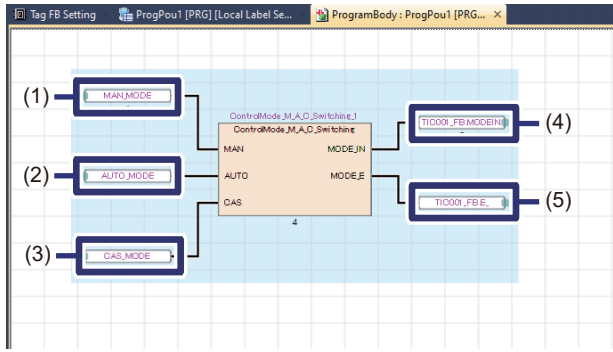
4. Paste the following variable elements and connect them as shown in the left figure.

	Label name	Registered destination	Class	Data type
(1)	MAN_MODE	Local Label(ProgPou1)	VAR	Bit
(2)	AUTO_MODE	Local Label(ProgPou1)	VAR	Bit
(3)	CAS_MODE	Local Label(ProgPou1)	VAR	Bit
(4)	TIC001_FB.MODEIN	—	—	—
(5)	TIC001_FB.E_	—	—	—

5. Convert the program. (☞ Page 106 Converting the program)

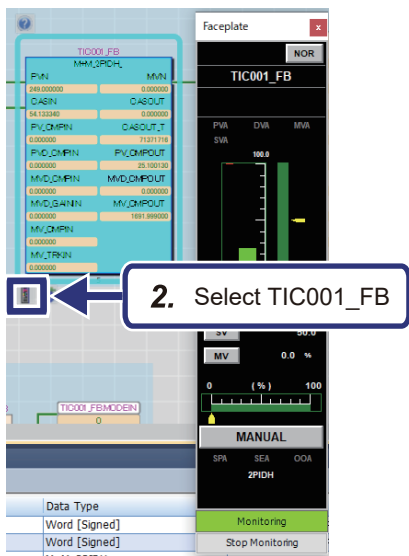**6.** Write the project. (☞ Page 108 Writing a Project)

Checking the operation

Switch the control mode with the created program.

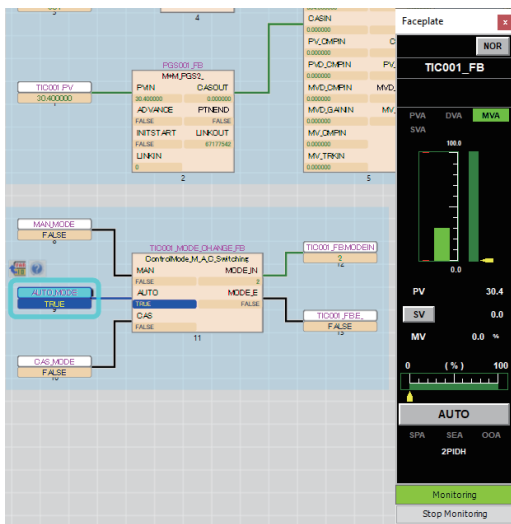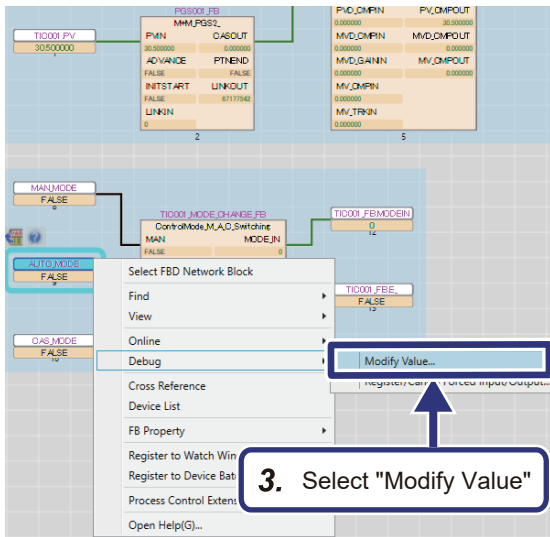## Operating procedure
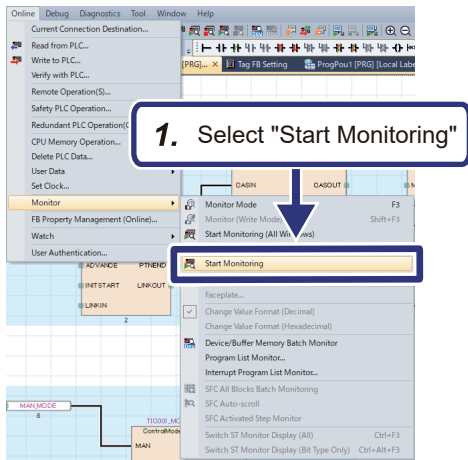
1. Select [Online] ⇨ [Monitor] ⇨ [Start Monitoring] on the menu bar.

2. Select the tag FBTIC001_FB to display the faceplate.

A

3. Select the newly added variable element "AUTO_MODE", right-click and select [Debug], and then select [Modify Value].

3. Select "Modify Value"

4. "AUTO_MODE" changes from FALSE to TRUE, and the control mode displayed on the faceplate changes from MANUAL to AUTO.
In addition, the control mode of "CAS_MODE" changes to CASCADE, and the control mode of "MAN_MODE" changes to MANUAL.

Mitsubishi Programmable Controllers Training Manual
MELSEC iQ-R Process Control Basic Course

MODEL	SCHOOL-R PROCESS-E
MODEL CODE	13JW60
SH(NA)-082349ENG-A(2008)MEE	

MITSUBISHI ELECTRIC CORPORATION

When exported from Japan, this manual does not require application to the
Ministry of Economy, Trade and Industry for service transaction permission.

Specifications subject to change without notice.