

# ***FATEC***

---

## **Mitsubishi Programmable Controllers Training Manual MELSEC iQ-R Series Advanced Course (for GX Works3)**

# SAFETY PRECAUTION

---

(Always read these instructions before using the products.)

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.
  - Do not touch the movable portion.
- 

### **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 (such as X/Y table) 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.
-



# MEMO

---

# REVISIONS

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

Revision date	*Manual number	Description
February 2016	SH(NA)-081897ENG-A	First edition

This manual confers no industrial property rights of any other kind, nor does it confer any patent licenses. Mitsubishi Electric Corporation cannot be held responsible for any problems involving industrial property rights which may occur as a result of using the contents noted in this manual.

© 2016 MITSUBISHI ELECTRIC CORPORATION

# MEMO

---

# CONTENTS

SAFETY PRECAUTION .....	A - 1
REVISIONS .....	A - 3
INTRODUCTION .....	A - 9
RELEVANT MANUALS .....	A - 9

## CHAPTER 1 BASIC INSTRUCTIONS -PART 3- 1 - 1

<b>1.1 Subroutine Program .....</b>	<b>1 - 1</b>
1.1.1 [CALL(P)] (Execution of a subroutine program)	
[RET] (End of a subroutine program) .....	1 - 2
1.1.2 [FEND] (End of a main routine program) .....	1 - 5
<b>1.2 Index Register (Z/LZ) .....</b>	<b>1 - 7</b>
<b>1.3 File Register (R/ZR) .....</b>	<b>1 - 11</b>
<b>1.4 Latch Function .....</b>	<b>1 - 16</b>
1.4.1 Latch types .....	1 - 16
1.4.2 Devices that can be latched .....	1 - 16
1.4.3 Optimal assignment of device points .....	1 - 17

## CHAPTER 2 FILE-BASED PROGRAM MANAGEMENT AND PROGRAM EXECUTION MANAGEMENT 2 - 1

<b>2.1 File-based Management .....</b>	<b>2 - 1</b>
2.1.1 CPU built-in memory and SD memory card .....	2 - 3
<b>2.2 Program Execution Management .....</b>	<b>2 - 9</b>
2.2.1 Program execution types .....	2 - 9
2.2.2 Initial execution type program .....	2 - 11
2.2.3 Scan execution type program .....	2 - 12
2.2.4 Standby type program .....	2 - 13
2.2.5 Fixed scan execution type program .....	2 - 14
2.2.6 Event execution type program .....	2 - 17

## CHAPTER 3 BASIC OPERATIONS OF GX Works3 (MULTIPLE PROGRAMS) 3 - 1

<b>3.1 Multiple Programs .....</b>	<b>3 - 1</b>
3.1.1 Creating multiple programs .....	3 - 2
3.1.2 Creating control programs .....	3 - 9
3.1.3 Setting parameters .....	3 - 11
<b>3.2 Monitoring .....</b>	<b>3 - 18</b>
3.2.1 Program list monitor .....	3 - 18

## CHAPTER 4 FUNCTIONS OF RCPU 4 - 1

<b>4.1 Constant Scan Function .....</b>	<b>4 - 1</b>
<b>4.2 Device Initial Value .....</b>	<b>4 - 3</b>
4.2.1 Setting values in the device memory .....	4 - 4
4.2.2 Setting a device initial value file name .....	4 - 8
4.2.3 Checking the operation of device initial values .....	4 - 11
<b>4.3 Security .....</b>	<b>4 - 12</b>
4.3.1 Creating a security key .....	4 - 13
4.3.2 Registering a security key in a program file .....	4 - 15
4.3.3 Exporting a security key .....	4 - 17
4.3.4 Deleting a security key .....	4 - 18

4.3.5	Importing a security key. ....	4 - 19
<b>4.4</b>	<b>Sampling Trace</b> .....	<b>4 - 20</b>
4.4.1	Realtime monitor function .....	4 - 20
4.4.2	Memory dump function .....	4 - 21
4.4.3	Data logging function. ....	4 - 22

---

## **CHAPTER 5 PROGRAMMING FOR INTELLIGENT FUNCTION MODULES** **5 - 1**

---

<b>5.1</b>	<b>Intelligent Function Module</b> .....	<b>5 - 1</b>
<b>5.2</b>	<b>Data Communication Between Intelligent Function Modules and the CPU Module</b> .....	<b>5 - 2</b>
5.2.1	I/O signals to the CPU module .....	5 - 3
5.2.2	Data communication with intelligent function modules .....	5 - 4
<b>5.3</b>	<b>Methods of the Communication with Intelligent Function Modules.</b> .....	<b>5 - 5</b>
5.3.1	Communications with intelligent function modules .....	5 - 5
<b>5.4</b>	<b>Configuration of the Intelligent Function Modules for Demonstrations</b> .....	<b>5 - 6</b>
<b>5.5</b>	<b>Creating the Module Configuration</b> .....	<b>5 - 7</b>
5.5.1	Arranging objects. ....	5 - 7
<b>5.6</b>	<b>R60AD4 Analog-digital Converter Module</b> .....	<b>5 - 11</b>
5.6.1	Part names .....	5 - 11
5.6.2	A/D conversion characteristics .....	5 - 12
5.6.3	List of I/O signals and buffer memory assignment .....	5 - 13
5.6.4	Setting intelligent function module data. ....	5 - 19
5.6.5	Exercise with the demonstration machine. ....	5 - 22
<b>5.7</b>	<b>R60DA4 Digital-analog Converter Module</b> .....	<b>5 - 23</b>
5.7.1	Part names .....	5 - 23
5.7.2	D/A conversion characteristics .....	5 - 24
5.7.3	List of I/O signals and buffer memory assignment .....	5 - 25
5.7.4	Setting intelligent function module data. ....	5 - 30
5.7.5	Exercise with the demonstration machine. ....	5 - 33

---

## **CHAPTER 6 MULTIPLE CPU SYSTEM** **6 - 1**

---

<b>6.1</b>	<b>Overview.</b> .....	<b>6 - 1</b>
<b>6.2</b>	<b>Features</b> .....	<b>6 - 1</b>
<b>6.3</b>	<b>Differences with a Single CPU System</b> .....	<b>6 - 3</b>
6.3.1	Mounting positions of modules .....	6 - 3
6.3.2	I/O number assignment. ....	6 - 4
6.3.3	Data communications among CPU modules and controlled modules. ....	6 - 5
<b>6.4</b>	<b>Data Communications Among CPU Modules in a Multiple CPU System.</b> .....	<b>6 - 6</b>
6.4.1	Memory to be used .....	6 - 6
6.4.2	Communication through refresh .....	6 - 7
6.4.3	Communication through direct access .....	6 - 11
<b>6.5</b>	<b>Starting Up Multiple CPU System</b> .....	<b>6 - 13</b>
6.5.1	Procedure for starting up a multiple CPU system .....	6 - 13
6.5.2	System configuration of the demonstration machine. ....	6 - 15
6.5.3	Creating a program for the CPU No.1 .....	6 - 16
6.5.4	Creating the module configuration. ....	6 - 17
6.5.5	Parameter setting of the CPU No.1. ....	6 - 21
6.5.6	Creating a program for the CPU No.2 .....	6 - 28
6.5.7	Parameter setting of the CPU No.2. ....	6 - 29
6.5.8	Writing data to the CPU modules .....	6 - 37
6.5.9	Operation check .....	6 - 41

<b>CHAPTER 7</b>	<b>NEW FUNCTIONS OF MELSEC iQ-R/GX Works3</b>	<b>7 - 1</b>
7.1	Features of MELSEC iQ-R. ....	7 - 1
7.2	Differences Between the MELSEC-Q Series and the MELSEC iQ-R Series. ....	7 - 4
7.3	Functions of GX Works3 ....	7 - 5
<b>APPENDICES</b>		<b>App. - 1</b>
<b>Appendix 1</b>	<b>System Configuration of the Demonstration Machine. ....</b>	<b>App. - 1</b>
<b>Appendix 2</b>	<b>Creating Ladders with GX Works3. ....</b>	<b>App. - 3</b>
<b>Appendix 3</b>	<b>Offset/Gain Setting ....</b>	<b>App. - 7</b>
Appendix 3.1	Offset value and gain value ....	App. - 12
<b>Appendix 4</b>	<b>Specifications of the A/D Converter Module and D/A Converter Module. ....</b>	<b>App. - 18</b>
Appendix 4.1	A/D converter module ....	App. - 18
Appendix 4.2	D/A converter module ....	App. - 21
<b>Appendix 5</b>	<b>Timer, Counter ....</b>	<b>App. - 24</b>
Appendix 5.1	Timer ....	App. - 24
Appendix 5.2	Counter ....	App. - 25
<b>Appendix 6</b>	<b>Inline ST ....</b>	<b>App. - 26</b>
Appendix 6.1	Using the inline ST. ....	App. - 27
Appendix 6.2	Precautions for using the inline ST ....	App. - 29
<b>Appendix 7</b>	<b>Battery. ....</b>	<b>App. - 30</b>
<b>Appendix 8</b>	<b>Real Number (Floating-point Data). ....</b>	<b>App. - 31</b>
<b>Appendix 9</b>	<b>Changing Input Response Time ....</b>	<b>App. - 34</b>

# MEMO

---

# INTRODUCTION

This textbook describes a programmable controller, the methods of editing a program with GX Works3, sequence instructions, and application instructions for helping users to understand the programming for the MELSEC iQ-R series.

## RELEVANT MANUALS

Manual name [manual number]	Description	Available form
MELSEC iQ-R CPU Module User's Manual (Startup) [SH-081263ENG]	Performance specifications, procedures before operation, and troubleshooting of the CPU module	e-Manual EPUB PDF
MELSEC iQ-R CPU Module User's Manual (Application) [SH-081264ENG]	Memory, functions, devices, and parameters of the CPU module	e-Manual EPUB PDF
MELSEC iQ-R Programming Manual (Program Design) [SH-081265ENG]	Program specifications such as of ladder programs and ST programs, and labels	e-Manual EPUB PDF
MELSEC iQ-R Programming Manual (Instructions, Standard Functions/ Function Blocks) [SH-081266ENG]	Instructions for the CPU module, instructions dedicated for intelligent function modules, and standard functions/function blocks	e-Manual EPUB PDF
MELSEC iQ-R Module Configuration Manual [SH-081262ENG]	System configuration, specifications, mounting, wiring, and maintenance and inspection required for using the MELSEC iQ-R series programmable controller	e-Manual EPUB PDF
GX Works3 Operating Manual [SH-081215ENG]	System configuration of GX Works3, parameter setting, and operation method of the online function	e-Manual EPUB PDF
MELSEC iQ-R Analog-Digital Converter Module User's Manual (Startup) [SH-081232ENG]	Performance specifications, procedures before operation, wiring, and operation examples required for using the MELSEC iQ-R analog-digital converter module	e-Manual EPUB PDF
MELSEC iQ-R Analog-Digital Converter Module User's Manual (Application) [SH-081233ENG]	Functions, parameter settings, and troubleshooting required for using the MELSEC iQ-R analog-digital converter module	e-Manual EPUB PDF
MELSEC iQ-R Digital-Analog Converter Module User's Manual (Startup) [SH-081235ENG]	Performance specifications, procedures before operation, wiring, and operation examples required for using the MELSEC iQ-R digital-analog converter module	e-Manual EPUB PDF
MELSEC iQ-R Digital-Analog Converter Module User's Manual (Application) [SH-081237ENG]	Functions, parameter settings, and troubleshooting required for using the MELSEC iQ-R digital-analog converter module	e-Manual EPUB PDF



e-Manual refers to the Mitsubishi 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.



# MEMO

---

# 1 BASIC INSTRUCTIONS -PART 3-

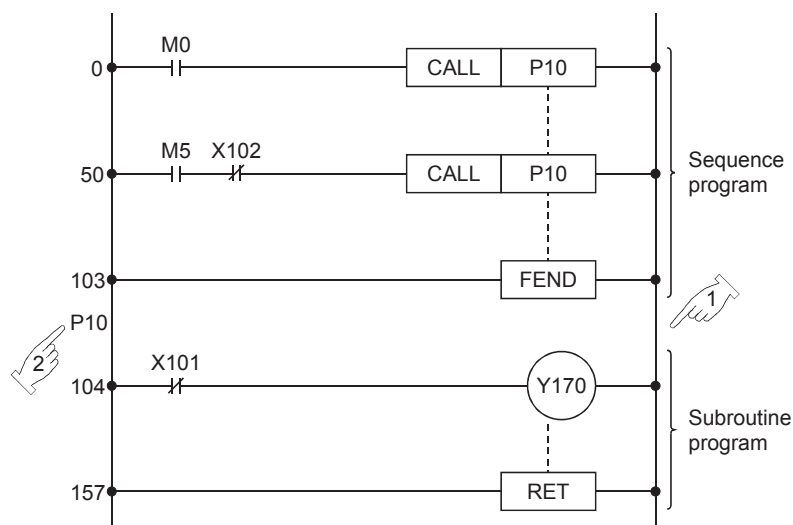
For the basic instructions part 1 and 2, refer to the MELSEC iQ-R Series Basic Course (for GX Works3).

## 1.1 Subroutine Program

### Point

- This section describes the concept of subroutine.
- This section describes how to create a subroutine program with a programmable controller.
- This section describes that a programmable controller repeatedly executes operations (scans a program).

# 1.1.1 [CALL(P)] (Execution of a subroutine program) [RET] (End of a subroutine program)



- The above program shows the basic usage of the CALL and RET instructions used to execute a subroutine program.

Without such a program, an error occurs and a programmable controller will stop its operation.



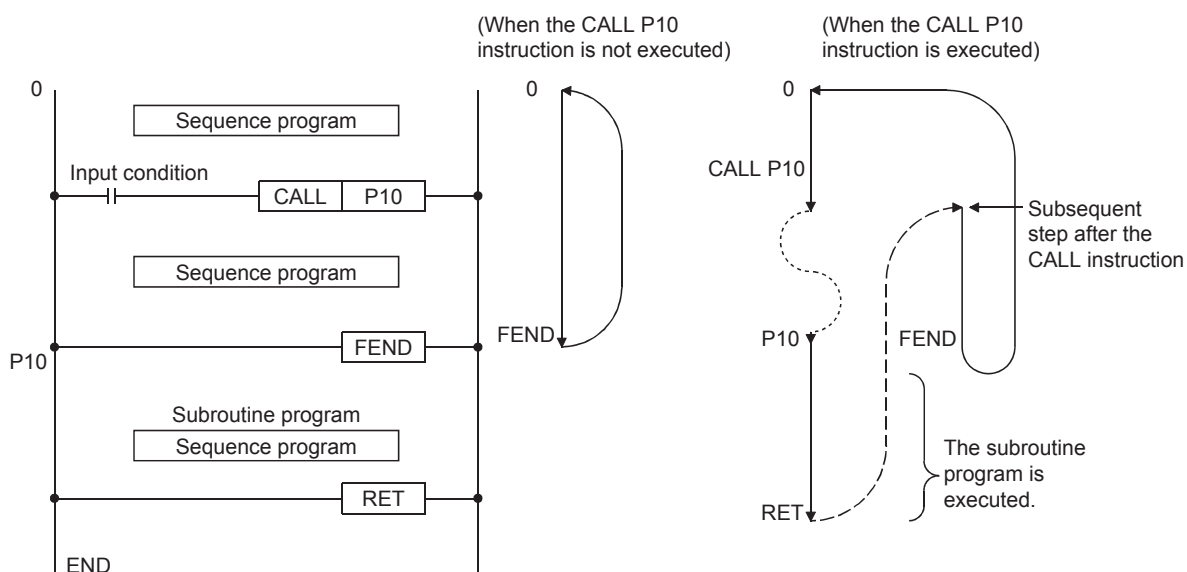
- A subroutine program has ladders containing the processing that needs to be executed more than once. The program starts with a pointer P□ and ends with the RET instruction.



- Set the pointer (P) number within the following range. (This number is also used by the CJ and SCJ instructions.)

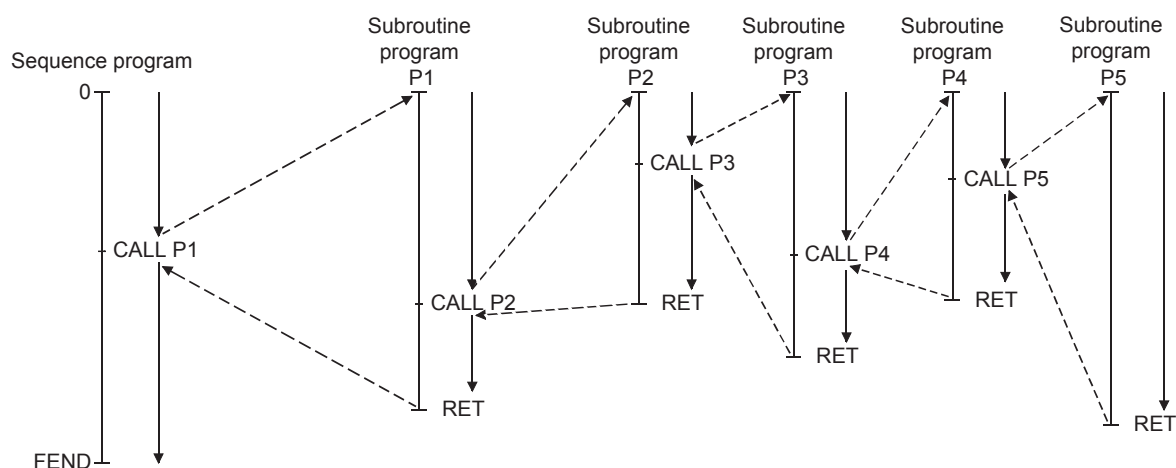
Setting range	Default
<ul style="list-style-type: none"> <li>R04CPU, R04ENCPU, R08CPU, R08ENCPU, R08PCPU, R08SFCPU, R16CPU, R16ENCPU, R16PCPU, R16SFCPU, R32CPU, R32ENCPU, R32PCPU, R32SFCPU: 0 to 16384 points (in increments of one point)</li> </ul>	4096 points
<ul style="list-style-type: none"> <li>R120CPU, R120ENCPU, R120PCPU, R120SFCPU: 0 to 32768 points (in increments of one point)</li> </ul>	8192 Points

- The following figures show execution of a subroutine program.

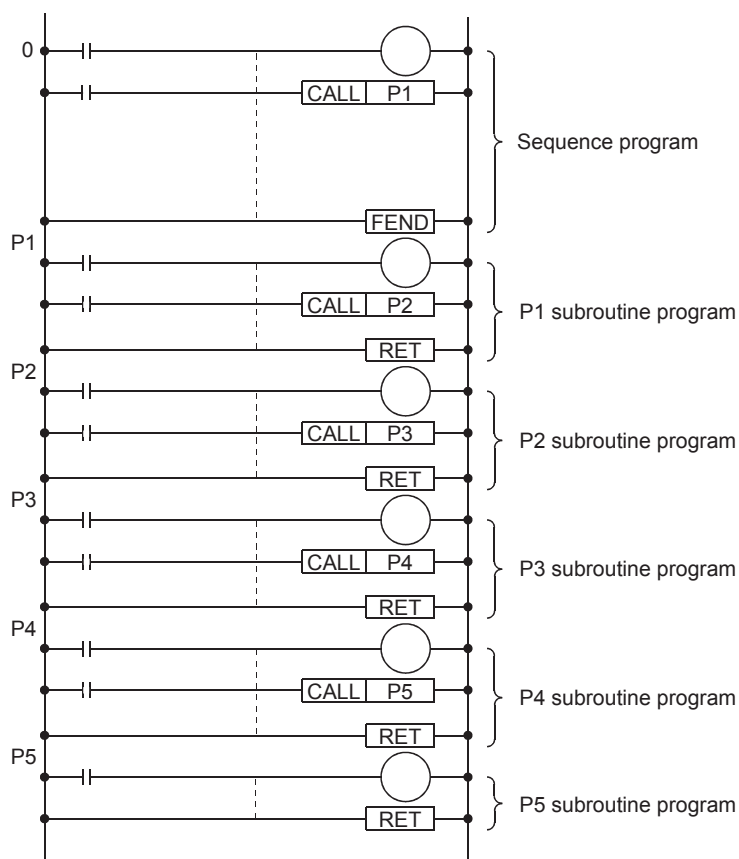


## Nesting

- Up to 16 CALL(P) instructions can be nested.



The following ladder program shows the above nesting structure.



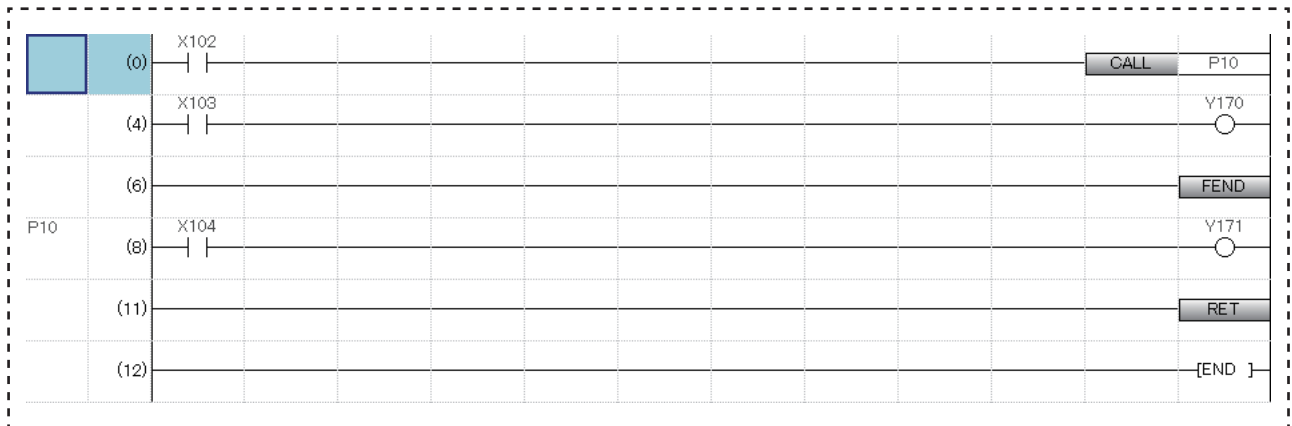
### CALL(P)

Operand	Bit	Word	Double word	Indirect specification	Constant			Others (P)	Number of basic steps
	X, Y, M, L, SM, F, B, SB, FX, FY	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	LT, LST, LC		K, H	E	\$		
(P)	—	—	—	—	—	—	—	—	CALL: 3 CALLP: 4

## ■Ladder example

Create the following ladder program with GX Works3 and write it to the programmable controller CPU of the demonstration machine. Then, check execution of the CALL and RET instructions.

Project name	EX1
Program name	MAIN

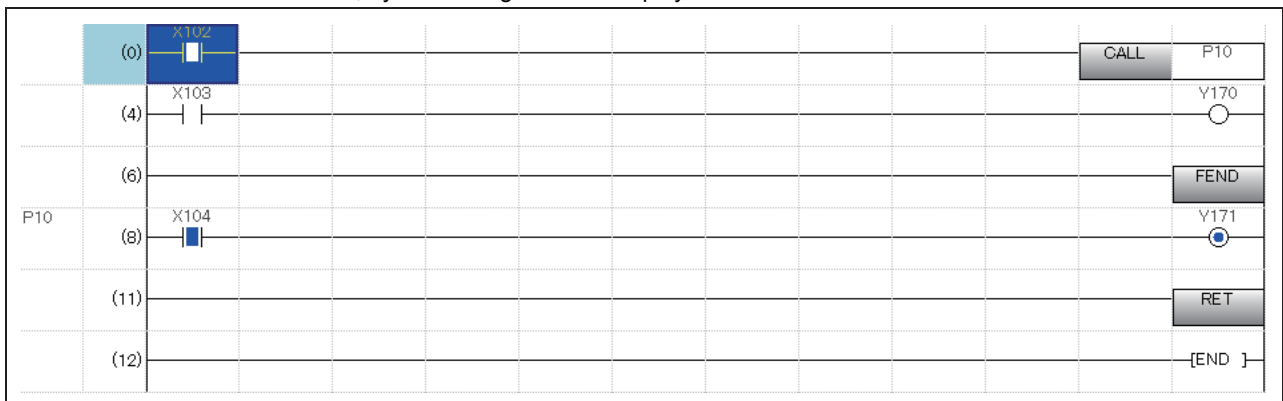


## Operating procedure

1. Creating a new project
2. Creating a program
3. Writing data to the programmable controller
4. Monitoring the ladder

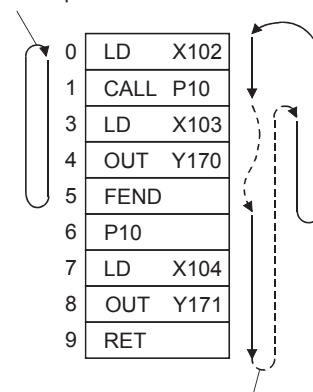
## ■Operation practice

Check the operation of the ladder program, which was created with GX Works3 and written to the programmable controller CPU of the demonstration machine, by monitoring it on the display.



- (1) When X102 is off
  - 1) Operations from 0 to the FEND instruction are processed.
  - 2) Turning on/off X103 turns on/off Y170.
  - 3) Turning on/off X104 does not change the on/off status of Y171.
- (2) When X102 is on
  - 1) After the P10 subroutine program is executed, operations from the step 3 to the FEND instruction are executed.
  - 2) Turning on/off X103 turns on/off Y170.
  - 3) Turning on/off X104 turns on/off Y171.

Operation performed when X102 is off



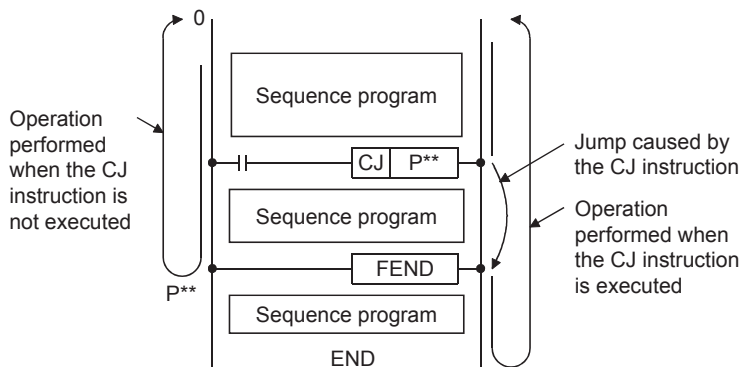
Operation performed when X102 is on

## 1.1.2 [FEND] (End of a main routine program)

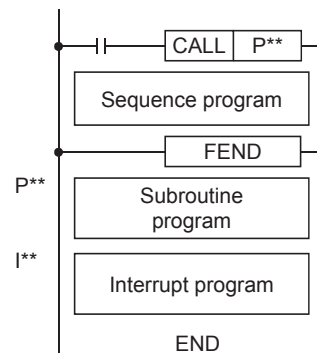


- The FEND instruction is used as an instruction equivalent to the END instruction in the following cases.
  - When sequence programs need to be executed and terminated in each ladder block.  
For example, the FEND instruction is used with the CJ and SCJ instructions.
  - When subroutine programs (the CALL and RET instructions) are used
  - When interrupt programs are used
- After execution of the FEND instruction, the programmable controller executes self-diagnostic check and then processes operations again from the step 0.

■ Operations performed for each ladder block by the CJ instruction



■ When a subroutine program and an interrupt program are included



### Precautions

- Although multiple FEND instructions can be used in sequence programs, the instruction cannot be used in subroutine programs and interrupt programs.
- The FEND instruction cannot be used at the end of a sequence program. Always use the END instruction at the end of a sequence program.

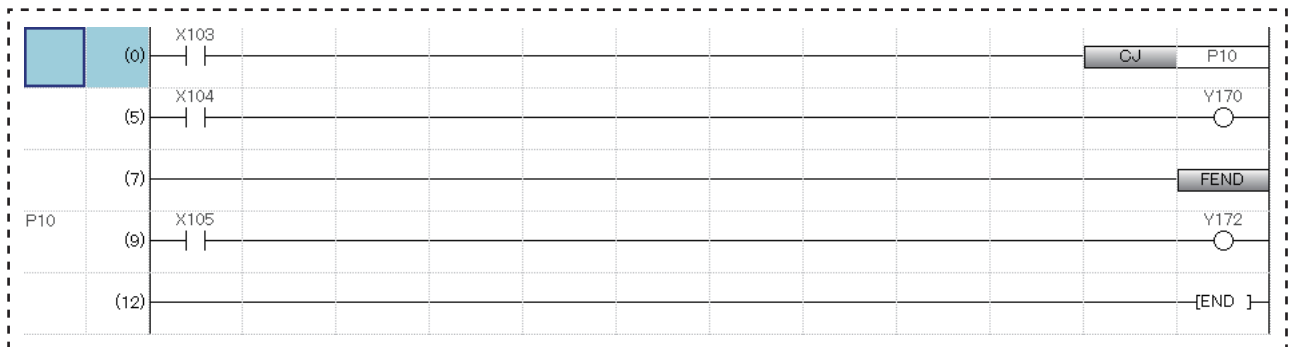
**Point**

An interrupt program interrupts operations in a normal program and is processed when an interrupt factor occurs during execution of the normal program.

## ■Ladder example

Create the following ladder program with GX Works3 and write it to the programmable controller CPU of the demonstration machine. Then, check execution of the FEND instruction.

Project name	EX2
Program name	MAIN

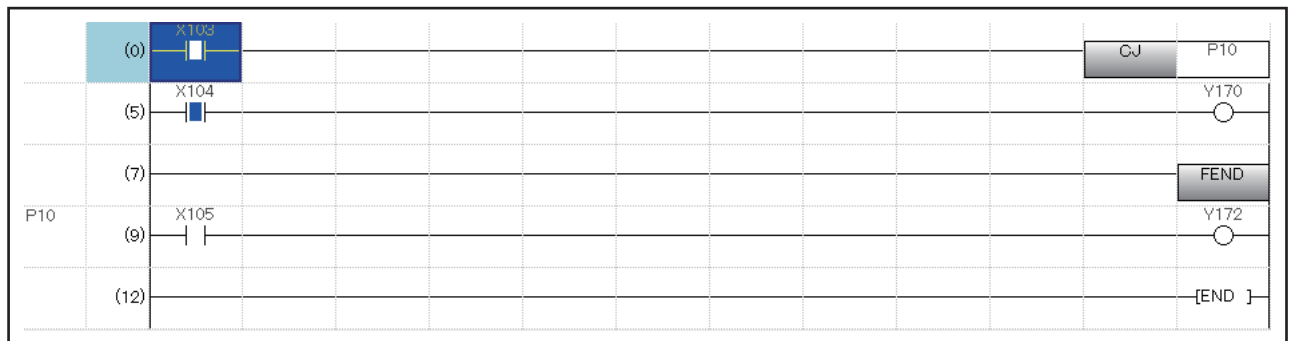


## Operating procedure

1. Creating a new project
2. Creating a program
3. Writing data to the programmable controller
4. Monitoring the ladder

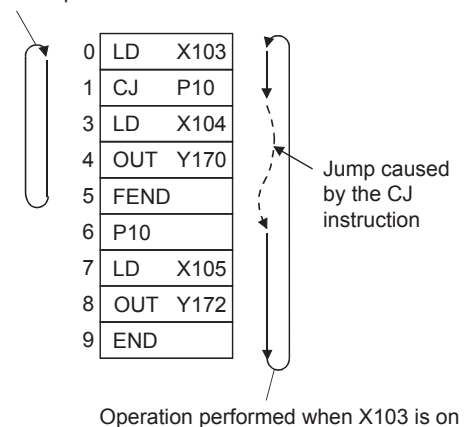
## ■Operation practice

Check the operation of the ladder program, which was created with GX Works3 and written to the programmable controller CPU of the demonstration machine, by monitoring it on the display.



- (1) When X103 is off
  - 1) Operations from 0 to the FEND instruction are processed.
  - 2) Turning on/off X104 turns on/off Y170.
  - 3) Turning on/off X105 does not change the on/off status of Y172.
- (2) When X103 is on
  - 1) Jumping to the pointer P10 is triggered by the CJ instruction.
  - 2) Turning on/off X104 does not change the on/off status of Y170.
  - 3) Turning on/off X105 turns on/off Y172.

Operation performed when X103 is off



## 1.2 Index Register (Z/LZ)

### Point

- This section describes the concept of index modification.
- This section describes how to use the index register with a programmable controller.

This device is used for the index modification of a device.

This device is used to indirectly specify a device number. The device number obtained by the addition of a device number in the index register and a directly-specified device number is specified.

Ex.

**D0Z0** → This means D (0 + Z0).  
Device number

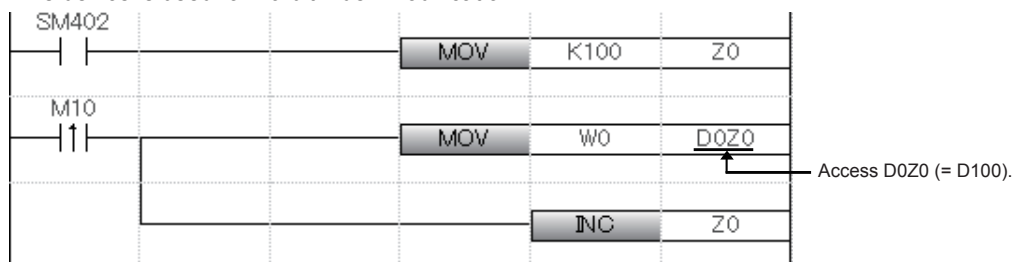
For example, when Z0 is 0, D0Z0 means D0.  
When Z0 is 50, D0Z0 means D50.

### Types of index registers

There are two types of index registers: index register (Z) and long index register (LZ).

#### (1) Index register (Z)

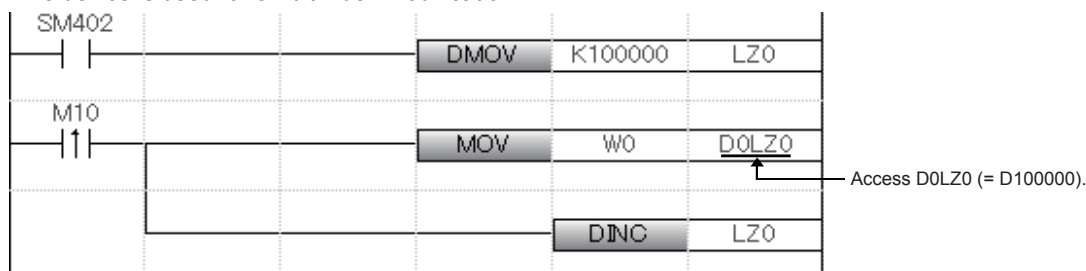
This device is used for 16-bit index modification.



32-bit index modification with ZZ expression using two points of the index register is also available.

#### (2) Long index register (LZ)

This device is used for 32-bit index modification.

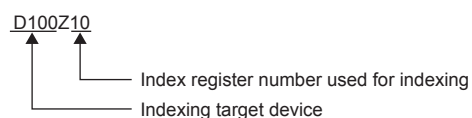


### Index modification

Specify the device number using the index register. The device number to be used is "Device number of device targeted for modification" + "Contents of index register".

#### (1) 16-bit index modification

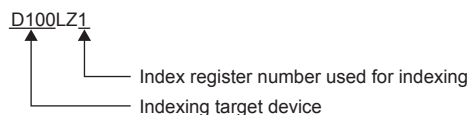
The device number is modified using the index register (Z). The modification range for the device in the case of the 16-bit index modification is -32768 to 32767.





## (2) 32-bit index modification

The device number is modified using the long index register (LZ). The modification range for the device in the case of the 32-bit index modification is -2147483648 to 2147483647.



## Devices for which index modification can be performed

The following table lists the devices that can be targeted for index modification.

Item	Device
16-bit index modification	X, Y, M, L, B, F, SB, V, T <sup>*1</sup> , LT <sup>*1</sup> , ST <sup>*1</sup> , LST <sup>*1</sup> , C <sup>*1</sup> , LC <sup>*1</sup> , D, W, SW, SM, SD, Jn\X, Jn\Y, Jn\B, Jn\SB, Jn\W, Jn\SW, Un\G, U3En\G, U3En\HG, R, ZR, RD, P <sup>*3</sup> , I <sup>*3</sup> , J, U, K, H
32-bit index modification	M, B, SB, T <sup>*1</sup> , LT <sup>*1</sup> , ST <sup>*1</sup> , LST <sup>*1</sup> , C <sup>*1</sup> , LC <sup>*1</sup> , D, W, SW, Jn\B <sup>*2</sup> , Jn\W <sup>*2</sup> , Un\G <sup>*2</sup> , U3En\G <sup>*2</sup> , U3En\HG <sup>*2</sup> , R, ZR, RD, K, H

\*1 Can be used for the contact, coil and current value.

\*2 For network numbers and the specification source of I/O numbers, 32-bit-based index modification cannot be used.

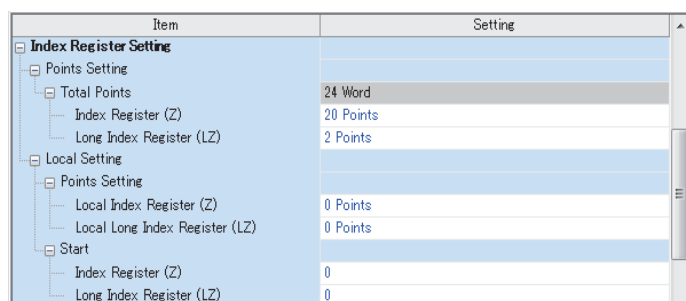
\*3 When it is used as an interrupt pointer, index modification cannot be performed.

## Index register setting

The following window allows to specify the number of points for the index register (Z) and long index register (LZ) and the range where they are used as a local device on GX Works3. Set 24 words as the total number of index register (Z) points and long index register (LZ) points.

[CPU Parameter] → "Memory/Device Setting" → "Index Register Setting"

### Window



### Displayed items

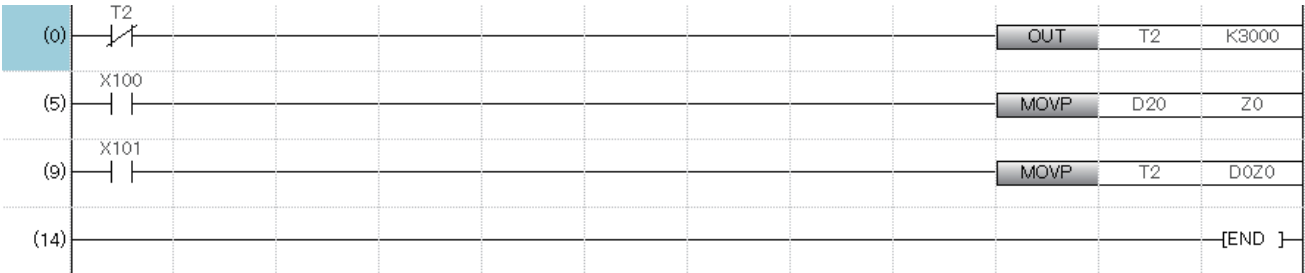
Item	Description	Setting range	Default
Points Setting	Total Points	—	—
	Index Register (Z)	0 to 24 points (in increments of two points)	20 Points
	Long Index Register (LZ)	0 to 12 points (in increments of one point)	2 Points

Item			Description	Setting range	Default
Local Setting	Points Setting	Local Index Register (Z)	Set the number of points of index registers used as local devices. Set it within the range of index registers.	0 to 24 points (in increments of one point)	0 Points
		Local Long Index Register (LZ)	Set the number of points of long index registers used as local devices. Set it within the range of long index registers.	0 to 12 points (in increments of one point)	0 Points
	Start	Index Register (Z)	Set the start number of local index registers. Set it within the range of index registers.	0 to 23	0
		Long Index Register (LZ)	Set the start number of local long index registers. Set it within the range of long index registers.	0 to 11	0

■Example

The following program writes data to the data register having the number specified with a digital switch.

Project name	EX3
Program name	MAIN



Execute the device batch monitor and check the operation of the ladder program.

Set a two-digit number for the initial input device D20 and turn on X100.

For how to input a number for the initial input device D20, refer to GOT screen display on page App. -2.

Initial input device D20



Z0 = 50

D0Z0 = D50

Device Name	+7	+6	+5	+4	+3	+2	+1	+0
D24	0	0	0	0	0	0	0	0
D32	0	0	0	0	0	0	0	0
D40	0	0	0	0	0	0	0	0
D48	0	0	0	0	0	660	0	0
D56	0	0	0	0	0	0	0	0
D64	0	0	0	0	0	0	0	0
D72	0	0	0	0	0	0	0	0

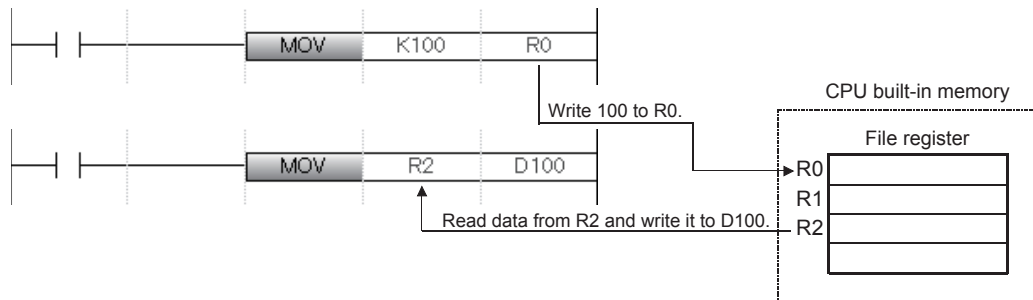
The current value of T2 is transferred to D50.

# 1.3 File Register (R/ZR)

## Point

- This section describes differences between the file register and data register.
- This section describes how to create file registers.
- This section describes storage locations of device data.

The file register is a word device for extension. Data is stored in a file for the file register in the CPU built-in memory. This data is a file register file existing in the file storage area on the device/label memory.



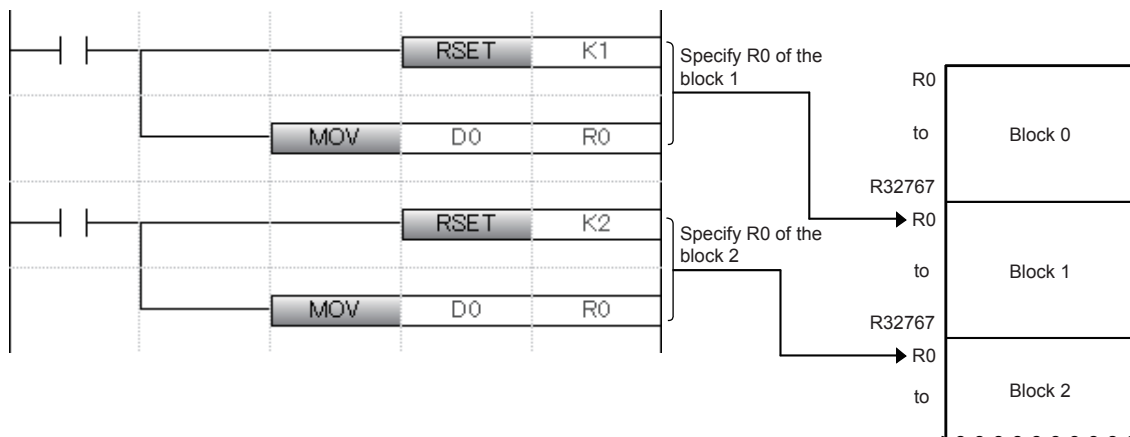
## Specification method

Two specification methods of file registers are provided: block switching method and serial number method.

### (1) Block switching method

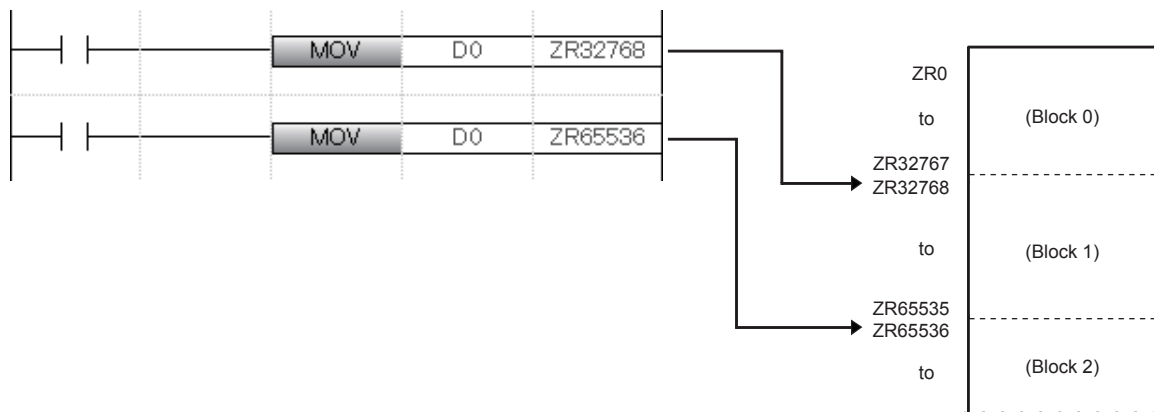
In this method, points of file registers in use are specified by being divided in increments of 32K points (R0 to R32767). When using multiple blocks, the points are specified by being switched to the block numbers used by the RSET instruction. R is used as the device symbol. The R range is R0 to R32767. In the following cases, however, the upper limit of device numbers will be "block size (unit: word) -1".

- The file register file size is smaller than 64K bytes.
- The file register file size is not a multiple of 64K bytes and the end block is specified by the RSET instruction.



(2) Serial number method

In this method, file registers having more than 32K points are specified with serial device numbers. File registers of multiple blocks can be used as consecutive file registers. ZR is used as the device symbol. The ZR range is "ZR0 to (file register file size (unit: word) -1)."

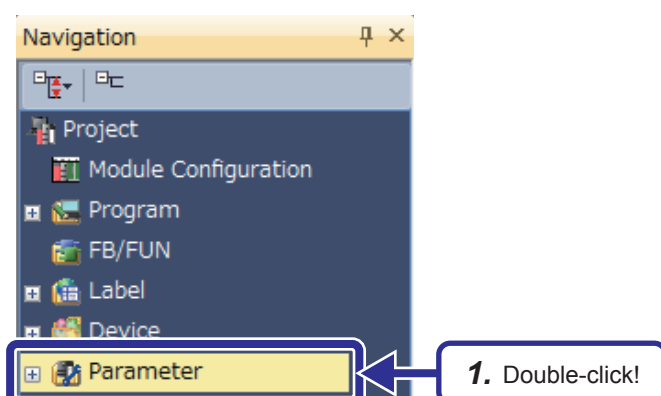


## ■Example

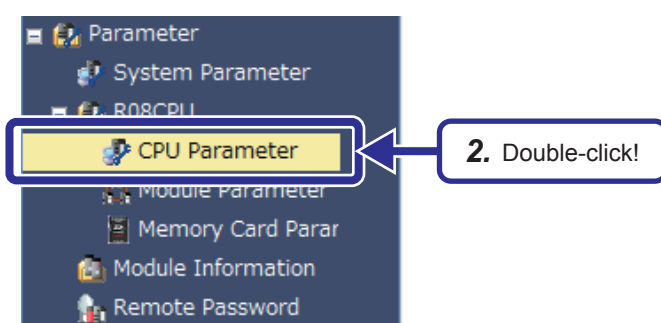
Set file registers of 32K points, R0 to R32767, and use them in a program.

### Operating procedure

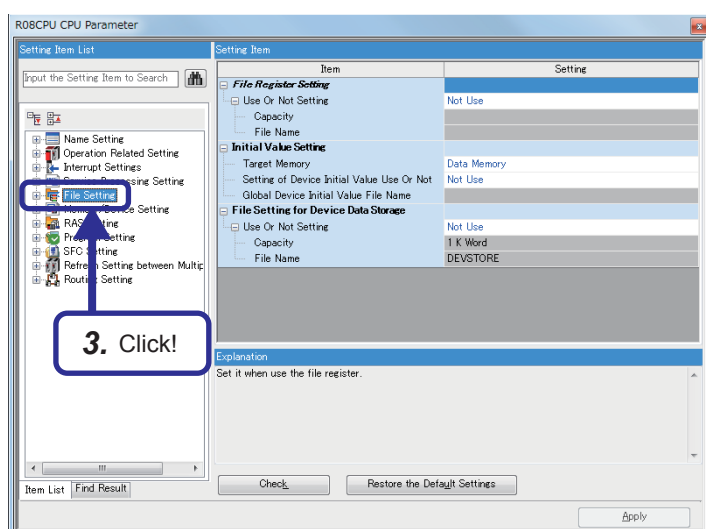
1. Double-click "Parameter" in the "Project" view.



2. Double-click "R08CPU" and "CPU Parameter".

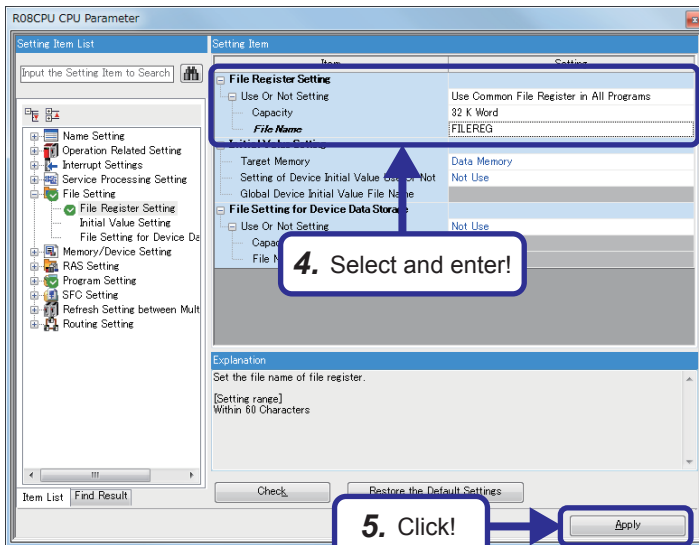


3. The "CPU Parameter" dialog box appears. Click "File Setting".



(To the next page)

(From the previous page)



4. Select "Use Common File Register in All Programs" for "Use or Not Setting" of "File Register Setting".

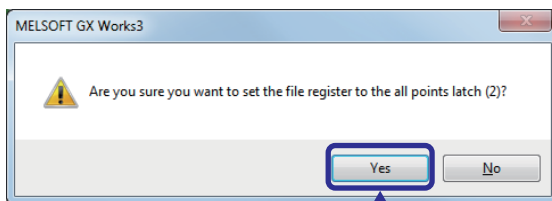
Set the following parameters.

[Parameters to be set]

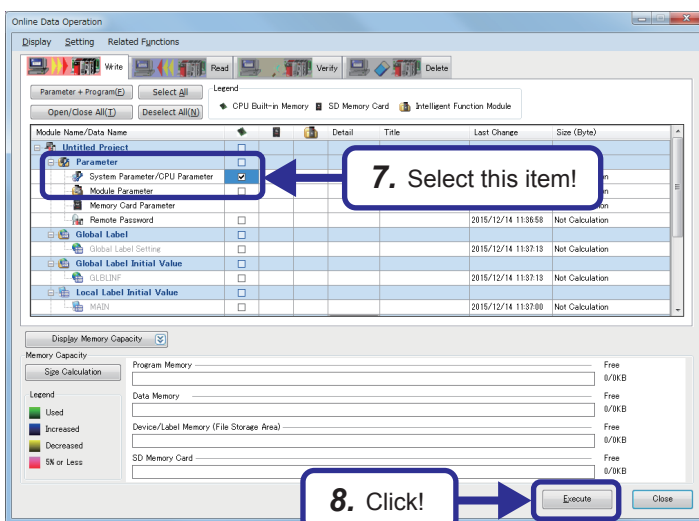
Capacity: 32 K Word

File Name: FILEREG

5. After the setting is completed, click the [Apply] button.



6. The message shown on the left appears. Click the [Yes] button.

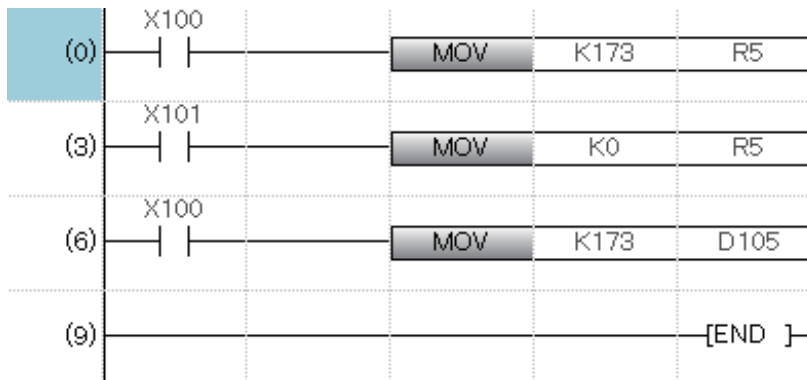


7. Click [Online] → [Write to PLC] to display the "Online Data Operation" dialog box. Select "System Parameter/CPU Parameter" in "Parameter".

8. Click the [Execute] button and write data.

To write or clear file register data with a program, write the following program.  
Turning on X100 writes data, and turning on X101 clears the data.

Project name	EX4
Program name	MAIN



The battery holds data in the file register of the RAM.  
The data cannot be cleared by resetting the CPU module or powering off the system.  
To clear the data, write "0".

{ For comparison



# 1.4 Latch Function

## Point

- This section describes the word "latch".
- This section describes how to create a latch.

Data in each device/label of a CPU module is cleared to each default value in the following cases.

- When the CPU module is powered off and on
- When the CPU module is reset
- When the power failure time has exceeded the allowable momentary power failure time

Data in each device/label set in the latch setting is held even under the above conditions. For this reason, controls can be continuously performed with the held data even after the CPU module is powered off or when the power failure time has exceeded the allowable momentary power failure time while data is managed by sequence controls.

## 1.4.1 Latch types

The following two latch types are provided.

- Latch (1): Data within this range can be cleared by a latch clear operation.
- Latch (2): Data within this range cannot be cleared by a latch clear operation.

## 1.4.2 Devices that can be latched

The following devices can be latched. Local devices cannot be latched.

Device	Specification method	Applicable latch type
Internal relay (M)	Set the latch range.	Latch (1) or Latch (2)
Latch relay (L)	Set the number of points. (Data is latched for the set number of points.)	Latch (2)
Link relay (B)	Set the latch range.	Latch (1) or Latch (2)
Annunciator (F)	Set the latch range.	Latch (1) or Latch (2)
Edge relay (V)	Set the latch range.	Latch (1) or Latch (2)
Timer (T)/Long timer (LT)/Retentive timer (ST)/Long retentive timer (LST)	Set the latch range.	Latch (1) or Latch (2)
Counter (C)/Long counter (LC)	Set the latch range.	Latch (1) or Latch (2)
Data register (D)	Set the latch range.	Latch (1) or Latch (2)
Link register (W)	Set the latch range.	Latch (1) or Latch (2)
File register (R, ZR)	Set the number of points. (Data is latched for the set number of points.)/ Set the latch range.* <sup>1</sup>	Latch (2)

\*1 When "Use Common File Register in All Programs" is enabled, all the points are registered within the range for Latch (2). The range can be specified in "Latch (2)" of "Latch Range Setting". (Values out of the specified range are cleared to 0.)

## Point

- When "Use File Register of Each Program" is enabled, the latch range for file registers cannot be specified. (File registers in the whole range are held.)
- When the file register file in use is switched with another one by the QDRSET instruction, the latch range setting for file registers is disabled. After the change, file registers within the whole range are held regardless of the latch range setting.

## 1.4.3 Optimal assignment of device points

The device points to be used can be optimally assigned according to a system.

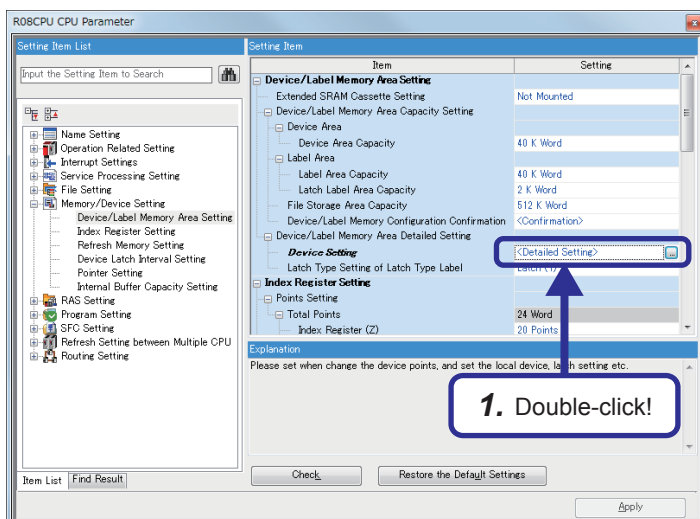
The following table lists setting details.

Item	Description	Setting range	Default
Device points	Set the number of internal device points.	Total number of points of each device ≤ Device area capacity	X: 12K (fixed) Y: 12K (fixed) M: 12K B: 8K SB: 2K F: 2K V: 2K T: 1K LT: 1K ST: 0 LST: 0 C: 512 LC: 512 D: 18K W: 8K SW: 2K L: 8K
Latch range (1) (latch clear operation enabled)	Set a latch range where data can be cleared by a latch clear operation.	Total of Latch (1) + Total of Latch (2) ≤ 32 ranges	No setting
Latch range (2) (latch clear operation disabled)	Set a latch range where data cannot be cleared by a latch clear operation.		No setting

### ■ Example

The following describes how to change the number of internal relay (M) points to 10K and the number of data registers (D) to 1K points (D0 to D500: Latch clear operation enabled, D501 to D1023: Latch clear operation disabled).

### Operating procedure



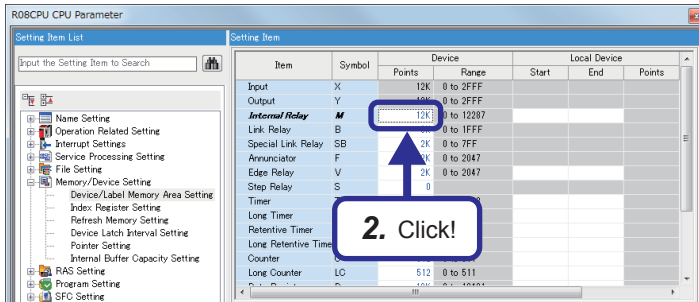
1. Double-click "Detailed Setting" of "Device Setting".

[CPU Parameter] → "Memory/Device Setting" → "Device/Label Memory Area Detailed Setting" → "Device Setting" → "Detailed Setting"

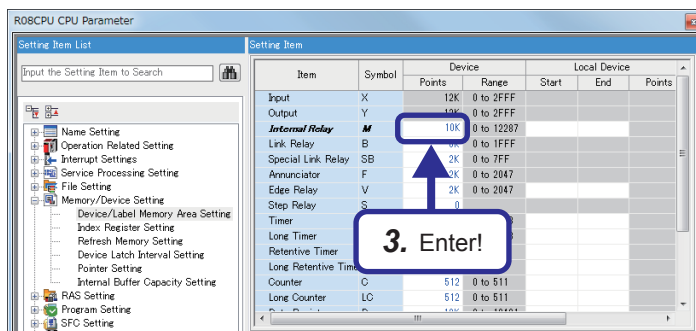


(To the next page)

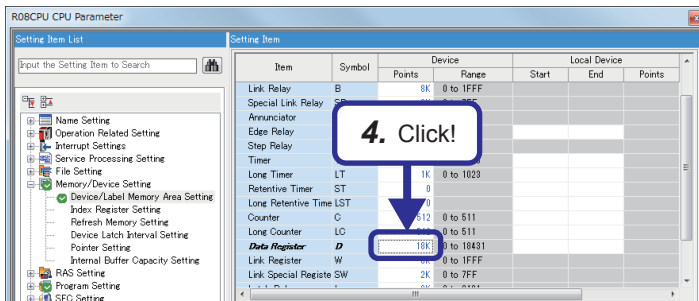
(From the previous page)



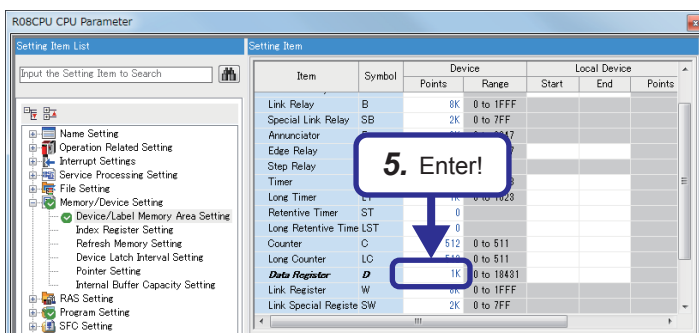
- Click "Points" of "Internal Relay" (M) in the "Device" column to move the cursor.



- Enter "10K" and press the  key.



- Click "Points" of "Data Register" (D) to move the cursor.

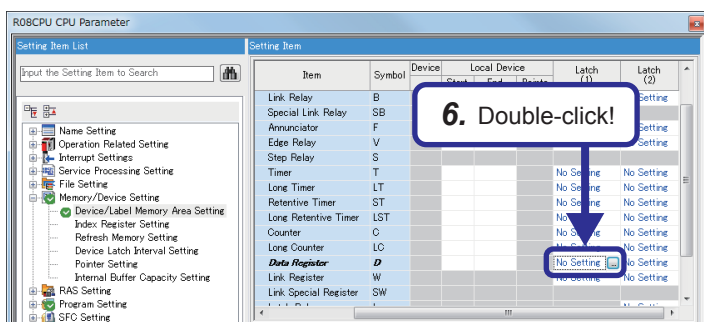


- Enter "1K" and press the  key.

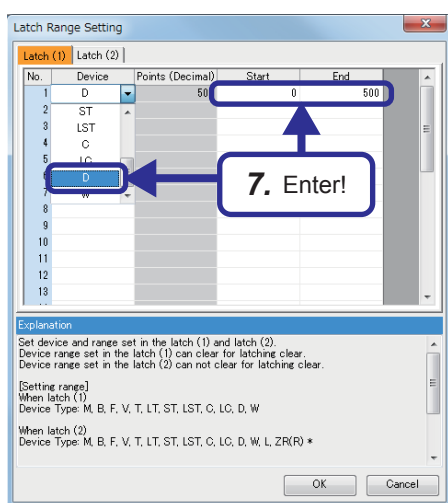


(To the next page)

(From the previous page)



6. Move the cursor to the right and double-click the "Latch (1)" column.



7. The "Latch Range Setting" window appears.

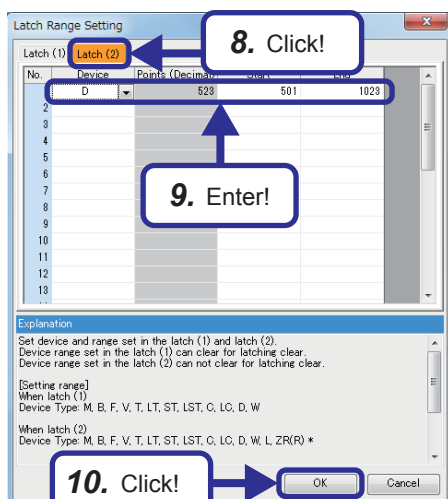
Set the following parameters.

[Parameters to be set]

Device: D

Start: 0

End: 500



8. Click the "Latch (2)" tab.

9. Set the following parameters.

[Parameters to be set]

Device: D

Start: 501

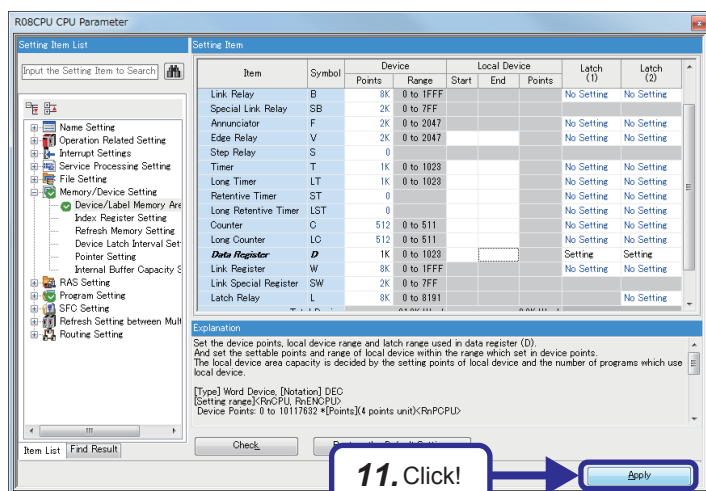
End: 1023

10. Click the [OK] button.



(To the next page)

(From the previous page)



11. Click the [Apply] button to determine the settings.  
Setting of device points is completed.



When the assignment range of (internal user) devices has been changed, the sequence program created with the parameters before the change cannot be used.  
After the change, write the sequence program and the parameters to the CPU module again.

# 2 FILE-BASED PROGRAM MANAGEMENT AND PROGRAM EXECUTION MANAGEMENT

## 2.1 File-based Management

2



This section describes advantages of multiple programs.

With the MELSEC iQ-R CPU module, multiple program designers can create a single program by dividing it into multiple files by function or process.

### Advantages of file-based management

- (1) Programs can be stored into the memory in file units. Thus, addition or modification of a file in response to partial changes in a program does not affect other files.
- (2) Users can manage programs and data with names and time.
- (3) Write protection can be applied to each file.

## File types and storage memory

The following table lists file types and storage destination memory types.

⊙: Can be stored (required for operation), ○: Can be stored, ×: Cannot be stored

File type		CPU built-in memory			SD memory card	File name and extension
		Program memory	Device/label memory	Data memory		
		Drive 0	Drive 3	Drive 4	Drive 2	
Program		⊙ <sup>*4</sup>	×	⊙ <sup>*4</sup>	○	ANY_STRING.PRG
FB program		○ <sup>*4</sup>	×	○ <sup>*4</sup>	○	ANY_STRING.PFB
CPU parameter		×	×	⊙	○	CPU.PRM
System parameter		×	×	⊙	○	SYSTEM.PRM
Module parameter		×	×	○	○	UNIT.PRM
Module extension parameter		×	×	○	○	• UEXmmmnn.PRM <sup>*1</sup> • UEXmmm00.PPR <sup>*5</sup>
Memory card parameter		×	×	×	○	MEMCARD.PRM
Device comment		×	×	○	○	ANY_STRING.DCM
Device initial value		×	×	○	○	ANY_STRING.DID
Global label setting file		×	×	○	○	GLBLINF.IFG
Initial label value file	Initial global label value file	×	×	○	○	GLBLINF.LID
	Initial local label value file	×	×	○	○	PROGRAM_NAME.LID
File register		×	○	×	○ <sup>*3</sup>	ANY_STRING.QDR
Event history		×	×	○	○	EVENT.LOG
Device data storage file		×	×	○	○ <sup>*3</sup>	DEVSTORE.QST
General-purpose data		×	×	○	○	ANY_STRING.CSV/BIN
Data logging setting file	Common setting file	×	×	×	○	LOGCOM.LCS
	Individual setting file	×	×	○	○	LOGnn.LIS <sup>*2</sup>
Remote password		×	×	○	○	00000001.SYP

\*1 mmm indicates a value calculated by dividing the module I/O No. by 10H (3 digits in hexadecimal). For the CPU module, it will be 3FFH. Also, nn is the serial number (2-digit hexadecimal number) of a module extension parameter of each module.

\*2 nn corresponds to the setting number and is 01 through 10.

\*3 Can be stored but cannot operate as a function.

\*4 When a program or a FB (function) program is stored in the built-in memory of the CPU module, it is divided for the program memory and the data memory.

\*5 Module extension parameter for the protocol setting, storing protocol setting information in the predefined protocol support function

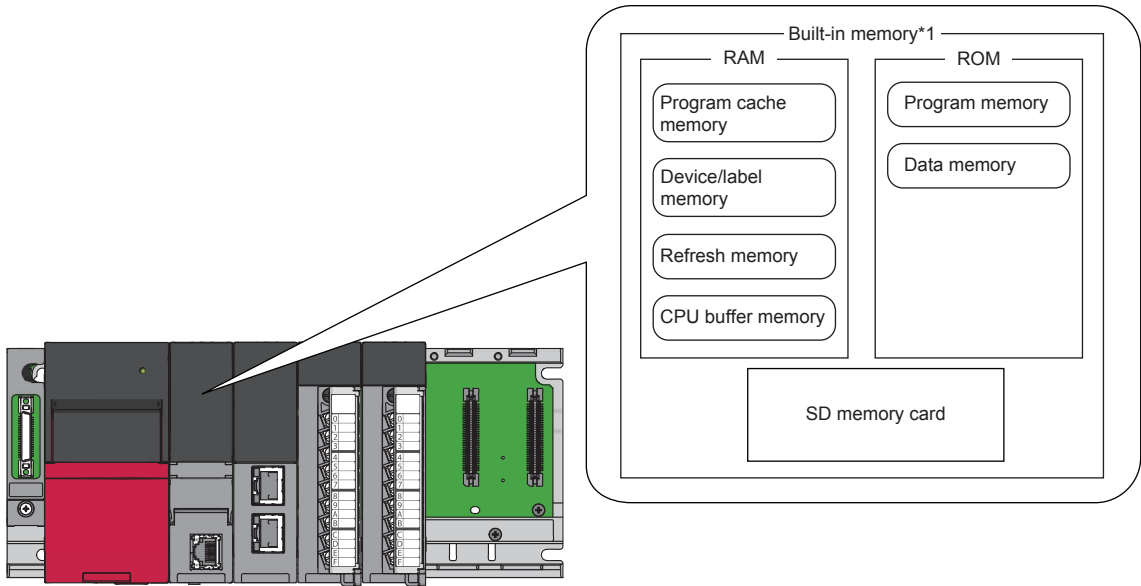
# 2.1.1 CPU built-in memory and SD memory card

The RCPU built-in memory and an SD memory card are used to store files.  
The following table lists the memory capacity of each CPU module.

Item	R04CPU	R08CPU	R16CPU	R32CPU	R120CPU
Program memory	160K bytes	320K bytes	640K bytes	1280K bytes	4800K bytes
SD memory card	Differs depending on the SD memory card used. (SD/SDHC memory card: 32G bytes maximum)				

## CPU built-in memory

The following figure shows the memory configuration of the CPU module.



\*1 The built-in memory is a generic term for internal memory areas of the CPU module.



The usage of the memory can be checked from the engineering tool. For details, refer to the following.  
 GX Works3 Operating Manual



## ■Program memory/program cache memory

The program memory and program cache memory store necessary programs for the CPU module to perform operations. At the following timing, data in the program memory is transferred to the program cache memory<sup>\*1</sup> and an operation is performed.

<sup>\*1</sup> This memory is used for program operations.

- When the CPU module is powered on
- When the CPU module is reset

### (1) Application

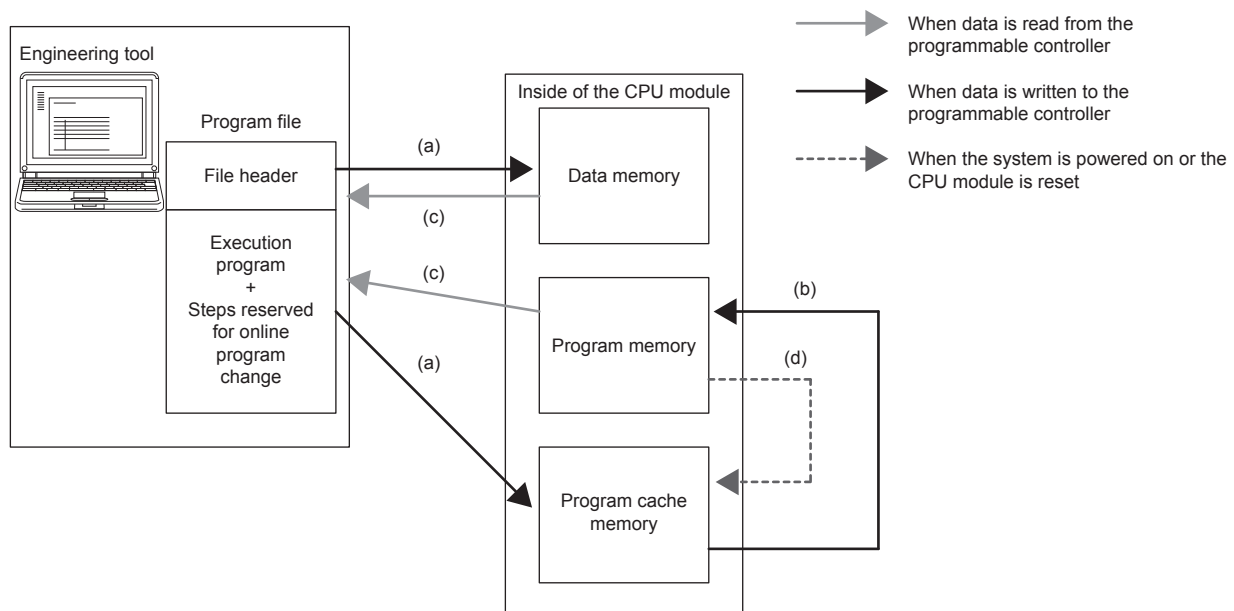
The program memory and program cache memory store execution programs of program files and FB (function block) files. Execution programs include instruction codes, statements, and notes.

### (2) Data to be allocated

Only the execution programs that exist in program files and FB files are allocated to the program memory.

### (3) Data allocation and procedure of read/write operations

The following figure shows the data allocation of the program memory and program cache memory and the procedure of read/write operations from/to the programmable controller.



- (a) When data is written to the programmable controller, execution programs and steps reserved for online program change are written to the program cache memory, and the file header is written to the data memory.
- (b) After being written to the program cache memory, the data is automatically transferred to the program memory.
- (c) When the data is read from the programmable controller, the execution programs and steps reserved for online program change are read from the program memory, and the file header is read from the data memory.
- (d) When the CPU module is powered off and on or reset, the data in the program memory is transferred to the program cache memory and operations are executed.

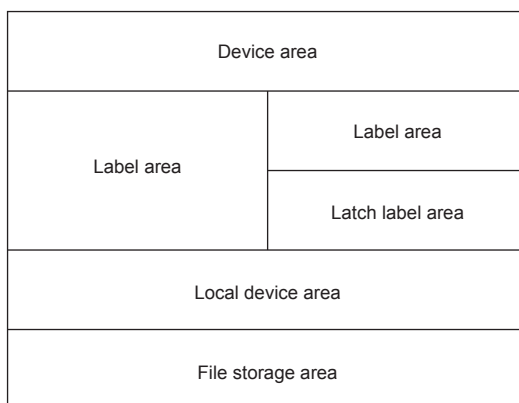
### Point

For read/write operations from/to the programmable controller, refer to the following.

GX Works3 Operating Manual

## ■Device/label memory

The device/label memory has the following areas.




### Data to be allocated

The following table lists the data allocated to each area.

Area		Application
Device area		User device
Label area	Label area	Global label and local label
	Latch label area	Global label and local label with latch specified
Local device area		Local device (excluding index register)
File storage area		File register file and other data <sup>*1</sup>

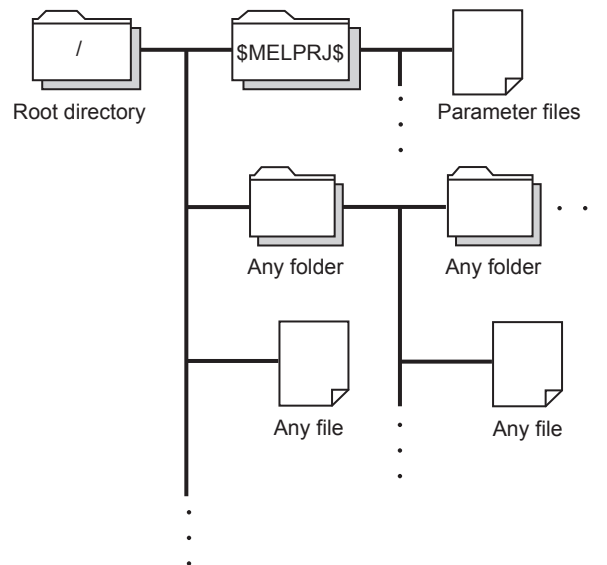
\*1 File register files which are stored in the area for storing file register files can be written or read in file unit.

#### Point

Free space of areas can be checked by clicking [CPU Parameter] → "Memory/Device Setting" → "Device/Label Memory Area Capacity Setting" →  button.

## ■Data memory

This memory is used to store parameter files, device comment files, and/or user's folders/files. Data written by the engineering tool (e.g. parameter files and device comment files) is stored in the "\$MELPRJ\$" folder. The "\$MELPRJ\$" folder is created after memory initialization. Note that the "\$MELPRJ\$" folder cannot be deleted. (Folders under the "\$MELPRJ\$" folder can be deleted.)



## Point

For details on how to create and delete user's folders, refer to the following.

GX Works3 Operating Manual

## ■ Refresh memory

This memory is used to store data used in refresh communication with intelligent function modules.

■CPU buffer memory

This memory is used by the Ethernet function or in data communication between multiple CPU modules. (📖 MELSEC iQ-R Ethernet User's Manual (Application))

■SD memory card

This memory is used to store user's folders/files and the folders/files created by a function using the SD memory card. The folder configuration is the same as the one of the data memory. However, in the case of the SD memory card, the "\$MELPRJ\$" folder will be created when the SD memory card becomes available (when the SD memory card is mounted).

## File size unit in memory

The minimum unit of capacity for storing a file in the memory is referred to as file size unit (cluster size).


CPU module	CPU built-in memory			SD memory card
	Program memory	Device/label memory	Data memory	
R04CPU	4 bytes	512 bytes	1024 bytes	32K bytes
R08CPU			2048 bytes	
R16CPU			4096 bytes	
R32CPU			8192 bytes	
R120CPU			16384 bytes	

### Point

Data is written in file size (cluster size) units. For example, when 464 bytes of CPU parameter is written to the data memory of the R04CPU, it is written as 1024 bytes of data because the file size unit of the data memory is 1024 bytes.

## Memory operation

Using an engineering tool, each memory can be initialized and cleared to zero. For details on the operation method, refer to the following.

 GX Works3 Operating Manual

Items to be specified in the engineering tool				Target
Initialization	Data memory			Deletes all the folders and files in the program memory and data memory.
	Device/label memory			Deletes all the files in the file storage areas in the device/label memory.
	SD memory card			Deletes all the folders and files in the SD memory card.
Value clear	Device, label	Zero clear		Excluding the devices and labels specified in the latch range, this operation clears the following devices to zero: X, Y, M, B, F, SB, V, T, ST, LT, LST, C, LC, D, W, SW, FX, FY, FD, Z, LZ, RD, and all labels (including module labels).
		Zero clear (including Latches (1) and (2))		Including the devices and labels specified in the latch range, this operation clears the following devices to zero: X, Y, M, B, F, SB, V, T, ST, LT, LST, C, LC, D, W, SW, FX, FY, FD, Z, LZ, RD, and all labels (including module labels).
	File register	Zero clear	All files	Clears the contents of all the file registers to zero.
			File specification	Clears only the contents of the specified file register(s) to zero.
		Zero clear excluding Latch (2)		Clears the file registers other than Latch (2) to zero.
	Device / label / file register latch clear			Clears devices, labels, and file registers other than Latch (2) to zero.

### Point

If the power goes off during initialization or zero clear, the memory is left in the state at that point. Thus, execute the same memory operation again.

## SD memory card

This section describes the performance specifications of SD memory cards.

Item		NZ1MEM-2GBSD	NZ1MEM-4GBSD	NZ1MEM-8GBSD	NZ1MEM-16GBSD
Type		SD	SDHC		
Capacity		2G bytes	4G bytes	8G bytes	16G bytes
Number of writes		60000 times	100000 times		
External dimensions	Height	32mm			
	Width	24mm			
	Depth	2.1mm			
Weight		2g			

### ■ Handling precautions

- The operation of the SD memory cards manufactured by Mitsubishi (NZ1MEM-□GBSD) has been tested on the MELSEC iQ-R series modules. If an SD memory card\*<sup>1</sup> manufactured by other companies is used, a problem, such as corruption of data in the SD memory card and system operation stop, may occur.
- SD memory cards manufactured by Mitsubishi conform to IEC 61131-2 when being used in a module.
- SDHC-type SD memory cards can be used only with CPU modules that support the use of those cards. (An SDHC logo is shown on the module or described in the manuals of the module.)
- All SD memory cards used by the CPU modules need to be formatted. An SD memory card is unformatted when purchased. Before using the card, insert it into the CPU module and format it using the engineering tool. Do not format an SD memory card using a personal computer. (GX Works3 Operating Manual)
- Data in the SD memory card may corrupt if any of the following operations is performed while the SD memory card is being accessed: powering off the system, resetting the CPU module, or removing the SD memory card. If the CARD ACCESS LED is on, terminate the access to the SD memory card with the SD memory card access control switch, and then power off the system, reset the CPU module, or remove the SD memory card. The use of the SD memory card can be disabled by SM606 (SD memory card forced disable instruction) and the disabled status can be checked by SM607 (SD memory card forced disable status flag).
- Regularly backup important data on other media, such as CDs and DVDs.

\*<sup>1</sup> For the SD memory cards that is commercially available and can be used with the CPU module, refer to the following. Before using a commercially available SD memory card, check that the card does not affect the control of the target system.  
[www.MitsubishiElectric.co.jp/fa](http://www.MitsubishiElectric.co.jp/fa)  
 TECHNICAL BULLETIN No.FA-A-0078

### ■ Precautions

If the system is powered off, the CPU module is reset, or the SD memory card is removed while the SD memory card is being accessed as below, data in the SD memory card may corrupt.

- Data in the buffer memory is being saved to the SD memory card. (Data logging function is being executed.)
- During deletion of a folder by "User Data Operation" in the engineering tool

In the above cases, SD memory card diagnostics such as file system check and restoration is performed at the timing to power on the system or to cancel the reset operation.

Note that data may not be completely restored depending on the file system status.

The following shows the time required to diagnose an SD memory card (when 10000 files (100K bytes of data) are stored in the SD memory card).<sup>\*1\*2</sup>

- NZ1MEM-2GBSD: approximately 7 seconds, NZ1MEM-4GBSD: approximately 8 seconds, NZ1MEM-8GBSD: approximately 9 seconds, NZ1MEM-16GBSD: approximately 10 seconds

\*<sup>1</sup> The more the number of files in the SD memory card, the longer the diagnosis time.

\*<sup>2</sup> When the CPU module is in connection with an external device, simultaneously running the CPU module and the external device can cause communication timeout on the external device side.

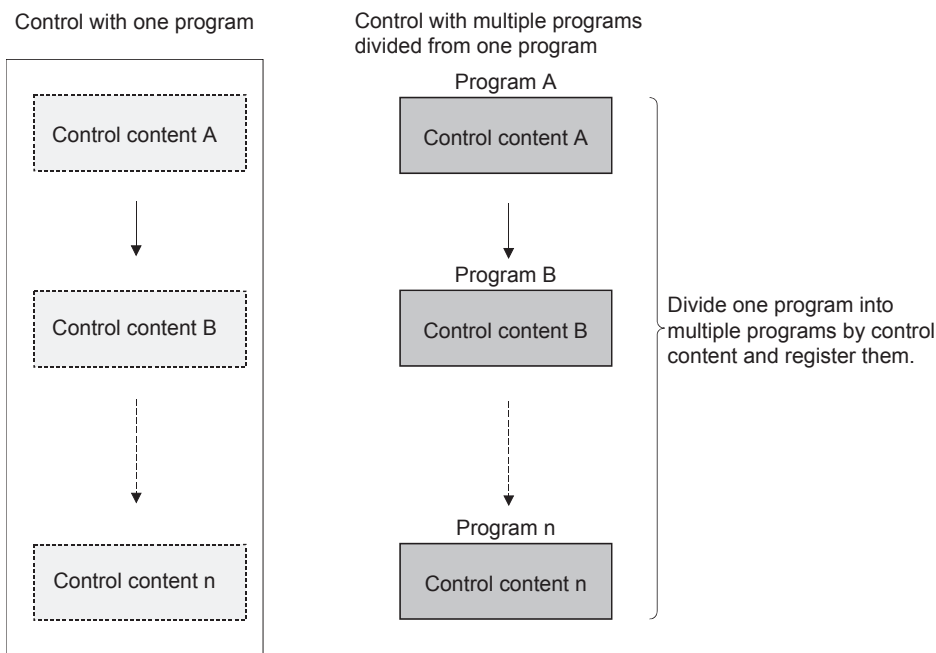
# 2.2 Program Execution Management



This section describes program execution types.

## 2.2.1 Program execution types

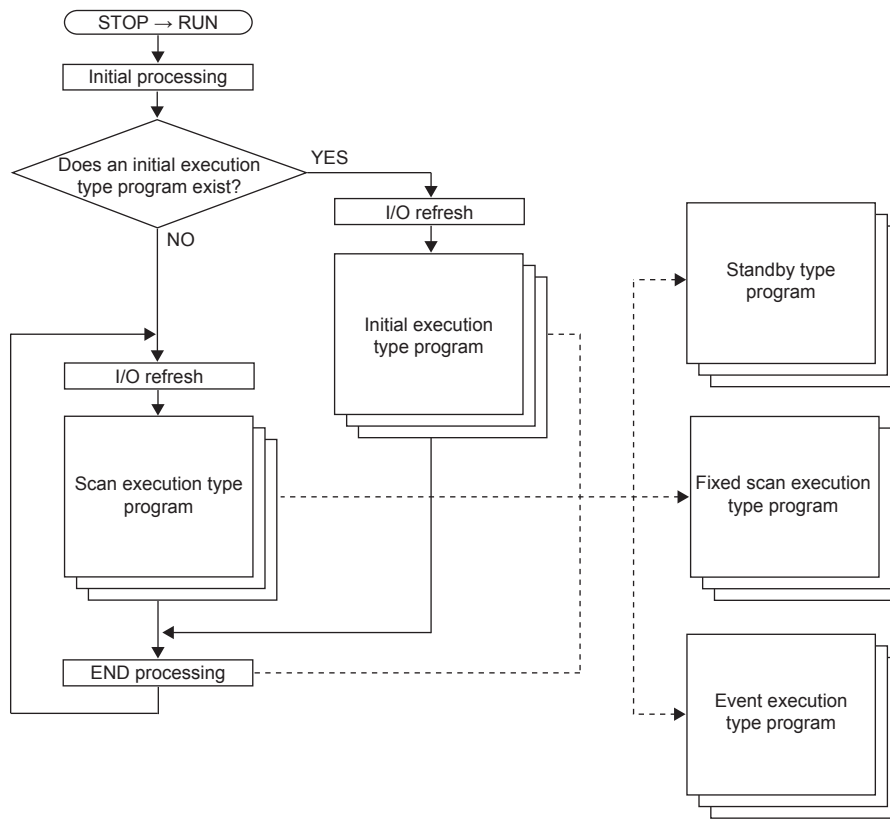
Execution of programs can be managed in the conventional method, in which control contents are integrated into one program, or in the method in which a program is divided into multiple programs by control unit. When dividing a program into multiple programs, set execution types in the program setting of the CPU parameter with GX Works3 to specify the execution type of each divided program.



The RCPU executes each divided program according to the execution types set in the CPU parameter program setting. The following five execution types are provided: initial execution type, scan execution type, standby type, fixed scan execution type, and event execution type.

## Flow of program execution

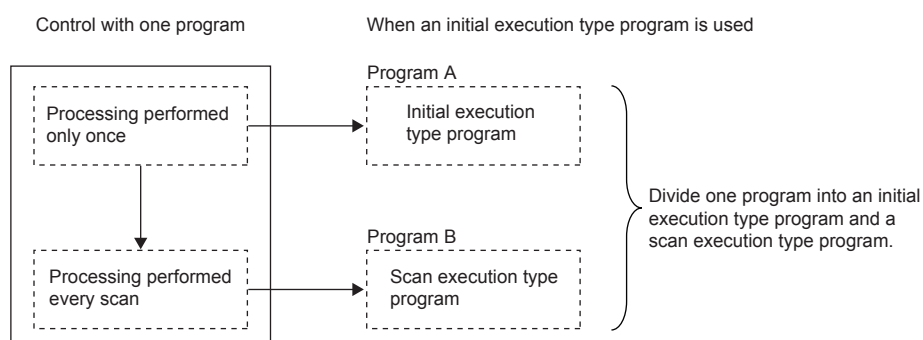
Programs are executed in order according to the set execution types of the programs and execution order when the CPU module is switched to the RUN state.



Programs with the same execution type are executed according to the set execution order.

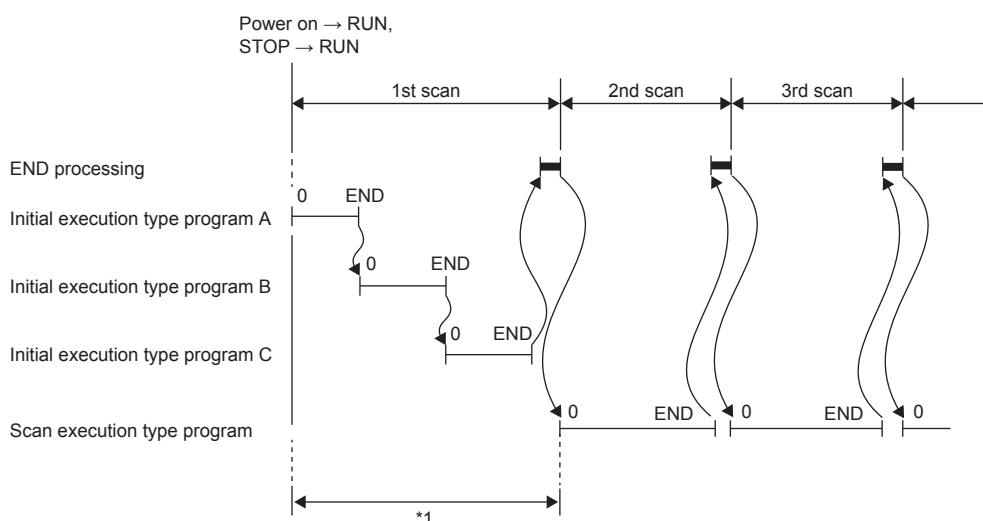
## 2.2.2 Initial execution type program

This type of program is executed only once when the CPU module is powered off and on, or switched from the STOP state to the RUN state. Set this type for a program that does not need to be executed after executed once in the initial scan, such as the initial processing for an intelligent function module.



Note that the execution time of an initial execution type program is the initial scan time.

When multiple initial execution type programs are executed, the execution time of the initial execution type programs will be the time to be taken until execution of all the initial execution type programs is completed.



\*1 The sum of the initial execution type program execution time and END processing time is the initial scan time.

### Precautions

The following lists the precautions for using initial execution type programs.

#### ■ Restrictions on program creation

In initial execution type programs, do not use an instruction which requires several scans to complete processing, such as an instruction with a completion device.

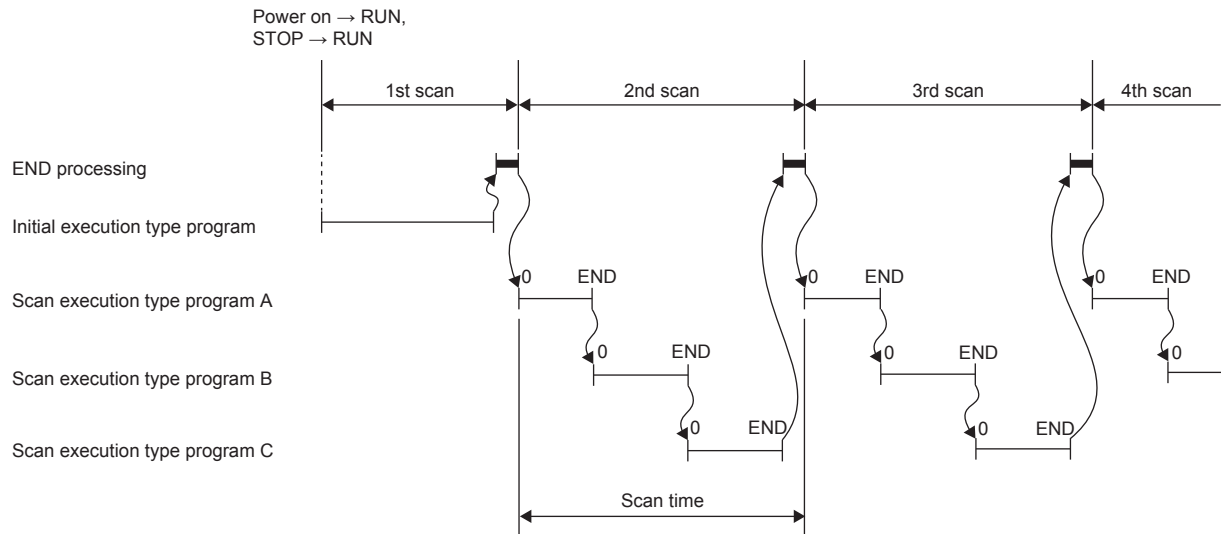
**Ex.**

Example) SEND and RECV instructions



## 2.2.3 Scan execution type program

This type of program is executed only once per every scan starting from the scan after the scan in which an initial execution type program was executed.



When multiple scan execution type programs are executed, the execution time of the scan execution type programs will be the time to be taken until execution of all the scan execution type programs is completed.

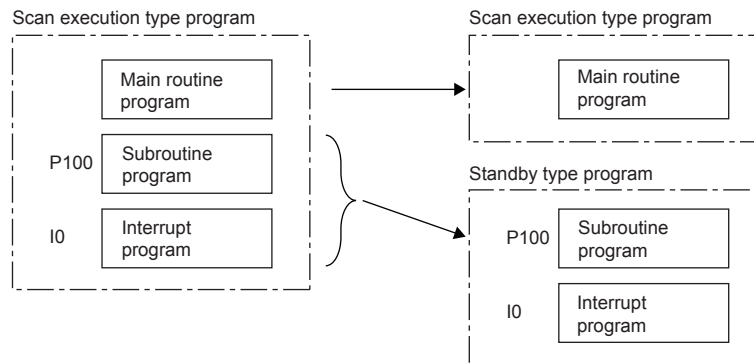
Note that if an interrupt program, fixed execution type program, and/or event execution type program is/are executed before execution of the scan execution type program is completed, the execution time of these programs is also included.

## 2.2.4 Standby type program

This type of program is executed only when its execution is requested.

### Librarization of programs

Use this type of programs when making a subroutine program or an interrupt program as a standby type program and managing it separately from a main routine program. In a single standby type program, multiple subroutine programs and interrupt programs can be created.

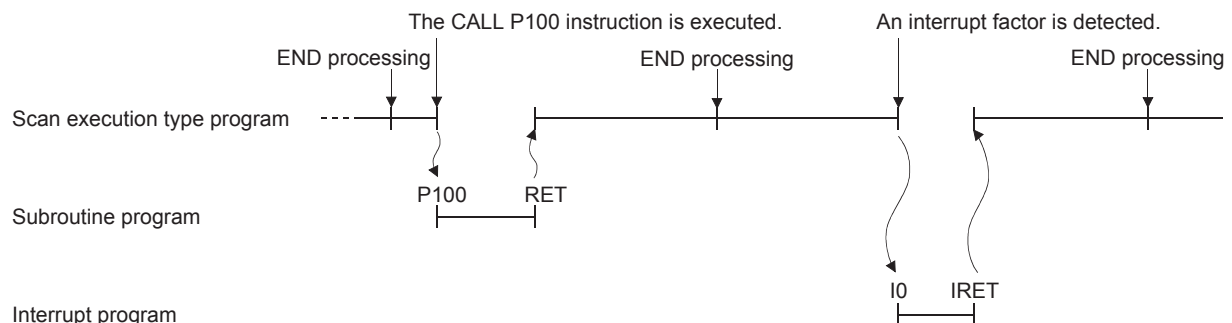


### Execution method

Execute a standby type program as follows.

- Create a subroutine program and an interrupt program in the standby type program, and then call them with occurrence of an interrupt or by specifying them with pointers.
- Switch to another execution type program.

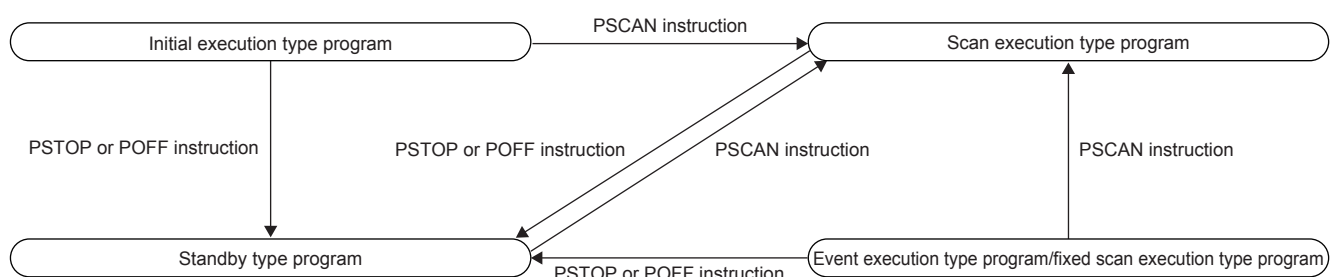
The following figure shows the operation when the subroutine program and interrupt program in the standby type program are executed.



### Program arrangement change

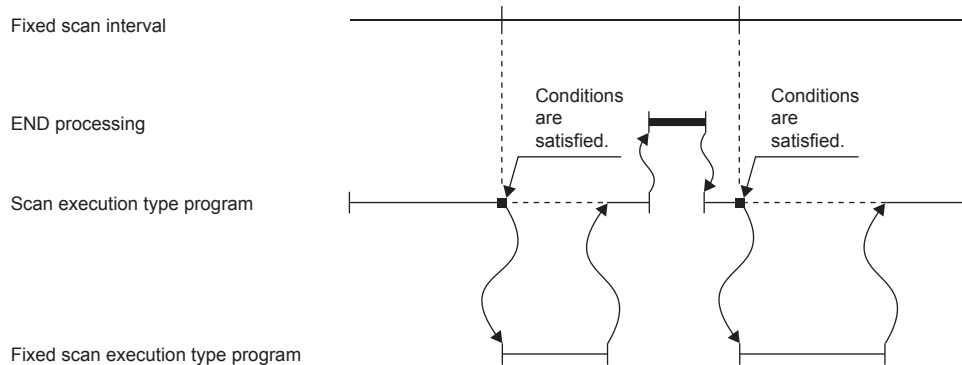
Use this type of programs when programs available for all systems have been prepared and only required ones are executed. For example, a program set as a standby type program in parameters can be changed to a scan execution type program with a program and executed.

Use the PSCAN instruction, PSTOP instruction, and POFF instruction for switching the execution type.



## 2.2.5 Fixed scan execution type program

This type of program is an interrupt program to be executed at a specified time interval. Unlike the normal interrupt programs, this type of program does not require interrupt pointers (I) and the IRET instruction, and is executed in program file units.

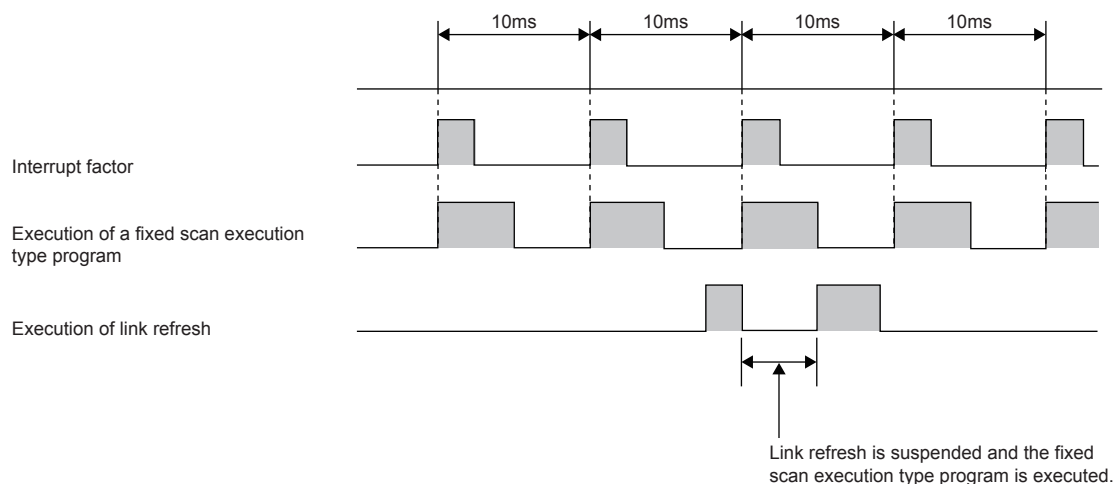


To execute a fixed scan execution type program, use the EI instruction to enable interrupts.

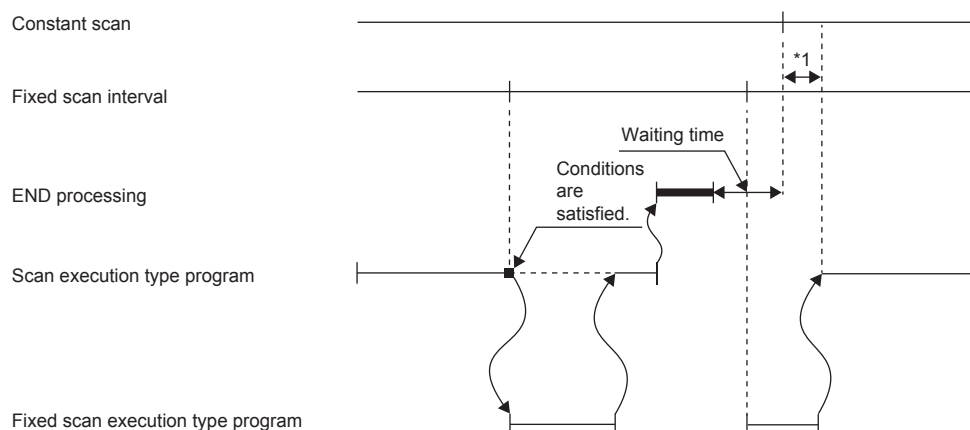
### Operation when execution conditions are satisfied

A fixed scan execution type program performs the following operations.

- (1) When execution conditions are satisfied before interrupts are enabled by the EI instruction  
The program enters the waiting status and will be executed when an interrupt is enabled. Note that if execution conditions for this fixed scan execution type program are satisfied more than once while the program is in the waiting status, the program is executed only once when an interrupt is enabled.
- (2) When there are multiple fixed scan execution type programs  
When the specified time for these programs has passed simultaneously, they are executed in order according to the specified execution order.
- (3) When other or the same execution conditions are satisfied while a fixed scan execution type program is being executed  
The fixed scan execution type program is executed according to the fixed scan execution mode setting.
- (4) When execution conditions are satisfied while interrupts are disabled by the system  
The fixed scan execution type program is executed according to the fixed scan execution mode setting.
- (5) When an interrupt factor occurs during link refresh  
Link refresh is suspended and the fixed scan execution type program is executed. Even while station-based block data assurance is enabled for cyclic data during refresh of such links as CC-Link IE, if the fixed scan execution type program uses a device specified as the refresh target, station-based block data assurance for cyclic data is not available.



- (6) If an interrupt factor occurs during the waiting time after END processing in constant scan execution  
The fixed scan execution type program is executed.



\*1 If processing is not completed during the waiting time, the scan time is extended.

- (7) If another interrupt occurs while a fixed scan execution type program is being executed  
If an interrupt program (including an interrupt which triggers an event execution type program) is triggered while a fixed scan execution type program is being executed, programs are executed in accordance with the interrupt priority.

## Fixed scan execution mode

For fixed scan interrupts (I28 to I31, I48, I49) triggered by a fixed scan execution type program or the internal timer of the CPU module, users can specify the program execution operation to be performed when multiple interrupt occur.

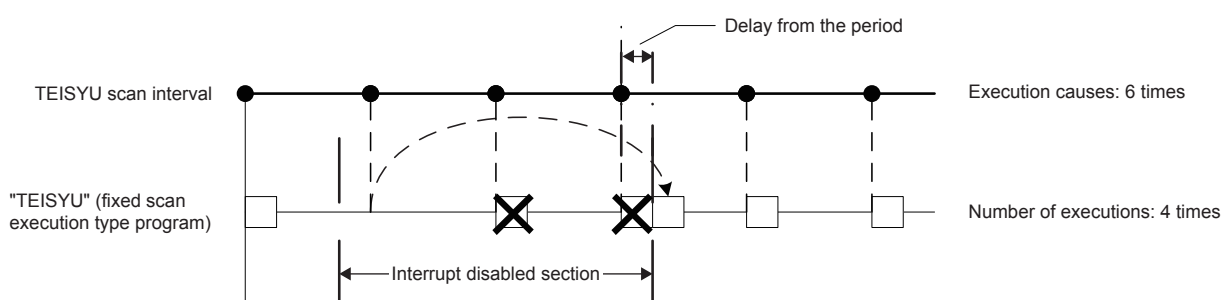
However, an execution cause which occurs while interrupts are disabled by the DI instruction is excluded from the fixed scan execution mode target.

- (1) Operation in the fixed scan execution mode

The following describes the operations that can be performed in the fixed scan execution mode.

- Execution Count Takes Priority

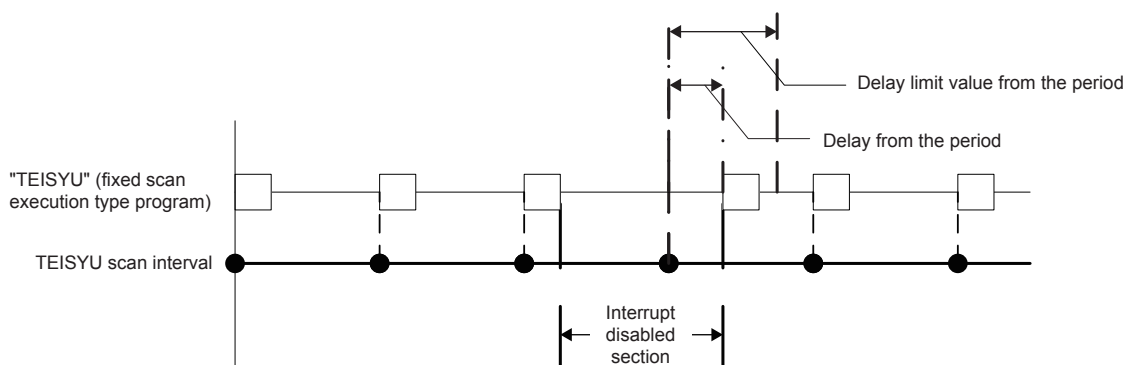
A program is executed for the total number of pending executions and execution causes that has been generated after interrupts are enabled.



- Precede Fixed Scan

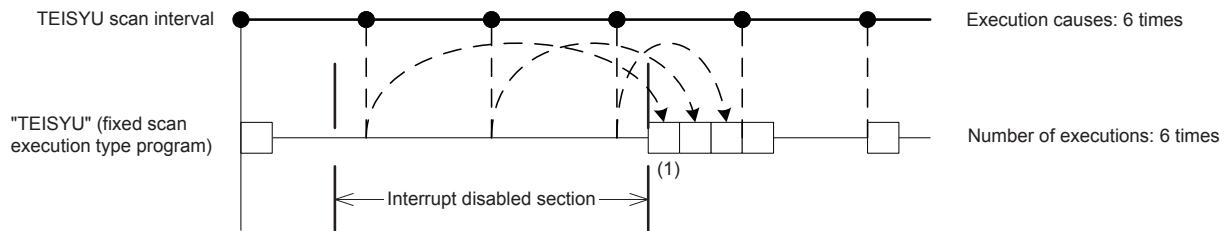
When a pending execution exists, a program is performed within the delay limit value range.

However, even if multiple pending executions exist, the program is performed only once.



(2) Delay limit value from the period

Set the allowable range of delay from the period. A pending program is executed when interrupts are enabled within this range. If interrupts are enabled outside the range, the program is not executed.

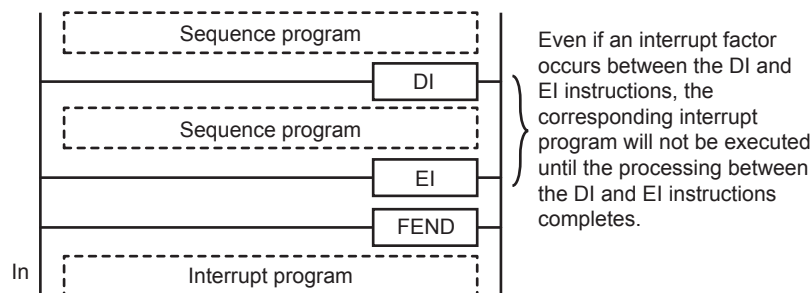


The delay limit value from the period differ as follows.

- For fixed scan execution type programs  
The delay limit values from the period will be the greatest common divisor of the fixed scan intervals of all the fixed scan execution type programs. For example, if there are fixed scan execution type programs with 2ms, 4ms, and 12ms scan periods, the delay limit value from the period is 2ms.
- For fixed scan interrupts using the internal timer of the CPU module (I28 to I31, I48, and I49)  
The delay limit value from the period is equal to the fixed scan interval and fixed. For example, if the periods are I28: 100ms and I29: 20ms, the delay limit values from the period are I28: 100ms and I29: 20ms.

## Executing the EI instruction

The EI instruction clears the interrupt disabled state that has been set by the DI instruction, and enables execution of fixed scan execution type programs and interrupt programs with the interrupt pointer numbers permitted by the IMASK instruction. When the IMASK instruction is not executed, I32 to I47 are disabled.



### Point

#### Terms

(1) DI

- This instruction disables execution of interrupt programs until the EI instruction is executed even though they are requested.
- When the system is powered on or the CPU module is reset, the system is in the state where the DI instruction has been executed.

(2) IMASK

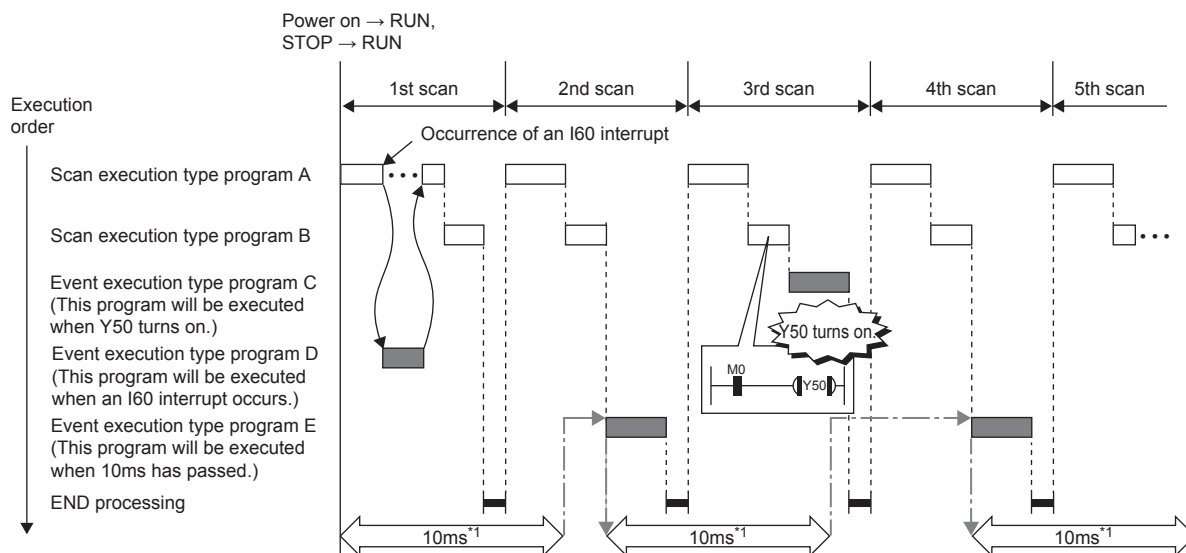
This instruction enables or disables execution of the interrupt program with the interrupt pointer number specified by the 16-point bit pattern starting from the device specified by the start number of device areas (16-bit binary data) where interrupt factor mask data has been stored.

- 1 (on): Execution of interrupt programs is enabled.
- 0 (off): Execution of interrupt programs is disabled.

## 2.2.6 Event execution type program

Execution of this type of program is triggered by a specified event.

Programs are executed in the execution order specified in the program setting of the CPU parameters, and when execution conditions of a specified trigger are satisfied when the execution turn of the corresponding event execution type program comes, the program is executed.



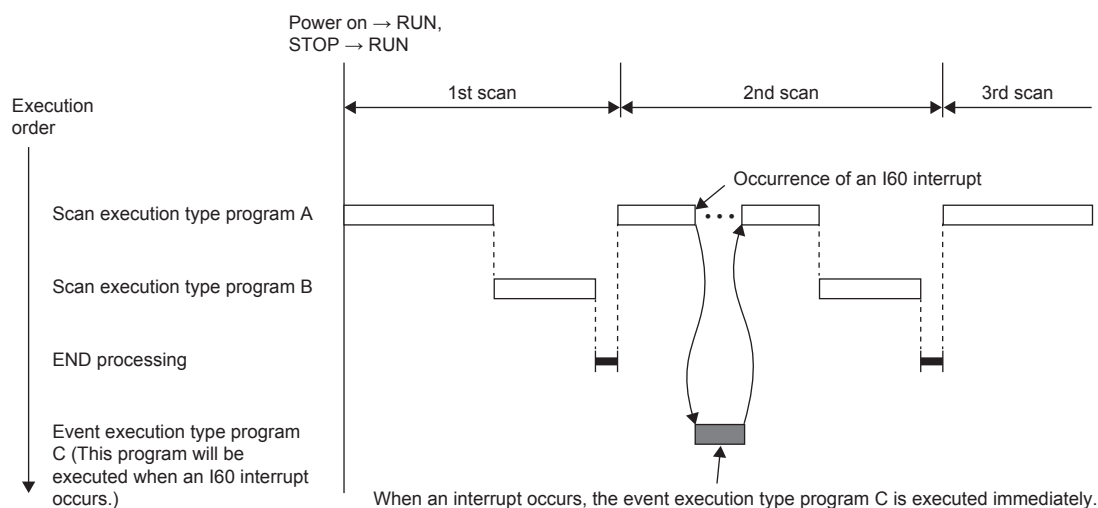
\*1 Measurement of elapsed time is 10ms or more because it is determined depending on the scan time.

### Trigger type

The following lists the triggers for the event execution type programs.

#### (1) Interrupt occurrence by interrupt pointers (I)

When the specified interrupt factor occurs, the program is immediately executed once. However, before that, an interrupt must be enabled by executing the EI instruction. Enabling programs to be independently executed as interrupt programs, this method eliminates the need to write the FEND instruction, interrupt pointers, and the IRET instruction within scan execution type programs.



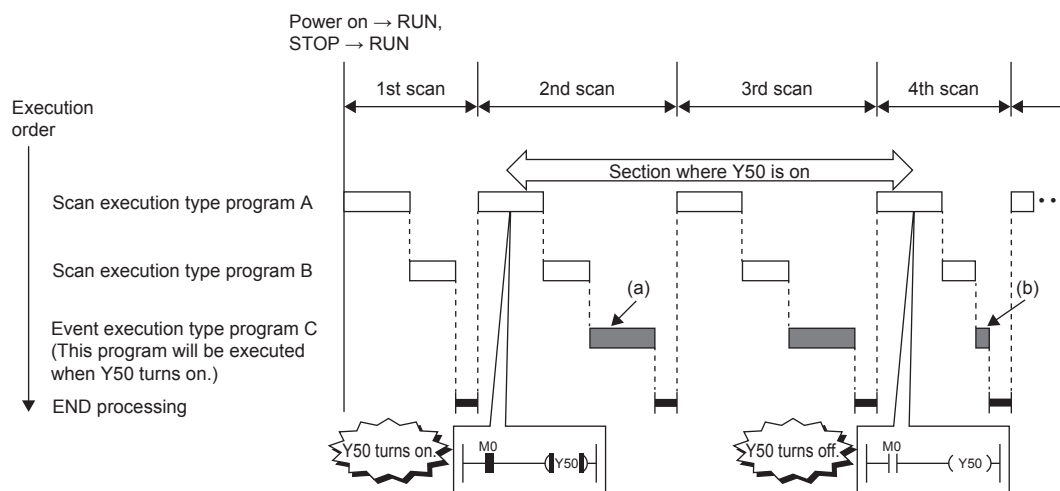
Interrupt pointers (I) which can be specified are I0 to I15, I28 to I31, I44, I45, I48, I49, and I50 to I1023.

#### Point

Execution conditions for the event execution type program which is triggered by an interrupt occurred by an interrupt pointer (I) are the same as those for general interrupt programs.

## (2) Turning on (TRUE) of bit data

Programs are executed in the execution order specified in the program setting of the CPU parameters, and when specified bit data is on (TRUE) when the execution turn of the corresponding event execution type program comes, the program is executed. The current values of the output (Y), timer (T), and long timer (LT) used in this program can be cleared at the execution turn that comes after the specified bit data is changed from on (TRUE) to off (FALSE).



- When Y50 is on at the execution turn of the event execution type program C, the program is executed. When Y50 is off, the program is not executed.
- When "Clear Output and Current Value of Timer" is set, the current values of the output (Y), timer (T), and long timer (LT) used in this program are cleared at the execution turn of the event execution type program that comes after Y50 is off.

The following shows applicable devices.

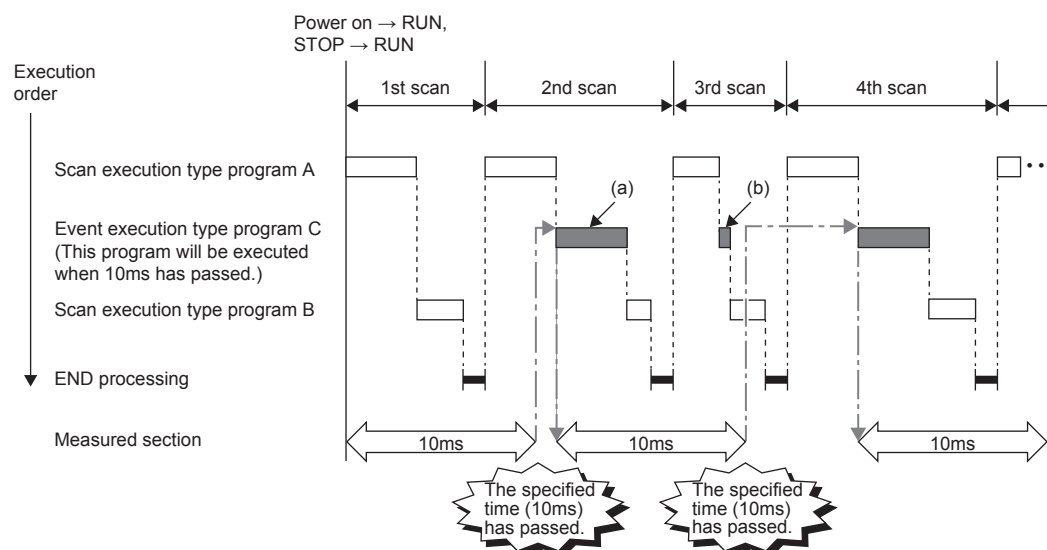
Item	Description	
Device <sup>*1</sup>	Bit device	X(DX), Y, M, L, F, SM, B, SB, Jn\X, Jn\Y, Jn\B, Jn\SB
	Bit specification of word device	D, SD, W, SW, R <sup>*2</sup> , ZR <sup>*2</sup> , RD, Un\G, Jn\W, Jn\SW

<sup>\*1</sup> A local device or index-modified device cannot be specified.

<sup>\*2</sup> This device is not available when "Use File Register of Each Program" is set. However, if no file register is assigned or if specification is out of the range, the specified bit data is not turned on (TRUE).

### (3) Passing time

After the status of the CPU module is changed into the RUN state, programs are executed in the execution order specified in "Program Setting" of "CPU Parameter". When the specified time has passed and the execution turn of the event execution type program comes, the program is executed once. The time is measured again from the start of the program execution, and the above operation is repeated. This method can be used when the processing is executed regularly after the specified time has passed. This method is not for executing an interrupt that interrupts another program execution to secure the punctuality. The current values of the output (Y), timer (T), and long timer (LT) used in this program can be cleared at the execution turn that comes after the specified time passes.



- When the specified time has passed and the execution turn of the event execution type program C comes, the program is executed.
- When "Clear Output and Current Value of Timer" is set, the current values of the output (Y), timer (T), and long timer (LT) used in this program are cleared at the execution turn of the event execution type program that comes after the specified time has passed.

#### Point

Even though "Clear Output and Current Value of Timer" is set to "Clear" in the CPU parameters, the output value and the current value of the timer are not cleared when the scan time is longer than the value set to "Passing Time".



# MEMO

---

# 3 BASIC OPERATIONS OF GX Works3 (MULTIPLE PROGRAMS)

---

## 3.1 Multiple Programs

---

3

### *Point*

- This section describes how to create multiple programs.
  - This section describes program execution control instructions.
  - This section describes the program list monitor.
- 

Execute the sequence programs divided into multiple files in the order set in parameters.

Check differences due to execution types of programs and check that each execution type can be changed with an instruction.

## 3.1.1 Creating multiple programs

Create multiple programs in "Write Mode".

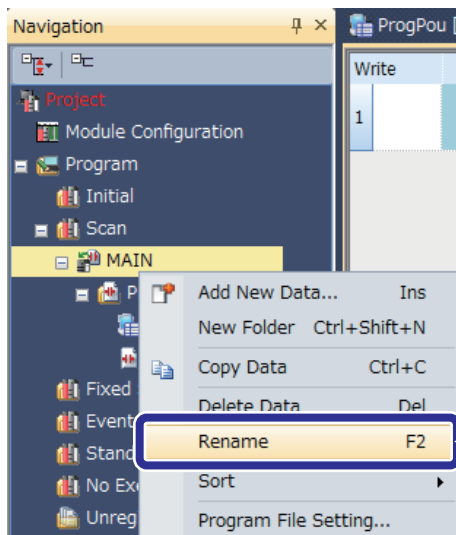
Create a new project with the file name "EX5.gx3".

(For how to create a new project, refer to Section 2.2.2 in the MELSEC iQ-R Series Basic Course (for GX Works3).)

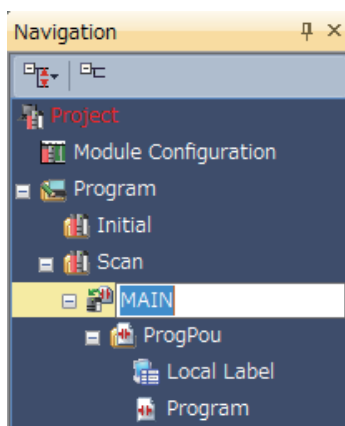
### Changing a program file name and a program block name

Change a program file name and a program block name.

#### Operating procedure



1. Right-click "MAIN" in the "Project" view and click "Rename".

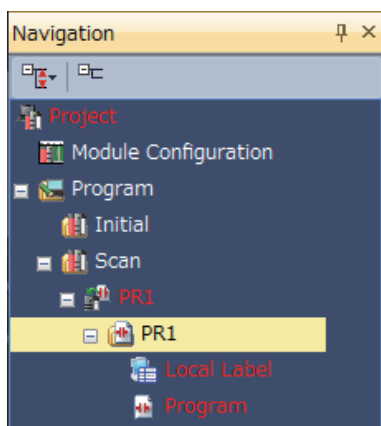
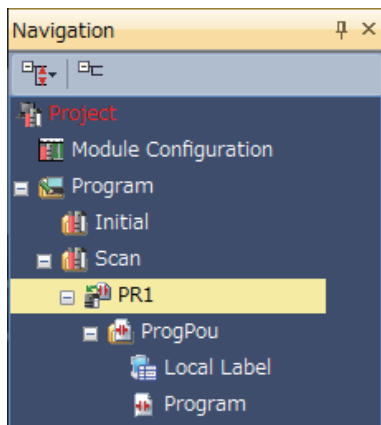


2. Enter "PR1" and change the data name.



(To the next page)

(From the previous page)



3. The program file name is changed from "MAIN" to "PR1".

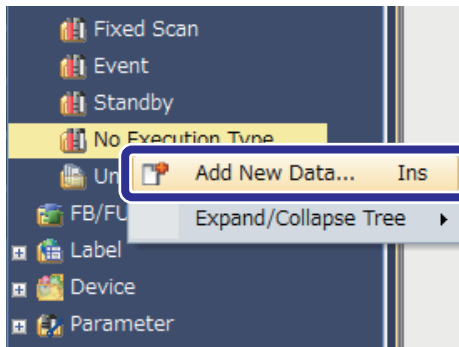
3

4. Change the program block name "ProgPou" to "PR1" in the same way.

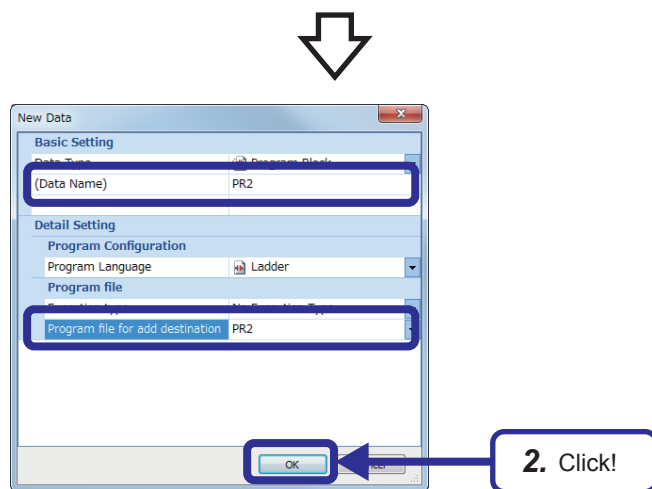
## Adding a program

Add a new program in the project.

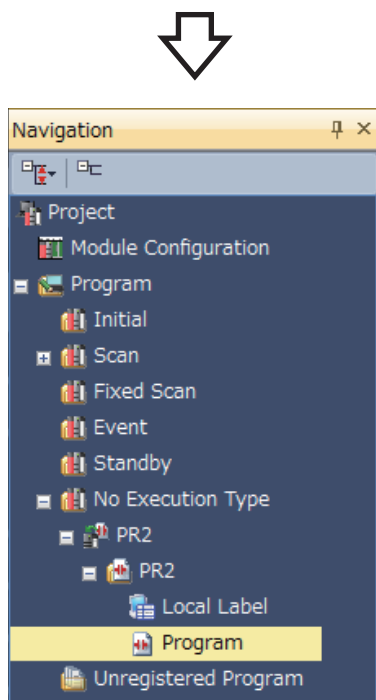
### Operating procedure



1. Right-click "No Execution Type" in the "Project" view and click "Add New Data".



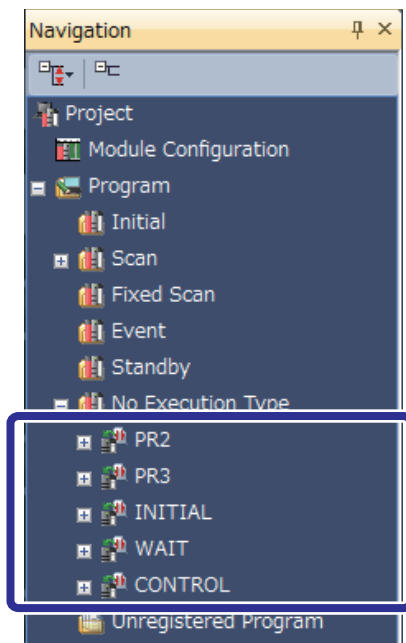
2. The "New Data" dialog box appears. Enter "PR2" in "Data Name" and "Program file for add destination", and click the [OK] button.



3. The program block "PR2" is newly created.

(To the next page)

(From the previous page)



4. Create all required programs in the same way.

## Creating the scan execution type program PR1

### (1) Program to be created

Create a program that counts the number of program executions and displays the value in seven segments.  
This program flickers Y170 and Y171.

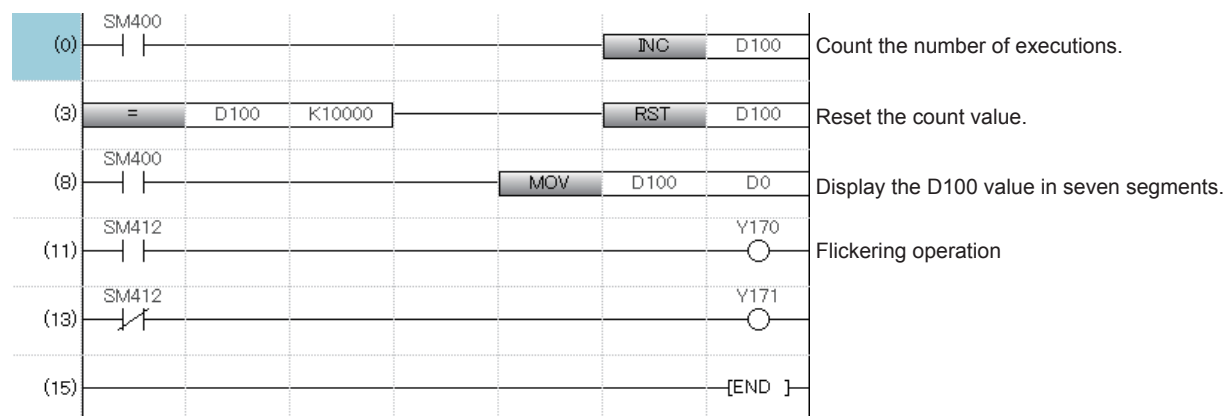
### (2) Device to be used

- D100 ..... : Register for counting the number of program executions
- D0 ..... : Device used to display the D100 value in seven segments
- Y170, Y171 ..... : Devices that flicker

### (3) Creating a program

Create the following program.

Project name	EX5
Program name	PR1



## Creating the scan execution type program PR2

### (1) Program to be created

Create a program that counts the number of program executions and displays the value in seven segments.  
This program flickers Y172 and Y173.

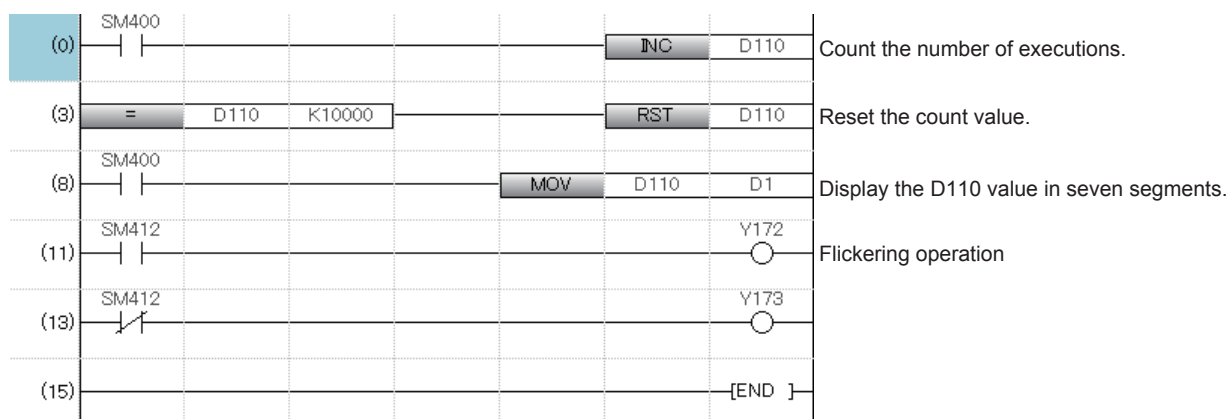
### (2) Device to be used

- D110 ..... : Register for counting the number of program executions
- D1 ..... : Device used to display the D110 value in seven segments
- Y172, Y173 ..... : Devices that flicker

### (3) Creating a program

Create the following program.

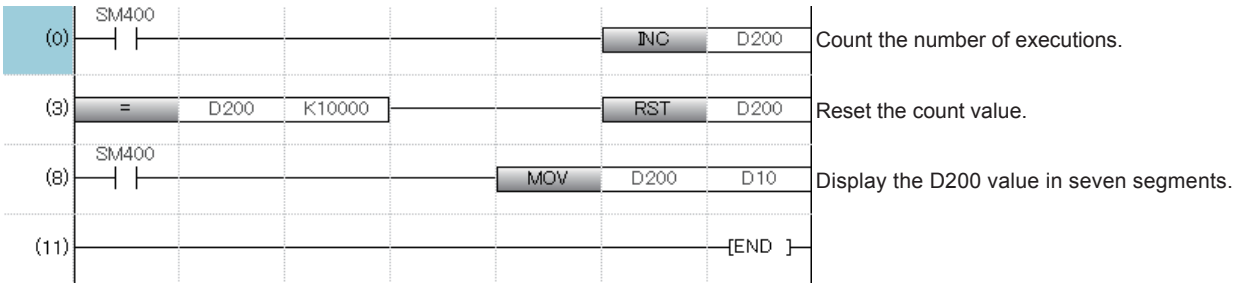
Project name	EX5
Program name	PR2



### Creating the fixed scan execution type program PR3

- (1) Program to be created  
Create a program that counts the number of program executions and displays the value in seven segments.
- (2) Device to be used
  - D200 ..... : Register for counting the number of program executions
  - D10 ..... : Device used to display the D200 value in seven segments
- (3) Creating a program  
Create the following program.

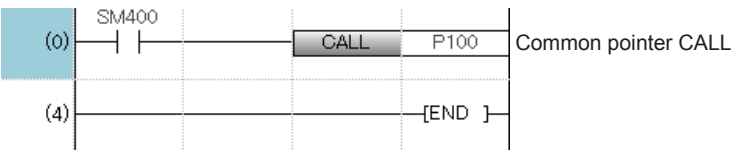
Project name	EX5
Program name	PR3



### Creating the initial execution type program INITIAL

- (1) Program to be created  
Create a program that resets the register for storing the numbers of executions of PR1, PR2, and PR3 at the start of operations. WAIT (standby type program) contains the reset processing.
- (2) Device to be used
  - P100 ..... : Common pointer to WAIT
- (3) Creating a program  
Create the following program.

Project name	EX5
Program name	INITIAL

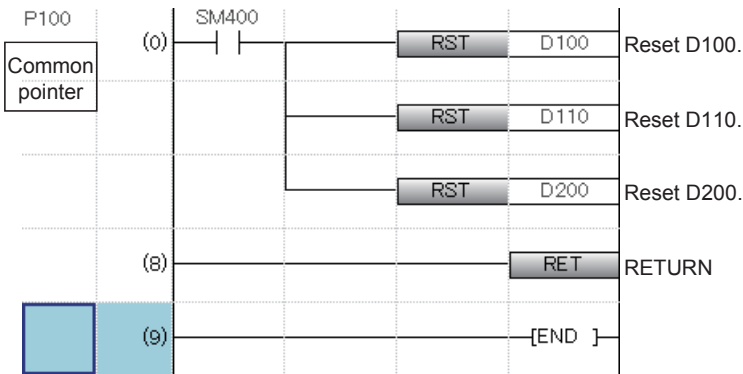




Creating the standby type program WAIT

- (1) Program to be created
- Create a program that resets the register for storing the numbers of executions of PR1, PR2, and PR3. This program is called from the INITIAL or CONTROL program and executed.
- (2) Device to be used
- P100 ..... : Common pointer to WAIT
  - D100, D110, D200... : Initialization targets
- (3) Creating a program
- Create the following program.

Project name	EX5
Program name	WAIT



Point

When creating a common pointer, move the cursor to the position to insert the pointer in the pointer input areas and enter "P100". For details, refer to the following.

GX Works3 Operating Manual

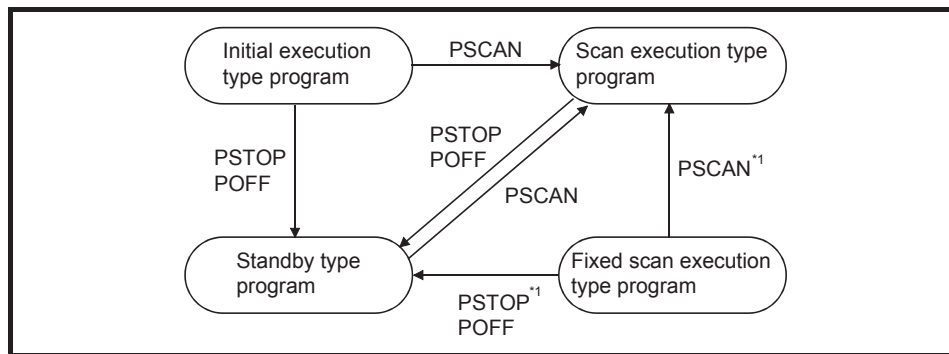
## 3.1.2 Creating control programs

Create programs that control the operating status of each of the created programs.

### Program execution control instructions

Program execution control instructions change the execution types of programs when the CPU module is in the RUN state. The following three types of program execution control instructions are provided.

- PSCAN instruction
- POFF instruction
- PSTOP instruction



\*1 Once a fixed scan execution type program is changed to another execution type program, it cannot be restored to the fixed scan execution type.

The POFF instruction is used to securely turn off external outputs (Y) when a program is changed to a standby type program.

#### Point

Program execution control instructions change the execution type of the program that has been read in the program memory.

These instructions do not change the execution type of the program that has not been read in the program memory from a memory card.

## Creating the initial execution type program INITIAL

### (1) Program to be created

- PSTOP: Turning on/off the switch X102 switches the execution type of PR1 between the scan execution type and standby type.
- POFF: Turning on/off the switch X103 switches the execution type of PR2 between the scan execution type and the standby type.
- EI/DI: Turning on/off the switch X104 enables/disables interrupts of PR3.
- Turning on the switch X100 calls the WAIT program and resets the register for storing the numbers of executions of PR1, PR2, and PR3.

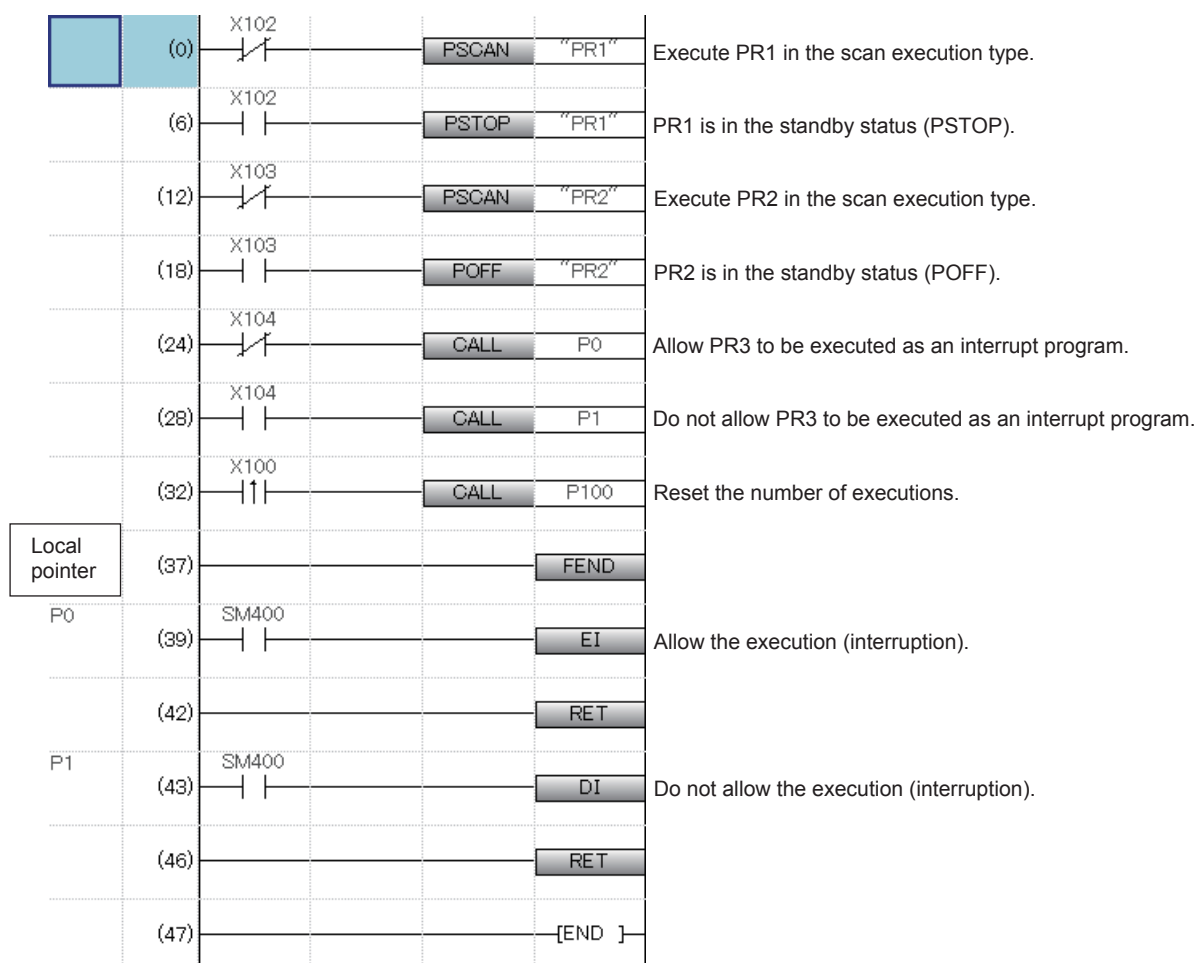
### (2) Device to be used

- X100 : Turning on this device resets the register for storing the numbers of executions of PR1, PR2, and PR3.
- X102 : Turning on this device switches the execution type of PR1 to the standby type (PSTOP).  
: Turning off this device switches the execution type of PR1 to the scan execution type.
- X103 : Turning on this device switches the execution type of PR2 to the standby type (POFF).  
: Turning off this device switches the execution type of PR2 to the scan execution type.
- X104 : Turning on this device disables execution of PR3.  
: Turning off this device enables execution of PR3.
- P0, P1 : Local pointers
- P100 : Common pointer

### (3) Creating a program

Create the following program.

Project name	EX5
Program name	CONTROL



### 3.1.3 Setting parameters

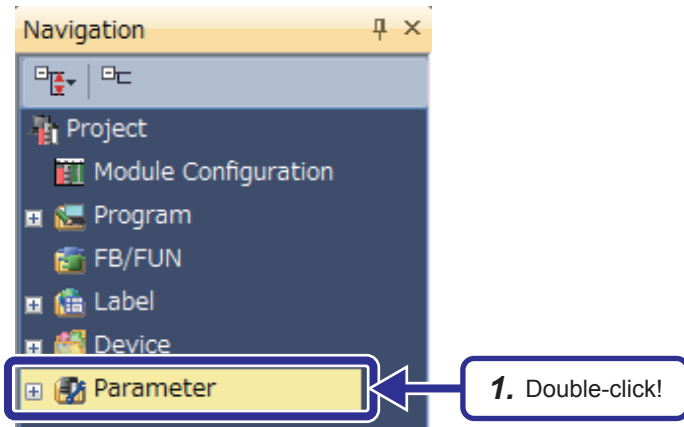
Set parameters for controlling the created sequence programs and write the parameters and the programs together to the CPU module.

#### Program setting

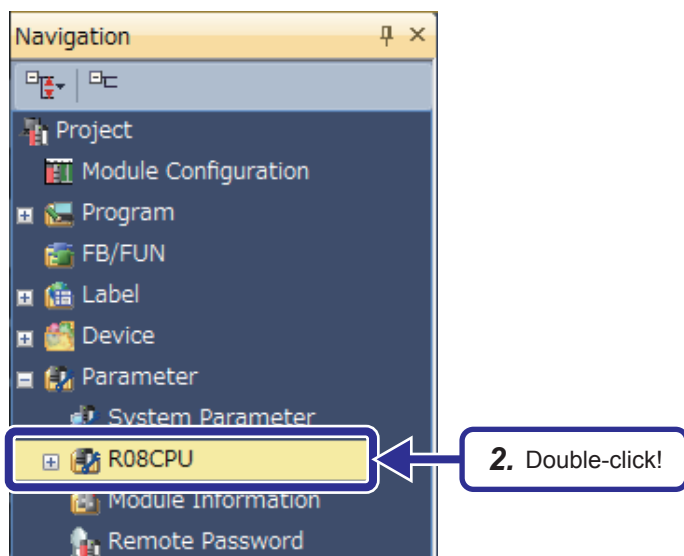
##### Operating procedure

3

1. Double-click "Parameter" in the "Project" view.

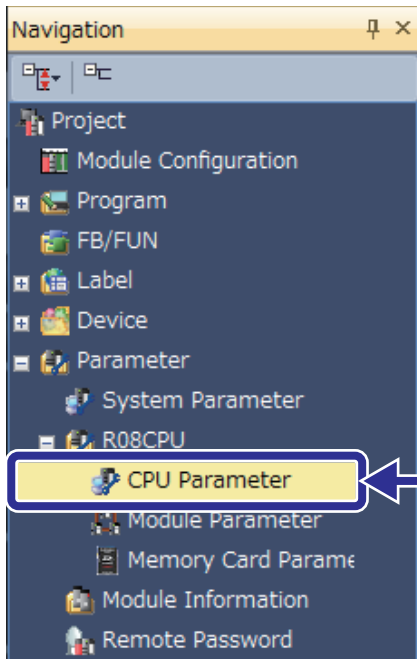


2. Double-click "R08CPU".

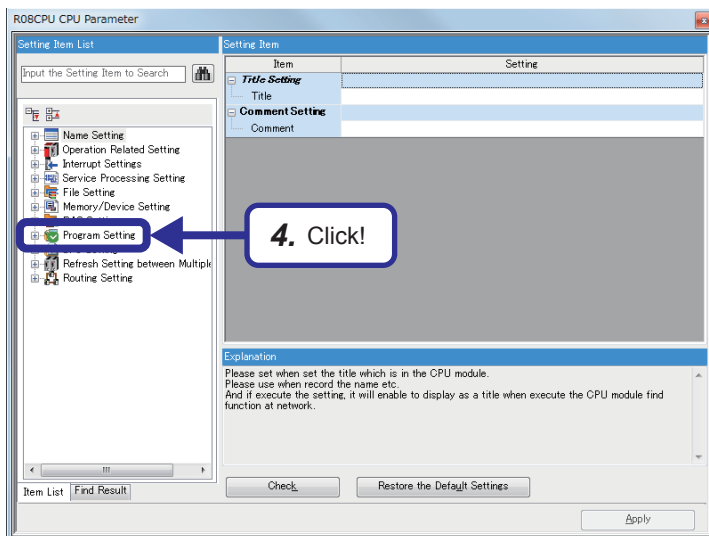


(To the next page)

(From the previous page)



**3.** Double-click "CPU Parameter".

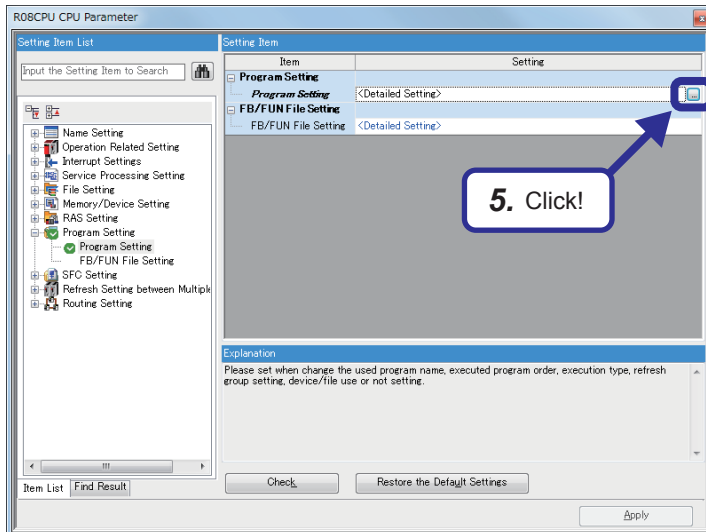



**4.** The "R08CPU CPU Parameter" dialog box appears. Click "Program Setting" in "Setting Item List".



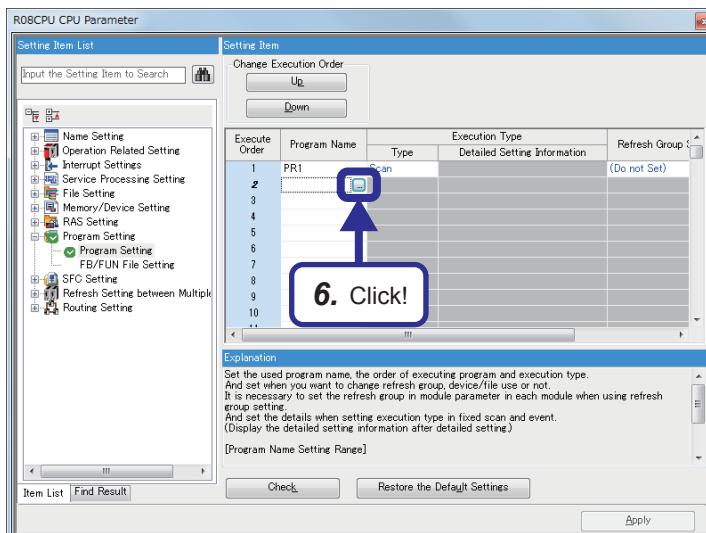
(To the next page)


(From the previous page)

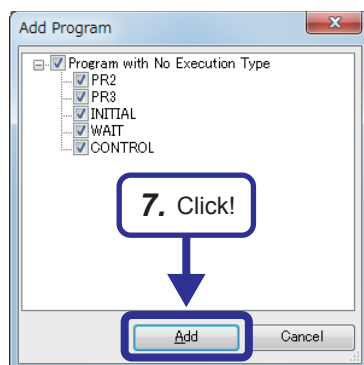


5. Click "Detailed Setting" of "Program Setting", and click the  button.

3



6. After the window is switched, click the second cell in the "Program Name" column and click the  button.

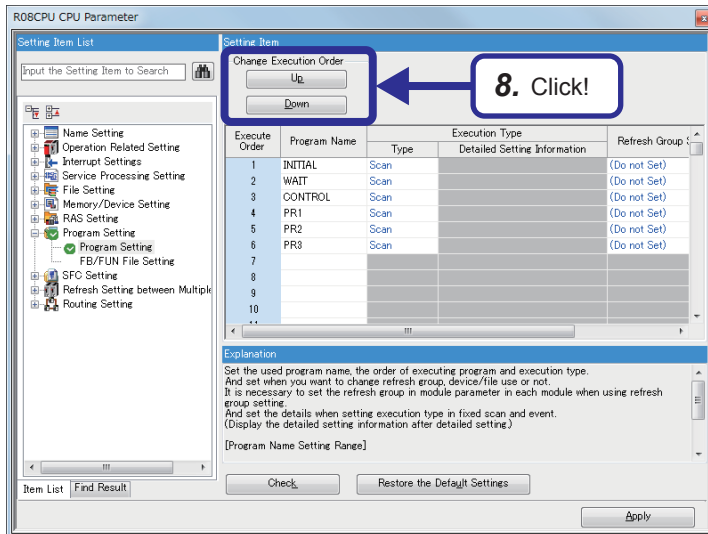




7. The "Add Program" dialog box appears. Check that the programs to be added have been checked, and click the [Add] button.



(To the next page)

(From the previous page)



8. Click the  /  button of "Change Execution Order" to change the execution order of the programs as follows.

1st cell → INITIAL

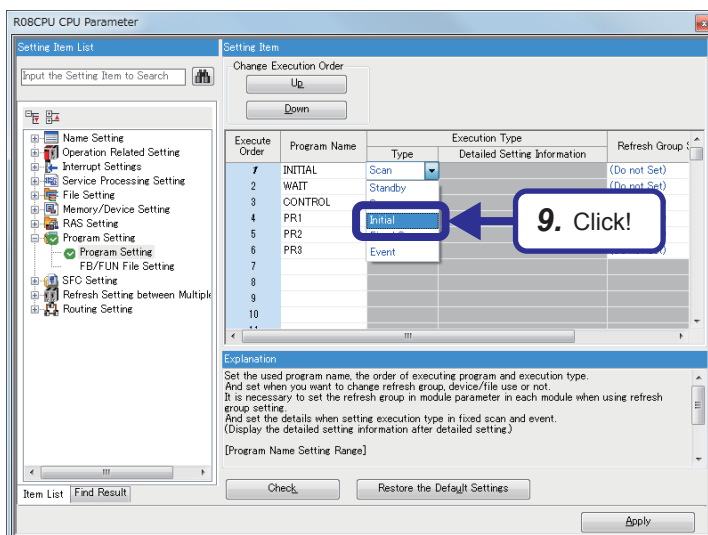
2nd cell → WAIT


3rd cell → CONTROL

4th cell → PR1

5th cell → PR2

6th cell → PR3

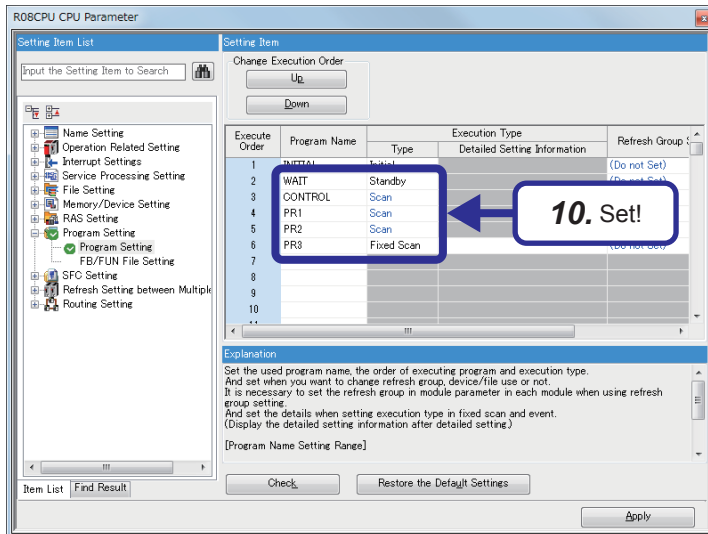


9. Click the  button in "Type" of "INITIAL" and set "Initial".



(To the next page)

(From the previous page)



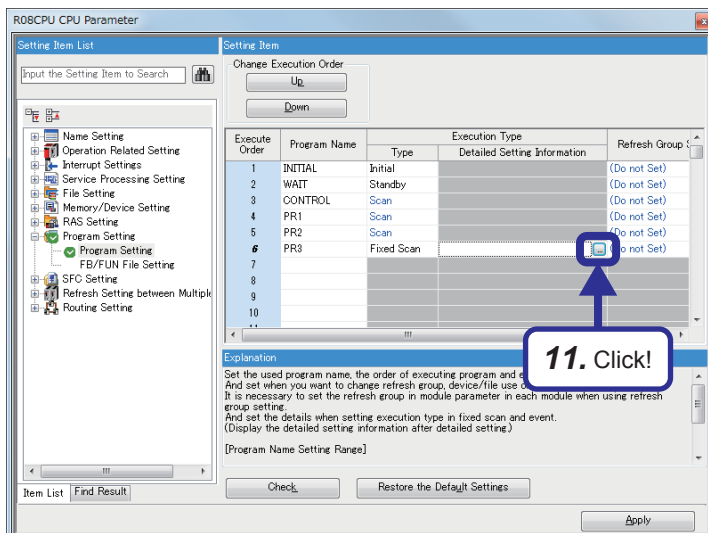
10. Set the parameters with the same procedure described in 9.


[Parameters to be set]

WAIT → Standby

PR3 → Fixed Scan

3



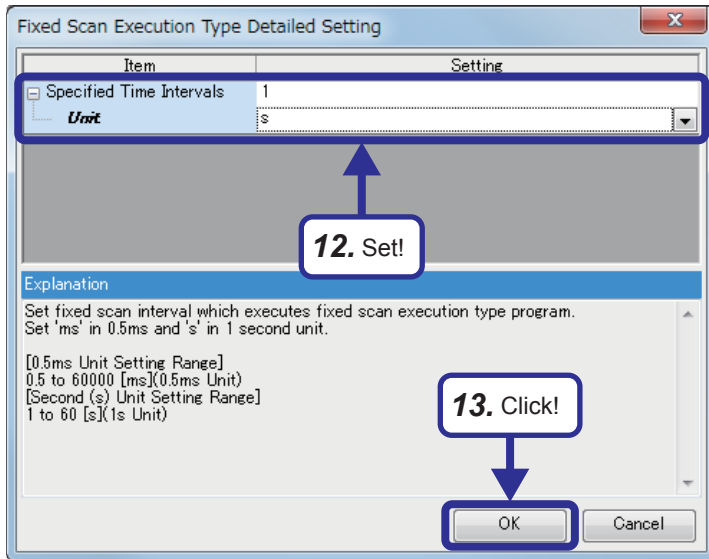
11. Click "Detailed Setting Information" of "PR3", and click the  button.



(To the next page)



(From the previous page)



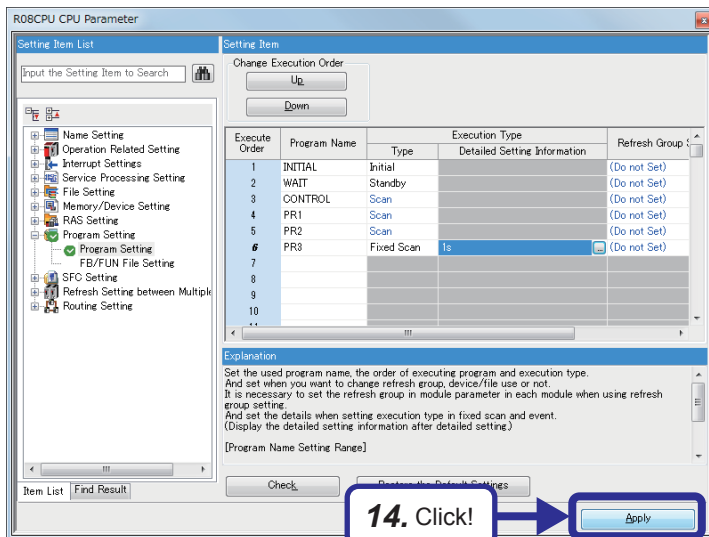
**12.** The "Fixed Scan Execution Type Detailed Setting" dialog box appears. Set the following parameters.

[Parameters to be set]

Specified Time Intervals: 1

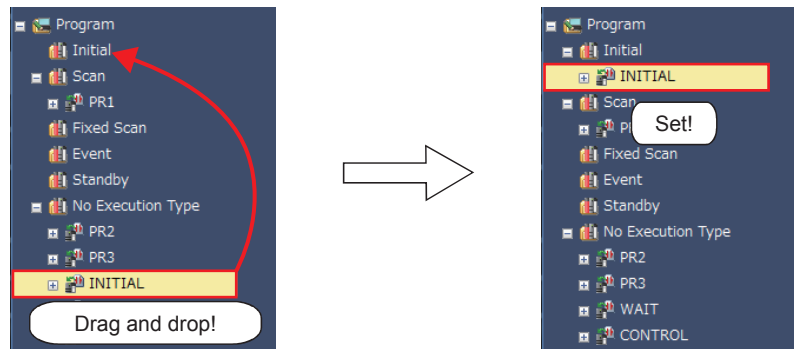
Unit: s

**13.** Click the [OK] button.



**14.** After the setting, click the [Apply] button and click [Project] → [Save] to save the setting.

Users can set the program type of each program by dragging and dropping a program on the "Navigation" window.



## Writing data to the CPU module

Write the created programs (PR1, PR2, PR3, INITIAL, WAIT, and CONTROL) and parameters to the CPU module.

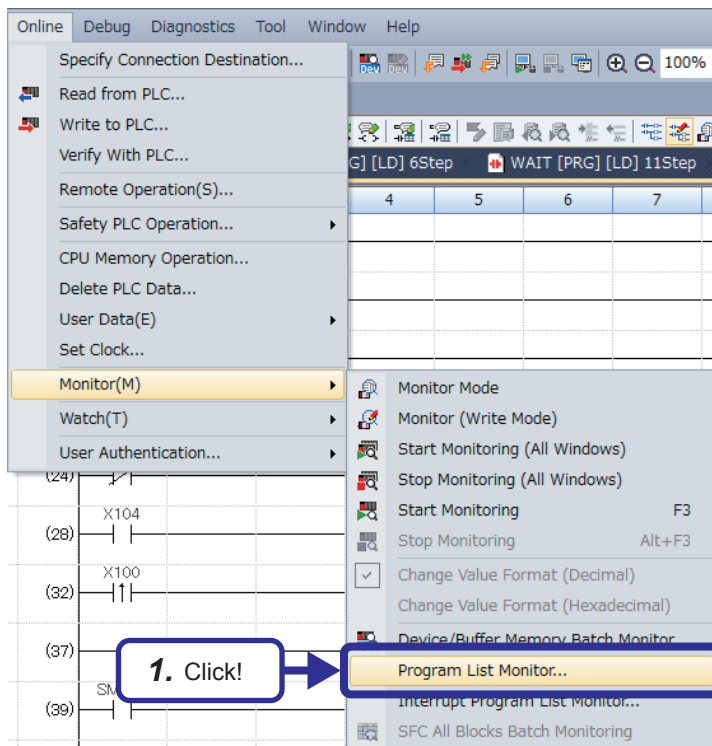
## 3.2 Monitoring

### 3.2.1 Program list monitor

Monitor executions of the multiple programs created in Section 3.1.

Before monitoring, reset the CPU module and set the module in the RUN state.

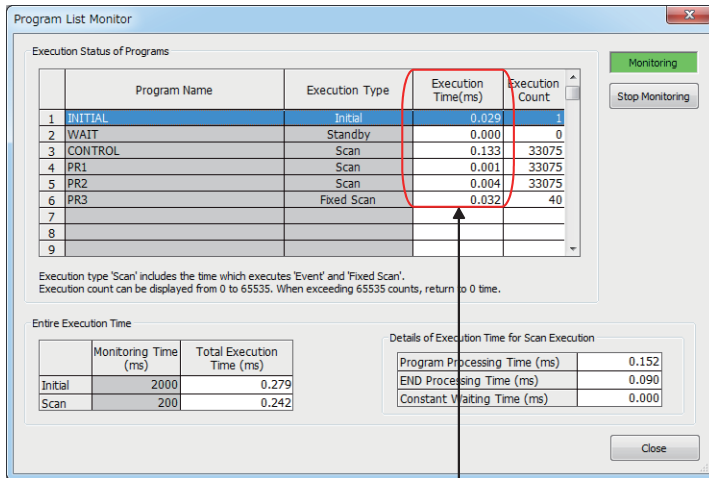
#### Operating procedure



1. Click [Online] → [Monitor] → [Program List Monitor] from the menu.

  
(To the next page)

(From the previous page)



With this setting, the execution time display does not change because the actual program processing time is short.

2. The "Program List Monitor" dialog box appears. Turn on and off the switches X100, X102, X103, and X104 to check that the execution status of each program changes.

[Execution status changes]

X100 : Turning on this device calls the WAIT program and resets the number of executions.

X102 : Turning on this device switches the PR1 to the standby type. (PSTOP)  
: Turning off this device switches the PR1 to the scan execution type. (PSCAN)

X103 : Turning on this device switches the PR2 to the standby type. (POFF)  
: Turning off this device switches the PR2 to the scan execution type. (PSCAN)

X104 : Turning on this device disables execution of PR3.  
: Turning off this device enables PR3 execution of PR3.

## "Execution Status of Programs"

This area displays the execution status of each program set in "Program Setting" of the CPU parameters.

- (1) "Program Name"  
This area displays the program names in the order set in "Program Setting".
- (2) "Execution Type"  
This area displays the execution type of each program set in "Program Setting".
- (3) "Execution Time"  
This area displays the actual execution time (current value) of each program.  
When a program has stopped (in the standby status), "0.000ms" is displayed as the execution time of the program.  
Display contents of programs vary depending on the execution type of each program.  
\* When the actual program processing time is short, the execution time display does not change.
- (4) "Execution Count"  
This area displays the number of executions counted starting from 0 at the point of when the CPU module is set to the RUN state.  
The number of executions is held even after the program has stopped.

## "Entire Execution Time"

This area displays the monitoring time set in "RAS Setting" of the CPU parameters and the total execution time of each program type.

- (1) "Monitoring Time"  
This area displays each monitoring time of scan programs and initial programs.  
When the execution time exceeds this time, a watchdog timer error is detected in the CPU module.
- (2) "Total Execution Time"  
This area displays the total scan time of the programs being executed by the CPU module.  
Scan programs include END processing time.

## "Details of Execution Time for Scan Execution"

This area displays details of the execution time.

- (1) "Program Processing Time"  
This area displays the total execution time of scan programs.
- (2) "END Processing Time"  
This area displays the END processing time.
- (3) "Constant Waiting Time"  
This area displays the waiting time of the constant scan when the constant scan has been set.

# 4 FUNCTIONS OF RCPU

## 4.1 Constant Scan Function

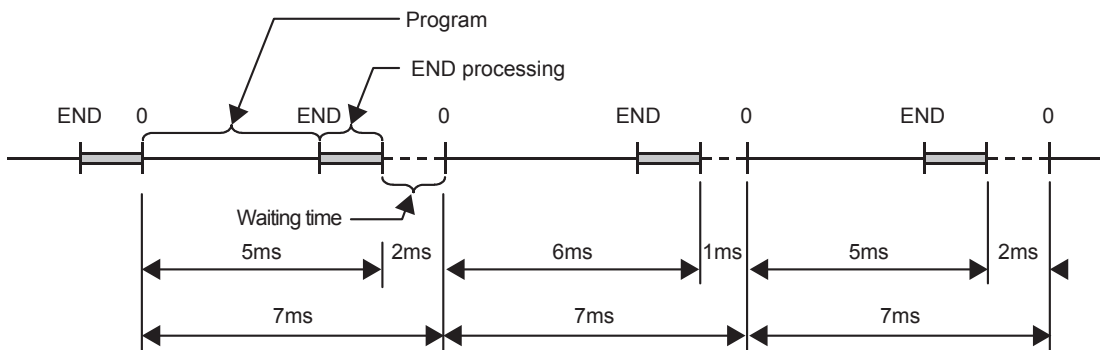
### Point

This section describes the function that keeps scan time constant.

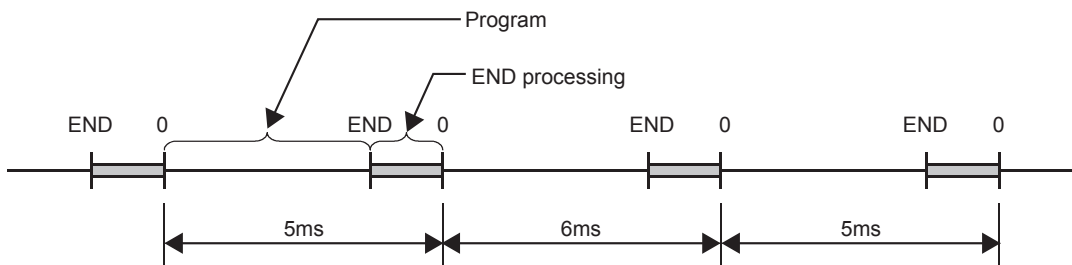
Scan time changes in each scan because its processing time varies depending on whether instructions used in a program are executed or not.

With constant scan, the I/O refresh interval can be kept constant even when the program execution time varies because the program can be executed repeatedly with the constant scan time.

- When constant scan is set (Setting value = 7ms)



- When constant scan is not set



### Point

Set the constant scan time in "RAS setting" of "CPU Parameter".

The setting time for the constant scan must be a value that satisfies the relational expression below.

"WDT setting time" > "Constant scan setting time" > "Maximum scan time of the program"

If the maximum scan time of the program is longer than the constant scan setting time, it is assumed as an error to ignore the constant scan, and the scan time of the program is applied.

## Accuracy of constant scan

The accuracy of constant scan is 0.01ms. However, if the processing, which should be executed during waiting time from the completion of the END processing to the start of the subsequent scan, is executed, the constant scan cannot be completed even though the constant scan time has come. The constant scan may increase by the program execution time of an interrupt factor.

## Precautions

Processing of a program stops during waiting time from the completion of the END processing to the start of the subsequent scan. When any of the following processing requests is issued during waiting time, the corresponding processing will be executed.

- Interrupt program
- Fixed scan execution type program
- Event execution type program triggered by occurrence of an interrupt
- Device/label access service processing

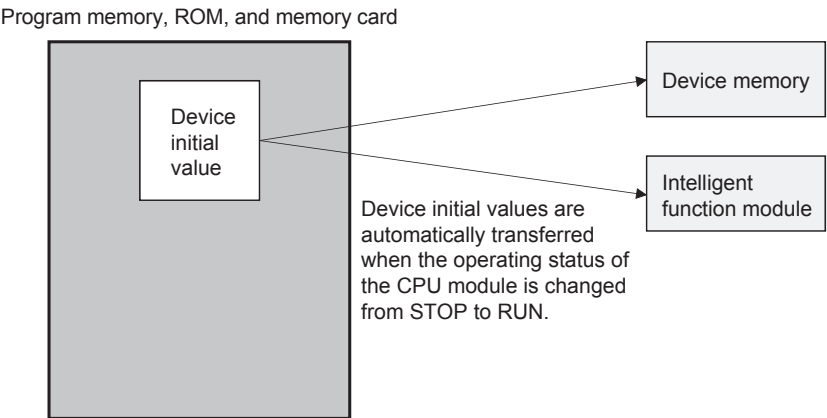
# 4.2 Device Initial Value



This section describes how to set initial values in devices.

To use device initial values, set the values in advance. When the system is powered on, the CPU module is reset, or the operating status of the CPU module is changed from STOP to RUN, data in a specified device initial value file is automatically transferred to specified devices or the buffer memory of an intelligent function module. Thus, no program for setting device data is required.

Ex.



## Required setting

To use the device initial value setting function, set "Device Memory" and "Device Initial Value" in the "Project" view. Write the device initial value file to which these settings have been reflected to the program memory, ROM, or a memory card of the RCP.

## Devices where initial values can be set

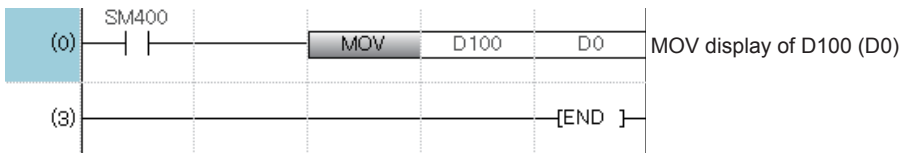
The following table lists devices where device initial values can be set.

Device name	Symbol	Device name	Symbol
Timer	T	Special register	SD
Retentive timer	ST	Link register	W
Long timer	LT	Link special register	SW
Retentive long timer	LST	Module access device	G
Counter	C	CPU buffer memory access device	G/HG
Long counter	LC	File register	ZR
Data register	D	Module refresh register	RD

## Preparations for initial value setting operation check

Create the following program to easily check the operation.

Project name	EX6
Program name	MAIN



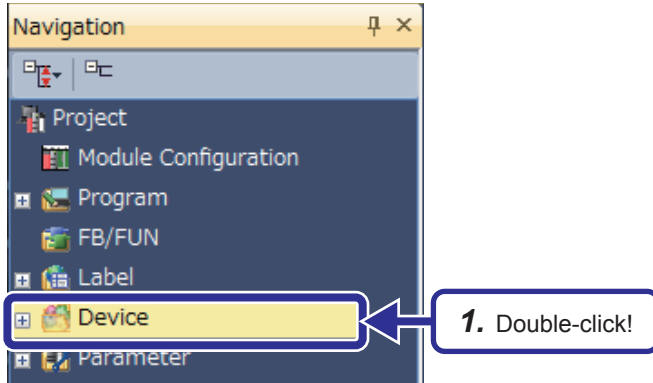


## 4.2.1 Setting values in the device memory

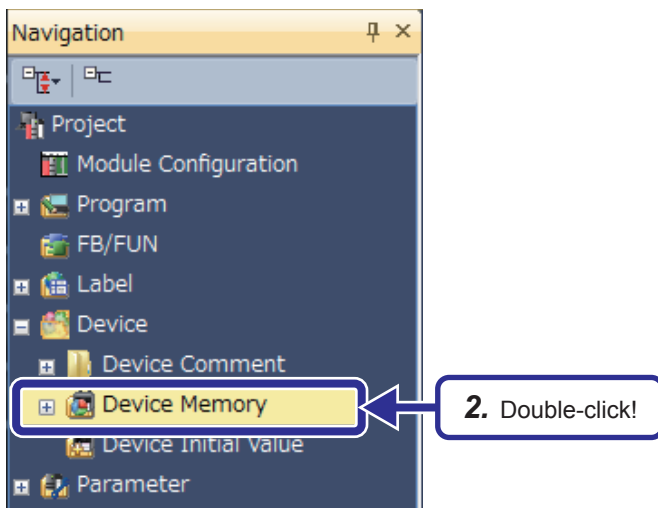
- (1) Set device initial values in the device memory.  
Set "1234" in D100 with the following procedure.

### Operating procedure

- 1.** Double-click "Device" in the "Project" view.



- 2.** Double-click "Device Memory".

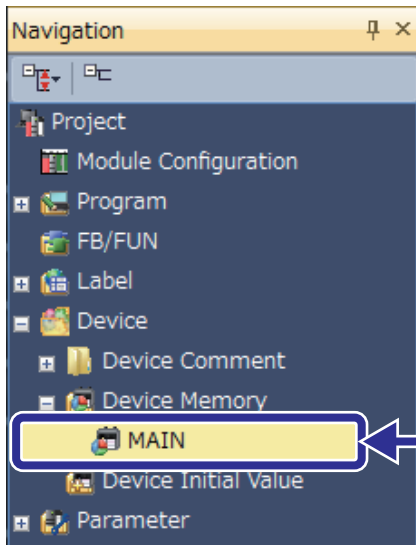


(To the next page)

(From the previous page)



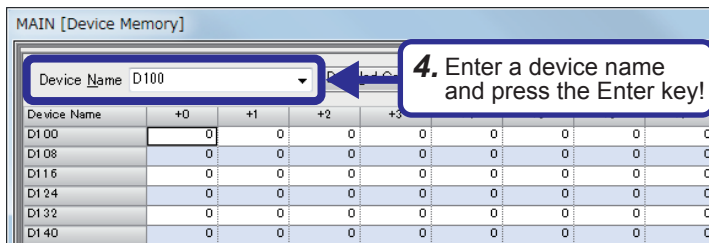
3. Double-click "MAIN".



3. Double-click!

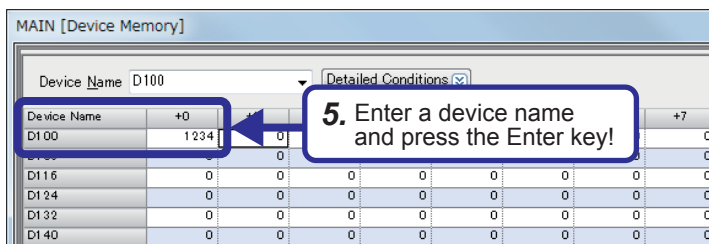


4. The "MAIN [Device Memory]" dialog box appears. Enter "D100" for "Device Name" and press the  key.



5. Enter "1234" in the cell of D100 and +0, and press the  key.

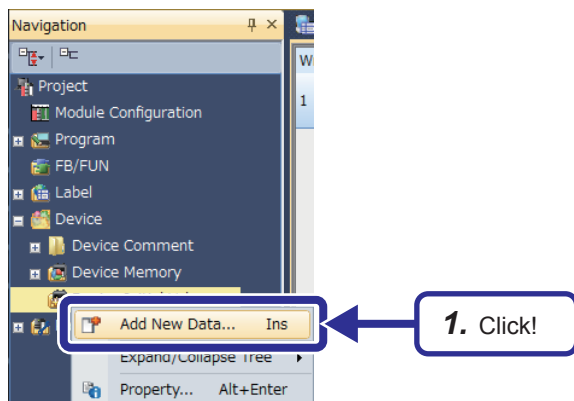
The initial value is set in D100.



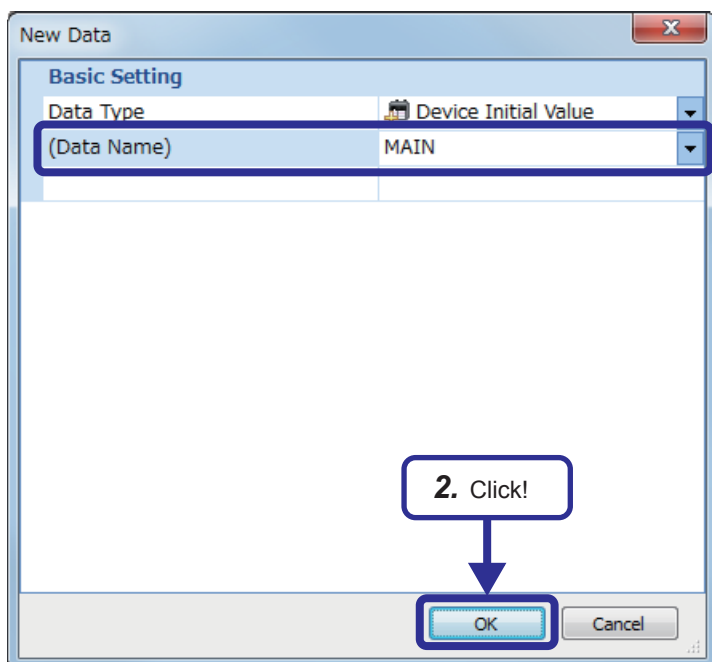
- (2) Set the range of devices where device initial values are stored.  
Set device initial values in D100 to D109.

## Operating procedure

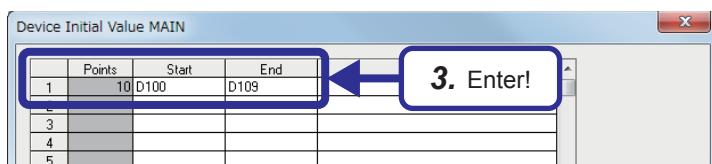
1. Click "Device", right-click "Device Initial Value", and click [Add New Data] from the menu.



2. The "New Data" dialog box appears. Check that "Data Name" is "MAIN" and then click the [OK] button.

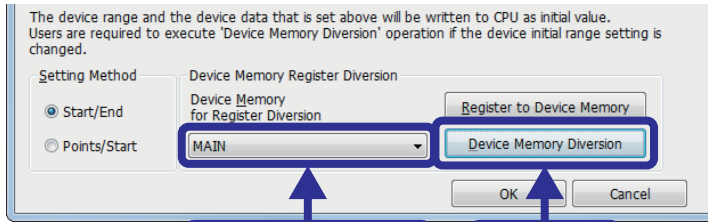


3. The "Device Initial Value MAIN" dialog box appears. Enter "D100" in "Start", and "D109" in "End".



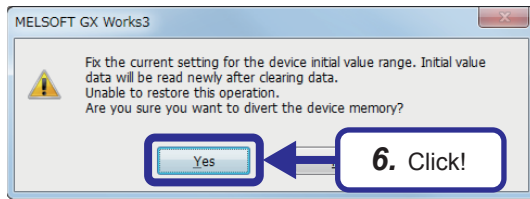
(To the next page)

(From the previous page)

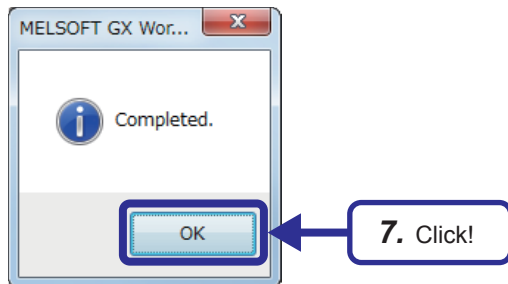


4. Select this item!

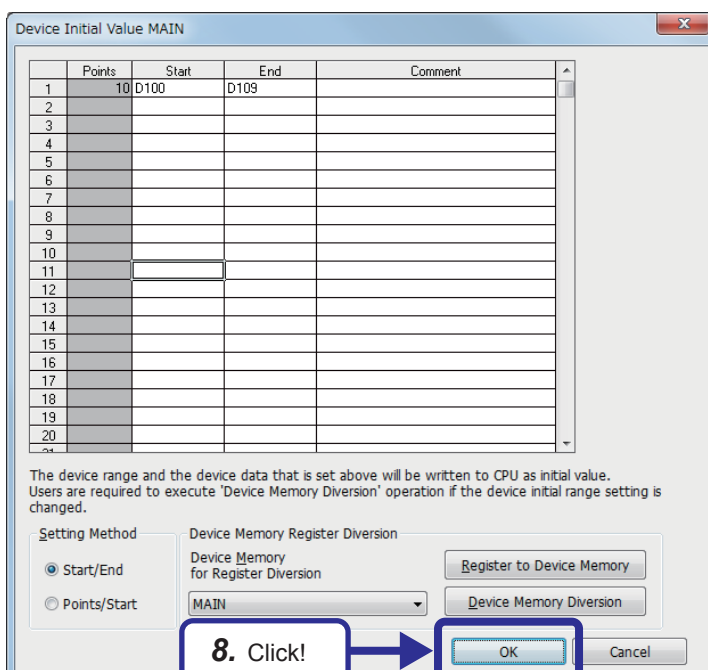
5. Click!



6. Click!



7. Click!



8. Click!

4. Click  of "Device Memory for Register Division" and select "MAIN".

5. Click the [Device Memory Division] button.

6. The dialog box shown on the left appears. Click the [OK] button.

7. Click the [OK] button.

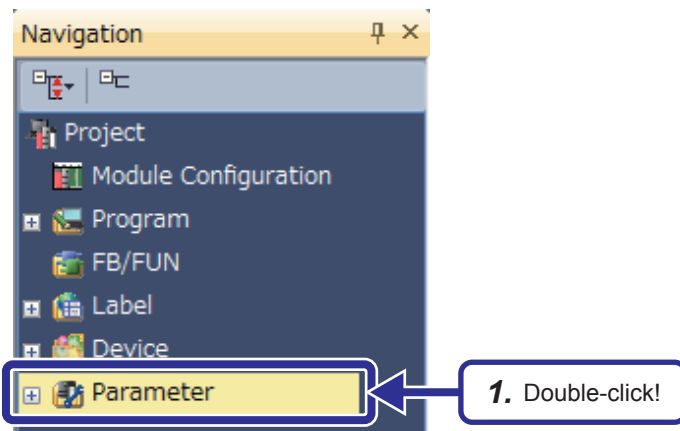
8. Click the [OK] button to close the dialog box.

## 4.2.2 Setting a device initial value file name

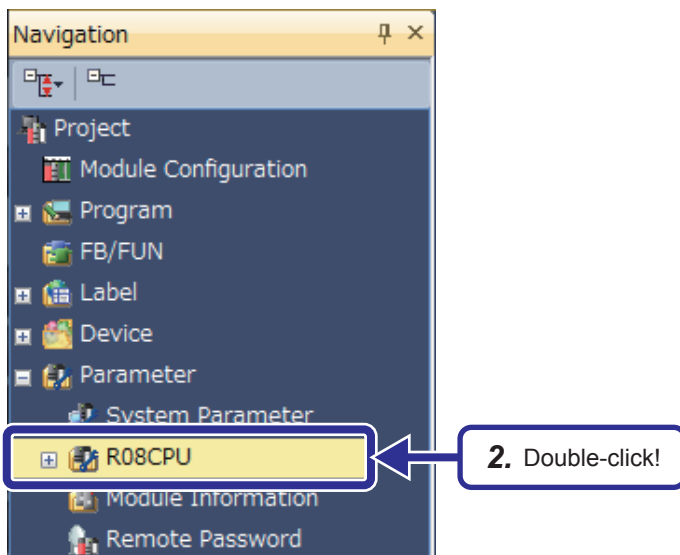
Set a file to be used as a device initial value file in "File Setting" of "CPU Parameter".

### Operating procedure

1. Double-click "Parameter" in the "Project" view.

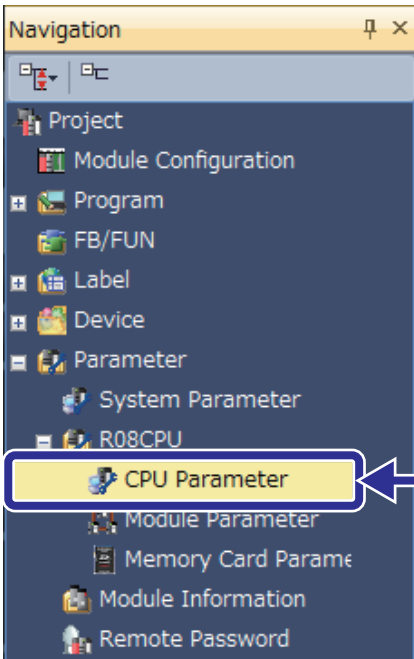


2. Double-click "R08CPU".



(To the next page)

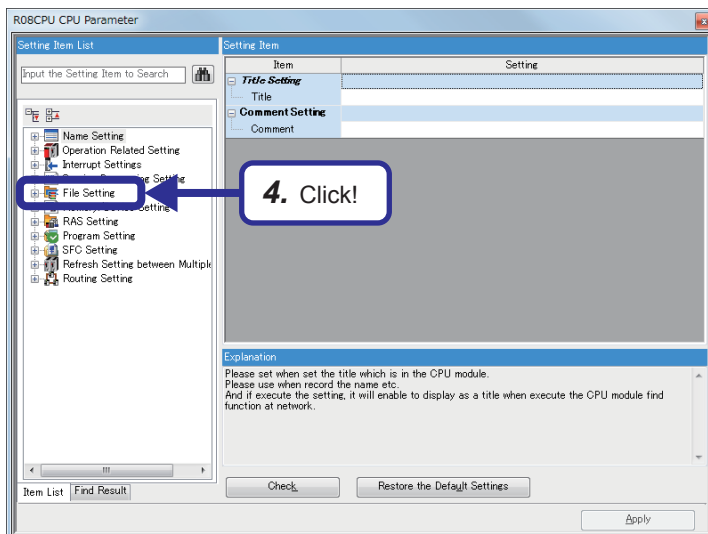
(From the previous page)



3. Double-click "CPU Parameter".

4

3. Double-click!



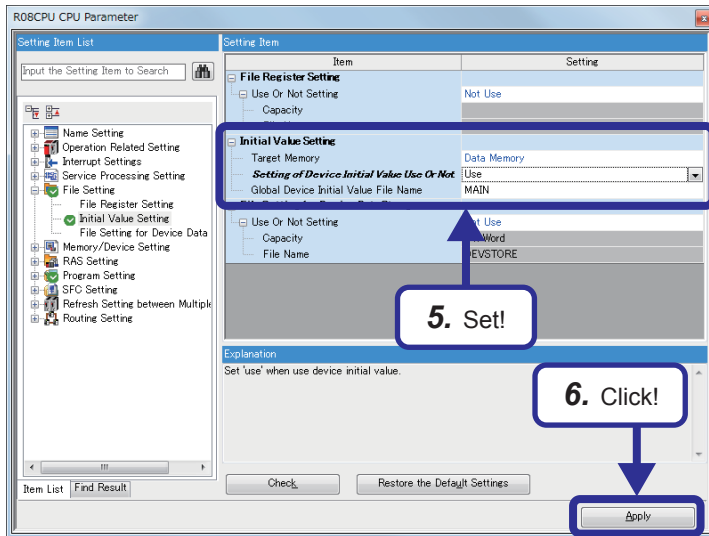
4. The "R08CPU CPU Parameter" dialog box appears. Click "File Setting" in "Setting Item List".

4. Click!



(To the next page)

(From the previous page)



5. The window is switched. Set the items of "Initial Value Setting" as follows.

[Parameters to be set]

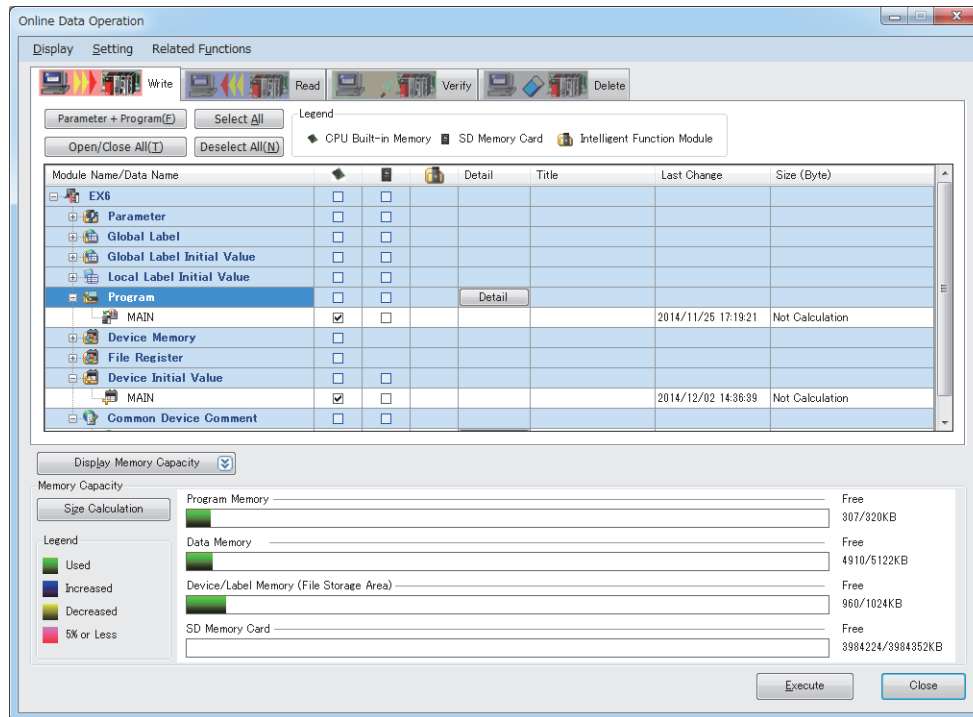
Setting of Device Initial Value Use Or Not: Use  
Global Device Initial Value File Name: MAIN

6. After the setting is completed, click the [Apply] button.

## 4.2.3 Checking the operation of device initial values

### (1) Writing data to the CPU module

Write the programs, parameters, and device initial values to the CPU module.



4

### Point

In this exercise, the device initial values can also be written to the programmable controller at the same time with parameters and programs.

Reason: The device initial values, programs, and parameters use the same target memory.

(In this exercise, the program memory is used for all data.)

Generally, the target memory area for the device initial values differs from the one for programs and parameters. Each of them can be separately written to each target memory.

### (2) Operation check

When the RUN/STOP/RESET switch of the CPU module is switched from STOP to RUN or when the system is powered on, the values in the device initial values are transferred to devices of the CPU module.

In this exercise, the initial indication value (D0) of the demonstration machine displays "1234", the initial setting value of the register D100.

### Point

When the CPU device areas are within the latch range, the priority is given to the device initial values.



# 4.3 Security

## Point

This section describes how to set a security key to a project.

To prevent unauthorized browsing of programs (in units of program files) with a security key, use the security key authentication function.

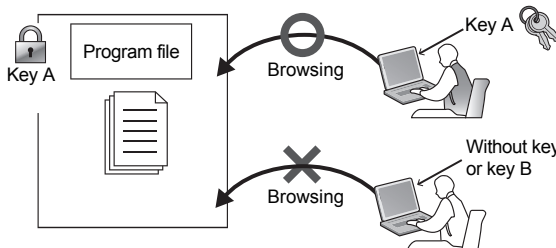
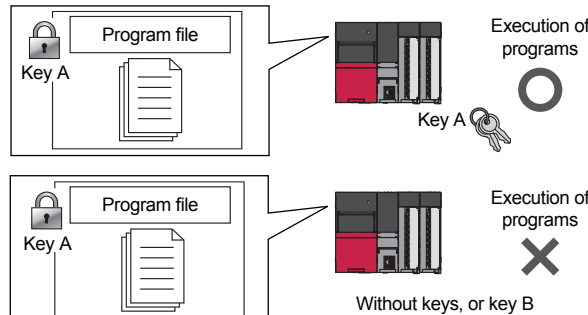
### Security key authentication function

Operations on a project can be restricted by setting a security key to the project.

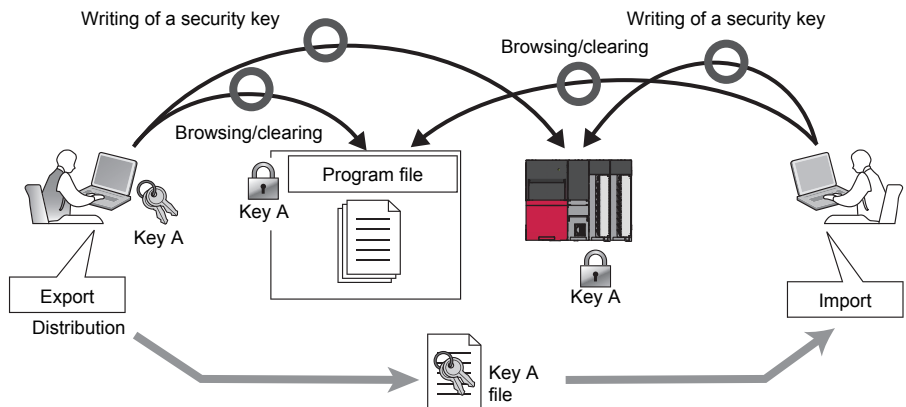
Targets of security key settings

Target	Application
■Program files (program files, FB files, FUN files) Although security keys can be registered in units of program files, all program files in a project will use a single security key.	<ul style="list-style-type: none"><li>• To restrict browsing of program files in a project</li><li>• To restrict execution of programs with a CPU module</li></ul>
■Personal computer Multiple security keys can be registered to one personal computer.	To enable browsing of the program files protected with security keys
■CPU module Only one security key can be written to one CPU module.	To enable execution of the program restricted with a security key

The following two operations can be restricted.

Browsing of programs (program files, FB files, FUN files)	Execution of programs in the CPU module (STOP/PAUSE → RUN)
Operations are restricted by detecting mismatches of the security keys of a program file and personal computer. 	Operations are restricted by detecting mismatches of the security keys of a program file written to the CPU module and the CPU module. 

Copying of a security key: The security key registered to a personal computer can be copied by importing/exporting it from/to another personal computer. When a security key is copied, the expiration date of the security key can be set.



### Precautions

When GX Works3 is uninstalled from a personal computer, the security key registered to the computer will not be deleted. Delete the security key on the "Security Key Management" window.

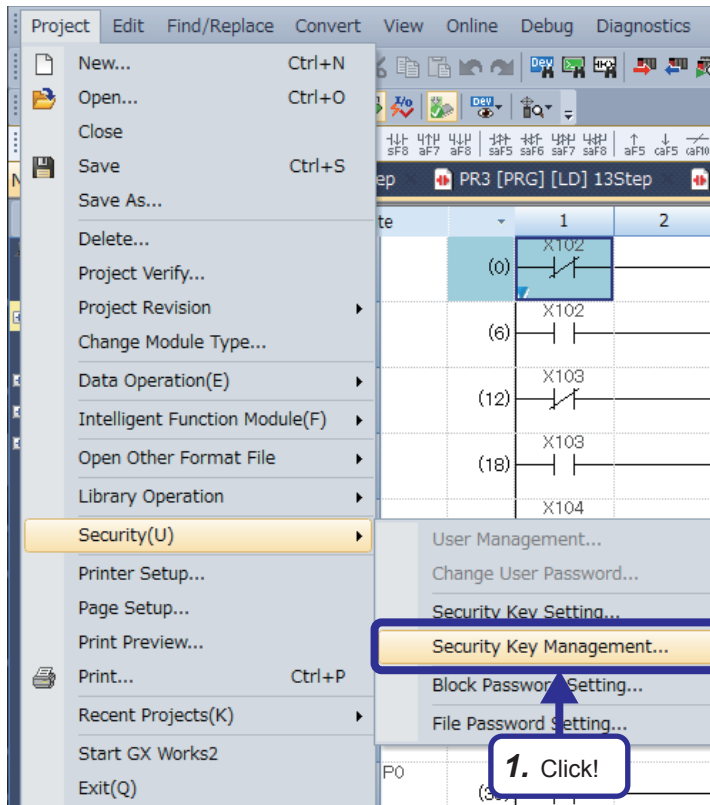
## 4.3.1 Creating a security key

Create a security key and register it to the personal computer.

Number of security key registration: The security keys can be created for each logon user. Up to 128 security keys including created security keys and imported ones can be registered for each logon user.

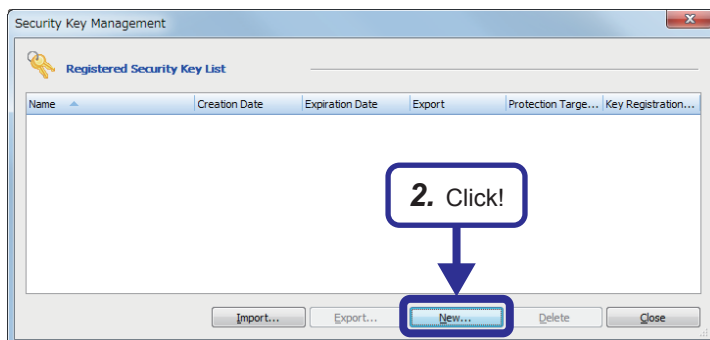
Set a security key to the "EX5.gx3" project created in Chapter 3.

### Operating procedure



1. Click [Project] → [Security] → [Security Key Management] from the menu.

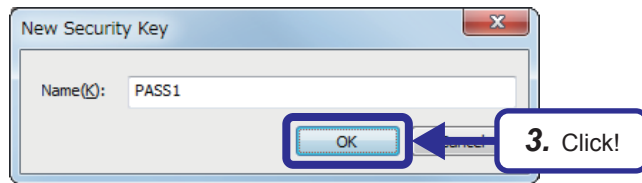
4



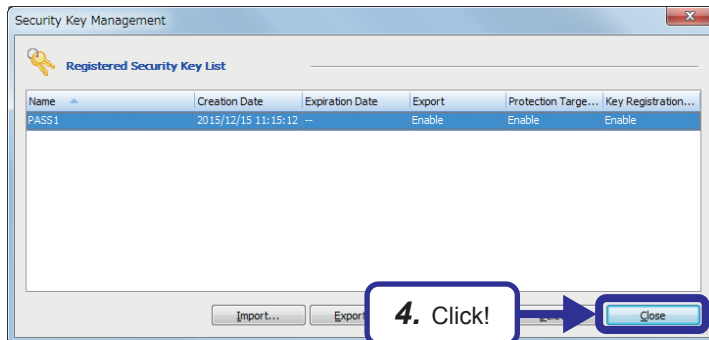
2. The "Security Key Management" dialog box appears. Click the [New] button.

(To the next page)

(From the previous page)



- 3.** The "New Security Key" dialog box appears. Enter "PASS1" in "Name" and click the [OK] button.



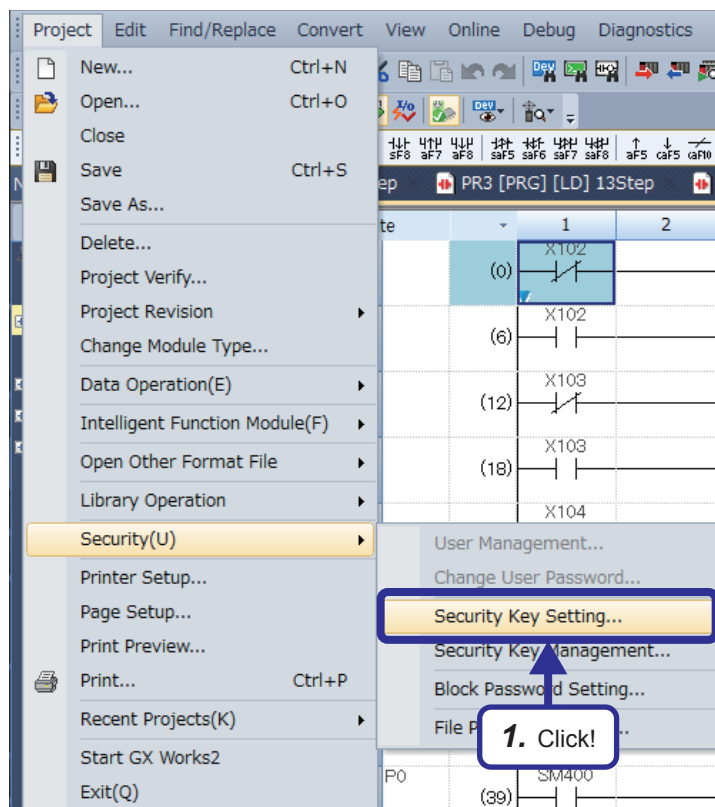
- 4.** Check that the security key "PASS1" has been created, and click the [Close] button.

## 4.3.2 Registering a security key in a program file

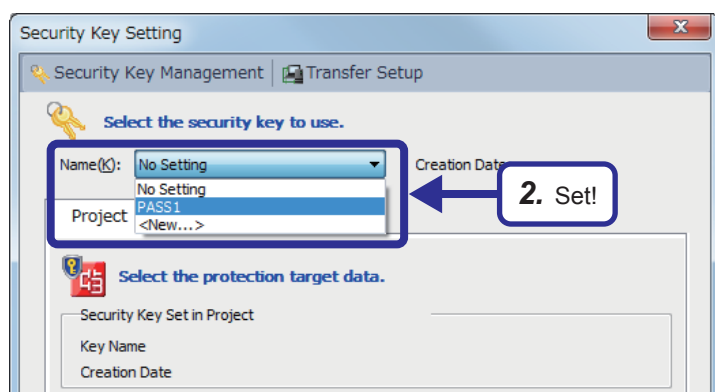
Create a security key on the "Security Key Management" window in advance.

Although security keys can be registered in units of program files, all program files in a project will use a single security key.

### Operating procedure



1. Click [Project] → [Security] → [Security Key Setting] from the menu.

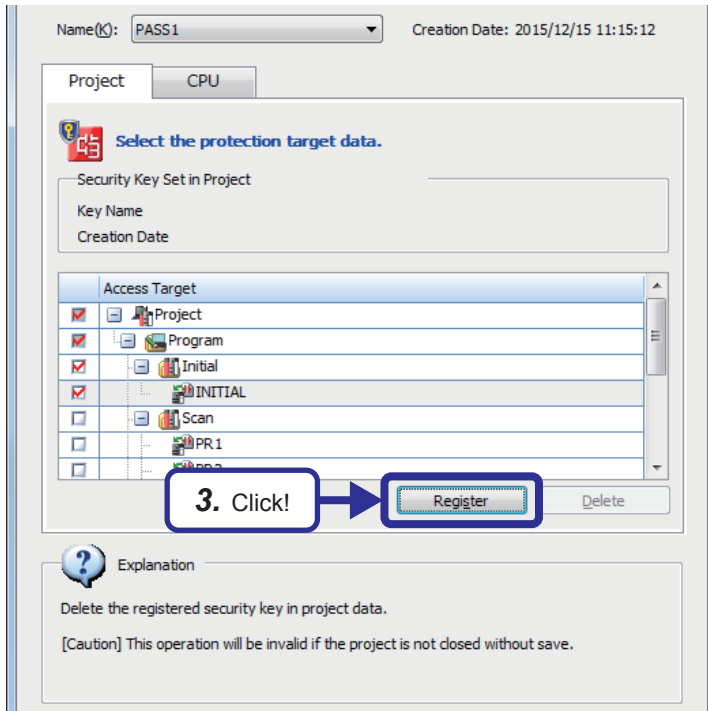


2. The "Security Key Setting" dialog box appears. Set "PASS1" as the security key.

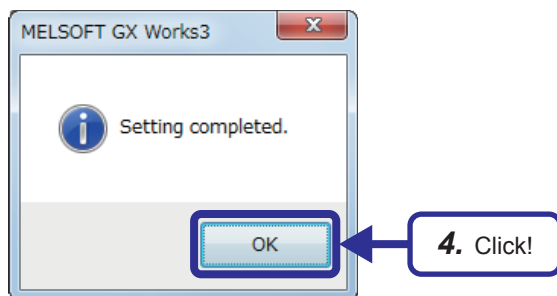


(To the next page)

(From the previous page)



3. Select protection target data from the items in "Access Target", and click the [Register] button.

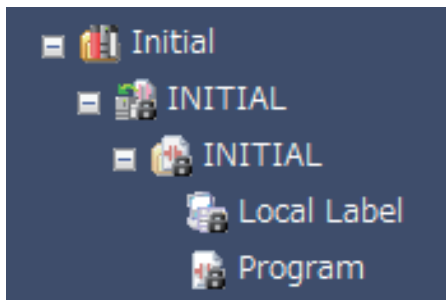


4. The security key setting is completed. Click the [OK] button.
5. Save and close the project.  
The security key registered in the program file will be valid after the project is saved.

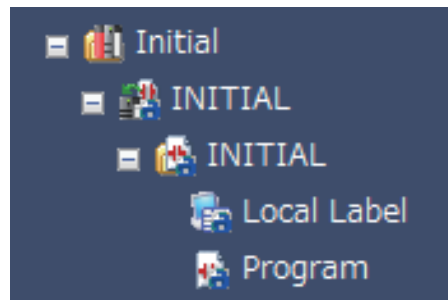
#### Point

The program file where a security key has been registered is displayed in the "Project" view as follows.

When the security key of the personal computer and the one of the project do not match



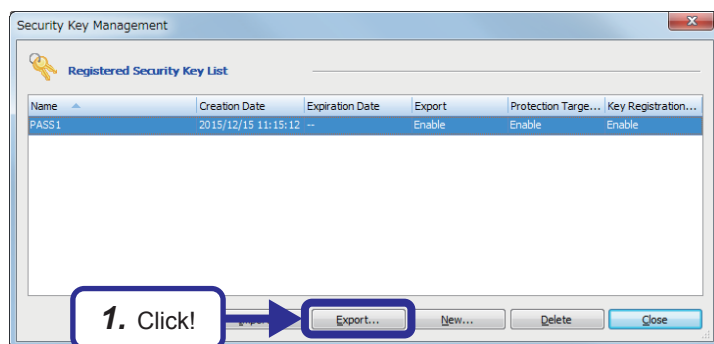
When the security key of the personal computer and the one of the project match



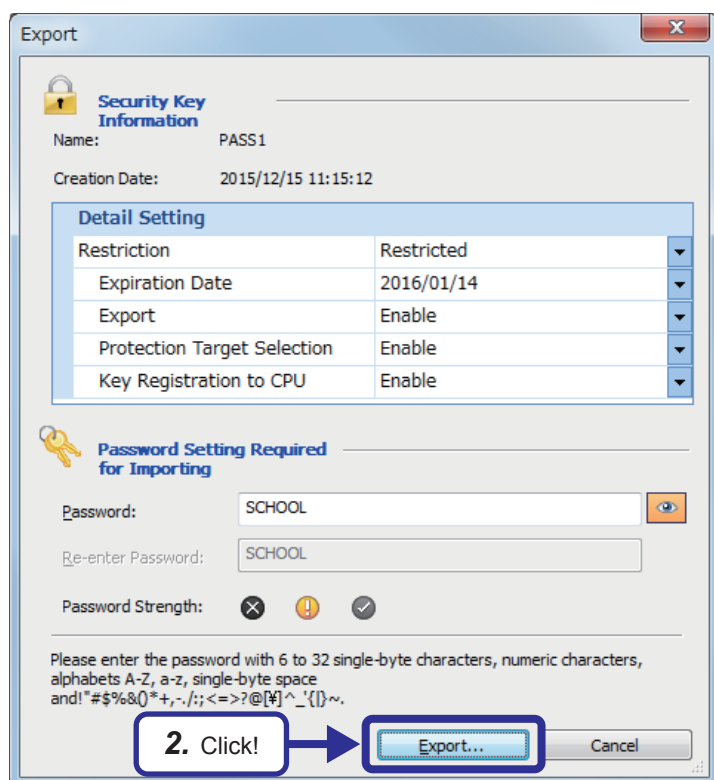
## 4.3.3 Exporting a security key

Export the security key registered to the personal computer to a file in the importable format (\*.ity).

### Operating procedure



1. Select the security key to be exported on the "Security Key Management" dialog box, and click the [Export] button.



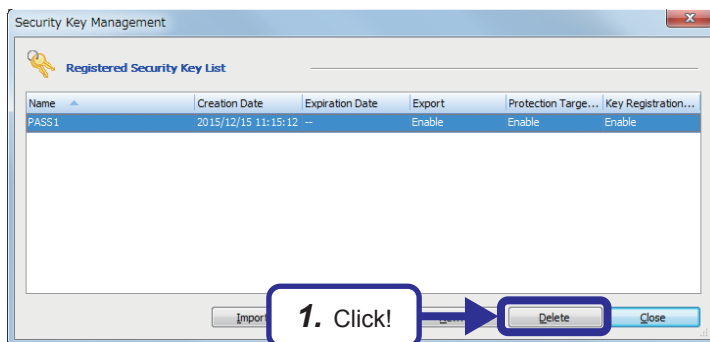
2. The "Export" dialog box appears. Enter "SCHOOL" in "Password", and click the [Export] button.

The security key is exported to a specified export destination.

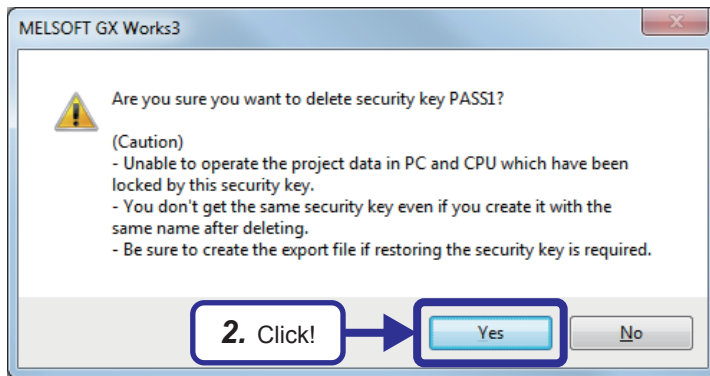
## 4.3.4 Deleting a security key

Once the security key used to lock a project is deleted, the locked data in the project cannot be browsed/edited. After the security key has been deleted, even though a new security key with the same name as the deleted one is created, the project cannot be accessed/edited. Before deleting a security key, create a new project.

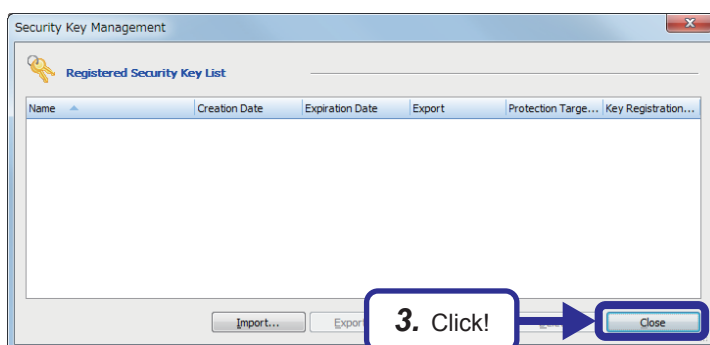
### Operating procedure



1. Select the security key to be deleted on the "Security Key Management" dialog box, and click the [Delete] button.



2. The dialog box shown on the left appears. Click the [Yes] button.



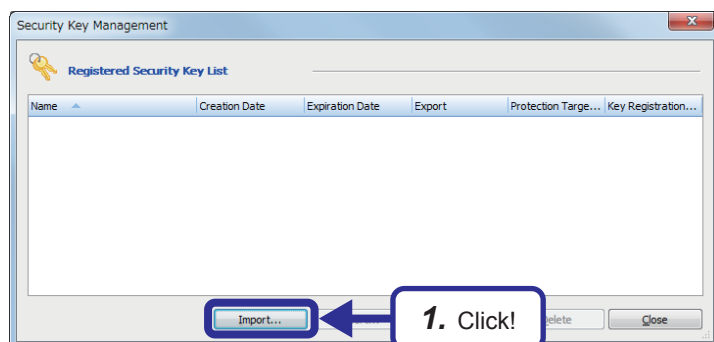
3. Check that the security key "PASS1" has been deleted, and click the [Close] button.

The program file to which the deleted security key has been set can no longer be edited.

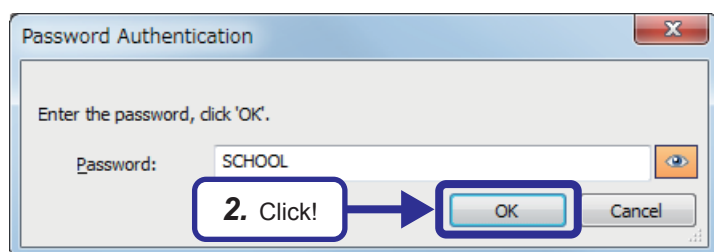
## 4.3.5 Importing a security key

Import and register an exported security key file to a personal computer.

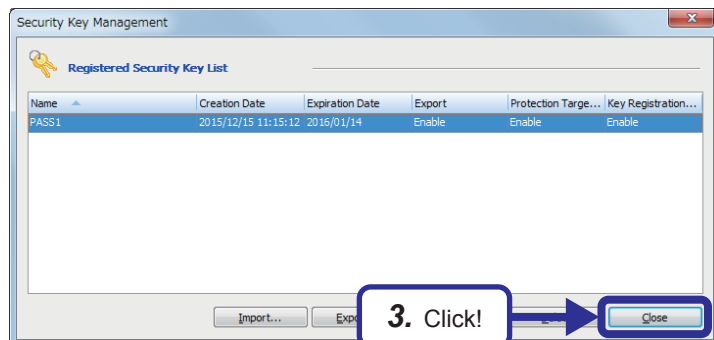
### Operating procedure



1. Click the [Import] button on the "Security Key Management" dialog box, and select the security key to be imported.



2. The "Password Authentication" dialog box appears. Enter "SCHOOL", the password set at the export, and click the [OK] button.



3. Check that the security key "PASS1" has been imported, and click the [Close] button.

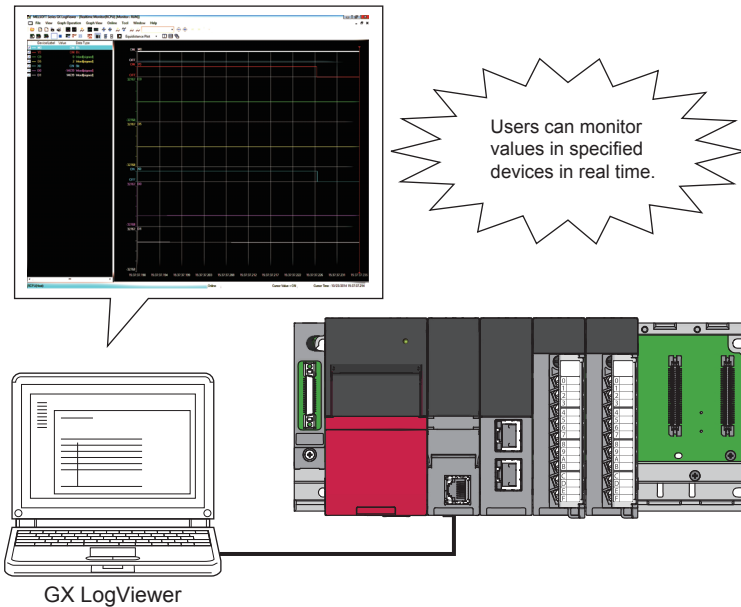
By importing the security key, the program file to which the security key has been set can be edited.



## 4.4 Sampling Trace

### 4.4.1 Realtime monitor function

This function monitors the contents in specified devices of the CPU module in real time at a specified interval or desired timing. The function can be set with GX LogViewer, which graphically shows value changes of specified devices. Saving set data and displayed graphs makes it possible to simplify settings and check the graphs later.



#### Restriction

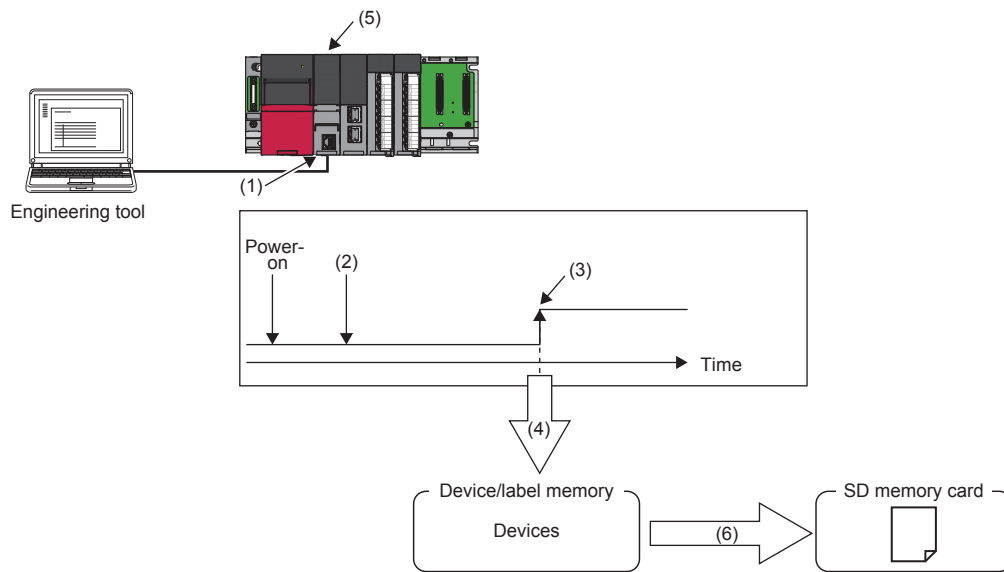
- Before executing the realtime monitor function, check the versions of the CPU module and GX LogViewer used.

For details on the realtime monitor function, refer to the following.

 GX LogViewer Version1 Operating Manual

## 4.4.2 Memory dump function

This function saves data in the devices of the CPU module at desired timing. Checking data at desired timing through the function facilitates analysis of troubles, which occur under particular conditions.



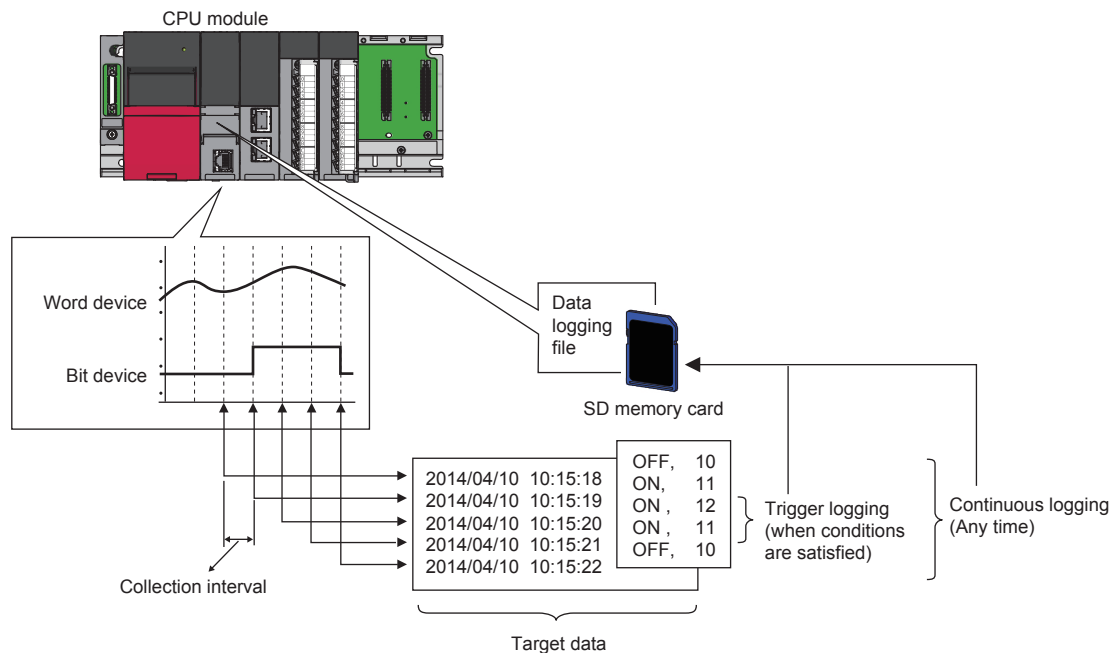
- (1) Perform memory dump settings.
- (2) Writing a memory dump setting file sets the waiting status for a trigger.
- (3) A trigger condition is satisfied.
- (4) Data collection starts.
- (5) The LED on the front of the CPU module shows the memory dump status.
- (6) A memory dump file is stored in the SD memory card.

### Restriction

- Before executing the memory dump function, check the versions of the CPU module and engineering tool used.

# 4.4.3 Data logging function

This function collects data at a specified interval or desired timing and saves the data as a file on the SD memory card. The data logging function allows the entry of up to 10 data logging settings.



# 5 PROGRAMMING FOR INTELLIGENT FUNCTION MODULES

## Point

- This chapter describes types of intelligent function modules.
- This chapter describes buffer memory areas.
- This chapter describes communications between the CPU module and intelligent function modules.
- This chapter describes how to use each monitor window.

## 5.1 Intelligent Function Module

5

### Types of intelligent function modules

Intelligent function modules can execute functions that CPU modules (hereinafter referred to as RCPU) cannot execute, or they realize functions that are limited to some applications.

Thus, select an intelligent function module appropriate to purposes.

RCPU-compatible intelligent function modules can be used with the RCPU.

The following table lists some intelligent function modules as an example.

Name	Number of occupied I/O points	Function	Internal current consumption (5VDC)
A/D converter module (R60AD4)	16 points (I/O assignment: Intelligent 16 points)	This input module converts the following values. 0 to 20mADC → 0 to 32000 0 to $\pm 10V$ → 0 to $\pm 32000$	0.22A
D/A converter module (R60DA4)	16 points (I/O assignment: Intelligent 16 points)	This output module converts the following values. 0 to 32000 → 0 to 20mADC 0 to $\pm 32000$ → 0 to $\pm 10V$	0.16A

### Combination use of intelligent function modules and the CPU module

Mount intelligent function modules to I/O slots of a main base unit or extension base units.

## 5.2 Data Communication Between Intelligent Function Modules and the CPU Module

Intelligent function modules and the CPU module transfer data of the following two types in data communication.

Bit data: Input (X) and output (Y) signals

Word data: 16-bit or 32-bit data

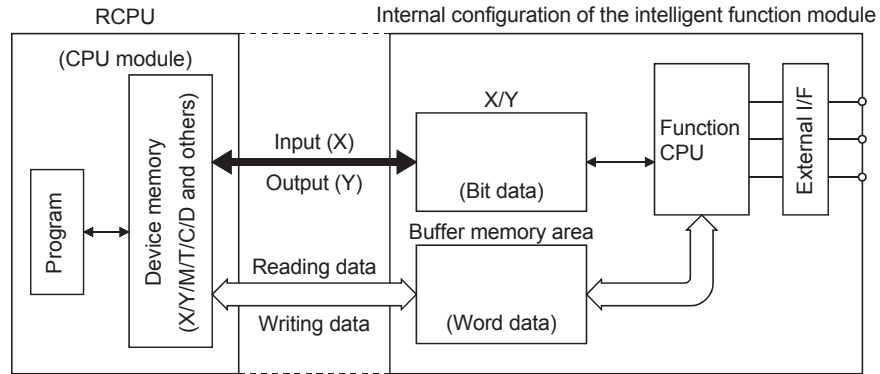


Figure 5.1 Internal configuration of the intelligent function module

## 5.2.1 I/O signals to the CPU module

The symbols X (input) and Y (output) are used for 1-bit signals transferred between the RCPU and intelligent function modules.

These X and Y signals are different from external inputs and outputs. These X and Y signals are dedicated for intelligent function modules and used in sequence programs. I/O numbers are assigned according to the slot where the intelligent function module has been mounted.

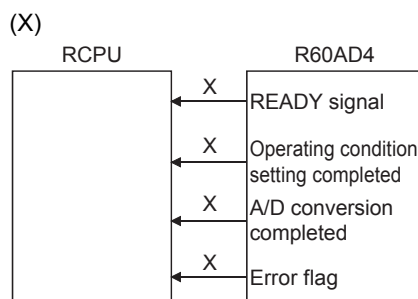


Figure 5.2 Input (X) signals from the R60AD4

X signals used in sequence programs are input to the RCPU from an intelligent function module. These signals are generated in the intelligent function module. X signals are used as contacts in a program. The following shows some X signals as examples.

(1) READY signal

This signal notifies the RCPU that the intelligent function module has normally been started at power-on and is ready.

(2) Operating condition setting completed

This signal is used as an interlock condition for turning on or off Operating condition setting request (Y9) when any of the following settings is changed.

- A/D conversion enable/disable setting
- CH□ Time Average/Count Average setting
- Averaging process setting

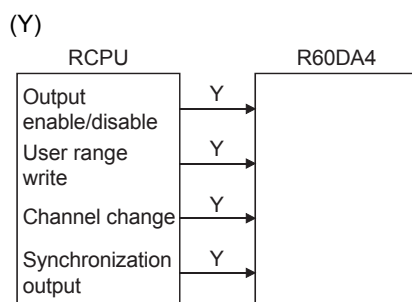


Figure 5.3 Output (Y) signals from the CPU module

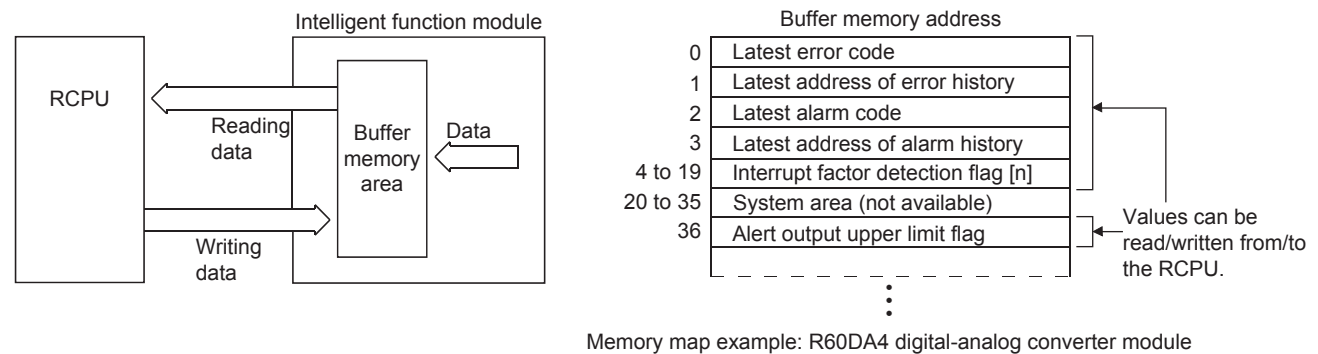
The SET, RST, and OUT-Y instructions used in sequence programs are output to the intelligent function module from the RCPU. These signals are generated in the RCPU. Y signals are used as coils or contacts in a program.

**Ex.**

A D/A converter module outputs an instruction to allow external output of an analog value converted from a digital value.

# 5.2.2 Data communication with intelligent function modules

Data is transferred in units of 16 bits or 32 bits. An intelligent function module has buffer memory areas for storing the data.



Memory map example: R60DA4 digital-analog converter module

Figure 5.4 Buffer memory area

- (1) The RCPU reads/writes data from/to buffer memory areas. Some modules can write data to the buffer memory areas through an interface of peripheral.
- (2) In the buffer memory, addresses unique to each intelligent function module are assigned in units of one word (16 bits). The start address is 0, and these addresses are used to specify areas from/to which data is read or written. The minimum unit is one word. 17-bit to 32-bit data is handled as two words (32-bit data).

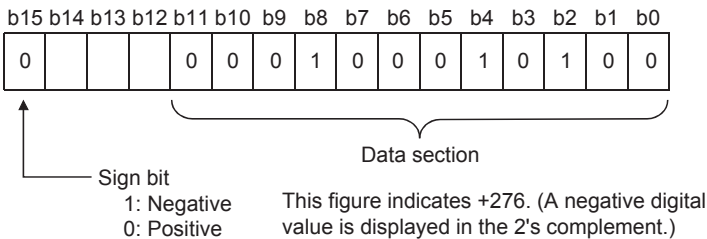


Figure 5.5 Example of a buffer memory area (D/A converter module)

Figure 5.5 shows 16 bit data in a D/A converter module as an example. A digital value has been written from the RCPU. A digital value of -32000 to 32000 is set in 16-bit signed binary.

- (3) The buffer memory is RAM.

## 5.3 Methods of the Communication with Intelligent Function Modules

### 5.3.1 Communications with intelligent function modules

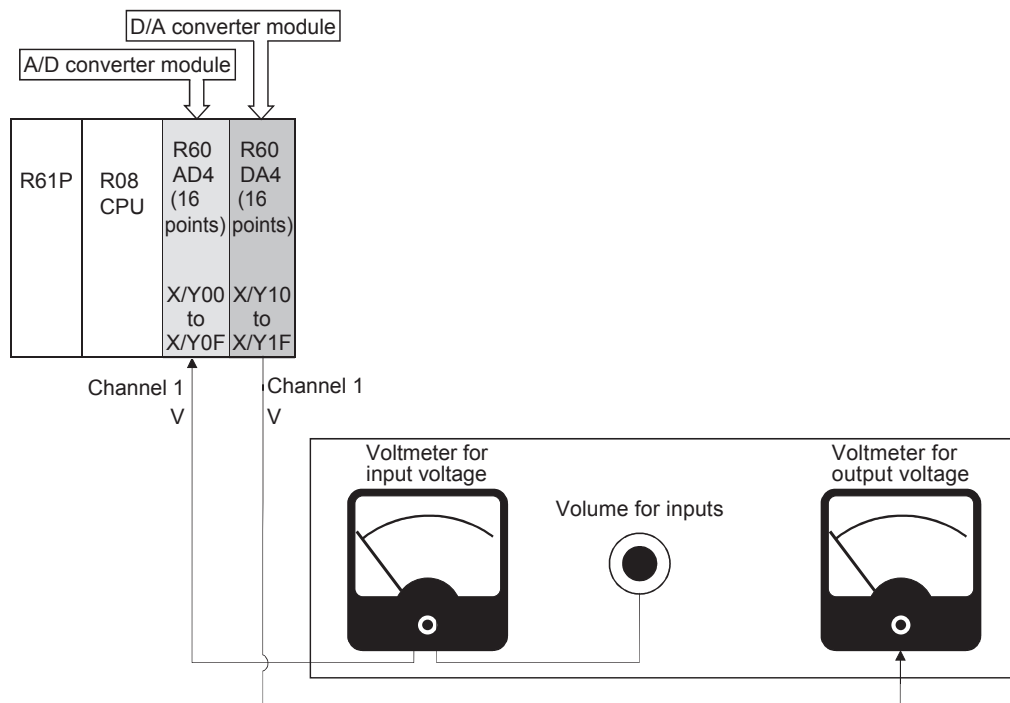
The following table lists the methods of communications between the RCPU and intelligent function modules.

Communication method	Function	Setting method
Initial setting, auto refresh setting	<p>Configure the initial settings and auto refresh settings of intelligent function modules. Users can read/write data between the RCPU and intelligent function modules without the need to create programs for communications or consider buffer memory addresses.</p> <p>Example: When the A/D converter module R60AD4 is used</p> <ul style="list-style-type: none"><li>Initial setting:<ul style="list-style-type: none"><li>A/D conversion enable/disable setting</li><li>Sampling/averaging process specification</li><li>Time average/count average specification</li><li>Average time/average count specification</li></ul>(The initial setting is stored in the intelligent parameters of the RCPU.)</li><li>Auto refresh setting: Set devices of the RCPU for storing the following data.<ul style="list-style-type: none"><li>Digital output of the R60AD4</li><li>Maximum value/minimum value of the R60AD4</li><li>Error codes</li></ul>(Auto refresh setting data is stored in the intelligent parameters of the RCPU.)</li></ul>	Use GX Works3.
Device initial value	<p>The initial setting is written to intelligent function modules at the following timing.</p> <ul style="list-style-type: none"><li>When the RCPU is powered on</li><li>When the CPU module is reset</li><li>When the operating status of the CPU module is changed from STOP to RUN</li></ul>	Use GX Works3 and specify the range of the intelligent function module device (U□\G□).
FROM/TO instructions	With the FROM/TO instructions, read/write data from/to the buffer memory of an intelligent function module.	Use the instructions in sequence programs.
Intelligent function module device (U□\G□)	<p>Buffer memory areas of an intelligent function module can be directly expressed as device areas of the RCPU.</p> <p>Unlike the FROM/TO instructions, data read from an intelligent function module can be processed with one instruction.</p>	Specify the buffer memory areas as device areas in sequence programs.
Intelligent function module dedicated instruction	This instruction is used to simplify programs for using the functions of intelligent function modules.	Use the instructions in sequence programs.



## 5.4 Configuration of the Intelligent Function Modules for Demonstrations

The A/D converter module and D/A converter module convert analog signals and digital data input with the volume for inputs of the demonstration machine or ladder programs.



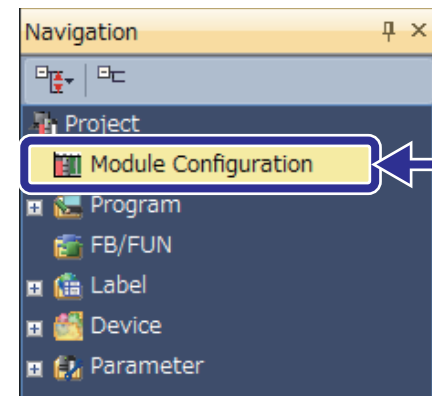
## 5.5 Creating the Module Configuration

Arrange program elements (objects) in the "Module Configuration" window so that the same configuration as that of the actual system is created.

The configuration that can be created in the "Module Configuration" window of GX Works3 is the one to be managed with the CPU module of the project.

### 5.5.1 Arranging objects

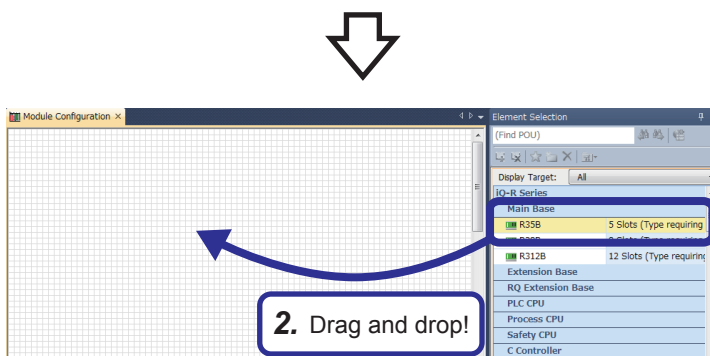
#### Operating procedure



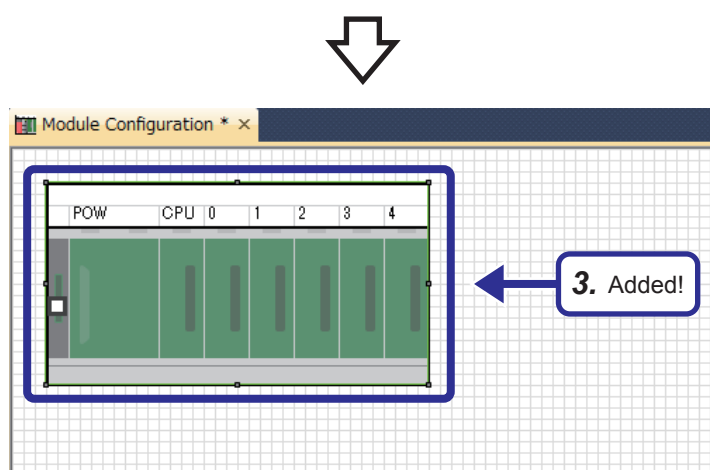
1. Double-click "Module Configuration" in the "Project" view.

When the dialog box about parameter information appears, click the [OK] button.

5



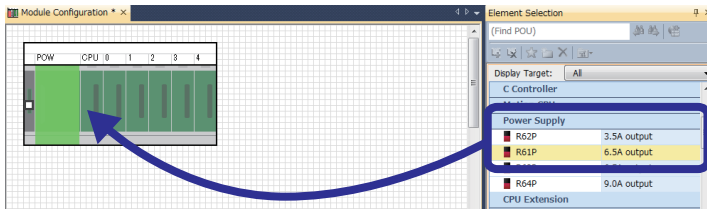
2. The "Module Configuration" window appears. Select "R35B" from "Main Base" in the "Element Selection" window, and drag and drop it to the "Module Configuration" window.



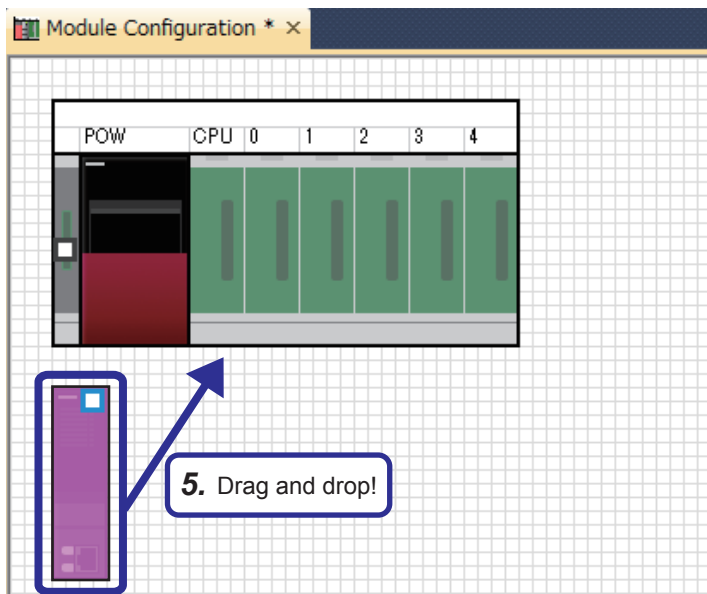
3. "R35B" is added on the "Module Configuration" window.

(To the next page)

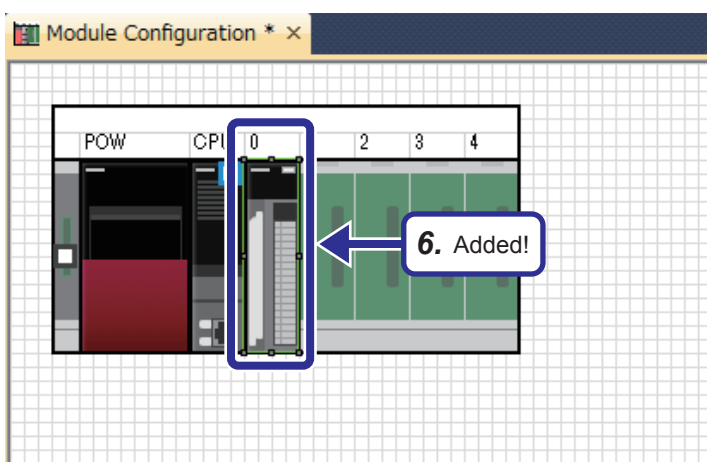
(From the previous page)



4. Drag and drop!



5. Drag and drop!



6. Added!



(To the next page)

4. Select "R61P" from "Power Supply" in the "Element Selection" window, and drag and drop it to the power supply slot of the R35B on the "Module Configuration" window.

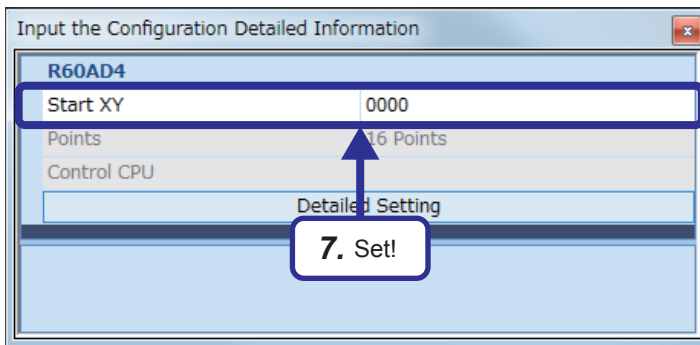
While the power supply module is being dragged and dropped, the area where the power supply module can be arranged is highlighted.

5. Add the R08CPU that has already been arranged in the module configuration to the CPU slot of the R35B.

When the R08CPU has not been arranged in the module configuration, add the R08CPU from the "Element Selection" window in the same way as for the power supply module.

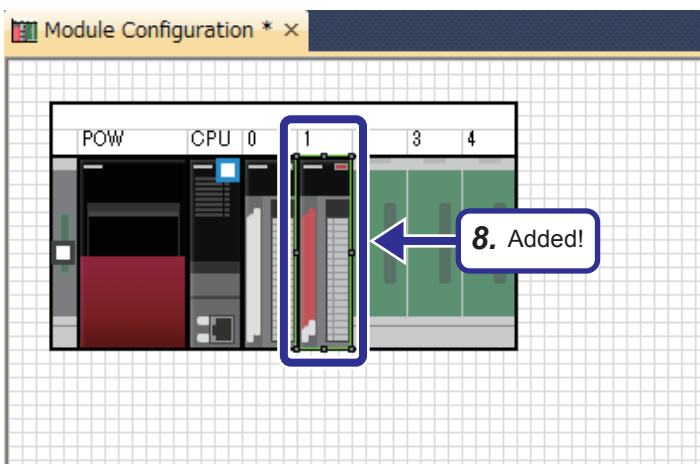
6. Select "R60AD4" from "Analog Input" in the "Element Selection" window, and drag and drop it to the slot No. 0 of the R35B in the same way as the power supply module was added.

(From the previous page)



7. Right-click the R60AD4, and click [Parameter] → [Input Detailed Configuration Information Window] from the menu to display the "Input the Configuration Detailed Information" window. Set the following item.

[Parameters to be set]  
Start XY: 0000

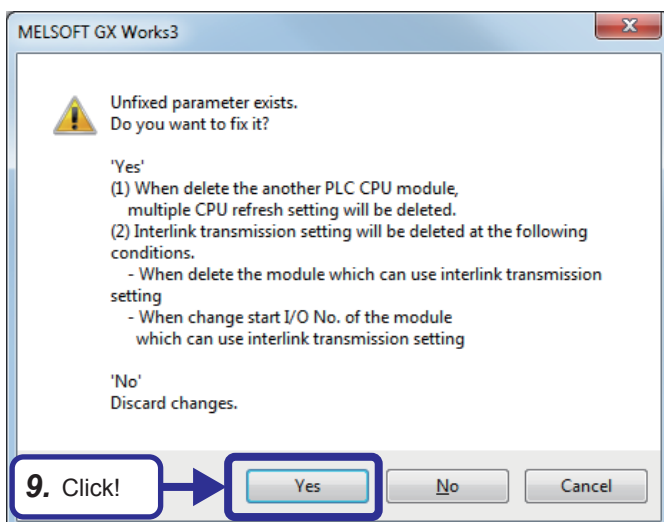


8. Select "R60DA4" from "Analog Output" in the "Element Selection" window, and drag and drop it to the slot No. 1 of the R35B. Set the following item on the "Input the Configuration Detailed Information" window.

[Parameters to be set]  
Start XY: 0010

Save the created and set data.

Project name: EX7

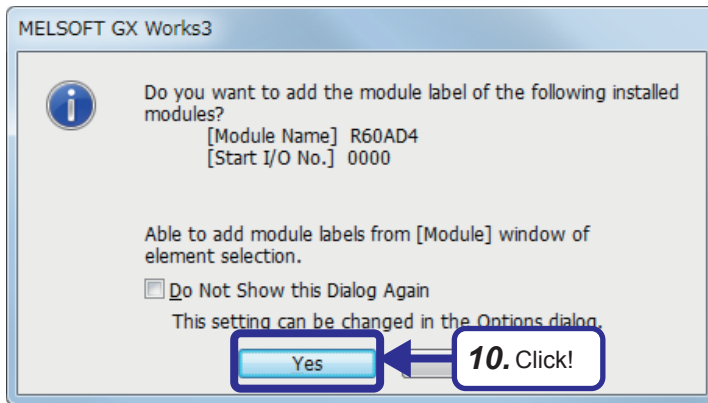


9. The dialog box shown on the left appears. Click the [Yes] button.

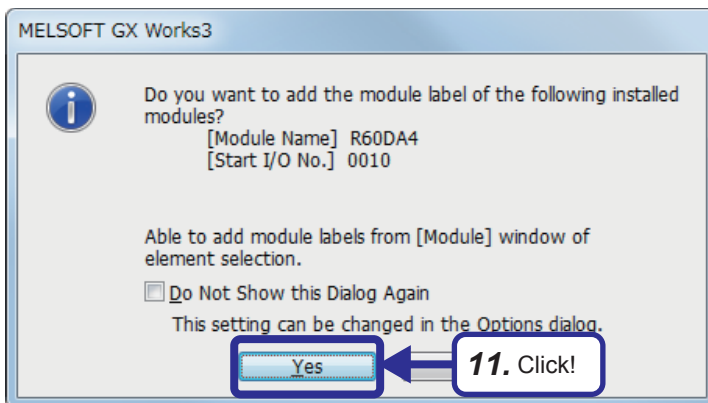


(To the next page)

(From the previous page)



**10.** The dialog box shown on the left appears. Click the [Yes] button.



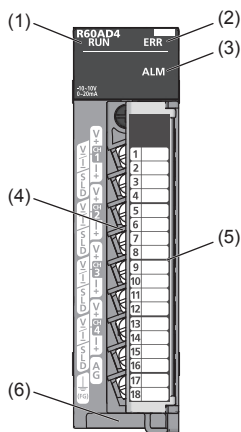
**11.** The dialog box shown on the left appears. Click the [Yes] button.

## 5.6 R60AD4 Analog-digital Converter Module

### 5.6.1 Part names

This section describes part names of the R60AD4.

For details, refer to the user's manual of the module.



No.	Name	Description
(1)	RUN LED	Indicates the operating status of the module. On: Normal operation Flashing: In offset/gain setting mode Off: 5V power supply interrupted or watchdog timer error occurred
(2)	ERR LED	Indicates the error status of the module.* <sup>1</sup> On: Error occurred Off: Normal operation
(3)	ALM LED	Indicates the alarm status of the module.* <sup>1</sup> On: Alert (process alarm or rate alarm) issued Flashing: Input signal error detected Off: Normal operation
(4)	Terminal block	18-point screw terminal block for connecting input signal wires of external devices and others
(5)	Terminal block cover	Covers for preventing electric shock while the power is on
(6)	Production information marking	Shows the product information (16 digits) of the module.

\*1 For details, refer to the following.

📖 MELSEC iQ-R Analog-Digital Converter Module User's Manual (Application)

## 5.6.2 A/D conversion characteristics

### A/D conversion characteristics at the voltage input

(When the input range is set to -10 to 10V)

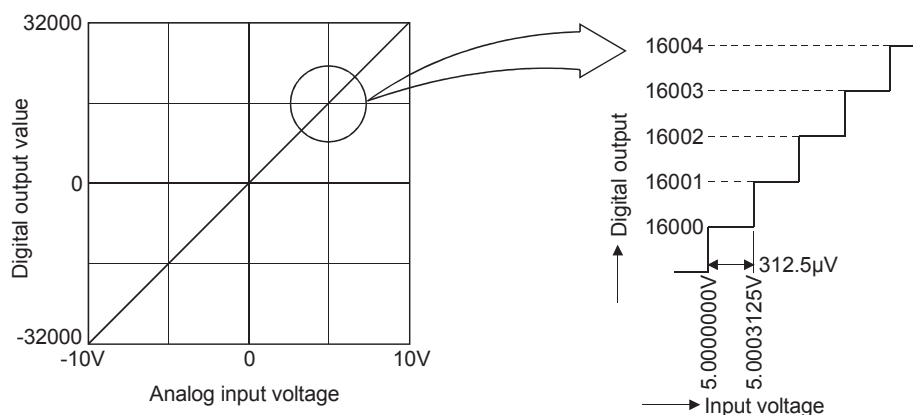


Figure 5.6 A/D conversion characteristics (voltage input)

An A/D converter module converts external analog inputs to digital values, and allows the CPU module to perform operations. At the voltage input, the A/D converter module outputs a digital value of -32000 at -10V and a digital value of 32000 at 10V. Thus, an input voltage of 312.5 $\mu$ V corresponds to a digital value of 1. An input voltage lower than 312.5 $\mu$ V cannot be converted and is omitted.

### A/D conversion characteristics at the current input

(When the input range is set to 0 to 20mA)

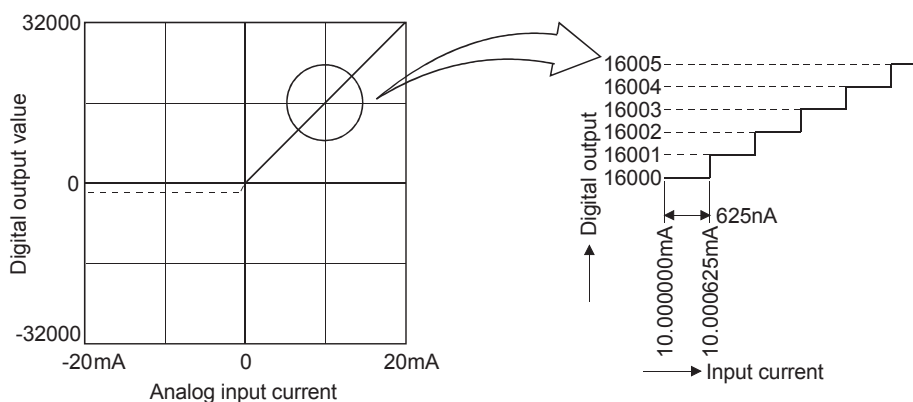


Figure 5.7 A/D conversion characteristics (current input)

The A/D converter module outputs a digital value of 0 at 0mA and a digital value of 32000 at 20mA. An input current of 625nA corresponds to a digital value of 1. An input current lower than 625nA cannot be converted and is omitted.



A voltage or current value (maximum resolution) corresponding to a digital value of 1 in A/D conversion differs depending on the input range setting.

## 5.6.3 List of I/O signals and buffer memory assignment

### List of I/O signals

The following tables list the I/O signals of the A/D converter module.

The I/O numbers (X/Y) described in this section are for the case when the start I/O number of the A/D converter module is set to 0.

#### ■Input signal

Device No.	Signal name
X0	Module READY
X1 to X7	Use prohibited
X8	Alert output signal
X9	Operating condition setting completed flag
XA	Offset/gain setting mode status flag
XB	Channel change completed flag
XC	Input signal error detection signal
XD	Maximum value/minimum value reset completed flag
XE	A/D conversion completed flag
XF	Error flag

#### ■Output signal

Device No.	Signal name
Y0 to Y8	Use prohibited
Y9	Operating condition setting request
YA	User range write request
YB	Channel change request
YC	Use prohibited
YD	Maximum value/minimum value reset request
YE	Use prohibited
YF	Error clear request



## Buffer memory assignment

The module has two operation modes. In R mode, the module operates with the buffer memory map that has been newly provided for the MELSEC iQ-R series. In Q compatible mode, the module operates with the buffer memory map converted to the equivalent one for the MELSEC-Q series.

The following list of buffer memory addresses is for the R mode.

For details, refer to the following.

 MELSEC iQ-R Analog-Digital Converter Module User's Manual (Application)

### Point

Do not write data to the system areas and areas whose data types are monitor in the buffer memory. Writing data to these areas can cause the malfunction of the module.

## ■Un\G0 to Un\G399

Address (decimal)	Address (hexadecimal)	Name	Default	Data type	Auto refresh	Enabled by Y9 <sup>*1</sup>
0	0H	Latest error code	0	Monitor	○	—
1	1H	Latest address of error history	0	Monitor	○	—
2	2H	Latest alarm code	0	Monitor	○	—
3	3H	Latest address of alarm history	0	Monitor	○	—
4 to 19	4H to 13H	Interrupt factor detection flag [n] <sup>*2</sup>	0	Monitor	○	—
20 to 35	14H to 23H	System area	—	—	—	—
36	24H	Alert output flag (Process alarm upper limit)	0000H	Monitor	○	—
37	25H	Alert output flag (Process alarm lower limit)	0000H	Monitor	○	—
38	26H	Alert output flag (Rate alarm upper limit)	0000H	Monitor	○	—
39	27H	Alert output flag (Rate alarm lower limit)	0000H	Monitor	○	—
40	28H	Input signal error detection flag	0000H	Monitor	○	—
41	29H	System area	0000H	—	—	—
42	2AH	A/D conversion completed flag	0000H	Monitor	○	—
43 to 89	2BH to 59H	System area	—	—	—	—
90	5AH	Level data 0	0	Control	○	—
91	5BH	Level data 1	0	Control	○	—
92	5CH	Level data 2	0	Control	○	—
93	5DH	Level data 3	0	Control	○	—
94	5EH	Level data 4	0	Control	○	—
95	5FH	Level data 5	0	Control	○	—
96	60H	Level data 6	0	Control	○	—
97	61H	Level data 7	0	Control	○	—
98	62H	Level data 8	0	Control	○	—
99	63H	Level data 9	0	Control	○	—
100 to 123	64H to 7BH	System area	—	—	—	—
124 to 139	7CH to 8BH	Interrupt factor mask [n] <sup>*2</sup>	0	Control	×	—
140 to 155	8CH to 9BH	System area	—	—	—	—
156 to 171	9CH to ABH	Interrupt factor reset request [n] <sup>*2</sup>	0	Control	×	—
172 to 199	ACH to C7H	System area	—	—	—	—
200 to 215	C8H to D7H	Interrupt factor generation setting [n] <sup>*2</sup>	0	Setting	×	○
216 to 231	D8H to E7H	System area	—	—	—	—
232 to 247	E8H to F7H	Condition target setting [n] <sup>*2</sup>	0	Setting	×	○
248 to 263	F8H to 107H	System area	—	—	—	—
264 to 279	108H to 117H	Condition target channel setting [n] <sup>*2</sup>	0	Setting	×	○
280 to 295	118H to 127H	System area	—	—	—	—
296, 297	128H, 129H	Mode switching setting	0	Setting	×	○
298 to 399	130H to 18FH	System area	—	—	—	—

\*1 Item enabled by turning on and off Operating condition setting request (Y9)

\*2 [n] in the table indicates an interrupt setting number (n = 1 to 16).

## ■Un\G400 to Un\G3599

Address Decimal (hexadecimal)								Name	Default	Data type	Auto refresh	Enabled by Y9*1
CH1	CH2	CH3	CH4	CH5	CH6	CH7	CH8					
400 (190H)	600 (258H)	800 (320H)	1000 (3E8H)	1200 (4B0H)	1400 (578H)	1600 (640H)	1800 (708H)	CH□ Digital output value	0	Monitor	○	—
401 (191H)	601 (259H)	801 (321H)	1001 (3E9H)	1201 (4B1H)	1401 (579H)	1601 (641H)	1801 (709H)	System area	—	—	—	—
402 (192H)	602 (25AH)	802 (322H)	1002 (3EAH)	1202 (4B2H)	1402 (57AH)	1602 (642H)	1802 (70AH)	CH□ Digital operation value	0	Monitor	○	—
403 (193H)	603 (25BH)	803 (323H)	1003 (3EBH)	1203 (4B3H)	1403 (57BH)	1603 (643H)	1803 (70BH)	System area	—	—	—	—
404 (194H)	604 (25CH)	804 (324H)	1004 (3ECH)	1204 (4B4H)	1404 (57CH)	1604 (644H)	1804 (70CH)	CH□ Maximum value	0	Monitor	○	—
405 (195H)	605 (25DH)	805 (325H)	1005 (3EDH)	1205 (4B5H)	1405 (57DH)	1605 (645H)	1805 (70DH)	System area	—	—	—	—
406 (196H)	606 (25EH)	806 (326H)	1006 (3EEH)	1206 (4B6H)	1406 (57EH)	1606 (646H)	1806 (70EH)	CH□ Minimum value	0	Monitor	○	—
407 (197H)	607 (25FH)	807 (327H)	1007 (3EFH)	1207 (4B7H)	1407 (57FH)	1607 (647H)	1807 (70FH)	System area	—	—	—	—
408 (198H)	608 (260H)	808 (328H)	1008 (3F0H)	1208 (4B8H)	1408 (580H)	1608 (648H)	1808 (710H)	CH□ Difference conversion status flag	0	Monitor	○	—
409 (199H)	609 (261H)	809 (329H)	1009 (3F1H)	1209 (4B9H)	1409 (581H)	1609 (649H)	1809 (711H)	CH□ Logging hold flag	0	Monitor	○	—
410 to 429 (19AH to 1ADH)	610 to 629 (262H to 275H)	810 to 829 (32AH to 33DH)	1010 to 1029 (3F2H to 405H)	1210 to 1229 (4BAH to 4CDH)	1410 to 1429 (582H to 595H)	1610 to 1629 (64AH to 65DH)	1810 to 1829 (712H to 725H)	System area	—	—	—	—
430 (1AEH)	630 (276H)	830 (33EH)	1030 (406H)	1230 (4CEH)	1430 (596H)	1630 (65EH)	1830 (726H)	CH□ Range setting monitor	0000H	Monitor	×	—
431 (1AFH)	631 (277H)	831 (33FH)	1031 (407H)	1231 (4CFH)	1431 (597H)	1631 (65FH)	1831 (727H)	System area	—	—	—	—
432 (1B0H)	632 (278H)	832 (340H)	1032 (408H)	1232 (4D0H)	1432 (598H)	1632 (660H)	1832 (728H)	CH□ Difference conversion reference value	0000H	Monitor	×	—
433 (1B1H)	633 (279H)	833 (341H)	1033 (409H)	1233 (4D1H)	1433 (599H)	1633 (661H)	1833 (729H)	System area	—	—	—	—
434 (1B2H)	634 (27AH)	834 (342H)	1034 (40AH)	1234 (4D2H)	1434 (59AH)	1634 (662H)	1834 (72AH)	CH□ Head pointer	0	Monitor	×	—
435 (1B3H)	635 (27BH)	835 (343H)	1035 (40BH)	1235 (4D3H)	1435 (59BH)	1635 (663H)	1835 (72BH)	CH□ Latest pointer	0	Monitor	×	—
436 (1B4H)	636 (27CH)	836 (344H)	1036 (40CH)	1236 (4D4H)	1436 (59CH)	1636 (664H)	1836 (72CH)	CH□ Number of logging data	0	Monitor	×	—
437 (1B5H)	637 (27DH)	837 (345H)	1037 (40DH)	1237 (4D5H)	1437 (59DH)	1637 (665H)	1837 (72DH)	CH□ Trigger pointer	0	Monitor	×	—
438 (1B6H)	638 (27EH)	838 (346H)	1038 (40EH)	1238 (4D6H)	1438 (59EH)	1638 (666H)	1838 (72EH)	CH□ Current logging read pointer	-1	Monitor	×	—
439 (1B7H)	639 (27FH)	839 (347H)	1039 (40FH)	1239 (4D7H)	1439 (59FH)	1639 (667H)	1839 (72FH)	CH□ Previous logging read pointer	-1	Monitor	×	—
440 (1B8H)	640 (280H)	840 (348H)	1040 (410H)	1240 (4D8H)	1440 (5A0H)	1640 (668H)	1840 (730H)	CH□ Logging read points monitor value	0	Monitor	×	—
441 (1B9H)	641 (281H)	841 (349H)	1041 (411H)	1241 (4D9H)	1441 (5A1H)	1641 (669H)	1841 (731H)	CH□ Logging cycle monitor value (s)	0	Monitor	×	—
442 (1BAH)	642 (282H)	842 (34AH)	1042 (412H)	1242 (4DAH)	1442 (5A2H)	1642 (66AH)	1842 (732H)	CH□ Logging cycle monitor value (ms)	0	Monitor	×	—
443 (1BBH)	643 (283H)	843 (34BH)	1043 (413H)	1243 (4DBH)	1443 (5A3H)	1643 (66BH)	1843 (733H)	CH□ Logging cycle monitor value (μs)	0	Monitor	×	—
444 (1BCH)	644 (284H)	844 (34CH)	1044 (414H)	1244 (4DCH)	1444 (5A4H)	1644 (66CH)	1844 (734H)	CH□ Trigger generation time (First/Last two digits of the year)	0	Monitor	×	—
445 (1BDH)	645 (285H)	845 (34DH)	1045 (415H)	1245 (4DDH)	1445 (5A5H)	1645 (66DH)	1845 (735H)	CH□ Trigger generation time (Month/Day)	0	Monitor	×	—

Address Decimal (hexadecimal)								Name	Default	Data type	Auto refresh	Enabled by Y9*1
CH1	CH2	CH3	CH4	CH5	CH6	CH7	CH8					
446 (1BEH)	646 (286H)	846 (34EH)	1046 (416H)	1246 (4DEH)	1446 (5A6H)	1646 (66EH)	1846 (736H)	CH□ Trigger generation time (Hour/Minute)	0	Monitor	×	—
447 (1BFH)	647 (287H)	847 (34FH)	1047 (417H)	1247 (4DFH)	1447 (5A7H)	1647 (66FH)	1847 (737H)	CH□ Trigger generation time (Second/Day of the week)	0	Monitor	×	—
448 (1C0H)	648 (288H)	848 (350H)	1048 (418H)	1248 (4E0H)	1448 (5A8H)	1648 (670H)	1848 (738H)	CH□ Trigger generation time (Millisecond)	0	Monitor	×	—
449 to 469 (1C1H to 1D5H)	649 to 669 (289H to 29DH)	849 to 869 (351H to 365H)	1049 to 1069 (419H to 42DH)	1249 to 1269 (4E1H to 4F5H)	1449 to 1469 (5A9H to 5BDH)	1649 to 1669 (671H to 685H)	1849 to 1869 (739H to 74DH)	System area	—	—	—	—
470 (1D6H)	670 (29EH)	870 (366H)	1070 (42EH)	1270 (4F6H)	1470 (5BEH)	1670 (686H)	1870 (74EH)	CH□ Difference conversion trigger	0	Control	○	—
471 (1D7H)	671 (29FH)	871 (367H)	1071 (42FH)	1271 (4F7H)	1471 (5BFH)	1671 (687H)	1871 (74FH)	CH□ Logging hold request	0	Control	○	—
472 (1D8H)	672 (2A0H)	872 (368H)	1072 (430H)	1272 (4F8H)	1472 (5C0H)	1672 (688H)	1872 (750H)	CH□ Conversion value shift amount	0	Control	○	—
473 to 499 (1D9H to 1F3H)	673 to 699 (2A1H to 2BBH)	873 to 899 (369H to 383H)	1073 to 1099 (431H to 44BH)	1273 to 1299 (4F9H to 513H)	1473 to 1499 (5C1H to 5DBH)	1673 to 1699 (689H to 6A3H)	1873 to 1899 (751H to 76BH)	System area	—	—	—	—
500 (1F4H)	700 (2BCH)	900 (384H)	1100 (44CH)	1300 (514H)	1500 (5DCH)	1700 (6A4H)	1900 (76CH)	CH□ A/D conversion enable/disable setting	0	Setting	×	○
501 (1F5H)	701 (2BDH)	901 (385H)	1101 (44DH)	1301 (515H)	1501 (5DDH)	1701 (6A5H)	1901 (76DH)	CH□ Averaging process specification	0	Setting	×	○
502 (1F6H)	702 (2BEH)	902 (386H)	1102 (44EH)	1302 (516H)	1502 (5DEH)	1702 (6A6H)	1902 (76EH)	CH□ Time average/Count average/Moving average/ Primary delay filter constant setting	0	Setting	×	○
503 (1F7H)	703 (2BFH)	903 (387H)	1103 (44FH)	1303 (517H)	1503 (5DFH)	1703 (6A7H)	1903 (76FH)	System area	—	—	—	—
504 (1F8H)	704 (2C0H)	904 (388H)	1104 (450H)	1304 (518H)	1504 (5E0H)	1704 (6A8H)	1904 (770H)	CH□ Scaling enable/ disable setting	1	Setting	×	○
505 (1F9H)	705 (2C1H)	905 (389H)	1105 (451H)	1305 (519H)	1505 (5E1H)	1705 (6A9H)	1905 (771H)	System area	—	—	—	—
506 (1FAH)	706 (2C2H)	906 (38AH)	1106 (452H)	1306 (51AH)	1506 (5E2H)	1706 (6AAH)	1906 (772H)	CH□ Scaling upper limit value	0	Setting	×	○
507 (1FBH)	707 (2C3H)	907 (38BH)	1107 (453H)	1307 (51BH)	1507 (5E3H)	1707 (6ABH)	1907 (773H)	System area	—	—	—	—
508 (1FCH)	708 (2C4H)	908 (38CH)	1108 (454H)	1308 (51CH)	1508 (5E4H)	1708 (6ACH)	1908 (774H)	CH□ Scaling lower limit value	0	Setting	×	○
509 (1FDH)	709 (2C5H)	909 (38DH)	1109 (455H)	1309 (51DH)	1509 (5E5H)	1709 (6ADH)	1909 (775H)	System area	—	—	—	—
510 (1FEH)	710 (2C6H)	910 (38EH)	1110 (456H)	1310 (51EH)	1510 (5E6H)	1710 (6AEH)	1910 (776H)	CH□ Digital clipping enable/disable setting	1	Setting	×	○
511 (1FFH)	711 (2C7H)	911 (38FH)	1111 (457H)	1311 (51FH)	1511 (5E7H)	1711 (6AFH)	1911 (777H)	System area	—	—	—	—
512 (200H)	712 (2C8H)	912 (390H)	1112 (458H)	1312 (520H)	1512 (5E8H)	1712 (6B0H)	1912 (778H)	CH□ Alert output setting (Process alarm)	1	Setting	×	○
513 (201H)	713 (2C9H)	913 (391H)	1113 (459H)	1313 (521H)	1513 (5E9H)	1713 (6B1H)	1913 (779H)	CH□ Alert output setting (Rate alarm)	1	Setting	×	○
514 (202H)	714 (2CAH)	914 (392H)	1114 (45AH)	1314 (522H)	1514 (5EAH)	1714 (6B2H)	1914 (77AH)	CH□ Process alarm upper upper limit value	0	Setting	×	○
515 (203H)	715 (2CBH)	915 (393H)	1115 (45BH)	1315 (523H)	1515 (5EBH)	1715 (6B3H)	1915 (77BH)	System area	—	—	—	—
516 (204H)	716 (2CCH)	916 (394H)	1116 (45CH)	1316 (524H)	1516 (5ECH)	1716 (6B4H)	1916 (77CH)	CH□ Process alarm upper lower limit value	0	Setting	×	○
517 (205H)	717 (2CDH)	917 (395H)	1117 (45DH)	1317 (525H)	1517 (5EDH)	1717 (6B5H)	1917 (77DH)	System area	—	—	—	—

Address Decimal (hexadecimal)								Name	Default	Data type	Auto refresh	Enabled by Y9*1
CH1	CH2	CH3	CH4	CH5	CH6	CH7	CH8					
518 (206H)	718 (2CEH)	918 (396H)	1118 (45EH)	1318 (526H)	1518 (5EEH)	1718 (6B6H)	1918 (77EH)	CH□ Process alarm lower upper limit value	0	Setting	×	○
519 (207H)	719 (2CFH)	919 (397H)	1119 (45FH)	1319 (527H)	1519 (5EFH)	1719 (6B7H)	1919 (77FH)	System area	—	—	—	—
520 (208H)	720 (2D0H)	920 (398H)	1120 (460H)	1320 (528H)	1520 (5F0H)	1720 (6B8H)	1920 (780H)	CH□ Process alarm lower lower limit value	0	Setting	×	○
521 (209H)	721 (2D1H)	921 (399H)	1121 (461H)	1321 (529H)	1521 (5F1H)	1721 (6B9H)	1921 (781H)	System area	—	—	—	—
522 (20AH)	722 (2D2H)	922 (39AH)	1122 (462H)	1322 (52AH)	1522 (5F2H)	1722 (6BAH)	1922 (782H)	CH□ Rate alarm alert detection cycle setting	0	Setting	×	○
523 (20BH)	723 (2D3H)	923 (39BH)	1123 (463H)	1323 (52BH)	1523 (5F3H)	1723 (6BBH)	1923 (783H)	System area	—	—	—	—
524 (20CH)	724 (2D4H)	924 (39CH)	1124 (464H)	1324 (52CH)	1524 (5F4H)	1724 (6BCH)	1924 (784H)	CH□ Rate alarm upper limit value	0	Setting	×	○
525 (20DH)	725 (2D5H)	925 (39DH)	1125 (465H)	1325 (52DH)	1525 (5F5H)	1725 (6BDH)	1925 (785H)	System area	—	—	—	—
526 (20EH)	726 (2D6H)	926 (39EH)	1126 (466H)	1326 (52EH)	1526 (5F6H)	1726 (6BEH)	1926 (786H)	CH□ Rate alarm lower limit value	0	Setting	×	○
527 (20FH)	727 (2D7H)	927 (39FH)	1127 (467H)	1327 (52FH)	1527 (5F7H)	1727 (6BFH)	1927 (787H)	System area	—	—	—	—
528 (210H)	728 (2D8H)	928 (3A0H)	1128 (468H)	1328 (530H)	1528 (5F8H)	1728 (6C0H)	1928 (788H)	CH□ Input signal error detection setting	0	Setting	×	○
529 (211H)	729 (2D9H)	929 (3A1H)	1129 (469H)	1329 (531H)	1529 (5F9H)	1729 (6C1H)	1929 (789H)	CH□ Input signal error detection setting value	50	Setting	×	○
530 to 534 (212H to 216H)	730 to 734 (2DAH to 2DEH)	930 to 934 (3A2H to 3A6H)	1130 to 1134 (46AH to 46EH)	1330 to 1334 (532H to 536H)	1530 to 1534 (5FAH to 5FEH)	1730 to 1734 (6C2H to 6C6H)	1930 to 1934 (78AH to 78EH)	System area	—	—	—	—
535 (217H)	735 (2DFH)	935 (3A7H)	1135 (46FH)	1335 (537H)	1535 (5FFH)	1735 (6C7H)	1935 (78FH)	CH□ Logging enable/ disable setting	1	Setting	×	○
536 (218H)	736 (2E0H)	936 (3A8H)	1136 (470H)	1336 (538H)	1536 (600H)	1736 (6C8H)	1936 (790H)	CH□ Logging data setting	1	Setting	×	○
537 (219H)	737 (2E1H)	937 (3A9H)	1137 (471H)	1337 (539H)	1537 (601H)	1737 (6C9H)	1937 (791H)	CH□ Logging cycle setting value	4	Setting	×	○
538 (21AH)	738 (2E2H)	938 (3AAH)	1138 (472H)	1338 (53AH)	1538 (602H)	1738 (6CAH)	1938 (792H)	CH□ Logging cycle unit setting	1	Setting	×	○
539 (21BH)	739 (2E3H)	939 (3ABH)	1139 (473H)	1339 (53BH)	1539 (603H)	1739 (6CBH)	1939 (793H)	CH□ Post-trigger logging points	5000	Setting	×	○
540 (21CH)	740 (2E4H)	940 (3ACH)	1140 (474H)	1340 (53CH)	1540 (604H)	1740 (6CCH)	1940 (794H)	CH□ Level trigger condition setting	0	Setting	×	○
541 (21DH)	741 (2E5H)	941 (3ADH)	1141 (475H)	1341 (53DH)	1541 (605H)	1741 (6CDH)	1941 (795H)	CH□ Trigger data	*2	Setting	×	○
542 (21EH)	742 (2E6H)	942 (3AEH)	1142 (476H)	1342 (53EH)	1542 (606H)	1742 (6CEH)	1942 (796H)	CH□ Trigger setting value	0	Setting	×	○
543 (21FH)	743 (2E7H)	943 (3AFH)	1143 (477H)	1343 (53FH)	1543 (607H)	1743 (6CFH)	1943 (797H)	System area	—	—	—	—
544 (220H)	744 (2E8H)	944 (3B0H)	1144 (478H)	1344 (540H)	1544 (608H)	1744 (6D0H)	1944 (798H)	CH□ Loading interrupt enable/disable setting	1	Setting	×	○
545 (221H)	745 (2E9H)	945 (3B1H)	1145 (479H)	1345 (541H)	1545 (609H)	1745 (6D1H)	1945 (799H)	CH□ Logging read points setting value	1000	Setting	×	○
546 to 597 (222H to 255H)	746 to 797 (2EAH to 31DH)	946 to 997 (3B2H to 3E5H)	1146 to 1197 (47AH to 4ADH)	1346 to 1397 (542H to 575H)	1546 to 1597 (60AH to 63DH)	1746 to 1797 (6D2 to 705H)	1946 to 1997 (79AH to 7CDH)	System area	—	—	—	—
598 (256H)	798 (31EH)	998 (3E6H)	1198 (4AEH)	1398 (576H)	1598 (63EH)	1798 (706H)	1998 (7CEH)	CH□ Range setting	0	Setting	×	○

Address Decimal (hexadecimal)								Name	Default	Data type	Auto refresh	Enabled by Y9 <sup>*1</sup>
CH1	CH2	CH3	CH4	CH5	CH6	CH7	CH8					
599 (257H)	799 (31FH)	999 (3E7H)	1199 (4AFH)	1399 (577H)	1599 (63FH)	1799 (707H)	1999 (7CFH)	System area	—	—	—	—
2000 to 3599 (7D0H to E0FH)												

\*1 Item enabled by turning on and off Operating condition setting request (Y9)

\*2 The following shows the default values.

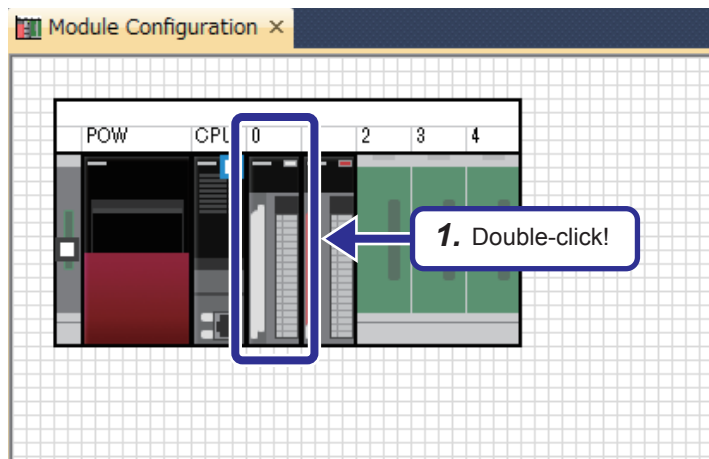
CH1: 402, CH2: 602, CH3: 802, CH4: 1002, CH5: 1202, CH6: 1402, CH7: 1602, CH8: 1802

## 5.6.4 Setting intelligent function module data

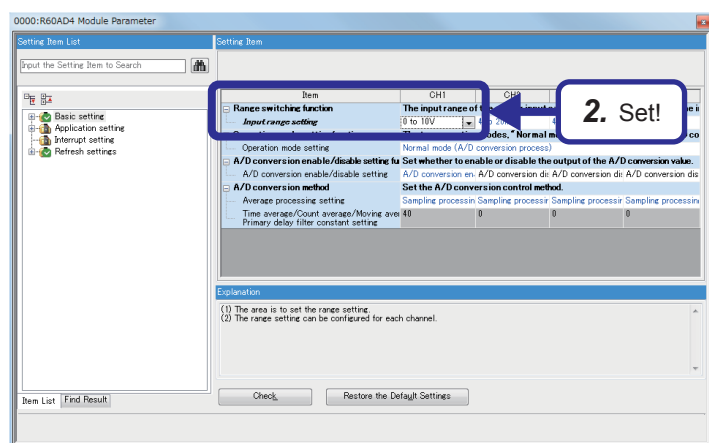
This section describes how to set intelligent function module data.

Set data (including parameters or switch settings) of intelligent function modules from the "Module Configuration" window.

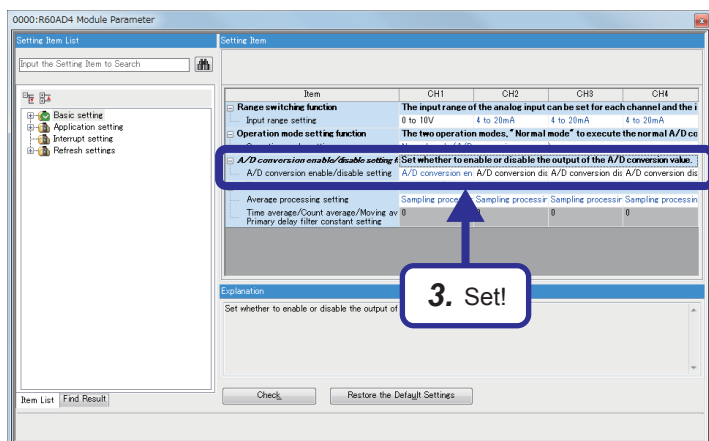
### Operating procedure



1. Double-click the R60AD4 module on the "Module Configuration" window.



2. The "R60AD4 Module Parameter" dialog box appears. Set the input range of CH1 to "0 to 10V".

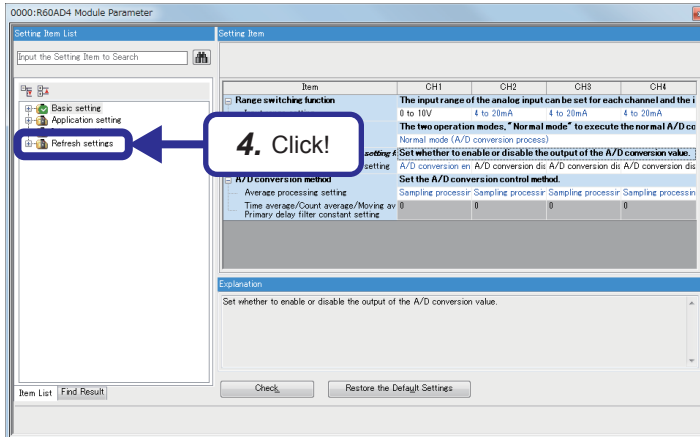


3. Set the A/D conversion enable/disable settings of CH2 to CH4 to "A/D conversion disable".

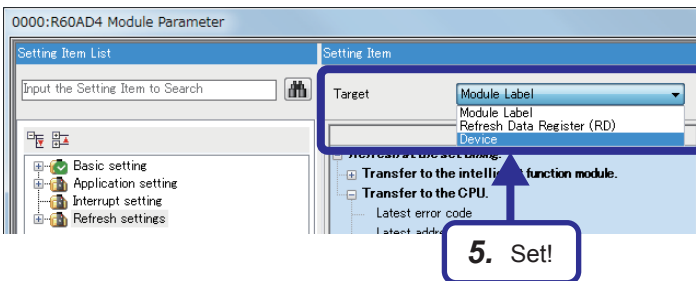


(To the next page)

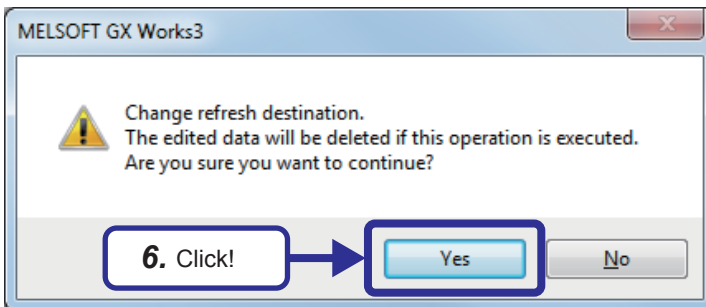
(From the previous page)



4. Click "Refresh settings" in "Setting Item List".



5. Set "Device" for "Target".

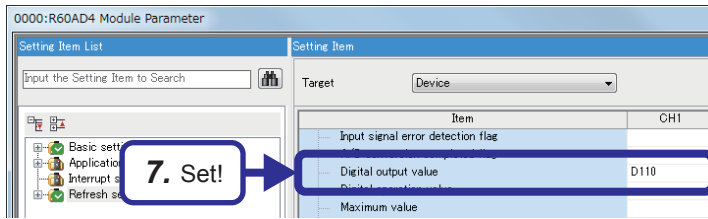


6. The confirmation dialog box appears. Click the [Yes] button.

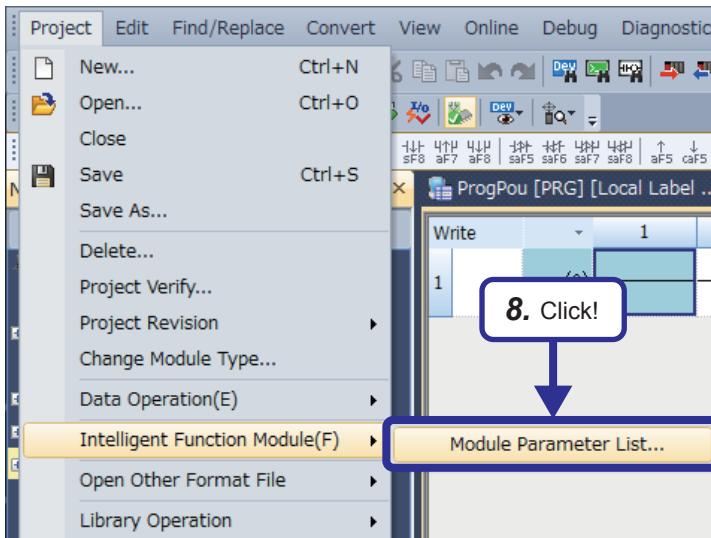


(To the next page)

(From the previous page)

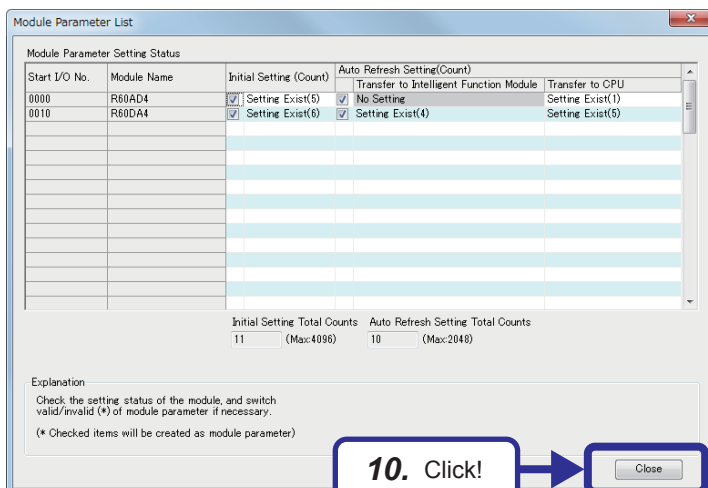


7. Set the digital output value of CH1 to "D110".



8. Click [Project] → [Intelligent Function Module] → [Module Parameter List] from the menu.

5



9. The "Module Parameter List" dialog box appears. Check that the initial setting and auto refresh setting of the R60AD4 have been configured as shown on the left.
10. Click the [Close] button to close the dialog box.

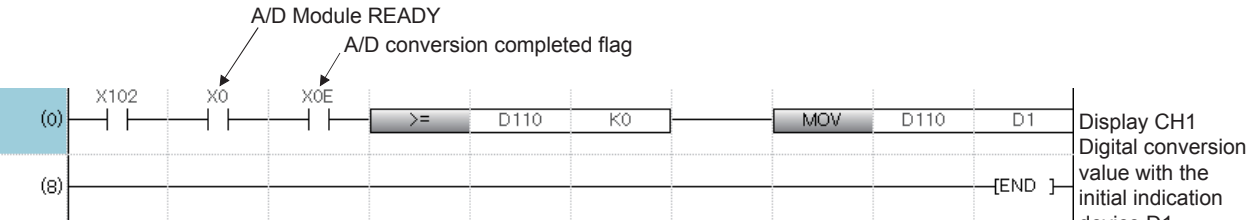


# 5.6.5 Exercise with the demonstration machine

## Sequence program

This sequence program performs the sampling processing on an analog voltage input from CH1 of the R60AD4, and converts it into a digital value.

Project name	EX7
Program name	R60AD4




X0: Module READY signal

X0E: A/D conversion completed flag

This signal turns on to indicate the preparation for the A/D conversion is completed (Module READY) after the power-on or after the reset operation of the CPU module, and the A/D conversion processing will be performed.

Save the created and set program.

## Operating the demonstration machine

Set the CPU module in the STOP state, and click .

Select write target files and parameters and the destination, and click the [Execute] button. Then, set the CPU module in the RUN state, and check that the following operations are performed.

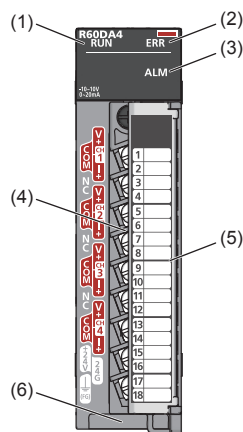
1. Turn on X102, and change the input voltage of the A/D converter module with the volume on the demonstration machine. The analog value input to the channel 1 (CH1) of the R60AD4 is stored in a buffer memory area (digital value). Then, the value is read to the RCPU (data register D110) with the auto refresh setting.
2. The initial indication device (D1) displays the digital conversion value.

## 5.7 R60DA4 Digital-analog Converter Module

### 5.7.1 Part names

This section describes part names of the R60DA4.

For details, refer to the user's manual of the module.



No.	Name	Description
(1)	RUN LED	Indicates the operating status of the module. On: Normal operation Flashing: In offset/gain setting mode Off: 5V power supply interrupted or watchdog timer error occurred
(2)	ERR LED	Indicates the error status of the module.* 1 On: Error occurred Off: Normal operation
(3)	ALM LED	Indicates the alarm status of the module.* 1 On: Alert output occurred Off: Normal operation
(4)	Terminal block	18-point screw terminal block for connecting input signal wires of external devices and others
(5)	Terminal block cover	Covers for preventing electric shock while the power is on
(6)	Production information marking	Shows the product information (16 digits) of the module.

\*1 For details, refer to the following.

MELSEC iQ-R Digital-Analog Converter Module User's Manual (Application)

## 5.7.2 D/A conversion characteristics

### D/A conversion characteristics at the voltage output

(When the output range is set to -10 to 10V)

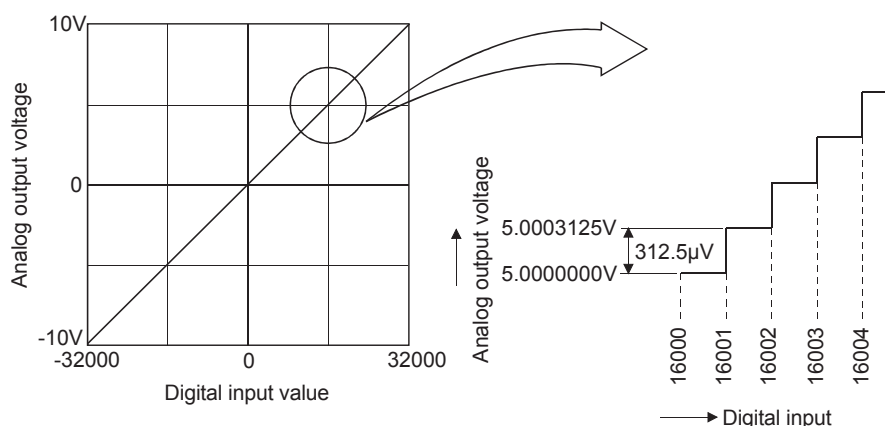


Figure 5.8 D/A conversion characteristics (voltage output)

A D/A converter module converts a digital value output from the RCPU into an analog value, and externally outputs it. The D/A converter module outputs an analog value of -10V when a digital value of -32000 is input and an analog value of 10V when a digital value of 32000 is input. Thus, a digital value of 1 corresponds to an analog value of 312.5μV. If a digital value is a numerical figure after a decimal point, the value cannot be converted.

### D/A conversion characteristics at the current output

(When the output range is set to 0 to 20mA)

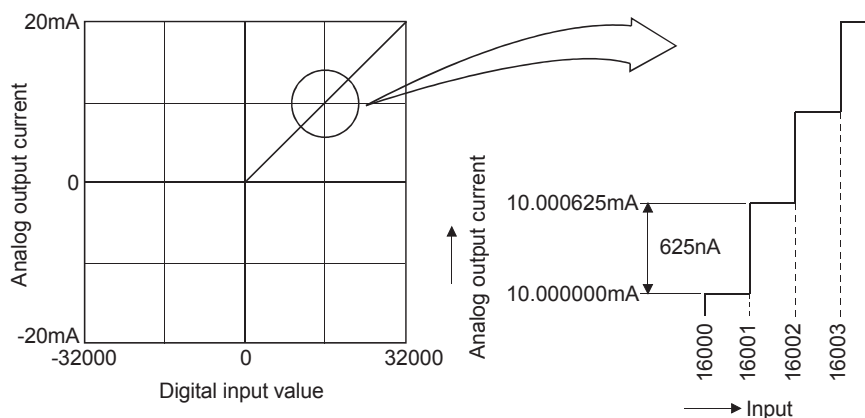


Figure 5.9 D/A conversion characteristics (current output)

At the current output, the D/A converter module converts a digital value of 0 into 0mA and a digital value of 32000 into 20mA. A digital value of 1 corresponds to an analog value of 625nA. If a digital value is a numerical figure after a decimal point, the value cannot be converted.

#### Point

A voltage or current value (maximum resolution) corresponding to a digital value of 1 in D/A conversion differs depending on the input range setting.

## 5.7.3 List of I/O signals and buffer memory assignment

### List of I/O signals

The following table lists the I/O signals of the D/A converter module.

The I/O numbers (X/Y) described in this section are for the case when the start I/O number of the D/A converter module is set to 0.



Do not use the "Use prohibited" signals shown below because the system uses them. If users use (turn on) the signals, the functions of the D/A converter module cannot be guaranteed.

#### Input signal

Device No.	Signal name
X0	Module READY
X1 to X6	Use prohibited
X7	External power supply READY flag
X8	Use prohibited
X9	Operating condition setting completed flag
XA	Offset/gain setting mode status flag
XB	Channel change completed flag
XC	Set value change completed flag
XD	Disconnection detection signal
XE	Alert output signal
XF	Error flag

#### Output signal

Device No.	Signal name
Y0	Use prohibited
Y1	CH1 Output enable/disable flag
Y2	CH2 Output enable/disable flag
Y3	CH3 Output enable/disable flag
Y4	CH4 Output enable/disable flag
Y5 <sup>*1</sup>	CH5 Output enable/disable flag
Y6 <sup>*1</sup>	CH6 Output enable/disable flag
Y7 <sup>*1</sup>	CH7 Output enable/disable flag
Y8 <sup>*1</sup>	CH8 Output enable/disable flag
Y9	Operating condition setting request
YA	User range write request
YB	Channel change request
YC	Value change request
YD	Use prohibited
YE	Alert output clear request
YF	Error clear request

\*1 For the R60DA4, using Y5 to Y8 is prohibited.

## Buffer memory assignment

The module has two operation modes. In R mode, the module operates with the buffer memory map that has been newly provided for the MELSEC iQ-R series. In Q compatible mode, the module operates with the buffer memory map converted to the equivalent one for the MELSEC-Q series.

The following list of buffer memory addresses is for the R mode.

For details, refer to the following.

📖 MELSEC iQ-R Digital-Analog Converter Module User's Manual (Application)

### ■Un\G0 to Un\G399

Address (decimal)	Address (hexadecimal)	Name	Default	Data type	Auto refresh	Enabled by Y9 <sup>*1</sup>
0	0H	Latest error code	0	Monitor	○	—
1	1H	Latest address of error history	0	Monitor	○	—
2	2H	Latest alarm code	0	Monitor	○	—
3	3H	Latest address of alarm history	0	Monitor	○	—
4 to 19	4H to 13H	Interrupt factor detection flag [n] <sup>*2</sup>	0	Monitor	○	—
20 to 35	14H to 23H	System area	—	—	—	—
36	24H	Alert output upper limit flag	0000H	Monitor	○	—
37	25H	Alert output lower limit flag	0000H	Monitor	○	—
38	26H	Disconnection detection flag	0000H	Monitor	○	—
39 to 59	27H to 3BH	System area	—	—	—	—
60	3CH	Output mode	0000H	Monitor	×	○
61 to 123	3DH to 7BH	System area	—	—	—	—
124 to 139	7CH to 8BH	Interrupt factor mask [n] <sup>*2</sup>	0	Control	×	—
140 to 155	8CH to 9BH	System area	—	—	—	—
156 to 171	9CH to ABH	Interrupt factor reset request [n] <sup>*2</sup>	0	Control	×	—
172 to 187	ACH to BBH	System area	—	—	—	—
188	BCH	Step action wave output request	0	Control	×	—
189 to 199	BDH to C7H	System area	—	—	—	—
200 to 215	C8H to D7H	Interrupt factor generation setting [n] <sup>*2</sup>	0	Setting	×	○
216 to 231	D8H to E7H	System area	—	—	—	—
232 to 247	E8H to F7H	Condition target setting [n] <sup>*2</sup>	0	Setting	×	○
248 to 263	F8H to 107H	System area	—	—	—	—
264 to 279	108H to 117H	Condition target channel setting [n] <sup>*2</sup>	0	Setting	×	○
280 to 295	118H to 127H	System area	—	—	—	—
296, 297	128H, 129H	Mode switching setting	0	Setting	×	○
298 to 399	130H to 18FH	System area	—	—	—	—

\*1 Item enabled by turning on and off 'Operating condition setting request' (Y9)

\*2 [n] in the table indicates an interrupt setting number (n = 1 to 16).

## ■Un\G400 to Un\G3599

Address Decimal (hexadecimal)								Name	Default	Data type	Auto refresh	Enabled by Y9*1
CH1	CH2	CH3	CH4	CH5	CH6	CH7	CH8					
400 (190H)	600 (258H)	800 (320H)	1000 (3E8H)	1200 (4B0H)	1400 (578H)	1600 (640H)	1800 (708H)	CH□ Setting value check code	0	Monitor	○	—
401 (191H)	601 (259H)	801 (321H)	1001 (3E9H)	1201 (4B1H)	1401 (579H)	1601 (641H)	1801 (709H)	CH□ Wave output status monitor	0	Monitor	○	—
402 to 429 (192H to 1ADH)	602 to 629 (25AH to 275H)	802 to 829 (322H to 33DH)	1002 to 1029 (3EAH to 405H)	1202 to 1229 (4B2H to 4CDH)	1402 to 1429 (57AH to 595H)	1602 to 1629 (642H to 65DH)	1802 to 1829 (70AH to 725H)	System area	—	—	—	—
430 (1AEH)	630 (276H)	830 (33EH)	1030 (406H)	1230 (4CEH)	1430 (596H)	1630 (65EH)	1830 (726H)	CH□ Range setting monitor	0 —	Monitor	×	—
431 (1AFH)	631 (277H)	831 (33FH)	1031 (407H)	1231 (4CFH)	1431 (597H)	1631 (65FH)	1831 (727H)	CH□ HOLD/CLEAR function setting monitor	0	Monitor	×	—
432 (1B0H)	632 (278H)	832 (340H)	1032 (408H)	1232 (4D0H)	1432 (598H)	1632 (660H)	1832 (728H)	CH□ Wave output conversion cycle monitor (L)	0	Monitor	×	—
433 (1B1H)	633 (279H)	833 (341H)	1033 (409H)	1233 (4D1H)	1433 (599H)	1633 (661H)	1833 (729H)	CH□ Wave output conversion cycle monitor (H)		Monitor	×	—
434 (1B2H)	634 (27AH)	834 (342H)	1034 (40AH)	1234 (4D2H)	1434 (59AH)	1634 (662H)	1834 (72AH)	CH□ Wave pattern output count monitor	0	Monitor	×	—
435 (1B3H)	635 (27BH)	835 (343H)	1035 (40BH)	1235 (4D3H)	1435 (59BH)	1635 (663H)	1835 (72BH)	System area	—	—	—	—
436 (1B4H)	636 (27CH)	836 (344H)	1036 (40CH)	1236 (4D4H)	1436 (59CH)	1636 (664H)	1836 (72CH)	CH□ Wave output current address monitor (L)	0	Monitor	×	—
437 (1B5H)	637 (27DH)	837 (345H)	1037 (40DH)	1237 (4D5H)	1437 (59DH)	1637 (665H)	1837 (72DH)	CH□ Wave output current address monitor (H)		Monitor	×	—
438 (1B6H)	638 (27EH)	838 (346H)	1038 (40EH)	1238 (4D6H)	1438 (59EH)	1638 (666H)	1838 (72EH)	CH□ Wave output current digital value monitor	0	Monitor	×	—
439 (1B7H)	639 (27FH)	839 (347H)	1039 (40FH)	1239 (4D7H)	1439 (59FH)	1639 (667H)	1839 (72FH)	System area	—	—	—	—
440 (1B8H)	640 (280H)	840 (348H)	1040 (410H)	1240 (4D8H)	1440 (5A0H)	1640 (668H)	1840 (730H)	CH□ Wave output digital value out-of-range address monitor (L)	0	Monitor	×	—
441 (1B9H)	641 (281H)	841 (349H)	1041 (411H)	1241 (4D9H)	1441 (5A1H)	1641 (669H)	1841 (731H)	CH□ Wave output digital value out-of-range address monitor (H)		Monitor	×	—
442 (1BAH)	642 (282H)	842 (34AH)	1042 (412H)	1242 (4DAH)	1442 (5A2H)	1642 (66AH)	1842 (732H)	CH□ Wave output warning address monitor (L)	0	Monitor	×	—
443 (1BBH)	643 (283H)	843 (34BH)	1043 (413H)	1243 (4DBH)	1443 (5A3H)	1643 (66BH)	1843 (733H)	CH□ Wave output warning address monitor (H)		Monitor	×	—
444 to 459 (1BCH to 1CBH)	644 to 659 (284H to 293H)	844 to 859 (34CH to 35BH)	1044 to 1059 (414H to 423H)	1244 to 1259 (4DCH to 4EBH)	1444 to 1459 (5A4H to 5B3H)	1644 to 1659 (66CH to 67BH)	1844 to 1859 (734H to 743H)	System area	—	—	—	—
460 (1CCH)	660 (294H)	860 (35CH)	1060 (424H)	1260 (4ECH)	1460 (5B4H)	1660 (67CH)	1860 (744H)	CH□ Digital value	0	Control	○	—
461 (1CDH)	661 (295H)	861 (35DH)	1061 (425H)	1261 (4EDH)	1461 (5B5H)	1661 (67DH)	1861 (745H)	System area	—	—	—	—
462 (1CEH)	662 (296H)	862 (35EH)	1062 (426H)	1262 (4EEH)	1462 (5B6H)	1662 (67EH)	1862 (746H)	CH□ Wave output start/ stop request	0	Control	×	—

Address Decimal (hexadecimal)								Name	Default	Data type	Auto refresh	Enabled by Y9*1
CH1	CH2	CH3	CH4	CH5	CH6	CH7	CH8					
463 to 479 (1CFH to 1DFH)	663 to 679 (297H to 2A7H)	863 to 879 (35FH to 36FH)	1063 to 1079 (427H to 437H)	1263 to 1279 (4EFH to 4FFH)	1463 to 1479 (5B7H to 5C7H)	1663 to 1679 (67FH to 68FH)	1863 to 1879 (747H to 757H)	System area	—	—	—	—
480 (1E0H)	680 (2A8H)	880 (370H)	1080 (438H)	1280 (500H)	1480 (5C8H)	1680 (690H)	1880 (758H)	CH□ Input value shift amount	0	Control	×	—
481 (1E1H)	681 (2A9H)	881 (371H)	1081 (439H)	1281 (501H)	1481 (5C9H)	1681 (691H)	1881 (759H)	System area	—	—	—	—
482 (1E2H)	682 (2AAH)	882 (372H)	1082 (43AH)	1282 (502H)	1482 (5CAH)	1682 (692H)	1882 (75AH)	CH□ Wave output step action movement amount	0	Control	×	—
483 to 499 (1E3H to 1F3H)	683 to 699 (2ABH to 2BBH)	883 to 899 (373H to 383H)	1083 to 1099 (43BH to 44BH)	1283 to 1299 (503H to 513H)	1483 to 1499 (5CBH to 5DBH)	1683 to 1699 (693H to 6A3H)	1883 to 1899 (75BH to 76BH)	System area	—	—	—	—
500 (1F4H)	700 (2BCH)	900 (384H)	1100 (44CH)	1300 (514H)	1500 (5DCH)	1700 (6A4H)	1900 (76CH)	CH□ D/A conversion enable/disable setting	1	Setting	×	○
501 (1F5H)	701 (2BDH)	901 (385H)	1101 (44DH)	1301 (515H)	1501 (5DDH)	1701 (6A5H)	1901 (76DH)	System area	—	—	—	—
502 (1F6H)	702 (2BEH)	902 (386H)	1102 (44EH)	1302 (516H)	1502 (5DEH)	1702 (6A6H)	1902 (76EH)	CH□ Scaling enable/ disable setting	1	Setting	×	○
503 (1F7H)	703 (2BFH)	903 (387H)	1103 (44FH)	1303 (517H)	1503 (5DFH)	1703 (6A7H)	1903 (76FH)	System area	—	—	—	—
504 (1F8H)	704 (2C0H)	904 (388H)	1104 (450H)	1304 (518H)	1504 (5E0H)	1704 (6A8H)	1904 (770H)	CH□ Scaling upper limit value	0	Setting	×	○
505 (1F9H)	705 (2C1H)	905 (389H)	1105 (451H)	1305 (519H)	1505 (5E1H)	1705 (6A9H)	1905 (771H)	System area	—	—	—	—
506 (1FAH)	706 (2C2H)	906 (38AH)	1106 (452H)	1306 (51AH)	1506 (5E2H)	1706 (6AAH)	1906 (772H)	CH□ Scaling lower limit value	0	Setting	×	○
507 (1FBH)	707 (2C3H)	907 (38BH)	1107 (453H)	1307 (51BH)	1507 (5E3H)	1707 (6ABH)	1907 (773H)	System area	—	—	—	—
508 (1FCH)	708 (2C4H)	908 (38CH)	1108 (454H)	1308 (51CH)	1508 (5E4H)	1708 (6ACH)	1908 (774H)	CH□ Alert output setting	1	Setting	×	○
509 (1FDH)	709 (2C5H)	909 (38DH)	1109 (455H)	1309 (51DH)	1509 (5E5H)	1709 (6ADH)	1909 (775H)	CH□ Rate control enable/disable setting	1	Setting	×	○
510 (1FEH)	710 (2C6H)	910 (38EH)	1110 (456H)	1310 (51EH)	1510 (5E6H)	1710 (6AEH)	1910 (776H)	CH□ Alert output upper limit value	0	Setting	×	○
511 (1FFH)	711 (2C7H)	911 (38FH)	1111 (457H)	1311 (51FH)	1511 (5E7H)	1711 (6AFH)	1911 (777H)	System area	—	—	—	—
512 (200H)	712 (2C8H)	912 (390H)	1112 (458H)	1312 (520H)	1512 (5E8H)	1712 (6B0H)	1912 (778H)	CH□ Alert output lower limit value	0	Setting	×	○
513 (201H)	713 (2C9H)	913 (391H)	1113 (459H)	1313 (521H)	1513 (5E9H)	1713 (6B1H)	1913 (779H)	System area	—	—	—	—
514 (202H)	714 (2CAH)	914 (392H)	1114 (45AH)	1314 (522H)	1514 (5EAH)	1714 (6B2H)	1914 (77AH)	CH□ Increase digital limit value	64000	Setting	×	○
515 (203H)	715 (2CBH)	915 (393H)	1115 (45BH)	1315 (523H)	1515 (5EBH)	1715 (6B3H)	1915 (77BH)	System area	—	—	—	—
516 (204H)	716 (2CCH)	916 (394H)	1116 (45CH)	1316 (524H)	1516 (5ECH)	1716 (6B4H)	1916 (77CH)	CH□ Decrease digital limit value	64000	Setting	×	○
517 to 523 (205H to 20BH)	717 to 723 (2CDH to 2D3H)	917 to 923 (395H to 39BH)	1117 to 1123 (45DH to 463H)	1317 to 1323 (525H to 52BH)	1517 to 1523 (5EDH to 5F3H)	1717 to 1723 (6B5H to 6BBH)	1917 to 1923 (77DH to 783H)	System area	—	—	—	—
524 (20CH)	724 (2D4H)	924 (39CH)	1124 (464H)	1324 (52CH)	1524 (5F4H)	1724 (6BCH)	1924 (784H)	CH□ Output setting during wave output stop	1	Setting	×	—
525 (20DH)	725 (2D5H)	925 (39DH)	1125 (465H)	1325 (52DH)	1525 (5F5H)	1725 (6BDH)	1925 (785H)	CH□ Output value during wave output stop	0	Setting	×	—

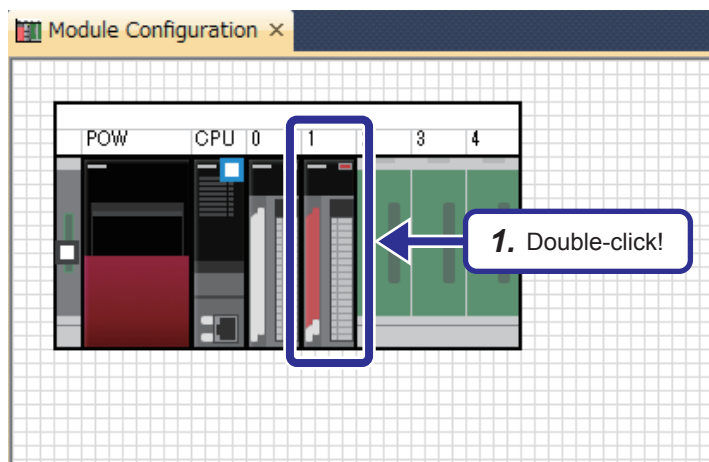
Address Decimal (hexadecimal)								Name	Default	Data type	Auto refresh	Enabled by Y9 <sup>*1</sup>
CH1	CH2	CH3	CH4	CH5	CH6	CH7	CH8					
526 (20EH)	726 (2D6H)	926 (39EH)	1126 (466H)	1326 (52EH)	1526 (5F6H)	1726 (6BEH)	1926 (786H)	CH□ Wave pattern start address setting (L)	10000	Setting	×	—
527 (20FH)	727 (2D7H)	927 (39FH)	1127 (467H)	1327 (52FH)	1527 (5F7H)	1727 (6BFH)	1927 (787H)	CH□ Wave pattern start address setting (H)		Setting	×	—
528 (210H)	728 (2D8H)	928 (3A0H)	1128 (468H)	1328 (530H)	1528 (5F8H)	1728 (6C0H)	1928 (788H)	CH□ Wave pattern data points setting (L)	0	Setting	×	—
529 (211H)	729 (2D9H)	929 (3A1H)	1129 (469H)	1329 (531H)	1529 (5F9H)	1729 (6C1H)	1929 (789H)	CH□ Wave pattern data points setting (H)		Setting	×	—
530 (212H)	730 (2DAH)	930 (3A2H)	1130 (46AH)	1330 (532H)	1530 (5FAH)	1730 (6C2H)	1930 (78AH)	CH□ Wave pattern output repetition setting	1	Setting	×	—
531 (213H)	731 (2DBH)	931 (3A3H)	1131 (46BH)	1331 (533H)	1531 (5FBH)	1731 (6C3H)	1931 (78BH)	CH□ Constant for wave output conversion cycle	1	Setting	×	—
532 to 597 (214H to 255H)	732 to 797 (2DCH to 31DH)	932 to 997 (3A4H to 3E5H)	1132 to 1197 (46CH to 4ADH)	1332 to 1397 (534H to 575H)	1532 to 1597 (5FCH to 63DH)	1732 to 1797 (6C4H to 705H)	1932 to 1997 (78CH to 7CDH)	System area	—	—	—	—
598 (256H)	798 (31EH)	998 (3E6H)	1198 (4AEH)	1398 (576H)	1598 (63EH)	1798 (706H)	1998 (7CEH)	CH□ Range setting	0	Setting	×	○
599 (257H)	799 (31FH)	999 (3E7H)	1199 (4AFH)	1399 (577H)	1599 (63FH)	1799 (707H)	1999 (7CFH)	System area	—	—	—	—
2000 to 3599 (7D0H to E0FH)								System area	—	—	—	—

\*1 Item enabled by turning on and off 'Operating condition setting request' (Y9)

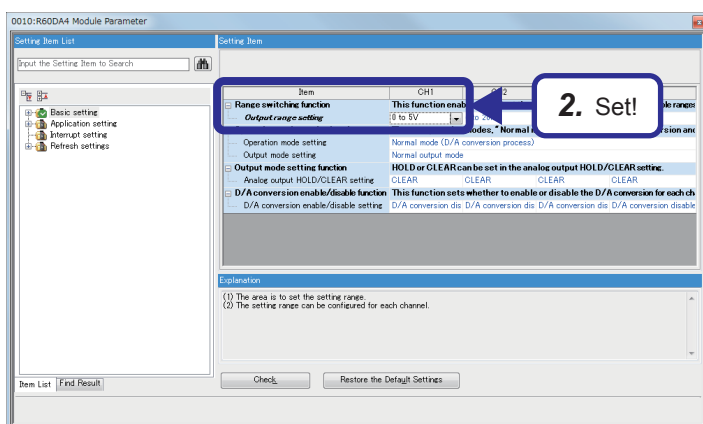


## 5.7.4 Setting intelligent function module data

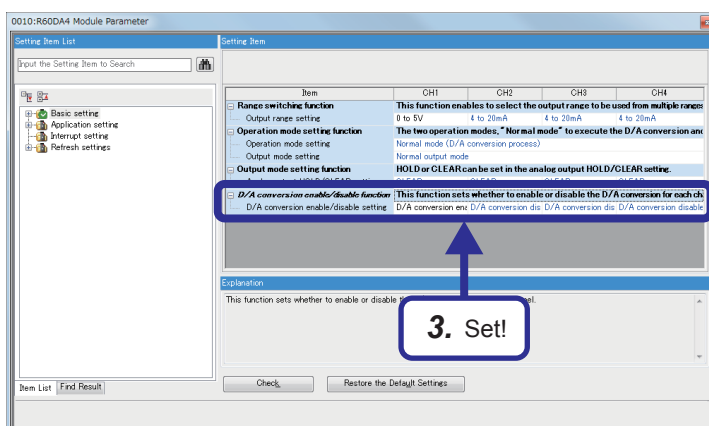
### Operating procedure



1. Double-click the R60DA4 module on the "Module Configuration" window.



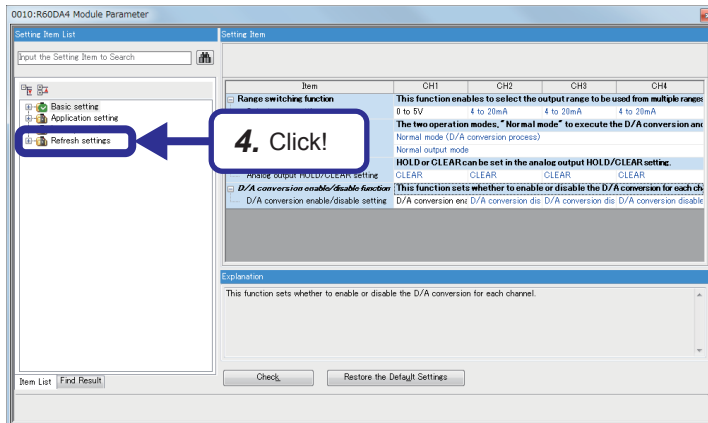
2. The "R60DA4 Module Parameter" dialog box appears. Set the output range of CH1 to "0 to 5V".



3. Set the D/A conversion enable/disable setting of CH1 to "D/A conversion enable".

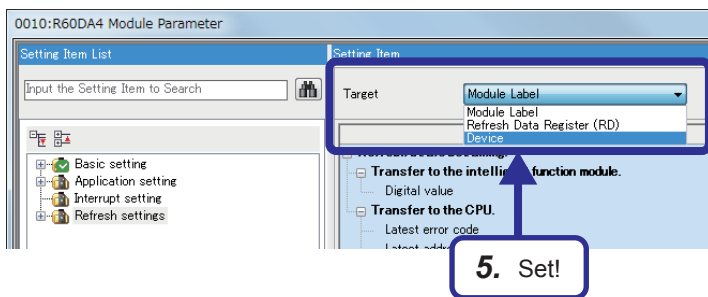
(To the next page)

(From the previous page)

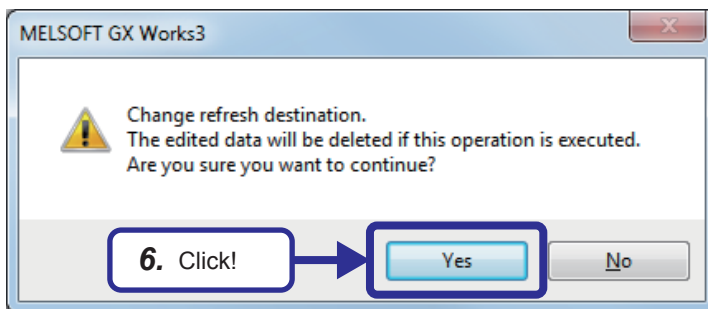


4. Click "Refresh settings" in "Setting Item List".

5



5. Set "Device" for "Target".

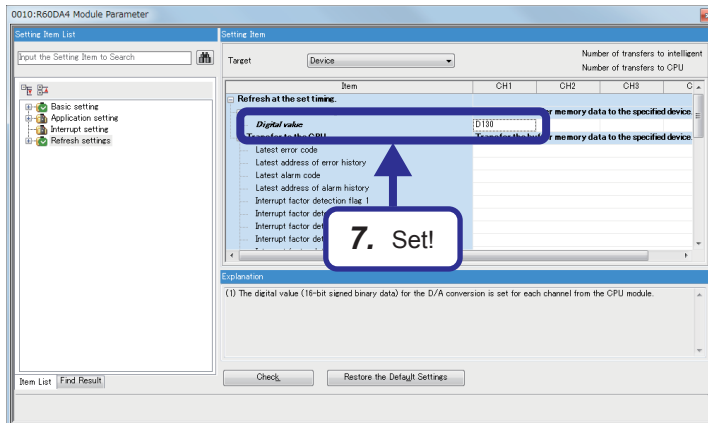


6. The confirmation dialog box appears. Click the [Yes] button.

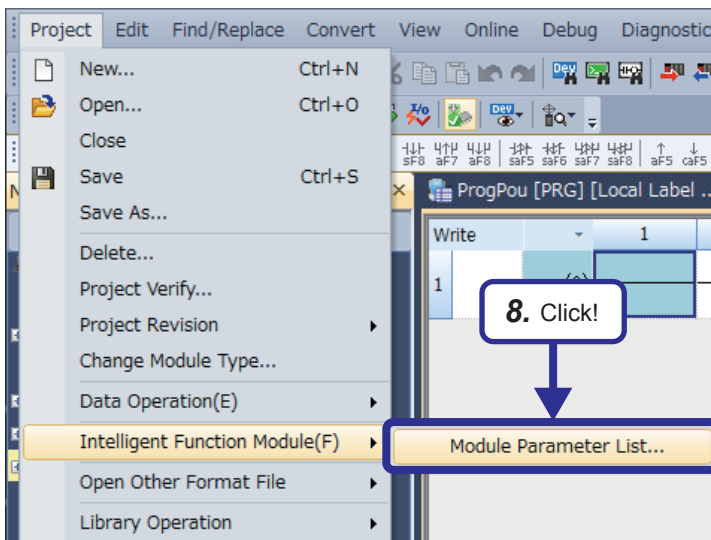


(To the next page)

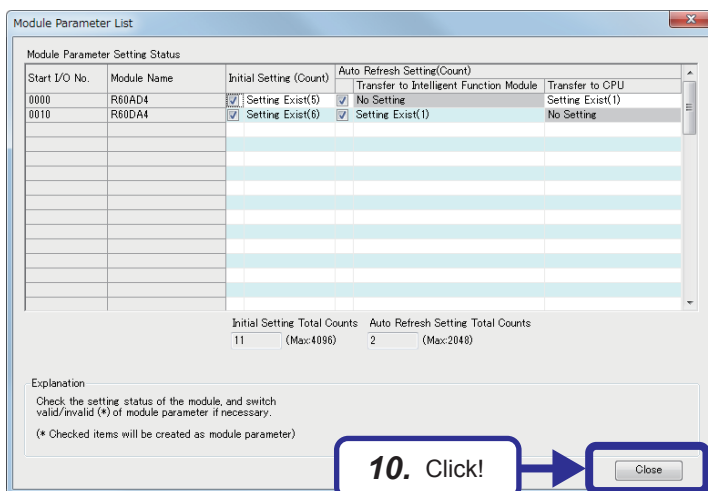
(From the previous page)



7. Set the digital value of CH1 to "D130".



8. Click [Project] → [Intelligent Function Module] → [Module Parameter List] from the menu.



9. The "Module Parameter List" dialog box appears. Check that the initial setting and auto refresh setting of the R60DA4 have been configured as shown on the left.

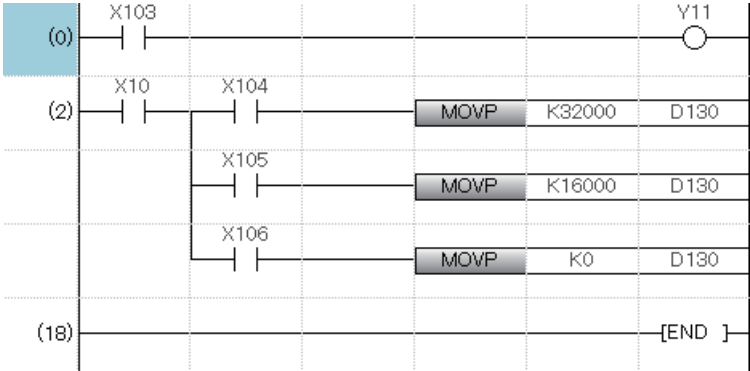
10. Click the [Close] button to close the dialog box.

# 5.7.5     Exercise with the demonstration machine

## Sequence program

This sequence program converts an output voltage into an analog signal.

Project name	EX7
Program name	R60DA4



- X10: Module READY signal  
This signal turns on to indicate the preparation for the D/A conversion is completed after the power-on or after the reset operation of the CPU module, and the D/A conversion processing will be performed.
- Y11: CH1 Output enable/disable flag  
Set whether to output D/A conversion values or offset values for each channel.  
On: D/A conversion value, off: Offset value

## Operating the demonstration machine

Set the CPU module in the STOP state, and click .  
Select write target files and parameters and the destination, and click the [Execute] button. Then, set the CPU module in the RUN state, and check that the following operations are performed.

1. Turn on X103, and enable outputting of a D/A conversion value from CH1.
2. Voltages corresponding to X104 to X106 are output.
3. The D/A OUTPUT voltmeter on the screen 2 displays the output voltage of the R60DA4.

# MEMO

---

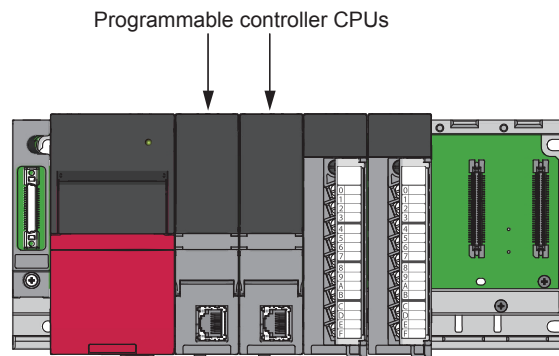
# 6 MULTIPLE CPU SYSTEM

## Point

- This section describes features of a multiple CPU system.
- This section describes how to start a multiple CPU system.
- This section describes how to perform data communications among CPU modules.

## 6.1 Overview

In a multiple CPU system, more than one CPU module is mounted on the main base unit and each CPU module controls I/O modules and intelligent function modules separately.



## 6.2 Features

### Multiple controllers

- (1) Configuration suitable for the system to be used  
Users can build one system using not only one RCPUs but also multiple CPUs of RCPUs, Motion CPUs, and C Controller modules according to the system to be used. Creating a multiple CPU system improves the efficiency and maintainability of the system.
- (2) Controlling of modules  
Each CPU module of the multiple CPU system controls I/O modules and intelligent function modules mounted on a base unit by slot.  
The I/O modules and intelligent function modules controlled by each CPU module of the multiple CPU system are set with GX Works3.

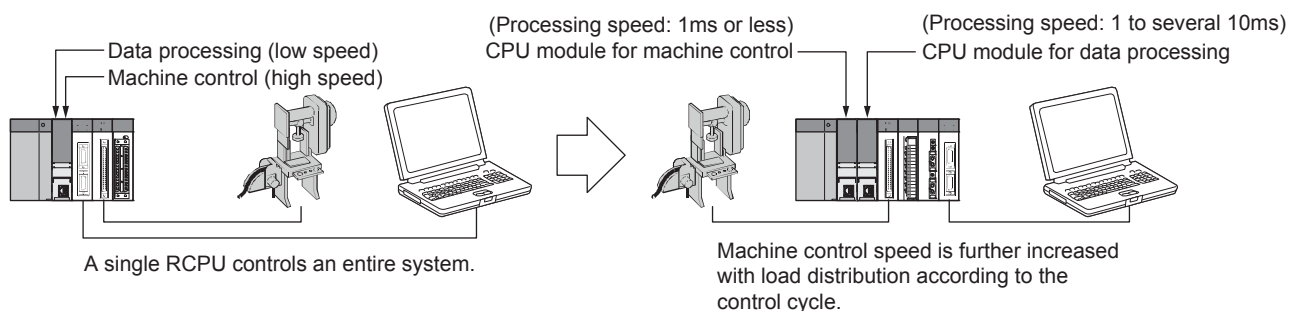
### Building a sequence control system and motion control system on a single base unit

With the multiple CPU system of the RCPUs and Motion CPU, the motion system that is the combination of sequence control and motion control on one base unit can be established.

## Distribution of loads

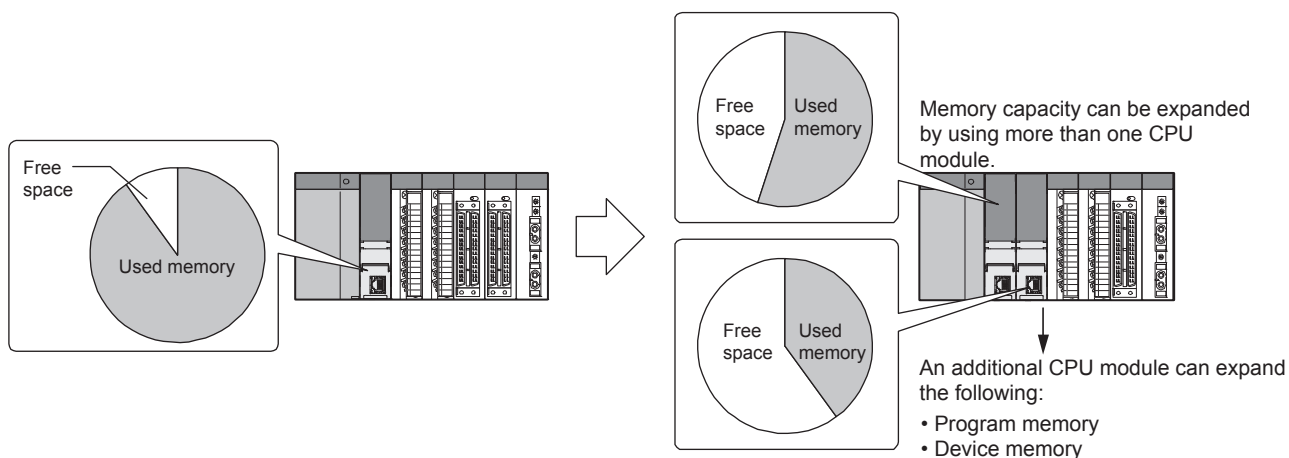
### (1) Distribution of processing

The overall system scan time can be reduced by distributing the high-load processing performed in a single RCPU over multiple CPU modules.



### (2) Distribution of memory

The memory capacity used for the entire system can be increased by distributing the memory areas over multiple CPU modules.



## Distribution of functions

Programs can be developed easily by distributing the functions, for example, having different CPU modules control production line A and production line B.

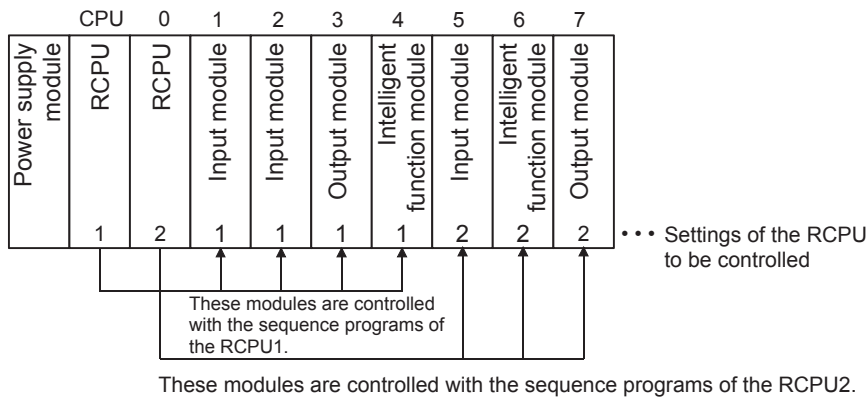
# 6.3 Differences with a Single CPU System

This section describes differences between a single CPU system and a multiple CPU system.

## 6.3.1 Mounting positions of modules

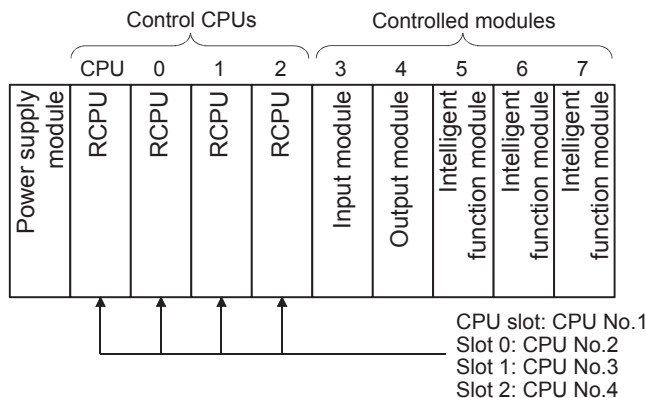
In a multiple CPU system, more than one CPU module (up to four CPU modules) is mounted on the main base unit and each CPU module controls I/O modules and intelligent function modules separately.

Loads can be distributed by processing and highly-loaded processing can be executed with multiple CPU modules.



In a multiple CPU system, each module is defined as follows.

- Control CPU : RCPU that controls I/O modules and intelligent function modules
- Controlled module : I/O modules and intelligent function modules controlled by control CPUs
- Non-controlled module : Modules controlled by another CPU module
- CPU No. : Number assigned to each CPU module. The CPU module mounted on the CPU slot is CPU No.1, and CPU No.2, CPU No.3, and CPU No.4 are assigned to CPU modules mounted on the slots on the left of the CPU slot.



Up to four CPU modules can be mounted in the CPU slot (the slot on the right of the power supply module) to the slot No.2 of the main base unit.



## 6.3.2 I/O number assignment

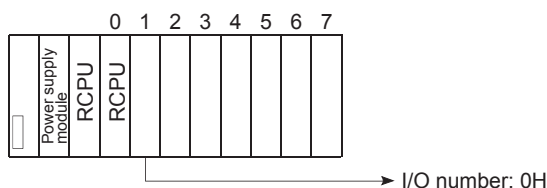
In a multiple CPU system, the position (slot) of the I/O number 0H differs from that in the single CPU system.

However, I/O numbers of extension base units, I/O numbers of each slot, and I/O numbers of empty slots are all assigned in the same way as the ones of a single CPU system.

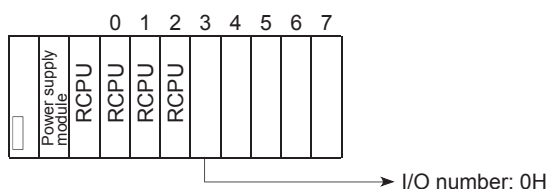
### Position of the I/O number "0H"

- In a multiple CPU system, CPU modules occupy for the number of slots set in the multiple CPU settings of the system parameter.
- I/O modules and intelligent function modules are mounted to the right of CPU modules.
- The I/O number "0H" is assigned to the I/O module or intelligent function module mounted to the right of the CPU module. The subsequent I/O numbers are assigned sequentially to the right.

1) RCPU: When two CPU modules are mounted

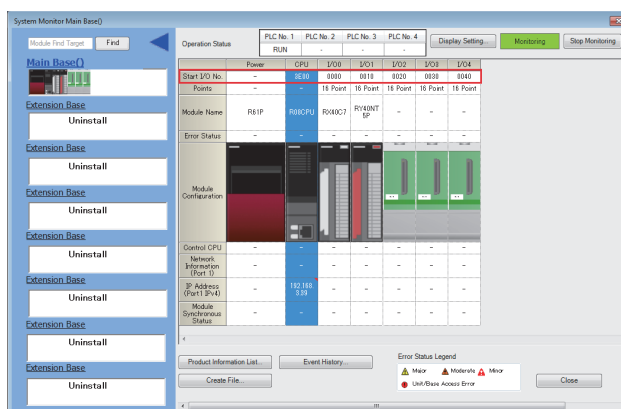


2) RCPU: When four CPU modules are mounted



The mounted modules and their I/O numbers can be checked on the "System Monitor" window using the engineering tool.

( GX Works3 Operating Manual)



## 6.3.3 Data communications among CPU modules and controlled modules

### Access to controlled modules

The range of accesses to controlled modules of the CPU modules is the same as that of a single CPU system. I/O refresh for the controlled modules and reading/writing of data from/to the buffer memory areas of intelligent function modules can be performed.

### Access to non-controlled modules

Access from the CPU modules to non-controlled modules is limited to reading of data from the input device (DX) and buffer memory of the intelligent function modules. On/off data of input (X) and output (Y) devices of non-controlled modules can be loaded with the out-of-group I/O fetch function.

○: Accessible, ×: Not accessible

Communication method	Access target		Access range	
			When "Import" is selected in the settings for out-of-group I/O fetch	When "Not Imported" is selected in the settings for out-of-group I/O fetch
Communication through refresh	Input (X)	Read from another CPU module	○ <sup>*1</sup>	× (Non-processing)
		Write to another CPU module	× (Non-processing)	
	Output (Y)	Read from another CPU module	○ <sup>*1</sup>	
		Write to another CPU module	× (Non-processing)	× (Cannot be specified with the engineering tool.)
	Buffer memory of intelligent function modules	Read	× (Cannot be specified with the engineering tool.)	
Communication through direct access	Input (DX)	Read from another CPU module	○	○
		Write to another CPU module	× (Non-processing)	× (Non-processing)
	Output (DY)	Read from another CPU module	× (Cannot be specified.)	× (Cannot be specified.)
		Write to another CPU module	× (Non-processing)	× (Non-processing)
	Buffer memory of intelligent function modules	Read	○	○
		Write	× (An error occurs in the CPU module.)	× (An error occurs in the CPU module.)

\*1 Data cannot be loaded from CPU modules that are set as the targets of the inter-module synchronization function.



Access ranges vary depending on the type of the CPU module in use. For details, refer to the manual of each CPU module.

## 6.4 Data Communications Among CPU Modules in a Multiple CPU System

CPU modules in a multiple CPU system transfer data each other. Refresh communications and direct access communications allow the CPU modules to write and read data each other. The following table describes data communication methods.

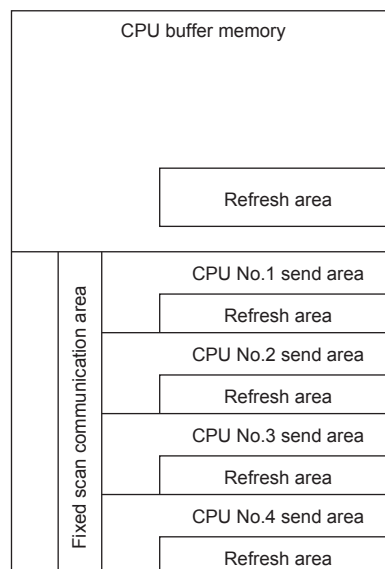
Communication method	Application	Description
Data communication with CPU buffer memory	Use this method to transfer data at the timing of each CPU module.	The sender CPU module writes data to its CPU buffer memory. The receiver CPU module reads the data from the CPU buffer memory of the sender CPU module (another CPU module).
Data communication with fixed scan communication areas	Use this method to transfer data by adjusting the timing of CPU modules.	The sender CPU module writes data to its fixed scan communication area (send area). The receiver CPU module reads the data from the fixed scan communication area (receive area) of the sender CPU module.

### 6.4.1 Memory to be used

CPU buffer memory is used for data communications among CPU modules.

#### Memory configuration of CPU buffer memory

The following figure shows the configuration of CPU buffer memory.



Memory	Communication method	Description	Area size
CPU buffer memory	Communication through direct access	This area is used to read/write data from/to areas of the host CPU module or another CPU module.	512K words (fixed)  Size of the areas can be changed within the range of 0 to 24K words in the entire system. Size of send area of one CPU module can be set within the range of 0 to 12K words.
Refresh area	Communication through refresh	Data communication is performed through refresh at END processing.	
Fixed scan communication area	Communication through direct access	The host CPU module reads/writes data from/to its fixed scan communication area, and the data is transferred between the area and fixed scan communication area of another CPU module at fixed scan communication cycles.	
Refresh area in fixed scan communication area	Communication through refresh	Data is refreshed at a fixed scan communication cycle.	

## 6.4.2 Communication through refresh

The device data for each CPU module is written/read only by the parameter settings. Using refresh areas enables data communications among all or some of the CPU modules in a multiple CPU system, and each CPU module can use devices of other CPU modules.

### Types of refresh operations

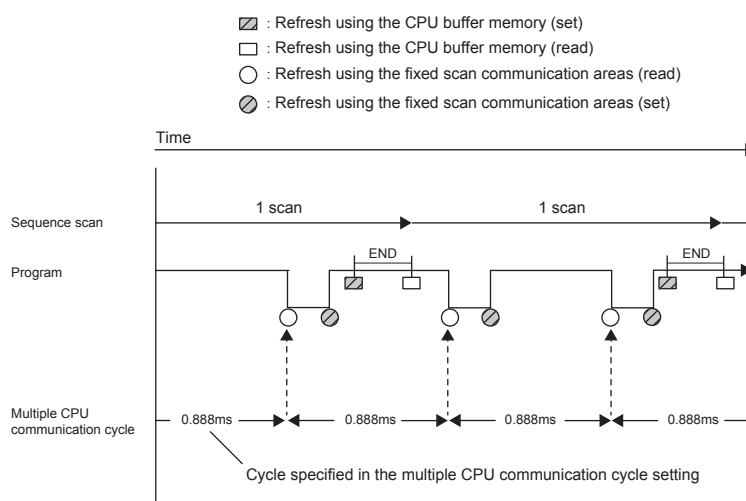
The following table lists the types of refresh operations.

Refresh timing	Application	Memory to be used
At the END processing	For data communications of devices in synchronization with scan	CPU buffer memory
At execution of a multiple CPU synchronous interrupt program (I45) <sup>*1*2</sup>	For data communications of devices at the cycle specified in the fixed scan communication cycle setting	Fixed scan communication area

\*1 Data communications are executed at the cycle specified in the fixed scan communication cycle setting.

\*2 A refresh operation fails if no multiple CPU synchronous interrupt program (I45) exists in the program.

The following figure shows the timing of each refresh operation.

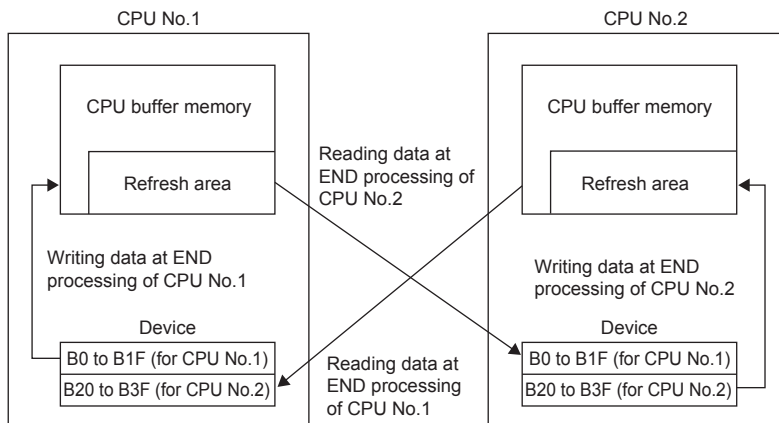


(1) Refresh using the CPU buffer memory

At END processing of the host CPU module, device data of the CPU module is written to the refresh area in the CPU buffer memory of the module. The data written to the refresh area is transferred to devices of another CPU module with the END processing of another CPU module.

**Ex.**

When CPU No.1 refreshes 32 points (B0 to B1F) and CPU No.2 refreshes 32 points (B20 to B3F):

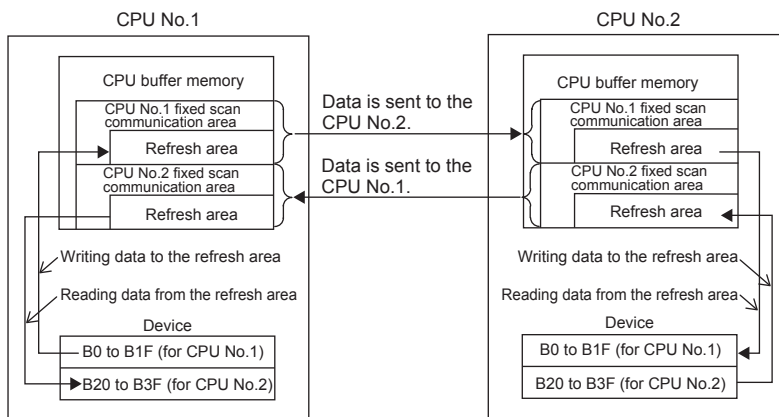


(2) Refresh using the fixed scan communication areas

Device data of the host CPU module is written to the refresh area in the fixed scan communication area of the CPU module at the cycle specified in the fixed scan communication cycle setting. The data written to the refresh area is sent to the refresh area in the fixed scan communication area of another CPU module, and this CPU module reads the transferred data into device areas.

**Ex.**

When CPU No.1 refreshes 32 points (B0 to B1F) and CPU No.2 refreshes 32 points (B20 to B3F):



## Executing refresh operations

Refresh operations are executed when the CPU module is in the RUN or STOP (PAUSE) state.

### (1) Operation at execution of the multiple CPU synchronous interrupt program (I45)

When refresh operations are set to be performed at execution of the multiple CPU synchronous interrupt program (I45), refresh operations to be taken when the CPU module is in the RUN state varies due to one of the following conditions.

- A program (I45 to IRET) containing the multiple CPU synchronous interrupt program (I45) exists.
- An event execution type program exists.

The following tables list refresh operations.

○: Refreshed, ×: Not refreshed

Execution type		Refresh operation		Refresh operation after execution of a program control instruction	
		STOP state	RUN state	RUN state	
Other than event execution type	No interrupt pointer I45	○	×	×	
	I45 interrupt pointer	○	○	○	
Event execution type (with I45 interrupt)		○	○	×	(The operation will stop after the execution type is changed*1.)

\*1 The PSCAN/PSTOP instruction changes the execution type in the subsequent scan, and the POFF instruction changes the type in the scan after the subsequent two scans.

○: Refreshed, ×: Not refreshed

Execution type		Refresh operation triggered by change of the operating status of the CPU module						
		RUN state	When an I45 interrupt pointer is added with online program change	When an I45 interrupt pointer is deleted with online program change	RUN → STOP	STOP state	When parameters are written after the refresh settings have changed	STOP → RUN, when the power is turned off and on, or when the CPU module is reset
Other than event execution type	No interrupt pointer I45	×	○	×	○	○	○*3	×
	I45 interrupt pointer	○	○	×	○*2	○	○*3	○*4
Event execution type (with I45 interrupt)		○	—	—	○*2	○	○*3	○*4

\*2 This change does not affect refresh operations.

\*3 The CPU module operates with the parameters before the change.

\*4 The CPU module operates with the parameters after the change.

## Data that can be set

The following describes data that can be refreshed.

(1) Maximum number of settings

Up to 32 settings can be set in each CPU module with each refresh.

(2) Number of data points

The number of data points can be set within the number of points assigned to the fixed scan communication area (within send range of the area) in increments of two points. Duplications of devices among multiple setting numbers are not allowed.

(3) Data that can be specified

Devices other than local devices can be specified. However, when "Use File Register of Each Program" is enabled, file registers cannot be specified. Otherwise, refresh operations are not performed with file registers of each program depending on the operating status of a CPU module.

Type	Device
Bit device <sup>*1</sup>	X, Y, M, L, B
Word device	D, W, R, ZR, RD

<sup>\*1</sup> Specify bit devices only in increments of 16 points (one word).

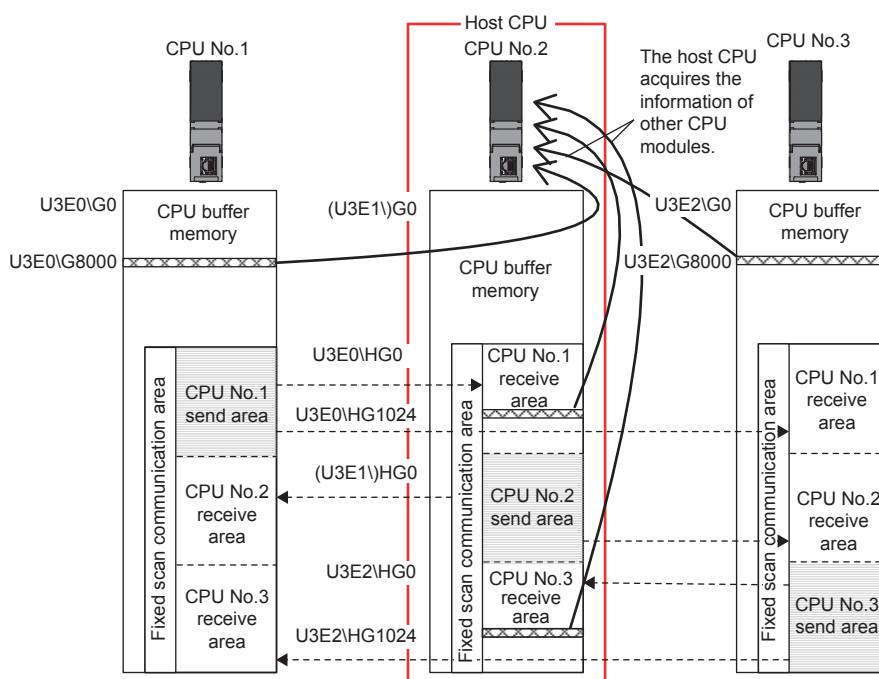
## 6.4.3 Communication through direct access

This method uses programs for data communications with other CPU modules. The following table lists data communications using the direct access method.

Communication method	Description	Instruction to be used
Data communications using CPU buffer memory	Data is transferred among CPU modules using desired areas on the CPU buffer memory.	<ul style="list-style-type: none"> <li>• FROM/TO instructions</li> <li>• DFROM/DTO instructions</li> <li>• Instruction using the CPU buffer memory access device (U3En\G□)</li> </ul>
Data communications using fixed scan communication areas	Data is transferred among CPU modules using fixed scan communication areas.	Instruction using the CPU buffer memory access device (U3En\HG□)
Data communications using dedicated instructions	Data is transferred among CPU modules and/or control directions are sent to other CPU modules using dedicated instructions.	DDWR/DDRD instructions and others

### Specification method using CPU buffer memory access device

To access CPU buffer memory, specify "U3En\Gn". To access fixed scan communication areas, specify "U3En\HGn".



- (1) Obtaining information stored in CPU buffer memory  
Access the CPU buffer memory.
- (2) Obtaining information stored in the fixed scan communication area of another CPU module  
Access the fixed scan communication area of the host CPU module. Obtain data sent to the area of the CPU module at a fixed scan communication cycle.



## Data communications using CPU buffer memory and fixed scan communication areas

The following describes the data communications using CPU buffer memory and fixed scan communication areas.

### (1) Areas used for data communications

The following areas can be used for data communications.

Area	Description
CPU buffer memory	All areas (excluding refresh area) in the CPU buffer memory can be used. The start address of usable areas in each CPU module varies depending on the refresh settings. The end address of the areas is fixed by the model of the CPU module used.
Fixed scan communication area	All areas (excluding refresh area) in the fixed scan communication area can be used. The start address of usable areas in each CPU module is HG0, and the end address varies depending on the refresh settings.

### (2) Instructions used for data communications

Data communications with each CPU module are performed using the following read/write instructions to each area.

- Write instructions: Instructions using the CPU buffer memory access device<sup>\*1</sup>, TO/DTO instructions
- Read instructions: Instructions using the CPU buffer memory access device<sup>\*1</sup>, FROM/DFROM instructions

<sup>\*1</sup> Specify U3En\G□ when accessing the CPU buffer memory. Specify U3En\HG□ when accessing the fixed scan communication area.

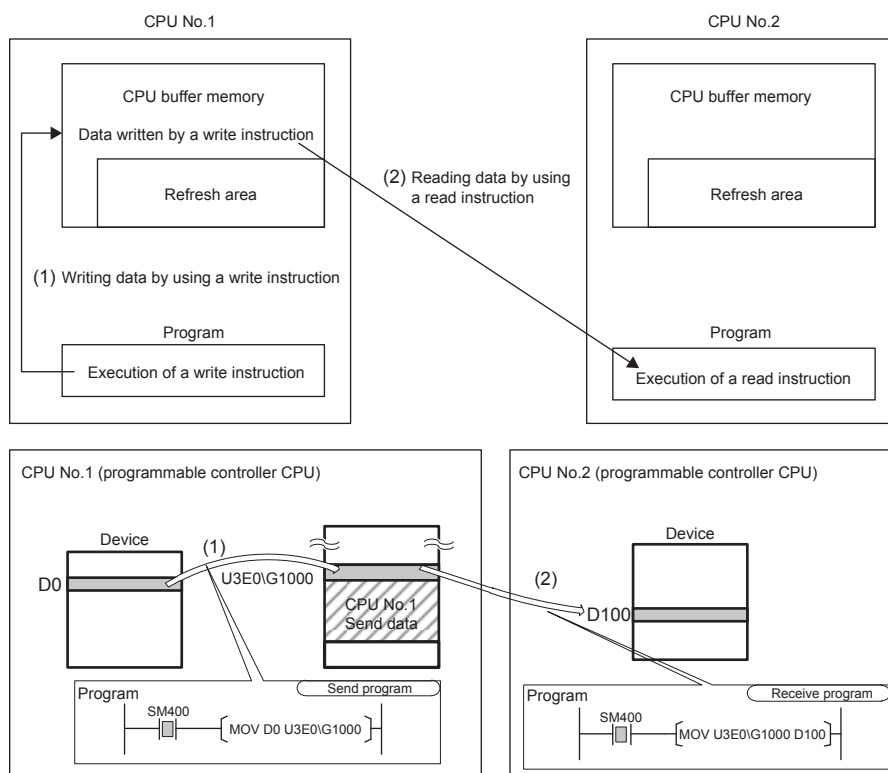
### (3) Data communication operation

- When areas in the CPU buffer memory are used

With a read instruction, other CPU modules read data that the host CPU module have written to areas in its CPU buffer memory with a write instruction. Unlike refresh operations, data that has been stored at execution of an instruction can be directly read.

#### Ex.

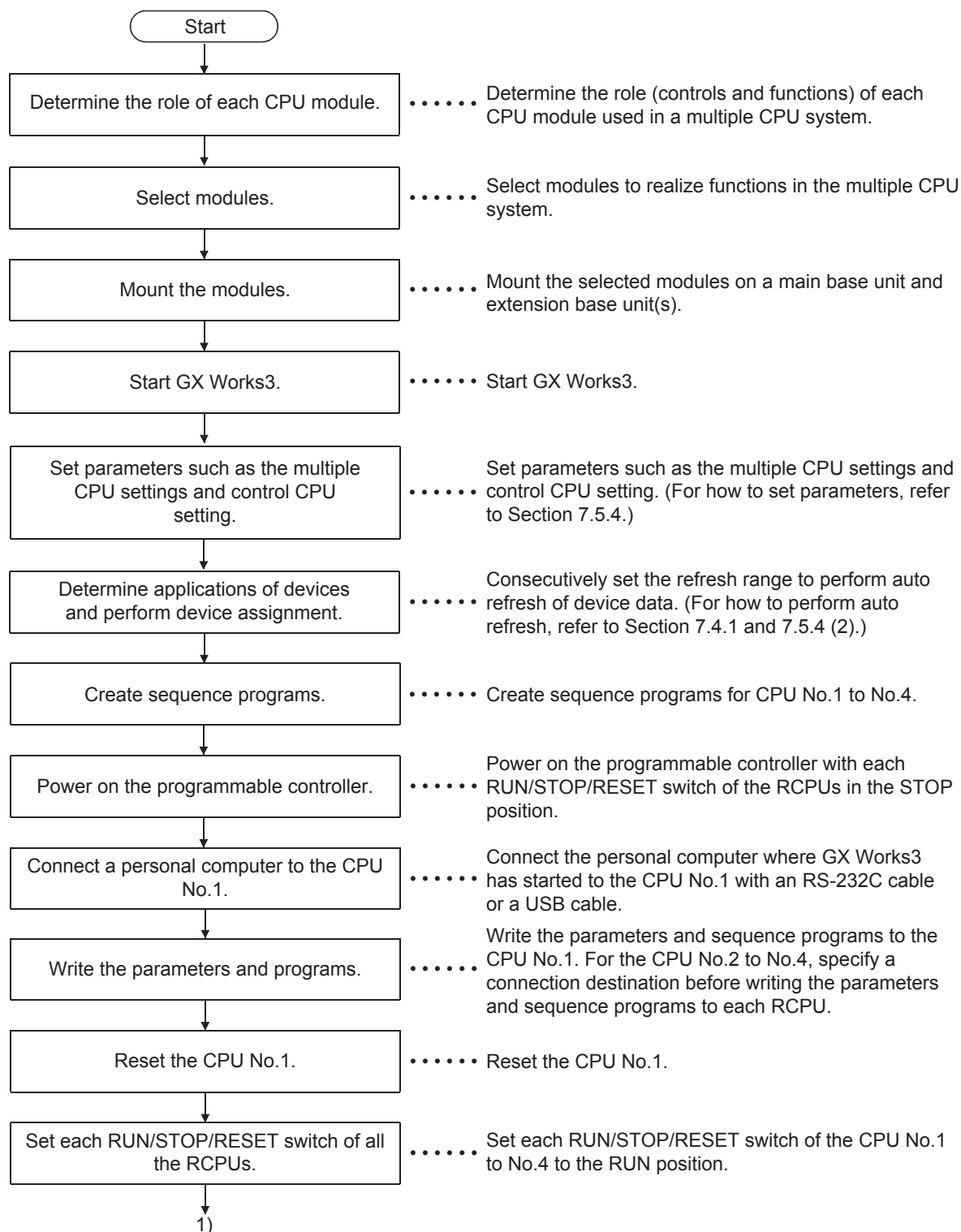
When the CPU No.2 uses a read instruction to read data that the CPU No.1 have written to its CPU buffer memory with write instructions

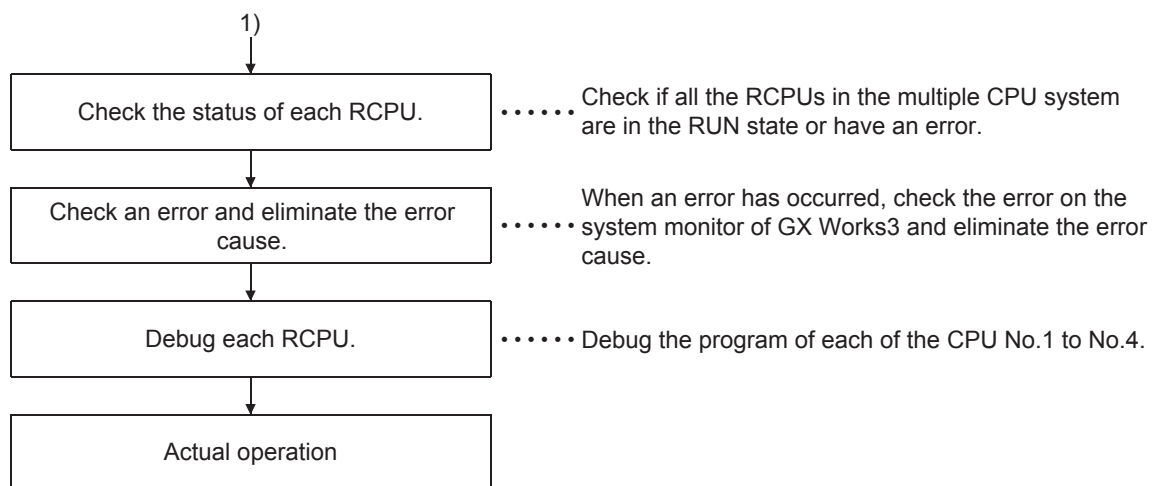


## 6.5 Starting Up Multiple CPU System

This section describes the procedure for starting up a multiple CPU system.

### 6.5.1 Procedure for starting up a multiple CPU system

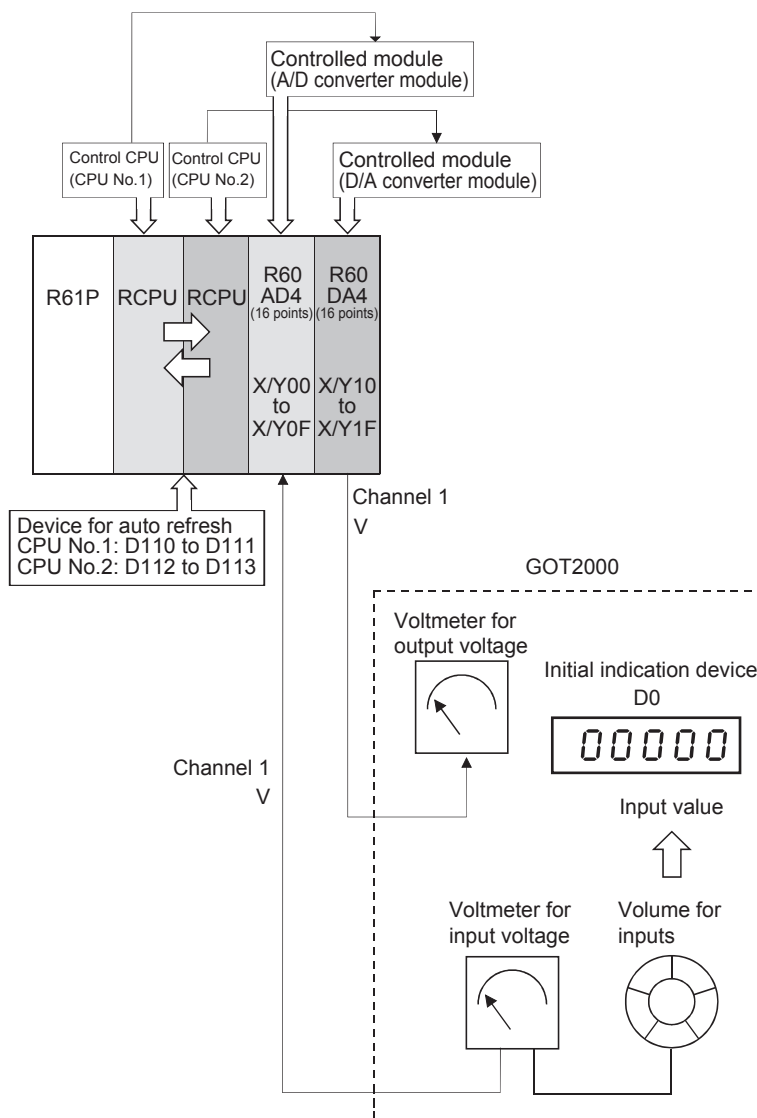




## 6.5.2 System configuration of the demonstration machine

This section describes the operation overview of the multiple CPU system through a simple example.

The system has two CPU modules (CPU No.1 and No.2), and each CPU module controls a different intelligent function module (A/D converter module or D/A converter module) and transfers data in each intelligent function module between the CPU modules.



### 6.5.3 Creating a program for the CPU No.1

Create the following program for checking the operations of the multiple CPU system.

Generally, in the multiple CPU system with auto refresh settings, when multiple data communications are performed among CPU modules, old data may be mixed with new data.

Interlocks are required as a measure against this problem.

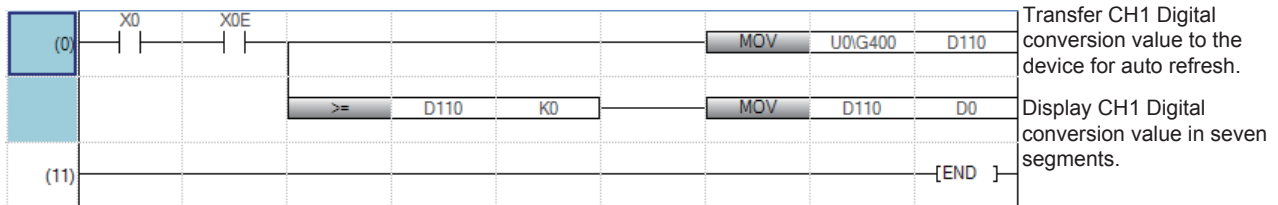
For this demonstration, no interlock is required since communication data is 1 word. Thus, create a simple program as shown below.

## Sequence program for the CPU No.1

This sequence program performs the sampling processing on an analog voltage input from CH1 of the R60AD4, and converts it into a digital value.

The converted digital value is stored in the device for auto refresh (D110).

Project name	EX8
Program name	MULTI1



X0: Module READY

X0E: A/D conversion completed flag

U0\G400 (Un\G400): CH1 Digital output value

## 6.5.4 Creating the module configuration

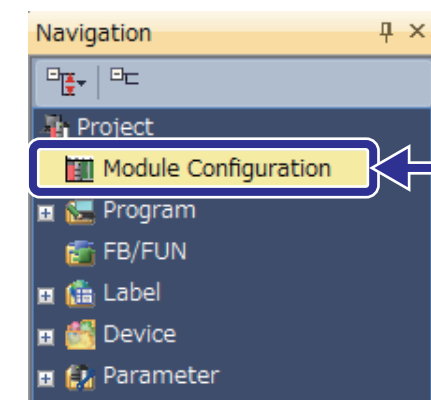
Arrange program elements (objects) in the "Module Configuration" window so that the configuration will be the same as that of the actual system.

The configuration that can be created in the "Module Configuration" window of GX Works3 is the one to be managed with the CPU module of the project.

### Arranging objects

In a multiple CPU system, set the CPU module that controls a mounted module as a control CPU.

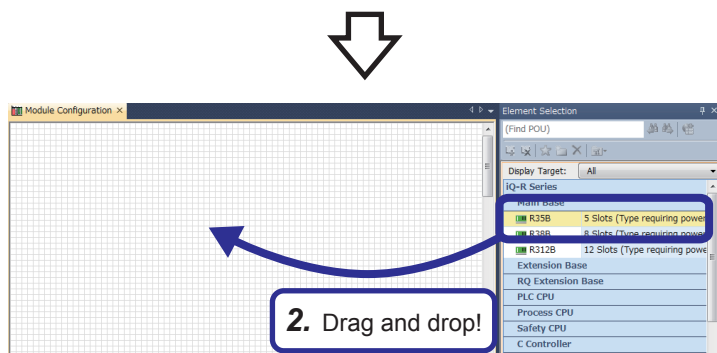
#### Operating procedure



1. Double-click!

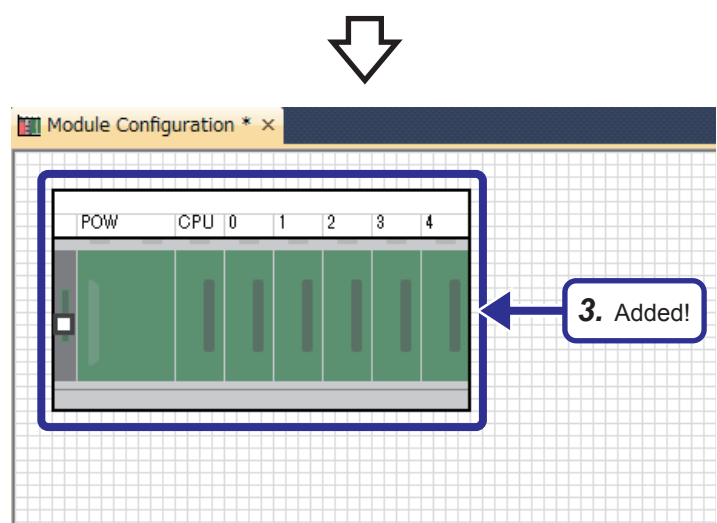
1. Double-click "Module Configuration" in the "Project" view.

When the dialog box about parameter information appears, click the [OK] button.



2. Drag and drop!

2. The "Module Configuration" window appears. Select "R35B" from "Main Base" in the "Element Selection" window, and drag and drop it to the "Module Configuration" window.

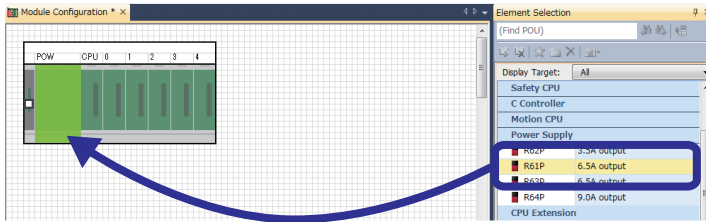


3. Added!

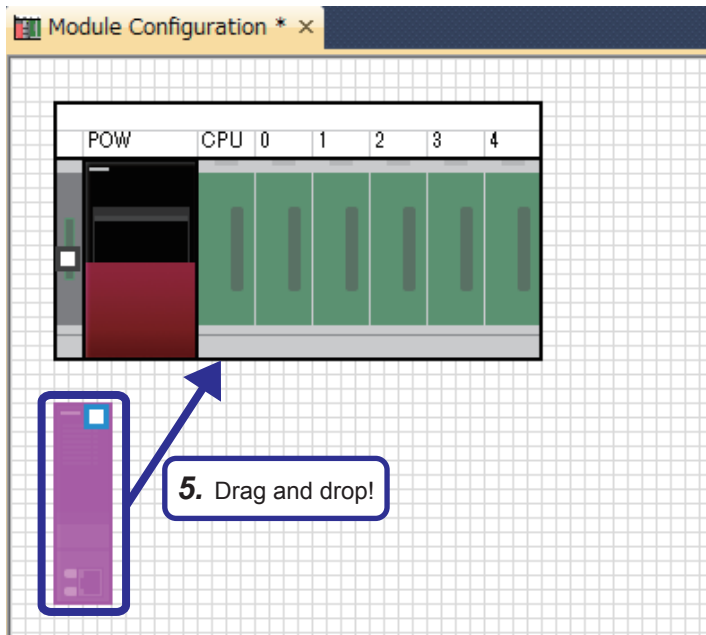
3. "R35B" is added on the "Module Configuration" window.

(To the next page)

(From the previous page)



4. Drag and drop!



(To the next page)

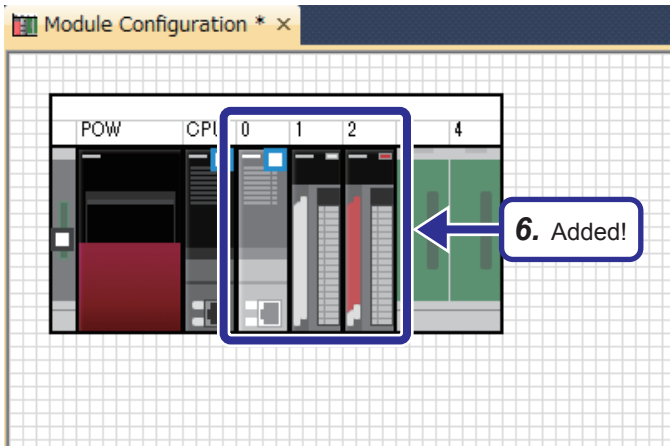
4. Select "R61P" from "Power Supply" in the "Element Selection" window, and drag and drop it to the power supply slot of the R35B on the "Module Configuration" window.

While the power supply module is being dragged and dropped, the slot where the power supply module can be arranged is highlighted.

5. Add the R08CPU that has already been arranged in the module configuration to the CPU slot of the R35B.

When the R08CPU has not been arranged in the module configuration, add the R08CPU from the "Element Selection" window in the same way as for the power supply module.

(From the previous page)



6. Add and set the following items with the same procedure for the power supply module. Set start XY numbers and control CPUs on the "Input the Configuration Detailed Information" window.

[Parameters to be set]

Module: PLC CPU

Module Name: R08CPU

Mounting Slot No.: 0

Module: Analog Input

Module Name: R60AD4

Mounting Slot No.: 1

Start XY: 0000

Control CPU: No.1

Module: Analog Output

Module Name: R60DA4

Mounting Slot No.: 2

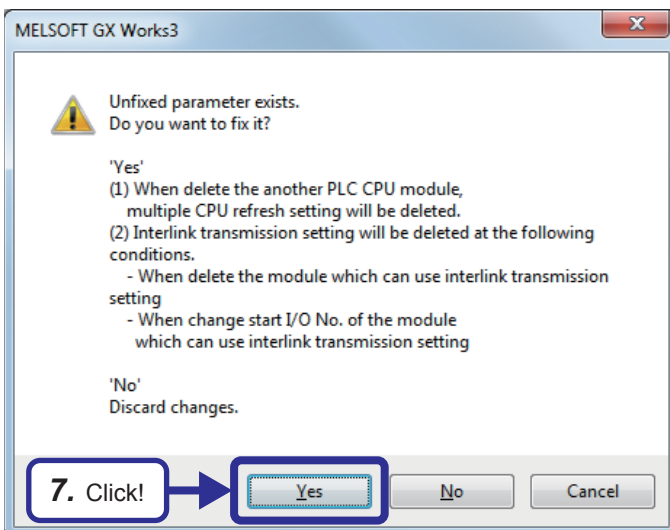
Start XY: 0010

Control CPU: No.2

Save the created and set data.

Project name: EX8

6



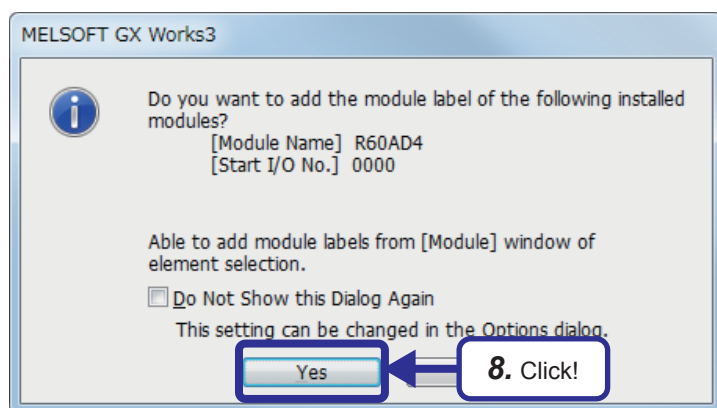
7. The dialog box shown on the left appears. Click the [Yes] button.



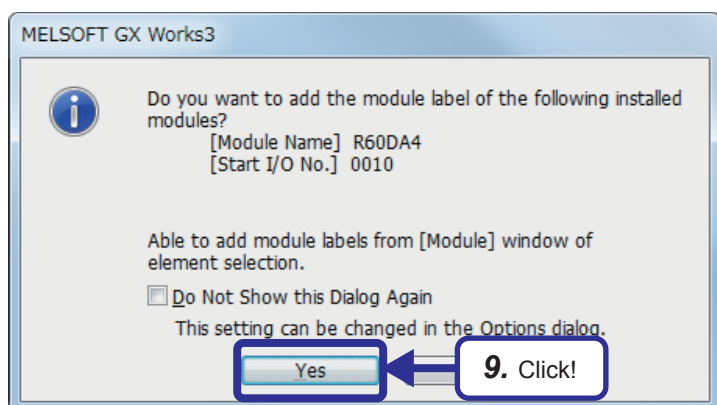
(To the next page)



(From the previous page)



- 8.** The dialog box shown on the left appears. Click the [Yes] button.



- 9.** The dialog box shown on the left appears. Click the [Yes] button.

## 6.5.5 Parameter setting of the CPU No.1

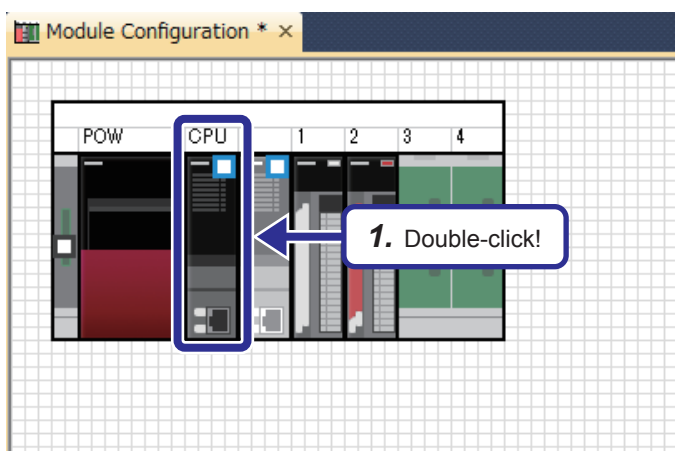
### Point

All the RCPUs and Motion CPUs used in the multiple CPU system need to have the same parameter settings (multiple CPU settings, PLC system settings (number of empty slots), and I/O assignment).

### Auto refresh setting

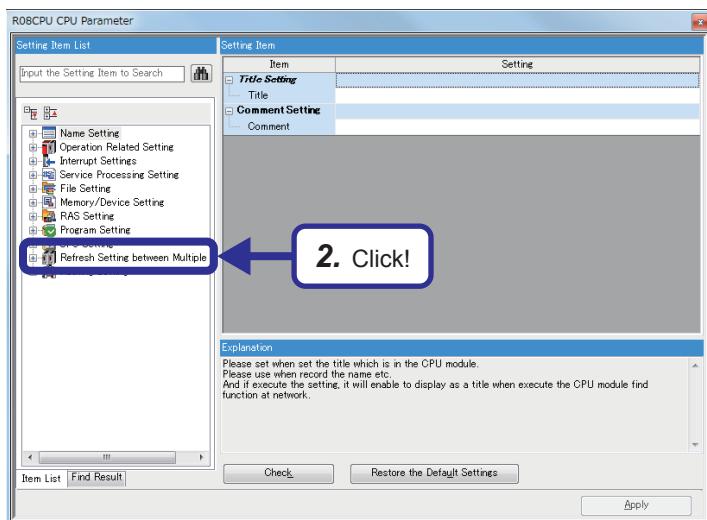
To perform auto refresh of device data, set the number of points of data sent by each CPU module and the device where each CPU module store the data. (When auto refresh is not performed, this setting is not required.)

### Operating procedure



1. Double-click the R08CPU module on the "Module Configuration" window.

6

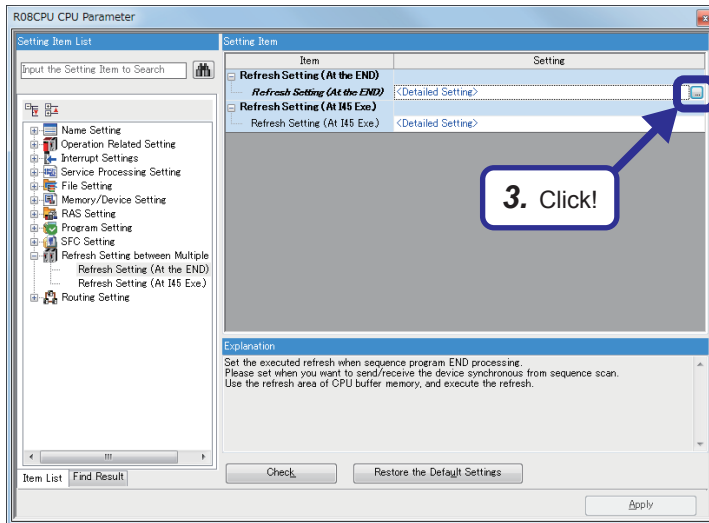



2. The "R08CPU CPU Parameter" dialog box appears. Click "Refresh Setting between Multiple CPU" in "Setting Item List".

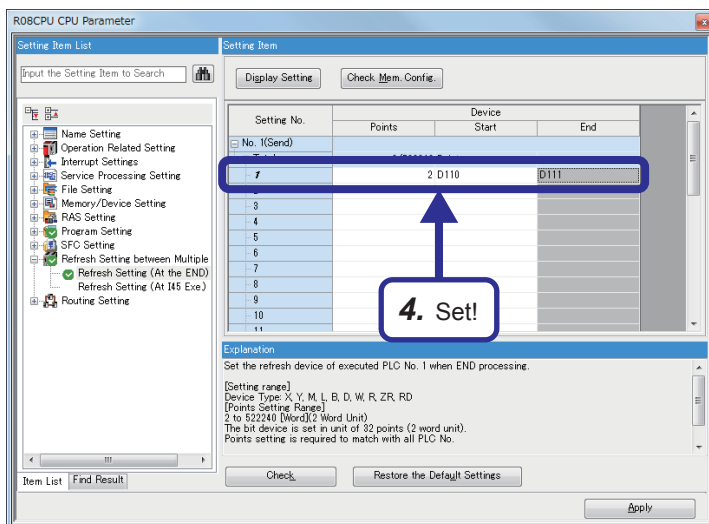


(To the next page)

(From the previous page)



3. Click "Detailed Setting" of "Refresh Setting (At the END)", and click the  button.



4. Set the device for storing the data to be sent from the CPU No.1 to another CPU module as follows.

[Parameters to be set]

Setting No.1

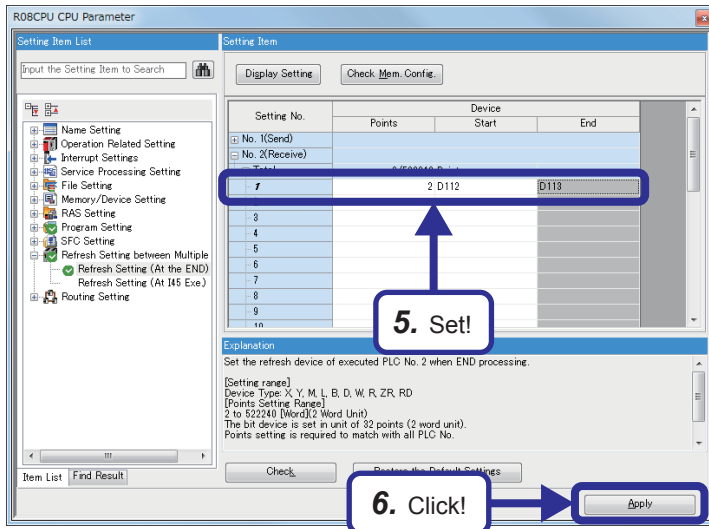
Points: 2

Start: D110



(To the next page)

(From the previous page)



5. Set the device for storing the data that CPU No.1 receives from the CPU No.2 as follows.

[Parameters to be set]

Setting No.1

Points: 2

Start: D112

6. Click the [Apply] button.



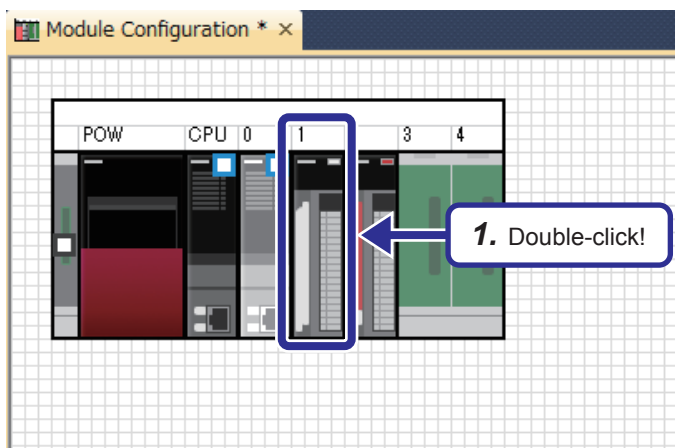
The following devices can be used for auto refresh.

- Devices that can be specified as send ranges: X, Y, M, L, B, D, W, R, ZR, SM, SD, SB, and SW
- Devices that can be specified as receive ranges: X, Y, M, L, B, D, W, R, and ZR

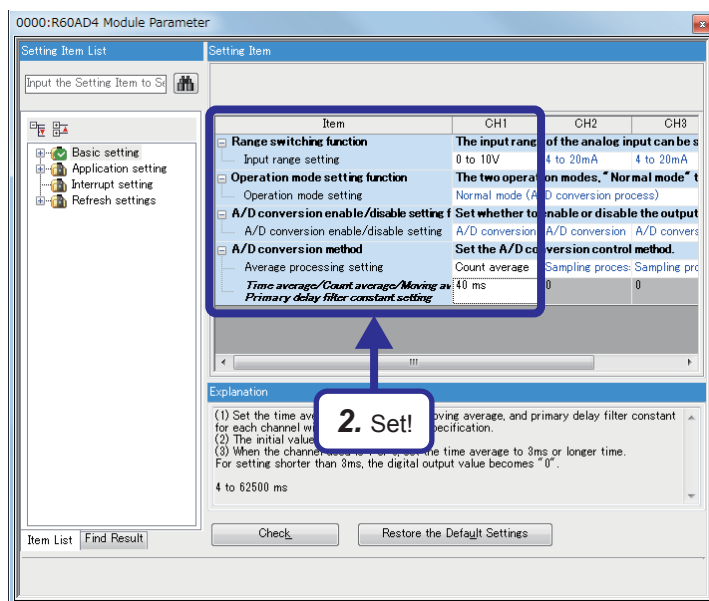
## Setting intelligent function module data

Set an analog module with the same procedure described in Chapter 5. This section describes the setting of the R60AD4 that is controlled by the CPU No.1.

### Operating procedure



1. Double-click the R60AD4 module on the "Module Configuration" window.



2. The "0000: R60AD4 Module Parameter" dialog box appears. Set items of CH1 as follows.

[Parameters to be set]

Input range setting: 0 to 10V

Average processing setting: Count average

Count average: 40

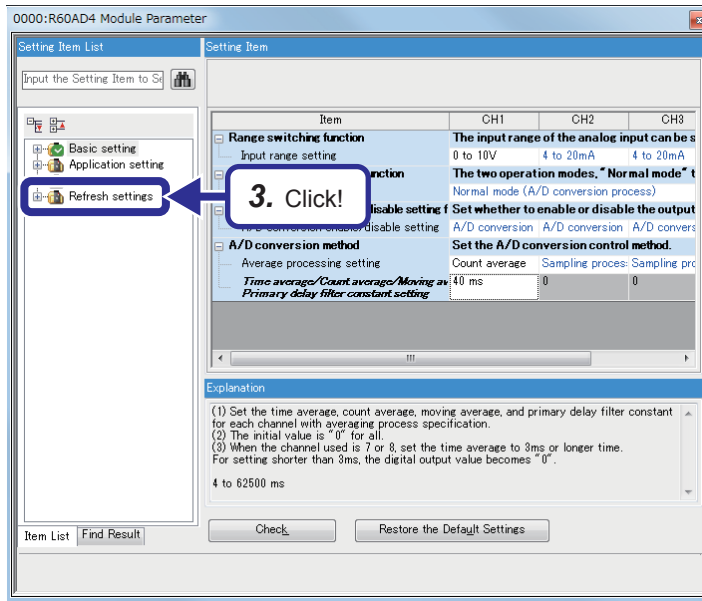


(To the next page)

(From the previous page)



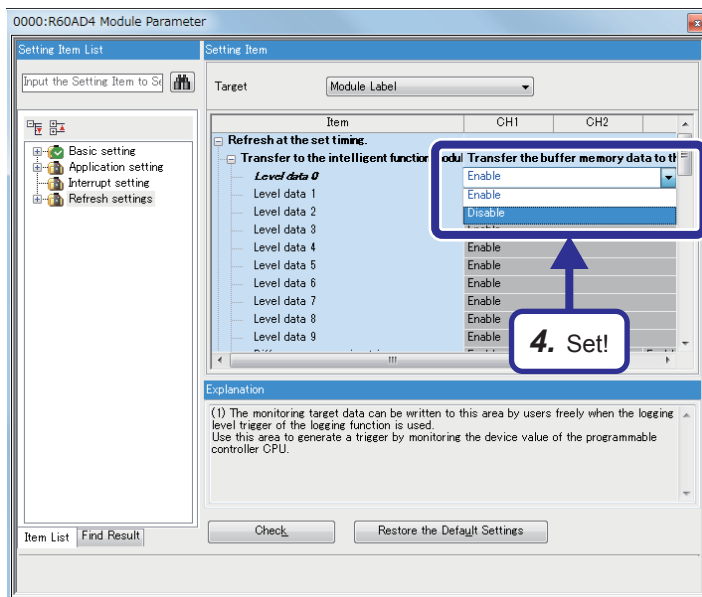
3. Click "Refresh settings" in "Setting Item List".



6



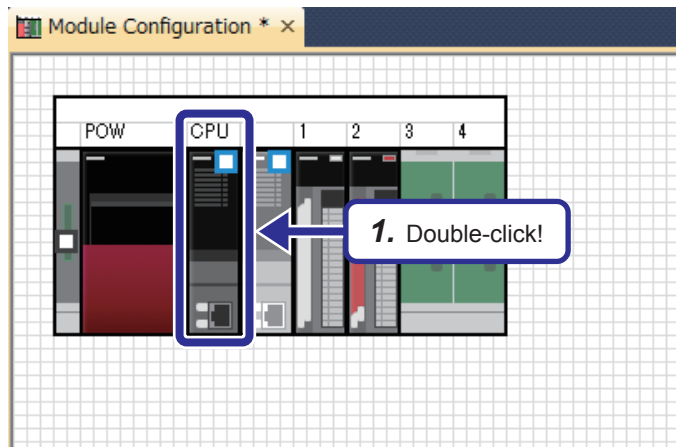
4. Set the refresh settings to "Disable".



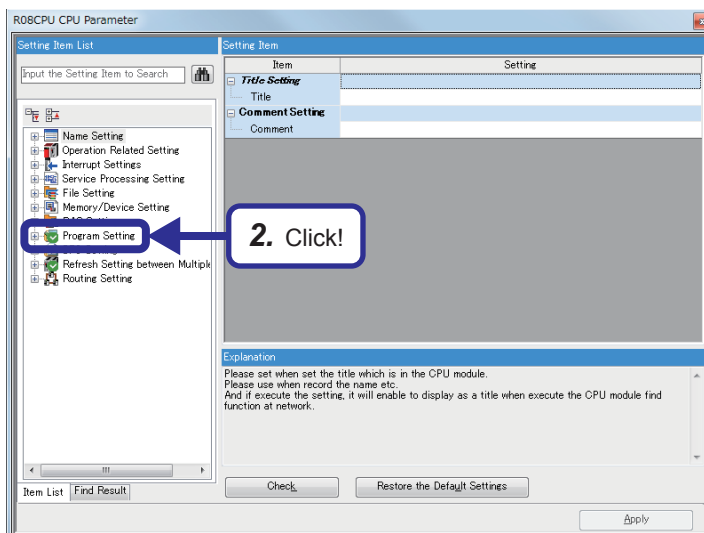
## Program setting

When multiple programs exist for a single CPU module, set an execution program in the program setting as a preventive measure against a CPU parameter error, although this is not always necessary.

### Operating procedure



1. Double-click the R08CPU module on the "Module Configuration" window.

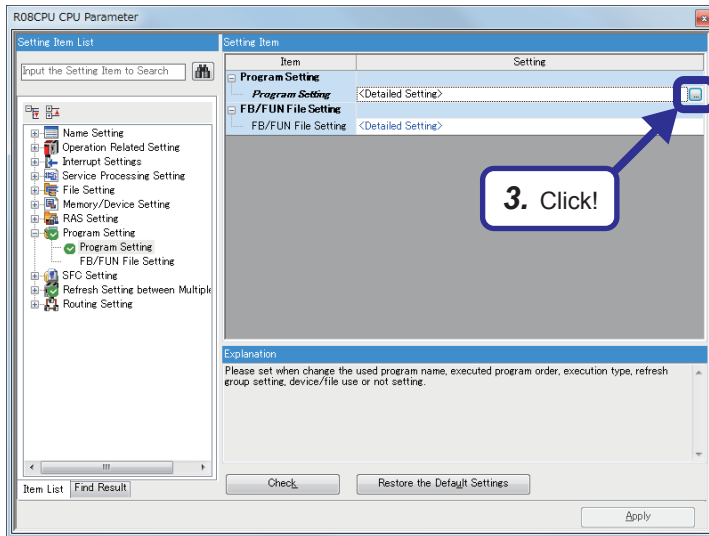



2. The "R08CPU CPU Parameter" dialog box appears. Click "Program Setting" in "Setting Item List".

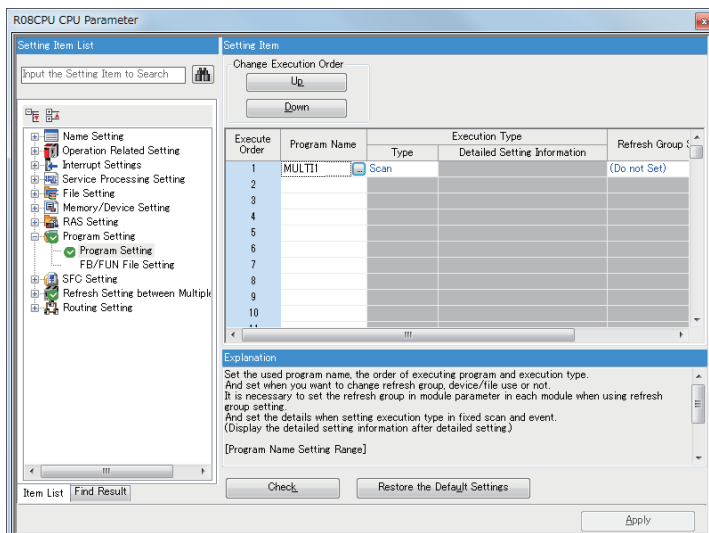


(To the next page)

(From the previous page)



3. Click "Detailed Setting" of "Program Setting", and click the  button.



4. Check that the type of the program "MULTI1" is the scan execution type.

Save the created and set programs and parameters.

Project name	EX8
Program name	MULTI1



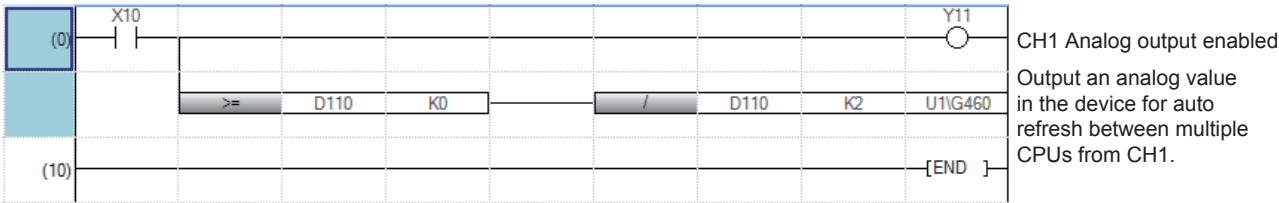
# 6.5.6     Creating a program for the CPU No.2

Create a program for the CPU No.2 in the same way for the CPU No.1.  
For this demonstration, an interlock among CPU modules is not required since communication data is 1 word. Thus, create a simple program as shown below.  
In this section, another project is created for the CPU No.2.

## Sequence program for the CPU No.2

This program divides the digital conversion value stored in the device for auto refresh (D110) of the RCP (CPU No.1) in half and converts the value into an analog signal from the CH1 of the R60DA4.

Project name	EX9
Program name	MULTI2



X10 (X0): Module READY  
U1\G460 (Un\G460): CH1 Digital value  
Y11 (Y1): CH1 Output enable/disable flag

## 6.5.7 Parameter setting of the CPU No.2

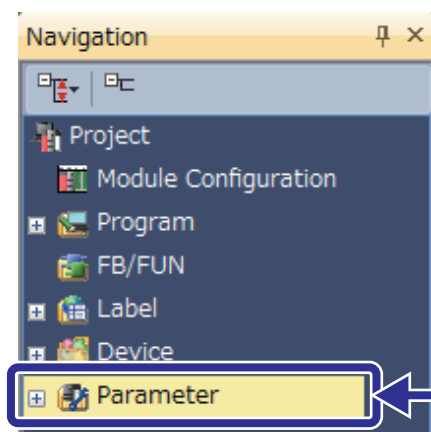
Among the parameter items set for the CPU No.1, some items related to the multiple CPU system configuration can be used for the CPU No.2 with "System Parameter Diversion" of GX Works3.

### Point

All the RCPUs and Motion CPUs used in the multiple CPU system need to have the same parameter settings (multiple CPU settings, PLC system settings (number of empty slots), and I/O assignment).

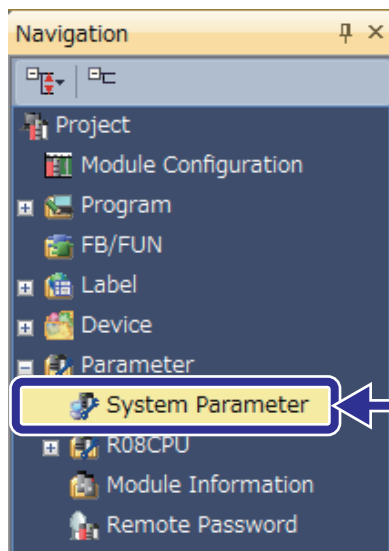
## Diverting parameters

### Operating procedure



1. Double-click "Parameter" in the "Project" view.

6

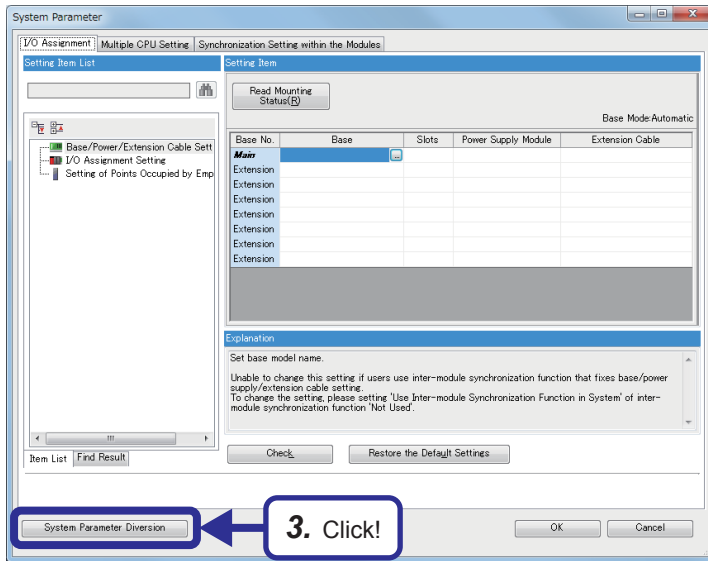


2. Double-click "System Parameter".

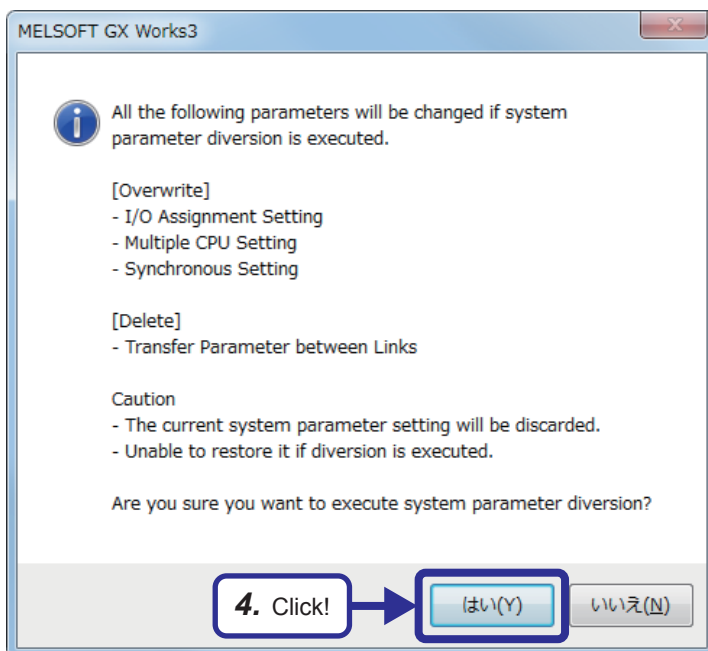


(To the next page)

(From the previous page)



3. The "System Parameter" dialog box appears. Click the [System Parameter Diversion] button.

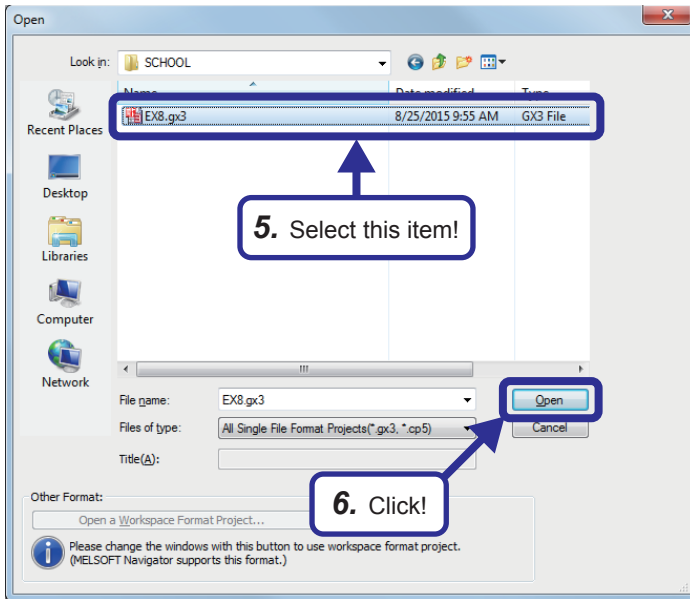


4. The confirmation dialog box appears. Click the [Yes] button.



(To the next page)

(From the previous page)

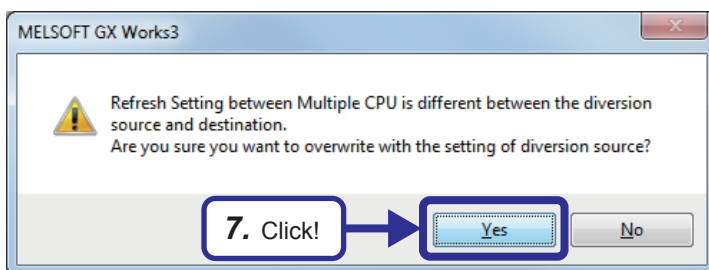


5. The "Open" dialog box appears. Select the project used.

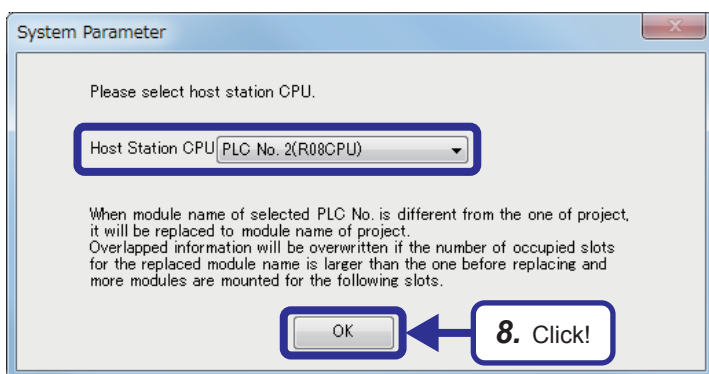
Project name: EX8

6. Click the [Open] button.

6



7. The confirmation dialog box appears. Click the [Yes] button.

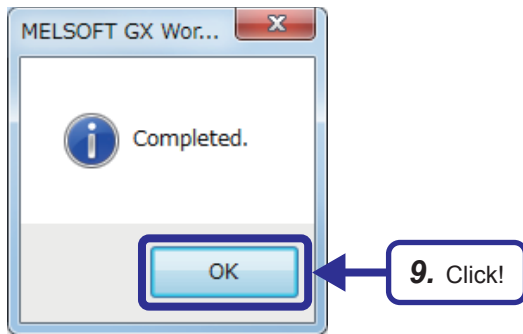


8. The "System Parameter" dialog box appears. Select "PLC No. 2(R08CPU)" for "Host Station CPU" and click the [OK] button.



(To the next page)

(From the previous page)

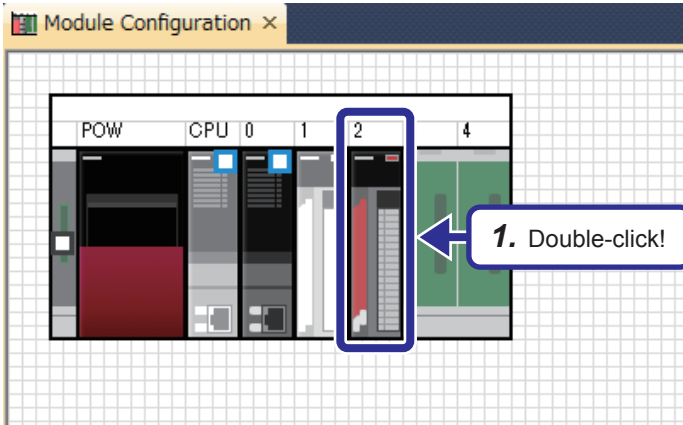


9. When the diversion is completed, the dialog box shown on the left appears. Click the [OK] button. (The dialog box for adding module labels appears. Click the [OK] button.)

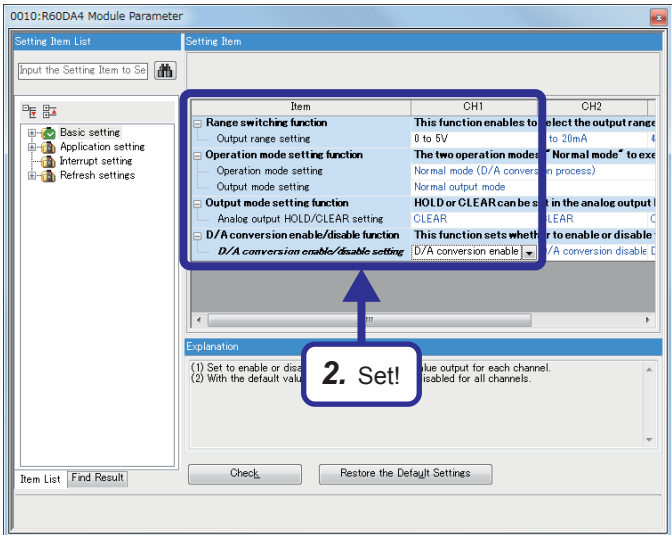
Setting intelligent function module data

Set an analog module with the same procedure described in Chapter 5. This section describes the setting of the R60DA4 that is controlled by the CPU No.2.

Operating procedure



- 1. Double-click the R60DA4 module on the "Module Configuration" window.



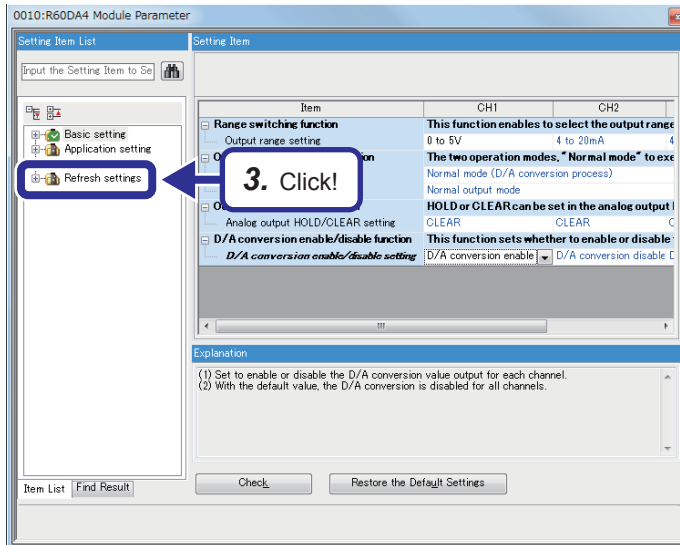
- 2. The "0010: R60DA4 Module Parameter" dialog box appears. Set items of CH1 as follows.

[Parameters to be set]  
"Output range setting": 0 to 5V  
"D/A conversion enable/disable setting": "D/A conversion enable"

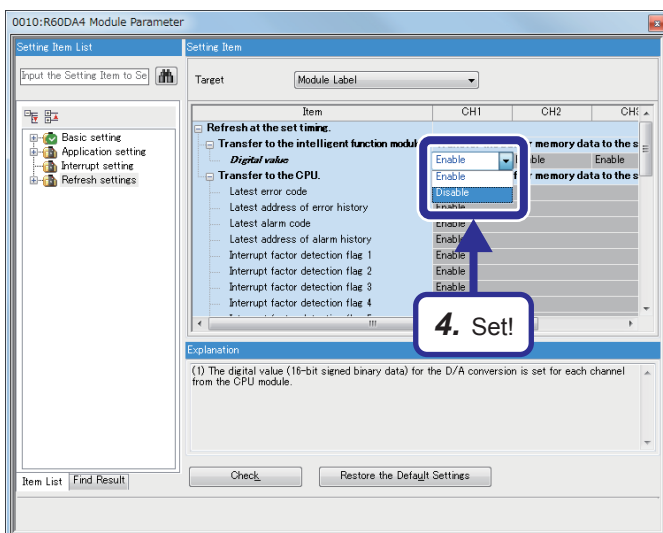


(To the next page)

(From the previous page)



3. Click "Refresh settings" in "Setting Item List".

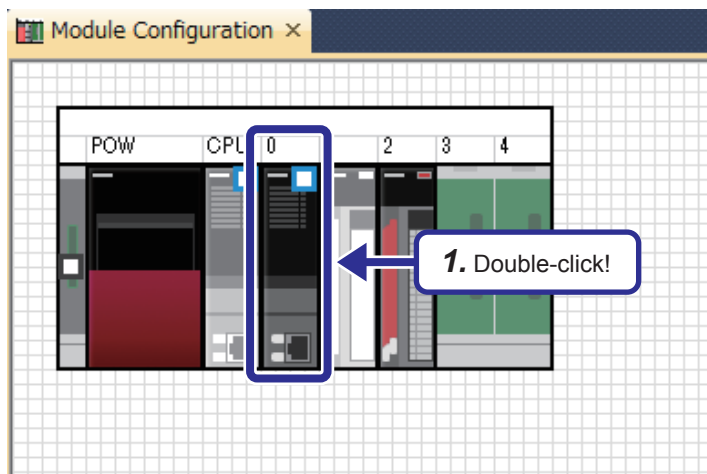


4. Set the refresh settings to "Disable".

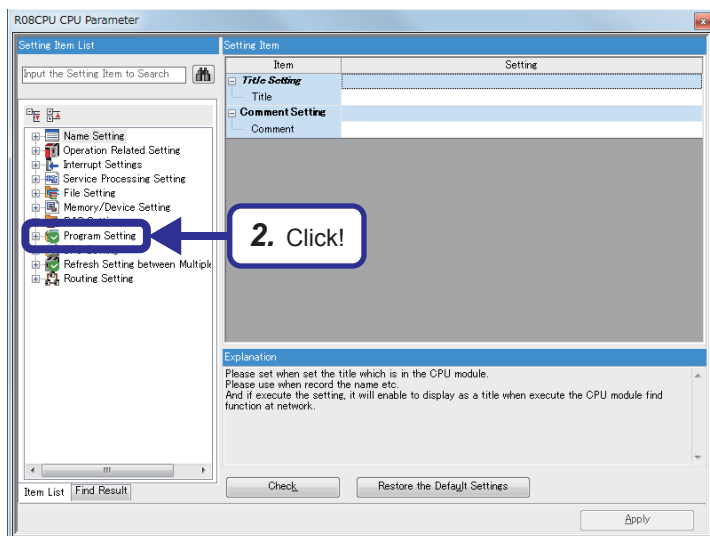
## Program setting

When multiple programs exist for a single CPU module, set an execution program in the program setting as a preventive measure against a CPU parameter error, although this is not always necessary.

### Operating procedure



1. Double-click the R08CPU module on the "Module Configuration" window.



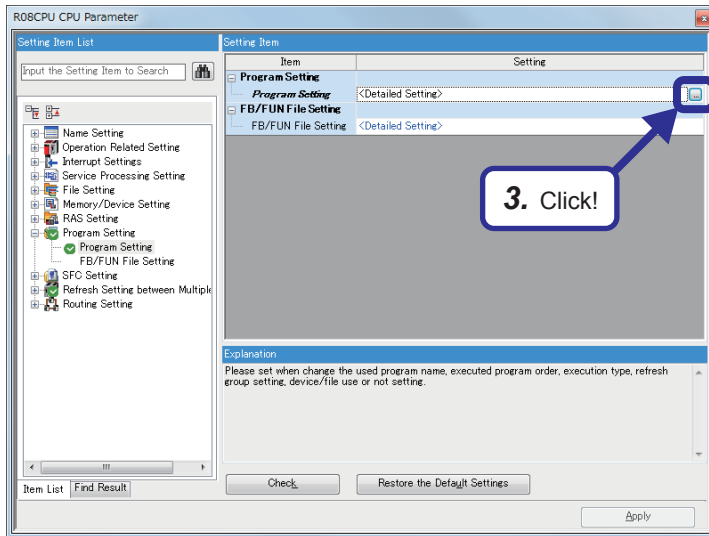
2. The "R08CPU CPU Parameter" dialog box appears. Click "Program Setting" in "Setting Item List".




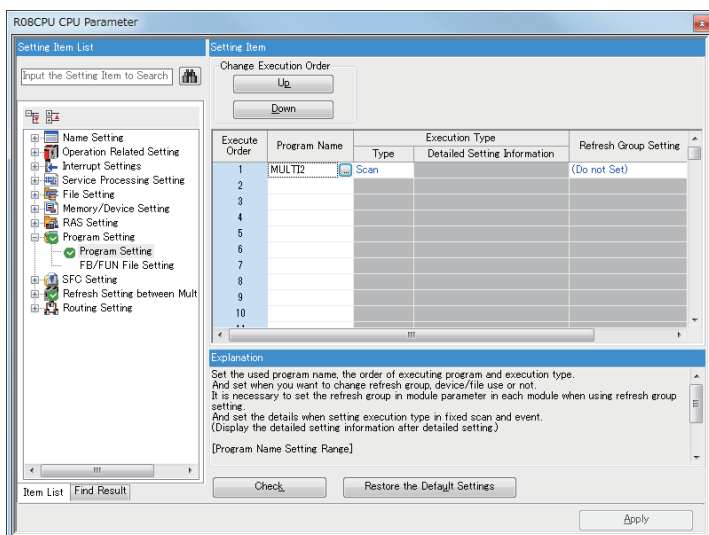
(To the next page)



(From the previous page)



3. Click "Detailed Setting" of "Program Setting", and click the  button.



4. Check that the type of the program "MULTI2" is the scan execution type.

Save the created and set programs and parameters.

Project name	EX9
Program name	MULTI2

## 6.5.8 Writing data to the CPU modules

Write the created sequence programs and parameter settings to each RCP.

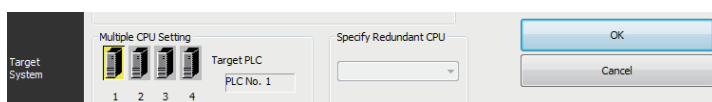
Set each RUN/STOP/RESET switch of the CPU modules (CPU No.1 and No.2) to the STOP position.

### Writing data to the CPU No.1

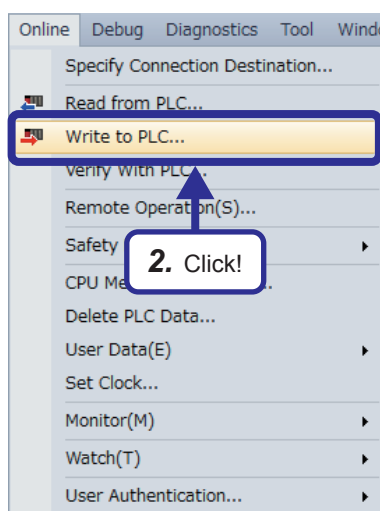
Write the created program to the CPU No.1 using GX Works3.

Project name: EX8, program name: MULTI1

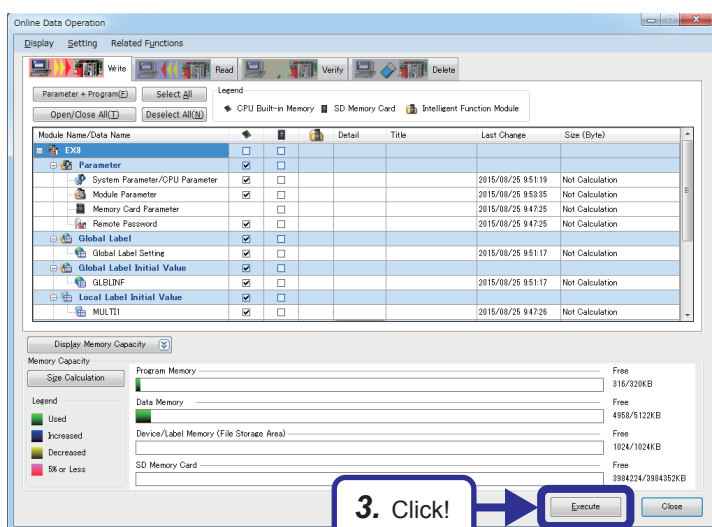
#### Operating procedure



1. Check that "PLC No. 1" has been set for "Target PLC" in the "Specify Connection Destination" window.



2. Click [Online] → [Write to PLC] from the menu.

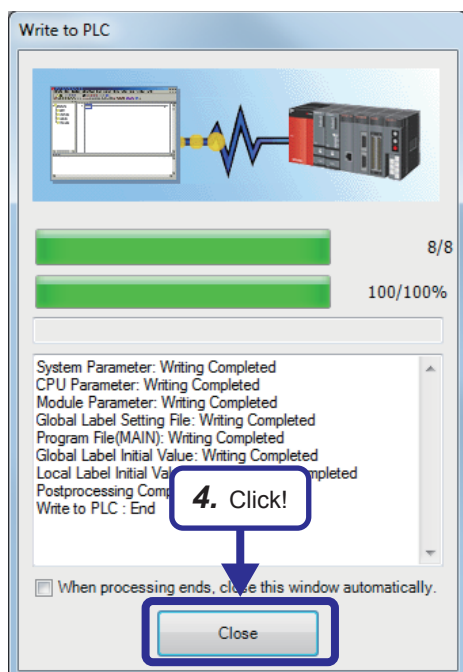


3. The "Online Data Operation" dialog box appears. Check the items to be written to the CPU module and click the [Execute] button.



(To the next page)

(From the previous page)



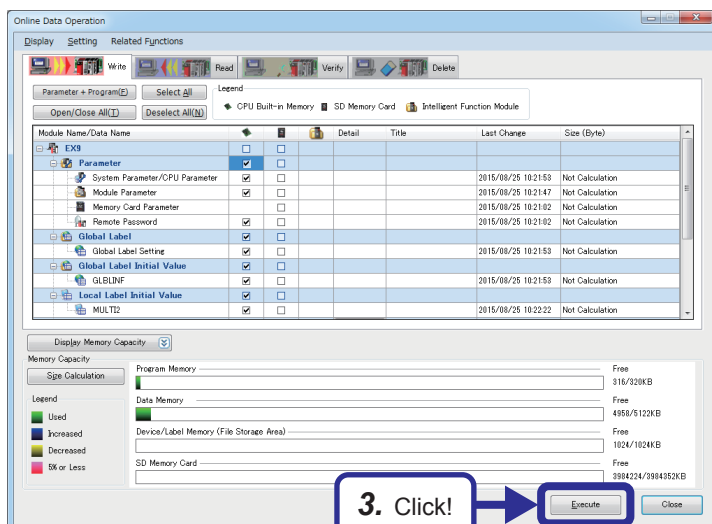
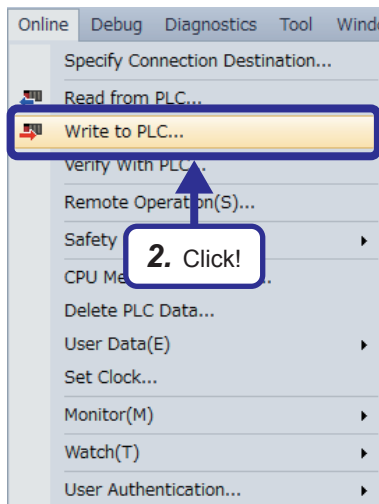
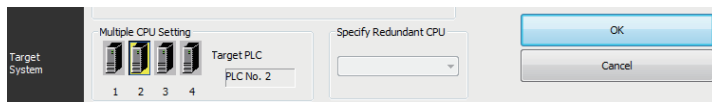
4. When data is written to the CPU No.1, the dialog box shown on the left appears. Click the [Close] button.

## Writing data to the CPU No.2

Write the created program to the CPU No.2 using GX Works3.

Project name: EX9, program name: MULTI2

### Operating procedure



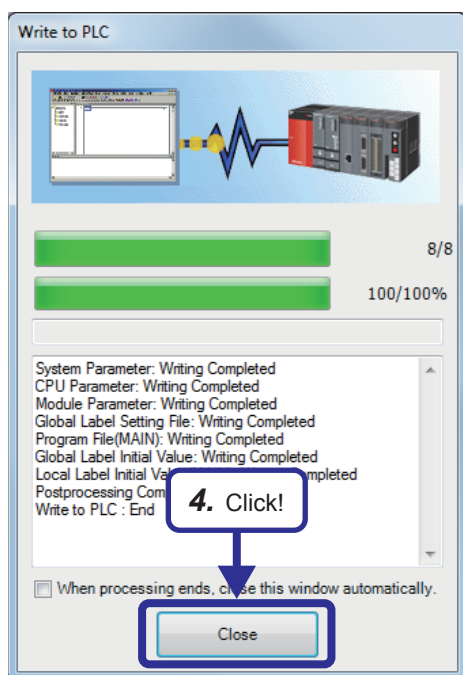
(To the next page)

1. Check that "PLC No. 2" has been set for "Target PLC" in the "Specify Connection Destination Connection" window.

2. Click [Online] → [Write to PLC] from the menu.

3. The "Online Data Operation" dialog box appears. Check the items to be written to the CPU module and click the [Execute] button.

(From the previous page)



4. When data is written to the CPU No.2, the dialog box shown on the left appears. Click the [Close] button.

## 6.5.9 Operation check

Check the operation with the following procedure.

### Resetting the multiple CPU modules

- 1) Reset the CPU No.1.
- 2) Set each RUN/STOP/RESET switch of the CPU No.1 and No.2 to the RUN position.

### Operation check

- 1) Change the input voltage of the A/D converter module with the volume of the demonstration machine.  
The digital conversion value is displayed in the initial indication device (D0) in seven segments.
- 2) The D/A OUTPUT voltmeter on the screen 2 displays the output voltage of the D/A converter module.  
The displayed value is a quarter of the value that the A/D INPUT voltmeter shows.  
It is because the A/D input range is 0 to 10V and the D/A output range is 0 to 5V, and the CPU module divides a digital conversion value in half.



# 7 NEW FUNCTIONS OF MELSEC iQ-R/GX Works3

## 7.1 Features of MELSEC iQ-R

### Productivity

#### ■Newly-developed high-speed system bus that greatly shortens takt time

The newly-developed high-speed system bus (40 times as fast as our conventional products) greatly speeds up the data communication among multiple CPU modules and the large-capacity data communication with network modules. This feature maximizes the performance and functions of the MELSEC iQ-R series.

#### ■Multiple CPU system to realize a high-accuracy motion control

The cycle of data exchange between a programmable controller CPU and a Motion CPU has been speeded up (approximately 4 times as fast as our conventional products), realizing a high-accuracy motion control.

#### ■Synchronization function to realize high-accuracy processing

The inter-module synchronization function operates intelligent function modules and I/O modules in synchronization with the program execution timing of a programmable controller CPU or a Motion CPU, realizing a high-accuracy control of systems and devices.

In addition, the CC-Link IE Field Network or SSCNET III/H synchronous communication synchronizes the operation timing of nodes on the network. This feature reduces variations caused by the network transmission delay time, allowing users to establish a stable system.

### Engineering

Users can reduce development costs by intuitively programming with GX Works3.

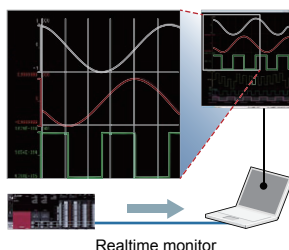
For details on GX Works3, refer to Page 7 - 5 Functions of GX Works3.

### Maintenance

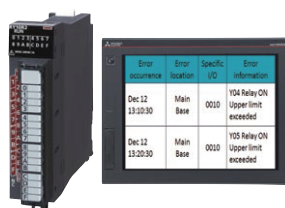
The MELSEC iQ-R products are equipped with the preventive maintenance to prevent troubles from occurring and various maintenance functions for quickly recovering the system at occurrence of troubles to shorten downtime, improve the productivity, and maintain the quality of manufactured products.

#### ■Collecting production information of production processes

(1) Users can monitor values in specified devices in real time at desired intervals or timing. (CPU module)

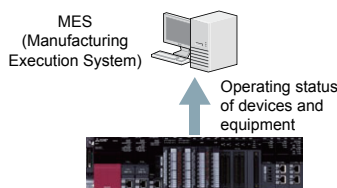


(2) External devices such as GOTs can be used to check lives of relays, facilitating planned replacement of modules. (output modules)





- (3) Because data can be directly written to the database in an upper system, users can collect data such as the operating status of devices and equipment for improvement activities before occurrence of troubles. (MES interface module)



## ■Operation/error information history to solve troubles quickly

Users can check and save the history of events such as writing of programs, occurrence of errors, and power-off in a list. This feature enables users to quickly detect troubles caused by operation mistakes.

No.	Occurrence Date	Event Type	Status	Event Code	Overview
00002	2014/06/19 16:23:13.140	System	U	00400	Power-on reset
00003	2014/06/19 14:26:53.627	System	U	02003	In-rail module
00004	2014/06/19 14:26:55.798	System	U	00400	Power-on reset
00005	2014/06/19 14:26:54.436	System	U	03000	Power shut-off
00006	2014/06/19 14:27:00.100	Operation	U	24000	Detection of new battery
00007	2014/06/19 14:27:00.117	Operation	U	24000	Detection of new battery

List of event history data

## Quality

### ■Improving the reliability of a production system

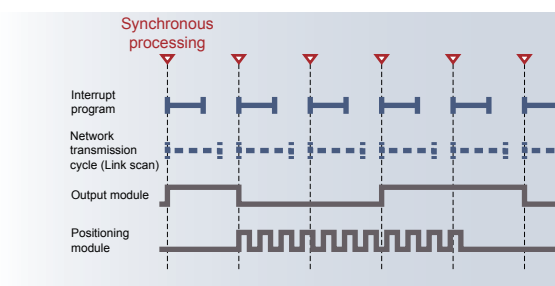
MELSEC iQ-R series products have passed our strict quality evaluation tests implemented in various industrial scenes, such as EMC (ElectroMagnetic Compatibility) tests, LSI tests, temperature tests, vibration tests, and HALT tests.

QR codes are used to manage the quality information at the time of manufacturing and to offer high-quality products to our customers.



### ■Improved quality of products to be manufactured

This function reduces variations of data communication (network transmission delay time) between a programmable controller and devices on a network, improving the quality of products to be manufactured.



## Connectivity

With SLMP (Seamless Message Protocol), users can perform seamless data communication from the production control level of an entire automation system to the device level such as sensors, without considering layers of the network.

### ■Seamless information linkage

With SLMP, users can access the production control system, programmable controllers, and other devices seamlessly in an identical manner without considering layers and boundaries of the networks. Users can easily monitor devices and collect data from anywhere.

### ■Simple connection to external devices with MELSOFT Library

With the predefined protocol support function of GX Works3, only selecting a protocol to be used and data to be sent or received enables the simple communication with external devices, such as vision sensors and temperature controllers. Users do not need to create programs for communication, reducing the man-hour for developing programs.

## Security

The MELSEC iQ-R series products are equipped with strong security functions such as the security key authentication to protect programs and an IP filter to prevent unauthorized accesses to control system.

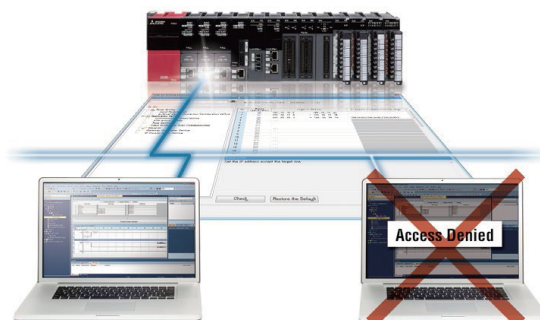
### ■Security authentication to protect project data

The security key authentication function locks programs so that they cannot be opened in the personal computer where no security key has been registered.



### ■IP filter function

The IP filter function registers IP addresses of devices that can access the CPU module to prevent accesses from devices other than the registered ones. This function reduces risks of programs being hacked by an outsider, unauthorized modifications, or others.



## Compatibility

Properties such as MELSEC-Q series programs used in the existing system and various modules can be utilized for the MELSEC iQ-R series.

### ■Utilizable program properties

MELSEC-Q series programs can be converted<sup>\*1</sup> into the ones for the MELSEC iQ-R series and utilized.

Stored program properties can be effectively used to reduce the man-hours for developing programs and to shorten the development period.

<sup>\*1</sup> Part of the programs cannot be converted. For details, refer to the GX Works3 Operating Manual.



### ■Utilizable modules

With dedicated extension base units, users can use the MELSEC-Q series modules in the MELSEC iQ-R series system. (For details on the Q series modules that can be used in the MELSEC iQ-R system, refer to the Module Configuration Manual.)

Users can reduce costs required for spare parts or others and introduce the high-performance MELSEC iQ-R series.



## 7.2 Differences Between the MELSEC-Q Series and the MELSEC iQ-R Series

For differences between the MELSEC-Q series and the MELSEC iQ-R series, refer to the TECHNICAL BULLETIN No. FA-A-0171.

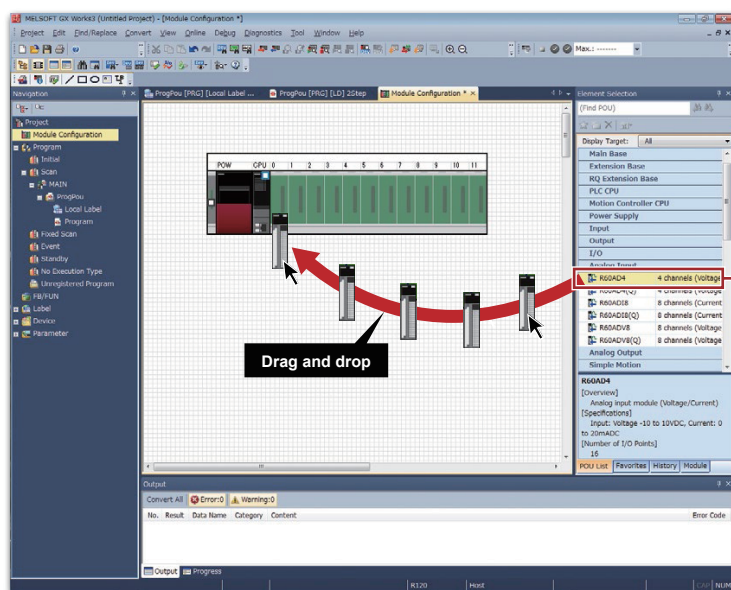
## 7.3 Functions of GX Works3

GX Works3, an engineering tool, has the functions that facilitate users to create projects (system configuration, programming) and perform maintenance (debugging, diagnostics, and management).

### System design

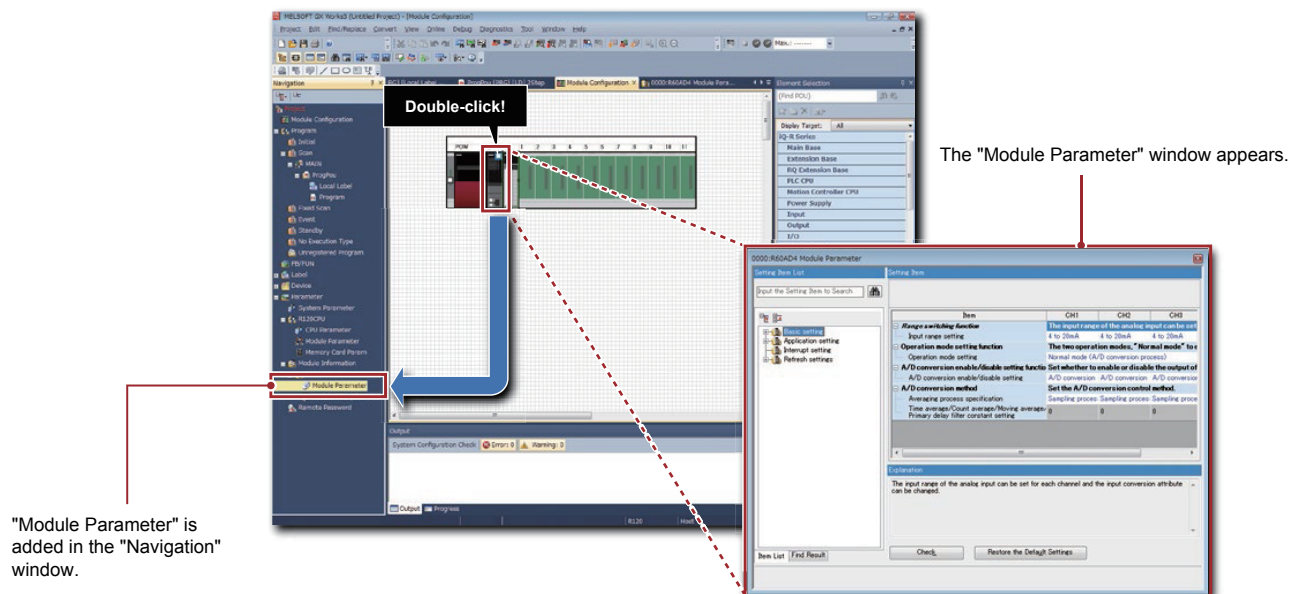
#### ■ Simple system design

Creation of a project starts from system design. GX Works3 helps users to easily design a system. Users can create a module configuration only by selecting program elements and dragging and dropping them into the "Module Configuration" window of GX Works3.



#### ■ Easy creation of module parameters

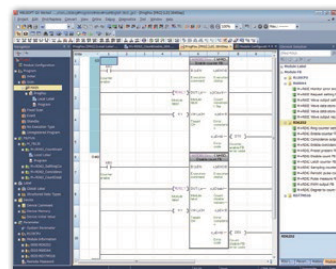
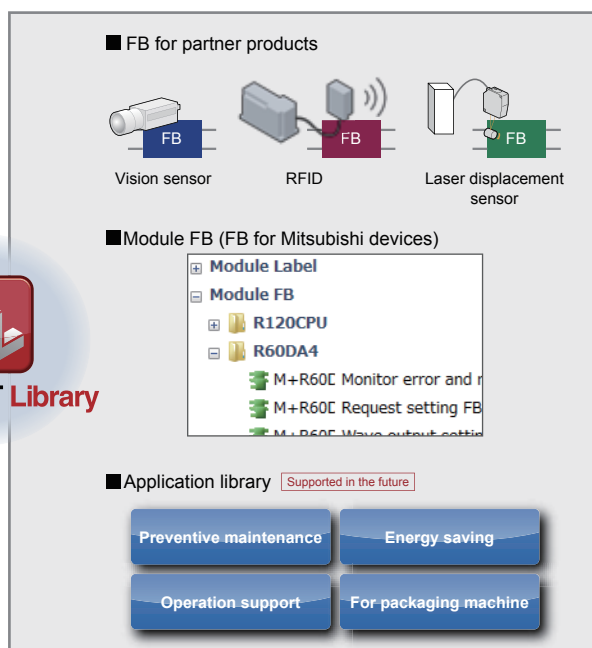
Module parameters can be automatically created in the creation of a module configuration. Users can create module parameters of a project only by double-clicking a module on the "Module Configuration" window. Related parameters are displayed as the work window and parameters can be set.



## Programming

### ■MELSOFT Library prepared to reduce man-hours

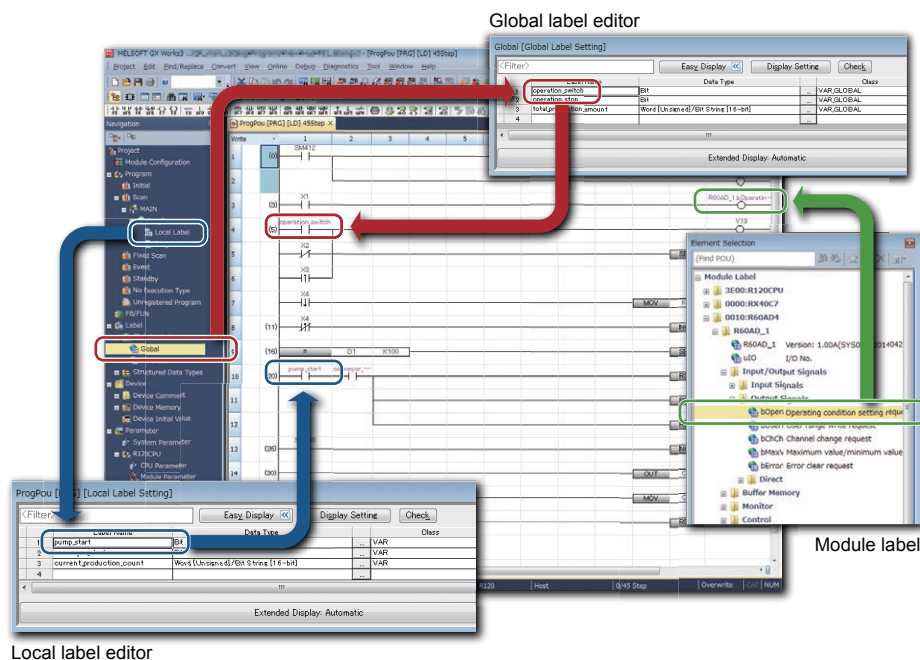
A variety of libraries (FBs for partner products/module FBs/application libraries or others) are available in MELSOFT Library. Using FBs reduces the man-hours for developing programs.



Easy programming - What users need to do is to select parts!

## Labels prepared to reduce loads

GX Works3 allows users to use global labels, local labels, and module labels. Global labels can be shared and used in multiple programs and other MELSOFT software applications. Local labels can be used in a program and FB where the labels are registered. Module labels have information about I/O signals and the buffer memory areas of each intelligent function module. Thus, users can create a program without considering I/O addresses and buffer memory addresses.





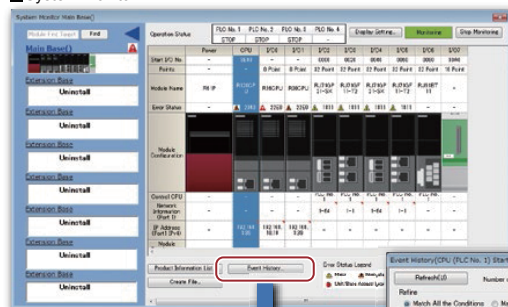
## Maintenance

### ■Diagnostic function

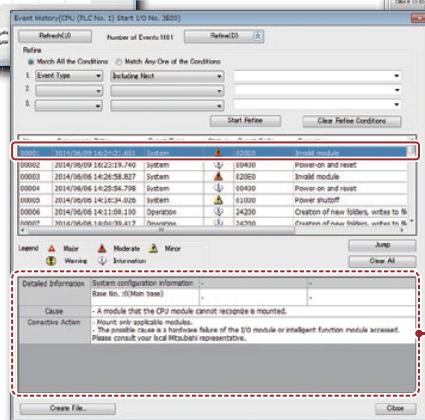
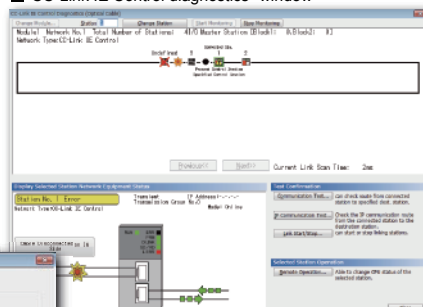
Users can easily identify faulty areas with the diagnostic function of GX Works3. Users can check a module configuration and error status in the system with the system monitor. Users can check errors that have occurred and operations performed in each module in chronological order with the event history display.

Because faulty areas on the network are graphically displayed in the various network diagnostics, downtime can be shortened.

■System Monitor



■"CC-Link IE Control diagnostics" window

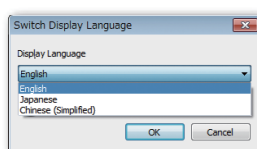
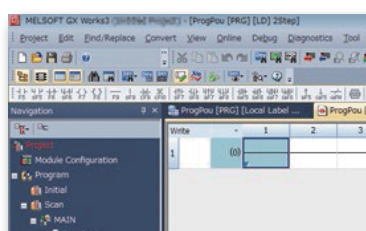


■Event History display

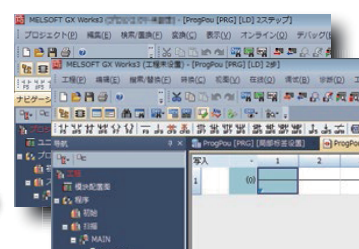
This area displays detailed information, causes, and actions for errors.

### ■Language switching

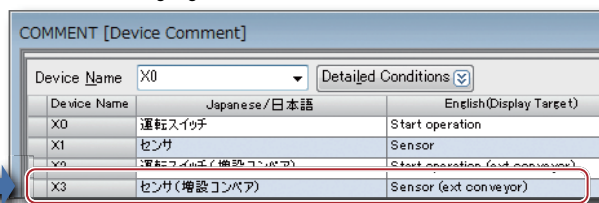
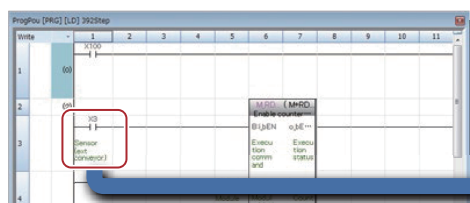
Users can switch the language of the menu display or others in GX Works3. Users can create comments in each language and easily switch the display. Thus, when foreign engineers perform maintenance, they can easily understand programs only by switching the language of comments to their native language, helping their operations.



The display language can be easily switched in one package.



The comment language can be switched.



# MEMO

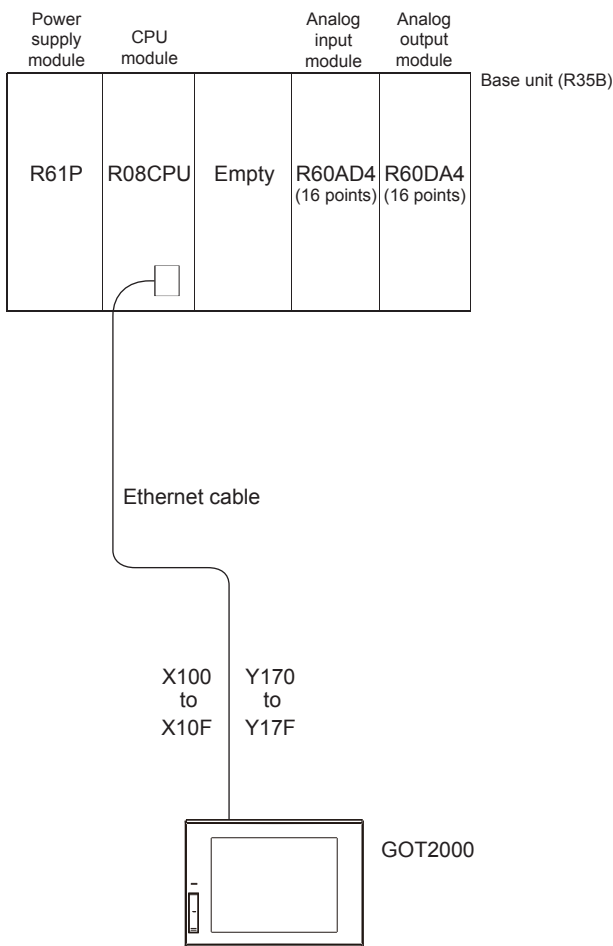
---

# APPENDICES

## Appendix 1 System Configuration of the Demonstration Machine

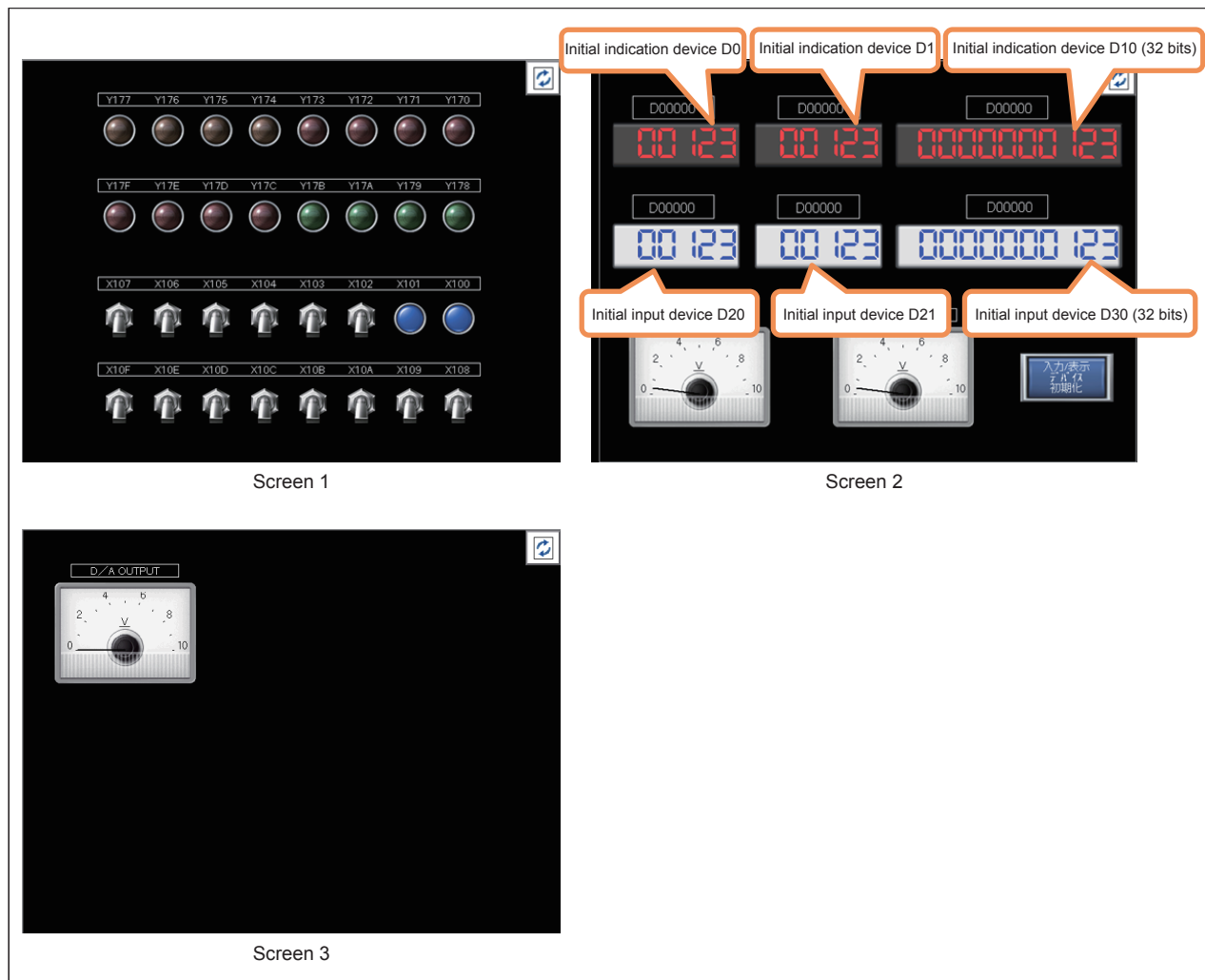
The following figure shows the system configuration for the main demonstrations in this textbook.  
For the demonstrations with a multiple CPU system, a CPU module will be added to the empty slot.

■System configuration of the demonstration machine






## ■GOT screen display



Upper section: The indication device can be changed.  
Lower section: Data is displayed.



Upper section: The input device can be changed.  
Lower section: The input data can be set and displayed.




























- Touching  switches the screen.
- The initial value is automatically set to a device number in the upper section (trigger action function).
- Touching the "Initialize Input/Indication Device" button also initializes the device number.



















# Appendix 2 Creating Ladders with GX Works3
























The following five methods of creating ladders are provided.

- Entering list expressions (mnemonic language) with a keyboard on the ladder editor
- Using tool buttons on the toolbar
- Using function keys
- Selecting menu items on the toolbar
- Directly entering devices (\* only I/O devices)

The following table describes how to input each item.


Example	List expression	Tool button	Function	Menu bar
	LD X1 → <input type="text" value="Enter"/>	Click  . → Enter "X1" in the device instruction entry field. → Click the [OK] button.	 → Enter "X1" in the device instruction entry field. → <input type="text" value="Enter"/>	[Edit] → [Ladder Symbol] → [Open Contact] → Enter "X1" in the device instruction entry field. → <input type="text" value="Enter"/>
	LDI X1 → <input type="text" value="Enter"/>	Click  . → Enter "X1" in the device instruction entry field. → Click the [OK] button.	 → Enter "X1" in the device instruction entry field. → <input type="text" value="Enter"/>	[Edit] → [Ladder Symbol] → [Close Contact] → Enter "X1" in the device instruction entry field. → <input type="text" value="Enter"/>
	OR X1 → <input type="text" value="Enter"/>	Click  . → Enter "X1" in the device instruction entry field. → Click the [OK] button.	 +  → Enter "X1" in the device instruction entry field. → <input type="text" value="Enter"/>	[Edit] → [Ladder Symbol] → [Open Branch] → Enter "X1" in the device instruction entry field. → <input type="text" value="Enter"/>
	ORI X1 → <input type="text" value="Enter"/>	Click  . → Enter "X1" in the device instruction entry field. → Click the [OK] button.	 +  → Enter "X1" in the device instruction entry field. → <input type="text" value="Enter"/>	[Edit] → [Ladder Symbol] → [Close Branch] → Enter "X1" in the device instruction entry field. → <input type="text" value="Enter"/>
	OUT Y1 → <input type="text" value="Enter"/>	Click  . → Enter "Y1" in the device instruction entry field. → Click the [OK] button.	 → Enter "Y1" in the device instruction entry field. → <input type="text" value="Enter"/>	[Edit] → [Ladder Symbol] → [Coil] → Enter "Y1" in the device instruction entry field. → <input type="text" value="Enter"/>
	MOV K1 D0 → <input type="text" value="Enter"/>	Click  . → Enter "MOV K1 D0" in the device instruction entry field. → Click the [OK] button.	 → Enter "MOV K1 D0" in the device instruction entry field. → <input type="text" value="Enter"/>	[Edit] → [Ladder Symbol] → [Application Instruction] → Enter "MOV K1 D0" in the device instruction entry field. → <input type="text" value="Enter"/>
	LDP X1 → <input type="text" value="Enter"/>	Click  . → Enter "X1" in the device instruction entry field. → Click the [OK] button.	 +  → Enter "X1" in the device instruction entry field. → <input type="text" value="Enter"/>	[Edit] → [Ladder Symbol] → [Pulse Contact Symbol] → [Rising Pulse] → Enter "X1" in the device instruction entry field. → <input type="text" value="Enter"/>
	LDPI X1 → <input type="text" value="Enter"/>	Click  . → Enter "X1" in the device instruction entry field. → Click the [OK] button.	 +  +  → Enter "X1" in the device instruction entry field. → <input type="text" value="Enter"/>	[Edit] → [Ladder Symbol] → [Pulse Contact Symbol] → [Rising Pulse Close] → Enter "X1" in the device instruction entry field. → <input type="text" value="Enter"/>

Example	List expression	Tool button	Function	Menu bar
	LDF X1 → <input type="text" value="Enter"/>	Click  . → Enter "X1" in the device instruction entry field. → Click the [OK] button.	<input type="text" value="Shift"/> + <input type="text" value="F8"/> → Enter "X1" in the device instruction entry field. → <input type="text" value="Enter"/>	[Edit] → [Ladder Symbol] → [Pulse Contact Symbol] → [Falling Pulse] → Enter "X1" in the device instruction entry field. → <input type="text" value="Enter"/>
	LDFI X1 → <input type="text" value="Enter"/>	Click  . → Enter "X1" in the device instruction entry field. → Click the [OK] button.	<input type="text" value="Shift"/> + <input type="text" value="Alt"/> + <input type="text" value="F6"/> → Enter "X1" in the device instruction entry field. → <input type="text" value="Enter"/>	[Edit] → [Ladder Symbol] → [Pulse Contact Symbol] → [Falling Pulse Close] → Enter "X1" in the device instruction entry field. → <input type="text" value="Enter"/>
	ORP X1 → <input type="text" value="Enter"/>	Click  . → Enter "X1" in the device instruction entry field. → Click the [OK] button.	<input type="text" value="Alt"/> + <input type="text" value="F7"/> → Enter "X1" in the device instruction entry field. → <input type="text" value="Enter"/>	[Edit] → [Ladder Symbol] → [Pulse Contact Symbol] → [Ring Pulse Branch] → Enter "X1" in the device instruction entry field. → <input type="text" value="Enter"/>
	ORPI X1 → <input type="text" value="Enter"/>	Click  . → Enter "X1" in the device instruction entry field. → Click the [OK] button.	<input type="text" value="Shift"/> + <input type="text" value="Alt"/> + <input type="text" value="F7"/> → Enter "X1" in the device instruction entry field. → <input type="text" value="Enter"/>	[Edit] → [Ladder Symbol] → [Pulse Contact Symbol] → [Rising Pulse Close Branch] → Enter "X1" in the device instruction entry field. → <input type="text" value="Enter"/>
	ORF X1 → <input type="text" value="Enter"/>	Click  . → Enter "X1" in the device instruction entry field. → Click the [OK] button.	<input type="text" value="Alt"/> + <input type="text" value="F8"/> → Enter "X1" in the device instruction entry field. → <input type="text" value="Enter"/>	[Edit] → [Ladder Symbol] → [Pulse Contact Symbol] → [Falling Pulse Branch] → Enter "X1" in the device instruction entry field. → <input type="text" value="Enter"/>
	ORFI X1 → <input type="text" value="Enter"/>	Click  . → Enter "X1" in the device instruction entry field. → Click the [OK] button.	<input type="text" value="Shift"/> + <input type="text" value="Alt"/> + <input type="text" value="F8"/> → Enter "X1" in the device instruction entry field. → <input type="text" value="Enter"/>	[Edit] → [Ladder Symbol] → [Pulse Contact Symbol] → [Falling Pulse Close Branch] → Enter "X1" in the device instruction entry field. → <input type="text" value="Enter"/>
	EGP V0 → <input type="text" value="Enter"/>	Click  . → Enter "V0" in the device instruction entry field. → Click the [OK] button.	<input type="text" value="Alt"/> + <input type="text" value="F5"/> → Enter "V0" in the device instruction entry field. → <input type="text" value="Enter"/>	[Edit] → [Ladder Symbol] → [Operation Result Rising Pulse] → Enter "V0" in the device instruction entry field. → <input type="text" value="Enter"/>
	EGF V0 → <input type="text" value="Enter"/>	Click  . → Enter "V0" in the device instruction entry field. → Click the [OK] button.	<input type="text" value="Ctrl"/> + <input type="text" value="Alt"/> + <input type="text" value="F5"/> → Enter "V0" in the device instruction entry field. → <input type="text" value="Enter"/>	[Edit] → [Ladder Symbol] → [Operation Result Falling Pulse] → Enter "V0" in the device instruction entry field. → <input type="text" value="Enter"/>
	INV → <input type="text" value="Enter"/>	Click  . → Click the [OK] button.	<input type="text" value="Ctrl"/> + <input type="text" value="Alt"/> + <input type="text" value="F10"/> → <input type="text" value="Enter"/>	[Edit] → [Ladder Symbol] → [Invert Operation Results] → <input type="text" value="Enter"/>





Example	List expression	Tool button	Function	Menu bar
Horizontal line 	—	Click  . → Enter the number of lines to be added in the device instruction entry field. → Click the [OK] button.	 → Enter the number of lines to be added in the device instruction entry field. → 	[Edit] → [Ladder Symbol] → [Horizontal Line] → Enter the number of lines to be added in the device instruction entry field. → 
Vertical line 	—	Click  . → Enter the number of lines to be added in the device instruction entry field. → Click the [OK] button.	 +  → Enter the number of lines to be added in the device instruction entry field. → 	[Edit] → [Ladder Symbol] → [Vertical Line] → Enter the number of lines to be added in the device instruction entry field. → 
Deletion of a horizontal line 	—	Click  . → Enter the number of lines to be deleted in the device instruction entry field. → Click the [OK] button.	 +  → Enter the number of lines to be deleted in the device instruction entry field. → 	[Edit] → [Ladder Symbol] → [Delete Horizontal Line] → Enter the number of lines to be deleted in the device instruction entry field. → 
Deletion of a vertical line 	—	Click  . → Enter the number of lines to be deleted in the device instruction entry field. → Click the [OK] button.	 +  → Enter the number of lines to be deleted in the device instruction entry field. → 	[Edit] → [Ladder Symbol] → [Delete Vertical Line] → Enter the number of lines to be deleted in the device instruction entry field. → 

## Point

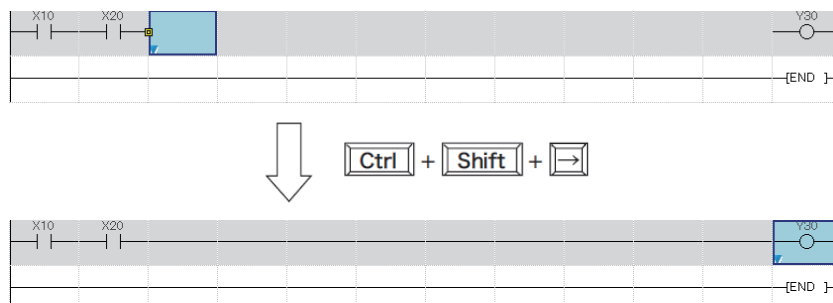
- Adding and deleting lines

Lines can also be added and deleted using the  key and an arrow key.

- Continuous addition of horizontal lines

Horizontal lines can be continuously added from the cursor position using the  key,  key, and an arrow key (/).

(Horizontal lines can be continuously deleted in the same way.)



A

The following figures show how to enter instructions for a low-speed timer, high-speed timer, low-speed retentive timer, high-speed retentive timer, and edge relay.

- Low-speed timer

		OUT T0 K100	OK	Cancel	Extd Dspl
		T0 K100	OK	Cancel	Extd Dspl

- High-speed timer

		OUTH T0 K100	OK	Cancel	Extd Dspl
		H T0 K100	OK	Cancel	Extd Dspl

- Low-speed retentive timer

		OUT ST0 K100	OK	Cancel	Extd Dspl
		ST0 K100	OK	Cancel	Extd Dspl

- High-speed retentive timer

		OUTH ST0 K100	OK	Cancel	Extd Dspl
		H ST0 K100	OK	Cancel	Extd Dspl

- Edge relay

		EGP V0	OK	Cancel	Extd Dspl
		V0	OK	Cancel	Extd Dspl

## Precautions

To enter a retentive timer, set the number of device points in the device setting of the CPU parameter.

# Appendix 3 Offset/Gain Setting


Configure the offset/gain setting as needed for the A/D converter module and D/A converter module, which handle analog signals, to secure the signal accuracy.

For the MELSEC iQ-R series A/D converter modules and D/A converter modules, the offset/gain setting can be easily configured with GX Works3.

This section describes how to configure the offset/gain setting with the A/D converter module (R60AD4).

Configure the offset/gain setting with the D/A converter module (R60DA4) with the same procedure.

For details, refer to the following.

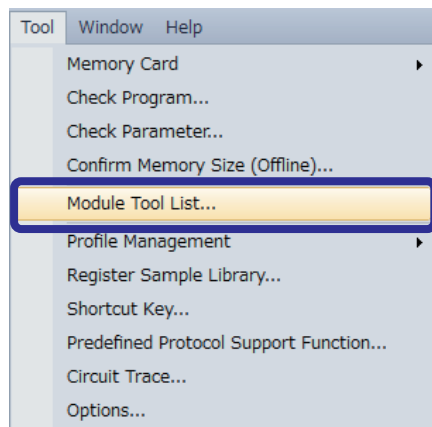
 MELSEC iQ-R Analog-Digital Converter Module User's Manual (Startup)

 MELSEC iQ-R Digital-Analog Converter Module User's Manual (Startup)

## Configuring the offset/gain setting with GX Works3

The setting procedure for the offset/gain setting of the A/D converter module is as follows:

### Operating procedure

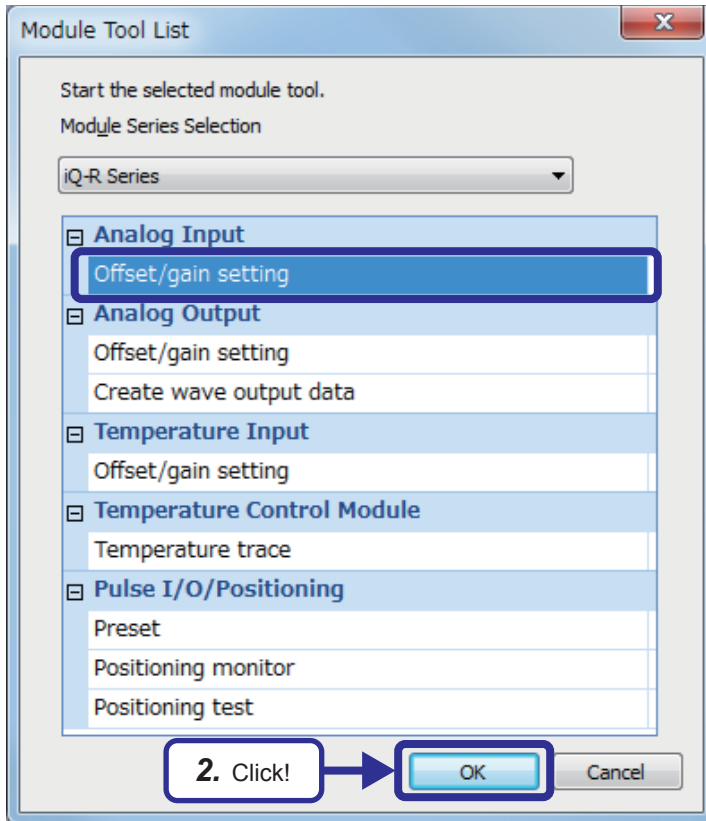


1. Click [Tool] → [Module Tool List] from the menu.

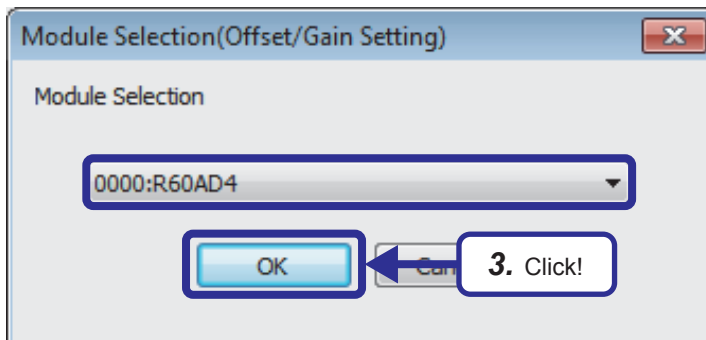


(To the next page)

(From the previous page)



2. In "Analog Input", select "Offset/gain setting" and click the [OK] button.

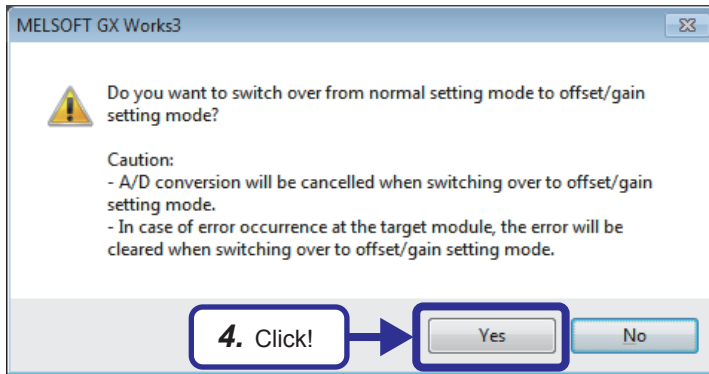


3. Select "R60AD4" and click the [OK] button.

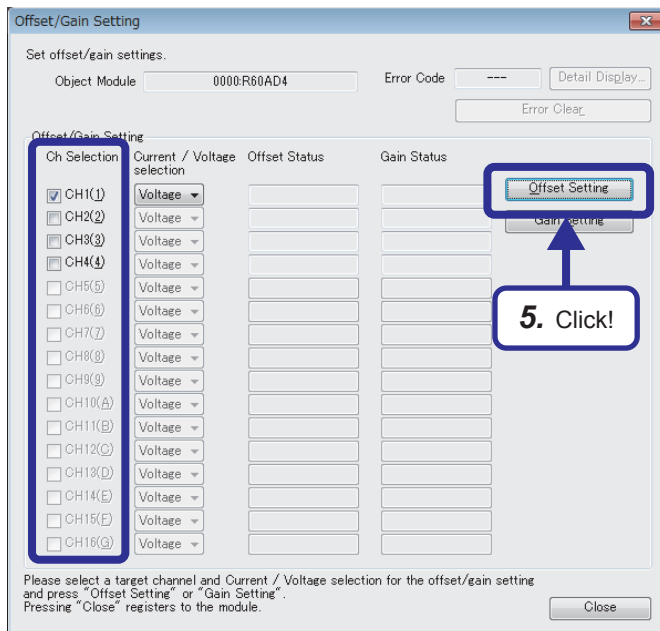


(To the next page)

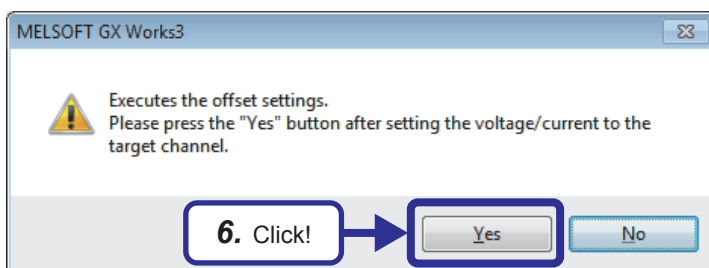
(From the previous page)



4. Click the [Yes] button.



5. Mark the checkbox of the channel where offset and gain values are to be set, and click the [Offset Setting] button.



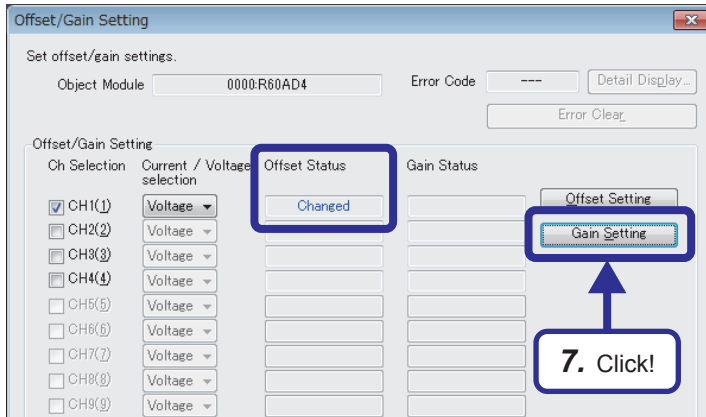
6. Apply the offset voltage or current to the terminal of the corresponding channel, and click the [Yes] button.



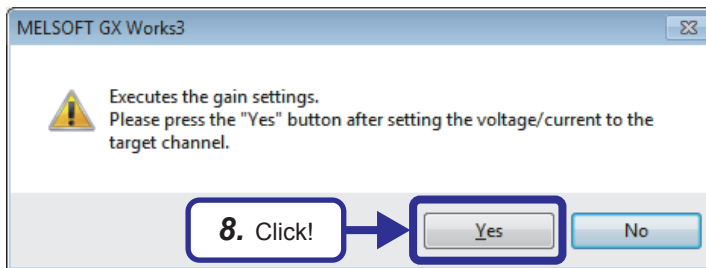
(To the next page)



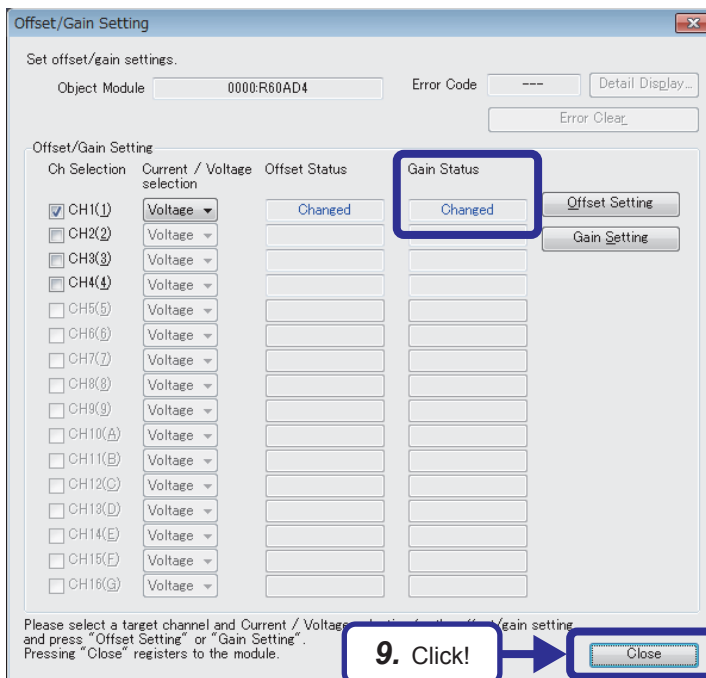
(From the previous page)



7. Check that "Offset Status" has changed to "Changed", and click the [Gain Setting] button.



8. Apply the gain voltage or current to the terminal of the corresponding channel, and click the [Yes] button.



9. Check that "Gain Status" has changed to "Changed", and click the [Close] button.

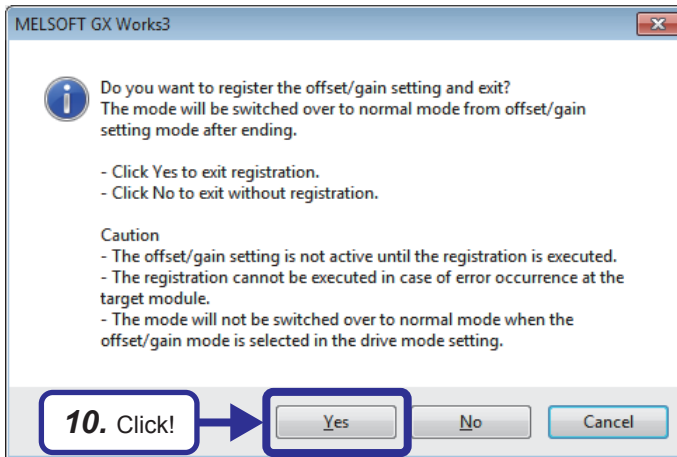


(To the next page)

(From the previous page)



**10.** Click the [Yes] button.



#### Point

When an error code is displayed in the middle of settings, error details and actions can be checked by clicking the [Detail Display] button on the right of the error code display area. Error codes can be cleared by clicking the [Error Clear] button.

# Appendix 3.1 Offset value and gain value

---

## I/O characteristics of A/D conversion

An I/O conversion characteristic of A/D conversion is expressed by the slope of the straight line connecting the offset value and the gain value at the time when an analog signal (voltage or current) from outside the programmable controller is converted to a digital value.

### ■Offset value

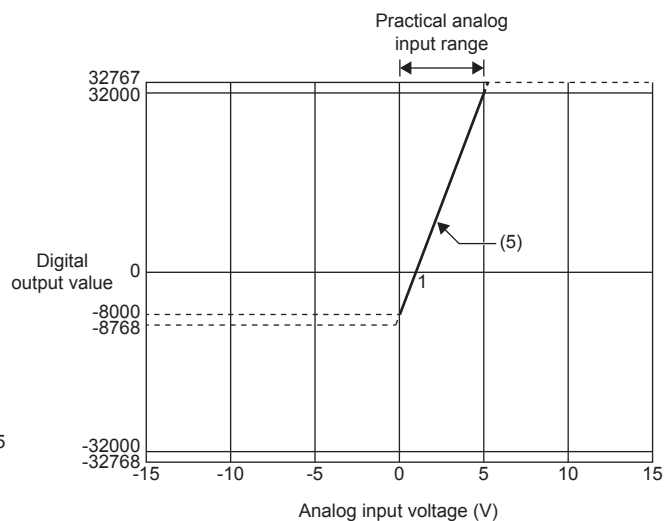
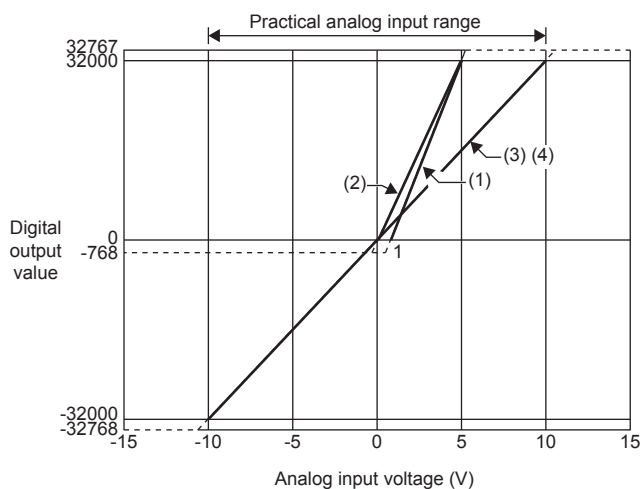
This value is the analog input value (voltage or current) where the corresponding digital output value is 0.

### ■Gain value

This value is the analog input value (voltage or current) where the corresponding digital output value is 32000.

## ■Voltage input characteristics

The following shows the list of the analog input ranges and the graphs of each voltage input characteristic, at the voltage input.



No.	Input range setting	Offset value	Gain value	Digital output value <sup>*3</sup>	Resolution
(1)	1 to 5V	1V	5V	0 to 32000	125.0 $\mu$ V
(2)	0 to 5V	0V	5V		156.3 $\mu$ V
(3)	-10 to 10V	0V	10V	-32000 to 32000	312.5 $\mu$ V
(4)	0 to 10V	0V	10V	0 to 32000	
(5)	1 to 5V (extended mode)	1V	5V	-8000 to 32000	125.0 $\mu$ V
—	User range setting	*1	*1	-32000 to 32000	47.7 $\mu$ V <sup>*2</sup>

\*1 Set the offset value and gain value in the user range setting within a range that satisfies the following conditions. If the following conditions are not satisfied, A/D conversion may not be performed properly.

Setting range of the offset value and gain value: -10 to 10V

$((\text{gain value}) - (\text{offset value})) \geq 1.6\text{V}$

\*2 Maximum resolution in the user range setting.

\*3 When analog input exceeds the range of the digital output value, the digital output value is fixed to the maximum or minimum.

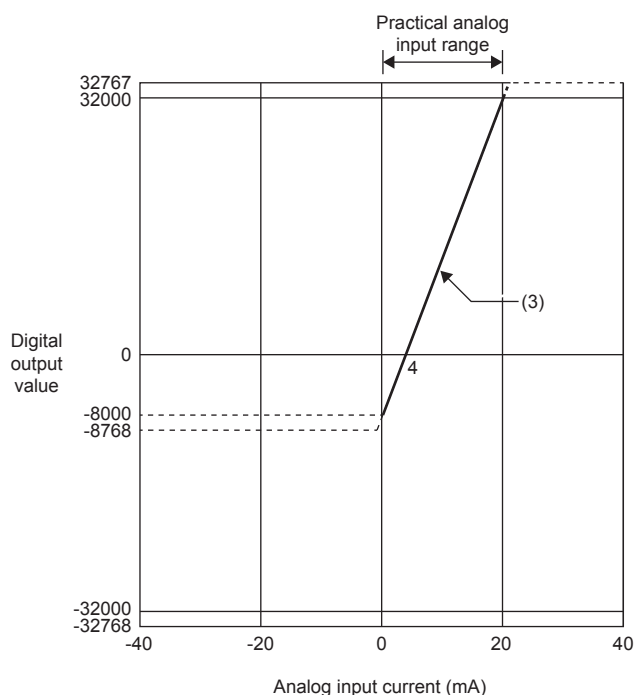
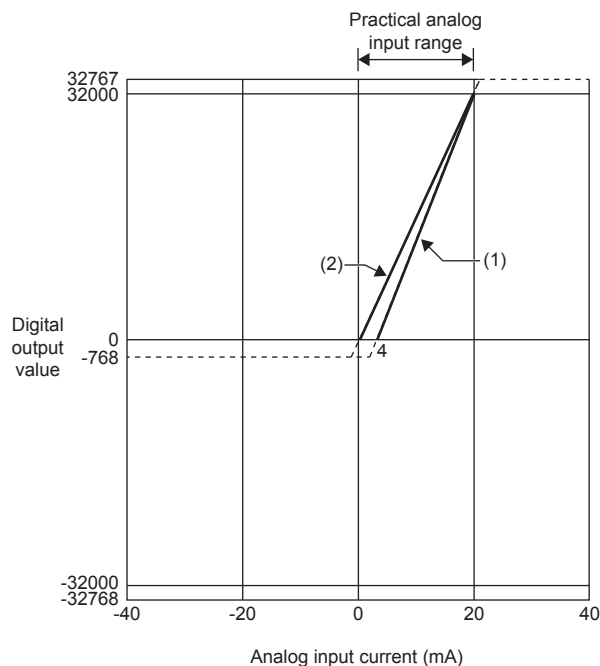
Input range setting	Digital output value	
	Minimum	Maximum
1 to 5V	-768	32767
0 to 5V		
-10 to 10V	-32768	
0 to 10V	-768	
1 to 5V (extended mode)	-8768	
User range setting	-32768	

### Point

- Set values within the practical range of the analog input and digital output at each input range. If the values are out of the range, the resolution and accuracy may not fall within the range of the performance specifications. (Do not use values in the dotted line region in the graph of voltage input characteristics.)
- The range of 1 to 5V (extended mode) allows extending the lower limit value of analog input. The upper limit value of analog input cannot be extended.
- Do not set the voltage over  $\pm 15\text{V}$ . Doing so can cause breakdown of the elements.

## ■Current input characteristics

The following shows the list of the analog input ranges and the graph of each current input characteristic, at the current input.



No.	Input range setting	Offset value	Gain value	Digital output value <sup>*3</sup>	Resolution
(1)	4 to 20mA	4mA	20mA	0 to 32000	500.0nA
(2)	0 to 20mA	0mA	20mA		625.0nA
(3)	4 to 20mA (extended mode)	4mA	20mA	-8000 to 32000	500.0nA
—	User range setting	*1	*1	-32000 to 32000	190.7nA*2

\*1 Set the offset value and gain value in the user range setting within a range that satisfies the following conditions. If the following conditions are not satisfied, A/D conversion may not be performed properly.

Gain value  $\leq$  20mA, offset value  $\geq$  0mA

((gain value) - (offset value))  $\geq$  6.1mA

\*2 Maximum resolution in the user range setting.

\*3 When analog input exceeds the range of the digital output value, the digital output value is fixed to the maximum or minimum.

Input range setting	Digital output value	
	Minimum	Maximum
4 to 20mA	-768	32767
0 to 20mA		
4 to 20mA (extended mode)	-8768	
User range setting	-32768	

### Point

- Set values within the practical range of the analog input and digital output at each input range. If the values are out of the range, the resolution and accuracy may not fall within the range of the performance specifications. (Do not use values in the dotted line region in the graph of current input characteristics.)
- The range of 4 to 20mA (extended mode) allows extending the lower limit value of analog input. The upper limit value of analog input cannot be extended.
- Do not set the voltage over  $\pm 30$ mA. Doing so can cause breakdown of the elements.

## I/O characteristics of D/A conversion

An I/O conversion characteristic of D/A conversion is expressed by the slope of the straight line connecting the offset value and the gain value at the time when a digital value written from the CPU module is converted to an analog output value (voltage or current).

### ■Offset value

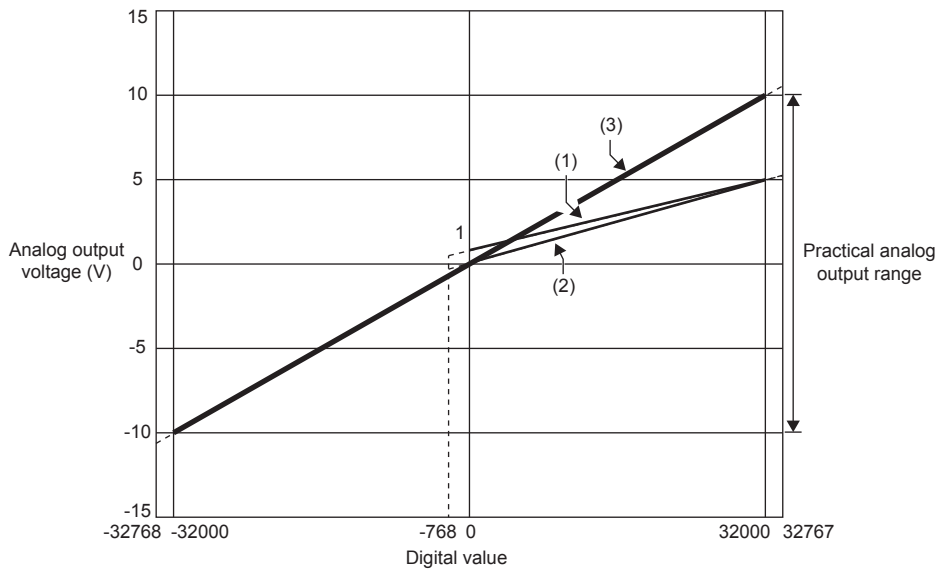
This analog output value (voltage or current) corresponds to a digital value of 0 that is set through the CPU module.

### ■Gain value

This analog output value (voltage or current) corresponds to a digital value of 32000 that is set through the CPU module.

## ■Voltage output characteristics

The following shows the list of the analog output ranges and the graphs of each voltage output characteristic, at the voltage output.



No.	Analog output range setting	Offset value	Gain value	Digital output value	Resolution
(1)	1 to 5V	1V	5V	0 to 32000	125.0 $\mu$ V
(2)	0 to 5V	0V	5V		156.3 $\mu$ V
(3)	-10 to 10V	0V	10V	-32000 to 32000	312.5 $\mu$ V
—	User range setting (voltage)	*1	*1	-32000 to 32000	312.5 $\mu$ V <sup>*2</sup>

\*1 Set the offset value and gain value in the user range setting within a range that satisfies the following conditions. If the following conditions are not satisfied, D/A conversion may not be performed properly.

- Setting range of the offset value and gain value: -10 to 10V
- ((gain value) - (offset value))  $\geq$  4.0V

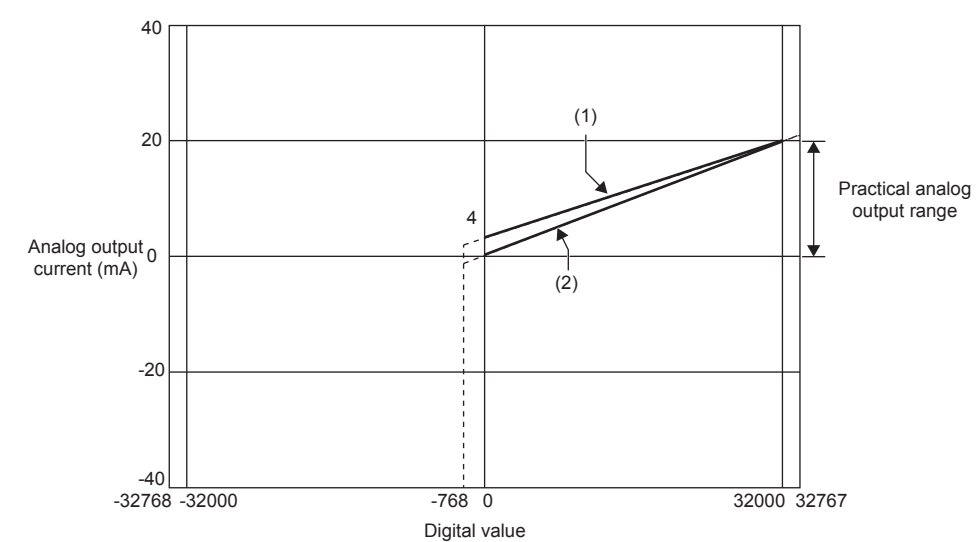
\*2 Maximum resolution in the user range setting.

### Point

- Set values within the practical range of the digital input and analog output at each output range. If the values are out of the range, the resolution and accuracy may not fall within the range of the performance specifications. (Do not use values in the dotted line region in the graph of voltage output characteristics.)

■Current output characteristics

The following shows the list of the analog output ranges and the graphs of each current output characteristic, at the current output.



No.	Analog output range setting	Offset value	Gain value	Digital output value	Resolution
(1)	4 to 20mA	4mA	20mA	0 to 32000	500.0nA
(2)	0 to 20mA	0mA	20mA		625.0nA
—	User range setting (current)	*1	*1	-32000 to 32000	350.9nA*2

\*1 Set the offset value and gain value in the user range setting within a range that satisfies the following conditions. If the following conditions are not satisfied, D/A conversion may not be performed properly.

- Offset value  $\geq 0\text{mA}$ , gain value  $\leq 20\text{mA}$
- ((gain value) - (offset value))  $\geq 11.3\text{mA}$

\*2 Maximum resolution in the user range setting.

Point

- Set values within the practical range of the digital input and analog output at each output range. If the values are out of the range, the resolution and accuracy may not fall within the range of the performance specifications. (Do not use values in the dotted line region in the graph of current output characteristics.)

A



# Appendix 4 Specifications of the A/D Converter Module and D/A Converter Module

## Appendix 4.1 A/D converter module

For details, refer to the following.

 MELSEC iQ-R Analog-Digital Converter Module User's Manual (Startup)

### Performance specifications

#### ■R60AD4

Item		Specifications		
Number of analog input channels		4 channels		
Analog input voltage		-10 to 10VDC (input resistance 1MΩ)		
Analog input current		0 to 20mA (input resistance 250Ω)		
Digital output value		16-bit signed binary value (-32768 to 32767)		
I/O characteristics, resolution <sup>*1</sup>		Analog input range		Digital output value
		Voltage	0 to 10V	0 to 32000
			0 to 5V	
			1 to 5V	
			1 to 5V (extended mode)	-8000 to 32000
			-10 to 10V	-32000 to 32000
			User range setting	
		Current	0 to 20mA	0 to 32000
			4 to 20mA	
			4 to 20mA (extended mode)	-8000 to 32000
			User range setting	-32000 to 32000
				Resolution
				312.5μV
				156.3μV
				125.0μV
Accuracy (accuracy of the maximum digital output value) <sup>*2</sup>		Ambient temperature 25±5°C: within ±0.1% (±32 digit) Ambient temperature 0 to 55°C: within ±0.3% (±96 digit)		
Conversion speed		80μs/CH		
Absolute maximum input		Voltage: ±15V, Current: 30mA <sup>*3</sup>		
Number of offset/gain settings <sup>*4</sup>		50000 times maximum		
Isolation method		Between I/O terminals and programmable controller power supply: Photocoupler Between input channels: Non-isolation		
Withstand voltage		Between I/O terminals and programmable controller power supply: 500VACrms for 1 minute		
Insulation resistance		Between I/O terminals and programmable controller power supply: 10MΩ or higher, at 500VDC		
Number of occupied I/O points		16 points (I/O assignment: Intelligent 16 points)		
External interface		18-point terminal block		
Applicable wire size		0.3 to 0.75mm <sup>2</sup> (22 to 18 AWG)		
Applicable solderless terminal		R1.25-3 (solderless terminal with an insulation sleeve cannot be used)		
Internal current consumption (5VDC)		0.22A		
External dimensions	Height	106mm (Base unit mounting side: 98mm)		
	Width	27.8mm		
	Depth	131mm		
Weight		0.12kg		

\*1 For details on the I/O conversion characteristics, refer to the following.

 Appendix - 12 I/O characteristics of A/D conversion

\*2 Except for the conditions under noise influence.

\*3 These voltage and current values are instantaneous values at which no breakdown occurs in the internal resistance of the module.

\*4 A count more than 50000 times causes Number of writes to offset/gain settings reach limit error (error code: 1080H).

## ■R60ADV8

Item		Specifications			
Number of analog input channels		8 channels			
Analog input voltage		-10 to 10VDC (input resistance 1MΩ)			
Analog input current		—			
Digital output value		16-bit signed binary value (-32768 to 32767)			
I/O characteristics, 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 32000	125.0μV
			-10 to 10V	-32000 to 32000	312.5μV
			User range setting		47.7μV
Accuracy (accuracy of the maximum digital output value)*2		Ambient temperature 25±5°C: within ±0.1% (±32 digit) Ambient temperature 0 to 55°C: within ±0.3% (±96 digit)			
Conversion speed		80μs/CH			
Absolute maximum input		Voltage: ±15V*3			
Number of offset/gain settings*4		50000 times maximum			
Isolation method		Between I/O terminals and programmable controller power supply: Photocoupler Between input channels: Non-isolation			
Withstand voltage		Between I/O terminals and programmable controller power supply: 500VACrms for 1 minute			
Insulation resistance		Between I/O terminals and programmable controller power supply: 10MΩ or higher, at 500VDC			
Number of occupied I/O points		16 points (I/O assignment: Intelligent 16 points)			
External interface		18-point terminal block			
Applicable wire size		0.3 to 0.75mm <sup>2</sup> (22 to 18 AWG)			
Applicable solderless terminal		R1.25-3 (solderless terminal with an insulation sleeve cannot be used)			
Internal current consumption (5VDC)		0.23A			
External dimensions	Height	106mm (Base unit mounting side: 98mm)			
	Width	27.8mm			
	Depth	131mm			
Weight		0.12kg			

\*1 For details on the I/O conversion characteristics, refer to the following.

☞ Appendix - 12 I/O characteristics of A/D conversion

\*2 Except for the conditions under noise influence.

\*3 This voltage value is an instantaneous value at which no breakdown occurs in the internal resistance of the module.

\*4 A count more than 50000 times causes Number of writes to offset/gain settings reach limit error (error code: 1080H).

A

## ■R60ADI8

Item		Specifications			
Number of analog input channels		8 channels			
Analog input voltage		—			
Analog input current		0 to 20mADC (input resistance 250Ω)			
Digital output value		16-bit signed binary value (-32768 to 32767)			
I/O characteristics, resolution*1		Analog input range		Digital output value	Resolution
		Current	0 to 20mA	0 to 32000	625.0nA
			4 to 20mA		500.0nA
			4 to 20mA (extended mode)	-8000 to 32000	500.0nA
			User range setting	-32000 to 32000	190.7nA
Accuracy (accuracy of the maximum digital output value)*2		Ambient temperature 25±5°C: within ±0.1% (±32 digit) Ambient temperature 0 to 55°C: within ±0.3% (±96 digit)			
Conversion speed		80μs/CH			
Absolute maximum input		Current: 30mA*3			
Number of offset/gain settings*4		50000 times maximum			
Isolation method		Between I/O terminals and programmable controller power supply: Photocoupler Between input channels: Non-isolation			
Withstand voltage		Between I/O terminals and programmable controller power supply: 500VACrms for 1 minute			
Insulation resistance		Between I/O terminals and programmable controller power supply: 10MΩ or higher, at 500VDC			
Number of occupied I/O points		16 points (I/O assignment: Intelligent 16 points)			
External interface		18-point terminal block			
Applicable wire size		0.3 to 0.75mm <sup>2</sup> (22 to 18 AWG)			
Applicable solderless terminal		R1.25-3 (solderless terminal with an insulation sleeve cannot be used)			
Internal current consumption (5VDC)		0.22A			
External dimensions	Height	106mm (Base unit mounting side: 98mm)			
	Width	27.8mm			
	Depth	131mm			
Weight		0.12kg			

\*1 For details on the I/O conversion characteristics, refer to the following.

☞ Appendix - 12 I/O characteristics of A/D conversion

\*2 Except for the conditions under noise influence.

\*3 This current value is an instantaneous value at which no breakdown occurs in the internal resistance of the module.

\*4 A count more than 50000 times causes Number of writes to offset/gain settings reach limit error (error code: 1080H).

# Appendix 4.2 D/A converter module

For details, refer to the following.


 MELSEC iQ-R Digital-Analog Converter Module User's Manual (Startup)

## Performance specifications

### ■R60DA4

Item		Specifications		
Number of analog output channels		4 channels		
Digital input		16-bit signed binary value (-32768 to 32767)		
Analog output voltage		-10 to 10VDC (external load resistance value 1kΩ or more) 0 to 5VDC (external load resistance value 500Ω or more)		
Analog output current		0 to 20mADC (external load resistance value 0 to 600Ω)		
I/O characteristics, resolution <sup>*1</sup>		Analog output range		Digital output value
		Voltage	0 to 5V	0 to 32000
			1 to 5V	
			-10 to 10V	-32000 to 32000
			User range setting	
		Current	0 to 20mA	0 to 32000
			4 to 20mA	
			User range setting	-32000 to 32000
Accuracy (accuracy of the maximum analog output value) <sup>*2</sup>		Ambient temperature 25±5°C: Within ±0.1% (Voltage ±10mV, current ±20μA) Ambient temperature 0 to 55°C: Within ±0.3% (Voltage ±30mV, current ±60μA)		
Conversion speed	Normal output mode	80μs/CH		
	Wave output mode	80μs/CH		
Number of offset/gain settings <sup>*3</sup>		50000 times maximum		
Output short circuit protection		Protected		
Isolation method		Between I/O terminals and programmable controller power supply: Photocoupler Between output channels: Non-isolation Between the external power supply and analog outputs: Transformer isolation		
Withstand voltage		Between I/O terminals and programmable controller power supply: 500VACrms for 1 minute Between the external power supply and analog outputs: 500VACrms for 1 minute		
Insulation resistance		Between I/O terminals and programmable controller power supply: 10MΩ or higher, at 500VDC		
Number of occupied I/O points		16 points (I/O assignment: Intelligent 16 points)		
External interface		18-point terminal block		
Applicable wire size		0.3 to 0.75mm <sup>2</sup>		
Applicable solderless terminal		R1.25-3 (solderless terminal with an insulation sleeve cannot be used)		
External power supply		24VDC +20%, -15%		
		Ripple, spike 500mV <sub>P-P</sub> or lower		
		Inrush current: 5.0A, 690μs or shorter		
		Current consumption: 0.14A		
Internal current consumption (5VDC)		0.16A		
External dimensions	Height	106mm (Base unit mounting side: 98mm)		
	Width	27.8mm		
	Depth	131mm		
Weight		0.14kg		

\*1 For details on the I/O conversion characteristics, refer to the following.

 Appendix - 15 I/O characteristics of D/A conversion

\*2 Except for the conditions under noise influence.


\*3 A count more than 50000 times causes Number of writes to offset/gain settings reach limit error (error code: 1080H).

A

## ■R60DAV8

Item		Specifications		
Number of analog output channels		8 channels		
Digital input		16-bit signed binary value (-32768 to 32767)		
Analog output voltage		-10 to 10VDC (external load resistance value 1kΩ or more) 0 to 5VDC (external load resistance value 500Ω or more)		
Analog output current		—		
I/O characteristics, resolution*1		Analog output range		Digital output value
		Voltage	0 to 5V	0 to 32000
			1 to 5V	
			-10 to 10V	-32000 to 32000
			User range setting	
				Resolution
				156.3μV
				125.0μV
				312.5μV
Accuracy (accuracy of the maximum analog output value) *2		Ambient temperature 25±5°C: Within ±0.1% (Voltage ±10mV) Ambient temperature 0 to 55°C: Within ±0.3% (Voltage ±30mV)		
Conversion speed		80μs/CH		
Number of offset/gain settings*3		50000 times maximum		
Output short circuit protection		Protected		
Isolation method		Between I/O terminals and programmable controller power supply: Photocoupler Between output channels: Non-isolation Between the external power supply and analog outputs: Transformer isolation		
Withstand voltage		Between I/O terminals and programmable controller power supply: 500VACrms for 1 minute Between the external power supply and analog outputs: 500VACrms for 1 minute		
Insulation resistance		Between I/O terminals and programmable controller power supply: 10MΩ or higher, at 500VDC		
Number of occupied I/O points		16 points (I/O assignment: Intelligent 16 points)		
External interface		18-point terminal block		
Applicable wire size		0.3 to 0.75mm <sup>2</sup>		
Applicable solderless terminal		R1.25-3 (solderless terminal with an insulation sleeve cannot be used)		
External power supply		24VDC +20%, -15%		
		Ripple, spike 500mV <sub>P-P</sub> or lower		
		Inrush current: 5.0A, 670μs or shorter		
		Current consumption: 0.16A		
Internal current consumption (5VDC)		0.16A		
External dimensions	Height	106mm (Base unit mounting side: 98mm)		
	Width	27.8mm		
	Depth	131mm		
Weight		0.14kg		

\*1 For details on the I/O conversion characteristics, refer to the following.

 Appendix - 15 I/O characteristics of D/A conversion

\*2 Except for the conditions under noise influence.

\*3 A count more than 50000 times causes Number of writes to offset/gain settings reach limit error (error code: 1080H).

## ■R60DAI8

Item		Specifications		
Number of analog output channels		8 channels		
Digital input		16-bit signed binary value (-32768 to 32767)		
Analog output voltage		—		
Analog output current		0 to 20mADC (external load resistance value 0 to 600Ω)		
I/O characteristics, resolution*1		Analog output range	Digital output value	Resolution
		Current	0 to 32000	625.0nA
				500.0nA
			-32000 to 32000	350.9nA
Accuracy (accuracy of the maximum analog output value) *2		Ambient temperature 25±5°C: Within ±0.1% (Current ±20μA) Ambient temperature 0 to 55°C: Within ±0.3% (Current ±60μA)		
Conversion speed		80μs/CH		
Number of offset/gain settings*3		50000 times maximum		
Output short circuit protection		Protected		
Isolation method		Between I/O terminals and programmable controller power supply: Photocoupler Between output channels: Non-isolation Between the external power supply and analog outputs: Transformer isolation		
Withstand voltage		Between I/O terminals and programmable controller power supply: 500VACrms for 1 minute Between the external power supply and analog outputs: 500VACrms for 1 minute		
Insulation resistance		Between I/O terminals and programmable controller power supply: 10MΩ or higher, at 500VDC		
Number of occupied I/O points		16 points (I/O assignment: Intelligent 16 points)		
External interface		18-point terminal block		
Applicable wire size		0.3 to 0.75mm <sup>2</sup>		
Applicable solderless terminal		R1.25-3 (solderless terminal with an insulation sleeve cannot be used)		
External power supply		24VDC +20%, -15%		
		Ripple, spike 500mV <sub>P-P</sub> or lower		
		Inrush current: 5.0A, 700μs or shorter		
		Current consumption: 0.26A		
Internal current consumption (5VDC)		0.16A		
External dimensions	Height	106mm (Base unit mounting side: 98mm)		
	Width	27.8mm		
	Depth	131mm		
Weight		0.14kg		

\*1 For details on the I/O conversion characteristics, refer to the following.


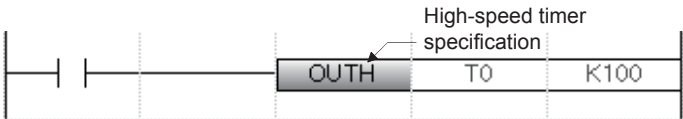
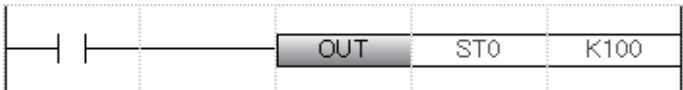
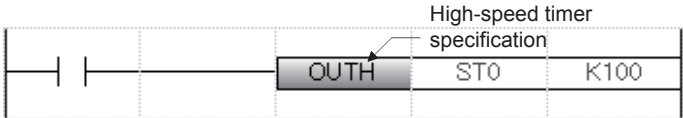
☞ Appendix - 15 I/O characteristics of D/A conversion

\*2 Except for the conditions under noise influence.

\*3 A count more than 50000 times causes Number of writes to offset/gain settings reach limit error (error code: 1080H).

# Appendix 5 Timer, Counter

## Appendix 5.1 Timer

Function		RCPU
Low-speed timer	Measurement unit	<ul style="list-style-type: none"> <li>100ms (default)</li> <li>The unit can be changed within the range of 1 to 100ms in parameters.</li> </ul>
	Specification method	
High-speed timer	Measurement unit	<ul style="list-style-type: none"> <li>10ms (default)</li> <li>The unit can be changed within the range of 1 to 100ms in parameters.</li> </ul>
	Specification method	 <p>* Configure the high-speed timer setting with sequence programs.</p>
Retentive timer	Measurement unit	Same as that of the low-speed timer
	Specification method	
High-speed retentive timer	Measurement unit	Same as that of the high-speed timer
	Specification method	 <p>* Configure the high-speed timer setting with sequence programs.</p>
Setting range		<ul style="list-style-type: none"> <li>1 to 32767</li> </ul>
Processing of the setting value 0		<ul style="list-style-type: none"> <li>Momentarily on</li> </ul>
Index modification	Contact	<ul style="list-style-type: none"> <li>Available (Only Z0 and Z1 can be used.)</li> </ul>
	Coil	<ul style="list-style-type: none"> <li>Available (Only Z0 and Z1 can be used.)</li> </ul>
	Setting value	<ul style="list-style-type: none"> <li>Not available</li> </ul>
	Current value	<ul style="list-style-type: none"> <li>Available (Z0 to Z15 can be used.)</li> </ul>
Current value update processing		<ul style="list-style-type: none"> <li>When the OUT T□ instruction is executed</li> </ul>
Contact on/off processing		

### Precautions for using timers

The RCPU updates the current values of timers and turns on or off contacts of them at execution of the OUT T□ instruction. Thus, if the current value of a timer becomes equal to or larger than the set value when the coil of the timer turns on, the contact of the timer will turn on.

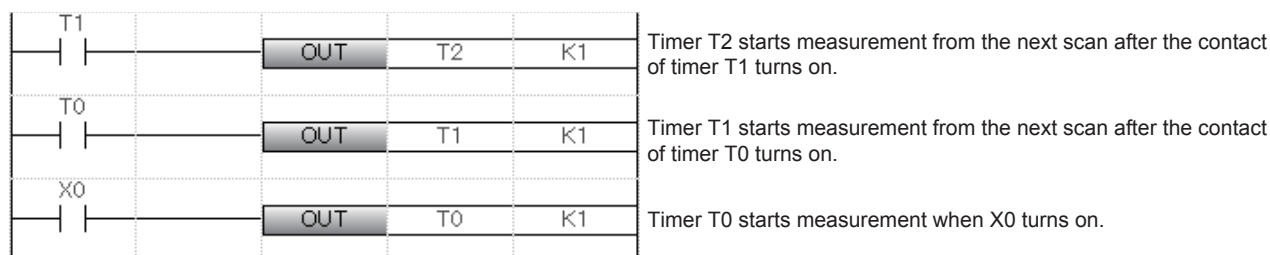
To create a program in which the operation of a timer contact triggers the operation of another timer, program the timers in order from the one that operates last.

In the following cases, if a program is created in order of timer measurements, all timers turn on in the same scan.

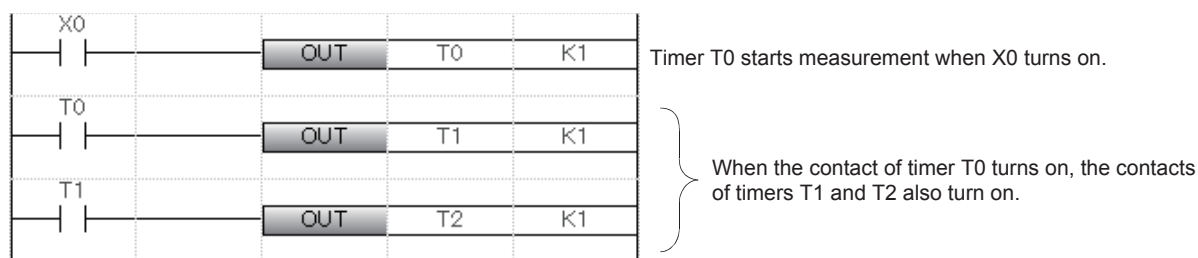
- The value set to a high-speed timer is smaller than the scan time.
- The value set to a low-speed timer is 1.

Ex.

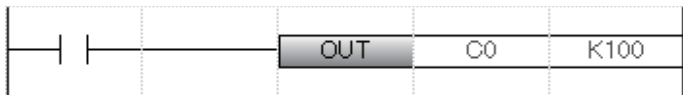
- When timers T0 to T2 are programmed in order from the one that measures last



- When timers T0 to T2 are programmed in order of measurement



## Appendix 5.2 Counter

Function		RCP
Specification method		
Index modification	Contact	• Available (Only Z0 and Z1 can be used.)
	Coil	• Available (Only Z0 and Z1 can be used.)
	Setting value	• Not available
	Current value	• Available (Z0 to Z15 can be used.)
Current value update processing		• When the OUT Tn instruction is executed
Contact on/off processing		

A

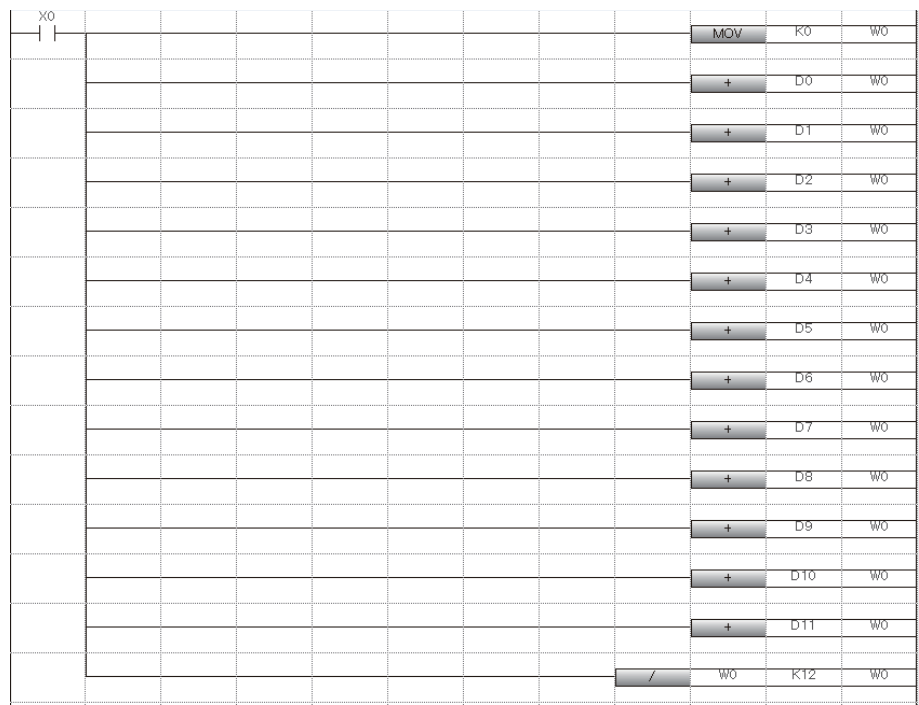


# Appendix 6    Inline ST

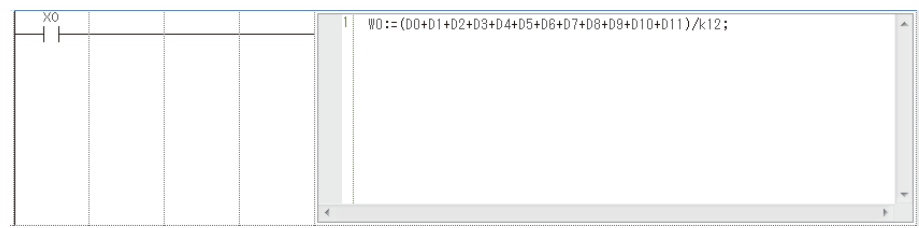
The inline ST is the function used to create an inline ST box that displays an ST program in the cell of the instruction corresponding to a coil in the ladder editor, and edit and monitor it.

This function enables to create numerical operations and character string processing easily in ladder programs.

- Program that does not use the inline ST



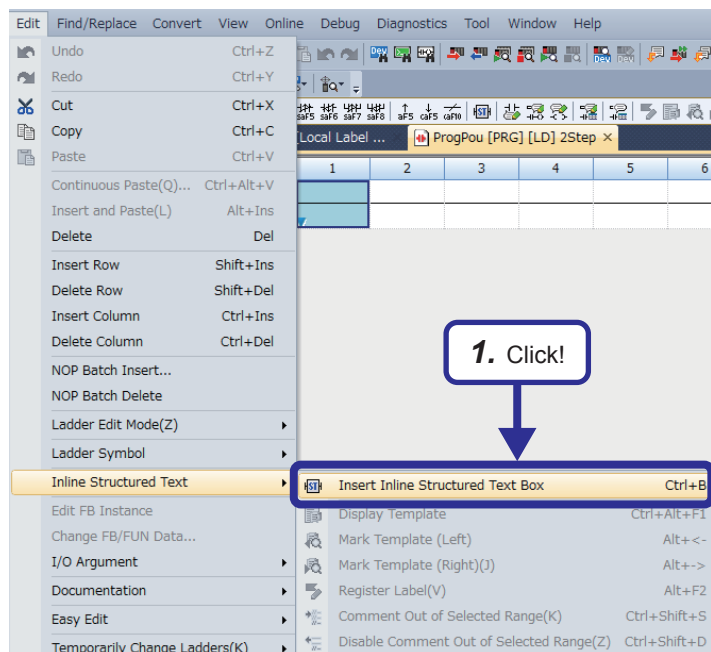
- Program that uses the inline ST




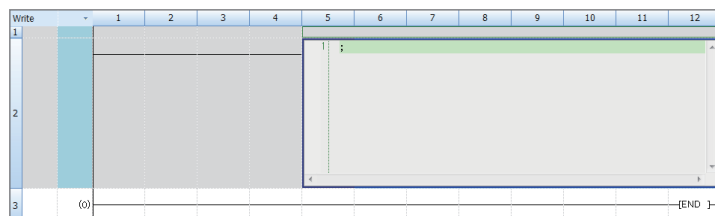
# Appendix 6.1 Using the inline ST

## Inserting the inline ST

### Operating procedure



1. Click  on the toolbar, or select [Edit] → [Inline Structured Text] → [Insert Inline Structured Text Box] from the menu.

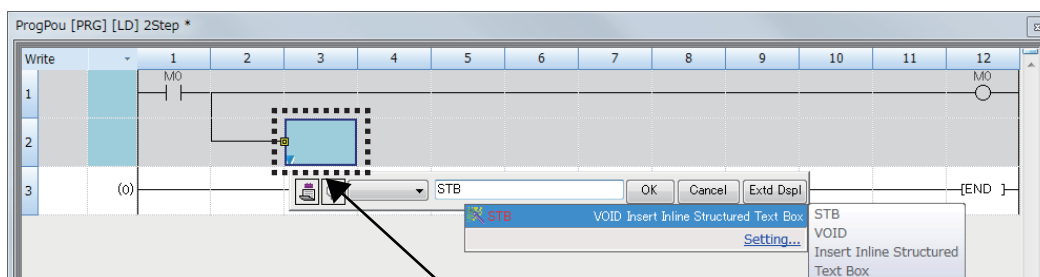


2. An inline ST box is inserted on the ladder editor.

A

### Point

- Entering "STB" in the ladder entry dialog box inserts an inline ST box.
- To insert the inline ST in parallel, create lines and then insert an inline ST box.



Insert an inline ST box at an input part in parallel.

## Editing the inline ST

Double-click an inline ST box.

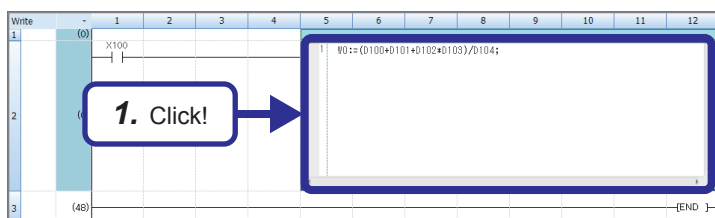
The program in an inline ST box can be edited in the same way as that is written in ST.

For details, refer to the following.

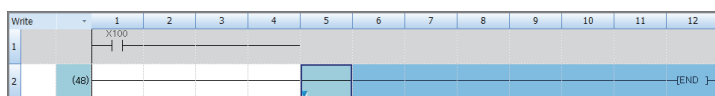
 GX Works3 Operating Manual

## Deleting an inline ST box

### Operating procedure



**1.** Select an inline ST box to be deleted.



**2.** Click [Edit] → [Delete] from the menu.

The inline ST box is deleted.

## Appendix 6.2 Precautions for using the inline ST

### ■Precautions for creating ladders

- Only one inline ST box can be created in one ladder block.
- An FB instance and inline ST box cannot be arranged in one ladder block.
- Creating an inline ST box at the position of an instruction corresponding to a contact creates an inline ST box at the position of an instruction corresponding to a coil.
- When an unconverted inline ST exists on the ladder editor, the program cannot be edited. Convert the program and edit it.

### ■Precautions on the inside of an inline ST box

- Up to 2048 characters can be entered in an inline ST box. (A line break is counted as one character.)
- Up to 23 local labels can be used in an inline ST box (except for constants).
- Labels of the following data types cannot be used.
  - Counter
  - Timer
  - Retentive timer
  - Pointer
  - Structure
  - Array
  - Function block
- Lower-case device names cannot be used as labels regardless of option settings.
- Instructions cannot be entered using the "Element Selection" window.
- The template function of ST programs cannot be used.
- Label suggestions are not displayed.
- The editing status cannot be restored to the previous one by performing operations such as selecting [Edit] → [Undo].

### ■Verifying a program containing an inline ST

When a project or project revision is verified by selecting "Program", an inline ST is not the verification target. To verify a program containing an inline ST, select "Program File".

### ■Finding or replacing items in a program containing an inline ST

The inside of an inline ST box is not the target of the following find/replace functions.

- [Replace String]
- [Find Device/Label]/[Replace Device/Label]
- [Find Instruction]/[Replace Instruction]
- [Find Contact or Coil]
- [Change Open/Close Contact]
- [Device Batch Replace]

When searching the inside of an inline ST box, use [Cross Reference] or [Device List].

### ■Copying an inline ST box

When copying an inline ST box, select a ladder block including the left side of the left rail. Copying only contacts in a ladder block containing an inline ST box or only an inline ST box is not allowed.

Copying a ladder block containing an unconverted inline ST box is also not allowed. Convert the program and then copy the ladder block.

### ■Jumping to an inline ST box during monitoring

When jumping to an inline ST box is performed by a search or verification operation during monitoring, the cursor will be moved to the target position but the range will not be selected.

# Appendix 7 Battery

Install a battery in the CPU module to hold data in the program memory, RAM, and latch devices at power-off. The following table lists the performance specifications of the battery used for the CPU module.

Item	Q6BAT	Q7BAT*1
Type	Manganese dioxide lithium primary battery	
Initial voltage	3.0V	
Nominal current	1800mAh	5000mAh
Battery life when not used	Approximately 5 years (room temperature)	
Lithium content	0.49g	1.52g
Application	To hold backup data and clock data	

\*1 The Q7BAT-SET includes a battery holder.

# Appendix 8 Real Number (Floating-point Data)

## Data size and data range

Real number data includes single-precision 32-bit real number data and double-precision 64-bit real number data.

Real number data can be stored only in devices other than bit devices or in single-precision or double-precision real data type labels.

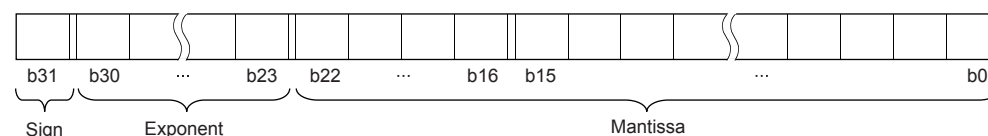
Data name		Data size	Value range
Single-precision real number data (single-precision floating-point data)	Positive number	32 bits (2 words)	$2^{-126} \leq \text{real number} < 2^{126}$
	Zero		0
	Negative number		$-2^{128} < \text{real number} \leq -2^{-126}$
Double-precision real number data (double-precision floating-point data)	Positive number	64 bits (4 words)	$2^{-1022} \leq \text{real number} < 2^{1024}$
	Zero		0
	Negative number		$-2^{1024} < \text{real number} \leq -2^{-1022}$

## Configuration of single-precision real number data

Single-precision real number data consists of a sign, mantissa, and exponent, and is expressed as shown below.

$$\boxed{\text{Sign}} \cdot 1. \boxed{\text{Mantissa}} \times 2^{\boxed{\text{Exponent}}}$$

The following figure shows the bit configuration of the internal expression of single-precision real number data and the meaning of each part.



### ■Sign (1 bit)

This bit represents the positive or negative sign of a numerical value. "0" indicates a positive number or 0. "1" Indicates a negative number.

### ■Mantissa (23 bits)

A mantissa means XXXXX... of  $1.XXXXX... \times 2^N$  representing a single-precision real number in binary.

### ■Exponent (8 bits)

An exponent means N of  $1.XXXXX... \times 2^N$  representing a single-precision real number in binary. The following table shows the relationships between the exponent value and N of a single-precision real number.

Exponent (b24 to b30)	FFH	FEH	FDH	...	81H	80H	7FH	7EH	...	02H	01H	00H
N	Not used	127	126	...	2	1	0	-1	...	-125	-126	Not used

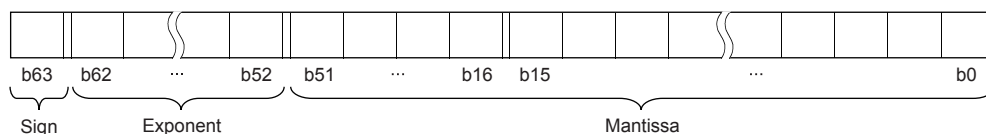
A

## Configuration of double-precision real number data

Double-precision real number data consists of a sign, mantissa, and exponent, and is expressed as shown below.



The following figure shows the bit configuration of the internal expression of double-precision real number data and the meaning of each part.



### ■Sign (1 bit)

This bit represents the positive or negative sign of a numerical value. "0" indicates a positive number or 0. "1" Indicates a negative number.

### ■Mantissa (52 bits)

A mantissa means  $\text{XXXXX} \dots$  of  $1.\text{XXXXX} \dots \times 2^N$  representing a single-precision real number in binary.

### ■Exponent (11 bits)

An exponent means  $N$  of  $1.\text{XXXXX} \dots \times 2^N$  representing a single-precision real number in binary. The following table shows the relationships between the exponent value and  $N$  of a single-precision real number.

Exponent (b52 to b62)	7FFH	7FEH	7FDH	...	401H	400H	3FFH	3FEH	...	02H	01H	00H
N	Not used	1023	1022	...	2	1	0	-1	...	-1021	-1022	Not used

## Precautions

### ■When setting an input value of single-precision real number from the engineering tool

The number of significant digits is about 7 because the engineering tool processes single precision real number data in 32-bit single precision.

When the input value of single-precision real number data exceeds 7 digits, the 8th digit is rounded off.

Therefore, if the rounded-off value goes out of the range from -2147483648 to 2147483647, it will not be an intended value.

**Ex.**

When "2147483647" is set as an input value, it is handled as "2147484000" because the 8th digit "6" is rounded off.

**Ex.**

When "E1.1754943562" is set as an input value, it is handled as "E1.175494" because the 8th digit "3" is rounded off.

### ■When setting an input value of double-precision real number from the engineering tool

The number of significant digits is about 15 because the engineering tool processes double precision real number data in 64-bit double precision.

When the input value of double-precision real number data exceeds 15 digits, the 16th digit is rounded off.

Therefore, if the rounded-off value goes out of the range from -2147483648 to 2147483647, it will not be an intended value.

**Ex.**

When "2147483646.12345678" is set as an input value, it is handled as "2147483646.12346" because the 16th digit "8" is rounded off.

**Ex.**

When "E1.7976931348623157+307" is set as an input value, it is handled as "E1.79769313486232+307" because the 16th digit "5" is rounded off.

The monitor function of the engineering tool can monitor real number data of CPU modules.

To represent "0" in real number data, set all numbers in each of the following range to 0.


- Single-precision real number data: b0 to b31
- Double-precision real number data: b0 to b63

The setting range of real number data is as follows. <sup>\*1</sup>

- Single-precision real number data:  $-2^{128} < [\text{single-precision real number data}] \leq -2^{126}$ ,  $0, 2^{126} \leq [\text{single-precision real number data}] < 2^{128}$
- Double-precision real number data:  $-2^{1024} < [\text{double-precision real number data}] \leq -2^{1022}$ ,  $0, 2^{1022} \leq [\text{double-precision real number data}] < 2^{1024}$

Do not specify "-0" (only the most significant bit is 1) in real number data. Performing a real number operation using -0 results in an operation error.

<sup>\*1</sup> For the operations to be performed when an overflow or underflow occurs or when a special value is input, refer to the following.

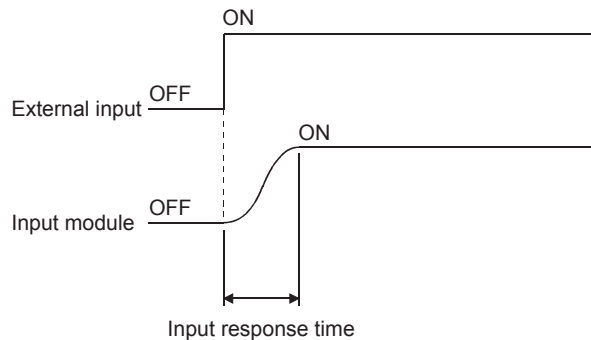
 MELSEC iQ-R CPU Module User's Manual (Application)



# Appendix 9 Changing Input Response Time

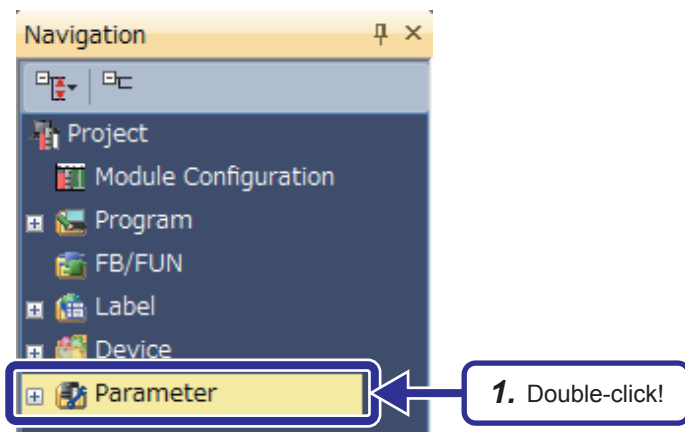
For an input module, high-speed input module, or I/O combined module, the input response time can be changed in the module parameters of GX Works3.

The input module obtains external inputs at a selected speed.

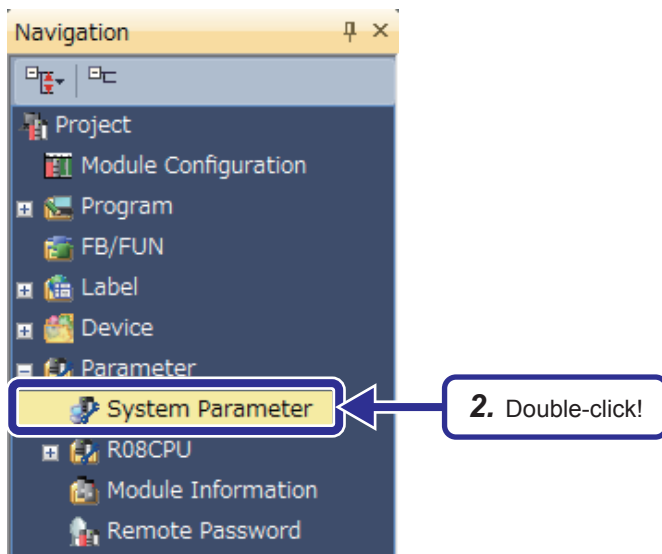


## Operating procedure

1. Double-click "Parameter" in the "Project" view.

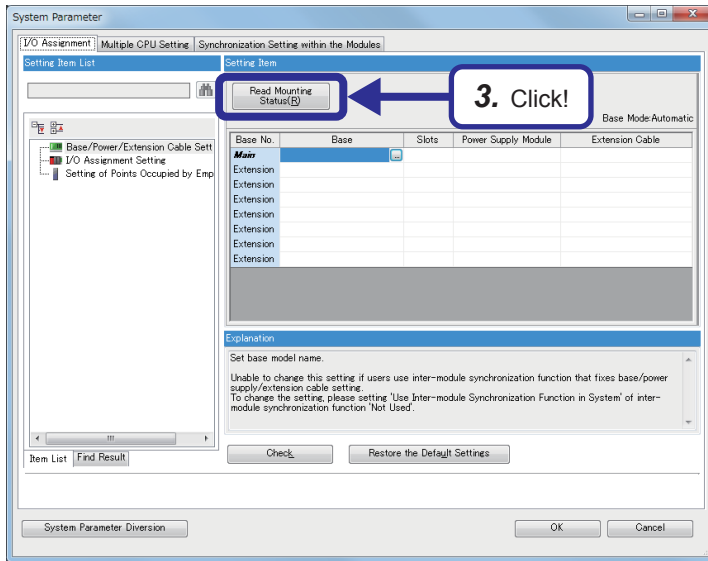


2. Double-click "System Parameter".

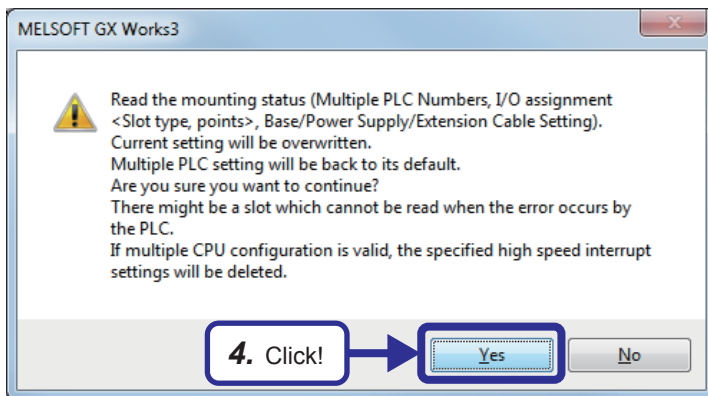


(To the next page)

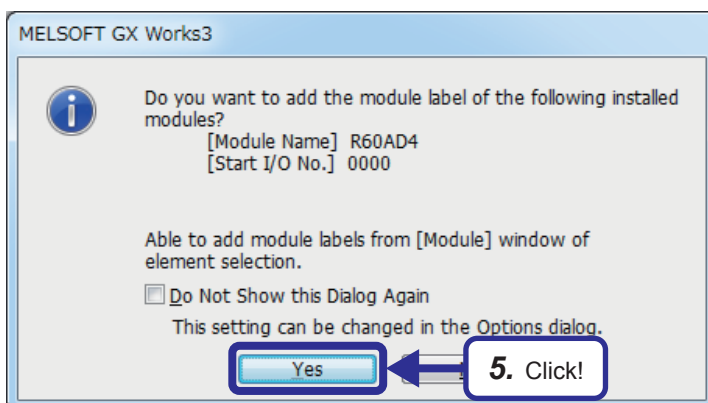
(From the previous page)



3. The "System Parameter" dialog box appears. Click the "Read Mounting Status" button.



4. The confirmation dialog box appears. Click the [Yes] button.

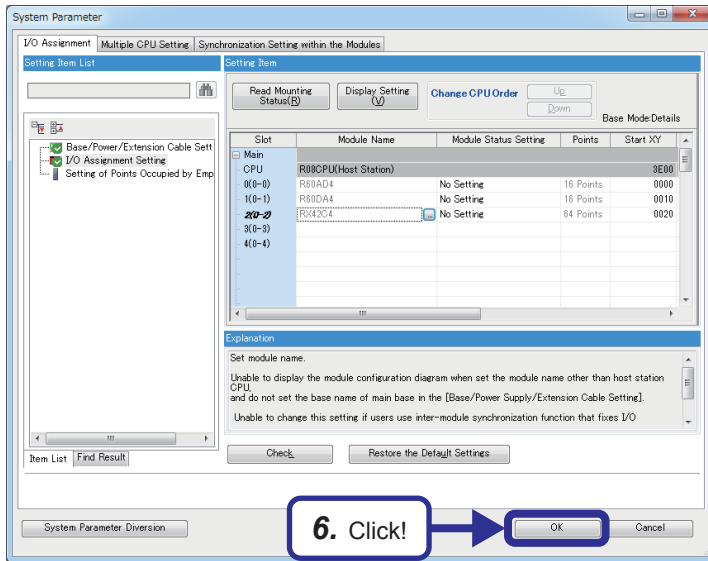


5. The confirmation dialog box for adding the module label of the mounted module appears. Click the [Yes] button.

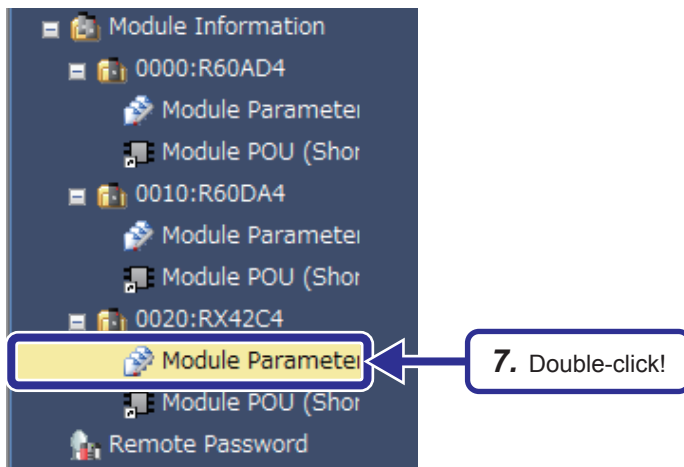


(To the next page)

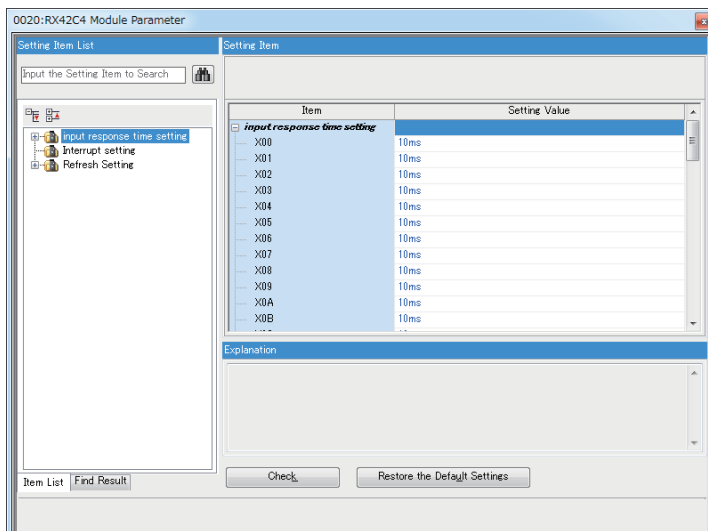
(From the previous page)



6. The mounting status is read and displayed on the "System Parameter" dialog box. Click the [OK] button.



7. Double-click "Module Parameter" of the RX42C4.



8. The "RX42C4 Module Parameter" dialog box appears. Configure the input response time setting.

# Mitsubishi Programmable Controllers Training Manual

## MELSEC iQ-R Series Advanced Course (for GX Works3)

MODEL	
MODEL CODE	
SH(NA)081897ENG-A (1602) MEE	



HEAD OFFICE : TOKYO BUILDING, 2-7-3 MARUNOUCHI, CHIYODA-KU, TOKYO 100-8310, JAPAN  
NAGOYA WORKS : 1-14 , YADA-MINAMI 5-CHOME , HIGASHI-KU, NAGOYA , JAPAN

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.