MITSUBISHI ELECTRIC

Changes for the Better

for a greener tomorrow

Programmable Controller C Controller Quick Start Guide

e-F@ctory

Let's start C Controller!

MELSEC iQ-R series

INTRODUCTION 1

USING C CONTROLLER MODULE 2

Preparing for Operation 2.1

Configuring the System 2.2

Setting the Module 2.3

Knowledge Required for Programming 2.4

Programming 2.5

Checking Operations 2.6

FREQUENTLY-USED FUNCTIONS 3

Checking Errors 3.1

Monitoring Module Status and Testing Operations 3.2

CONTENTS

RELEVANT MANUALS 2

CHAPTER 1 INTRODUCTION 3

CHAPTER 2 USING C CONTROLLER MODULE 5

2.1 **Preparing for Operation** 5
2.2 **Configuring the System** 6
System configuration example 6
Mounting the modules 7
Wiring the modules 8
Checking the power supply module. 9
2.3 **Setting the Module** 10
Initializing the C Controller module 10
Setting parameters 11
2.4 **Knowledge Required for Programming.** 19
C Controller module dedicated function 19
C Controller module dedicated function used in this guide 20
2.5 **Programming** 22
Program example and control description 22
Creating a project 25
Preparing a user program 34
2.6 **Checking Operations** 51
Enabling outputs (Y) from the user program 51
Checking operations with the dot matrix LED and lamps 52

CHAPTER 3 FREQUENTLY-USED FUNCTIONS 53

3.1 **Checking Errors.** 53
Checking an error 53
Checking error history 55
3.2 **Monitoring Module Status and Testing Operations** 58
Monitoring module status 58
Testing operations by forced output. 60

TRADEMARKS 62
Precautions before use 62
For safe use 62

RELEVANT MANUALS

This guide explains the basic operations of a C Controller module.

To make maximum use of a C Controller module, refer to the following manuals according to a purpose.

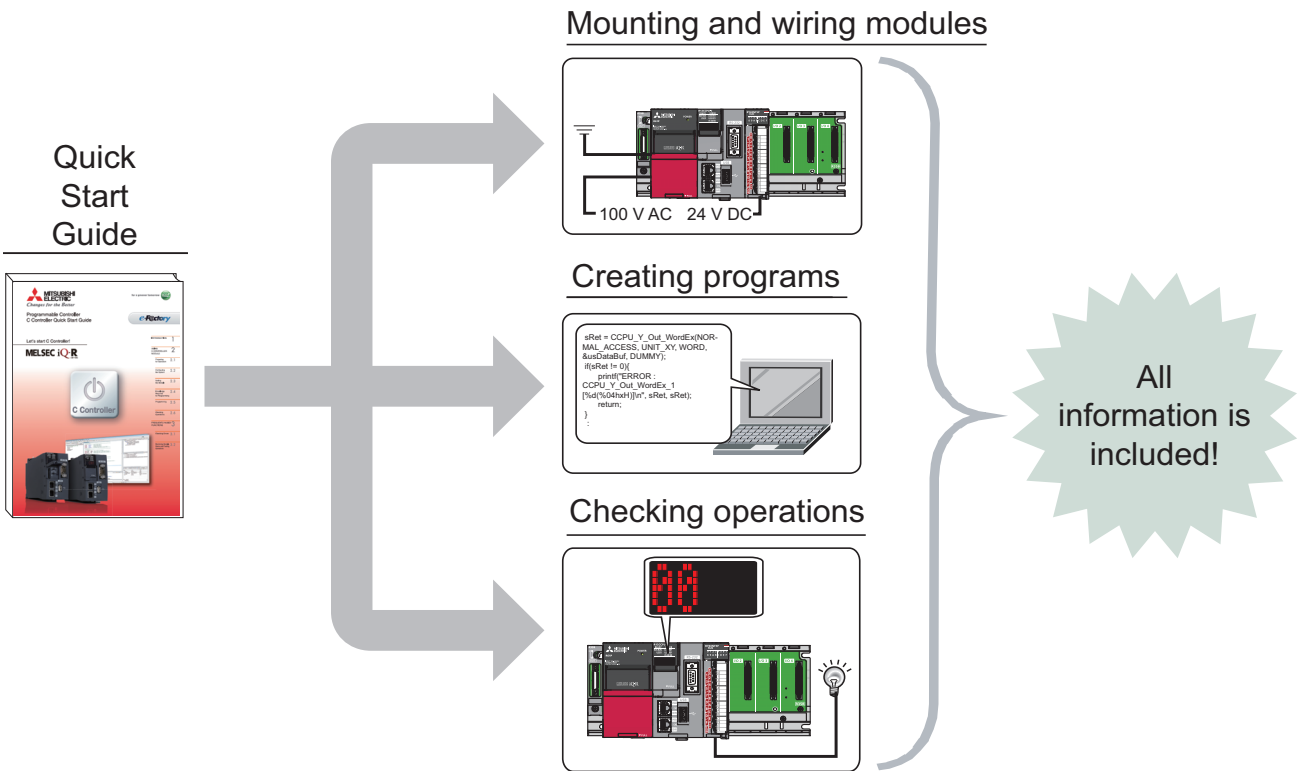
Manual name [manual number]	Description	Available form
MELSEC iQ-R C Controller Module User's Manual (Startup) [SH-081367ENG]	Explains the performance specifications, procedure before operation, and troubleshooting of a C Controller module.	Print book e-Manual PDF
MELSEC iQ-R C Controller Module User's Manual (Application) [SH-081369ENG]	Explains the functions, devices, and parameters of a C Controller module.	Print book e-Manual PDF
MELSEC iQ-R C Controller Module Programming Manual [SH-081371ENG]	Explains the programming specifications and dedicated function library of a C Controller module.	e-Manual PDF
CW Workbench/CW-Sim Operating Manual [SH-081373ENG]	Explains the system configuration, specifications, functions, and troubleshooting of CW Workbench/CW-Sim.	e-Manual PDF
CW Configurator Operating Manual [SH-081382ENG]	Explains the system configuration, parameter settings, and operation methods for the online function of CW Configurator.	e-Manual PDF

1 INTRODUCTION

This guide simply explains the basic operations of a C Controller module for the first-time users of the Mitsubishi Electric programmable controller, MELSEC iQ-R series C Controller module R12CCPU-V (hereafter abbreviated as a C Controller module).

This guide is targeted for users who use the MELSEC iQ-R series for the first time and are in the following situations:

- Users with experience in C language or C++ language programming
- Users considering to replace the microcomputer board or the personal computer system with a C Controller system



Point

- For the safety precautions for using a C Controller module, refer to the following manual.
  (MELSEC iQ-R C Controller Module User's Manual (Startup))
- This guide explains the settings in Windows 7.

Precautions

This guide explains operations using the system configuration in 'Page 6 System configuration example'.

When designing/operating a system, be sure to read the manuals listed in the following section.

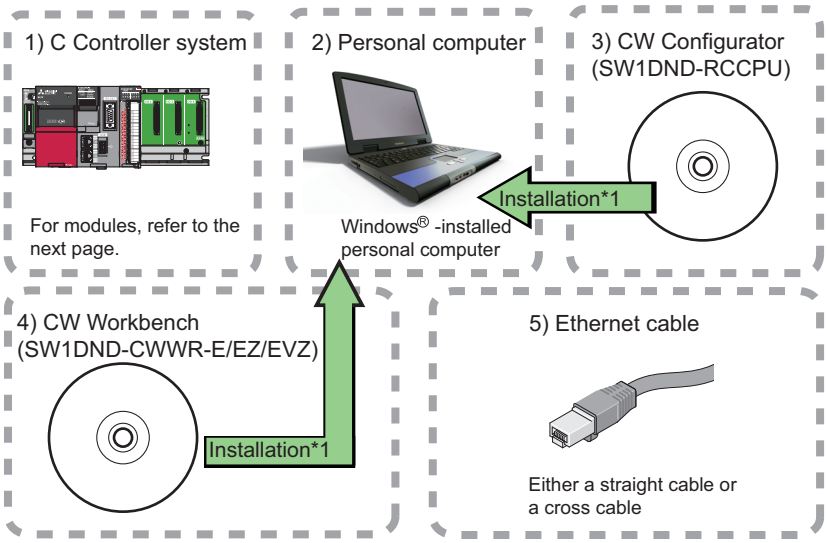
☞ RELEVANT MANUALS

MEMO

2 USING C CONTROLLER MODULE

2.1 Preparing for Operation

Prepare necessary devices.



*1 Install CW Configurator and CW Workbench on the same personal computer beforehand.

Point

For installation of CW Configurator, refer to the following manual.

CW Configurator Installation Instructions

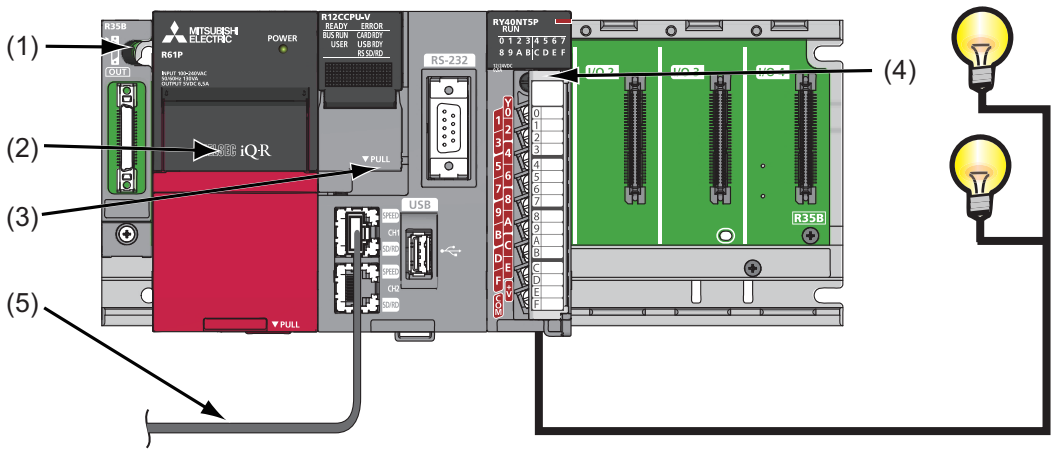
For installation of CW Workbench, refer to the following manual.

CW Workbench/CW-Sim Operating Manual

2.2 Configuring the System

System configuration example

This guide uses the following system configuration as an example.



No.	Name	Model name	Description
(1)	Base unit	R35B	A unit to mount a power supply module, a C Controller module, and I/O modules.
(2)	Power supply module	R62P	A module which supplies power to modules, such as a C Controller module and I/O modules.
(3)	C Controller module	R12CCPU-V	A module which unifies control process of a C Controller system.
(4)	Output module	RY40NT5P	—
(5)	Cable (Ethernet cable)	An Ethernet cable that meets 10BASE-T/100BASE-TX/ 1000BASE-T standards	A cable which connects a personal computer in which CW Configurator and CW Workbench are installed and a C Controller module.

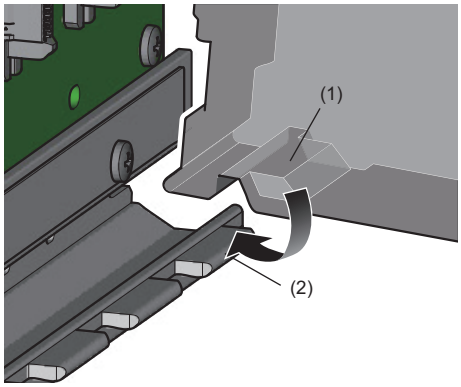
Mounting the modules

Mount the prepared modules on a base unit.

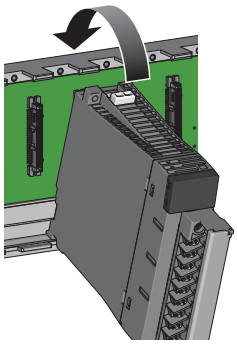
Precautions

Before mounting modules, be sure to power OFF the system.

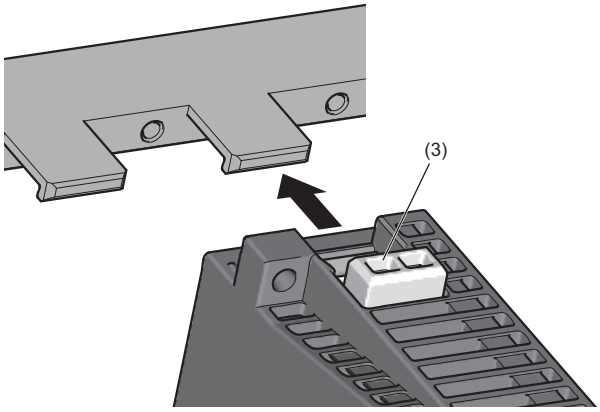
Mounting procedure



1. When a cap is attached to the module connector on the base unit, remove it.
2. Place the concave part (1) of the module onto the guide (2) of the base unit.



3. Push the module to the direction of the allow until the module fixing hook (3) clicks to mount the module to the base unit.



4. Check that the module fixing hook (3) is fixed and the module is mounted securely on the base unit.

Wiring the modules

Wire the power supply module.

Precautions

Before wiring modules, be sure to power OFF the system.

Point

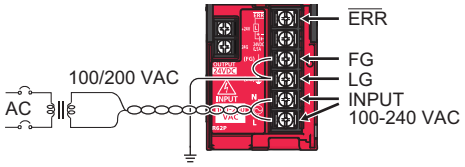
For wiring precautions, refer to the following manual.

MELSEC iQ-R Module Configuration Manual

1. Wiring the power supply module.

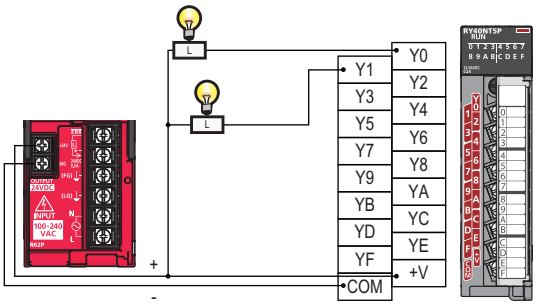
The following shows an example of wiring the power wire and the ground wire to the power supply module.

Grounding should be provided to prevent electric shock and malfunction.



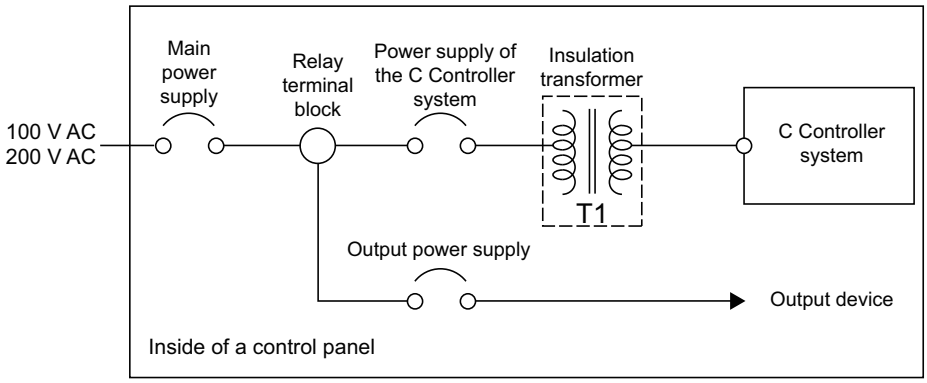
2. Wiring the output module.

The following shows an example of wiring the output module RY40NT5P.



Point

The power supply line of the output device and that of the C Controller system should be wired separately as shown below.



Checking the power supply module

After installing the system, mounting the modules, and wiring the system, check that the power supply module works properly.

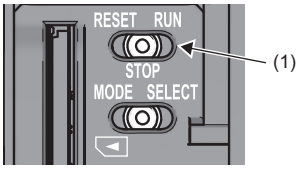
Operating procedure

1. Check the following before powering ON the system.

- Wiring to the power supply module
- Power supply voltage

2. Set the C Controller module to STOP.

Open the cover on the front of the C Controller module and set the RESET/STOP/RUN switch (1) to STOP.



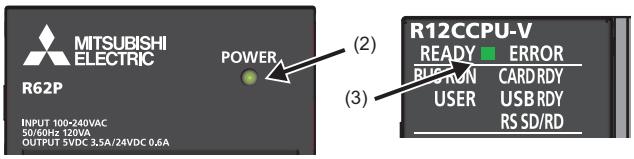
3. Power ON the power supply module.

4. Check that the power supply module runs normally.

Check the front LED on each module.

The normal status of each LED is as follows.

- Power supply module: The POWER LED (2) lights in green.
- C Controller module: The READY LED (3) lights in green.



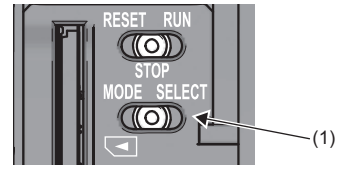
Point

If the POWER LED of the power supply module remains OFF even after power-on, check that the power supply module is correctly wired and mounted.

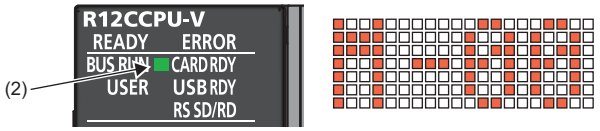
2.3 Setting the Module

Initializing the C Controller module

Check that the RESET/STOP/RUN switch is positioned at the center at first.

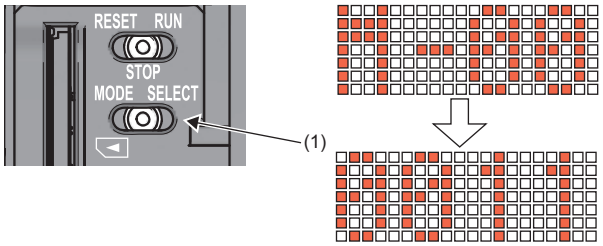


1. Put and hold the MODE/SELECT switch (1) on the MODE position.

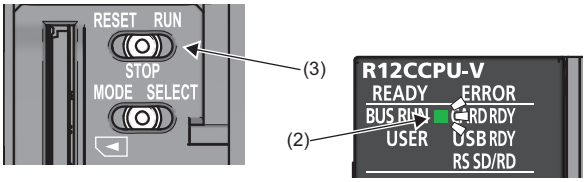


2. Turn the power of the C Controller module ON. The BUS RUN LED (2) turns ON, and "M-00" is displayed on the dot matrix LED.

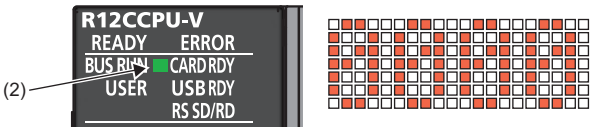
3. Release the MODE/SELECT switch (1) and put it back to the center position.



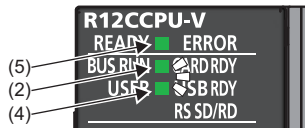
4. Set the MODE/SELECT switch (1) to the SELECT position. Every time the switch is set to the SELECT position, the value of mode displayed on the dot matrix LED is changed. Repeat this switch movement until "0011" is displayed on the dot matrix LED.



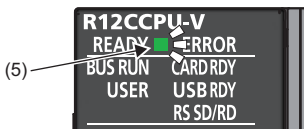
5. Set the RESET/STOP/RUN switch (3) to the RUN position. The selected mode is executed. The BUS RUN LED (2) will be flashing during initialization.



6. Check that the BUS RUN LED (2) turns ON and "0000" is displayed on the dot matrix LED, then reset the C Controller module. For the reset procedure, refer to the following section.
(☞ Page 11 Resetting procedure)



7. The initialization is performed by resetting the module. The READY LED (5) will turn ON, and the BUS RUN LED (2) and USER LED (4) will be flashing during the initialization.

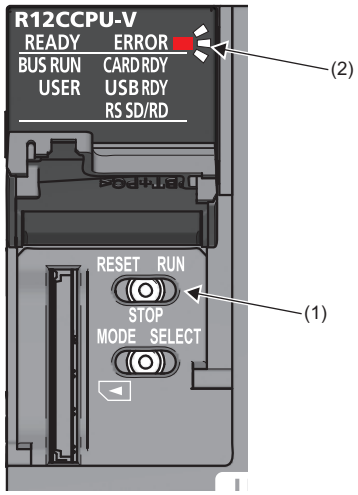


8. Upon normal completion of the initialization, the BUS RUN LED (2) and USER LED (4) turns OFF, and the READY LED (5) starts flashing.

9. Reset the C Controller module. For the procedure, refer to the following section.
(☞ Page 11 Resetting procedure)

Resetting procedure

C Controller module can be reset by operating the switch by the following procedure.



1. Hold the RESET/STOP/RUN switch (1) in the RESET position.
2. Check that all LEDs turn OFF after the ERROR LED (2) flashes several times.
3. Release the RESET/STOP/RUN switch (1) and put it back to the STOP position.

Point

Do not reset the C Controller module during initialization.
Perform the initialization again if the module has been reset in error.

Setting parameters

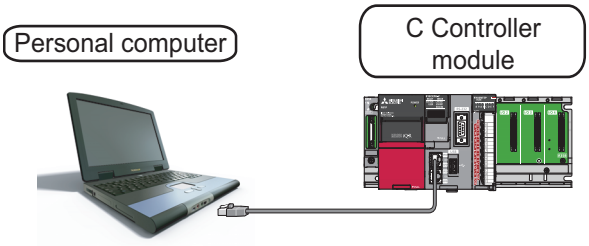
Set parameters for the C Controller module.

Point

Parameter: Setting data required for a C Controller system to operate. Set modules and a network in CW Configurator for the C Controller module.

Connecting the C Controller module to the personal computer

Connect the CH1 on the C Controller module to the personal computer using an Ethernet cable.



Precautions

The IP address of the C Controller module and that of the personal computer must be set to the same segment. Since this guide uses the default IP address for the C Controller module (192.168.3.3), set the IP address for the personal computer to '192.168.3.* (*: except for 0, 3, and 255)'.
Set the subnet mask for the personal computer to '255.255.255.0'.

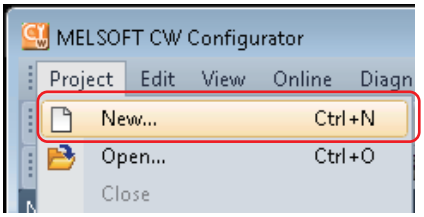
Point

For how to change an IP address, refer to the following manual.
MELSEC iQ-R C Controller Module User's Manual (Application)

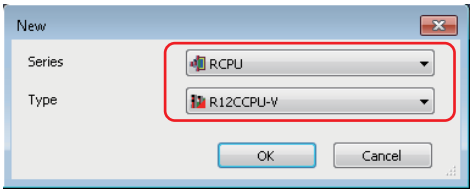
Starting CW Configurator to create a project

Operating procedure

1. Select [Start] ⇨ [All Programs] ⇨ [MELSOFT] ⇨ [CW Configurator] ⇨ [CW Configurator].
2. Select [Project] ⇨ [New].



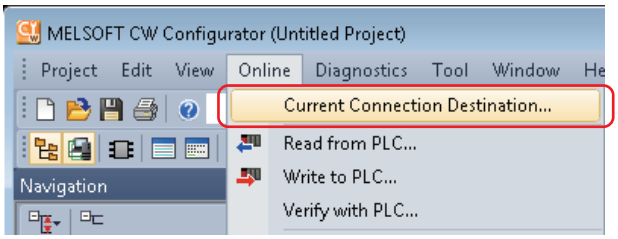
3. Check that "RCPU" and "R12CCPU-V" are selected, and click the [OK] button.



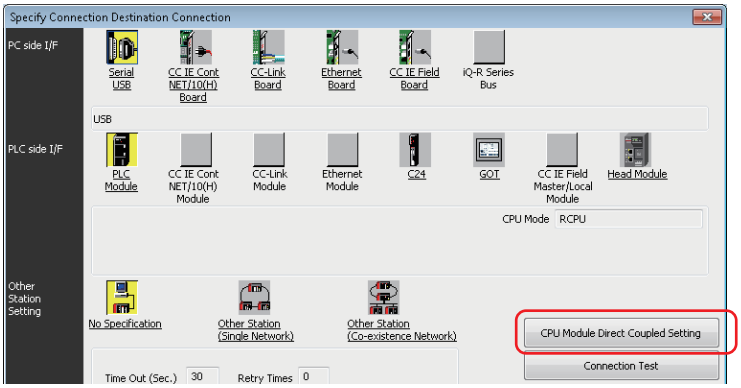
Establishing communication with the C Controller module

Operating procedure

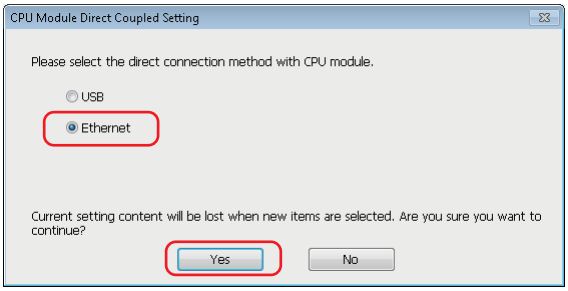
1. Select [Online] ⇨ [Current Connection Destination] from the menu of CW Configurator.



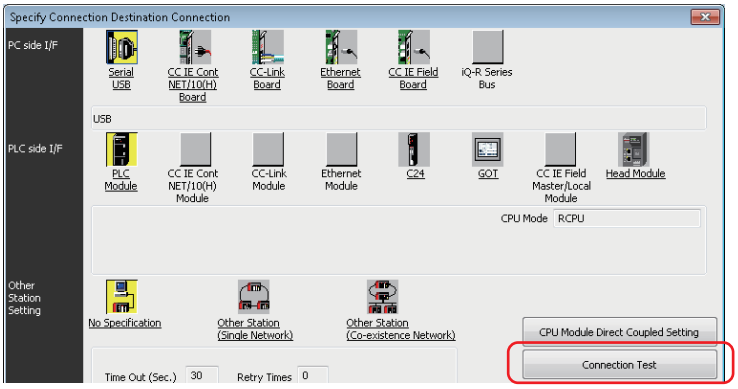
2. Click the [CPU Module Direct Coupled Setting] button in the "Specify Connection Destination" screen.

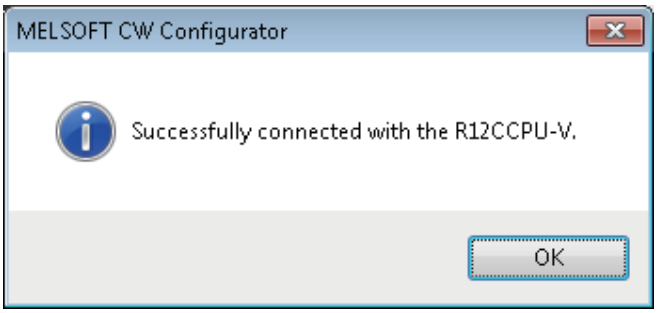


3. Select "Ethernet", and click the [Yes] button.



4. Click the [Connection Test] button, and check that the message "Successfully connected with the R12CCPU-V." appears.



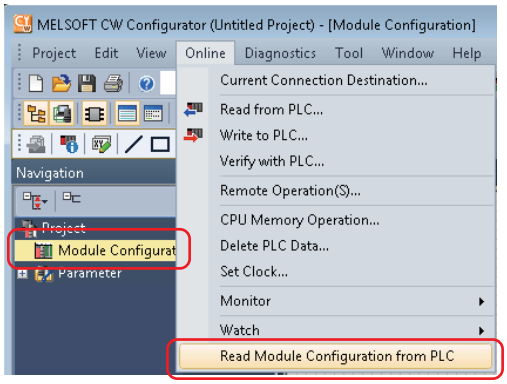


Setting parameters

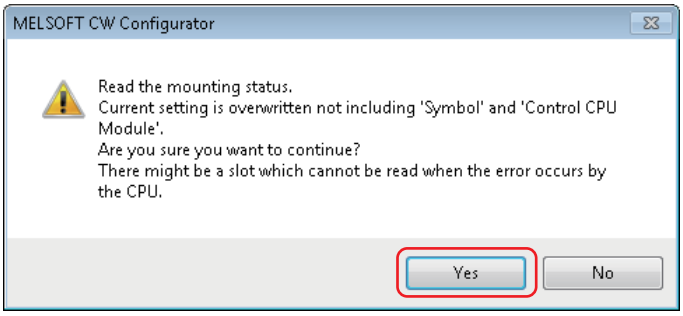
Set the parameters for the system and modules.

Operating procedure

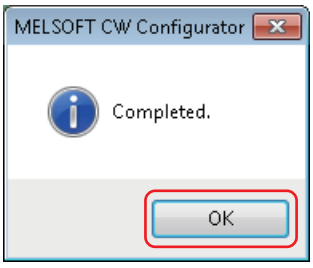
1. Double-click "Module Configuration" in the "Navigation" window to open, and select [Online] ⇨ [Read Module Configuration from PLC].



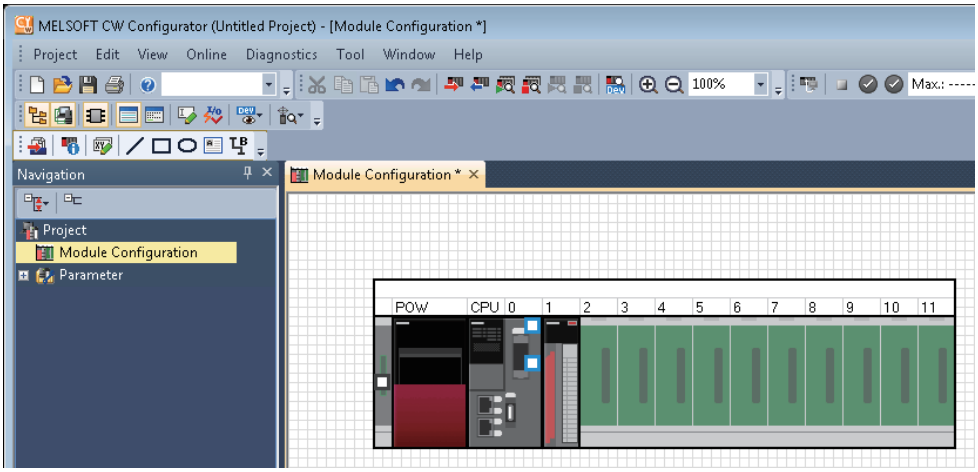
2. When the following message appears, click the [Yes] button.



3. Click the [OK] button when the following message appears after reading is completed.



4. The system parameters are automatically set, and the actual system configuration is displayed in the "Module Configuration" window.

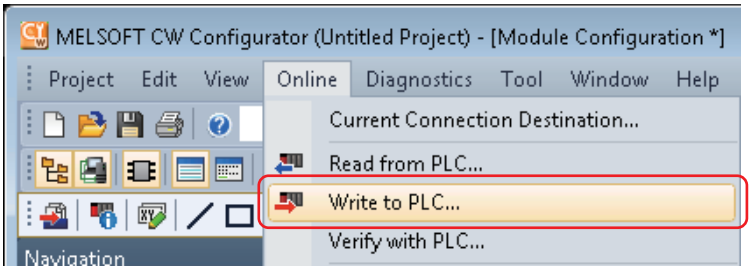


Writing parameters to the C Controller module

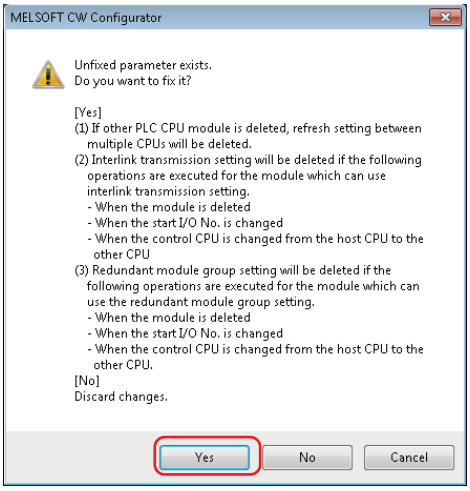
Write the parameters to the C Controller module using CW Configurator.

Operating procedure

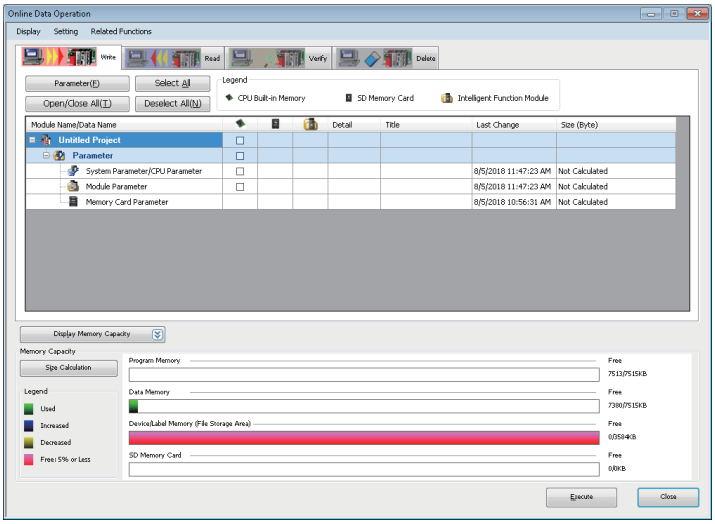
1. Select [Online] ⇨ [Write to PLC].



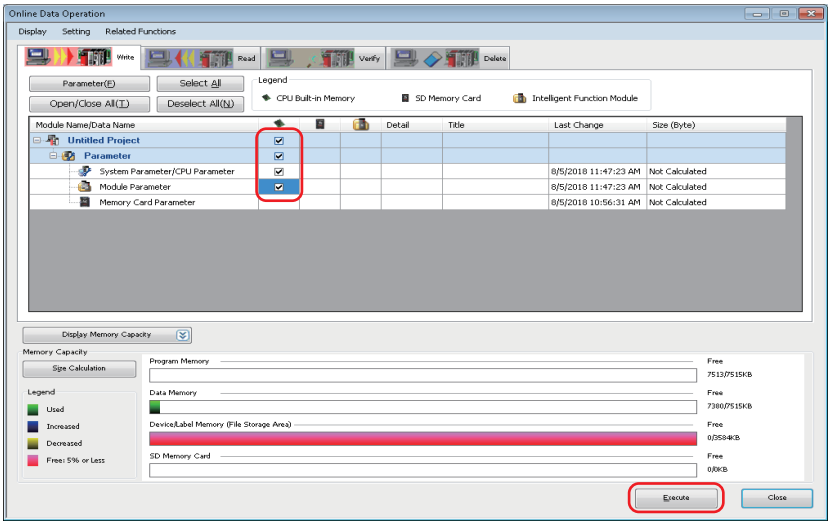
2. Click the [Yes] button.



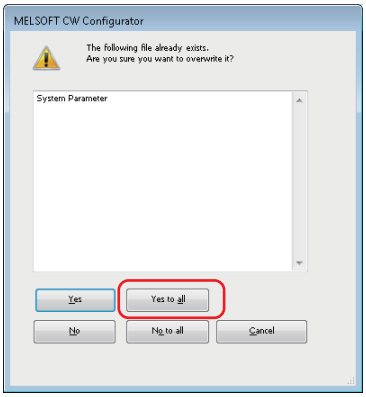
3. Check that the "Online Data Operation" screen appears.



4. Select "System Parameter/CPU Parameter" and "Module Parameter", and click the [Execute] button.

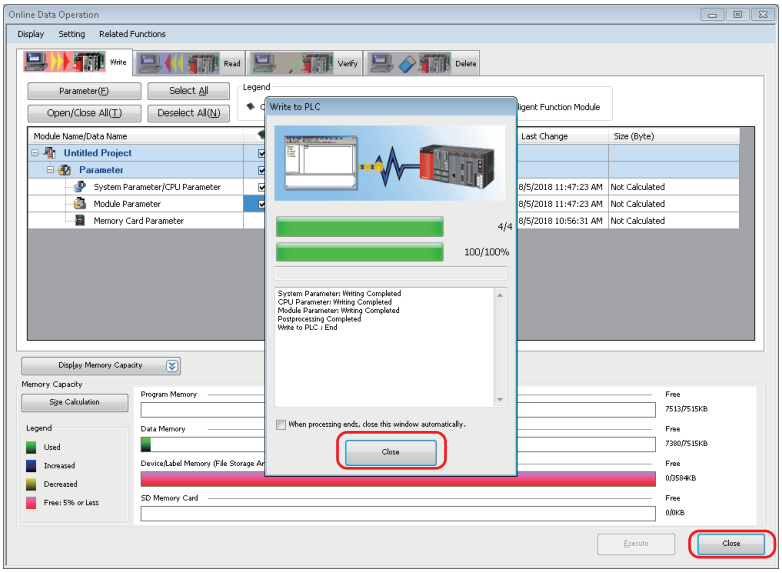


5. Click the [Yes to all] button.



Writing the parameters starts.

6. Click the [Close] button when writing to the C Controller module is completed.



2.4 Knowledge Required for Programming

C Controller module dedicated function

C Controller module dedicated function is one of the 'dedicated function libraries' for C Controller modules. Using this function in a user program allows a C Controller module to easily control MELSEC iQ-R series modules.

I/O access

1-point access and 1-word access are available.

■Bit access: A function that treats 1-point data (ON/OFF of switches and lamps)

Example of bit access functions

Function name	Description
CCPU_X_In_BitEx	To read an input signal (X) in bit (1-point) units.
CCPU_Y_Out_BitEx	To output an output signal (Y) in bit (1-point) units.
CCPU_Y_In_BitEx	To read an output signal (Y) in bit (1-point) units.

■Word access: A function that treats 16-point (1-word) data (numeric values, characters)

Example of 1-word access functions

Function name	Description
CCPU_X_In_WordEx	To read an input signal (X) in word (16-point) units.
CCPU_Y_Out_WordEx	To output an output signal (Y) in word (16-point) units.
CCPU_Y_In_WordEx	To read an output signal (Y) in word (16-point) units.

User LED control

User LED control is a method for controlling the USER LED and the dot-matrix LED on the front of a C Controller module.

Example of user LED control functions

Function name	Description
CCPU_SetLEDStatus	To set the LED status of a C Controller module.
CCPU_SetDotMatrixLED	To set a value to be displayed on the dot matrix LED of C Controller module.

Point

Only the basic C Controller module dedicated functions are explained in this section.
Other than the functions above, C Controller module dedicated functions and MELSEC communication functions are available for controlling modules. For each function, refer to the following manual.
MELSEC iQ-R C Controller Module Programming Manual

C Controller module dedicated function used in this guide

Create a program using basic C Controller module dedicated functions (output access and dot matrix LED control).

Output access: CCPU_Y_Out_WordEx function

■Format

short CCPU_Y_Out_WordEx (short sFlg, unsigned, short usYNo, unsigned short usSize, unsigned short* pusDataBuf, unsigned short usBufSize);

■Argument

Argument	Name	Description	IN/OUT
sFlg	Access flag	Specify an access flag. • 0: Normal access • Others: Reserved	IN
usYNo	Start output signal	Specify a start output signal (Y). (Specify a multiple of 16.)	IN
usSize	Output size	Specify the output size in word units.	IN
pusDataBuf	Data storage destination	Specify the storage destination of output data.	IN
usBufSize	Data storage destination size	Specify '0'.	IN

Dot matrix LED control: CCPU_SetDotMatrixLED function

■Format

short CCPU_SetDotMatrixLED(unsigned short usLedMode, char* pcData);

■Argument

Argument	Name	Description	IN/OUT
usLedMode	Output mode	Specify the output mode to the dot matrix LED. (When 'Reserved' is specified, this function ends normally without processing.) • 0: Dot mode • 1: ASCII mode • Others: Reserved	IN
pcData	LED data	Specify the LED data.	IN

• Mode 1: ASCII mode

The character string specified for pcData[0] to pcData[3] is displayed.

Available characters (ASCII code) are shown below.

×: Character string specification not allowed

Bit		Upper bit															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Lower bit	0	×	×	SP	0	×	P	×	×	×	×	×	×	×	×	×	×
	1	×	×	×	1	A	Q	×	×	×	×	×	×	×	×	×	×
	2	×	×	×	2	B	R	×	×	×	×	×	×	×	×	×	×
	3	×	×	×	3	C	S	×	×	×	×	×	×	×	×	×	×
	4	×	×	×	4	D	T	×	×	×	×	×	×	×	×	×	×
	5	×	×	%	5	E	U	×	×	×	×	×	×	×	×	×	×
	6	×	×	×	6	F	V	×	×	×	×	×	×	×	×	×	×
	7	×	×	×	7	G	W	×	×	×	×	×	×	×	×	×	×
	8	×	×	×	8	H	X	×	×	×	×	×	×	×	×	×	×
	9	×	×	×	9	I	Y	×	×	×	×	×	×	×	×	×	×
	A	×	×	×	×	J	Z	×	×	×	×	×	×	×	×	×	×
	B	×	×	×	×	K	×	×	×	×	×	×	×	×	×	×	×
	C	×	×	×	×	L	×	×	×	×	×	×	×	×	×	×	×
	D	×	×	-	×	M	×	×	×	×	×	×	×	×	×	×	×
	E	×	×	.	×	N	×	×	×	×	×	×	×	×	×	×	×
	F	×	×	/	×	O	×	×	×	×	×	×	×	×	×	×	×

When a character other than above is specified, an error will be returned.

If a NULL is included in a character string, data after the NULL is not displayed (blank). (They are displayed with left-aligned.)

Point

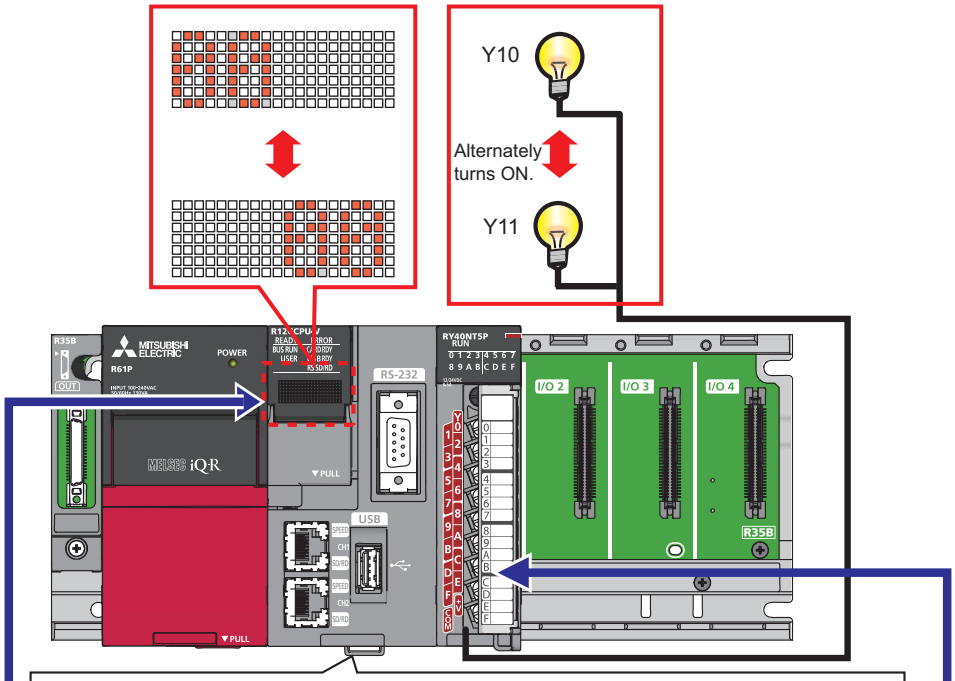
In this guide, ASCII mode (1) is specified for the output mode (usLedMode).

2.5 Programming

Create a program in which lamps connected to an output module and the dot matrix LED on the front of the C Controller module flash.

Program example and control description

When the C Controller module is set to RUN, the output lamps Y10 and Y11 alternately turn ON.
Synchronizing with the output lamps, the dot matrix LED on the front of the C Controller module switches alternately between "00__" and "__00".




```
/* Output control */
sRet = CCPU_Y_Out_WordEx(NORMAL_ACCESS, UNIT_XY, WORD, &usDataBuf, DUMMY);
if(sRet != 0){
    printf("ERROR : CCPU_Y_Out_WordEx_1 [%d(%04hxH)]\n", sRet, sRet);
    return;
}

/* Dot matrix LED control */
sRet = CCPU_SetDotMatrixLED(DISPMODE_ASCII, &pcdata[0]);
if(sRet != 0){
    printf("ERROR : CCPU_SetDotMatrixLED_1 [%d(%04hxH)]\n", sRet, sRet);
    return;
}
```

Source code

The following describes source codes.

```
/************************************************************************************/
/* Function header */
/************************************************************************************/
#include        <vxworks.h>     /* VxWorks function header */
#include        <taskLib.h>     /* VxWorks function header */
#include        <stdio.h>       /* Standard function header */
#include        <string.h>      /* Standard function header */
#include        "CCPUFunc.h"    /* C Controller module dedicated function header */
/************************************************************************************/
/* Definition */
/************************************************************************************/
/* For debugging */
#define UNIT_XY         0x0010 /* Start I/O number of the module */
#define RY_LED          0x5555 /* Initial output value of Y signals (even bit: on) */
#define LED_0           0x30    /* Initial output value on the dot matrix LED (LED1,2) */
#define LED_SPACE       0x20    /* Initial output value on the dot matrix LED (LED3,4) */

/* For C Controller module dedicated function */
#define WORD                    1 /* 1-word specification */
#define NORMAL_ACCESS           0 /* General access specification */
#define DUMMY                   0 /* Dummy */
#define DISPMODE_ASCII          1 /* Dot matrix LED output mode */
/************************************************************************************/
/* Processing to output Y signals and to control dot matrix LED */
/************************************************************************************/
void R12_SampleTask()
{
        /* Declare local variables. */
        short           sRet;           /* Return value of the C Controller module dedicated function */
        unsigned short  usDataBuf;              /* For accessing Y signals */
        unsigned short  usEmptyDataBuf;         /* For resetting Y signals */
        char            pcdata[4];              /* Value on the dot matrix LED */
        short           i;                      /* For loop */

        /* Set the output value of Y signals (turn on the even bit). */
        usDataBuf = RY_LED;

        /* Set the output value of the dot matrix LED (LED1,2: on). */
        pcdata[0] = LED_0;
        pcdata[1] = LED_0;
        pcdata[2] = LED_SPACE;
        pcdata[3] = LED_SPACE;

        /* Perform an output control and dot matrix LED control in turns for 20 times. */
        for(i = 0; i < 20; i++){
                /* Output control */
                sRet = CCPU_Y_Out_WordEx(NORMAL_ACCESS, UNIT_XY, WORD, &usDataBuf, DUMMY);
                if(sRet != 0){
                        printf("ERROR : CCPU_Y_Out_WordEx_1 [%d(%04hxH)]\n", sRet, sRet);
                        return;
                }
```

```
                /* Dot matrix LED control */
                sRet = CCPU_SetDotMatrixLED(DISPMODE_ASCII, &pcdata[0]);
                if(sRet != 0){
                        printf("ERROR : CCPU_SetDotMatrixLED_1 [%d(%04hxH)]\n", sRet, sRet);
                        return;
                }

                /* Invert the output value of Y signals (turn on the bits in order of odd bit -> even bit -> ...). */
                usDataBuf = ~usDataBuf;

                /* Switch the output of the dot matrix LED (LED1,2: on -> LED3,4: on). */
                if(i%2 ==0){
                        pcdata[0] = LED_SPACE;
                        pcdata[1] = LED_SPACE;
                        pcdata[2] = LED_0;
                        pcdata[3] = LED_0;
                }else{
                        pcdata[0] = LED_0;
                        pcdata[1] = LED_0;
                        pcdata[2] = LED_SPACE;
                        pcdata[3] = LED_SPACE;
                }
                /* Wait */
                taskDelay(60);
        }

        /* Reset Y signals. */
        usEmptyDataBuf = 0x00;
        sRet = CCPU_Y_Out_WordEx(NORMAL_ACCESS, UNIT_XY, WORD,&usEmptyDataBuf, DUMMY);
        if(sRet != 0){
                printf("ERROR : CCPU_Y_Out_WordEx_2 [%d(%04hxH)]\n", sRet, sRet);
                return;
        }

        /* Reset the dot matrix LED. */
        pcdata[0] = LED_SPACE;
        pcdata[1] = LED_SPACE;
        pcdata[2] = LED_SPACE;
        pcdata[3] = LED_SPACE;
        sRet = CCPU_SetDotMatrixLED(DISPMODE_ASCII, &pcdata[0]);
        if(sRet != 0){
                printf("ERROR : CCPU_SetDotMatrixLED_2 [%d(%04hxH)]\n", sRet, sRet);
                return;
        }
}
```

Creating a project

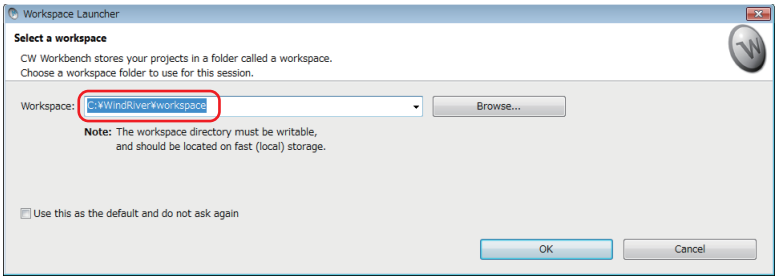
Starting CW Workbench

Operating procedure

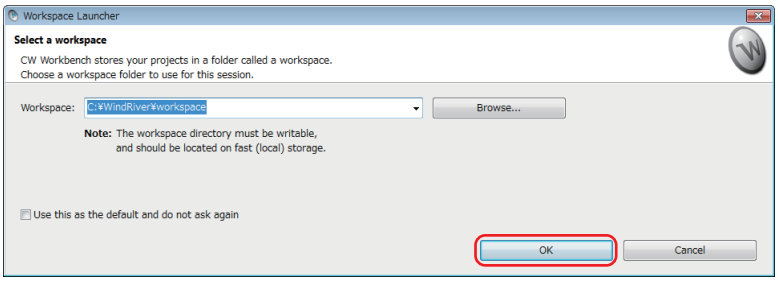
1. Select [Start] ⇨ [All Programs] ⇨ [Wind River] ⇨ [CW Workbench 3.3] ⇨ [CW Workbench 3.3].

2. After CW Workbench is started, enter the save destination folder for a project.

In this procedure, enter 'C:\WindRiver\workspace'.



3. Click the [OK] button.



The main window of CW Workbench appears.

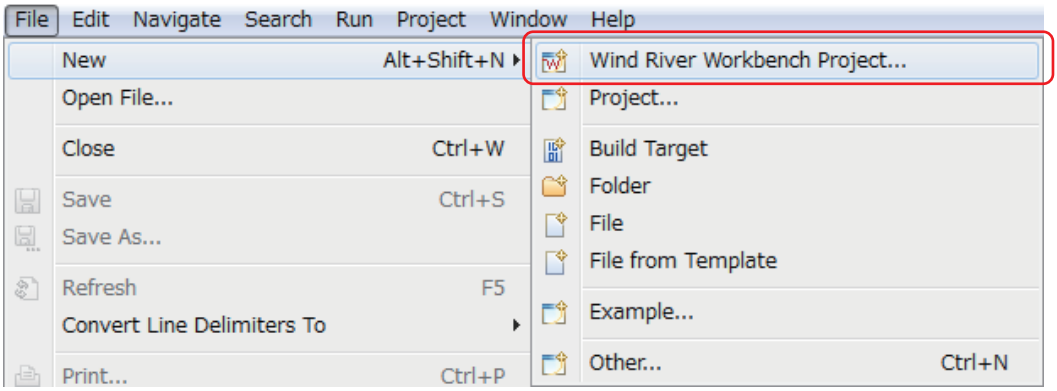
Point

- The default window sizes and icon positions on CW Workbench depends on a personal computer. If a window size differs from that shown in this guide, adjust the size.
- To default an enlarged/deleted window, select [Window] ⇨ [New Window].

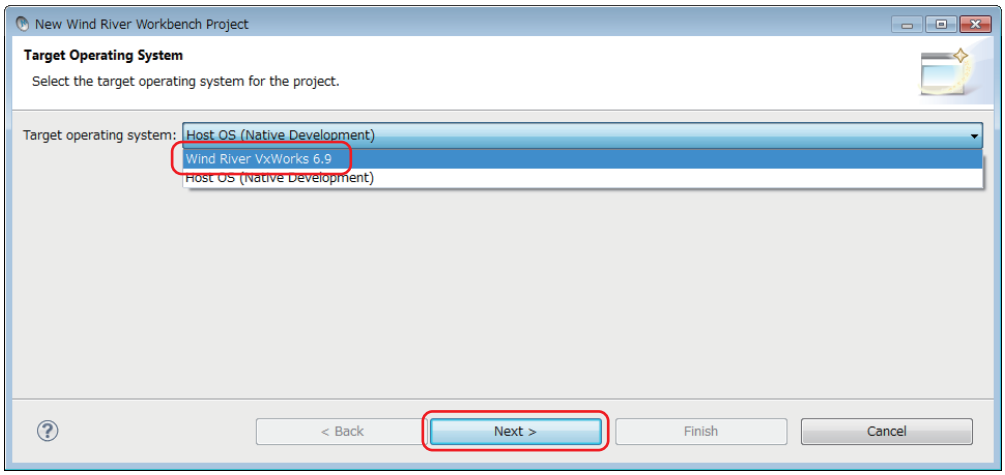
Creating a new project

Operating procedure

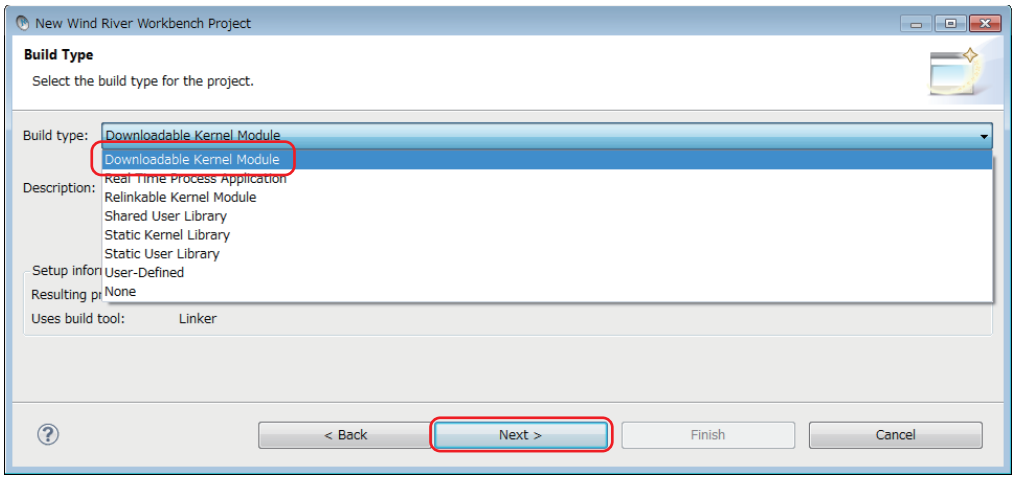
1. Select [File] ⇨ [New] ⇨ [Wind River Workbench Project].



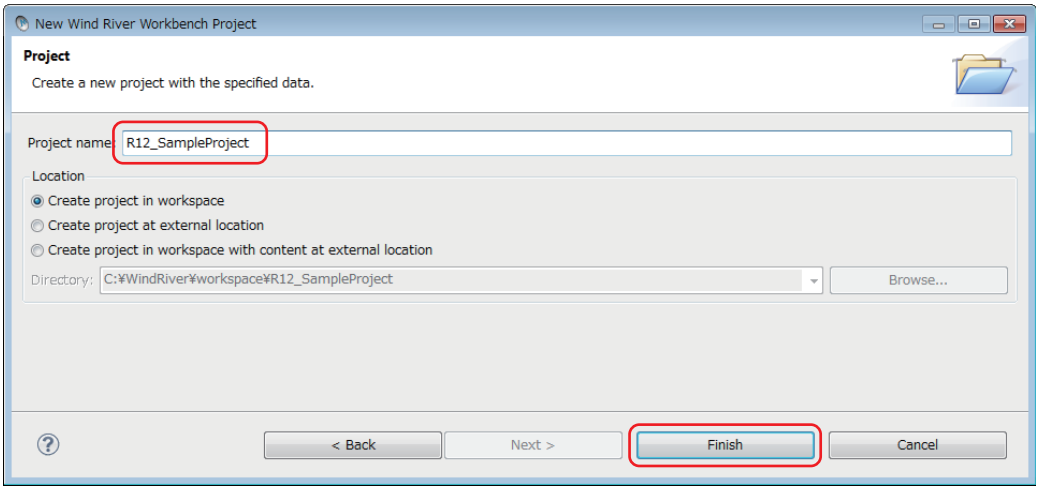
2. Select "Wind River VxWorks 6.9", and click the [Next] button.



3. Select "Downloadable Kernel Module", and click the [Next] button.



4. Enter a project name, and click the [Finish] button.
(Project name: 'R12_SampleProject' in this chapter)



Setting properties of the project

Configure the settings to convert (build) the created project into a module that can be executed on the C Controller module.

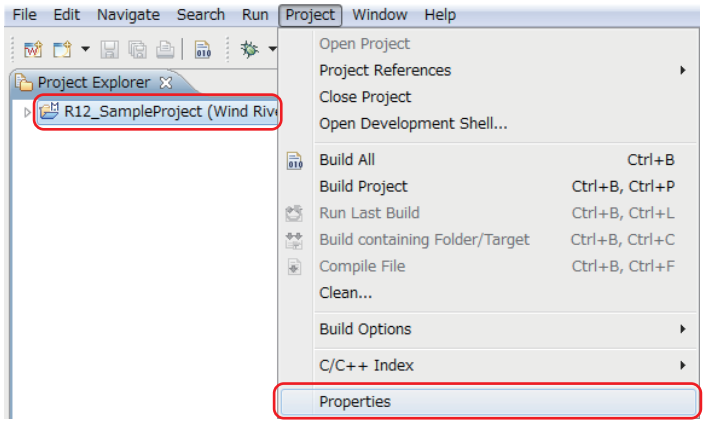
Point

Build: An operation that compiles source codes according to a processor and links the code to the include file.

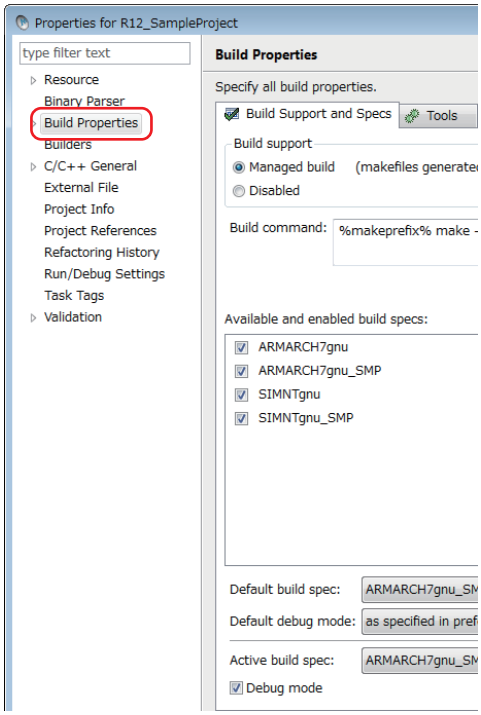
■Setting a processor

Operating procedure

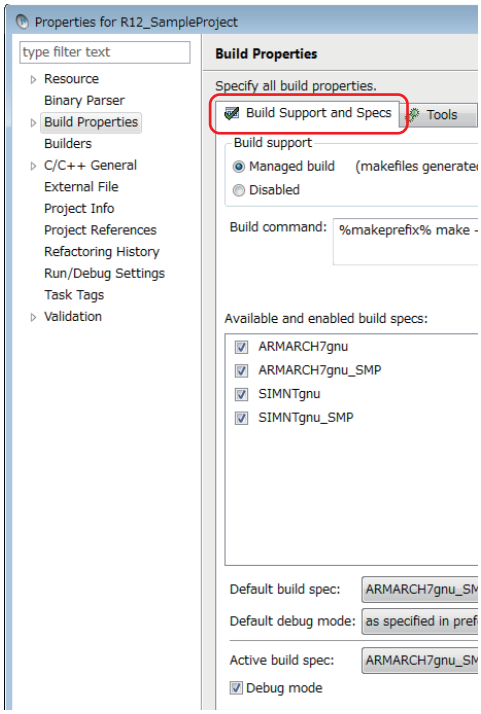
1. Select the created project in the "Project Explorer" window, and click [Project] ⇨ [Properties].



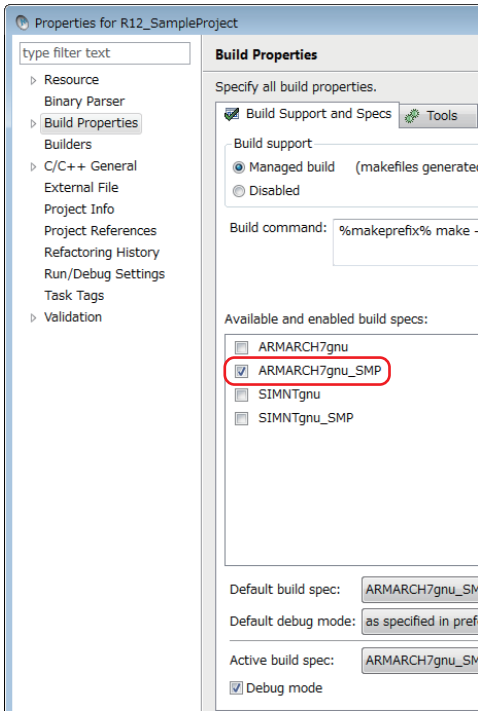
2. Select "Build Properties" from the tree to the left in the screen.



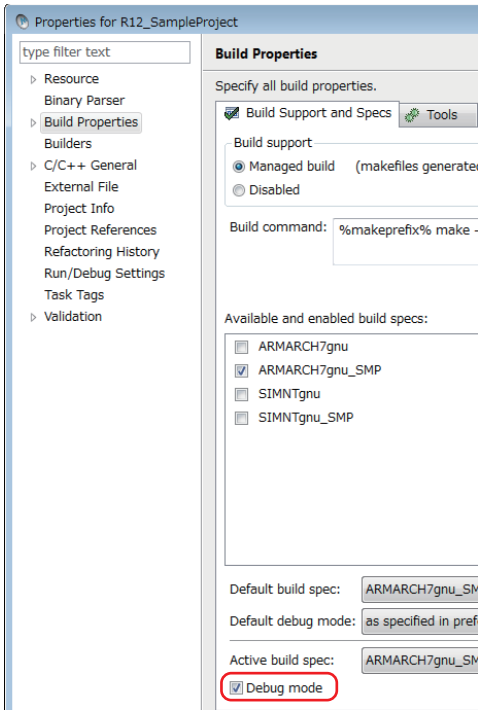
3. Select the [Build Support and Specs] tab.



4. Select only the checkbox of "ARMARCH7gnu_SMP" in "Available and enabled build specs".



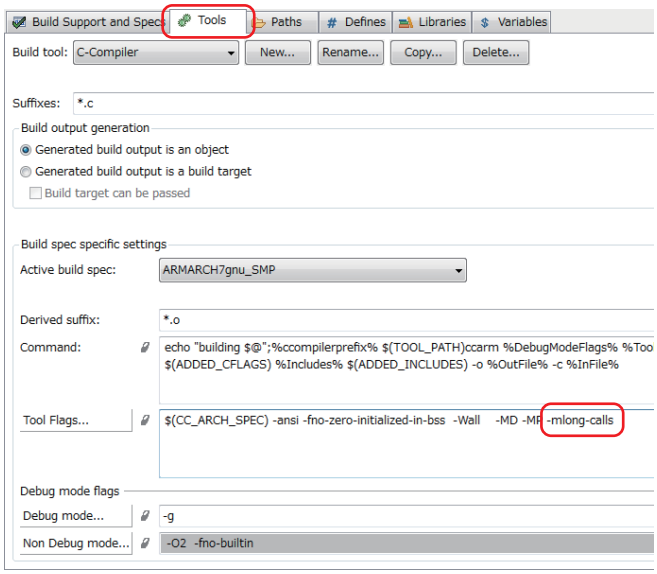
5. Select the checkbox of "Debug mode".



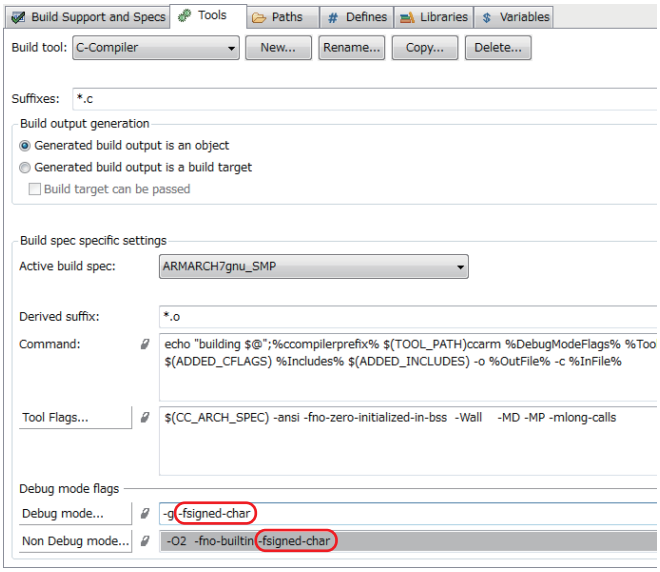
Point

For the actual system operation, unselect the checkbox of "Debug mode".

6. Select the [Tools] tab, enter "-mlong-calls" in the field next to the [Tool Flags] button.



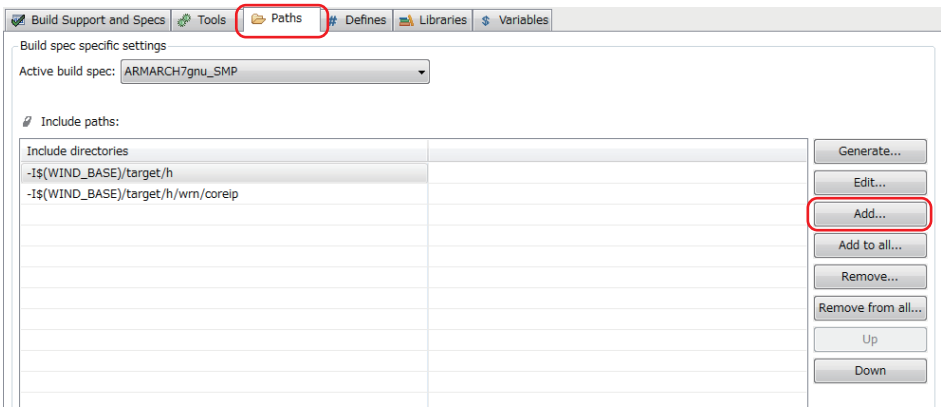
7. Enter "-fsigned-char" in the fields next to the [Debug mode] button and [Non Debug mode] button under "Debug mode flags".



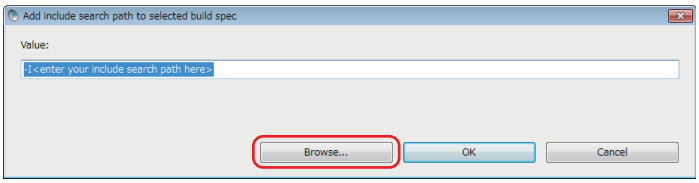
■Setting an include file

Operating procedure

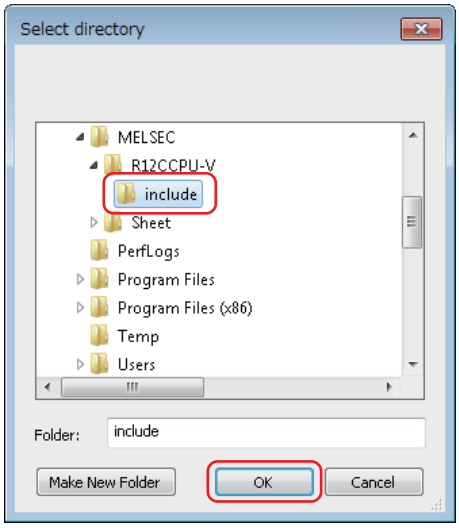
1. Select the [Paths] tab, and click the [Add] button.



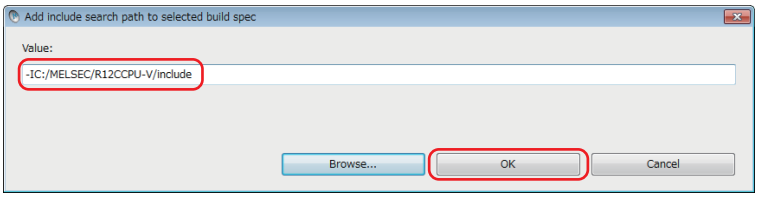
2. Click the [Browse] button.



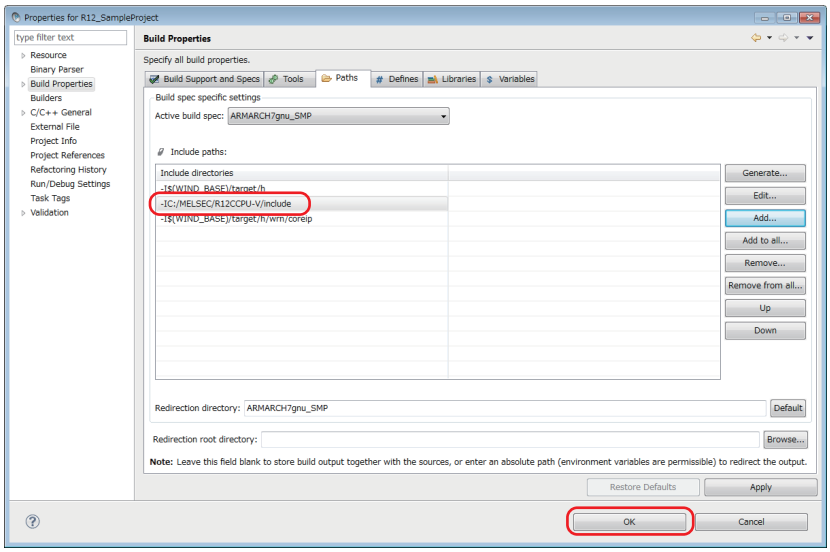
3. Select the include folder for the C Controller module in the "Select directory" screen, and click the [OK] button.
In this procedure, the include file is stored in "C:\MELSEC\R12CCPU-V".



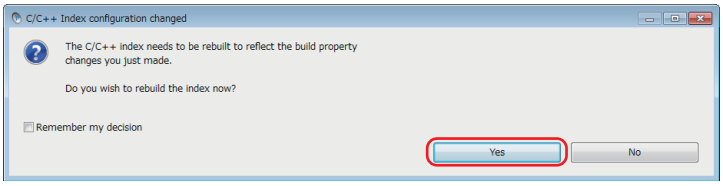
4. Check that the selected folder is specified in the "Add include search path to selected build spec" screen, and click the [OK] button.



5. Check that the added include path is displayed in "Include paths", and click the [OK] button.



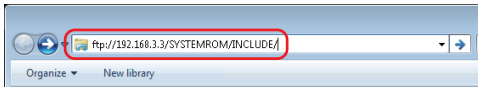
6. If the following message appears after clicking the [OK] button, click the [Yes] button.



7. Add the include file to the include folder added in step 1 to 5.

To acquire an include file stored in the C Controller module, start Explorer and enter the following address in the address bar.

ftp://192.168.3.3/SYSTEMROM/INCLUDE/

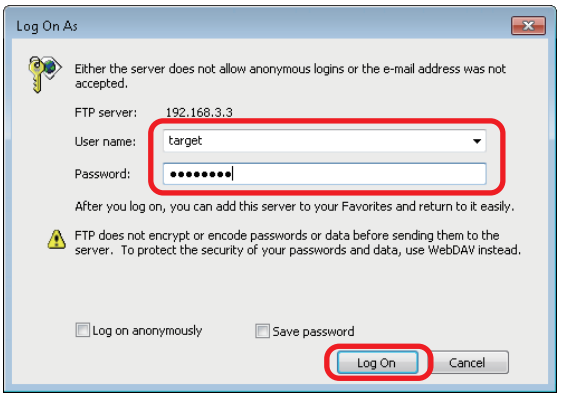


The "Log On As" screen appears.

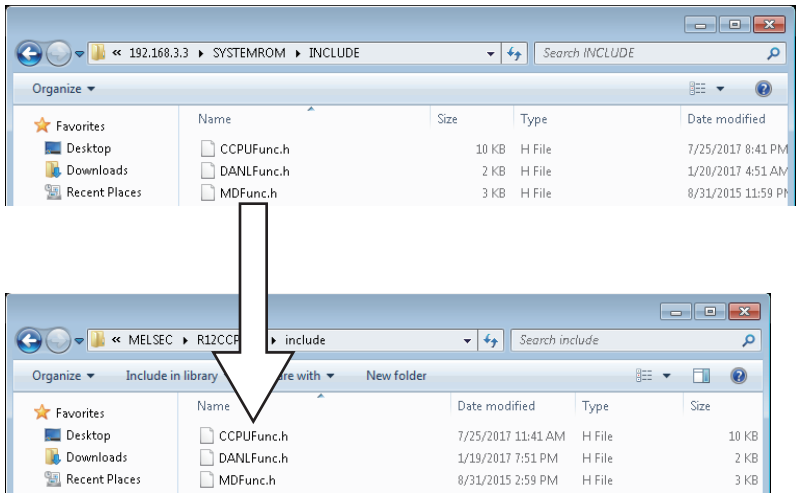
8. Enter the following user name and password in the "Log On As" screen.

- User name: target
- Password: password

9. Click the [Log On] button.



10. Copy the header file to the include folder added in step 1 to 5.



Preparing a user program

Prepare a user program that controls the C Controller system.
A sample program for C Controller modules is used in this guide.

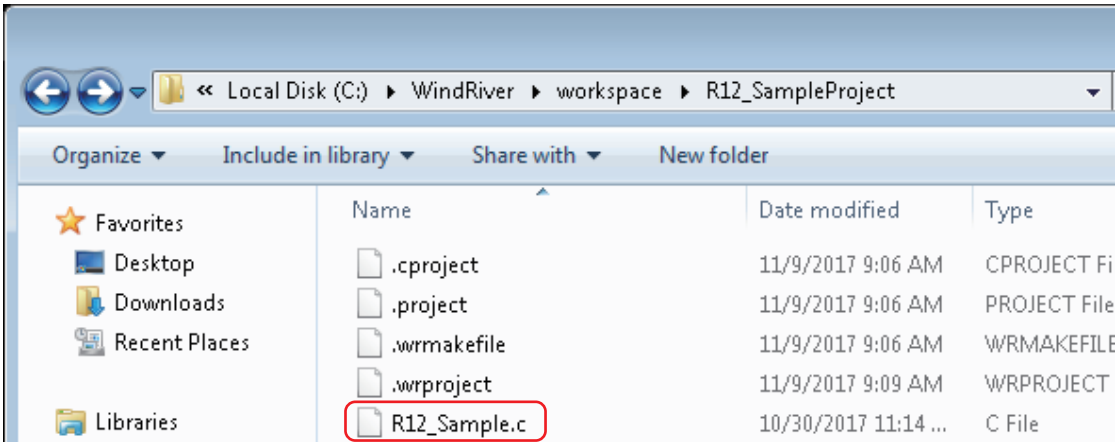
Point

For sample programs for this guide, please contact your local Mitsubishi Electric sales office or representative.

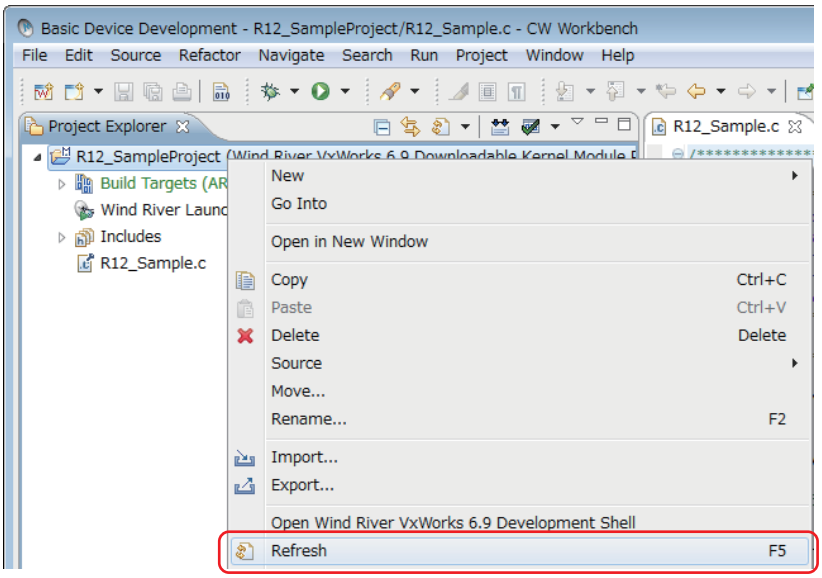
Adding a sample program

Operating procedure

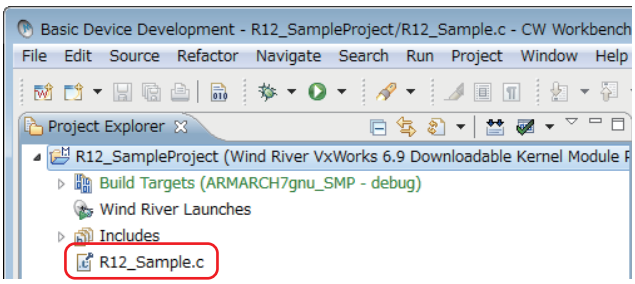
1. Store a sample program for this guide immediately under the created project folder.
"C:\WindRiver\workspace\R12_SampleProject"



2. Select and right-click the created project in the "Project Explorer" window, and click [Refresh].



3. The sample program stored in step 1 is added to the project.

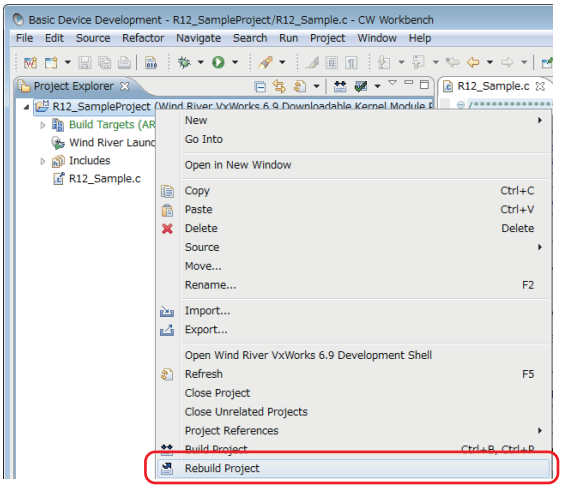


Generating an execution module from the user program

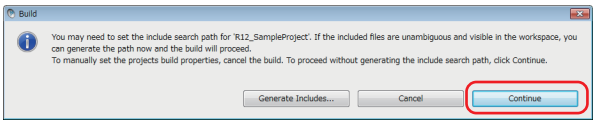
Convert (build) the created program into a module that can be executed on the C Controller module.

Operating procedure

1. Select and right-click the created project in the "Project Explorer" window, and click [Rebuild Project].

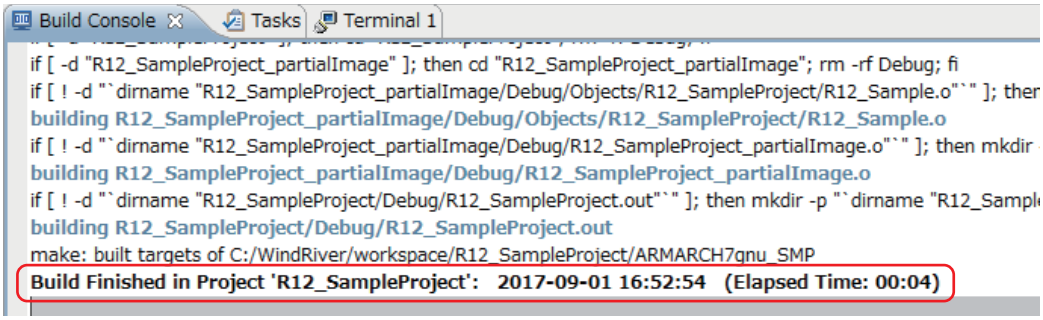


2. If the following message appears after selecting [Rebuild Project], click the [Continue] button.



The project starts to be built. The progress is displayed in the "Build Console" window.

3. Check that "Build Finished..." is displayed in the "Build Console" window.



Point

If "Build Finished..." is not displayed and an error occurs, check the error and modify the program.
After modifying the program, perform the operation again from 'Page 36 Generating an execution module from the user program'.

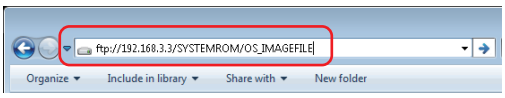
Connecting the C Controller module with CW Workbench

Connect the user Ethernet port CH1 of the C Controller module with CW Workbench to perform debugging in CW Workbench.

Operating procedure

1. To acquire a VxWorks image file from the C Controller module, start Explorer and enter the following address in the address bar.

ftp://192.168.3.3/SYSTEMROM/OS_IMAGEFILE/



The "Log On As" screen appears.

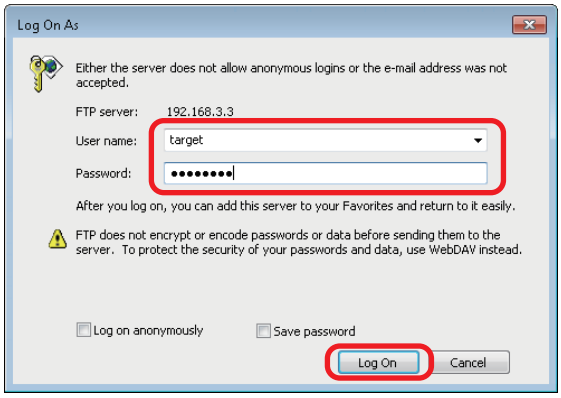
Point

To communicate between the C Controller module and the personal computer, specify the same VxWorks image file for both.

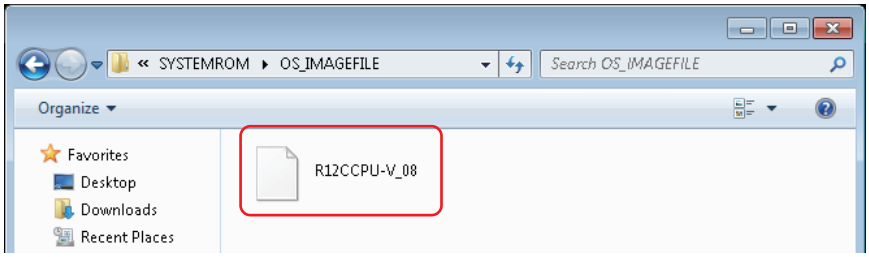
2. Enter the following user name and password in the "Log On As" screen.

- User name: target
- Password: password

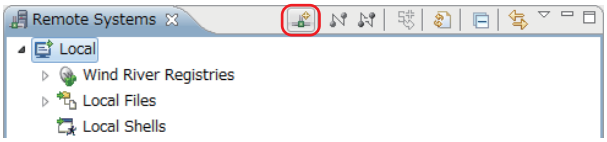
3. Click the [Log On] button.



4. Create "C:\MELSEC\R12CCPU-V\CCPUTool" folder, and copy the VxWorks image file stored in the C Controller module to the folder.

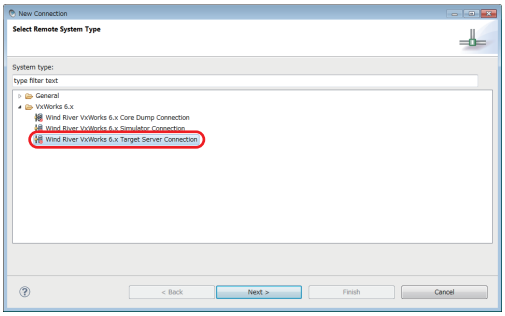


5. Click in the "Remote Systems" window.

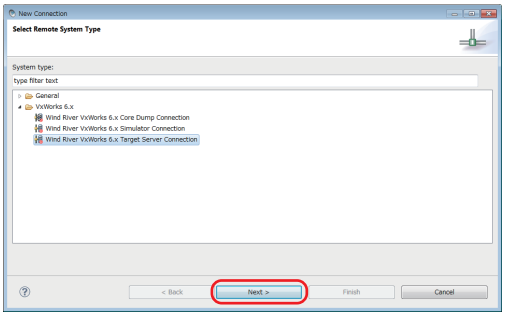


The "New Connection" screen appears.

6. Select "Wind River VxWorks 6.x Target Server Connection" in the "New Connection" screen.

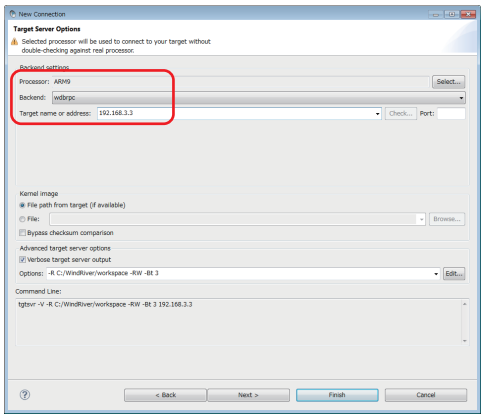


7. Click the [Next] button.

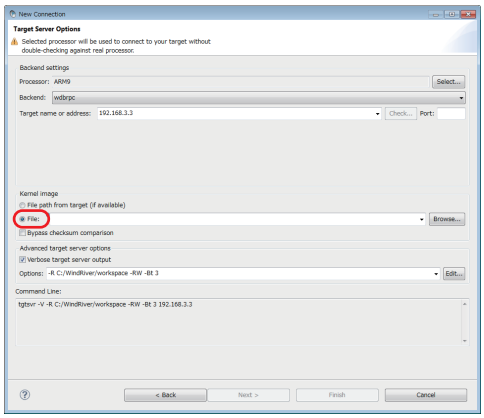


8. Set the following items in "Backend settings".

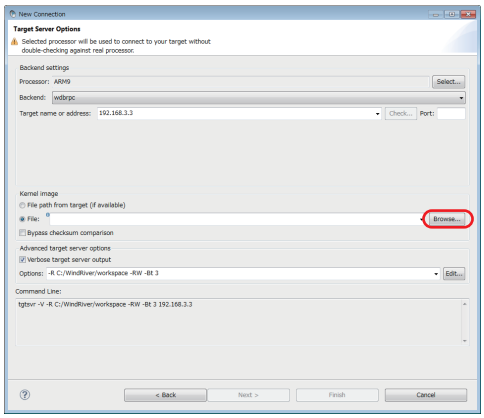
- Processor : ARM9 (Click the [Select] button and select the processor.)
- Backend: wdbrpc
- IP Address: 192.168.3.3 (default)
- Port: blank



9. Select "File" in "Kernel image".

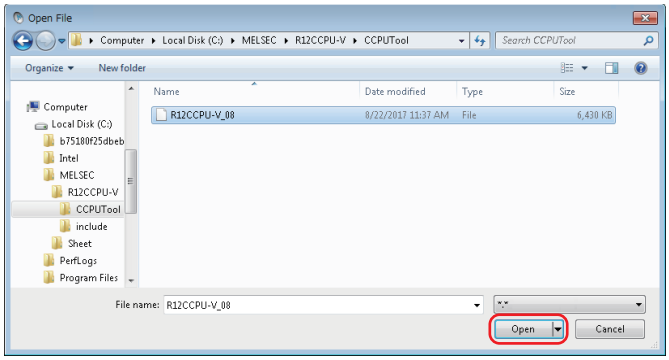


10. Click the [Browse] button.

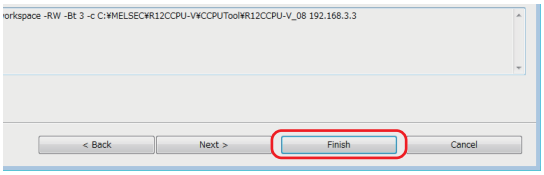


The "Open File" screen appears.

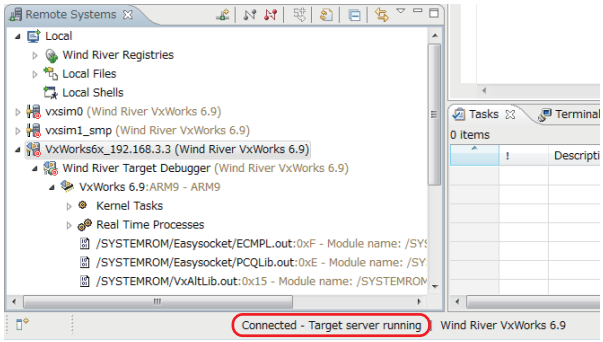
11. Select the VxWorks image file copied to the "C:\MELSEC\R12CCPU-V\CCPUTool" folder in step 4, and click the [Open] button.



12. Click the [Finish] button.



13. The connection is completed when "Connected - Target server running" is displayed at the bottom of the "Remote Systems" window.



Point

If "Connected - Target server running" is not displayed, check that the C Controller module is normally powered ON, and perform the operation again from 'Page 37 Connecting the C Controller module with CW Workbench'.

Debugging the user program

Check if the created program runs properly.

■Downloading the user program on the C Controller module

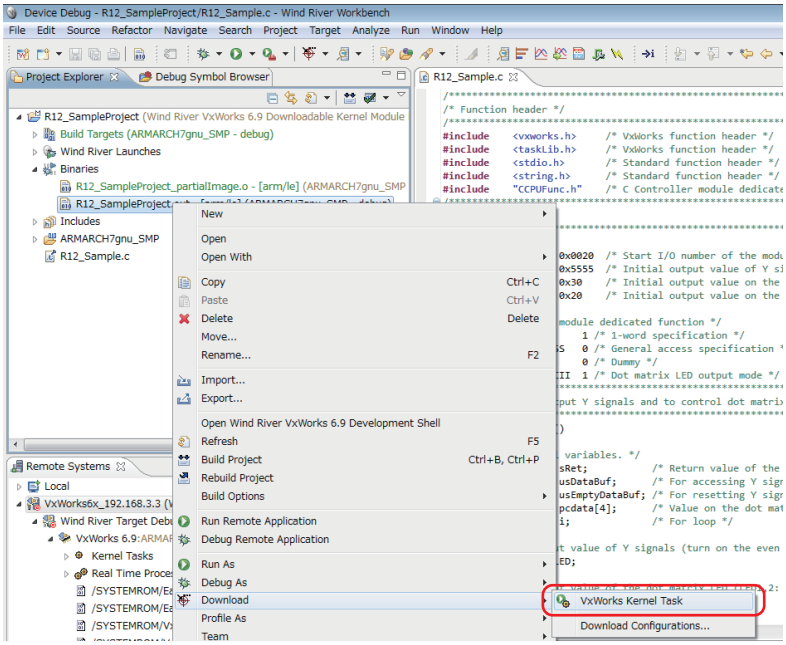
To debug the user program, download the execution module on the memory in the C Controller module.
By downloading a user program, the program can be executed with no script file.

Point

Script file: A file to describe a loading location of a user program that starts at the start of a C Controller module and a startup order of the user program.

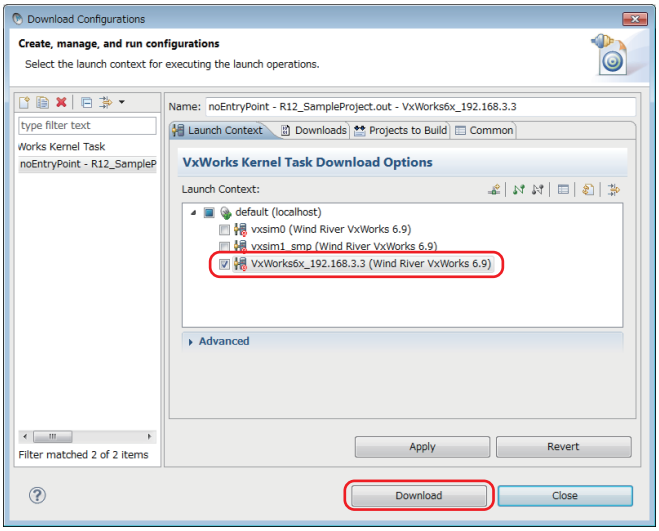
Operating procedure

1. Select and right-click the created module file "R12_SampleProject.out" in the "Project Explorer" window, and select [Download] ⇨ [VxWorks Kernel Task].



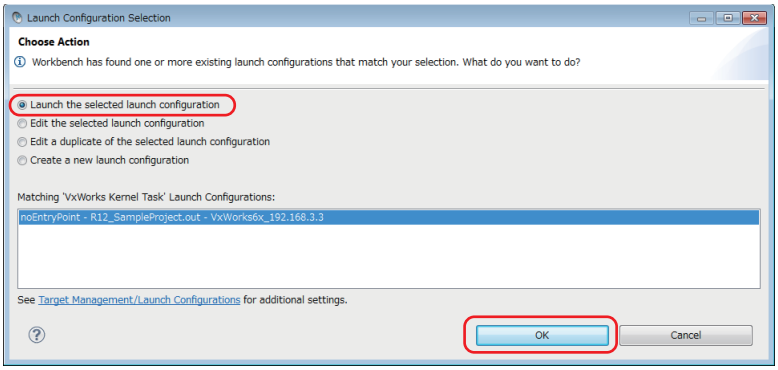
The "Download Configurations" screen appears.

2. Select only the checkbox of "VxWorks6x_192.168.3.3 (Wind River VxWorks 6.9)" in the [Launch Context] tab, and click the [Download] button.



Point

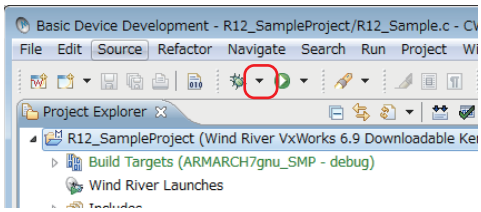
The "Launch Configuration Selection" screen appears on and after the second operation of the step 2. Select "Launch the selected launch configuration", and click the [OK] button.



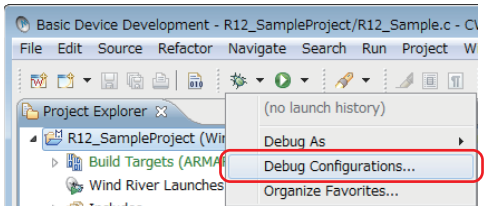
■Debugging the user program

Operating procedure

1. Select the created project in the "Project Explorer" window, and click the [▼] button on the right side of on the toolbar.

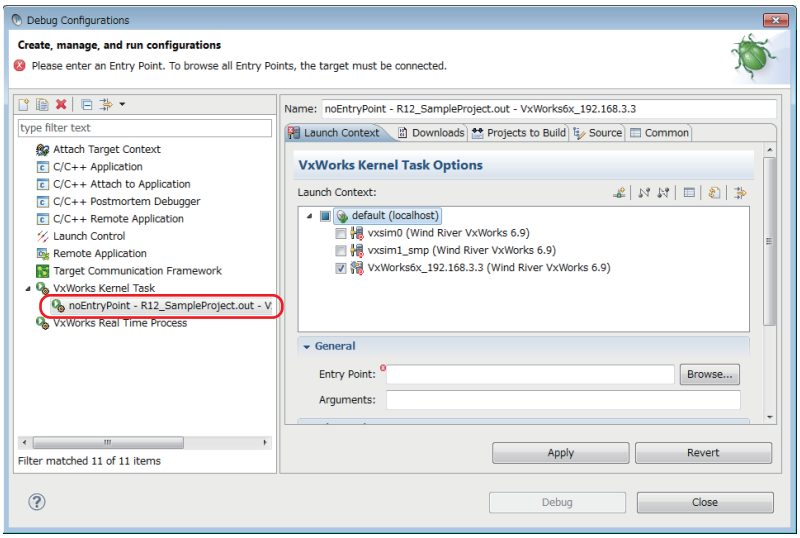


2. Select [Debug Configurations].

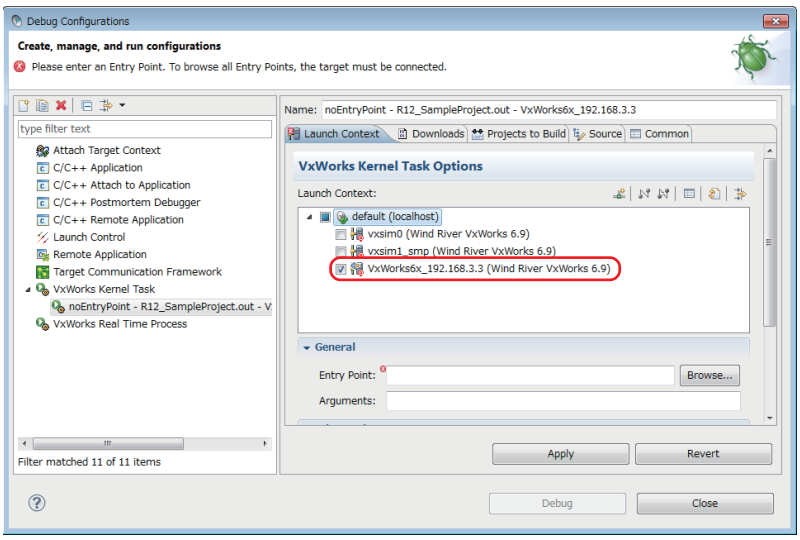


The "Debug Configurations" screen appears.

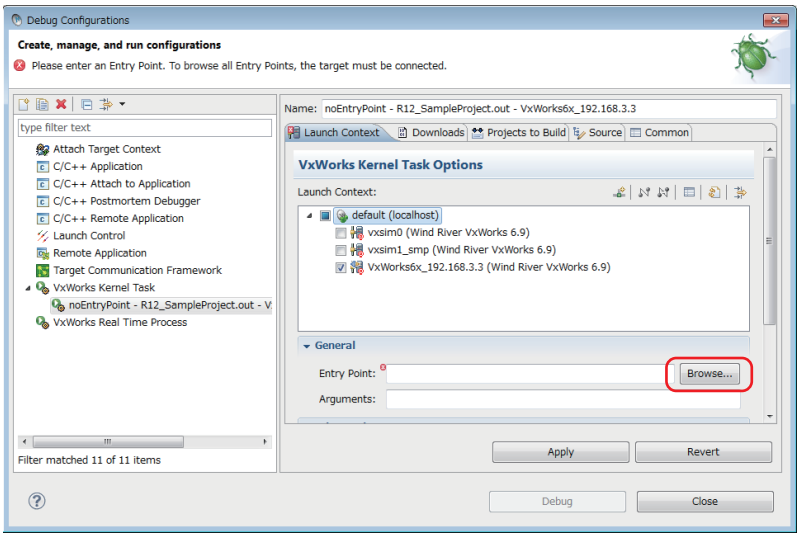
3. Click the downloaded module "R12_SampleProject.out" under "VxWorks Kernel Task".



4. Select a target server indicating connection to the C Controller module.

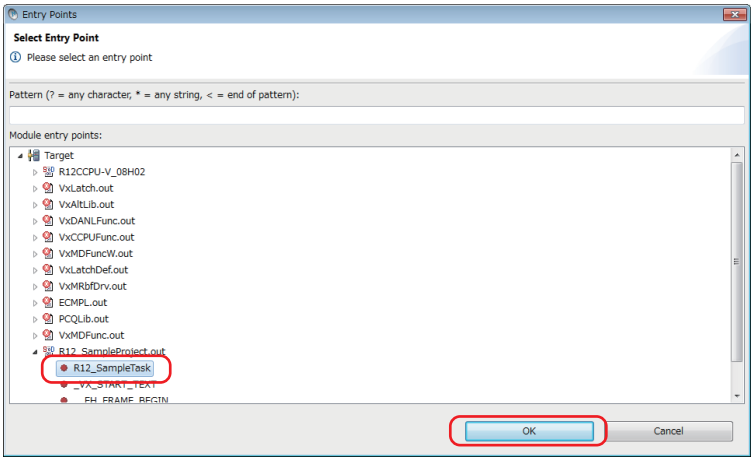


5. Click the [Browse] button.

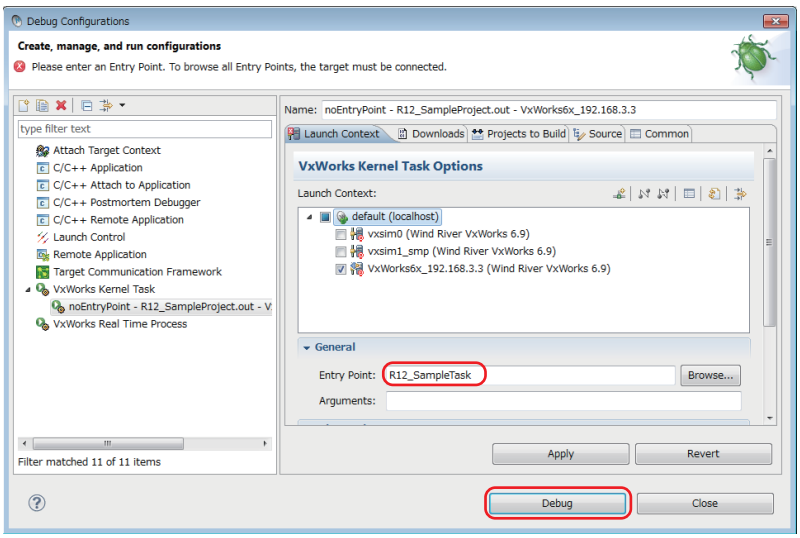


The "Entry Points" screen appears.

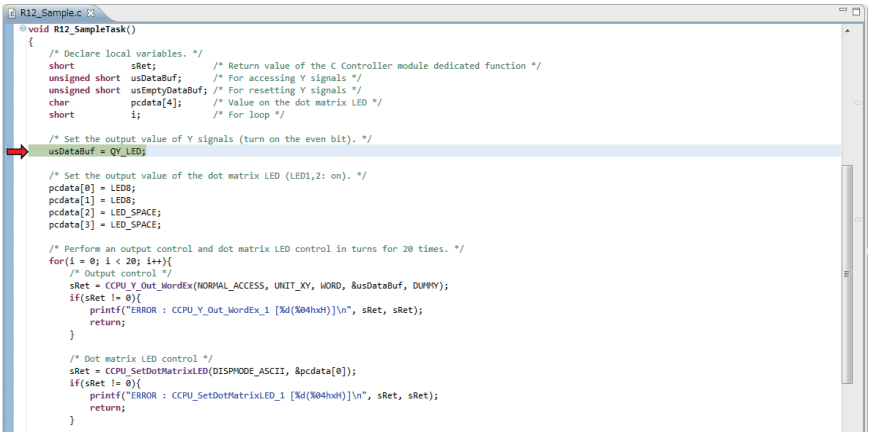
6. Select the function "R12_SampleTask" that starts debugging, and click the [OK] button.



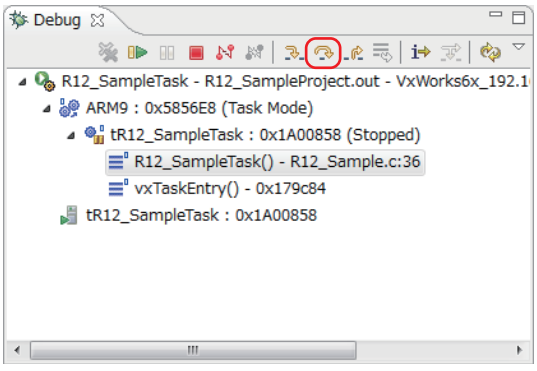
7. Check that the function name selected in step 6 is set for "Entry Point", and click the [Debug] button.



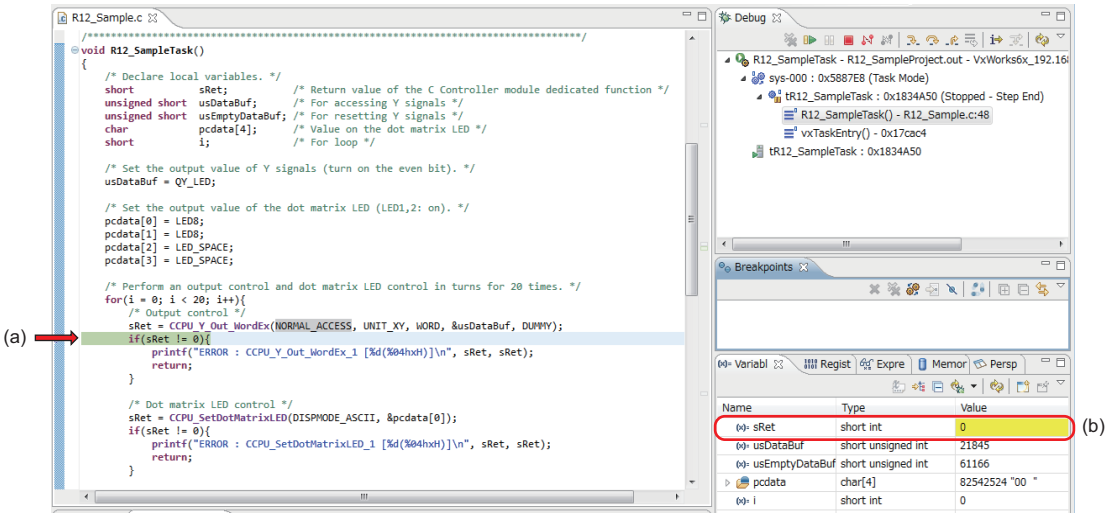
8. Debugging starts. Program execution stops at the start of the function specified for "Entry Point".



9. Click in the "Debug" window to debug a program by one step.



10. Variable values can be checked and changed in the "Variables" window on the bottom right of the screen.
Check that 'sRet', return value of the CCPU function, is '0' (normal value) in this step.
(a) Run the programs to the line indicated with the red arrow '→' by the step execution in step 9.
(b) Check that the value of sRet is '0' (normal value) in the [Variables] tab.



Repeat steps 9 and 10 to debug the whole created program.

Point

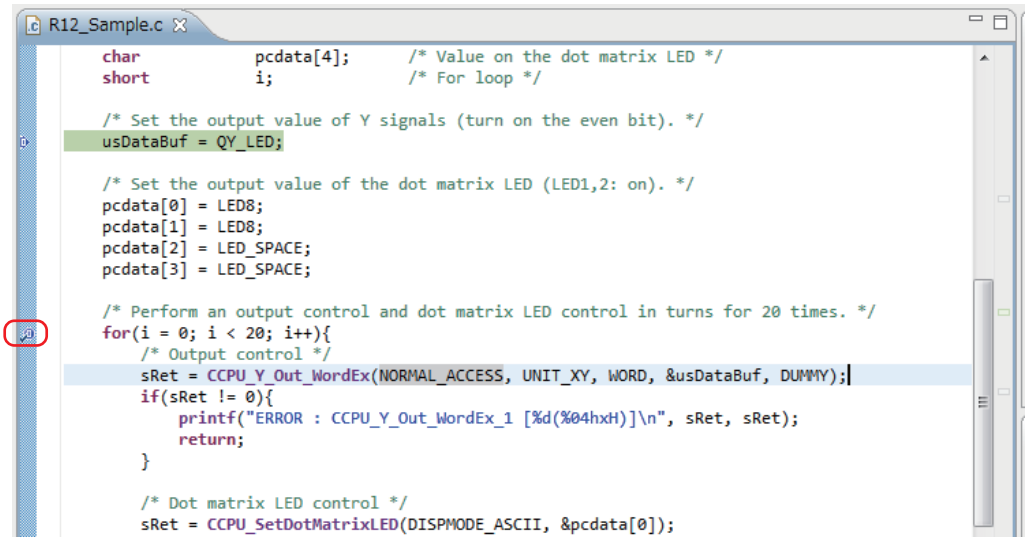
When the return value of the C Controller module dedicated function is other than '0', refer to the following manual and troubleshoot the symptom.
MELSEC iQ-R C Controller Module Programming Manual

■Debugging using breakpoint

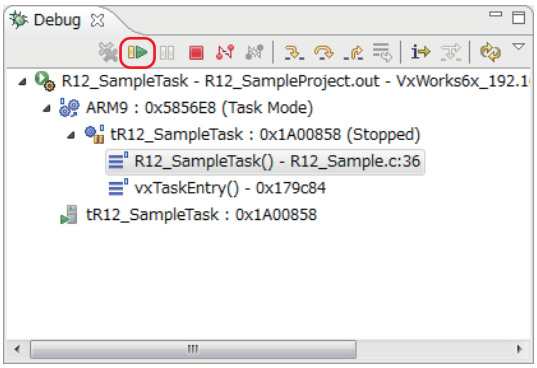
As well as debugging in units of one step described in step 9 , debugging using a breakpoint is available.

Operating procedure

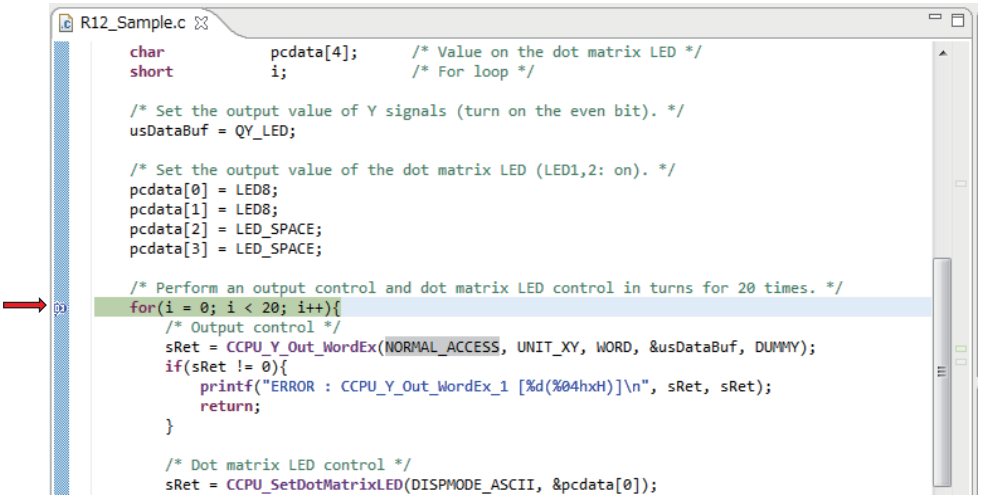
1. Double-click the left edge of the source file window and insert a breakpoint.



2. Click .



The program runs to the position where the breakpoint is specified.



Point

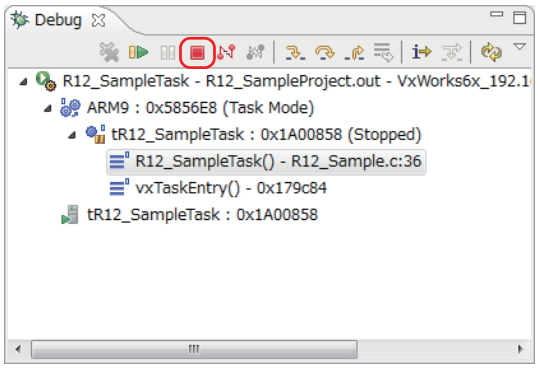
The descriptions of icons are as follows:

- : Step Into

  Steps into the called function and stops at the first line of the function.
- : Step Over

  Executes the current line of the function and then stops at the next line of the function.
- : Continues execution of the current function until it returns to its caller.
- : Executes a program.
- : Stops program execution.
- : Ends debugging.

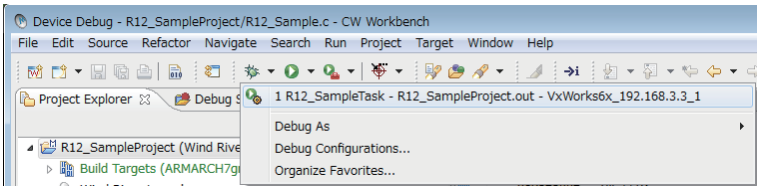
3. Click in the "Debug" window to end the debugging.



Point

To start debugging again, click the [▼] button on the right side of on the toolbar, and select the created debug configuration at the top of the pop-up menu.

By the operations above, the steps 1 to 8 in 'Page 42 Debugging the user program' can be skipped.



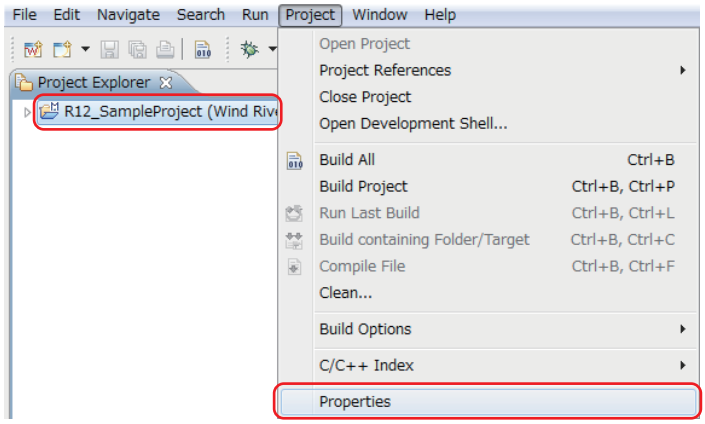
Registering the execution module

Build the created program for operation, and store the program on the C Controller module.

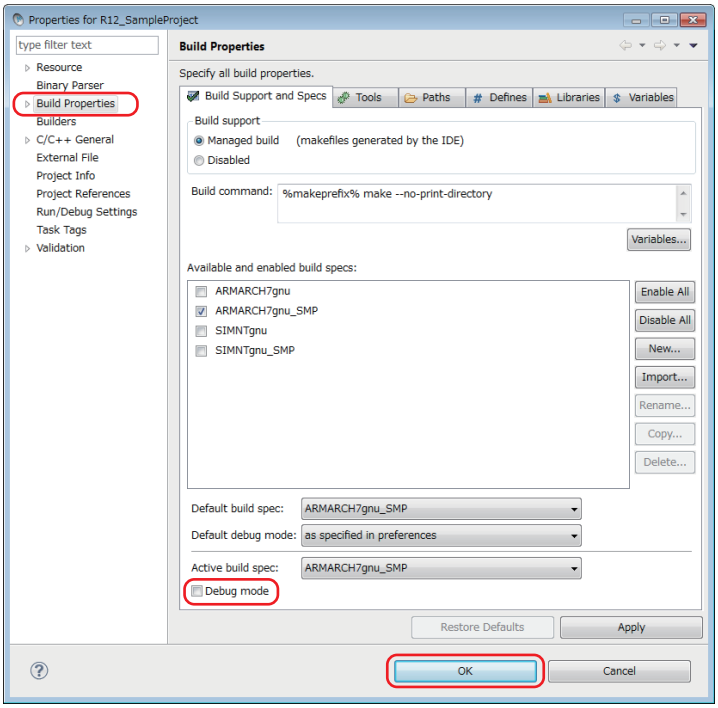
■Building the user program

Operating procedure

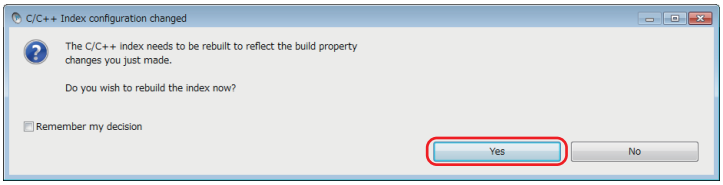
1. Select the created project in the "Project Explorer" window, and click [Project] ⇨ [Properties].



2. Select "Build Properties" from the tree to the left in the screen, unselect the "Debug mode" checkbox, and click the [OK] button.



3. If the following message appears after clicking the [OK] button, click the [Yes] button.



4. Build the program by following the procedure in 'Page 36 Generating an execution module from the user program'.

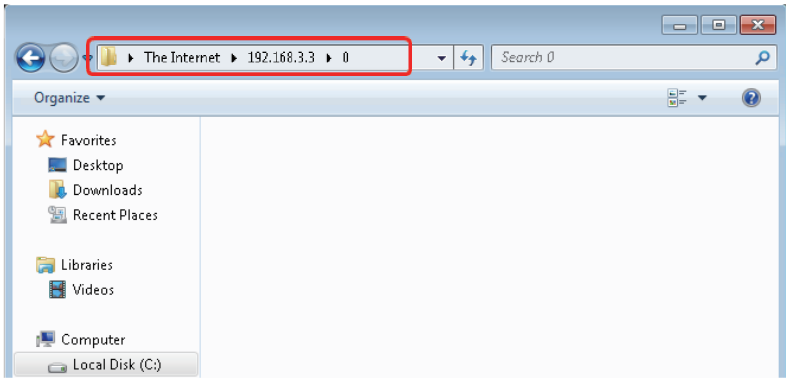
■Storing the user program

Operating procedure

1. Start Explorer, and enter the following address in the address bar.

ftp://192.168.3.3/0

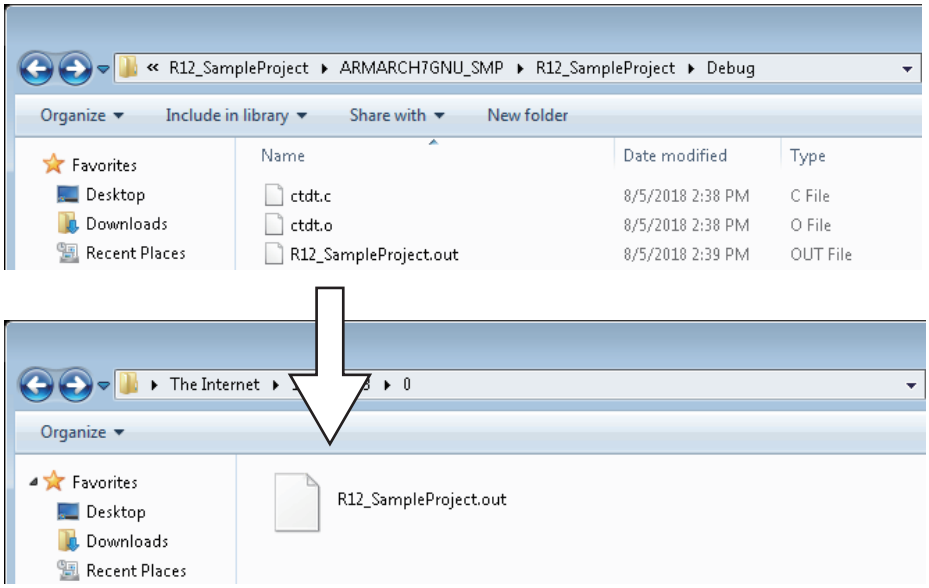
After login to the C Controller module, the address is displayed as follows.



2. Copy the created user program "R12_SampleProject.out" to the program memory '0' in the C Controller module by a drag and drop operation.

The user program created in this guide is stored in the following directory:

C:\WindRiver\workspace\R12_SampleProject\ARMARCH7gnu_SMP\R12_SampleProject\Debug

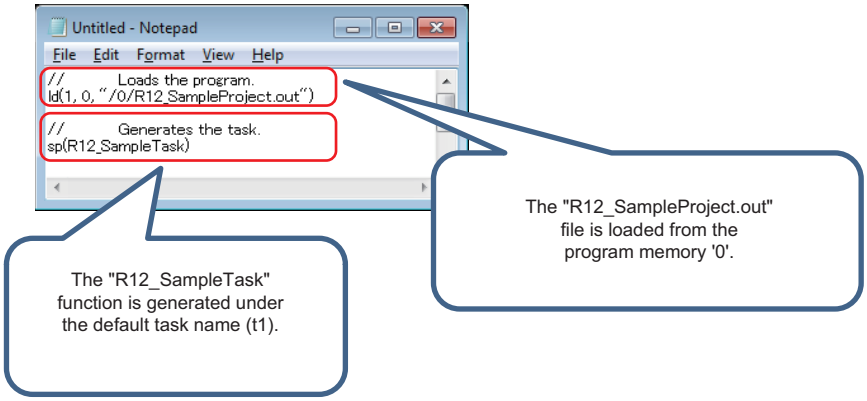


■Creating and storing a script file

Create a script file that automatically downloads the execution module at the start of the C Controller module.

Operating procedure

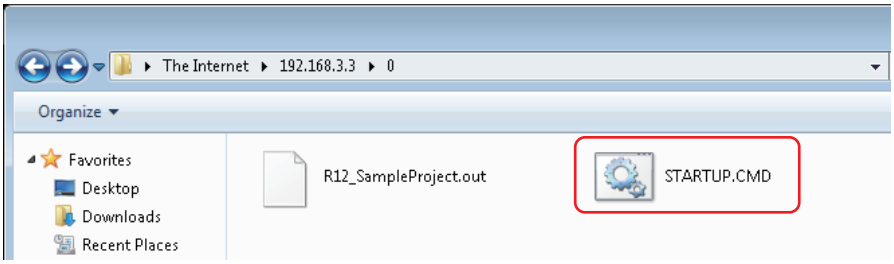
1. Open a text file and describe a script file that downloads a user program and generates a task, as shown below.



2. Name the file as "STARTUP.CMD" and save the file.

3. Copy the created script file to the program memory in the C Controller module.

ftp://192.168.3.3/0



The script file has been created and stored.

Point

A user program and a script file can be stored on the SD memory card as well.
When a script file is stored in both the program memory and the SD memory card, one in the SD memory card is started by priority.

2.6 Checking Operations

Run the program registered in the C Controller module to check its operations.
The status of the C Controller module can be changed by the RESET/STOP/RUN switch on the front of the module.
Change the status of the RESET/STOP/RUN switch as follows, depending on the purpose.

- RUN: State where output (Y) from a user program and writing to the buffer memory are permitted
- STOP: State where output (Y) from a user program and writing to the buffer memory are prohibited
- RESET: To reset the module.

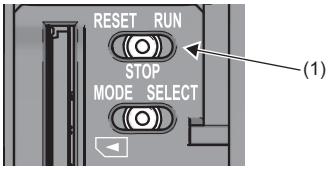
Point

- The C Controller module executes program operation regardless of the switch status (RUN/STOP).
- For details on the RUN/STOP/RESET switch, refer to the following manual.
  (MELSEC iQ-R C Controller Module User's Manual (Startup))

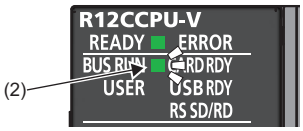
Enabling outputs (Y) from the user program

Operating procedure

1. Set the RESET/STOP/RUN switch (1) on the front of the C Controller module to the RUN position.



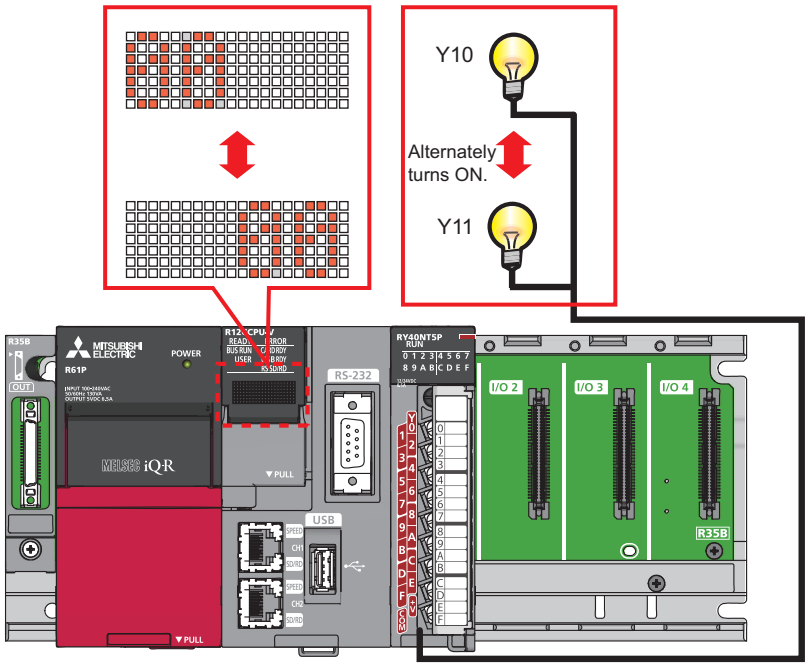
2. When the BUS RUN LED (2) lights in green, the program is running normally.



Checking operations with the dot matrix LED and lamps

The dot matrix LED on the front of the C Controller module and output lamps operate as follows:

1. The display of the dot matrix LED on the front of the C Controller module switches alternately ten times.
2. Synchronizing with the dot matrix LED, the output lamps Y10 and Y11 alternately turn ON.



3. To check the operations again, reset the C Controller module.

3 FREQUENTLY-USED FUNCTIONS

This chapter describes functions frequently used for the start-up and the maintenance after operation of a C Controller system.

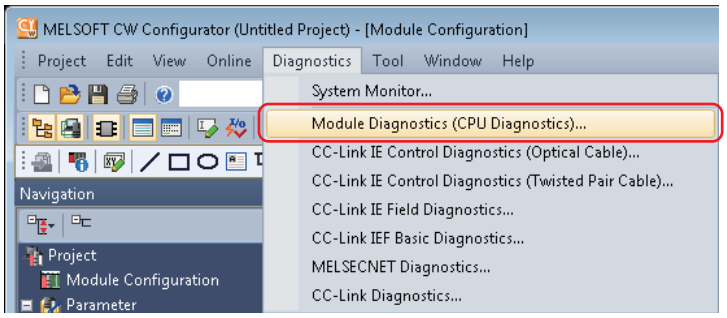
3.1 Checking Errors

An error can be checked and the corrective action can be taken using CW Configurator.

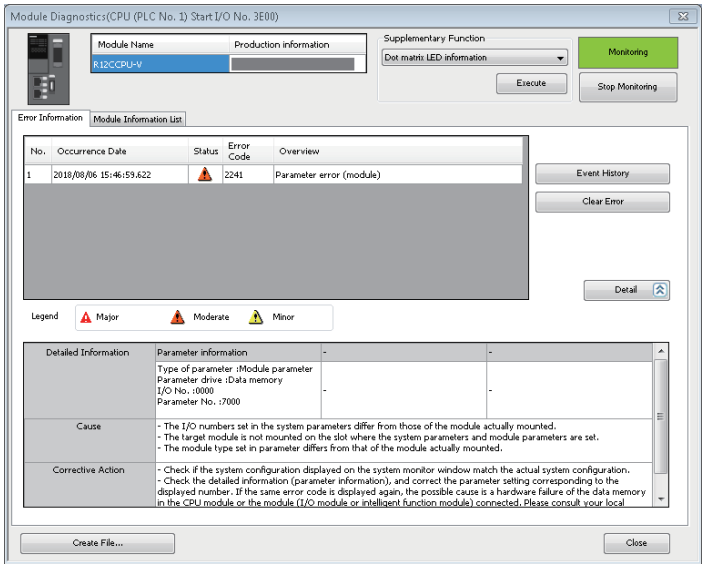
Checking an error

Operating procedure

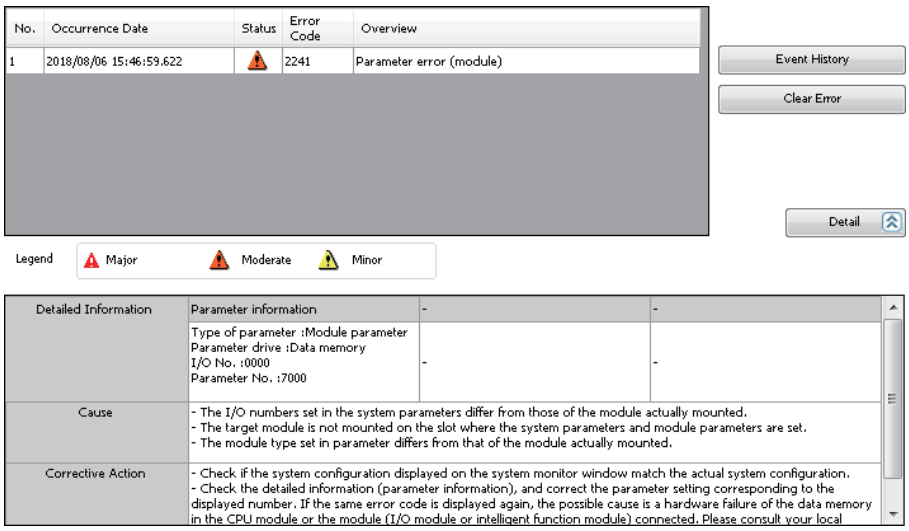
1. Start CW Configurator.
2. Select [Diagnostics] ⇨ [Module Diagnostics (CPU Diagnostics)].



3. The "Module Diagnostics" screen appears.



4. An error code is displayed in the screen. The detailed information such as cause and corrective action can be checked.

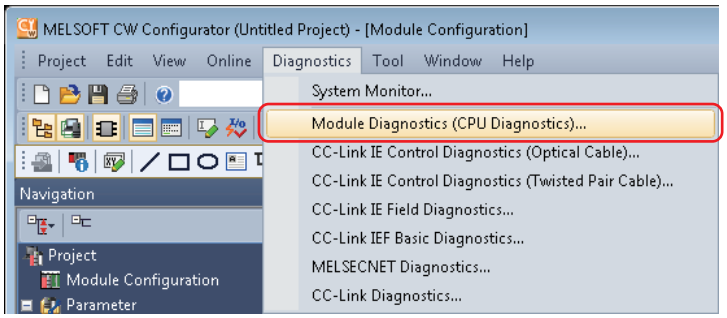


Checking error history

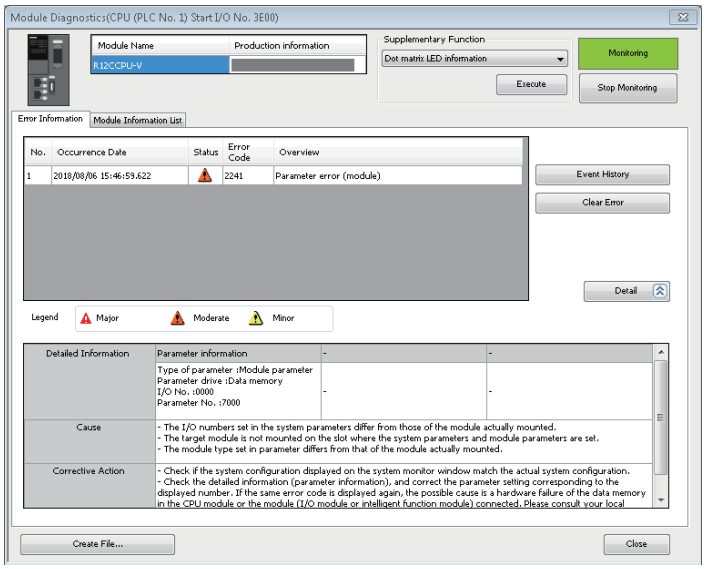
Errors occurred up to the present and the error details can be checked.
When and what kind of error occurs can be checked, useful in error analysis.

Operating procedure

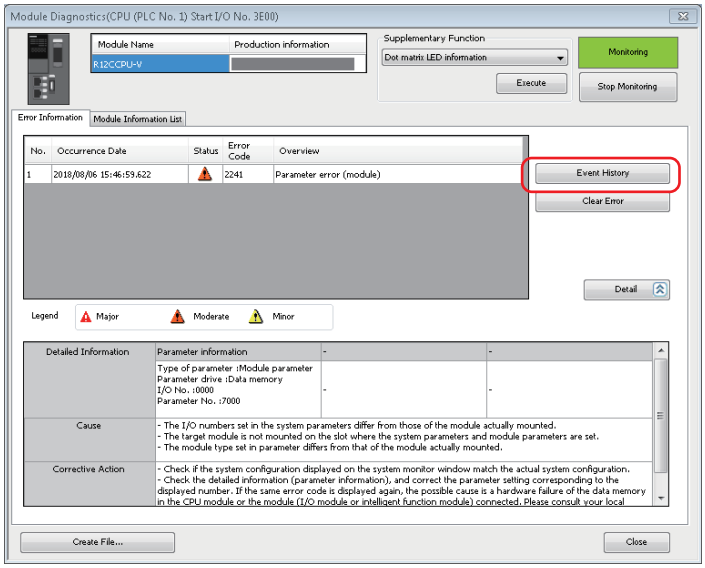
1. Start CW Configurator.
2. Select [Diagnostics] ⇨ [Module Diagnostics (CPU Diagnostics)].



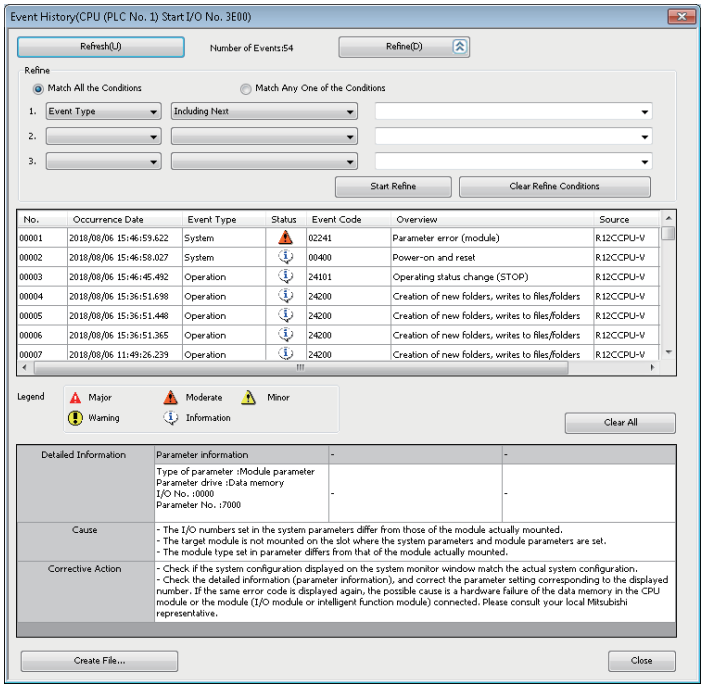
3. The "Module Diagnostics" screen appears.



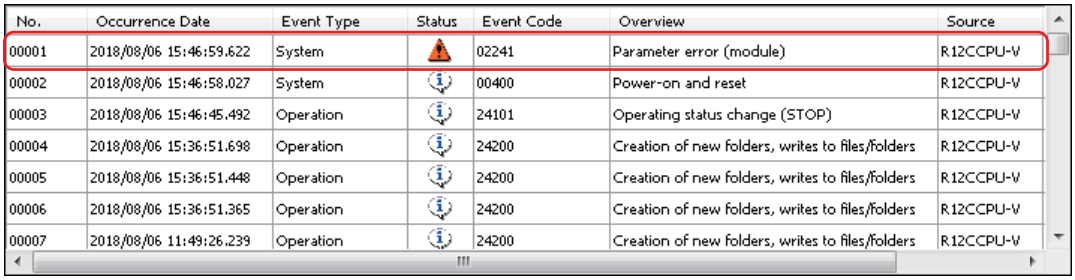
4. Click the [Event History] button.



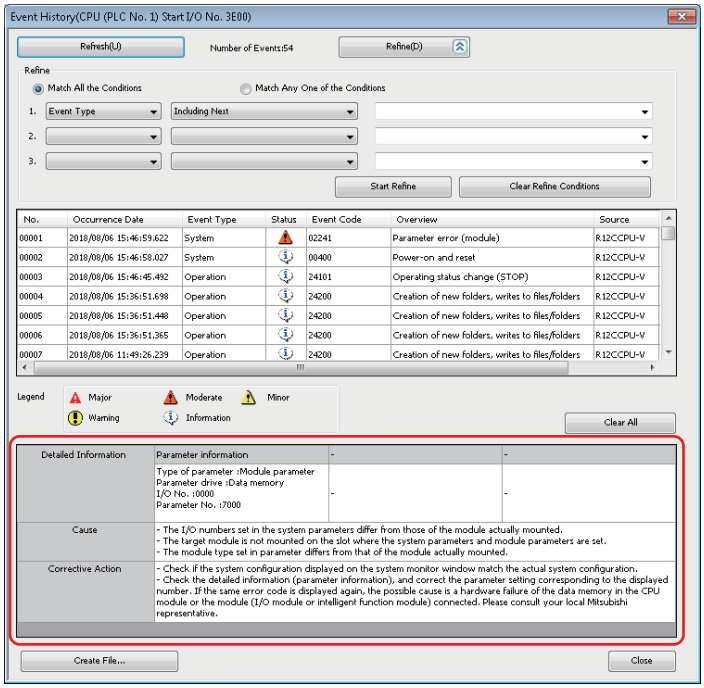
5. Error history and the error details are displayed.



6. To see more details of an error, click the error.



7. The detailed information is displayed.



3.2 Monitoring Module Status and Testing Operations

Module I/O status and buffer memory status can be checked through CW Configurator.
In addition, I/O status can be checked and operations can be tested at start-up and maintenance.

Monitoring module status

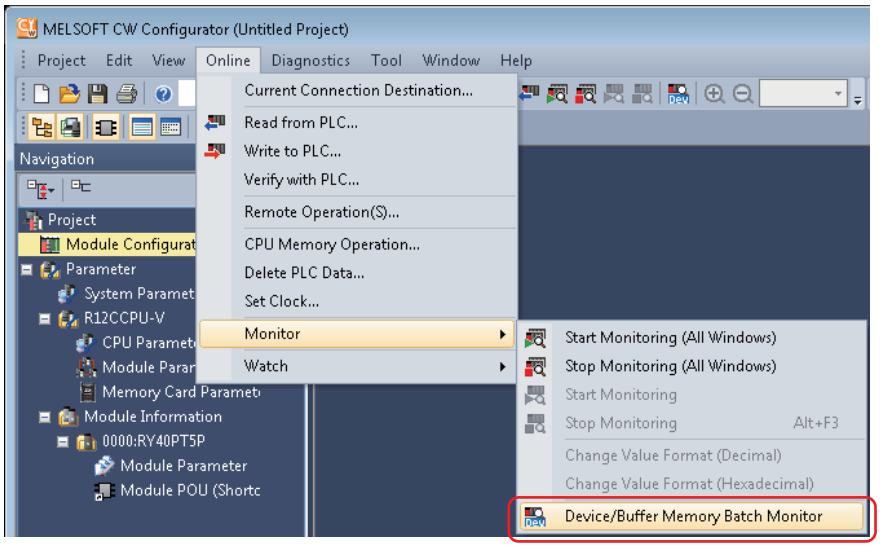
The status of input (X), output (Y), and buffer memory in the module can be monitored.

Point

Buffer memory: The internal memory of an intelligent function module (such as A/D conversion module and D/A conversion module having a function other than input and output) used to send and receive data (such as setting values and monitored values) for communication with a C Controller module

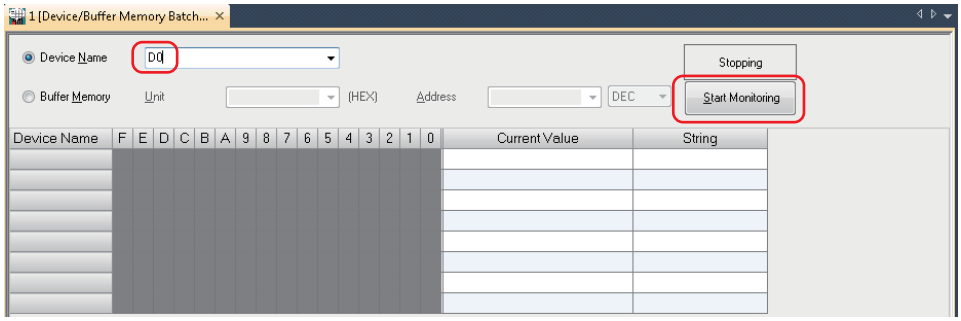
Operating procedure

1. Start CW Configurator.
2. Select [Online] ⇨ [Monitor] ⇨ [Device/Buffer Memory Batch Monitor].

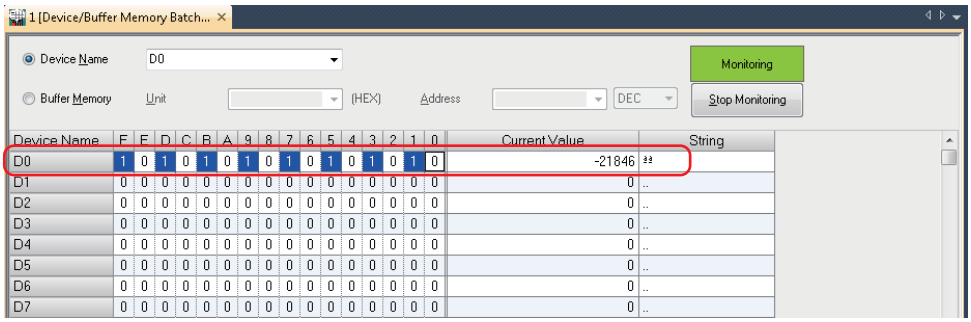


3. The "Device/Buffer Memory Batch Monitor" screen appears. Enter a target device (start) for "Device Name", and click the [Start Monitoring] button.

The following shows an example for "D0".



The status of "D0" can be checked.

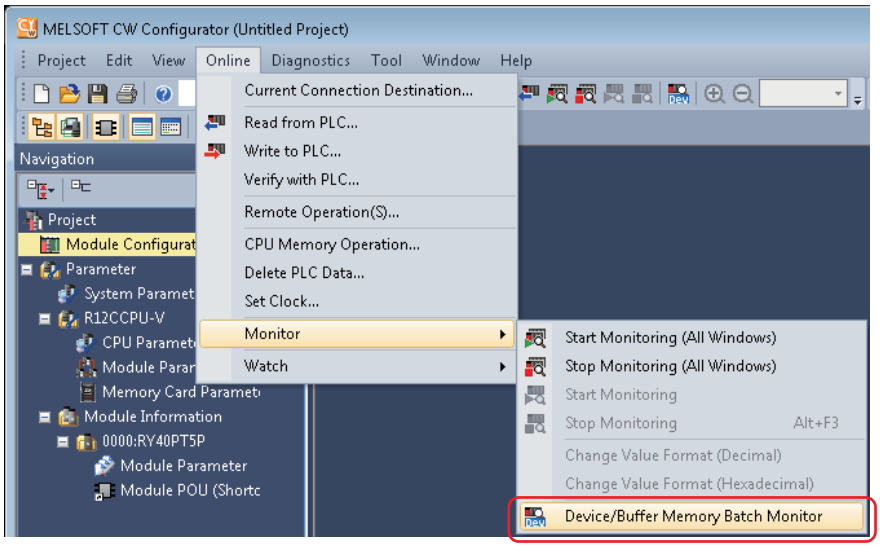


Testing operations by forced output

Module operations can be tested by forced output from an output (Y).
The following shows the procedure for forced output (Y).

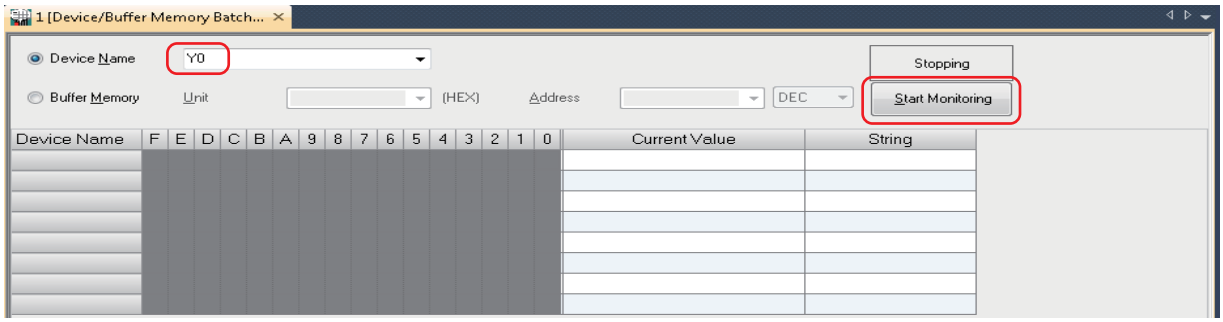
Operating procedure

1. Start CW Configurator.
2. Select [Online] ⇨ [Monitor] ⇨ [Device/Buffer Memory Batch Monitor].



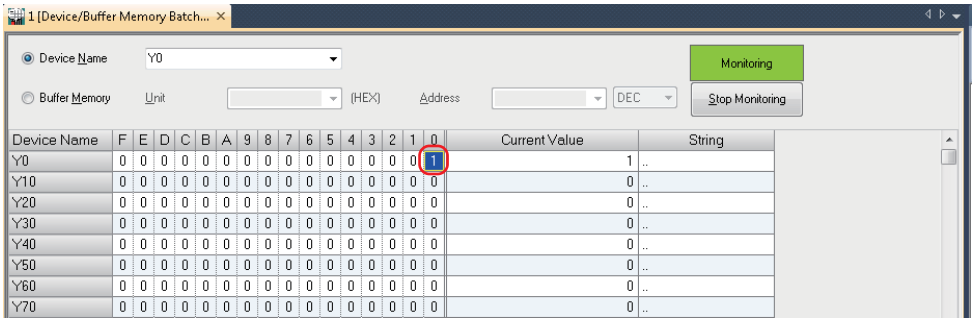
3. The "Device/Buffer Memory Batch Monitor" screen appears. Enter a target device (start) for "Device Name", and click the [Start Monitoring] button.

The following shows an example for "Y0".



4. Double-click the 0th bit of Y0.

The 0th bit of Y0 is changed from '0' to '1', and the output (Y) is output forcibly.



Point

An operation test by forced write to a buffer memory can be executed in the same manner.

TRADEMARKS

VxWorks and Wind River are either registered trademarks or trademarks of Wind River Systems, Inc.
Windows is either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
In general, the names of companies, systems, products, etc. that appear in this manual are either registered trademarks or trademarks of their respective companies.
In some cases, trademark symbols such as '™' or '®' are not specified in this manual.

Precautions before use

This publication explains the typical features and functions of the products herein and does not provide restrictions or other information related to usage and module combinations. Before using the products, always read the product user manuals. Mitsubishi Electric will not be held liable for damage caused by factors found not to be the cause of Mitsubishi Electric; opportunity loss or lost profits caused by faults in Mitsubishi Electric products; damage, secondary damage, or accident compensation, whether foreseeable or not, caused by special factors; damage to products other than Mitsubishi Electric products; or any other duties.

⚠For safe use

- To use the products given in this publication properly, always read the relevant manuals before use.
- The products have been manufactured as general-purpose parts for general industries, and have not been designed or manufactured to be incorprated in a device or system used in purposes related to human life.
- Before using the products for special purposes such as nuclear power, electric power, aerospace, medicine or passenger movement vehicles, consult with Mitsubishi.
- The products have been manufactured under strict quality control. However, when installing the products where major accidents or losses could occur if the products fail, install appropriate backup or fail-safe functionss in the system.

Programmable Controller
C Controller Quick Start Guide

Country/Region	Sales office	Tel/Fax
USA	MITSUBISHI ELECTRIC AUTOMATION, INC. 500 Corporate Woods Parkway, Vernon Hills, IL 60061, U.S.A.	Tel : +1-847-478-2100 Fax : +1-847-478-2253
Mexico	MITSUBISHI ELECTRIC AUTOMATION, INC. Mexico Branch Mariano Escobedo #69, Col. Zona Industrial, Tlalnepantla Edo. Mexico, C.P.54030	Tel : +52-55-3067-7500
Brazil	MITSUBISHI ELECTRIC DO BRASIL COMÉRCIO E SERVIÇOS LTDA. Avenida Adelino Cardana, 293, 21 andar, Bethaville, Barueri SP, Brazil	Tel : +55-11-4689-3000 Fax : +55-11-4689-3016
Germany	MITSUBISHI ELECTRIC EUROPE B.V. German Branch Mitsubishi-Electric-Platz 1, 40882 Ratingen, Germany	Tel : +49-2102-486-0 Fax : +49-2102-486-1120
UK	MITSUBISHI ELECTRIC EUROPE B.V. UK Branch Travellers Lane, Hatfield, Hertfordshire, AL10 8XB, U.K.	Tel : +44-1707-28-8780 Fax : +44-1707-27-8695
Ireland	MITSUBISHI ELECTRIC EUROPE B.V. Irish Branch Westgate Business Park, Ballymount, Dublin 24, Ireland	Tel : +353-1-4198800 Fax : +353-1-4198890
Italy	MITSUBISHI ELECTRIC EUROPE B.V. Italian Branch Centro Direzionale Colleoni-Palazzo Sirio Viale Colleoni 7, 20864 Agrate Brianza(Milano) Italy	Tel : +39-039-60531 Fax : +39-039-6053-312
Spain	MITSUBISHI ELECTRIC EUROPE, B.V. Spanish Branch Carretera de Rubí, 76-80-Apdo. 420, 08190 Sant Cugat del Vallés (Barcelona), Spain	Tel : +34-935-65-3131 Fax : +34-935-89-1579
France	MITSUBISHI ELECTRIC EUROPE B.V. French Branch 25, Boulevard des Bouvets, 92741 Nanterre Cedex, France	Tel : +33-1-55-68-55-68 Fax : +33-1-55-68-57-57
Czech Republic	MITSUBISHI ELECTRIC EUROPE B.V. Czech Branch Avenir Business Park, Radlicka 751/113e, 158 00 Praha5, Czech Republic	Tel : +420-251-551-470 Fax : +420-251-551-471
Poland	MITSUBISHI ELECTRIC EUROPE B.V. Polish Branch ul. Krakowska 50, 32-083 Balice, Poland	Tel : +48-12-347-65-00 Fax : +48-12-630-47-01
Sweden	MITSUBISHI ELECTRIC EUROPE B.V. (Scandinavia) Fjelievägen 8, SE-22736 Lund, Sweden	Tel : +46-8-625-10-00 Fax : +46-46-39-70-18
Russia	MITSUBISHI ELECTRIC (RUSSIA) LLC St. Petersburg Branch Piskarevsky pr. 2, bld 2, lit “Sch”, BC “Benua”, office 720; 195027 St. Petersburg, Russia	Tel : +7-812-633-3497 Fax : +7-812-633-3499
Turkey	MITSUBISHI ELECTRIC TURKEY A.Ş Ümraniye Branch Serifali Mahallesi Nutuk Sokak No:5, TR-34775 Umraniye/Istanbul, Turkey	Tel : +90-216-526-3990 Fax : +90-216-526-3995
UAE	MITSUBISHI ELECTRIC EUROPE B.V. Dubai Branch Dubai Silicon Oasis, P.O.BOX 341241, Dubai, U.A.E.	Tel : +971-4-3724716 Fax : +971-4-3724721
South Africa	ADROIT TECHNOLOGIES 20 Waterford Office Park, 189 Witkoppen Road, Fourways, South Africa	Tel : +27-11-658-8100 Fax : +27-11-658-8101
China	MITSUBISHI ELECTRIC AUTOMATION (CHINA) LTD. No.1386 Hongqiao Road, Mitsubishi Electric Automation Center, Shanghai, China	Tel : +86-21-2322-3030 Fax : +86-21-2322-3000
Taiwan	SETSUYO ENTERPRISE CO., LTD. 6F, No.105, Wugong 3rd Road, Wugu District, New Taipei City 24889, Taiwan	Tel : +886-2-2299-2499 Fax : +886-2-2299-2509
Korea	MITSUBISHI ELECTRIC AUTOMATION KOREA CO., LTD. 7F-9F, Gangseo Hangang Xi-tower A, 401, Yangcheon-ro, Gangseo-Gu, Seoul 07528, Korea	Tel : +82-2-3660-9530 Fax : +82-2-3664-8372
Singapore	MITSUBISHI ELECTRIC ASIA PTE. LTD. 307, Alexandra Road, Mitsubishi Electric Building, Singapore 159943	Tel : +65-6473-2308 Fax : +65-6476-7439
Thailand	MITSUBISHI ELECTRIC FACTORY AUTOMATION (THAILAND) CO., LTD. 12th Floor, SV.City Building, Office Tower 1, No. 896/19 and 20 Rama 3 Road, Kwaeng Bangpongpang, Khet Yannawa, Bangkok 10120, Thailand	Tel : +66-2682-6522 Fax : +66-2682-6020
Vietnam	MITSUBISHI ELECTRIC VIETNAM COMPANY LIMITED Hanoi Branch 6th Floor, Detech Tower, 8 Ton That Thuyet Street, My Dinh 2 Ward, Nam Tu Liem District, Hanoi, Vietnam	Tel : +84-4-3937-8075 Fax : +84-4-3937-8076
Indonesia	PT. MITSUBISHI ELECTRIC INDONESIA Gedung Jaya 11th Floor, JL. MH. Thamrin No.12, Jakarta Pusat 10340, Indonesia	Tel : +62-21-3192-6461 Fax : +62-21-3192-3942
India	MITSUBISHI ELECTRIC INDIA PVT. LTD. Pune Branch Emerald House, EL-3, J Block, M.I.D.C., Bhosari, Pune-411026, Maharashtra, India	Tel : +91-20-2710-2000 Fax : +91-20-2710-2100
Australia	MITSUBISHI ELECTRIC AUSTRALIA PTY. LTD. 348 Victoria Road, P.O. Box 11, Rydalmere, N.S.W 2116, Australia	Tel : +61-2-9684-7777 Fax : +61-2-9684-7245

Mitsubishi Electric Corporation Nagoya Works is a factory certified for ISO 14001 (standards for environmental management systems) and ISO 9001(standards for quality assurance management systems)

EC 98 J 2017 0051

ISO 9001 BUREAU VERITAS Certification 0008

MITSUBISHI ELECTRIC CORPORATION

HEAD OFFICE: TOKYO BUILDING, 2-7-3, MARUNOUCHI, CHIYODA-KU, TOKYO 100-8310, JAPAN
NAGOYA WORKS: 1-14, YADA-MINAMI 5, HIGASHI-KU, NAGOYA, JAPAN

L(NA)08587ENG-B 1903(KWIX)

New publication effective Mar. 2019.
Specifications subject to change without notice.