MITSUBISHI ELECTRIC
Changes for the Better

for a greener tomorrow

FATEC

Mitsubishi Programmable Controller Training Manual MELSEC iQ-F Series Basic Course (for GX Works3)

● Safety Precautions ●

(Be sure to read this before the training.)

Before designing a system, be sure to read this manual and pay close attention to safety.
During the training, pay attention to the following points to ensure correct handling.

[Precautions for Training]

⚠WARNING

- To prevent electric shock, do not touch the terminals while they are powered ON.
- Before removing safety covers, either turn the power supply OFF or confirm safety.
- Do not put your hand into moving parts.

⚠CAUTION

- Proceed with the training under the guidance of a teacher.
- Connect the power plug with grounding terminal to the 3-pole socket.
  When using a 3-pole to 2-pole Conversion adapter, connect the grounding wire on the Conversion adapter to the grounding terminal on the socket.
- Do not remove the training machine module or change the wiring without permission. Doing so may result in malfunction, misoperation, injury or fire.
- Before attaching or detaching the module, turn the power OFF.
  Attaching or detaching the module while it is still ON may cause the module to malfunction or cause an electric shock.
- If unusual odor or abnormal noise is detected with the training machine immediately turn the power switch to OFF.
- If an abnormal event occurs, immediately contact your teacher.

INTRODUCTION

This manual covers general information on the basics to the application of the MELSEC iQ-F Series.
The contents include explanations on the PLC, and explanations on the instructions with examples.
The PLC is explained based on the FX5U.
Persons who do not have a basic understanding of the PLC should prepare by reading the "Your First PLC Introduction" FX-NYUM-TEXT textbook.

The following related materials are available.

(1) Your First PLC Introduction JY997D22101

(2) MELSEC iQ-F FX5 User's Manual (Startup) JY997D58201

(3) MELSEC iQ-F FX5U User's Manual (Hardware) JY997D55301

(4) MELSEC iQ-F FX5 User's Manual (Application) JY997D55401

(5) MELSEC iQ-F FX5 Programming Manual (Program Design) JY997D55701

(6) MELSEC iQ-F FX5 Programming Manual
(Instructions, Standard Functions/Function Blocks) JY997D55801

(7) MELSEC iQ-F FX5 User's Manual (Analog Control) JY997D60501

(8) GX Works3 Operation Manual SH-081215ENG

(9) MELSEC iQ-F Series iQ Platform-compatible PLC ..... L(NA)08428ENG

CONTENTS

CONTENTS

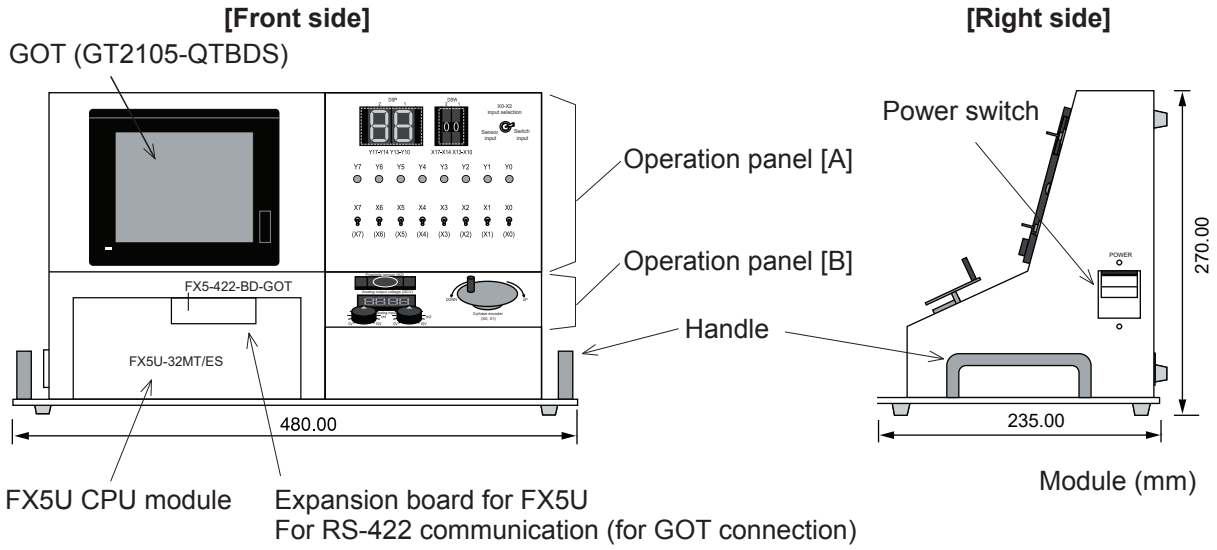
Chapter 1
INTRODUCTION

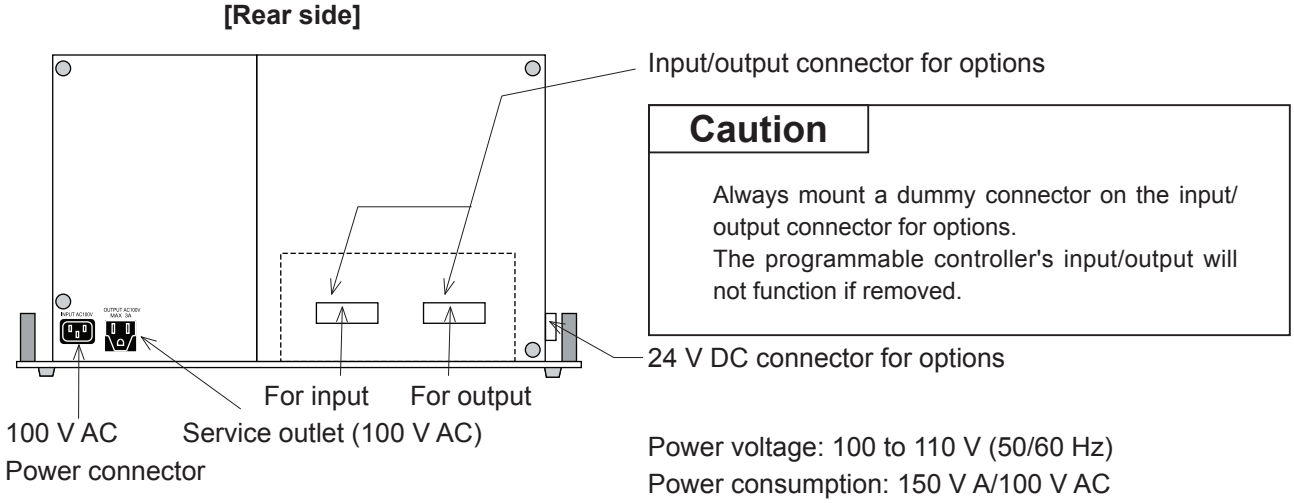
Let's understand the configuration of the training machine you will use!

The key to creating sequence programs is to first understand what equipment is connected to each input or output terminal of the PLC.
In this chapter, the configuration of the training machine will be described.

1.1 Configuration of the Training Machine (FX5U-32MT-SIM)

1.1.1 Appearance and Operation Panel

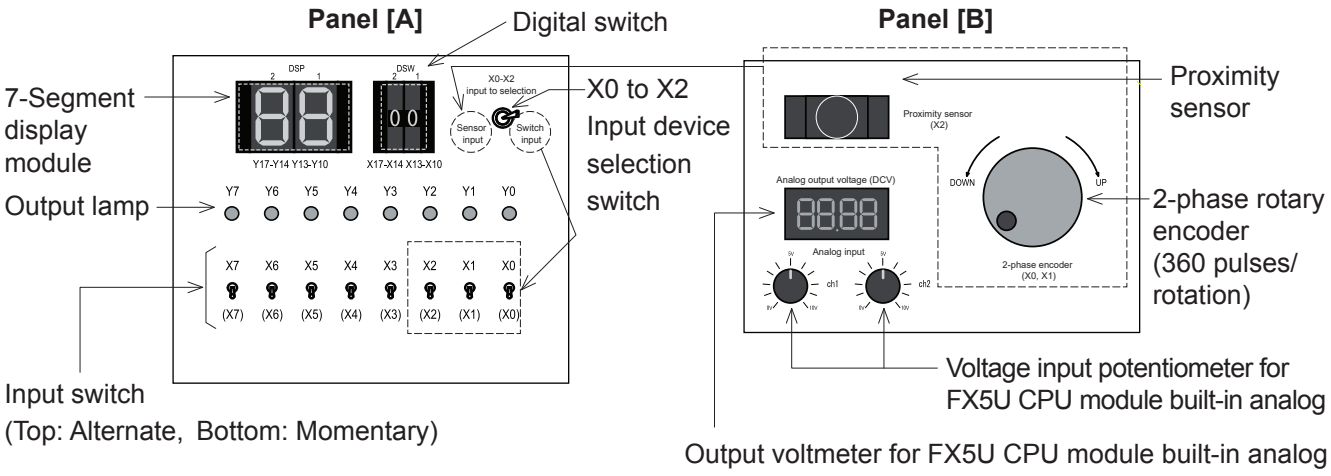




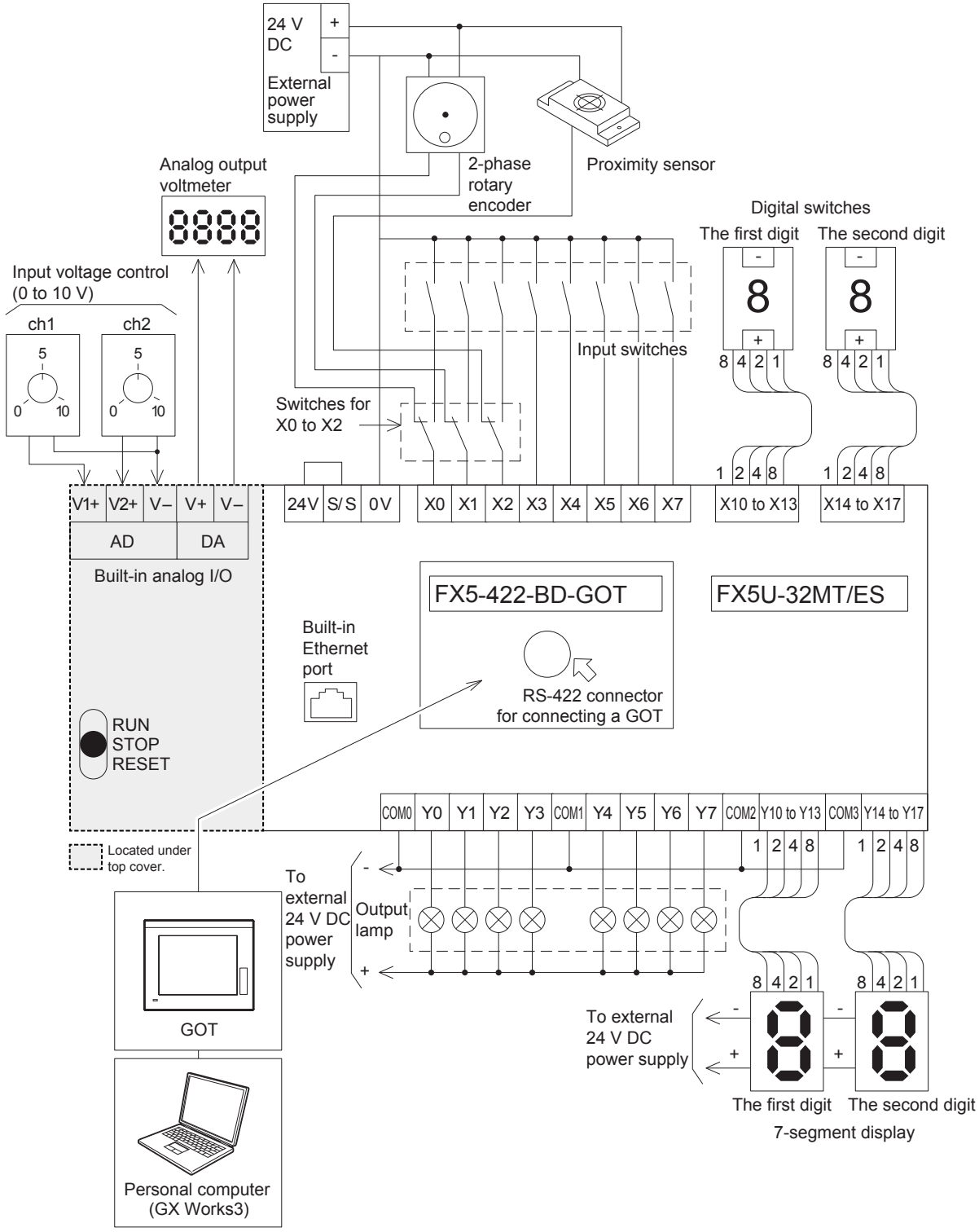
Caution

Always mount a dummy connector on the input/output connector for options.
The programmable controller's input/output will not function if removed.

Power voltage: 100 to 110 V (50/60 Hz)
Power consumption: 150 V A/100 V AC
Rush current: maximum 40 A 7.3 ms/100 V AC
Exterior color: Munsell 0.08GY/7.64/0.81
Weight: approx. 10 kg



1.1.2 External I/O Assignment and Wiring



<<Note>> Handling the S/S terminal

- When the S/S terminal and 24 V terminal are connected, the sink input is activated.
- When the S/S terminal and 0 V terminal are connected, the source input is activated.

MEMO

Chapter 2
GUIDING COURSE: DO YOU REMEMBER?

The Definition of a PLC…

A Programmable Controller (PLC) is also referred to as a Programmable Controller or Sequence Controller.
A PLC is defined as "an electronic device which controls many types of systems through its I/O ports and incorporates a memory to store programmable instructions."

Actual Usages…

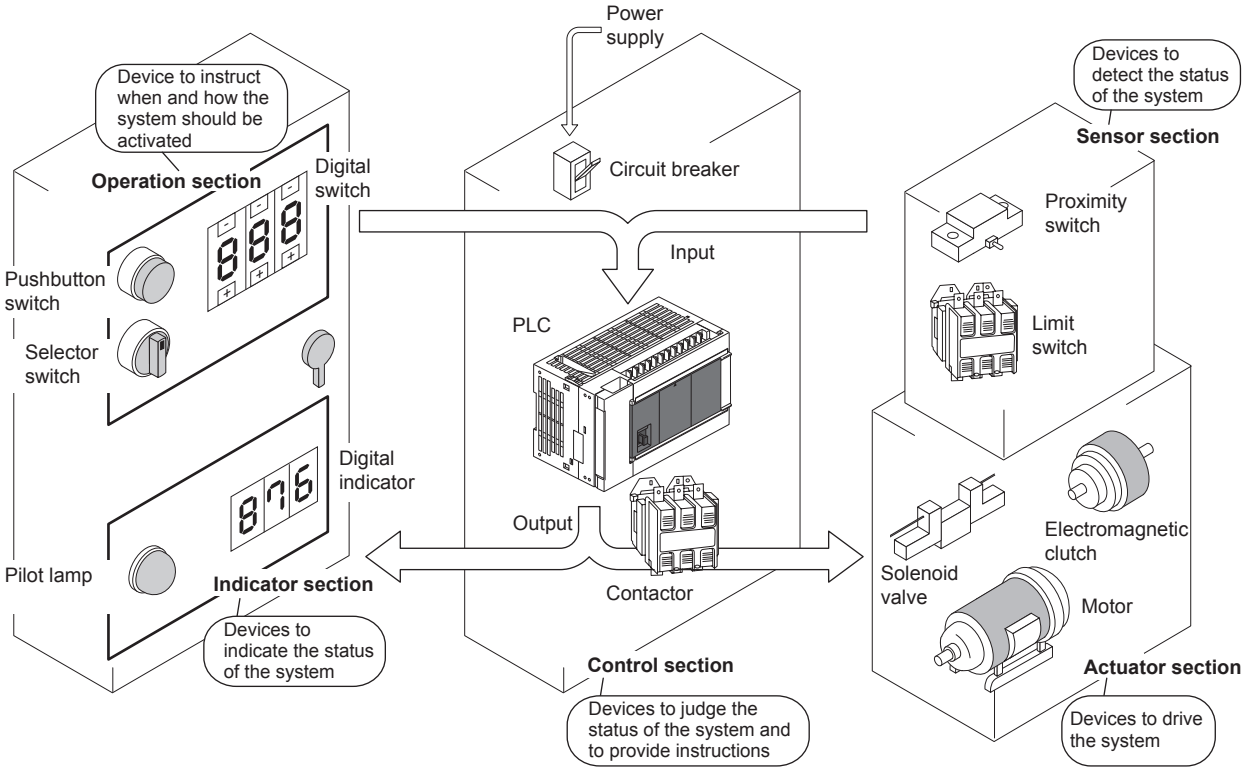
PLCs are broadly used as core components for FA (Factory Automation) and as electronic application products essential for saving labor costs and improving automation.
PLCs can be used for many types of applications such as systematic applications which control the entire factory, standalone types that control various dispersed machines independently, and PLCs for non-FA applications such as for leisure devices.

In this chapter…

The functions, construction, and features of PLCs, mainly in regard to small standalone PLCs, are described in a summarized manner.

2.1 PLC - Small, reliable, flexible brain

An automation solution for the machining, assembly, transfer, inspection, and packaging of a workpiece



The PLC is activated by **command inputs** such as inputs from pushbutton switches, selector switches and digital switches located at the operating panel, and by **sensor inputs**, such as inputs from limit switches, proximity switches and photoelectric switches, which detect the status of the system, in order to control **drive loads** such as solenoid valves, motors and electromagnetic clutches, and **indication loads** such as pilot lamps and digital indicators.

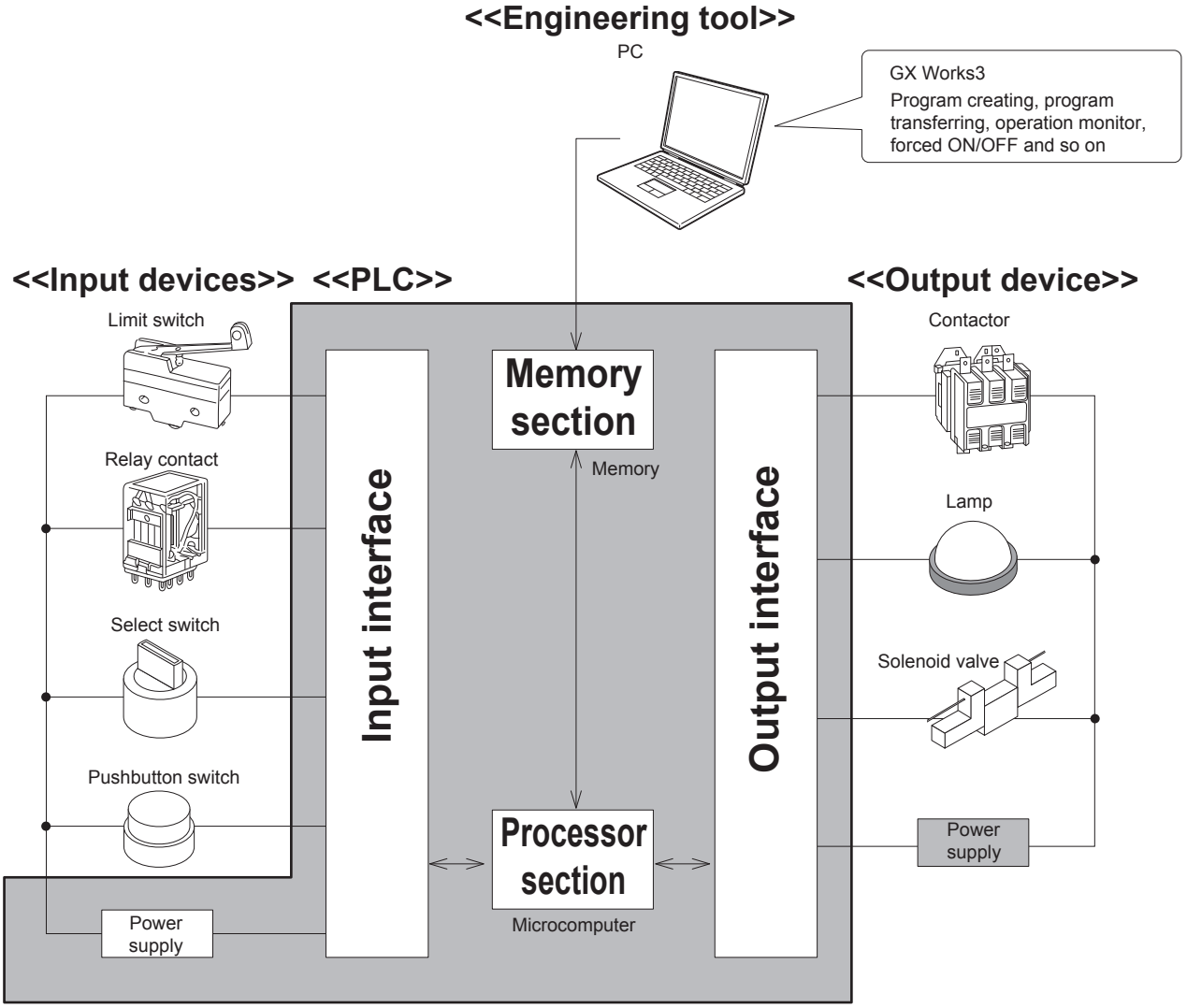
The behaviors of output signals corresponding to the input signals are determined by the contents of programs provided to the PLC.

Light loads such as small solenoid valves and pilot lamps can be directly driven by a PLC, but loads such as 3-phase motors and large solenoid valves must be driven through contactors and intermediate relays.

As well as PLCs, contactors, intermediate relays and circuit breakers for the power supply are installed in the control box.

2.2 Mechanism of PLC

2.2.1 The PLC is a microcomputer for industrial purposes.



A PLC incorporates an electrical circuit mainly comprised of a microcomputer and memory.
Input/output interfaces exist between input/output devices and the electronic circuit to connect them.
The engineering tool is used to write a program to the memory in the PLC.

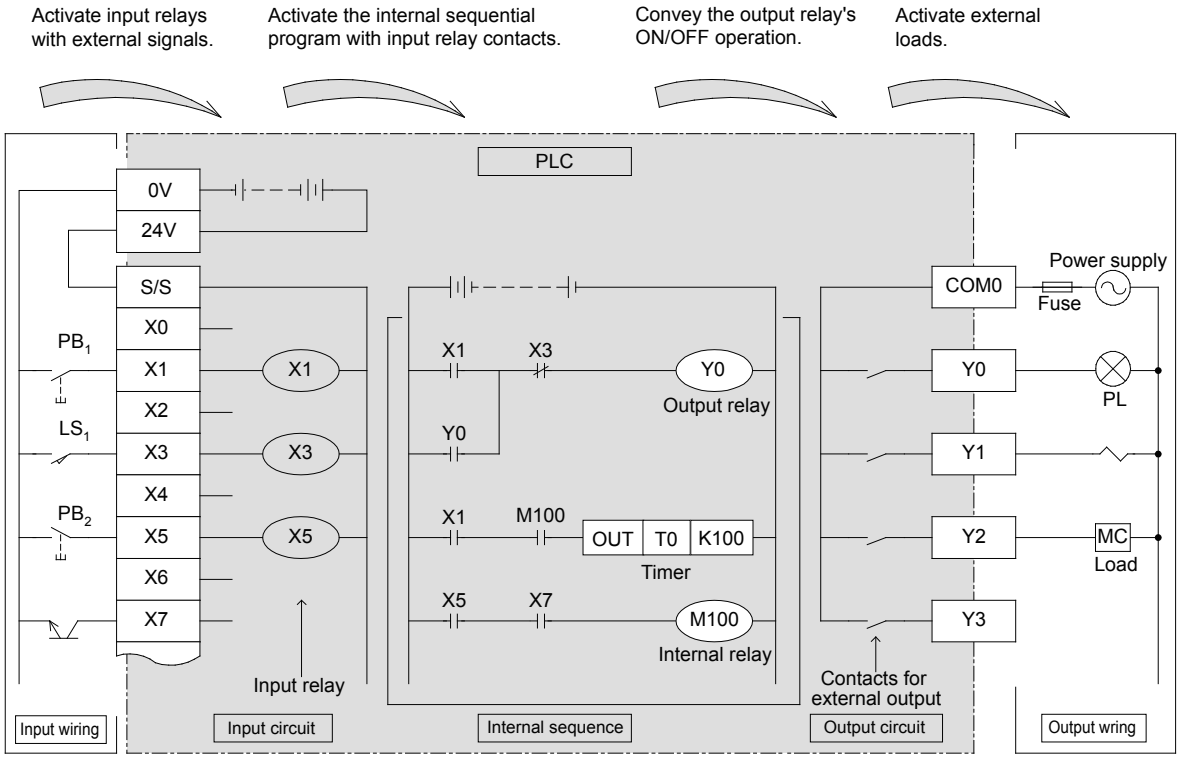
Reference

Is the term "sequencer" coined by Mitsubishi Electric?

In Japan, the term "sequencer" is widely used. While Japan Electrical Manufacturer's Association (JEMA) officially names them Programmable Logic Controller (PLC), the name "sequencer" seems easier to pronounce and more widely known.

Though there is evidence that the term "sequencer" was used before PLCs were invented, it is a fact that Mitsubishi Electric made it popular by releasing K and F series PLCs with the name of "sequencer."

2.2.2 The PLC can be, in effect, considered as an aggregate of relays and timers.



The PLC is an electronic device mainly comprised of a microcomputer.

However in effect…

The user does not need any knowledge of a microcomputer to operate a PLC and it can be regarded as an aggregate of relays, timers, and counters.

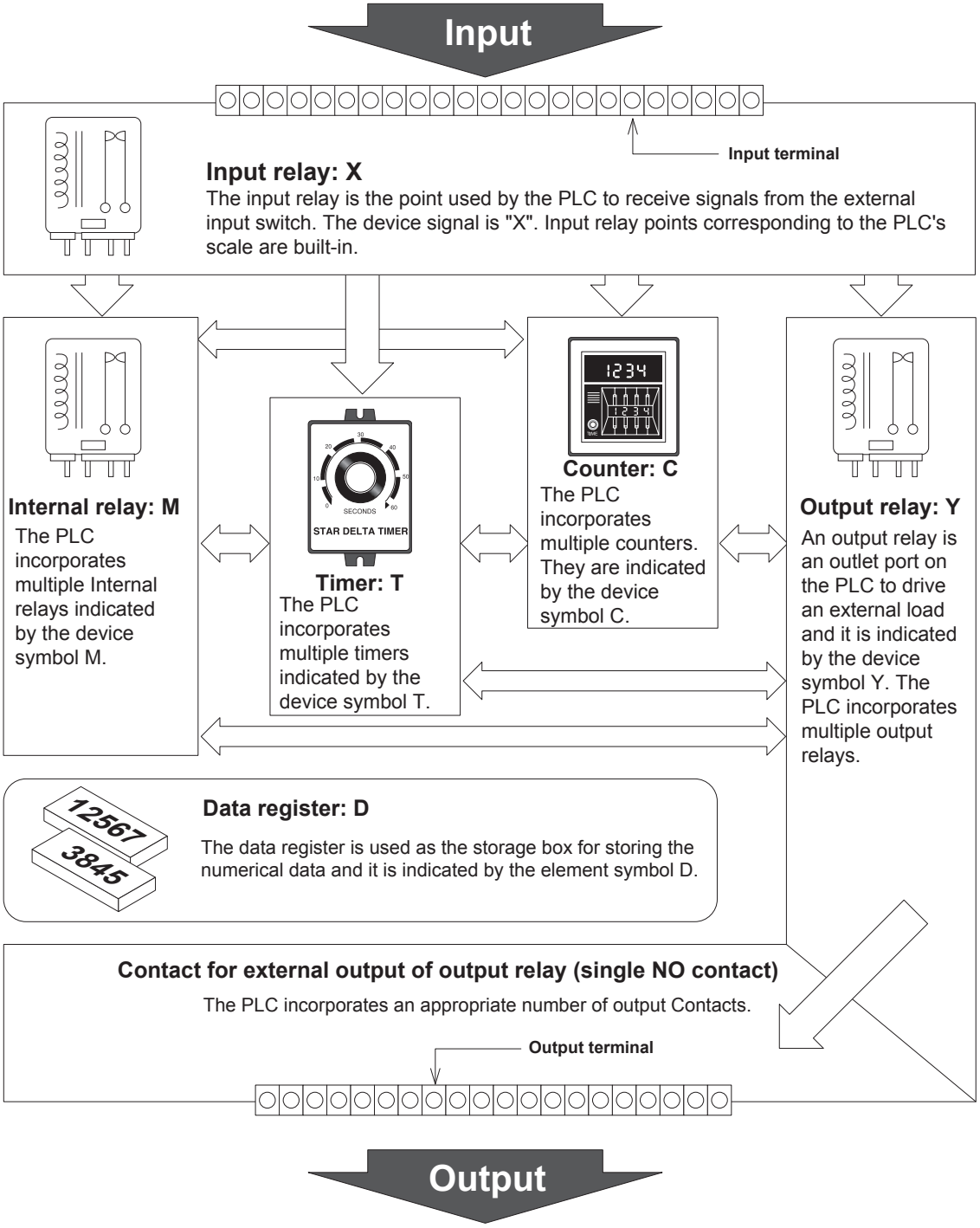
Internal operation of PLC

Signal Flow of PLC

- When the pushbutton switch PB1 is pushed, the coil of the input relay X1 is energized.
- When the coil of the input relay X1 is energized, the NO contact of X1 is closed and the coil of the output relay Y0 is energized.
- When the coil of the output relay Y0 is energized, the NO contact of Y0 is closed, then the pilot lamp PL is illuminated.
- When the pushbutton switch PB1 is released, the coil of the input relay X1 is de-energized and the NO contact of X1 is opened.
  But the output relay Y0 is still energized since the NO contact is closed. (Self-maintaining action)
- When the input relay X3 is energized by closing the limit switch LS1, the NC contact of X3 is opened, then the coil of the output relay Y0 is de-energized (Reset). As a result, the pilot lamp PL turns off and the self-maintaining action of the output relay Y0 is cleared.

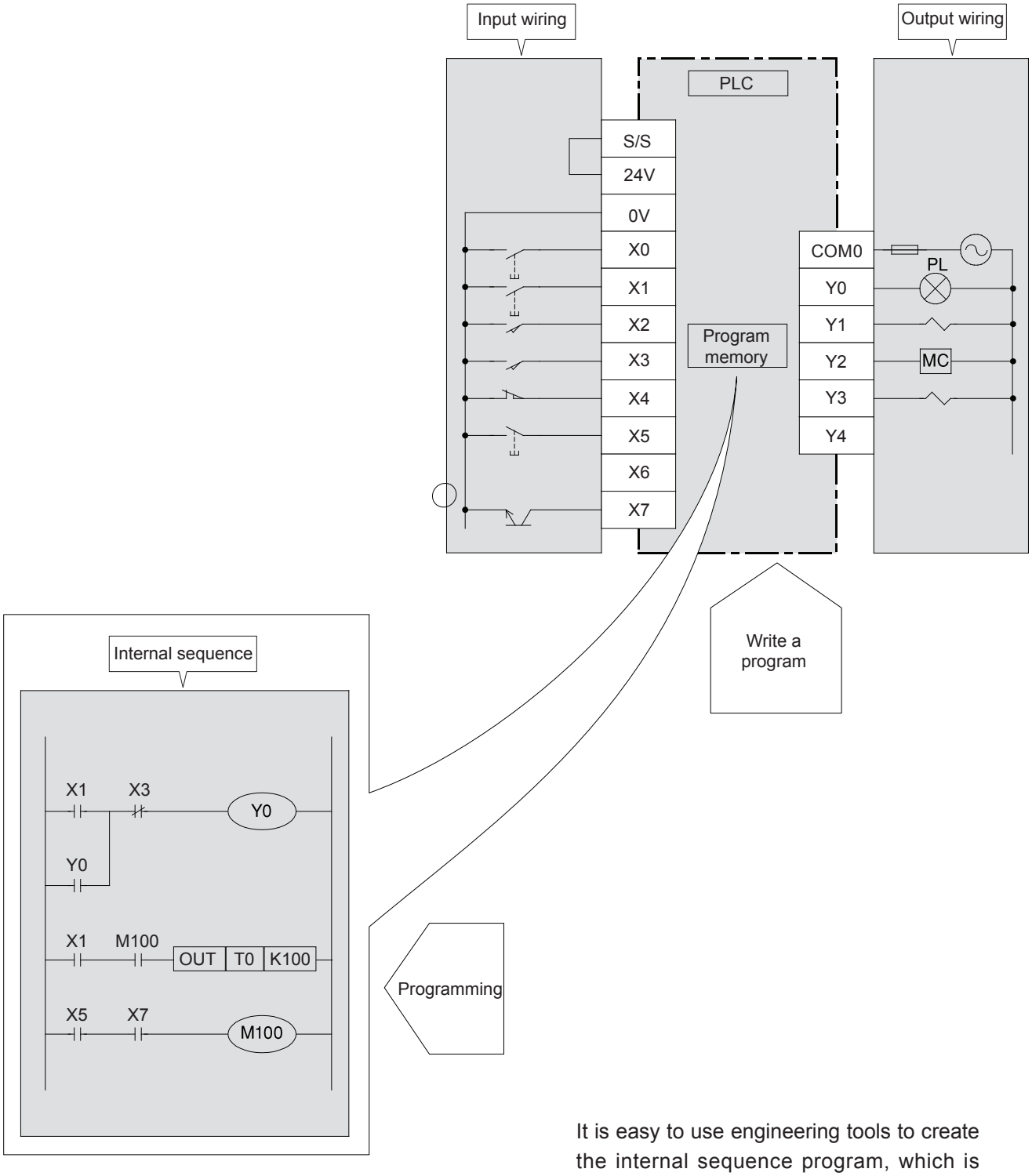
2.2.3 Types of relay and timers

As shown below, a PLC incorporates multiple relays, timers and counters with countless NO and NC contacts. A sequential circuit is formed by connecting the contacts and coils.
The PLC, which is a microcomputer application device, features multiple value data storage boxes called "data registers".



2.3 Wiring and instructions

Perform the wiring work for input and output devices.



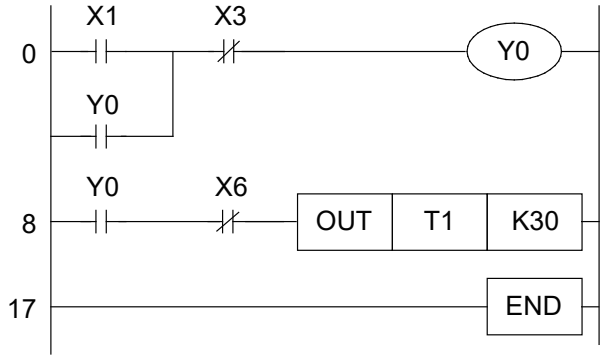
It is easy to use engineering tools to create the internal sequence program, which is equivalent to internal PLC wiring.

2.4 Instruction and programs

2.4.1 Mechanism of programs

The internal sequence for the sequence control is created as a sequence program in the format of circuit diagram (ladder diagram) and instruction list.

Circuit diagram (ladder diagram)



Instruction list (program list)

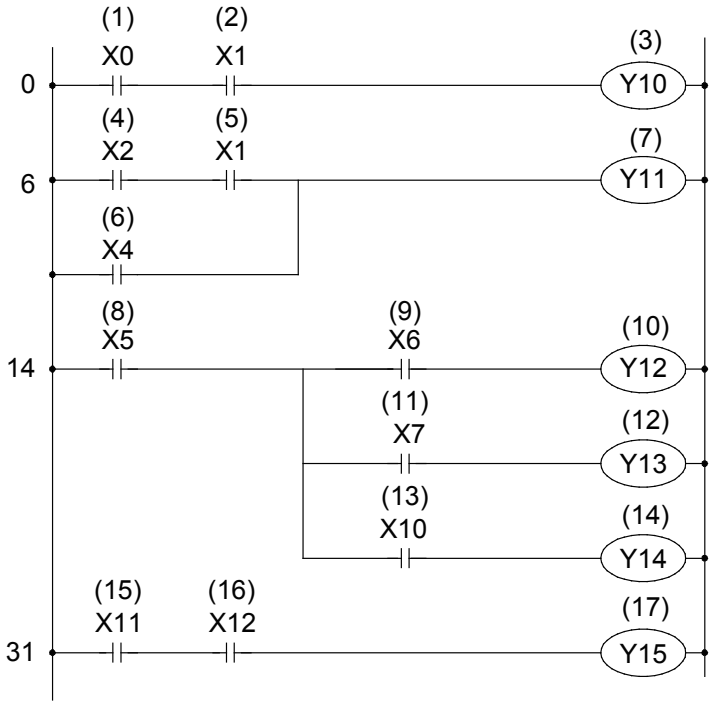
Step number	Instruction	
	Instruction code	Device (number) (operand)
0	LD	X1
2	OR	Y0
4	ANI	X3
6	OUT	Y0
8	LD	Y0
10	ANI	X6
12	OUT	T1 K30
17	END	

cyclic operation

- A program is comprised of multiple instruction codes and device numbers (operands).
  These instructions are numbered in turn. This number is referred to as **the step number**. (Step numbers are automatically controlled.)
- Each "**instruction**" is comprised of "**instruction code + device**". However, there are some instructions without devices.
  Also in some cases, instruction codes are just referred to as instructions.
- The max steps that can be programmed depend on the "program memory capacity" of the PLC that is used. This is called the **PLC program capacity**.
  There is a program memory with the capacity of "64000" steps in the FX5U CPU module.
- The PLC repeatedly performs instructions from step 0 to the END instruction. This operation is referred to as **cyclic operation**, and the time required to perform one cycle is referred to as **the operation cycle (scan time)**.
  The operation cycle will change according to the contents of the programs and the actual operating orders, ranging from several msec to several tens of msec.

2.4.2 The processing order of the program

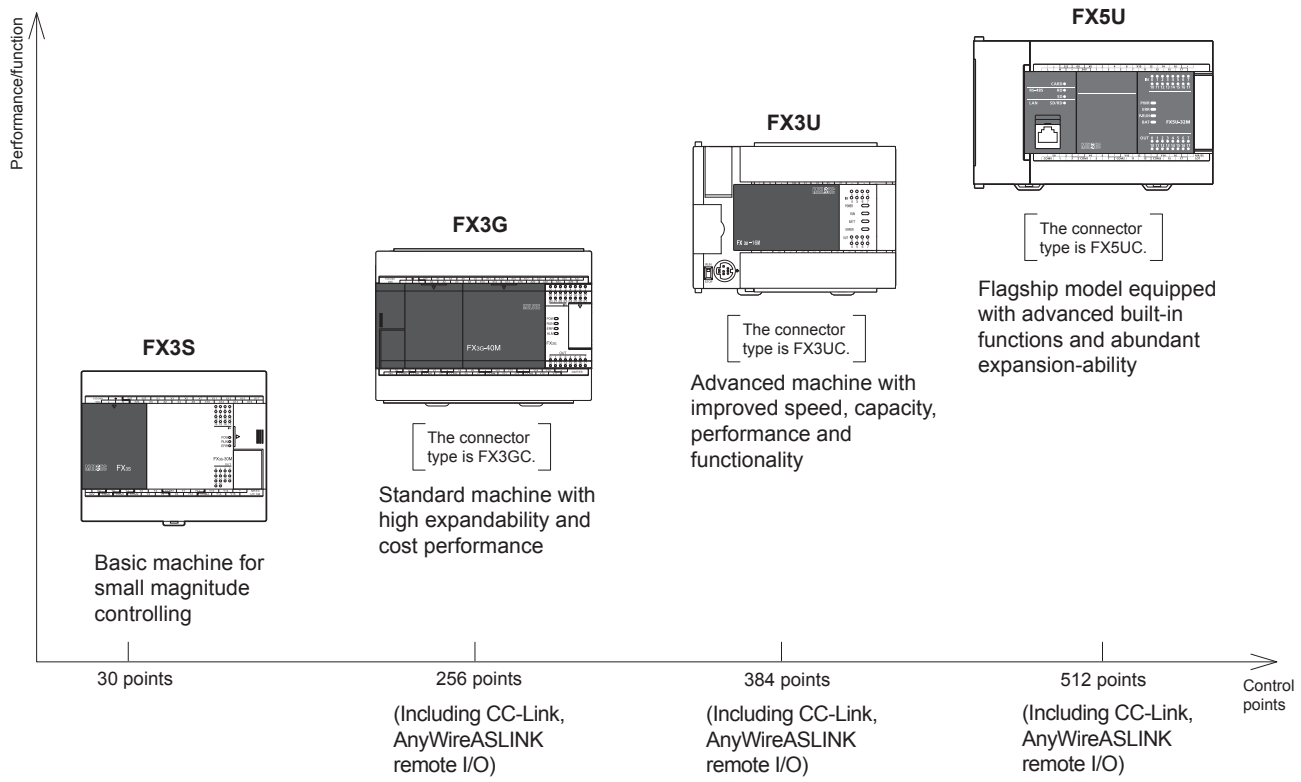
The program processes sequentially from the first step to the end of the program memory in block modules with the order from left to right, up to down. (in order of (1), (2), ……(17))



2.5 The configuration of a micro PLC

2.5.1 Brief introduction of the CPU module

A micro PLC is a standalone module that can be easily used as a PC, so it has a series of advantages such as high speed, high performance and good expandability.



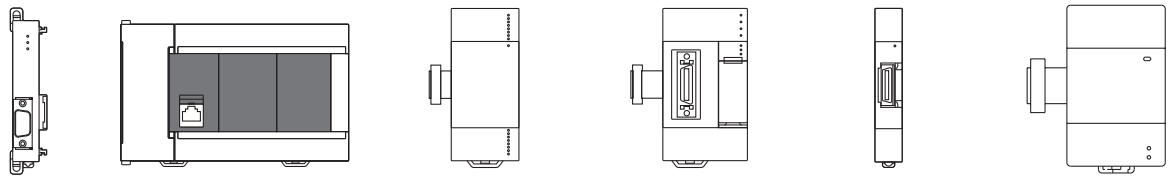
■ Function list (provided on the screw terminal type PLC)

✓: Available –: Unavailable

Function	Terminal block type			
	FX3S	FX3G	FX3U	FX5U
Memory capacity (step)	16000	32000	64000	64000
Input/output extension	–	✓	✓	✓
Intelligent function module connection	–	✓	✓	✓
Extension board installation	✓	✓	✓	✓
Special adapter	✓	✓	✓	✓
Display module installation	✓	✓	✓	–
Built-in high speed counter function	✓	✓	✓	✓
High speed processing by input interrupt/ pulse catch function	✓	✓	✓	✓
High speed processing by timer interrupt/ counter interrupt function	Timer interrupt enabled	Timer interrupt enabled	✓	✓
Built-in real time clock (clock function)	✓	✓	✓	✓
Built-in analog volume	✓	✓	–	–
Built-in 24 V DC service power supply	✓	✓	✓	✓
Constant scan function	✓	✓	✓	✓
Input filter adjustment function	✓	✓	✓	✓
Comment registration function	✓	✓	✓	✓
Function modifying the program during RUN	✓	✓	✓	✓
Built-in RUN/STOP switch	✓	✓	✓	✓
Function protecting the program by keywords	✓	✓	✓	✓
Built-in Ethernet function	–	–	–	✓

2.5.2 The basic configuration of the system

The configuration of a micro PLC will be described by taking an example using the FX5U CPU module.



[Expansion adapter]
Adapter expanding the functions such as communication and analog

[Space with the function expansion board]
The space in the CPU module for expansion with a communication function extension board.

[Main module (PLC)]
The PLC with "CPU, Input/output, power supply, and memory" all in one module

Power can be supplied to extension devices connected to the subsequent stage.

[I/O module]
Module that allows input/output to be expanded.
Multiple CPU modules and extension modules within the specified number can be connected

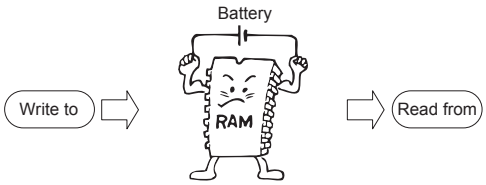
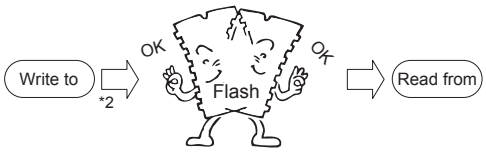
[Bus conversion module]
Module for connecting FX3 intelligent function module.

[Intelligent function module]
The extension eqipment controlling communication, analog, network and positioning

The types and the number of the equipments that can be connected depend on the series and the model name of the CPU module.

2.5.3 The types and advantages of the program memory

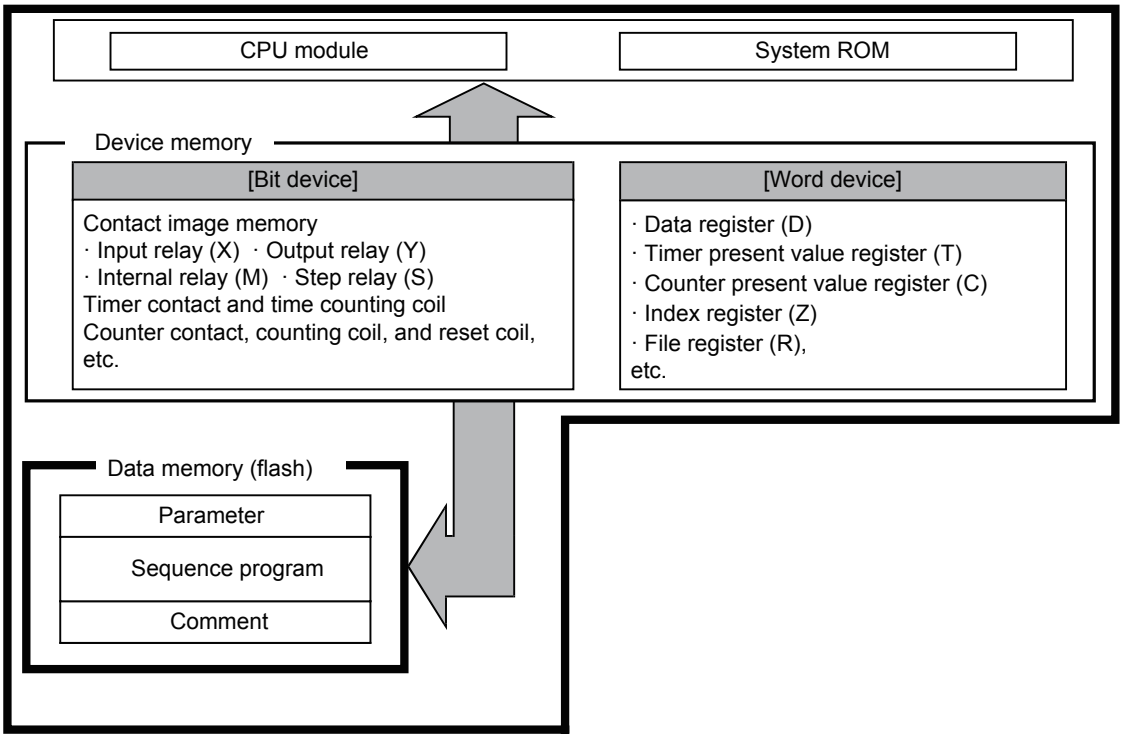
The following table lists the types of built-in program memories for micro PLCs.

Series	Built-in memory			Advantage
	Type	Memory capacity	Backup method	
 FX3S	EEPROM memory	16000 steps	Backup unnecessary	It is easy to write to/read from the memory, and battery backup is not required.  *1: The memory can be written up to 20000 times.
 FX3G/FX3GC		32000 steps		
 FX3U/FX3UC	RAM memory	64000 steps	Battery backup	It is easy to write/read at high speed. The content in the memory is stored by using the backup battery.  There are optional memories (EEPROM/ FLASH) which do not require battery backup. However, it is necessary to use the battery if the latch memory and clock function are used.
 FX5U/FX5UC	FLASH memory	64000 steps	Backup unnecessary	It is easy to write to/read from the memory, and battery backup is not required.  *2: The memory can be written up to 20000 times.

Reference

PLC's Memory Structure (FX5U CPU module)

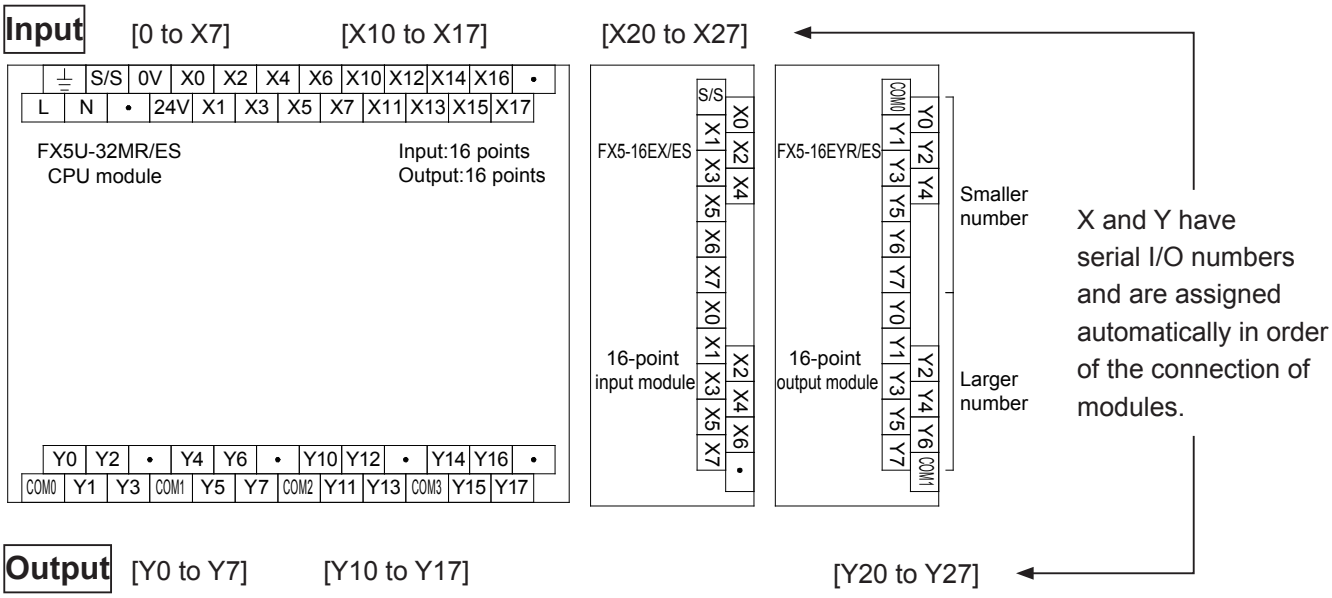
FX5U CPU module is supplied with flash memory.



2.5.4 Micro PLC I/O number assignment

Each CPU module has I/O numbers assigned by an octal number system such as X0 to X7, X10 to X17, Y0 to Y7, Y10 to Y17 and so forth.
The I/O module is assigned a serial No. that follows the CPU module.

[System configuration example and I/O numbers]



- I/O numbers of expansion modules are assigned with subsequent numbers to those of the CPU module with the I/O more adjacent to the CPU module having lower numbers.
  It is not necessary to set the parameters by using engineering tools.

MEMO

Chapter 3
THE OPERATION OF GX Works3

Using a personal computer, programming becomes easy…

GX Works3*1 software provides an efficient and easy way to create and edit sequence programs for PLCs.
Once the basic operations are mastered, programming often involves straightforward repetition.
This is software with many easy-to-use functions, but the required operations should be mastered in order first.

*1: GX Works3 is an engineering tool for setting, programming, debugging, and maintenance of projects for the programmable controllers including the MELSEC iQ-F series on a personal computer.

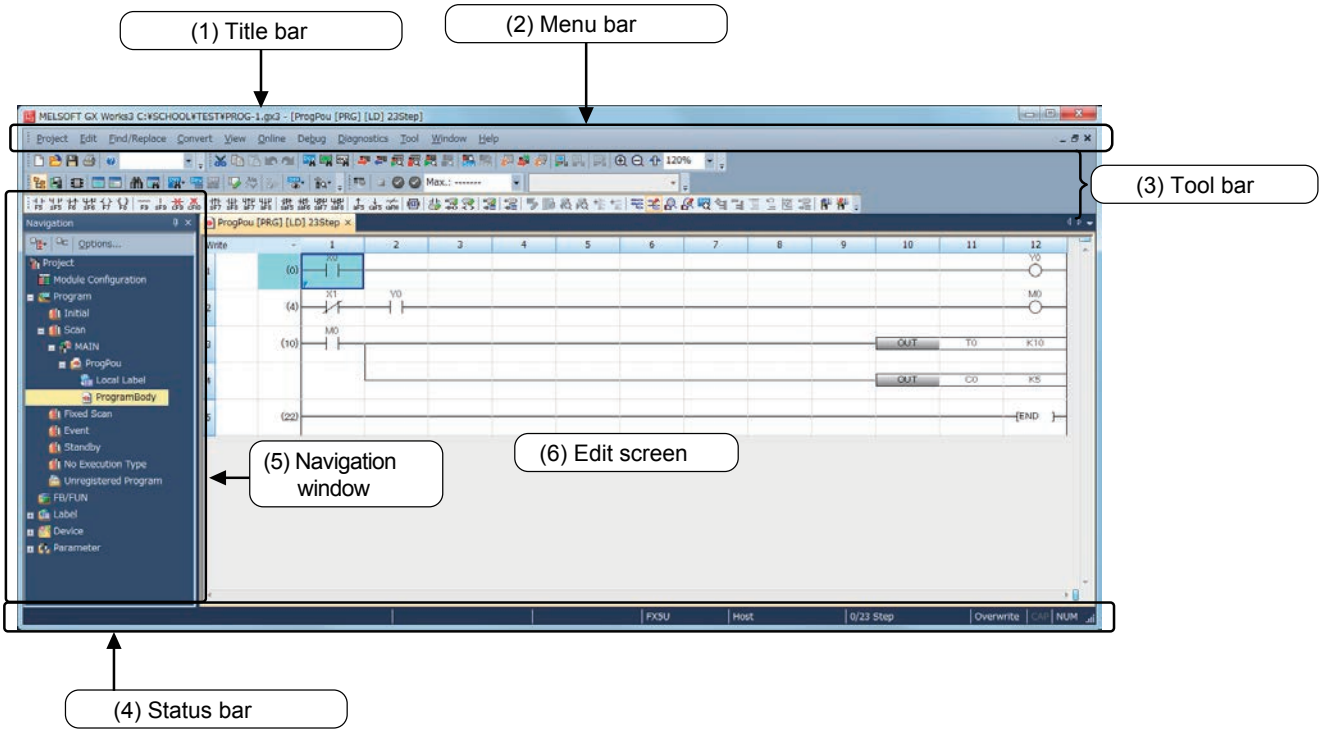
Smoothly begin new projects and update them with ease…

Programs require debugging.
The operation status of the PLC and program can be monitored with the personal computer screen, so if some parts are not working as planned, changes and updates can be conducted at once.

Make the program easy to read…

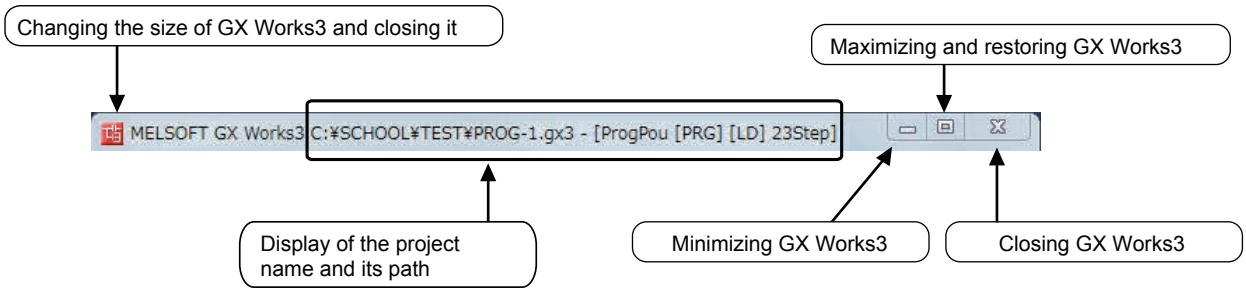
There is a "comment input function" in GX Works3 to make sequence programs easier to read.
Comments can improve the efficiency of creating and debugging ladder programs.

3.1 The layout of the GX Works3 screen

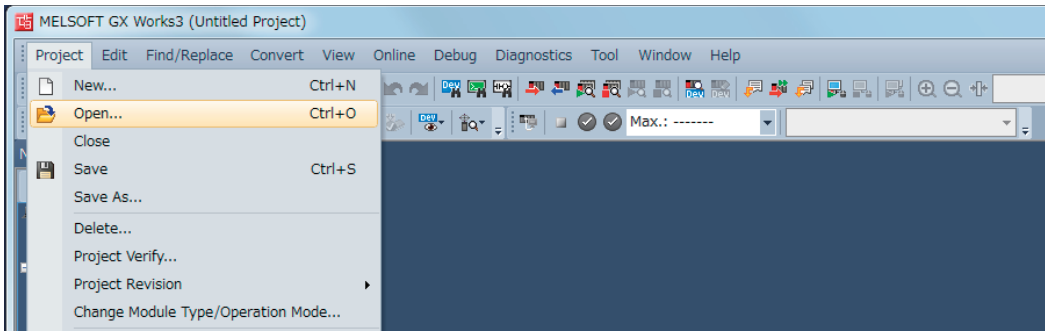


(1) Title bar

The name of the opened project and the window operation icons are displayed.

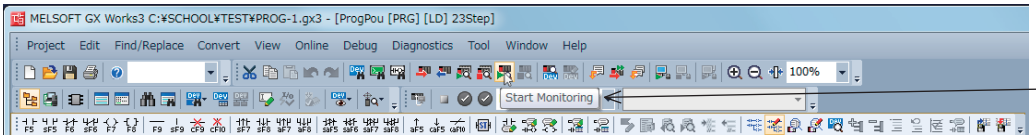


(2) Menu bar



Drop down menu items are displayed when a menu is selected.

(3) Tool bar*



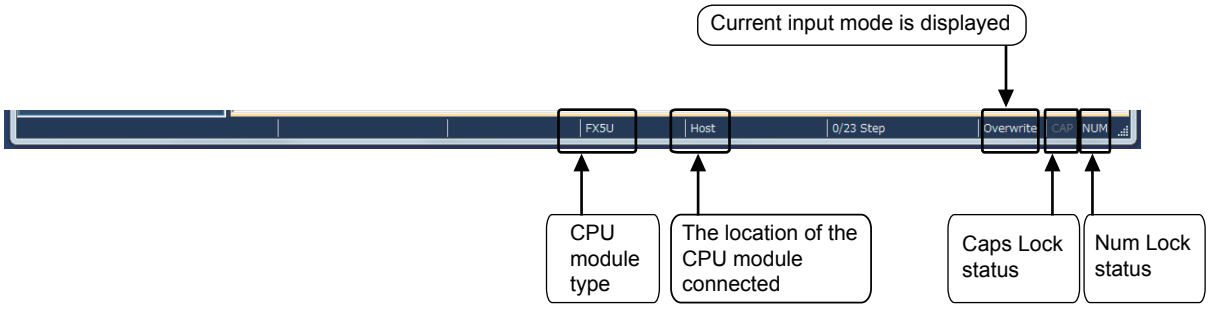
A description of the function is displayed when the mouse cursor stops over each button.

* : The contents of the tool bar can be moved, added, and removed. Therefore, the displayed items and layout depend on saved environments.

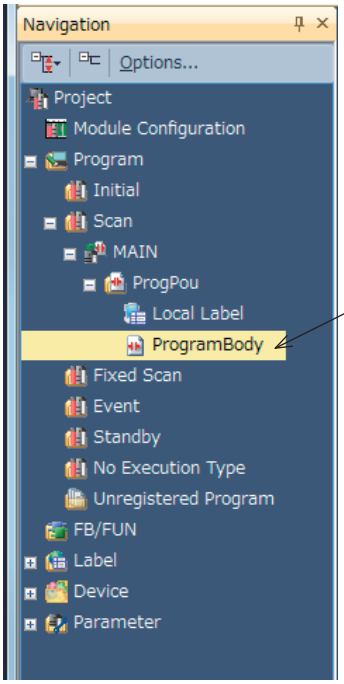
Frequently used functions are displayed with icon buttons. Compared to selecting from the menu bar, desired functions can be directly executed.

(4) Status bar

The status of the operation and keyboard settings are displayed.

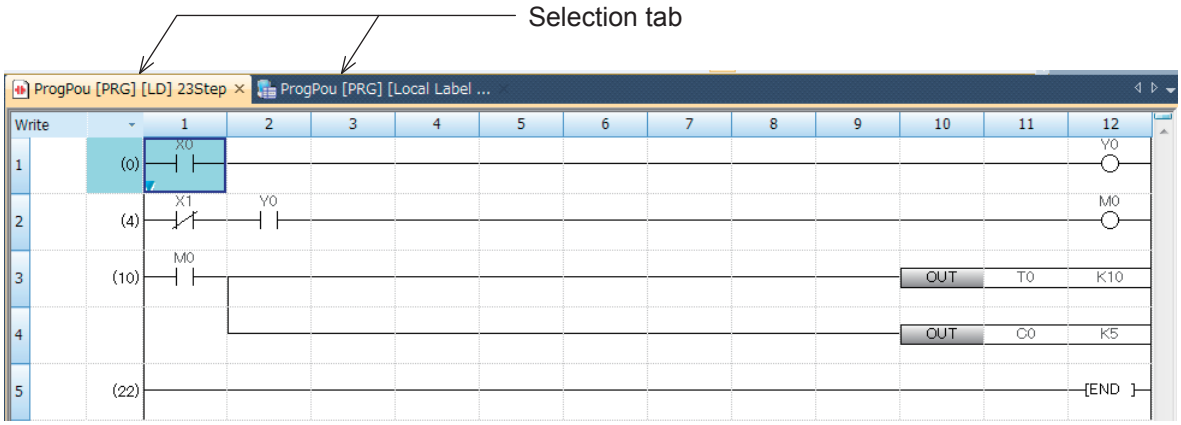


(5) Navigation window



Double-click [Program] → [Scan] → [MAIN] → [ProgPou] → [Program Body] on the navigation window. The sequence circuit diagram will appear.

(6) Edit screen

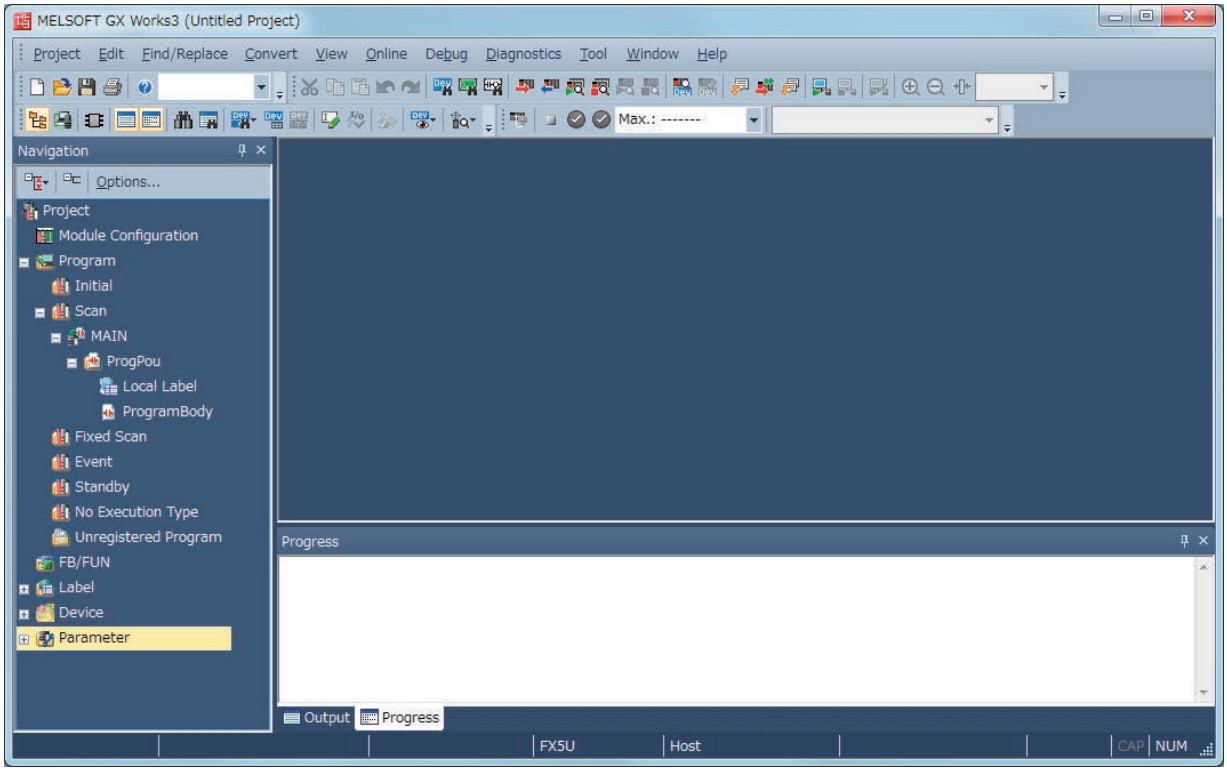


3.2 Starting GX Works3 and creating a new project

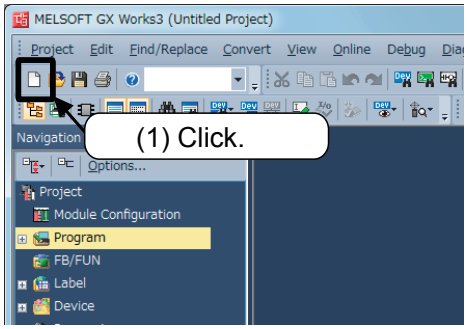
3.2.1 Starting GX Works3

Select [MELSOFT] → [GX Works3] → [GX Works3] from the Windows® Start menu*1.

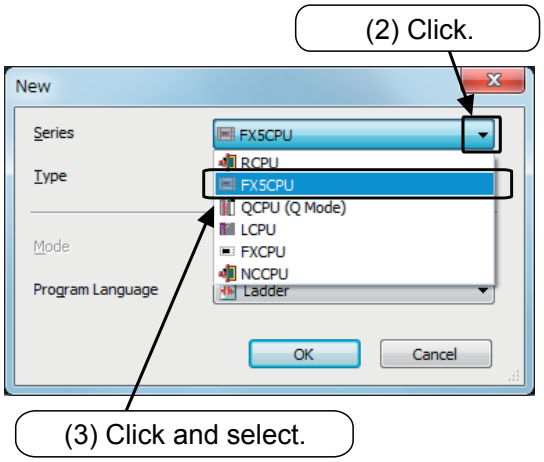
*1: Select [Start] → [All apps] or [Start] → [All Programs].



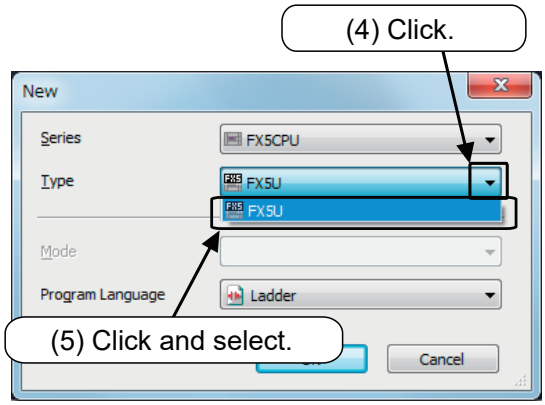
3.2.2 Creating a new project



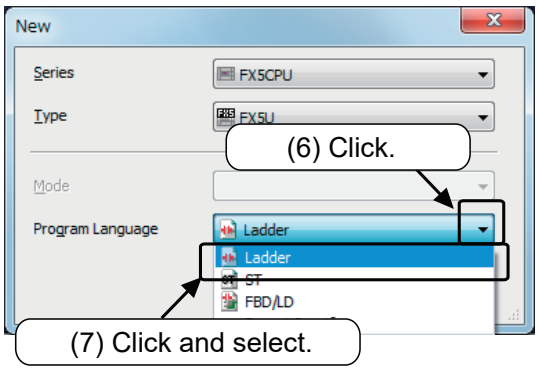
(1) Click  from the tool bar, or select [Project] → [New project] (Ctrl + N) from the menu bar.



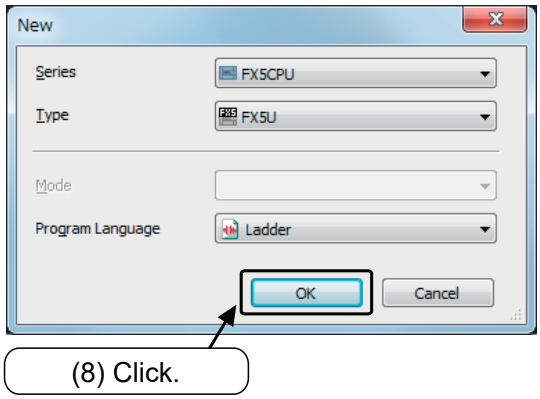
(2) Click the list button of "Series".
(3) Click and select "FX5CPU" from the drop-down menu.



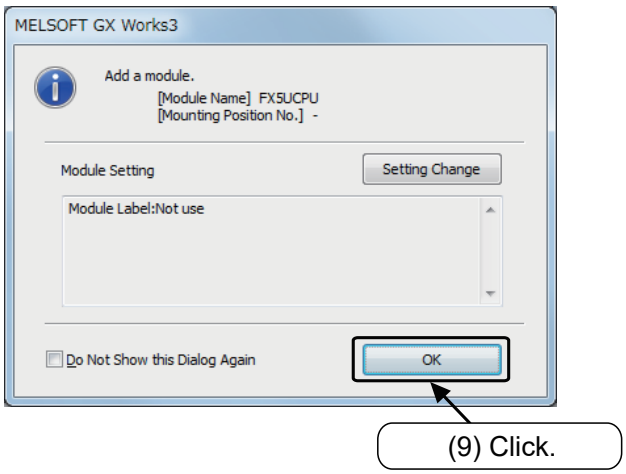
(4) Click the list button of "Type".
(5) Click and select "FX5U" from the drop-down menu.



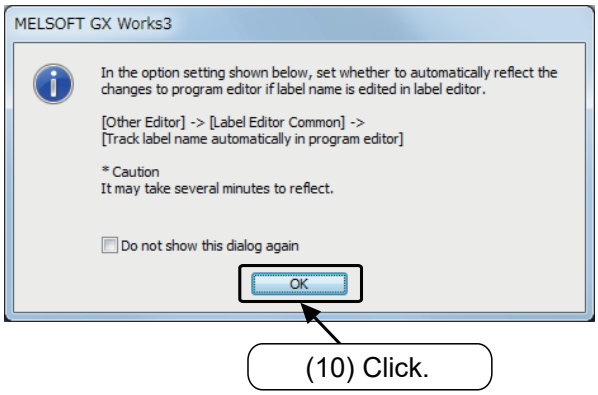
(6) Click the list button of "Program Language".
(7) Click and select "Ladder" from the drop-down menu.



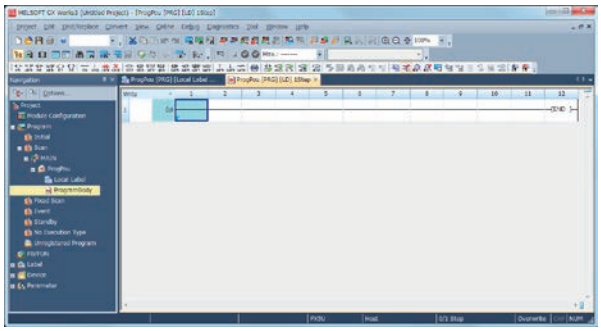
(8) Click the [OK] button.



(9) The confirmation window for adding the module label of the selected module type ("FX5U" in this case) appears. Confirm that "Module Label: Not use" is selected, and then click the [OK] button.
(If this screen does not appear, proceed to the next step.)



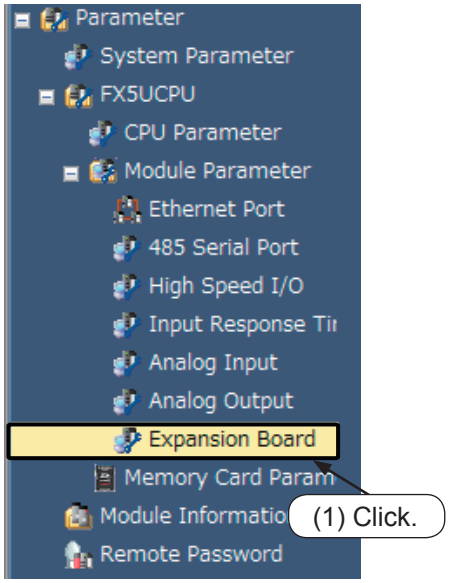
(10) The option function explanation screen will appear, so click the [OK] button.
(If this screen does not appear, proceed to the next step.)



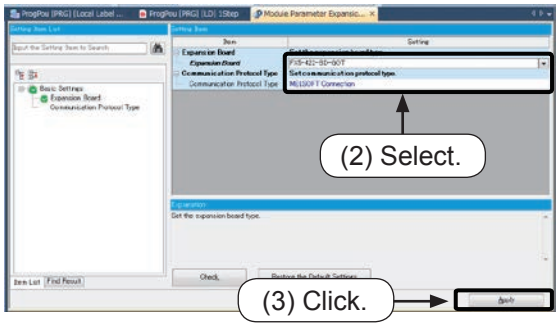
(11) A new project is created.

3.2.3 Setting the parameters

This chapter explains the parameter settings required to connect the GOT to the FX5-422-BD-GOT.



(1) Double-click [Parameter] → [FX5UCPU] → [Module Parameters] → [Expansion Board] on the navigation window.



(2) Select each item as shown below.
- Expansion board: FX5-422-BD-GOT
- Communication Protocol type

(3) After the setting is completed, click the [Apply] button.

Reference

Communication setting for Demonstration Machine

The Demonstration Machine (FX5U-32MT-SIM) communicates between the FX5U CPU module and PC using the GOT (direct connection) transparent function.
The FX5U CPU module and GOT settings are as shown below.

FX5U CPU module

Item		Description
Module Parameter, extension board	Extension board	FX5-422-BD-GOT
	Protocol format	MELSOFT connection

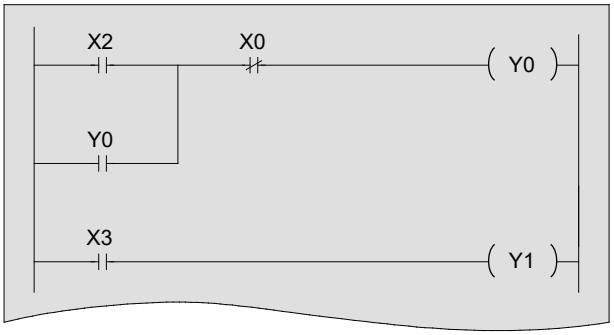
GOT

Item		Description
Connected device	Model	MELSEC iQ-F
	I/F	Standard I/F (RS422/485)
	Driver	Serial (MELSEC)

3.3 Creating a circuit

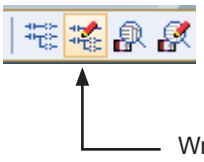
3.3.1 Creating a circuit by using the function keys

[The circuit to be created]



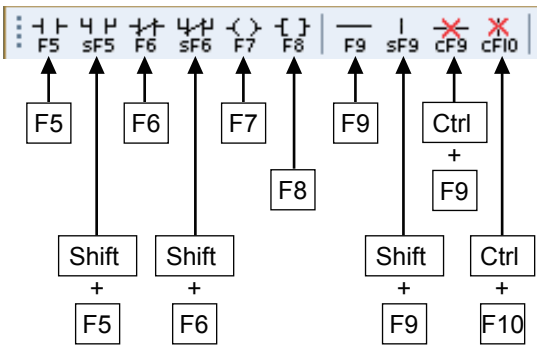
Point

- When creating a circuit, make sure to set the mode to "Write Mode".



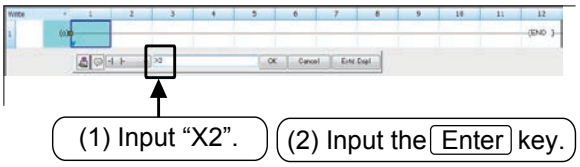
Write Mode (F2)

- The relationship between the function keys and the symbols of the circuit are displayed on the buttons of the tool bar.

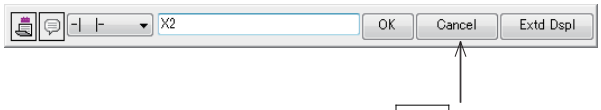


Main key operations

- Use only one-byte characters.

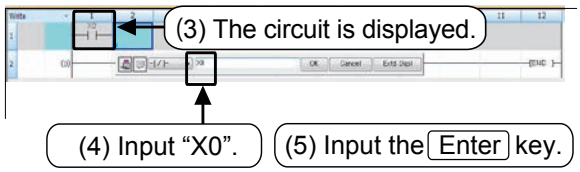


(1) Press the F5 (⊣⊢) key.
Input "X2".

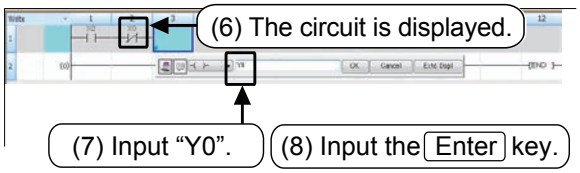


Cancel it by ESC or [Cancel].

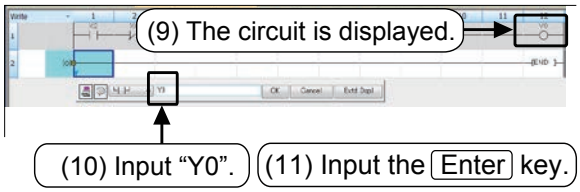
(2) Confirm by pressing the Enter key or [OK].



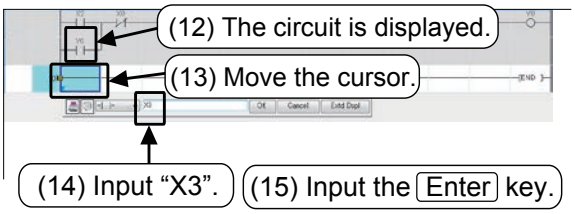
(3) The circuit input (X2 ⊣⊢) is displayed.
(4) Press the F6 (⊣/⊢) key.
Input "X0".
(5) Confirm by pressing the Enter key or [OK].



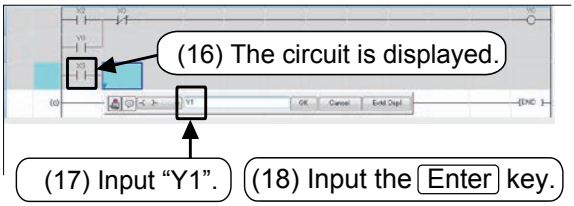
(6) The circuit input (X0 ⊣/⊢) is displayed.
(7) Press the F7 (⊣( )⊢) key.
Input "Y0".
(8) Confirm by pressing the Enter key or [OK].



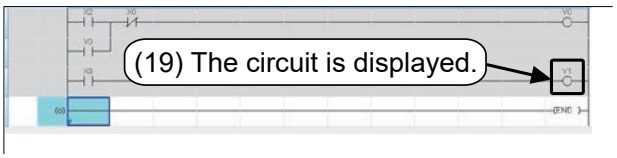
(9) The circuit input (⊣(Y0)⊢) is displayed.
(10) Press the Shift + F5 (⊣⊢) key.
Input "Y0".
(11) Confirm by pressing the Enter key or [OK].



(12) The circuit input （ Y0 ） is displayed.
(13) Move the cursor to the beginning of the next line.
(14) Press the F5 (⊣⊢) key.
Input "X3".
(15) Confirm by pressing the Enter key or [OK].



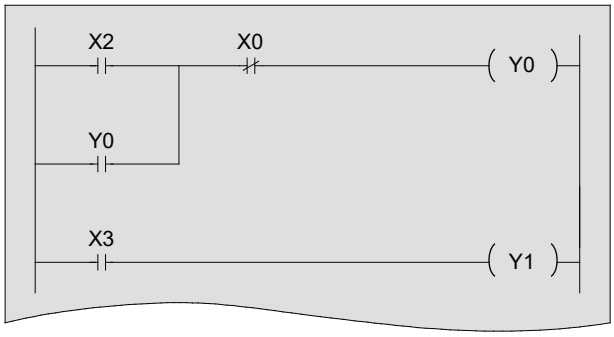
(16) The circuit input （ X3 ） is displayed.
(17) Press the F7 (-( )-) key.
Input "Y1".
(18) Confirm by pressing the Enter key or [OK].



(19) The circuit input (-(Y1)-) is displayed.
The circuit is created!!

3.3.2 Creating a circuit by using the tool buttons

[The circuit to be created]

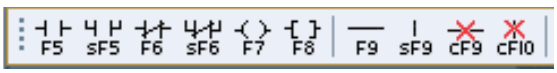


Point

- When creating a circuit, make sure to set the mode to "Write Mode".

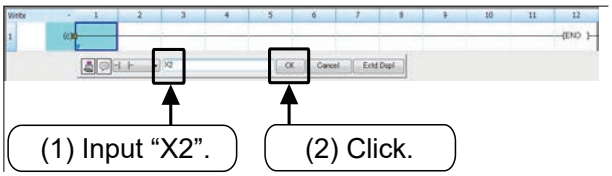
Write Mode (F2)

- Click the tool buttons to input the symbols of the circuit.

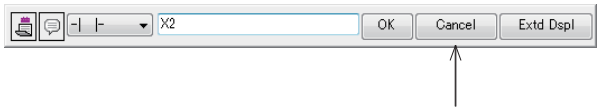


Main tool buttons

- Use only one-byte characters.

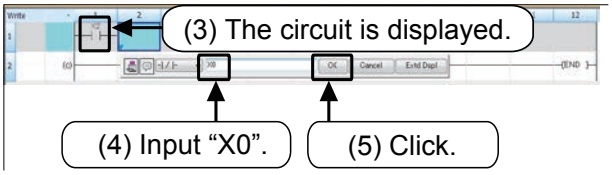


(1) Click the tool button F5.
Input "X2".

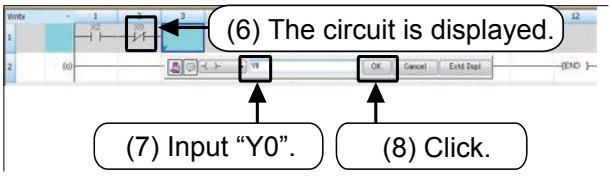


Cancel it by ESC or [Cancel].

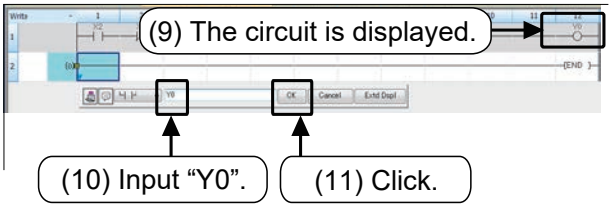
(2) Confirm by pressing the Enter key or [OK].



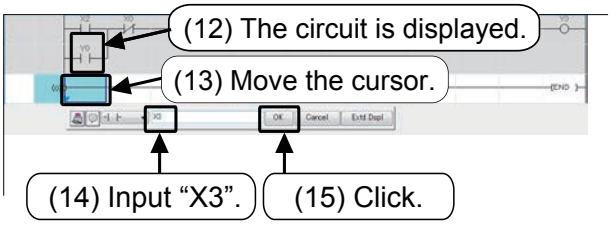
(3) The circuit input X2 is displayed.
(4) Press the tool button F6.
Input "X0".
(5) Confirm by pressing the Enter key or [OK].



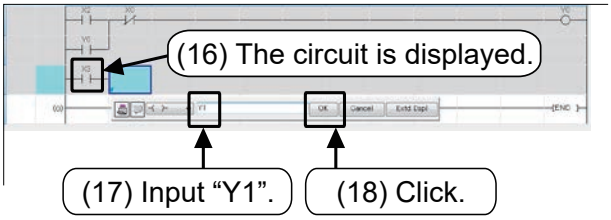
(6) The circuit input X0 is displayed.
(7) Click the tool button F7.
Input "Y0".
(8) Confirm by pressing the Enter key or [OK].



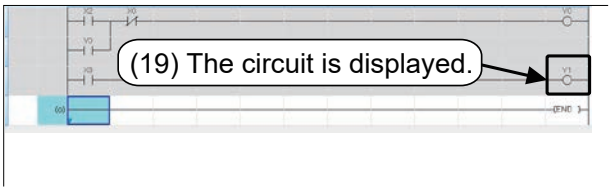
(9) The circuit input (Y0) is displayed.
(10) Click the tool button SF5.
Input "Y0".
(11) Confirm by pressing the Enter key or [OK].



(12) The circuit input (Y0) is displayed.
(13) Move the cursor to the beginning of the next line.
(14) Click the tool button F5.
Input "X3".
(15) Confirm by pressing the Enter key or [OK].

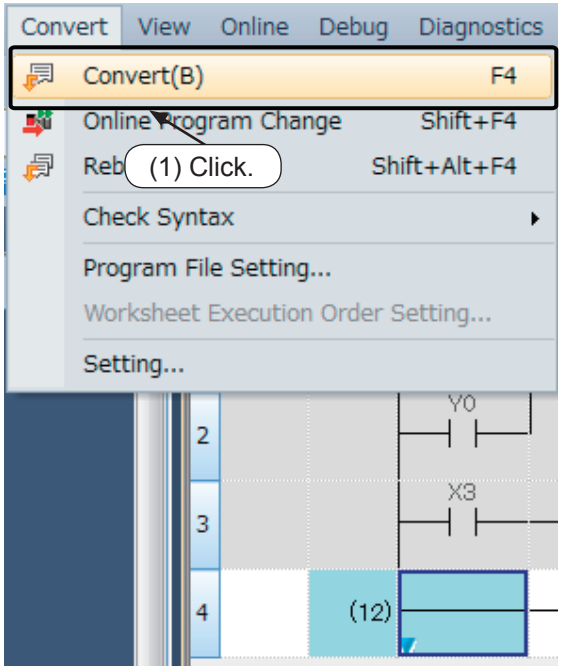


(16) The circuit input (X3) is displayed.
(17) Click the tool button F7.
Input "Y1".
(18) Confirm by pressing the Enter key or [OK].

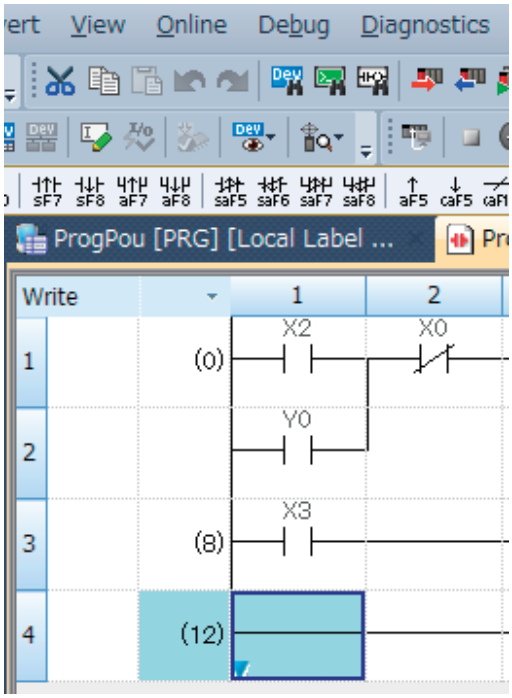


(19) The circuit input (Y1) is displayed.
The circuit is created!!

3.3.3 Converting a Created Ladder Program (Circuit Conversion)



(1) Click [Convert] → [Convert] from the menu bar.



(2) The ladder program is converted. When the conversion processing is completed and the input ladder blocks are finalized, the color of those ladder blocks changes from gray to white.

When an error has occurred during conversion, the results will appear on the output window.
When the error/warning message displayed on the output window is double-clicked, the cursor is moved to the position where the error has occurred. Check the ladder.

3.4 Preparations for writing to PLC

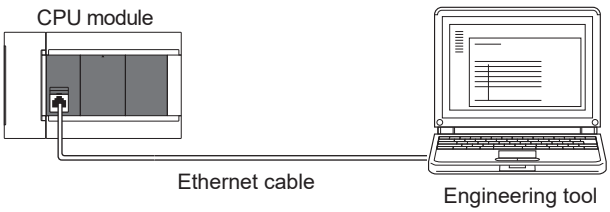
3.4.1 Specifying connection destination

(1) Ethernet direct connection

The FX5U CPU module and engineering tool (GX Works3) can be easily and directly connected without a hub by using one Ethernet cable.

The IP address and host name do not need to be set when using a direct connection.

A straight cable or cross cable can be used for the Ethernet cable.

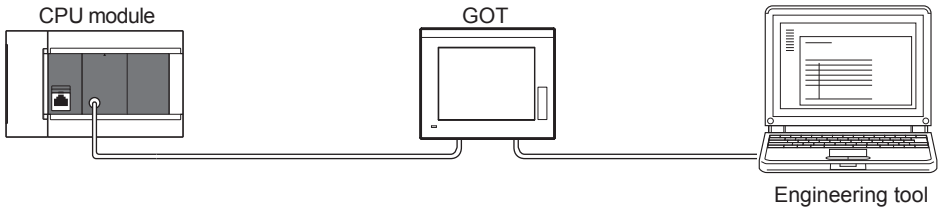


(2) GOT (direct connection) transparent connection

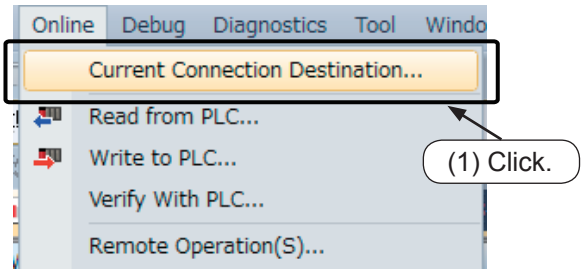
When the engineering tool (GX Works3) is connected to the GOT directly connected to FX5U CPU module, the FX5U CPU module program can be written, read and monitored via the GOT.

The demonstration model uses the GOT (direct connection) transparent connection.

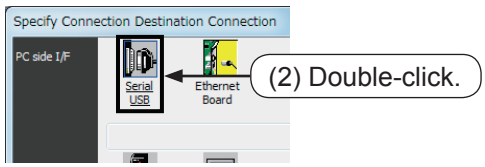
The following page explains the connection destination settings for this demonstration machine.



Complete the settings for communicating with the FX5U CPU module using the GOT (direct connection) transparent connection.

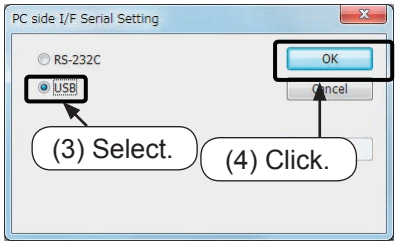


(1) Click [Online] → [Current Connection Destination] on the menu bar.



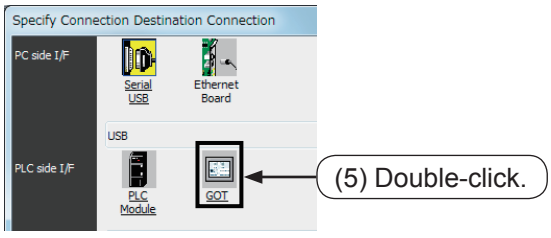
(2) Double-click  .

[Supplement]
When the message "Current settings will be lost when new item is selected. OK?" appears, click [Yes], and perform step (2) again.

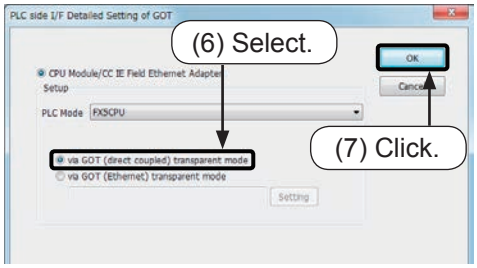


(3) Select the communication port on the PC side. (Select [USB] for the demonstration machine.)

(4) Click [Yes].



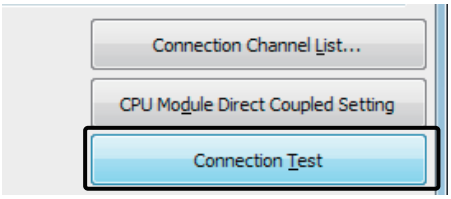
(5) Double-click  .



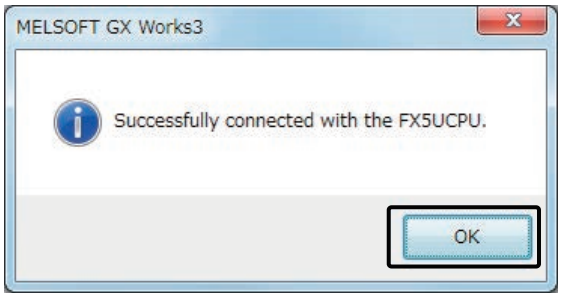
(6) Select [Use GOT (direct connection) transparent function].

(7) Click [OK].

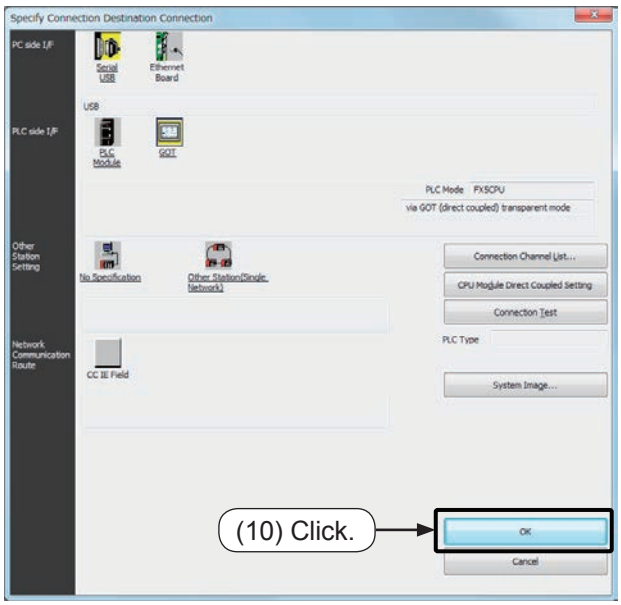
(8) Double-click [Connection Test].



(9) Check that the CPU module is successfully connected, and click [OK].

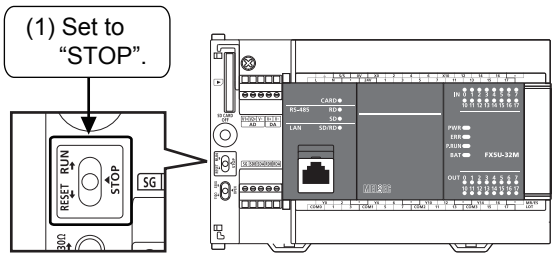


(10) Click [OK].

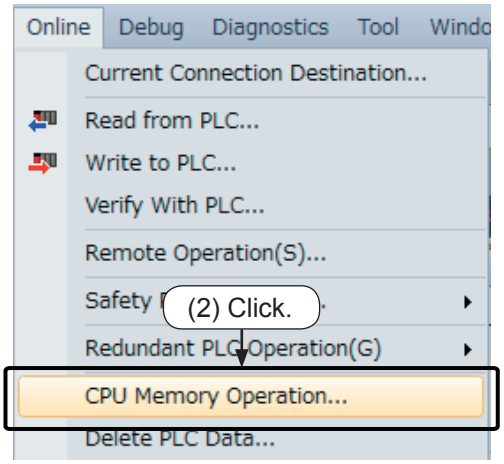


3.4.2 Initializing the PLC

This section explains the methods for initializing the FX5U CPU module memory.

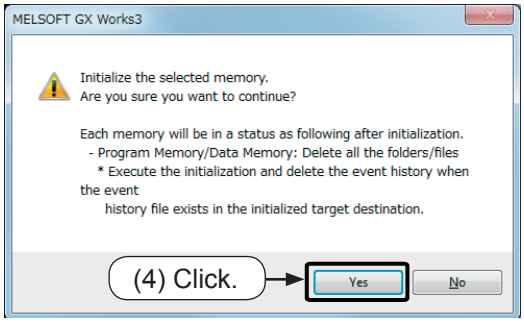


(1) Set the "RUN/STOP/RESET" switch of the CPU module to "STOP".



(2) Click [Online] → [CPU Memory Operation] from the menu bar.

(3) Click the [Initialization] button.

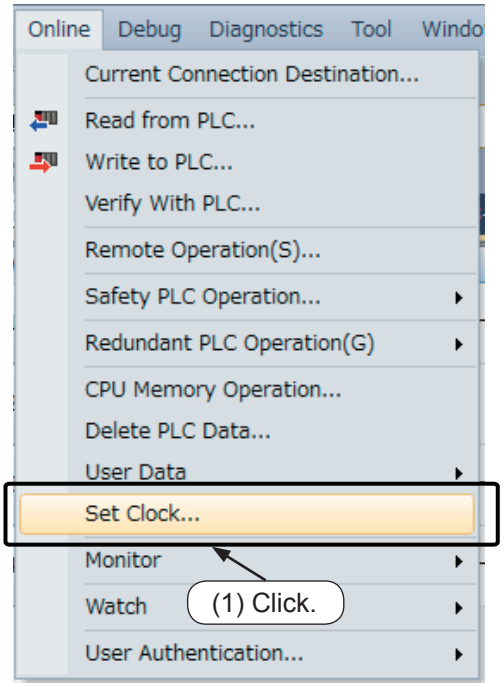


(4) The confirmation dialog box appears. Click the [Yes] button.

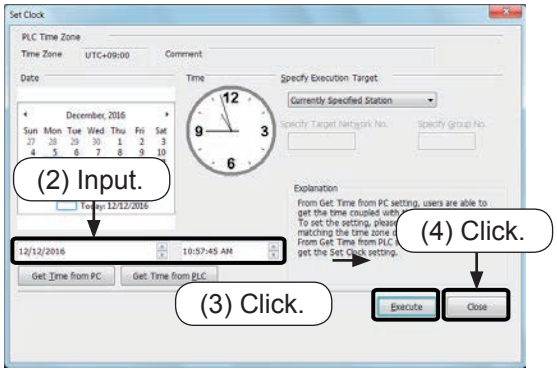
(5) When the initialization is completed, the dialog box shown on the left appears. Click the [OK] button.

3.4.3 Setting the clock of the PLC

The FX5U CPU module has a built-in clock function.
The method of setting the FX5U CPU's built-in clock using GX Works 3 is explained in this section.

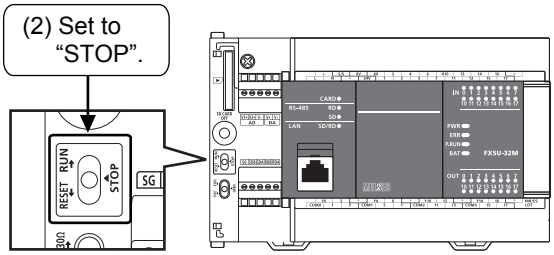


(1) Click [Online] → [Set Clock] from the menu bar.



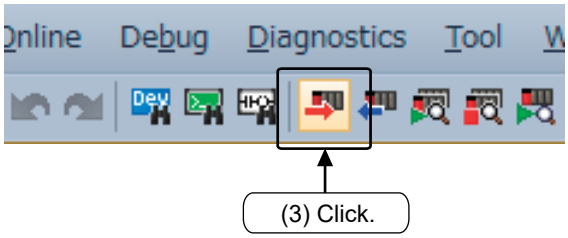
(2) Set a year, month, day, hour, minute and second on the "Set Clock" dialog box.
(3) Click the [Execute] button.
(4) Click the [Close] button.

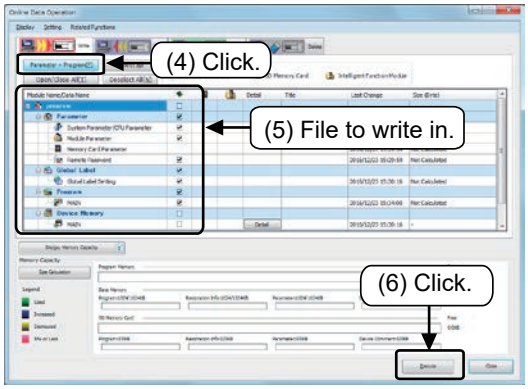
3.5 Writing programs to the PLC



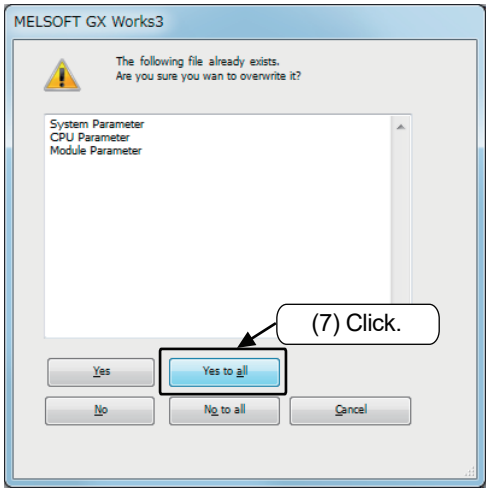
(1) Prior to this operation, create a ladder program (sequence program) with GX Works3.
(2) Set the RUN/STOP/RESET switch to the STOP position.

[Supplement: Automatic RUN/STOP function from GX Works3]
If writing is executed while the PLC is running, the message "Execute write to PLC after remote STOP?" will appear after step (6).
To execute write, click [Yes].
(3) Click on the toolbar, or click [Online] → [Write to PLC] from the menu bar.

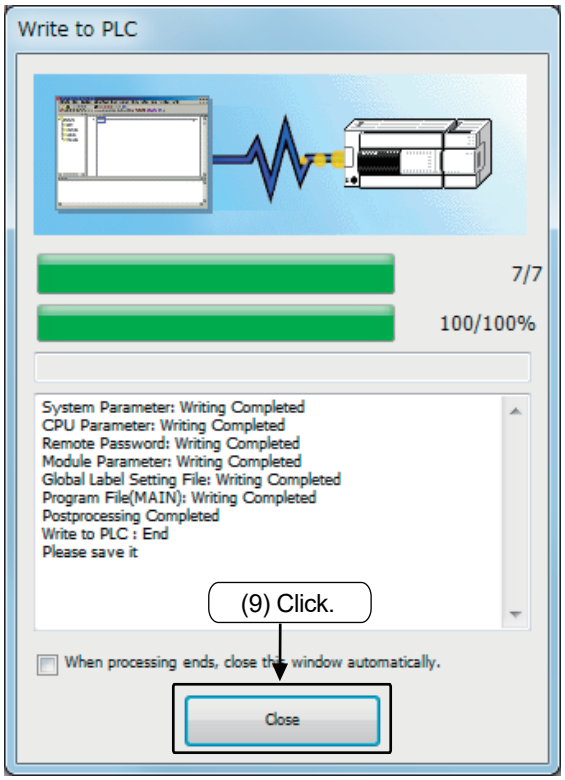




(4) Click [Paramater + Program].
(5) Confirm that the file to be written is selected.
(6) Click [Execute].
(Also refer to the supplement for step (2).)

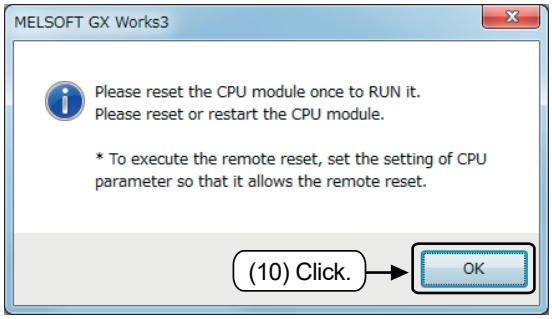


(7) If parameters or programs already exist in the CPU module, a message asking whether to overwrite will appear. Click [Yes to all].



(8) The dialog box indicating that writing is in progress appears.

(9) When writing the data is completed, the [Close] button is displayed. Click the [Close] button.
(This process is not required if "When processing ends, close window automatically." is checked.)



(10) Click [OK].

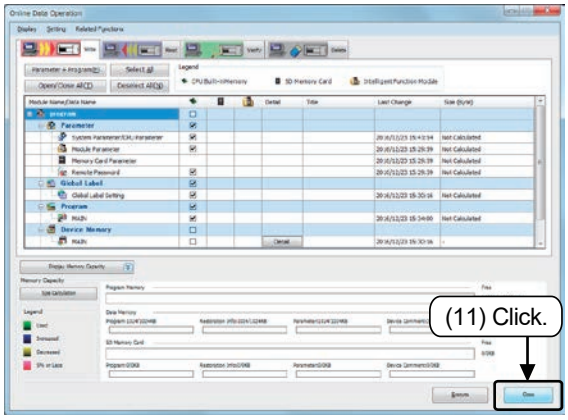
To run with the project written into the PLC, reset the PLC or turn the power OFF and ON.

[Supplement: How to reset]

(1) Hold the CPU module RUN/STOP/RESET switch to the RESET side for approx. one second.

(2) Confirm that the ERR LED flashes several times and then turns OFF.

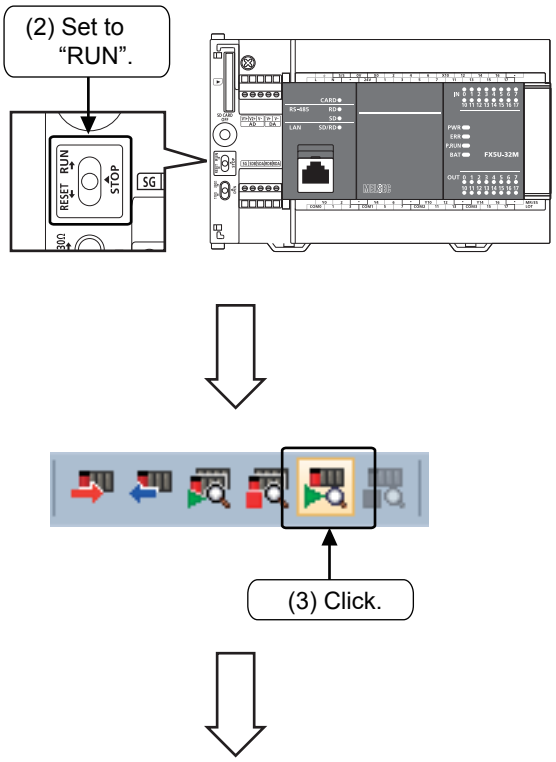
(3) Return the RUN/STOP/RESET to the STOP position.



(11) Click the [Close] button to close the dialog box.

3.6 Monitoring the PLC

3.6.1 Operation monitor of a program

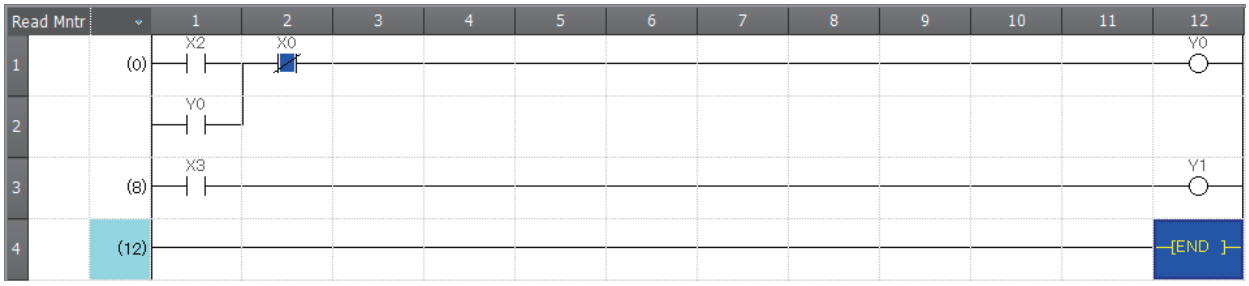


(1) It is assumed that the ladder program (sequence program) has been written to the CPU module, and the PLC has been reset or the power turned OFF and ON prior to this operation.
(Refer to the previous page for details on resetting.)

(2) Set the RUN/STOP/RESET switch of the CPU module to the "RUN" position.

(3) Click on the toolbar, or click [Online] → [Monitor] → [Monitor Mode] from the menu bar.

Operation check by circuit monitor



(1) Set [Switch X2 is "ON"] with the status [Switch X0 is "OFF"], and then check [Output Y0 is "ON"].
(2) Check [Output Y0 is "ON"] while [Switch X2 is "OFF"].
(3) Set [Switch X0 is "ON"] and then check [Output Y0 is "OFF"].
(4) Check [Output Y1 is "ON/OFF"] in accordance with [Switch X3 is "ON/OFF"].

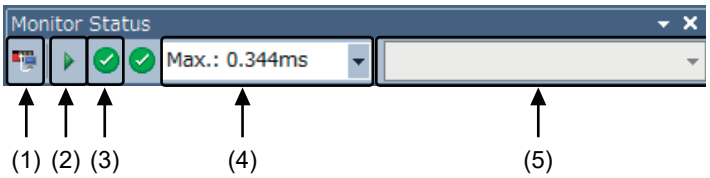
Point

Stopping the monitor and editing

- Select the (Monitor Stop) icon, or [Online] → [Monitor] → [Monitor Stop] to stop the monitor. The circuit can be edited in the write mode.

Monitoring on the monitor status bar

In the monitor mode, the following "Monitor Status" dialog box appears regardless of whether the operation status is monitoring in progress or not.



(1) Connection status

The connection status with the CPU module is displayed.

(2) CPU module operating status

The operating status of the CPU module is displayed. RUN: ►, STOP: ■, PAUSE: II. Remote operations can also be performed from GX Works3.

(3) ERROR status

The ERROR LED status of the CPU module is displayed.

Clicking the icon opens the "Module Diagnostics" window.

Refer to [Diagnosing the PLC] described later for the details.

(4) Scan time details

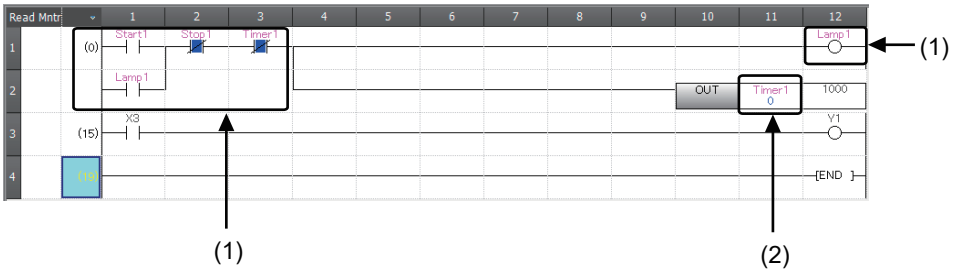
The scan time details are displayed. Select the value to be displayed from the drop-down list (current value, maximum value, or minimum value).

(5) Monitor target selection

Specify the monitor target FB instance when monitoring a FB program.

Monitoring on the ladder editor

The following figure shows how the ladder status is displayed on the ladder editor.

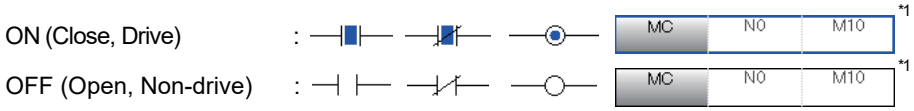


(1) The on/off states of contacts and coils are displayed.

(2) The current value of the word/double word type data is displayed.

● On/off state display

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

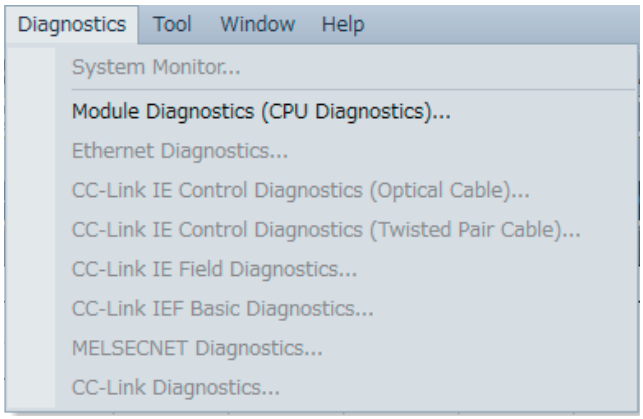


*1 Only comparison instructions that are equivalent to contacts and the instructions that are equivalent to coils are supported.

Comparison instructions equivalent to contacts : 16-bit binary data comparison, 32-bit binary data comparison, floating-point data comparison

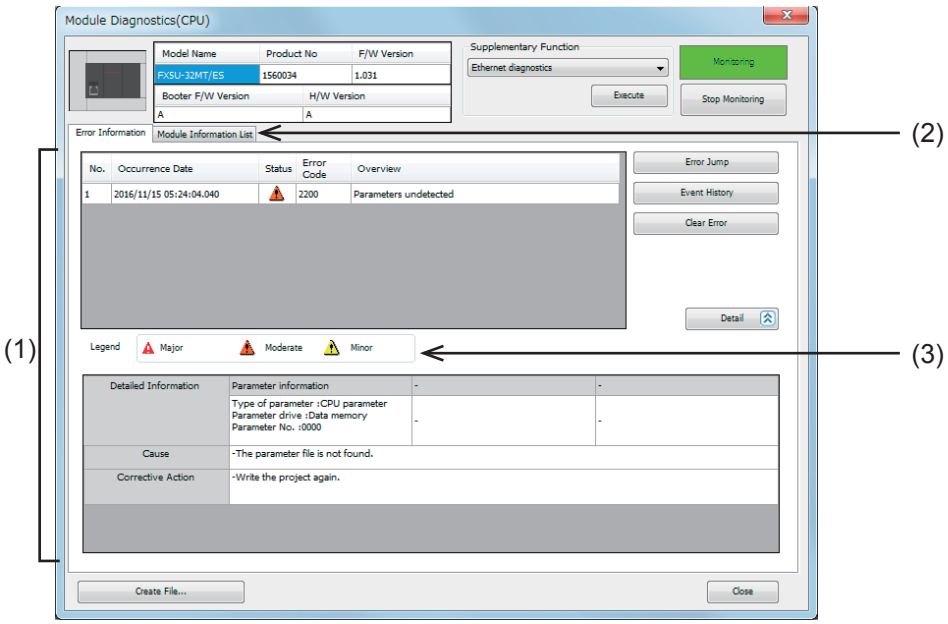
Instructions equivalent to coils : SET, RST, PLS, PLF, SFT, SFTP, MC, FF

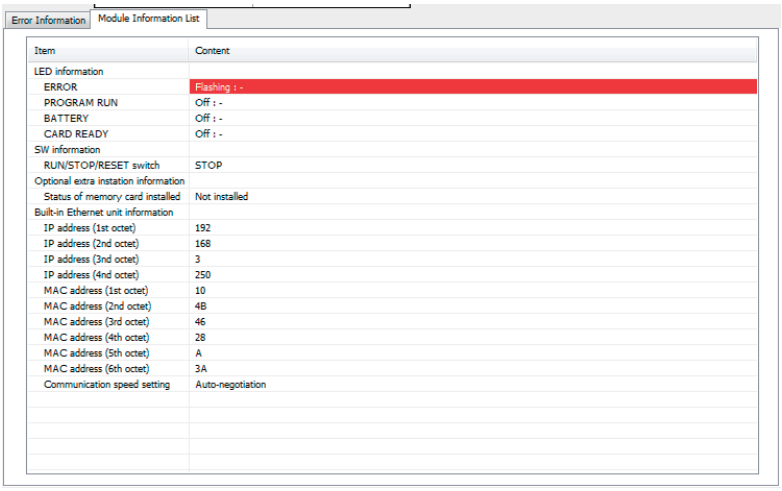
3.6.2 Diagnosing the PLC



(1) Click [Diagnostics] → [Module Diagnostics (CPU Diagnostics)] from the menu bar.

(2) The "Module Diagnostics(CPU)" window appears.



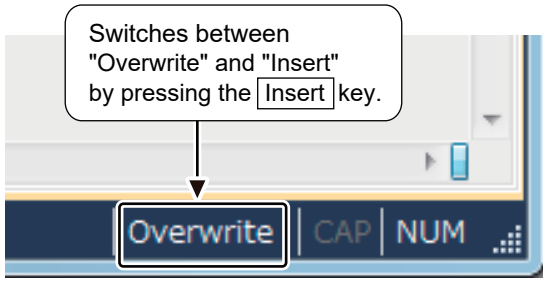
	Item	Description
(1)	Error Information	Select this tab to display the error information of the current CPU module.
(2)	Module Information List	Select this tab to display the status information of the CPU module. 
(3)	Legend	Displays the legend of icons displayed on the window.

3.7 Editing a circuit

3.7.1 Correcting a circuit

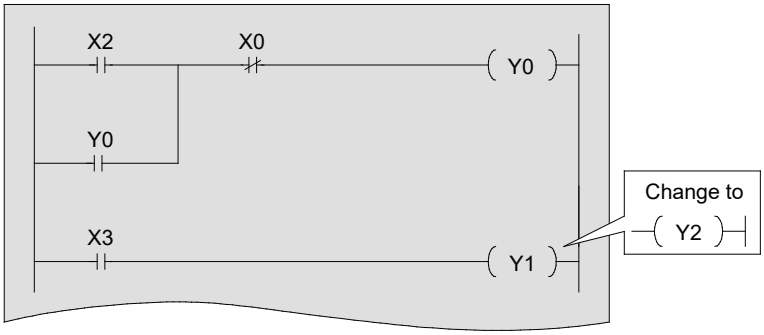
Point

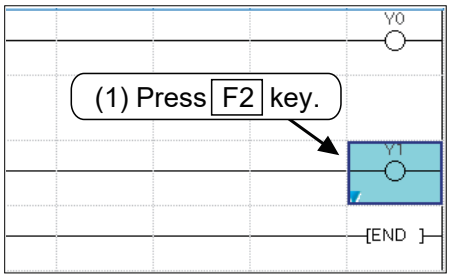
- Use only one-byte characters.
- Confirm that the write mode is enabled.
- Switch between "Overwrite" and "Insert"
  - Set to "Overwrite" when correcting and overwriting a circuit diagram.
  - A new circuit will be inserted when the "Insert" mode is on.



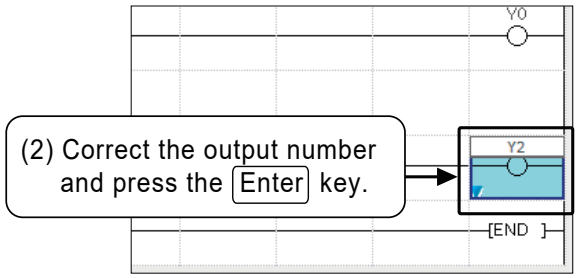
(1) Changing the OUT coils and numbers of contacts

[The circuit to be corrected]

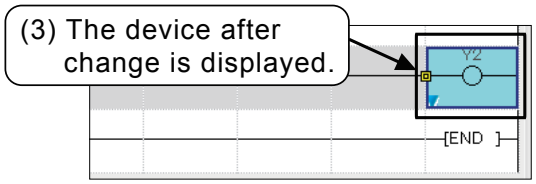




(1) Click the position to be modified, and press the F2 key.



(2) The device can be modified. Modify the device to "Y2", and press the Enter key.

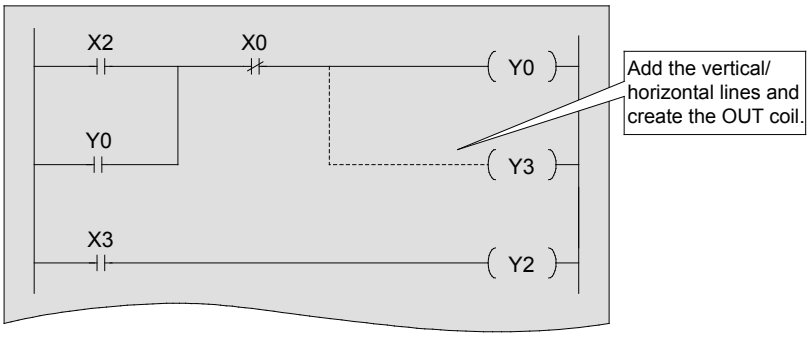


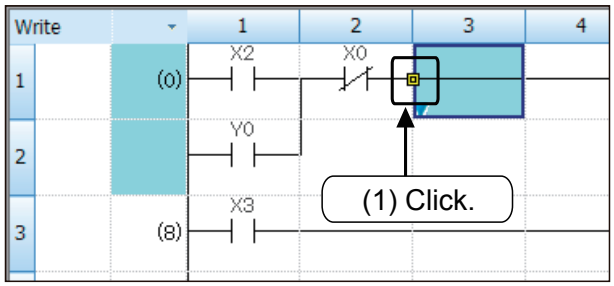
(3) The ladder program after the modification is displayed.
To change only the device number, click the F2 key.

(4) Confirm the changes by pressing the F4 (Convert) key.

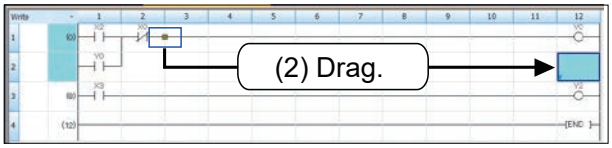
(2) Adding a circuit

[The circuit where lines are to be added]

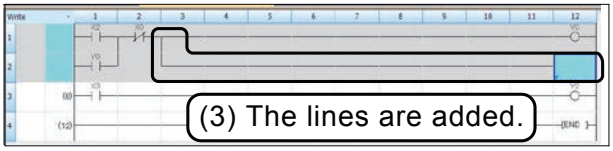




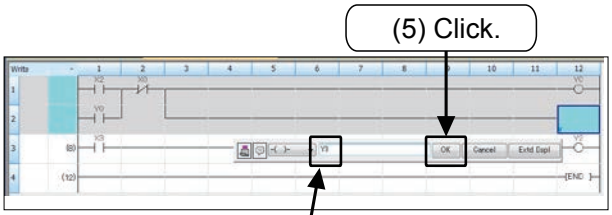
(1) Move the mouse pointer close to the exiting line, and click the displayed icon.



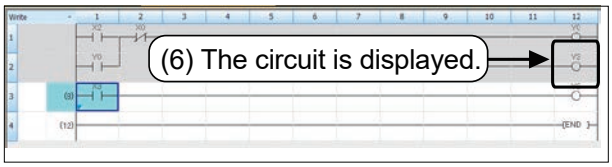
(2) Drag the icon from the position to the end position. A vertical line is drawn on the left side of the cursor.



(3) Release the left button of the mouse to create a line.



(4) Click on the toolbar and enter "Y3".
(5) Click the [OK] button.



(6) The added symbol (⊣(Y3)⊢) is displayed.
(7) Confirm the changes by pressing the F4 (Convert) key.

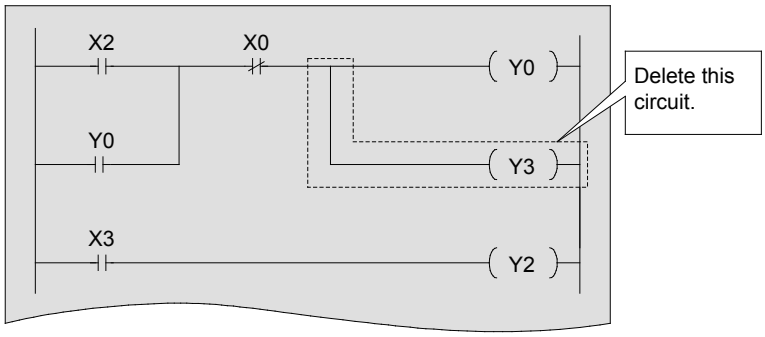
Reference

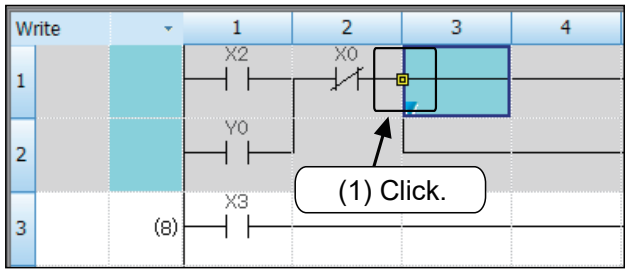
Adding or deleting a line with the key operation

- In GX Works3, lines can be added or deleted with the Ctrl key + → , ← , ↑ , or ↓ .

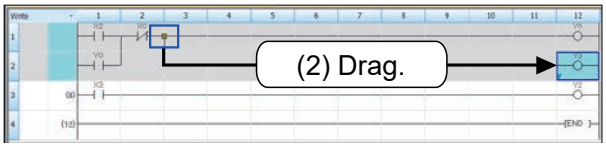
(3) Deleting lines

[The circuit where lines are to be deleted]

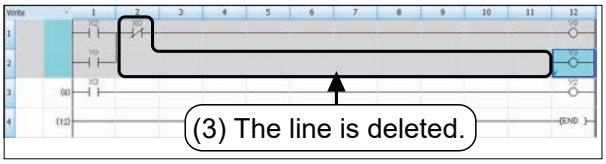




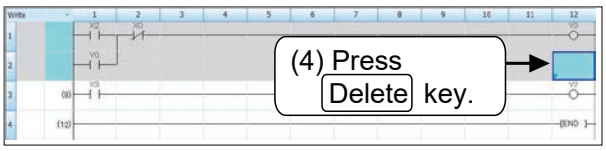
(1) Move the mouse pointer close to the exiting line, and click the displayed icon.



(2) Drag the icon along the line to be deleted.



(3) Release the left button of the mouse to delete the line.
The line connected to "END" cannot be deleted.

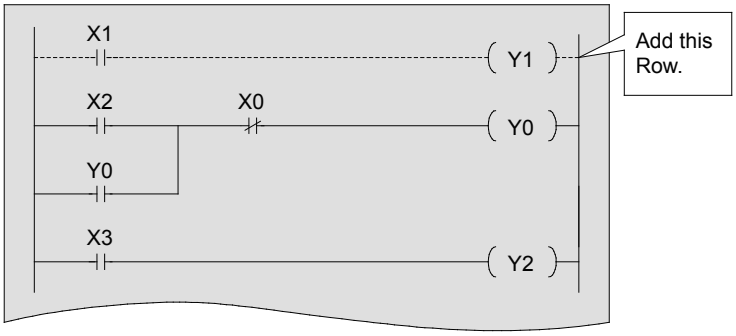


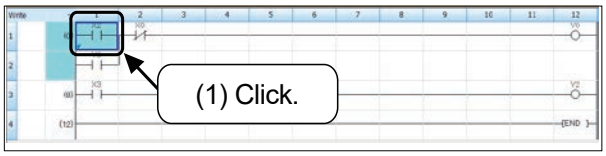
(4) Press the Delete key to delete (—(Y3)—).
(5) Confirm the changes by pressing the F4 (Convert) key.

3.7.2 Inserting and deleting Rows

(1) Inserting Rows

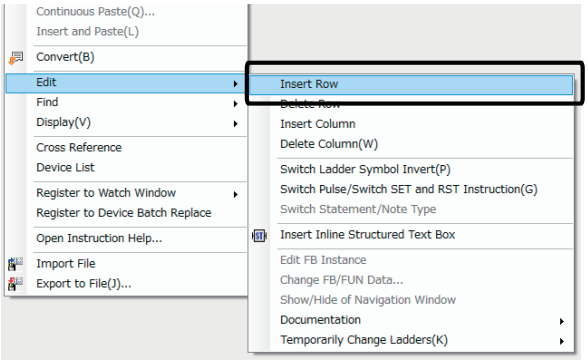
[The circuit where a Row is to be inserted]



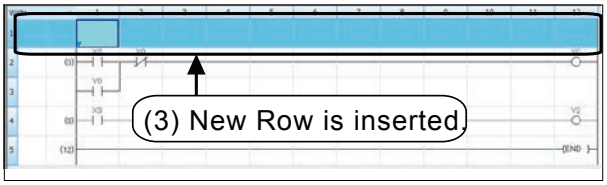


A line is inserted above the Row where the cursor is located

(1) Locate the cursor on the Row below the one to be inserted.

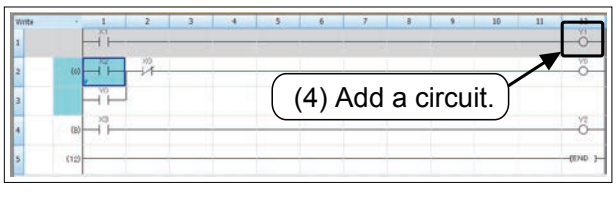


(2) Right click the mouse at any place, and select [Insert Row].



(3) A Row is inserted.

* Insert is also possible by shift + Insert

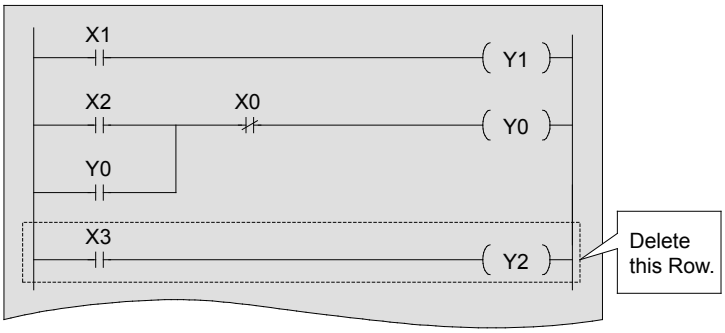


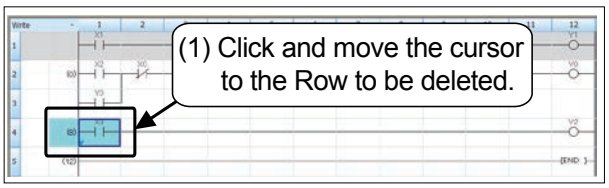
(4) Add a program in the inserted Row.

(5) Confirm the changes by pressing the F4 (Convert) key.

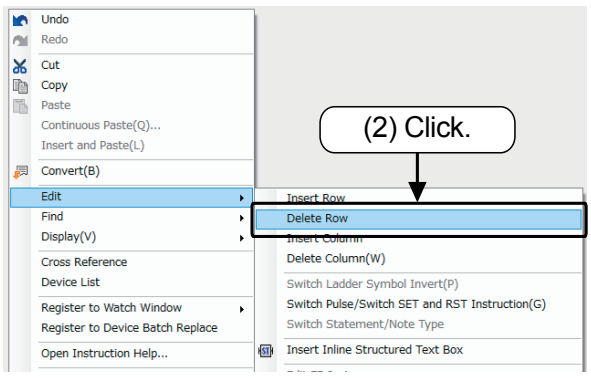
(2) Deleting Rows

[The circuit where a Row is to be deleted]

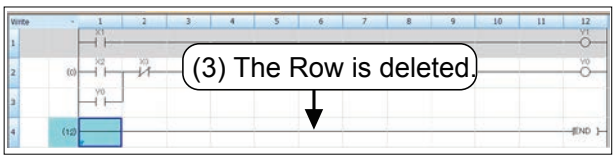




(1) Move the cursor to the Row to be deleted.



(2) Right click the mouse at any place, and select [Delete Row.]



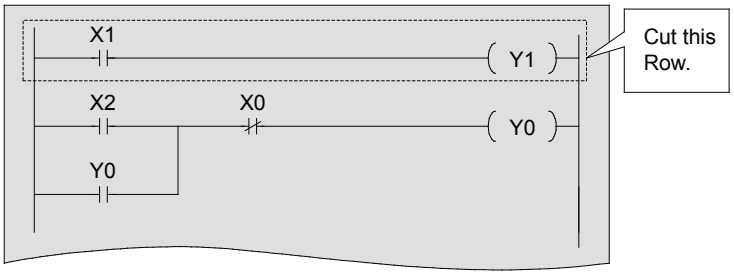
(3) The Row is deleted.

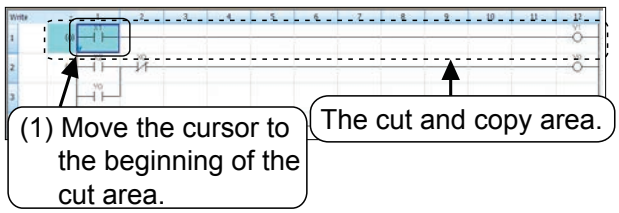
(4) Confirm the changes by pressing the F4 (Convert) key.

3.7.3 Cutting and copying (pasting) a circuit

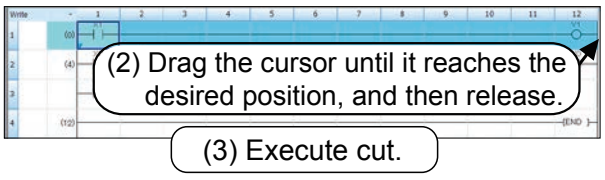
(1) Cut

[The circuit to be edited]



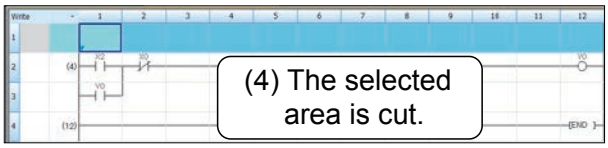


(1) Move the cursor to the beginning of the circuit to be cut.



(2) Drag the cursor until it reaches the desired position, and then release.

(3) Select from the tool bar or select [Edit] → [Cut] (Ctrl + X) from the menu bar, and execute the cut.

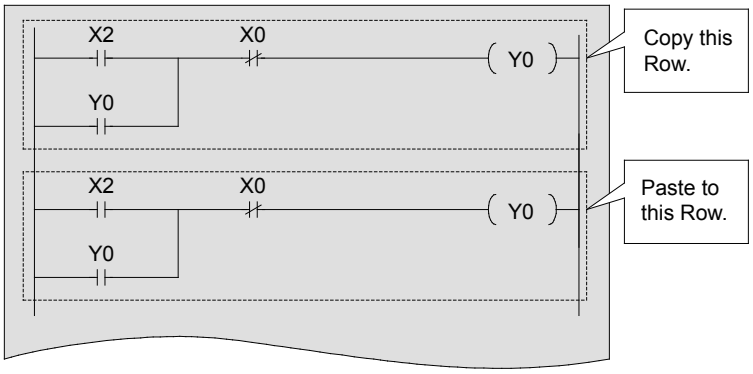


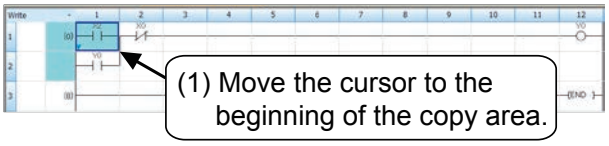
(4) The selected area is cut.

(5) Confirm the changes by pressing the F4 (Convert) key.

(2) Copy (paste)

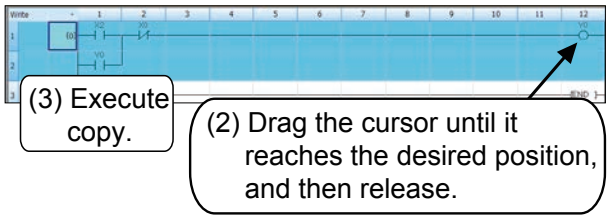
[The circuit to be copied (pasted)]





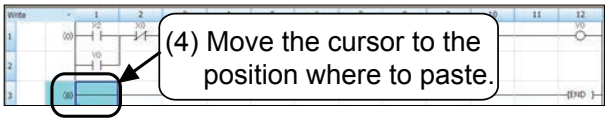
Continue to edit the circuit with the "cut" operation performed in the previous steps.

(1) Move the cursor to the beginning of the circuit to be copied.



(2) Drag the cursor until it reaches the desired position, and then release.

(3) Select from the tool bar or select [Edit] → [Copy] (Ctrl + C) from the menu bar.



(4) Move the cursor to the position where to paste.

Point

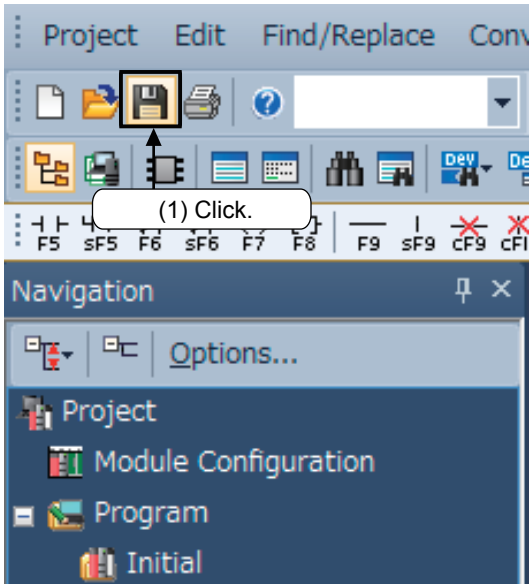
Using by the Insert key
"Overwrite" mode : Pastes by overwriting data from the cursor position.
"Insert" mode : Pastes it by inserting data at the cursor position.

(5) Select from the tool bar or select [Edit] → [Paste] (Ctrl + V) from the menu bar.

(6) Confirm the changes by pressing the F4 (Convert) key.

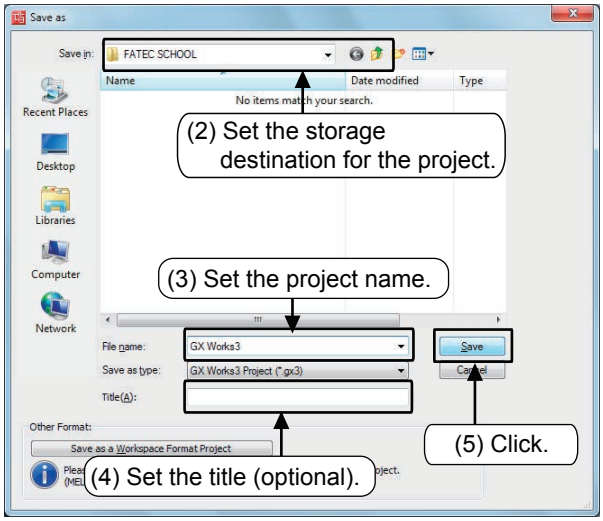
3.8 Saving a created circuit

Saving a created project in the single file format.



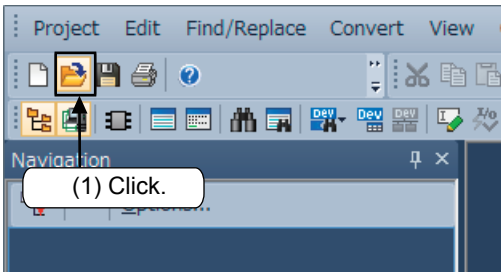
(1) Select from the tool bar or select [Project] → [Save] (Ctrl + S) from the menu bar.

If the project is overwritten, the saving operation is completed in this step.

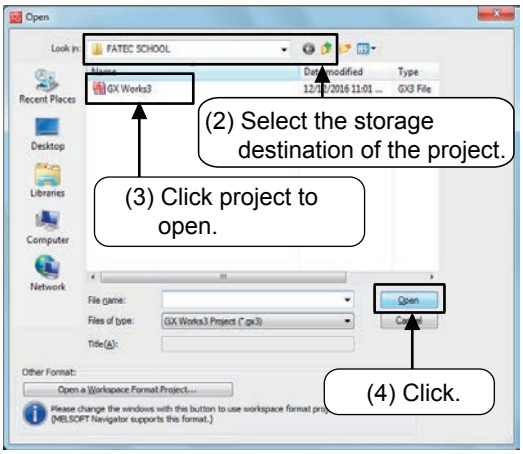


(2) Select the storage destination of the project.
(3) Set a project name.
(4) Set a title as necessary.
(5) After setting each item, click the [Save] button.
The saving operation is completed.

3.9 Opening a Saved Project



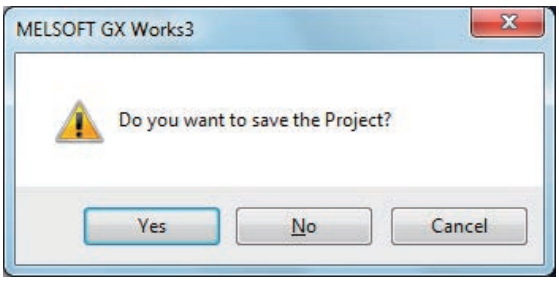
(1) Select from the tool bar or select [Project] → [Open project] (Ctrl + O) from the menu bar.



(2) Specify the storage destination for the project.
(3) Click the project to open.
(4) After setting each item, click the [Open] button.

Reference

If another project is open and has not been saved, the following dialog box appears.



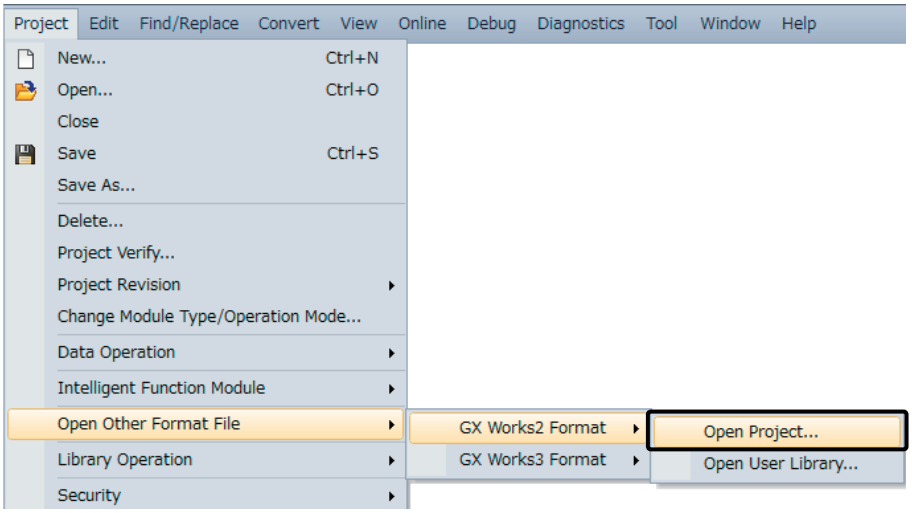
[Yes] ..... The project is saved and closed.
[No] ..... The project is not saved but closed.
[Cancel] ..... The project is not closed.
Reading of the project is paused.

Reference

Opening a Project in Another Format

The model of a project created with GX Works2 can be changed with GX Works3.
This function is supported with projects for the FX3U/FX3UC model.

To open a project in another format, open the target project by selecting [Project] → [Open Other Format File] → [GX Works2 Format] → [Open Project] from the menu bar.



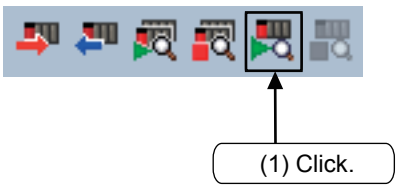
* Can only be executed in an evironment where GX Works2 is installed.

3.10 Debugging a program

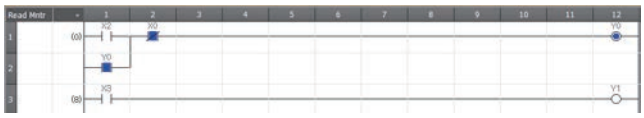
See the previous section "Writing programs to the PLC" for details on connecting with the FX5U CPU module and on writing a program.

3.10.1 Circuit monitor

Display the circuit, and monitor the conduction status of the contacts and the driving status of the coils. (Refer to the previous section "Operation monitor of a program" for details on the display.)



(1) Select from the tool bar or select [Online] → [Monitor] → [Monitor mode] from the menu bar.



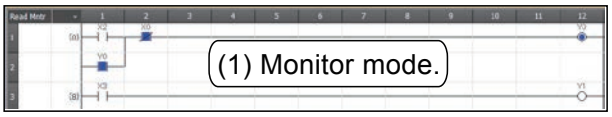
(2) The ON/OFF status of the circuit and the current value of the word device (timer, counter and data register) are displayed in the circuit monitor window.

Reference

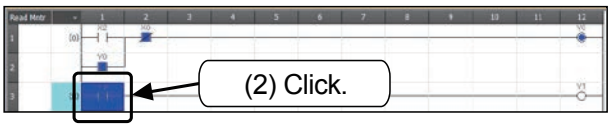
- With GX Works3, select the (Monitor Stop) icon, or select [Online] → [Monitor] → [Monitor stop] from the menu.

3.10.2 Forced ON/OFF of the device

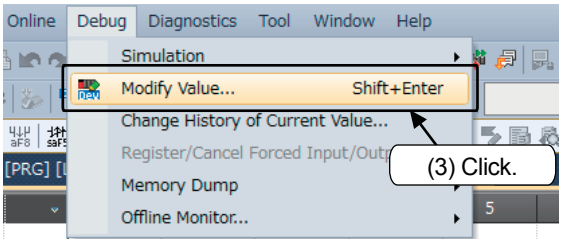
Using the device test screen, forcedly turn ON/OFF the bit devices of the PLC (M,X,Y,T,C and so on).
To confirm the operation of the output (Y), STOP the PLC.



(1) Set to the circuit monitor mode. (Refer to the previous section "Circuit monitor".)



(2) On the Circuit Monitor screen, click and select the bit device to be forcibly turned ON and OFF.



(3) The ON/OFF state will be highlighted when [Debug] → [Modify Value] are selected from the menu bar.

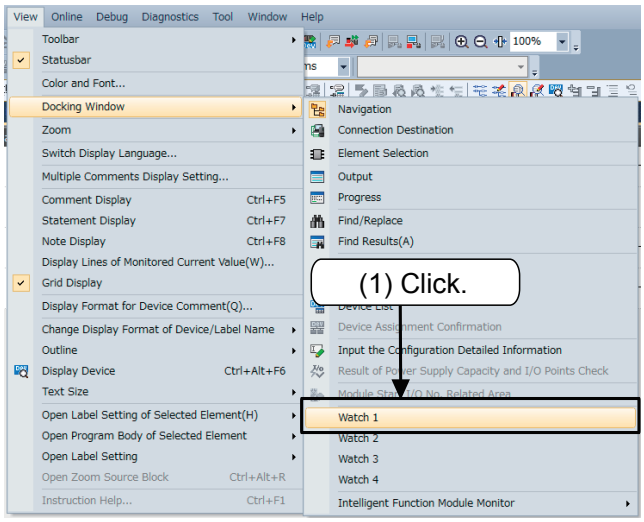
Reference

Forced ON/OFF of the device (Circuit monitor screen)

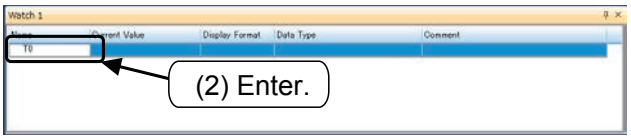
A specified device can be forcedly turned on/off by double-clicking any bit device (contact, coil) in the [Ladder monitor screen] while pressing the [Shift] key.

3.10.3 Watch Window

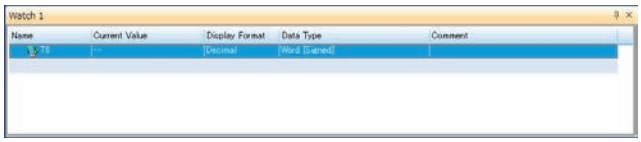
Designate a circuit diagram range on the Circuit Monitor screen, and register the device in this section in the Watch Window.



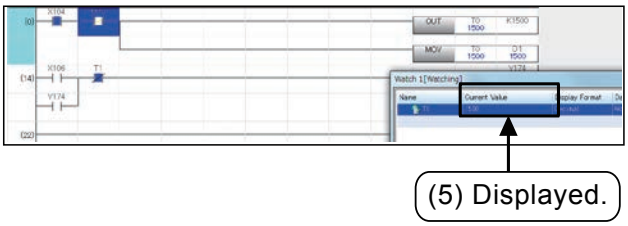
(1) Click [View] → [Docking Window] → one of [Watch 1] to [Watch 4] from the menu bar.
* In this example, select [Watch 1].



(2) The "Watch 1" window appears. Select a row to be edited, and click "Name" and input "T0".



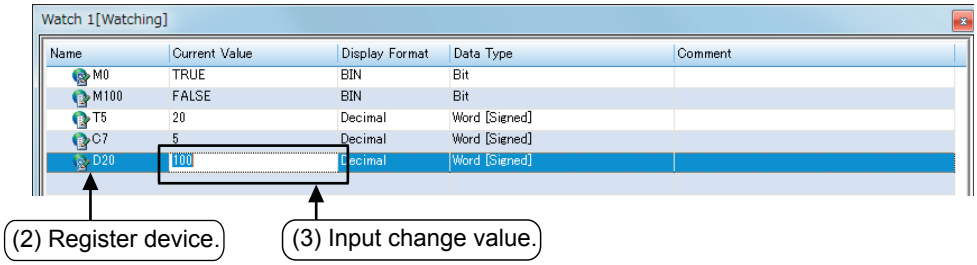
(3) The input device or label is registered.



(4) Click [Online] → [Watch] → [Start Watching] from the menu.
(5) The current value of the registered device or label is displayed in the window.

3.10.4 Forced ON/OFF and current value change using Watch Window

The bit device (M, Y, etc.) can be forcibly turned ON/OFF or the current value for the word device (T, C, D, etc.) can be changed from the Watch Window.
To confirm the operation of the output (Y), STOP the PLC.



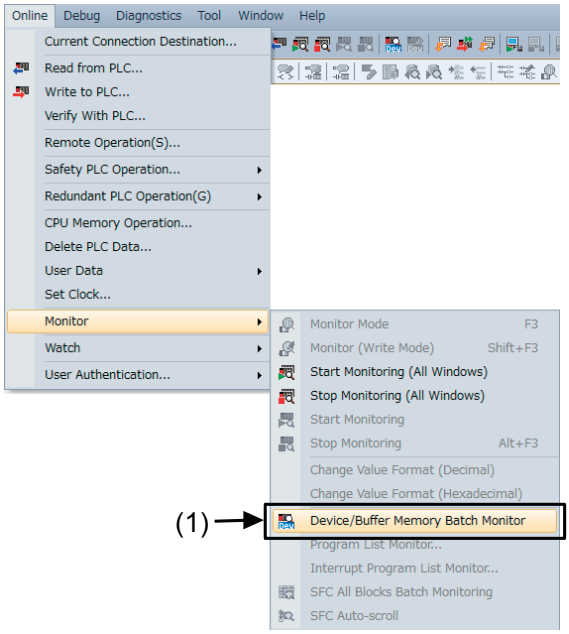
(1) Open the Watch Window. Refer to the previous section "Watch Window" for details on opening the window.

(2) Input the device number in the "Name" field, and register it into the Watch Window.
(Note) After registering, select [Online] → [Watch] → [Start Watching] from the menu.

(3) Input the changed value in the "Current Value" field, and press the Enter key.
- Bit device (X, Y, etc.) : Input "1" for ON, and "0" for OFF.
  (ON indicates "TRUE", and OFF indicates "FALSE".)
- Word device (T, C, D, etc.) : Input the range of values the device can handle.

3.10.5 Device batch monitor

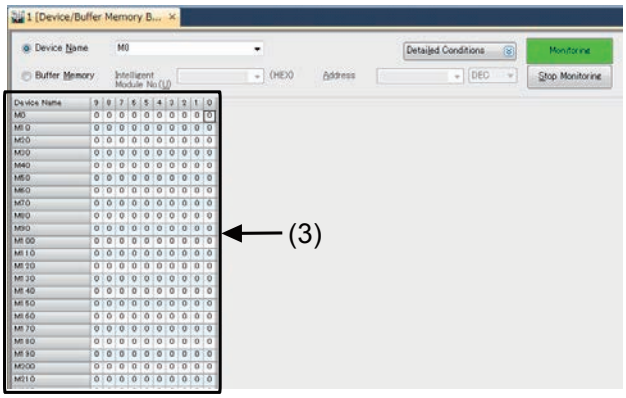
Specify a device and monitor a continuous range of devices that follow it.



(1) Select [Online] → [Monitor] from the menu bar. Or right click the circuit window and select [Device/Buffer Memory Batch Monitor].

(2) Input the first number of the devices to be monitored in the "Device/Buffer Memory Batch Monitor" window and then press the Enter key to begin monitoring.

(Note) When designating a timer or counter, click [Detailed Conditions], and designate the program part [ProgPou] from the program reference destination.



(3) The operation state corresponding to the device operation will display.

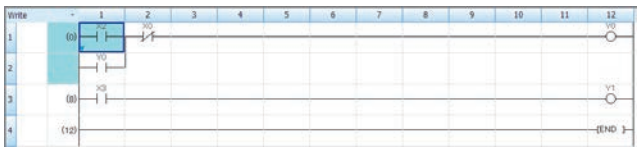
- Bit device (X, Y, M, S): ON = 1, OFF = 0
- Timer, counter: Contact/coil ON/OFF state, setting value, current value
- Data register: Current value

Changing the display format

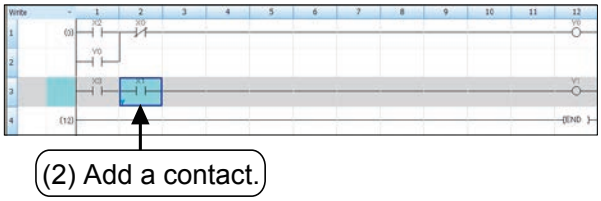
The format can be changed with [Display] → [Display Format Detailed Setting] on the menu bar.

3.10.6 Writing a program to the PLC during RUN

Write the corrected part of the circuit to the PLC when the PLC is running.
Less time is needed for writing during RUN since the entire program is not transferred.



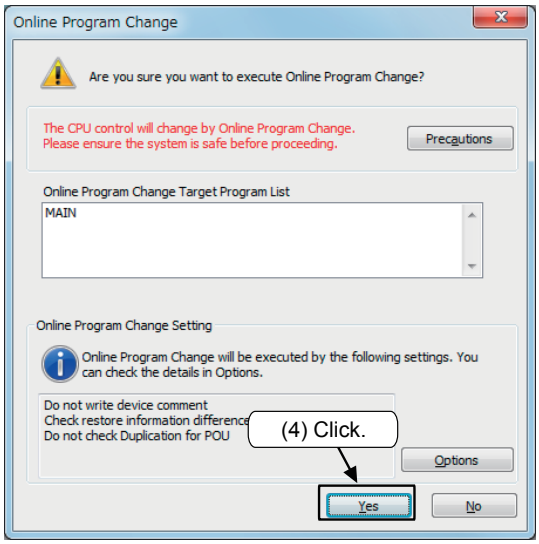
(1) A contact will be added to the circuit on the left as an example.



(2) Add a contact.
The circuit block is displayed in gray.

Write during RUN → [Shift]+[F4]

(3) Press [Shift] and [F4] together, or select [Convert] → [Online Program Change] from the menu bar.



(4) Confirm the details in the warning message, such as that there will be no safety risks if the control changes due to changes in the program, and then click [Yes].

Caution

It is impossible to write the program to the PLC if the program in the PLC is different from the one in GX Works3. Verify in advance, or transfer the program first by using [Write to PLC].

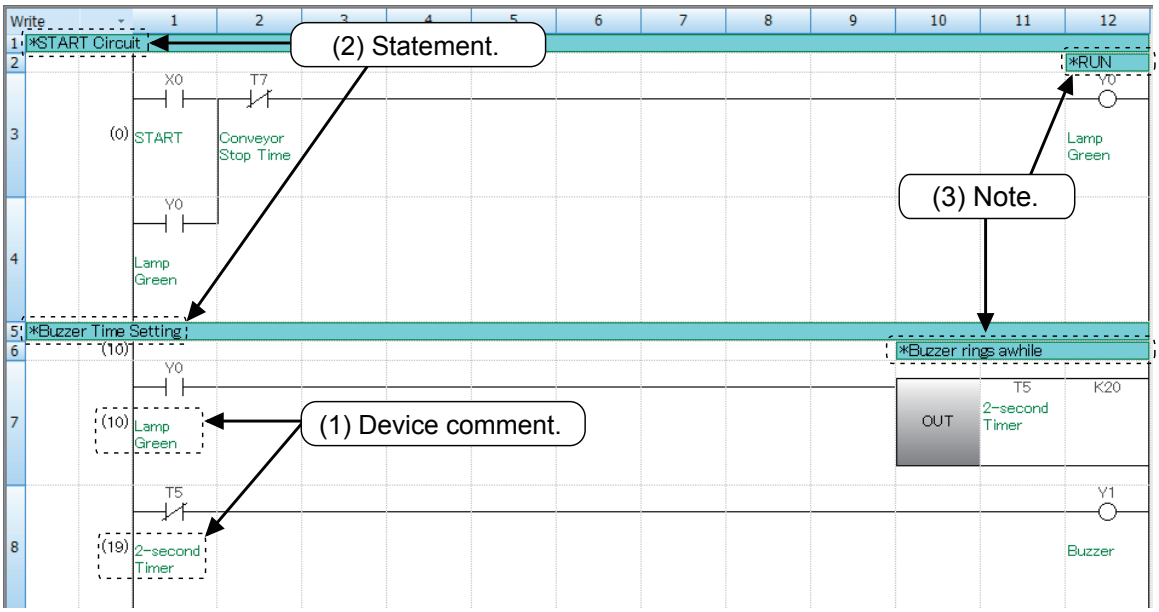
3.11 Inputting comments

3.11.1 Types of comments

The following 3 types of comments can be input.

Type	Purpose	The number of characters (full-width)
(1) Device comment	A comment describing the role and function of each device	1024
(2) Statement	A comment describing the role and function of circuit blocks	5000
(3) Note	A comment describing the role and function of output instructions	5000

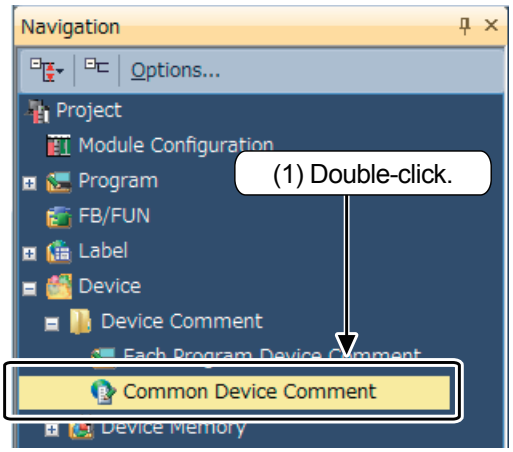
[Comment Examples]



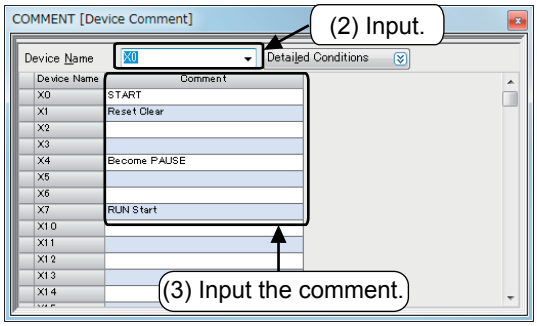
Point

- Device comments include [Common] and [By Program].
  - [Common Device Comment]: Comments used commonly by several programs.
  - [Comment By Program]: Comments used within the program file having the same name. These cannot be written to the PLC.
- Statements include "In PLC" and "In Peripheral".
  - [In PLC Statement/Note]: Stored in the PLC.
  - [In Peripheral Statement/Note]: Cannot be stored in the PLC.

3.11.2 Operation for creating device comments



(1) Double-click [Device] → [Device Comment] → [Common Device Comment] on the navigation window.



(2) Input the start number of the devices which are to be commented in "Device name", and click Enter.

(3) Input comments in the "Comment" column.

● When inputting comments for another device, input the device number again following step (2).

Displaying the comment

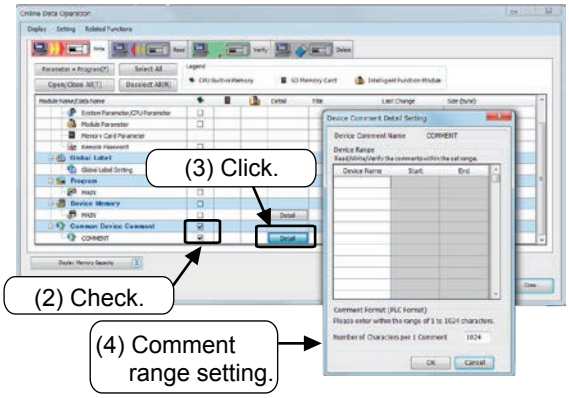
Click on [Display] → [Comment Display] in the menu, and input " ✓ ".

Point

Setting for writing the device comments to the PLC

- To write the device comments to the PLC, the writing range must be set.

Detailed setting for device comments

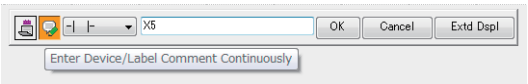


(1) Select [Online] → [Write to PLC] on the menu.
(2) Check "Common Device Comment".
(3) Click [Details].
(4) On the Device Comment Details Setting dialog, set the type and range of devices written to the PLC, and number of characters (1 to 1024) in one comment.

- All device comments will be write targets if the device comment range is not set.

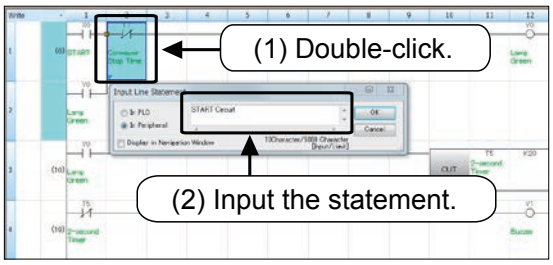
Reference

How to input comments when creating a circuit (Two methods)

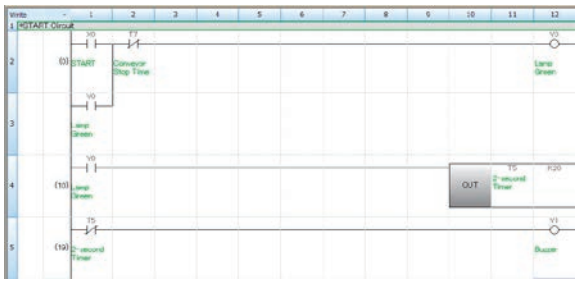


- Select [Tool] → [Options] from the menu, select [Program Editor] → [Ladder Editor] from the setting tree, and select [Yes] for [Enter label comment and device comment].
  When the above setting is made, the [Comment Input] window will open after the circuit input operation during circuit creation.
- When inputting the circuit, click the icon shown on the left to changeover.

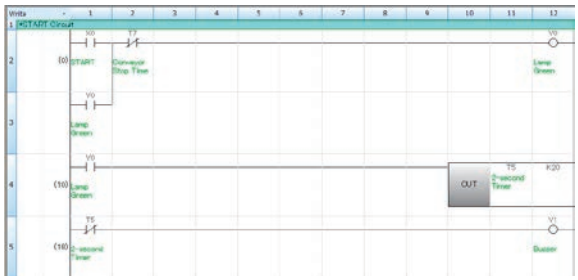
3.11.3 Operation for creating statements



(1) Click from the tool bar, and double-click anywhere on the circuit block where the statement is to be written.
(2) Input the statement in the "Input Line Statement" window and click [OK].



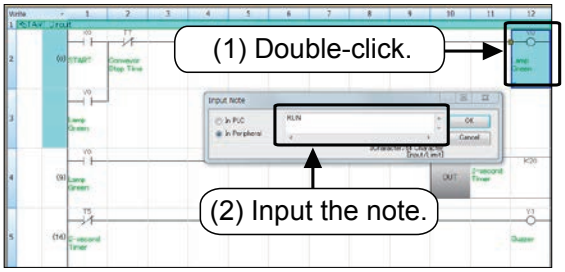
(3) F4 (Convert)



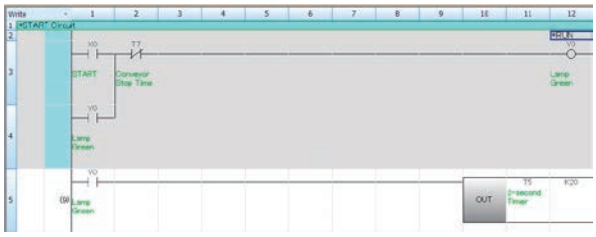
(3) Set with the F4 (Convert) key.

● Click on the tool bar again to finish the operation.

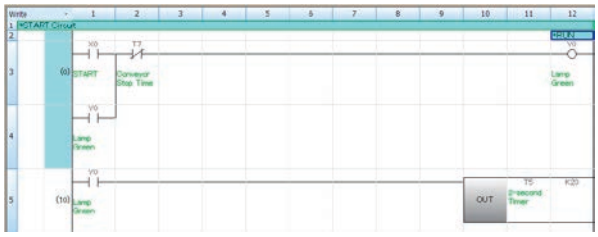
3.11.4 Operation for creating notes



(1) Click from the tool bar, and double-click the output instruction symbol where the note is to be written.
(2) Input the note in the "Input Note" window and click [OK].



(3) F4 (Convert)



(3) Set with the F4 (Convert) key.

- Click on the tool bar again to finish the operation.

MEMO

Chapter 4
SUMMARY OF PLC INSTRUCTIONS

So far…

It was explained that the PLC is a collection of many relays, timers and counters, and the internal sequence is created with personal computer software, etc.
It was also explained that when creating this sequence program, there are rules according to how the contacts and coils are connected and according to the types of coils. These rules are the instructions.

In the instructions…

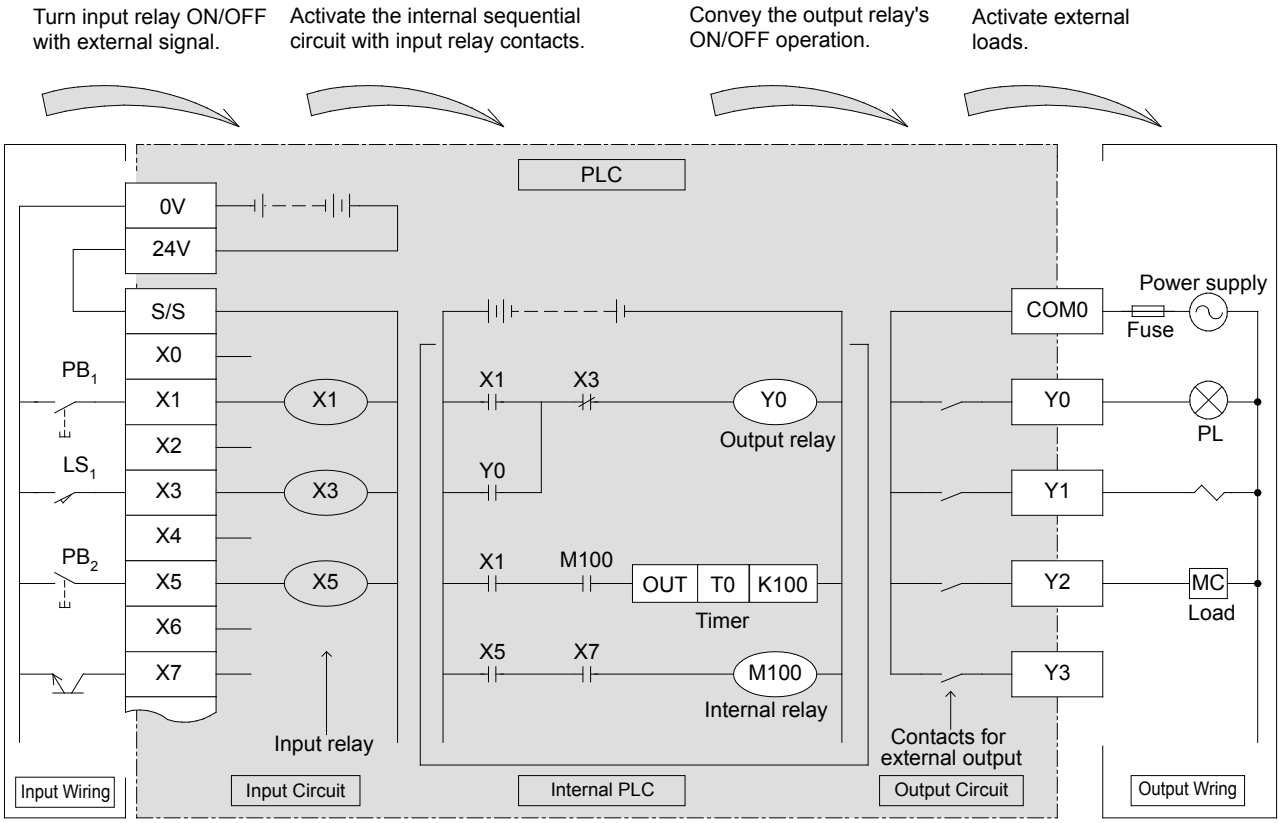
There are some that function with just the instruction word + device, or with just the instruction word. Thus, it is necessary to know how the device number is configured.

In this chapter…

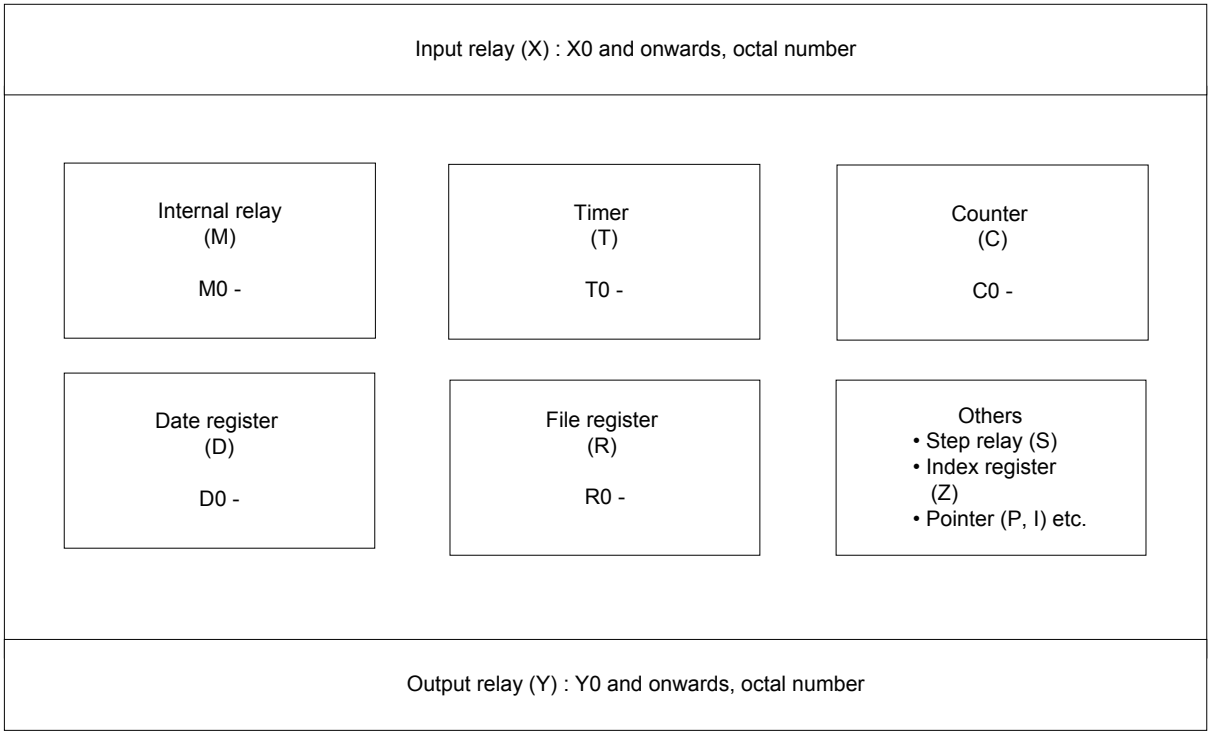
Instructions for the PLC are described. Note that there are also many application instructions which are used to simplify complicated sequential circuit designs.
If you would like to perform the programming training, please learn the basic personal computer operations referenced in Chapter 3 beforehand.
Now, let’s understand the contents of instructions.

4.1 Devices and device numbers

There are various relay and timer identification numbers.



(1) Types of PLC devices



(2) Number of device points for FX5U CPU module main devices

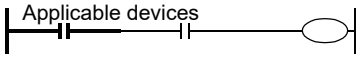
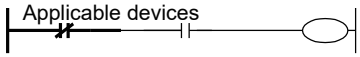
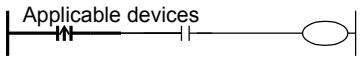
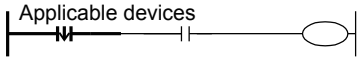
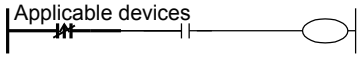
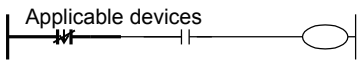
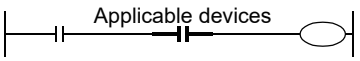
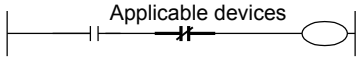
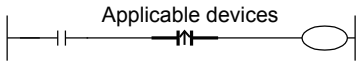
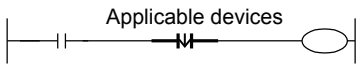
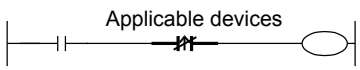
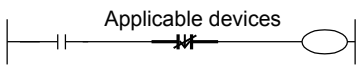
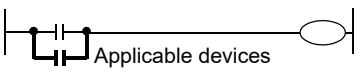
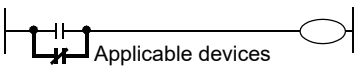
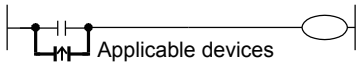
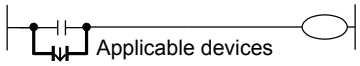
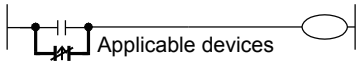
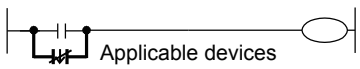
Item			Base	Maximum number of points	
Number of user device points	Input relay (X)		8	1024 points or less	The total of X and Y assigned to the input/output is maximum 256 points.
	Output relay (Y)		8	1024 points or less	
	Internal relay (M)		10	32768 points (variable with parameters) *1	
	Latch relay (L)		10	32768 points (variable with parameters) *1	
	Link relay (B)		16	32768 points (variable with parameters) *1	
	Annunciator (F)		10	32768 points (variable with parameters) *1	
	Link special relay (SB)		16	32768 points (variable with parameters) *1	
	Step relay (S)		10	4096 points (fixed)	
	Timer series	Timer (T)	10	1024 points (variable with parameters) *1	
	Retentive timer series	Retentive timer (ST)	10	1024 points (variable with parameters) *1	
	Counter series	Counter (C)	10	1024 points (variable with parameters) *1	
		Long counter (LC)	10	1024 points (variable with parameters) *1	
	Data register (D)		10	8000 points (variable with parameters) *1	
	Link register (W)		16	32768 points (variable with parameters) *1	
	Link special register (SW)		16	32768 points (variable with parameters) *1	
Number of system device points	Special relay (SM)		10	10000 points (fixed)	
	Special register (SD)		10	12000 points (fixed)	
Module access device	Intelligent function module device		10	65536 points (Specified with U□¥G□)	
Number of index register points	Index register (Z) *2		10	24 points	
	Long index register (LZ) *2		10	12 points	
Number of file register points	File register (R)		10	32768 points (variable with parameters) *1	
Number of nesting points	Nesting (N)		10	15 points (fixed)	
Number of pointer points	Pointer (P)		10	4096 points	
	Interrupt pointer (I)		10	178 points (fixed)	
Others	Decimal constant (K)	Signed	—	For 16 bits: -32768 to +32767, For 32 bits: -2147483648 to +2147483647	
		Unsigned	—	For 16 bits: 0 to 65535, For 32 bits: 0 to 4294967295	
	Hexadecimal constant (H)		—	For 16 bits: 0 to FFFF, For 32 bits: 0 to FFFFFFFF	
	Real constant (E)	Single precision	—	E-3.40282347+38 to E-1.17549435-38, 0, E1.17549435-38 to E3.40282347+38	
	String		—	Shift-JIS Code maximum half-byte 255 characters (256 characters including NULL)	

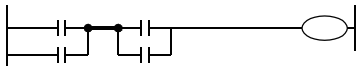
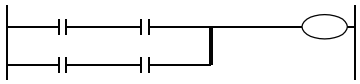
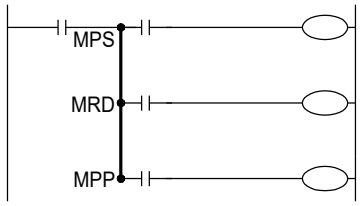
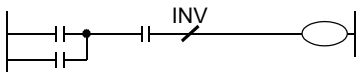
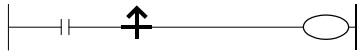
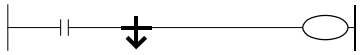
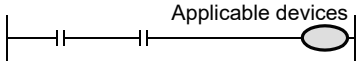
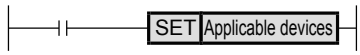
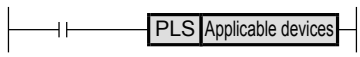
*1: Variable with parameters within the range of the CPU module built-in memory's capacity.

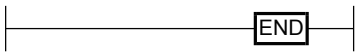
*2: The index register (Z) and long index register (LZ) can be set with a total of 24 words or less.

4.2 Types of PLC instructions

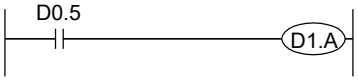
The following table lists the available PLC instructions for FX5U CPU module programming.

Mnemonic	Name	Symbol	Function	Applicable Devices*1
Contact Instruction				
LD	Load		Initial logical operation contact type NO (normally open)	X,Y,M,L,SM,F, B,SB,S,T,ST,C, LC,DX D,W,SD,SW,R, U□\G□
LDI	Load inverse		Initial logical operation contact type NC (normally closed)	
LDP	Load pulse		Initial logical operation of Rising edge pulse	
LDF	Load falling pulse		Initial logical operation of Falling/trailing edge pulse	
LDPI	Load pulse inverse		Start of Rising edge pulse NOT operation	
LDFI	Load falling pulse inverse		Start of Falling edge pulse NOT operation	
AND	AND		Serial connection of NO (normally open) contacts	
ANI	AND inverse		Serial connection of NC (normally closed) contacts	
ANDP	AND pulse		Serial connection of Rising edge pulse	
ANDF	AND falling pulse		Serial connection of Falling/ trailing edge pulse	
ANDPI	AND pulse inverse		Serial connection of Rising edge pulse negate	
ANDFI	AND falling pulse inverse		Serial connection of Falling edge pulse negate	
OR	OR		Parallel connection of NO (normally open) contacts	
ORI	OR inverse		Parallel connection of NC (normally closed) contacts	
ORP	OR pulse		Parallel connection of Rising edge pulse	
ORF	OR falling pulse		Parallel connection of Falling/trailing edge pulse	
ORPI	OR pulse inverse		Parallel connection of Rising edge pulse negate	
ORFI	OR falling pulse inverse		Parallel connection of Falling edge pulse negate	

Mnemonic	Name	Symbol	Function	Applicable Devices*1
Connection Instruction				
ANB	AND Block		Serial connection of multiple parallel circuits	–
ORB	OR Block		Parallel connection of multiple contact circuits	–
MPS	Memory Point Store		Stores the current result of the internal PLC operations	–
MRD	Memory Read		Reads the current result of the internal PLC operations	
MPP	Memory POP		Pops (recalls and removes) the currently stored result	
INV	Inverse		Invert the current result of the internal PLC operations	–
MEP	MEP		Conversion of operation result to leading edge pulse	–
MEF	MEF		Conversion of operation result to trailing edge pulse	–
Out Instruction				
OUT	OUT		Final logical operation type coil drive	X,Y,M,L,SM,F,B,SB,S,DY D,W,SD,SW,R,U□\G□
SET	SET		SET Bit device latch ON	X,Y,M,L,SM,F,B, SB,S,T,ST,C,LC,DY D,W,SD,SW,R,U□\G□
RST	Reset		RESET Bit device OFF	X,Y,M,L,SM,F,B, SB,S,DY ST,C,D,W,SD,SW, R,U□\G□,Z,LC,LZ
PLS	Pulse		Rising edge pulse	X,Y,M,L,SM,F,B,SB,S,DY D,W,SD,SW,R,U□\G□
PLF	Pulse Falling		Falling/trailing edge pulse	
ALT	Alternate		Output reversal	X,Y,M,L,SM,F,B,SB,S,DY D,W,SD,SW,R,U□\G□
Master Control Instruction				
MC	Master Control		Denotes the start of a master control block	X,Y,M,L,SM,F,B,SB,S, D,W,SD,SW,R
MCR	Master Control Reset		Denotes the end of a master control block	–

Mnemonic	Name	Symbol	Function	Applicable Devices*1
End Instruction				
END	END		End program, and return to input/output process and 0 step.	—

*1: When the device No. is specified, the word device (D,W,SD,SW,R,U□\G□) can handle the bit data for the specified bit No.
The bit No. can be specified with a hexadecimal in the range of 0 to F.
For example, the D0 bit 5 (b5) is specified as "D0.5", and the D1 bit 10 (b10) is specified as "D1.A".

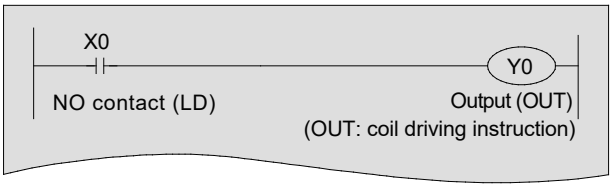


4.3 Let's master PLC instructions

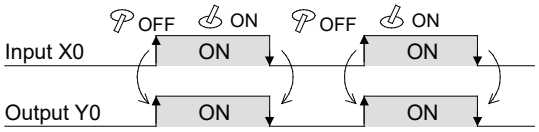
4.3.1 Contact instruction and out instruction

(1) [Program of NO contact] Normally open instructions

Ladder display



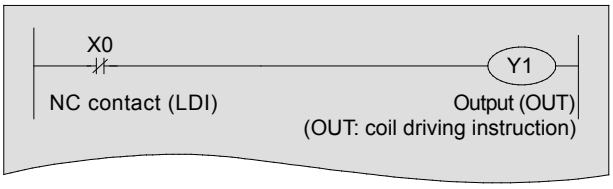
<<Operation>>



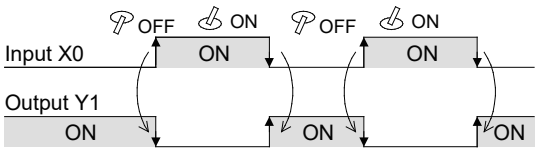
- If the input condition X0 is "ON", Y0 is "ON".
- If X0 is "OFF", Y0 is "OFF".

(2) [Program of NC contact] Normally closed instructions

Ladder display



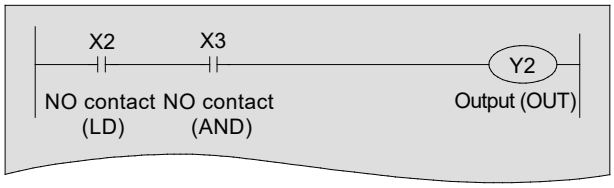
<<Action>>



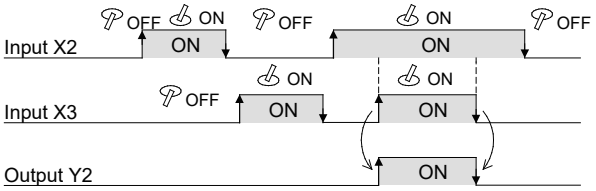
- If the input condition X0 is "OFF", Y1 is "ON".
- If X0 is "ON", Y1 is "OFF".

(3) [Program of a serial circuit (1)]

Ladder display



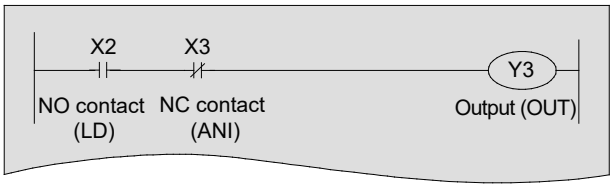
<<Action>>



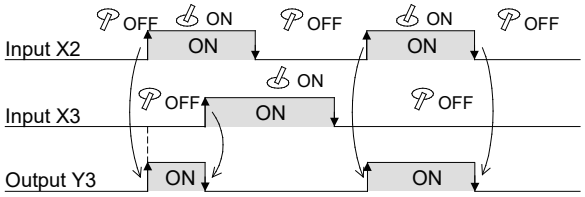
- If the input condition X2 and X3 are both "ON", Y2 is "ON".
- If X2 or X3 is "OFF", Y2 is "OFF".

[Program of a serial circuit (2)]

Ladder display



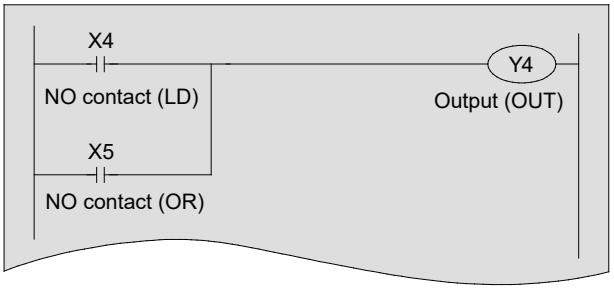
<<Action>>



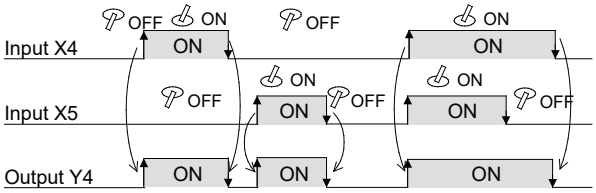
- If the input condition X2 is "ON" and X3 is "OFF", Y3 is "ON".
- If X2 is "OFF" or X3 is "ON", Y3 is "OFF".

(4) [Program of a parallel circuit (1)]

Ladder display



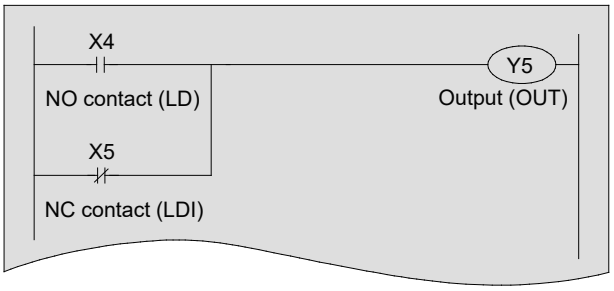
<<Action>>



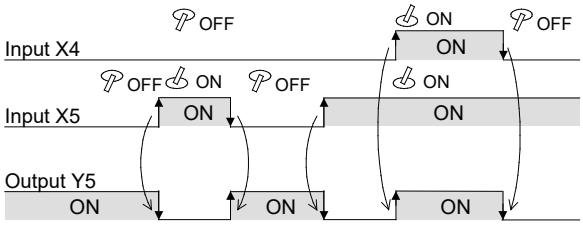
- If the input condition X4 or X5 is "ON", Y4 is "ON".
- If X4 and X5 are both "OFF", Y4 is "OFF".

[Program of a parallel circuit (2)]

Ladder display



<<Action>>

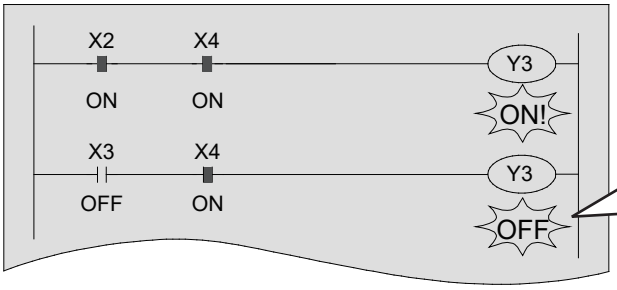


- If the input condition X4 is "ON" or X5 is "OFF", Y5 is "ON".
- If X4 is "OFF" and X5 is "ON", Y5 is "OFF".

Important point

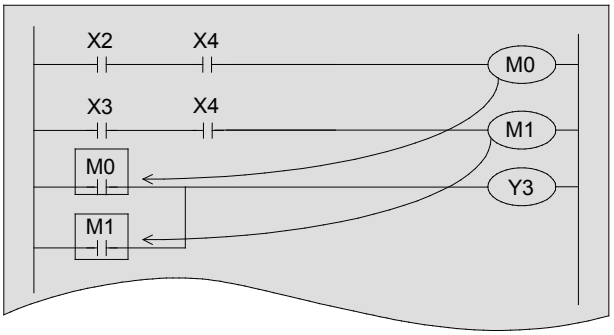
Prohibition of double output (double coil), and countermeasures

- Note that it is prohibited to specify multiple OUT instructions to a single coil.



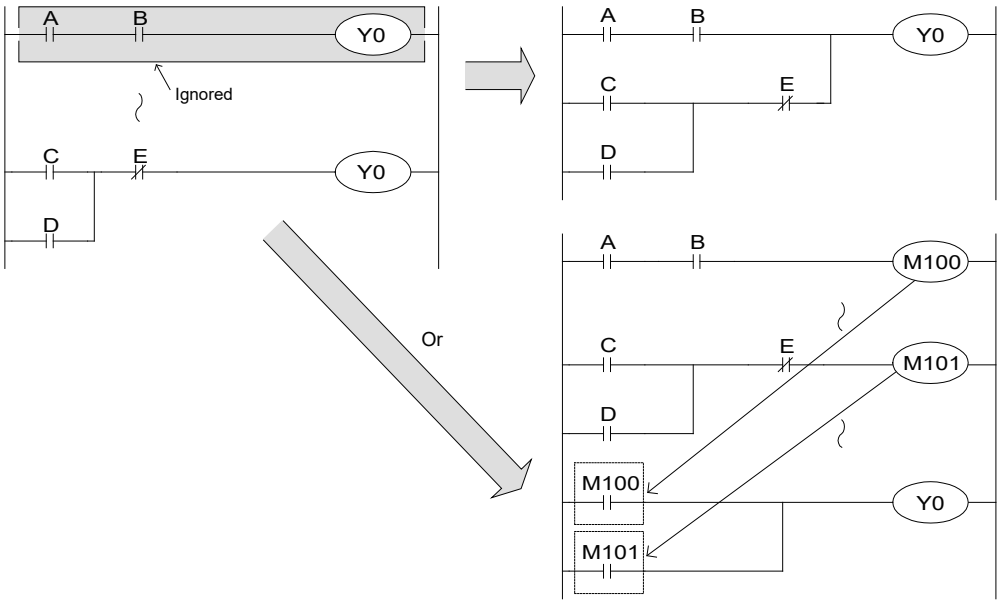
The output results are OFF, and the ON/OFF state of the OUT instruction activated at the end of one scan is output.

Change it like this.



A program with redundant outputs (double coils) is an errorless program. However, the operation becomes complex and therefore it is recommended to change the program as follows.

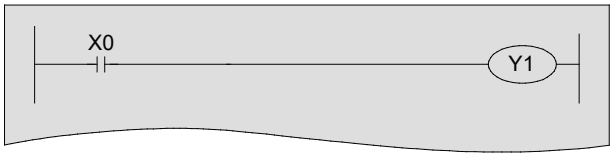
Another changing example



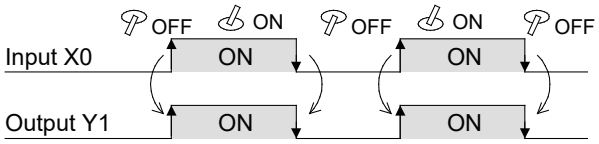
4.3.2 Difference between OUT instruction and SET/RST

(1) [OUT instruction] OUT (Coil driving instruction)

Ladder display



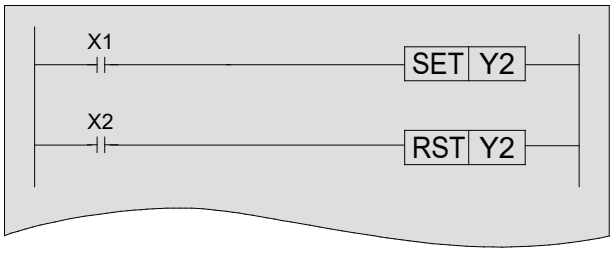
<<Action>>



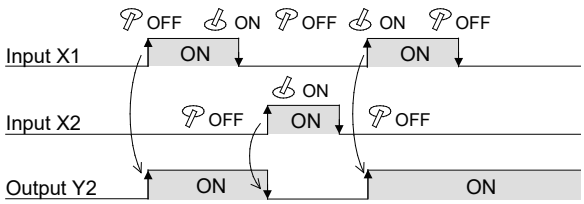
- If the input condition is "ON", the OUT instruction will turn on the specified device.
- If the input condition is "OFF", the specified device will also be turned "OFF".

(2) [SET/RST instruction] SET (Instruction to maintain the energized status), Reset (Instruction to reset the energized status)

Ladder display



<<Action>>



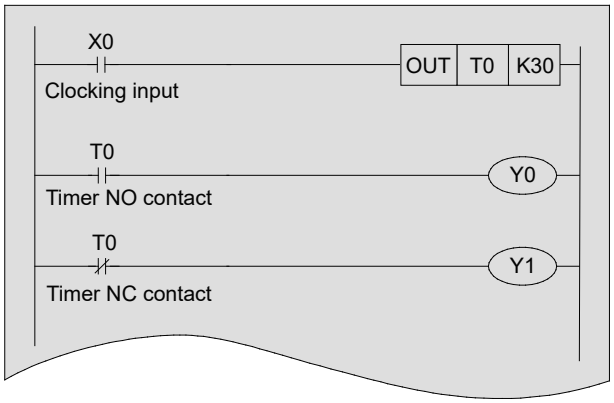
- If the input condition is "ON", the SET instruction will turn on the specified device and keep it "ON" even after the input condition turns "OFF".
- In order to turn "OFF" the set device, use the RST instruction.

4.3.3 Clocking of timers

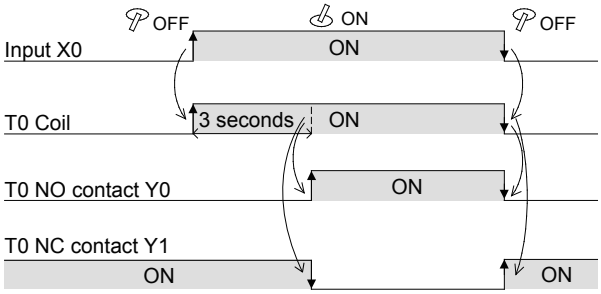
Timers count with clock pulses of 1 ms, 10 ms, 100 ms and so on. When they reach their set value, the output contact turns on. (On-delay timer)
The set value may be a constant (K) or indirectly specified by a value in a data register (D).
The timers include the Timer (T) for which the current value is set to 0 when the timer coil turns OFF, and the retentive timer (ST) that holds the current value even if the coil turns OFF.

(1) Timer

Ladder display



<<Action>>



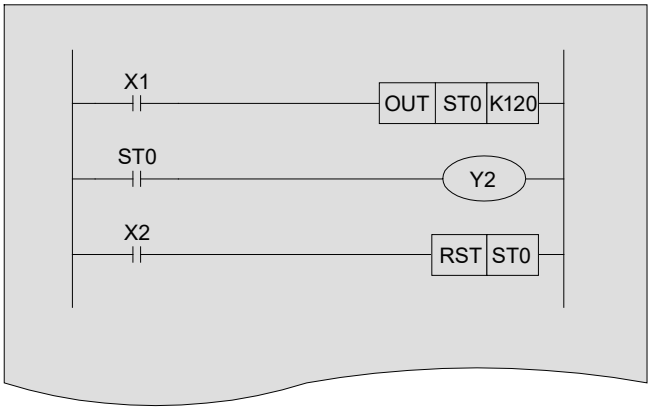
- If the input condition is "ON", the timer T0 begins clocking, and the T0 contact turns "ON" after the specified period (T0: 100 ms base × 30 = 3 seconds).
- If X0 is "OFF", the clocking of the timer is reset and the contact T0 turns "OFF".

Reference

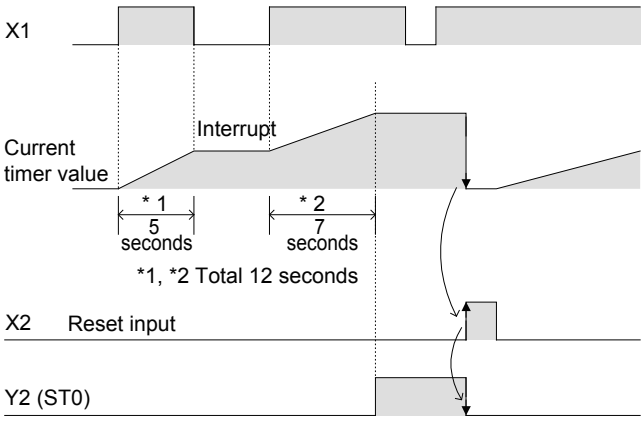
- The values of timers and counters can also be set with a data register (D). (Indirect specification of the value)

(2) Retentive timer

Ladder display



<<Action>>



- The timer operates only when the clock input X1 is "ON". The clock is interrupted when the input turns "OFF".
- The output contact of the timer operates when the total "ON" time of the input X1 reaches the predetermined value.
- If the reset input X2 is "ON", the current value of the timer will become 0 and the output contact will be "OFF".

Reference

Types of timers

Timer module:	The timer is a low-speed timer, timer, or high-speed timer according to the timer designation (how the instruction is written). For example, the same T0 can be a low-speed timer (100 ms) if specified as OUT T0, can be a timer (10 ms) if specified as OUTH T0, and a high-speed timer (1 ms) if specified as OUTHS T0. This also applies to the retentive timer.
Set value of the timer:	The constant K is an integer from 1 to 32,767. If the constant is K120, it is 12 seconds for low-speed timers based on 100 ms, and 0.12 seconds for high-speed timers based on 1 ms.
Latched function:	Even if the power is turned off during clocking, the current value of the timer will be saved and the timer will operate according to the total driving time before and after the power is off.

4.3.4 Counting of counters

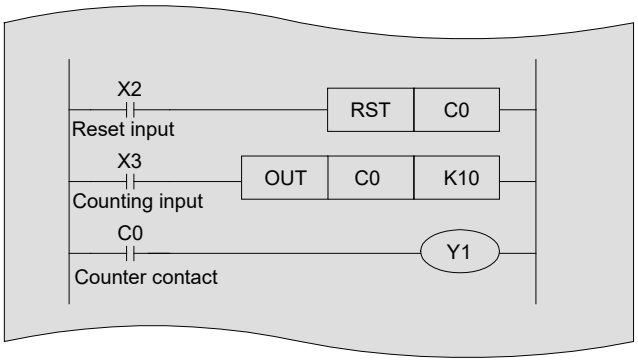
This section describes the counter (C) and long counter (LC).

Features of the counter

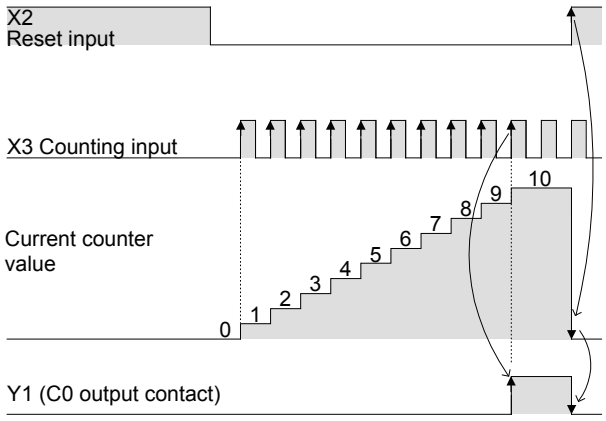
The features of the counter (C) and long counter (LC) are described below. They can be used according to the count range.

Item	Counter (C)	Long counter (LC)
Counting direction	Up-counting	Same as the left.
Current value register	16 bits	32 bits
Set value	1 to 32767	1 to 4294967295
Specification of the set value	By the constant K or data register	Same as the left. However, the data registers are used in pair form (2 registers).
Change of the current value	No changes after counting up	Same as the left.
Output contact	Latched after counting up	Same as the left.
Reset operation	The current value of the counter will become 0 and the output contact will be restored when the RST instruction is executed.	

Ladder display



<<Action>>



- The counter's current value increments each time the count input X3 changes from "OFF" to "ON". When this value reaches the set value, the output contact activates.
- After reaching the predetermined value, the current value and the output contact keep their status.
- When reset input X2 turns "ON", the counter's current value changes to 0, and the output contact is restored.

Reference

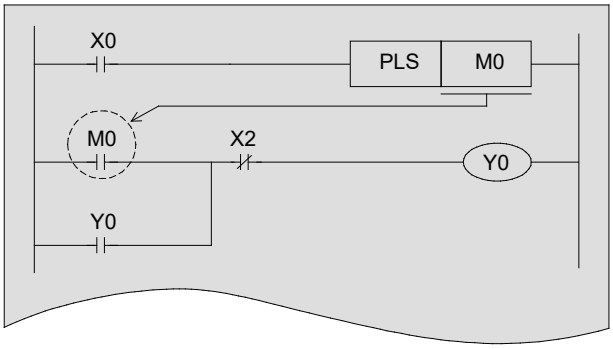
High speed inputs can be counted with high speed counters

If a high speed counter is used, inputs will not be missed and high speed signals can be counted.
For details on high speed counters, see Chapter 10.

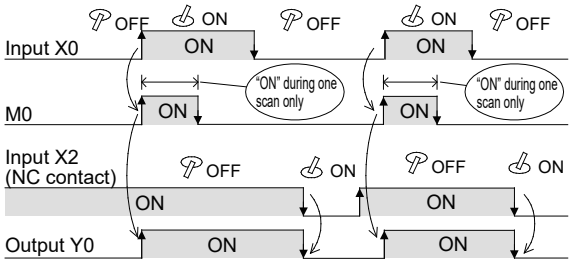
4.3.5 PLS/PLF instruction

(1) [PLS instruction] Pulse (Rising edge pulse output)

Ladder display



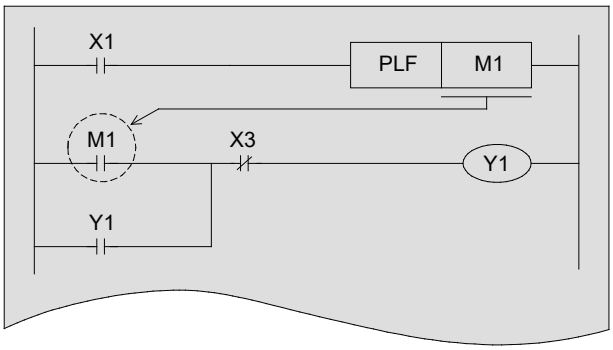
<<Action>>



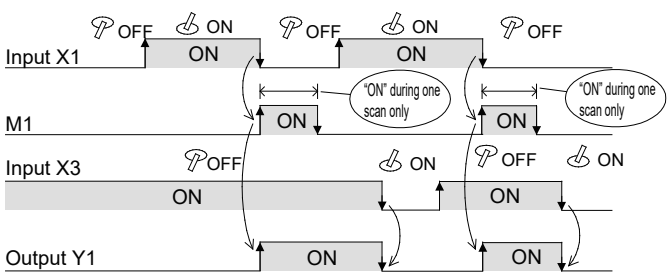
- When the input condition X0 turns "ON", the specified device turns "ON" for just one scan (1 operation cycle) at the rising edge.

(2) [PLF instruction] Pulse falling (Falling edge pulse output)

Ladder display



<<Action>>



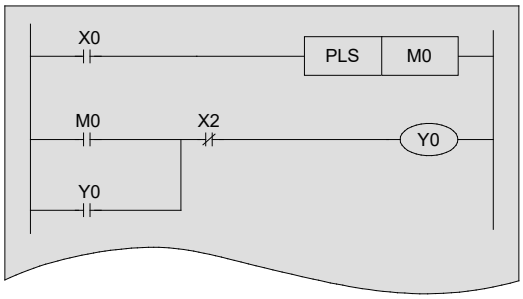
- When the input conditions X1 changes from "ON" to "OFF", the specified scan turns "ON" for just one scan (1 operation cycle) at the falling edge.

Reference

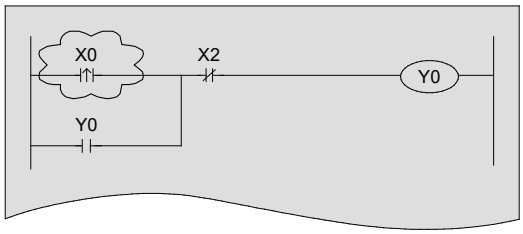
Simplification by using the rising/falling edge pulse contact instructions

If the rising edge pulse contact —|↑|— and the falling edge pulse contact —|↓|— are used, the operation of the previously described PLS/PLF instruction can be written more simply.
They can be used according to the content and function of the following programs.

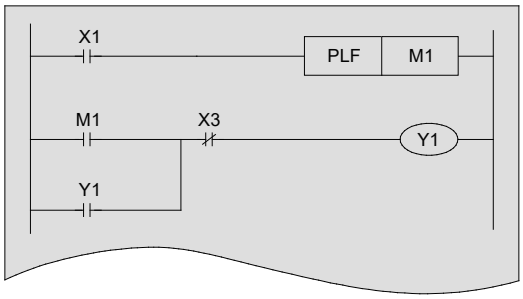
[When the PLS instruction is used]



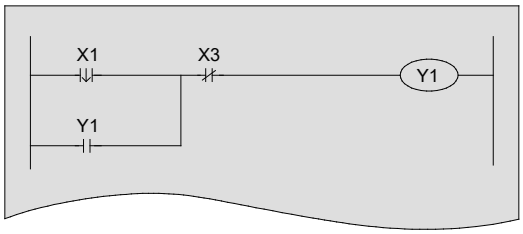
[When the rising edge pulse contact instruction is used]



[When the PLF instruction is used]



[When the falling edge pulse contact instruction is used]

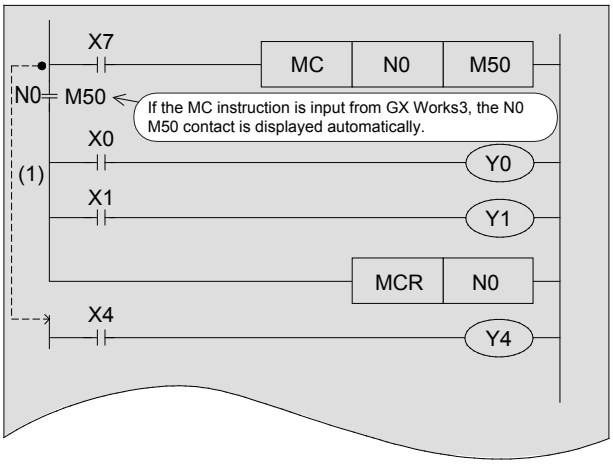


4.3.6 MC/MCR instruction

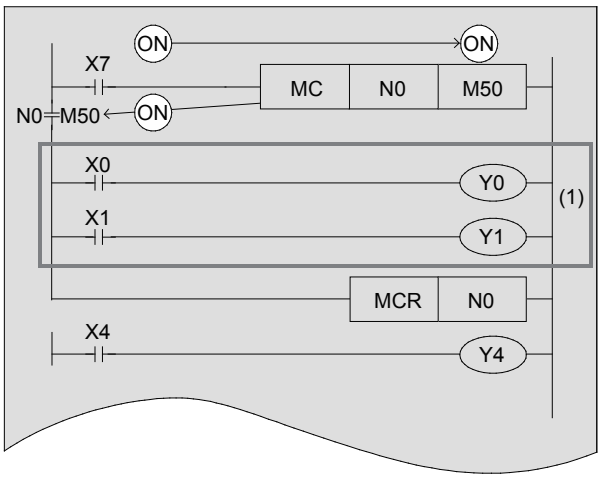
[MC instruction] Master Control (Denotes the start of a master control block)

[MCR instruction] Master Control Reset (Denotes the end of a master control block)

Ladder display



<<Action>>



- While the input condition X7 is "ON", the circuit indicated by (1) becomes valid, [Y0 is "ON" if X0 is "ON"], and [Y1 is "ON" if X1 is "ON"].
- When X7 is "OFF", Y0/Y1 does not operate.
- Since Y4 is not subject to MC/MCR, it turns "ON and OFF" independently according to the operation of X4.

Point

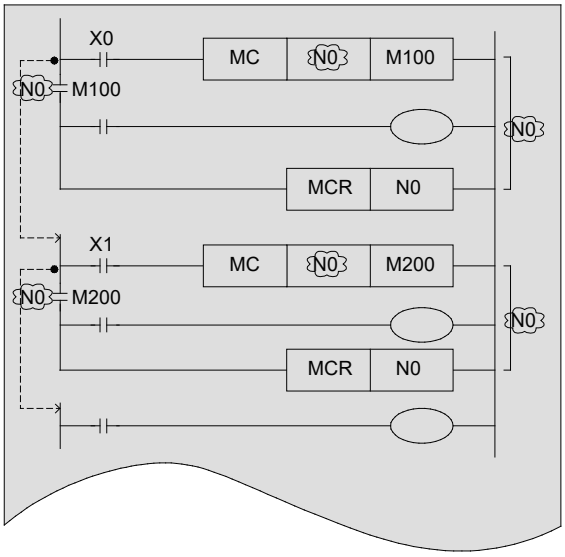
Status of MC/MCR block devices when MC does not operate

- Held with the current status: Devices driven by retentive timer values, counter values and SET/RST.
- OFF: Devices driven by non-retentive timers and devices driven by OUT instructions. With a regular timer, the current value is also set to 0.

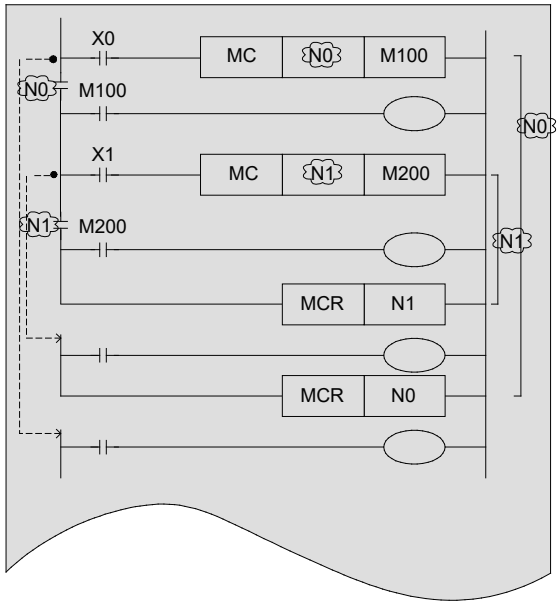
Reference

Nesting with MC/MCR

[No nest structure]
The nesting number N0 is consecutively used to program.
(No limitation on the number of use)



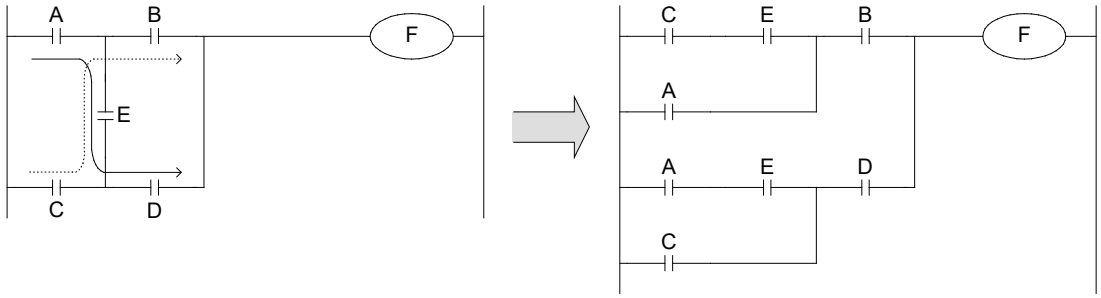
[Nest structured]
Nesting numbers N0 to N14 are sequentially used from the small number to big one to program. (Max. 15 layers)



4

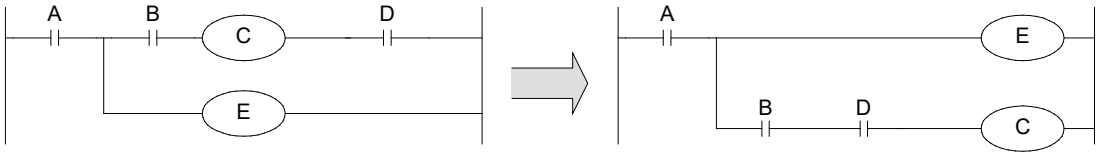
4.3.7 Non-programmable circuits and solutions

(1) Bridge Circuit



A circuit in which current flows in both directions must be rewritten as shown above. (The circuit when having no D, and the circuit when having no B, in parallel. It was connected.)

(2) Location of Coil

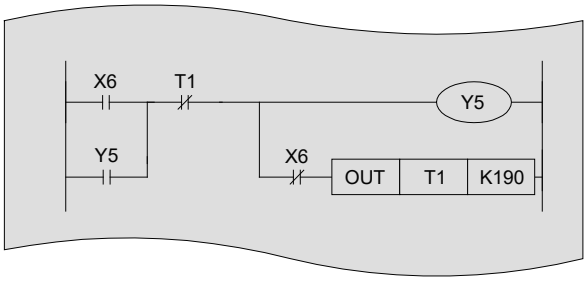


- Contacts cannot be located on the right side of coils.
- It is recommended that coils internally used between contacts be programmed prior to the output.

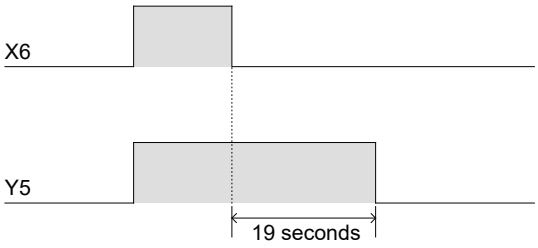
4.4 Circuit examples with PLC instructions

(1) Off delay timer

Ladder display



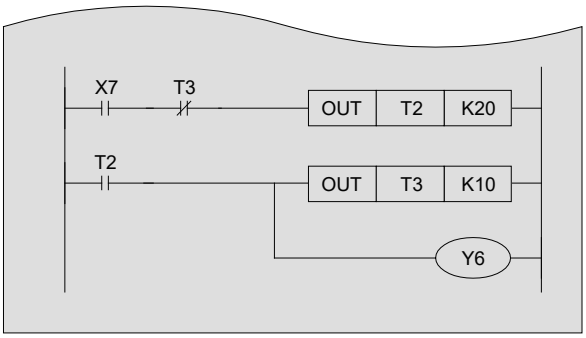
<<Action>>



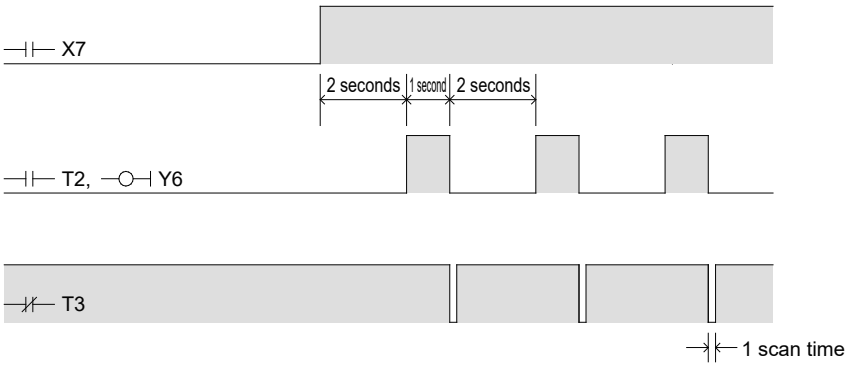
- Y5 will turn OFF 19 seconds after X6 turns OFF.
  A timer, which turns the input contact on or off with a certain time delay if the input contact is OFF, is referred to as off delay timer.

(2) Flickering (Flashing)

Ladder display



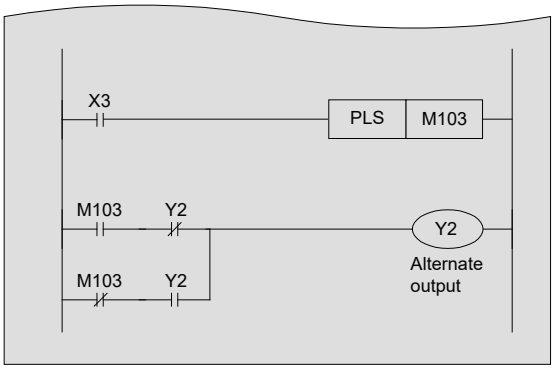
<<Action>>



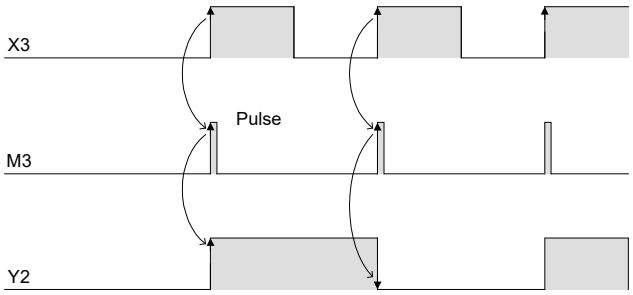
- Two seconds after X7 turns ON, a flashing operation is repeated in which Y6 turns ON for one second and then turns OFF for two seconds.

(3) Alternate Circuit with pulse output circuit (Alternating operation circuit)

Ladder display



<<Action>>

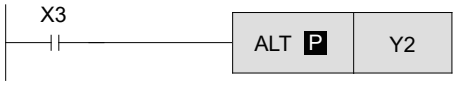


- Once X3 is ON, Y2 is ON. If X3 is ON again, Y2 is, in turn, OFF. (Alternating operation)

Reference

- If the circuit is written with a different instruction, the circuit can be simplified as follows:

● Alternate circuit



This circuit is equivalent to the above relay ladder circuit.

- Although relay ladder circuits can represent many controls, PLC instructions have the ability to greatly simplify the relay ladder circuits.

Chapter 5

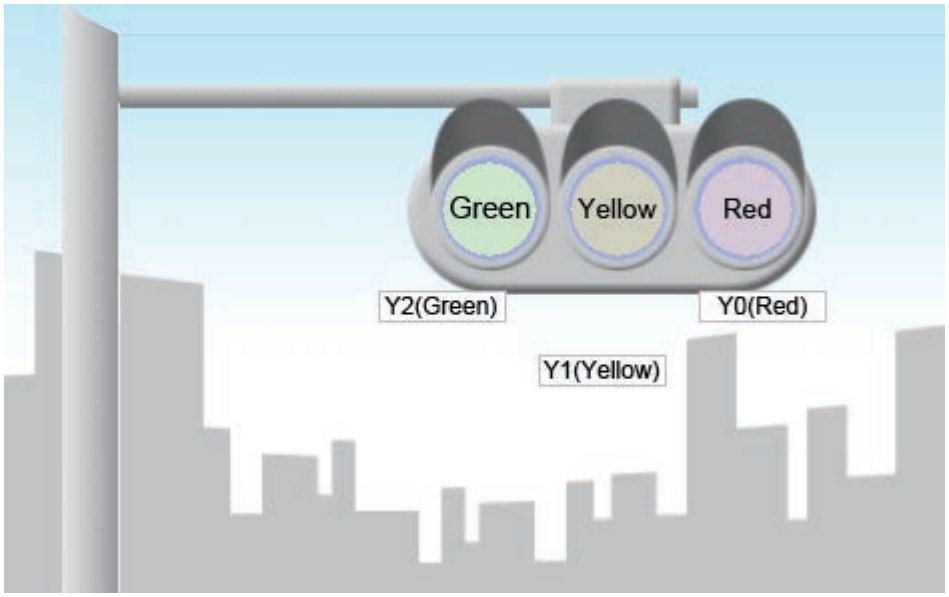
INTRODUCTION EXAMPLES AND PROGRAM OPERATION

Let's practice…

Through monitoring the program examples outlined in this chapter, the user can master sequence programming.

In this chapter, you will practice programming using the universal simulation module FX5U-32MT-SIM and with a personal computer simulating the engineering tool.
For the operation of the GX Works3, see Chapter 3.

5.1 Introduction example <<1>> [Traffic light control]



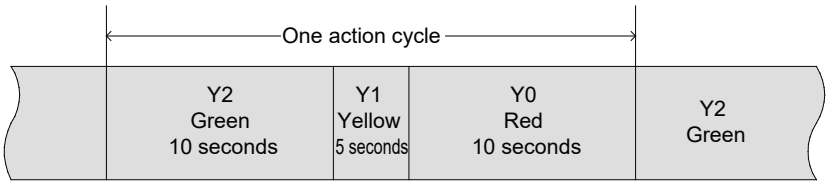
<<Assignment of I/O>>

Input	
X0	Control start
X1	Repeated stop

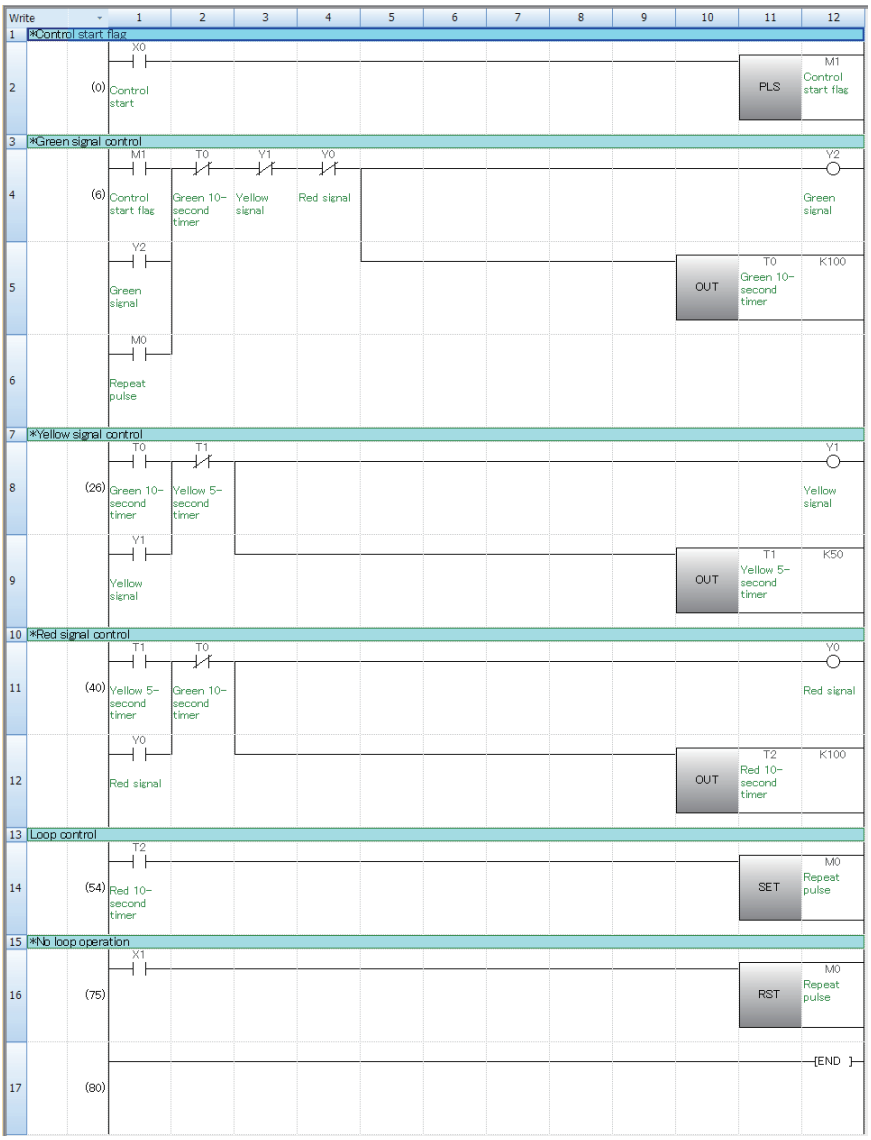
Output	
Y0	Red signal
Y1	Yellow signal
Y2	Green signal

<<Control specification>>

If the PLC is running, the lamps of the traffic light are operated in the following order.
The operations are repeated after one action cycle. However, when X1 is ON, the operation will stop after one cycle.



<<Example of sequence program with comments>>



If X0 turns ON, M1 will turn ON for one scan cycle.

If M1 turns ON for one scan cycle, Y2 (Green signal) will turn ON for 10 seconds.

If Y2 (Green signal) turns OFF, Y1 (Yellow signal) will turn ON for 5 seconds.

If Y1 (Yellow signal) turns OFF, Y0 (Red signal) will turn ON for 10 seconds.

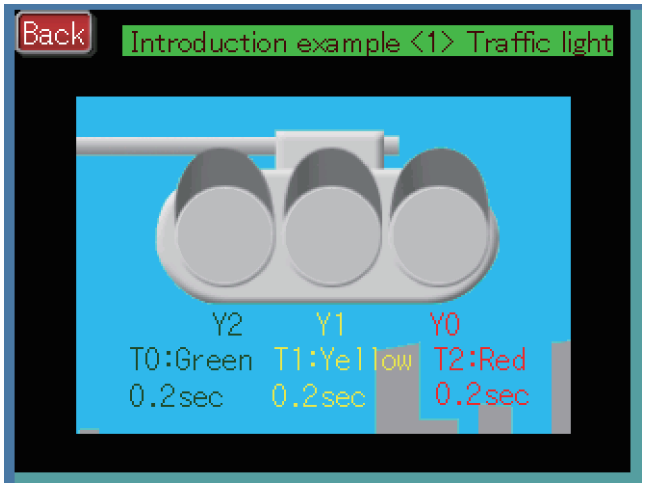
If Y0 (Red signal) turns OFF, the program is repeated from the control of the green signal.

When X1 input is ON, repeated control is prohibited.

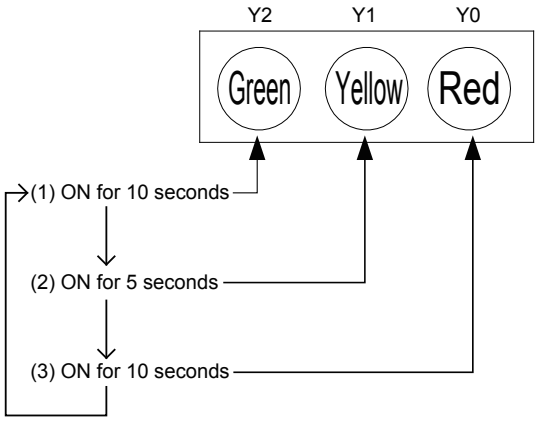
<<Operation check>>

Use GX Works3 to monitor the circuit.

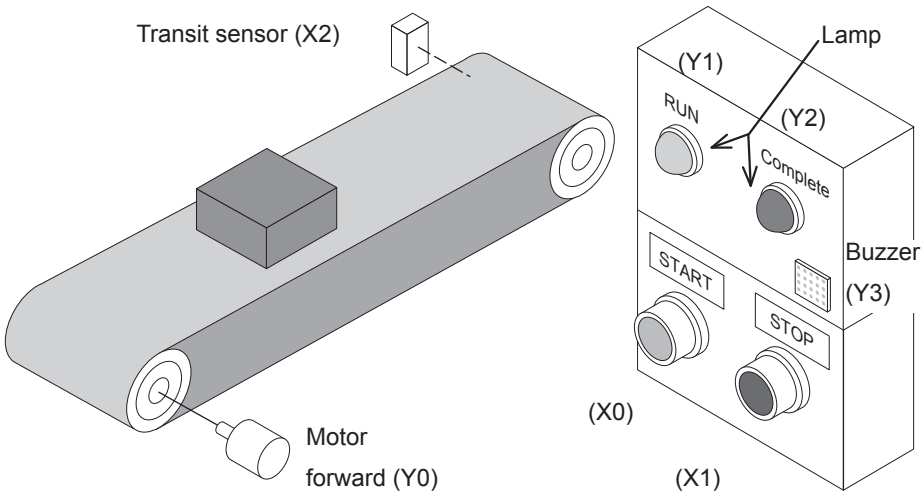
Training Machine Screen **[Basics: MELSEC iQ-F Programming (GX Works3 Version)]**



- If X0 turns ON, the signals will turn on in the following order.



5.2 Introduction example <<2>> [Conveyor control]



<<Assignment of I/O>>

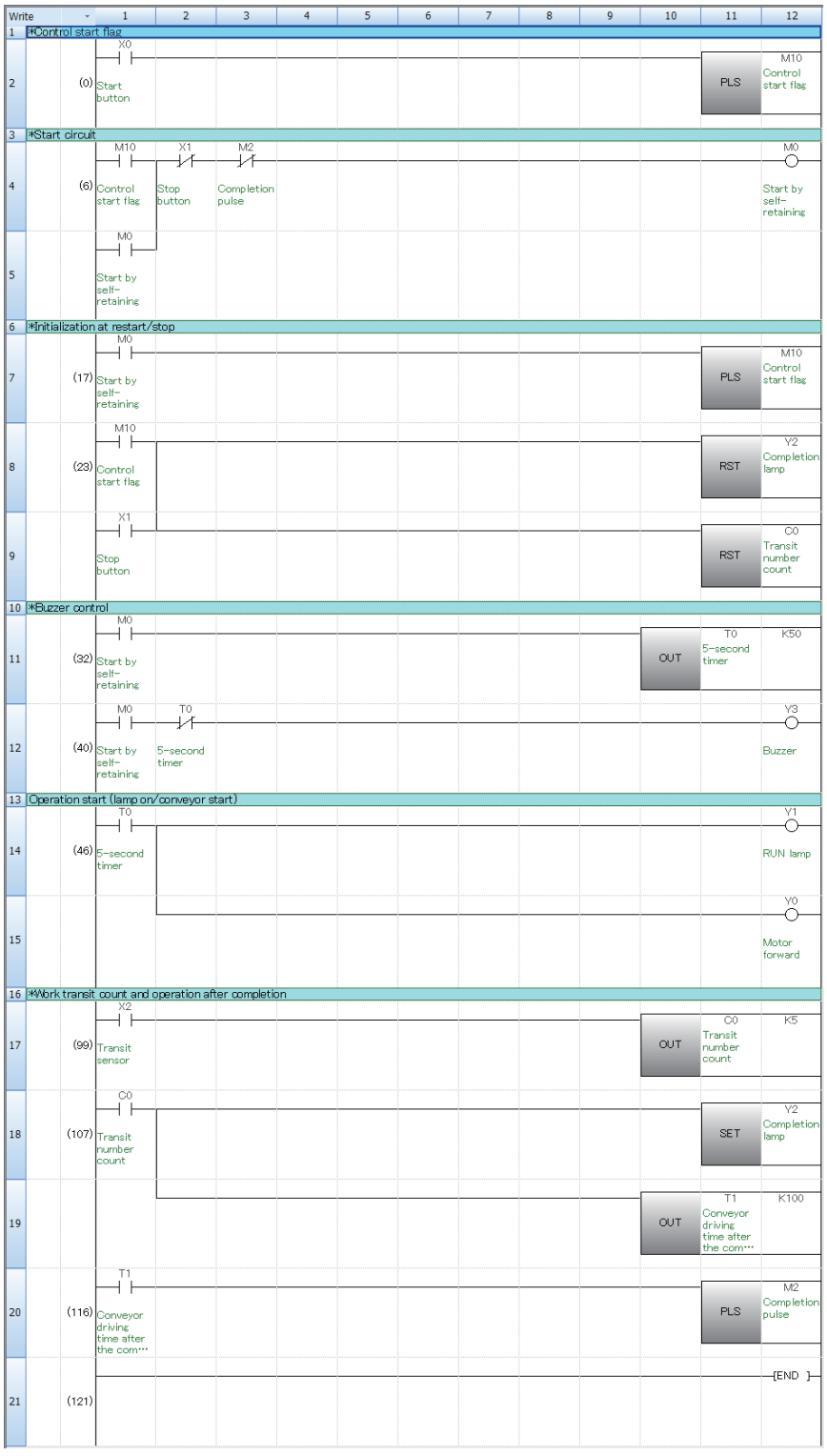
Input	
X0	Start button
X1	Stop button
X2	Transit sensor

Output	
Y0	Motor forward
Y1	RUN lamp
Y2	Complete lamp
Y3	Buzzer

<<Control specification>>

(1) If [Start button (X0)] is pressed, [Buzzer (Y3)] sounds for 5 seconds.
(2) After that, [Motor forward (Y0)] is activated, and the conveyor begins operating. [RUN lamp (Y1)] is on when the motor is rotating in the forward direction.
(3) If [Transit sensor (X2)] detects 5 workpieces, [Complete lamp (Y2)] is turned on and the conveyor stops in 10 seconds.
(4) Stop the control by [Stop button (X1)]. Turn on [Start button (X0)] to restart.

<<Example of sequence program with comments>>



If X0 turns ON, M10 will turn ON for one scan cycle.

If M10 turns ON for one scan cycle, M0 is self-retained.

Before the operation, the pulse signal M1 is transmitted to initialize the status of the lamp and the current value of the counter.

Buzzer (Y3) sounds for 5 seconds.

RUN lamp (Y1) and motor forward (Y0) turn on after the buzzer (Y3) stops.

The counter (C0) counts the number of times the transit sensor (X2) turns on.

The completion lamp (Y2) will turn ON after the completion of the count.

The conveyor continues operating for 10 seconds by the timer (T1) after the completion of the count.

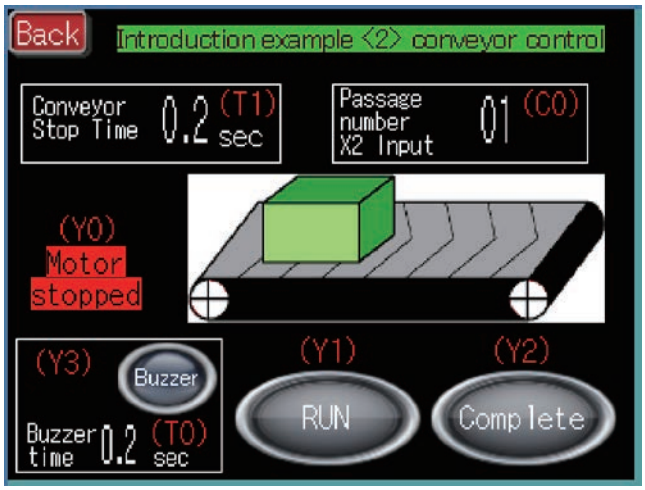
The pulse to stop the operation will be ON after the conveyor stops.

<<Operation check>>

Use GX Works3 to monitor the circuit.

Training Machine Screen

[Basics: MELSEC iQ-F Programming (GX Works3 Version)]



(1) X0 (Start button) is turned ON. ⇨ Y3 (Buzzer) is turned on for 5 seconds.

(2) 5 seconds later ⇨ Y1 (RUN lamp) is turned on, and Y0 (Motor forward) will be ON.

(3) X2 (Transit sensor) is turned on for 5 times. ⇨ Y2 (Completion lamp) is turned on after the fifth ON signal is detected.

(4) 10 seconds later ⇨ Y1 (RUN lamp) and Y0 (Motor forward) will be OFF, and the conveyor stops.

MEMO

Chapter 6
EXPRESSING NUMERIC VALUES (DATA)

It is wasteful to use PLC as a simple relay board…

The PLC is not only used as a substitute for the relay board.
In recent times, the PLC has become a powerful machine with higher added value.
In order to master this machine, it is necessary to learn the instructions that handle numeric values described in this textbook.

In order to learn numeric data handling …

This chapter describes the basics of the values handled by the PLC and the numeric data storage destination, etc.

Simple introductions for easy understanding…

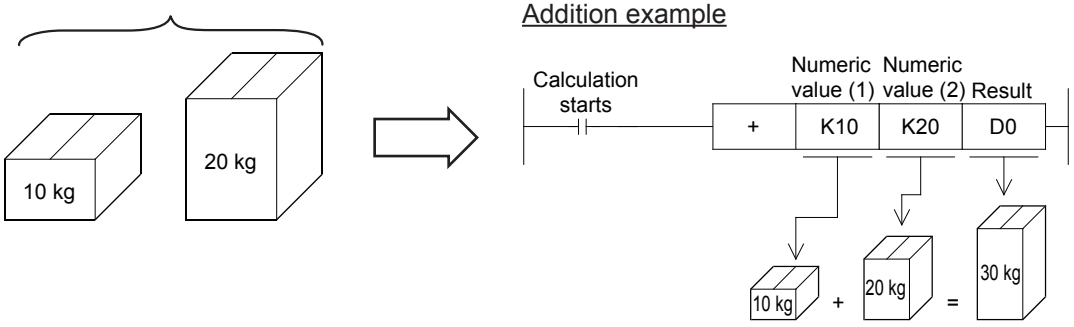
To begin, let's examine applied instructions without a complicated environment. This approach will be valuable for using applied instructions in the future.

6.1 Numeric values

Along with the instructions for simple ON/OFF control as just described, instructions can be used for simple numeric data operation.
Operations for simple addition and comparison are described below.

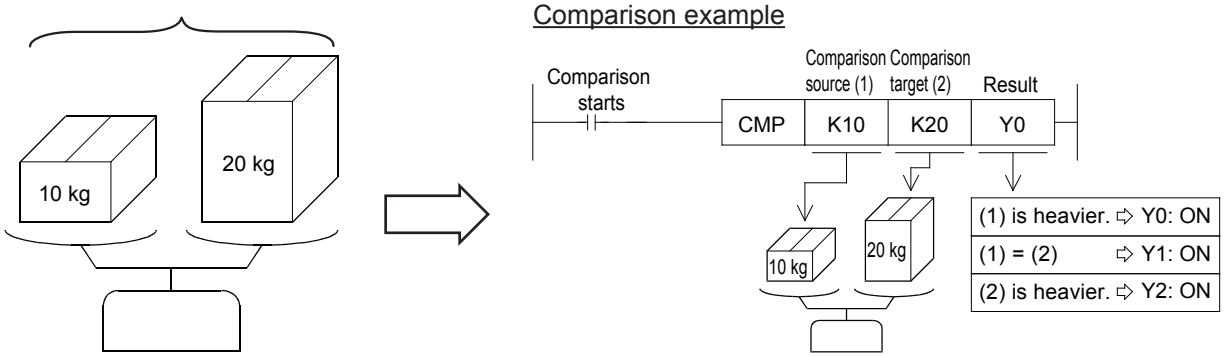
<<Arithmetic operation>>

Summing two products



<<Comparison>>

Which is heavier?



Reference

Types of instructions

These are types of instructions.

(1) Dealing with numeric data regarding comparison, and arithmetic operations.

(2) Controlling program flow, executing jumps, subroutines, loops, interrupts and so on.

(3) Executing data communications with various devices using FROM/TO instructions and other dedicated instructions.

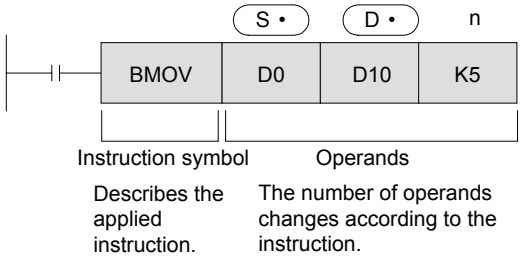
(4) Target-oriented instructions such as those to cut down the number of I/Os, those to execute high-speed processing, and instructions similar to the ALT instruction described before.

To use many instructions, it is necessary to know what "numeric data" means for a PLC and where "numeric data" is stored in a PLC.
This will be described in the latter half of this chapter.

Point

Understanding instructions and how to input them using GX Works3

- Each instruction in the PLC includes a mnemonic to describe the instruction. Following the instruction symbol, operands are used to define the devices and numeric values for processing.



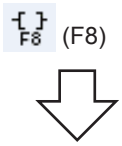
(S •) : The so-called source is the operand which does not change according to the execution of the instructions.

(D •) : The so-called destination is the operand which changes according to the execution of the instructions.

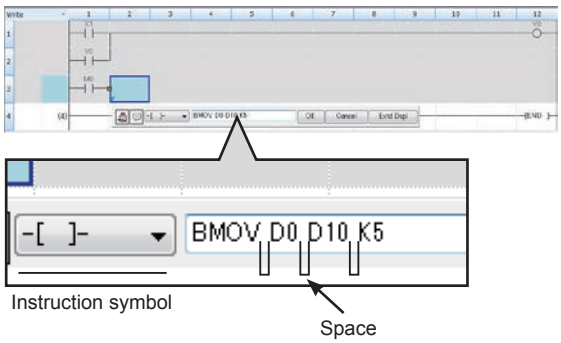
m, n: Operands that specify the number of devices, number of transfers, number of data items, and number of strings, etc.
Operands that can be indexed are indicated with "·" (indexing is described later).
Example (S •) (D •)

- The input method of GX Works3

(1) The applied instructions are turned on by using contacts similar to the OUT instructions and SET instructions. (Some applied instructions do not need contacts.)



(2) Press F8 or [F8] key.
* If the mnemonic has been directly input, the above operation can be omitted.



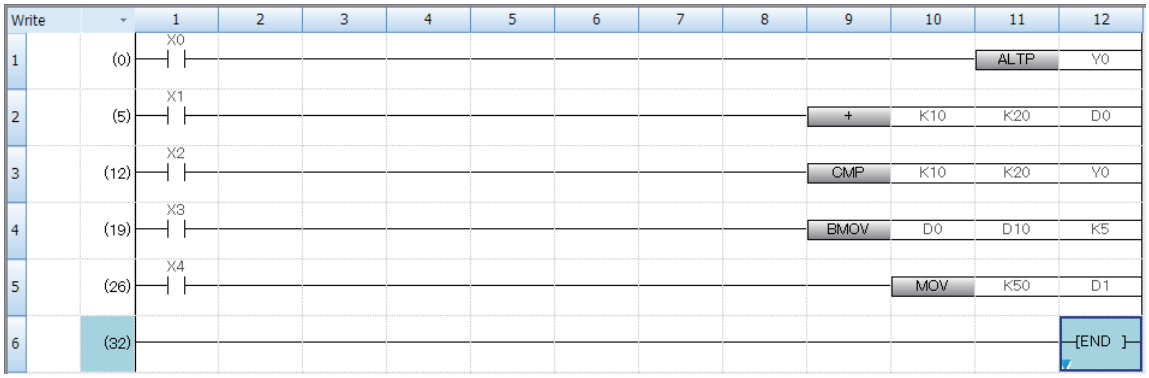
(3) After the mnemonic, input necessary operands and separate each of them with a space.

(4) Press [Enter] key or click [OK].

6

<<Let's try inputting>>

We will study about the command functions and operations, so let's try inputting a ladder with GX Works3.
Refer to the "Point" on the previous page for details on inputting.



6.2 Numeric values used in a PLC

6.2.1 Decimal numbers

In our daily lives, we use decimals.
The micro PLC uses decimals for the following applications.

(1) PLC device No. (excluding input relay (X) and output relay (Y))
[Typical examples]

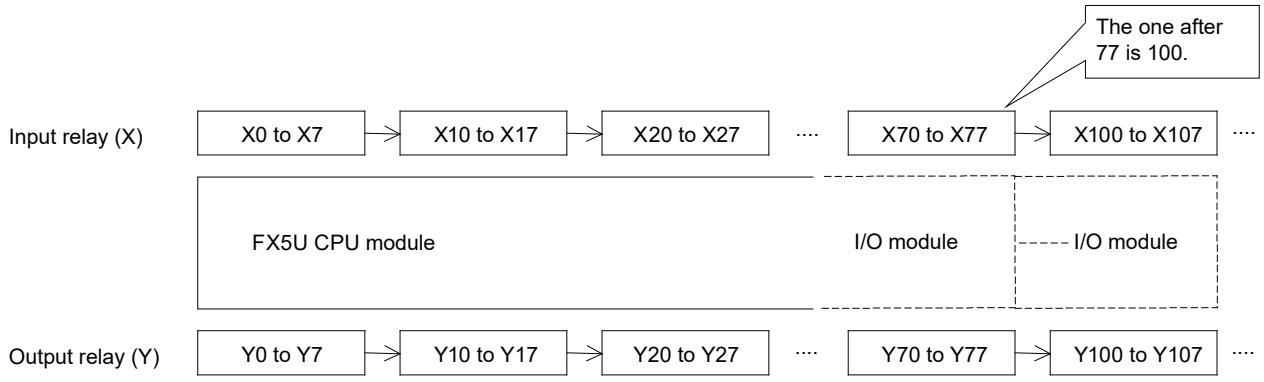
Internal relay (M)	M0	,M1	,M2	…	M8	,M9	,M10	,M11	…
Timer (T)	T0	,T1	,T2	…	T8	,T9	,T10	,T11	…
Counter (C)	C0	,C1	,C2	…	C8	,C9	,C10	,C11	…
Data register (D)	D0	,D1	,D2	…	D8	,D9	,D10	,D11	…

(2) When constants (K) are used for timers, counters, or numeric values in applied instructions, a decimal number is used with "K," such as "K20" to represent the constant.

6.2.2 Octal numbers

Octal numbers are base octal numbers, which use eight unique digits from 0 through 7 as follows: 0 to 7, 10 to 17, 20 to 27…
The micro PLC uses octal numbers as device numbers for input relays (X) and output relays (Y).

- **The micro PLC I/O numbers are represented with octal numbers**



For information on how to assign I/O numbers according to the structure of the system, see "2.5.4 micro PLC I/O number assignment".

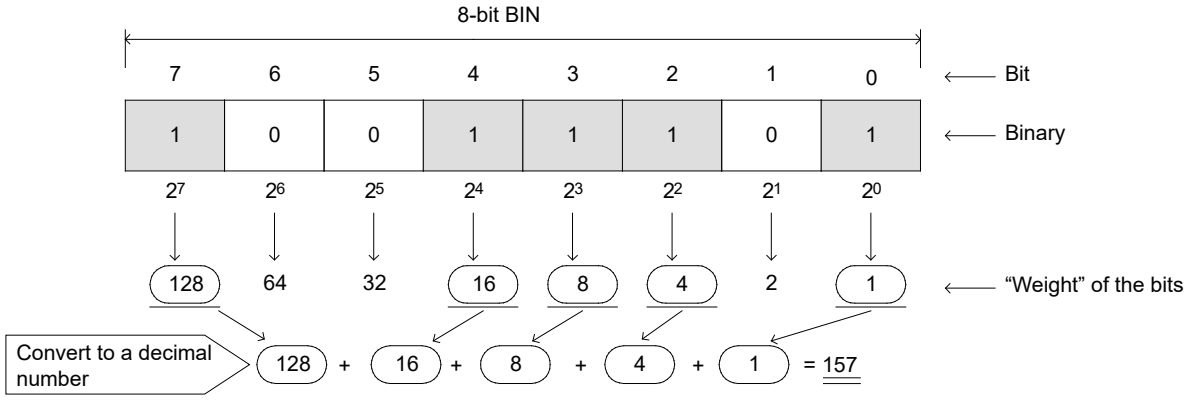
6.2.3 Binary numbers

The numbers most people are familiar with are decimal numbers, which use a base 10 representation.
Computers and PLCs, however, use binary numbers, which utilize a base 2 system.
For example, it is convenient to use the binary number 0 or 1 to correspond to the ON/OFF status of memory locations and relays.
Let's spend some time to learn the difference between binary and decimal numbers.

[Comparison of binary and decimal]

	Decimal	Binary	
	0	0000	0000
	1	0000	0001
	2	0000	0010
	3	0000	0011
	4	0000	0100
	5	0000	0101
	6	0000	0110
	7	0000	0111
	8	0000	1000
	9	0000	1001
	10	0000	1010
	11	0000	1011
	12	0000	1100
	⋮	⋮	⋮
Main use	Constant K, internal device No. (M, D, etc.)	Internal processing of PLC	

- What is the value of the binary number "10011101" if represented in decimal?
  The "weight" of each bit is described below. Sum the "weights" that have a binary number of "1". The result is a decimal value.



The value of binary number "10011101" is "157" in decimal format.

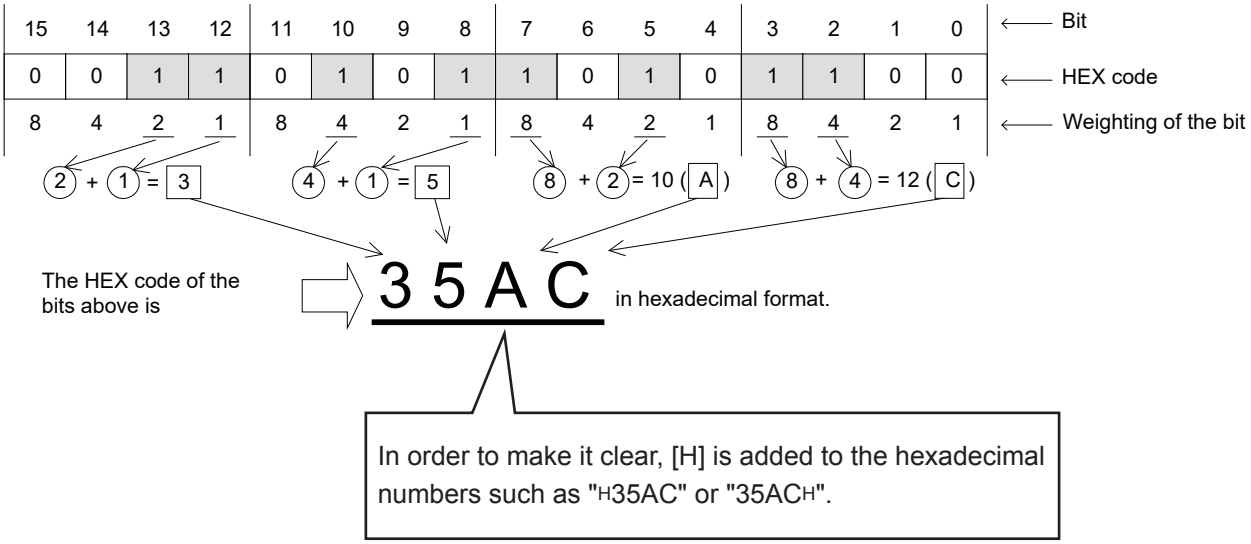
6.2.4 Hexadecimal numbers

There are 4 bits for each digit of a hexadecimal number, and each digit of a hexadecimal number is represented by 0 to 9, A (10), B (11), C (12), D (13), E (14), F (15).
Hexadecimal numbers are base 16 numbers, and the 16th value is F.

[Comparison of decimal and hexadecimal]

	Decimal	Hexadecimal	Binary	
	0	00	0000	0000
	1	01	0000	0001
	2	02	0000	0010
	3	03	0000	0011
	4	04	0000	0100
	5	05	0000	0101
	6	06	0000	0110
	7	07	0000	0111
	8	08	0000	1000
	9	09	0000	1001
	10	0A	0000	1010
	11	0B	0000	1011
	12	0C	0000	1100
	13	0D	0000	1101
	14	0E	0000	1110
	15	0F	0000	1111
	16	10	0001	0000
	⋮	⋮	⋮	⋮
Main use	Constant K, internal device No. (M, D, etc.)	Constant H and so on	Internal processing of PLC	

- With the micro PLC, there are cases when a hexadecimal is used for the operation setting (buffer memory setting), etc., when using an intelligent function module.



6.2.5 Binary-coded decimal numbers (BCD code)

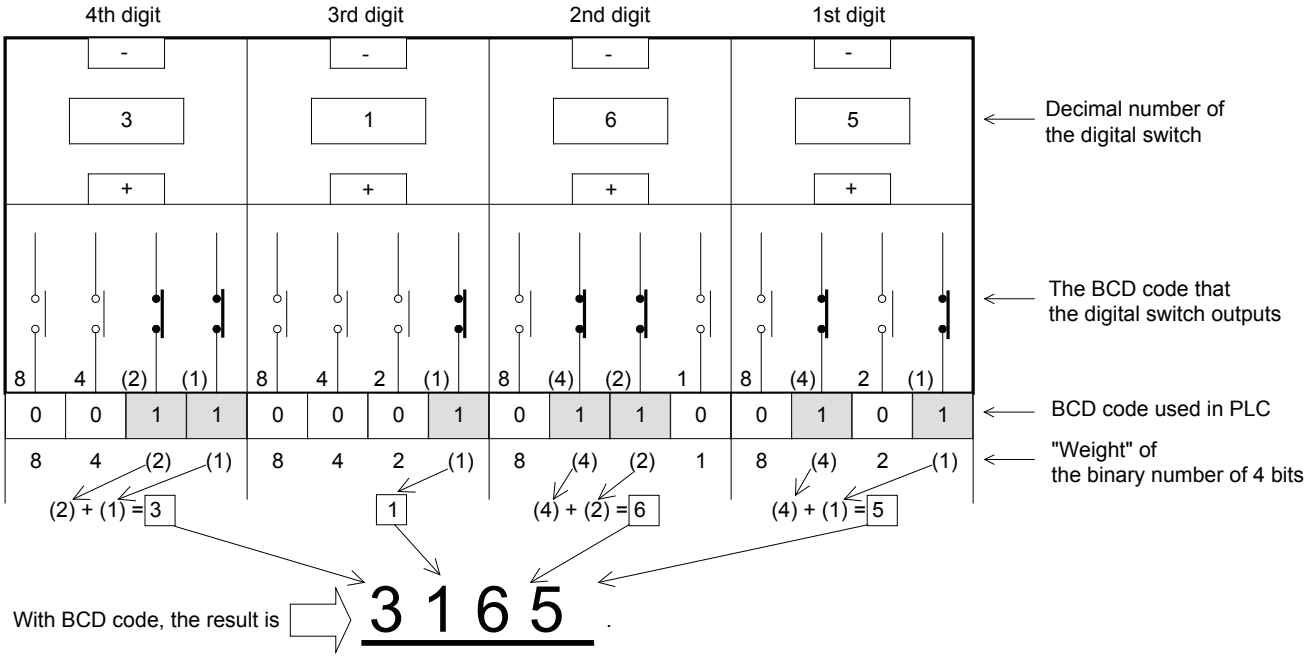
The decimal values 0 to 9 of each digit can be represented in binary numbers of 4 bits by BCD code.
BCD code is used for the output signal of a digital switch, the control signal of a 7-segment display module, or for the signals of various measuring instruments.

[Comparison of decimal and BCD code]

	Decimal	BCD	
	0	0000	0000
	1	0000	0001
	2	0000	0010
	3	0000	0011
	4	0000	0100
	5	0000	0101
	6	0000	0110
	7	0000	0111
	8	0000	1000
	9	0000	1001
	10	0001	0000
	11	0001	0001
	12	0001	0010
	⋮	⋮	⋮
Main use	Constant K, internal device No. (M, D, etc.)	BCD digital switch, 7-segment display module	

- BCD code uses binary numbers of 4 bits to represent one digit 0 to 9 of a decimal number.

[Example of 4-digit BCD code (decimal "3165") output by digital switch]



Reference

Numeric values used in the micro PLC

Comparison of decimal and other numeric values

Decimal	Octal	Hexadecimal	Binary		BCD	
0	0	00	0000	0000	0000	0000
1	1	01	0000	0001	0000	0001
2	2	02	0000	0010	0000	0010
3	3	03	0000	0011	0000	0011
4	4	04	0000	0100	0000	0100
5	5	05	0000	0101	0000	0101
6	6	06	0000	0110	0000	0110
7	7	07	0000	0111	0000	0111
8	10	08	0000	1000	0000	1000
9	11	09	0000	1001	0000	1001
10	12	0A	0000	1010	0001	0000
11	13	0B	0000	1011	0001	0001
12	14	0C	0000	1100	0001	0010
13	15	0D	0000	1101	0001	0011
14	16	0E	0000	1110	0001	0100
15	17	0F	0000	1111	0001	0101
16	20	10	0001	0000	0001	0110
⋮	⋮	⋮	⋮	⋮	⋮	⋮
99	143	63	0110	0011	1001	1001
⋮	⋮	⋮	⋮	⋮	⋮	⋮

Main use

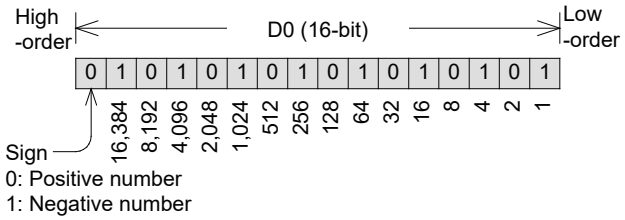
Decimal	Octal	Hexadecimal	Binary	BCD
Constant K, internal device No. (M, D, etc.)	The device numbers of the input relays and output relays	Constant H	Internal processing of PLC	BCD digital switch, 7-segment display module

6.3 Storing numeric data

6.3.1 Operating word devices

A word device is a register used to store 16-bit and 32-bit numeric data. Word device types include the following:

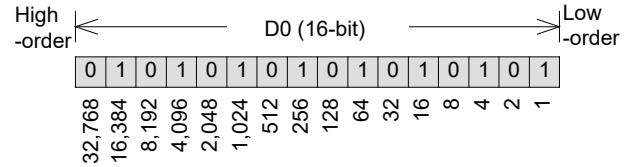
<<Signed 16-bit data>>



A signed16-bit data register can represent numeric values from -32,768 to +32,767.

Example (1) When all 16-bit data are "0", the numeric value is 0.
(2) When only the most significant bit is "0", and the others are "1", it is +32,767.
(3) When all the 16-bit data are "1", it is -1.

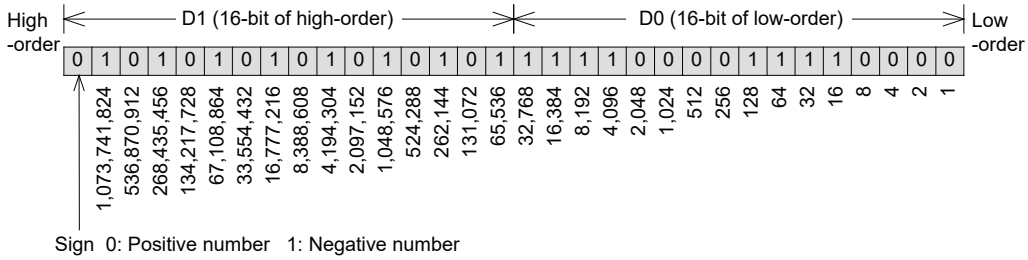
<<Unsigned 16-bit data>>



An unsigned16-bit data register can represent numeric values from 0 to +65,535.

Example (1) When all 16-bit data are "0", the numeric value is 0.
(2) When all the 16-bit data are "1", it is +65,535.

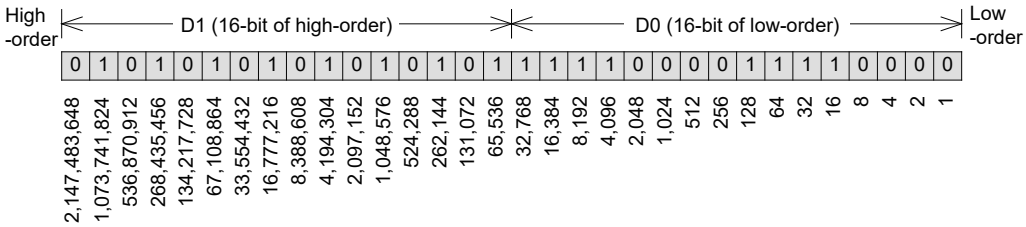
<<Signed 32-bit data>>



32-bit data is represented by a group of data registers with adjacent numbers. (The high-order bits are larger numbers and the low-order bits are small numbers.)
A signed 32-bit data register can represent numeric values from -2,147,483,648 to +2,147,483,647.

Example (1) When all 32-bit data are "0", the numeric value is 0.
(2) When only the most significant bit is "0", and the others are "1", it is +2,147,483,647.
(3) When all the 32-bit data are "1", it is -1.

<<Unsigned 32-bit data>>



32-bit data is represented by a group of data registers with adjacent numbers. (The high-order bits are larger numbers and the low-order bits are small numbers.)

An unsigned 32-bit data register can represent numeric values from 0 to +4,294,967,295.

Example (1) When all 32-bit data are "0", the numeric value is 0.

(2) When all the 32-bit data are "1", it is +4,294,967,295.

6

Point

- For 32-bit data, the device number on the low-order side can be either even or odd. However, in order to avoid confusion, normally the even numbers are used for the low-order side.

Data registers

- Data register

  D0 to D7999 (8,000 points)

- File register

  R0 to R32767 (32,768 points)

- Special type

  SD0 to SD10899 (10,900 points)

- In all of the data registers below, 16 bits are used for one point. 32-bit data can be stored by combining two data registers.

The current value register for timers

- Timer

  T0 to T511 (512 points)

- Retentive timer

  ST0 to ST15 (16 points)

- Each is a 16-bit register, but as a timer, the highest order bit is always "0", and an integer between 0 and 32,767 is handled. (An out-of-range number will lead to an error and the counter will not work.)
- For the timer, the instruction is written with a 100 ms, 10 ms, or 1 ms module.
  Refer to the previous section "Clocking of timers" for details.

The current value register for counters

C0 to C255 (256 points)

- Each is a 16-bit register, but when using as a counter, the highest order bit is always "0", and an integer between 0 and 32,767 is handled. (An out-of-range number will lead to an error and the counter will not work.)

The current value register for long counters

LC0 to LC63 (64 points)

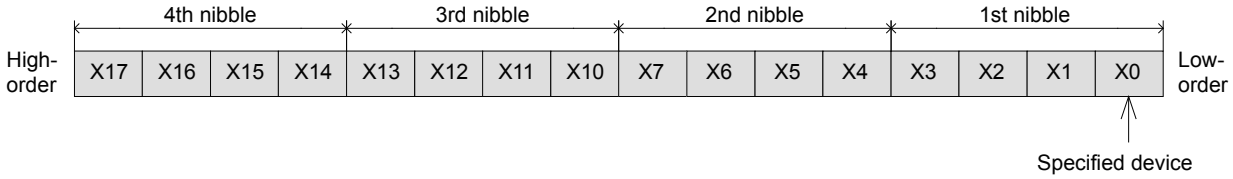
- Each is an unsigned 32-bit register, but when using as a counter, an integer between 0 and 4,294,967,295 is handled. (An out-of-range number will lead to an error and the counter will not work.)
- When timer registers or counter registers are needed for storing 32-bit data, join the two devices with adjacent numbers, as described above.

- The number of device points above is the default parameter setting. The number of device points can be changed with the parameter.

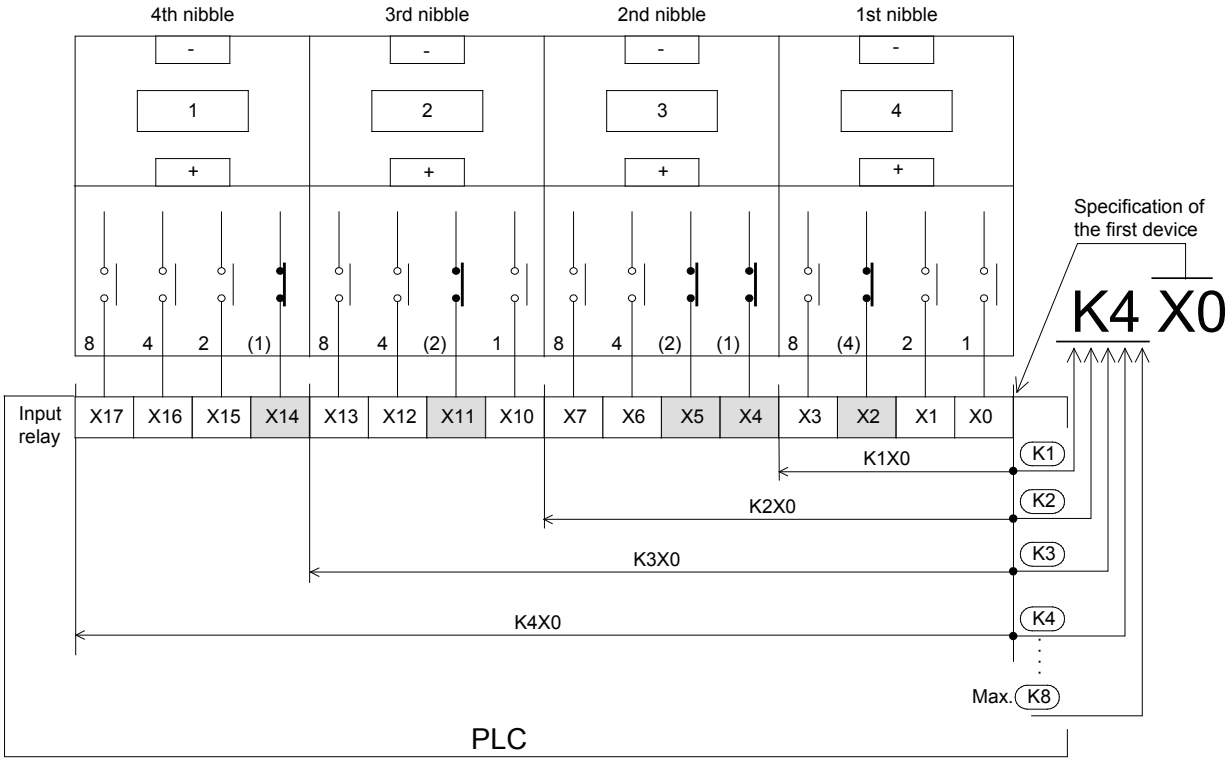
6.3.2 Operating bit devices as word devices (The method to specify the digits)

Bit devices are only used for the ON/OFF operation of input relays X, output relays Y, Internal relays M, state relays S and so no. However, bit devices can handle up to 32-bit values by combining 4 points into one nibble and using up to 8 nibbles together.

<<Example of K4X0>>



- Specifying the 4 digits of a digital switch with 4-bit module



K 4 X 0
(1) (2) (3) (4)

(1) Make sure to use K to specify the nibbles.
(2) Up to 4 nibbles (K1 to K4) can be specified for a 16-bit operation while up to 8 nibbles can be specified for a 32-bit operation.
(3) The device symbols X, Y, M, S can be specified. (Bit device)
(4) Always specify the device number of the least significant bit.

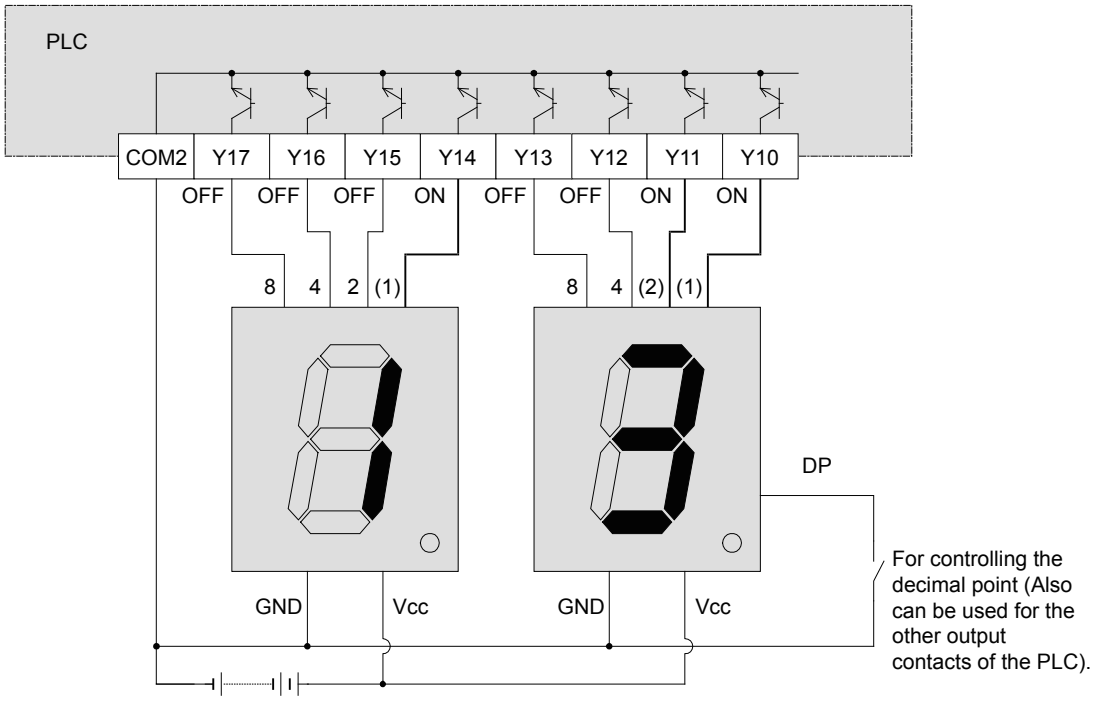
Device numbers can be specified arbitrarily. (Example: K4M13). However, generally, the least significant digit is recommended to be set to 0 such as K1X0, K2Y10, K3M20 and K8S130. (Multiples of 8 are ideal for M and S, but in order to avoid confusion, 0 is recommended to be used.)

Reference

7-Segment driver

To display numbers on a 7-segment display, a PLC outputs BCD of two digits from Y17 to Y10.
If the inputs 1, 2, 4, 8 of the 7-segment driver are driven by a PLC, 0 to 9 will be indicated according to the total value of the numbers input.

The digit is specified to "K2Y10" in the following example.



Chapter 7

TRANSFER INSTRUCTIONS FOR NUMERIC VALUES

Storing and reading numeric values...

As an extension of the previous chapter, this chapter helps to describe how instructions are used for storing and reading numeric data. The most basic of these instructions are the simple transfer instructions, which are used to read numeric values from storage sources S (transfer sources) and transfer them to storage destinations D (transfer destinations).

The concept of transferring data is easy...

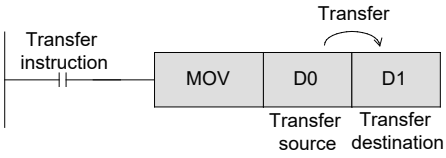
From simple to complicated applications, data transfer and data manipulation are regarded as common tasks.

PLC data transfer instructions provide an easy way to transfer a variety of data between devices. Also, by indirectly specifying transfer devices through indexing, data transfer instructions become much more flexible for moving ranges of data.

7.1 Data transfer instruction (MOV)

The data transfer instruction is the instruction to transfer bit data or numeric data from a transfer source to a transfer destination. It is a typical instruction that must be used for temporary data retention and storage.
The transfer instructions include instructions for simple data transfer and instructions (such as BCD and BIN) for converting data during transfer.

<<Operation outline>>

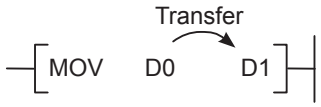


- Flow of data
  If the transfer instruction input is ON, the data in the transfer source is written to the transfer destination.
  In this situation, the data in the transfer source does not change.
  When the transfer instruction input is OFF, data is not transferred, and the transfer destination data does not change.

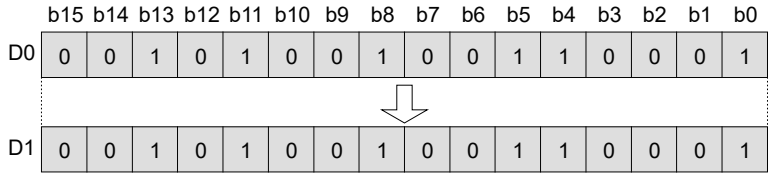
- The data is transferred in BIN.

(1) Transferring between two [word devices (16 bits)]

Example

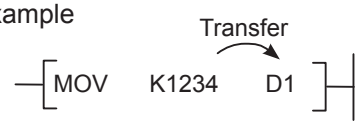


• Transfers data from D0 (16 bits) to D1 (16 bits).

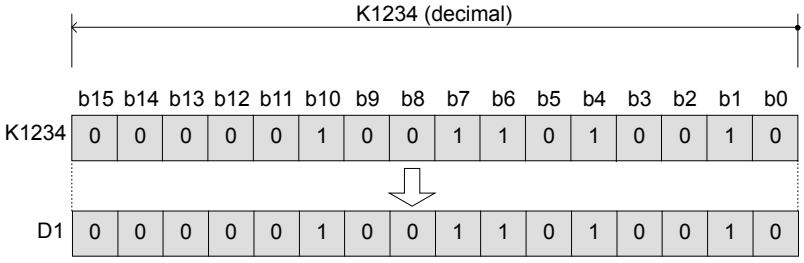


(2) Transferring a [constant K (decimal)] to a [word device]

Example

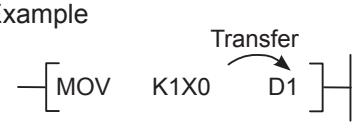


• The PLC treats K1234 as a BIN value and transfers it to D1 (16 bits).

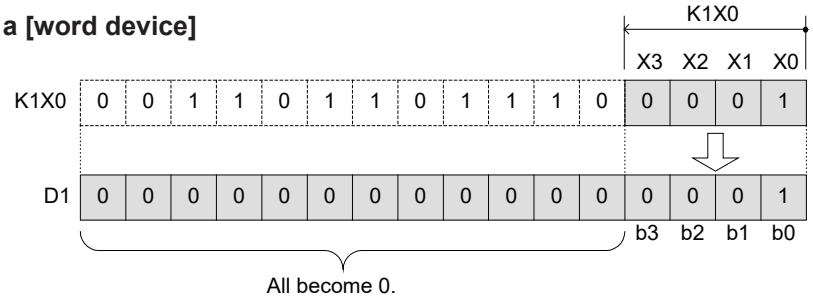


(3) Transferring from a [bit device] to a [word device]

Example

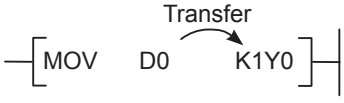


• When the transfer source does not include high-order bits (the data is not 16-bit data), the values in the high-order bits will not be transferred (e.g. the transfer source data is K1X0 to K3X0).

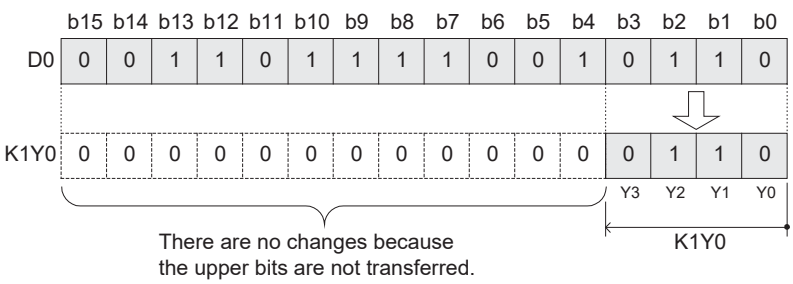


(4) Transferring from a [word device] to a [bit device]

Example

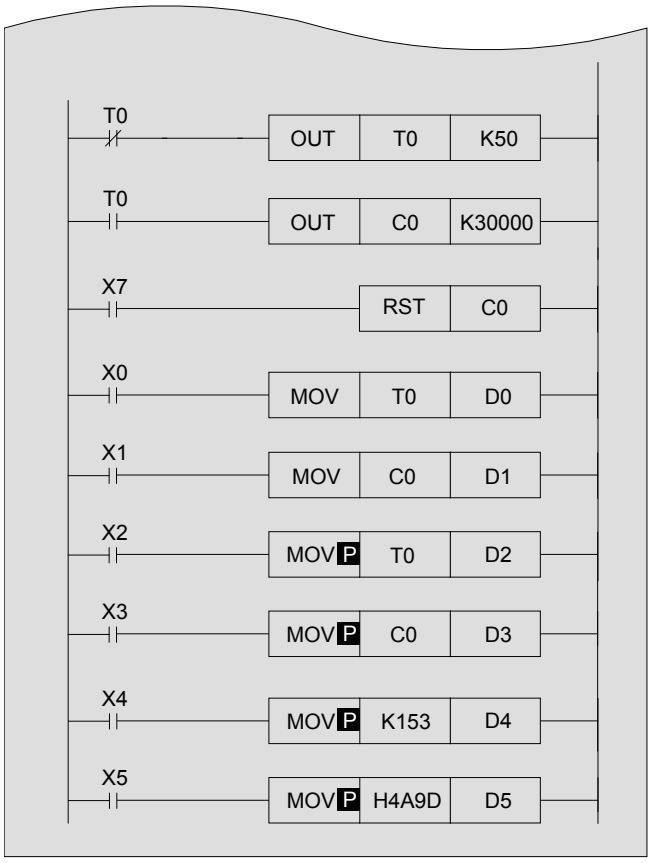


• When the transfer destination does not include high-order bits (the data is not 16-bit data), the values in the high-order bits will not be transferred (e.g. the transfer source data is K1Y0 to K3Y0).



<<Instruction operation>>

Let's check the operation of the MOV instruction.



- The current value of timer T0 repeatedly changes from 0 to 50.
- The current value of counter C0 increases every 5 seconds.

(1) Constantly transfer current value of timer T0 to D0

(2) Constantly transfer current value of counter C0 to D1

(3) Transfer current value of timer T0 to D2 only during MOV instruction drive

(4) Transfer current value of counter C0 to D3 only during MOV instruction drive

(5) Transfer constant K153 to D4 only during MOV instruction drive

(6) Transfer constant H4A9D to D5 only during MOV instruction drive

7

<<Operation check>>

Use GX Works3 to monitor the circuit.

(1) Turn "ON" X0 ⇨ ● The current value of timer T0 is always transferred to the data register D0 when X0 is ON.
If X0 is turned off, the data register D0 keeps the current value of the timer at the time when X0 is turned off.

(2) Turn "ON" X1 ⇨ ● The current value of counter C0 is always transferred to D1 when X1 is ON. If X1 is turned off, the data register D1 keeps the current value of the counter at the time when X1 is turned off.

(3) Turn "ON" X2 ⇨ ● The current value of timer T0 at the time when X2 is turned on is transferred to data register D2.

(4) Turn "ON" X3 ⇨ ● The current value of the counter at the time when X3 is turned on is transferred to data register D3

(5) Turn "ON" X4 ⇨ ● "153 (Decimal)" is directly transferred to D4. (Initial data setting)

(6) Turn "ON" X5 ⇨ ● The decimal "19101" is transferred to D5. (Initial data setting)
Select [Online]→[Monitor]→[Change current monitor value (Hexadecimal)]. "H4A9D" is displayed.
To return it to decimal, select [Change current monitor value (decimal)] from the same menu .

Point

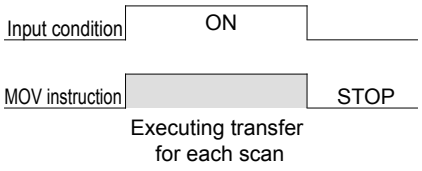
Continuous execution vs. Pulse execution

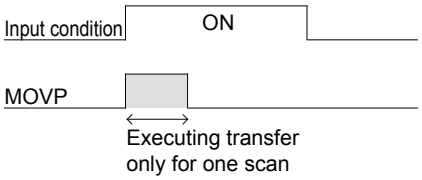
[Continuous execution instruction]

- For the continuous execution type instruction, when the drive input is ON, the instruction is executed and the transfer is performed for each scan.

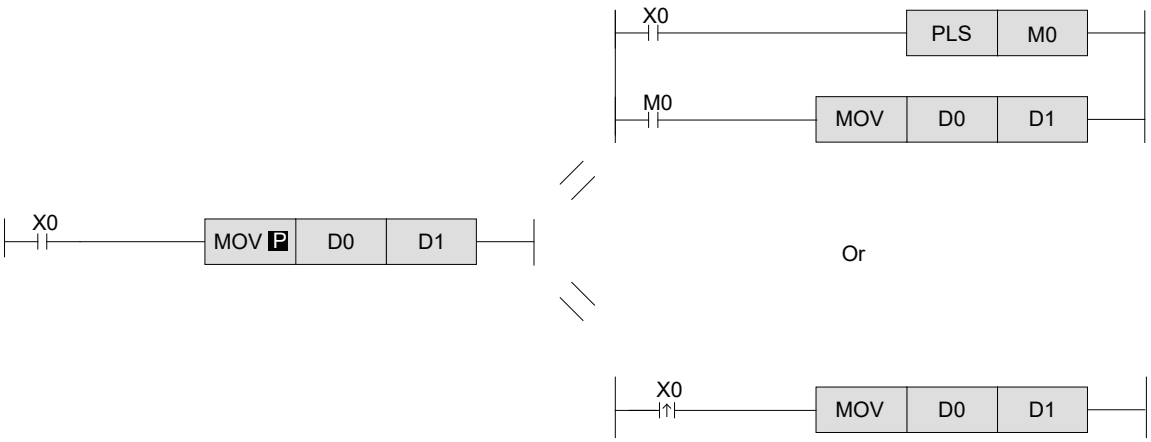
[Pulse execution instruction]

- P is added to the end of the pulse execution type instruction.
- For the pulse execution type, the instruction is executed only once when Input condition the input drive changes from OFF to ON.





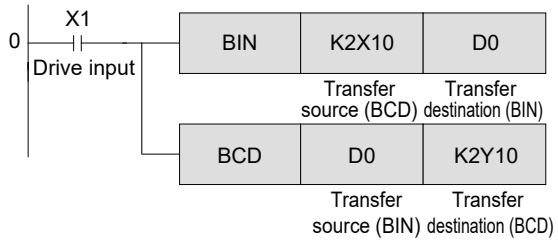
- Thus, the continuous execution instruction is used when data is constantly changing, while the pulse execution instruction is used for transfers that are needed only when the instruction is input turns on (When setting the initial value or the value at the specified time.)
- If the drive input is OFF, the MOV instruction is not executed and the transfer destination data does not change.
- The pulse execution operation can also be programmed as follows:



7.2 Conversion transfer instruction (BCD/BIN)

The operations of values in a PLC are all performed in BIN format. Thus, when inputting the digital switch values of a BCD to a PLC, use the BCD-to-BIN conversion transfer instruction. Also, to output to a 7-segment display module using BCD, it is necessary to use the BIN-to-BCD conversion transfer instruction.

<<Operation outline>>

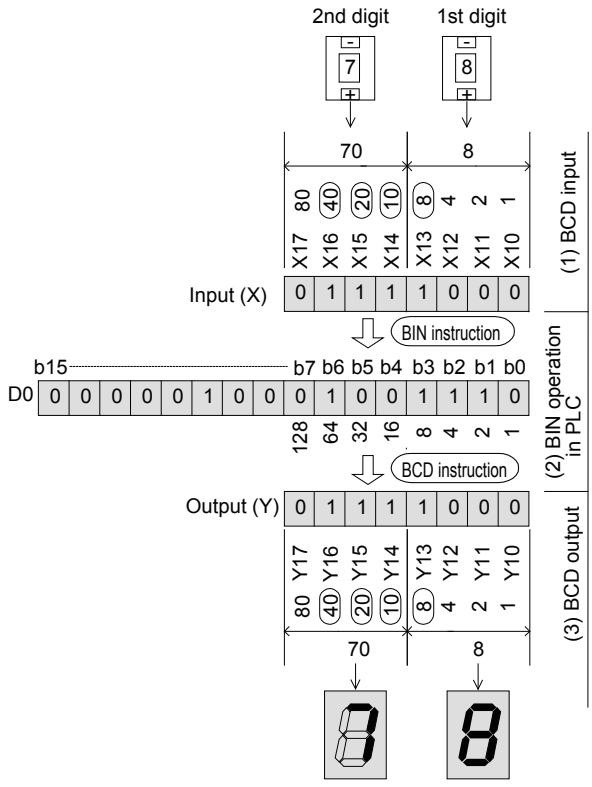


BIN instruction..... When the drive input is ON, the BCD value of the transfer source is converted to BIN and transferred.

BCD instruction..... When the drive input is ON, the BIN value of the transfer source is converted to BCD and transferred.

<<Instruction operation>>

Input the above program.



<<Operation check>>

(1) BCD codes are input to devices X10 to X13 (the first digit) and X14 to X17 (the second digit) according to the changes of the digital switch (BIN instruction).

(2) In the case of the left example, the input value "78" (Decimal: 1001110) is stored to D0.

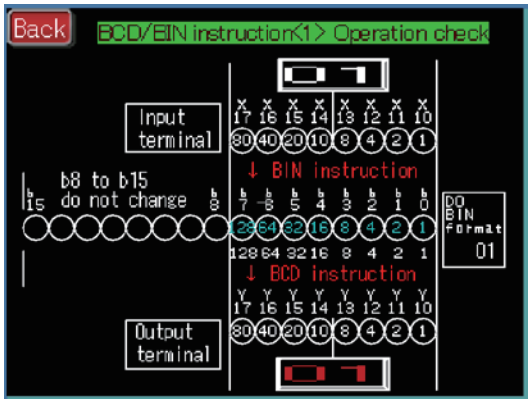
(3) BCD codes are output to devices Y10 to Y13 (the first digit) and Y14 to Y17 (the second digit) for operating the 7-segment display module (BCD instruction).

(Reference)

Input of digital SW	BCD value
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Training Machine Screen

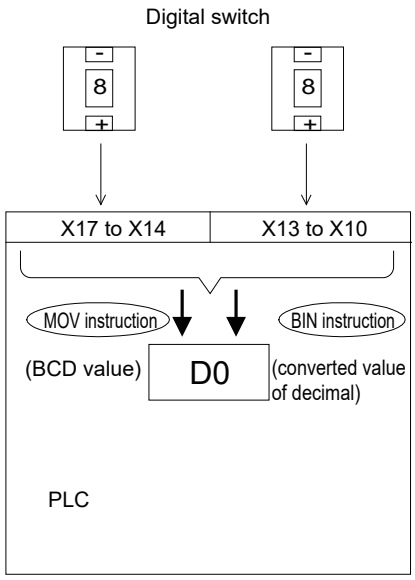
[Basics: MELSEC iQ-F Programming (GX Works3 Version)] → [BCD/BIN<1>]



It is possible to check the bit operations of the input relays (X), output relays (Y), and data registers in the GOT screen monitor.

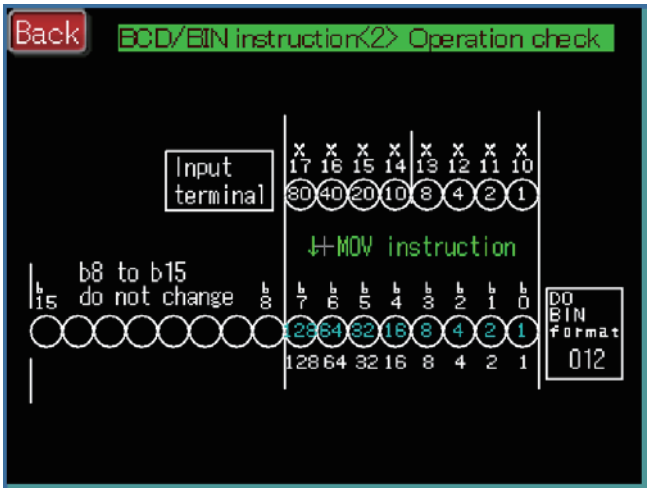
<<Comparing the operation>>

Let's compare the BIN/BCD instruction to the MOV instruction and check the operation.

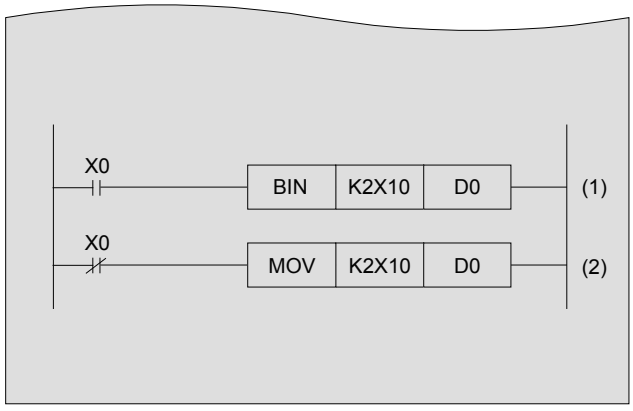


Training Machine Screen

[Basics: MELSEC iQ-F Programming (GX Works3 Version)] → [BCD/BIN<2>]



It is possible to check the bit operations of the input relays (X) and data registers (D0) in the GOT screen monitor.



Transfer BIN-converted X10 to X17 ON/OFF (BCD value) to D0.

Transfer X10 to X17 ON/OFF (BCD value) to D0.

<<Operation check>>

Use GX Works3 to monitor device D0.

(1) Turn "ON" X0 ⇨ The value of the digital switch (BCD input) is converted to a decimal value by "BIN instruction" and transferred to D0.

(2) Turn "OFF" X0 ⇨ The input is the same as (2) above, and the value is not converted to a decimal value because the instruction used is the MOV instruction, which directly transfers the value to D0.

Reference

Operation error and error step number

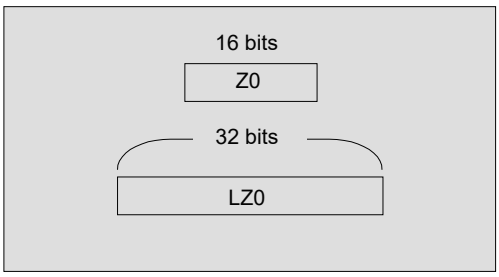
- When the value of the transfer source is not BCD, the BIN instruction will lead to an "operation error" and cannot be executed.
- The code and the step number of errors are stored to the following special Internal relay or special data register.
  The check can also be performed by selecting [Diagnostics] → [PLC diagnostics] from the menu.

SM0
SD0
SD88

If an operation error occurs, SM0 will turn on, and the error code of the operation error will be stored to SD0. The error step number will be stored to SD88.
If a new error occurs at another step, the error code of this instruction and the step number will be updated sequentially.
The error code is cleared when the PLC is reset, the power is turned ON, or SM50 is turned OFF and ON.

7.3 Indirectly specifying the transfer source and transfer destination

Devices in instructions can be specified directly (as described so far) or indirectly for data transfer operations.
The index registers V and Z can be used to specify devices indirectly.
The index register includes the index register (Z) and long index register (LZ).



- Z is 16-bit data register where numeric values can be written to and read from in the same manner as the general data register.
- LZ are 32-bit data registers where numeric values can be written to and read from.
- The index register (Z) and long index register (LZ) can use a total of 24 words.

- Changing constants and device numbers according to the values of Z, as shown on the left, is called indexing.

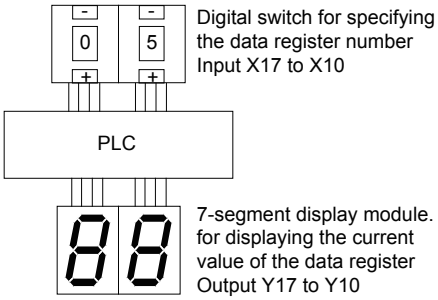
The devices that can be indexed

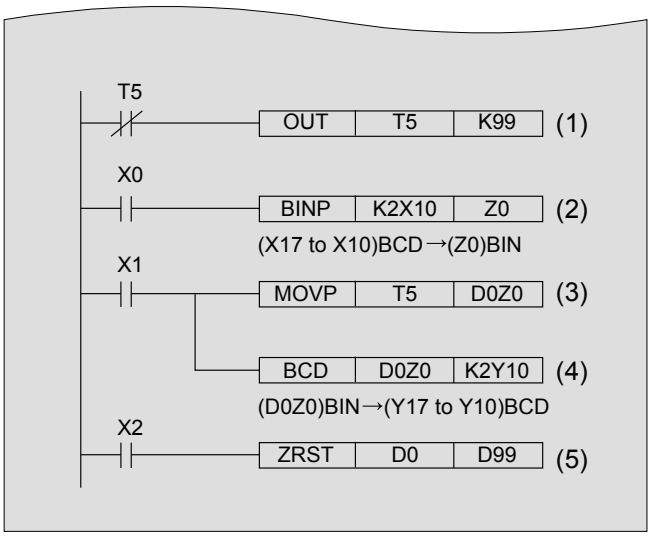
Z: X, Y, M, B, SB, F, S, L
T, ST, C, LC, D, R, W, SW, U□\G□
K, H, KnX, KnY, KnM, KnB,
KnSB, KnF, KnS, KnL
LZ: U□¥G□

The main devices that can be indexed with index registers are those listed on the left. The devices on the left are used in the instructions.
However, as shown below, Kn for nibble specification cannot be indexed.
(K4M0Z0 is valid, while K0Z0M0 is invalid)
P cannot be indexed as a label number of a jump destination.

<<Instruction operation>>

- Monitor the current values of the data registers (D0 to D99) that are specified by the index register Z.





<<Operation check>>

(1) Measure T5 timer

(2) The BCD value of the digital switch (X17 to X10) is transferred to the index register Z0 in BIN.

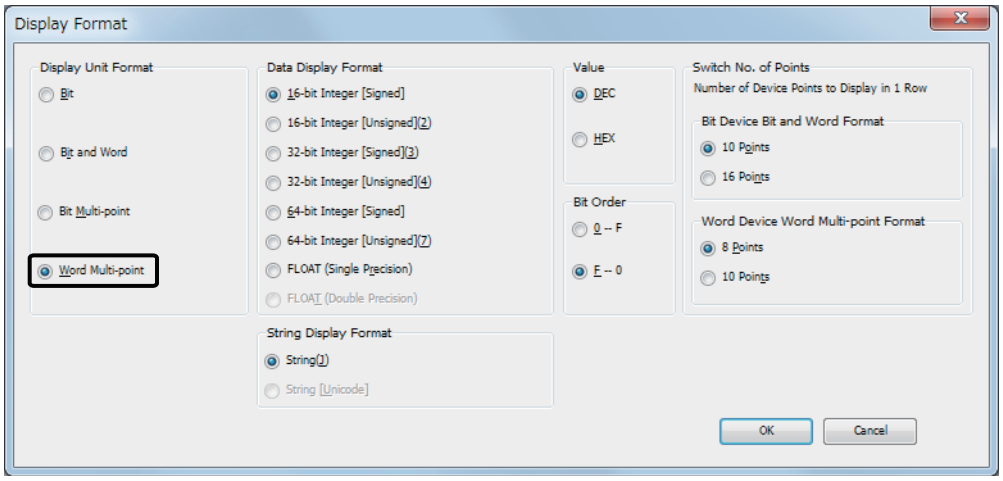
(3) Write current value of timer (T5) to the data register (D0 to D99).
*In the transfer destination D0Z0, the data register number changes between D0 and D99 according to the value of Z0.

(4) The current value in the data register whose number is indexed by index register Z0 is displayed in the 7-segment display module (Y17 to Y10).

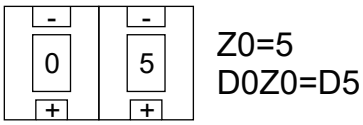
(5) Reset D0 to D99 value in a batch.

Check the current value of data registers (D0 to D99) with the Device Batch Monitor.
(Refer to "3.10.5 Device batch monitor" for the operation procedures.)

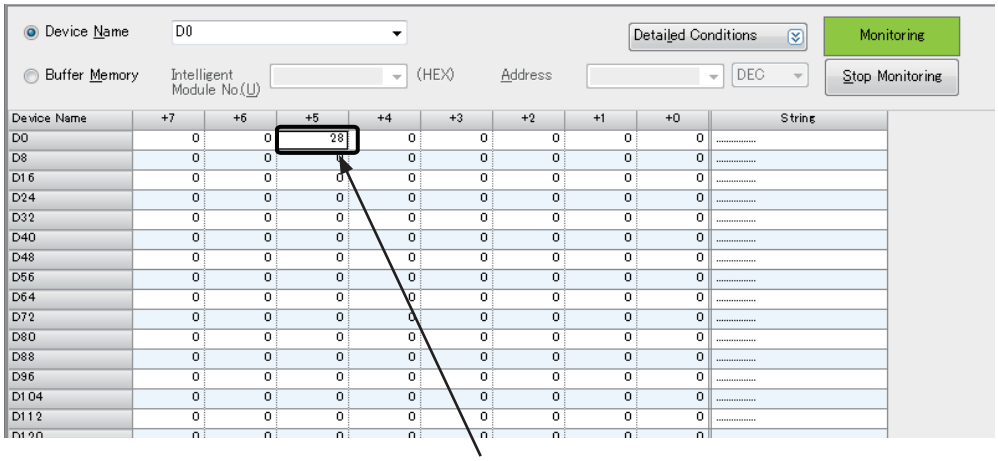
Select "Word Multi-point" for the display module type.



When digital switch (X17 to X10) is "5"



Z0=5
D0Z0=D5

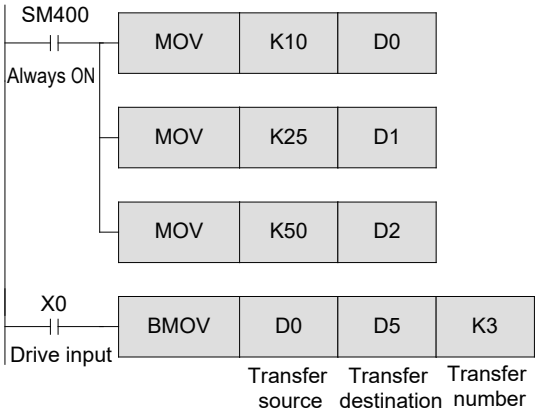


Current value of T5 is transferred to D5.

7.4 Other transfer instructions

7.4.1 Block Move (BMOV)

n pieces of data starting from the device specified in the transfer source are transferred in a batch to n devices starting from the one specified in the transfer destination.

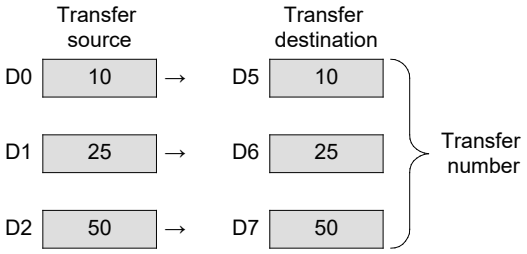


Transfer K10 to D0

Transfer K25 to D1

Transfer K50 to D2

3 data registers, D0 to D2, are transferred to D5 to D7.



- If the transfer number is larger than the device range, the data within the range will be transferred.
- In the case of bit devices for nibble specification, set the same number of nibbles between the transfer source and destination.

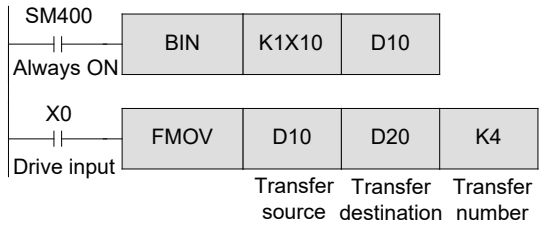
BMOV K2X0 K2Y0 K5

The number of nibble is the same!

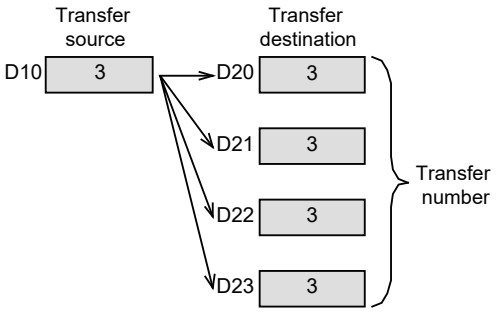
7

7.4.2 Fill move (FMOV)

The value of the device specified in the transfer source is transferred to n consecutive devices starting from the transfer destination device.
The n pieces of transferred data are the same.



The data set by DSW1 is converted to BIN and then transferred to D10.
The value of D10 is transferred to 4 points from D20 to D23.

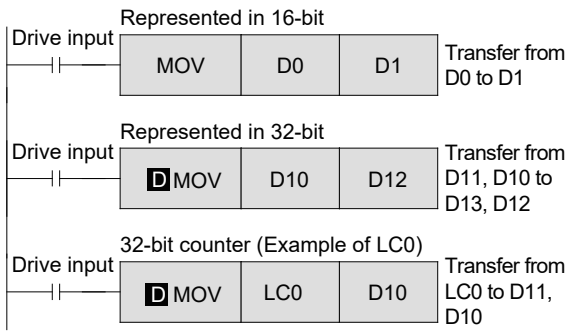


- If the transfer number is larger than the device range, the data within the range will be transferred.

Reference

Bit length for numeric values

- The instructions for numeric operations can be divided into 16-bit type and 32-bit type depending on the length of the numeric data.



- 32-bit instructions are represented with an added "D".
- The designated device can handle even or odd numbers, but the device with the number after the designated device is occupied.
- 1 point for a 32-bit counter occupies 32 bits. An operation error will occur if 16-bit instructions are used.

Chapter 8

THE COMPARISON INSTRUCTION FOR NUMERIC DATA

Comparing values in the PLC…

With basic value comparison instructions, two or more values can be compared to control a series of bit devices for the result.

For example, a changing numeric value can be instantaneously compared with a selected target value to determine which one is larger, which one is smaller, or whether they are equal.

Using zone comparison…

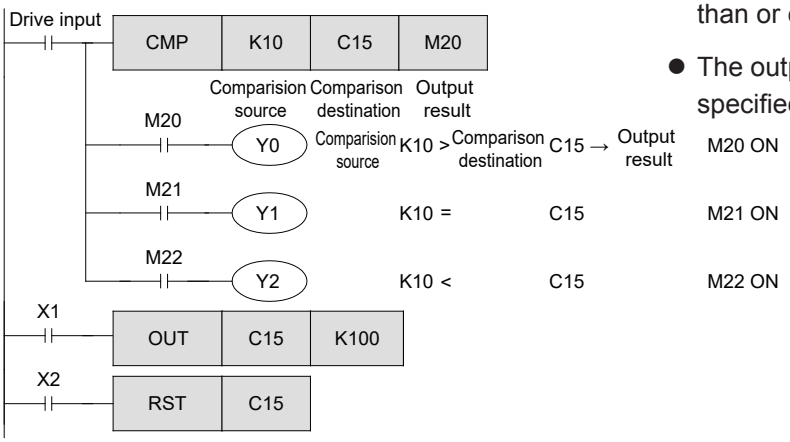
Zone comparison is beneficial for determining when a number is within a specified range of values. For example, depending on whether a device value is below, between, or above the range 100±2, a specified bit can be turned on to specify where the value exists in relation to the zone.

8.1 Data comparison instructions CMP, ZCP

The comparison instructions are used when comparing the current values stored in the data registers, the timers and the counters, or the values representing the combined relays of X, Y, M, or S, along with constants K. There are 1-point comparison and 2-point zone comparison methods for comparing data. Both methods produce three results to determine whether a value is "less than," "greater than," or "equal to" the comparison source(s).

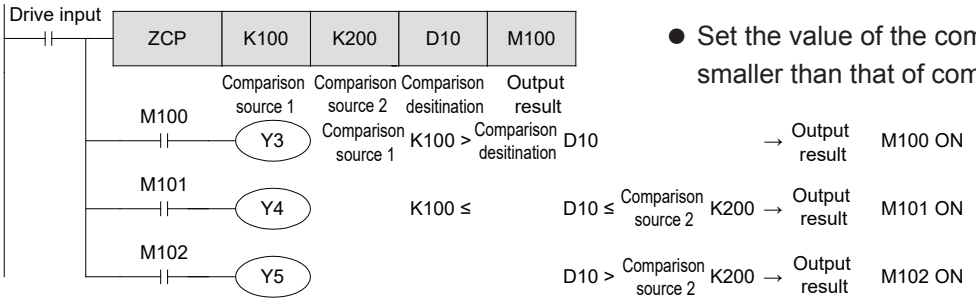
<<Operation outline>>

One point comparison



- If the drive input is turned on, a result is output to display whether the target is larger than, smaller than or equal to the comparison source.
- The output result occupies the first 3 points of the specified device.

Zone comparison



- The zone comparison instruction is an instruction that performs the same operation as the 1-point comparison except that the comparison source is a 2-point zone.
- Set the value of the comparison source 1 to a value smaller than that of comparison source 2.

Reference

Operation of output result

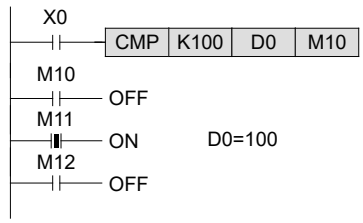
In the following cases, the output result for data comparison instructions will not change:

1. When the drive input is ON, but the description of the instruction is mistaken.

   Example
   - When the number of applicable compare device is indexed so that it exceeds the device range. (The operation error flag SM0 turns on in this case.)
   - When the value of the comparison source 1 is larger than the value of comparison source 2 for zone comparison.

2. The drive input is turned on once, and then turned off after the comparison instruction is executed and the compare output does not change.
   (The operation error flag SM0 is not turned on in this case.)

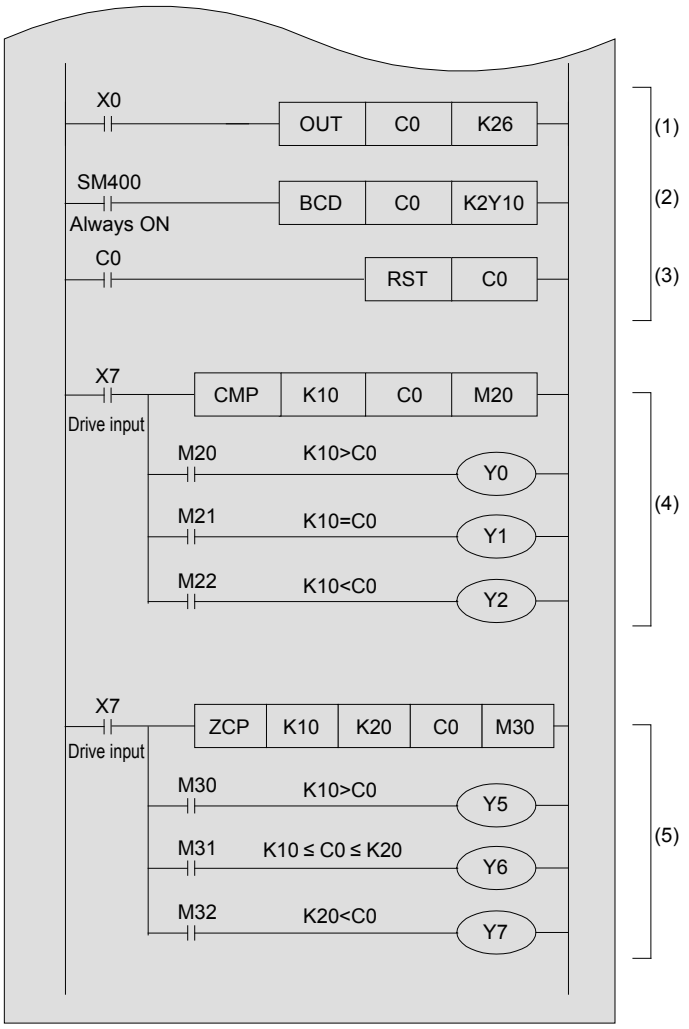
   Example



When the output result directly starts from the bus line as shown in the left diagram and the drive input X0 is turned off, the output M11 remains ON without regard to changes in D0.
It is recommended to serially connect the result input to the drive input as described in the above diagram, or to turn off the output destination devices using the RST instruction after the drive input is turned off.

<<Instruction operation>>

Use the count value of the counter to check the operations of the CMP instruction and the ZCP instruction.



<<Operation check>>

(1) The current value of the counter C0 increases by "1" when the input of X0 is "ON"

(2) The current value of the counter C0 is indicated in the 7-segment display module.

(3) When the current value of the counter reaches 26, reset the counter and return it to the initial status.

(4) Value comparison

Current value of counter	11 or more	10	0 to 9
	Y2: ON (M22: ON)	Y1: ON (M21: ON)	Y0: ON (M20: ON)

(5) Zone comparison

Current value of counter	21 or more	10 to 20	0 to 9
	Y7: ON (M32: ON)	Y6: ON (M31: ON)	Y5: ON (M30: ON)

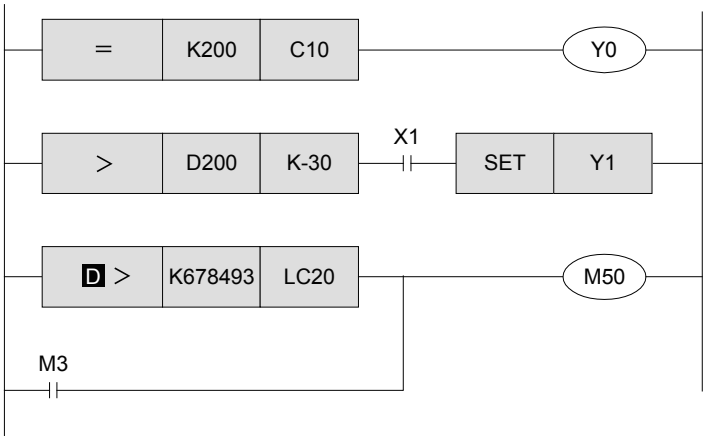
8.2 Contact comparison instructions (LD * , AND * , OR *)

Contact comparison instructions can use a comparison result as the contact information in a circuit.
Therefore, the expression method is intuitive and easy-to-understand, and the sequence can be calculated according to the comparison results.

<<Operation outline>>

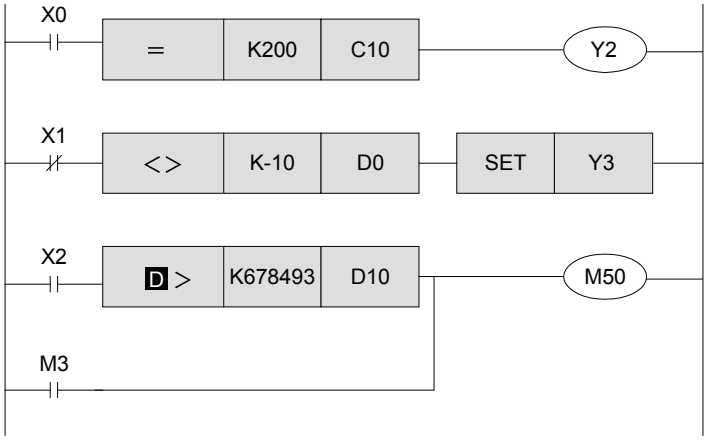
- The contact compare instructions are divided into 3 types depending on their placement within a program: LD contacts (connected from the bus line), AND contacts (connected to others serially), and OR contacts (connected to others in parallel).
- When inputting with the GX Works3 tool button, "=, >, <, <>, ≤, ≥" can be input after [ ] F8

LD * * : =,>,<,<>,≤,≥



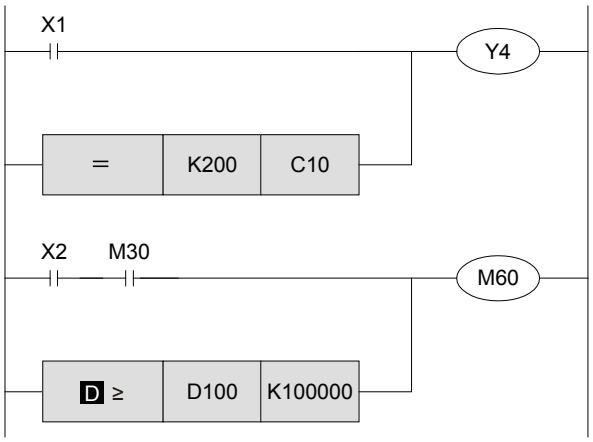
- Y0 is turned on when the current value of the counter C10 is 200.
- Y1 is turned on when the value of D200 is -29 or more and X1 is ON.
- M50 is turned on when the value of the long counter LC20 is smaller than 678493 or M3 is ON.
- 32-bit instructions are used for long counters (LC0 onwards).

AND * * : =,>,<,<>,≤,≥



- Y2 is turned on when X0 is ON and the current value of the data register C10 is 200.
- Y3 is SET when X1 is OFF and the value of the data register D0 is not -10.
- When X2 is ON, M50 is ON in the case that the value of D11 and D10 is smaller than 678493, or M3 is turned on.
- 32-bit instructions are used for 32-bit data.

OR * * : =,>,<,<>,≤,≥

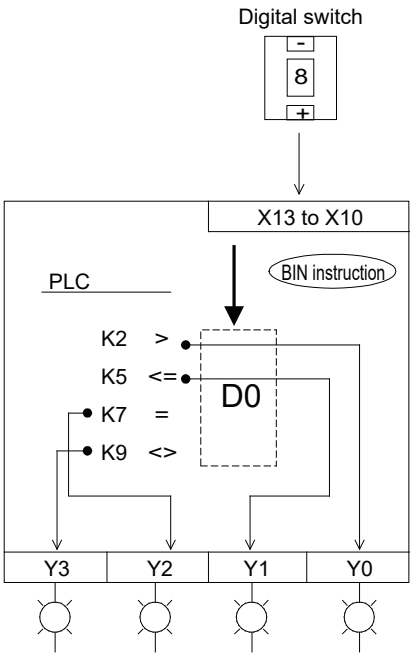


- Y4 is turned on when X1 is turned on or the current value of the counter C10 is 200.

- M60 is turned on when both X2 and M30 are turned on, or the values of the data register D101 and D100 is 100000 or more.
- 32-bit instructions are used for 32-bit data.

<<Instruction operation>>

Compare the input value from a digital switch to a value set in advance and then turn on an output.

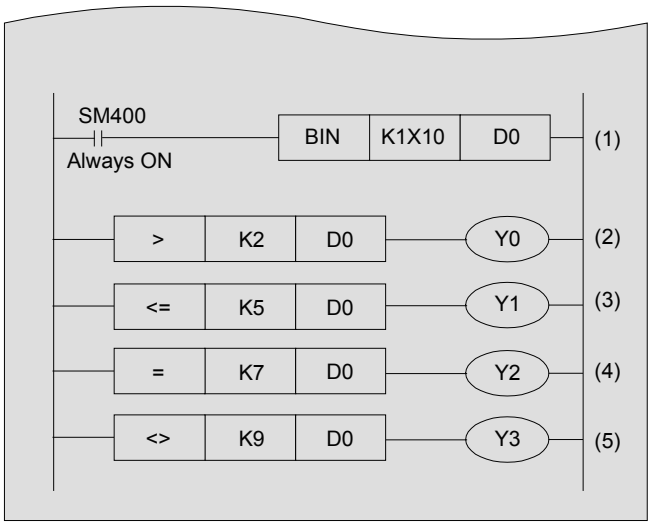


<<Operation check>>

(1) The input value from the digital switch is transferred to D0.
(2) to (5)The values of Y0 to Y3 change according to the values of the digital switches as show below.

●: Output ON　　−: Output OFF

DSW value	Output			
	Y3	Y2	Y1	Y0
0	●	−	−	●
1	●	−	−	●
2	●	−	−	−
3	●	−	−	−
4	●	−	−	−
5	●	−	●	−
6	●	−	●	−
7	●	●	●	−
8	●	−	●	−
9	−	−	●	−



BIN convert the X10 to X13 ON/OFF (BCD) and transfer to D0

Y0 is ON when D0 value is less than K2

Y1 is ON when D0 value is K5 or more

Y2 is ON when D0 value is K7

Y3 is ON when D0 value is other than K9

Chapter 9
ARITHMETIC OPERATION

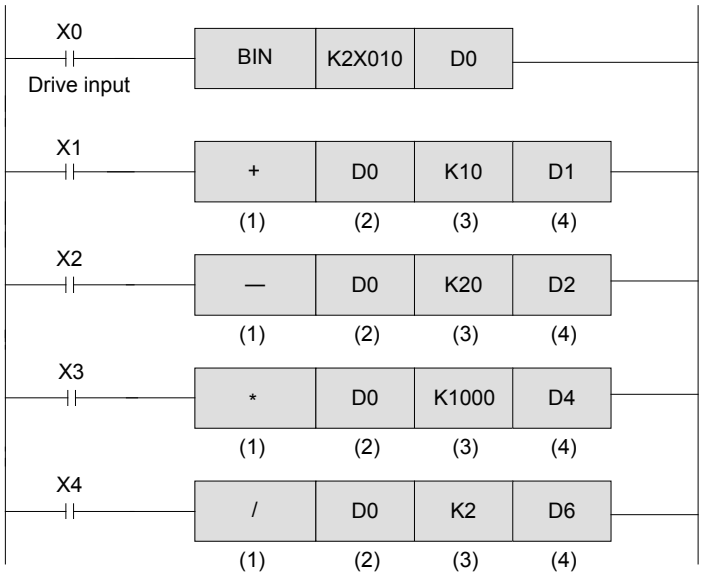
Arithmetic operation (+, −, *, /)

For basic arithmetic operations such as multiplication, addition and subtraction, specific applied instructions are available.
This chapter describes the basic instructions needed for binary arithmetic control.

9.1 Arithmetic operation instructions +, −, *, /

PLC arithmetic instructions are used to perform operations of addition, subtraction, multiplication and division.

<<Operation outline>>



The BCD values X10 to X17 are converted to BIN, and then transferred to D0.

D0 + K10 = D1
←The following changes occur according to the instructions.

D0 - K20 = D2

D0 × K1000 = D5, D4

D0 ÷ K2 = D6…(D7) Remainder

(1) Instructions
- + : BIN ADDITION
- - : BIN SUBTRACTION
- * : BIN MULTIPLICATION
- / : BIN DIVISION

(2) Specify the augend, minuend, multiplicand or dividend.

(3) Specify the addend, subtrahend, multiplier or divisor.

(4) Specify the storage destination for the operation result.

For the 16-bit instruction +, -, *, /

Addition (D0) + (K10) = (D1)	Sum
Subtraction (D0) - (K20) = (D2)	Difference
Multiplication (D0) × (K1000) = (D5, D4)	Product (The result is 32 bits.)
Division (D0) ÷ (K2) =(D6)	Quotient…(D7) Remainder

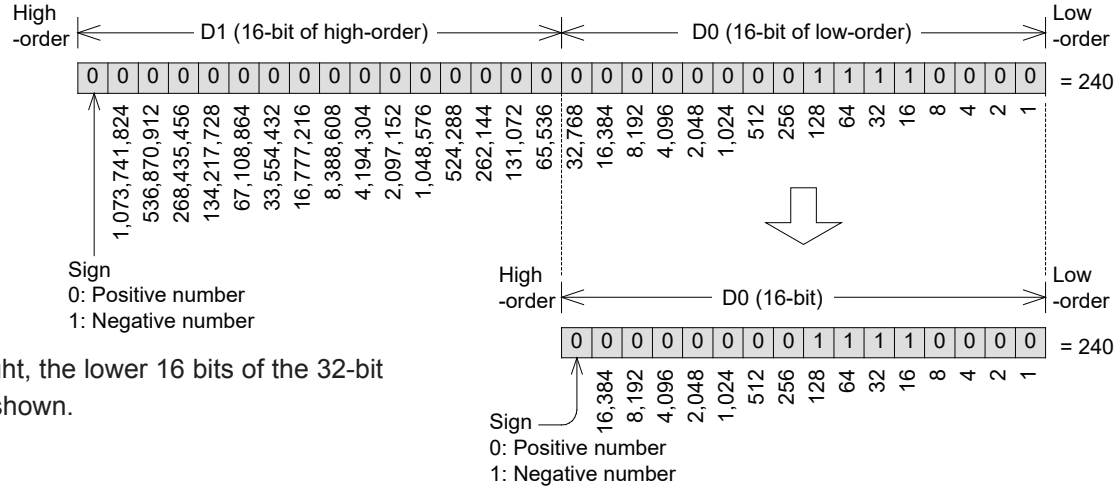
For the 32-bit instruction **D** +, **D** -, **D** *, **D** /

Addition (D1, D0) + (K10) = (D2, D1)	Sum
Subtraction (D1, D0) - (K20) = (D3, D2)	Difference
Multiplication (D1, D0) × (K5) = (D7, D6, D5, D4)	Product (The result is 64 bits.)
Division (D1, D0) ÷ (K2) = (D7, D6)	Quotient…(D9, D8) Remainder

- The operation result is 32-bit for multiplication (64 bits for the **D** instruction), and for division, a register is occupied by the remainder. Thus, it is necessary to avoid storage destination conflicts for operation results of multiplication and division instructions.
- For the 32-bit multiplication instruction (**D** *), the operation result becomes 64 bits.
  In this situation, note that there are no applied instructions or peripheral equipment for the 64-bit data. For example, when division is also used, it is recommended to perform the division first in order to make the data for the multiplication as small as possible.
- If the INT/FLT instruction is used, floating point operation for values below decimal-point is also available.

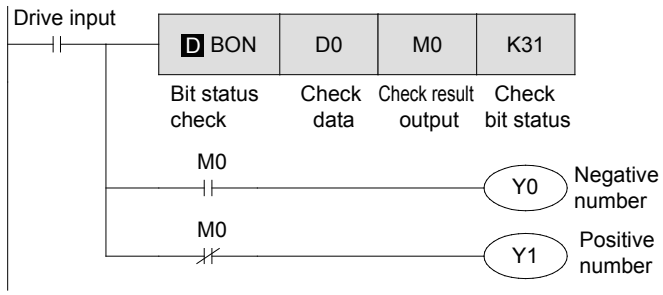
The operation result of the multiplication instruction (product) is represented as 32-bit data with the 16-bit operation, while 64-bit data is required for the result of the 32-bit operation.

For example, if the product is K240, the 16-bit operation is shown below.
In this situation, only the upper 16 bits can be used since the upper 16 bits are 0.

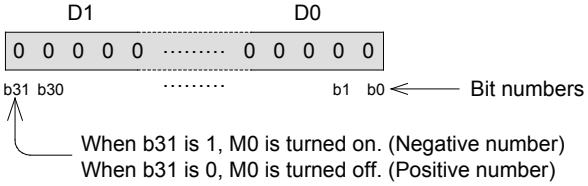


On the right, the lower 16 bits of the 32-bit data are shown.

When the operation result is within 16 bits (32 bits), the data of the lower words can be used as the operation data for the next time.
Moreover, it is convenient to use the bit status check instruction (BON) to check the positive and negative numbers.



When the 31st bit (highest bit) of the 32-bit data D1 and D0 is ON, M0 is turned on.



MEMO

Chapter 10

HIGH SPEED PROCESSING FUNCTIONS AND COMMANDS

Capturing high speed input signals to a PLC…

Normally, in order to securely capture an input signal to a PLC, a signal width of at least "scan cycle + filter time (10 ms)" is required.
In addition to normal inputs, however, the micro PLCs have built-in functions to capture high-speed signals, which makes it possible to process high frequency signals.

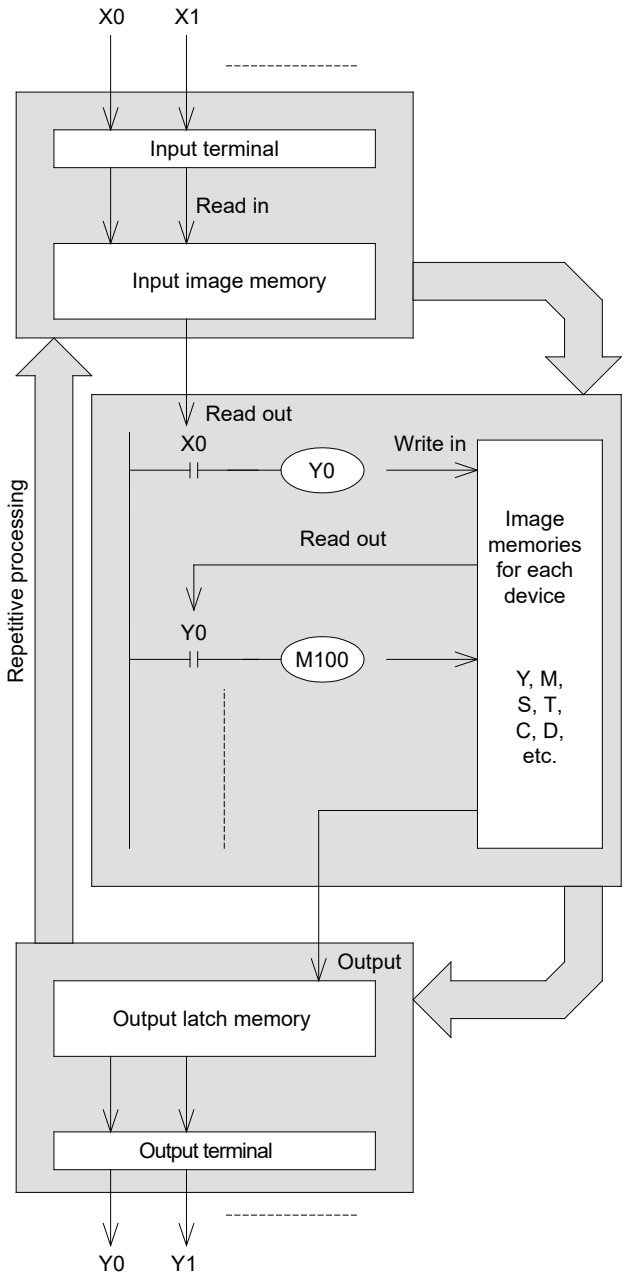
Using high speed processing…

"Input interrupts" and "high speed counters" are two examples of control mechanisms that use high speed processing procedures.
With micro PLCs, a variety of built-in high speed functions are available for capturing and controlling high speed signals.

10.1 Concept of high speed processing

The micro PLC repeats a sequence of "input processing" → "program processing" → "output processing". This is called "batch refresh", and is commonly used with the PLC.
This series of processing actions (one scan cycle) only needs approximately 10 ms, enough to normally control a sequence. But 10 ms is not enough to execute processing immediately after an input is turned on or to capture an input signal shorter than one scan cycle of the PLC.
This section describes the processing methods that are not subject to the influence of the scan cycle of the program.

Outline of refresh mode



- **Input operation**
  Prior to execution of a program, the PLC reads all ON/OFF statuses of input terminals into the input image memory.
  If an input changes its status during execution of the program, the input image memory does not change the contents at this time. The change will be read in the next input process cycle.

- **Process of program**
  The PLC reads the ON/OFF statuses of required devices from the input image memory or other device's memory, in accordance with the contents of instructions stored in the program memory. Hence the image memory of each device can sequentially change its content in accordance with the progress of the program.

- **Output operation**
  When all instructions have been executed, the PLC transfers the ON/OFF statuses of outputs Y to the output latch memory, which are the physical outputs.

Types of high speed processing supported with FX5U CPU module

For interrupt processing

Interruption, including input interrupts and timer interrupts, can be executed during the program operation.

- **Input interrupt :** Eight points of X0 to X17 can be used. The specified program is executed when an input point is turned on or off.
- **Timer interrupt:** The specified program is executed at the set time.
  For time specification, four points from 1 to 60,000 ms can be used.

For high speed counter

High speed counters count how many times the inputs X0 to X17 are turned ON or OFF.
FX5U CPU module can count short pulses within 200 kHz.

For pulse catch

When the input relay (X0 to X17) changes from OFF→ON, the input device stays ON during the next scan.

Instructions that execute high speed processes

In addition to the interrupt counters and high speed counters, the following instructions are executed at high speed.

MTR	Input Matrix	
HSCS	Compare and Set	for high speed counter
HSCR	Compare and reset	
HSZ	High Speed Counter Zone Compare	
SPD	Speed Detection	
PLSY	Pulse Y Output	
PWM	Pulse Width Modulation	

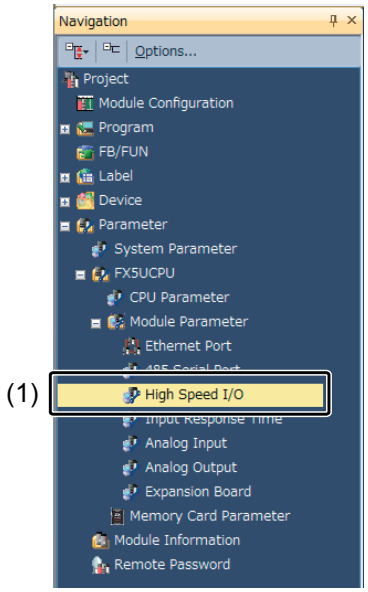
<<Additional note>> For input designation

Inputs X0 to X17 can be used for high speed processing. If they are set for high speed processing, they cannot be used for other processing instructions.
For example, when X0 is used for an interrupt instruction, it cannot be used for a high speed counter. If X0 is programmed for both purposes, an error will occur.

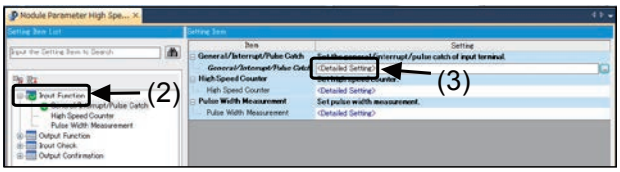
10.2 Using input interrupts

X0 to X17 are available for input interrupts. Up to eight points can be used.
Parameter settings and an interrupt program are required to use the input interrupt.

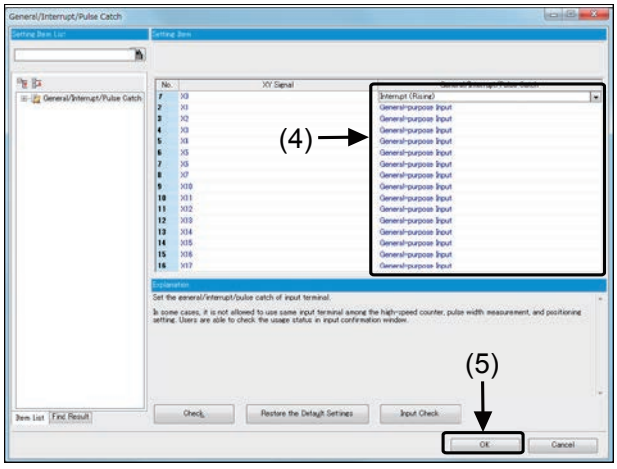
Settings for the input interrupt are explained here.



(1) Double-click [Parameter] → [FX5UCPU] → [Module Parameter] → [High Speed I/O] on the navigation window.



(2) Select [Input Function].
(3) Double-click [General/Interrupt/Pulse Catch] → [Detailed Setting].

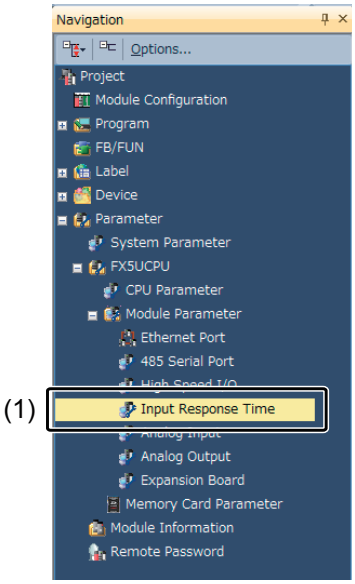


(4) Select the input interrupt settings from the following.

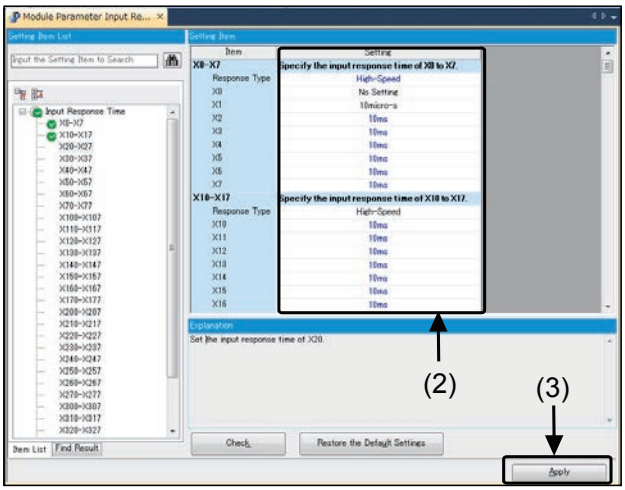
- Interrupt (Rising)
  The interrupt program is executed when OFF→ON of the input signal is detected.
- Interrupt (falling edge)
  The interrupt program is executed when ON→OFF of the input signal is detected.
- Interrupt (rising edge + falling edge)
  The interrupt program is executed when OFF→ON or ON→OFF of the input signal is detected.
- Interrupt (rising edge) + pulse catch
  The interrupt program is executed when OFF→ON of the input signal is detected.
  The pulse catch function can also be used.

(5) After setting, click [OK].

The settings for the input response time are explained next.



(1) Double-click [Parameter] → [FX5UCPU] → [Module Parameter] → [Input Response Time] on the navigation window.



(2) Set the input response time. (Default value: 10 ms)

- Response type
  High speed: 1-point module
  Standard: 8-point module

- Input response time
  10 µs, 50 µs, 0.1 ms, 0.4 ms, 0.6 ms, 1 ms, 5 ms, 10 ms, 20 ms, 70 ms

(3) After setting, click [Apply].

10

Reference

The actual input response time will be the value to which the hardware filter value has been added.
The FX5U CPU module hardware filter delay time is shown below.

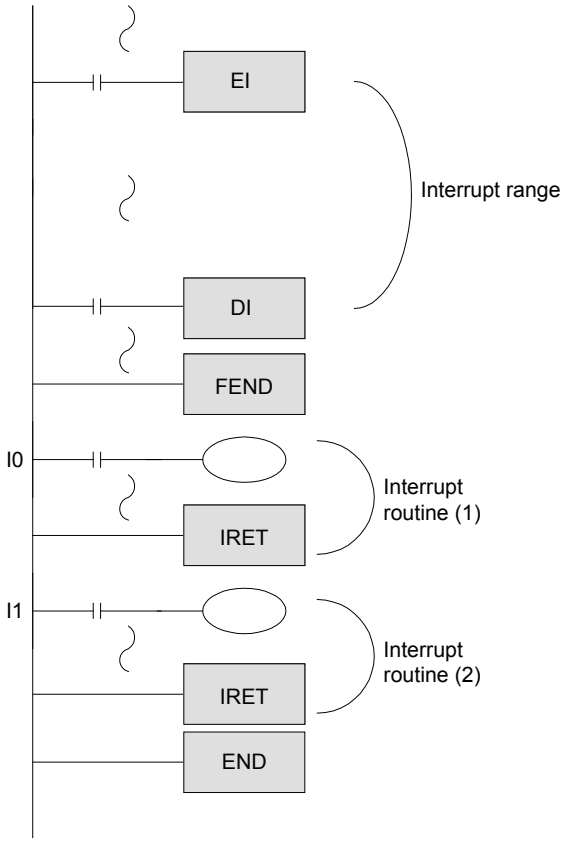
Input No.		Hardware filter value	
FX5U-32M□	FX5U-64M□, FX5U-80M□	When ON	When OFF
X0 to X5	X0 to X7	2.5 µs	2.5 µs
X6 to X17	X10 to X17	30 µs	50 µs
—	X20 and following	50 µs	150 µs

<<Operation outline>>

Pointer

I □

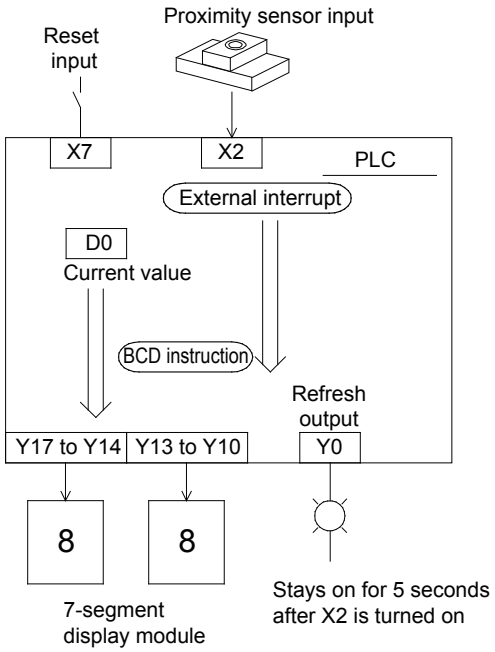
Input X0 to X7 : No. 0 to 7
Input X10 to X15 : No. 8 to 15



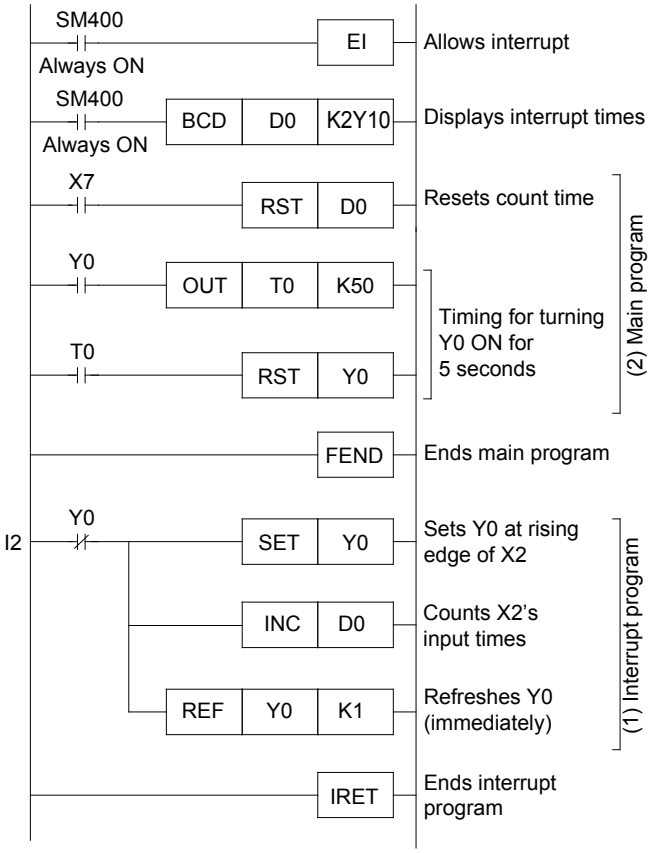
- The PLC is usually in the interrupt prohibit state. If X0 or X1 is turned on when the program is executed between the EI and the DE instructions, the interrupt routine (1) or (2) is executed. The program returns to the main program by the IRET instruction.
- Basically, an interrupt program cannot be executed while another interrupt program is in progress. However, it can be accepted if multiple interrupt is “enabled” with the parameters.
- Use a routine timer in the subroutine or interrupt routine.
  The parameters must be set to use the routine timer.
- If the signal width is input response time + hardware filter, the input interrupt operation can be executed.
- If multiple interrupts occur in succession, the interrupt with the smaller order of priority will have the higher priority. If the interrupt order of priority is the same, the interrupts will be executed in the interrupt order of priority.
- Interrupts that occur after the DI instruction are processed after the EI instruction is executed.

<<Instruction operation>>

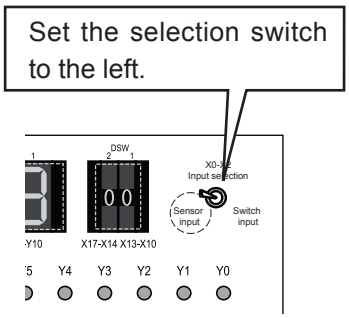
Confirm the actions of the external interrupt signal of X2.
Operate with the image that an object is passing by the proximity sensor connected to X2 at a high speed.



- Parameter setting <using X2>
  - High speed I/O : Interrupt (rising edge)
  - Input response time : No setting
- Program example



[Preparing the training machine]



<<Operation check>>

(1) • Y0 is turned on when X2 is turned on, and the data is output immediately by the REF instruction. X2 is used for high speed input. The input signal needs to stay on for only 2.5 μs to be recognized.
  • In this example program, the PLC does not recognize a signal from X2 while Y0 is ON (5 seconds).
  • The number of times X2 is turned on is counted by the INC instruction (increment a value by 1), and stored in the data register D0.

(2) • The current value of D0 is reset when X7 is turned on.
  • The timer T0 times out and is reset after Y0 stays on for 5 seconds.

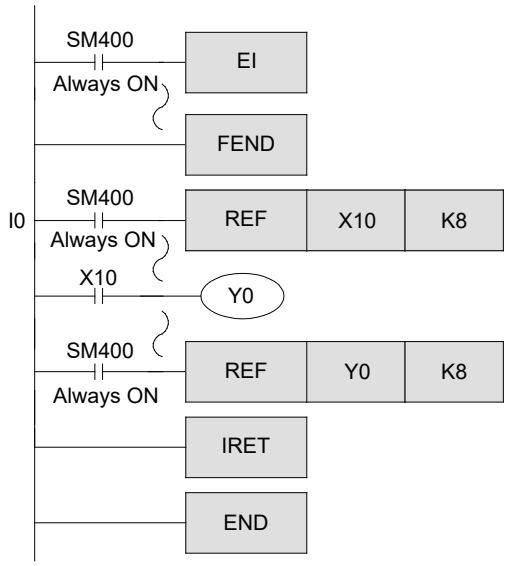
Reference

- When input interrupt is used, short input signals can be accepted without missing any. In this program example, the operation to accurately count the workpieces that pass over the proximity sensor at a high speed is realized.
- The INC instruction is an instruction that adds "1" to the specified device.
  If the INC instruction of the continuous execution type is used, 1 is added every operation cycle. To avoid this, generally, the "INCP" instruction of the pulse execution type is used.
  On the left program, 1 is added by an instruction only in response to an interrupt input. Therefore the program runs properly with the "INC" instruction of the continuous execution type.

Point

Using an interrupt instruction with an I/O refresh instruction

- In the program between the pointer and the IRET instruction, input processing is executed by way of "ON/OFF" general input processing. The result of the interrupt program is not executed until the whole program is finished.
- For this reason, the result of the interrupt program may not be prompt even though it is an interrupt program.
- However, if the I/O refresh instruction is used, the latest I/O information can be used for the operation.



- End of the main program
- I0 is the X0 interrupt pointer.
- The values of X10 to X17 are stored.
- X10 is refreshed and then processed.
- The result of the operation is output. (Y0 to Y7)
- End of the interrupt

Reference

HIOEN instruction

This command controls the start and stop of the high speed input/output function.

HIOEN	(1)	(2)	(3)

(1) Set the number of the function to be started and stopped.

Function number	Function name
K0	High speed counter
K10	Pulse density
K20	High speed comparison table
K30	Multi-point output high speed comparison table
K40	Pulse width measurement
K50	PWM

(2) Set the CH No. bit for starting the function.

For function No. K0, the start of the high-speed counter can be individually controlled for each parameter high-speed counter CH.

Bit position															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—								CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1

(3) Set the CH No. bit for stopping the function.

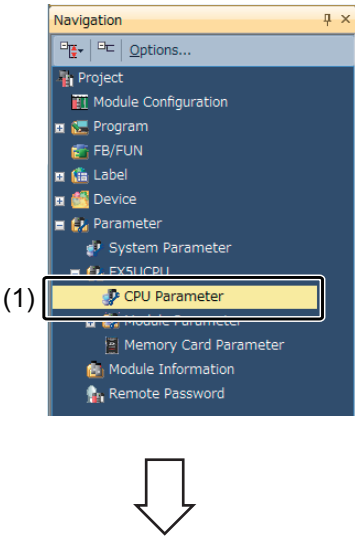
For function No. K0, the stop of the high-speed counter can be individually controlled for each parameter high-speed counter CH.

Bit position															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—								CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1

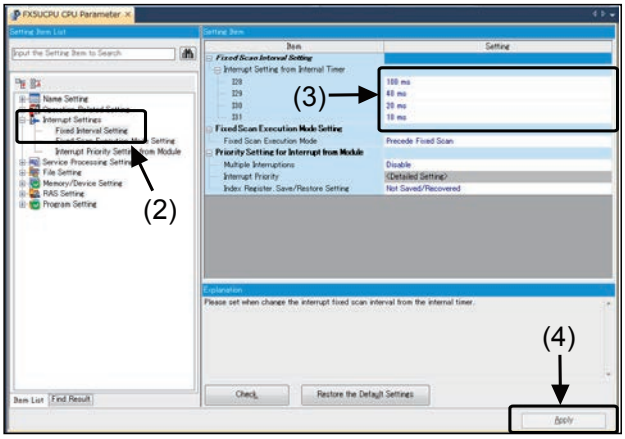
- If a value that turns the same CH ON is set for the start and stop, the stop operation will have the priority.

10.3 Using a timer interrupt program

The four points I28 to I31 can be used for the interrupt timer.
The timer interrupt cycle is set with the parameters.



(1) Double-click [Parameter] → [FX5UCPU] → [CPU Parameter] on the navigation window.



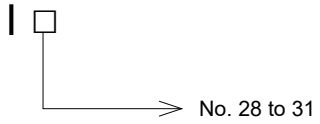
(2) Click [Interrupt Settings] → [Fixed Interval Setting].

(3) Set the timer interrupt set cycle interval.
(Setting range: 1 ms to 60000 ms)

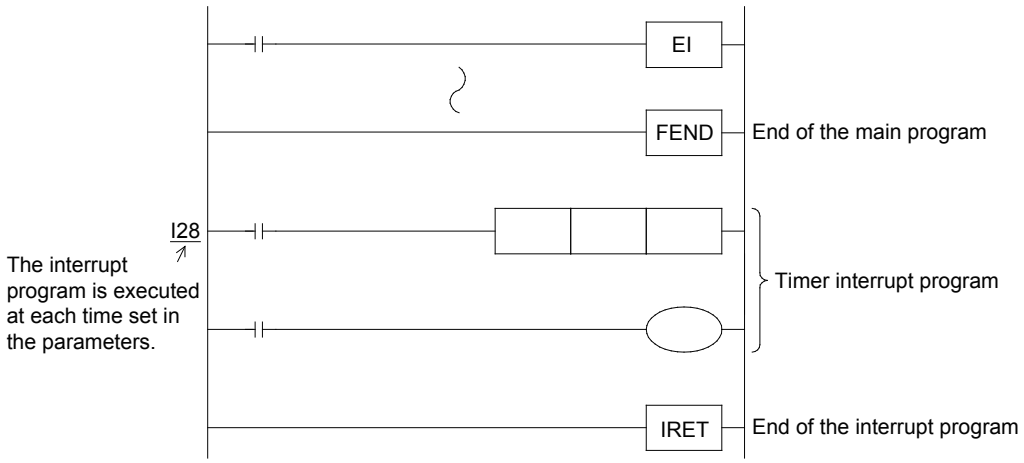
(4) After setting, click [Apply].

<<Operation outline>>

Pointer



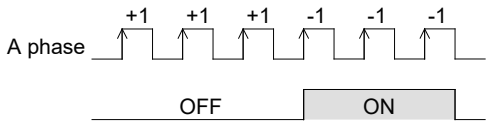
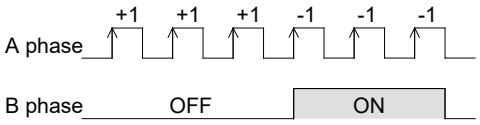
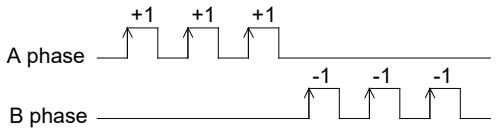
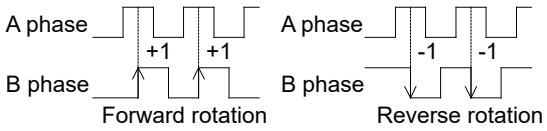
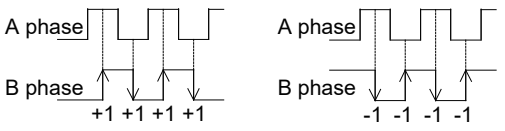
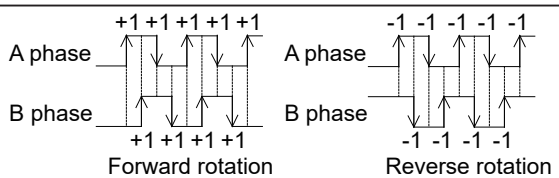
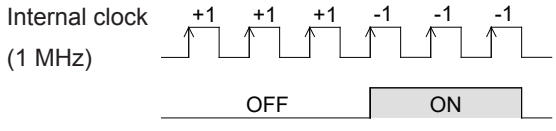
The interrupt program starts when the time specified by the pointer has arrived.
Other conditions for timer interrupt programs are the same as those for the input interrupt programs. The time specified in the parameter is used as the timer value, so there is no need to program another timer circuit for interrupt.
The following instructions use several operation cycles to execute a series of actions: RAMP, SEGL, PLSY
Using these instructions may take a long time to complete all of the actions, or may not even complete all of the actions successfully due to time fluctuation. Use a timer interrupt program for such cases.



10.4 Using high speed counters

10.4.1 Types of high speed counters

A high speed counter counts how many times an input device turns on or off in interrupt processing. It does not depend on the operation cycle.
As shown in the table on the next page, the input of the high speed counter is specified by the device number of the high speed counter. The input can be X0 to X17.
There are four types of high speed counters (including the internal clock counter).

<table>
<tr><th colspan="2"></th><th>Input signals form</th><th>Counting directions</th></tr>
<tr><td rowspan="2">1-phase
1-counting
input</td><td>S/W</td><td> 
</td><td>The counting direction (up or down) is specified with SM4580 to SM4587.
ON: Down-counting
OFF: Up-counting</td></tr>
<tr><td>H/W</td><td> 
</td><td>Down-counting or up-counting are designated by the ON or OFF state of the B-phase. The counting direction of the counter can be monitored with SM4564 to SM4571.
ON: Down-counting
OFF: Up-counting</td></tr>
<tr><td colspan="2">1-phase
2-counting input</td><td> 
</td><td>This counter counts up or down as shown on the left.
The counting direction of the counter can be monitored with SM4564 to SM4571.
ON: Down-counting
OFF: Up-counting</td></tr>
<tr><td rowspan="3">2-phase
2-counting
input</td><td>1-edge
count</td><td> 
</td><td rowspan="3">This counter counts up or down automatically according to the input status in A- and B-phases.
The counting direction of the counter can be monitored with SM4564 to SM4571.
ON: Down-counting
OFF: Up-counting</td></tr>
<tr><td>2-edge
count</td><td> 
</td></tr>
<tr><td>4-edge
count</td><td> 
</td></tr>
<tr><td colspan="2">Internal clock</td><td> 
</td><td>The counting direction (up or down) can be monitored with SM4580 to SM4587.
ON: Down-counting
OFF: Up-counting</td></tr>
</table>

10.4.2 High speed counters and input terminal numbers

Assign the high speed counter input device with the parameter.
The corresponding assignment is determined according to the type of counter set for each channel with the parameter.
For example, if CH1 is set to 1-phase 1-input (H/W), a 1-phase 1-input counter with X0 for the A-phase input and X1 for the B-phase input is created. If 1-phase 1-input (H/W) is used with CH1, 1-phase 1-input (S/W) cannot be used with CH2.

CH	High speed counter type	X0	X1	X2	X3	X4	X5	X6	X7	X10	X11	X12	X13	X14	X15	X16	X17
CH1	1-phase 1-input (S/W)	A								P	E						
	1-phase 1-input (H/W)	A	B							P	E						
	1-phase 2-input	A	B							P	E						
	2-phase 2-input	A	B							P	E						
CH2	1-phase 1-input (S/W)		A									P	E				
	1-phase 1-input (H/W)			A	B							P	E				
	1-phase 2-input			A	B							P	E				
	2-phase 2-input			A	B							P	E				
CH3	1-phase 1-input (S/W)			A										P	E		
	1-phase 1-input (H/W)					A	B							P	E		
	1-phase 2-input					A	B							P	E		
	2-phase 2-input					A	B							P	E		
CH4	1-phase 1-input (S/W)				A											P	E
	1-phase 1-input (H/W)							A	B							P	E
	1-phase 2-input							A	B							P	E
	2-phase 2-input							A	B							P	E
CH5	1-phase 1-input (S/W)					A				P	E						
	1-phase 1-input (H/W)									A	B	P	E				
	1-phase 2-input									A	B	P	E				
	2-phase 2-input									A	B	P	E				
CH6	1-phase 1-input (S/W)						A					P	E				
	1-phase 1-input (H/W)											A	B	P	E		
	1-phase 2-input											A	B	P	E		
	2-phase 2-input											A	B	P	E		
CH7	1-phase 1-input (S/W)							A						P	E		
	1-phase 1-input (H/W)													A	B	P	E
	1-phase 2-input													A	B	P	E
	2-phase 2-input													A	B	P	E
CH8	1-phase 1-input (S/W)								A							P	E
	1-phase 1-input (H/W)															A	B
	1-phase 2-input															A	B
	2-phase 2-input															A	B

A: A phase input
B: B phase input (Directional input used for 1-phase 1-input (H/W))
P: External preset input
E: External enable input

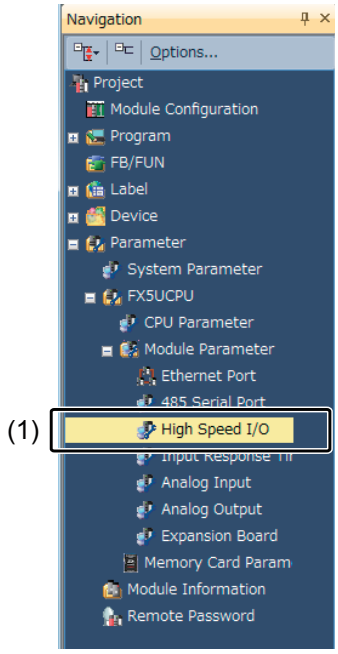
Reference

The approximate maximum frequency that can be counted for each high speed counter type is shown below.

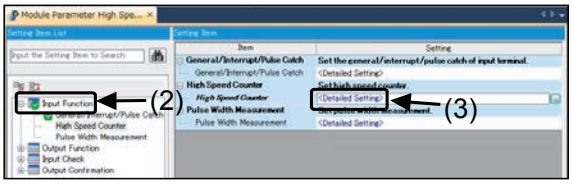
1-phase 1-input (S/H)	Maximum frequency
1-phase 1-input counter (S/W)	200 kHz
1-phase 1-input counter (H/W)	200 kHz
1-phase 2-input counter	200 kHz
2-phase 2-input counter [1-edge count]	200 kHz
2-phase 2-input counter [2-edge count]	100 kHz
2-phase 2-input counter [4-edge count]	50 kHz
Internal clock	1 MHz (fixed)

10.4.3 High speed counter actions

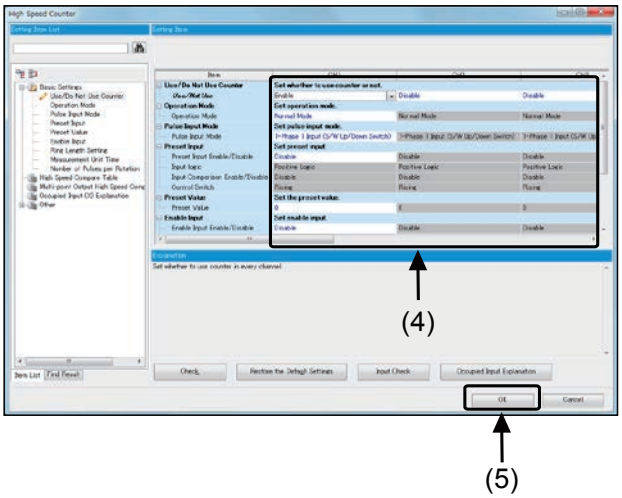
<<Outline of parameters>>



(1) Double-click [Parameter] → [FX5UCPU] → [Module Parameters] → [High Speed I/O] on the navigation window.



(2) Select [Input Function].
(3) Double-click [High speed Counter] → [Detail Settings].

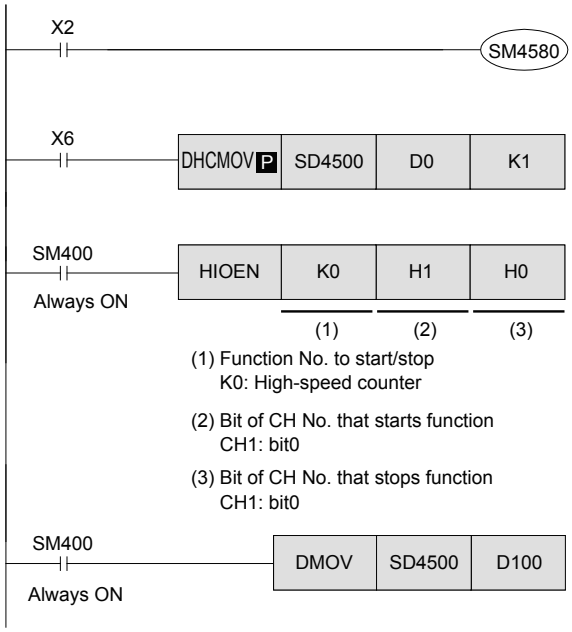


(4) Set each item for the high speed counter.
(5) After setting, click [OK].

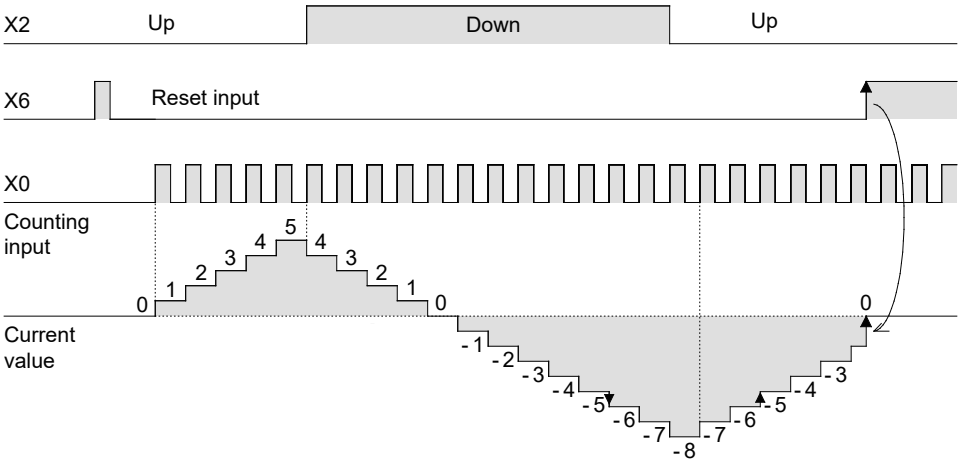
- When using the high speed counter, also set the input response time.
  Refer to the previous section “10.2 Using input interrupts” for details on the settings.

<<Operation outline>>

- Example of high speed counter CH1: 1-phase 1-input (software)



- Select the up counter or the down counter.
  ON: Down counter, OFF: Up counter
- When X6 changes from OFF to ON, the counter's current value (SD4501, SD4500) is transferred to D1, D2, and the counter's current value (SD4501, SD4500) is set to 0.
- The high speed counter (CH1) measurement starts.
- The high speed counter (CH1) counter current value (SD4501, SD4500) is always transferred to D101, D100.



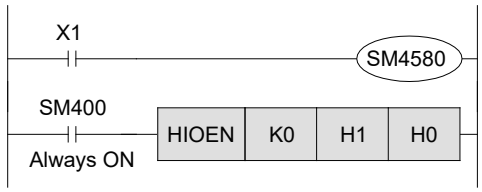
- The current value changes regardless of the output contact ON/OFF status. If the counter counts up from 2,147,483,647, the value is changed to -2,147,483,648, and if the counter counts down from -2,147,483,648, the resulting value is 2,147,483,647.
  (This counting operation is called a ring counter.)
- The current value of the counter is latched.

10.4.4 1-phase high speed counter operation

- The above counters are latched binary counters that use 32 bits.
- By using the high speed counting inputs, interrupt instructions are executed to count independently from the sequence operation. (The PLC provides instructions that are used to execute interrupt processes for output or comparison.)
- The parameters explained previously must be set to use the high speed counter.
- If external enable input or external preset input is valid, the count can be started and reset with the interrupt input.

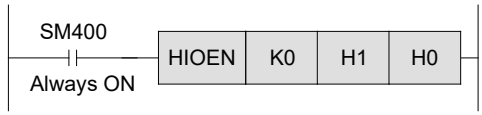
The operation when using high speed counter CH1 with preset and enable validated is explained here.
Refer to the previous section “High speed counter and input terminal numbers” for details on the preset and enable device assignments.
(The preset and enable input logic is “Positive Logic”.)

1-phase 1-input (S/W)



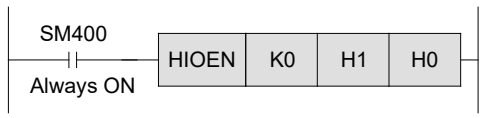
- Changes to down or up with the X1 ON/OFF state.
- When X11 is ON, input X0 OFF to ON is counted.
- When X10 turns ON, the counter’s current value is preset.

1-phase 1-input (H/W)



- Changes to down or up with the X1 ON/OFF state.
- When X11 is ON, input X0 OFF to ON is counted.
- When X10 turns ON, the counter’s current value is preset.

1-phase 2-input



- Counting starts immediately when X11 turns ON. Counts up when X0 changes from OFF to ON, and counts down when X1 changes from OFF to ON.
- When X10 turns ON, the counter’s current value is preset.

<<Additional note>>

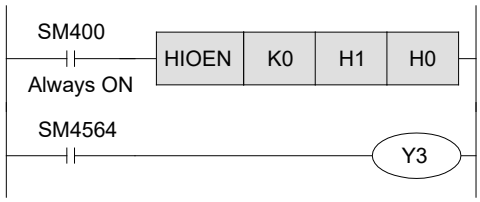
Note that a malfunction occurs in counters due to switch chattering when high speed counters are activated with simulation switches.

10.4.5 2-phase high speed counter operation

- The above counters are latched binary counters that use 32 bits.
- By using the high speed counting inputs, interrupt instructions are executed to count independently from the sequence operation. (The PLC provides instructions that are used to execute interrupt processes for output or comparison.)
- The parameters explained previously must be set to use the high speed counter.
- If external enable input or external preset input is valid, the count can be started and reset with the interrupt input.
- While an A-phase input is ON, this counter counts up when the B-phase input is changed from OFF to ON, and counts down when the B-phase input is changed from ON to OFF. Additionally, the counting direction (up or down) can be recognized by monitoring the ON/OFF status of SM4564 to SM4571.

The operation when using high speed counter CH1 with preset and enable validated is explained here.
Refer to the previous section “High speed counter and input terminal numbers” for details on the preset and enable device assignments.
(The preset and enable input logic is “Positive Logic”.)

2-phase 2-input



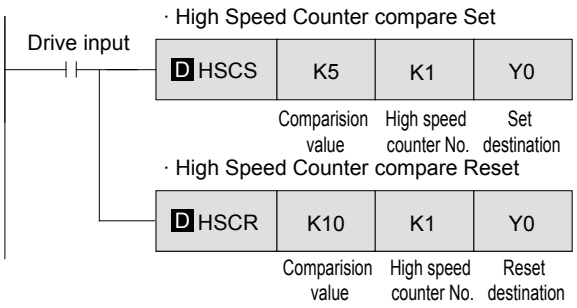
- When X11 is ON, the input X0 (A-phase) and X1 (B-phase) operation is counted.
- When X10 turns ON, the counter’s current value is preset.
- Y3 is turned on (count-down) or off (count-up) according to the counting direction.

10.4.6 Instructions and their actions for high speed counters

The previous section described basic ways on how to use the high speed counters. When the current value reaches the setting value for a counter, the following instructions are used to output signals immediately.
Similar to the high speed counters, the applied instructions are executed independently from the sequence operation. Thus, outputs can be used without any operation delays.

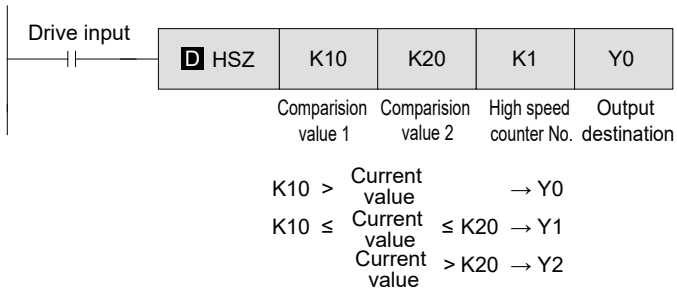
<<Operation outline>>

[High Speed Counter Set/Reset instructions]



- When the current value reaches the comparison value, interrupt processing is used to operate output signals.
- High speed counters use 32 bits. Thus, 32-bit instructions must be used with D added.

[High Speed Counter Zone Compare]

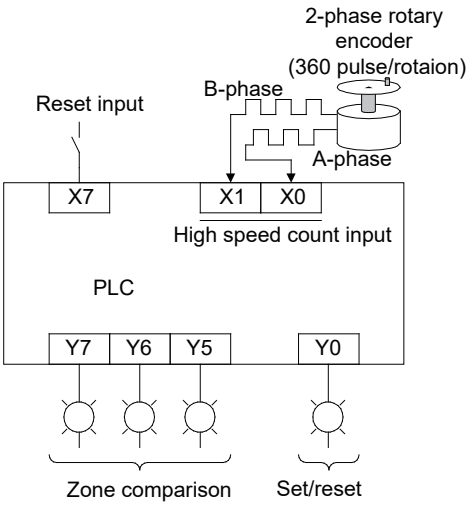


- This instruction is a High Speed Counter Zone Compare instruction.
- Three output points are controlled according to the current value of the high speed counter.

- Simultaneous execution of the above instructions must be limited to 32 instructions or less (More than 32 of the instructions can be programmed if they are not simultaneously executed.)

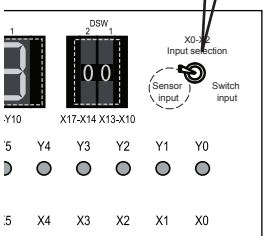
<<Instruction operation>>

Using the training machine, create a sequence program with a high speed counter to count input signals from a 2-phase rotary encoder (high speed output device) and to operate the following outputs accordingly.



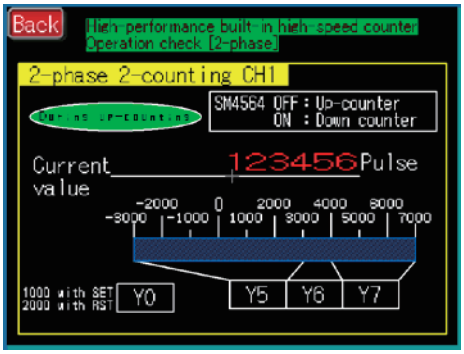
[Preparing the training machine]

Set the selection switch to the left.



Training Machine screen

[Basics: MELSEC iQ-F Programming (GX Works3 Version)] → [High speed counter]

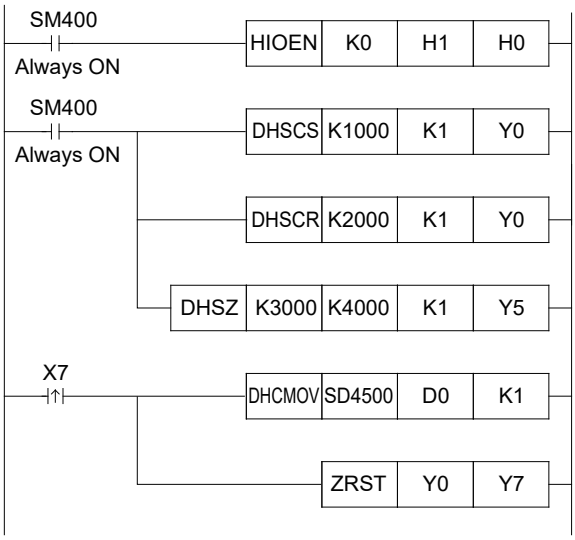


- Parameter setting

  [High speed I/O] – [High speed counter]
  CH1 enabled/disabled : Enabled
  Operation mode : Normal mode
  Pulse input mode : 2-phase 1-edge count
  Preset input : Invalid
  Enable input : Invalid

  [Input response time]
  X0: No setting
  X1: No setting

- Program example



Always count high speed counter CH1

Y0 is set when counter's current value reaches 1000

Y0 is reset when counter's current value reaches 2000

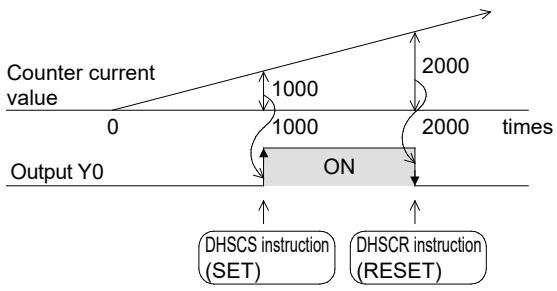
Y6 is ON while counter's current value is between 3000 and 4000
If less than 3000, Y5 turns ON
If 4000 is exceeded, Y7 turns ON

Transmit counter's current value (SD4501, SD4500) to D1, D0, and set SD4501, SD4500 to 0

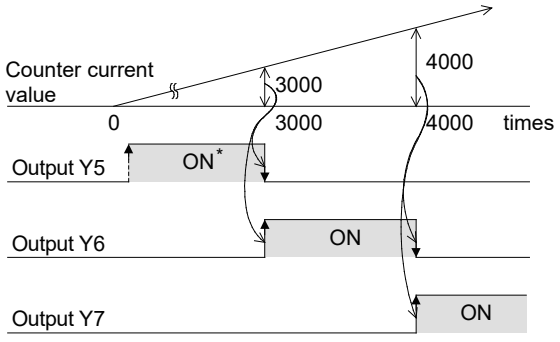
Batch reset Y0 to Y7

<<Operation check>>

[Count value is 0 to 2000]



[Count value is 3000 to 4000]



* Y5 turns on when the counter counts up from 0 to 1.

MEMO

Chapter 11
ANALOG FUNCTION

In this chapter...

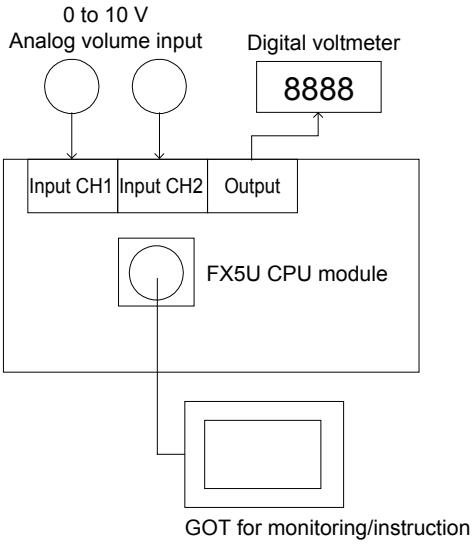
The analog input and output will be explained with setting examples using the FX5U CPU module built-in analog input/output function.

11.1 Using the built-in analog function

<<Instruction operation>>

The FX5U CPU module has two analog input points (voltage) and one analog output point (voltage).
In this chapter, we will use the built-in analog input/output function.

[Exercise configuration]



[Related devices]

Devices	Contents	R: read W: write
SD6020	Analog input CH1 (digital value)	R
SD6060	Analog input CH2 (digital value)	R
SD6180	Analog output (digital value)	R/W

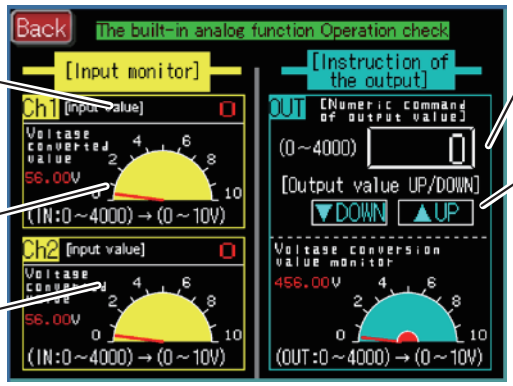
- For this exercise, only the required device numbers have been extracted. Do not use the other numbers.

Training Machine screen **[Basics: MELSEC iQ-F Programming (GX Works3 Version)] → [Analog]**

[Analog input value SD6020 monitor]
The digital value retrieved using the potentiometer is displayed.

The retrieved digital value
0 to 4000 is converted to 0 to 10 V by the GOT and displayed.

[Analog input value SD6060 monitor]

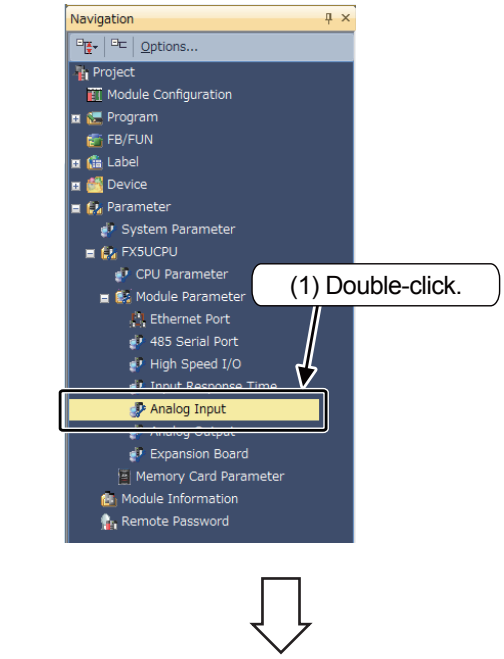


[Output value SD6180 monitor]
The output value can be instructed by touching the screen.

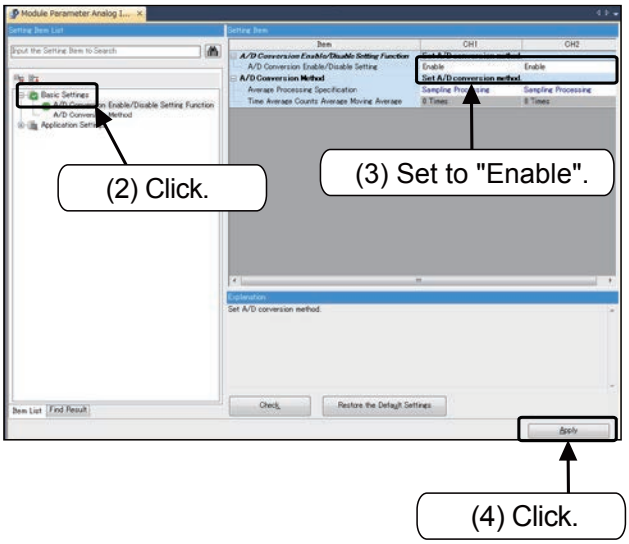
[Output value SD6180 value UP/DOWN]

The digital value 0 to 4000 being output is converted to 0 to 10 V by the GOT and displayed.

● Analog input parameter settings

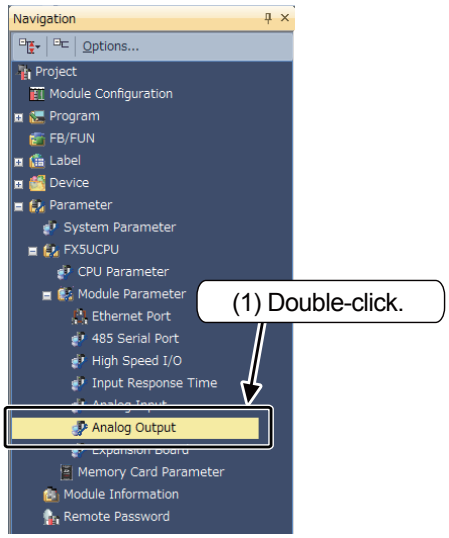


(1) Double-click [Parameter] → [FX5UCPU] → [Module Parameter] → [Analog Input] on the navigation window.

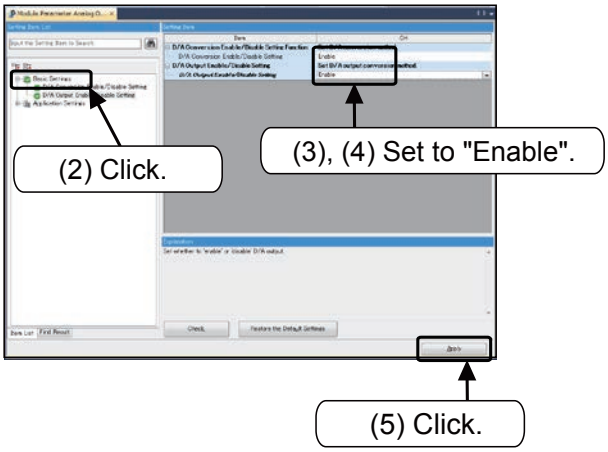


(2) Select [Basic Settings].
(3) Set the A/D conversion Enable/Disable setting to "Enable".
(Set for CH1 and CH2.)
(4) After setting, click [Apply].

● Analog output parameter setting



(1) Double-click [Parameter] → [FX5UCPU] → [Module Parameters] → [Analog output] on the navigation window.

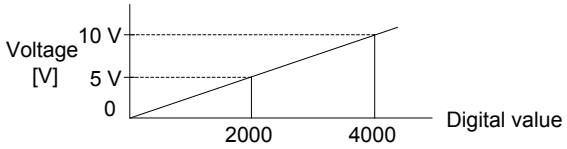


(2) Select [Basic Settings].
(3) Set the D/A Conversion Enable/Disable Setting to “Enable”.
(4) Set the D/A Output Enable/Disable Setting to “Enable”.
(5) After setting, click [Apply].

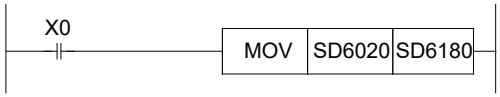
<<Operation check>>

Use GX Works3 to monitor SD6020, SD6060 and SD6180.
Display the training machine screen from the previous page.

(1) Rotate the knob of ch1/ch2 and confirm on the GOT screen that the value of [0 to 4000] is input.
(2) Operate the GOT screen, designate the instruction value of the output within 0 to 4000 and confirm whether the voltage is output.



(3) Example
When X0 is set to ON, the ch1 volume input value (0 to 4000) can be used as the analog output instruction (0 to 10 V output).



Chapter 12
LET'S LEARN THE PROGRAM FLOW

The sequence of program execution may be changed

A PLC is not just a machine that conducts cyclic operations by following program steps in a fixed sequence of operations.
The sequence of program execution can be changed by using various instructions. Additionally, as described in Chapter 10, the sequence of a program may be changed with interrupt processing.

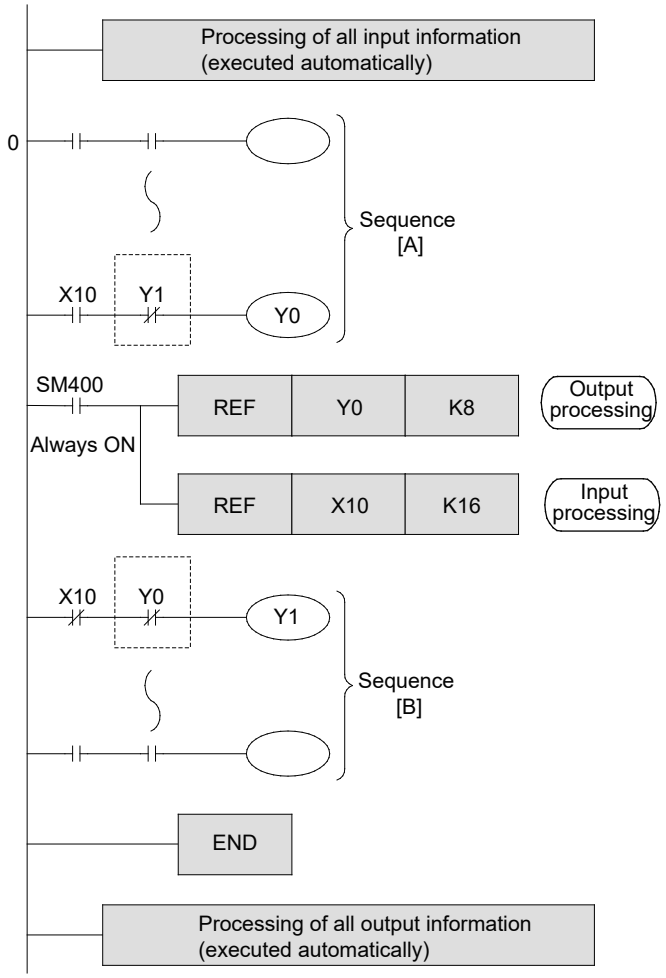
In this chapter...

In order to create programs for efficient execution, the major control instructions that have a direct affect on the program flow for changing the sequence of program execution will be explained in detail.
These control instructions include I/O refresh, jump instructions, subroutines, loop instructions, and other important instructions.
This chapter will also examine the operation methods of the PLC.

12.1 I/O refresh instructions (REF)

The I/O processing mode for micro PLCs is known as a batch refresh mode. The information at all of the input terminals (input ON or OFF) is stored into an input image memory prior to the operation of step 0.
After the END instruction (or the FEND instruction) is executed, the information is output from the output image memory to the latch memory, and then simultaneously transferred from the latch memory to the output terminals (See page 10-2.)
To acquire the latest input information during the sequence operation, or to output the result of the operation as quickly as possible, the I/O refresh instruction can be used.
With the FX5U CPU module, by specifying a direct access input/output (DX, DY), the input/output can be directly accessed in 1-point modules.

<<Operation outline>>



- The entire sequence is divided into sections A and B. After sequence A is completed, the sequence program proceeds with output processing. Before the execution of sequence B, input processing must take place.
- At this phase, the output information of the 8 output points from Y0 to Y7 is output.
- At this phase, the input information of the 16 input points from X10 to X27 is stored into the memory.
- This example shows that I/O processing can be executed two times respectively in one operation cycle. This makes it possible to output the operation result at the earliest timing possible, using the latest input information.

Reference

Output interlock

In a program where I/O processing is executed more than one time in a single operation cycle, the operation result may differ between sequences A and B if the input has changed between ON and OFF during the operation cycle. As in the above figure, if no interlock is provided on outputs Y0 and Y1, the outputs may be activated simultaneously.

12.2 Input filter adjustment function

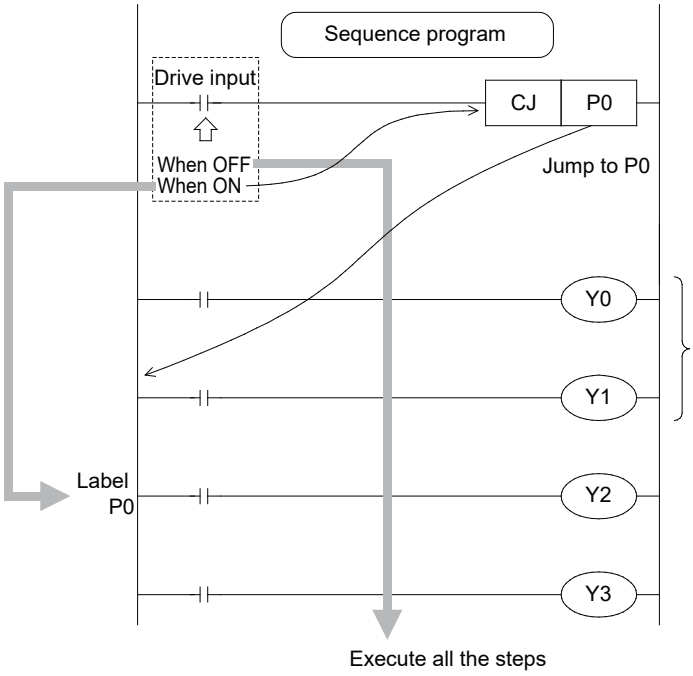
- Generally, to avoid chattering and noise at the input contacts, the inputs of the PLC are equipped with a 10 ms C-R filter.
  However, if the PLC uses non-contact inputs to avoid noise, the use of the above-mentioned filter will just impede the execution of high-speed import.
- With the FX5U CPU module, a digital filter is used for all inputs. The filter value can be changed to 10 μs, 50 μs, 0.1 ms, 0.4 ms, 0.6 ms, 1 ms, 5 ms, 10 ms, 20 ms or 70 ms.
  Note that a 2.5 μs to 150 μs hardware filter is applied on each input of the CPU model and I/O module.
  Thus, the actual filter time is the value to which the hardware filter has been added.
- Change the input response time with the parameter when using the following functions or instructions.
  - Interrupt pointer
  - High speed counter function
  - When using X10 to X17 with MTR instruction
  - SPD instruction
- Refer to the previous section “10.2 Using input interrupt” for details on setting the input response time parameter.

12.3 Jump instructions (CJ)

The jump instruction is an instruction that can shorten the operation cycle and enable the use of dual coils by preventing some parts of the sequence program from running.

<<Operation outline>>

When the drive input is turned on, a jump instruction is executed and the program jumps to the step labeled with PXXX. The executed actions may vary with the device and the number as follows.



- When the drive input is OFF, steps in the program will be executed.
- When the drive input is ON, the program jumps to pointer P0. In this case, the steps before pointer P0 will not be executed.

Reference

Pointer numbers

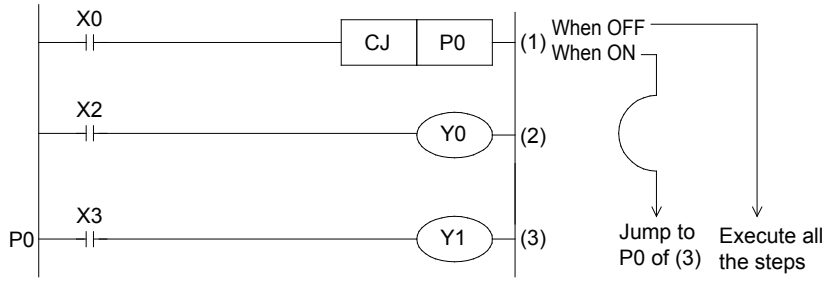
- The FX5U CPU module pointers include the global pointer and label assignment pointer.

Pointer	Description
Global pointer	This pointer can be referred to from all programs.
Label assignment pointer	This pointer is assigned to a label and used. The pointer number assigned to the label is automatically determined by the GX Works3, so the user cannot specify the pointer number to be assigned.

- Use the GOEND instruction to jump to the END instruction or FEND instruction.
- Label numbers are also used by the CALL instruction or XCALL instruction described later. The numbers cannot be overlapped.

<<Instruction operation>>

Let's confirm the execution of the jump instruction.



<<Operation check>>

Use GX Works3 to monitor the circuit.

When X0 is OFF

(1) [Turn "OFF" X0]

⇩

(2) Y0 turns ON or OFF when X2 is turned ON or OFF.

⇩

(3) Y1 turns ON or OFF when X3 is turned ON or OFF.

When X0 is ON

(1) [Turn "ON" X0]

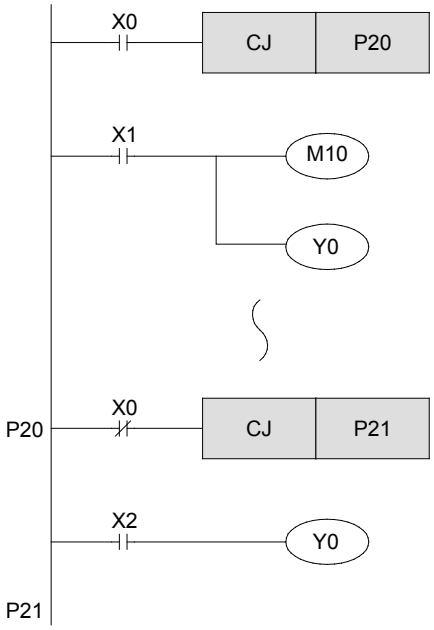
⇩

(2) Y0 does turn ON or OFF when X2 is turned ON or OFF.
(The program at (2) is skipped by the jump instruction.)

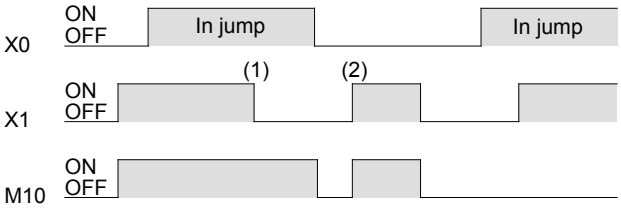
⇩

(3) Y1 turns ON or OFF when X3 is turned ON or OFF.
(During a jump to the label P0)

◎ Functional principles of Y, M coils



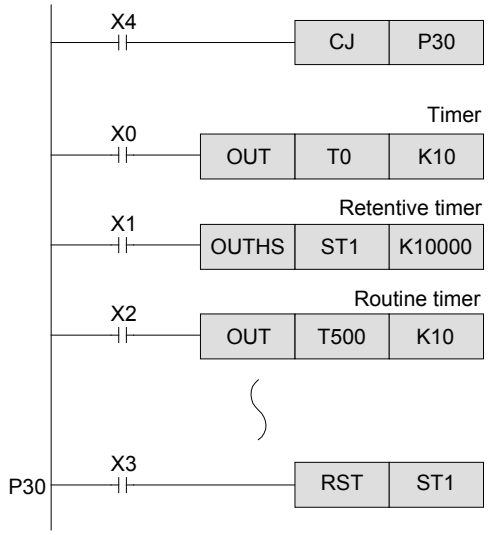
- When the coil of a Y (output), M (Internal relay), or S (step relay) is skipped, the ON/OFF status that was stored prior to the jump will be maintained.



(1) Even if X0 turns off, M10 stays ON.
(2) Without jump instruction, M10 turns ON or OFF according to the status of X0.

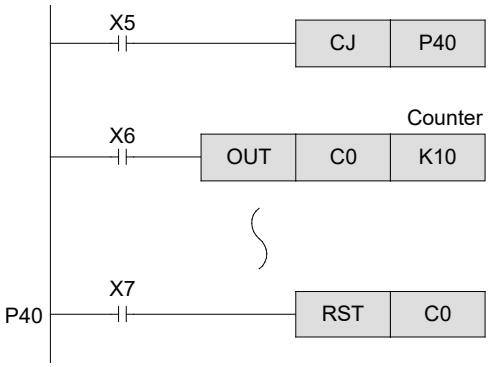
- The output Y0 is a double coil. When X0 = OFF, the program runs according to the status of X1. When X0 = ON, the program runs according to the status of X2.
- Even with a double coil, if one of the two is skipped, only the other can be activated. As a result, the coils may be activated separately.

◎ Functional principles of timers



- The general timer suspends clocking when skipped by a jump instruction, and resumes clocking after the jump instruction is deactivated.
- The routine timer continues clocking even when skipped by a jump instruction. If the timers timed out while a jump instruction is activated, the output contact of the routine timer opens. The parameters must be set to use the routine timer.
- If the reset instruction for a retentive timer is programmed after or before a jump instruction is programmed, the reset instruction (e.g. to reset a contact or clear a current value) can be executed even if the coil of the retentive timer is skipped.

◎ Functional principles of counters



- A general counter suspends counting when skipped by a jump instruction.

- If the reset instruction for a counter is programmed after or before the location where a jump instruction is programmed, the reset instruction (e.g. reset a contact or clear a current value) can be executed even if the coil of the counter is skipped.

◎ Functional principles of instructions

- An instruction will not be executed when skipped by a jump instruction.

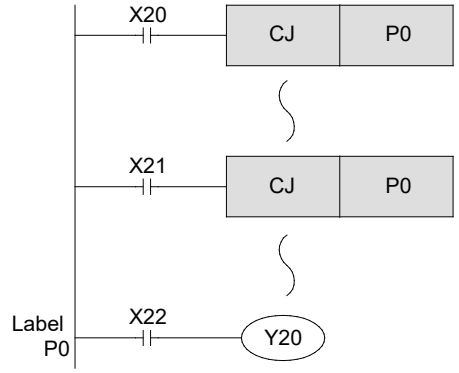
  However, the operation of the following instructions and some other instructions continues.
  TMR instruction, SPD instruction, DSPD instruction, PLSY instruction *1, DPLSY instruction *1, PWM instruction

  *1: While the PLSY instruction or DPSLY instruction is being executed, normal operation will not take place in an environment were execution at every scan is not possible such as in the user program, or when jumping with the CJ(P) instruction. However, pulse output will continue.

Reference

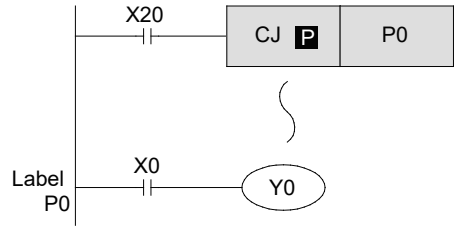
Jump methods

[Jump to the same pointer]



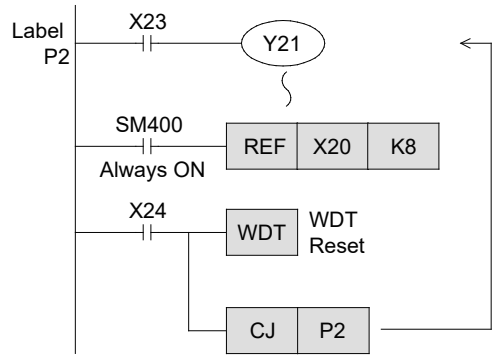
- The pointer number in the operands to the left have the same number.
- When X20 is ON, the program will jump to label P0 from here. When X20 is OFF but X21 is ON, the program will jump to label P0 from CJ of X21.
- Duplicate label numbers (including labels for CALL instructions) cannot be used. If two label numbers are the same, a PLC error will result.

[Jump only for one operational cycle]



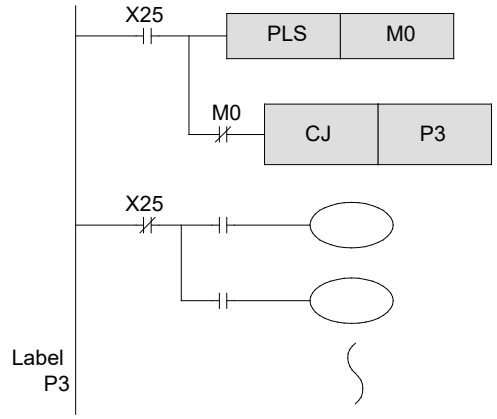
- The CJ **P** is used. If the drive input changes from OFF to ON , the program jumps to label P0 only for one operational cycle.

[Jump upward]



- Although a label can be programmed at an step earlier than the CJ instruction, if X24 turns ON for more than 200 ms (scan monitor time), a watch dog error will occur, and the PLC will stop (ERR LED will light).
- One of the following procedures must be taken.
  - Change the scan time monitor time in the CPU module parameter RAS setting to a longer time.
  - Using the program, rewrite the contents of the special register SD8000 (watch dog timer time) to a longer time.
  - Program the WDT instruction (watch dog timer reset instruction) as shown on the left.
    Write a condition that turns the CJ instruction OFF so a repeated flow is not created.
    (In the left example, the X24 input refresh is executed with the REF instruction.)

[Delay jump by one operation cycle]

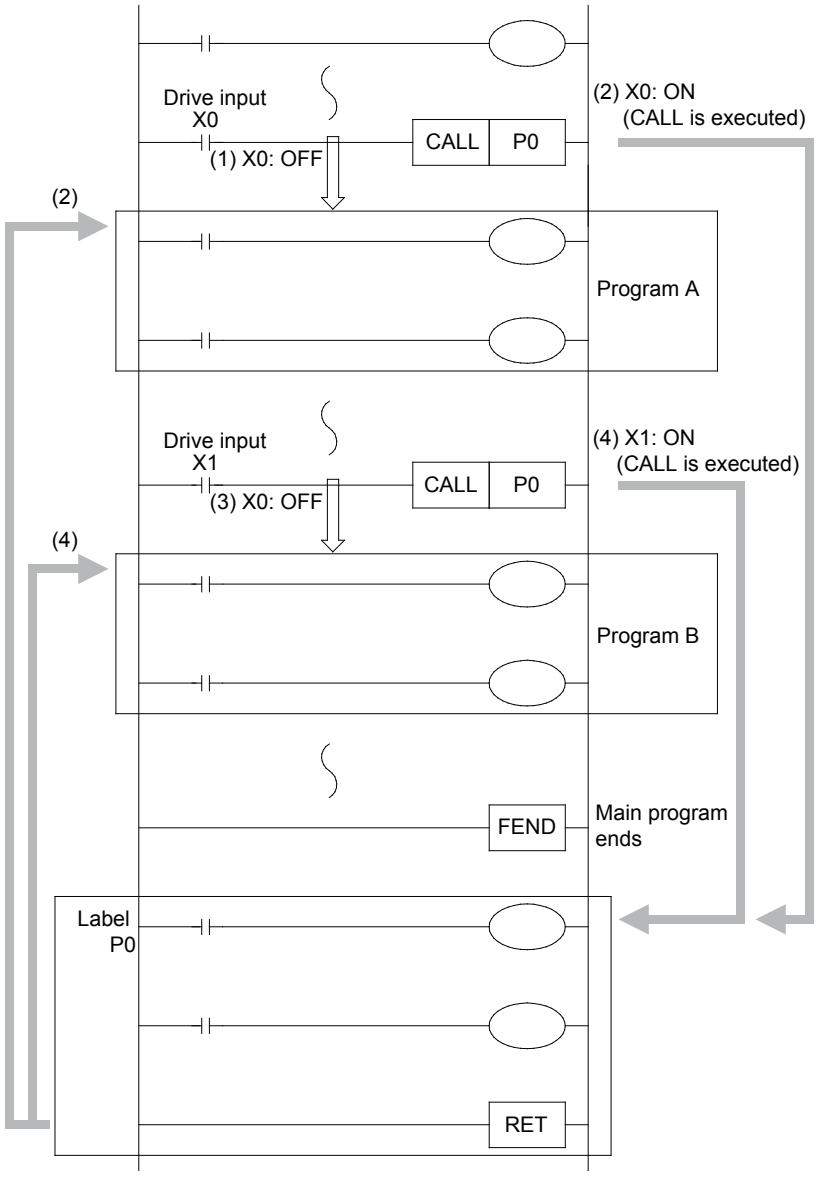


- In the circuit to the left, if X25 turns ON, the jump to P3 will be turned on in the second operational cycle.
  During the first operation cycle, all the outputs between CJP3 and P3 are turned off.

12.4 Call subroutine instructions (CALL, RET)

The CALL Subroutine instruction and XCALL subroutine program call instruction are instructions for executing a subroutine program within the main program.
A subroutine program can be a program that contains actions to be executed several times or a program that executes only the necessary parts of an action.
The following explanation uses the CALL instruction.

<<Operation outline>>

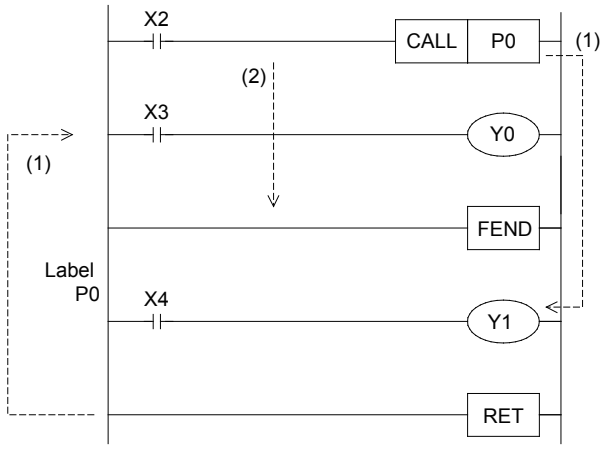


(1) When X0 is "OFF", "Program A" will be executed.
(2) When X0 is "ON", "Program A" will be executed after the subroutine program of "label P0" is executed.
(3) When X1 is "OFF", "Program B" will be executed.
(4) When X1 is "ON", "Program B" will be executed after the subroutine program of "label P0" is executed.

- When the CALL instruction is executed, the program will jump to the designated label P0. After the subprogram is executed at P0, it returns to the original program step by the SRET instruction.
- The label used by a CALL instruction must be programmed after the FEND instruction.
- The same label number must not be used including labels for CJ instructions.
- Generally, the I/O refresh instruction is used before and after a subroutine program. Additionally, for timers in subroutine programs (as with interrupt routine programs) it is necessary to use the routine timers.

<<Instruction operation>>

Execute a subroutine program with a CALL instruction, and confirm the actions of the subroutine program.



Jump to label P0 with X2 ON

Execute Y0 with X3 ON/OFF

Quit main program

Program that is valid only while X2 is ON

Quit subroutine

<<Operation check>>

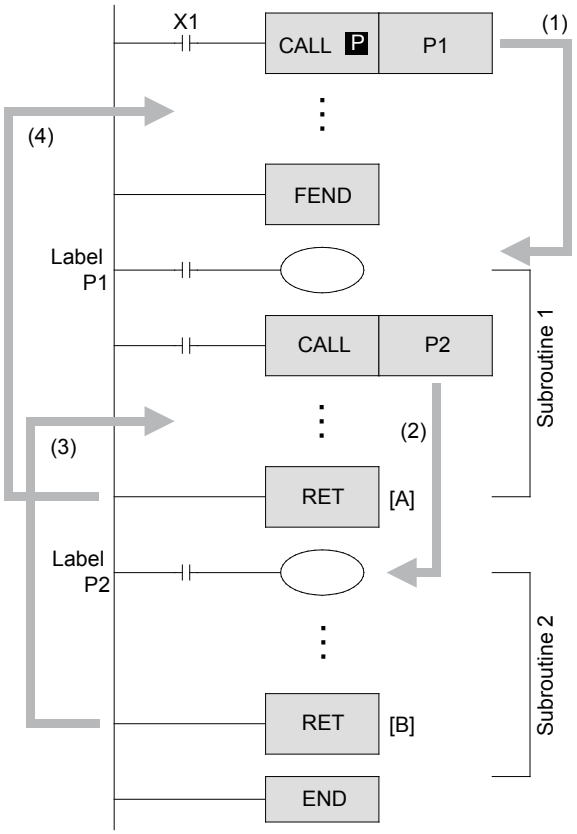
Use GX Works3 to monitor the circuit.

(1) [When X2 is OFF]
- Y0 turns ON or OFF when X3 is turned ON or OFF.
- Y1 does turn ON or OFF when X4 is turned ON or OFF (since the subroutine was not called).

(2) [When X2 is ON]
- Y0 turns ON or OFF when X3 is turned ON or OFF.
- Y1 when X4 is turned ON or OFF (since the subroutine).

[CALL instruction or XCALL instruction nesting (multi-nesting)]

If a CALL instruction or XCALL instruction is programmed within a subroutine, it is known as multi-nesting. This type of CALL instruction or XCALL instruction can be programmed up to 15 times. In all, a 16-level nest is permitted.
The following explanation uses the CALL instruction.



(1) When the CALL **P** instruction is used, the CALL instruction is executed only when the input X1 is turned on, and the program jumps to label P1.

(2) If the CALL P2 instruction is executed in the subroutine program labeled as P1, the program will jump to label P2.

(3) The second subroutine with label P2 is executed. When the operation moves to the RET instruction [B], the program will return to the next step of CALL P2.

(4) Similarly, when the operation moves to RET instruction [A], program will return to the next instruction of CALL **P** P1.

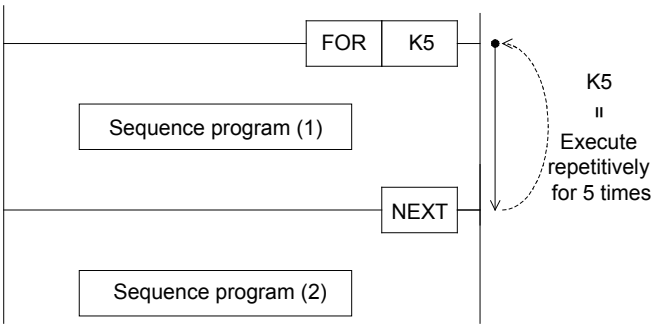
Reference

- Actions of timers and counters in subroutine and interrupt routine programs…
  If general timers are used in a subroutine or interrupt routine program, clocking is activated only in the main program.
  This is the same for counters. Generally, counters should not be used in subroutine or interrupt routine programs. However, in a routine program that executes at each operation cycle, as shown in this example if the CALL instructions are always driven , general timers and counters can be used. The actions of the timer and counter in a subroutine are the same as those for the jump instruction.
- Label numbers are also used by the CJ instruction described previously. The numbers cannot be overlapped.
- In the P-RET instruction, it is not allowed to use MC-MCR, STL-RETSTL, I-IRET and another P-RET instructions for programming.
  In addition, the P-RET instruction cannot be used in MC-MCR, FOR-NEXT, STL-RETSTL and I-IRET instructions.

12.5 Loop instruction (FOR-NEXT)

The loop instruction is an instruction that executes program segments from a FOR instruction to a NEXT instruction for n times and then executes the program content after the NEXT instruction.
n is set to 1 to 32767 times. Specifying "n = -32768 to 0" is equal to specifying "n = 1".

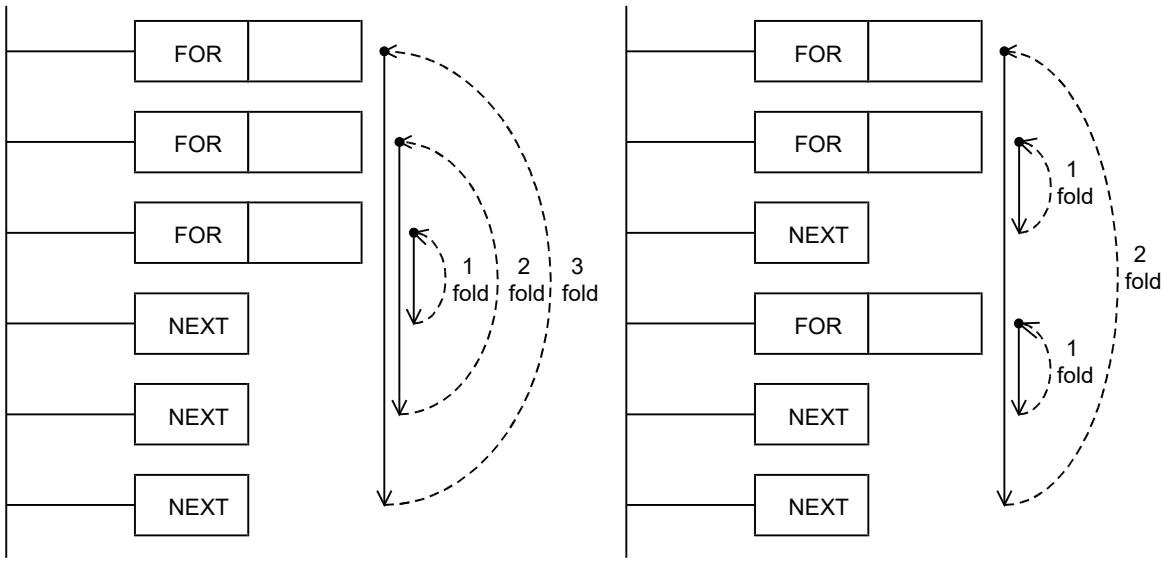
<<Operation outline>>



- The "sequence program (1) " is executed for the specified number of times between the FOR-NEXT instructions.
- After the repetitive execution for the specified number of times, the program proceeds with the execution of "sequence program (2) ", which is placed after the NEXT instruction.

Reference

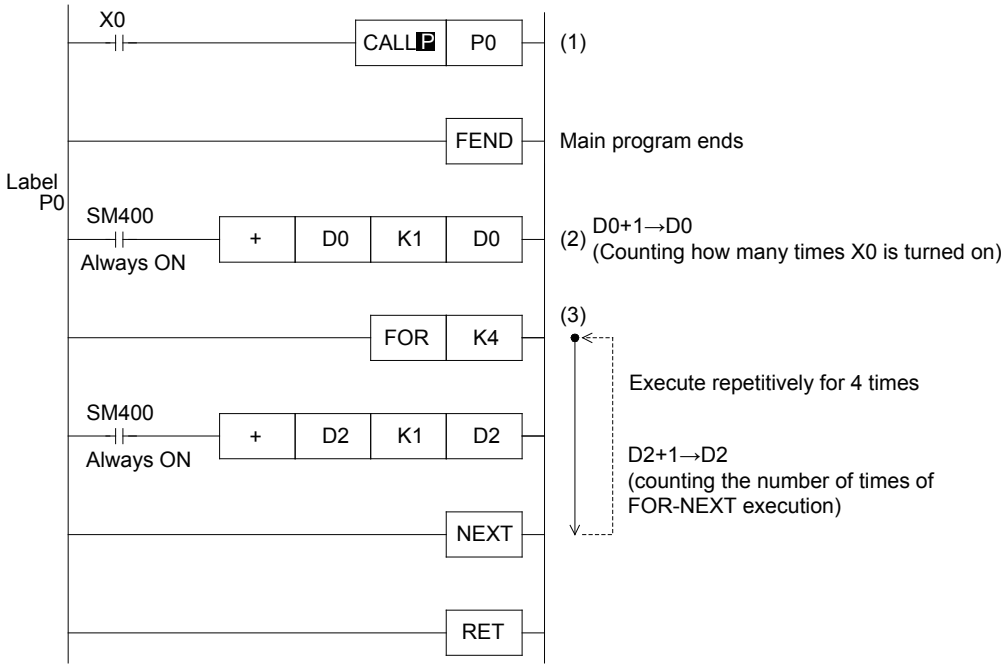
- About nesting (multi-nesting)
  When a FOR-NEXT instruction is programmed in the FOR-NEXT program, up to 16 nesting levels are possible.



If the numbers of FOR and NEXT are not identical, an error may occur.

<<Instruction operation>>

Use FOR-NEXT instructions and confirm the actions of the program.



<<Operation check>>

Use GX Works3 to monitor the circuit.

(1) [Set X0 to ON.]

(2)

(3)

⇨ ● [CALLP P0] instruction will be executed, and the subroutine at label P0 will be executed for only one operation cycle.

⇨ ● By [+ D0 K1 D0] instruction, the number of times the subroutine is executed is counted at D0.

⇨ ● [+ D2 K1 D2] instruction is executed repetitively for four times by the [FOR K4] instruction.

● As a result of the execution, [4] is stored into D2. (Then, 4 is added to D2 every time X0 is turned ON.)

Since the execution ends in a split second, switch X0 between ON and OFF, and try repeating the action.

MEMO

Chapter 13

GUIDE TO HANDY INSTRUCTIONS AND FUNCTIONS

Handy purpose-oriented instructions

Handy purpose-oriented instructions are principle control that can become extremely complicated when combined with general instructions and which have been bundled into a single macro instruction.

Instructions for intelligent function module

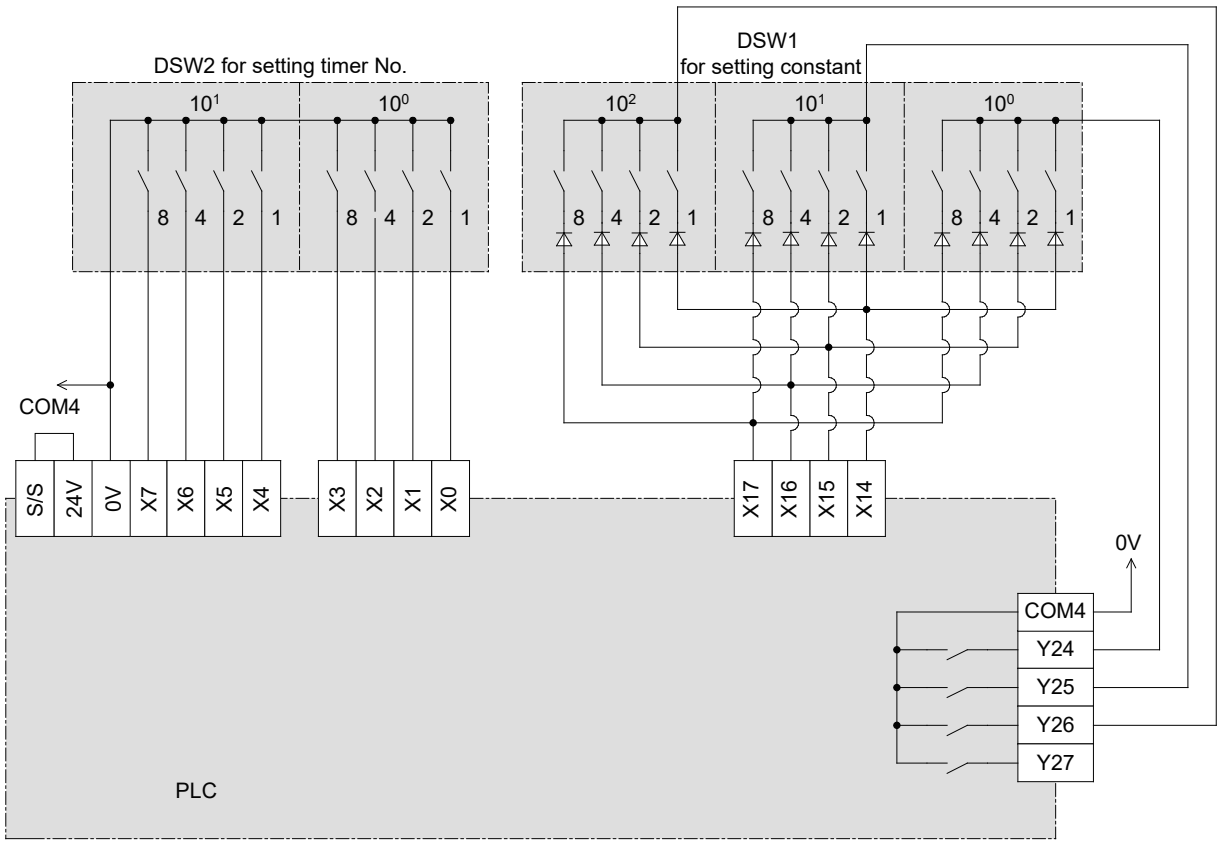
Data can be exchanged with various handy devices (intelligent function modules) using FROM/TO instructions.

In this chapter...

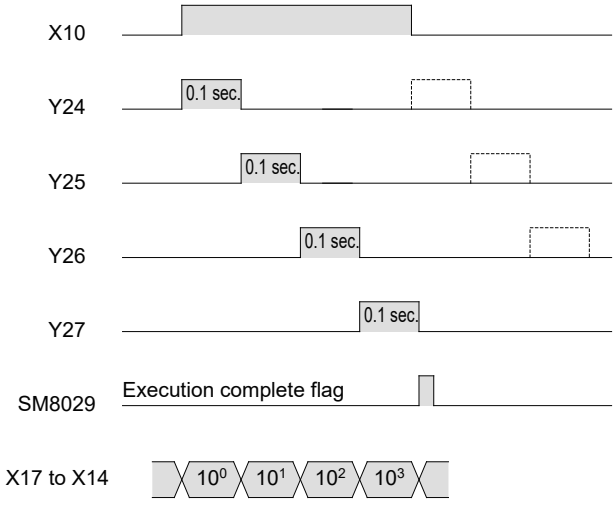
Of the many handy instructions, the ones that are not restricted by the PLC's number of input/output points and that do not require special devices, and the mechanism of communication with the intelligent function module are introduced. By understanding each function, you will see how handy the instructions are.

13.1 Digital switch time-division input instruction DSW

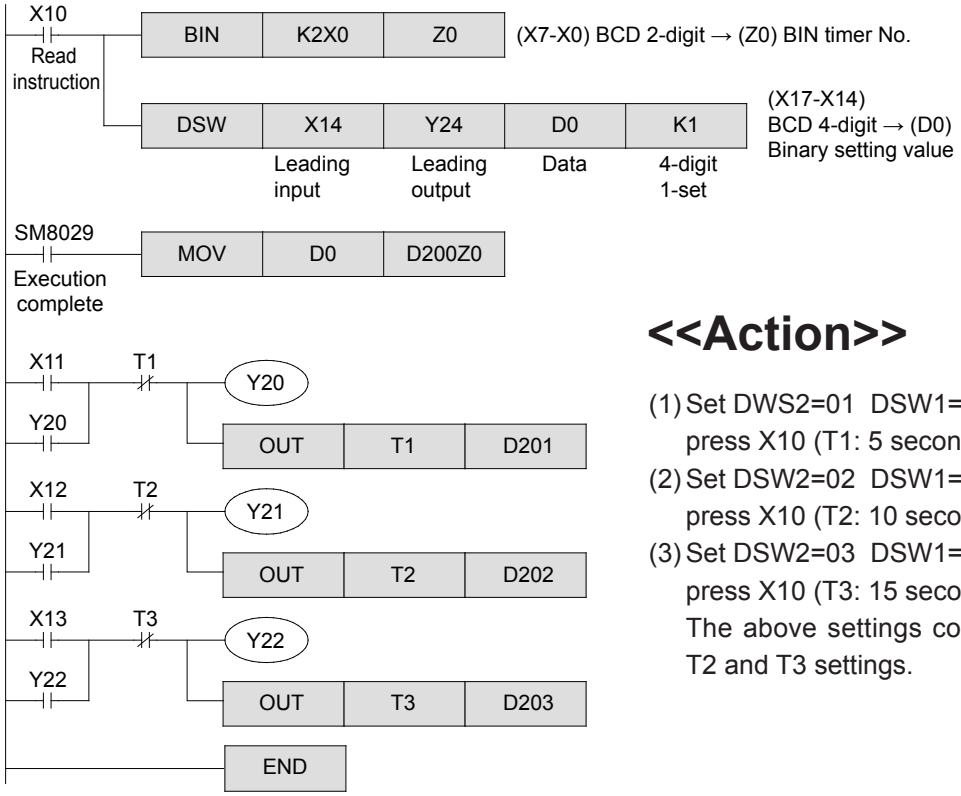
The digital switch (DSW) is used as a simple value setting module.
In this section, examples of circuit that inputs the timer setting value by directly reading a 2-digit digital switch and by using time-division input with a the 3-digit digital switch are explained.
By using the DSW instruction, 4-digit 2-set digital switches can be time-division read.



The timing for reading the DSW instruction is as follows.



- When the instruction is executed for an instant, Y24 to Y27 are sequentially operated at a 0.1 second pitch, and each time the DSW 10^0 to 10^3 digits are input into the PLC.
- When retrieval of the data is completed, the execution complete flag SM8029 activates.
- The above operation can be repeated by continuously turning the drive input X10 ON. However, the PLC will require a transistor output in this case.



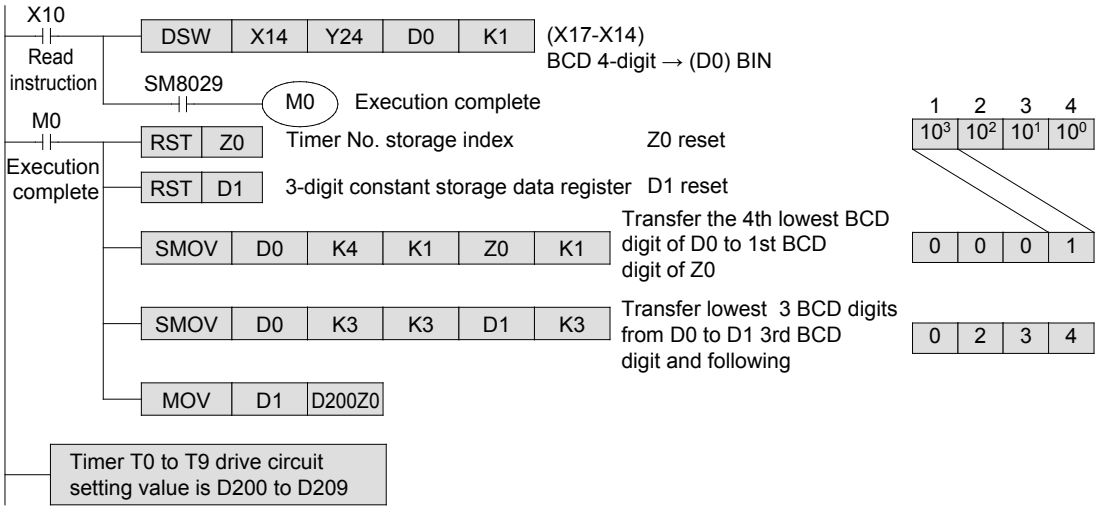
<<Action>>

(1) Set DWS2=01 DSW1=050, and press X10 (T1: 5 seconds).
(2) Set DSW2=02 DSW1=100, and press X10 (T2: 10 seconds).
(3) Set DSW2=03 DSW1=150, and press X10 (T3: 15 seconds).
The above settings complete the timer T1, T2 and T3 settings.

Reference

Other usage methods

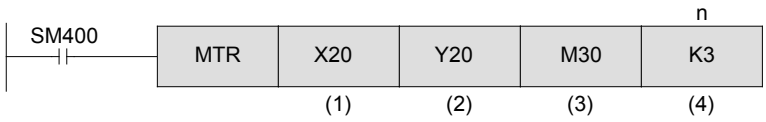
A circuit in which the uppermost digits of the 4-digit 1-set digital switch DSW are used for the timer number setting, and the last three digits are used for setting the timer is as follows.



The SMOV instruction handles BIN data, but when this instruction is used, the data is automatically converted to BCD. Each BCD digit is moved, before it is returned to a BIN value.
Thus, this instruction is suitable for distributing and combining BCD digits for the BCD calculation.

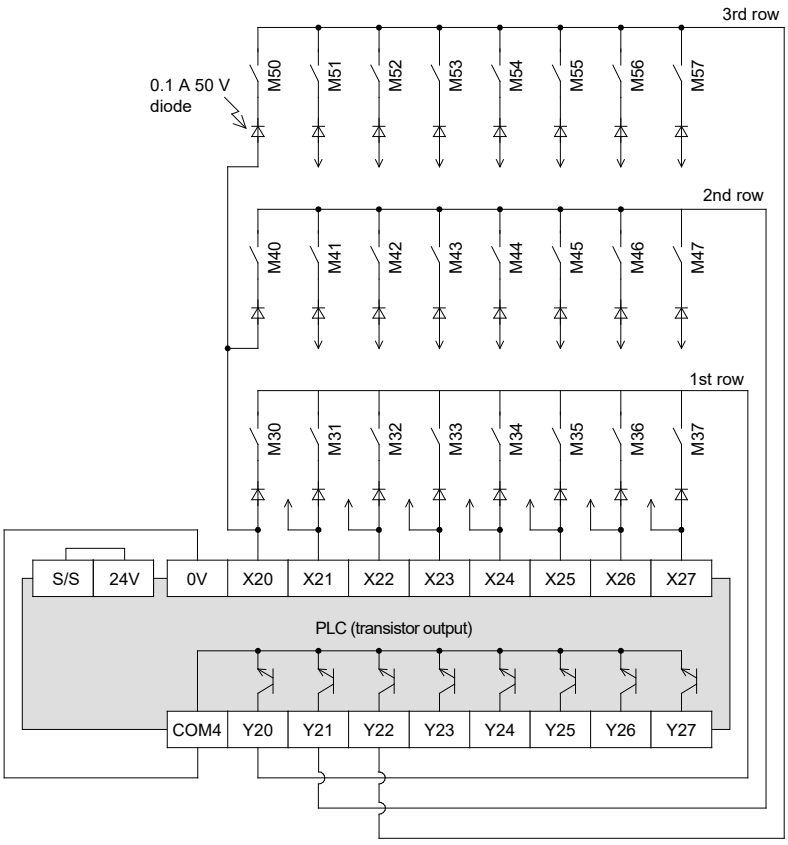
13.2 Matrix input instruction MTR

The MTR instruction is a handy instruction that can greatly reduce the number of PLC input points by acting as a transistor output for general contact inputs that do not require a high speed response.

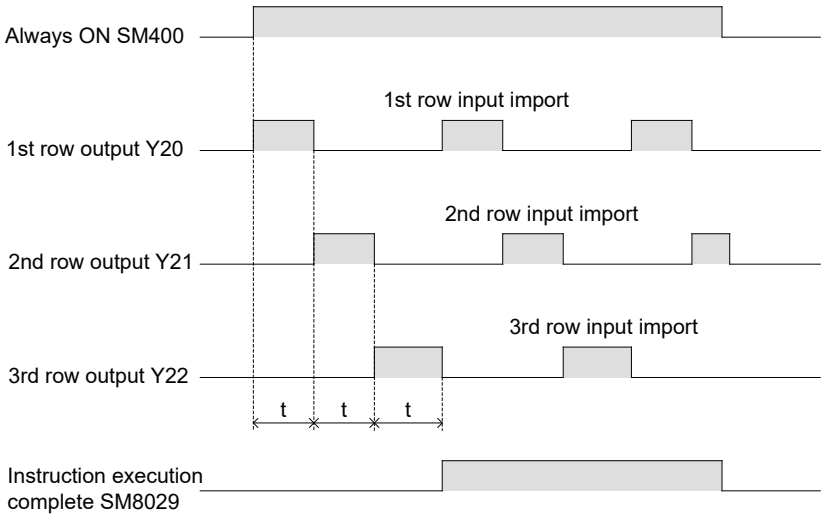


(1) Lowest order digit such as X0, X10, X20 is 0 input
(2) Lowest order digit such as Y0, Y10, Y20 is 0 output
(3) Element such as Y, M, S with lowest digit of 0
(4) n = K2 to K8 is valid

- This instruction is used to sequentially read the 8 point x n row (n=2 to 8) input signal using the 8-point input and n-point output.



- A compact diode must be serially connected to each input as shown on the left.
- The ON/OFF storage destination number for each external switch is M30 to M57 in this example.
- When the PLC runs, the output Y20, Y21 and Y22 turn ON sequentially.
- Allowing for a delay in the input filter response, the input is read after a specified time and then the next row's output turns ON.



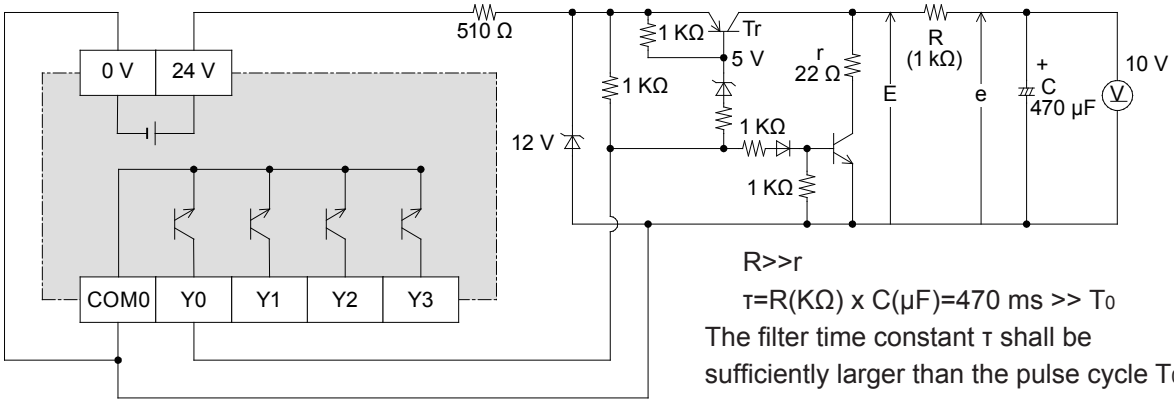
<<Control timing>>

- When using the input X0 to X17, t=10 ms and it takes 80 ms to retrieve the 8-row 64-point input.
  In this case, change the input response time set in the module parameters to 5 ms, and connect a pull-up resistor to the transistor output being used with the MTR instruction.
- When using input X20 and following, t=20 ms, and it takes 160 ms to read the 8-row 64-point input.
- This instruction is executed with a 10 ms or 20 ms cyclic interrupt (EI instruction is not required), and is not affected by the PLC's operation cycle.
  The row switchover output and input retrieval are also conducted immediately with the refresh operation.
- The 80 ms or 160 ms retrieval cycle for the 8-row 64-point input means that the ON/OFF width for each input must be larger than the following value.

13.3 Pulse width modulation function PWM output

The PWM output outputs pulse train at the specified cycle and pulse width.
The cycle and pulse width module, and the output number assignment, etc., are set with the parameters.
The HIOEN instruction is used to start and stop the pulse output.
The pulse width/cycle setting range is 1 to 2147483 ms or 1 to 2147483647 μs.

These can be used to control the ON/OFF proportion, to provide a smoothing circuit and display a meter, or to output a speed instruction for the inverter.

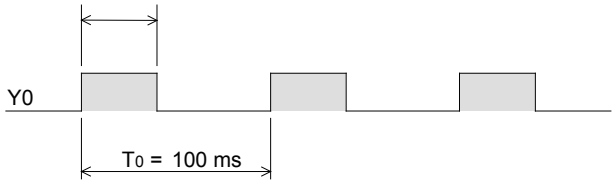


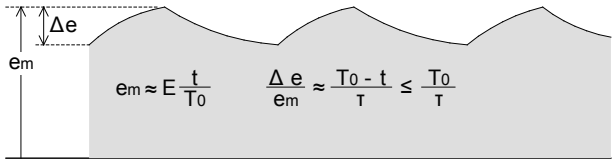
R>>r

τ=R(KΩ) x C(μF)=470 ms >> T_0

The filter time constant τ shall be sufficiently larger than the pulse cycle T_0.

PWM CH1 pulse width t
(High-order: SD5303, low-order: SD5302)

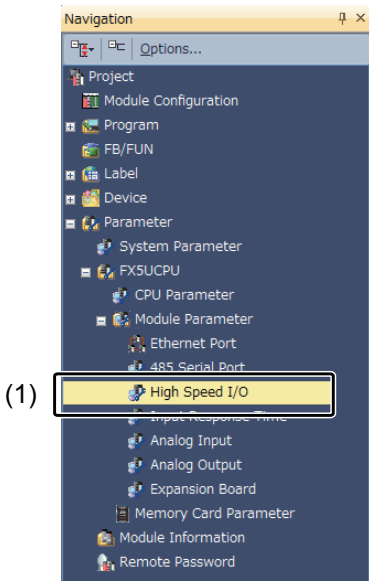




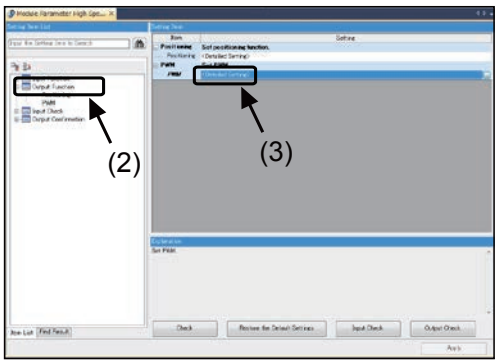
- The pulse cycle is 100 ms in the left row, and if the CH1 pulse width (high-order: SD5303, low-order: SD5302) is changed between 1 and 100, an output of 1 to 100% will be attained.
- The smoothing circuit's output has a ripple as shown on the left. The maximum value em and ripple Δe can be calculated with the expression on the left.
- E is the collector side voltage when transistor Tr turns ON, and the τ>>T_0 condition is required.

<<Operation outline>>

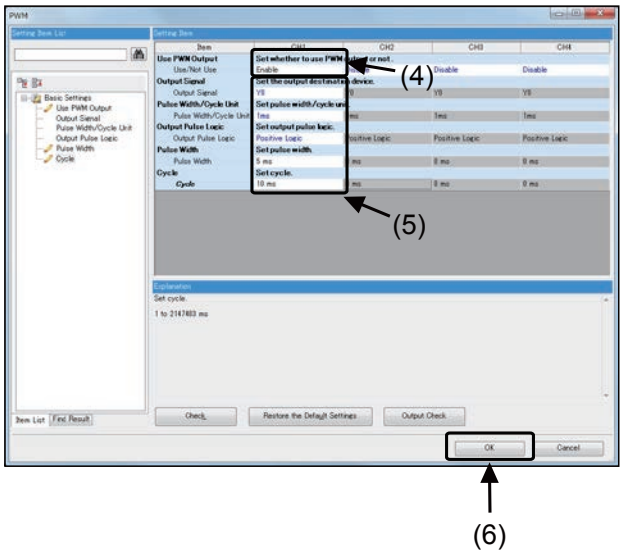
● Parameter setting



(1) Double-click [Parameter] → [FX5UCPU] → [Module Parameter] → [High speed I/O] on the navigation window.



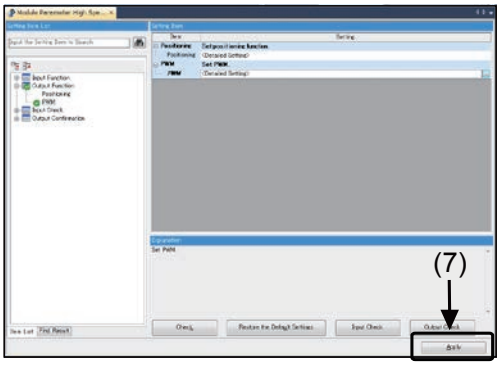
(2) Click [Output Function].
(3) Double-click [PWM] → [Detailed Setting].



(4) Set CH1 to “Enable”.
(5) Set each item as shown below.

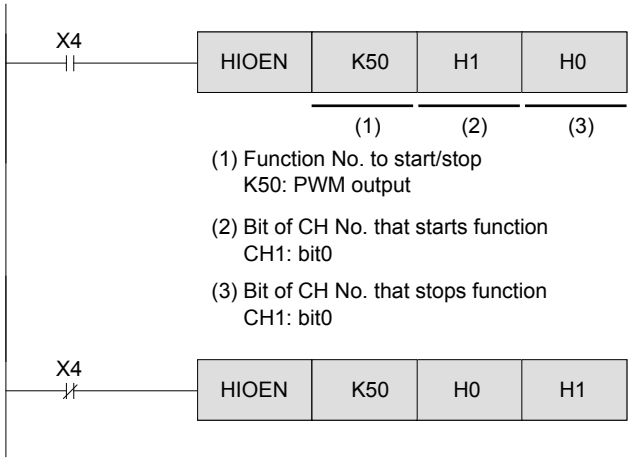
Output signal	: Y0
Pulse width/cycle module	: 1 ms
Output pulse logic	: Positive logic
Pulse width	: 5 ms
Cycle	: 10 ms

(6) After setting, click [OK].



(7) Click [Apply].

● Program example

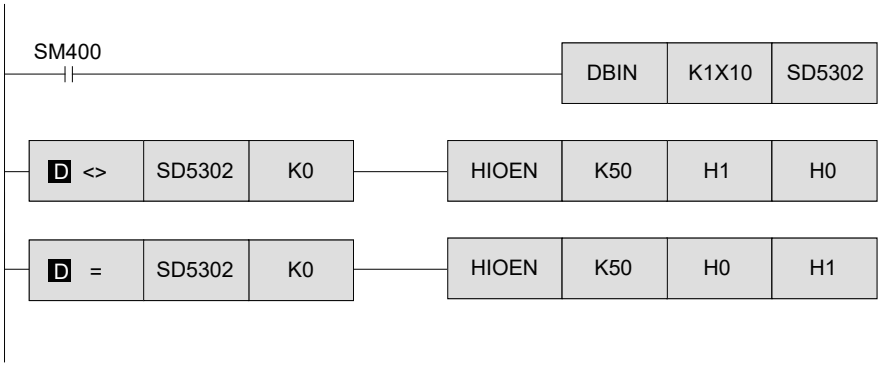


● Start the CH1 (Y0) PWM output.

● Stop the CH1 (Y0) PWM output.

Let's try

Set the demonstration machine's DSW between 1 to 9, and adjust the brightness of the Y0 LED.
As the setting value increases from 0 to 9, the Y0 LED brightness will change.
In this example, ON/OFF control is executed at a 10 ms cycle.



13.4 Module access instruction FROM/TO

An intelligent function module can be connected to the FX5U CPU module.
This intelligent function module can be used for analog control, and positioning control, etc.
To perform these controls, the intelligent function module has an internal 16-bit RAM memory called a buffer memory (BFM).
The FROM/TO instructions can be used to communicate with this buffer memory. It is similar to a monitoring office and satellite that conduct individual work and communicate when necessary.

- What is analog control…
  The PLC mainly uses ON/OFF (1 or 0) digital control.
  Thus, it is difficult to directly control matters that continuously change, such as temperature, flow rate or wind rate. A dedicated module is required to perform analog control. The FX5U CPU module has a 2-point analog voltage input and a 1-point analog voltage output function built-in.
  Refer to the previous section “ANALOG FUNCTION” for details on the built-in analog function.

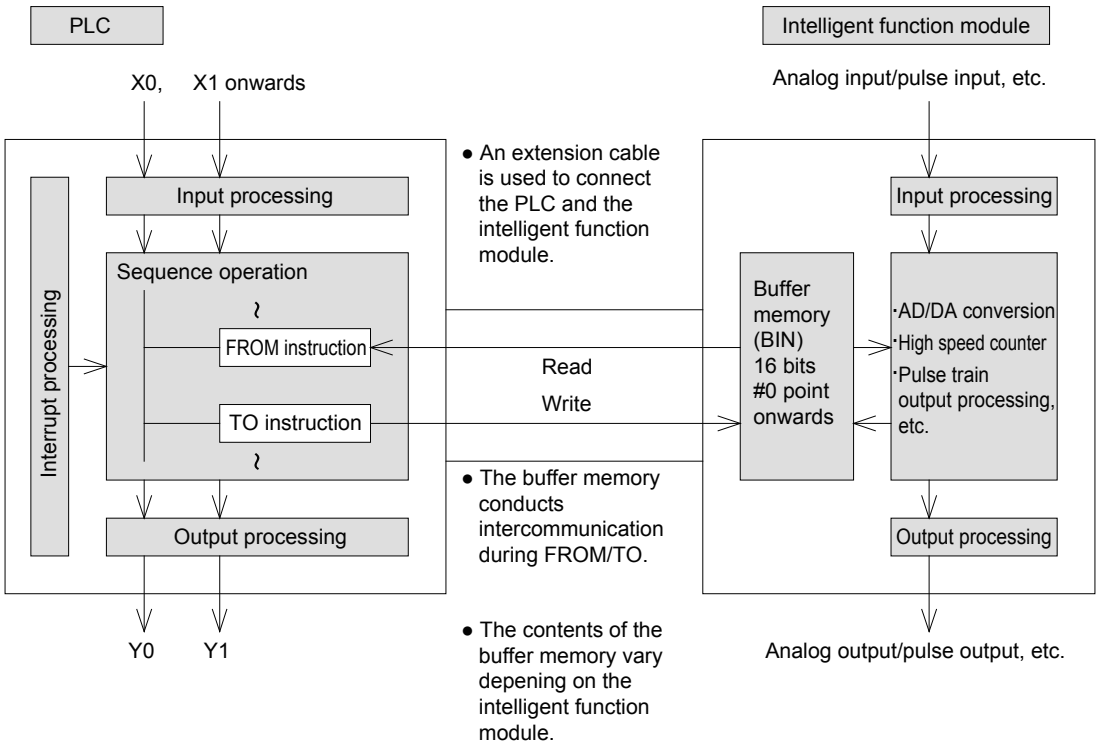
  Examples of analog control
  - Temperature control
  - Flow rate control
  - Speed control
  - Tension control
  - Pressure control
  - Wind power control
  - Voltage/current monitoring control, etc.

- What is positioning control?
  For example, if we consider the control used to transfer workpieces with a belt conveyor, etc., how would you make sure that the position where the workpiece stops is accurate?
  You could install a sensor at the stop position, and use the input ON state to stop the conveyor.
  However, if you want to feed the workpieces to that stop position at a high speed, you will face various problems.
  The positioning module is a dedicated module used to ensure that the workpiece moves to the designated position at a high speed and stops with a high precision.

  Examples of positioning control
  - Constant step and stop control
  - Constant multi-step feed control
  - Rotary angle control
  - Variable speed pattern control
  - 2-axis synchronous control, etc.

Communication with the intelligent function module's buffer memory is carried out when the PLC executes the FROM/TO instruction in the sequence operation.
The intelligent function module's current value and status information is read out with the FROM instruction, and the various settings for operating the intelligent function module are written in with the TO instruction.

◎ Mechanism of communication between PLC and intelligent function module



Reference

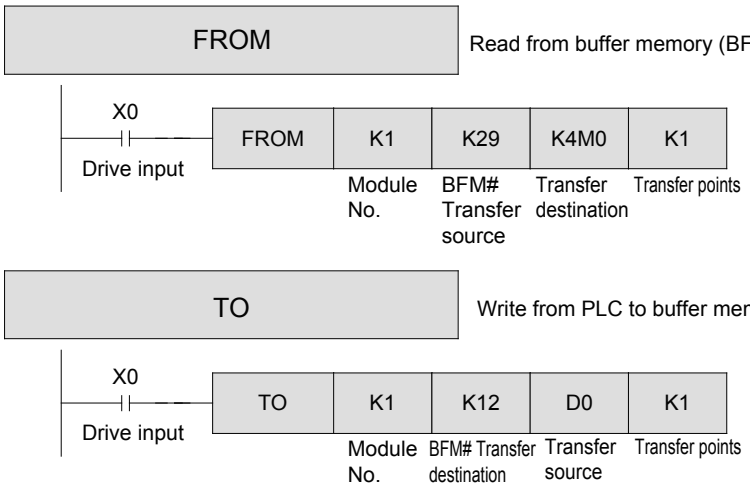
Main types of intelligent function modules

The following types of intelligent function modules are available.

- Analog input
- Analog output
- Temperature regulator
- High speed counter
- Simple motion module (SSCNETIII/H compatible)
- Pulse train output (for 1 axis)
- CC-Link interface
- CC-Link master
- AnyWireASLINK master

Up to 16 module including input modules, output modules, intelligent function modules, and bus conversion modules (excluding some products) can be connected to one FX5U CPU module.

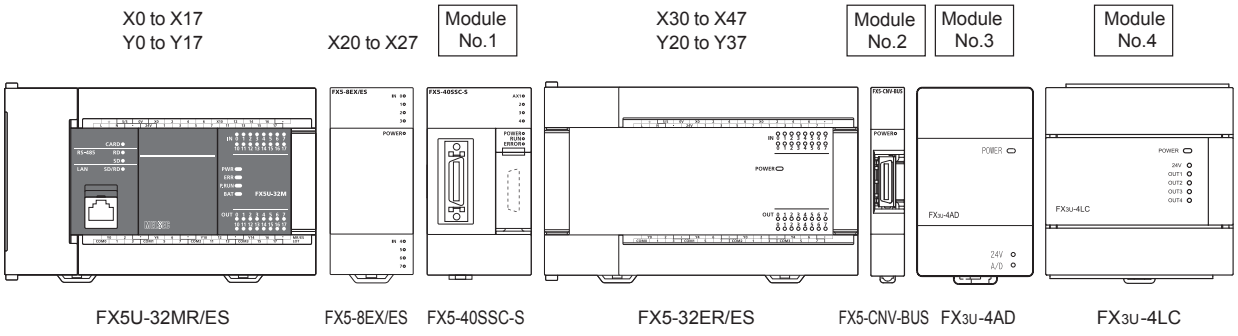
The FROM/TO instructions are expressed in the following manner.



- The buffer memory for the designated module number is read out.
  The drive input turns ON, and reading is executed.
  Reading is not executed when the drive input is OFF, and the data at the transfer destination does not change.
- The data is written to the buffer memory for the designated module number.
- The drive input turns ON, and writing is executed. Reading is not executed when the drive input is OFF, and the data at the transfer destination does not change.

◎ Module number

- Up to 16 modules including input modules, output modules, intelligent function modules, and bus conversion modules (excluding some products) can be connected to one FX5U CPU module.
  Module numbers are assigned to identify with which intelligent function module communication is performed.

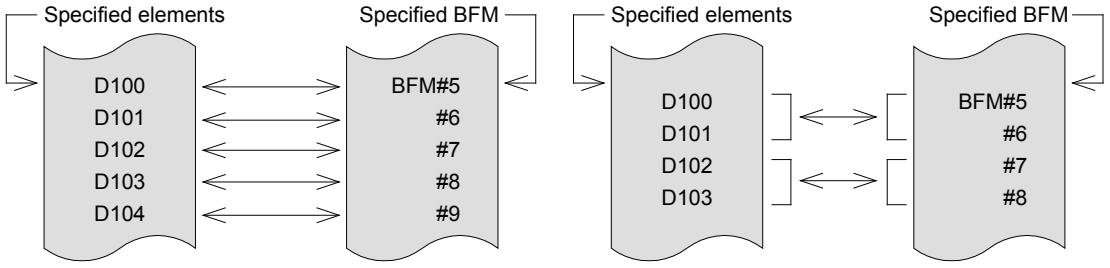


- The intelligent function module is connected with the extension cable on the right side of the PLC in the same manner as the input module, output module and input/output module.
  At this time, I/O numbers are assigned to the input module, output module and I/O module, and a module number is automatically assigned to the intelligent function module.
- The module number is a serial number between No. 1 (H01) and No. 16 (H10) that is assigned in order from the PLC CPU module.
- Each intelligent function module occupies eight input/output points (either input or output can be counted), but the I/O numbers are not assigned.
  The number of PLC input/output points are expressed with the following expression when the intelligent function module is connected.
  Number of input/output points = 256 - number of occupied points (8 points) x number of intelligent function modules[*1]
  *1: Depending on the intelligent function module, more than eight points may be occupied.
- For the intelligent function module, the 5 V power is supplied from the CPU module, power built-in I/O module or extended power module, so the total current consumption must be less than the specified value.

◎ Transfer origin, transfer destination

- The buffer memory number is input in the transfer origin for the FROM instruction and in the transfer destination for the TO instruction.
  The buffer memory has 16 bits. The details and number of points will vary according to the intelligent function module.
- The PLC word device (K2M10, K4X0, etc., including the bit device nibble designation) is entered in the transfer destination for the FROM instruction and in the transfer origin for the TO instruction.
- Some buffer memories have 32 bit content. In this case, D must be attached to create a 32-bit instruction.

◎ Number of transfer points



In the case of a 16-bit instruction with n=5

In the case of a 32-bit instruction with n=2

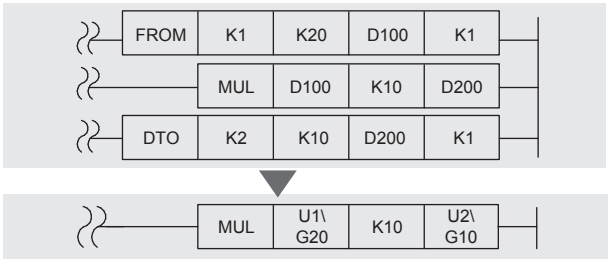
The number of transfer points is designated with n. The 16-bit instruction's n=2 and the 32-bit instructions n=1 have the same significance.

Handy tip!

Using intelligent function module's data directly in applied instruction

By using the module access device, the intelligent function module data can be handled without using FROM/TO instructions.
The module access device is designated with U [intelligent function module's module No.] \ [buffer memory address].

Example: Multiply the intelligent function module No. 1 buffer memory (BFM) #20 data by 10, and write into the intelligent function module No. 2 buffer memory (BFM) #10 and #11.



Chapter 14

IMPLEMENTATION PROCEDURES OF PLC

Anyone can use PLC …

Recent PLCs have many handy functions.
Measures have been taken so that even beginner engineers, who are unfamiliar with sequence design, can use the PLC.
It's important to try using the PLC first. The implementation procedures are explained in this chapter.

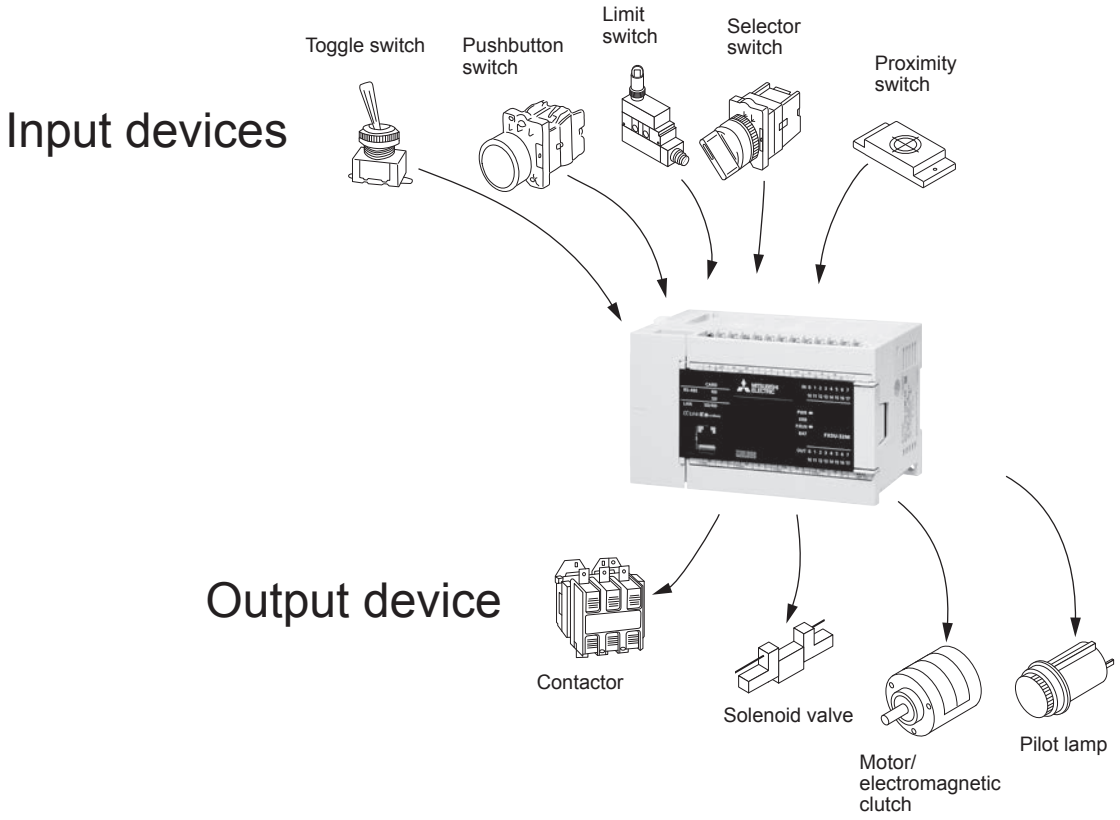
In this chapter…

The procedures follow the steps for replacing a conventional relay panel type sequence circuit with a PLC.
The steps are basically the same as when using the PLC from the start. Gain experience, and learn how to realize the PLC's capabilities to the maximum.

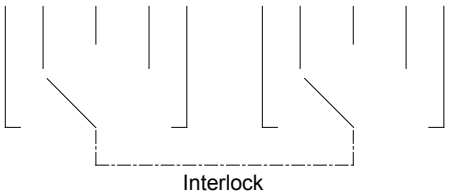
14.1 Selecting a model

14.1.1 Counting the number of input/output points

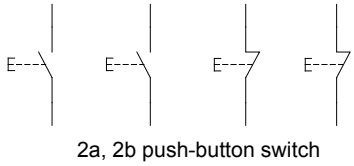
When selecting the PLC, the number of input points and output points must be counted to clarify the scale of the PLC.



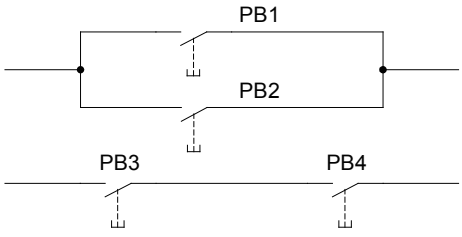
The number of input points does not always match the number of input devices.



If there is one 2-step 5-notch select switch (rotary switch) as shown on the left, the PLC input will be 5 points.



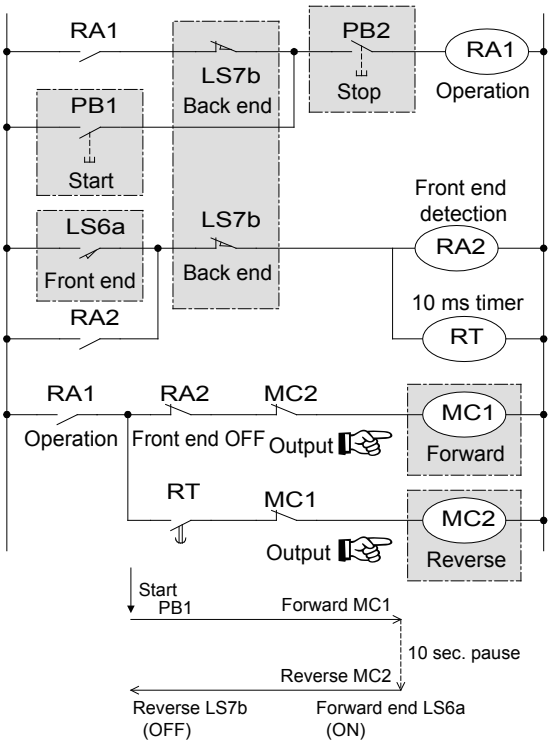
When a pushbutton switch has 2a, 2b contacts as shown on the left, the PLC needs one input for the 1a or 1b contact.



An input switch always connected in parallel or an input switch used as a serial connection can each be handled as one input.

14.1.2 Determining the number of input/output points from the sequence diagram

The following figure shows part of a sequence related to the conveyor's bidirectional control.



(1) When the start button PB1 is pressed, the operation output RA1 continues self-hold operation until the stop button PB2 is pressed or the back end is reached.

(2) When the front end limit switch LS6a turns ON, the front end detection RA2 operates until the back end limit switch LS7b turns OFF, and the timer RT contact activates 10 seconds later.

(3) When the operation output RA1 activates, the advance output MC1 operates until the front end detection RA2 activates.

(4) 10 seconds after the front end is reached and the MC1 is de-energized, the retract output MC2 will energize. When the back end is reached, the timer RT is reset, and the retract output MC2 is de-energized.

(5) In this example, four input points and two output points are required for the PLC.
The PLC's internal relays and timers are used for the relays RA1 and RA2 and for the timer RT, etc.

Reference

The FX5U and FX5UC can handle up to a total of 256 input/output points*1 by using together with the I/O module.

*1: Maximum 512 points including remote I/O for CC-Link and AnyWireASLINK

[FX5U Lineup]

FX5U CPU module

Type	Input	Output	No. of extension block points
32-point type	16 points	16 points	Max. 512 points (Including remote I/O for CC-Link and AnyWireASLINK)
64-point type	32 points	32 points	
80-point type	40 points	40 points	

I/O module (extended cable type)

Type	Input	Output
8-point input dedicated	8 points	—
8-point output dedicated	—	8 points
16-point input dedicated	16 points	—
16-point output dedicated	—	16 points
32-point type	16 points	16 points

[FX5UC Lineup]

FX5UC CPU module

Type	Input	Output	No. of extension block points
32-point type	16 points	16 points	Max. 512 points (Including remote I/O for CC-Link and AnyWireASLINK)
64-point type	32 points	32 points	
96-point type	48 points	48 points	

I/O module (extended connector type)

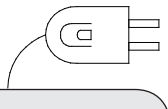
Type	Input	Output
16-point input dedicated	16 points	—
16-point output dedicated	—	16 points
32-point input dedicated	32 points	—
32-point output dedicated	—	32 points
32-point type	16 points	16 points

I/O module (extended cable type)

Type	Input	Output
8-point input dedicated	8 points	—
8-point output dedicated	—	8 points
16-point input dedicated	16 points	—
16-point output dedicated	—	16 points

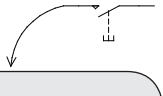
14.1.3 Selecting the PLC type

What is the power voltage?



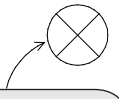
Typically, the power voltage for the PLC is common with the 100/200 V AC system. A 24 V DC power type is used for special applications such as a battery cart or emergency power.

What is the input/output format?



A screw-type terminal block is typically used. The block has connectors and is compact, so various devices can be downsized.

What is the output format?



The relay output type that can be used commonly for the AC load and DC load is often used. A transistor output available as a long-life type or no-contact type is used for the DC load.

For information on the specifications available in each PLC series, refer to the individual catalogs.

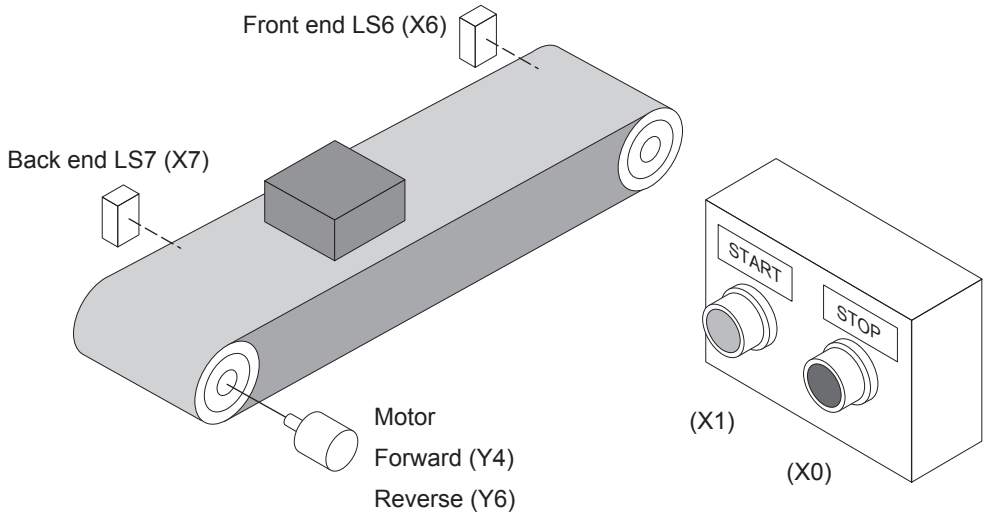
14.2 Design setup

14.2.1 Assigning the input/output devices

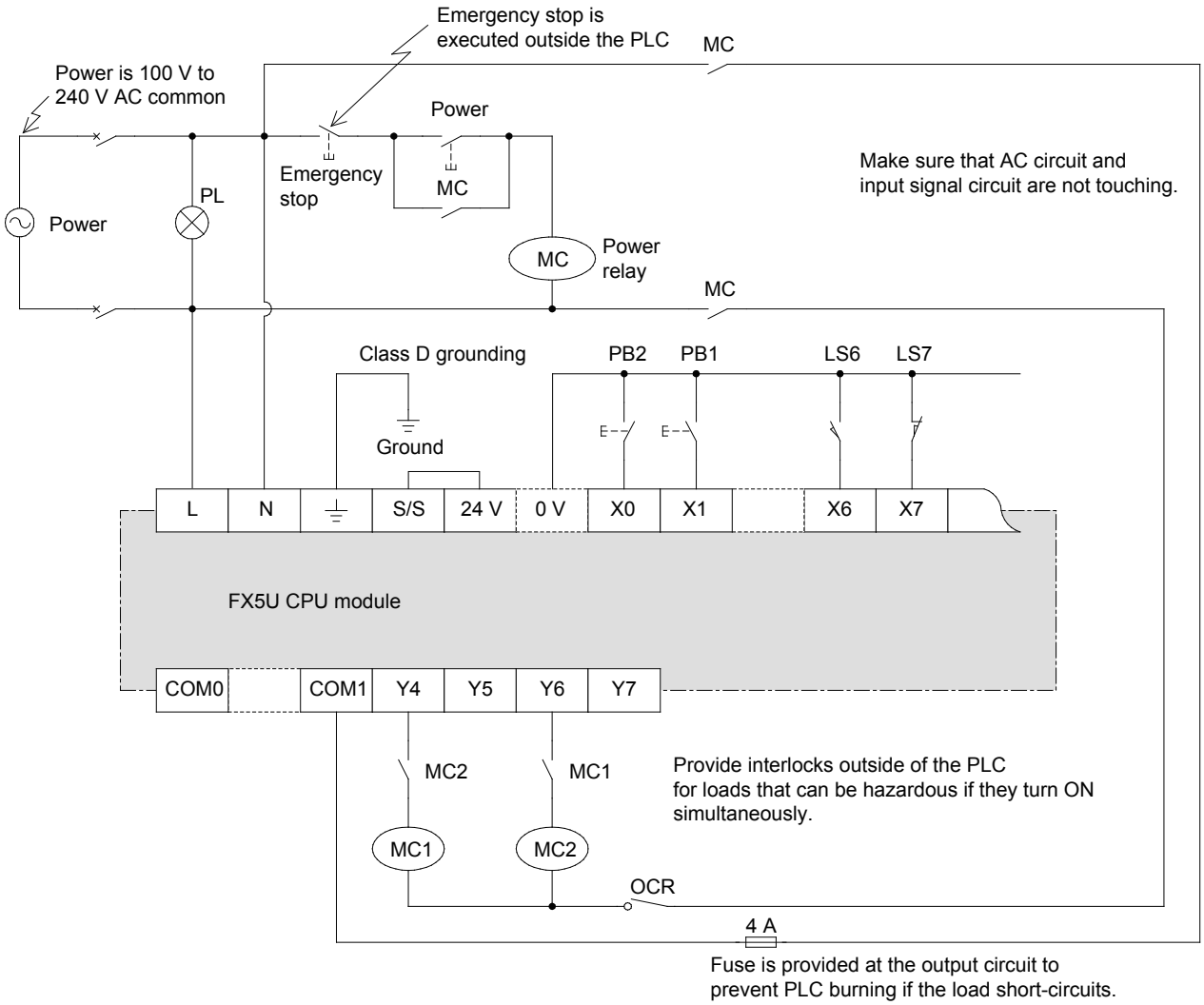
The type of PLC to be used was selected in the previous section.
Next, clarify the correspondence of the input/output device and PLC input/output relay numbers.
After this step, the input/output wiring diagram and sequence explained later can be designed.

Input devices		Input relay No.
Start button	PB1	X1
Stop button	PB2	X0
Front end limit	LS6	X6
Back end limit	LS7	X7

Output device		Output relay No.
Advance output	MC1	Y4
Retract output	MC2	Y6

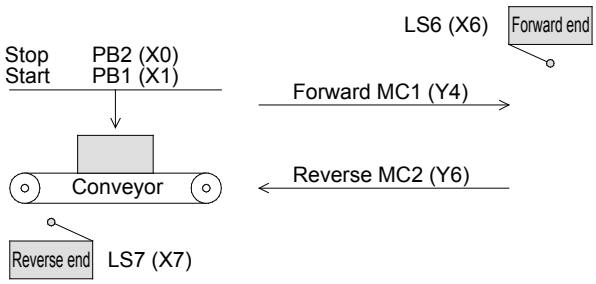


14.2.2 Creating an external connection diagram



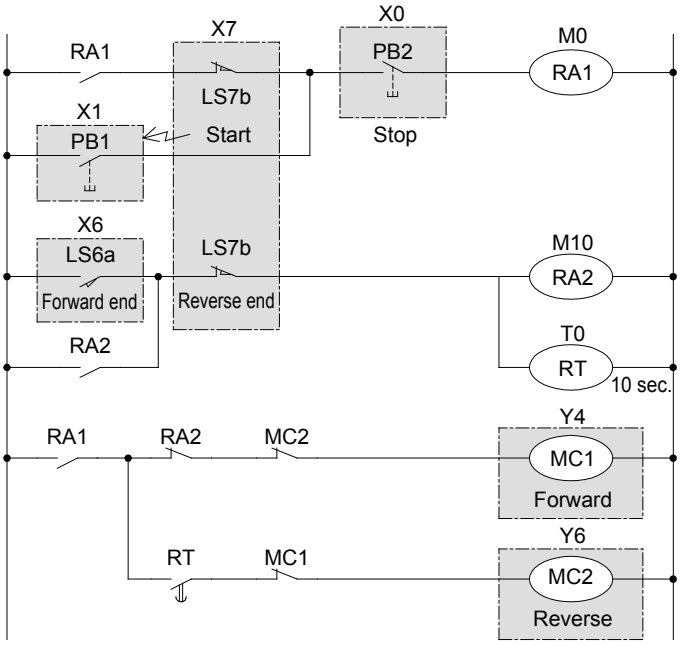
- When using an I/O module, the input/output numbers are assigned sequentially starting from that closest to the CPU module.

14.2.3 Deciding the device numbers



The following sequence circuit expresses the movement of a conveyor as shown on the left.
The limit switch LS6 N.C. contact and the LS7 N.C. contact are de-energized at the conveyor's forward end and reverse end.

In addition to the input/output relays, the timer RT and controller relays RA1 and RS2 are assigned to T0, M0 and M10.

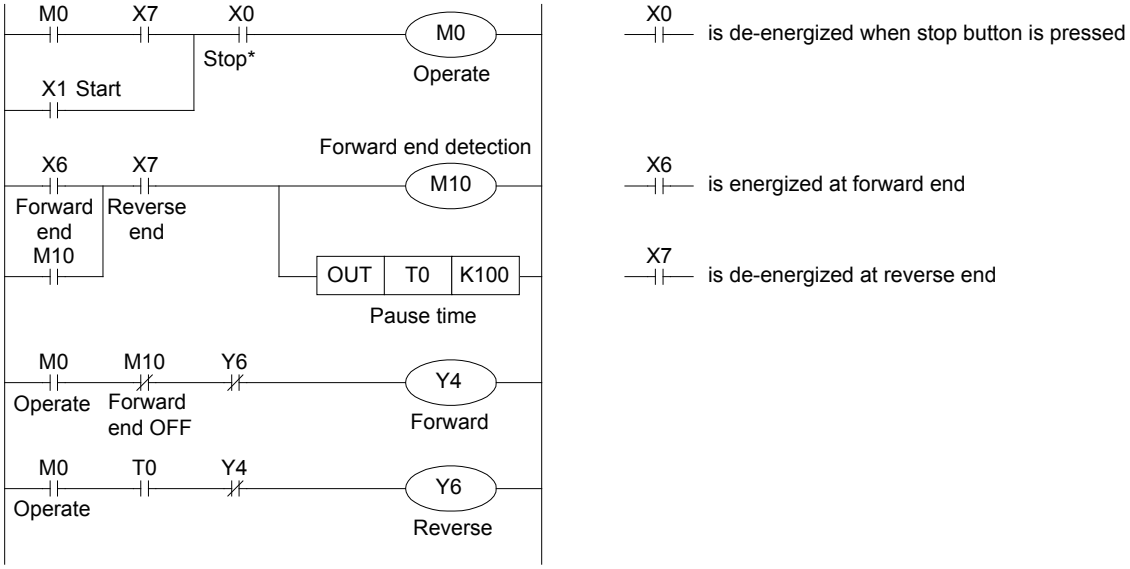


Reference

Designing the sequence circuit

The sequence circuit in the PLC can be designed with basically the same procedures as the sequence circuit in the relay panel. However, in this example the circuit for the section excluding the external connection configured by the input/output devices is designed.

14.2.4 Upgrading to the PLC internal circuit

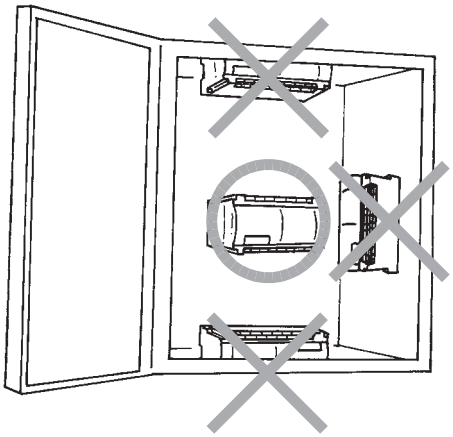


* A N.C. contact (normally closed contact) is used for the stop button PB2, so the N.O. contact (normally open contact) is used on the sequence circuit.

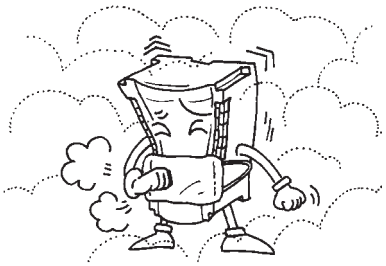
<<Explanation of operation>>

- When the start button PB1 is pressed, the input X1 operates and the operation output M0 starts the self-hold operation. When the stop button PB2 (N.C. contact) is pressed, the N.O. contact of the input X0 is de-energized, and the self-hold of the operation output M0 is released. The self-hold is also released when the reverse end is reached.
- When X6 is energized at the forward end, the forward end detection M10 starts self-holding, and continues until the X7 N.C. contact is de-energized at the reverse end.
- When the start button PB1 is pressed and the operation output M8 operates, the advance output Y4 operates until the forward end detection M10 N.C. contact is de-energized.
- After the front-end detection M10 operates and the timer T0 reaches the time-up, the retract output Y6 operates, and continues to operate until the reverse end is reached and the timer T0 contact is de-energized.

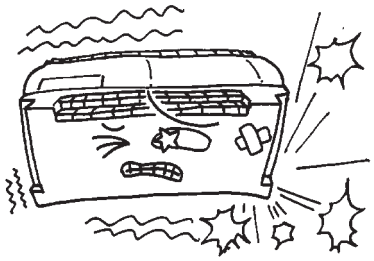
14.3 Installation and wiring work



The PLC can be directly mounted onto the control panel using the mounting holes on the four corners, or directly mounted onto a DIN rail.
To prevent the temperature from rising, do not mount onto the floor or ceiling.
Provide a space of 50 mm or more around the PLC to ensure air circulation.



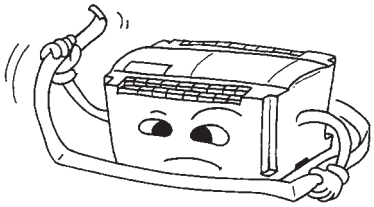
Do not install directly in a place where there is dust, oil mist, conductive gas or corrosive gas.



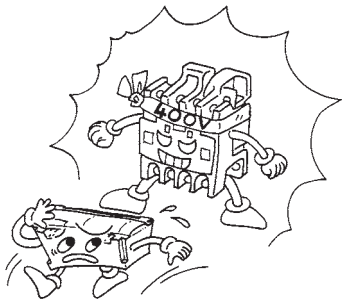
Do not install directly in a place subject to vibration or impact.

Do not install near a heating element, where the module could be subject to direct sunlight, dew condensation, rain or wind.

Pay special caution during drilling or wiring work so that cutting chips or wire scraps do not get into the PLC.

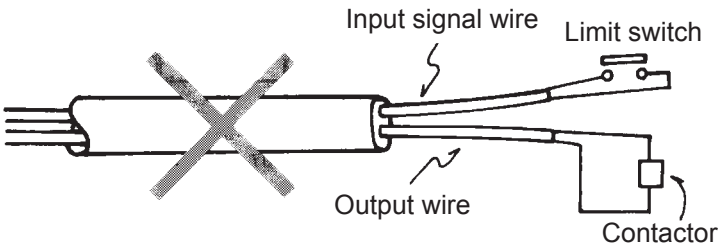


A dust-proof sheet is wrapped around the exterior of the PLC. When the work is completed, remove the sheet to prevent heating.



Separate the module as far away as possible from high-voltage wires, high-voltage devices and power devices.

Do not pass the input signal wire and output wire through the same cable as other power wires, and do not bind them.



14.4 Operation

14.4.1 Preparing the program

- Is programming cable connected?
- Turn OFF the PLC RUN input.
- Turn ON the PLC power.

Writing the program

- Operations for writing the program

Set the cursor to position to write and click. → Select symbol from function key guidance and click. → Input element number from keyboard.

→ After program input ends, click [Convert] on function key guidance.

- Transferring the program

Click "Online" on the menu bar. → Click "Write to PLC". → Click [Parameter + Program], and then click [Execute].

- Printing the program

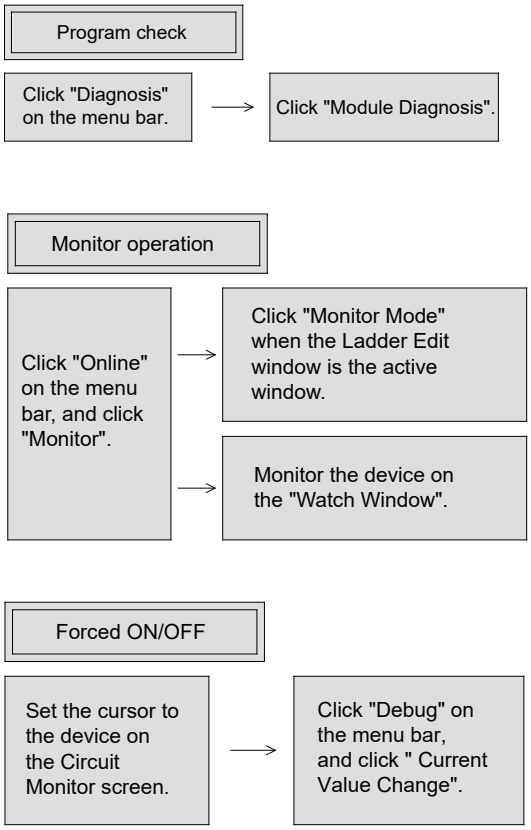
Click "Project" on the menu bar. → Select the paper and printer from "Printer Settings". → Click "Print", select the items to click, and then click [Print].

- Saving the program

Click "Project" on the menu bar. → Click "Save" or "Save As".
"Save As" appears for a newly created program.

14.4.2 Trial operation and adjustment

The PLC starts operation when the PLC RUN input turns ON.
The roles of the personal computer's engineering tool, etc., in the test operation stage are explained below.



- If there is an error in the program, the ERR display on the front of the PLC will flash or light when the PLC RUN input is turned ON, and the PLC may not RUN. The step No. in which the error occurred will display when the steps shown on the left are taken. (The PLC RUN input is ON or OFF)
- The error code is written into the special register SD0, and the cause of the error can be identified by monitoring this register.
- Each contact and coil ON/OFF status can be inspected while displaying the circuit.
- The ON/OFF status of X, Y, M, S,T and C can be inspected by designating the device No.
  The current value is also displayed for T and C.
- Forced ON/OFF can be performed from the personal computer by designating the device No.
  When the PLC is running, this is valid for output relays and internal relays being driven with the self-hold operation or set instruction.
  When forcing ON or OFF while the PLC is stopped, ON and OFF are valid even if there is no self-hold circuit.

Important

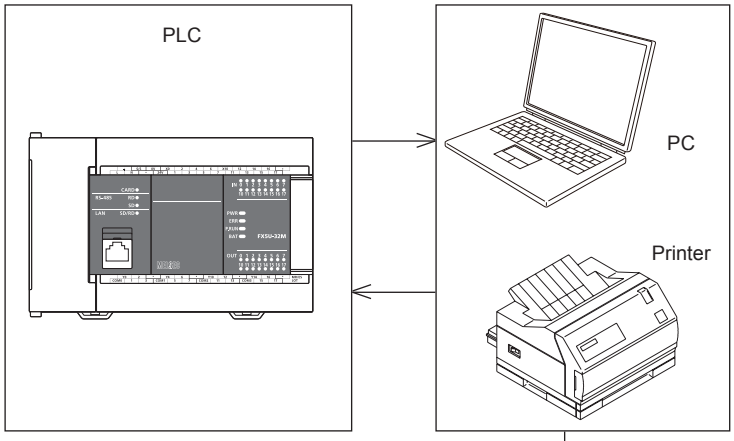
Checking the wiring

The following type of incorrect wiring will damage the PLC. Always carry out a preliminary inspection before turning the power ON.

- Power terminal : Is the power connected to the L and N terminals ?
- Input terminal : Are the 24 V DC series terminals contacting the 100 V AC or 200 V AC?
- Output terminal : Is the load short-circuited?

14.5 Basics of maintenance

14.5.1 Saving and documenting the program

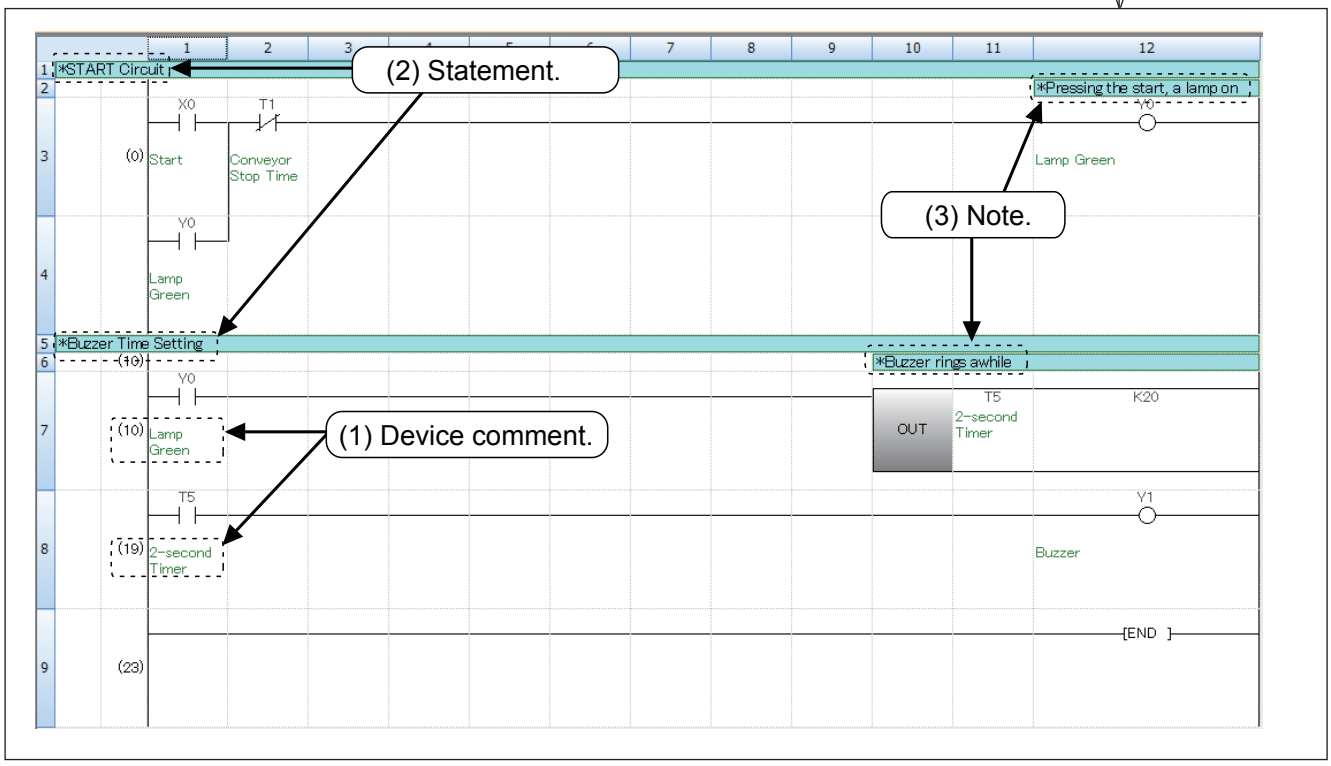


<<Saving the program>>

Saving the program on a USB memory or hard disk (HD) using an engineering tool such as a personal computer is recommended.

<<Documentation>>

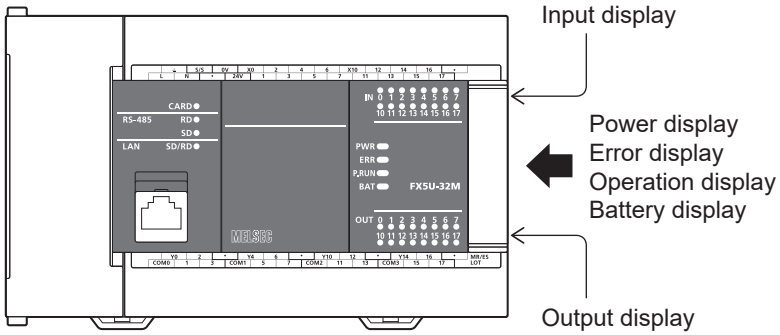
Comments can be added to a program created with an engineering tool, such as a personal computer, and then printed with the printer. This allows the program documentation to be created at the same time as the program.



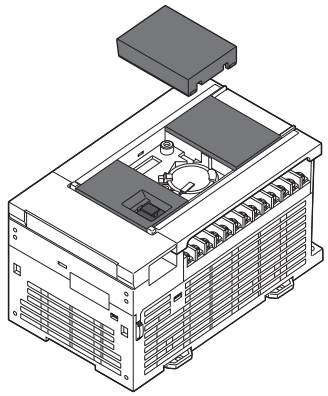
14.5.2 Troubleshooting

Trouble will rarely occur if the input/output wires are correctly connected to the PLC, and the PLC is installed in an appropriate environment. However, the environment is not always ideal, and troubles can occur in the handling of the PLC.
The PLC has various displays and self-diagnosis functions to simplify troubleshooting.

Power display "PWR"	Turns on when voltage is applied on the PLC and the control power in the PLC is operating. Turns off when the sensor power is overloaded.
Operation display "P.RUN"	Turns on when the PLC is in the operation status. If an error occurs, this display turns off and the PLC output turns "OFF". (The output holds the current state when the special relay SM8033 is driven.)
Battery display "BAT"	This is valid when the optional battery is used and the LED display at battery error is set to "Display" with the parameters. This display turns on when the battery voltage drops abnormally and when voltage is applied on the PLC.
Error display "ERR"	This display turns on when an error occurs. This display flashes when the program or parameters are not written in, when the parameters are set incorrectly, or when a watch dog timer error occurs. Refer to the following section "Error Codes" for details on the errors.
Input display Output display	The ON/OFF status of the input signal switches and the output relay drive status are displayed.



14.5.3 Periodic Inspections



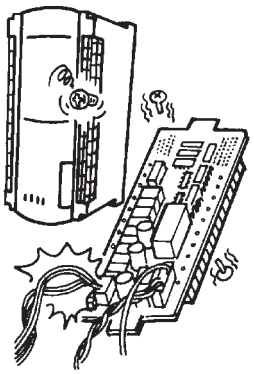
The PLC does not contain any consumable parts that could cause a short-term life.
When using the optional battery, it is recommended to replace it periodically with the procedures given in the instruction manual.
For the output relay, caution must be paid to the life when the relay operates at an abnormally high frequency or when switching a large capacity load.

In addition, pay attention to the following points when inspecting the other devices.

Has the temperature in the panel increased abnormally because the module is near a heating element or in direct sunlight, etc.?

Has dirt or conductive dust entered the panel.?

Are any of the terminals loose or rusted? Are any of the wires damaged, etc.?



Contact capacity

Resistance load	2 A or less

- Circuit voltage is 250 V AC or less, 30 V DC or less
- When the DC conductive load does not have a built-in protection circuit, an external rectifying diode must be installed.

Guide to contact life

Load capacity (VA)	Contact life (10,000 times)
20	300
35	100
80	20

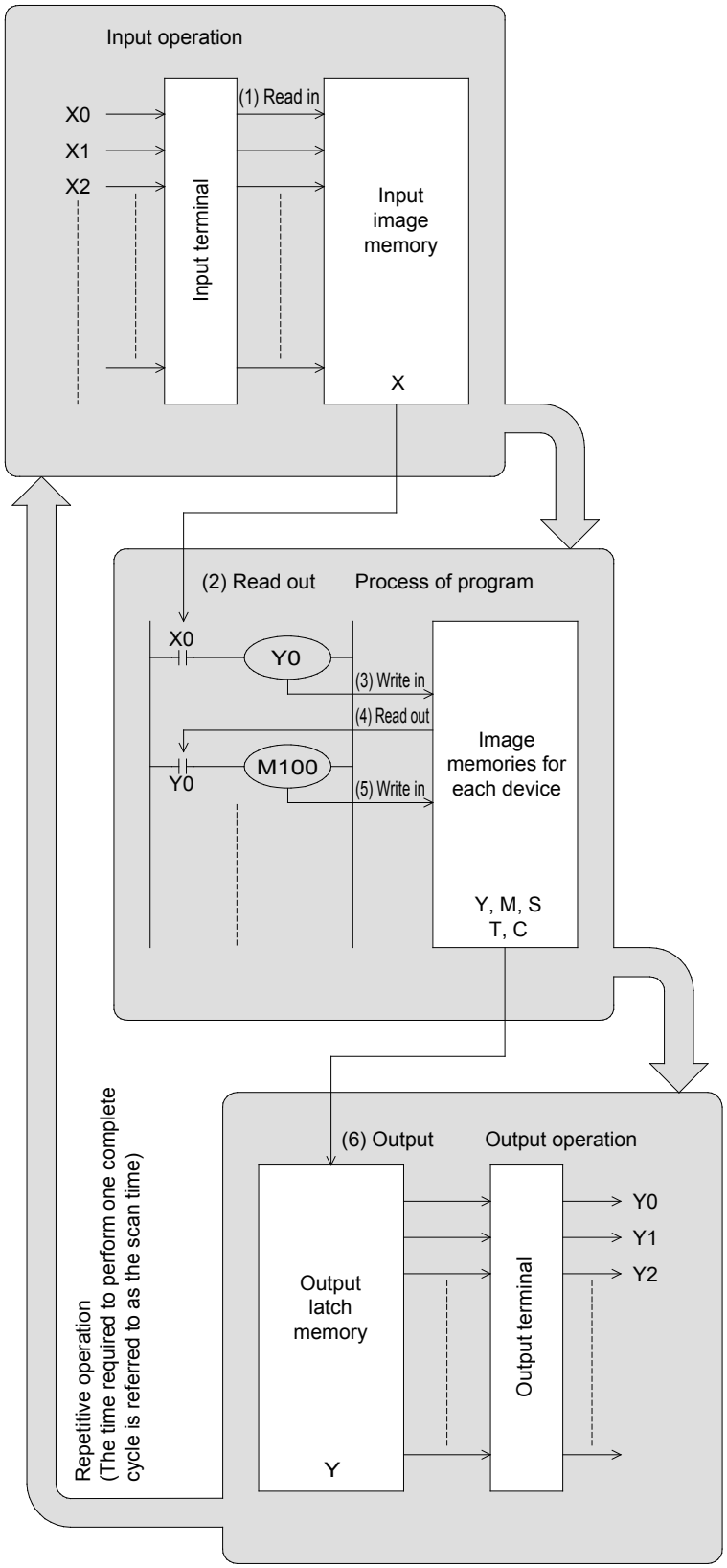
Chapter 15
SUMMARY OF THE POINTS

There are differences in the operating principles between PLCs and relay boards. However, there are many users who do not recognize these differences. To fully master PLC programming, it is essential to understand the differences.

In this chapter…

The operating principles of PLCs will be summarized, along with Internal relays, timers, and the role of the battery.
After reviewing all topics and confirming your knowledge of the points that have been covered, you will take the first step toward becoming a professional sequence circuit designer.

15.1 Input/output process for the PLC



● Input operation

Prior to execution of a program, the PLC reads all ON/OFF statuses of the input terminals into the input image memory. If an input changes its status during the execution of a program, the input image memory does not change its contents at that time. Instead, the change will be read in the next input process cycle.

● Process of program

The PLC reads the ON/OFF statuses of required elements from the input image memory or other element's memory, in accordance with the contents of the instructions stored in the program memory.
Hence, the image memory of each element can sequentially change its content in accordance with the progress of the program.

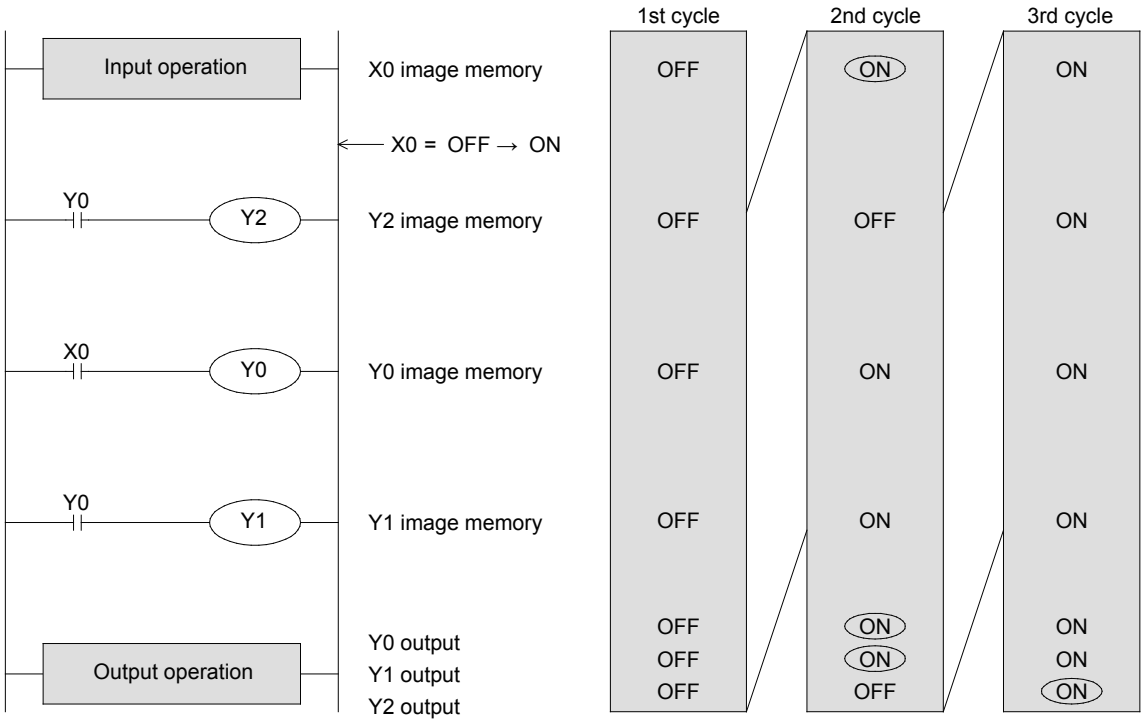
● Output operation

When all instructions have been executed, the PLC transfers the ON/OFF statuses of outputs Y to the output latch memory, which represent the physical outputs.

The above type of method is called the batch input/output method (or refresh method).

15.2 Response delay for Input/Output

There are not only electrical delays from input filters (approx. 10 ms) and mechanical response delays from output relays (approx. 10 ms), but there is also a response delay due to the affect of the scan time.
For example, assume that input X0 changes from OFF to ON just after the input process is finished, as shown in the sequence circuit below. (Note: the input switch changes to ON approximately 10 ms before the process is monitored.)



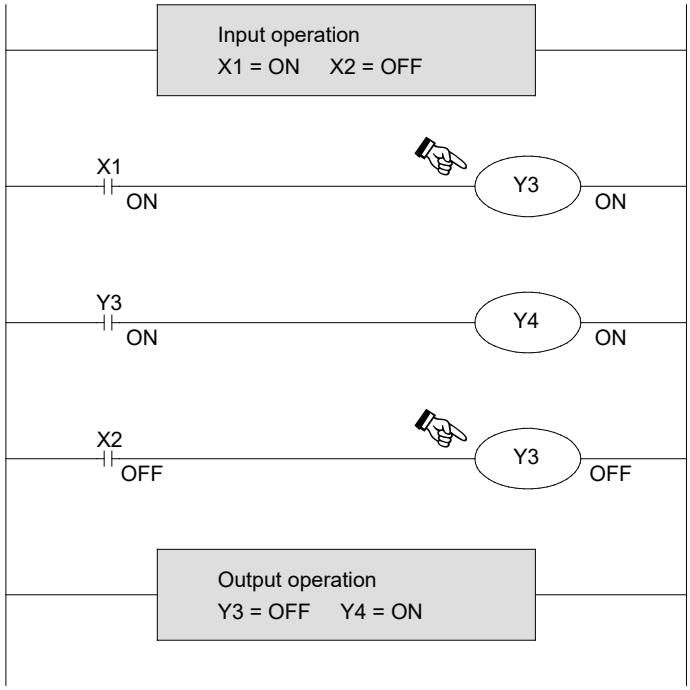
As shown above, Y0 and Y1 may have up to a 2-cycle response delay. (The output contact will be ON approximately 10 ms later.)
Y2 is energized 1 cycle later, since the contact Y0 which drives Y2 is programmed before X0.

Reference

Countermeasures against I/O response delay

There is an instruction (Refresh REF) which can perform input or output processing during the program execution. For transistor type output terminals, the response relay for the output is 0.2 ms or less. For inputs, the input filter's response delay can be shortened with the parameters.

15.3 Dual output operation



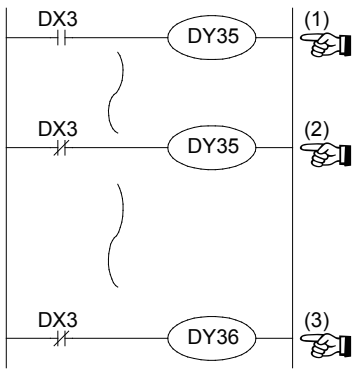
In this example, assume that coil Y3 is used in multiple locations.
Also, assume that X1 = ON and X2 = OFF.
For the first Y3, the image memory is set to ON because X1 is ON, hence output Y4 is also ON.
However, for the second Y3, the image memory is changed to OFF since input X2 is OFF.
Therefore, the output is in fact, Y3 = OFF and Y4 = ON.

As previously described, if dual output (double coil) is performed, the latter content overrides the former.

Reference

Direct I/O Mode

The PLC has a direct access input (DX) and direct access output (DY) that performs the input process or output process simultaneously with the execution of the instruction.
In this case, the output responses for inputs become faster, but the action for dual output is different.
In addition, for the following circuit, DY35 and DY36 may change to ON at the same time.



Assume that DX3 = ON when executing this circuit. Thus, DY35 is energized.

If DX3 is not changed until executing this line of code, DY35 may become de-energized. Hence, DY35 can be ON or OFF, which may cause problems for the dual output during the scan time.

If the input DX3 is changed from ON to OFF before executing this line, DY36 will be energized. If DX003 line (2) is turned OFF then both DY35 and DY36 will turn to ON.

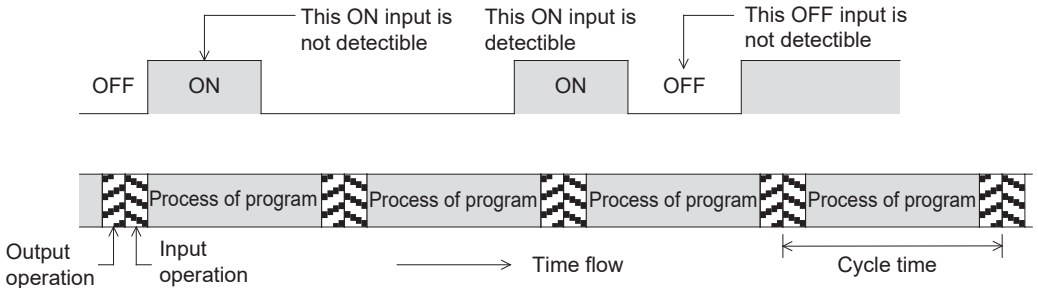
15.4 No limitation on the number of contacts

Since the PLC can use the contents of image memories for each element as many times as needed, there is no limitation to the numbers of N.O. and N.C. contacts.
However, there is a limitation on the program capacity.
The number of serial contacts and parallel contacts is not limited, but circuit display and printer printing of engineering tool such as a personal computer, etc., may be limited.

- At the maximum, 11 contacts and 1 coil (in the case for timers or counters, 9 contacts and 1 coil) can be included in a single line on the GPP.
- For easier viewing on PC screen, it is recommended to set the display so that there is no wrapping.

Reference

Narrow pulses are not detectible

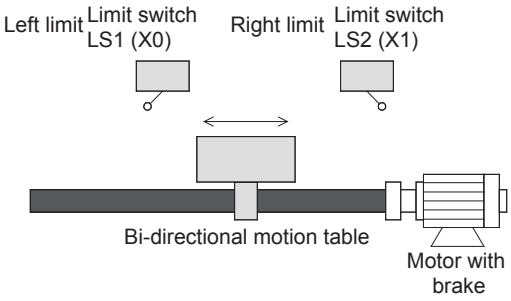


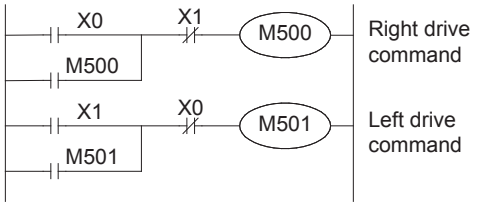
In principle, the widths of input pulses (ON and OFF inputs) for a PLC must be longer than the cycle time of the PLC.
When the response delay of the input filter, 10 ms, is taken into account, the time durations for ON and OFF inputs must be 20 ms at the shortest, if the cycle time is 10 ms.
Hence, input pulses with frequencies of higher than 25 Hz (= 1000 / (20 + 20)) cannot be read normally. However if a special function or application instruction is used, these signals can be detected.

15.5 Role of the latch device

The FX5U CPU module does not have a battery inside when shipped from the factory.
The optional battery is used to hold (latch) the clock data in a power failure or to increase the latched devices.
The parameters must be set when using a battery.

How to use the internal relay (latch)



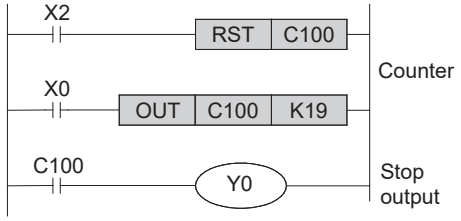


In some cases, it is required to move the positioning table in the same direction after power failure.

X0=ON (Left limit)
→ M500 (Right drive command) = ON
→ Power failure
→ Positioning table stops halfway
→ Restart (M500 = ON)
→ X1 = ON (Right limit)
→ M500 = OFF, M501 = ON
→ Left drive

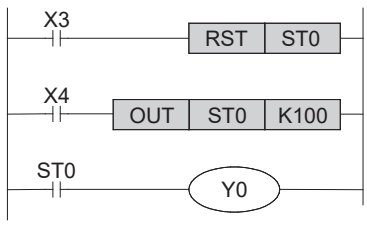
The latched internal relay may be called a latched relay.

How to use counters



This counter counts the activation of the left limit switch shown above, and stops the table after counting 19 times.
In this case, the counter keeps its counting value even if a power failure occurs during the counting.

How to use retentive timers

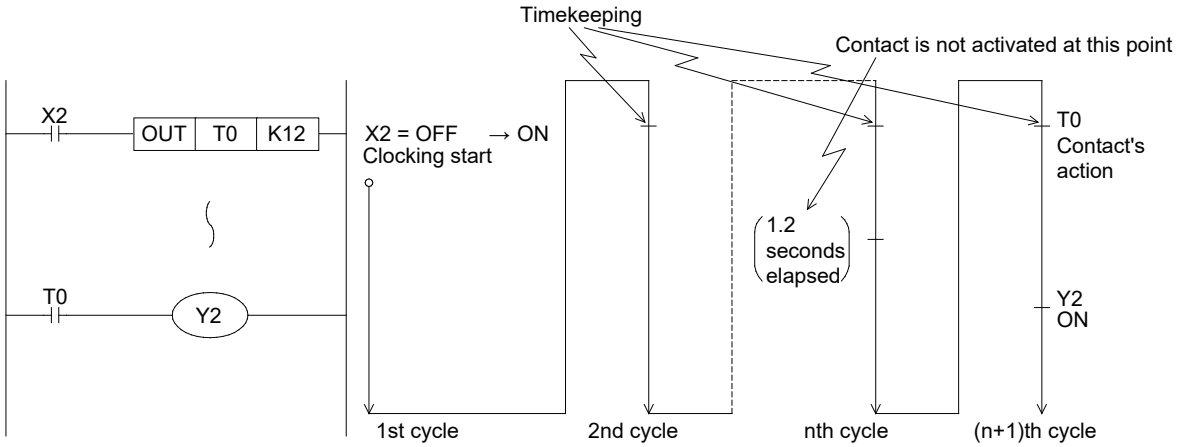


The timer ST0 starts timing at the moment X4 changes to ON.
The timer keeps its current value if X4 is set to OFF or if the power is turned off during timing.
When the timer starts again, the timer clock sustains its time value and the output contact ST0 will then be energized.
When X3 is set to ON, the current value of the timer is cleared and reset to 0 and the output contact is set to OFF.

- The device latch range can be changed with the parameters.

15.6 Timers and their accuracy

Each timer in a PLC begins timing when its drive contact is closed, and its output contact is activated after the timer reaches its set value.



Hence, the approximate accuracy of a timer contact can be expressed with the following formula:

$$T^{+T_0}_{-\alpha}$$

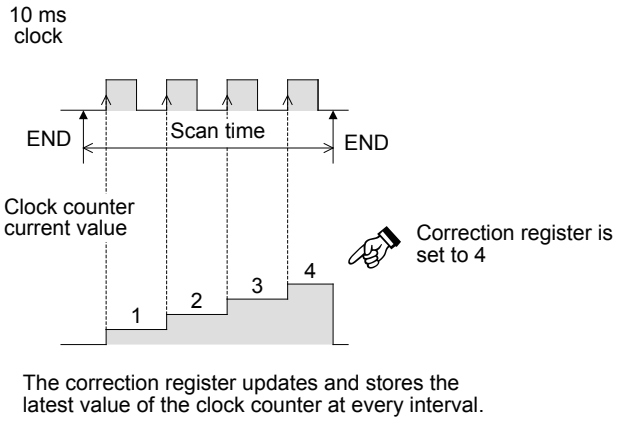
α : 0.001 for 1 ms timer, 0.01 for 10 ms timer, 0.1 for 100 ms timer (seconds)
T : Set time of timer (seconds)
T_0 : Scan time (seconds)

In the worst case, if the timer drive contact is programmed prior to the timer coil, the accuracy becomes $+2T_0$. If the set value of the timer is 0, the output contact will be activated at the next execution of the coil instruction. When the timer setting value is 1, the accuracy above may cause the output contact to activate when the next cycle's coil instruction is executed.

Reference

Details on timer operation

- For 1 ms, 10 ms and 100 ms timers, the time is measured at each execution of the coil instruction, and when it reaches a set value, the output contact is energized. However, the following type of compensation is applied when using a 10 ms timer with a calculation cycle of several 10 ms.



The correction register updates and stores the latest value of the clock counter at every interval.

- There is a timer correction register, which stores the previous scan time, in the PLC.
- When the OUTH T instruction is executed, the value in the correction register is added to the timer's current value register.

MEMO

Appendix 1

In this appendix...

The applied instructions, special internal relays, special data registers, and parameters that have been described in this textbook are arranged in lists.
In addition, the details of the error codes are listed. Refer to these as necessary.

Appendix 1.1 List of instructions

The applied instructions that can be used with the micro-PLC MELSEC iQ-F Series are listed below.
Refer to the MELSEC iQ-F FX5 Programming Manual (Instructions, Standard Functions/Function Blocks).

CPU module application instruction

Classification	Instruction symbol	Function	Compatible CPU module	
			FX5U	FX5UC
Rotation	ROR(P)	16-bit data right rotation	○	○
	RCR(P)	Right rotation with 16-bit data carry	○	○
	ROL(P)	16-bit data left rotation	○	○
	RCL(P)	Left rotation with 16-bit data carry	○	○
	DROR(P)	32-bit data right rotation	○	○
	DRCR(P)	Right rotation with 32-bit data carry	○	○
	DROL(P)	32-bit data left rotation	○	○
	DRCL(P)	Left rotation with bit data carry	○	○
Program branch	CJ(P)	Pointer branch	○	○
	GOEND	Jump to END	○	○
Program execution control	DI	Interrupt disable	○	○
	EI	Interrupt enable	○	○
	DI	Interrupt disable when lower than specified priority	○	○
	IMASK	Interrupt program mask	○	○
	SIMASK	Specified interrupt pointer disable/enable	○	○
	IRET	Return from interrupt program	○	○
	WDT(P)	WDT reset	○	○
Structured instruction	FOR	Executed (n) times between ROM instruction and NEXT instruction	○	○
	NEXT		○	○
	BREAK(P)	FOR to NEXT forced end	○	○
	CALL(P)	Subroutine program call	○	○
	RET	Return from subroutine program	○	○
	SRET		○	○
	XCALL	Subroutine program call	○	○
Data table operation	SFRD(P)	First-in data read from data table	○	○
	POP(P)	Last-in data read from data table	○	○
	SFWR(P)	Data write to data table	○	○
	FINS(P)	Data insertion to data table	○	○
	FDEL(P)	Data delete from data table	○	○
Character string processing	LD$=	Character string comparison LD (S1) = (S2)	○	○
	LD$<>	Character string comparison LD (S1) <> (S2)	○	○
	LD$>	Character string comparison LD (S1) > (S2)	○	○
	LD$<=	Character string comparison LD (S1) <= (S2)	○	○
	LD$<	Character string comparison LD (S1) < (S2)	○	○
	LD$>=	Character string comparison LD (S1) >= (S2)	○	○
	AND$=	Character string comparison AND (S1) = (S2)	○	○
	AND$<>	Character string comparison AND (S1) <> (S2)	○	○
	AND$>	Character string comparison AND (S1) > (S2)	○	○
	AND$<=	Character string comparison AND (S1) <= (S2)	○	○
	AND$<	Character string comparison AND (S1) < (S2)	○	○
	AND$>=	Character string comparison AND (S1) >= (S2)	○	○
	OR$=	Character string comparison OR (S1) = (S2)	○	○
	OR$<>	Character string comparison OR (S1) <> (S2)	○	○
	OR$>	Character string comparison OR (S1) > (S2)	○	○
	OR$<=	Character string comparison OR (S1) <= (S2)	○	○

Classification	Instruction symbol	Function	Compatible CPU module	
			FX5U	FX5UC
Character string processing	OR$<	Character string comparison OR (S1) < (S2)	○	○
	OR$>=	Character string comparison OR (S1) >= (S2)	○	○
	$+(P)	Combination of character strings	○	○
	$MOV(P)	Transfer of character string	○	○
	BINDA(P)(_U)	BIN 16-bit data→Decimal ASCII conversion	○	○
	DBINDA(P)(_U)	BIN 32-bit data→Decimal ASCII conversion	○	○
	ASCI(P)	HEX code data→ASCII conversion	○	○
	STR(P)(_U)	BIN 16-bit data→Character string conversion	○	○
	DSTR(P)(_U)	BIN 32-bit data→Character string conversion	○	○
	ESTR(P)	Single precision actual number→ Character string conversion	○	○
	DESTR(P)		○	○
	LEN(P)	Detection of character string length	○	○
	RIGHT(P)	Extraction from right side of character string	○	○
	LEFT(P)	Extraction from left side of character string	○	○
	MIDR(P)	Extraction of any part from the middle of character string	○	○
	MIDW(P)	Replacement of any part in the middle of character string	○	○
	INSTR(P)	Character string search	○	○
	STRINS(P)	Character string insertion	○	○
	STRDEL(P)	Character string deletion	○	○
Actual number	LDE$=	Single precision actual number comparison LDE (S1) = (S2)	○	○
	LDE$<>	Single precision actual number comparison LDE (S1) <> (S2)	○	○
	LDE$>	Single precision actual number comparison LDE (S1) > (S2)	○	○
	LDE$<=	Single precision actual number comparison LDE (S1) <= (S2)	○	○
	LDE$<	Single precision actual number comparison LDE (S1) > (S2)	○	○
	LDE$>=	Single precision actual number comparison LDE (S1) >= (S2)	○	○
	ANDE$=	Single precision actual number comparison ANDE (S1) = (S2)	○	○
	ANDE$<>	Single precision actual number comparison ANDE (S1) <> (S2)	○	○
	ANDE$>	Single precision actual number comparison ANDE (S1) > (S2)	○	○
	ANDE$<=	Single precision actual number comparison ANDE (S1) <= (S2)	○	○
	ANDE$<	Single precision actual number comparison ANDE (S1) < (S2)	○	○
	ANDE$>=	Single precision actual number comparison ANDE (S1) >= (S2)	○	○
	ORE$=	Single precision actual number comparison ORE (S1) = (S2)	○	○
	ORE$<>	Single precision actual number comparison ORE (S1) <> (S2)	○	○
	ORE$>	Single precision actual number comparison ORE (S1) > (S2)	○	○
	ORE$<=	Single precision actual number comparison ORE (S1) <= (S2)	○	○
	ORE$<	Single precision actual number comparison ORE (S1) < (S2)	○	○
	ORE$>=	Single precision actual number comparison ORE (S1) >= (S2)	○	○
	DECMP(P)	Single precision actual number comparison	○	○
	DEZCP(P)	Binary floating point bandwidth comparison	○	○
	E+(P)	Single precision actual number addition	○	○
	E-(P)	Single precision actual number subtraction	○	○
	DEADD(P)	Single precision actual number addition	○	○
	DESUB(P)	Single precision actual number subtraction	○	○
	E*(P)	Single precision actual number multiplication	○	○
	E/(P)	Single precision actual number division	○	○
	DEMUL(P)	Single precision actual number multiplication	○	○
	DEDIV(P)	Single precision actual number division	○	○
	INT2FLT(P)	Signed BIN 16-bit data→Single precision actual number conversion	○	○
	UINT2FLT(P)	Unsigned BIN 16-bit data→Single precision actual number conversion	○	○
	DINT2FLT(P)	Signed BIN 32-bit data→Single precision actual number conversion	○	○

Classification	Instruction symbol	Function	Compatible CPU module	
			FX5U	FX5UC
Actual number	EVAL(P)	Character string→Single precision actual number conversion	○	○
	DEVAL(P)		○	○
	DEBCD(P)	Binary floating point→Decimal floating point conversion	○	○
	DEBIN(P)	Decimal floating point→Binary floating point conversion	○	○
	ENEG(P)	Reverse of single precision actual number sign	○	○
	DENEG(P)		○	○
	EMOV(P)	Transfer of single precision actual number data	○	○
	DEMOV(P)		○	○
	SIN(P)	Single precision actual number SIN operation	○	○
	DSIN(P)		○	○
	COS(P)	Single precision actual number COS operation	○	○
	DCOS(P)		○	○
	TAN(P)	Single precision actual number TAN operation	○	○
	DTAN(P)		○	○
	ASIN(P)	Single precision actual number SIN^{-1} operation	○	○
	DASIN(P)		○	○
	ACOS(P)	Single precision actual number COS^{-1} Operation	○	○
	DACOS(P)		○	○
	ATAN(P)	Single precision accuracy TAN^{-1} operation	○	○
	DATAN(P)		○	○
	RAD(P)	Single precision actual number angle → Radian conversion	○	○
	DRAD(P)		○	○
	DEG(P)	Single precision actual number radian → Angle conversion	○	○
	DDEG(P)		○	○
	DESQR(P)	Square root of single precision actual number	○	○
	ESQRT(P)		○	○
	EXP(P)	Index operation of single precision actual number	○	○
	DEXP(P)		○	○
	LOG(P)	Inferior logarithm operation of single precision actual number	○	○
	DLOGE(P)		○	○
	POW(P)	Exponentiation operation of single precision actual number	○	○
	LOG10(P)	Common logarithm operation of single precision actual number	○	○
	DLOG10(P)		○	○
	EMAX(P)	Search for maximum value of single precision actual number	○	○
	EMIN(P)	Search for minimum value of single precision actual number	○	○
Random number	RND(P)	Random number generation	○	○
Index register operation	ZPUSH(P)	Collective saving of index register	○	○
	ZPOP(P)	Corrective return of index register	○	○
	ZPUSH(P)	Selection and saving of index register/long index register	○	○
	ZPOP(P)	Selection and return of index register/long index register	○	○
Data control	LIMIT(P)(_U)	BIN 16-bit data upper-/lower-limit control	○	○
	DLIMIT(P)(_U)	BIN 32-bit data upper-/lower-limit control	○	○
	BAND(P)(_U)	BIN 16-bit data dead band control	○	○
	DBAND(P)(_U)	BIN 32-bit data dead band control	○	○
	ZONE(P)(_U)	BIN 16-bit data zone control	○	○
	DZONE(P)(_U)	BIN 32-bit data zone control	○	○

Classification	Instruction symbol	Function	Compatible CPU module	
			FX5U	FX5UC
Data control	SCL(P)(_U)	BIN 16-bit unit scaling (point-specific coordinate data)	○	○
	DSCL(P)(_U)	BIN 32-bit unit scaling (point-specific coordinate data)	○	○
	SCL2(P)(_U)	BIN 16-bit unit scaling (X-/Y-specific coordinate data)	○	○
	DSCL2(P)(_U)	BIN 32-bit unit scaling (X-/Y-specific coordinate data)	○	○
Special timer	TTMR	Teaching timer	○	○
	STMR	Special function timer	○	○
Special counter	UDCNTF	Signed 32-bit up/down counter	○	○
Shortcut control	ROTC	Rotary table shortcut control	○	○
Inclination signal	RAMPF	Control inclination signal	○	○
Pulse system	SPD	Measurement of BIN 16-bit pulse density	○	○
	DSPD	Measurement of BIN 32-bit pulse density	○	○
	PLSY	BIN 16-bit pulse output	○	○
	DPLSY	BIN 32-bit pulse output	○	○
	PWM	BIN 16 pulse width modulation	○	○
	DPWM	BIN 32-bit pulse width modulation	○	○
Matrix input	MTR	Matrix input	○	○
Initial state	IST	Initial state	○	○
Drum sequence	ABSD	BIN 16-bit data absolute method	○	○
	DABSD	BIN 32-bit data absolute method	○	○
	INCD	Relative method	○	○
Check code	CCD(P)	Check code	○	○
Data processing instruction	SERMM(P)	Data processing instruction	○	○
	DSERMM(P)	32-bit data search	○	○
	SUM(P)	16-bit data bit check	○	○
	DSUM(P)	32-bit data bit check	○	○
	BON(P)	Bit detection of 16-bit data	○	○
	DBON(P)	Bit detection of 32-bit data	○	○
	MAX(P)(_U)	Search for maximum value of 16-bit data	○	○
	DMAX(P)(_U)	Search for maximum value of 32-bit data	○	○
	MIN(P)(_U)	Search for minimum value of 16-bit data	○	○
	DMIN(P)(_U)	Search for minimum value of 32-bit data	○	○
	SORTTBL(_U)	16-bit data sort	○	○
	SORTTBL2(_U)	16-bit data alignment 2	○	○
	DSORTTBL2(_U)	32-bit data alignment 2	○	○
	WSUM(P)(_U)	16-bit data total value calculation	○	○
	DWSUM(P)(_U)	32-bit data total value calculation	○	○
	MEAN(P)(_U)	16-bit data average value calculation	○	○
	DMEAN(P)(_U)	32-bit data average value calculation	○	○
	SQRT(P)	Calculation of 16-bit square root	○	○
	DSQRT(P)	Calculation of 32-bit square root	○	○
	CRC(P)	CRC calculation	○	○
Indirect address read	ADRSET(P)	Indirect address read	○	○

Classification	Instruction symbol	Function	Compatible CPU module	
			FX5U	FX5UC
For clock	TRD(P)	Clock data read	○	○
	TWR(P)	Clock data write	○	○
	TADD(P)	Addition of clock data	○	○
	TSUB(P)	Subtraction of clock data	○	○
	HTOS(P)	16-bit data conversion of time data (hour/minute/second→second)	○	○
	DHTOS(P)	32-bit data conversion of time data (hour/minute/second→second)	○	○
	STOH(P)	16-bit data conversion of time data (second→hour/minute/second	○	○
	DSTOH(P)	32-bit data conversion of time data (second→hour/minute/second)	○	○
	LDDT$=	Date comparison LDDT (S1) = (S2)	○	○
	LDDT$<>	Date comparison LDDT (S1) <> (S2)	○	○
	LDDT$>	Date comparison LDDT (S1) > (S2)	○	○
	LDDT$<=	Date comparison LDDT (S1) <= (S2)	○	○
	LDDT$<	Date comparison LDDT (S1) < (S2)	○	○
	LDDT$>=	Date comparison LDDT (S1) >= (S2)	○	○
	ANDDT$=	Date comparison ANDDT (S1) = (S2)	○	○
	ANDDT$<>	Date comparison ANDDT (S1) <> (S2)	○	○
	ANDDT$>	Date comparison ANDDT (S1) > (S2)	○	○
	ANDDT$<=	Date comparison ANDDT (S1) <= (S2)	○	○
	ANDDT$<	Date comparison ANDDT (S1) < (S2)	○	○
	ANDDT$>=	Date comparison ANDDT (S1) >= (S2)	○	○
	ORDT$=	Date comparison ORDT (S1) = (S2)	○	○
	ORDT$<>	Date comparison ORDT (S1) <> (S2)	○	○
	ORDT$>	Date comparison ORDT (S1) > (S2)	○	○
	ORDT$<=	Date comparison ORDT (S1) <= (S2)	○	○
	ORDT$<	Date comparison ORDT (S1) < (S2)	○	○
	ORDT$>=	Date comparison ORDT (S1) >= (S2)	○	○
	LDTM$=	Time comparison LDTM (S1) = (S2)	○	○
	LDTM$<>	Time comparison LDTM (S1) <> (S2)	○	○
	LDTM$>	Time comparison LDTM (S1) > (S2)	○	○
	LDTM$<=	Time comparison LDTM (S1) <= (S2)	○	○
	LDTM$<	Time comparison LDTM (S1) < (S2)	○	○
	LDTM$>=	Time comparison LDTM (S1) >= (S2)	○	○
	ANDTM$=	Time comparison ANDTM (S1) = (S2)	○	○
	ANDTM$<>	Time comparison ANDTM (S1) <> (S2)	○	○
	ANDTM$>	Time comparison ANDTM (S1) > (S2)	○	○
	ANDTM$<=	Time comparison ANDTM (S1) <= (S2)	○	○
	ANDTM$<	Time comparison ANDTM (S1) < (S2)	○	○
	ANDTM$>=	Time comparison ANDTM (S1) >= (S2)	○	○
	ORTM$=	Time comparison ORTM (S1) = (S2)	○	○
	ORTM$<>	Time comparison ORTM (S1) <> (S2)	○	○
	ORTM$>	Time comparison ORTM (S1) > (S2)	○	○
	ORTM$<=	Time comparison ORTM (S1) <= (S2)	○	○
	ORTM$<	Time comparison ORTM (S1) < (S2)	○	○
	ORTM$>=	Time comparison ORTM (S1) >= (S2)	○	○
	TCMP(P)	Clock data comparison	○	○
	TZCP(P)	Clock data bandwidth comparison	○	○

Classification	Instruction symbol	Function	Compatible CPU module	
			FX5U	FX5UC
Timing measurement	DUTY	Timing pulse generation	○	○
	HOURM	Hour meter (BIN 16-bit data)	○	○
	DHOURM	Hour meter (BIN 32-bit data)	○	○
Module access	REF(P)	I/O refresh	○	○
	RFS(P)		○	○
	FROM(P)	Read of 1-word data from other module (16-bit specified)	○	○
	DFROM(P)	Read of 2-word data from other module (16-bit specified)	○	○
	TO(P)	Write of 1-word data from other module (16-bit specified)	○	○
	DTO(P)	Write of 2-word data from other module (16-bit specified)	○	○
	FROMD(P)	Read of 1-word data from other module (32-bit specified)	○	○
	DFROMD(P)	Read of 2-word data from other module (32-bit specified)	○	○
	TOD(P)	Write of 1-word data from other module (32-bit specified)	○	○
	DTOD(P)	Write of 2-word data from other module (32-bit specified)	○	○

Step ladder instruction

Classification	Instruction symbol	Function	Compatible CPU module	
			FX5U	FX5UC
Step ladder	STL	Start of step ladder	○	○
	RETSTL	End of step ladder	○	○

Built-in Ethernet function instruction

Classification	Instruction symbol	Function	Compatible CPU module	
			FX5U	FX5UC
Built-in Ethernet function instruction	SP.SOCOPEN	Connection establishment	○	○
	SP.SOCCLOSE	Connection disconnection	○	○
Socket Communication function	SP.SOCRCV	Read of received data during END processing	○	○
	SP.SOCSND	Data transmission	○	○
	SP.SOCCINF	Read of connection information	○	○
	S(P).SOCRDATA	Read of received data of socket communication	○	○
Communication protocol support function	SP.ECPRTCL	Execution of registration protocol of communication protocol support function	○	○

PID control instruction

Classification	Instruction symbol	Function	Compatible CPU module	
			FX5U	FX5UC
PID control	PID	PID operation	○	○

List of module dedicated instructions

Classification	Instruction symbol	Function	Compatible CPU module	
			FX5U	FX5UC
High speed counter	DHSCS	32-bit data comparison set	○	○
	DHSCR	32-bit comparison reset	○	○
	DHSZ	32-bit data bandwidth comparison	○	○
	HIOEN(P)	Start and stop of 16-bit data high speed input/output function	○	○
	DHIOEN(P)	Start and stop of 32-bit data high speed input/output function	○	○

Classification	Instruction symbol	Function	Compatible CPU module	
			FX5U	FX5UC
High-speed transfer of current value	HCMOV(P)	High-speed transfer of 16-bit data current value	○	○
	DHCMOV(P)	High-speed transfer of 32-bit data current value	○	○
External device communication	RS2	Serial data transfer 2	○	○
Inverter communication	IVCK	Inverter operation monitor	○	○
	IVDR	Inverter operation control	○	○
	IVRD	Inverter parameter read	○	○
	IVWR	Inverter parameter write	○	○
	IVBWR	Inverter parameter batch write	○	○
	IVMC	Multiple commands of inverter	○	○
MODBUS®	ADPRW	MODBUS data read/write	○	○
Communication protocol support function	S(P).CPRTCL	Execution of communication protocol registered by engineering tool	○	○
Positioning	DSZR	Home position return with 16-bit data dog search	○	○
	DDSZR	Home position return with 32-bit data dog search	○	○
	DVIT	16-bit data interrupt positioning	○	○
	DDVIT	32-bit data interrupt positioning	○	○
	TBL	Positioning by 1-table operation	○	○
	DRVTBL	Positioning by multiple-table operation	○	○
	DRVMUL	Multiple axis simultaneous drive positioning	○	○
	DABS	32-bit data ABS current value read	○	○
	PLSV	16-bit data variable speed pulse	○	○
	DPLSV	32-bit data variable speed pulse	○	○
	DRVI	16-bit data relative positioning	○	○
	DDRVI	32-bit data relative positioning	○	○
	DRVA	16-bit data absolute positioning	○	○
	DDRVA	32-bit data absolute positioning	○	○
BFM split read/ write	RBFM	BFM split read	○	○
	WBFM	BFM split write	○	○

Appendix 1.2 Special Relay List (SM)

The PLC contains internal relays with fixed specifications, so it cannot be used in the program like a conventional internal relay.

Diagnostic information

The special relays for diagnostic information are shown below.

R: Read only, R/W: Read/Write

No.	Name	Description	R/W
SM0	Latest self diagnostics error (including annunciator ON)	OFF: No error ON: Error	R
SM1	Latest self diagnostics error (not including annunciator On)	OFF: No self-diagnosis errors ON: Self-diagnosis error	R
SM50	Error reset	OFF→ON: Error reset request ON→OFF: Error reset completion	R/W
SM51	Battery low latch	OFF: Normal ON: Battery low	R
SM52	Battery low	OFF: Normal ON: Battery low	R
SM53	AC/DC DOWN	OFF: No AC/DC down detection ON: AC/DC down is detected	R
SM56	Instruction execution fault	OFF: Normal ON: Operation error	R
SM61	I/O module verify error	OFF: Normal ON: Error	R
SM62	Annunciator	OFF: Not detected ON: Detected	R/W
SM80	Detailed information 1: Flag in use	OFF: Not used ON: In use	R
SM112	Detailed information 2: Flag in use	OFF: Not used ON: In use	R/W

System information

The special relays for system information are shown below.

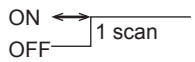
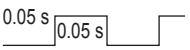
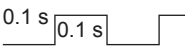
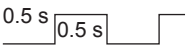
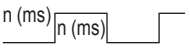
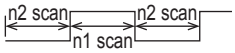
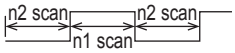
R: Read only, R/W: Read/Write

No.	Name	Description	R/W
SM203	STOP contact	OFF: Other than STOP state ON: STOP state	R
SM204	PAUSE contact	OFF: Other than PAUSE state ON: PAUSE state	R
SM210	Clock data set request	OFF→ON: Set Request ON→OFF: Set completed	R
SM211	Clock data set error	OFF: No error ON: Error	R
SM213	Clock data read request	OFF: Ignored ON: Read request	R

System clock

The special relay about system clock is shown below.

R: Read only, R/W: Read/Write

No.	Name	Description	R/W
SM400	Always ON		R
SM401	Always OFF		R
SM402	After RUN, ON for one scan only		R
SM403	After RUN, OFF for one scan only		R
SM409	0.01 second clock		R
SM410	0.1 second clock		R
SM411	0.2 second clock		R
SM412	1 second clock		R
SM413	2 second clock		R
SM414	2n second clock		R
SM415	2n ms clock		R
SM420	Timing clock output 0		R
SM421	Timing clock output 1		R
SM422	Timing clock output 2		R
SM423	Timing clock output 3		R
SM424	Timing clock output 4		R

Drive information

The special relays for drive information are shown below.

R: Read only, R/W: Read/Write

No.	Name	Description	R/W
SM600	Memory card usable	OFF: Unusable ON: Use enabled	R
SM601	Memory card protect	OFF: Not protected ON: Protected	R
SM603	Memory card insertion	OFF: No drive 2 ON: Drive 2 present	R
SM605	Memory card interchange protect	OFF: Remove/insert enabled ON: Remove/insert prohibited	R/W
SM606	Memory card disable request	OFF: Clear command ON: Command	R/W
SM607	Memory card disable status	OFF: Not disabled by SD memory card forced stop request ON: Disabled by SD memory card forced stop request	R
SM632	Data memory write error detection	OFF: Write not executed/normal ON: Write error	R
SM633	Data memory writing	OFF: Write not executed ON: Writing	R
SM634	Data memory write count error detection flag	OFF: Overwrite count is less than 20,000 ON: Overwrite count is 20,000 or more	R

Instruction related

The special relays related to instruction execution are shown below.

R: Read only, R/W: Read/Write

No.	Name	Description	R/W
SM700	Carry flag	OFF: Carry OFF ON: Carry ON	R
SM701	Output characters selection	OFF: NULL code output ON: No change	R/W
SM703	Sort order	OFF: Ascending order ON: Descending order	R/W
SM704	Block comparison	OFF: Non-match found ON: All match	R
SM709	DT/TM instruction improper data detection	OFF: Improper data not detected ON: Improper data detected	R/W

FX high-speed input/output

The special relays for FX high-speed input/output are shown below.

R: Read only, R/W: Read/Write

No.	Name	Description	R/W
SM4500	High-speed counter operation (CH1)	OFF: Stopped ON: Operation	R
SM4501	High-speed counter operation (CH2)	OFF: Stopped ON: Operation	R
SM4502	High-speed counter operation (CH3)	OFF: Stopped ON: Operation	R
SM4503	High-speed counter operation (CH4)	OFF: Stopped ON: Operation	R
SM4504	High-speed counter operation (CH5)	OFF: Stopped ON: Operation	R
SM4505	High-speed counter operation (CH6)	OFF: Stopped ON: Operation	R
SM4506	High-speed counter operation (CH7)	OFF: Stopped ON: Operation	R

No.	Name	Description	R/W
SM4507	High-speed counter operation (CH8)	OFF: Stopped ON: Operation	R
SM4516	High-speed counter pulse density/Rotation speed measurement (CH1)	OFF: Stopped ON: Measurement	R
SM4517	High-speed counter pulse density/Rotation speed measurement (CH2)	OFF: Stopped ON: Measurement	R
SM4518	High-speed counter pulse density/Rotation speed measurement (CH3)	OFF: Stopped ON: Measurement	R
SM4519	High-speed counter pulse density/Rotation speed measurement (CH4)	OFF: Stopped ON: Measurement	R
SM4520	High-speed counter pulse density/Rotation speed measurement (CH5)	OFF: Stopped ON: Measurement	R
SM4521	High-speed counter pulse density/Rotation speed measurement (CH6)	OFF: Stopped ON: Measurement	R
SM4522	High-speed counter pulse density/Rotation speed measurement (CH7)	OFF: Stopped ON: Measurement	R
SM4523	High-speed counter pulse density/Rotation speed measurement (CH8)	OFF: Stopped ON: Measurement	R
SM4532	High-speed counter overflow (CH1)	OFF: No error ON: Overflow	R/W
SM4533	High-speed counter overflow (CH2)	OFF: No error ON: Overflow	R/W
SM4534	High-speed counter overflow (CH3)	OFF: No error ON: Overflow	R/W
SM4535	High-speed counter overflow (CH4)	OFF: No error ON: Overflow	R/W
SM4536	High-speed counter overflow (CH5)	OFF: No error ON: Overflow	R/W
SM4537	High-speed counter overflow (CH6)	OFF: No error ON: Overflow	R/W
SM4538	High-speed counter overflow (CH7)	OFF: No error ON: Overflow	R/W
SM4539	High-speed counter overflow (CH8)	OFF: No error ON: Overflow	R/W
SM4548	High-speed counter underflow (CH1)	OFF: No error ON: Underflow	R/W
SM4549	High-speed counter underflow (CH2)	OFF: No error ON: Underflow	R/W
SM4550	High-speed counter underflow (CH3)	OFF: No error ON: Underflow	R/W
SM4551	High-speed counter underflow (CH4)	OFF: No error ON: Underflow	R/W
SM4552	High-speed counter underflow (CH5)	OFF: No error ON: Underflow	R/W
SM4553	High-speed counter underflow (CH6)	OFF: No error ON: Underflow	R/W
SM4554	High-speed counter underflow (CH7)	OFF: No error ON: Underflow	R/W
SM4555	High-speed counter underflow (CH8)	OFF: No error ON: Underflow	R/W
SM4564	High-speed counter count direction monitor (CH1) (1-phase 2-input, 2-phase 2-input)	OFF: Up-counting ON: Down-counting	R
SM4565	High-speed counter count direction monitor (CH2) (1-phase 2-input, 2-phase 2-input)	OFF: Up-counting ON: Down-counting	R
SM4566	High-speed counter count direction monitor (CH3) (1-phase 2-input, 2-phase 2-input)	OFF: Up-counting ON: Down-counting	R
SM4567	High-speed counter count direction monitor (CH4) (1-phase 2-input, 2-phase 2-input)	OFF: Up-counting ON: Down-counting	R
SM4568	High-speed counter count direction monitor (CH5) (1-phase 2-input, 2-phase 2-input)	OFF: Up-counting ON: Down-counting	R

No.	Name	Description	R/W
SM4569	High-speed counter count direction monitor (CH6) (1-phase 2-input, 2-phase 2-input)	OFF: Up-counting ON: Down-counting	R
SM4570	High-speed counter count direction monitor (CH7) (1-phase 2-input, 2-phase 2-input)	OFF: Up-counting ON: Down-counting	R
SM4571	High-speed counter count direction monitor (CH8) (1-phase 2-input, 2-phase 2-input)	OFF: Up-counting ON: Down-counting	R
SM4580	High-speed counter count switching (CH1) (1-phase 1-input S/W)	OFF: Up-counting ON: Down-counting	R/W
SM4581	High-speed counter count switching (CH2) (1-phase 1-input S/W)	OFF: Up-counting ON: Down-counting	R/W
SM4582	High-speed counter count switching (CH3) (1-phase 1-input S/W)	OFF: Up-counting ON: Down-counting	R/W
SM4583	High-speed counter count switching (CH4) (1-phase 1-input S/W)	OFF: Up-counting ON: Down-counting	R/W
SM4584	High-speed counter count switching (CH5) (1-phase 1-input S/W)	OFF: Up-counting ON: Down-counting	R/W
SM4585	High-speed counter count switching (CH6) (1-phase 1-input S/W)	OFF: Up-counting ON: Down-counting	R/W
SM4586	High-speed counter count switching (CH7) (1-phase 1-input S/W)	OFF: Up-counting ON: Down-counting	R/W
SM4587	High-speed counter count switching (CH8) (1-phase 1-input S/W)	OFF: Up-counting ON: Down-counting	R/W
SM4596	High-speed counter preset input logic (CH1)	OFF: Positive logic ON: Negative logic	R/W
SM4597	High-speed counter preset input logic (CH2)	OFF: Positive logic ON: Negative logic	R/W
SM4598	High-speed counter preset input logic (CH3)	OFF: Positive logic ON: Negative logic	R/W
SM4599	High-speed counter preset input logic (CH4)	OFF: Positive logic ON: Negative logic	R/W
SM4600	High-speed counter preset input logic (CH5)	OFF: Positive logic ON: Negative logic	R/W
SM4601	High-speed counter preset input logic (CH6)	OFF: Positive logic ON: Negative logic	R/W
SM4602	High-speed counter preset input logic (CH7)	OFF: Positive logic ON: Negative logic	R/W
SM4603	High-speed counter preset input logic (CH8)	OFF: Positive logic ON: Negative logic	R/W
SM4612	High-speed counter preset input comparison (CH1)	OFF: Disabled ON: Enabled	R/W
SM4613	High-speed counter preset input comparison (CH2)	OFF: Disabled ON: Enabled	R/W
SM4614	High-speed counter preset input comparison (CH3)	OFF: Disabled ON: Enabled	R/W
SM4615	High-speed counter preset input comparison (CH4)	OFF: Disabled ON: Enabled	R/W
SM4616	High-speed counter preset input comparison (CH5)	OFF: Disabled ON: Enabled	R/W
SM4617	High-speed counter preset input comparison (CH6)	OFF: Disabled ON: Enabled	R/W
SM4618	High-speed counter preset input comparison (CH7)	OFF: Disabled ON: Enabled	R/W
SM4619	High-speed counter preset input comparison (CH8)	OFF: Disabled ON: Enabled	R/W
SM4628	High-speed counter enable input logic (CH1)	OFF: Positive logic ON: Negative logic	R/W
SM4629	High-speed counter enable input logic (CH2)	OFF: Positive logic ON: Negative logic	R/W
SM4630	High-speed counter enable input logic (CH3)	OFF: Positive logic ON: Negative logic	R/W

No.	Name	Description	R/W
SM4631	High-speed counter enable input logic (CH4)	OFF: Positive logic ON: Negative logic	R/W
SM4632	High-speed counter enable input logic (CH5)	OFF: Positive logic ON: Negative logic	R/W
SM4633	High-speed counter enable input logic (CH6)	OFF: Positive logic ON: Negative logic	R/W
SM4634	High-speed counter enable input logic (CH7)	OFF: Positive logic ON: Negative logic	R/W
SM4635	High-speed counter enable input logic (CH8)	OFF: Positive logic ON: Negative logic	R/W
SM4644	High-speed counter ring length (CH1)	OFF: Disabled ON: Enabled	R/W
SM4645	High-speed counter ring length (CH2)	OFF: Disabled ON: Enabled	R/W
SM4646	High-speed counter ring length (CH3)	OFF: Disabled ON: Enabled	R/W
SM4647	High-speed counter ring length (CH4)	OFF: Disabled ON: Enabled	R/W
SM4648	High-speed counter ring length (CH5)	OFF: Disabled ON: Enabled	R/W
SM4649	High-speed counter ring length (CH6)	OFF: Disabled ON: Enabled	R/W
SM4650	High-speed counter ring length (CH7)	OFF: Disabled ON: Enabled	R/W
SM4651	High-speed counter ring length (CH8)	OFF: Disabled ON: Enabled	R/W
SM4980	High-speed comparison table (high-speed compare instruction) operation	OFF: Stopped ON: Operation	R
SM4982	High-speed comparison table (high-speed compare instruv ction) error occurrence	OFF: No error ON: Error	R/W
SM5000	Multi-point output high-speed comparison table operation	OFF: Stopped ON: Operation	R
SM5001	Multi-point output high-speed comparison table completion	OFF: Not completed ON: Completion	R/W
SM5020	Pulse width measurement operation (CH1)	OFF: Stopped ON: Operation	R
SM5021	Pulse width measurement operation (CH2)	OFF: Stopped ON: Operation	R
SM5022	Pulse width measurement operation (CH3)	OFF: Stopped ON: Operation	R
SM5023	Pulse width measurement operation (CH4)	OFF: Stopped ON: Operation	R
SM5036	Pulse width measurement period measurement complete (CH1)	OFF: Cycle measurement not completed ON: Cycle measurement completion	R
SM5037	Pulse width measurement period measurement complete (CH2)	OFF: Cycle measurement not completed ON: Cycle measurement completion	R
SM5038	Pulse width measurement period measurement complete (CH3)	OFF: Cycle measurement not completed ON: Cycle measurement completion	R
SM5039	Pulse width measurement period measurement complete (CH4)	OFF: Cycle measurement not completed ON: Cycle measurement completion	R
SM5052	Pulse width measurement pulse width measurement complete (CH1)	OFF: Pulse width measurement not completed ON: Pulse width measurement completion	R
SM5053	Pulse width measurement pulse width measurement complete (CH2)	OFF: Pulse width measurement not completed ON: Pulse width measurement completion	R
SM5054	Pulse width measurement pulse width measurement complete (CH3)	OFF: Pulse width measurement not completed ON: Pulse width measurement completion	R
SM5055	Pulse width measurement pulse width measurement complete (CH4)	OFF: Pulse width measurement not completed ON: Pulse width measurement completion	R
SM5068	Pulse width measurement mode (CH1)	OFF: Always measurement mode ON: 1 time measurement mode	R/W

No.	Name	Description	R/W
SM5069	Pulse width measurement mode (CH2)	OFF: Always measurement mode ON: 1 time measurement mode	R/W
SM5070	Pulse width measurement mode (CH3)	OFF: Always measurement mode ON: 1 time measurement mode	R/W
SM5071	Pulse width measurement mode (CH4)	OFF: Always measurement mode ON: 1 time measurement mode	R/W
SM5300	PWM function operation (CH1)	OFF: Stopped ON: Operation	R
SM5301	PWM function operation (CH2)	OFF: Stopped ON: Operation	R
SM5302	PWM function operation (CH3)	OFF: Stopped ON: Operation	R
SM5303	PWM function operation (CH4)	OFF: Stopped ON: Operation	R
SM5500	Positioning instruction activation (axis 1)	OFF: Stopped ON: Operation	R
SM5501	Positioning instruction activation (axis 2)	OFF: Stopped ON: Operation	R
SM5502	Positioning instruction activation (axis 3)	OFF: Stopped ON: Operation	R
SM5503	Positioning instruction activation (axis 4)	OFF: Stopped ON: Operation	R
SM5516	Positioning pulse output monitor (axis 1)	OFF: Stopped ON: Output	R
SM5517	Positioning pulse output monitor (axis 2)	OFF: Stopped ON: Output	R
SM5518	Positioning pulse output monitor (axis 3)	OFF: Stopped ON: Output	R
SM5519	Positioning pulse output monitor (axis 4)	OFF: Stopped ON: Output	R
SM5532	Positioning error (axis 1)	OFF: No error ON: Error	R/W
SM5533	Positioning error (axis 2)	OFF: No error ON: Error	R/W
SM5534	Positioning error (axis 3)	OFF: No error ON: Error	R/W
SM5535	Positioning error (axis 4)	OFF: No error ON: Error	R/W
SM5580	Positioning table shift instructions (axis 1)	OFF: No table shift ON: Table shift start	R/W
SM5581	Positioning table shift instructions (axis 2)	OFF: No table shift ON: Table shift start	R/W
SM5582	Positioning table shift instructions (axis 3)	OFF: No table shift ON: Table shift start	R/W
SM5583	Positioning table shift instructions (axis 4)	OFF: No table shift ON: Table shift start	R/W
SM5596	Positioning remaining distance operation enabled (axis 1)	OFF: Remaining distance operation disabled ON: Remaining distance operation enabled	R/W
SM5597	Positioning remaining distance operation enabled (axis 2)	OFF: Remaining distance operation disabled ON: Remaining distance operation enabled	R/W
SM5598	Positioning remaining distance operation enabled (axis 3)	OFF: Remaining distance operation disabled ON: Remaining distance operation enabled	R/W
SM5599	Positioning remaining distance operation enabled (axis 4)	OFF: Remaining distance operation disabled ON: Remaining distance operation enabled	R/W
SM5612	Positioning remaining distance operation start (axis 1)	OFF: Remaining distance operation standby ON: Remaining distance operation start	R/W
SM5613	Positioning remaining distance operation start (axis 2)	OFF: Remaining distance operation standby ON: Remaining distance operation start	R/W
SM5614	Positioning remaining distance operation start (axis 3)	OFF: Remaining distance operation standby ON: Remaining distance operation start	R/W

No.	Name	Description	R/W
SM5615	Positioning remaining distance operation start (axis 4)	OFF: Remaining distance operation standby ON: Remaining distance operation start	R/W
SM5628	Positioning pulse output stop command (axis 1)	OFF: Pulse output is not stopped ON: Pulse output immediate stop	R/W
SM5629	Positioning pulse output stop command (axis 2)	OFF: Pulse output is not stopped ON: Pulse output immediate stop	R/W
SM5630	Positioning pulse output stop command (axis 3)	OFF: Pulse output is not stopped ON: Pulse output immediate stop	R/W
SM5631	Positioning pulse output stop command (axis 4)	OFF: Pulse output is not stopped ON: Pulse output immediate stop	R/W
SM5644	Positioning pulse decelerates stop command (axis 1) (With remaining distance operation)	OFF: Pulse output is not stopped ON: Pulse output decelerates stop	R/W
SM5645	Positioning pulse decelerates stop command (axis 2) (With remaining distance operation)	OFF: Pulse output is not stopped ON: Pulse output decelerates stop	R/W
SM5646	Positioning pulse decelerates stop command (axis 3) (With remaining distance operation)	OFF: Pulse output is not stopped ON: Pulse output decelerates stop	R/W
SM5647	Positioning pulse decelerates stop command (axis 4) (With remaining distance operation)	OFF: Pulse output is not stopped ON: Pulse output decelerates stop	R/W
SM5660	Positioning forward rotation limit (axis 1)	OFF: Forward rotation limit OFF ON: Forward rotation limit ON	R/W
SM5661	Positioning forward rotation limit (axis 2)	OFF: Forward rotation limit OFF ON: Forward rotation limit ON	R/W
SM5662	Positioning forward rotation limit (axis 3)	OFF: Forward rotation limit OFF ON: Forward rotation limit ON	R/W
SM5663	Positioning forward rotation limit (axis 4)	OFF: Forward rotation limit OFF ON: Forward rotation limit ON	R/W
SM5676	Positioning reverse rotation limit (axis 1)	OFF: Reverse rotation limit OFF ON: Reverse rotation limit ON	R/W
SM5677	Positioning reverse rotation limit (axis 2)	OFF: Reverse rotation limit OFF ON: Reverse rotation limit ON	R/W
SM5678	Positioning reverse rotation limit (axis 3)	OFF: Reverse rotation limit OFF ON: Reverse rotation limit ON	R/W
SM5679	Positioning reverse rotation limit (axis 4)	OFF: Reverse rotation limit OFF ON: Reverse rotation limit ON	R/W
SM5772	Positioning rotational direction (axis 1)	OFF: Forward rotation (Current address increases) ON: Reverse rotation (Current address increases)	R/W
SM5773	Positioning rotational direction (axis 2)	OFF: Forward rotation (Current address increases) ON: Reverse rotation (Current address increases)	R/W
SM5774	Positioning rotational direction (axis 3)	OFF: Forward rotation (Current address increases) ON: Reverse rotation (Current address increases)	R/W
SM5775	Positioning rotational direction (axis 4)	OFF: Forward rotation (Current address increases) ON: Reverse rotation (Current address increases)	R/W
SM5804	Positioning zero return direction (axis 1)	OFF: Zero return start (Reverse rotation direction) ON: Zero return start (Forward rotation direction)	R/W
SM5805	Positioning zero return direction (axis 2)	OFF: Zero return start (Reverse rotation direction) ON: Zero return start (Forward rotation direction)	R/W
SM5806	Positioning zero return direction (axis 3)	OFF: Zero return start (Reverse rotation direction) ON: Zero return start (Forward rotation direction)	R/W
SM5807	Positioning zero return direction (axis 4)	OFF: Zero return start (Reverse rotation direction) ON: Zero return start (Forward rotation direction)	R/W
SM5820	Positioning clear signal function (axis 1)	OFF: Clear signal disabled ON: Clear signal enabled	R/W
SM5821	Positioning clear signal function (axis 2)	OFF: Clear signal disabled ON: Clear signal enabled	R/W
SM5822	Positioning clear signal function (axis 3)	OFF: Clear signal disabled ON: Clear signal enabled	R/W
SM5823	Positioning clear signal function (axis 4)	OFF: Clear signal disabled ON: Clear signal enabled	R/W
SM5868	Positioning zero-point signal count start (axis 1)	OFF: Near point DOG backward end ON: Near point DOG forward end	R/W

No.	Name	Description	R/W
SM5869	Positioning zero-point signal count start (axis 2)	OFF: Near point DOG backward end ON: Near point DOG forward end	R/W
SM5870	Positioning zero-point signal count start (axis 3)	OFF: Near point DOG backward end ON: Near point DOG forward end	R/W
SM5871	Positioning zero-point signal count start (axis 4)	OFF: Near point DOG backward end ON: Near point DOG forward end	R/W

Built-in analog

The special relays for built-in analog are shown below.

R: Read only, R/W: Read/Write

No.	Name	Description	R/W
SM6020	CH1 A/D conversion completed flag	OFF: A/D conversion not completed ON: A/D conversion completed	R
SM6021	CH1 A/D conversion enable/disable setting	OFF: A/D conversion enable ON: A/D conversion disable	R/W
SM6022	CH1 Over scaling detection flag	OFF: No over scaling ON: Over Scaling	R
SM6024	CH1 Over scaling over detection setting	OFF: Enable ON: Disable	R/W
SM6025	CH1 Maximum value/minimum value reset completed flag	OFF: Reset not completed ON: Reset completed	R
SM6026	CH1 Maximum value reset request	OFF: No reset request ON: Reset request	R
SM6027	CH1 Minimum value reset request	OFF: No reset request ON: Reset request	R
SM6028	CH1 A/D scaling enable/disable setting	OFF: Enable ON: Disable	R/W
SM6029	CH1 Digital clipping enable/disable setting	OFF: Enable ON: Disable	R/W
SM6031	CH1 Warning output flag (Process alarm upper limit)	OFF: No alarm ON: Alarm	R
SM6032	CH1 Warning output flag (Process alarm lower limit)	OFF: No alarm ON: Alarm	R
SM6033	CH1 Warning output setting (Process alarm)	OFF: Enabled ON: Disabled	R/W
SM6057	CH1 A/D alarm clear request	OFF: No clear request ON: Clear request	R/W
SM6058	CH1 A/D alarm flag	OFF: No alarm ON: Alarm	R
SM6059	CH1 A/D error flag	OFF: No error ON: Error	R
SM6060	CH2 A/D conversion completed flag	OFF: A/D conversion not completed ON: A/D conversion completed	R
SM6061	CH2 A/D conversion enable/disable setting	OFF: A/D conversion enable ON: A/D conversion disable	R/W
SM6062	CH2 Over scaling detection flag	OFF: No over scaling ON: Over scaling	R
SM6064	CH2 Over scaling over detection	OFF: Enable ON: Disable	R/W
SM6065	CH2 Maximum value/minimum value reset completed flag	OFF: Reset not completed ON: Reset completed	R
SM6066	CH2 Maximum value reset request	OFF: No reset request ON: Reset request	R
SM6067	CH2 Minimum value reset request	OFF: No reset request ON: Reset request	R
SM6068	CH2 A/D scaling enable/disable setting	OFF: Enable ON: Disable	R/W
SM6069	CH2 Digital clipping enable/disable setting	OFF: Enable ON: Disable	R/W

No.	Name	Description	R/W
SM6071	CH2 Warning output flag (Process alarm upper limit)	OFF: No alarm ON: Alarm	R
SM6072	CH2 Warning output flag (Process alarm lower limit)	OFF: No alarm ON: Alarm	R
SM6073	CH2 Warning output setting (Process alarm)	OFF: Enabled ON: Disabled	R/W
SM6097	CH2 A/D alarm clear request	OFF: No clear request ON: Clear request	R/W
SM6098	CH2 A/D alarm flag	OFF: No alarm ON: Alarm	R/W
SM6099	CH2 A/D error flag	OFF: No error ON: Error	R/W
SM6180	D/A conversion enable/disable setting	OFF: D/A conversion enable ON: D/A conversion disable	R/W
SM6181	D/A output enable/disable	OFF: Output enable ON: Output disable	R/W
SM6188	D/A scaling enable/disable setting	OFF: Enable ON: Disable	R/W
SM6191	Warning output upper limit value flag	OFF: No alarm ON: Alarm	R
SM6192	Warning output lower limit value flag	OFF: No alarm ON: Alarm	R
SM6193	Warning output setting	OFF: Disabled ON: Enabled	R/W
SM6217	D/A alarm clear request	OFF: No clear request ON: Clear request	R/W
SM6218	D/A alarm flag	OFF: No alarm ON: Alarm	R
SM6219	D/A error flag	OFF: No error ON: Error	R

FX compatible area

The special relays of FX compatible area are shown below.

R: Read only, R/W: Read/Write

No.	Name	Description	R/W
SM8000	RUN monitor NO contact	OFF: STOP ON: RUN	R
SM8001	RUN monitor NC contact	OFF: RUN ON: STOP	R
SM8002	Initial pulse NO contact	OFF: SM8002 turns off except during 1 scan at the time of RUN ON: SM8002 turns on during 1 scan at the time of RUN	R
SM8003	Initial pulse NC contact	OFF: SM8003 turns on during 1 scan at the time of RUN ON: SM8003 turns off except during 1 scan at the time of RUN	R
SM8004	Error occurrence	OFF: No error ON: Error	R
SM8005	Battery voltage low	OFF: Battery normal ON: Battery voltage low	R
SM8006	Battery error latch	OFF: Battery normal ON: Battery voltage low latch	R
SM8007	Momentary power failure	OFF: No momentary power failure ON: Momentary power failure detected	R
SM8008	Power failure detected	OFF: No momentary power failure ON: During momentary power failure	R
SM8011	10 msec clock pulse	ON and OFF in 10 ms cycles OFF: 5 ms ON: 5 ms	R
SM8012	100 msec clock pulse	ON and OFF in 100 ms cycles OFF: 50 ms ON: 50 ms	R

No.	Name	Description	R/W
SM8013	1 sec clock pulse	ON and OFF in 1 sec cycles OFF: 500 ms ON: 500 ms	R
SM8014	1 min clock pulse	ON and OFF in 1 min cycles OFF: 30 s ON: 30 s	R
SM8015	Clock stop and preset	When SM8015 turns ON, the real time clock is stopped. At the edge from ON to OFF, the time from SD8013 to SD8019 is written to the PLC and the clock is started again.	R/W
SM8016	Time read display is stopped	When SM8016 turns ON, the time display is stopped.	R/W
SM8017	±30 seconds correction	At the edge from OFF to ON, the RTC is set to the nearest minute. (When the second data is from 0 to 29, it is set to 0. When the second data is from 30 to 59, it is set to 0 and the minute data is incriminated by "1".)	R/W
SM8019	Real time clock error	When the data stored in special registers is outside the allowable time setting range, this device turns ON.	R
SM8020	Zero	OFF: Carry flag OFF ON: Carry flag ON	R
SM8021	Borrow	OFF: Borrow flag OFF ON: Borrow flag ON	R
SM8022	Carry	OFF: Carry flag OFF ON: Carry flag ON	R
SM8023	Real time clock access error	SM8023 turns ON at the time of RTC access (reading/writing) error occurrence.	R
SM8026	RAMP mode	OFF: Standard mode ON: RAMP mode	R
SM8029	Instruction execution complete	OFF: Instruction execution not complete ON: Instruction execution complete	R
SM8031	Non-latch memory all clear	OFF: No clear ON: Non-latch memory all clear	R
SM8032	Latch memory all clear	OFF: No clear ON: Latch memory all clear	R
SM8033	Memory hold stop	OFF: Clear ON: Hold	R
SM8034	All output disable	OFF: Normal operation ON: All output disable	R
SM8039	Constant scan mode	OFF: Normal operation ON: Constant scan mode	R/W
SM8040	STL transfer disable	OFF: Normal operation ON: Transfer disable	R/W
SM8041	Transfer start	Transfer from initial state is enabled in automatic operation mode	R
SM8042	Transfer start	Pulse output is given in response to a start input	R
SM8043	Zero return complete	Set this in the last state of zero return mode	R/W
SM8044	Zero point condition	Set this when machine zero return is detected	R/W
SM8045	All output reset disable	Disables the 'all output reset' function when the operation mode is changed	R/W
SM8046	STL state ON	ON when SM8047 is ON and any state (S) is active	R/W
SM8047	Enable STL monitoring	SD8040 to SD8047 are enabled when SM8047 is ON	R/W
SM8048	Annunciator ON	ON when SM8049 is ON and any state (S900 to S999) is active	R/W
SM8049	Enable annunciator monitoring	SD8049 is enabled when SM8049 is ON.	R/W
SM8050	I00☐ disable	OFF: Interrupt enabled ON: Interrupt disabled	R/W
SM8051	I10☐ disable	OFF: Interrupt enabled ON: Interrupt disabled	R/W
SM8052	I20☐ disable	OFF: Interrupt enabled ON: Interrupt disabled	R/W
SM8053	I30☐ disable	OFF: Interrupt enabled ON: Interrupt disabled	R/W
SM8054	I40☐ disable	OFF: Interrupt enabled ON: Interrupt disabled	R/W

No.	Name	Description	R/W
SM8055	I50□ disable	OFF: Interrupt enabled ON: Interrupt disabled	R/W
SM8056	I60□ disable	OFF: Interrupt enabled ON: Interrupt disabled	R/W
SM8057	I70□ disable	OFF: Interrupt enabled ON: Interrupt disabled	R/W
SM8058	I80□ disable	OFF: Interrupt enabled ON: Interrupt disabled	R/W
SM8059	I0□0 disable (Counter interrupt disable)	OFF: Interrupt enabled ON: Interrupt disabled	R/W
SM8063	Serial communication error1 (ch1)	OFF: No error ON: Error	R
SM8067	Operation error	OFF: No error ON: Error	R
SM8068	Operation error latch	OFF: No error ON: Error (latch)	R
SM8090	Block comparison signal	Block comparison signal ON when all comparison results are ON.	R
SM8099	High-speed ring counter	OFF: High-speed ring counter stop ON: High-speed ring counter start	R/W
SM8151	Inverter communication (ch1)	ON during inverter communication.	R
SM8152	Inverter communication error (ch1)	OFF: No error ON: Error	R
SM8153	Inverter communication error latch (ch1)	OFF: No error ON: Error (latch)	R
SM8154	IVBWR instruction error (ch1)	OFF: No error ON: Error	R
SM8156	Inverter communication (ch2)	ON during inverter communication.	R
SM8157	Inverter communication error (ch2)	OFF: No error ON: Error	R
SM8158	Inverter communication error latch (ch2)	OFF: No error ON: Error (latch)	R
SM8159	IVBWR instruction error (ch2)	OFF: No error ON: Error	R
SM8161	8 bit operation mode	OFF: 16 bit operation mode ON: 8 bit operation mode	R/W
SM8168	SMOV data mode	BIN→BCD conversion will not be performed, if a SMOV instruction is executed after turning on SM8168.	R/W
SM8170	X0 pulse catch	Pulse catch ON when X0 is OFF→ON	R/W
SM8171	X1 pulse catch	Pulse catch ON when X1 is OFF→ON	R/W
SM8172	X2 pulse catch	Pulse catch ON when X2 is OFF→ON	R/W
SM8173	X3 pulse catch	Pulse catch ON when X3 is OFF→ON	R/W
SM8174	X4 pulse catch	Pulse catch ON when X4 is OFF→ON	R/W
SM8175	X5 pulse catch	Pulse catch ON when X5 is OFF→ON	R/W
SM8176	X6 pulse catch	Pulse catch ON when X6 is OFF→ON	R/W
SM8177	X7 pulse catch	Pulse catch ON when X7 is OFF→ON	R/W
SM8183	Data communication error (Master station)	OFF: No error ON: Error	R
SM8184	Data communication error (Slave station No.1)	OFF: No error ON: Error	R
SM8185	Data communication error (Slave station No.2)	OFF: No error ON: Error	R
SM8186	Data communication error (Slave station No.3)	OFF: No error ON: Error	R
SM8187	Data communication error (Slave station No.4)	OFF: No error ON: Error	R
SM8188	Data communication error (Slave station No.5)	OFF: No error ON: Error	R
SM8189	Data communication error (Slave station No.6)	OFF: No error ON: Error	R
SM8190	Data communication error (Slave station No.7)	OFF: No error ON: Error	R

No.	Name	Description	R/W
SM8191	Data communication in execution	OFF: Data communication in execution ON: Data communication in nonexecution	R
SM8246	LC46 counting direction monitoring	OFF: Down count operation ON: Up count operation	R
SM8247	LC47 counting direction monitoring	OFF: Down count operation ON: Up count operation	R
SM8248	LC48 counting direction monitoring	OFF: Down count operation ON: Up count operation	R
SM8249	LC49 counting direction monitoring	OFF: Down count operation ON: Up count operation	R
SM8250	LC50 counting direction monitoring	OFF: Down count operation ON: Up count operation	R
SM8251	LC51 counting direction monitoring	OFF: Down count operation ON: Up count operation	R
SM8252	LC52 counting direction monitoring	OFF: Down count operation ON: Up count operation	R
SM8253	LC53 counting direction monitoring	OFF: Down count operation ON: Up count operation	R
SM8254	LC54 counting direction monitoring	OFF: Down count operation ON: Up count operation	R
SM8255	LC55 counting direction monitoring	OFF: Down count operation ON: Up count operation	R
SM8304	Zero	OFF: Zero flag OFF ON: Zero flag ON	R
SM8306	Carry	OFF: Carry flag OFF ON: Carry flag ON	R
SM8329	Instruction execution error	OFF: Instruction execution normal ON: Instruction execution error complete	R
SM8330	Timing clock output 1	DUTY instruction: Timing clock output 1	R
SM8331	Timing clock output 2	DUTY instruction: Timing clock output 2	R
SM8332	Timing clock output 3	DUTY instruction: Timing clock output 3	R
SM8333	Timing clock output 4	DUTY instruction: Timing clock output 4	R
SM8334	Timing clock output 5	DUTY instruction: Timing clock output 5	R
SM8340	Axis 1 pulse output monitor	OFF: Stopped ON: Pulse output	R
SM8348	Axis 1 positioning instruction executing	OFF: Positioning instruction not executing ON: Positioning instruction executing	R
SM8350	Axis 2 pulse output monitor	OFF: Stopped ON: Output	R
SM8358	Axis 2 positioning instruction executing	OFF: Positioning instruction not executing ON: Positioning instruction executing	R
SM8360	Axis 3 pulse output monitor	OFF: Stopped ON: Output	R
SM8368	Axis 3 positioning instruction executing	OFF: Positioning instruction not executing ON: Positioning instruction executing	R
SM8370	Axis 4 pulse output monitor	OFF: Stopped ON: Output	R
SM8378	Axis 4 positioning instruction executing	OFF: Positioning instruction not executing ON: Positioning instruction executing	R
SM8401	RS2 Send wait flag (ch1)/MODBUS request in process (ch1)	ON during send wait or MODBUS communication.	R
SM8402	MODBUS communication error (ch1)	OFF: No error ON: Error	R
SM8403	MODBUS communication error (latched) (ch1)	OFF: No error ON: Error (latch)	R
SM8404	RS2 Carrier detection flag (ch1)/MODBUS communication mode (ch1)	ON when carrier detection or listen only mode	R
SM8405	RS2 Data set ready (DSR) flag (ch1)	OFF: DSR not detected ON: DSR detected	R
SM8408	MODBUS retry (ch1)	OFF: Not retry ON: Retry	R
SM8409	RS2 Time-out check flag (ch1)/MODBUS Timeout (ch1)	ON when time-out occurs.	R

No.	Name	Description	R/W
SM8421	RS2 Send wait flag (ch2)/MODBUS request in process (ch2)	ON during send wait or MODBUS communication	R
SM8422	MODBUS communication error (ch2)	OFF: No error ON: Error	R
SM8423	MODBUS communication error (latched) (ch2)	OFF: No error ON: Error (latch)	R
SM8424	RS2 Carrier detection flag (ch2)/MODBUS communication mode (ch2)	Carrier detection flag or listen only mode ON when operating.	R
SM8425	RS2 Data set ready (DSR) flag (ch2)	OFF: DSR not detected ON: DSR detected	R
SM8428	MODBUS retry (ch2)	OFF: No retry ON: Retry	R
SM8429	RS2 Time-out check flag (ch2)/MODBUS Timeout (ch2)	ON when timeout occurs.	R
SM8438	Serial communication error 2 (ch2)	OFF: No error ON: Error	R
SM8492	IP address storage area write request	If OFF to ON, the IP address setting stored in SD8492 to SD8497 will be written in the IP address storage area.	R/W
SM8493	IP address storage area write completed	• It turns on, if the write to the IP address storage area is completed. Moreover, it turns on also at the time of the write-in failure. • Turns OFF when IP address storage area write request (SM8492) turns from ON to OFF.	R
SM8494	IP address storage area write error	• Turns ON when writing to IP address storage area is failed. • Turns ON if there is a problem in contents of IP address storage area, when PLC power supply is turned from OFF to ON. • Turns OFF when IP address storage area write request (SM8492) turns from ON to OFF.	R
SM8495	IP address storage area clear request	Contents of IP address storage area are cleared when this device turns from OFF to ON.	R/W
SM8496	IP address storage area clear completed	• It turns on, if the clear to the IP address storage area is completed. Moreover, it turns on also at the time of the clear-in failure. • Turns OFF when IP address storage area clear request (SM8495) turns from ON to OFF.	R
SM8497	IP address storage area clear error	• Turns ON when clear to IP address storage area is failed. • Turns OFF when IP address storage area clear request (SM8495) turns from ON to OFF.	R
SM8498	IP address change function enable flag	Turns ON when IP address is changed by IP address change function	R

Serial communication

The special relays for serial communication are shown below.

R: Read only, R/W: Read/Write

No.	Name	Description	R/W
SM8500	Serial communication error (ch1)	OFF: No error ON: Error	R
SM8510	Serial communication error (ch2)	OFF: No error ON: Error	R
SM8520	Serial communication error (ch3)	OFF: No error ON: Error	R
SM8530	Serial communication error (ch4)	OFF: No error ON: Error	R
SM8560	Data transfer delayed (ch1)	This device remains ON while the PLC is waiting to send.	R
SM8561	Data transfer flag (ch1)	When this device is set to ON, the PLC starts to send.	R
SM8562	Receive completion flag (ch1)	This device turns ON when receiving is completed.	R
SM8563	Carrier detection flag (ch1)	This device turns ON in synchronization with the CD (DCD) signal.	R
SM8564	Data set ready flag (ch1)	This device turns ON in synchronization with the DR (DSR) signal.	R
SM8565	Time-out check flag (ch1)	This device turns ON when data receiving is suspended and the next set of receive data is not given within the time set by the timeout time setting device.	R
SM8570	Data transfer delayed (ch2)	This device remains ON while the PLC is waiting to send.	R

No.	Name	Description	R/W
SM8571	Data transfer flag (ch2)	When this device is set to ON, the PLC starts to send.	R
SM8572	Receive completion flag (ch2)	This device turns ON when receiving is completed.	R
SM8573	Carrier detection flag (ch2)	This device turns ON in synchronization with the CD (DCD) signal.	R
SM8574	Data set ready flag (ch2)	This device turns ON in synchronization with the DR (DSR) signal.	R
SM8575	Time-out check flag (ch2)	This device turns ON when data receiving is suspended and the next set of receive data is not given within the time set by the timeout time setting device.	R
SM8580	Data transfer delayed (ch3)	This device remains ON while the PLC is waiting to send.	R
SM8581	Data transfer flag (ch3)	When this device is set to ON, the PLC starts to send.	R
SM8582	Receive completion flag (ch3)	This device turns ON when receiving is completed.	R
SM8583	Carrier detection flag (ch3)	This device turns ON in synchronization with the CD (DCD) signal.	R
SM8584	Data set ready flag (ch3)	This device turns ON in synchronization with the DR (DSR) signal.	R
SM8585	Time-out check flag (ch3)	This device turns ON when data receiving is suspended and the next set of receive data is not given within the time set by the timeout time setting device	R
SM8590	Data transfer delayed (ch4)	This device remains ON while the PLC is waiting to send	R
SM8591	Data transfer flag (ch4)	When this device is set to ON, the PLC starts to send	R
SM8592	Receive completion flag (ch4)	This device turns ON when receiving is completed	R
SM8593	Carrier detection flag (ch4)	This device turns ON in synchronization with the CD (DCD) signal	R
SM8594	Data set ready flag (ch4)	This device turns ON in synchronization with the DR (DSR) signal	R
SM8595	Time-out check flag (ch4)	This device turns ON when data receiving is suspended and the next set of receive data is not given within the time set by the timeout time setting device	R
SM8740	Station No. setting SD latch enabled (ch1)	OFF: Latch disabled ON: Latch enabled	R
SM8750	Station No. setting SD latch enabled (ch2)	OFF: Latch disabled ON: Latch enabled	R
SM8760	Station No. setting SD latch enabled (ch3)	OFF: Latch disabled ON: Latch enabled	R
SM8770	Station No. setting SD latch enabled (ch4)	OFF: Latch disabled ON: Latch enabled	R
SM8800	MODBUS RTU communication (ch1)	OFF: Communication stop ON: Communication	R
SM8801	Retry (ch1)	OFF: No retry ON: Retry	R
SM8802	Timeout (ch1)	OFF: No timeout ON: Timeout	R
SM8810	MODBUS RTU communication (ch2)	OFF: Communication stop ON: Communication	R
SM8811	Retry (ch2)	OFF: No retry ON: Retry	R
SM8812	Timeout (ch2)	OFF: Not timeout ON: Timeout	R
SM8820	MODBUS RTU communication (ch3)	OFF: Communication stop ON: Communication	R
SM8821	Retry (ch3)	OFF: No retry ON: Retry	R
SM8822	Timeout (ch3)	OFF: No timeout ON: Timeout	R
SM8830	MODBUS RTU communication (ch4)	OFF: Communication stop ON: Communication	R
SM8831	Retry (ch4)	OFF: No retry ON: Retry	R
SM8832	Timeout (ch4)	OFF: No timeout ON: Timeout	R
SM8861	Host station No. setting SD latch enabled (ch1)	OFF: Latch disabled ON: Latch enabled	R
SM8871	Host station No. setting SD latch enabled (ch2)	OFF: Latch disabled ON: Latch enabled	R
SM8881	Host station No. setting SD latch enabled (ch3)	OFF: Latch disabled ON: Latch enabled	R

No.	Name	Description	R/W
SM8891	Host station No. setting SD latch enabled (ch4)	OFF: Latch disabled ON: Latch enabled	R
SM8920	Inverter communication (ch1)	OFF: No communication ON: Communication	R
SM8921	IVBWR instruction error (ch1)	OFF: No error ON: Error	R
SM8930	Inverter communication (ch2)	OFF: No communication ON: Communication	R
SM8931	IVBWR instruction error (ch2)	OFF: No error ON: Error	R
SM8940	Inverter communication (ch3)	OFF: No communication ON: Communication	R
SM8941	IVBWR instruction error (ch3)	OFF: No error ON: Error	R
SM8950	Inverter communication (ch4)	OFF: No communication ON: Communication	R
SM8951	IVBWR instruction error (ch4)	OFF: No error ON: Error	R
SM9040	Data communication error (Master station)	OFF: No error ON: Error	R
SM9041	Data communication error (Slave station No.1)	OFF: No error ON: Error	R
SM9042	Data communication error (Slave station No.2)	OFF: No error ON: Error	R
SM9043	Data communication error (Slave station No.3)	OFF: No error ON: Error	R
SM9044	Data communication error (Slave station No.4)	OFF: No error ON: Error	R
SM9045	Data communication error (Slave station No.5)	OFF: No error ON: Error	R
SM9046	Data communication error (Slave station No.6)	OFF: No error ON: Error	R
SM9047	Data communication error (Slave station No.7)	OFF: No error ON: Error	R
SM9056	Data communication in execution	OFF: Data communication in execution ON: Data communication in nonexecution	R
SM9080	Station No. setting SD latch enabled	OFF: Latch disabled ON: Latch enabled	R
SM9081	Slave station total number setting SD latch enabled	OFF: Latch disabled ON: Latch enabled	R

Appendix 1.3 Special Register List (SD)

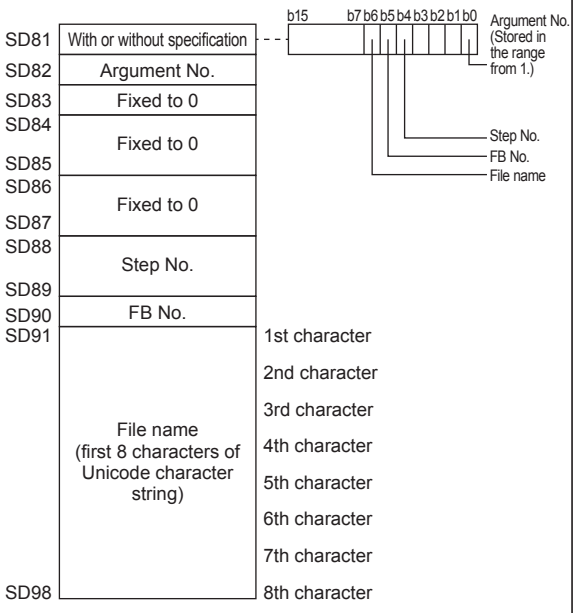
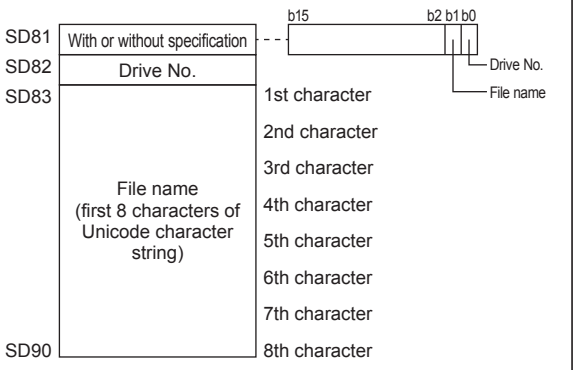
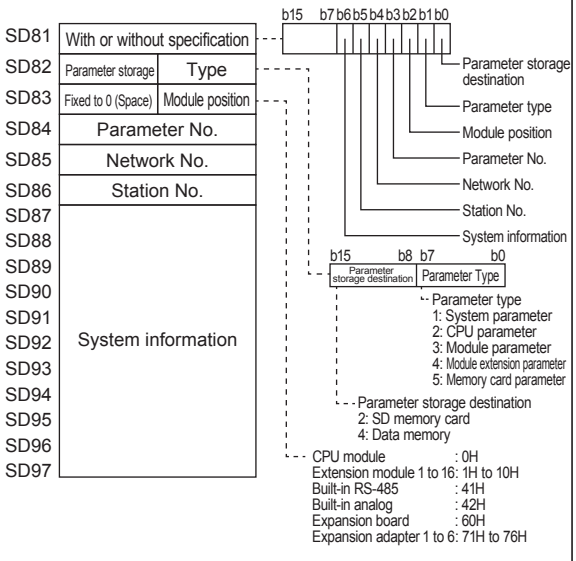
The PLC contains internal register with fixed specifications, so it cannot be used in the program like a conventional internal register.

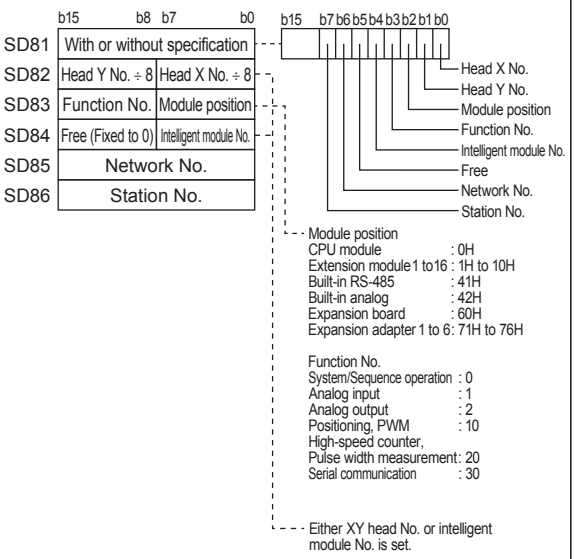
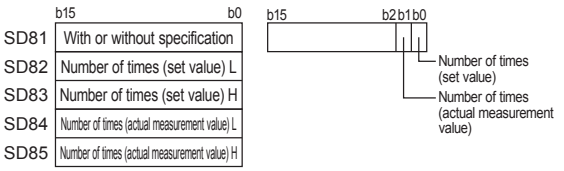
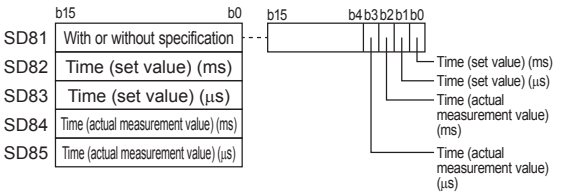
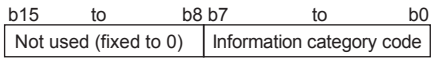
Diagnostic information

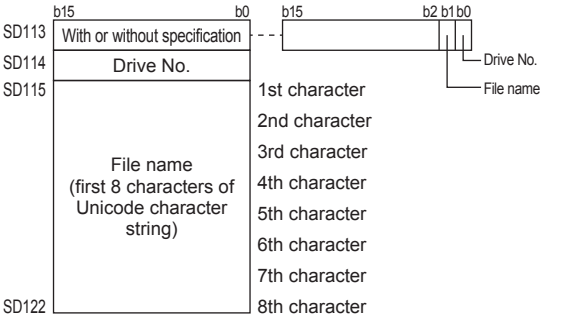
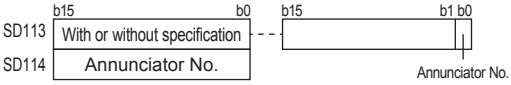
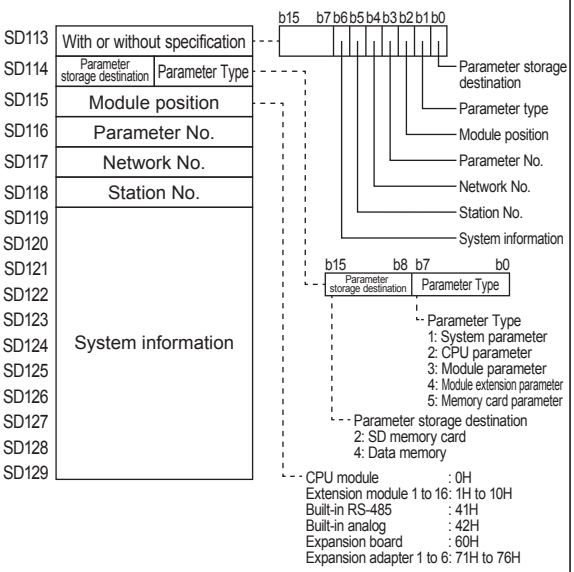
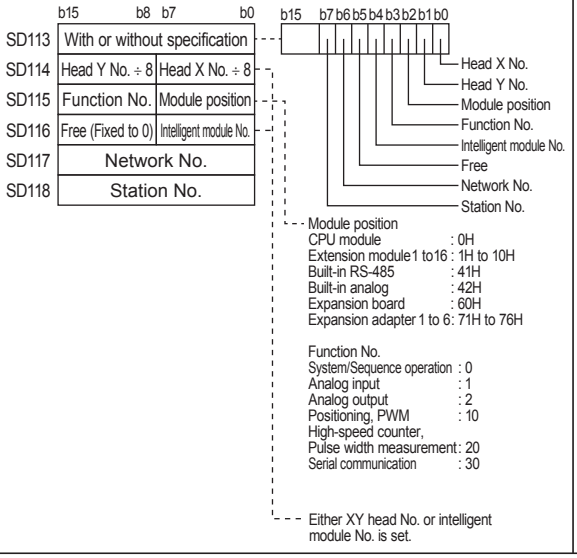
The special register for diagnostic information are shown below.

R: Read only, R/W: Read/Write

No.	Name	Description	R/W
SD0	Latest self diagnostics error code	This register stores the latest self-diagnosis error code.	R
SD1	Clock time for self diagnosis error occurrence (Year)	This register stores the latest self-diagnosis error time (Year).	R
SD2	Clock time for self diagnosis error occurrence (Month)	This register stores the latest self-diagnosis error time (Month).	R
SD3	Clock time for self diagnosis error occurrence (Day)	This register stores the latest self-diagnosis error time (Day).	R
SD4	Clock time for self diagnosis error occurrence (Hour)	This register stores the latest self-diagnosis error time (Hour).	R
SD5	Clock time for self diagnosis error occurrence (Minute)	This register stores the latest self-diagnosis error time (Minute).	R
SD6	Clock time for self diagnosis error occurrence (Second)	This register stores the latest self-diagnosis error time (Second).	R
SD7	Clock time for self diagnosis error occurrence (Day Week)	This register stores the latest self-diagnosis error time (Day Week).	R
SD10	Self diagnostics error code 1	This register stores the self-diagnosis error code.	R
SD11	Self diagnostics error code 2	This register stores the self-diagnosis error code.	R
SD12	Self diagnostics error code 3	This register stores the self-diagnosis error code.	R
SD13	Self diagnostics error code 4	This register stores the self-diagnosis error code.	R
SD14	Self diagnostics error code 5	This register stores the self-diagnosis error code.	R
SD15	Self diagnostics error code 6	This register stores the self-diagnosis error code.	R
SD16	Self diagnostics error code 7	This register stores the self-diagnosis error code.	R
SD17	Self diagnostics error code 8	This register stores the self-diagnosis error code.	R
SD18	Self diagnostics error code 9	This register stores the self-diagnosis error code.	R
SD19	Self diagnostics error code 10	This register stores the self-diagnosis error code.	R
SD20	Self diagnostics error code 11	This register stores the self-diagnosis error code.	R
SD21	Self diagnostics error code 12	This register stores the self-diagnosis error code.	R
SD22	Self diagnostics error code 13	This register stores the self-diagnosis error code.	R
SD23	Self diagnostics error code 14	This register stores the self-diagnosis error code.	R
SD24	Self diagnostics error code 15	This register stores the self-diagnosis error code.	R
SD25	Self diagnostics error code 16	This register stores the self-diagnosis error code.	R
SD53	The number of AC/DC DOWN detections	This register stores the number of times of momentary power failure.	R
SD61	I/O Module Verify Error Module No.	This register stores the I/O module verify error module No.	R
SD62	Annunciator (F) Detection No.	This register stores the earliest detected annunciator (F) No.	R
SD63	Annunciator (F) Detection Number	This register stores the number of annunciator (F) detections.	R
SD64 to SD79	Annunciator (F) Detection No. table	This register stores the annunciator (F) detection No.	R
SD80	Detailed information 1 information category	• Detailed information 1 information category code is stored. b15 to b8 b7 to b0 Not used (fixed to 0) | Information category code • The following codes are stored into the information category code. 0: N/A 1: Program position information 2: Drive number and file name 4: Parameter information 5: System configuration information 6: Number of times information 7: Time information	R

No.	Name	Description	R/W
SD81 to SD111	Detailed information 1	• Detailed information 1 corresponding to the error code (SD0) is stored. • There are six types of information to be stored as shown in the following figures. • The type of detailed information 1 can be obtained using SD80 (the value of the "Detailed information 1 information category code" stored in SD80 corresponds to the following figures (1), (2), (4) to (7)). (1) Program location information  (2) Drive number and file name  (4) Parameter information  	R

No.	Name	Description	R/W
SD81 to SD111	Detailed information 1	(5) System configuration information  (6) Number of times information  (7) Time information 	R
SD112	Detailed information 2 information category	• Detailed information 2 information category code is stored.  • The following codes are stored into the information category code. 0: N/A 2: Drive number and file name 3: Annunciator number 4: Parameter information 5: System configuration information	R

No.	Name	Description	R/W
SD113 to SD143	Detailed information 2	• Detailed information 2 corresponding to the error code (SD0) is stored. • There are four types of information to be stored as shown in the following figures. • The type of detailed information 2 can be obtained using SD112 (the value of the "Detailed information 2 information category code" stored in SD112 corresponds to the following figures (2) to (5)). (2) Drive number and file name  (3) Annunciator number  (4) Parameter information  (5) System configuration information  	R

System information

The special registers for system information are shown below.

R: Read only, R/W: Read/Write

No.	Name	Description	R/W
SD200	Switch Status	This register stores the CPU switch status. 0: RUN 1: STOP	R
SD201	LED Status	This register stores the LED status.	R
SD203	CPU Status	This register stores the CPU Status. b0: RUN b2: STOP b3: PAUSE	R
SD210	Clock Data (Year)	This register stores the clock data (Year).	R
SD211	Clock Data (Month)	This register stores the clock data (Month).	R
SD212	Clock Data (Day)	This register stores the clock data (Day).	R
SD213	Clock Data (Hour)	This register stores the clock data (Hour).	R
SD214	Clock Data (Minute)	This register stores the clock data (Minute).	R
SD215	Clock Data (Second)	This register stores the clock data (Second).	R
SD216	Clock Data (Day Week)	This register stores the clock data (Day of the Week).	R
SD218	Time zone setting value	The time zone setting value specified in the parameter is stored in increments of minutes.	R
SD250	Loaded Max I/O	This register stores high-order 2 digits of the final I/O number of connected modules +1 in 8-bit binary.	R
SD260	X Device Size [Lower]	This register stores the number of X device points used as 32-bit value.	R
SD261	X Device Size [Upper]		
SD262	Y Device Size [Lower]	This register stores the number of Y device points used as 32-bit value.	R
SD263	Y Device Size [Upper]		
SD264	M Device Size [Lower]	This register stores the number of M device points used as 32-bit value.	R
SD265	M Device Size [Upper]		
SD266	B Device Size [Lower]	This register stores the number of B device points used as 32-bit value.	R
SD267	B Device Size [Upper]		
SD268	SB Device Size [Lower]	This register stores the number of SB device points used as 32-bit value.	R
SD269	SB Device Size [Upper]		
SD270	F Device Size [Lower]	This register stores the number of F device points used as 32-bit value.	R
SD271	F Device Size [Upper]		
SD274	L Device Size [Lower]	This register stores the number of L device points used as 32-bit value.	R
SD275	L Device Size [Upper]		
SD280	D Device Size [Lower]	This register stores the number of D device points used as 32-bit value.	R
SD281	D Device Size [Upper]		
SD282	W Device Size [Lower]	This register stores the number of W device points used as 32-bit value.	R
SD283	W Device Size [Upper]		
SD284	SW Device Size [Lower]	This register stores the number of SW device points used as 32-bit value.	R
SD285	SW Device Size [Upper]		
SD288	T Device Size [Lower]	This register stores the number of T device points used as 32-bit value.	R
SD289	T Device Size [Upper]		
SD290	ST Device Size [Lower]	This register stores the number of ST device points used as 32-bit value.	R
SD291	ST Device Size [Upper]		
SD292	C Device Size [Lower]	This register stores the number of C device points used as 32-bit value.	R
SD293	C Device Size [Upper]		
SD298	LC Device Size [Lower]	This register stores the number of LC device points used as 32-bit value.	R
SD299	LC Device Size [Upper]		
SD300	Z Device Size	This register stores the number of Z device points used.	R
SD302	LZ Device Size	This register stores the number of LZ device points used.	R
SD304	R Device Size [Lower]	This register stores the number of R device points used as 32-bit value.	R
SD305	R Device Size [Upper]		

System clock

The special registers for system clock are shown below.

R: Read only, R/W: Read/Write

No.	Name	Description	R/W
SD412	One second counter	• This register is incremented by 1 for each second after the CPU module is set to RUN. • A counting cycle from 0 to 32767 to -32768 to 0 is repeated.	R
SD414	2n second clock setting	• Stores value n of 2n second clock (Default is 30) • Setting can be made between 1 and 32767.	R/W
SD415	2nms second clock setting	• Stores value n of 2n ms clock (Default is 30) • Setting can be made between 1 and 32767.	R/W
SD420	Scan counter	• This register is incremented by 1 each scan after the CPU module is set to RUN. (Not incremented for each scan of an initial execution type program.) • A counting cycle from 0 to 32767 to -32768 to 0 is repeated.	R

Scan information

The special registers for scan information are shown below.
R: Read only, R/W: Read/Write

No.	Name	Description	R/W
SD500	Execution program number	Program number of program currently being executed is stored.	R
SD518	Initial scan time (ms)	This register stores the initial scan time (ms).	R
SD519	Initial scan time (μs)	This register stores the initial scan time (μs).	R
SD520	Current scan time (ms)	This register stores the current scan time (ms).	R
SD521	Current scan time (μs)	This register stores the current scan time (μs).	R
SD522	Minimum scan time (ms)	This register stores the minimum scan time (ms).	R
SD523	Minimum scan time(μs)	This register stores the minimum scan time (μs).	R
SD524	Maximum scan time (ms)	This register stores the maximum scan time (ms).	R
SD525	Maximum scan time (μs)	This register stores the maximum scan time (μs).	R
SD526	END processing time (ms)	This register stores the END processing time (ms).	R
SD527	END processing time (μs)	This register stores the END processing time (μs).	R
SD528	Constant scan waiting time (ms)	This register stores the constant scan wait time (ms).	R
SD529	Constant scan waiting time (μs)	This register stores the constant scan wait time (μs).	R
SD530	Scan program execution time (ms)	This register stores the scan program execution time (ms).	R
SD531	Scan program execution time (μs)	This register stores the scan program execution time (μs).	R

Drive information

The special registers for drive information are shown below.
R: Read only, R/W: Read/Write

No.	Name	Description	R/W
SD600	Memory Card Installation	This register stores the enable/disable classification of the inserted SD memory card.	R
SD604	SD memory card usage status	This register stores the memory card usage condition.	R
SD606	SD memory card capacity	This register stores the drive 2 storage capacity (unit: 1 K byte). (Free space value after formatting is stored.)	R
SD607	SD memory card capacity	This register stores the drive 2 storage capacity (unit: 1 K byte). (Free space value after formatting is stored.)	R
SD608	SD memory card capacity	This register stores the drive 2 storage capacity (unit: 1 K byte). (Free space value after formatting is stored.)	R
SD609	SD memory card capacity	This register stores the drive 2 storage capacity (unit: 1 K byte). (Free space value after formatting is stored.)	R
SD610	SD memory card free space capacity	This register stores the free space value in drive 2 (unit: 1 K byte).	R
SD611	SD memory card free space capacity	This register stores the free space value in drive 2 (unit: 1 K byte).	R
SD612	SD memory card free space capacity	This register stores the free space value in drive 2 (unit: 1 K byte).	R
SD613	SD memory card free space capacity	This register stores the free space value in drive 2 (unit: 1 K byte).	R

No.	Name	Description	R/W
SD634	Index for the number of data memory write operations	Stores an index for the number of write operations to data memory currently. However, the index does not equal the actual number of write operations.	R
SD635			

Instruction related

R: Read only, R/W: Read/Write

No.	Name	Description	R/W
SD757	Current interrupt priority	This register stores the interrupt priority of the interrupt program being executed. 1 to 3: The interrupt priority of interrupt program executed. 0: The interrupt is not executed. (default value)	R
SD758	Interrupt disabling for each priority setting value	This register stores the disable interrupt priority according to the disable interrupt instruction (DI), disable interrupt after the setting priority instruction (DI), and enable interrupt instruction (EI). 1: Disable interrupt priority 1 or less. (Disable interrupt of all priority) (default value) 2: Disable interrupt priority 2 or 3. 3: Disable interrupt priority 3. 0: No priority. (Enable interrupt of all priority)	R

Mask pattern of interrupt pointers

The special registers for the mask pattern of interrupt pointers are shown below.
R: Read only, R/W: Read/Write

No.	Name	Description	R/W
SD1400	Mask pattern I	This register stores the IMASK instruction mask pattern I. b15 to b0: I15 to I0	R/W
SD1401	Mask pattern I	This register stores the IMASK instruction mask pattern I. b15 to b0: I31 to I16	R/W

FX dedicated

The special registers dedicated to FX are shown below.
R: Read only, R/W: Read/Write

No.	Name	Description	R/W
SD4110	Error code 1 details	This register stores the self-diagnosis error code details. • Module position [Low order 8 bit] 0H: CPU module 41H: Built-in RS-485 42H: Built-in analog 60H: Expansion board 71H to 76H: Expansion adapter • Function No. [Higher order 8 bit] 0: System/Sequence operation 1: Analog input 2: Analog output 10: Positioning, PWM 20: High-speed counter, Pulse width measurement	R
SD4111	Error code 2 details		
SD4112	Error code 3 details		
SD4113	Error code 4 details		
SD4114	Error code 5 details		
SD4115	Error code 6 details		
SD4116	Error code 7 details		
SD4117	Error code 8 details		
SD4118	Error code 9 details		
SD4119	Error code 10 details		
SD4120	Error code 11 details		
SD4121	Error code 12 details		
SD4122	Error code 13 details		
SD4123	Error code 14 details		
SD4124	Error code 15 details		
SD4125	Error code 16 details		
SD4150	Module 1 status information	This register stores the module 1 status information.	R
SD4151	Module 1 error information	This register stores the module 1 error information.	R
SD4152	Module 2 status information	This register stores the module 2 status information	R
SD4153	Module 2 error information	This register stores the module 2 error information.	R
SD4154	Module 3 status information	This register stores the module 3 status information.	R
SD4155	Module 3 error information	This register stores the module 3 error information.	R
SD4156	Module 4 status information	This register stores the module 4 status information.	R
SD4157	Module 4 error information	This register stores the module 4 error information.	R
SD4158	Module 5 status information	This register stores the module 5 status information.	R
SD4159	Module 5 error information	This register stores the module 5 error information.	R
SD4160	Module 6 status information	This register stores the module 6 status information.	R
SD4161	Module 6 error information	This register stores the module 6 error information.	R
SD4162	Module 7 status information	This register stores the module 7 status information.	R
SD4163	Module 7 error information	This register stores the module 7 error information.	R
SD4164	Module 8 status information	This register stores the module 8 status information	R
SD4165	Module 8 error information	This register stores the module 8 error information.	R
SD4166	Module 9 status information	This register stores the module 9 status information.	R
SD4167	Module 9 error information	This register stores the module 9 error information.	R
SD4168	Module 10 status information	This register stores the module 10 status information.	R
SD4169	Module 10 error information	This register stores the module 10 error information.	R
SD4170	Module 11 status information	This register stores the module 11 status information.	R
SD4171	Module 11 error information	This register stores the module 11 error information.	R
SD4172	Module 12 status information	This register stores the module 12 status information.	R
SD4173	Module 12 error information	This register stores the module 12 error information	R

No.	Name	Description	R/W
SD4174	Module 13 status information	This register stores the module 13 status information.	R
SD4175	Module 13 error information	This register stores the module 13 error information.	R
SD4176	Module 14 status information	This register stores the module 14 status information.	R
SD4177	Module 14 error information	This register stores the module 14 error information.	R
SD4178	Module 15 status information	This register stores the module 15 status information.	R
SD4179	Module 15 error information	This register stores the module 15 error information.	R
SD4180	Module 16 status information	This register stores the module 16 status information.	R
SD4181	Module 16 error information	This register stores the module 16 error information.	R

FX high-speed input/output

The special registers for FX high-speed input/output are shown below.
R: Read only, R/W: Read/Write

No.	Name	Description	R/W
SD4500	High-speed counter current value [Low-order] (CH1)	This register stores the high-speed counter current value (CH1).	R/W
SD4501	High-speed counter current value [High-order] (CH1)		
SD4502	High-speed counter maximum value [Low-order] (CH1)	This register stores the high-speed counter maximum value (CH1).	R/W
SD4503	High-speed counter maximum value [High-order] (CH1)		
SD4504	High-speed counter minimum value [Low-order] (CH1)	This register stores the high-speed counter minimum value (CH1).	R/W
SD4505	High-speed counter minimum value [High-order] (CH1)		
SD4506	High-speed counter pulse density [Low-order] (CH1)	This register stores the high-speed counter pulse density (CH1).	R/W
SD4507	High-speed counter pulse density [High-order] (CH1)		
SD4508	High-speed counter rotation speed [Low-order] (CH1)	This register stores the high-speed counter rotation speed (CH1).	R/W
SD4509	High-speed counter rotation speed [High-order] (CH1)		
SD4510	High-speed counter preset control switch (CH1)	This register stores the high-speed counter preset control switch (CH1).	R/W
SD4512	High-speed counter preset value [Low-order] (CH1)	This register stores the high-speed counter preset value (CH1).	R/W
SD4513	High-speed counter preset value [High-order] (CH1)		
SD4514	High-speed counter ring length [Low-order] (CH1)	This register stores the high-speed counter ring length (CH1).	R/W
SD4515	High-speed counter ring length [High-order] (CH1)		
SD4516	High-speed counter measurement-unit time [Low-order] (CH1)	This register stores the high-speed counter measurement-unit time (CH1).	R/W
SD4517	High-speed counter measurement-unit time [High-order] (CH1)		
SD4518	High-speed counter number of pulses per rotation [Loworder] (CH1)	This register stores the high-speed counter number of pulses per rotation (CH1).	R/W
SD4519	High-speed counter number of pulses per rotation [Highorder] (CH1)		
SD4530	High-speed counter current value [Low-order] (CH2)	This register stores the high-speed counter current value (CH2).	R/W
SD4531	High-speed counter current value [High-order] (CH2)		
SD4532	High-speed counter maximum value [Low-order] (CH2)	This register stores the high-speed counter maximum value (CH2).	R/W
SD4533	High-speed counter maximum value [High-order] (CH2)		
SD4534	High-speed counter minimum value [Low-order] (CH2)	This register stores the high-speed counter minimum value (CH2).	R/W
SD4535	High-speed counter minimum value [High-order] (CH2)		
SD4536	High-speed counter pulse density [Low-order] (CH2)	This register stores the high-speed counter pulse density (CH2).	R/W
SD4537	High-speed counter pulse density [High-order] (CH2)		
SD4538	High-speed counter rotation speed [Low-order] (CH2)	This register stores the high-speed counter rotation speed (CH2).	R/W
SD4539	High-speed counter rotation speed [High-order] (CH2)		
SD4540	High-speed counter preset control switch (CH2)	This register stores the high-speed counter preset control switch (CH2).	R/W
SD4542	High-speed counter preset value [Low-order] (CH2)	This register stores the high-speed counter preset value (CH2).	R/W
SD4543	High-speed counter preset value [High-order] (CH2)		
SD4544	High-speed counter ring length [Low-order] (CH2)	This register stores the high-speed counter ring length (CH2).	R/W
SD4545	High-speed counter ring length [High-order] (CH2)		

No.	Name	Description	R/W
SD4546	High-speed counter measurement-unit time [Low-order] (CH2)	This register stores the high-speed counter measurement-unit time (CH2).	R/W
SD4547	High-speed counter measurement-unit time [High-order] (CH2)		
SD4548	High-speed counter number of pulses per rotation [Loworder] (CH2)	This register stores the high-speed counter number of pulses per rotation (CH2).	R/W
SD4549	High-speed counter number of pulses per rotation [Highorder] (CH2)		
SD4560	High-speed counter current value [Low-order] (CH3)	This register stores the high-speed counter current value (CH3).	R/W
SD4561	High-speed counter current value [High-order] (CH3)		
SD4562	High-speed counter maximum value [Low-order] (CH3)	This register stores the high-speed counter maximum value (CH3).	R/W
SD4563	High-speed counter maximum value [High-order] (CH3)		
SD4564	High-speed counter minimum value [Low-order] (CH3)	This register stores the high-speed counter minimum value (CH3).	R/W
SD4565	High-speed counter minimum value [High-order] (CH3)		
SD4566	High-speed counter pulse density [Low-order] (CH3)	This register stores the high-speed counter pulse density (CH3).	R/W
SD4567	High-speed counter pulse density [High-order] (CH3)		
SD4568	High-speed counter rotation speed [Low-order] (CH3)	This register stores the high-speed counter rotation speed (CH3).	R/W
SD4569	High-speed counter rotation speed [High-order] (CH3)		
SD4570	High-speed counter preset control switch (CH3)	This register stores the high-speed counter preset control switch (CH3).	R/W
SD4572	High-speed counter preset value [Low-order] (CH3)	This register stores the high-speed counter preset value (CH3).	R/W
SD4573	High-speed counter preset value [High-order] (CH3)		
SD4574	High-speed counter ring length [Low-order] (CH3)	This register stores the high-speed counter ring length (CH3).	R/W
SD4575	High-speed counter ring length [High-order] (CH3)		
SD4576	High-speed counter measurement-unit time [Low-order] (CH3)	This register stores the high-speed counter measurement-unit time (CH3).	R/W
SD4577	High-speed counter measurement-unit time [High-order] (CH3)		
SD4578	High-speed counter number of pulses per rotation [Loworder] (CH3)	This register stores the high-speed counter number of pulses per rotation (CH3).	R/W
SD4579	High-speed counter number of pulses per rotation [Highorder] (CH3)		
SD4590	High-speed counter current value [Low-order] (CH4)	This register stores the high-speed counter current value (CH4).	R/W
SD4591	High-speed counter current value [High-order] (CH4)		
SD4592	High-speed counter maximum value [Low-order] (CH4)	This register stores the high-speed counter maximum value (CH4).	R/W
SD4593	High-speed counter maximum value [High-order] (CH4)		
SD4594	High-speed counter minimum value [Low-order] (CH4)	This register stores the high-speed counter minimum value (CH4).	R/W
SD4595	High-speed counter minimum value [High-order] (CH4)		
SD4596	High-speed counter pulse density [Low-order] (CH4)	This register stores the high-speed counter pulse density (CH4).	R/W
SD4597	High-speed counter pulse density [High-order] (CH4)		
SD4598	High-speed counter rotation speed [Low-order] (CH4)	This register stores the high-speed counter rotation speed (CH4).	R/W
SD4599	High-speed counter rotation speed [High-order] (CH4)		
SD4600	High-speed counter preset control switch (CH4)	This register stores the high-speed counter preset control switch (CH4).	R/W
SD4602	High-speed counter preset value [Low-order] (CH4)	This register stores the high-speed counter preset value (CH4).	R/W
SD4603	High-speed counter preset value [High-order] (CH4)		
SD4604	High-speed counter ring length [Low-order] (CH4)	This register stores the high-speed counter ring length (CH4).	R/W
SD4605	High-speed counter ring length [High-order] (CH4)		
SD4606	High-speed counter measurement-unit time [Low-order] (CH4)	This register stores the high-speed counter measurement-unit time (CH4).	R/W
SD4607	High-speed counter measurement-unit time [High-order] (CH4)		
SD4608	High-speed counter number of pulses per rotation [Loworder] (CH4)	This register stores the high-speed counter number of pulses per rotation (CH4).	R/W
SD4609	High-speed counter number of pulses per rotation [Highorder] (CH4)		

No.	Name	Description	R/W
SD4620	High-speed counter current value [Low-order] (CH5)	This register stores the high-speed counter current value (CH5).	R/W
SD4621	High-speed counter current value [High-order] (CH5)		
SD4622	High-speed counter maximum value [Low-order] (CH5)	This register stores the high-speed counter maximum value (CH5).	R/W
SD4623	High-speed counter maximum value [High-order] (CH5)		
SD4624	High-speed counter minimum value [Low-order] (CH5)	This register stores the high-speed counter minimum value (CH5).	R/W
SD4625	High-speed counter minimum value [High-order] (CH5)		
SD4626	High-speed counter pulse density [Low-order] (CH5)	This register stores the high-speed counter pulse density (CH5).	R/W
SD4627	High-speed counter pulse density [High-order] (CH5)		
SD4628	High-speed counter rotation speed [Low-order] (CH5)	This register stores the high-speed counter rotation speed (CH5).	R/W
SD4629	High-speed counter rotation speed [High-order] (CH5)		
SD4630	High-speed counter preset control switch (CH5)	This register stores the high-speed counter preset control switch (CH5).	R/W
SD4632	High-speed counter preset value [Low-order] (CH5)	This register stores the high-speed counter preset value (CH5).	R/W
SD4633	High-speed counter preset value [High-order] (CH5)		
SD4634	High-speed counter ring length [Low-order] (CH5)	This register stores the high-speed counter ring length (CH5).	R/W
SD4635	High-speed counter ring length [High-order] (CH5)		
SD4636	High-speed counter measurement-unit time [Low-order] (CH5)	This register stores the high-speed counter measurement-unit time (CH5).	R/W
SD4637	High-speed counter measurement-unit time [High-order] (CH5)		
SD4638	High-speed counter number of pulses per rotation [Loworder] (CH5)	This register stores the high-speed counter number of pulses per rotation (CH5).	R/W
SD4639	High-speed counter number of pulses per rotation [Highorder] (CH5)		
SD4650	High-speed counter current value [Low-order] (CH6)	This register stores the high-speed counter current value (CH6).	R/W
SD4651	High-speed counter current value [High-order] (CH6)		
SD4652	High-speed counter maximum value [Low-order] (CH6)	This register stores the high-speed counter maximum value (CH6).	R/W
SD4653	High-speed counter maximum value [High-order] (CH6)		
SD4654	High-speed counter minimum value [Low-order] (CH6)	This register stores the high-speed counter minimum value (CH6).	R/W
SD4655	High-speed counter minimum value [High-order] (CH6)		
SD4656	High-speed counter pulse density [Low-order] (CH6)	This register stores the high-speed counter pulse density (CH6).	R/W
SD4657	High-speed counter pulse density [High-order] (CH6)		
SD4658	High-speed counter rotation speed [Low-order] (CH6)	This register stores the high-speed counter rotation speed (CH6).	R/W
SD4659	High-speed counter rotation speed [High-order] (CH6)		
SD4660	High-speed counter preset control switch (CH6)	This register stores the high-speed counter preset control switch (CH6).	R/W
SD4662	High-speed counter preset value [Low-order] (CH6)	This register stores the high-speed counter preset value (CH6).	R/W
SD4663	High-speed counter preset value [High-order] (CH6)		
SD4664	High-speed counter ring length [Low-order] (CH6)	This register stores the high-speed counter ring length (CH6).	R/W
SD4665	High-speed counter ring length [High-order] (CH6)		
SD4666	High-speed counter measurement-unit time [Low-order] (CH6)	This register stores the high-speed counter measurement-unit time (CH6).	R/W
SD4667	High-speed counter measurement-unit time [High-order] (CH6)		
SD4668	High-speed counter number of pulses per rotation [Loworder] (CH6)	This register stores the high-speed counter number of pulses per rotation (CH6).	R/W
SD4669	High-speed counter number of pulses per rotation [Highorder] (CH6)		
SD4680	High-speed counter current value [Low-order] (CH7)	This register stores the high-speed counter current value (CH7).	R/W
SD4681	High-speed counter current value [High-order] (CH7)		
SD4682	High-speed counter maximum value [Low-order] (CH7)	This register stores the high-speed counter maximum value (CH7).	R/W
SD4683	High-speed counter maximum value [High-order] (CH7)		
SD4684	High-speed counter minimum value [Low-order] (CH7)	This register stores the high-speed counter minimum value (CH7).	R/W
SD4685	High-speed counter minimum value [High-order] (CH7)		
SD4686	High-speed counter pulse density [Low-order] (CH7)	This register stores the high-speed counter pulse density (CH7).	R/W
SD4687	High-speed counter pulse density [High-order] (CH7)		

No.	Name	Description	R/W
SD4688	High-speed counter rotation speed [Low-order] (CH7)	This register stores the high-speed counter rotation speed (CH7).	R/W
SD4689	High-speed counter rotation speed [High-order] (CH7)		
SD4690	High-speed counter preset control switch (CH7)	This register stores the high-speed counter preset control switch (CH7).	R/W
SD4692	High-speed counter preset value [Low-order] (CH7)	This register stores the high-speed counter preset value (CH7).	R/W
SD4693	High-speed counter preset value [High-order] (CH7)		
SD4694	High-speed counter ring length [Low-order] (CH7)	This register stores the high-speed counter ring length (CH7).	R/W
SD4695	High-speed counter ring length [High-order] (CH7)		
SD4696	High-speed counter measurement-unit time [Low-order] (CH7)	This register stores the high-speed counter measurement-unit time (CH7).	R/W
SD4697	High-speed counter measurement-unit time [High-order] (CH7)		
SD4698	High-speed counter number of pulses per rotation [Loworder] (CH7)	This register stores the high-speed counter number of pulses per rotation (CH7).	R/W
SD4699	High-speed counter number of pulses per rotation [Highorder] (CH7)		
SD4710	High-speed counter current value [Low-order] (CH8)	This register stores the high-speed counter current value (CH8).	R/W
SD4711	High-speed counter current value [High-order] (CH8)		
SD4712	High-speed counter maximum value [Low-order] (CH8)	This register stores the high-speed counter maximum value (CH8).	R/W
SD4713	High-speed counter maximum value [High-order] (CH8)		
SD4714	High-speed counter minimum value [Low-order] (CH8)	This register stores the high-speed counter minimum value (CH8).	R/W
SD4715	High-speed counter minimum value [High-order] (CH8)		
SD4716	High-speed counter pulse density [Low-order] (CH8)	This register stores the high-speed counter pulse density (CH8).	R/W
SD4717	High-speed counter pulse density [High-order] (CH8)		
SD4718	High-speed counter rotation speed [Low-order] (CH8)	This register stores the high-speed counter rotation speed (CH8).	R/W
SD4719	High-speed counter rotation speed [High-order] (CH8)		
SD4720	High-speed counter preset control switch (CH8)	This register stores the high-speed counter preset control switch (CH8).	R/W
SD4722	High-speed counter preset value [Low-order] (CH8)	This register stores the high-speed counter preset value (CH8).	R/W
SD4723	High-speed counter preset value [High-order] (CH8)		
SD4724	High-speed counter ring length [Low-order] (CH8)	This register stores the high-speed counter ring length (CH8).	R/W
SD4725	High-speed counter ring length [High-order] (CH8)		
SD4726	High-speed counter measurement-unit time [Low-order] (CH8)	This register stores the high-speed counter measurement-unit time (CH8).	R/W
SD4727	High-speed counter measurement-unit time [High-order] (CH8)		
SD4728	High-speed counter number of pulses per rotation [Loworder] (CH8)	This register stores the high-speed counter number of pulses per rotation (CH8).	R/W
SD4729	High-speed counter number of pulses per rotation [Highorder] (CH8)		
SD4982	High-speed comparison table (high-speed compare instruction) error code (CPU module)	This register stores the high-speed comparison table (high-speed compare instruction) error code.	R/W
SD5000	Multi-point output high-speed comparison table comparison number	This register stores the multi-point output high-speed comparison table comparison number	R
SD5020	Pulse width measurement rising ring counter value [Loworder] (CH1)	This register stores the pulse width measurement rising ring counter value (CH1).	R
SD5021	Pulse width measurement rising ring counter value [Highorder] (CH1)		
SD5022	Pulse width measurement falling ring counter value [Loworder] (CH1)	This register stores the pulse width measurement falling ring counter value (CH1).	R
SD5023	Pulse width measurement falling ring counter value [Highorder] (CH1)		
SD5024	Pulse width measurement latest value [Low-order] (CH1)	This register stores the pulse width measurement latest value (CH1).	R
SD5025	Pulse width measurement latest value [High-order] (CH1)		
SD5026	Pulse width measurement maximum value [Low-order] (CH1)	This register stores the pulse width measurement maximum value (CH1).	R/W
SD5027	Pulse width measurement maximum value [High-order] (CH1)		

No.	Name	Description	R/W
SD5028	Pulse width measurement minimum value [Low-order] (CH1)	This register stores the pulse width measurement minimum value (CH1).	R/W
SD5029	Pulse width measurement minimum value [High-order] (CH1)		
SD5030	Pulse width measurement cycle latest value [Low-order] (CH1)	This register stores the pulse width measurement cycle latest value (CH1).	R/W
SD5031	Pulse width measurement cycle latest value [High-order] (CH1)		
SD5032	Pulse width measurement cycle maximum value [Loworder] (CH1)	This register stores the pulse width measurement cycle maximum value (CH1).	R/W
SD5033	Pulse width measurement cycle maximum value [Highorder] (CH1)		
SD5034	Pulse width measurement cycle minimum value [Loworder] (CH1)	This register stores the pulse width measurement cycle minimum value (CH1).	R/W
SD5035	Pulse width measurement cycle minimum value [Highorder] (CH1)		
SD5040	Pulse width measurement rising ring counter value [Loworder] (CH2)	This register stores the pulse width measurement rising ring counter value (CH2).	R/W
SD5041	Pulse width measurement rising ring counter value [Highorder] (CH2)		
SD5042	Pulse width measurement falling ring counter value [Loworder] (CH2)	This register stores the pulse width measurement falling ring counter value (CH2).	R/W
SD5043	Pulse width measurement falling ring counter value [Highorder] (CH2)		
SD5044	Pulse width measurement latest value [Low-order] (CH2)	This register stores the pulse width measurement latest value (CH2).	R/W
SD5045	Pulse width measurement latest value [High-order] (CH2)		
SD5046	Pulse width measurement maximum value [Low-order] (CH2)	This register stores the pulse width measurement maximum value (CH2).	R/W
SD5047	Pulse width measurement maximum value [High-order] (CH2)		
SD5048	Pulse width measurement minimum value [Low-order] (CH2)	This register stores the pulse width measurement minimum value (CH2).	R/W
SD5049	Pulse width measurement minimum value [High-order] (CH2)		
SD5050	Pulse width measurement cycle latest value [Low-order] (CH2)	This register stores the pulse width measurement cycle latest value (CH2).	R/W
SD5051	Pulse width measurement cycle latest value [High-order] (CH2)		
SD5052	Pulse width measurement cycle maximum value [Loworder] (CH2)	This register stores the pulse width measurement cycle maximum value (CH2).	R/W
SD5053	Pulse width measurement cycle maximum value [Highorder] (CH2)		
SD5054	Pulse width measurement cycle minimum value [Loworder] (CH2)	This register stores the pulse width measurement cycle minimum value (CH2).	R/W
SD5055	Pulse width measurement cycle minimum value [Highorder] (CH2)		
SD5060	Pulse width measurement rising ring counter value [Loworder] (CH3)	This register stores the pulse width measurement rising ring counter value (CH3).	R/W
SD5061	Pulse width measurement rising ring counter value [Highorder] (CH3)		
SD5062	Pulse width measurement falling ring counter value [Loworder] (CH3)	This register stores the pulse width measurement falling ring counter value (CH3).	R/W
SD5063	Pulse width measurement falling ring counter value [Highorder] (CH3)		
SD5064	Pulse width measurement latest value [Low-order] (CH3)	This register stores the pulse width measurement latest value (CH3).	R/W
SD5065	Pulse width measurement latest value [High-order] (CH3)		
SD5066	Pulse width measurement maximum value [Low-order] (CH3)	This register stores the pulse width measurement maximum value (CH3).	R/W
SD5067	Pulse width measurement maximum value [High-order] (CH3)		

No.	Name	Description	R/W
SD5068	Pulse width measurement minimum value [Low-order] (CH3)	This register stores the pulse width measurement minimum value (CH3).	R/W
SD5069	Pulse width measurement minimum value [High-order] (CH3)		
SD5070	Pulse width measurement cycle latest value [Low-order] (CH3)	This register stores the pulse width measurement cycle latest value (CH3).	R/W
SD5071	Pulse width measurement cycle latest value [High-order] (CH3)		
SD5072	Pulse width measurement cycle maximum value [Loworder] (CH3)	This register stores the pulse width measurement cycle maximum value (CH3).	R/W
SD5073	Pulse width measurement cycle maximum value [Highorder] (CH3)		
SD5074	Pulse width measurement cycle minimum value [Loworder] (CH3)	This register stores the pulse width measurement cycle minimum value (CH3).	R/W
SD5075	Pulse width measurement cycle minimum value [Highorder] (CH3)		
SD5080	Pulse width measurement rising ring counter value [Loworder] (CH4)	This register stores the pulse width measurement rising ring counter value (CH4).	R/W
SD5081	Pulse width measurement rising ring counter value [Highorder] (CH4)		
SD5082	Pulse width measurement falling ring counter value [Loworder] (CH4)	This register stores the pulse width measurement falling ring counter value (CH4).	R/W
SD5083	Pulse width measurement falling ring counter value [Highorder] (CH4)		
SD5084	Pulse width measurement latest value [Low-order] (CH4)	This register stores the pulse width measurement latest value (CH4).	R/W
SD5085	Pulse width measurement latest value [High-order] (CH4)		
SD5086	Pulse width measurement maximum value [Low-order] (CH4)	This register stores the pulse width measurement maximum value (CH4).	R/W
SD5087	Pulse width measurement maximum value [High-order] (CH4)		
SD5088	Pulse width measurement minimum value [Low-order] (CH4)	This register stores the pulse width measurement minimum value (CH4).	R/W
SD5089	Pulse width measurement minimum value [High-order] (CH4)		
SD5090	Pulse width measurement cycle latest value [Low-order] (CH4)	This register stores the pulse width measurement cycle latest value (CH4).	R/W
SD5091	Pulse width measurement cycle latest value [High-order] (CH4)		
SD5092	Pulse width measurement cycle maximum value [Loworder] (CH4)	This register stores the pulse width measurement cycle maximum value (CH4).	R/W
SD5093	Pulse width measurement cycle maximum value [Highorder] (CH4)		
SD5094	Pulse width measurement cycle minimum value [Loworder] (CH4)	This register stores the pulse width measurement cycle minimum value (CH4).	R/W
SD5095	Pulse width measurement cycle minimum value [Highorder] (CH4)		
SD5300	PWM pulse output number [Low-order] (CH1)	This register stores the PWM pulse output number (CH1).	R/W
SD5301	PWM pulse output number [High-order] (CH1)		
SD5302	PWM pulse width [Low-order] (CH1)	This register stores the PWM pulse width (CH1).	R/W
SD5303	PWM pulse width [High-order] (CH1)		
SD5304	PWM cycle [Low-order] (CH1)	This register stores the PWM cycle (CH1).	R/W
SD5305	PWM cycle [High-order] (CH1)		
SD5306	PWM Number of output pulses current value monitor [Low-order] (CH1)	This register stores the PWM pulse output number current value (CH1).	R/W
SD5307	PWM Number of output pulses current value monitor [High-order] (CH1)		
SD5316	PWM pulse output number [Low-order] (CH2)	This register stores the PWM pulse output number (CH2).	R/W
SD5317	PWM pulse output number [High-order] (CH2)		
SD5318	PWM pulse width [Low-order] (CH2)	This register stores the PWM pulse width (CH2).	R/W
SD5319	PWM pulse width [High-order] (CH2)		

No.	Name	Description	R/W
SD5320	PWM cycle [Low-order] (CH2)	This register stores the PWM cycle (CH2).	R/W
SD5321	PWM cycle [High-order] (CH2)		
SD5322	PWM Number of output pulses current value monitor [Low-order] (CH2)	This register stores the PWM pulse output number current value (CH2).	R
SD5323	PWM Number of output pulses current value monitor [High-order] (CH2)		
SD5332	PWM pulse output number [Low-order] (CH3)	This register stores the PWM pulse output number (CH3).	R/W
SD5333	PWM pulse output number [High-order] (CH3)		
SD5334	PWM pulse width [Low-order] (CH3)	This register stores the PWM pulse width (CH3).	R/W
SD5335	PWM pulse width [High-order] (CH3)		
SD5336	PWM cycle [Low-order] (CH3)	This register stores the PWM cycle (CH3).	R/W
SD5337	PWM cycle [High-order] (CH3)		
SD5338	PWM Number of output pulses current value monitor [Low-order] (CH3)	This register stores the PWM pulse output number current value (CH3).	R
SD5339	PWM Number of output pulses current value monitor [High-order] (CH3)		
SD5348	PWM pulse output number [Low-order] (CH4)	This register stores the PWM pulse output number (CH4).	R/W
SD5349	PWM pulse output number [High-order] (CH4)		
SD5350	PWM pulse width [Low-order] (CH4)	This register stores the PWM pulse width (CH4).	R/W
SD5351	PWM pulse width [High-order] (CH4)		
SD5352	PWM cycle [Low-order] (CH4)	This register stores the PWM cycle (CH4).	R/W
SD5353	PWM cycle [High-order] (CH4)		
SD5354	PWM Number of output pulses current value monitor [Low-order] (CH4)	This register stores the PWM pulse output number current value (CH4).	R
SD5355	PWM Number of output pulses current value monitor [High-order] (CH4)		
SD5500	Positioning current address (user unit) [Low-order] (axis 1)	This register stores the current address (user unit) of positioning (axis 1).	R/W
SD5501	Positioning current address (user unit) [High-order] (axis 1)		
SD5502	Positioning current address (pulse unit) [Low-order] (axis 1)	This register stores the current address (pulse unit) of positioning (axis 1).	R/W
SD5503	Positioning current address (pulse unit) [High-order] (axis 1)		
SD5504	Positioning current speed (user unit) [Low-order] (axis 1)	This register stores the current speed (user unit) of positioning (axis 1).	R
SD5505	Positioning current speed (user unit) [High-order] (axis 1)		
SD5506	Positioning execution table number (axis 1)	This register stores the execution table number of positioning (axis 1).	R
SD5510	Positioning error code (axis 1)	This register stores the error code of positioning (axis 1).	R/W
SD5511	Positioning error table number (axis 1)	This register stores the error table number of positioning (axis 1).	R/W
SD5516	Positioning maximum speed [Low-order] (axis 1)	This register stores the maximum speed of positioning (axis 1).	R/W
SD5517	Positioning maximum speed [High-order] (axis 1)		
SD5518	Positioning bias speed [Low-order] (axis 1)	This register stores the bias speed of positioning (axis 1).	R/W
SD5519	Positioning bias speed [High-order] (axis 1)		
SD5520	Positioning acceleration time (axis 1)	This register stores the acceleration time of positioning (axis 1).	R/W
SD5521	Positioning deceleration time (axis 1)	This register stores the deceleration time of positioning (axis 1).	R/W
SD5526	Positioning zero-return speed [Low-order] (axis 1)	This register stores the zero-return speed of positioning (axis 1).	R/W
SD5527	Positioning zero-return speed [High-order] (axis 1)		
SD5528	Positioning creep speed [Low-order] (axis 1)	This register stores the creep speed of positioning (axis 1).	R/W
SD5529	Positioning creep speed [High-order] (axis 1)		
SD5530	Positioning zero-point address [Low-order] (axis 1)	This register stores the zero-point address of positioning (axis 1).	R/W
SD5531	Positioning zero-point address [High-order] (axis 1)		
SD5532	Positioning number of zero-point signal for zero return	This register stores the number of zero-point signal for zero return of positioning (axis 1).	R/W
SD5533	Positioning zero-return dwell time (axis 1)	This register stores the zero-return dwell time of positioning (axis 1).	R/W

No.	Name	Description	R/W
SD5540	Positioning current address (user unit) [Low-order] (axis 2)	This register stores the current address (user unit) of positioning (axis 2).	R/W
SD5541	Positioning current address (user unit) [High-order] (axis 2)		
SD5542	Positioning current address (pulse unit) [Low-order] (axis 2)	This register stores the current address (pulse unit) of positioning (axis 2).	R/W
SD5543	Positioning current address (pulse unit) [High-order] (axis 2)		
SD5544	Positioning current speed (user unit) [Low-order] (axis 2)	This register stores the current speed (user unit) of positioning (axis 2).	R
SD5545	Positioning current speed (user unit) [High-order] (axis 2)		
SD5546	Positioning execution table number (axis 2)	This register stores the execution table number of positioning (axis 2).	R
SD5550	Positioning error code (axis 2)	This register stores the error code of positioning (axis 2).	R/W
SD5551	Positioning error table number (axis 2)	This register stores the error table number of positioning (axis 2).	R/W
SD5556	Positioning maximum speed [Low-order] (axis 2)	This register stores the maximum speed of positioning (axis 2).	R/W
SD5557	Positioning maximum speed [High-order] (axis 2)		
SD5558	Positioning bias speed [Low-order] (axis 2)	This register stores the bias speed of positioning (axis 2).	R/W
SD5559	Positioning bias speed [High-order] (axis 2)		
SD5560	Positioning acceleration time (axis 2)	This register stores the acceleration time of positioning (axis 2).	R/W
SD5561	Positioning deceleration time (axis 2)	This register stores the deceleration time of positioning (axis 2).	R/W
SD5566	Positioning zero-return speed [Low-order] (axis 2)	This register stores the zero-return speed of positioning (axis 2).	R/W
SD5567	Positioning zero-return speed [High-order] (axis 2)		
SD5568	Positioning creep speed [Low-order] (axis 2)	This register stores the creep speed of positioning (axis 2).	R/W
SD5569	Positioning creep speed [High-order] (axis 2)		
SD5570	Positioning zero-point address [Low-order] (axis 2)	This register stores the zero-point address of positioning (axis 2).	R/W
SD5571	Positioning zero-point address [High-order] (axis 2)		
SD5572	Positioning number of zero-point signal for zero return (axis 2)	This register stores the number of zero-point signal for zero return of positioning (axis 2).	R/W
SD5573	Positioning zero-return dwell time (axis 2)	This register stores the zero-return dwell time of positioning (axis 2).	R/W
SD5580	Positioning current address (user unit) [Low-order] (axis 3)	This register stores the current address (user unit) of positioning (axis 3).	R/W
SD5581	Positioning current address (user unit) [High-order] (axis 3)		
SD5582	Positioning current address (pulse unit) [Low-order] (axis 3)	This register stores the current address (pulse unit) of positioning (axis 3).	R/W
SD5583	Positioning current address (pulse unit) [High-order] (axis 3)		
SD5584	Positioning current speed (user unit) [Low-order] (axis 3)	This register stores the current speed (user unit) of positioning (axis 3).	R
SD5585	Positioning current speed (user unit) [High-order] (axis 3)		
SD5586	Positioning execution table number (axis 3)	This register stores the execution table number of positioning (axis 3).	R
SD5590	Positioning error code (axis 3)	This register stores the error code of positioning (axis 3).	R/W
SD5591	Positioning error table number (axis 3)	This register stores the error table number of positioning (axis 3).	R/W
SD5596	Positioning maximum speed [Low-order] (axis 3)	This register stores the maximum speed of positioning (axis 3).	R/W
SD5597	Positioning maximum speed [High-order] (axis 3)		
SD5598	Positioning bias speed [Low-order] (axis 3)	This register stores the bias speed of positioning (axis 3).	R/W
SD5599	Positioning bias speed [High-order] (axis 3)		
SD5600	Positioning acceleration time (axis 3)	This register stores the acceleration time of positioning (axis 3).	R/W
SD5601	Positioning deceleration time (axis 3)	This register stores the deceleration time of positioning (axis 3).	R/W
SD5606	Positioning zero-return speed [Low-order] (axis 3)	This register stores the zero-return speed of positioning (axis 3).	R/W
SD5607	Positioning zero-return speed [High-order] (axis 3)		
SD5608	Positioning creep speed [Low-order] (axis 3)	This register stores the creep speed of positioning (axis 3).	R/W
SD5609	Positioning creep speed [High-order] (axis 3)		
SD5610	Positioning zero-point address [Low-order] (axis 3)	This register stores the zero-point address of positioning (axis 3).	R/W
SD5611	Positioning zero-point address [High-order] (axis 3)		
SD5612	Positioning number of zero-point signal for zero return (axis 3)	This register stores the number of zero-point signal for zero return of positioning (axis 3).	R/W

No.	Name	Description	R/W
SD5613	Positioning zero-return dwell time (axis 3)	This register stores the zero-return dwell time of positioning (axis 3).	R/W
SD5620	Positioning current address (user unit) [Low-order] (axis 4)	This register stores the current address (user unit) of positioning (axis 4).	R/W
SD5621	Positioning current address (user unit) [High-order] (axis 4)		
SD5622	Positioning current address (pulse unit) [Low-order] (axis 4)	This register stores the current address (pulse unit) of positioning (axis 4).	R/W
SD5623	Positioning current address (pulse unit) [High-order] (axis 4)		
SD5624	Positioning current speed (user unit) [Low-order] (axis 4)	This register stores the current speed (user unit) of positioning (axis 4).	R
SD5625	Positioning current speed (user unit) [High-order] (axis 4)		
SD5626	Positioning execution table number (axis 4)	This register stores the execution table number of positioning (axis 4).	R
SD5630	Positioning error code (axis 4)	This register stores the error code of positioning (axis 4).	R/W
SD5631	Positioning error table number (axis 4)	This register stores the error table number of positioning (axis 4).	R/W
SD5636	Positioning maximum speed [Low-order] (axis 4)	This register stores the maximum speed of positioning (axis 4).	R/W
SD5637	Positioning maximum speed [High-order] (axis 4)		
SD5638	Positioning bias speed [Low-order] (axis 4)	This register stores the bias speed of positioning (axis 4).	R/W
SD5639	Positioning bias speed [High-order] (axis 4)		
SD5640	Positioning acceleration time (axis 4)	This register stores the acceleration time of positioning (axis 4).	R/W
SD5641	Positioning deceleration time (axis 4)	This register stores the deceleration time of positioning (axis 4).	R/W
SD5646	Positioning zero-return speed [Low-order] (axis 4)	This register stores the zero-return speed of positioning (axis 4).	R/W
SD5647	Positioning zero-return speed [High-order] (axis 4)		
SD5648	Positioning creep speed [Low-order] (axis 4)	This register stores the creep speed of positioning (axis 4).	R/W
SD5649	Positioning creep speed [High-order] (axis 4)		
SD5650	Positioning zero-point address [Low-order] (axis 4)	This register stores the zero-point address of positioning (axis 4).	R/W
SD5651	Positioning zero-point address [High-order] (axis 4)		
SD5652	Positioning number of zero-point signal for zero return (axis 4)	This register stores the number of zero-point signal for zero return of positioning (axis 4).	R/W
SD5653	Positioning zero-return dwell time (axis 4)	This register stores the zero-return dwell time of positioning (axis 4).	R/W

Built-in analog

The special registers for built-in analog are shown below.
R: Read only, R/W: Read/Write

No.	Name	Description	R/W
SD6020	CH1 Digital output value	This register stores the digital output value.	R
SD6021	CH1 Digital operation value	This register stores the digital operation value.	R
SD6022	CH1 Analog input voltage monitor	This register stores the analog input voltage value.	R
SD6023	CH1 Averaging process setting	This register stores the averaging process setting.	R/W
SD6024	CH1 Time Average/Frequency Average/Moving Average	This register stores the time average/frequency average/moving average.	R/W
SD6026	CH1 Maximum value	This register stores the maximum value.	R
SD6027	CH1 Minimum value	This register stores the minimum value.	R
SD6028	CH1 Scaling upper limit value	This register stores the scaling upper limit value.	R/W
SD6029	CH1 Scaling lower limit value	This register stores the scaling lower limit value.	R/W
SD6030	CH1 Shifting amount to conversion value	This register stores the shifting amount of conversion value.	R/W
SD6031	CH1 Process alarm upper upper limit value	This register stores the process alarm upper upper limit value.	R/W
SD6032	CH1 Process alarm upper lower limit value	This register stores the process alarm upper lower limit value.	R/W
SD6033	CH1 Process alarm lower upper limit value	This register stores the process alarm lower upper limit value.	R/W
SD6034	CH1 Process alarm lower lower limit value	This register stores the process alarm lower lower limit value.	R/W
SD6058	CH1 Latest alarm code	This register stores the latest alarm code.	R
SD6059	CH1 Latest error code	This register stores the latest error code.	R
SD6060	Ch2 Digital output value	This register stores the digital output value.	R
SD6061	CH2 Digital operation value	This register stores the digital operation value.	R

No.	Name	Description	R/W
SD6062	CH2 Analog input voltage monitor	This register stores the analog input voltage value.	R
SD6063	CH2 Averaging process setting	This register stores the averaging process setting.	R/W
SD6064	CH2 Time Average/Frequency Average/Moving Average	This register stores the time average/frequency average/moving average.	R/W
SD6066	CH2 Maximum value	This register stores the maximum value.	R
SD6067	CH2 Minimum value	This register stores the minimum value.	R
SD6068	CH2 Scaling upper limit value	This register stores the scaling upper limit value.	R/W
SD6069	CH2 Scaling lower limit value	This register stores the scaling lower limit value.	R/W
SD6070	CH2 Shifting amount to conversion value	This register stores the shifting amount of conversion value.	R/W
SD6071	CH2 Process alarm upper upper limit value	This register stores the process alarm upper upper limit value.	R/W
SD6072	CH2 Process alarm upper lower limit value	This register stores the process alarm upper lower limit value.	R/W
SD6073	CH2 Process alarm lower upper limit value	This register stores the process alarm lower upper limit value.	R/W
SD6074	CH2 Process alarm lower lower limit value	This register stores the process alarm lower lower limit value.	R/W
SD6098	CH2 Latest alarm code	This register stores the latest alarm code.	R
SD6099	CH2 Latest error code	This register stores the latest error code.	R
SD6180	Digital input value	This register stores the digital input value.	R/W
SD6181	Digital operation value	This register stores the digital operation value.	R
SD6182	Analog output voltage monitor	This register stores the analog output voltage value.	R
SD6183	HOLD/CLEAR setting	This register stores the HOLD/CLEAR setting.	R/W
SD6184	HOLD setting value	This register stores the HOLD setting value.	R/W
SD6188	Scaling upper limit value	This register stores the scaling upper limit value.	R/W
SD6189	Scaling lower limit value	This register stores the scaling lower limit value.	R/W
SD6190	Input value shift amount	This register stores the input value shift amount.	R/W
SD6191	Warning output upper limit value	This register stores the warning output upper limit value.	R/W
SD6192	Warning output lower limit value	This register stores the warning output lower limit value.	R/W
SD6218	Latest alarm code	This register stores the latest alarm code.	R
SD6219	Latest error code	This register stores the latest error code.	R

FX Compatible area

The special registers for FX compatible area are shown below.
R: Read only, R/W: Read/Write

No.	Name	Description	R/W
SD8000	Watchdog timer	This register stores the watchdog timer.	R/W
SD8001	PLC type and system version	This register stores the PLC type and system version.	R
SD8005	Battery voltage	This register stores the battery voltage.	R
SD8006	Low battery voltage	This register stores the low battery voltage.(units: 0.1 V)	R/W
SD8007	Power failure count	This register stores the power failure count.	R
SD8008	Power failure detection period	This register stores the power failure detection period. When the power supply voltage is 200 V AC, the time can be change to 10 to 100 ms.	R/W
SD8010	Current scan time	This register stores the current scan time.	R
SD8011	Minimum scan time	This register stores the minimum scan time.	R
SD8012	Maximum scan time	This register stores the maximum scan time.	R
SD8013	RTC: Seconds	This register stores the seconds data.	R
SD8014	RTC: Minute data	This register stores the minute data.	R
SD8015	RTC: Hour data	This register stores the hour data.	R
SD8016	RTC: Day data	This register stores the day data.	R
SD8017	RTC: Month data	This register stores the month data.	R
SD8018	RTC: Year data	This register stores the year data.	R
SD8019	RTC: Day of week data	This register stores the day of week data.	R
SD8039	Constant scan duration	This register stores the constant scan duration.	R/W
SD8040	ON state number 1	This register stores the ON state number 1.	R/W
SD8041	ON state number 2	This register stores the ON state number 2.	R/W

No.	Name	Description	R/W
SD8042	ON state number 3	This register stores the ON state number 3.	R/W
SD8043	ON state number 4	This register stores the ON state number 4.	R/W
SD8044	ON state number 5	This register stores the ON state number 5.	R/W
SD8045	ON state number 6	This register stores the ON state number 6.	R/W
SD8046	ON state number 7	This register stores the ON state number 7.	R/W
SD8047	ON state number 8	This register stores the ON state number 8.	R/W
SD8049	Lowest active Annunciator	This register stores the lowest active annunciator.	R/W
SD8063	Serial communication error code (ch1)	This register stores the serial communication error code (ch1).	R
SD8067	Operation error	This register stores the error code number of operation error.	R
SD8099	High speed ring counter	This register stores the high speed ring counter count value. (units: 0.1 ms)	R/W
SD8136	PLSY Output number [Low-order]	This register stores the PLSY instruction output pulse number.	R
SD8137	PLSY Output number [High-order]		
SD8140	PLSY Accumulated number of pulses output [Low-order] (axis 1)	This register stores the PLSY instruction accumulated number of pulses output (to axis 1).	R
SD8141	PLSY Accumulated number of pulses output [High-order] (axis 1)		
SD8142	PLSY Accumulated number of pulses output [Low-order] (axis 2)	This register stores the PLSY instruction accumulated number of pulses output (to axis 2).	R
SD8143	PLSY Accumulated number of pulses output [High-order] (axis 2)		
SD8152	Error No. of Inverter communication (ch1)	This register stores the error code of Inverter communication (ch1).	R
SD8154	Error parameter No. of IVBWR (ch1)	This register stores the error parameter No. of IVBWR instruction (ch1).	R
SD8157	Error No. of Inverter communication (ch2)	This register stores the error code of Inverter communication (ch2).	R
SD8159	Error parameter No. of IVBWR (ch2)	This register stores the error parameter No. of IVBWR instruction (ch2).	R
SD8173	Station number	This register stores the station number.	R/W
SD8174	Total number of slave stations	This register stores the total number of slave stations.	R/W
SD8175	Refresh range	This register stores the refresh range.	R
SD8201	Current link scan time	This register stores the current link scan time.	R
SD8202	Maximum link scan time	This register stores the maximum link scan time.	R
SD8203	Number of communication error at master station	This register stores the number of communication error at master station.	R
SD8204	Number of communication error at slave station No.1	This register stores the number of communication error at slave station No.1.	R
SD8205	Number of communication error at slave station No.2	This register stores the number of communication error at slave station No.2.	R
SD8206	Number of communication error at slave station No.3	This register stores the number of communication error at slave station No.3.	R
SD8207	Number of communication error at slave station No.4	This register stores the number of communication error at slave station No.4.	R
SD8208	Number of communication error at slave station No.5	This register stores the number of communication error at slave station No.5.	R
SD8209	Number of communication error at slave station No.6	This register stores the number of communication error at slave station No.6.	R
SD8210	Number of communication error at slave station No.7	This register stores the number of communication error at slave station No.7.	R
SD8211	Code of communication error at master station	This register stores the code of communication error at master station.	R
SD8212	Code of communication error at slave station No.1	This register stores the code of communication error at slave station No.1.	R
SD8213	Code of communication error at slave station No.2	This register stores the code of communication error at slave station No.2.	R
SD8214	Code of communication error at slave station No.3	This register stores the code of communication error at slave station No.3.	R

No.	Name	Description	R/W
SD8215	Code of communication error at slave station No.4	This register stores the code of communication error at slave station No.4.	R
SD8216	Code of communication error at slave station No.5	This register stores the code of communication error at slave station No.5.	R
SD8217	Code of communication error at slave station No.6	This register stores the code of communication error at slave station No.6.	R
SD8218	Code of communication error at slave station No.7	This register stores the code of communication error at slave station No.7.	R
SD8230	Number of communication error at master station	This register stores the number of communication error at master station.	R
SD8231	Number of communication error at slave station No.1	This register stores the number of communication error at slave station No.1.	R
SD8232	Number of communication error at slave station No.2	This register stores the number of communication error at slave station No.2.	R
SD8233	Number of communication error at slave station No.3	This register stores the number of communication error at slave station No.3.	R
SD8234	Number of communication error at slave station No.4	This register stores the number of communication error at slave station No.4.	R
SD8235	Number of communication error at slave station No.5	This register stores the number of communication error at slave station No.5.	R
SD8236	Number of communication error at slave station No.6	This register stores the number of communication error at slave station No.6.	R
SD8237	Number of communication error at slave station No.7	This register stores the number of communication error at slave station No.7.	R
SD8310	RND Random number generation [Low-order]	This register stores the RND random number generation data.	R
SD8311	RND Random number generation [High-order]		
SD8330	Counted number of scans for timing clock output 1	This register stores the scan count for timing clock output 1.	R
SD8331	Counted number of scans for timing clock output 2	This register stores the scan count for timing clock output 2	R
SD8332	Counted number of scans for timing clock output 3	This register stores the scan count for timing clock output 3.	R
SD8333	Counted number of scans for timing clock output 4	This register stores the scan count for timing clock output 4.	R
SD8334	Counted number of scans for timing clock output 5	This register stores the scan count for timing clock output 5.	R
SD8340	Current address [Low-order] (axis 1: pulse units)	This register stores the current address (axis 1: pulse units).	R
SD8341	Current address [High-order] (axis 1: pulse units)		
SD8350	Current address [Low-order] (axis 2: pulse units)	This register stores the current address (axis 2: pulse units).	R
SD8351	Current address [High-order] (axis 2: pulse units)		
SD8360	Current address [Low-order] (axis 3: pulse units)	This register stores the current address (axis 3: pulse units).	R
SD8361	Current address [High-order] (axis 3: pulse units)		
SD8370	Current address [Low-order] (axis 4: pulse units)	This register stores the current address (axis 4: pulse units).	R
SD8371	Current address [High-order] (axis 4: pulse units)		
SD8398	1 ms ring counter [Low-order]	This register stores the 1 ms ring counter.	R
SD8399	1 ms ring counter [High-order]		
SD8402	RS2 amount of remaining data (ch1)/MODBUS communication error code (ch1)	This register stores the amount of remaining data(ch1)/MODBUS communication error code (ch1).	R
SD8403	RS2 receive data points (ch1)/MODBUS communication error details (ch1)	This register stores the receive data points (ch1)/MODBUS communication error details (ch1).	R
SD8405	RS2 communication parameter display (ch1)/MODBUS communication format display (ch1)	This register stores the communication parameter display (ch1)/ MODBUS communication format display (ch1).	R
SD8408	MODBUS communication retry times (ch1)	This register stores the MODBUS communication current retry times (ch1).	R
SD8414	RS2 receive sum (received data) (ch1)	This register stores the ch1 receive sum (received data).	R
SD8415	RS2 receive sum (calculated result) (ch1)	This register stores the ch1 receive sum (calculated result) .	R
SD8416	RS2 send sum (ch1)	This register stores the send sum (ch1).	R
SD8419	Operation mode (ch1)	This register stores the operation mode (ch1).	R
SD8422	RS2 amount of remaining data (ch2)/MODBUS communication error code (ch2)	This register stores the amount of remaining data (ch2)/MODBUS communication error code (ch2).	R
SD8423	RS2 receive data points (ch2)/MODBUS communication error details (ch2)	This register stores the receive data points (ch2)/MODBUS communication error details (ch2).	R

No.	Name	Description	R/W
SD8425	RS2 receive sum (calculated result) (ch2)	This register stores the receive sum (calculated result).	R
SD8428	MODBUS communication retry times (ch2)	This register stores the MODBUS communication current retry times (ch2).	R
SD8434	RS2 receive sum (received data) (ch2)	This register stores the ch2 receive sum (received data).	R
SD8435	RS2 receive sum (calculated result) (ch2)	This register stores the ch2 receive sum (calculated result).	R
SD8436	RS2 send sum (ch2)	This register stores the send sum (ch2).	R
SD8438	Serial communication error code (ch2)	This register stores the serial communication error code (ch2).	R
SD8439	Operation mode (ch2)	This register stores the operation mode (ch2).	R
SD8492	IP address setting [Low-order]	This register stores the IP address.	R/W
SD8493	IP address setting [High-order]		
SD8494	Subnet mask setting [Low-order]	This register stores the subnet mask.	R/W
SD8495	Subnet mask setting [High-order]		
SD8496	Default gateway IP address setting [Low-order]	This register stores the default gateway IP address.	R/W
SD8497	Default gateway IP address setting [High-order]		
SD8498	IP address storage area write error code	This register stores error codes if writing to IP address storage area is failed.	R
SD8499	IP address storage area clear error code	This register stores error codes if clear to IP address storage area is failed.	R

Serial communication

The special registers for serial communication are shown below.

R: Read only, R/W: Read/Write

No.	Name	Description	R/W
SD8500	Serial communication error code (ch1)	This register stores the serial communication error code 1 (ch1).	R
SD8501	Serial communication error details (ch1)	This register stores the serial communication error details 1 (ch1).	R
SD8502	Serial communication setting (ch1)	This register stores the serial communication setting (ch1).	R
SD8503	Serial communication operational mode (ch1)	This register stores the serial communication operational mode 1 (ch1).	R
SD8510	Serial communication error code (ch2)	This register stores the serial communication error code 2 (ch2).	R
SD8511	Serial communication error details (ch2)	This register stores the serial communication error details 2 (ch2).	R
SD8512	Serial communication setting (ch2)	This register stores the serial communication setting (ch2).	R
SD8513	Serial communication operational mode (ch2)	This register stores the serial communication operational mode 2 (ch2).	R
SD8520	Serial communication error code (ch3)	This register stores the serial communication error code 3 (ch3).	R
SD8521	Serial communication error details (ch3)	This register stores the serial communication error details 3 (ch3).	R
SD8522	Serial communication setting (ch3)	This register stores the serial communication setting (ch3).	R
SD8523	Serial communication operational mode (ch3)	This register stores the serial communication operational mode 3 (ch3).	R
SD8530	Serial communication error code (ch4)	This register stores the serial communication error code 4 (ch4).	R
SD8531	Serial communication error details (ch4)	This register stores the serial communication error details 4 (ch4).	R
SD8532	Serial communication setting (ch4)	This register stores the serial communication setting (ch4).	R
SD8533	Serial communication operational mode (ch4)	This register stores the serial communication operational mode 4 (ch4).	R
SD8560	Remaining points of send data (ch1)	This register stores the remaining points of send data (ch1).	R
SD8561	Receive data points monitor (ch1)	This register stores the receive data points monitor (ch1).	R
SD8563	Receive sum (received data) (ch1)	This register stores the receive sum (received data) (ch1).	R
SD8564	Receive sum (received result) (ch1)	This register stores the receive sum (received result) (ch1).	R
SD8565	Send sum (ch1)	This register stores the send sum (ch1).	R
SD8570	Remaining points of send data (ch2)	This register stores the remaining points of send data (ch2).	R
SD8571	Receive data points monitor (ch2)	This register stores the receive data points monitor (ch2).	R
SD8573	Receive sum (received data) (ch2)	This register stores the receive sum (received data) (ch2).	R
SD8574	Receive sum (received result) (ch2)	This register stores the receive sum (received result) (ch2).	R
SD8575	Send sum (ch2)	This register stores the send sum (ch2).	R
SD8580	Remaining points of send data (ch3)	This register stores the remaining points of send data (ch3).	R

No.	Name	Description	R/W
SD8581	Receive data points monitor (ch3)	This register stores the receive data points monitor (ch3).	R
SD8583	Receive sum (received data) (ch3)	This register stores the receive sum (received data) (ch3).	R
SD8584	Receive sum (received result) (ch3)	This register stores the receive sum (received result) (ch3).	R
SD8585	Send sum (ch3)	This register stores the send sum (ch3).	R
SD8590	Remaining points of send data (ch4)	This register stores the remaining points of send data (ch4).	R
SD8591	Receive data points monitor (ch4)	This register stores the receive data points monitor (ch4).	R
SD8593	Receive sum (received data) (ch4)	This register stores the receive sum (received data) (ch4).	R
SD8594	Receive sum (received result) (ch4)	This register stores the receive sum (received result) (ch4).	R
SD8595	Send sum (ch4)	This register stores the send sum (ch4).	R
SD8621	Timeout time (ch1)	This register stores the timeout time (ch1).	R
SD8622	8-bit processing mode (ch1)	This register stores the 8-bit processing mode (ch1).	R
SD8623	Header 1 and 2 (ch1)	This register stores the header 1 and 2 (ch1).	R
SD8624	Header 3 and 4 (ch1)	This register stores the header 3 and 4 (ch1).	R
SD8625	Terminator 1 and 2 (ch1)	This register stores the terminator 1 and 2 (ch1).	R
SD8626	Terminator 3 and 4 (ch1)	This register stores the terminator 3 and 4 (ch1).	R
SD8631	Timeout time (ch2)	This register stores the timeout time (ch2).	R
SD8632	8-bit processing mode (ch2)	This register stores the 8-bit processing mode (ch2).	R
SD8633	Header 1 and 2 (ch2)	This register stores the header 1 and 2 (ch2).	R
SD8634	Header 3 and 4 (ch2)	This register stores the header 3 and 4 (ch2).	R
SD8635	Terminator 1 and 2 (ch2)	This register stores the terminator 1 and 2 (ch2).	R
SD8636	Terminator 3 and 4 (ch2)	This register stores the terminator 3 and 4 (ch2).	R
SD8641	Timeout time (ch3)	This register stores the timeout time (ch3).	R
SD8642	8-bit processing mode (ch3)	This register stores the 8-bit processing mode (ch3).	R
SD8643	Header 1 and 2 (ch3)	This register stores the header 1 and 2 (ch3).	R
SD8644	Header 3 and 4 (ch3)	This register stores the header 3 and 4 (ch3).	R
SD8645	Terminator 1 and 2 (ch3)	This register stores the terminator 1 and 2 (ch3).	R
SD8646	Terminator 3 and 4 (ch3)	This register stores the terminator 3 and 4 (ch3).	R
SD8651	Timeout time (ch4)	This register stores the timeout time (ch4).	R
SD8652	8-bit processing mode (ch4)	This register stores the 8-bit processing mode (ch4).	R
SD8653	Header 1 and 2 (ch4)	This register stores the header 1 and 2 (ch4).	R
SD8654	Header 3 and 4 (ch4)	This register stores the header 3 and 4 (ch4).	R
SD8655	Terminator 1 and 2 (ch4)	This register stores the terminator 1 and 2 (ch4).	R
SD8656	Terminator 3 and 4 (ch4)	This register stores the terminator 3 and 4 (ch4).	R
SD8740	Station number setting (ch1)	This register stores the station number setting (ch1).	R
SD8741	Message frame and form (ch1)	This register stores the message frame and form (ch1).	R
SD8742	Timeout time (ch1)	This register stores the timeout time (ch1).	R
SD8750	Station number setting (ch2)	This register stores the station number setting (ch2).	R/W
SD8751	Message frame and form (ch2)	This register stores the message frame and form (ch2).	R
SD8752	Timeout time (ch2)	This register stores the timeout time (ch2).	R
SD8760	Station number setting (ch3)	This register stores the station number setting (ch3).	R/W
SD8761	Message frame and form (ch3)	This register stores the message frame and form (ch3).	R
SD8762	Timeout time (ch3)	This register stores the timeout time (ch3).	R
SD8770	Station number setting (ch4)	This register stores the station number setting (ch4).	R/W
SD8771	Message frame and form (ch4)	This register stores the message frame and form (ch4).	R
SD8772	Timeout time (ch4)	This register stores the timeout time (ch4).	R
SD8800	Current retry value (ch1)	This register stores the current retry value (ch1).	R
SD8810	Current retry value (ch2)	This register stores the current retry value (ch2).	R
SD8820	Current retry value (ch3)	This register stores the current retry value (ch3).	R
SD8830	Current retry value (ch4)	This register stores the current retry value (ch4).	R
SD8861	Slave node address (ch1)	This register stores the host station number (ch1).	R
SD8862	Slave response timeout (ch1)	This register stores the slave response timeout (ch1).	R
SD8863	Turn around delay (ch1)	This register stores the broadcast delay (ch1).	R
SD8864	Message to message delay (ch1)	This register stores the request to request delay (ch1).	R

No.	Name	Description	R/W
SD8865	Number of retries (ch1)	This register stores the number of retries during timeout (ch1).	R
SD8871	Slave node address (ch2)	This register stores the host station number (ch2).	R
SD8872	Slave response timeout (ch2)	This register stores the slave response timeout (ch2).	R
SD8873	Turn around delay (ch2)	This register stores the broadcast delay (ch12).	R
SD8874	Message to message delay (ch2)	This register stores the request to request delay (ch2).	R
SD8875	Number of retries (ch2)	This register stores the number of retries during timeout (ch2).	R
SD8881	Slave node address (ch3)	This register stores the host station number (ch3).	R
SD8882	Slave response timeout (ch3)	This register stores the slave response timeout (ch3).	R
SD8883	Turn around delay (ch3)	This register stores the broadcast delay (ch3).	R
SD8884	Message to message delay (ch3)	This register stores the request to request delay (ch3).	R
SD8885	Number of retries (ch3)	This register stores the number of retries during timeout (ch3).	R
SD8891	Slave node address (ch4)	This register stores the host station number (ch4).	R
SD8892	Slave response timeout (ch4)	This register stores the slave response timeout (ch4).	R
SD8893	Turn around delay (ch4)	This register stores the broadcast delay (ch4).	R
SD8894	Message to message delay (ch4)	This register stores the request to request delay (ch4).	R
SD8895	Number of retries (ch4)	This register stores the number of retries during timeout (ch4).	R
SD8921	IVBWR instruction error parameter number (ch1)	This register stores the IVBWR instruction error parameter number (ch1).	R
SD8931	IVBWR instruction error parameter number (ch2)	This register stores the IVBWR instruction error parameter number (ch2).	R
SD8941	IVBWR instruction error parameter number (ch3)	This register stores the IVBWR instruction error parameter number (ch3).	R
SD8951	IVBWR instruction error parameter number (ch4)	This register stores the IVBWR instruction error parameter number (ch4).	R
SD8981	Response wait time (ch1)	This register stores the response wait time (ch1).	R
SD8991	Response wait time (ch2)	This register stores the response wait time (ch2).	R
SD9001	Response wait time (ch3)	This register stores the response wait time (ch3).	R
SD9011	Response wait time (ch4)	This register stores the response wait time (ch4).	R
SD9040	Station number	This register stores the station number.	R
SD9041	Total number of slave stations	This register stores the total number of slave stations.	R
SD9043	Current link scan time	This register stores the current link scan time.	R
SD9044	Maximum link scan time	This register stores the maximum link scan time.	R
SD9045	Number of communication error at master station	This register stores the number of communication error at master station.	R
SD9046	Number of communication error at slave station No.1	This register stores the number of communication error at slave station No.1.	R
SD9047	Number of communication error at slave station No.2	This register stores the number of communication error at slave station No.2.	R
SD9048	Number of communication error at slave station No.3	This register stores the number of communication error at slave station No.3.	R
SD9049	Number of communication error at slave station No.4	This register stores the number of communication error at slave station No.4.	R
SD9050	Number of communication error at slave station No.5	This register stores the number of communication error at slave station No.5.	R
SD9051	Number of communication error at slave station No.6	This register stores the number of communication error at slave station No.6.	R
SD9052	Number of communication error at slave station No.7	This register stores the number of communication error at slave station No.7.	R
SD9061	Code of communication error at master station	This register stores the code of communication error at master station.	R
SD9062	Code of communication error at slave station No.1	This register stores the code of communication error at slave station No.1.	R
SD9063	Code of communication error at slave station No.2	This register stores the code of communication error at slave station No.2.	R
SD9064	Code of communication error at slave station No.3	This register stores the code of communication error at slave station No.3.	R

No.	Name	Description	R/W
SD9065	Code of communication error at slave station No.4	This register stores the code of communication error at slave station No.4.	R
SD9066	Code of communication error at slave station No.5	This register stores the code of communication error at slave station No.5.	R
SD9067	Code of communication error at slave station No.6	This register stores the code of communication error at slave station No.6.	R
SD9068	Code of communication error at slave station No.7	This register stores the code of communication error at slave station No.7.	R
SD9080	Station number setting	This register stores the station number setting.	R/W
SD9081	Total slave station number setting	This register stores the total slave station number setting.	R/W
SD9082	Refresh range setting	This register stores the refresh range setting.	R
SD9083	Retry count setting	This register stores the retry count setting.	R
SD9084	Communication time-out setting	This register stores the communication time-out setting.	R

Built-in Ethernet

The special registers for built-in Ethernet are shown below.

R: Read only, R/W: Read/Write

No.	Name	Description	R/W
SD10050	Local node IP address [Low-order]	This register stores the local node IP address.	R
SD10051	Local node IP address [High-order]		
SD10060	Subnet mask [Low-order]	This register stores the subnet mask.	R
SD10061	Subnet mask [High-order]		
SD10064	Default gateway IP address [Low-order]	This register stores the default gateway IP address.	R
SD10065	Default gateway IP address [High-order]		
SD10074	Local node MAC address	This register stores the local node MAC address (5 and 6 bytes).	R
SD10075	Local node MAC address	This register stores the local node MAC address (3 and 4 bytes).	R
SD10076	Local node MAC address	This register stores the local node MAC address (1 and 2 bytes).	R
SD10082	Communication speed setting	This register stores the communication speed setting.	R
SD10084	MELSOFT connection TCP port No.	This register stores the MELSOFT connection TCP port No.	R
SD10086	MELSOFT direct connection port No.	This register stores the MELSOFT direct connection port No.	R
SD10130	Connection No.1 latest error code	This register stores the connection No.1 latest error code.	R
SD10131	Connection No.2 latest error code	This register stores the connection No.2 latest error code.	R
SD10132	Connection No.3 latest error code	This register stores the connection No.3 latest error code.	R
SD10133	Connection No.4 latest error code	This register stores the connection No.4 latest error code.	R
SD10134	Connection No.5 latest error code	This register stores the connection No.5 latest error code.	R
SD10135	Connection No.6 latest error code	This register stores the connection No.6 latest error code.	R
SD10136	Connection No.7 latest error code	This register stores the connection No.7 latest error code.	R
SD10137	Connection No.8 latest error code	This register stores the connection No.8 latest error code.	R
SD10270	Remote password lock status connection No. 1 to 8	b0: Connection No.1 b1: Connection No.2 b2: Connection No.3 b3: Connection No.4 b4: Connection No.5 b5: Connection No.6 b6: Connection No.7 b7: Connection No.8 0: Unlock status/remote password setting none 1: Lock status	R
SD10271	Remote password lock status system port	b2: MELSOFT application communication port (TCP) b3: MELSOFT direct connection 0: Unlock status/remote password setting none 1: Lock status	R
SD10320	Connection 1 continuous unlock failure number of times	This register stores the connection 1 continuous unlock failure number of times.	R
SD10321	Connection 2 continuous unlock failure number of times	This register stores the connection 2 continuous unlock failure number of times.	R

No.	Name	Description	R/W
SD10322	Connection 3 continuous unlock failure number of times	This register stores the connection 3 continuous unlock failure number of times.	R
SD10323	Connection 4 continuous unlock failure number of times	This register stores the connection 4 continuous unlock failure number of times.	R
SD10324	Connection 5 continuous unlock failure number of times	This register stores the connection 5 continuous unlock failure number of times.	R
SD10325	Connection 6 continuous unlock failure number of times	This register stores the connection 6 continuous unlock failure number of times.	R
SD10326	Connection 7 continuous unlock failure number of times	This register stores the connection 7 continuous unlock failure number of times.	R
SD10327	Connection 8 continuous unlock failure number of times	This register stores the connection 8 continuous unlock failure number of times.	R
SD10338	MELSOFT communication port (TCP/IP) continuous unlock failure number of times	This register stores the MELSOFT communication port (TCP/IP) continuous unlock failure number of times.	R
SD10340	MELSOFT direct connection continuous unlock failure number of times	This register stores the MELSOFT direct connection continuous unlock failure number of times.	R
SD10680	Open completion signal	b0: Connection No.1 b1: Connection No.2 b2: Connection No.3 b3: Connection No.4 b4: Connection No.5 b5: Connection No.6 b6: Connection No.7 b7: Connection No.8 0: Close/Open not completed 1: Open completed	R
SD10681	Open request signal	b0: Connection No.1 b1: Connection No.2 b2: Connection No.3 b3: Connection No.4 b4: Connection No.5 b5: Connection No.6 b6: Connection No.7 b7: Connection No.8 0: No open request 1: Open request exists	R
SD10682	Socket communications receive status signal	b0: Connection No.1 b1: Connection No.2 b2: Connection No.3 b3: Connection No.4 b4: Connection No.5 b5: Connection No.6 b6: Connection No.7 b7: Connection No.8 0: No data received 1: Data receiving completed	R
SD10692	Predefined protocol ready	0: — 1: Ready	R
SD10710	Predefined protocol setting data error information protocol number	When a protocol setting data error is detected, stores the protocol number where the error was detected.	R
SD10711	Predefined protocol setting data error information setting type	0 is stored if an error is detected in the packet setting or element setting. 1 is stored if an error is detected in the protocol detailed setting.	R
SD10712	Predefined protocol setting data error information packet number	When an error is detected in the protocol setting data, stores the packet number that detected the error.	R
SD10713	Predefined protocol setting data error information Element number	When an error is detected in the protocol setting data, stores the element number where the error was detected.	R
SD10714	Number of registered predefined protocols	Stores the protocol number of the registered protocol setting data.	R
SD10722	Predefined protocol registration (1 to 16)	Whether protocol setting data is registered or not is stored.	R
SD10723	Predefined protocol registration (17 to 32)		
SD10724	Predefined protocol registration (33 to 48)		
SD10725	Predefined protocol registration (49 to 64)		

No.	Name	Description	R/W
SD10740	Connection No.1 protocol execution status	Stores the status of the protocol being executed at connection No.1. 0: Unexecuted 1: Waiting for transmission 2: Sending 3: Waiting for data reception 4: Receiving 5: Execution completed	R
SD10742	Connection No.1 received data verification result (receive packet No.1)	Stores the verification results of receive packet No.1. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10743	Connection No.1 received data verification result (receive packet No.2)	Stores the verification results of receive packet No.2. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10744	Connection No.1 received data verification result (receive packet No.3)	Stores the verification results of receive packet No.3. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10745	Connection No.1 received data verification result (receive packet No.4)	Stores the verification results of receive packet No.4. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10746	Connection No.1 received data verification result (receive packet No.5)	Stores the verification results of receive packet No.5. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10747	Connection No.1 received data verification result (receive packet No.6)	Stores the verification results of receive packet No.6. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10748	Connection No.1 received data verification result (receive packet No.7)	Stores the verification results of receive packet No.7. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10749	Connection No.1 received data verification result (receive packet No.8)	Stores the verification results of receive packet No.8. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10750	Connection No.1 received data verification result (receive packet No.9)	Stores the verification results of receive packet No.9. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10751	Connection No.1 received data verification result (receive packet No.10)	Stores the verification results of receive packet No.10. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10752	Connection No.1 received data verification result (receive packet No.11)	Stores the verification results of receive packet No.11. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10753	Connection No.1 received data verification result (receive packet No.12)	Stores the verification results of receive packet No.12. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10754	Connection No.1 received data verification result (receive packet No.13)	Stores the verification results of receive packet No.13. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10755	Connection No.1 received data verification result (receive packet No.14)	Stores the verification results of receive packet No.14. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10756	Connection No.1 received data verification result (receive packet No.15)	Stores the verification results of receive packet No.15. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10757	Connection No.1 received data verification result (receive packet No.16)	Stores the verification results of receive packet No.16. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10758	Connection No.1 protocol execution count	Stores the number of protocol executions in Connection No.1. 0: Protocol not executed 1 to 65535: Number of executions	R
SD10759	Connection No.1 protocol cancellation specification	Cancels the protocol executed in connection No.1. 0: No cancellation instruction 1: Cancellation request (set by user) 2: Cancellation completed (set by system)	R/W

No.	Name	Description	R/W
SD10760	Connection No.2 protocol execution status	Stores the status of the protocol being executed at connection No.2. 0: Unexecuted 1: Waiting for transmission 2: Sending 3: Waiting for data reception 4: Receiving 5: Execution completed	R
SD10762	Connection No.2 received data verification result (receive packet No.1)	Stores the verification results of receive packet No.1. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10763	Connection No.2 received data verification result (receive packet No.2)	Stores the verification results of receive packet No.2. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10764	Connection No.2 received data verification result (receive packet No.3)	Stores the verification results of receive packet No.3. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10765	Connection No.2 received data verification result (receive packet No.4)	Stores the verification results of receive packet No.4. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10766	Connection No.2 received data verification result (receive packet No.5)	Stores the verification results of receive packet No.5. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10767	Connection No.2 received data verification result (receive packet No.6)	Stores the verification results of receive packet No.6. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10768	Connection No.2 received data verification result (receive packet No.7)	Stores the verification results of receive packet No.7. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10769	Connection No.2 received data verification result (receive packet No.8)	Stores the verification results of receive packet No.8. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10770	Connection No.2 received data verification result (receive packet No.9)	Stores the verification results of receive packet No.9. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10771	Connection No.2 received data verification result (receive packet No.10)	Stores the verification results of receive packet No.10. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10772	Connection No.2 received data verification result (receive packet No.11)	Stores the verification results of receive packet No.11. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10773	Connection No.2 received data verification result (receive packet No.12)	Stores the verification results of receive packet No.12. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10774	Connection No.2 received data verification result (receive packet No.13)	Stores the verification results of receive packet No.13. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10775	Connection No.2 received data verification result (receive packet No.14)	Stores the verification results of receive packet No.14. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10776	Connection No.2 received data verification result (receive packet No.15)	Stores the verification results of receive packet No.15. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10777	Connection No.2 received data verification result (receive packet No.16)	Stores the verification results of receive packet No.16. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10778	Connection No.2 protocol execution count	Stores the number of protocol executions in connection No.2. 0: Protocol not executed 1 to 65535: Number of executions	R
SD10779	Connection No.2 protocol cancellation specification	Cancels the protocol executed in connection No.2. 0: No cancellation instruction 1: Cancellation request (set by user) 2: Cancellation completed (set by system)	R/W

No.	Name	Description	R/W
SD10780	Connection No.3 protocol execution status	Stores the status of the protocol being executed at connection No.3. 0: Unexecuted 1: Waiting for transmission 2: Sending 3: Waiting for data reception 4: Receiving 5: Execution completed	R
SD10782	Connection No.3 received data verification result (receive packet No.1)	Stores the verification results of receive packet No.1. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10783	Connection No.3 received data verification result (receive packet No.2)	Stores the verification results of receive packet No.2. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10784	Connection No.3 received data verification result (receive packet No.3)	Stores the verification results of receive packet No.3. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10785	Connection No.3 received data verification result (receive packet No.4)	Stores the verification results of receive packet No.4. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10786	Connection No.3 received data verification result (receive packet No.5)	Stores the verification results of receive packet No.5. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10787	Connection No.3 received data verification result (receive packet No.6)	Stores the verification results of receive packet No.6. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10788	Connection No.3 received data verification result (receive packet No.7)	Stores the verification results of receive packet No.7. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10789	Connection No.3 received data verification result (receive packet No.8)	Stores the verification results of receive packet No.8. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10790	Connection No.3 received data verification result (receive packet No.9)	Stores the verification results of receive packet No.9. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10791	Connection No.3 received data verification result (receive packet No.10)	Stores the verification results of receive packet No.10. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10792	Connection No.3 received data verification result (receive packet No.11)	Stores the verification results of receive packet No.11. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10793	Connection No.3 received data verification result (receive packet No.12)	Stores the verification results of receive packet No.12. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10794	Connection No.3 received data verification result (receive packet No.13)	Stores the verification results of receive packet No.13. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10795	Connection No.3 received data verification result (receive packet No.14)	Stores the verification results of receive packet No.14. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10796	Connection No.3 received data verification result (receive packet No.15)	Stores the verification results of receive packet No.15. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10797	Connection No.3 received data verification result (receive packet No.16)	Stores the verification results of receive packet No.16. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10798	Connection No.3 protocol execution count	Stores the number of protocol executions in connection No.3. 0: Protocol not executed 1 to 65535: Number of executions	R
SD10799	Connection No.3 protocol cancellation specification	Cancels the protocol executed in connection No.3. 0: No cancellation instruction 1: Cancellation request (set by user) 2: Cancellation completed (set by system)	R/W

No.	Name	Description	R/W
SD10800	Connection No.4 protocol execution status	Stores the status of the protocol being executed at connection No.4. 0: Unexecuted 1: Waiting for transmission 2: Sending 3: Waiting for data reception 4: Receiving 5: Execution completed	R
SD10802	Connection No.4 received data verification result (receive packet No.1)	Stores the verification results of receive packet No.1. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10803	Connection No.4 received data verification result (receive packet No.2)	Stores the verification results of receive packet No.2. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10804	Connection No.4 received data verification result (receive packet No.3)	Stores the verification results of receive packet No.3. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10805	Connection No.4 received data verification result (receive packet No.4)	Stores the verification results of receive packet No.4. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10806	Connection No.4 received data verification result (receive packet No.5)	Stores the verification results of receive packet No.5. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10807	Connection No.4 received data verification result (receive packet No.6)	Stores the verification results of receive packet No.6. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10808	Connection No.4 received data verification result (receive packet No.7)	Stores the verification results of receive packet No.7. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10809	Connection No.4 received data verification result (receive packet No.8)	Stores the verification results of receive packet No.8. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10810	Connection No.4 received data verification result (receive packet No.9)	Stores the verification results of receive packet No.9. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10811	Connection No.4 received data verification result (receive packet No.10)	Stores the verification results of receive packet No.10. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10812	Connection No.4 received data verification result (receive packet No.11)	Stores the verification results of receive packet No.11. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10813	Connection No.4 received data verification result (receive packet No.12)	Stores the verification results of receive packet No.12. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10814	Connection No.4 received data verification result (receive packet No.13)	Stores the verification results of receive packet No.13. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10815	Connection No.4 received data verification result (receive packet No.14)	Stores the verification results of receive packet No.14. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10816	Connection No.4 received data verification result (receive packet No.15)	Stores the verification results of receive packet No.15. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10817	Connection No.4 received data verification result (receive packet No.16)	Stores the verification results of receive packet No.16. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10818	Connection No.4 protocol execution count	Stores the number of protocol executions in connection No.4. 0: Protocol not executed 1 to 65535: Number of executions	R
SD10819	Connection No.4 protocol cancellation specification	Cancels the protocol executed in connection No.4. 0: No cancellation instruction 1: Cancellation request (set by user) 2: Cancellation completed (set by system)	R/W

No.	Name	Description	R/W
SD10820	Connection No.5 protocol execution status	Stores the status of the protocol being executed at connection No.5. 0: Unexecuted 1: Waiting for transmission 2: Sending 3: Waiting for data reception 4: Receiving 5: Execution completed	R
SD10822	Connection No.5 received data verification result (receive packet No.1)	Stores the verification results of receive packet No.1. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10823	Connection No.5 received data verification result (receive packet No.2)	Stores the verification results of receive packet No.2. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10824	Connection No.5 received data verification result (receive packet No.3)	Stores the verification results of receive packet No.3. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10825	Connection No.5 received data verification result (receive packet No.4)	Stores the verification results of receive packet No.4. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10826	Connection No.5 received data verification result (receive packet No.5)	Stores the verification results of receive packet No.5. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10827	Connection No.5 received data verification result (receive packet No.6)	Stores the verification results of receive packet No.6. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10828	Connection No.5 received data verification result (receive packet No.7)	Stores the verification results of receive packet No.7. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10829	Connection No.5 received data verification result (receive packet No.8)	Stores the verification results of receive packet No.8. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10830	Connection No.5 received data verification result (receive packet No.9)	Stores the verification results of receive packet No.9. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10831	Connection No.5 received data verification result (receive packet No.10)	Stores the verification results of receive packet No.10. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10832	Connection No.5 received data verification result (receive packet No.11)	Stores the verification results of receive packet No.11. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10833	Connection No.5 received data verification result (receive packet No.12)	Stores the verification results of receive packet No.12. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10834	Connection No.5 received data verification result (receive packet No.13)	Stores the verification results of receive packet No.13. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10835	Connection No.5 received data verification result (receive packet No.14)	Stores the verification results of receive packet No.14. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10836	Connection No.5 received data verification result (receive packet No.15)	Stores the verification results of receive packet No.15. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10837	Connection No.5 received data verification result (receive packet No.16)	Stores the verification results of receive packet No.16. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10838	Connection No.5 protocol execution count	Stores the number of protocol executions in connection No.5. 0: Protocol not executed 1 to 65535: Number of executions	R
SD10839	Connection No.5 protocol cancellation specification	Cancels the protocol executed in connection No.5. 0: No cancellation instruction 1: Cancellation request (set by user) 2: Cancellation completed (set by system)	R/W

No.	Name	Description	R/W
SD10840	Connection No.6 protocol execution status	Stores the status of the protocol being executed at connection No.6. 0: Unexecuted 1: Waiting for transmission 2: Sending 3: Waiting for data reception 4: Receiving 5: Execution completed	R
SD10842	Connection No.6 received data verification result (receive packet No.1)	Stores the verification results of receive packet No.1. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10843	Connection No.6 received data verification result (receive packet No.2)	Stores the verification results of receive packet No.2. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10844	Connection No.6 received data verification result (receive packet No.3)	Stores the verification results of receive packet No.3. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10845	Connection No.6 received data verification result (receive packet No.4)	Stores the verification results of receive packet No.4. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10846	Connection No.6 received data verification result (receive packet No.5)	Stores the verification results of receive packet No.5. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10847	Connection No.6 received data verification result (receive packet No.6)	Stores the verification results of receive packet No.6. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10848	Connection No.6 received data verification result (receive packet No.7)	Stores the verification results of receive packet No.7. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10849	Connection No.6 received data verification result (receive packet No.8)	Stores the verification results of receive packet No.8. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10850	Connection No.6 received data verification result (receive packet No.9)	Stores the verification results of receive packet No.9. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10851	Connection No.6 received data verification result (receive packet No.10)	Stores the verification results of receive packet No.10. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10852	Connection No.6 received data verification result (receive packet No.11)	Stores the verification results of receive packet No.11. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10853	Connection No.6 received data verification result (receive packet No.12)	Stores the verification results of receive packet No.12. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10854	Connection No.6 received data verification result (receive packet No.13)	Stores the verification results of receive packet No.13. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10855	Connection No.6 received data verification result (receive packet No.14)	Stores the verification results of receive packet No.14. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10856	Connection No.6 received data verification result (receive packet No.15)	Stores the verification results of receive packet No.15. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10857	Connection No.6 received data verification result (receive packet No.16)	Stores the verification results of receive packet No.16. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10858	Connection No.6 protocol execution count	Stores the number of protocol executions in connection No.6. 0: Protocol not executed 1 to 65535: Number of executions	R
SD10859	Connection No.6 protocol cancellation specification	Cancels the protocol executed in connection No.6. 0: No cancellation instruction 1: Cancellation request (set by user) 2: Cancellation completed (set by system)	R/W

No.	Name	Description	R/W
SD10860	Connection No.7 protocol execution status	Stores the status of the protocol being executed at connection No.7. 0: Unexecuted 1: Waiting for transmission 2: Sending 3: Waiting for data reception 4: Receiving 5: Execution completed	R
SD10862	Connection No.7 received data verification result (receive packet No.1)	Stores the verification results of receive packet No.1. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10863	Connection No.7 received data verification result (receive packet No.2)	Stores the verification results of receive packet No.2. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10864	Connection No.7 received data verification result (receive packet No.3)	Stores the verification results of receive packet No.3. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10865	Connection No.7 received data verification result (receive packet No.4)	Stores the verification results of receive packet No.4. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10866	Connection No.7 received data verification result (receive packet No.5)	Stores the verification results of receive packet No.5. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10867	Connection No.7 received data verification result (receive packet No.6)	Stores the verification results of receive packet No.6. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10868	Connection No.7 received data verification result (receive packet No.7)	Stores the verification results of receive packet No.7. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10869	Connection No.7 received data verification result (receive packet No.8)	Stores the verification results of receive packet No.8. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10870	Connection No.7 received data verification result (receive packet No.9)	Stores the verification results of receive packet No.9. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10871	Connection No.7 received data verification result (receive packet No.10)	Stores the verification results of receive packet No.10. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10872	Connection No.7 received data verification result (receive packet No.11)	Stores the verification results of receive packet No.11. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10873	Connection No.7 received data verification result (receive packet No.12)	Stores the verification results of receive packet No.12. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10874	Connection No.7 received data verification result (receive packet No.13)	Stores the verification results of receive packet No.13. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10875	Connection No.7 received data verification result (receive packet No.14)	Stores the verification results of receive packet No.14. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10876	Connection No.7 received data verification result (receive packet No.15)	Stores the verification results of receive packet No.15. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10877	Connection No.7 received data verification result (receive packet No.16)	Stores the verification results of receive packet No.16. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10878	Connection No.7 protocol execution count	Stores the number of protocol executions in connection No.7. 0: Protocol not executed 1 to 65535: Number of executions	R
SD10879	Connection No.7 protocol cancellation specification	Cancels the protocol executed in connection No.7. 0: No cancellation instruction 1: Cancellation request (set by user) 2: Cancellation completed (set by system)	R/W

No.	Name	Description	R/W
SD10880	Connection No.8 protocol execution status	Stores the status of the protocol being executed at connection No.8. 0: Unexecuted 1: Waiting for transmission 2: Sending 3: Waiting for data reception 4: Receiving 5: Execution completed	R
SD10882	Connection No.8 received data verification result (receive packet No.1)	Stores the verification results of receive packet No.1. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10883	Connection No.8 received data verification result (receive packet No.2)	Stores the verification results of receive packet No.2. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10884	Connection No.8 received data verification result (receive packet No.3)	Stores the verification results of receive packet No.3. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10885	Connection No.8 received data verification result (receive packet No.4)	Stores the verification results of receive packet No.4. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10886	Connection No.8 received data verification result (receive packet No.5)	Stores the verification results of receive packet No.5. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10887	Connection No.8 received data verification result (receive packet No.6)	Stores the verification results of receive packet No.6. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10888	Connection No.8 received data verification result (receive packet No.7)	Stores the verification results of receive packet No.7. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10889	Connection No.8 received data verification result (receive packet No.8)	Stores the verification results of receive packet No.8. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10890	Connection No.8 received data verification result (receive packet No.9)	Stores the verification results of receive packet No.9. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10891	Connection No.8 received data verification result (receive packet No.10)	Stores the verification results of receive packet No.10. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10892	Connection No.8 received data verification result (receive packet No.11)	Stores the verification results of receive packet No.11. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10893	Connection No.8 received data verification result (receive packet No.12)	Stores the verification results of receive packet No.12. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10894	Connection No.8 received data verification result (receive packet No.13)	Stores the verification results of receive packet No.13. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10895	Connection No.8 received data verification result (receive packet No.14)	Stores the verification results of receive packet No.14. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10896	Connection No.8 received data verification result (receive packet No.15)	Stores the verification results of receive packet No.15. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10897	Connection No.8 received data verification result (receive packet No.16)	Stores the verification results of receive packet No.16. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15)	R
SD10898	Connection No.8 protocol execution count	Stores the number of protocol executions in connection No.8. 0: Protocol not executed 1 to 65535: Number of executions	R
SD10899	Connection No.8 protocol cancellation specification	Cancels the protocol executed in connection No.8. 0: No cancellation instruction 1: Cancellation request (set by user) 2: Cancellation completed (set by system)	R/W

MEMO

Appendix 1.4 Error Code

The CPU module stores error code in special register (SD) upon detection of an error using the self-diagnostics function. The error details and cause can be identified by checking the error code. The error code can be checked in either of the following ways.

- Module diagnostics of engineering tool (MELSEC iQ-F FX5 User's Manual (Startup))
- Special register (SD0 (latest self-diagnostics error code), SD10 to SD25 (self-diagnostics error code)) (Page 319 Special Register List)

This section describes errors that may occur in the CPU module and actions to be taken for the errors.

Error code system

All error codes are given in hexadecimal format (4 digits) (16-bit unsigned integer). The following table lists the error detection type and the error code ranges

Error detection type	Range	Description
Detection by the self-diagnostics function of each module	0001H to 3FFFH	Error code specific to each module, such as self-diagnostics errors
Detected during communication between CPU modules	4000H to 4FFFH	CPU module error
	7000H to 7FFFH	MELSEC iQ-F FX5 User's Manual (Serial Communication) MELSEC iQ-F FX5 User's Manual (MODBUS Communication)
	C000H to CFBFH	MELSEC iQ-F FX5 User's Manual (Ethernet Communication)

Detailed information

Upon detection of error through self-diagnostics function, the detailed information of the error cause is stored all together. The following detailed information is added to each error code (up to two types of information are stored for each error code. The types differ depending on error code.) Detailed information 1 to 2 of the latest error code(s) can be checked with special register (SD).

Detailed information	Item	Description
Detailed information 1	Error location information*1	Information on the location in a program
	Drive/File information	Information on drive names and file names
	Parameter information	The information for the parameter, such as parameter storage location and parameter type, is indicated.
	System configuration information	The information for the system configuration, such as I/O No is indicated.
	Frequency information	This section describes the information for frequency such as the write frequency into memory.
	Time information	The information for the time is indicated.
Detailed information 2	Drive/File information	Information on drive names and file names
	Annunciator information	Information about annunciators
	Parameter information	The information for the parameter, such as parameter storage location and parameter type, is indicated.
	System configuration information	The information for the system configuration, such as I/O No is indicated.

*1 The step No, which is displayed in the program position information, is the step No that is counted from the head of the file. It might be sometimes different from the step No of the program which is displayed in error jump of engineering tool.

Operation when an error occurs

There are two types of errors: continuation errors and stop errors.

Stop error

If a stop error occurs, the CPU module stops its operation and the operating state will be in STOP. Modules can communicate with the CPU module even after a stop error occurs in the CPU module.

Continuation error

If a continuation error occurs, the CPU module continues its operation. (The operating state will remain the same.)

How to clear errors

Continuation errors can be cleared.

Errors that can be cleared

Error code	Error name
1080H	ROM write count error
1090H	Battery error
1800H	Annunciator ON
1810H, 1811H	Operation error
1900H	Constant scan time error
1920H	IP address setting error
1921H	IP address writing/clear request simultaneous detection
1FE0 to 1FE6H, 2008H	Module configuration error
2120H, 2121H	Memory card error
2400H	Module verification error
2440H, 2441H	Module major error
2522H	Invalid interrupt
2801H	Module specification error
2820H, 2821H, 2822H, 2823H	Device specification error
2840H	File name specification error
3360H to 3362H	Nesting depth error
3380H	Pointer execution error
3400H to 3406H, 3420H, 3500H, 3502H to 3506H, 350AH, 350C to 350FH, 3510H to 351EH, 3580H, 3581H, 3600H, 3611H to 361CH, 3621H to 362CH, 3631H to 363CH, 3641H to 364CH, 3651H to 365CH, 3661H to 366CH, 3671H to 367CH, 3681H to 368CH, 3691H to 369CH, 36A1H to 36ACH, 36B1H to 36BCH, 36F0H	Operation error
3780H	High-speed comparison table maximum excess error
3781H	Preset value range outside error

How to clear errors

Errors can be cleared in two ways:

■Using the engineering tool

Clear errors with the module diagnostics function of engineering tool. (Refer to GX Works3 Operating Manual)

■Using SM/SD

Clear errors by operating SM/SD.

1. Check SD0 (Latest self-diagnostics error code) to identify what errors are detected.
2. Clear the cause of each of the currently detected continuation errors.
3. Turn off and on SM50 (error reset).

Precautions

This section describes some precautions to take when using the error clear function:

- Since the function clears all of the currently detected continuation errors at once, errors that should not yet be cleared may be cleared.
- Use the RST instruction to reset each annunciator individually.

List of error codes

Self-diagnostics error codes of the CPU module (1000H to 3FFFH)

The following table lists the error codes detected by the self-diagnostics function of the CPU module.

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
1080H	ROM write count error	• The number of writes to the data memory exceeded 20,000 times.	• Replace the CPU module.	Frequency information	At write
1090H	Battery error	• Low battery voltage was detected. An error was also detected in a battery latched (backed) device.	• Check the connection of the battery. • Replace the battery as soon as possible.	—	At END instruction execution
1130H	IP address duplication error	• Overlapping IP addresses were detected.	• Check the IP address.	—	—
1800H	Annunciator ON	• An annunciator that was turned ON by the SET F instruction or OUT F instruction was detected.	• Check the program of that number (annunciator number).	Error location information and annunciator information	At instruction execution
1810H	Operation error	• The channel specified by instructions using communication functions or built-in I/O is already used by other instructions.	• Verify that the channel specified by instructions using communication functions or built-in I/O is not used by other instructions.	Error location information	At instruction execution
1811H	Operation error	• The number of times that applied instructions are used in the program exceeded the specified limit.	• Verify that the number of times that applied instructions are used in the program does not exceed the specified limit.	Error location information	At instruction execution
1900H	Constant scan time error	• The scan time exceeded the constant scan setting value.	• Check and correct the constant scan time setting. Recheck the setting time of the constant scan.	Time information	At END instruction execution
1920H	IP address setting error	• Values such as the IP address setting (SD8492 to SD8497) are outside the set range.	• Recheck the values such as the IP address setting (SD8492 to SD8497).	—	At END instruction execution
1921H	IP address writing/ clear request simultaneous detection	• Write request and clear request (M8492 and SM8495) turned from OFF to ON simultaneously.	• Verify that write request and clear request (SM8492 and MS8495) do not turn from OFF to ON simultaneously.	—	At END instruction execution
1930H	Online change error	• An error was detected when writing was executed during RUN.	• Set the CPU module to STOP and write a set of project data.	—	At END instruction execution
1931H	Online change error	• An error was detected when writing was executed during RUN.	• Set the CPU module to STOP and write a set of project data.	—	At END instruction execution
1FE0H	Module configuration error	• The number of I/O points specified in the I/O assignment setting of the parameters is different from that of the module connected.	• Make sure that the parameters are consistent with the connections.	System configuration information	At power-on, at RESET
1FE1H	Module configuration error	• The module position specified in the I/O assignment setting of the parameters is different from that of the module connected.	• Make sure that the parameters are consistent with the connections.	System configuration information	At power-on, at RESET
1FE2H	Module configuration error	• No parameters available for the module connected exist.	• Make sure that the parameters are consistent with the connections.	System configuration information	At power-on, at RESET
1FE3H	Module configuration error	• The module specified in the I/O assignment setting of the parameters is not connected.	• Make sure that the parameters are consistent with the connections.	System configuration information	At power-on, at RESET
1FE4H	Module configuration error	• Parameters for a standard input/output module are set to a high-speed pulse input/output module.	• Make sure that the parameters are consistent with the connections.	System configuration information	At power-on, at RESET
1FE5H	Module configuration error	• The I/O numbers of the reserved module specified in the I/O assignment setting of the parameters overlap those of other modules.	• Make sure that the parameters are consistent with the connections.	System configuration information	At power-on, at RESET

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
1FE6H	Module configuration error	• The I/O method of the input/output module is different.	• Make sure that the parameters are consistent with the connections.	System configuration information	At power-on, at RESET
1FE7H	Module configuration error	• The type of the CPU module is different.	• Make sure that the parameters are consistent with the connections.	System configuration information	At power-on, at RESET
2003H	Module configuration error	• The model of the module connected is different from that of the module set in the parameters.	• Make sure the model of the module to be set is consistent with the parameters of the module connected.	System configuration information	At power-on, at RESET
2008H	Module configuration error	• The total number of I/O points (excluding remote I/O) exceeded 256.	• Do not use more than 256 I/O points in programs.	System configuration information	At power-on, at RESET
2042H	CPU module configuration error	• The number of input, output, input/output, and intelligent function modules connected is equal to or greater than 17. • The number of communication adapters connected is equal to or greater than 3. • The number of analog adapters connected is equal to or greater than 5. • The number of extension power supply modules connected is equal to or greater than 3. • The number of expansion boards connected is equal to or greater than 2.	• Use up to 16 input, output, input/output, and intelligent function modules. • Use up to 2 communication adapters. • Use up to 4 analog adapters. • Use up to 2 extension power supply modules. • Use up to 1 expansion board.	System configuration information	At power-on, at RESET
20E0H	Invalid module detection	• An unsupported module was detected.	• Verify that the version of the CPU module is compatible with the module where the error was detected. • If the version of the CPU module is correct, there may be a malfunction in the connected module. Replace the connected module.	System configuration information	At power-on, at RESET
2120H	Memory card error	• An SD memory card error was detected. • The SD memory card may have been removed without the SD memory card disabled.	• Check the connection of the SD memory card. If the problem persists, there may be a malfunction in the SD memory card or CPU module.	Drive/file information	Always
2121H	Memory card error	• An SD memory card error was detected. • The SD memory card may not be correctly formatted.	• Format the SD memory card. If the problem persists, there may be a malfunction in the SD memory card or CPU module.	Drive/file information	Always
2180H	Invalid file	• An error was found in the data of the file.	• Recreate the file.	Drive/file information	At power-on, at RESET, at STOP → RUN state
21A0H	File specification error	• The file specified in the parameters does not exist.	• Rewrite the project.	Drive/file information Parameter information	At power-on, at RESET, at STOP → RUN state
2220H	Parameter error	• The contents of the parameters are corrupted.	• Rewrite the project.	Parameter information	At power-on, at RESET
2221H	Parameter error	• The parameter set value is out of range.	• Modify the parameter set value and rewrite the project.	Parameter information	At power-on, at RESET
2222H	Parameter error	• The parameter set value is out of range.	• Modify the parameter set value and rewrite the project.	Parameter information	At power-on, at RESET
2241H	Parameter error (module)	• The module parameter settings and the target module are different.	• Modify the module parameter set value and rewrite the project.	Parameter information	At power-on, at RESET
2300H	Security key authentication error	• The security key locking the program does not match the security key written in the CPU module.	• Write the correct security key to the CPU module.	Drive/file information	At power-on, at RESET, at STOP → RUN state

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
2301H	Security key authentication error	• The program is locked by the security key, but the security key is not written in the CPU module.	• Write the security key to the CPU module.	Drive/file information	At power-on, at RESET, at STOP → RUN state
2302H	Security key authentication error	• The security key written in the CPU module is corrupted.	• Rewrite the security key to the CPU module.	—	At power-on, at RESET, at STOP → RUN state
2320H	Remote password setting error	• A module supporting remote passwords is not connected to the module number specified in the remote password parameter.	• Recheck the remote password parameter setting or module configuration.	System configuration information	At power-on, at RESET
2400H	Module verification error	• The power of a module connected is OFF or a connection error has been detected.	• Verify that the connected module is powered on. • Verify that extension cables are correctly connected. • Implement anti-noise measures. • If there is no problem, there may be a malfunction in the connected module. Replace the connected module.	System configuration information	Always
2401H	Module verification error	• A module was connected during operation.	• Avoid connecting a module during operation.	System configuration information	Always
2440H	Module major error	• The communication procedure with a module failed during initial processing.	• Verify that extension cables are correctly connected. • Verify that the version of the CPU module is compatible with the module where the error was detected. • If the version of the CPU module is correct, there may be a malfunction in the connected module. Replace the connected module.	System configuration information	At power-on, at RESET
2441H	Module major error	• The communication procedure with a module failed when an instruction was executed.	• Review the program and check the contents of the operands used in the applied instructions. • Verify that the specified buffer memory exists in the counterpart equipment. • Verify that extension cables are correctly connected.	Error location information and system configuration information	At instruction execution
2500H	WDT error	• The initial scan time exceeded the set value of execution monitor time.	• Recheck the set value of execution monitor time or program.	Time information	Always
2501H	WDT error	• The scan time of the second and subsequent scans exceeded the set value of execution monitor time.	• Recheck the set value of execution monitor time or program.	Time information	Always
2522H	Invalid interrupt	• An interrupt request was detected from a module that does not have an interrupt pointer specified in the parameters.	• Correctly set the interrupt pointer for module interrupt.	System configuration information	At interrupt occurrence
2801H	Module specification error	• Verify that the module with the specified module number exists.	• Specify the correct module number.	Error location information and system configuration information	At instruction execution
2820H	Device specification error	• A device used as an instruction operand is outside the allowable device range.	• Check the device range and modify the program.	Error location information	At power-on, at RESET, at instruction execution
2821H	Device specification error	• There are duplicate devices used as an instruction operand.	• Check the range of devices used by each operand and modify the program.	Error location information	At instruction execution
2822H	Device specification error	• A device or modification that cannot be used as an instruction operand is used.	• Check the usage of the instruction and modify the program.	Error location information	At power-on, at RESET

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
2823H	Device specification error	• Verify that the specified module has buffer memory. • Check the buffer memory range of the specified module. • Verify that the size specified from the specified buffer memory number is within the buffer memory range.	• Review the program or check the contents of the operands used in applied instructions. • Verify that the specified buffer memory exists in the counterpart equipment.	Error location information	At instruction execution
2840H	File name specification error	• The program file specified does not exist.	• Rewrite the project.	Error location information	At power-on, at RESET
3000H	Boot function execution error	• An error was found in the boot file.	• Replace the boot file in the SD memory card with the correct file and turn the PLC power ON again.	Drive/file information	At power-on, at RESET
3001H	Boot function execution error	• Formatting failed during booting.	• Reset the CPU module, and then execute the boot function again. If the same error appears, the hardware of the CPU module may be malfunctioning. Consult your local Mitsubishi Electric representative.	Drive/file information	At power-on, at RESET
3003H	Boot function execution error	• A mismatch between the file password 32 of the boot source file and that of the boot destination file was detected during booting.	• Check the file password 32 of the boot source file.	Drive/file information	At power-on, at RESET
3004H	Boot function execution error	• The capacity of the boot destination data memory becomes insufficient due to booting.	• Allow sufficient capacity on the boot destination or recheck the file size of the boot source.	Drive/file information	At power-on, at RESET
3005H	Boot function execution error	• A mismatch between the security information of the boot source file and that of the boot destination file was detected during booting.	• Check the security information of the boot source file.	Drive/file information	At power-on, at RESET
3048H	Online change error	• An error was detected when writing was executed during RUN.	• Set the CPU module to STOP and write a set of project data.	—	At END instruction execution
3049H	Online change error	• An error was detected when writing was executed during RUN.	• Set the CPU module to STOP and write a set of project data.	—	At END instruction execution
304AH	Online change error	• An error was detected when writing was executed during RUN.	• Set the CPU module to STOP and write a set of project data.	—	At END instruction execution
304BH	Online change error	• An error was detected when writing was executed during RUN.	• Set the CPU module to STOP and write a set of project data.	—	At END instruction execution
3050H	System bus error	• Communication with the module failed due to power discontinuity or the like.	• Verify that the connected module is powered on. • Verify that extension cables are correctly connected. • Verify that the version of the CPU module is compatible with the module where the error was detected. • Implement anti-noise measures. • If there is no problem, there may be a malfunction in the connected module or in the extension cables.	System configuration information	At power-on, at RESET
3056H	System bus error	• A timeout occurred during communication with a connected module when an instruction was executed.	• Verify that extension cables are correctly connected. • Verify that the version of the CPU module is compatible with the module where the error was detected. • Implement anti-noise measures. • If there is no problem, there may be a malfunction in the connected module or in the extension cables. • When an error occurs alongside positioning function, positioning may not stop at the specified position. After removing the cause of the system bus error, perform zero return and clear error.	Error location information and system configuration information	At instruction execution

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
3057H	System bus error	• A timeout occurred during communication with a connected module during system processing.	• Verify that extension cables are correctly connected. • Verify that the version of the CPU module is compatible with the module where the error was detected. • Implement anti-noise measures. • If there is no problem, there may be a malfunction in the connected module or in the extension cables. • When an error occurs alongside positioning function, positioning may not stop at the specified position. After removing the cause of the system bus error, perform zero return and clear error.	System configuration information	At END instruction execution, at interrupt occurrence, at module access
3060H	System bus error	• A signal error was detected with a connected module when an instruction was executed.	• Verify that extension cables are correctly connected. • Verify that the version of the CPU module is compatible with the module where the error was detected. • Implement anti-noise measures. • If there is no problem, there may be a malfunction in the connected module or in the extension cables. • When an error occurs alongside positioning function, positioning may not stop at the specified position. After removing the cause of the system bus error, perform zero return and clear error.	Error location information	At instruction execution
3061H	System bus error	• A signal error was detected during system processing.	• Verify that extension cables are correctly connected. • Verify that the version of the CPU module is compatible with the module where the error was detected. • Implement anti-noise measures. • If there is no problem, there may be a malfunction in the connected module or in the extension cables. • When an error occurs alongside positioning function, positioning may not stop at the specified position. After removing the cause of the system bus error, perform zero return and clear error.	System configuration information	At instruction execution
3142H	Program structure error	• The temporary area was used incorrectly.	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and check the program. The step number displayed in the error location information is counted from the top of the file. (It may be different from the step number in the program displayed by the jump function.)	Error location information	At instruction execution
3200H	Program execution error	• The device/label assignment does not match the device/label assignment in the program. (After the device assignment was changed, only the parameters were written to the CPU module.)	• If the index modification setting of the PLC parameter is changed, write the parameter and program file to the CPU module at the same time.	Drive/file information	At power-on, at RESET
3202H	Program execution error	• The program file is invalid or the file does not contain a program.	• Write the correct program file.	Drive/file information	At power-on, at RESET
3203H	Program execution error	• No program file exists.	• Write a program file.	Drive/file information	At power-on, at RESET
3210H	Program execution error	• A program with more than 64 k steps was written.	• Reduce the number of steps in the program.	—	At power-on, at RESET

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
3211H	Program execution error	• An FB program larger than the internal memory capacity was written.	• Reduce the number of steps in the FB program.	—	At power-on, at RESET
3212H	Program execution error	• No program setting is found in the parameters.	• Specify the program to execute in the parameters.	—	At power-on, at RESET
3213H	Program execution error	• The parameter set value is out of range.	• To use this parameter, a new version of the CPU module is required. Replace the CPU module or perform version upgrade.	Parameter information	At power-on, at RESET
3302H	Pointer setting error	• Duplicate pointers are programmed.	• Modify the program to not use duplicate pointers in a program.	Error location information	At power-on, at RESET
3320H	Interrupt pointer setting error	• Duplicate interrupt pointers are programmed.	• Modify the program to not use duplicate interrupt pointers in a program.	Error location information	At power-on, at RESET
3340H	FOR-NEXT instruction error	• The relationship between FOR and NEXT instructions is invalid.	• Make sure that FOR and NEXT instructions are each executed the same number of times. In addition, check the FOR syntax for any invalid jump instructions.	Error location information	At END instruction execution
3341H	FOR-NEXT instruction error	• The relationship between FOR and NEXT instructions is invalid.	• Make sure that FOR and NEXT instructions are each executed the same number of times. In addition, check syntax for any invalid jump instructions.	Error location information	At END instruction execution
3342H	FOR-NEXT instruction error	• A BREAK instruction was executed outside the FOR syntax.	• The BREAK instruction must be executed inside the FOR syntax.	Error location information	At instruction execution
3360H	Nesting depth error	• The number of nesting levels of subroutine calls is invalid.	• Make sure that the number of nesting levels is 16 or lower. In addition, check subroutine programs for any invalid jump instructions.	Error location information	At instruction execution
3361H	Nesting depth error	• The number of nesting levels of FOR instructions is invalid.	• Make sure that the number of nesting levels is 16 or lower. In addition, check the FOR syntax for any invalid jump instructions.	Error location information	At instruction execution
3362H	Nesting depth error	• The number of nesting levels of DI instructions is invalid.	• Make sure that the number of nesting levels is 16 or lower. In addition, check the relationship between DI and EI instructions.	Error location information	At instruction execution
3380H	Pointer execution error	• There is no pointer to the jump destination.	• Specify the correct jump destination in the program.	Error location information	At instruction execution
3381H	Pointer execution error	• There is an END, FEND, GOEND, or STOP instruction in a subroutine program.	• The END, FEND, GOEND, and STOP instructions can be executed only in the main routine program.	Error location information	At END instruction execution
3382H	Pointer execution error	• A RET instruction was executed without a CALL or XCALL instruction executed.	• Check where there is any invalid jump to subroutine programs.	Error location information	At instruction execution
33D0H	Temporary area exceeded	• The temporary area was used incorrectly.	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and check the program. The step number displayed in the error location information is counted from the top of the file. (It may be different from the step number in the program displayed by the jump function.)	Error location information	At instruction execution
33E0H	Program structure error	• The relationship between LD/LDI/LDP/LDF/LDPI/LDFI and ANB/ORB instructions is invalid.	• Rewrite the program file.	Error location information	At power-on, at RESET
33E1H	Program structure error	• The relationship among MPS, MRD, and MPP is invalid.	• Rewrite the program file.	Error location information	At power-on, at RESET
33E2H	Program structure error	• An instruction that should start from the bus line is not connected to the bus line.	• Rewrite the program file.	Error location information	At power-on, at RESET

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
33E3H	Program structure error	• The relationship between FOR and NEXT instructions is invalid.	• Modify the program so that the mutual relationship between instructions becomes correct.	Error location information	At power-on, at RESET
33E4H	Program structure error	• The relationship between MC and MCR instructions is invalid.	• Modify the program so that the mutual relationship between instructions becomes correct.	Error location information	At power-on, at RESET
33E5H	Program structure error	• The relationship between STL and RETSTL instructions is invalid.	• Modify the program so that the mutual relationship between instructions becomes correct.	Error location information	At power-on, at RESET
33E6H	Program structure error	• An instruction or interrupt pointer that cannot be used in the main routine program is used.	• Modify the program so that instruction or pointer use becomes correct.	Error location information	At power-on, at RESET
33E7H	Program structure error	• The relationship among a global pointer, interrupt pointer, and return instruction is invalid.	• Modify the program so that the mutual relationship between pointer and return instruction becomes correct.	Error location information	At power-on, at RESET
33F1H	Program structure error	• The program structure of the ST language, FB, and functions is invalid.	• Check the syntax of the ST language, FB, and functions.	Error location information	At END instruction execution, at interrupt occurrence
33F2H	Program structure error	• The program structure of the ST language, FB, and functions is invalid.	• Check the syntax of the ST language, FB, and functions.	Error location information	At instruction execution
33E3H	Program structure error	• The relationship between FOR and NEXT instructions is invalid.	• Modify the program so that the mutual relationship between instructions becomes correct.	Error location information	At power-on, at RESET
3400H	Operation error	• A value of 0 was input as a divisor in an applied instruction.	• Review the data specified as the divisor in the applied instruction.	Error location information	At instruction execution
3401H	Operation error	• Data that cannot be converted was input in an applied instruction.	• Review the data specified in the applied instruction.	Error location information	At instruction execution
3402H	Operation error	• A value of -0, a denormalized number, a non-number, or ±∞ was input in an applied instruction.	• Review the data specified in the applied instruction.	Error location information	At instruction execution
3403H	Operation error	• An overflow occurred in an applied instruction.	• Review the data specified in the applied instruction.	Error location information	At instruction execution
3405H	Operation error	• Data that is outside the allowable range was input in an applied instruction.	• Review the data specified in the applied instruction.	Error location information	At instruction execution
3406H	Operation error	• The output result is outside the allowable device range in an applied instruction.	• Review the data specified in the applied instruction.	Error location information	At instruction execution
3420H	Operation error	• A module access device is specified to both (s) and (d) in a BMOV instruction.	• Review the device specified in the BMOV instruction.	Error location information	At instruction execution
3500H	Operation error	• A value outside the allowable range was set to the sampling time (TS).	• Check the contents of the parameters.	Error location information	At instruction execution
3502H	Operation error	• A value outside the allowable range was set to the input filter constant (α).	• Check the contents of the parameters.	Error location information	At instruction execution
3503H	Operation error	• A value outside the allowable range was set to the proportional gain (KP).	• Check the contents of the parameters.	Error location information	At instruction execution
3504H	Operation error	• A value outside the allowable range was set to the integral time (TI).	• Check the contents of the parameters.	Error location information	At instruction execution
3505H	Operation error	• A value outside the allowable range was set to the derivative gain (KD).	• Check the contents of the parameters.	Error location information	At instruction execution

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
3506H	Operation error	• A value outside the allowable range was set to the derivative time (TD).	• Check the contents of the parameters.	Error location information	At instruction execution
350AH	Operation error	• The sampling time is lower than the scan time.	• The operation is continued in the condition "sampling time (TS) = cyclic time (scan time)".	Error location information	At instruction execution
350CH	Operation error	• The variation of measured value is greater than the maximum value or lower than the minimum value.	• The operation is continued with the maximum or minimum value.	Error location information	At instruction execution
350DH	Operation error	• The deviation is greater than the maximum value or lower than the minimum value.	• The operation is continued with the maximum or minimum value.	Error location information	At instruction execution
350EH	Operation error	• The integral result is greater than the maximum value or lower than the minimum value.	• The operation is continued with the maximum or minimum value.	Error location information	At instruction execution
350FH	Operation error	• The derivative value is greater than the maximum value or lower than the minimum value due to the derivative gain (KP).	• The operation is continued with the maximum or minimum value.	Error location information	At instruction execution
3510H	Operation error	• The derivative result is greater than the maximum value or lower than the minimum value.	• The operation is continued with the maximum or minimum value.	Error location information	At instruction execution
3511H	Operation error	• The PID operation result is greater than the maximum value or lower than the minimum value.	• The operation is continued with the maximum or minimum value.	Error location information	At instruction execution
3512H	Operation error	• The output upper limit value is lower than the output lower limit value.	• Calculation is continued with the output upper limit value and output lower limit value transposed.	Error location information	At instruction execution
3513H	Operation error	• The input variation alarm set value or output variation alarm set value is outside the allowable range.	• The operation is continued without alarm output.	Error location information	At instruction execution
3514H	Operation error	• The auto tuning result in the step response method is abnormal. • The deviation at start of auto tuning is 75 or less. • The deviation at end of auto tuning is 1/3 or less of the deviation at start of auto tuning.	• Check the measured value and target value, and then execute auto tuning again.	Error location information	At instruction execution
3515H	Operation error	• The operation direction estimated from the measured value at the start of auto tuning in the step response method was different from the actual operation direction of the output during auto tuning.	• Correct the relationship among the target value, output value for auto tuning, and the measured value, and then execute auto tuning again.	Error location information	At instruction execution
3516H	Operation error	• Because the set value fluctuated during auto tuning in the step response method, auto tuning was not executed correctly.	• Set the sampling time to a value larger than the output change cycle, or set a larger value for the input filter constant. After changing the setting, execute auto tuning again.	Error location information	At instruction execution
3517H	Operation error	• The output set value upper limit for auto tuning is lower than the lower limit.	• Verify that the target setting contents are correct.	Error location information	At instruction execution
3518H	Operation error	• A value outside the allowable range was set to the PV threshold for auto tuning.	• Verify that the target setting contents are correct.	Error location information	At instruction execution
3519H	Operation error	• Operation is not performed normally because devices occupied by the PID instruction were overwritten.	• Ensure that devices occupied by PID instruction are not overwritten in the program.	Error location information	At instruction execution
351AH	Operation error	• The auto tuning time is longer than necessary.	• Increase the difference (ULV - LLV) between the upper limit and lower limit of the output value for auto tuning, set a smaller value to the input filter constant (α), or set a smaller value to the PV threshold (SHPV) for auto tuning, and then check the result for improvement.	Error location information	At instruction execution

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
351BH	Operation error	• The variation of the measured value is too small compared with the output value.	• Multiply the measured value (PV) by "10" so that the variation of the measured value will increase during auto tuning. The operation is continued with KP = 32767.	Error location information	At instruction execution
351CH	Operation error	• The auto tuning time is longer than necessary.	• Increase the difference (ULV - LLV) between the upper limit and lower limit of the output value for auto tuning, set a smaller value to the input filter constant (α), or set a smaller value to the PV threshold (SHPV) for auto tuning, and then check the result for improvement. The operation is continued with KP = 32767.	Error location information	At instruction execution
351DH	Operation error	• The auto tuning time is longer than necessary.	• Increase the difference (ULV - LLV) between the upper limit and lower limit of the output value for auto tuning, set a smaller value to the input filter constant (α), or set a smaller value to the PV threshold (SHPV) for auto tuning, and then check the result for improvement. The operation is continued with KP = 32767.	Error location information	At instruction execution
3580H	Operation error	• An instruction that cannot be used in an interrupt routine program is used.	• Modify the program so that no instruction whose use is disabled by the interrupt routine program is used.	Error location information	At instruction execution
3581H	Operation error	• Modules subsequent to the bus conversion module are using an operand that cannot be used.	• Modify the program so that no operand whose use is disabled for modules subsequent to the bus conversion module is used.	Error location information	At instruction execution
3582H	Operation error	• An instruction that cannot be used in an interrupt routine program is used.	• Modify the program so that no instruction whose use is disabled by the interrupt routine program is used.	Error location information	At instruction execution
3600H	Operation error	• The channel specified by instructions using communication functions or built-in I/O does not have the appropriate parameter.	• Verify that the parameter setting of the channel specified by instructions using communication functions or built-in I/O is correct.	Error location information	At instruction execution
3611H	CH1 pulse width, period setting error	• The value of the special register to set the pulse width and cycle of the PWM/ DPWM instruction is abnormal.	• Modify the value of the special register and restart PWM.	Error location information and system configuration information	At END instruction execution
3612H	CH2 pulse width, period setting error	• The value of the special register to set the pulse width and cycle of the PWM/ DPWM instruction is abnormal.	• Modify the value of the special register and restart PWM.	Error location information and system configuration information	At END instruction execution
3613H	CH3 pulse width, period setting error	• The value of the special register to set the pulse width and cycle of the PWM/ DPWM instruction is abnormal.	• Modify the value of the special register and restart PWM.	Error location information and system configuration information	At END instruction execution
3614H	CH4 pulse width, period setting error	• The value of the special register to set the pulse width and cycle of the PWM/ DPWM instruction is abnormal.	• Modify the value of the special register and restart PWM.	Error location information and system configuration information	At END instruction execution
3621H	Axis 1 limit detection error	• Both the forward and reverse limits were detected at the time of zero return or the limit of the moving direction was detected after the near-point dog was detected.	• Recheck the relationship between the near-point dog and limits.	Error location information and system configuration information	At END instruction execution, at instruction execution
3622H	Axis 2 limit detection error	• Both the forward and reverse limits were detected at the time of zero return or the limit of the moving direction was detected after the near-point dog was detected.	• Recheck the relationship between the near-point dog and limits.	Error location information and system configuration information	At END instruction execution, at instruction execution

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
3623H	Axis 3 limit detection error	• Both the forward and reverse limits were detected at the time of zero return or the limit of the moving direction was detected after the near-point dog was detected.	• Recheck the relationship between the near-point dog and limits.	Error location information and system configuration information	At END instruction execution, at instruction execution
3624H	Axis 4 limit detection error	• Both the forward and reverse limits were detected at the time of zero return or the limit of the moving direction was detected after the near-point dog was detected.	• Recheck the relationship between the near-point dog and limits.	Error location information and system configuration information	At END instruction execution, at instruction execution
3631H	Axis 1 positioning address error	• The 32-bit range was exceeded when the unit of the positioning address was converted. • The total transfer distance before and after the interrupt of the DVIT/DDVIT instruction or 1-speed positioning with interruption exceeded 7FFFFFFFH. Or, when the operation was started, the positioning address was set to 0. • Pulses of 7FFFFFFFH or greater are needed to specify an absolute address.	• Start the positioning within specifications.	Error location information and system configuration information	At END instruction execution, at interrupt occurrence
3632H	Axis 2 positioning address error	• The 32-bit range was exceeded when the unit of the positioning address was converted. • The total transfer distance before and after the interrupt of the DVIT/DDVIT instruction or 1-speed positioning with interruption exceeded 7FFFFFFFH. Or, when the operation was started, the positioning address was set to 0. • Pulses of 7FFFFFFFH or greater are needed to specify an absolute address.	• Start the positioning within specifications.	Error location information and system configuration information	At END instruction execution, at interrupt occurrence
3633H	Axis 3 positioning address error	• The 32-bit range was exceeded when the unit of the positioning address was converted. • The total transfer distance before and after the interrupt of the DVIT/DDVIT instruction or 1-speed positioning with interruption exceeded 7FFFFFFFH. Or, when the operation was started, the positioning address was set to 0. • Pulses of 7FFFFFFFH or greater are needed to specify an absolute address.	• Start the positioning within specifications.	Error location information and system configuration information	At END instruction execution, at interrupt occurrence
3634H	Axis 4 positioning address error	• The 32-bit range was exceeded when the unit of the positioning address was converted. • The total transfer distance before and after the interrupt of the DVIT/DDVIT instruction or 1-speed positioning with interruption exceeded 7FFFFFFFH. Or, when the operation was started, the positioning address was set to 0. • Pulses of 7FFFFFFFH or greater are needed to specify an absolute address.	• Start the positioning within specifications.	Error location information and system configuration information	At END instruction execution, at interrupt occurrence
3641H	Axis 1 command speed error	• The 32-bit range was exceeded when the unit of the command speed was converted.	• Start the positioning within specifications.	Error location information and system configuration information	At instruction execution
3642H	Axis 2 command speed error	• The 32-bit range was exceeded when the unit of the command speed was converted.	• Start the positioning within specifications.	Error location information and system configuration information	At instruction execution

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
3643H	Axis 3 command speed error	• The 32-bit range was exceeded when the unit of the command speed was converted.	• Start the positioning within specifications.	Error location information and system configuration information	At instruction execution
3644H	Axis 4 command speed error	• The 32-bit range was exceeded when the unit of the command speed was converted.	• Start the positioning within specifications.	Error location information and system configuration information	At instruction execution
3651H	Axis 1 error stop (deceleration stop)	• When pulses were being output or positioning was starting, the PLC decelerated and stopped the pulse output due to the limit of the moving direction or writing during RUN. (The PLSY/DPLSY instruction stops pulse output immediately at both limits.)	• Eliminate the error that has caused the stop and restart the positioning.	Error location information and system configuration information	At END instruction execution, at instruction execution
3652H	Axis 2 error stop (deceleration stop)	• When pulses were being output or positioning was starting, the PLC decelerated and stopped the pulse output due to the limit of the moving direction or writing during RUN. (The PLSY/DPLSY instruction stops pulse output immediately at both limits.)	• Eliminate the error that has caused the stop and restart the positioning.	Error location information and system configuration information	At END instruction execution, at instruction execution
3653H	Axis 3 error stop (deceleration stop)	• When pulses were being output or positioning was starting, the PLC decelerated and stopped the pulse output due to the limit of the moving direction or writing during RUN. (The PLSY/DPLSY instruction stops pulse output immediately at both limits.)	• Eliminate the error that has caused the stop and restart the positioning.	Error location information and system configuration information	At END instruction execution, at instruction execution
3654H	Axis 4 error stop (deceleration stop)	• When pulses were being output or positioning was starting, the PLC decelerated and stopped the pulse output due to the limit of the moving direction or writing during RUN. (The PLSY/DPLSY instruction stops pulse output immediately at both limits.)	• Eliminate the error that has caused the stop and restart the positioning.	Error location information and system configuration information	At END instruction execution, at instruction execution
3661H	Axis 1 error stop (immediately stop)	• When pulses were being output or positioning was rising, the PLC stopped the pulse output immediately by the pulse stop command or detection of the all outputs disable flag.	• Eliminate the error that has caused the stop and restart the positioning.	Error location information and system configuration information	At END instruction execution, at instruction execution
3662H	Axis 2 error stop (immediately stop)	• When pulses were being output or positioning was rising, the PLC stopped the pulse output immediately by the pulse stop command or detection of the all outputs disable flag.	• Eliminate the error that has caused the stop and restart the positioning.	Error location information and system configuration information	At END instruction execution, at instruction execution
3663H	Axis 3 error stop (immediately stop)	• When pulses were being output or positioning was rising, the PLC stopped the pulse output immediately by the pulse stop command or detection of the all outputs disable flag.	• Eliminate the error that has caused the stop and restart the positioning.	Error location information and system configuration information	At END instruction execution, at instruction execution
3664H	Axis 4 error stop (immediately stop)	• When pulses were being output or positioning was rising, the PLC stopped the pulse output immediately by the pulse stop command or detection of the all outputs disable flag.	• Eliminate the error that has caused the stop and restart the positioning.	Error location information and system configuration information	At END instruction execution, at instruction execution
3671H	Axis 1 positioning table operand error	• The value of an operand in the table is abnormal. (Other than the positioning address and command speed)	• Set the correct value to the table.	Error location information and system configuration information	At END instruction execution, at interrupt occurrence

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
3672H	Axis 2 positioning table operand error	• The value of an operand in the table is abnormal. (Other than the positioning address and command speed)	• Set the correct value to the table.	Error location information and system configuration information	At END instruction execution, at interrupt occurrence
3673H	Axis 3 positioning table operand error	• The value of an operand in the table is abnormal. (Other than the positioning address and command speed)	• Set the correct value to the table.	Error location information and system configuration information	At END instruction execution, at interrupt occurrence
3674H	Axis 4 positioning table operand error	• The value of an operand in the table is abnormal. (Other than the positioning address and command speed)	• Set the correct value to the table.	Error location information and system configuration information	At END instruction execution, at interrupt occurrence
3681H	Axis 1 positioning table shift error (table specification)	• Tables which cannot be used together were specified for continuous operation. • The counterpart axis for the interpolation operation table was specified.	• Observe the restrictions on table operation.	Error location information and system configuration information	At END instruction execution, at interrupt occurrence
3682H	Axis 2 positioning table shift error (table specification)	• Tables which cannot be used together were specified for continuous operation. • The counterpart axis for the interpolation operation table was specified.	• Observe the restrictions on table operation.	Error location information and system configuration information	At END instruction execution, at interrupt occurrence
3683H	Axis 3 positioning table shift error (table specification)	• Tables which cannot be used together were specified for continuous operation. • The counterpart axis for the interpolation operation table was specified.	• Observe the restrictions on table operation.	Error location information and system configuration information	At END instruction execution, at interrupt occurrence
3684H	Axis 4 positioning table shift error (table specification)	• Tables which cannot be used together were specified for continuous operation. • The counterpart axis for the interpolation operation table was specified.	• Observe the restrictions on table operation.	Error location information and system configuration information	At END instruction execution, at interrupt occurrence
3691H	Axis 1 positioning table shift error (table shift)	• Table shift cannot be completed in time because one or more tables shifted per 10 ms.	• Set the interval of table shifts to 10 ms or greater.	Error location information and system configuration information	At interrupt occurrence
3692H	Axis 2 positioning table shift error (table shift)	• Table shift cannot be completed in time because one or more tables shifted per 10 ms.	• Set the interval of table shifts to 10 ms or greater.	Error location information and system configuration information	At interrupt occurrence
3693H	Axis 3 positioning table shift error (table shift)	• Table shift cannot be completed in time because one or more tables shifted per 10 ms.	• Set the interval of table shifts to 10 ms or greater.	Error location information and system configuration information	At interrupt occurrence
3694H	Axis 4 positioning table shift error (table shift)	• Table shift cannot be completed in time because one or more tables shifted per 10 ms.	• Set the interval of table shifts to 10 ms or greater.	Error location information and system configuration information	At interrupt occurrence
36A1H	Axis 1 interpolation operation error (no counterpart axis)	• The counterpart axis table for the interpolation operation cannot be found.	• Set the table of the counterpart axis correctly.	Error location information and system configuration information	At instruction execution
36A2H	Axis 2 interpolation operation error (no counterpart axis)	• The counterpart axis table for the interpolation operation cannot be found.	• Set the table of the counterpart axis correctly.	Error location information and system configuration information	At instruction execution

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
36A3H	Axis 3 interpolation operation error (no counterpart axis)	• The counterpart axis table for the interpolation operation cannot be found.	• Set the table of the counterpart axis correctly.	Error location information and system configuration information	At instruction execution
36A4H	Axis 4 interpolation operation error (no counterpart axis)	• The counterpart axis table for the interpolation operation cannot be found.	• Set the table of the counterpart axis correctly.	Error location information and system configuration information	At instruction execution
36B1H	Axis 1 interpolation operation error (reference/ counterpart axis error)	• Conditions such as limits were satisfied to stop pulses in the reference axis or counterpart axis. • Pulses are being output.	• Verify that the reference axis and counterpart axis are not in use and the stop conditions are not satisfied.	Error location information and system configuration information	At instruction execution
36B2H	Axis 2 interpolation operation error (reference/ counterpart axis error)	• Conditions such as limits were satisfied to stop pulses in the reference axis or counterpart axis. • Pulses are being output.	• Verify that the reference axis and counterpart axis are not in use and the stop conditions are not satisfied.	Error location information and system configuration information	At instruction execution
36B3H	Axis 3 interpolation operation error (reference/ counterpart axis error)	• Conditions such as limits were satisfied to stop pulses in the reference axis or counterpart axis. • Pulses are being output.	• Verify that the reference axis and counterpart axis are not in use and the stop conditions are not satisfied.	Error location information and system configuration information	At instruction execution
36B4H	Axis 4 interpolation operation error (reference/ counterpart axis error)	• Conditions such as limits were satisfied to stop pulses in the reference axis or counterpart axis. • Pulses are being output.	• Verify that the reference axis and counterpart axis are not in use and the stop conditions are not satisfied.	Error location information and system configuration information	At instruction execution
36F0H	ABS sum error	• There is a sum check error in ABS data read from servo.	• Check servo wiring and setting.	Error location information and system configuration information	At instruction execution
3780H	High-speed comparison table maximum excess error	• The number of high-speed comparison tables registered is greater than the upper limit.	• Check the total number of tables in the parameters and tables registered in the comparison match instruction.	Error location information	At END instruction execution, at instruction execution
3781H	Preset value range outside error	• The preset value is greater than the ring length set value.	• Disable the ring length. • Set the preset value within the ring length range.	Error location information and system configuration information	At instruction execution
3C00H	Hardware failure	• A hardware failure was detected.	• Reset the CPU module and perform RUN. If the same error appears, the hardware of the CPU module may be malfunctioning. Consult your local Mitsubishi Electric representative.	System configuration information	At power-on, at RESET
3C01H	Hardware failure	• A hardware failure was detected.	• Reset the CPU module and perform RUN. If the same error appears, the hardware of the CPU module may be malfunctioning. Consult your local Mitsubishi Electric representative.	System configuration information	At power-on, at RESET
3C02H	Hardware failure	• A hardware failure was detected.	• Reset the CPU module and perform RUN. If the same error appears, the hardware of the CPU module may be malfunctioning. Consult your local Mitsubishi Electric representative.	System configuration information	At power-on, at RESET
3C03H	Hardware failure	• A hardware failure was detected.	• Reset the CPU module and perform RUN. If the same error appears, the hardware of the CPU module may be malfunctioning. Consult your local Mitsubishi Electric representative.	—	At power-on, at RESET

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
3C0FH	Hardware failure	• A hardware failure was detected.	• Reset the CPU module and perform RUN. If the same error appears, the hardware of the CPU module may be malfunctioning. Consult your local Mitsubishi Electric representative.	—	At power-on, at RESET
3C20H	Memory error	• A memory error was detected.	• Reset the CPU module and perform RUN. If the same error appears, the hardware of the CPU module may be malfunctioning. Consult your local Mitsubishi Electric representative.	—	At power-on, at RESET
3C22H	Memory error	• A memory error was detected.	• Reset the CPU module and perform RUN. If the same error appears, the hardware of the CPU module may be malfunctioning. Consult your local Mitsubishi Electric representative.	—	At power-on, at RESET
3C2FH	Memory error	• A memory error was detected.	• Reset the CPU module and perform RUN. If the same error appears, the hardware of the CPU module may be malfunctioning. Consult your local Mitsubishi Electric representative.	—	At power-on, at RESET
3E20H	Program execution error	• A program larger than the internal memory capacity was written.	• Reset the CPU module and perform RUN. If the same error appears, the hardware of the CPU module may be malfunctioning. Consult your local Mitsubishi Electric representative.	—	At memory card attachment or detachment

Error codes of the CPU module (4000H to 4FFFH)

The following table lists the error codes detected by other causes than the self-diagnostics function of the CPU module.

Error code	Error name	Error details and cause	Action
4000H	Common error	• Serial communication sum check error.	• Connect the serial communication cable correctly. • Take measures to reduce noise.
4001H	Common error	• An unsupported request was executed.	• Check the command data of SLMP/MC protocol. • Check the CPU module model name selected in the engineering tool. • Check the target CPU module model name.
4002H	Common error	• An unsupported request was executed.	• Check the command data of SLMP/MC protocol. • Check the CPU module model name selected in the engineering tool. • Execute the request again. • If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.
4005H	Common error	• The volume of data handled according to the specified request is too large.	• Check the command data of SLMP/MC protocol.
4006H	Common error	• Initial communication has failed.	• When using serial communication, check with the external device manufacturer for support conditions. • When using serial communication, check the CPU module model name selected in the engineering tool. • When using Ethernet communication, shift the communication start timing.
4010H	CPU module operation error	• Since the CPU module is running, the request contents cannot be executed.	• Execute after setting the CPU module to STOP status.
4013H	CPU module operation error	• Since the CPU module is not in a STOP status, the request contents cannot be executed.	• Execute after setting the CPU module to STOP status.
4021H	File related error	• The specified drive (memory) does not exist or there is an error.	• Check the specified drive (memory) status. • Back up data in the CPU module, and then initialize the memory.
4022H	File related error	• The file with the specified file name or file No. does not exist.	• Check the specified file name and file No.
4025H	File related error	• The specified file is processing the request from another engineering tool.	• Forcibly execute the request. Or, execute the request again after the processing being performed ends.
4027H	File related error	• The specified range is larger than the file size.	• Check the specified range and access within that range.
4029H	File related error	• The specified file capacity cannot be obtained.	• Review the specified file capacity, and execute the request again.

Error code	Error name	Error details and cause	Action
402CH	File related error	• The requested operation cannot be executed currently.	• Execute again after a while.
4030H	Device specification error	• The specified device name cannot be handled.	• Check the specified device name.
4031H	Device specification error	• The specified device No. is outside the range. • The CPU module cannot handle the specified device.	• Check the specified device No. • Check the device assignment of the CPU module. • Check the specified device name.
4040H	Intelligent function module specification error	• The request contents cannot be executed in the specified intelligent function module.	• Check whether the specified module is the intelligent function module having the buffer memory.
4041H	Intelligent function module specification error	• The access range exceeds the buffer memory range of the specified intelligent function module.	• Check the start address and access number of points and access within the range that exists in the intelligent function module.
4042H	Intelligent function module specification error	• The specified intelligent function module cannot be accessed.	• Check that the specified intelligent function module is operating normally. • Check the specified module for a hardware fault.
4043H	Intelligent function module specification error	• The intelligent function module does not exist in the specified position.	• Check the I/O number of the specified intelligent function module.
4053H	Protect error	• An error occurred when writing data to the specified drive (memory).	• Check the specified drive (memory). Or, write data again after changing the corresponding drive (memory).
4060H	Online registration error	• The online debug function and the data logging function are being executed with another engineering tool.	• Finish the operation of the other engineering tool and then execute the function again. • If the operation of the other engineering tool is on hold, resume and finish the operation of the other engineering tool, and then execute the function again.
4080H	Other errors	• Request data error.	• Check the request data that has been specified.
4081H	Other errors	• The search target data cannot be detected.	• Check the data to be searched.
408BH	Other errors	• The remote request cannot be executed.	• Reexecute after the CPU module is in a status where the remote request can be executed. • For remote operation, set the parameter to "Enable remote reset".
4121H	File related error	• The specified drive (memory) or file does not exist.	• Execute again after checking the specified drive (memory) or file.
4122H	File related error	• The specified drive (memory) or file does not exist.	• Execute again after checking the specified drive (memory) or file.
4127H	File related error	• File password 32 mismatch.	• Execute again after checking the file password 32.
4135H	File related error	• The date/time data of the engineering tool (personal computer) is out of range.	• Execute again after checking the clock setting of the engineering tool (personal computer).
4139H	File related error	• The size of the specified file has exceeded that of the existing file.	• Execute again after checking the size of the specified file.
413AH	File related error	• The specified file has exceeded the already existing file size.	• Execute again after checking the size of the specified file.
413BH	File related error	• The same file was simultaneously accessed from different engineering tools.	• Execute again after a while.
413EH	File related error	• Operation is disabled for the specified drive (memory).	• Execute again after changing the target drive (memory).
4181H	CPU module built-in Ethernet port error	• Transmission to the receiving modules is unsuccessful.	• Check the external device operation. • Check the status of the lines, such as cables, hubs and routes, connected to receiving modules. • Some line packets may be engaged. Retry to communicate a little while later. • The receiving module may have no free space in receive area (TCP window size is small). Check whether the receiving module processes receive data, or whether the CPU module does not send unnecessary data. • Check whether the settings of the subnet mask pattern and the default router IP address of the CPU module and the receiving modules are correct, or whether the class of the IP address is correct.

MEMO

Revised History

Date	Revision	Description
2/2017	A	First Edition

Mitsubishi Programmable Controller Training Manual
MELSEC iQ-F Series Basic Course (for GX Works3)

MODEL	
MODEL CODE	
JY997D69701	

MITSUBISHI ELECTRIC CORPORATION

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