MITSUBISHI

PROGRAMMABLE CONTROLLERS

MELSEC-F

Changes for the Better

Discovering Control

An Intermediate Guide to Learning PLCs

● 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]

⚠ DANGER

- 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

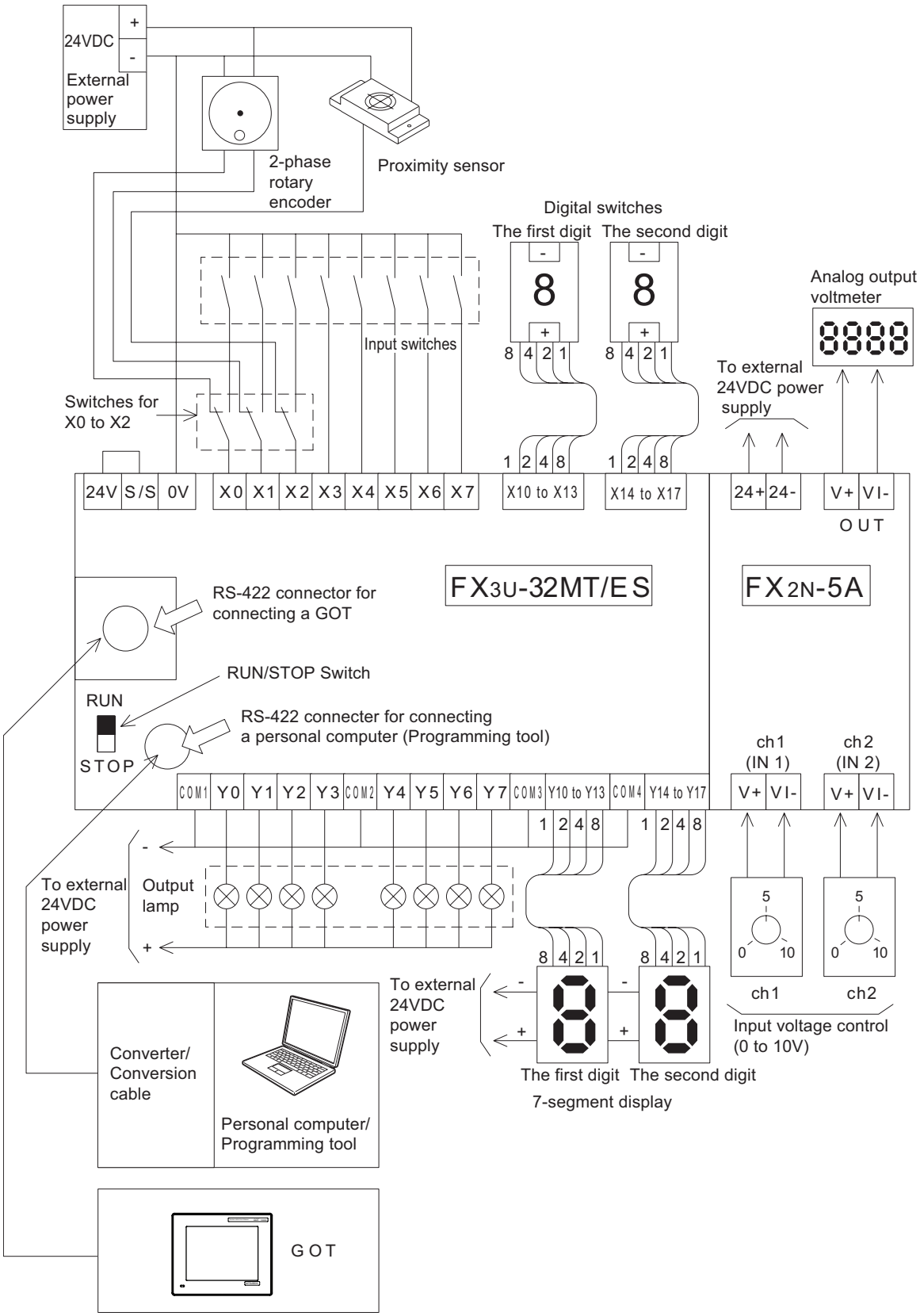
- Proceed with the training under the guidance of a teacher.
- Do not remove the training machine module or change the wiring without permission. Doing so may result in malfunction, misoperation, injury or fire.
- Before attaching or detaching the module, turn the power OFF. Attaching or detaching the 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 (X/Y table, etc.) immediately turn the power switch to OFF.
- If an abnormal event occurs, immediately contact your teacher.

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 External I/O Assignment and Wiring



Chapter 2

GUIDING COURSE: DO YOU REMEMBER?

The Definition of a PLC···

A programmable Logic Controller (PLC) is referred to as a Programmable Controller (PC) or Sequence Controller (SC).
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···

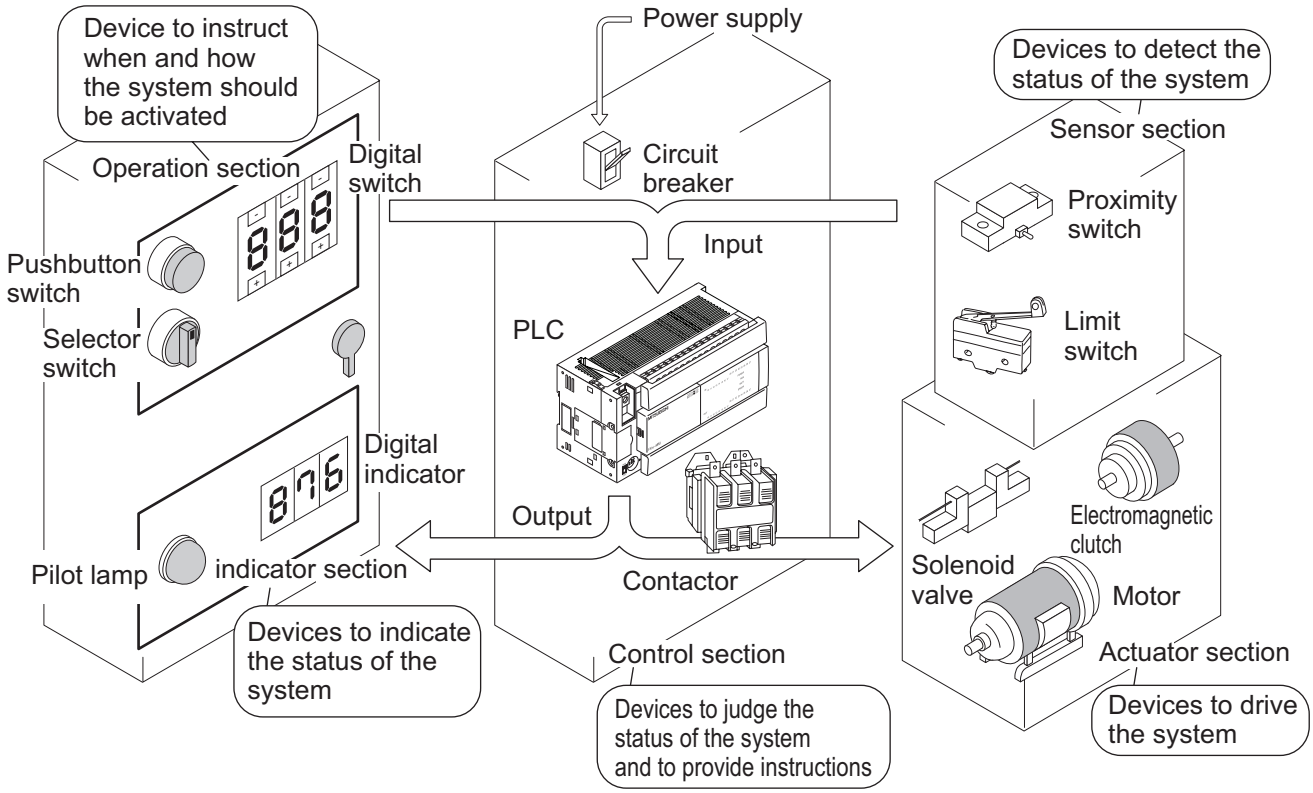
PLCs are broadly used as core components for FA (Factory Automation) and as electronic application products essential for saving labor costs and improving automation.
PLCs can be used for many types of applications such as systematical applications which supply control all over a factory or as standalone applications to control an independent machine.

In this chapter···

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

2.1 PLC - Small, reliable, flexible brain.

2.2.1 An automation solution for the machining, assembly, transfer, inspection, packaging and so on 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.

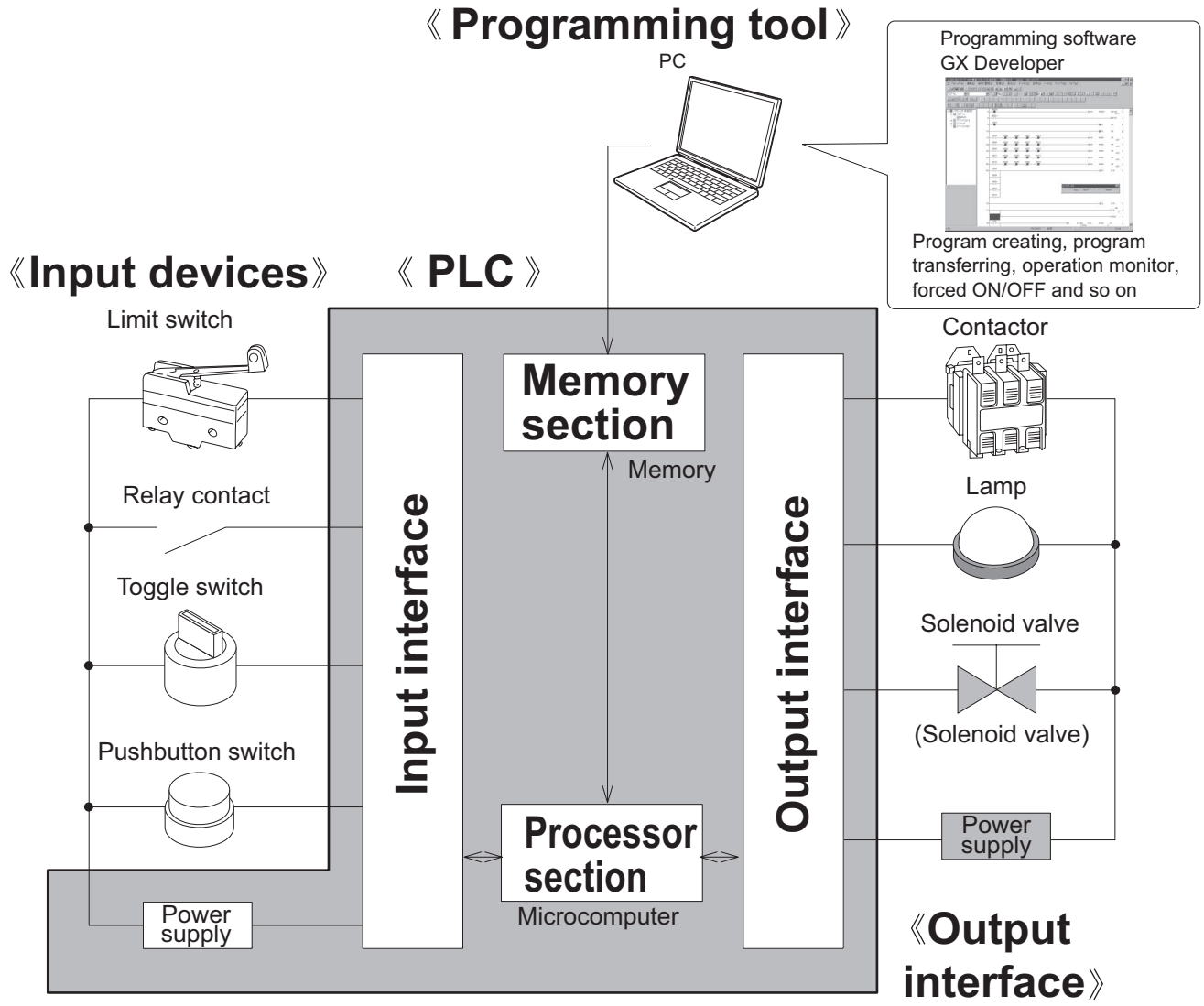
The behaviors of output signals corresponding to the input signals are determined by the contents of programs provided to the PLC.

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

2.2.2 The PLC is a microcomputer for industrial purposes.



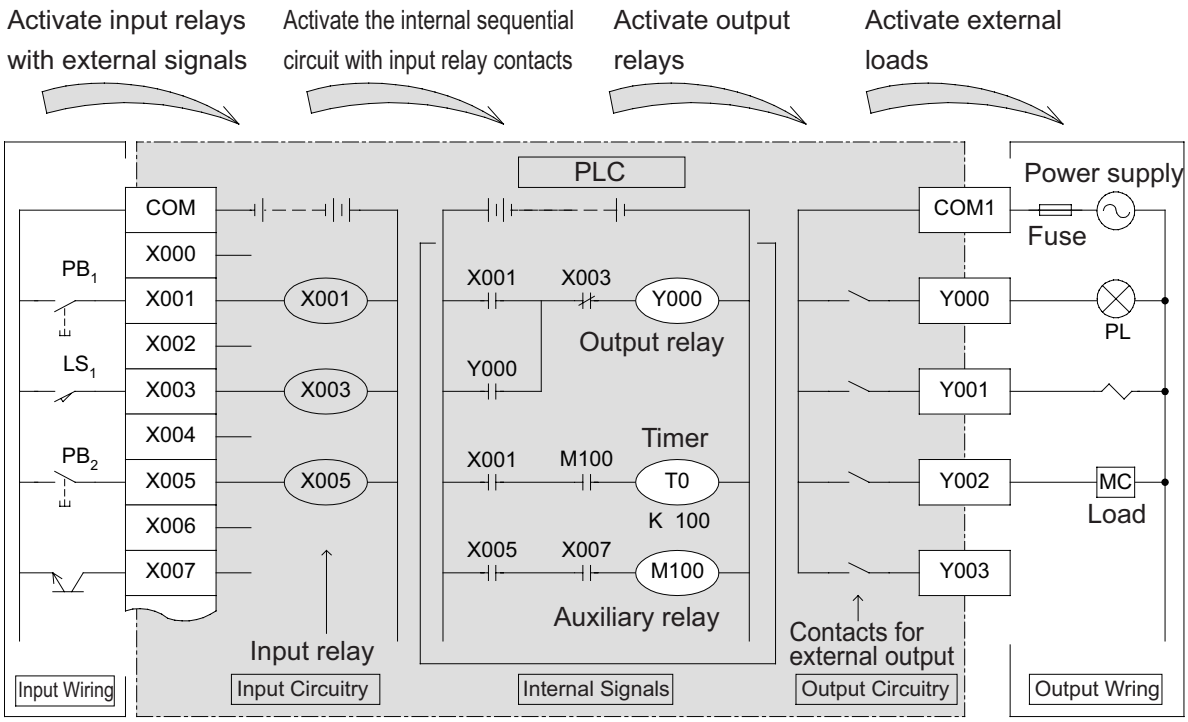
A PLC incorporates an electrical circuit mainly comprised of a microcomputer and memory.
Input/output interfaces exist between input/output devices and the electronic circuit to connect them.
The programming panel is used to write a program to the memory in the PLC.

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."

2.2.3 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.

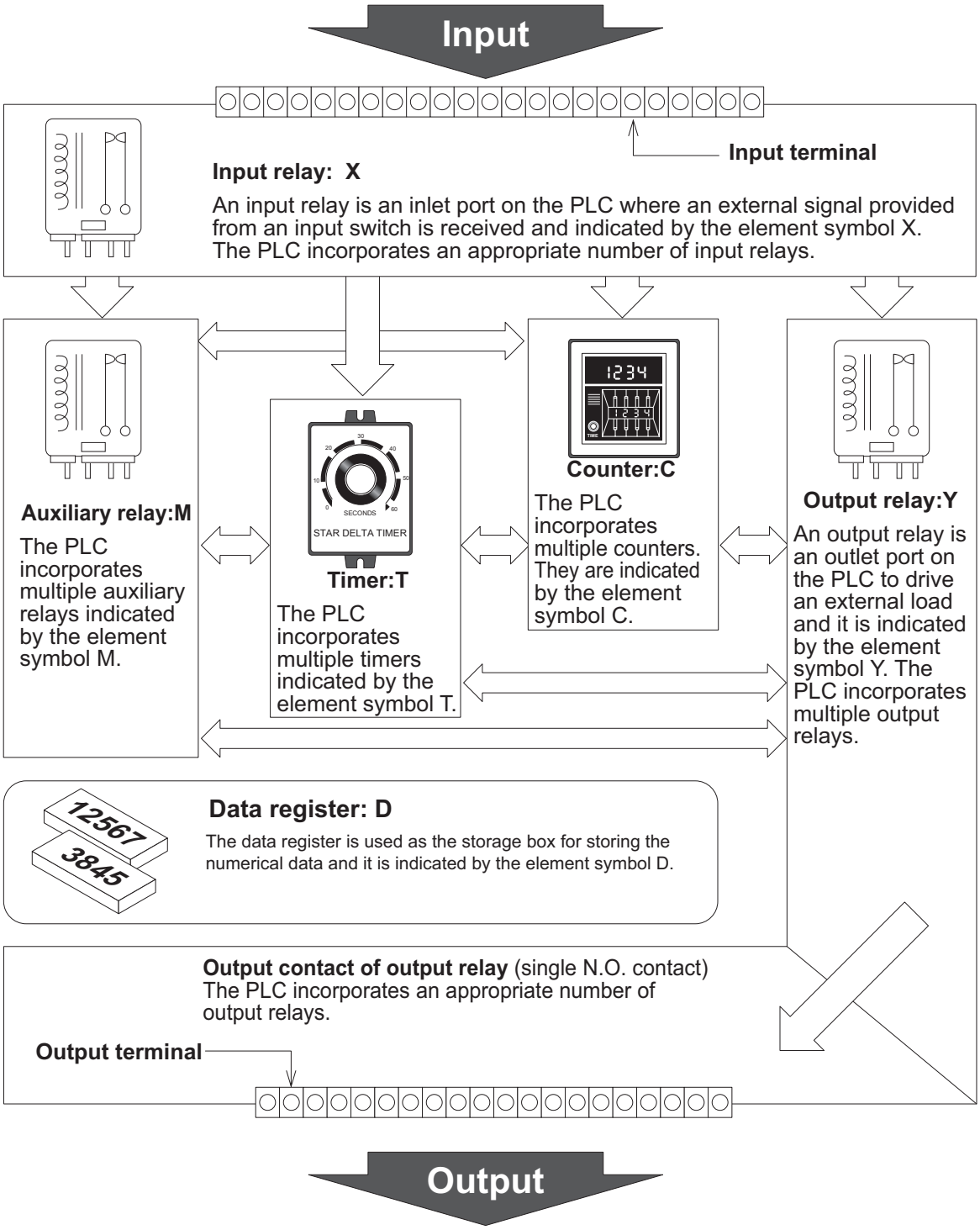
Internal operation of PLC

Signal Flow of PLC

- When the pushbutton switch PB1 is pushed, the coil of the input relay X001 is energized.
- When the coil of the input relay X001 is energized, the N.O. contact of X001 is closed and the coil of the output relay Y000 is energized.
- When the coil of the output relay Y000 is energized, the contact of Y000 is closed, then the pilot lamp PL is illuminated.
- When the pushbutton switch PB1 is released, the coil of the input relay X001 is de-energized and the N.O. contact of X001 is opened.
  But the output relay Y000 is still energized since the N.O. contact is closed. (Self-maintaining action)
- When the input relay is energized by closing the limit switch LS1, the N.C. contact of X003 is opened, then the coil of the output relay Y000 is de-energized (Reset).
  As a result, the pilot lamp; PL distinguishes and the self-maintaining action of the output relay Y000 is cleared.

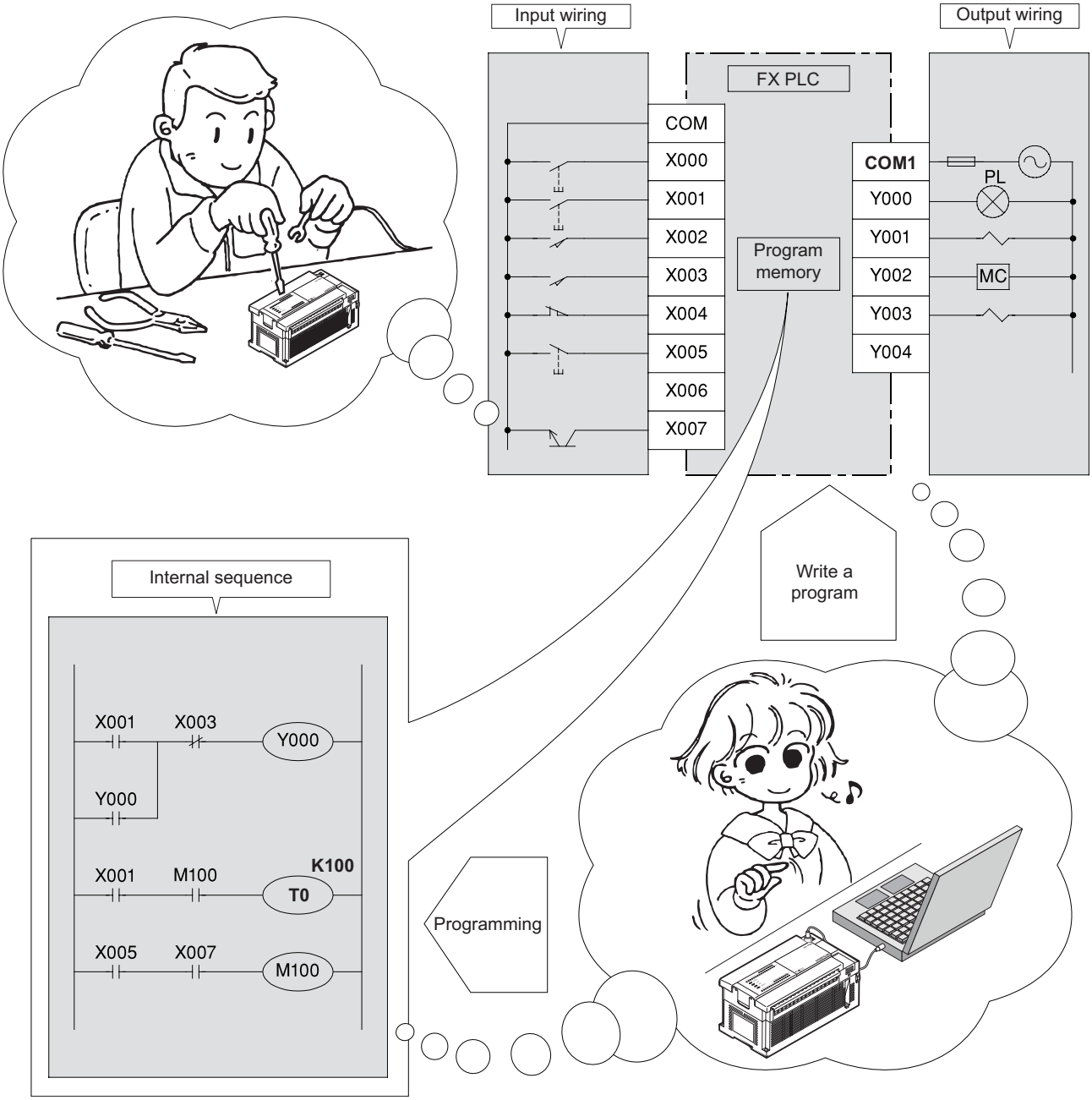
2.2.4 Types of relay and timers

As shown below, a PLC incorporates multiple relays, timers and counters with countless N.O. and N.C. contacts.
A sequential circuit is formed by connecting the contacts and coils.
Also, one advantage of using a PLC is that a lot of storage cases called "data registers" are included.



2.3 Wiring and instructions

Perform the wiring work for input and output devices.



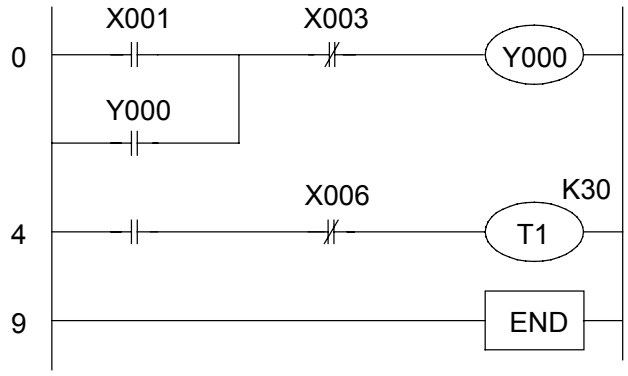
It is easy to use personal computer software to create the internal sequence program, which is equivalent to internal PLC wiring.

2.4 Commands and programs

2.4.1 Mechanism of programs

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

Circuit diagram (ladder diagram)



Instruction list (program list)

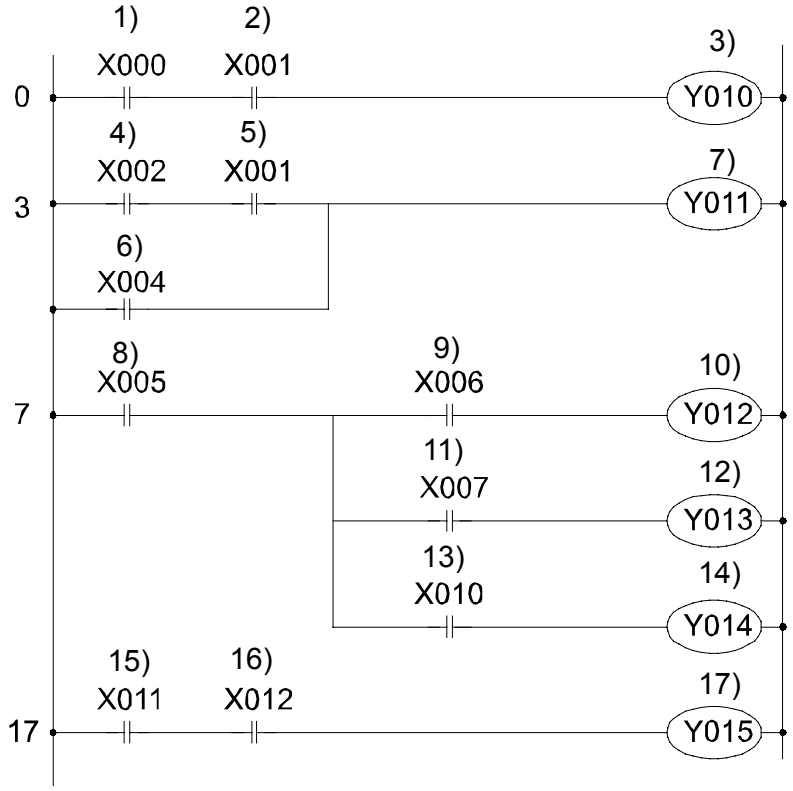
Step number	Instruction	
	Instruction code	Device (number) (operand)
0	LD	X001
1	OR	Y000
2	ANI	X003
3	OUT	Y000
4	LD	Y000
5	ANI	X006
6	OUT	T1 K30
9	END	

Repeat 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 number"**. 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. For example, there is a program memory with the capacity of "2000" steps in the FX1S PLC, "8000" steps in FX1N and FX2N, and "64000" steps in the FX3U.
- 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.
- A PLC program created by the format of circuit diagram (ladder diagram) is also stored in the program memory of the PLC with the format of instruction list (program list).
  The conversion between instruction list (program list) and circuit diagram (ladder diagram) can be done by using the programming software on a personal computer.

2.4.2 The processing order of the program

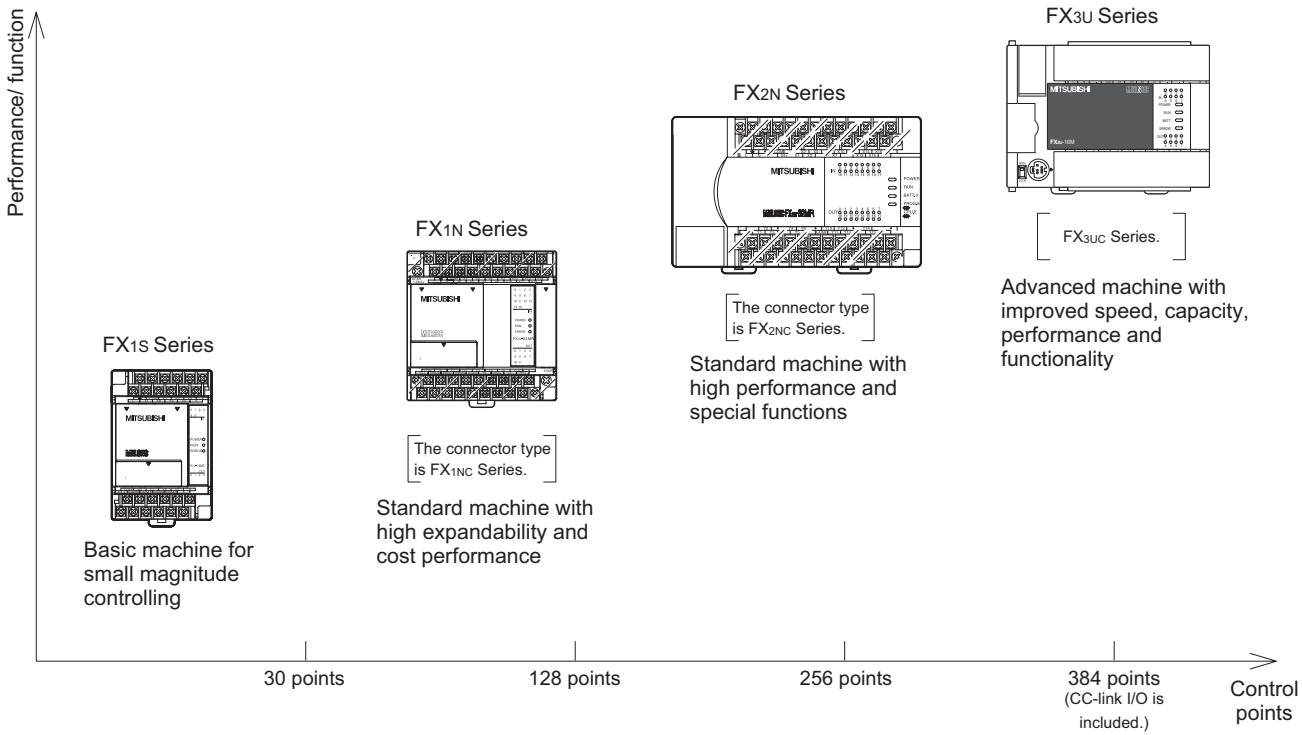
The program processes sequentially from the first step to the end of the program memory in block units with the order from left to right, up to down. (in order of 1), 2), ……17))



2.5 The configuration of an FX PLC

2.5.1 Brief introduction of the main unit

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



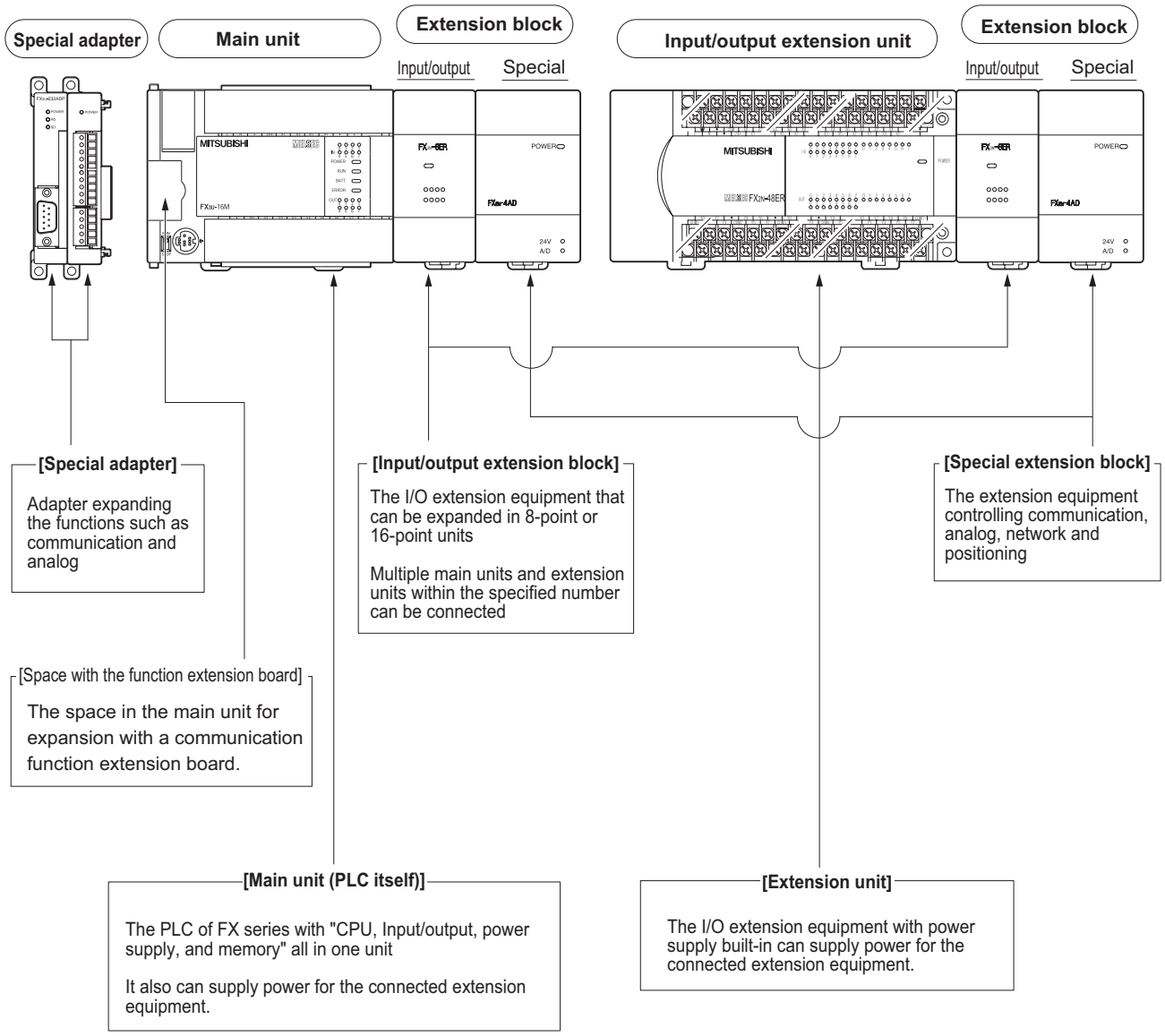
■ Function list (the PLC of the terminal board type is equipped)

○ : Available ×: Unavailable

Function	Terminal board type			
	FX1S	FX1N	FX2N	FX3U
Memory capacity (step)	2000	8000	8000 Max 16000	64000
Input/output extension	×	○	○	○
Special function units/blocks 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	×	×	○	○
Built-in real time clock (clock function)	○	○	○	○
Built-in analog volume	○	○	×	×
Built-in 24V 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	○	○	○	○

2.5.2 The basic configuration of the system

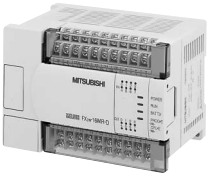
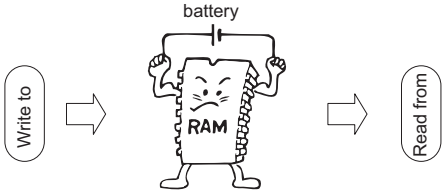
The configuration of an FX PLC will be described by taking an example using the FX3U Series.



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

2.5.3 The types and advantages of the program memory

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

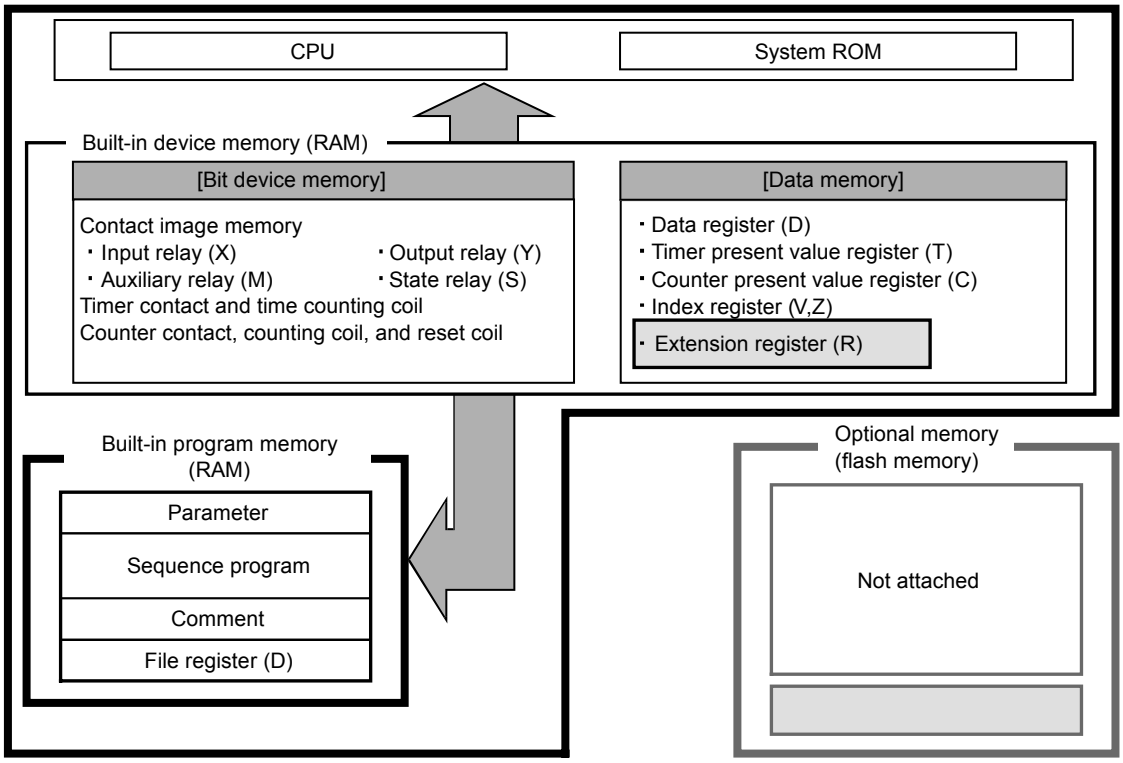
Series	Built-in memory			Advantage
	Type	Memory capacity	Backup method	
 FX1S FX1N/FX1NC	EEPROM memory	2000 steps 8000 steps	Backup unnecessary*1	It is easy to write to/read from the memory, and battery backup is not required. *1: There is a capacitor latched field in the FX1N/FX1NC's latched (battery-backed) devices. *2: The memory can be written up to 20000 times.
 FX2N/FX2NC FX3U/FX3UC	RAM memory	8000 steps 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.

Reference

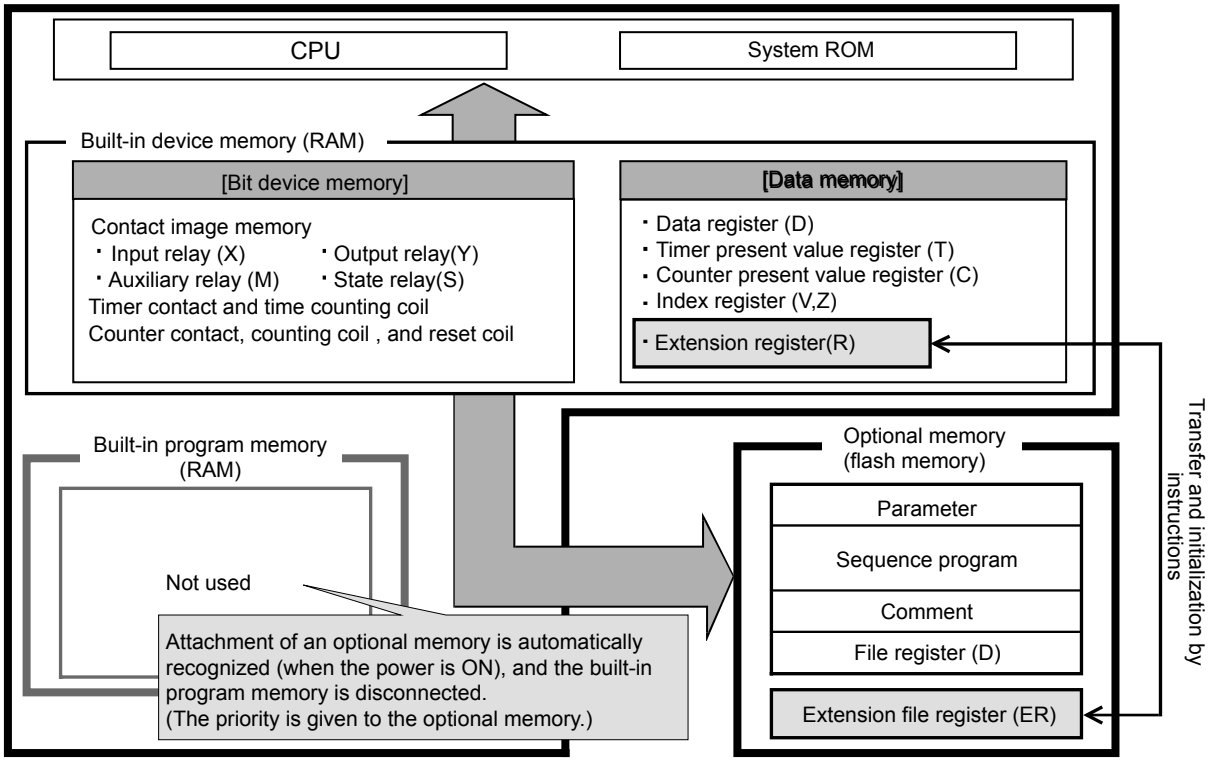
FX PLC's Memory Structure (FX3U,FX3UC Examples)

FX3U/FX3UC PLCs are supplied with RAM memory.
By mounting an optional memory device, the memory type can be changed.

1. When using the built-in memory (without attached optional memory)



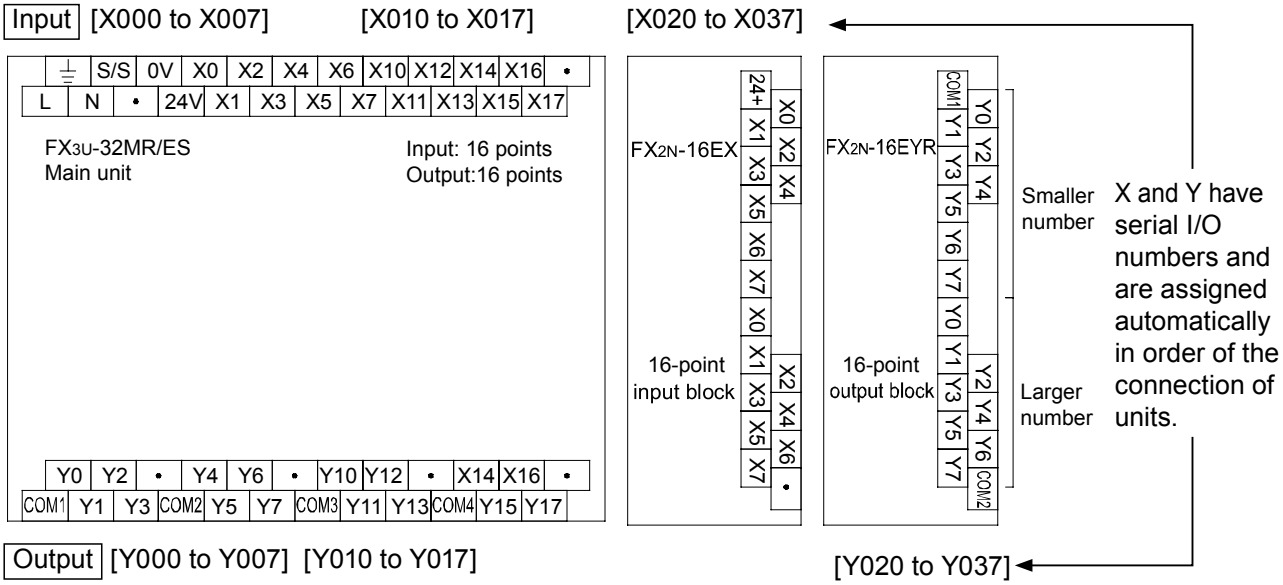
2. When using an attached optional memory (without using the built-in memory)



2.5.4 FX PLC I/O number assignment

Each main unit has I/O numbers assigned by an octal number system such as X000 to X007, X010 to X017, Y000 to Y007, Y010 to Y017 and so forth. (Devices other than I/O relays are numbered by a decimal number system.)
Expansion modules and expansion blocks have I/O numbers subsequent to those of the main unit.

[System configuration example and I/O numbers]



- I/O numbers of expansion modules are assigned with subsequent numbers to those of the main unit with the I/O more adjacent to the main unit having lower numbers.
  It is not necessary to set the parameters by using programming tools such as GX Developer.
- There are some I/O points that have a null value according to the number of I/O on the main unit and expansion modules.
  [Example]
  In the case of the FX_{1N}-24M basic module, 14/10 points (X000 to X015/Y000 to Y011) of the 16/16-point I/O are required, and the relay numbers X016 to X017 and Y012 to Y017 are not used.
  In the case of the FX_{2N}-8ER (mixed with I/O = 4/4), 4/4 points of the 8/8 -point I/O are required, and the remaining 8 relay numbers are not used.
  (The numbers of null value are also counted as occupied points.)

Chapter 3

THE OPERATION OF GX Developer

Using a personal computer, programming becomes easy···

GX Developer 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. Beginning with the most necessary operations, let's learn the programming operations from the start.

Smoothly begin new projects and update them with ease···

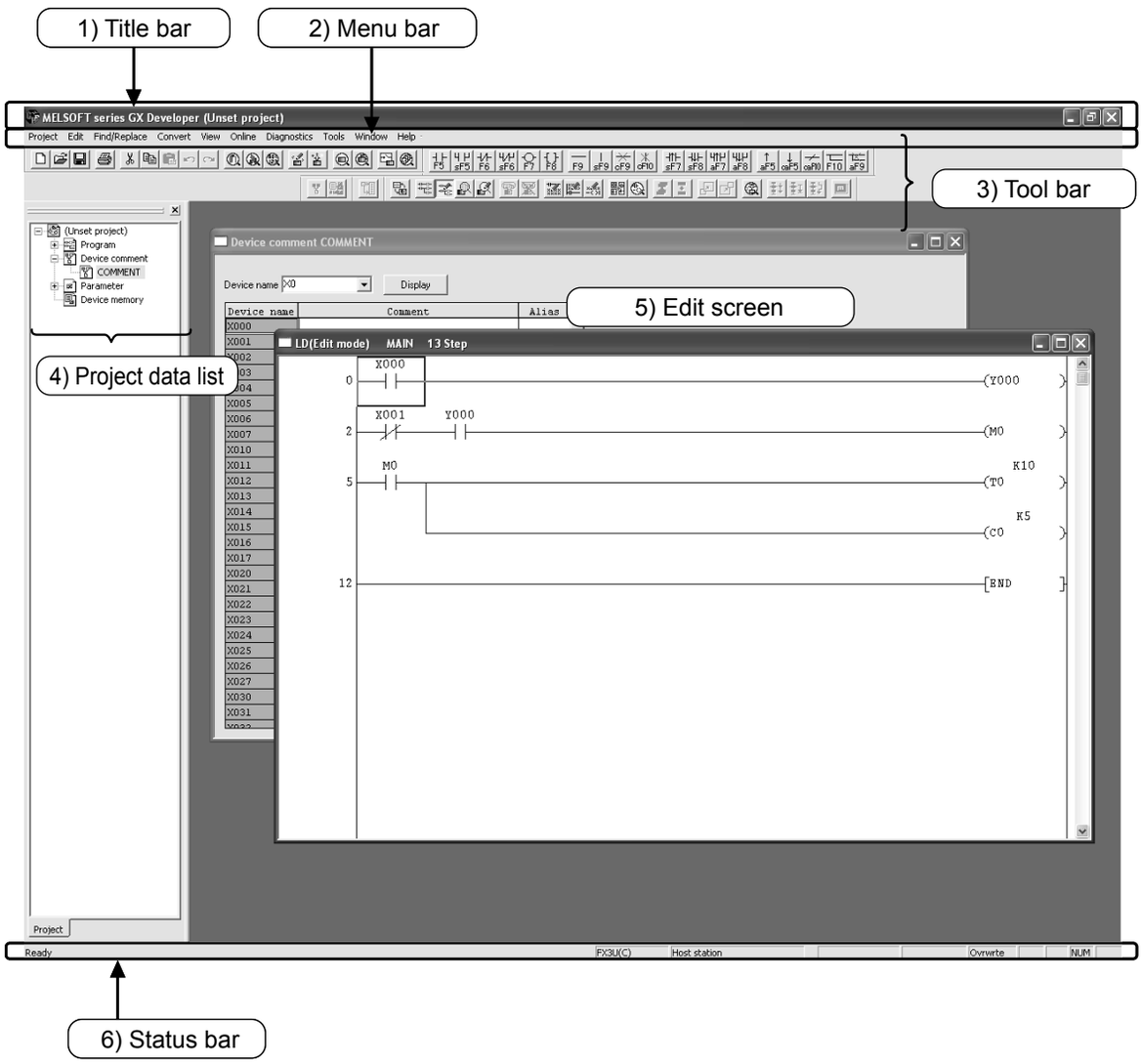
It is easy to debug programs with GX Developer and update them as necessary. The operation status of the PLC and program can be monitored with the personal computer screen, so if some parts are not working as planned, changes and updates can be conducted at once.

Make the program easy to read···

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

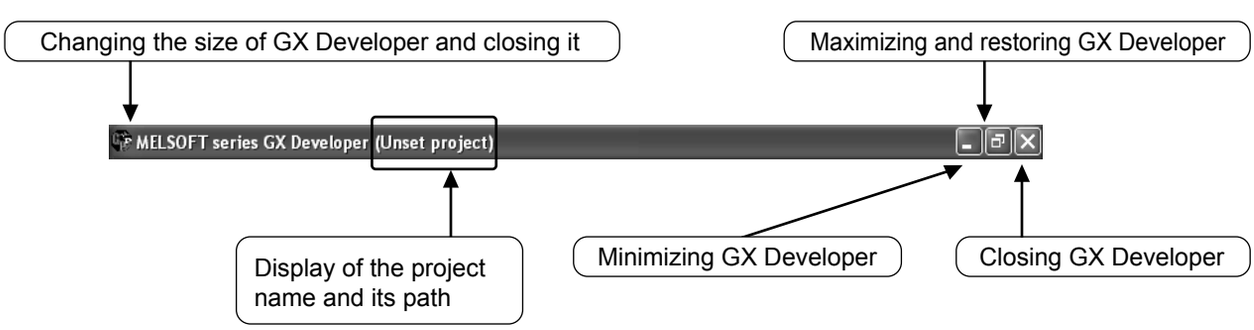
3.1 Basic knowledge for operating GX Developer

3.1.1 The layout of the GX Developer screen

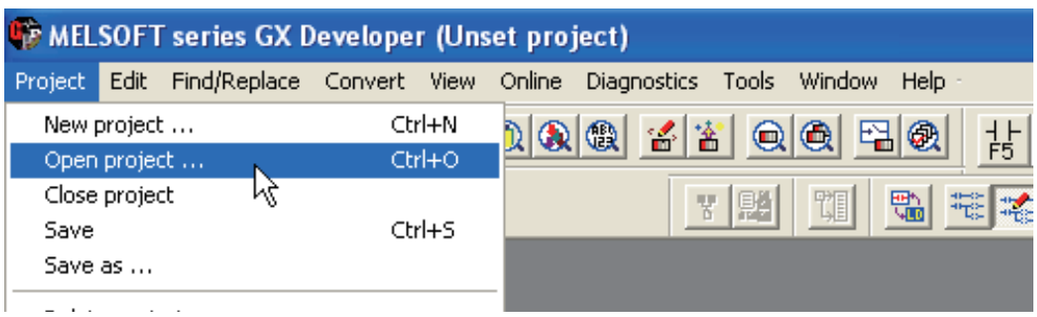


1) Title bar

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

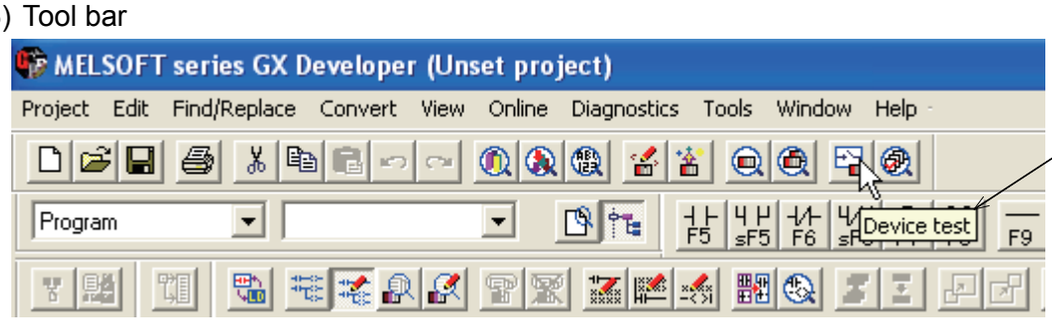


2) Menu bar



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

3) Tool bar

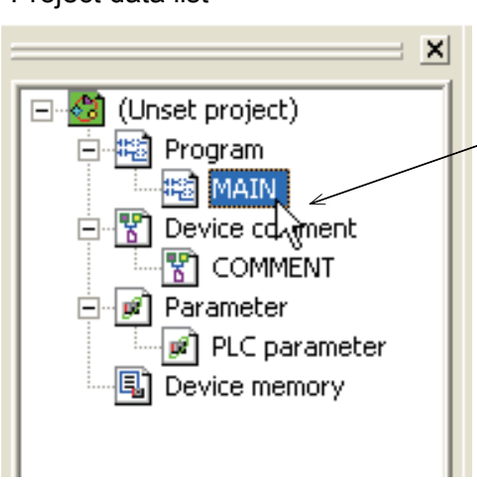


A description of the function is displayed when the mouse cursor stops over each button.

*: 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, desired functions can be directly executed.

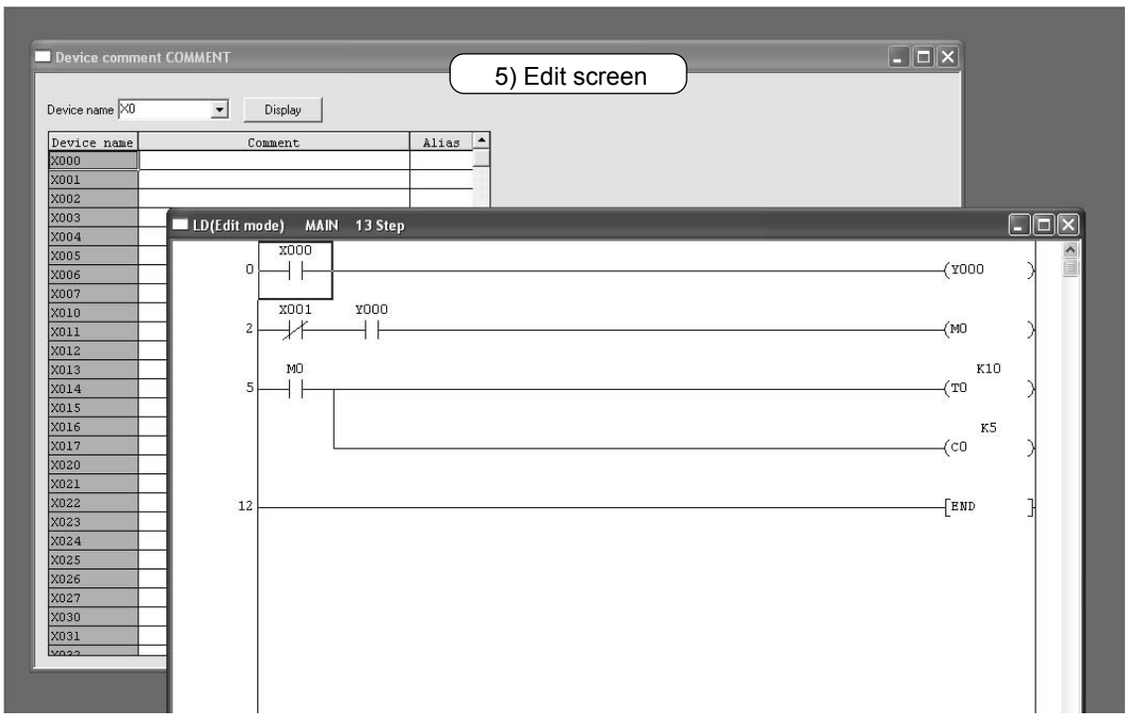
4) Project data list



Directly specify the items displayed by mouse clicking.

Circuit creating window, parameter setting screen and so on are displayed by tree structure.

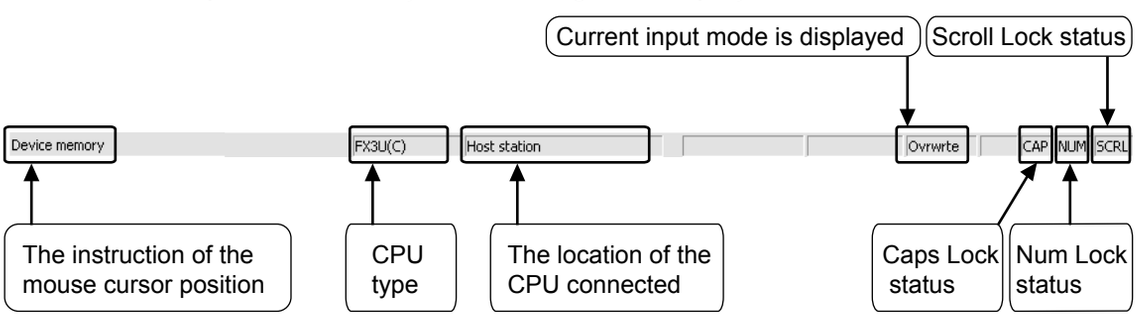
5) Edit screen



The circuit creating screen, monitor screen and so on are multiply displayed with windows.

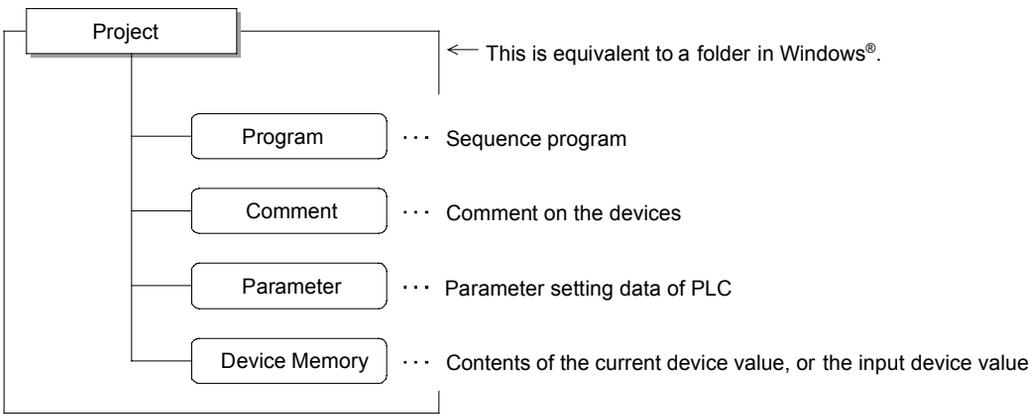
6) Status bar

The status of the operation and keyboard settings are displayed.



3.1.2 About the "Project"

A "Project" consists of Program, Comment, Parameter and Device Memory.
An aggregate of a series of data in GX Developer is called "Project", and stored as a folder in Windows®.

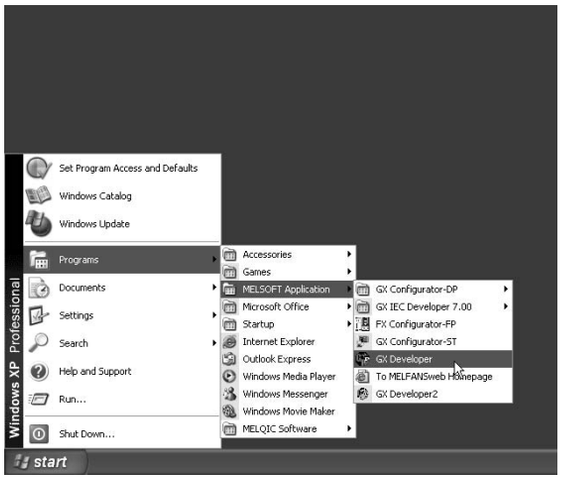


● Editing multiple projects

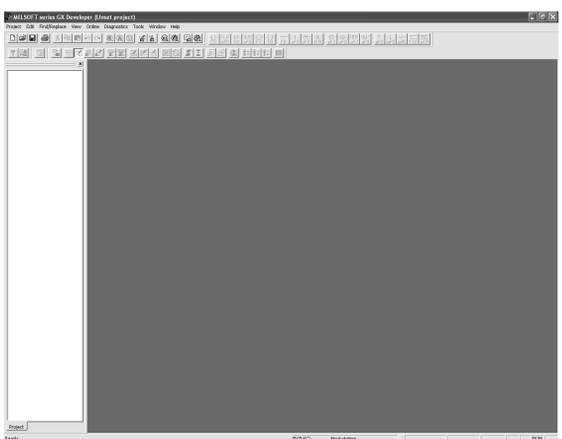
Start multiple instances of GX Developer when more than one project is to be edited.

3.2 Starting GX Developer and creating a new project

3.2.2 Starting GX Developer



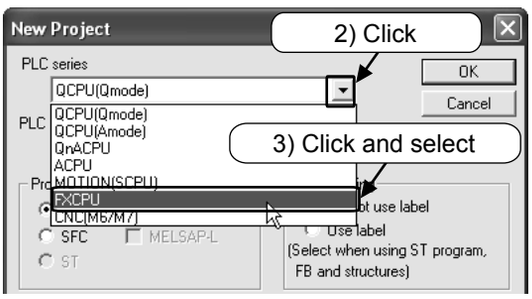
1) Start from the Start button of Windows, and select the application as follows:
   [Programs]
   ↓
   [MELSOFT Application]
   ↓
   [GX Developer]



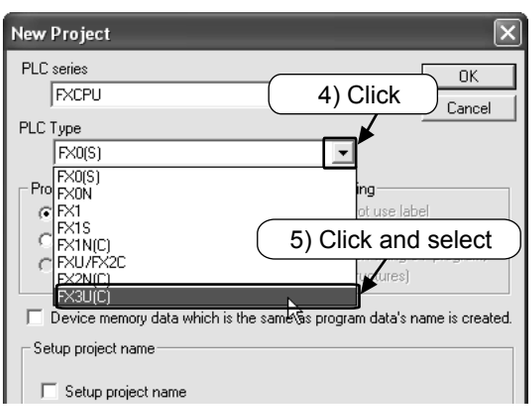
2) GX Developer is started.

3.2.3 Creating a new project

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

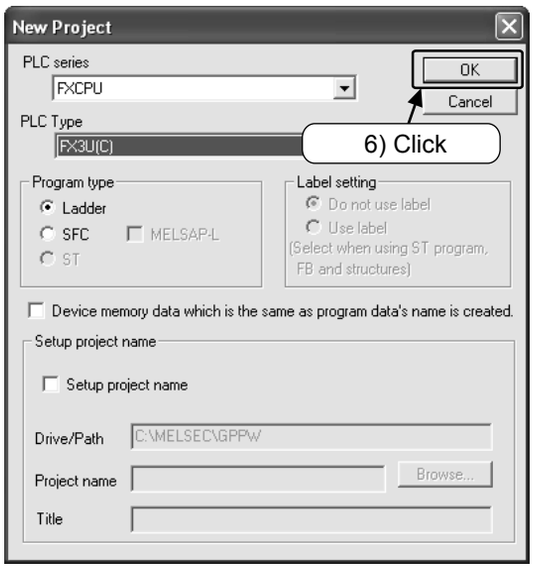


2) Click the [▼] button of [PLC series].
3) Select "FXCPU".

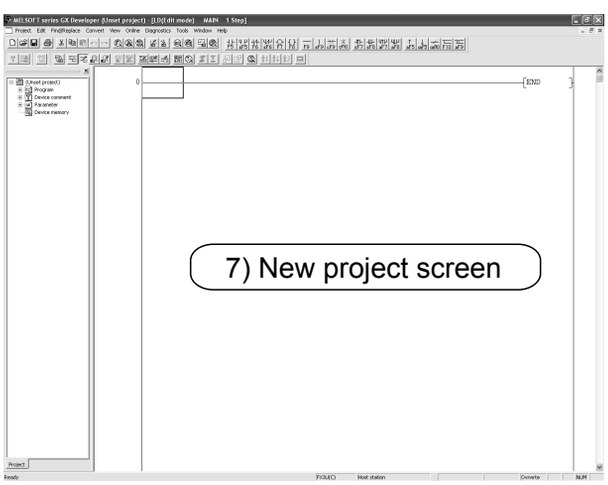


4) Click the [▼] button of [PLC type].
5) Select "FX3U(C)".

Note: Select the series name that is actually used.



6) Click OK .



7) A new project screen is displayed for project data to be input.

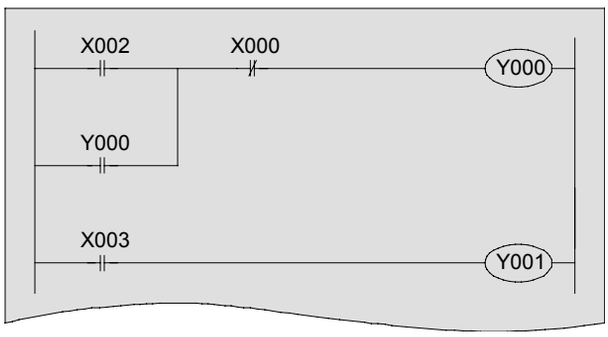
Point

- About the parameter setting
  For FX PLCs, it is not necessary to set the parameters when the comments stored in the PLC are not set or file registers are not used.
  For details on the parameters, see the Appendix.

3.3 Creating a circuit

3.3.1 Creating a circuit by using the function keys

[The circuit to be created]

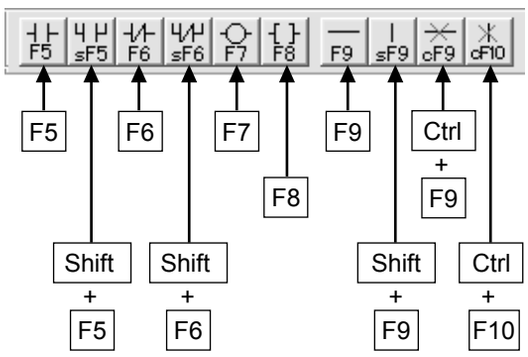


Point

In this book, the input and output relay numbers are displayed with three digits, such as "X000," and "Y000." When using GX Developer, however, "X0," "Y1," etc. may be input.

Point

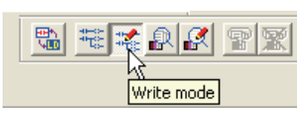
- The relationship between the function keys and the symbols of the circuit are displayed on the buttons of the tool bar.



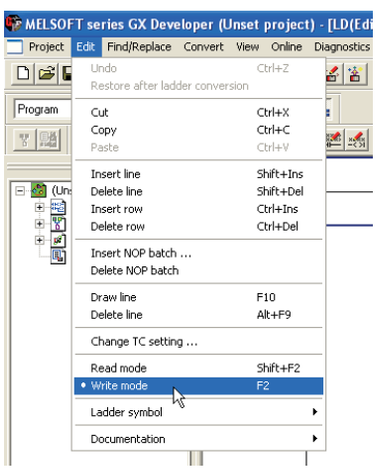
Main key operations

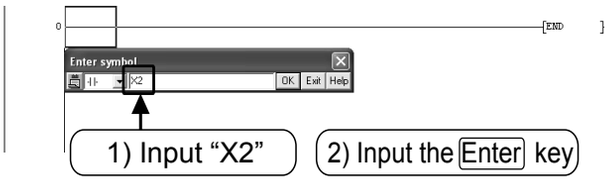
- When creating a circuit, make sure to set the mode to "Write Mode".

Select from the tool bar.



Select from the menu ([Edit] → [Write mode]).

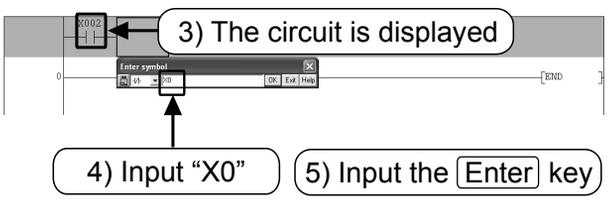




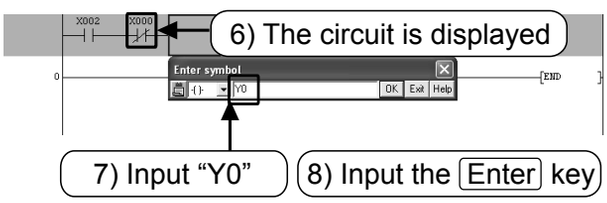
1) Press the F5 key.
Input "X2".

Cancel it by ESC or [Cancel].

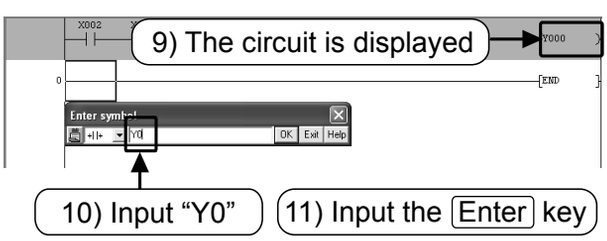
2) Confirm by pressing the Enter key or [OK].



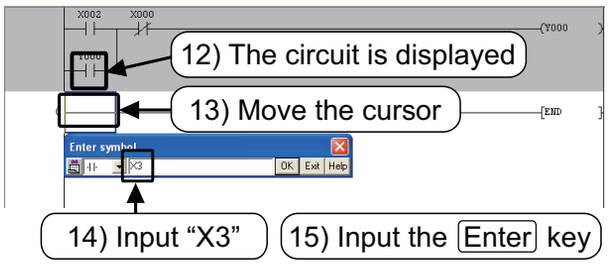
3) The circuit input X2 is displayed.
4) Press the F6 key.
Input "X0".
5) Confirm by pressing the Enter key or [OK].



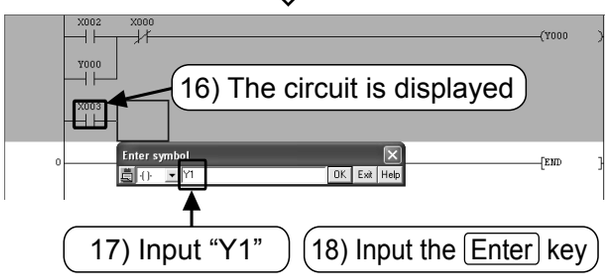
6) The circuit input X0 is displayed.
7) Press the F7 key.
Input "Y0".
8) Confirm by pressing the Enter key or [OK].



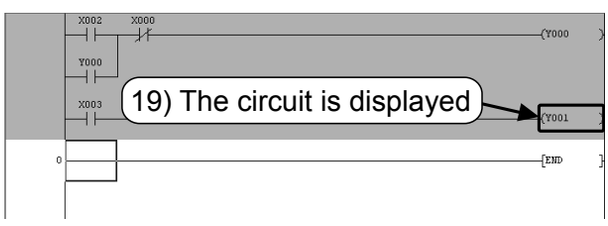
9) The circuit input (Y0) is displayed.
10) Press the Shift + F5 key.
Input "Y0".
11) Confirm by pressing the Enter key or [OK].



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].

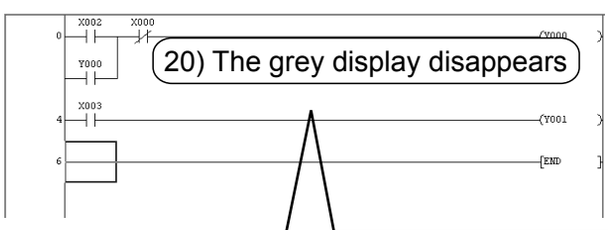


16) The circuit input (X3) is displayed.
17) Press the F7 key.
Input "Y1".
18) Confirm by pressing the Enter key or [OK].



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

F4 (Convert)



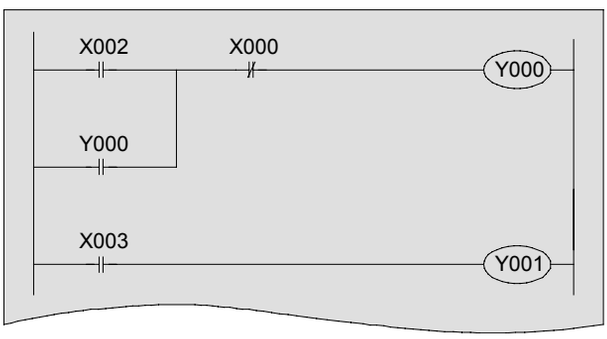
20) Circuit-Conversion Operation [Important].
Do the "Conversion" operation to confirm or compile the circuit diagram that has not been confirmed (the grey displayed part).

Press the F4 (Convert) key.
Or select from the tool bar, or select [Convert] → [Convert] from the menu.

The grey display disappears and the circuit is confirmed.
If an error occurs, the cursor moves to the failure part of the created circuit. Correct the circuit.

3.3.2 Creating a circuit by using the tool buttons

[The circuit to be created]

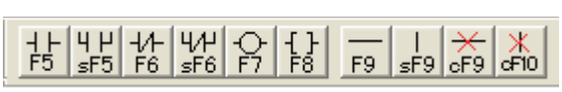


Point

In this book, the input and output relay numbers are displayed with three digits, such as "X000," and "Y000." When using GX Developer, however, "X0," "Y1," etc. may be input.

Point

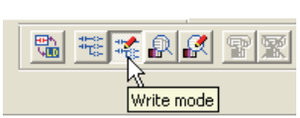
- Click the tool buttons to input the symbols of the circuit.



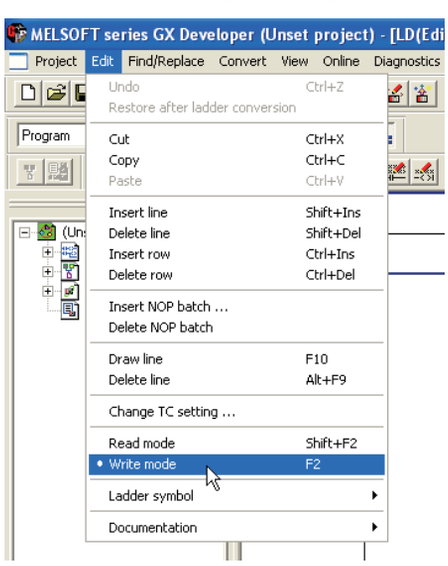
Main tool buttons

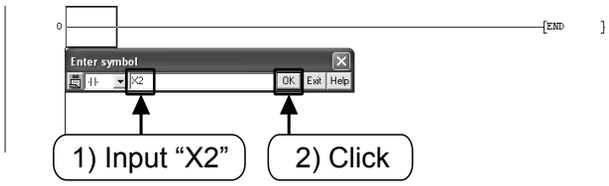
- When creating a circuit, make sure to set the mode to "Write Mode".

Select from the tool bar.



Select from the menu ([Edit] → [Write mode])

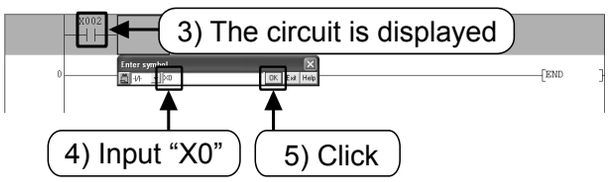




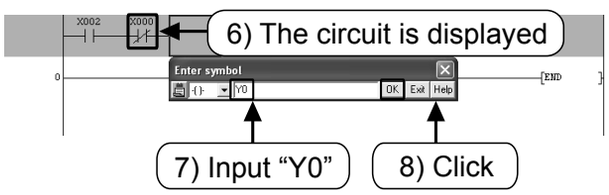
1) Click the tool button [F5].
   Input "X2".

Cancel it by [ESC] or [Cancel].

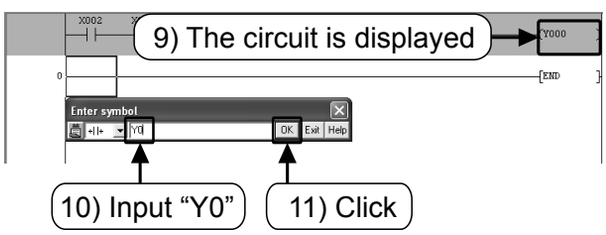
2) Confirm by pressing the [Enter] key or [OK].



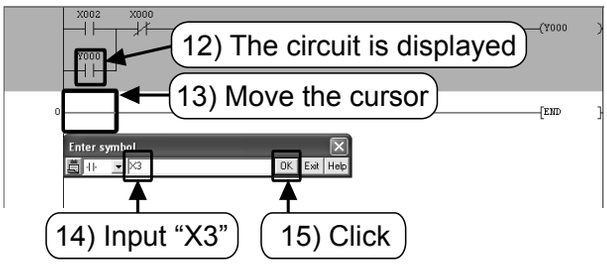
3) The circuit input (X2) is displayed.
4) Press the tool button [F6].
   Input "X0".
5) Confirm by pressing the [Enter] key or [OK].



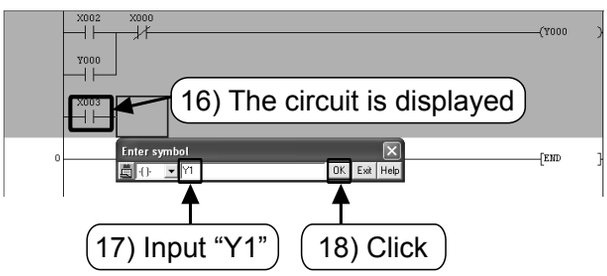
6) The circuit input (X0) is displayed.
7) Click the tool button [F7].
   Input "Y0".
8) Confirm by pressing the [Enter] key or [OK].



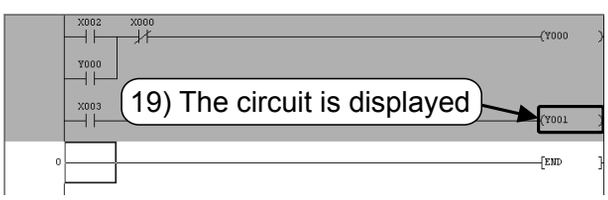
9) The circuit input (Y0) is displayed.
10) Click the tool button [sF5].
    Input "Y0".
11) Confirm by pressing the [Enter] key or [OK].



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].

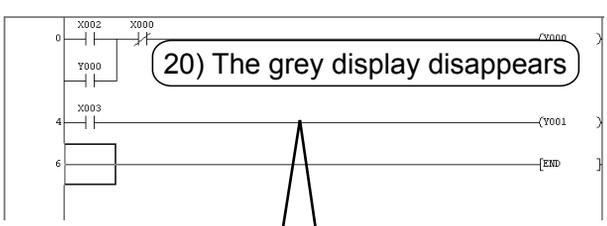


16) The circuit input (X3) is displayed.
17) Click the tool button F7 .
Input "Y1".
18) Confirm by pressing the Enter key or [OK].



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

F4 (Convert)



20) Circuit-Conversion Operation [Important].
Do the "Conversion" operation to confirm or compile the circuit diagram that has not been confirmed (grey display part).

Press the F4 (Convert) key.
Or select from the tool bar, or select [Convert] → [Convert] from the menu.

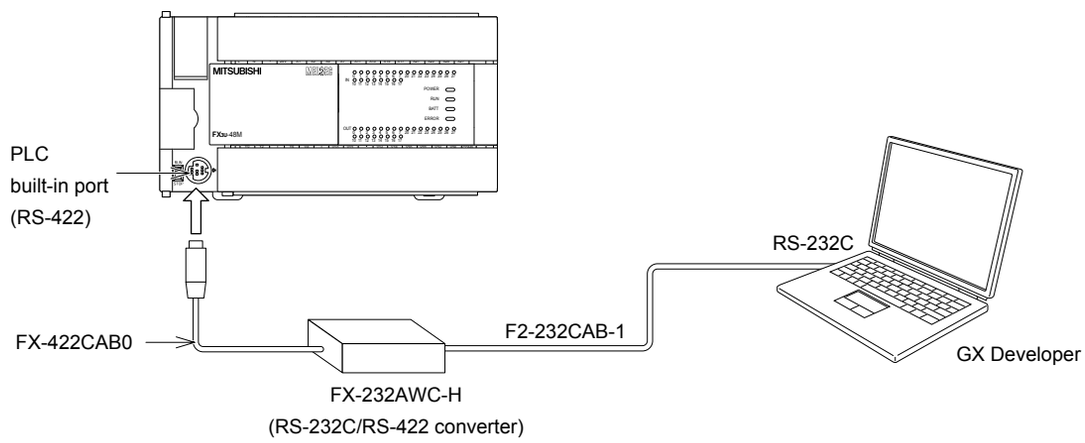
The grey display disappears and the circuit is confirmed.
If an error occurs, the cursor moves to the failure part of the created circuit. Correct the circuit.

3.4 Writing programs to the PLC

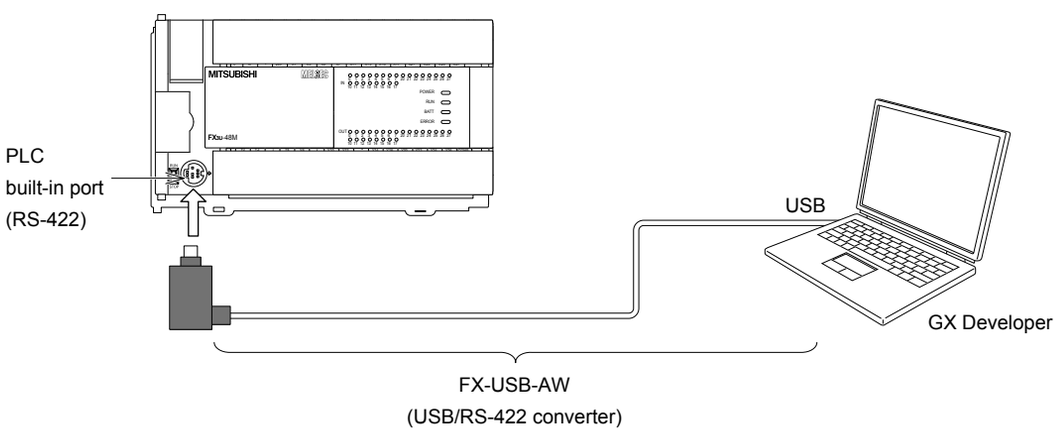
Write the created sequence program to the FX PLC.

3.4.1 Connecting PC to PLC

1) Example for connection (Personal computer side: RS-232C)

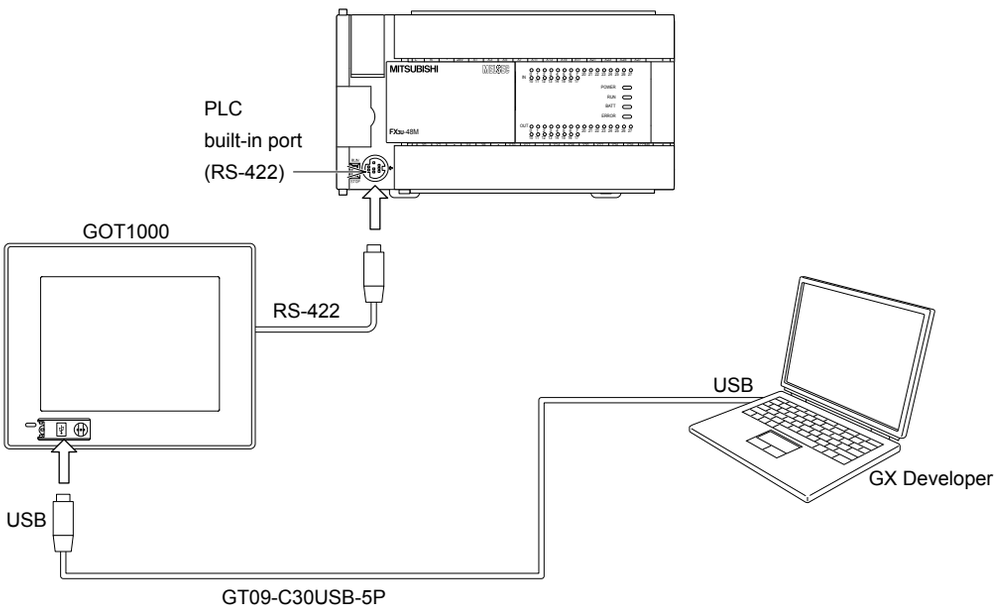


2) Example for connection (Personal computer side: USB)



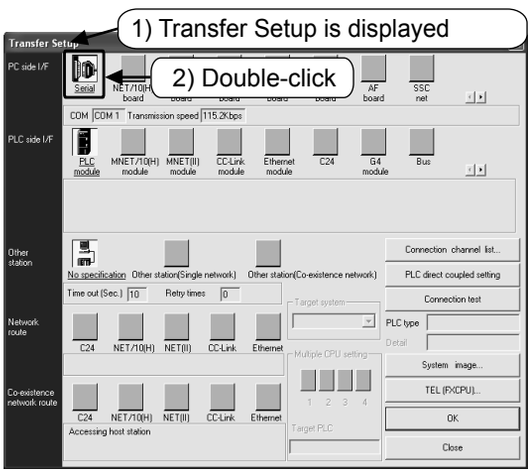
- Check the COM port number that the driver of the FX-USB-AW assigns to the personal computer. For the check procedures, see the manual of the FX-USB-AW.

3) Transparent function of GOT1000 (Personal computer side: USB)

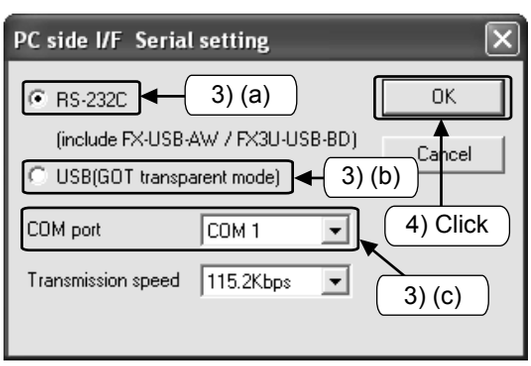


3.4.2 "Transfer Setup" in GX Developer

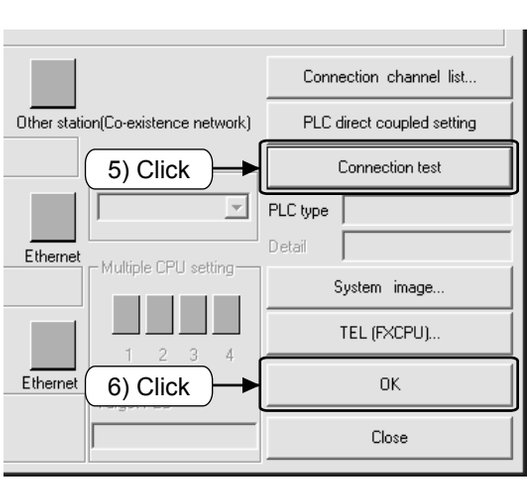
Configure the settings of GX Developer to communicate with the PLC.



1) Select [Online] → [Transfer Setup].
2) Double-click the icon .

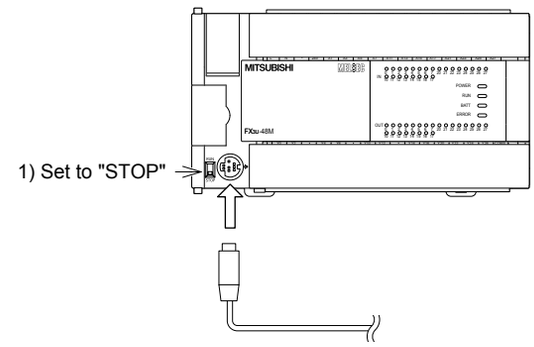


3) Set the communication port of the personal computer side.
   (a) Select "RS-232" when an RS-232 connector is used at the personal computer side or an FX-USB-AW is used with the USB connector at the personal computer side.
   (b) Select "USB (GOT transparent mode)" when the transparent function of GT1000 is used with the USB connector at the personal computer side.
   (c) · When an RS-232C connector is used at the personal computer side, the port is usually COM1.
   (This may change depending on the personal computer.)
   · Specify the COM port number that the driver assigns when using the FX-USB-AW. (See Section 3.7.1)

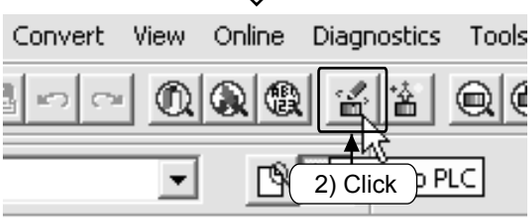


4) Click [OK] after the setting is completed.
5) Click [Connection test], to check the communication with the PLC.
6) After checking, click [OK] to confirm the configured setting.

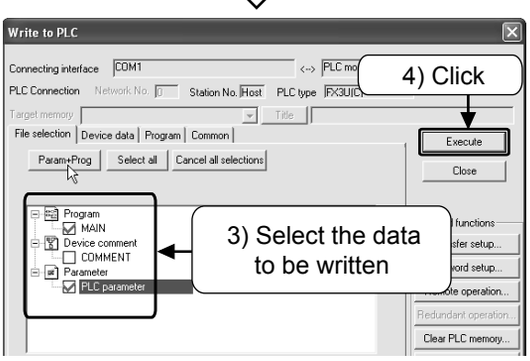
3.4.3 Writing a program to the PLC



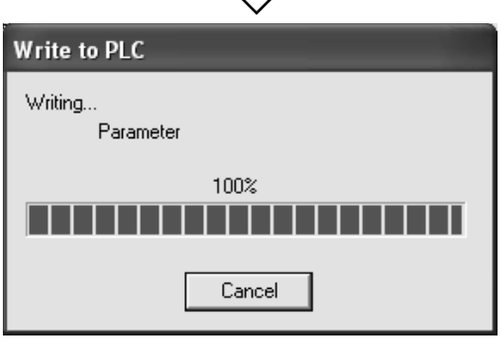
1) Set the "RUN/STOP" switch of the PLC to "STOP".



2) Select from the tool bar or select [Online] → [Write to PLC] from the menu.



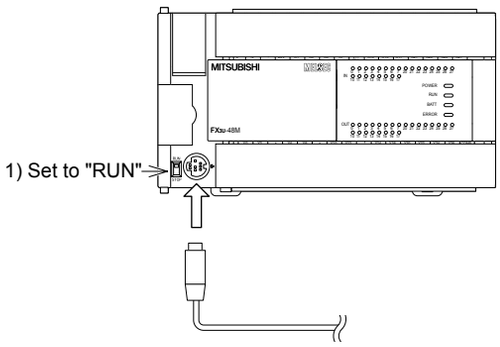
3) Click [Param + Prog].
4) Click [Execute].



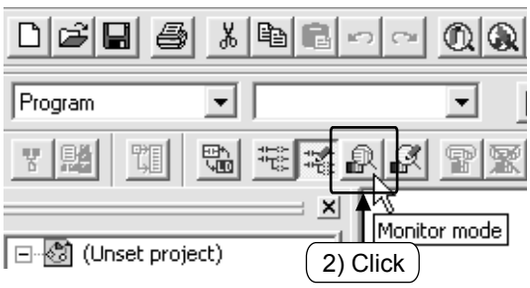
The dialog box of the progressing rate is displayed.

5) Click [OK] after it is completed.

3.4.4 Operation monitor of a program

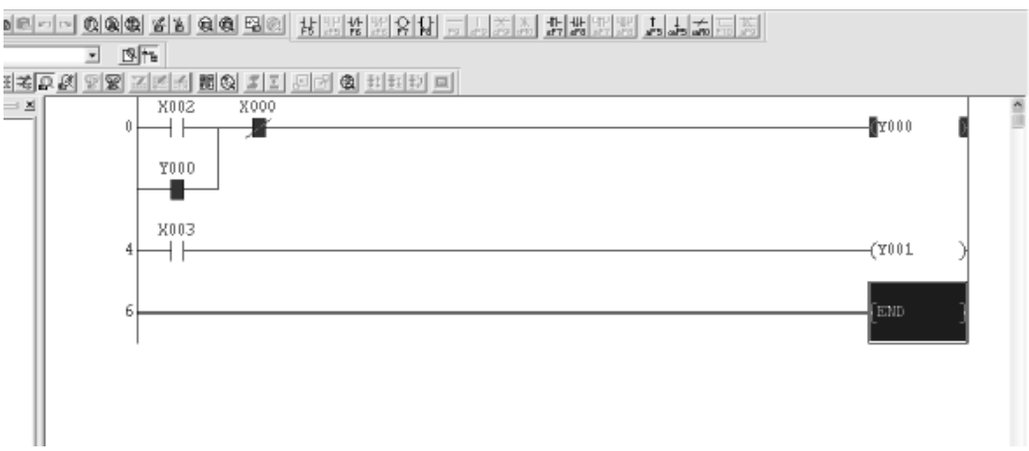


1) Set the "RUN / STOP" switch of the PLC to "RUN".



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

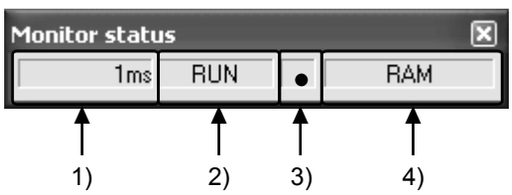
Operation check by operation monitor



1) Set [Switch X002 is "ON"] with the status [Switch X000 is "OFF"], and then check [Output Y000 is "ON"].
2) Check [Output Y000 is "ON"] while [Switch X002 is "OFF"].
3) Set [Switch X000 is "ON"] and then check [Output Y000 is "OFF"].
4) Check [Output Y001 is "ON/OFF"] in accordance with [Switch X003 is "ON/OFF"].

Reference

(1) The display of the monitor status dialog



1) Scan time
   The maximum scan time of the sequence program is displayed.
2) PLC status
   The status of the PLC is displayed.
3) The execution status of the monitor.
   This icon is flashing during monitor mode.
4) Memory type display
   The memory type of the PLC is displayed.

(2) The interpretation of the status display for the ladder monitor

1) Contact Instruction

Type \ Input contact	X0: OFF	X0: ON
NO contact	X000 ⊣ ⊢ Circuit open	X000 ⊣ ■ ⊢ Circuit close
NC contact	X000 ⊣ ■ ⊢ Circuit close	X000 ⊣ / ⊢ Circuit open

2) Out Instruction

Type \ Driving status	Non-execution/ Non-drive	Execution/Drive
⊣( )⊢ OUT instruction	⊣(Y000)⊣	⊣ ■Y000■ ⊢
⊣[ ]⊢ SET instruction, etc	⊣[SET M0]⊢	⊣ ■SET M0■ ⊢

The ON/OFF status of the device to be reset is displayed during monitor mode using the RST instruction.

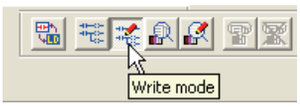
Type \ Device status	When device to be reset is OFF	When device to be reset is ON
⊣[ ]⊢ RST instruction	⊣ ■RST M0■ ⊢	⊣[RST M0]⊢

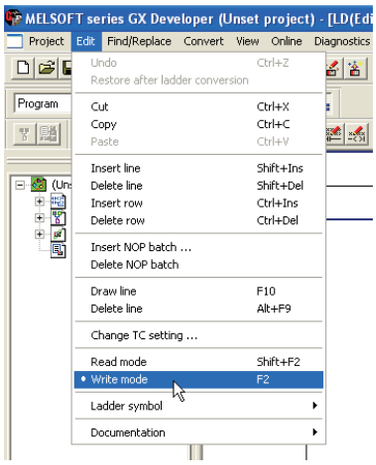
3.5 Editing a circuit

3.5.1 Correcting a circuit

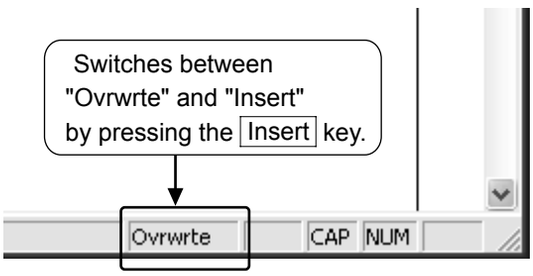
Point

- Make sure to set the mode to "Write Mode" when amending the circuit.
  Select from the tool bar. Select from the menu ([Edit]→[Write mode]).





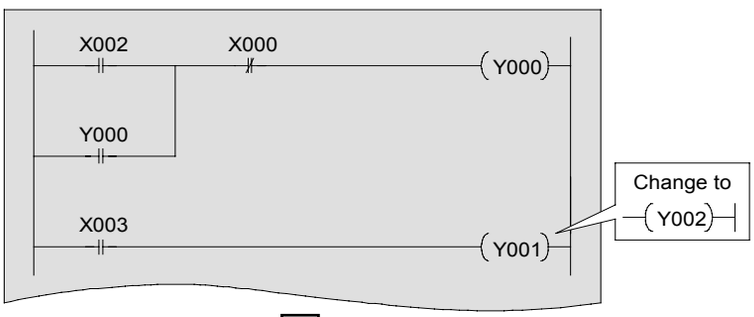
- Switch between "Ovrwrte" and "Insert"
  - Set to "Ovrwrte" when correcting and overwriting a circuit diagram.
  - A new circuit will be inserted when the "Insert" mode is on.

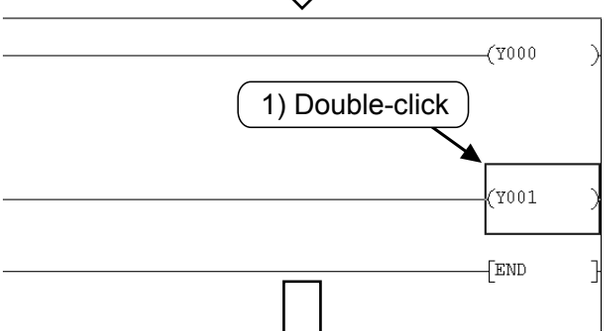


3

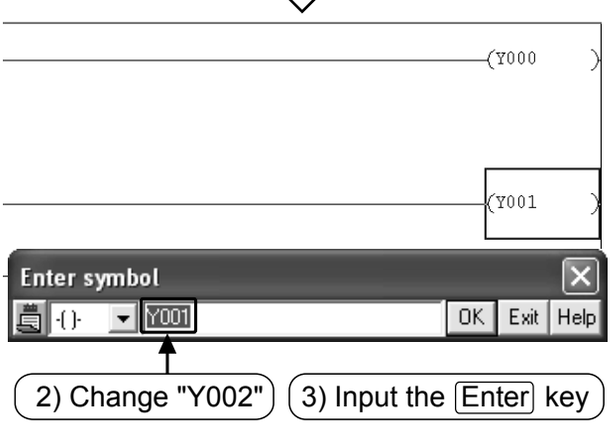
1) Changing the OUT coils and contacts

[The circuit to be corrected]

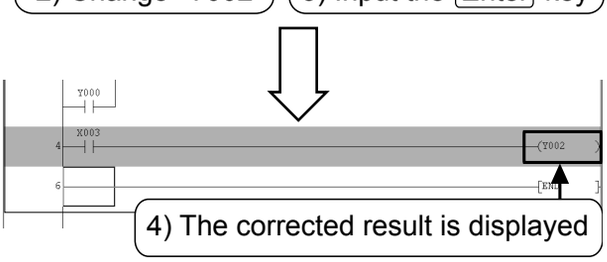




1) Double-click the part that needs to be corrected.

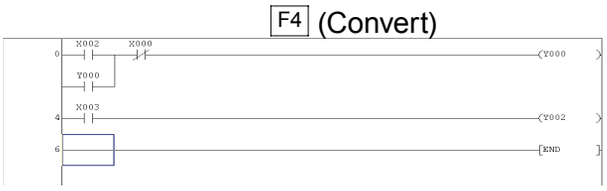


2) Change "Y001" to "Y002".



3) Confirm by pressing the Enter key or [OK].
4) The corrected result is displayed and the circuit block is displayed in grey.

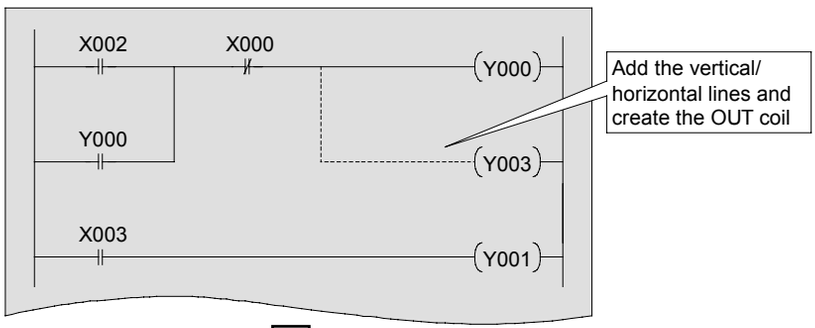
F4 (Convert)

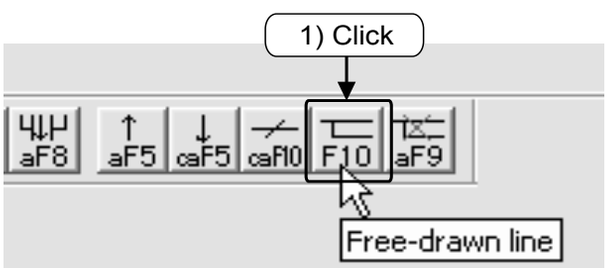


5) Confirm the changes by pressing the F4 (Convert) key.

2) Adding lines

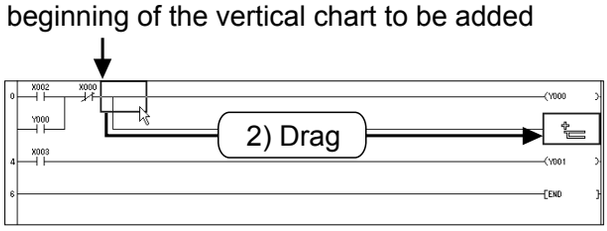
[The circuit where lines are to be added]



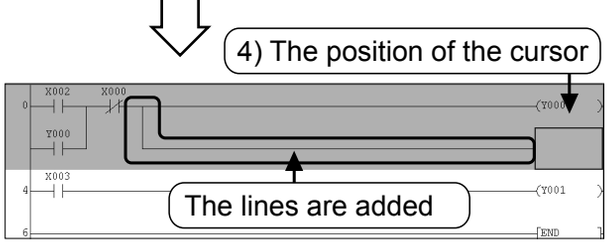


1) Click (F10) on the tool bar.

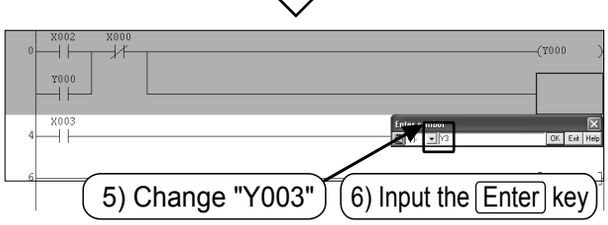
Locate the cursor on the upper right of the beginning of the vertical chart to be added



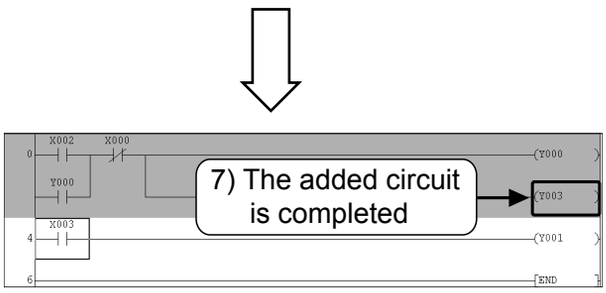
2) Locate the cursor on the upper right of the desired vertical line to be added, and then drag it until it reaches the desired position, and then drop it.



3) The line reaching the dropped position is added.
4) Locate the cursor on the position where the OUT coil is to be added and click on the tool bar.

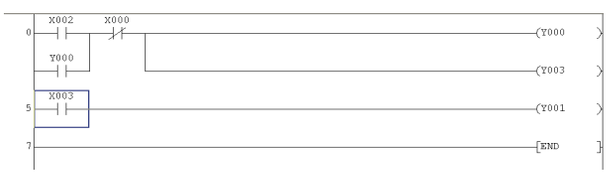


5) Input "Y3".
6) Confirm by pressing the Enter key or [OK].



7) The added circuit is finished and the circuit block is displayed in grey.

F4 (Convert)

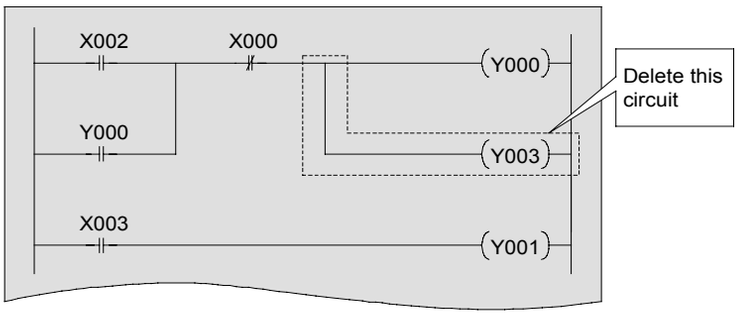


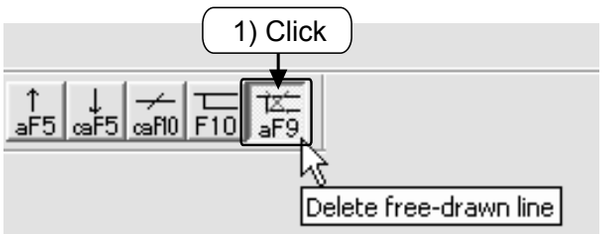
8) Confirm the changes by pressing the F4 (Convert) key

- Click F10 on the tool bar again to finish the operation.

3) Deleting lines

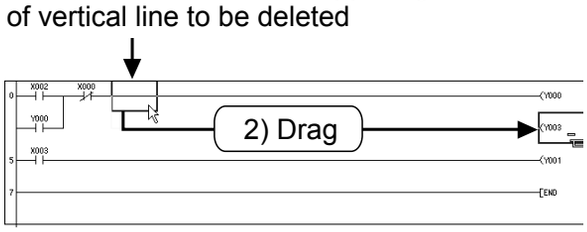
[The circuit where lines are to be deleted]



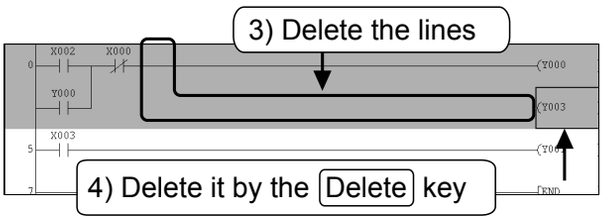


1) Click aF9 Alt + F9 on the tool bar.

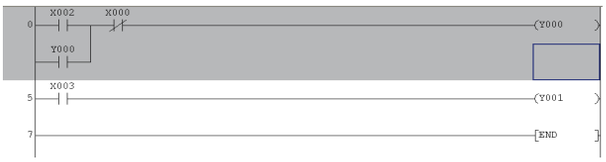
Locate the cursor on the upper right of vertical line to be deleted



2) Locate the cursor on the upper right of the desired vertical line to be deleted, and then drag it until it reaches the desired position, and then drop it.

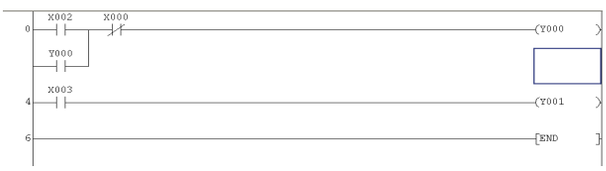


3) The lines are deleted.
4) Delete the OUT coil by pressing the Delete key.



5) The deleted circuit block is displayed in grey.

F4 (Convert)



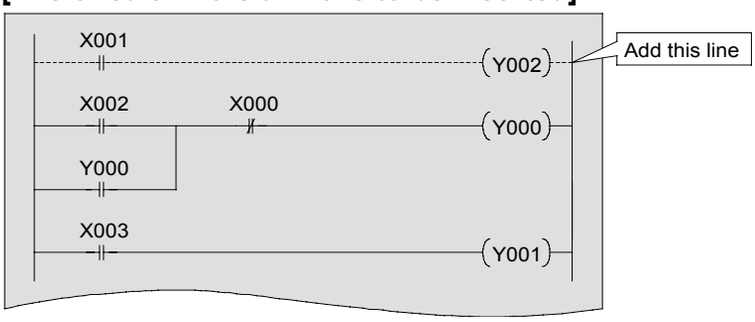
6) Confirm the changes by pressing the F4 (Convert) key.

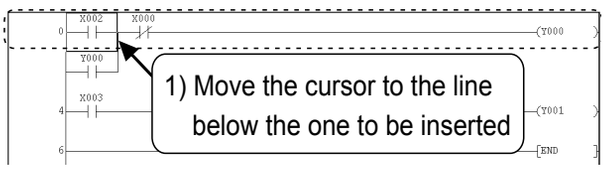
• Click on the tool bar again to finish the operation.

3.5.2 Inserting and deleting lines

1) Adding lines

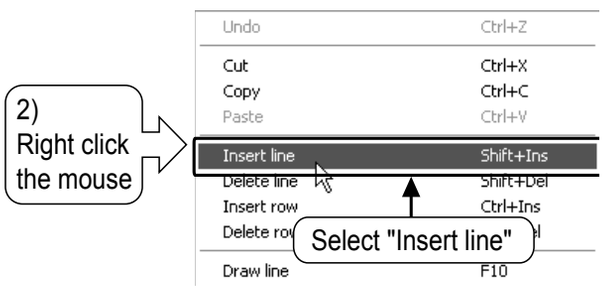
[The circuit where a line is to be inserted]



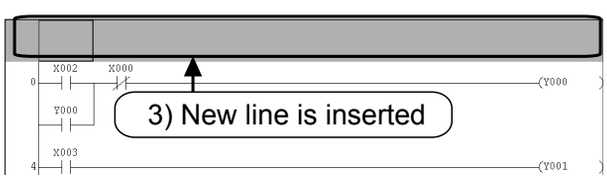


A line is inserted above the line where the cursor is located

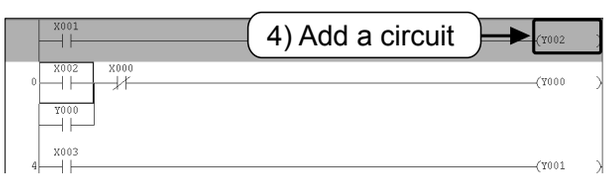
1) Locate the cursor on the line below the one to be inserted.



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



3) A line is inserted.

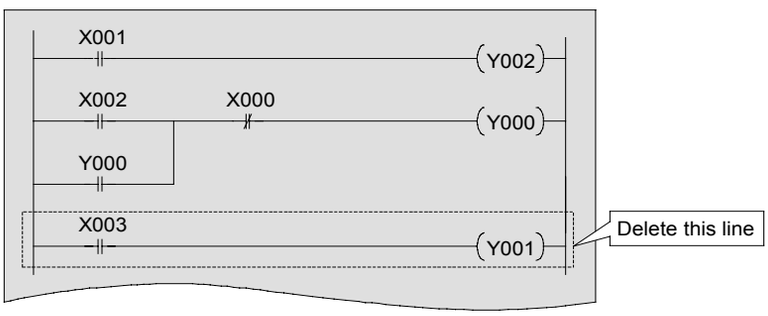


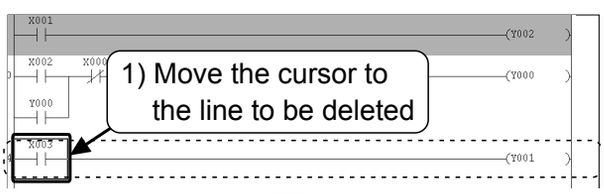
4) Add a program in the inserted line.

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

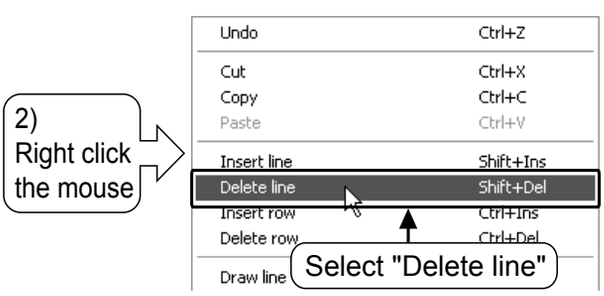
2) Deleting lines

[The circuit where a line is to be deleted]

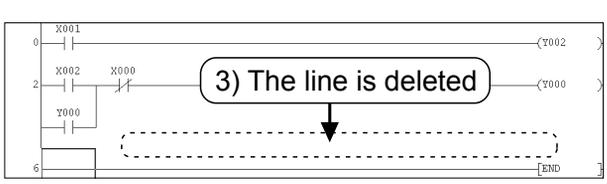




1) Move to the line to be deleted



2) Right click the mouse at any place, and select [Delete line]



3) The line is deleted

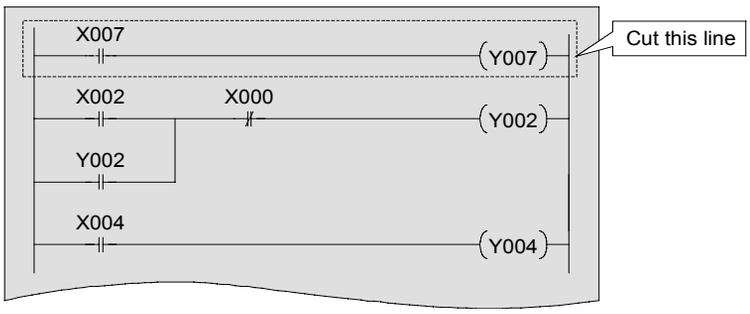
Point

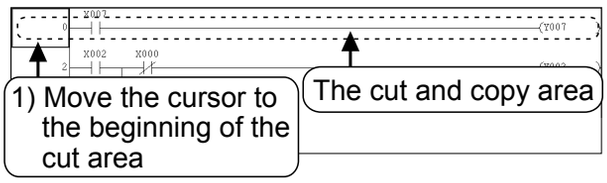
Confirm it by pressing Convert (F4).

3.5.3 Cutting and copying (pasting) a circuit

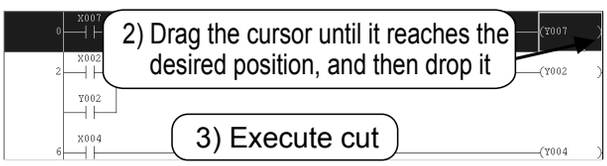
1) Cut

[The circuit to be edited]

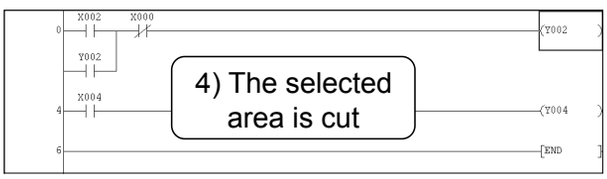




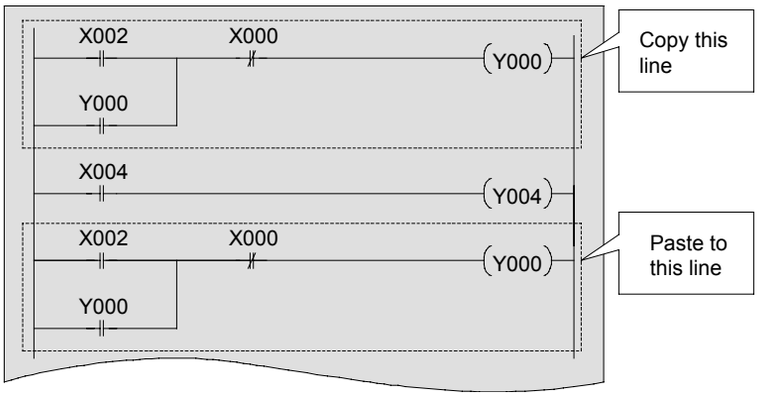
1) Move the cursor to the beginning of the circuit to be cut.

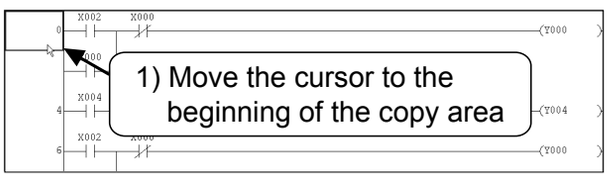


2) Drag it until it reaches the desired position, and then drop it.
3) Select from the tool bar or select [Edit] → [Cut] (Ctrl + X) from the menu, and execute the cut.



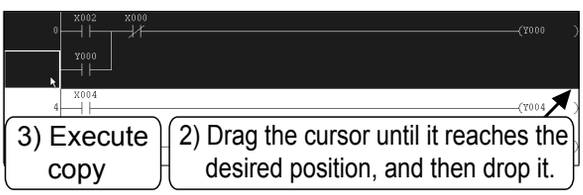
4) The selected area is cut
A grey part remains when a smaller portion of the circuit is cut. After ammending the circuit, confirm the changes by pressing the F4 (Convert) key.



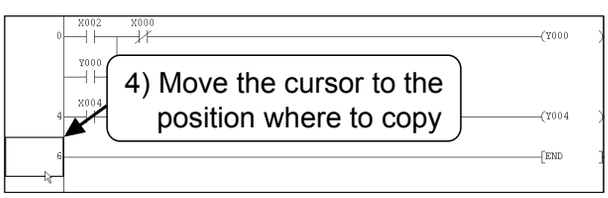


Continue to edit the circuit with the "cut" operation performed in the previous steps.

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



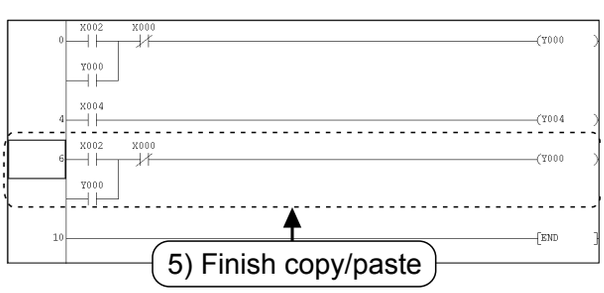
2) Drag the cursor until it reaches the desired position, and then drop it.
3) Select from the tool bar or select [Edit] → [Copy] (Ctrl + C) from the menu.



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

Point

Using by the Insert key
"Ovrwrte" 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.

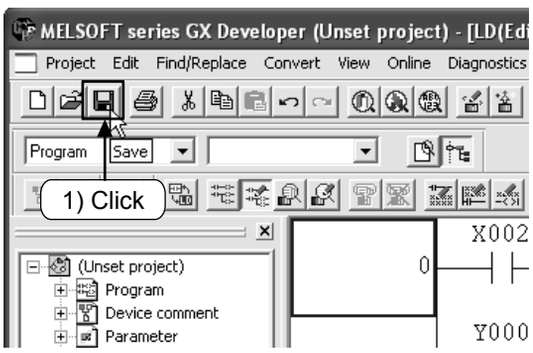
A grey part remains when a smaller portion of the circuit is pasted. After ammending the circuit, confirm the changes by pressing the F4 (Convert) key.

3.6 Saving a created circuit

3.6.1 Save and Save as

Point

If there are circuits that have not been converted in the program, press Convert (F4).

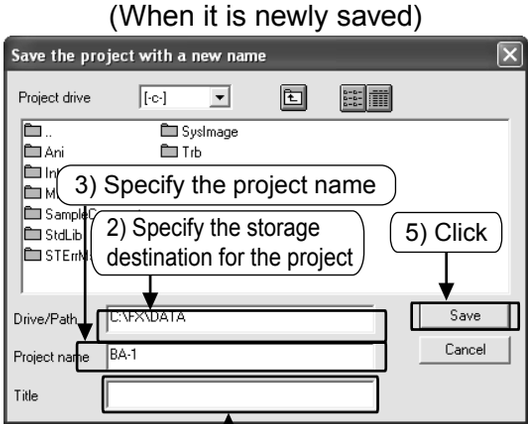


1) Select from the tool bar or select [Project] → [Save] (Ctrl + S) from the menu.

(When it is saved by overwriting)

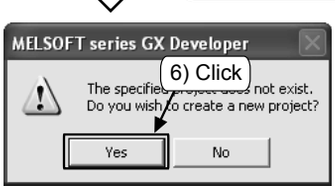
Project saving is finished

(When it is newly saved)



2) Specify the storage destination for the project.
3) Specify the project name.
4) Specify the title describing the project (optional).
5) Click Save .

4) Specify the title (optional)



6) Click Yes in the confirmation dialog to finish.

If there is not enough space to save to a floppy disk, temporarily save the project to the hard disk and then move it to another floppy disk.

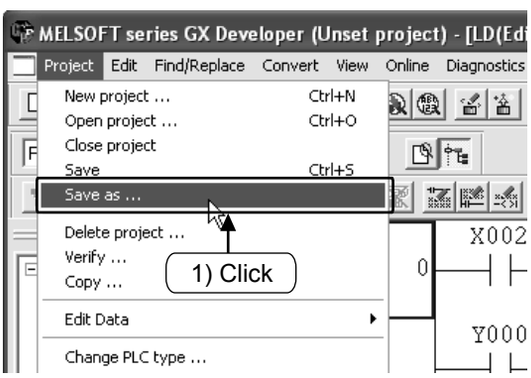
Reference

- The following characters cannot be used in the project name.
  / ,¥ ,> ,< ,* ,? ," ," , | , : , ; (; , ¥ are only used to specify the driver)
  Also, do not use a "." (period) at the end of the project name.
- When the project name is specified with 8 or more characters by GX Developer (later than SW6D5-GPPW), characters past the 8th character will not be displayed if read by the old versions (older than SW2D5-GPPW) of GX Developer.
- The project path plus its name is within 150 half-width characters (75 full-width characters).
- The tile is within 32 half-width characters (16 full-width characters).
- If there are spaces in the project path and project name, GX Developer cannot start normally even if GPPW.gpj, ***.gps is double-clicked in the Explorer window.
  If there are spaces in the project path and project name, open the project by starting GX Developer → selecting [Project] → [Open project] from the menu.

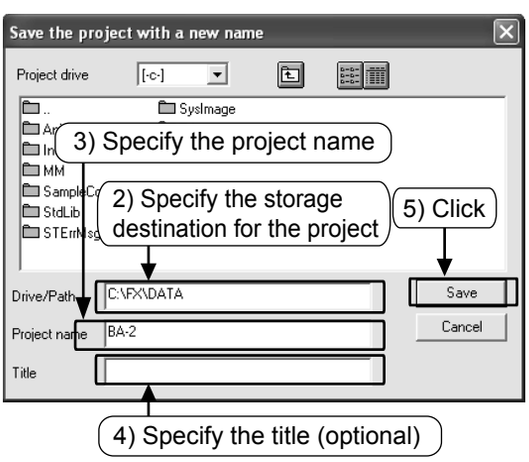
3.6.2 Saving a project as a new one

Point

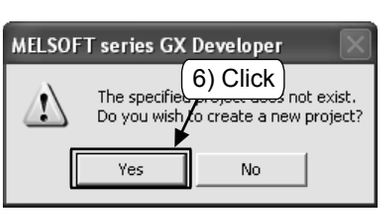
If there are circuits that have not been converted in the program, press Convert (F4).



1) Select [Project] → [Save as] from the menu.



2) Specify the storage destination for the project.
3) Specify the project name.
4) Specify the title describing the project (optional).
5) Click Save .



6) Click Yes in the confirmation dialog to finish.

For details on how to name the driver/path and the project, see the previous page.

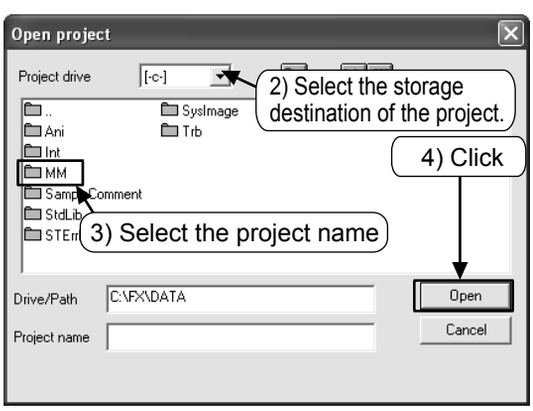
If there is not enough space to save to a floppy disk, temporarily save the project to the hard disk and then move it to another floppy disk.

3.6.3 Reading a project

Reference

If another project is open when reading/opening a new file, the current project is closed.
If there are circuits not converted in the project or the project is not saved, a warning message is displayed.

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



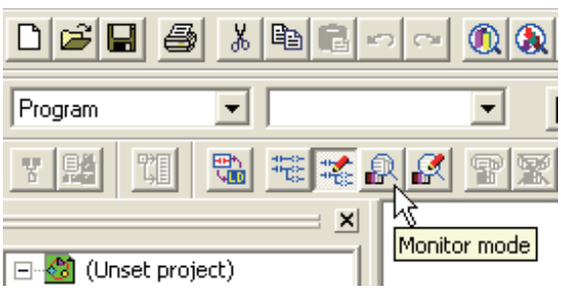
2) Select the storage destination of the project.
3) Select the project to be read.
4) Click Open and read the project.

3.7 Necessary operation for debugging a program

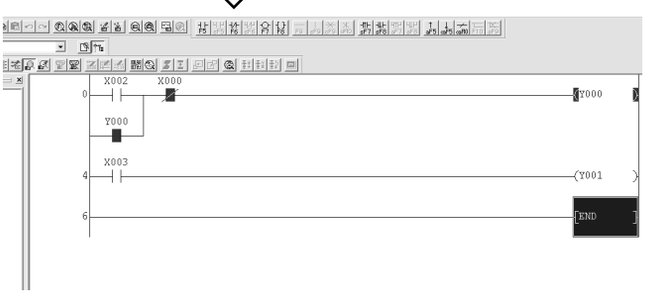
For connecting to the PLC and writing a program to the PLC, see "3.4 Writing programs to the PLC".

3.7.1 Circuit monitor

Display the circuit, and monitor the conduction status of the contacts and the driving status of the coils



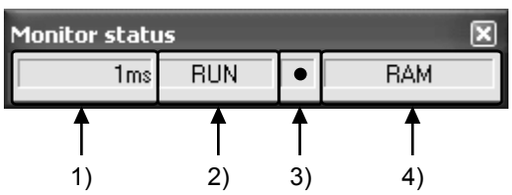
1) Select from the tool bar or select [Online] → [Monitor] → [Monitor mode] from the menu.



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.
3) Right click the window, select [Stop monitor] to quit the circuit monitor.
4) In order to correct and write the program, select from the tool bar or select [Edit] → [Write mode] (F2) from the menu.

Reference

(1) the display of the monitor status dialog



1) Scan time
   The maximum scan time of the sequence program is displayed.
2) PLC status
   The status of the PLC is displayed.
3) The execution status of the monitor
   This icon is flashing during monitor mode.
4) Memory type display
   The memory type of the PLC is displayed.

(2) The interpretation of the status display for monitor mode

1) Contact Instruction

Type \ Input contact	X0: OFF	X0: ON
NO contact	X000 ⊣⊢ Circuit open	X000 ⊣▌⊢ Circuit close
NC contact	X000 ⊣▌⊢ Circuit close	X000 ⊣/⊢ Circuit open

2) Out Instruction

Type \ Driving status	Non-execution/ Non-drive	Execution/Drive
—( )⊣ OUT instruction	—(Y000)⊣	—▌Y000▌⊣
—[ ]⊣ SET instruction, etc.	—[SET M0]⊣	—▌SET M0▌⊣

The ON/OFF status of the device to be reset is displayed during monitor mode using the RST instruction.

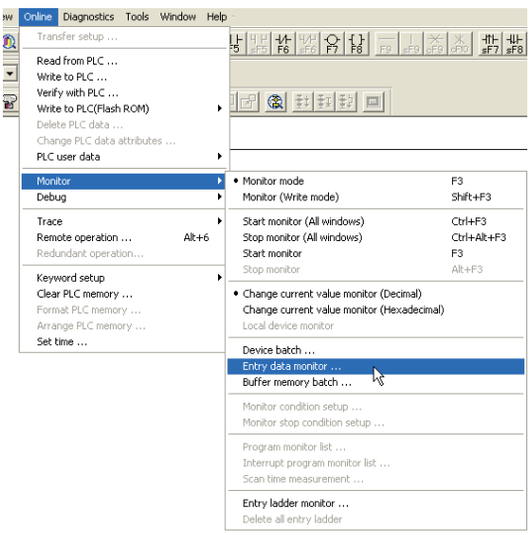
Type \ Device status	When device to be reset is OFF	When device to be reset is ON
—[ ]⊣ RST instruction	—▌RST M0▌⊣	—[RST M0]⊣

3.7.2 Device registration monitor

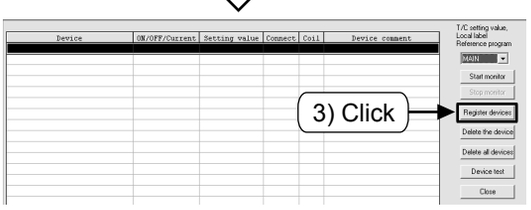
1) Registering optional devices

Register optional devices in the monitor window and monitor only the necessary parts.

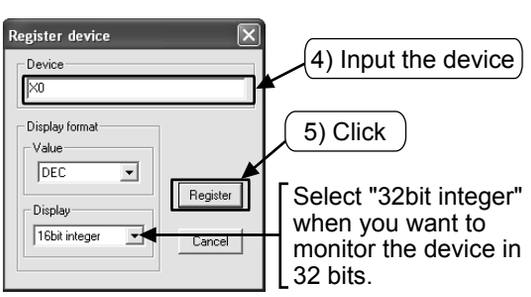
1) Set to monitor mode. (See Section 3.7.1.)



2) Select [Monitor] → [Entry data monitor] from the menu. Or right click the circuit window and select [Entry data monitor].

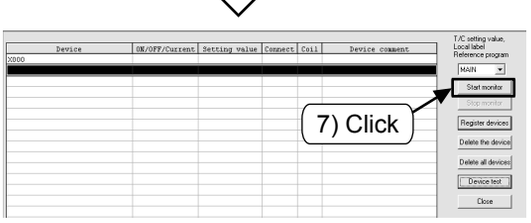


3) Click [Register devices] in the "Entry data monitor" window.



4) Input the device number to be registered in the Register device window.

5) Click [Register].

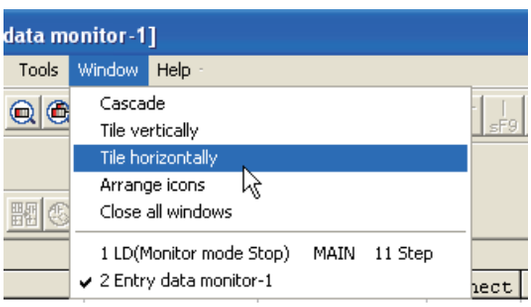


6) The device is registered in the monitor window.

7) Click [Start monitor], and the value showing the device action and the ON/OFF status of the contacts and coils are displayed.

2) Registering the devices displayed in monitor mode

Specify the area of the ladder diagram in the ladder monitor window and register all of the devices in the area.

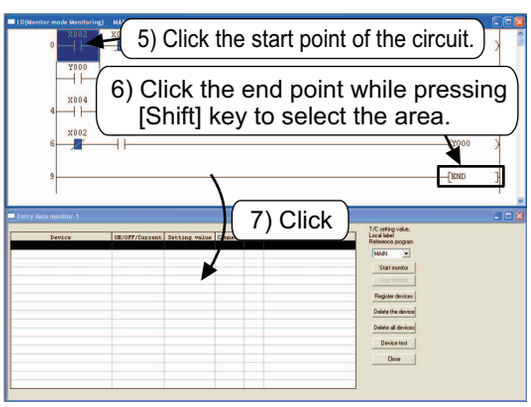


1) Set to monitor mode. (See Section 3.7.1.)

2) Select [Monitor] → [Entry data monitor] from the menu. Or right click the ladder window and select [Entry data monitor]. (See the previous page.)

3) Select [Window] → [Tile horizontally] from the menu to display the "Ladder window" and "Entry data monitor window" to show both windows together. (Set "Entry data monitor window" to the status of stop monitoring.)

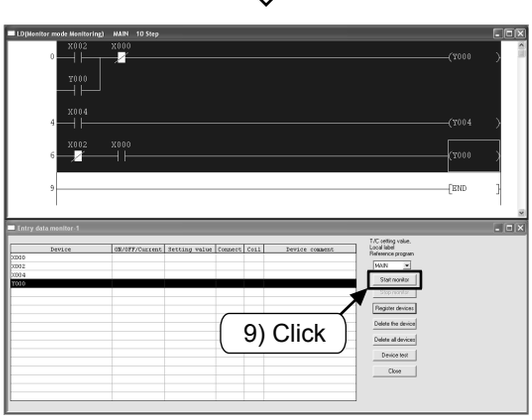
4) The "Ladder window" and "Entry data monitor window" are displayed horizontally.



5) Click the start point of the circuit.

6) Click the end point while pressing the [Shift] key to select the area.

7) Drag the selected area to the "Entry data monitor window".



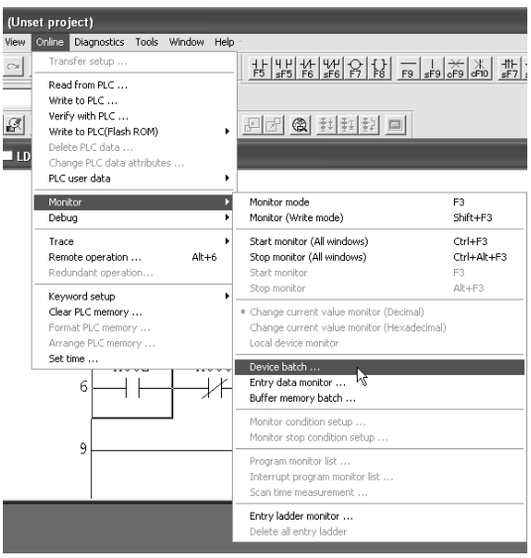
8) The devices are registered to the monitor window.

9) Click [Start monitor], and the value showing the device action and the ON/OFF status of the contacts and coils are displayed.

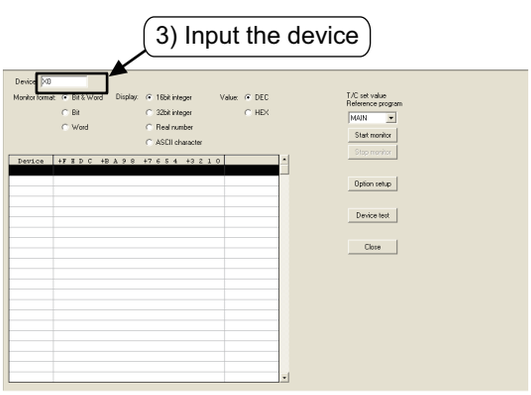
3.7.3 Device batch monitor

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

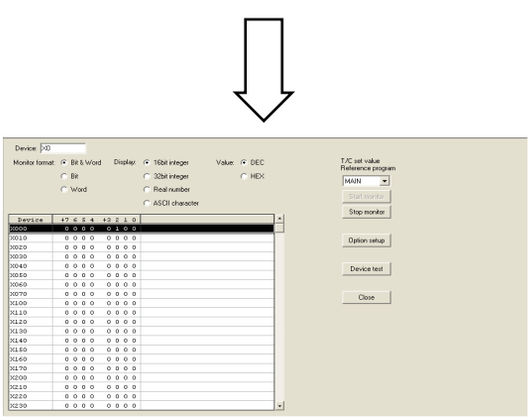
1) Set to monitor mode. (See Section 3.7.1.)



2) Select [Monitor] → [Device batch] from the menu. Or right click the circuit window and select [Device batch].



3) Input the first number of the devices to be monitored in the "Device batch monitor" window and then press the Enter key to begin monitoring.

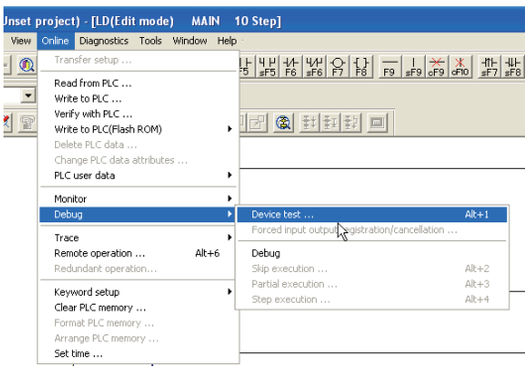


4) The value showing the device action and the ON/OFF status of the contacts and coils are displayed.

3.7.4 Device test

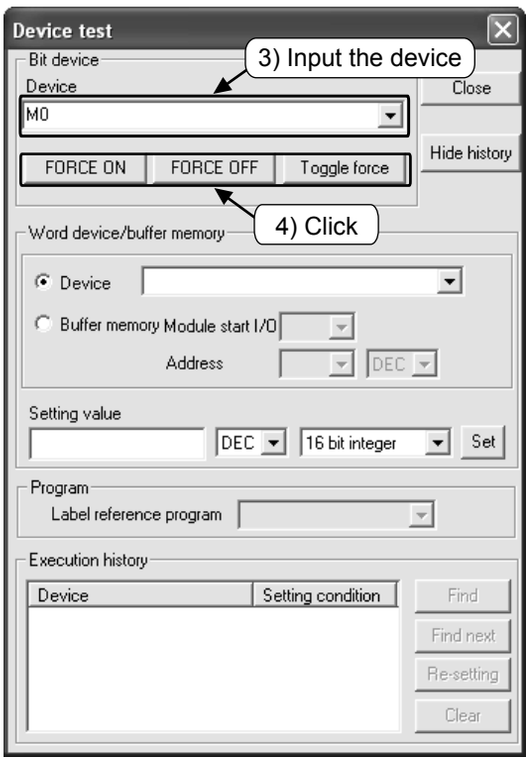
1) Forced ON/OFF

Using the device test screen, forcedly turn ON/OFF the bit devices of the PLC (M,Y,T,C and so on). (The forced ON/OFF function for X is not available.)
When the PLC is running, the forced ON/OFF function can turn on or off specific devices.



1) Set to monitor mode. (See Section 3.7.1.)

2) Select [Online] → [Debug] → [Device test] from the menu. Or right click the circuit window and select [Device test].



3) Input the device number to be forcedly turned on/off.

4) • [FORCE ON]: Turns on the device.
• [FORCE OFF]: Turns off the device.
• [Toggle force]: Switches the ON/OFF status of the device each time it is pressed.

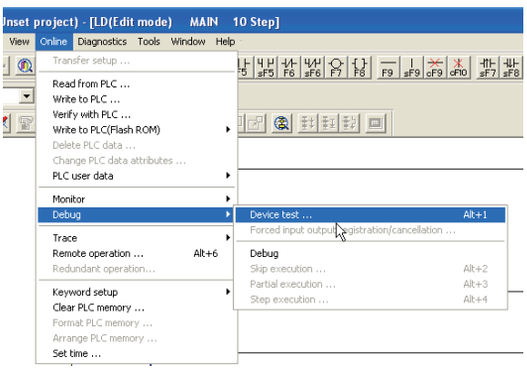
Reference

Forced ON/OFF (Ladder monitor window)

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

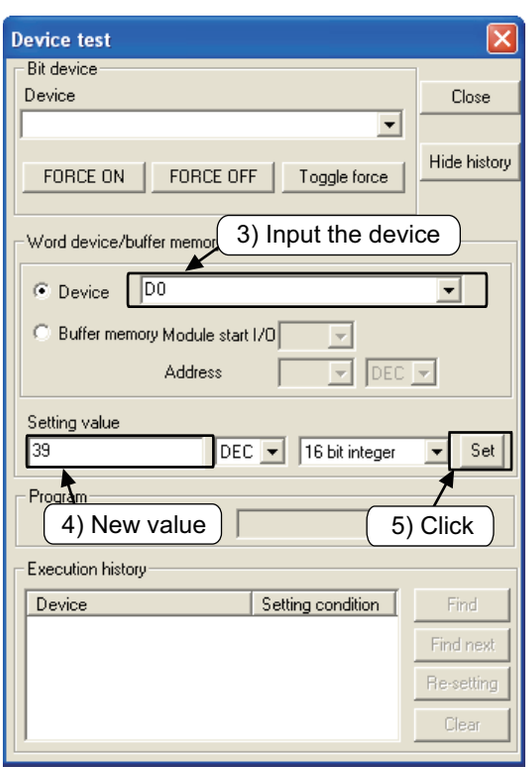
2) Changing the current value of a word device

Change the current specified value of the PLC's word device (T, C, D and so on).



1) Set to monitor mode. (See Section 3.7.1.)

2) Select [Online] → [Debug] → [Device test] from the menu. Or right click the circuit window and select [Device test].



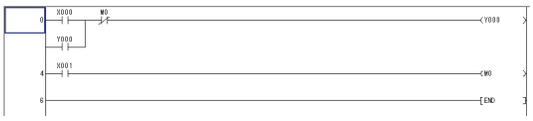
3) Input the device number to be changed.

4) Input a new value.

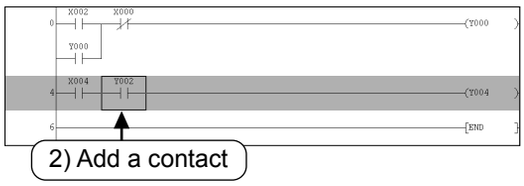
5) Click [Set].

3.7.5 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.



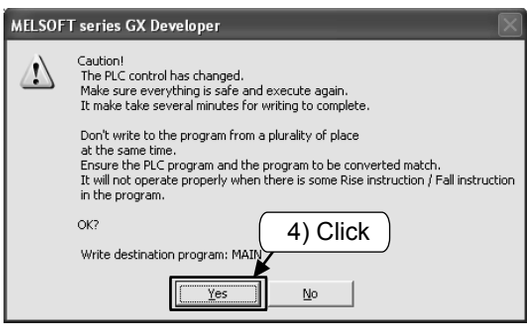
1) A contact will be added to the circuit on the left as an example. Within the circuit diagram view, set the mode to write mode ( ).



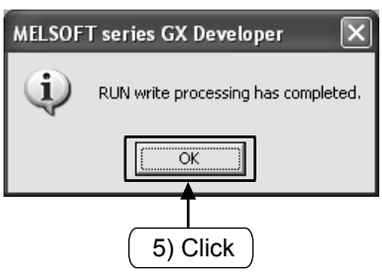
2) Add a contact.
The circuit block is displayed in grey.

Write during RUN Shift+F4

3) Press [Shift] and [F4] together, or select [Convert] → [Convert (online change)] from the menu.



4) Click [Yes] to confirm the warning message about PLC safety regarding online changes.



5) The message "RUN write processing has completed." is displayed. Click [OK].

Caution

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

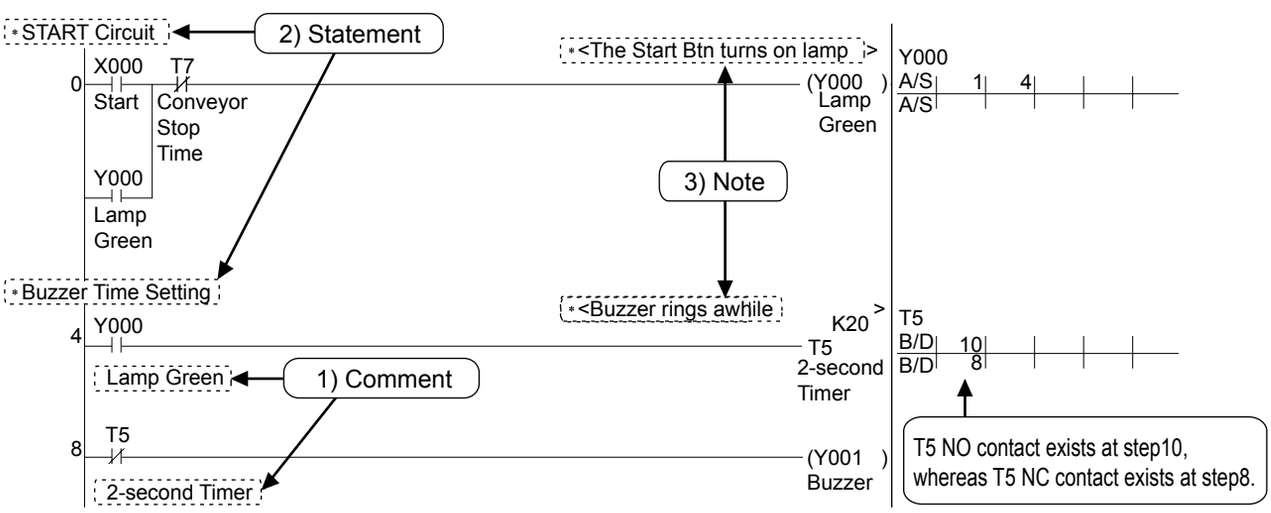
3.8 Inputting comments

3.8.1 Types of comments

The following 3 types of comments can be input.

Type	Purpose	The number of characters (full-width)	Remark
1) Device comment	A comment describing the role and function of each device	16	It is necessary to set the "Comments capacity" from the parameter setting when writing to the PLC. The "Comment range setting" must also be set.
2) Statement	A comment describing the role and function of circuit blocks	32	This is a comment (peripheral) on GX Developer's side. (It is not downloaded to the PLC)
3) Note	A comment describing the role and function of output instructions	16	This is a comment (peripheral) on GX Developer's side. (It is not downloaded to the PLC)

[Comment Examples]



Point

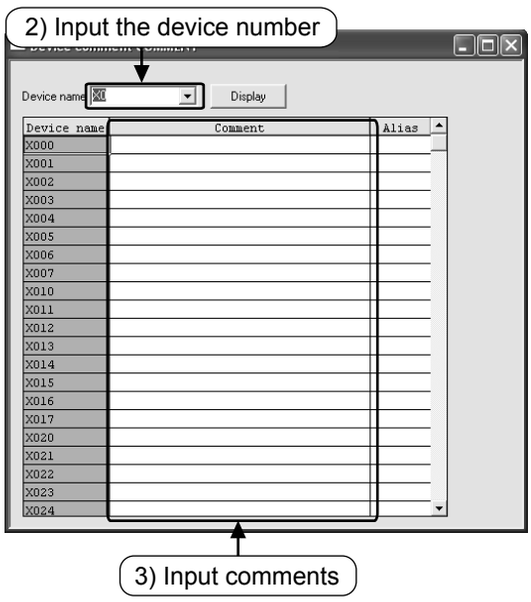
How to display comments

- Select [View] → [Comment] from the menu and then the comments are displayed.
- Repeat the operation above to stop displaying comments.

3.8.2 Operation for creating device comments

1) How to input device comments using a list

1) Double click [Device comment] → [COMMENT] in the project list.

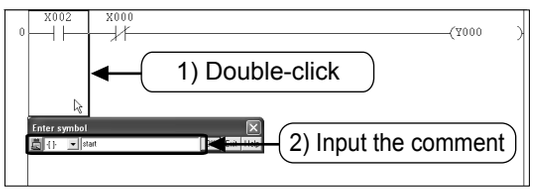


2) Input the start number of the devices which are to be commented in "Device name", and click [Display].

3) Input comments in the "Comment" column.

- When inputting comments for another device, input the device number again following step 2.

2) How to input device comments in the circuit diagram



1) Click from the tool bar and double-click the circuit diagram symbol to be commented.

2) Input the comment in the "Enter symbol" window and click [OK].

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

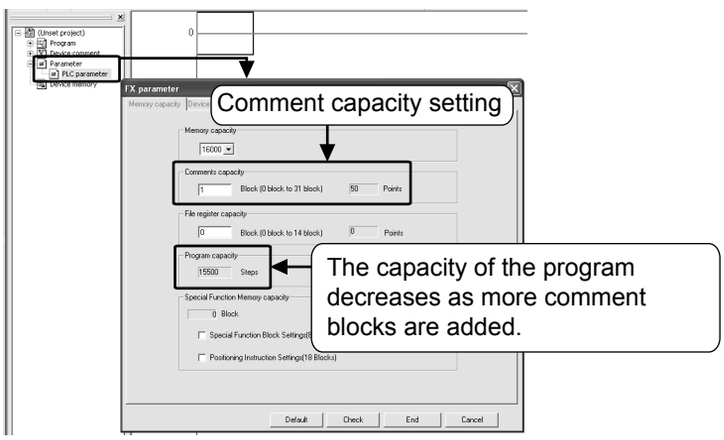
Point

Setting for writing the device comments to the PLC

It is necessary to set "Parameter setting" and "Comment range setting" in order to write the device comments to the PLC.

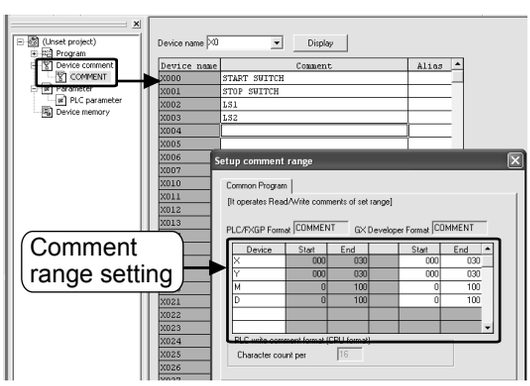
1) Parameter setting

- Double click [Parameter] → [PLC parameter] in the project list.
- Set the "number of blocks" in the "Comments capacity" setting.
  One block is equivalent to a 50-point comment, occupying the capacity of 500-steps of program memory.



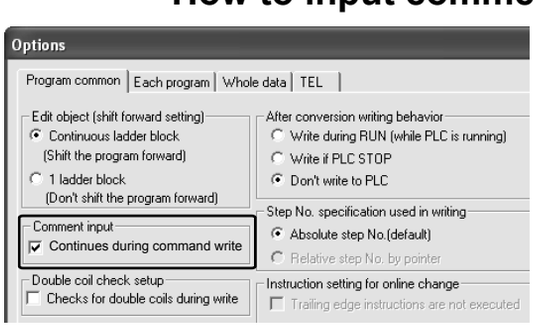
2) Comment range setting

- Double click [Device comment] → [Comment] in the project list to display the device comment list.
- Select [Edit] → [Setup comment range] from the menu.
- Set the type and the range of the devices to be written to the PLC in the Setup comment range dialog.



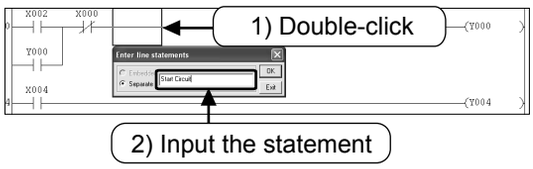
Reference

How to input comments when creating a circuit



Select [Tool] → [Options] from the menu, Check "Continues during command write" of the [Comment input] box in the [Program common] tab.
After configuring the setting above, the operation of circuit input continues and the "Enter symbol" window described in step 2) is displayed when the circuit is being created.

3.8.3 Operation for creating statements

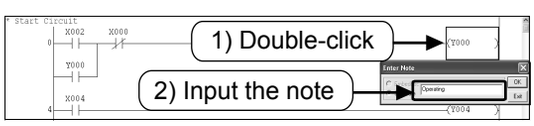


1) Click from the tool bar, and double-click anywhere on the circuit block where the statement is to be written.

2) Input the statement in the "Enter line statements" window and click [OK].

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

3.8.4 Operation for creating notes



1) Click from the tool bar, and double-click the output instruction symbol where the note is to be written.

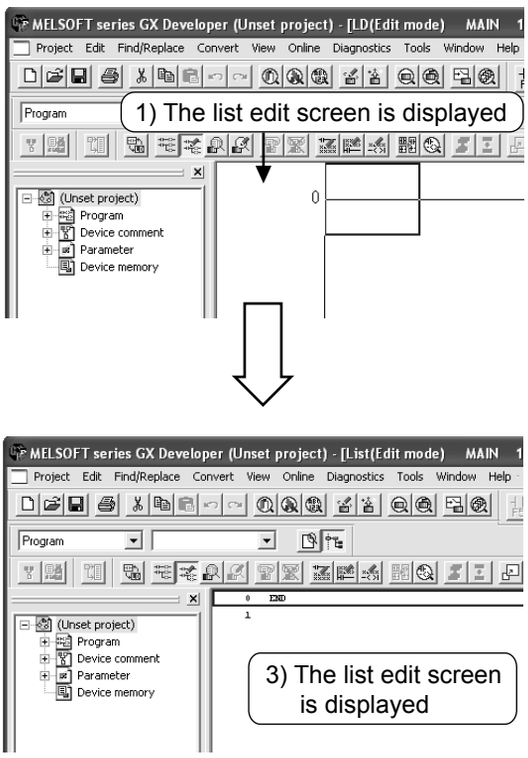
2) Input the note in the "Enter Note" window and click [OK].

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

3.9 Operation for creating instruction list

In GX Developer, a program can also be created by list logic.

3.9.1 Displaying the list edit screen



1) Create a new project (See Section 3.2.2.) or display the circuit of the existing project.

2) Select from the tool bar or select [View] → [Instruction list] (Alt+F1) from the menu.

3) The list edit screen is displayed.
Click on the tool bar again or select [View] → [Ladder] (Alt+F1) from the menu to return to the ladder display.

3.9.2 How to input instructions

Initial screen

```
0   END
1
```

After list input

```
    ;* Start Circuit
0   LD      X002
1   OR      Y000
2   ANI     X000
3   OUT     Y000
4   LD      X004
5   OUT     Y004
6   LDI     X002
7   AND     X000
8   OUT     Y000
9   LD      X003
10  OUT     M0
11  END
12
```

1) Input the instruction language from step 0 sequentially.
The step number is added automatically when each instruction is input. (For input procedures, refer to the next page.)

● How to input basic instructions and applied instructions

A "space" is input between the instruction language, device number and operand.

[Examples for basic instructions]

LD	X0	Enter	**Connection and OUT instructions**
OUT	Y0	Enter	
LDI	X0	Enter	
AND	Y0	Enter	
OUT	M0	Enter	
LD	M0	Enter	
OUT	T0 K10	Enter	**Coil instructions for the timer and counter**
OUT	C0 K5	Enter	

[Examples for applied instructions]

MOV	K1 D0	Enter
CMP	K20 D3 M10	Enter

Reference

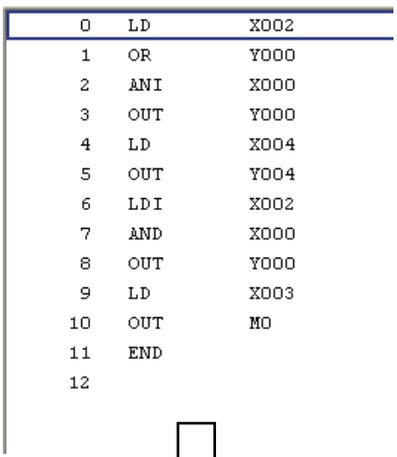
Key operation for inputting/editing

- "Ovrwrte "and" Insert" modes can be switched between by pressing the Insert key.
- An instruction can be deleted by using the Delete key.

[Insert line] and [Delete line] operations can be done by right clicking the mouse

3.9.3 Checking the content of the list input

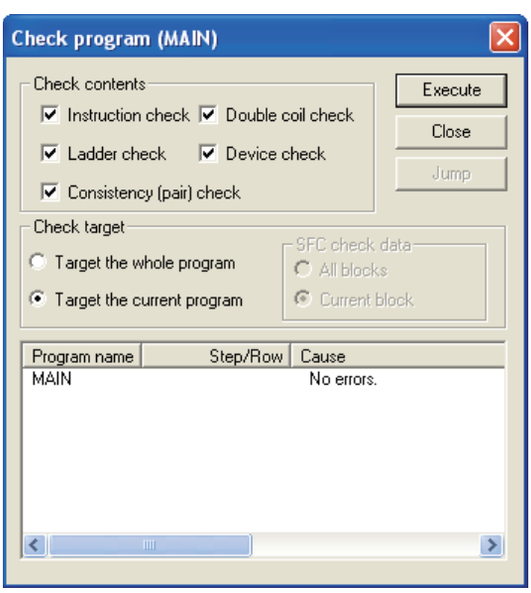
Confirm that there are no errors in the program created by list input in the circuit display.



0	LD	X002
1	OR	Y000
2	ANI	X000
3	OUT	Y000
4	LD	X004
5	OUT	Y004
6	LDI	X002
7	AND	X000
8	OUT	Y000
9	LD	X003
10	OUT	M0
11	END	
12		

1) Select from the tool bar, or select [View] → [Ladder] from the menu.

2) Check whether the circuit created by list input is displayed.



3) Select [Tools] → [Check program] to execute the program check to see if the logic has any errors.

Chapter 4

SUMMARY OF PLC BASIC INSTRUCTIONS

So far…

It has been described that the PLC is an aggregate of relays, timers and counters, and that it is sequentially controlled with imaginary internal wiring which is created through key operations on a programming panel.
It has also been described that according to the way contacts and coils are connected, and what types of coils are being used, rules, or instructions, are required for this imaginary wiring.

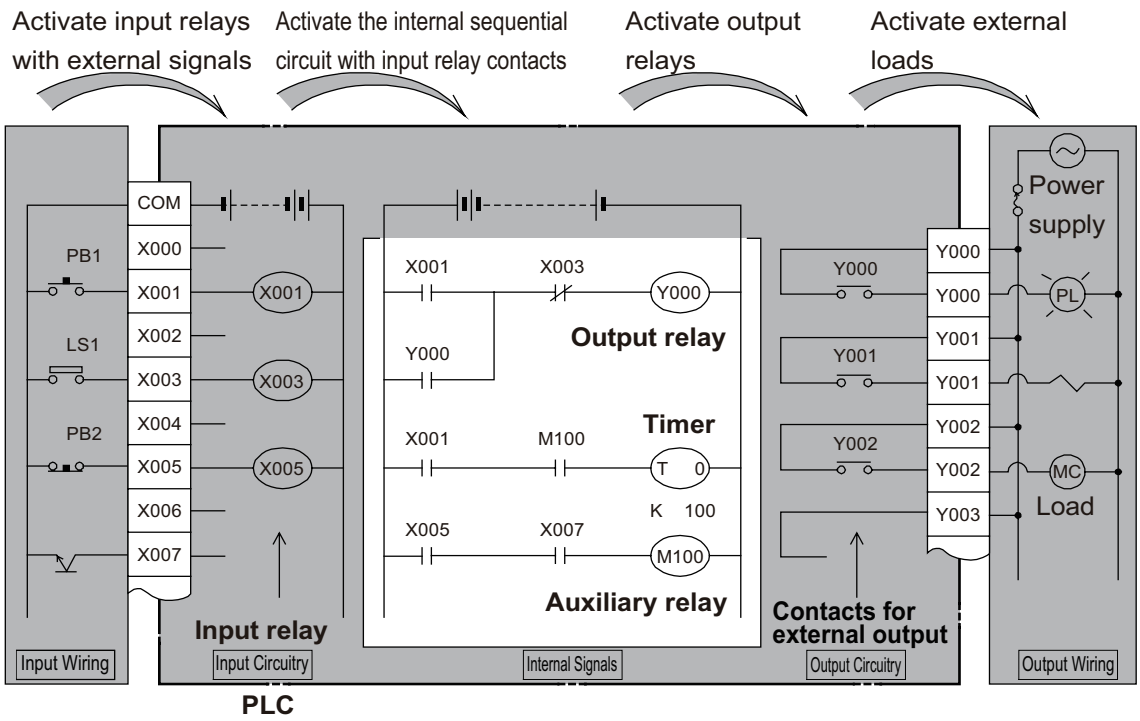
In the instructions…

Instructions can be divided into those functioning with element numbers and those functioning independently. Therefore, you should know the meanings of the device numbers as well.

In this chapter…

Basic 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



1) Types of PLC devices

Input relay (X) : X000 and onwards, 8 octal number

Auxiliary relay (M) M0 ~	Timer (T) T0 ~	Counter (C) C0 ~
Date register (D) D0 ~	Others ・State (S) ・Index register (V, Z) ・Pointer (P, I) and so on	High speed counter (C) C235 ~

Output relay (Y) : Y000 and onwards, 8 octal number

2) Device number ranges for the FX3U main devices

Device name	Content			
I/O relay				
Input relay	X000 to X367	248 points	The number of the input/output terminals are assigned in octal format.	
Output relay	Y000 to Y367	248 points		
Auxiliary relay				
General type	M0 to M499	500 points	These are the internal relays of the PLC.	
Latched (battery backed) type	M500 to M1023	524 points		
Latched (battery backed) type	M1024 to M7679	6656 points		
Special type	M8000 to M8511	512 points		
Timer (on-delay timer)				
100 ms	T0 to T191	192 points	0.1 to 3,276.7 seconds	The timers are used for clocking. The clocking range depends on the timer.
100 ms [for subroutine or interrupt routine]	T192 to T199	8 points	0.1 to 3,276.7 seconds	
10 ms	T200 to T245	46 points	0.01 to 327.67 seconds	
Retentive type for 1 ms	T246 to T249	4 points	0.001 to 32.767 seconds	
Retentive type for 100 ms	T250 to T255	6 points	0.1 to 3,276.7 seconds	
1 ms	T256 to T511	256 points	0.001 to 32.767 seconds	
Counter				
General type up-counter (16 bits)	C0 to C99	100 points	0 to 32,767 counts	Counters are used for counting. 32-bit counters can be switched between up/down.
Latched (battery backed) type up-counter (16 bits)	C100 to C199	100 points		
General type bi-directional counter (32 bits)	C200 to C219	20 points	– 2,147,483,648 to + 2,147,483,647 counts	
Latched (battery backed) type bi-directional counter (32 bits)	C220 to C234	15 points		
High speed counter				
1-phase 1-counting input Bi-directional (32 bits)	C235 to C245	Up to 8 points can be used in C235 to C255 [Latched (battery backed) type]-2,147,483,648 to 2,147,483,647 counts		These counters are used for counting fast signals from the PLC's input terminals.
1-phase 2-counting input Bi-directional (32 bits)	C246 to C250			
2-phase 2-counting input Bi-directional (32 bits)	C251 to C255			
Data register (32 bits when used in pair form)				
General type (16 bits)	D0 to D199	200 points	Registers for storing number data	
Latched (battery backed) type (16 bits)	D200 to D511	312 points		
Latched (battery backed) type (16 bits) <file register>	D512 to D7999 <D1000 to D7999>	7488 points <7000 points>		
Special type (16 bits)	D8000 to D8511	512 points		
Index type (16 bits)	V0 to V7, Z0 to Z7	16 points		
Pointer				
For jump and branch call	P0 to P4095	4096 points	Pointers for CJ instruction and CALL instruction	
Input interrupt Input delay interrupt	I0□□ to I5□□	6 points	Pointers for input interrupt and timer interrupt	
Timer interrupt	I6□□ to I8□□	3 points		
Counter interrupt	I010 to I060	6 points	Pointers for HSCS instruction	
Nesting				
For master control	N0 to N7	8 points	Nesting pointers for MC instruction	
Constant				
Decimal (K)	16 bits	– 32,768 to + 32,767		
	32 bits	– 2,147,483,648 to + 2,147,483,647		

4.2 Types of basic instructions

The following table lists the available basic instructions for FX3U PLC programming.

Mnemonic	Name	Symbol	Function	Applicable Devices*1
Contact Instruction				
LD	Load	Applicable devices	Initial logical operation contact type NO (normally open)	X,Y,M,S, D□.b,T,C
LDI	Load Inverse	Applicable devices	Initial logical operation contact type NC (normally closed)	X,Y,M,S, D□.b,T,C
LDP	Load Pulse	Applicable devices	Initial logical operation of Rising edge pulse	X,Y,M,S, D□.b,T,C
LDF	Load Falling Pulse	Applicable devices	Initial logical operation of Falling/trailing edge pulse	X,Y,M,S, D□.b,T,C
AND	AND	Applicable devices	Serial connection of NO (normally open) contacts	X,Y,M,S, D□.b,T,C
ANI	AND Inverse	Applicable devices	Serial connection of NC (normally closed) contacts	X,Y,M,S, D□.b,T,C
ANDP	AND Pulse	Applicable devices	Serial connection of Rising edge pulse	X,Y,M,S, D□.b,T,C
ANDF	AND Falling Pulse	Applicable devices	Serial connection of Falling/trailing edge pulse	X,Y,M,S, D□.b,T,C
OR	OR	Applicable devices	Parallel connection of NO (normally open) contacts	X,Y,M,S, D□.b,T,C
ORI	OR Inverse	Applicable devices	Parallel connection of NC (normally closed) contacts	X,Y,M,S, D□.b,T,C
ORP	OR Pulse	Applicable devices	Parallel connection of Rising edge pulse	X,Y,M,S, D□.b,T,C
ORF	OR Falling Pulse	Applicable devices	Parallel connection of Falling/trailing edge pulse	X,Y,M,S, D□.b,T,C
Connection Instruction				
ANB	AND Block		Serial connection of multiple parallel circuits	–
ORB	OR Block		Parallel connection of multiple contact circuits	–
MPS	Memory Point Store	MPS	Stores the current result of the internal PLC operations	–
MRD	Memory Read	MRD	Reads the current result of the internal PLC operations	
MPP	Memory POP	MPP	Pops (recalls and removes) the currently stored result	

Mnemonic	Name	Symbol	Function	Applicable Devices*1
Connection Instruction				
INV	Inverse	INV	Invert the current result of the internal PLC operations	–
MEP*2	MEP		Conversion of operation result to leading edge pulse	–
MEF*2	MEF		Conversion of operation result to trailing edge pulse	–
Out Instruction				
OUT	OUT	Applicable devices	Final logical operation type coil drive	Y,M,S,D□.b,T,C
SET	SET	SET Applicable devices	SET Bit device latch ON	Y,M,S,D□.b
RST	Reset	RST Applicable devices	RESET Bit device OFF	Y,M,S, D□.b,T, C,D,R,V,Z
PLS	Pulse	PLS Applicable devices	Rising edge pulse	Y,M
PLF	Pulse Falling	PLF Applicable devices	Falling/trailing edge pulse	Y,M
Master Control Instruction				
MC	Master Control	MC N Applicable devices	Denotes the start of a master control block	Y,M
MCR	Master Control Reset	MCR N	Denotes the end of a master control block	–
Other Instruction				
NOP	No Operation	–	No operation or null step	–
End Instruction				
END	END	END	Program END, I/O refresh and Return to Step 0	–

*1: "D□.b" and "R" are available only in FX_{3U} and FX_{3UC} PLCs

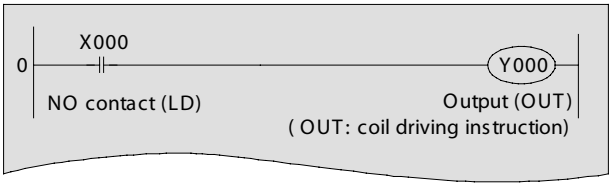
*2: Available only in FX_{3U} and FX_{3UC} PLCs

4.3 Let's master basic instructions

4.3.1 Contact instruction and out instruction

1) [Program of NO contact] Normally open instructions

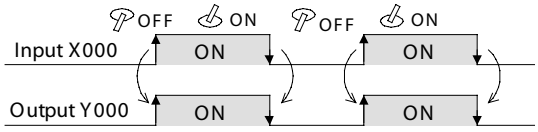
Ladder display



List display (reference)

Step	Instruction	
0	LD	X000
1	OUT	Y000

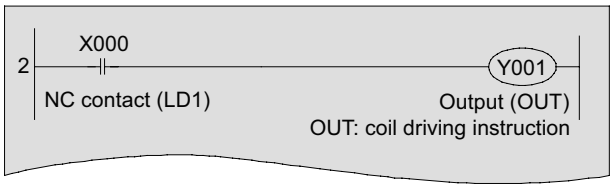
《Action》



- If the input condition X000 is "ON", Y000 is "ON".
- If X000 is "OFF", Y000 is also "OFF".

2) [Program of NC contact] Normally closed instructions

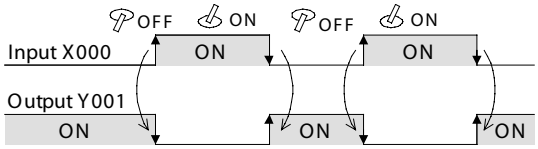
Ladder display



List display (reference)

Step	Instruction	
2	LDI	X000
3	OUT	Y001

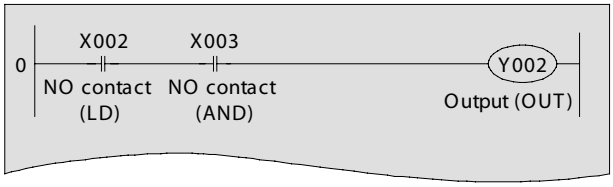
《Action》



- If the input condition X000 is "OFF", Y001 is "ON".
- If X000 is "ON", Y001 is "OFF".

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

Ladder display

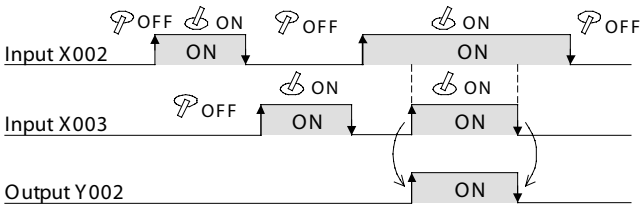


List display (reference)

Step	Instruction	
0	LD	X002
1	AND	X003
2	OUT	Y002

4

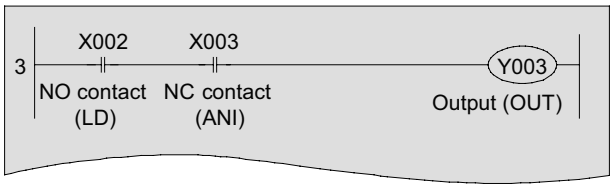
《Action》



- If the input condition X002 and X003 are both "ON", Y002 is "ON".
- If X002 or X003 is "OFF", Y002 is also "OFF".

[Program of a serial circuit (2)]

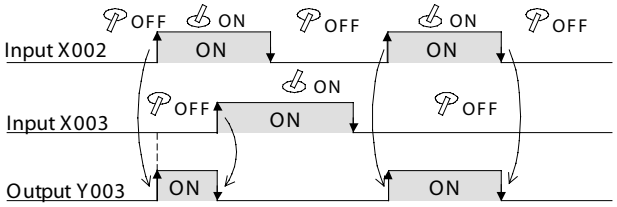
Ladder display



List display (reference)

Step	Instruction	
3	LD	X002
4	ANI	X003
5	OUT	Y003

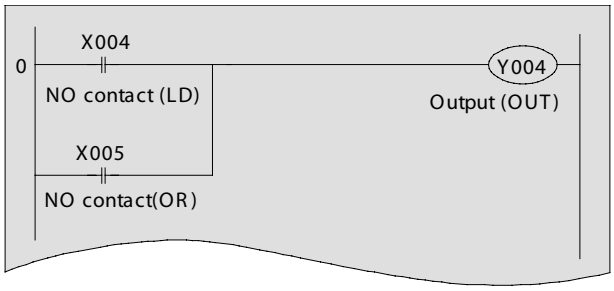
《Action》



- If the input condition X002 is "ON" and X003 is "OFF", Y003 is "ON".
- If X002 is "OFF" or X003 is "ON", Y003 is "OFF".

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

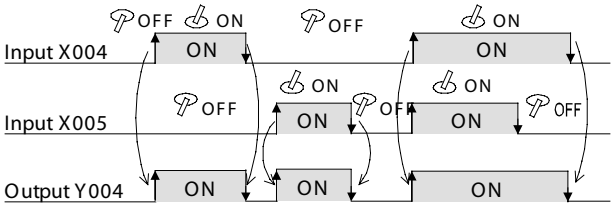
Ladder display



List display (reference)

Step	Instruction	
0	LD	X004
1	OR	X005
2	OUT	Y004

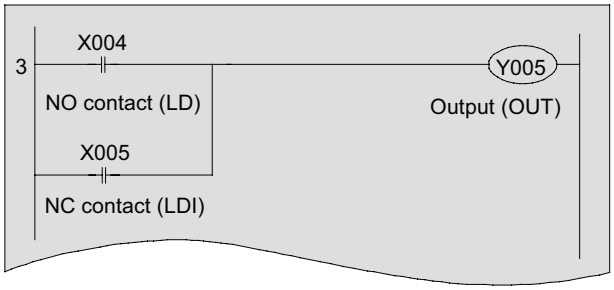
《Action》



- If the input condition X004 or X005 is "ON", Y004 is "ON".
- If X004 and X005 are both "OFF", Y004 is "OFF".

[Program of a parallel circuit (2)]

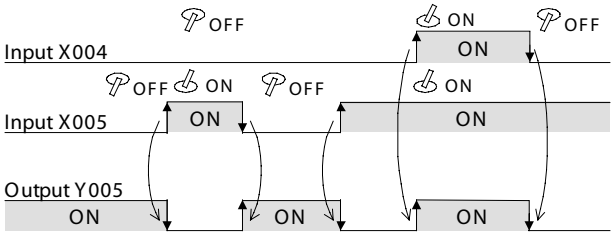
Circuit display



List display (reference)

Step	Instruction	
3	LD	X004
4	ORI	X005
5	OUT	Y005

《Action》

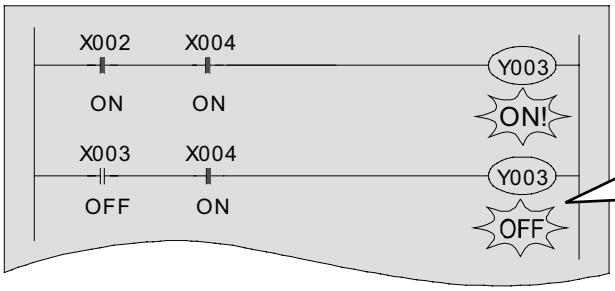


- If the input condition X004 is "ON" or X005 is "OFF", Y005 is "ON".
- If X004 is "OFF" and X005 is "ON", Y005 is "OFF".

Important point

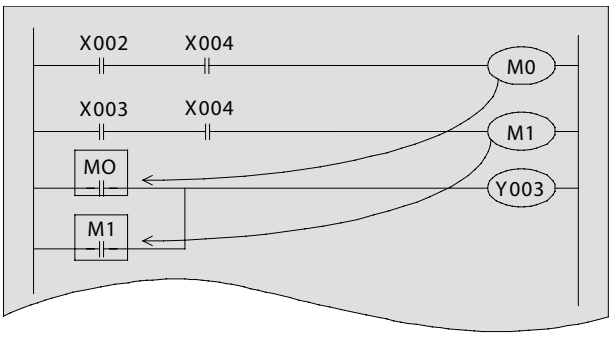
Avoid redundant outputs (double coils)

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



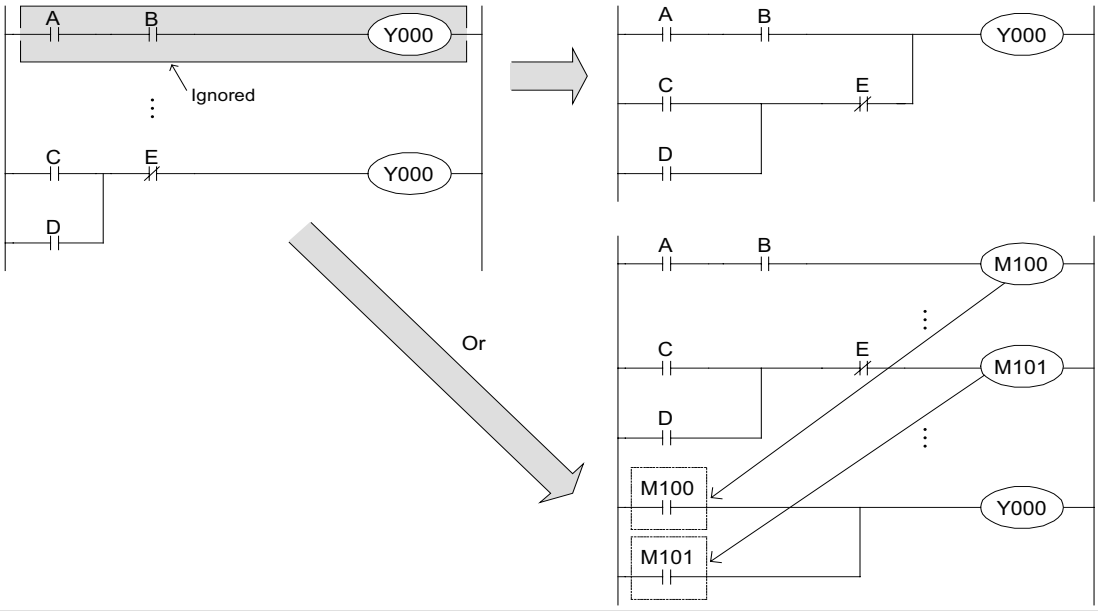
The output result is "OFF" which is not desired.

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.

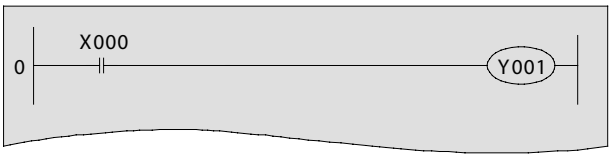
Another changing example



4.3.2 Difference between OUT instruction and SET/RST

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

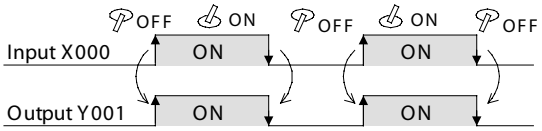
Ladder display



List display (reference)

Step	Instruction	
0	LD	X000
1	OUT	Y001

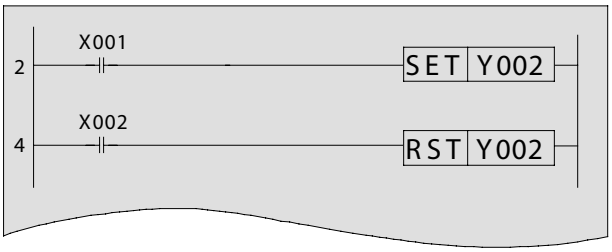
《Action》



- 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)

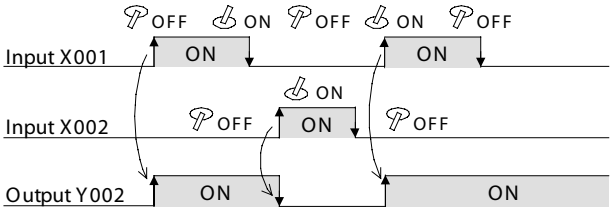
Ladder display



List display (reference)

Step	Instruction	
2	LD	X001
3	SET	Y002
4	LD	X002
5	RST	Y002

《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 OUT T □ instruction: Clocking of timers

Timers count with clock pulses of 1 ms, 10 ms, 100 ms and so on. When they reach their set valve, 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).

Timer (T) numbers for the FX3U PLC (Numbers are assigned in decimal format)

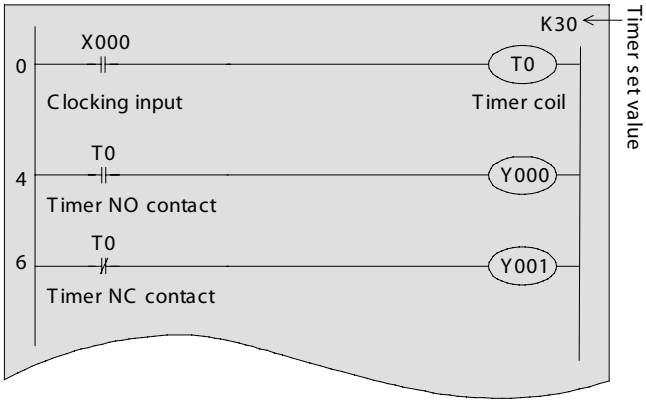
100 ms type 0.1 to 3276.7 seconds	10 ms type 0.01 to 327.67 seconds	Retentive type for 1 ms *1 0.001 to 32.767 seconds	Retentive type for 100 ms *1 0.1 to 3276.7 seconds	1 ms type 0.001 to 32.767 seconds
T0 to T199 200 points For routine program T192 to T199	T200 to T245 46 points	T246 to T249 4 points Interrupt execution Latched (battery backed) *1	T250 to T255 6 points Latched (battery backed) *1	T256 to T511 256 points

*1. The timer for the retentive type is latched by the battery when the power is off.

4

1) General timer

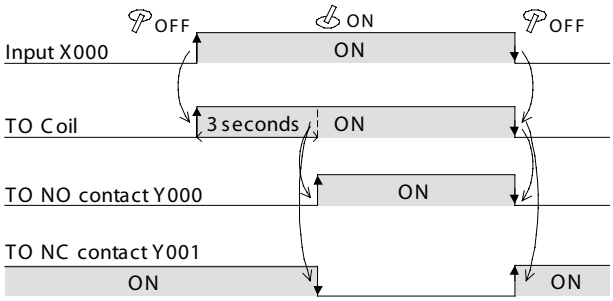
Ladder display



List display (reference)

Step	Instruction	
0	LD	X000
1	OUT	T0 K30
4	LD	T0
5	OUT	Y000
6	LDI	T0
7	OUT	Y001

《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 X000 is "OFF", the clocking of the timer is reset and the contact T0 also turns off.

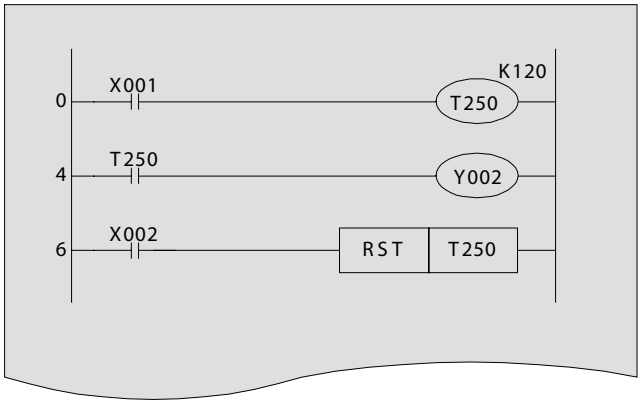
Reference

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

2) Retentive timer

T246 to T249 (4 points) are timers based on 1 ms, and T250 to T255 (6 points) are timers based on 100 ms.

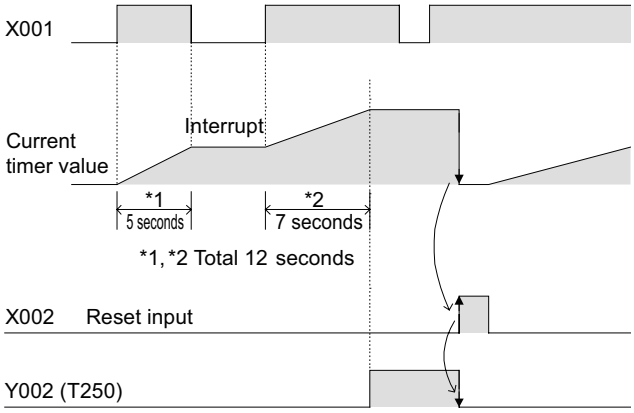
Ladder display



List display (reference)

Step	Instruction		
0	LD	X001	
1	OUT	T250	K120
4	LD	T250	
5	OUT	Y002	
6	LD	X002	
7	RST	T250	

《Action》



- The timer operates only when the clock input X001 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 X001 reaches the predetermined value.
- If the reset input X002 is "ON", the current value of the timer will become 0 and the output contact will also be "OFF".

Reference

Types of counters

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 timers based on 100 ms, and 0.12 seconds for timers based on 1 ms.
Latched (battery-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 OUT C □ instruction: Counting of counters

The types of counters include 16-bit counters, 32-bit counters, and high speed counters. In this section, the 16-bit up-counter is described.

Counter numbers for the FX3U PLC (Numbers are assigned in decimal format)

16-bit up-counter 0 to 32767		32-bit up/down counter – 2,147,483,648 to + 2,147,483,647	
General type	Latched (battery-backed)	General type	Latched (battery-backed)
C0 to C99 100 points*1	C100 to C199 100 points*2	C200 to C219 20 points*1	C220 to C234 15 points*2

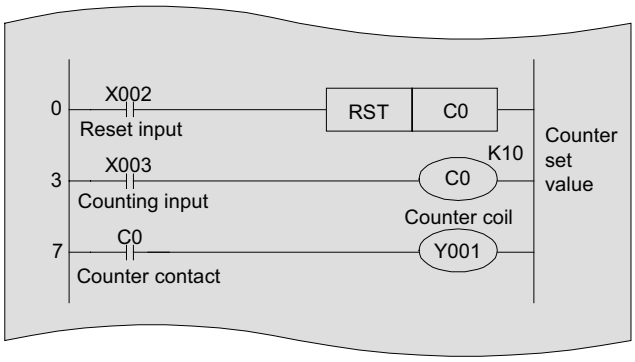
*1. Non-latched area. It can be changed to a latched (battery-backed) area by setting the parameters.
*2. Latched (battery-backed) area. It can be changed to a non-latched (non-battery-backed) area by setting the parameters.

Features of the counter

The following table lists the features of the 16-bit and 32-bit counters. They can be used according to the counting direction, count range, etc.

Item	16-bit counter	32-bit counter
Counting direction	Up-counting	Available to switch between up and down (C200: M8200 to C234: M8234).
Current value register	16 bits	32 bits
Set value	1 to 32767	–2,147,483,648 to +2,147,483,647
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	Changes after counting up (ring counter).
Output contact	Latched after counting up	Latched by up, reset by down.
Reset operation	The current value of the counter will become 0 and the output contact will be restored when the RST instruction is executed.	

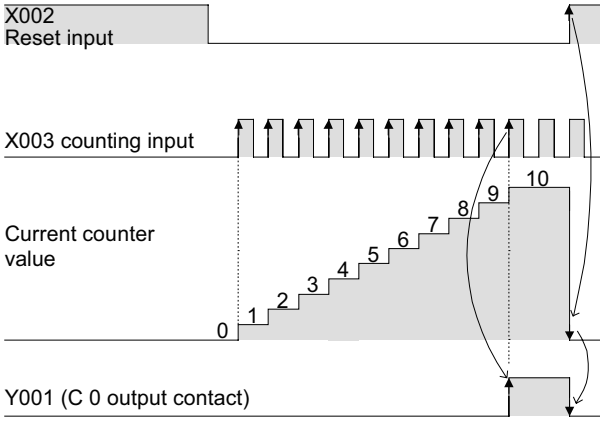
Ladder display



List display (reference)

Step	Instruction
0	LD X002
1	RST C0
3	LD X003
4	OUT C0 K10
7	LD C0
8	OUT Y001

《Action》



- The current value of counter increments every time the count input relay changes from OFF to ON, and when the value reaches the predetermined value, the output contact is closed.
- After reaching the predetermined value, the current value and the output contact keep their status.
- At the moment the reset input relay X002 is closed, the current value of the counter becomes 0 and the output contact is opened.
- 16-bit up counters are assigned from C0 to C199, and the timers C100 to C199 are backed up with the battery to maintain their current value when the power fails.
  The battery-backed counters continue to count up from their stored value when the power is restored.

Reference

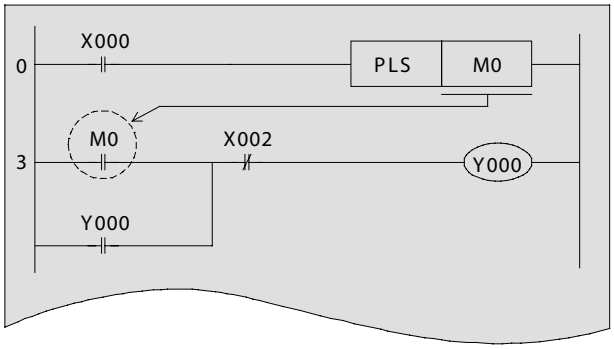
High speed inputs can be counted with high speed counters

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

[PLS instruction] Pulse (Rising edge pulse output)

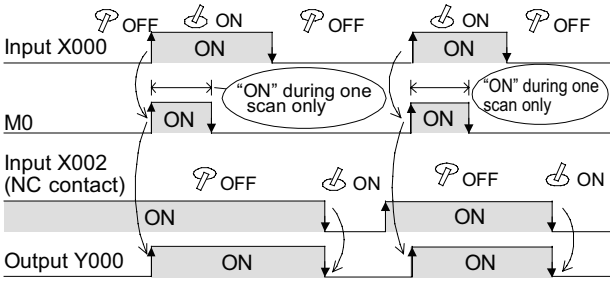
Ladder display



List display (reference)

Step	Instruction	
0	LD	X000
1	PLS	M0
3	LD	M0
4	OR	Y000
5	ANI	X002
6	OUT	Y000

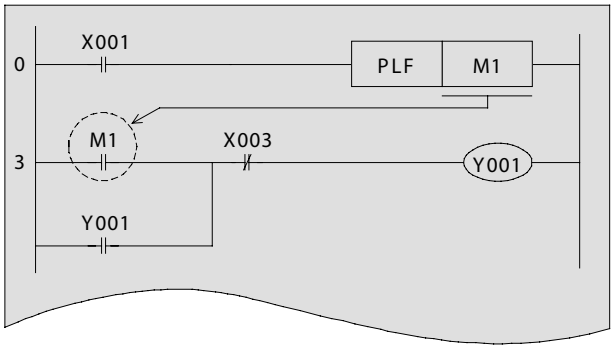
《Action》



● If the input condition X000 is "ON" and remains on, the specified device will be turned on for one scan (one operation cycle) only.

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

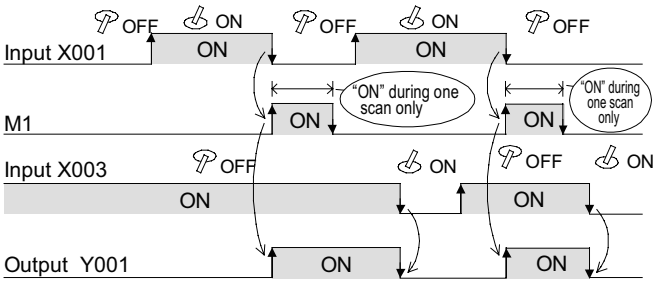
Ladder display



List display (reference)

Step	Instruction	
0	LD	X001
1	PLF	M1
3	LD	M1
4	OR	Y001
5	ANI	X003
6	OUT	Y001

《Action》



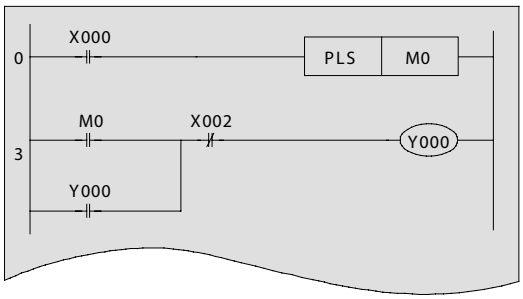
● If the input condition X001 is turned on and then off, the specified device will be turned on during one scan only.

Reference

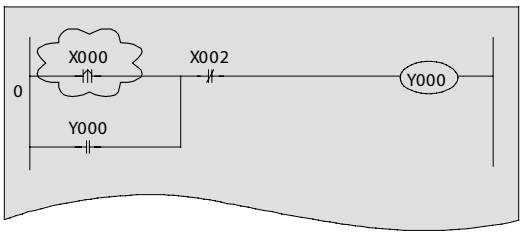
Simplification by using the rising/falling edge pulse contact instructions

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.

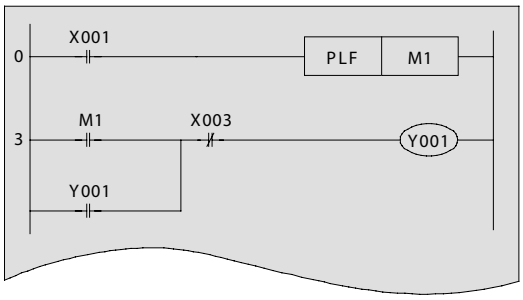
When the PLS instruction is used



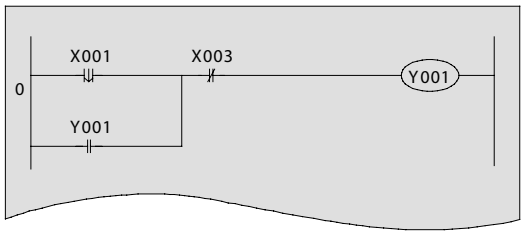
When the rising edge pulse contact instruction is used



When the PLF instruction is used



When the falling edge pulse contact instruction is used

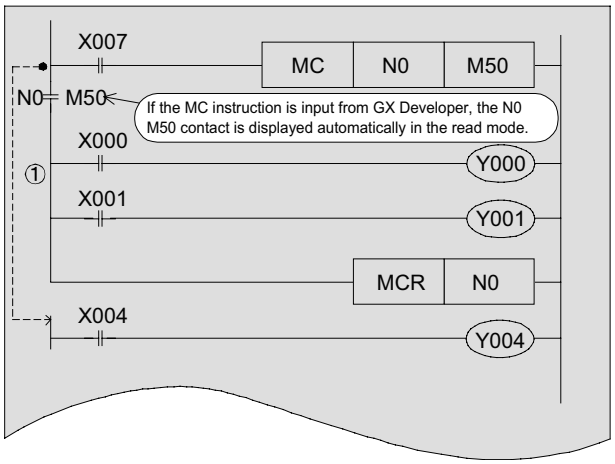


4.3.6 MC/MCR instruction

[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)

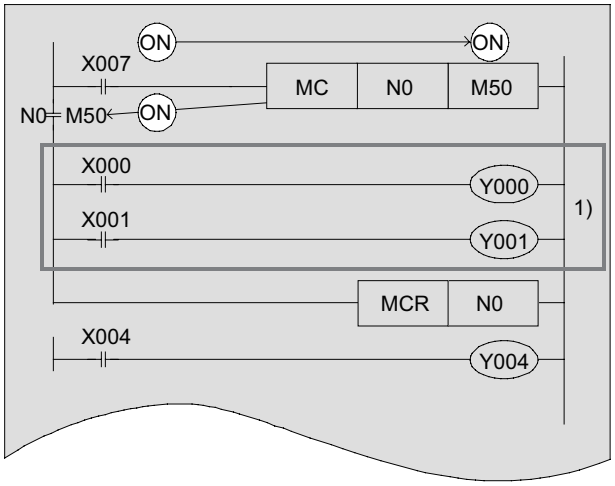
Circuit display



List display (reference)

Step	Instruction	
0	LD	X007
1	MC	N0 M50
4	LD	X000
5	OUT	Y000
6	LD	X001
7	OUT	Y001
8	MCR	N0
10	LD	X004
11	OUT	Y004

《Action》



- While the input condition X007 is "ON", the circuit indicated by 1) becomes valid, [Y000 is "ON" if X000 is "ON"], and [Y001 is "ON" if X001 is "ON"].
- When X007 is "OFF", Y000/Y001 does not operate.
- Since Y004 is not subject to MC/MCR, it turns on and off independently according to the operation of X004.

Point

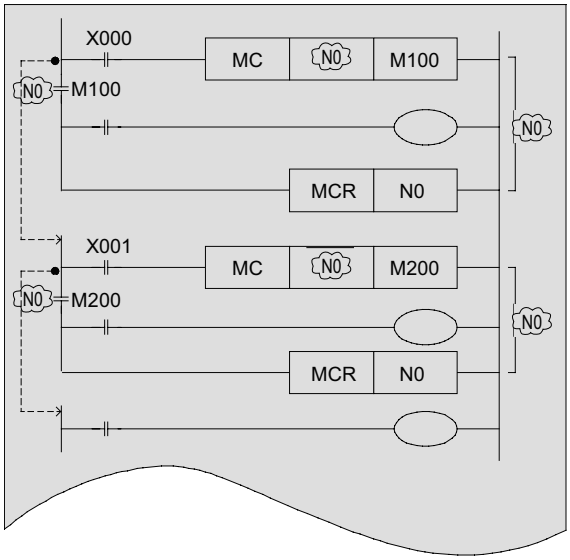
Status of MC block devices when MC does not operate

- 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.

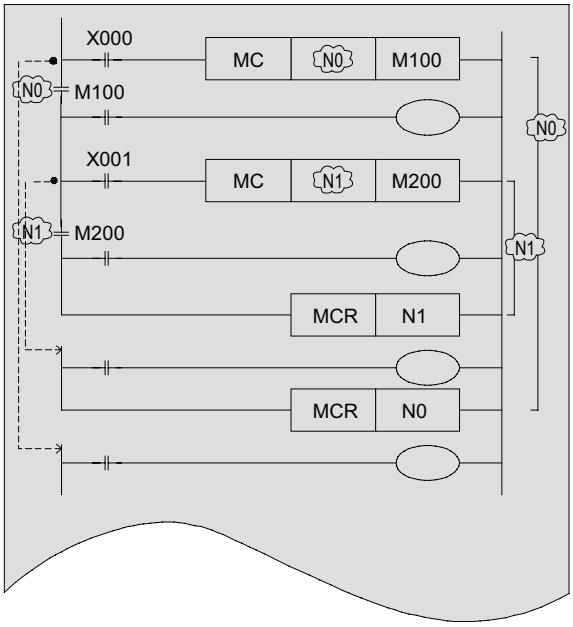
Reference

Nesting with MC/MCR

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

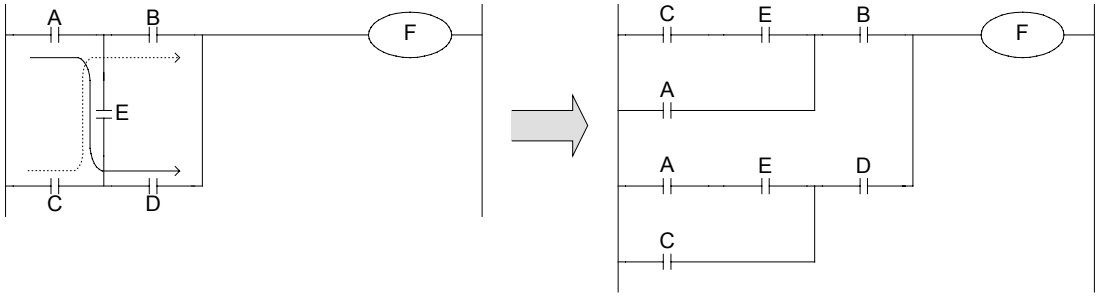


[Nest structured]
Nesting numbers N0 to N7 are sequentially used from the small number to big one to program.
(Max. 8 layers)



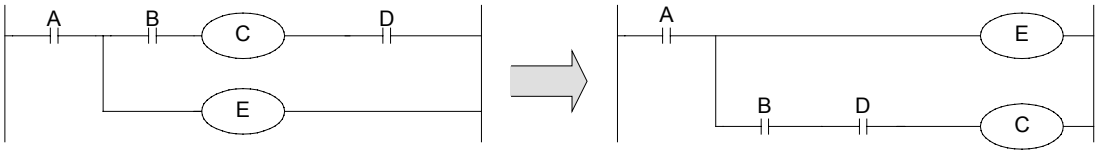
4.3.7 Non-programmable circuits and solutions

1) Bridge Circuit



A circuit in which current flows in both directions must be rewritten as shown above. (The left and right circuits are electrically identical.)

2) Location of Coil



● Contacts cannot be located on the right side of coils.

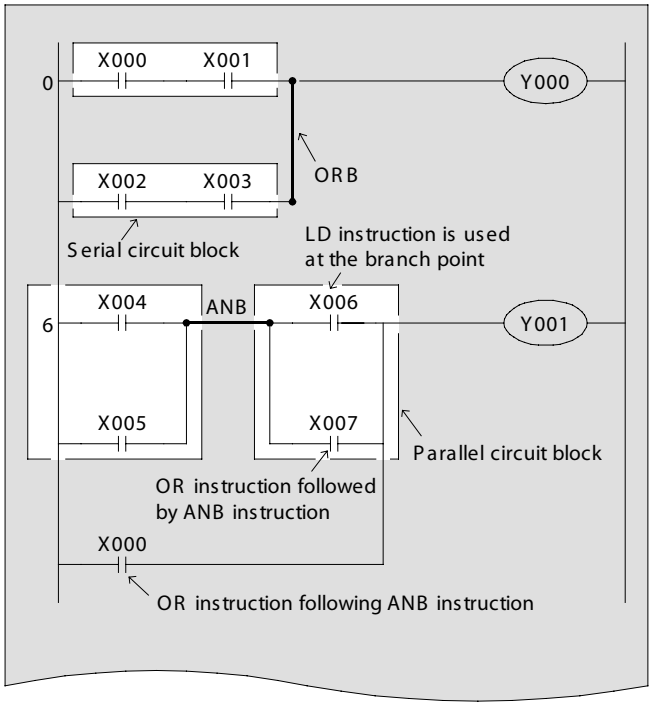
● It is recommended that coils internally used between contacts be programmed prior to the output.

4.3.8 Additional information for list programming (reference)

This section describes the basic instructions necessary when a list program is executed with FX-20P, etc.

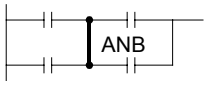
1) [ORB instruction] OR Block (Instruction to connect serial circuit blocks in parallel) [ANB instruction] AND Block (Instruction to connect parallel circuit blocks in series)

Use the OR (or ORI) instruction commands to connect contacts to the previous LD (or LDI) contacts. However, if OR (or ORI) is used subsequent to an ANB instruction, the contact is connected not to the previous LD (or LDI) contact, but to the one before that.

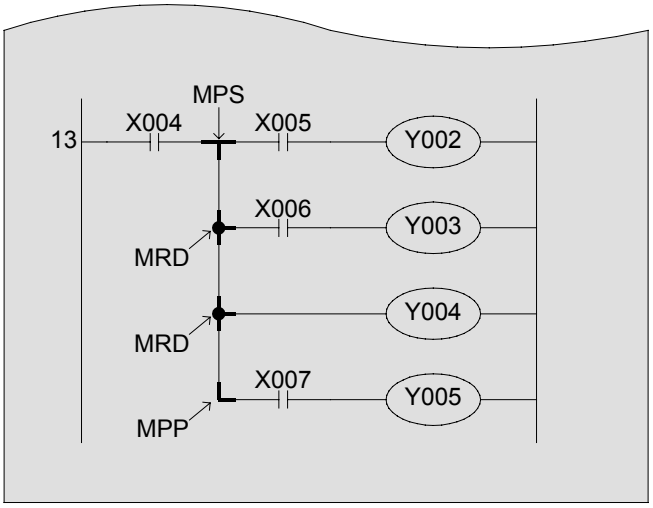


Step	Instruction	
0	LD	X000
1	AND	X001
2	LD	X002
3	AND	X003
4	ORB	
5	OUT	Y000
6	LD	X004
7	OR	X005
8	LD	X006
9	OR	X007
10	ANB	
11	OR	X000
12	OUT	Y001

The ANB instruction is displayed in the circuit as shown by the connection on the right.



2) [MPS instruction] Memory push (Instruction to store the intermediate operation result)
[MRD instruction] Memory read (Instruction to read the intermediate operation result)
[MPP instruction] Memory pop (Instruction to read and reset the intermediate operation result)



Step	Instruction	
13	LD	X004
14	MPS	
15	AND	X005
16	OUT	Y002
17	MRD	
18	AND	X006
19	OUT	Y003
20	MRD	
21	OUT	Y004
22	MPP	
23	AND	X007
24	OUT	Y005
25	END	

《Description》

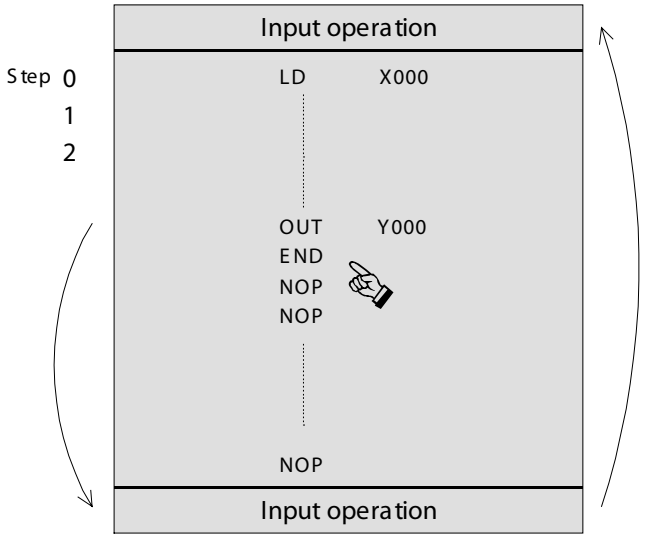
- These instructions are convenient to program a circuit including multiple branches as shown in the figure above. The MPS instruction stores the intermediate operation result, then drives the output relay Y002. The MRD instruction reads the stored memory, and then drives the output relay Y003.
- Although the MRD instruction is designed to be used repeatedly in a single program without numerical limitation, its number must be limited to a certain range for the diagram to be properly printed by the printer or displayed on the graphic programming panel. (The number of parallel outputs in a single circuit is limited to 24 lines or less.)
- For the last output line, instead of using an MRD instruction, an MPP instruction is used. This allows the memory stored in the previous step to be read and reset.

3) [NOP instruction] No operation (Instruction to perform no operation)

When the whole program is deleted, all instructions become NOPs.
When an NOP instruction exists between general instructions, the PLC neglects the NOP instruction to operate.
Since each NOP instruction requires one additional program step, they should be deleted as much as possible.

4) [END instruction] END (Program END, I/O refresh and Return to Step 0]

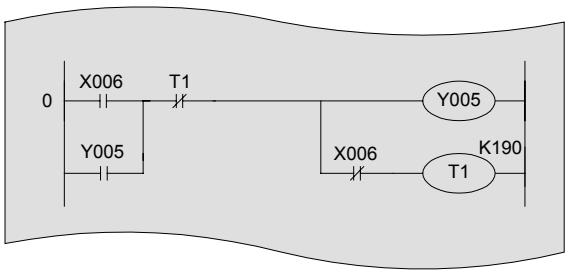
The PLC is designed to perform cycles of input operation, program execution, and output operation, again and again. If an END instruction is described at the virtual end of the program, the PLC omits the rest of the steps and directly executes the output operation.
END instructions are useful when you attempt to do a trial run. END instructions inserted at the end of each sequential block allow you to check the operation of each block and to gradually expand the area check .
In this case, do not forget to delete each applicable END instruction after checking the completeness of each circuit block.



4.4 Circuit examples with basic instructions

1) Off delay timer

Ladder display

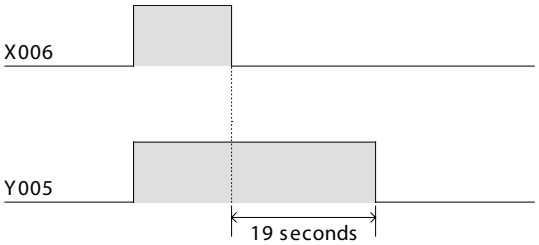


List display (reference)

Step	Instruction
0	LD X006
1	OR Y005
2	ANI T1
3	OUT Y005
4	ANI X006
5	OUT T1 K190

4

《Action》

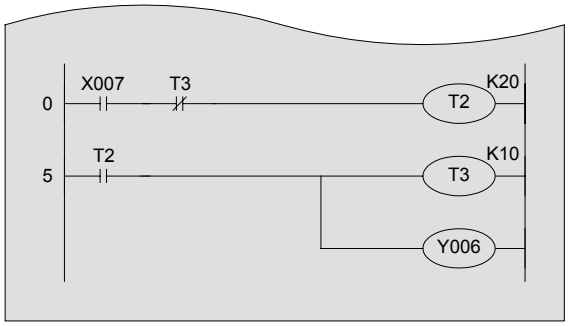


Y005 will turn OFF 19 seconds after X006 turns OFF.

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)

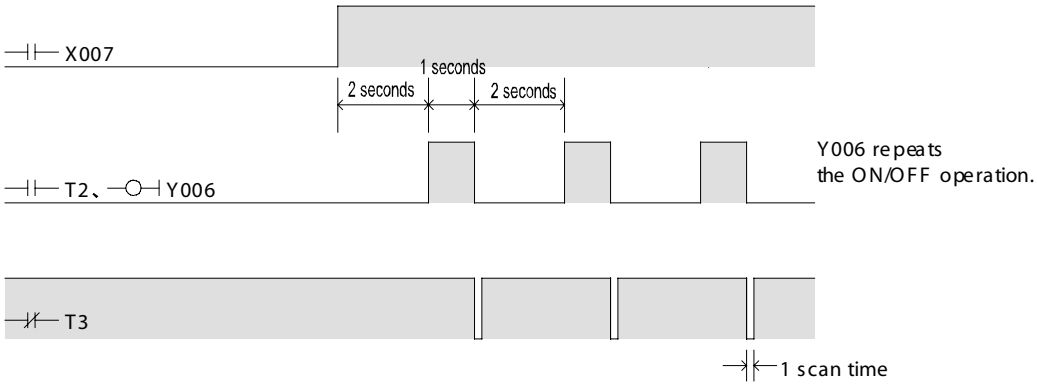
Ladder display



List display (reference)

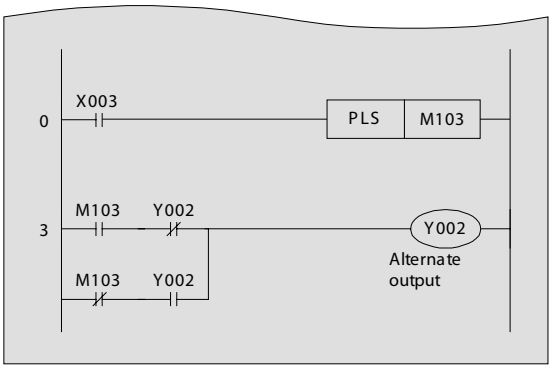
Step	Instruction
0	LD X007
1	ANI T3
2	OUT T2 K20
5	LD T2
6	OUT T3 K10
9	OUT Y006

《Action》



3) Alternate Circuit with pulse output circuit (Alternate action)

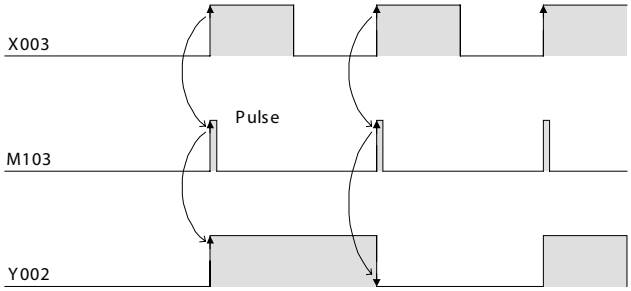
Ladder display



List display (reference)

Step	Instruction	
0	LD	X003
1	PLS	M103
3	LD	M103
4	ANI	Y002
5	LDI	M103
6	AND	Y002
7	ORB	
8	OUT	Y002

《Action》



● Once X003 is ON, Y002 is ON. If X003 is ON again, Y002 is, in turn, OFF. (Alternate action)

Chapter 5

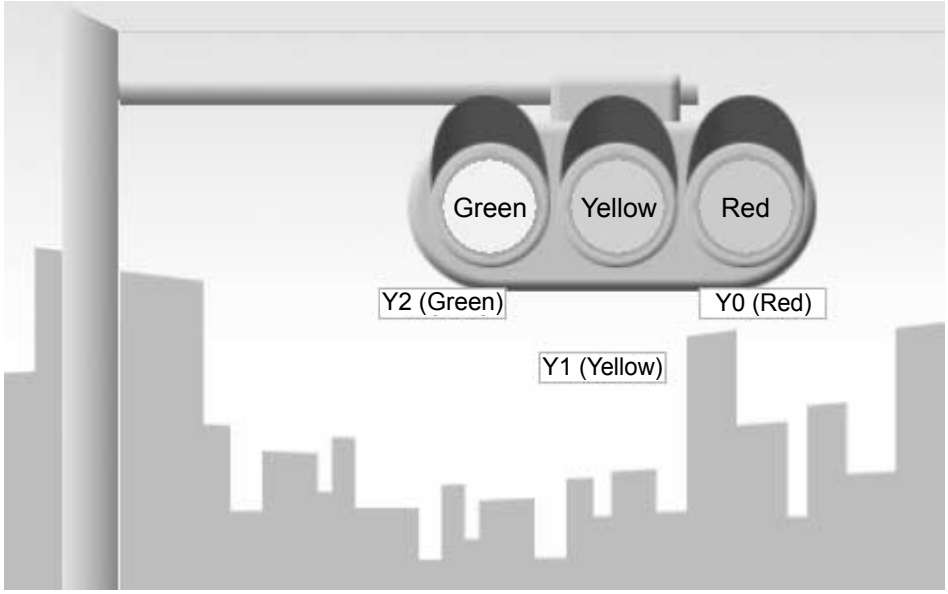
INTRODUCTION EXAMPLES AND PROGRAM OPERATION

Let's practice···

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

In this chapter, you can practice programming with the universal simulation unit, the FX3U-32MT-SIM. Use a personal computer as the programming tool.
For the operation of the personal computer software (GX Developer), see Chapter 3.

5.1 Introduction example《1》[Traffic light control]



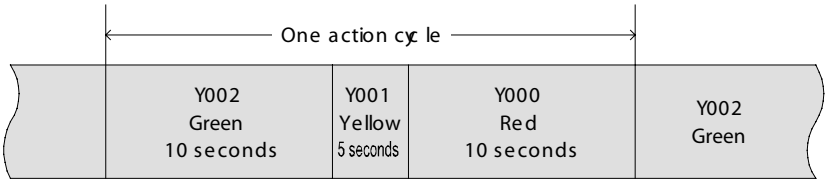
《Assignment of I/O》

Input	
X000	Control start

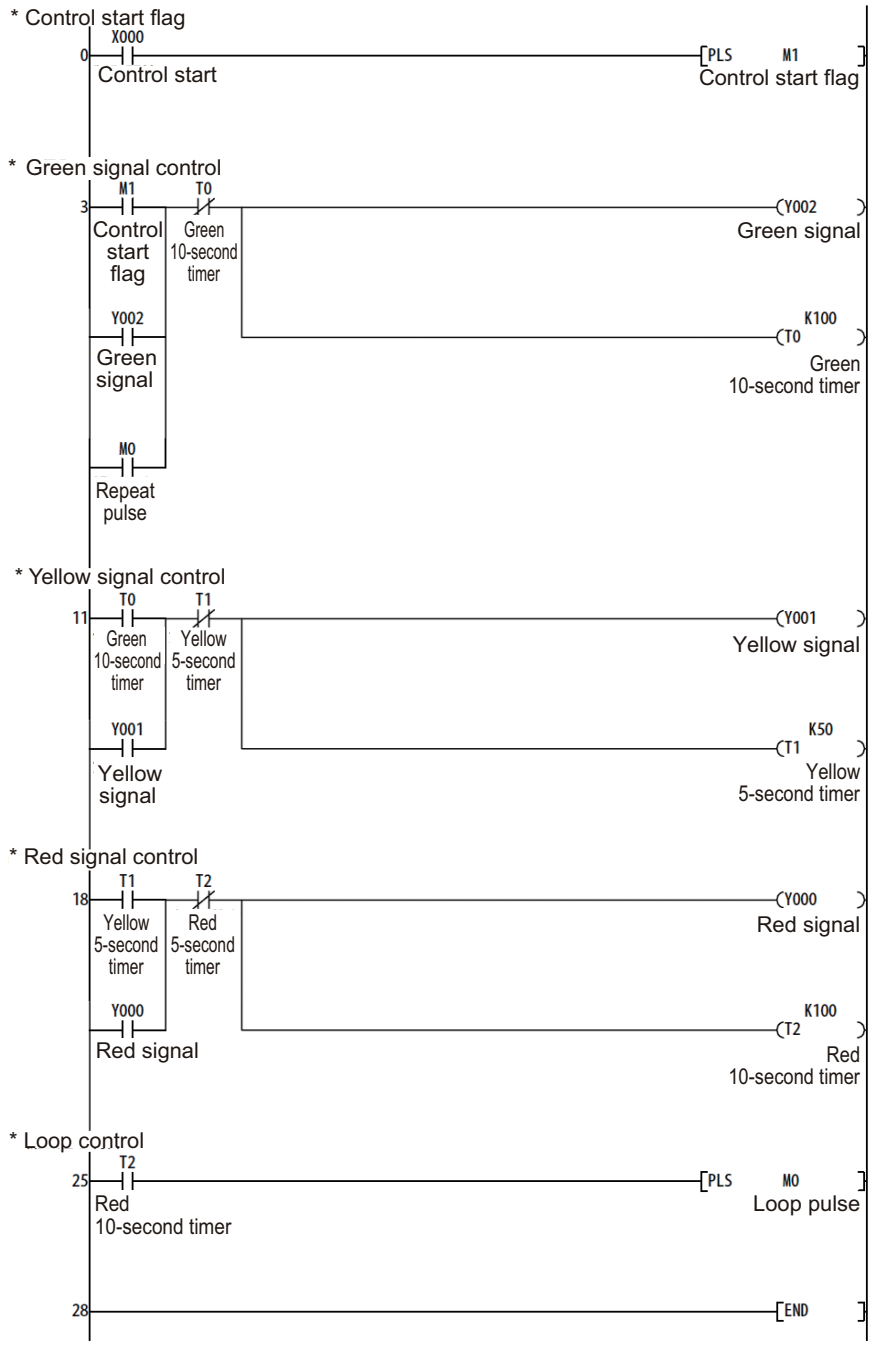
Output	
Y000	Red signal
Y001	Yellow signal
Y002	Green signal

《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.



《Example of sequence circuit with comments》

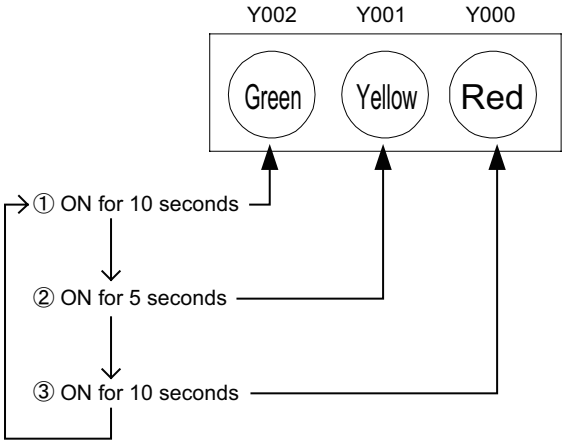


- If X000 turns ON, M1 will turn ON for one scan cycle.
- If M1 turns ON for one scan cycle, Y002 (Green signal) will turn ON for 10 seconds.
- If Y002 (Green signal) turns OFF, Y001 (Yellow signal) will turn ON for 5 seconds.
- If Y001 (Yellow signal) turns OFF, Y000 (Red signal) will turn ON for 10 seconds.
- If Y000 (Red signal) turns OFF, the program is repeated from the control of the green signal.

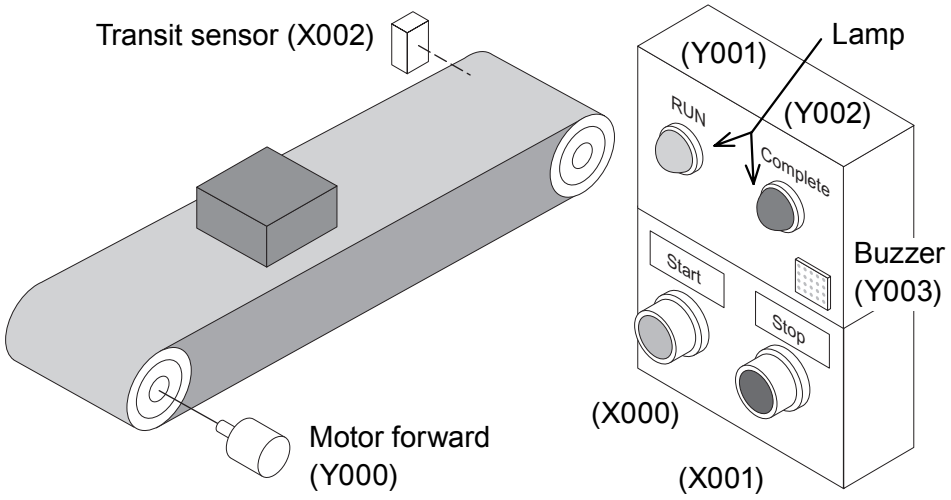
《Operation check》

Use GX Developer to monitor the circuit.

● If X000 turns ON, the signals will turn on in the following order.



5.2 Introduction example 《2》 [Conveyer control]



《Assignment of I/O》

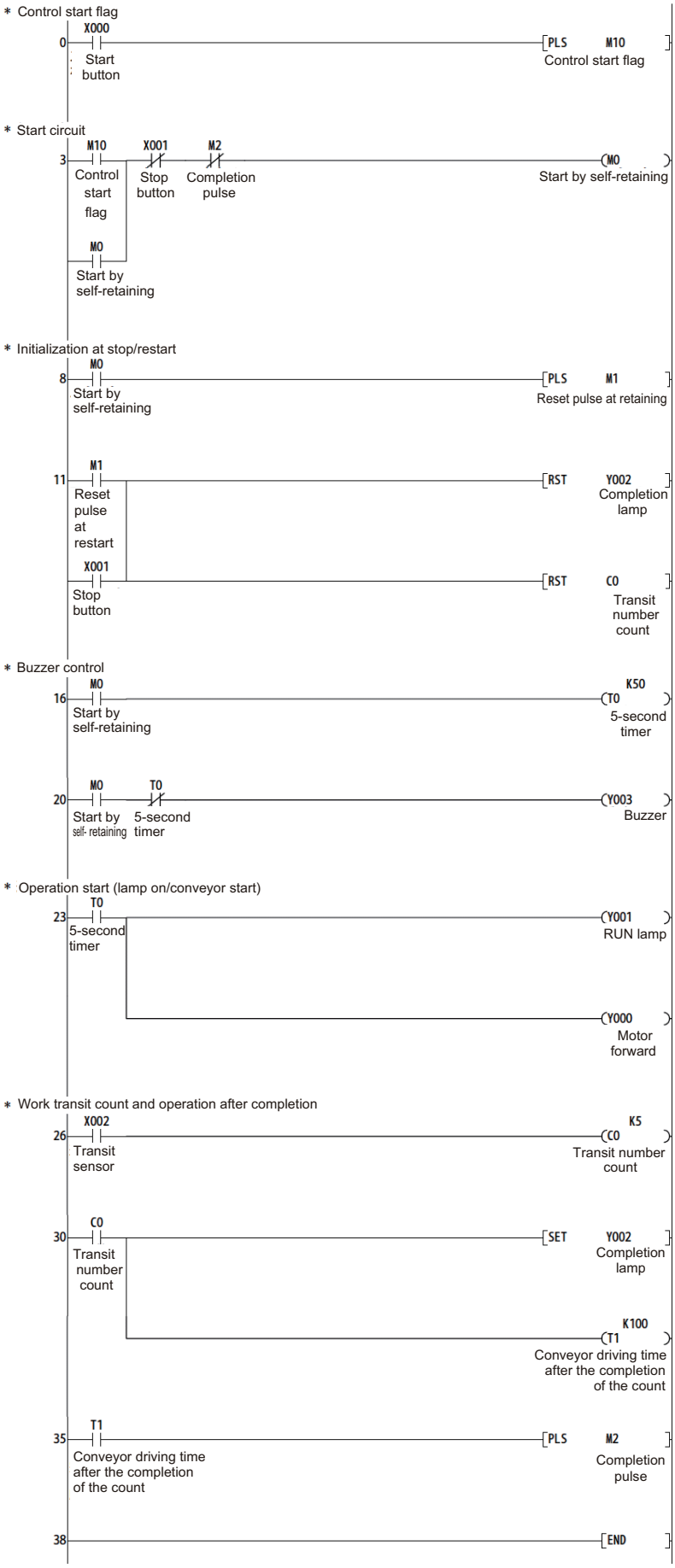
Input	
X000	Start button
X001	Stop button
X002	Transit sensor

output	
Y000	Motor forward
Y001	RUN lamp
Y002	Complete lamp
Y003	Buzzer

《Control specification》

1) If [Start button (X000)] is pressed, [Buzzer (Y003)] sounds for 5 seconds.
2) After that, [Motor forward (Y000)] is activated, and the conveyor begins operating. [RUN lamp (Y001)] is on when the motor is rotating in the forward direction.
3) If [Transit sensor (X002)] detects 5 workpieces, [Complete lamp (Y002)] is turned on and the conveyer stops in 10 seconds.
4) Stop the control by [Stop button (X001)]. Turn on [Start button (X000)] to restart.

《Example of sequence circuit with comments》



- If X000 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 (Y003) sounds for 5 seconds.
- RUN lamp (Y001) and motor forward (Y000) turn on after the buzzer (Y003) stops.
- The counter (C0) counts the number of times the transit sensor (X002) turns on.
- The completion lamp (Y002) will turn ON after the completion of the count.
- The conveyer 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 conveyer stops.

《Operation check》

Use GX Developer to monitor the circuit.

1) X000 (Start button) is turned ON. ⇨ Y003 (Buzzer) is turned on for 5 seconds.

2) 5 seconds later ⇨ Y001 (RUN lamp) is turned on, and Y000 (Motor forward) will be ON.

3) X002 (Transit sensor) is turned on for 5 times. ⇨ Y002 (Completion lamp) is turned on after the fifth ON signal is detected.

4) 10 seconds later ⇨ Y001 (RUN lamp) and Y000 (Motor forward) will be OFF, and the conveyer stops.

MEMO

Chapter 6

THE BASICS OF APPLIED INSTRUCTIONS

It is wasteful to use PLC relay boards as simple interfaces···

The PLC is not only used as a substation for the relay board.
In recent times, the PLC has become a strong assistant with higher added value.
In order to master this assistant, it is necessary to learn the applied instructions described in this textbook.

In order to learn the applied instructions···

The handling of numeric values must be understood as the most basic principle for applied instructions.
In order to provide a better understanding of the applied instructions, this chapter describes the format with which a PLC handles numeric values and it outlines where the numeric data is stored.

Simple introductions for easy understanding···

The following information will gradually introduce the benefits of using applied instructions.
To begin, let's examine applied instructions without a complicated environment. This approach will be valuable for using applied instructions in the future.

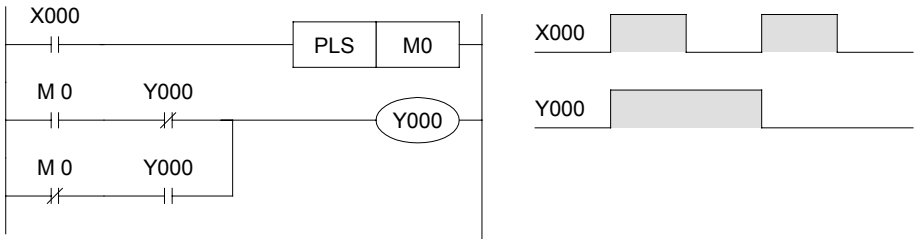
6.1 Applied instructions

Not all applied instructions are complicated. For example, some instructions can provide great benefit for simple control mechanisms such as the alternating circuit below.

● Consider the following circuit.

The purpose of the following control is to turn a lamp on and off with a single pushbutton. The relay ladder circuit is programmed as follows.

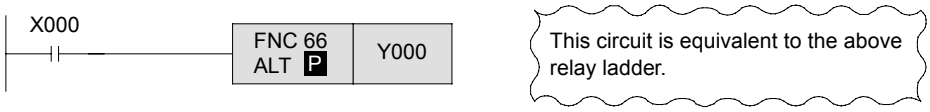
▪ **Alternating Circuit**



▪ This is a familiar circuit for many applications.

▪ If the circuit is written with an applied instruction, the circuit can be simplified as follows:

▪ **Alternating Circuit**



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

Reference

The number of applied instructions for micro PLCs

Many applied instructions are included as standard.

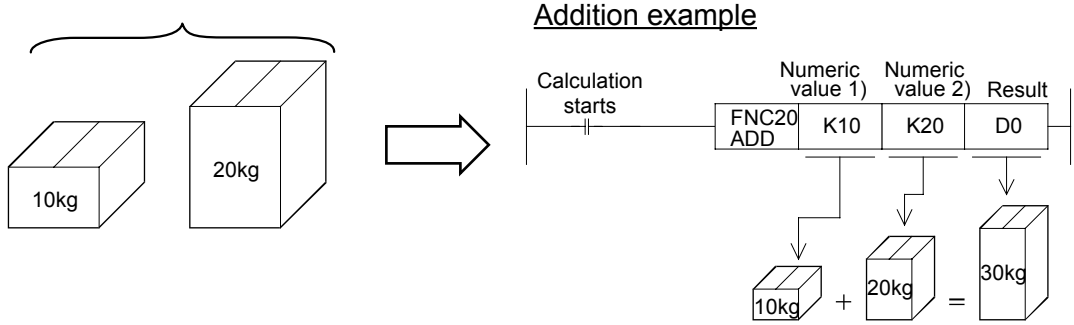
FX_{1S} Series	85	FX_{2N},FX_{2NC} Series	132
FX_{1N},FX_{1NC} Series	89	FX_{3U},FX_{3UC} Series	209

● Along with the applied instructions for simple ON/OFF control as just described, applied instructions can be used for simple numeric data operation.

Operations for simple addition and comparison are described below.

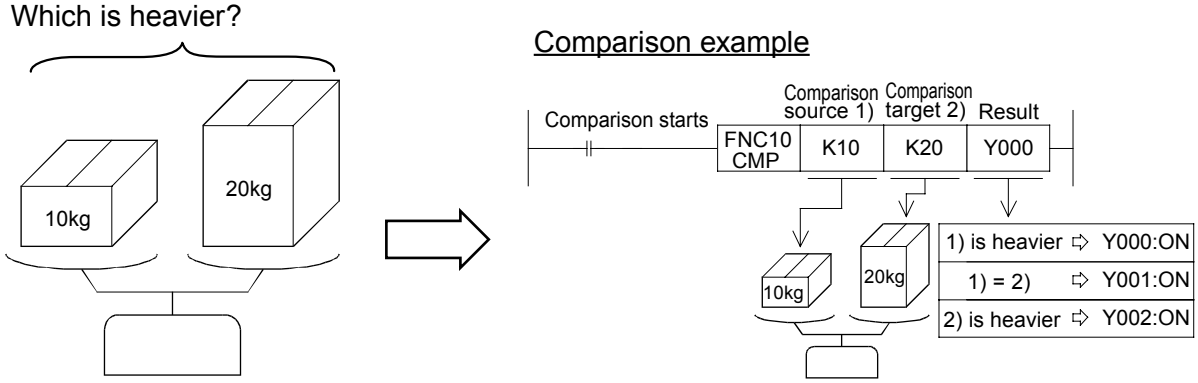
《Arithmetic operation》

Summing two products



《Comparison》

Which is heavier?



Reference

Types of applied instructions

Applied instructions are beneficial for:

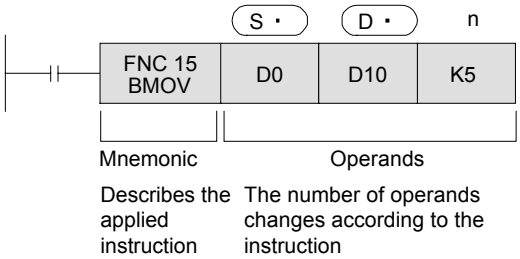
1) Dealing with numeric data regarding comparison, and arithmetic operations.
2) Controlling program flow, executing jumps, subroutines, loops, interrupts and so on.
3) Executing data communications with various handy machines using FROM/TO instructions and other dedicated instructions.
4) Target-oriented 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 applied 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 applied instructions and how to input them using GX Developer

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

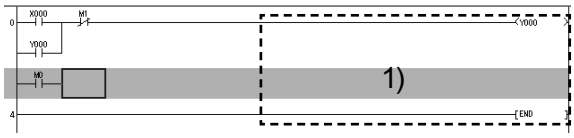


S・ : The so-called source is the operand which does not change according to the execution of the instructions .

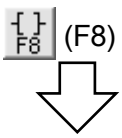
D・ : The so-called destination is the operand which changes according to the execution of the instructions.

m, n : Only a constant K (decimal) or constant H (hexadecimal) can be specified for this operand.
Operands that can be indexed are indicated with "・" (indexing is described later.) Example S・ D・

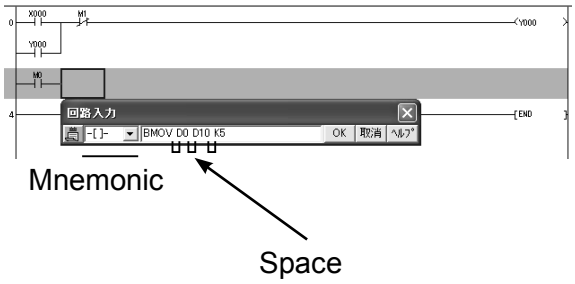
● The input method of GX Developer



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.



1) After the mnemonic, enter unnecessary operands and separate each of them with a space.

1) Press Enter key or click OK key.

6.2 Numeric values used in a PLC

6.2.1 Decimal numbers

The FX PLC uses decimal numbers to label timers, counters, and other devices.

[Typical examples]

Auxiliary 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	…

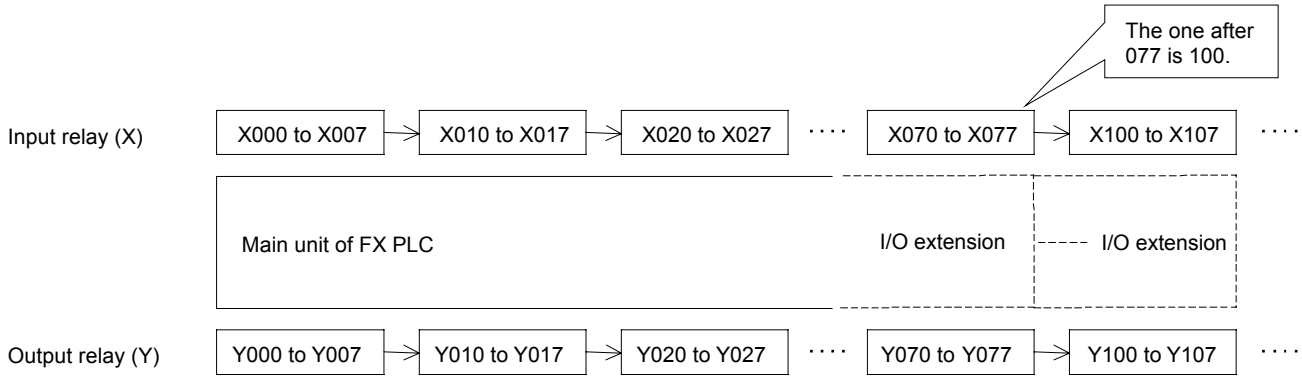
When constants (K) are used for timers, counters, or numeric values in applied instructions, a decimal number is used with "K," such as "K20" to represent the constant.

6

6.2.2 Octal numbers

Octal numbers are base 8 numbers, which use eight unique digits from 0 through 7 as follows: 0 to 7, 10 to 17, 20 to 27…
The FX PLC uses octal numbers as device numbers for input relays (X) and output relays (Y).

● **The FX 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 FX PLC I/O number assignment".

6.2.3 Binary numbers

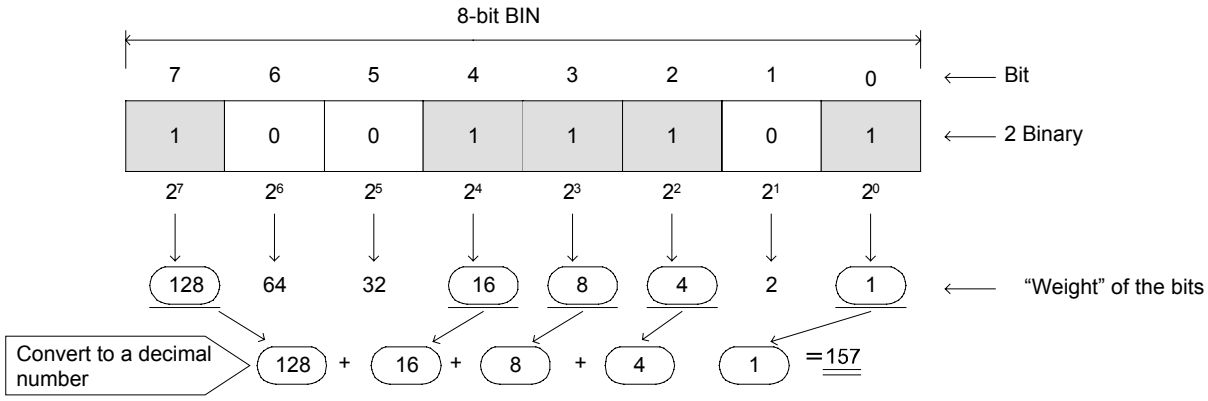
The numbers most people are familiar with are decimal numbers, which use a base 10 representation. Computers and PLCs, however, use binary numbers, which utilize a base 2 system. For example, it is convenient to use the binary number 0 or 1 to correspond to the ON/OFF status of memory locations and relays.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

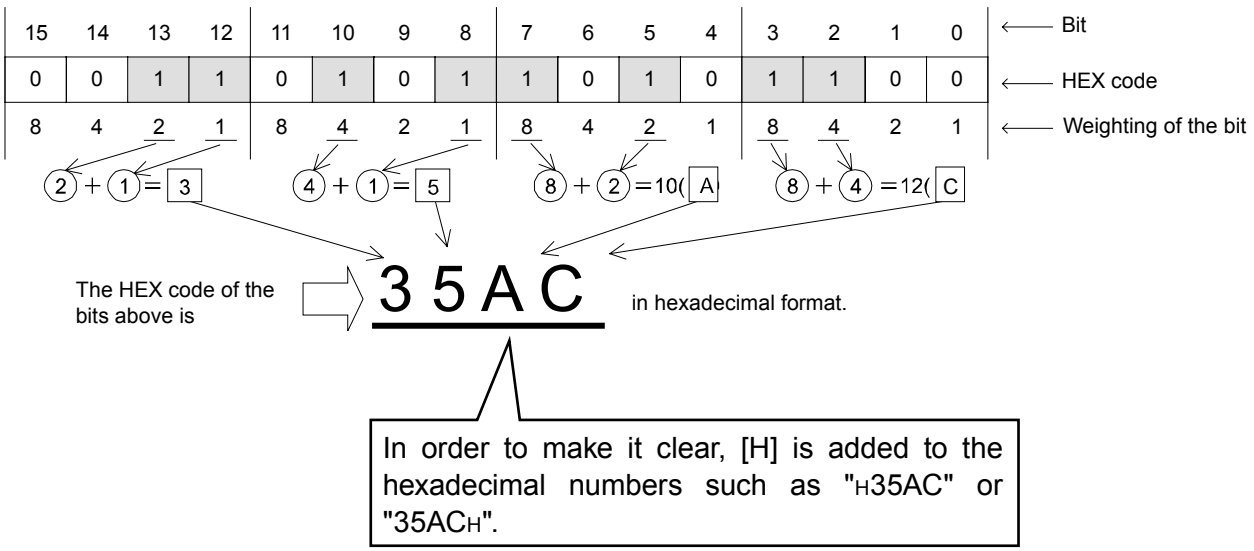
6.2.4 Hexadecimal numbers

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	Constant H and so on	Internal processing of PLC	

● In some cases, the FX PLC uses hexadecimal numbers for the operation setting (buffer memory setting) when special extension equipment is used.



6.2.5 Binary-coded decimal numbers (BCD code)

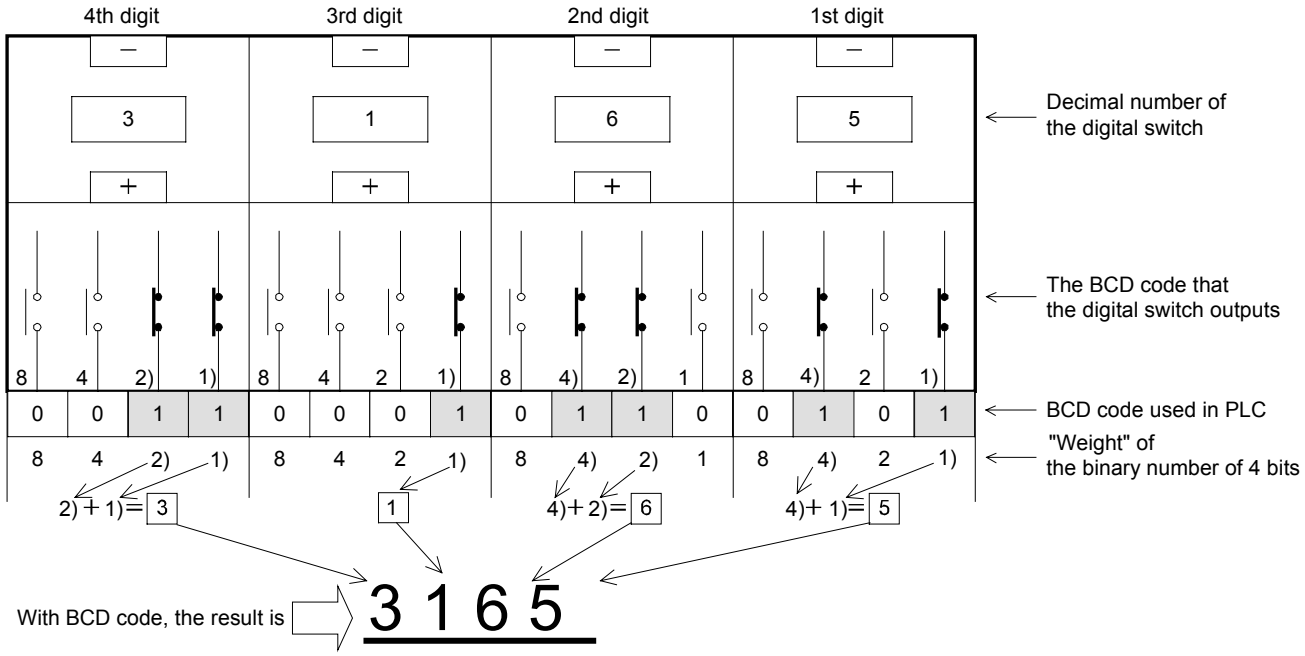
The decimal values 0 to 9 of each digit can be represented in binary numbers of 4 bits by BCD code. BCD code is used for the output signal of a digital switch, the control signal of a 7-segment display unit, or for the signals of various measuring instruments.

[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	BCD digital switch, 7-segment display unit	

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

[Example of 4 digits of the BCD code that the digital switch outputs]



Reference

Numeric values used in the FX 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 (DEC)	Octal (OCT)	Hexadecimal (HEX)	Binary (BIN)	BCD
Constant K	The device numbers of the input relays and output relays	Constant H and so on	The internal processing of PLC	BCD digital switch, 7-segment display unit

6.3 Storing numeric data

6.3.1 Operating word devices

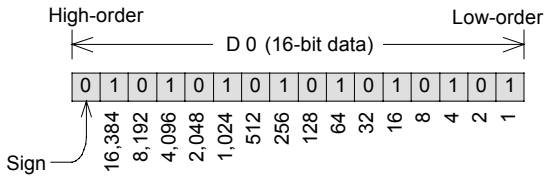
A word device is a register used to store 16-bit and 32-bit numerical data. Word device types (in FX3U, FX3UC PLCs) include the following:

Data registers	
▪ For general use	
D0 to D199	(200 points)
▪ Latched (battery backed) type	
D200 to D7999	(7,800 points)
(File register)	
D1000 to D7999	(7,000 points)
▪ Extension register	
R0 to R32767	(32,768 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.

▪ Special type	
D8000 to D8511	(512 points)
▪ Index type	
V0 to V7	(16 points)
Z0 to Z7	

《16-bit data》

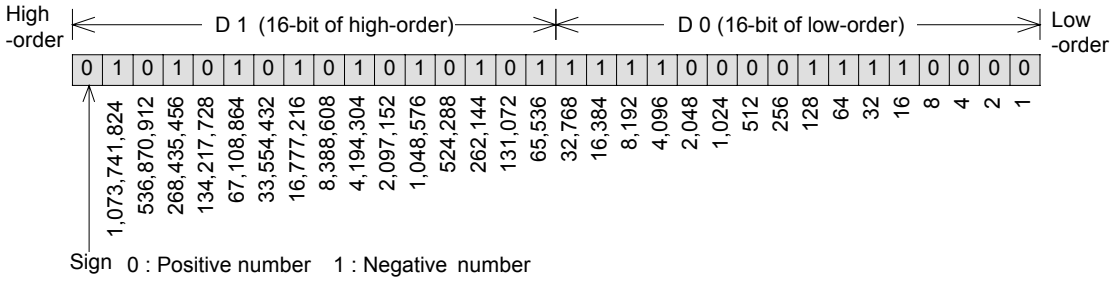


1) A 16-bit data register can represent numeric values from -32768 to 32767.

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.

《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. In index registers, V represents the high-order side while Z represents the low-order side.)
Therefore, one register can represent positive and negative numbers from -2,147,483,648 to +2,147,483,647



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.

The values of timers and counters are stored in current value registers. They can substitute for data registers when they are not used as timers and counters.

The current value register for timers

- For 100 ms

  T0 to T199 (200 points)

- For 10 ms

  T200 to T245 (46 points)

- For 1 ms retentive (Latched (battery-backed))

  T246 to T249 (4 points)

- For 100 ms retentive (Latched (battery-backed))

  T250 to T255 (6 points)

- For 1 ms

  T256 to T511 (256 points)

- All timer registers are a 16-bit. When used as a timer, the most significant bit is fixed to "0", and only integers 0 to 32767 can be stored. (A negative number will lead to an error and the timer will not work.)
- Also, as a timer, units are set at 100 ms, 10 ms, or 1 ms depending on the device number. For example, when T0 is 100, it is equivalent to 10 seconds.

The current value register for up-counters

For general use

C0 to C99 (100 points)

For latched (battery-backed)

C100 to C199 (100 points)

- All up-counting counter registers are a 16-bit. When used as a counter, the most significant bit is fixed to "0", and only integers 0 to 32767 can be stored. (A negative number will lead to an error and the counter will not work.)

The current value register for up- and down-counters

For 100 ms

C200 to C219 (20 points)

For latched (battery-backed)

C220 to C234 (15 points)

For high speed counters (latched (battery-backed))

C235 to C255 (21 points)

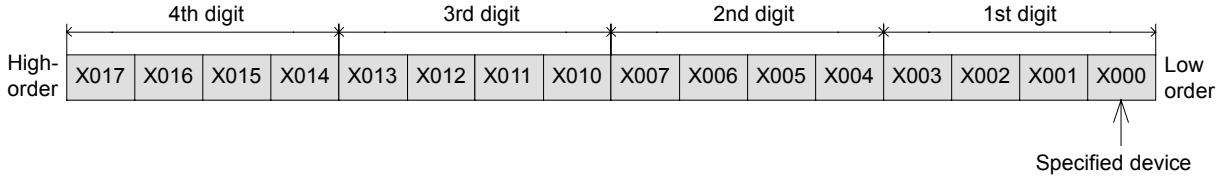
- All bi-directional counter registers are 32-bit. The most significant bit declares whether the value is positive (1) or negative (0).
- 1 register can represent the positive and negative numbers from -2,147,483,648 to +2,147,483,647
- When timer registers or up-counting counter registers are needed for storing 32-bit data, join the two devices with adjacent numbers, as described on the previous page.

- All timers and counters used in applied instructions are all current value registers and cannot be used for coils and output contacts. The sequence instructions (LD, AND, OUT) and so on can be used for the contacts and coils.

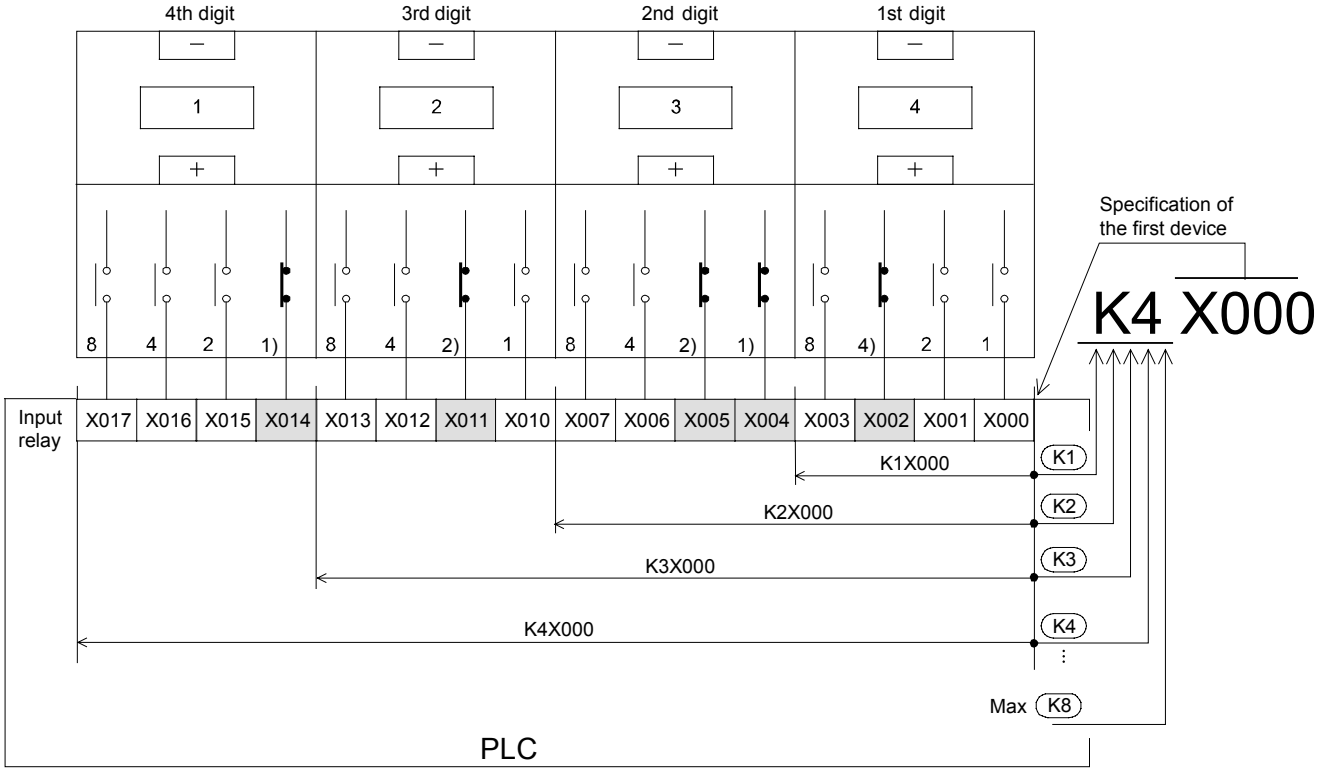
6.3.2 Operating bit devices as word devices (The method to specify the digits)

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

[Example of K4X000]



● Specifying the 4 digits of a digital switch with 4-bit units



K	4	X	000
1)	2)	3)	4)

1) Make sure to use K to specify the digits.
2) Up to 4 digits (K1 to K4) can be specified for a 16-bit operation while up to 8 digits 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

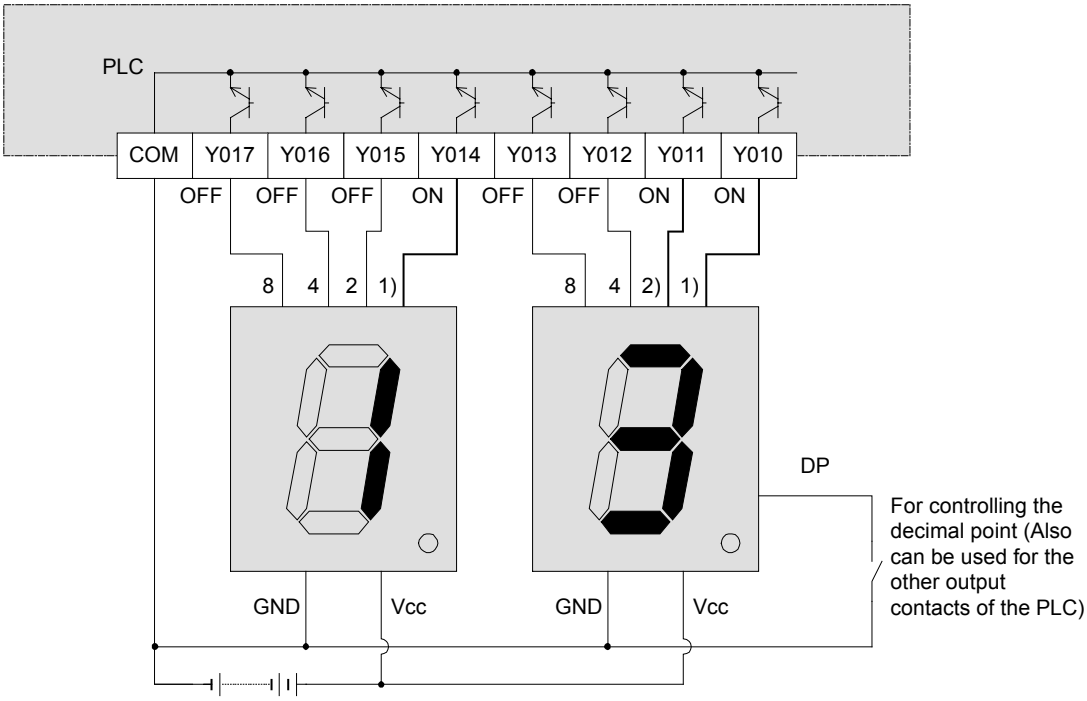
Device numbers can be specified arbitrarily. (Example: K4M13). However, generally, the least significant digit is recommended to be set to 0 such as K1X000, K2Y010, K3M20 and K8S103. (Multiples of 8 are ideal for M and S, but in order to avoid confusion, 0 is recommended to be used.)

Reference

7-Segment driver

To display numbers on a 7-segment display, a PLC outputs BCD of two digits from Y017 to Y010.
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 "K2Y010" in the following example.



6

MEMO

Chapter 7

TRANSFER INSTRUCTIONS FOR NUMERICAL VALUES

Storing and reading numeric values···

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

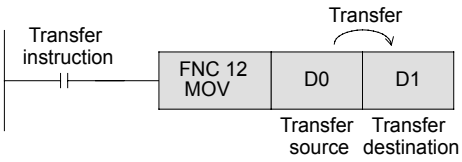
The concept of transferring data is easy···

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 numerical data from a transfer source to a transfer destination. It is a typical applied 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》

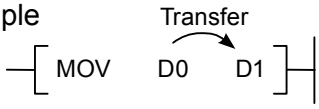


● Flow of data
If the transfer instruction input is ON, the data in the transfer source is written to the transfer destination.
In this situation, the data in the transfer source does not change. When the transfer instruction input is OFF, data is not transferred, and the transfer destination data does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

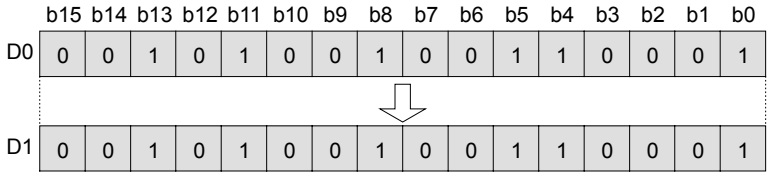
● The data is transferred in BIN.

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

Example

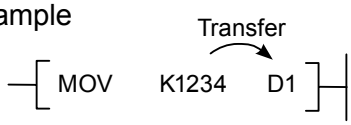


● Transfers data from D0 (16 bits) to D1 (16 bits).

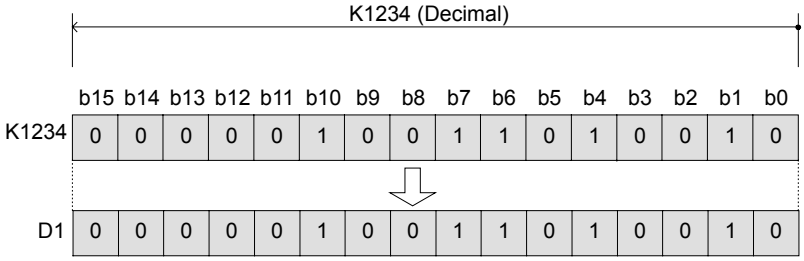


2) Transferring a [constant K (decimal)] to a [word device]

Example

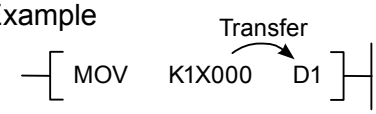


● The PLC treats K1234 as a BIN value and transfers it to D1 (16 bits)

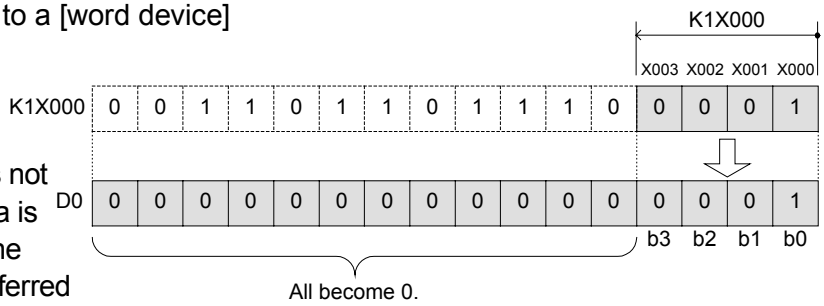


3) Transferring from a [bit device] to a [word device]

Example

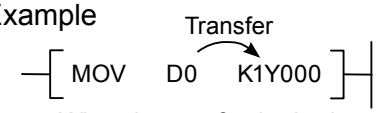


● When the transfer source does not include high-order bits (the data is not 16-bit data), the values in the high-order bits will not be transferred (e.g. the transfer source data is K1X000 to K3X000).

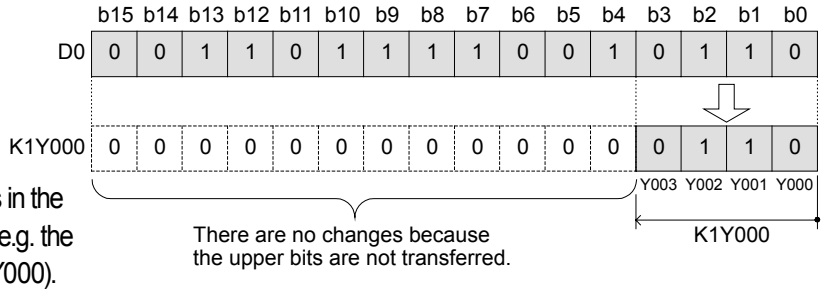


4) Transferring from a [word device] to a [bit device]

Example

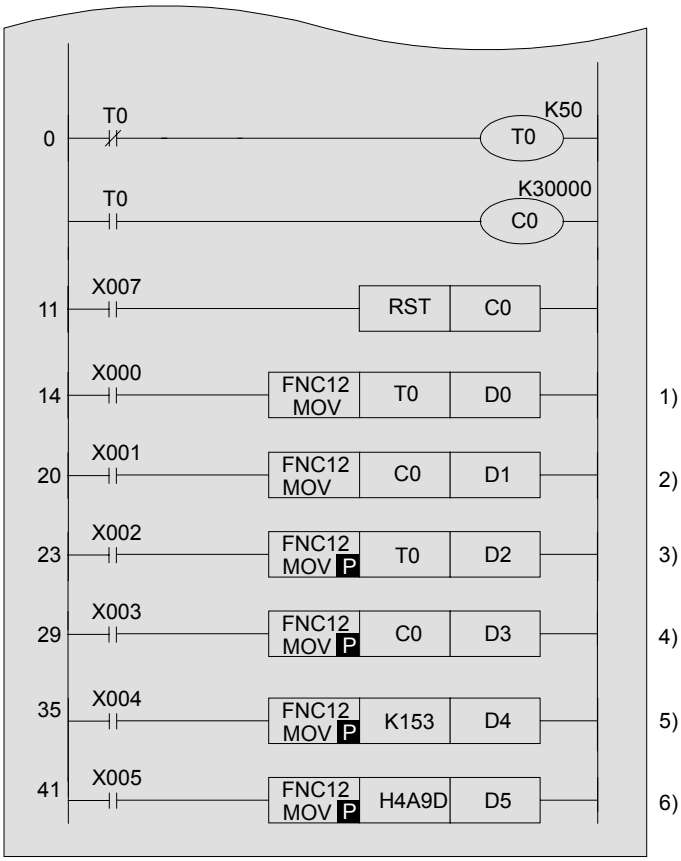


● When the transfer destination does not include high-order bits (the data is not 16-bit data), the values in the high-order bits will not be transferred (e.g. the transfer source data is K1Y000 to K3Y000).



《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.

《Operation check》 Use GX Developer to monitor the circuit.

1) Turn "ON" X000 ⇨
- The current value of timer T0 is always transferred to the data register D0 when X000 is ON.
  If X000 is turned off, the data register D0 keeps the current value of the timer at the time when X000 is turned off.

2) Turn "ON" X001 ⇨
- The current value of counter C0 is always transferred to D1 when X001 is ON.
  If X001 is turned off, the data register D1 keeps the current value of the counter at the time when X001 is turned off.

3) Turn "ON" X002 ⇨
- The current value of timer T0 at the time when X002 is turned on is transferred to data register D2.

4) Turn "ON" X003 ⇨
- The current value of the counter at the time when X003 is turned on is transferred to data register D3

5) Turn "ON" X004 ⇨
- "153 (Decimal)" is directly transferred to D4. (Initial data setting)

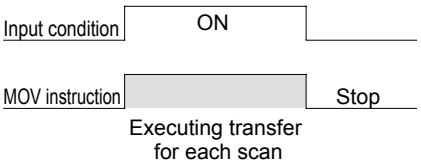
6) Turn "ON" X005 ⇨
- The decimal "19101" is transferred to D5. (Initial data setting)
  Select [Online]—[Monitor]—[Change current monitor value (Hexadecimal)]. "H4A9D" is displayed.
  To return it to decimal, select [Change current monitor value (decimal)] from the same menu .

Point

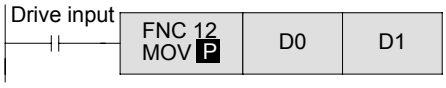
Continuous execution vs. Pulse execution

[Continuous execution instruction]

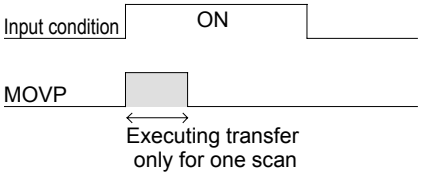
- For the continuous execution type instruction, when the drive input is ON, the instruction is executed and the transfer is performed for each scan.



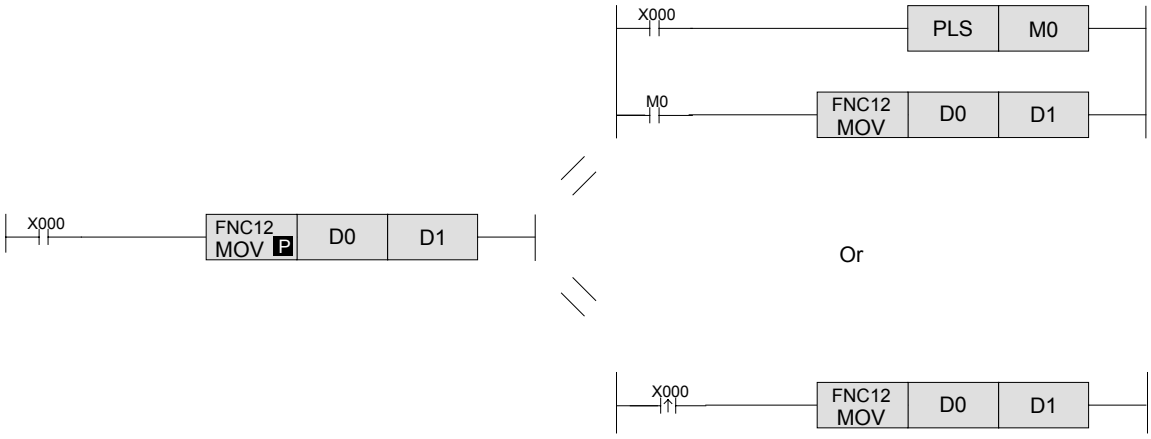
[Pulse execution instruction]



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



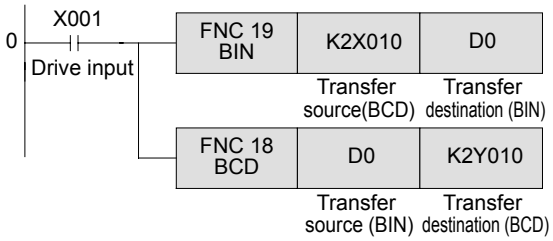
- Thus, the continuous execution instruction is used when data is constantly changing, while the pulse execution instruction is used for transfers that are needed only when the instruction is input turns on (When setting the initial value or the value at the specified time.)
- If the drive input is OFF, 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 unit using BCD, it is necessary to use the BIN-to-BCD conversion transfer instruction.

《Operation outline》

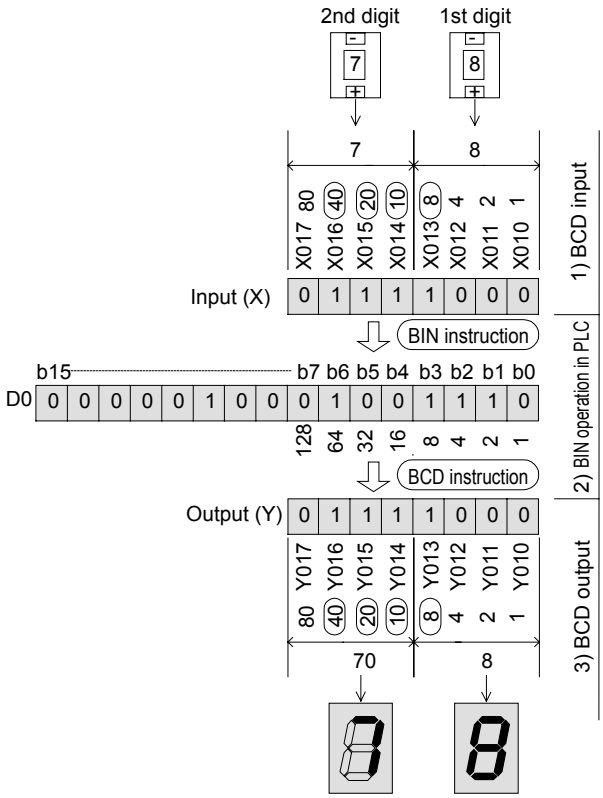


BIN instruction.....When the drive input is ON, the BCD value of the transfer source is converted to BIN and transferred.

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 X010 to X013 (the first digit) and X014 to X017 (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 Y010 to Y013 (the first digit) and Y014 to Y017 (the second digit) for operating the 7-segment display unit (BCD instruction).

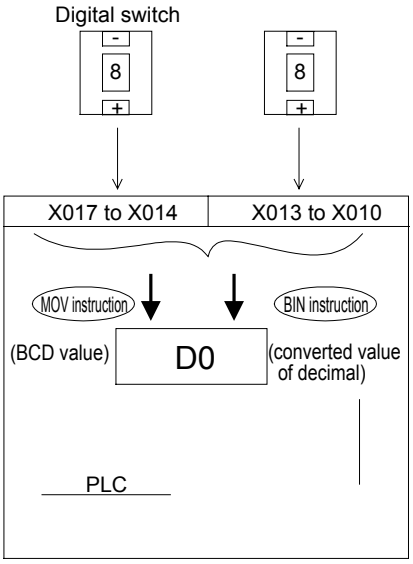
(Reference)

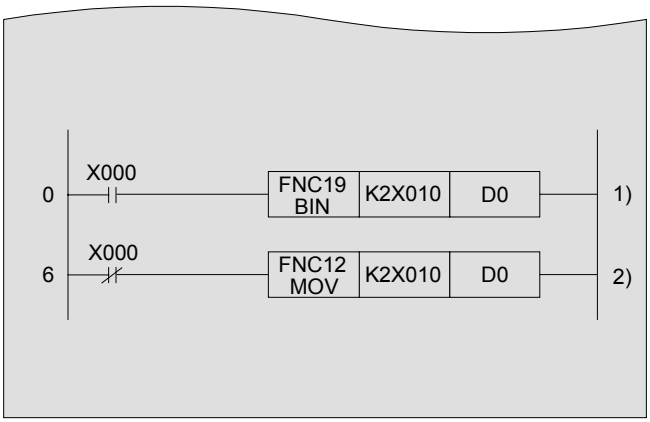
Input of digital SW	BCD value
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

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.





《Operation check》

Use GX Developer to monitor device D0.

1) Turn "ON" X000 ⇨ The value of the digital switch (BCD input) is converted to a decimal value by "BIN instruction" and transferred to D0.

2) Turn "OFF" X000 ⇨ The input is the same as 1) 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

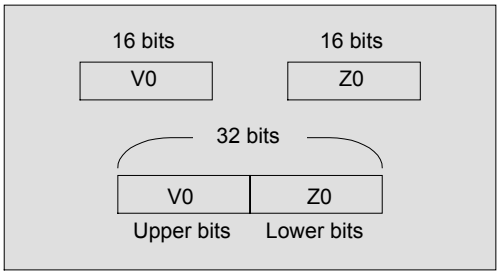
Operation error and error step number

- 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 auxiliary relay or special data register.
  Moreover, the check can be performed by selecting [Diagnostics]-[PLC diagnostics] from the menu of GX Developer.

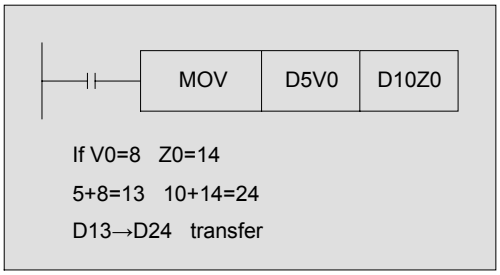
M8067 D8067 D8069	If an operation error occurs, M8067 will turn on, and the error code of the operation error will be stored to D8067. The error step number will be stored to D8069. If a new error occurs at another step, the error code of this instruction and the step number will be updated sequentially. The errors are cleared instantly when the status of PLC changes from STOP to RUN.
M8068 D8068	If an operation error occurs, M8068 will be turn on, and the error step number will be stored to D8068. The value will not be updated even if a new error occurs and it will remain until the device is reset or the power is turned off.

7.3 Indirectly specifying the transfer source and transfer destination

Devices in applied 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.



- V and Z are 16-bit data registers where numeric values can be written to and read from.
  - V0 to V7 : 8 points
  - Z0 to Z7 : 8 points
- In the case of 32-bit operations, V and Z can be used together. Specify the Z side in such a case.



- Changing device numbers according to the values of V and Z is called indexing.
  In the case of a constant, for example, when V0=8, K20V0 stands for K28. (20+8=28)

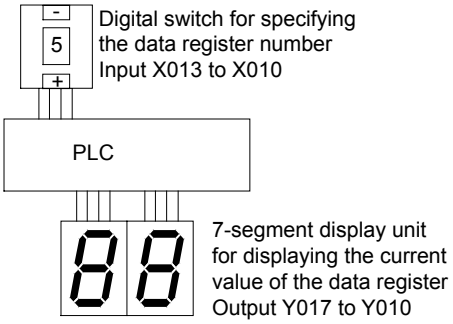
The devices that can be indexed

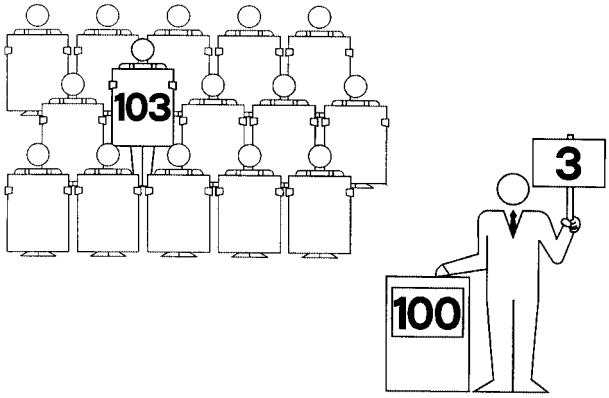
X, Y, M, S, P
T, C, D
K, H, KnX
KnY, KnM, KnS

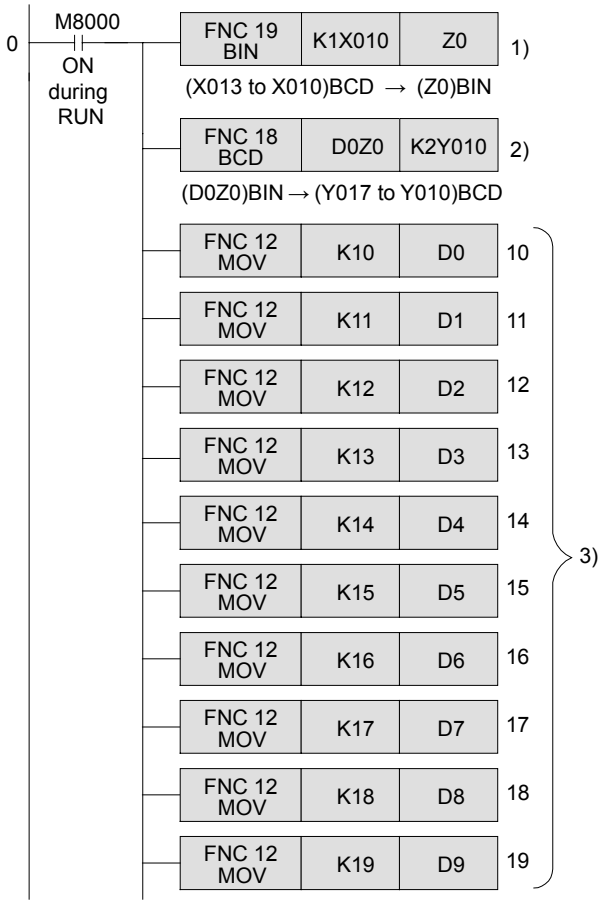
The main devices that can be indexed with index registers are those listed on the left. The devices on the left are used in the applied instructions.
However, as shown below, Kn for the digit specification cannot be indexed.
(K4M0Z0 is valid, while K0Z0M0 is invalid)

《Instruction operation》

● Monitor the current values of the data registers (D0 to D9) that are specified by the index register Z.







《Operation check》

1) The BCD value of the digital switch DSW1 (X013 to X010) is transferred to the index register Z0 in BIN.

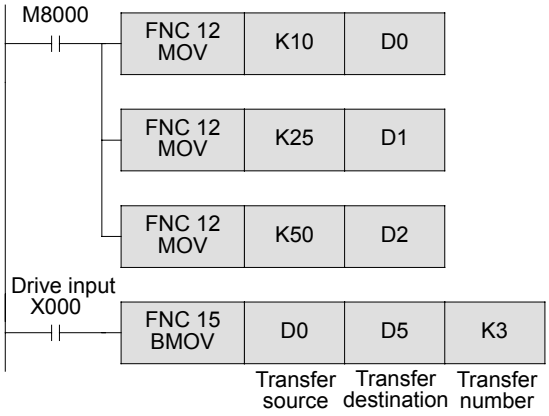
2) The current value in the data register whose number is indexed by index register Z0 is displayed in the 7-segment display unit (Y017 to Y010).
*In the transfer source D0Z0, the data register number changes between D0 and D9 according to the value of Z0.

3) The program writes the values 10 to 19 to data registers (D0 to D9).

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.

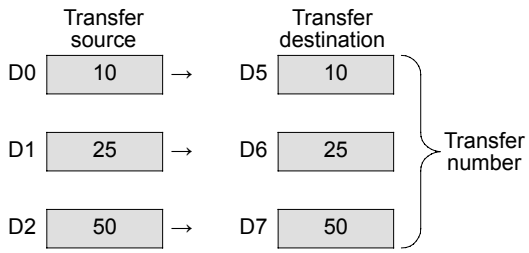


Transfer K10 to D0

Transfer K25 to D1

Transfer K50 to D2

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



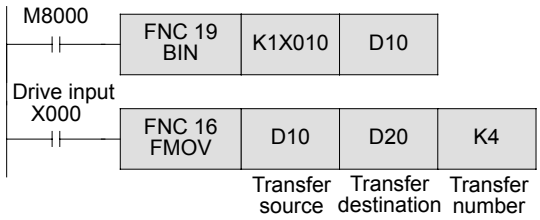
- If the transfer number is larger than the device range, the data within the range will be transferred.
- In the case of bit devices for digit specification, set the same number of digits between the transfer source and destination.

  BMOV K2X000 K2Y000 K5

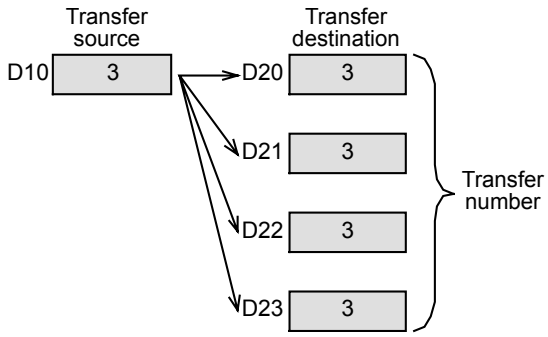
  The number of digits is the same!

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.



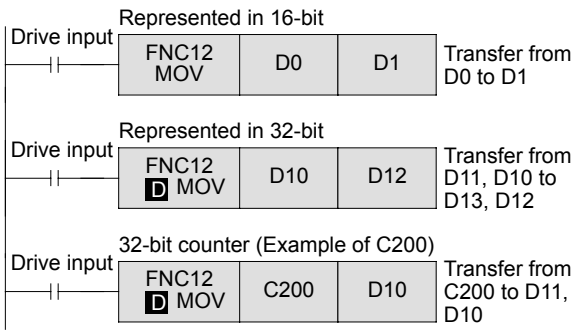
● If the transfer number is larger than the device range, the data within the range will be transferred.

7

Reference

Bit length for numeric values

● The applied 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".

● For 32-bit instructions, the transfer source and transfer destination device numbers can be either even or odd. In every case, however, 32-bits of data (ie - two data registers) are transferred.

● 1 point for a 32-bit counter occupies 32 bits. An operation error will occur if 16-bit instructions are used.

MEMO

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

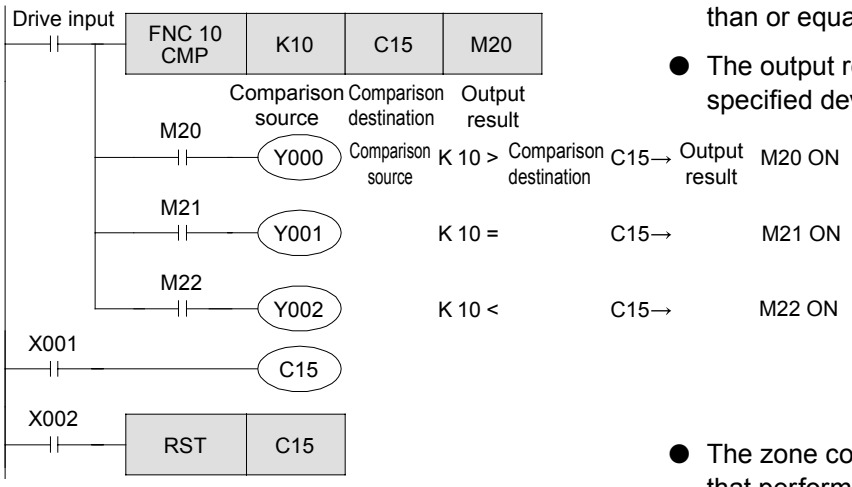
The comparison instructions are used when comparing the current values stored in the data registers, the timers and the counters, or the values representing the combined relays of X, Y, M, or S, along with constants K.

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》

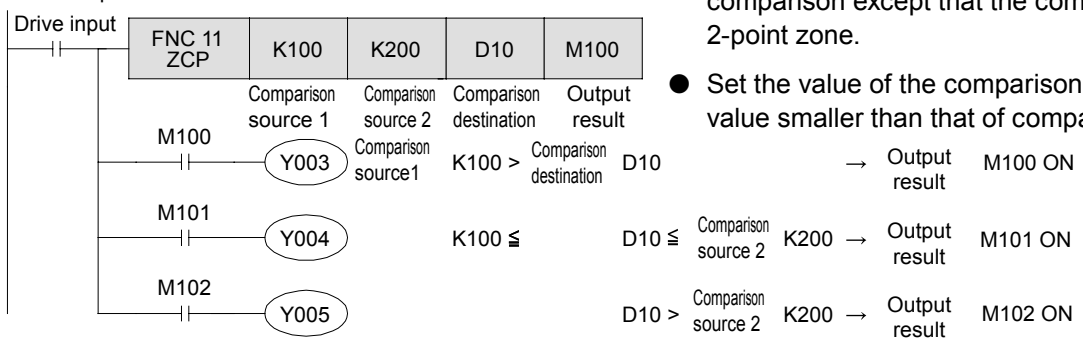
- If the drive input is turned on, a result is output to display, whether the target is larger than, smaller than or equal to the comparison source.
- The output result occupies the first 3 points of the specified device.

One point 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.

Zone comparison



Reference

Operation of output result

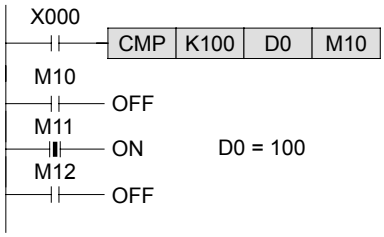
In the following cases, the output result for data comparison instructions will not change:

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 M8067 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 M8067 is not turned on in this case.)

Example

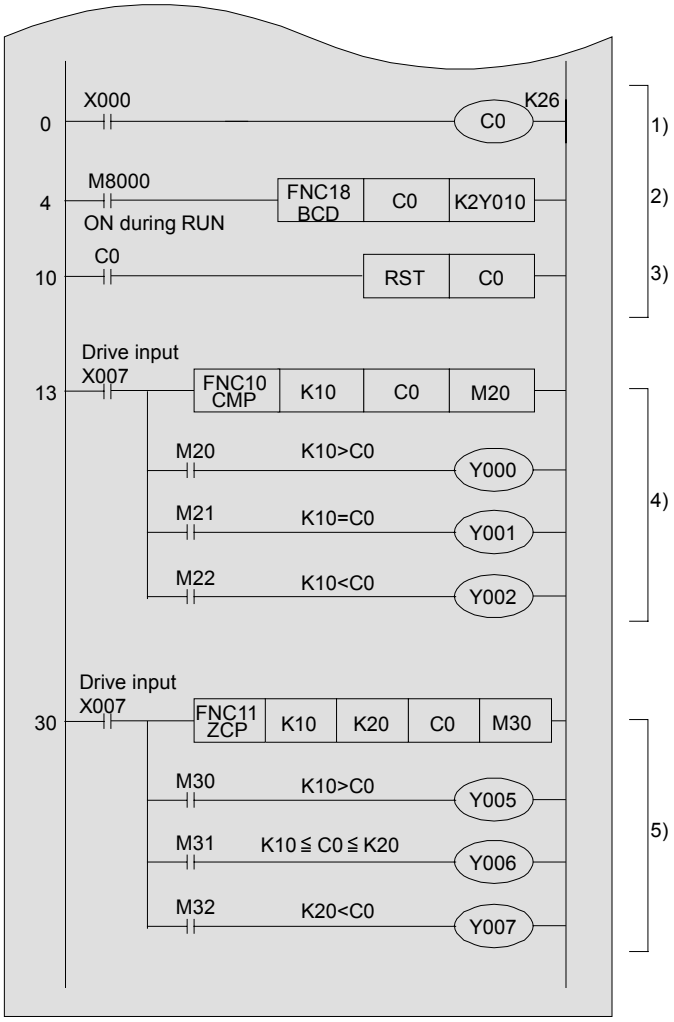


When the output result directly starts from the bus line as shown in the left diagram and the drive input X0 is turned off, the output M11 remains ON without regard to changges in D0.

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 X000 is "ON".
2) The current value of the counter C0 is indicated in the 7-segment display unit.
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
	Y002: ON (M22: ON)	Y001: ON (M21: ON)	Y000: ON (M20: ON)

5) Zone comparison

Current value of counter	21 or more	10 to 20	0 to 9
	Y007: ON (M32: ON)	Y006: ON (M31: ON)	Y005: ON (M30: ON)

8

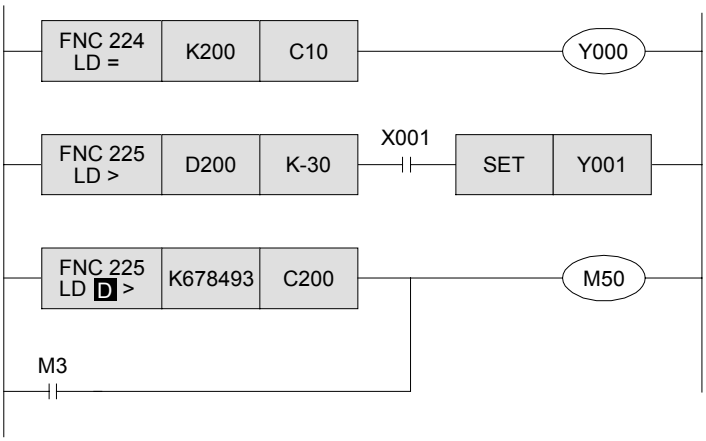
8.2 Contact comparison instructions (LD□, AND□, OR□)

Contact comparison instructions can use a comparison result as the contact information in a circuit.

《Operation outline》

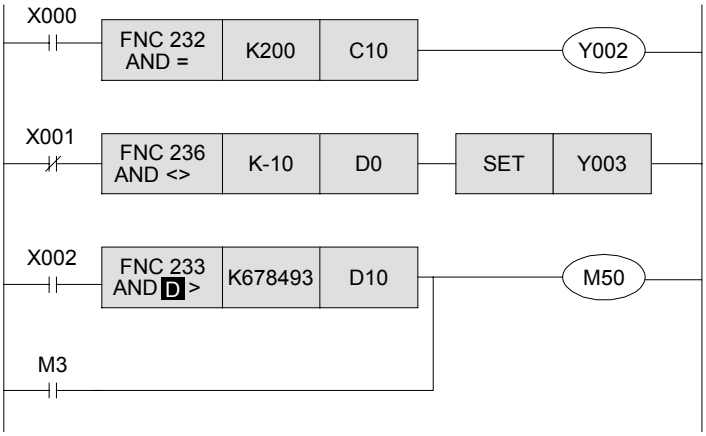
- The contact compare instructions are divided into 3 types depending on their placement within a program: LD contacts (connected from the bus line), AND contacts (connected to others serially), and OR contacts (connected to others in parallel).
- "=, >, <, <>, <=, >=" can be input after [ ] F8 in a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

LD □ □: =, >, <, <>, ≦, ≧



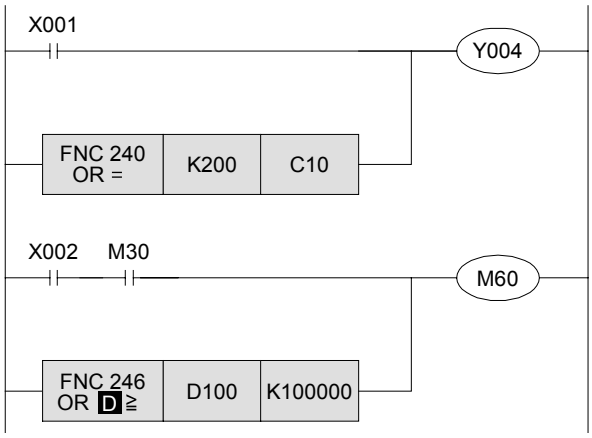
- Y000 is turned on when the current value of the counter C10 is 200.
- Y001 is turned on when the value of D200 is -29 or more and X001 is ON.
- M50 is turned on when the value of the counter C200 is smaller than 678493 or M3 is ON.
- 32-bit instructions are used for 32-bit counters (C200 onwards).

AND □ □: =, >, <, <>, ≦, ≧



- Y002 is turned on when X000 is ON and the current value of the data register C10 is 200.
- Y003 is SET when X001 is OFF and the value of the data register D0 is not -10.
- When X002 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 □ □: =, >, <, <>, ≦, ≧



- Y004 is turned on when X001 is turned on or the current value of the counter C10 is 200.

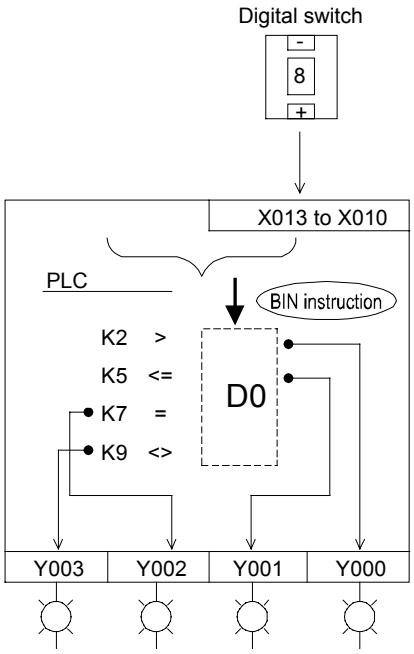
- M60 is turned on when both X002 and M30 are turned on, or the values of the data register D101 and D100 is 100000 or more.

- 32-bit instructions are used for 32-bit data .

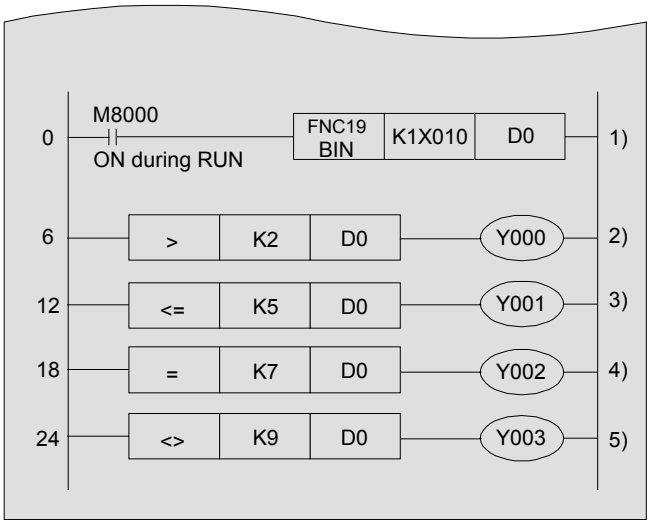
8

《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) The values of Y000 to Y003 change according to the values of the digital switches 2) to 5) as show below.

●: Input ON —: Output OFF

DSW value	Output			
	Y003	Y002	Y001	Y000
0	●	—	—	●
1	●	—	—	●
2	●	—	—	—
3	●	—	—	—
4	●	—	—	—
5	●	—	●	—
6	●	—	●	—
7	●	●	●	—
8	●	—	●	—
9	—	—	●	—

Chapter 9
ARITHMETIC OPERATION

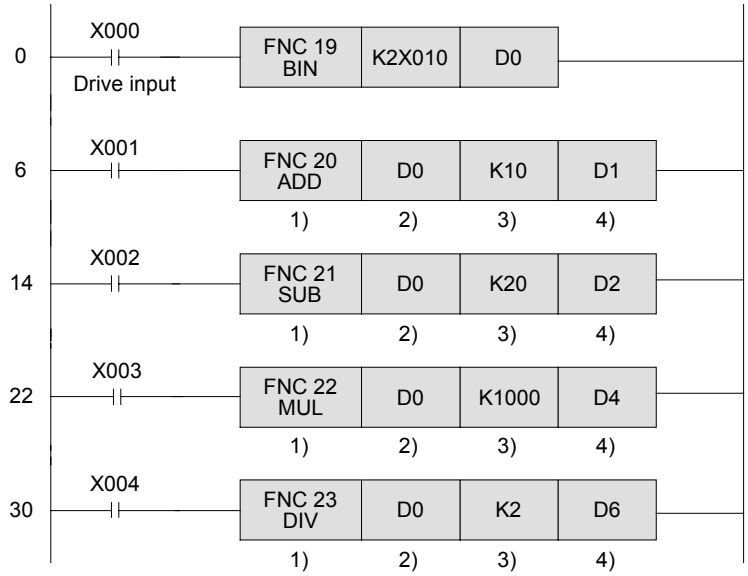
Arithmetic operation (+, −, ×, ÷)

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 (ADD, SUB, MUL, DIV)

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

《Operation outline》



The BCD values X010 to X017 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
   FNC20 ADD : BIN ADDITION
   FNC21 SUB : BIN SUBTRACTION
   FNC22 MUL : BIN MULTIPLICATION
   FNC23 DIV : 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 ADD, SUB, MUL, DIV

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 ADD, D SUB, D MUL, D DIV

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

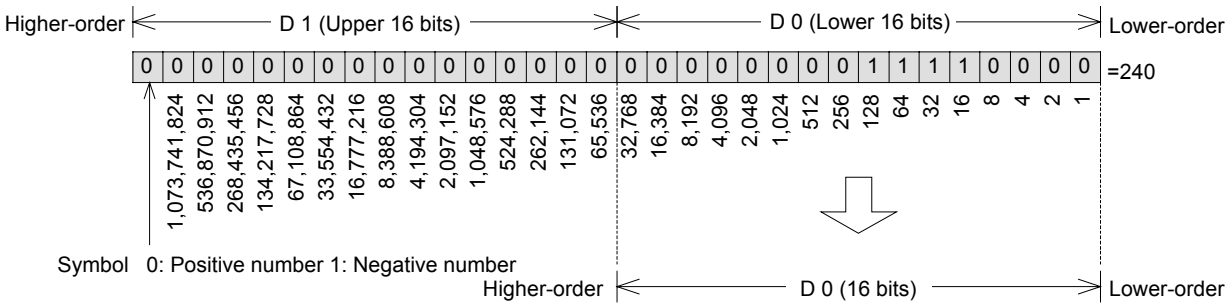
* The operation result is 32-bit for multiplication (64 bits for the D instruction), and for division, a register is occupied by the remainder. Thus, it is necessary to avoid storage destination conflicts for operation results of multiplication and division instructions.

* For the 32-bit multiplication instruction (D MUL), 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 FNC49FLT instruction is used, floating point operation for values below decimal-point is also available.

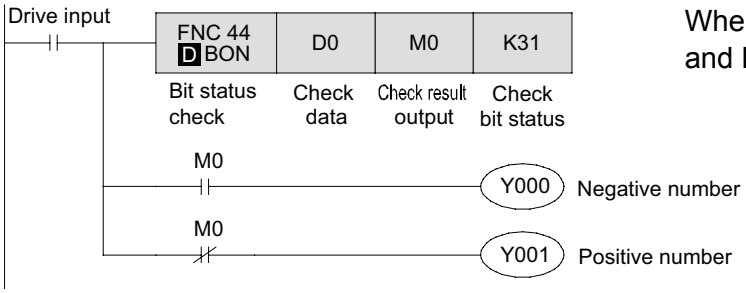
The operation result of the multiplication instruction (product) is represented as 32-bit data with the 16-bit operation, while 64-bit data is required for the result of the 32-bit operation.

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.

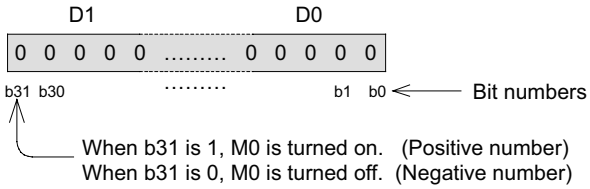


On the right, the lower 16 bits of the 32-bit data are shown.

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 (FNC44 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

Importing high speed input signals to a PLC…

Normally, in order to securely import 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 FX Series PLCs have built-in functions to import high-speed signals, which makes it possible to process high frequency signals.

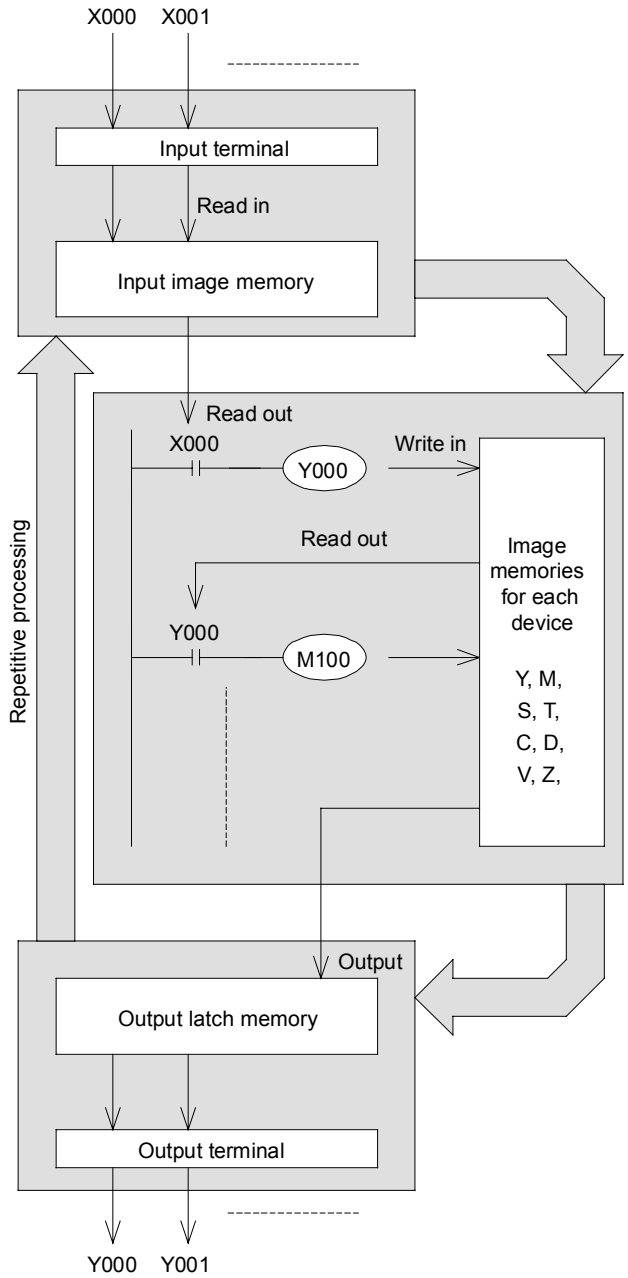
Using high speed processing…

"Input interrupts" and "high speed counters" are two examples of control mechanisms that use high speed processing procedures. With FX PLCs, a variety of built-in high speed functions are available for for importing 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" called a "batch refresh".
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 import 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 an 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.

Categories of high speed processing in FX Series PLCs

For interruption processing

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

● Input interrupt: Six points of X000 to X005 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, three points from 10 to 99 ms can be used.

For high speed counter

High speed counters count how many times the inputs X000 to X007 are turned on or off.
FX3U and FX3UC Series PLCs can count short pulses within 100 kHz for 1-phase or 50 kHz for 2-phase.
If the output is required immediately when the current value of the counter reaches its set value, it is necessary to use the high speed counter FNC53 HSCS comparison set, FNC54 HSCR comparison reset and FNC55 HSZ band comparison instructions in conjunction.

For pulse catch

When an input relay (X000 to X007) is turned on, the special auxiliary relay (M8170 to M8177) is set through interrupt processing.
The set special auxiliary relay can be reset by the RST instruction. After the reset, the relay can be used again to read inputs.

For applied instruction

In addition to the interrupt counters and high speed counters, the following applied instructions from FNC52 to FNC58 are executed at high speed.

FNC52 :	MTR	Input Matrix
FNC53 :	HSCS	High Speed Counter Set
FNC54 :	HSCR	High Speed Counter Reset
FNC55 :	HSZ	High Speed Counter Zone Compare
FNC56 :	SPD	Speed Detection
FNC57 :	PLSY	Pulse Y Output
FNC58 :	PWM	Pulse Width Modulation

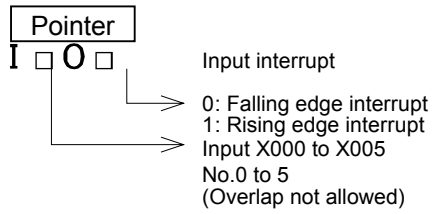
《Additional note》 For input designation

Inputs X000 to X007 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 X000 is used for an interrupt instruction, it cannot be used for a high speed counter. If X000 is programmed for both purposes, an error will occur.

10.2 Using input interrupts

For interrupt inputs, six points from X000 to X005 can be used.
Interrupt programs are started with interrupt pointers, which are used as labels and placed after the FEND instruction in programs. They are ended with IRET instructions.
A single input number cannot be used for multiple interrupt pointers (overlap of numbers is not allowed).

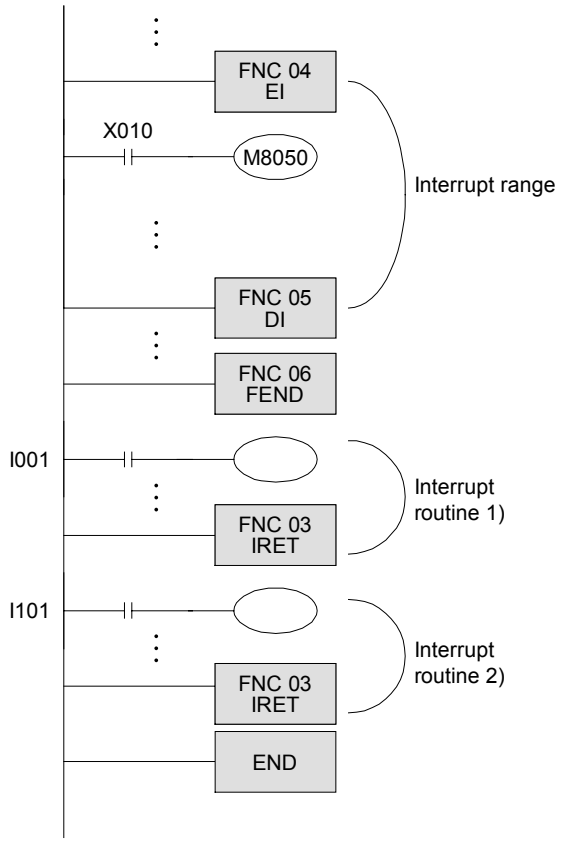
《Operation outline》



Input	Input interrupt pointer		Disable interrupt flag
	Rising edge interrupt	Falling edge interrupt	
X000	I001	I000	M8050
X001	I101	I100	M8051
X002	I201	I200	M8052
X003	I301	I300	M8053
X004	I401	I400	M8054
X005	I501	I500	M8055

《Operation outline》

For example, when X000 is turned on, the program moves to the label of the pointer I001 and executes the sequence instructions that follow the label. The program returns to the suspended instruction when the IRET instruction is executed.

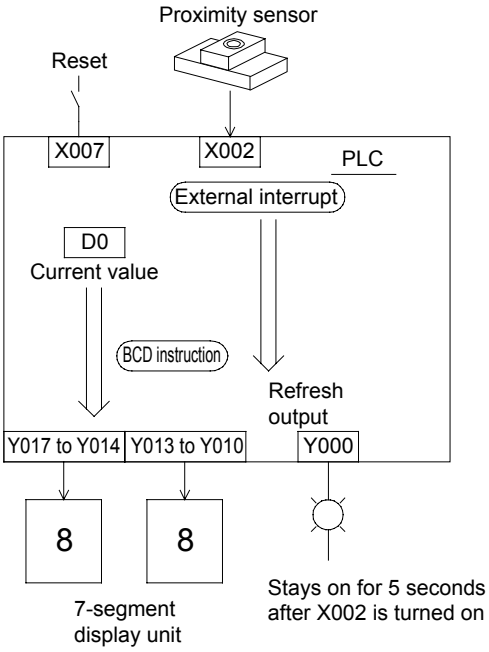


- If X000 or X001 is turned on when the program is executed between the EI and the DE instructions, the interrupt routine 1) or 2) is executed. The program returns to the main program by the IRET instruction.
- While the special auxiliary relay M805 △ (△ represents 1 to 5) is ON, the interrupt routine of I △ ** is not executed.
  In the figure to the left, the interrupt routine for I001 is not executed by the rising edge of X000 when X010 is ON.
- Basically, an interrupt program cannot be executed while another interrupt program is in progress. However, by placing the EI and the DI instructions in an interrupt program, up to two interrupt programs can be executed together.
- In the subroutine or interrupt routine, use T192 to T199 or T246 to T249 for the timers. The actions are identical to that of the jump instruction.

- If multiple interrupt routines are triggered sequentially, the routine that is triggered earlier is executed. If two interrupt programs are triggered at exactly the same time, the one with lower pointer number is executed first.
- When an interrupt program is executed between the DI and the EI instructions (the interrupt-disabled segment), the interrupt program is saved, and then executed after the EI instruction (excluding cases where the special auxiliary relay M805 △ is ON). The longer the interrupt-disabled segment, the longer it takes to accept the interrupt instruction.

《Instruction operation》

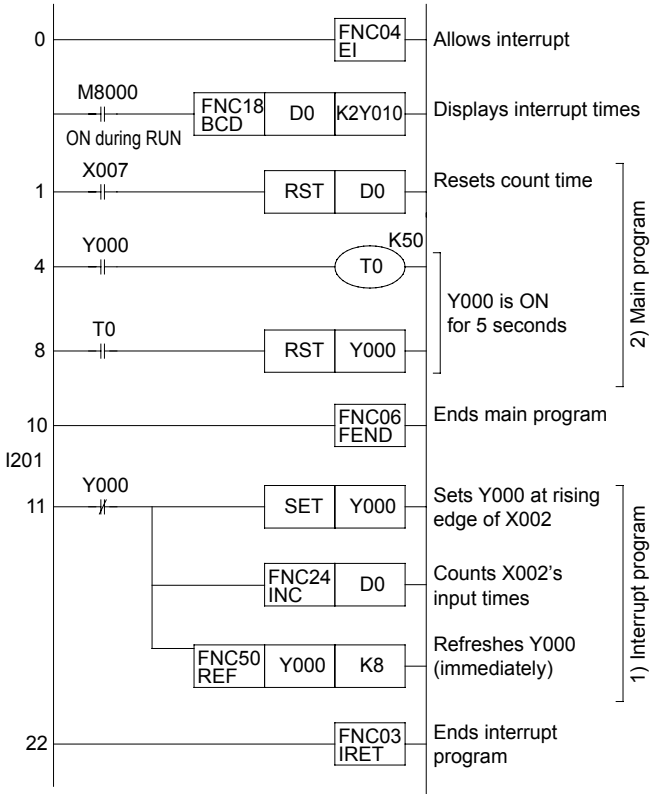
Confirm the actions of the external interrupt signal of X002.



《Operation check》

1) • Y000 is turned on when X002 is turned on, and the data is output immediately by the REF instruction.
X002 is used for high speed input. The input signal needs to stay on for only 5 μs to be recognized.
• In this example program, the PLC does not recognize a signal from X002 while Y000 is ON (5 seconds).
• The number of times X002 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 X007 is turned on.
• The timer T0 times out and is reset after Y000 stays on for 5 seconds.



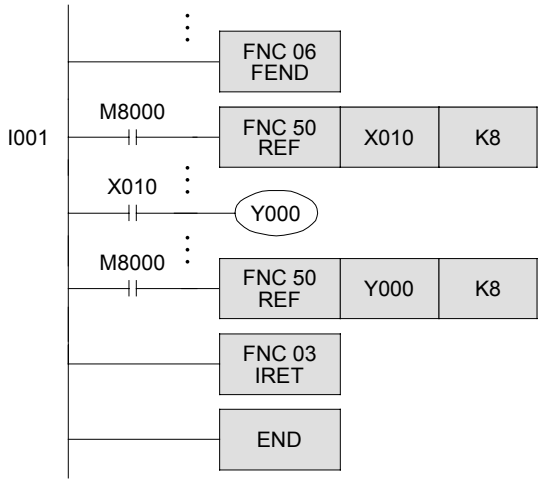
Reference

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
- The interrupt program of I001 is executed when X000 is turned on.
- The values of X010 to X017 are stored.
- X010 is refreshed and then processed.
- The result of the operation is output. (Y000 to Y007)
- End of the interrupt program

10.3 Using a timer interrupt program

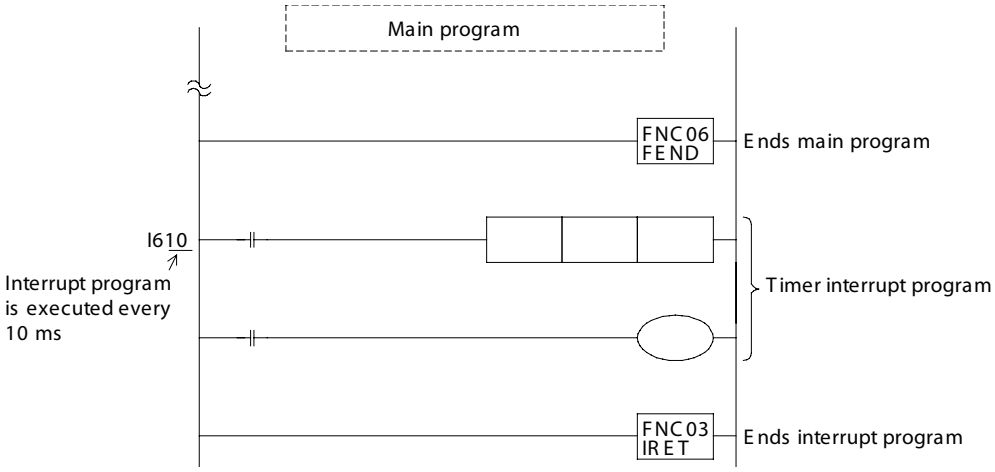
Three points can be used in units of 10 to 99 ms for timer interrupt programs.
A timer interrupt program begins with an interrupt pointer, which is used as a label and placed after the FEND instruction in the program, and ends with the IRET instruction.

Pointer

I ○○○
1) 2)

1) Only 6, 7, and 8 can be assigned for timer interrupt programs. The number must not be overlapped.
2) A number between 10 to 99 ms is specified as the cycle for the interrupt program.

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 pointer I is used as the timer value, so there is no need to program another timer circuit.
The following instructions use several operation cycles to execute a series of actions: FNC76(RAMP), FNC71(HKY), FNC74(SEGL), FNC77(PR)
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

10.4 Using high speed counters

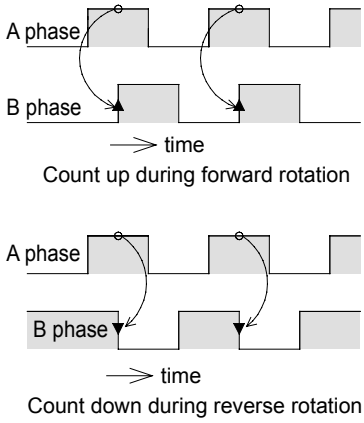
10.4.1 Types of high speed counters

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 right page, the input of the high speed counter is specified by the device number of the high speed counter. The input can be X000 to X007.
The high speed counter can be classified into three types by their counting system.

		Input signals form	Counting directions
1-phase 1-counting input		UP/DOWN	The counting direction (up or down) is specified with M8235 to M8245. ON: Down-counting OFF: Up-counting
1-phase 2-counting input		UP: +1 +1 +1 DOWN: -1 -1 -1	This counter counts up or down as shown on the left. The counting direction of the counter is specified with M8246 to M8250. ON: Down-counting OFF: Up-counting
2-phases 2-counting input	1-edge count	A phase / B phase: +1 +1 Forward rotation A phase / B phase: -1 -1 Reverse rotation	This counter counts up or down automatically according to the input status in A- and B-phases. The counting direction of the counter is specified with M8251 to M8255. ON: Down-counting OFF: Up-counting
	4-edge count (only FX3U, FX3UC)	A phase / B phase: +1 +1 +1 +1 +1 / +1 +1 +1 +1 Forward rotation A phase / B phase: -1 -1 -1 -1 -1 / -1 -1 -1 -1 Reverse rotation	

Reference

Definition of a 2-phase type encoder



- A 2-phase type encoder is installed on the rotary shaft of some machinery, which generates a signal that indicates the rotation direction (forward or reverse) of the machinery. To make a counter count up or down automatically according to the rotation direction of the machinery, the counter must be a 2-phase counter.
- The 2-phase encoder generates outputs for A-phase and B-phase, which are 90 degrees apart from each other. As a result, the high speed counter automatically counts up or down as shown on the left.

10.4.2 High speed counters and input terminal numbers

Device numbers for high speed counters correspond to those of inputs X000 to X007 as shown in the table below. X000 to X007 cannot be overlapped and used for other purposes within the same program.
For example, C235 is a 1-phase counter with one input. This counter uses X000 as its input, and does not have an interrupt reset input or interrupt start input.
C235 cannot be used with C241, C244, C246, C247, C249, C251, C252, or C254.
Another example is C255. C255 is a 2-phase counter with two inputs. This counter uses X003 as its A-phase input and X004 as its B-phase input. It also uses X005 as its interrupt reset input and X007 as its interrupt start input.

《List of high speed counter numbers》

Interrupt input	1-phase 1-counting input											1-phase 2-counting input					2-phase 2-counting input				
	C235	C236	C237	C238	C239	C240	C241	C242	C243	C244	C245	C246	C247	C248	C249	C250	C251	C252	C253	C254	C255
X000	U/D						U/D			U/D		U	U		U		A	A		A	
X001		U/D					R			R		D	D		D		B	B		B	
X002			U/D					U/D			U/D		R		R			R		R	
X003				U/D				R			R			U		U			A		A
X004					U/D				U/D					D		D			B		B
X005						U/D			R					R		R			R		R
X006										S					S					S	
X007											S					S					S

U: Up input D: Down input A: A-phase input B: B-phase input R: Reset input S: Start input

- A high speed counter input, interrupt input, and FNC56SPD instruction input cannot be overlapped.
- For an input used with a high speed counter, the setting of the input filter is automatically changed to accomodate high speed signals.

Reference

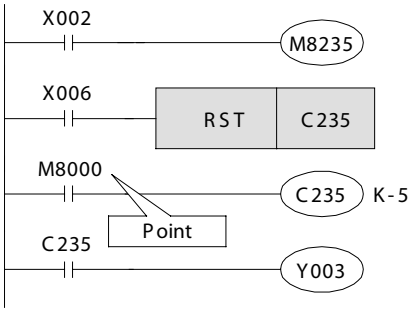
A high speed counter can be used as a "hardware counter" or "software counter". Response frequencies available for each type are as follows:

Signal \ Type	Hardware counter	Software counter *1
1-phase 1-counting input	Max. 100 kHz	Max. 40 kHz
1-phase 2-counting input	Max. 100 kHz	Max. 40 kHz
2-phase 2-counting input	Max. 50 kHz	Max. 40 kHz

*1: This assumes that the high speed counter dedicated comparison instructions, described later, are used. The response frequency varies by the type and number of instructions to be used. Also, the total value of available frequencies is defined.

10.4.3 High speed counter actions

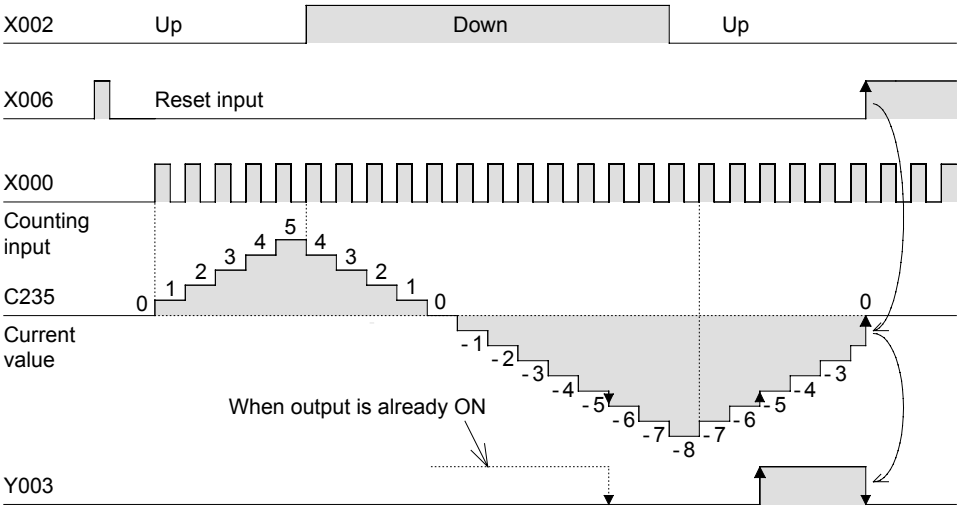
A high speed counter is a latched (battery-backed) up/down counter. One point can use 32-bit data. To specify high speed counters for applied instructions such as transfer instructions or for the arithmetic operation instructions, 32-bit instructions are used with D added.



- Select the up counter or the down counter.
  ON: Down counter, OFF: Up counter
  M8 △△△: △ specifies the counter number.
- The counter reset input is not necessary for counters with reset input (e.g. C241).
- When C235 is programmed, X000 is used for the counting input. The counter counts up or down every time X000 is turned on while the counter's OUT coil is turned on.

POINT

Do not enter the same device number for the counter drive input as that for the counting input (X000 to X007).



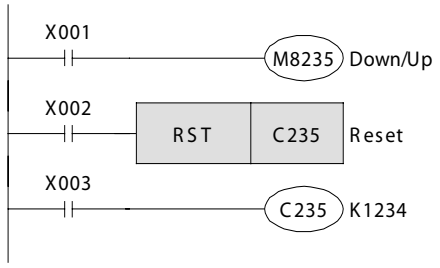
C235 counts up or down every time the counting input X000 is turned on after the C235 coil is turned on.

- The output contact is closed when the current value of the counter increases from -6 to -5. When decreases from -5 to -6, the output contact is opened.
- 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.)
- When the reset input X006 is turned on, the current value of the counter changes to 0, and the output contact is reset.
- The current value of the counter, the output contact and the reset status is latched (battery-backed).

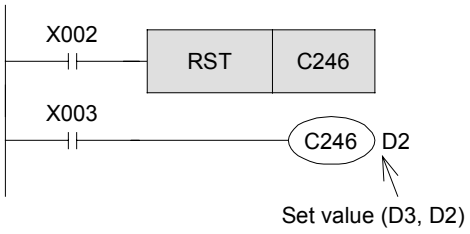
10.4.4 1-phase high speed counters

1-phase 1-input C235 to C245 (6 points or less)	1-phase 2-input C246 to C250 (2 points or less)

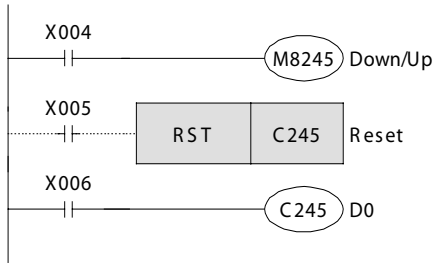
The above counters are latched (battery-backed) binary counters that use 32 bits. The output contact of the counters reacts to changes in the current value, similar to that of the previously described 32-bit counters for counting internal signals.
However, by using the high speed counting inputs, interrupt instructions are executed to count independently from the sequence operation. (The PLC provides applied instructions that are used to execute interrupt processes for output or comparison.)
Also, by selecting a counter number in a specific way, counting can be started or reset with specified interrupt inputs.



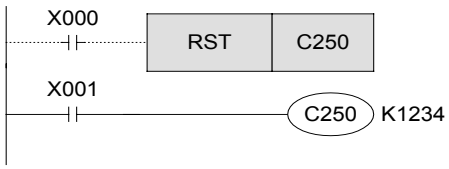
- While X003 is ON, C235 counts up or down every time the input X000 turns on.
- C235 is reset when X002 turns on.



- While X003 is ON, C246 counts up every time the input X000 turns on, and counts down every time the input X001 turns on.



- C245 counts up or down while X006 is ON. It starts counting immediately after the input X007 turns on.
- The counting input is X002. The setting value is (D1, D0) in this example.
- C245 can be reset with X005 as shown in the above sequence circuit. However, this is essentially not needed since X003 is used to reset C245.



- C250 counts up or down while X001 is ON. It starts counting immediately after the input X007 turns on. The counting input is X003 for counting up, while X004 is used for counting down.
- C250 can be reset with X000 as shown in the above sequence circuit. However, this is essentially not needed since X005 is used to reset C250.

- C △△△ counts down or up depending on whether M8 △△△ is ON or OFF.
- The counting direction (up or down) of C△△△ can be recognized by monitoring the ON/OFF status of M8△△△.

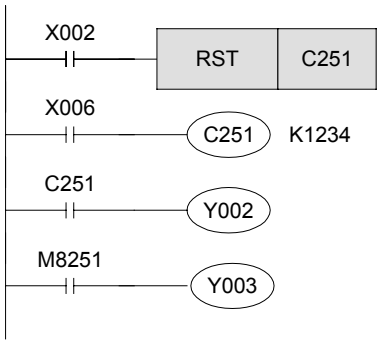
《Additional note》

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

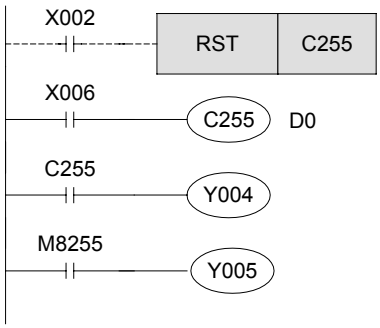
10.4.5 2-phase high speed counters

2-phase 2-input C251 to C255 (2 points or less)

- The above counter is a latched (battery-backed) binary counter that uses 32 bits. The output contact of the counter reacts to changes in the current value, similar to that of the previously described 32-bit counters for counting internal signals.
- However, by using the high speed counting inputs, interrupt instructions are executed to count independently from the sequence operation. (The PLC provides applied instructions that are used to execute interrupt processes for output or comparison.)
- Also, by selecting a counter number in a specific way, counting can be started or reset with specified interrupt inputs.
- 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) of C △△△ can be recognized by monitoring the ON/OFF status of M8 △△△.



- While X006 is ON, C251 counts up or down every time the input X000 (A-phase) or X001 (B-phase) turns on.
- This counter is reset when X002 is turned on.
- Y002 is turned on when the current value exceeds the setting value, and is turned off when the current value changes to a value below the setting value.
- Y003 is turned on (count-down) or off (count-up) according to the counting direction.



- While X006 is ON, C255 counts up or down every time X003 (A-phase) or X004 (B-phase) turns on. It starts counting immediately after the input X007 is turns on.
- This counter is reset with X002 in the left sequence circuit. It is also reset when X005 is turned on.
- Y004 is turned on when the current value exceeds the setting value (D1, D0) and turned off when the current value changes to a value below the setting value.
- Y005 is turned on (count-down) or off (count-up) according to the counting direction.

《Additional note》

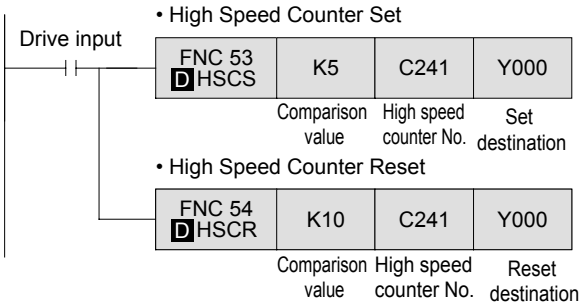
When a counting pulse is not provided, none of the high speed counter output contacts will turn ON, even if the PLC executes an instruction where "current value > set value."

10.4.6 Applied 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 applied 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.

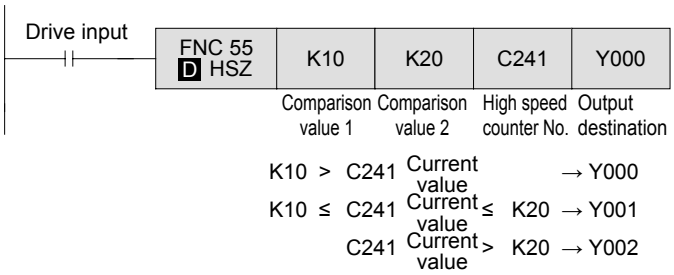
《Operation outline》

[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]



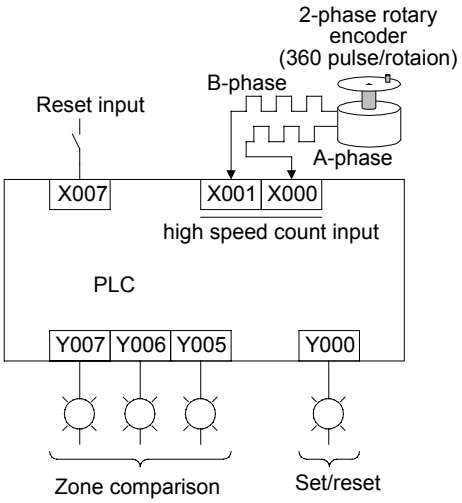
- This instruction is a High Speed Counter Zone Compare instruction.
- Three output points are controlled according to the current value of the high speed counter.

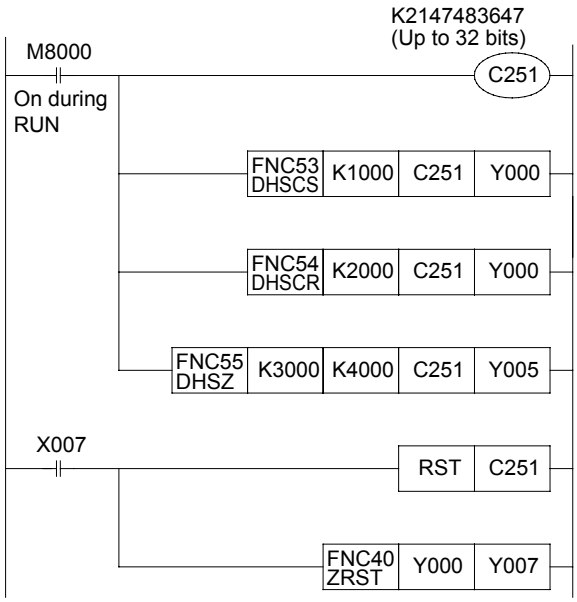
10

- Do not execute more than 32 of the FNC53 to 55 instructions simultaneously with FX3U or FX3UC PLCs (six instructions or less in other series).
  (More than 32 of the FNC53 to 55 instructions can be programmed if they are not simultaneously executed.)

《Instruction operation》

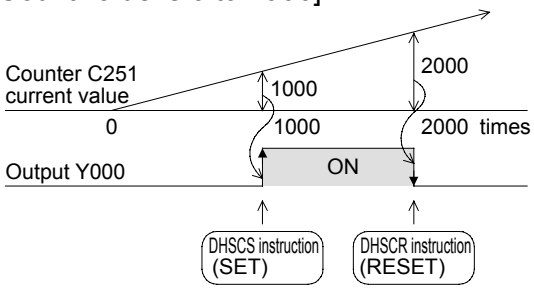
Using the training machine, create a sequence program with a high speed counter to count input signals from a 2-phase rotary encoder (high speed output device) and to operate the following outputs accordingly.



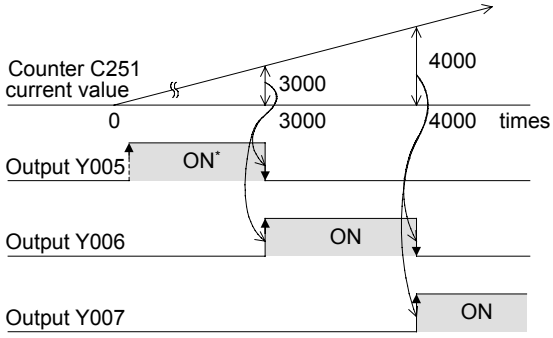


《Operation check》

[Count value is 0 to 2000]



[Count value is 3000 to 4000]



* Y005 turns on when the counter counts up from 0 to 1.

Chapter 11

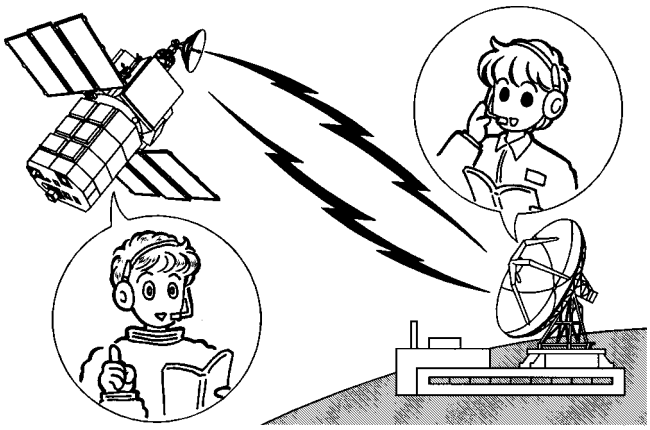
SPECIAL FUNCTION UNIT/BLOCK INSTRUCTIONS

Using special function blocks···

This chapter introduces how to communicate with equipment known as special function units/blocks. The FX Series special function units and blocks offer a range of additional features including extra communication capabilities, analog control and powerful positioning control.

11.1 Special Function Unit/Block Instructions FROM/TO

In FX_{3U} and FX_{2N} Series PLCs, it is possible to connect special function units/blocks.
The categories of special function units/blocks as referenced on the following page make it possible for the PLC to control analog signals, positioning instructions, etc. In order to execute such controls, 16-bit RAM memory called buffer memory (BFM) is built into the special function unit/block.
The intercommunication with buffer memory is executed via the FROM/TO instruction. This principle is similar to the monitoring that happens between a tower and satellite. When other operations are in progress at the same time, intercommunication gets performed as necessary.



● Control for analog signals …

The PLC conducts digital control for ON/OFF (1 or 0) signals.

Therefore, direct control is difficult to perform for things that continuously change, such as temperature, flux and air volume. For these entities, it is necessary to use a special function block for analog control.

Analog control examples
- Temperature control
- Flux control
- Speed control
- Tension control
- Pressure control
- Wind force control
- Voltage/current monitoring and control, etc.

● Control for positioning applications …

In order to control the position of a workpiece on a conveyor belt, for example, positioning operations can be used.

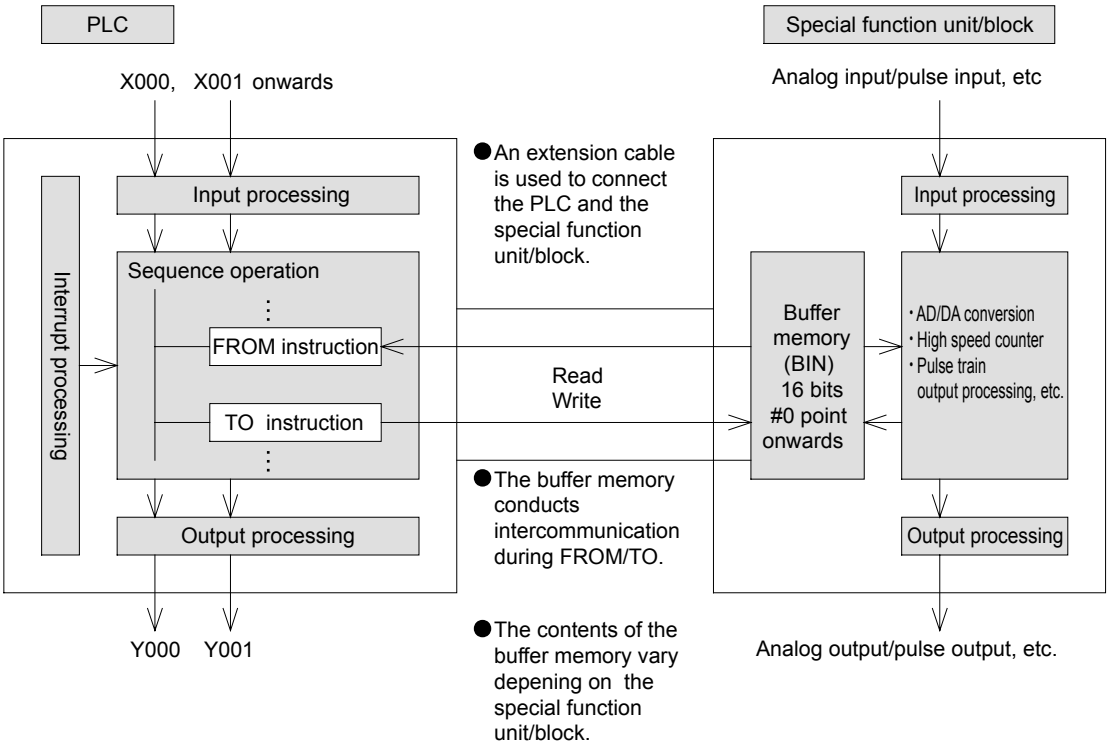
For stopping a workpiece on a conveyor belt, a sensor can be used at the stop position to stop the belt. But, for transferring objects at high speeds to stop them at specified positions, various problems can arise.

The positioning special function blocks are special-purpose modules specifically designed for this purpose; To transmit objects at high speeds and to stop them at accurate positions.

Positioning control examples
- Constant quantify feed control
- Constant multiple stage-feed control
- Rotary angle control
- Variable speed control
- 2-axis periodic control, etc.

Intercommunication with special function unit/block buffer memory is executed with FROM/TO instructions via sequence programs from the PLC.
The FROM instruction reads the current value and status information of the special function unit/block, while the T0 instruction writes in the various setting values for the special function unit/block actions.

◎ Diagram of Intercommunication between the PLC and special function unit/block



Reference

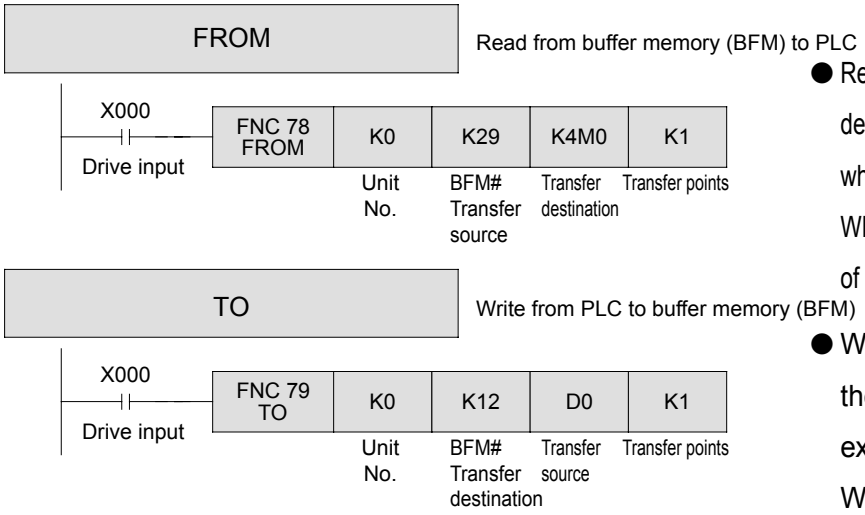
Main special function unit/block categories

The following special function units/blocks are available.

• Analog input module	FX_{2N}-2AD, 4AD, 8AD, FX_{3U}-4AD
• Analog output module	FX_{2N}-2DA, 4DA, FX_{3U}-4DA
• Analog input/output module	FX_{0N}-3A, FX_{2N}-5A
• Analog input module for Pt100 inputs	FX_{2N}-4AD-PT
• Analog input module for thermocouples	FX_{2N}-4AD-TC
• Temperature control block	FX_{2N}-2LC
• High speed counter module	FX_{2N}-1HC
• SSCNET Ⅲ compatible positioning block	FX_{3U}-20SSC-H
• Pulse train output (for 1 axis) positioning module	FX_{2N}-1PG, 10PG
• Pulse train output (for 1 axis) positioning module	FX_{2N}-10GM } Operable without PLC
• Pulse train output (for 2 axes) positioning module	FX_{2N}-20GM } Operable without PLC
• CC-Link interface block	FX_{2N}-32CCL
• CC-Link master block	FX_{2N}-16CCL-M

Including equipments for FX_{3U} and FX_{3UC} PLCs.
A main unit FX_{2N} or FX_{3U} PLC may connect up to 8 special function units/blocks.

FROM/TO instruction operation is shown as follows.

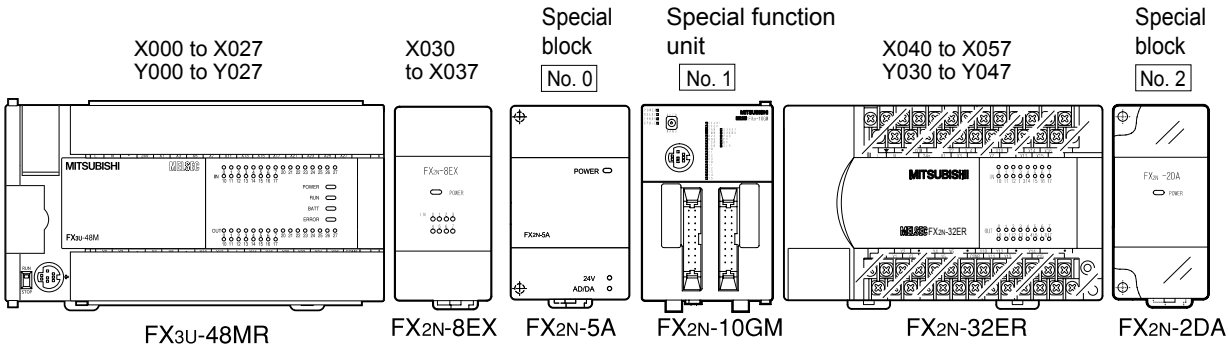


● Reads the data from the buffer memory of the designated unit number. Reading is executed when drive input is ON.
When OFF, reading is not executed and the data of the transfer destination does not change.

● Writes the data to the buffer memory of the designated unit number. Writing is executed when the drive input is ON.
When OFF, writing is not executed and the data of the transfer destination does not change.

◎ For unit numbers

● Each FX3U PLC base unit may connect up to 8 special function units/blocks. Therefore, unit numbers are distributed for intercommunication to the correct unit/block.



● Each special function unit/block is connected with an extension cable (supplied with the unit/block) on the right side of the PLC, extension unit, or extension block.
I/O numbers are distributed to the added units/blocks, and the special function units/blocks are assigned unit numbers automatically.

● Unit numbers are distributed as No.0 to No.7 from the closest one to the PLC base unit and outwards.

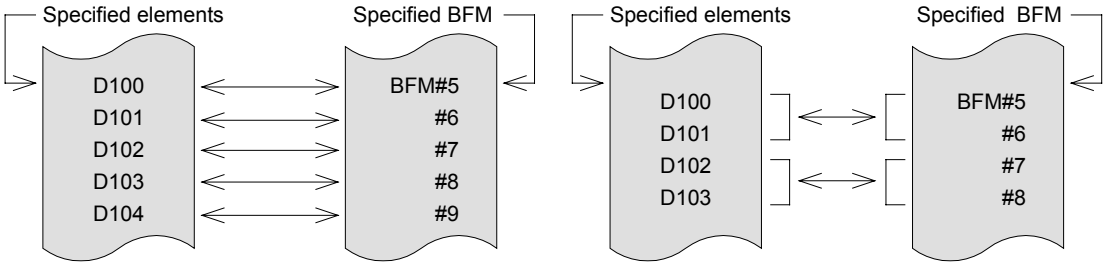
● Each special function unit/block occupies 8 I/O points (deducted from either of the inputs or outputs). However, I/O numbers are not assigned to them.
The maximum I/O points of a PLC with special function units/blocks is shown in the following formula:
Maximum I/O points = 256- occupied points number (8 points) × Number of special function units/blocks

● When a special function block is used, 5 V power is supplied from the main unit or extension unit. Therefore, the total current consumption must be below a specified value.

◎ For transfer source and destination

- TO/FROM instructions involve communication with buffer memory numbers in the special function blocks. Buffer memory addresses are 16 bit addresses whose contents and range vary depending on the special function unit/block.
- The FROM instruction to the receiving station and the TO instruction to the transfer source are instructions that both accept PLC word device inputs (such as K2M10 and K4X000 including the digit specification of bit devices).
- In some buffer memories, 32 bit data is used. In these cases, it is necessary to use 32-bit instructions from the PLC.

◎ For 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 transferred points for FROM/TO instructions is specified with n. 16-bit instructions with n=2 are identical to 32-bit instructions with n=1.

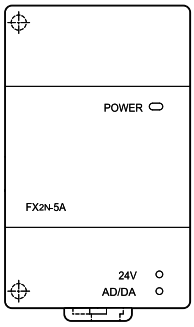
Reference

Function of special auxiliary relay M8028

- When M8028 is OFF
  All interrupts are disabled during the execution of FROM/TO instructions. The input interrupts and timer interrupts cannot be executed.
  The interrupts that occur during this period are executed immediately after the execution of the FROM/TO instruction.
  FROM/TO instructions may still be used during interrupt programs.
- When M8028 is ON
  If an interrupt occurs during the execution of a FROM/TO instruction, the FROM/TO execution stops and the interrupt program begins to run.
  However, FROM/TO instructions cannot be used during the interrupt program.

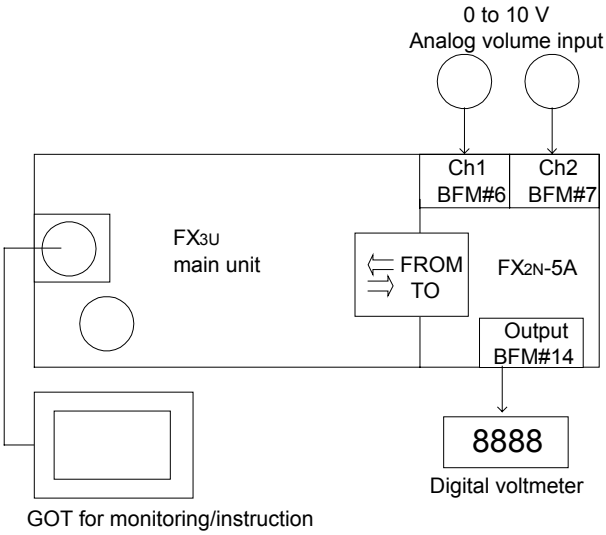
11.2 FX2N-5A application examples

《Instruction operation》



- The FX2N-5A analog input/output block has 4 analog input points and 1 analog output point.
  Let's confirm the application method of this special block using the FROM/TO instruction.
- In the following exercise for the FX2N-5A, the required configuration and buffer memory settings are as follows.

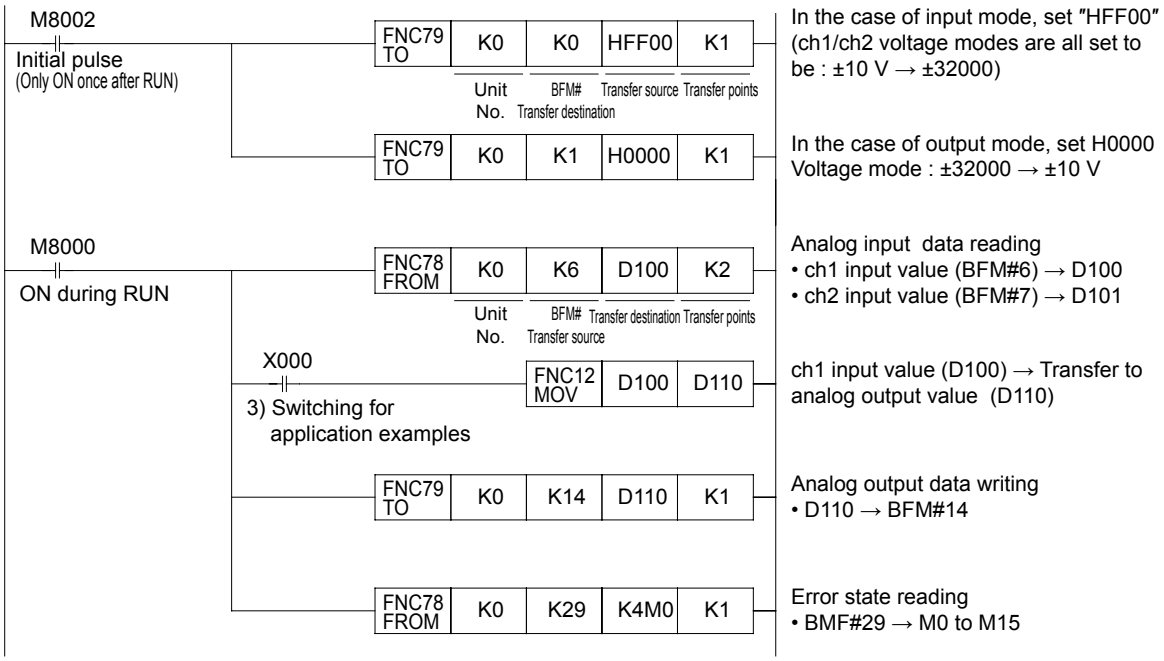
[Exercise configuration]



[Buffer memory]

BFM number	Contents	R: read W: write
#0	Input mode setting for CH1 to 04	R/W
#1	Analog output mode setting	R/W
#6	Analog input ch1 mean value (8 times) data	R
#7	Mode input ch2 mean value (8 times) data	R
#14	Analog output setting data	R/W
#29	Error status	R

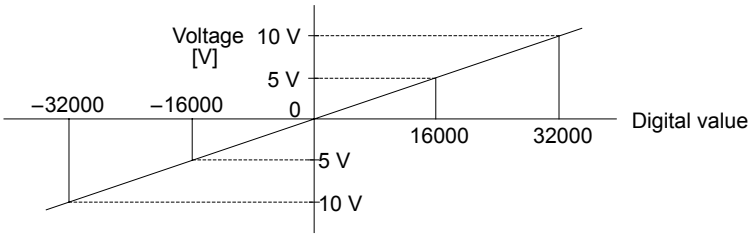
- For this exercise, only the required BFM numbers have been extracted. Do not use the other numbers.
- Adjustment for analog I/O characteristics is assumed to be completed and is omitted from this exercise program.



《Operation check》

Use GX Developer to monitor D100,D101 and D100.

1) Rotate the knob of ch1/ch2 and confirm on the GOT screen that the value of [0 to 32000] is input.
2) Operate the GOT screen, designate the instruction value of the output within ±32000 and confirm whether the voltage is output.

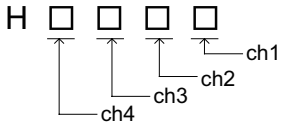


3) Application example
When X000 is set to ON, the ch1 volume input value (0 to 32000) can be used as the analog output instruction (0 to 10 V output).

Reference

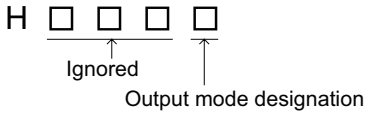
Buffer memory setting and contents

1) [BFM#0: Input mode setting]
Set ch1 and ch2 to be used in "Voltage input mode (±10 V → ±32000)". (Set value: HFF00)
• Setting method: Each digit of the 4-digit hexadecimal number is defined in the figure below.



(□setting example)
□=0: voltage input mode (±10 V → ±32000)
□=1: current input module (4 mA-20 mA → 0-32000)
□=F: does not use channel.
.
.
.
Possible to choose various input modes.

2) [BFM#1: Analog output mode setting]
Sets the output mode of the analog output to "Voltage output mode (±32000 → ±10 V)". (Set value: H0000)
• Setting method: The number is designated in hexadecimal format.



(□setting example)
□=0 : Voltage output mode (±32000 → ±10 V)
□=1 : Current output mode (±2000 → ±10 V)
B to F: cannot be set
.
.
.
Possible to choose various output modes.

3) [BFM#6 to 7: Mean value data of analog input]
Inputs the voltage value of ch1 or ch2 input "0 to 10 V" to a numerical value from "0 to 32000" (including the mean value after reading for 8 times).

4) [BFM#14: Set data of analog output]
• Writes the value of the analog output voltage (±10 V) to a numerical value of "±32000".

5) [BFM#29: Error status]
• Contains the error contents of the FX_{2N}-5A. The output program in this exercise is set to be [b0 to b15 → M0 to M15].

Bit number	Contents
b0 (M0)	With error
b1 (M1)	OFFSET/GAIN set value error
b2 (M2)	Power supply error
b3 (M3)	Hardware error
b4 (M4)	A/D converted value error
b5 (M5)	D/A converted value error
b6 (M6)	NA
b7 (M7)	NA
b8 (M8)	Set value error
b9 (M9)	Input/output mode setting error
b10 (M10)	Average number of times setting error
b11 (M11)	Conducted the change of input/output characteristic while the change of input/output characteristic is prohibited.
b12 (M12)	Abrupt testing set value error
b13 (M13)	Up/down limit testing set value error
b14 (M14)	Filter setting error
b15 (M15)	Internal operation function setting error

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.

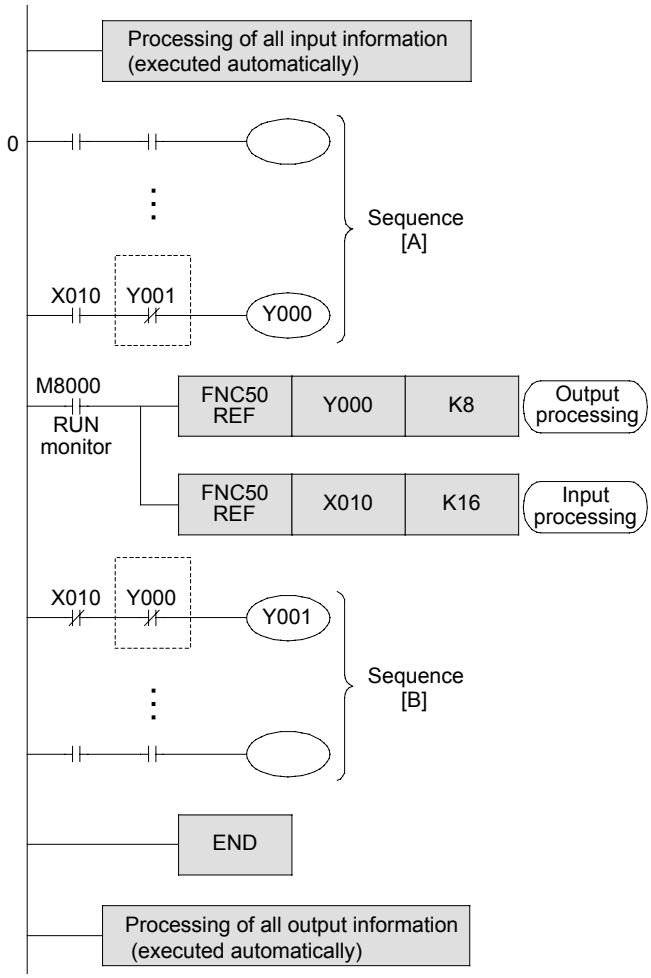
In this chapter···

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.

《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 Y000 to Y007 is output. (The refresh points number must be set to be a multiple of 8.)
- At this phase, the input information of the 16 input points from X010 to X027 is stored into the memory. (The refresh points number must be set to be a multiple of 8.)

- 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

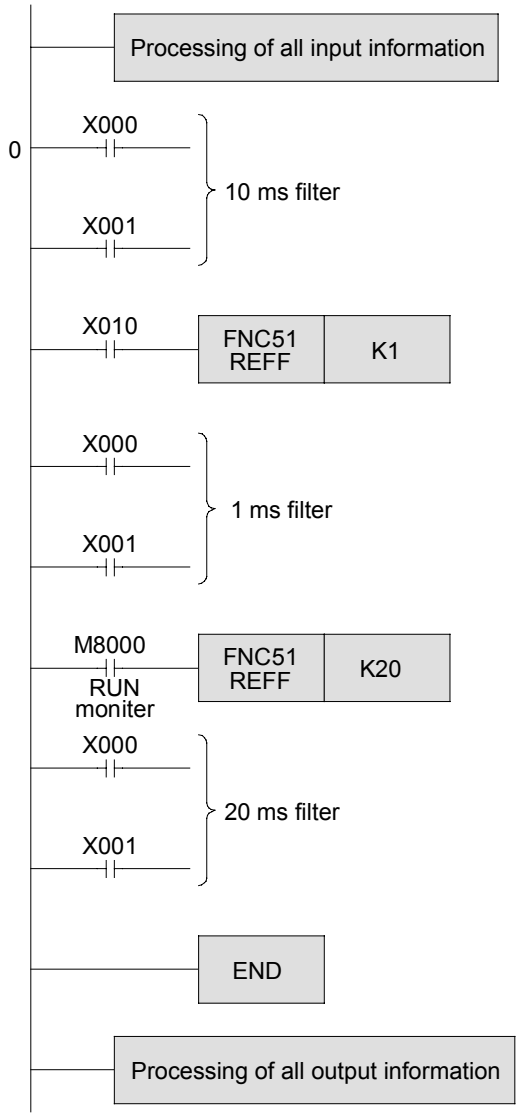
Output interlock

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 Y000 and Y001, the outputs may be activated simultaneously.

12.2 Input filter adjustment instructions (REFF)

- 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.
- The FX PLC adopts a digital filter for inputs X000 to X017 (X000 to X007 in 16-point type basic units), and allows for the instructions to change values of the input filters between 0 to 60 ms. However, since the inputs also have the smallest C-R filter, the minimum value varies between 5 μs to 200 μs, depending on the model or the input terminal number.
- In addition, the value of the input filter is changed to the minimum value in the processing of these instructions if: an interrupt pointer is used, X000 to X007 are used with high speed counters, or the FNC56 SPD instruction is used.

《Operation outline》



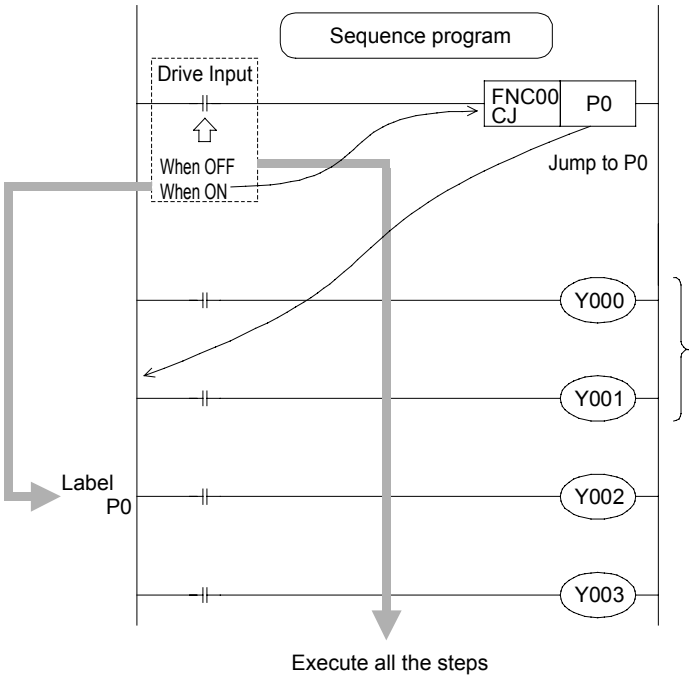
- For general input processing, X000 to X017 use a 10 ms filter.
- When X010 is ON, the inputs use a 1 ms filter. When this instruction is executed, X000 to X017 are refreshed.
- When X010 is OFF, the inputs use a 10 ms filter, and are processed with the same ON/OFF information that was used in the initial "input processing".
- After this instruction is executed, the inputs X000 to X017 are refreshed with a filter constant of 20 ms.
- In one operation cycle, the filter constant can be changed between 0 to 60 ms as many times as needed.

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.

When the drive input is ON, these steps are skipped and not executed.
The specific contents will be covered later on.

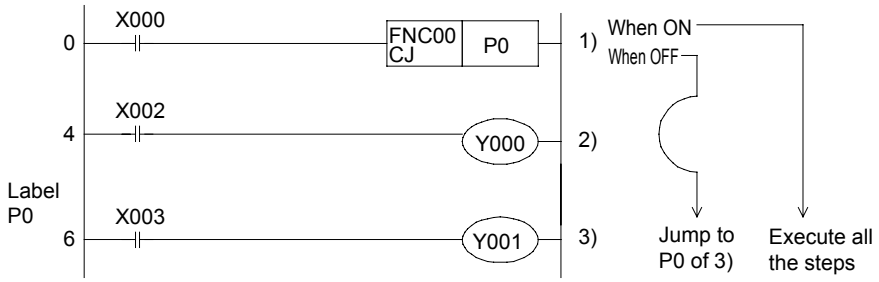
Reference

Pointer numbers

- The pointer numbers for each PLC model are as follows:
  - FX_{1S} : P0 to P62
  - FX_{1N}, FX_{2N}, FX_{1NC}, FX_{2NC} : P0 to P62
    P64 to P127
  - FX_{3U}, FX_{3UC} : P0 to P62
    P64 to P4095
  - If "P63" is specified, the program jumps to END.
- Lable numbers are also used by the CALL instruction described later. The numbers cannot be overlapped.

《Instruction operation》

Let's confirm the execution of the jump instruction.



《Operation check》

Use GX Developer to monitor the circuit.

When X000 is OFF

1) [Turn off X000]

⇩

2) Y000 turns on or off when X002 is turned on or off.

⇩

3) Y001 turns on or off when X003 is turned on or off.

When X000 is ON

1) [Turn on X000]

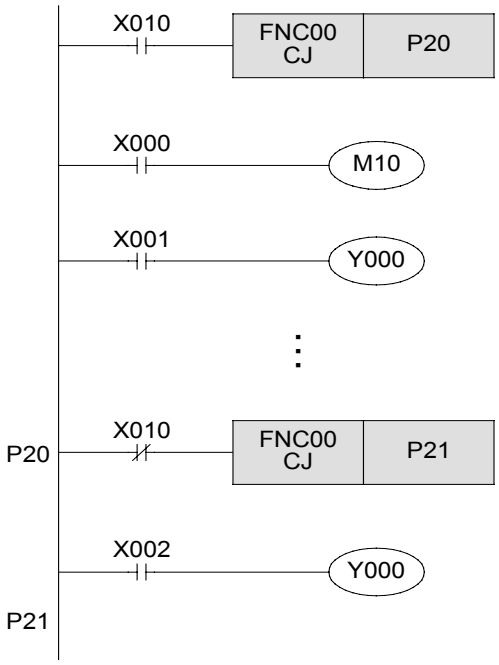
⇩

2) Y000 does turn on or off when X002 is turned on or off.
(The program at 2) is skipped by the jump instruction.)

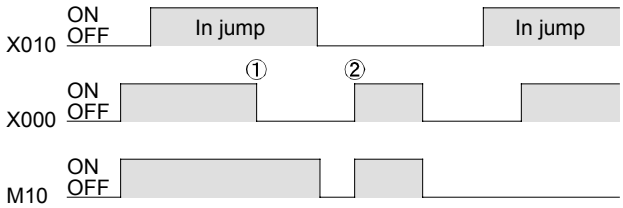
⇩

3) Y001 turns on or off when X003 is turned on or off.
(During a jump to the label P0)

◎ Functional principles of Y, M, S coils



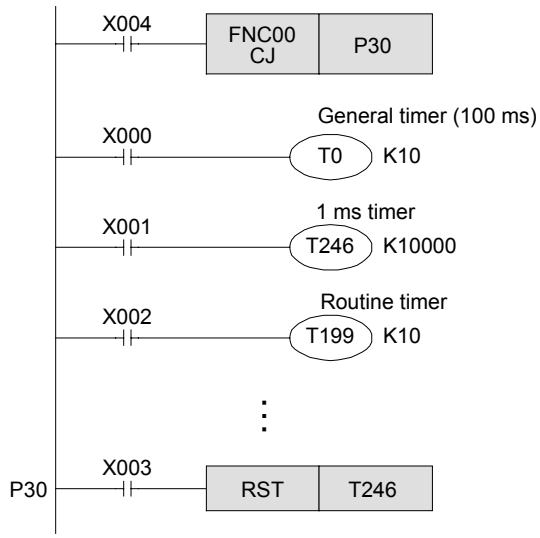
- When the coil of a Y (output), M (auxiliary relay), or S (state) is skipped, the ON/OFF status that was stored prior to the jump will be maintained.



1) Even if X00 turns off, M10 stays on.
2) Without jump instruction, M10 turns on or off according to the status of X000.

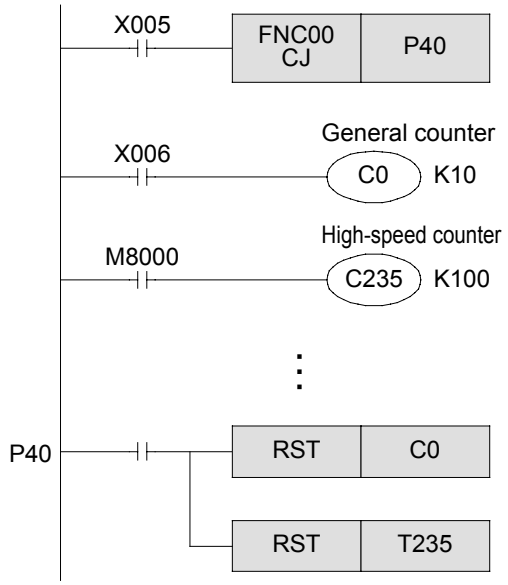
- The output Y000 is a dual coil. When X010 = OFF, the program runs according to the status of X001. When X010 = ON, the program runs according to the status of X002.
- Even with a dual coil, if one of the two is skipped, only the other can be activated. As a result, the coils may be activated separately.

◎ 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 1 ms timer (T246 to T249) and the routine timer (T192 to T199) resume 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 while that of the 1 ms timer does not open. The output resumes after the jump instruction is deactivated.
- If the reset instruction for a retentive timer (T246 to T255) 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.
- When a high speed counter (C235 to C255) starts counting, however, it will resume counting even when its coil is skipped by a jump instruction. The output contact will remain closed.
- 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 applied instructions

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

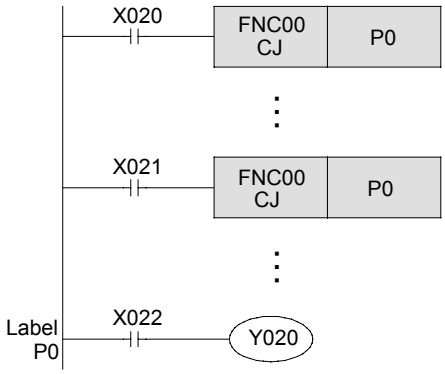
  However, the high-speed processing instructions from FNC52 to FNC58 and other applied instructions will be executed intermittently

FNC52:	MTR	FNC56	:	SPD
FNC53:	HSCS	FNC57	:	PLSY
FNC54:	HSCR	FNC58	:	PWM
FNC55:	HSZ			

Reference

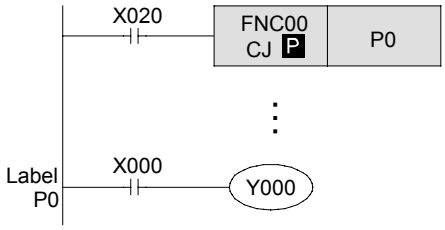
Jump methods

[Jump to the same pointer]



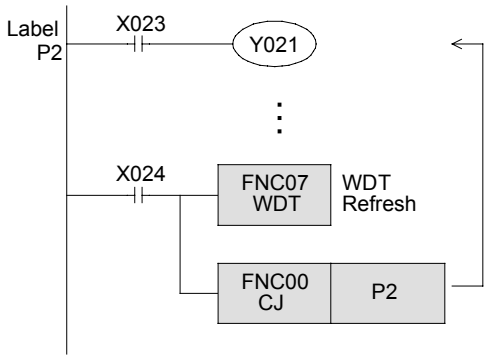
- The pointer number in the operands to the left have the same number.
- When X020 is ON, the program will jump to label P0 from here. When X020 is OFF but X021 is ON, the program will jump to label P0 from CJ of X021.
- Duplicate label numbers (including labels for CALL instructions) cannot be used. If two label numbers are the same, a PLC error will result.

[Jump in one operational cycle]



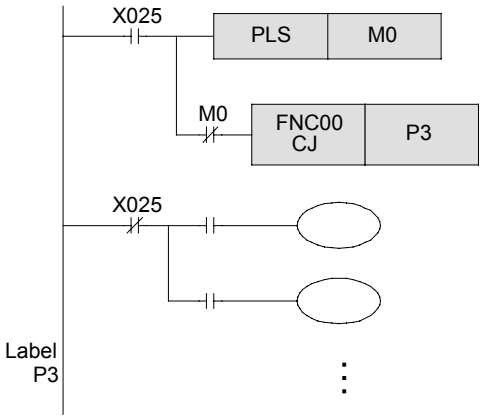
- The CJ P is used. If the drive input changes from OFF to ON , the program jumps to label P0 in one operational cycle.

[Jump upward]



- Although it is possible to program the jump label at a step number lower than that of its CJ command, if X024 is turned on for more than 200 ms (the set time of the PLC's watch dog timer, stored in D8000), a watch dog timer error may occur. If this error occurs, the PLC will stop (CPU error).
- In the above case, it is required to write a longer time for the watch dog timer in special data register D8000. Or as shown in the left figure, program an instruction that refreshes the watch dog timer.

[Delay jump by one operation cycle]

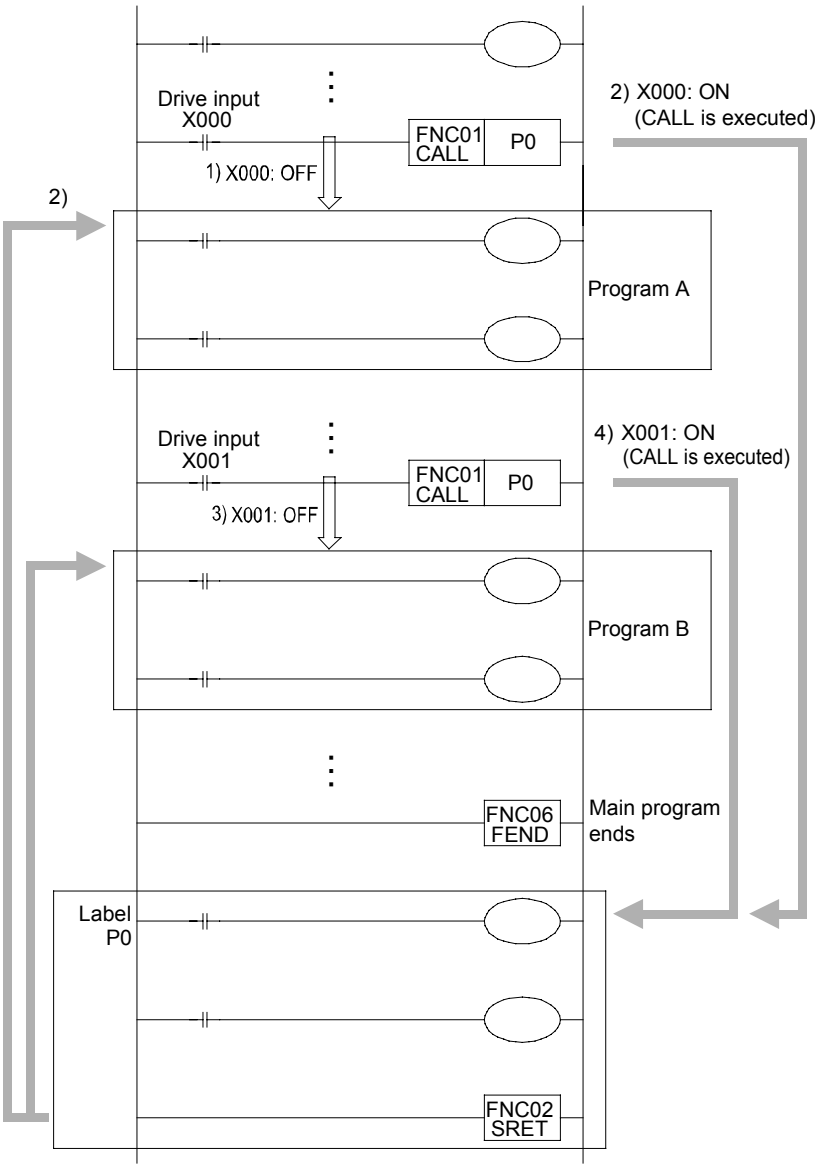


- In the circuit to the left, if X025 turns ON, the jump to P3 will be turned on in the second operational cycle.
  During the first operation cycle, all the outputs between CJP3 and P3 are turned off.

12.4 Call subroutine instructions (CALL, SRET)

The CALL Subroutine instruction is an instruction to execute a subroutine program whithin 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.

《Operation outline》

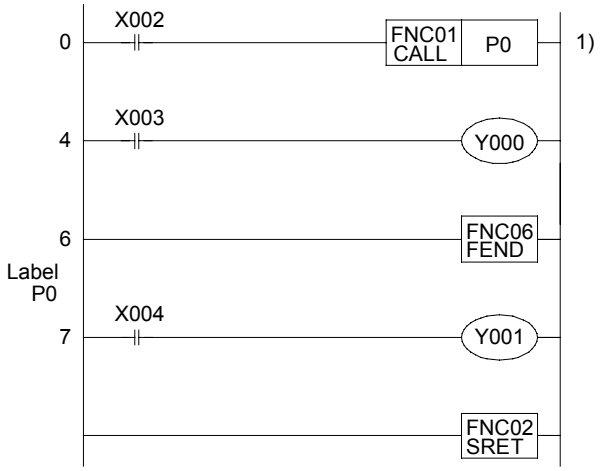


1) When X000 is "OFF", "Program A" will be executed.
2) When X000 is "ON", "Program A" will be executed after the subroutine program of "label P0" is executed.
3) When X001 is "OFF", "Program B" will be executed.
4) When X001 is "ON", "Program B" will be executed after the subroutine program of "label 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 label numbers include P0 to P62, and P64 to P127. The same 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 T192 to T199, and 1 ms timers T246 to T249.

《Instruction operation》

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



《Operation check》

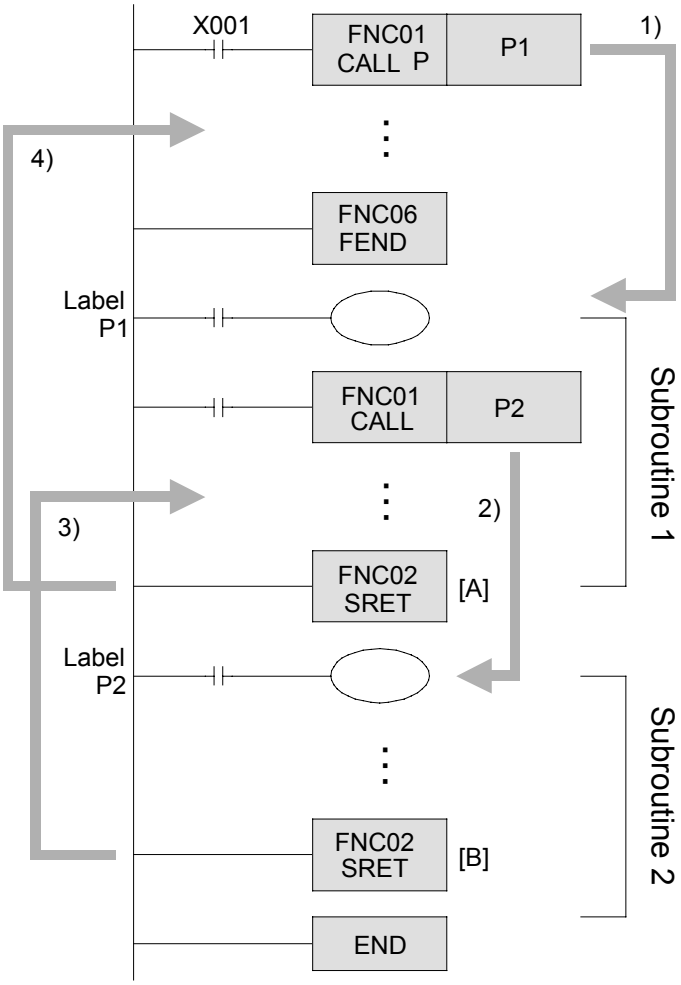
Use GX Developer to monitor the circuit.

1) [When X002 is OFF]
   - Y000 turns on or off when X003 is turned on or off.
   - Y001 does not turn on or off when X004 is turned on or off (since the subroutine was not called).

2) [When X002 is ON]
   - Y000 turns on or off when X003 is turned on or off.
   - Y001 turns on or off when X004 is turned on or off (since the subroutine was called).

[CALL instruction nesting (multi-nesting)]

If a CALL instruction is programmed within a subroutine , it is known as multi-nesting. This type of CALL instruction can be programmed 4 times at most. i.e., fivefold nesting is allowed as a whole.



1) When the CALL P instruction is used, The CALL instruction is executed only when the input X001 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 SRET instruction [B], the program will return to the next step of CALL P2.
4) Similarly, when the operation moves to SRET instruction [A], program will return to the next instruction of CALL P1. The numbers available for labels are P0 to P62, and P64 to P127. The same number must not be used including labels for CJ instructions.

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. The actions of the timer and counter in a subroutine are the same as those for the jump instruction. Refer to the previous pages.

- Available pointer numbers differ as follows:
  - FX_{1S} : P0 to P62
  - FX_{1N}, FX_{2N}, FX_{1NC}, FX_{2NC} : P0 to P62
    P64 to P127
    The program will jump to "END" if [p63] is specified.
  - FX_{3U}, FX_{3UC} : P0 to P62
    P64 to P4095

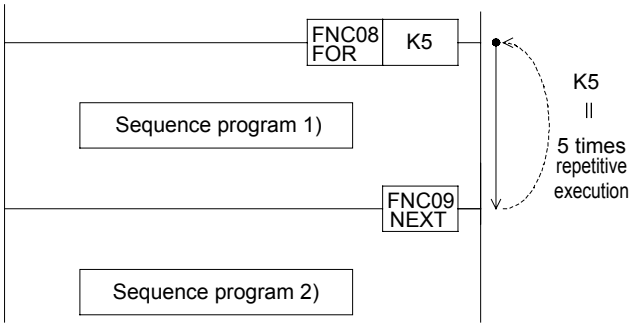
- Label numbers are also used by the CJ instruction described previously. The numbers cannot be overlapped.

- In the P-SRET instruction, it is not allowed to use MC-MCR, STL-RET, I-IRET and another P-SRET instructions for programming.
  In addition, the P-SRET instruction cannot be used in MC-MCR, FOR-NEXT, STL-RET 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.
When n is set to 1 to 32767 times, specifying "n = -32768 to 0" is equal to specifying "n = 1".

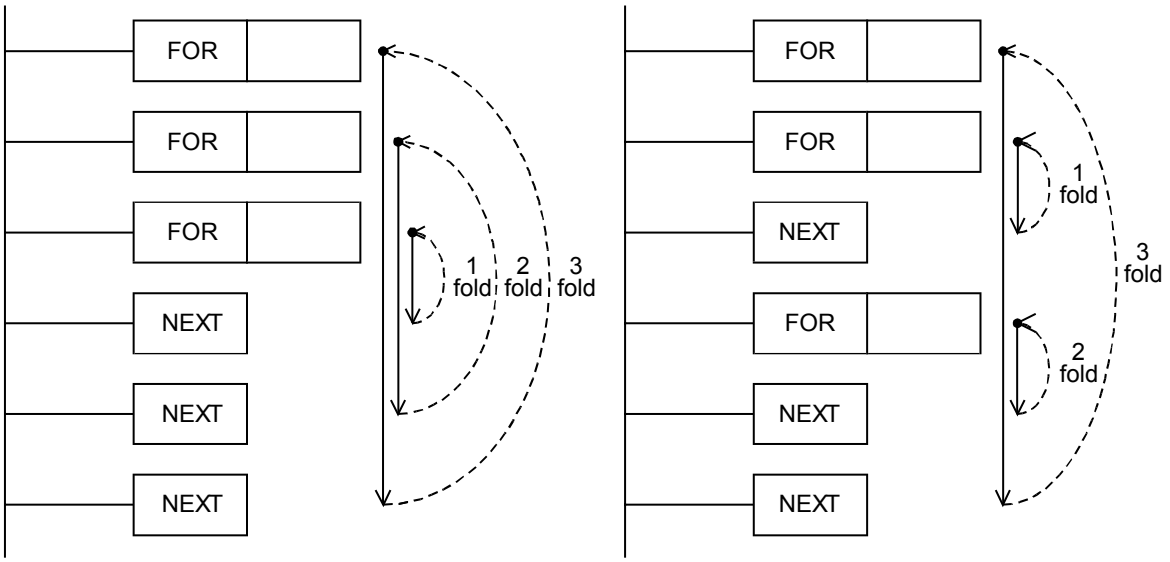
《Operation outline》



- 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

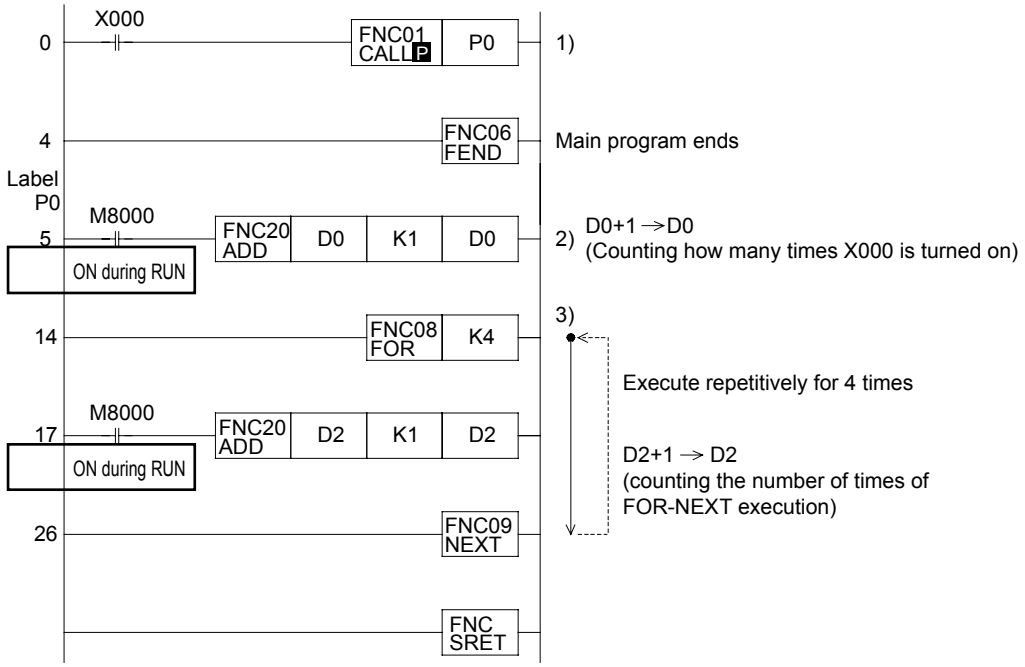
- About nesting (multi-nesting)
  When FOR-NEXT instructions are used between FOR-NEXT instructions for nesting programs, fivefold nesting is allowed at most.



If the numbers for the 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 Developer to monitor the circuit.

1) [Set X000 to ON.] ⇨ ● [CALLP P0] instruction will be executed, and the subroutine at label P0 will be executed in only one operation cycle.

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

3) ⇨ ● [ADD D0 K1 D0] instruction is executed repetitively for four times by the [FORK4] instruction.

● As a result of the execution, [4] is stored into D2.
(Then, 4 is added to D2 every time X000 is turned on.)

Since the execution ends in a split second, switch X000 between ON and OFF, and try repeating the action.

12

MEMO

Chapter 15

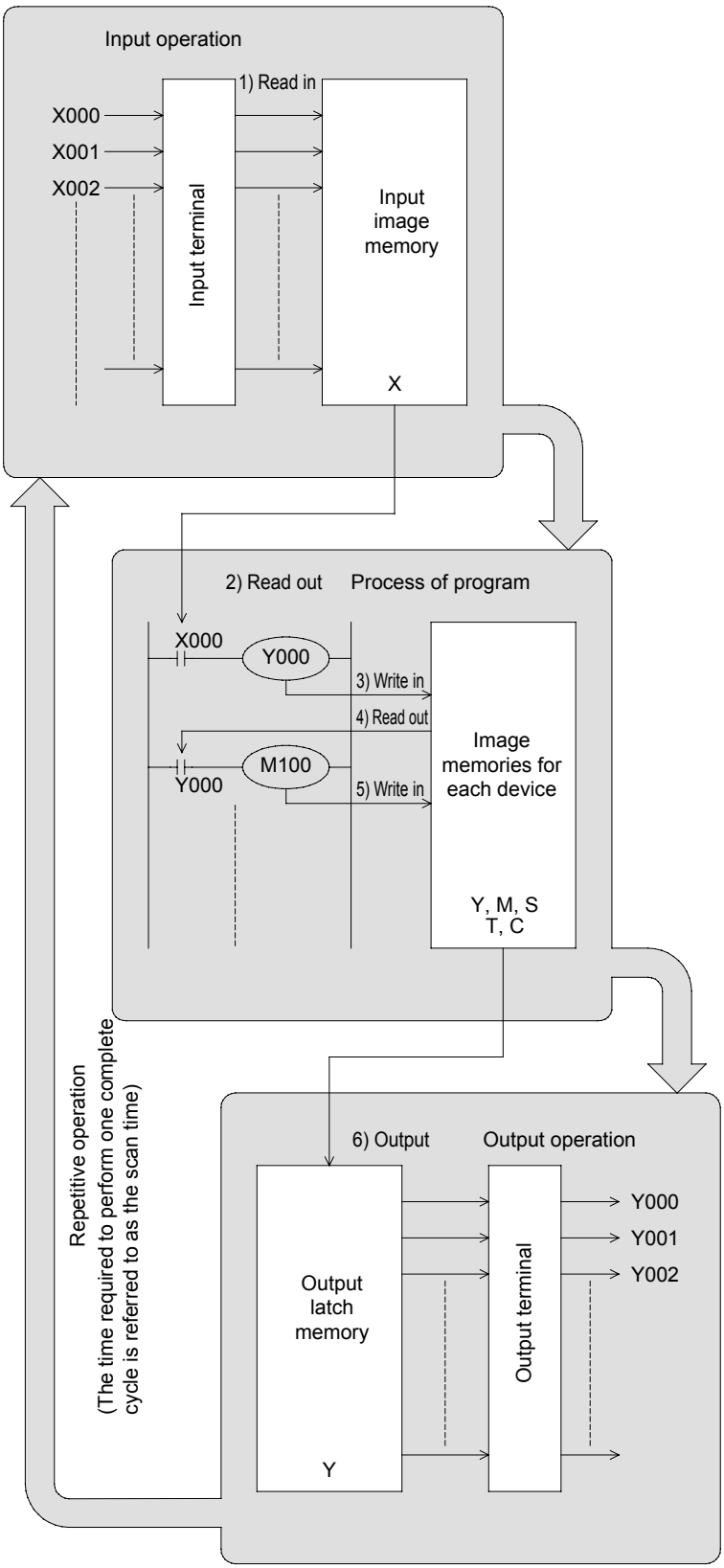
SUMMARY OF THE POINTS

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 auxiliary 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**

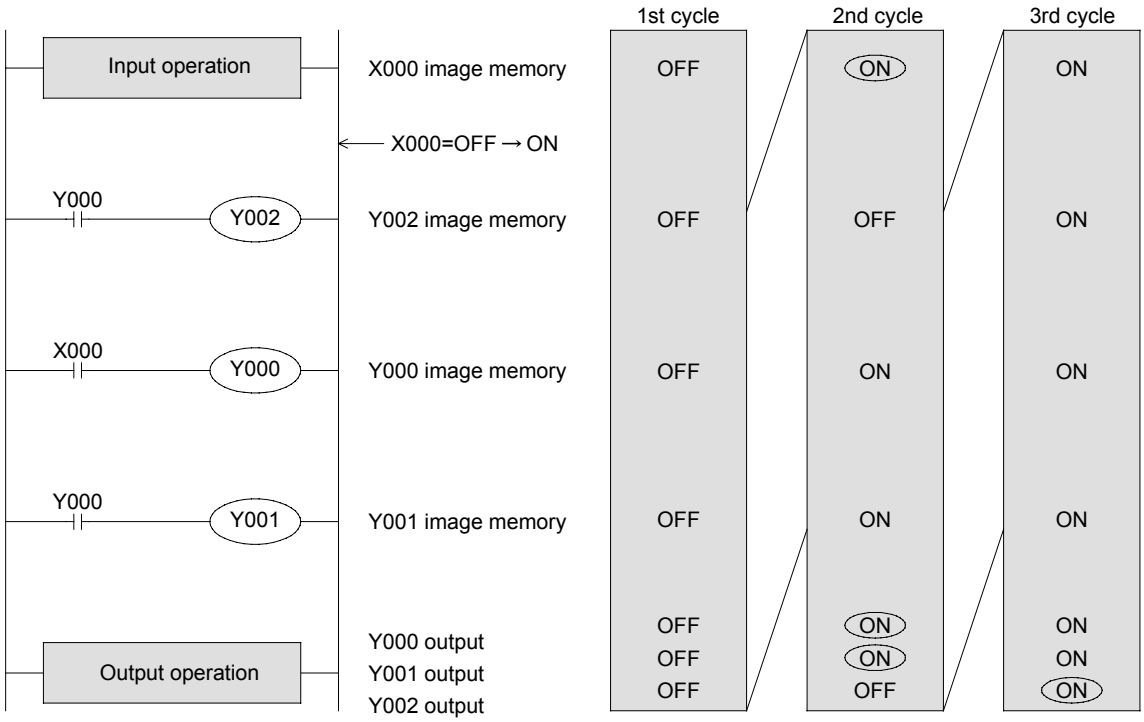
  The PLC reads the ON/OFF statuses of required elements from the input image memory or other element's memory, in accordance with the contents of the instructions stored in the program memory. Hence, the image memory of each element can sequentially change its content in accordance with the progress of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

15.2 Response delay for Input/Output

There are not only electrical delays from input filters (approx. 10ms) and mechanical response delays from output relays (approx. 10ms), but there is also a response delay due to the affect of the scan time.
For example, assume that input X000 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.)



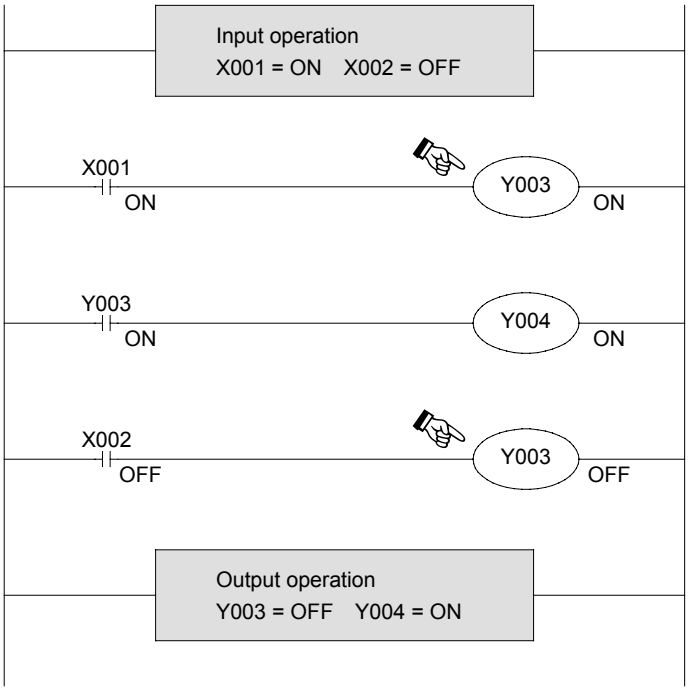
As shown above, Y000 and Y001 may have up to a 2-cycle response delay. (The output contact will be ON approximately 10 ms later.)
Y002 is energized 1 cycle later, since the contact Y000 which drives Y002 is programmed before X000.

Reference

Countermeasures against I/O response delay

There is an application instruction (Refresh FNC50 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. Additionally, for inputs X000 to X007, there are applications instructions (Refresh and filter adjust FNC51 REFF) that can be used to reduce the response delay by filtering the inputs within the program.

15.3 Dual output operation



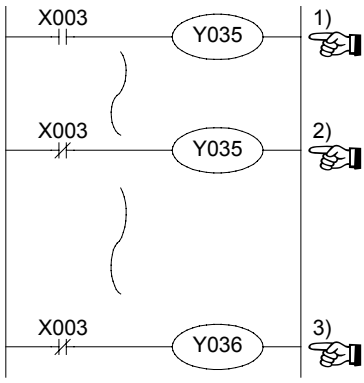
In this example, assume that coil Y003 is used in multiple locations.
Also, assume that X001 = ON and X002 = OFF.
For the first Y003, the image memory is set to ON because X001 is ON, hence output Y004 is also ON.
However, for the second Y003, the image memory is changed to OFF since input X002 is OFF.
Therefore, the output is in fact, Y003 = OFF and Y004 = ON.

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

Reference

Direct I/O Mode

There are PLCs which perform I/O processes at the same instant that the instructions are executed (direct I/O mode).
In this case, the output responses for inputs become faster, but the action for dual output is different.
In addition, for the following circuit, Y035 and Y036 may change to ON at the same time.



Assume that X003 = ON when executing this circuit. Thus, Y035 is energized.

If X003 is not changed until executing this line of code, Y035 may become de-energized. Hence, Y035 can be ON or OFF, which may cause problems for the dual output during the scan time.

If the input X003 is changed from ON to OFF before executing this line, Y036 will be energized. If X003 line 2) is turned OFF then both Y035 and Y036 will turn to ON.

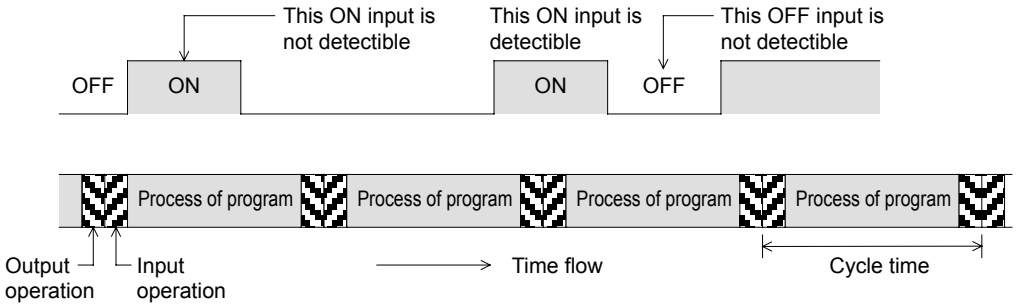
15.4 No limitation on the number of contacts

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 for series and parallel contacts. Other limitations include the display area of the graphic programming panel and the number of printed characters that can be included on a single line.

- At the maximum, 11 contacts and 1 coil (in the case for timers or counters, 10 contacts and 1 coil) can be included in a single line on the GPP.
  The number of lines running in parallel from a single relay must be 24 or less.
- It is recommended that each line include no more than 10 contacts and 1 coil and that lines be limited to 5 or less, in order to easily display them on the GPP.
  If this programming method is followed, 1 circuit block with comments can be displayed on a single display area.

Reference

Narrow pulses are not detectible

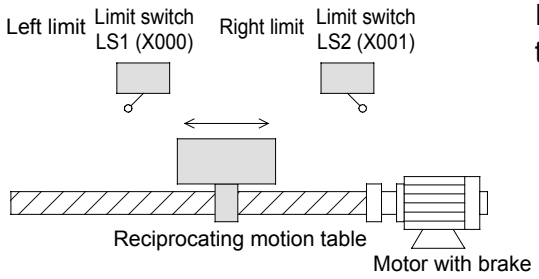


In principle, the widths of input pulses (ON and OFF inputs) for a PLC must be longer than the cycle time of the PLC.
When the response delay of the input filter, 10 ms, is taken into account, the time durations for ON and OFF inputs must be 20 ms at the shortest, if the cycle time is 10 ms.
Hence, input pulses with frequencies of more 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 The role of the battery

The PLC incorporates a non-rechangeable lithium battery which provides backup utility for auxiliary relays, states of devices in the ladder instructions, timers, and counters as well as the backup of program memory during power failure.
Moreover, some PLCs without a built-in battery are latched by EEPROM memory.

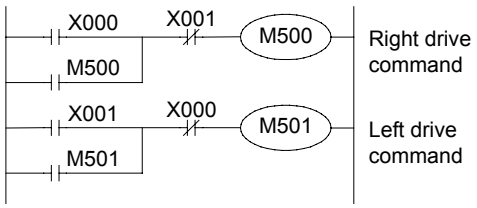
How to use latched (battery backed) relays



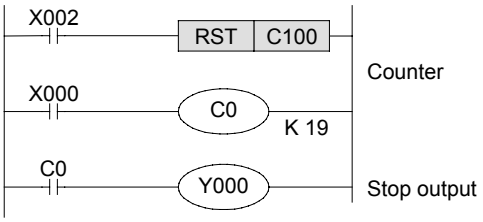
In some cases, it is required to move the positioning table in the same direction after power failure.

X000=ON(Left limit)
→ M500(Right drive command)=ON
→ Power failure
→ Positioning table stops halfway
→ Restart(M500=ON)
→ X001=ON(Right limit)
→ M500=OFF, M501=ON
→ Left drive

Relays from M500 are backed up with the battery, and often referred to as latched (battery backed) relays.

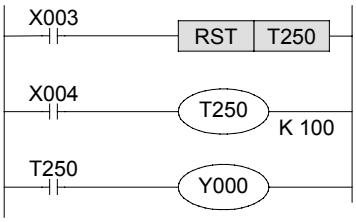


How to use counters



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.

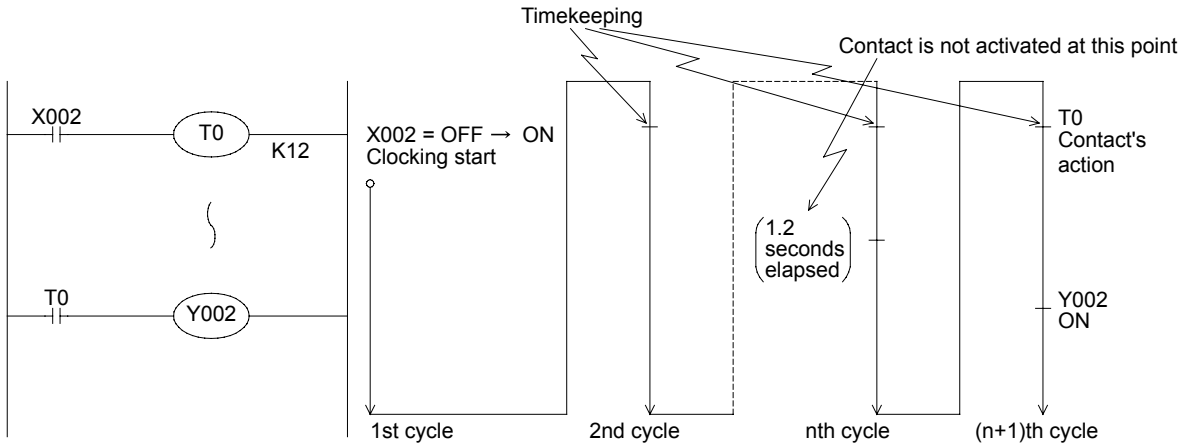
How to use retentive timers (FX2N, FX3U, FX2NC, FX3UC Series)



The timer T250 starts timing at the moment X004 changes to ON.
The timer keeps its current value if X004 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 T250 will then be energized.
When X003 is set to ON, the current value of the timer is cleared and reset to 0 and the output contact is set to OFF.

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:

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

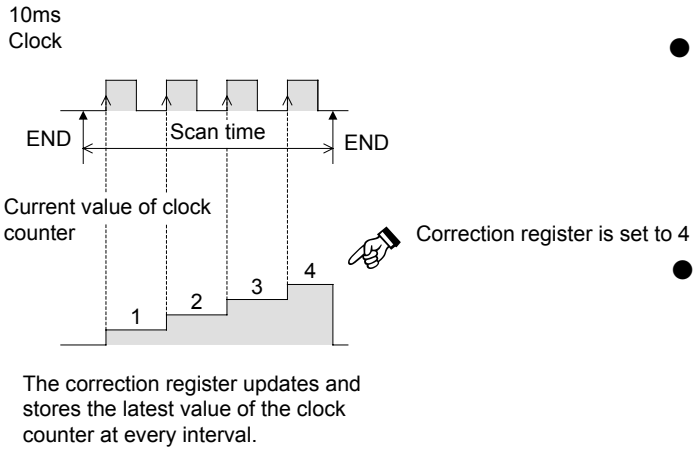
α : 0.01 for 10ms timer, 0.1 for 100 ms timer (seconds)
T : Set time of timer (seconds)
T0 : Scan time (seconds)

In the worst case, if the timer drive contact is programmed prior to the timer coil, the accuracy becomes 2T0. If the set value of the timer is 0, the output contact will be activated at the next execution of the coil instruction.

Reference

Details on timer operation

- For 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, in the case where a 10 ms timer is used in a program with a scan time of 30 ms, the following procedures are carried out.



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

15

MEMO

Appendix

In this appendix···

The applied instructions, special auxiliary relays and special data registers that have been described in this textbook are arranged in lists.
In addition, the contents of the parameter settings are included. Set them when necessary.

It's easy to learn PLCs!

For someone who is just beginning to learn PLCs or does not know anything about sequence control, the learning process can take a long time.
The FX Series Micro PLCs facilitate an easy way to learn basic sequencing knowledge.
Along with this book, training seminars and PLC training software are available for learning PLCs at the appropriate pace.

Appendix 1.1 Application instruction list

The application instructions for the Micro PLCs are listed in the following tables.

Classification	FNC No.	Mnemonic	Function	Applicable PLC						
				FX1S	FX1N	FX2N	FX3U	FX1NC	FX2NC	FX3UC
Program Flow	00	CJ	Conditional Jump	○	○	○	○	○	○	○
	01	CALL	Call Subroutine	○	○	○	○	○	○	○
	02	SRET	Subroutine Return	○	○	○	○	○	○	○
	03	IRET	Interrupt Return	○	○	○	○	○	○	○
	04	EI	Enable Interrupt	○	○	○	○	○	○	○
	05	DI	Disable Interrupt	○	○	○	○	○	○	○
	06	FEND	Main Routine Program End	○	○	○	○	○	○	○
	07	WDT	Watchdog Timer Refresh	○	○	○	○	○	○	○
	08	FOR	Start a FOR/NEXT Loop	○	○	○	○	○	○	○
	09	NEXT	End a FOR/NEXT Loop	○	○	○	○	○	○	○
Move and Compare	10	CMP	Compare	○	○	○	○	○	○	○
	11	ZCP	Zone Compare	○	○	○	○	○	○	○
	12	MOV	Move	○	○	○	○	○	○	○
	13	SMOV	Shift Move	—	—	○	○	—	○	○
	14	CML	Complement	—	—	○	○	—	○	○
	15	BMOV	Block Move	○	○	○	○	○	○	○
	16	FMOV	Fill Move	—	—	○	○	—	○	○
	17	XCH	Exchange	—	—	○	○	—	○	○
	18	BCD	Conversion to Binary Coded Decimal	○	○	○	○	○	○	○
	19	BIN	Conversion to Binary	○	○	○	○	○	○	○
Arithmetic and Logical Operation (+, -, ×, ÷)	20	ADD	Addition	○	○	○	○	○	○	○
	21	SUB	Subtraction	○	○	○	○	○	○	○
	22	MUL	Multiplication	○	○	○	○	○	○	○
	23	DIV	Division	○	○	○	○	○	○	○
	24	INC	Increment	○	○	○	○	○	○	○
	25	DEC	Decrement	○	○	○	○	○	○	○
	26	WAND	Logical Word AND	○	○	○	○	○	○	○
	27	WOR	Logical Word OR	○	○	○	○	○	○	○
	28	WXOR	Logical Exclusive OR	○	○	○	○	○	○	○
	29	NEG	Negation	—	—	○	○	—	○	○
Rotation and Shift Operation	30	ROR	Rotation Right	—	—	○	○	—	○	○
	31	ROL	Rotation Left	—	—	○	○	—	○	○
	32	RCR	Rotation Right with Carry	—	—	○	○	—	○	○
	33	RCL	Rotation Left with Carry	—	—	○	○	—	○	○
	34	SFTR	Bit Shift Right	○	○	○	○	○	○	○
	35	SFTL	Bit Shift Left	○	○	○	○	○	○	○
	36	WSFR	Word Shift Right	—	—	○	○	—	○	○
	37	WSFL	Word Shift Left	—	—	○	○	—	○	○
	38	SFWR	Shift Write [FIFO/FILO Control]	○	○	○	○	○	○	○
	39	SFRD	Shift Read [FIFO Control]	○	○	○	○	○	○	○
Data Operation	40	ZRST	Zone Reset	○	○	○	○	○	○	○
	41	DECO	Decode	○	○	○	○	○	○	○
	42	ENCO	Encode	○	○	○	○	○	○	○
	43	SUM	Sum of Active Bits	—	—	○	○	—	○	○
	44	BON	Check Specified Bit Status	—	—	○	○	—	○	○
	45	MEAN	Mean	—	—	○	○	—	○	○
	46	ANS	Timed Annunciator Set	—	—	○	○	—	○	○
	47	ANR	Annunciator Reset	—	—	○	○	—	○	○
	48	SQR	Square Root	—	—	○	○	—	○	○
	49	FLT	Conversion to Floating Point	—	—	○	○	—	○	○
High Speed Processing	50	REF	Refresh	○	○	○	○	○	○	○
	51	REFF	Refresh and Filter Adjust	—	—	○	○	—	○	○
	52	MTR	Input Matrix	○	○	○	○	○	○	○
	53	HSCS	High Speed Counter Set	○	○	○	○	○	○	○
	54	HSCR	High Speed Counter Reset	○	○	○	○	○	○	○

Classification	FNC No.	Mnemonic	Function	Applicable PLC						
				FX1S	FX1N	FX2N	FX3U	FX1NC	FX2NC	FX3UC
High Speed Processing	55	HSZ	High Speed Counter Zone Compare	—	—	○	○	—	○	○
	56	SPD	Speed Detection	○	○	○	○	○	○	○
	57	PLSY	Pulse Y Output	○	○	○	○	○	○	○
	58	PWM	Pulse Width Modulation	○	○	○	○	○	○	○
	59	PLSR	Acceleration/Deceleration Setup	○	○	○	○	○	○	○
Handy Instruction	60	IST	Initial State	○	○	○	○	○	○	○
	61	SER	Search a Data Stack	—	—	○	○	—	○	○
	62	ABSD	Absolute Drum Sequencer	○	○	○	○	○	○	○
	63	INCD	Incremental Drum Sequencer	○	○	○	○	○	○	○
	64	TTMR	Teaching Timer	—	—	○	○	—	○	○
	65	STMR	Special Timer	—	—	○	○	—	○	○
	66	ALT	Alternate State	○	○	○	○	○	○	○
	67	RAMP	Ramp Variable Value	○	○	○	○	○	○	○
	68	ROTC	Rotary Table Control	—	—	○	○	—	○	○
	69	SORT	SORT Tabulated Data	—	—	○	○	—	○	○
External FX I/O Device	70	TKY	Ten Key Input	—	—	○	○	—	○	○
	71	HKY	Hexadecimal Input	—	—	○	○	—	○	○
	72	DSW	Digital Switch (Thumbwheel Input)	○	○	○	○	○	○	○
	73	SEGD	Seven Segment Decoder	—	—	○	○	—	○	○
	74	SEGL	Seven Segment With Latch	○	○	○	○	○	○	○
	75	ARWS	Arrow Switch	—	—	○	○	—	○	○
	76	ASC	ASCII Code Data Input	—	—	○	○	—	○	○
	77	PR	Print (ASCII Code)	—	—	○	○	—	○	○
	78	FROM	Read From A Special Function Block	—	○	○	○	○	○	○
	79	TO	Write To A Special Function Block	—	○	○	○	○	○	○
External FX Device	80	RS	Serial Communication	○	○	○	○	○	○	○
	81	PRUN	Parallel Run (Octal Mode)	○	○	○	○	○	○	○
	82	ASCI	Hexadecimal to ASCII Conversion	○	○	○	○	○	○	○
	83	HEX	ASCII to Hexadecimal Conversion	○	○	○	○	○	○	○
	84	CCD	Check Code	○	○	○	○	○	○	○
	85	VRRD	Volume Read	○	○	○	—	—	—	—
	86	VRSC	Volume Scale	○	○	○	—	—	—	—
	87	RS2	Serial Communication 2	—	—	—	○	—	—	○
	88	PID	PID Control Loop	○	○	○	○	○	○	○
	89									
*1	102	ZPUSH	Batch Store of Index Register	—	—	—	□	—	—	□
	103	ZPOP	Batch POP of Index Register	—	—	—	□	—	—	□
Floating Point	110	ECMP	Floating Point Compare	—	—	○	○	—	○	○
	111	EZCP	Floating Point Zone Compare	—	—	○	○	—	○	○
	112	EMOV	Floating Point Move	—	—	—	○	—	—	○
	116	ESTR	Floating Point to Character String Conversion	—	—	—	○	—	—	○
	117	EVAL	Character String to Floating Point Conversion	—	—	—	○	—	—	○
	118	EBCD	Floating Point to Scientific Notation Conversion	—	—	○	○	—	○	○
	119	EBIN	Scientific Notation to Floating Point Conversion	—	—	○	○	—	○	○
	120	EADD	Floating Point Addition	—	—	○	○	—	○	○
	121	ESUB	Floating Point Subtraction	—	—	○	○	—	○	○
	122	EMUL	Floating Point Multiplication	—	—	○	○	—	○	○
	123	EDIV	Floating Point Division	—	—	○	○	—	○	○
	124	EXP	Floating Point Exponent	—	—	—	○	—	—	○
	125	LOGE	Floating Point Natural Logarithm	—	—	—	○	—	—	○
	126	LOG10	Floating Point Common Logarithm	—	—	—	○	—	—	○
	127	ESQR	Floating Point Square Root	—	—	○	○	—	○	○
	128	ENEG	Floating Point Negation	—	—	—	○	—	—	○
	129	INT	Floating Point to Integer Conversion	—	—	○	○	—	○	○

Classification	FNC No.	Mnemonic	Function	Applicable PLC						
				FX1S	FX1N	FX2N	FX3U	FX1NC	FX2NC	FX3UC
Floating Point	130	SIN	Floating Point Sine	—	—	○	○	—	○	○
	131	COS	Floating Point Cosine	—	—	○	○	—	○	○
	132	TAN	Floating Point Tangent	—	—	○	○	—	○	○
	133	ASIN	Floating Point Arc Sine	—	—	—	○	—	—	○
	134	ACOS	Floating Point Arc Cosine	—	—	—	○	—	—	○
	135	ATAN	Floating Point Arc Tangent	—	—	—	○	—	—	○
	136	RAD	Floating Point Degree to Radian Conversion	—	—	—	○	—	—	○
	137	DEG	Floating Point Radian to Degree Conversion	—	—	—	○	—	—	○
Data Operation 2	140	WSUM	Sum of Word Data	—	—	—	○	—	—	□
	141	WTOB	WORD to BYTE	—	—	—	○	—	—	□
	142	BTOW	BYTE to WORD	—	—	—	○	—	—	□
	143	UNI	4-bit Linking of Word Data	—	—	—	○	—	—	□
	144	DIS	4-bit Grouping of Word Data	—	—	—	○	—	—	□
	147	SWAP	Byte Swap	—	—	○	○	—	○	○
	149	SORT2	Sort Tabulated Data 2	—	—	—	○	—	—	□
Positioning Control	150	DSZR	DOG Search Zero Return	—	—	—	○	—	—	○
	151	DVIT	Interrupt Positioning	—	—	—	○	—	—	○
	152	TBL	Batch Data Positioning Mode	—	—	—	□	—	—	□
	155	ABS	Absolute Current Value Read	○	○	◎	○	○	◎	○
	156	ZRN	Zero Return	○	○	—	○	○	—	○
	157	PLSV	Variable Speed Pulse Output	○	○	—	○	○	—	○
	158	DRVI	Drive to Increment	○	○	—	○	○	—	○
	159	DRVA	Drive to Absolute	○	○	—	○	○	—	○
Real Time Clock Control	160	TCMP	RTC Data Compare	○	○	○	○	○	○	○
	161	TZCP	RTC Data Zone Compare	○	○	○	○	○	○	○
	162	TADD	RTC Data Addition	○	○	○	○	○	○	○
	163	TSUB	RTC Data Subtraction	○	○	○	○	○	○	○
	164	HTOS	Hour to Second Conversion	—	—	—	○	—	—	○
	165	STOH	Second to Hour Conversion	—	—	—	○	—	—	○
	166	TRD	Read RTC data	○	○	○	○	○	○	○
	167	TWR	Set RTC data	○	○	○	○	○	○	○
	169	HOUR	Hour Meter	○	○	◎	○	○	◎	○
External Device	170	GRY	Decimal to Gray Code Conversion	—	—	○	○	—	○	○
	171	GBIN	Gray Code to Decimal Conversion	—	—	○	○	—	○	○
	176	RD3A	Read form Dedicated Analog Block	—	○	◎	○	○	◎	○
	177	WR3A	Write to Dedicated Analog Block	—	○	◎	○	○	◎	○
*2	180	EXTR	External ROM Function (FX2N/FX2NC)	—	—	◎	—	—	◎	—
Others	182	COMRD	Read Device Comment Data	—	—	—	○	—	—	□
	184	RND	Random Number Generation	—	—	—	○	—	—	○
	186	DUTY	Timing Pulse Generation	—	—	—	○	—	—	□
	188	CRC	Cyclic Redundancy Check	—	—	—	○	—	—	○
	189	HCMOV	High Speed Counter Move	—	—	—	○	—	—	○
Block Data Operation	192	BK+	Block Data Addition	—	—	—	○	—	—	□
	193	BK-	Block Data Subtraction	—	—	—	○	—	—	□
	194	BKCMP=	Block Data Compare (S1) = (S2)	—	—	—	○	—	—	□
	195	BKCMP>	Block Data Compare (S1) > (S2)	—	—	—	○	—	—	□
	196	BKCMP<	Block Data Compare (S1) < (S2)	—	—	—	○	—	—	□
	197	BKCMP< >	Block Data Compare (S1) ≠ (S2)	—	—	—	○	—	—	□
	198	BKCMP< =	Block Data Compare (S1) ≦ (S2)	—	—	—	○	—	—	□
	199	BKCMP> =	Block Data Compare (S1) ≧ (S2)	—	—	—	○	—	—	□
Character String Control	200	STR	BIN to Character String Conversion	—	—	—	○	—	—	□
	201	VAL	Character String to BIN Conversion	—	—	—	○	—	—	□
	202	$+	Link Character Strings	—	—	—	○	—	—	○
	203	LEN	Character String Length Detection	—	—	—	○	—	—	○
	204	RIGHT	Extracting Character String Data from the Right	—	—	—	○	—	—	○
	205	LEFT	Extracting Character String Data from the Left	—	—	—	○	—	—	○

Classification	FNC No.	Mnemonic	Function	Applicable PLC						
				FX1S	FX1N	FX2N	FX3U	FX1NC	FX2NC	FX3UC
Character String Control	206	MIDR	Random Selection of Character Strings	—	—	—	○	—	—	○
	207	MIDW	Random Replacement of Character Strings	—	—	—	○	—	—	○
	208	INSTR	Character string search	—	—	—	○	—	—	□
	209	$MOV	Character String Transfer	—	—	—	○	—	—	○
Data Operation 3	210	FDEL	Deleting Data from Tables	—	—	—	○	—	—	□
	211	FINS	Inserting Data to Tables	—	—	—	○	—	—	□
	212	POP	Shift Last Data Read [FILO Control]	—	—	—	○	—	—	○
	213	SFR	Bit Shift Right with Carry	—	—	—	○	—	—	○
	214	SFL	Bit Shift Left with Carry	—	—	—	○	—	—	○
Data Comparison	224	LD=	Load Compare (S1) = (S2)	○	○	○	○	○	○	○
	225	LD>	Load Compare (S1) > (S2)	○	○	○	○	○	○	○
	226	LD<	Load Compare (S1) < (S2)	○	○	○	○	○	○	○
	228	LD<>	Load Compare (S1) ≠ (S2)	○	○	○	○	○	○	○
	229	LD<=	Load Compare (S1) ≦ (S2)	○	○	○	○	○	○	○
	230	LD>=	Load Compare (S1) ≧ (S2)	○	○	○	○	○	○	○
	232	AND=	AND Compare (S1) = (S2)	○	○	○	○	○	○	○
	233	AND>	AND Compare (S1) > (S2)	○	○	○	○	○	○	○
	234	AND<	AND Compare (S1) < (S2)	○	○	○	○	○	○	○
	236	AND<>	AND Compare (S1) ≠ (S2)	○	○	○	○	○	○	○
	237	AND<=	AND Compare (S1) ≦ (S2)	○	○	○	○	○	○	○
	238	AND>=	AND Compare (S1) ≧ (S2)	○	○	○	○	○	○	○
	240	OR=	OR Compare (S1) = (S2)	○	○	○	○	○	○	○
	241	OR>	OR Compare (S1) > (S2)	○	○	○	○	○	○	○
	242	OR<	OR Compare (S1) < (S2)	○	○	○	○	○	○	○
	244	OR<>	OR Compare (S1) ≠ (S2)	○	○	○	○	○	○	○
	245	OR<=	OR Compare (S1) ≦ (S2)	○	○	○	○	○	○	○
	246	OR>=	OR Compare (S1) ≧ (S2)	○	○	○	○	○	○	○
Data Table Operation	256	LIMIT	Limit Control	—	—	—	○	—	—	○
	257	BAND	Dead Band Control	—	—	—	○	—	—	○
	258	ZONE	Zone Control	—	—	—	○	—	—	○
	259	SCL	Scaling (Coordinate by Point Data)	—	—	—	○	—	—	○
	260	DABIN	Decimal ASCII to BIN Conversion	—	—	—	○	—	—	□
	261	BINDA	BIN to Decimal ASCII Conversion	—	—	—	○	—	—	□
	269	SCL2	Scaling 2 (Coordinate by X/Y Data)	—	—	—	○	—	—	◇
Inverter Communication	270	IVCK	Inverter Status Check	—	—	—	○	—	—	○
	271	IVDR	Inverter Drive	—	—	—	○	—	—	○
	272	IVRD	Inverter Parameter Read	—	—	—	○	—	—	○
	273	IVWR	Inverter Parameter Write	—	—	—	○	—	—	○
	274	IVBWR	Inverter Parameter Block Write	—	—	—	○	—	—	○
*3	278	RBFM	Divided BFM Read	—	—	—	○	—	—	□
	279	WBFM	Divided BFM Write	—	—	—	○	—	—	□
*4	280	HSCT	High Speed Counter Compare With Data Table	—	—	—	○	—	—	○
Extension File Register Control	290	LOADR	Load From ER	—	—	—	○	—	—	○
	291	SAVER	Save to ER	—	—	—	○	—	—	○
	292	INITR	Initialize R and ER	—	—	—	○	—	—	○
	293	LOGR	Logging R and ER	—	—	—	○	—	—	○
	294	RWER	Rewrite to ER	—	—	—	○	—	—	◇
	295	INITER	Initialize ER	—	—	—	○	—	—	◇

◎: Supported by version 3.00 or later
◇: Supported by version 1.30 or later
□: Supported by version 2.20 or later

*1: Data transfer 1 *3: Data transfer 3
*2: Extension Function *4: High Speed Processing 2

A

Appendix 1.2 Main special device list

Special devices are used to operate the built-in PLC functions that are factory prepared and ready to use. There are special auxiliary relays and special data registers.
The following tables list the main special devices for FX PLCs.

● PC status

Number	Name	FX1S	FX1N	FX2N	FX3U	FX1NC	FX2NC	FX3UC
[M]8000	RUN monitor NO contact	○	○	○	○	○	○	○
[M]8001	RUN monitor NC contact	○	○	○	○	○	○	○
[M]8002	Initial pulse NO contact	○	○	○	○	○	○	○
[M]8003	Initial pulse NC contact	○	○	○	○	○	○	○
[M]8004	Error occurrence	○	○	○	○	○	○	○
[M]8005	Battery voltage low	–	–	○	○	–	○	○
[M]8006	Battery error latch	–	–	○	○	–	○	○
[M]8007*1	Momentary power failure	–	–	○	○	–	○	○
[M]8008*1	Power failure detected	–	–	○	○	–	○	○
[M]8009	24V DC down	–	–	○	○	–	○	○

Number	Name	FX1S	FX1N	FX2N	FX3U	FX1NC	FX2NC	FX3UC
D 8000	Watchdog timer	200 ms	200 ms	200 ms	200 ms	200 ms	200 ms	200 ms
[D]8001	PLC type and system version	22	26	24	24	26	24	24
[D]8101	PLC type and system version	–	–	–	16	–	–	16
[D]8002	Memory capacity	○ In the case of 16 K steps or more, D8002 becomes "K8" while "16" and "64" are input to D8102.						
[D]8003*2	Memory type	○	○	○	○	○	○	○
[D]8004	Error number M	○	○	○	○	○	○	○
[D]8005	Battery voltage	–	–	○	○	–	○	○
[D]8006	Low battery voltage detection level	–	–	○	○	–	○	○
[D]8007	Momentary power failure count	–	–	○	○	–	○	○
D 8008*2	Power failure detection	–	–	○	○	–	○	○
[D]8009	24V DC failed device	–	–	○	○	–	○	○

*1: Changes of the power failure detection (D8008)

*2: Contents of the memory type

● Clock

Number	Name	FX1S	FX1N	FX2N	FX3U	FX1NC	FX2NC	FX3UC
[M]8010								
[M]8011	10 ms clock pulse	○	○	○	○	○	○	○
[M]8012	100 ms clock pulse	○	○	○	○	○	○	○
[M]8013	1 sec clock pulse	○	○	○	○	○	○	○
[M]8014	1 min clock pulse	○	○	○	○	○	○	○
M 8015	Clock stop and preset	○	○	○	○	○	○	○
M 8016	Time read display is stopped	○	○	○	○	○	○	○
M 8017	±30 seconds correction	○	○	○	○	○	○	○
[M]8018	Installation detection	○ (Always ON)						
M 8019	Real time clock (RTC) error	○	○	○	○	○	○	○

Number	Name	FX1S	FX1N	FX2N	FX3U	FX1NC	FX2NC	FX3UC
[D]8010	Present scan time	○ The indicated value includes the waiting time of the constant scan driven by M8039 as well.						
[D]8011	Minimum scan time							
[D]8012	Maximum scan time							
D 8013	Second data	○	○	○	○	○	○	○
D 8014	Minute data	○	○	○	○	○	○	○
D 8015	Hour data	○	○	○	○	○	○	○
D 8016	Day data	○	○	○	○	○	○	○
D 8017	Month data	○	○	○	○	○	○	○
D 8018	Year data	○	○	○	○	○	○	○
D 8019	Day-of-the-week data	○	○	○	○	○	○	○

The time data of D8013 to D8019 is latched (battery-backed). In addition, D8018 (year data) can be switched to a four-digit year mode from 1980 to 2079.

● Flag

Number	Name	FX1S	FX1N	FX2N	FX3U	FX1NC	FX2NC	FX3UC
[M]8020	Zero	○	○	○	○	○	○	○
[M]8021	Borrow	○	○	○	○	○	○	○
M 8022	Carry	○	○	○	○	○	○	○
[M]8023								
M 8024	BMOV direction specification	–	○	○	○	○	○	○
M 8025	HSC mode	–	–	○	○	–	○	○
M 8026	RAMP mode	–	–	○	○	–	○	○
M 8027	PR mode	–	–	○	○	–	○	○
M 8028	100 ms/10 ms timer changeover	○	–	–	–	–	–	–
M 8028	Interrupt permission during FROM/TO instruction execution	–	–	○	○	–	○	○
[M]8029	Instruction execution complete	○	○	○	○	○	○	○

Number	Name	FX1S	FX1N	FX2N	FX3U	FX1NC	FX2NC	FX3UC
D 8020	Input filter adjustment	○	○	○	○	○	○	○
[D]8021								
[D]8022								
[D]8023								
[D]8024								
[D]8025								
[D]8026								
[D]8027								
[D]8028	Value of Z0 (Z) register*3	○	○	○	○	○	○	○

*3: The contents in Z1 to Z7 and V1 to V7 are stored in D8182 to D8195.

● PC mode

Number	Name	FX1S	FX1N	FX2N	FX3U	FX1NC	FX2NC	FX3UC
M 8030*4	Battery LED OFF	–	–	○	○	–	○	○
M 8031*4	Non-latch memory all clear	○	○	○	○	○	○	○
M 8032*4	Latch memory all clear	○	○	○	○	○	○	○
M 8033	Memory hold STOP	○	○	○	○	○	○	○
M 8034*4	All outputs disable	○	○	○	○	○	○	○
M 8035*5	Forced RUN mode	○	○	○	○	○	○	○
M 8036*5	Forced RUN signal	○	○	○	○	○	○	○
M 8037*5	Forced STOP signal	○	○	○	○	○	○	○
[M]8038	Parameter setting	○	○	○	○	○	○	○
M 8039	Constant scan mode	○	○	○	○	○	○	○

Number	Name	FX1S	FX1N	FX2N	FX3U	FX1NC	FX2NC	FX3UC
[D]8030	Value of Z0 (Z) register	○	○	–	–	–	–	–
[D]8031	Value of analog volume	○	○	–	–	–	–	–
[D]8032	NA							
[D]8033	NA							
[D]8034	NA							
[D]8035	NA							
[D]8036	NA							
[D]8037	NA							
[D]8038	NA							

*4: Processed when END instruction is executed.

*5: Cleared when switching from RUN to STOP

● Error detected

Number	Name	FX1S	FX1N	FX2N	FX3U	FX1NC	FX2NC	FX3UC
[M]8060	I/O configuration error	–	–	○	○	–	○	○
[M]8061	PLC hardware error	○	○	○	○	○	○	○
[M]8062	PLC/PP communication error	–	–	○	–	–	○	–
[M]8063	Serial communication error 1 [ch1]*6	○	○	○	○	○	○	○
[M]8064	Parameter error	○	○	○	○	○	○	○
[M]8065	Syntax error	○	○	○	○	○	○	○
[M]8066	Ladder error	○	○	○	○	○	○	○
[M]8067	Operation error*6	○	○	○	○	○	○	○
M 8068	Operation error latch	○	○	○	○	○	○	○
M 8069	I/O bus check*7	–	–	○	○	–	○	○
[M]8109	Output refresh error	–	–	○	○	–	○	○

Number	Name	FX1S	FX1N	FX2N	FX3U	FX1NC	FX2NC	FX3UC
[D]8060	The first device number of unconnected I/Os	–	–	○	○	–	○	○
[D]8061	Error code for PLC hardware error	○	○	○	○	○	○	○
[D]8062	Error code for PLC/ PP communication error	–	–	○	○	–	○	○
[D]8063	Error code for serial communication error 1 [ch1]	○	○	○	○	○	○	○
[D]8064	Error code for parameter error	○	○	○	○	○	○	○
[D]8065	Error code for syntax error	○	○	○	○	○	○	○
[D]8066	Error code for ladder error	○	○	○	○	○	○	○
[D]8067	Error code for operation error	○	○	○	○	○	○	○
D 8068	Operation error step number latched	○	○	○	○	○	○	○
[D]8069	Error step number of M8065 to M8067	○	○	○	○*9	○	○	○*9
[D]8109	Y number where output refresh error occurs	–	–	○	○*10	–	○	○*10

*6: Cleared when PLC switches from RUN to STOP. Note that M8068 and D8068 are not cleared.

*7: If M8069 is activated, the I/O bus check is executed. When an error occurs, the error code 6103 or 6104 is written to D8061, and M8061 is turned on. However, if 6104 is written to D8061, M8009 is also turned on and the I/O numbers of the units whose 24 VDC supply is OFF are written to D8009.

*8: When the units and blocks with the programmed I/O numbers are not mounted, M8060 operates and the first element numbers are written to D8060.

*9: Stored in D8312, D8313 if there are 32 K steps or more.

*10: Stored in D8314, D8315 if there are 32 K steps or more.

Appendix 1.3 Supplement of special devices

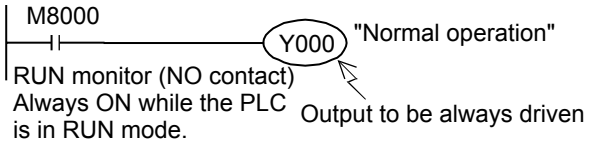
This section explains how to use the provided special devices to activate the built-in PLC functions for additional program control.

1. RUN monitor and initial pulse [M8000 to M8003]

● RUN monitor (M8000 and M8001)

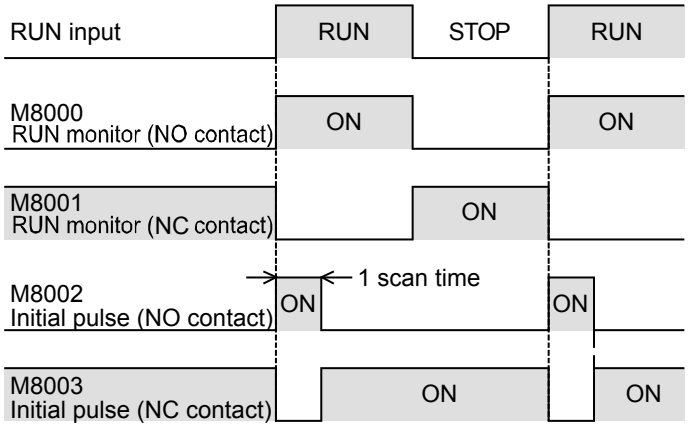
The RUN monitor (M8000 and M8001) may be used to continually drive an output during PLC "normal operation."

1) Example program



M8001 is always OFF while the PLC is in the RUN mode.

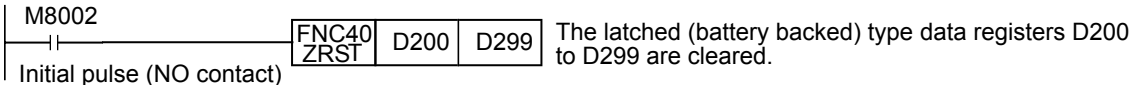
2) Flag operation timing



● Initial pulse (M8002 and M8003)

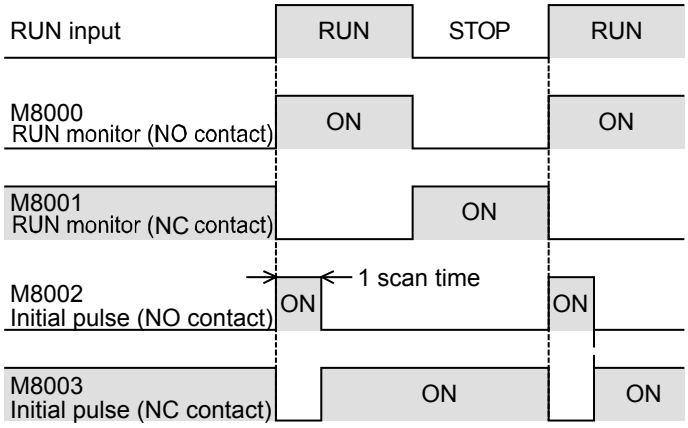
The initial pulse M8002 & M8003 is turned to ON or OFF respectively during the 1st scan of the PLC program.
It can be utilized as an initial setting signal in a program for initializing the program, for writing a specified value, or for another purpose.

1) Example program



M8003 turns OFF momentarily (for only 1 scan time) when the PLC enters the RUN mode.

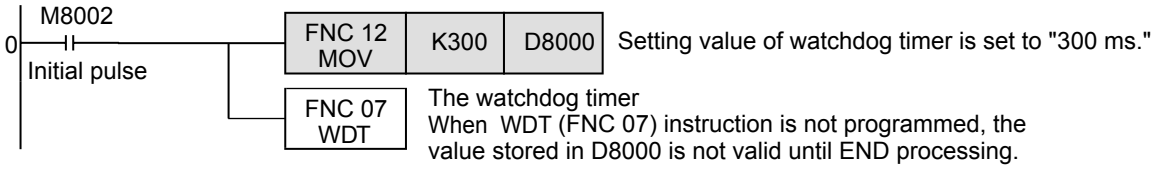
2) Flag operation timing



2. Watchdog timer [D8000]

The watchdog timer monitors the operation (scan) time of the PLC. When the operation is not completed within the specified time, ERROR (ERR) LED light turns on and all outputs are turned OFF. When the power is initially turned ON, "200 ms" is transferred from the system to D8000 as the default value. For executing a program beyond 200 ms, the contents of D8000 must be changed by the user program.

● Example program



● When a watchdog timer error occurs

A watchdog timer error may occur in the following cases. Add the above program to somewhere near the first step or adjust the number of execution FROM/TO instructions at the same scan.

1) When using many special function units/blocks
   When many special function units/blocks (such as positioning, cam switches, link and analog) are used, buffer memory initial setting time becomes long at turning on the PLC, thus extending the operation time and allowing the possibility for a watchdog timer error to occur.
2) When executing many FROM/TO instructions at the same time
   When many FROM/TO instructions are executed or when many buffer memories are transferred, it extends the scan time, and a watchdog timer error may occur.
3) When using many high speed counters (software counters)
   When many high speed counters are programmed and high frequency is counted at the same time, it extends the scan time, and a watchdog timer error may occur.

● How to reset the watchdog timer

The watchdog timer can be reset in the middle of a sequence program using WDT (FNC 07) instruction.
It is recommended to reset the watchdog timer by WDT (FNC 07) instruction when the scan time of a particular sequence program is extended or when many special function units/blocks are connected.

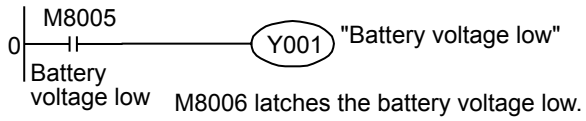
● Cautions on changing the watchdog timer

The watchdog timer time can be set to a maximum of 32767 ms. However, CPU error detection is delayed when the watchdog timer time is extended.
It is recommended to use the default value (200 ms) when no problems are to be expected in operation.

3. Battery voltage low detection [M8005 and M8006]

This special device detects low voltage in the lithium battery for memory backup.
When the PLC detects low battery voltage, BATT (BAT) LED light turns on. The following program demonstrates its use.

● **Example program**

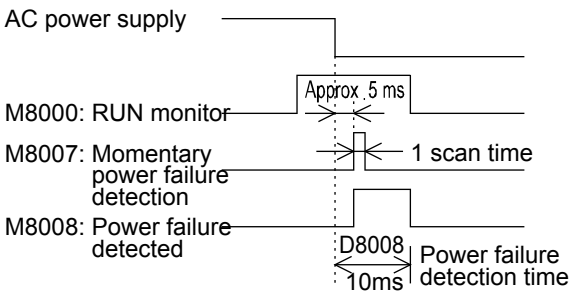


4. Power failure detection time [D8008, M8008 and M8007]

● **AC Power Supply Type PLC**

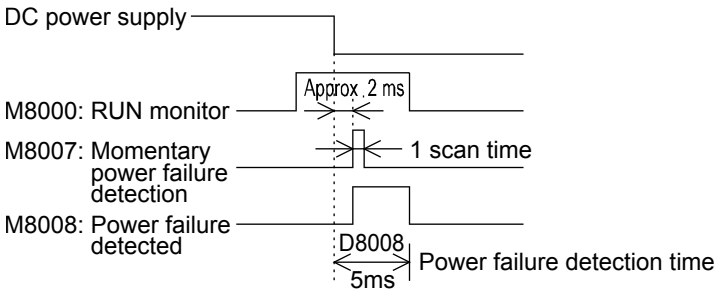
The table below shows the allowable momentary power failure time in FX3U PLCs (AC power supply type).

Supply voltage	Allowable momentary power failure time
100 VAC system	10 ms
200 VAC system	Setting range: 10 to 100 ms Set a value to D8008. Default: 10 ms



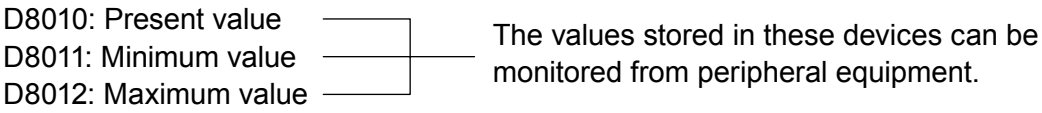
● **DC Power Supply Type PLC**

The allowable momentary power failure time in the FX3UC PLC (DC power supply type) is 5 ms.
Do not overwrite the power failure detection time in device D8008.



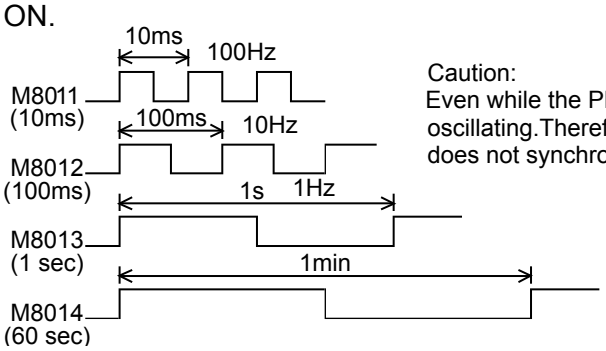
5. Scan time (monitor) [D8010 to D8012]

The present value, minimum value and maximum value of the PLC scan time (operation time) are stored in D8010, D8011 and D8012 respectively.
When using the constant scan mode, the values stored in these devices include the waiting time for the constant scan time.



6. Internal clock [M8011 to M8014]

The PLC has the following four types of internal clocks which are always oscillating while the PLC power is ON.



Caution:
Even while the PLC is in the STOP mode, these clocks are always oscillating.Therefore, the rising edge of the RUN monitor (M8000) does not synchronize with the clock start timing.

7. Real time clock [M8015 to M8019 and D8013 to D8019]

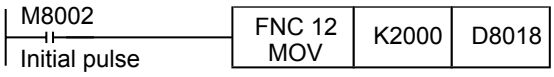
● Assignment of special auxiliary relays (M8015 to M8019) and special data registers (D8013 to D8019)

Number	Name	Operation and function
M8015	Clock stop and preset	When M8015 turns ON, the real time clock is stopped. At the edge from ON to OFF, the time from D8013 to D8019 is written to the PLC and the clock is started again.
M8016	Time read display is stopped	When M8016 turns ON, the time display is stopped (but RTC is continued).
M8017	±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".)
M8018	Installation detection	This device is always ON.
M8019	Real time clock (RTC) error	When the data stored in special data registers is outside the allowable time setting range, this device turns ON.

Number	Name	Set value range	Operation and function
D8013	Second data	0 to 59	Use these devices for writing the initial value in time setting or read the present time. • D8018 (year data) can be changed over to the four-digit year mode. In the four-digit year mode, 1980 to 2079 can be displayed. • Clock accuracy: ±45 sec/month (at 25 °C) • Leap year correction: Provided
D8014	Minute data	0 to 59	
D8015	Hour data	0 to 23	
D8016	Day data	1 to 31	
D8017	Month data	1 to 12	
D8018	Year data	0 to 99 (last two digits of year)	
D8019	Day-of-the-week data	0 to 6 (which corresponds to Sunday to Saturday)	

● Changeover of the year display (to the four-digit mode)

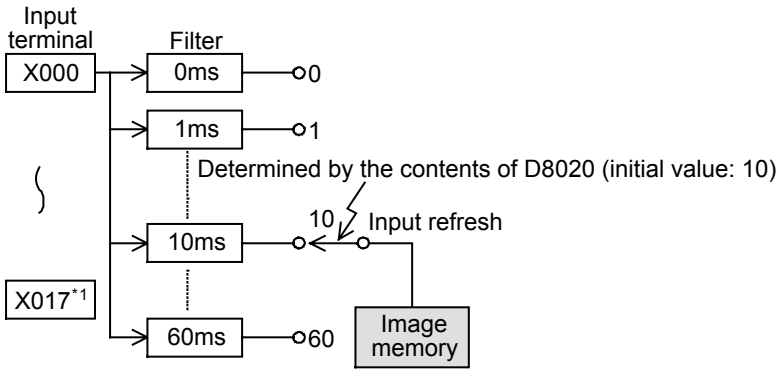
When changing the year data to the four-digit mode, add the following program.
D8018 is set to the four-digit year mode on the second ladder scan in RUN mode.



1) The PLC is usually operating in the two-digit year mode.
   When the above instruction is executed during RUN and "K2000 (fixed value)" is transferred to D8018 (year data), D8018 switches to the four-digit year mode.
2) Execute this program every time the PLC enters RUN.
   Only the year data display switches to four-digit mode when "K2000" is transferred. The actual time date is not affected.
3) In the four-digit year mode, the values "80" to "99" correspond to "1980" to "1999" and "00" to "79" correspond to "2000" to "2079".
   Examples: 80 = 1980, 99 = 1999, 00 = 2000, 79 = 2079
4) When connecting the data access unit FX-10DU-E, FX-20DU-E or FX-25DU-E, use the two-digit year mode.
   If the four-digit year mode is selected, the year data will not be displayed correctly.

8. Input filter adjustment [D8020]

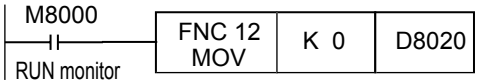
The inputs X000 to X017*1 have a digital filter circuit with a setting range of 0 to 60 ms. The digital filter setting value is set between 0 to 60 ms (in 1 ms steps) in special data register D8020. After PLC powers ON, D8020 is automatically set to K10 (10 ms).



*1. X000 to X007 in the FX3U-16M□

● Program example for adjusting the input filter

When the program shown below is executed, the filter constant is changed to 0 ms.
Because the C-R filter is provided in the hardware, however, the filter constant is shown in the table below when "0" is specified.



Input number	Input filter value when "0" is set
X000 to X005	5 μs*1
X006, X007	50 μs
X010 to X017*2	200 μs

*1 When setting the input filter to "5 μs" or when receiving pulses whose response frequency is 50 k to 100 kHz using high speed counters, perform the following:
- Input wiring length should be 5 m (16' 4") or less.
- Connect a bleeder resistor (1.5 kΩ, 1/2 W) to an input terminal. The load current of the open collector transistor output in the device on the other end should be 20 mA or more including the input current of the PLC.

*2 X000 to X007 in the FX3U-16M□

- The input filter constant can be changed as many times as needed in the user program.
- This input filter adjustment described here is not required when using high speed counter, input interrupt, or pulse catch (M8170 to M8175) functions.

9. Battery [BATT (BAT)] LED OFF command [M8030]

- **Batteryless operation**

  When M8030 is set to ON, the battery LED does not turn ON even if the voltage in the battery for memory backup becomes low.
  When the indication of "battery voltage low error" is not required or when the battery is removed, set M8030 to ON.
  When the batteryless operation is required, however, do not use M8030 but refer to "2. Parameter setting in peripheral equipment" below.

- **Parameter setting in peripheral equipment**

  Specify "batteryless operation" mode in the parameter settings.

  1) When the batteryless operation option is specified
     When "batteryless mode" is specified, the control to turn OFF the BATT (BAT) LED and initialization of the latch area for the devices shown below are automatically executed by the PLC system.
     - Auxiliary relay (M) - Counter (C) - State relay (S)
     - Data register (D) - Timer (T) - Extension data register (R)
  2) Applicable programming tool
     Some programming tool versions do not support "batteryless mode." In such versions, input a sequence program to enable the batteryless operation as explained below.

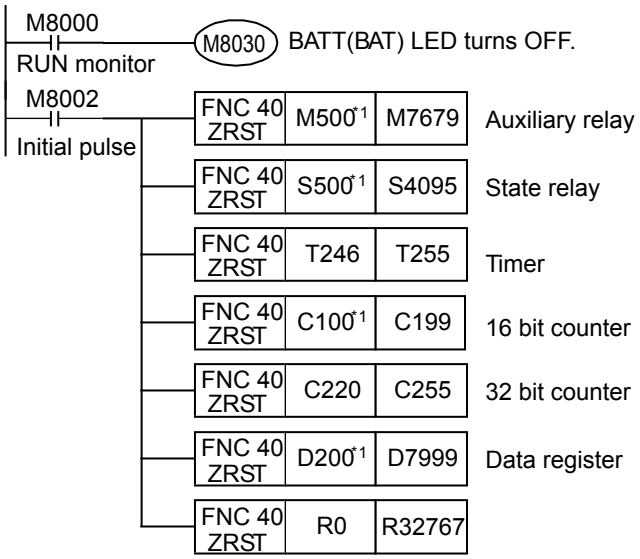
- **Conditions for batteryless operation**

  1) An FLROM (optional memory cassette) for program memory is installed so that programs are not erased.
  2) The latch (battery backed) devices such as auxiliary relays and data registers are not used for control.
  3) The sampling trace function is not used.
  4) The real time clock function is not used.

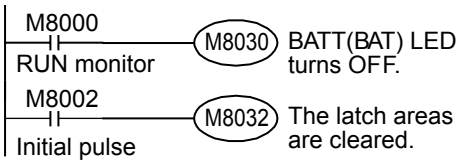
- **Example program for batteryless mode**

  When a parameter setting for "batteryless mode" is not available, create the sequence program shown below.

  - Example program for clearing the memory backup area
    {when the latch (battery backed) ranges in the parameter settings are set to their initial values.}



or



*1 The device number is the default number of the latch (battery backed) device range limits in the parameter settings.
If the latch (battery backed) device numbers in the parameter settings are changed to other values, change the device numbers here in accordance with the changed latch ranges.

- **Cautions for communication setting devices (D8120, D8121 and D8129)**

The special data register D8120 (communication format setting), D8121 (station number setting), and D8129 (time-out check time) are backed up by the battery.
When using the batteryless function, reset these special data registers once, and then transfer proper set values to them by program.
The communication conditions can be set in the parameter settings.
When the communication conditions are set in the parameter settings, the PLC transfers the parameter values to the above special data registers before operation. Thus it is recommended to set the communication conditions via the parameter settings.

A

10. Clear command [M8031 and M8032]

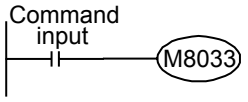
For all devices (image memory) in the PLC, the latch (battery backed) areas and non-latch areas can be cleared.
M8031 (non-latch memory all clear), M8032 (latch memory all clear)

Device number	Cleared devices
M8031 (non-latch memory all clear)	• Contact image of output relay (Y), general type auxiliary relay (M), and general type state relay (S) • Contact and coil of timer (T) • Contact, counting coil, and reset coil of general type counter (C) • Present value of general type data register (D) • Present value register of timer (T) • Present value register of general type counter (C)
M8032 (latch memory all clear)	• Contact image of latch (battery backed) type auxiliary relay (M) and latch (battery backed) type state relay (S) • Contact and coil of retentive type timer (T) • Contact, counting coil, and reset coil of latch (battery backed) type counter and high speed counter • Present value of latch (battery backed) type data register (D) • Present value register of retentive type timer (T) and 1-ms timer (T) • Present value register of latch (battery backed) type counter and high speed counter

11. Memory hold stop [M8033] (output hold in STOP mode)

When the special auxiliary relay M8033 is turned ON, the output status in the RUN mode is retained even if the PLC status switches from RUN to STOP.

● **Example program**



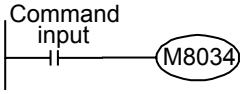
Outputs are retained even during STOP mode.

For example, when a heater is driven by the PLC, the PLC can be stopped while the heater and other
equipment are kept driven, and then the PLC can be started again after program changes.

12. All outputs disable [M8034]

When M8034 is turned ON, the output memory is cleared. As a result, all actual output relay contacts are turned OFF and the PLC is operated in the image memory.

● **Example program**

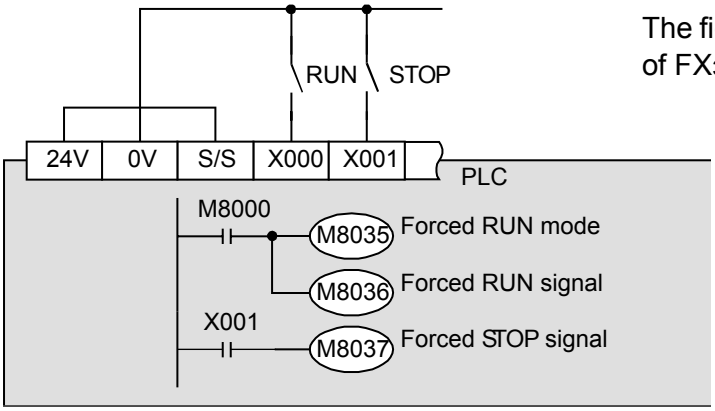


All outputs are disabled (image RUN status).

13. Individual operation for RUN/STOP input [M8035 to M8037]

When using external push button switches to control the PLC's RUN/STOP mode, operate the switches using the following procedure.
The PLC enters RUN mode by one-shot input of the RUN switch, while one-shot input of the STOP switch drives the STOP mode.

● Example program



The figure on the left shows an example of FX3U PLC (sink input).

Input the above program in the PLC.

● Setting method

1) Turn the built-in RUN/STOP switch to STOP.
2) Specify the RUN input switch, input (X) (X000 is specified in the above circuit diagram example.)
   Make the external RUN/STOP input valid by specifying an input between X000 and X017*1 for the RUN input signal.
   *1. X000 to X007 in the FX3U-16M□
   a) Display the parameter setting in the programming tool
      In GX Developer case, double-click [Parameter] - [PC parameter] in the project tree to display the dialog box.
      Click "PLC system (1)" tab, and set "RUN terminal input."
   b) Specify the input (X) number to switch from STOP mode to RUN mode.
3) Specify the STOP switch input (X)
   Specify an arbitrary input terminal (actual I/O on the PLC) in the sequence program.
   Refer to the above program.
4) Transfer the program and parameters to the PLC.
5) For the parameter settings to become valid, the PLC power must be turned from OFF to ON.

● Cautions

1) When both RUN and STOP switches are pressed at the same time, priority is given to the STOP switch.
2) When the built-in RUN/STOP switch is turned to RUN, the PLC can be set to RUN mode. However, when the STOP switch assigned to an arbitrary input is activated, the PLC will enter STOP mode. (Even if the built-in switch is turned to RUN, priority is given to the STOP command.)

A

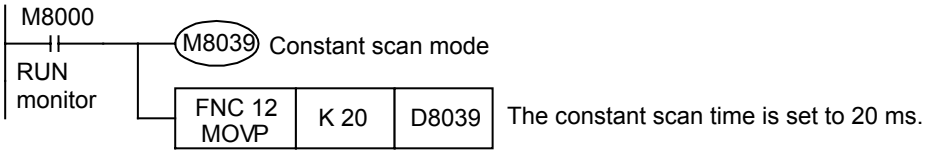
● **RUN/STOP command via the programming tool**

1) Using the programming software for personal computer
   There is a remote RUN/STOP function in the programming software.
   By using the programming software, the PLC can be set to the RUN or STOP mode by giving a command from the personal computer.
2) Using any other programming tool
   When M8035 (forced RUN mode) and M8036 (forced RUN signal) are set to ON in the forced ON/OFF procedure, the PLC begins RUN mode.
   When M8037 (forced STOP signal) is set to ON, the PLC changes to STOP mode.
3) Even when the built-in RUN/STOP switch is on the RUN side of the PLC.
   The remote STOP command via the programming tool or M8037 (forced STOP signal) are valid.

14. Constant scan mode [M8039 and D8039](Averaging the scan time)

When the special auxiliary relay M8039 is set to ON and a setting value for the constant scan time (in 1-ms units) is stored in special data register D8039, the scan time in the PLC does not become shorter than the value stored in D8039.
The PLC pauses for the remaining time when the operation ends earlier, and then returns to step 0.

● **Example program**



● **Cautions**

1) When using an instruction executed in synchronization with a scan
   a) When using an instruction executed in synchronization with a scan such as RAMP (FNC 67), HKY (FNC 71), SEGL (FNC 74), ARWS (FNC 75) and PR (FNC 77), it is recommended to use the constant scan mode or to use the instruction in a timer interrupt program.
   b) When using HKY (FNC 71) instruction
      It is necessary to use a scan time of 20 ms or more due to the response delays of the key input filter.
2) Scan time display (D8010 to D8012)
   The scan time specified in the constant scan time is included in the scan time display stored in D8010 to D8012.

Appendix 1.4 Types and setting of parameters

Specifying parameter settings means setting the environment where the PLC operates.
Almost all FX3U/FX3UC PLCs can be used with factory default values. When it is necessary to add optional memory, set the comment capacity, set the communication condition for serial ports, etc., change the parameter settings by a programming tool such as personal computer.

● Parameter list

The following items may be set in the parameter settings.

Classification	Item	Description
Memory capacity	Memory capacity	This parameter specifies the maximum value for the number of steps to which a sequence program can be input. 1) The upper limit is determined by the capacity of the built-in memory or optional memory. 2) The program memory, file register, comment area, and other special setting capacities are contained in this memory capacity.
	Comment area	This parameter incorporates comments into the program memory. 1) Because comments remain in the PLC, the contents can be easily understood at the time of maintenance. 2) Up to 50 comments can be input when one block is specified, but the program memory capacity is reduced because the comment area requires 500 steps in the memory capacity.
	File register	This parameter incorporates data registers into the program memory. 1) A sequence program and control data such as machining set values can be handled together, which is convenient. 2) Up to 500 file registers can be created when one block is specified, but the program memory capacity is reduced because file registers require 500 steps in the memory capacity.
	Other special setting capacity	1) This parameter sets whether or not the special block/unit initial value setting function is used. When this function is used, the program memory capacity is reduced because this function requires 4000 steps (8 blocks) in the memory capacity. 2) This parameter sets whether or not the positioning setting (constants and setting table) in TBL (FNC152) instruction is used. When this setting is used, the program memory capacity is reduced because this setting requires 9000 steps (18 blocks) in the memory capacity.
Device setting	Latch range setting	This parameter enables to change the latched (battery backed) device range and the non-latch device range inside the PLC.
I/O assignment setting	I/O assignment setting	This setting is not written to the PLC. When the I/O range is set according to the system configuration, however, inputs and outputs are checked by the program check in GX Developer.
	Special unit setting	This parameter sets the initial values of the buffer memory (BFM) for each special block/ unit number. It is necessary to set the memory capacity.
PLC system setting (1) [PLC mode]	Batteryless mode	This parameter sets the PLC operation mode without a battery. When the baterryless mode is set, detection of battery voltage low level error is stopped automatically, and consequently, contents of latched (battery backed) devices becomes inconsistent and are initialized automatically.
	Modem initialization	This parameter automatically sends a specified AT command as an initialization command to a modem connected to the serial port.
	RUN terminal input setting	This parameter sets whether one input terminal in the PLC is used for RUN input.
	RUN terminal input number	This parameter specifies the input number of the RUN input described above within the range from X000 to X017.
PLC system setting (2) [Serial communication]	Serial port operation setting	This parameter corresponds to the following settings by specifying each contents on the PC screen: Setting of communication format (D8120, D8400 and D8420) Setting of station number (D8121 and D8421) Setting of timeout check (D8129, D8409 and D8429)
Positioning setting	Constant setting	This parameter sets interrupt inputs for the maximum speed, bias speed, creep speed, zero return speed, acceleration time, deceleration time, and DVIT instruction. It is necessary to set the memory capacity.
	Detailed setting	This parameter sets the operation table. It is necessary to set the memory capacity.
Others	Entry code	This parameter sets protection to prevent erroneous writing and plagiarism of a sequence program. The entry code can be specified in 8 hexadecimal characters among A to F and 0 to 9. In FX3U and FX3UC PLCs Ver.2.20 or later, the second entry code (in 8 characters) can be added to allow specification of the entry code in 16 characters.
	Program title	This parameter enables to set a character string to be used as the program title.

A

Appendix 2 PLC GIOSSARY

A

ADDRESS
Location in memory where data is stored in the PLC.
Addresses in program memory are called step numbers.

ALLOWABLE MOMENTARY POWER FAILURE PERIOD
Refers to the maximum time that the PLC can continue normal operation during a power failure.
This time is 10ms for FX series PLCs.

ANALOG
Refers to a signal that varies over continuous range and that cannot be represented clearly in numerical values (digital values).
Pressure, temperature, voltage, current, time, etc. are examples of ANALOG.

ANALOG TIMER
Refers to a timer that calibrates time with volume. FX-8AV volume input adapter is available to use.
On the other hand, there are timers that are incorporated inside PLCs that are set with key operation of the programming panel.
These timers sometimes are called soft timers.

AND
Refers to serial connection of N.O. contacts in PLC programs.

AND INVERSE
Refers to serial connection of N.C. contacts in PLC programs.

B

BATCH I/O PROCESSING SYSTEM
Refers to a system where all input signals to a PLC are stored in the image memory. This data is then processed, and the results are output.
The other type of I/O system called direct I/O system where I/Os are performed every time an instruction is executed.

BATTERY BACKUP
At power failure, data in program memory, counters, and some sub relays in a PLC are retained via battery back up.

BIT
The smallest unit of data stored. 1 bit can be 0 (OFF) or 1 (ON). For example, 3-bit data can be 8 different information: 000, 001, 010, 011, 100, 101, 110, 111.

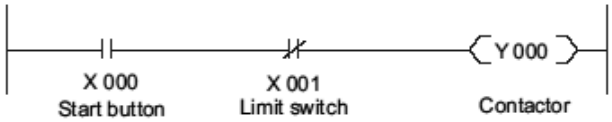
BYTE
A unit of data stored that is equivalent to 8 bits.
Instructions provided by the FX series PLCs use 2 bytes for each step.

C

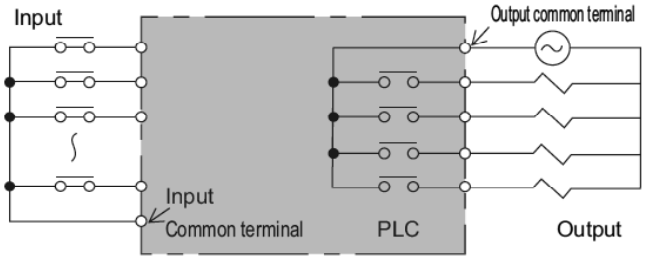
CLEAR
(1) Refers to initializing display of a programming panel.
(2) Refers to clearing the current value of a counter and timer to zero.

CODING
Refers to rewriting sequence circuits in a set instruction language to make an instruction list.
Instructions on this list are stored in PLC's program memory.

COMMENT
This refers to details users can add to elements and circuit blocks in ladder diagram.



COMMON
Refers to common terminals.
All inputs to a PLC are passed through common terminals, and outputs are passed through 4-point terminal or independent.



CONTACT OUTPUT
This refers to an output terminal in the PLC. This is also known as Relay output.
When applying large current or performing frequently-conducted operation, service life of the contacts is shortened.

COUNTER
Refers to a device that counts how many pulse inputs are supplied to its input and closes a contact when the accumulated count reaches the set value.

CPU (CENTRAL PROCESSING UNIT)
CPUs are the brains of PLCs, therefore they are the important component in a PLC.
As hardware, a CPU is an ultra micro scale integrated circuit (UMSI) which includes a microcomputer and memory.

CURRENT VALUE
Refers to the current value of changing number in a timer or counter.
For example, in a timer with 10 seconds set as a set value, a current value increments from 0 in steps of 1 to 10 seconds. The output contact of this timer is closed when a current value reaches 10 seconds.

CYCLIC OPERATION METHOD
Refers to the operation system where a program executes all of its steps and repeats steps from 0 step after the last step has been executed.
If an operation cycles short, it appears that control operations are batch-made as seen with relay panels.

D

DEBUG
Refers to correcting errors in a program.

DEVICE
Refers to elements used in a program, such as relays, timers, and counters in a PLC.
→ Also see ELEMENT NUMBER

DIGITAL
Refers to amount that can be clearly represented by numerical values, such as ON (1), OFF (0), and other numerical values (i,e, 1, 2, 3, 4 ...).

DIRECT SYSTEM
Refers to one of PLC's operation systems, which carries out I/Os immediately after an instruction is executed.

DOCUMENTATION
Refers to paper documents. Examples of PLC-related paper records are circuit diagrams, instruction lists, and so on.

DOWNLOAD
Refers to writing and transferring programs from an A6GPP/PHP or HPP (off line mode) to a PLC. The reverse of this process is known as upload.

E

EEPROM
(ELECTRICALLY ERASABLE AND PROGRAMMABLE READ ONLY MEMORY)
A type of memory exclusively used for reading. Stored data in an EEPROM is not lost at power failure.
(Writing program to an EEPROM from a programming panel is possible but takes longer than writing to RAMs).

ELEMENT NUMBER
Refers to numbers assigned to elements (devices) in a PLC, such as relays, timers, and counters.

END
Refers to an instruction that is written at the end of a program indicating that there are no more instructions in a program.
Note: the END instructions take extra time for I/O processing before completing execution of the program.

EPROM
(ERASABLE AND PROGRAMMABLE READ ONLY MEMORY)
A type of memory exclusively used for reading. Stored data in an EPROM is not lost at power failure.
A ROM writer or ultraviolet eraser (eraser) is used to write to an EPROM.

ERASER
Refers to devices that delete programs stored in an EPROM (ultraviolet eraser).

ERROR CODE
Refers to classification numbers that identify the cause of an error that occurred in PLC programs.

EXTENSION MODULE,
EXTENSION BLOCK
Expansion modules and blocks are used in combination with a base unit. Relays within these produces only input and output relays.
Expansion modules are also equipped with an internal powersupply circuit.

EXTERNAL POWER SUPPLY
Refers to a power supply that feeds a PLC or a load. Also refers to a power supply located outside a PLC to feed sensors.
PLCs are fed by an external power supply and create a power supply of 5VDC, 12VDC, or 24VDC.
These direct current power supplies are called internal power supply.

F

FA (FACTORY AUTOMATION)
Refers to automating various operation of electric devices conducted in a factory.
OA stands for Office Automation.
HA stands for Home Automation.

FMS (FLEXIBLE MANUFACTURING SYSTEM)
Refers to a system that automates production procedures on high-mix/low-volume production.

FORCED ON/OFF
Refers to forcing on or off each device by using keys on a programming panel along with sequence operation.
A Forced ON/OFF instruction is enabled for one operation

cycle.
This characteristic is well suited for the followings.
(1) Output relays or sub relays that are closed or opened according to the self-maintaining action or set instructions.
(2) Timer and counter
Note that this instruction works for output relays not having a self-maintaining circuit or set instructions as long as a PLC is at stop.
This is usable in test mode operation.

G

GPP (GRAPHIC PROGRAMMING PANEL)
A device that allows users to write sequence programs, create lists, and perform monitoring on a CRT display.
It also enables programs to be transferred to floppy disks and EPROMs.

H

H/W (HARDWARE)
A general term for equipment that physically exist.
In PLC fields, hardware means PLCs. Hardware cannot run without software.

HARD COPY
Refers to sequence programs or sequence circuit diagrams printed on paper. Printers are available for preparing hard copies.

HIGH-SPEED COUNTERS
Refers to counters with a special input terminal. With special terminals, these counters can obtain pulses of 10KHz or less. Counting proceeds in reaction to sequence operation and interrupt pulses. Generally, standard counters incorporated in a PLC only count pulses of 20 to 30Hz.

HPP (HANDY PROGRAMMING PANEL)
Refers to a simple device with which users can write/read programs to/from a PLC and monitor a PLC.

I

I/F (INTERFACE)
A boundary across which two independent devices exchange signals.
For example, an FX-232AW interface module comes in between a personal computer and a FX series PLC.

I/O (INPUT AND OUTPUT)
Refers to inputting and outputting data.

IC-RAM
Refers to a RAM that is provides a type of integrated circuit.
→ Also see RAM.

IMAGE MEMORY
Refers to memory that stores ON/OFF status of PLC I/Os, sub relays, timers, counters, etc.
At power failure, image memory is partially retained via the battery.

INDUCTIVE LOAD
Refers to a load that generates surge voltage if applied current is shut off. Coils (winding wire) are inductive loads. Other types of loads are resistance loads (generates no surge voltage), capacitive loads (generates inrush current), and so on.

INITIAL
Refers to an initial state.
For example, all output Ys are initially off at power-on of a PLC.
Another example of an initial set operation is when a PLC starts running, it generates initial pulses to initialize counters.

INPUT DEVICE
Refers to operational devices or detectors such as pushbutton switches, limit switches, selector switches, proximity switches, photoelectric switches. Input devices are connected to an input
Refers to operational devices or detectors such as pushbutton switches, limit switches, selector switches, proximity switches, photoelectric switches. Input devices are connected to an input terminal of a PLC.

INPUT IMPEDANCE
Refers to the equivalent resistance value of a PLC's input circuit.
Value of the input current is obtained by dividing an input voltage with this value.

INPUT VOLTAGE
Refers to the voltage applied to the PLC's input circuits.
The F series PLC support 24VDC or 100/200VAC.A PLC supporting 24VDC has the power supply inside the PLC. Therefore another 24VDC power supply does not need to be installed in the system.

INRUSH CURRENT
When using DC loads such as contactors (electromagnetic contactors) or solenoid valves, lamps, and capacitive loads (condensers), current much larger than rated current (approx. 6 to 10 times larger) is generated right after voltage is applied to them.
This large current is called inrush current, and blocking the inrush current by relays significantly shortens their product life.

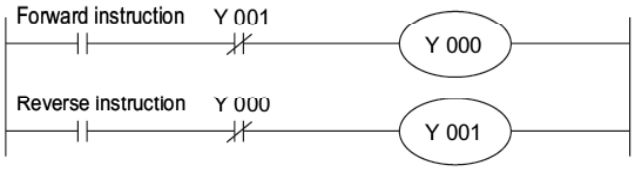
INSTRUCTION
Refers to a building block within the program.
After a PLC starts running, the CPU reads and executes these instructions in a cyclic manner.

INSTRUCTION EXECUTION TIME
Time needed for completing one instruction of a program.
→ Also see OPERATION TIME

INSULATION TRANSFORMER
Refers to a transformer whose primary coil and secondary coil are wound separately and are therefore disconnected electrically.
These type of transformers feature noise reduction.

INTERFACE
Refers to intermediate circuits or modules that function between two devices. Interface rules include insulating circuits to exchange signals, converting voltage level or signal formats, etc.

INTERLOCK
Refers to an action that prevents undesired operations happening simultaneously.



For example, it is dangerous that a forward contactor and reverse contactor operate together. To use them, an interlock must be placed outside of the PLC as well as inside it.
If an interlock is placed only inside a PLC, they may be energized momentarily due to the on-delay and off-delay of PLC's output relays.

INTERNAL RELAY
Dedicated relays provided in a PLC. Internal relays can be written in a program but cannot be used for external outputs. Sub relays and temporary relays are another name of internal relays.

L

LADDER DIAGRAM
Refers to a circuit diagram showing a program with relay symbols.
It is named as such because it looks like a ladder.

LATCH
Refers to a memory where a signal is stored until the next signal comes in.
The ON/OFF status of the PLC's output contacts is stored in the output latch memory.

LATCHED (BATTERY BACKED) RELAY
Some sub relays have a backup battery. Such relays are called Latched (Battery Backed) relays.

LEAKAGE CURRENT
→ See OPEN CIRCUIT LEAKAGE CURRENT

LED (LIGHT EMITTING DIODE)
Usually called LED.
LEDs are compact lamps made of semiconductors. They can operate with small-current and feature a long service life.

LIMIT SWITCH
Refers to a switch that detects machinery reaching its movement limit. This switch is useful for scheduling machinery to stop at a planed location.

LITHIUM BATTERY
Non-changeable battery that is used as backup power supply for PLC's memory during a power failure.
The service lives of lithium batteries are about five years. Exchange accordingly.

M

MAIN UNIT
Refers to a PLC body including CPUs and I/Os. It also incorporates timers, counters, sub relays, etc.

MASTER CONTROL
Refers to an instruction that is issued for connecting series of sequence circuits to a main line via common contacts.
The instruction that is used to cancel master control is the master control reset instruction.

MEMORY
Refers to elements where programs are stored.
→ Also see RAM.
→ Also see EPROM.
→ Also see EEPROM.

MEMORY CASSETTE
A cassette that houses memory. This makes it easier to handle, insert and remove memory.

MICROCOMPUTER
Refers to a compact type CPU composed of ultra micro scale integrated circuits (UMSI). A micro processor and memory are incorporated in a single UMSI.

MNEMONIC
One of programming languages used in a sequence program.
MNEMONIC is written in the form of easy-to-remember codes such as LD, AND and OR.

MONITOR
Refers to watching how devices inside a PLC are behave. With an A6GPP/PHP, ON/OFF status of contacts and coils can be judged by circuit status.

MTBF (MEAN TIME BETWEEN FAILURES)
Refers to average failure interval. More concretely, averaged time in which a device can operate without having any failures.
For example, when 150 PLCs with MTBF of 15 years are used, this means 10 PLCs out of those 150 may go out of order within a year.
PLCs feature much longer MTBF than relay panels.

MTTR (MEAN TIME TO REPAIR)
Refers to average time needed for repair. This value is obtained by dividing repair time by times of repairs.
PLC' s repair time can be reduced by replacing modules directly.

N

N.O. (Normally Open) Contacts
Refers to contacts that are Open Normally and closed when a coil is excited.

N.C. (Normally Closed) Contacts
Refers to contacts that are normally closed and open when a coil is excited.

NESTING N0 to N7
If a master control instruction is placed within a master control instruction , nesting occurs, codes of N0 to N7 are used to indicate which layer an instruction is located.
This restriction position is also applied to sub routine instructions.

NOISE RESISTANCE
Refers to the upper limit of noise with which an electric device can operate properly.
Usually, noise resistance is represented by pulse width of noise and maximum voltage of pulse.

NOISE SIMULATOR
Refers to a noise generator that can change voltages and noise width to test noise resistance of electric devices.

NO-VOLTAGE CONTACT
Refers to an input contact that is provided in a PLC and is not connected with external power supply circuits.
PLCs incorporating their own input power supply use this contact.

O

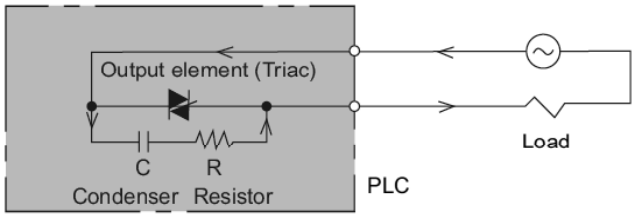
OCTAL
Refers to a numerical system that does not contain the numbers 8 and 9, such as 0 to 7, 10 to 17, 20 to 27, and so on. The FX series PLCs provide I/O numbers in octal. Unlike I/O numbers, step numbers and constants for timers/counters are basically represented in decimal, however treated inside a PLC, they are represented in binary. Digital switches of "binary coded to decimal" are used as input devices of a PLC."

OFF DELAY TIMER
Refers to a timer that opens a contact after a set period of time. (Delay timer energized by voltage drop)

OFFLINE PROGRAMMING
Refers to programming a peripheral device independently without connection to a PLC using M26A6GPP and PHP, etc.

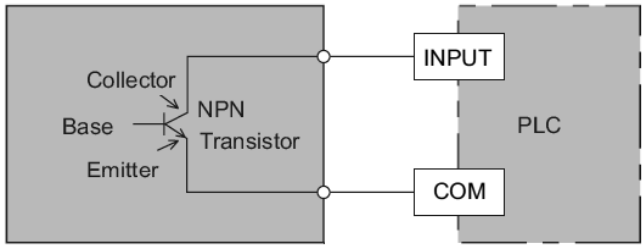
ON-DELAY TIMER
Refers to a timer that closes a contact after a set period of time. Delay timer energized by voltage suction.

OPEN CIRCUIT LEAKAGE CURRENT
SSRs have C-R absorbers in parallel with the outputs. Because of this, when an output element is opened for an AC load, the load is still slightly energized.
This micro current is called open circuit leakage current.



In FX series PLCs, minimum load for SSR outputs are preset.
For micro current load smaller than this minimum value, a bypass circuit breaker such as a lamp, resistor or surge absorber
must be installed in parallel with the load.

OPEN COLLECTOR
Refers to a collector on a transistor is directly connected to PLC's input terminal without going through other circuits.



OPERATING CURRENT
Refers to the limit input current for energizing or de-energizing PLC' s internal input circuit.
Operating voltage is obtained by multiplying a value of operating current with input impedance.

OPERATION CYCLE
Refers to the time obtained by multiplying an average operation speed with the number of program steps, and multiplying that result with a given coefficient.
Operation cycle is also called cycle time or scan time.
The more interrupt processes, the larger coefficient.

OPERATION SPEED
Refers to the time taken for executing one instruction. Basic sequence instructions, such as LD, AND, OR and OUT, take 0.74 μs or less. Application instructions take longer time ranging from several dozen to several hundred us depending on their contents. One operation cycle is obtained by multiplying a total of whole operation time taken for all program steps and time taken for I/O processing with a given magnification.

OR
Refers to a parallel connection of N.O. contacts in PLC.

OR INVERSE
Refers to parallel connection of N.C. contacts in PLC programs.

OUT
In PLC fields, OUT refers to a drive instruction issued in coils.
Combined with the meaning "coil", winding wire of relays, OUT in PLC files refers to this meaning in comparison with electromagnetic relays.

OUTPUT EQUIPMENT
Output equipment of a PLC includes pilot lamps, contactors (electromagnetic contactors), solenoid valves, electromagnetic clutch brakes, and more. All of them are connected to output terminals of a PLC.
They are sometimes called a PLC load.

OUTPUT FORMAT
Output formats supported by PLCs fall into three categories;
Relay output using contacts, no-contact output for AC load (SSR), no-contact output for DC load (transistor).

P

PC (PROGRAMMABLE CONTROLLER)
Digital electric devices that have a programmable memory and perform sequence control.
MELSEC, PLCs that do not process numerical values are called PLC (Programmable Logic Controller) or SC (Sequence controller).

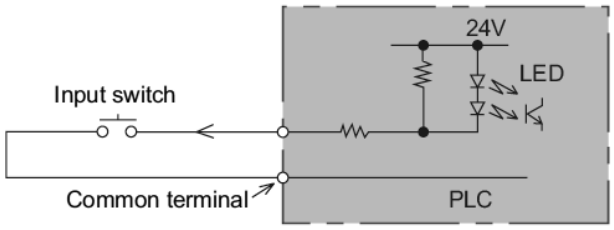
PERIPHERAL DEVICE
Refers equipment that is used for writing and storing programs in a PLC, monitoring a PLC, or creating documents.

PHOTO-COUPLER
Refers to semiconductor elements that insulate an input circuits an output circuits, and transfer signals using light.
Photo couplers reduce noise operation, therefore they are used in PLC input circuits.
On the other hand, circuits electrically and physically connected to other circuits are called non-insulation circuits.

PHOTOELECTRIC SWITCH
Refers to non-contact switches that are closed or opened in reaction to a light path between a projector and photoreceiver is blocked or not.
For an input to a PLC, the NPN transistor of the open collector type is usually used. This type uses current consumption of 50mA/24VDC. A power supply must be selected based on these values.

PHP (PLASMA HANDY GRAPHIC PROGRAMMING PANEL)
Unlike GPPs, PHPs employ plasma display instead of CRT. This makes PHPs lighter than GPPs.

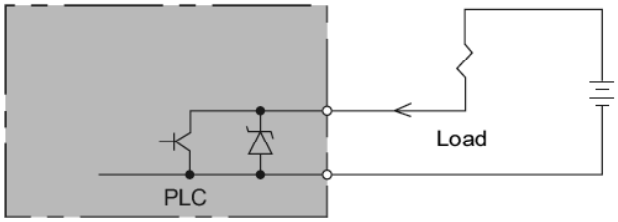
POLAR CHARACTERISTICS OF INPUT SIGNAL
(1) Negative common input
Refers to an input common terminal is located on the negative side of voltage. (Commonly used in Japan)



(2) Positive common input
Refers to that an input common terminal is located on the positive side of voltage. (Commonly used in Europe)

POLAR CHARACTERISTICS OF OUTPUT SIGNAL
(1) NPN output
Refers to that NPN transistor outputs are connected to the negative side of load. (Commonly used in Japan)

A



(2) PNP output
Refers to that PNP transistor outputs are connected to the positive side of load. (Commonly used in Europe)

PROGRAM CAPACITY
How many steps an instruction takes depends on the contents of the instruction. While some instructions take only one step, others take 10 to 20 steps.
Program capacity indicates how many steps in total a program can have. For example, the FX2 PLC provides program capacity of 8K step (0 to 7,999 steps). Additionally, one comment (up to 15 alphanumeric characters) takes 10 steps out of program memory and uses a program memory with 4,000 steps in 500 steps units.
One file register takes one step and uses program memory with 2,000 steps in 500 steps units.

PROGRAM
An organized list of instructions. Programming is to write these instructions to the memory of a PLC (program memory or a user memory).

PROGRAMMING LANGUAGE
A programming language that uses symbols from a relay sequence diagrams, i, e, LD, AND and OR, is called relay symbol language. Whereas, a programming language that uses the STL or RET instructions according to SFC (Sequential Function Chart,) is called step ladder language. Micro PLCs can use both languages according to what operation is required.

PROXIMITY SWITCH
Refers to non-contact switches that are closed when an object approaches. Most of these switches support transistor output.
For an input to a PLC, an open type collector NPN transistor is usually used.
Usually, proximity switches use current consumption of 10mA/24VDC. A power supply must be selected based on this value.

PULSE
Refers to a signal with a narrow width.
The function of a pulse instruction is provided by PLCs to enable the outputs for one scan time if the input conditions are met.

R

RAM (RANDOM ACCESS MEMORY)
Memory that is writable and readable at anytime. PLCs incorporate RAMs. RAMs have a backup battery.

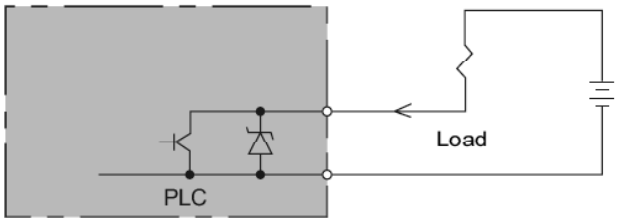
READ
Refers to displaying the contents of a program stored in memory on a programming panel.
It also means transferring the program from a PLC to A6GPP/PHPs and HPPs (off line mode).

RELAY
Refers to an element with an electromagnetic coil and an open/close contact. A relay transfers signals from another device to the coil, where this coil opens or closes the contact accordingly. With this contact operation a relay energizes another load.
With relays, larger contact current can be applied compared to coil drive current (amplification function).
It is also possible to have two separate power supplies, one for the coil and the other for the output contact circuit (insulation function). Output contacts can be two or more.
Relays are also referred as electromagnetic relay.

REPEAT OPERATION METHOD
→ See CYCLIC OPERATION METHOD

RESET
Refers to performing an initialization.
RST instructions are issued to counters, retentive timers, sub relays, output relays, and so on.

RESPONSE TIME
(1) When a different signal is input to an external input terminal, PLC's logic needs some time to recognize the signal change.
This delay time is called input response time.
(2) When output data is generated inside a PLC, the data does not go out of the PLC until an output terminal is opened or closed. This delay time is called output response time.
For the relay output type, this corresponds with the mechanical operation delay time of the relay.



(3) In addition to input response time and output response time, there is another response delay due to a scan cycle of a PLC.

RETENTIVE TIMER
Refers to a timer that retains its current value even when the time-counting coil is de-energized or a power failure occurs. After it is reenergized, the timer starts counting the remaining time and outputs the results.
To clear a current value or open an output contact, use the RST instruction.

ROM WRITER
Refers to a device used for writing programs to EPROMs. HPPs and A6GPP/PHPs are examples of ROM Writers.

RUN
Refers to a state where a PLC is running.
PLCs output signals according to the types input of signals received.

S

S/W (SOFTWARE)
Refers to a program that controls the hardware's behavior.

SCAN
Refers to the execution of the PLC program from the beginningto the end of the program. The time taken for executing all steps in a program is called scan time (cycle time, operation cycle), which is monitored by a watchdog timer.

SEARCH
Refers to finding a desired instruction within the program.

SELF-DIAGNOSIS FUNCTION
Refers to one of PLC' s functions, by which a PLC detects errors by itself.
Detectable errors with this function include:
(1) Watchdog timer error
(2) SUM check error
(3) Power voltage drop
(4) Battery voltage drop

SELF-MAINTAINING CIRCUIT
Refers to a circuit that can keep coils open or closed.

SENSOR POWER SUPPLY
When a PLC uses proximity and photoelectric switches for inputs, sensors of these switches can be fed with 24VDC from the PLC.
However in the case of a large load, a separate power supply must be installed outside the PLC.

SHIFT
Refers to an instruction that is issued to shift the ON/OFF status of a coil from one auxiliary relay to another sequentially. (In the FX series PLCs, this instruction is include in application instructions.) Auxiliary relays arranged in sequence for such use are called shift registers.

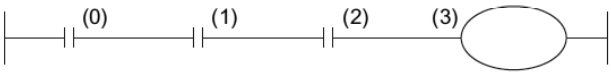
SIMULATION
Refers to testing how a PLC behaves by not using actual devices but through using simulated input switches installed on the PLC.

SOFTWARE TIMER
Refers to timers configured in a PLC program.
→ Also see ANALOG TIMER.

SPECIAL AUXILIARY RELAY
Refers to a type of sub relay provided in a PLC.
Special sub relays are designed for a specific function.
(1) Contact type special relay
In this type of relay, a contact is controlled by the user and a coil is controlled by a sequence program.
(ex.) M8002: Initial pulse
(2) Coil energized type special sub relay
With this type of relay, a PLC takes a certain action in reaction to a coil being energized by user.
(ex.) M8030 = Battery LED turning off an instruction

STATE: S0 to S999
For a step ladder program, there are states (types of contacts) for initial state, general-purpose, and battery backup.
These 100 states, from S900 to S999, are well suited for annunciators. To know what type of error was occurred, users in advance can write an error diagnosis program to drive states S900 to S999, and monitor the special data register D8049.

STEP NUMBER
Refers to numbers assigned to instructions.



1 to 3 steps are needed for each contact or coil. One program can contain from 0 to 1,999 steps (or 0 to 7,999 steps).

STOP
Refers to stopping the PLC.
In the FX series PLCs, turning the RUN input terminal off stops a PLC. Basically, users can program while a PLC is at stop since all outputs are turned off during stop. An exception is when carrying out forced ON/OFF commands.

STORED PROGRAM
If a PLC has memory. This is where the program is stored.

SUM CHECK
A PLC performs a SUM operation, which adds the contents in the program memory in binary digits and stores the results in the register of the PLC. This SUM operation is performed when:
(1) Program panel mode is changed. (Read, write, insert, delete, or other key operation is performed.)
(2) All clear, write, insert, or delete is performed to a program.
(3) A constant is changed during monitoring.
Whether a PLC status corresponds to the results of the above operation is checked when;
(1) Power-on
(2) A sum check is performed on the programming panel.
(3) A PLC enters RUN state.
This check is called SUM check.
This check is to know if the contents of the program are changed or not. If any changes are found, the PROG E LED on PLC' s surface flashes, and then the PLC stops.

SURGE ABSORBER
Refers to elements that are designed to absorb surge.
A condenser and resistor are used in a SSR for outputs, which may cause open circuit leakage current. Surge absorbers are also called surge killers.

SURGE
Refers to abnormal noises. There also is a term called surge voltage, which means high voltage that is instantaneously generated when current in a coil is shut off. This surge voltage may severely damage semiconductors or shorten service lives of contacts.
It also may case malfunction of a PLC due to noise.

T

TIMER
Refers to a relay whose contact is opened or closed a certain time after its coil is energized.
→ Also see OFF DELAY TIMER
→ Also see ON DELAY TIMER
The FX series PLCs incorporate delay timers in 0.1 seconds or 0.01 seconds units.

TRANSISTOR OUTPUT
Refers to a no-contact output for DC loads. A transistor is used by a PLC for output instead of relay contacts.

TRIAC OUTPUT
Refers to no-contact output for AC loads. A triac is used by a PLC for output instead of relay contacts.
Note that using a triac with micro load may cause open circuit leakage current to occur. Triac output is also called SSR output.

U

UPLOAD
Refers to reading and transferring programs from a PLC to peripheral devices such as A6GPP, PHP, HPP (off line mode), etc.
→ Also see Download.

USER'S MEMORY
Refers to program memory storing user-created programs that are needed for a PLC operation.

W

WATCHDOG TIMER
A timer that detects PLC operation errors. This timer monitors the scan time of a program, and issues an alert if a scan failed to complete within the set time.
FX series PLC's watchdog timer illuminates CPU E provided on the PLC's surface. When this occurs, all outputs from the PLC are automatically turned off.

WRITE
Refers to an action of saving programs to memory.
To do this, write programs to a PLC from a programming panel or write and transfer programs using a A6GPP/PHP.

MITSUBISHI ELECTRIC CORPORATION
HEAD OFFICE: TOKYO BUILDING, 2-7-3 MARUNOUCHI, CHIYODA-KU, TOKYO 100-8310, JAPAN
HIMEJI WORKS: 840, CHIYODA CHO, HIMEJI, JAPAN

JY997D26101A
(MEE)

Effective
Specifications are subject to change without notice.