MITSUBISHI ELECTRIC

Programmable Controller

MELSEC iQ-F series

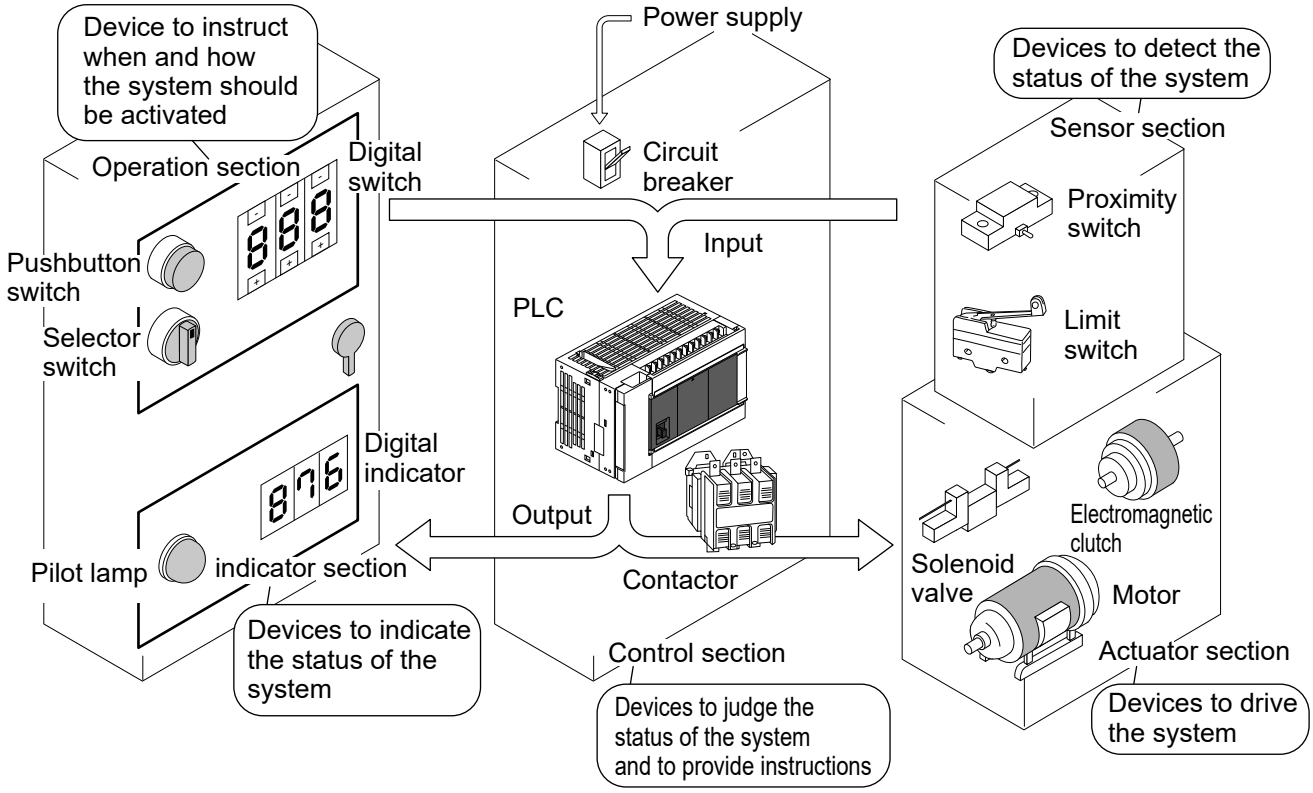
FX5U Training Manual

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

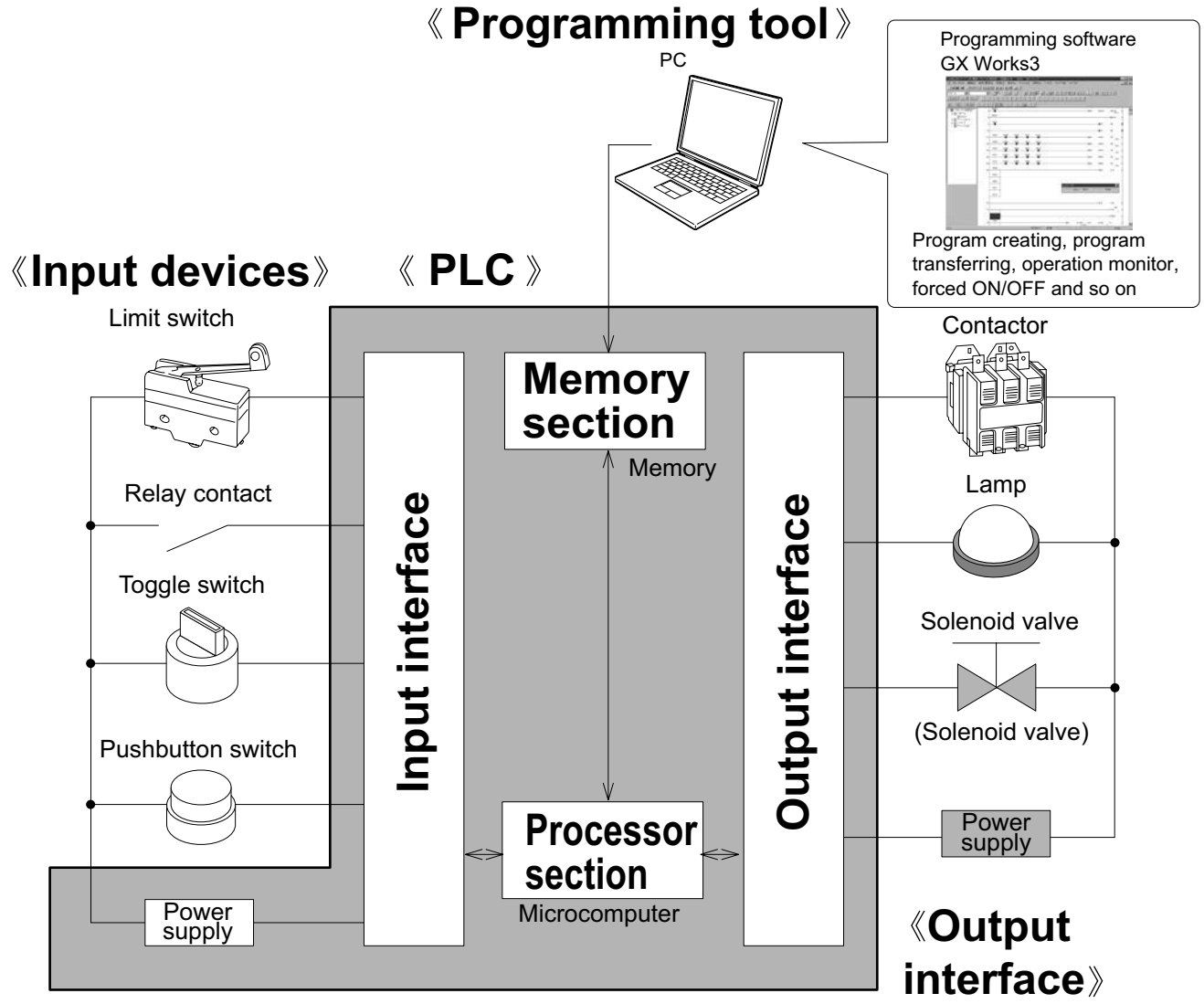
PLCs are broadly used as core components for FA (Factory Automation) and as electronic application products essential for saving labor costs and improving automation.
PLCs can be used for many types of applications such as systematical applications which supply control all over a factory or as standalone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



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.

Mechanism of PLC

The PLC is a microcomputer for industrial purposes.



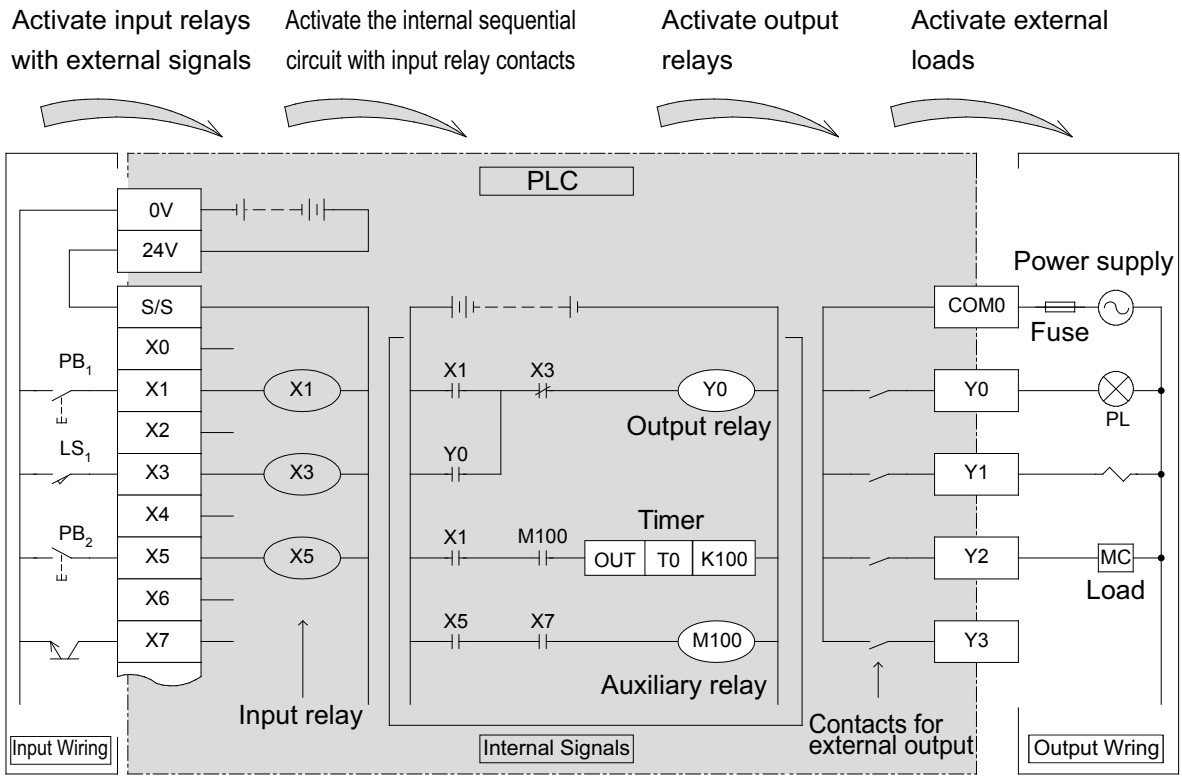
A PLC incorporates an electrical circuit mainly comprised of a microcomputer and memory.
Input/output interfaces exist between input/output devices and the electronic circuit to connect them.
The programming panel is used to write a program to the memory in the PLC.

Reference

The term "sequencer"

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.

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



The PLC is an electronic device mainly comprised of a microcomputer.
However in effect…
The user does not need any knowledge of a microcomputer to operate a PLC and it can be regarded as an aggregate of relays, timers, and counters.

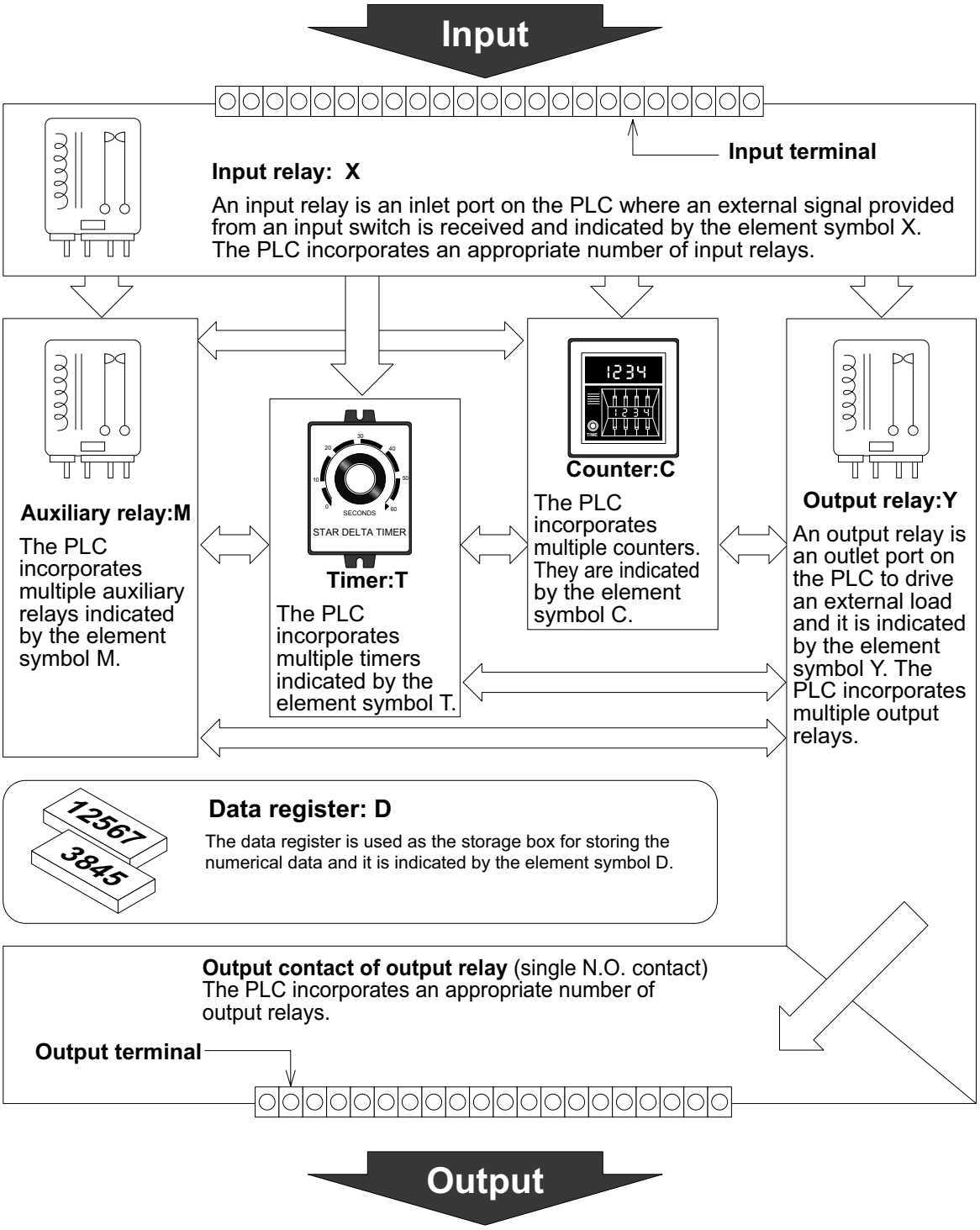
Internal operation of PLC

Signal Flow of PLC

- When the pushbutton switch PB1 is pushed, the coil of the input relay X1 is energized.
- When the coil of the input relay X1 is energized, the N.O. contact of X1 is closed and the coil of the output relay Y0 is energized.
- When the coil of the output relay Y0 is energized, the contact of Y0 is closed, then the pilot lamp PL is illuminated.
- When the pushbutton switch PB1 is released, the coil of the input relay X1 is de-energized and the N.O. contact of X1 is opened.
  But the output relay Y0 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 X3 is opened, then the coil of the output relay Y0 is de-energized (Reset).
  As a result, the pilot lamp; PL distinguishes and the self-maintaining action of the output relay Y0 is cleared.

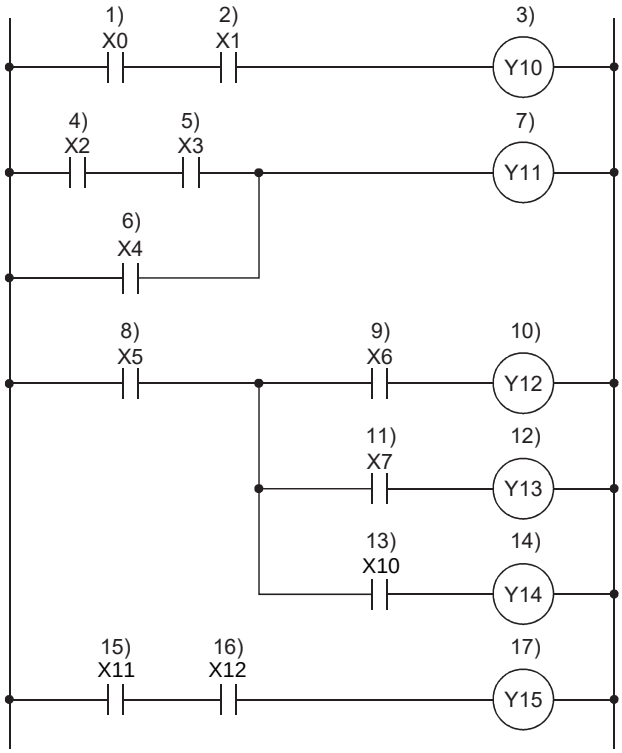
Types of relay and timers

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.



Program processing sequence

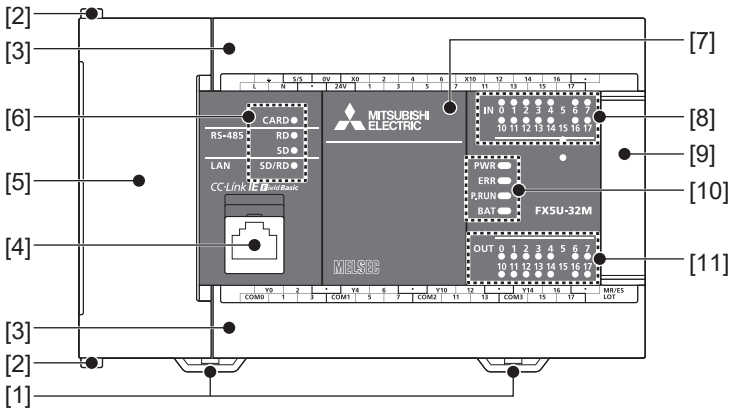
A CPU module executes operations in series from the start step of the program memory from left to right and from top to bottom (in the order of 1), 2) ... and 17)) in units of a ladder block as shown below.



FX5U Hardware

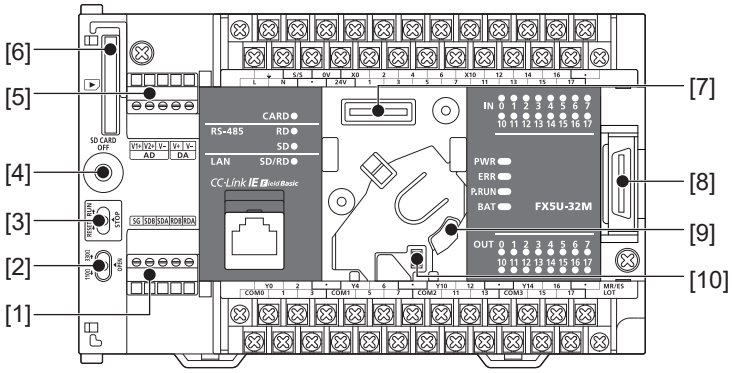
Part Names

Front panel



No.	Name	Description
[1]	DIN rail mounting hooks	Hook for mounting the CPU module on a DIN rail of DIN46277 (35 mm wide).
[2]	Expansion adapter connecting hooks	When connecting an expansion adapter, secure it with these hooks.
[3]	Terminal block cover	Cover for protecting the terminal block. The cover can be opened for wiring. Keep the covers closed while equipment is running (power is on).
[4]	Built-in Ethernet communication connector	Connector for connection with Ethernet-compatible devices. (with cover) For details, refer to MELSEC iQ-F FX5 User's Manual (Ethernet Communication).
[5]	Top cover	Cover for protecting the SD memory card slot, the RUN/STOP/RESET switch, and others. The built-in RS-485 communication terminal block, built-in analog I/O terminal block, RUN/STOP/RESET switch, SD memory card slot, and others are located under this cover.
[6]	CARD LED	Indicates whether an SD memory card can be used or not. Lit: Can be used or cannot be removed. Flashing: In preparation Off: Not inserted or can be removed.
	RD LED	Lit when the CPU module is receiving data through built-in RS-485 communication.
	SD LED	Lit when the CPU module is sending data through built-in RS-485 communication.
	SD/RD LED	Lit when the CPU module is sending or receiving data through built-in Ethernet communication.
[7]	Expansion board connector cover	Cover for protecting expansion board connectors, battery, or others. Connect the battery under this cover.
[8]	Input display LED	Lit when input is on.
[9]	Extension connector cover	Cover for protecting the extension connector. Connect the extension cable of an extension module to the extension connector under the cover.
[10]	PWR LED	Indicates whether the CPU module is powered or not. Lit: Powered Off: Not powered or hardware error
	ERR LED	Indicates the error status of the CPU module. Lit: Error or hardware error Flashing: Factory default setting, error, hardware error, or resetting Off: Operating normally
	P.RUN LED	Indicates the program running status. Lit: Operating normally Flashing: Paused Off: Stopped or stop error
	BAT LED	Indicates the battery status. Flashing: Battery error Off: Operating normally
[11]	Output display LED	Lit when output is on.

With cover open

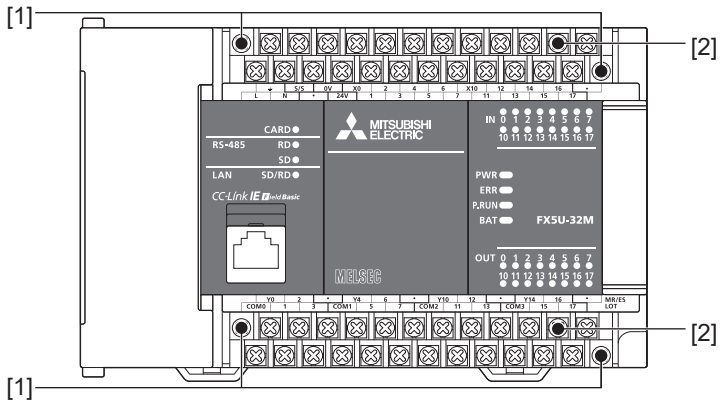


No.	Name	Description
[1]	Built-in RS-485 communication terminal block	Terminal block for connection with RS-485-compatible devices
[2]	RS-485 terminal resistor selector switch	Switch for switching terminal resistance for built-in RS-485 communication.
[3]	RUN/STOP/RESET switch	Switch for operating the CPU module. RUN: Runs the program STOP: Stops the program RESET: Resets the CPU module (hold the switch on the RESET side for approximately 1 second.)
[4]	SD memory card disable switch	Switch for disabling access to the SD memory card when the card is to be removed.
[5]	Built-in analog I/O terminal block	Terminal block for using the built-in analog function.
[6]	SD memory card slot	Slot for inserting an SD memory card.
[7]	Expansion board connector	Connector for connecting an expansion board.
[8]	Extension connector	Connector for connecting the extension cable of an extension module.
[9]	Battery holder	Holder for storing an optional battery.
[10]	Battery connector	Connector for connecting an optional battery.

Point

Use a tool such as a screwdriver to operate RS-485 terminal resistor selector switch.
Make sure that the edge of the tool does not damage the switch or the case.

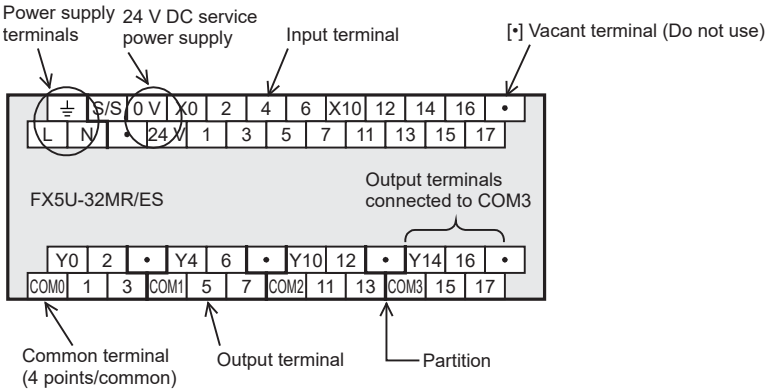
When the terminal block covers are open



No.	Name	Description
[1]	Terminal block mounting screws	Gradually loosen the left and right screws (alternately), and remove the top of the terminal blocks.
[2]	Terminal	Terminals for power, input, and output.

Power, input/output terminal block

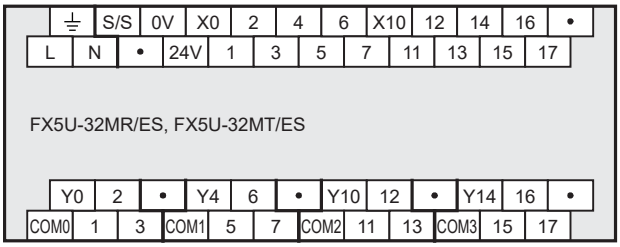
■Interpretation of terminal block layout



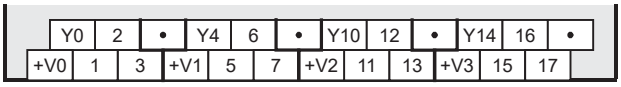
- Indication of power supply terminals
  The AC power supply type has [L] and [N] terminals, whereas the DC power type has [+] and [-] terminals.
- Indication of 24 V DC service power supply
  The AC power supply type has [0V] and [24V] terminals.
  The DC power supply type does not have a 24 V DC service power supply. [·] indicates open terminals.
  Do not wire [·] terminals.
- Indication of input terminal
  Both AC and DC power supply type have the same input terminals, however, the external input wiring differs from each other.
- Indication of output terminals connected to common (COM□)
  One common terminal covers 4 or 8 output points.
  The output number (Y) connected to common is the range inside the thick "separation line."
  For transistor output (source) type, the "COM□" terminal is the "+V□" terminal.

■ FX5U-32M□

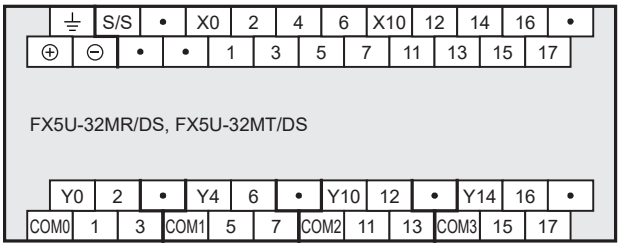
• AC power supply type



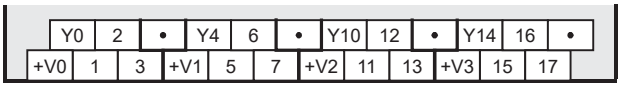
FX5U-32MT/ESS



• DC power supply type



FX5U-32MT/DSS



Power Supply Specifications

The CPU module power supply specifications are explained below.

AC power supply type

Item		Specifications
Rated voltage		100 to 240 V AC
Allowable supply voltage range		85 to 264 V AC
Frequency rating		50/60 Hz
Allowable instantaneous power failure time		Operation can be continued upon occurrence of instantaneous power failure for 10 ms or less. When the supply voltage is 200 V AC, the time can be change to 10 to 100 ms by editing the user program.
Power fuse	FX5U-32M□/E□	250 V, 3.15 A Time-lag fuse
	FX5U-64M□/E□, FX5U-80M□/E□	250 V, 5 A Time-lag fuse
Rush current	FX5U-32M□/E□	25 A max. 5 ms or less/100 V AC 50 A max. 5 ms or less/200 V AC
	FX5U-64M□/E□, FX5U-80M□/E□	30 A max. 5 ms or less/100 V AC 60 A max. 5 ms or less/200 V AC
Power consumption*1	FX5U-32M□/E□	30 W
	FX5U-64M□/E□	40 W
	FX5U-80M□/E□	45 W
24 V DC service power supply capacity*2	FX5U-32M□/E□	400 mA [300 mA*3] (Supply capacity when service power supply is used for input circuit of the CPU module)
		480 mA [380 mA*3] (Supply capacity when external power supply is used for input circuit of the CPU module)
	FX5U-64M□/E□	600 mA [300 mA*3] (Supply capacity when service power supply is used for input circuit of the CPU module)
		740 mA [440 mA*3] (Supply capacity when external power supply is used for input circuit of the CPU module)
	FX5U-80M□/E□	600 mA [300 mA*3] (Supply capacity when service power supply is used for input circuit of the CPU module)
		770 mA [470 mA*3] (Supply capacity when external power supply is used for input circuit of the CPU module)
5 V DC built-in power supply capacity	FX5U-32M□/E□	900 mA
	FX5U-64M□/E□, FX5U-80M□/E□	1100 mA

*1 This item shows value when all 24 V DC service power supplies are used in the maximum configuration connectable to the CPU module. (The current of the input circuit is included.)

*2 When I/O modules are connected, they consume current from the 24 V DC service power.

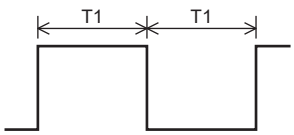
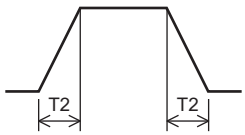
*3 The value in [] is capacity of 24 V DC power supply in the case where operating ambient temperature is lower than 0℃.

Input Specifications

The CPU module input specifications are explained below.

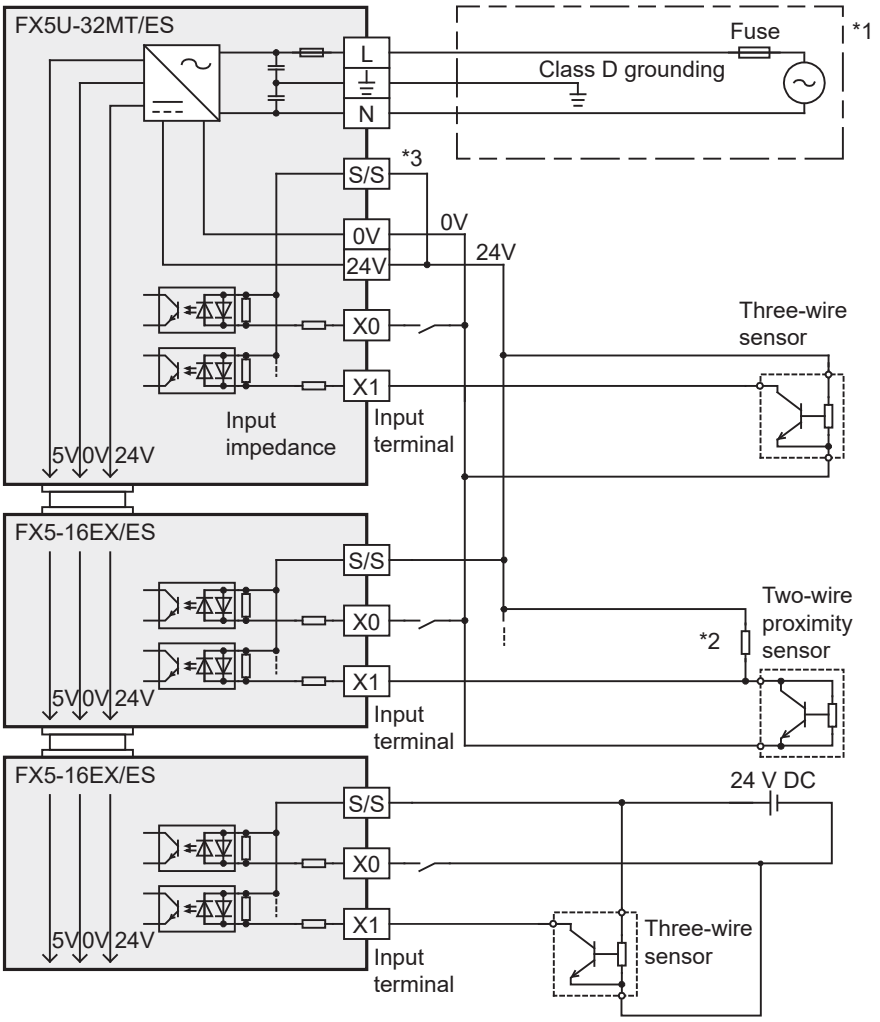
24 V DC Input (sink/source)

The input points in the table below indicate the CPU module terminal points.

<table>
<tr><th colspan="3">Item</th><th colspan="2">Specifications</th></tr>
<tr><td rowspan="3">No. of input points</td><td colspan="2">FX5U-32M☐</td><td colspan="2">16 points</td></tr>
<tr><td colspan="2">FX5U-64M☐</td><td colspan="2">32 points</td></tr>
<tr><td colspan="2">FX5U-80M☐</td><td colspan="2">40 points</td></tr>
<tr><td colspan="3">Connection type</td><td colspan="2">Removable terminal block (M3 screws)</td></tr>
<tr><td colspan="3">Input type</td><td colspan="2">Sink/source</td></tr>
<tr><td colspan="3">Input signal voltage</td><td colspan="2">24 V DC +20 %, -15%</td></tr>
<tr><td rowspan="2">Input signal current</td><td colspan="2">X0 to X17</td><td colspan="2">5.3 mA/24 V DC</td></tr>
<tr><td colspan="2">X20 and subsequent</td><td colspan="2">4.0 mA/24 V DC</td></tr>
<tr><td rowspan="2">Input impedance</td><td colspan="2">X0 to X17</td><td colspan="2">4.3 kΩ</td></tr>
<tr><td colspan="2">X20 and subsequent</td><td colspan="2">5.6 kΩ</td></tr>
<tr><td rowspan="2">ON input sensitivity current</td><td colspan="2">X0 to X17</td><td colspan="2">3.5 mA or more</td></tr>
<tr><td colspan="2">X20 and subsequent</td><td colspan="2">3.0 mA or more</td></tr>
<tr><td colspan="3">OFF input sensitivity current</td><td colspan="2">1.5 mA or less</td></tr>
<tr><td rowspan="5">Input response frequency</td><td>FX5U-32M☐</td><td>X0 to X5</td><td colspan="2" rowspan="2">200 kHz
When capturing pulses of a response frequency of 50 to 200 kHz, in the case of capturing high-speed pulses.</td></tr>
<tr><td>FX5U-64M☐,
FX5U-80M☐</td><td>X0 to X7</td></tr>
<tr><td>FX5U-32M☐</td><td>X6 to X17</td><td colspan="2" rowspan="2">10 kHz</td></tr>
<tr><td>FX5U-64M☐,
FX5U-80M☐</td><td>X10 to X17</td></tr>
<tr><td>FX5U-64M☐,
FX5U-80M☐</td><td>X20 and subsequent</td><td colspan="2">0.1±0.05 kHz</td></tr>
<tr><td rowspan="6">Pulse waveform</td><td colspan="2" rowspan="2">Waveform</td><td> 
</td><td> 
</td></tr>
<tr><td>T1 (pulse width)</td><td>T2 (rise/fall time)</td></tr>
<tr><td>FX5U-32M☐</td><td>X0 to X5</td><td rowspan="2">2.5 μs or more</td><td rowspan="2">1.25 μs or less</td></tr>
<tr><td>FX5U-64M☐,
FX5U-80M☐</td><td>X0 to X7</td></tr>
<tr><td>FX5U-32M☐</td><td>X6 to X17</td><td rowspan="2">50 μs or more</td><td rowspan="2">25 μs or less</td></tr>
<tr><td>FX5U-64M☐,
FX5U-80M☐</td><td>X10 to X17</td></tr>
<tr><td rowspan="5">Input response time
(H/W filter delay)</td><td>FX5U-32M☐</td><td>X0 to X5</td><td colspan="2" rowspan="2">ON: 2.5 μs or less
OFF: 2.5 μs or less</td></tr>
<tr><td>FX5U-64M☐,
FX5U-80M☐</td><td>X0 to X7</td></tr>
<tr><td>FX5U-32M☐</td><td>X6 to X17</td><td colspan="2" rowspan="2">ON: 30 μs or less
OFF: 50 μs or less</td></tr>
<tr><td>FX5U-64M☐,
FX5U-80M☐</td><td>X10 to X17</td></tr>
<tr><td>FX5U-64M☐,
FX5U-80M☐</td><td>X20 and subsequent</td><td colspan="2">ON: 50 μs or less
OFF: 150 μs or less</td></tr>
<tr><td colspan="3">Input response time
(Digital filter setting value)</td><td colspan="2">None, 10 μs, 50 μs, 0.1 ms, 0.2 ms, 0.4 ms, 0.6 ms, 1 ms, 5 ms, 10 ms (initial values), 20 ms, 70 ms
When using this product in an environment with much noise, set the digital filter.</td></tr>
<tr><td colspan="3">Input signal format</td><td colspan="2">No-voltage contact input
Sink: NPN open collector transistor
Source: PNP open collector transistor</td></tr>
<tr><td colspan="3">Input circuit insulation</td><td colspan="2">Photo-coupler insulation</td></tr>
<tr><td colspan="3">Indication of input operation</td><td colspan="2">LED is lit when input is on</td></tr>
</table>

Input wiring example

Sink input [AC power supply type]



*1 Handle the power supply circuit properly in accordance with "Power Supply Wiring."

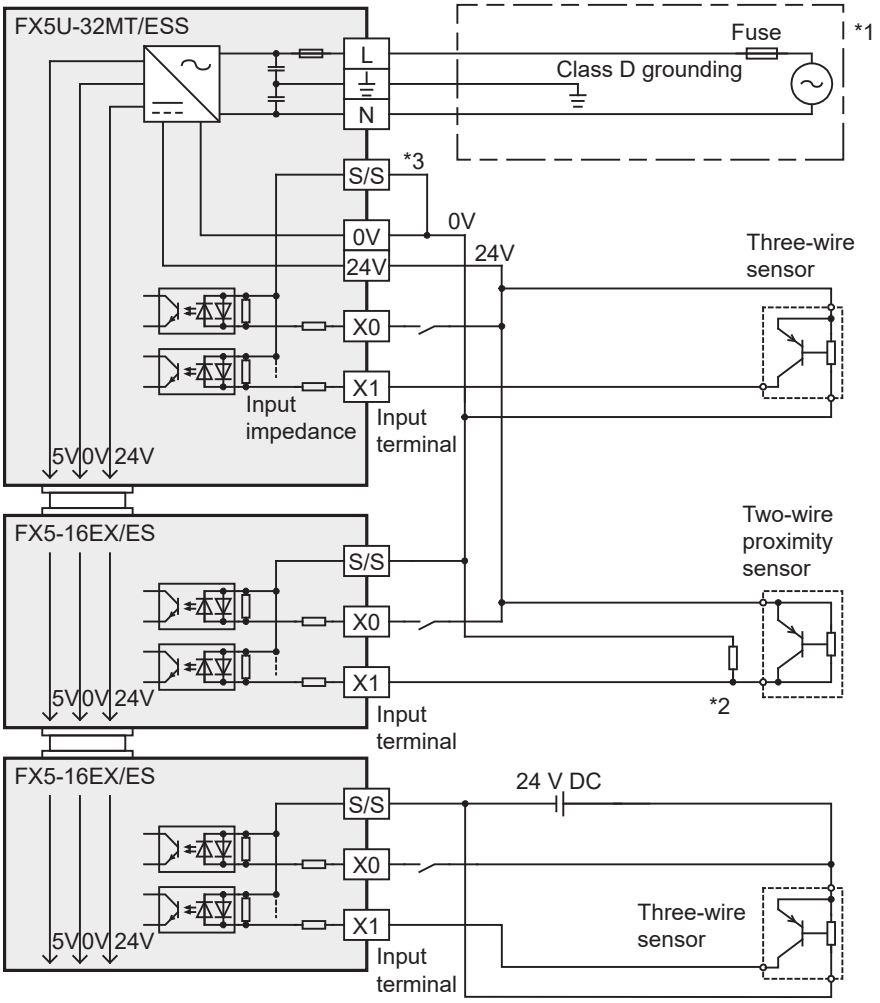
*2 For an input device having a parallel resistance or a two-wire proximity switch, a bleeder resistance may be required.

*3 In the case of sink input wiring, short-circuit the [S/S] and [24V] terminals of the CPU module.

Point

The service power supply or external power supply can be used for all inputs (X) of the CPU module. However you need to select either one for each CPU module or I/O module. Both power supplies cannot be used together in the same module.

Source input [AC power supply type]



*1 Handle the power supply circuit properly in accordance with "Power Supply Wiring."

*2 For an input device having a parallel resistance or a two-wire proximity switch, a bleeder resistance may be required.

*3 In the case of source input wiring, short-circuit the [S/S] and [0V] terminals of the CPU module.

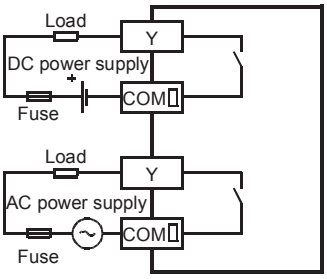
Point

The service power supply or external power supply can be used for all inputs (X) of the CPU module. However you need to select either one for each CPU module or I/O module. Both power supplies cannot be used together in the same module.

Output Specifications

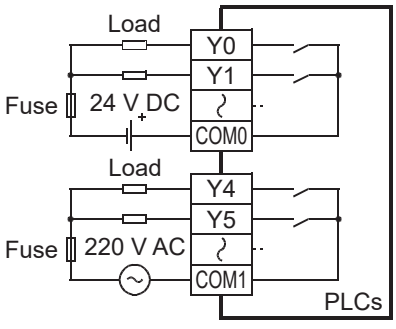
The CPU module output specifications are explained below.

Relay output

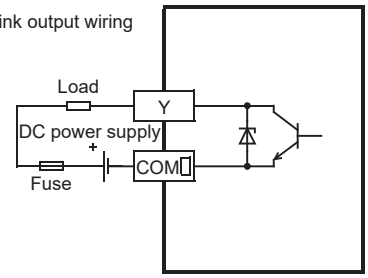
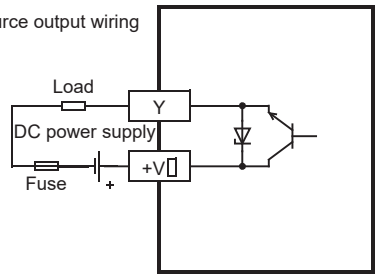
Item		Output Specifications
No. of output points	FX5U-32MR/□	16 points
	FX5U-64MR/□	32 points
	FX5U-80MR/□	40 points
Connection type		Removable terminal block (M3 screws)
Output type		Relay
External power supply		30 V DC or less 240 V AC or less ("250 V AC or less" if not a CE, UL, cUL compliant item)
Max. load		2 A/point The total load current per common terminal should be the following value. • 4 output points/common terminal: 8 A or less • 8 output points/common terminal: 8 A or less
Min. load		5 V DC, 2 mA (reference values)
Open circuit leakage current		—
Response time	OFF→ON	Approx. 10 ms
	ON→OFF	Approx. 10 ms
Insulation of circuit		Mechanical insulation
Indication of output operation		LED is lit when output is on
Output circuit configuration		 A number is entered in the □ of [COM □].

One common terminal is used for 4 or 8 relay output points.

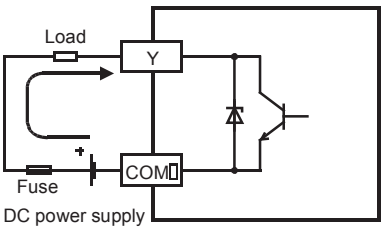
The common terminal blocks can drive loads of different circuit voltage systems (for example, 220 V AC and 24 V DC).



Transistor output

Item		Output specifications
No. of output points	FX5U-32MT/□	16 points
	FX5U-64MT/□	32 points
	FX5U-80MT/□	40 points
Connection type		Removable terminal block (M3 screws)
Output type	FX5U-□MT/□S	Transistor/sink output
	FX5U-□MT/□SS	Transistor/source output
External power supply		5 to 30 V DC
Max. load		0.5 A/point The total load current per common terminal should be the following value. • 4 output points/common terminal: 0.8 A or less • 8 output points/common terminal: 1.6 A or less
Open circuit leakage current		0.1 mA or less/30 V DC
Voltage drop when ON	Y0 to Y3	1.0 V or less
	Y4 and subsequent	1.5 V or less
Response time	Y0 to Y3	2.5 µs or less/10 mA or more (5 to 24 V DC)
	Y4 and subsequent	0.2 ms or less/200 mA or more (24 V DC)
Insulation of circuit		Photo-coupler insulation
Indication of output operation		LED is lit when output is on
Output circuit configuration		Sink output wiring  A number is entered in the □ of [COM□]. Source output wiring  A number is entered in the □ of [+V□].

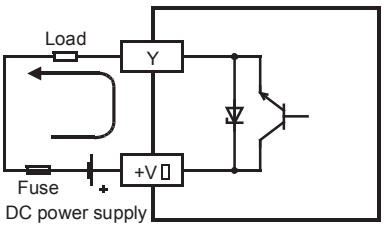
- Sink output [-common]
  Output to make load current flow into the output (Y) terminal is called sink output.



A number is entered in the □ of [COM□].

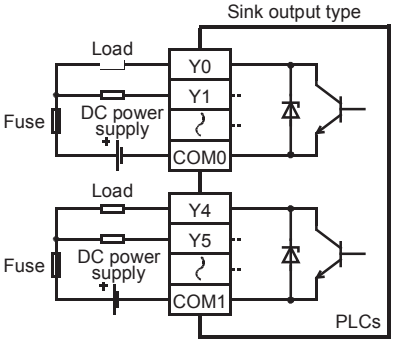
Connect each COM□ (number) terminal to the minus side of the load power supply.
The COM□ terminals are not connected internally.

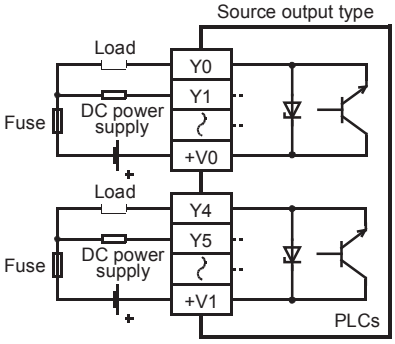
- Source output [+common]
  Output to make load current flow out of the output (Y) terminal is called source output.



A number is entered in the □ of [+V□].

Connect +V□ (number) terminal to the plus side of the load power supply.
The +V□ terminals are not connected internally.





The transistor output type is a 4-point or 8-point common output type.

Built-in Analog Specifications

The analog input/output specifications of the built-in analog function are explained below.

For details on the analog built-in function, refer to 📖MELSEC iQ-F FX5 User's Manual (Analog Control).

Analog input

Item		Specifications
Analog input points		2 points (2 channels)
Analog input	Voltage	0 to 10 V DC (input resistance 115.7 kΩ)
Digital output		Unsigned 12-bit binary
Device allocation		SD6020 (Input data of ch1) SD6060 (Input data of ch2)
I/O characteristics, Maximum resolution	Digital output value	0 to 4000
	Maximum resolution	2.5 mV
Accuracy (Accuracy for the full scale of the digital output value)	Ambient temperature 25 ±5℃	Within ±0.5% (±20 digit*2)
	Ambient temperature 0 to 55℃	Within ±1.0% (±40 digit*2)
	Ambient temperature -20 to 0℃*1	Within ±1.5% (±60 digit*2)
Conversion speed		30 μs /channels (data refreshed every operation cycle)
Absolute maximum input		-0.5 V, +15 V
Insulation method		Inside the CPU module and the analog input circuit are not insulated. Between input terminals (channels) is not insulated.
Occupied points		0 points (does not pertain to the max. No. of input/output points of the CPU module.)
Terminal block used		European-type terminal block

*1 Products manufactured before June 2016 do not adapt to this specification.

*2 "Digit" refers to digital values.

Analog output

Item		Specifications
Analog output points		1 points (1 channels)
Digital input		Unsigned 12-bit binary
Analog output	Voltage	0 to 10 V DC (external load resistance 2 k to 1 MΩ)
Device allocation		SD6180 (Output setting data of ch1)
I/O characteristics, Maximum resolution*1	Digital input value	0 to 4000
	Maximum resolution	2.5 mV
Accuracy*2 (Accuracy for the full scale of the analog output value)	Ambient temperature 25 ±5℃	Within ±0.5% (±20 digit*4)
	Ambient temperature 0 to 55℃	Within ±1.0% (±40 digit*4)
	Ambient temperature -20 to 0℃*3	Within ±1.5% (±60 digit*4)
Conversion speed		30 μs (data refreshed every operation cycle)
Insulation method		Inside the CPU module and the analog output circuit are not insulated.
Occupied points		0 points (does not pertain to the max. No. of input/output points of the CPU module.)
Terminal block used		European-type terminal block

*1 There is a dead band near 0 V output, which is an area where some analog output values do not reflect digital input values.

*2 External load resistance is set to 2 kΩ when shipped from the factory. Thus, output voltage will increase somewhat if the resistance is set higher than 2 kΩ. When the resistance is 1 MΩ, output voltage increases maximum 2%.

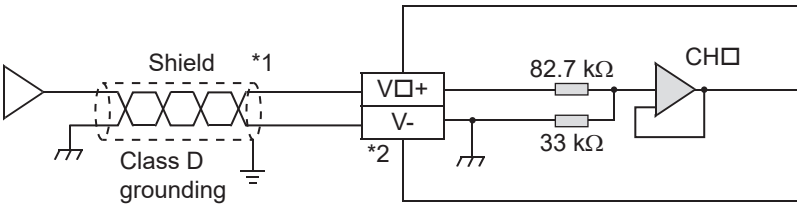
*3 Products manufactured before June 2016 do not adapt to this specification.

*4 "Digit" refers to digital values.

Analog Wiring

Wiring to the built-in analog I/O terminals of the CPU module is explained below.

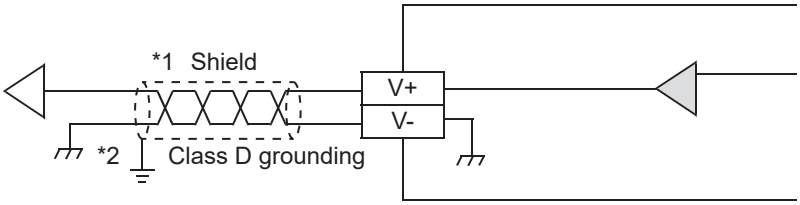
Analog input wiring



*1 For analog input wiring, use shielded twisted-pair cables (double-core type). Separate them from other power lines or lines which can be induced by others.

*2 For unused channels, short-circuit the "V☐+" and "V-" terminals.

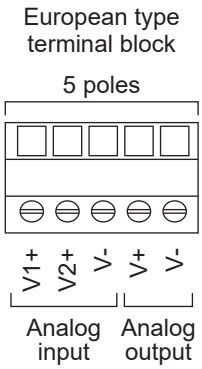
Analog output wiring



*1 For analog output wiring, use shielded twisted-pair cables (double-core type). Separate them from other power lines or lines which can be induced by others.

*2 Ground a shielded cable at one point on the signal receiving side.

Built-in analog terminal



Communication Specifications

The built-in Ethernet and built-in RS-485 communication specifications are as explained below.

Built-in Ethernet communication

For details of built-in Ethernet communication, refer to the following.

📖MELSEC iQ-F FX5 User's Manual (Ethernet Communication)

📖MELSEC iQ-F FX5 User's Manual (SLMP)

Item		Specifications
Data transmission speed		100/10 Mbps
Communication mode		Full-duplex (FDX) / Half-duplex (HDX)*1
Interface		RJ45 connector
Transmission method		Base band
Maximum segment length (The distance between hub and node)		100 m
Cascade connection	100BASE-TX	Max. 2 stages*2
	10BASE-T	Max. 4 stages*2
Protocol type		CC-Link IE field network Basic
		MELSOFT connection
		SLMP (3E frame)
		Socket communication
		Predefined protocol support
		FTP server
Number of simultaneously open connections allowed		Total of 8 connections*3*4 (Up to 8 external devices can access one CPU module at the same time.)
Hub*1		Hubs with 100BASE-TX or 10BASE-T ports can be used.
Insulation method		Pulse transformer insulation
IP address*5		Initial value: 192.168.3.250
Cable used*6	For 100BASE-TX connection	Ethernet standard-compatible cable, category 5 or higher (STP cable)
	For 10BASE-T connection	Ethernet standard-compatible cable, category 3 or higher (STP cable)

*1 IEEE802.3x flow control is not supported.

*2 Number of stages that can be connected when a repeater hub is used. When a switching hub is used, check the specifications of the switching hub used.

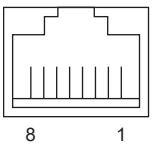
*3 The first device for MELSOFT connection is not included in the number of connections. (The second and the following devices are included.)

*4 The CC-Link IE field network Basic and FTP server are not included in the number of connections.

*5 If the first octet is 0 or 127, a parameter error (2222H) will occur. (Example: 0.0.0.0, 127.0.0.0, etc.)

*6 A straight cable can be used. If a personal computer or GOT and CPU module are directly connected a cross cable can be used.

Built-in Ethernet connector



Pin	Signal name
1	TXD+
2	TXD-
3	RXD+
4	Not used
5	Not used
6	RXD-
7	Not used
8	Not used

Built-in RS-485 communication

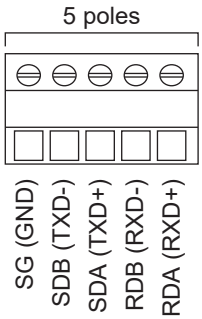
For details of built-in RS-485 communication, refer to the following.

📖MELSEC iQ-F FX5 User's Manual (Serial Communication)
📖MELSEC iQ-F FX5 User's Manual (MELSEC Communication Protocol)
📖MELSEC iQ-F FX5 User's Manual (MODBUS Communication)

Item	Specifications
Transmission standards	Conforms to RS-485/RS-422 specifications
Data transmission speed	Max. 115.2 kbps
Communication method	Full-duplex (FDX) / Half-duplex (HDX)
Maximum total extension distance	50 m
Protocol type	MELSOFT connection
	Non-protocol communication
	MELSEC Communication protocol (3C/4C frames)
	MODBUS RTU
	Predefined protocol support
	Inverter communication
	N:N network
Insulation method	Not insulated
Terminal resistors	Built-in (OPEN/110 Ω/330 Ω)
Terminal block used	European-type terminal block

Built-in RS-485 terminal

European type terminal block



OPERATING GX Works3

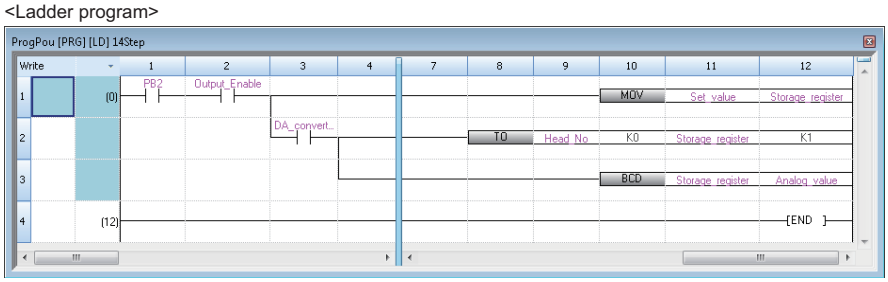
Main Functions of GX Works3

GX Works3 manages programs and parameters in a project for each CPU module. It has the following main functions.

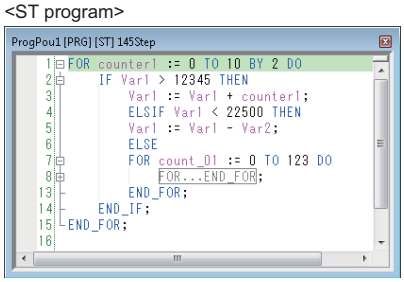
Creating programs

Users can create programs in a desired programming language, depending on the processing.

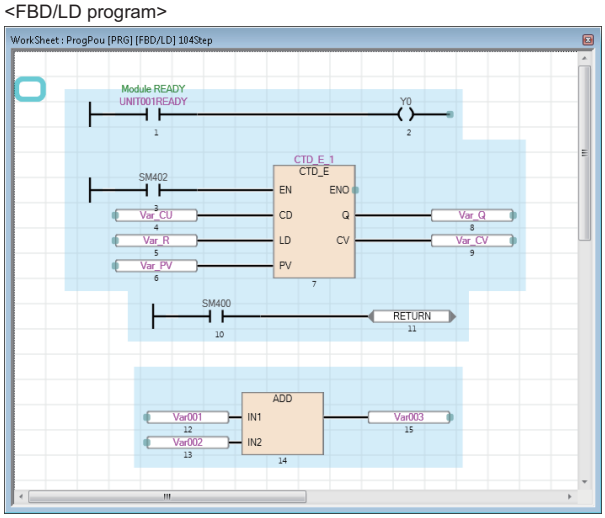
<Ladder program>



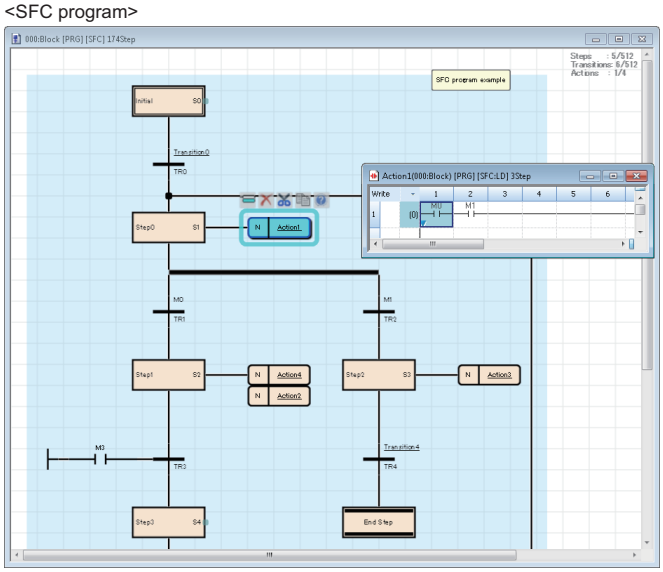
<ST program>



<FBD/LD program>

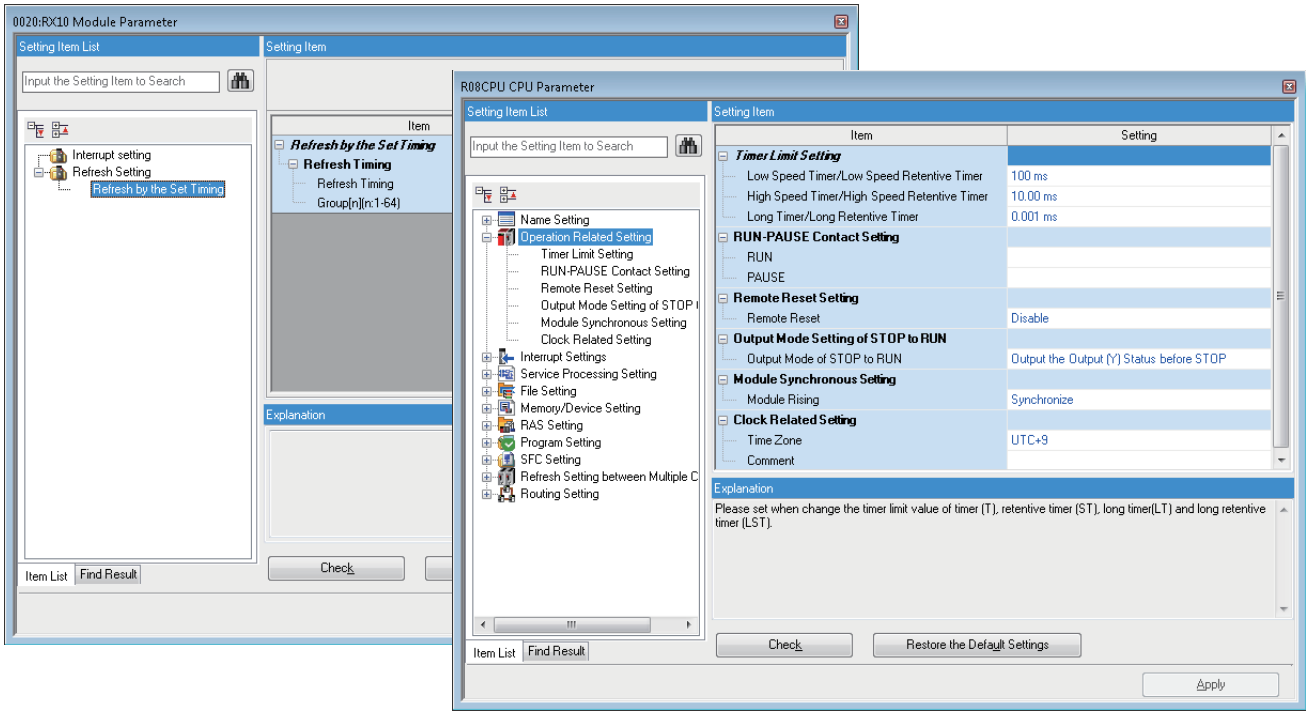


<SFC program>



Setting parameters

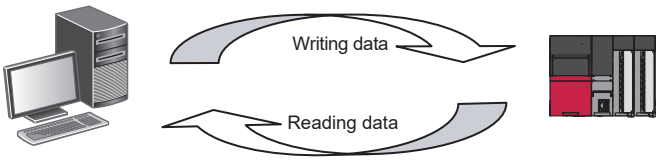
Users can set parameters for CPU modules, I/O modules, and intelligent function modules.



Reading/writing data from/to the CPU module

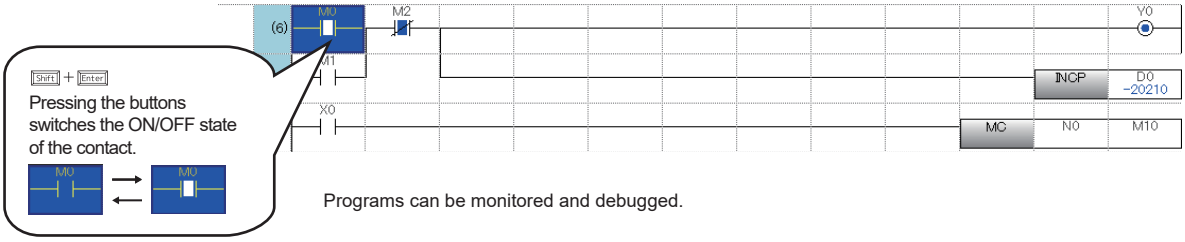
Users can read/write created sequence programs from/to the CPU module by using the "Write to PLC" and "Read from PLC" functions.

Users can edit sequence programs with the online change function even while the CPU module is in the RUN state.



Monitoring and debugging programs

Users can write created sequence programs to the CPU module and monitor data during operation, such as device values.



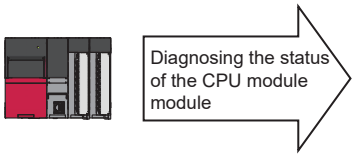
Programs can be monitored and debugged.

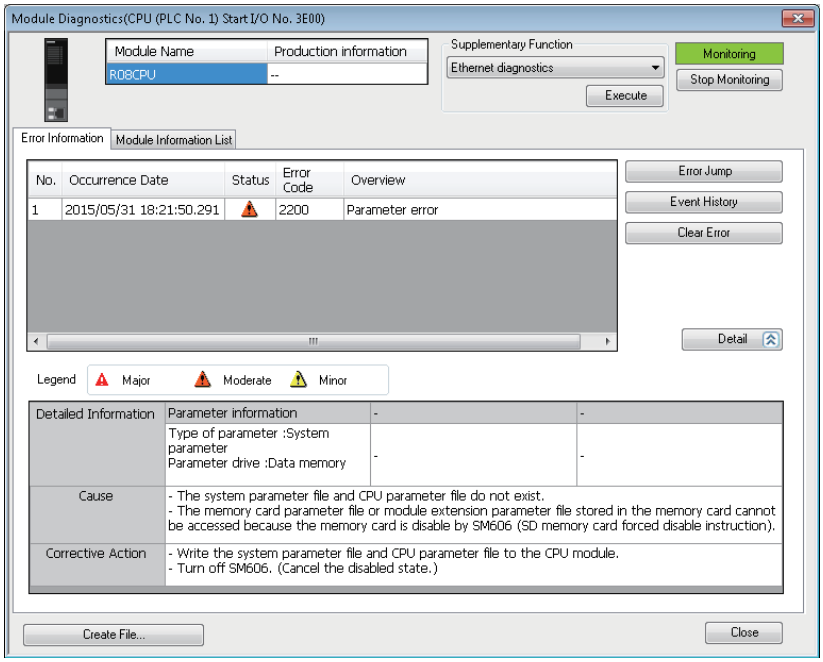
Diagnostic function

GX Works3 make diagnoses on the current error status and error history of the CPU module or network. With the diagnostic function, system recovery can be completed in a short time.

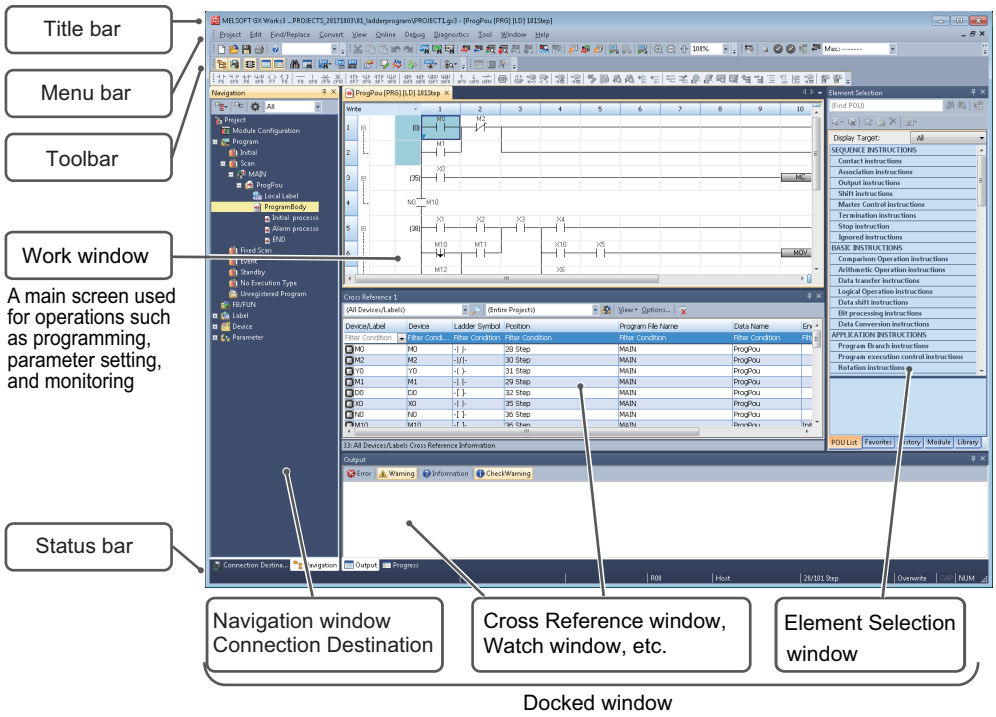
The system monitor shows detailed information on intelligent function modules and others. This feature helps users to shorten the time taken for system recovery when an error occurs.

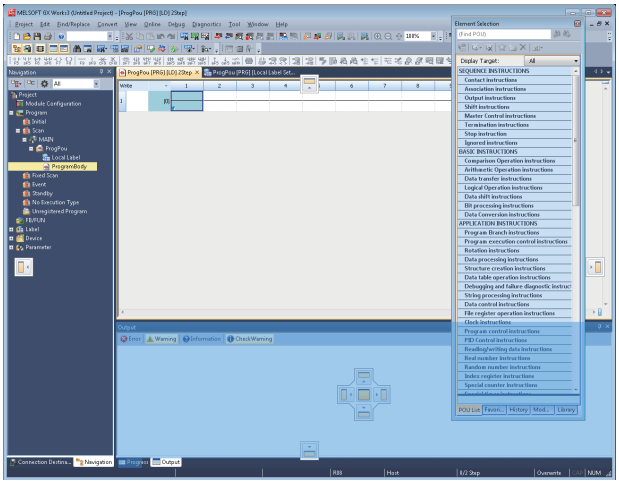
Diagnosing the CPU module module ("Module Diagnostics" window)





Screen Configuration





Docking/floating dockable windows

- Docked windows can be switched between the docked display and the floating display by double-clicking the title bar.
- Floating display: Drag the title bar of a docked window and drop it to the arbitrary position to float from the main frame.

Starting GX Works3

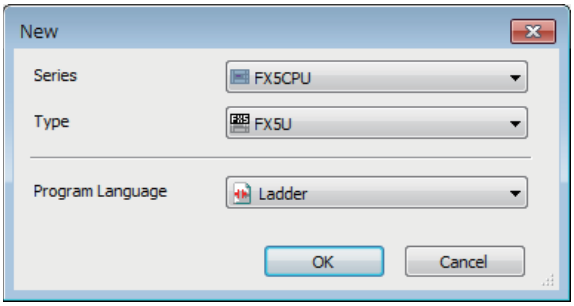
Select [MELSOFT] ⇨ [GX Works3] ⇨ [GX Works3] from the Windows® Start menu*1.

*1 Select [Start] ⇨ [All apps] or [Start] ⇨ [All Programs].

Creating a Project

[Project] ⇨ [New]

Create a program, following the procedure below. This procedure is for programs described in ladder diagram.



Select the model of the CPU module used. Then, specify a programming language used for the project. Select "Ladder", and click the [OK] button.

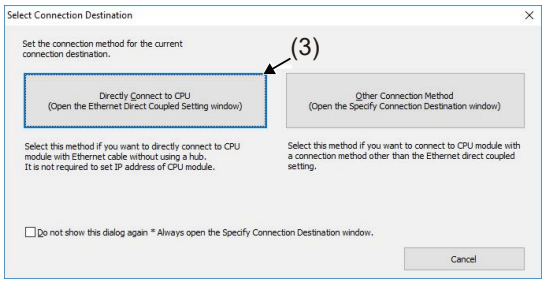
Click [OK] button when the next window appears.

Connecting a Personal Computer

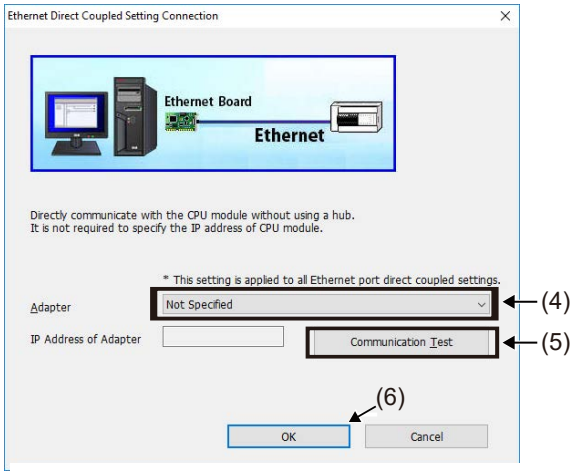
When Ethernet cable is used (direct connection)

Connect a personal computer directly to the CPU module, following the procedure below.

1. Connect a personal computer to the CPU module using the Ethernet cable.
2. Select [Online] ⇨ [Current Connection Destination] on the menubar of the engineering tool.



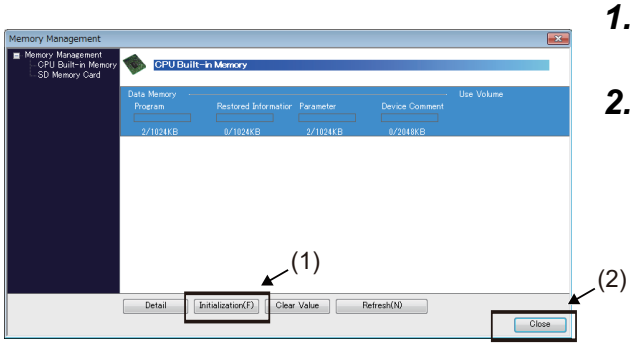
3. Click the [Directy Connect to CPU] button.



4. Select Ethernet adapter the personal computer uses.
5. Click the [Connection Test] button, and check if the personal computer is connected to the CPU module.
6. Click the [OK] button.

Initializing the CPU Module

[Online] ⇨ [CPU Memory Operation]

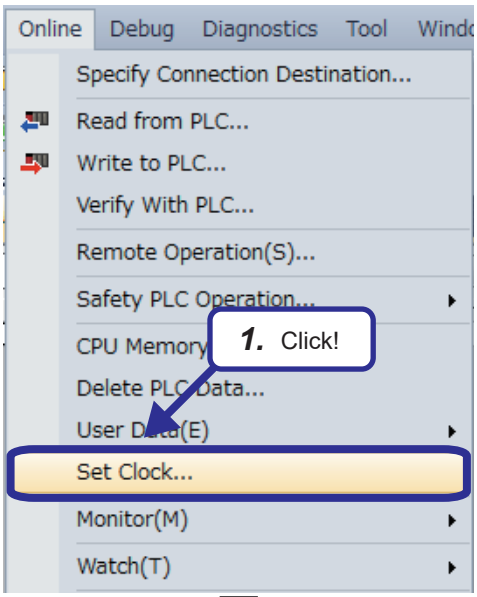


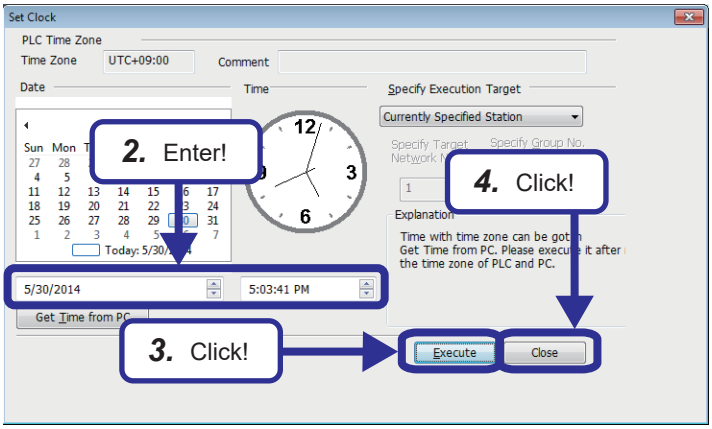
1. On the "Memory Management" window, click the [Initialization] button.
2. After the initialization processing completes, click the [Close] button.

CLOCK FUNCTION

The CPU module has an internal clock and is used to manage time in functions performed by the system such as dates of the event history function and the data logging function. Time operation continues with the large internal capacitor in the CPU module even though the power in the CPU module is turned OFF or the power failure exceeds the allowable momentary power failure time. If an optional battery is used, operation continues by the battery.

Setting the clock of the CPU module





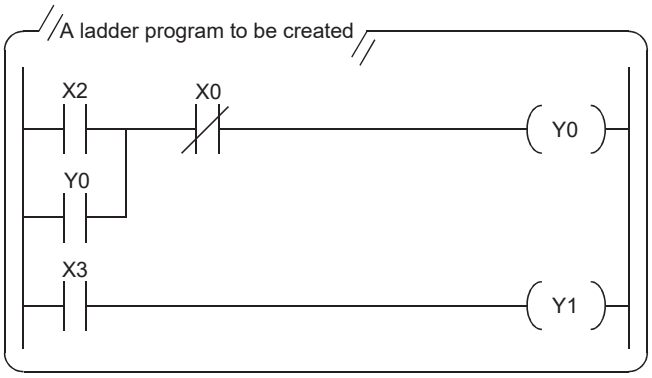
1. Click [Online] → [Set Clock] from the menu to display the "Set Clock" dialog box.
2. Set a year, month, day, hour, minute, second, and day of week on the "Set Clock" dialog box.
3. Click the [Execute] button.
4. Click the [Close] button.

The clock data handled in the CPU unit is described below.

Data name	Special register	Name	Description
Year	SD210	Clock Data (Year)	4 digits in calendar year (1980 to 2079)
Month	SD211	Clock Data (Month)	1 to 12
Day	SD212	Clock Data (Day)	1 to 31 (Leap year auto detect)
Hour	SD213	Clock Data (Hour)	0 to 23 (24-hour system)
Minute	SD214	Clock Data (Minute)	0 to 59
Second	SD215	Clock Data (Second)	0 to 59
Day-of-the-week	SD216	Clock Data (Day Week)	0: Sunday, 1: Monday, 2: Tuesday, 3: Wednesday, 4: Thursday, 5: Friday, 6: Saturday

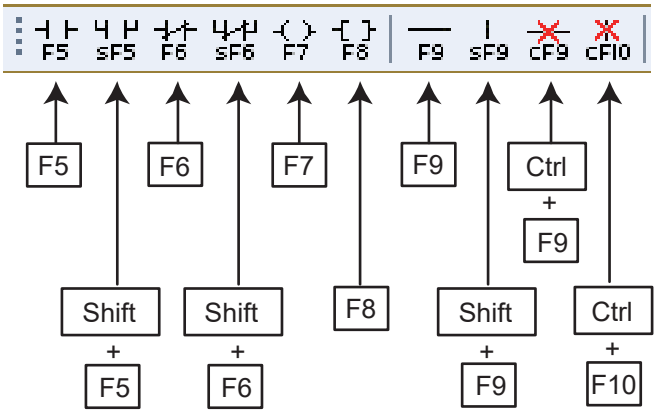
Creating a Ladder Program

This section describes how to create a ladder program such as the one shown below.

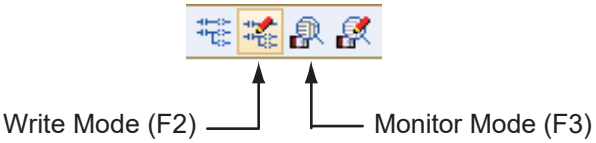


Point

The following figure shows the buttons on the toolbar. The character below each ladder symbol indicates each function key.



- Use only one-byte characters.
- Check that "Write Mode" is active.



Point

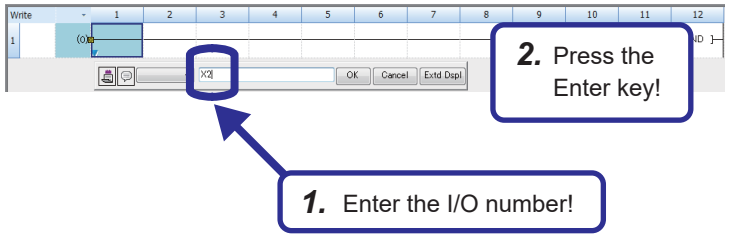
How to input contacts and coils

Users can create ladders with the function keys and tool buttons. To input a contact or coil, specify a position where a contact or coil is to be input with the cursor and enter a device and label.

Users can switch a normally open contact and normally closed contact with the "/" key.

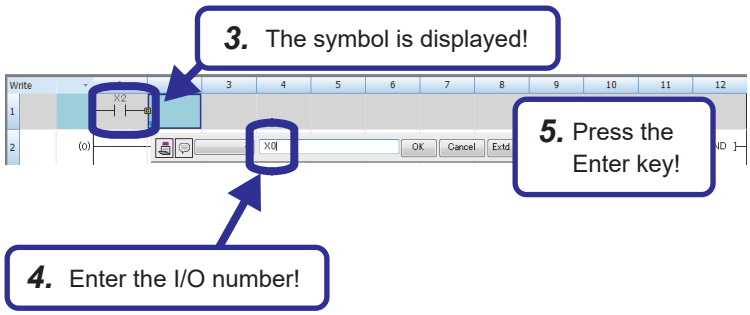
If an added ladder is in contact with the right rail or is an output device (Y, DY), the ladder is recognized as a coil. If not, it is recognized as a contact.

Creating a ladder program by entering devices and labels



1. Move the cursor to the position where a ladder is added, and enter "X2". (When entering of the number starts, the ladder input window appears.)
To cancel an incorrect entry, press the [Esc] key.

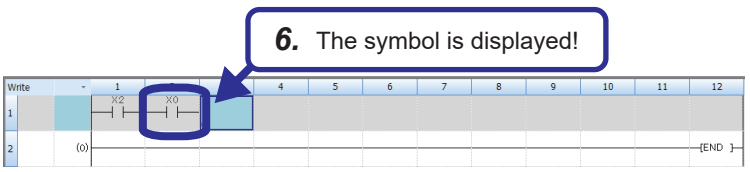
2. To confirm the entry, press the [Enter] key.
- Clicking the [OK] button also confirms the entry.
- Clicking the [Cancel] button also cancels the entry.



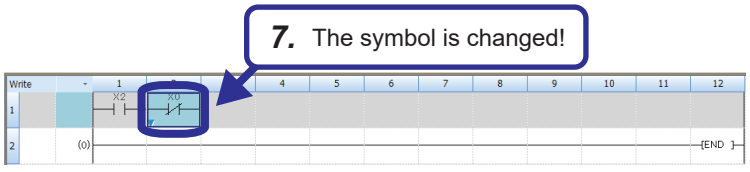
3. The added symbol (X2) is displayed.

4. Move the cursor to the next position and enter "X0".

5. Press the [Enter] key.



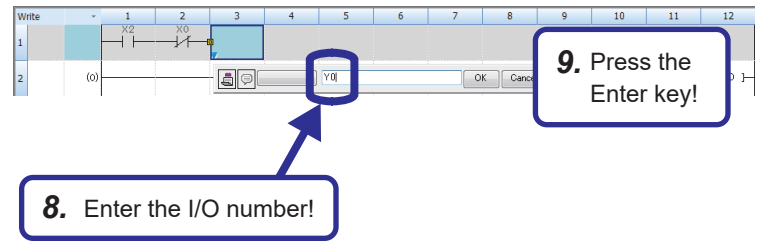
6. The added symbol (X0) is displayed.



7. Select the symbol and press the "/" key to switch the symbol with (X0).

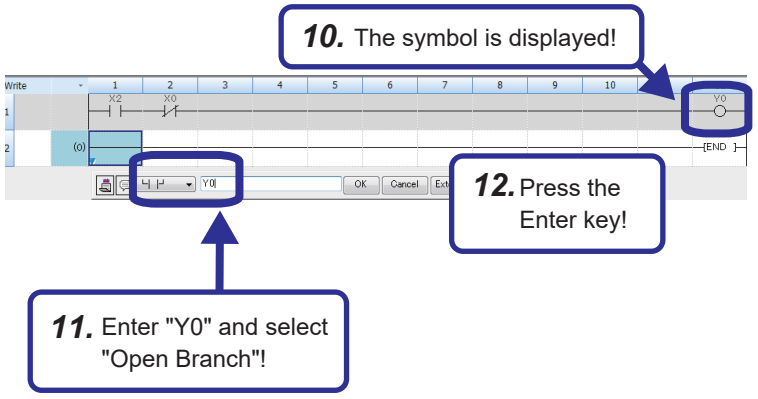
(To the next page)

(From the previous page)



8. Move the cursor to the next position and enter "Y0".

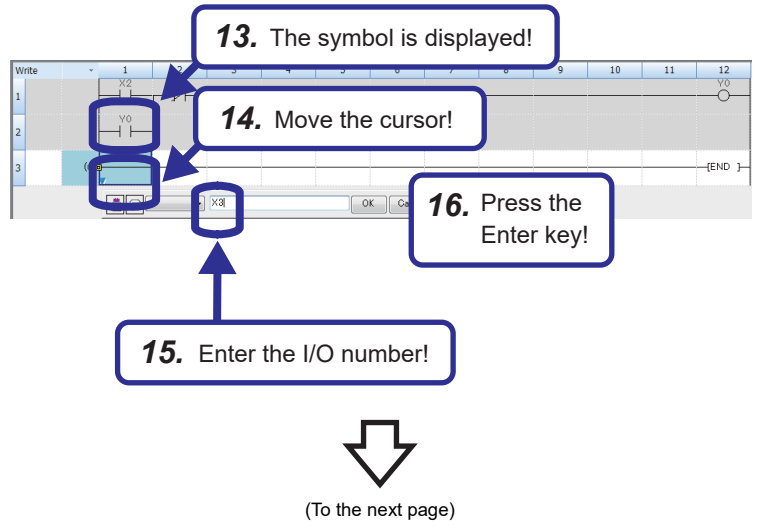
9. Press the Enter key.



10. The added symbol (—〔 Y0 〕—) is displayed.

11. Move the cursor to the next position, enter "Y0", and select "Open Branch".

12. Press the Enter key.



13. The added symbol (Y0) is displayed.

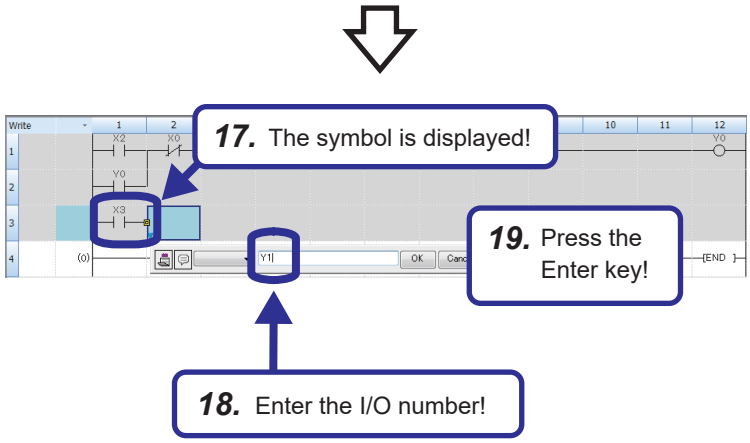
14. Move the cursor to the ladder under Y0.

15. Enter "X3".

16. Press the Enter key.

(To the next page)

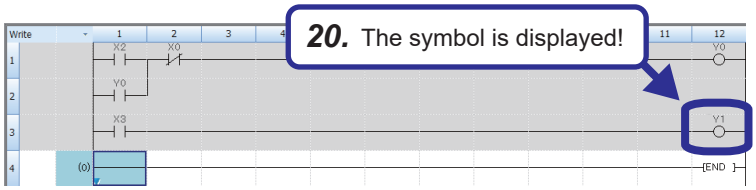
(From the previous page)



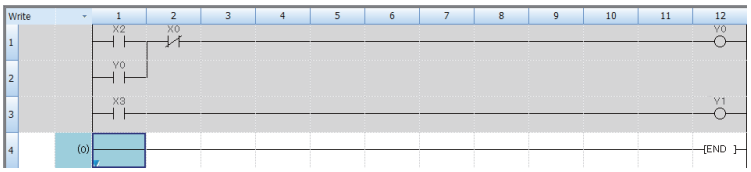
17. The added symbol (⊣X3⊢) is displayed.

18. Move the cursor to the next position and enter "Y1".

19. Press the Enter key.



20. The added symbol (⊣(Y1)⊢) is displayed.



21. Creating a ladder program is completed.

Converting the Program

Compile the input ladder blocks.

1. Select [Convert] ⇨ [Convert] on the menu bar.

2. When the conversion processing completes and the input ladder blocks are compiled, the color of those ladder blocks changes from gray to white.

Saving the Project

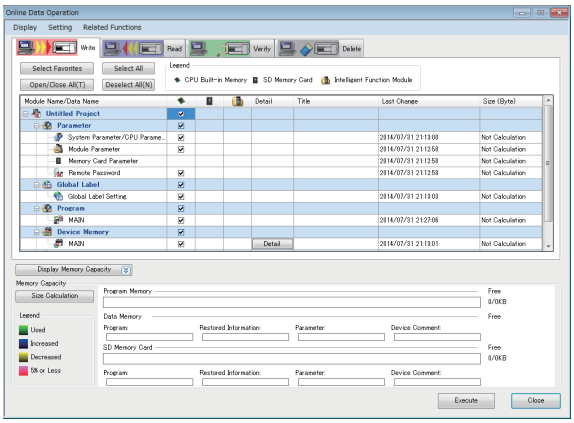
Save the created project.

[Project] ⇨ [Save as]

Writing Data to the CPU Module

Write the set parameters and the created program to the CPU module.

[Online] ⇨ [Write to PLC]



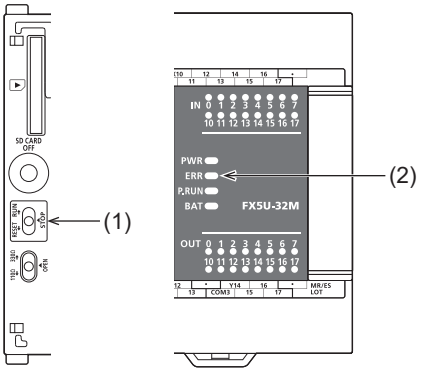
1. Select the system parameter file, CPU parameter file, module parameter file, and program file on the "Online Data Operation" window.
2. Click the [Execute] button.
3. After the write processing completes, click the [Close] button.

Point

- For the operation of the CPU module, the system parameter file, CPU parameter file, and program file must be written. For the operation of the expansion adapter, I/O modules and intelligent function modules, the module parameter files must be written as well.

Resetting the CPU Module

Reset the CPU module using the RUN/STOP/RESET switch located on the front of the CPU module.



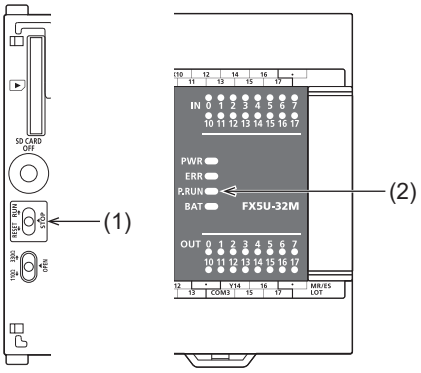
1. Set the RUN/STOP/RESET switch (1) to the RESET position for a second or longer.
2. Check that the ERROR LED (2) flashes for several times and turns off.
3. Set the switch back to the STOP position.

Point

Operate the RUN/STOP/RESET switch with your fingers. Use of a tool such as a screwdriver may damage the switch.

Executing the Program

Execute the program written to the programmable controller by using the RUN/STOP/RESET switch.



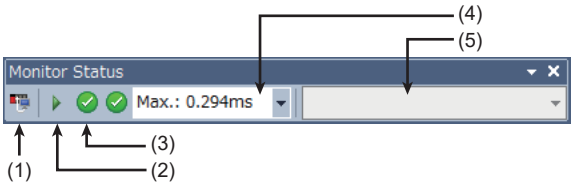
1. Set the RUN/STOP/RESET switch (1) to the RUN position.
2. Check that the P.RUN (2) turns on.

Monitoring the Program

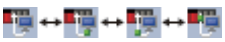
Select [Online] ⇨ [Monitor] ⇨ [Monitor Mode] (F3) on the menu bar.

Monitoring on the monitor status bar

The LED status of the CPU module and the scan time can be monitored on the monitor status bar.

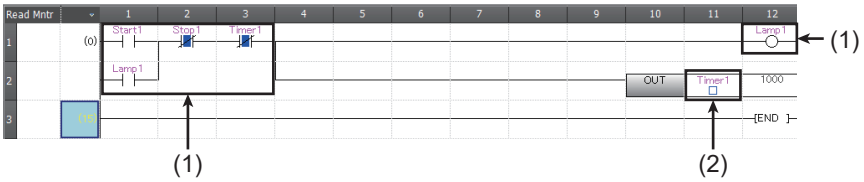


■Displayed item

No.	Item	Description	Icon	Meaning
(1)	Connection status	The connection status with the CPU module is displayed.		Connected with the CPU module
				Not connected with the CPU module
(2)	Operating status	The operating status of the CPU module in accordance with the RUN/STOP/RESET switch of the CPU module or the remote operation by the engineering tool is displayed. Click the icon. Then, "Remote operation" window appears. (GX Works3 Operating Manual)		RUN
				STOP
				PAUSE
(3)	ERROR LED status	The ERROR LED status of the CPU module is displayed. Click the icon. Then, "Module diagnostics" window appears.		Off
				On
				Flashing
(4)	Scan time details	The scan time details are displayed. Select the value to be displayed from the drop-down list (current value, maximum value, or minimum value).		
(5)	Monitor target selection	Specify the monitor target FB instance when monitoring a FB program.		

Monitoring on the ladder editor

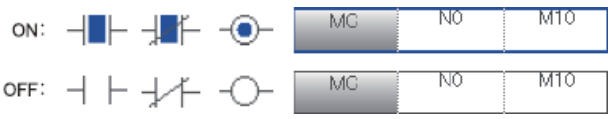
The on/off states of contacts and coils and the current values of word devices and labels can be monitored on the ladder editor.



(1) The on/off states of the contacts and coils are displayed.
(2) The current value of the word/double word type data is displayed.

■On/off state display

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



■Changing the current value

To change the current value, select the cell on the ladder editor, and press Shift + double-click the cell or press Shift + Enter while the program is being monitored.

Point

The program can also be monitored on the Device/Buffer Memory Batch window of the Watch window.

Troubleshooting

When the system has any trouble, perform troubleshooting in the following order.

1. Check that each module is properly mounted and wired.
2. Check the LED status of the CPU module.
3. Check the LED status of each intelligent function module. (User's manuals for each module)
4. Connects the engineering tool and execute the Module diagnostics(CPU Diagnostics) function. The error cause and the action to be taken can be displayed.
5. If the error cause cannot be identified in steps 1 to 3, troubleshoot by symptom. (User's manuals for each module)

Checking with LEDs

Check the module status with LEDs as a means to perform the primary diagnosis.

Checking the PWR LED

If the PWR LED is turned off, check the following items.

Item to check	Cause and action
Measure the voltage input to the CPU module.	If the power source is not supplying the proper voltage, provide a power source that does.
Remove extension modules and all wiring other than the one for the power supply, and turn on the power supply.	If the PWR LED turns on in this condition, the service power supply may be shorted or overloaded. Reconnect the modules and wire one by one to identify the cause.

If the PWR LED still does not turn on, there may be a hardware issue. Consult your local Mitsubishi Electric representative.

Checking the BAT LED

If the BAT LED is turned on, check the following items.

The LED is valid when the optional battery is installed and the LED is set to "display" for battery errors by using the parameter.

Item to check	Cause and action
Check the installation of the battery.	Install the battery properly.
Check the battery voltage.	If the battery voltage is reduced, replace the battery with a new one. Battery voltage can be monitored with PLC data register SD8005. (MELSEC iQ-F FX5U User's Manual (Hardware))

Checking the ERR LED

If the ERR LED flashes, check the following items.

Item to check	Cause and action
Write system parameters, CPU parameters, and program files, and turn on the power supply again.	Programs or parameters may not written or damaged. Since a CPU module with the factory default setting has no programs, the ERR LED flashes.

If the ERR LED is turned on, check the following items.

Item to check	Cause and action
Set the CPU module to the STOP status and turn on the power supply again.	When the ERR LED turns off, a watchdog timer error has probably occurred. Take any of the following measures. • Review the program and avoid the maximum value (SD524, SD525) of the scan time exceeding the scan time monitoring time setting value set by the parameter. • Check that the input used for input interruption or pulse catch is not being abnormally turned on and off in one scan. • Check that the frequency of the pulse input to the high-speed counter is within the specified range. • Add some WDT instructions to the program and reset the watchdog timer several times in one scan.
Provide a different power supply to the CPU module.	If the ERR LED turns off, noise may have affected the module. Consider taking the following measures. • Check the ground wiring, and reexamine the wiring route and installation location. • Fit a noise filter onto the power supply line.

If the ERR LED still does not turn off, there may be a hardware issue. Consult your local Mitsubishi Electric representative.

Checking the P.RUN LED

If the P.RUN LED is turned off, check the status of the ERR LED and take corrective actions.

Troubleshooting using the engineering tool

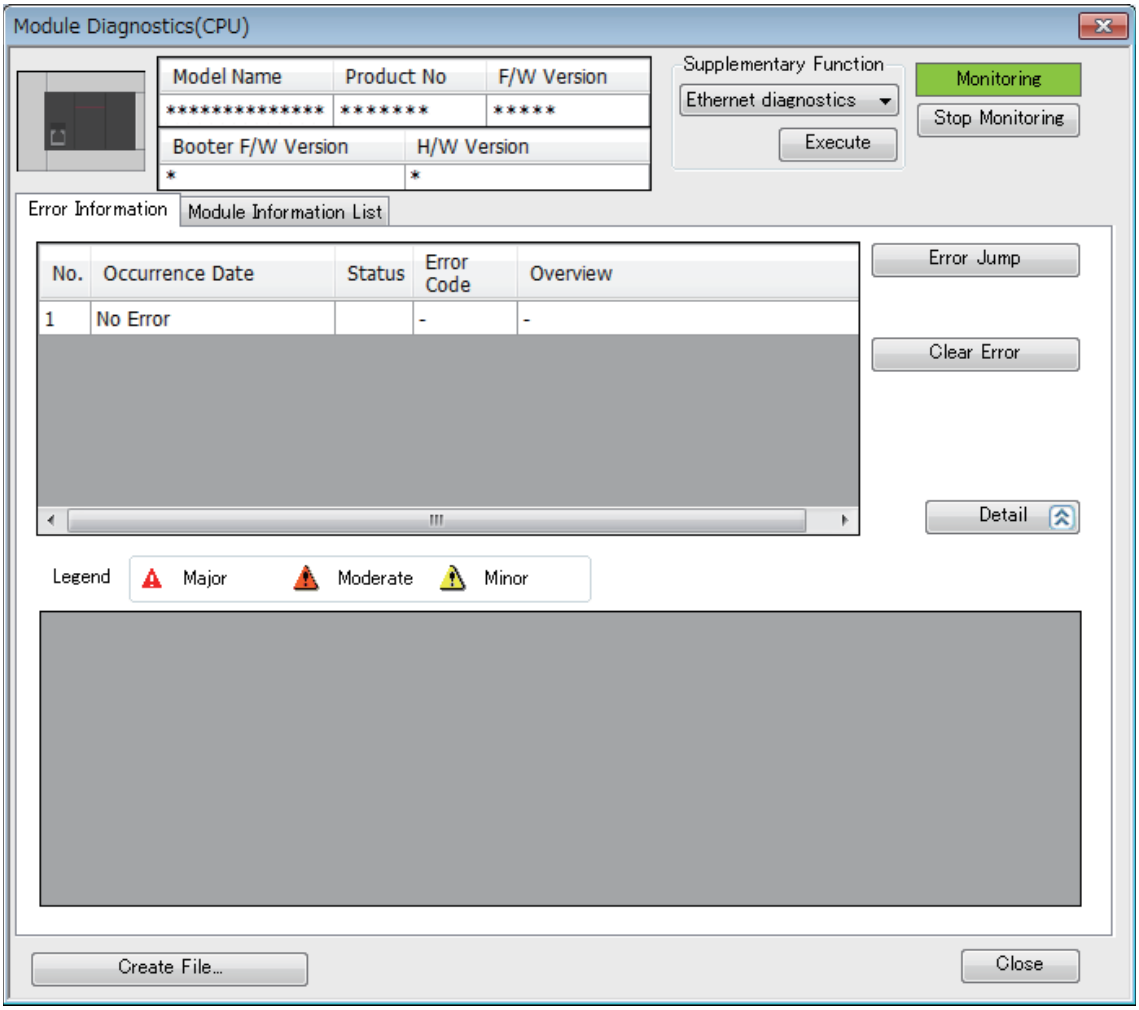
Check the error or history using the engineering tool, and identify the error cause. More detailed information on the error as well as the error cause and action to be taken can be checked by using the engineering tool.
The engineering tool has the following functions for troubleshooting.

Function	Description
Module diagnostics (CPU Diagnostics)	This function diagnoses the module. (The current error and its details can be checked.)

Module diagnostics (CPU Diagnostics)

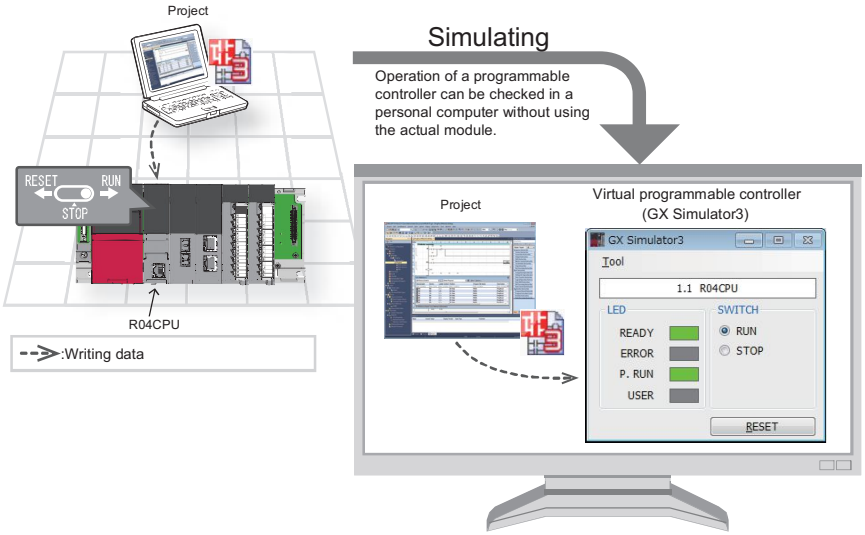
This function diagnoses CPU module, expansion adapter, and expansion board. (The current error and its details can be checked.)
Information required for troubleshooting, such as the current error, details and cause of the error, and action to be taken, are displayed. The incorrect parameter setting and program error location can also be identified by selecting the error and clicking the [Error Jump] button. On the [Module Information List] tab, the LED status and the switch status of the target module can be checked.

[Diagnostics]⇨[Module diagnostics(CPU Diagnostics)]



Simulation Function

The Simulation function debugs programs using a virtual programmable controller on a personal computer.
GX Simulator3 is used for the Simulation function. This function is useful to check programs before operating them actually since it enables debugging without the connection with CPU module.



Starting a simulation

[Debug] ⇨ [Simulation] ⇨ [Start Simulation] ( )

Select the checkbox of the data to be written on the "Online Data Operation" screen, and click the [Execute] button.

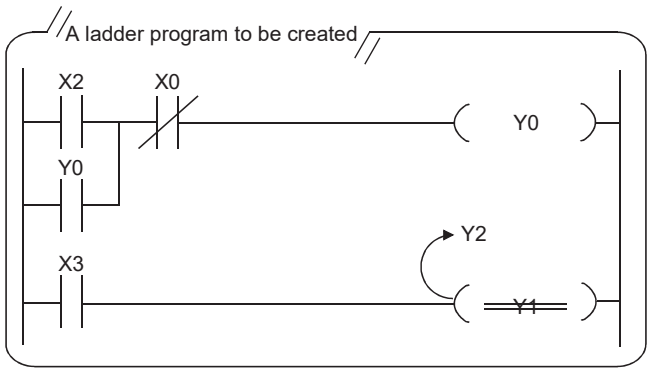
GX Simulator3 is set as the connection destination during the simulation,
and "Simulation ((system number of the simulator).(CPU number))" is displayed on the status bar.

Ending a simulation

Select [Debug] ⇨ [Simulation] ⇨ [Stop Simulation] ( ).

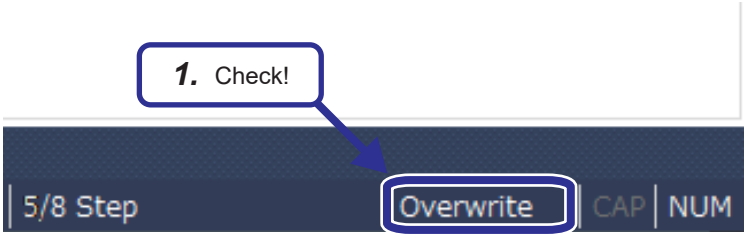
Editing a Ladder Program

Modifying a part of a ladder program



This section describes how to modify a part of the ladder program shown on the left.
(OUT Y1 → OUT Y2)

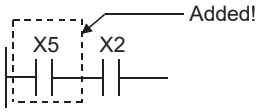
- Use only one-byte characters.
  Two-byte characters cannot be used.

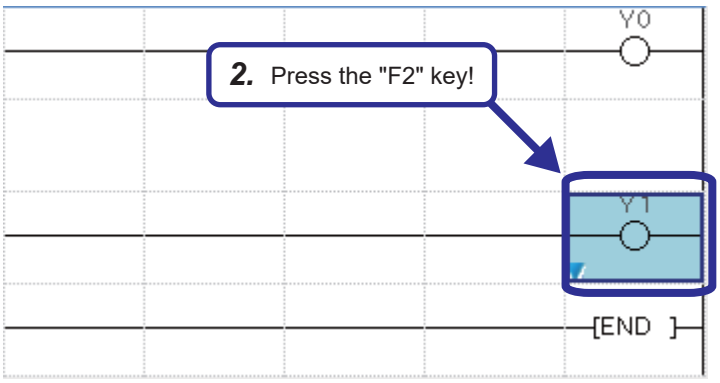


1. Check that "Overwrite" is displayed at the bottom right of the window.

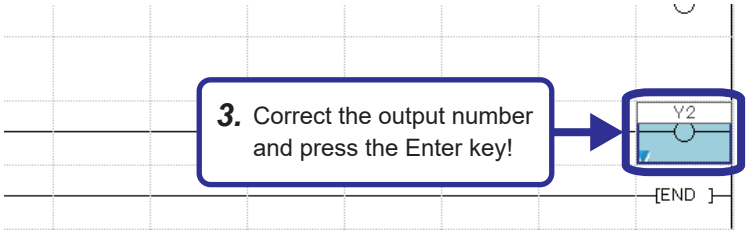
When "Insert" is displayed, press the Insert key to switch to "Overwrite".
When "Insert" is displayed, contacts or coils are added.

<When changing X2 with X5 is attempted>



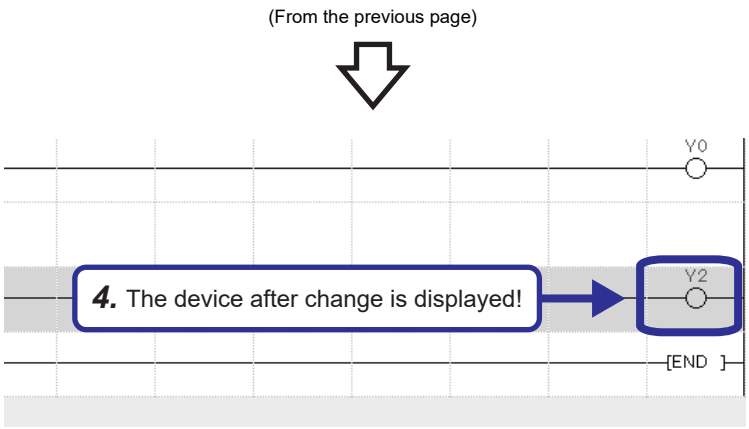


2. Click the position to be modified, and press the "F2" key.



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

(To the next page)



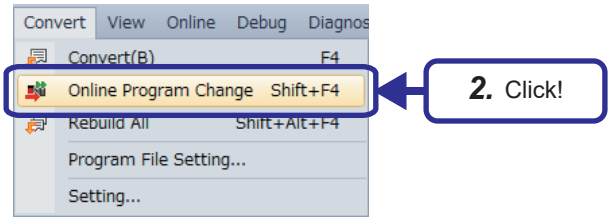
4. The ladder program after the modification is displayed.
To change only the device number, click the F2 key.

Online Change

Point

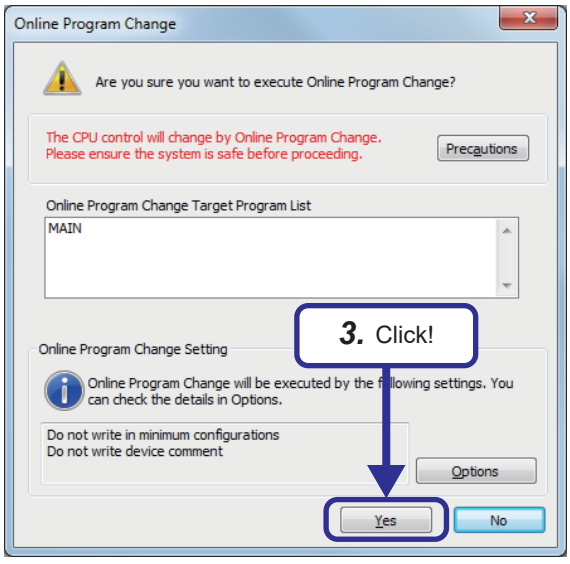
This section describes how to change a program while the CPU module is in the RUN state.

This function allows users to write a program even while the CPU module is in the RUN state.



1. Change the ladder.

2. After the change, click [Convert] → [Online Program Change] from the menu.
Or, press Shift + F4.



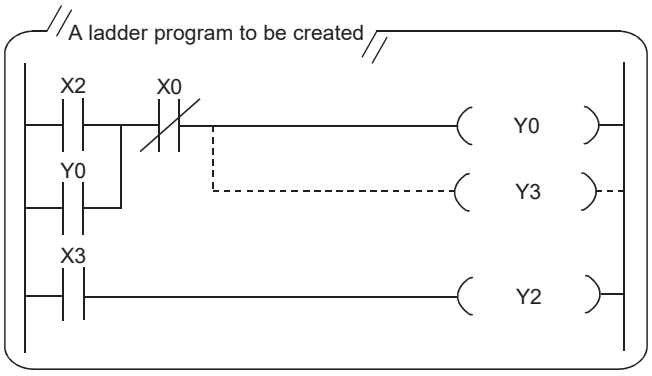
3. The message "CAUTION" appears. Click the [Yes] button to accept the change.

4. Online change is completed.

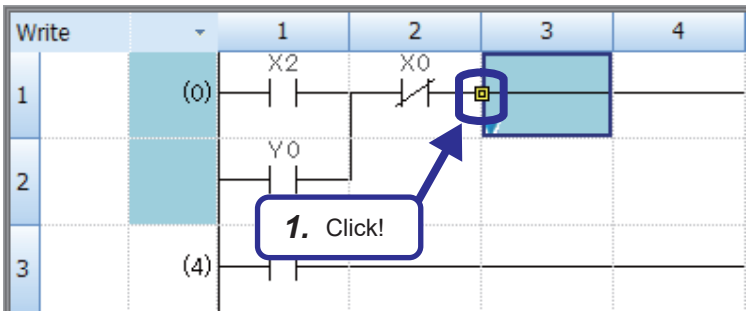
Precautions

Online change cannot be executed when the program in the CPU module and the program before the modification in GX Works3 do not match. Thus, when whether the programs match or not is unclear, verify them before the modification with GX Works3, and execute the online change.

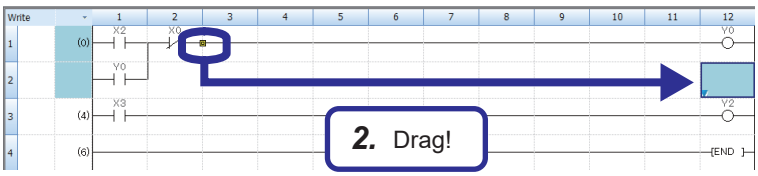
Drawing a line



This section describes how to draw a line in the ladder program shown on the left.

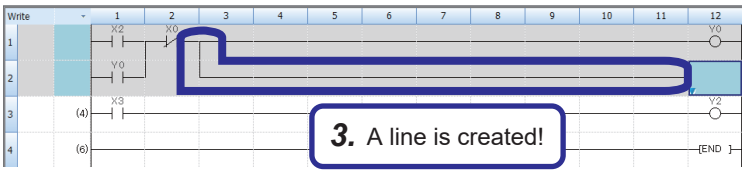


1. Move the mouse pointer close to the exiting line, and click the displayed icon.

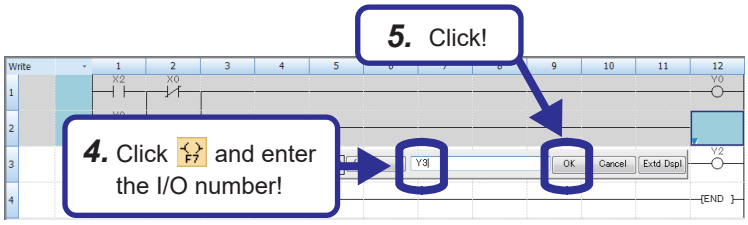


2. Drag the icon from the position to the end position.

A vertical line is drawn on the left side of the cursor.



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

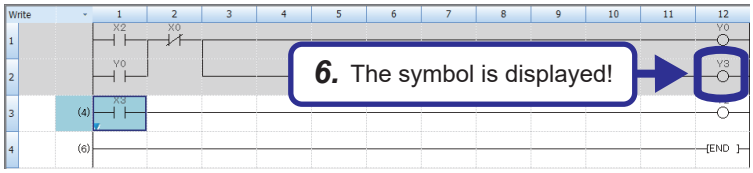


4. Click F7 on the toolbar and enter "Y3".

5. Click the [OK] button.

(To the next page)

(From the previous page)



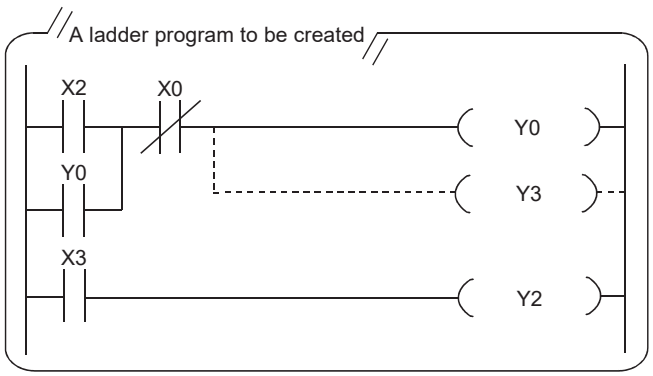
6. The added symbol (—〔 Y3 〕—) is displayed.

7. Click [Convert] → [Convert] ([F4]) from the menu to convert the ladder program.

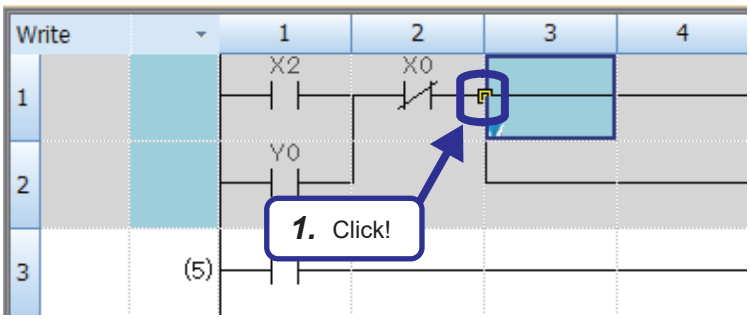
Point

Lines can be added or deleted with the [Ctrl] key + [↑], [→], [←], or [↓]. Users can draw a horizontal line from the cursor position to the position of the next contact, coil, or line by pressing [Ctrl] + [Shift] + [←] or [→].

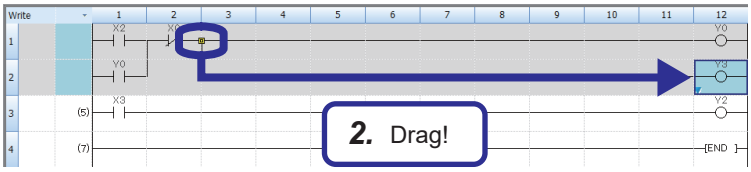
Deleting a line



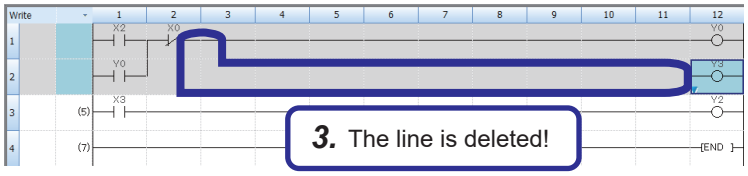
This section describes how to delete a line in the ladder program shown on the left.



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

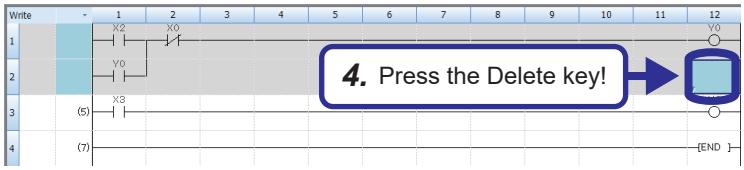


2. Drag the icon along the line to be deleted.



3. Release the left button of the mouse to delete the line.

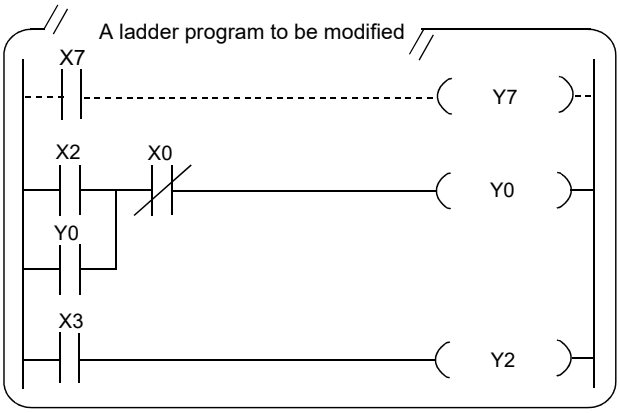
The line connected to "END" cannot be deleted.



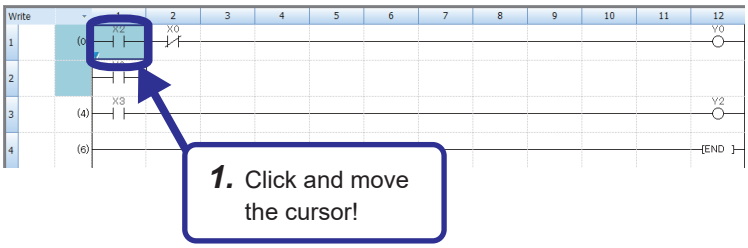
4. Press the [Delete] key to delete (—〔 Y3 〕—).

5. Click [Convert] → [Convert] ([F4]) from the menu to convert the ladder program.

Inserting a row

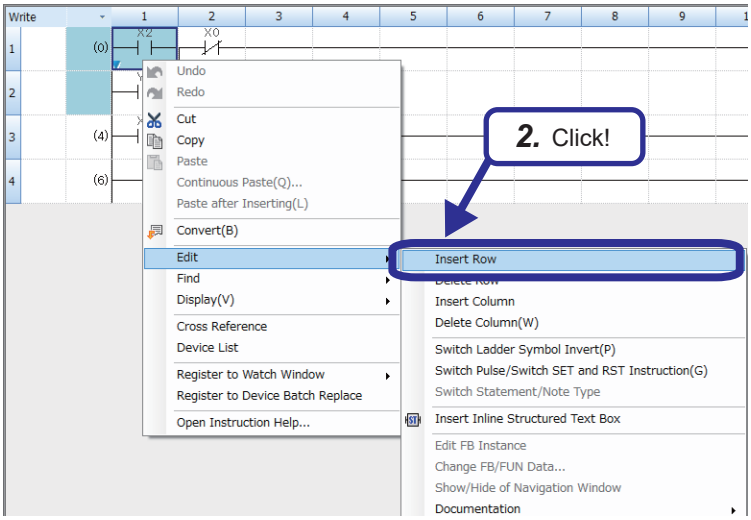


This section describes how to insert a row in the ladder program shown on the left.



1. Click and move the cursor on the row (desired position on the row) where a new row is inserted above.

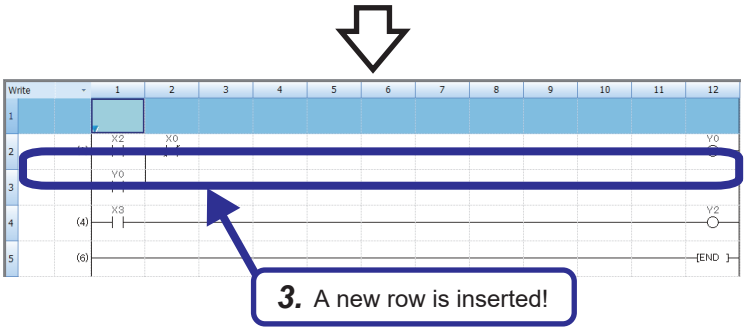
A new row is inserted above the row.



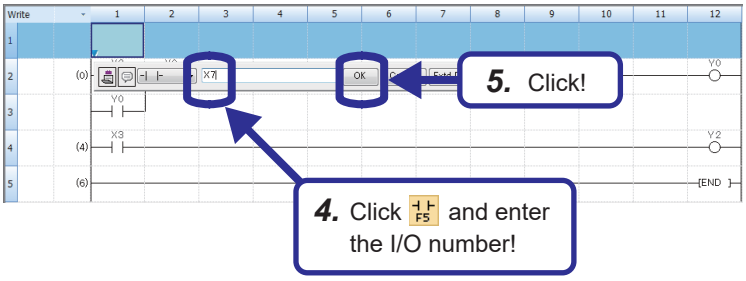
2. Right-click on the ladder editor, and click [Edit] → [Insert Row] (Shift + Insert) from the menu.

(To the next page)

(From the previous page)

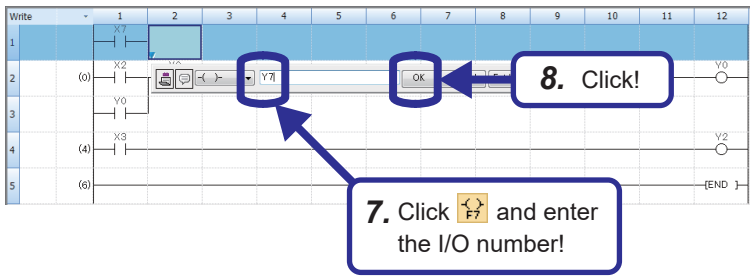


3. A new row is inserted above the row.



4. Click F5 on the toolbar to open the ladder input window, and enter "X7".

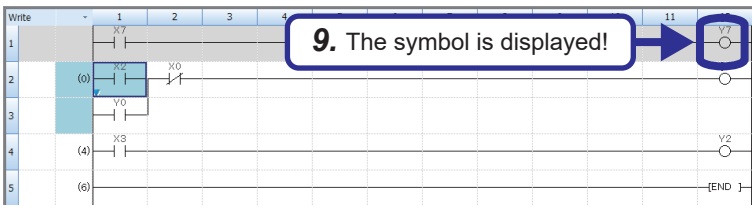
5. Click the [OK] button to confirm the entry.



6. The added symbol (X7) is displayed.

7. Click F7 on the toolbar and enter "Y7".

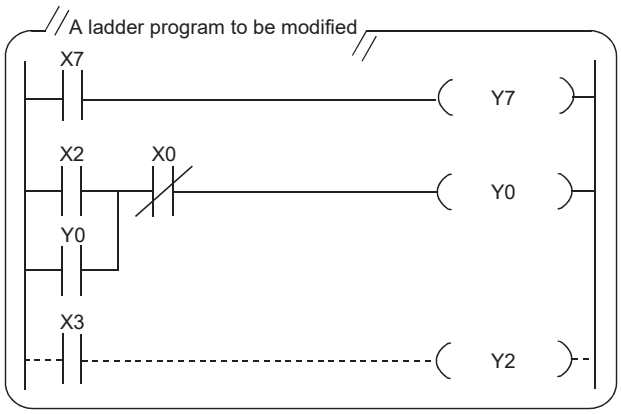
8. Click the [OK] button.



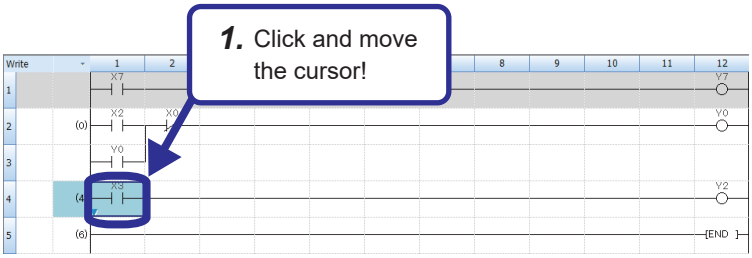
9. The added symbol (Y7) is displayed.

10. Click [Convert] → [Convert] (F4) from the menu to convert the ladder program.

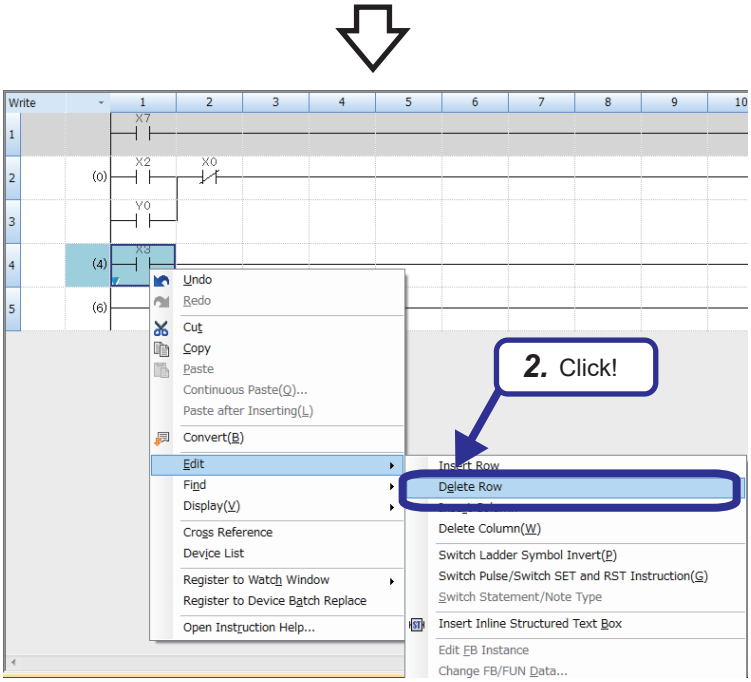
Deleting a row



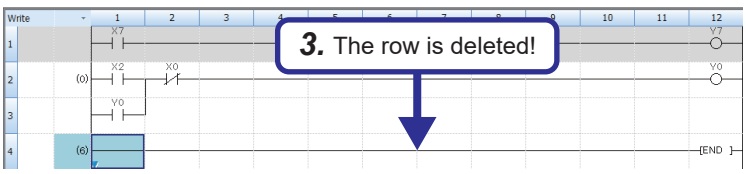
This section describes how to delete a row in the ladder program shown on the left.



1. Click and move the cursor on the row (desired position on the row) to be deleted.



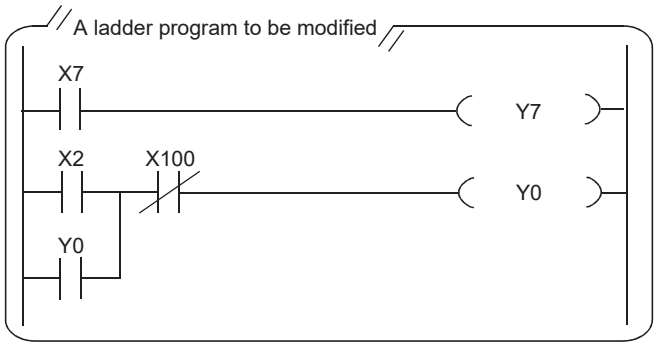
2. Right-click on the ladder editor, and click [Edit] → [Delete Row] (Shift + Delete) from the right-click menu.



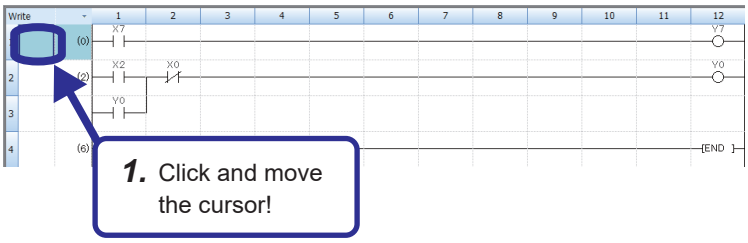
3. The row is deleted.

4. Click [Convert] → [Convert] (F4) from the menu to convert the ladder program.

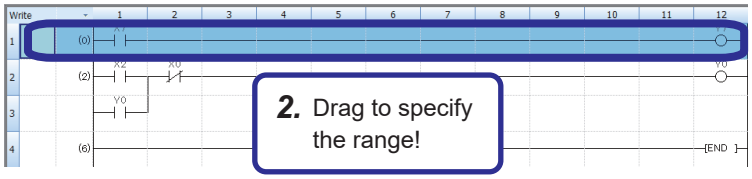
Cutting or copying a ladder



This section describes how to cut or copy a part of the ladder program shown on the left and paste the cut part or the copy of the part to any desired location in the ladder.

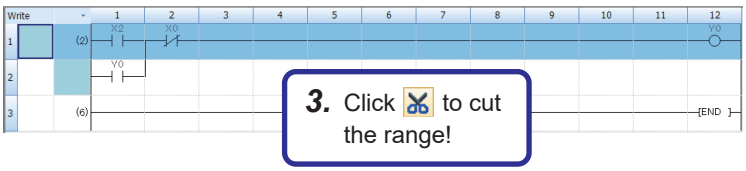


1. Click and move the cursor to the position where a part of the ladder program is to be cut.



2. Drag the mouse to specify the cutting range. The color of the specified range is highlighted.

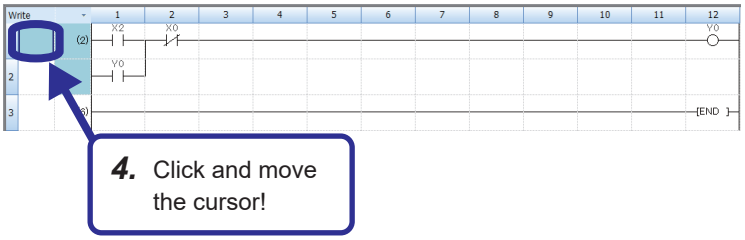
To easily specify the range in units of ladder blocks, click the position where a step number is displayed and drag the mouse vertically.



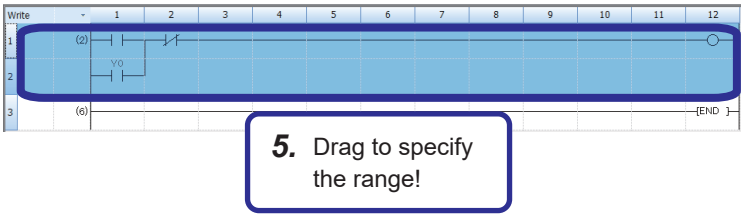
3. Click on the toolbar, or select [Edit] → [Cut] (Ctrl + X) to cut the ladder in the specified range.

(To the next page)

(From the previous page)

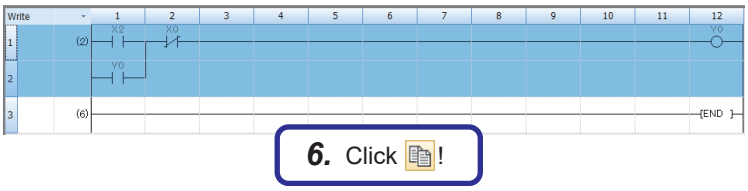


4. Click and the move the cursor to the position where a part of the ladder program is to be copied.

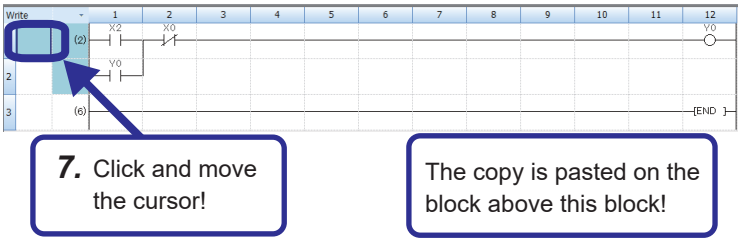


5. Drag the mouse to specify the copy range.
The color of the specified range is highlighted.

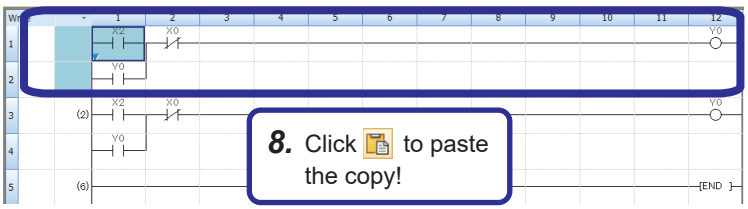
To easily specify the range in units of ladder blocks, click the position where a step number is displayed and drag the mouse vertically.



6. Click on the toolbar, or select [Edit] → [Copy] (Ctrl + C).



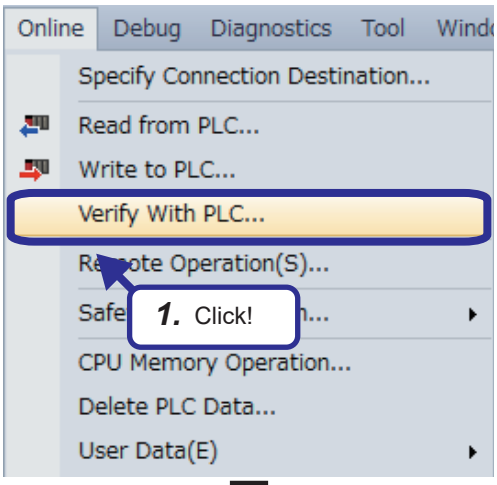
7. Click and move the cursor to a position on the ladder block (any position on the block) where the copy is to be pasted.



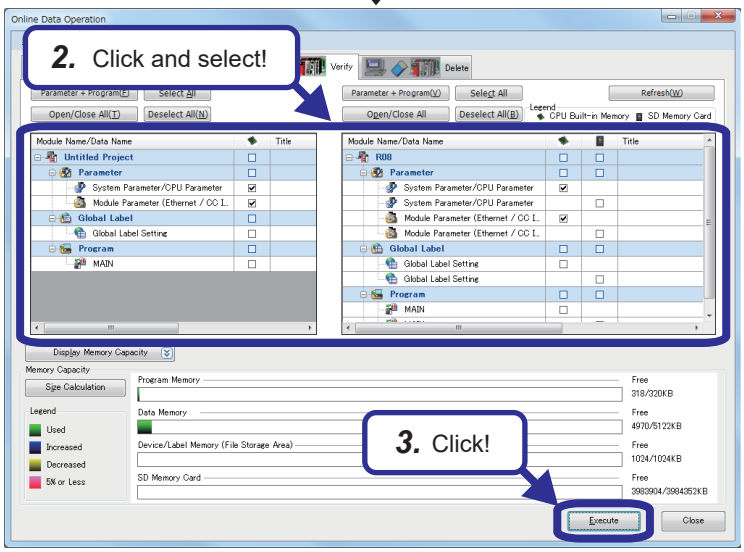
8. Click on the toolbar, or select [Edit] → [Paste] (Ctrl + V) from the menu.

Verifying Data

This section describes how to verify the currently-opened project and the data stored in the CPU module.
Perform this operation to check whether the projects are identical or to check changes in a program.

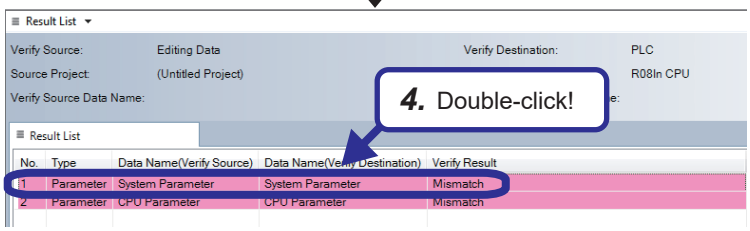


1. Click [Online] → [Verify with PLC] from the menu.



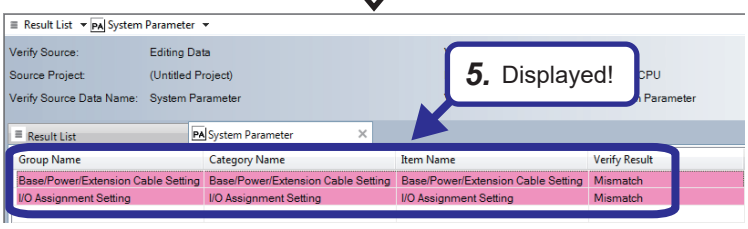
2. The "Online Data Operation" dialog box appears. Select data to be verified.

3. Click the [Execute] button.



4. Verification results are displayed in the "Result List" window.

To check details of data, double-click the row of the data.



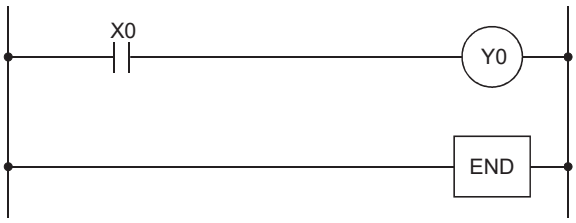
5. Detailed results are displayed.

Differences Between [OUT] and [SET]/[RST]

Point

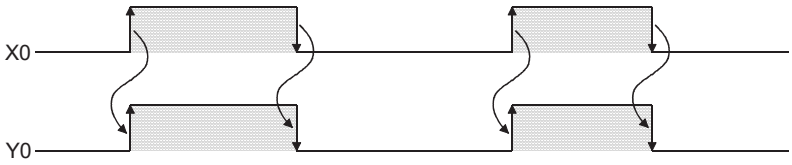
This section describes the OUT and SET/RST instructions and the operation of a self-holding ladder.

[OUT] (Coil output)

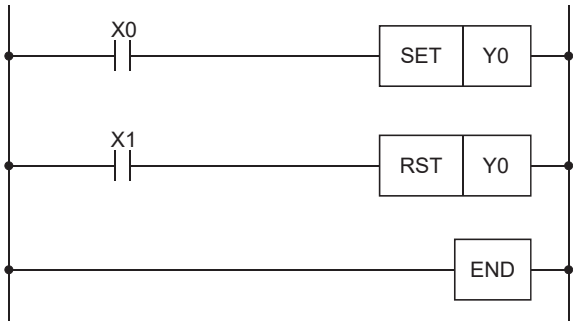


The OUT instruction turns on a specified device when the input condition turns on, and turns off the device when the condition turns off.

■Timing chart

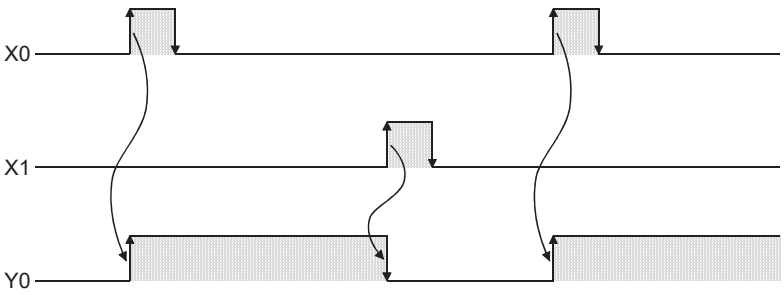


[SET]/[RST](Setting/resetting devices)



The SET instruction turns on a specified device when the input condition turns on, and holds the on state of the device even though the condition turns off.
To turn off the device, use the RST instruction.

■Timing chart

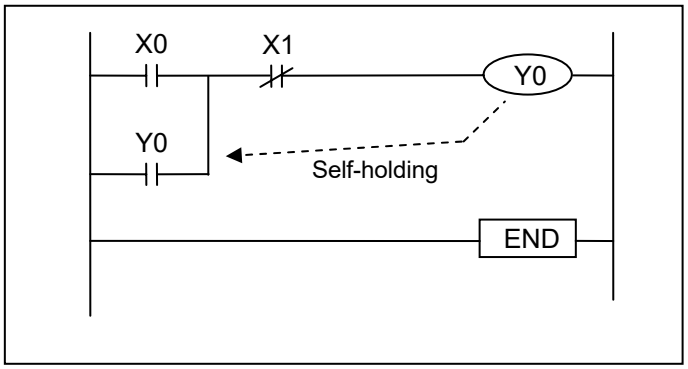


Self-holding Ladder

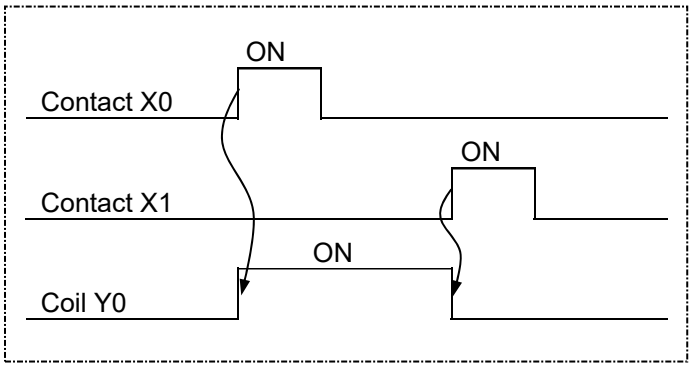
A self-holding ladder holds the operation status of the coil by its contact.

By configuring the sequence, the output status can be held or canceled.

[Example of ladder diagram]



[Timing chart]



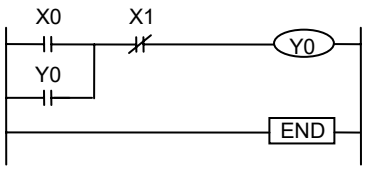
[Sequence]

When the input X0 is on and X1 is off, the output Y0 turns on.

The output Y0 remains on even when the input X0 is turned off (self-holding). Turning on the input X1 turns off the output Y0.

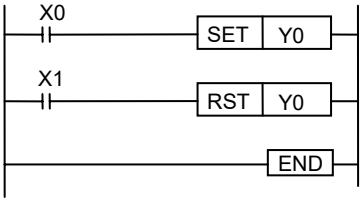
Both of the following ladders hold the output.

[Self-holding ladder of output]



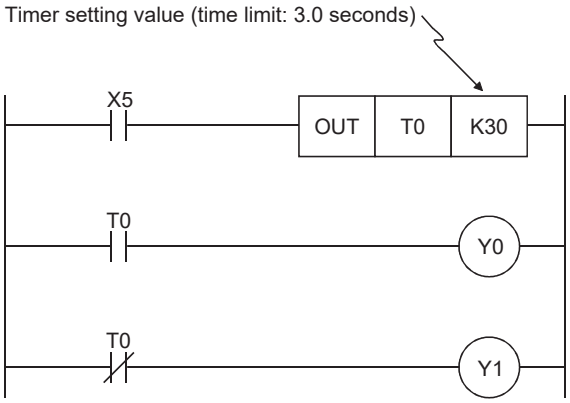
Operation will be the same.

[SET/RST ladder of output]



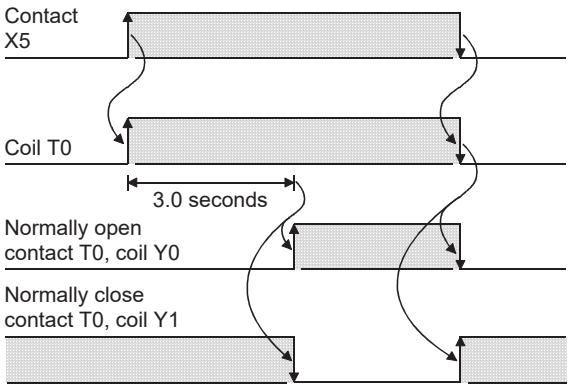
Using the SET instruction can hold the on status of the output even when the contact which drives the coil turns off.

Timer, High-speed Timer, Retentive Timer



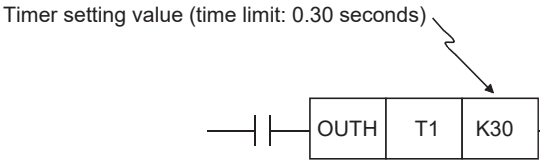
- The operation of the timer contact delays by a set time after the coil is energized. (On delay timer)
- The setting range of a timer value is K1 to K32767.
  Low-speed (100ms) timer: 0.1 to 3276.7 seconds
  High-speed (10ms) timer: 0.01 to 327.67 seconds
- When the value set to a timer is 0, it is turned on (timeout) by the execution of the instruction.

■Timing chart



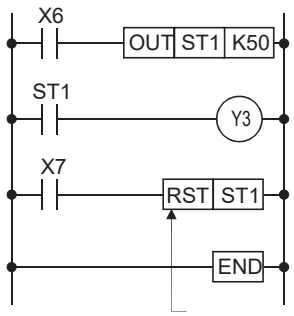
High-speed timer

Change the output instruction "OUT" to "OUTH" to select a high-speed timer.

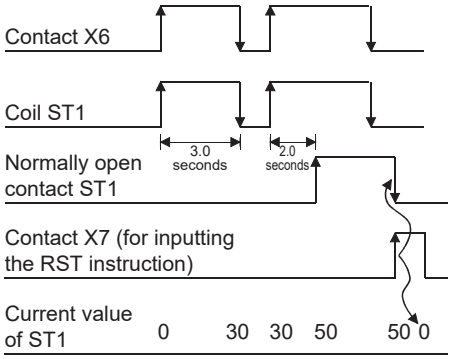


Retentive timers

When an input condition turns on, the coil turns on and the value of a retentive timer starts increasing. When the current value becomes equal to a set value, the retentive timer goes timeout and the contact turns on. When the input condition turns off during the addition, the coil turns off but the current value is held. When the input condition turns on again, the coil turns on and the current value is used in the integration to continuously increase the value of the timer.



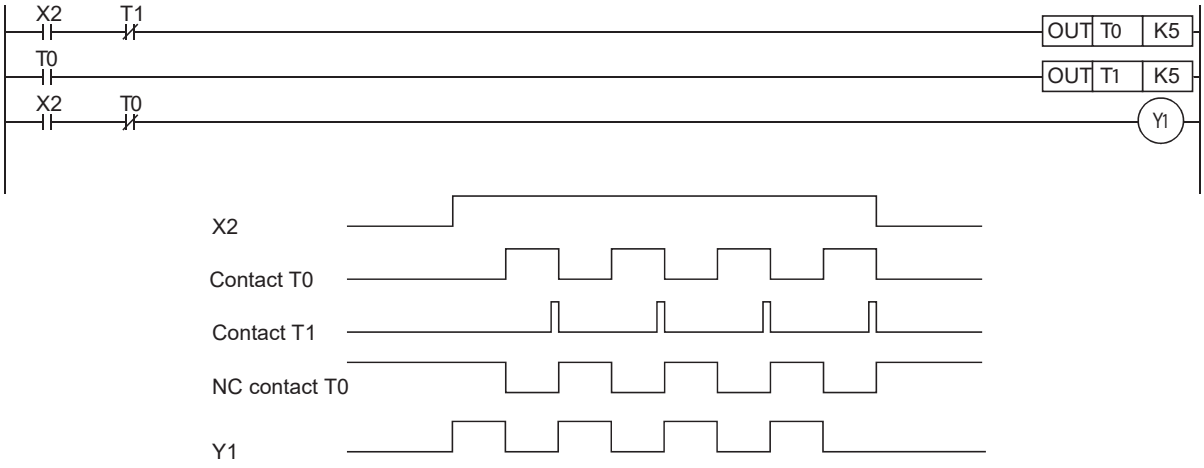
Always use the RST instruction for turning off the contact and clearing the current value after the retentive timer goes timeout.



High-speed retentive timers

Change the output instruction "OUT" to "OUTH" to select a high-speed retentive timer.

When X2 turns on, Y1 turns on, or turns off. This flip-flop operation is repeated.



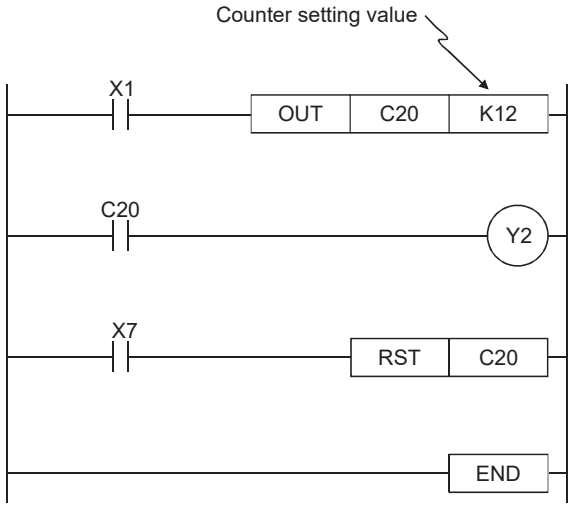
System clock

The special relay about system clock is shown below.

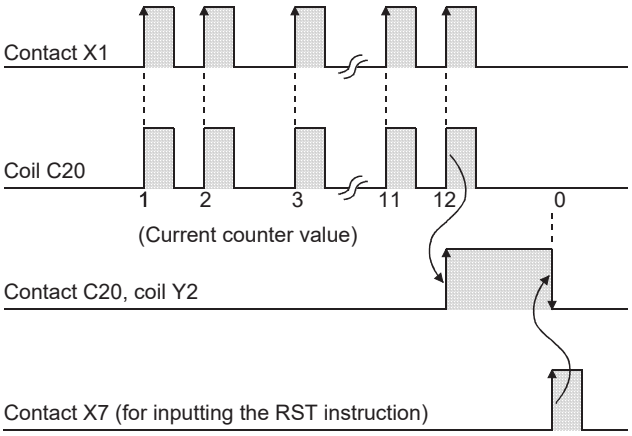
No.	Name	Description
SM400	Always ON	ON OFF
SM401	Always OFF	ON OFF
SM402	After RUN, ON for one scan only	ON OFF 1 scan
SM403	After RUN, OFF for one scan only	ON OFF 1 scan
SM409	0.01 second clock	0.005 s 0.005 s
SM410	0.1 second clock	0.05 s 0.05 s
SM411	0.2 second clock	0.1 s 0.1 s
SM412	1 second clock	0.5 s 0.5 s
SM413	2 second clock	1 s 1 s

Counter

- This section describes how to input a counter.
- This section describes the words "rise (rising edge)" and "fall (falling edge)".



■Timing chart

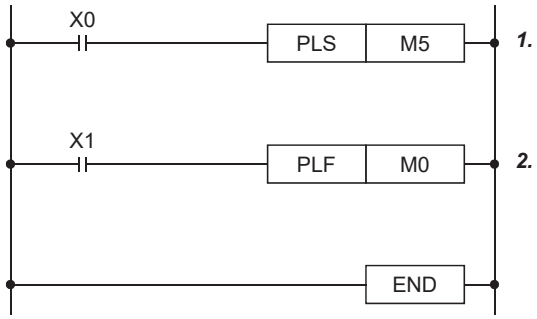


- A counter counts at the rising edge of an input signal.
- After counting is up, the counter does not count at the rising edges of the subsequent input signals.
- Once counting is up, the contact status and the current value (count value of the counter) do not change until the RST instruction is executed.
- Executing the RST instruction before counting is up clears the counter value to 0.
- The setting range of a counter value is K0 and K32767. (K0 turns on (starts counting) at execution of the instruction.)

[PLS] (Turning on a Specified Device for One Scan at the Rising Edge of an Input Condition) [PLF] (Turning on a Specified Device for One Scan at the Falling Edge of an Input Condition)

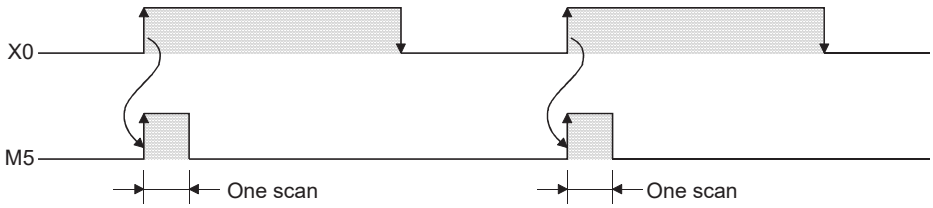
Point

- This section describes the concept of one scan.
- This section describes the operation timing of the PLS/PLF instruction.



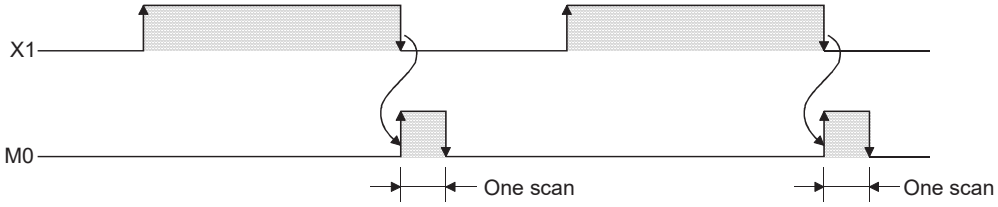
1. The PLS instruction turns on a specified device only for one scan at the rising edge of the commanded condition.

■Timing chart

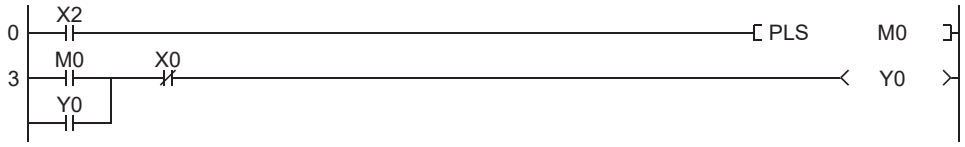


2. The PLF instruction turns on a specified device only for one scan at the falling edge of the commanded condition.

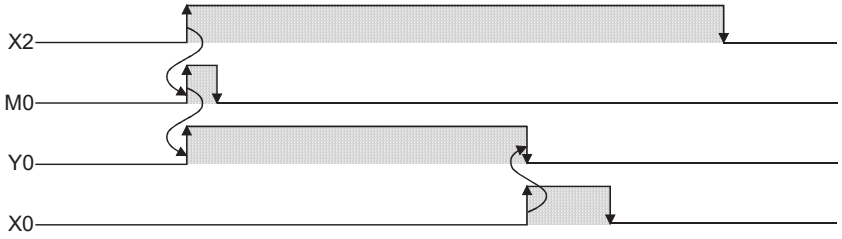
■Timing chart



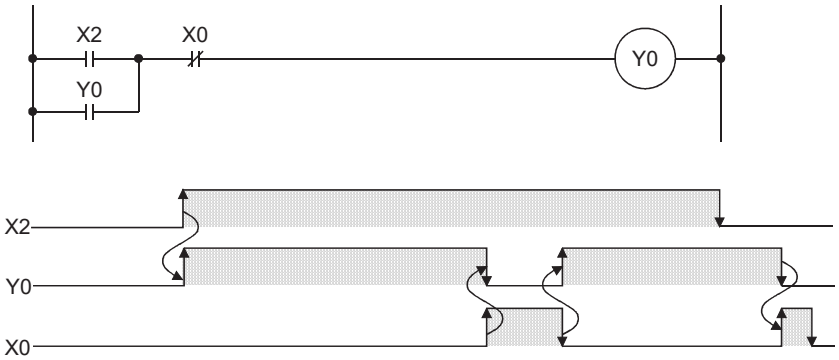
Useful application of the PLS/PLF instructions (Part 1)



■Timing chart



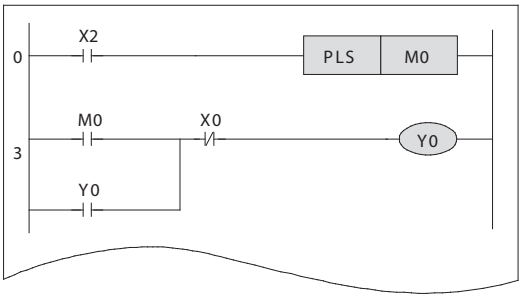
The following shows the timing chart of a self-holding ladder created with the OUT instruction.
Compare this timing chart with the one of the self-holding ladder created with the PLS instruction.



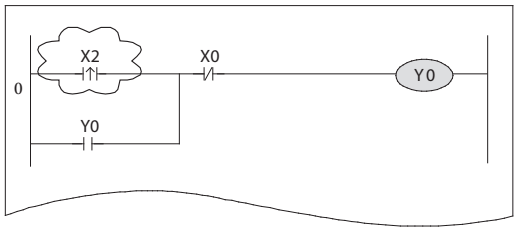
Point

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

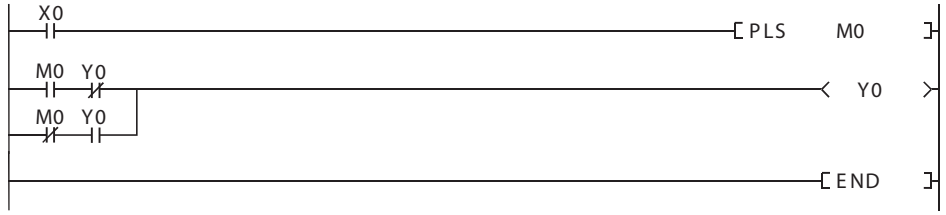


When the rising edge pulse contact instruction is used

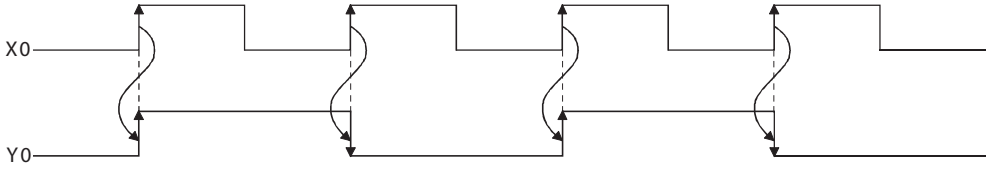


Useful application of the PLS/PLF instructions (Part 2)

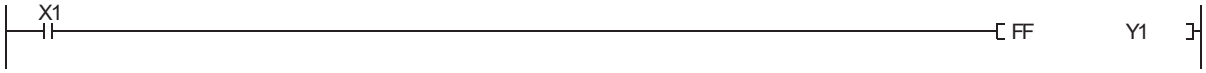
These instructions can be used in a program that executes a repetitive operation such as switching the on/off status every time a push button switch (snap switch) is pressed. (If the PLS instruction is used in the program, the program is executed at the rising edge caused when the push button switch is pressed.)



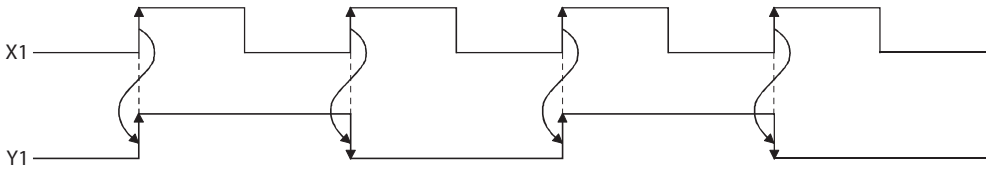
■Timing chart



In an on/off repeat ladder, Y1 turns on when X1 turns on for the first time, and turns off when X1 turns on again.



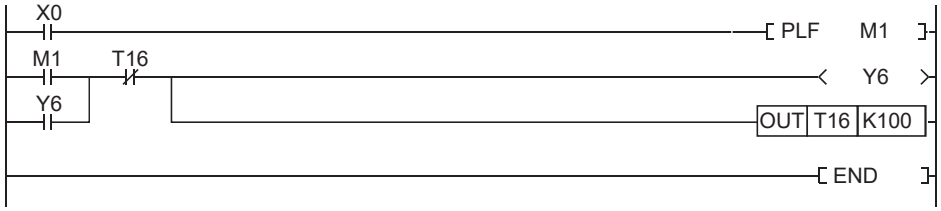
■Timing chart



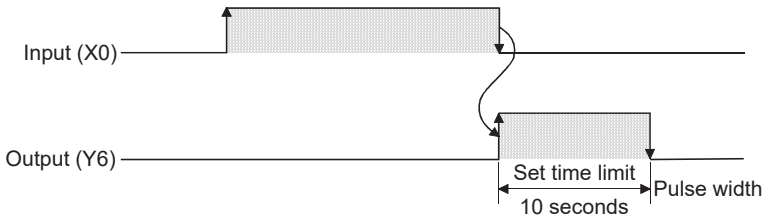
Useful application of the PLS/PLF instructions (Part 3)

These instructions can be used in a program that executes an output operation for a set period of time at timing when an input signal turns on.

■Program example

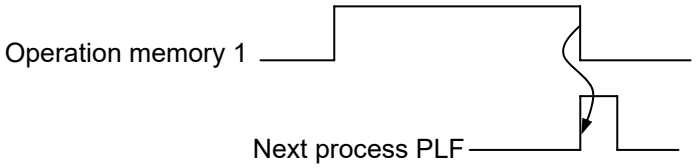


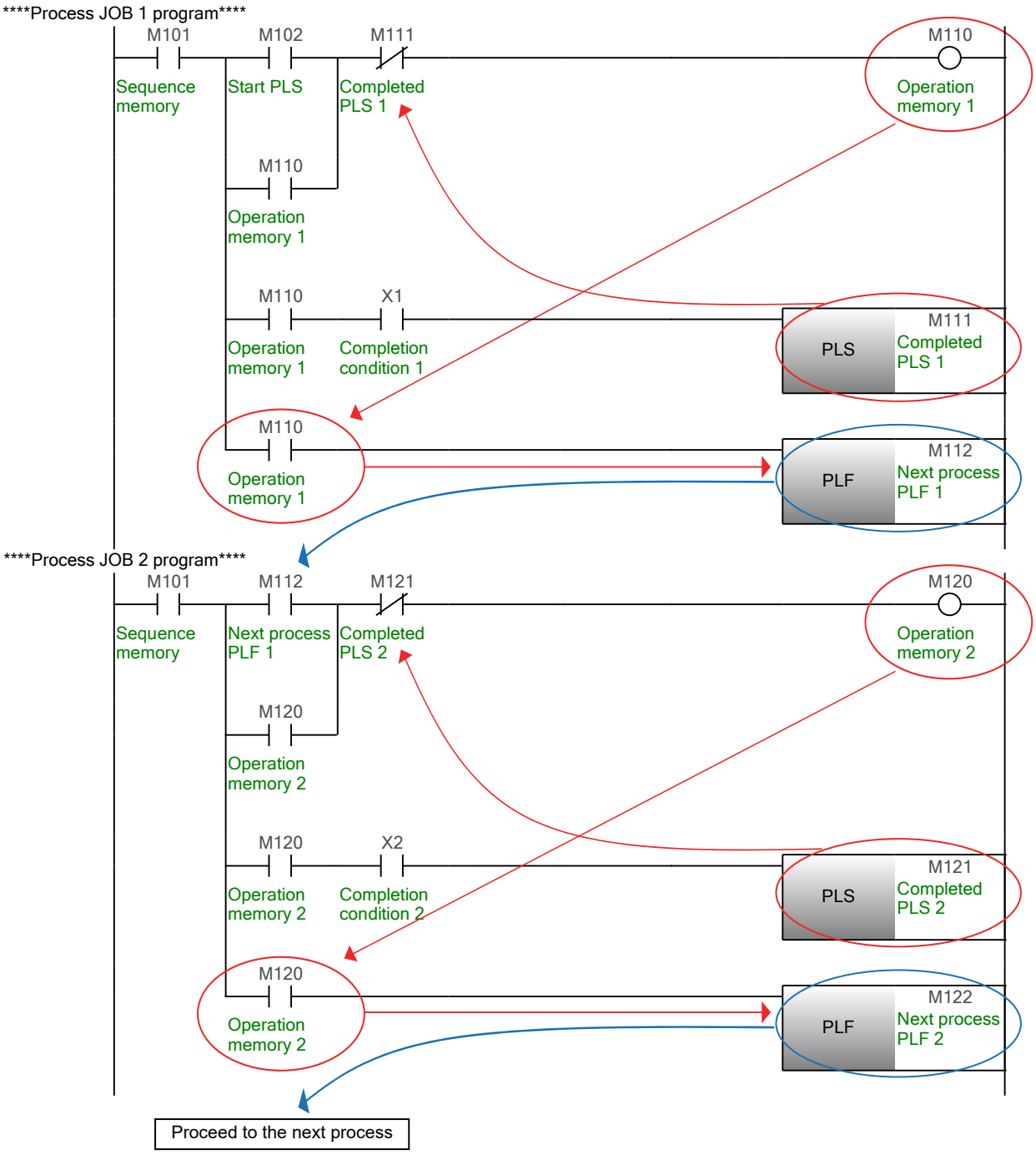
■Timing chart



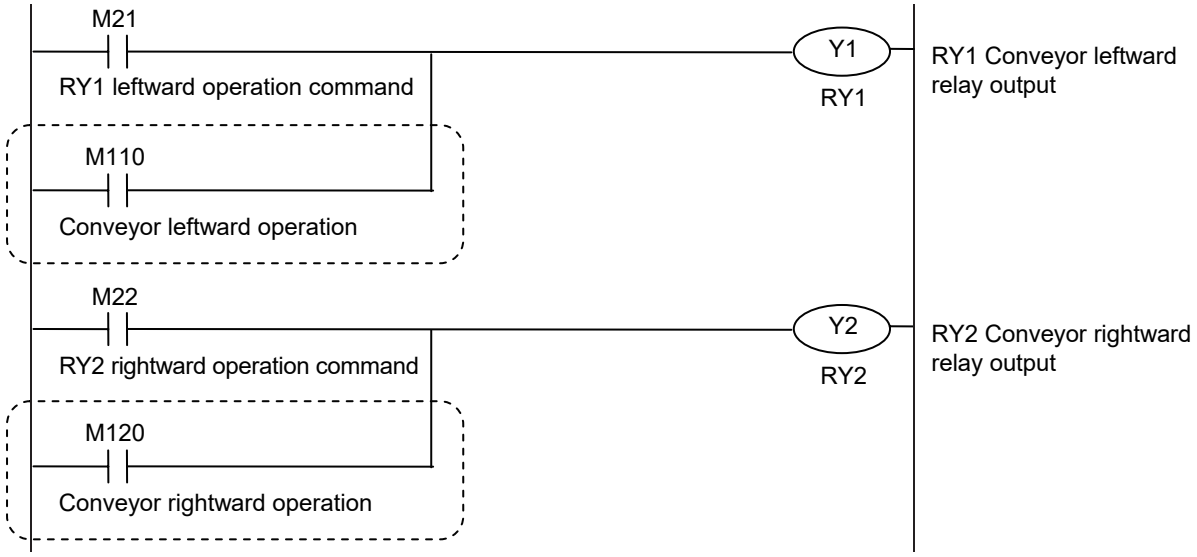
Sequence control example with self-holding function and PLF

When executing a control target in sequence control, create "Operation memory 1" and create an actual control program with the condition of "Operation memory 1".
"Operation memory 1"of the actual control program is reset with "Completed PLS 1".
To proceed to the next process, create "Next process PLF", which is a falling signal, to turn off "Operation memory 1" and start "Operation memory 2".





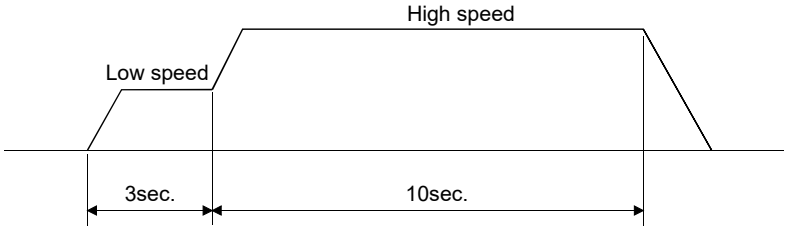
Output to device: Add auxiliary relays according to the device outputs.

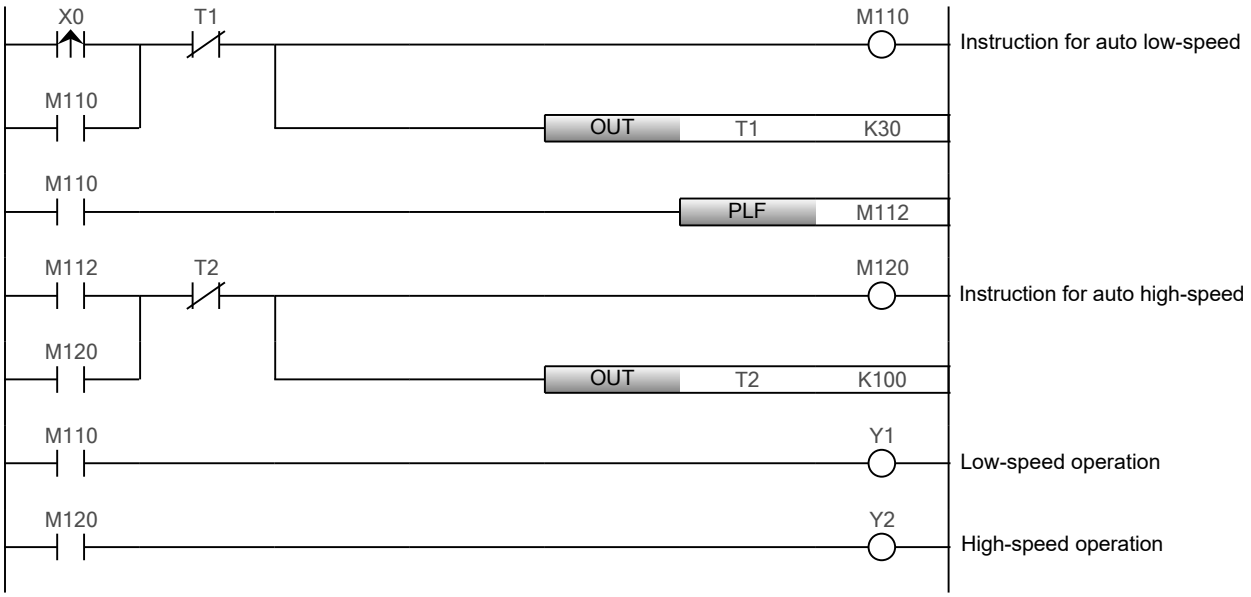


Ladder example 1

The following program runs automatic operations.

- After X0 is turned on, the system operates in the low-speed mode for 3sec. Then the system operates in the high-speed mode for 10sec. and stops.

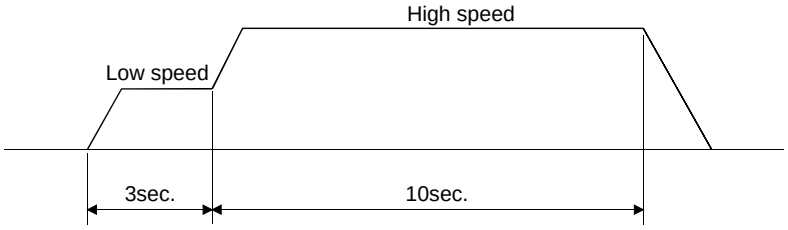


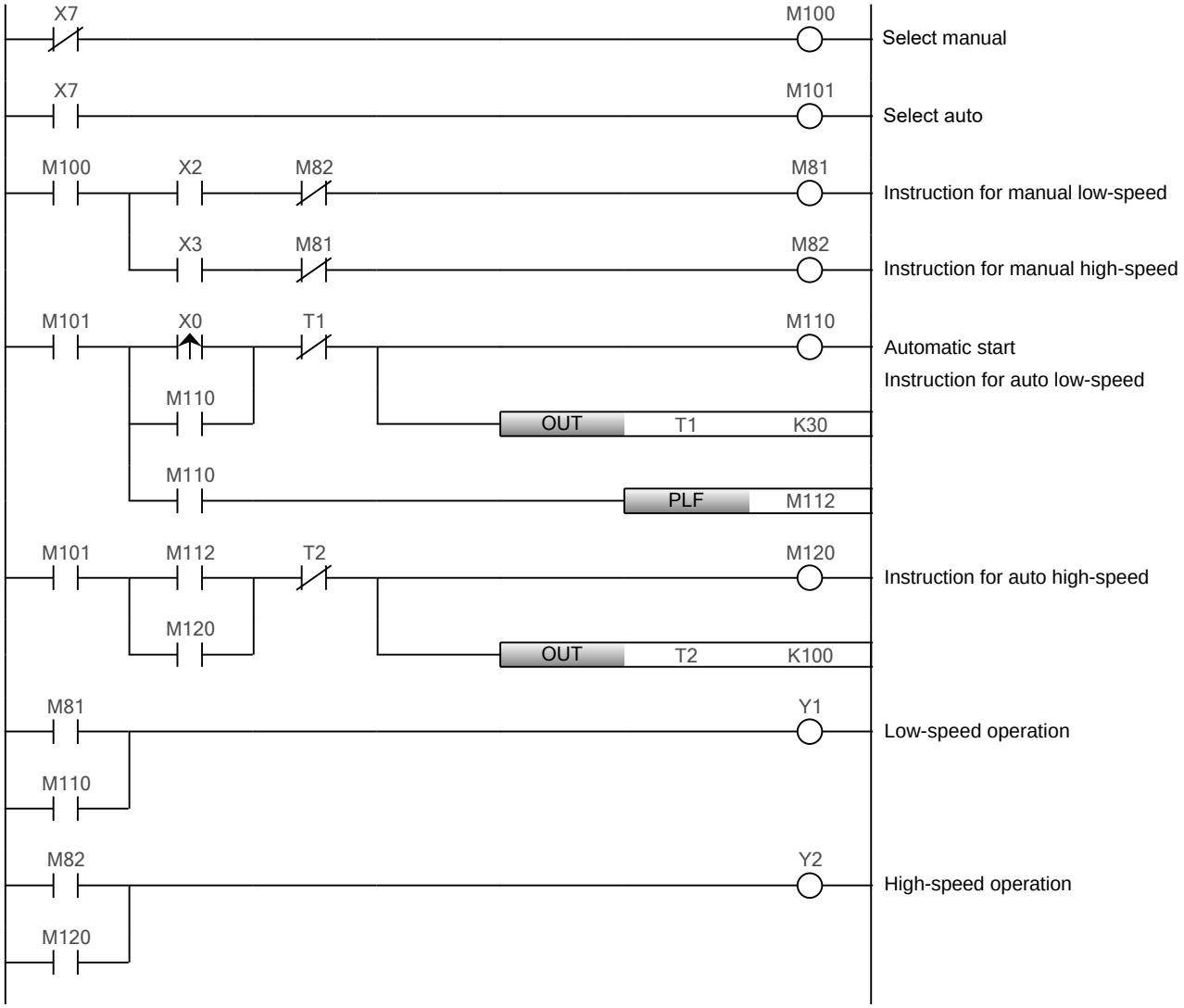


Ladder example 2

The following program switches between manual and automatic operations.

- When the manual operation is selected by turning off X7;
  1) Turning X2 sets the system to the low-speed operation mode.
  2) Turning X3 sets the system to the high-speed operation mode.
- When the automatic operation is selected by turning on X7.
  After X0 is turned on the system operates in the low-speed mode for 3sec.
  Then the system operates in the high-speed mode for 10sec. and stops.



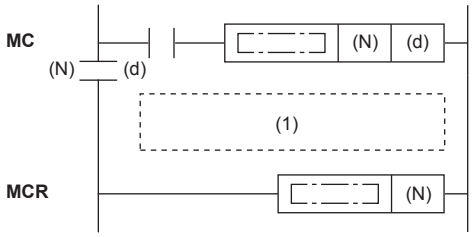


Master Control Instruction

Setting/resetting the master control

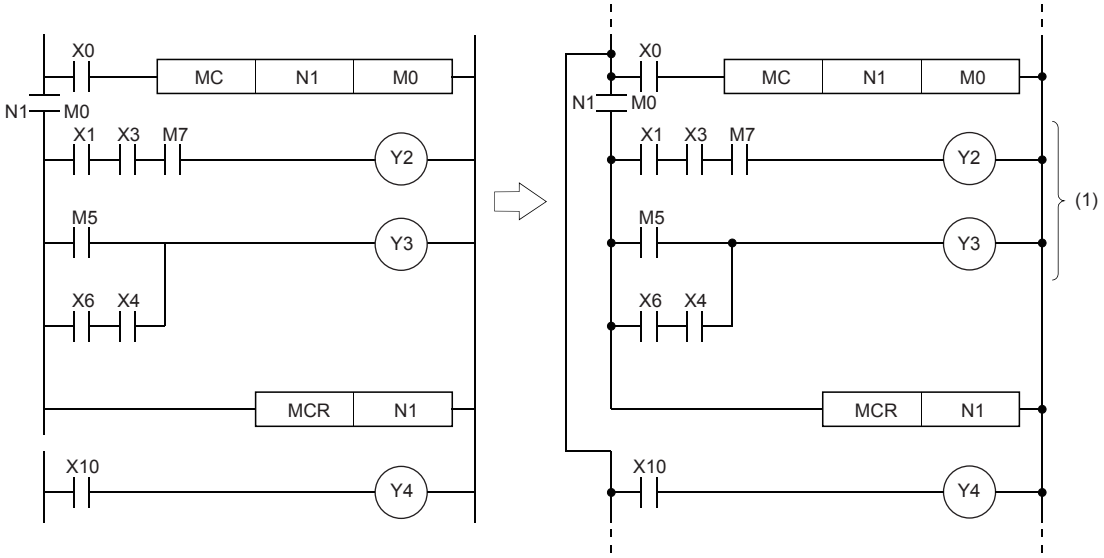
MC, MCR

- MC: This instruction starts master control.
- MCR: This instruction ends master control.



(1): Master control ladder

These instructions create program with efficient ladder switching by opening/closing common buses in ladders.
Ladder using master control is illustrated below. (Left: Display on the engineering tool, Right: Actual operation)



(1) Executed only when X0 is on

■ MC

- When the execution command of the MC instruction turns ON at the start of master control, the operation result between the MC and MCR instructions is as per the instructions (according to ladder). When the execution command of MC instruction turns OFF, the operation result between the MC and MCR instructions becomes as follows.

Device	Device status
Timer	The count value becomes 0, and both coils and contacts turn OFF.
Counters, retentive timers	Coils turn OFF but the current status of both count values and contacts is maintained.
Devices in OUT instruction	Forcibly turned OFF.
Devices in SET and RST instructions Devices in SFT(P) instruction Devices in basic instructions and applied instructions	Current status is maintained.

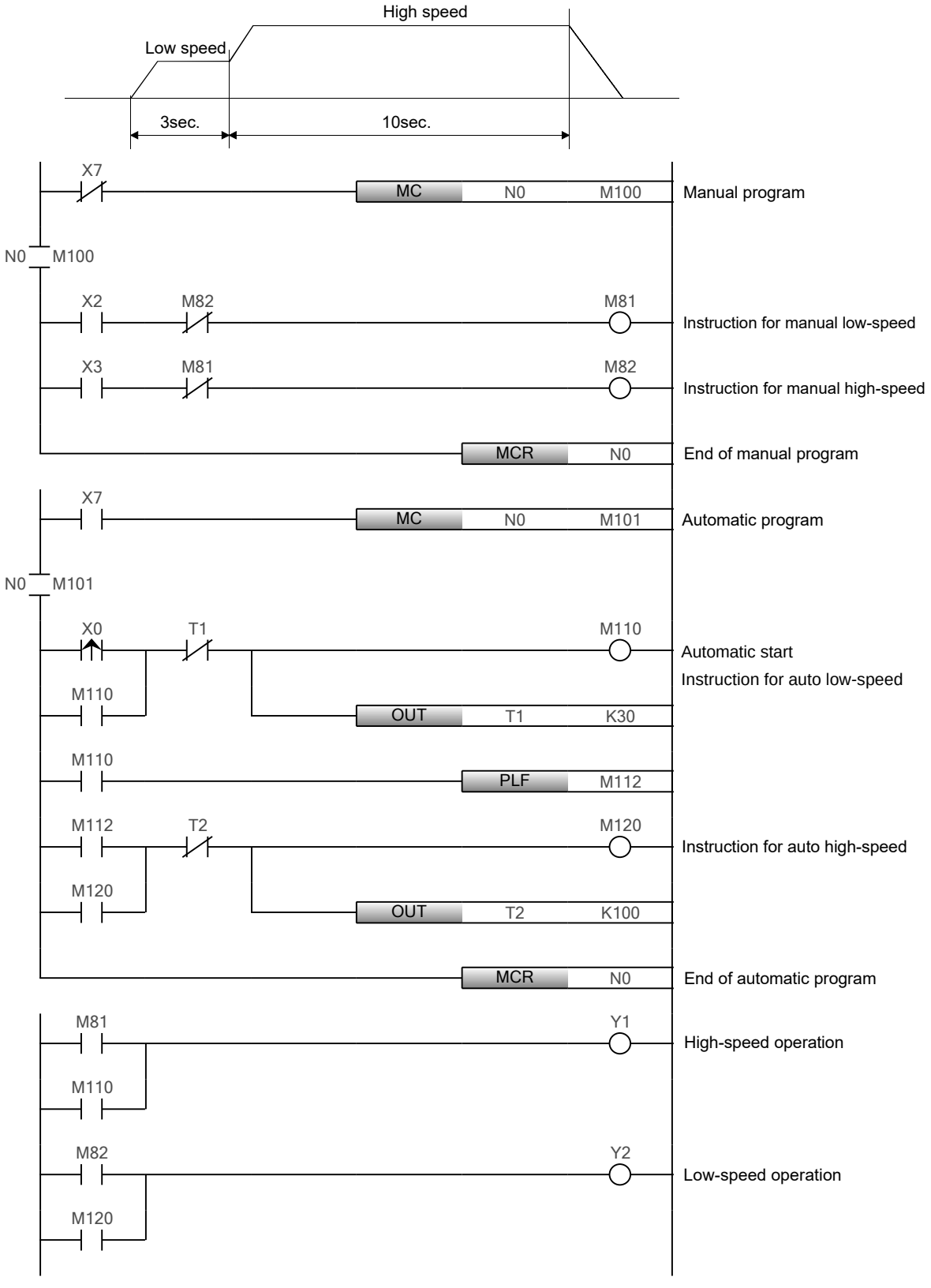
■ MCR

- This instruction indicates the end of the master control range by the master control release instruction.
- Do not prefix this instruction with NO contact instruction.
- Use these (MC and MCR) instructions with same nesting number as a pair. Note, however, that when this instruction is nested at a single location, all master controls can be ended by just one (N) number, the smallest number.

Ladder example

The program switches between manual and automatic operations using the MC and MCR instructions.

- When the manual operation is selected by turning off X7;
  1) Turning X2 sets the system to the low-speed operation mode.
  2) Turning X3 sets the system to the high-speed operation mode.
- When the automatic operation is selected by turning on X7.
  After X0 is turned on the system operates in the low-speed mode for 3sec.
  Then the system operates in the high-speed mode for 10sec. and stops.



Notation of Values (Data)

The programmable controller CPU converts all information into on or off signals (logical 1 or 0) to store and process them. Thus, the programmable controller executes numerical operations using the numerical values stored as logical 1 or 0 (binary numbers = BIN).

In daily life, decimal values are commonly used. Thus, the decimal-to-binary conversion or the binary-to-decimal conversion are required when values are read (monitored) or written from/to the programmable controller. The engineering tool and some instructions have the functions for those conversions.

This section describes how values (data) are expressed in decimal, binary, hexadecimal or binary-coded decimal notation (BCD), and how to convert values.

Decimal

- A decimal value consists of ten symbols, 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9, which represent the order and size (amount). After a digit reaches 9, an increment resets it to 0, causing an increment of the next digit to the left.
- The following shows how a decimal value (in this case 153) is represented.

$$153 = 100 + 50 + 3$$
$$= 1 \times 100 + 5 \times 10 + 3 \times 1$$
$$= 1 \times 10^2 + 5 \times 10^1 + 3 \times 10^0$$

Decimal symbol (0 to 9)

"Power of digit"

"Power of digit" can be expressed as follows.

$_n$: Digit number (0, 1, 2 ...)

10: Decimal value

- In the programmable controller, the symbol "K" is used to represent a value in decimal.

Binary (BIN)

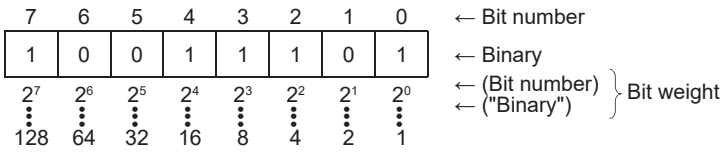
- A binary value consists of two symbols, 0 and 1, which represent the order and size (amount). After a digit reaches 1, an increment resets it to 0, causing an increment of the next digit to the left. One digit of 0 or 1 is called a bit.

Binary	Decimal
0	0
1	1
10	2
11	3
100	4
101	5
110	6
111	7
1000	8
⋮	⋮

- The following example describes how to convert a binary value into a decimal value.

  "10011101"

  The following figure shows the binary value with bit numbers and binary bit weights.



  The binary value is broken as follows.

  $= 1 \times 128 + 0 \times 64 + 0 \times 32 + 1 \times 16 + 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1$

  $= 128 + 16 + 8 + 4 + 1$

  $= 157$

  A binary value can be converted into a decimal value by the addition of the weight of each bit whose code is 1.

Hexadecimal

- A hexadecimal value consists of 16 symbols, 0 to 9 and A to F, which represent the order and size (amount). After a digit reaches F, an increment resets it to 0, causing an increment of the next digit to the left.

Decimal	Hexadecimal	Binary
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111
16	10	10000
17	11	10001
18	12	10010
⋮	⋮	⋮
19101	4A9D	0100 1010 1001 1101

$= (4) \times \underline{16^3} + (A) \times \underline{16^2} + (9) \times \underline{16^1} + (D) \times \underline{16^0}$

$= 4 \times 4096 + 10 \times 256 + 9 \times 16 + 13 \times 1$

$= 19101$

"Power of digit"

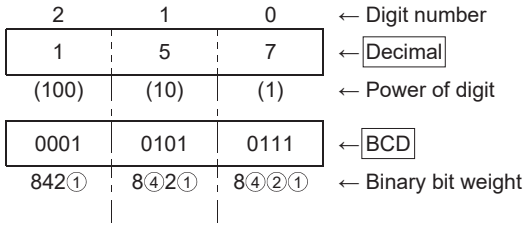
n •••• Digit number

16 •••••• Hexadecimal

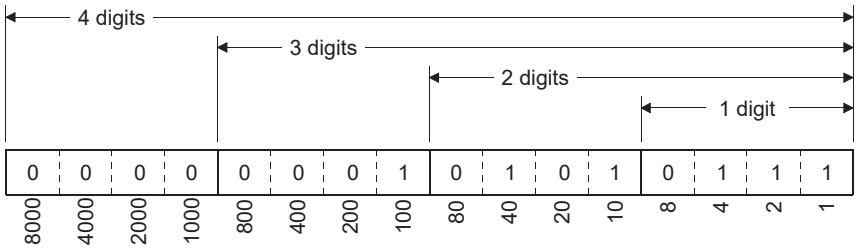
- Four bits of a binary value are equivalent to one digit of a hexadecimal value.
- In the programmable controller, the symbol "H" is used to represent a value in hexadecimal.
- Hexadecimal values are used to represent the following device numbers.
  - Link relay (B)
  - Link register (W)
  - Link special relay (SB)
  - Link special register (SW)

Binary-coded decimal (BCD)

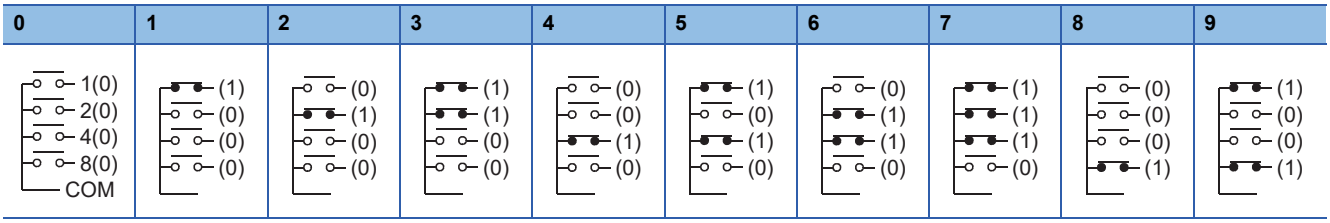
- The binary-coded decimal system uses a binary value to represent each digit of a decimal value.
  The decimal value 157, for example, is expressed as follows.



- In BCD, decimal values of 0 to 9999 (the maximum 4-digit value) can be represented with 16 bits.
  The following figure shows the weight of each bit in BCD.



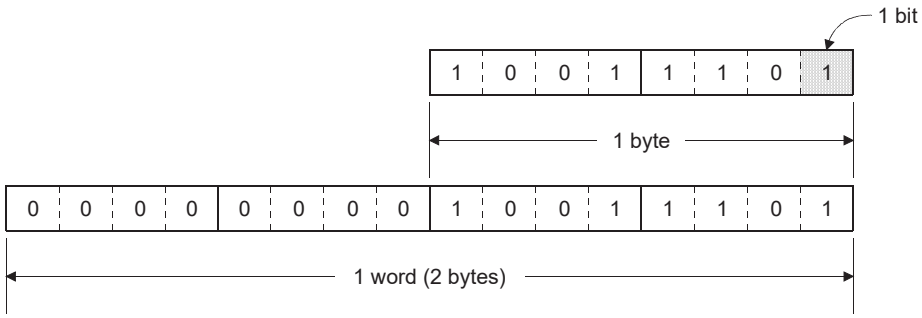
- BCD is used for the following signals.
  1) Output signals of digital switches
  2) Signals of seven-element display (digital HMI)



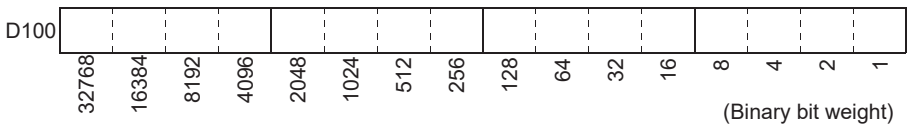
BCD code digital switch

Numerical values used by the programmable controller

- Usually, 8 bits are called one byte, and 16 bits (two bytes) are called one word.



- Registers of each word device in the programmable controller consist of 16 bits.
  - Data register (D)
  - Current value of a timer (T)
  - Current value of a counter (C)
  - File register (R)
  - Link register (W)

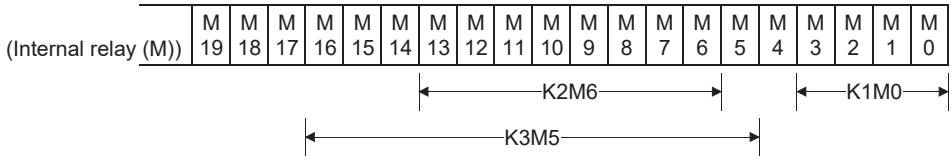


- Values in the following two ranges can be processed in 16 bits (one word).
  1) 0 to 65535
  2) -32768 to 0 to +32767
- The programmable controller uses the range 2).

Binary-coded decimal (BCD)	Binary (BIN)	Decimal (K)	Hexadecimal (H)
00000000 00000000	00000000 00000000	0	0000
00000000 00000001	00000000 00000001	1	0001
00000000 00000010	00000000 00000010	2	0002
00000000 00000011	00000000 00000011	3	0003
00000000 00000100	00000000 00000100	4	0004
00000000 00000101	00000000 00000101	5	0005
00000000 00000110	00000000 00000110	6	0006
00000000 00000111	00000000 00000111	7	0007
00000000 00001000	00000000 00001000	8	0008
00000000 00001001	00000000 00001001	9	0009
00000000 00010000	00000000 00001010	10	000A
00000000 00010001	00000000 00001011	11	000B
00000000 00010010	00000000 00001100	12	000C
00000000 00010011	00000000 00001101	13	000D
00000000 00010100	00000000 00001110	14	000E
00000000 00010101	00000000 00001111	15	000F
00000000 00010110	00000000 00010000	16	0010
00000000 00010111	00000000 00010001	17	0011
00000000 00011000	00000000 00010010	18	0012
00000000 00011001	00000000 00010011	19	0013
00000000 00100000	00000000 00010100	20	0014
00000000 00100001	00000000 00010101	21	0015
00000000 00100010	00000000 00010110	22	0016
00000000 00100011	00000000 00010111	23	0017
00000001 00000000	00000000 01100100	100	0064
00000001 00100111	00000000 01111111	127	007F
00000010 01010101	00000000 11111111	255	00FF
00010000 00000000	00000011 11101000	1000	03E8
00100000 01000111	00000111 11111111	2047	07FF
01000000 10010101	00001111 11111111	4095	0FFF
	00100111 00010000	10000	2710
	01111111 11111111	32767	7FFF
	11111111 11111111	-1	FFFF
	11111111 11111110	-2	FFFE
	10000000 00000000	-32768	8000

Digit Specification of Bit Devices (K1M0)

- A word device D (data register), T (current value of a timer), or C (current value of a counter) consists of 16 bits (one word), and data is basically transferred in one device.
- With 16 bit devices (such as X, Y, and M), data of the same size as a word device can be handled. The device numbers allocated to the bit devices must be in consecutive order.
- Bit devices can process data in units of four points.
- Other bit devices can also process data in the same way.

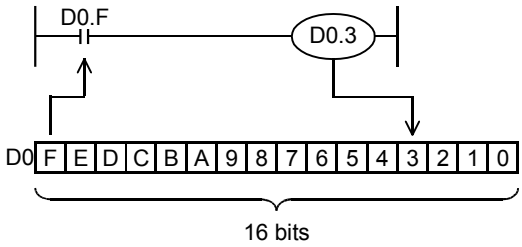


- As long as the device numbers of four bit device areas are in consecutive order, any bit device can be specified as the start device.

Bit Specification of a Word Device (D[ ].b)

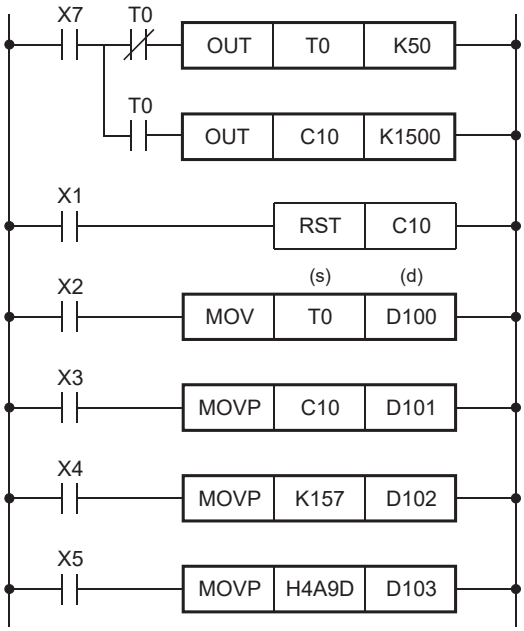
By specifying a bit of a word device, the specified bit can be used as bit data.
When specifying a bit of a word device, use a word device number and bit number (hexadecimal).
(Example: D0.0 ... Indicates the bit 0 of data register (D).)

Target word device : Data register or special data register
Bit number : 0 to F (hexadecimal)

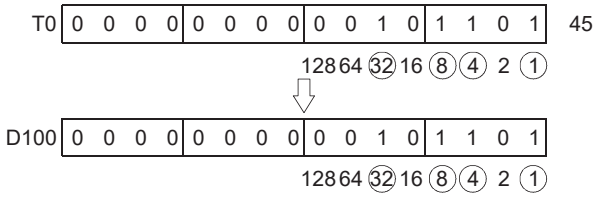


Transfer Instructions

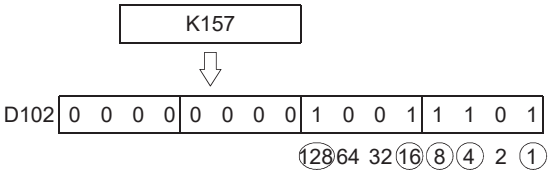
[MOV(P)] (Transferring 16-bit data)



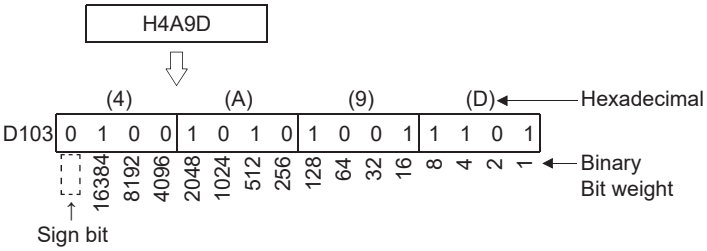
- When the input condition turns on, the current value of the timer T0 (source) is transferred into the data register D100 (destination).
- The current value of T0 in binary is transferred into D100 as it is. (Data conversion is not performed.)



- When the input condition turns on, the decimal number 157 is transferred into the data register D102. The decimal number (K) is automatically converted into a binary value, transferred to the data register D102, and stored there in binary.

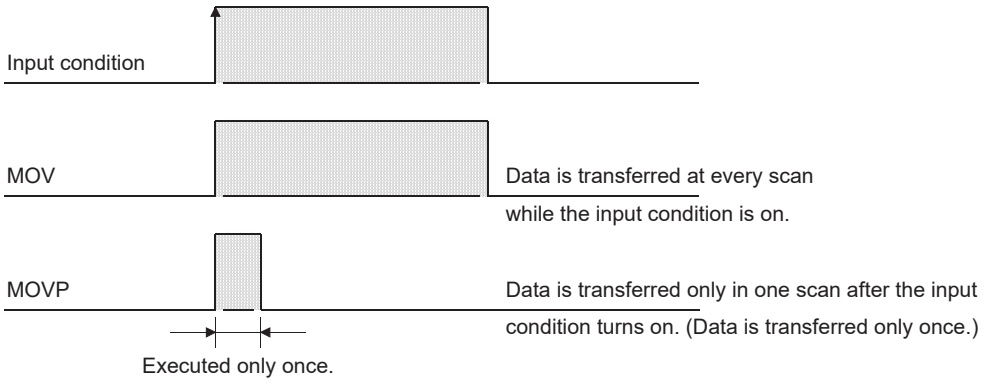


- When the input condition turns on, the hexadecimal value 4A9D is converted into a binary value and transferred into the data register D103.

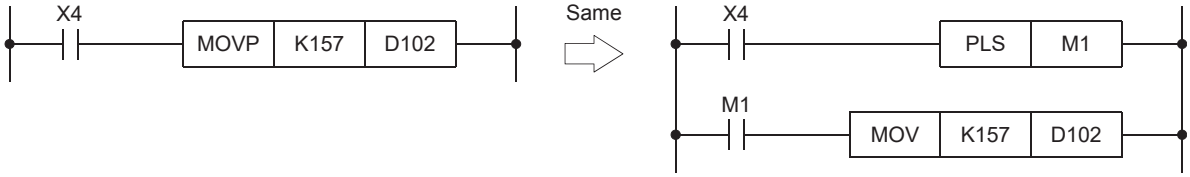


Differences between MOV and MOVP

The P in the MOVP instruction stands for a pulse.



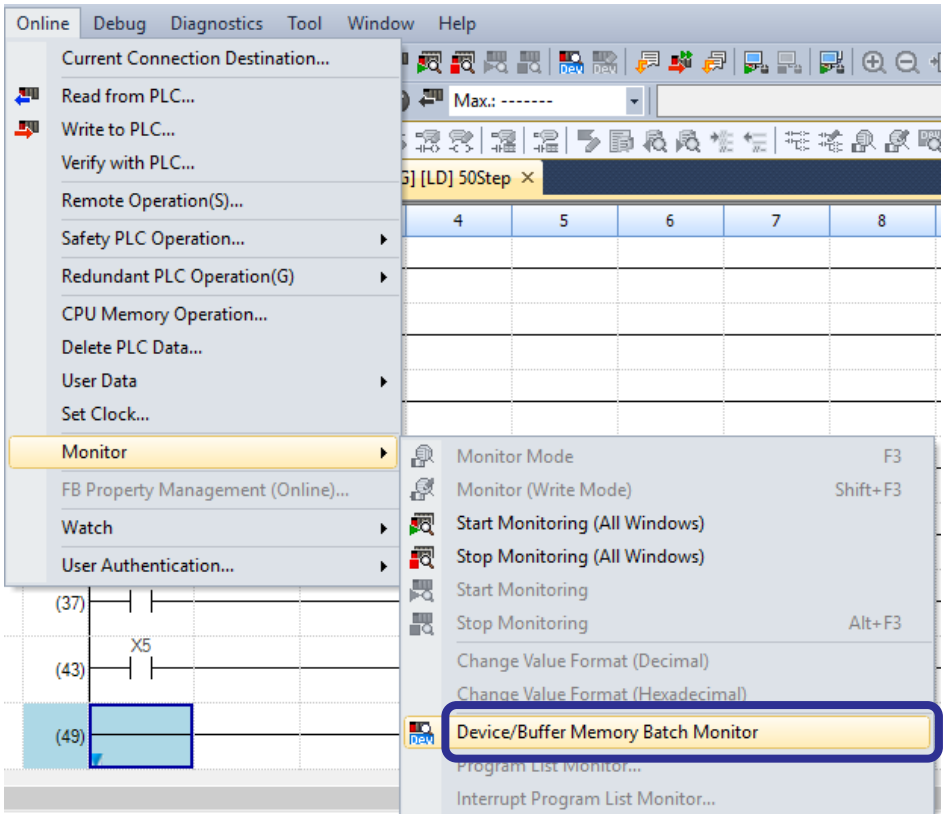
- Use the MOV instruction to read changing data all the time.
  Use the MOVP instruction to instantaneously transfer data such as when setting data or reading data at the occurrence of an error.



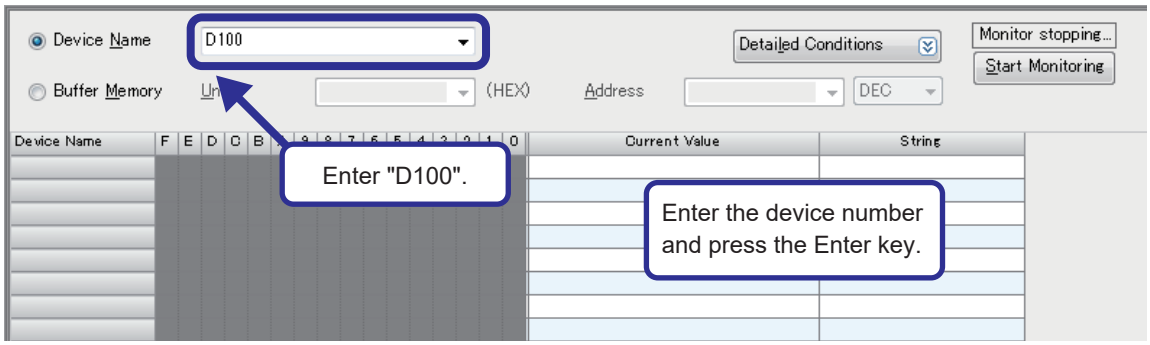
Items to be checked

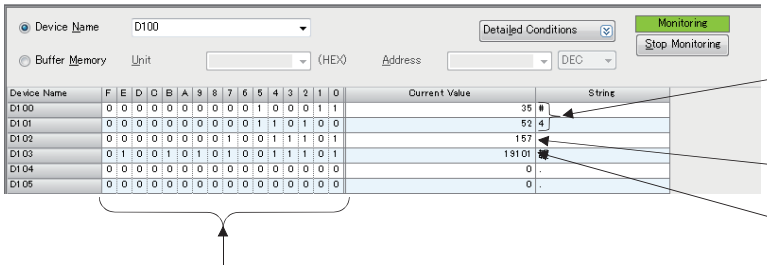
Inputs X2, X3, X4, X5, and X7: On

- Monitor the values in the data register (D100 to D103).
  After writing data to the programmable controller, click [Online] → [Monitor] → [Device/Buffer Memory Batch Monitor].
  The "Device/Buffer Memory Batch Monitor" dialog box appears.



• Enter "D100" in "Device Name" and press the Enter key.



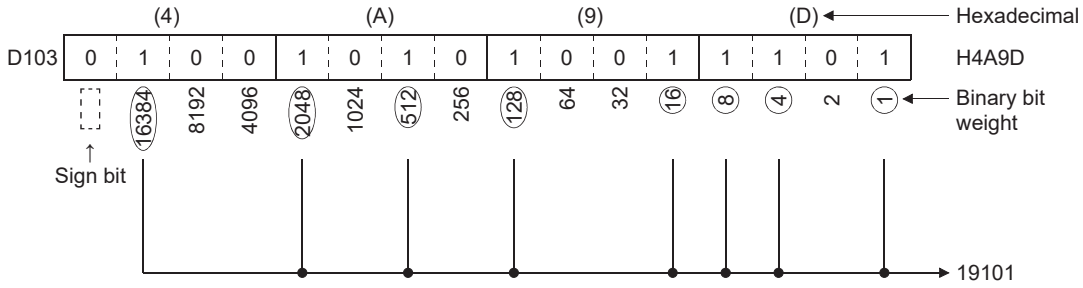


Current values of a timer and counter are monitored. (The values change.)

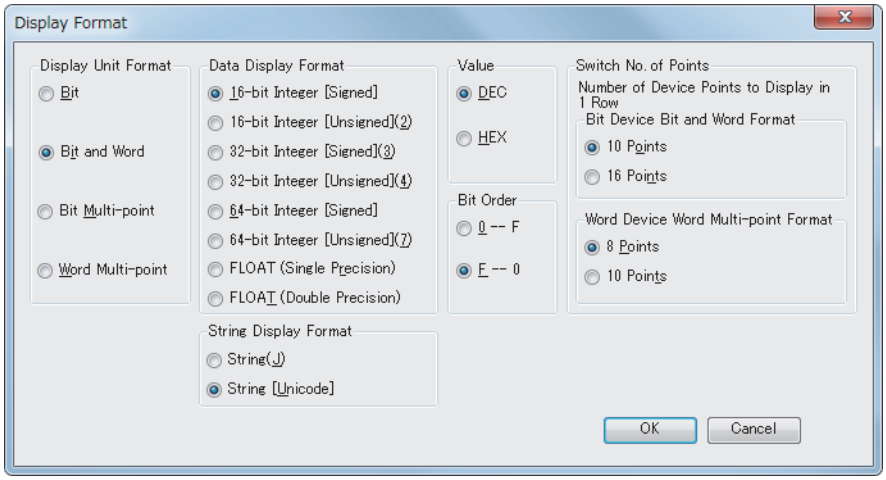
This value indicates that a decimal number 157 (K157) has been stored.

This value indicates the decimal number of the hexadecimal number 4A9D.

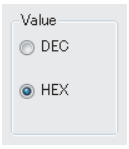
Word devices are expressed with the on/off states of bits.
0: Off (0 in binary)
1: On (1 in binary)



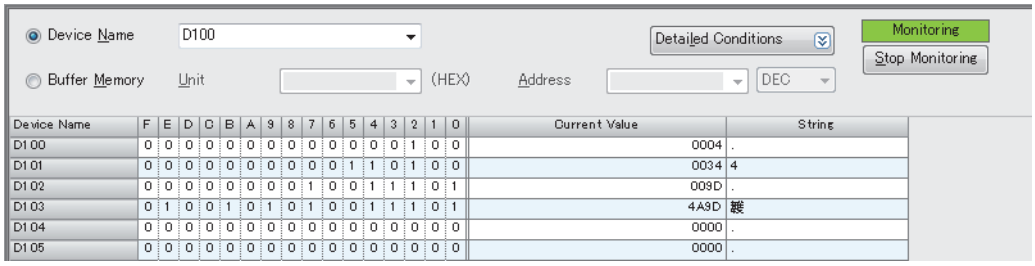
- Click on the toolbar or select [View] → [Display Format Detailed Setting] from the menu.
  The "Display Format" dialog box appears.



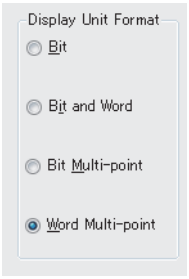
- Change the display of the numerical values being monitored to the hexadecimal notation.
  Select "HEX" for "Value" in the "Device Format" dialog box.



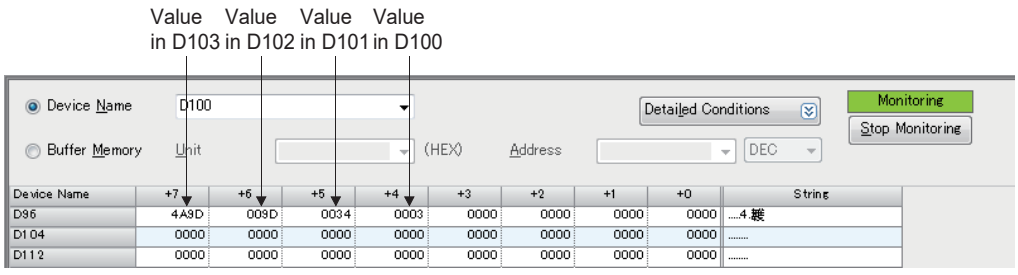
["Device/Buffer Memory Batch Monitor" window]



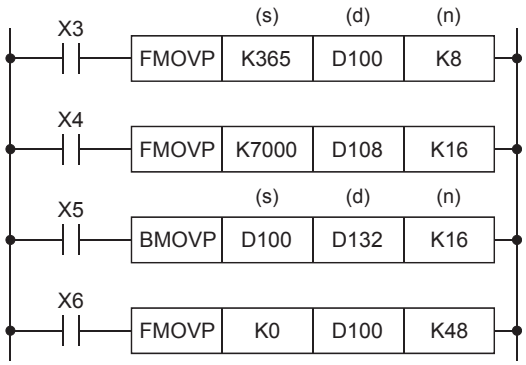
- Change the display of the numerical values being monitored in a multi-point format.
  Select "Word Multi-point" for "Display Unit Format" in the "Device Format" dialog box.



["Device/Buffer Memory Batch Monitor" window]

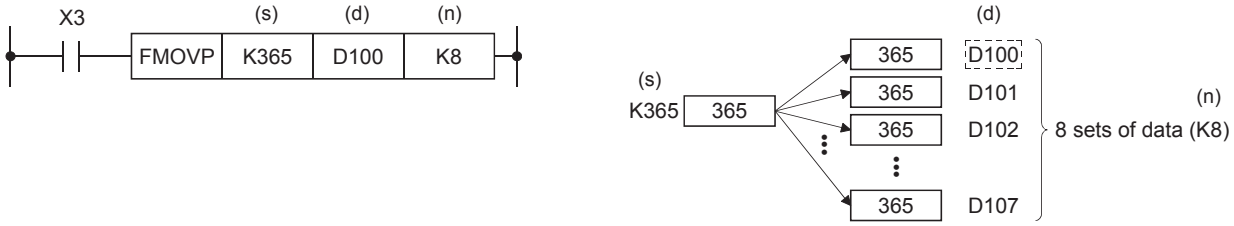


[FMOV(P)] (Transferring the same data in a batch)
[BMOV(P)] (Transferring block data in a batch)

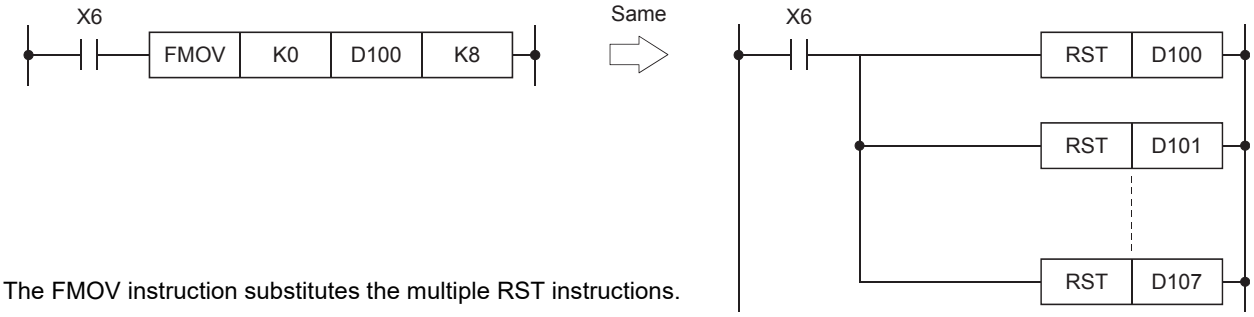


■FMOV When the input condition turns on, the FMOV instruction transfers data in the device specified by (s) to the (n) points of device areas starting from the device specified by (d).

The following figure shows the operation of when X3 turns on and the FMOV instruction is executed.



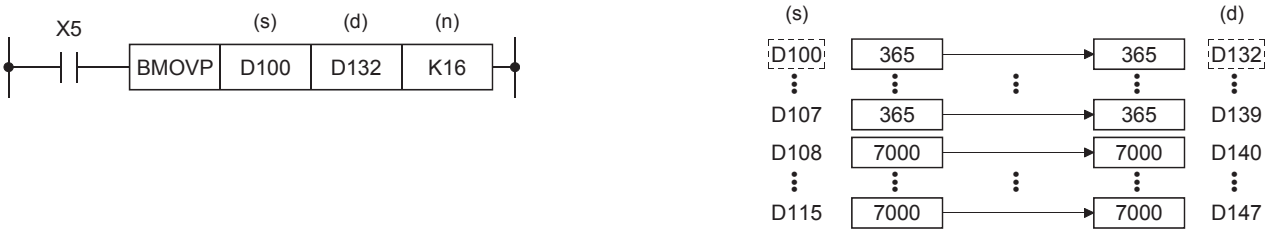
- The FMOV instruction is useful for clearing many data sets in a batch.



The FMOV instruction substitutes the multiple RST instructions.

■BMOV When the input condition turns on, the BMOV instruction batch-transfers the (n) points of data stored starting from the device specified by (s) to the areas starting from the device specified by (d).

The following figure shows the operation of when X5 turns on and the BMOV instruction is executed.



- The BMOV instruction is useful for the following applications.
  - Saving logging data in files
  - Saving important data (such as automatic operation data and measurement data) into the latch areas, for example the data register set for backing up data at power-off in parameter, to prevent data loss at an unintended power failure

Data Conversion Instructions

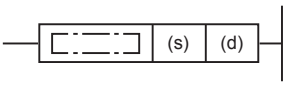
Converting binary data to BCD 4-digit data

BCD(P)

These instructions convert the binary data in the device specified by (s) to BCD data, and store the converted data in the device specified by (d).

Binary data is used in operations in CPU module. Use this instruction to display numeric values on seven-segment display unit equipped with BCD decoder.

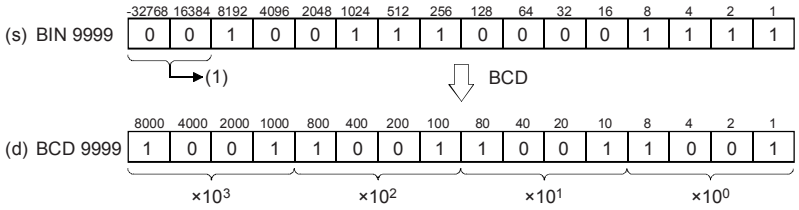
Ladder diagram



Operand	Description	Range	Data type
(s)	Binary data or the head device where the binary data is stored	0 to 9999	16-bit signed binary
(d)	Head device for storing the BCD data	—	BCD 4-digit

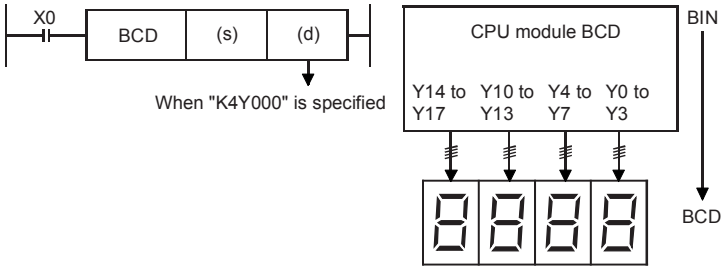
Processing details

- These instructions convert the 16-bit binary data (0 to 9999) in the device specified by (s) to BCD 4-digit data, and store the converted data in the device specified by (d).



(1): Set 0s.

- Data specified by (s) can be converted if it is within the range from K0 to K9999 BCD (decimal).
- The table below shows nibble specification for the data in the device specified by (s) and (d).



(d)	Number of digits	Data range
K1Y0	1-digit	0 to 9
K2Y0	2-digit	00 to 99
K3Y0	3-digit	000 to 999
K4Y0	4-digit	0000 to 9999

Precautions

- Binary data is used in all operations in CPU module including arithmetic operations (+-÷×), increment and decrement instructions. When receiving digital switch information in binary-coded decimal (BCD) format into a CPU module, use the BIN(P) instructions (for converting BCD data into binary data). Furthermore, to output data to seven-segment display unit handling binary-coded decimal (BCD) data, use the BCD(P) instructions (for converting binary data into BCD data).

Operation error

Error code (SD0/SD8067)	Description
3401H	Data in the device specified by (s) is out of the valid range (0 to 9999).

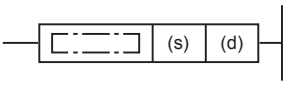
Converting BCD 4-digit data to binary data

BIN(P)

These instructions convert the binary-coded decimal data in the device specified by (s) to binary data, and store the converted data in the device specified by (d).

Use this instruction to convert a binary-coded decimal (BCD) value such as a value set by a digital switch into binary (BIN) data and to receive the converted binary data so that the data can be handled in operations in CPU module.

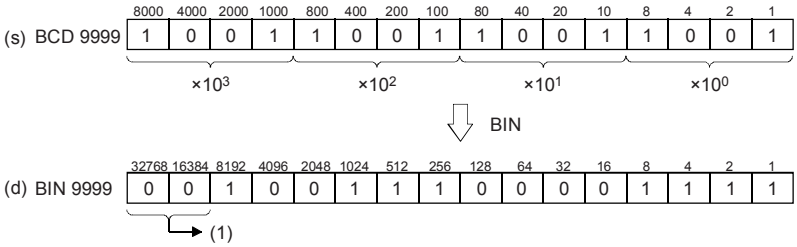
Ladder diagram



Operand	Description	Range	Data type
(s)	Binary-coded decimal data or the head device where the binary-coded decimal data is stored	0 to 9999	BCD 4-digit
(d)	Head device for storing the binary data	—	16-bit signed binary

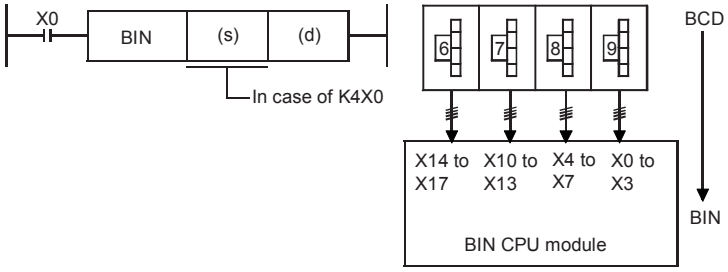
Processing details

- These instructions convert the BCD 4-digit data (0 to 9999) in the device specified by (s) to 16-bit binary data, and store the converted data in the device specified by (d).



(1): Filled with 0s.

- The data in the device specified by (s) can be converted if it is in the range from 0 to 9999 (BCD).
- The table below shows nibble specification for the data in the device specified by (s) and (d).



(d)	Number of digits	Data range
K1X0	1-digit	0 to 9
K2X0	2-digit	00 to 99
K3X0	3-digit	000 to 999
K4X0	4-digit	0000 to 9999

Precautions

- Binary data is used in all operations in CPU module including arithmetic operations (+-÷×), increment and decrement instructions. When receiving digital switch information in binary-coded decimal (BCD) format into a CPU module, use the BIN(P) instructions (for converting BCD data into binary data). Furthermore, to output data to seven-segment display unit handling binary-coded decimal (BCD) data, use the BCD(P) instructions (for converting binary data into BCD data).

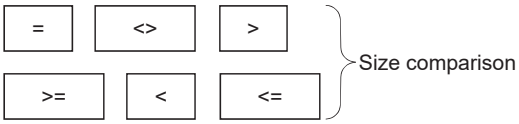
Operation error

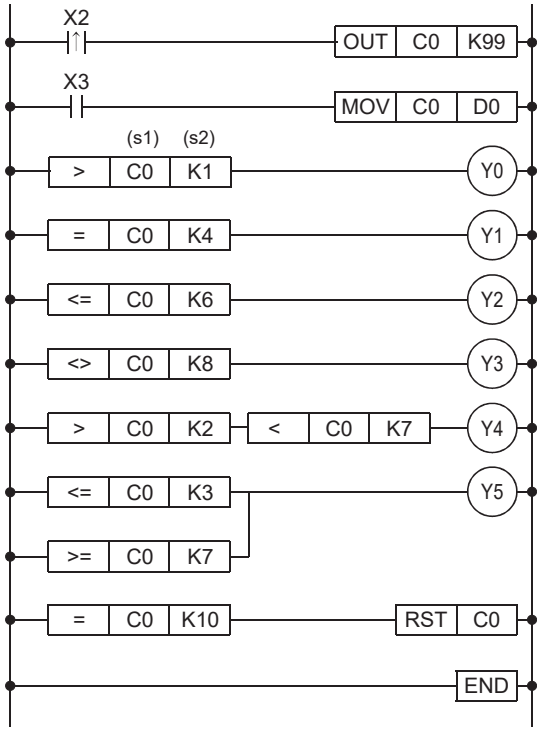
Error code (SD0/SD8067)	Description
3401H	The value of each digit of the device specified by (s) is other than 0 to 9. (The data is not binary-coded decimal data.)

Comparison Operation Instructions

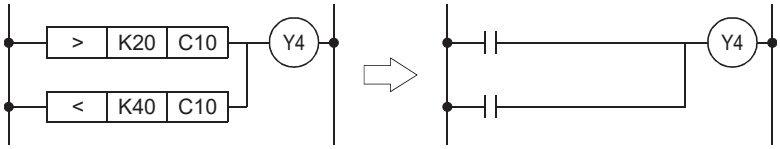
Point

This section describes how to compare numerical values.





- The comparison instruction compares the value in the source 1 (s1) and that in the source 2 (s2), and brings the devices in the continuity state when conditions are satisfied.
- The instruction can be regarded as one normally open contact (⊣⊢) because it goes in the continuity state only when conditions are satisfied.



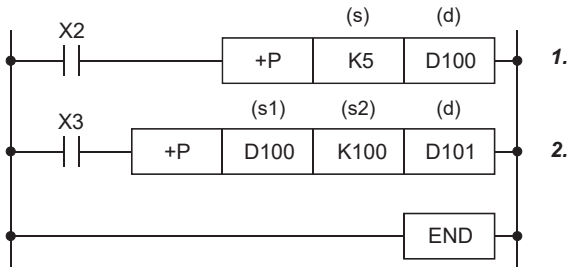
- = (s1) (s2) ••••• The ladder goes in the continuity state when the value in the source 1 is equal to that in the source 2.
- < (s1) (s2) ••••• The ladder goes in the continuity state when the value in the source 1 is smaller than that in the source 2.
- > (s1) (s2) ••••• The ladder goes in the continuity state when the value in the source 1 is larger than that in the source 2.
- <= (s1) (s2) ••••• The ladder goes in the continuity state when the value in the source 1 is equal to or smaller than that in the source 2.
- >= (s1) (s2) ••••• The ladder goes in the continuity state when the value in the source 1 is equal to or larger than that in the source 2.
- <> (s1) (s2) ••••• The ladder goes in the continuity state when the value in the source 1 are not equal to that in the source 2.

Arithmetic Operation Instructions

[+(P)] (Addition of 16-bit binary data)
[-(P)] (Subtraction of 16-bit binary data)

Point

- This section describes addition or subtraction.
- This section describes the differences between the instructions with P and the one without P.



1. Every time the input condition turns on, the value in the device specified in (s) is added to the value in the device specified in (d), and the result is stored in the device specified in (d).

	(d)		(s)		(d)
	D100 □	+	(5)	→	D100 □
(Input condition)					
First: ON	0 (example)	+	5	→	5
Second: ON	5	+	5	→	10
Third: ON	10	+	5	→	15

The value in D100 changes.

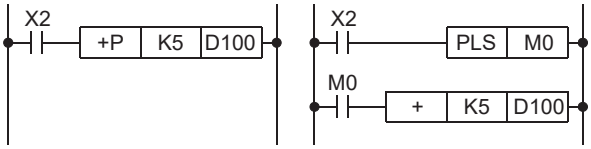
2. When the input condition turns on, the value in the device specified in (s1) is added to the value in the device specified in (s2), and the result is stored in the device specified in (d).

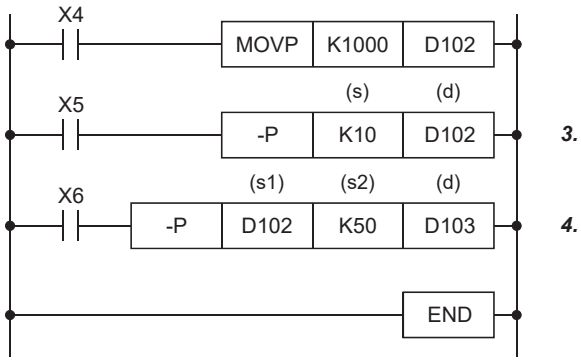
	(s1)		(s2)		(d)
	D100 □	+	(100)	→	D101 □
(Input condition)					
ON	15 (example)	+	100	→	115

The value in D100 is not changed by the addition.

Precautions

- Always use +P or -P as the addition or subtraction instructions.
- When + or - is used, an addition or subtraction operation is executed at every scan. To use + or -, convert operands into pulse in advance.





3. Every time the input condition turns on, the value in the device specified in (s) is subtracted from the value in the device specified in (d), and the result is stored in the device specified in (d).

	(d)		(s)		(d)
	D102	-	(10)	→	D102
(Input condition)					
First: ON	1000 (example)	-	10	→	990
Second: ON	990	-	10	→	980
Third: ON	980	-	10	→	970

The value in D102 changes.

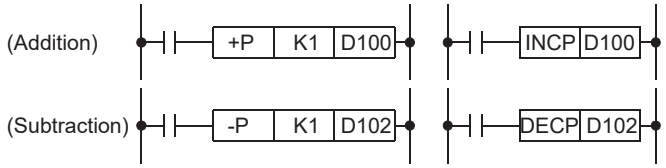
4. When the input condition turns on, the value in the device specified in (S2) is subtracted from the value in the device specified in (S1), and the result is stored in the device specified in (d).

	(s1)		(s2)		(d)
	D102	-	(50)	→	D103
(Input condition)					
ON	970 (example)	-	50	→	920

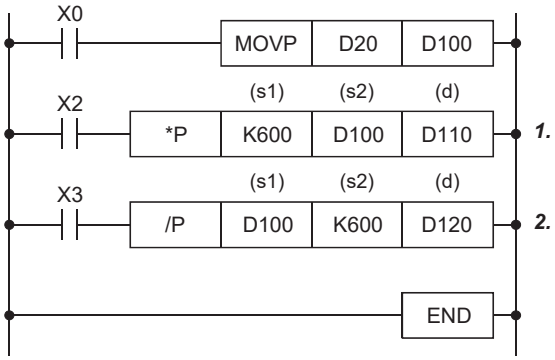
The value in D102 is not changed by the subtraction.

Point

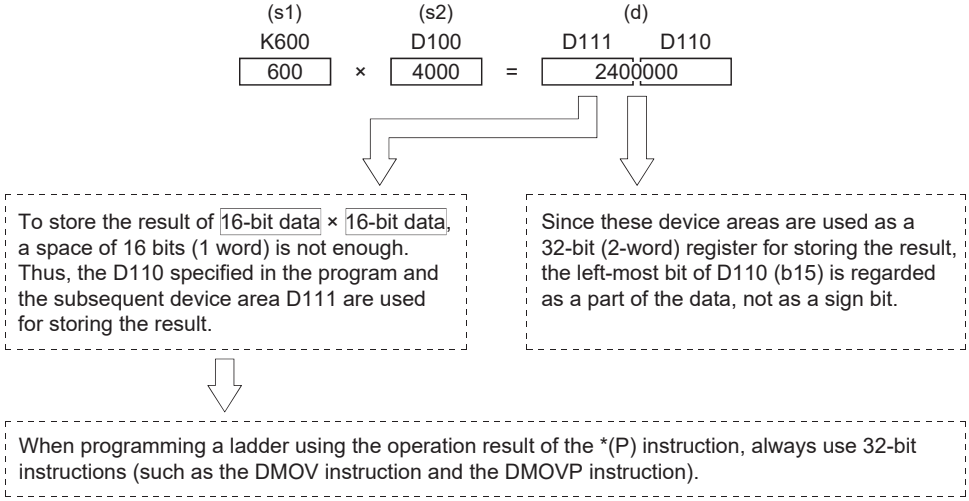
The following two instructions work on the same principle in the addition or subtraction processing.



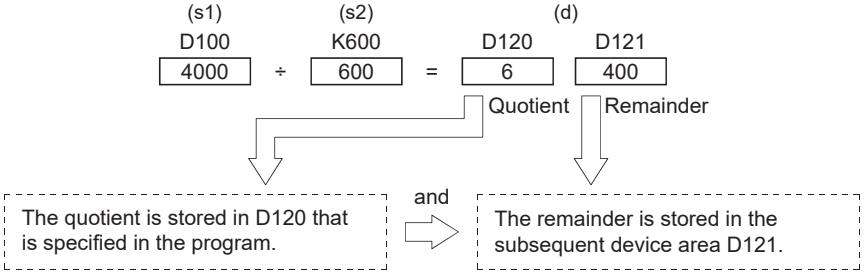
[*(P)] (Multiplication of 16-bit binary data)
[/(P)] (Division of 16-bit binary data)



1. When the input condition turns on, the value in the device specified in (s1) is multiplied by the value in the device specified in (s2), and the result is stored in the device specified in (d).



2. When the input condition turns on, the value in the device specified in (s1) is divided by the value in the device specified in (s2), and the result is stored in the device specified in (d). Values after the decimal point of the division result are ignored.



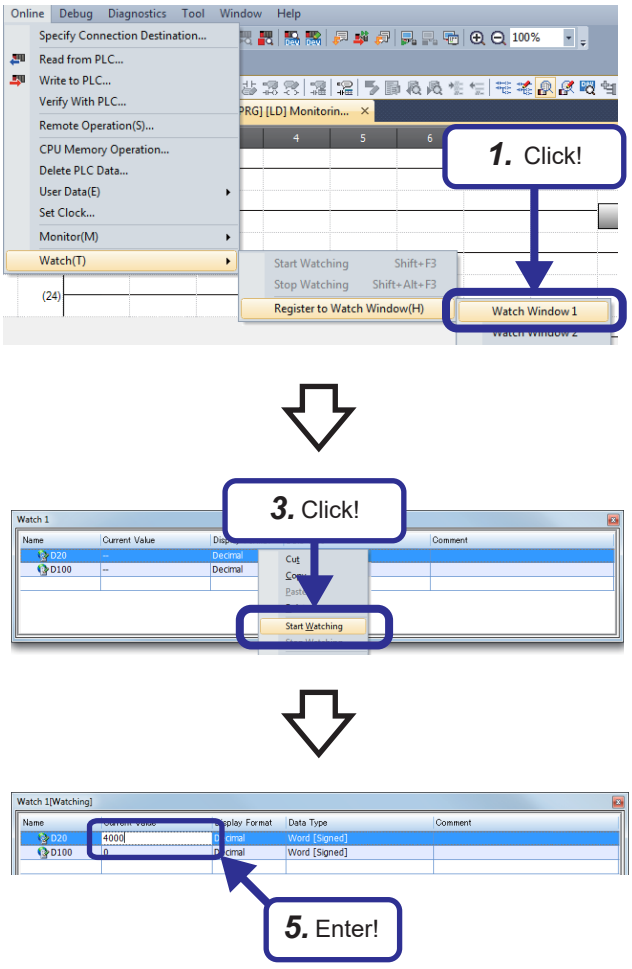
When a bit device is specified in (d), the quotient is stored but the remainder is not stored.
The following shows examples of processing of negative values.

Example -5 ÷ (-3) = 1, remainder = -2
5 ÷ (-3) = -1, remainder = 2

The following shows examples of dividing a value by 0 or dividing 0 by a value.

Example 0 ÷ 0, 1 ÷ 0 } Error "OPERATION ERROR"
0 ÷ 1 Quotient and remainder = 0

Current value change of the device



Set the CPU module to the RUN state before this operation.

1. Select the "D20" cell on the ladder editor and click [Online] → [Watch] → [Register to Watch Window] → [Watch Window 1].

2. "D20" is registered in the "Watch 1" window.

3. Right-click "D20" in the "Watch 1" window and click [Start Watching].

4. Watching of "Watch 1" starts.

5. Enter "4000" in "Current Value".
After entering the current value, press the Enter key and check that the value in the initial indication device (D20) changes to 4000.

Point

The current values of word devices can also be registered, started watching and changed in the watch window with selection the cell on the ladder editor, and press Shift + Enter while the program is being monitored.

32-bit data instructions and their necessities

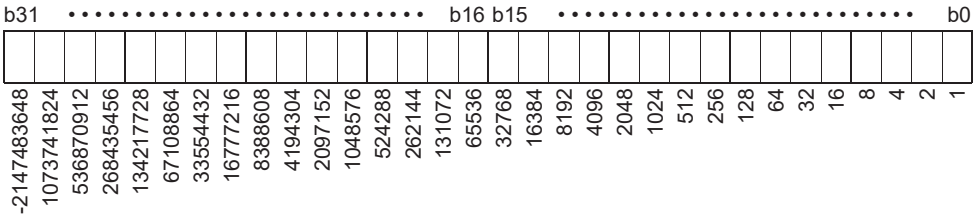
Point

- This section describes the concept of two words.
- This section describes the differences between a one-word instruction and two-word instruction.

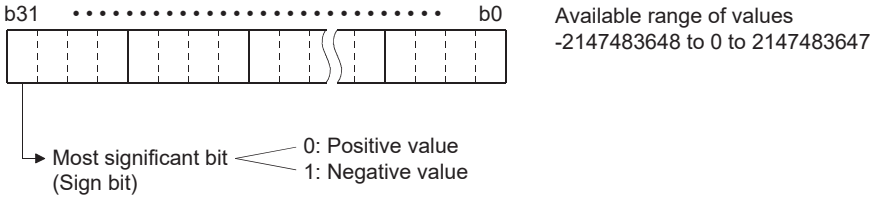
- The unit of the data memory of the programmable controller is one word that consists of 16 bits. Thus, data is typically processed in units of one word at the transfer processing, comparison, and arithmetic operation.
- The programmable controller can process data in units of two words (32 bits). In that case, "D" is added at the beginning of each instruction to indicate that the instruction processes two-word data. The following shows examples.

Instruction	1 word 16 bits	2 words 32 bits
Transfer	MOV(P)	DMOV(P)
Comparison	<, >, <=, >=, =, <>	D<, D>, D<=, D>=, D=, D<>
Arithmetic operation	+(P)	D+(P)
	-(P)	D-(P)
	*(P)	D*(P)
	/(P)	D/(P)
Available range of numerical values	-32768 to 32767	-2147483648 to 2147483647
Available range of digit specification	K1 to K4	K1 to K8

- The following shows the weights of 32 bits.

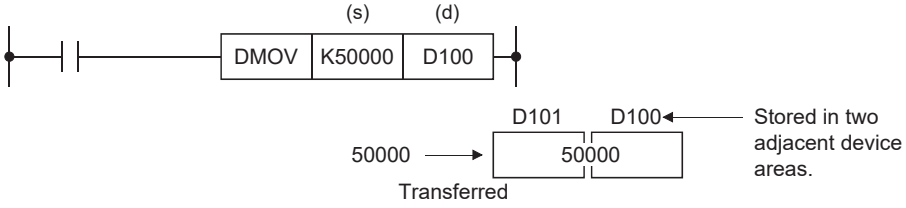


As the case of 16-bit data processing, the programmable controller takes the 2's complement in 32-bit data processing. Thus, the most significant bit b31 (b15 for 16-bit data) is processed as a sign bit.

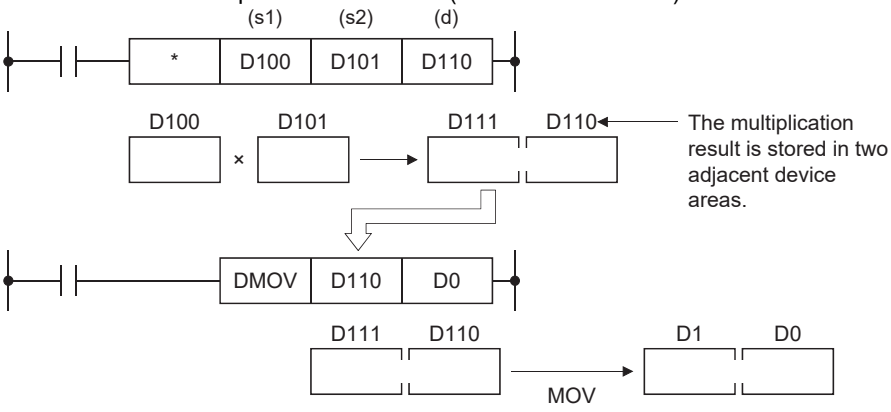


- Whether data is processed as two-word (32-bit) data or not depends on the size of the data.
  In the following cases, use two-word instructions.

(1) When the data size exceeds the range (-32768 to 32767) in which data can be processed as one word

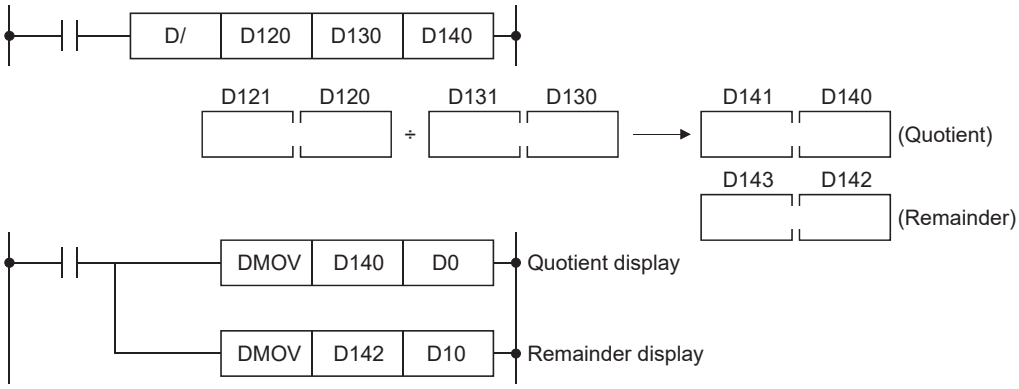


(2) When the result of the 16-bit multiplication instruction (one-word instruction) is transferred



*1 The result of the 32-bit data multiplication will be 64-bit data.

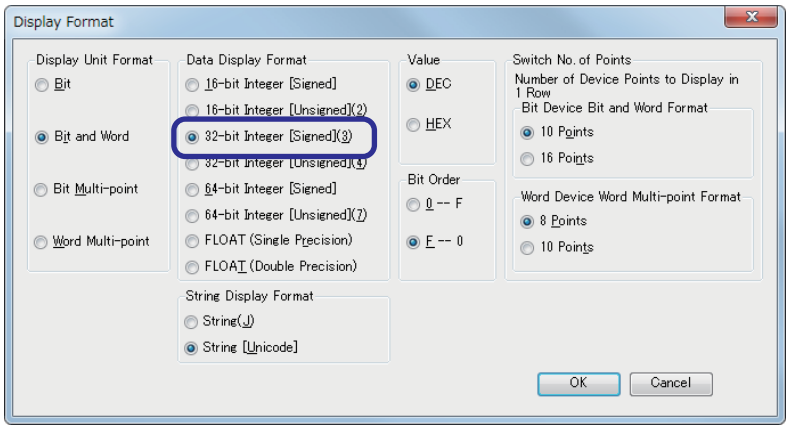
(3) When the result of the 32-bit division instruction is used



Point

When the value is outside the range of 0 to 32767, it cannot be properly displayed.

- Set "32-bit Integer [Signed]" for "Data Display Format". Click the [OK] button.



- Values are monitored properly.

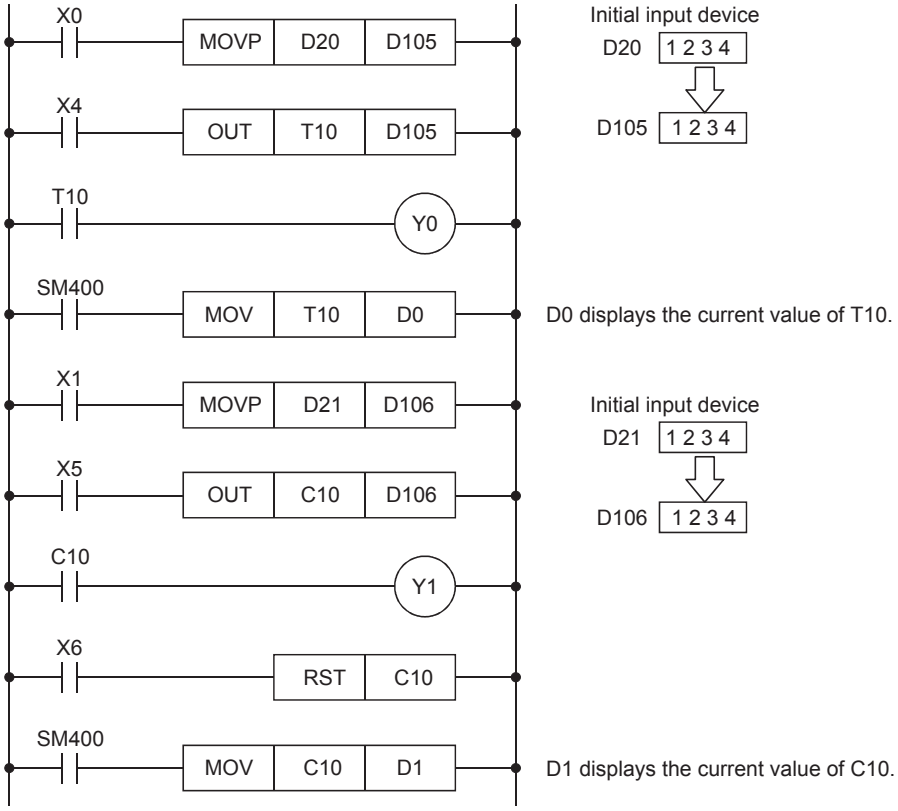
Device Name	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	Current Value
D100	0	0	0	0	1	1	1	1	1	0	1	0	0	0	0	0	4000
D101	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
D102	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D103	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
D104	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D105	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
D106	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D107	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
D108	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D109	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
D110	1	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	2400000

External Setting of Timer/Counter Values and External Display of Current Values

Point

This section describes how to indirectly specify a setting value of a timer/counter.

A setting value of a timer or counter can be directly specified with K (decimal constant) or indirectly specified with D (data register). In the program shown below, the setting value can be changed from an external device.

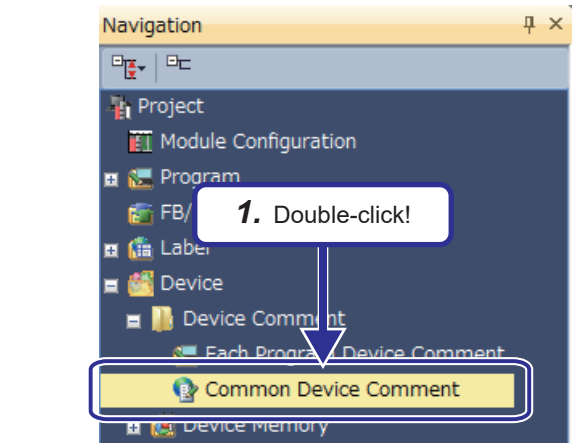


How to Create Comments

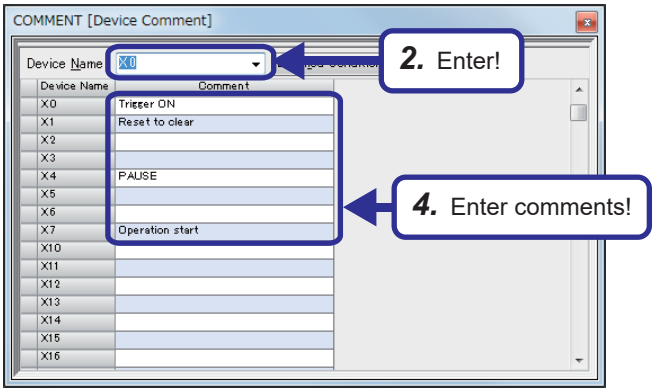
Point

This section describes how to create comments (device comments, statements, and notes) in a program.

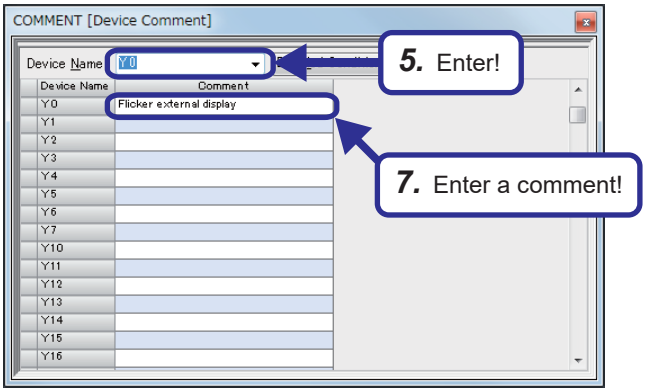
Creating comments



1. Click [Device] → [Device Comment] in the "Project" view and double-click [Common Device Comment] to display the "Device Comment" window.



2. Click "Device Name" and enter "X0" in the list box.
3. Press the Enter key.
4. Click each of comment areas and enter comments as shown on the left.



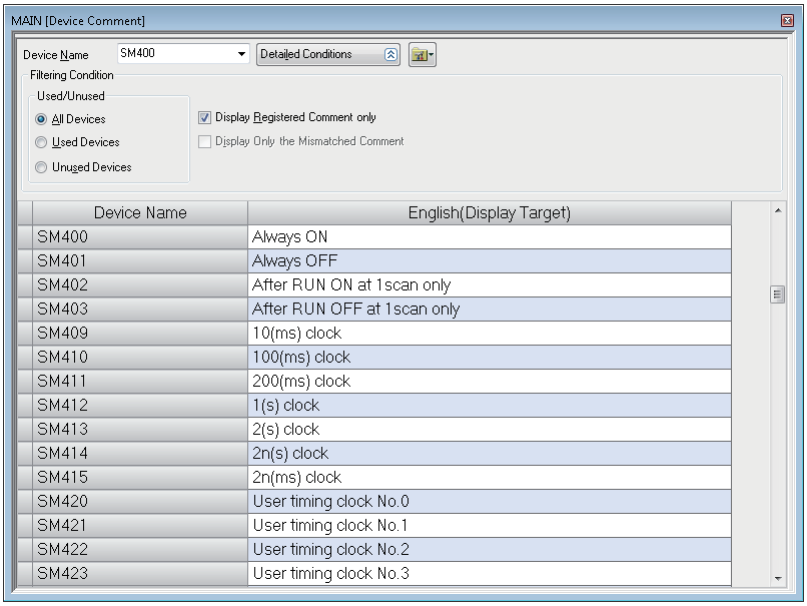
5. Click "Device Name" and enter "Y0" in the list box.
6. Press the Enter key.
7. Click a comment area and enter a comment as shown on the left.

Reading sample comments

The comments of system devices (special relays/special registers) can be applied to a device comment automatically.

1. Open the device comment editor.
2. Select [Edit] ⇨ [Read from Sample Comment].
3. Set each item in the "Read from Sample Comment" screen, and click the [OK] button.

The sample comments are applied to the device comment editor.



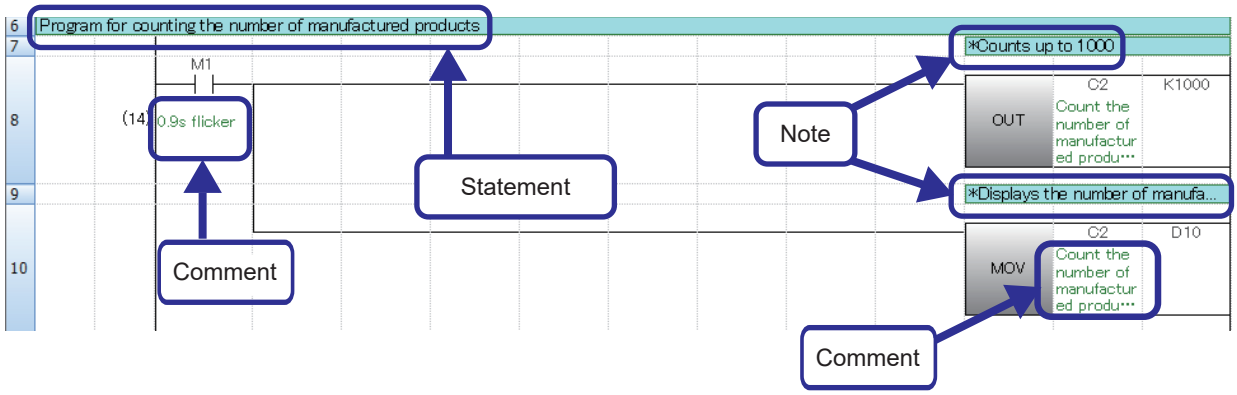
Displaying a ladder with comments in windows of GX Works3

Click [View] → [Comment Display] from the menu.

Statments and notes

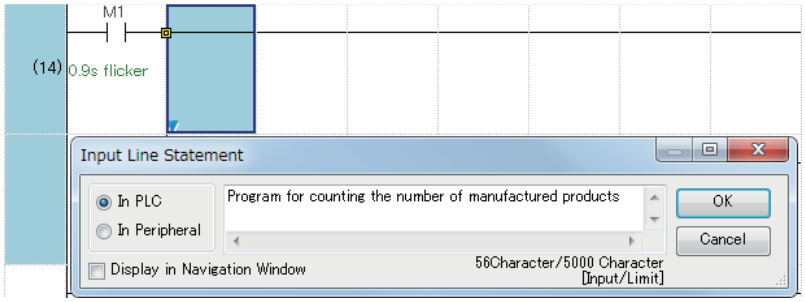
As well as device comments, statements and notes can be created in a ladder.

- Statement: A comment that describes the function or application of a ladder block
- Note: A comment that describes the function or application of an output or instruction
- Statements and notes are classified into two categories: "In PLC" (statements and notes can be stored in a CPU) and "In Peripheral" (statements and notes cannot be stored in a CPU).

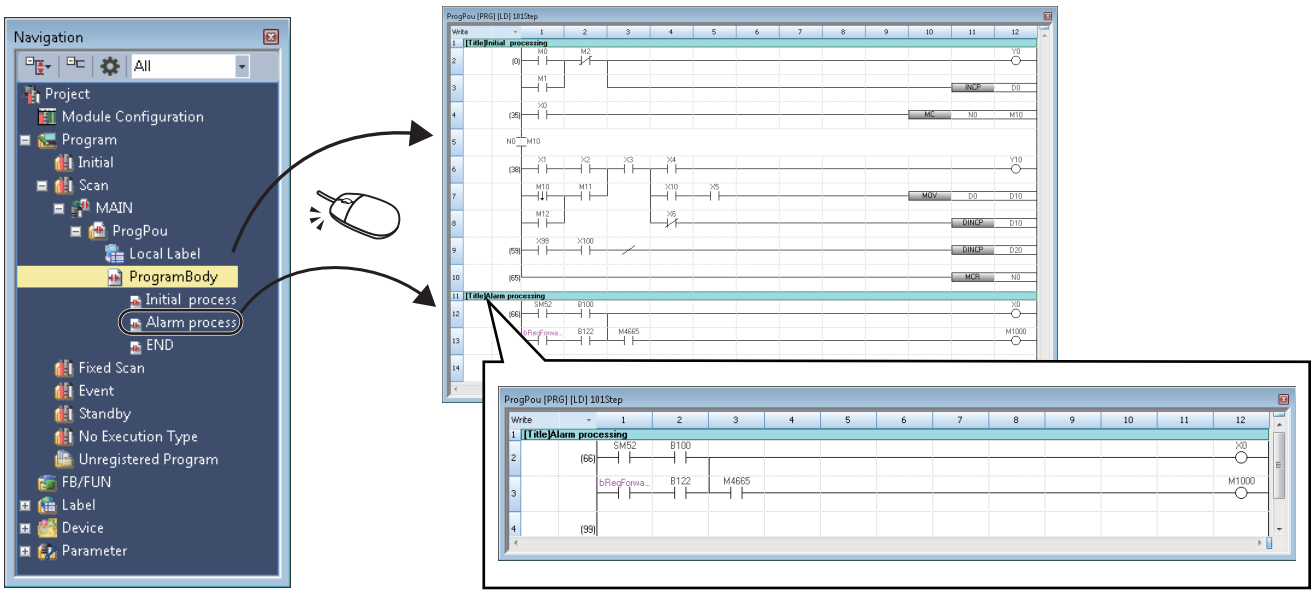


Creating statments

Select [Edit] ⇨ [Documentation] ⇨ [Edit Statment] ( ) and double-click a ladder block where a comment is to be created. The "Input Line Statement" dialog box appears. Enter a comment and click the [OK] button.



- After entering statements is completed, select [Edit] ⇨ [Documentation] ⇨ [Edit Statement] ( ) again.
- When "Display in Navigation Window" is selected, "[Title]" is added in front of the line statement to show a line statement in the Navigation window. By double-clicking a line statement in the Navigation window, only corresponding ladder blocks can be displayed.

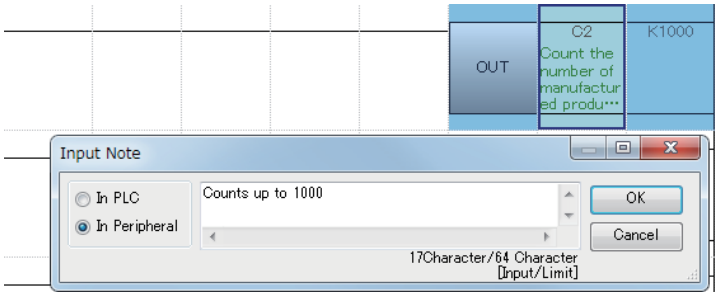


Precautions

When copying or deleting a line statement displayed in the Navigation window, the ladder blocks between the selected line statement and the next statement are copied or deleted.

Creating notes

Select [Edit] ⇨ [Documentation] ⇨ [Edit Note] ( ) and double-click an output or instruction where a comment is to be created. The "Input Note" dialog box appears. Enter a comment and click the [OK] buttton.



- After entering notes is completed, select [Edit] ⇨ [Documentation] ⇨ [Edit Note] ( ) again.

Splitting a ladder editor

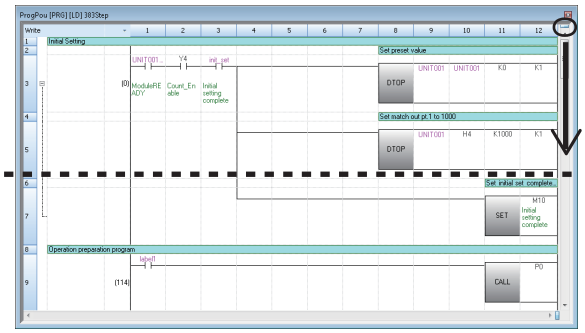
The following show the procedures for splitting a ladder editor.

■ Horizontal sprit

Drag the splitter bar (horizontal) downward or double-click it.

In another way, use the following menu.

- Select [Window] ⇨ [Split].
- Since split ladder editors can be scrolled separately, programs which are described in different parts can be displayed at once.



■ Vertical sprit

Drag the splitter bar (vertical) to the left or double-click it.