MITSUBISHI ELECTRIC

Programmable Controller

MELSEC iQ-F series

MELSEC iQ-F
FX5U Training Maunal

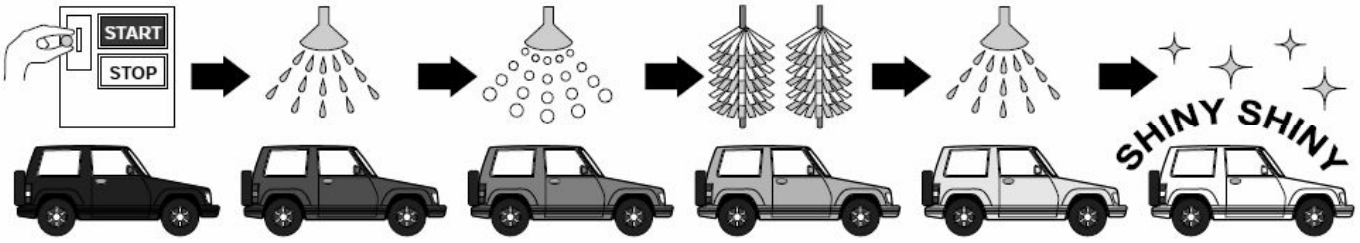
MELFT

Controller : PLC

- PLC = Programmable Logic Controller คืออุปกรณ์ควบคุมที่โปรแกรมได้ การนำ PLC มาประยุกต์ใช้งานจะต้องมีการเขียนโปรแกรม
- ในญี่ปุ่นนิยมเรียก PLC ว่า Sequence Controller หรือ Sequencer แปลว่า อุปกรณ์ควบคุมลำดับการทำงาน ซึ่งเป็นหน้าที่หลักของ PLC ในทุกระบบ และมีการทดสอบฝีมือแรงงาน PLC ตามมาตรฐานของรัฐบาลในชื่อ Electric equipment assembly (sequence control)
- มีการใช้ PLC นอกเหนือจากงาน Sequence control มากขึ้นจึงเรียก PLC ในชื่ออื่น เช่น Programmable Controller (ไม่นิยม), PAC (Programmable Automation Controller) ซึ่งชื่อนี้นิยมเรียก PLC ที่ออกแบบมาร่วมกันทำงานควบคุมได้ทั้งระบบ Automation
- มีการเพิ่มความสามารถด้านการเก็บข้อมูล (Data logging) และส่งข้อมูลไประบบ MES ในคอมพิวเตอร์เพื่อนำไปสู่การปฏิวัติอุตสาหกรรมครั้งที่ 4 หรือ Industry 4.0

MITSUBISHI ELECTRIC
Changes for the Better
for a greener tomorrow

- Sequence Control



Put your money in and press the START button.

The car is first washed with water.

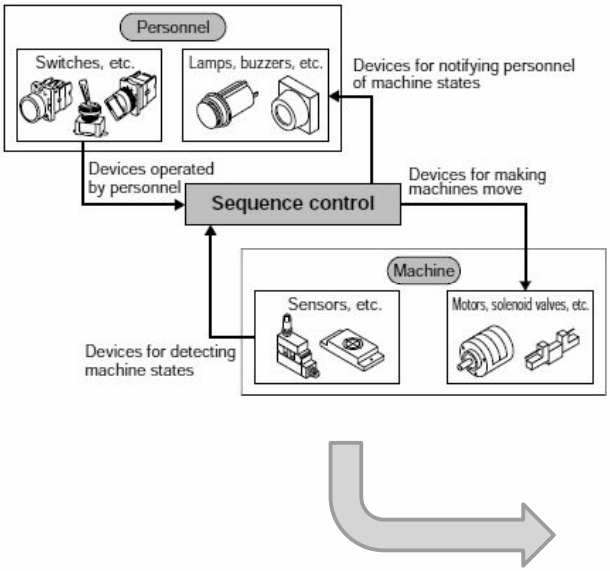
Dirt and mud is then removed from the car with detergent.

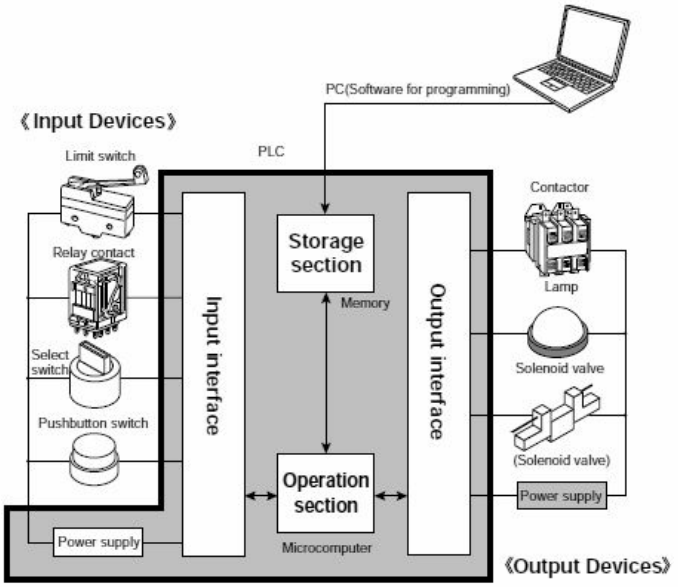
Brushed with a big rotating brush.

And then washed again with water.

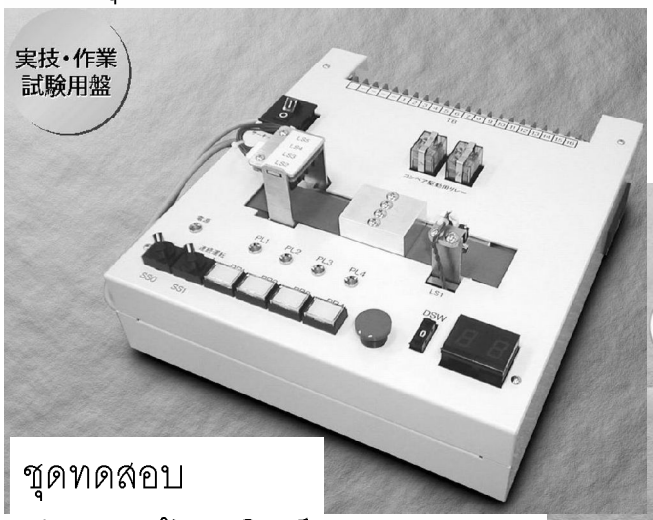
Water is brushed off the car and the car wash is complete.

- Sequence Control ⇨ PLC

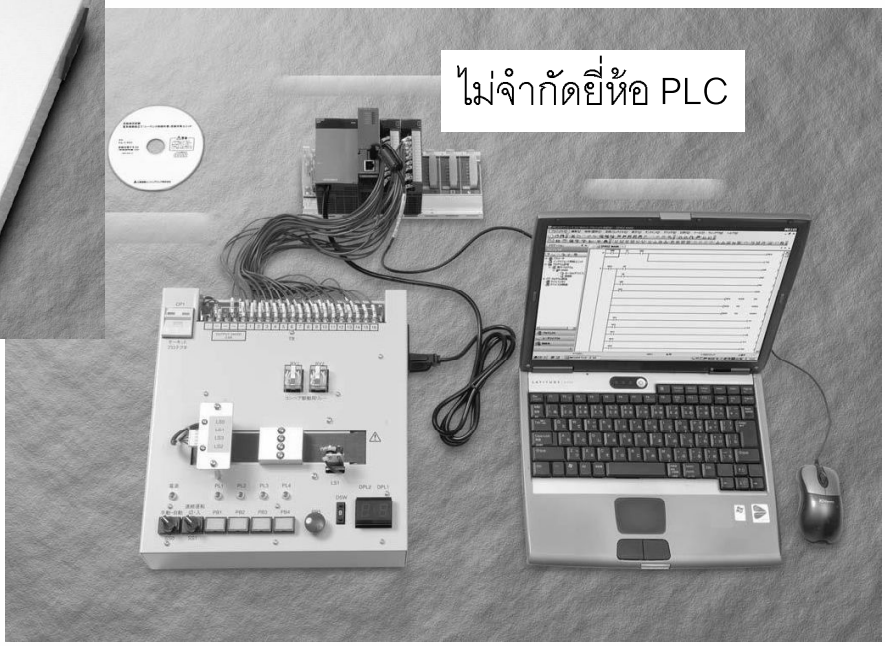


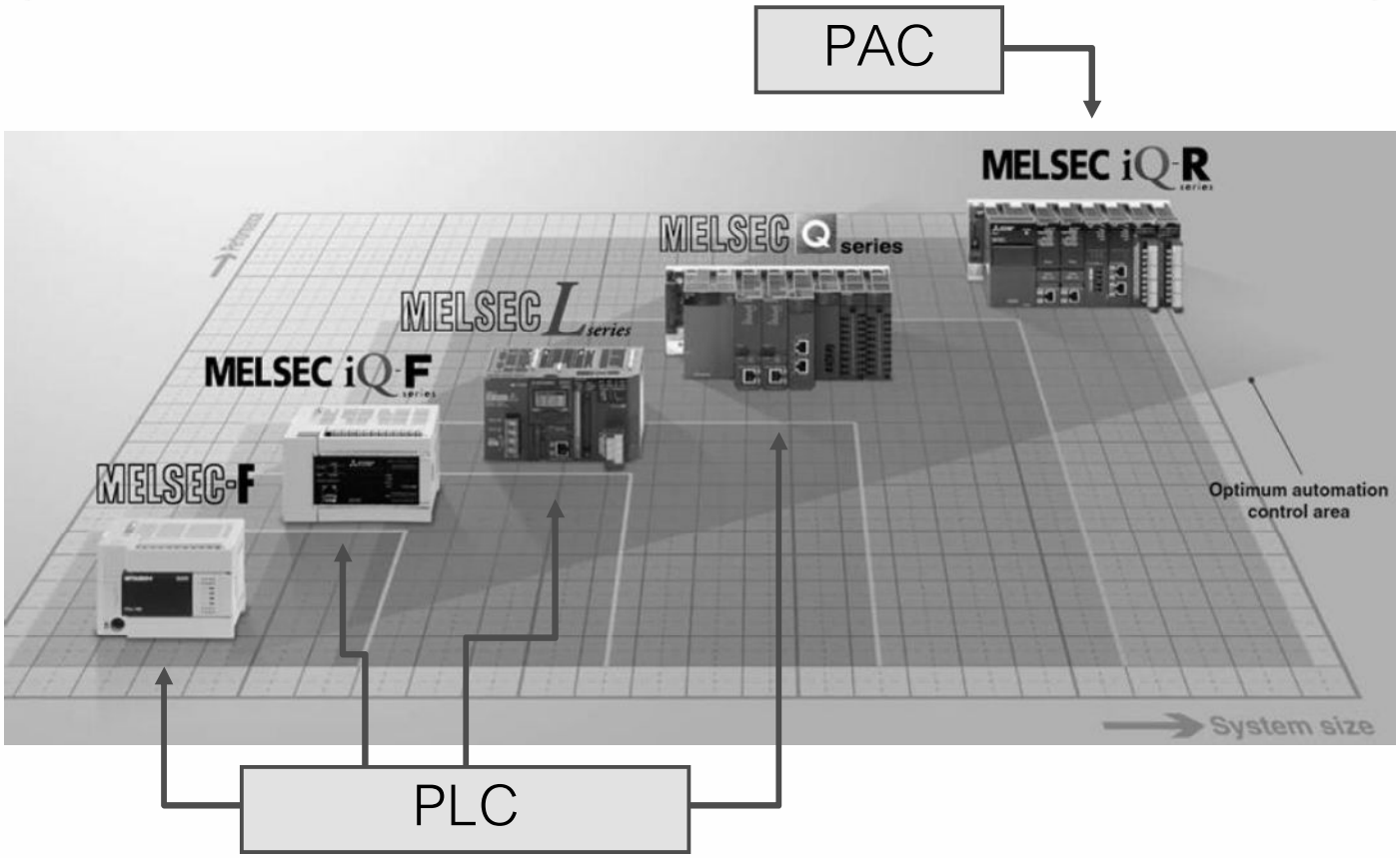


- ชุดทดสอบฝีมือแรงงาน Electric equipment assembly (sequence control)



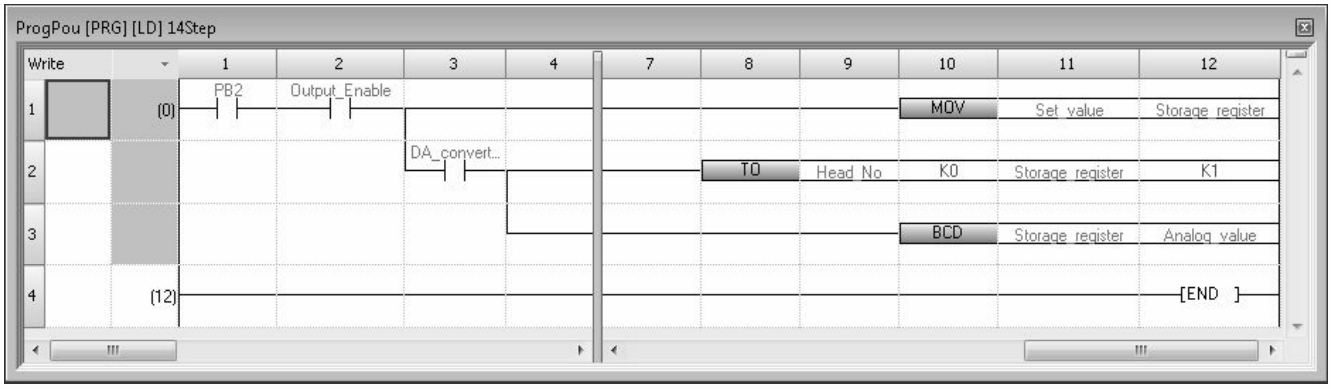
ชุดทดสอบประกอบด้วยสวิตช์ หลอดไฟ สายพาน และชิ้นงาน





ภาษาที่ใช้ในการเขียนโปรแกรม PLC (1)

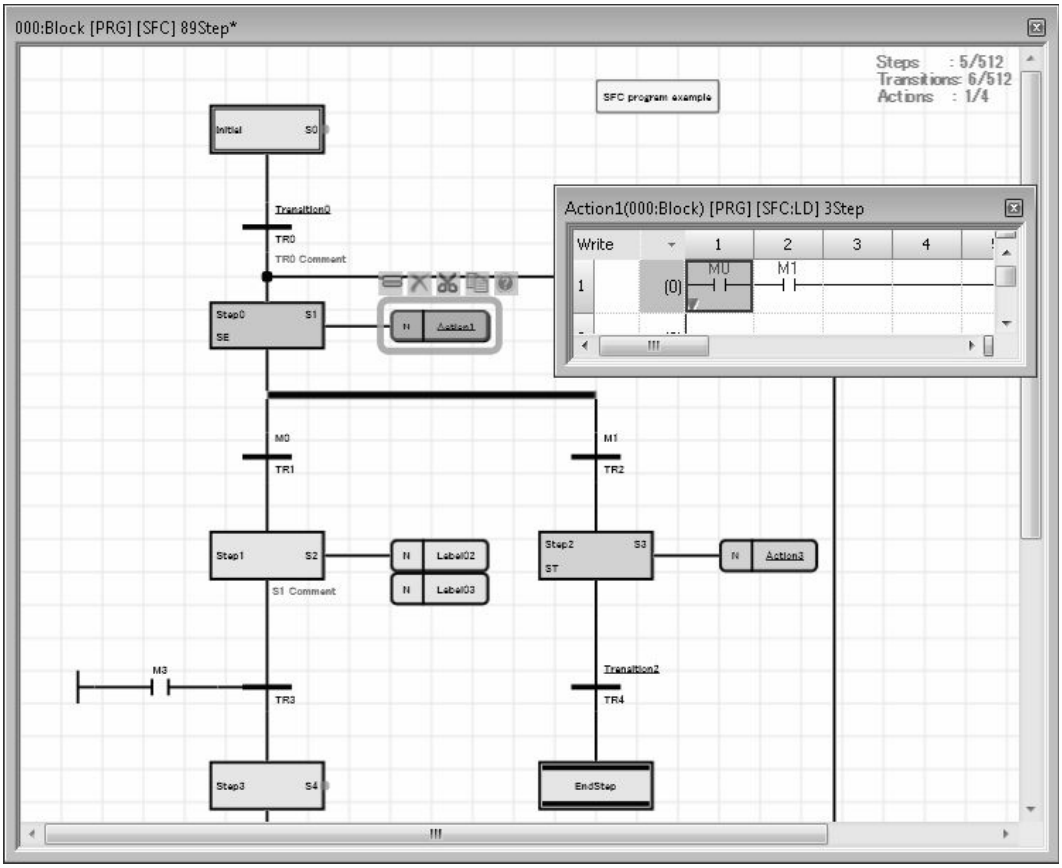
Ladder Diagram (LD)



- นิยมใช้มากที่สุด
- เป็นรูปแบบตามวงจรรีเลย์ที่นิยมใช้ในระบบควบคุมอัตโนมัติยุคแรก ๆ
- ใช้งาน Sequence control ที่ไม่ซับซ้อนได้ดี มีคำสั่งเพิ่มสำหรับการคำนวณและการใช้โมดูลพิเศษเพื่องานอื่น ๆ
- ต้องใช้ทักษะเฉพาะตัวมากในงานที่ซับซ้อน

ภาษาที่ใช้ในการเขียนโปรแกรม PLC (2)

SFC



MITSUBISHI ELECTRIC Changes for the Better

ภาษาที่ใช้ในการเขียนโปรแกรม PLC (2)

for a greener tomorrow

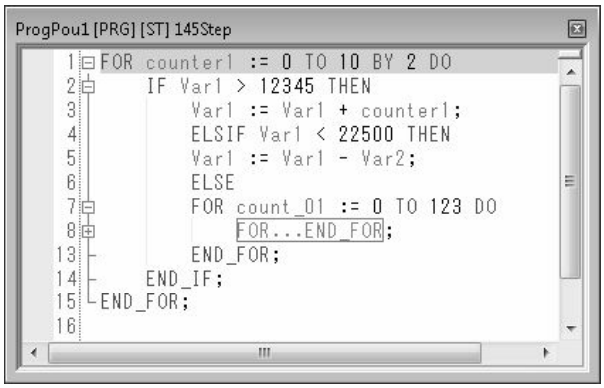
SFC (Sequential Function Chart)

- เขียนเหมือนอธิบายการทำงานของ Sequence มากที่สุด
- ร่วมกับ Ladder diagram ง่าย ๆ ในการลงรายละเอียดของงาน
- ใช้งาน Sequence control ที่ซับซ้อนได้ดีกว่าการเขียนแบบอื่น
- ไม่นิยมใช้มากนัก
- ในญี่ปุ่นมีการสอนภาษานี้ให้ผู้ต้องการทดสอบฝีมือแรงงาน PLC : Electric equipment assembly (sequence control)

ภาษาที่ใช้ในการเขียนโปรแกรม PLC (3)

ST (Structured Text)

- เขียนเหมือนภาษาคอมพิวเตอร์หลายภาษารวมกัน ทำความเข้าใจไม่ยาก
- เหมาะกับการเขียนโปรแกรมคำนวณ
- ไม่เหมาะกับงาน Sequence control
- PLC บางรุ่นใช้ภาษา ST เขียนร่วมกับ Ladder ได้ เรียกว่า Inline structured text
- น่าจะนิยมใช้มากขึ้น

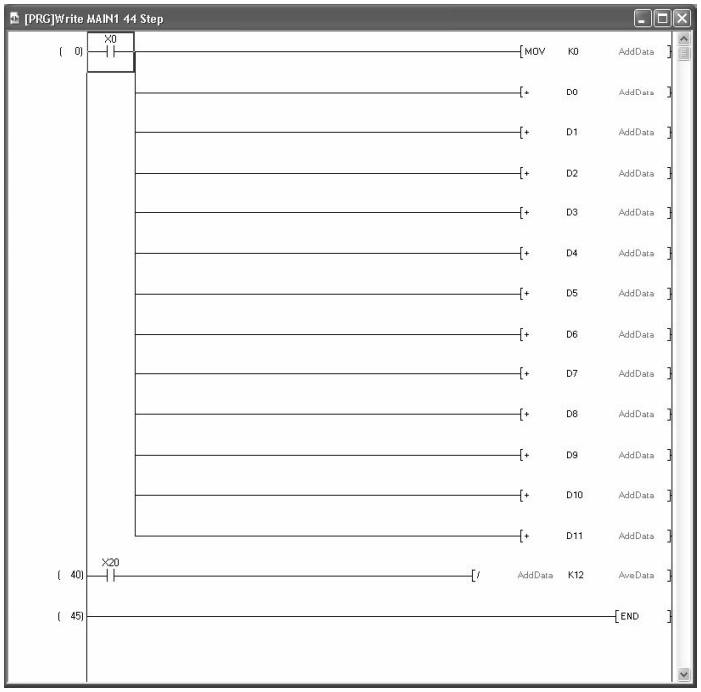


MITSUBISHI ELECTRIC Changes for the Better

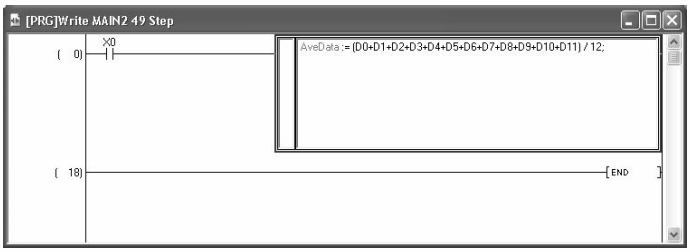
ภาษาที่ใช้ในการเขียนโปรแกรม PLC (3)

for a greener tomorrow

Ladder diagram



Inline structured text

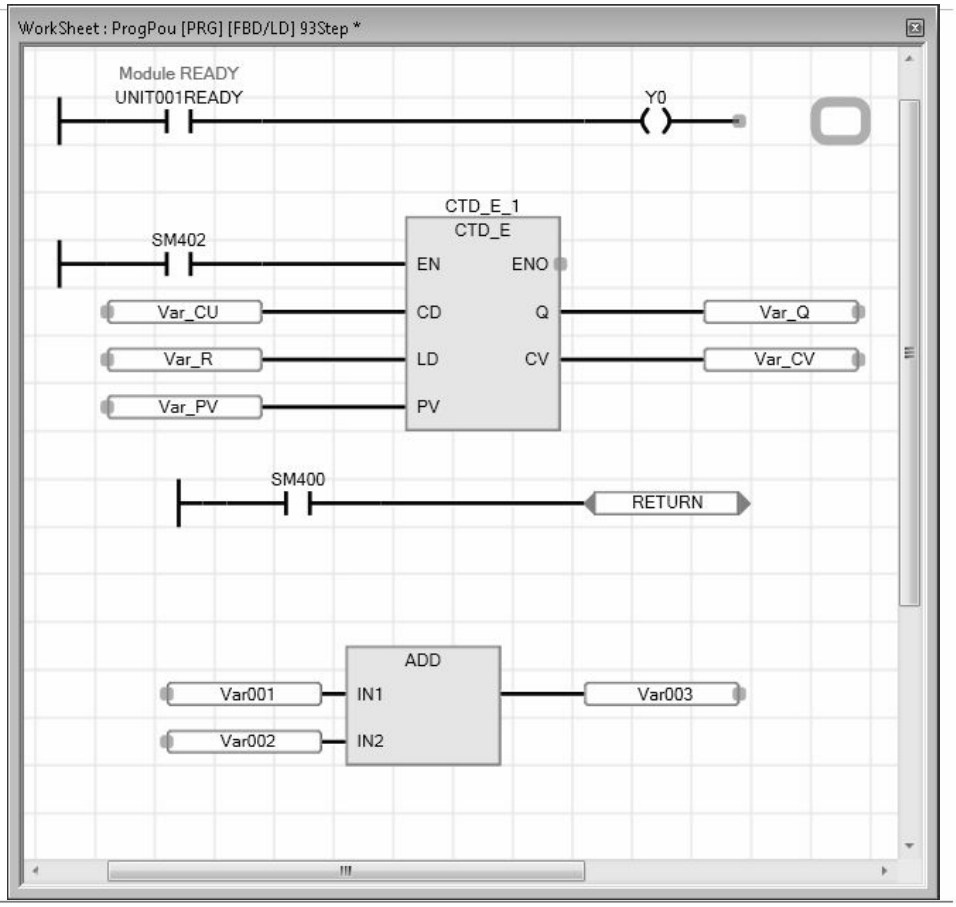


COPYRIGHT © 2016 MITSUBISHI ELECTRIC FACTORY AUTOMATION (THAILAND) ALL RIGHTS RESERVED

ภาษาที่ใช้ในการเขียนโปรแกรม PLC (4)

FBD (Function Block Diagram)

- เขียนเหมือนวงจรอิเล็กทรอนิคส์
- PLC บางรุ่นยอมให้ใช้ร่วมกับ Ladder ได้ (FBD/LD)
- เหมาะกับการเขียนโปรแกรม Process control



MITSUBISHI ELECTRIC Changes for the Better

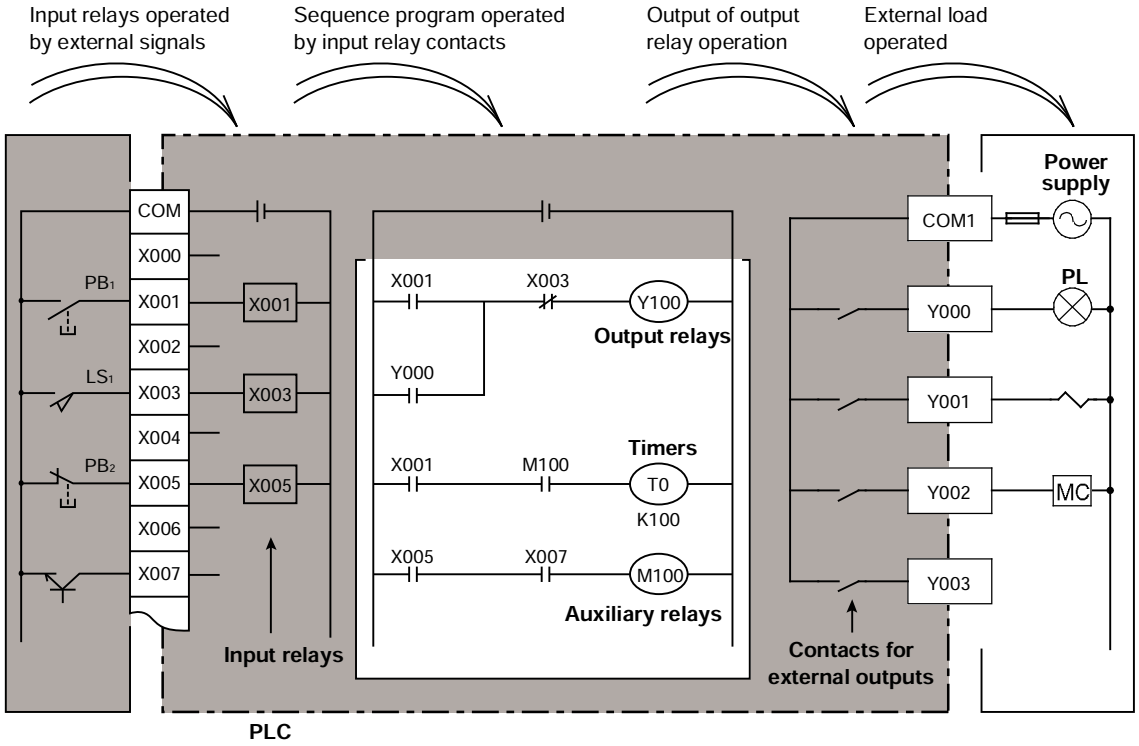
ภาษาที่ใช้ในการเขียนโปรแกรม PLC (4)

for a greener tomorrow

- PLC ที่ออกแบบมาให้ใช้งาน Process control หรือ Process automation โดยตรง เรียกว่า Process CPU จะโปรแกรมด้วย FBD เป็นหลัก คำสั่งที่ใช้ส่วนใหญ่เป็น PID loop control
- งาน Process automation มักเป็นงานผลิตต่อเนื่องไม่ได้ผลิตสินค้าเป็นชิ้น เช่น การกลั่นน้ำมัน โรงงานปิโตรเคมี โรงไฟฟ้า
- Controller ที่นิยมใช้ในงาน Process automation คือ DCS (Distributed Control System) ซึ่งแปลว่าระบบที่มี Controller หลายระดับเชื่อมต่อกันเป็น Network โดยอาจมี PLC ปกติหรือ PLC Process CPU ร่วมอยู่ด้วย(มักเลือกใช้ PLC ที่เป็น Redundant CPU ซึ่งจะมี CPU สำรองอยู่แบบ Hot stand-by)

COPYRIGHT © 2016 MITSUBISHI ELECTRIC FACTORY AUTOMATION (THAILAND) ALL RIGHTS RESERVED

Actually consider a PLC as a group of relays and timers

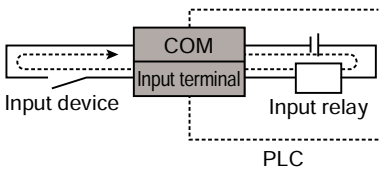


The figure above shows how the input devices, output devices and the sequence program are configured.
The input devices is connected to the PLC's input relays, and the output device is controlled via contacts for external outputs.

● Input relays

The input relays convert the signals from an external device to signals for the PLC. In the above figure, the input device is designed to operate merely by connecting across the input terminal and the COM terminal.
However it should be noted that PLCs have a limited number of contacts, even though, an infinite number of contacts is provided on the sequence program.

Operation



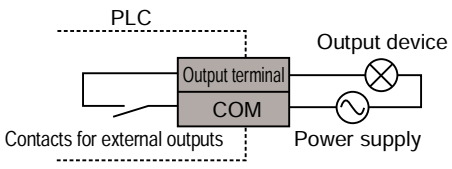
The PLC has a built-in power supply for the input relays. When the contact of the input device is conducting, current flows along the dotted line to drive the input relay.

Regardless of whether the external input device is a N.O. contact or a N.C. contact, the contact on the PLC that closes when the COM and input terminals are conducting is an N.O. contact, and the contact that opens is a N.C. contact.

● Output contacts

Output contacts are otherwise known as just outputs. These are coutrolled by the sequence program, and are necessary for driving an external load.
Devices having a different power supply (AC or DC) can be connected to the contacts for external output.

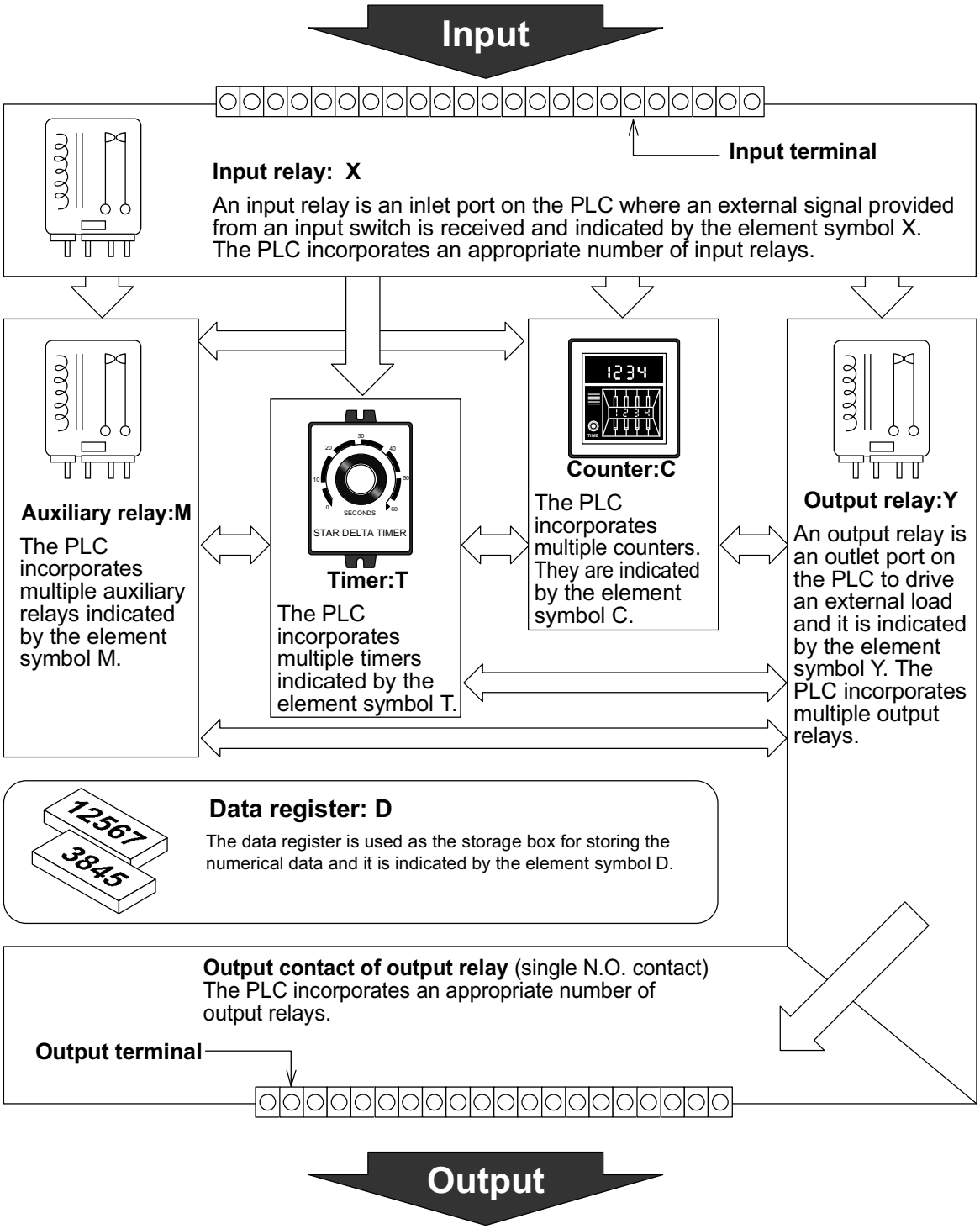
Operation



When the output relays on the sequence program are driven, the contacts for external output close. Provide a power supply for driving the external device outside the PLC.

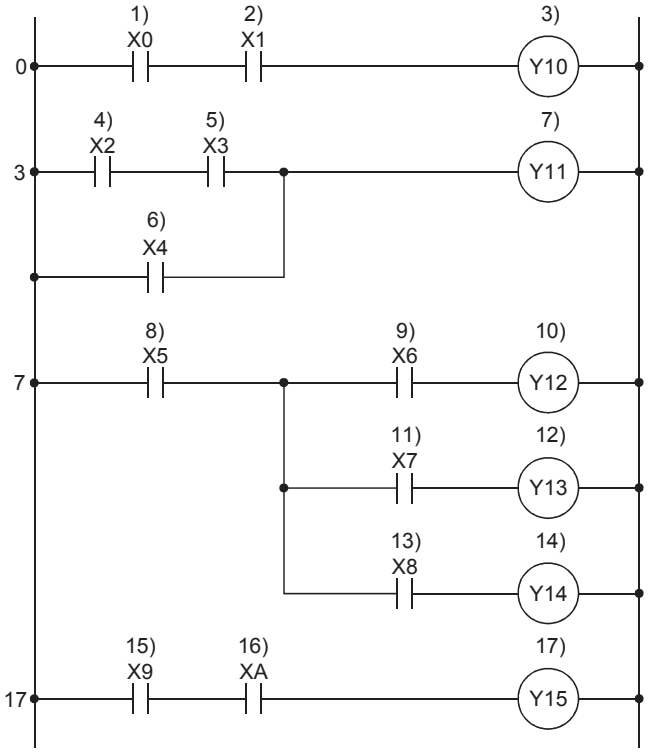
2.2.4 Types of relay and timers

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



Program processing sequence

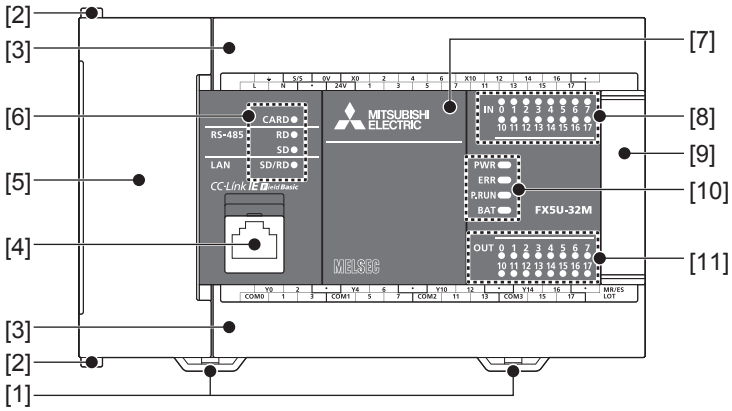
A CPU module executes operations in series from the start step of the program memory from left to right and from top to bottom (in the order of 1), 2) ... and 17)) in units of a ladder block as shown below.



1 OUTLINE

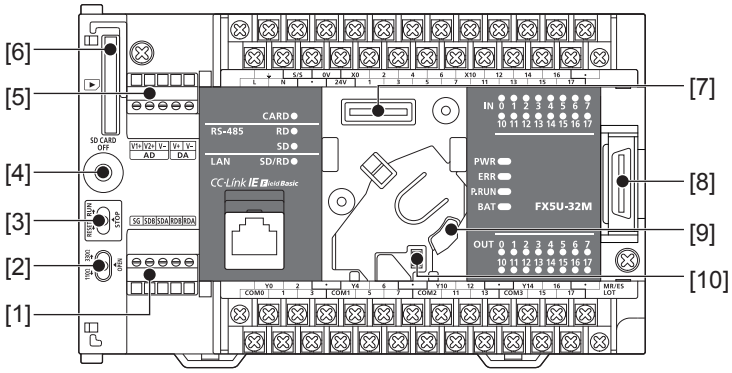
1.1 Part Names

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 (☞ Page 157 Checking with LEDs)
	ERR LED	Indicates the error status of the CPU module. (☞ Page 157 Checking with LEDs) 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 (☞ Page 157 Checking with LEDs)
[11]	Output display LED	Lit when output is on.

With cover open

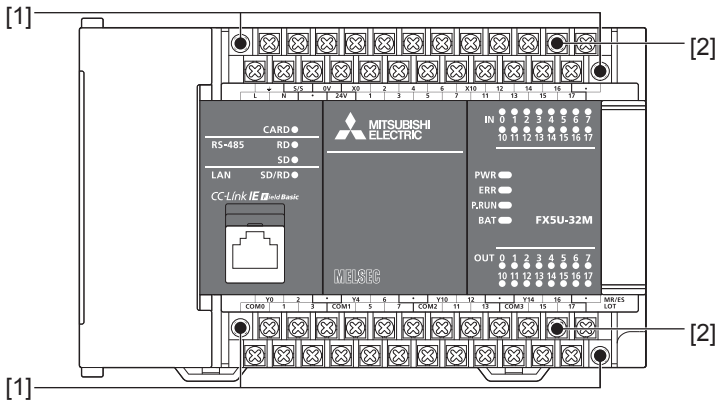


No.	Name	Description
[1]	Built-in RS-485 communication terminal block	Terminal block for connection with RS-485-compatible devices
[2]	RS-485 terminal resistor selector switch	Switch for switching terminal resistance for built-in RS-485 communication.
[3]	RUN/STOP/RESET switch	Switch for operating the CPU module. (☞ Page 151 Methods of running, stopping, and resetting) 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. For details on the terminal layout, refer to ☞ Page 34 Terminal Layout.

2.2 Power Supply Specifications

The CPU module power supply specifications are explained below.

For the consumption current of extension modules, refer to ☞ Page 37 PRODUCT LIST or manuals of each extension module.

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

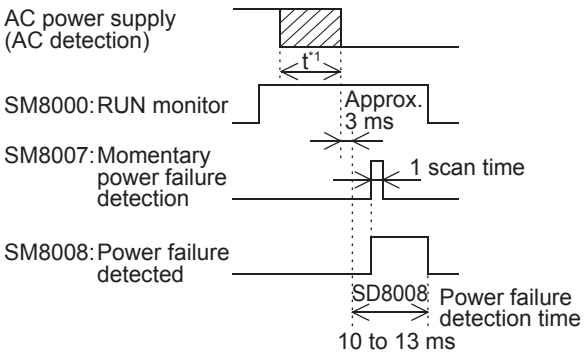
*1 This item shows value when all 24 V DC service power supplies are used in the maximum configuration connectable to the CPU module. (The current of the input circuit is included.)

*2 When I/O modules are connected, they consume current from the 24 V DC service power.
For details on the service power supply, refer to ☞ Page 57 Limitation on Current Consumption.

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

Point

The following shows the power failure detection time of the AC power supply type.



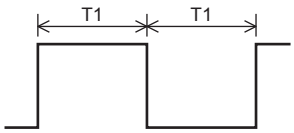
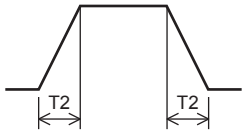
*1 t: time (as a guideline)
100 V AC: 0 ms to approximately 60 ms
200 V AC: 0 ms to approximately 100 ms

2.3 Input Specifications

The CPU module input specifications are explained below.

24 V DC Input (sink/source)

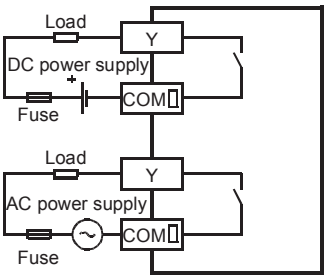
The input points in the table below indicate the CPU module terminal points.

Item			Specifications	
No. of input points	FX5U-32M□		16 points	
	FX5U-64M□		32 points	
	FX5U-80M□		40 points	
Connection type			Removable terminal block (M3 screws)	
Input type			Sink/source	
Input signal voltage			24 V DC +20 %, -15%	
Input signal current	X0 to X17		5.3 mA/24 V DC	
	X20 and subsequent		4.0 mA/24 V DC	
Input impedance	X0 to X17		4.3 kΩ	
	X20 and subsequent		5.6 kΩ	
ON input sensitivity current	X0 to X17		3.5 mA or more	
	X20 and subsequent		3.0 mA or more	
OFF input sensitivity current			1.5 mA or less	
Input response frequency	FX5U-32M□	X0 to X5	200 kHz When capturing pulses of a response frequency of 50 to 200 kHz, refer to ☞ Page 112 In the case of capturing high-speed pulses.	
	FX5U-64M□, FX5U-80M□	X0 to X7		
	FX5U-32M□	X6 to X17	10 kHz	
	FX5U-64M□, FX5U-80M□	X10 to X17		
	FX5U-64M□, FX5U-80M□	X20 and subsequent	0.1±0.05 kHz	
Pulse waveform	Waveform			
			T1 (pulse width)	T2 (rise/fall time)
	FX5U-32M□	X0 to X5	2.5 μs or more	1.25 μs or less
	FX5U-64M□, FX5U-80M□	X0 to X7		
	FX5U-32M□	X6 to X17	50 μs or more	25 μs or less
	FX5U-64M□, FX5U-80M□	X10 to X17		
Input response time (H/W filter delay)	FX5U-32M□	X0 to X5	ON: 2.5 μs or less OFF: 2.5 μs or less	
	FX5U-64M□, FX5U-80M□	X0 to X7		
	FX5U-32M□	X6 to X17	ON: 30 μs or less OFF: 50 μs or less	
	FX5U-64M□, FX5U-80M□	X10 to X17		
	FX5U-64M□, FX5U-80M□	X20 and subsequent	ON: 50 μs or less OFF: 150 μs or less	
Input response time (Digital filter setting value)			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.	
Input signal format			No-voltage contact input Sink: NPN open collector transistor Source: PNP open collector transistor	
Input circuit insulation			Photo-coupler insulation	
Indication of input operation			LED is lit when input is on	

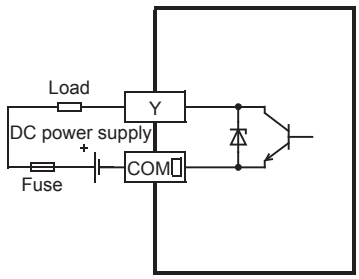
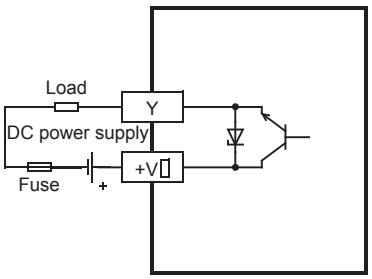
2.4 Output Specifications

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 For details on the common, refer to ☞ Page 29 Interpretation of terminal block layout.
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□].

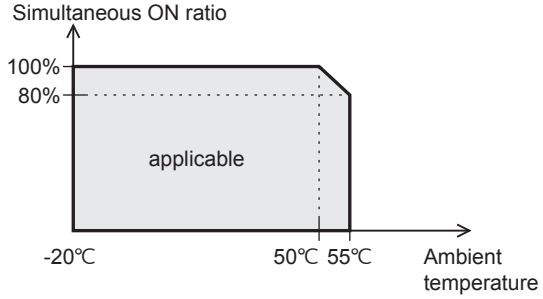
Transistor output

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 For details on the common, refer to ☞ Page 35 Interpretation of terminal block layout.
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□].

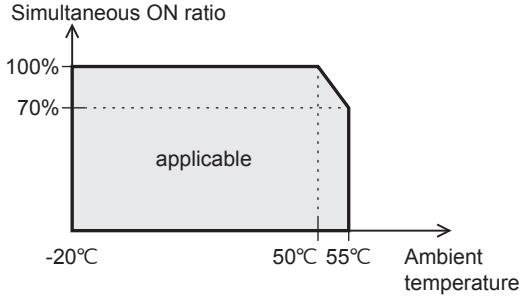
2.5 Input/Output Derating Curve

The derating curve below shows the simultaneous ON ratio of available PLC inputs or outputs with respect to the ambient temperature. Use the PLC within the simultaneous input or output ON ratio range shown in the figure.

- AC power supply type
  Derating curve
  When service power supply or external power supply (24 V DC) is used for input circuits



- DC power supply type
  Derating curve
  When external power supply (24 V DC) is used for input circuits



2.6 Performance Specifications

Item		Specification
Control system		Stored-program repetitive operation
Input/output control system		Refresh system (Direct access input/output allowed by specification of direct access input/output [DX, DY])
Programming specifications	Programming language	Ladder diagram (LD), structured text (ST), function block diagram/ladder diagram (FBD/LD)
	Programming extension function	Function block (FB), function (FUN), label programming (local/global)
	Constant scan	0.2 to 2000 ms (can be set in 0.1 ms increments)
	Fixed cycle interrupt	1 to 60000 ms (can be set in 1 ms increments)
	Timer performance specifications	100 ms, 10 ms, 1 ms
	No. of program executions	32
	No. of FB files	16 (Up to 15 for user)
Operation specifications	Execution type	Standby type, initial execution type, scan execution type, event execution type
	Interrupt type	Internal timer interrupt, input interruption, high-speed comparison match interrupt, interrupt by modules*1
Command processing time	LD X0	34 ns
	MOV D0 D1	34 ns
Memory capacity	Program capacity	64 k steps (128 kbytes, flash memory)
	SD memory card	Memory card capacity (SD/SDHC memory card: Max. 16 Gbytes)
	Device/label memory	120 kbytes
	Data memory/standard ROM	5 Mbytes
Flash memory (Flash ROM) write count		Maximum 20000 times
File storage capacity	Device/label memory	1
	Data memory P: No. of program files FB: No. of FB files	P: 32, FB: 16
	SD Memory Card	NZ1MEM-2GBSD: 511*2
		NZ1MEM-4GBSD, NZ1MEM-8GBSD, NZ1MEM-16GBSD: 65534*2
Clock function	Display data	Year, month, day, hour, minute, second, day of week (leap year automatic detection)
	Precision	Differences per month ±45 sec./25℃ (TYP)
Power failure retention (clock data*3)	Retention method	Large-capacity capacitor
	Retention time	10 days (Ambient temperature: 25℃)
No. of input/output points	(1) No. of input/output points	256 points or less
	(2) No. of remote I/O points	384 points or less
	Total No. of points of (1) and (2)	512 points or less
Power failure retention (device)	Power failure retention capacity	Maximum 12 k word*4

*1 Interrupt from the intelligent function module and high-speed pulse input/output module.

*2 The value listed above indicates the number of files stored in the root folder.

*3 Clock data is retained using the power accumulated in a large-capacity capacitor incorporated into the PLC. When voltage of the large-capacity capacitor drops, clock data is no longer accurately retained. The retention period of a fully charged capacitor (electricity is conducted across the PLC for at least 30 minutes) is 10 days (ambient temperature: 25℃). How long the capacitor can hold the data depends on the operating ambient temperature. When the operating ambient temperature is high, the holding period is short.

*4 It is possible to back up all devices in device (high-speed) area. When a battery is used, it is also possible to back up devices in device (standard) area.

Number of device points

Item			Base	Max. number of points	
No. of user device points	Input relay (X)		8	1024 points	The total number of X and Y assigned to input/output points is up to 256 points.
	Output relay (Y)		8	1024 points	
	Internal relay (M)		10	32768 points (can be changed with parameter)*1	
	Latch relay (L)		10	32768 points (can be changed with parameter)*1	
	Link relay (B)		16	32768 points (can be changed with parameter)*1	
	Annunciator (F)		10	32768 points (can be changed with parameter)*1	
	Link special relay (SB)		16	32768 points (can be changed with parameter)*1	
	Step relay (S)		10	4096 points (fixed)	
	Timer system	Timer (T)	10	1024 points (can be changed with parameter)*1	
	Accumulation timer system	Accumulation timer (ST)	10	1024 points (can be changed with parameter)*1	
	Counter system	Counter (C)	10	1024 points (can be changed with parameter)*1	
		Long counter (LC)	10	1024 points (can be changed with parameter)*1	
	Data register (D)		10	8000 points (can be changed with parameter)*1	
	Link register (W)		16	32768 points (can be changed with parameter)*1	
	Link special register (SW)		16	32768 points (can be changed with parameter)*1	
No. of system device points	Special relay (SM)		10	10000 points (fixed)	
	Special register (SD)		10	12000 points (fixed)	
Module access device	Intelligent function module device		10	65536 points (designated by U□\G□)	
No. of index register points	Index register (Z)*2		10	24 points	
	Long index register (LZ)*2		10	12 points	
No. of file register points	File register (R)		10	32768 points (can be changed with parameter)*1	
No. of nesting points	Nesting (N)		10	15 points (fixed)	
No. of pointer points	Pointer (P)		10	4096 points	
	Interrupt pointer (I)		10	178 points (fixed)	
Others	Decimal constant (K)	Signed	—	16 bits: -32768 to +32767, 32 bits: -2147483648 to +2147483647	
		Unsigned	—	16 bits: 0 to 65535, 32 bits: 0 to 4294967295	
	Hexadecimal constant (H)		—	16 bits: 0 to FFFF, 32 bits: 0 to FFFFFFFF	
	Real constant (E)	Single precision	—	E-3.40282347+38 to E-1.17549435-38, 0, E1.17549435-38 to E3.40282347+38	
	Character string		—	Shift-JIS code max. 255 single-byte characters (256 including NULL)	

*1 Can be changed with parameters within the capacity range of the CPU built-in memory.

*2 Total of the index register (Z) and long index register (LZ) is maximum 24 words.

2.7 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.

2.8 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 RS-485 communication

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

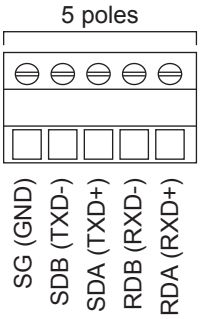
MELSEC iQ-F FX5 User's Manual (Serial Communication)
MELSEC iQ-F FX5 User's Manual (MELSEC Communication Protocol)
MELSEC iQ-F FX5 User's Manual (MODBUS Communication)

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

2.10 Terminal Layout

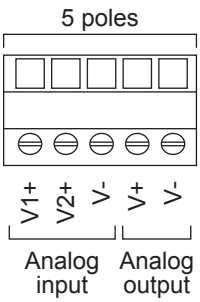
Built-in RS-485 terminal

European type terminal block

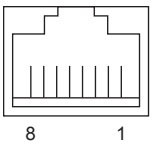


Built-in analog terminal

European type terminal block



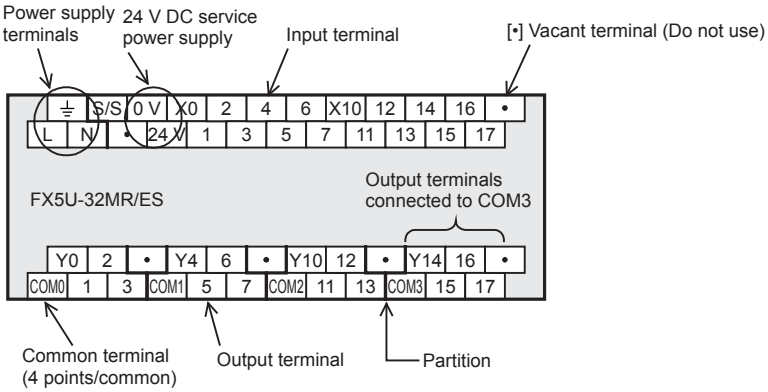
Built-in Ethernet connector



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

Power, input/output terminal block

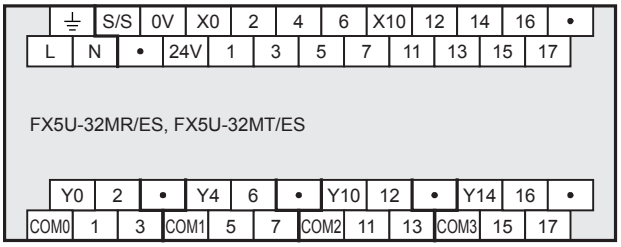
■Interpretation of terminal block layout



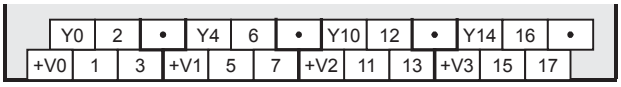
- Indication of power supply terminals
  The AC power supply type has [L] and [N] terminals, whereas the DC power type has [+] and [-] terminals.
  For external wiring, refer to ☞ Page 99 Power Supply Wiring.
- Indication of 24 V DC service power supply
  The AC power supply type has [0V] and [24V] terminals.
  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.
  For external wiring, refer to ☞ Page 107 Input Wiring.
- 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□

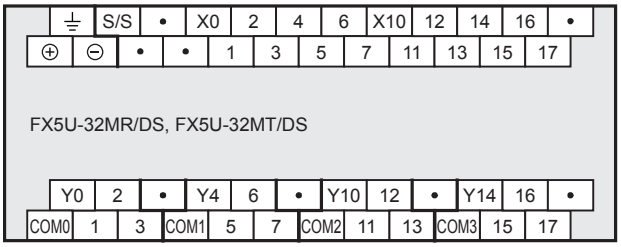
• AC power supply type



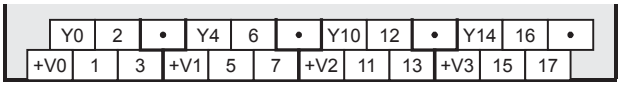
FX5U-32MT/ESS



• DC power supply type

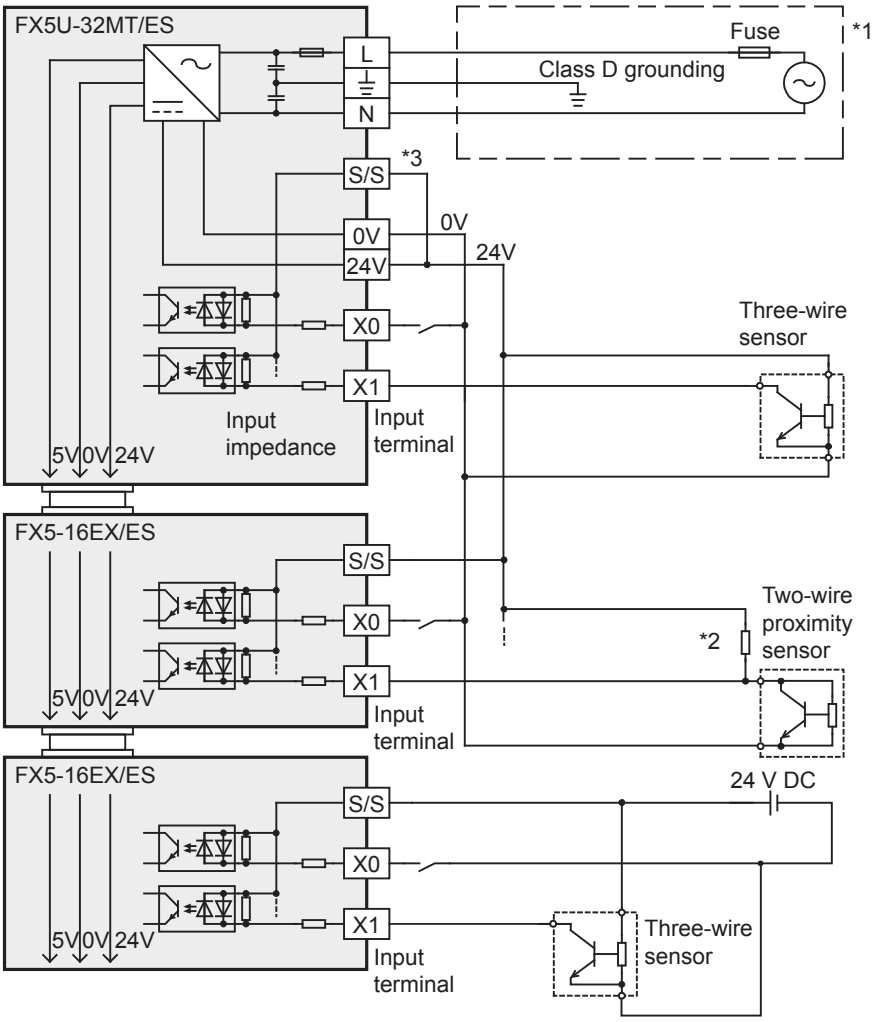


FX5U-32MT/DSS



Input wiring example

Sink input [AC power supply type]



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

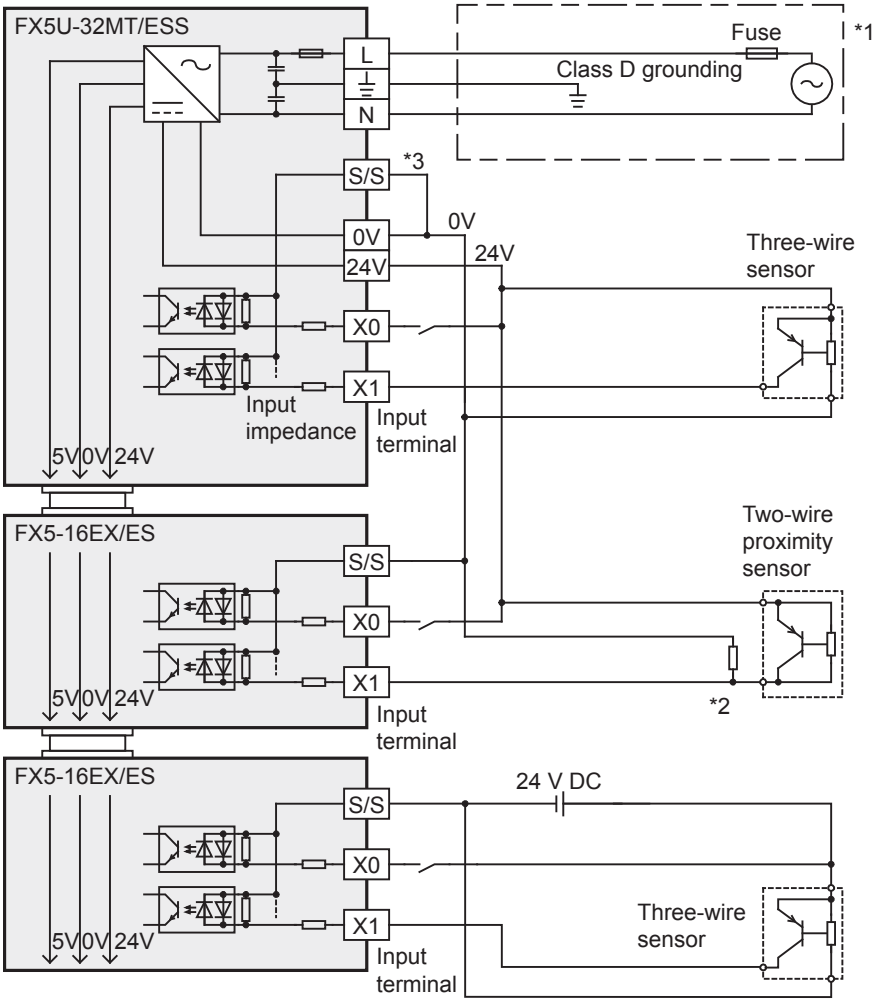
*2 For an input device having a parallel resistance or a two-wire proximity switch, a bleeder resistance may be required.

*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.

6.6 Output Wiring

The output wiring of the CPU module and I/O modules is explained below.

Relay output

For output specifications of the CPU module, refer to ☞ Page 21 Output Specifications.
For output specifications of the I/O modules, refer to ☞ Page 136 Output specifications.

Product life of relay output contacts

The product life of relay output contacts varies considerably depending on the load type used.
Note that loads generating reverse electromotive force or rush current may cause poor contact or welding of contacts which may lead to considerable reduction of the contact product life.

■Inductive load

Inductive loads generate large reverse electromotive force between contacts at shutdown, which may cause arc discharge. At a fixed current consumption, as the power factor (phase between current and voltage) gets smaller, the arc energy gets larger. The standard life of contacts used for inductive AC loads, such as general contactors and solenoid valves, is 500,000 operations at 20 VA.
The following table shows the approximate life of a relay based on the results of an operation life test.

• Test condition: 1 sec. ON/1 sec. OFF

Load capacity		Contact life
20 VA	0.2 A/100 V AC	3,000,000 times
	0.1 A/200 V AC	
35 VA	0.35 A/100 V AC	1,000,000 times
	0.17 A/200 V AC	
80 VA	0.8 A/100 V AC	200,000 times
	0.4 A/200 V AC	

The product life of relay contacts becomes considerably shorter than the above conditions when the rush overcurrent is shut down.
Please refer to the following measures regarding the inductive load.
☞ Page 91 Contact protection circuit for inductive loads
Some types of inductive loads generate rush current 5 to 15 times the stationary current at activation. Make sure that the rush current does not exceed the current corresponding to the maximum specified resistance load.

■Lamp load

Lamp loads generally generate rush current 10 to 15 times the stationary current. Make sure that the rush current does not exceed the current corresponding to the maximum specified resistance load.

■Capacitive load

Capacitive loads can generate rush current 20 to 40 times the stationary current. Make sure that the rush current does not exceed the current corresponding to the maximum specified resistance load.
Capacitive loads such as capacitors may be present in electronic circuit loads including inverters.

■Resistance load

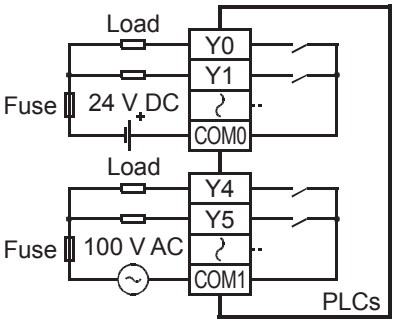
For maximum load specifications of resistance load, refer to the following.
For output specifications of the CPU module, refer to ☞ Page 21 Output Specifications.
For output specifications of the I/O modules, refer to ☞ Page 136 Output specifications.

Handling of relay output

■Output terminal

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,100 V AC and 24 V DC).



■External power supply

Use an external power supply of 30 V DC or less or 240 V AC or less (250 V AC or less when the module does not comply with CE, UL, cUL standards) for loads.

■Insulation of circuit

The PLC internal circuit and external load circuits are electrically insulated between the output relay coil and contact.

The common terminal blocks are separated from one another.

■Display of operation

When power is applied to the output relay coil, the LED is lit, and the output contact is turned on.

■Response time

The response time of the output relay from when the power is applied to the coil until the output contact is turned on and from when the coil is shut off until the output contact is turned off is approx. 10 ms.

■Output current

At a circuit voltage of 240 V AC or less (250 V AC or less when the module does not comply with CE, UL, cUL standards), a resistance load of 2 A per point or an inductive load of 80 VA or less (100 V AC or 200 V AC) can be driven.

For the life of the contact for switching an inductive load, refer to ☞ Page 89 Inductive load.

When an inductive load is switched, connect a diode (for commutation) or a surge absorber in parallel with this load.

DC circuit	AC circuit
Diode (for commutation)	Surge absorber

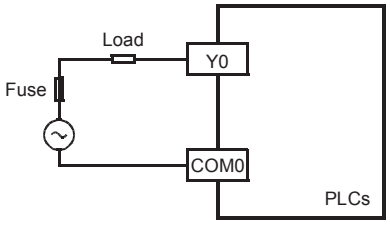
■Open circuit leakage current

There is no leakage current when the outputs are OFF.

Wiring precautions

■Protection circuit for load short-circuiting

A short-circuit at a load connected to an output terminal could cause burnout at the output element or the PCB. To prevent this, a protection fuse should be inserted at the output.



■Contact protection circuit for inductive loads

An internal protection circuit for the relays is not provided for the relay output circuit in this product. It is recommended to use inductive loads with built-in protection circuits.

When using loads without built-in protection circuits, insert an external contact protection circuit, etc. to reduce noise and extend product life.

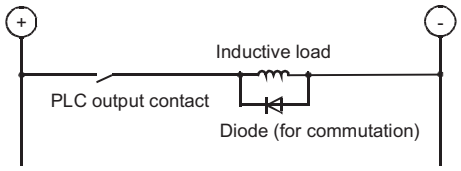
- DC circuit

  Connect a diode in parallel with the load.

  The diode (for commutation) must comply with the following specifications.

  · Reverse voltage: 5 to 10 times the load voltage

  · Forward current: Load current or more

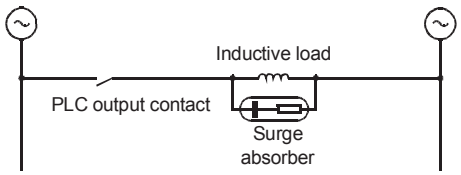


- AC circuit

  Connect a surge absorber (CR composite parts like surge killer, spark killer, etc.) parallel to the load. Select a surge absorber with voltage rating that is suitable for the output used.

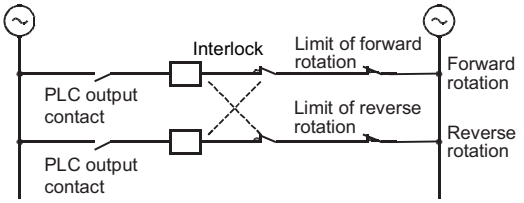
  · Electrostatic capacity: Approx. 0.1 μF

  · Resistance: Approx. 100 to 200 Ω



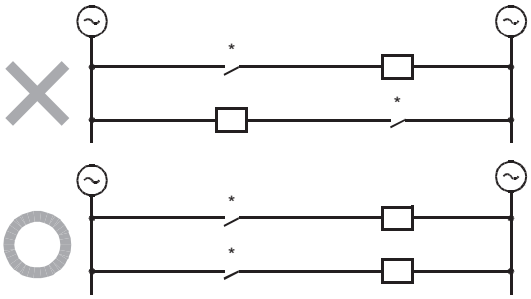
■Interlock

For loads such as forward/reverse contactors, etc., where a hazardous condition could result if switched ON simultaneously, an external interlock should be provided for interlocking along with an interlock in the PLC program, as shown to the right.



■In-phase

PLC output contacts (*) should be used in an "in-phase" manner.



Transistor output

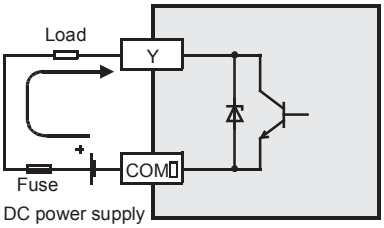
For output specifications of the CPU module, refer to ☞ Page 21 Output Specifications.
For output specifications of the I/O modules, refer to ☞ Page 136 Output specifications.

Sink and source output

Sink output and source output products are available for transistor outputs of the CPU module and I/O modules.

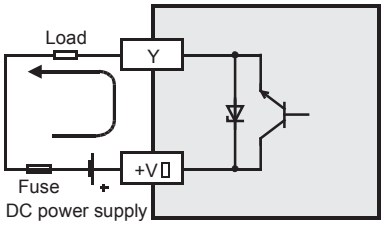
■Differences in circuit

- 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□].

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



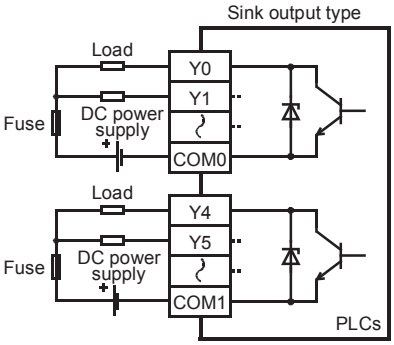
A number is entered in the □ of [+V□].

Handling of transistor output

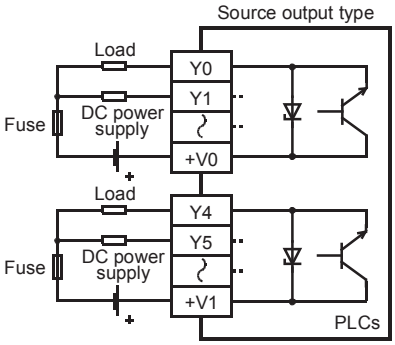
■Output terminal

4 or 8 transistor output points are covered by one common terminal.

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



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



■External power supply

For driving the load, use a smoothing power supply of 5 to 30 V DC that can output current two or more times the current rating of the fuse connected to the load circuit.

■Insulation of circuit

The internal circuit of the PLC and the output transistor are insulated with a photocoupler.
The common blocks are separated from one another.

■Display of operation

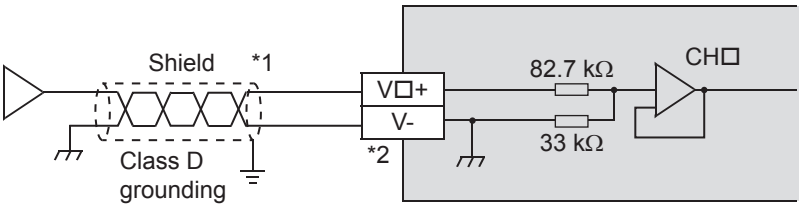
Operation indicator LEDs are built into the CPU module and output modules, and turn ON when photocouplers are activated, and the transistors are switched on.

6.7 Analog Wiring

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

For specifications of the built-in analog I/O terminals of the CPU module, refer to ☞ Page 30 Built-in Analog Specifications.

Analog input wiring



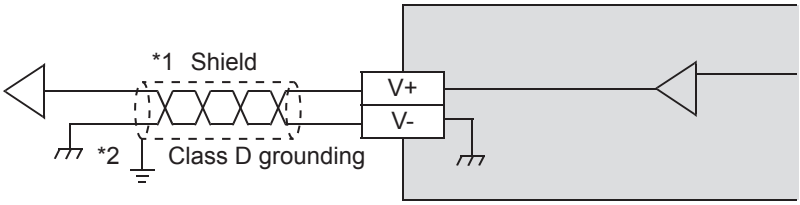
CH No. goes in □ of V□+, CH□.

*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.

When used as a current input, refer to ☞ Page 170 How to use analog inputs built in CPU module for current inputs.

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.

2 OPERATING GX Works3

Point

This chapter describes the basic operations of GX Works3.

GX Works3 is an engineering tool for setting, programming, debugging, and maintenance of projects for the MELSEC iQ-R series programmable controllers and others on Windows®.
Compared with GX Works2, the functionality and operability of GX Works3 have been improved.
For changes in the window display, refer to the following.
MELSEC iQ-R Module Configuration Manual

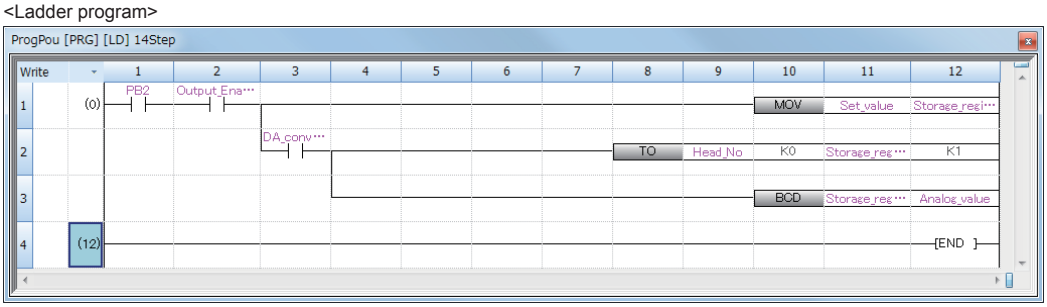
2.1 Main Functions of GX Works3

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

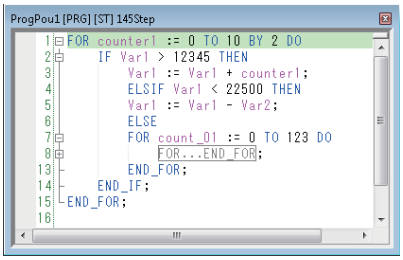
Creating programs

Users can create programs in a desired programming language, such as ladder or ST, depending on the processing.

<Ladder program>

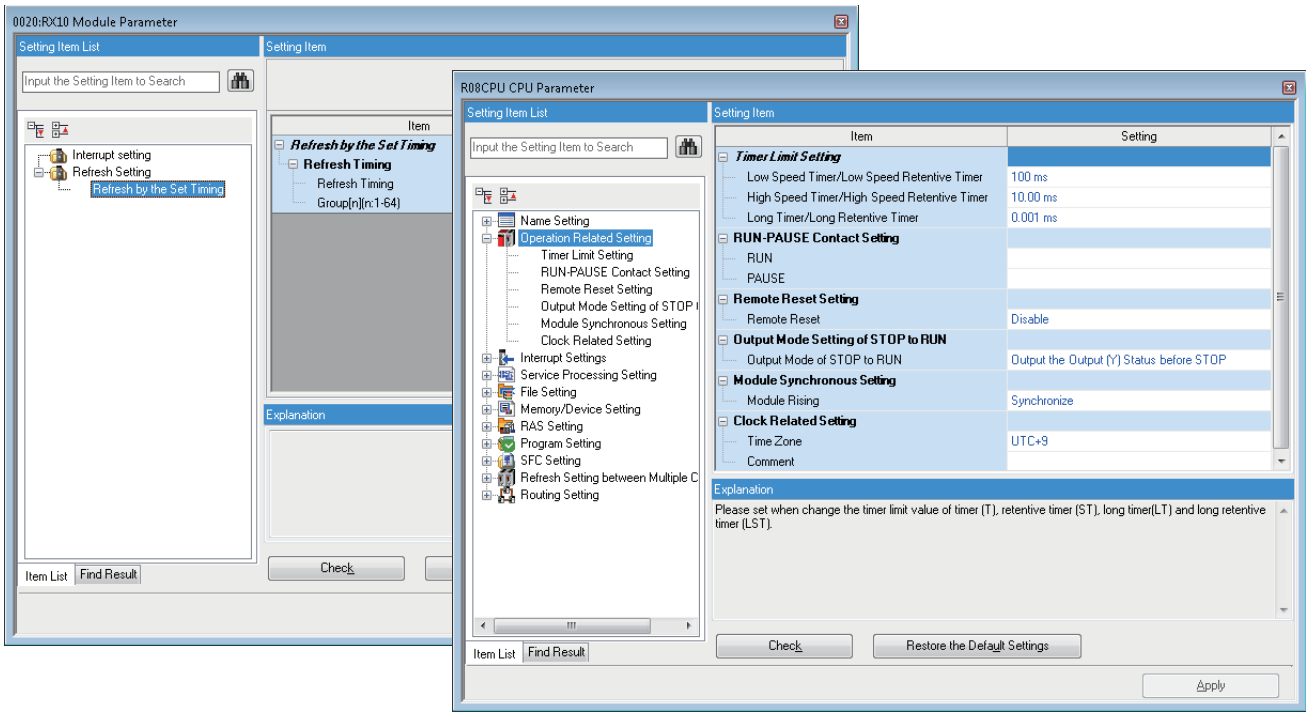


<ST program>



Setting parameters

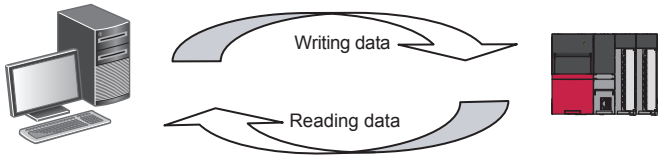
Users can set parameters for CPU modules, I/O modules, and intelligent function modules.



Reading/writing data from/to the CPU module

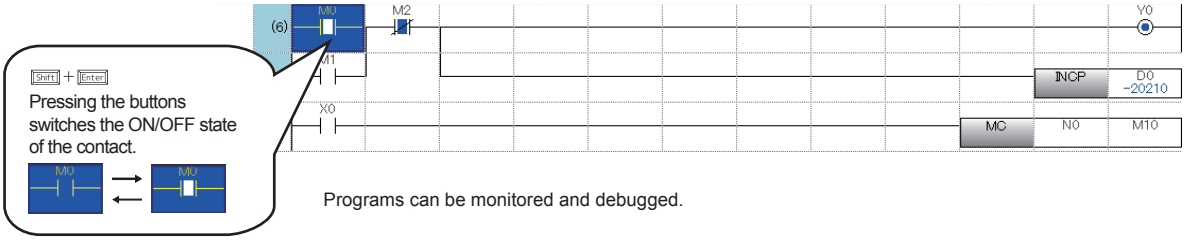
Users can read/write created sequence programs from/to the CPU module by using the "Write to PLC" and "Read from PLC" functions.

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.



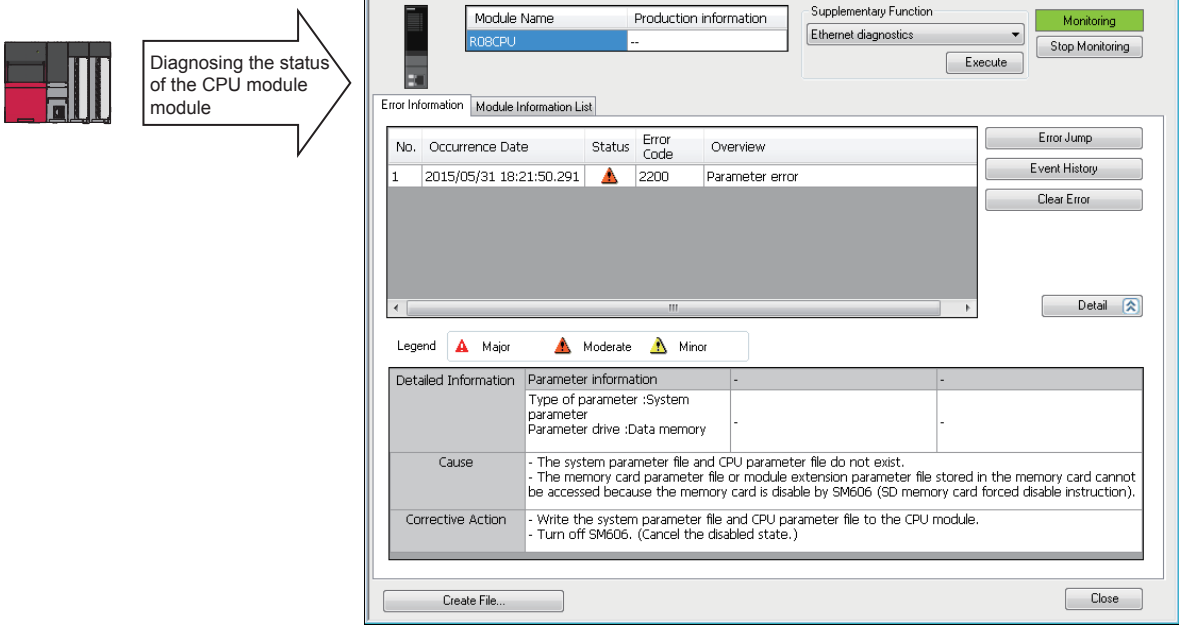
Programs can be monitored and debugged.

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)



2.2 Operations Before Creating a Ladder Program

2.2.1 Starting GX Works3

Operating procedure

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

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

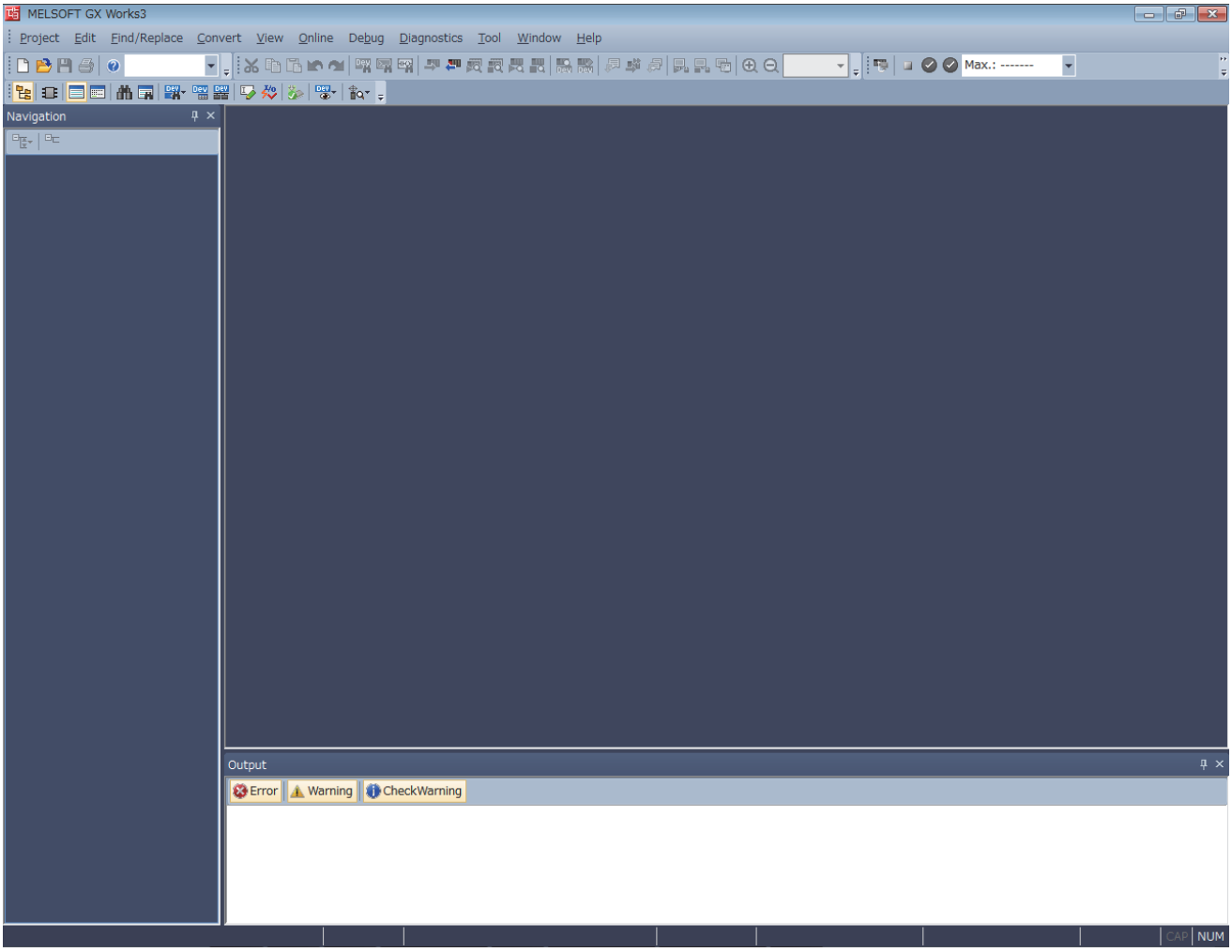


Figure 2.1 Startup window of GX Works3

Precautions

- When mounting or removing the SD memory card while the power is ON, follow the above-mentioned procedure. Otherwise, the data in the card may become corrupted.
- When removing an SD memory card which is being accessed by a function, the CARD LED is turned OFF after the access is completed. When the CARD LED is turned OFF depends on the accessing function.
- When SM605 (memory card interchange protect flag) is ON, the CARD LED is not turned OFF even if the SD memory card stop switch is pressed. If the CARD LED is not turned OFF, turn ON SM606 (memory card disable request) so that the SD memory card cannot be used.

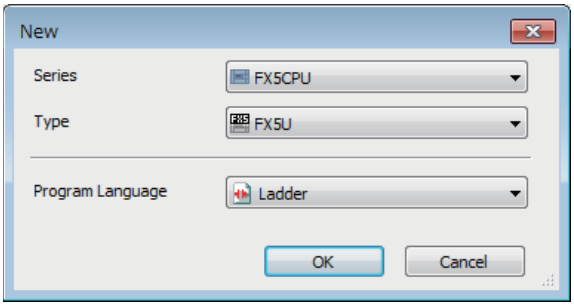
4.3 Creating a Project

Activate the engineering tool and create a project.

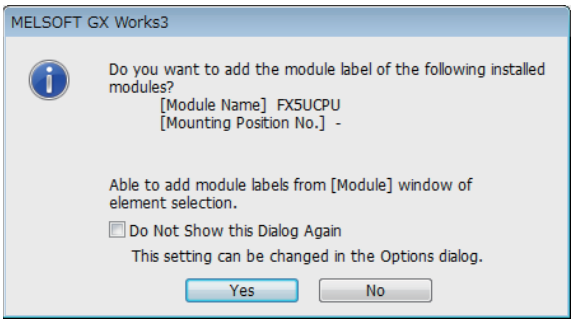
[Project] ⇨ [New]

Procedure

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



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



2. Click the [Yes] button when the window for adding module labels of the CPU module appears.

Point

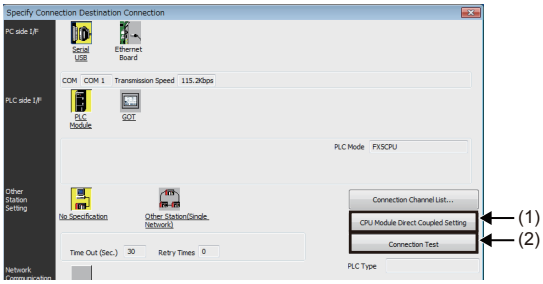
Module label is a label where the I/O signals and buffer memory areas of a module have already been defined. Use of module labels allows programming without being aware of module internal addresses.

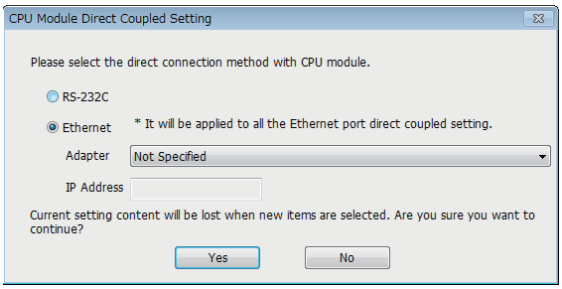
4.4 Connecting a Personal Computer

Connect the CPU module to a personal computer that has an engineering tool installed.
For details on the system configuration examples and GX Works3 settings, refer to the following manual.
MELSEC iQ-F FX5U User's Manual (Hardware)
MELSEC iQ-F FX5UC User's Manual (Hardware)

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 [CPU Module Direct Coupled Setting] button (1) on the "Specify Connection Destination Connection" window.
4. Select [Ethernet] for the connection method for the CPU module and click the [Yes] button.
5. Click the [Connection Test] button (2), and check if the personal computer is connected to the CPU module.

Point

For the FX5 CPU module, the Ethernet adapter on the personal computer side used for the Ethernet port direct connection can be specified.
Select the adapter on the "CPU Module Direct Coupled Setting" window.

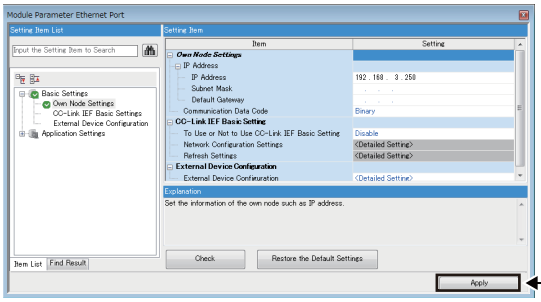
When Ethernet cable is used (connection via a hub)

To connect to Ethernet via a hub, setting of both the personal computer and the CPU module is required.
Connect a personal computer to the CPU module via a hub, following the procedure below.

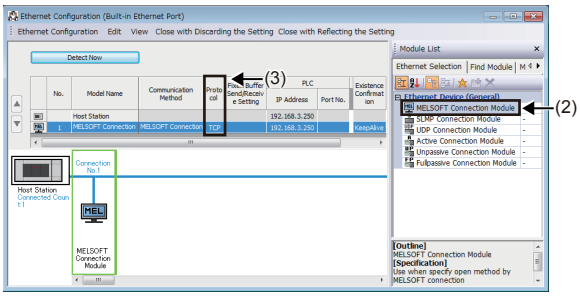
Setting the CPU module

■Setting module parameters

From the engineer tool, set on the “Module Parameter Ethernet Port” screen.
Select [Parameter] ⇨ [FX5UCPU] ⇨ [Module Parameter] ⇨ [Ethernet Port] ⇨ [Basic Settings] ⇨ [Own Node Settings] on the Navigation window.



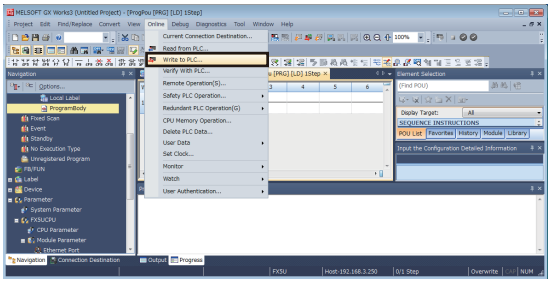
1. Set the IP address on the CPU module side and click the [Apply] button (1).



2. Setting for connection.
Select [Basic Settings] ⇨ [External Device Configuration] ⇨ [Detailed Setting] ⇨ [Ethernet Configuration (Built-in Ethernet Port)] on the “Module Parameter Ethernet Port” screen.

3. Drag and drop "MELSOFT Connection Module" from the "Module List" to the left side of the screen. Select the protocol corresponding to the external device in “Protocol” (3).

■Writing to the CPU module

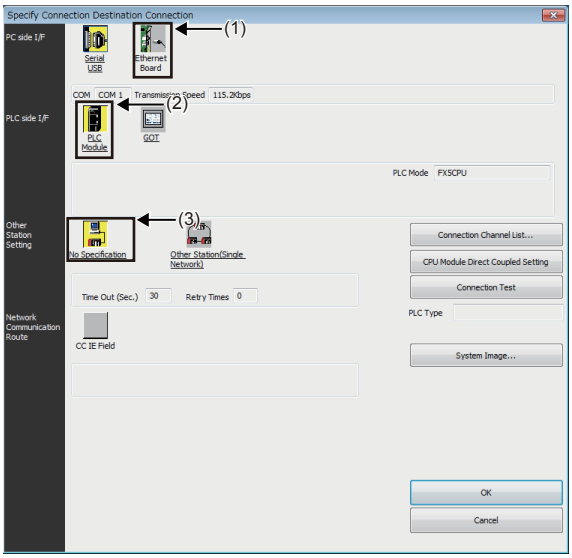


Write the parameters to be set in the CPU module.
Select [Online] ⇨ [Write to PLC] on the menu bar of the engineering tool.
After writing the parameters to the CPU module, power off and on or reset the CPU module to enable the parameters.

Engineering tool settings

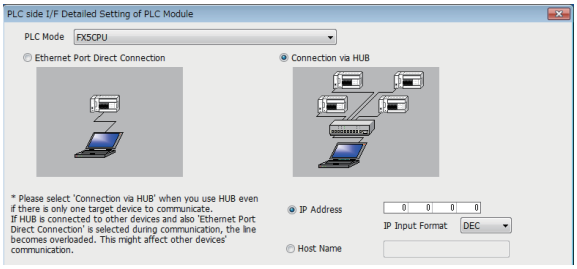
From the engineer tool, this is done on the “Specify Connection Destination Connection” window.

Select “Online” ⇨ [Current Connection Destination] .



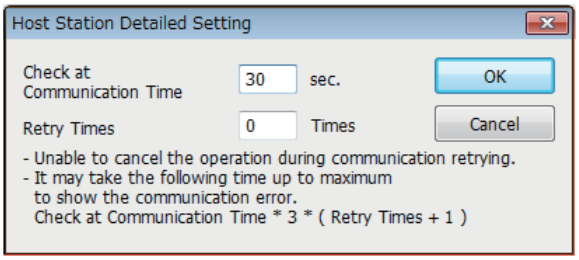
1. Select "Ethernet Board" for "PC side I/F"(1).

2. Select "PLC Module" for "PLC side I/F"(2).



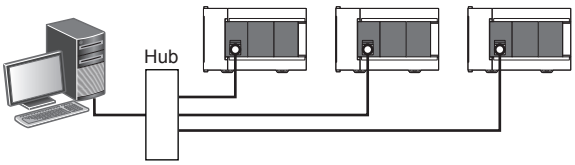
Input the CPU IP address or host name in the “PLC side I/F Detailed Setting of PLC Module” screen as shown in the left figure.

In case of host name, set the name specified in the Microsoft® Windows® hosts file.



3. Double-click "No Specification" (3) on the "Specify Connection Destination Connection" window to set "Other Station Setting". Select an item appropriate to the operating environment.

Connection of the Ethernet cable



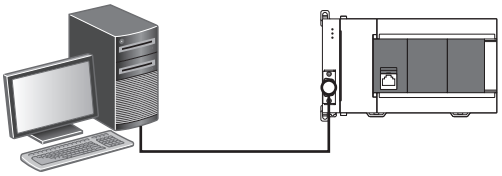
1. Connect a personal computer to the CPU module via a hub.

Point

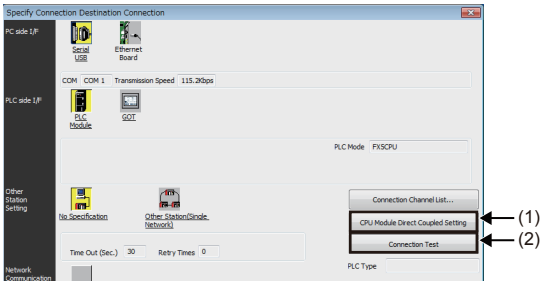
Use the [Find] button in the "PLC side I/F Detailed Setting of PLC Module". The IP address of the connected CPU module can be searched and can be set to “IP Address”.

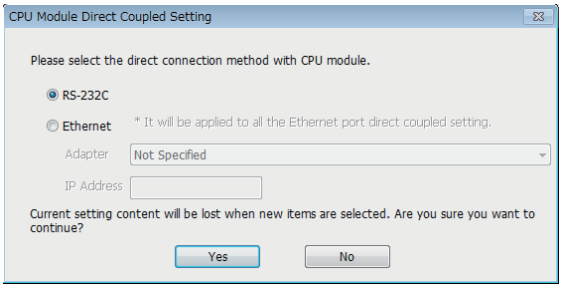
When RS-232C cable is used

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



1. Connect a personal computer to the CPU module using the RS-232C cable.*1
2. Select [Online] ⇨ [Current Connection Destination] on the menubar of the engineering tool.
3. Click the [CPU Module Direct Coupled Setting] button (1) on the "Specify Connection Destination Connection" window.





4. Select [RS-232C] for the connection method, and click the [Yes] button.
5. Click the [Connection Test] button (2), and check if the personal computer is connected to the CPU module.

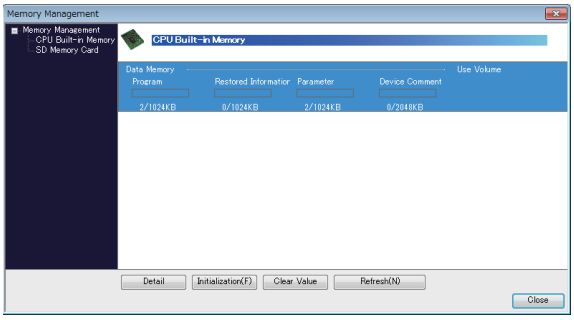
*1 When connecting by RS-232C cable, an expansion board or expansion adapter is required.

4.5 Initializing the CPU Module

Initialize the CPU module.

[Online] ⇨ [CPU Memory Operation]

Procedure

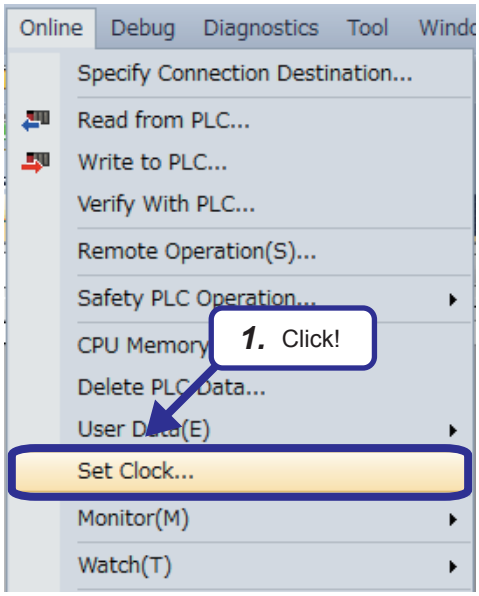


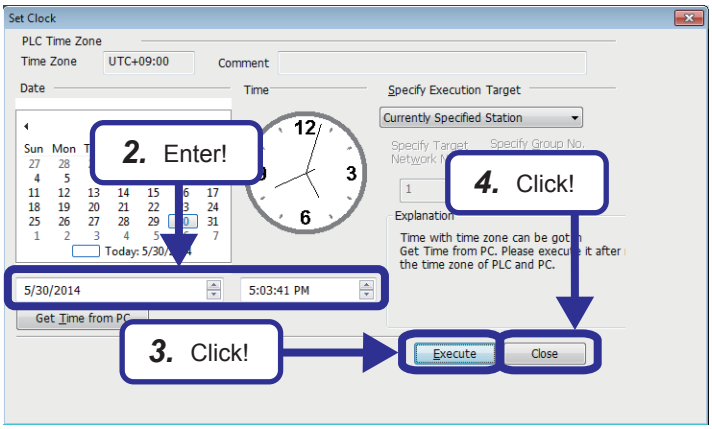
1. Select "Data Memory" on the "Memory Management" window, and click the [Initialization] button.
2. After the initialization processing completes, click the [Close] button.

2.3.7 Setting the clock of the CPU module

The year, month, day, hour, minute, second, and day of week can be set to the clock element of the CPU module.
To use the clock function, use GX Works3 or a program.
Set the clock and read the setting with GX Works3.

Operating procedure

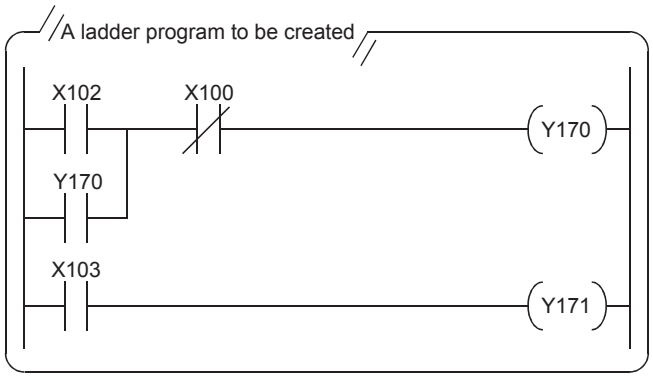




1. Click [Online] → [Set Clock] from the menu to display the "Set Clock" dialog box.
2. Set a year, month, day, hour, minute, second, and day of week on the "Set Clock" dialog box.
3. Click the [Execute] button.
4. Click the [Close] button.

2.4 Creating a Ladder Program

Operating procedure

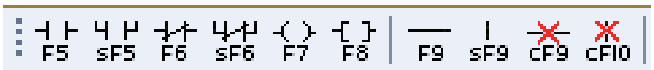


1. This section describes how to create a ladder program such as the one shown on the left.

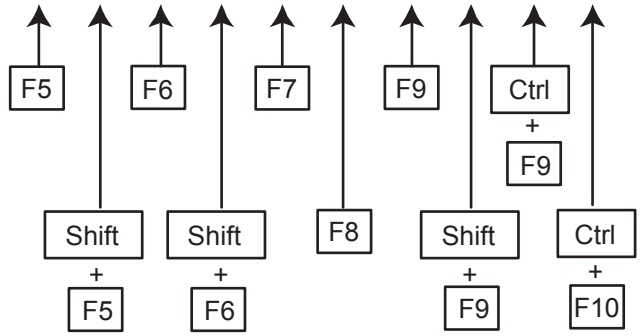
- Use only one-byte characters.
  Two-byte characters cannot be used.

Point

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



Main key operations



s: Shift key
c: Ctrl key
a: Alt key

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

Write Mode (F2) — Monitor Mode (F3)

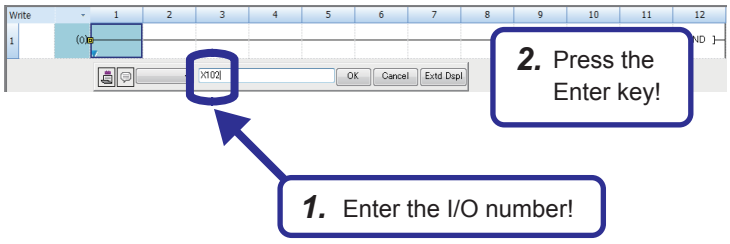
Point

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.

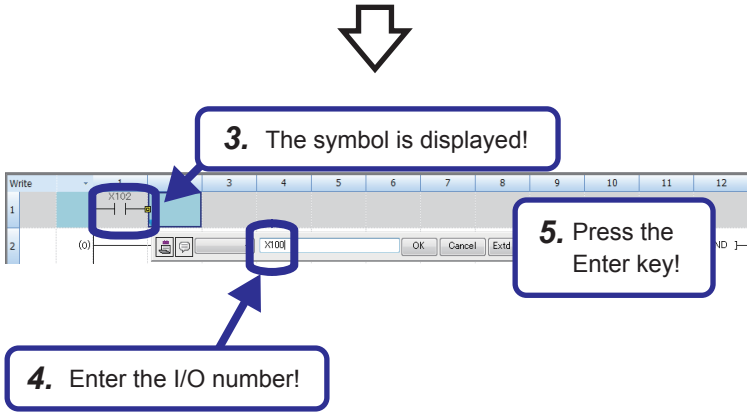
2.4.1 Creating a ladder program by entering devices and labels

Operating procedure



1. Move the cursor to the position where a ladder is added, and enter "X102". (When entering of the number starts, the ladder input window appears.)
To cancel an incorrect entry, press the Esc key.

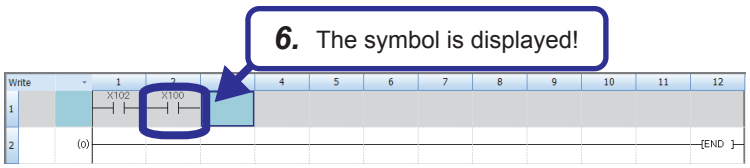
2. To confirm the entry, press the Enter key.
- Clicking the [OK] button also confirms the entry.
- Clicking the [Cancel] button also cancels the entry.



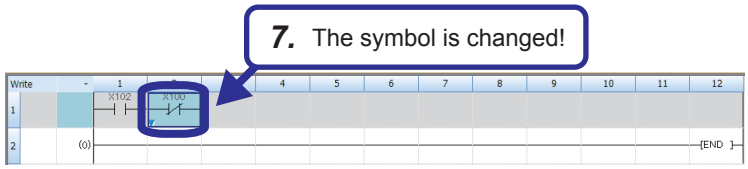
3. The added symbol (X102) is displayed.

4. Move the cursor to the next position and enter "X100".

5. Press the Enter key.



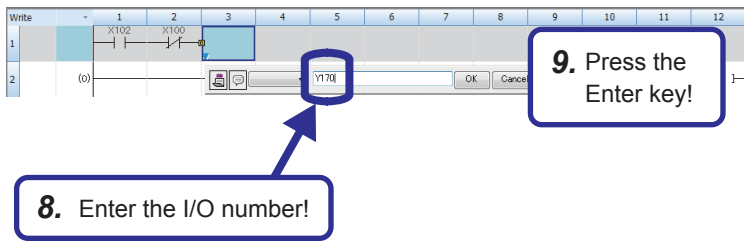
6. The added symbol (X100) is displayed.



7. Select the symbol and press the "/" key to switch the symbol with (X100).

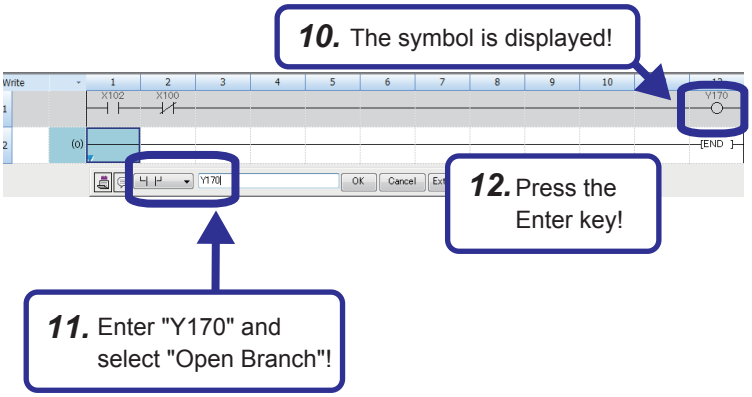
(To the next page)

(From the previous page)



8. Move the cursor to the next position and enter "Y170".

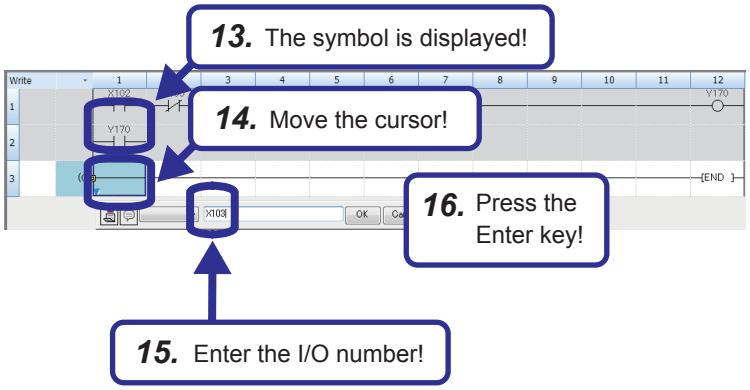
9. Press the Enter key.



10. The added symbol (—〔 Y170 〕—) is displayed.

11. Move the cursor to the next position, enter "Y170", and select "Open Branch".

12. Press the Enter key.



13. The added symbol (Y170) is displayed.

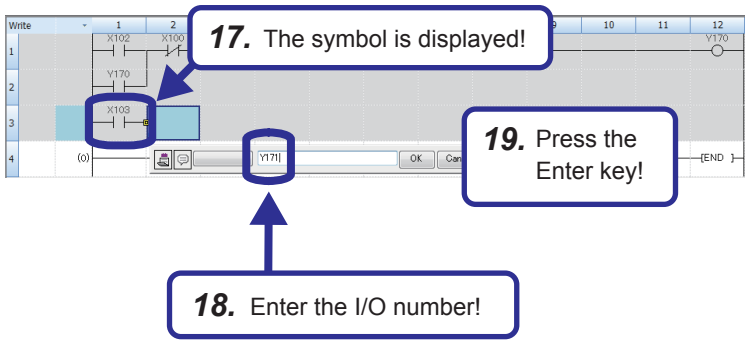
14. Move the cursor to the ladder under Y170.

15. Enter "X103".

16. Press the Enter key.

(To the next page)

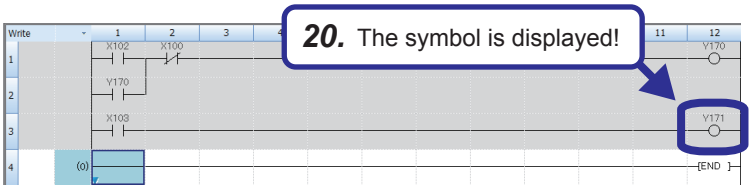
(From the previous page)



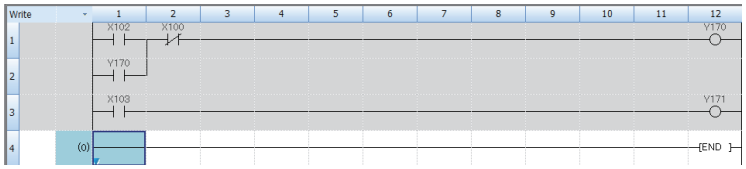
17. The added symbol (X103) is displayed.

18. Move the cursor to the next position and enter "Y171".

19. Press the Enter key.



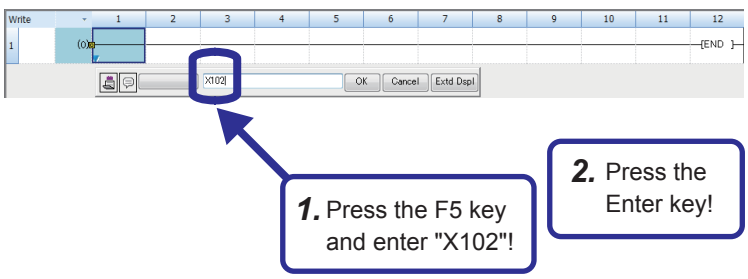
20. The added symbol (Y171) is displayed.



21. Creating a ladder program is completed.

2.4.2 Creating a ladder program with function keys

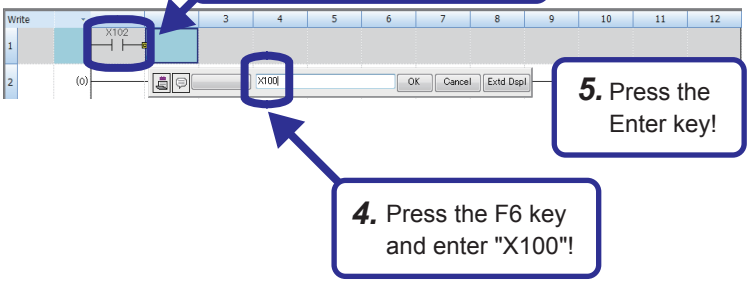
Operating procedure



1. Press the F5 key to open the ladder input window, and enter "X102".
To cancel an incorrect entry, press the Esc key.

2. To confirm the entry, press the Enter key.

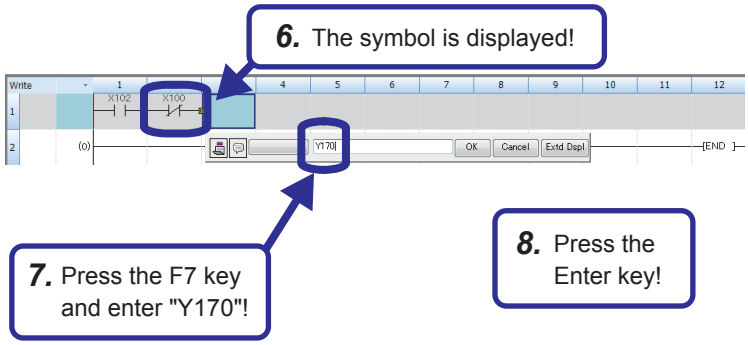
- Clicking the [OK] button also confirms the entry.
- Clicking the [Cancel] button also cancels the entry.



3. The added symbol (X102) is displayed.

4. Press the F6 key and enter "X100".

5. Press the Enter key after entering the device number.



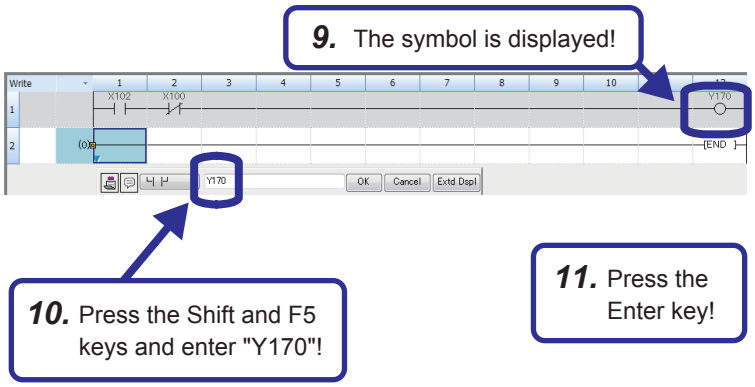
6. The added symbol (X100) is displayed.

7. Press the F7 key and enter "Y170".

8. Press the Enter key after entering the device number.

(To the next page)

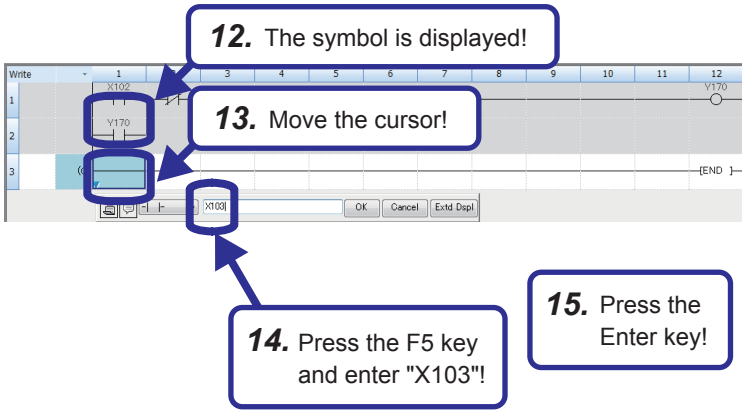
(From the previous page)



9. The added symbol (—[Y170]—) is displayed.

10. Press the Shift key and the F5 key and enter "Y170".

11. Press the Enter key after entering the device number.

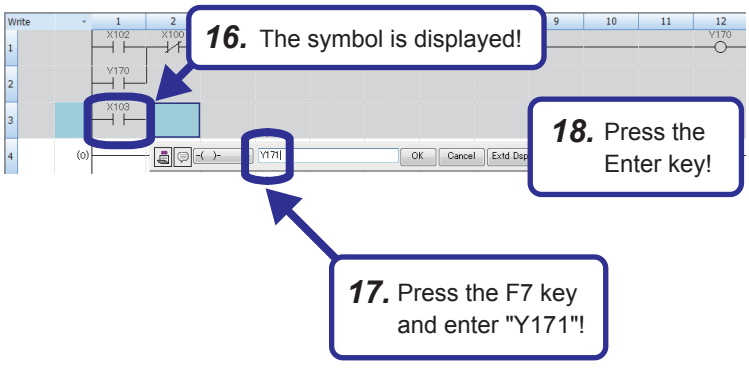


12. The added symbol (Y170) is displayed.

13. Move the cursor to the ladder under Y170.

14. Press the F5 key and enter "X103".

15. Press the Enter key after entering the device number.



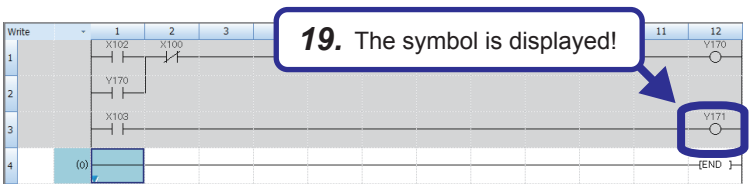
16. The added symbol (X103) is displayed.

17. Press the F7 key and enter "Y171".

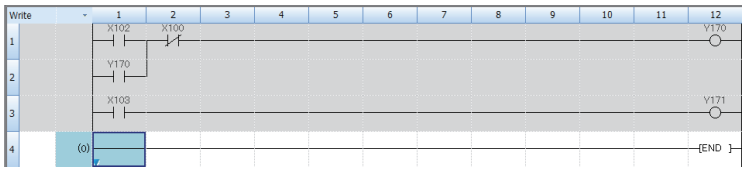
18. Press the Enter key after entering the device number.

(To the next page)

(From the previous page)



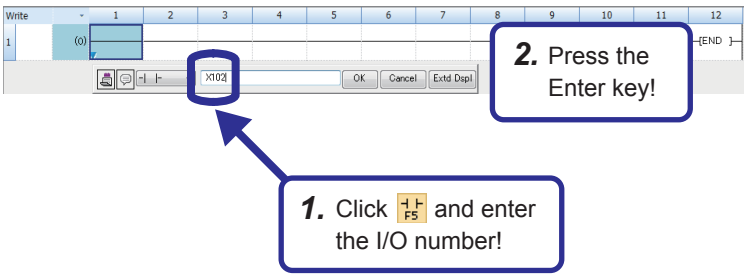
19. The added symbol (—〔 Y171 〕—) is displayed.



20. Creating a ladder program is completed.

2.4.3 Creating a ladder program with tool buttons

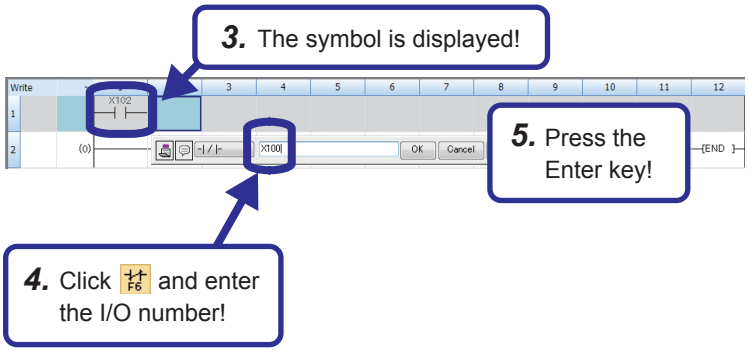
Operating procedure



1. Click F5 on the toolbar to open the ladder input window, and enter "X102".
To cancel an incorrect entry, press the Esc key.

2. To confirm the entry, press the Enter key.

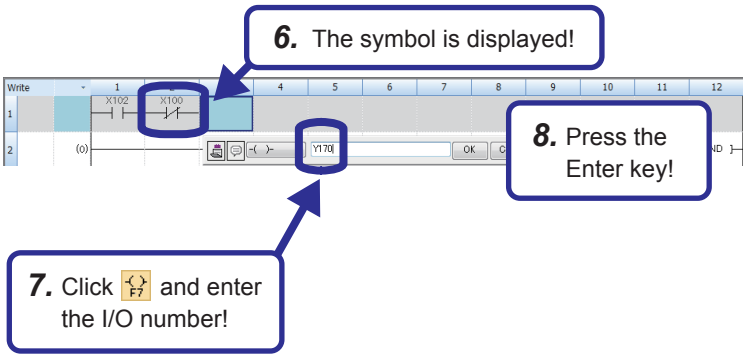
- Clicking the [OK] button also confirms the entry.
- Clicking the [Cancel] button also cancels the entry.



3. The added symbol (X102) is displayed.

4. Click F6 on the toolbar and enter "X100".

5. Press the Enter key.



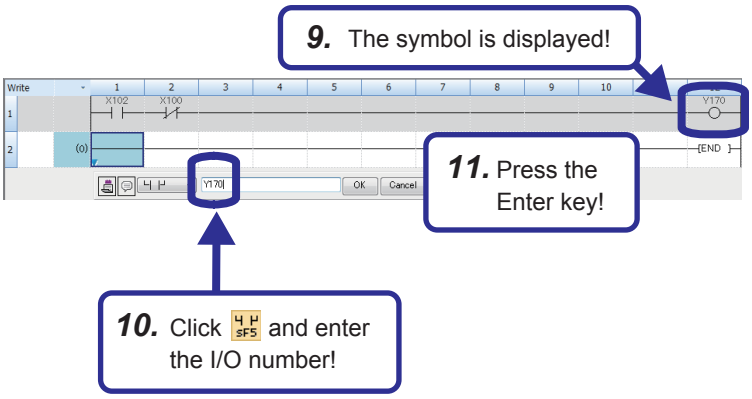
6. The added symbol (X100) is displayed.

7. Click F7 on the toolbar and enter "Y170".

8. Press the Enter key.

(To the next page)

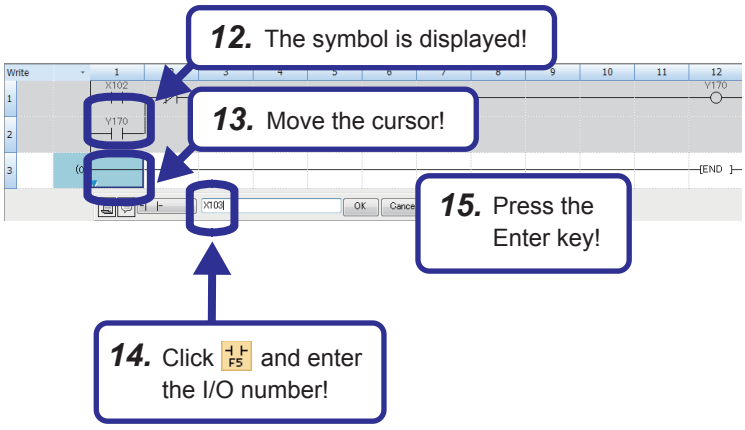
(From the previous page)



9. The added symbol (—(Y170)—) is displayed.

10. Click sF5 on the toolbar and enter "Y170".

11. Press the Enter key.

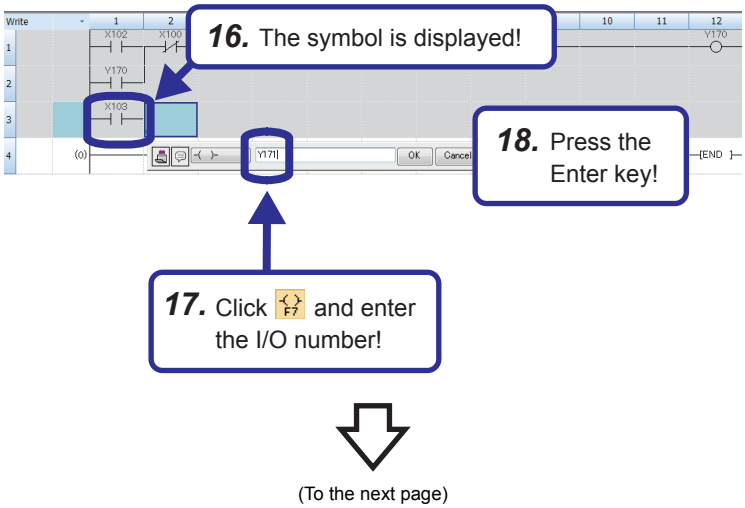


12. The added symbol (Y170) is displayed.

13. Move the cursor to the ladder under Y170.

14. Click F5 on the toolbar and enter "X103".

15. Press the Enter key.



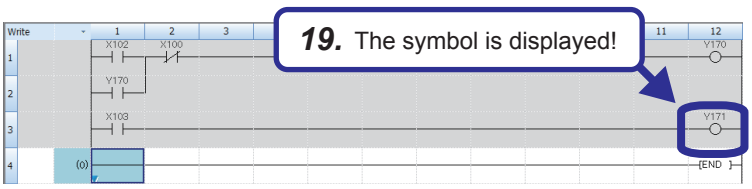
16. The added symbol (X103) is displayed.

17. Click F7 on the toolbar and enter "Y171".

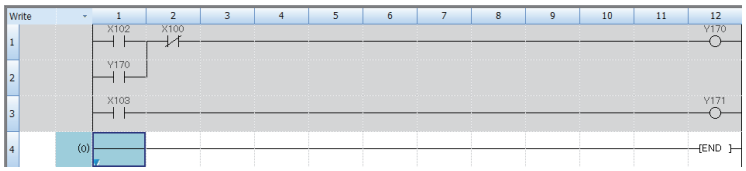
18. Press the Enter key.

(To the next page)

(From the previous page)



19. The added symbol (—〔 Y171 〕—) is displayed.



20. Creating a ladder program is completed.

4.8 Converting the Program

Compile the input ladder blocks.

Procedure

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.

4.9 Saving the Project

Save the created project.

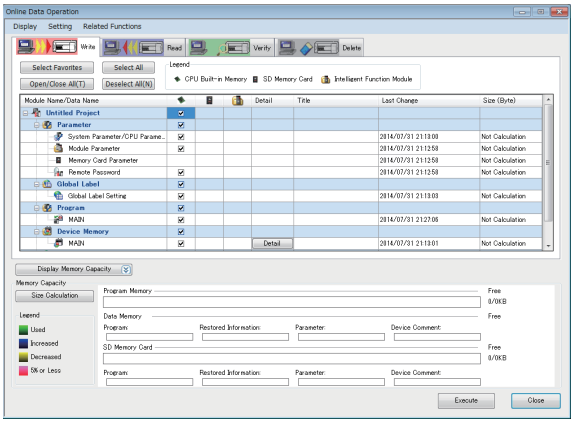
[Project] ⇨ [Save as]

4.10 Writing Data to the CPU Module

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

[Online] ⇨ [Write to PLC]

Procedure



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

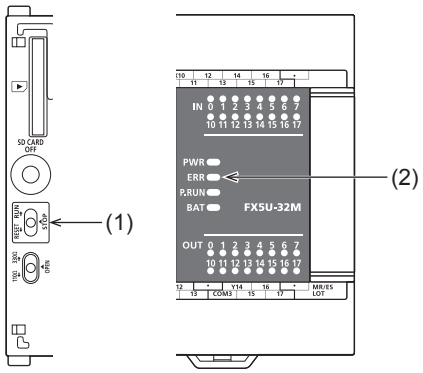
- For the operation of the CPU module, the system parameter file, CPU parameter file, and program file must be written. For the operation of the expansion adapter, I/O modules and intelligent function modules, the module parameter files must be written as well.
- Use of the [Select Favorites] button enables users to easily select frequently-used files, such as the system parameter file, CPU parameter file, and program file. Register items as favorites on the window opened by selecting [Setting] ⇨ [Register Favorites Selection] on the menu bar.

4.11 Resetting the CPU Module

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

Procedure

The following illustration is an example of the FX5U 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

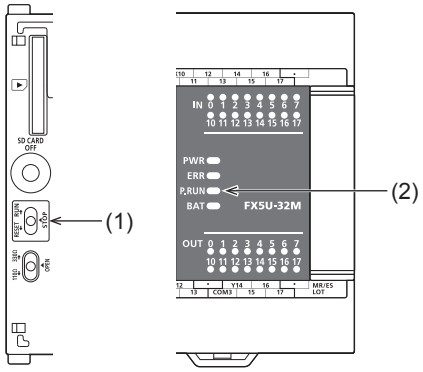
Operate the RUN/STOP/RESET switch with your fingers. Use of a tool such as a screwdriver may damage the switch.

4.12 Executing the Program

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

Procedure

The following illustration is an example of the FX5U CPU module.



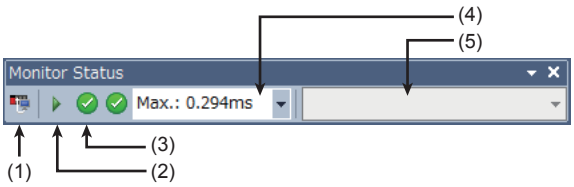
1. Set the RUN/STOP/RESET switch (1) to the RUN position.
2. Check that the P.RUN (2) turns on.

4.13 Monitoring the Program

Monitor the program operation using the engineering tool.

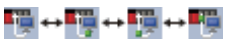
Monitoring on the monitor status bar

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



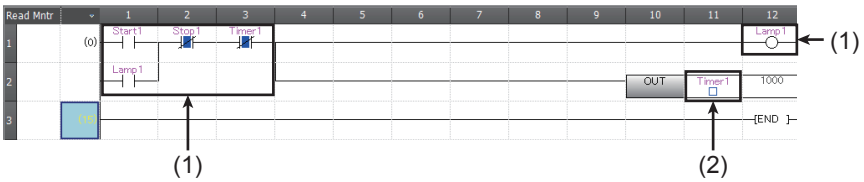
1. Select [Online] ⇨ [Monitor] ⇨ [Start Monitoring] on the menu bar.
2. Monitor the LED status of the CPU module and the scan time.

■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. (☞ Page 41 Module diagnostics (CPU Diagnostics))		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.

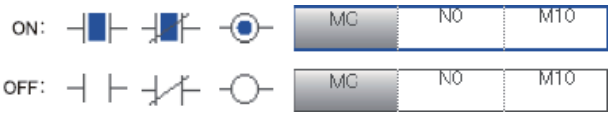


(1) The on/off states of the contacts and coils are displayed.
(2) The current value of the word/double word type data is displayed.

1. Select [Online] ⇨ [Monitor] ⇨ [Start Monitoring] on the menu bar.
2. Monitor the on/off states of contacts and coils and the current values of word devices and labels.

■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. (GX Works3 Operating Manual)

4.14 Troubleshooting

This section describes errors that may occur during system operation, error causes, and actions to be taken.
For the troubleshooting specific to each module, refer to the manual for the module used.

Point

Saving the program and devices at the time of an error helps to analyze the error cause. (GX Works3 Operating Manual)

Troubleshooting procedure

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. (☞ Page 40 Checking with LEDs)
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. (☞ Page 41 Module diagnostics (CPU Diagnostics))
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.

Point

Each LED status can be checked by using the Module diagnostics (CPU Diagnostics) of the engineering tool. (GX Works3 Operating Manual)

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 even after the items shown above are checked, 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)) (MELSEC iQ-F FX5UC 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 even after the items shown above are checked, 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. (☞ Page 40 Checking the ERR LED)

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.

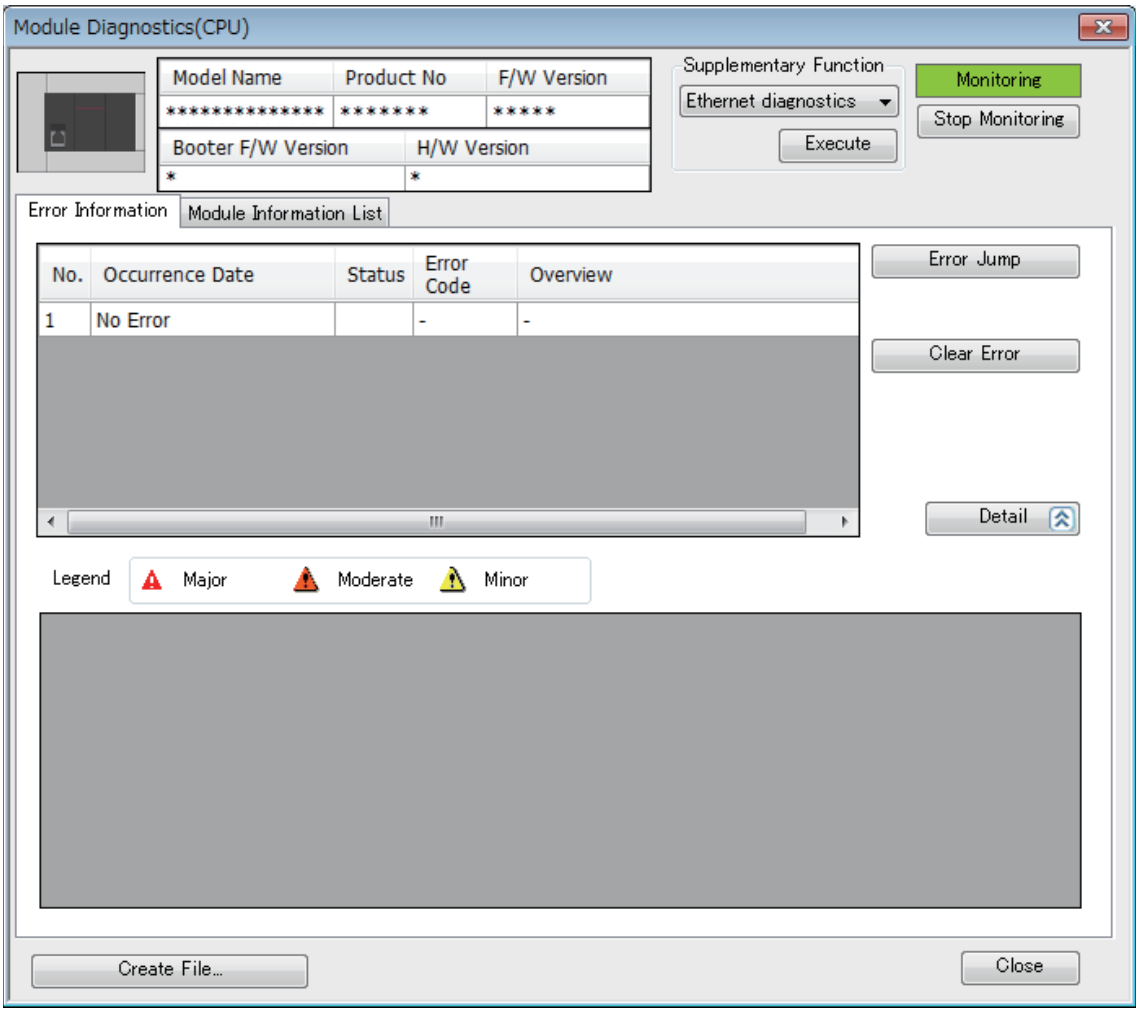
Function	Description
Module diagnostics (CPU Diagnostics)	This function diagnoses the module. (The current error and its details can be checked.)

For details on each function, refer to the following.
GX Works3 Operating Manual

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)]

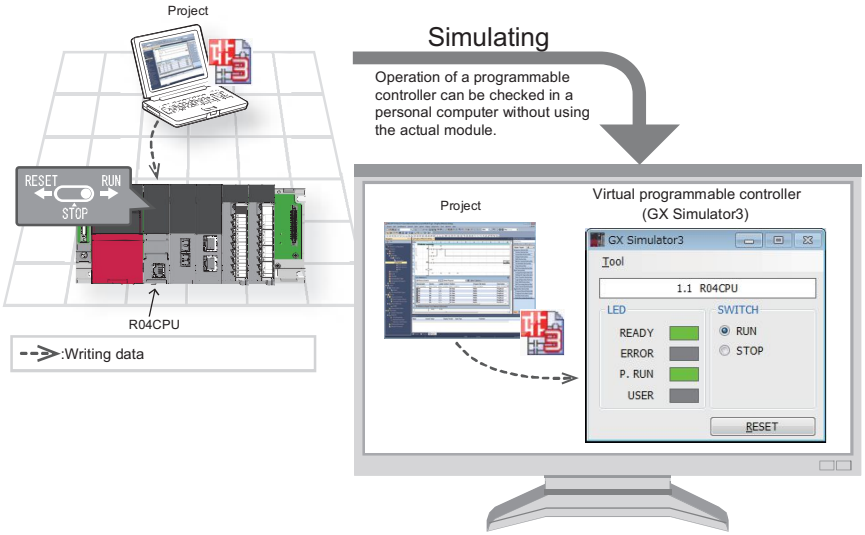
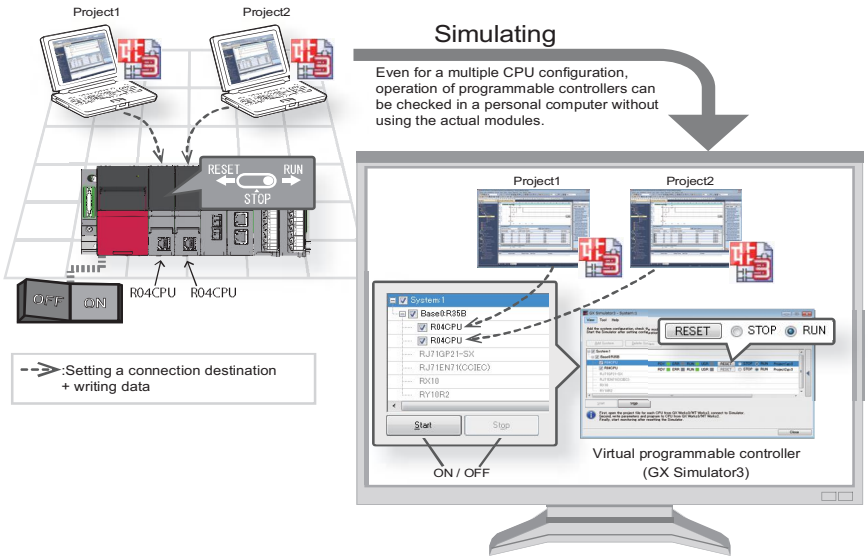


11 PROGRAM SIMULATION

This chapter explains how to debug programs offline by using the simulation function.
For modules supporting the simulation function, refer to the following section.
☞ Page 421 Using Simulation Function

11.1 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.
The following four simulations can be performed in GX Simulator3.

Purpose	Reference
Simulation of a CPU module (host CPU) 	Page 273 Simulation of a CPU module
Simulation of a multiple CPU system  FX5CPUs do not support it.	Page 274 Simulation of a multiple CPU system

11.2 Simulation

This section explains the simulation methods.
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.

Simulation of a CPU module

This can be used when performing the simulation to a CPU module.

Starting a simulation

Window

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

Operating procedure

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

■Simulation of an RnPCPU (redundant mode)

This function performs in the following condition:

- Operation mode: Separate mode
- Control system/standby system: Control system
- System A/B setting: System A
- Tracking transfer: No execution

The following function cannot be performed while simulating.
[Online] ⇨ [Redundant PLC Operation]
Only one base unit is acceptable for simulation. When constructing a system without setting the base setting of the system parameter, the number of base units is automatically set to eight. Therefore a redundant system configuration abnormal error occurs.

Ending a simulation

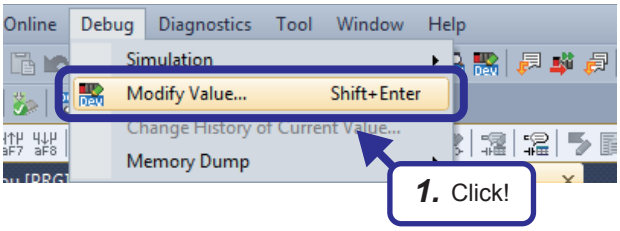
Operating procedure

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

11.3 System Simulation

The system simulation can be performed to simulate a program by interacting with other CPU modules and simple motion modules.
FX5CPUs do not support it.

6.1.1 Forced on/off of the device (Y)



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

1. Select the "Y170" cell on the ladder editor and click [Debug] → [Modify Value] from the menu.
Clicking the menu forcibly turns on or off "Y170".

Checking with the demonstration machine

Check that clicking the menu switches the on/off status of Y170 and the LED of Y170 on the demonstration machine also turns on and off depending on this operation.

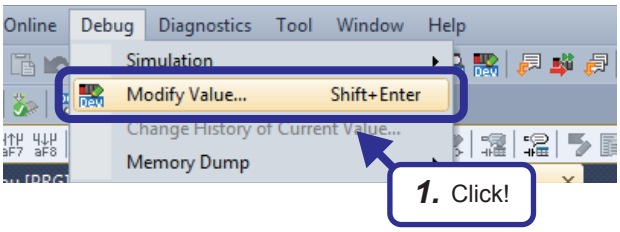
Precautions

When the CPU module is in the RUN state, operation results of the program are displayed preferentially. Thus, set the CPU module to the STOP state before checking with the demonstration machine.

Point

Setting and resetting of contacts, changing current values of word devices, and forced output can also be performed with the test function during ladder monitoring with GX Works3.
Double-clicking a contact (pressing the Enter key) while holding the Shift key in the ladder monitor window of GX Works3 forcibly switches the on/off status of the contact.
For word devices, this operation registers the change target devices on the watch window and changes the current values.

6.1.2 Setting/resetting of the device (M)



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

1. Select the "M10" cell on the ladder editor and click [Debug] → [Modify Value] from the menu.
Clicking the menu sets or resets "M10".

Checking with the demonstration machine

Turn off X104 and check the following.

❶ When M10 is set, M10 goes in the non-continuity state and the current value of the timer T0 is cleared to 0.
Check that the display of the initial indication device (D1) stops.

❷ When M10 is reset, M10 goes in the continuity state and the timer T0 starts counting from 0. The counted value increases by 10 every second.
Check that the value in the initial indication device (D1) increases by 10 every second.

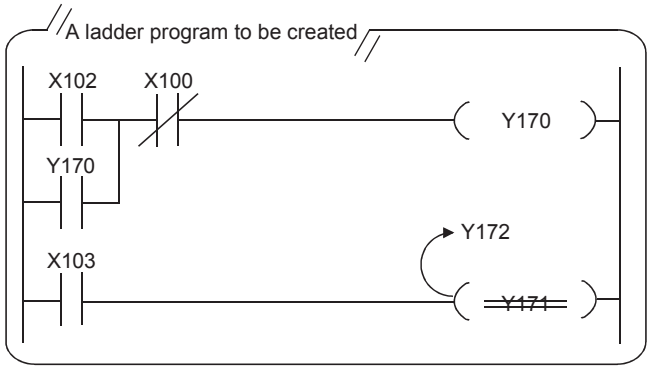
Point

With the same procedure, bit devices other than the internal relay (M) also can be set or reset forcibly.

2.9 Editing a Ladder Program

2.9.1 Modifying a part of a ladder program

Operating procedure



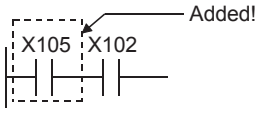
This section describes how to modify a part of the ladder program shown on the left. (OUT Y171 → OUT Y172)

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

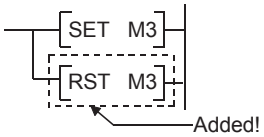
1. Check that "Overwrite" is displayed at the bottom right of the window.

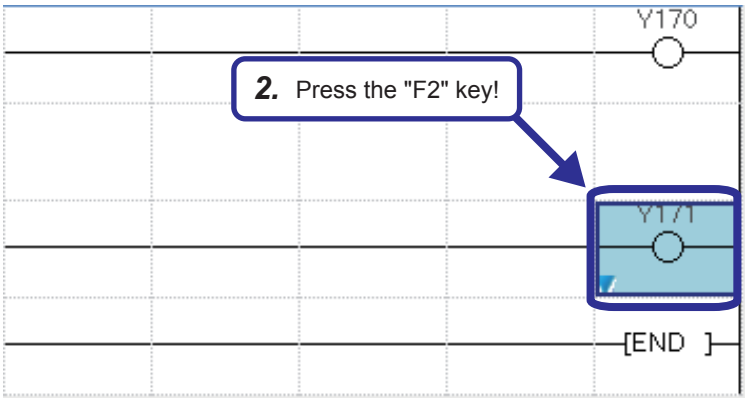
When "Insert" is displayed, press the Insert key to switch to "Overwrite".
When "Insert" is displayed, contacts or coils are added.

<When changing X102 with X105 is attempted>



<When changing SET with RST is attempted>

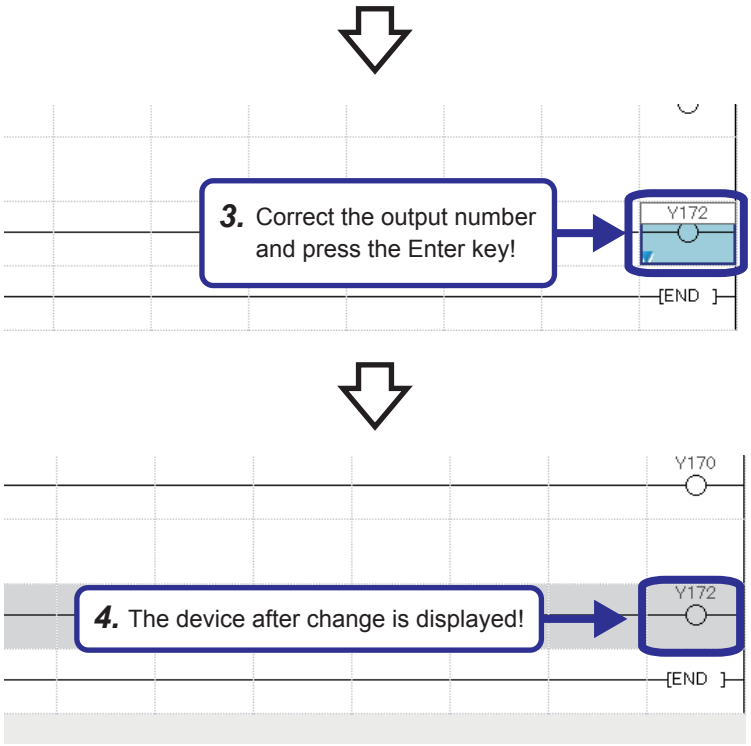




2. Click the position to be modified, and press the "F2" key.

(To the next page)

(From the previous page)



3. The device can be modified. Modify the device to "Y172", and press the Enter key.

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

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

Point

How to input contacts and coils

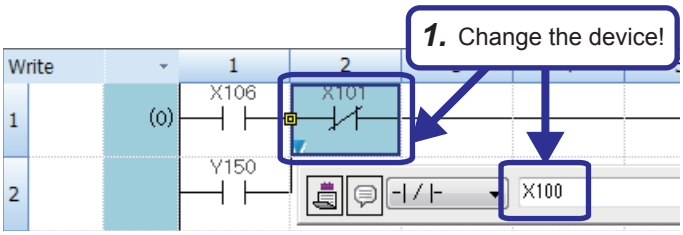
To input a contact or coil, specify a position where a contact or coil is to be input with the cursor and enter a device and label.
If an added ladder is in contact with the right rail or is an output device (Y, DY), the ladder is recognized as a coil. If not, it is recognized as a contact.

6.4 Online Change

Point

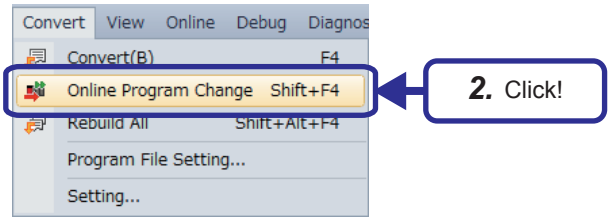
This section describes how to change a program while the CPU module is in the RUN state.

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

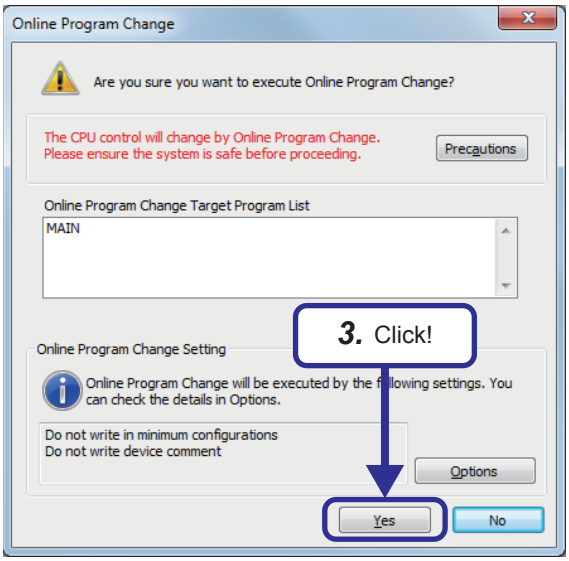


Set the CPU module to the RUN state before this operation.

1. Change the ladder.
(In this example, change "X101" to "X100".)



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.

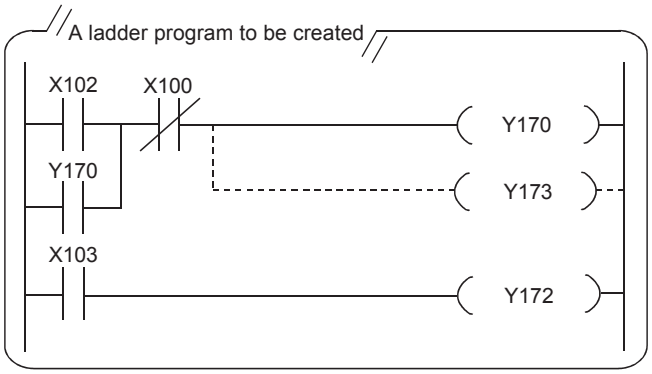
4. Online change is completed.

Precautions

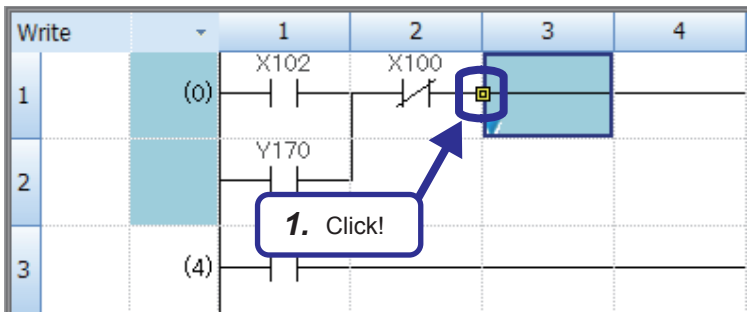
Online change cannot be executed when the program in the CPU module and the program before the modification in GX Works3 do not match. Thus, when whether the programs match or not is unclear, verify them before the modification with GX Works3, and execute the online change.

2.9.2 Drawing a line

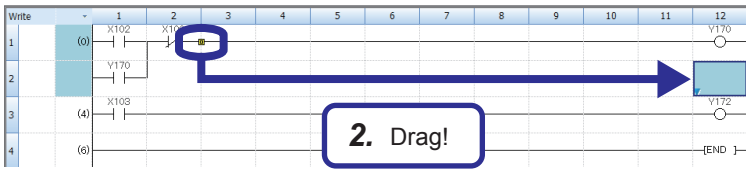
Operating procedure



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

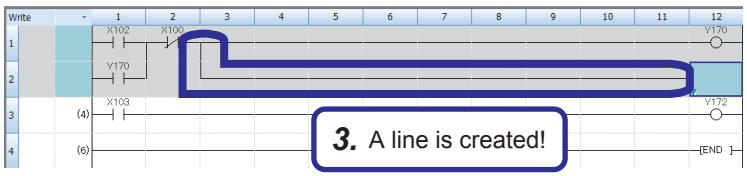


1. Move the mouse pointer close to the exiting line, and click the displayed icon.



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

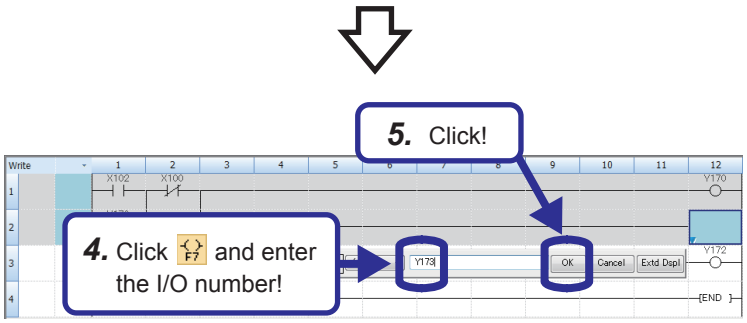
A vertical line is drawn on the left side of the cursor.



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

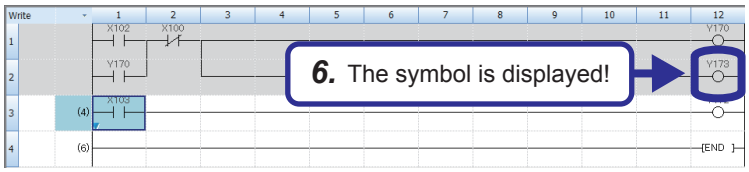
(To the next page)

(From the previous page)



4. Click F7 on the toolbar and enter "Y173".

5. Click the [OK] button.



6. The added symbol (—(Y173)—) is displayed.

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

Point

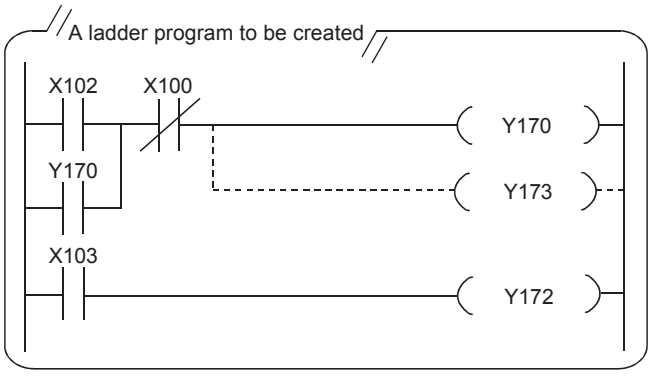
Adding or deleting a line with the key operation

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

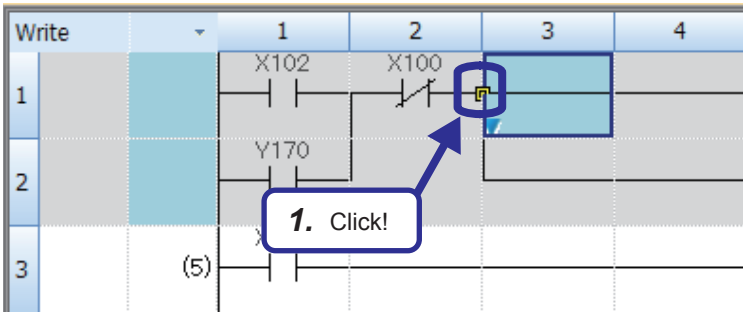
Users can draw a horizontal line from the cursor position to the position of the next contact, coil, or line by pressing Ctrl + Shift + ← or →.

2.9.3 Deleting a line

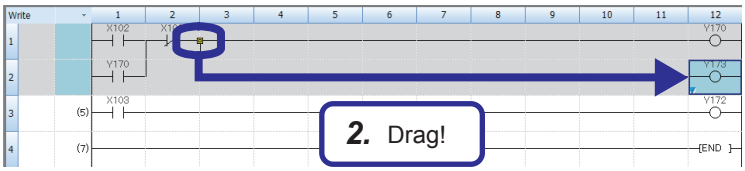
Operating procedure



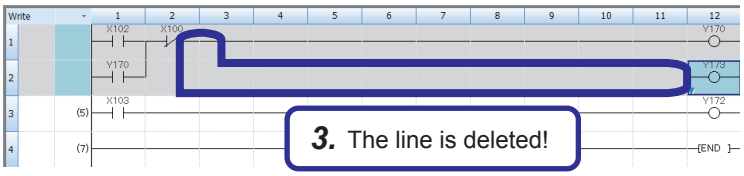
This section describes how to delete a line in the ladder program shown on the left.



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

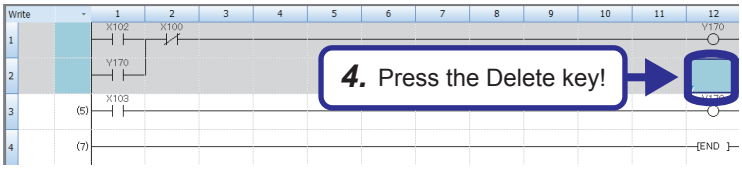


2. Drag the icon along the line to be deleted.



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 (—(Y173)—).

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

Point

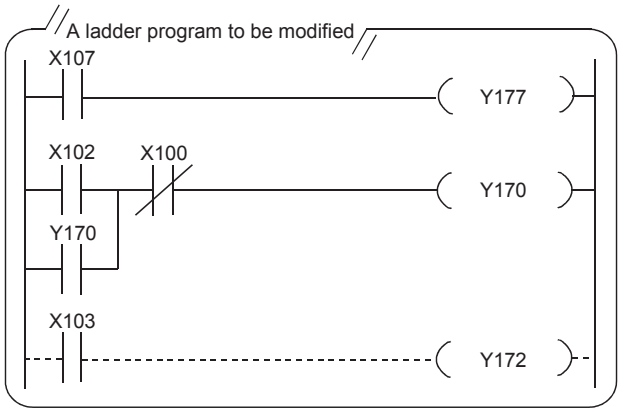
Adding or deleting a line with the key operation

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

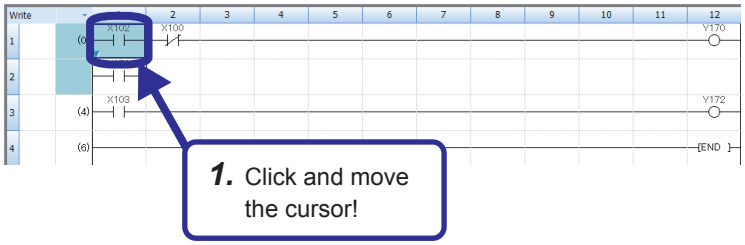
Users can draw a horizontal line from the cursor position to the position of the next contact, coil, or line by pressing Ctrl + Shift + ← or →.

2.9.4 Inserting a row

Operating procedure

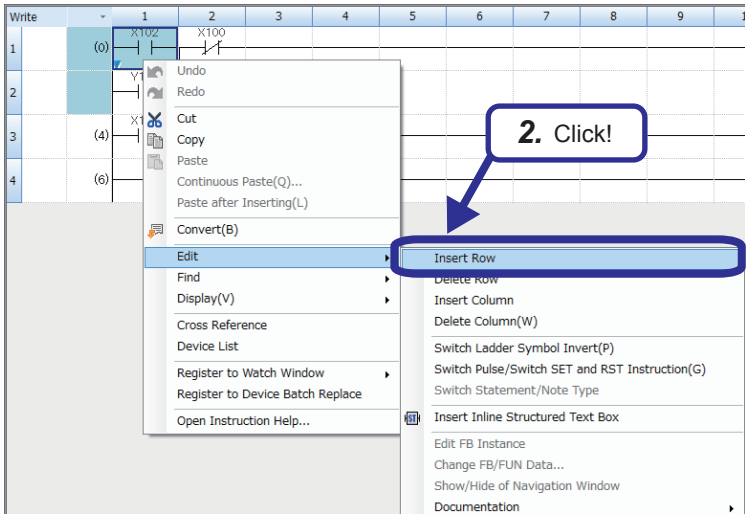


This section describes how to insert a row in the ladder program shown on the left.



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

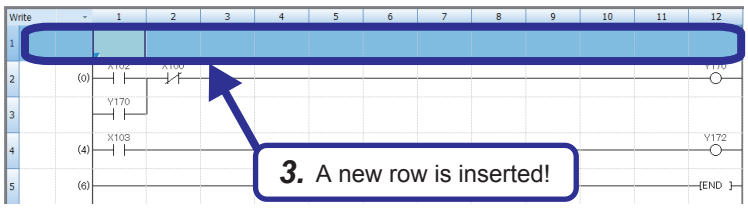
A new row is inserted above the row.



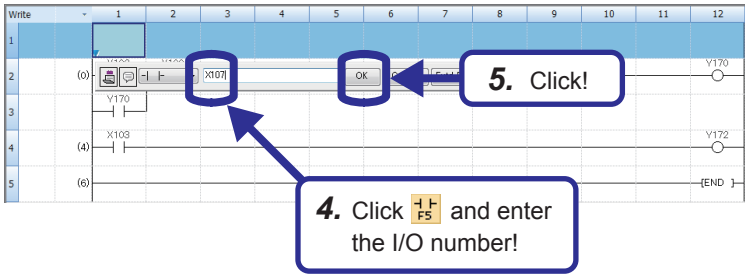
2. Right-click on the ladder editor, and click [Edit] → [Insert Row] (Shift + Insert) from the menu.

(To the next page)

(From the previous page)

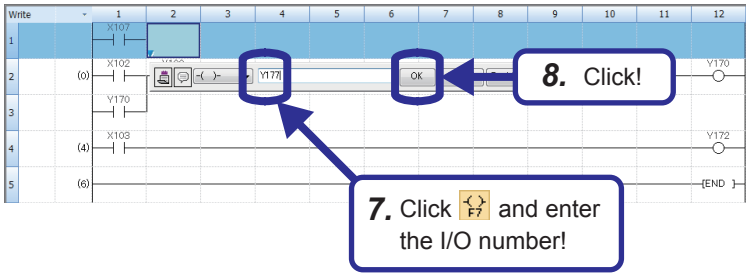


3. A new row is inserted above the row.



4. Click F5 on the toolbar to open the ladder input window, and enter "X107".

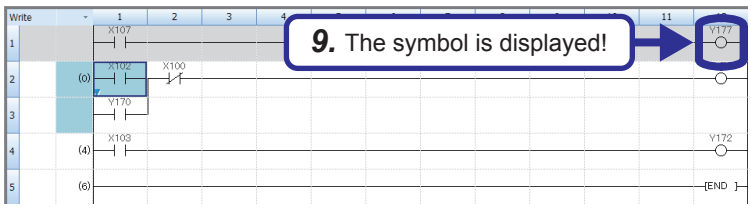
5. Click the [OK] button to confirm the entry.



6. The added symbol (X107) is displayed.

7. Click F7 on the toolbar and enter "Y177".

8. Click the [OK] button.

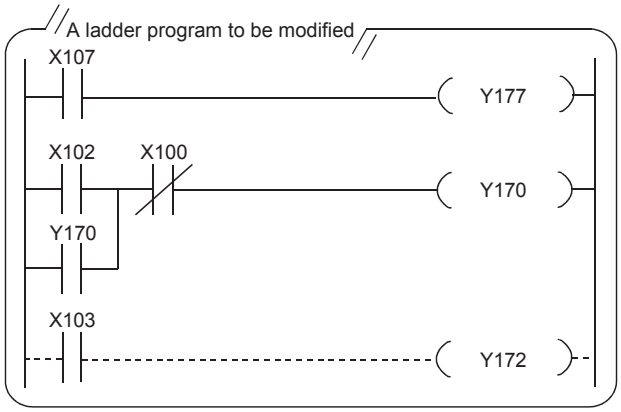


9. The added symbol (Y177) is displayed.

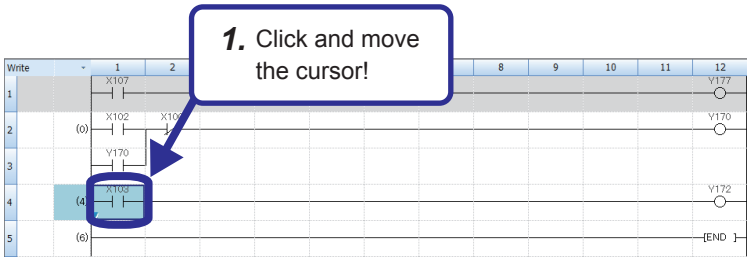
10. Click [Convert] → [Convert] (F4) from the menu to convert the ladder program.

2.9.5 Deleting a row

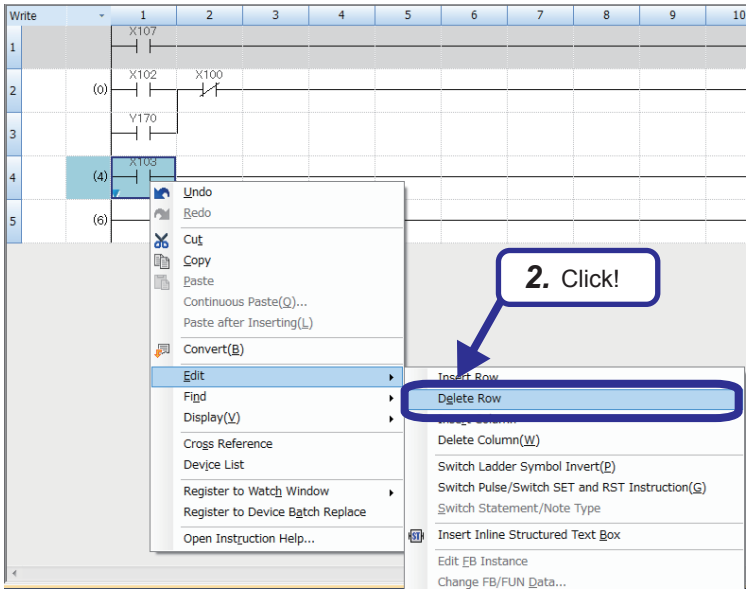
Operating procedure



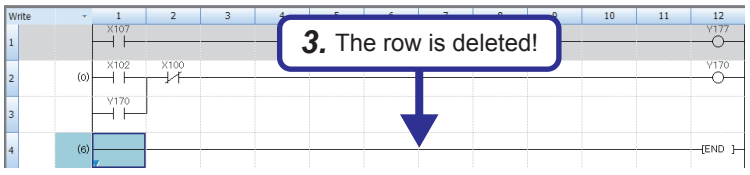
This section describes how to delete a row in the ladder program shown on the left.



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.

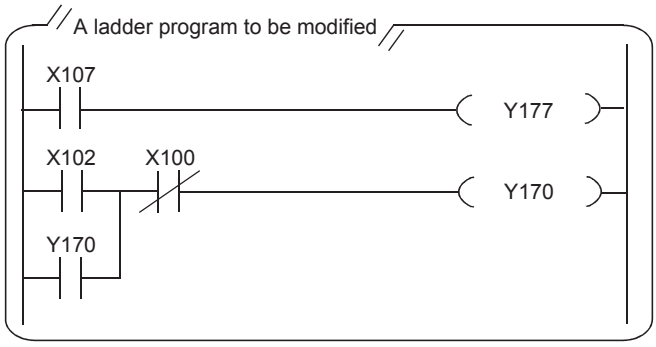


3. The row is deleted.

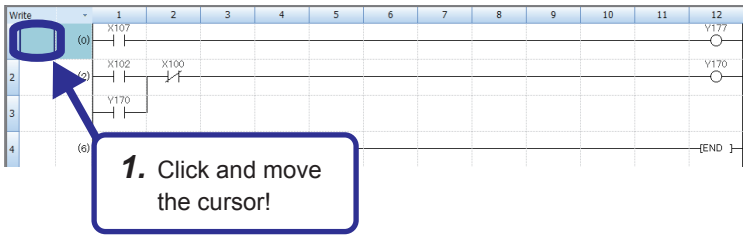
4. Click [Convert] → [Convert] (F4) from the menu to convert the ladder program.

2.9.6 Cutting or copying a ladder

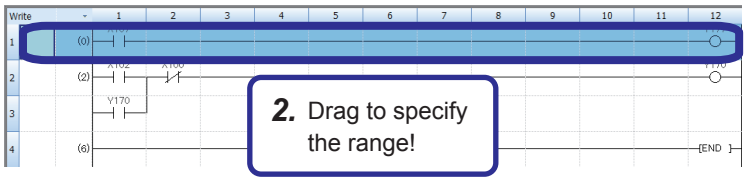
Operating procedure



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.

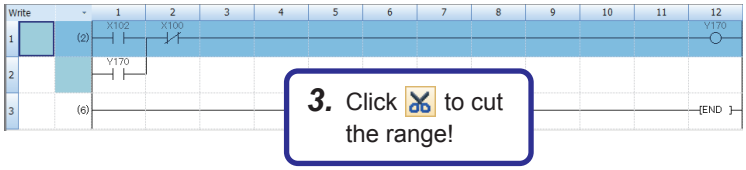


1. Click and move the cursor to the position where a part of the ladder program is to be cut.



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

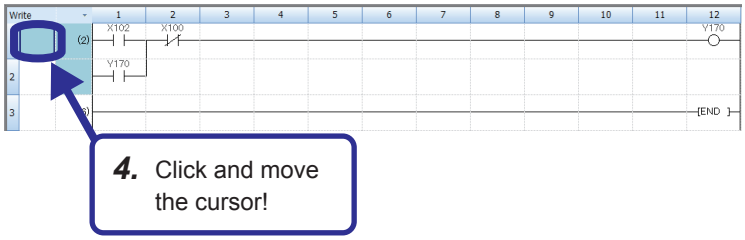
To easily specify the range in units of ladder blocks, click the position where a step number is displayed and drag the mouse vertically.



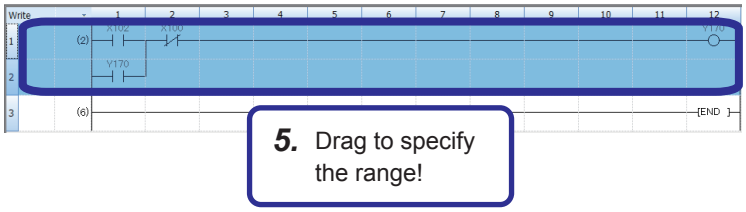
3. Click on the toolbar, or select [Edit] → [Cut] ([Ctrl] + [X]) to cut the ladder in the specified range.

(To the next page)

(From the previous page)

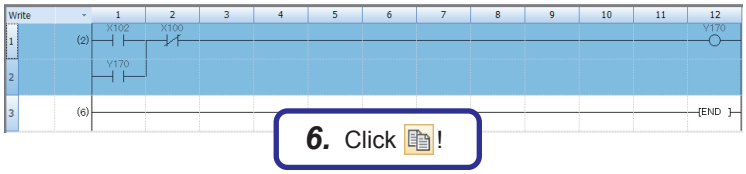


4. Click and the move the cursor to the position where a part of the ladder program is to be copied.

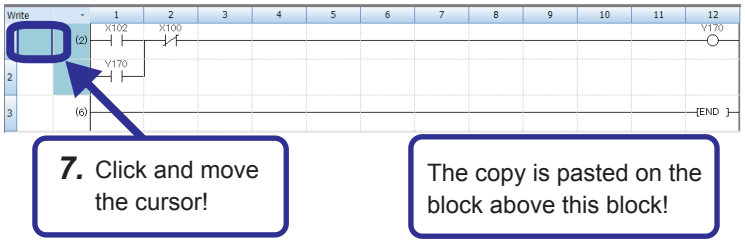


5. Drag the mouse to specify the copy range.
The color of the specified range is highlighted.

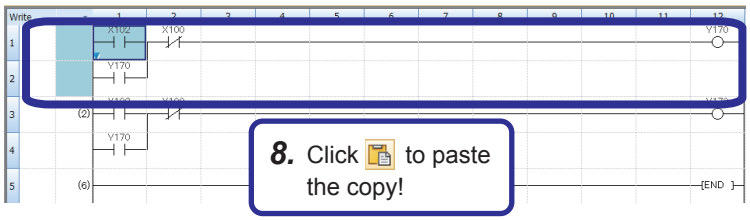
To easily specify the range in units of ladder blocks, click the position where a step number is displayed and drag the mouse vertically.



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.

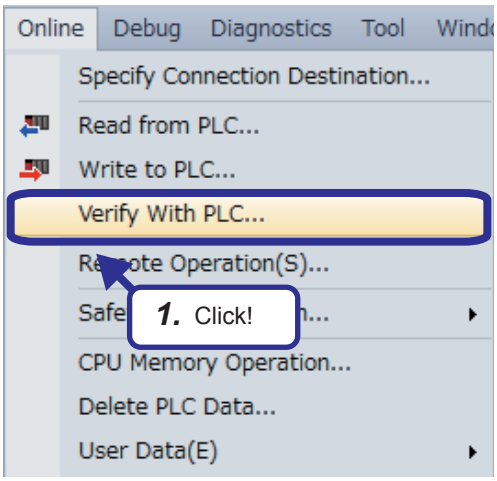


8. Click on the toolbar, or select [Edit] → [Paste] (Ctrl + V) from the menu.

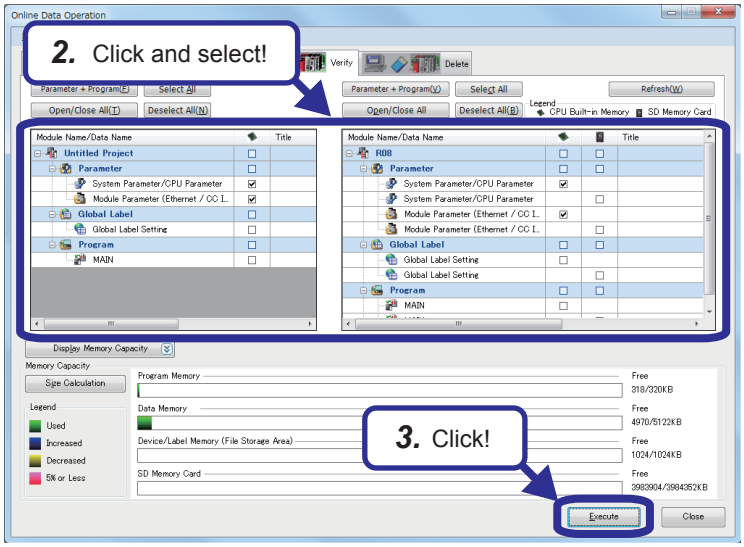
2.10 Verifying Data

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.

Operating procedure

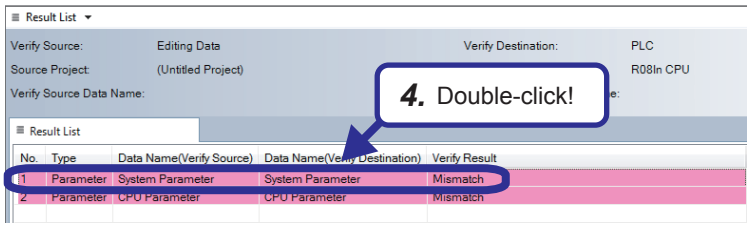


1. Click [Online] → [Verify with PLC] from the menu.



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

3. Click the [Execute] button.

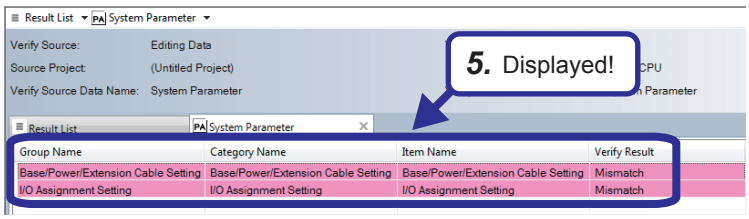


4. Verification results are displayed in the "Result List" window.

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

(To the next page)

(From the previous page)



5. Detailed results are displayed.

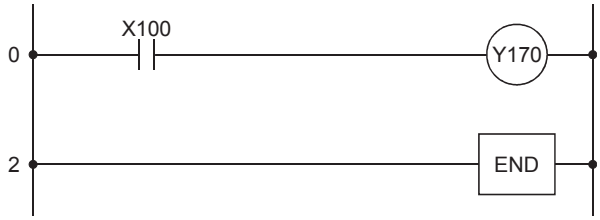
4.2 Differences Between [OUT] and [SET]/[RST]

Point

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

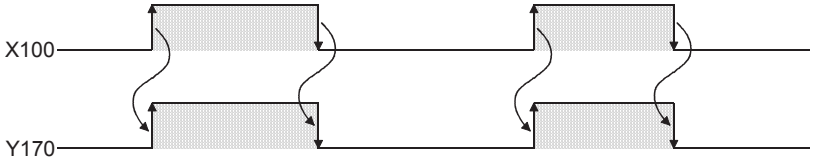
4.2.1 [OUT] (Coil output)

Project name	RB-1
Program name	MAIN



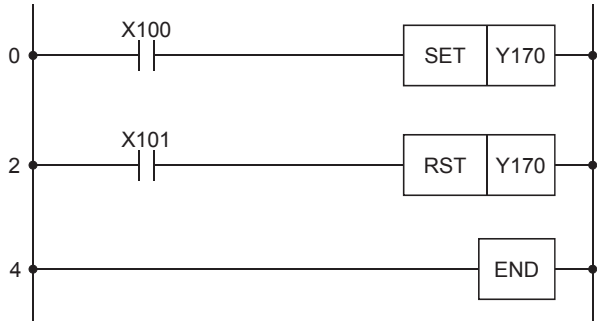
The OUT instruction turns on a specified device when the input condition turns on, and turns off the device when the condition turns off.

■Timing chart



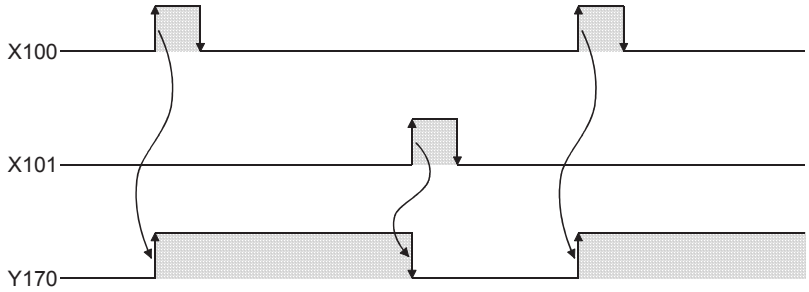
4.2.2 [SET]/[RST](Setting/resetting devices)

Project name	RB-2
Program name	MAIN



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.

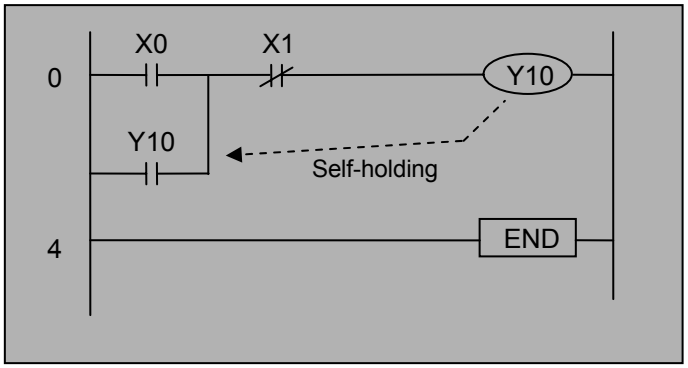
■Timing chart



2.4 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.

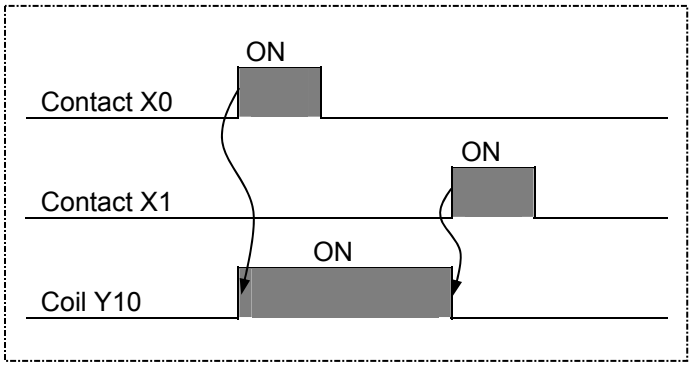
[Example of ladder diagram]



[List program]

Step	Instruction	
0	LD	X0
1	OR	Y10
2	ANI	X1
3	OUT	Y10
4	END	

[Timing chart]

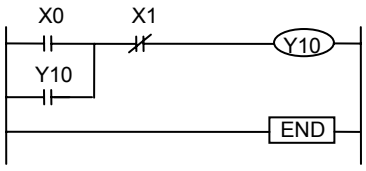


[Sequence]

When the input X0 is on and X1 is off, the output Y10 turns on.
The output Y10 remains on even when the input X0 is turned off (self-holding). Turning on the input X1 turns off the output Y10.

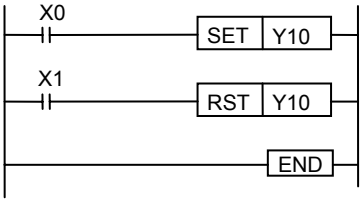
Both of the following ladders hold the output.

[Self-holding ladder of output]



Operation will be the same.

[SET/RST ladder of output]



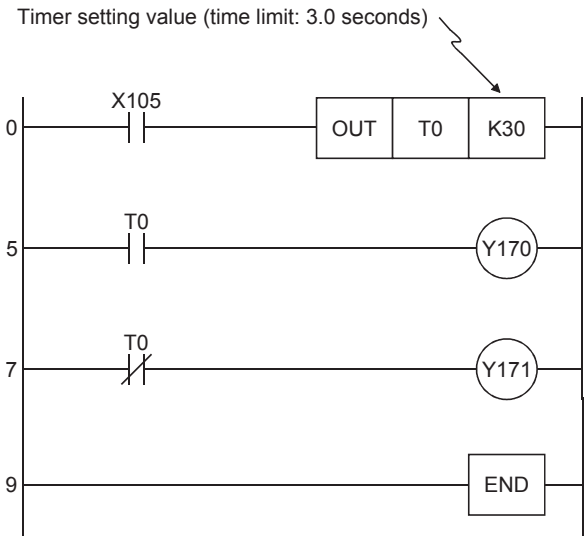
Using the SET instruction can hold the on status of the output even when the contact which drives the coil turns off.

4.3 Measuring Timers (Timer, High-speed Timer, Retentive Timer)

Point

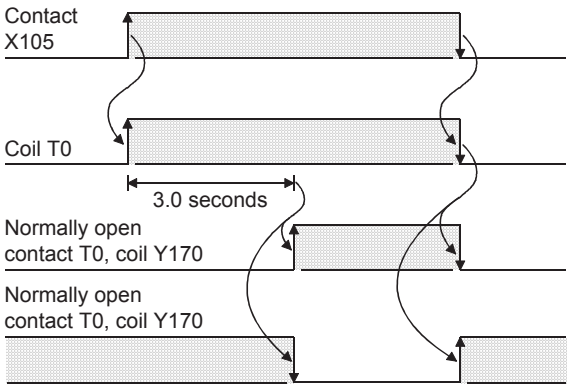
- This section describes how to input a timer.
- This section describes the parameter setting for using a retentive timer.
- This section describes the operation differences of various timers.

Project name	RB-3
Program name	MAIN



*OUT T is a 4-step instruction.

■Timing chart



- 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.

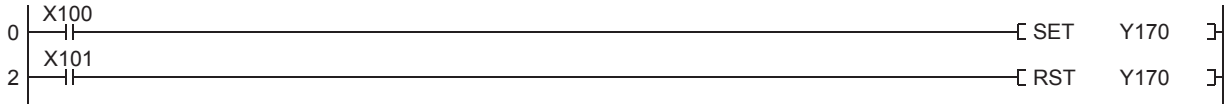
The following four types of timers are available.

Type	Timer No. (initial value)	
Low-speed timer •••••••• Counts time in increments of 100ms.	Initial value T0 to T2047 (2048 timers)	• Change the output instruction (OUT) to <^{H}T0> to select a high-speed timer or high-speed retentive timer.
High-speed timer •••••••• Counts time in increments of 10ms.		
Low-speed retentive timer ••••• Integrates time in increments of 100ms.	Initial value: 0 (The value can be changed with parameters.)	• To use retentive timers, set the number of device points used for retentive timers in the device setting of the CPU parameter.
High-speed retentive timer •••• Integrates time in increments of 10ms.		

Appendix 4 Program Examples

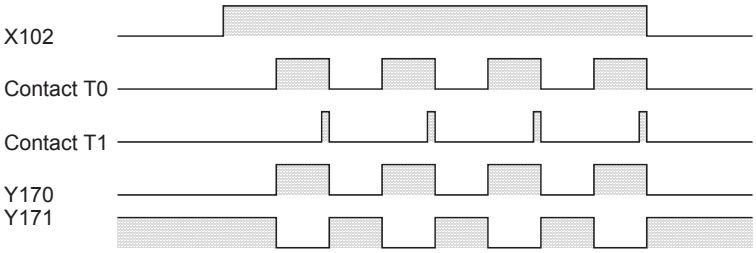
Appendix 4.1 Flip-flop ladder

1. When X100 turns on, Y170 turns on. When X101 turns on, Y170 turns off.



When X102 turns on, Y171 turns off if Y170 is on, or turns on if Y170 is off. This flip-flop operation is repeated.

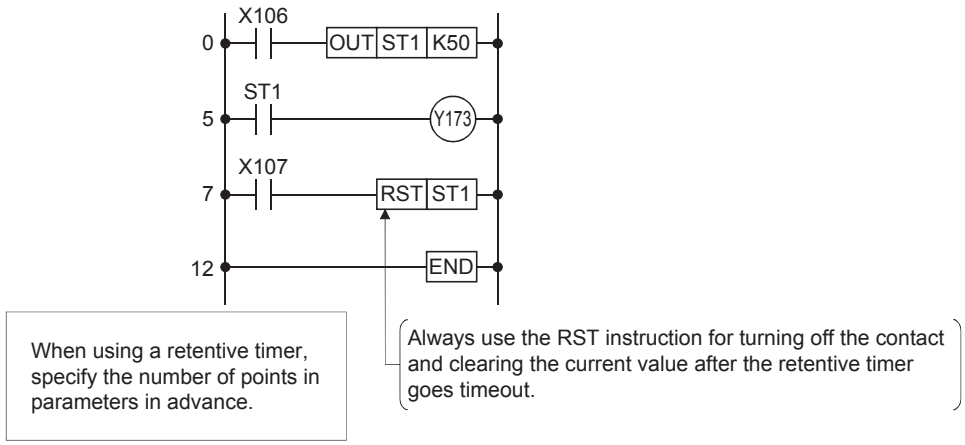
Project name	RA-16
Program name	MAIN



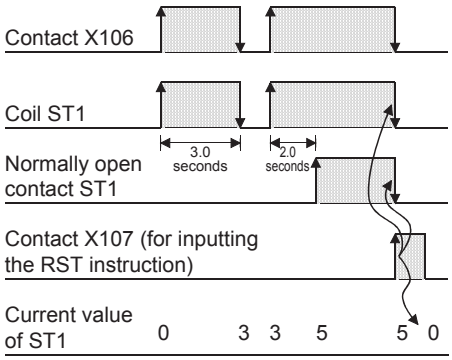
How to use retentive timers

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.

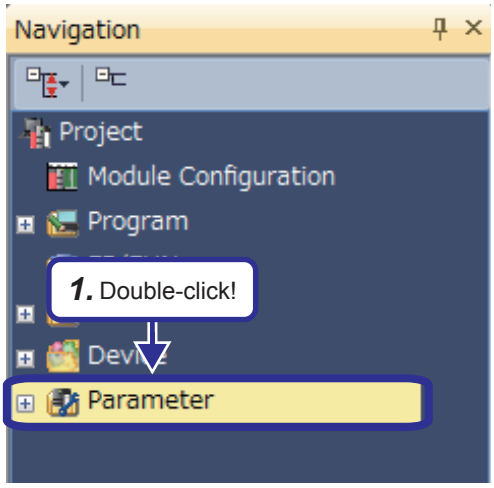
Project name	Retentive timer
Program name	MAIN



■Timing chart



The following describes the operation of when the retentive timer is set to ST0 to ST31.

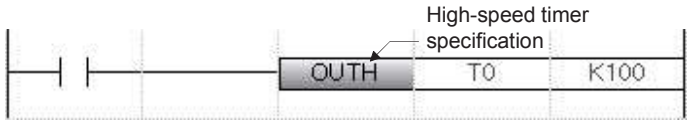
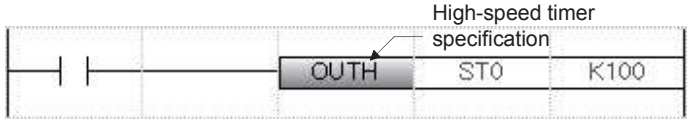


1. Double-click "Parameter" in the "Project" view.

(To the next page)

Appendix 5 Timer, Counter

Appendix 5.1 Timer

Function		RCPU
Low-speed timer	Measurement unit	• 100ms (default) The unit can be changed within the range of 1 to 100ms in parameters.
	Specification method	
High-speed timer	Measurement unit	• 10ms (default) The unit can be changed within the range of 1 to 100ms in parameters.
	Specification method	 * Configure the high-speed timer setting with sequence programs.
Retentive timer	Measurement unit	Same as that of the low-speed timer
	Specification method	
High-speed retentive timer	Measurement unit	Same as that of the high-speed timer
	Specification method	 * Configure the high-speed timer setting with sequence programs.
Setting range		• 1 to 32767
Processing of the setting value 0		• Momentarily on
Index modification	Contact	• Available (Only Z0 and Z1 can be used.)
	Coil	• Available (Only Z0 and Z1 can be used.)
	Setting value	• Not available
	Current value	• Available (Z0 to Z15 can be used.)
Current value update processing		• When the OUT T☐ instruction is executed
Contact on/off processing		

Precautions for using timers

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

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

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

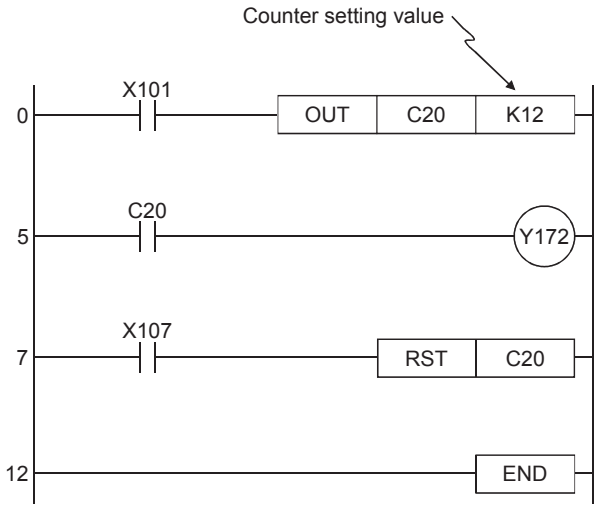
- The value set to a high-speed timer is smaller than the scan time.
- The value set to a low-speed timer is 1.

4.4 Counting with a Counter

Point

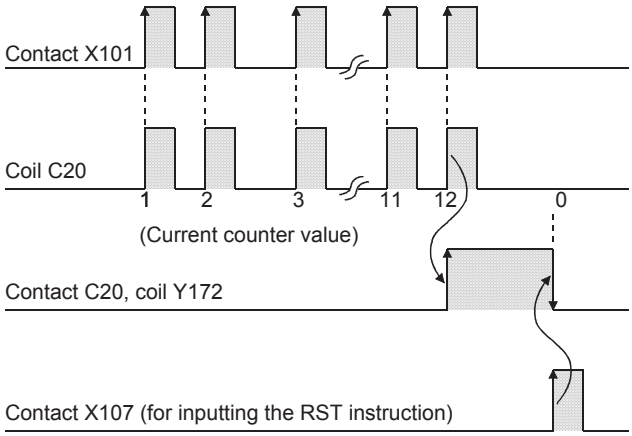
- This section describes how to input a counter.
- This section describes the words "rise (rising edge)" and "fall (falling edge)".

Project name	RB-4
Program name	MAIN



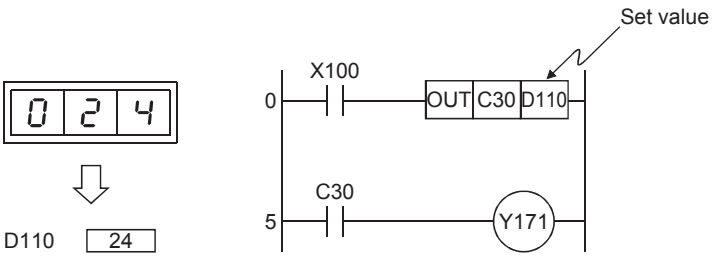
*OUT C is a 4-step instruction.

■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.)

A setting value can be directly specified with K or indirectly specified with D (data register).



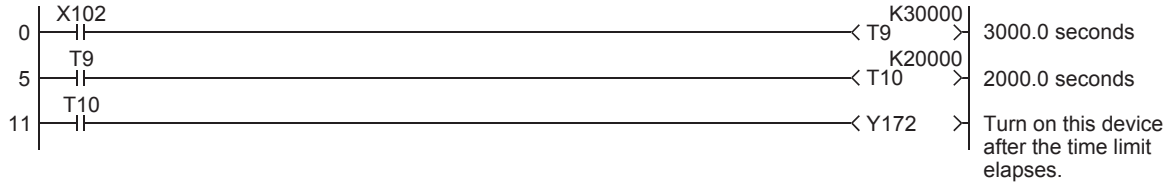
- The counter C30 counts up when the number of times that the input signal X100 turns on becomes equal to the value (such as 24) specified in the data register D110.
- The indirect specification is useful for using a value specified with an external device as the setting value of a counter.

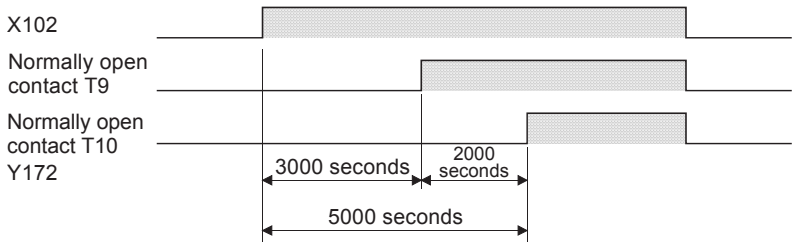
Point

A setting value of a timer can also be indirectly specified with the data register (D) in the same way as the one for counters.

Appendix 4.3 Long-time timer

1. Arrange timers in series to obtain necessary time.

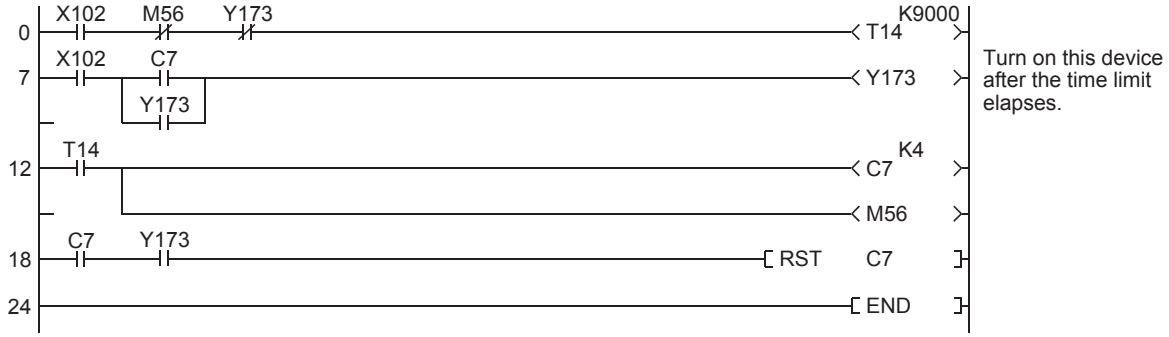


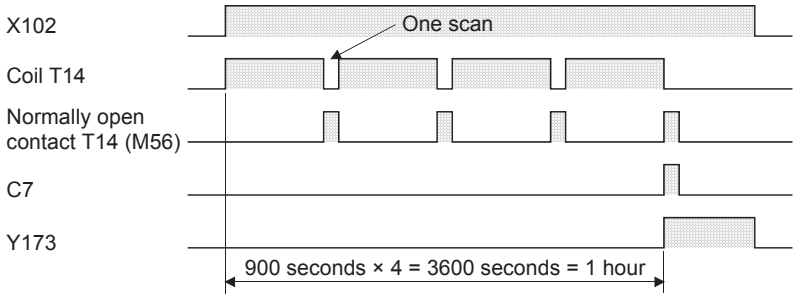


2. Use timers and counters to obtain necessary time.

Time limit of a timer × Set value of a counter = Long-time timer (Note that the accuracy of timers is accumulated.)

Project name	RA-18
Program name	MAIN





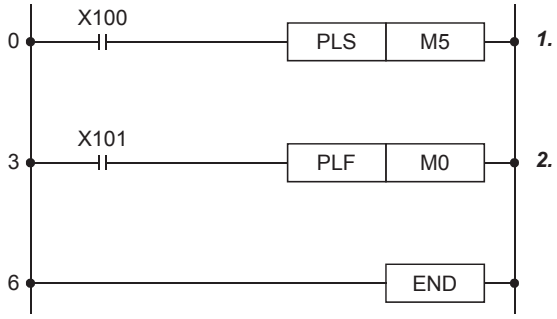
* Obtain necessary time by counting the number of timeouts of the timer T14 with the counter C7.
M56 resets T14 after a timeout. With C7, the output Y173 holds its ON state when counting is up. Y173 resets T14 and stops the subsequent time counting.

4.5 [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

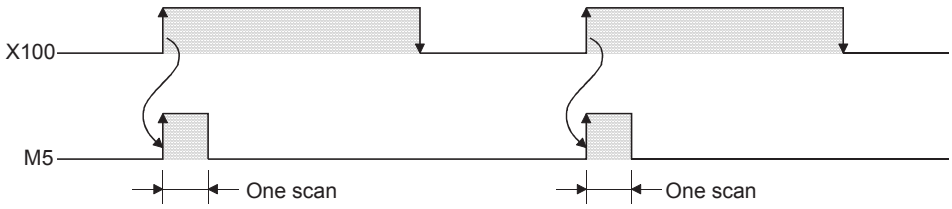
- This section describes the concept of one scan.
- This section describes the operation timing of the PLS/PLF instruction.

Project name	RB-5
Program name	MAIN



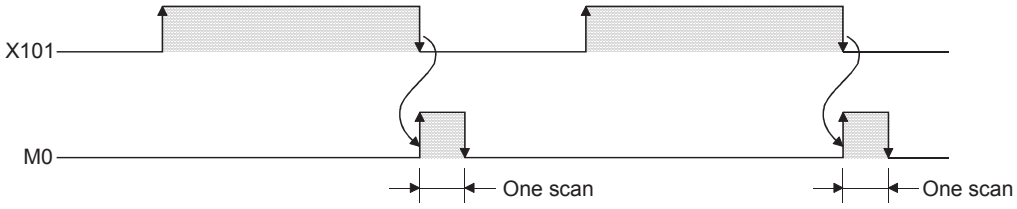
1. The PLS instruction turns on a specified device only for one scan at the rising edge of the commanded condition.

■Timing chart



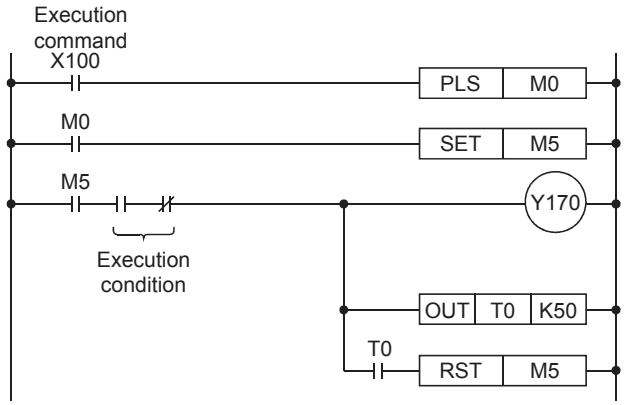
2. The PLF instruction turns on a specified device only for one scan at the falling edge of the commanded condition.

■Timing chart

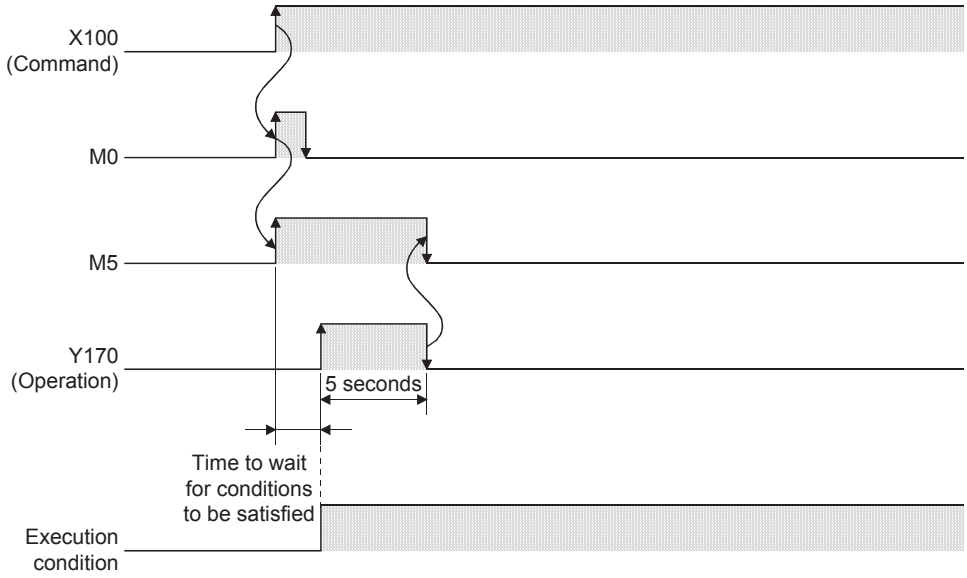


Application

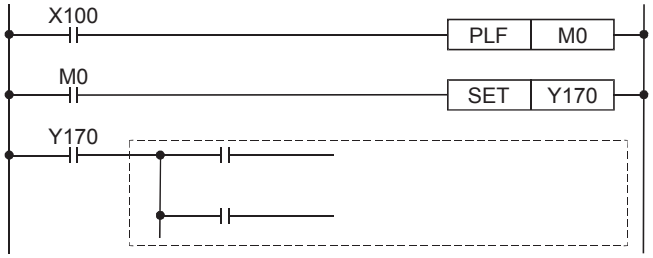
• These instructions can be used in a standby program that waits for an operation condition.

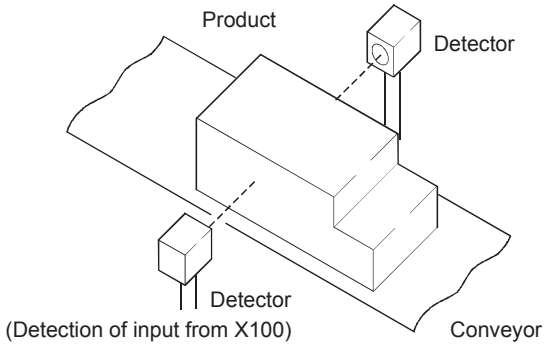


■Timing chart



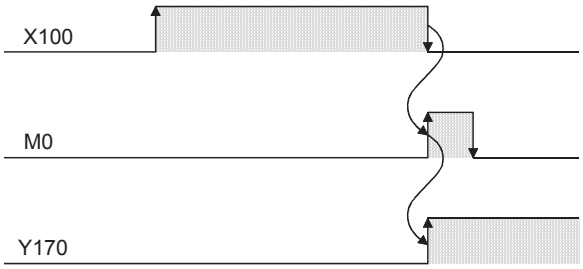
• These instructions can be used in a program that detects the passage of moving objects.
The program detects that products have passed through and starts the next processing for the products.





4

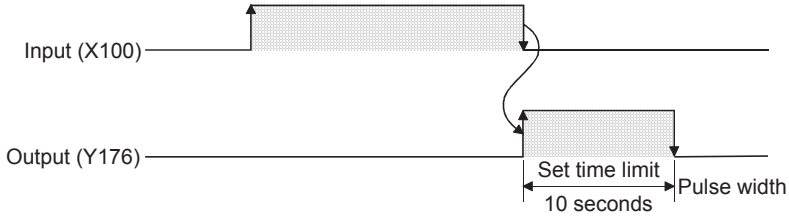
■Timing chart



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

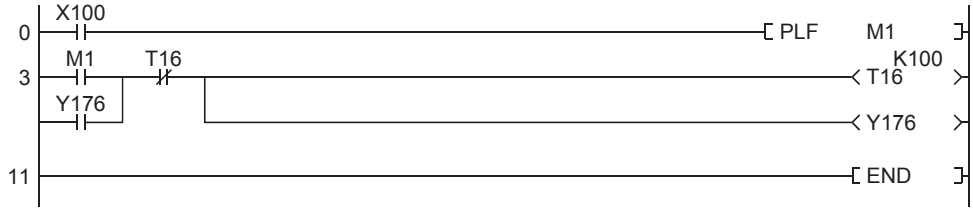
These instructions can be used in a program that executes an output operation for a set period of time at timing when an input signal turns on.

■Timing chart



■Program example

Project name	RB-6
Program name	MAIN

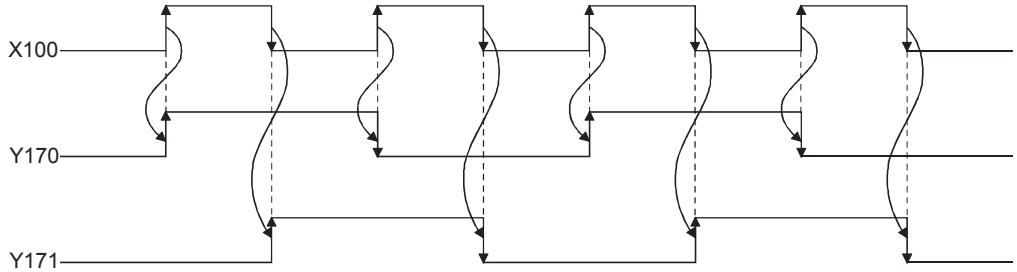


Useful application of the PLS/PLF instructions (Part 2)

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.

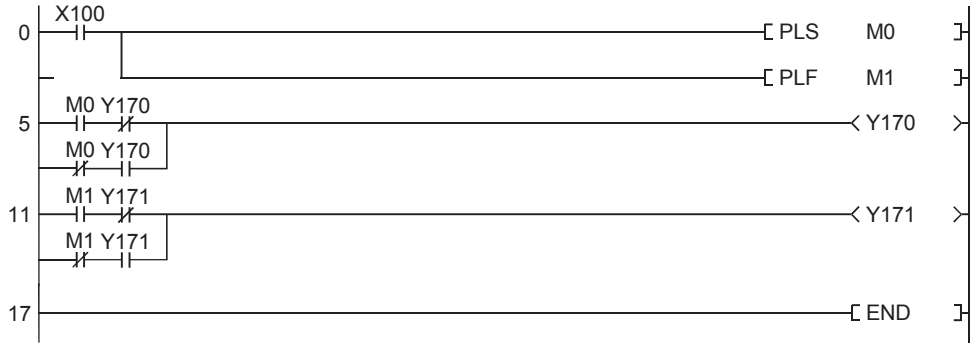
(If the PLS instruction is used in the program, the program is executed at the rising edge caused when the push button switch is pressed. If the PLF instruction is used, the program is executed at the falling edge caused when the switch is released.)

■Timing chart



■Program example

Project name	RB-7
Program name	MAIN



Pulse operation start, pulse series connection, pulse parallel connection

LDP, LDF, ANDP, ANDF, ORP, ORF

• LDP: Rising edge pulse operation start instruction

This becomes conductive (ON) only at the rising edge (OFF to ON) of the bit device specified by (s).

• LDF: Falling edge pulse operation start instruction

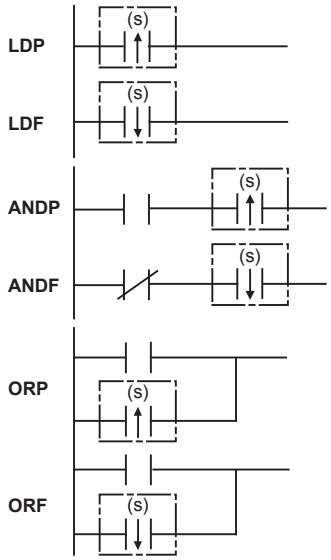
This becomes conductive (ON) only at the falling edge (ON to OFF) of the bit device specified by (s).

• ANDP: Rising edge pulse series connection instruction/ANDF: Falling edge pulse series connection instruction

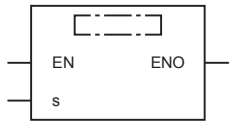
This instruction ANDs the bit device specified by (s) with the operation result so far, and uses the result as the operation result.

• ORP: Rising edge pulse parallel connection/ORF: Falling edge pulse parallel connection

This instruction ORs the bit device specified by (s) with the operation result so far, and uses the result as the operation result.

Ladder diagram	Structured text
	ENO:=LDP(EN,s); ENO:=LDF(EN,s); ENO:=ANDP(EN,s); ENO:=ANDF(EN,s); ENO:=ORP(EN,s); ENO:=ORF(EN,s);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Remarks	Range	Data type	Data type (label)
(s)	Device used as contact	—	Bit	ANY_BOOL

■Applicable devices

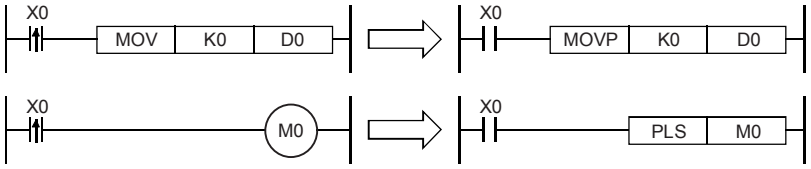
Operand	Bit	Word			Double word		Indirect specification	Constant			Others (DX)
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	—	○	—	—	—	—	—	○

Processing details

■LDP, LDF

- The LDP instruction is the rising edge pulse operation start instruction, and becomes conductive (ON) only at the rising edge (OFF to ON) of the specified bit device. When word devices are specified by bits, this instruction becomes conductive (ON) only when the status of the specified bit changes to 0→1. When only the LDP instruction is programmed, operation is the same as the conversion of the instruction under execution to pulse instruction (□P).

The following figure shows an example when a ladder using the LDP instruction is replaced with a ladder not using the LDP instruction.



- The LDF instruction is the falling edge pulse operation start instruction, and becomes conductive (ON) at the falling edge (ON to OFF) of the specified bit device. When word devices are specified by bits, this instruction becomes conductive only when the status of the specified bit changes to 1→0.

■ANDP, ANDF

- The ANDP instruction is the rising edge pulse series connection instruction, and the ANDF instruction is the falling edge pulse series connection. These instructions AND with the operation result so far, and uses the result as the operation result. The table below shows the ON/OFF information used by these instructions.

Device specified by ANDP, ANDF		ANDP status	ANDF status
Bit device	Bit specification of word device		
OFF to ON	0→1	ON	OFF
OFF	0	OFF	OFF
ON	1	OFF	OFF
ON to OFF	1→0	OFF	ON

■ORP, ORF

- The ORP instruction is the rising edge pulse parallel connection instruction, and the ORF instruction is the falling edge pulse parallel connection instruction. These instructions OR with the operation result so far, and use the result as the operation result. The table below shows the ON/OFF information used by these instructions.

Device specified by ORP, ORF		ORP status	ORF status
Bit device	Bit specification of word device		
OFF to ON	0→1	ON	OFF
OFF	0	OFF	OFF
ON	1	OFF	OFF
ON to OFF	1→0	OFF	ON

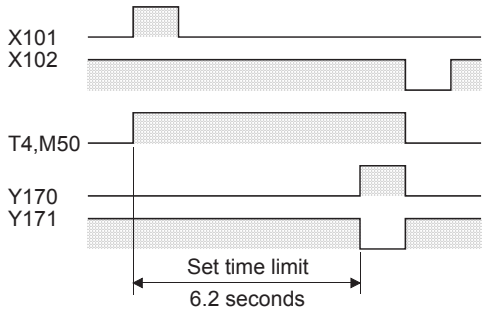
Operation error

There is no operation error.

Appendix 4.5 On delay timer (momentary input)

An on delay timer of a programmable controller operates easily with a continuous input. The internal relay (M) is used with a momentary input.

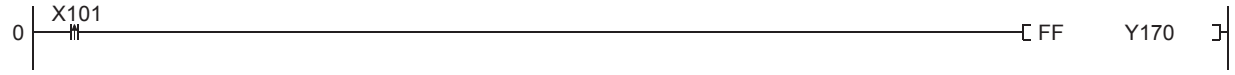
Project name	RA-19
Program name	MAIN



* The above ladder operates as an on delay timer by momentarily turning on the inputs X101 and X102.

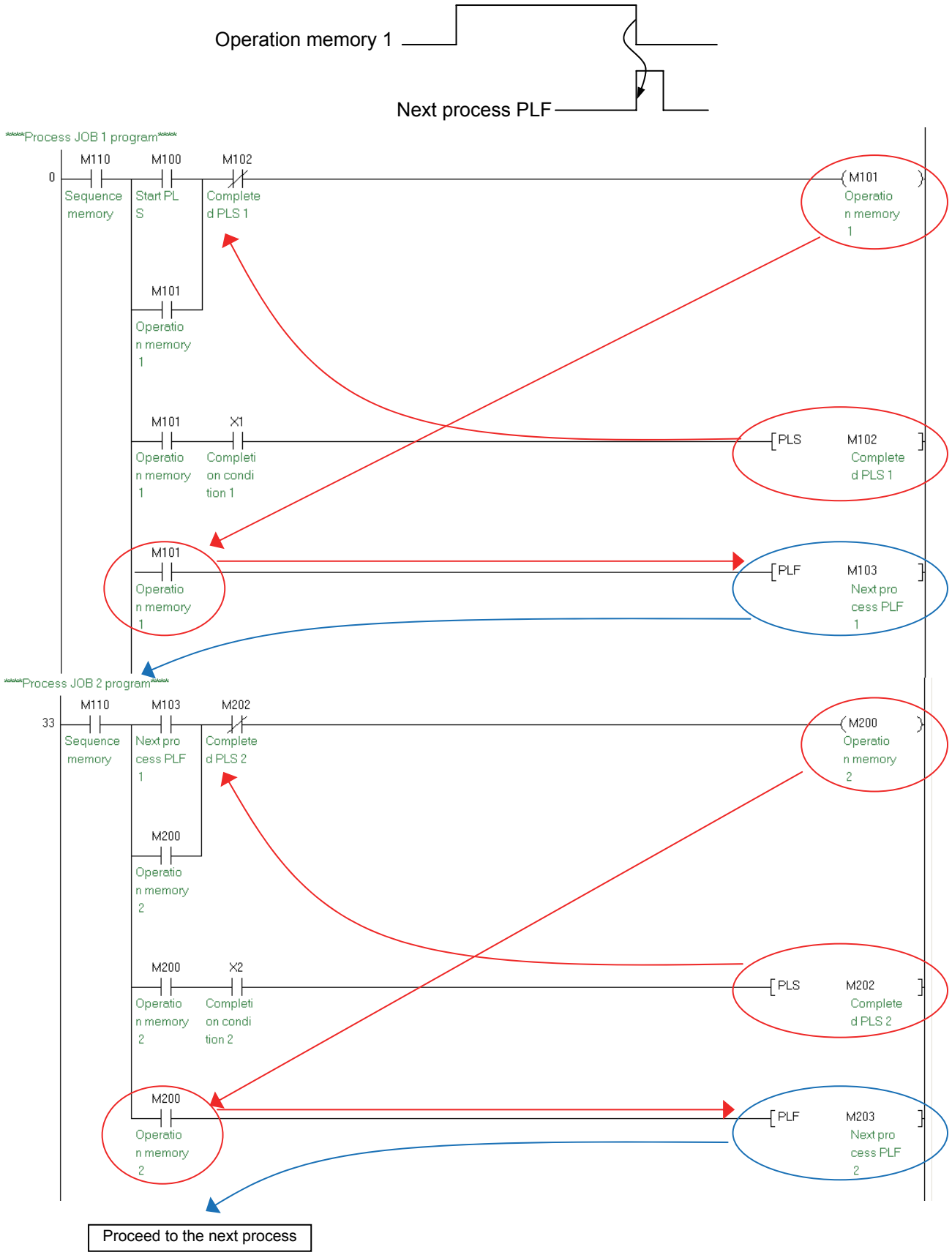
Appendix 4.6 On/off repeat ladder

In an on/off repeat ladder, Y170 turns on when X100 turns on for the first time, and turns off when X100 turns on again.



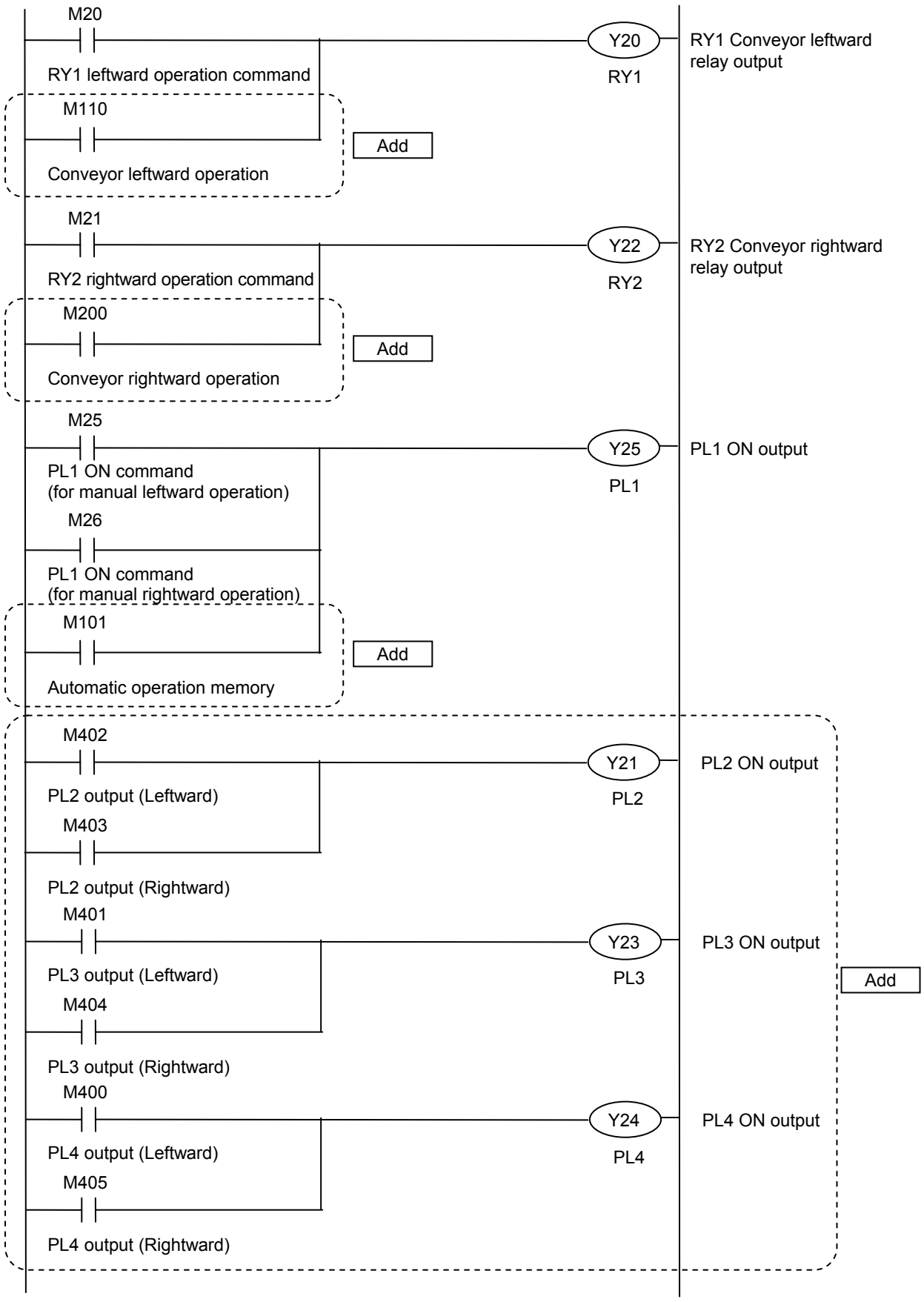
4.4.2 Sequence control example with self-holding function and PLF

When executing a control target (process: JOB1) 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 PLS1".
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".
The following shows the examples.



4.6.3 Output to device (additional)

Add auxiliary relays according to the device outputs.



5 BASIC INSTRUCTIONS -PART 2-

5.1 Notation of Values (Data)

Point

- This section describes decimal, binary, and hexadecimal notations.
- This section describes a method of interconversion.

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 MELSEC iQ-R series programmable controller, the symbol "K" is used to represent a value in decimal.

Binary (BIN)

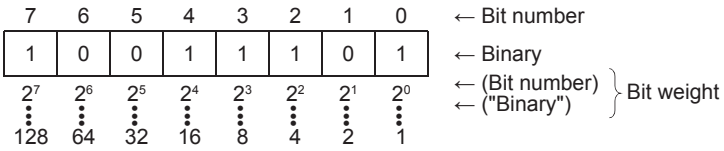
- A binary value consists of two symbols, 0 and 1, which represent the order and size (amount). After a digit reaches 1, an increment resets it to 0, causing an increment of the next digit to the left. One digit of 0 or 1 is called a bit.

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 × 128 + 0 × 64 + 0 × 32 + 1 × 16 + 1 × 8 + 1 × 4 + 0 × 2 + 1 × 1

  = 128 + 16 + 8 + 4 + 1

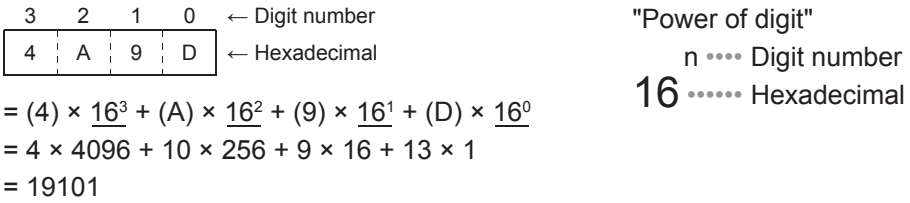
  = 157

  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	0
1	1	1
2	2	10
3	3	11
4	4	100
5	5	101
6	6	110
7	7	111
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"

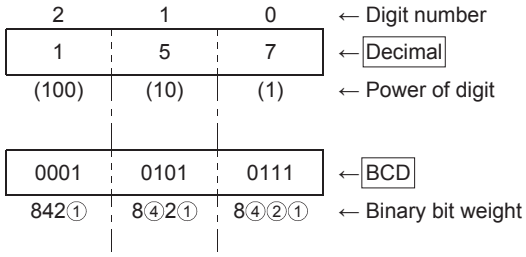
n •••• Digit number

16 •••••• Hexadecimal

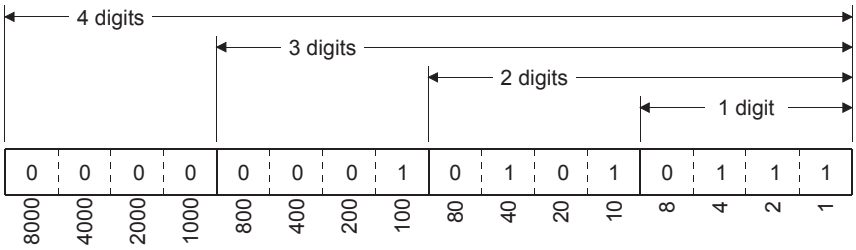
- Four bits of a binary value are equivalent to one digit of a hexadecimal value.
- In the MELSEC iQ-R series programmable controller, the symbol "H" is used to represent a value in hexadecimal.
- Hexadecimal values are used to represent the following device numbers.
  - Input and output (X, Y)
  - Function input and output (FX, FY)
  - Link relay (B)
  - Link register (W)
  - Link special relay (SB)
  - Link special register (SW)
  - Link direct device (Jn\X, Jn\Y, Jn\B, Jn\SB, Jn\W, Jn\SW)

Binary-coded decimal (BCD)

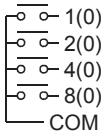
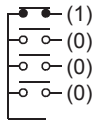
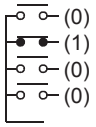
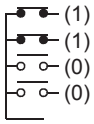
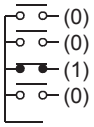
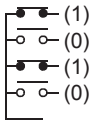
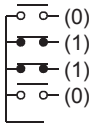
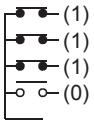
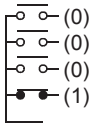
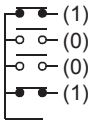
- The binary-coded decimal system uses a binary value to represent each digit of a decimal value.
  The decimal value 157, for example, is expressed as follows.



- 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.



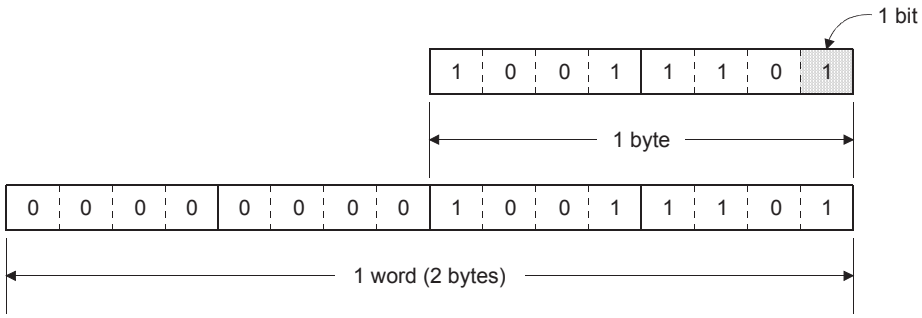
- BCD is used for the following signals.
  1) Output signals of digital switches
  2) Signals of seven-element display (digital HMI)

0	1	2	3	4	5	6	7	8	9
									

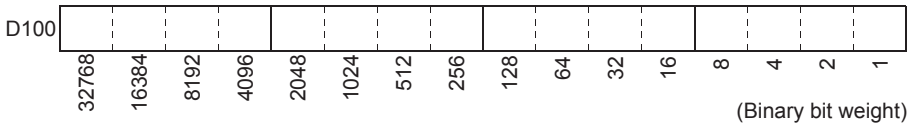
BCD code digital switch

Numerical values used by the MELSEC iQ-R series programmable controller

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



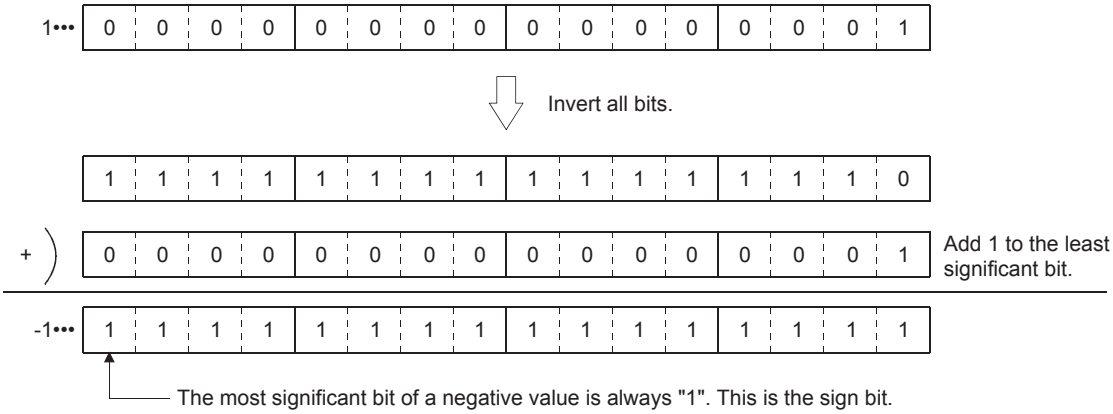
- Registers of each word device in the MELSEC iQ-R series programmable controller consist of 16 bits.
  - Data register (D)
  - Current value of a timer (T)
  - Current value of a counter (C)
  - File register (R)
  - Link register (W)



- 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 MELSEC iQ-R series programmable controller uses the range 2).
  A negative value uses the 2's complement against a positive number (1 to +32767).
- In the 2's complement, each binary bit is inverted, and 1 is added to the least significant bit.

Ex.

How to calculate the 2's complement against 1



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

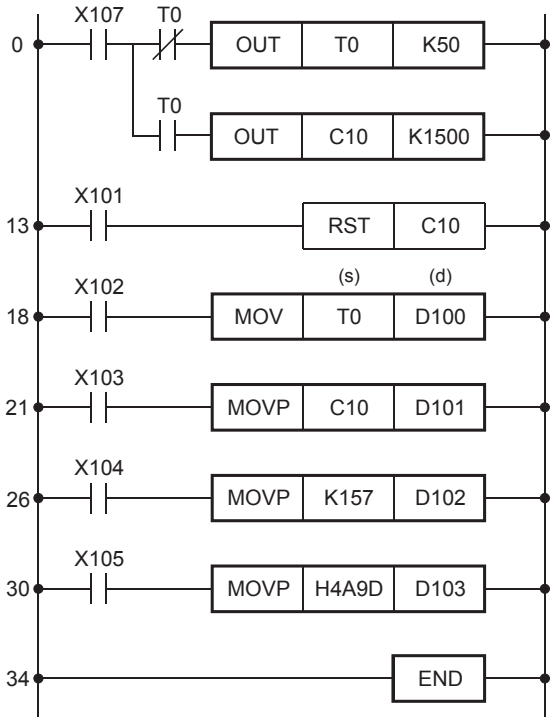
5.2 Transfer Instructions

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

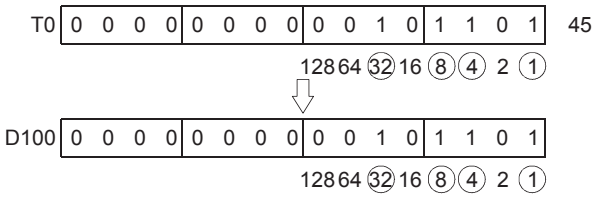
Point

- This section describes that data at the (s) side remains with the instruction for transferring data from the (s) side to the (d) side.
- This section describes the operation differences between the instructions with P and the one without P.

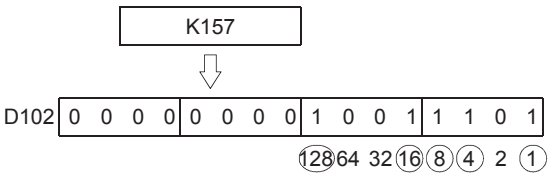
Project name	RB-11
Program name	MAIN



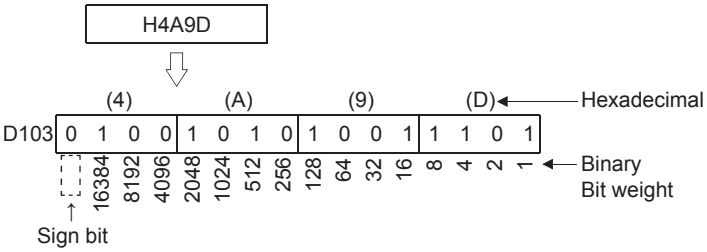
- When the input condition turns on, the current value of the timer T0 (source) is transferred into the data register D100 (destination).
- The current value of T0 in binary is transferred into D100 as it is. (Data conversion is not performed.)



- 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.

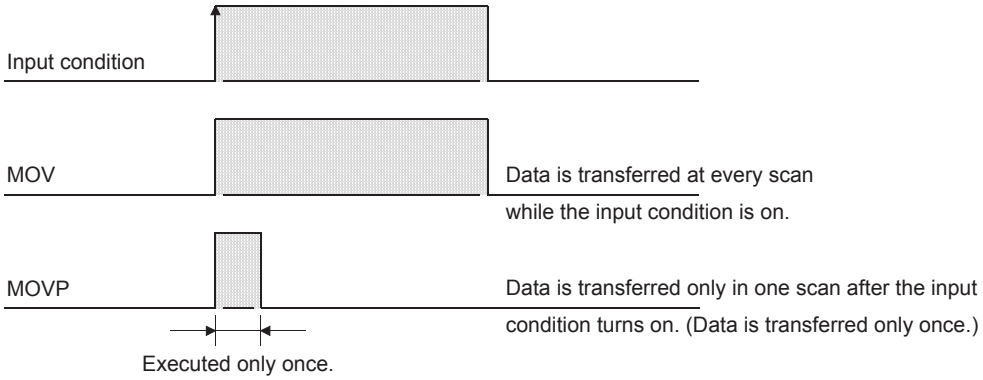


• When the input condition turns on, the hexadecimal value 4A9D is converted into a binary value and transferred into the data register D103.



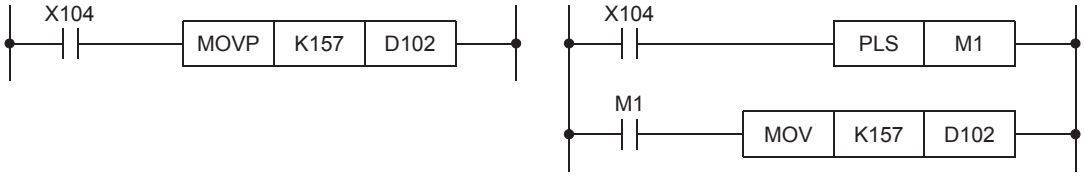
Differences between MOV and MOVP

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.

• Both of the following ladder programs function similarly.



Operand	Bit		Word			Double word		Indirect specification	Constant			Others	Number of basic steps
	X, Y, M, L, SM, F, B, SB, FX, FY	J□\□	T, ST, C, D, W, SD,SW, FD, R, ZR, RD	U□\G□, J□\□, U3E□\(H)G□	Z	LT, LST, LC	LZ		K, H	E	$		
(s)	○	—	○	○	○	—	—	○	○	—	—	—	*1
(d)	○	—	○	○	○	—	—	○	—	—	—	—	

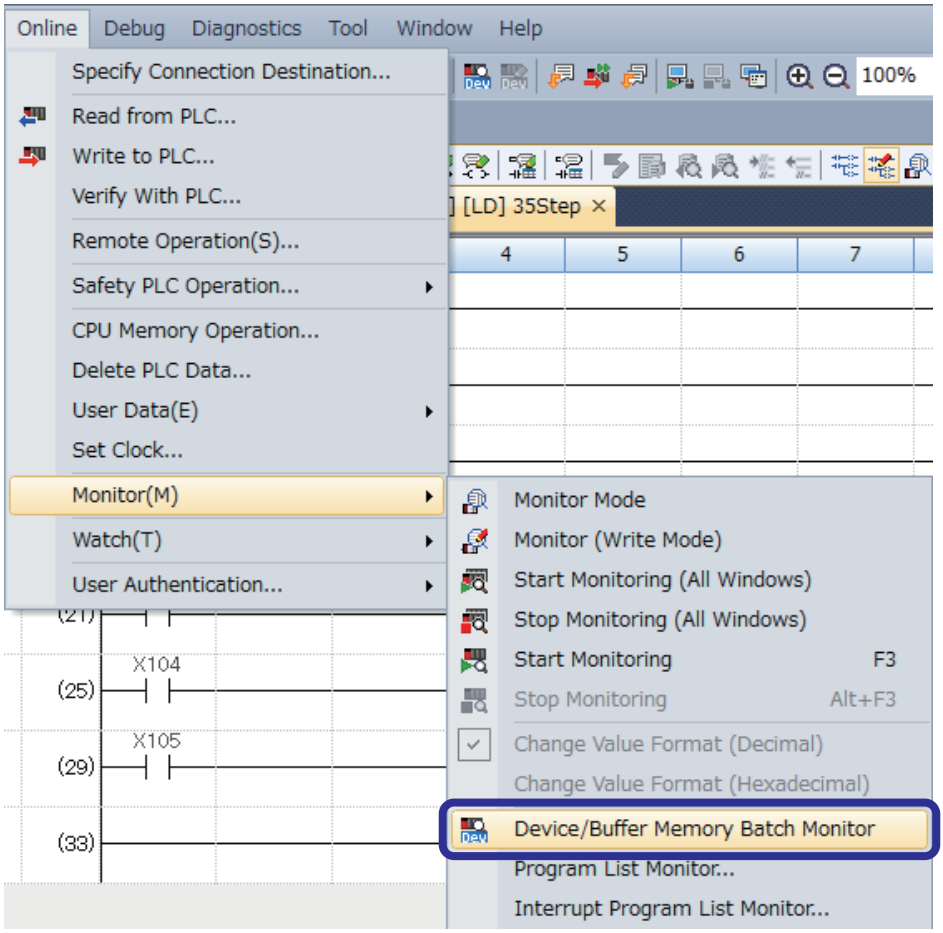
*1 The number of steps varies depending on the devices to be used.

Items to be checked

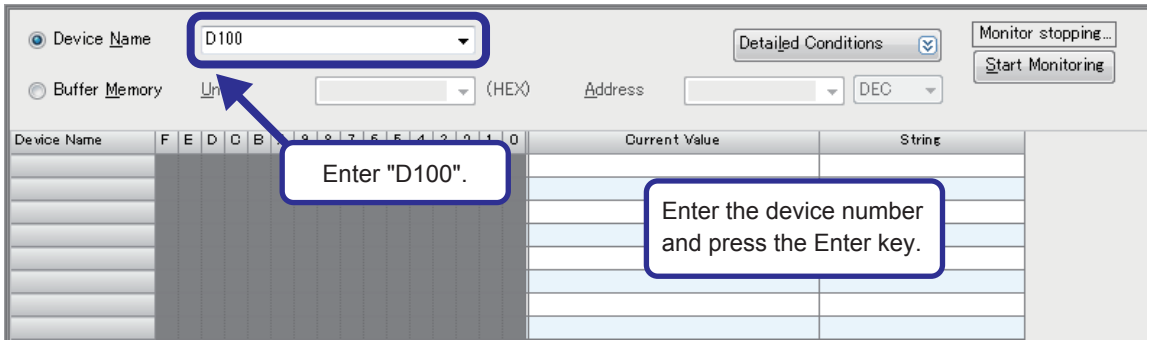
CPU module: RUN

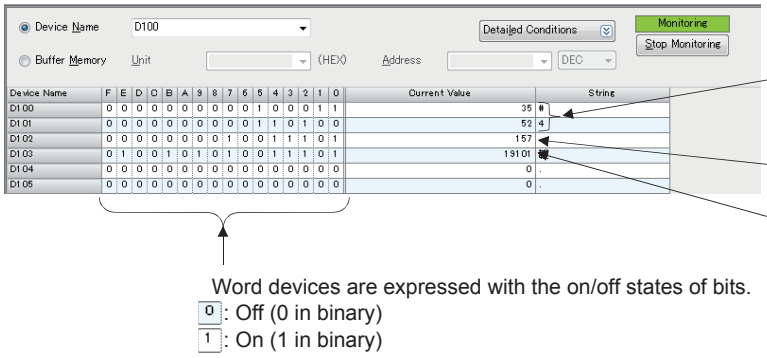
Inputs X102, X103, X104, X105, and X107: On

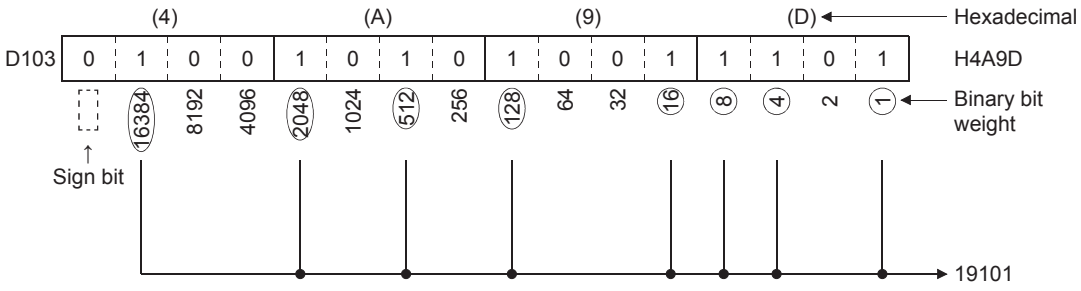
- Monitor the values in the data register (D100 to D103).
  After writing data to the programmable controller, click [Online] → [Monitor] → [Device/Buffer Memory Batch Monitor].
  The "Device/Buffer Memory Batch Monitor" dialog box appears.



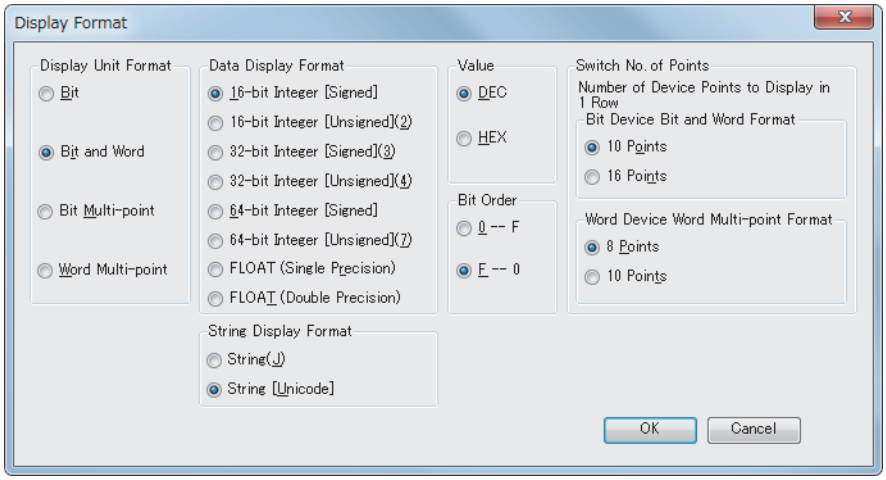
- Enter "D100" in "Device Name" and press the Enter key.



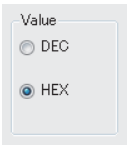

Device Name
D100
Detailed Conditions
Monitoring
Stop Monitoring
Buffer Memory
Unit
(HEX)
Address
DEC
Device Name
Current Value
String
D100
D101
D102
D103
D104
D105
35
52
157
19101
0
Current values of a timer and counter are monitored. (The values change.)
This value indicates that a decimal number 157 (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)


(4)
(A)
(9)
(D)
Hexadecimal
D103
0 1 0 0 1 0 1 0 1 0 0 1 1 1 0 1
H4A9D
16384
8192
4096
2048
1024
512
256
128
64
32
16
8
4
2
1
Binary bit weight
Sign bit
19101

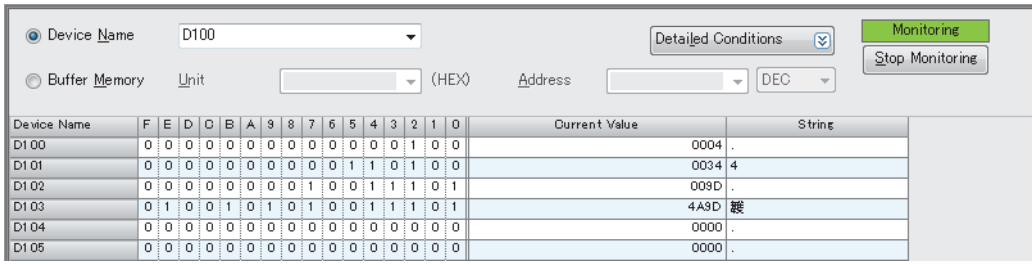
- Click on the toolbar or select [View] → [Display Format Detailed Setting] from the menu.
  The "Display Format" dialog box appears.



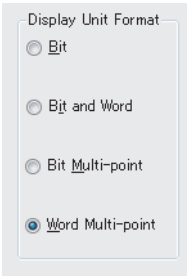
- Change the display of the numerical values being monitored to the hexadecimal notation.
  Select "HEX" for "Value" in the "Device Format" dialog box.



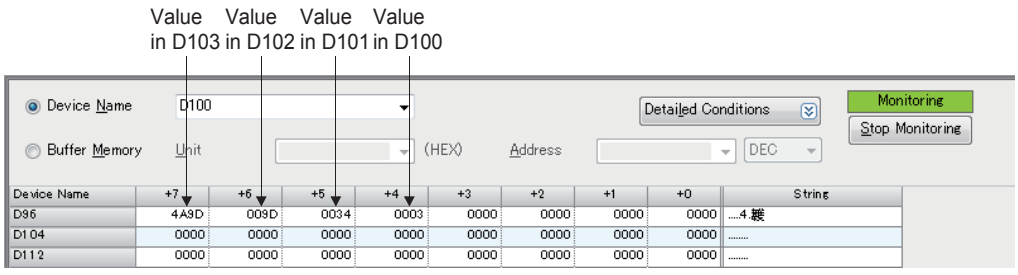
["Device/Buffer Memory Batch Monitor" window]



- 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.

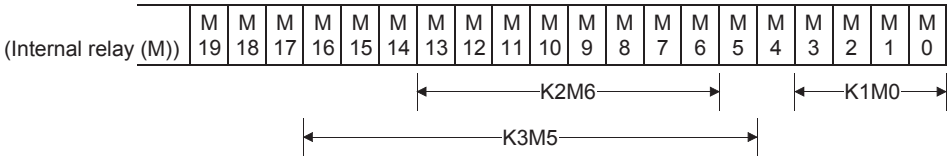


["Device/Buffer Memory Batch Monitor" window]



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.

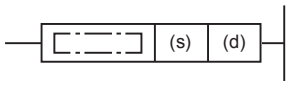
7.5 Data Conversion Instructions

Converting binary data to BCD 4-digit data

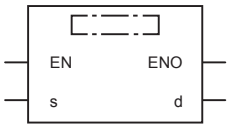
BCD(P)

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	Structured text
	ENO:=BCD(EN,s,d); ENO:=BCDP(EN,s,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

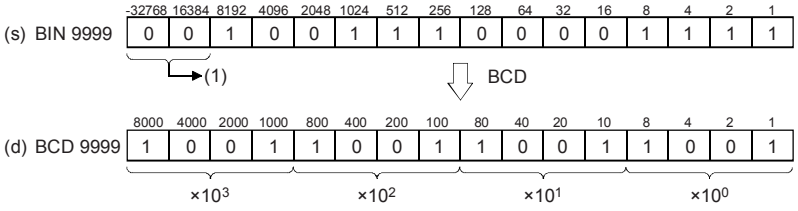
Operand	Description	Range	Data type	Data type (label)
(s)	Binary data or the head device where the binary data is stored	0 to 9999	16-bit signed binary	ANY16
(d)	Head device for storing the BCD data	—	BCD 4-digit	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—

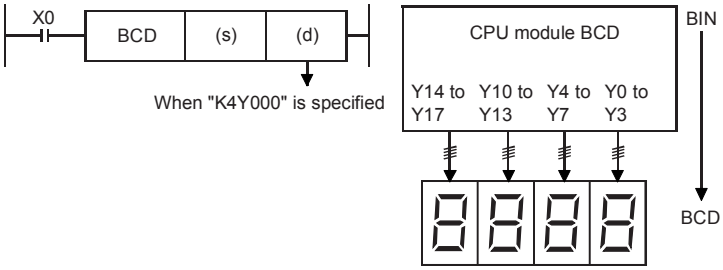
Processing details

- 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

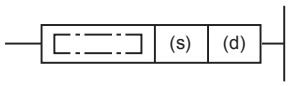
Error code (SD0/SD8067)	Description
3401H	Data in the device specified by (s) is out of the valid range (0 to 9999).

Converting BCD 4-digit data to binary data

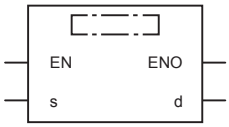
BIN(P)

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	Structured text
	ENO:=BIN(EN,s,d); ENO:=BINP(EN,s,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

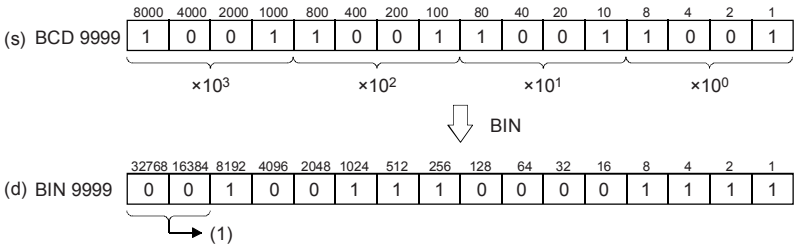
Operand	Description	Range	Data type	Data type (label)
(s)	Binary-coded decimal data or the head device where the binary-coded decimal data is stored	0 to 9999	BCD 4-digit	ANY16
(d)	Head device for storing the binary data	—	16-bit signed binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—

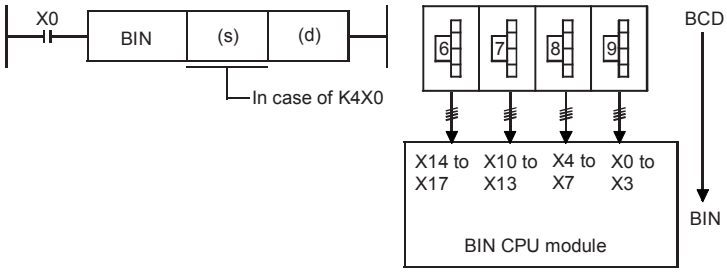
Processing details

- 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

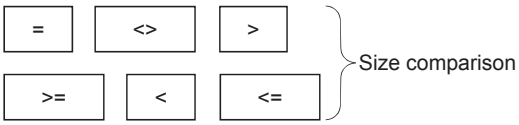
Error code (SD0/SD8067)	Description
3401H	The value of each digit of the device specified by (s) is other than 0 to 9. (The data is not binary-coded decimal data.)

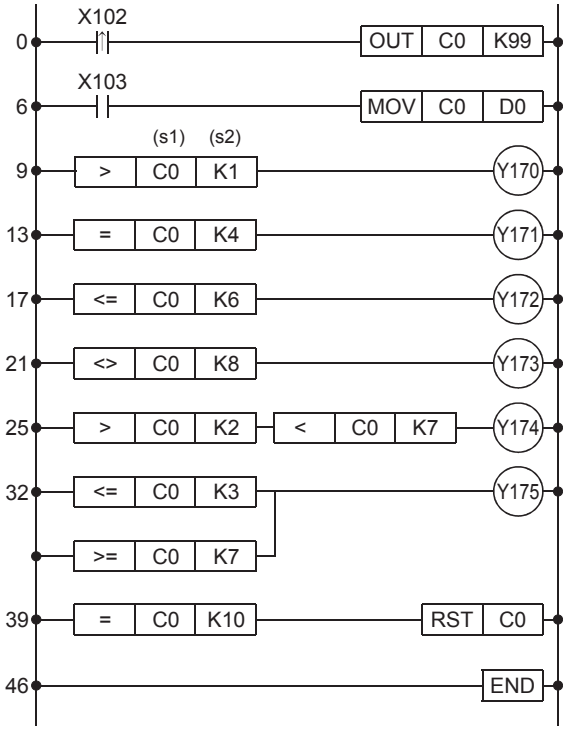
5.3 Comparison Operation Instructions

Point

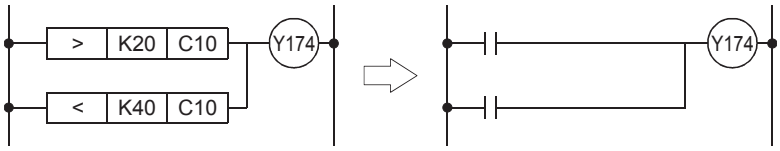
This section describes how to compare numerical values.

Project name	RB-15
Program name	MAIN





- 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.

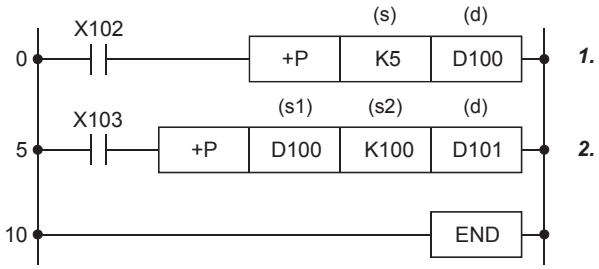
5.4 Arithmetic Operation Instructions

5.4.1 [+(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.

Project name	RB-16
Program name	MAIN



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.

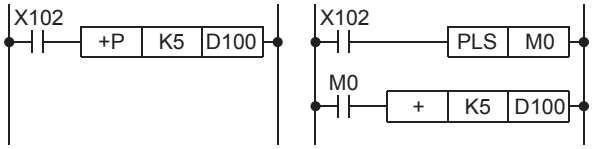
2. When the input condition turns on, the value in the device specified in (s1) is added to the value in the device specified in (s2), and the result is stored in the device specified in (d).

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

The value in D100 is not changed by the addition.

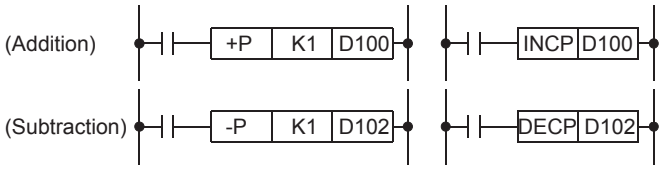
Precautions

- 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.

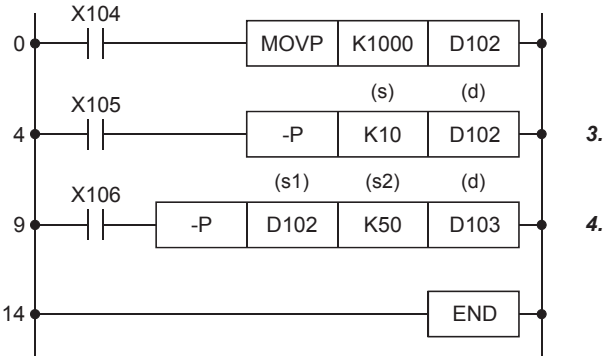


Point

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



Project name	RB-17
Program name	MAIN



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.

Operand	Bit		Word			Double word		Indirect specification	Constant			Others	Number of basic steps
	X, Y, M, L, SM, F, B, SB, FX, FY	J□\□	T, ST, C, D, W, SD,SW, FD, R, ZR, RD	U□\G□, J□\□, U3E□\(H)G□	Z	LT, LST, LC	LZ		K, H	E	$		
(s1)	○	—	○	○	○	—	—	○	○	—	—	—	3 or 4*1
(s2)	○	—	○	○	○	—	—	○	○	—	—	—	
(d)	○	—	○	○	○	—	—	○	—	—	—	—	

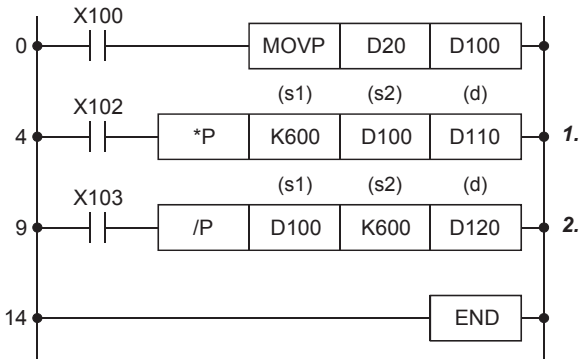
*1 The number of basic steps is four for □ (s1) (s2) (d).

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

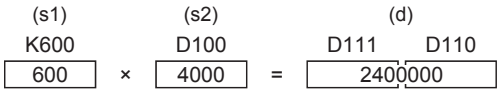
Point

- This section describes multiplication or division.
- This section describes the concept of two words.

Project name	RB-18
Program name	MAIN



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).

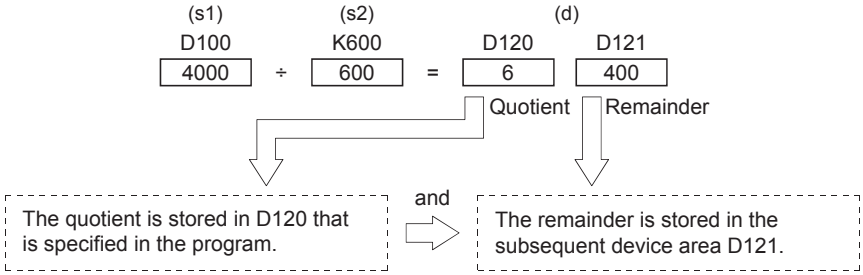


To store the result of 16-bit data × 16-bit data, a space of 16 bits (1 word) is not enough. Thus, the D110 specified in the program and the subsequent device area D111 are used for storing the result.

Since these device areas are used as a 32-bit (2-word) register for storing the result, the left-most bit of D110 (b15) is regarded as a part of the data, not as a sign bit.

When programming a ladder using the operation result of the *(P) instruction, always use 32-bit instructions (such as the DMOV instruction and the DMOVP instruction).

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

■Operation practice

❶ Write the program to the CPU module and set the operating status of the CPU module to RUN.

❷ Turn on X100 and store the value of the initial input device D20 in D100.

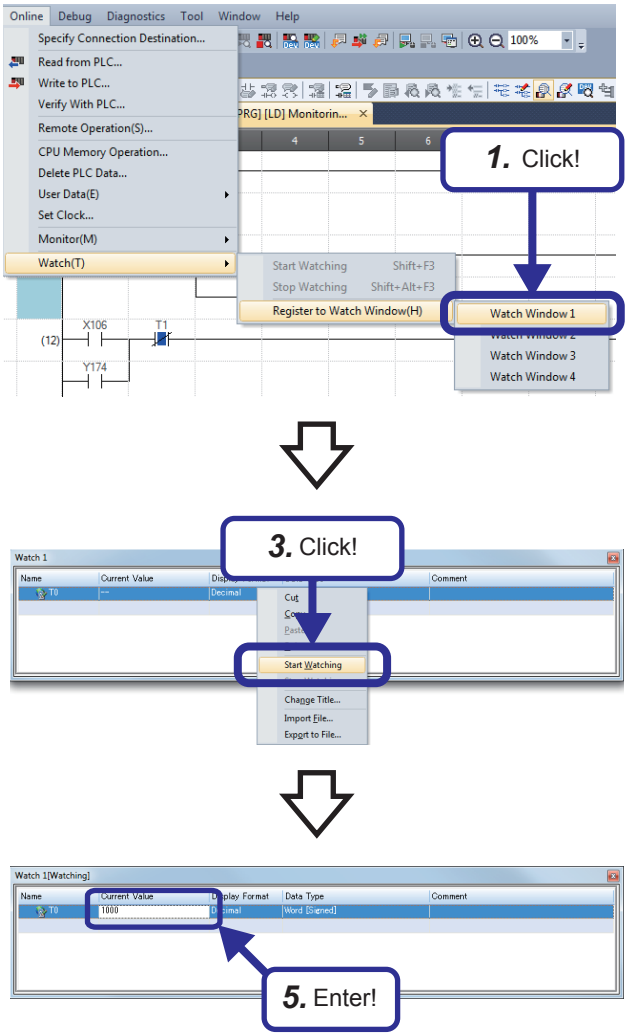
❸ Turn on X102. The multiplication of 600 × D100 is executed.

❹ Turn on X103. The division of D100 ÷ 600 is executed.

Operand	Bit		Word			Double word		Indirect specification	Constant			Others	Number of basic steps
	X, Y, M, L, SM, F, B, SB, FX, FY	J□\□	T, ST, C, D, W, SD,SW, FD, R, ZR, RD	U□\G□, J□\□, U3E□\(H)G□	Z	LT, LST, LC	LZ		K, H	E	$		
(s1)	○	—	○	○	○	—	—	○	○	—	—	—	Multiplication instruction: 3 or 4*1 Division instruction: 4
(s2)	○	—	○	○	○	—	—	○	○	—	—	—	
(d)	○	—	○	○	○	—	—	○	—	—	—	—	

*1 The number of steps in a multiplication instruction varies depending on the devices to be used.

6.1.3 Current value change of the device (D20)



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

1. Select "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 "Watch 1" window and click [Start Watching].

4. Watching of "Watch 1" starts.

5. Enter "1000" in "Current Value".

Checking with the demonstration machine

After entering the current value, press the Enter key and check that the value in the initial indication device D20 changes to 1000.

Point

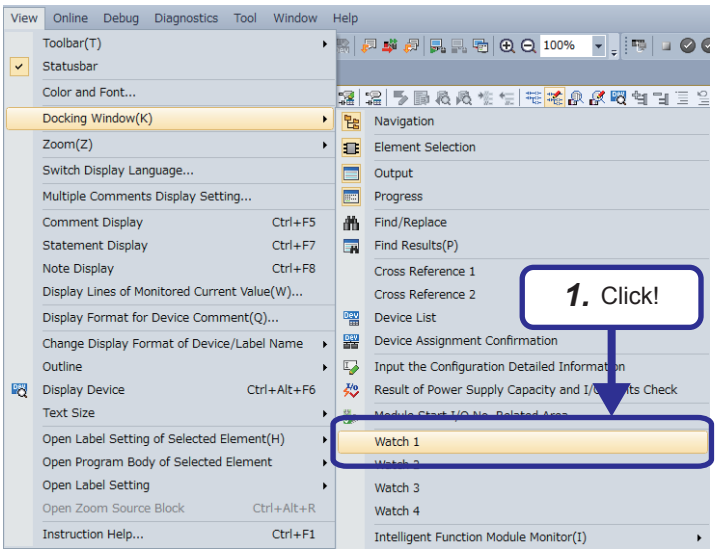
With the same procedure, the current values of word devices can also be changed.

6.5 Watch Window

Point

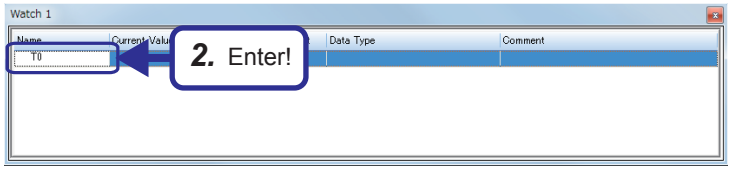
This section describes the "Watch" window where devices can be checked at once.

This section describes how to register multiple devices or labels in one window and to monitor them at the same time.

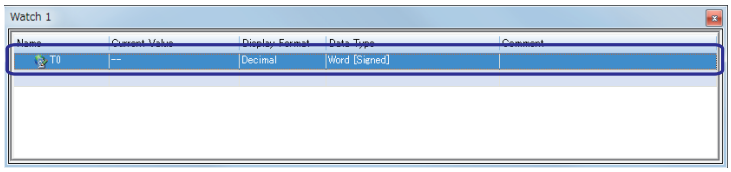


1. Click [View] → [Docking Window] → one of [Watch 1] to [Watch 4] from the menu.

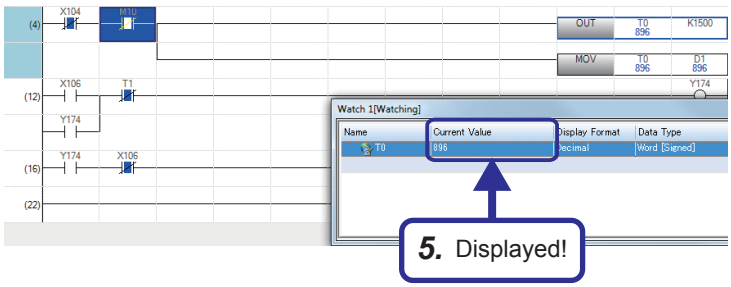
* In this example, select [Watch 1].



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



3. The device or label is registered.



4. Click [Online] → [Watch] → [Start Watching] from the menu.

5. The current value of the registered device or label is displayed in the window.

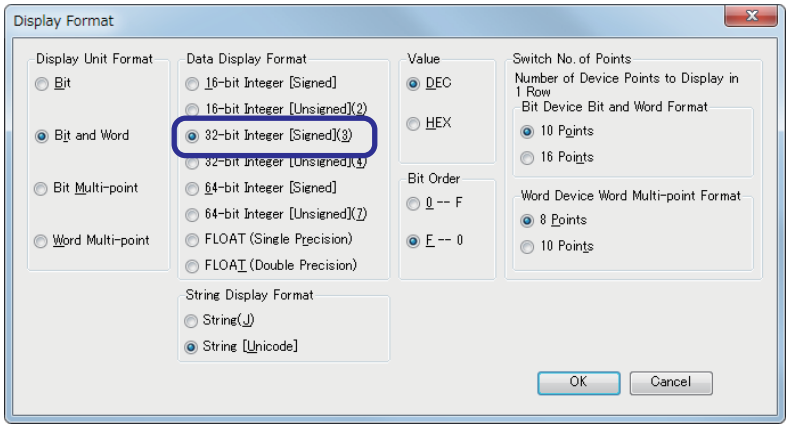
Point

How to monitor 32-bit integer data

When the operation result of the multiplication instruction is outside the range of 0 to 32767, the result cannot be properly displayed even though the value is regarded as a 16-bit integer and the values in the lower registers are monitored in ladder.

To monitor the values properly, follow the procedure below.

- Click [View] → [Display Format Detailed Setting] from the menu to display the "Display Format" dialog box. Then, 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

5.4.3 32-bit data instructions and their necessities

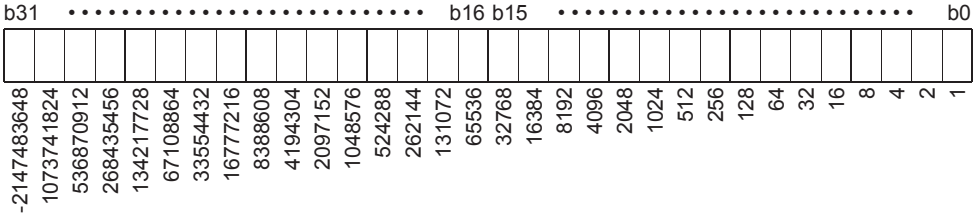
Point

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

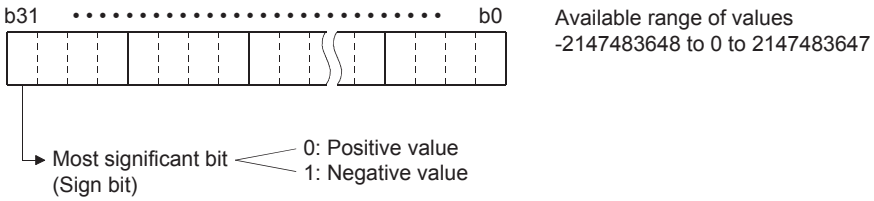
- The unit of the data memory of the MELSEC iQ-R series programmable controller is one word that consists of 16 bits. Thus, data is typically processed in units of one word at the transfer processing, comparison, and arithmetic operation.
- The MELSEC iQ-R series programmable controller can process data 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.

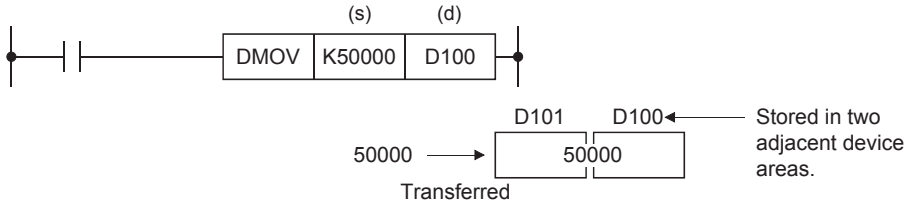


As the case of 16-bit data processing, the programmable controller takes the 2's complement in 32-bit data processing. Thus, the most significant bit b31 (b15 for 16-bit data) is processed as a sign bit.

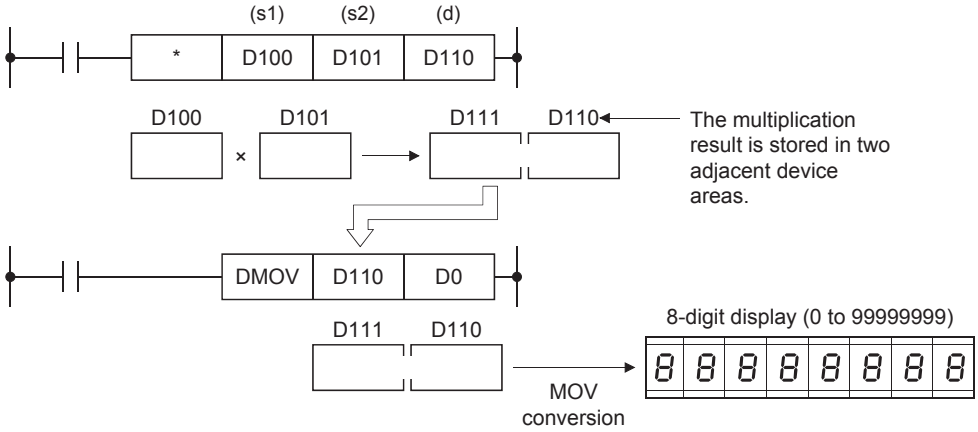


• Whether data is processed as two-word (32-bit) data or not depends on the size of the data.
In the following cases, use two-word instructions.

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

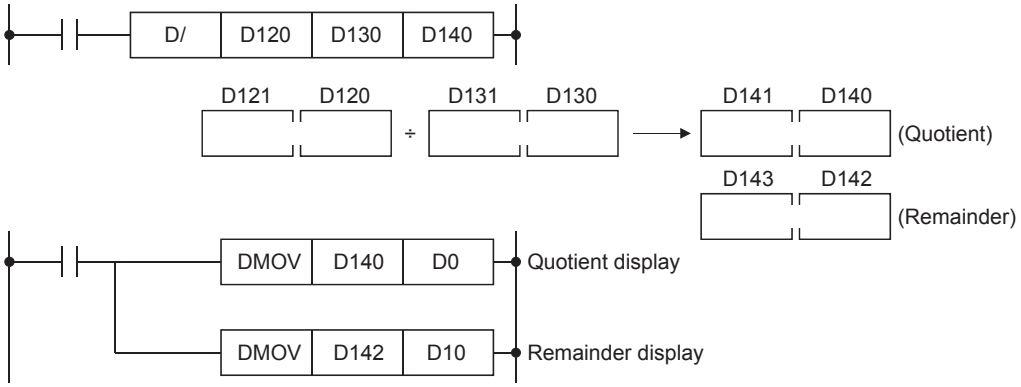


(2) When the result of the 16-bit multiplication instruction (one-word instruction) is transferred



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

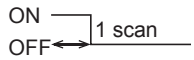
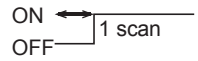
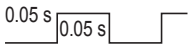
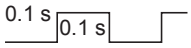
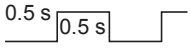
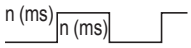
(3) When the result of the 32-bit division instruction is used



System clock

The special relay about system clock is shown below.

R: Read only, R/W: Read/Write

No.	Name	Description	R/W
SM400	Always ON		R
SM401	Always OFF		R
SM402	After RUN, ON for one scan only		R
SM403	After RUN, OFF for one scan only		R
SM409	0.01 second clock		R
SM410	0.1 second clock		R
SM411	0.2 second clock		R
SM412	1 second clock		R
SM413	2 second clock		R
SM414	2n second clock		R
SM415	2n ms clock		R
SM420	Timing clock output 1		R
SM421	Timing clock output 2		R
SM422	Timing clock output 3		R
SM423	Timing clock output 4		R
SM424	Timing clock output 5		R

A

7 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.

7.1 Time Setting

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.

Clock data

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

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

Changing the clock data

The clock data can be changed using the following methods.

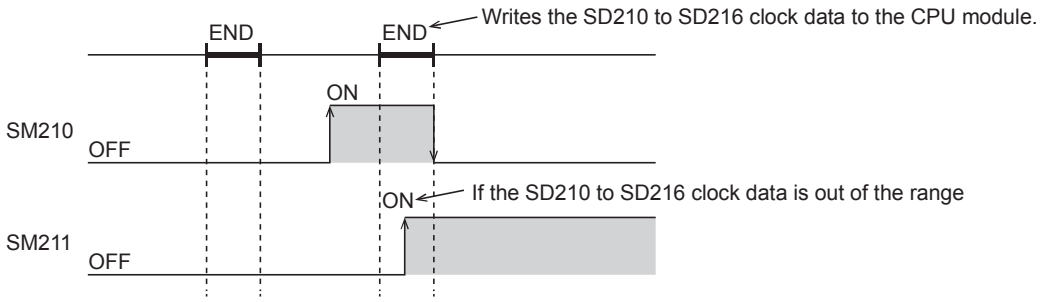
- Using engineering tools
- Using SM/SD
- Using instructions

Using the engineering tool

Clock data can be changed using Set Clock from the menu. (GX Works3 Operating Manual)

Using SM/SD

The values stored in SD210 (clock data) to SD216 (clock data) are written to the CPU module after END processing execution of scan when SM210 (clock data set request) is changed from OFF→ON. If the data from SD210 to SD216 is out of the valid range, SM211 (clock data set error) is turned ON, the values from SD210 to SD216 are not written in the CPU module.



Using instructions

Writes the clock data to the CPU module, using the TWR(P) instruction. (MELSEC iQ-F FX5 Programming Manual (Instructions, Standard Functions/Function Blocks))

System information

The special registers for system information are shown below.

R: Read only, R/W: Read/Write

No.	Name	Description	R/W
SD200	Switch Status	This register stores the CPU switch status. 0: RUN 1: STOP	R
SD201	LED Status	This register stores the LED status. b0: STOP b4: RUN b5: PAUSE	R
SD203	CPU Status	This register stores the CPU Status. 0: RUN 2: STOP 3: PAUSE	R
SD210	Clock Data (Year)	This register stores the clock data (Year).	R/W
SD211	Clock Data (Month)	This register stores the clock data (Month).	R/W
SD212	Clock Data (Day)	This register stores the clock data (Day).	R/W
SD213	Clock Data (Hour)	This register stores the clock data (Hour).	R/W
SD214	Clock Data (Minute)	This register stores the clock data (Minute).	R/W
SD215	Clock Data (Second)	This register stores the clock data (Second).	R/W
SD216	Clock Data (Day Week)	This register stores the clock data (Day of the Week).	R/W
SD218	Time zone setting value	The time zone setting value specified in the parameter is stored in increments of minutes.	R
SD250	Loaded Max I/O	This register stores high-order 2 digits of the final I/O number of connected modules +1 in 8-bit binary.	R
SD260	X Device Size [Lower]	This register stores the number of X device points used as 32-bit value.	R
SD261	X Device Size [Upper]		
SD262	Y Device Size [Lower]	This register stores the number of Y device points used as 32-bit value.	R
SD263	Y Device Size [Upper]		
SD264	M Device Size [Lower]	This register stores the number of M device points used as 32-bit value.	R
SD265	M Device Size [Upper]		
SD266	B Device Size [Lower]	This register stores the number of B device points used as 32-bit value.	R
SD267	B Device Size [Upper]		
SD268	SB Device Size [Lower]	This register stores the number of SB device points used as 32-bit value.	R
SD269	SB Device Size [Upper]		
SD270	F Device Size [Lower]	This register stores the number of F device points used as 32-bit value.	R
SD271	F Device Size [Upper]		
SD274	L Device Size [Lower]	This register stores the number of L device points used as 32-bit value.	R
SD275	L Device Size [Upper]		
SD280	D Device Size [Lower]	This register stores the number of D device points used as 32-bit value.	R
SD281	D Device Size [Upper]		
SD282	W Device Size [Lower]	This register stores the number of W device points used as 32-bit value.	R
SD283	W Device Size [Upper]		
SD284	SW Device Size [Lower]	This register stores the number of SW device points used as 32-bit value.	R
SD285	SW Device Size [Upper]		
SD288	T Device Size [Lower]	This register stores the number of T device points used as 32-bit value.	R
SD289	T Device Size [Upper]		
SD290	ST Device Size [Lower]	This register stores the number of ST device points used as 32-bit value.	R
SD291	ST Device Size [Upper]		
SD292	C Device Size [Lower]	This register stores the number of C device points used as 32-bit value.	R
SD293	C Device Size [Upper]		
SD298	LC Device Size [Lower]	This register stores the number of LC device points used as 32-bit value.	R
SD299	LC Device Size [Upper]		
SD300	Z Device Size	This register stores the number of Z device points used.	R

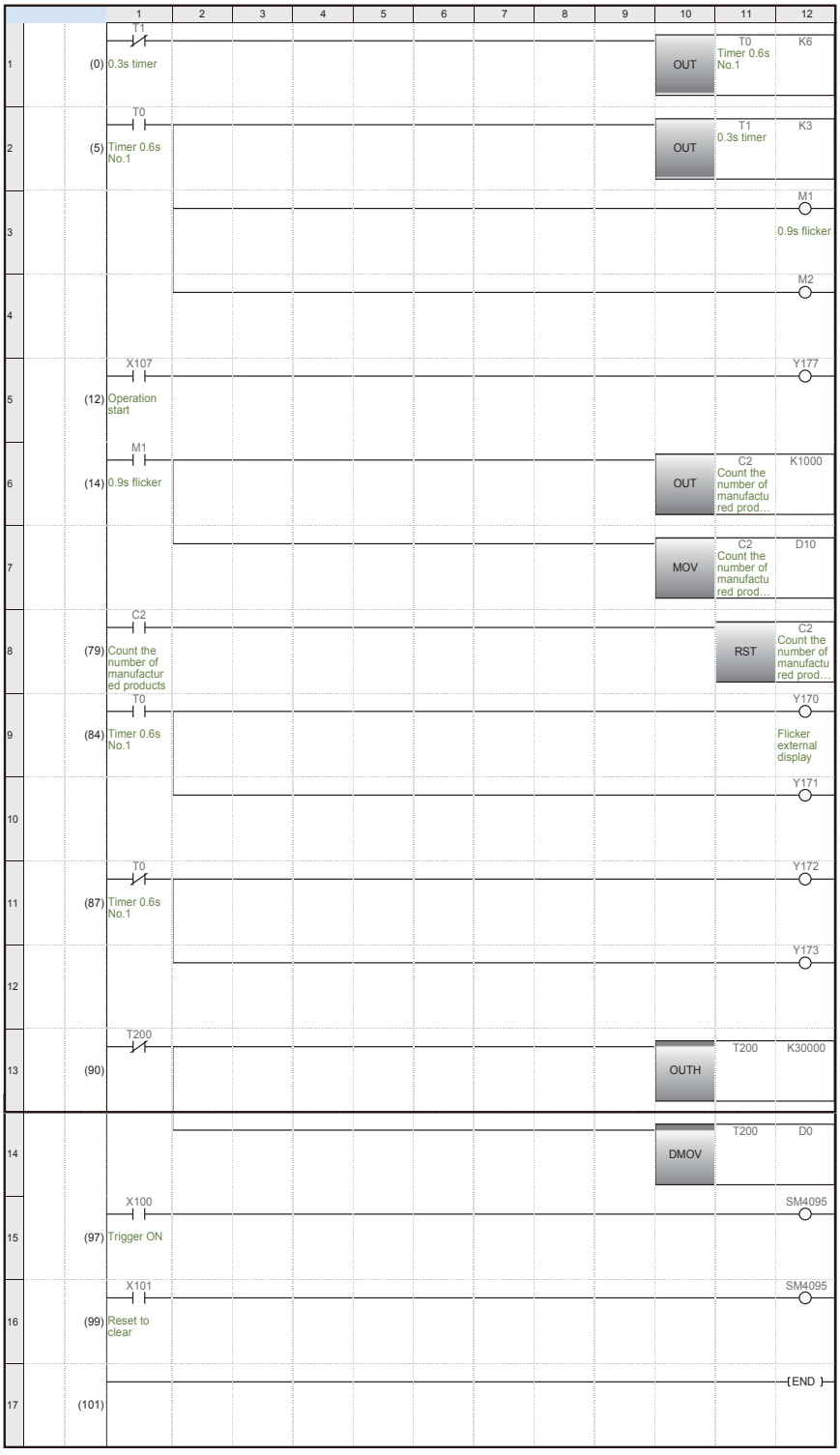
6.6 How to Create Comments

Point

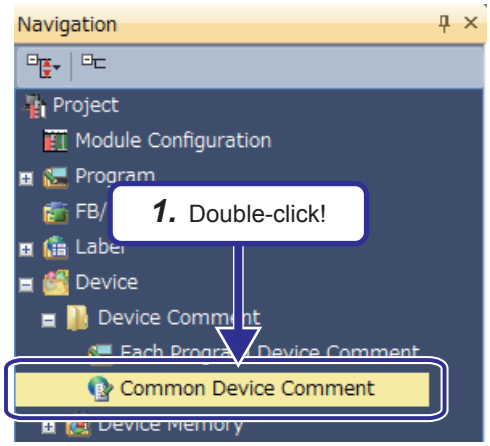
This section describes how to create comments (device comments, statements, and notes) in a program.

Project name	REX15
Program name	MAIN

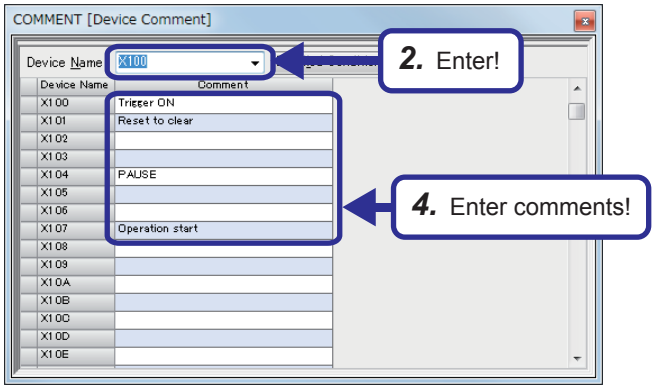
Example of a printed ladder program with comments



Creating comments



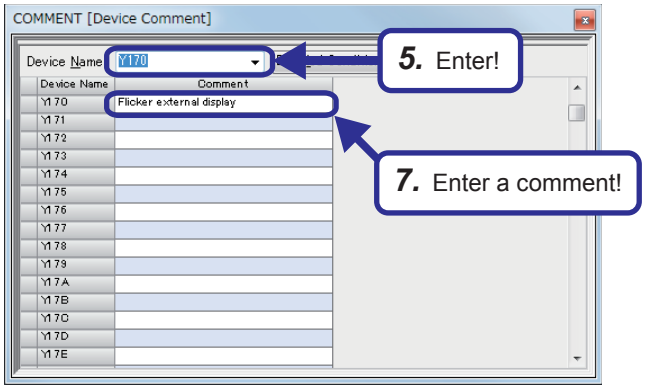
1. Click [Device] → [Device Comment] in the "Project" view and double-click [Common Device Comment] to display the "Device Comment" window.



2. Click "Device Name" and enter "X100" in the list box.

3. Press the Enter key.

4. Click each of comment areas and enter comments as shown on the left.



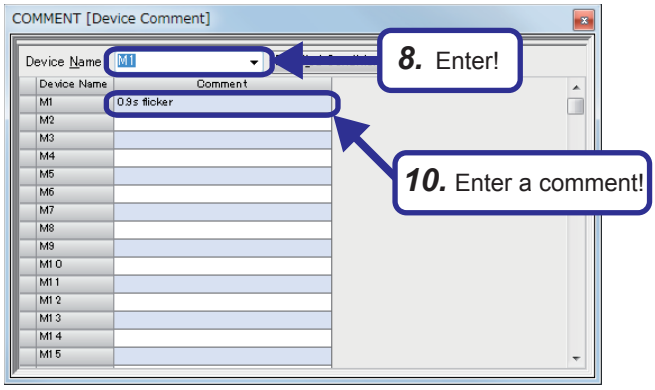
5. Click "Device Name" and enter "Y170" in the list box.

6. Press the Enter key.

7. Click a comment area and enter a comment as shown on the left.

(To the next page)

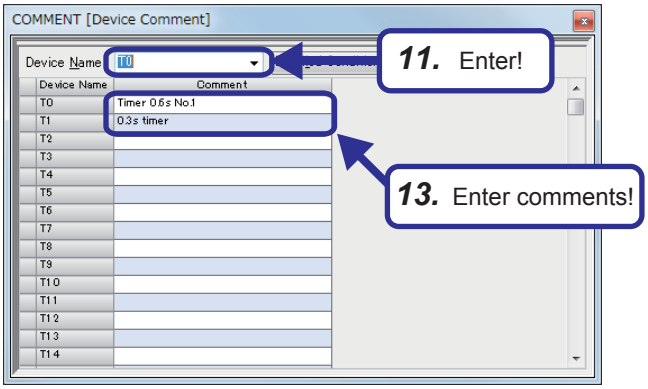
(From the previous page)



8. Click "Device Name" and enter "M1" in the list box.

9. Press the Enter key.

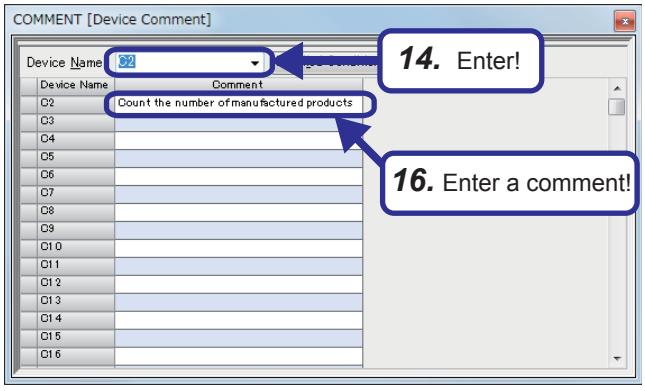
10. Click a comment area and enter a comment as shown on the left.



11. Click "Device Name" and enter "T0" in the list box.

12. Press the Enter key.

13. Click each of comment areas and enter comments as shown on the left.



14. Click "Device Name" and enter "C2" in the list box.

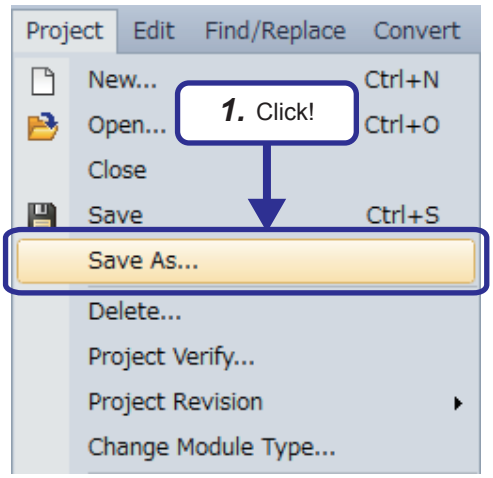
15. Press the Enter key.

16. Click a comment area and enter a comment as shown on the left.

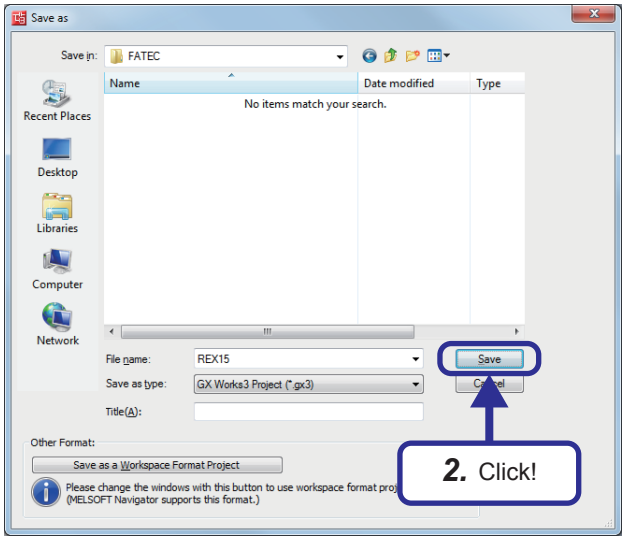
Point

Comments are used for indicating the function or application of each device. Up to 1024 characters can be entered in a comment.

Saving comments

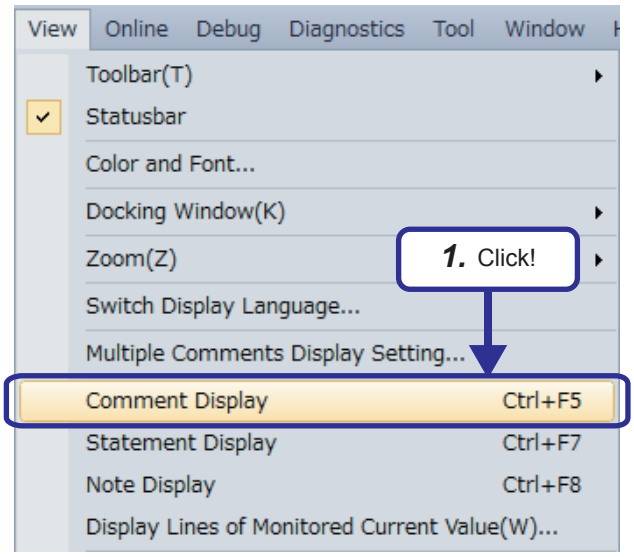


1. Click [Project] → [Save As] from the menu.



2. The "Save as" dialog box appears. Specify the save destination and a project name and click the [Save] button.

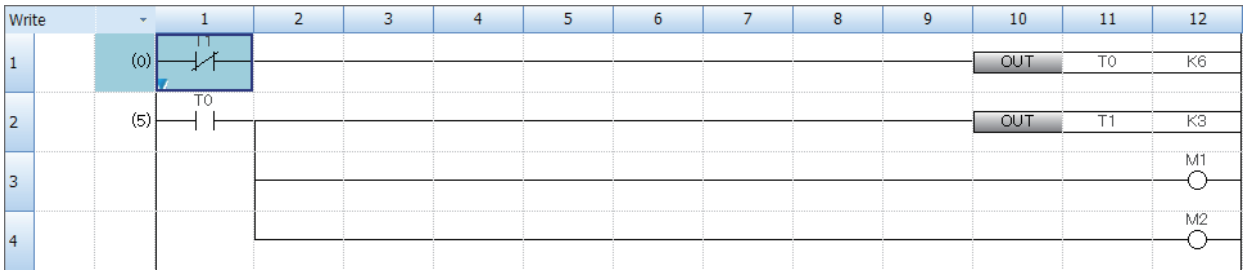
Displaying a ladder with comments in windows of GX Works3



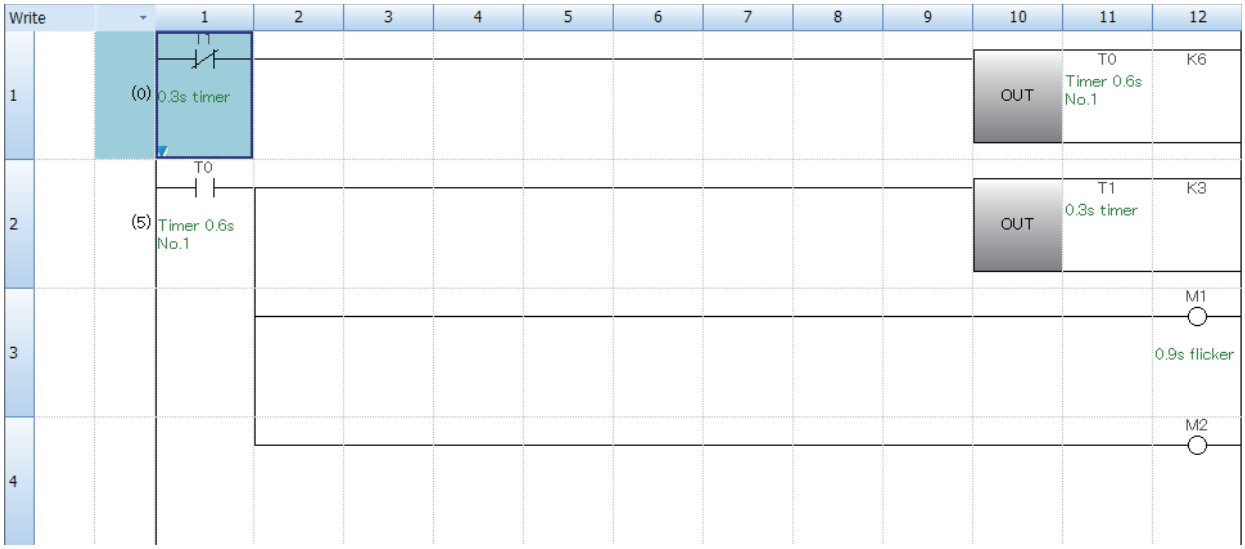
1. Click [View] → [Comment Display] from the menu.

2. The ladder program is displayed with comments.

(When comments are not displayed)



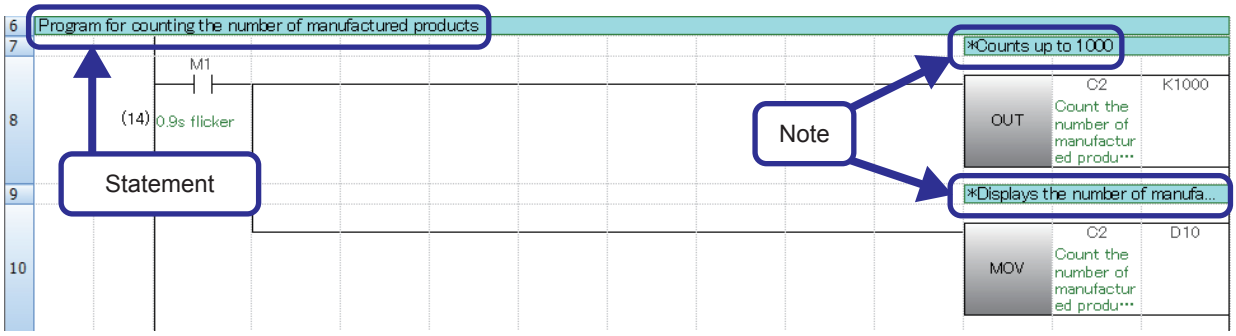
(When comments are displayed)



Point

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

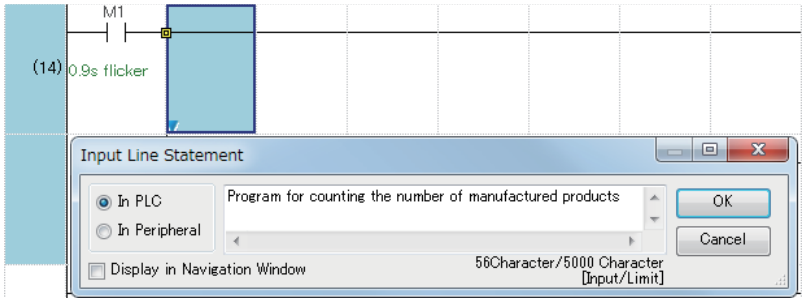
- Statement: A comment that describes the function or application of a ladder block
- Note: A comment that describes the function or application of an output or instruction



- Creating statements

  Click and double-click a ladder block where a comment is to be created.

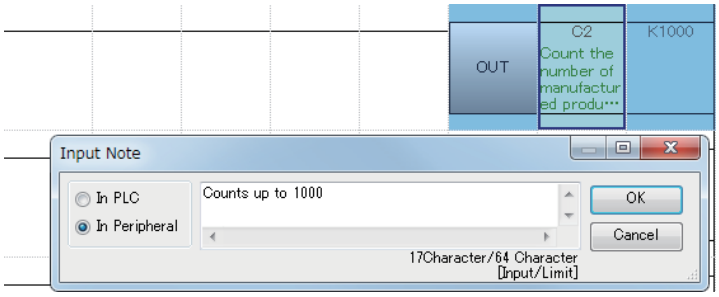
  The "Input Line Statement" dialog box appears. Enter a comment and click the [OK] button.



- Creating notes

  Click 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] button.



- Statements and notes are classified into two categories: "In PLC" and "In Peripheral".

Category	Type	Description
In PLC	• Line statement • P statement • I statement • Note	Statements and notes can be stored in a CPU module. This type of comments uses the following number of steps. (Assumed that only one-byte characters are entered. Values after the decimal point are rounded up.) • 2 + Number of characters ÷ 2 (steps)
In Peripheral	• Line statement • P statement • I statement • Note	Statements and notes cannot be stored in a CPU module. (Only position information is stored.) Statements and notes need to be stored in a peripheral. One line consumes one step. A text that has been entered is automatically preceded by an asterisk "*".