MITSUBISHI

Changes for the Better

Mitsubishi Programmable
Controller

Training Manual

Q-series basic course (for GX Works2)

● SAFETY PRECAUTION ●

(Always read these instructions before using the products.)

When designing the system, always read the relevant manuals and give sufficient consideration to safety.
During the exercise, pay full attention to the following points and handle the product correctly.

[EXERCISE PRECAUTIONS]

◇! WARNING

- Do not touch the terminals while the power is on to prevent electric shock.
- Before opening the safety cover, make sure to turn off the power or ensure the safety.
- Do not touch the movable portion.

⚠ CAUTION

- Follow the instructor's direction during the exercise.
- Do not remove the module of the demonstration machine or change wirings without permission.
  Doing so may cause failures, malfunctions, personal injuries and/or a fire.
- Turn off the power before installing or removing the module.
  Failure to do so may result in malfunctions of the module or electric shock.
- When the demonstration machine (such as X/Y table) emits abnormal odor/sound, press "Power switch" or "Emergency switch" to turn off.
- When a problem occurs, notify the instructor as soon as possible.

REVISIONS

*The textbook number is written at the bottom left of the back cover.

Print date	*Textbook number	Revision
Oct., 2012	SH-081123ENG-A	First edition

This textbook confers no industrial property rights or any rights of any other kind, nor does it confer any patent licenses. Mitsubishi Electric Corporation cannot be held responsible for any problems involving industrial property rights which may occur as a result of using the contents noted in this textbook.

CONTENTS

CHAPTER 4 SEQUENCE AND BASIC INSTRUCTIONS -PART 1- 4- 1 to 4-42

CHAPTER 5 BASIC INSTRUCTION -PART 2- 5- 1 to 5-58

INTRODUCTION

This textbook explains the programmable controller, the program editing methods with GX Works2, the sequence instructions and the application instructions for understanding the MELSEC-Q series programming.
The multiple CPU system is available for the MELSEC-Q series with multiple CPU modules, but this textbook explains the case in which one CPU module is used.

The related manuals are shown below.

(1) QCPU User's Manual (Hardware Design, Maintenance and Inspection)
······························ SH-(NA)080483ENG
Explains the hardware.

(2) QnUCPU User's Manual (Function Explanation, Program Fundamentals)
······························ SH(NA)-080807ENG
Explains the functions and programming method.

(3) MELSEC-Q/L Programming Manual (Common Instruction)
······························ SH(NA)-080809ENG
Explains details of each instruction.

(4) GX Works2 Beginner's Manual (Simple Project)
······························ SH(NA)-080787ENG

(5) GX Works2 Version 1 Operating Manual (Common)
······························ SH(NA)-080779ENG

(6) GX Works2 Version 1 Operating Manual (Simple Project)
······························ SH(NA)-080780ENG

(7) Before Using the Product
······························ BCN-P5782

(8) Analog-Digital Converter Module User's Manual
······························ SH(NA)-080055

(9) Digital-Analog Converter Module User's Manual
······························ SH(NA)-080054

(10) I/O Module Type Building Block User's Manual
······························ SH(NA)-080042

(11) MELSOFT GX Works2 FB Quick Start Guide
······························ L-08182ENG

CHAPTER 1 BASICS OF PROGRAMMABLE CONTROLLER

1.1 Program

If a programmable controller is assumed as a control ladder, it can be described by an input ladder, output ladder, and internal sequential operation.

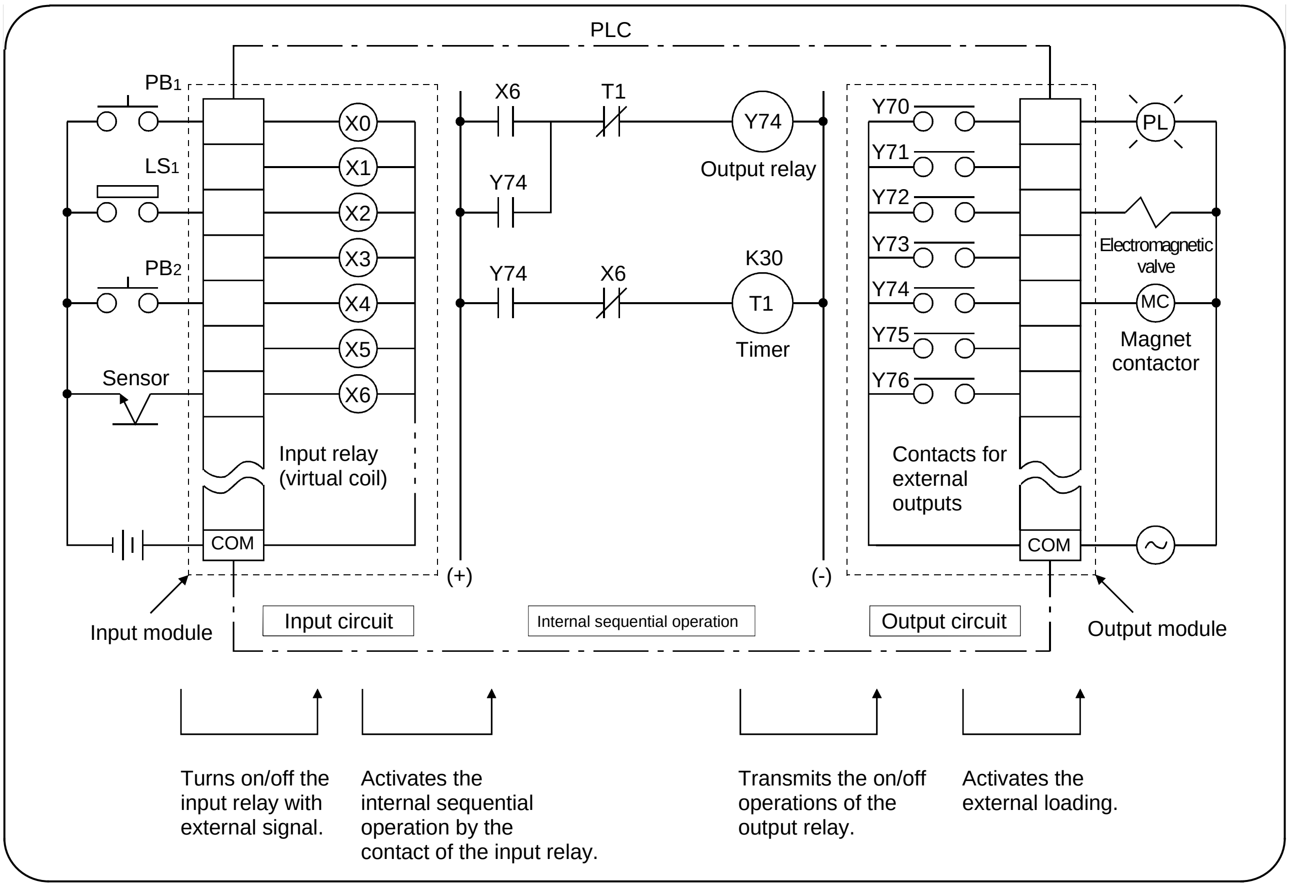
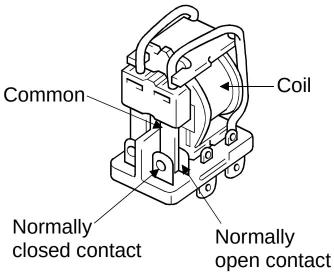

PLC
PB1
LS1
PB2
Sensor
COM
X0
X1
X2
X3
X4
X5
X6
Input relay (virtual coil)
X6
T1
Y74
Y74
X6
Y74
Output relay
K30
T1
Timer
(+)
(-)
Y70
Y71
Y72
Y73
Y74
Y75
Y76
Contacts for external outputs
COM
PL
Electromagnetic valve
MC
Magnet contactor
Input module
Input circuit
Internal sequential operation
Output circuit
Output module
Turns on/off the input relay with external signal.
Activates the internal sequential operation by the contact of the input relay.
Transmits the on/off operations of the output relay.
Activates the external loading.

Figure 1.1 Programmable controller configuration

A programmable controller is an electronic device centered around microcomputers. Actually, a programmable controller is assemblies of relays, timers, and counters. As shown in figure 1.1, the internal sequential operation is executed by turning on or off the coil. The on/off condition of the coil depends on the connection condition (in series or in parallel) and results of the normally open or normally closed contacts

"Relay", which is also called an electromagnetic relay, is a switch to relay signals. The relay is a key component to make up a logic ladder.

1) Energizing the coil ⇨ Magnetization
   - The normally open contact closes. (Conducted)
   - The normally closed contact opens. (Not conducted)
2) De-energizing the coil ⇨ Demagnetization
   - The normally open contact opens. (Not conducted)
   - The normally closed contact closes. (Conducted)



	Coil off (always)	Coil on (in operation)
Normally open contact ⊣⊢	Not conducted	Conducted
Normally closed contact ⊣/⊢	Conducted	Not conducted

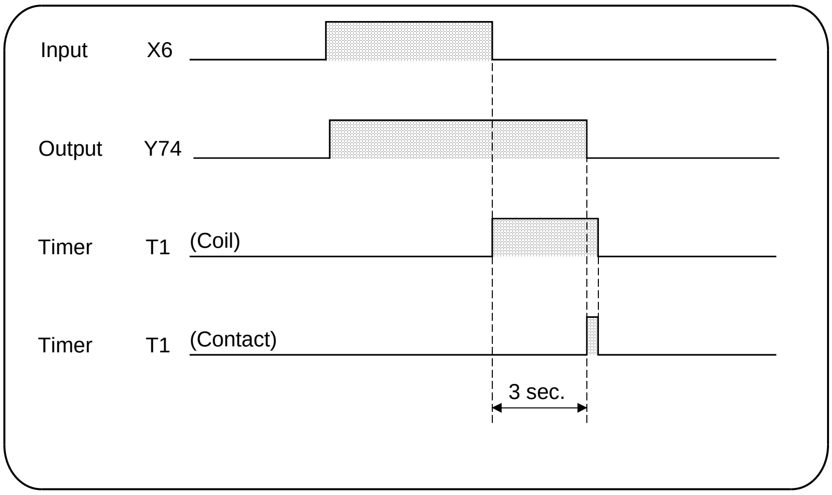
Internal Sequential Operation

The following shows the signal flow of the internal sequential operation of figure 1.1.

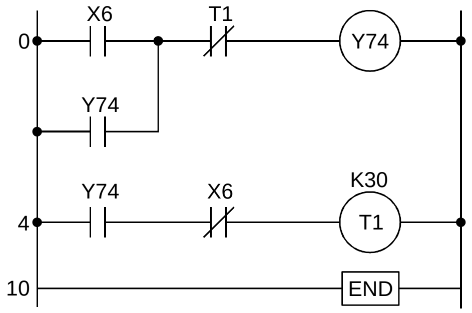
1) When the sensor turns on, the coil of the input relay X6 is magnetized.
2) Magnetizing the coil of the input relay X6 conducts the normally open contact X6 and magnetizes the coil of the output relay Y74.
   (As the timer is not magnetized at this time, the normally closed contact remains conducted.)
3) Once the coil of the output relay Y74 is magnetized, the external output contact Y74 is conducted and the magnetic contactor (MC) is turned on.
4) Turning off the sensor demagnetizes the coil of the input relay X6 and the normally open contact X6 becomes non-conductive.
   As the self-maintaining normally open contact Y74 is conducted, the coil remains magnetized. (Self-maintaining operation)
5) When the coil of the output relay Y74 is magnetized (with the normally open contact Y74 conducted), turning off the sensor (with normally closed contact X6 conducted) magnetizes the coil of the timer T1 and the timer starts measuring the time.
   After three sec. (K30 indicates 3.0sec.), the normally open contact of the timer becomes conducted and the normally closed contact becomes non-conductive.
6) As a result, the coil of the output relay Y74 demagnetizes and the load magnet contactor drops.
   Also, the output relay self-maintenance is released.

Operation diagram

The following time chart explains the input/output relays and timer operations.



The internal sequential operation can be regarded as the program of the programmable controller. The program is saved in the program memory as similar to the instruction list



(a) Ladder diagram

Step number	Instruction word	Device
0	LD	X6
1	OR	Y74
2	ANI	T1
3	OUT	Y74
4	LD	Y74
5	ANI	X6
6	OUT	T1 K30
10	END	

Repeat operation

(b) Instruction list (program list)

Figure 1.2 Program

- A program consists of a large number of instruction words and devices.

- The instructions contain instruction words and devices. In addition, the instructions are numbered to represent the order of operations. The numbers are called step numbers.
  (Instruction words are also called instructions.)

- The number of steps varies depending on the types of instructions or the setting method for the values to be used for the I/O numbers and operations. (The more steps are needed for the operation with complicated operation.)

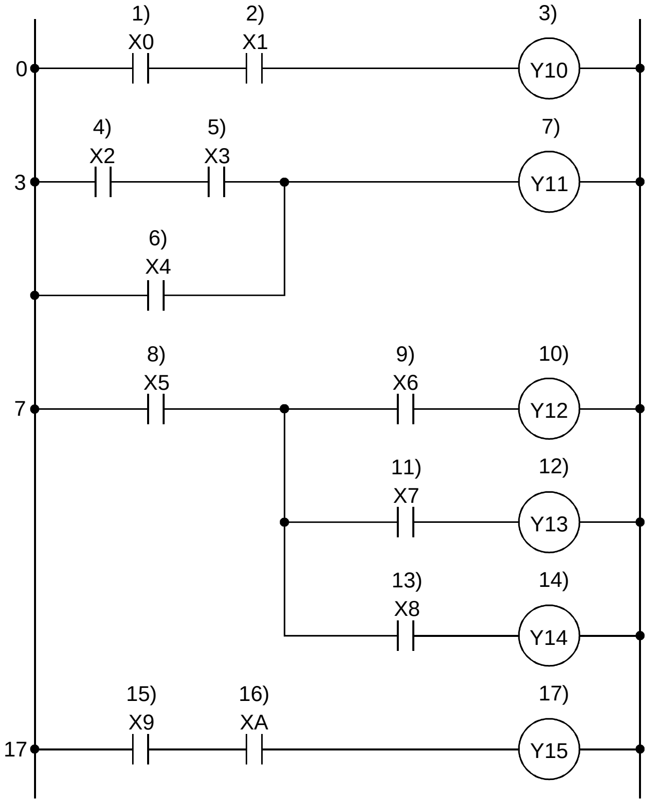
- The instructions repeat from the step number 0 to the END instruction. (This is called "repeat operation", "cyclic operation" or "scanning".)
  Amount of time necessary for one cycle is called operation cycle (scan time).

- The number of steps from the step number 0 to the END instruction is the length or size of the program.

- The program is stored in the program memory inside the CPU. The operation is executed in a ladder block unit.
  One ladder block ranges from the operation start instruction (LD, LDI) to the OUT instruction (including the data instruction).

1.2 Program Processing Procedure

The operation process is executed in series from the start step of the program memory left to right and top to bottom (in the order of 1), 2) ... 17)) in a ladder block unit as shown below.



1.3 MELSEC-QnUD Module Configuration

(1) Universal model

The Universal model QCPU is used for a training in this textbook, therefore, "QCPU" indicates "Universal model QCPU" unless otherwise noted.

(2) Basic configuration of a programmable controller system

The following figure shows an actual programmable controller configuration.

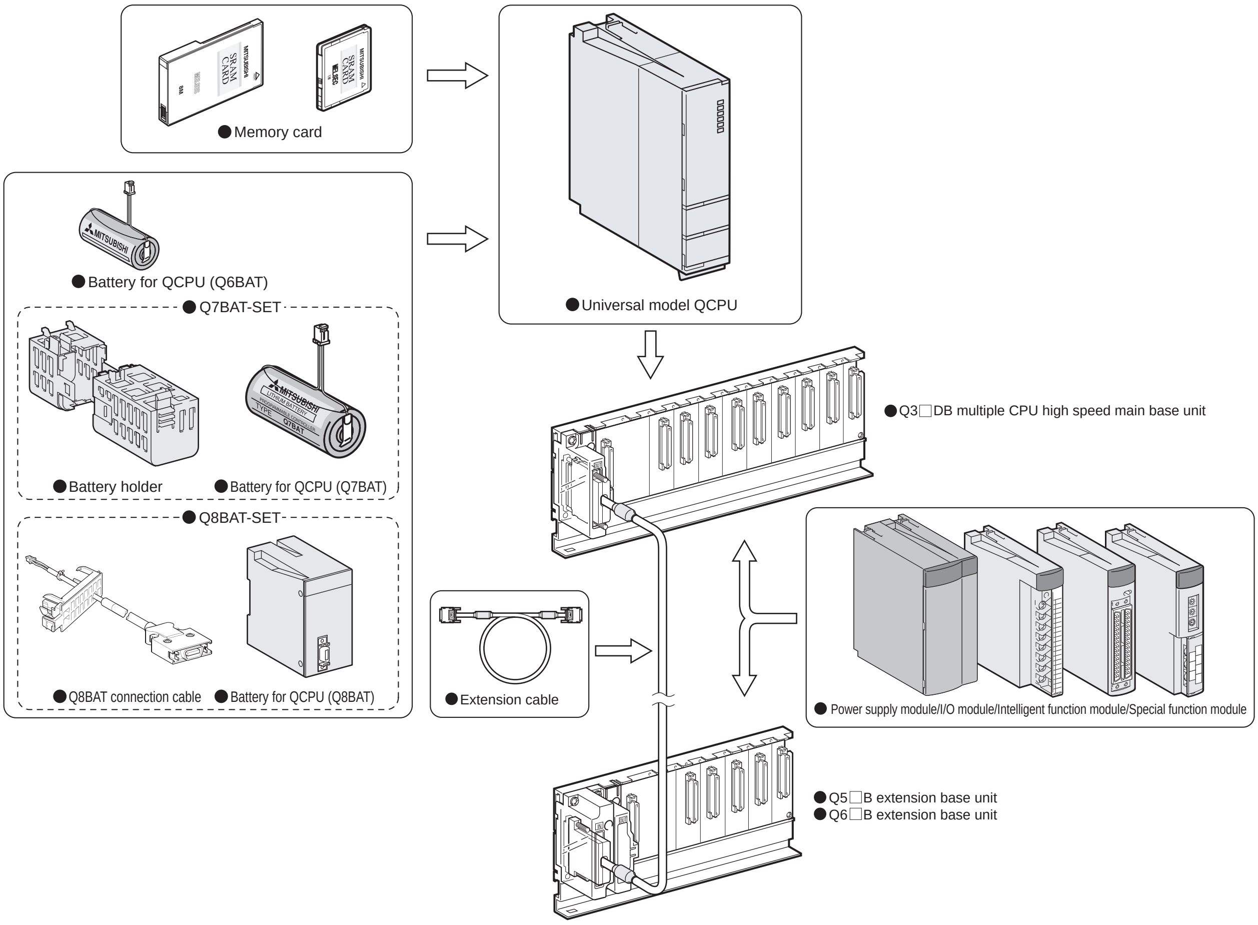
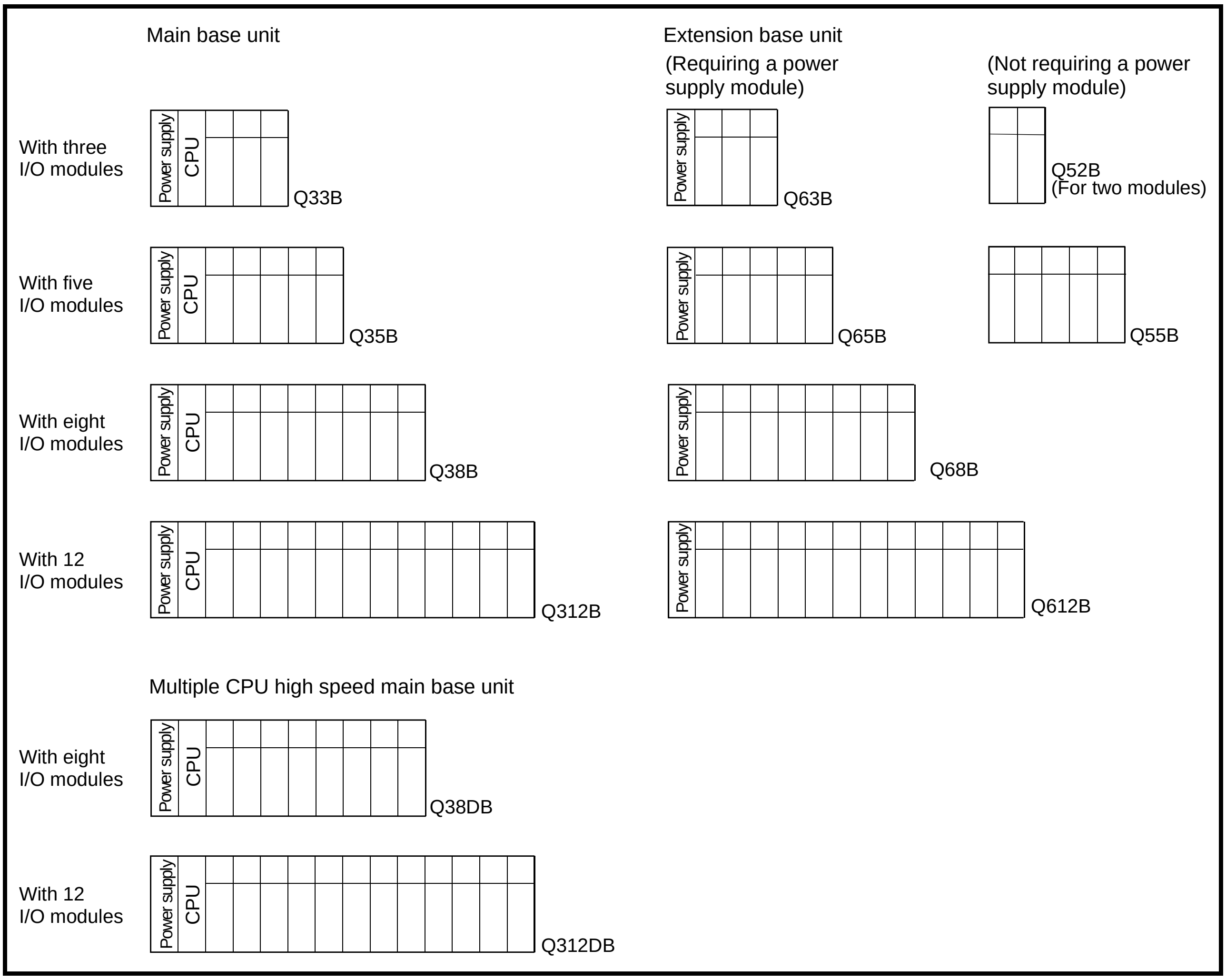

●Memory card
●Universal model QCPU
●Battery for QCPU (Q6BAT)
●Q7BAT-SET
●Battery holder
●Battery for QCPU (Q7BAT)
●Q8BAT-SET
●Q8BAT connection cable
●Battery for QCPU (Q8BAT)
●Extension cable
●Q3□DB multiple CPU high speed main base unit
● Power supply module/I/O module/Intelligent function module/Special function module
●Q5□B extension base unit
●Q6□B extension base unit

Figure 1.3 MELSEC-QnUD module configuration (when Q3□DB is used)

Base Unit



- The main roles of the base unit are; fixing the power supply module, CPU module, and I/O modules, supplying 5VDC power from the power supply module to the CPU module and I/O modules, and transmitting the control signals to each module.

Power Supply Module

Module name	Input	Output
Q61P	100V to 240VAC	5VDC 6A
Q62P	100V to 240VAC	5VDC 3A, 24VDC 0.6A
Q63P	24VDC	5VDC 6A
Q64P(N)	100V to 120V/AC200 to 240VAC	5VDC 8.5A
Q61P-D	100V to 240VAC	5VDC 6A

CPU Module

CPU type	Program capacity (maximum)	Basic instruction processing speed	Maximum I/O points for connecting to a programmable controller
Q00UJCPU	10K steps	120ns	256 points
Q00UCPU	10K steps	80ns	1024 points
Q01UCPU	15K steps	60ns	1024 points
Q02UCPU	20K steps	40ns	2048 points
Q03UD(E)CPU	30K steps	20ns	4096 points
Q04UD(E)HCPU	40K steps	9.5ns	
Q06UD(E)HCPU	60K steps		
Q10UD(E)HCPU	100K steps		
Q13UD(E)HCPU	130K steps		
Q20UD(E)HCPU	200K steps		
Q26UD(E)HCPU	260K steps		
Q50UDEHCPU	500K steps		
Q100UDEHCPU	1000K steps		

I/O Module

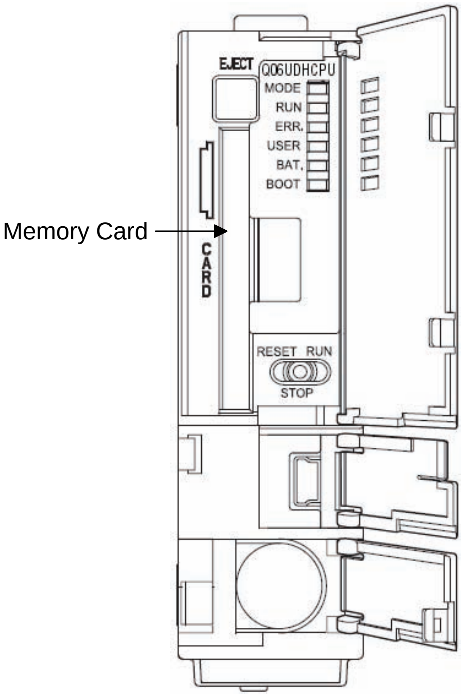
	I/O points Format	8 points	16 points	32 points	64 points
Input module	120VAC	–	○	–	–
	240VAC	○	–	–	–
	24VDC (positive common)	–	○	○	○
	24VDC (high-speed input)	○	–	–	–
	24VDC (negative common)	–	○	○	–
	5/12VDC	–	○	○	○
Output module	Contact output	–	○	–	–
	Independent contact output	○	–	–	–
	Triac output	–	○	–	–
	Transistor output (sink)	○	○	○	○
	Transistor output (source)	–	○	○	–
I/O mixed		○	–	○	–

Memory Card

A QCPU equips a built-in memory as standard for storing parameters and programs, therefore, the programs can be executed without a memory card.

The memory cards are required for the situations in the table below.

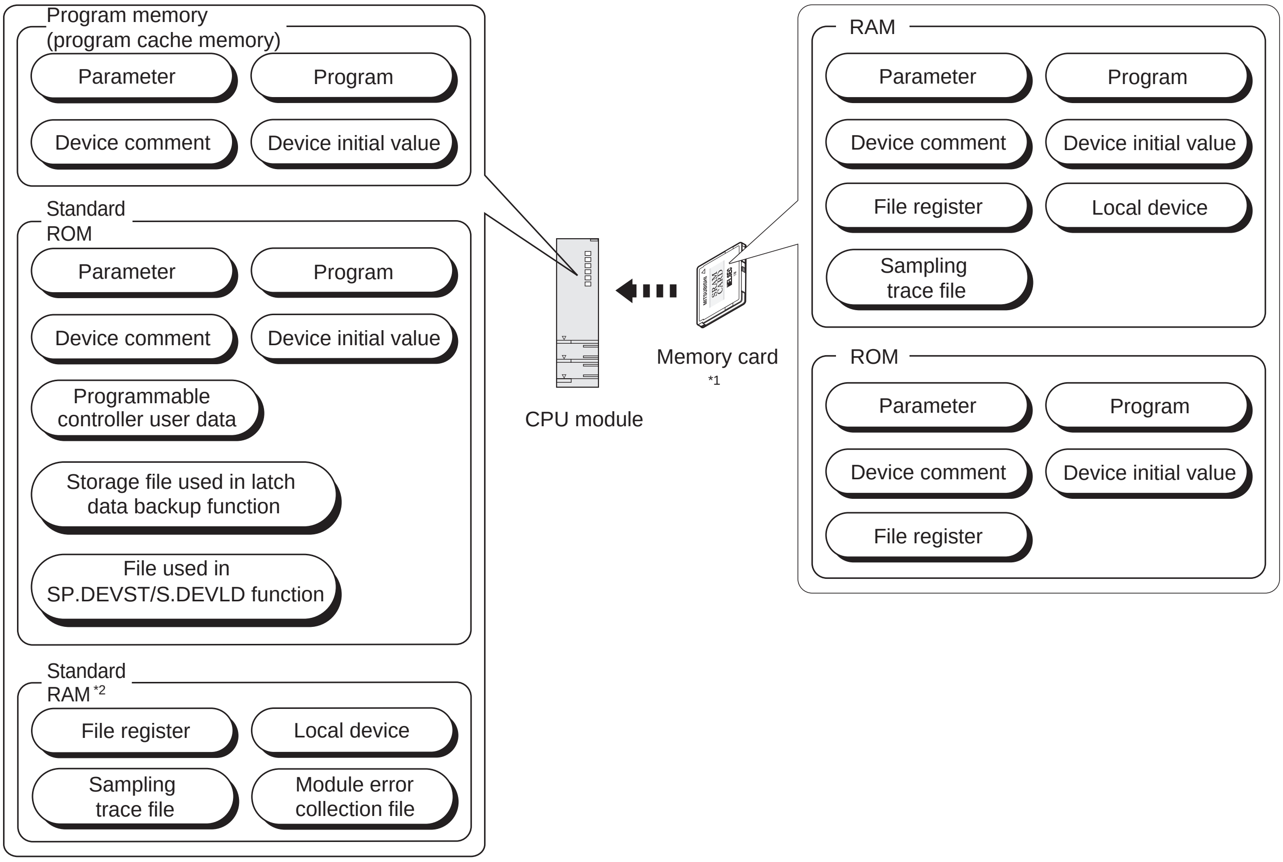
Type	Description
SRAM card	Data can be written or changed within the memory capacity. <Example of the usage> • For the boot operation • For storing the sampling trace data • For storing the SFC trace data • For storing the error history data
Flash card	The contents of the program memory or the specified file can be written at a time. The newly written data replaces all original data. Data can be read by the READ instruction of the sequence program. <Example of the usage> • For the boot operation • When the changing the data is unnecessary
ATA card	Data can be written or changed within the program capacity. Programmable controller user data of an ATA card can be accessed by the file access instruction (such as the FWRITE instruction) in a sequence program through a CSV format or binary format. <Example of the usage> • For the boot operation • For programmable controller user data (general-purpose data)



- Memory cards are required when the data capacity exceeds the capacity of the built-in program memory, standard RAM, and standard ROM.
- Select the memory card according to the size of the program or the type of the data to be stored.
- Install the enclosed backup battery before using the SRAM-type RAM card first. The SRAM card data cannot be baked up unless the battery is installed.
- Format the memory card before using it.
- Data can be written to a Flash card for 100,000 times, and for an ATA card, data can be written for 1,000,000 times.

<Reference: Universal model QCPU memory system configuration>

The memory of the Universal model QCPU consists of the following blocks.



*1: A memory card cannot be used for Q00UJCPU, Q00UCPU, Q01UCPU.

*2: Q00UJCPU has no standard RAM.

• Program memory:	A memory for storing programs and parameters for a CPU module operation A program operation is executed by transferring a program stored in the program memory to the program cache memory.
• Program cache memory:	A memory for operating programs A program operation is executed by transferring a program stored in the program memory to the program cache memory.
• Standard RAM:	A memory for using file registers, local devices, and sampling trace files without a memory card Using the standard RAM as the file registers enables the high-speed access as well as data registers. The standard RAM is also used for storing the module error collection file.
• Standard ROM:	A memory for storing data such as parameters and programs
• Memory card (RAM):	A card for storing the local device, debug data, SFC trace data, and error history data with the parameters and program.
• Memory card (ROM):	A Flash card for storing parameters, programs, and file registers. An ATA card stores parameters, programs, and the programmable controller user data (general-purpose files).

POINT

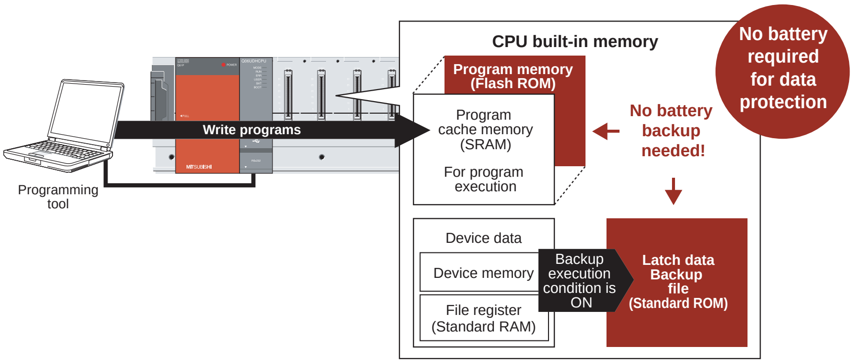
Secure backup by long-term storage

Programs and parameter files are automatically backed up to the program memory (Flash ROM) which does not require a battery backup. This prevents a loss of the program and parameter data due to the flat battery.

The battery backup time is also reduced significantly.

In addition, the important data (such as device data) can be backed up to the standard ROM to prevent a loss of the data due to the flat battery in case of consecutive holidays.

The backup data is restored automatically when the power is turned on next time.

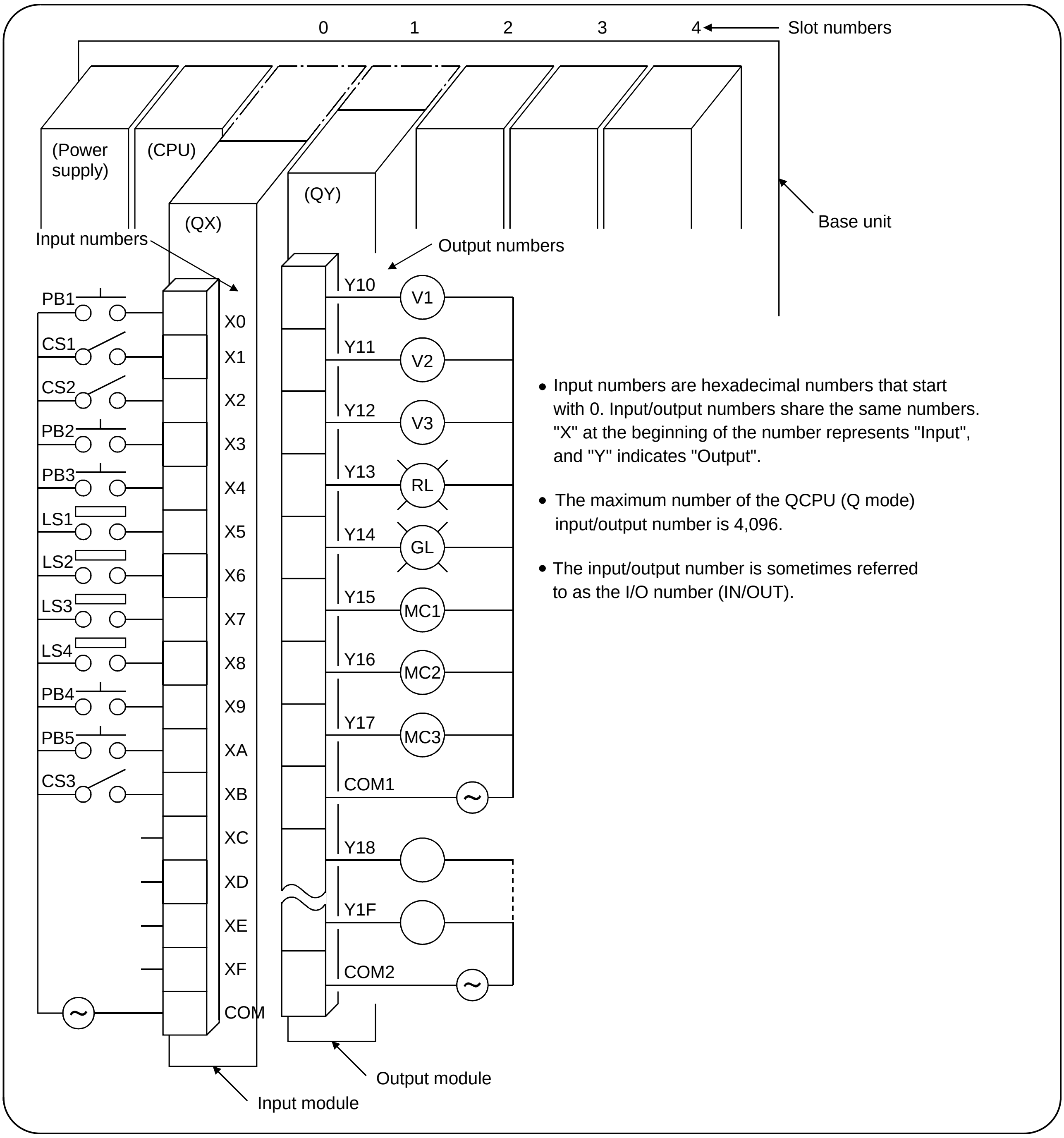

Programming tool
Write programs
CPU built-in memory
Program memory (Flash ROM)
Program cache memory (SRAM)
For program execution
No battery required for data protection
No battery backup needed!
Device data
Device memory
File register (Standard RAM)
Backup execution condition is ON
Latch data Backup file (Standard ROM)

1.4 External I/O Signal and I/O Number

(1) Wiring of I/O devices

The signals output from the external input devices are substituted by the input numbers which are determined by the installation positions and terminal numbers of the connected input module and used in a program.

For the operation results output (coil), use the output numbers which are determined by the installation position and the terminal number of the output module to which the external output module is connected.

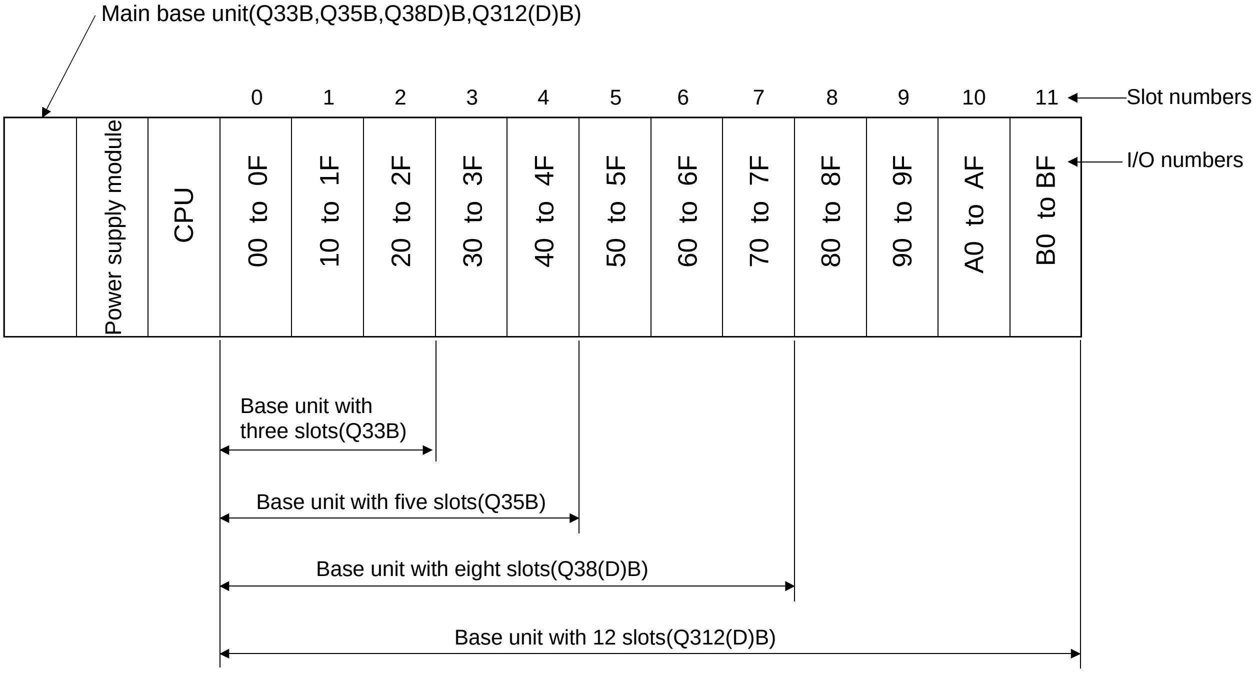


- Input numbers are hexadecimal numbers that start with 0. Input/output numbers share the same numbers. "X" at the beginning of the number represents "Input", and "Y" indicates "Output".
- The maximum number of the QCPU (Q mode) input/output number is 4,096.
- The input/output number is sometimes referred to as the I/O number (IN/OUT).

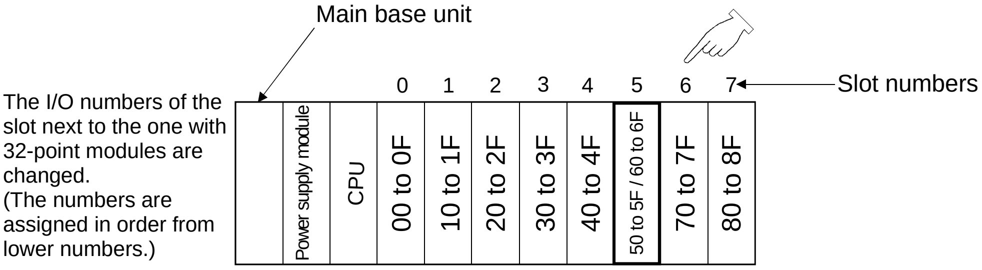
Figure 1.4 Wiring of I/O devices

(2) I/O numbers of a main base unit

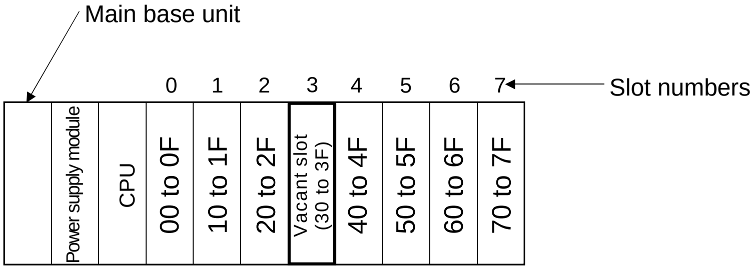
The I/O numbers of I/O modules which are attached to a main base unit are assigned as follows. This configuration applies to both I/O modules and intelligent function modules.


Main base unit(Q33B,Q35B,Q38D)B,Q312(D)B)
0
1
2
3
4
5
6
7
8
9
10
11
Slot numbers
I/O numbers
Power supply module
CPU
00 to 0F
10 to 1F
20 to 2F
30 to 3F
40 to 4F
50 to 5F
60 to 6F
70 to 7F
80 to 8F
90 to 9F
A0 to AF
B0 to BF
Base unit with three slots(Q33B)
Base unit with five slots(Q35B)
Base unit with eight slots(Q38(D)B)
Base unit with 12 slots(Q312(D)B)

- The I/O numbers of one slot (one module) are assigned in ascending order in 16-point unit (0 to F^H).
  As a standard, 16-point modules should be attached to all slots.
  For example, the following figure shows the I/O numbers of when a 32-point module is attached to the fifth slot.


Main base unit
The I/O numbers of the slot next to the one with 32-point modules are changed. (The numbers are assigned in order from lower numbers.)
0
1
2
3
4
5
6
7
Slot numbers
Power supply module
CPU
00 to 0F
10 to 1F
20 to 2F
30 to 3F
40 to 4F
50 to 5F / 60 to 6F
70 to 7F
80 to 8F

- The I/O numbers are also assigned to a vacant slot (a slot with no I/O module installed).
  For example, if the third slot is vacant, the I/O numbers are assigned as shown below. (in the initial setting)
  The number of assigned points can be changed by the setting.



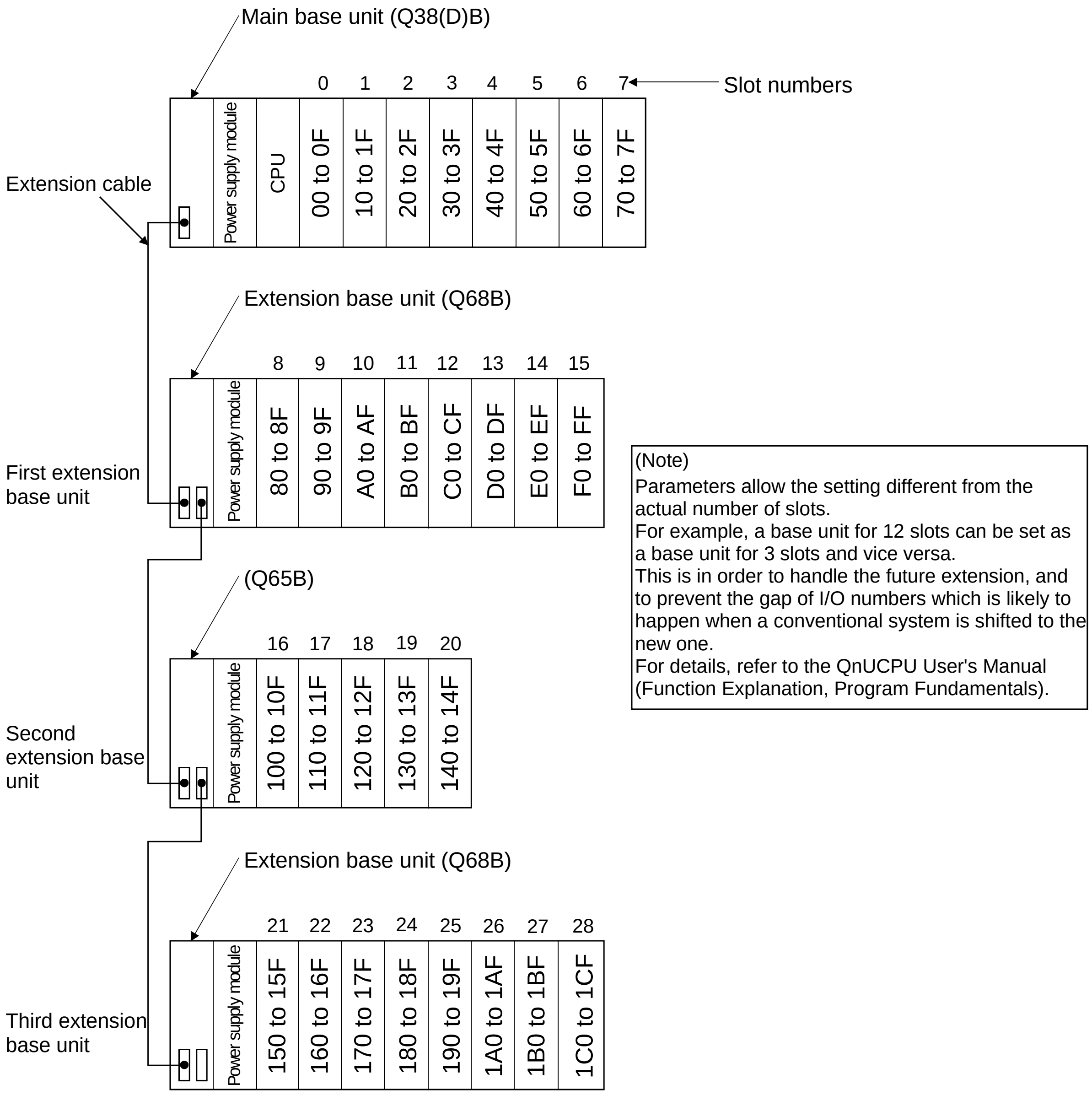
- For the multiple CPU configuration (two to four CPUs), the I/O numbers are assigned from a slot next to a slot where a CPU is attached.

(3) I/O numbers of an extension base unit

Connect an extension base unit when the number of slots of the main base unit is insufficient.

The I/O numbers are assigned as follows in the initial setting.

This configuration applies to both I/O modules and intelligent function modules.



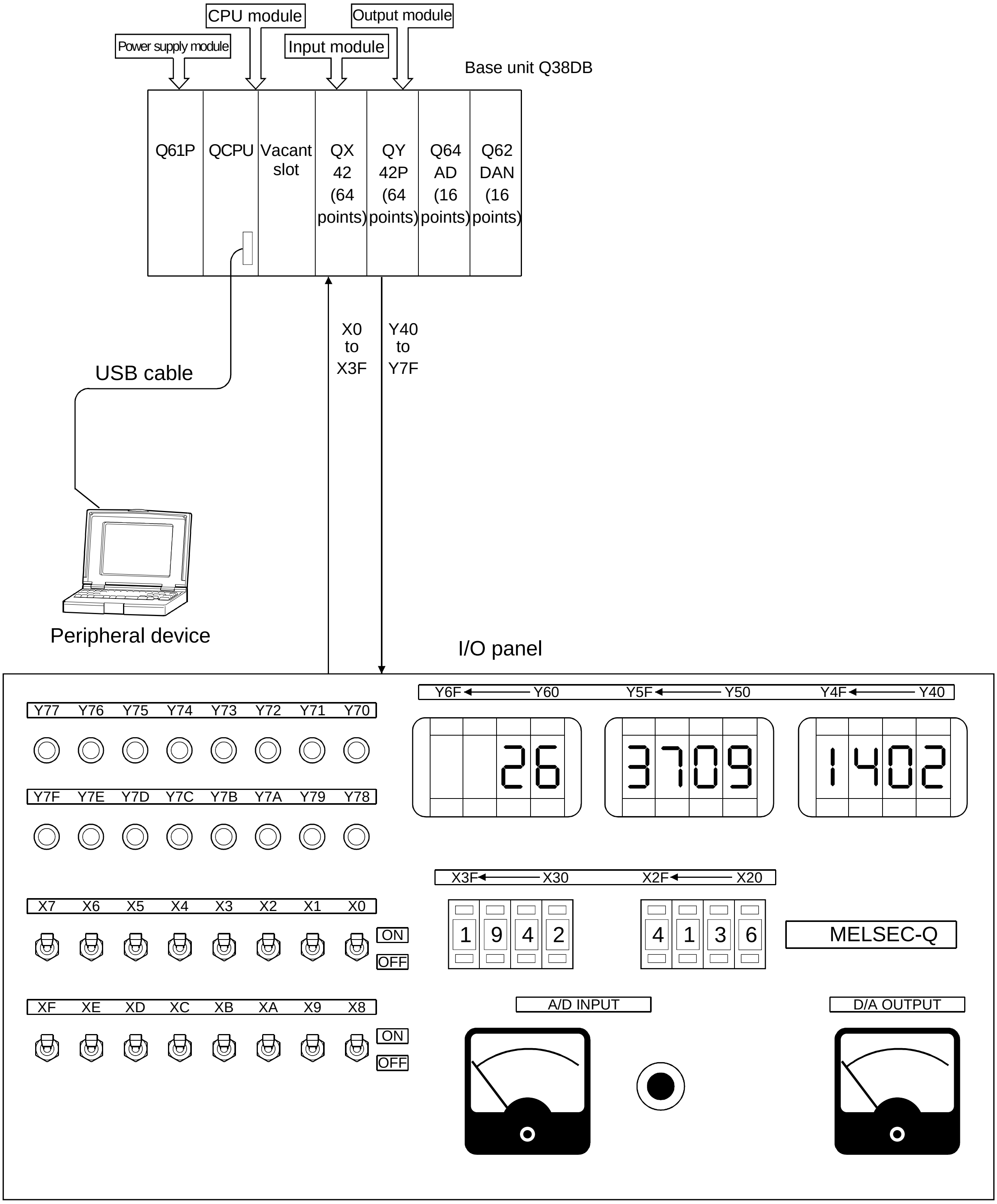
(Note)
Parameters allow the setting different from the actual number of slots.
For example, a base unit for 12 slots can be set as a base unit for 3 slots and vice versa.
This is in order to handle the future extension, and to prevent the gap of I/O numbers which is likely to happen when a conventional system is shifted to the new one.
For details, refer to the QnUCPU User's Manual (Function Explanation, Program Fundamentals).

- The slots of the extension base unit are also assigned in ascending order in 16-point unit.
- The start I/O number of the extension base unit is assigned from the last number of the main base unit or of the previous extension base unit.
- Setting "0" to the parameter can assign the I/O number to the vacant slot or areas with no slot.

The following table shows the number of available extension base units.

CPU type		Number of stages (including the ones connected with GOT in bus connection)
Universal model	Q00UJCPU	2
	Q00UCPU, Q01UCPU, Q02UCPU	4
	Other than the above	7

1.5 System Configuration and I/O Number of Demonstration Machine



CHAPTER 2 OPERATING GX Works2

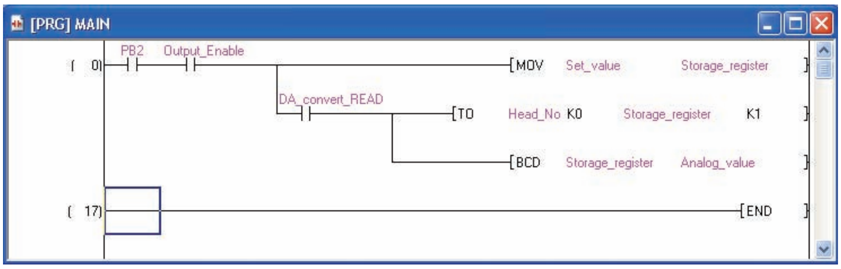
GX Works2 is a programming tool for designing, debugging, and maintaining programs on Windows®.
GX Works2 has improved functionality and operability, with easier-to-use features compared to existing GX Developer.

■ Main functions of GX Works2

GX Works2 can manage programs and parameters in units of projects for each programmable controller CPU.

● Programming

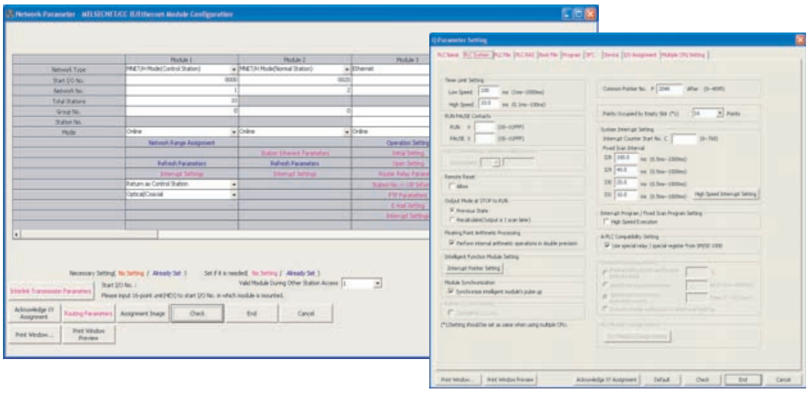
Programs can be created in a Simple project in a similar way with existing GX Developer.
Structured programming in a Structured project is also available with GX Works2.



```
[PRG] MAIN
(   0) PB2  Output_Enable                 [MOV  Set_value          Storage_register   ]
            DA_convert_READ        [TO    Head_No  K0  Storage_register   K1          ]
                                          [BCD  Storage_register   Analog_value       ]
(  17)                                                                         [END   ]
```

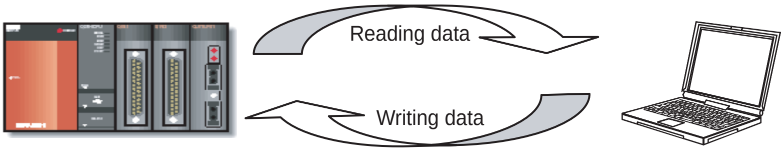
● Setting parameters

The parameters for programmable controller CPUs and network parameters can be set with GX Works2.
Intelligent function module parameter can be set as well.

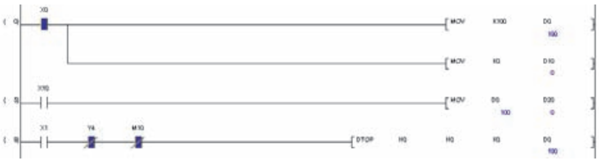


● Writing/reading data to/from a programmable controller CPU

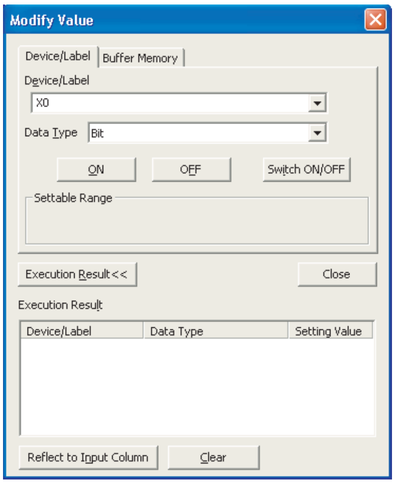
Created sequence programs can be written to/read from a programmable controller CPU using the Read from PLC/Write to PLC function. Also, with the Online program change function, the sequence programs can be changed even when the programmable controller CPU is in RUN.



- Monitoring/debugging
  Created sequence programs can be written to the programmable controller CPU and device values at operation can be monitored online/offline.



Programs can be monitored and debugged.



Modify Value

Device/Label | Buffer Memory

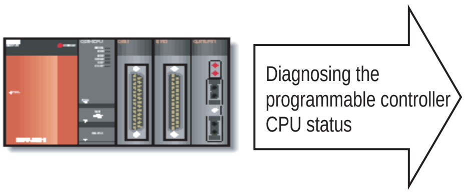
Device/Label: X0

Data Type: Bit

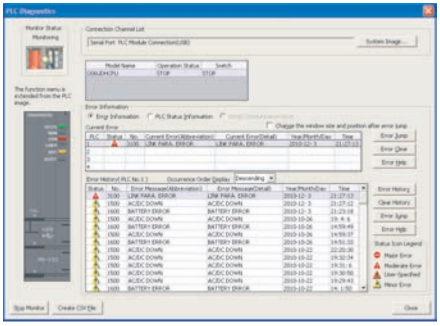
Execution Result

Device/Label	Data Type	Setting Value

- Diagnostics
  The current error status and error history of the programmable controller CPU can be diagnosed.
  With the diagnostics function, the recovery work is completed in a short time.
  With the System monitor function (for QCPU (Q mode)/LCPU), detailed information on such as intelligent function modules can be obtained. This helps to shorten the recovery work time at error occurrence.



Diagnosing the programmable controller CPU status (PLC diagnostics screen)



2.1 Features of GX Works2

This section explains the features of GX Works2.

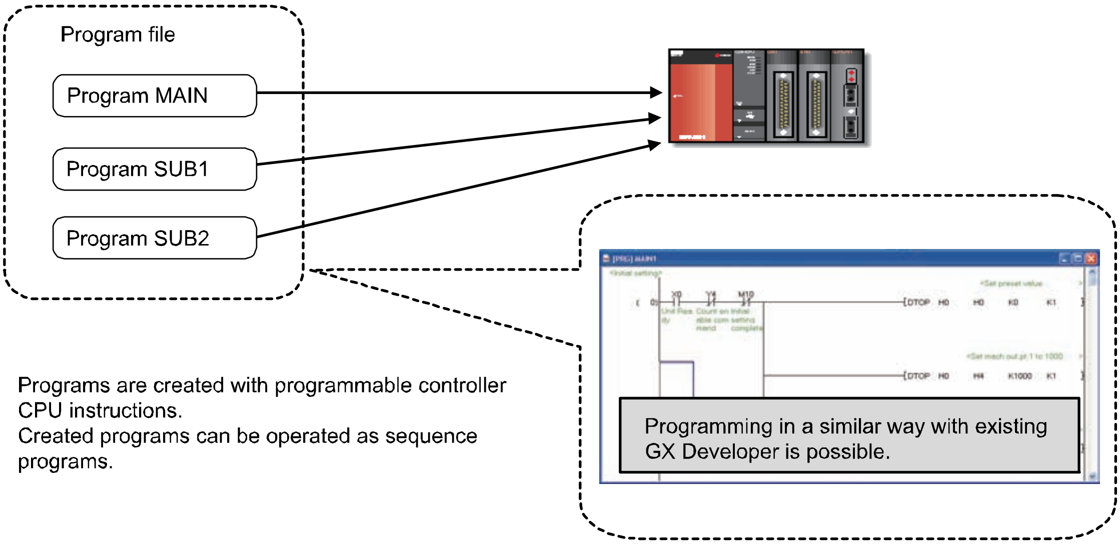
(1) Project types in GX Works2

In GX Works2, the project type can be selected from either of Simple project or Structured project.

(a) Simple project

The Simple project creates sequence programs using instructions for Mitsubishi programmable controller CPU.

Programs in a Simple project can be created in a similar way to existing GX Developer.



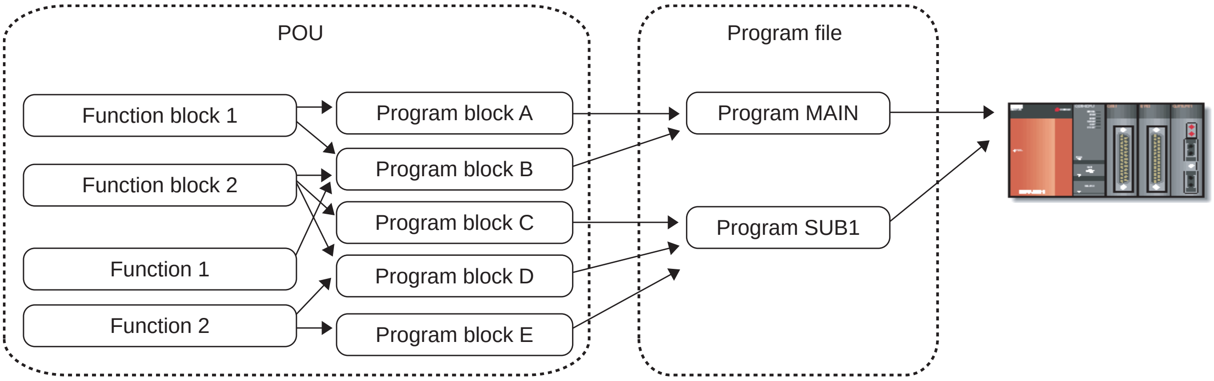
Programs are created with programmable controller CPU instructions.
Created programs can be operated as sequence programs.

Programming in a similar way with existing GX Developer is possible.

(b) Structured project

In a Structured project, programs can be created by structured programming.

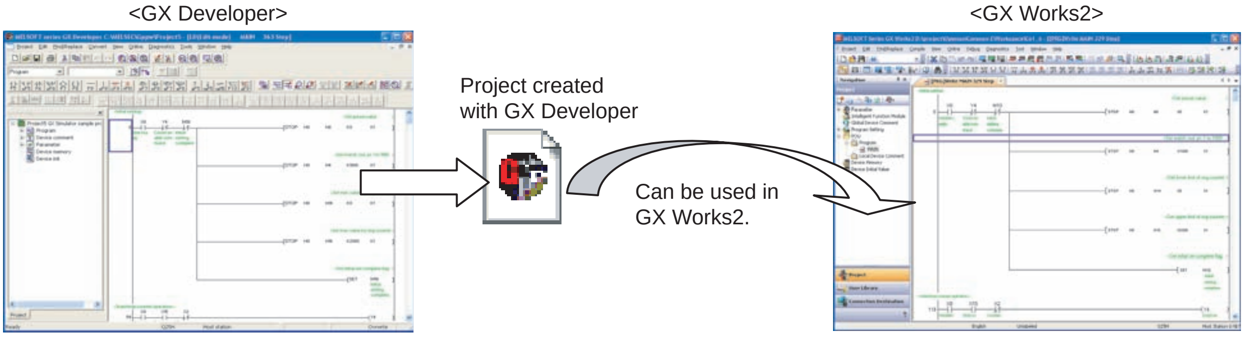
By segmenting a whole control process program into common program parts, highly manageable and usable programming (structured programming) is possible.



Sequence programs are created by combining POU (Program Organization Unit) s.

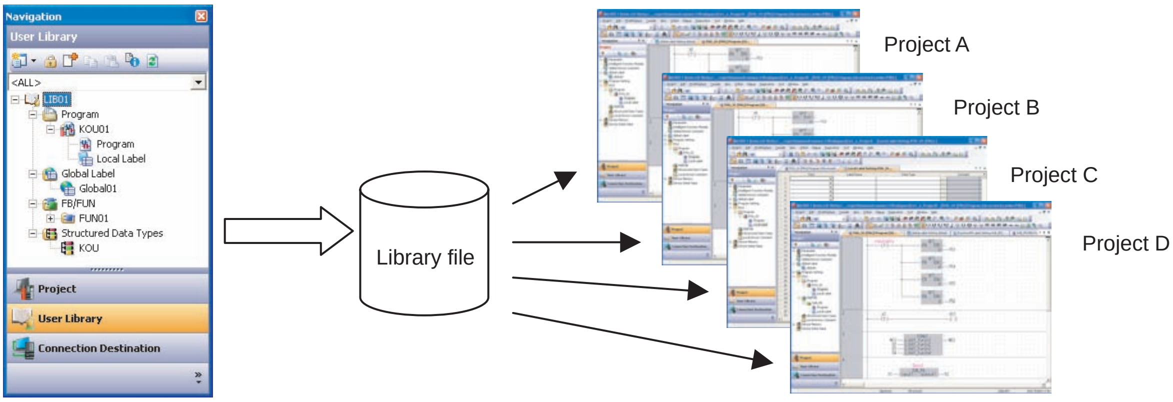
(2) Enhanced use of program assets

Projects created with existing GX Developer can be utilized in a Simple project. Utilizing the past assets improves the efficiency of program design.



(3) Sharing Program Organization Unit (POU) registered as libraries

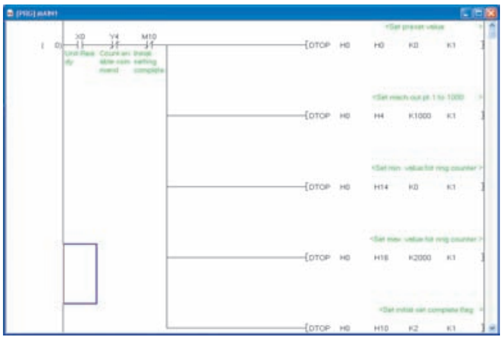
In a Structured project, programs, global labels, and structures frequently used can be registered as user libraries. Utilizing these user libraries reduces time required for creating programs.



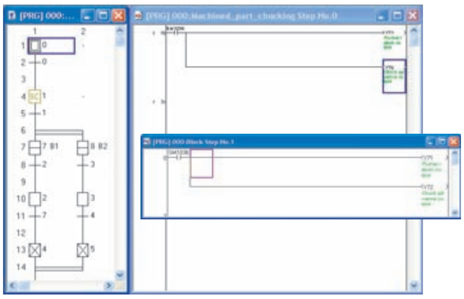
(4) Wide variety of programming languages

The wide variety of programming languages available with GX Works2 enables to select the optimum programming language according to control.

<Ladder>
Programming similar to existing GX Developer

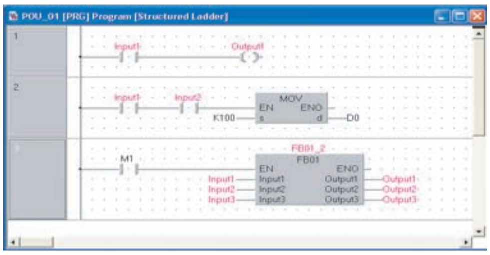


<SFC>
Programming to clarify the procedure

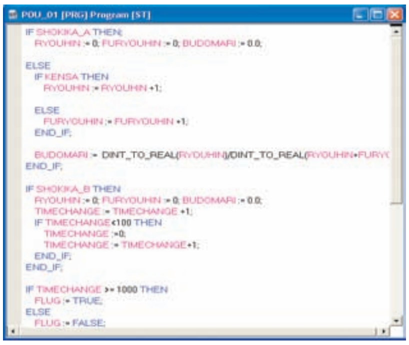


[PRG] 000:...
[PRG] 000:Machined_part_checking Step No.0
[PRG] 000 Block Step No.1

<Structured ladder>
Programming a ladder program graphically



<ST>
Programming in a text language similar to C language



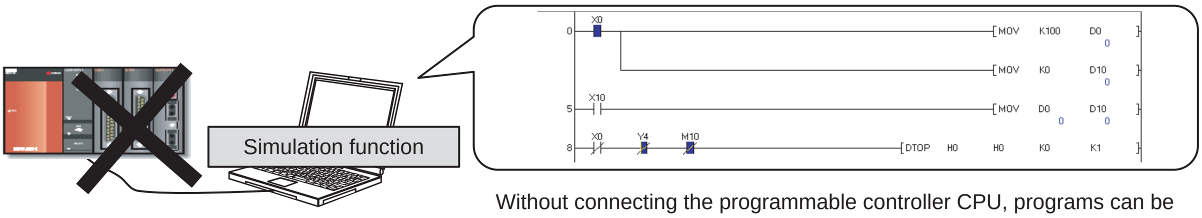
POU_01 [PRG] Program [ST]

```
IF SHOKIKA_A THEN;
  RYOUHIN := 0; FURYOUHIN := 0; BUDOMARI := 0.0;

ELSE
  IF KENSA THEN
    RYOUHIN := RYOUHIN +1;

  ELSE
    FURYOUHIN := FURYOUHIN +1;
  END_IF;

  BUDOMARI := DINT_TO_REAL(RYOUHIN)/DINT_TO_REAL(RYOUHIN+FURY(
END_IF;

IF SHOKIKA_B THEN
  RYOUHIN := 0; FURYOUHIN := 0; BUDOMARI := 0.0;
  TIMECHANGE := TIMECHANGE +1;
  IF TIMECHANGE<100 THEN
    TIMECHANGE :=0;
    TIMECHANGE := TIMECHANGE+1;
  END_IF;
END_IF;

IF TIMECHANGE >= 1000 THEN
  FLUG := TRUE;
ELSE
  FLUG := FALSE;
```

(5) Other features

(a) Offline debugging

Offline debugging using the simulation function is possible with GX Works2. This enables debugging to ensure the normal operation of created sequence programs without connecting GX Works2 to the programmable controller CPU.

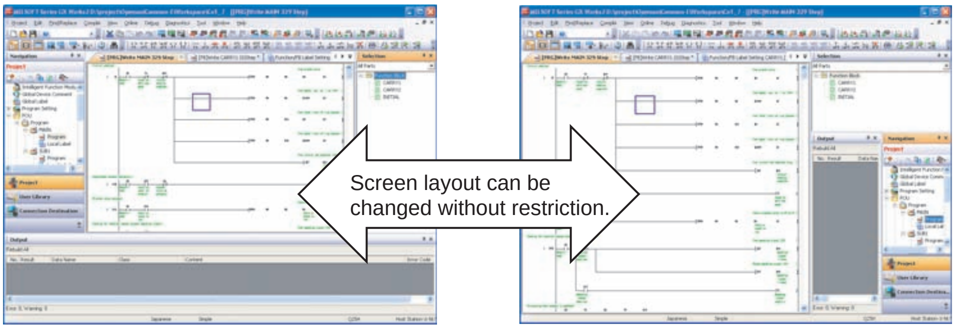


Connecting the programmable controller CPU is unnecessary.

Without connecting the programmable controller CPU, programs can be monitored and debugged in the same way with debugging by the programmable controller CPU.

(b) The screen layout can be customized to the user's preference

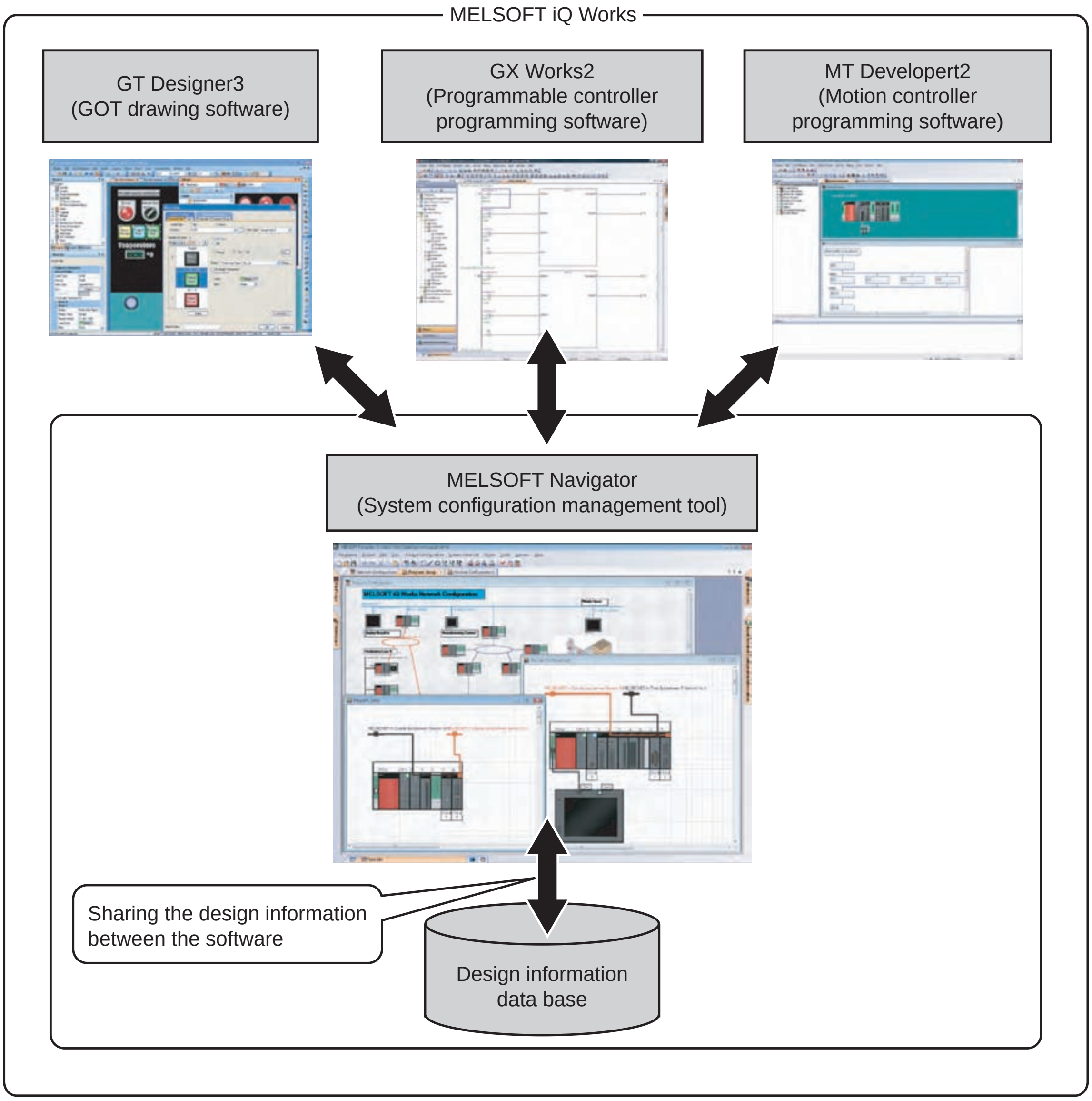
The docking windows enable to change the screen layout of GX Works2 without restriction.



Screen layout can be changed without restriction.

2.1.1 MELSOFT iQ Works

MELSOFT iQ Works integrates the engineering software (GX Works2, MT Developer2, and GT Designer3).
Sharing the design information such as the system design and programming in the whole control system improves the efficiency of program design and efficiency of programming, which reduces costs.

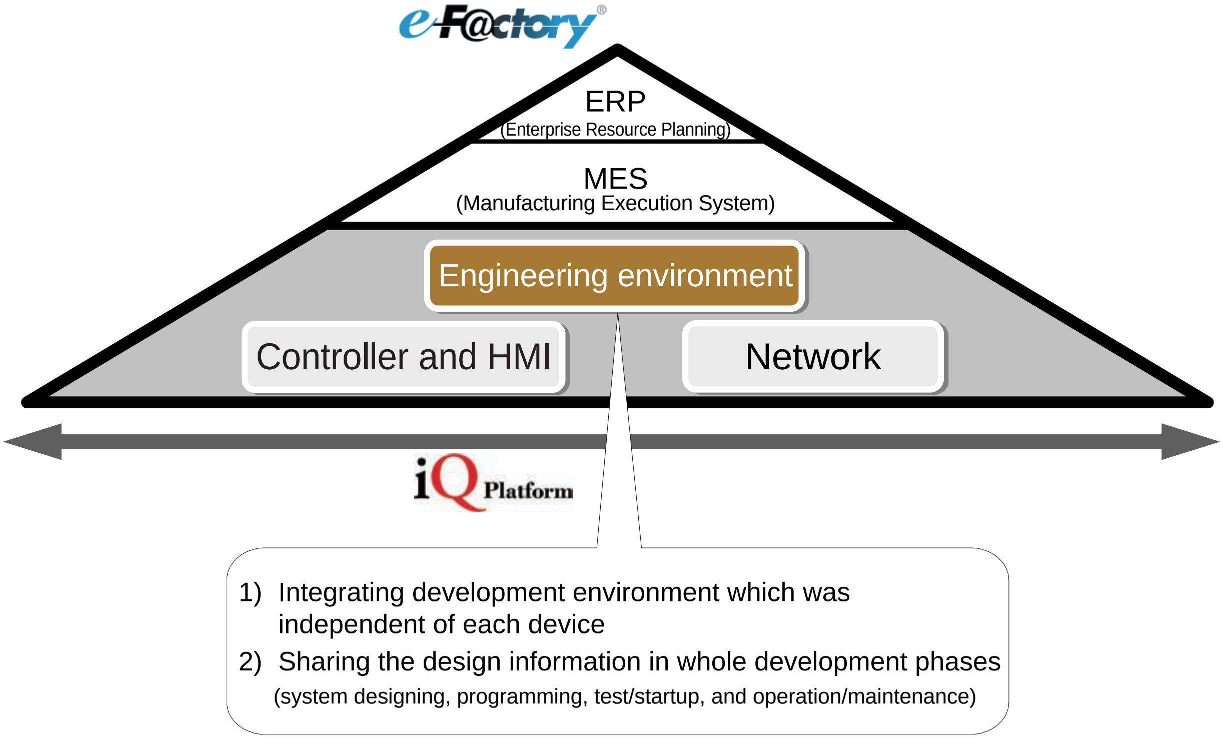


POINT

To start MELSOFT Navigator and each engineering software, click the Start button and follow the procedure below.

• MELSOFT Navigator:	[MELSOFT Application] → [MELSOFT iQ Works] → [MELSOFT Navigator]
• GX Works2:	[MELSOFT Application] → [GX Works2] → [GX Works2]
• MT Developer2:	[MELSOFT Application] → [MT Works2] → [MT Developer2]
• GT Designer3:	[MELSOFT Application] → [GT Works3] → [GT Designer3]

[Purpose of the engineering environment]



1) Integrating development environment which was independent of each device

2) Sharing the design information in whole development phases (system designing, programming, test/startup, and operation/maintenance)

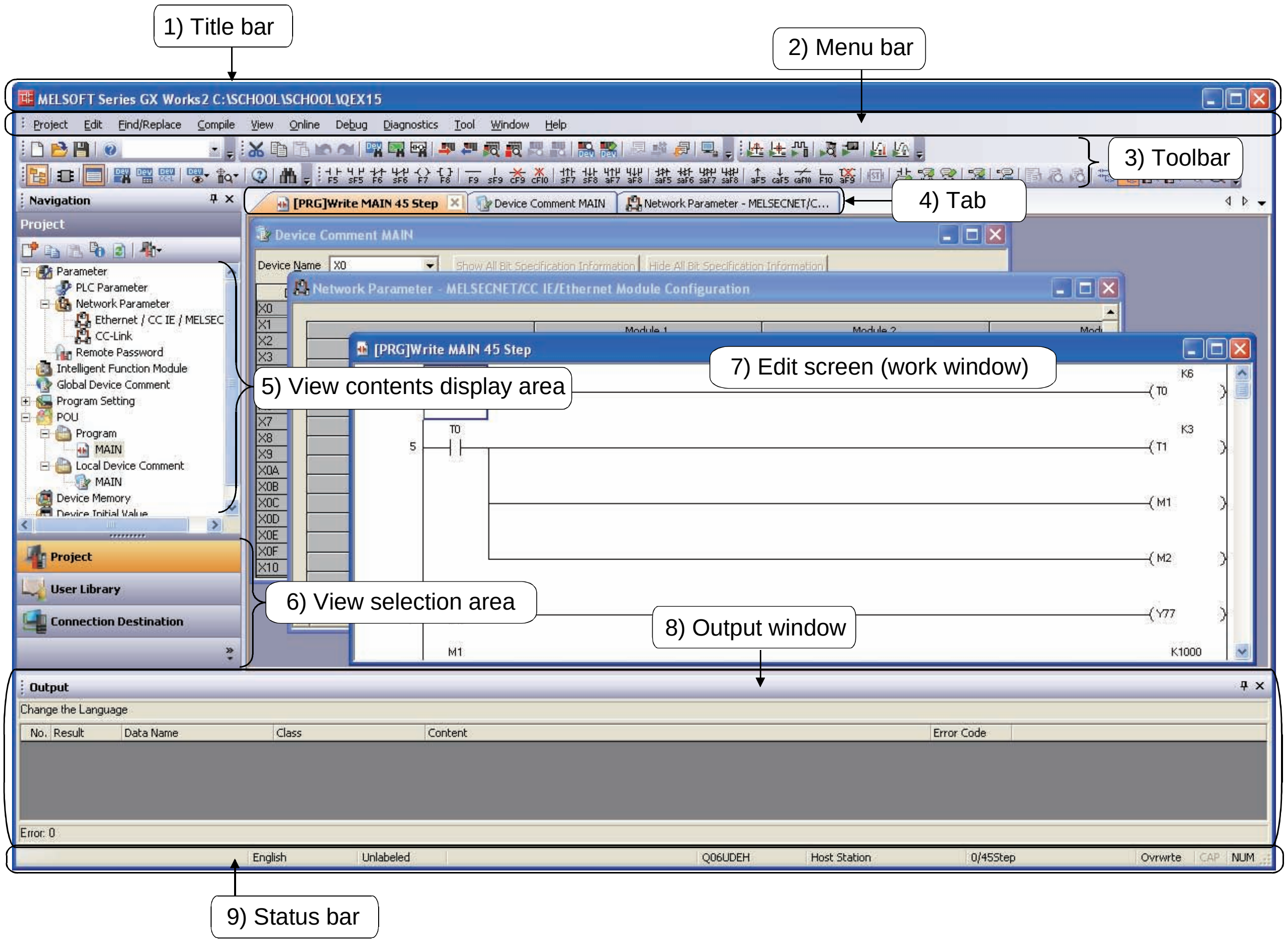
POINT

iQ Platform is a FA integrated concept of MITSUBISHI ELECTRIC.

Integrated Q/improved Quality/intelligent&Quick/innovation&Quest

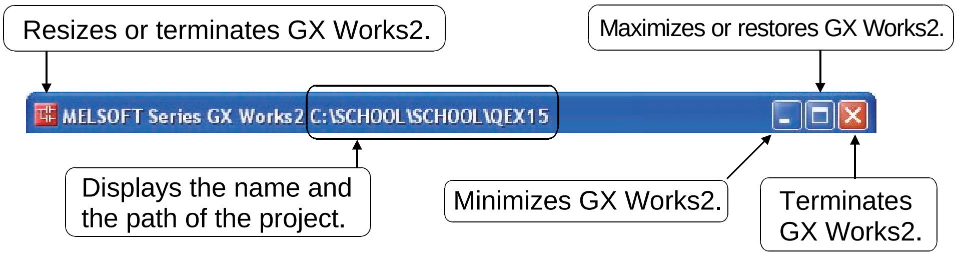
2.2 Basic Knowledge Required for Operating GX Works2

2.2.1 Screen configuration in GX Works2


1) Title bar
2) Menu bar
3) Toolbar
4) Tab
5) View contents display area
6) View selection area
7) Edit screen (work window)
8) Output window
9) Status bar
MELSOFT Series GX Works2 C:\SCHOOL\SCHOOL\QEX15
Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help
Navigation
Project
Parameter
PLC Parameter
Network Parameter
Ethernet / CC IE / MELSEC
CC-Link
Remote Password
Intelligent Function Module
Global Device Comment
Program Setting
POU
Program
MAIN
Local Device Comment
MAIN
Device Memory
Device Initial Value
Project
User Library
Connection Destination
[PRG]Write MAIN 45 Step
Device Comment MAIN
Network Parameter - MELSECNET/C...
Device Comment MAIN
Device Name X0
Show All Bit Specification Information
Hide All Bit Specification Information
Network Parameter - MELSECNET/CC IE/Ethernet Module Configuration
Module 1
Module 2
[PRG]Write MAIN 45 Step
K6
T0
T0
K3
T1
M1
M2
Y77
M1
K1000
Output
Change the Language
No. Result Data Name Class Content Error Code
Error: 0
English Unlabeled Q06UDEH Host Station 0/45Step Ovrwrte CAP NUM

1) Title bar

Title bar displays the name of the active project.



Resizes or terminates GX Works2.

Maximizes or restores GX Works2.

Displays the name and the path of the project.

Minimizes GX Works2.

Terminates GX Works2.

2) Menu bar

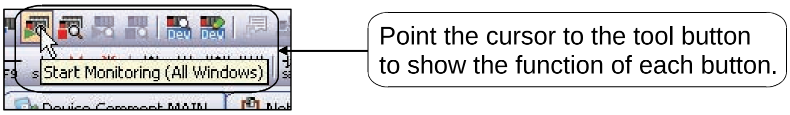
Menu bar is the most frequently used item when operating GX Works2.

Click the menu bar to select a variety of functions from the drop-down menu.

3) Toolbar

Toolbar equips buttons to easily access the commonly-used functions.

This enables a quicker operation.



Point the cursor to the tool button to show the function of each button.

4) Tab

When multiple work windows are open, they are displayed in the tab browser format. Clicking a tab activates the corresponding work window.

5) View contents display area

View contents display area displays the contents of the currently selected view.

6) View selection area

View selection area allows selection of the view to be displayed.

7) Edit screen (work window)

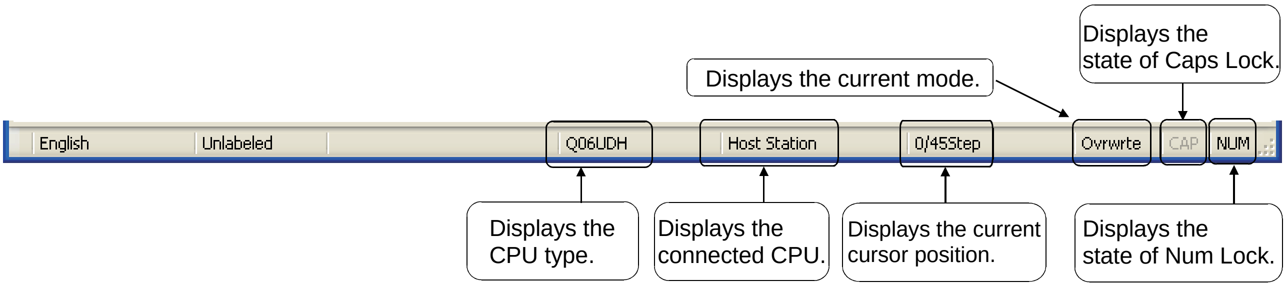
Edit screen displays various screens such as ladder program creation screen and comment creation screen for editing ladder diagrams, comments, and parameters.

8) Output window

Output window displays compilation and check results (such as errors and warnings).

9) Status bar

Status bar displays the status information of GX Works2.



Displays the state of Caps Lock.

Displays the current mode.

Displays the CPU type.

Displays the connected CPU.

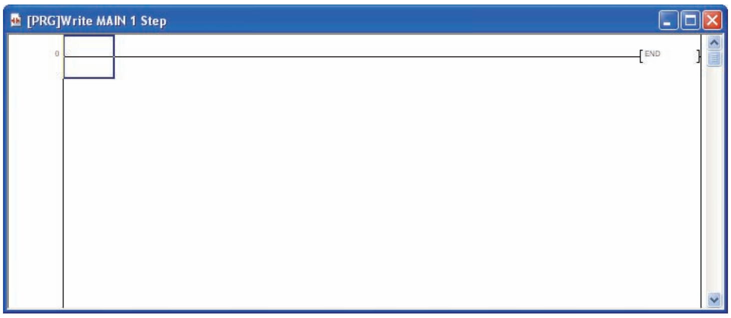
Displays the current cursor position.

Displays the state of Num Lock.

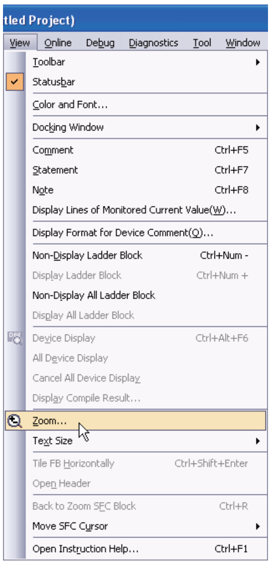
2.2.2 Ladder editor

This section explains the screen display of the GX Works2 ladder editor and its basic operations.

(1) Edit screen

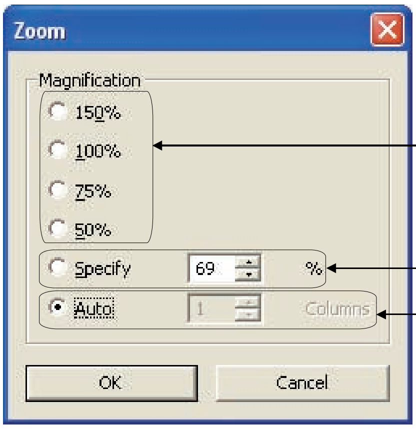


(2) Changing the display size of the edit screen
The display size of the edit screen can be changed.


tled Project)
View Online Debug Diagnostics Tool Window
Toolbar
Statusbar
Color and Font...
Docking Window
Comment Ctrl+F5
Statement Ctrl+F7
Note Ctrl+F8
Display Lines of Monitored Current Value(W)...
Display Format for Device Comment(O)...
Non-Display Ladder Block Ctrl+Num -
Display Ladder Block Ctrl+Num +
Non-Display All Ladder Block
Display All Ladder Block
Device Display Ctrl+Alt+F6
All Device Display
Cancel All Device Display
Display Compile Result...
Zoom...
Text Size
Tile FB Horizontally Ctrl+Shift+Enter
Open Header
Back to Zoom SFC Block Ctrl+R
Move SFC Cursor
Open Instruction Help... Ctrl+F1

1) Click [View] → [Zoom].

The Zoom dialog box is displayed.



Zoom

Magnification

○ 150%

○ 100%

○ 75%

○ 50%

○ Specify 69 %

● Auto 1 Columns

OK Cancel

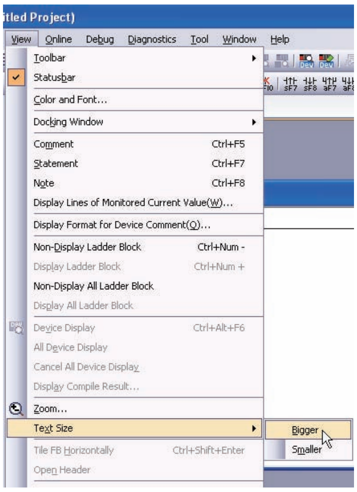
Change the display size according to the selected zoom ratio.

Change the display size according to the specified zoom ratio. (available range: 50 to 150%)

Adjust the width of the ladder automatically to display the entire ladder.

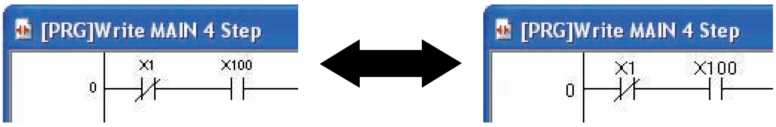
(3) Changing the text size on the edit screen

The text size displayed on the edit screen can be changed.


itled Project)
View Online Debug Diagnostics Tool Window Help
Toolbar
Statusbar
Color and Font...
Docking Window
Comment Ctrl+F5
Statement Ctrl+F7
Note Ctrl+F8
Display Lines of Monitored Current Value(W)...
Display Format for Device Comment(O)...
Non-Display Ladder Block Ctrl+Num -
Display Ladder Block Ctrl+Num +
Non-Display All Ladder Block
Display All Ladder Block
Device Display Ctrl+Alt+F6
All Device Display
Cancel All Device Display
Display Compile Result...
Zoom...
Text Size
Tile FB Horizontally Ctrl+Shift+Enter
Open Header
Bigger
Smaller

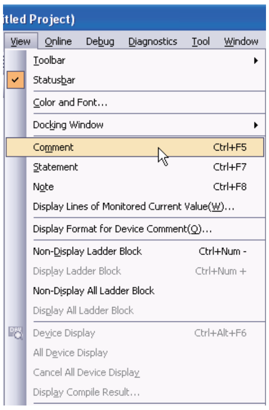
1) Select [View] → [Text Size] → [Bigger]/[Smaller].

The text size is changed one step at each setting within the range of 10 steps.



(4) Displaying/hiding comments

Device comments (label comments), notes, and statements can be displayed and hidden.


itled Project)
View Online Debug Diagnostics Tool Window
Toolbar
Statusbar
Color and Font...
Docking Window
Comment Ctrl+F5
Statement Ctrl+F7
Note Ctrl+F8
Display Lines of Monitored Current Value(W)...
Display Format for Device Comment(O)...
Non-Display Ladder Block Ctrl+Num -
Display Ladder Block Ctrl+Num +
Non-Display All Ladder Block
Display All Ladder Block
Device Display Ctrl+Alt+F6
All Device Display
Cancel All Device Display
Display Compile Result...

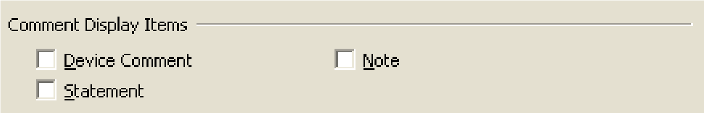
1) Select [View] → [Comment]/[Statement]/[Note].

POINT

Displaying/hiding comments

Comments also can be displayed or hidden by the following operation.

[Tool] → [Option] → "Program Editor" → "Ladder" → "Comment"



Comment Display Items

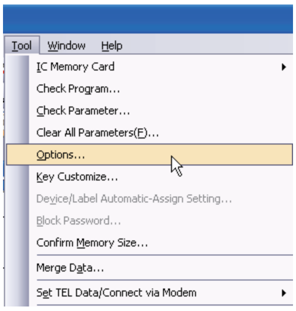
□ Device Comment □ Note

□ Statement

* The details of this operation are explained in the next page.

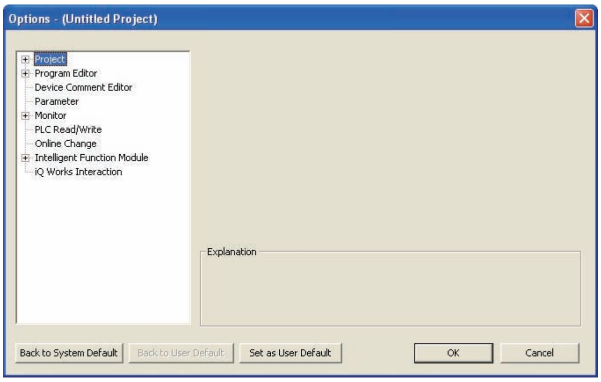
(5) Setting the number of rows and columns for displaying comments

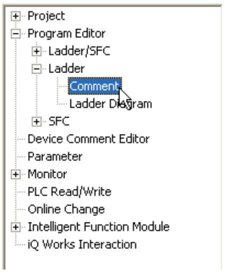
The option setting allows switching the number of rows and columns for displaying a device comment.



1) Click [Tool] → [Option].

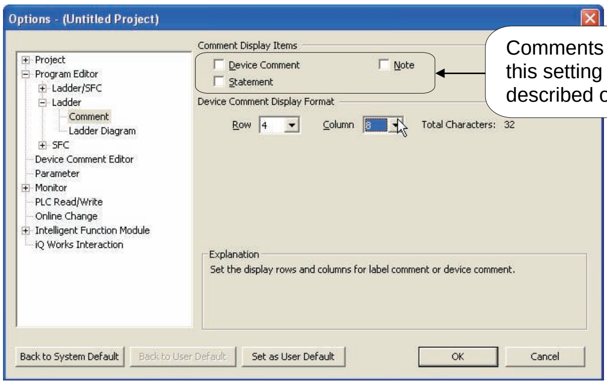
The Options screen is displayed.





2) Click "Program Editor" → "Ladder" → "Comment".

The screen for setting Device Comment Display Format is displayed.



Comment Display Items

□ Device Comment □ Note

□ Statement

Device Comment Display Format

Row 4 Column 8 Total Characters: 32

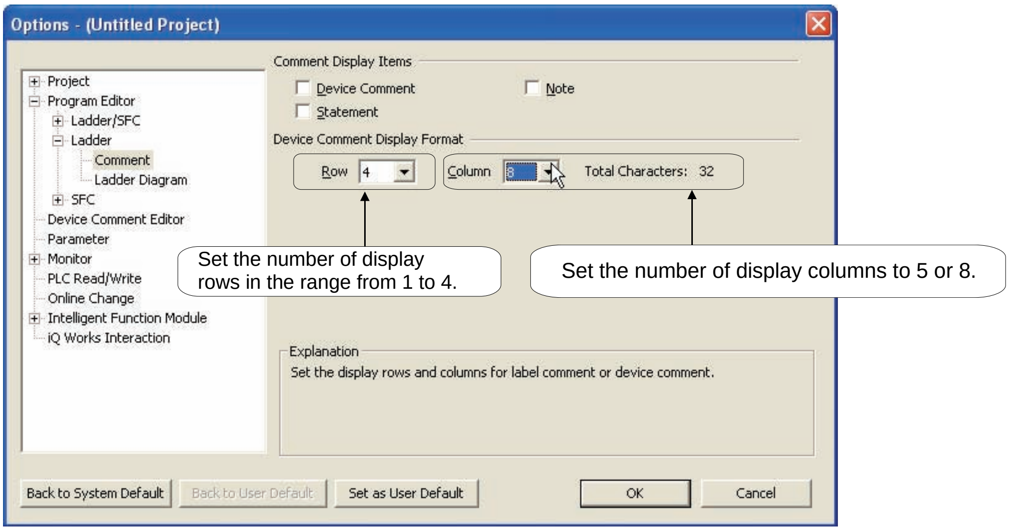
Explanation

Set the display rows and columns for label comment or device comment.

Comments can be displayed or hidden by this setting in addition to by the method described on the previous page.

(To the next page)

(From the previous page)



Options - (Untitled Project)

- Project
- Program Editor
  - Ladder/SFC
  - Ladder
    - Comment
    - Ladder Diagram
  - SFC
- Device Comment Editor
- Parameter
- Monitor
- PLC Read/Write
- Online Change
- Intelligent Function Module
- iQ Works Interaction

Comment Display Items

□ Device Comment □ Note

□ Statement

Device Comment Display Format

Row 4 Column 8 Total Characters: 32

Explanation

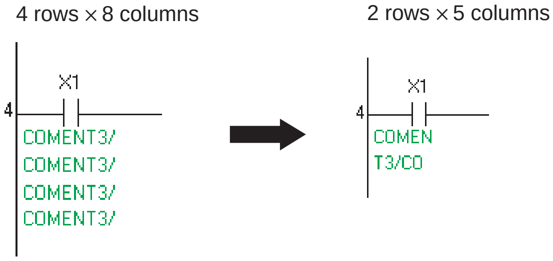
Set the display rows and columns for label comment or device comment.

Back to System Default | Back to User Default | Set as User Default | OK | Cancel

Set the number of display rows in the range from 1 to 4.

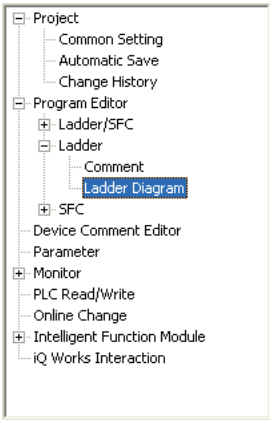
Set the number of display columns to 5 or 8.

Example)



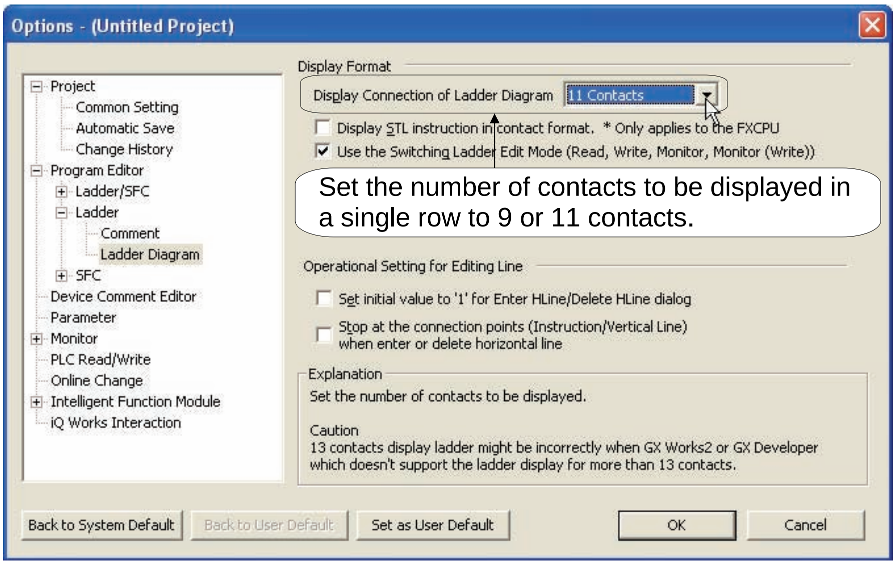
(6) Setting the number of contacts to be displayed in ladder programs

The option setting allows switching the number of contacts to be displayed in a single row.



1) Click "Program Editor" → "Ladder" → "Ladder Diagram" in the Options screen.

The screen for setting Display Format for the ladder diagram is displayed.



Display Format

Display Connection of Ladder Diagram: 11 Contacts

□ Display STL instruction in contact format. * Only applies to the FXCPU

☑ Use the Switching Ladder Edit Mode (Read, Write, Monitor, Monitor (Write))

Set the number of contacts to be displayed in a single row to 9 or 11 contacts.

Operational Setting for Editing Line

□ Set initial value to '1' for Enter HLine/Delete HLine dialog

□ Stop at the connection points (Instruction/Vertical Line) when enter or delete horizontal line

Explanation

Set the number of contacts to be displayed.

Caution

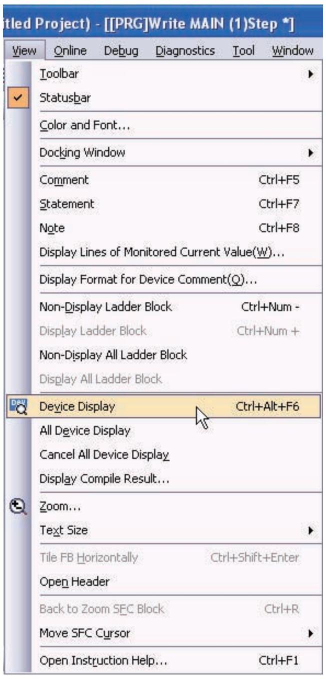
13 contacts display ladder might be incorrectly when GX Works2 or GX Developer which doesn't support the ladder display for more than 13 contacts.

(7) Switching the label name display and device display

The display of a program that uses labels can be switched between the label name display and device display.

If label comments or device comments are set, the corresponding comments are displayed.

Devices assigned by the compilation can be checked by switching the program display from the label name display to the device display.


itled Project) - [[PRG]Write MAIN (1)Step *]
View
Online
Debug
Diagnostics
Tool
Window
Toolbar
Statusbar
Color and Font...
Docking Window
Comment Ctrl+F5
Statement Ctrl+F7
Note Ctrl+F8
Display Lines of Monitored Current Value(W)...
Display Format for Device Comment(O)...
Non-Display Ladder Block Ctrl+Num -
Display Ladder Block Ctrl+Num +
Non-Display All Ladder Block
Display All Ladder Block
Device Display Ctrl+Alt+F6
All Device Display
Cancel All Device Display
Display Compile Result...
Zoom...
Text Size
Tile FB Horizontally Ctrl+Shift+Enter
Open Header
Back to Zoom SFC Block Ctrl+R
Move SFC Cursor
Open Instruction Help... Ctrl+F1

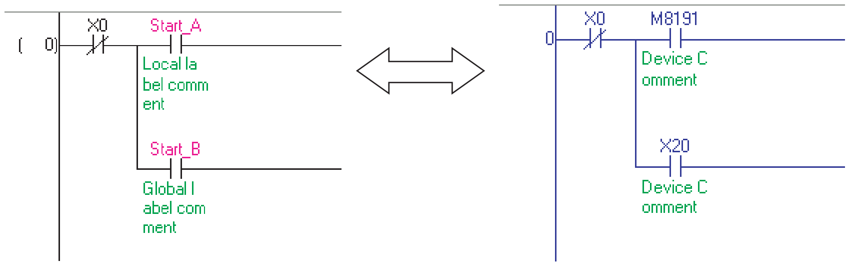
1) Click [View] → [Device Display].

The screen for setting Display Format for the ladder diagram is displayed.

Example)

Label name display

Device display



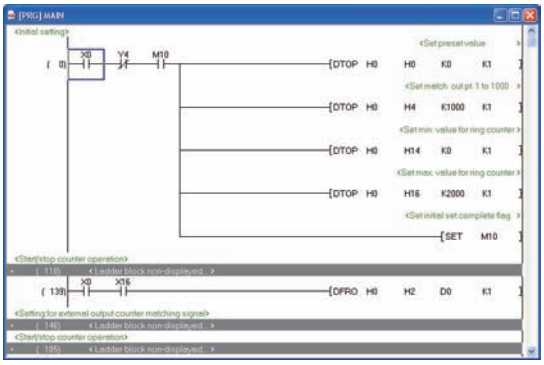
POINT

Displaying/hiding label comments and device comments
To check the set label comments and device comments, set the setting to display comments. (Refer to section 2.2.2 (4))

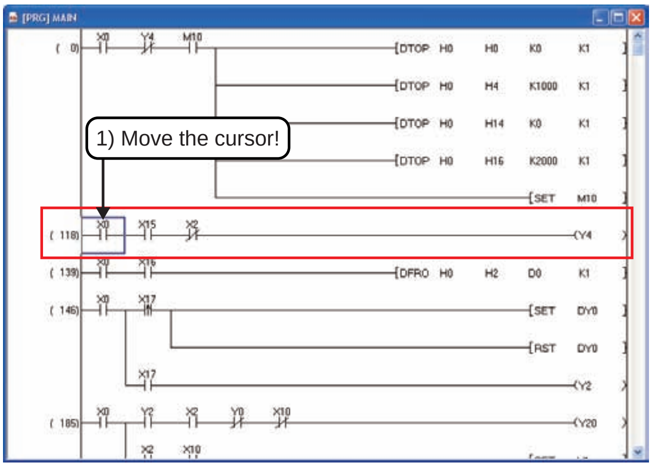
(8) Hiding a ladder block

The ladder block after the ladder conversion can be hidden.

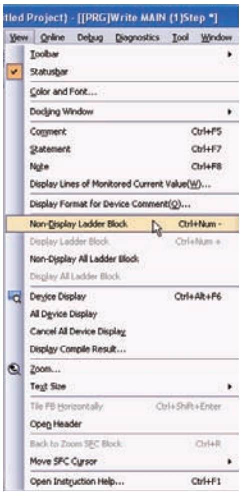
The ladder block in which the statements are set is hidden with the statements displayed.


[PRG] MAIN
<Initial setting>
<Set preset value>
<Set match. out pt 1 to 1000 >
<Set min. value for ring counter>
<Set max. value for ring counter>
<Set initial set complete flag>
<Start/stop counter operation>
(118) < Ladder block non-displayed... >
<Setting for external output counter matching signal>
(146) < Ladder block non-displayed... >
<Start/stop counter operation>
(185) < Ladder block non-displayed... >

(a) Hiding a ladder block



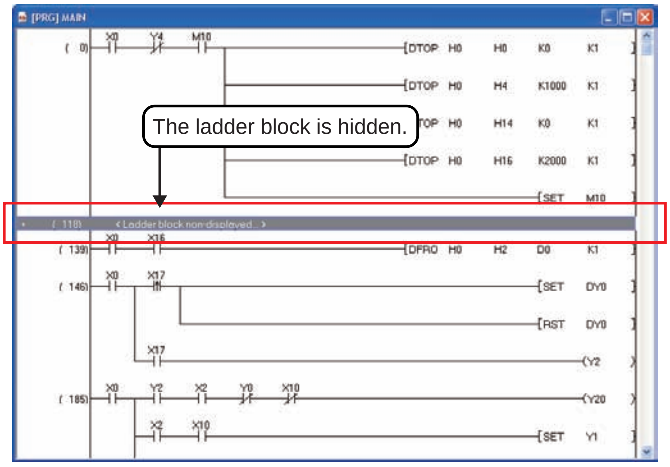
1) Move the cursor to the ladder block to be hidden.


tled Project] - [[PRG]Write MAIN (1)Step *]
View Online Debug Diagnostics Tool Window
Toolbar
Statusbar
Color and Font...
Docking Window
Comment Ctrl+F5
Statement Ctrl+F7
Note Ctrl+F8
Display Lines of Monitored Current Value(W)...
Display Format for Device Comment(Q)...
Non-Display Ladder Block Ctrl+Num -
Display Ladder Block Ctrl+Num +
Non-Display All Ladder Block
Display All Ladder Block
Device Display Ctrl+Alt+F6
All Device Display
Cancel All Device Display
Display Compile Result...
Zoom...
Text Size
Tile FB Horizontally Ctrl+Shift+Enter
Open Header
Back to Zoom SFC Block Ctrl+B
Move SFC Cursor
Open Instruction Help... Ctrl+F1

2) Click [View] → [Non-Display Ladder Block].

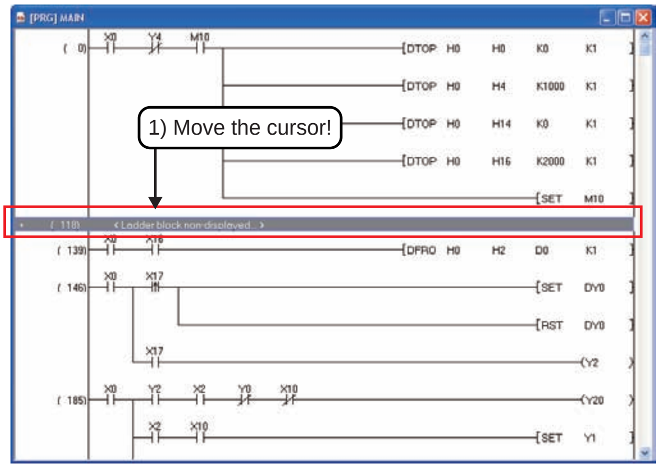
(To the next page)

(From the previous page)

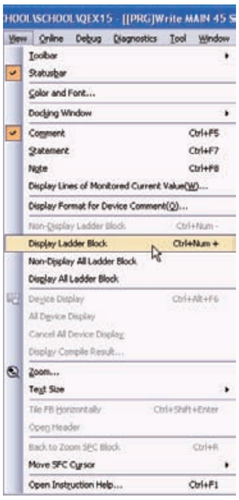

[PRG] MAIN
The ladder block is hidden.
(0) X0 Y4 M10 DTOP H0 H0 K0 K1
DTOP H0 H4 K1000 K1
DTOP H0 H14 K0 K1
DTOP H0 H16 K2000 K1
SET M10
(118) < Ladder block non-displayed... >
(139) X0 X16 DFRO H0 H2 D0 K1
(146) X0 X17 SET DY0
RST DY0
X17 Y2
(185) X0 Y2 X2 Y0 X10 Y20
X2 X10 SET Y1

3) The selected ladder blocks are hidden.

(b) Canceling the hidden ladder block.



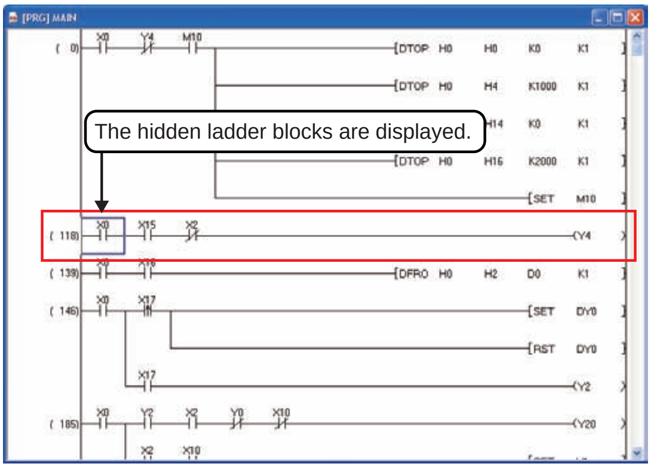
1) Move the cursor to the hidden ladder block displayed in gray.


HOOL\SCHOOL\QEX15 - [[PRG]Write MAIN 45 S
View Online Debug Diagnostics Tool Window
Toolbar
Statusbar
Color and Font...
Docking Window
Comment Ctrl+F5
Statement Ctrl+F7
Note Ctrl+F8
Display Lines of Monitored Current Value(W)...
Display Format for Device Comment(O)...
Non-Display Ladder Block Ctrl+Num -
Display Ladder Block Ctrl+Num +
Non-Display All Ladder Block
Display All Ladder Block
Device Display Ctrl+Alt+F6
All Device Display
Cancel All Device Display
Display Compile Result...
Zoom...
Text Size
Tile FB Horizontally Ctrl+Shift+Enter
Open Header
Back to Zoom SFC Block Ctrl+R
Move SFC Cursor
Open Instruction Help... Ctrl+F1

2) Click [View] → [Display Ladder Block].

(To the next page)

(From the previous page)



The hidden ladder blocks are displayed.

3) The hidden ladder blocks are displayed.

POINT

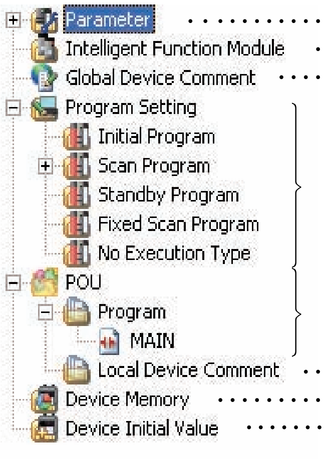
Displaying/hiding ladder blocks

- Multiple ladder blocks also can be displayed and hidden.
- All ladder blocks can be displayed and hidden by the operation of [View] → [Display All Ladder Block]/[Non-Display All Ladder Block].
- Ladder blocks also can be displayed and hidden by Right-click → [Displaying/hiding ladder blocks].

2.2.3 Project

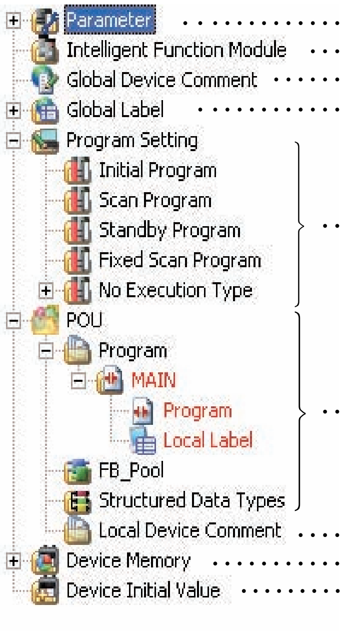
This section explains the configurations of a project that is displayed in a tree format in the Project view. The display contents differ according to the programmable controller type and the project type. The following is an example for a Simple project of QCPU (Q mode).

< Simple project (without labels) >



· · · · · · · · · · · · · · Set various parameters.
· · · · Make settings for the intelligent function modules.
· · · · · · · Set global device comments.
· · · Set an execution type of each program.
Create programs.
· · · ☞ GX Works2 Version 1 Operating Manual Simple Project
· · · · · Set local device comments.
· · · · · · · · · · · · Make settings for device memory.
· · · · · · · · · · Set device initial values.

< Simple project (with labels) >



· · · · · · · · · · · · · · Set various parameters.
· · · · Make settings for the intelligent function modules.
· · · · · · · Set global device comments.
· · · · · · · · · · · · · Set global labels.
☞ GX Works2 Version 1 Operating Manual Simple Project
· · · Set an execution type of each program.
· · · Create programs.
☞ GX Works2 Version 1 Operating Manual Simple Project
· · · · · Set local device comments.
· · · · · · · · · · · · Make settings for device memory.
· · · · · · · · · · Set device initial values.

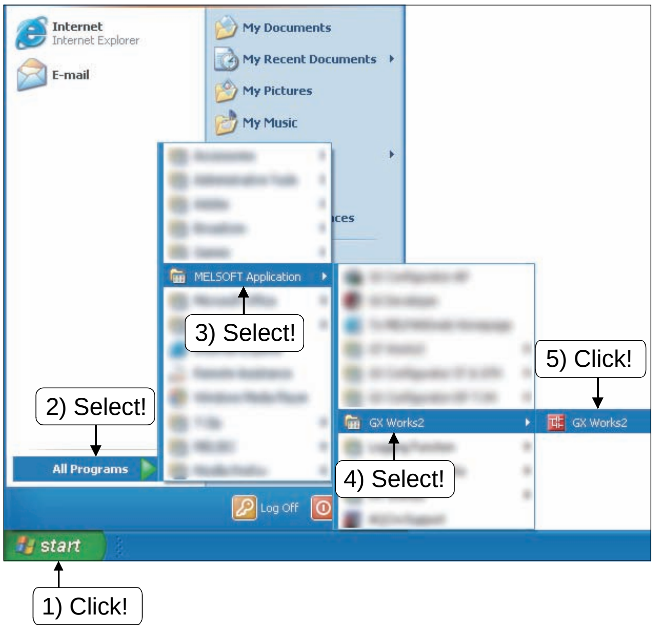
1) One project per GX Works2
One GX Works2 can edit only one project unit.
To edit two or more projects at a time, run as many GX Works2 as the number of projects.

2) Device comments
Device comment of GX Works2 is categorized into global device comment and local device comment.

Comment type	Number of comments	Description
Global comment	1	A device comment created automatically when a new project is created. Global comments are set to use common device comment data among multiple programs.
Local comment	Equals the number of the programs.	A device comment created by the user. No local device comment exists when a new project is created. Therefore, create a local device comment if necessary. Set the same names as sequence programs.

2.3 Operation Before Creating Ladder Program

2.3.1 Starting up GX Works2



1) Click the start button.

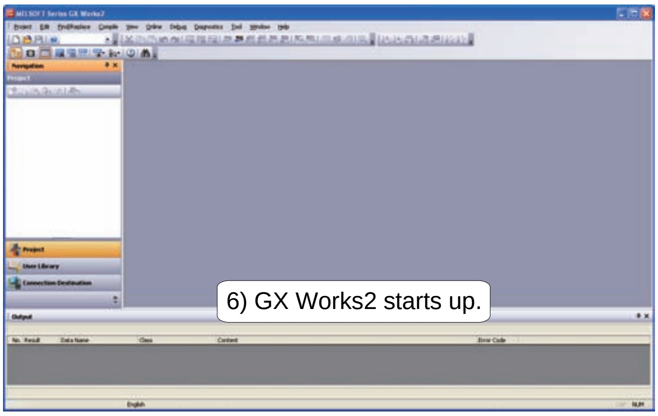
2) Select [All Programs].

3) Select [MELSOFT Application].

4) Select [GX Works2].

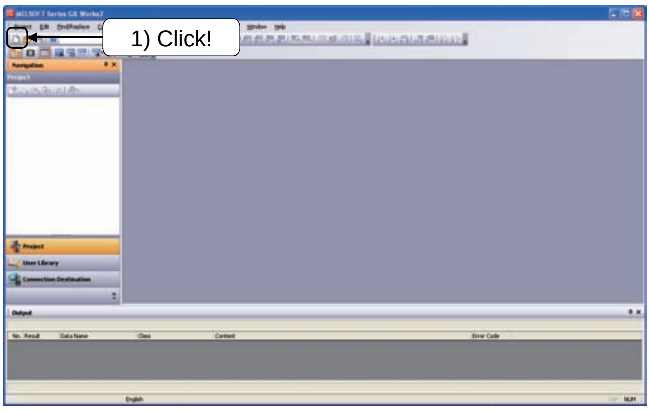
Put the mouse cursor over the items to select the menu.
(Clicking or double-clicking the mouse is not required.)

5) Click [GX Works2].

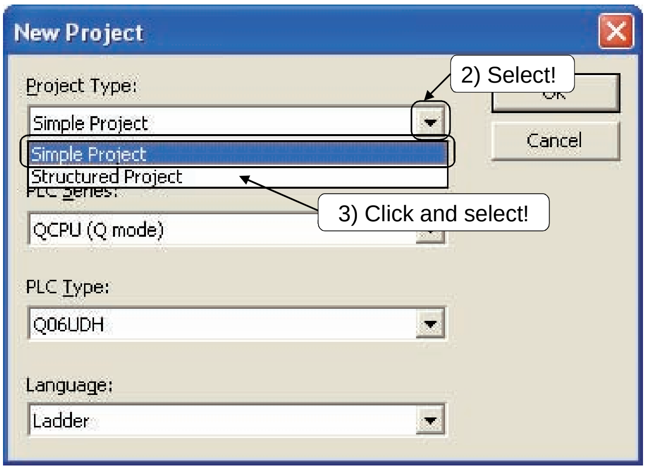


6) GX Works2 starts up.

2.3.2 Creating a new project



1) Click 🗋 on the toolbar or select [Project] → [New Project] (Ctrl+N).



Project Type: Simple Project

- Simple Project
- Structured Project

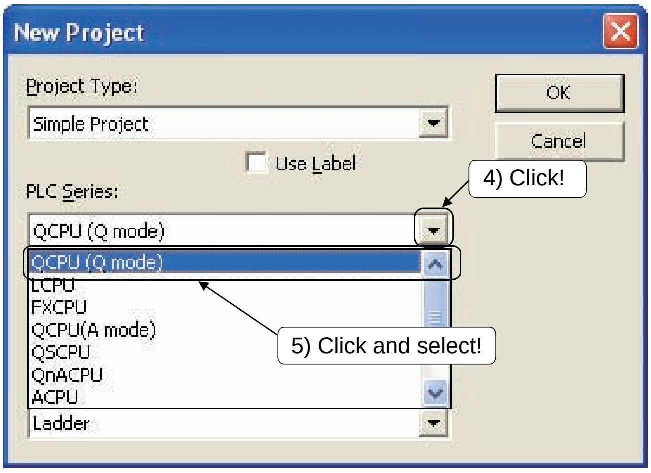
PLC Series: QCPU (Q mode)

PLC Type: Q06UDH

Language: Ladder

2) Click the "Project Type" list button.

3) The "Project Type" list is displayed. Select "Simple Project".



Project Type: Simple Project

☐ Use Label

PLC Series: QCPU (Q mode)

- QCPU (Q mode)
- LCPU
- FXCPU
- QCPU(A mode)
- QSCPU
- QnACPU
- ACPU

Ladder

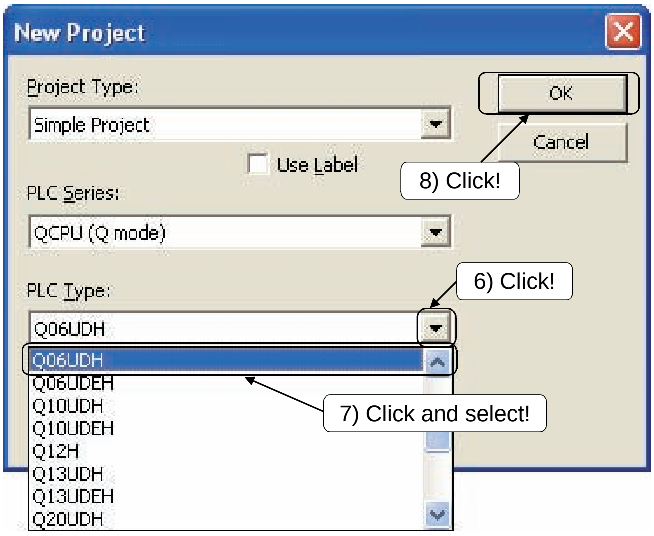
4) Click the "PLC Series" list button.

5) The "PLC Series" list is displayed. Select "QCPU (Q mode)".

(To the next page)

(From the previous page)

⇩



Project Type: Simple Project

□ Use Label

PLC Series: QCPU (Q mode)

PLC Type: Q06UDH

8) Click!

6) Click!

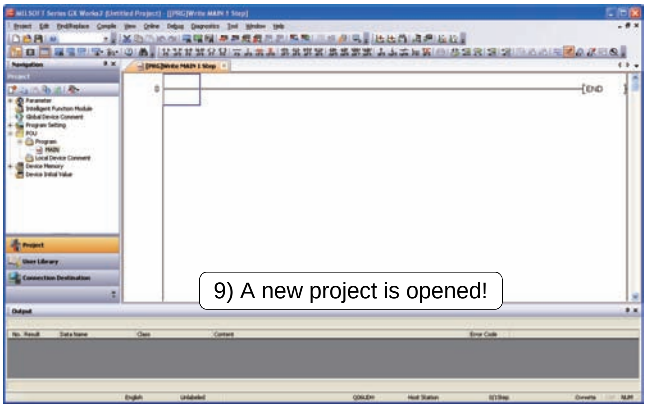
7) Click and select!

6) Click the "PLC Type" list button.

7) The "PLC Type" list is displayed. Select "Q06UDH".

8) Click the OK button.

⇩



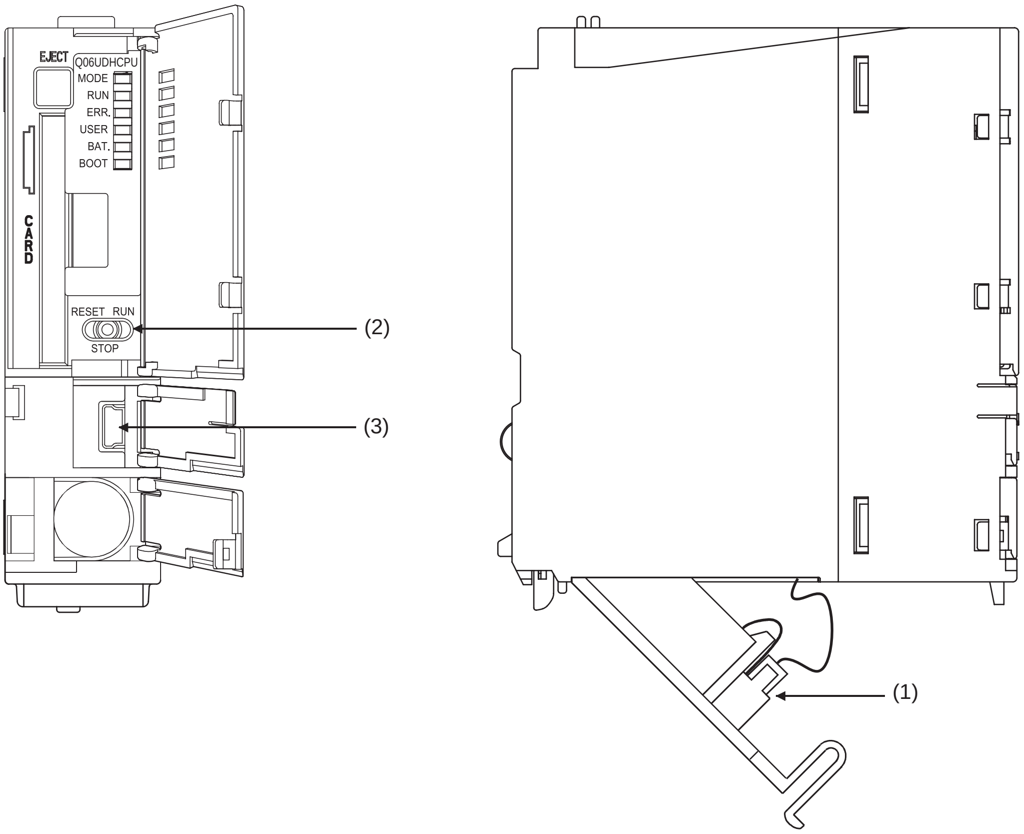
9) A new project is opened!

9) A new project is opened.

2.4 Preparation for Starting Up CPU

Setting switches and formatting the built-in memory are required before writing a program to the CPU.

Connect or set the connectors and the switches of (1) to (3) shown below.
(The figures below are example of Q06UDHCPU.)



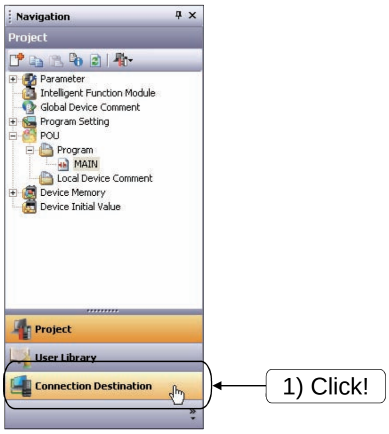
(1) Connecting a battery
Connect the battery since the lead wire of the battery connector is disconnected at the factory shipment.

(2) Setting the switches
Set the RUN/STOP/RESET switch to the STOP position.

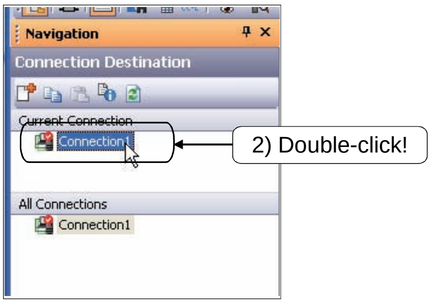
(3) Connecting the USB cable

(4) Setting the connection destination

This section explains how to set the connection destination for accessing the programmable controller CPU.

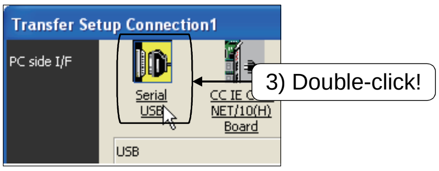


1) Click "Connection Destination" in the view selection area on the navigation window.

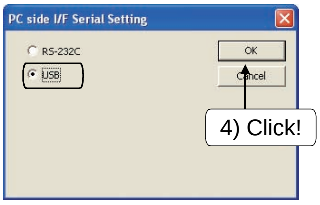


2) The Connection Destination view is displayed. Double-click "Connection1" in Current Connection.

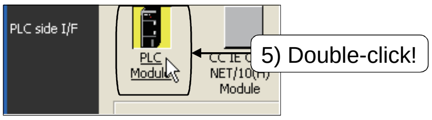
The Transfer Setup dialog box is displayed.



3) Double-click "Serial USB" of PC side I/F.



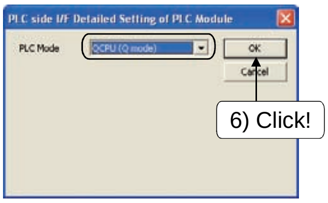
4) The PC side I/F Serial Setting dialog box is displayed. Check "USB" and click the OK button.



5) Double-click "PLC Module" of PLC side I/F.

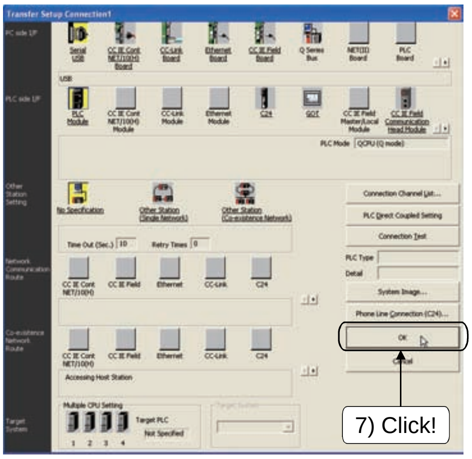
(To the next page)

(From the previous page)



PLC Mode: QCPU (Q mode)

6) The PLC side I/F Detailed Setting of PLC Module dialog box is displayed. Select "QCPU (Q mode) " and click the OK button.


Transfer Setup Connection1
PC side I/F
Serial USB
CC IE Cont NET/10(H) Board
CC-Link Board
Ethernet Board
CC IE Field Board
Q Series Bus
NET(II) Board
PLC Board
PLC side I/F
PLC Module
CC IE Cont NET/10(H) Module
CC-Link Module
Ethernet Module
C24
GOT
CC IE Field Master/Local Module
CC IE Field Communication Head Module
Other Station Setting
No Specification
Other Station (Single Network)
Other Station (Co-existence Network)
Network Communication Route
Co-existence Network Route
Accessing Host Station
Multiple CPU Setting
Target PLC
Not Specified
Target System
Connection Channel List...
PLC Direct Coupled Setting
Connection Test
PLC Type
Detail
System Image...
Phone Line Connection (C24)...
OK
Cancel
7) Click!

USB

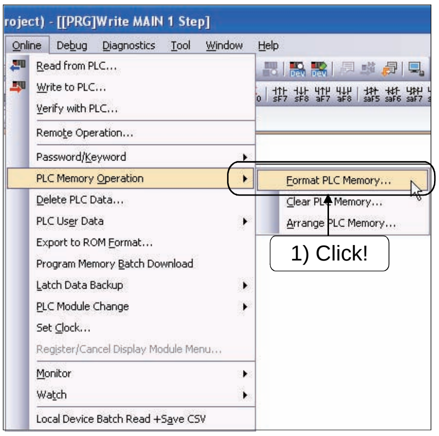
PLC Mode: QCPU (Q mode)

Time Out (Sec.) 10 Retry Times 0

7) Click the OK button.

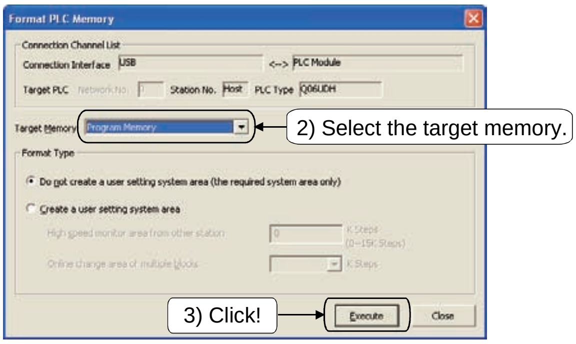
(5) Formatting the built-in memory of the CPU

This section explains how to format the program memory of the QCPU.



1) Click!

1) Click [Online] → [PLC Memory Operation] → [Format PLC Memory].



Connection Interface USB <--> PLC Module

Target PLC Station No. Host PLC Type Q06UDH

Target Memory Program Memory

Format Type

● Do not create a user setting system area (the required system area only)

○ Create a user setting system area

High speed monitor area from other station 0 K Steps

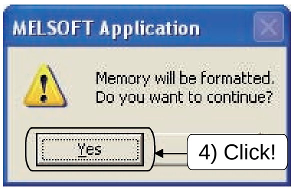
Online change area of multiple blocks K Steps

2) Select the target memory.

3) Click!

2) The Format PLC Memory dialog box is displayed. Select "Program Memory" from the Target Memory drop-down menu.

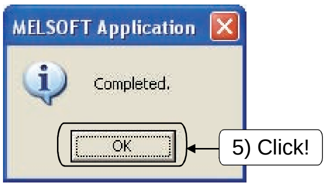
3) Click the Execute button.



Memory will be formatted.
Do you want to continue?

4) Click!

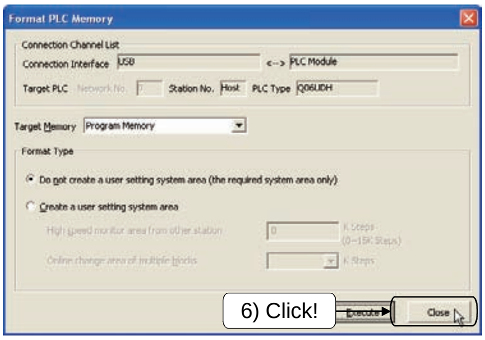
4) Click the Yes button to start formatting.



Completed.

5) Click!

5) When format is completed, the dialog box on the left is displayed. Click the OK button.



Connection Interface USB <--> PLC Module

Target PLC Station No. Host PLC Type Q06UDH

Target Memory Program Memory

Format Type

● Do not create a user setting system area (the required system area only)

○ Create a user setting system area

High speed monitor area from other station 0 K Steps

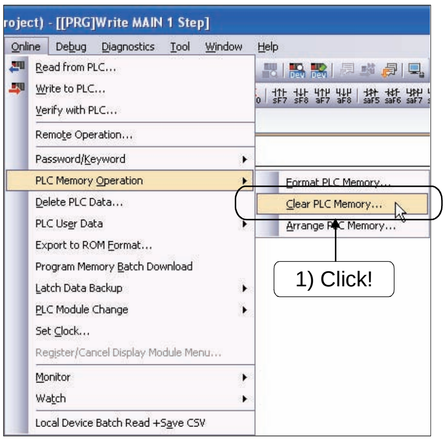
Online change area of multiple blocks K Steps

6) Click!

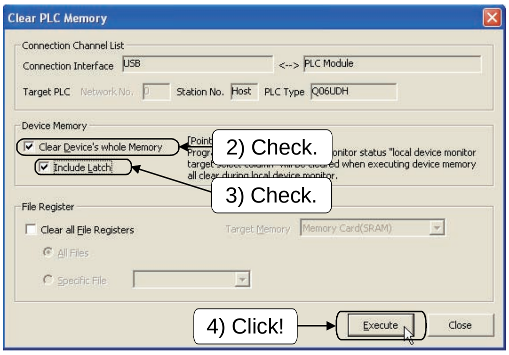
6) Click the Close button to close the dialog box.

(6) Clearing all the device memory from the CPU

This section explains how to clear the device memory of the QCPU.



1) Click [Online] → [PLC Memory Operation] → [Clear PLC Memory].



Connection Channel List

Connection Interface: USB <--> PLC Module

Target PLC: Network No. 0, Station No. Host, PLC Type Q06UDH

Device Memory

☑ Clear Device's whole Memory

☑ Include Latch

File Register

☐ Clear all File Registers

Target Memory: Memory Card(SRAM)

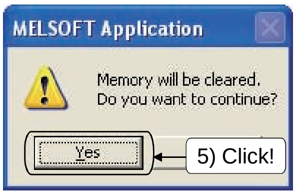
● All Files

○ Specific File

2) The Clear PLC Memory dialog box is displayed. Check that "Clear Device's whole Memory" is checked.

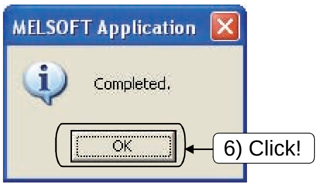
3) Check "Include Latch".

4) Click the Execute button.



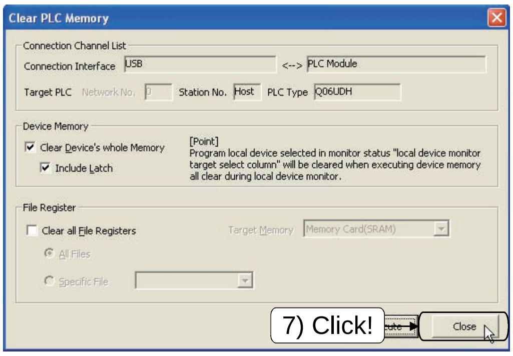
Memory will be cleared.
Do you want to continue?

5) Click the Yes button to clear the latch device.



Completed.

6) When the clearing the latch device is completed, the dialog box on the left is displayed. Click the OK button.



Connection Channel List

Connection Interface: USB <--> PLC Module

Target PLC: Network No. 0, Station No. Host, PLC Type Q06UDH

Device Memory

☑ Clear Device's whole Memory

☑ Include Latch

[Point]
Program local device selected in monitor status "local device monitor target select column" will be cleared when executing device memory all clear during local device monitor.

File Register

☐ Clear all File Registers

Target Memory: Memory Card(SRAM)

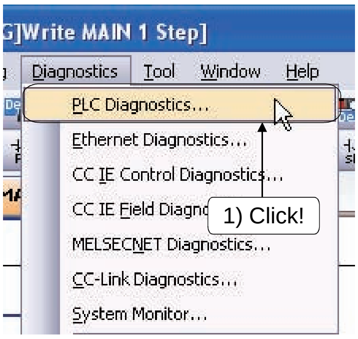
● All Files

○ Specific File

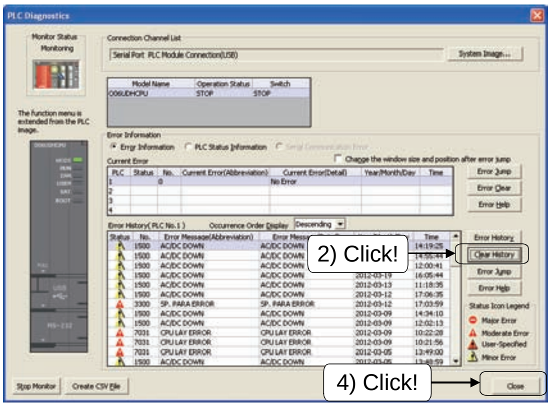
7) Click the Close button to close the dialog box.

(7) Clearing the error history in the CPU

This section explains how to clear the error history data stored in the QCPU.



1) Click [Diagnostics] → [PLC Diagnostics].



Serial Port PLC Module Connection(USB)

Model Name	Operation Status	Switch
Q06UDHCPU	STOP	STOP

◉ Error Information ○ PLC Status Information ○ Serial Communication Error

Current Error

□ Change the window size and position after error jump

PLC	Status	No.	Current Error(Abbreviation)	Current Error(Detail)	Year/Month/Day	Time
1		0		No Error		
2						
3						
4						

Error History(PLC No.1) Occurrence Order Display Descending

Status	No.	Error Message(Abbreviation)	Error Message		Time
⚠	1500	AC/DC DOWN	AC/DC DOWN		14:19:25
⚠	1500	AC/DC DOWN	AC/DC DOWN		14:55:44
⚠	1500	AC/DC DOWN	AC/DC DOWN		12:00:41
⚠	1500	AC/DC DOWN	AC/DC DOWN	2012-03-19	16:05:44
⚠	1500	AC/DC DOWN	AC/DC DOWN	2012-03-13	11:18:35
⚠	1500	AC/DC DOWN	AC/DC DOWN	2012-03-12	17:06:35
▲	3300	SP. PARA ERROR	SP. PARA ERROR	2012-03-12	17:03:59
⚠	1500	AC/DC DOWN	AC/DC DOWN	2012-03-09	14:34:10
⚠	1500	AC/DC DOWN	AC/DC DOWN	2012-03-09	12:02:13
▲	7031	CPU LAY ERROR	CPU LAY ERROR	2012-03-09	10:22:28
▲	7031	CPU LAY ERROR	CPU LAY ERROR	2012-03-09	10:21:56
▲	7031	CPU LAY ERROR	CPU LAY ERROR	2012-03-05	13:49:00
⚠	1500	AC/DC DOWN	AC/DC DOWN	2012-03-05	13:48:59

2) The PLC Diagnostics dialog box is displayed. Click the Clear History button.

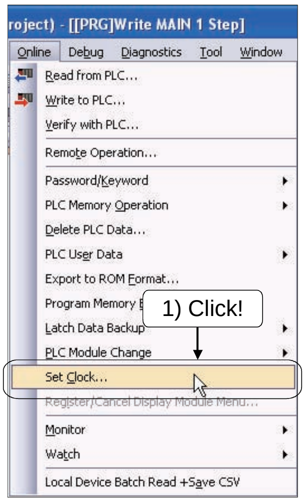
3) The confirmation dialog box is displayed. Click the Yes button.

4) Click the Close button to close the dialog box.

(8) Setting the clock on the programmable controller CPU

Setting a year, month, date, time, minute, second, and day of the week to the clock on the programmable controller CPU is available.
To use the clock function, use GX Works2 or a sequence program.
Set or read the clock data in GX Works2.

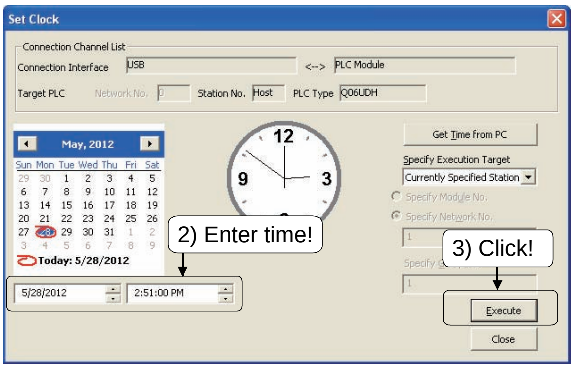
1) Click [Online] → [Set Clock] to display the Set Clock dialog box.



1) Click!

2) Enter a year, month, date, time, minute, second, and day of the week in the Set Clock dialog box.

3) Click the Execute button.



Connection Interface: USB <--> PLC Module

Target PLC Network No. 0 Station No. Host PLC Type Q06UDH

Today: 5/28/2012

5/28/2012 2:51:00 PM

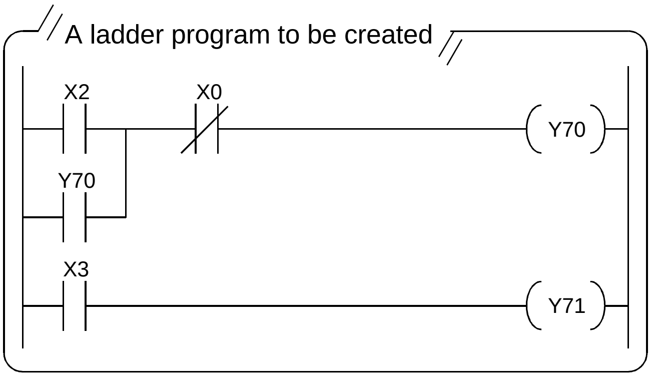
Specify Execution Target: Currently Specified Station

2) Enter time!

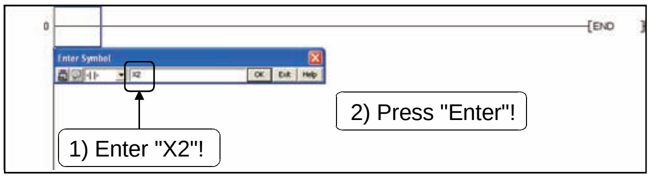
3) Click!

2.5 Creating Ladder Program

2.5.1 Creating a ladder program using the function keys

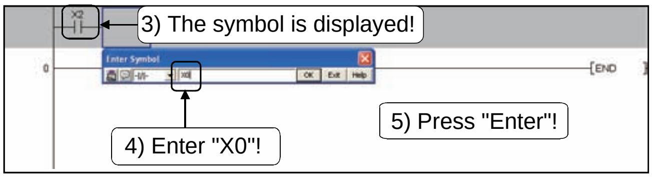


Follow the steps below to create the ladder program as shown on the left.



1) Press the F5 key to open the Enter Symbol window. Enter "X2".
   If any other key is pressed by mistake, press the Esc key and retype.
2) Press the Enter key to confirm the entry.

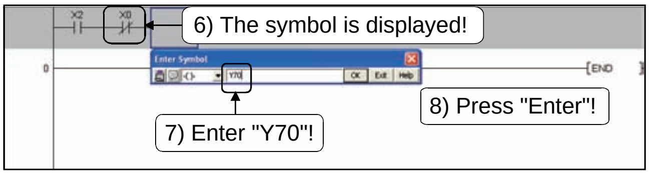
- The OK or Exit button also can be used to confirm or cancel the entry.



3) The entered symbol ($\overset{\text{X2}}{\dashv\vdash}$) is displayed.

4) Press the Shift + F5 keys , and enter "X0".

5) Press the Enter key to confirm the entry.



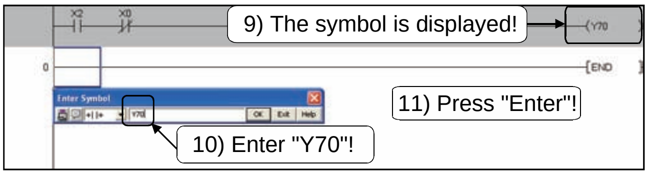
6) The entered symbol ($\overset{\text{X0}}{\dashv\!/\!\vdash}$) is displayed.

7) Press the F7 key, and enter "Y70".

8) Press the Enter key to confirm the entry.

(To the next page)

(From the previous page)



9) The symbol is displayed!

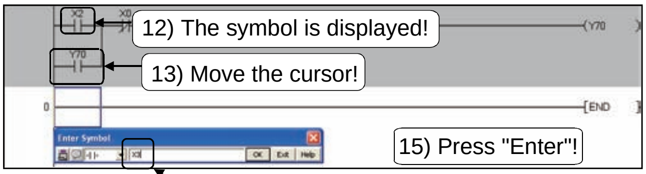
11) Press "Enter"!

10) Enter "Y70"!

9) The entered symbol (–(Y70)–) is displayed.

10) Press the F6 key, and enter "Y70".

11) Press the Enter key to confirm the entry.



12) The symbol is displayed!

13) Move the cursor!

15) Press "Enter"!

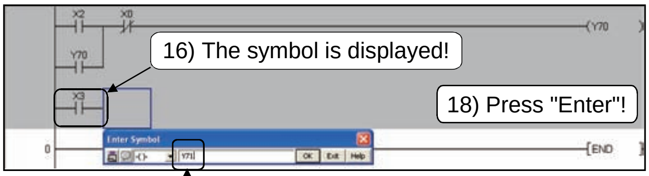
14) Enter "X3"!

12) The entered symbol ($\overset{\text{Y70}}{\dashv\vdash}$) is displayed.

13) Move the cursor to the symbol under $\overset{\text{Y70}}{\dashv\vdash}$.

14) Press the F5 key, and enter "X3".

15) Press the Enter key to confirm the entry.



16) The symbol is displayed!

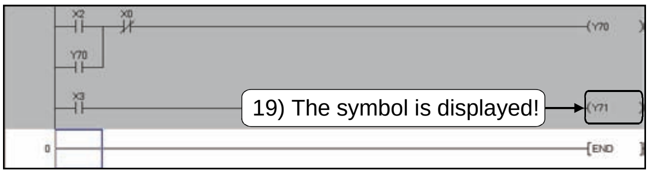
18) Press "Enter"!

17) Enter "Y71"!

16) The entered symbol ($\overset{\text{X3}}{\dashv\vdash}$) is displayed.

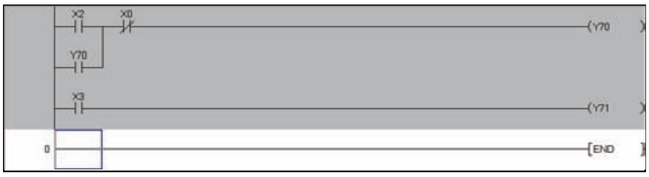
17) Press the F7 key, and enter "Y71".

18) Press the Enter key to confirm the entry.



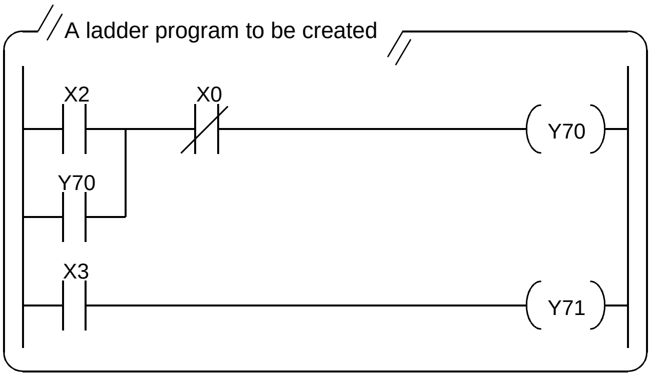
19) The symbol is displayed!

19) The entered symbol (–(Y71)–) is displayed.

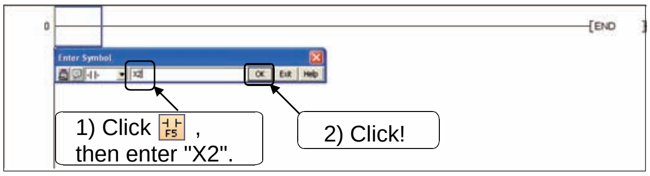


20) The procedure is finished.

2.5.2 Creating a ladder program using the tool buttons

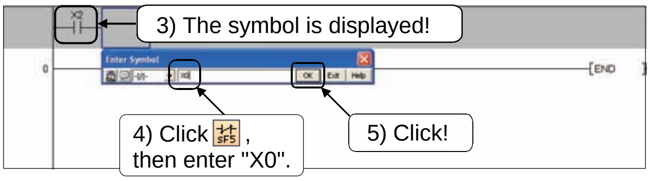


Follow the steps below to create the ladder program as shown on the left.



1) Click F5 on the toolbar to open the Enter Symbol window. Enter "X2".
If any other button is pressed by mistake, click the Exit button.

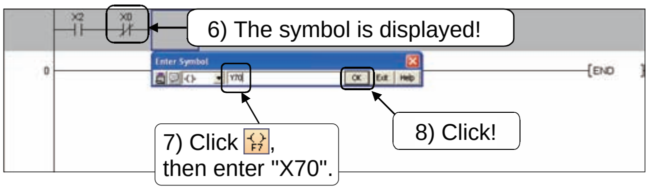
2) Click the OK button to confirm the entry.



3) The entered symbol ($\overset{\text{X2}}{\dashv\vdash}$) is displayed.

4) Click sF5 on the toolbar, and enter "X0".

5) Click the OK button.



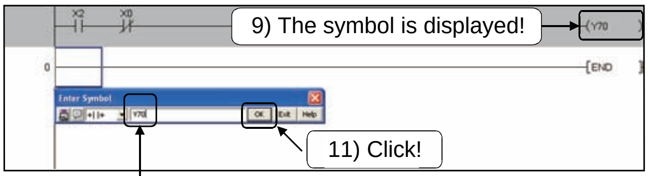
6) The entered symbol ($\overset{\text{X0}}{\dashv\!/\!\vdash}$) is displayed.

7) Click F7 on the toolbar, and enter "Y70".

8) Click the OK button.

(To the next page)

(From the previous page)



9) The symbol is displayed!

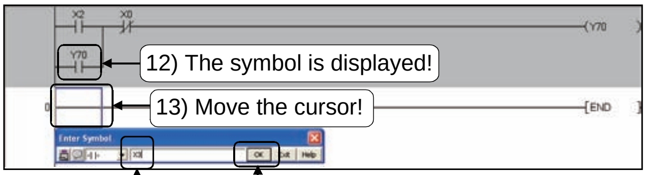
11) Click!

10) Click F6 , then enter "X0".

9) The entered symbol (–(Y70)–) is displayed.

10) Click F6 on the toolbar, and enter "Y70".

11) Click the OK button.



12) The symbol is displayed!

13) Move the cursor!

14) Click F5 , then enter "X3".

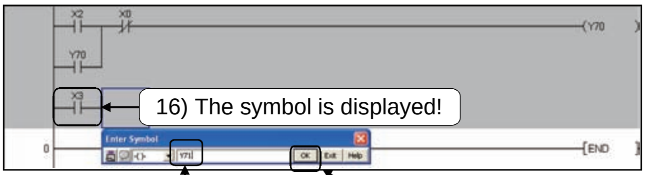
15) Click!

12) The entered symbol ($\overset{\text{Y70}}{┤├}$) is displayed.

13) Move the cursor to the symbol under $\overset{\text{Y70}}{┤├}$.

14) Click F5 on the toolbar, and enter "X3".

15) Click the OK button.



16) The symbol is displayed!

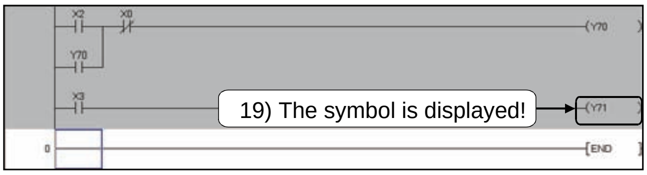
17) Click F7 , then enter "Y71"!

18) Click!

16) The entered symbol ($\overset{\text{X3}}{┤├}$) is displayed.

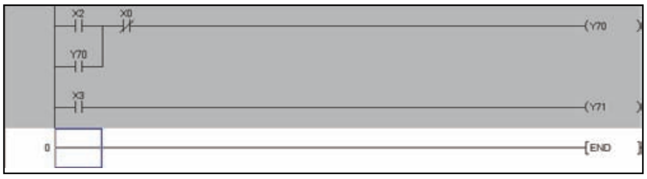
17) Click F7 on the toolbar, and enter "Y71".

18) Click the OK button.



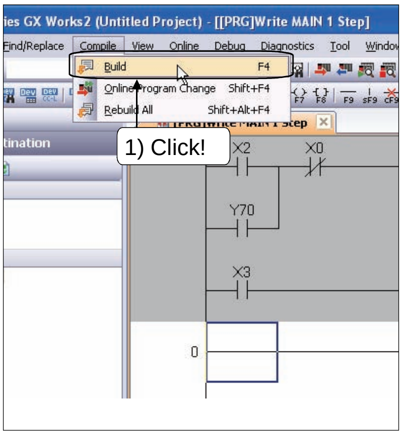
19) The symbol is displayed!

19) The entered symbol (–(Y71)–) is displayed.



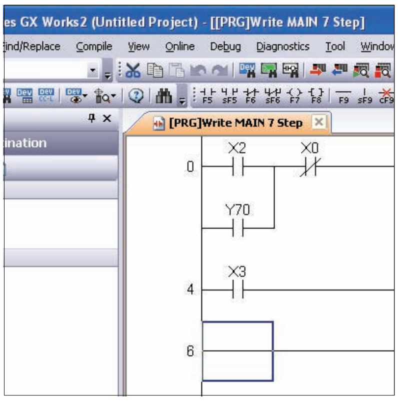
20) The procedure is finished.

2.6 Converting Program (Ladder Conversion)



1) Click!

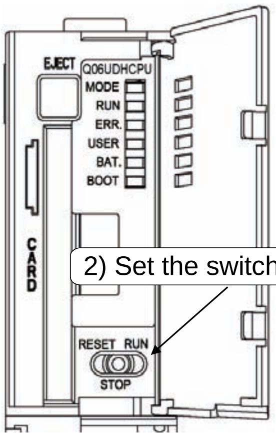
1) Click [Compile] → [Build] (F4).



2) The ladder program has been converted.

If an error occurs during a conversion, the cursor will automatically move to the defective point of the ladder program. Check the point and correct the program as necessary.

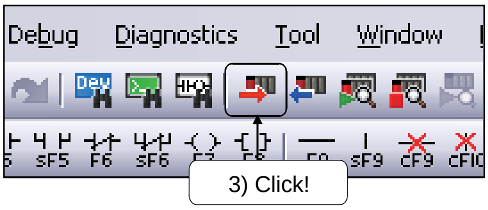
(1) Writing data to the CPU



2) Set the switch to "STOP"!

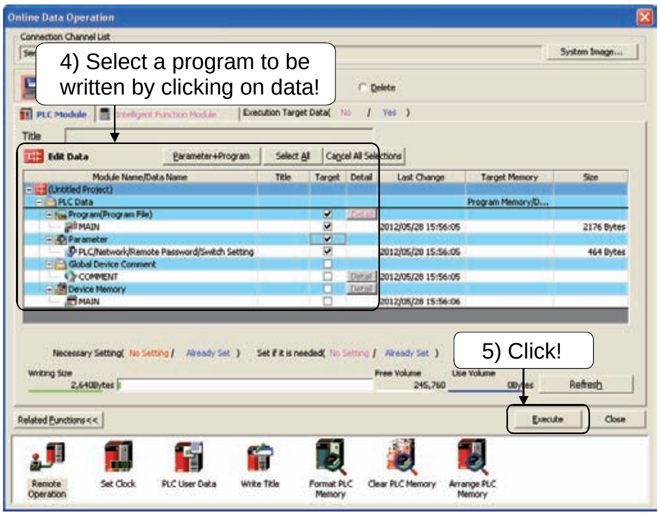
1) Suppose that the ladder program (sequence program) has been created with GX Works2 to proceed to the next step.

2) Set the RUN/STOP/RESET switch on the CPU to STOP.



3) Click!

3) Click on the toolbar or click [Online] → [Write to PLC].



Module Name/Data Name	Title	Target	Detail	Last Change	Target Memory	Size
(Untitled Project)						
PLC Data					Program Memory/D...	
Program(Program File)		☑	Detail			
MAIN		☑		2012/05/28 15:56:05		2176 Bytes
Parameter		☑				
PLC/Network/Remote Password/Switch Setting		☑		2012/05/28 15:56:05		464 Bytes
Global Device Comment		□				
COMMENT		□	Detail	2012/05/28 15:56:05		
Device Memory		□	Detail			
MAIN		□		2012/05/28 15:56:06		

Necessary Setting(No Setting / Already Set) Set if it is needed(No Setting / Already Set)

Writing Size 2,640Bytes

Free Volume 245,760

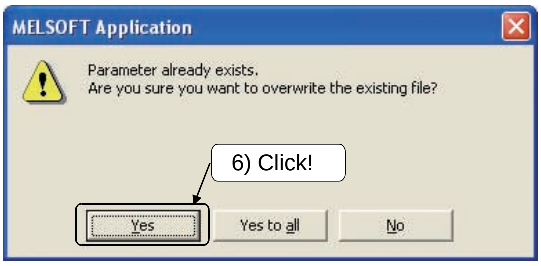
Use Volume 0Bytes

4) Select a program to be written by clicking on data!

5) Click!

4) From the "PLC Module" tab, click to select the program and parameter to write to the CPU. Or click Parameter + Program to select the target program and parameter.

5) Click Execute to accept the selection.



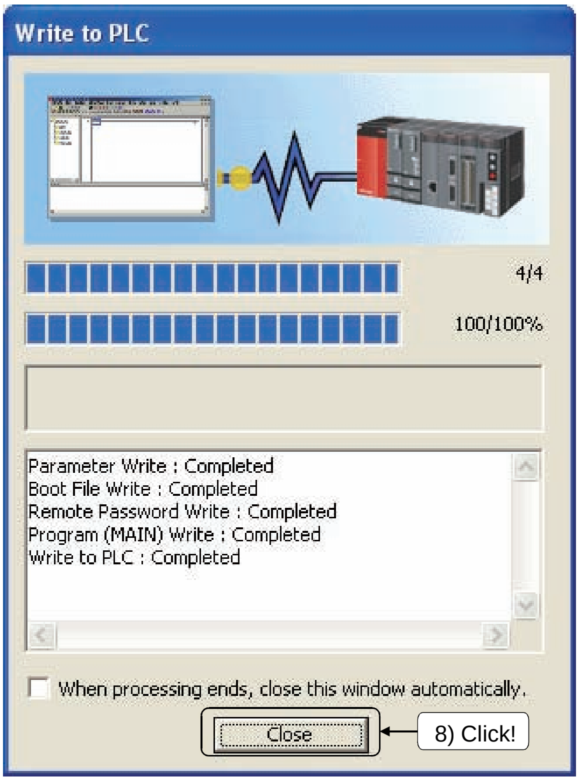
Parameter already exists.
Are you sure you want to overwrite the existing file?

6) Click!

6) If a parameter or program has already been written, the confirmation dialog box for overwriting the data is displayed. Click Yes .

(To the next page)

(From the previous page)



Write to PLC

4/4

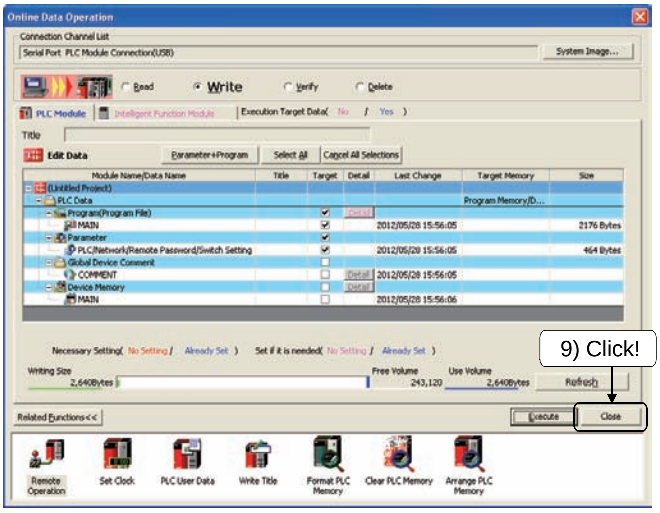
100/100%

Parameter Write : Completed
Boot File Write : Completed
Remote Password Write : Completed
Program (MAIN) Write : Completed
Write to PLC : Completed

□ When processing ends, close this window automatically.

7) The progress dialog box is displayed.

8) The message "Completed" is displayed when the writing is completed. Click Close .



Online Data Operation

Connection Channel List: Serial Port PLC Module Connection(USB)

○ Read ● Write ○ Verify ○ Delete

PLC Module | Intelligent Function Module | Execution Target Data(No / Yes)

Title

Edit Data

Module Name/Data Name	Title	Target	Detail	Last Change	Target Memory	Size
(Untitled Project)						
PLC Data					Program Memory/D...	
Program(Program File)		☑	Detail			
MAIN		☑		2012/05/28 15:56:05		2176 Bytes
Parameter		☑				
PLC/Network/Remote Password/Switch Setting		☑		2012/05/28 15:56:05		464 Bytes
Global Device Comment		□				
COMMENT		□	Detail	2012/05/28 15:56:05		
Device Memory		□	Detail			
MAIN		□		2012/05/28 15:56:06		

Necessary Setting(No Setting / Already Set) Set if it is needed(No Setting / Already Set)

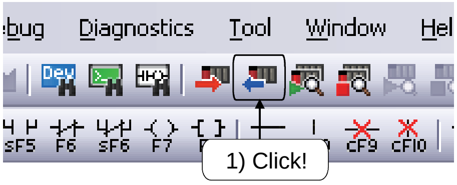
Writing Size 2,640Bytes

Free Volume 243,120

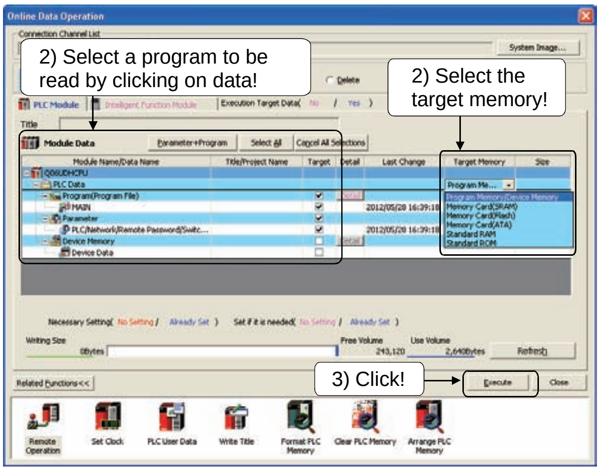
Use Volume 2,640Bytes

9) Click the Close button to close the dialog box.

(2) Reading data from the CPU



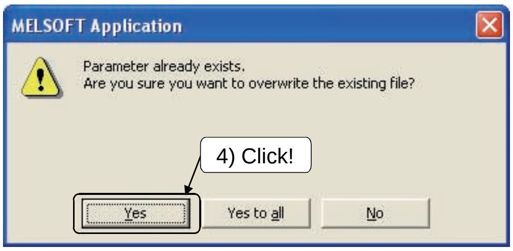
1) Click the icon on the toolbar or click [Online] → [Read from PLC].



2) From the "PLC Module" tab, click to select the program and parameter to read from the CPU. Or click [Parameter + Program] to select the target program and parameter.

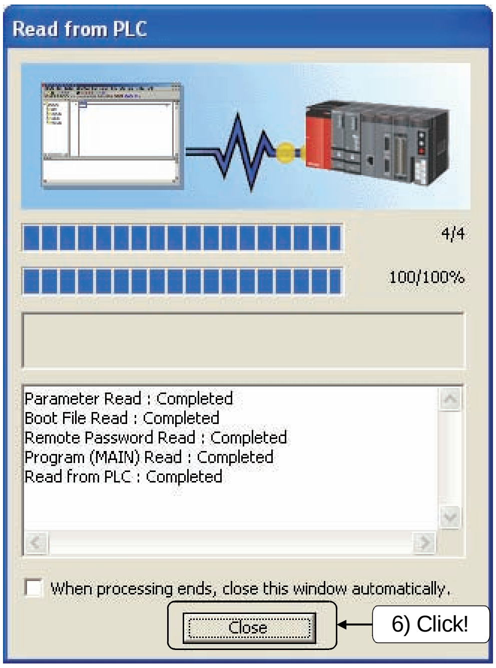
Select "Program Memory/Device Memory" for "Target Memory".

3) Click [Execute] to accept the selection.


MELSOFT Application

Parameter already exists.
Are you sure you want to overwrite the existing file?

4) If a parameter or program exits, the confirmation dialog box for overwriting the data is displayed. Click [Yes].


Read from PLC

4/4

100/100%

```
Parameter Read : Completed
Boot File Read : Completed
Remote Password Read : Completed
Program (MAIN) Read : Completed
Read from PLC : Completed
```

☐ When processing ends, close this window automatically.

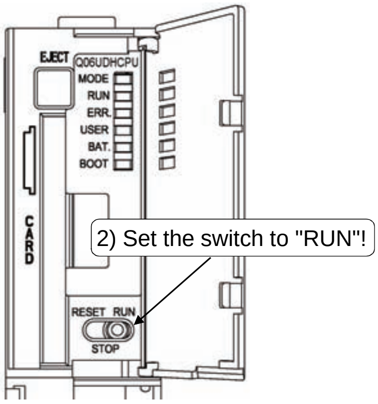
5) The progress dialog box is displayed.

6) The message "Completed" is displayed when the reading is completed. Click [Close].

2.8 Monitoring Ladder Program Status

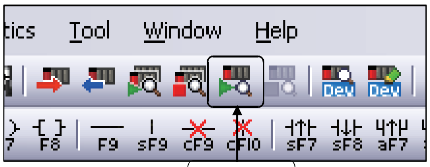
1) Suppose that the ladder program (sequence program) has been written into the programmable controller CPU to proceed to the next step.

2) Set the RUN/STOP/RESET switch on the CPU to RESET once (for about one sec.), return it to STOP, then set it to RUN.



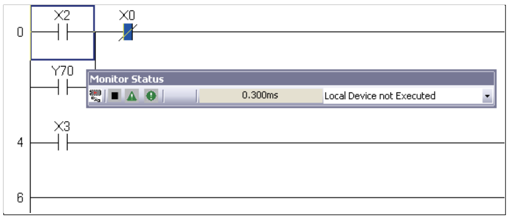
2) Set the switch to "RUN"!

3) Click on the toolbar or click [Online] → [Monitor] →[Start Monitoring].



3) Click!

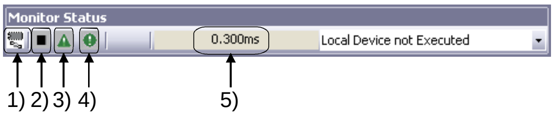
4) Selecting another menu ends the monitor mode.



Operation Practice

1) Confirm that the LED indicator Y70 lights up by turning on the snap switch X2, and that the indicator remains lit after the snap switch is turned off.
2) Confirm that the LED indicator Y70 turns off by pressing (turning on) the push button (snap switch) X0, and that the indicator does not light up when the button (snap switch) is released (turned off).
3) Turning on the snap switch X3 turns on the LED indicator Y71.

(1) In the monitor mode, the Monitor Status dialog box shown below is displayed regardless of the monitor status.



1) Connection status
Displays the connection status between a programmable controller CPU and personal computer in which the simulation function is started.

2) RUN/STOP status
Displays the programmable controller CPU status operated by the key switch on the programmable controller CPU or the remote operation from GX Works2.

3) ERR. status (PLC diagnostics)
Displays the error status of the programmable controller CPU.
Clicking the icon displays the PLC Diagnostics screen (*1).

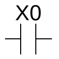
4) USER status (PLC diagnostics)
Displays the user error status of the programmable controller CPU.
Clicking the icon displays the PLC Diagnostics screen (*1).

5) Scan time
Displays the maximum scan time of the monitored programmable controller CPU.
The Q-series programmable controller displays the scan time in units of 0.1msec.

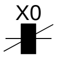
*1: For the PLC diagnostics, refer to section 2.8.

(2) The statuses of the ladder are indicated as shown below.

1) Display of a contact when X0 = OFF

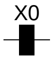


Normally open contact (not conducting)

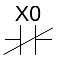


Normally closed contact (conducting)

Display of a contact when X0 = ON



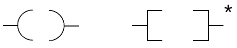
Normally open contact (conducting)



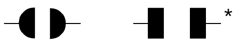
Normally closed contact (not conducting)

2) Display of a coil output instruction, contact-equivalent comparison instruction, and coil-equivalent instruction

Not executed,
conditions not established



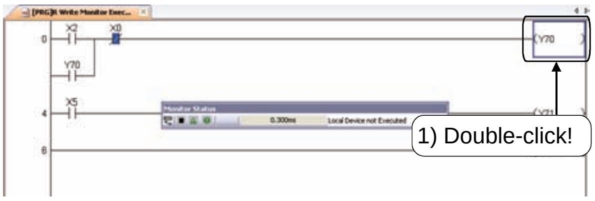
Executed,
conditions established



*: Available contact-equivalent comparison and coil-equivalent instructions are SET, RST, PLS, PLF, SFT, SFTP, MC, FF, DELTA, and DELTAP.

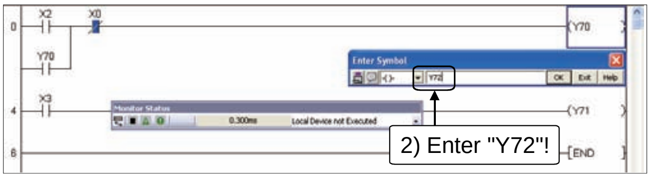
(3) Ladder conversion during the monitoring

This section explains the procedure to convert Y70 into Y72 during the monitoring.



1) Double-click!

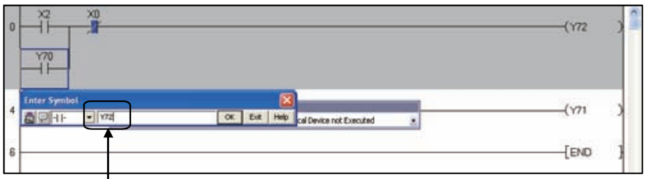
1) Double-click (−(Y70)−).



2) Enter "Y72"!

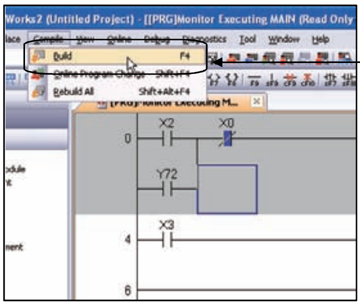
2) The Enter Symbol window is displayed. Enter "Y72".

3) Press the Enter key.



4) Enter "Y72"!

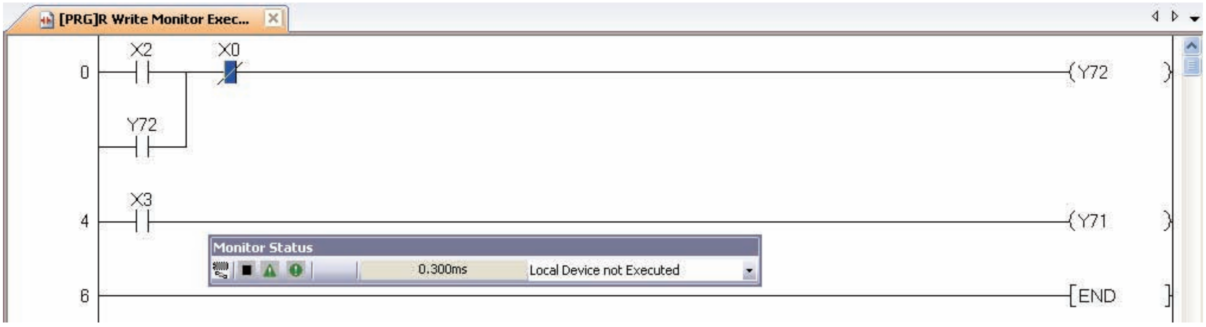
4) Double-click ($\overset{Y70}{\dashv\vdash}$) and change "Y70" to "Y72".

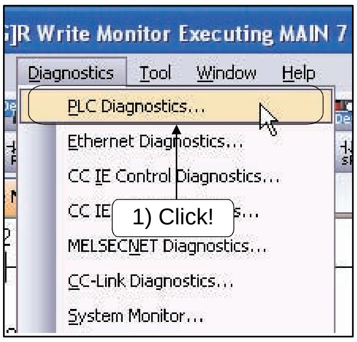

Works2 (Untitled Project) - [[PRG]Monitor Executing MAIN (Read Only
Compile
View
Online
Debug
Diagnostics
Tool
Window
Help
Build F4
Online Program Change Shift+F4
Rebuild All Shift+Alt+F4
0
X2
X0
Y72
X3
4
6

5) Click!

5) Click [Compile] → [Build] (F4).

6) The conversion is completed.

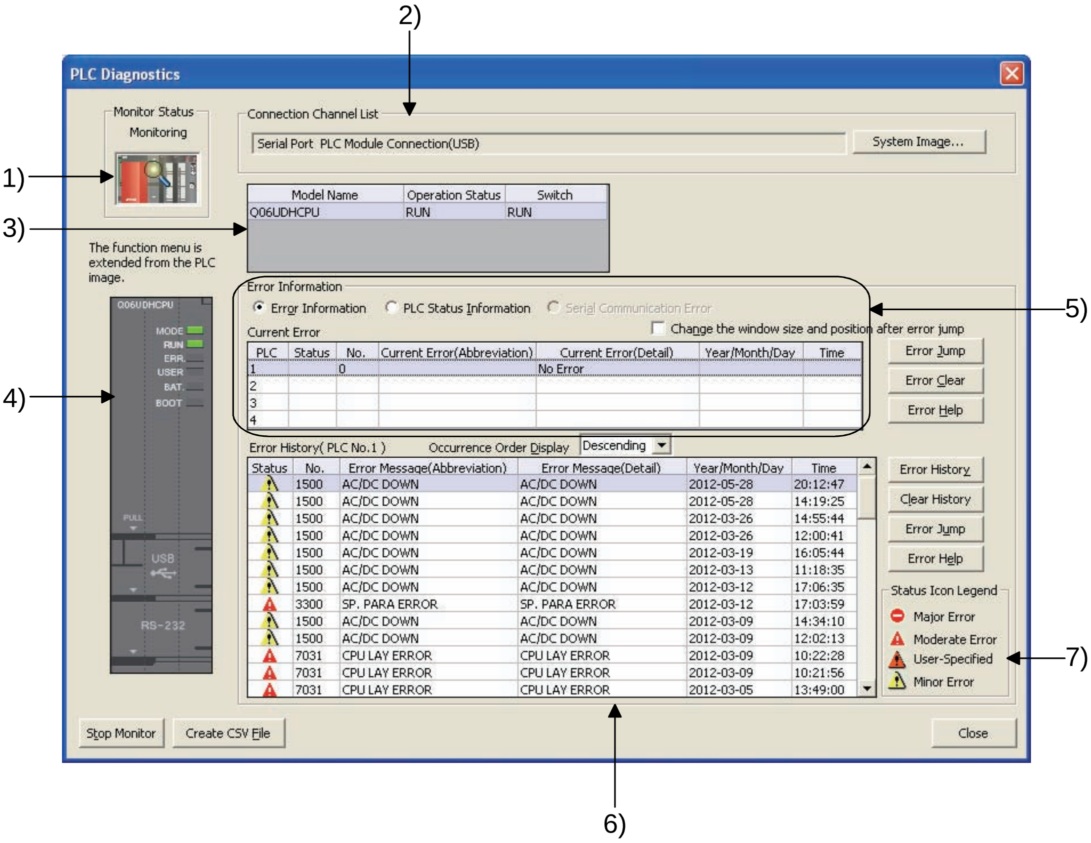




1) Click [Diagnostics] → [PLC Diagnostics].

⇩

2) The PLC Diagnostics screen is displayed.



Serial Port PLC Module Connection(USB)

Model Name	Operation Status	Switch
Q06UDHCPU	RUN	RUN

Error Information

● Error Information ○ PLC Status Information ○ Serial Communication Error

Current Error

□ Change the window size and position after error jump

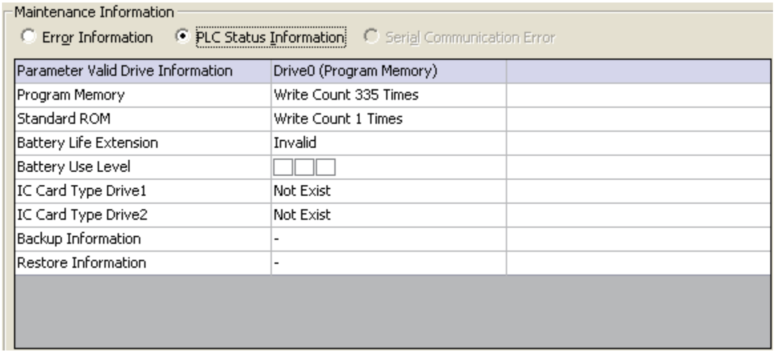
PLC	Status	No.	Current Error(Abbreviation)	Current Error(Detail)	Year/Month/Day	Time
1		0		No Error		
2						
3						
4						

Error History(PLC No.1) Occurrence Order Display Descending

Status	No.	Error Message(Abbreviation)	Error Message(Detail)	Year/Month/Day	Time
⚠	1500	AC/DC DOWN	AC/DC DOWN	2012-05-28	20:12:47
⚠	1500	AC/DC DOWN	AC/DC DOWN	2012-05-28	14:19:25
⚠	1500	AC/DC DOWN	AC/DC DOWN	2012-03-26	14:55:44
⚠	1500	AC/DC DOWN	AC/DC DOWN	2012-03-26	12:00:41
⚠	1500	AC/DC DOWN	AC/DC DOWN	2012-03-19	16:05:44
⚠	1500	AC/DC DOWN	AC/DC DOWN	2012-03-13	11:18:35
⚠	1500	AC/DC DOWN	AC/DC DOWN	2012-03-12	17:06:35
▲	3300	SP. PARA ERROR	SP. PARA ERROR	2012-03-12	17:03:59
⚠	1500	AC/DC DOWN	AC/DC DOWN	2012-03-09	14:34:10
⚠	1500	AC/DC DOWN	AC/DC DOWN	2012-03-09	12:02:13
▲	7031	CPU LAY ERROR	CPU LAY ERROR	2012-03-09	10:22:28
▲	7031	CPU LAY ERROR	CPU LAY ERROR	2012-03-09	10:21:56
▲	7031	CPU LAY ERROR	CPU LAY ERROR	2012-03-05	13:49:00

Status Icon Legend

- ⛔ Major Error
- ▲ Moderate Error
- ⚠ User-Specified
- ⚠ Minor Error

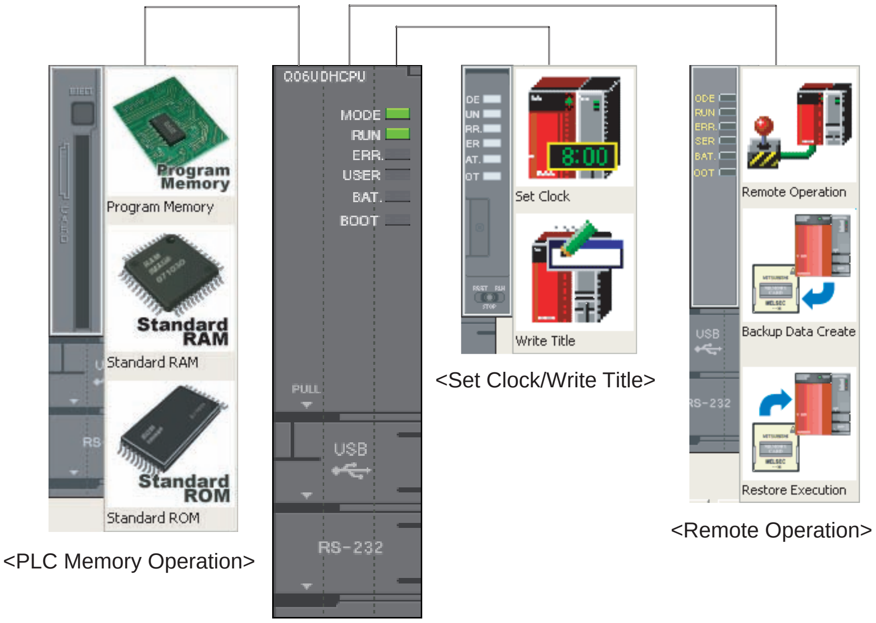
<table>
<tr><th></th><th>Item</th><th>Description</th></tr>
<tr><td>1)</td><td>Monitor Status</td><td>Displays the current monitor status.</td></tr>
<tr><td>2)</td><td>Connection Channel List</td><td>Displays the connection route which has been set.</td></tr>
<tr><td>3)</td><td>CPU operation status</td><td>● For single CPU system
Displays the operation status and switch status of the programmable controller CPU.
● For multiple CPU system
Displays the operaton status and the switch status of CPU No. 1 to No. 4.
• "Uninstallable/Blank" is displayed for a slot with no module mounted.</td></tr>
<tr><td>4)</td><td>Image of programmable controller CPU</td><td>Perform online operations of the programmable controller CPU. (refer to POINT)</td></tr>
<tr><td rowspan="2">5)</td><td>Error Information</td><td>Select this to display the current error information of the programmable controller CPU.</td></tr>
<tr><td>PLC Status Information</td><td>Select this to display the status information of the programmable controller CPU.

Maintenance Information
○ Error Information ● PLC Status Information ○ Serial Communication Error
<table>
<tr><td>Parameter Valid Drive Information</td><td>Drive0 (Program Memory)</td><td></td></tr>
<tr><td>Program Memory</td><td>Write Count 335 Times</td><td></td></tr>
<tr><td>Standard ROM</td><td>Write Count 1 Times</td><td></td></tr>
<tr><td>Battery Life Extension</td><td>Invalid</td><td></td></tr>
<tr><td>Battery Use Level</td><td>□□□</td><td></td></tr>
<tr><td>IC Card Type Drive1</td><td>Not Exist</td><td></td></tr>
<tr><td>IC Card Type Drive2</td><td>Not Exist</td><td></td></tr>
<tr><td>Backup Information</td><td>-</td><td></td></tr>
<tr><td>Restore Information</td><td>-</td><td></td></tr>
</table></td></tr>
<tr><td>6)</td><td>Error History</td><td>Displays the latest error history by clicking the [Error History] button.</td></tr>
<tr><td>7)</td><td>Status Icon Legend</td><td>Displays the status icons on the screen.</td></tr>
</table>

POINT

Online operations

The PLC Memory Operation function and the Remote Operation function can be executed from the image of the programmable controller CPU.

When the cursor is moved to the image of the programmable controller CPU, the function menu is expanded. Click the image of the programmable controller CPU to display the items to be set.

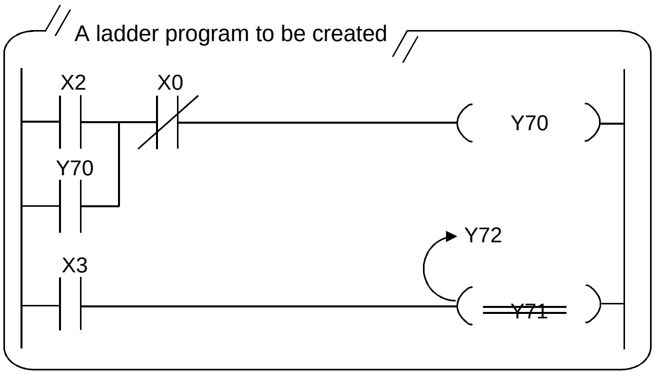


<PLC Memory Operation>

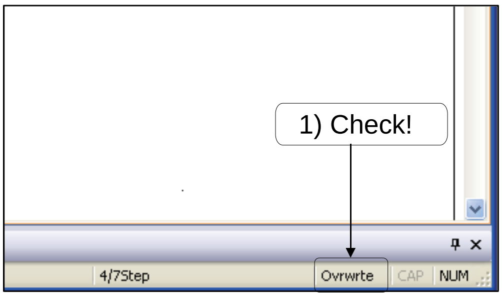
<Set Clock/Write Title>

<Remote Operation>

2.10.1 Modifying a part of the ladder program



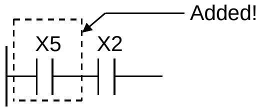
This section explains how to modify a part of the ladder program shown on the left.
(OUT␣Y71 → OUT␣Y72)



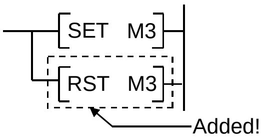
1) Confirm that "Ovrwrte" is shown at the lower-right portion of the screen.

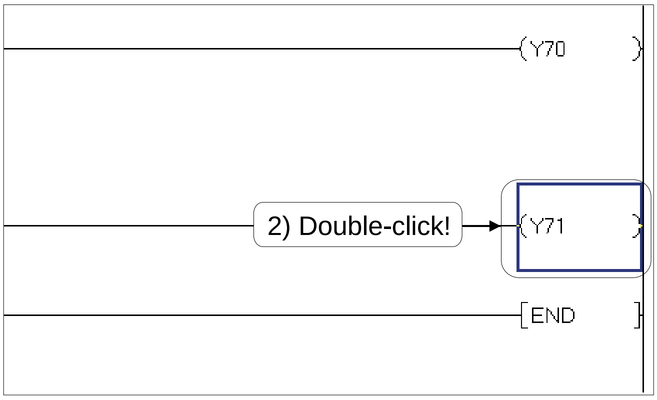
If "Insert" is shown on the screen, click the Ins key to change the display to "Ovrwrte".
If "Insert" is shown on the screen, contacts or coils are added to the diagram.

<When correcting X2 to X5>



<When correcting SET to RST>

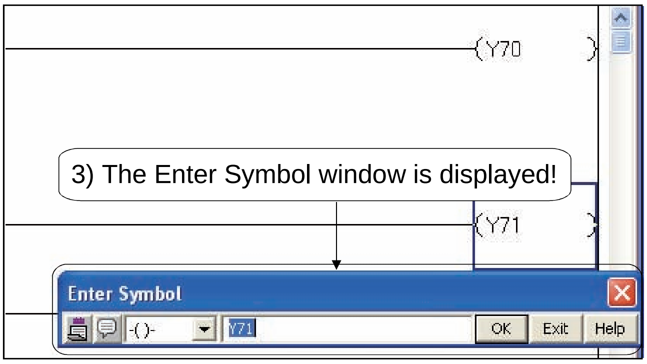




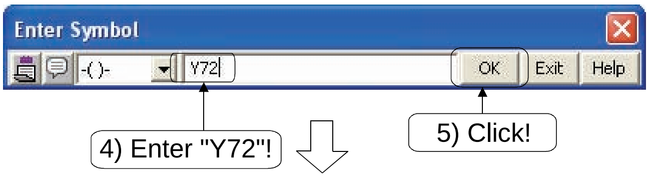
2) Double-click the point to be corrected.

(To the next page)

(From the previous page)


3) The Enter Symbol window is displayed!

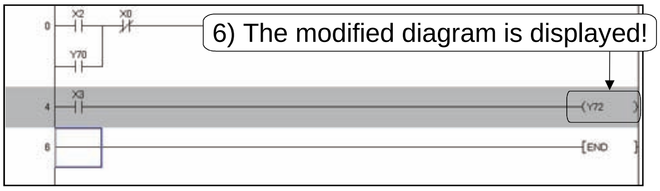
3) The Enter Symbol window is displayed.


4) Enter "Y72"!

5) Click!

4) Click the edit box and enter "Y72".

5) Click the OK button to accept the change.

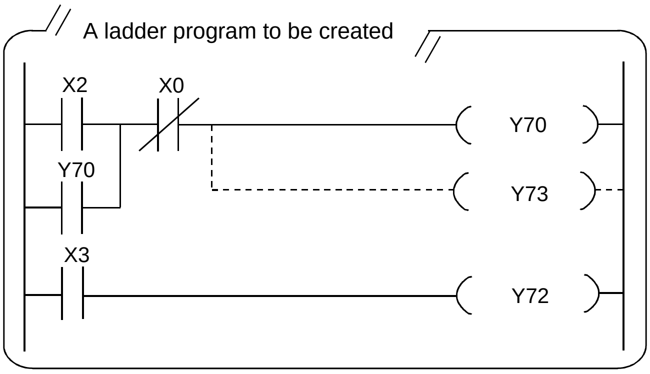

6) The modified diagram is displayed!

6) The modified ladder program is displayed.

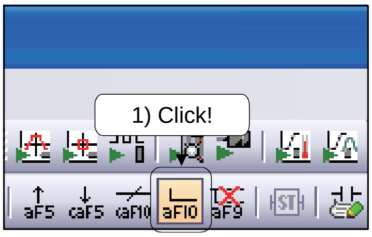
7) To convert the edited ladder program, click [Compile] → [Build] (F4).

2.10.2 Drawing/deleting lines

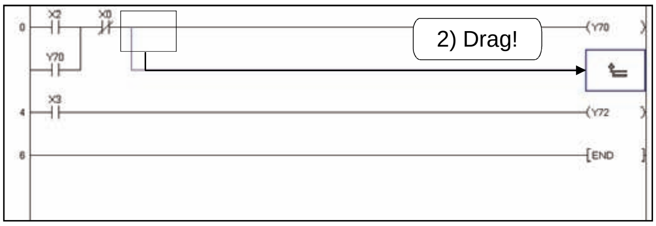
(1) Drawing lines



This section explains how to add a line to the ladder program shown on the left.

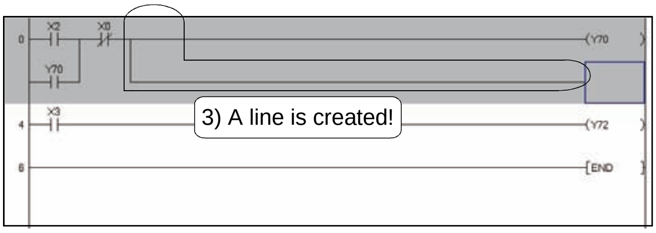


1) Click aF10 (Alt + F10) on the toolbar.



2) Drag the mouse from the start position to the end position.

A vertical line is created to the left of the cursor.

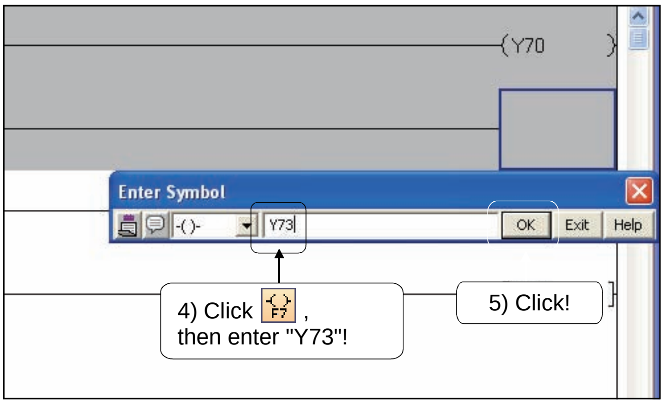


3) A line is created when the left button of the mouse is released.

(To the next page)

(From the previous page)

⇩



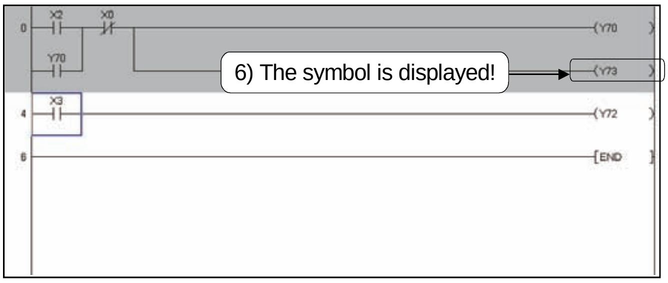
4) Click {}F7 , then enter "Y73"!

5) Click!

4) Click {}F7 on the toolbar, and enter "Y73".

5) Click the OK button.

⇩

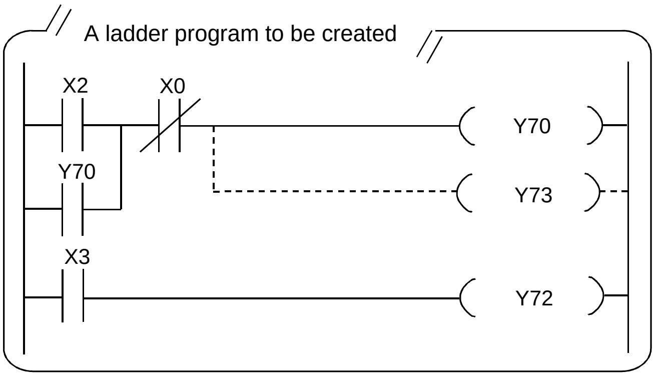


6) The symbol is displayed!

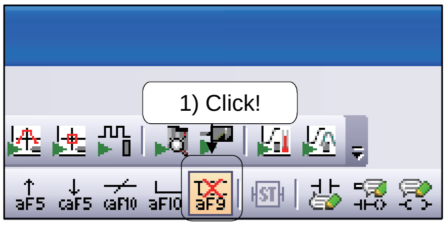
6) The entered symbol (–(Y73)–) is displayed.

7) To convert the edited ladder program, click [Compile] → [Build] (F4).

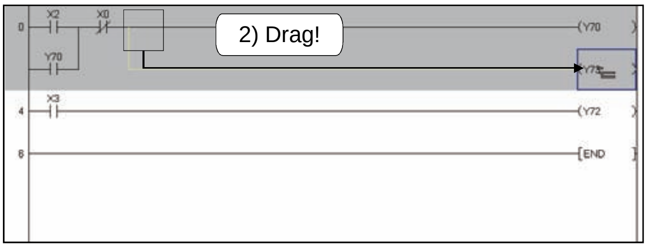
(2) Deleting lines



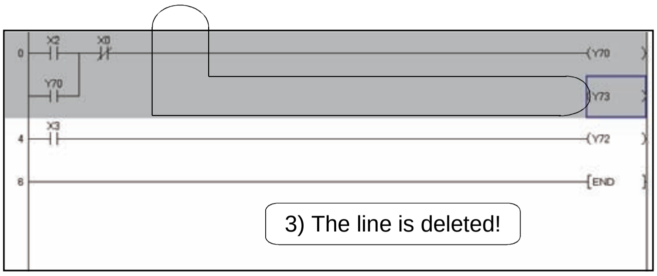
Perform the following steps to delete the line from the ladder shown on the left.



1) Click 🗙 (Alt + F9) on the toolbar.

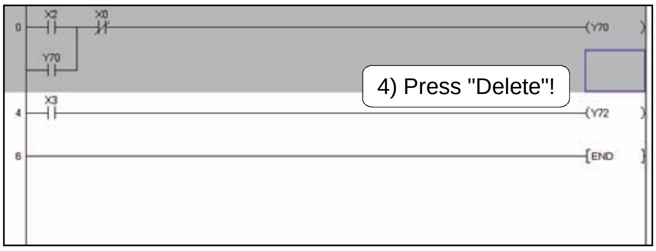


2) Drag the mouse from the start position to the end position.



3) The line is deleted when the left button of the mouse is released.

The line drawn for the END instruction cannot be removed.

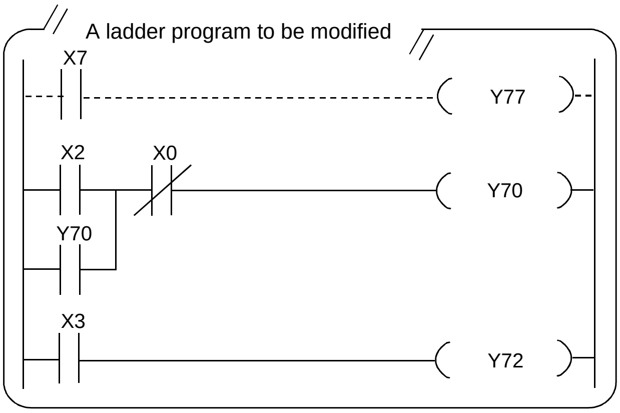


4) Press the Delete key to delete ‒(Y73)‒.

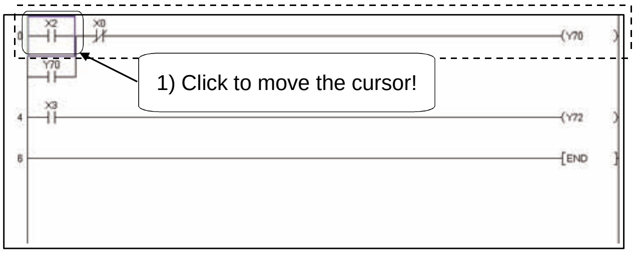
5) To convert the edited ladder program, click [Compile] → [Build] (F4).

2.10.3 Inserting/deleting rows

(1) Inserting rows



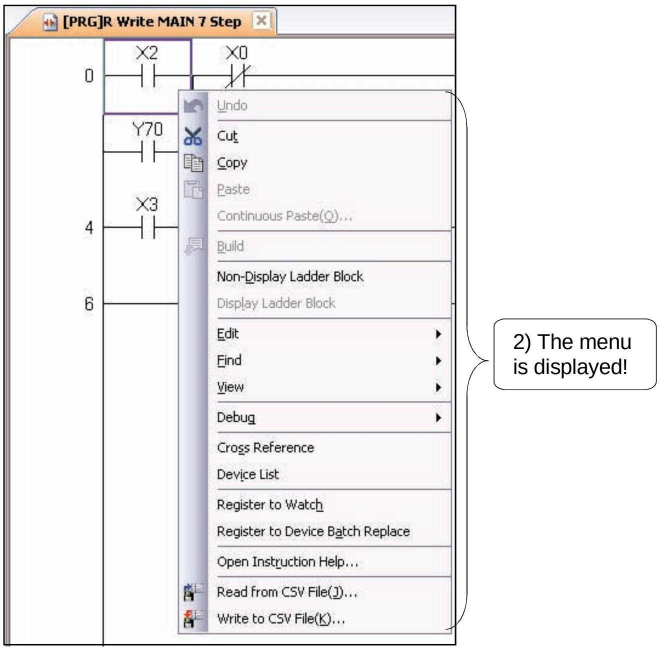
This section explains how to add a row to the ladder program shown on the left.



1) Click to move the cursor!

1) Click on any point of the row to move the cursor.

A new row is inserted above the row selected with the cursor.

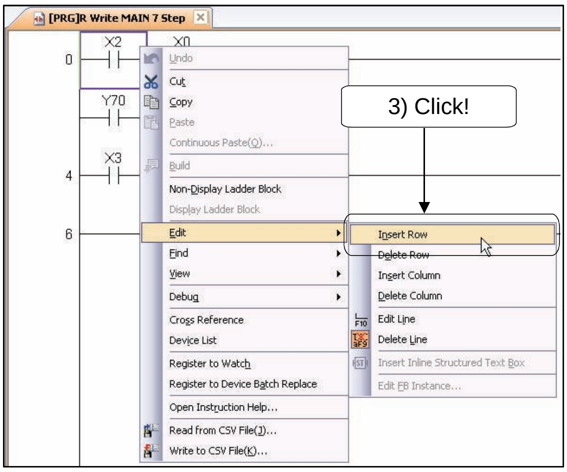


2) The menu is displayed!

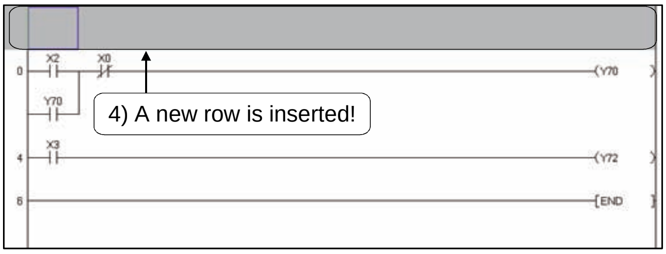
2) Right-click on any point on the ladder program creation screen to display the menu.

(To the next page)

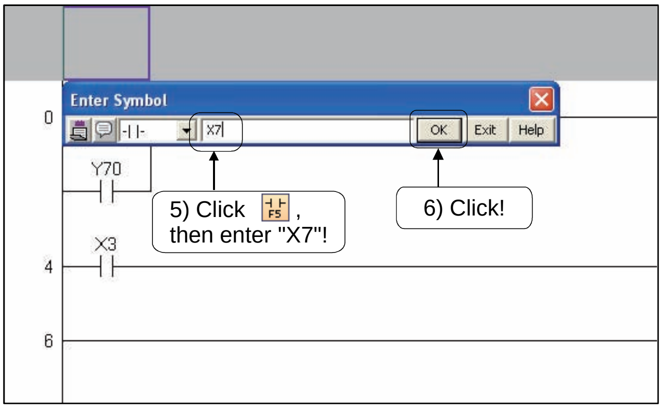
(From the previous page)



3) Select the [Edit] → [Insert Row] (Shift + Ins).



4) A new row is inserted above the selected row.

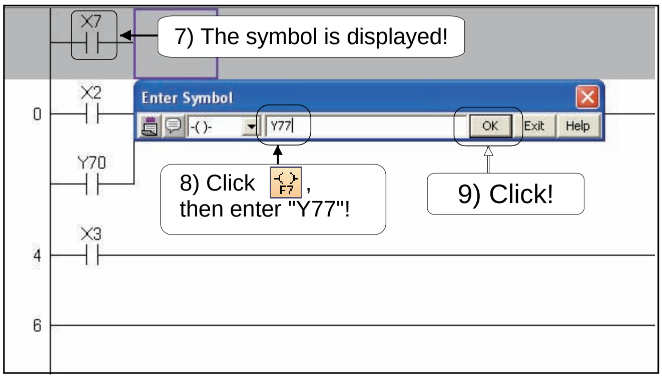


5) Click F5 on the toolbar to open the Enter Symbol window. Enter "X7".

6) Click the OK button to accept the entry.

(To the next page)

(From the previous page)



7) The symbol is displayed!

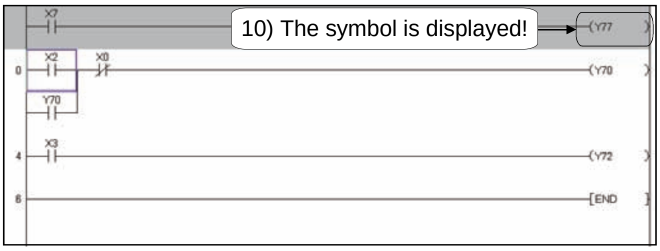
8) Click F7, then enter "Y77"!

9) Click!

7) The entered symbol ($\overset{X7}{\dashv\vdash}$) is displayed.

8) Click F7 on the toolbar, and enter "Y77".

9) Click the OK button.

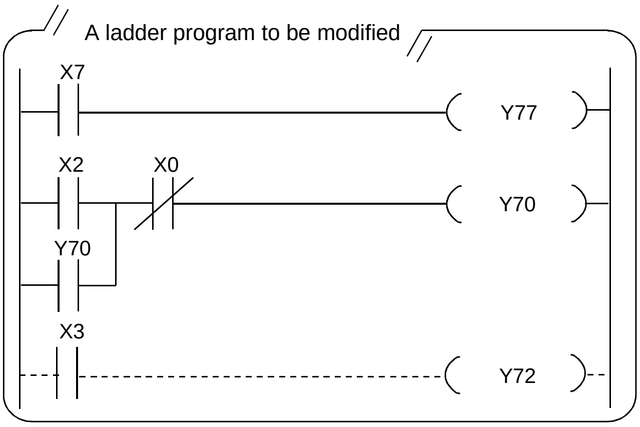


10) The symbol is displayed!

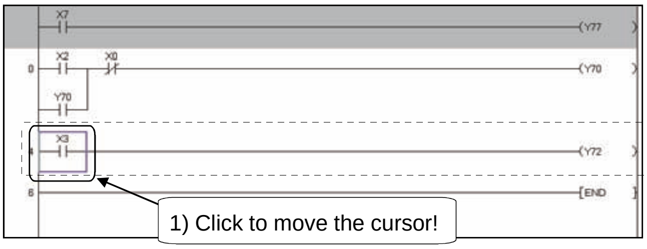
10) The entered symbol (–(Y77)–) is displayed.

11) To convert the edited ladder program, click [Compile] → [Build] (F4).

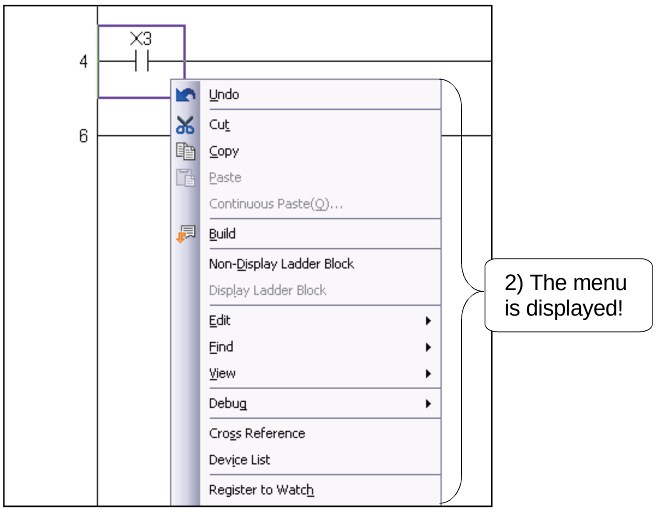
(2) Deleting rows



This section explains how to delete the row from the ladder program shown on the left.



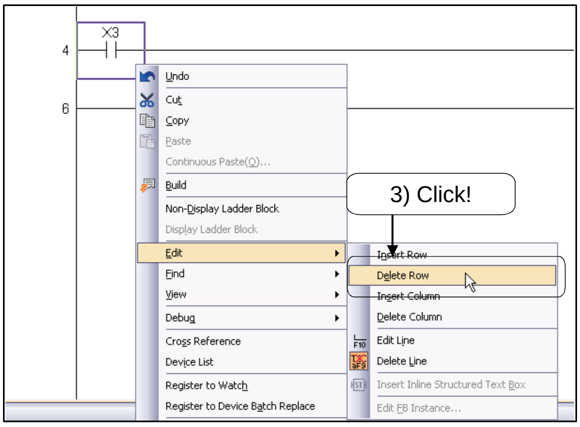
1) Click on any point of the row to be deleted to move the cursor.



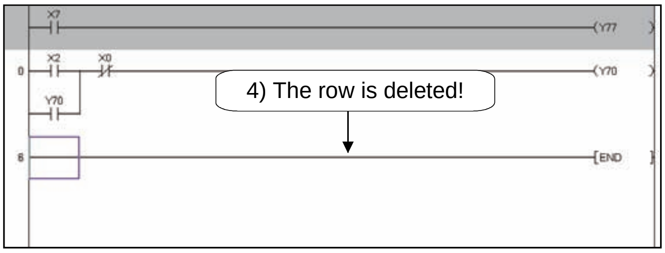
2) Right-click on any point on the ladder program creation screen to display the menu.

(To the next page)

(From the previous page)



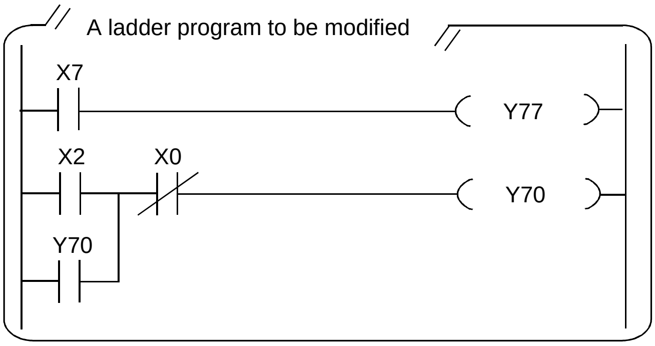
3) Select the [Edit] → [Delete Row] (Shift + Del).



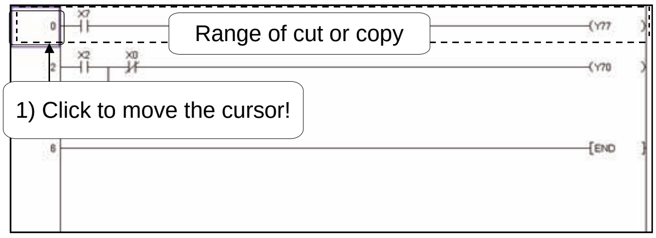
4) The selected row is deleted.

5) To convert the edited ladder program, click [Compile] → [Build] (F4).

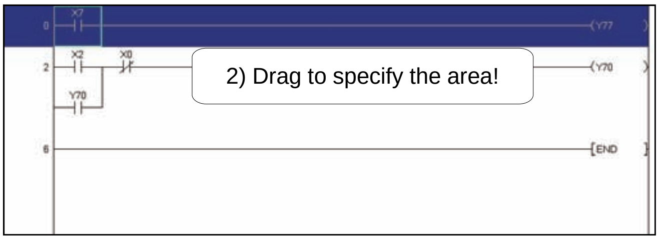
2.10.4 Cutting/copying ladder program



This section explains how to copy and cut the ladder program shown on the left.

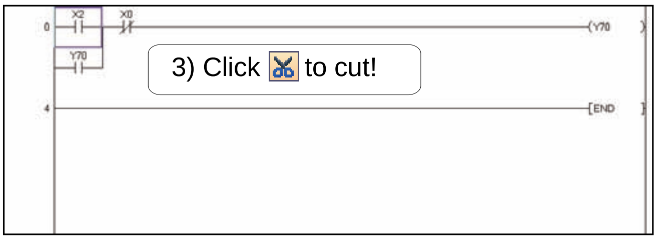


1) Click on the start point of the ladder program to be cut to move the cursor.



2) Drag the mouse over the ladder to specify the area.
   The selected area is highlighted.

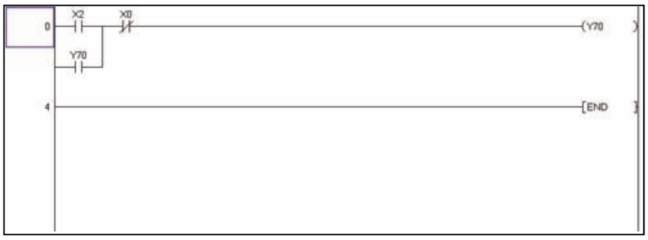
Click the step numbers and drag the mouse vertically to specify the area in ladder block units.



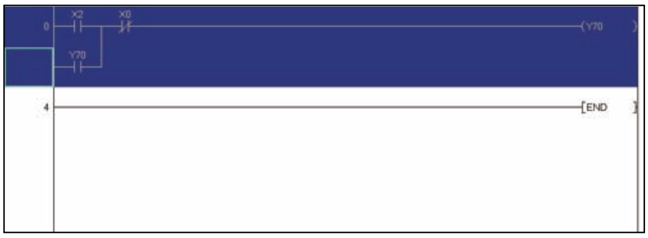
3) Click on the toolbar or select [Edit] → [Cut] (Ctrl + X) to cut the specified area.

(To the next page)

(From the previous page)

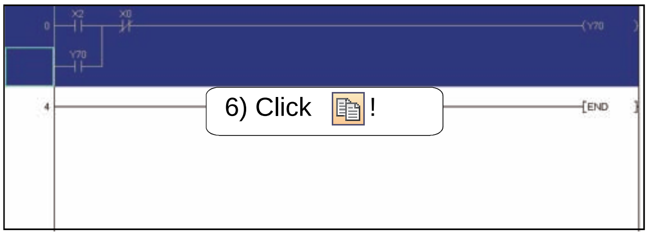


4) Click on the start point of the ladder program to be copied to move the cursor.



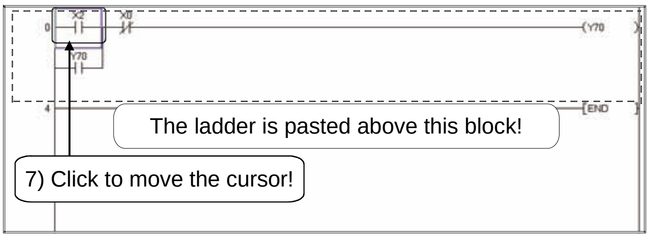
5) Drag the mouse over the ladder to specify the area.
The selected area is highlighted.

Click the step numbers and drag the mouse vertically to specify the area in ladder block units.



6) Click 📋!

6) Click 📋 on the toolbar or select [Edit] → [Copy] (Ctrl + C) to copy the specified area.



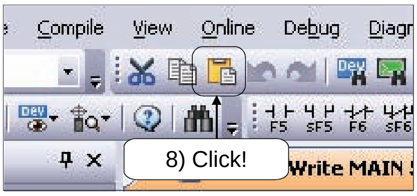
The ladder is pasted above this block!

7) Click to move the cursor!

7) Click any ladder block to move the cursor to the ladder. The copied ladder is pasted above the row with the cursor.

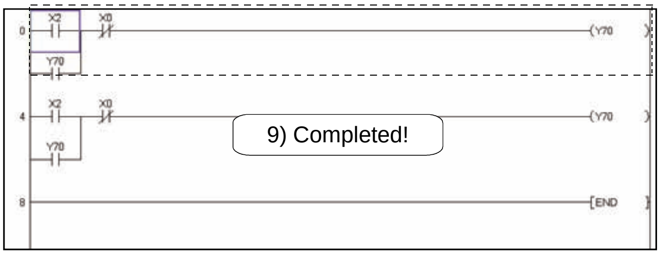
(To the next page)

(From the previous page)



8) Click!

8) Click 📋 on the toolbar or select [Edit] → [Paste] (Ctrl + V) to paste the cut or copied area.

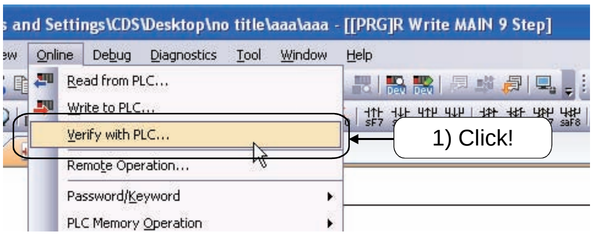


9) Completed!

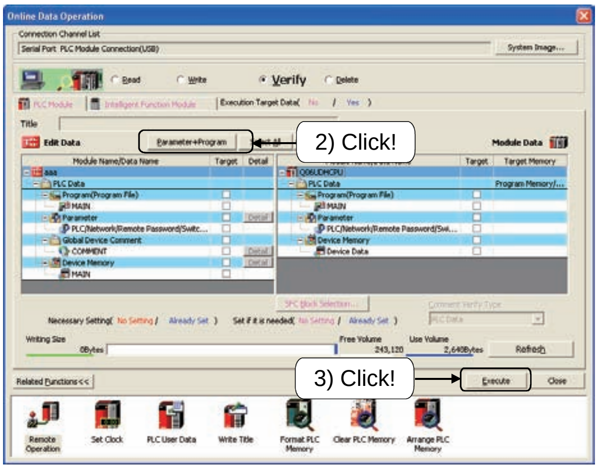
9) The cut or copied ladder is pasted.

2.11 Verifying Data

This section explains how to verify the open project against the data on the programmable controller CPU.
The verification function is used to compare the contents of two projects or to locate program changes made in the programs.

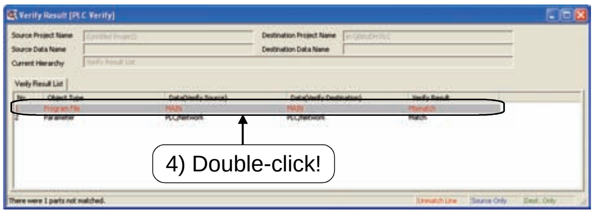


1) Click [Online] → [Verify with PLC].



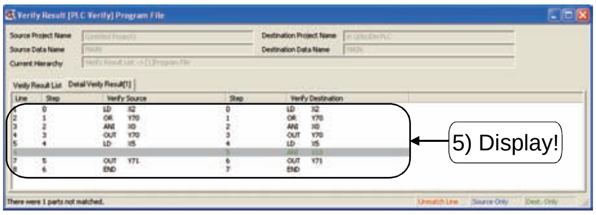
2) The Online Data Operation dialog box is displayed.
Click the Parameter + Program button.

3) Click the Execute button.



4) The verification result is displayed in the Verify Result window.

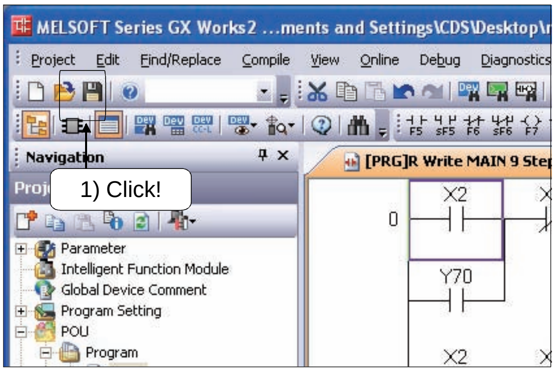
To check the detail of the data, double-click the corresponding row.


Verify Result [PLC Verify] Program File
5) Display!
There were 1 parts not matched.

5) The detail of the result is displayed.

2.12 Saving Ladder Program

2.12.1 Saving newly-created or overwritten projects

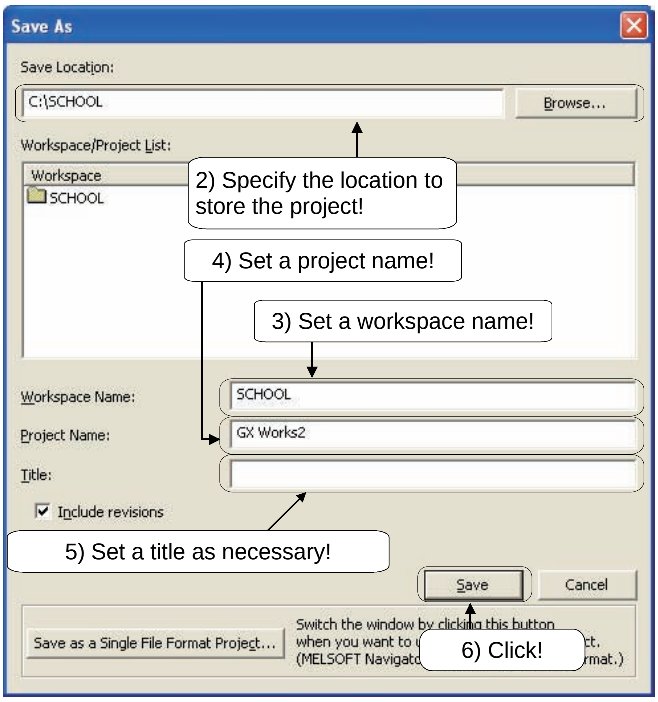

MELSOFT Series GX Works2 ...ments and Settings\CDS\Desktop\
Project Edit Find/Replace Compile View Online Debug Diagnostics
Navigation
Parameter
Intelligent Function Module
Global Device Comment
Program Setting
POU
Program
[PRG]R Write MAIN 9 Step
X2
Y70
X2

1) Click!

1) Click 💾 on the toolbar or select [Project] → [Save] (Ctrl + S).

Saving the existing project is completed at this step.

(Only when a newly-created project is saved)



Save Location: C:\SCHOOL

Workspace Name: SCHOOL

Project Name: GX Works2

Title:

☑ Include revisions

Switch the window by clicking this button when you want to ... ct. (MELSOFT Navigat... rmat.)

2) Specify the location to store the project!

4) Set a project name!

3) Set a workspace name!

5) Set a title as necessary!

6) Click!

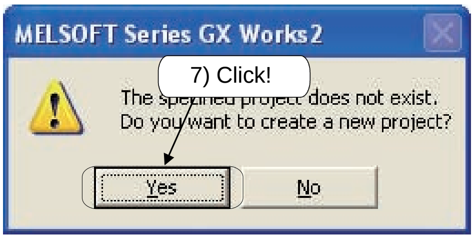
2) Specify the location to store the project.

3) Set a workspace name.

4) Set a project name.

5) Set a title as necessary.

6) Click the Save button to accept the entry.



The specified project does not exist.
Do you want to create a new project?

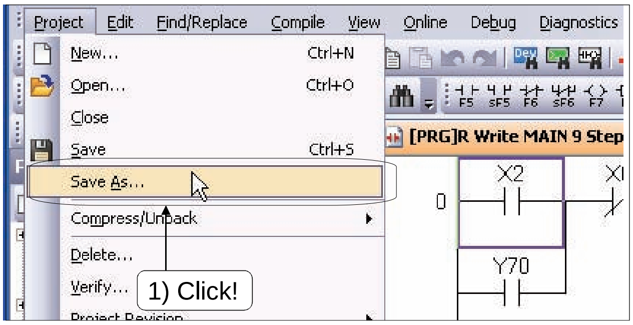
7) Click!

7) Click the Yes button.
The new project is saved.

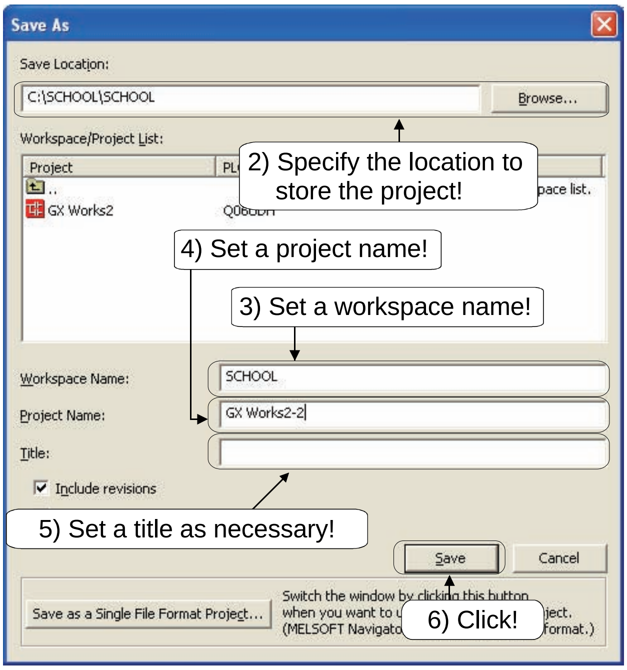
POINT

- Workspace
  Workspace enables GX Works2 to manage several projects with one name.
- When the save destination exists
  When the save destination (workspace and project) exists, the folder where the workspace is saved can be specified in "Workspace/Project List".
- Number of the characters for a workspace name, project name, and title
  Specify a workspace name, project name, and title within 128 characters each.
  However, the total number of the characters of the save destination path name + workspace name + project name must be within 150.

2.12.2 Saving a project with another name



1) Click [Project] → [Save as].



Save Location: C:\SCHOOL\SCHOOL

Workspace Name: SCHOOL

Project Name: GX Works2-2

Title:

☑ Include revisions

Switch the window by clicking this button when you want to ... (MELSOFT Navigato... ...ject. ...format.)

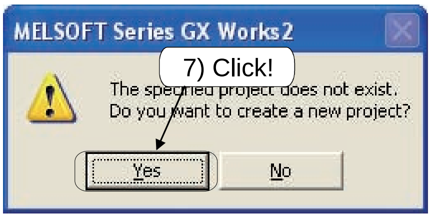
2) Specify the location to store the project.

3) Set a workspace name.

4) Set a project name.

5) Set a title as necessary.

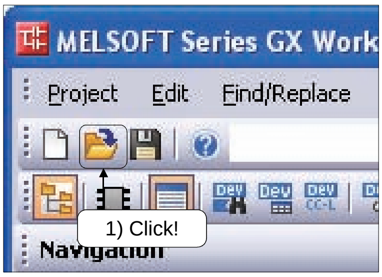
6) Click the Save button to accept the entry.



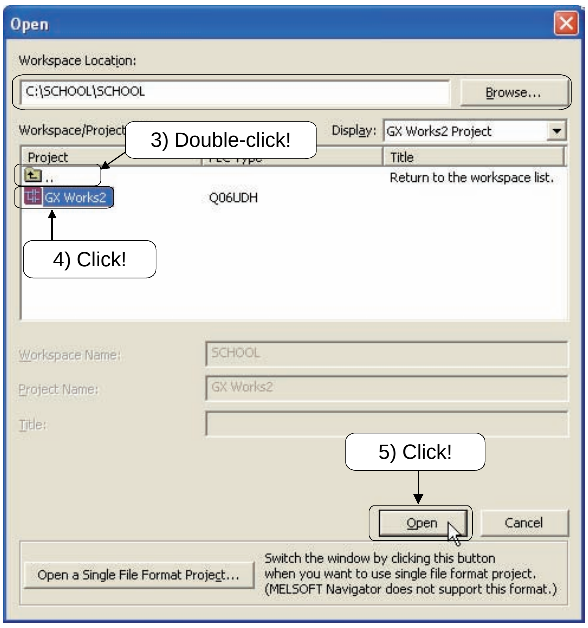
The specified project does not exist.
Do you want to create a new project?

7) Click the Yes button.
The new project is saved.

2.13 Reading the saved project



1) Click [icon] on the toolbar or select [Project] → [Open] (Ctrl + O).



Workspace Location: C:\SCHOOL\SCHOOL

Workspace/Project

Display: GX Works2 Project

Project	File Type	Title
..		Return to the workspace list.
GX Works2	Q06UDH	

Workspace Name: SCHOOL

Project Name: GX Works2

Title:

Switch the window by clicking this button when you want to use single file format project. (MELSOFT Navigator does not support this format.)

2) Specify the location where the project to be read is stored.

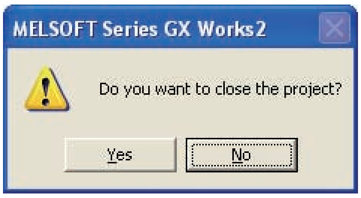
3) Double-click the workspace to be read.

4) Click the project to be read.

5) Click the button to start reading the specified project.

Each confirmation dialog box below is displayed in the following cases;

(When another project has been open)

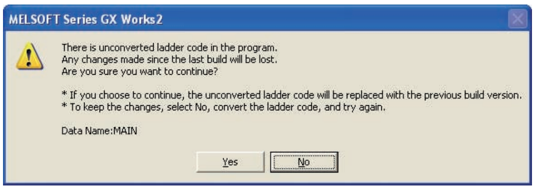


Do you want to close the project?

Yes ·········Terminates the project.

No ··········Keeps the project open.

(When another project has been open without being converted)



There is unconverted ladder code in the program.
Any changes made since the last build will be lost.
Are you sure you want to continue?

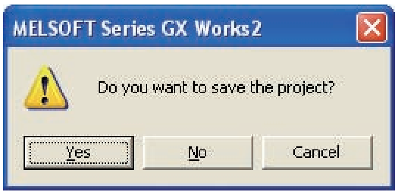
* If you choose to continue, the unconverted ladder code will be replaced with the previous build version.
* To keep the changes, select No, convert the ladder code, and try again.

Data Name:MAIN

Yes ·········Terminates the project without converting the project.

No ··········Keeps the project open. (Continue editing the ladder program.)

(When another project has been open without being saved)



Do you want to save the project?

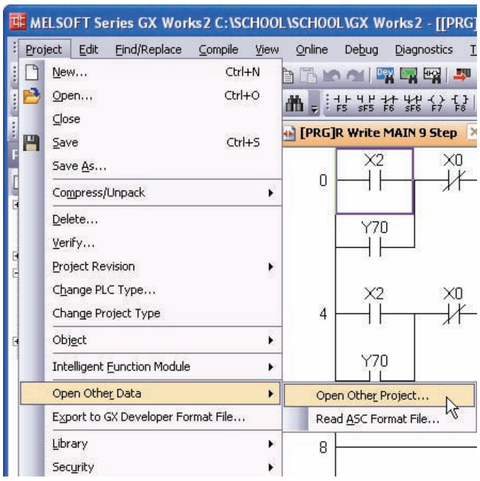
Yes ·········Terminates the project after saving it.

No ··········Terminates the project without saving it.

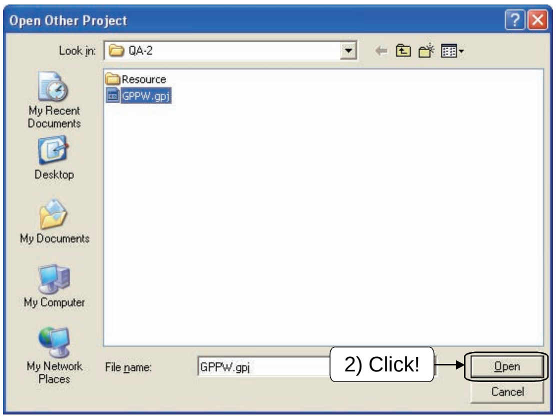
Cancel ····Keeps the project open.

2.14 Opening Projects in Different Format

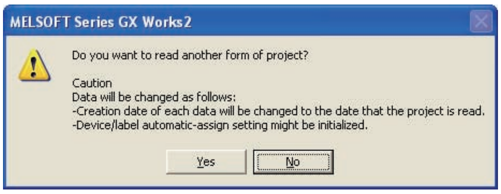
This section explains how to open a project created with GX Developer in GX Works2.



1) Click [Project] → [Open Other Data] → [Open Other Project].



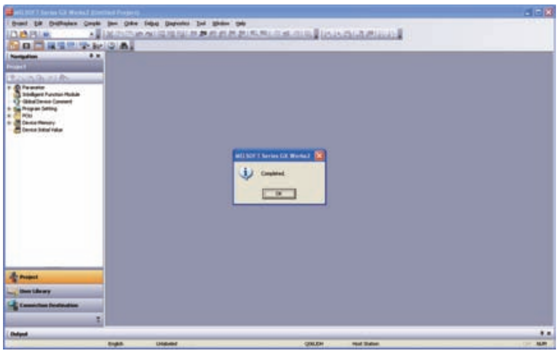
2) The Open Other Project dialog box is displayed. Specify the project and click the Open button.


MELSOFT Series GX Works2

Do you want to read another form of project?

Caution
Data will be changed as follows:
-Creation date of each data will be changed to the date that the project is read.
-Device/label automatic-assign setting might be initialized.

3) The message on the left is displayed. Click the Yes button.



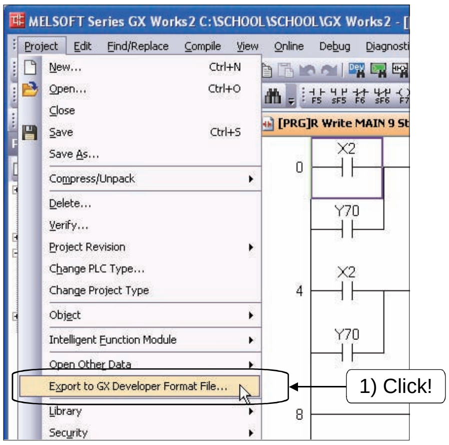
4) The project created with GX Developer is read.

POINT

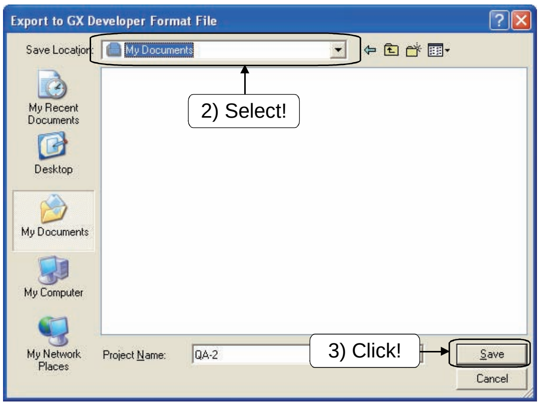
- Status after a project in a different format are opened
  When a project in a different format is opened, the project is in the uncompiled status.
  Compile all programs in the project before executing online operations such as writing data and monitoring.
  When a compile error occurs, correct the corresponding program according to the programming manual.

2.15 Saving Projects in Different Format

This section explains how to save a Simple project of GX Works2 in the GX Developer format.

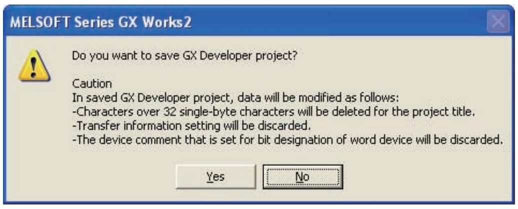


1) Click [Project] → [Export to GX Developer Format File].



2) The Export to GX Developer Format File dialog box is displayed. Specify the destination to save the project.

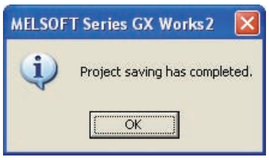
3) Enter a project name and click the Save button.


MELSOFT Series GX Works2

Do you want to save GX Developer project?

Caution
In saved GX Developer project, data will be modified as follows:
-Characters over 32 single-byte characters will be deleted for the project title.
-Transfer information setting will be discarded.
-The device comment that is set for bit designation of word device will be discarded.

4) The message on the left is displayed. Click the Yes button.


MELSOFT Series GX Works2

Project saving has completed.

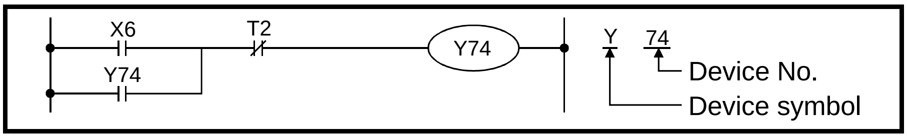
5) The project is saved in the GX Developer format.

MEMO

CHAPTER 3 DEVICE AND PARAMETER OF PROGRAMMABLE CONTROLLER

3.1 Device

A device is an imaginary element for programming in the programmable controller CPU, as well as the components (such as contacts and coils) that compose a program.



Type		Description	Remark
X	Input	Sends commands and data to a programmable controller through external devices such as push buttons, selector switches, limit switches, and digital switches.	• Bit device • Mainly handles on/off signals.
Y	Output	Outputs control results to solenoids, electromagnetic switches, signal lights, and digital indicators.	
M	Internal relay	Auxiliary relay inside a programmable controller that cannot output directly to external devices	
L	Latch relay	Uninterruptible auxiliary relay inside the programmable controller that cannot output directly to external devices	
B	Link relay	Internal relay for data link that cannot output directly to external devices The area not assigned by initial link information setting can be used as an internal relay.	
F	Annunciator	Used for failure detection. Create a failure detection program beforehand and turn on the program while the programmable controller is running to store numerical values in the special register D.	
V	Edge relay	Internal relay that stores an operation result (on/off information) from the top of a circuit block	
SM	Special relay	Internal relay that stores CPU statuses	
SB	Special link relay	Internal relay for data link that indicates a communication status and errors	
FX	Function input	Internal relay that captures the on/off data specified by a subroutine call instructions with arguments in a subroutine program	
FY	Function output	Internal relay that passes an operation result (on/off data) in a subroutine program to a subroutine program call source	
T(ST)	Timer	Accumulative timers of four types: low-speed timer, high-speed timer, low-speed integrator, and high-speed integrator	• Word device • Mainly handles data. • One word consists of 16 bits. • Can be specified by entering ~.* (* = 0 to F (hexadecimal)).
C	Counter	Accumulative counters of two types: counters for sequence programs and counters for interruption sequence programs	
D	Data register	Memory that stores data in the programmable controller	
W	Link register	Data register for data link	
R	File register	Register for an extensive use of data registers, which uses the standard RAM or memory card	
SD	Special register	Register that stores CPU statuses	
SW	Link data register	Data register for data link that stores a communication status and failure information	
FD	Function register	Register for the exchange data between a subroutine call source and a subroutine program	
Z	Index register	Register for modification to the devices (X, Y, M, L, B, F, T, C, D, W, R, K, H, and P)	

Type		Description	Remark
FD	Function register	Register for the exchange data between a subroutine call source and a subroutine program	
Z	Index register	Register for modification to the devices (X, Y, M, L, B, F, T, C, D, W, R, K, H, and P)	
N	Nesting	Shows the nesting (nested structure) of the master control.	
P	Pointer	Locates the jump addresses of the branch instructions (CJ, SCJ, CALL and JMP).	
I	Interruption pointer	Locates a jump address that corresponds to the factor of the interruption when an interruption occurs.	
J	Network No. specification device	Used to specify the network number in the data link instructions.	
U	I/O No. specification device	Used to specify the I/O number in the intelligent function module dedicated instructions.	
K	Decimal constant	Used to specify the following; timer counter set value, pointer number, interruption pointer number, number of digits of bit device, and basic/application instruction values.	
H	Hexadecimal constant	Used to specify the basic/application instruction values.	
E	Real number constant	Used to specify real numbers as instructions.	
"Character String"	Character string constant	Used to specify character strings as instructions.	
Jn\X Jn\Y Jn\B Jn\SB	Link direct device	Device that can access directly to a link device of a network module (The refresh parameter setting is not required.)	• Bit device • Mainly handles on/off signals.
Jn\W Jn\SW			• Word device • Mainly handles data. • One word consists of 16 bits.
Un\G	Intelligent function module device	Device that can access directly to the buffer memory of a intelligent function module	

3.2 Parameter

The parameters are basic setting values applied to a programmable controller in order to control objects as planned.
The parameters are divided into the PLC parameter, network parameter, and remote password as shown below.

* A shaded area in the following table indicates the items to be set in this textbook.

Item			Description
PLC parameter	PLC name	Label	Sets a label (name and application) of a programmable controller CPU.
		Comment	Sets a comment for the label of a programmable controller CPU.
	PLC system	Timer limit setting	Sets the time limit of the low-speed or high-speed timer.
		RUN-PAUSE contacts	Sets contacts for controlling RUN and PAUSE of a programmable controller CPU.
		Latch data backup operation valid contact	Sets contact devices in order to execute the latch data backup operation. (Only for Universal model QCPU)
		Remote reset	Sets whether to allow a remote reset operation from GX Works2.
		Output mode at STOP to RUN	Sets the status of an output (Y) when the programmable controller is switched from STOP to RUN.
		Floating point arithmetic processing	Sets whether to execute floating-point processing in double precision. (Only for high performance model QCPU)
		Intelligent function module setting	• Sets the interruption pointer assignment of a module. • Sets the start I/O number and start SI number.
		Common pointer No.	Sets the start number of the pointer used as a common pointer.
		Points occupied by empty slot	Sets the number of empty slots for the main or extension base unit.
		System interrupt settings	• Sets the start number of the interrupt counters. • Sets the execution interval for the interrupt pointers.
		Interrupt program/fixed scan program setting	Set whether to execute high-speed execution of an interrupt program.
		Module synchronization	Set whether to synchronize the start-up of the programmable controller CPU with that of the intelligent function module.
		A-PLC compatibility setting	Set whether to use the MELSEC-A series special relays/special registers.
		Service processing setting	Sets the processing time and the number of times of service processing. (Only for Universal model QCPU)
		PLC module change setting	Set this parameter to replace the CPU module using a memory card (Only for Universal model QCPU).
	PLC file	File register	• Sets the file register file to be used in a program. • Sets whether to transfer data to the standard ROM at a latch data backup operation. (Only for Universal model QCPU)
		Comment file used in a command	Sets the device comment file to be used in a program.
		Initial device value	Sets the device initial value file to be used on the programmable controller CPU.
		File for local device	Sets the local device file to be used in a program.
		File used for SP.DEVST/S.DEVLD instruction	Sets the device data ROM write/read instruction file to be used in a program. (Only for Universal model QCPU)
	PLC RAS	WDT (watchdog timer) setting	Sets the WDT of the programmable controller CPU.
		Error check	Sets whether to detect specified errors.
		Operation mode when there is an error	Sets the programmable controller CPU operation mode when an error is detected.
		Constant scanning	Sets the constant scan time.
		Breakdown history	Sets the storage destination for error histories of the programmable controller CPU. (Only for high performance model QCPU)
		Low speed program execution time	Sets the execution time of a low-speed program in every scan. (Only for high performance model QCPU)

Item			Description
PLC parameter	Boot file	Boot option	Sets whether to clear the program memory when booting up.
		Boot file setting	Sets the type, data name, transfer source drive, and transfer destination drive of the boot file.
	Program		Sets the file name and execution type (execution condition) of the program when several programs are written to the programmable controller CPU
	SFC		Sets the startup mode and startup condition of an SFC program and the output mode at block stop
	Device	Device points	Sets the number of points used for each device of the programmable controller CPU.
		Latch (1) start/end	Sets the latch range (start device number/end device number) clearable with the RESET/L.CLR switch or a remote latch clear operation.
		Latch (2) start/east	Sets the latch range (start device number/end device number) not clearable with the RESET/L.CLR switch or a remote latch clear operation.
		Local device start/end	Sets the range (start device number/end device number) of devices used as a local device.
		File register extended setting	Sets the extended data register and extended link register. (Only for Universal model QCPU)
		Indexing setting for ZR device	Sets the start number of Z to be 32-bit indexed, or use the index register ZZ for 32-bit index setting. (Only for Universal model QCPU)
	I/O assignment	I/O assignment	Sets the type, model, number of occupied I/O points, and start I/O number of each module mounted on the base unit.
		Basic setting	Sets the model and the number of slots of the base unit, the model of the power supply module, and the model of the extension cable.
	Multiple CPU setting	No. of PLC	Sets the number of programmable controller CPUs used in the multiple CPU system.
		Operation mode	Sets the operation mode of the multiple CPU system when a stop error occurs in any of the programmable controller CPU No. 2 to No. 4.
		Host station	Sets the CPU number for the host CPU.
		Multiple CPU synchronous startup setting	Selects the CPU modules to be started up synchronously.
		Online module change	Sets whether to allow the online module change in the multiple CPU system.
		I/O sharing when using multiple CPUs	Sets whether to retrieve the I/O status of the I/O module or intelligent function module controlled by other programmable controller CPUs.
		Communication area setting (refresh setting)	Sets the CPU shared memory to enable data sharing among multiple CPUs.
		Multiple CPU high speed transmission area setting	Sets the user setting area, auto refresh, assignment confirmation, and system area.
	Built-in Ethernet port setting	IP address setting	Sets the IP address and the input format of the IP address.
		Communication data code	Selects the Binary code or ASCII code for communication.
		Open setting button	Sets the protocol, open system, and host station port number.
		FTP setting button	Selects whether to use the FTP function
		Time setting button	Sets whether to use the SNTP function and the timing of setting the time.
	Serial communication	Transmission speed	Sets the transmission speed.
		Sum check	Sets the sum check.
		Transmission wait time	Sets the transmission wait time.
		Online change	Sets whether to allow the online program change.
Network parameter	Ehternet/CC IE/MELSECNET		Sets the network parameters for Ehternet, MELSECNET/10, MELSECNET/H, and CC-Link IE controller network.
	CC-Link		Sets the parameters for CC-Link.
Remote password			Sets the password that limits the access via the Ethernet or serial communication modules.

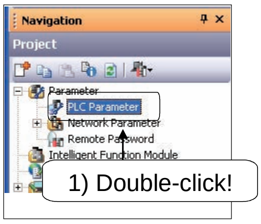
- When GX Works2 starts, it employs the preset values as the parameters. These values are called the default (initial values).
- The programmable controller can run with those values unchanged, however, modify them within a specified range as necessary.

Operation example: Changing the operation mode when an error exists

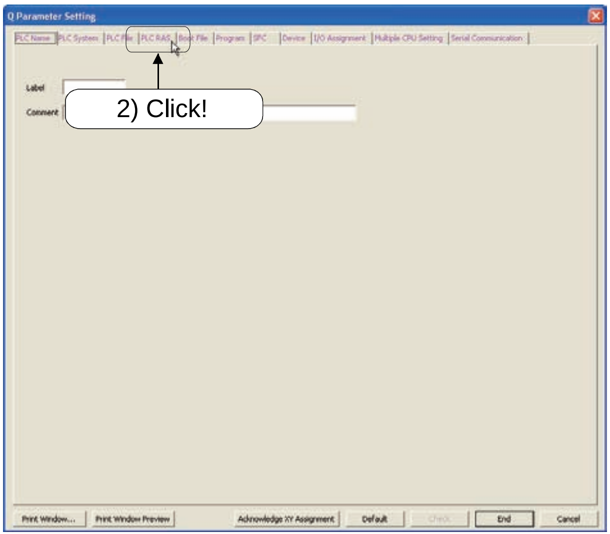
When a computation error is caused, the programmable controller CPU changes to the STOP status at the default value, however, changing the parameters continues the programmable controller CPU to run.

Computation error example

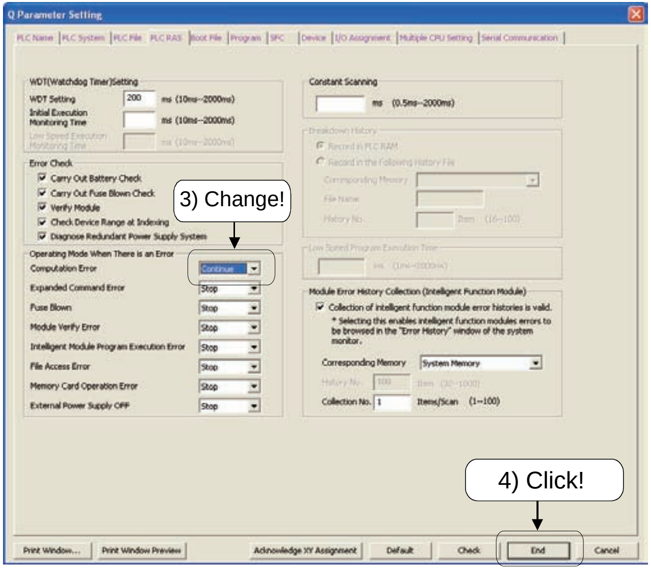
- In the division instruction, the processing to divide by 0 is executed.



1) Double-click "PLC parameter" on the navigation window.



2) The Q Parameter Setting dialog box is displayed. Click the "PLC RAS" tab.



WDT(Watchdog Timer)Setting
WDT Setting 200 ms (10ms--2000ms)
Initial Execution Monitoring Time ms (10ms--2000ms)
Low Speed Execution Monitoring Time ms (10ms--2000ms)

Error Check
☑ Carry Out Battery Check
☑ Carry Out Fuse Blown Check
☑ Verify Module
☑ Check Device Range at Indexing
☑ Diagnose Redundant Power Supply System

Operating Mode When There is an Error

Computation Error	Continue
Expanded Command Error	Stop
Fuse Blown	Stop
Module Verify Error	Stop
Intelligent Module Program Execution Error	Stop
File Access Error	Stop
Memory Card Operation Error	Stop
External Power Supply OFF	Stop

Constant Scanning
ms (0.5ms--2000ms)

Breakdown History
● Record in PLC RAM
○ Record in the Following History File
Corresponding Memory
File Name
History No. Item (16--100)

Low Speed Program Execution Time
ms (1ms--2000ms)

Module Error History Collection (Intelligent Function Module)
☑ Collection of intelligent function module error histories is valid.
* Selecting this enables intelligent function modules errors to be browsed in the "Error History" window of the system monitor.
Corresponding Memory System Memory
History No. 100 Item (32--1000)
Collection No. 1 Items/Scan (1--100)

3) Change the setting of "Computation Error" in "Operation Mode When There Is an Error" to "Continue"

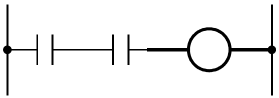
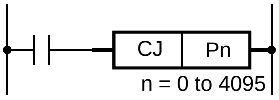
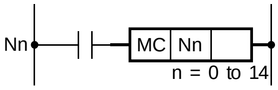
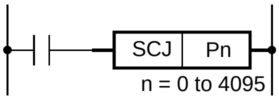
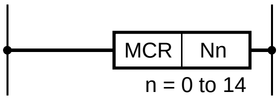
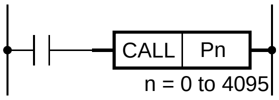
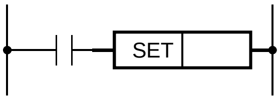
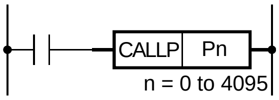
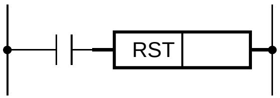
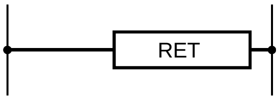
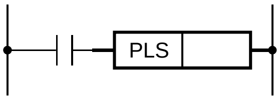
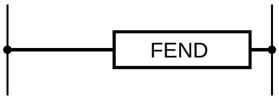
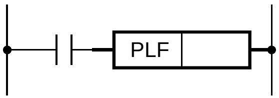
4) Click the End button.

MEMO

CHAPTER 4 SEQUENCE AND BASIC INSTRUCTIONS -PART 1-

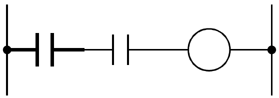
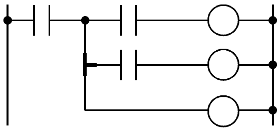
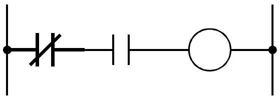
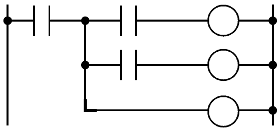
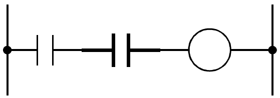
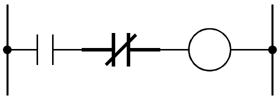
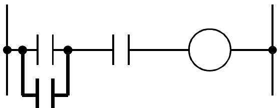
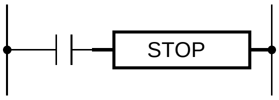
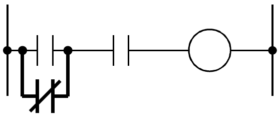
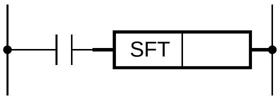
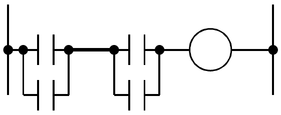
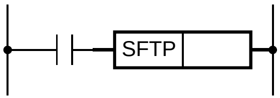
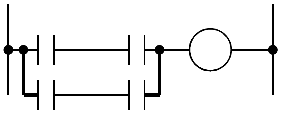
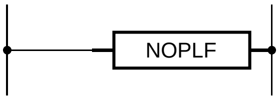
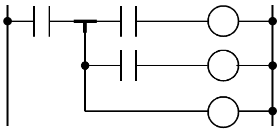
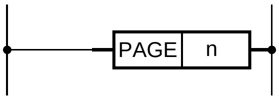
4.1 List of Instruction Explained in this Chapter

This chapter explains the sequence instructions and basic instructions as shown below.

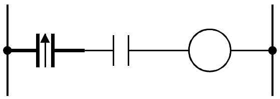
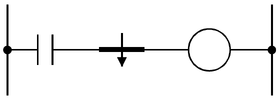
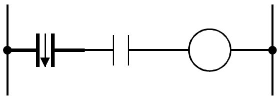
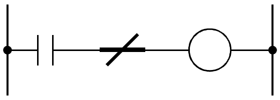
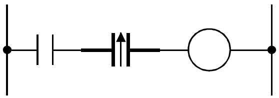
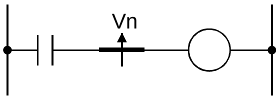
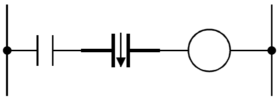
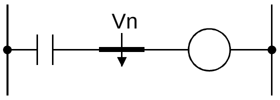
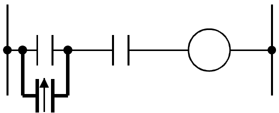
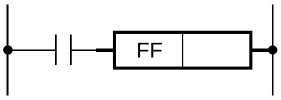
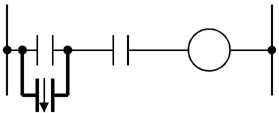
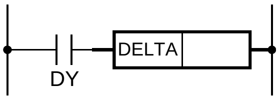
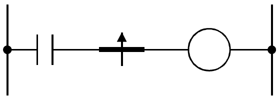
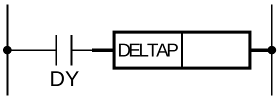
Instruction symbol (Name)	Function	Drawing (devices to be used)	Instruction symbol (Name)	Function	Drawing (devices to be used)
OUT Out	Coil output	 Specifies a bit of a bit device or word device.	CJ	Conditional jump (non-delay)	 n = 0 to 4095 Pointer
MC Master control	Starting master control[*1]	Specifies a bit of a bit device or word device.  n = 0 to 14 Nesting	SCJ	Conditional jump (Jumps after one scan)	 n = 0 to 4095 Pointer
MCR Master control reset	Terminating master control	 n = 0 to 14 Nesting	CALL	Calling subroutine program	 n = 0 to 4095 Pointer
SET Set	Setting devices	 Specifies a bit of a bit device or word device.	CALLP	Calling a subroutine program (pulsing operation)	 n = 0 to 4095 Pointer
RST Reset	Resetting devices	 Specifies a bit of a bit device or word device.	RET Return	Returning from a subroutine program	
PLS Pulse	Pulse (Generating the pulses for one program cycle when a input signal turns off)	 Specifies a bit of a bit device or word device.	FEND	Terminating a main routine program	
PLF Pulf	Pulf (Generating the pulses for one program cycle when a input signal turns off)	 Specifies a bit of a bit device or word device.			

*1: In GX Works2, the on/off status of the master control is displayed in the title tag on the monitor screen.

<List of instructions not explained in this chapter: part 1>
"Introduction: PLC Course" covers the instructions shown below. The conventional A series also support them.
Refer to "MELSEC-Q/L Programming Manual Common Instruction" for more details.

Instruction symbol (Name)	Function	Drawing (devices to be used)	Instruction symbol (Name)	Function	Drawing (devices to be used)
LD Load	Starting a logical operation (Starting to operate a normally open contact)	 Specifies a bit of a bit device or word device.	MRD Lead	Intermediate branching	
LDI Load inverse	Starting a logical inverse operation (Starting to operate a normally closed contact)	 Specifies a bit of a bit device or word device.	MPP Pop	Terminating branching	
AND And	Logical AND operation (Series connection of normally open contacts)	 Specifies a bit of a bit device or word device.	NOP Nop	Ignored	For a space or deleting a program
ANI And inverse	Logical AND inverse operation (Series connection of normally closed contacts)	 Specifies a bit of a bit device or word device.	END End	END processing of terminating a program	Must be used as an end of a program.
OR Or	Logical OR operation (Parallel connection of normally open contacts)	 Specifies a bit of a bit device or word device.	STOP	Stopping operation	
ORI Or inverse	Logical OR inverse operation (Parallel connection of normally closed contacts)	 Specifies a bit of a bit device or word device.	SFT Shift	1-bit shift for devices	 Specifies a bit of a bit device or word device.
ANB And block	AND operation between logical blocks (Series connection of blocks)		SFTP Shift P	1-bit shift for devices (pulsing operation)	 Specifies a bit of a bit device or word device.
ORB Or block	OR operation between logical blocks (Parallel connection of blocks)		NOPLF	Ignored (for a page break at printing)	
MPS Push	Starting a branch		PAGE	Ignored (Recognized as zero step of n-page)	

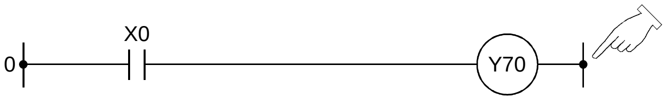
<List of instructions not explained in this chapter: part 2>
The instructions listed below are intended for the Q series and not supported by the A series.
Some of them are explained in "Q Programming Applied Course".
Refer to "MELSEC-Q/L Programming Manual Common Instruction" for more details.

Instruction symbol (Name)	Function	Drawing (devices to be used)	Instruction symbol (Name)	Function	Drawing (devices to be used)
LDP Load P	Starting to operate a rising pulse	 Specifies a bit of a bit device or word device.	MEF	Converting a result into a falling pulse	 Specifies a bit of a bit device or word device.
LDF Load F	Starting to operate a falling pulse	 Specifies a bit of a bit device or word device.	INV Inverse	Inverting the operation results	 Specifies a bit of a bit device or word device.
ANDP And P	Series connection of rising pulses	 Specifies a bit of a bit device or word device.	EGP Edge P	Converting a result into a rising pulse (Memorized by Vn)	 Specifies a bit of a bit device or word device.
ANDF And F	Series connection of falling pulses	 Specifies a bit of a bit device or word device.	EGF Edge F	Converting a result into a falling pulse (Memorized by Vn)	 Specifies a bit of a bit device or word device.
ORP Or P	Parallel connection of rising pulses	 Specifies a bit of a bit device or word device.	FF	Inverting a device output	 Specifies a bit of a bit device or word device.
ORF Or F	Parallel connection of falling pulses	 Specifies a bit of a bit device or word device.	DELTA Delta	Converting a direct output into a pulse	
MEP	Converting a result into a rising pulse	 Specifies a bit of a bit device or word device.	DELTAP Delta P	Converting a direct output into a pulse	

4.2 Differences between OUT and SET/RST

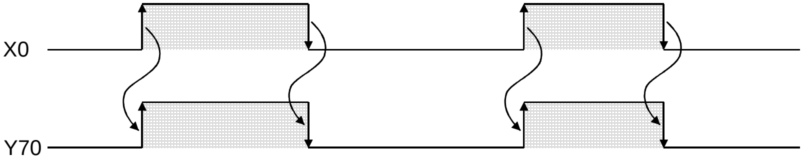
Project name	QB-1
Program name	MAIN

OUT instruction



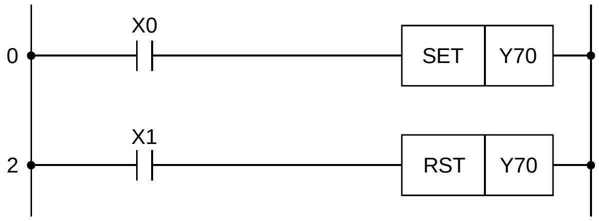
• The OUT instruction turns the specified device on when the input condition turns on, and turns the device off when the condition turns off.

[Timing chart]



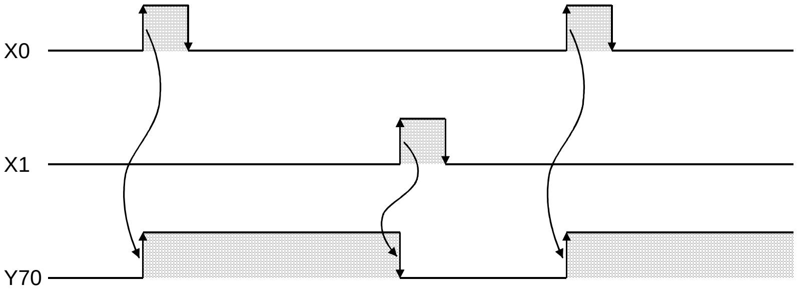
Project name	QB-2
Program name	MAIN

SET/RST instruction



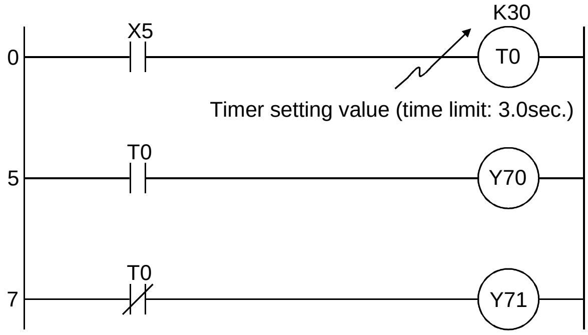
• The SET instruction turns the specified device on when the input condition turns on, and holds the on status of the device even when the condition turns off.
To turn off the device, use the RST instruction.

[Timing chart]



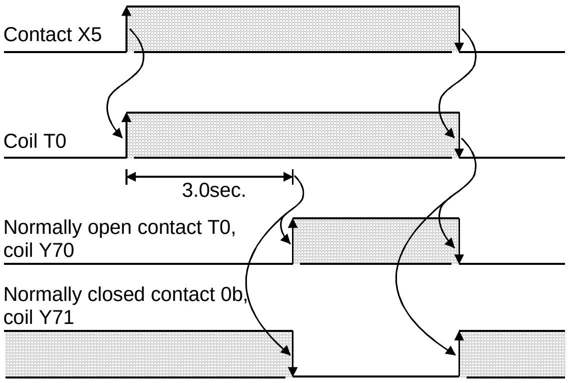
4.3 Measuring Timer

Project name	QB-3
Program name	MAIN



*: OUT T is a 4-step instruction.

[Timing chart]


Contact X5
Coil T0
3.0sec.
Normally open contact T0, coil Y70
Normally closed contact 0b, coil Y71

- The timer contact operates delaying by a set time after the coil is energized. (On delay timer)
- The timer setting range is from K1 to K32767.
  Low-speed (100ms) timer 0.1 to 3276.7sec.
  High-speed (10ms) timer 0.01 to 327.67sec.
- When the timer setting value is set to 0, it turns on (time-out) by the execution of the instruction.

• The following four types of timer are available.

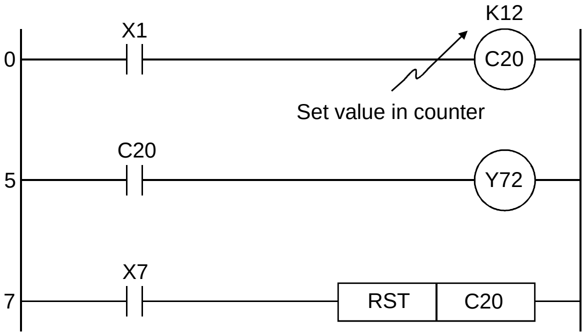
Type		Timer No. (default)
Low-speed timer ············	Counts time in units of 100ms.	Default T0 to T2047 (2048)
High-speed timer ············	Counts time in units of 10ms.	
Low-speed retentive timer ············	Accumulates time in units of 100ms.	Default: 0 (The value can be changed using the parameter.)
High-speed retentive timer ············	Accumulates time in units of 10ms.	

Refer to section 6.4 for explanation on the retentive timers.

- Change the output instruction (OUT) to OUTH to select the high-speed timer or high-speed retentive timer.
- To use the retentive timer, set the device points for the retentive timer in the device setting of the PLC parameter.

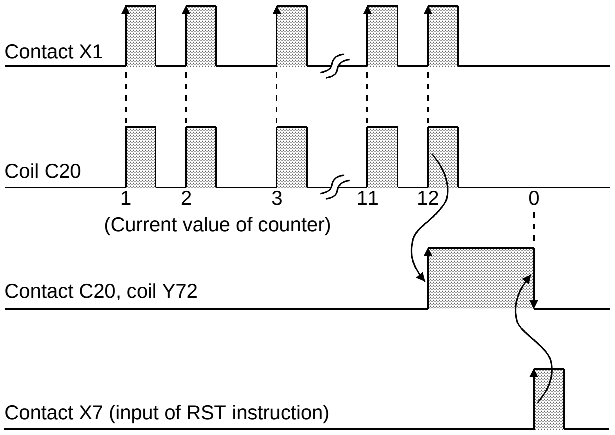
4.4 Counting by Counter

Project name	QB-4
Program name	MAIN



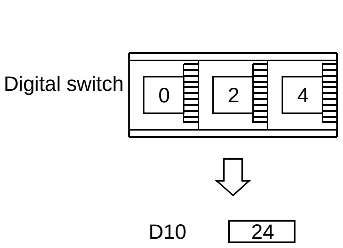
*: OUT C is a 4-step instruction.

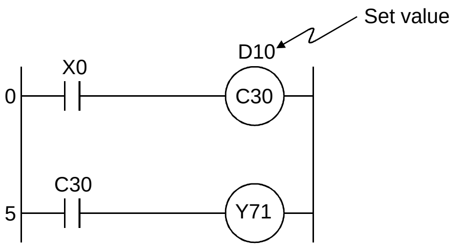
[Timing chart]



- The counter counts when an input signal rises.
- After the count, the subsequent input signals are not counted.
- Once the counter counts, the contact status and the current counter value do not change until the RST instruction is executed.
- Executing the RST instruction before the count returns the counter to 0.
- The counter setting range is from K0 and K32767. (K0 turns on (counts up) by the execution of the instruction.)

- In addition to the direct specification using K, indirect specification using D (data register) is available.





- The counter C30 counts when the number of rising edges on the input signal X0 becomes the same as the number (such as 24) specified by the data register D10.
- This indirect specification is useful for applying a value specified with an external digital switch to the counter.

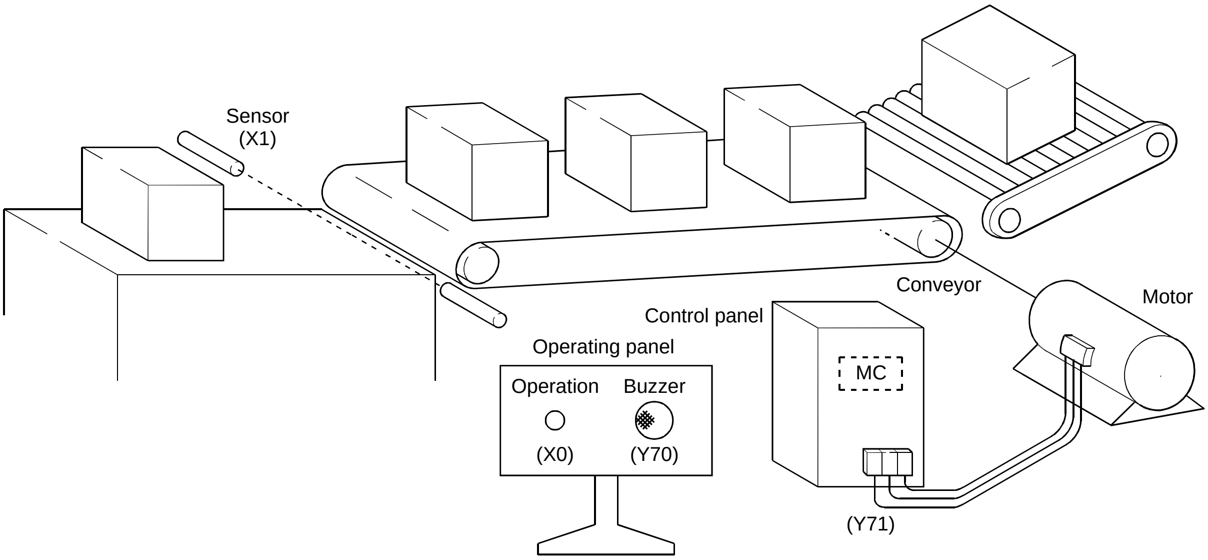
The indirect specification using the data register D is also available for the timer.

Project name	QEX1
Program name	MAIN

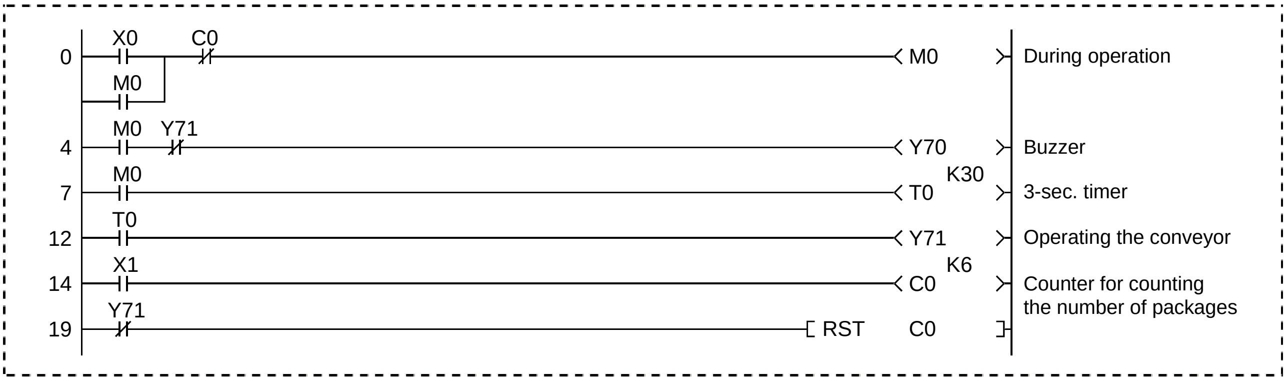
Ladder example

When the conveyor belt operation start switch (X0) is turned on, the buzzer (Y70) beeps for three seconds and the conveyor belt (Y71) starts to operate.

The conveyor belt automatically stops when the sensor (X1) detects that six packages have passed through.



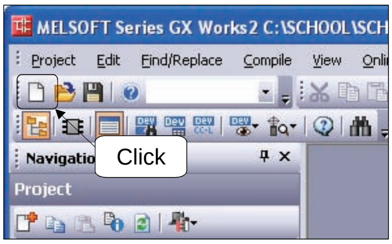
Create the following ladder and check that it operates properly.



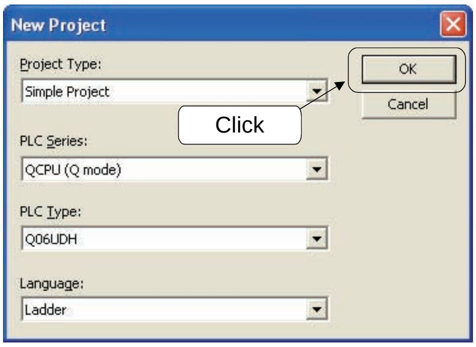
Operating Procedure

(1) Creating a new project

(a) Click 🗋 on the toolbar.



(b) The New Project dialog box is displayed.
Set "Project Type" to "Simple Project", "PLC Series" to "QCPU (Q mode)", and "PLC Type" to "Q06UDH". Then click the OK button.



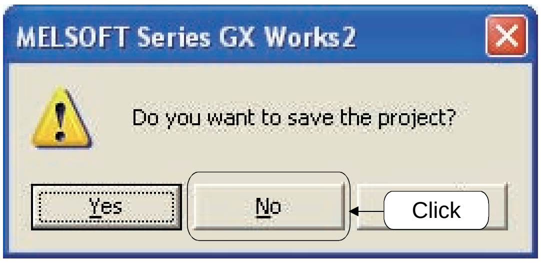
Project Type: Simple Project

PLC Series: QCPU (Q mode)

PLC Type: Q06UDH

Language: Ladder

(c) If the project in preparation exists, the confirmation dialog box for saving the project is displayed.
Click the No button.

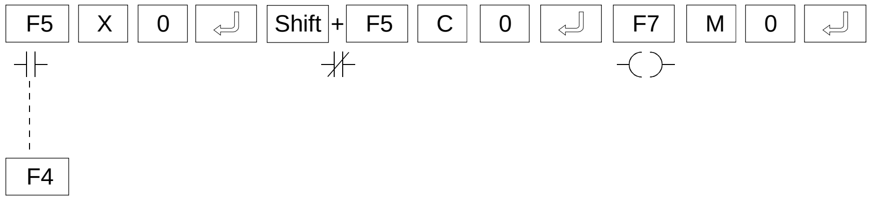


Do you want to save the project?

(d) The screen shifts to the new project creation mode.

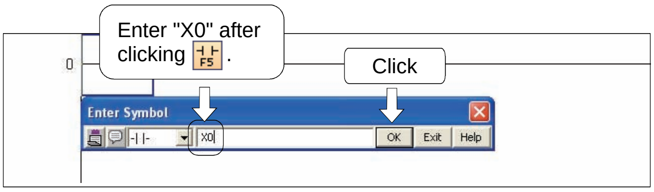
(2) Creating a program

[Using the keyboard]



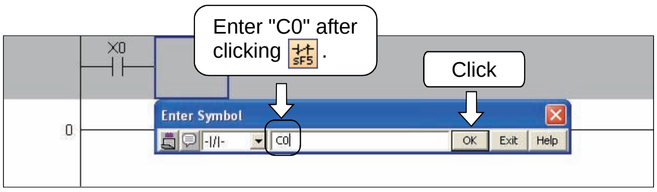
Conversion

[Using the tool buttons]



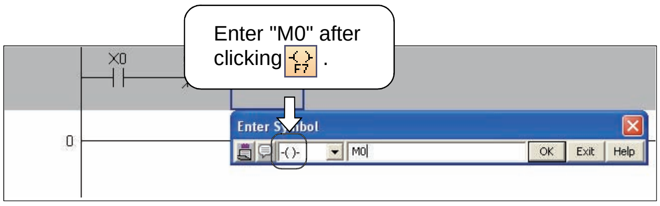
(a) Click ⊣⊢ on the toolbar to open the Enter Symbol window.

(b) Enter "X0" with the keyboard and click the OK button.



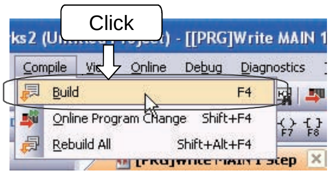
(c) Click ⊣/⊢ on the toolbar to open the Enter Symbol window.

(d) Enter "C0" with the keyboard and click the OK button.



(e) Click -( )- on the toolbar to open the Enter Symbol window.

(f) Enter "M0" with the keyboard and click the OK button.



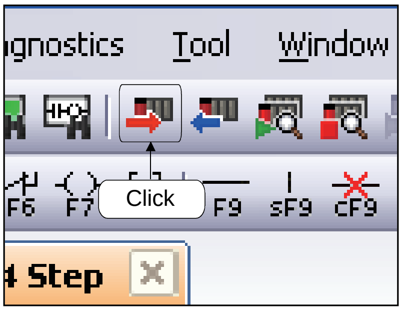
(g) When creating the circuit is finished, click [Compile] → [Build].

(3) Writing the project to the programmable controller

(a) Write the created ladder to the memory on the programmable controller.

Set the RUN/STOP switch of the CPU to STOP.

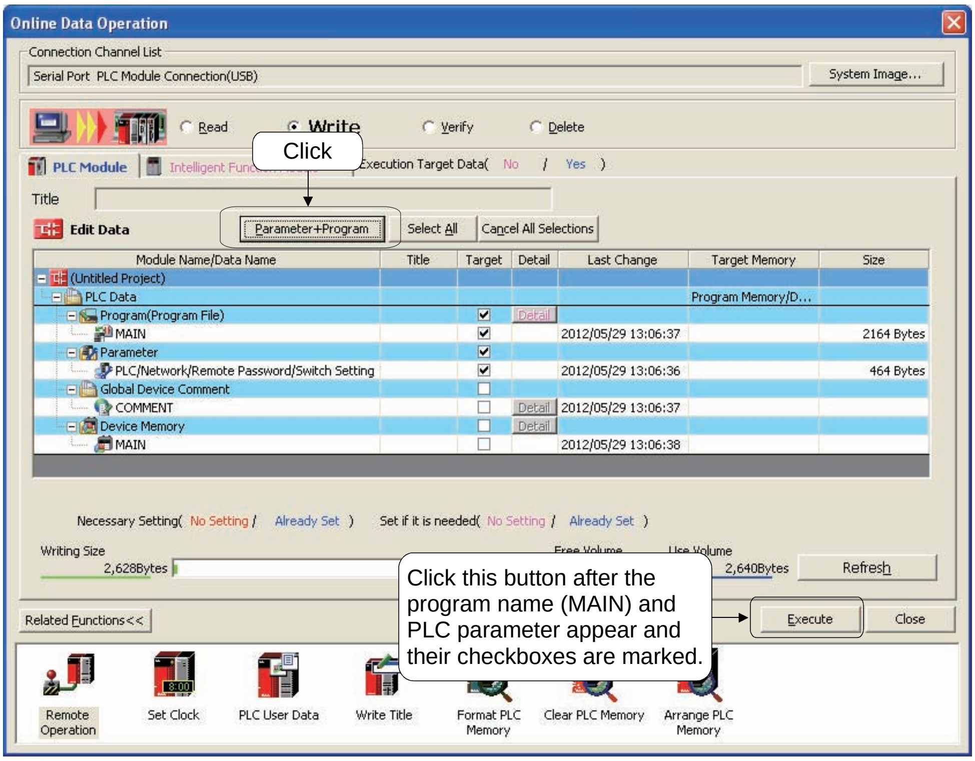
Click on the toolbar.
The Online Data Operation dialog box is displayed.



Click

(b) Click the Parameter + Program button. Checkboxes for the target program and the target parameter displayed in the window are automatically marked (✓).

(c) Click the Execute button.



Connection Channel List
Serial Port PLC Module Connection(USB)

○ Read ● Write ○ Verify ○ Delete

Execution Target Data(No / Yes)

Title

Module Name/Data Name	Title	Target	Detail	Last Change	Target Memory	Size
(Untitled Project)						
PLC Data					Program Memory/D...	
Program(Program File)		☑	Detail			
MAIN		☑		2012/05/29 13:06:37		2164 Bytes
Parameter		☑				
PLC/Network/Remote Password/Switch Setting		☑		2012/05/29 13:06:36		464 Bytes
Global Device Comment		☐				
COMMENT		☐	Detail	2012/05/29 13:06:37		
Device Memory		☐	Detail			
MAIN		☐		2012/05/29 13:06:38		

Necessary Setting(No Setting / Already Set) Set if it is needed(No Setting / Already Set)

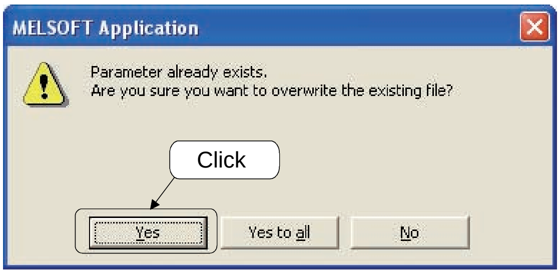
Writing Size 2,628Bytes

Free Volume Use Volume 2,640Bytes

Click

Click this button after the program name (MAIN) and PLC parameter appear and their checkboxes are marked.

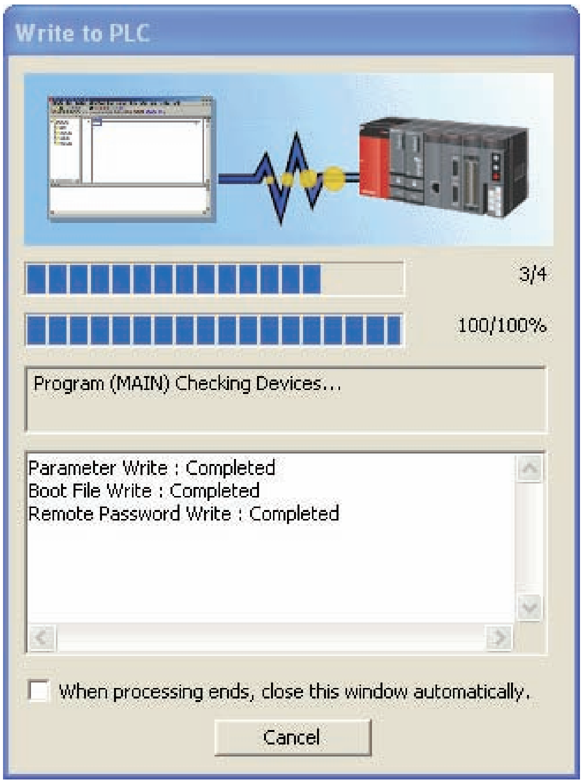
(d) If parameters have been already written, the confirmation dialog box for overwriting the parameters is displayed. Click the Yes button.



MELSOFT Application

Parameter already exists.
Are you sure you want to overwrite the existing file?

(e) The Write to PLC dialog box is displayed.



Write to PLC

3/4

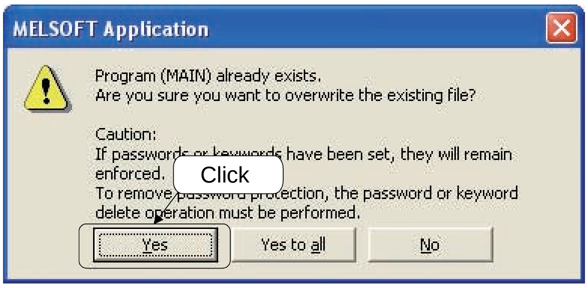
100/100%

Program (MAIN) Checking Devices...

Parameter Write : Completed
Boot File Write : Completed
Remote Password Write : Completed

□ When processing ends, close this window automatically.

(f) If a program has already been written, the confirmation dialog box for overwriting the program is displayed. Click the Yes button.

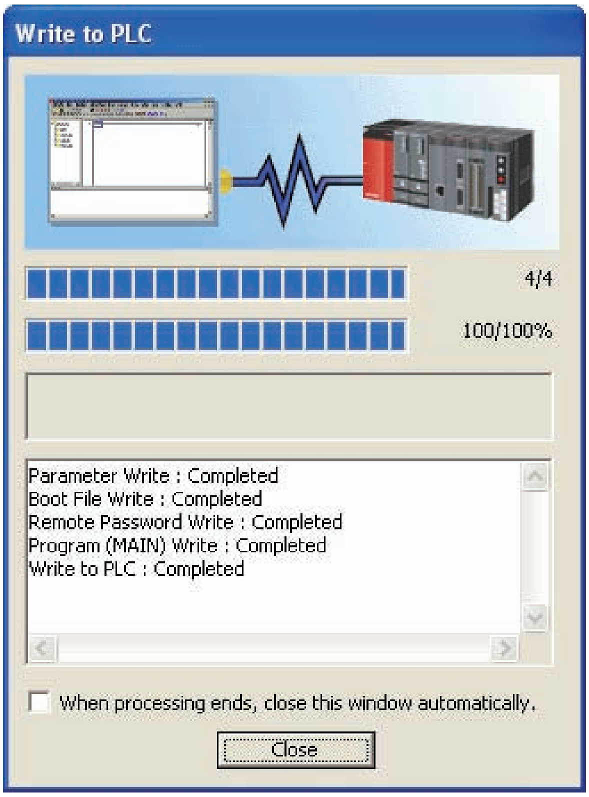


MELSOFT Application

Program (MAIN) already exists.
Are you sure you want to overwrite the existing file?

Caution:
If passwords or keywords have been set, they will remain enforced.
To remove password protection, the password or keyword delete operation must be performed.

(g) Writing the program to the programmable controller is finished.



Parameter Write : Completed
Boot File Write : Completed
Remote Password Write : Completed
Program (MAIN) Write : Completed
Write to PLC : Completed

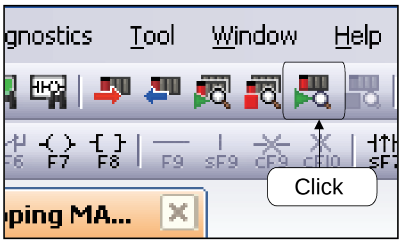
□ When processing ends, close this window automatically.

(4) Monitoring the ladder

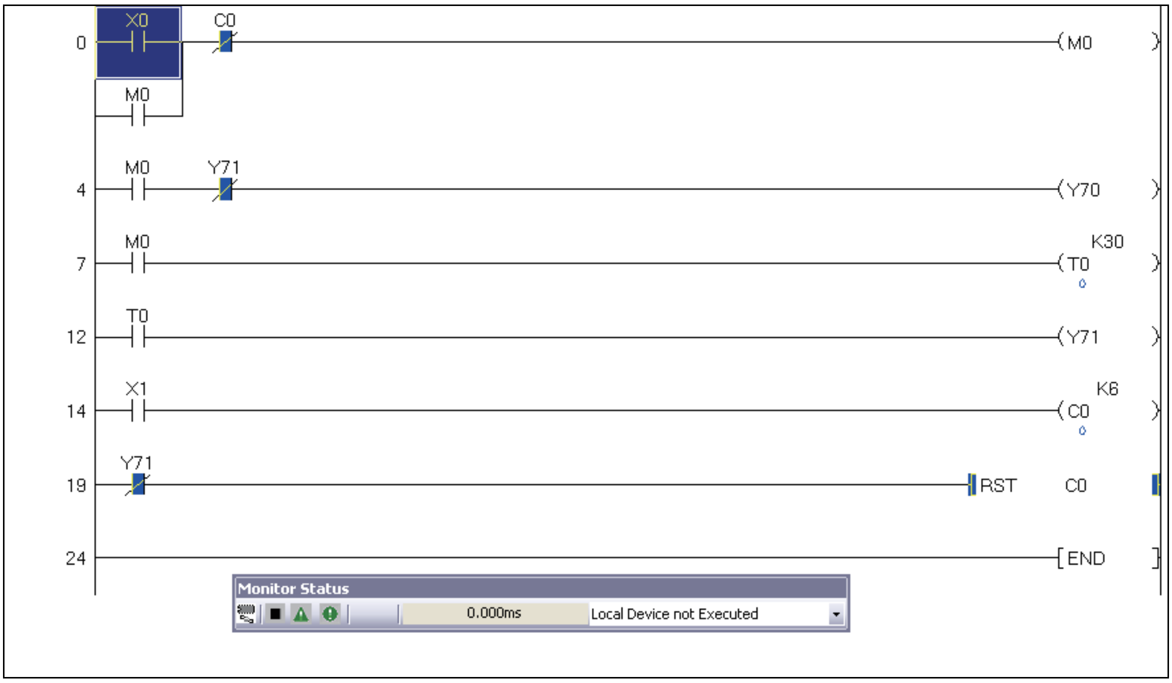
Monitor the ladder.

> Hold the RESET/STOP/RUN switch on the CPU at the RESET position for one second or more, then set the switch to RUN.

(a) Click on the toolbar.



(b) The ladder (write) screen is used to monitor the ladder.



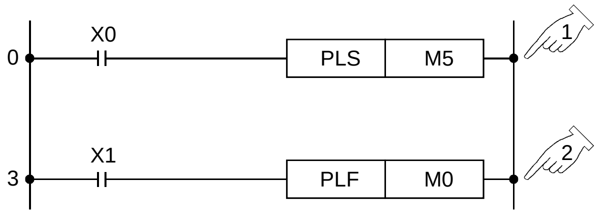
Operation Practice

1) Turning on the push button switch (X0) turns on Y70 and starts T0 at the same time.
2) When the timer T0 counts three seconds (time-out), Y70 turns off and Y71 turns on at the same time.
3) Turn on or off (push or release) the push button switch (X1). The counter C0 counts the number of ON to turn off Y71 after counting on six times.

4.5 PLS Pulse (turns on the specified device for one scan at rising edge of an input condition.)

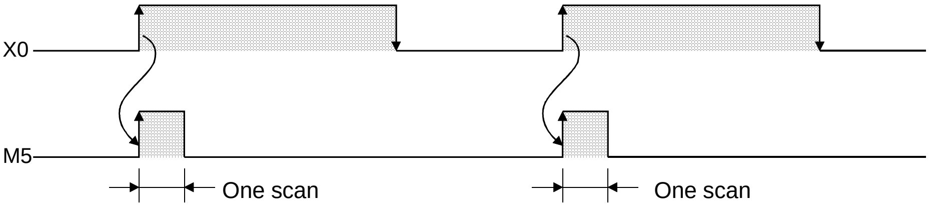
PLF Pulf (turns on the specified device for one scan at falling edge of an input condition.)

Project name	QB-5
Program name	MAIN



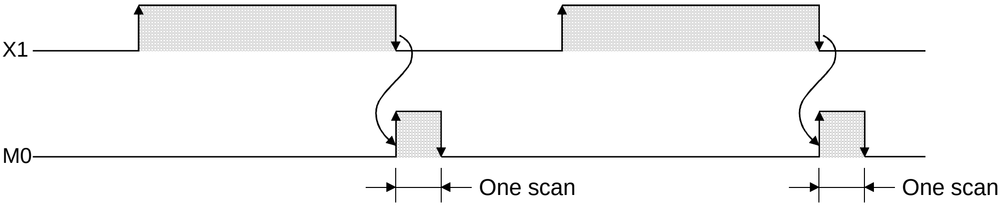
1 • The PLS instruction turns on the specified device only for one scan when the execution command is turned on from off.

[Timing chart]



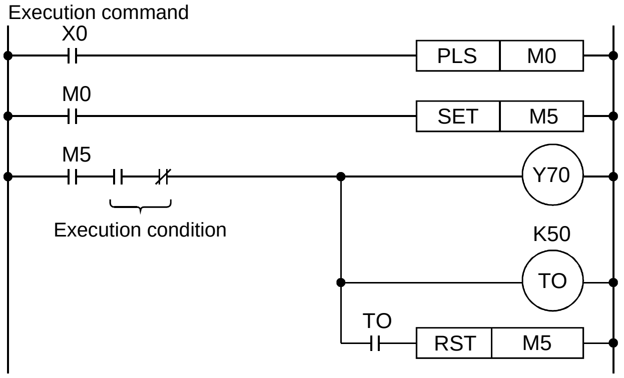
2 • The PLF instruction turns on the specified device only for one scan when the execution command is turned off from on.

[Timing chart]

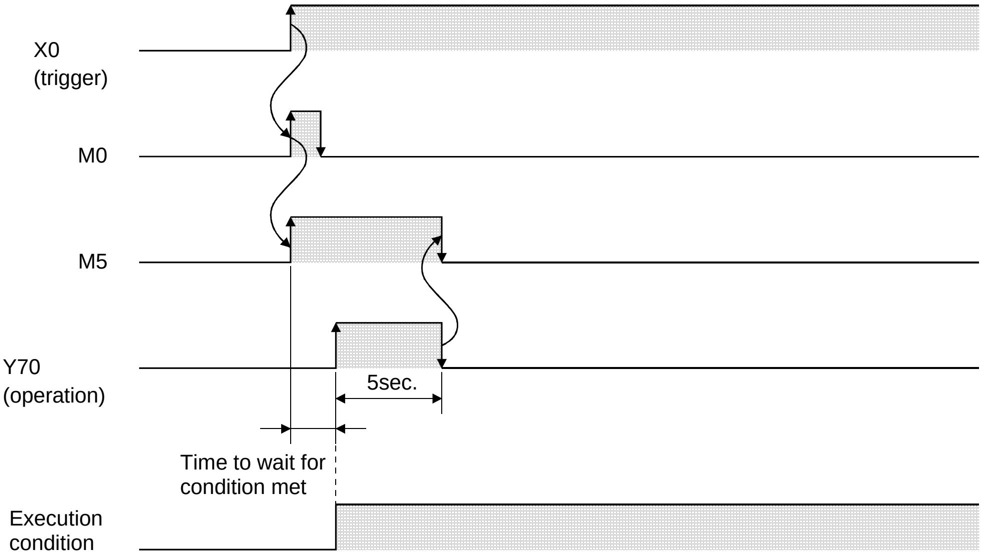


Application

- The instructions can be used in the standby program that waits for the operation condition.

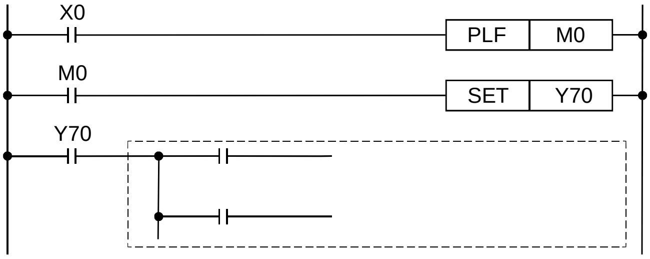


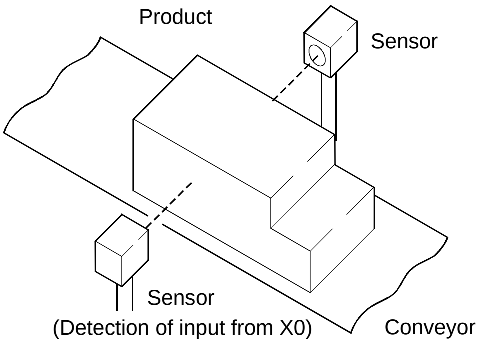
[Timing chart]



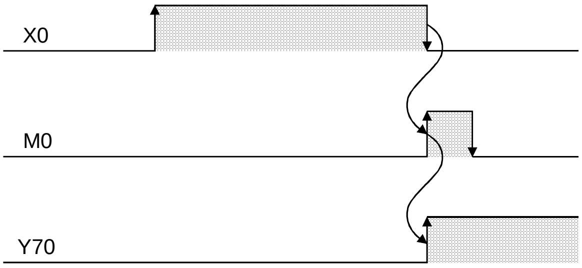
		Applicable device												Digit	Number of basic steps
		Internal device (system or user)		File register	MELSECNET/ 10 (H) Direct Jn\		Intelligent function module Un\G	Index register	Constant		Pointer		Level		
		Bit	Word	R	Bit	Word		Z	K	H	P	I	N		
PLS Ⓓ PLF Ⓓ	Ⓓ	○	○	○	○	○	○								2

- The instructions can be used for a program that detects passage of moving objects.
  After the passage of a product is detected, the next process for the product is started.





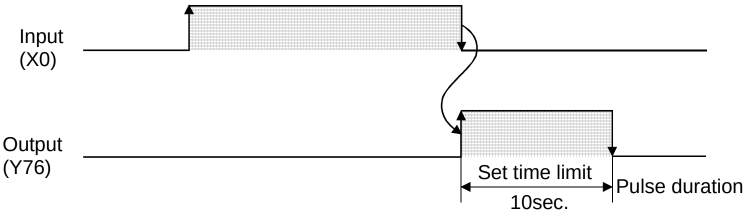
[Timing chart]



Part 1

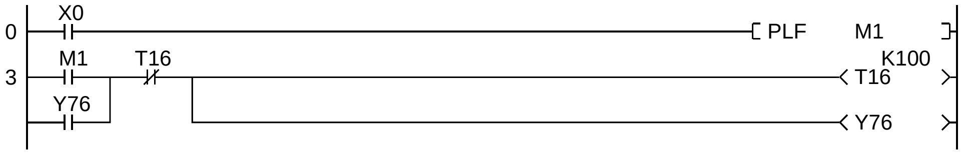
- The instructions can be used for a program that executes the output operation for a set period of time when the input signal changes from on to off.

[Timing chart]



[Program example]

Project name	QB-6
Program name	MAIN

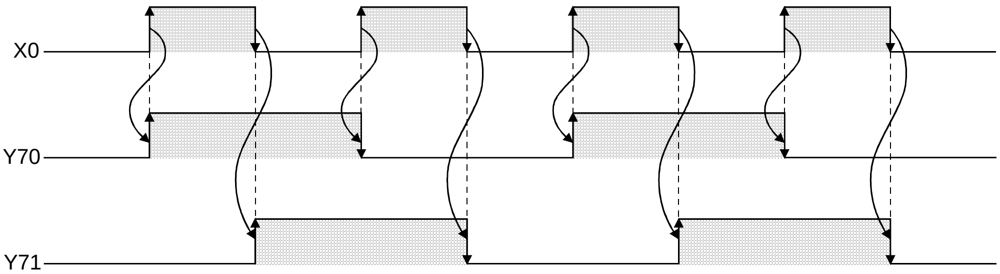


Other Useful Ways of PLS and PLF Part 2

- The program for the repeated operation such as switching on/off status alternately by pressing the push button switch can be made with the instructions.

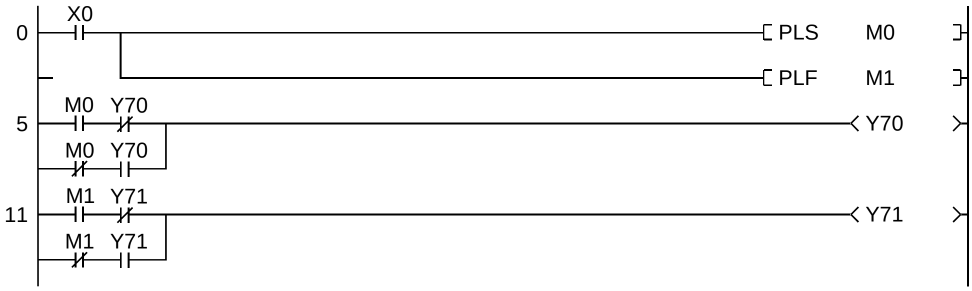
  (If the PLS instruction is used in the above program, the rising edge caused when the push button switch is pressed triggers the program. If the PLF instruction is used, the falling edge caused when the switch is released is the trigger.)

[Timing chart]



[Program example]

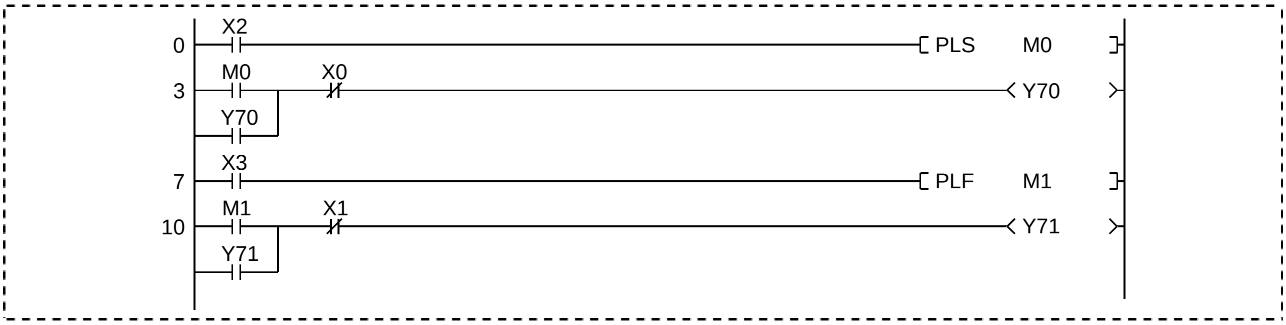
Project name	QB-7
Program name	MAIN



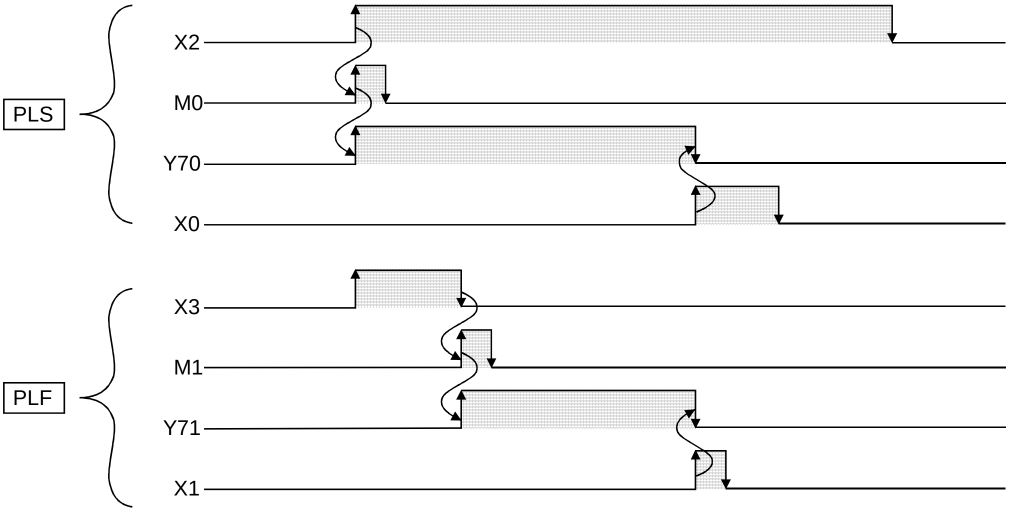
Project name	QEX2
Program name	MAIN

Ladder example

Create the following ladder and check that it operates properly.

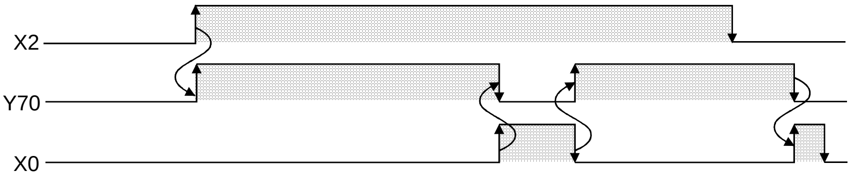


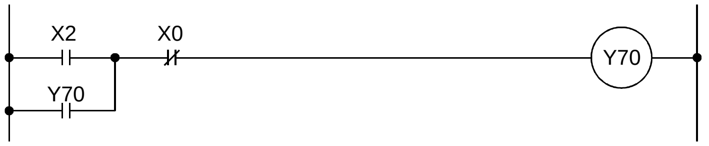
[Timing chart]



REFERENCE

The following is a timing chart of a lockup ladder programmed using the OUT instruction. Compare this with the lockup ladder created using the PLS instruction.





Operating Procedure

The following procedures are the same as the Operating Procedure in section 4.4.

(1) Creating a new project

(2) Creating a program

(3) Writing the project to the programmable controller

(4) Monitoring the ladder

Operation Practice

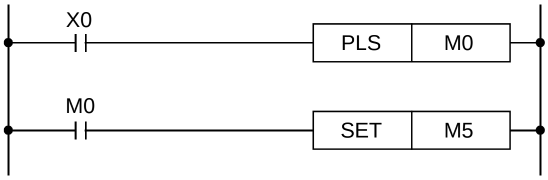
- Turning on X2 turns on Y70, and turning on X0 turns off Y70. (Even when X2 stays on, turning on X0 turns off Y70.)
- Turning on X3 turns on Y71, and turning on X1 turns off Y71.

Related Exercise —— Exercise 3

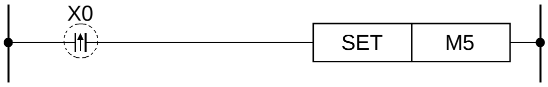
REMARK

Input pulse processing is not required for the QCPU as it uses a derivative contact (|↑|/|↓|).

[For A/AnSCPU]



[For QCPU]

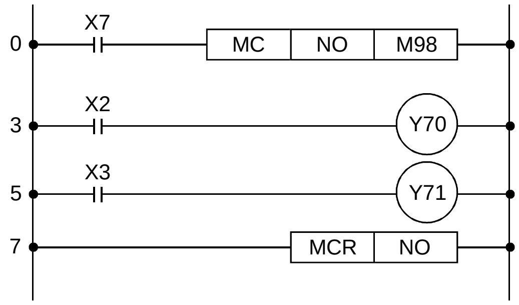


Supported instructions are; LDP, LDF, ANDP, ANDF, ORP, and ORF.

4.6 MC Master Control (Start)

MCR Master Control Reset (End)

Project name	QB-8
Program name	MAIN



- The above program is a basic one.
- MC N□ M□ to MCR N□ (indicated as "MC to MCR" hereafter.)
  The available nesting (N) numbers for "MC to MCR" are from N0 and N14.
- The scan time skipped by "MC to MCR" hardly changes.

The device status of the program skipped by "MC to MCR" becomes as follows;
All the devices in the OUT instruction are turned off.
The devices in the SET, RST, and SFT instructions, the counter, and retentive timer keep their statuses.
The 100ms timer and 10ms timer are reset to 0.

Application

- The instructions can be used for a program for switching between manual and automatic operations. (Refer to Ladder example.)

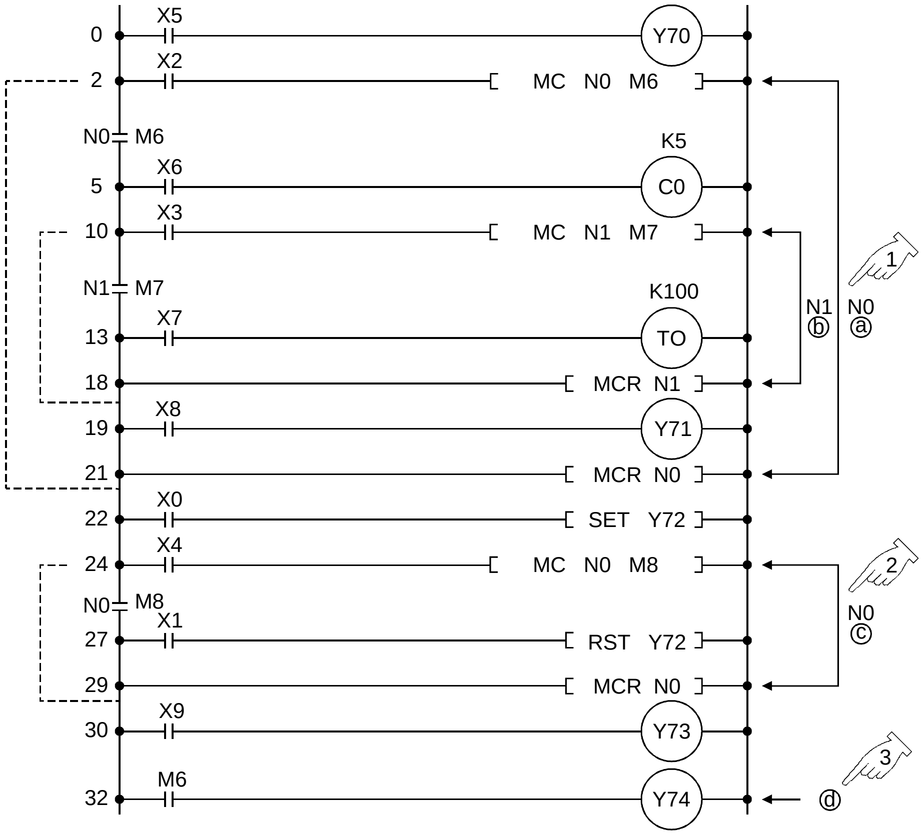
		Applicable device													Digit	Number of basic steps
		Internal device (system or user)		File register	MELSECNET/10 (H) Direct Jn\		Intelligent function module Un\G	Index register	Constant		Pointer		Level			
		Bit	Word	R	Bit	Word		Z	K	H	P	I	N			
MC n Ⓓ MCR n	n												○			2/1
	Ⓓ	○	○	○	○	○	○									

The number of basic steps of the MC instruction is two, and that of the MCR instruction is one.

Nested "MC to MCR" Program Example

- The MC and MCR instructions can be nested as shown below.

Project name	QB-9
Program name	MAIN



1 • The "MC to MCR" program ⓐ is nested under the "MC to MCR" program ⓑ. (It is called "nested structure".)
In this case;
1) Assign the nesting number (N) of the MC instructions in ascending order.
2) Assign the nesting number (N) of the MCR instructions used for the MC in descending order.

2 • The "MC to MCR" program ⓐ can be independent from the ⓒ program. The same nesting numbers (N) can be used in the both programs.
• The internal relay number (M) must be changed for each MC instruction.

3 • As shown in the ⓓ program, the internal relay number M □ of MC can be used as a contact.

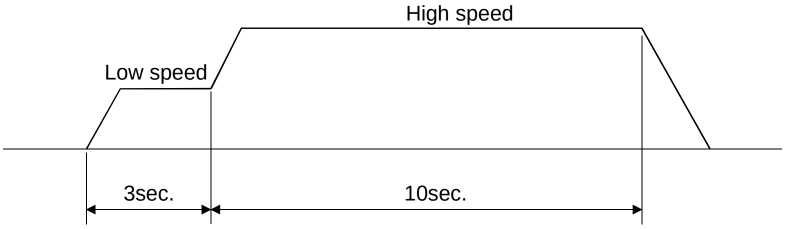
Note) In GX Works2, the on/off status of the master control is displayed in the title tag on the monitor screen.

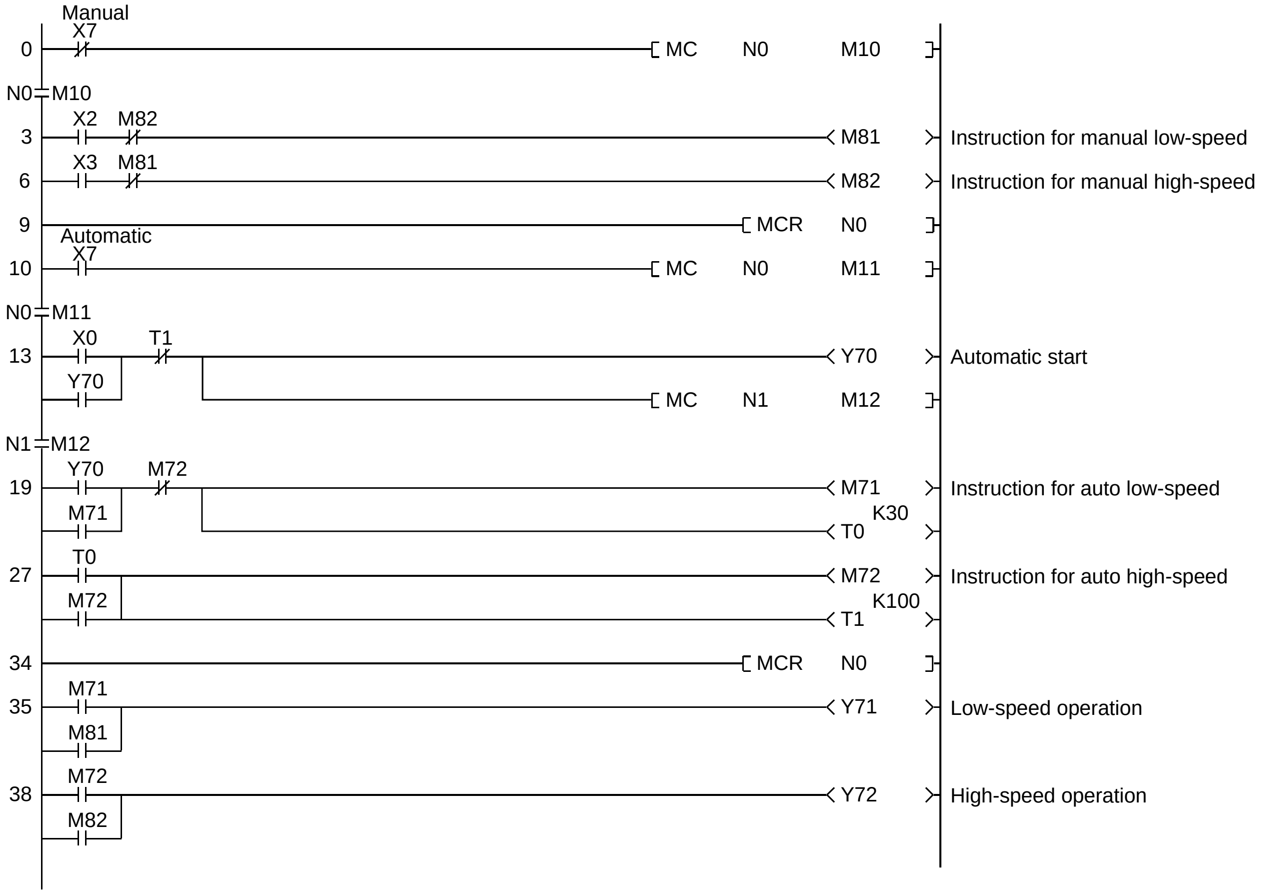
Project name	QEX3
Program name	MAIN

Ladder example

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

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





Note) In GX Works2, the on/off status of the master control is displayed in the title tag on the monitor screen.

Operating Procedure

The following procedures are the same as the Operating Procedure in section 4.4.

(1) Creating a new project

(2) Creating a program

(3) Writing the project to the programmable controller

(4) Monitoring the ladder

Operation Practice

- The manual operation is selected by turning off the X7 switch.
  When the X2 switch is turned on, Y71 lights and the low-speed operation is executed. To select the high-speed operation, turn on the X3 switch. Y72 lights and the high-speed operation starts.
- The automatic operation is selected by turning on the X7 switch.
  When the X0 switch is turned on, Y70 lights indicating that the automatic operation is activated.
  At the same time, Y71 also lights for 3sec. indicating the system is in the low-speed mode. After the 3sec. have elapsed, Y72 lights for 10sec. indicating that the system is in high-speed mode. Then the operation is stopped. (Y70, Y71, and Y72 have stopped lighting at the end.)

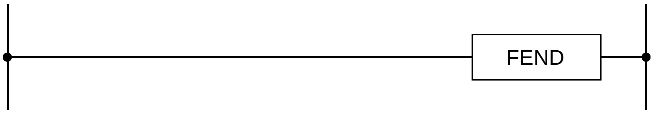
NOTE

For the MCR instructions in one nested program block, all master controls in the program can be terminated with the lowest nesting (N) number only.

4.7 FEND / CJ / SCJ / CALL / RET

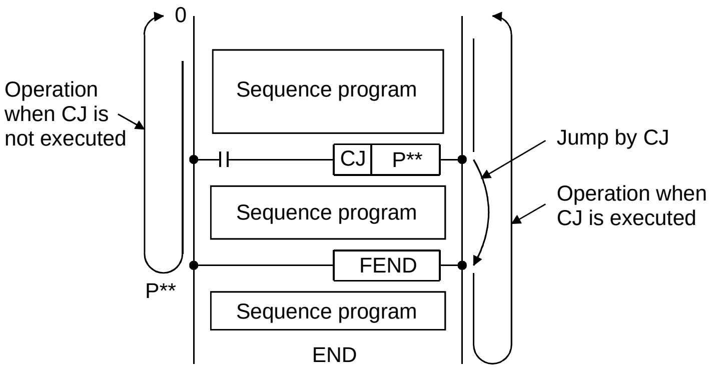
Project name	QB-10
Program name	MAIN

4.7.1 FEND F end

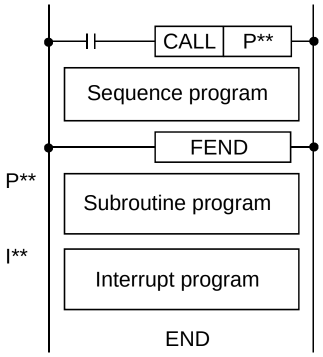


FEND is a 1-step instruction.

- Use the FEND instruction as the END instruction under the following conditions;
  1) When a sequence program must be executed and terminated in each program block.
     For example, use the FEND instruction with the CJ and SCJ instructions.
  2) When using subroutine programs (CALL and RET instructions)
  3) When using an interrupt program
- After each execution of the FEND instruction, the programmable controller processes the current value of the timer and counter and executes self-diagnostic check, and then re-operates from the step 0.



(a) When operating in each program block by the CJ instruction



(b) When using the subroutine and interrupt programs

NOTE

- There is no limit on the number of the FEND instructions in a sequence program, however, they cannot be used in the subroutine and interrupt programs.
- The FEND instruction cannot be used to terminate the main or sub sequence program.
  Make sure to use the END instruction for the end of a whole program.

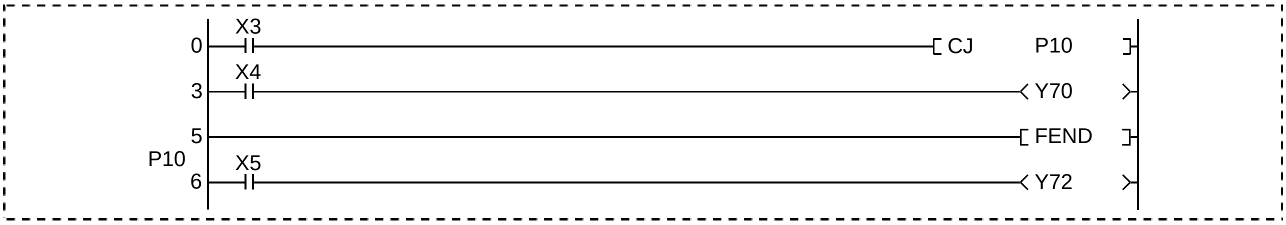
REFERENCE

The interrupt program stops the current process and processes an interrupt upon receiving an interrupt request while a normal program is being processed.

Project name	QEX6
Program name	MAIN

Ladder example

Create the following ladder with GX Works2 and write it to the CPU of the demonstration machine. Then check that the FEND instruction operates properly.



Operating Procedure

The following procedures are the same as the Operating Procedure in section 4.4.

(1) Creating a new project

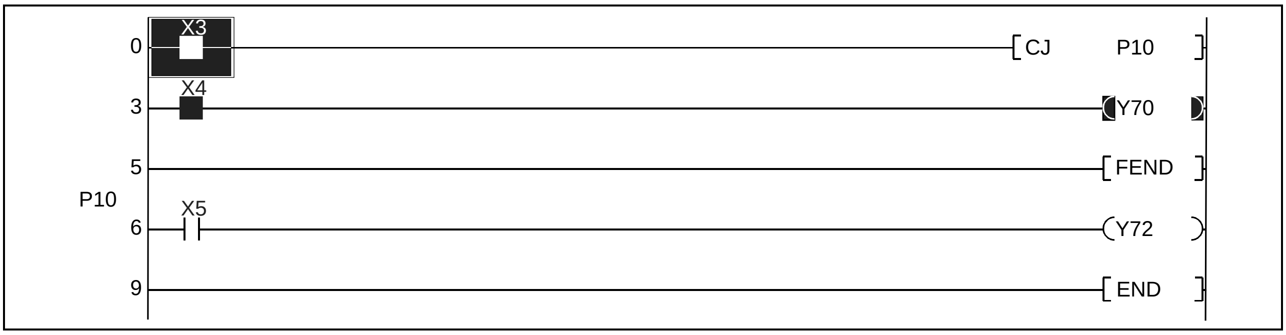
(2) Creating a program

(3) Writing the project to the programmable controller

(4) Monitoring the ladder

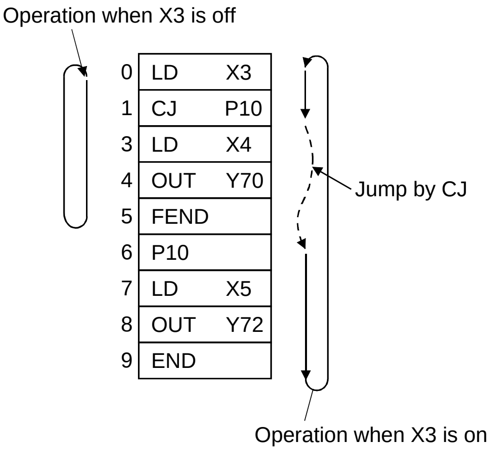
Operation Practice

Verify the operation of the ladder, which was created with GX Works2 and written to the CPU of the demonstration machine, by monitoring the ladder on the screen.



(1) When X3 is off
   (a) The operation is executed from 0 to FEND.
   (b) Turning on or off X4 turns on or off Y70.
   (c) Turning on or off X5 does not change Y72.

(2) When X3 is on
   (a) The program jumps to the pointer P10 by the CJ instruction.
   (b) Turning on or off X4 does not change Y70.
   (c) Turning on or off X5 turns on or off Y72.

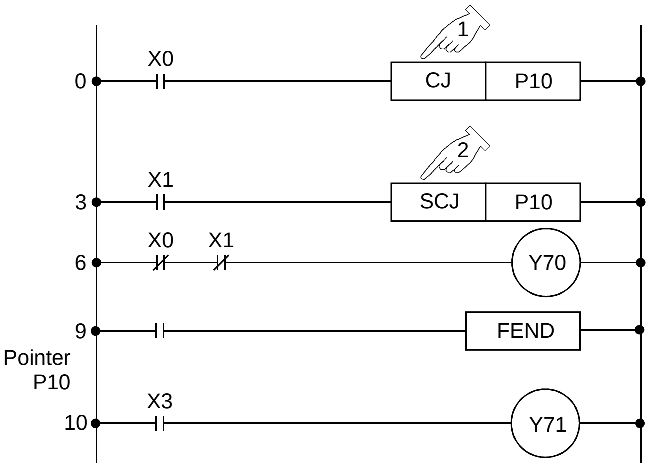


0	LD	X3
1	CJ	P10
3	LD	X4
4	OUT	Y70
5	FEND	
6	P10	
7	LD	X5
8	OUT	Y72
9	END	

Related Exercise —— Exercise 4

4.7.2 CJ (Conditional jump: instantaneous execution condition jump)

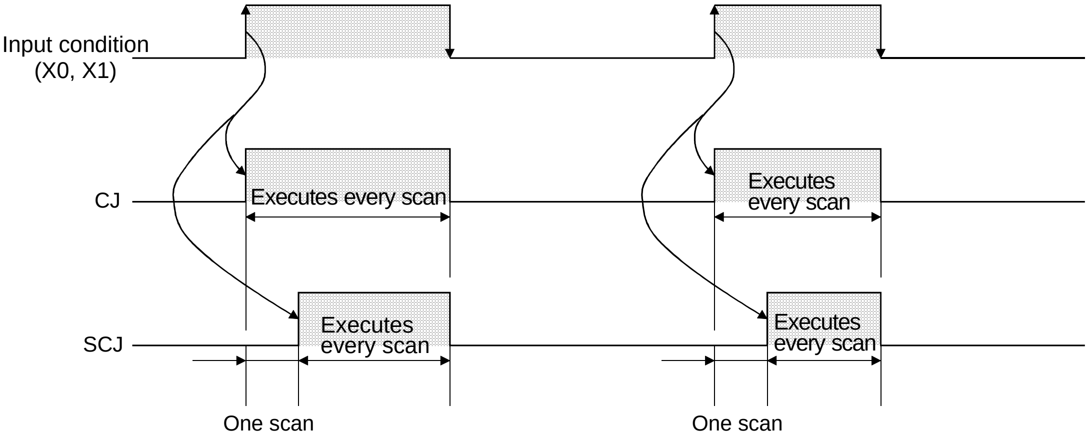
SCJ (S conditional jump: execution condition jump after one scan)



1 • The CJ instruction instantaneously executes a program jumping it to the specified address (pointer number) when the execution command is on.
When the command is off, the program is not jumped.

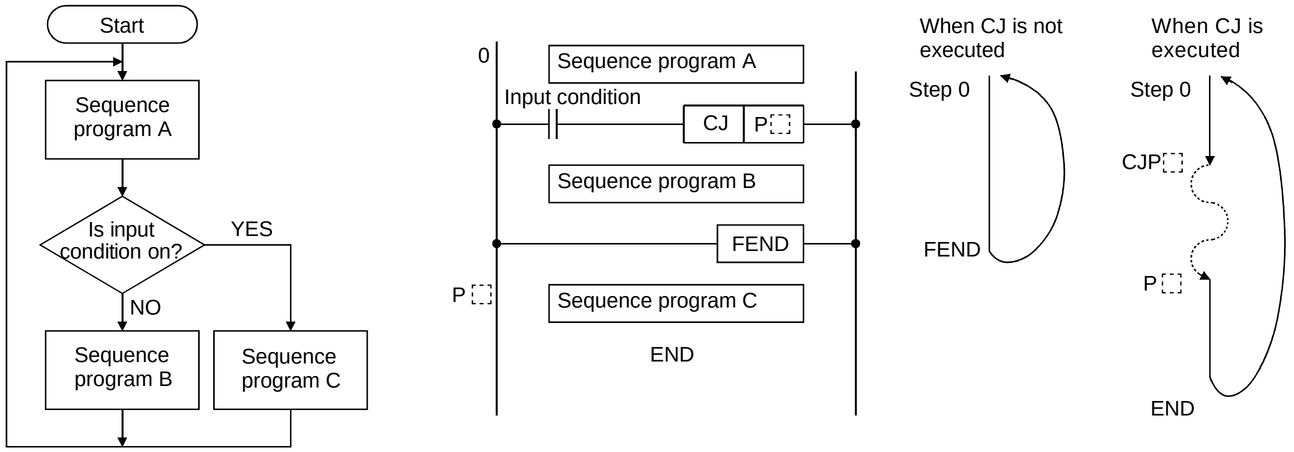
2 • The SCJ instruction executes a program without jumping it for the scan when the execution command is turned on. From the next scan, the instruction executes the program jumping it to the specified address (pointer number).
When the command is off, the program is not jumped.
• The SCJ instruction is used when some operations must be executed before jumping the program.
For example, when the output needs to be on or reset in advance.

[Timing chart]

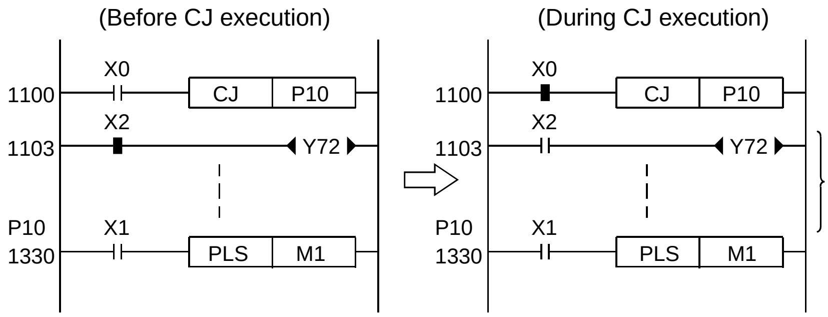


NOTE

- The pointer numbers available for both CJ and SCJ instructions are P0 to P4095.
- Use the FEND instruction as shown below when a program using the CJ and SCJ instructions must be concluded in each program block. (Refer to section 4.7.1 for FEND.)



- The status of ladders skipped by the CJ instruction remains unchanged.



Because X0 is on, all instructions within this area are not executed. Hence Y72 remains on even after X2 is turned off.

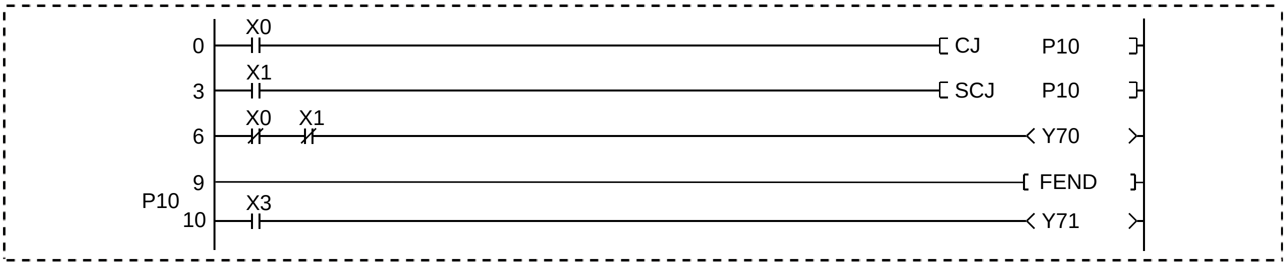
- After the timer coil has turned on, jumping the timer of a coil that is on using the CJ, SCJ, or JMP instruction interrupts an accurate measurement.

		Applicable device													Digit	Number of basic steps
		Internal device (system or user)		File register	MELSECNET/ 10 (H) Direct Jn\		Intelligent function module Un\G	Index register	Constant		Pointer		Level			
		Bit	Word	R	Bit	Word		Z	K	H	P	I	N			
CJ P** SCJ P**	P										○					2

Project name	QEX4
Program name	MAIN

Ladder example

Create the following ladder with GX Works2 and write it to the CPU of the demonstration machine. Then check the difference between the CJ and SCJ instructions.



Operating Procedure

The following procedures are the same as the Operating Procedure in section 4.4.

(1) Creating a new project

(2) Creating a program

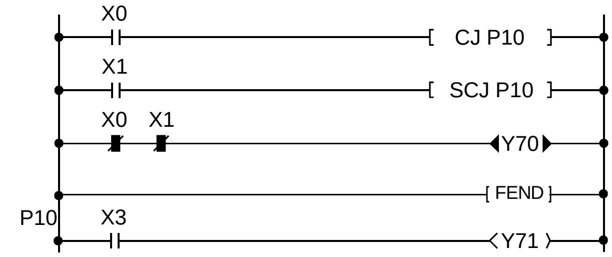
(3) Writing the project to the programmable controller

(4) Monitoring the ladder

Operation Practice

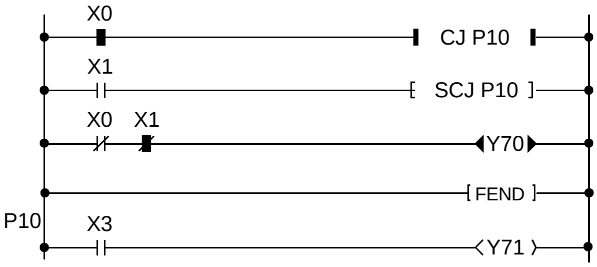
(1) When X0 and X1 are off, the CJ and SCJ instructions are not executed. Therefore, Y70 is on.

[Before CJ and SCJ execution]



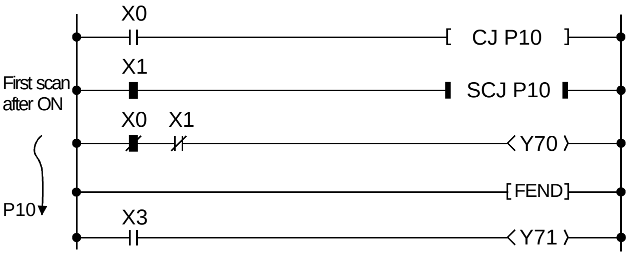
(2) When X0 is turned on, the CJ instruction is executed and the program jumps to P10. Therefore, Y70 remains on.

[During CJ execution] First scan and subsequent scans

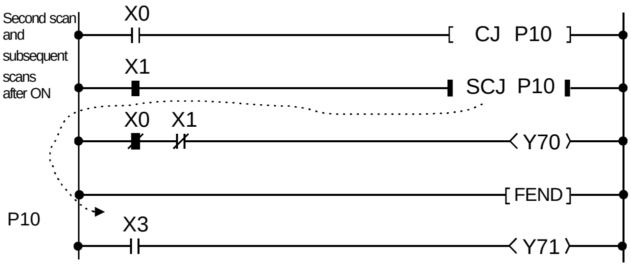


(3) Turning off X0 and on X1 executes the SCJ instruction and jumps the program to P10 from the second scan. Therefore, Y70 turns off.

[During SCJ execution] First scan

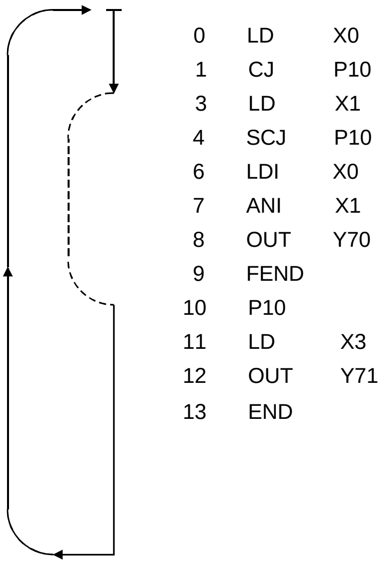


[During SCJ execution] Second scan and subsequent scans



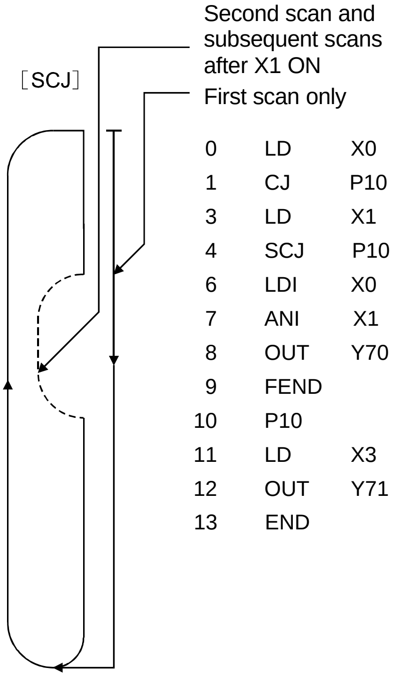
(4) Y71 is turned on or off when the CJ and SCJ instructions are executed.
- The following lists explain the difference between the CJ and SCJ instructions.

[CJ]



```
0    LD    X0
1    CJ    P10
3    LD    X1
4    SCJ   P10
6    LDI   X0
7    ANI   X1
8    OUT   Y70
9    FEND
10   P10
11   LD    X3
12   OUT   Y71
13   END
```

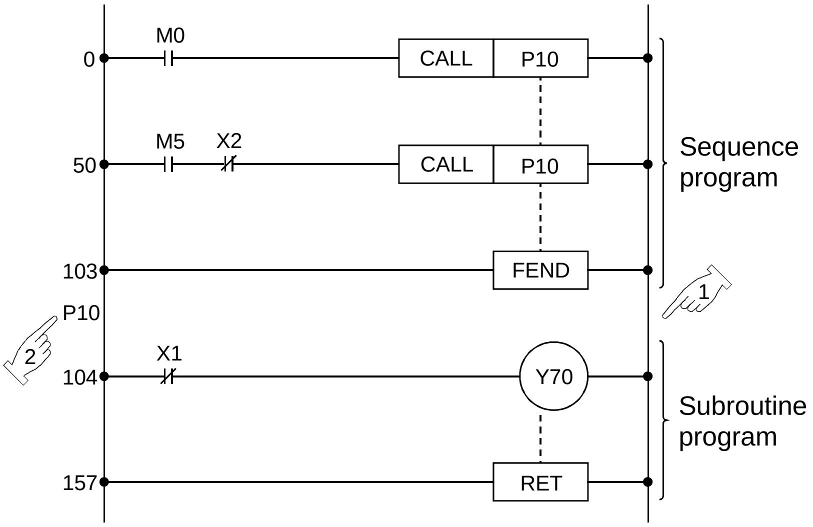
[SCJ]



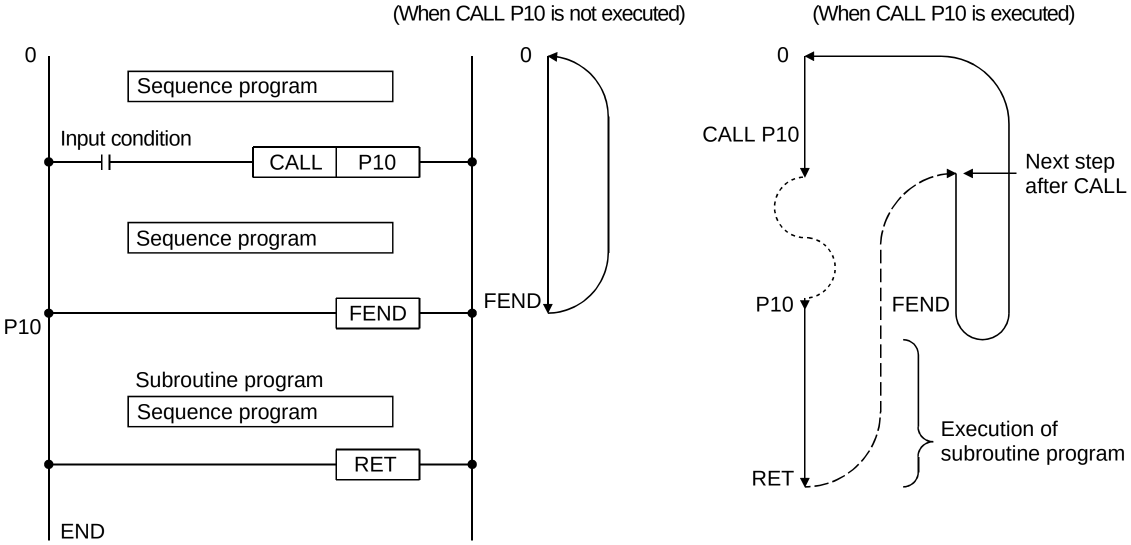
```
0    LD    X0
1    CJ    P10
3    LD    X1
4    SCJ   P10
6    LDI   X0
7    ANI   X1
8    OUT   Y70
9    FEND
10   P10
11   LD    X3
12   OUT   Y71
13   END
```

Related Exercise —— Exercise 4

4.7.3 CALL(P) Call / RET Return — Executing a subroutine program

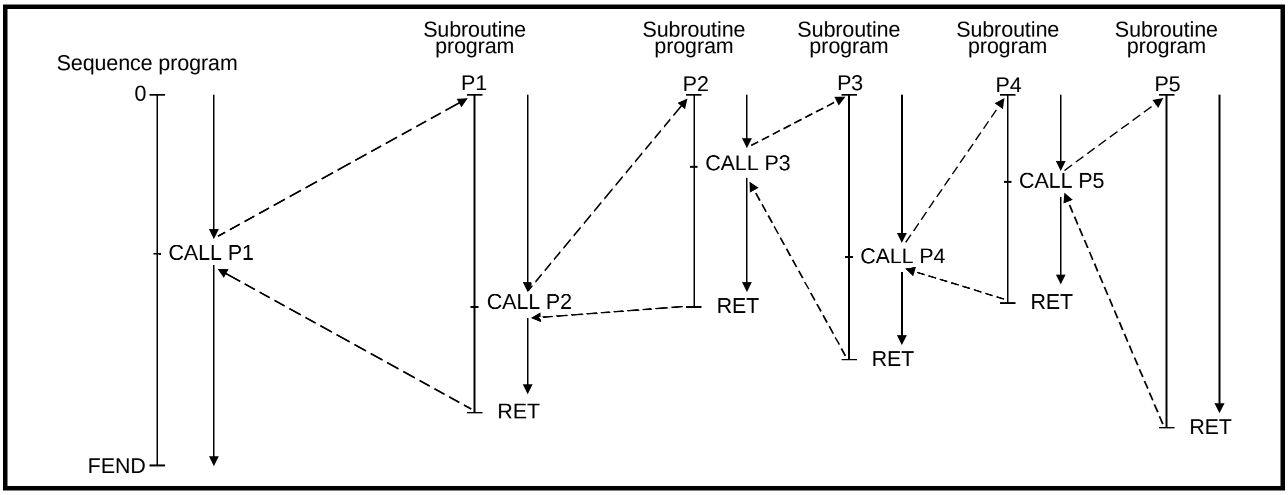


- The above program is a basic style to execute the subroutine program using the CALL and RET instructions.
  Keep this structure, otherwise an error occurs and the programmable controller stops.
- 1 A subroutine program consists of the ladders for executing the same data many times in one program.
  A subroutine program starts at a pointer P□ and ends with the RET instruction.
- 2 The pointer P number is from 0 to 4095. (Same as the pointer numbers used for the CJ and SCJ instructions.)
- A subroutine program is executed as shown in the following diagrams.

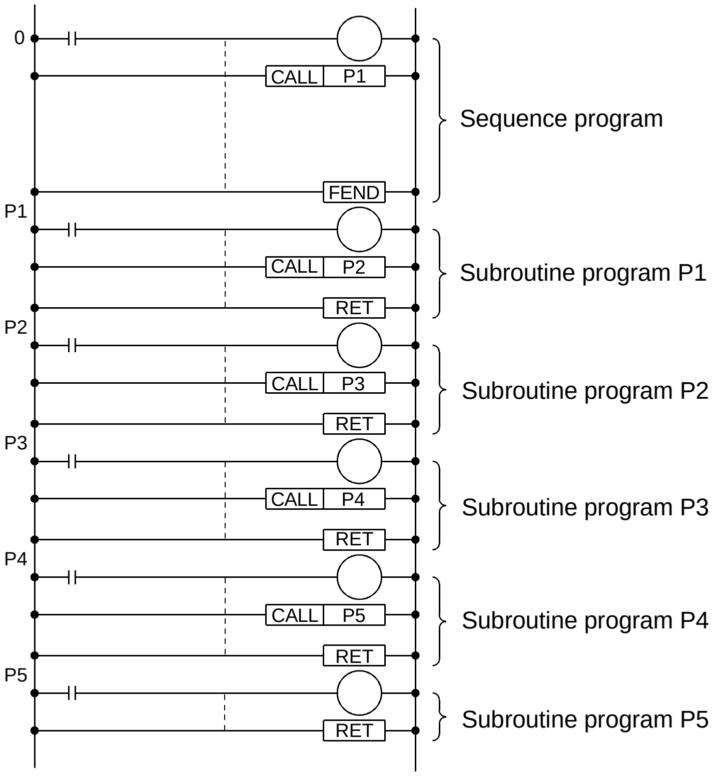


Nesting

- The CALL (P) instructions can be nested up to 16 levels.



The following ladder circuit shows the above nested program.



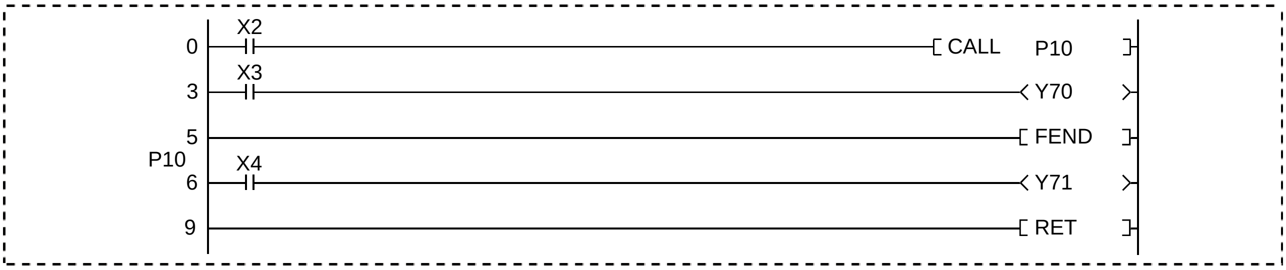
		Applicable device												Digit	Number of basic steps
		Internal device (system or user)		File register	MELSECNET/ 10 (H) Direct Jn\		Intelligent function module Un\G	Index register	Constant		Pointer		Level		
		Bit	Word	R	Bit	Word		Z	K	H	P	I	N		
CALL(P) P**	P										○				2
RET															1

The number of basic steps of CALL (P) is 2 tn, and that of the RET instruction is one. ("n" is an argument passed to the subroutine.)

Project name	QEX5
Program name	MAIN

Ladder example

Create the following ladder with GX Works2 and write it to the CPU of the demonstration machine. Then check that the CALL and RET instructions operate properly.



Operating Procedure

The following procedures are the same as the Operating Procedure in section 4.4.

(1) Creating a new project

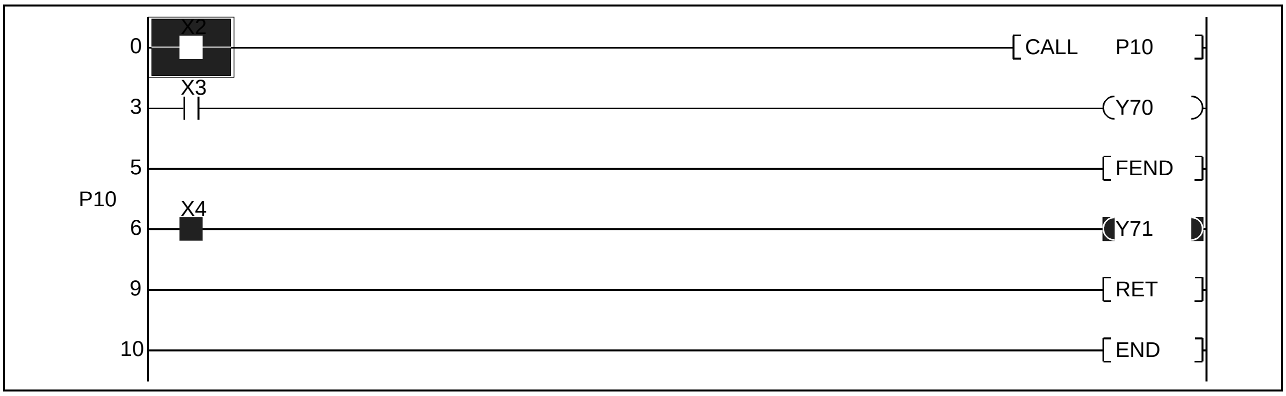
(2) Creating a program

(3) Writing the project to the programmable controller

(4) Monitoring the ladder

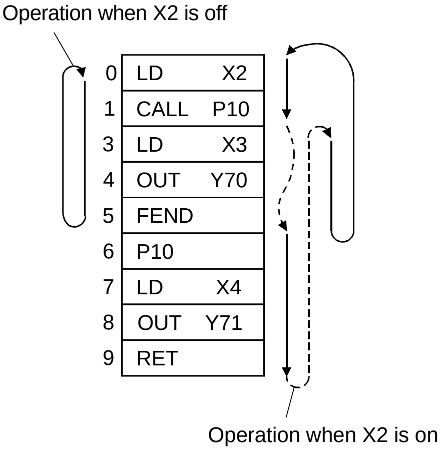
Operation Practice

Verify the operation of the ladder, which was created with GX Works2 and written to the CPU of the demonstration machine, by monitoring the ladder on the screen.



(1) When X2 is off
- (a) The operation is executed from 0 to FEND.
- (b) Turning on or off X3 turns on or off Y70.
- (c) Turning on or off X4 does not change Y71.

(2) When X2 is on
- (a) After the subroutine of P10 is executed, the operation from step 3 to FEND is executed.
- (b) Turning on or off X3 turns on or off Y70.
- (c) Turning on or off X4 turns on or off Y71.



Operation when X2 is off

0	LD	X2
1	CALL	P10
3	LD	X3
4	OUT	Y70
5	FEND	
6	P10	
7	LD	X4
8	OUT	Y71
9	RET	

Operation when X2 is on

Related Exercise —— Exercise 4

Project name	QTEST1
Program name	MAIN

4.8 Exercise

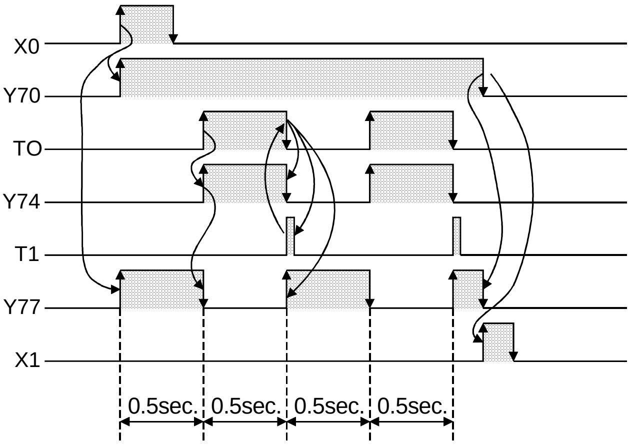
4.8.1 Exercise 1

LD to NOP

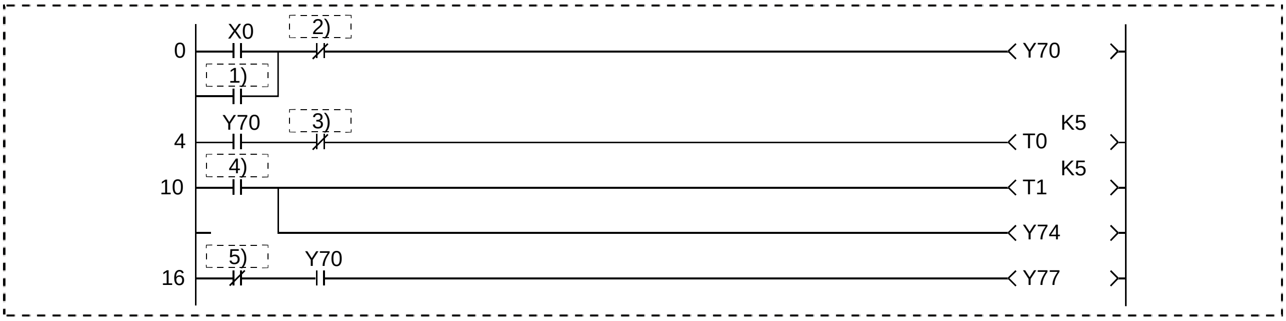
When X0 turns on, Y70 is self-maintained, and Y74 and Y77 flicker alternately every 0.5sec.

When X1 turns on, Y70 turns off and flickering of Y74 and Y77 also stops.

[Timing chart]



Create the following program with GX Works2 filling in the blanks ⬚. Then, check the operation using the demonstration machine.



Project name	QTEST2
Program name	MAIN

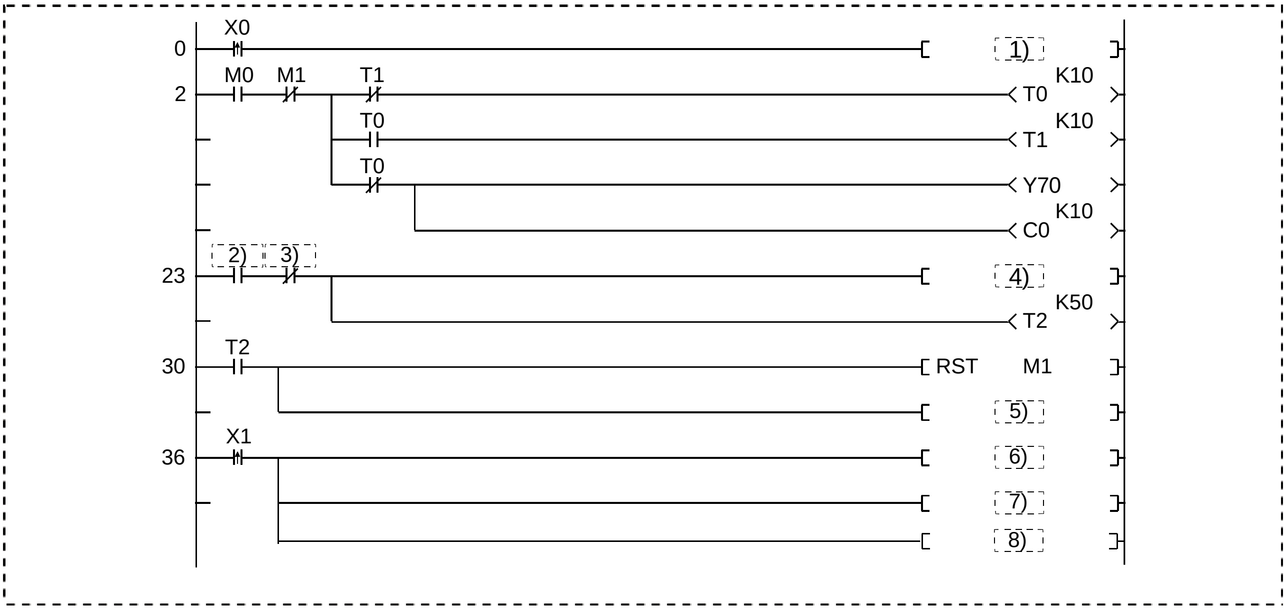
4.8.2 Exercise 2

SET, RST

When X0 is turned on, Y70 starts to flicker at one-second intervals and stops the flickering for five seconds after flickering 10 times, then restarts flickering.

The flickering of Y70 can be stopped by turning on X1.

Create the following program with GX Works2 filling in the blanks □. Then, check the operation using the demonstration machine.



1) ____________________

2) ____________________

3) ____________________

4) ____________________

5) ____________________

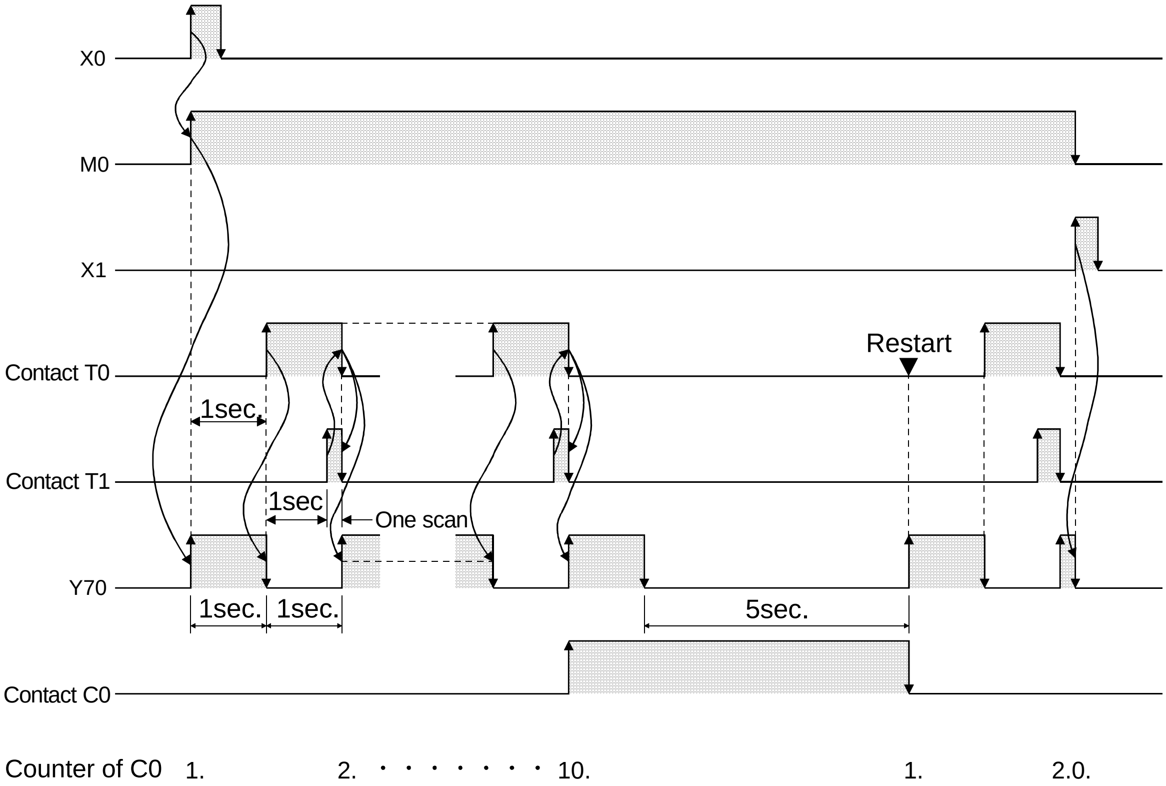
6) ____________________

7) ____________________

8) ____________________

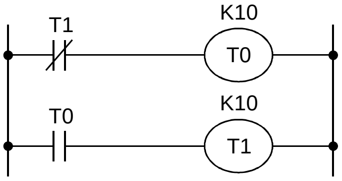
Hint

(1) The following shows the timing chart of the program.

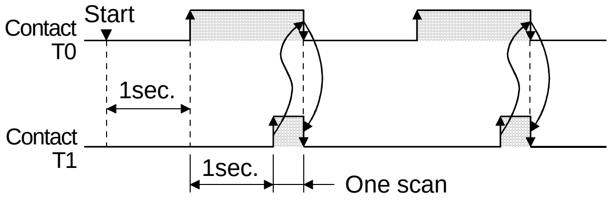


(2) The following shows the basic flickering ladder and its timing chart.

[Ladder]

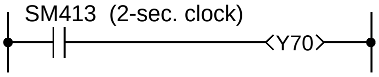


[Timing chart]

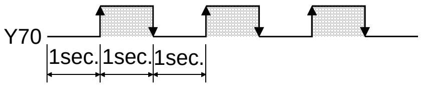


REFERENCE

- The flickering ladder can be created using the special relay that generates clock as shown below.



[Timing chart]



In addition to the SM413 (2-sec. clock) on the left, the following can be used.

SM409 (0.01-sec. clock)
SM410 (0.1-sec. clock)
SM411 (0.2-sec. clock)
SM412 (1-sec. clock)
SM414 (2n-sec. clock)
SM415 (2n-msec. clock)

The clock starts from OFF when the programmable controller is reset or the power is turned on.

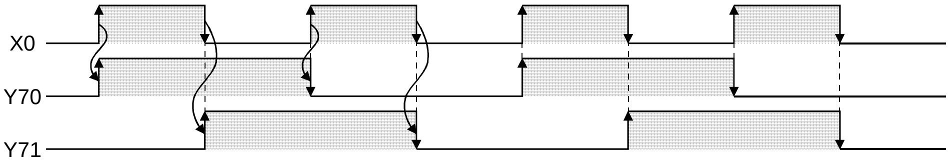
Project name	QTEST3
Program name	MAIN

4.8.3 Exercise 3

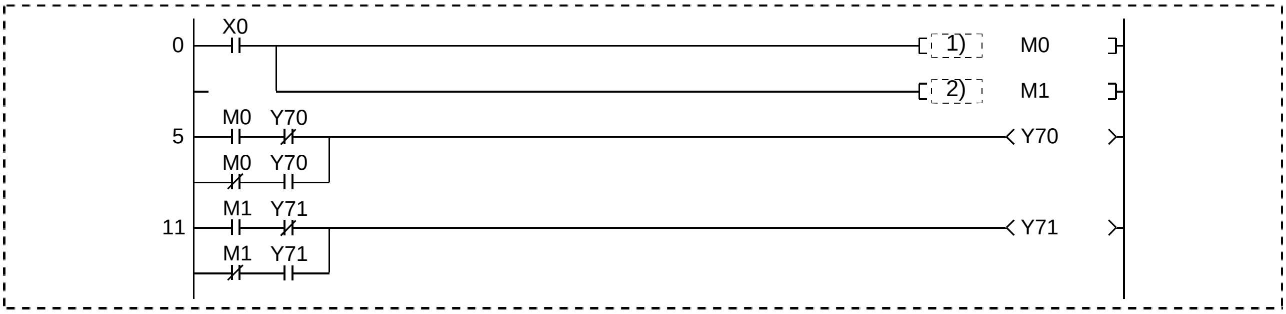
PLS, PLF

Y70 starts to switch between ON and OFF alternately when X0 is turned on, and turning off X0 triggers Y71 to operate in the same way as Y70 does.

[Timing chart]



Create the following program with GX Works2 filling in the blanks ⬚. Then, check the operation using the demonstration machine.



1) ______________________________

2) ______________________________

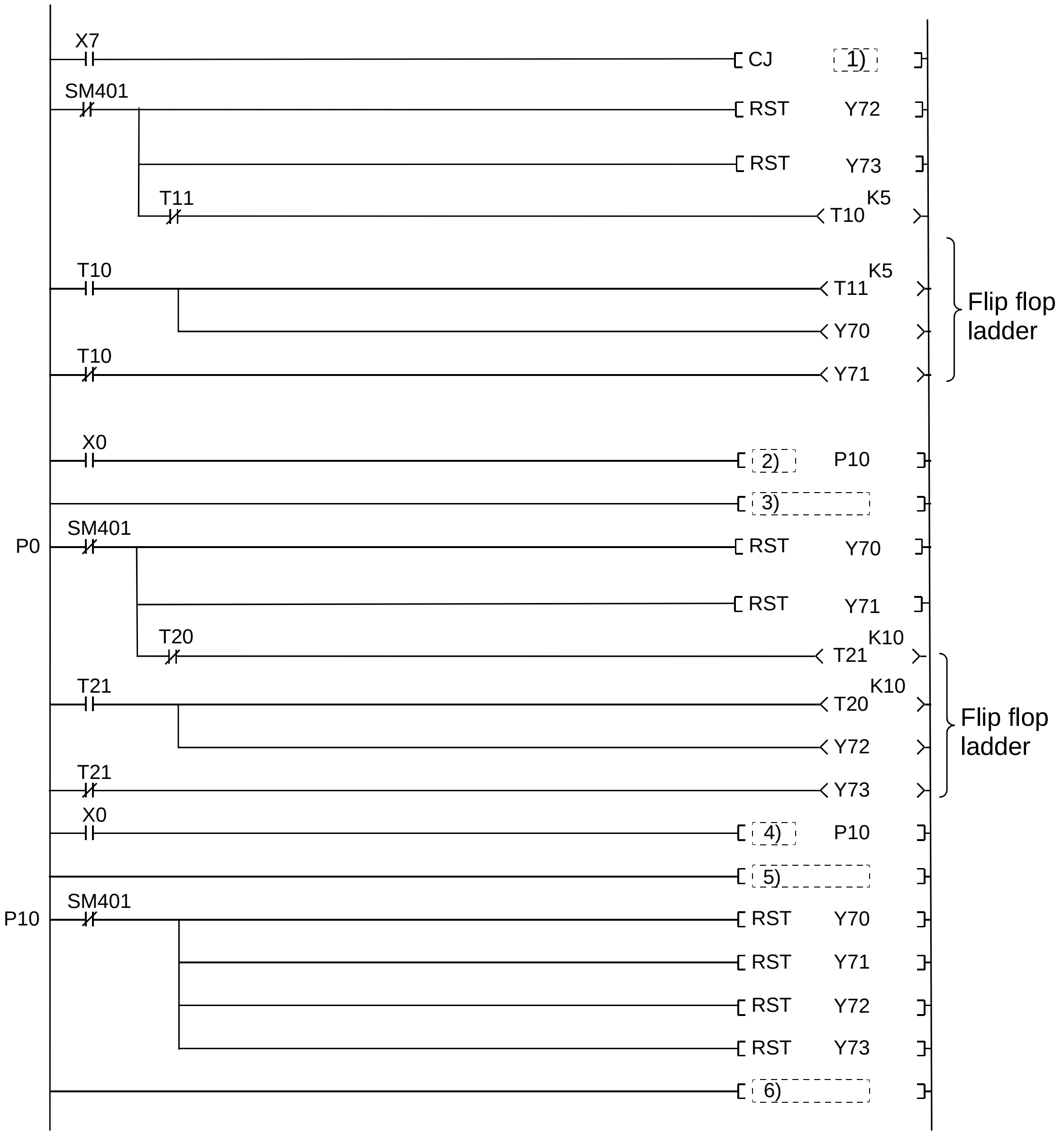
Project name	QTEST4
Program name	MAIN

4.8.4 Exercise 4

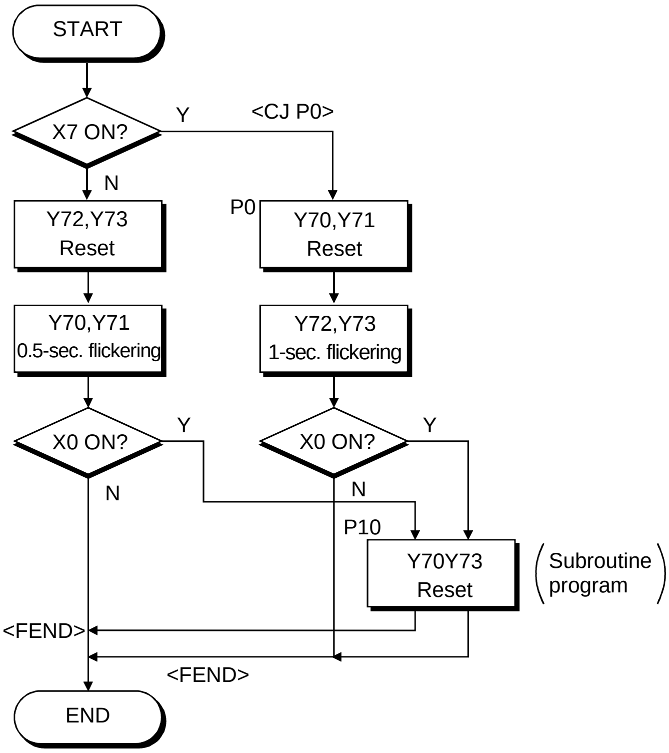
CJ, CALL, RET, FEND

Y70 and Y71 flicker for 0.5sec. alternately when X7 is off, and when X7 is on, Y72 and Y73 flicker for 1.0sec. alternately. Turning on X0 resets the currently flickering Y70 to Y73.

Fill in the blanks ▭. Then, check the operation using the demonstration machine.



Hint



START
X7 ON?
Y
<CJ P0>
N
Y72,Y73 Reset
P0
Y70,Y71 Reset
Y70,Y71 0.5-sec. flickering
Y72,Y73 1-sec. flickering
X0 ON?
Y
X0 ON?
Y
N
N
P10
Y70Y73 Reset
Subroutine program
<FEND>
<FEND>
END

1) ____________________

2) ____________________

3) ____________________

4) ____________________

5) ____________________

6) ____________________

Answers for the exercises in Chapter 4

Exercise No.		Answer
1	1)	Y70
	2)	X1
	3)	T1
	4)	T0
	5)	Y74
2	1)	SET M0
	2)	C0
	3)	Y70
	4)	SET M1
	5)	RST C0
	6)	RST M0
	7)	RST C0
	8)	RST M1
3	1)	PLS
	2)	PLF
4	1)	P0
	2)	CALL
	3)	FEND
	4)	CALL
	5)	FEND
	6)	RET

MEMO

CHAPTER 5 BASIC INSTRUCTION -PART 2-

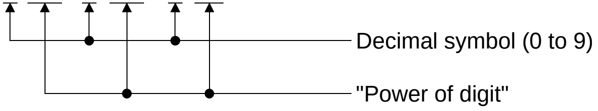
5.1 Notation of Values (Data)

The programmable controller CPU converts all input signals into ON or OFF signals (logical 1 or 0, respectively) to store and process them. Therefore, the programmable controller executes the numeric operation using the numeric values stored with the logical 1 or 0 (binary numbers = BIN).
In daily life, a decimal number is regarded as the most commonly and the easiest system. Therefore, the decimal-to-binary conversion or the reverse conversion is required when values are written or read (monitored) to or from the programmable controller. The programming system and some instructions have the function for the decimal-to-binary conversion and the binary-to-decimal conversion.
This section explains how to express values (data) in decimal, binary, hexadecimal and binary-coded decimal notation (BCD), and how to convert them.

Decimal

- A decimal number system 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 is reset to 0 and the next increment of the next digit (to the left) is incremented.
- The following shows how a decimal number (in this case 153) is represented.

$$
\begin{aligned}
153 &= 100+50+3 \\
&= 1\times100+5\times10+3\times1 \\
&= \underline{1}\times\underline{10}^2+\underline{5}\times\underline{10}^1+\underline{3}\times\underline{10}^0
\end{aligned}
$$



n : Digit number (0, 1, 2...)
10 : Decimal

- In the MELSEC-Q series programmable controller, symbol "K" is used for expressing values in decimal.

Binary (BIN)

- The binary number system consists of two symbols: 0 and 1 which represent the order and size (amount). After a digit reaches 1, an increment is reset to 0 and the next digit (to the left) is incremented. The two digits 0 and 1 are called bits.

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

- The following example explains how to convert a binary number into a decimal number.

  "10011101"

  The diagram below shows the binary number with the powers of two.

7	6	5	4	3	2	1	0	← Bit number
1	0	0	1	1	1	0	1	← Binary
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	← Base number raised to the power of digit } = Bit weights
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
128	64	32	16	8	4	2	1	← ("Binary")

  The binary number is broken as follows.

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

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

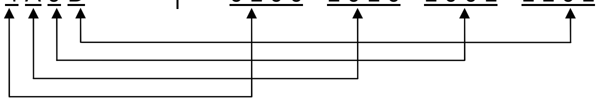
  $= 157$

  The binary number can be calculated by adding each weight of bits whose codes are 1.

Hexadecimal

- The hexadecimal number system 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 is reset to 0 and the next digit (to the left) is incremented.

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



3	2	1	0	←Digit number
4	A	9	D	←Hexadecimal

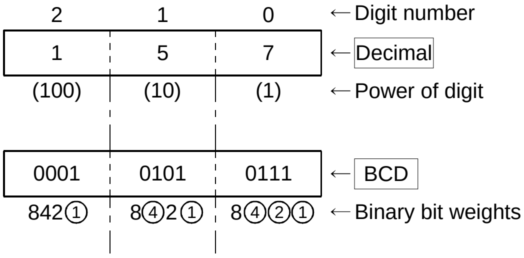
$$= (4) \times \underline{16^3} + (A) \times \underline{16^2} + (9) \times \underline{16^1} + (D) \times \underline{16^0}$$
$$= 4 \times 4096 + 10 \times 256 + 9 \times 16 + 13 \times 1$$
$$= 19101$$

"Power of digit"
n: Digit number
16: Hexadecimal

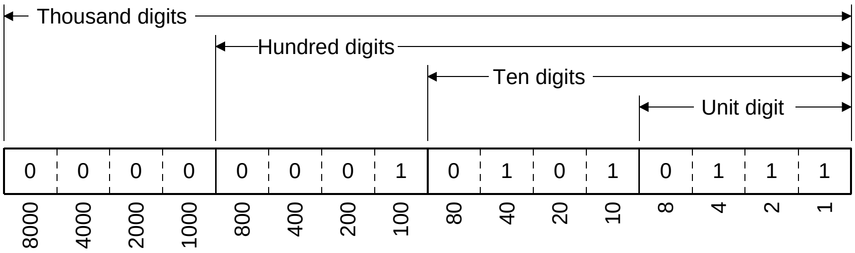
- Four bits of a binary number equal to one digit of a hexadecimal number.
- In the MELSEC-Q series programmable controller, symbol "H" is used for indicating a hexadecimal number.
- The hexadecimal system is used to represent the following device numbers.
  - • Input and output (X, Y)
  - • Function 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)

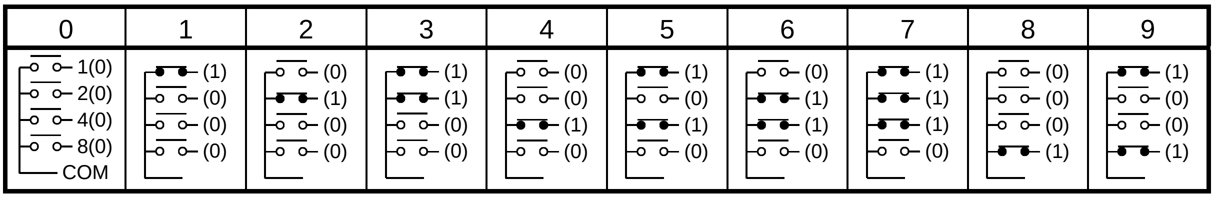
- The binary-coded decimal is "a numerical system using a binary number to represent a decimal number".
  A decimal number 157, for example, is expressed as shown below.



- In BCD, decimal numbers 0 to 9999 (the biggest 4-digit number) can be represented by 16 bits.
  The diagram below shows the bit weights of BCD.



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



0	1	2	3	4	5	6	7	8	9

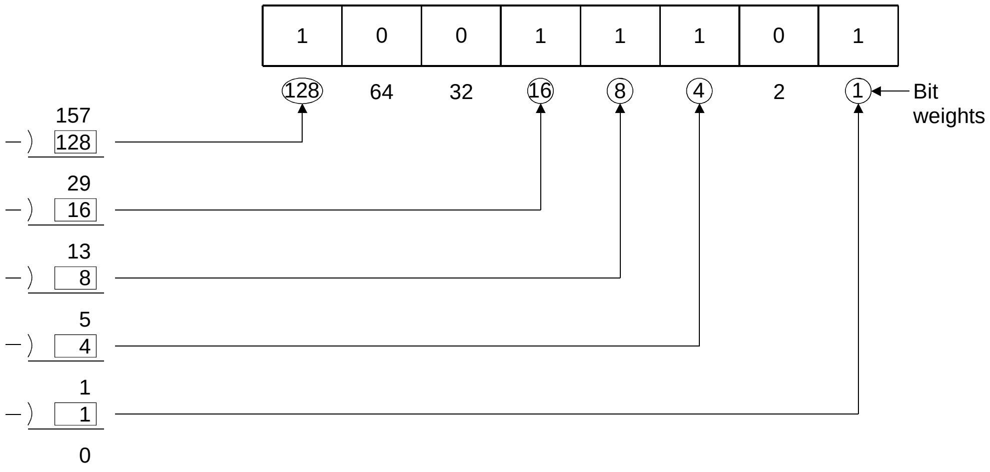
1(0)
2(0)
4(0)
8(0)
COM
	(1)
	(0)
	(0)
	(0)
		(0)
		(1)
		(0)
		(0)
			(1)
			(1)
			(0)
			(0)
				(0)
				(0)
				(1)
				(0)
					(1)
					(0)
					(1)
					(0)
						(0)
						(1)
						(1)
						(0)
							(1)
							(1)
							(1)
							(0)
								(0)
								(0)
								(0)
								(1)
									(1)
									(0)
									(0)
									(1)

BCD code digital switch

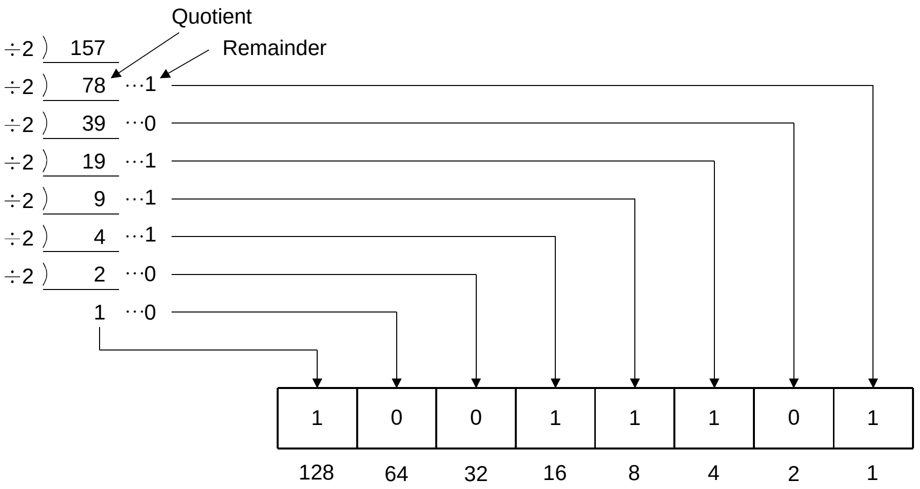
How to convert the decimal number into the binary number

In the example below, a decimal number 157 is converted into the binary number.

1)



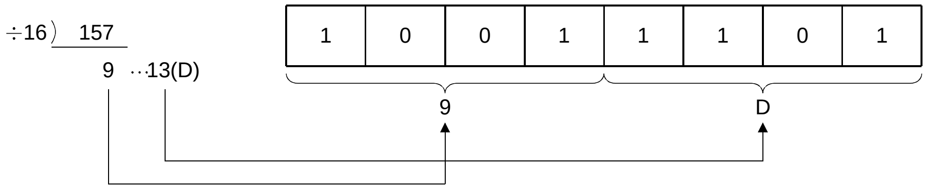
2)



How to convert the decimal number into the hexadecimal number

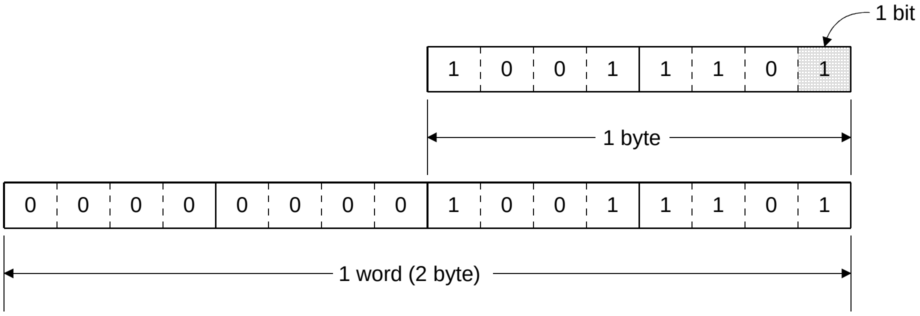
In the example below, a decimal number 157 is converted into the hexadecimal number.

1)

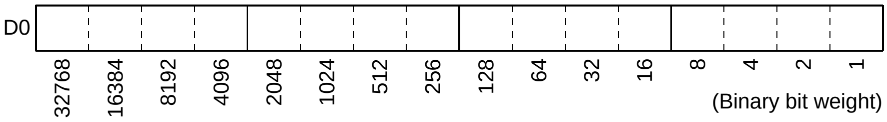


Numerical values used by MELSEC-Q series programmable controller

- Usually, 8 bits are called 1 byte, and 16 bits (2 bytes) are called 1 word.

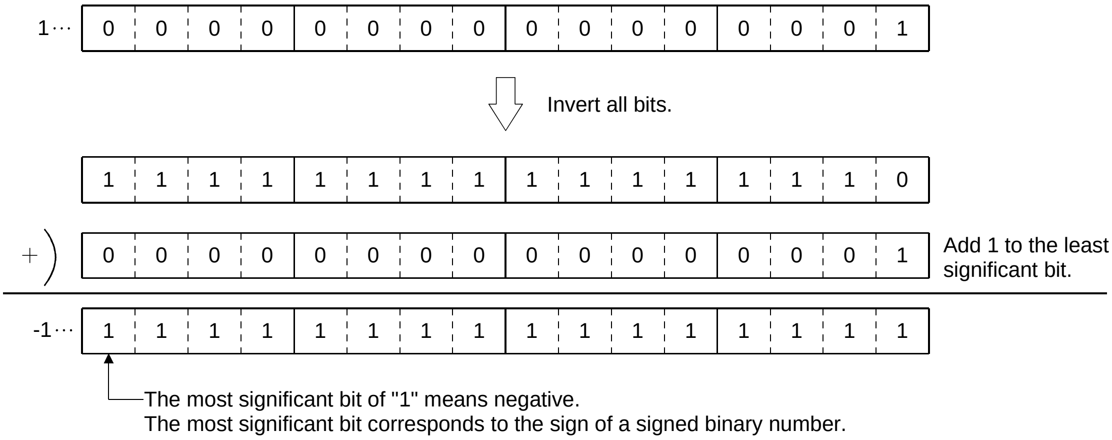


- Registers of each word device in the MELSEC-Q series programmable controller consist of 16 bits.
  - Data register D
  - Current value of timer T
  - Current value of counter C
  - File register R
  - Link register W



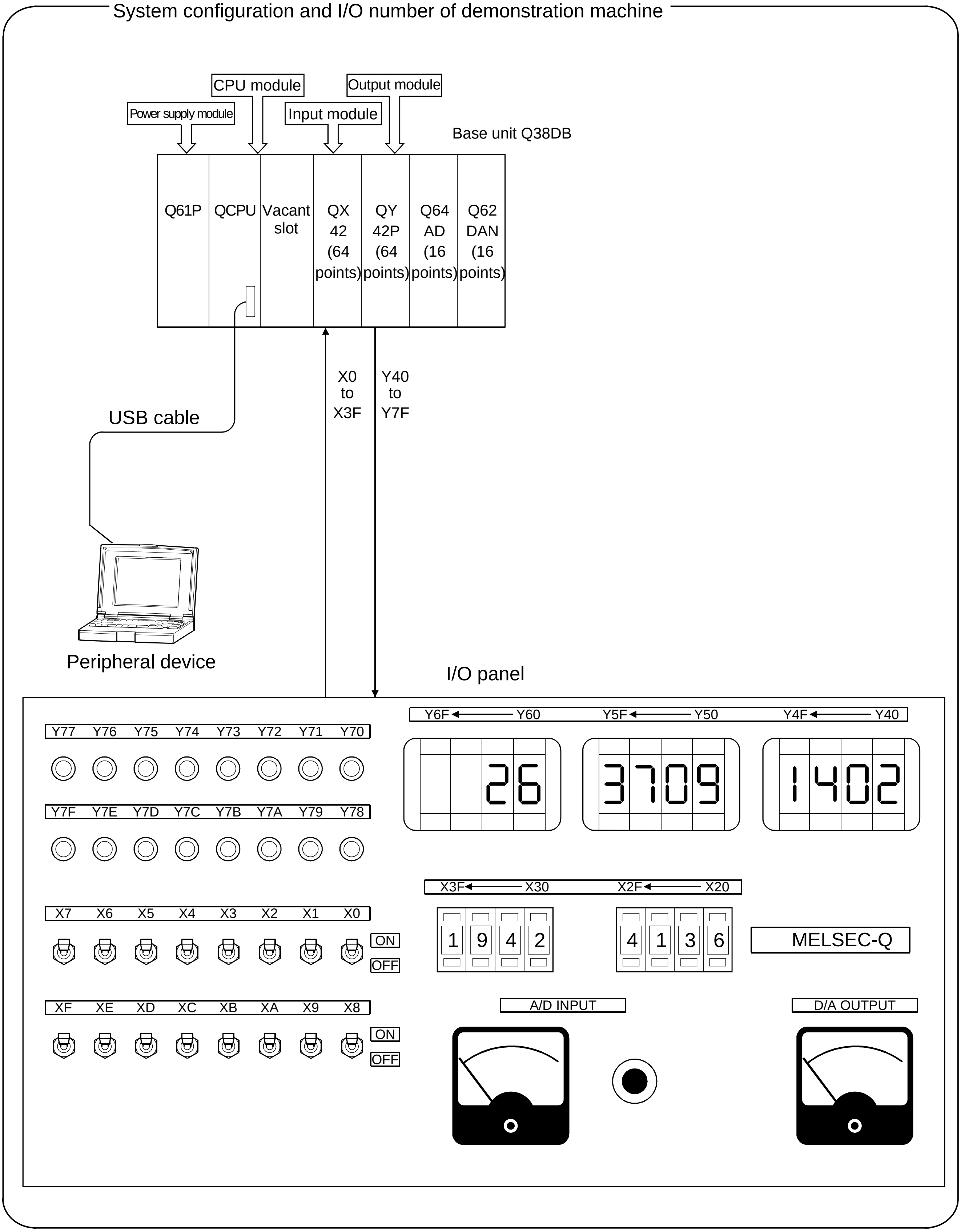
- The following two ranges of numbers can be processed in 16 bits (1 word).
  1) 0 to 65535
  2) -32768 to +32767

- The range 2) is available for the MELSEC-Q programmable controller.
  The negative numbers adopt two's complement against the positive numbers (1 to +32767).
- In the two's complement, each binary bit is inverted, and 1 is added to the least significant bit.
  Example)
  How to calculate the two's complement against 1


1···
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
Invert all bits.
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0
+)
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
Add 1 to the least significant bit.
-1···
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
The most significant bit of "1" means negative.
The most significant bit corresponds to the sign of a signed binary number.

BCD (binary coded decimal)	BIN (binary)	K (decimal)	H (hexadecimal)
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

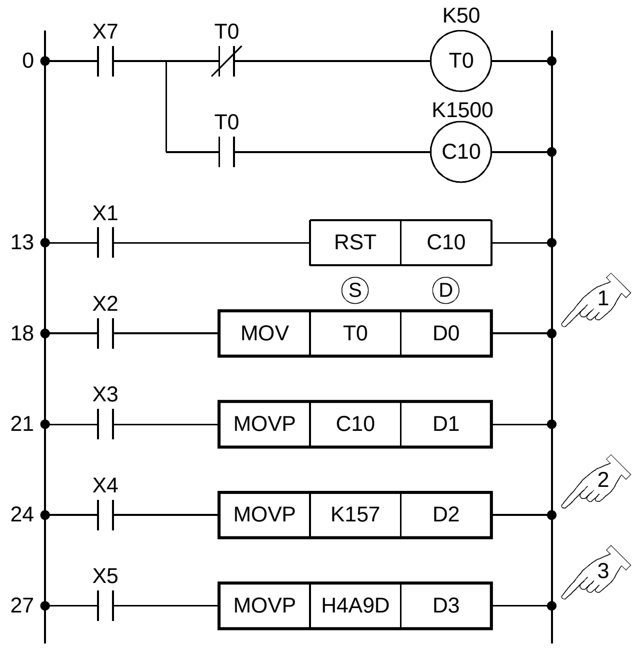
System configuration and I/O number of demonstration machine



5.2 Transfer Instruction

Project name	QB-11
Program name	MAIN

5.2.1 MOV (P) 16-bit data transfer



☞1
- When the input condition turns on, the current value of the timer T0 is transferred to the data register D0.
  Ⓢ... Source, Ⓓ... Destination
- The current value of T0 is stored in the register in binary (BIN code). And the value is transferred to the data register D0 in binary (The code is not converted at the transfer.)

T0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	1	45
									128	64	㉜	16	⑧	④	2	①	

⇩

D0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	1
									128	64	㉜	16	⑧	④	2	①

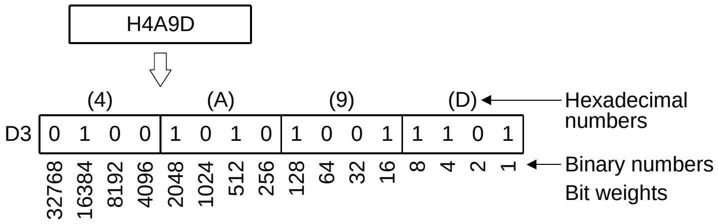
☞2
- When the input condition turns on, the decimal number 157 is transferred to the data register D2. And the value is stored in the register in binary. The decimal number (K) is converted into binary automatically, then transferred.

K157

⇩

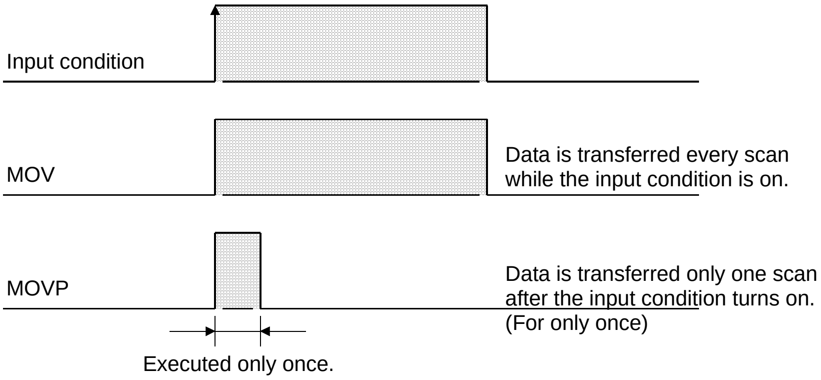
D2	0	0	0	0	0	0	0	0	1	0	0	1	1	1	0	1
									$\textcircled{128}$	64	32	⑯	⑧	④	2	①

3 ● When the input condition turns on, the hexadecimal number 4A9D is transferred to the data register D3.

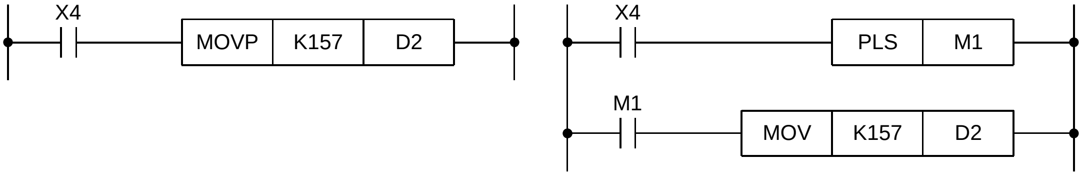


Difference between MOV and MOVP

The P of MOVP stands for a pulse.


Input condition
MOV
Data is transferred every scan while the input condition is on.
MOVP
Data is transferred only one scan after the input condition turns on. (For only once)
Executed only once.

- Use the MOV instruction when reading the changing data for all the time. Use the MOVP instruction to transfer data instantaneously such as setting data or reading data at an error.
- Both of the following ladder programs function similarly.



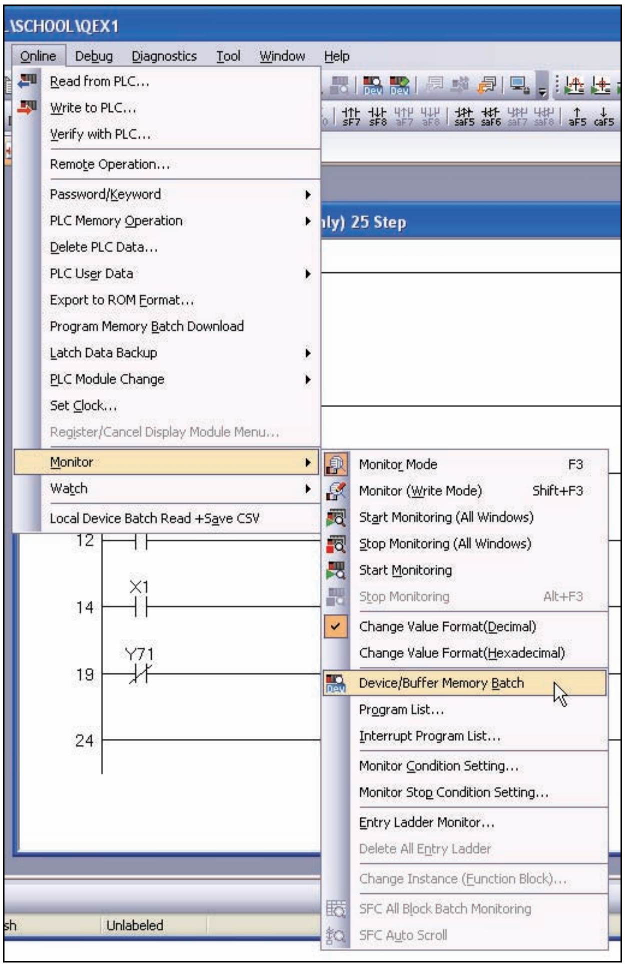
		Applicable device												Digit	Number of basic steps
		Internal device (system or user)		File register	MELSECNET/10 (H) Direct Jn\		Intelligent function module Un\G	Index register	Constant		Pointer		Level		
		Bit	Word	R	Bit	Word		Z	K	H	P	I	N		
MOV Ⓢ Ⓓ	S	○	○	○	○	○	○	○	○	○					*
	D	○	○	○	○	○	○	○							

*: The number of steps varies depending on the device to be used.

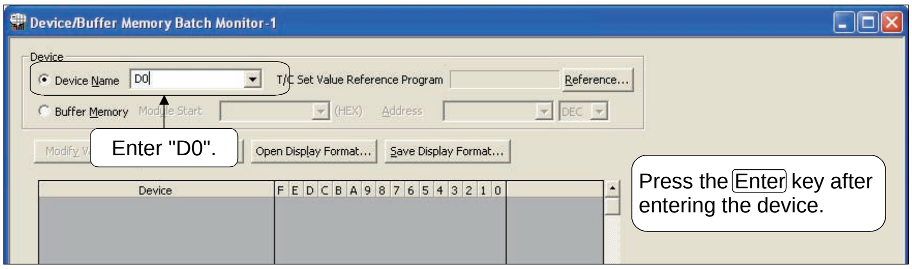
Check

The CPU is running.
The inputs X2, X3, X4, X5, and X7 are on.

- Monitor the contents of the data registers D0 to D3.
  - After writing the data to the programmable controller, click [Online] → [Monitor] → [Device/Buffer Memory Batch].
    The Device/Buffer Memory Batch Monitor dialog box is displayed.



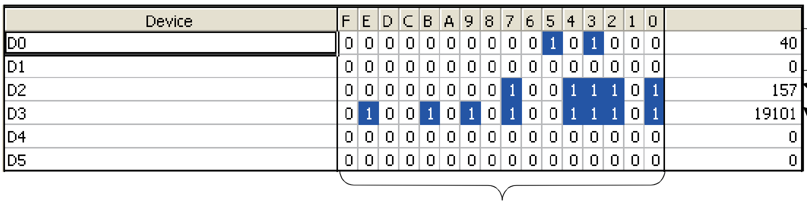
  - Enter "D0" in the Device Name column of the Device/Buffer Memory Batch Monitor dialog box and press the Enter key.



○ Device Name: D0

Enter "D0".

Press the Enter key after entering the device.



Device	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	
D0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	40
D1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D2	0	0	0	0	0	0	0	0	1	0	0	1	1	1	0	1	157
D3	0	1	0	0	1	0	1	0	1	0	0	1	1	1	0	1	19101
D4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Current values of a timer and counter are monitored. (They are changing.)

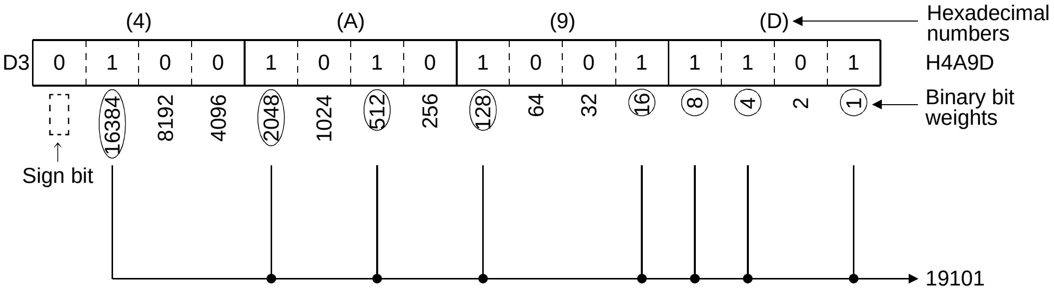
Indicates that a decimal number 157 (K157) is stored.

This is a decimal number equivalent to a hexadecimal 4A9D.

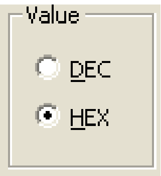
Indicates the word devices in the on/off of the bit units.

0 : OFF (0 in binary)

1 : ON (1 in binary)



- Click the Display Format button.
- Change the display of the numerical value in the monitor to the hexadecimal notation.
- Select "HEX" for the device batch monitoring.

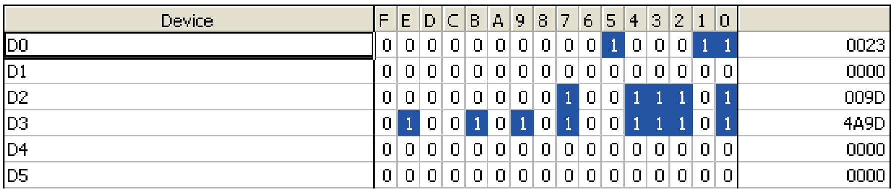


Value

○ DEC

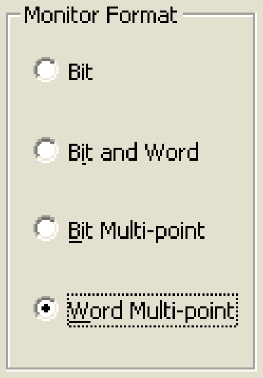
● HEX

[Device/Buffer Memory Batch Monitor screen]



Device	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	
D0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0023
D1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000
D2	0	0	0	0	0	0	0	0	1	0	0	1	1	1	0	1	009D
D3	0	1	0	0	1	0	1	0	1	0	0	1	1	1	0	1	4A9D
D4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000
D5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000

- Change the display of the numerical value in the monitor to the binary notation.
  Select "Word Multi-point" in Monitor Format for the device batch monitoring.



Monitor Format

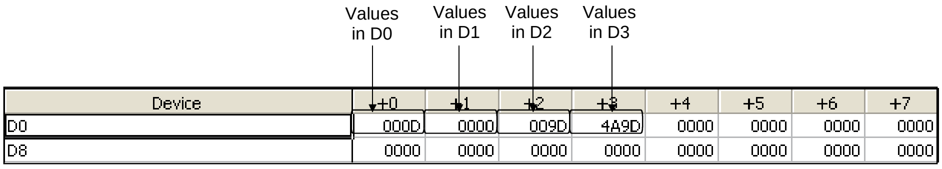
○ Bit

○ Bit and Word

○ Bit Multi-point

● Word Multi-point

[Device/Buffer Memory Batch Monitor screen]

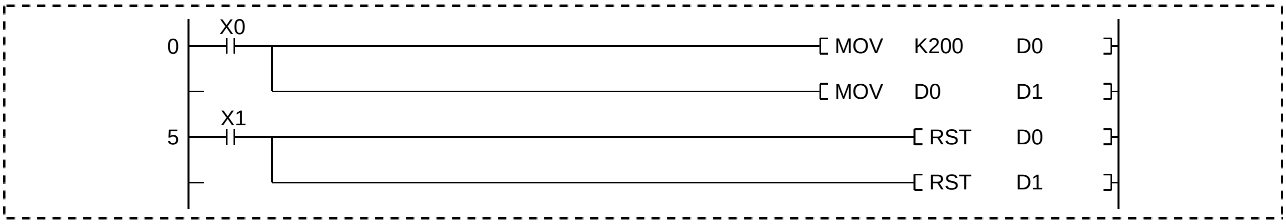


Device	+0	+1	+2	+3	+4	+5	+6	+7
D0	000D	0000	009D	4A9D	0000	0000	0000	0000
D8	0000	0000	0000	0000	0000	0000	0000	0000

Project name	QEX7
Program name	MAIN

Ladder example

Create the following ladder with GX Works2 and write it to the CPU of the demonstration machine. Then check that the MOV instruction works properly.



Operating Procedure

The following procedures are the same as the Operating Procedure in section 4.4.

(1) Creating a new project

(2) Creating a program

(3) Writing the project to the programmable controller

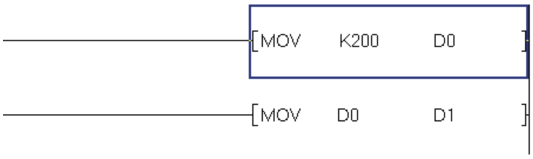
(4) Monitoring the ladder

- How to modify the transfer instruction

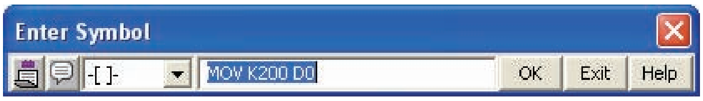
To modify the transfer instruction, follow the procedures below.

Example: Change the transfer data K200 of [MOV K200 D0] to K100

1) Double-click the instruction to be modified.



2) The Enter Symbol window is displayed.

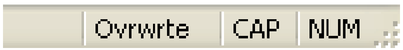


MOV K200 D0

3) Write "1" over "2" of "MOV K200 D0".

4) Click the OK button on the Enter Symbol window.

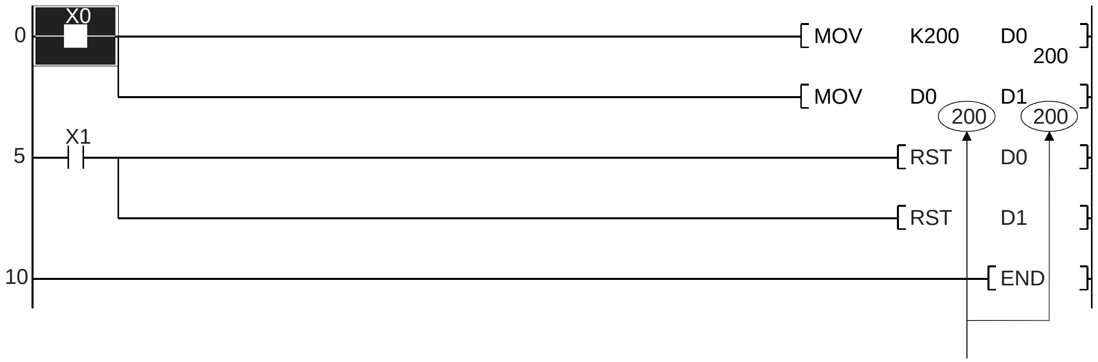
All data in -[ ]- can be modified with the above method.
When "Insert" is displayed, press the Insert key to change it to "Ovrwrte" before modifying.



5) After finishing modifications, click [Compile] → [Build].

Operation Practice

Check that "200" is displayed under both D0 and D1 on the monitor screen when X0 on the control panel of the demonstration machine is turned on.



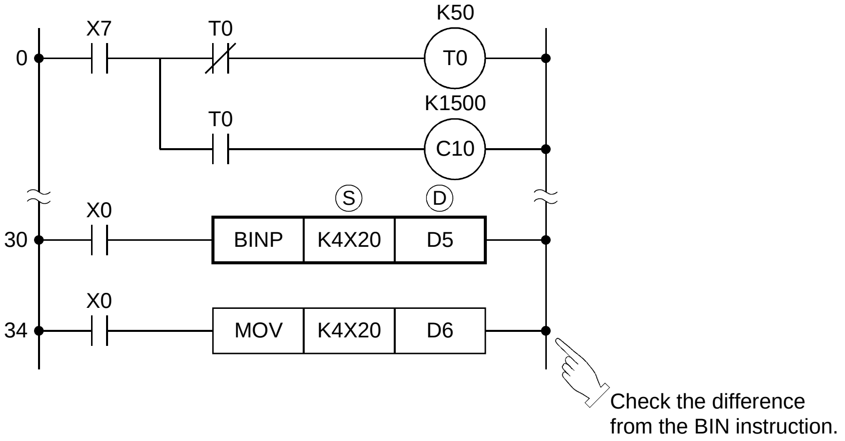
When X0 turns on, the current values of D0 and D1 become 200.

Related Exercise ---- Exercise 5

Project name	QB-12
Program name	MAIN

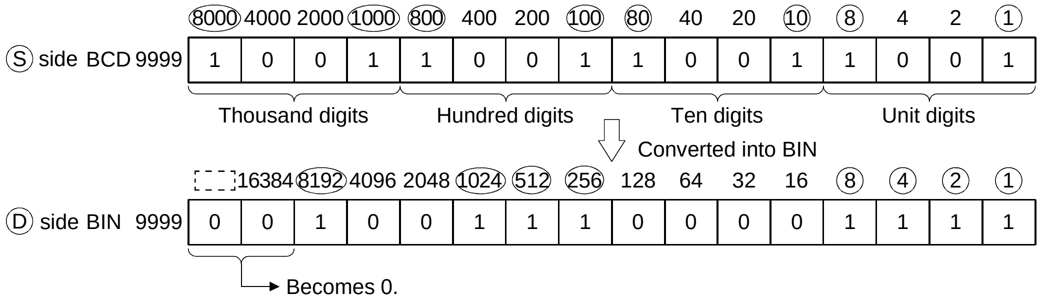
5.2.2 BIN (P) BCD → BIN data conversion instruction

Operations to read and write data after step 35



Check the difference from the BIN instruction.

- When the input condition is turned on, the data in the device specified in Ⓢ is recognized as a BCD code, converted into binary (BIN code), and transferred to the device specified in Ⓓ.



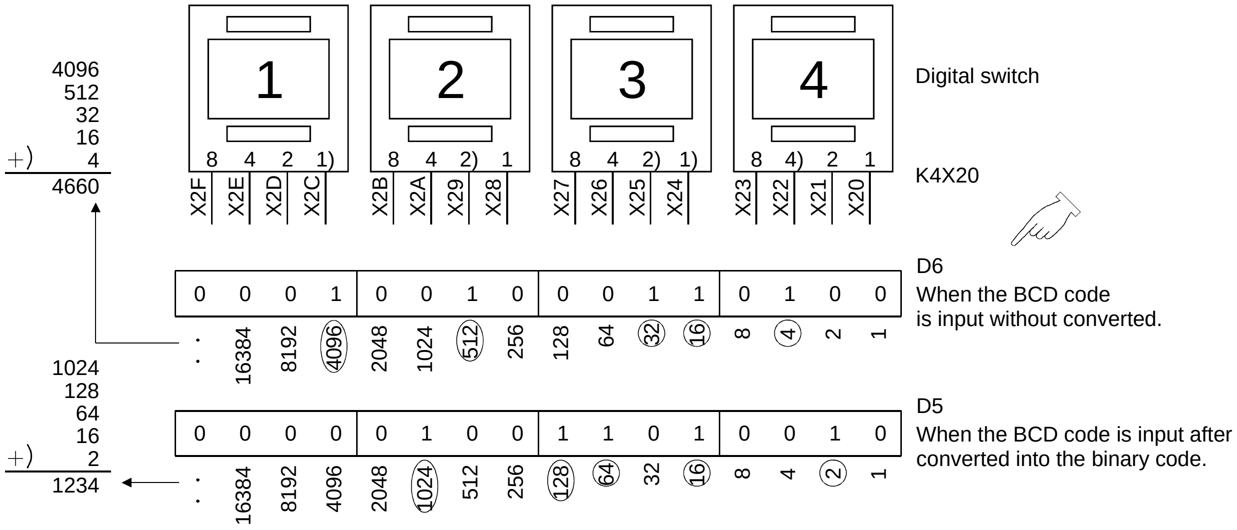
	8000	4000	2000	1000	800	400	200	100	80	40	20	10	8	4	2	1
Ⓢ side BCD 9999	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1
	Thousand digits				Hundred digits				Ten digits				Unit digits			

Converted into BIN

		16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1
Ⓓ side BIN 9999	0	0	1	0	0	1	1	1	0	0	0	0	1	1	1	1

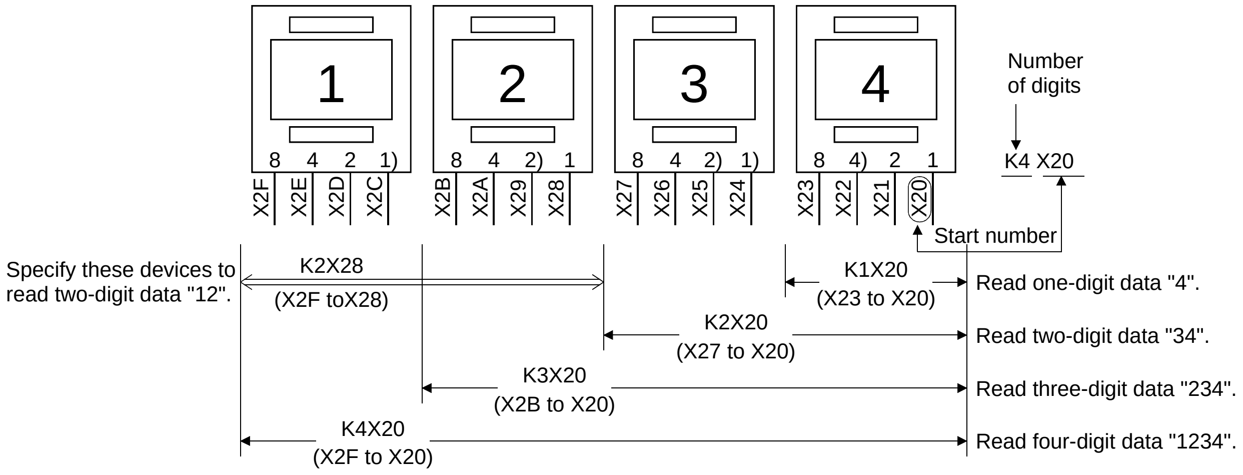
Becomes 0.

- The ordinary digital switches generate BCD codes. Therefore, the BIN instruction is required for writing data from the digital switches to the programmable controller.



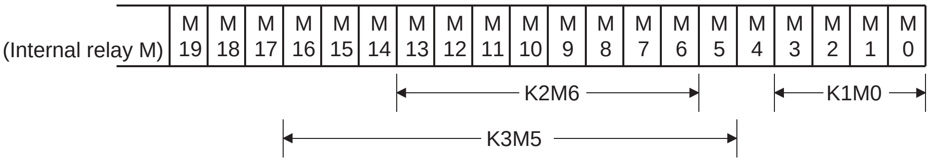
K4X20

- Word devices D (data register), T (timer current value), and C (counter current value) consist of 16 bits (1 word), and data is basically transferred among the units of one device.
- Collecting 16 bit devices (such as X, Y, and M) means processing the word device. The device numbers allocated to the bit devices must be in consecutive order.
- In the bit device, data are processed in units of four.



As long as four bit devices are in consecutive order, any bit device can be specified as the first.

- Other bit devices can be processed in the same way as described above.



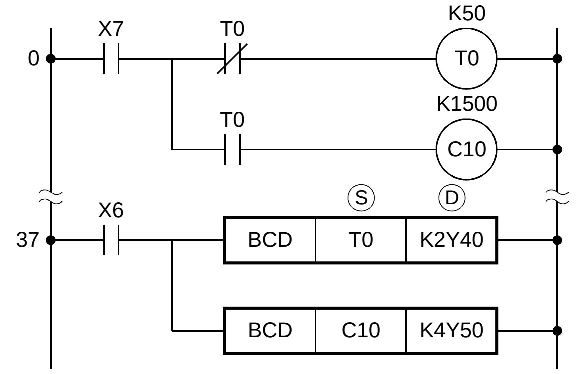
(Internal relay M)	M 19	M 18	M 17	M 16	M 15	M 14	M 13	M 12	M 11	M 10	M 9	M 8	M 7	M 6	M 5	M 4	M 3	M 2	M 1	M 0

* A sample program using a digital switch to import data is provided in page App. - 46.

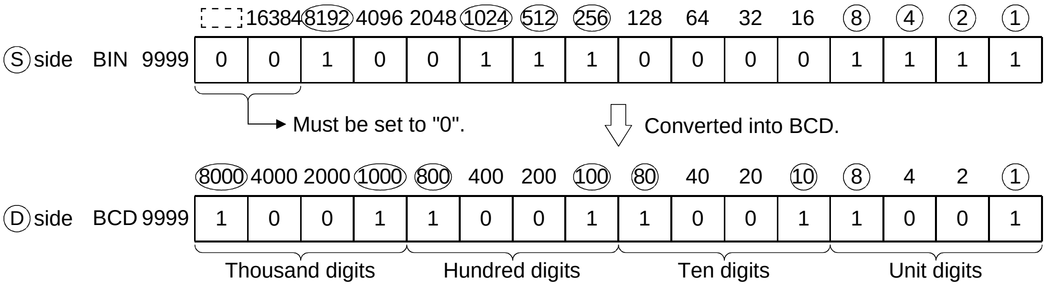
		Applicable device												Digit	Number of basic steps
		Internal device (system or user)		File register	MELSECNET/ 10 (H) Direct Jn\		Intelligent function module Un\G	Index register	Constant		Pointer		Level		
		Bit	Word	R	Bit	Word		Z	K	H	P	I	N		
MOV Ⓢ Ⓓ	Ⓢ	○	○	○	○	○	○	○						K1 to K4	3
	Ⓓ	○	○	○	○	○	○	○							

Project name	QB-13
Program name	MAIN

5.2.3 BCD (P) BIN → BCD data conversion instruction



- When the input condition is turned on, the data in the device specified in Ⓢ is recognized as a binary (BIN code), converted into a binary coded decimal (BCD code), and transferred to the device specified in Ⓓ.



			16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1
Ⓢside	BIN 9999	0	0	1	0	0	1	1	1	0	0	0	0	1	1	1	1

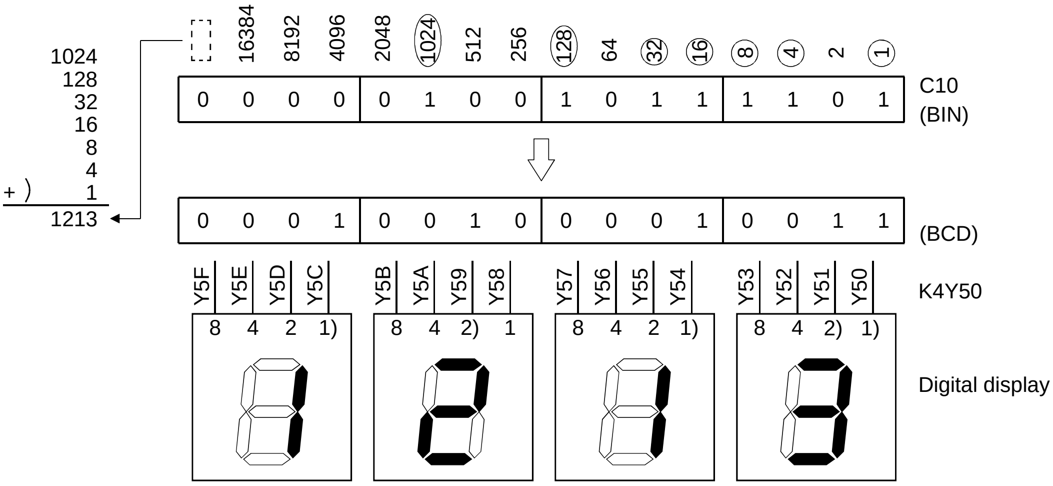
Must be set to "0".

Converted into BCD.

		8000	4000	2000	1000	800	400	200	100	80	40	20	10	8	4	2	1
Ⓓside	BCD 9999	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1

Thousand digits Hundred digits Ten digits Unit digits

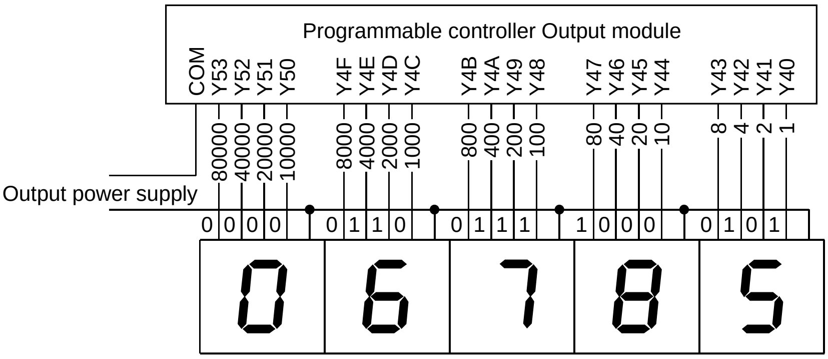
- The ordinary digital displays display numbers in the BCD code. Therefore, the BCD instruction is required for displaying data of the programmable controller (current values of timers and counters, data resister values of operation results).

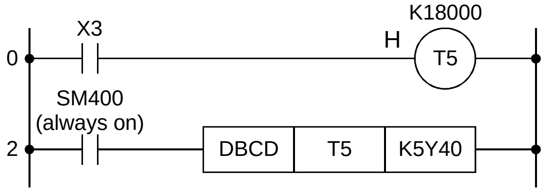


		16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1	
	0	0	0	0	0	1	0	0	1	0	1	1	1	1	0	1	C10 (BIN)
	0	0	0	1	0	0	1	0	0	0	0	1	0	0	1	1	(BCD)
	Y5F	Y5E	Y5D	Y5C	Y5B	Y5A	Y59	Y58	Y57	Y56	Y55	Y54	Y53	Y52	Y51	Y50	K4Y50
	8	4	2	1)	8	4	2)	1	8	4	2	1)	8	4	2)	1)	

Displayable Range with BCD Instruction

- The displayable range of data with the BCD instruction (to be converted from BIN into BCD) is between 0 and 9999. Any data which is outside the range causes an error.
  (Error code 4100: OPERATION ERROR)
- To display a timer current value more than 9,999, use the DBCD instruction. The instruction can handle 8-digit values (up to 99,999,999).



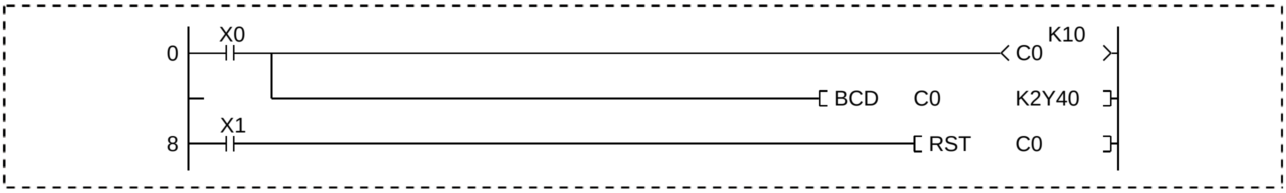


		Applicable device													Digit	Number of basic steps
		Internal device (system or user)		File register	MELSECNET/ 10 (H) Direct Jn\		Intelligent function module Un\G	Index register	Constant		Pointer		Level		
		Bit	Word	R	Bit	Word		Z	K	H	P	I	N		
BCD Ⓢ Ⓓ	Ⓢ	○	○	○	○	○	○	○						K1 to K4	3
	Ⓓ	○	○	○	○	○	○	○							

Project name	QEX8
Program name	MAIN

Ladder example

Create the following ladder with GX Works2 and write it to the CPU of the demonstration machine. Then check that the BCD instruction works properly.



Operating Procedure

The following procedures are the same as the Operating Procedure in section 4.4.

(1) Creating a new project

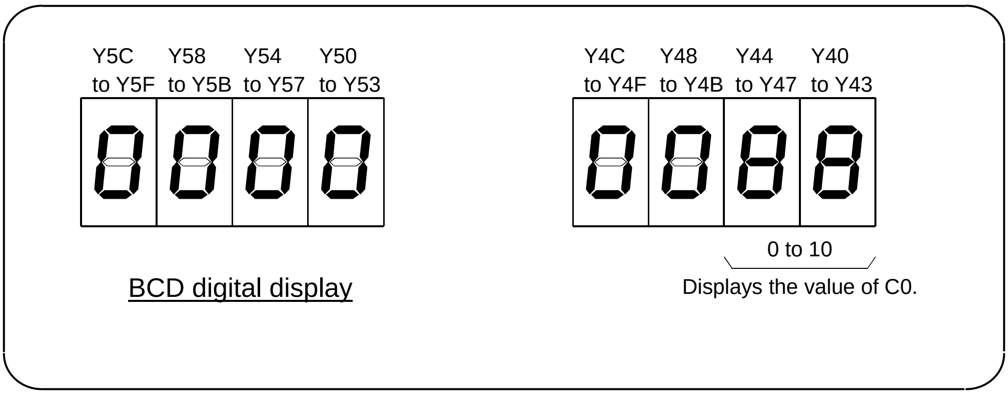
(2) Creating a program

(3) Writing the project to the programmable controller

(4) Monitoring the ladder

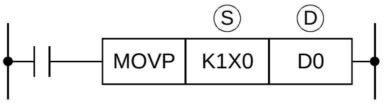
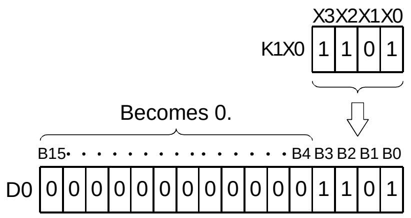
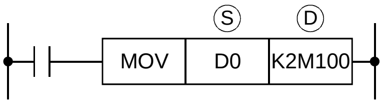
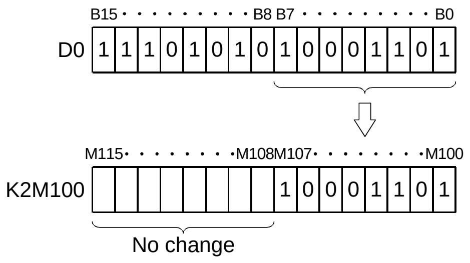
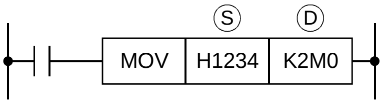
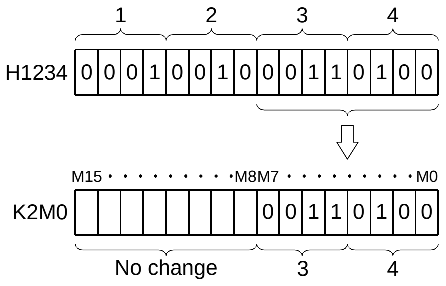
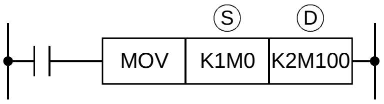
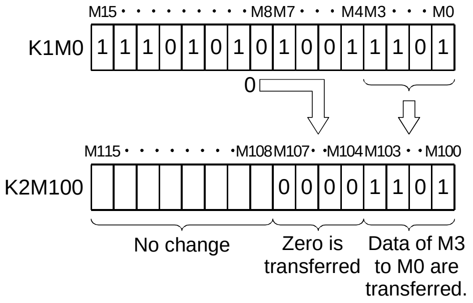
Operation Practice

Check that turning on X0 on the control panel for several times displays the value of C0 on the BCD digital displays of Y40 to Y47. Turning on X1 resets C0.



Related Exercise ---- Exercise 6

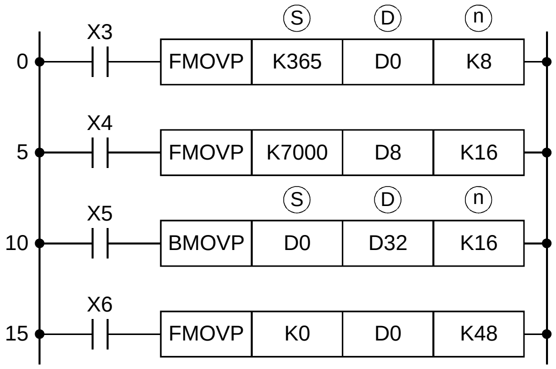
5.2.4 Example of specifying digit for bit devices and transferring data

Program example	Process
When the destination data Ⓓ is a word device  • Source: Source device • Destination: Destination device	 X3X2X1X0 K1X0 1 1 0 1 Becomes 0. B15• • • • • • • • • • • • • • • B4 B3 B2 B1 B0 D0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1
When the source data Ⓢ is a word device  	 B15 • • • • • • • • • B8 B7 • • • • • • • • • B0 D0 1 1 1 0 1 0 1 0 1 0 0 0 1 1 0 1 M115• • • • • • • •M108M107• • • • • • • •M100 K2M100 1 0 0 0 1 1 0 1 No change
When the source data Ⓢ is a constant  	 1 2 3 4 H1234 0 0 0 1 0 0 1 0 0 0 1 1 0 1 0 0 M15 • • • • • • • • •M8M7 • • • • • • • • • M0 K2M0 0 0 1 1 0 1 0 0 No change 3 4
When the source data Ⓢ is a bit device  	 M15 • • • • • • • • •M8M7 • • • M4M3 • • • M0 K1M0 1 1 1 0 1 0 1 0 1 0 0 1 1 1 0 1 0 M115• • • • • • • •M108M107• •M104M103 • • M100 K2M100 0 0 0 0 1 1 0 1 No change Zero is transferred Data of M3 to M0 are transferred.

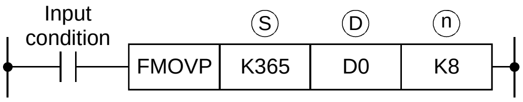
Project name	QB-14
Program name	MAIN

5.2.5 FMOV (P) FMOV (Batch transfer of the same data)

BMOV (P) BMOV (Batch transfer of the block data)



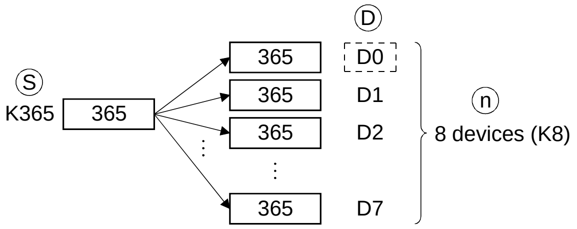
Operation Explanation



FMOV

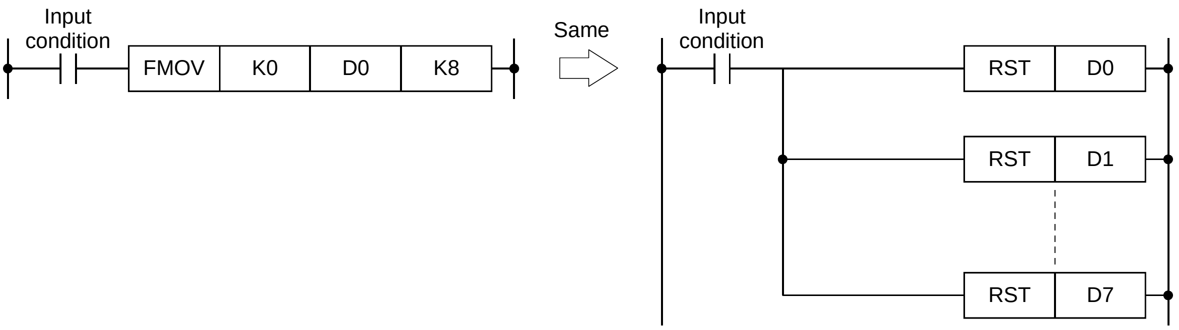
- When the input condition is turned on, the FMOV instruction transfers the data in the device specified in Ⓢ to the devices starting from the device specified in Ⓓ (the number of target devices is specified by ⓝ).

Example The FMOV instruction executes the following operation when X3 is turned on.

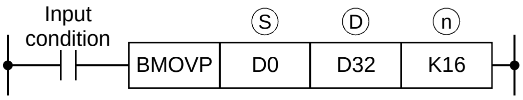


- The FMOV instruction is useful for clearing many data sets in batch.

Example



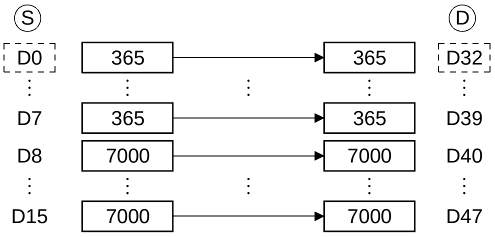
The FMOV instruction can substitute the RST instructions as shown above.



BMOV

- When the input condition is turned on, the BMOV instruction transfers the data in the devices starting from the device specified in Ⓢ to the devices starting from the device specified in Ⓓ in batch (the numbers of source devices and target devices are specified by ⓝ).

Example The BMOV instruction executes the following operation when X5 is turned on.



- The BMOV instruction is useful for the following:
  - Filing logging data
  - Saving important data (such as automatic operation data and measured data) into the latch area. This can prevent a data loss caused by a power failure.

		Applicable device												Digit	Number of basic steps
		Internal device (system or user)		File register	MELSECNET/10 (H) Direct Jn\		Intelligent function module Un\G	Index register	Constant		Pointer		Level		
		Bit	Word	R	Bit	Word		Z	K	H	P	I	N		
FMOV Ⓢ Ⓓ ⓝ BMOV Ⓢ Ⓓ ⓝ	Ⓢ	○	○	○	○	○	○	(Note) ○	(Note) ○	(Note) ○				K1 to K4	4
	Ⓓ	○	○	○	○	○	○								
	ⓝ	○	○	○	○	○	○	○	○	○					

(Note) Not available in the BMOV instruction.

Operation Practice

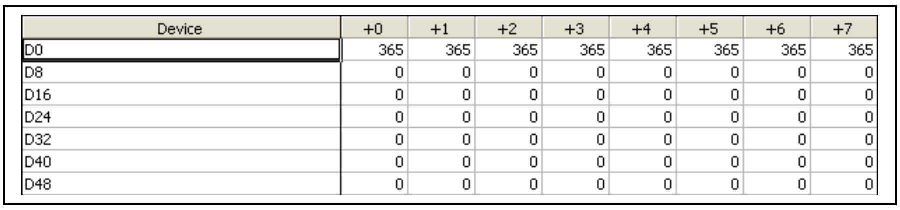
- Write the program on the previous page to the CPU, then run the CPU.
- Follow the procedures below to execute the device batch monitoring. The contents of D0 to D47 can be monitored.

  Write the program to the programmable controller ⇨ Click [Online] → [Monitor] → [Device/Buffer Memory Batch].

  Enter "D0" in the Device/Buffer Memory Batch Monitor dialog box and press the Enter key.
- Click the Display Format button and select "Word Multi-point" for Monitor Format.

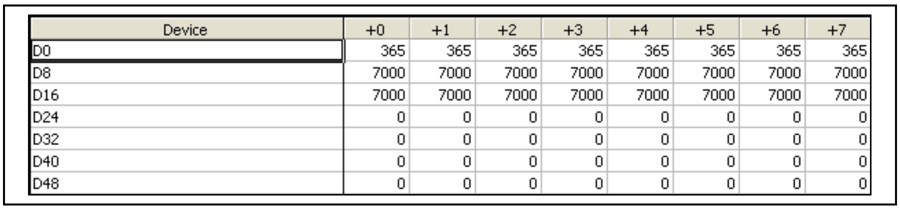
  → Click the OK button.

[Monitor screen]



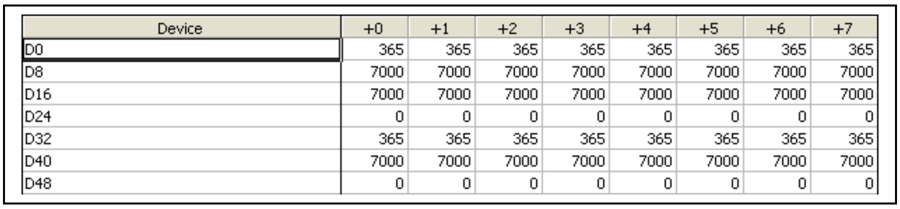
Device	+0	+1	+2	+3	+4	+5	+6	+7
D0	365	365	365	365	365	365	365	365
D8	0	0	0	0	0	0	0	0
D16	0	0	0	0	0	0	0	0
D24	0	0	0	0	0	0	0	0
D32	0	0	0	0	0	0	0	0
D40	0	0	0	0	0	0	0	0
D48	0	0	0	0	0	0	0	0

1) Turn on X3.
The numeric data 365 is sent to eight registers of D0 to D7 in batch.



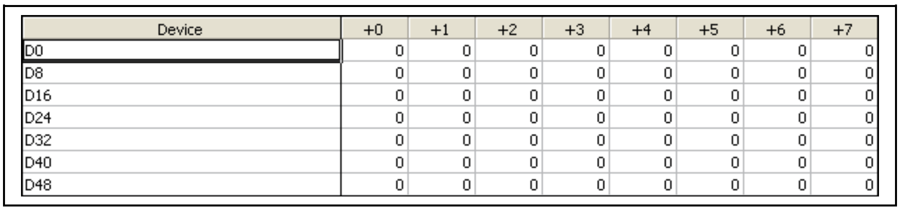
Device	+0	+1	+2	+3	+4	+5	+6	+7
D0	365	365	365	365	365	365	365	365
D8	7000	7000	7000	7000	7000	7000	7000	7000
D16	7000	7000	7000	7000	7000	7000	7000	7000
D24	0	0	0	0	0	0	0	0
D32	0	0	0	0	0	0	0	0
D40	0	0	0	0	0	0	0	0
D48	0	0	0	0	0	0	0	0

2) Turn on X4.
The numeric data 7000 is sent to 16 registers of D8 to D23 in batch.



Device	+0	+1	+2	+3	+4	+5	+6	+7
D0	365	365	365	365	365	365	365	365
D8	7000	7000	7000	7000	7000	7000	7000	7000
D16	7000	7000	7000	7000	7000	7000	7000	7000
D24	0	0	0	0	0	0	0	0
D32	365	365	365	365	365	365	365	365
D40	7000	7000	7000	7000	7000	7000	7000	7000
D48	0	0	0	0	0	0	0	0

3) Turn on X5.
The contents of the 16 registers of D0 to D15 are sent to the 16 registers of D32 to D47 in batch.



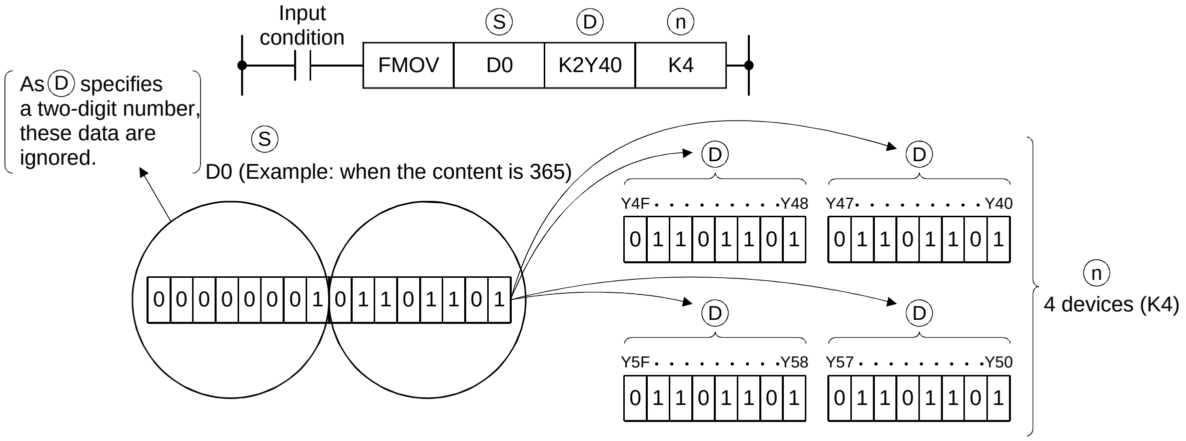
Device	+0	+1	+2	+3	+4	+5	+6	+7
D0	0	0	0	0	0	0	0	0
D8	0	0	0	0	0	0	0	0
D16	0	0	0	0	0	0	0	0
D24	0	0	0	0	0	0	0	0
D32	0	0	0	0	0	0	0	0
D40	0	0	0	0	0	0	0	0
D48	0	0	0	0	0	0	0	0

4) Turn on X6.
"0" is sent to the all 48 registers of D0 to D47 in batch. This means that all the 48 registers are cleared.

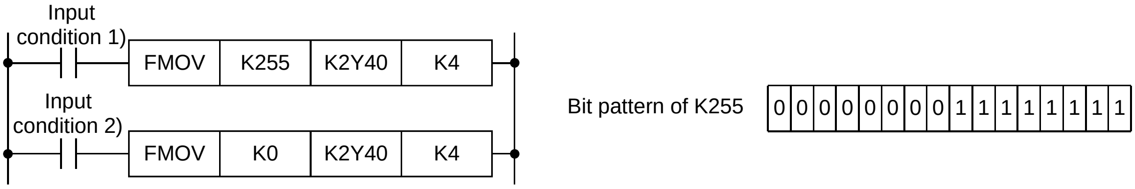
Reference

- If Ⓓ is a bit device, the operation becomes as follows;

FMOV instruction


Input
condition
Ⓢ
Ⓓ
ⓝ
FMOV
D0
K2Y40
K4
As Ⓓ specifies a two-digit number, these data are ignored.
Ⓢ
D0 (Example: when the content is 365)
0 0 0 0 0 0 0 1 0 1 1 0 1 1 0 1
Ⓓ
Ⓓ
Y4F • • • • • • • • • Y48
Y47 • • • • • • • • • Y40
0 1 1 0 1 1 0 1
0 1 1 0 1 1 0 1
Ⓓ
Ⓓ
Y5F • • • • • • • • • Y58
Y57 • • • • • • • • • Y50
0 1 1 0 1 1 0 1
0 1 1 0 1 1 0 1
ⓝ
4 devices (K4)

- Among the device of Y40 to Y5F, the devices specified as "1" are output first.
- In the program shown below, turning on the input condition 1) turns on all the outputs Y40 to Y5F and turning on the input condition 2) turns them off.

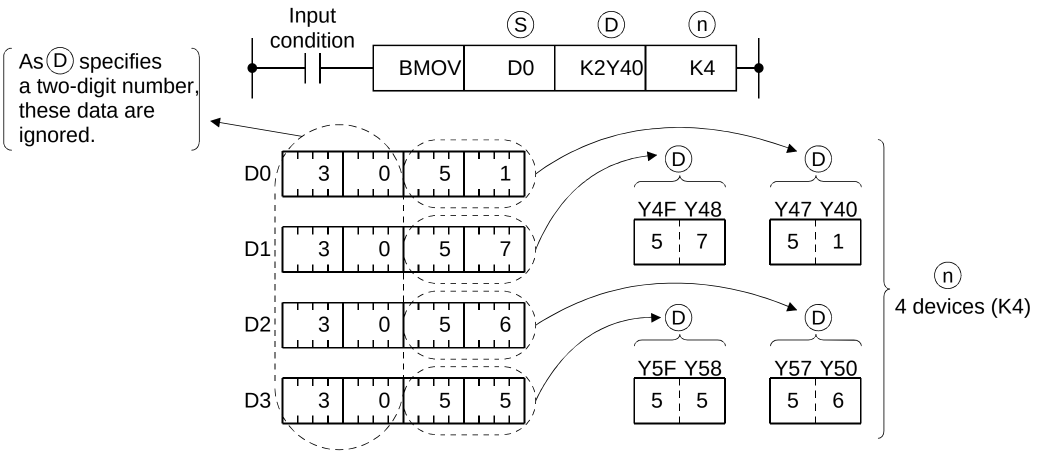


- In units of four bits, to turn off;

16 bit devices or less ⇨	MOV instruction	Example	MOV	K0	K4M0	
32 bit devices or less ⇨	DMOV instruction	Example	DMOV	K0	K8M0	
More than 32 bit devices ⇨	FMOV instruction	Example	FMOV	K0	K4M0	K4

(Turns off 64 bit devices)

BMOV instruction

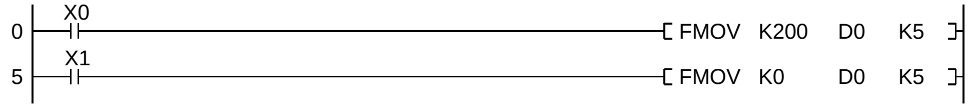


- In the example above, the devices D0 to D3 store the product code (16 bits). The BMOV instruction is useful for displaying and monitoring the last two digits representing their types.

Project name	QEX9
Program name	MAIN

Ladder example

Create the following ladder with GX Works2 and write it to the CPU of the demonstration machine. Then check that the FMOV instruction works properly.



Operating Procedure

The following procedures are the same as the Operating Procedure in section 4.4.

(1) Creating a new project

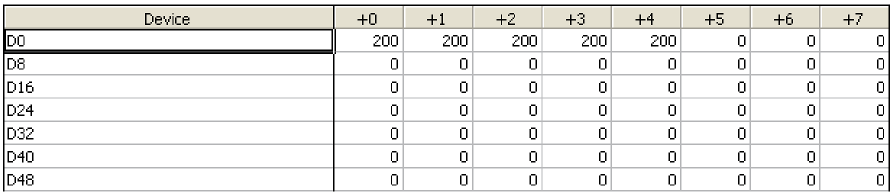
(2) Creating a program

(3) Writing the project to the programmable controller

(4) Monitoring the ladder

Operation Practice

Check that the contents of the devices of D0 to D4 become 200 on the batch monitor screen by turning on X0 on the control panel of the demonstration machine. Turning on X1 clears the data in the devices.



Device	+0	+1	+2	+3	+4	+5	+6	+7
D0	200	200	200	200	200	0	0	0
D8	0	0	0	0	0	0	0	0
D16	0	0	0	0	0	0	0	0
D24	0	0	0	0	0	0	0	0
D32	0	0	0	0	0	0	0	0
D40	0	0	0	0	0	0	0	0
D48	0	0	0	0	0	0	0	0

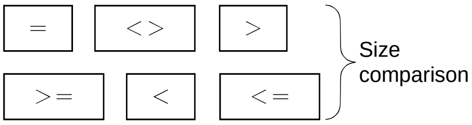
Change the setting of the device batch monitor as shown below to display the numbers in decimal, hexadecimal, or binary notation.

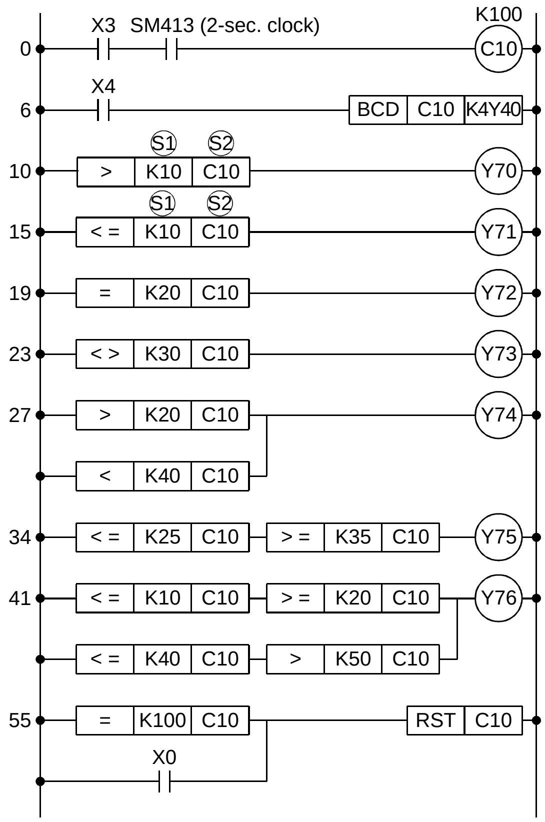
Value: DEC ···································displays numbers in decimal.
Value: HEX ···································displays numbers in hexadecimal.
Monitor Format: Bit Multi-point ·······displays numbers in binary.

Related Exercise ---- Exercise 7

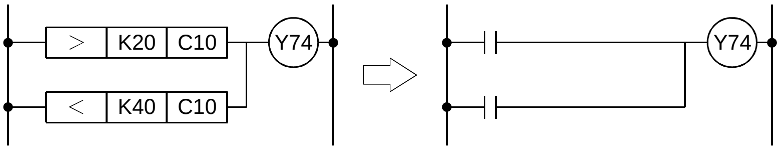
Project name	QB-15
Program name	MAIN

5.3 Comparison Operation Instruction





- The comparison operation instruction compares the data of source 1 (Ⓢ1)and source 2 (Ⓢ2), and brings the devices into conduction when the conditions are met.
- The instruction can be regarded as one normally open contact (⊣⊢) since it is conducted only when the conditions are met.

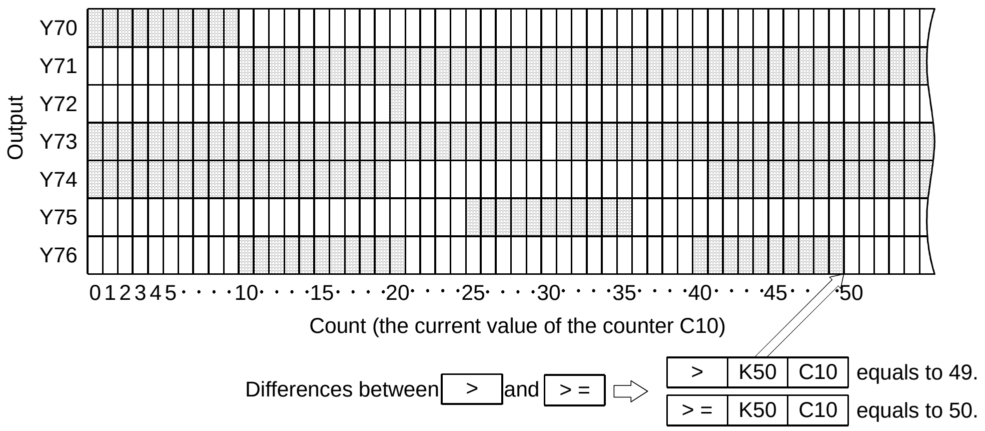


- = Ⓢ1 Ⓢ2 ……Becomes conducted when source 1 and source 2 match.
- < Ⓢ1 Ⓢ2 ……Becomes conducted when source 1 is smaller than source 2.
- > Ⓢ1 Ⓢ2 ……Becomes conducted when source 1 is larger than source 2.
- <= Ⓢ1 Ⓢ2 ……Becomes conducted when source 1 and source 2 match or when source 1 is smaller than source 2.
- >= Ⓢ1 Ⓢ2 ……Becomes conducted when source 1 and source 2 match or when source 1 is larger than source 2.
- <> Ⓢ1 Ⓢ2 ……Becomes conducted when source 1 and source 2 do not match.

Operation Practice

- Write the program to the CPU.
- Turn on X3 and X4.
- C10 starts to count. (one count every two sec.) The current counter value is displayed on the digital display (Y40 to Y4F).
- Make sure that the devices Y70 to Y76 turn on as follows.

The range where Y70 to Y76 turn on



- The counter is designed to be reset every 200sec.
- In this way, the comparison instruction does not only compare one data but also specifies the range. This function is commonly used for the program to judge the acceptances of products.

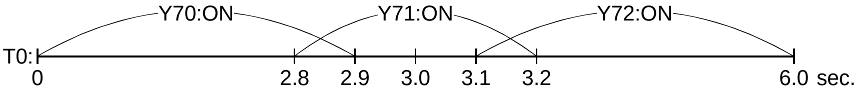
		Applicable device												Digit	Number of basic steps
		Internal device (system or user)		File register	MELSECNET /10 (H) Direct Jn\		Intelligent function module Un\G	Index register	Constant		Pointer		Level		
		Bit	Word	R	Bit	Word		Z	K	H	P	I	N		
Comparison instruction S1 S2	S1	○	○	○	○	○	○	○	○	○				K1 to K4	3
	S2	○	○	○	○	○	○	○	○	○					

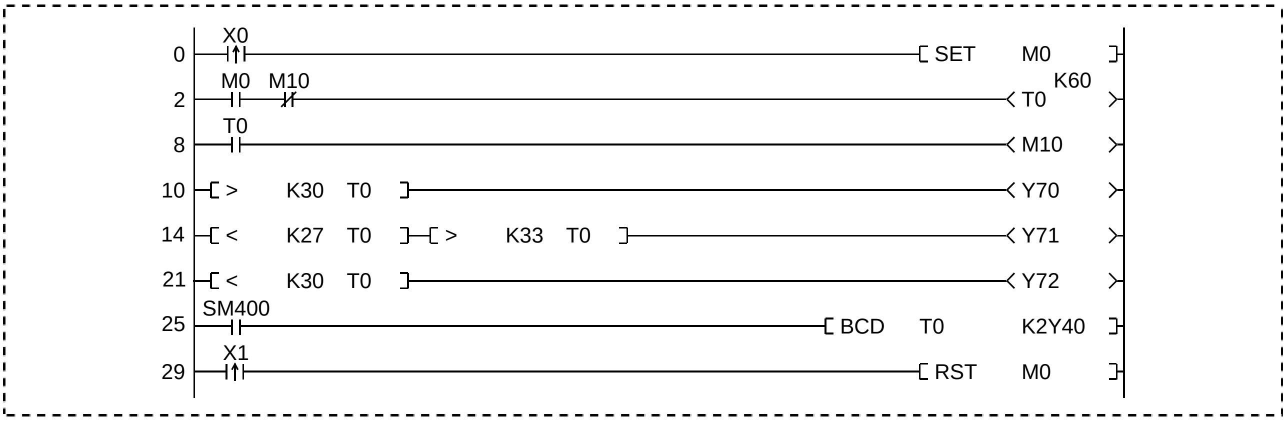
Project name	QEX10
Program name	MAIN

Ladder example

Read the following ladder and write it to the CPU of the demonstration machine. Then check that the > and < instructions work properly.

0sec. ≤ T0 < 3sec. → Y70: ON, 2.7sec. < T0 < 3.3sec. → Y71: ON, 3sec. < T0 ≤ 6sec. → Y72: ON



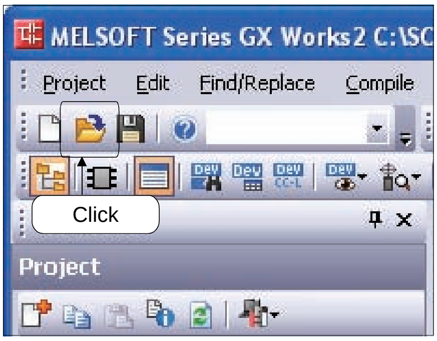


Operating Procedure

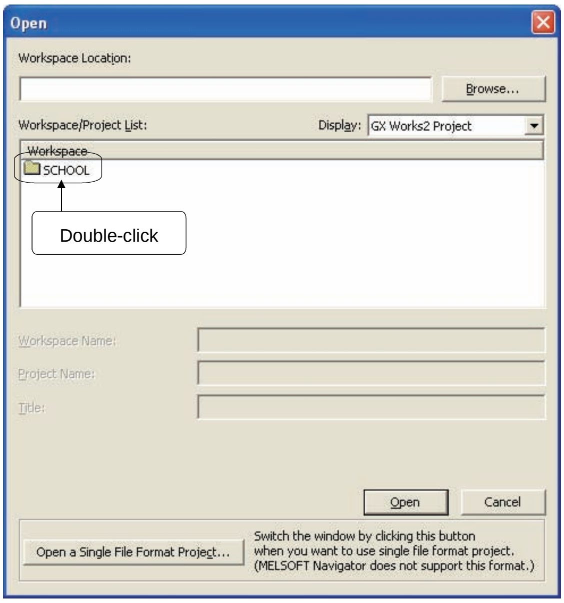
(1) Reading data

Read the project data.

- Click on the toolbar.



- The Open dialog box is displayed. Specify the save destination.
- Double-click the displayed workspace "SCHOOL".

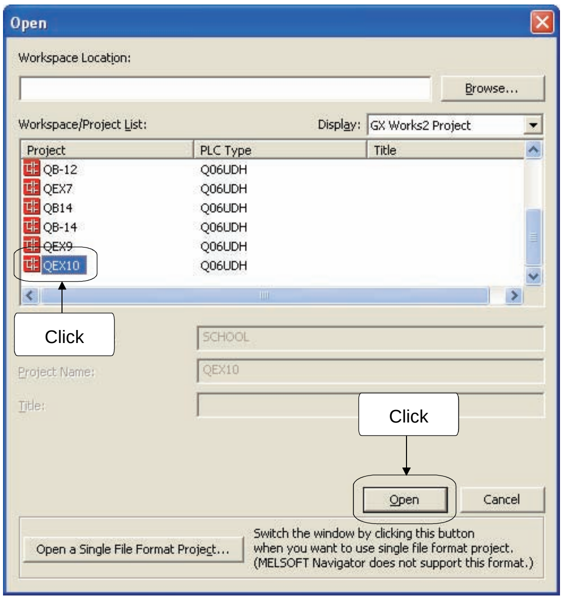


Display: GX Works2 Project

Double-click

Switch the window by clicking this button when you want to use single file format project. (MELSOFT Navigator does not support this format.)

- Click "QEX10" and click the Open button.



Display: GX Works2 Project

Project	PLC Type	Title
QB-12	Q06UDH	
QEX7	Q06UDH	
QB14	Q06UDH	
QB-14	Q06UDH	
QEX9	Q06UDH	
QEX10	Q06UDH	

SCHOOL

Project Name: QEX10

Click

Click

Switch the window by clicking this button when you want to use single file format project. (MELSOFT Navigator does not support this format.)

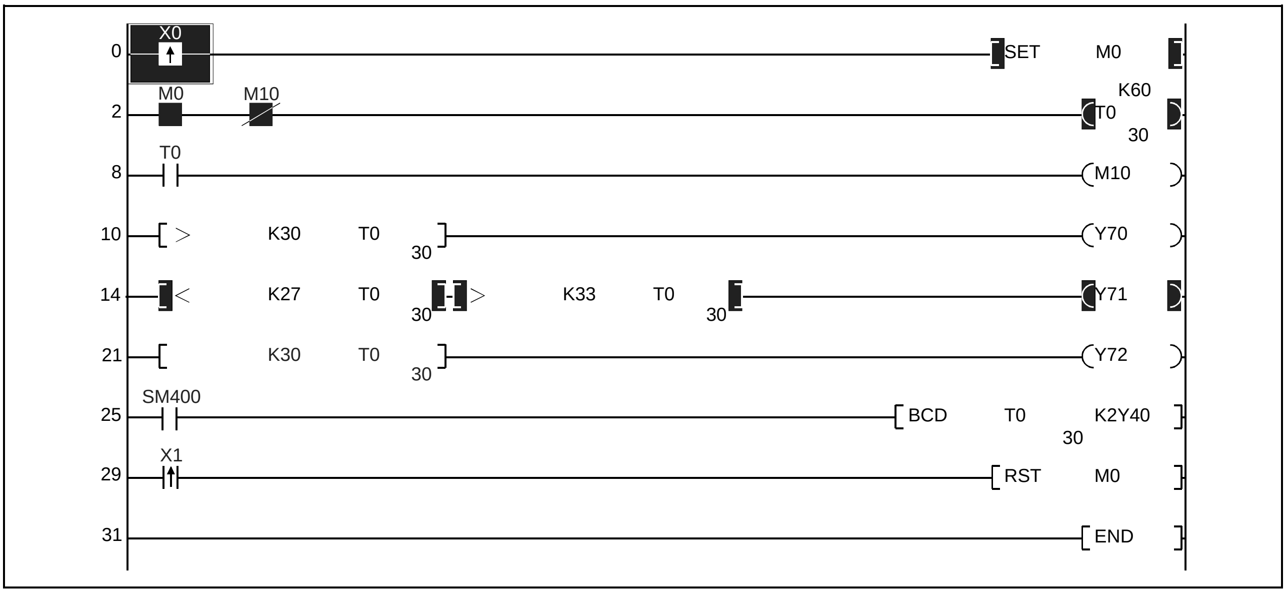
The following procedures are the same as the Operating Procedure in section 4.4.

(2) Writing the project to the programmable controller

(3) Monitoring the ladder

Operation Practice

- Turn on X0 and check that the program works properly.

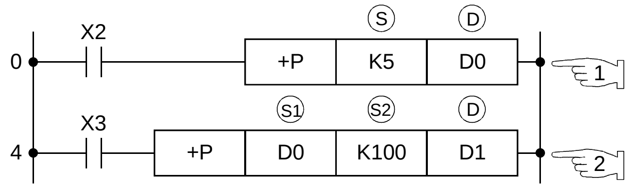


Related Exercise ---- Exercise 8

Project name	QB-16
Program name	MAIN

5.4.1 +(P) BIN 16-bit data addition

-(P) BIN 16-bit data subtraction



1 ● Every time the input condition is turned on, the content of the device specified in Ⓓ is added to the content of the device specified in Ⓢ and the result is stored in the device specified in Ⓓ.

	Ⓓ		Ⓢ		Ⓓ
	D0	+	(5)	→	D0
(Input condition)					
First ON	0 (example)	+	5	→	5
Second ON	5	+	5	→	10
Third ON	10	+	5	→	15

The content of D0 is changed.

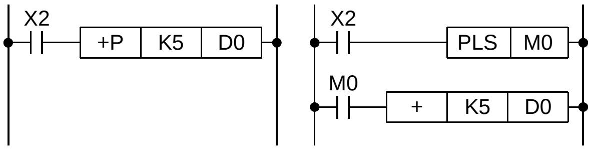
2 ● When the input condition is turned on, the content of the device specified in S1 is added to the content of the device specified in S2 and the result is stored in the device specified in Ⓓ.

	S1		S2		Ⓓ
	D0	+	(100)	→	D1
(Input condition)					
ON	15 (example)	+	100	→	115

The content of D0 is not changed.

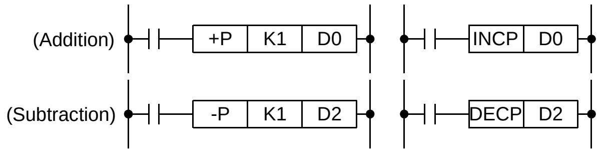
CAUTION

- +P or -P must be used for the addition or subtraction instructions.
- When + or - is used, an addition or subtraction operation is executed every scanning. To use + or -, operands must be converted into pulse in advance.

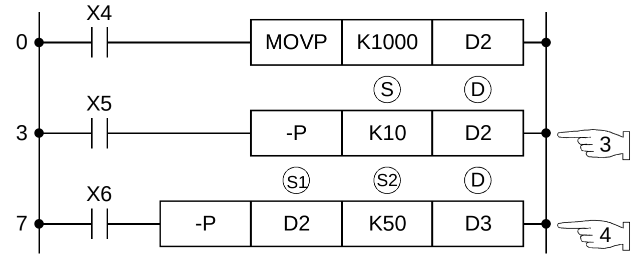


REFERENCE

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



Project name	QB-17
Program name	MAIN



3 ● Every time the input condition is turned on, the content of the device specified in Ⓢ is subtracted from the device specified in Ⓓ and the result is stored in the device specified in Ⓓ.

	Ⓓ		Ⓢ		Ⓓ
	D2 □	-	(10)	→	D2 □
(Input condition)					
First ON	1000 (example)	-	10	→	990
Second ON	990	-	10	→	980
Third ON	980	-	10	→	970

The content of D2 is changed.

4 ● When the input condition is turned on, the content of the device specified in Ⓢ2 is subtracted from the content of the device specified in Ⓢ1 and the result is stored in the device specified in Ⓓ.

	Ⓢ1		Ⓢ2		Ⓓ
	D2 □	-	(50)	→	D3 □
(Input condition)					
ON	970 (Assumption)	-	50	→	920

The content of D2 is not changed.

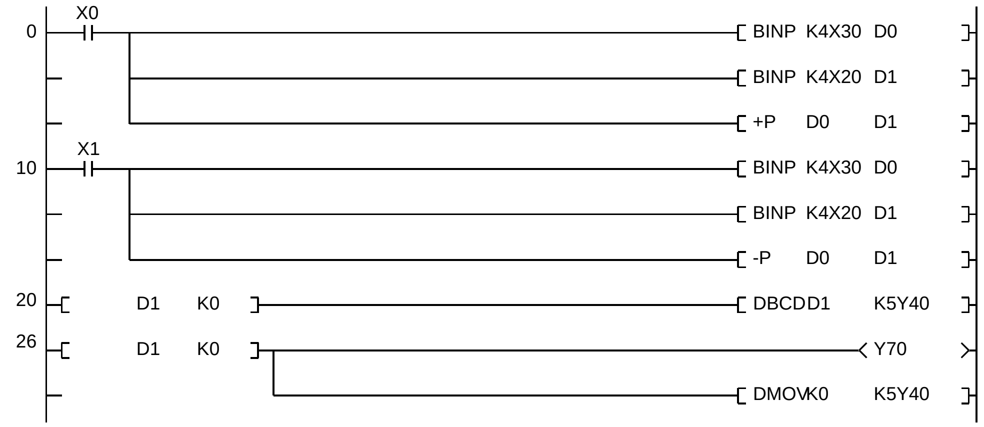
		Applicable device													Digit	Number of basic steps
		Internal device (system or user)		File register	MELSECNET/10 (H) Direct Jn\		Intelligent function module Un\G	Index register	Constant		Pointer		Level			
		Bit	Word	R	Bit	Word		Z	K	H	P	I	N			
Addition/subtraction instruction Ⓢ Ⓓ Addition/subtraction instruction Ⓢ1 Ⓢ2 Ⓓ	Ⓢ Ⓢ1 Ⓢ2	○	○	○	○	○	○	○	○	○				K1 to K4	3/4
	Ⓓ	○	○	○	○	○	○	○							

The number of basic steps is four for □ Ⓢ1 Ⓢ2 Ⓓ.

Project name	QEX11
Program name	MAIN

Ladder example

Create the following ladder with GX Works2 and write it to the CPU of the demonstration machine. Then check that the addition and subtraction instructions operate properly.



Operating Procedure

The following procedures are the same as the Operating Procedure in section 4.4.

(1) Creating a new project

(2) Creating a program

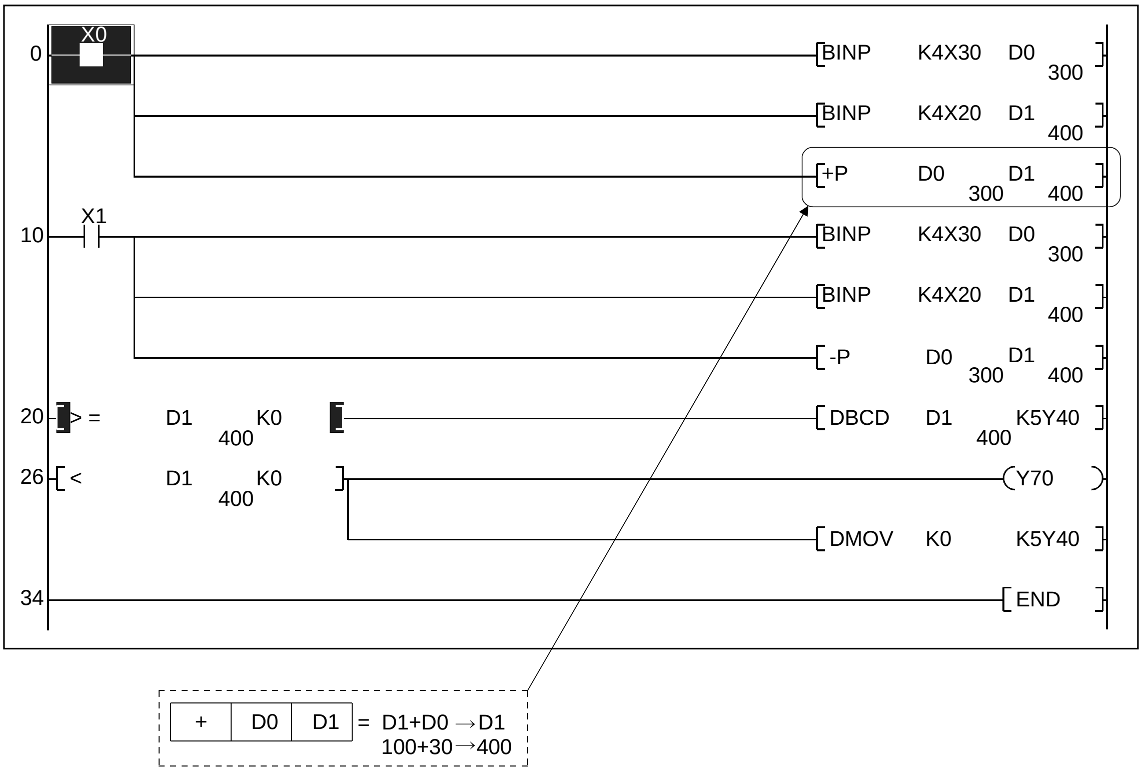
(3) Writing the project to the programmable controller

(4) Monitoring the ladder

Operation Practice

(1) When X0 is turned on, the data in X30 to 3F and X20 to 2F are added, and the result is output to Y40 to Y53.

(2) When X1 is turned on, the data in X30 to 3F is subtracted from the data in X20 to 2F, and the result is output to Y40 to Y53. When the result is a negative value, Y70 is turned on and Y40 to Y53 are cleared to 0.

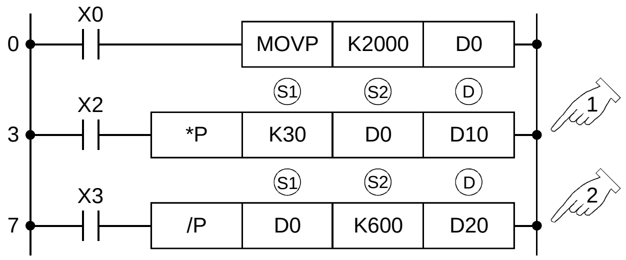


Related Exercise ---- Exercise 9

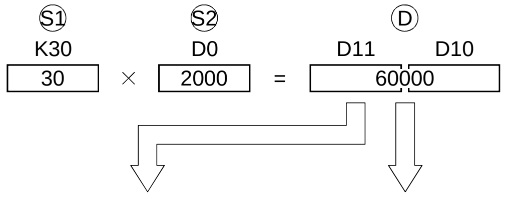
Project name	QB-18
Program name	MAIN

5.4.2 * (P) BIN 16-bit data multiplication

/ (P) BIN 16-bit data division



1 ● When the input condition is turned on, the content of the device specified in Ⓢ1 is multiplied by the content of the device specified in Ⓢ2 and the result is stored in the device specified in Ⓓ.

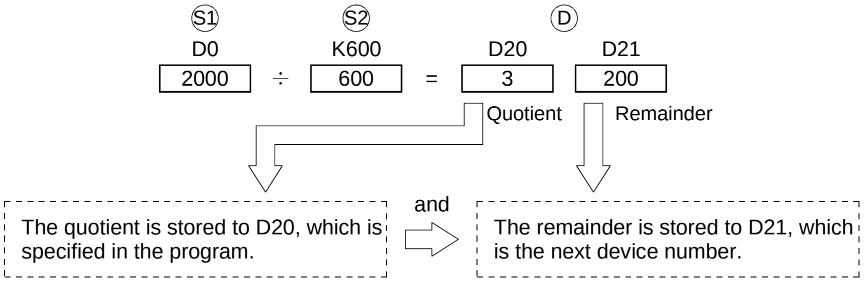


To store the result of 16-bit data × 16-bit data, 16 bits (1 word) is not enough. Therefore, D10 which is specified in the program and the next number D11 work as the holder of the result.

This device is regarded as a 32-bit register to hold the result. Left-most bit of D10 (B15) is not a bit to determine positive and negative. It is regarded as a part of the data.

The instructions must be regarded as 32 bits for programming with the calculation result of the *(P) instruction. (such as the DMOV instruction and the DBCD instruction)

☞2 ● When the input condition is turned on, the content of the device specified in Ⓢ1 is divided by the content of the device specified in Ⓢ2 and the result is stored in the device specified in Ⓓ.



The quotient is stored to D20, which is specified in the program.

The remainder is stored to D21, which is the next device number.

Values after the decimal point of the operation result are ignored.

- When a bit device is specified in Ⓓ, the quotient is stored, but the remainder is not stored.
- The following shows examples for processing negative values.

Example -5 / (-3) = 1, remainder: -2

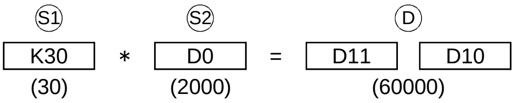
5 / (-3) = -1, remainder: 2

- The following shows examples for dividing a number by 0, or dividing 0 by a number.

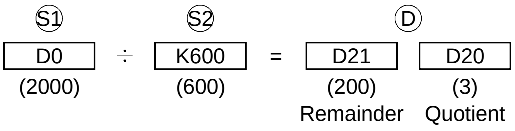
Example
0 / 0 }
1 / 0 } Error "OPERATION ERROR"
0 / 1, Both quotient and remainder are 0.

Operation Practice

- Write the program to the CPU and run it.
- Turn on X0 and store "2000" (BIN value) in D0.
- Turn on X2. The following operation is executed.
  If "60000" (operation result of D11 and D10 is regarded as a 16-bit integral number and only D10 is monitored, "-5536" is displayed. To prevent this, follow the procedures in the following pages.



- Turn on X3.



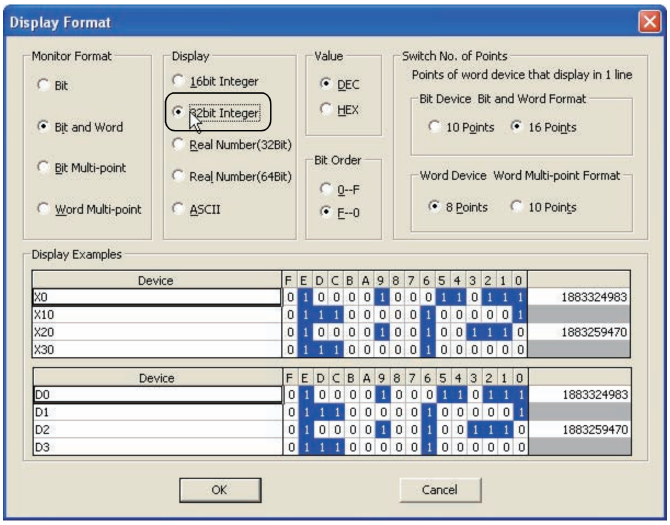
		Applicable device												Digit	Number of basic steps
		Internal device (system or user)		File register	MELSECNET/ 10 (H) Direct Jn\		Intelligent function module Un\G	Index register	Constant		Pointer		Level		
		Bit	Word	R	Bit	Word		Z	K	H	P	I	N		
Multiplication/division instruction Ⓢ1 Ⓢ2 Ⓓ	Ⓢ1	○	○	○	○	○	○	○	○	○				K1 to K4	* / 4
	Ⓢ2	○	○	○	○	○	○	○	○	○					
	Ⓓ	○	○	○	○	○	○	○							

The number of basic steps for the multiplication instruction is three or four, and that for division instruction is four.
*: The multiplication instruction varies depending on the device to be used.

● How to monitor 32-bit integral number data

When the operation result of the multiplication instruction is outside the range from 0 to 32,767, the result cannot be displayed properly even though the number is regarded as 16-bit integral number and the contents of the lower register are monitored in ladder.
To monitor those numbers properly, follow the procedures below.

- Click the Display Format button on the Device/Buffer Memory Batch Monitor dialog box and select "32bit Integer" of "Display".
  Click the OK button.



Display Format

Monitor Format
○ Bit
● Bit and Word
○ Bit Multi-point
○ Word Multi-point

Display
○ 16bit Integer
● 32bit Integer
○ Real Number(32Bit)
○ Real Number(64Bit)
○ ASCII

Value
● DEC
○ HEX

Bit Order
○ 0--F
● F--0

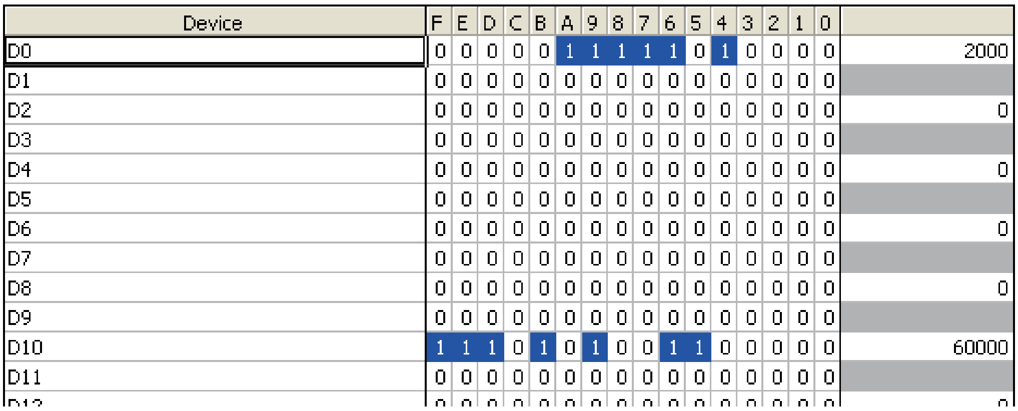
Switch No. of Points
Points of word device that display in 1 line
Bit Device Bit and Word Format
○ 10 Points ● 16 Points
Word Device Word Multi-point Format
● 8 Points ○ 10 Points

Display Examples

Device	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	
X0	0	1	0	0	0	0	1	0	0	0	1	1	0	1	1	1	1883324983
X10	0	1	1	1	0	0	0	0	0	1	0	0	0	0	0	1	
X20	0	1	0	0	0	0	1	0	0	1	0	0	1	1	1	0	1883259470
X30	0	1	1	1	0	0	0	0	0	1	0	0	0	0	0	0	

Device	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	
D0	0	1	0	0	0	0	1	0	0	0	1	1	0	1	1	1	1883324983
D1	0	1	1	1	0	0	0	0	0	1	0	0	0	0	0	1	
D2	0	1	0	0	0	0	1	0	0	1	0	0	1	1	1	0	1883259470
D3	0	1	1	1	0	0	0	0	0	1	0	0	0	0	0	0	

- The data is monitored properly.

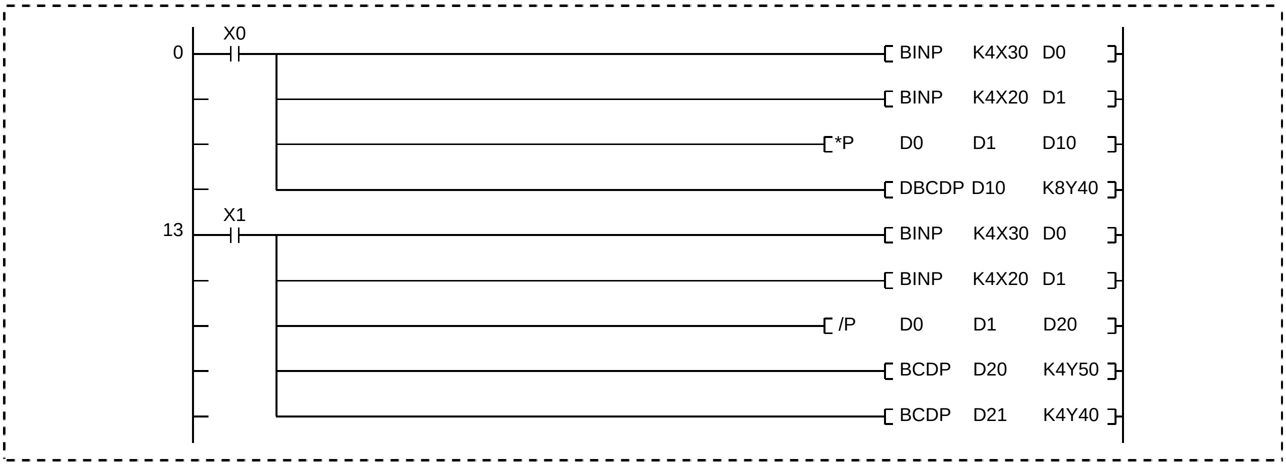


Device	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	
D0	0	0	0	0	0	1	1	1	1	1	0	1	0	0	0	0	2000
D1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
D2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
D4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
D6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
D8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
D10	1	1	1	0	1	0	1	0	0	1	1	0	0	0	0	0	60000
D11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
D12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Project name	QEX12
Program name	MAIN

Ladder example

Create the following ladder with GX Works2 and write it to the CPU of the demonstration machine. Then check that the multiplication and division instructions operate properly.



Operating Procedure

The following procedures are the same as the Operating Procedure in section 4.4.

(1) Creating a new project

(2) Creating a program

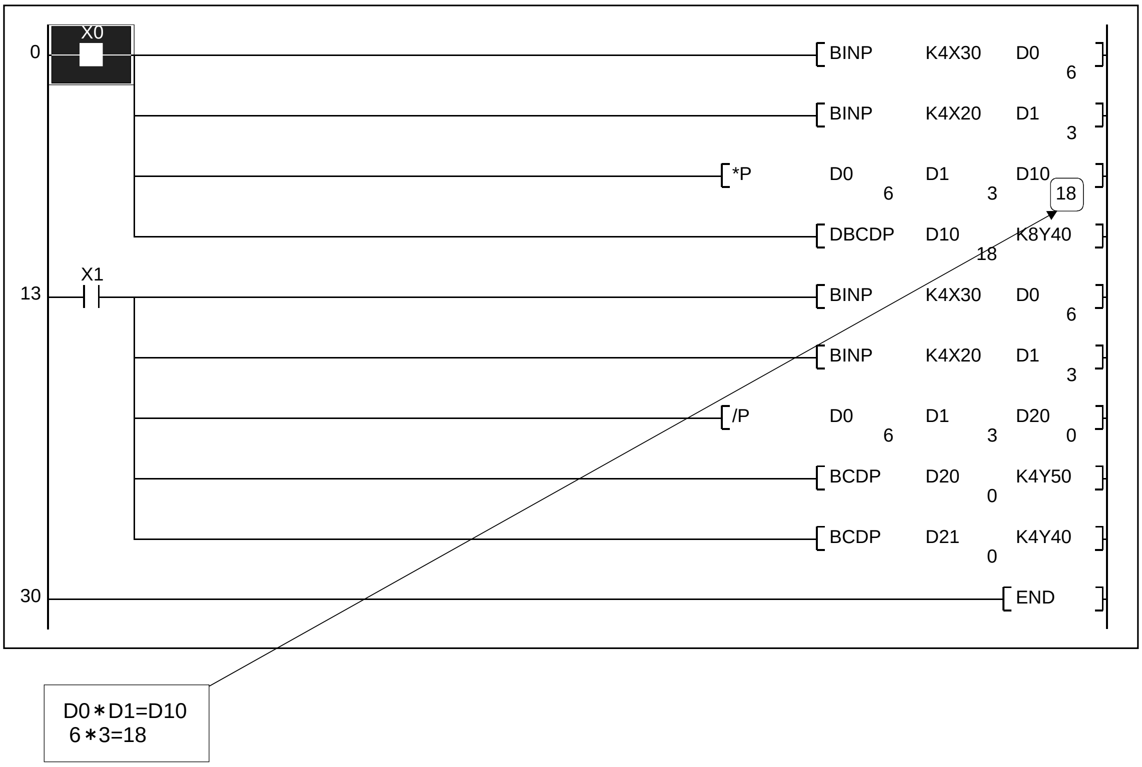
(3) Writing the project to the programmable controller

(4) Monitoring the ladder

Operation Practice

(1) When X0 is turned on, the data in X20 to X2F is multiplied by the data in X30 to 3F, and the result is output to Y40 to 5F.

(2) When X1 is turned on, the data in X30 to X3F is divided by the data in X20 to 2F. The quotient is output to Y50 to 5F, and the remainder is output to Y40 to 4F.



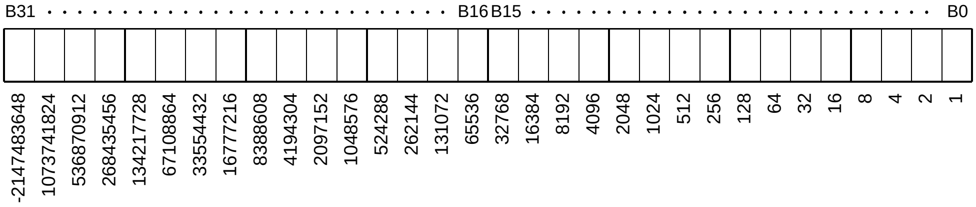
Related Exercise ---- Exercise 10, Exercise 11

5.4.3 32-bit data instructions and their necessity

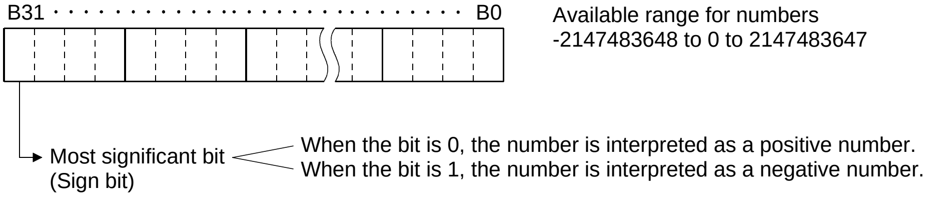
- The minimum unit in the data memory of the Q-series programmable controller is 1 word which consists of 16 bits. Therefore, in general, data is processed in 1-word basis at the transfer, comparison, and arithmetic operation.
- The Q-series programmable controller can process data in 2-word (32-bit) basis. In that case, "D" is added at the head of each instruction to indicate that the instruction is regarded as 2-word. The following shows the examples.

Data Instruction	1 word 16 bits	2 words 32 bits
Transfer	MOV(P)	DMOV(P)
	BIN(P)	DBIN(P)
	BCD(P)	DBCD(P)
Comparison	<, >, <= >=, =, <>	D<, D>, D<= D>=, D=, D<>
Arithmetic operations	+ (P)	D + (P)
	- (P)	D - (P)
	* (P)	D * (P)
	/(P)	D/(P)
Available range for values	-32,768 to 32,767 (0 to 9,999) Values in parentheses are for BIN(P), BCD(P) instructions.	-2,147,483,648 to 2,147,483,647 (0 to 99,999,999) Values in parentheses are for DBIN(P), DBCD(P) instructions.
Available range for digits	K1 to K4	K1 to K8

- The bit weights of the 32-bit configuration are as follows:



As the case of 16-bit data processing, the programmable controller processes a 32-bit negative value in two's complement. Therefore, the most significant bit B31 (B15 for 16-bit data), is a sign bit.



Available range for numbers
-2147483648 to 0 to 2147483647

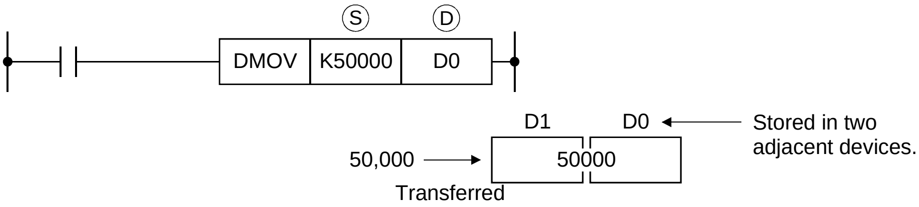
Most significant bit
(Sign bit)

When the bit is 0, the number is interpreted as a positive number.
When the bit is 1, the number is interpreted as a negative number.

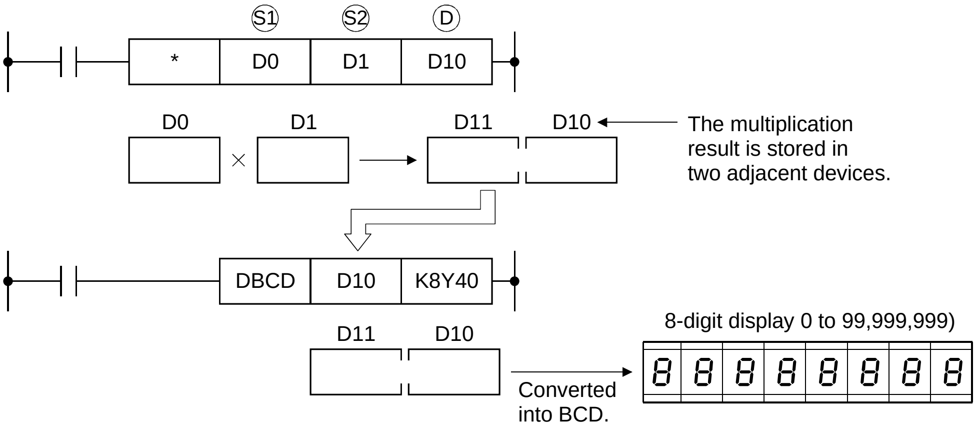
- Whether the data is processed in 2-word (32-bit) basis or not depends on the size of the data.

  In the following cases, 2-word instructions must be used.

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



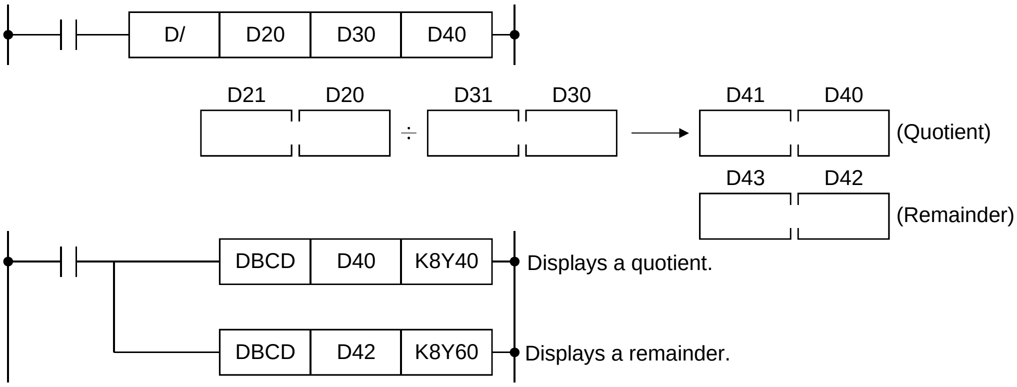
  2) When the result of the 16-bit multiplication instruction (1-word instruction) is transferred



S1
S2
D
*
D0
D1
D10
D0
D1
D11
D10
The multiplication result is stored in two adjacent devices.
DBCD
D10
K8Y40
8-digit display 0 to 99,999,999)
D11
D10
Converted into BCD.

*: The result of the 32-bit data multiplication is 64 bits.

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



Project name	QB-19
Program name	MAIN

5.4.4 Calculation examples for multiplication and division including decimal points (when the multiplication or division is used)

(Example 1) Calculation example to determine a circumference

$\boxed{\text{Digitalswitch value}}$ × 3.14 → $\boxed{\text{Integral part}}$ and $\boxed{\text{Decimal part}}$

(K4X30) (Circular constant) (K8Y50) (K2Y48)

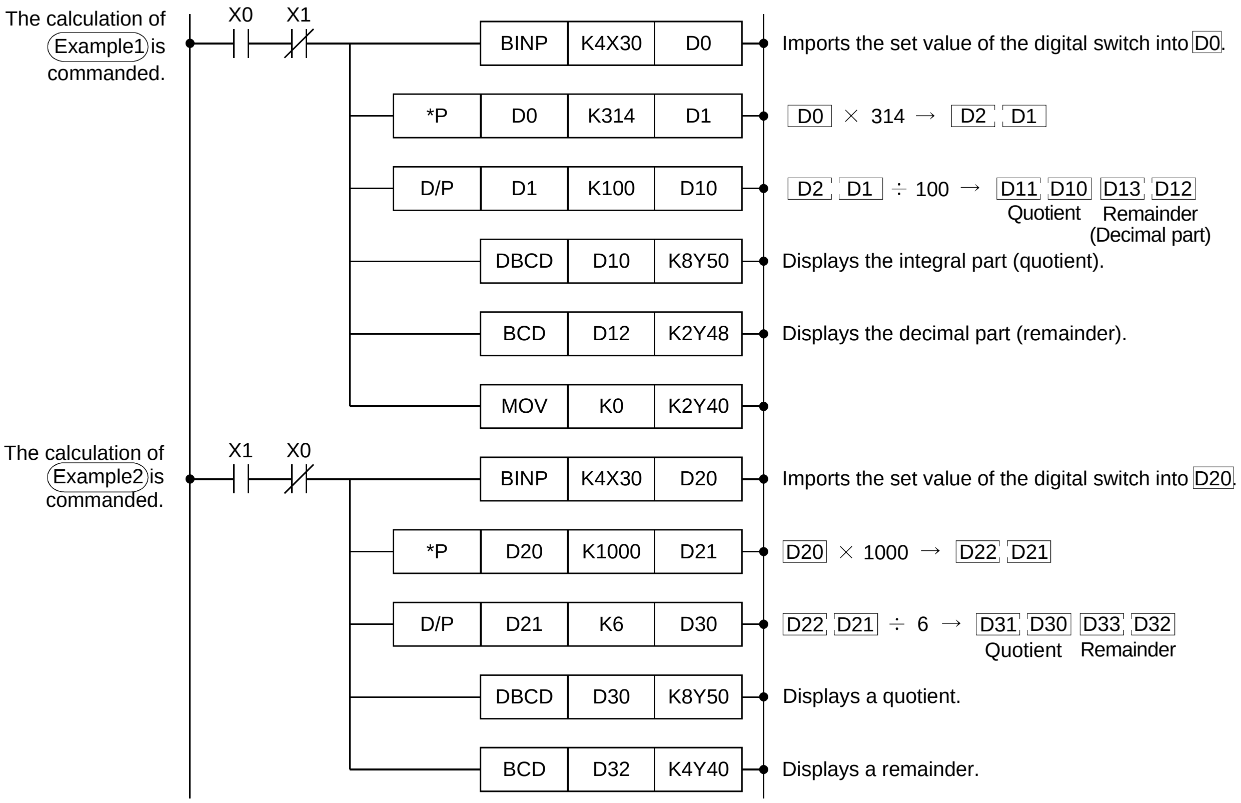
- Programming method
  Handle the circular constant as 314 (3.14 × 100), and divide the result by 100 afterward.

(Example 2) Calculation example to handle values after decimal point (division example)

$\boxed{\text{Digitalswitch value}}$ / 0.006 → $\boxed{\text{Quotient}}$ and $\boxed{\text{Remainder}}$

(K4X30) (K8Y50) (K4Y40)

- Programming method
  To handle 0.006 as an integer 6, multiply both the dividend and divisor by 1000.



The calculation of (Example1) is commanded.
X0
X1
BINP K4X30 D0
Imports the set value of the digital switch into D0.
*P D0 K314 D1
D0 × 314 → D2, D1
D/P D1 K100 D10
D2, D1 ÷ 100 → D11, D10 D13, D12
Quotient
Remainder (Decimal part)
DBCD D10 K8Y50
Displays the integral part (quotient).
BCD D12 K2Y48
Displays the decimal part (remainder).
MOV K0 K2Y40
The calculation of (Example2) is commanded.
X1
X0
BINP K4X30 D20
Imports the set value of the digital switch into D20.
*P D20 K1000 D21
D20 × 1000 → D22, D21
D/P D21 K6 D30
D22, D21 ÷ 6 → D31, D30 D33, D32
Quotient
Remainder
DBCD D30 K8Y50
Displays a quotient.
BCD D32 K4Y40
Displays a remainder.

REMARK

QCPU has instructions which can process actual number (floating decimal point) operation data for highly accurate operations.
As long as the instructions are used, careful attentions for the place of the decimal point as shown above are unnecessary.

5.5 Index Register and File Register

5.5.1 How to use index register Z

- The index register (Zn) is used to indirectly specify the device number. The result of an addition of data in the index register and the directly specified device number can be specified as the device number.

Example

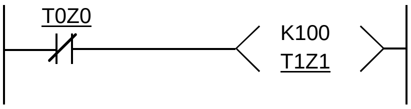
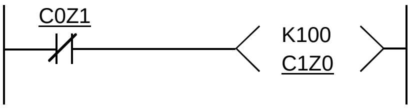
D0Z0 → Can be interpreted as D (0+Z0)
Device number

For example, when Z0 is 0, the device number becomes D0.
when Z0 is 50, the device number becomes D50.

- Z0 to Z19 can be used as the index register.
- The index register (Zn) is a word device which consists of 16 bits. Therefore, the available data size range is -32768 to +32767.
- The following devices can be used for the indexing.

Bit device ········	X, Y, M, L, S, B, F, Jn\X, Jn\Y, Jn\B, Jn\SB	(such as K4Y40Z0)
Word device ····	T (Note), C (Note), D, R, W, Jn\W, Jn\SW, Jn\G	(such as D0Z0)
Constant ·········	K, H	(such as K100Z0)
Pointer ···········	P	

(Note) Only the current value can be used for timer and counter.
The following restrictions are provided for using the index register for contact or coil.

Device	Description	Application example
T	• Only Z0 and Z1 are available for a contact and coil of a timer.	
C	• Only Z0 and Z1 are available for a contact and coil of a counter.	

REMARK

When the index register is used with 32-bit data instructions

Zn and Zn+1 are targets to be processed.
The lower 16 bits correspond to the specified index register number (Zn), and the higher 16 bits correspond to the specified index register number + 1.

32-bit indexing
(Only for Universal model QCPU (except for Q00UJCPU))

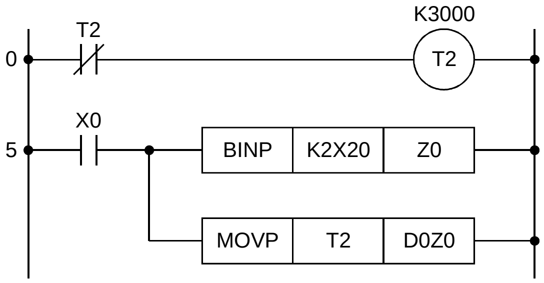
A method for specifying index registers for 32-bit indexing can be selected from following two methods.
- Specifying the index range used for 32-bit indexing
- Specifying the 32-bit indexing using "ZZ" specification

Refer to appendix 8 for the detail of indexing.

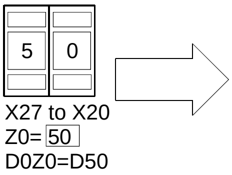
Application Example

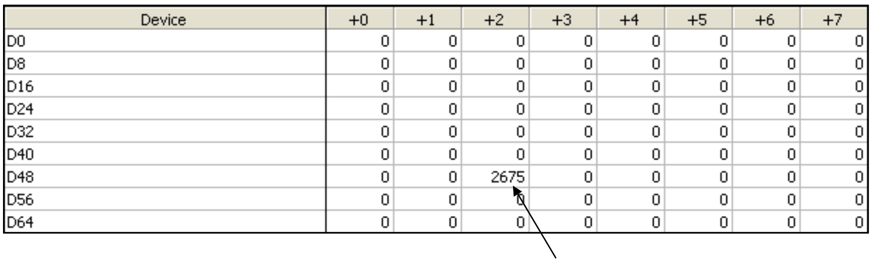
- Write the data to the data register with number which is specified with the digital switch.

Project name	Index register
Program name	MAIN



- Check the operation of the ladder executing the device batch monitoring.
  The operation procedure is the same as the one in section 5.2.1.
  Set any two-digit number in the digital switch column (X27 to X20) and turn on X0.



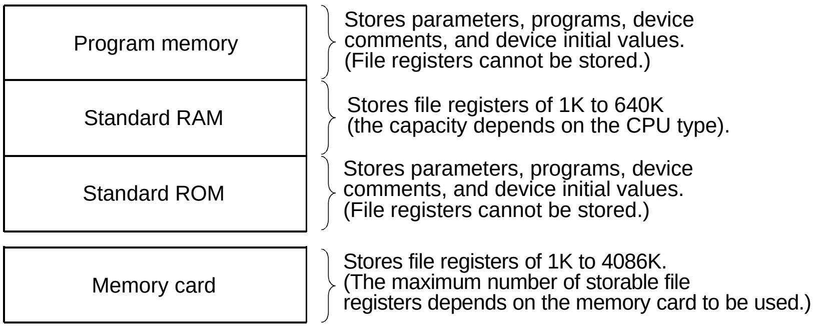


Device	+0	+1	+2	+3	+4	+5	+6	+7
D0	0	0	0	0	0	0	0	0
D8	0	0	0	0	0	0	0	0
D16	0	0	0	0	0	0	0	0
D24	0	0	0	0	0	0	0	0
D32	0	0	0	0	0	0	0	0
D40	0	0	0	0	0	0	0	0
D48	0	0	2675	0	0	0	0	0
D56	0	0	0	0	0	0	0	0
D64	0	0	0	0	0	0	0	0

The current value of T2 is transferred to D50.

5.5.2 How to use file register R

- The file register (R) consists of 16 bits as well as the data register (D).
- Set the file register in the standard RAM of the QCPU or a memory card (SRAM card and Flash card). The file register to be stored in the Flash card can be read from the program only. The data cannot be changed with the program.



Program memory: Stores parameters, programs, device comments, and device initial values. (File registers cannot be stored.)

Standard RAM: Stores file registers of 1K to 640K (the capacity depends on the CPU type).

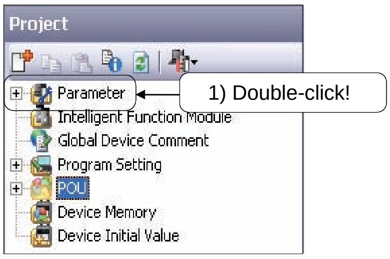
Standard ROM: Stores parameters, programs, device comments, and device initial values. (File registers cannot be stored.)

Memory card: Stores file registers of 1K to 4086K. (The maximum number of storable file registers depends on the memory card to be used.)

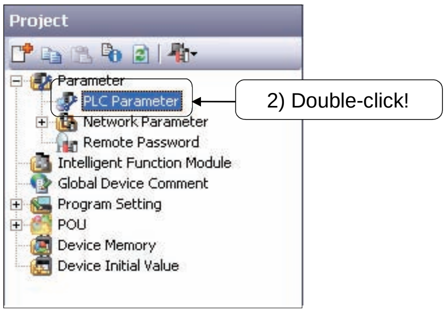
- The data in the file register remains after the reset operation or after the power is turned off.
- To clear the data, write 0 to the file register with the MOV(P) instruction or GX Works2.
- Use [Write to PLC] of GX Works2 or a sequence program to write data to the standard RAM or SRAM card.
- Use [Export to ROM Format] of GX Works 2 to copy data in the standard ROM or Flash card.
- Specify the area of the file register in 1K (1024 point) basis with the parameter.

Application Example

- Set 32K points of the file register R0 to R32767 to use in the program.
  Follow the procedures below to register the file register to the parameter.



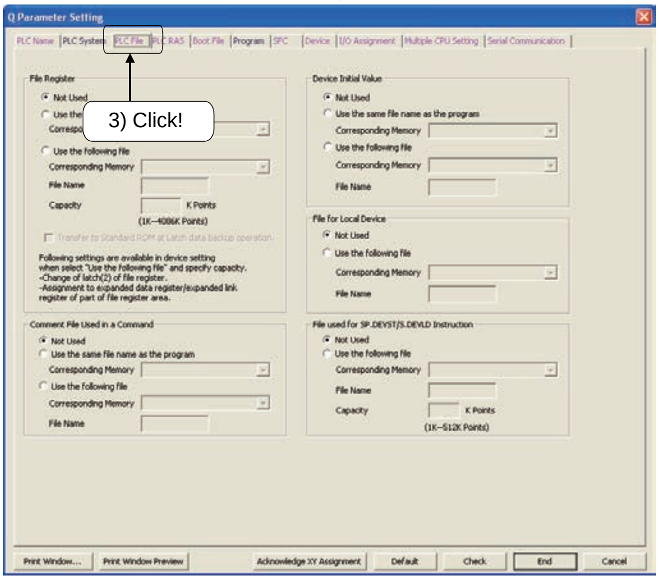
1) Double-click "Parameter" in the project list.



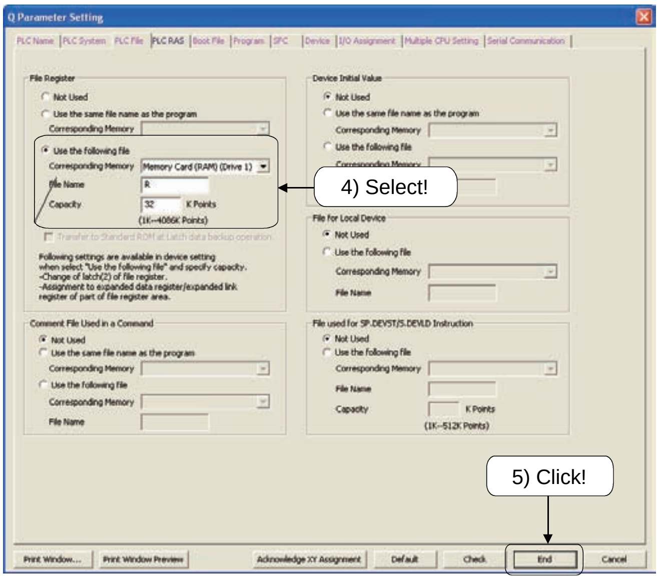
2) "PLC Parameter", "Network Parameter", and "Remote Password" are displayed.
Double-click "PLC Parameter"

(To the next page)

(From the previous page)



3) The Q Parameter Setting dialog box is displayed. Click the PLC File tab.



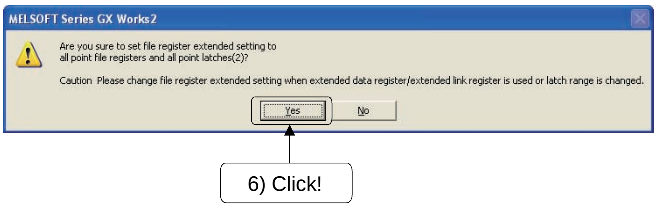
Corresponding Memory: Memory Card (RAM) (Drive 1)
File Name: R
Capacity: 32 K Points

4) Check the "Use the following file" check box and select "Memory Card (RAM) (Drive 1)" for "Corresponding Memory".

Enter the following items in "File Name" and "Capacity".

[Setting contents]
File Name : R
Capacity : 32

5) After the setting is completed, click the End button.



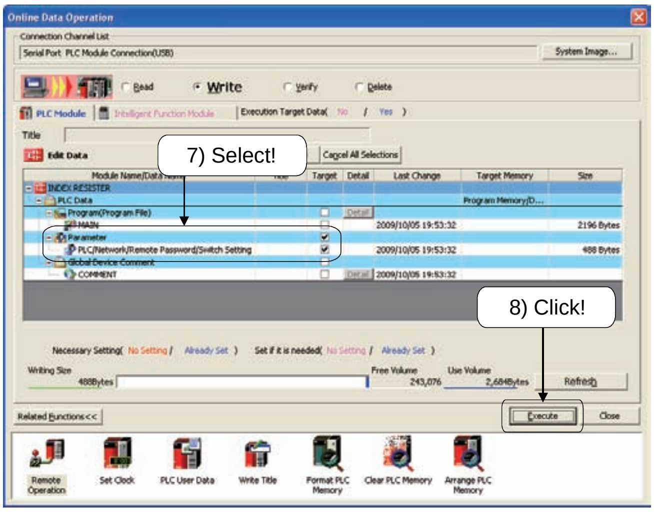
Are you sure to set file register extended setting to
all point file registers and all point latches(2)?

Caution Please change file register extended setting when extended data register/extended link register is used or latch range is changed.

6) The message on the left is displayed. Click the Yes button.

(To the next page)

(From the previous page)



7) Select!

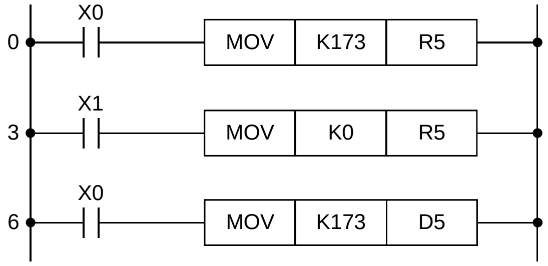
8) Click!

7) Click [Online] → [Write to PLC] to display the Online Data Operation dialog box. Select "Parameter" in the PLC Module tab.

8) Click the Execute button to write the data.

- To clear the file register data with the program, write the following program.
  For the operation procedure, refer to section 4.4.
  Turning on X0 can write the data, and turning on X1 can clear the data.

Project name	File register
Program name	MAIN



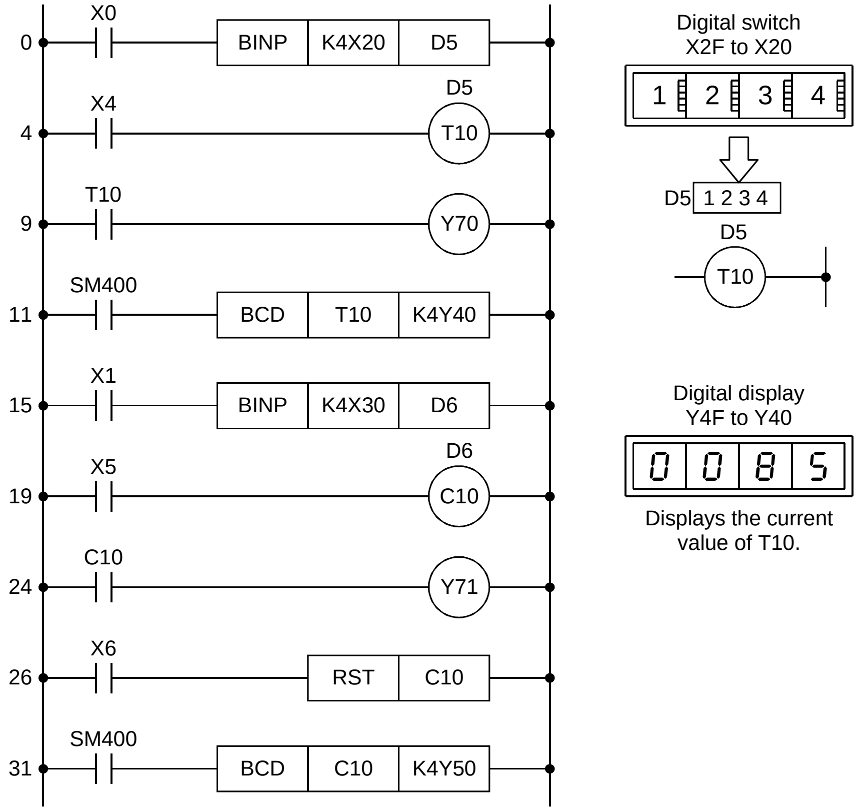
The file register data of the standard RAM is kept by the battery.
Resetting or turning off the power cannot clear the data. To clear the data, write "0".

For comparison

5.6 External Setting of Timer/Counter Set Value and External Display of Current Value

The timer and counter can be specified by K (decimal constant) directly or by D (data register) indirectly. In the program shown below, the external digital switch can change the set value.

Project name	QTC
Program name	MAIN



- After reading the program to GX Works2, write it to the programmable controller to check that it works properly.

Operating Procedure

The step (1) of the following procedure is the same as Operating Procedure in section 5.3.

The steps (2) to (4) of the following procedure are the same as Operating Procedure in section 4.4.

(1) Reading the data

(2) Creating a program

(3) Writing the project to the programmable controller

(4) Monitoring the ladder

Operation Practice

(1) External setting of the timer set value and display of the current value

- Set the timer set value in the digital switch (X20 to 2F), and turn on the switch X0.
- When the switch X4 is turned on, Y70 turns on after the time specified with the digital switch. (For example, Y70 turns on after 123.4sec.
  when $\boxed{1}\boxed{2}\boxed{3}\boxed{4}$ is set.)
- The digital display (Y40 to 4F) displays the current value of the timer T10.

(2) External setting of the counter set value and display of the current value

- Set the counter set value in the digital switch (X30 to 3F), and turn on the switch X1.
- Turn on and off the switch X5 repeatedly. When X5 has turned on for the times specified with the digital switch (count up), Y71 turns on.
- The digital display (Y50 to 5F) displays the current value of the counter C10 (the number of times that X5 is turned on).
- Turning on the switch X6 clears the counter C10 to 0. When the contact C10 is already turned on (count up), the contact is released.

Project name	QTEST5
Program name	MAIN

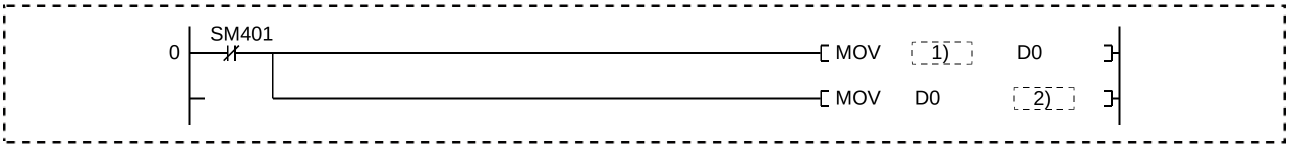
5.7 Exercise

5.7.1 Exercise 1 MOV

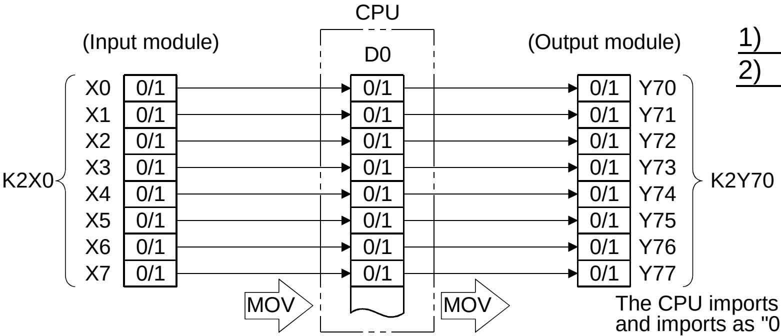
Transfer the eight input statuses (X0 to X7) to D0 once then output them to Y70 to Y77. (For example, Y70 turns on when X0 turns on.)

X0 → Y70
X1 → Y71
X2 → Y72
X3 → Y73
X4 → Y74
X5 → Y75
X6 → Y76
X7 → Y77

Create the following program with GX Works2 filling in the blanks ⬚.
Then, check the operation using the demonstration machine.



Hint

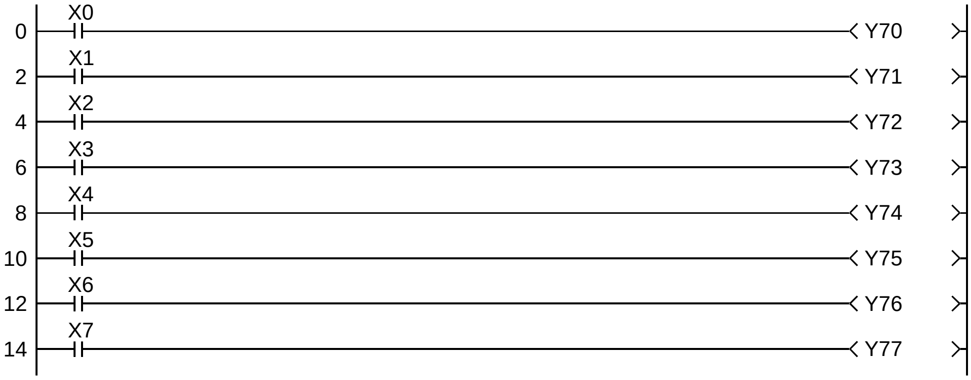


1) ______________________
2) ______________________

The CPU imports the input signal as "1" when it is on, and imports as "0" when it is off.
The output module turns on when the CPU outputs "1", and turns off when the CPU outputs "0".

Comparison

The following shows a program which is created with the sequence instructions, not with the MOV instruction.

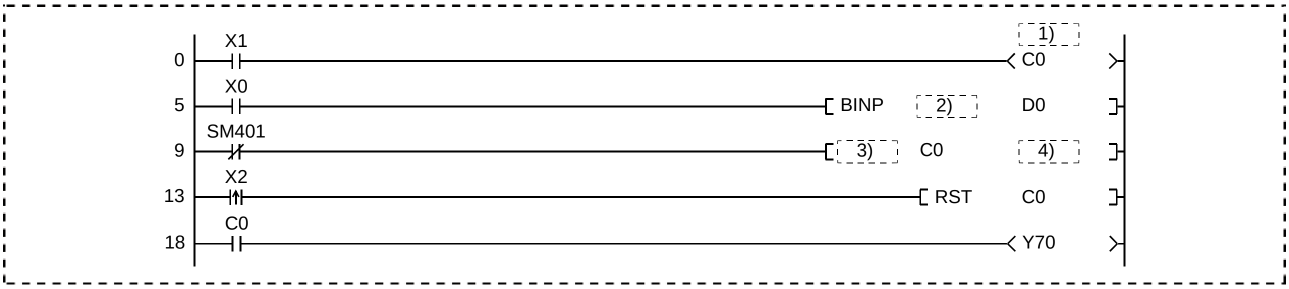


Project name	QTEST6
Program name	MAIN

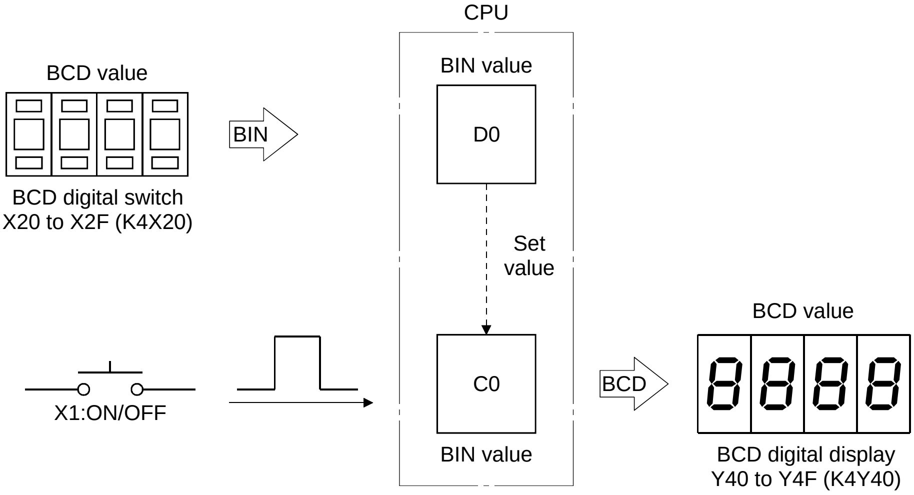
5.7.2 Exercise 2 BIN and BCD conversion

Output the number of times that X1 is turned on on the display connected to Y40 to Y4F in BCD. As a precondition, the set value of the counter (C0) can be input with the digital switch (X20 to X2F) and the setting will be available by turning on X0.

Create the following program with GX Works2 filling in the blanks ⬚ .
Then, check the operation using the demonstration machine.



Hint



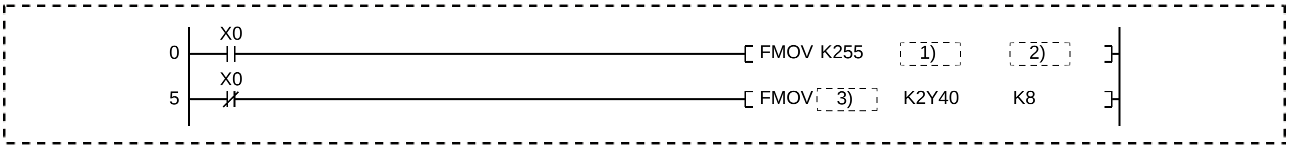
1) ______________________
2) ______________________
3) ______________________
4) ______________________

Project name	QTEST7
Program name	MAIN

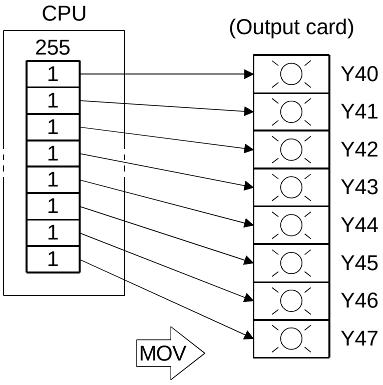
5.7.3 Exercise 3 FMOV

Create a program in which turning on X0 turns on the 64 outputs Y40 to Y7F and turning off X0 turns off the 64 outputs Y40 to Y7F.

Create the following program with GX Works2 filling in the blanks ⬚ .
Then, check the operation using the demonstration machine.



Hint



The constant is output from the CPU in the binary notation.

When 255 is output from Y40,

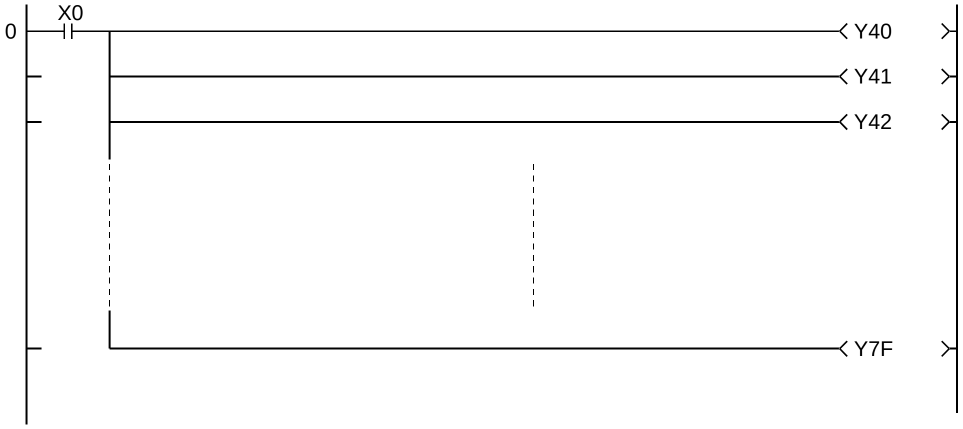
128	64	32	16	8	4	2	1	
1	1	1	1	1	1	1	1	=1+2+4+8+16+32+64+128=255
Y47	Y46	Y45	Y44	Y43	Y42	Y41	Y40	

In this exercise, a 64-point output module is used (Y40 to Y7F).
How many blocks are required for 255 on the output basis?

1) ____________________
2) ____________________
3) ____________________

Comparison

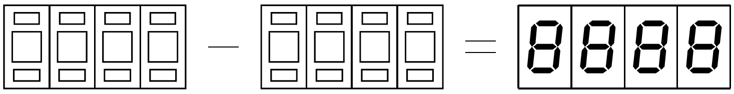
The following shows a program which is created with the sequence instructions, not with the FMOV instruction. The 130 steps are used.



Project name	QTEST8
Program name	MAIN

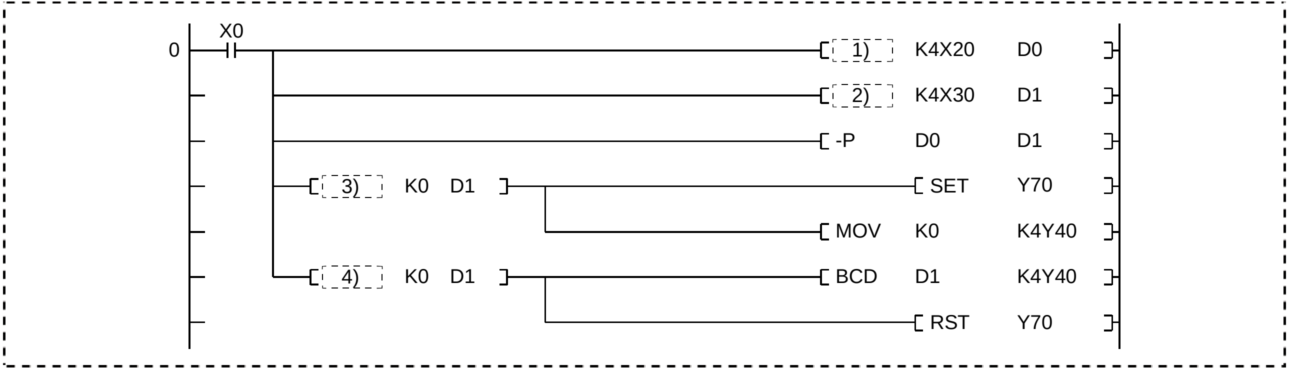
5.7.4 Exercise 4 Comparison instruction

Using the two BCD digital switches, execute the calculation of (A - B) and display the result on the BCD digital display (Y40 to Y4F).



Displays the result of the calculation of A - B on the BCD display of Y40 to Y4F.
When the result is a negative number, make sure that the display displays 0 and the LED of Y70 turns on.

Fill in the blanks ⬚ .
Then, check the operation using the demonstration machine.



Hint

The operation result is always output from the CPU in binary.

-	D0	D1	=D1-D0→D1

1) ____________________
2) ____________________
3) ____________________
4) ____________________

Project name	QTEST9
Program name	MAIN

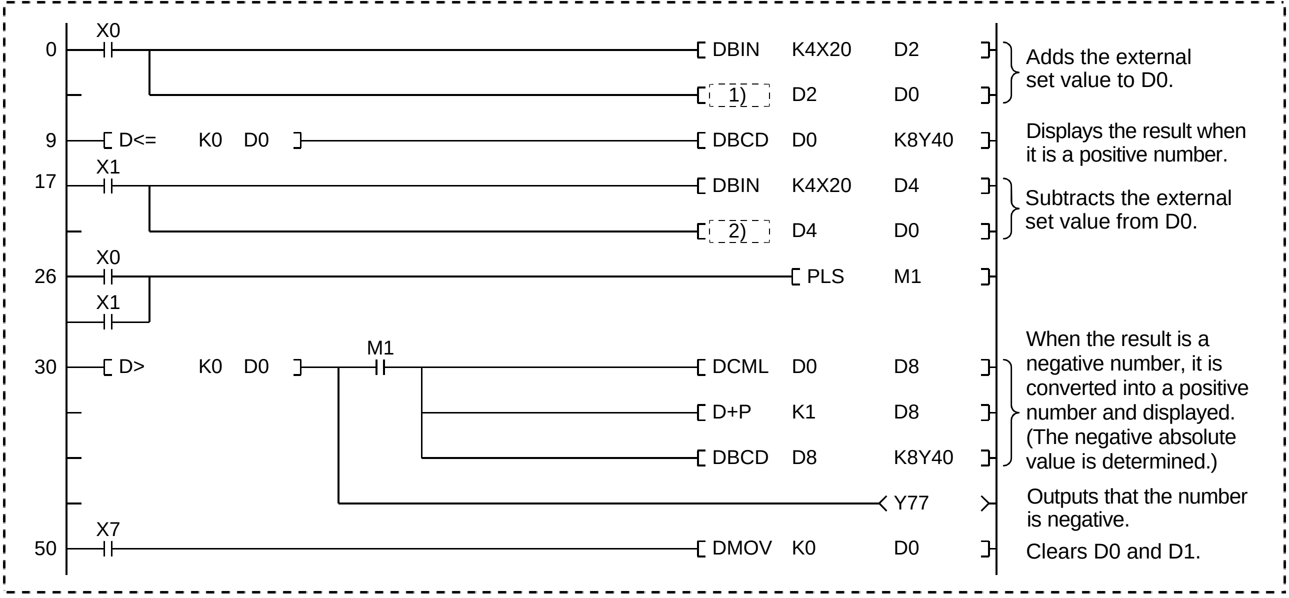
5.7.5 Exercise 5 Addition and subtraction instructions

Create a program that:

1) Imports the values specified by the digital switches (X20 to X2F) to D3 and D2 (32-bit data) when X0 is turned on, adds them to D1 and D0, and displays the result on the displays (Y40 to Y5F).
2) Imports the values specified by the digital switches (X20 to X2F) to D5 and D4 when X1 is turned on, subtracts them from D1 and D0, and displays the result.
3) When the result is a negative number, Y77 is turned on, the two's complement is determined from the result to obtain the absolute value, and displayed.

Fill in the blanks ⬚ in the following.

Then, check the operation using the demonstration machine.

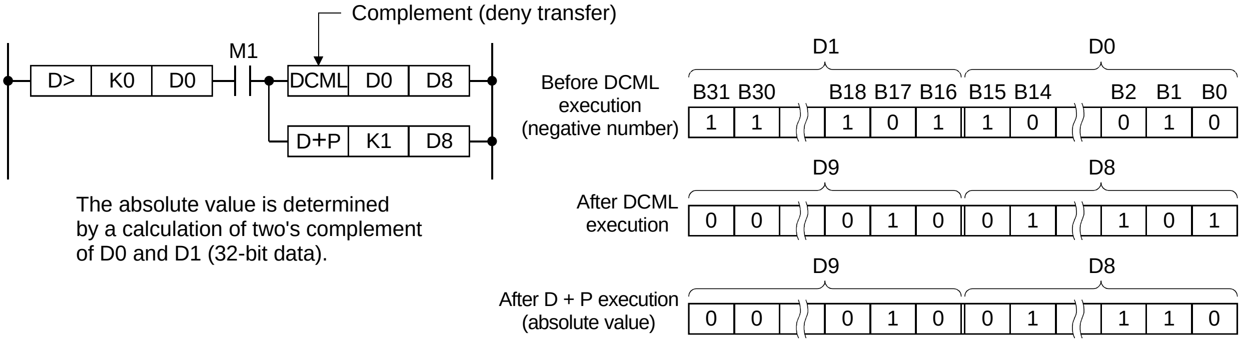


0
X0
DBIN K4X20 D2
1) D2 D0
Adds the external set value to D0.
9
D<= K0 D0
DBCD D0 K8Y40
Displays the result when it is a positive number.
17
X1
DBIN K4X20 D4
2) D4 D0
Subtracts the external set value from D0.
26
X0
X1
PLS M1
30
D> K0 D0
M1
DCML D0 D8
D+P K1 D8
DBCD D8 K8Y40
When the result is a negative number, it is converted into a positive number and displayed. (The negative absolute value is determined.)
Y77
Outputs that the number is negative.
50
X7
DMOV K0 D0
Clears D0 and D1.

1) ______________________

2) ______________________

Reference

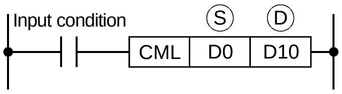


Complement (deny transfer)
M1
D> K0 D0
DCML D0 D8
D+P K1 D8
D1
D0
Before DCML execution (negative number)
B31 B30 B18 B17 B16 B15 B14 B2 B1 B0
1 1 1 0 1 1 0 0 1 0
D9
D8
After DCML execution
0 0 0 1 0 0 1 1 0 1
D9
D8
After D + P execution (absolute value)
0 0 0 1 0 0 1 1 1 0

The absolute value is determined by a calculation of two's complement of D0 and D1 (32-bit data).

REMARK

The CML instruction inverts the bit pattern of Ⓢ and transfers the data to Ⓓ when the input condition is turned on.



Project name	QTEST10
Program name	MAIN

5.7.6 Exercise 6 Multiplication and division instructions

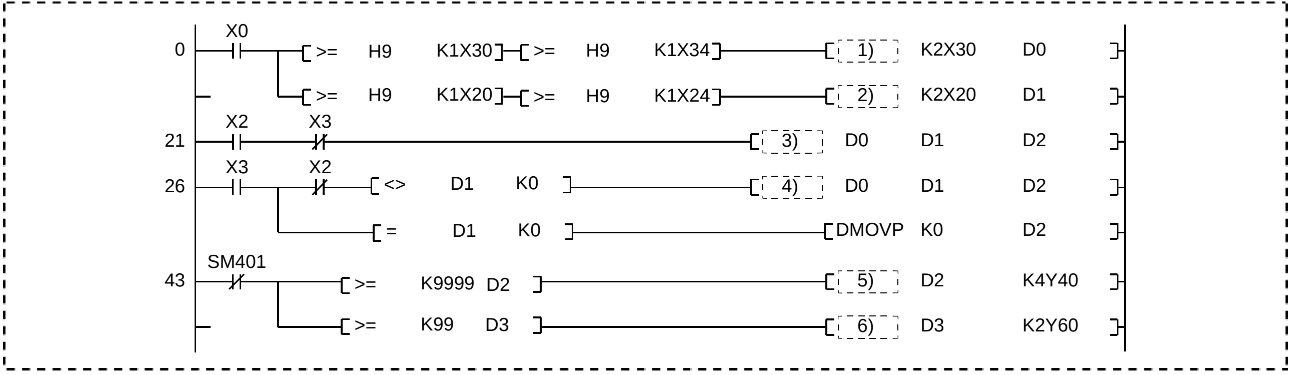
Create a program that:

1) Sets data for multiplication and division when X0 is turned on.
2) Multiplies the value specified by the digital switches X20 to X27 by the value specified by the digital switches X30 to X37 in binary when X2 is turned on.
3) Divides the value specified by the digital switches X30 to X37 by the value specified by the digital switches X20 to X27 in binary when X3 is turned on.
4) Outputs the result of the multiplication or division to the BCD displays Y40 to Y4F and the remainder to the BCD displays Y60 to Y67.

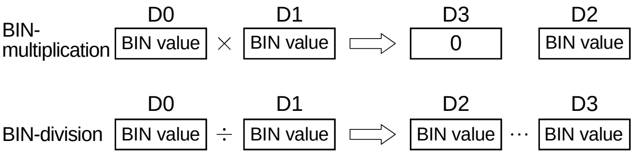
(X30 to X37) × (X20 to X27) ⇨ (Y40 to Y4F)

(X30 to X37) / (X20 to X27) ⇨ (Y40 to Y4F) ... (Y60 to Y67)

Create the program with GX Works2 filling in the blanks ⬚ in the following. Then, check the operation using the demonstration machine.



Hint



1) ____________________

2) ____________________

3) ____________________

4) ____________________

5) ____________________

6) ____________________

Project name	QTEST11
Program name	MAIN

5.7.7 Exercise 7 D-multiplication and D-division

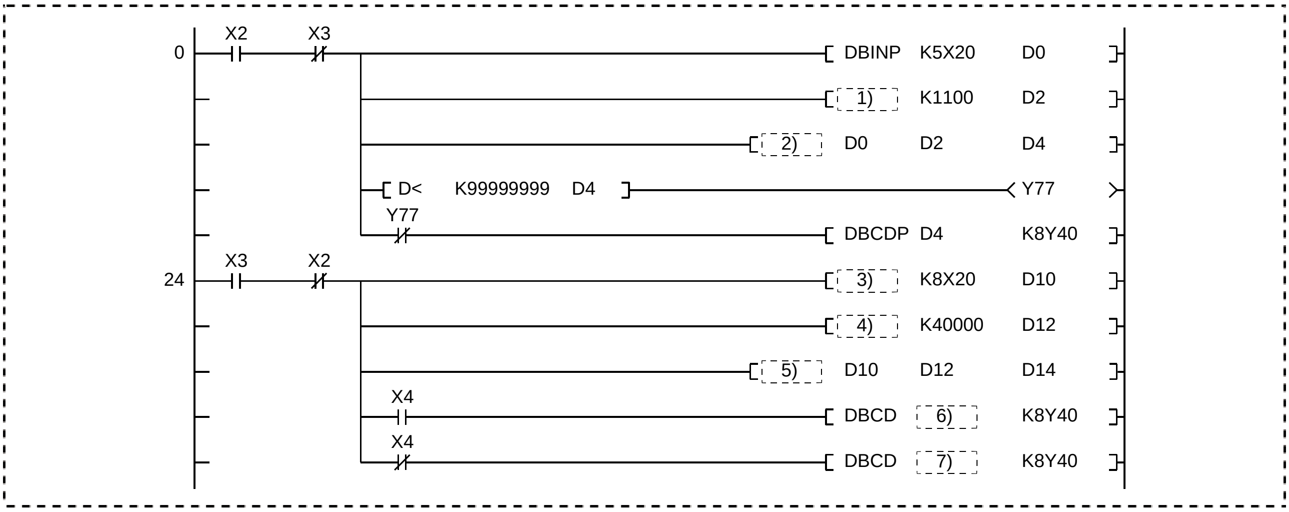
Create a program that:

1) Multiplies the value set by the 5-digit digital switches (X20 to X33) by 1,100 in binary when X2 is turned on. When the result is 99,999,999 or less, it is displayed on the displays (Y40 to Y5F).
2) Divides the value set by the 8-digit digital switches (X20 to X3F) by 40,000 in binary when X3 is turned on. When X4 is on, the quotient is displayed on the displays (Y40 to Y5F). When X4 is off, the remainder is displayed on the displays (Y40 to Y5F).

(X20 to X33) × 1100 ⇨ (Y40 to Y5F)

(X20 to X3F) / 40000 ⇨ Quotient (Y40 to Y5F) ... X4: ON
Remainder (Y40 to Y5F) ... X4: OFF

Create the program with GX Works2 filling in the blanks ⬚ in the following. Then, check the operation using the demonstration machine.



1) ____________________
2) ____________________
3) ____________________
4) ____________________
5) ____________________
6) ____________________
7) ____________________

Answers for the exercises in Chapter 5

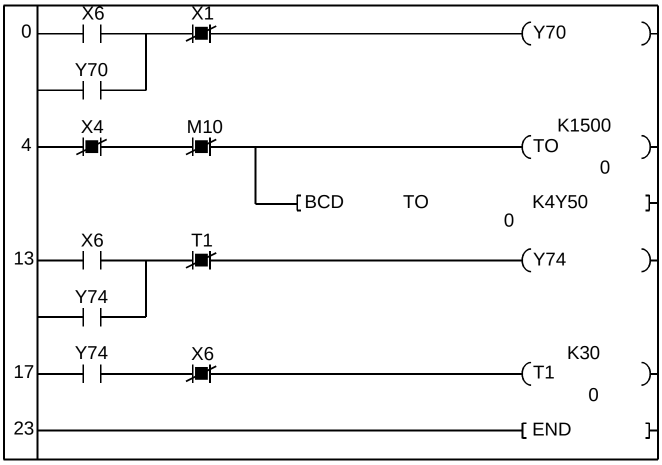
Exercise No.		Answer
1	1)	K2X0
	2)	K2Y70
2	1)	D0
	2)	K4X20
	3)	BCD
	4)	K4Y40
3	1)	K2Y40
	2)	K8
	3)	K0
4	1)	BINP
	2)	BINP
	3)	>
	4)	<=
5	1)	D + P
	2)	D - P
6	1)	BINP
	2)	BINP
	3)	*P
	4)	/P
	5)	BCD
	6)	BCD
7	1)	DMOVP
	2)	D*P
	3)	DBINP
	4)	DMOVP
	5)	D/P
	6)	D14
	7)	D16

CHAPTER 6 HOW TO USE OTHER FUNCTIONS

6.1 Test Function at Online

As a preparation, follow the procedure below.

Project name	QEX14
Program name	MAIN



For details on the operation method, refer to chapter 2.

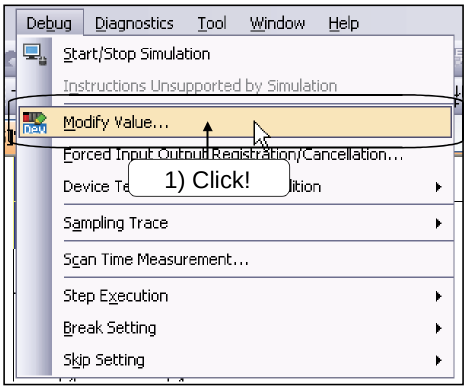
1) Read a project with GX Works2.

2) Write the parameter and program of the read project to the CPU (programmable controller). (The CPU must be stopped.)

3) Set GX Works2 to the monitor mode.

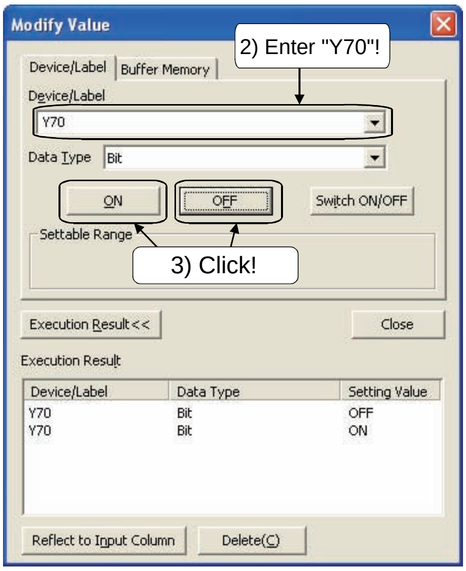
4) Confirm the program displayed on the screen.

6.1.1 Turning on and off the device "Y" forcibly



Stop the CPU before this operation.

1) Click [Debug] → [Modify Value].



Device/Label: Y70

Data Type: Bit

Execution Result

Device/Label	Data Type	Setting Value
Y70	Bit	OFF
Y70	Bit	ON

2) The Modify Value dialog box is displayed. Enter "Y70" in the "Device/Label" list box.

3) Click the ON or OFF button to turn on or off "Y70" forcibly.

Check with demonstration machine

1) Confirm that the on and off statuses on the Execution Result area switches according to the clicking of the ON or OFF button. Also, confirm that the LED of Y70 on the demonstration machine turns on and off according to the operation.

NOTE

When the CPU is in the RUN state, the operation results of programs are displayed preferentially. Therefore, stop the CPU first before the confirmation with the demonstration machine.

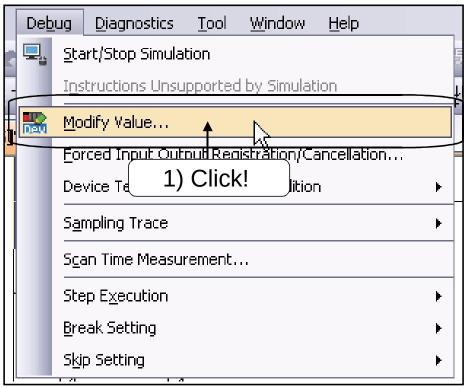
POINT

The test function during ladder monitoring of GX Works2 is also available for setting and resetting contacts, changing current values, and outputting forcibly word devices.

Double-clicking a contact (pressing the Enter key) holding the Shift key in the ladder monitoring screen of GX Works2 switches the contact open or close forcibly.

To display the Modify Value dialog box, double-click a word device (press the Enter key) holding the Shift key in the ladder monitoring screen of GX Works2.

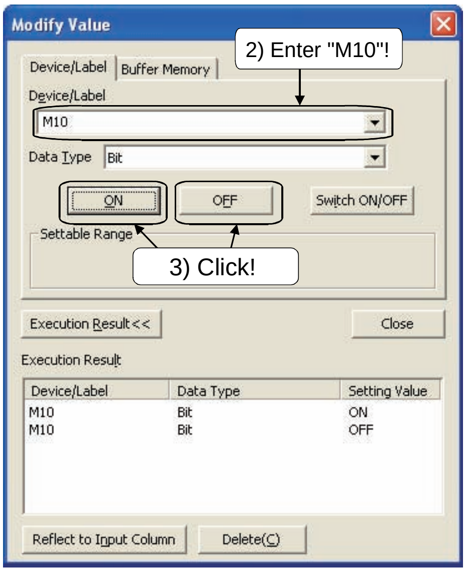
6.1.2 Setting and resetting the device "M"



1) Click!

Activate the CPU before this operation.

1) Click [Debug] → [Modify Value].



Device/Label: M10

Data Type: Bit

Execution Result

Device/Label	Data Type	Setting Value
M10	Bit	ON
M10	Bit	OFF

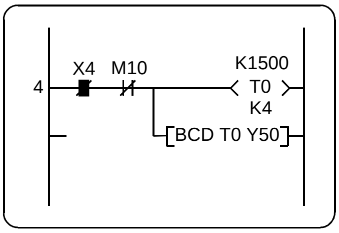
2) Enter "M10"!

3) Click!

2) The Modify Value dialog box is displayed. Enter "M10" in the "Device/Label" list box.

3) Click the ON or OFF button to set or reset "M10".

Check with demonstration machine



(Monitor screen when M10 is set)

Turn off X4 and check the following.

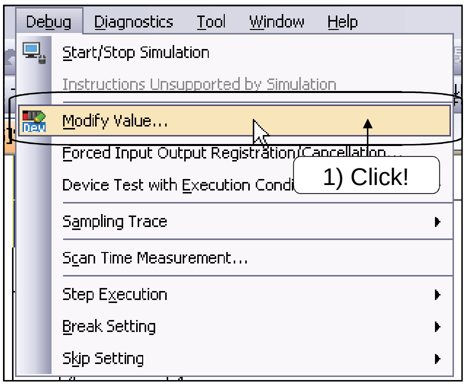
1) When M10 is set, $\overset{M10}{\nmid}$ becomes non-conductive and the current value of the timer T0 is cleared to 0.
Check that the value on the digital display (Y50-Y5F) does not change.

2) When M10 is reset, $\overset{M10}{\nmid}$ is conducted and the timer T0 starts counting from 0. This count value increases every 10 seconds.
Confirm that the value on the display (Y50-Y5F) increases every 10 seconds.

POINT

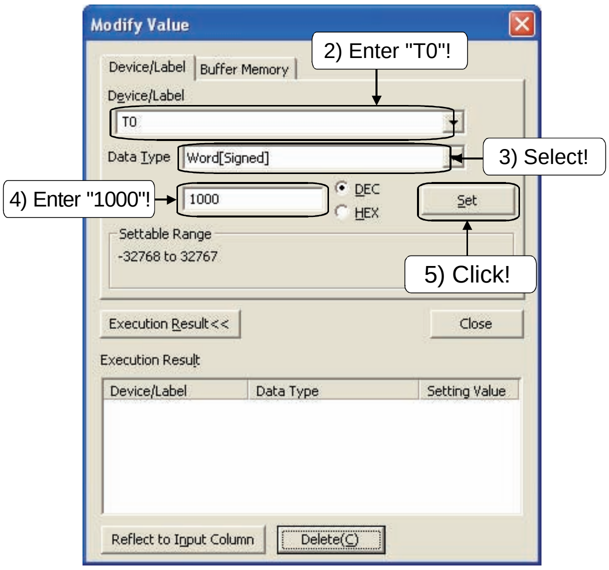
With the same procedure, bit devices other than the internal relay (M) also can be set or reset forcibly.

6.1.3 Changing the current value of the device "T"



Activate the CPU before this operation.

1) Click [Debug] → [Modify Value].



Device/Label: T0

Data Type: Word[Signed]

1000 ● DEC ○ HEX

Settable Range
-32768 to 32767

2) The Modify Value dialog box is displayed. Enter "T0" in the "Device/Label" list box.

3) Select "Word[Signed]" from the "Data Type" list box.

4) Enter "1000" in the "Value" column.

5) After the setting is completed, click the Set button to change the current value of T0 to 1000 forcibly.

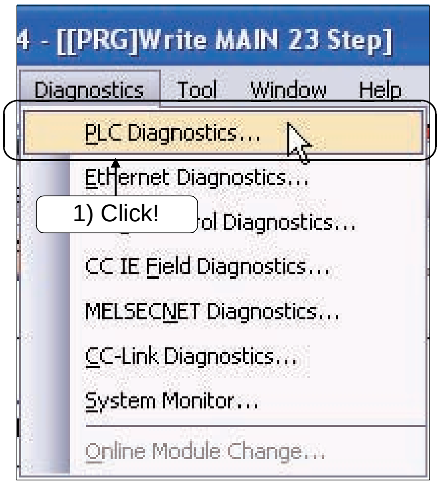
Check with demonstration machine

1) Confirm that the value on the digital display (Y50-Y5F) is 1000 when the ⏎ key is pressed.

POINT

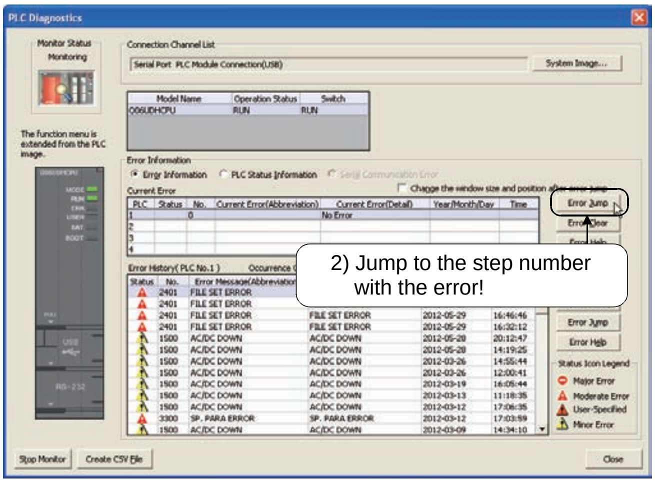
With the same procedure, the current values of word devices other than the timer (T) also can be changed.

6.1.4 Reading error steps



Activate the CPU before this operation.

1) Click [Diagnostics] → [PLC Diagnostics].



Connection Channel List: Serial Port PLC Module Connection(USB)

Model Name	Operation Status	Switch
Q06UDHCPU	RUN	RUN

The function menu is extended from the PLC image.

Error Information

● Error Information ○ PLC Status Information ○ Serial Communication Error

□ Change the window size and position after error jump

Current Error

PLC	Status	No.	Current Error(Abbreviation)	Current Error(Detail)	Year/Month/Day	Time
1		0		No Error		
2						
3						
4						

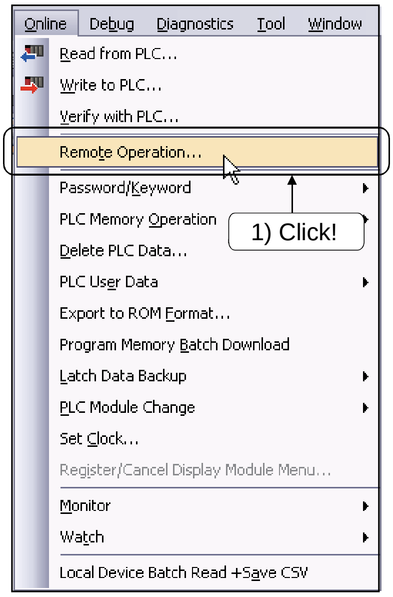
Error History(PLC No.1) Occurrence C

Status	No.	Error Message(Abbreviation)			
▲	2401	FILE SET ERROR			
▲	2401	FILE SET ERROR			
▲	2401	FILE SET ERROR	FILE SET ERROR	2012-05-29	16:46:46
▲	2401	FILE SET ERROR	FILE SET ERROR	2012-05-29	16:32:12
⚠	1500	AC/DC DOWN	AC/DC DOWN	2012-05-28	20:12:47
⚠	1500	AC/DC DOWN	AC/DC DOWN	2012-05-28	14:19:25
⚠	1500	AC/DC DOWN	AC/DC DOWN	2012-03-26	14:55:44
⚠	1500	AC/DC DOWN	AC/DC DOWN	2012-03-26	12:00:41
⚠	1500	AC/DC DOWN	AC/DC DOWN	2012-03-19	16:05:44
⚠	1500	AC/DC DOWN	AC/DC DOWN	2012-03-13	11:18:35
⚠	1500	AC/DC DOWN	AC/DC DOWN	2012-03-12	17:06:35
▲	3300	SP. PARA ERROR	SP. PARA ERROR	2012-03-12	17:03:59
⚠	1500	AC/DC DOWN	AC/DC DOWN	2012-03-09	14:34:10

2) The PLC Diagnostics dialog box is displayed. Click the Error JUMP button to jump to the highlighted sequence program step number where the selected error occurred.

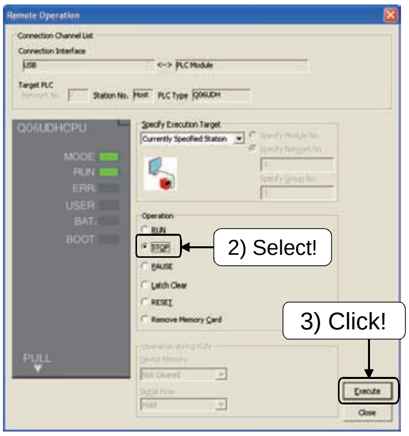
- An error number is displayed if an error occurred.
- "No Error" is displayed if no error occurred.

6.1.5 Remote STOP and RUN



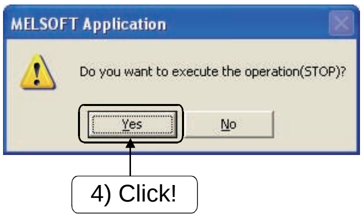
Activate the CPU before this operation.

1) Click [Online] → [Remote Operation].



2) The Remote Operation dialog box is displayed. Select "STOP" from the list in the Operation area.

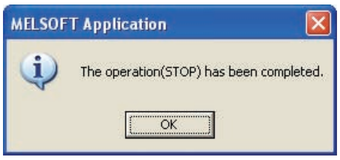
3) After the setting is completed, click the Execute button.



Do you want to execute the operation(STOP)?

4) The message "Do you want to execute the operation(STOP)?" is displayed. Click the Yes button.

The operation of the CPU stops.

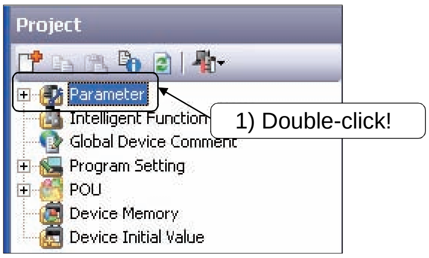


The operation(STOP) has been completed.

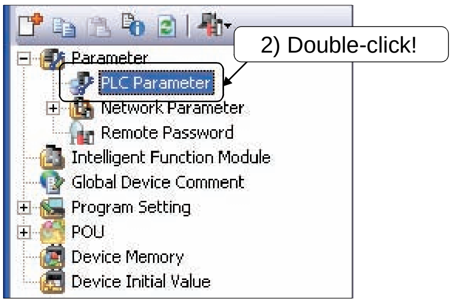
5) Select "RUN" in step 2), and perform steps 2) to 4) again.

The CPU, which was stopped in the above operation, starts the operation again.

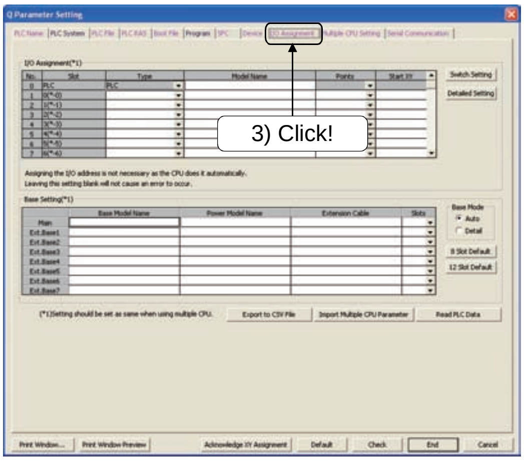
6.2 Forced I/O Assignment by Parameter Settings



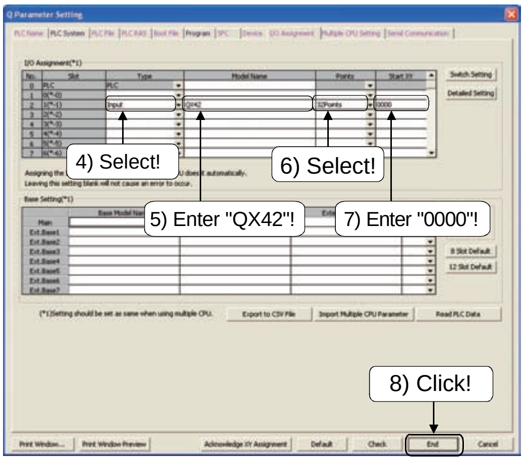
1) Double-click "Parameter" in the project list.



2) "PLC Parameter", the "Network Parameter" folder, and "Remote Password" are displayed. Double-click the "PLC Parameter".



3) The Q Parameter Setting dialog box is displayed. Click the "I/O Assignment" tab.



4) Select "Input" from the list box of the "Type" column.

5) Enter "QX42" in the "Model Name" column.

6) Select "32Points" from the list box of "Points" column.

7) Enter "0000" in the "Start XY" column.

8) After the setting is completed, click the End button.

After this exercise is finished, initialize the settings by the following procedure.

1) Click the Default button in the Q Parameter Setting dialog box to initialize the parameter settings.

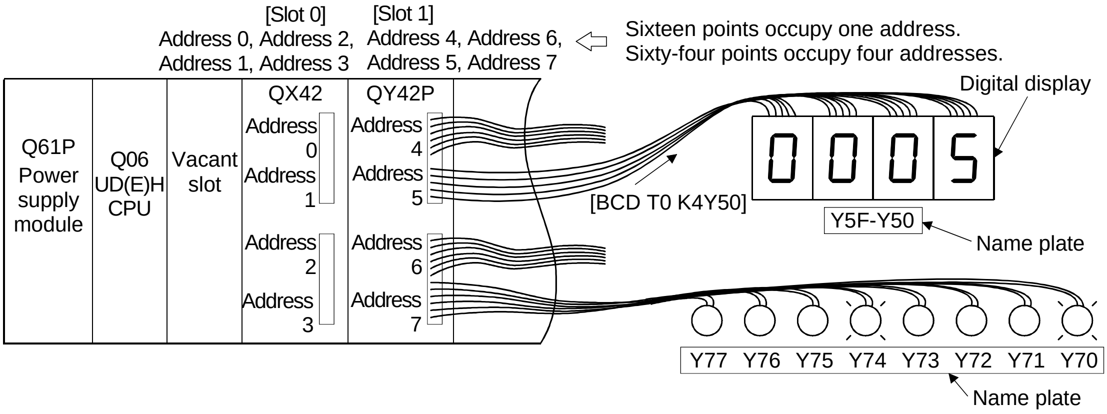
2) Click on the toolbar and write only the parameters to the CPU.

Check with demonstration machine

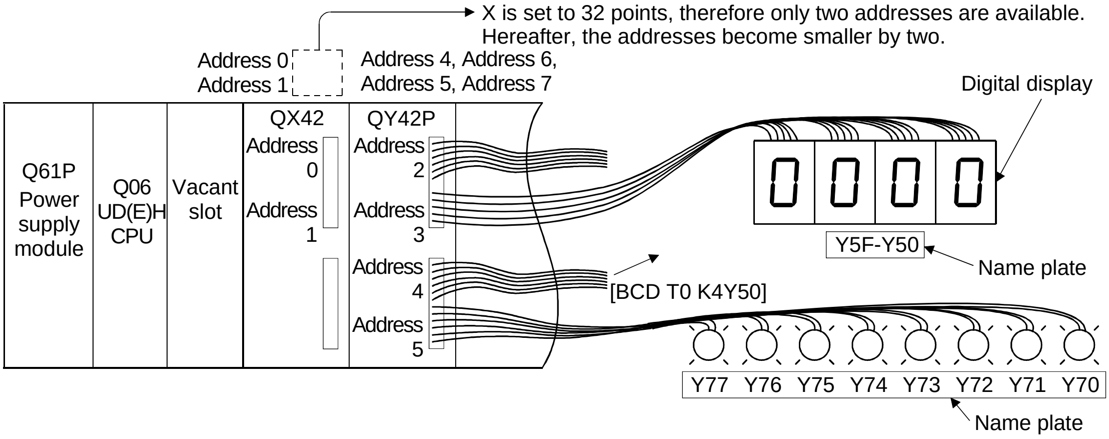
Stop the CPU and click on the toolbar.
The Online Data Operation dialog box is displayed. Click the parameter of the currently edited data, and click the Execute button to write only the parameters to the CPU. Then, activate the CPU and check the following.

1) The current value of the timer T0 disappears from the digital display (Y50 to Y5F). Then, the LEDs of Y70 to Y77 start flashing until the set values of Y70 to Y77 reach each set device value.
2) Turning on X6 to output the signal to Y70 and Y74 does not turn the LEDs of Y70 and Y74.

[I/O numbers before the forced assignment]



[I/O numbers after the forced assignment]



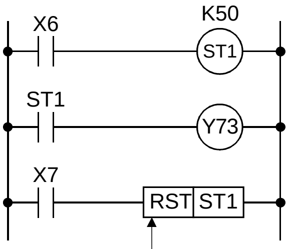
POINT

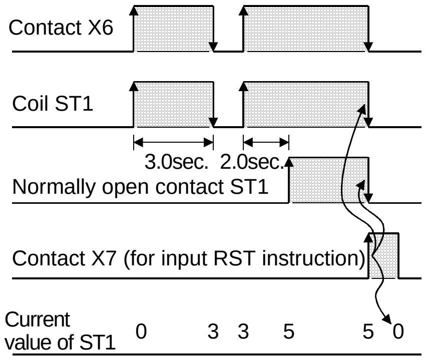
- The address 7 is replaced with the address 5. Therefore, the current value of the timer T0 is output to the newly assigned address 5, and LEDs of Y70 to Y77 which are connected to the address 5 flash.
- Results of outputting the signals to Y70 or Y74 are not displayed on any displays since the address 7 of the output modules no longer exists.
- To display the device numbers normally, change the device number K4Y50 ⇨ K4Y30, and Y70 to Y77 ⇨ Y50 to Y57.

6.3 How to Use Retentive Timers

When an input condition is turned on, the coil is energized. Then the value of a retentive timer starts increasing. When the current value reaches the set value, the retentive timer goes time-out and the contact turns on. When the input condition is turned off during the increasing, the coil is de-energized but the current value is kept. When the input condition is turned on again, the coil is re-energized and the current value is accumulated.

Project name	Retentive timer
Program name	MAIN

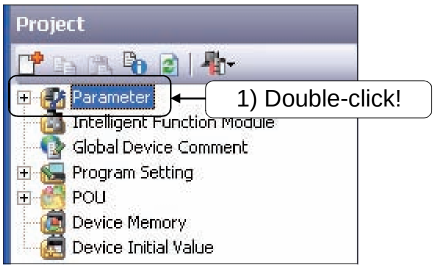




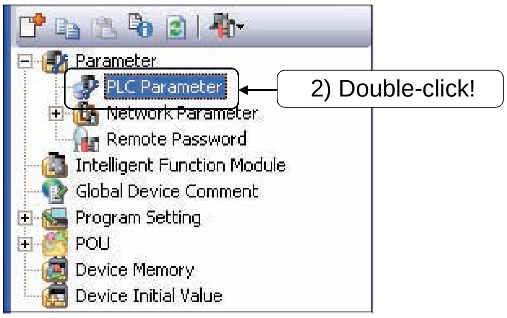
When using the program as a retentive timer, specify the points in parameters in advance.

Only the RST instruction is available for turning off the contact and clearing the current value after the retentive timer goes time-out.

In the example below, the retentive timer is set to ST0 to ST31.



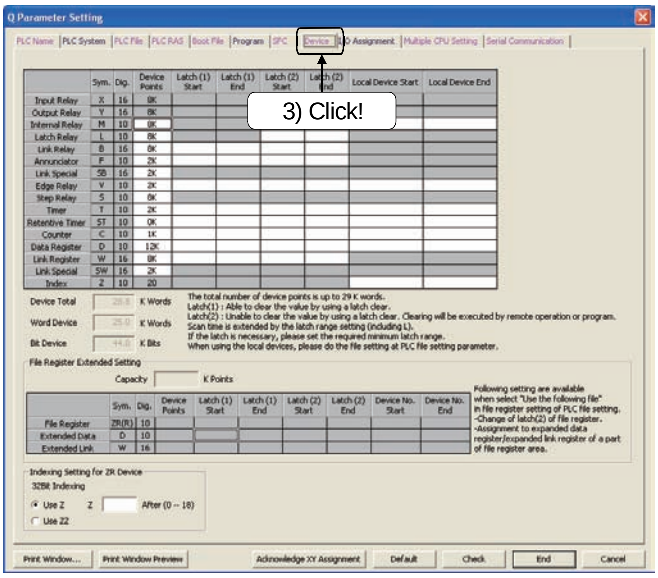
1) Double-click "Parameter" in the project list.



2) "PLC Parameter", the "Network Parameter" folder, and "Remote Password" are displayed. Double-click the "PLC Parameter".

(To the next page)

(From the previous page)



Q Parameter Setting

PLC Name | PLC System | PLC File | PLC RAS | Boot File | Program | SFC | Device | I/O Assignment | Multiple CPU Setting | Serial Communication

	Sym.	Dig.	Device Points	Latch (1) Start	Latch (1) End	Latch (2) Start	Latch (2) End	Local Device Start	Local Device End
Input Relay	X	16	8K						
Output Relay	Y	16	8K						
Internal Relay	M	10	8K						
Latch Relay	L	10	8K						
Link Relay	B	16	8K						
Annunciator	F	10	2K						
Link Special	SB	16	2K						
Edge Relay	V	10	2K						
Step Relay	S	10	8K						
Timer	T	10	2K						
Retentive Timer	ST	10	0K						
Counter	C	10	1K						
Data Register	D	10	12K						
Link Register	W	16	8K						
Link Special	SW	16	2K						
Index	Z	10	20						

Device Total 28.8 K Words

Word Device 25.0 K Words

Bit Device 44.0 K Bits

The total number of device points is up to 29 K words.
Latch(1) : Able to clear the value by using a latch clear.
Latch(2) : Unable to clear the value by using a latch clear. Clearing will be executed by remote operation or program.
Scan time is extended by the latch range setting (including L).
If the latch is necessary, please set the required minimum latch range.
When using the local devices, please do the file setting at PLC file setting parameter.

File Register Extended Setting

Capacity K Points

	Sym.	Dig.	Device Points	Latch (1) Start	Latch (1) End	Latch (2) Start	Latch (2) End	Device No. Start	Device No. End
File Register	ZR(R)	10							
Extended Data	D	10							
Extended Link	W	16							

Following setting are available when select "Use the following file" in file register setting of PLC file setting.
-Change of latch(2) of file register.
-Assignment to expanded data register/expanded link register of a part of file register area.

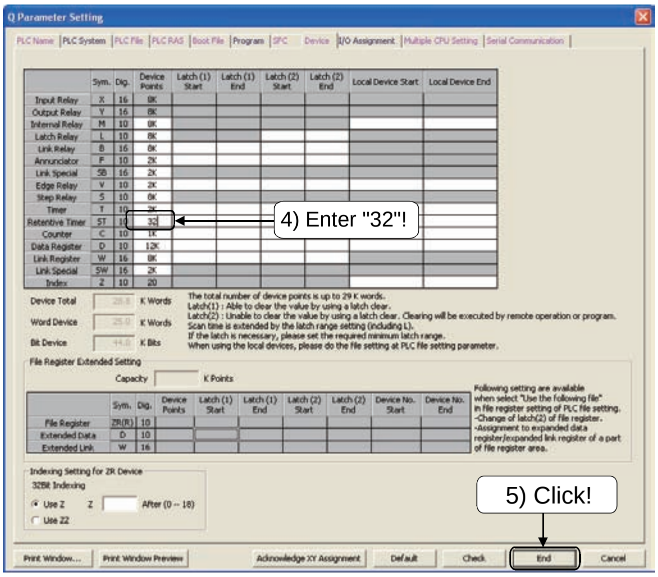
Indexing Setting for ZR Device

32Bit Indexing

● Use Z Z After (0 -- 18)

○ Use ZZ

3) The Q Parameter Setting dialog box is displayed. Click the "Device" tab.



Q Parameter Setting

PLC Name | PLC System | PLC File | PLC RAS | Boot File | Program | SFC | Device | I/O Assignment | Multiple CPU Setting | Serial Communication

	Sym.	Dig.	Device Points	Latch (1) Start	Latch (1) End	Latch (2) Start	Latch (2) End	Local Device Start	Local Device End
Input Relay	X	16	8K						
Output Relay	Y	16	8K						
Internal Relay	M	10	8K						
Latch Relay	L	10	8K						
Link Relay	B	16	8K						
Annunciator	F	10	2K						
Link Special	SB	16	2K						
Edge Relay	V	10	2K						
Step Relay	S	10	8K						
Timer	T	10	2K						
Retentive Timer	ST	10	32						
Counter	C	10	1K						
Data Register	D	10	12K						
Link Register	W	16	8K						
Link Special	SW	16	2K						
Index	Z	10	20						

Device Total 28.8 K Words

Word Device 25.0 K Words

Bit Device 44.0 K Bits

The total number of device points is up to 29 K words.
Latch(1) : Able to clear the value by using a latch clear.
Latch(2) : Unable to clear the value by using a latch clear. Clearing will be executed by remote operation or program.
Scan time is extended by the latch range setting (including L).
If the latch is necessary, please set the required minimum latch range.
When using the local devices, please do the file setting at PLC file setting parameter.

File Register Extended Setting

Capacity K Points

	Sym.	Dig.	Device Points	Latch (1) Start	Latch (1) End	Latch (2) Start	Latch (2) End	Device No. Start	Device No. End
File Register	ZR(R)	10							
Extended Data	D	10							
Extended Link	W	16							

Following setting are available when select "Use the following file" in file register setting of PLC file setting.
-Change of latch(2) of file register.
-Assignment to expanded data register/expanded link register of a part of file register area.

Indexing Setting for ZR Device

32Bit Indexing

● Use Z Z After (0 -- 18)

○ Use ZZ

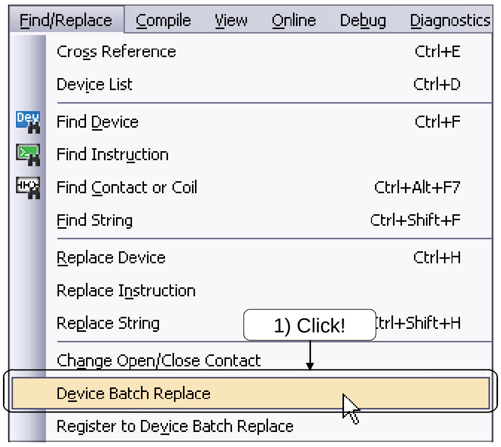
4) Click "Device Points" in the "Retentive Timer" row, and enter "32".

5) After the setting is completed, click the End button.

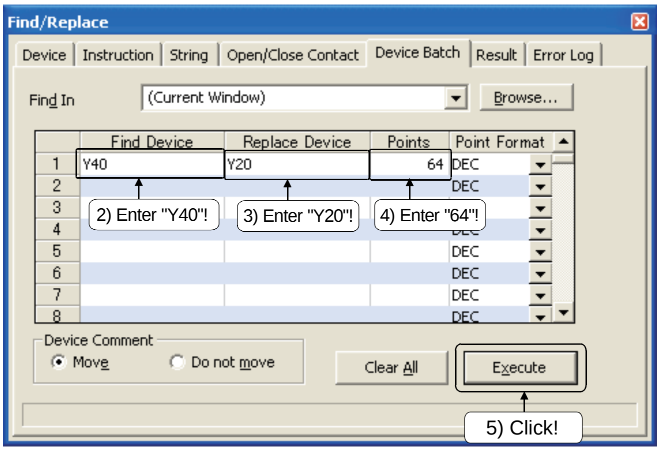
6.4 Device Batch Replacement

6.4.1 Batch replacement of device numbers

This section explains how to replace Y40 to Y7F (64 devices) with Y20 to Y5F (64 devices) in batch.



1) Click [Find/Replace] → [Device Batch Replace].



Find In: (Current Window)

	Find Device	Replace Device	Points	Point Format
1	Y40	Y20	64	DEC
2				DEC
3				DEC
4				DEC
5				DEC
6				DEC
7				DEC
8				DEC

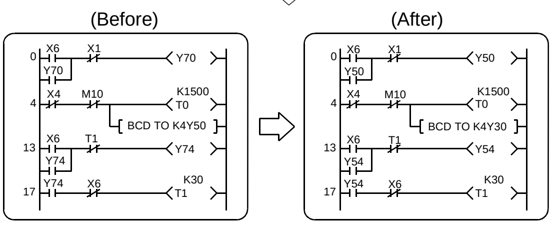
Device Comment: ● Move ○ Do not move

2) The Find/Replace dialog box is displayed. Enter "Y40" in the "Find Device" column.

3) Enter "Y20" in the "Replace Device" column.

4) Enter "64" in the "Points" column.

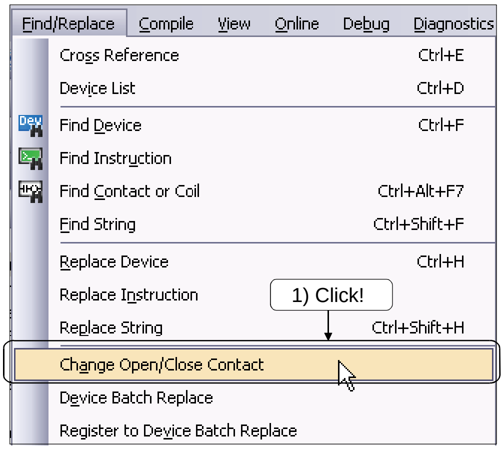
5) After the setting is completed, click the Execute button.



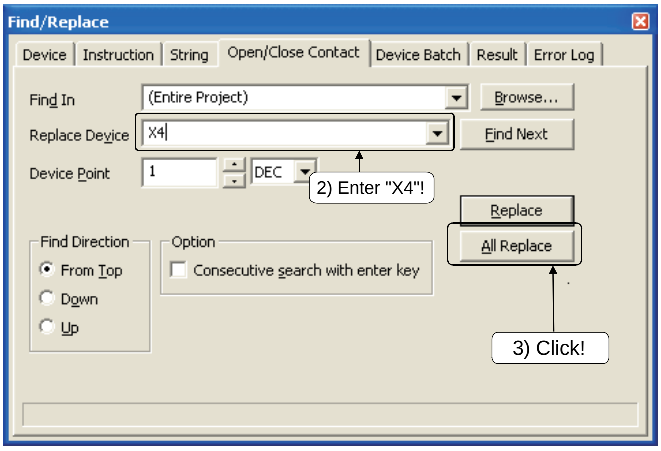
6) Confirm that the target device numbers are replaced.

6.4.2 Batch change of specified devices between normally open contacts and normally closed contacts

This section explains how to change the normally open contacts of the specified devices to the normally closed contact and vice versa in batch.

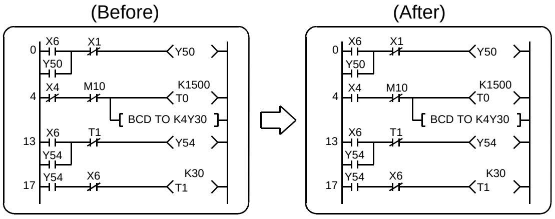


1) Click [Find/Replace] → [Change Open/Close Contact].



2) The Find/Replace dialog box is displayed. Enter "X4" in the "Replace Device" list box.

3) After the setting is completed, click the All Replace button.



4) Confirm that the normally open contact is changed to the normally closed contact and vice versa.

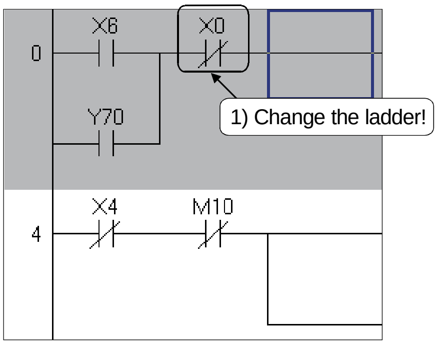
NOTE

Before exercising section 6.5 after this section, write the program in the personal computer to the CPU.
For the write operation, refer to section 2.7.

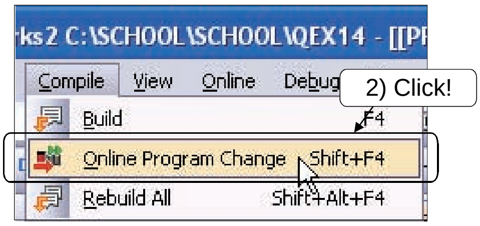
6.5 Online Program Change

This function is used to write programs to the CPU that is running.

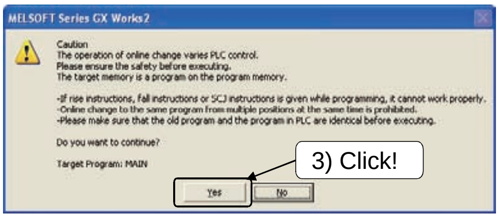
Activate the CPU before this operation.



1) Change the ladder.
(In the example, change "X1" to "X0".)


ks2 C:\SCHOOL\SCHOOL\QEX14 - [[PI
Compile
View
Online
Debug
2) Click!
Build F4
Online Program Change Shift+F4
Rebuild All Shift+Alt+F4

2) After the change, click [Compile]
→ [Online Program Change].


MELSOFT Series GX Works2

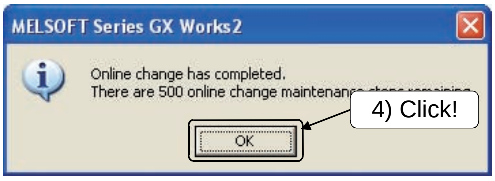
Caution
The operation of online change varies PLC control.
Please ensure the safety before executing.
The target memory is a program on the program memory.

-If rise instructions, fall instructions or SCJ instructions is given while programming, it cannot work properly.
-Online change to the same program from multiple positions at the same time is prohibited.
-Please make sure that the old program and the program in PLC are identical before executing.

Do you want to continue?

Target Program: MAIN

3) The dialog box for "Caution" is displayed. Click the Yes button to accept the change.


MELSOFT Series GX Works2

Online change has completed.
There are 500 online change maintenan

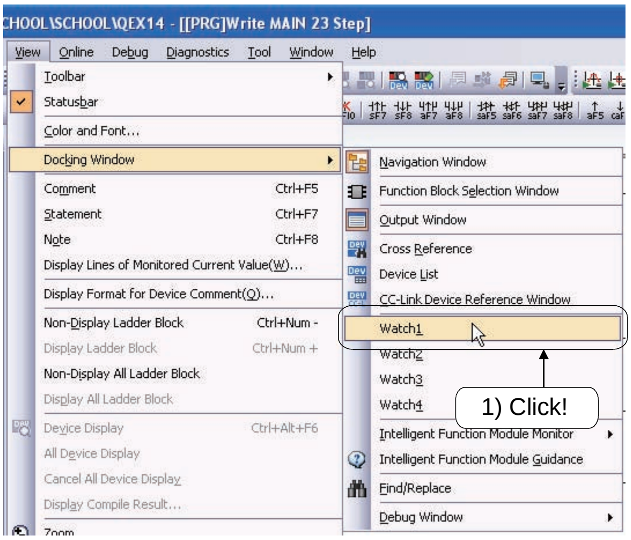
4) The message "Online change has completed." is displayed. Click the OK button.

NOTE

Online program change cannot be executed when the program in the programmable controller CPU and the program in GX Works2 before the modification do not match. Therefore, when whether the programs match or not is unclear, verify them before the modification with GX Works2, and execute the online program change.

6.6 Registering Devices

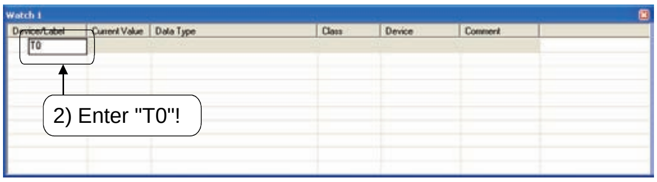
This section explains how to register multiple devices or labels in one screen and to monitor them at the same time.


CHOOL\SCHOOL\QEX14 - [[PRG]Write MAIN 23 Step]
View Online Debug Diagnostics Tool Window Help
Toolbar
Statusbar
Color and Font...
Docking Window
Comment Ctrl+F5
Statement Ctrl+F7
Note Ctrl+F8
Display Lines of Monitored Current Value(W)...
Display Format for Device Comment(O)...
Non-Display Ladder Block Ctrl+Num -
Display Ladder Block Ctrl+Num +
Non-Display All Ladder Block
Display All Ladder Block
Device Display Ctrl+Alt+F6
All Device Display
Cancel All Device Display
Display Compile Result...
Zoom
Navigation Window
Function Block Selection Window
Output Window
Cross Reference
Device List
CC-Link Device Reference Window
Watch1
Watch2
Watch3
Watch4
Intelligent Function Module Monitor
Intelligent Function Module Guidance
Find/Replace
Debug Window

1) Click!

1) Click [View] → [Docking Window] → [Watch(1 to 4)].

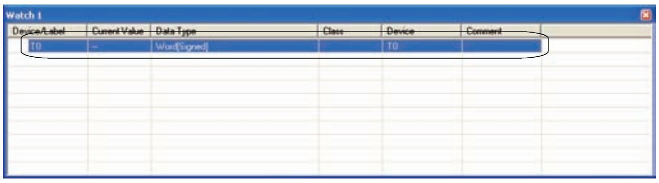
* In this example, select "1".



Device/Label	Current Value	Data Type	Class	Device	Comment
T0					

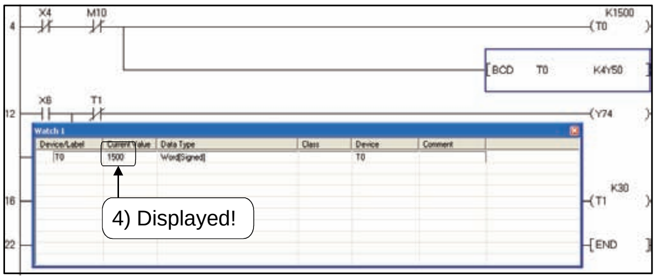
2) Enter "T0"!

2) The Watch 1 window is displayed. Select a row to be edited. Enter "T0" in the Device Label column.



Device/Label	Current Value	Data Type	Class	Device	Comment
T0	-	Word[Signed]		T0	

3) The input device or label is registered.



Device/Label	Current Value	Data Type	Class	Device	Comment
T0	1500	Word[Signed]		T0	

4) Displayed!

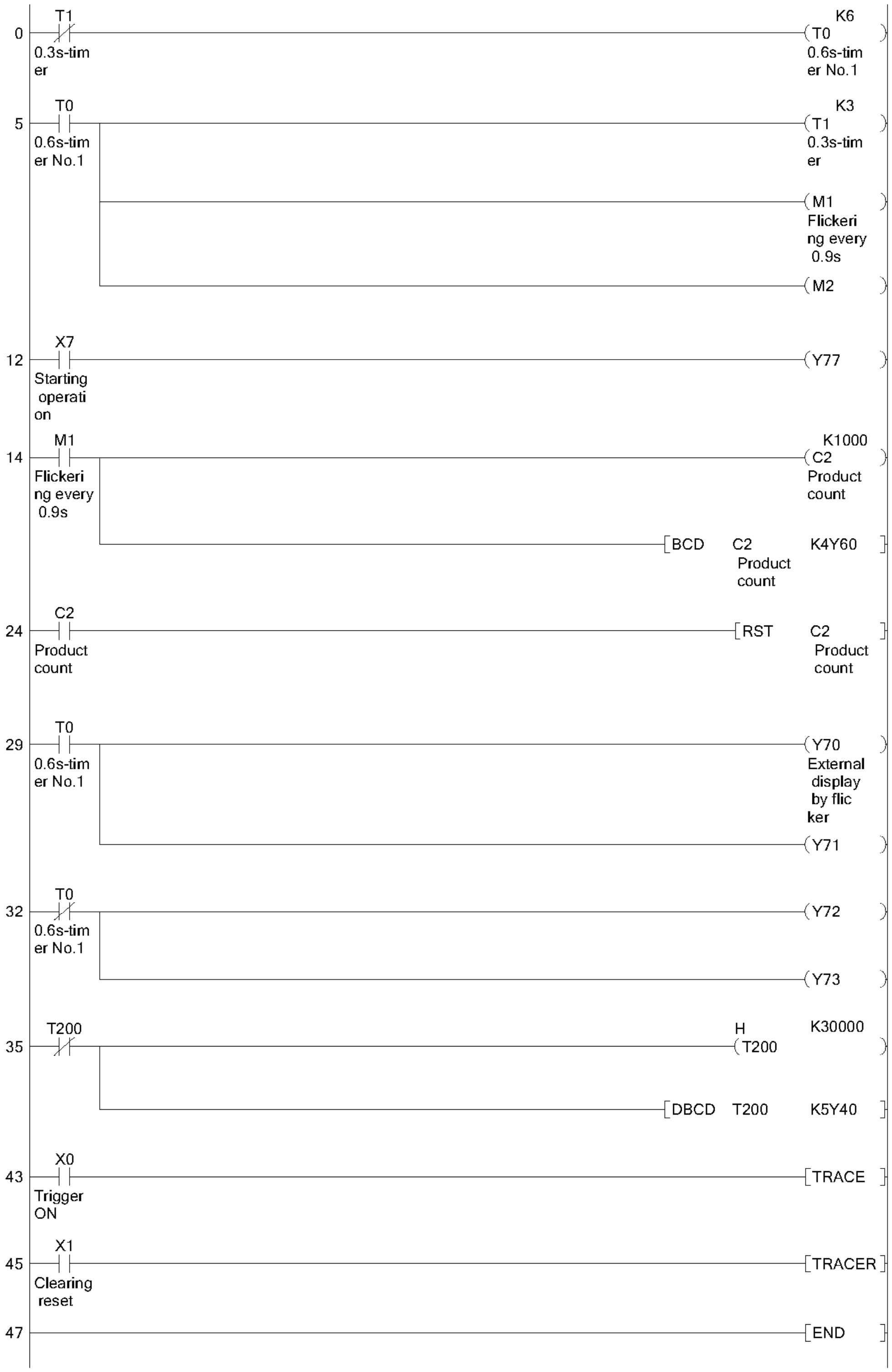
4) Click [Online] → [Watch] → [Start Watching].

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

6.7 How to Create Comments

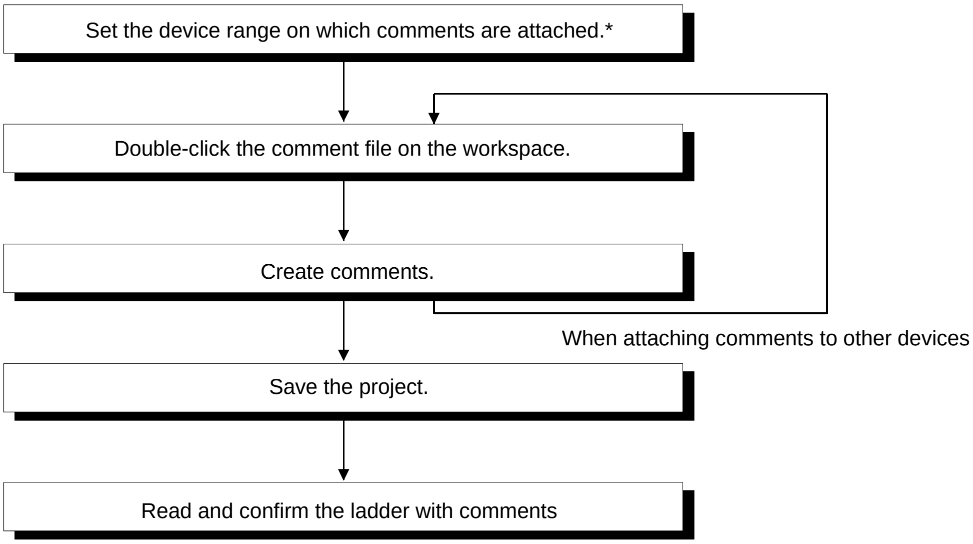
Project name	QEX15
Program name	MAIN

The following is an example of a printed out ladder with comments.



Use the keyboard to input the program above or read it from a folder on the desktop.

(1) Flowchart of when creating comments

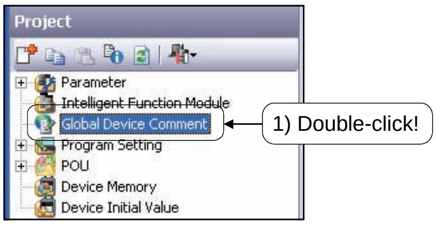

Set the device range on which comments are attached.*
Double-click the comment file on the workspace.
Create comments.
When attaching comments to other devices
Save the project.
Read and confirm the ladder with comments

*: This procedure is necessary for specifying the device comment range.

POINT

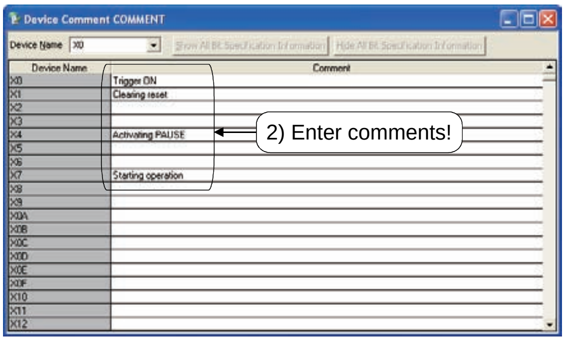
Comments are used for displaying functions or applications of each device.
Up to 32 characters are available.

(2) Creating comments



1) Double-click!

1) Double-click "Global Device Comment" in the project list. The Device Comment screen is displayed.

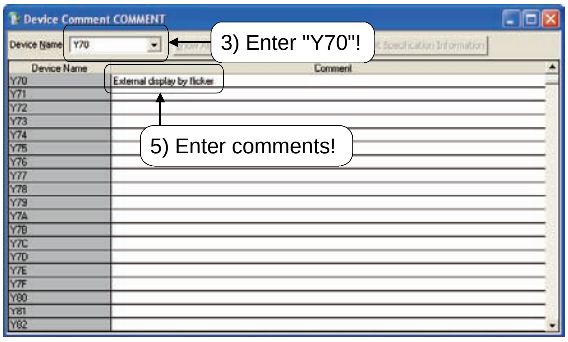

Device Comment COMMENT
Show All Bit Specification Information
Hide All Bit Specification Information

Device Name: X0

Device Name	Comment
X0	Trigger ON
X1	Clearing reset
X2	
X3	
X4	Activating PAUSE
X5	
X6	
X7	Starting operation
X8	
X9	
X0A	
X0B	
X0C	
X0D	
X0E	
X0F	
X10	
X11	
X12	

2) Enter comments!

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



Device Name: Y70

3) Enter "Y70"!

Device Name	Comment
Y70	External display by flicker
Y71	
Y72	
Y73	
Y74	
Y75	
Y76	
Y77	
Y78	
Y79	
Y7A	
Y7B	
Y7C	
Y7D	
Y7E	
Y7F	
Y80	
Y81	
Y82	

5) Enter comments!

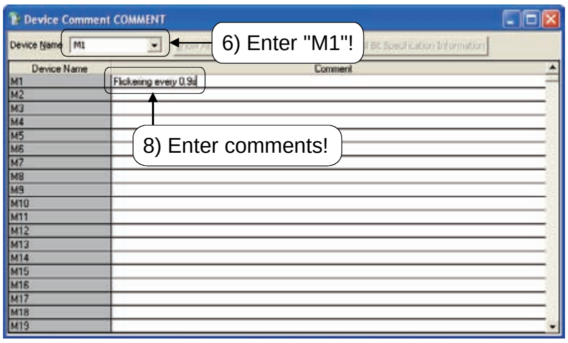
3) Enter "Y70" in the "Device Name" list box.

4) Press the Enter key.

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

(To the next page)

(From the previous page)



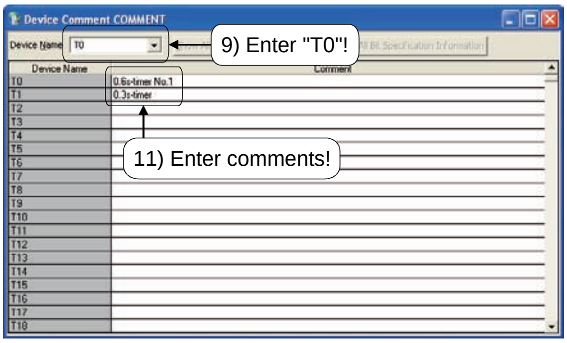
Device Name: M1

Device Name	Comment
M1	Flickering every 0.9s
M2	
M3	
M4	
M5	
M6	
M7	
M8	
M9	
M10	
M11	
M12	
M13	
M14	
M15	
M16	
M17	
M18	
M19	

6) Enter "M1" in the "Device Name" list box.

7) Press the Enter key.

8) Click a comment area and enter a comment as shown left.



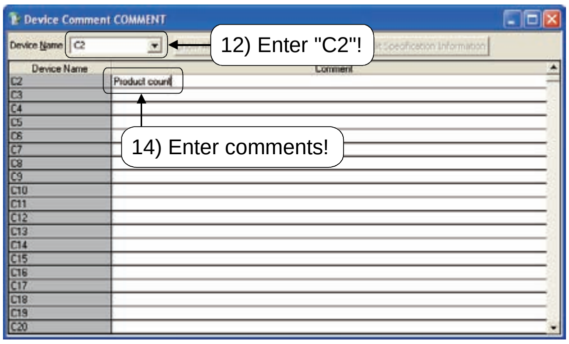
Device Name: T0

Device Name	Comment
T0	0.6s-timer No.1
T1	0.3s-timer
T2	
T3	
T4	
T5	
T6	
T7	
T8	
T9	
T10	
T11	
T12	
T13	
T14	
T15	
T16	
T17	
T18	

9) Enter "T0" in the "Device Name" list box.

10) Press the Enter key.

11) Click a comment area and enter a comment as shown left.



Device Name: C2

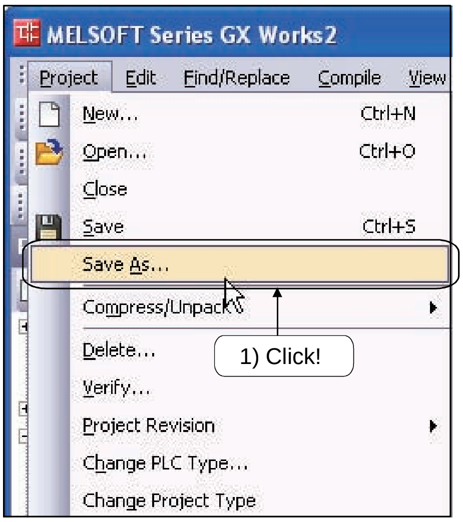
Device Name	Comment
C2	Product count
C3	
C4	
C5	
C6	
C7	
C8	
C9	
C10	
C11	
C12	
C13	
C14	
C15	
C16	
C17	
C18	
C19	
C20	

12) Enter "C2" in the "Device name" list box.

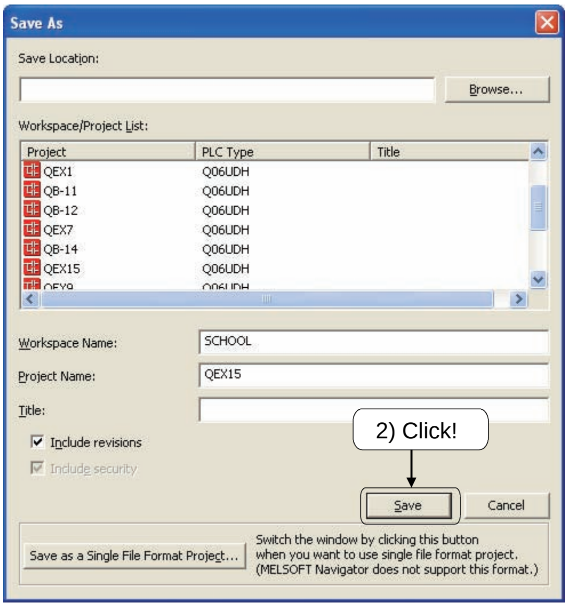
13) Press the Enter key.

14) Click a comment area and enter a comment as shown left.

(3) Saving comments



1) Click [Project] → [Save As].


Save As

Save Location:

Workspace/Project List:

Project	PLC Type	Title
QEX1	Q06UDH	
QB-11	Q06UDH	
QB-12	Q06UDH	
QEX7	Q06UDH	
QB-14	Q06UDH	
QEX15	Q06UDH	
QEX9	Q06UDH	

Workspace Name: SCHOOL

Project Name: QEX15

Title:

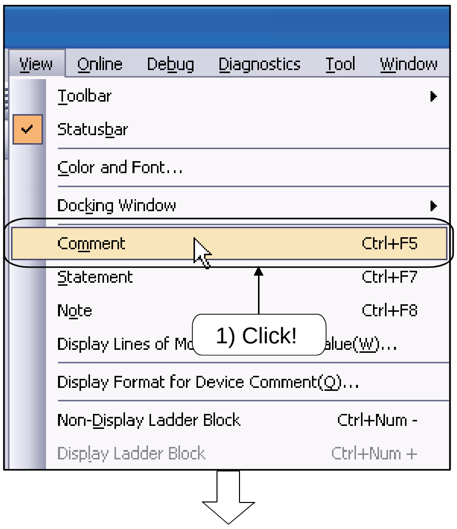
☑ Include revisions

☑ Include security

Switch the window by clicking this button when you want to use single file format project. (MELSOFT Navigator does not support this format.)

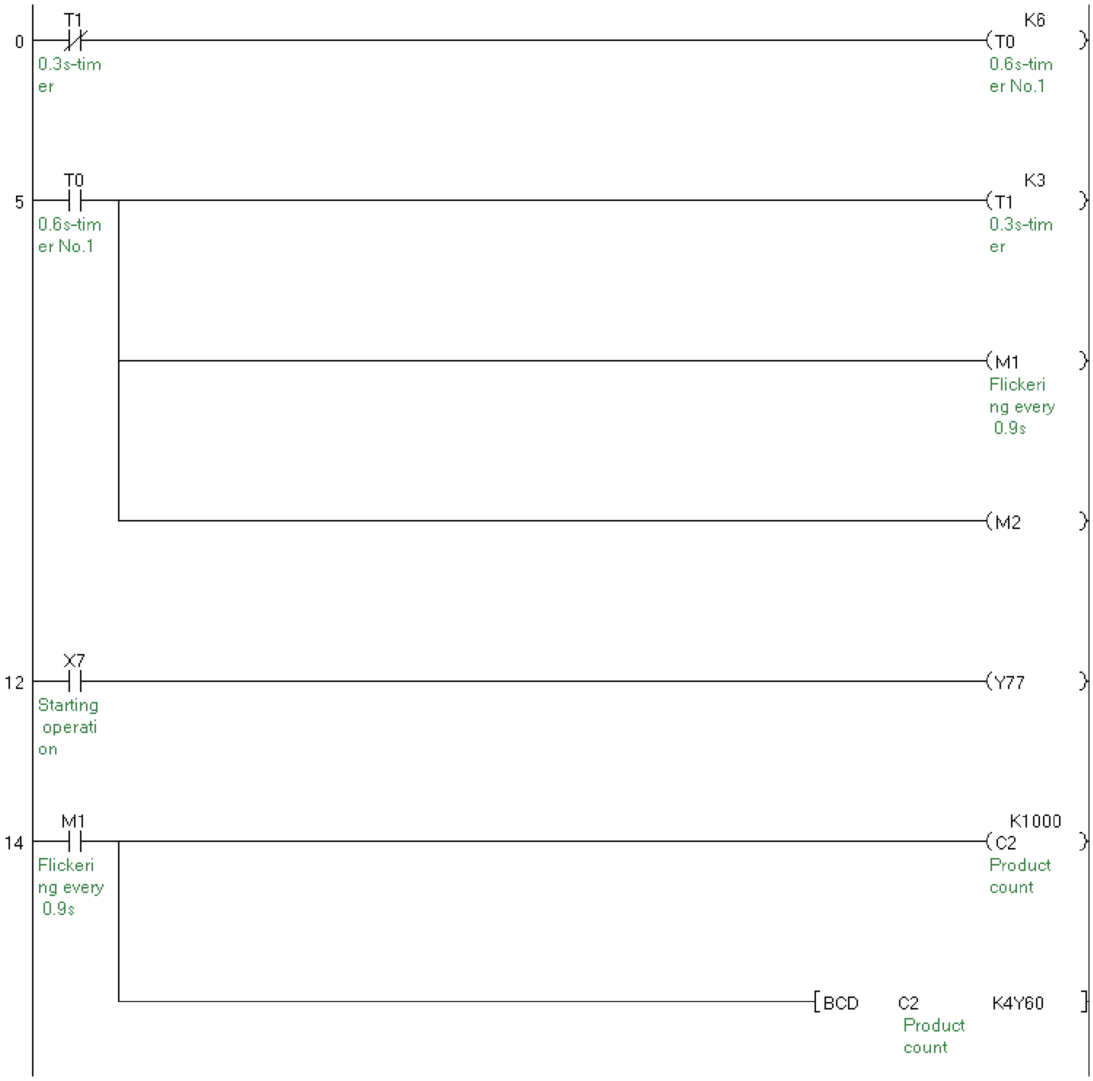
2) The Save As dialog box is displayed. Specify (or select) a workspace name and click the Save button.

(4) Displaying a ladder with comments on GX Works2 screens


View
Online
Debug
Diagnostics
Tool
Window
Toolbar
Statusbar
Color and Font...
Docking Window
Comment Ctrl+F5
Statement Ctrl+F7
Note Ctrl+F8
Display Lines of Mo...alue(W)...
Display Format for Device Comment(O)...
Non-Display Ladder Block Ctrl+Num -
Display Ladder Block Ctrl+Num +
1) Click!

1) Click [View] → [Comment].

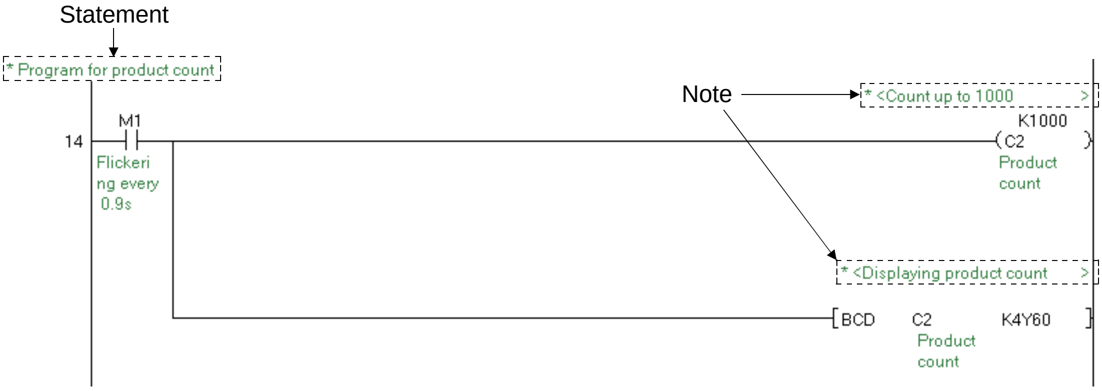
2) Comments are displayed on the ladder screen.



POINT

In addition to device comments, statements and notes can be created on the ladder screen.

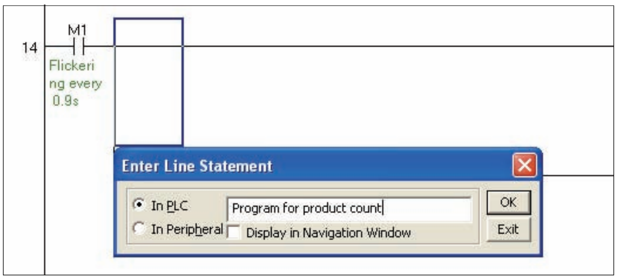
- Statement : Comment for explaining functions or applications for the ladder block. Up to 64 characters are available.
- Note : Comment for explaining functions or applications for outputs and commands. Up to 32 characters are available.



- Creating statements

  Click and double-click a symbol where a comment is to be attached.

  The Enter Line Statements dialog box is displayed. Enter a comment and click the OK button.



Enter Line Statement

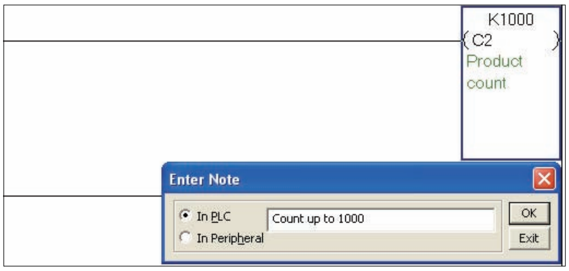
● In PLC Program for product count

○ In Peripheral □ Display in Navigation Window

- Creating notes

  Click and double-click an output or a command where a comment is to be attached.

  The Enter Symbol dialog box is displayed. Enter a comment and click the OK button.



Enter Note

● In PLC Count up to 1000

○ In Peripheral

- Select "In PLC" or "In Peripheral" for statements and notes.

  "In PLC" : The data of statements and notes is stored as a part of a program. This enables the data to be stored in CPUs at factories. However, a lot of program memory capacity of the programmable controller CPU is required.

  "In Peripheral" : The data of statements and notes is stored in the peripheral device (personal computer) separated from the program. Since a program requires only one extra step per one location, less program memory capacity is required on the programmable controller CPUs. However, when programs are modified at factories, programs in GX Works2 in the peripheral device (personal computer) and those in programmable controller CPUs do not match. Carefully handle the data.

6.8 Setting Security for Projects

This section explains how to set security for projects to protect the projects and the data in the projects.
Setting security restricts accesses to projects.
Also, setting security prevents data such as POUs, device comments, and parameters, which are created by the user, from erroneous modifications or disclosures to unauthorized users.

POINT

Access levels and access authority
Setting an access level to each user restricts accesses to each data.
An access level is an operating authority given to a login user of the project.
The following five levels are available as the access levels. Data that can be edited by a user having lower access level can also be edited by a user having higher access level.

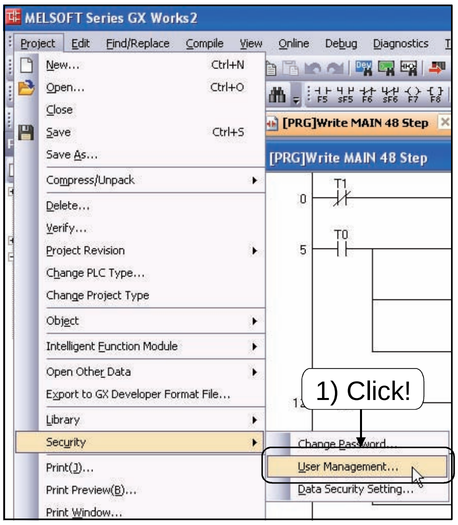
Access level		Operating authority
Higher	Administrators	<Administrator level> All operations are possible.
↕	Developers (Level 3)	<Developer level> Security setting, data access, and a part of operations are restricted.
	Developers (Level 2)	
	Developers (Level 1)	
Lower	Users	<Operator level> Only access to project data is possible. Data cannot be read from the programmable controller CPU.

<Example>
The data with access authority of Developers(Level 2) can be edited by login users with the access level of Developers(Level 2) or higher (Administrators, Developers(Level 3), or Developers(Level 2)).

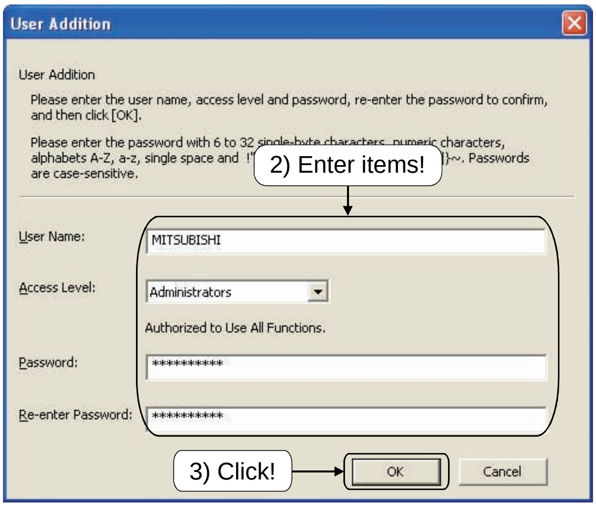
6.8.1 Setting and resetting security for projects

This section explains how to set security for an open project and how to reset the security.

(1) Setting security for projects
Set a security for a project.
Once security is set for a project, user authentication is required when the project is opened again.



1) Click [Project] → [Security] → [User Management].



User Addition

Please enter the user name, access level and password, re-enter the password to confirm, and then click [OK].

Please enter the password with 6 to 32 single-byte characters, numeric characters, alphabets A-Z, a-z, single space and !" }~. Passwords are case-sensitive.

User Name: MITSUBISHI

Access Level: Administrators

Authorized to Use All Functions.

Password: **********

Re-enter Password: **********

2) The Use Addition dialog box is displayed.
Enter the following items.

User Name	: MITSUBISHI
Password	: MITSUBISHI
Re-enter Password	: MITSUBISHI

* When the user name or login password is lost, logging in to the project is disabled. Do not enter any other user name or password other than the above.

3) After entering them, click the OK button.

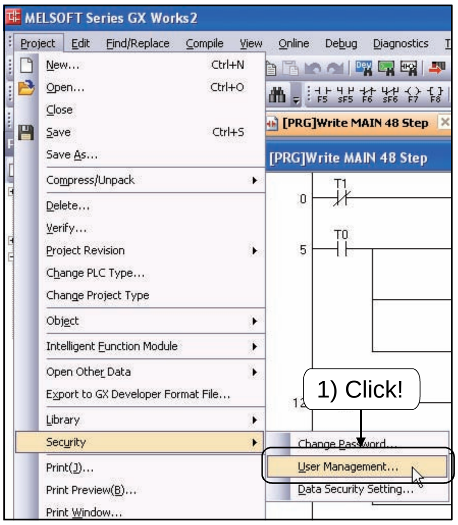
Security is set for the project.

(2) Resetting security for projects
Deleting all users resets the set security of a project and returns the project to the status without security. (Refer to section 6.8.2.)

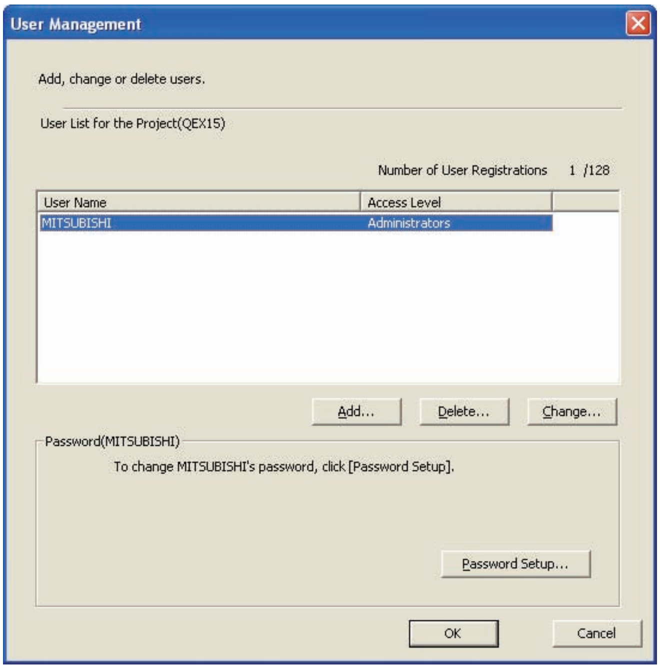
6.8.2 Managing (adding, deleting, and changing) users

This section explains how to manage the registered statuses of users for a project with security and how to add, delete, and change users.
This function is available only when a user logs in a project with the access level of "Administrators" or "Developers".

[Displaying the User Management screen]



1) Click [Project] → [Security] → [User Management].


User Management

Add, change or delete users.

User List for the Project(QEX15)

Number of User Registrations 1 /128

User Name	Access Level
MITSUBISHI	Administrators

Add... Delete... Change...

Password(MITSUBISHI)

To change MITSUBISHI's password, click [Password Setup].

Password Setup...

OK Cancel

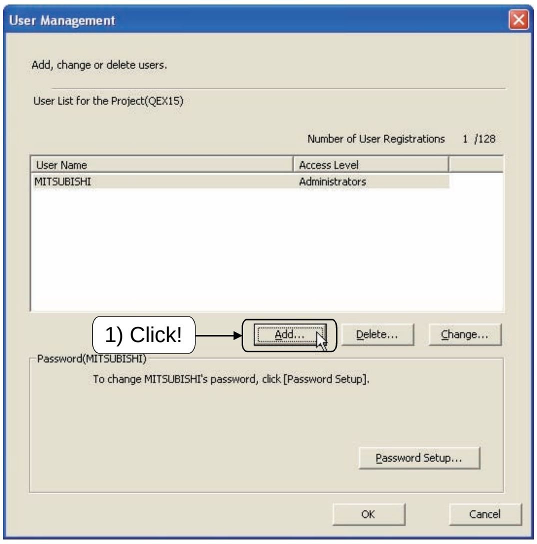
2) The User Management dialog box is displayed.

The methods for adding users, changing user information, changing passwords, and deleting users are explained from the next page.

[Adding users]

Add a user to a project with security.

A user whose access level is higher than that of the login user cannot be added.



User Management

Add, change or delete users.

User List for the Project(QEX15)

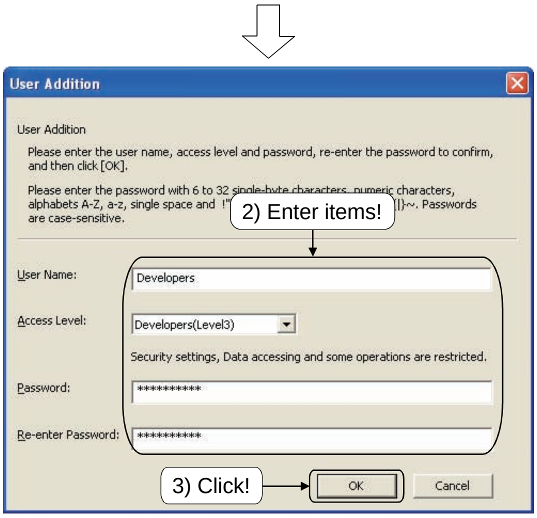
Number of User Registrations 1 /128

User Name	Access Level
MITSUBISHI	Administrators

Password(MITSUBISHI)

To change MITSUBISHI's password, click [Password Setup].

1) Click the Add... button on the User Management screen.



User Addition

User Addition

Please enter the user name, access level and password, re-enter the password to confirm, and then click [OK].

Please enter the password with 6 to 32 single-byte characters, numeric characters, alphabets A-Z, a-z, single space and !" {|}~. Passwords are case-sensitive.

User Name: Developers

Access Level: Developers(Level3)

Security settings, Data accessing and some operations are restricted.

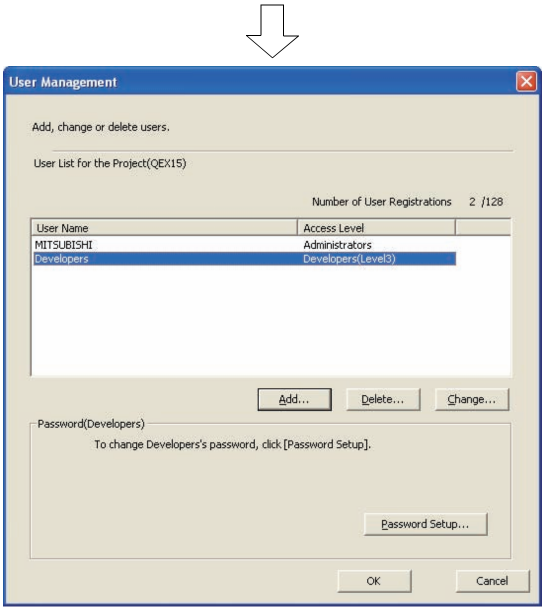
Password: **********

Re-enter Password: **********

2) The User Addition dialog box is displayed. Enter the following items.

User Name	: Developers
Access Level	: Developers(Level3)
Password	: Developers
Re-enter Password	: Developers

3) After entering them, click the OK button.



User Management

Add, change or delete users.

User List for the Project(QEX15)

Number of User Registrations 2 /128

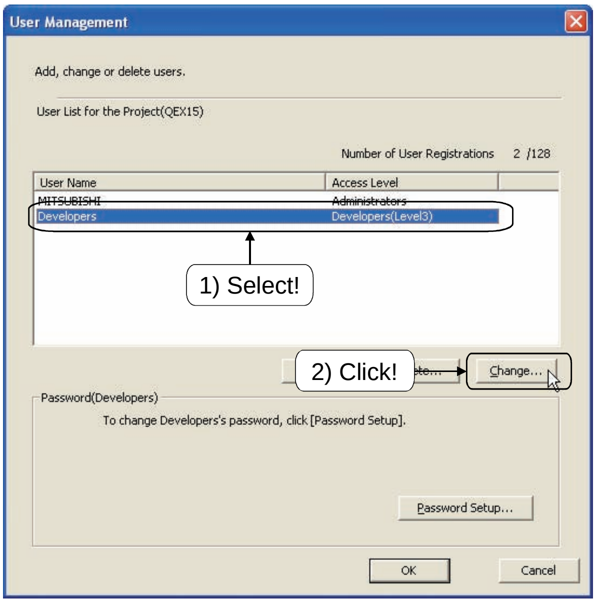
User Name	Access Level
MITSUBISHI	Administrators
Developers	Developers(Level3)

Password(Developers)

To change Developers's password, click [Password Setup].

4) The user (Developers(Level3)) is added.

[Changing user information]
Change the access level of the user added on the previous page from "Developers(Level3)" to "Users".
The information of the login user and of a user whose access level is higher than that of the login user cannot be changed.


User Management

Add, change or delete users.

User List for the Project(QEX15)

Number of User Registrations 2 /128

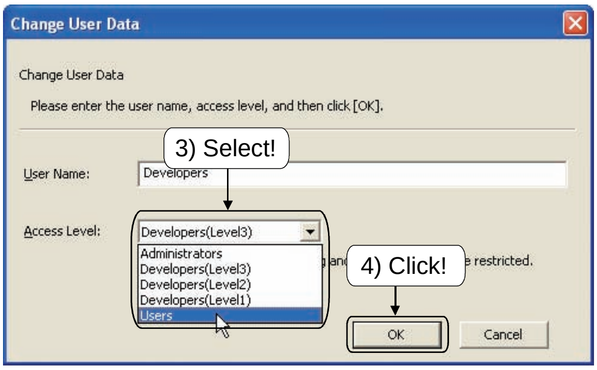
User Name	Access Level
MITSUBISHI	Administrators
Developers	Developers(Level3)

Password(Developers)

To change Developers's password, click [Password Setup].

1) Select the user name "Developers".

2) Click the Change button.


Change User Data

Change User Data

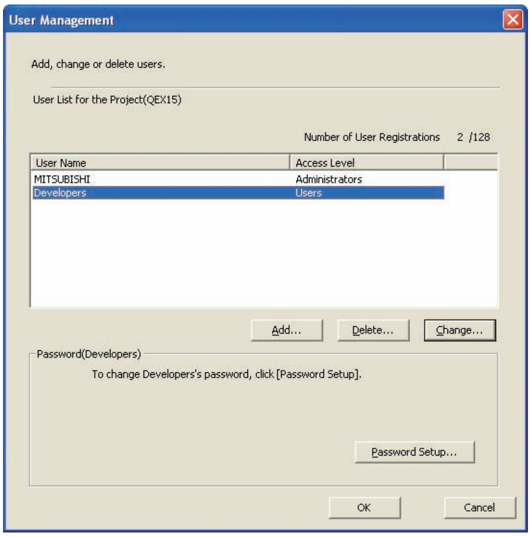
Please enter the user name, access level, and then click [OK].

User Name: Developers

Access Level: Developers(Level3)

3) The Change User Data dialog box is displayed. Select "Users" from the "Access Level" list box.

4) After selecting it, click the OK button.


User Management

Add, change or delete users.

User List for the Project(QEX15)

Number of User Registrations 2 /128

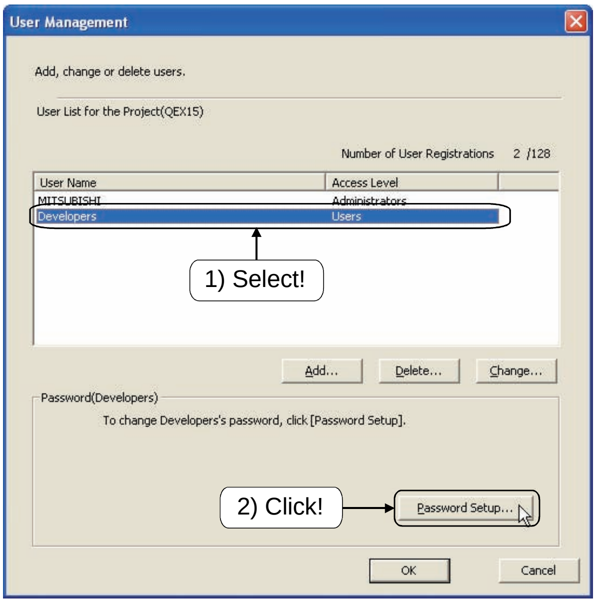
User Name	Access Level
MITSUBISHI	Administrators
Developers	Users

Password(Developers)

To change Developers's password, click [Password Setup].

5) The access level of the user "Developers" is changed.

[Changing passwords]
Change the password of a user selected in the list on the User Management screen. The password of the login user and of a user whose access level is higher than that of the login user cannot be changed.
To change the password of the login user, click [Project] → [Security] → [Change Password].



User Management

Add, change or delete users.

User List for the Project(QEX15)

Number of User Registrations 2 /128

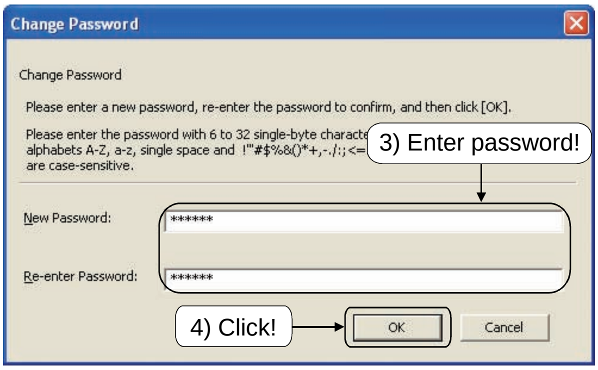
User Name	Access Level
MITSUBISHI	Administrators
Developers	Users

Password(Developers)

To change Developers's password, click [Password Setup].

1) Select the user name "Developers".

2) Click the Password Setup button.



Change Password

Change Password

Please enter a new password, re-enter the password to confirm, and then click [OK].

Please enter the password with 6 to 32 single-byte characte
alphabets A-Z, a-z, single space and !"#$%&()*+,-./:;<=
are case-sensitive.

New Password: ******

Re-enter Password: ******

3) The Change Password dialog box is displayed. Enter the following items.

New Password	: Users1
Re-enter Password	: Users1

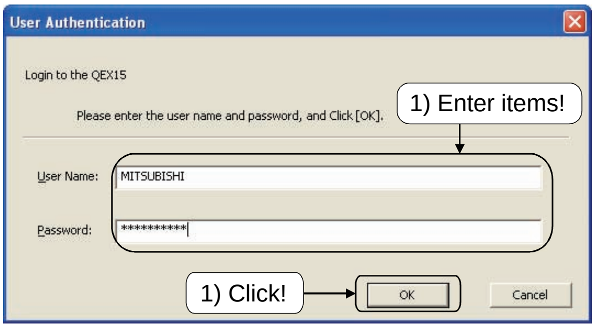
4) After entering them, click the OK button.

The password of the user "Developers" is changed.

[Deleting users]
Delete a user selected in the list with the Delete... button on the User Management dialog box.
The current login user cannot be deleted.
However, when the registered user is only "Administrators" and no other users to be deleted exist, the current login user can be deleted.
When all users are deleted, security is reset.

6.8.3 Logging in projects

A user authentication is required for opening a project with security.



User Authentication

Login to the QEX15

Please enter the user name and password, and Click [OK].

User Name: MITSUBISHI

Password: ***********

1) When a project with security is opened, the User Authentication screen is displayed. Enter a user name and a password for log-in, and click the OK button.

   Enter the following user name and password, which are set in section 6.8.1.

   User Name : MITSUBISHI
   Password : MITSUBISHI

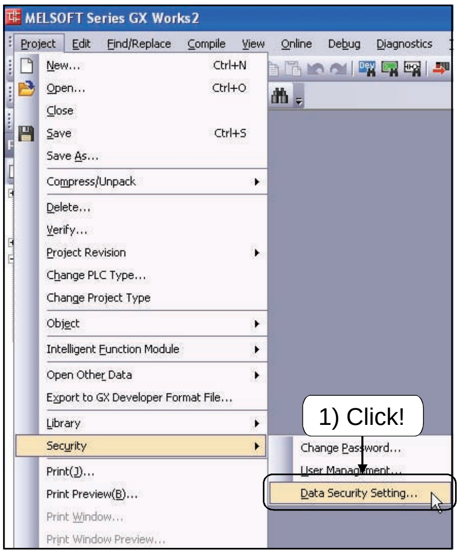
2) The project is displayed.

6.8.4 Changing access authority for each access level

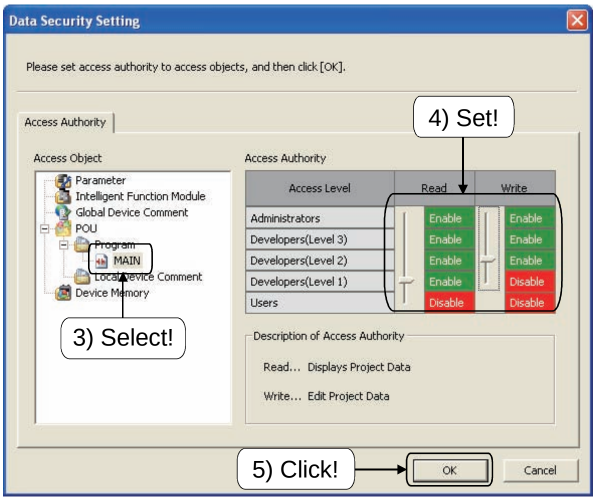
This section explains how to set an authorization of displaying and saving data for each access level.

The access authority of access levels higher than that of the login user cannot be changed.

When the access level of the current login user is "Users", the access authority cannot be changed.



1) Click [Project] → [Security] → [Data Security Setting].



Please set access authority to access objects, and then click [OK].

Access Authority

Access Level	Read	Write
Administrators	Enable	Enable
Developers(Level 3)	Enable	Enable
Developers(Level 2)	Enable	Enable
Developers(Level 1)	Enable	Disable
Users	Disable	Disable

Description of Access Authority

Read... Displays Project Data

Write... Edit Project Data

2) The Data Security Setting dialog box is displayed.

3) Select a target item from Access Object.

4) Set "Enable" or "Disable" for reading and writing data from Access Authority for each access level by moving the slider.

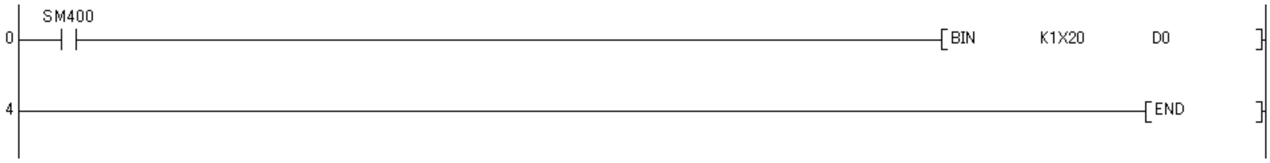
5) Click the OK button.

6.9 Sampling Trace Function

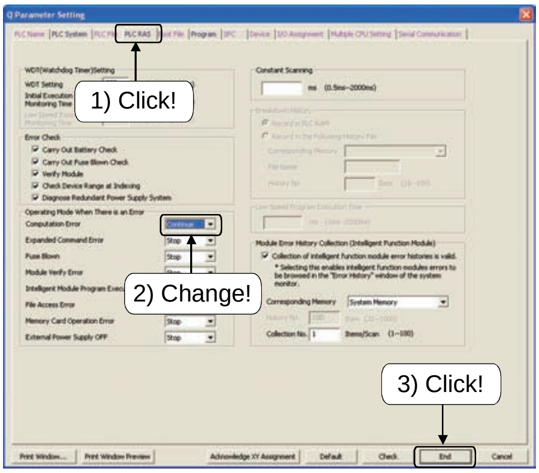
This function is used to acquire data at the specified timing to find how device values change during program operation and to trace the changes displayed in time series. For details of the sampling trace function, refer to the manuals of each CPU module.

In this example, the device value at an error occurrence is acquired.

Project name	TRACE
Program name	MAIN



As a preparation, follow the procedure below.



WDT(Watchdog Timer)Setting
WDT Setting
Initial Execution Monitoring Time
Low Speed Execution Monitoring Time

Error Check
☑ Carry Out Battery Check
☑ Carry Out Fuse Blown Check
☑ Verify Module
☑ Check Device Range at Indexing
☑ Diagnose Redundant Power Supply System

Operating Mode When There is an Error
Computation Error: Continue
Expanded Command Error: Stop
Fuse Blown: Stop
Module Verify Error: Stop
Intelligent Module Program Execution Error
File Access Error
Memory Card Operation Error: Stop
External Power Supply OFF: Stop

Constant Scanning
ms (0.5ms~2000ms)

Breakdown History
● Record in PLC RAM
○ Record in the Following History File
Corresponding Memory
File Name
History No. Item (16~100)

Low Speed Program Execution Time
ms (1ms~2000ms)

Module Error History Collection (Intelligent Function Module)
☑ Collection of intelligent function module error histories is valid.
* Selecting this enables intelligent function modules errors to be browsed in the "Error History" window of the system monitor.
Corresponding Memory: System Memory
History No. 100 Item (32~1000)
Collection No. 1 Items/Scan (1~100)

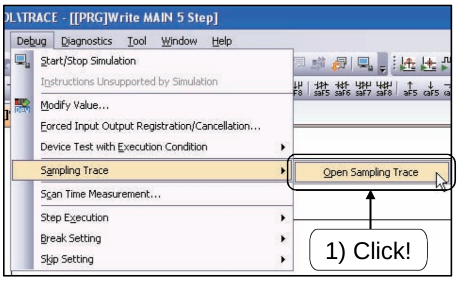
1) Click the "PLC RAS" tab on the Q Parameter Setting dialog box.

2) Select "Continue" from the Computation Error list box in the Operating Mode When There is an Error area.

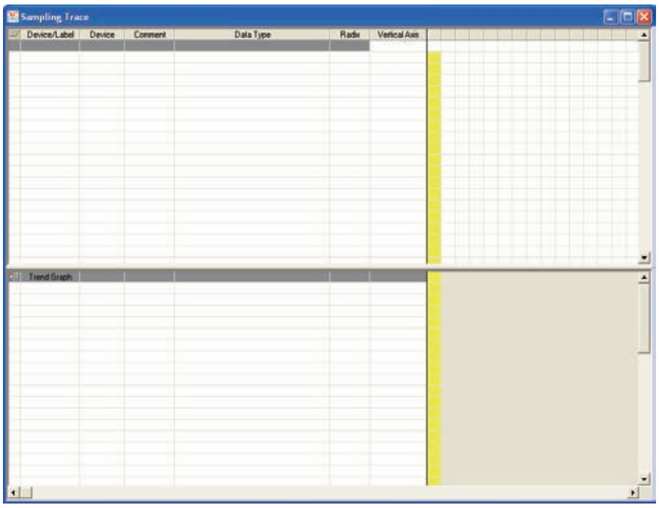
3) Click the End button.

4) Write parameters and programs to the CPU.

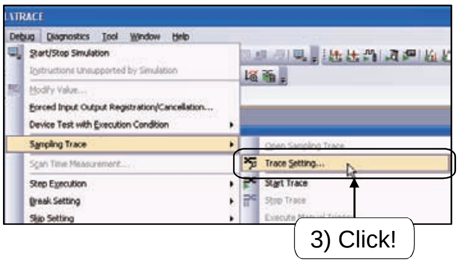
(1) Setting the sampling trace



1) Click [Debug] → [Sampling Trace] → [Open Sampling Trace].



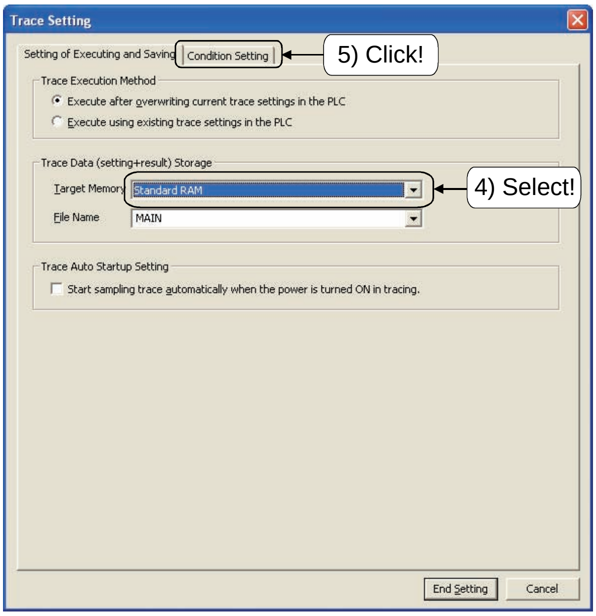
2) The Sampling Trace screen is displayed.



3) Click [Debug] → [Sampling Trace] → [Trace Setting].

(To the next page)

(From the previous page)



Trace Setting

Setting of Executing and Saving | Condition Setting

Trace Execution Method
● Execute after overwriting current trace settings in the PLC
○ Execute using existing trace settings in the PLC

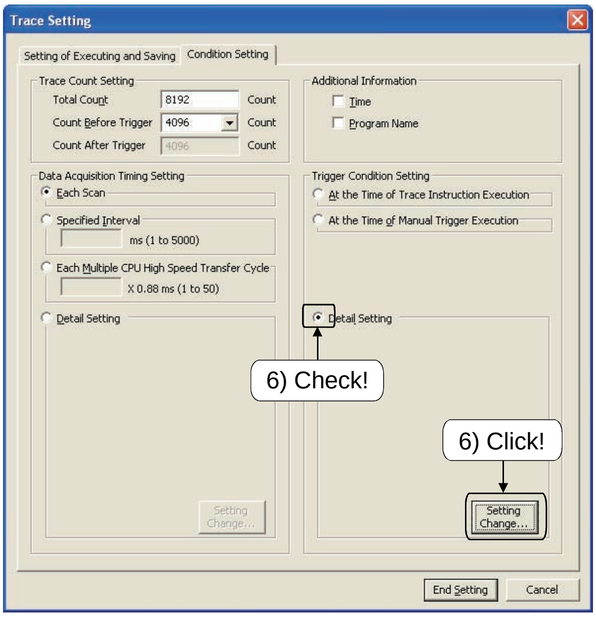
Trace Data (setting+result) Storage
Target Memory: Standard RAM
File Name: MAIN

Trace Auto Startup Setting
□ Start sampling trace automatically when the power is turned ON in tracing.

End Setting | Cancel

4) The Trace Setting dialog box is displayed. Select "Standard RAM" from the "Target Memory" list box.

5) Click the Condition Setting tab.



Trace Setting

Setting of Executing and Saving | Condition Setting

Trace Count Setting
Total Count: 8192 Count
Count Before Trigger: 4096 Count
Count After Trigger: 4096 Count

Additional Information
□ Time
□ Program Name

Data Acquisition Timing Setting
● Each Scan
○ Specified Interval
ms (1 to 5000)
○ Each Multiple CPU High Speed Transfer Cycle
X 0.88 ms (1 to 50)
○ Detail Setting
Setting Change...

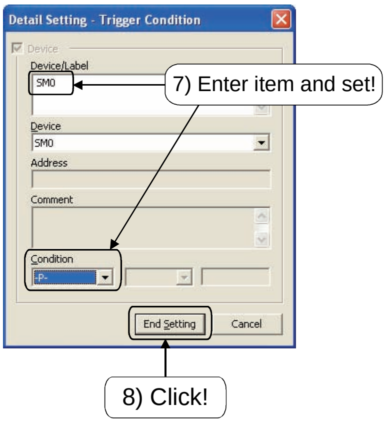
Trigger Condition Setting
○ At the Time of Trace Instruction Execution
○ At the Time of Manual Trigger Execution
● Detail Setting
Setting Change...

End Setting | Cancel

6) Check "Detail Setting" in the Trigger Condition Setting area and click the Setting Change button.

(To the next page)

(From the previous page)



Detail Setting - Trigger Condition

☑ Device

Device/Label: SM0

Device: SM0

Address:

Comment:

Condition: -P-

7) Enter item and set!

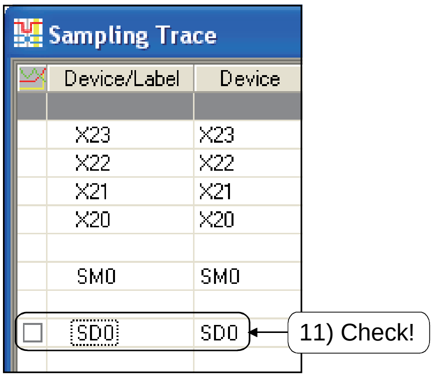
8) Click!

7) The Detail Setting - Trigger Condition dialog box is displayed. Set the following items. In this example, set error occurrence as trigger condition.

Device/Label : SM0
Condition : -P-

8) Click the End Setting button.

9) The Detail Setting - Trigger Condition dialog box disappears. Click the End Setting button to close the Trace Setting dialog box.



Sampling Trace

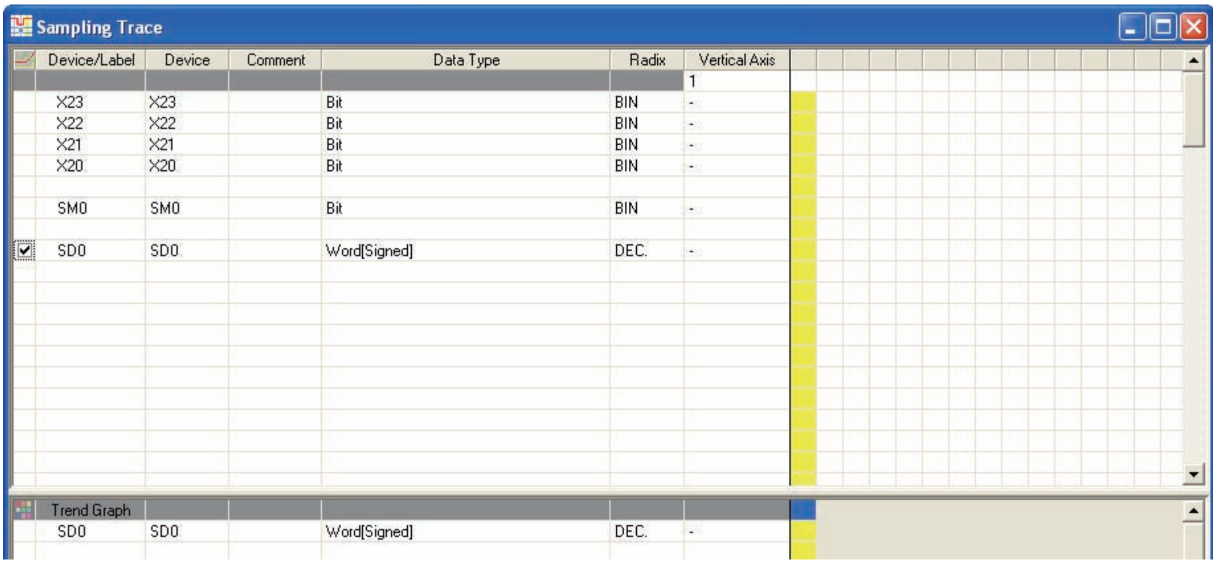
	Device/Label	Device
	X23	X23
	X22	X22
	X21	X21
	X20	X20
	SM0	SM0
☐	SD0	SD0

11) Check!

10) Set devices to be traced on the Sampling Trace screen as shown on the left.

11) Check the check box to display the trend graph of SD0.

12) "SD0" is displayed in the trend graph area on the Sampling Trace screen.

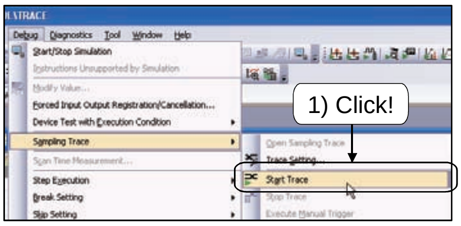


Sampling Trace

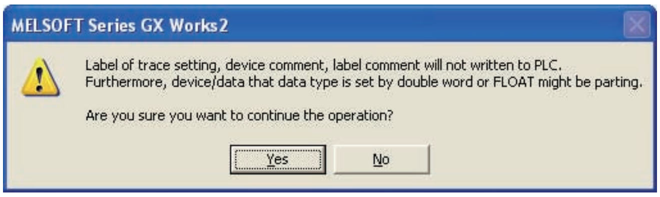
	Device/Label	Device	Comment	Data Type	Radix	Vertical Axis
						1
	X23	X23		Bit	BIN	-
	X22	X22		Bit	BIN	-
	X21	X21		Bit	BIN	-
	X20	X20		Bit	BIN	-
	SM0	SM0		Bit	BIN	-
☑	SD0	SD0		Word[Signed]	DEC.	-

Trend Graph					
SD0	SD0		Word[Signed]	DEC.	-

(2) Starting the sampling trace



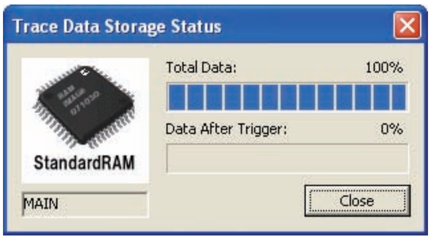
1) Click [Debug] → [Sampling Trace] → [Start Trace].


MELSOFT Series GX Works2

Label of trace setting, device comment, label comment will not written to PLC.
Furthermore, device/data that data type is set by double word or FLOAT might be parting.

Are you sure you want to continue the operation?

2) The message shown on the left is displayed.
Click the Yes button.

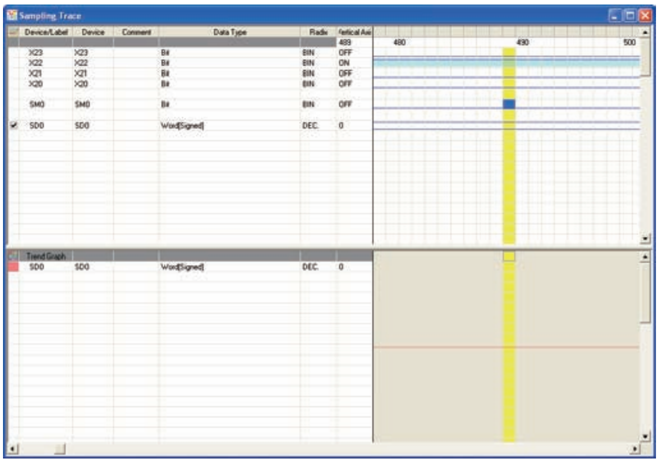

Trace Data Storage Status

Total Data: 100%

Data After Trigger: 0%

3) The Trace Data Storage Status screen is displayed when the sampling trace is started.

After confirming that the total data reaches 100%, operate digital switches to generate an error.


Sampling Trace

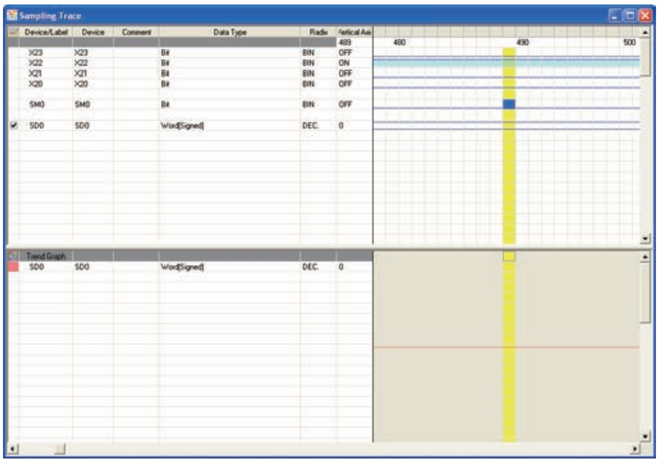
Device/Label	Device	Comment	Data Type	Radix	Vertical Axis
					489
X23	X23		Bit	BIN	OFF
X22	X22		Bit	BIN	ON
X21	X21		Bit	BIN	OFF
X20	X20		Bit	BIN	OFF
SM0	SM0		Bit	BIN	OFF
SD0	SD0		Word[Signed]	DEC.	0

Trend Graph

SD0	SD0		Word[Signed]	DEC.	0

4) The trace result is displayed on the Sampling Trace screen.

(3) Checking the trace result



Sampling Trace

	Device/Label	Device	Comment	Data Type	Radix	/ertical Axi
						489
	X23	X23		Bit	BIN	OFF
	X22	X22		Bit	BIN	ON
	X21	X21		Bit	BIN	OFF
	X20	X20		Bit	BIN	OFF
	SM0	SM0		Bit	BIN	OFF
✓	SD0	SD0		Word[Signed]	DEC.	0

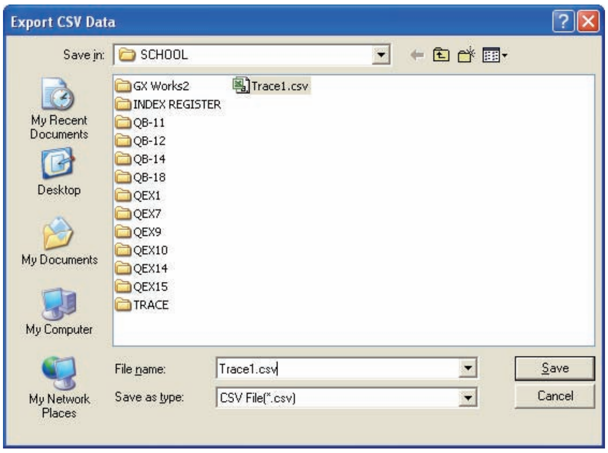
	Trend Graph					
🟥	SD0	SD0		Word[Signed]	DEC.	0

1) Scroll the trend graph screen to the trigger point to check the device value at an error occurrence.

POINT

Saving trace data to a personal computer

Click [Debug] → [Sampling Trace] → [Export CSV Data]. The following dialog box is displayed.



Export CSV Data

Save in: SCHOOL

- GX Works2
- INDEX REGISTER
- QB-11
- QB-12
- QB-14
- QB-18
- QEX1
- QEX7
- QEX9
- QEX10
- QEX14
- QEX15
- TRACE
- Trace1.csv

File name: Trace1.csv

Save as type: CSV File(*.csv)

After entering a file name, click the Save button.

CHAPTER 7 PROGRAMMING INTELLIGENT FUNCTION MODULE

7.1 Intelligent Function Module

(1) Intelligent function module type

On programmable controller CPUs (hereinafter referred to as QCPUs), some functions are not supported or are limited in use. Intelligent function modules support those functions instead of QCPUs.

Therefore users need to select an intelligent function module that is appropriate for the purpose involved.

QCPUs are compatible with QCPU-compatible intelligent function modules.

The following table shows examples of the intelligent function modules.

Table 7.1 Example of intelligent function module

Name	Number of I/O occupied points	Function	Module current consumption
Analog-digital converter module (Q64AD)	16 points	Input module that converts; 0 to 20mA → 0 to 4000 (in standard resolution mode), 0 to ±10V → 0 to ±4000 (in standard resolution mode)	5VDC 0.63A
Digital-analog converter module (Q62DAN)	16 points	Output module that converts; 0 to 4000 → 0 to 20mA (in standard resolution mode), 0 to ±4000 → 0 to ±10V (in standard resolution mode)	5VDC 0.33A 24VDC 0.12A

(2) Using intelligent function modules with CPUs

An intelligent function module can be installed on any I/O slots on a main base unit and extension base unit.

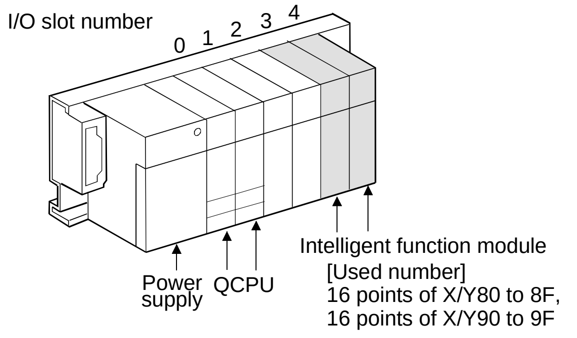


Figure 7.1 Installation of intelligent function module

7.2 Data Communication between Intelligent Function Modules and CPUs

An intelligent function module and a CPU exchange mainly two formats of data.
Bit data ------------Signals that use input Xs and output Ys
Word data --------16-bit data or 32-bit data

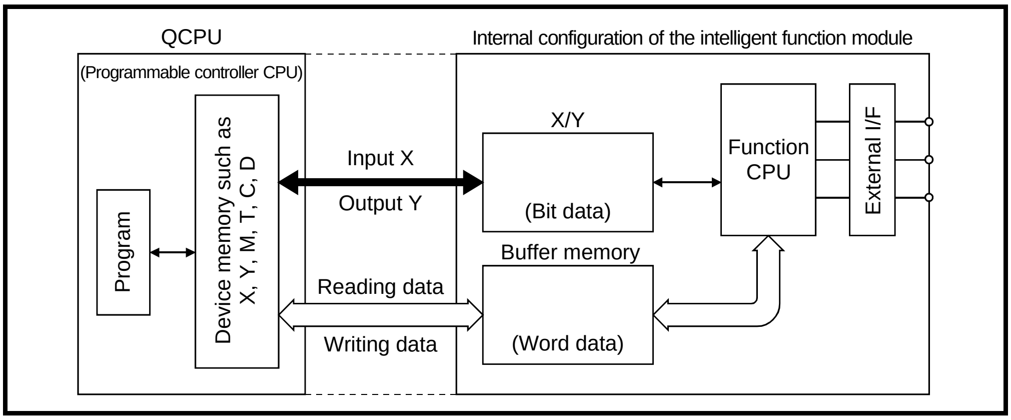

QCPU
(Programmable controller CPU)
Program
Device memory such as X, Y, M, T, C, D
Input X
Output Y
Reading data
Writing data
Internal configuration of the intelligent function module
X/Y
(Bit data)
Buffer memory
(Word data)
Function CPU
External I/F

Figure 7.2 Internal configuration of the intelligent function module

7.2.1 I/O signals to CPUs

For 1-bit signals exchanged between a QCPU and an intelligent function module, input Xs and output Ys are used.
Xs and Ys here do not mean external I/Os but symbols that are used in a sequence program to exclusively represent I/O signals of intelligent function modules. Also note that I/O numbers are assigned according to the slot where the intelligent function module is installed.

[X]

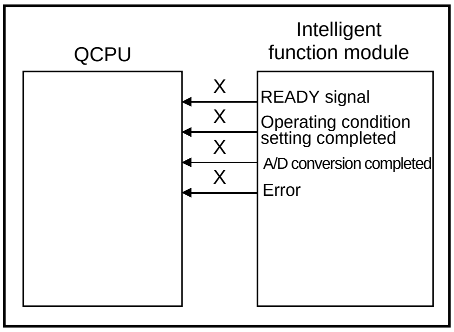

QCPU
Intelligent function module
X
X
X
X
READY signal
Operating condition setting completed
A/D conversion completed
Error

Figure 7.3
X from intelligent function module

Xs in a sequence program represent signals that are input to a QCPU from an intelligent function module. These signals are generated on an intelligent function module. Note that the Xs are used as contacts in a program. The following is examples of the signals.

(1) READY signal
This signal notifies a QCPU that an intelligent function module started up normally at power-on and is ready for operation.

(2) Operating condition setting completed
This signal is used as an interlock condition for turning Operating condition setting request (Y9) on/off when the following settings are changed.
- A/D conversion enable/disable setting (buffer memory address 0: Un\G0)
- CH□ Average time/average number of times
(buffer memory addresses 1 to 8: Un\G1 to Un\G8)
- Averaging process setting
(buffer memory address 9: Un\G0)

[Y]

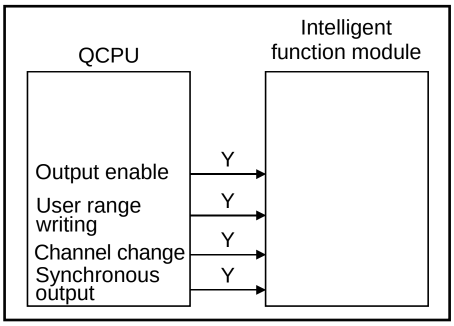


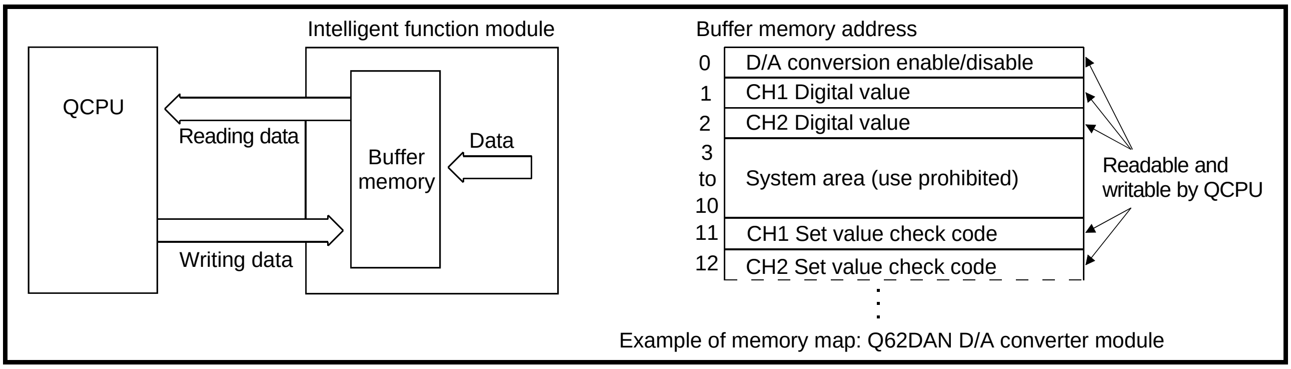
Figure 7.4 Y from CPU

SETs, RSTs, or OUT-Ys represent output signals transmitted from a QCPU to an intelligent function module. These signals are generated on a QCPU. Note that they are used as coils or contacts in a program.

(Example) D/A converter modules output an enable instruction (output enable) before outputting analog values that were converted from digital values.

7.2.2 Data communication with intelligent function modules

Data is transmitted or received in 16-bit or 32-bit units. Intelligent function modules have a buffer memory to store those data.

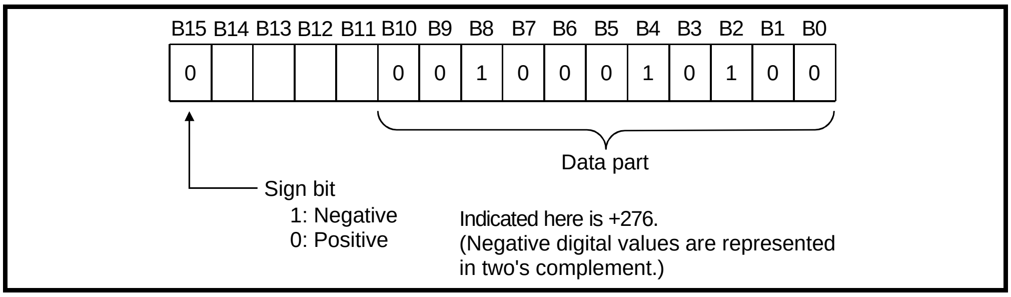

Intelligent function module
QCPU
Reading data
Buffer memory
Data
Writing data
Buffer memory address
Readable and writable by QCPU
Example of memory map: Q62DAN D/A converter module

Buffer memory address	
0	D/A conversion enable/disable
1	CH1 Digital value
2	CH2 Digital value
3 to 10	System area (use prohibited)
11	CH1 Set value check code
12	CH2 Set value check code

Figure 7.5 Buffer memory

(1) QCPUs can read and write data to and from the buffer memory. Also note that some modules can write data to buffer memory from peripheral device via an interface.

(2) In a buffer memory, space of one word (16 bits) is reserved for each intelligent function module's unique address.
The smallest address is 0, and these addresses are used to specify a target module to read or write. The minimum unit is one word. Data of 17 bits to 32 bits is treated as 2-word (32-bit) data.


Data part
Sign bit
1: Negative
0: Positive
Indicated here is +276.
(Negative digital values are represented in two's complement.)

B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
0					0	0	1	0	0	0	1	0	1	0	0

Figure 7.6 Example image of buffer memory content (D/A converter module)

Figure 7.6 shows 16 bits of the buffer memory of a D/A converter module where a digital quantities have been written. The number is obtained from digital quantity that a QCPU wrote to the buffer memory within the range from -4096 to +4095 in signed binary (16 bits long).

(3) A buffer memory is a RAM.

7.3 Communication with Intelligent Function Module

7.3.1 Communication methods with intelligent function modules

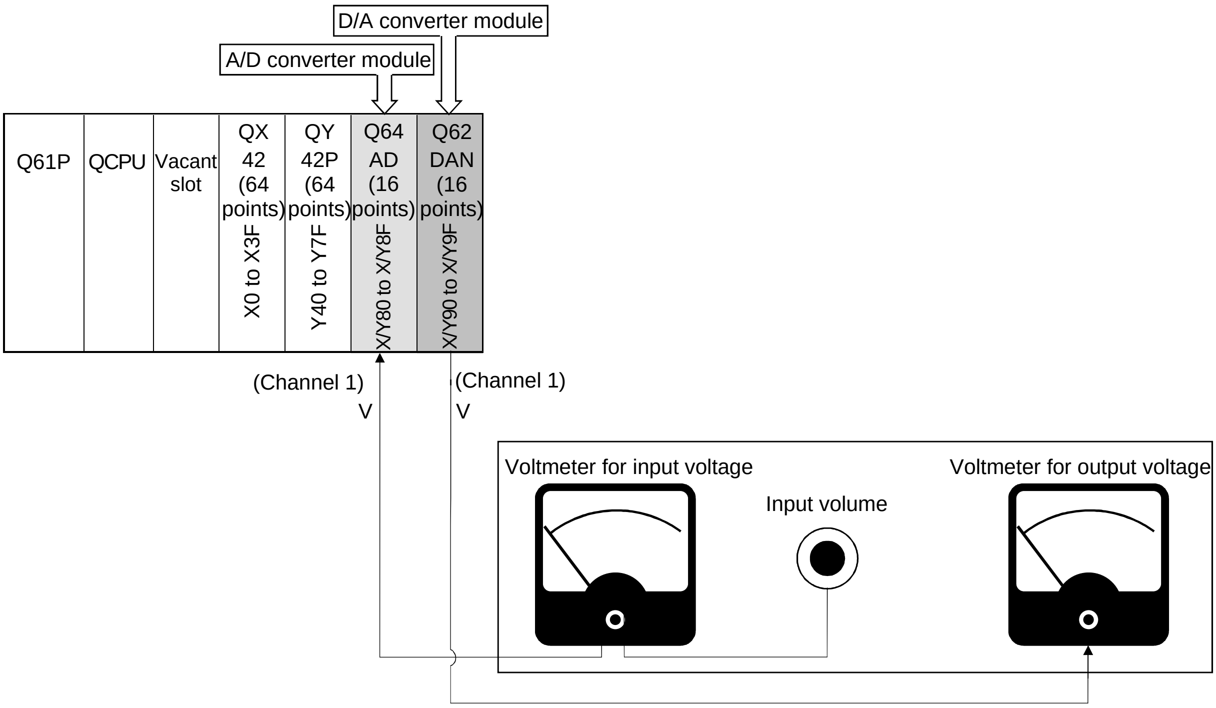
The following table shows the communication methods between a QCPU and an intelligent function module.

Table 7.2 Communication method with intelligent function modules

Communication method	Function	Setting method
Initial setting, Auto refresh setting	Performs initial settings and auto refresh settings of intelligent function modules. These settings allow writing/reading data to/from intelligent function modules regardless of communication program creation or buffer memory address. Ex.) When A/D converter module Q64AD is used • Initial setting : • A/D conversion enable/disable setting • Sampling/averaging processing specification, • Time average/number of times average specification, • Average time/average number of times specification (Set data in auto refresh settings is stored to the intelligent function module parameter on a QCPU.) • Auto refresh setting : Set a device on a QCPU to store the following data to. • Digital output from Q64AD • Maximum and minimum values of Q64AD • Error code (Set data in auto refresh settings is stored to the intelligent function module parameter on a QCPU.)	Use GX Works2.
Device initial value	Writes set data in device initial settings of intelligent function modules to the intelligent function modules at the following timings. • At power-on of a QCPU • At reset • At switching from STOP to RUN	Use GX Works2 to specify the range for intelligent function module devices (U□\G□).
FROM/TO instruction	Read or write data from or to the buffer memory on an intelligent function module.	Use this instruction in a sequence program.
Intelligent function module device (U□\G□)	Directly handles the buffer memory on an intelligent function module as a device of a QCPU. Unlike "FROM/TO instruction", this requires only one instruction for processing data that is read from an intelligent function module.	Specify this device as a device in a sequence program.
Intelligent function module dedicated instruction	Used to simplify programming for using the functions of intelligent function modules.	Use this instruction in a sequence program.

7.4 Intelligent Function Module System in Demonstration Machine

Use an A/D or D/A converter module to convert analog signals/digital data that are input with the volume or digital switch on the demonstration machine.

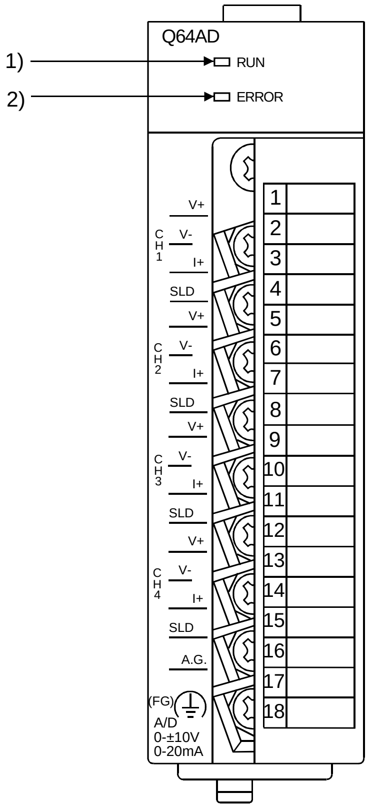


7.5 Q64AD Analog/Digital Converter Module

7.5.1 Names of parts

The following explains the parts of Q64AD.
For details, refer to the User's Manual.

Q64AD



No.	Name and appearance	Description
1)	RUN LED	Indicates the operation status of the A/D converter module. ON : In normal operation Flicker : In offset/gain setting mode OFF : 5V power failure or watchdog timer error occurred
2)	ERROR LED	Indicates errors and the status of the A/D converter module. ON : Error occurred OFF : In normal operation Flicker : Switch setting error occurred Values other than 0 has been set to the switch 5 on the intelligent function module.

7.5.2 A/D conversion characteristics

(1) A/D conversion characteristics on voltage inputs
(For analog input range from -10 to 10V in a standard resolution mode)

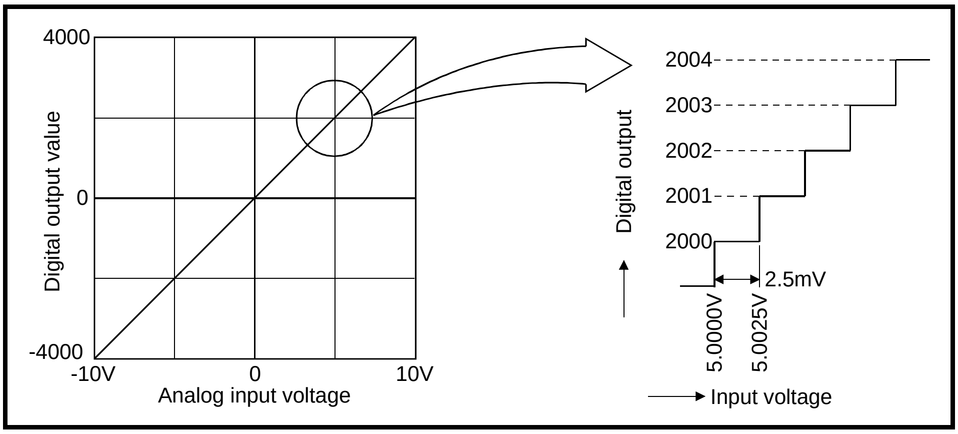


Figure 7.12 A/D conversion characteristics (voltage input)

A/D converter modules convert analog values input from other devices to digital quantities so that CPUs can operate those values. On voltage inputs, for example, A/D converter modules convert -10V to a quantity of -4000 and 10V to 4000. This means that the modules convert an input voltage of 2.5mV to a digital quantity of 1, and abandon values smaller than 2.5mV.

(2) A/D conversion characteristics on current inputs
(For analog input range from 0 to 20mA in a standard resolution mode)

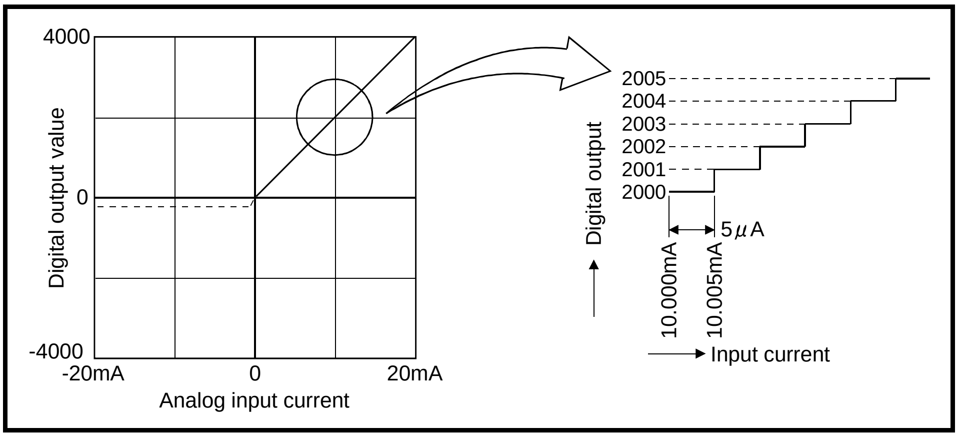


Figure 7.13 A/D conversion characteristics (current input)

The modules convert current an input of 0mA to 0 for an output, and 20mA to 4000. This means that the modules convert an input current of 5μA to a digital quantity of 1, and abandon values smaller than 5μA.

REMARK

A voltage or current value that is equivalent to a digital value of 1 through A/D conversion (maximum resolution) differs depending on the setting of the resolution mode (1/4000, 1/12000, 1/16000) or the output range.

7.5.3 List of I/O signals and buffer memory assignment

(1) List of I/O signals

The following shows a list of the I/O signals for the A/D converter modules. Note that I/O numbers (X/Y) shown in this section and thereafter are the values when the start I/O number for the A/D converter module is set to 0.

Signal direction: CPU ← A/D converter module		Signal direction: CPU → A/D converter module	
Device No. (input)	Signal name	Device No. (output)	Signal name
X0	Module READY	Y0	Use prohibited[*1]
X1	Temperature drift compensation flag	Y1	
X2	Use prohibited[*1]	Y2	
X3		Y3	
X4		Y4	
X5		Y5	
X6		Y6	
X7		Y7	
X8	High resolution mode status flag	Y8	
X9	Operating condition setting completed flag	Y9	Operating condition setting request
XA	Offset/gain setting mode flag	YA	User range writing request
XB	Channel change completed flag	YB	Channel change request
XC	Use prohibited[*1]	YC	Use prohibited[*1]
XD	Maximum value/minimum value reset completed flag	YD	Maximum value/minimum value reset request
XE	A/D conversion completed flag	YE	Use prohibited[*1]
XF	Error flag	YF	Error clear request

POINT

*1: These signals cannot be used by the user since they are for system use only. If these are turned on/off by the sequence program, the functioning of the A/D converter module cannot be guaranteed.

(2) Buffer memory assignment (Q64AD)

This section explains the assignment of the Q64AD buffer memory.

POINT

Do not write data to the system areas or areas to which writing data from a sequence program is disabled. Doing so may cause malfunction.

Buffer memory assignment (Q64AD) (1/2)

Address		Description	Default	Read/ write*1
Hexadecimal	Decimal			
0H	0	A/D conversion enable/disable setting	0	R/W
1H	1	CH1 Average time/average number of times	0	R/W
2H	2	CH2 Average time/average number of times	0	R/W
3H	3	CH3 Average time/average number of times	0	R/W
4H	4	CH4 Average time/average number of times	0	R/W
5H	5	System area	-	-
⋮	⋮			
8H	8			
9H	9	Averaging process setting	0	R/W
AH	10	A/D conversion completed flag	0	R
BH	11	CH1 Digital output value	0	R
CH	12	CH2 Digital output value	0	R
DH	13	CH3 Digital output value	0	R
EH	14	CH4 Digital output value	0	R
FH	15	System area	-	-
⋮	⋮			
12H	18			
13H	19	Error code	0	R
14H	20	Setting range (CH1 to CH4)	0	R
15H	21	System area	-	-
16H	22	Offset/gain setting mode Offset specification	0	R/W
17H	23	Offset/gain setting mode Gain specification	0	R/W

*1: Indicates whether reading from and writing to a sequence program are enabled.
R: Read enabled
W: Write enabled

Buffer memory assignment (Q64AD) (2/2)

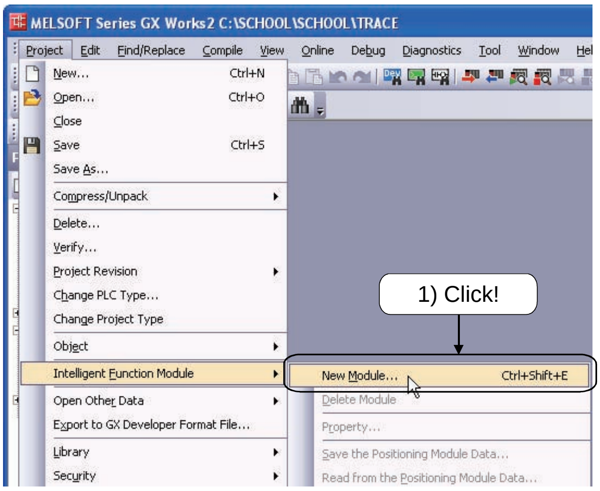
Address		Description	Default	Read/write*1
Hexadecimal	Decimal			
18H	24	System area	-	-
⋮	⋮			
1DH	29			
1EH	30	CH1 Maximum value	0	R/W
1FH	31	CH1 Maximum value	0	R/W
20H	32	CH2 Maximum value	0	R/W
21H	33	CH2 Maximum value	0	R/W
22H	34	CH3 Maximum value	0	R/W
23H	35	CH3 Maximum value	0	R/W
24H	36	CH4 Maximum value	0	R/W
25H	37	CH4 Maximum value	0	R/W
26H	38	System area	-	-
⋮	⋮			
9DH	157			
9EH	158	Mode switching setting	0	R/W
9FH	159			
A0H	160	System area	-	-
⋮	⋮			
C7H	199			
C8H	200	Pass data classification setting*2	0	R/W
C9H	201	System area	-	-
CAH	202	CH1 Industrial shipment settings offset value*2	0	R/W
CBH	203	CH1 Industrial shipment settings gain value*2	0	R/W
CCH	204	CH2 Industrial shipment settings offset value*2	0	R/W
CDH	205	CH2 Industrial shipment settings gain value*2	0	R/W
CEH	206	CH3 Industrial shipment settings offset value*2	0	R/W
CFH	207	CH3 Industrial shipment settings gain value*2	0	R/W
D0H	208	CH4 Industrial shipment settings offset value*2	0	R/W
D1H	209	CH4 Industrial shipment settings gain value*2	0	R/W
D2H	210	CH1 User range settings offset value*2	0	R/W
D3H	211	CH1 User range settings gain value*2	0	R/W
D4H	212	CH2 User range settings offset value*2	0	R/W
D5H	213	CH2 User range settings gain value*2	0	R/W
D6H	214	CH3 User range settings offset value*2	0	R/W
D7H	215	CH3 User range settings gain value*2	0	R/W
D8H	216	CH4 User range settings offset value*2	0	R/W
D9H	217	CH4 User range settings gain value*2	0	R/W

*1: Indicates whether reading from and writing to a sequence program are enabled.
R: Read enabled
W: Write enabled

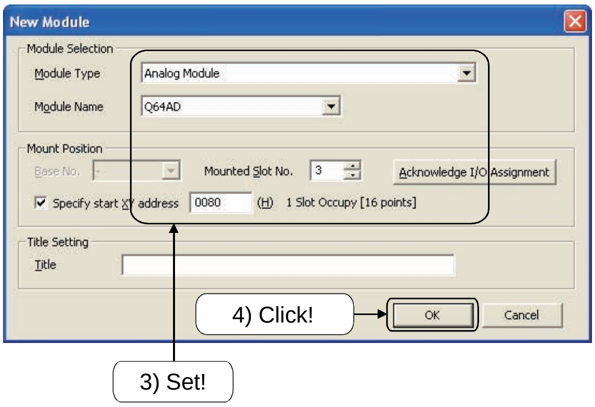
*2: Areas used to restore the user range settings offset/gain values when online module change is made.

7.5.4 Adding or setting intelligent function module data

This section explains how to set the intelligent function module data.
After an intelligent function module is added to a project, the data settings (parameters and switch settings) of the intelligent function module can be set.



1) Click [Project] → [Intelligent Function Module] → [New Module].

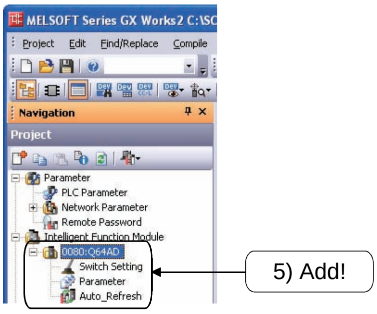


Module Type: Analog Module
Module Name: Q64AD
Base No.: -
Mounted Slot No.: 3
☑ Specify start XY address: 0080 (H) 1 Slot Occupy [16 points]

2) The New Module dialog box is displayed.

3) Set the A/D converter module setting as follows.
Module Type : Analog Module
Module Name : Q64AD
Mounted Slot No. : 3
(Specify start XY address: 0080)

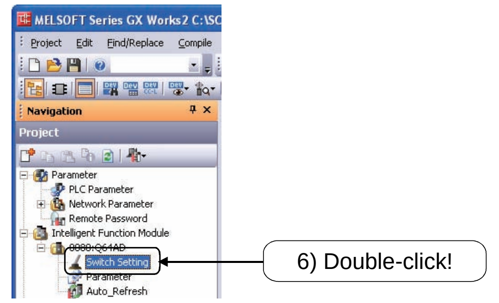
4) Click the OK button.



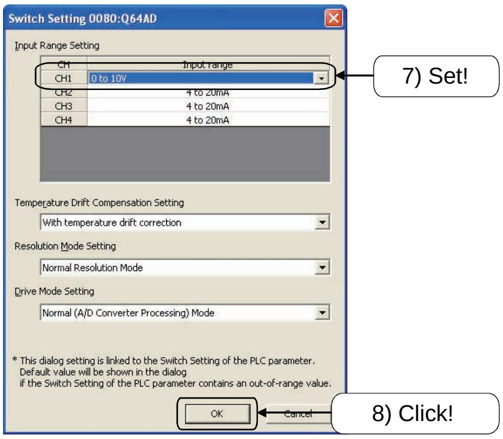
5) The specified intelligent function module data are added to the Project window.

(To the next page)

(From the previous page)



6) Double-click Switch Setting.



Input Range Setting

CH	Input range
CH1	0 to 10V
CH2	4 to 20mA
CH3	4 to 20mA
CH4	4 to 20mA

Temperature Drift Compensation Setting: With temperature drift correction

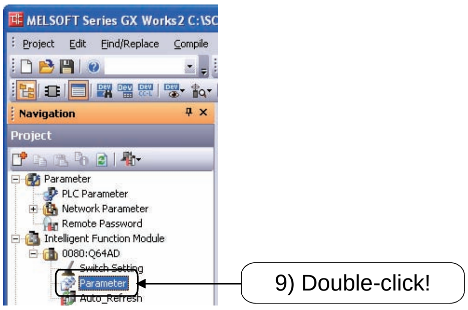
Resolution Mode Setting: Normal Resolution Mode

Drive Mode Setting: Normal (A/D Converter Processing) Mode

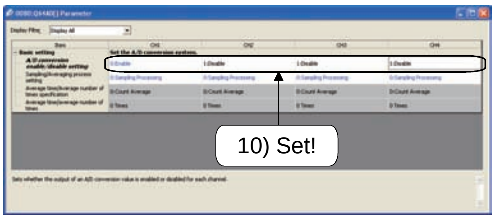
* This dialog setting is linked to the Switch Setting of the PLC parameter.
Default value will be shown in the dialog
if the Switch Setting of the PLC parameter contains an out-of-range value.

7) The Switch Setting screen is displayed.
Set Input range for CH1 to "0 to 10V".

8) Click the OK button.



9) Double-click Parameter.



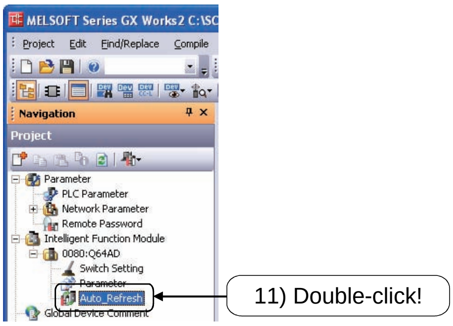
Item	CH1	CH2	CH3	CH4
Basic setting	Set the A/D conversion system.			
A/D conversion enable/disable setting	0:Enable	1:Disable	1:Disable	1:Disable
Sampling/Averaging process setting	0:Sampling Processing	0:Sampling Processing	0:Sampling Processing	0:Sampling Processing
Average time/Average number of times specification	0:Count Average	0:Count Average	0:Count Average	0:Count Average
Average time/average number of times	0 Times	0 Times	0 Times	0 Times

Sets whether the output of an A/D conversion value is enabled or disabled for each channel.

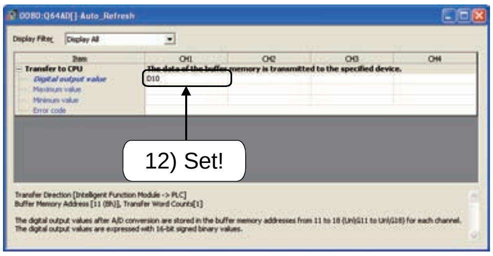
10) The Parameter screen is displayed.
Set "A/D conversion enable/disable setting" for CH2 to CH4 to "1:Disable". (Only CH1 is used.)

(To the next page)

(From the previous page)



11) Double-click Auto_Refresh.

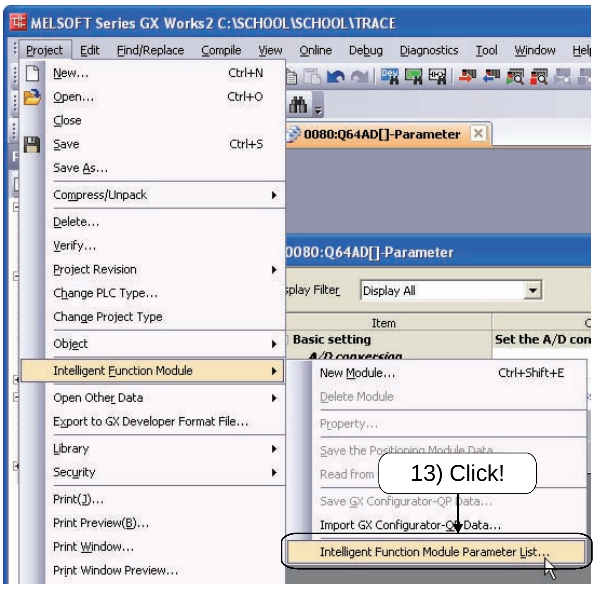


Item	CH1	CH2	CH3	CH4
Transfer to CPU	The data of the buffer memory is transmitted to the specified device.			
Digital output value	D10			
Maximum value				
Minimum value				
Error code				

Transfer Direction [Intelligent Function Module -> PLC]
Buffer Memory Address [11 (Bh)], Transfer Word Counts[1]

The digital output values after A/D conversion are stored in the buffer memory addresses from 11 to 18 (Un\G11 to Un\G18) for each channel.
The digital output values are expressed with 16-bit signed binary values.

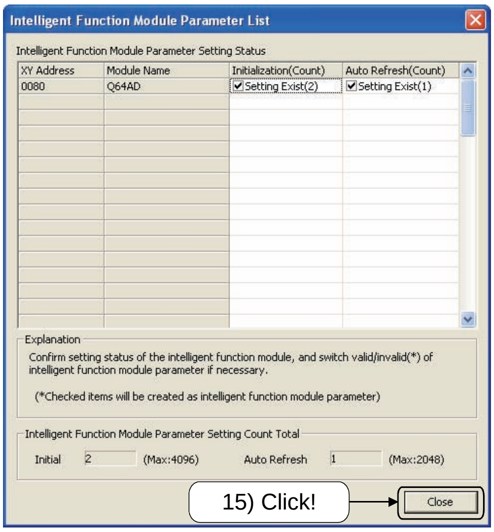
12) The Auto_Refresh screen is displayed.
Set Digital output value for CH1 to "D10".


MELSOFT Series GX Works2 C:\SCHOOL\SCHOOL\TRACE
Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help
New... Ctrl+N
Open... Ctrl+O
Close
Save Ctrl+S
Save As...
Compress/Unpack
Delete...
Verify...
Project Revision
Change PLC Type...
Change Project Type
Object
Intelligent Function Module
Open Other Data
Export to GX Developer Format File...
Library
Security
Print(J)...
Print Preview(B)...
Print Window...
Print Window Preview...
0080:Q64AD[]-Parameter
Display Filter Display All
Item
Basic setting
A/D conversion
Set the A/D con
New Module... Ctrl+Shift+E
Delete Module
Property...
Save the Positioning Module Data
Read from
Save GX Configurator-QP Data...
Import GX Configurator-QP Data...
Intelligent Function Module Parameter List...
13) Click!

13) Click [Project] → [Intelligent Function Module] → [Intelligent Function Module Parameter List].

(To the next page)

(From the previous page)


Intelligent Function Module Parameter List

Intelligent Function Module Parameter Setting Status

XY Address	Module Name	Initialization(Count)	Auto Refresh(Count)
0080	Q64AD	☑Setting Exist(2)	☑Setting Exist(1)

Explanation

Confirm setting status of the intelligent function module, and switch valid/invalid(*) of intelligent function module parameter if necessary.

(*Checked items will be created as intelligent function module parameter)

Intelligent Function Module Parameter Setting Count Total

Initial 2 (Max:4096) Auto Refresh 1 (Max:2048)

14) Check that "Setting Exist" is checked in Initialization (Count) and Auto Refresh (Count) for Q64AD in the Intelligent Function Module Parameter List dialog box.

15) Click the Close button.

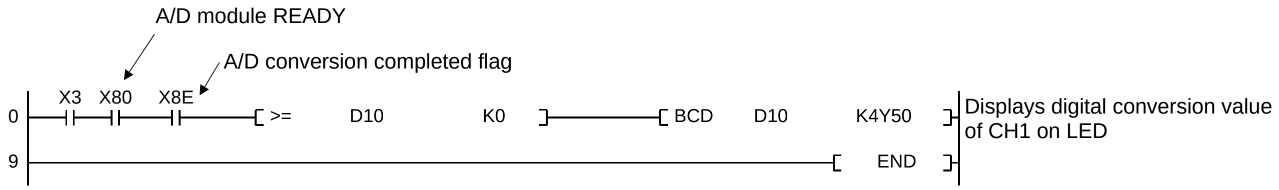
7.5.5 Exercise with the demonstration machine

(1) Sequence program

The sequence program executes a sampling processing on analog voltages input through CH1 of Q64AD, and then converts the analog values to digital values.

Set the start XY of Q64AD to 80 as explained before.

Project name	Q64AD
Program name	MAIN


A/D module READY
A/D conversion completed flag
0
X3
X80
X8E
>=
D10
K0
BCD
D10
K4Y50
Displays digital conversion value of CH1 on LED
9
END

X80: Module READY signal

X8E: A/D conversion completed flag

At power-on or reset of a programmable controller CPU, this flag turns on if A/D conversion is ready to be executed. A/D conversion is executed once this flag turned on.

(2) Operation of the demonstration machine

Stop the CPU and click on the toolbar.

The Online Data Operation dialog box is displayed. Click the Parameter + Program button, then click the Execute button to write data to the CPU. After that, activate the CPU and check the following items.

(a) Turn on X3, and change input voltages for an A/D converter module with the volume on the demonstration machine.
Analog values that have been input to the channel 1 (CH1) of Q64AD are stored to the buffer memory (in digital value). With the auto refresh settings, the QCPU reads the stored digital values and stores them in its data register D10.

(b) Whenever an analog value is "-1" or smaller, 0 is set.

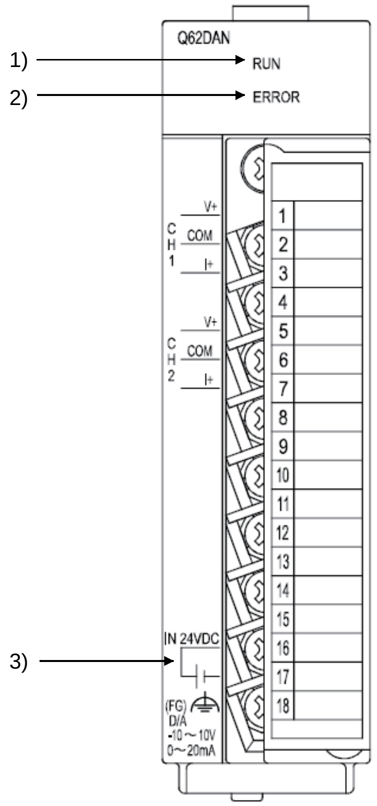
(c) The digital values are displayed on the digital display (Y50 to Y5F).

7.6 Q62DAN Digital/Analog Converter Module

7.6.1 Names of parts

The following explains the parts of Q62DAN.
For details, refer to the User's Manual.

Q62DAN



No.	Name and appearance	Description
1)	RUN LED	Indicates the operation status of the D/A converter module. ON : In normal operation Flicker : In offset/gain setting mode OFF : 5V power failure or watchdog timer error occurred
2)	ERROR LED	Indicates errors and the status of the D/A converter module. ON : Error occurred OFF : In normal operation Flicker : Switch settings error occurred Values other than 0 has been set to the switch 5 on the intelligent function module.
3)	External power supply terminal	Terminal for connecting a 24VDC external power supply

7.6.2 D/A conversion characteristics

(1) D/A conversion characteristics on voltage outputs
(For analog output range from -10 to 10V in a standard resolution mode)

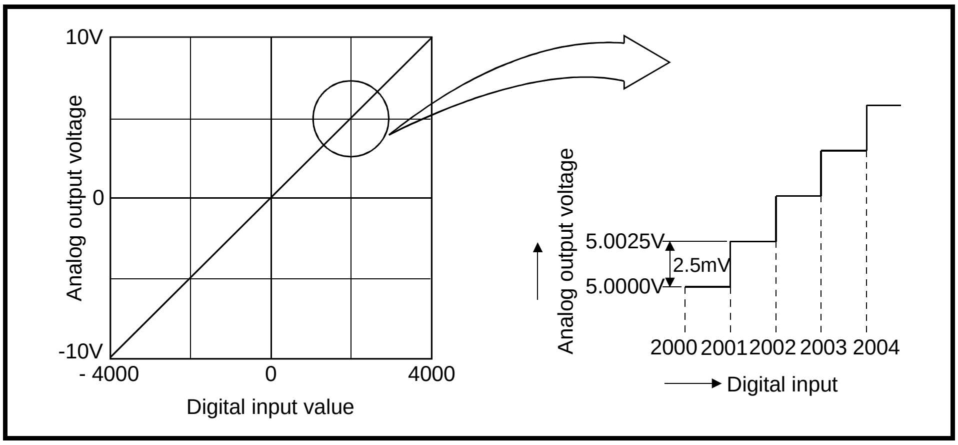


Figure 7.14 D/A conversion characteristics (current output)

D/A converter modules convert digital quantities that are input from a QCPU into analog values, and then output them. For example, the modules convert a digital quantity of -4000 to a analog quantity of -10V and 4000 to 10V before output. This means that the modules convert the digital input value of 1 to an analog quantity of 2.5mV, and abandon digital input values in decimal places.

(2) D/A conversion characteristics on current outputs
(For analog output range from 0 to 20mA in a standard resolution mode)

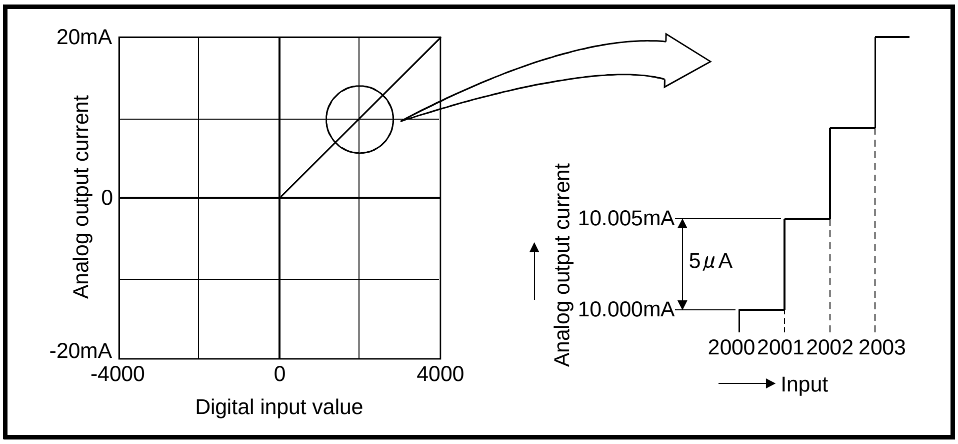


Figure 7.15 D/A conversion characteristics (current output)

For current outputs, the modules convert a digital value 0 to 0mA and 4000 to 20mA. This means that the modules convert the digital input value of 1 to an analog quantity of 5μA, and abandon digital input values in decimal places.

REMARK

A voltage or current value that is equivalent to a digital value of 1 through D/A conversion (maximum resolution) differs depending on the setting of the resolution mode (1/4000, 1/12000, 1/16000) or the output range.

7.6.3 List of I/O signals and buffer memory assignment

(1) List of I/O signals

The following shows a list of the I/O signals for the D/A converter modules.

The following explanation is mentioned based on the Q68DAVN, Q68DAIN, Q68DAV and Q68DAI with 8-channel analog output (CH1 to CH8).

Note that I/O numbers (X/Y) shown in this section and thereafter are the values when the start I/O number for the D/A converter module is set to 0.

Signal direction	D/A converter module → CPU module	Signal direction	CPU module → D/A converter module
Device No.	Signal name	Device No.	Signal name
X0	Module READY	Y0	Use prohibited*1
X1	Use prohibited*1	Y1	CH1 Output enable/disable flag
X2		Y2	CH2 Output enable/disable flag
X3		Y3*2	CH3 Output enable/disable flag
X4		Y4*2	CH4 Output enable/disable flag
X5		Y5*2	CH5 Output enable/disable flag
X6		Y6*2	CH6 Output enable/disable flag
X7		Y7*2	CH7 Output enable/disable flag
X8	High resolution mode status flag	Y8*2	CH8 Output enable/disable flag
X9	Operating condition setting completed flag	Y9	Operating condition setting request
XA	Offset/gain setting mode flag	YA	User range writing request
XB	Channel change completed flag	YB	Channel change request
XC	Set value change completed flag	YC	Set value change request
XD	Synchronous output mode flag	YD	Synchronous output request
XE	Use prohibited*1	YE	Use prohibited*1
XF	Error flag	YF	Error clear request

POINT

*1: These signals cannot be used by the user since they are for system use only. If these are turned on/off by the sequence program, the functioning of the D/A converter module cannot be guaranteed.

*2: For the Q62DAN and Q62DA, the use of Y3 to Y8 is prohibited.
For the Q64DAN and Q64DA, the use of Y5 to Y8 is prohibited.

(2) Buffer memory assignment (Q62DAN)

This section explains the assignment of the Q62DAN buffer memory.

POINT

Do not write data to the system areas or areas to which writing data from a sequence program is disabled.
Doing so may cause malfunction.

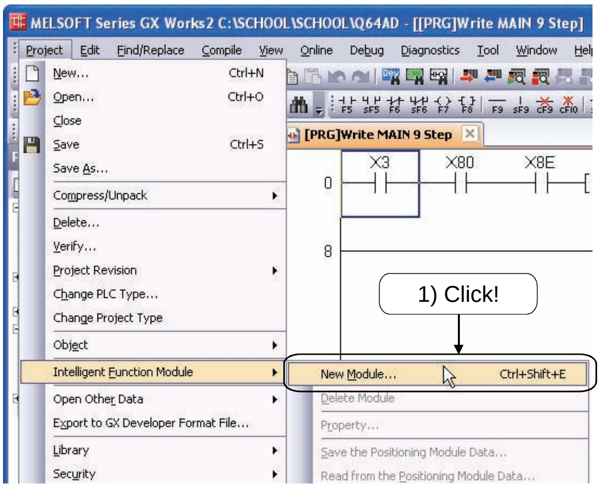
Address		Description	Default[*1]	Read/ write[*2]
Hexadecimal	Decimal			
0H	0	D/A conversion enable/disable	3H	R/W
1H	1	CH1 Digital value	0	R/W
2H	2	CH2 Digital value	0	R/W
3H	3	System area	-	-
⋮	⋮			
AH	10			
BH	11	CH1 Set value check code	0	R
CH	12	CH2 Set value check code	0	R
DH	13	System area	-	-
⋮	⋮			
12H	18			
13H	19	Error code	0	R
14H	20	Setting range (CH1 to CH2)	0H	R
15H	21	System area	-	-
16H	22	Offset/gain setting mode Offset specification	0	R/W
17H	23	Offset/gain setting mode Gain specification	0	R/W
18H	24	Offset/gain adjustment value specification	0	R/W
19H	25	System area	-	-
⋮	⋮			
9DH	157			
9EH	158	Mode switching setting	0	R/W
9FH	159		0	R/W
A0H	160	System area	-	-
⋮	⋮			
C7H	199			
C8H	200	Pass data classification setting[*3]	0	R/W
C9H	201	System area	-	-
CAH	202	CH1 Industrial shipment settings offset value[*3]	0	R/W
CBH	203	CH1 Industrial shipment settings gain value[*3]	0	R/W
CCH	204	CH2 Industrial shipment settings offset value[*3]	0	R/W
CDH	205	CH2 Industrial shipment settings gain value[*3]	0	R/W
CEH	206	CH1 User range settings offset value[*3]	0	R/W
CFH	207	CH1 User range settings gain value[*3]	0	R/W
D0H	208	CH2 User range settings offset value[*3]	0	R/W
D1H	209	CH2 User range settings gain value[*3]	0	R/W

[*1] This is the initial value set after the power is turned on or the programmable controller CPU is reset.

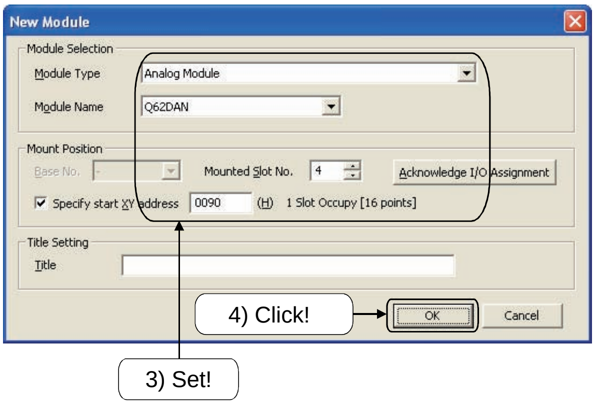
[*2] Indicates whether reading from and writing to a sequence program are enabled
R: Read enabled
W: Write enabled

[*3] Areas used to restore the user range settings offset/gain values when online module change is made.

7.6.4 Adding or setting intelligent function module data


MELSOFT Series GX Works2 C:\SCHOOL\SCHOOL\Q64AD - [[PRG]Write MAIN 9 Step]
Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help
New... Ctrl+N
Open... Ctrl+O
Close
Save Ctrl+S
Save As...
Compress/Unpack
Delete...
Verify...
Project Revision
Change PLC Type...
Change Project Type
Object
Intelligent Function Module
Open Other Data
Export to GX Developer Format File...
Library
Security
[PRG]Write MAIN 9 Step
X3
X80
X8E
0
8
1) Click!
New Module... Ctrl+Shift+E
Delete Module
Property...
Save the Positioning Module Data...
Read from the Positioning Module Data...

1) Click [Project] → [Intelligent Function Module] → [New Module].



Module Type: Analog Module

Module Name: Q62DAN

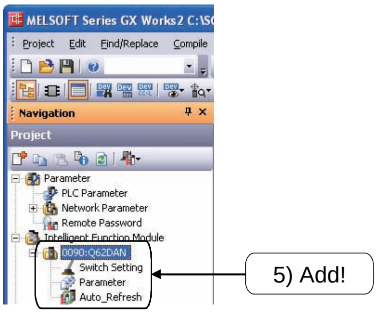
Mounted Slot No.: 4

☑ Specify start XY address: 0090 (H)

2) The New Module dialog box is displayed.

3) Set the A/D converter module setting as follows.
Module Type : Analog Module
Module Name : Q62DAN
Mounted Slot No. : 4
(Specify start XY address: 0090)

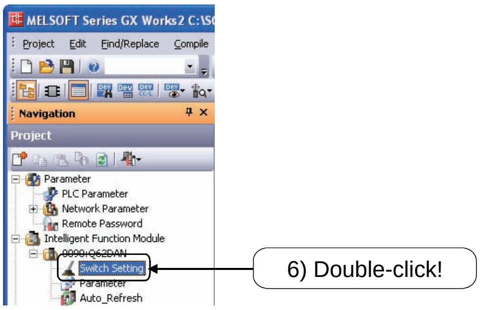
4) Click the OK button.



5) The specified intelligent function module data are added to the Project window.

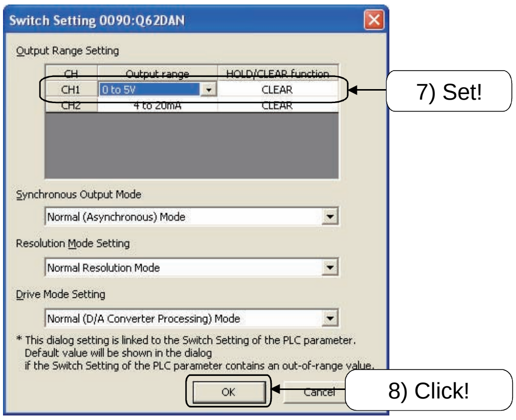
(To the next page)

(From the previous page)



6) Double-click!

6) Double-click Switch Setting.



Output Range Setting

CH	Output range	HOLD/CLEAR function
CH1	0 to 5V	CLEAR
CH2	4 to 20mA	CLEAR

Synchronous Output Mode: Normal (Asynchronous) Mode

Resolution Mode Setting: Normal Resolution Mode

Drive Mode Setting: Normal (D/A Converter Processing) Mode

* This dialog setting is linked to the Switch Setting of the PLC parameter.
Default value will be shown in the dialog
if the Switch Setting of the PLC parameter contains an out-of-range value.

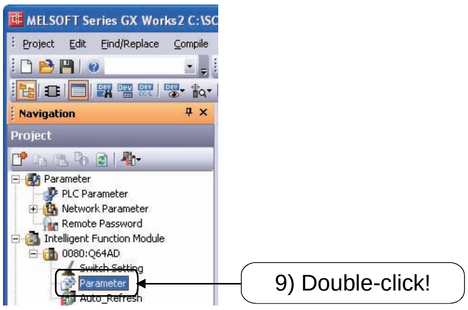
OK

7) Set!

8) Click!

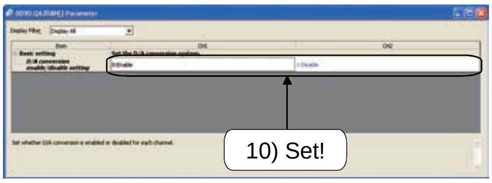
7) The Switch Setting screen is displayed.
Set Output range for CH1 to "0 to 5V".

8) Click the OK button.



9) Double-click!

9) Double-click Parameter.



Item	CH1	CH2
Basic setting	Set the D/A conversion system.	
D/A conversion enable/disable setting	0:Enable	1:Disable

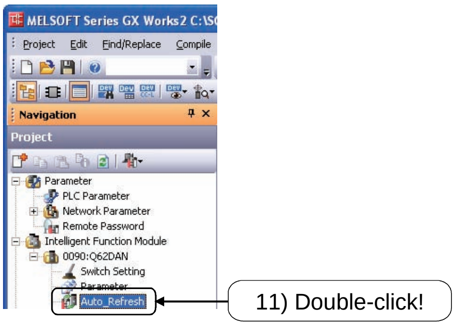
Set whether D/A conversion is enabled or disabled for each channel.

10) Set!

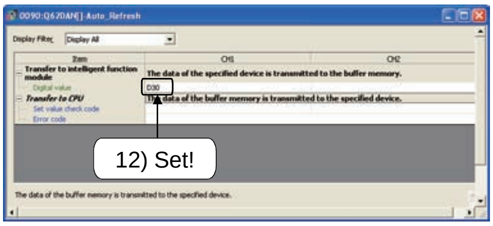
10) The Parameter screen is displayed.
Set "D/A conversion enable/disable setting" for CH1 to "0:Enable". (Only CH1 is used.)

(To the next page)

(From the previous page)



11) Double-click Auto_Refresh.

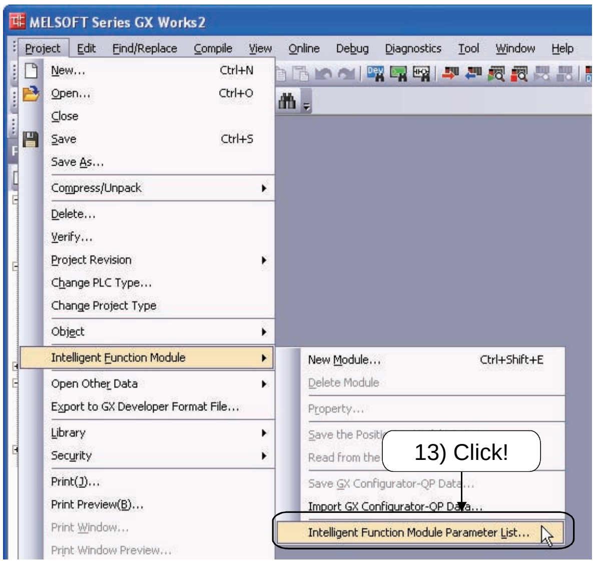

0090:Q62DAN[]-Auto_Refresh

Display Filter: Display All

Item	CH1	CH2
Transfer to intelligent function module	The data of the specified device is transmitted to the buffer memory.	
Digital value	D30	
Transfer to CPU	The data of the buffer memory is transmitted to the specified device.	
Set value check code		
Error code		

The data of the buffer memory is transmitted to the specified device.

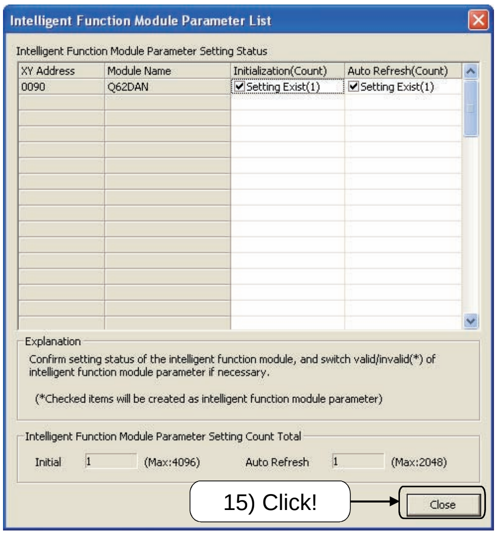
12) The Auto_Refresh screen is displayed.
Set Digital value for CH1 to "D30".



13) Click [Project] → [Intelligent Function Module]
→ [Intelligent Function Module Parameter List].

(To the next page)

(From the previous page)



Intelligent Function Module Parameter List

Intelligent Function Module Parameter Setting Status

XY Address	Module Name	Initialization(Count)	Auto Refresh(Count)
0090	Q62DAN	☑ Setting Exist(1)	☑ Setting Exist(1)

Explanation

Confirm setting status of the intelligent function module, and switch valid/invalid(*) of intelligent function module parameter if necessary.

(*Checked items will be created as intelligent function module parameter)

Intelligent Function Module Parameter Setting Count Total

Initial 1 (Max:4096) Auto Refresh 1 (Max:2048)

14) Check that "Setting Exist" is checked in Initialization (Count) and Auto Refresh (Count) for Q62DAN in the Intelligent Function Module Parameter List dialog box.

15) Click the Close button.

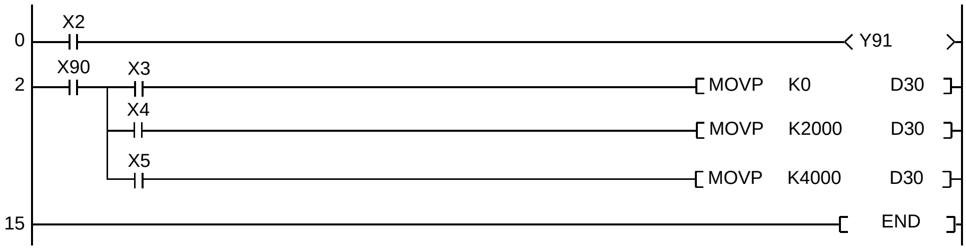
7.6.5 Exercise with the demonstration machine

(1) Sequence program

The sequence program converts values of the digital switches to analog signals.

Set the start XY to 90 and the digital value for CH1 to D30 for Q62DAN as explained before.

Project name	Q62DAN
Program name	MAIN



X90: Module READY signal

At power-on or reset of a programmable controller CPU, this signal turns on if D/A conversion is ready to be executed. D/A conversion is executed once this signal turned on.

Y91: CH1 Output enable/disable flag

Turning this flag on or off selects on each channel whether to output D/A converted values or offset values.

ON: D/A converted value, OFF: Offset value

(2) Operation of the demonstration machine

Stop the CPU and click on the toolbar.

The Online Data Operation dialog box is displayed. Click the Parameter + Program button, then click the Execute button to write data to the CPU. After that, activate the CPU and check the following items.

(a) Turn on X2 to enable D/A outputs of CH1.

(b) Voltage is output according to X3 to X5.

(c) The D/A OUTPUT voltmeter displays the voltage value that the D/A converter module outputs.

MEMO

CHAPTER 8 SIMULATION FUNCTION

8.1 Simulation Function

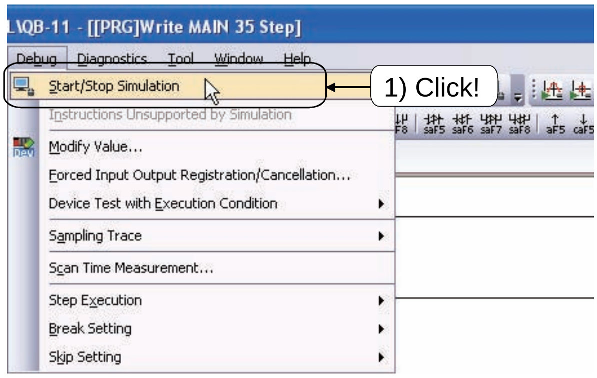
The simulation function is for debugging a sequence program using the virtual programmable controller on a personal computer.
The created sequence program can be immediately debugged without connecting a programmable controller CPU.

NOTE

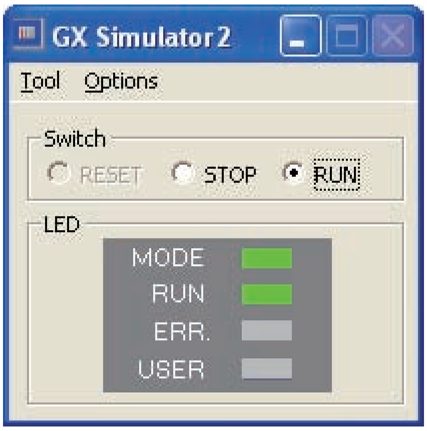
Safety and handling precautions of the simulation function

1) The simulation function simulates the actual programmable controller CPU to debug a created sequence program. However, this function does not guarantee the operation of the debugged sequence program.
2) The simulation function uses the memory for simulation to input and output data to/from the I/O module and intelligent function module. Some instructions, functions, and device memories are not supported. Therefore, the operation results obtained from the virtual programmable controller may differ from those obtained from the actual programmable controller CPU.

8.2 Starting/Stopping Simulation


\QB-11 - [[PRG]Write MAIN 35 Step]
Debug
Diagnostics
Tool
Window
Help
Start/Stop Simulation
1) Click!
Instructions Unsupported by Simulation
Modify Value...
Forced Input Output Registration/Cancellation...
Device Test with Execution Condition
Sampling Trace
Scan Time Measurement...
Step Execution
Break Setting
Skip Setting

1) Click [Debug] → [Start/Stop Simulation].



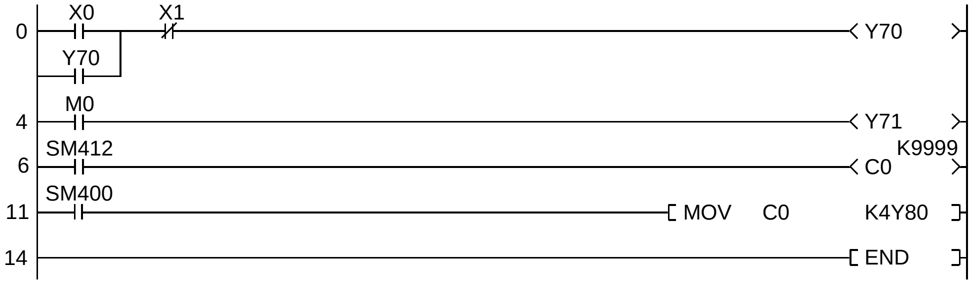
2) The GX Simulator2 screen is displayed, and the simulation starts.

3) To stop the simulation, click [Debug] → [Start/Stop Simulation] again.

8.3 Debugging with Example Program

Use the following example for exercise.

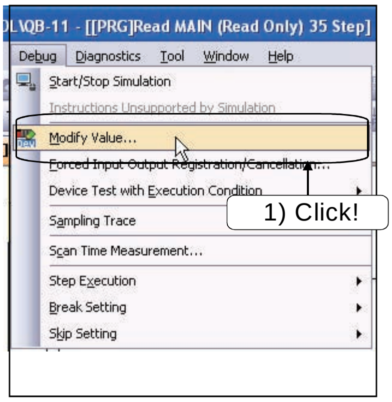
<<Example program>>



8.3.1 Monitoring and testing device status

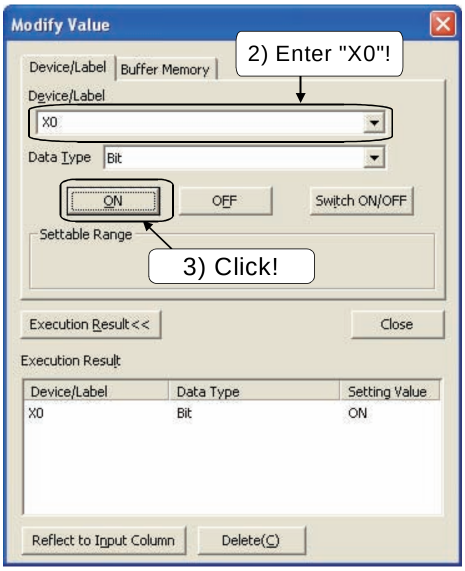
This section explains how to monitor device status, turn bit devices on/off forcibly, and change word device values.

(1) Turning bit devices on/off forcibly
In the example operation below, "X0" is forcibly turned on.


DL\QB-11 - [[PRG]Read MAIN (Read Only) 35 Step]
Debug
Diagnostics
Tool
Window
Help
Start/Stop Simulation
Instructions Unsupported by Simulation
Modify Value...
Forced Input Output Registration/Cancellation...
Device Test with Execution Condition
Sampling Trace
Scan Time Measurement...
Step Execution
Break Setting
Skip Setting

1) Click!

1) Click [Debug] → [Modify Value].



2) Enter "X0"!

Device/Label: X0

Data Type: Bit

3) Click!

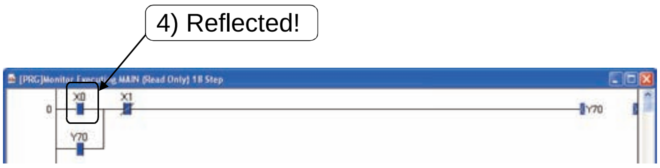
Execution Result

Device/Label	Data Type	Setting Value
X0	Bit	ON

2) The Modify Value dialog box is displayed. Input "X0" to the "Device/Label" list box.

3) Click the ON button to forcibly turn "X0" on.

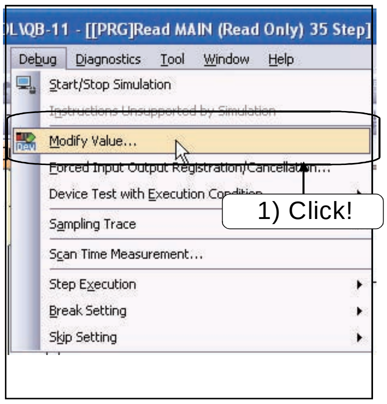
4) Reflected!



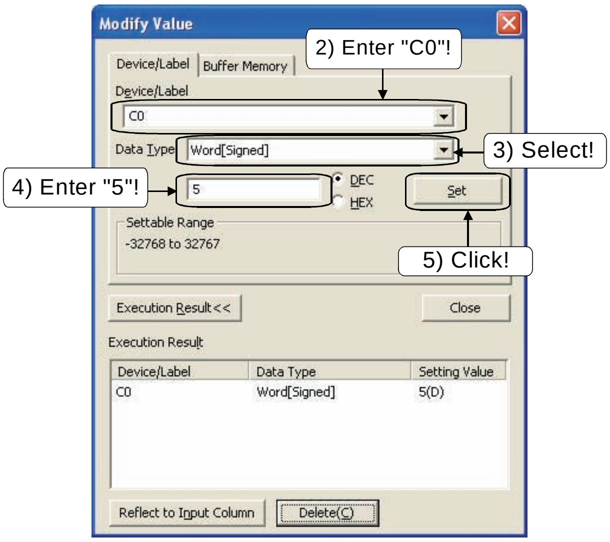
4) The result of the device being turned on is reflected on the ladder monitor screen.

(2) Changing the word device value

In the example operation below, the word device value "C0" is changed to "5".


L\QB-11 - [[PRG]Read MAIN (Read Only) 35 Step]
Debug
Diagnostics
Tool
Window
Help
Start/Stop Simulation
Instructions Unsupported by Simulation
Modify Value...
Forced Input Output Registration/Cancellation...
Device Test with Execution Condition
Sampling Trace
Scan Time Measurement...
Step Execution
Break Setting
Skip Setting
1) Click!

1) Click [Debug] → [Modify Value].



Device/Label: C0

Data Type: Word[Signed]

Value: 5

Execution Result

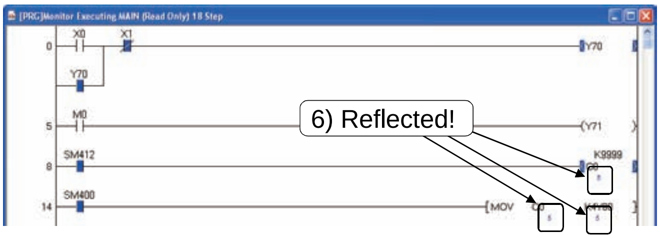
Device/Label	Data Type	Setting Value
C0	Word[Signed]	5(D)

2) The Modify Value dialog box is displayed. Input "C0" to the "Device/Label" list box.

3) Select the "Word[Signed]" from the "Data Type" list box.

4) Input "5" to the "Value" column.

5) After the setting is completed, click the Set button to forcibly change the current value of C0 to 5.



6) The change of the value of "C0" to "5" is reflected on the ladder monitor screen.

CHAPTER 9 MAINTENANCE

9.1 Typical Trouble

The following bar graph shows the ratio of faulty parts and causes of programmable controller errors.
[Source: Inspection made by JEMA (The Japan Electrical Manufacture's Association)]

Figure 9.1 Faulty parts on programmable controllers (multiple answers allowed)

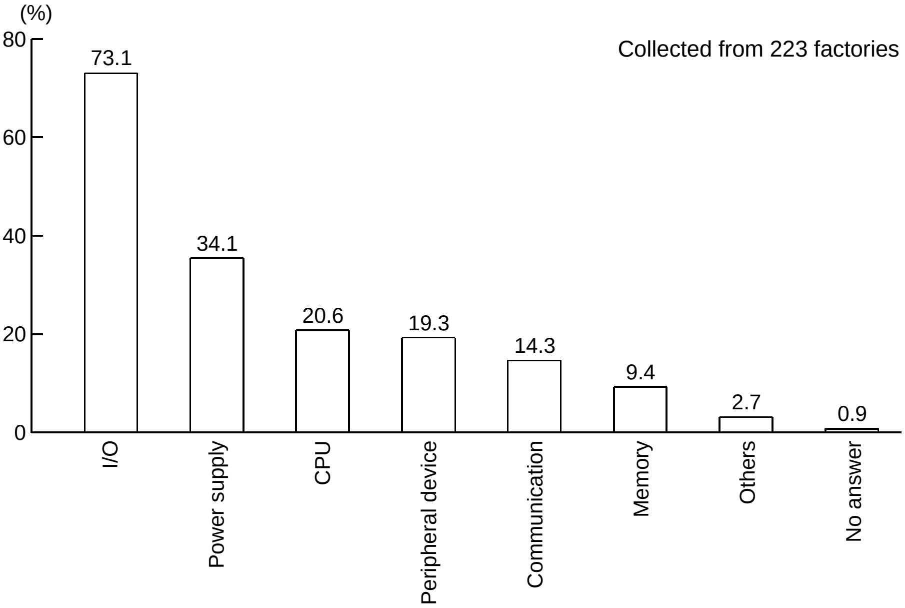
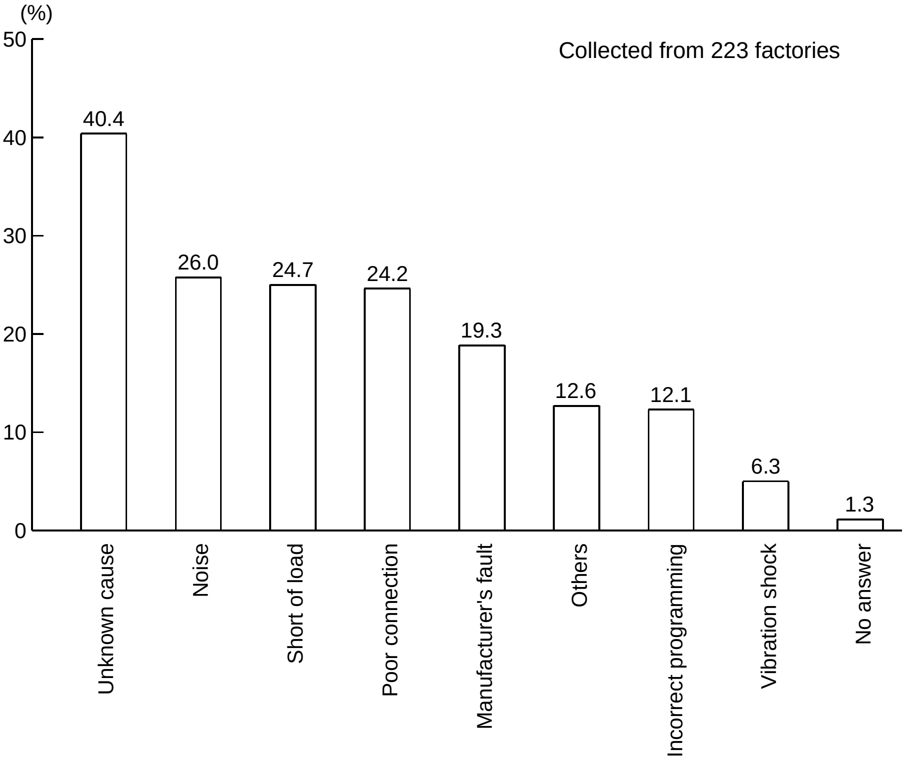


Figure 9.2 Causes of programmable controller faults (multiple answers allowed)



9.2 Maintenance

To keep programmable controllers in the best operating condition, conduct the following daily inspection and periodic inspection.

(1) Daily inspection
The following table lists the items that must be inspected daily.

Table 9.1 Daily inspection

Item	Inspection item		Inspection contents	Judgment criterion	Measures
1	Installation of base unit		Check that fixing screws are not loose and the cover is not dislocated.	The screws and cover must be installed securely.	Retighten the screws.
2	Installation of I/O module		Check that the module is not dislocated and the module fixing hook is engaged securely.	The module fixing hook must be engaged and installed securely.	Securely engage the module fixing hook. Or tighten the screw.
3	Connection conditions		Check for loosening of the terminal screws.	Screws must not be loose.	Retighten the terminal screws.
			Check for the distance between solderless terminals.	The proper distance must be provided between solderless terminals.	Set the proper distance.
			Check the connector part of the cable.	Connectors must not be loose.	Retighten the connector fixing screws.
4	Module indication LED	Power supply module "POWER" LED	Check that the LED is on.	The LED must be on. (Error if the LED is off)	Refer to QCPU (Q mode) User's Manual.
		CPU "RUN" LED	Check that the LED is on in RUN status.	The LED must be on. (Error if the LED is off)	
		CPU "ERROR" LED	Check that the LED is off.	The LED must be off. (Error if the LED is on or flashing)	
		CPU "BAT.ARM" LED	Check that the LED is off.	The LED must be off. (Error if the LED is on)	
		Input LED	Check that the LED turns on and off.	The LED must be on when the input power is turned on. The LED must be off when the input power is turned off. (Error if the LED does not turn on or turn off as indicated above)	
		Output LED	Check that the LED turns on and off.	The LED must be on when the output power is turned on. The LED must be off when the output power is turned off. (Error if the LED does not turn on or turn off as indicated above)	

(2) Periodic inspection

The following table lists the items that must be inspected one or two times every half year to a year. When the equipment has been relocated or modified, or wiring layout has been changed, perform this inspection.

Table 9.2 Periodic inspection

Item	Inspection item		Inspection contents	Judgment criterion	Measures
1	Ambient environment	Ambient temperature	Measure the temperature and humidity with a thermometer and a hygrometer. Measure corrosive gas.	0 to 55 °C	When the programmable controller is used in the board, the ambient humidity in the board is the ambient humidity.
		Ambient humidity		5 to 95% RH*1	
		Ambience		Corrosive gas must not be present.	
2	Power supply voltage		Measure the voltage across the terminals of 100/200VAC.	85 to 132VAC	Change the power supply.
				170 to 264VAC	
3	Installation	Looseness, rattling	Move the module to check for looseness and rattling.	The module must be installed securely.	Retighten the screws. If the CPU, I/O, or power supply module is loose, fix it with screws.
		Adhesion of dirt and foreign matter	Check visually.	Dirt and foreign matter must not be present.	Remove and clean the dirt and foreign matter.
4	Connection conditions	Looseness of terminal screws	Retighten screws with a screwdriver.	Screws must not be loose.	Retighten the terminal screws.
		Distance between solderless terminals	Check visually.	The proper distance must be provided between solderless terminals.	Set the proper distance.
		Looseness of connectors	Check visually.	Connectors must not be loose.	Retighten the connector fixing screws.
5	Battery		Check that SM51 or SM52 is turned off with GX Works2.	(Preventive maintenance)	Even if the lowering of a battery capacity is not displayed, replace the battery with a new one if the specified service life of the battery is exceeded.
6	Spare product		Install the product on the actual programmable controller and check the operation.	The operation must meet the specifications.	Use the normal product on the actual programmable controller as a spare product.
7	Check the stored program		Compare the stored program with the running program.	The two programs must be identical.	Correct if any difference is found.
8	Fan (heat exchanger) filter		Rotation status Rotation sound Clogging	The fan must rotate without abnormal sounds. The fan must rotate without clogging.	Replace if any error is found. Clean.
9	Analog I/O		Check the offset/gain value.	The value must be identical with the specifications (design value).	Correct if any difference is found.

*1: When AnS Series Module is used in the system, the judgment criteria will be from 10 to 90% RH.

9.3 Consumable Product

Backup batteries on programmable controllers are consumable products.

9.4 Service Life of Output Relay

The output relays of the modules are consumed by the switching operation.
A relay which is directly mounted on the print board of the output module is required to be replaced the output module itself after the consumption.

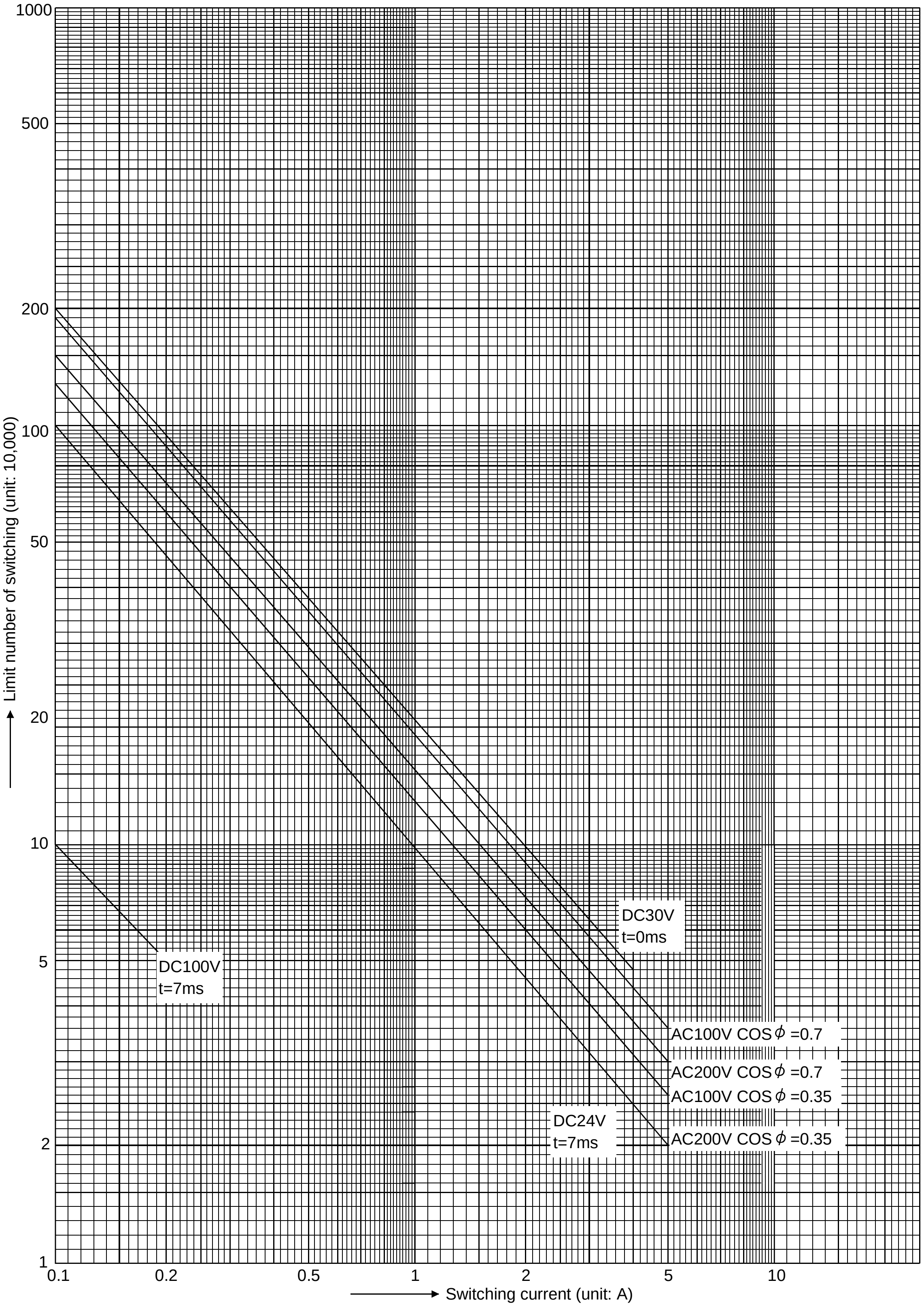


Figure 9.3 Life characteristics of output relay's contact (QY10, QY18A)

9.5 Spare Product

Alternative products are easily purchased through Mitsubishi service centers or local Mitsubishi representatives in Japan. Thus the alternative products can be prepared even after an accident. However, note that for foreign-related products such as exported products, alternative products must be sent beforehand.
Considering the following tips at design work makes the maintenance easier.

(1) Easily replaceable type
Replacing building block-type modules is easy. Only replacing the faulty module is required.

(2) Memory type
To use standard RAMs or SRAM memory cards, backup batteries are required. The standard ROMs, Flash cards, and ATA cards do not require the battery for use, besides, these memories prevent unintentional program changes due to human-related mistakes. These memories are recommended to be employed in products for export.

(3) Reducing the number of module types
Reducing the number of module types is efficient for reducing the number of spare product types.

(4) Reserving I/O points
By not using all the I/O points on 16-, 32-, and 64-point I/O modules but reserving 10% to 20% of them, it is possible to just make changes on wiring and programs (I/O signals) instead of replacing the faulty module with a spare module when there are no spare modules.

(5) Creating a document
Since sequence programs are easily modified, the inconsistency between an operating program and documents may occur (i.e. ladder diagram, program list). Keep updating the document.
To do this, using a printer is efficient.

(6) Mastering peripheral device
Mastering peripheral device such as a personal computer, GX Works2 helps the quick recovery from an accident.

(7) Spare product

Table 9.3 Spare products

	Product name	Quantity	Remark
1	Battery	One or two	Storage lives of lithium batteries are about five years. Therefore, the stock should not be kept all the time but batteries should be purchased when required. However, keep stock of one or two for accidental situation.
2	I/O module	One per each module type	Note that I/O modules tend to be faulty during a test operation. Also note that the contacts of output modules are consumed in long-term use.
3	CPU module	One for each used model	CPU modules and memory cards are the core parts of a programmable controller, which means that an error of them result in the system down.
4	Memory card	One for each used model	
5	Power supply module	One for each used model	Same as above. As the temperature of the power supply modules rises easily, and high ambient temperature may shorten their service lives.

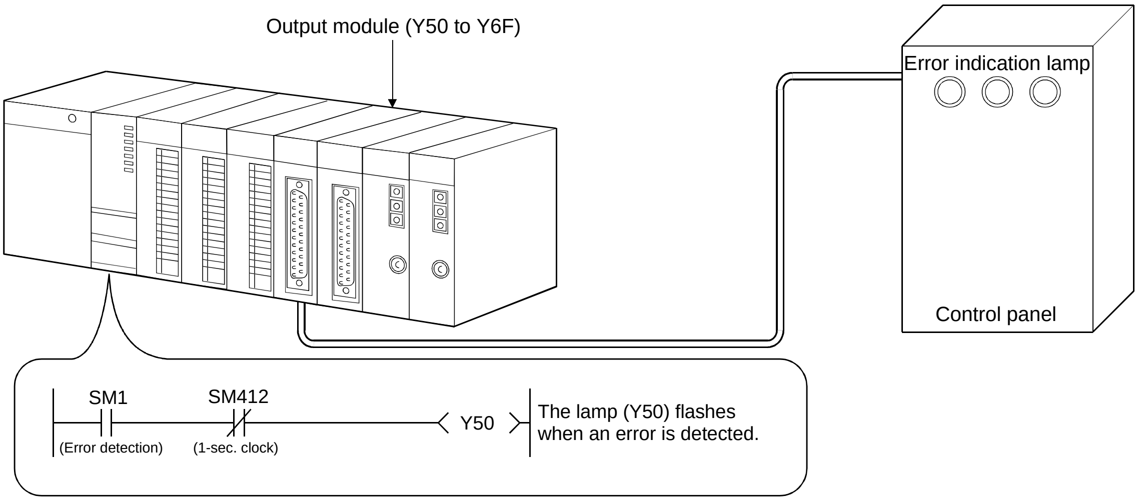
9.6 Using Support Equipment

The following shows examples of support equipment in which programmable controller-used systems or devices automatically notify a detected failure or operation status to an operator or maintenance personnel during an automatic control operation.

1. Displaying an error using a commercial lamp

Connect the error lamp to the output module of the programmable controller so that the lamp flashes when an error is detected.

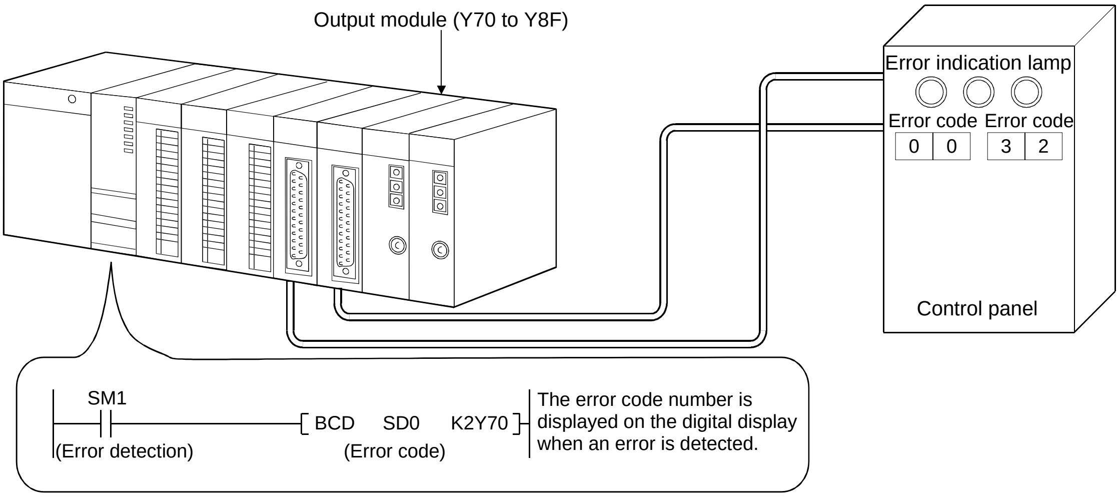
Lamp flicker



2. Displaying an error code on a commercial digital display

Connect the digital display to the output module of the programmable controller so that the error code number of the detected error is indicated on the digital display.

Numerical display


Output module (Y70 to Y8F)
Error indication lamp
Error code Error code
0 0
3 2
Control panel
SM1
(Error detection)
BCD SD0 K2Y70
(Error code)
The error code number is displayed on the digital display when an error is detected.

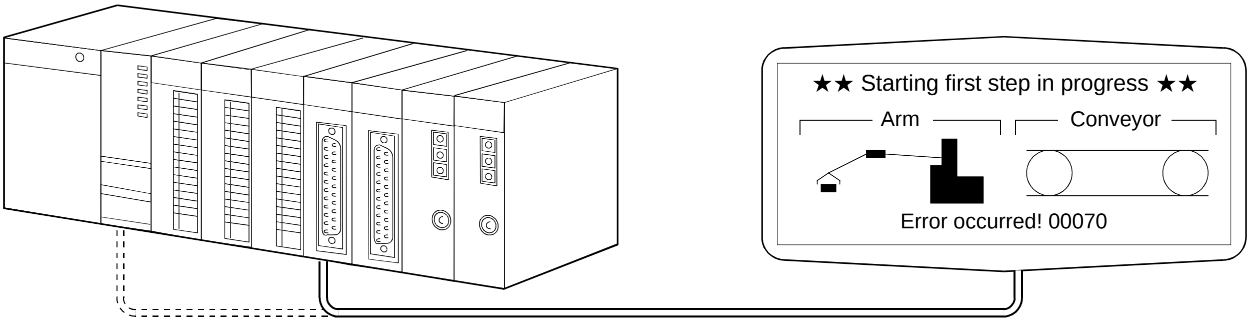
NOTE

The above programs cannot be executed when a stop error occurs.

3. Displaying the contents of the detected error on the screen

The errors details of the programmable controller can be displayed on an external CRT screen, plasma screen, and liquid crystal screen.

Screen display



★★ Starting first step in progress ★★

Error occurred! 00070

MELSEC-Q supports a wide variety of GOTs (Graphic Operation Terminals).
In addition to the error display function, GOTs have a lot of useful functions such as the graphic monitoring, ladder monitoring, device monitoring, touch-panel switch, and printing function.
(Refer to the catalogs for details.)

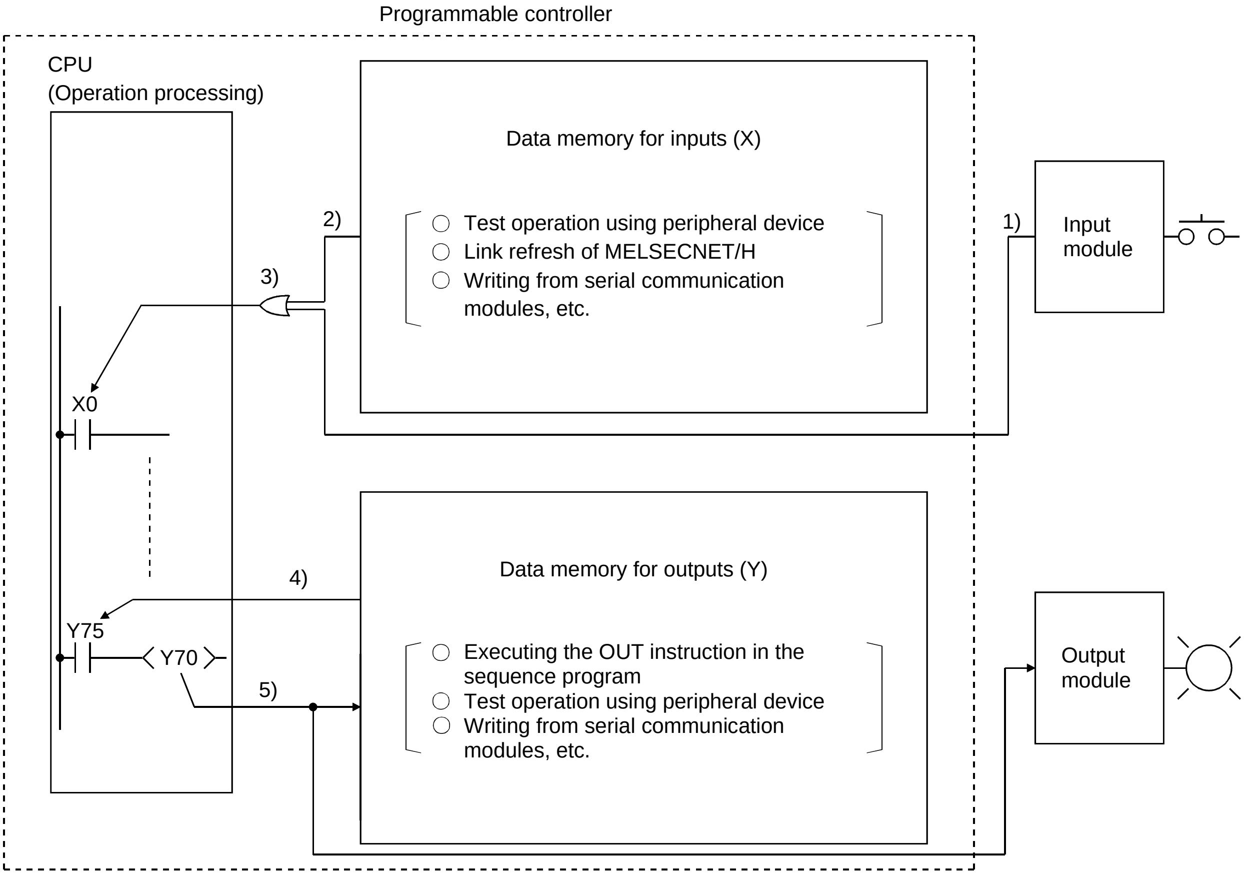
APPENDIX

Appendix 1 I/O Control Mode

The CPU supports two types of I/O control modes; the direct mode and refresh mode.

Appendix 1.1 Direct mode

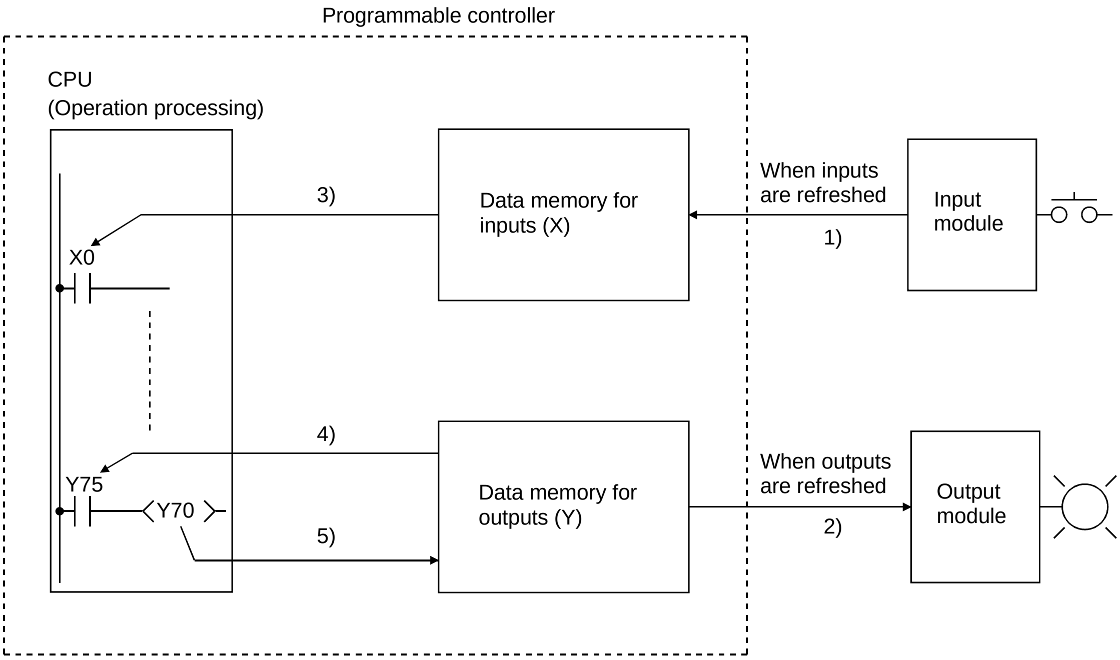
In the direct mode, input signals are imported to a programmable controller every time they are input and treated as input information. The operation results of a program are output to the output data memory and the output modules. The following diagram shows the flow of I/O data in the direct mode.


Programmable controller
CPU
(Operation processing)
Data memory for inputs (X)
○ Test operation using peripheral device
○ Link refresh of MELSECNET/H
○ Writing from serial communication modules, etc.
1)
2)
3)
X0
Input module
Data memory for outputs (Y)
○ Executing the OUT instruction in the sequence program
○ Test operation using peripheral device
○ Writing from serial communication modules, etc.
4)
5)
Y75
Y70
Output module

- When the input contact instruction is executed:
  An OR operation is executed in the input information 1) from the input module and input information 2) in the data memory. Then the result is used as input information 3) at sequence program execution.
- When the output contact instruction is executed:
  Output information 4) is read from the data memory for output (Y), and a sequence program is executed.
- When the output OUT instruction is executed:
  The operation result 5) of the sequence program is output to the output module, and is stored in the data memory for output (Y).
- When the QCPU executes I/O in the direct mode, a sequence program uses DX for inputs and DY for outputs.

Appendix 1.2 Refresh mode

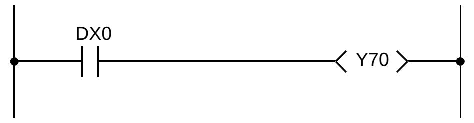
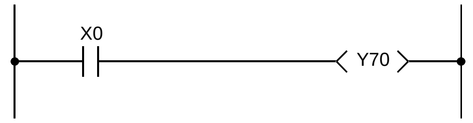
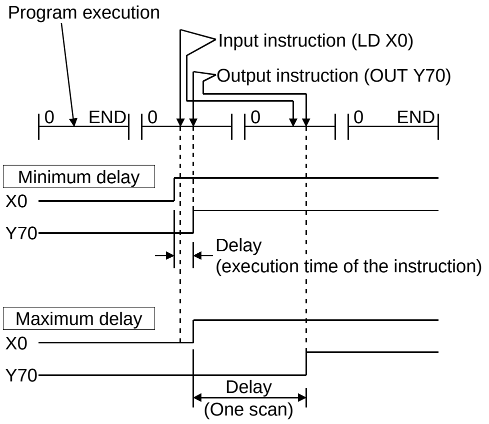
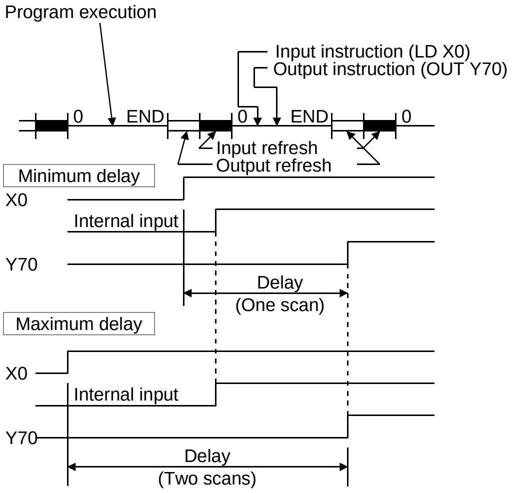
In the refresh mode, all changes caused in an input module are imported to the input data memory in a programmable controller CPU before every scan. The data in the data memory is used for an operation.
The operation results made in a program for output (Y) are stored to the output data memory at every operation. All the data stored in the output data memory is batch-output to the output module after the execution of the END instruction.
The following diagram shows the flow of I/O data in the refresh mode.


Programmable controller
CPU
(Operation processing)
3)
X0
Data memory for inputs (X)
When inputs are refreshed
1)
Input module
4)
Y75
Y70
5)
Data memory for outputs (Y)
When outputs are refreshed
2)
Output module

- Input refresh
  Input data in the input module is batch-read 1) before the execution of the step 0, and stored to the data memory for input (X).
- Output refresh
  Data 2) in the data memory for output (Y) is batch-output to the output module before the execution of the step 0.
- When the input contact instruction is executed:
  The input data is read from the data memory for input (X) 3), and a sequence program is executed.
- When the output contact instruction is executed:
  The output data 4) is read from the data memory for output (Y), and a sequence program is executed.
- When the output OUT instruction is executed:
  The operation result of the sequence program 5) is stored in the data memory for output (Y).

Appendix 1.3 Comparisons between the direct mode and refresh mode

In the example ladder given below, turning on input X0 turns on output Y70.

Item	Direct mode	Refresh mode
1. Ladder example	 	
2. Response lag from when input is changed to when output is changed accordingly	 Program execution Input instruction (LD X0) Output instruction (OUT Y70) 0 END 0 0 0 END Minimum delay X0 Y70 Delay (execution time of the instruction) Maximum delay X0 Y70 Delay (One scan) • The delay time ranges from 0 (only execution time of the instruction) to 1 scan. • The delay time is 0 to 1 scan.	 Program execution Input instruction (LD X0) Output instruction (OUT Y70) 0 END 0 END 0 Input refresh Output refresh Minimum delay X0 Internal input Y70 Delay (One scan) Maximum delay X0 Internal input Y70 Delay (Two scans) • The delay time ranges from 1 to 2 scans. • The delay time is 1 to 2 scans.
3. Execution time of the I/O instruction	• The direct mode needs longer time than the refresh mode since a programmable controller accesses I/O modules.	• Generally, only short time is needed since a programmable controller accesses data memory.
4. Scan time	• The scan time is longer for the execution time of the I/O instructions. • The actual scan time is the program execution time.	• The scan time is shorter for the execution time of the I/O instructions. • The actual scan time is the total time of a program execution, input transfer, and output transfer.

Appendix 2 Special Relay

The special relay (SM) is an internal relay whose application is fixed in the programmable controller. For this reason, the special register cannot be used in the same way as other internal registers are used in sequence programs. However, the bit of the special relay can be turned on or off as needed to control the CPU module.

The following shows how to read the items in the list.
For details of special relays, refer to QCPU User's Manual Hardware Design, Maintenance and Inspection.

Item	Description
Number	• Indicates the special relay number.
Name	• Indicates the special relay name.
Meaning	• Indicates the contents of the special relay.
Explanation	• Explains the contents of the special relay in detail.
Set by (When set)	• Indicates the setting side and setting timing of the special register. <Set by> S : Set by the system U : Set by user (in sequence program or test operation at a peripheral device) S/U : Set by both system and user <When set> → indicated only if setting is done by system. Every END processing : Set during every END processing Initial : Set during initial processing (after power-on or status change from STOP to RUN) Status change : Set when the operating status is changed Error : Set if an error occurs Instruction execution : Set when an instruction is executed Request : Set when requested by a user (using the special relay) When system is switched : Set when the system is switched (between the control system and the standby system)
Corresponding ACPU M9□□□	• Indicates a special relay (M9□□□) supported by the ACPU. ("M9□□□ format change" indicates the one whose application has been changed. Incompatible with the Q00J/Q00/Q01, and QnPRH.) • "New" indicates the one added for the Q-series CPU.
Corresponding CPU	Indicates the CPU module supporting the special relay. QCPU : All the Q-series CPU modules Q00J/Q00/Q01 : Basic model QCPU Qn(H) : High Performance model QCPU QnPH : Process CPU QnPRH : Redundant CPU QnU : Universal model QCPU CPU module name : Only the specified CPU model (Example: Q02U)

For details on the following items, refer to these manuals:

- For network related items → Manuals for each network module
- For SFC programs → MELSEC-Q/L/QnA Programming Manual (SFC)

POINT

Do not change the values of special relays set by the system using a program or by test operation.
Doing so may result in a system down or communication failure.

Appendix 3 Special Register

The special register (SD) is an internal register whose application is fixed in the programmable controller. For this reason, the special register cannot be used in the same way as other internal registers are used in sequence programs. However, data can be written to the special register to control the CPU module as needed.
Data is stored in binary format if not specified.

The following shows how to read the items in the list.
For details of special registers, refer to QCPU User's Manual Hardware Design, Maintenance and Inspection.

Item	Description
Number	• Indicates the special register number.
Name	• Indicates the special register name.
Meaning	• Indicates the contents of the special register.
Explanation	• Indicates the detailed contents of the special register.
Set by (When set)	• Indicates the setting side and setting timing of the special register. <Set by> S : Set by the system U : Set by user (in sequence program or test operation at a peripheral device) S/U : Set by both system and user <When set> → indicated only if setting is done by system. Every END processing : Set during every END processing Initial : Set during initial processing (after power-on or status change from STOP to RUN) Status change : Set when the operating status is changed Error : Set if an error occurs Instruction execution : Set when an instruction is executed Request : Set only when there is request from a user (through SM, etc.) When system is switched : Set when the system is switched (between the control system and the standby system)
Corresponding ACPU D9□□□	• Indicates special register (D9□□□) supported by the ACPU. ("D9□□□ format change" indicates the one whose application has been changed. Incompatible with the Q00J/Q00/Q01, and QnPRH.) • "New" indicates the one added for the Q-series CPU.
Corresponding CPU	Indicates the CPU module supporting the special relay. QCPU : All the Q-series CPU modules Q00J/Q00/Q01 : Basic model QCPU Qn(H) : High Performance model QCPU QnPH : Process CPU QnPRH : Redundant CPU QnU : Universal model QCPU CPU module name : Only the specified CPU model (Example: Q02U)

For details on the following items, refer to these manuals:

- For network related items → Manuals for each network module
- For SFC programs → MELSEC-Q/L/QnA Programming Manual (SFC)

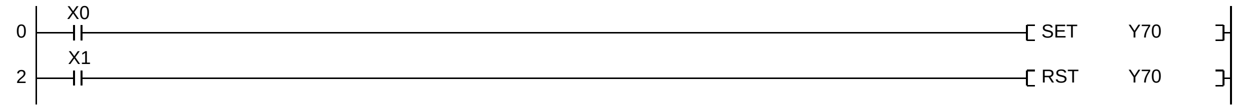
POINT

Do not change the values of special registers set by the system using a program or by test operation.
Doing so may result in a system down or communication failure.

Appendix 4 Application Program Example

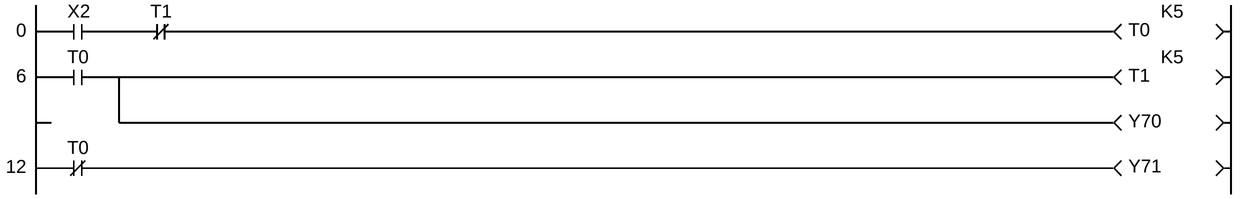
Appendix 4.1 Flip-flop ladder

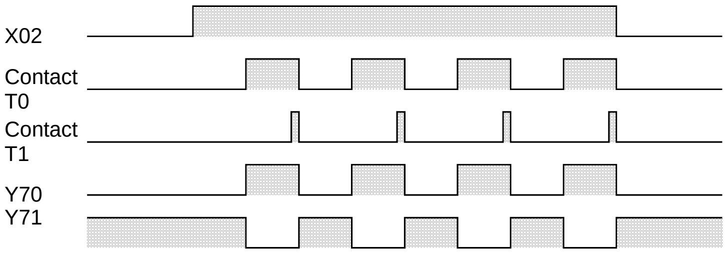
(1) Y70 turns on when X0 is turned on, and turns off when X1 is turned on.



(2) When X2 is turned on, Y71 turns off if Y70 is on, and turns on if Y70 is off. This flip-flop operation is repeated.

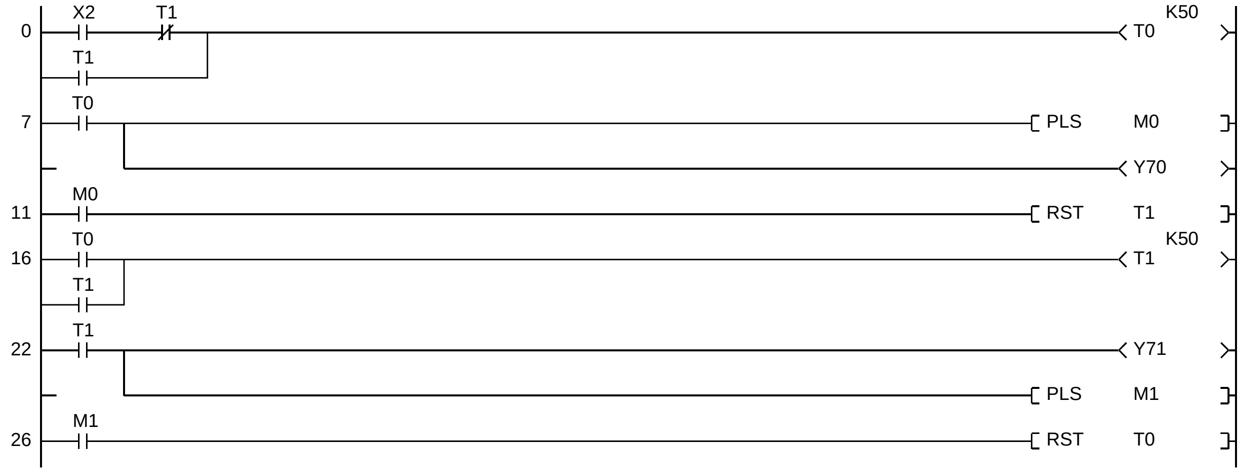
Project name	QA-16
Program name	MAIN

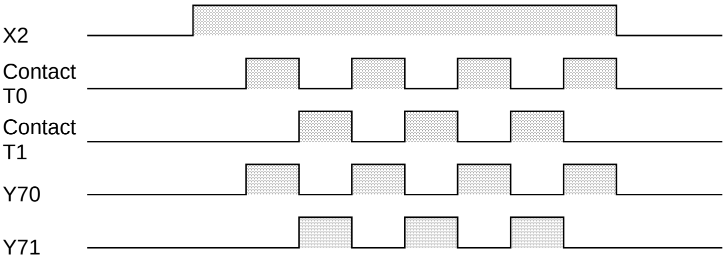




(3) The flip-flop operation starts when X2 is turned on. In this operation, Y70 turns on if the timer T0 is on, and Y71 turns on if the timer T1 is on. (Cycle: 10sec.)

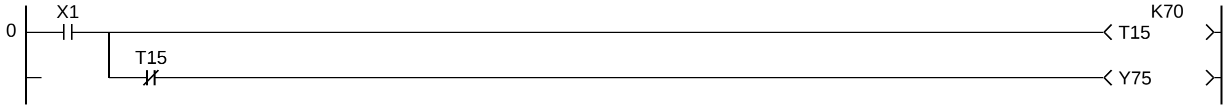
Project name	QA-17
Program name	MAIN

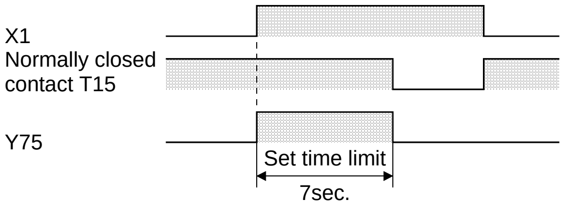




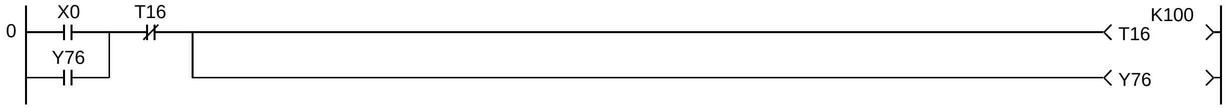
Appendix 4.2 One shot ladder

(1) Output starts and continues for a certain time after the input X1 is turned on. (Time for the input being on must be longer than the set time limit.)

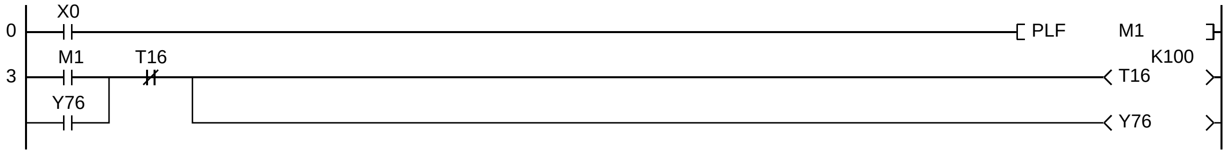


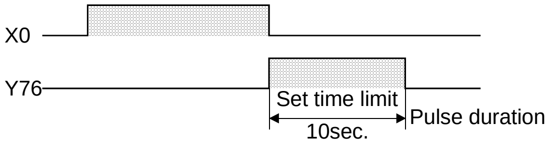


(2) When the input X0 is turned on momentarily, Y76 turns on for a certain time.



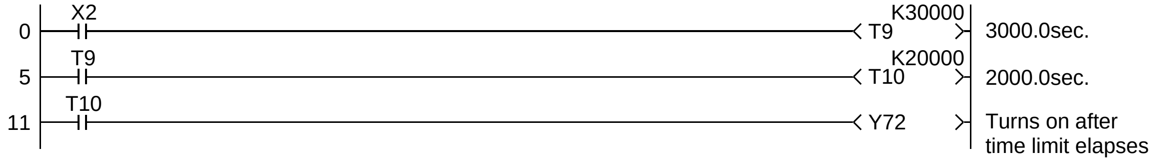
(3) Output starts and continues for a certain time when the input X0 is switched from on to off.

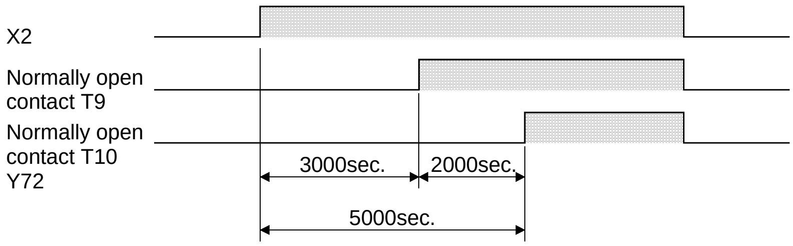




Appendix 4.3 Long-time timer

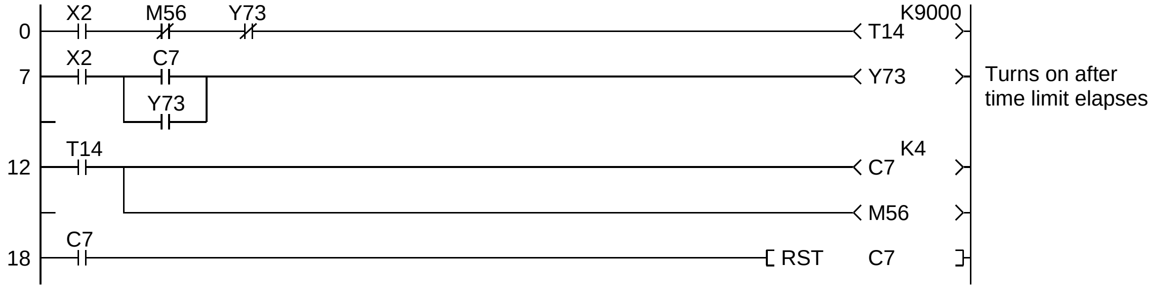
(1) Necessary time is obtained by connecting timers in serial.

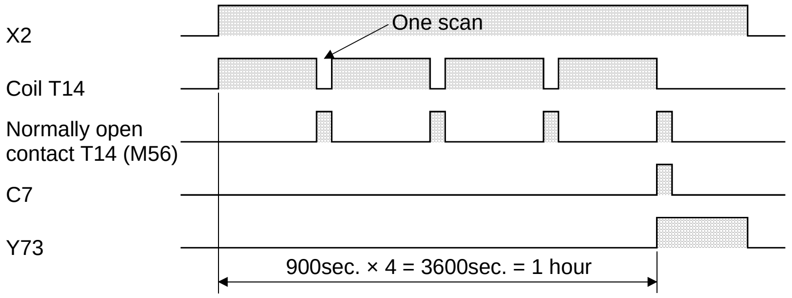




(2) Necessary time is obtained by using timers and counters.
Time limit of timer × Set value of counter = Long-time timer (note that accuracy of timers are accumulated.)

Project name	QA-18
Program name	MAIN



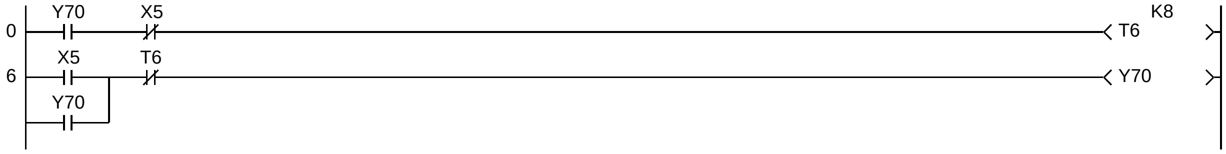

X2
One scan
Coil T14
Normally open contact T14 (M56)
C7
Y73
900sec. × 4 = 3600sec. = 1 hour

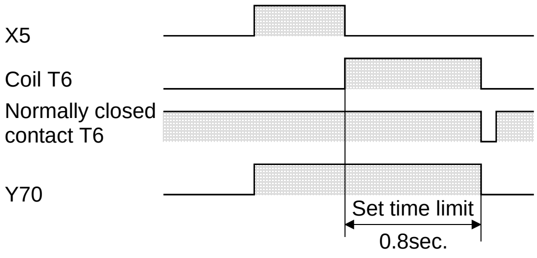
(Note) Sufficient time is obtained with the counter C7 which counts the number of time-outs of the timer T14.
M56 resets T14 after time-out. With C7, the output Y73 is self-energized while count up is in progress. With Y73, T14 is reset and the following time count is stopped.

Appendix 4.4 Off delay timer

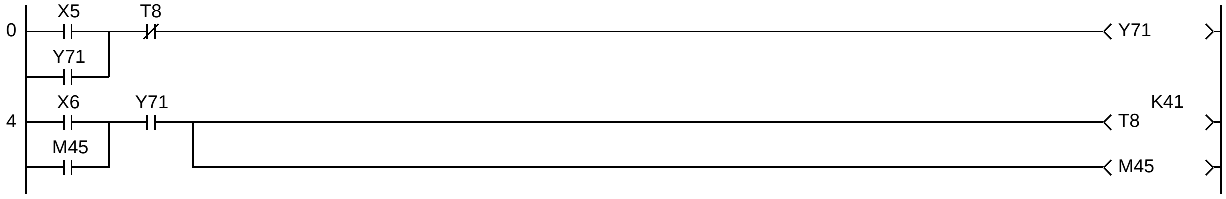
MELSEC-Q does not provide off delay timers. Configure an off delay timer as follows.

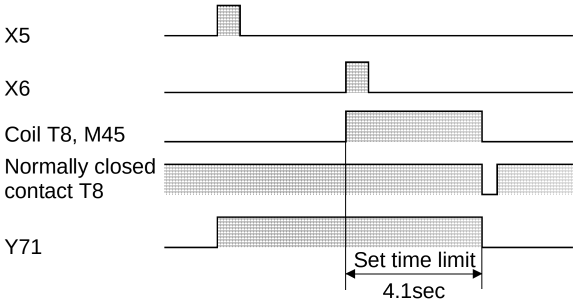
(1) The timer T6 starts operating when X5 is turned off.





(2) Turning on X5 momentarily sets the operation ready.
The timer T8 starts operating when X6 is momentarily turned on.



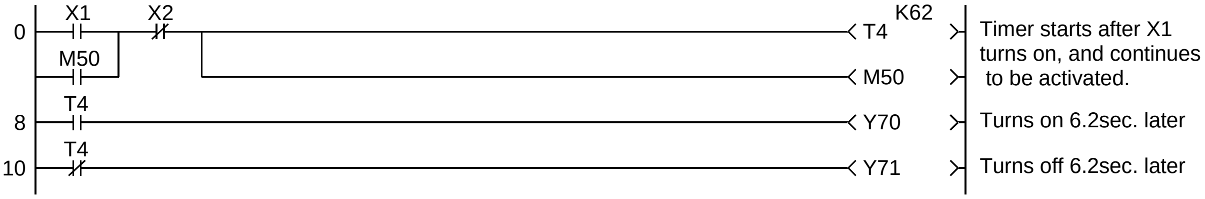


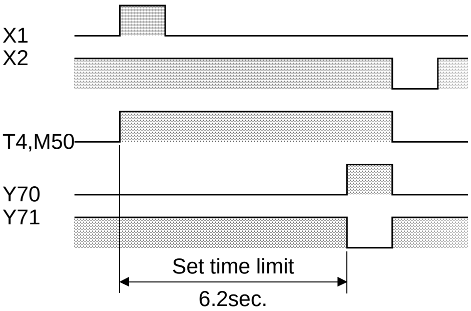
(Note) The above ladder operates as an off delay ladder by momentarily turning on inputs X5 and X6.
M45 is equivalent to a momentary contact of T8.

Appendix 4.5 On delay timer (momentary input)

An on delay timer of a programmable controller operates easily with a continuous input. A relay M must be used with a momentary input.

Project name	QA-19
Program name	MAIN

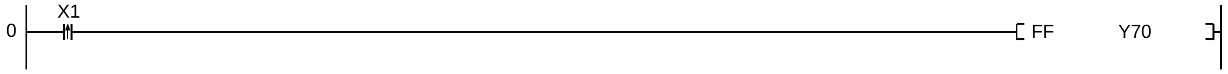




(Note) The above ladder operates as an on delay ladder by momentarily turning on inputs X1 and X2.

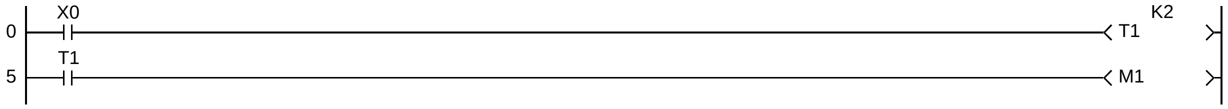
Appendix 4.6 ON-OFF repeat ladder

In an ON-OFF repeat ladder, Y70 turns on when X1 is turned on, and turns off when X1 is turned on again.



Appendix 4.7 Preventing chattering input

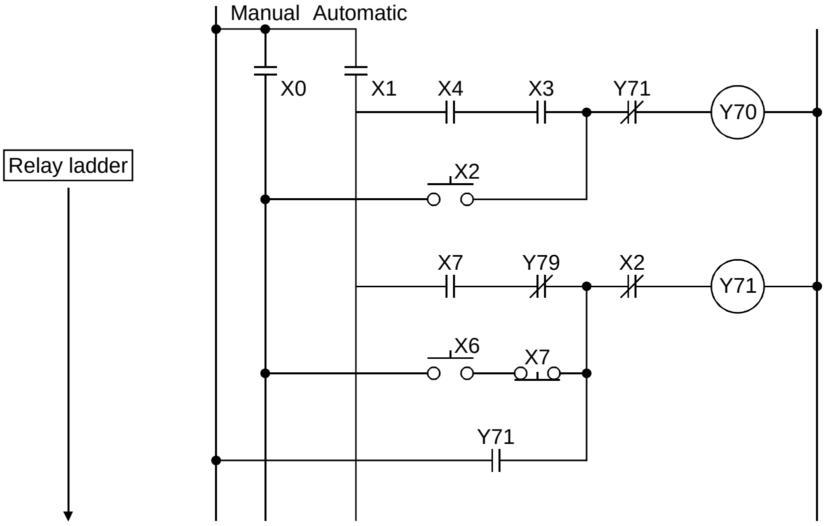
The timer is set so that it starts output when the input keeps being on for 0.2sec.



M1 turns on when X0 keeps being on for 0.2sec. or longer. Therefore, use M1 instead of X0 when creating a program.

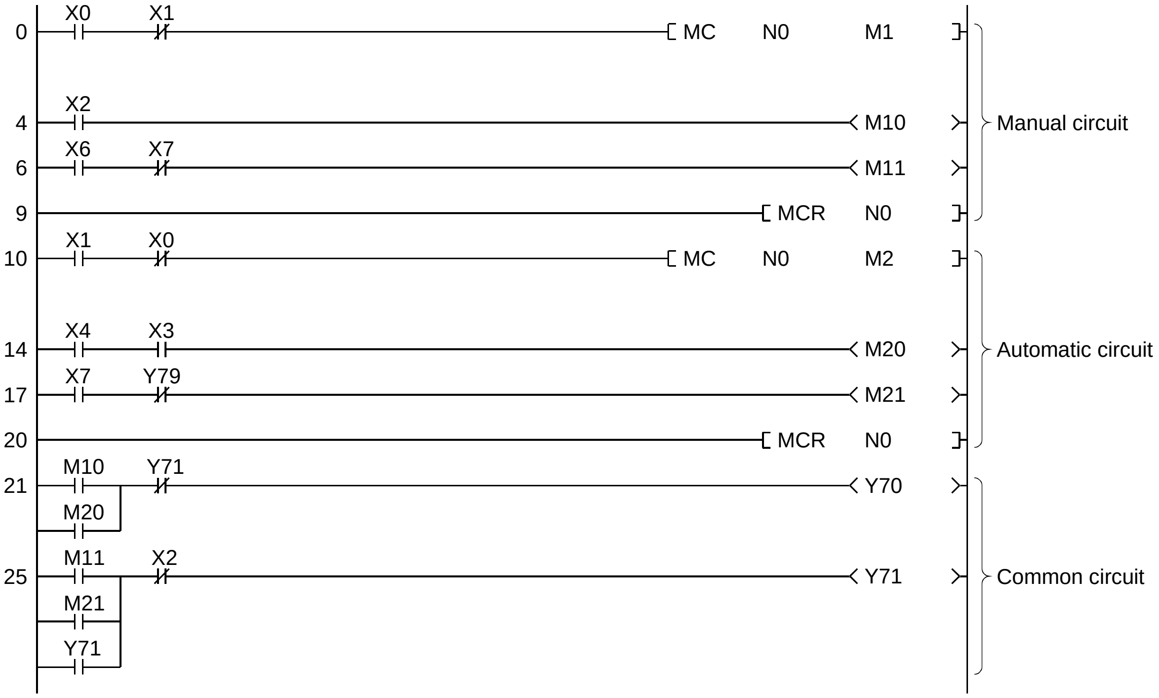
Appendix 4.8 Ladders with a common line

The following ladder cannot be operated as it is. To make such ladders controllable, use master control instructions (MC, MCR) in the program.



Project name	QA-1
Program name	MAIN

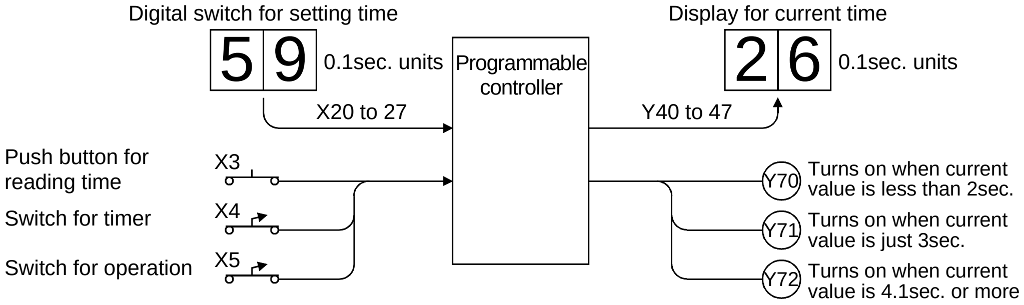
Sequence program with master control instructions



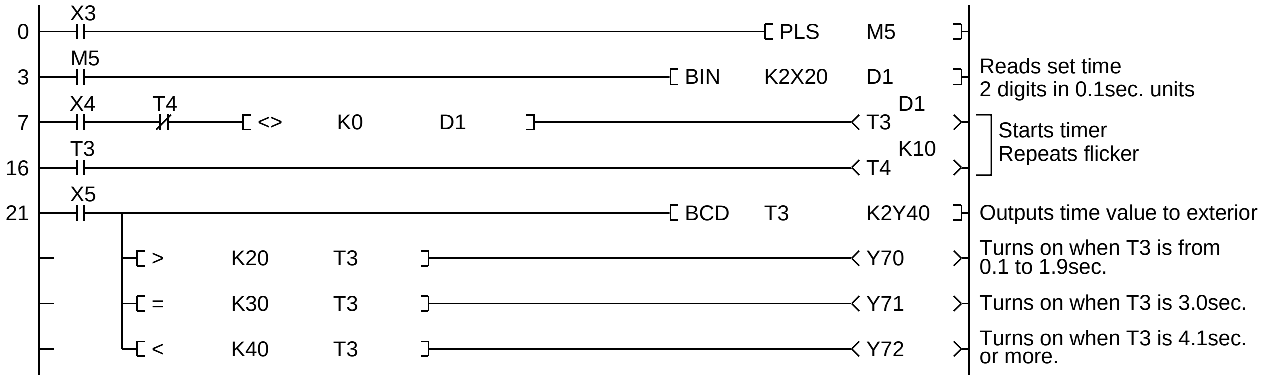
Note) In GX Works2, the on/off status of the master control is displayed in the title tag on the monitor screen.

Appendix 4.9 Time control program

The time value is set in the two digits of a digital switch. The currently elapsed time is displayed on Y40 to Y47 while the outputs Y70 to Y72 turn on after the set time limit has elapsed.
This operation is repeated.


Digital switch for setting time
5 9
0.1sec. units
X20 to 27
Programmable controller
Display for current time
2 6
0.1sec. units
Y40 to 47
Push button for reading time
X3
Switch for timer
X4
Switch for operation
X5
Y70
Turns on when current value is less than 2sec.
Y71
Turns on when current value is just 3sec.
Y72
Turns on when current value is 4.1sec. or more

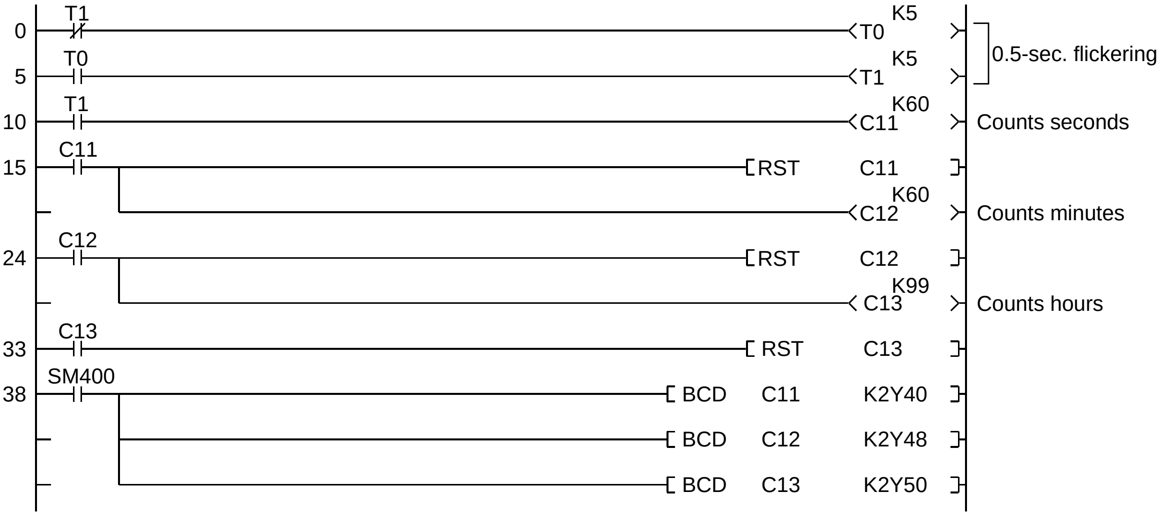
Project name	QA-2
Program name	MAIN

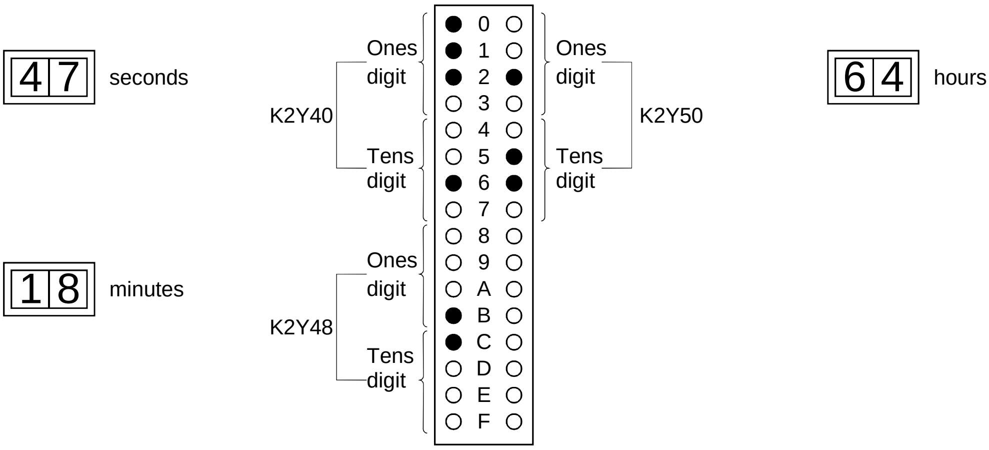

0
X3
PLS
M5
3
M5
BIN
K2X20
D1
Reads set time 2 digits in 0.1sec. units
7
X4
T4
<>
K0
D1
T3
D1
Starts timer
Repeats flicker
16
T3
T4
K10
21
X5
BCD
T3
K2Y40
Outputs time value to exterior
>
K20
T3
Y70
Turns on when T3 is from 0.1 to 1.9sec.
=
K30
T3
Y71
Turns on when T3 is 3.0sec.
<
K40
T3
Y72
Turns on when T3 is 4.1sec. or more.

Appendix 4.10 Clock ladder

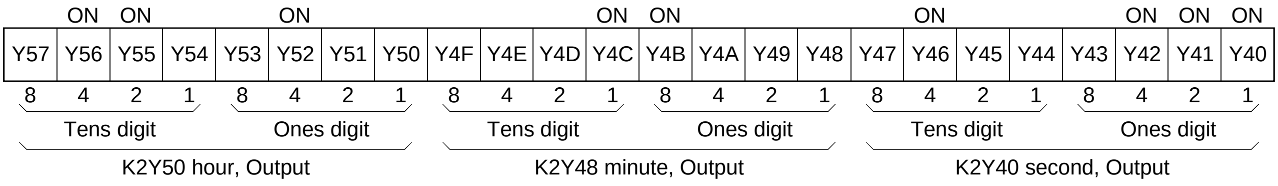
The clock data such as hour, minute, and second is output to a digital display.

Project name	QA-3
Program name	MAIN





LED of output module

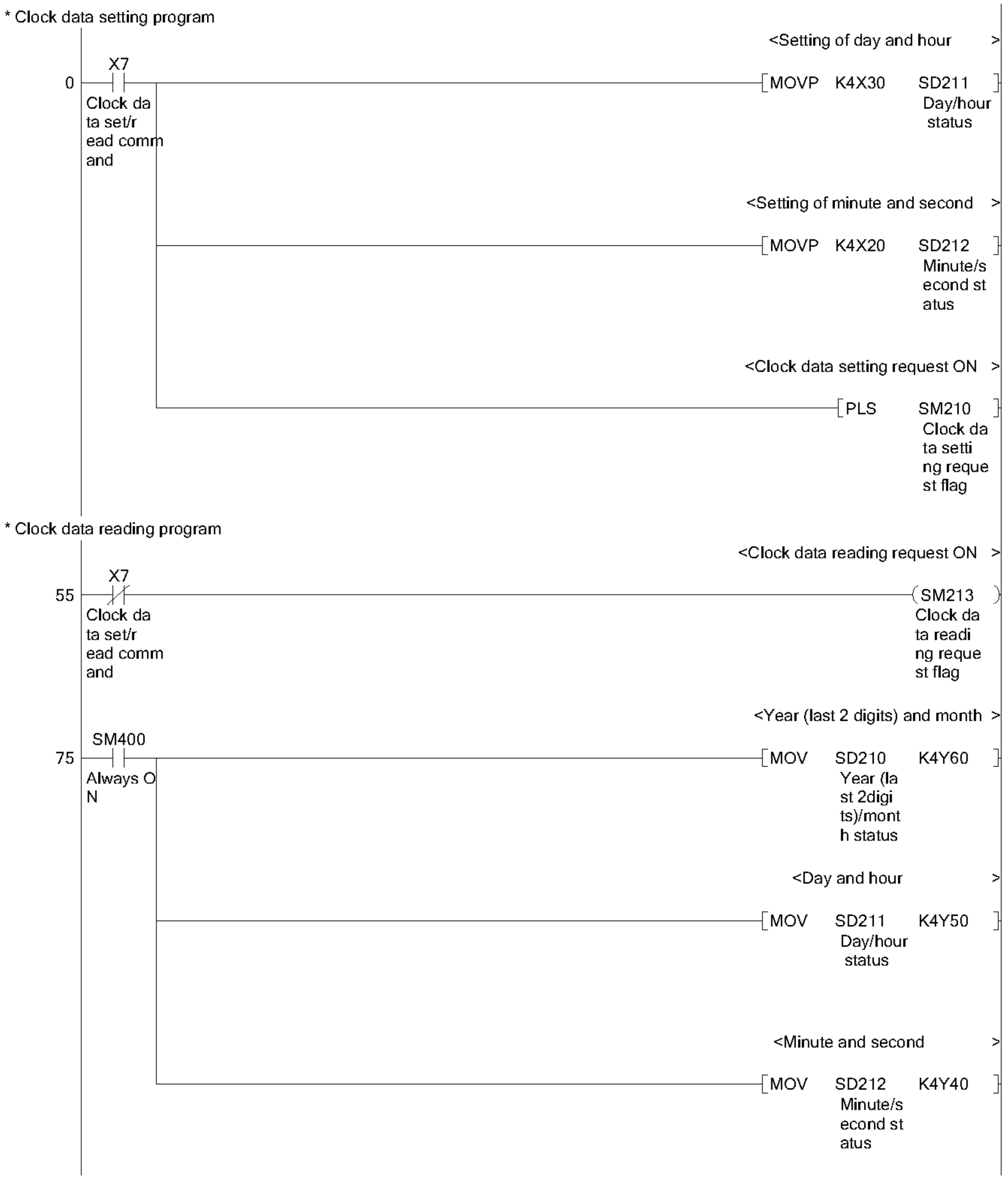


	ON	ON			ON						ON	ON					ON				ON	ON	ON
Y57	Y56	Y55	Y54	Y53	Y52	Y51	Y50	Y4F	Y4E	Y4D	Y4C	Y4B	Y4A	Y49	Y48	Y47	Y46	Y45	Y44	Y43	Y42	Y41	Y40
8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1
Tens digit				Ones digit				Tens digit				Ones digit				Tens digit				Ones digit			
K2Y50 hour, Output								K2Y48 minute, Output								K2Y40 second, Output							

Appendix 4.10.1 Clock function (supplement)

The following ladder displays the time setting set in GX Works2 to the Q demonstration machine.

Project name	QEX13



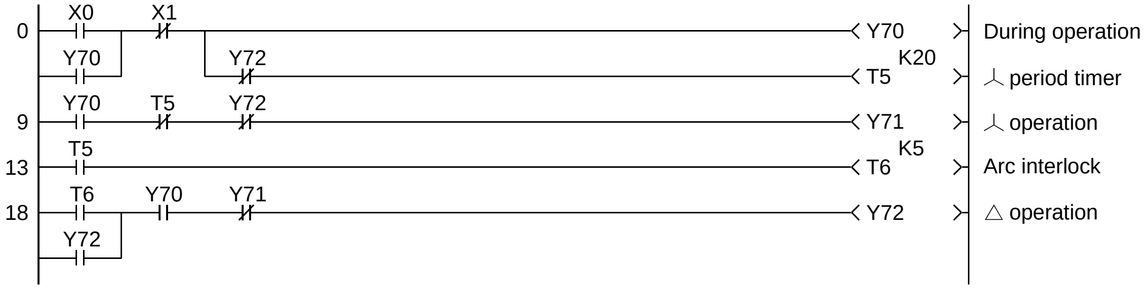
```
* Clock data setting program
                                              <Setting of day and hour         >
    X7
0 --| |--+-----------------------------------[MOVP  K4X30   SD211    ]
  Clock da                                                  Day/hour
  ta set/r                                                   status
  ead comm
  and
         |                                    <Setting of minute and second    >
         +-----------------------------------[MOVP  K4X20   SD212    ]
         |                                                  Minute/s
         |                                                  econd st
         |                                                  atus
         |                                    <Clock data setting request ON   >
         +-----------------------------------------[PLS     SM210    ]
                                                            Clock da
                                                            ta setti
                                                            ng reque
                                                            st flag
* Clock data reading program
                                              <Clock data reading request ON   >
    X7
55 --|/|--------------------------------------------------(SM213    )
  Clock da                                                 Clock da
  ta set/r                                                 ta readi
  ead comm                                                 ng reque
  and                                                      st flag
                                              <Year (last 2 digits) and month  >
   SM400
75 --| |--+-----------------------------------[MOV   SD210   K4Y60    ]
  Always O                                          Year (la
  N      |                                          st 2digi
         |                                          ts)/mont
         |                                          h status
         |                                    <Day and hour                    >
         +-----------------------------------[MOV   SD211   K4Y50    ]
         |                                          Day/hour
         |                                           status
         |                                    <Minute and second               >
         +-----------------------------------[MOV   SD212   K4Y40    ]
                                                    Minute/s
                                                    econd st
                                                    atus
```

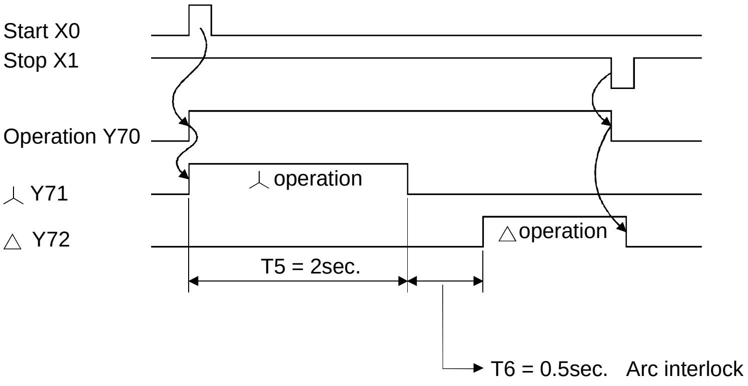
```
                                                                  <Display of Sunday        >
118 [=      H2000   SD213   ]---------------------------------------------(Y76        )
                     Year (fi
                     rst 2dig
                     its)/day
                      of week
                                                                  <Display of Monday        >
133 [=      H2001   SD213   ]---------------------------------------------(Y75        )
                     Year (fi
                     rst 2dig
                     its)/day
                      of week
                                                                  <Display of Tuesday       >
148 [=      H2002   SD213   ]---------------------------------------------(Y74        )
                     Year (fi
                     rst 2dig
                     its)/day
                      of week
                                                                <Display of Wednesday       >
163 [=      H2003   SD213   ]---------------------------------------------(Y73        )
                     Year (fi
                     rst 2dig
                     its)/day
                      of week
                                                                  <Display of Thursday      >
179 [=      H2004   SD213   ]---------------------------------------------(Y72        )
                     Year (fi
                     rst 2dig
                     its)/day
                      of week
                                                                   <Display of Friday       >
195 [=      H2005   SD213   ]---------------------------------------------(Y71        )
                     Year (fi
                     rst 2dig
                     its)/day
                      of week
                                                                  <Display of Saturday      >
210 [=      H2006   SD213   ]---------------------------------------------(Y70        )
                     Year (fi
                     rst 2dig
                     its)/day
                      of week

226 ----------------------------------------------------------------------[END        ]
```

Appendix 4.11 Starting 人 - △ operation of electrical machinery

Turning on the start switch starts the 人 operation. After the 人 operation time has elapsed, the △ operation mode is activated through an arc interlock state.

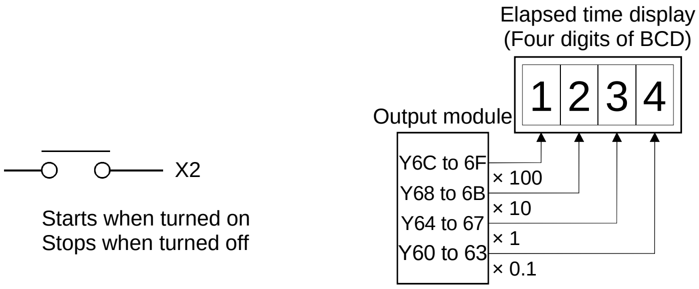
Project name	QA-20
Program name	MAIN

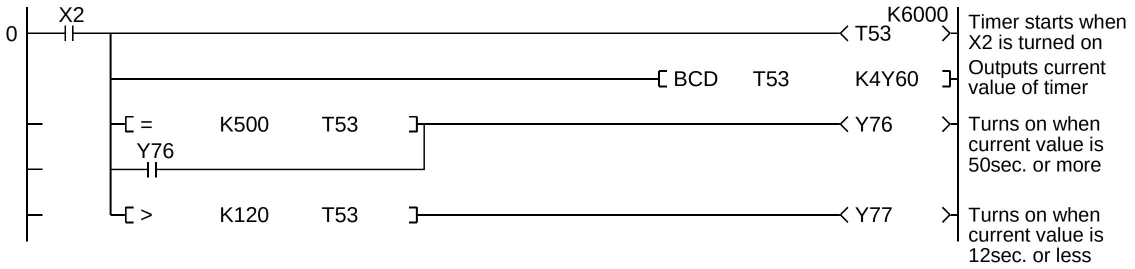




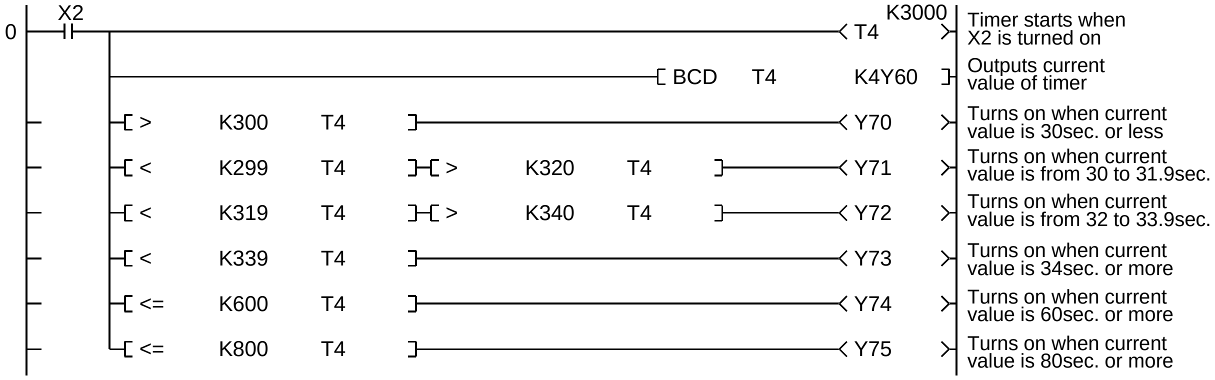
Appendix 4.12   Displaying elapsed time and outputting before time limit

The following ladder outputs the time elapsed in the timer on the LED display, and indicates that the set time limit has been reached. This system can also be applied to counters.





0
X2
K6000
T53
Timer starts when X2 is turned on
BCD T53 K4Y60
Outputs current value of timer
= K500 T53
Y76
Y76
Turns on when current value is 50sec. or more
> K120 T53
Y77
Turns on when current value is 12sec. or less



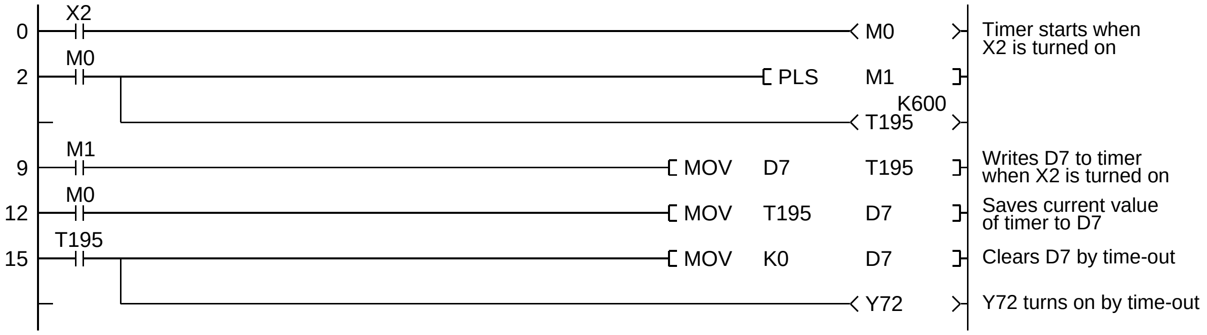
0
X2
K3000
T4
Timer starts when X2 is turned on
BCD T4 K4Y60
Outputs current value of timer
> K300 T4
Y70
Turns on when current value is 30sec. or less
< K299 T4 > K320 T4
Y71
Turns on when current value is from 30 to 31.9sec.
< K319 T4 > K340 T4
Y72
Turns on when current value is from 32 to 33.9sec.
< K339 T4
Y73
Turns on when current value is 34sec. or more
<= K600 T4
Y74
Turns on when current value is 60sec. or more
<= K800 T4
Y75
Turns on when current value is 80sec. or more

Appendix 4.13 Retentive timer

The input X2 switches between on and off continuously. The on-time of X2 is accumulated and Y72 turns on according to this accumulated value n.

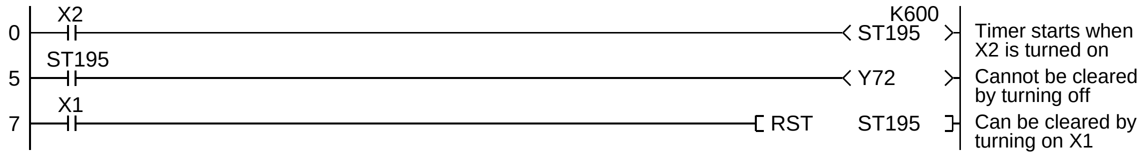
(1) For a ladder that accumulates a value without a retentive timer

Project name	QA-21
Program name	MAIN



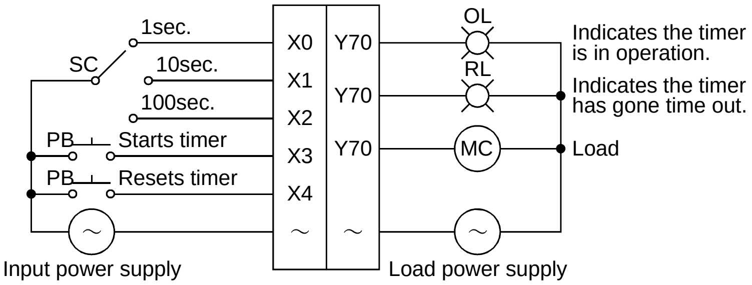
(2) When retentive timers are assigned in the device setting of the PLC parameter Retentive timer (ST): 224 points (ST0 to ST223)

Project name	QA-8
Program name	MAIN

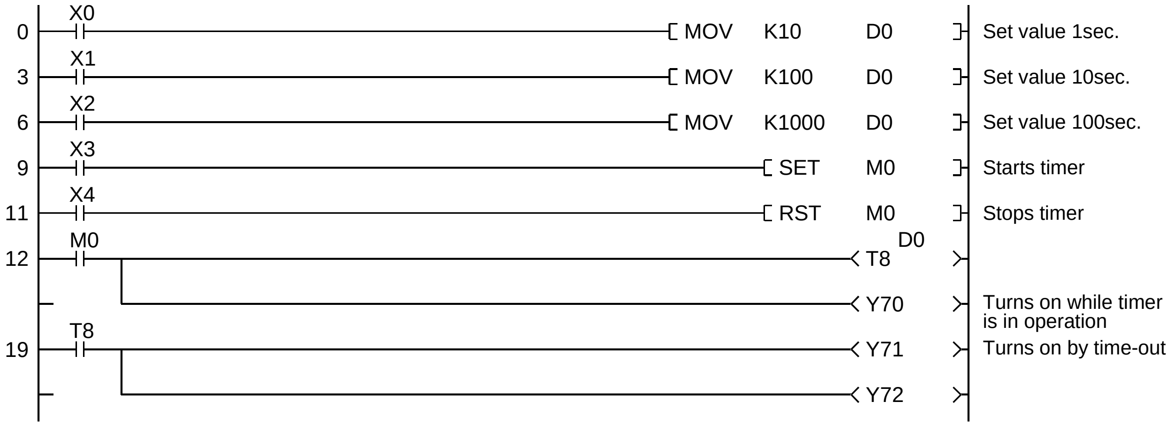

0
X2
K600
ST195
Timer starts when X2 is turned on
5
ST195
Y72
Cannot be cleared by turning off
7
X1
RST
ST195
Can be cleared by turning on X1

Appendix 4.14 Switching timer set value externally

(1) With an external switch, a value to be set in one timer can be selected from three patterns; 1sec., 10sec., and 100sec.
A timer is activated and reset with a push button switch.



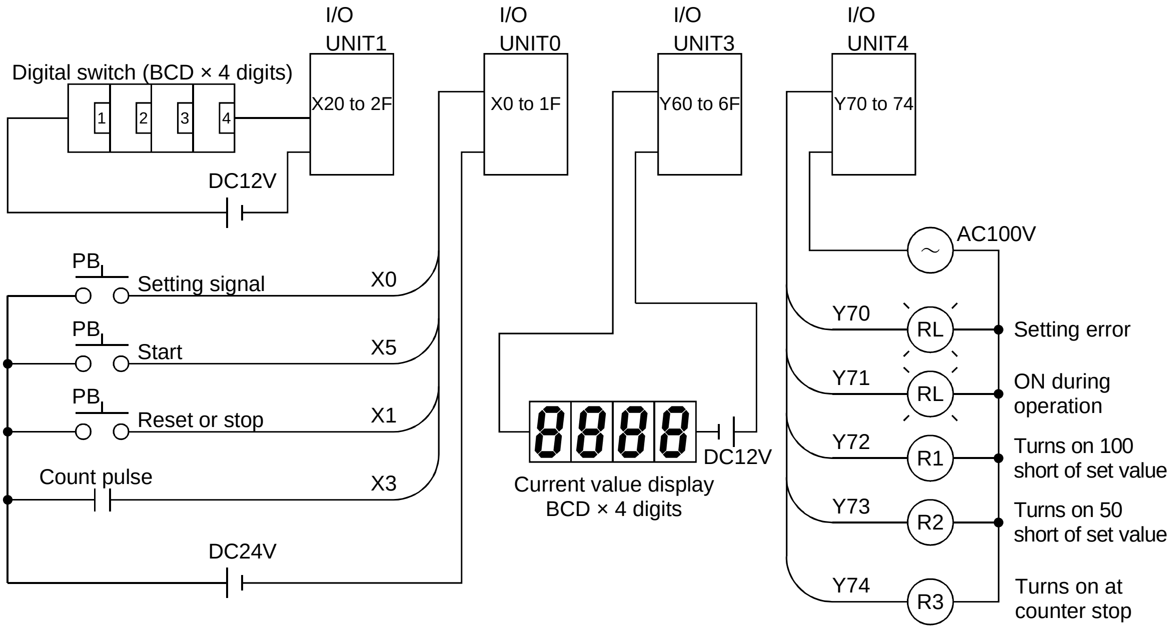
Project name	QA-22
Program name	MAIN

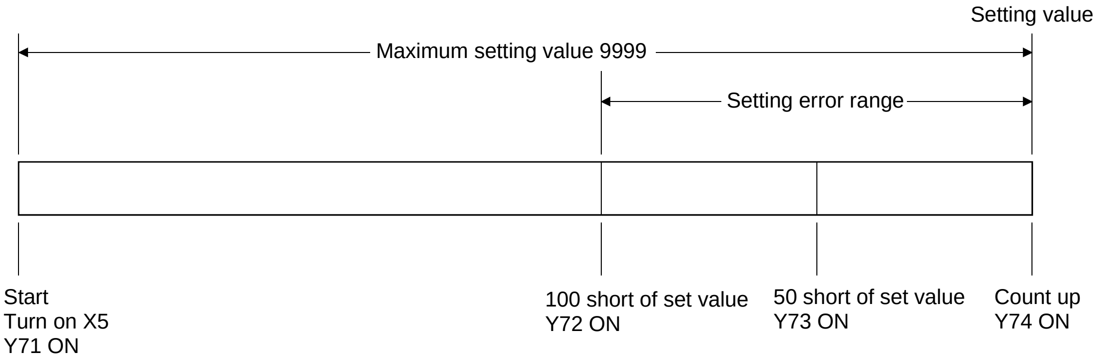


Appendix 4.15 Setting counters externally

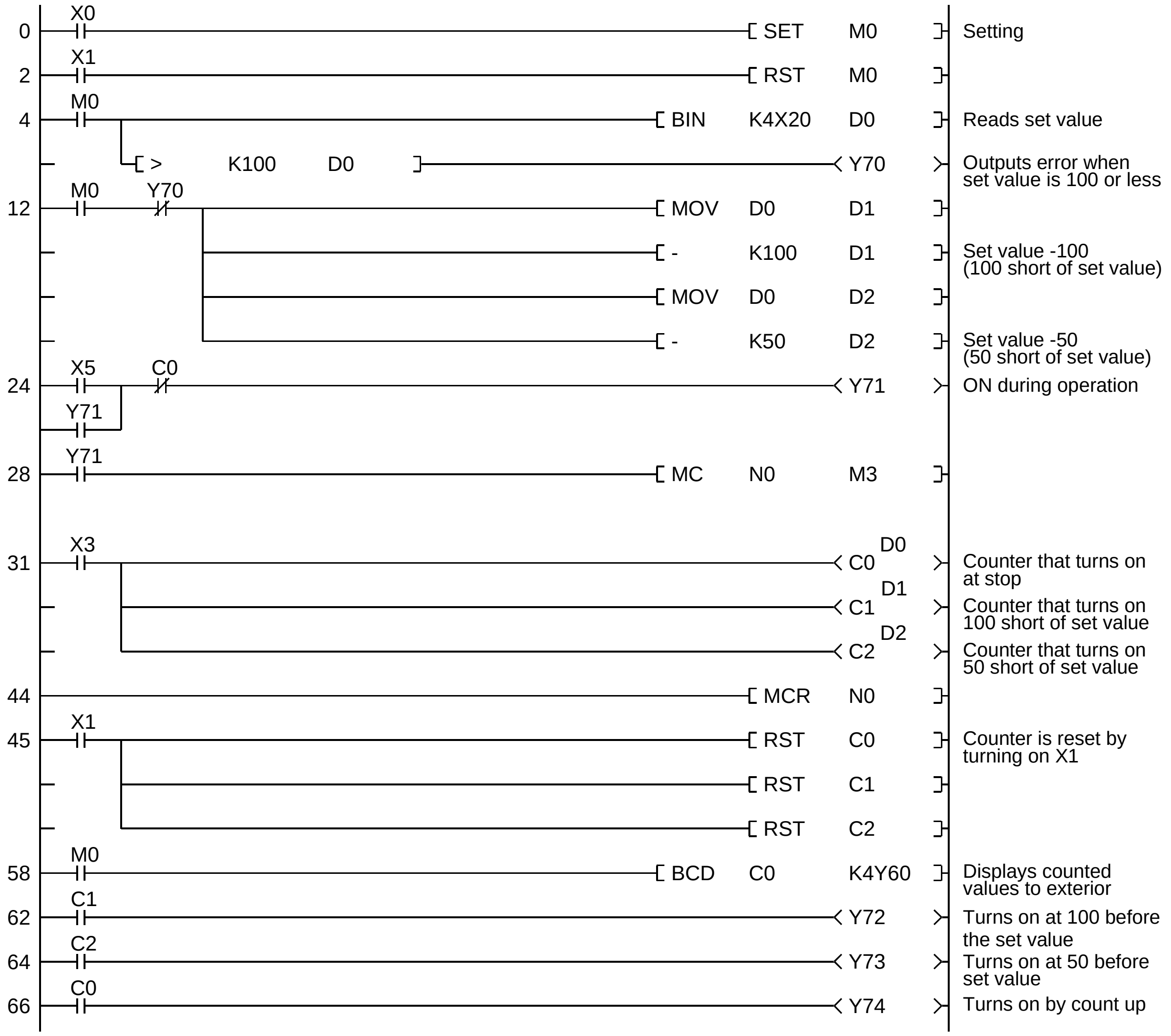
With an external digital switch having 4 digits, a counter can be set remotely and their current values are displayed in 4 digits. In addition to every count-up, the timer outputs data when it reaches a value 100 short of the set value and a value 50 short of the set value.

Note that a setting error is indicated if the set value of the counter is less than 100.





Project name	QA-4
Program name	MAIN

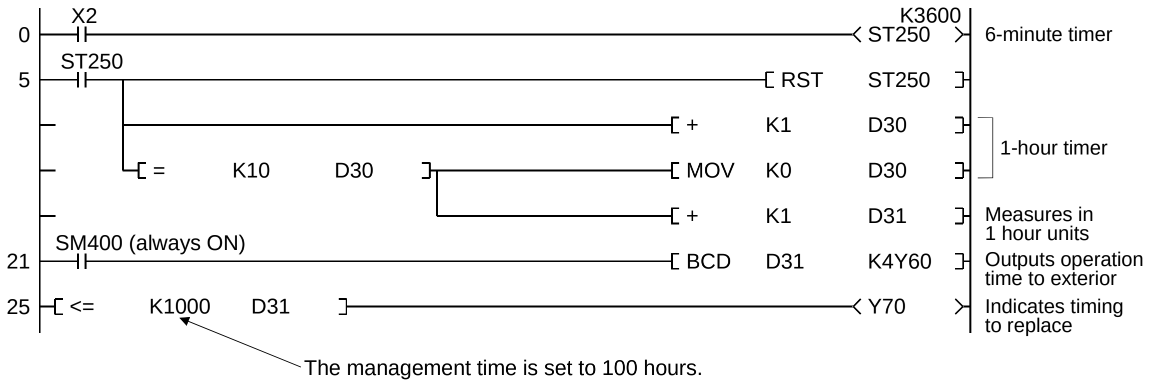

0 X0 [SET M0] Setting
2 X1 [RST M0]
4 M0 [BIN K4X20 D0] Reads set value
[> K100 D0] (Y70) Outputs error when set value is 100 or less
12 M0 Y70 [MOV D0 D1]
[- K100 D1] Set value -100 (100 short of set value)
[MOV D0 D2]
[- K50 D2] Set value -50 (50 short of set value)
24 X5 C0 (Y71) ON during operation
Y71
28 Y71 [MC N0 M3]
31 X3 (C0 D0) Counter that turns on at stop
(C1 D1) Counter that turns on 100 short of set value
(C2 D2) Counter that turns on 50 short of set value
44 [MCR N0]
45 X1 [RST C0] Counter is reset by turning on X1
[RST C1]
[RST C2]
58 M0 [BCD C0 K4Y60] Displays counted values to exterior
62 C1 (Y72) Turns on at 100 before the set value
64 C2 (Y73) Turns on at 50 before set value
66 C0 (Y74) Turns on by count up

Note) In GX Works2, the on/off status of the master control is displayed in the title tag on the monitor screen.

Appendix 4.16 Measuring operation time

Setting an operation time to a control target is useful for judging the timing of a component replacement and lubrication. The timer ST and data register D must have a backup power source so that they can continue operating at a power failure. With the contents of D31 (in one hour units) displayed externally, the program can work as an operation timer.

Project name	QA-23
Program name	MAIN

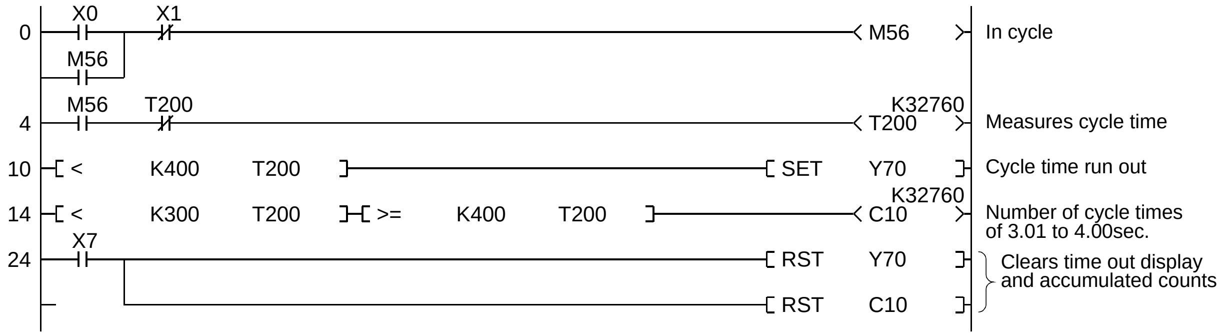

0 X2 K3600 ST250 6-minute timer
5 ST250 RST ST250
+ K1 D30
= K10 D30 MOV K0 D30 1-hour timer
+ K1 D31 Measures in 1 hour units
21 SM400 (always ON) BCD D31 K4Y60 Outputs operation time to exterior
25 <= K1000 D31 Y70 Indicates timing to replace

The management time is set to 100 hours.

Appendix 4.17 Measuring cycle time

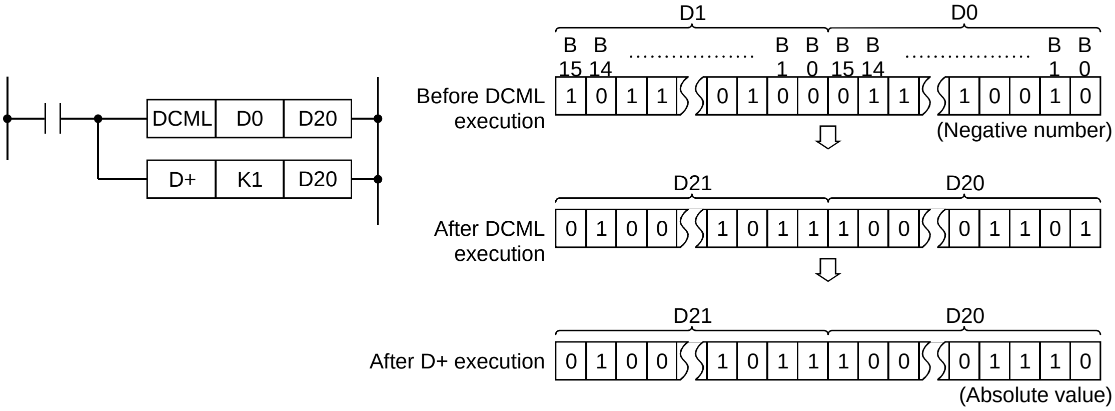
Measuring the operation time of a control target (from its start to end) allows displaying the cycle time-out and managing a control time lag.
The following ladder in which the <, >, and = instructions are used to determine the state of T200 indicates a cycle time-out and measures a time lag with the counter.

Project name	QA-24
Program name	MAIN

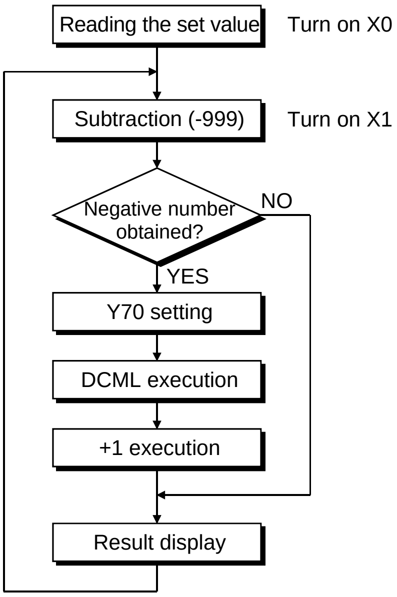

0 X0 X1 M56 In cycle
M56
4 M56 T200 K32760 T200 Measures cycle time
10 < K400 T200 SET Y70 Cycle time run out
14 < K300 T200 >= K400 T200 K32760 C10 Number of cycle times of 3.01 to 4.00sec.
24 X7 RST Y70
RST C10 Clears time out display and accumulated counts

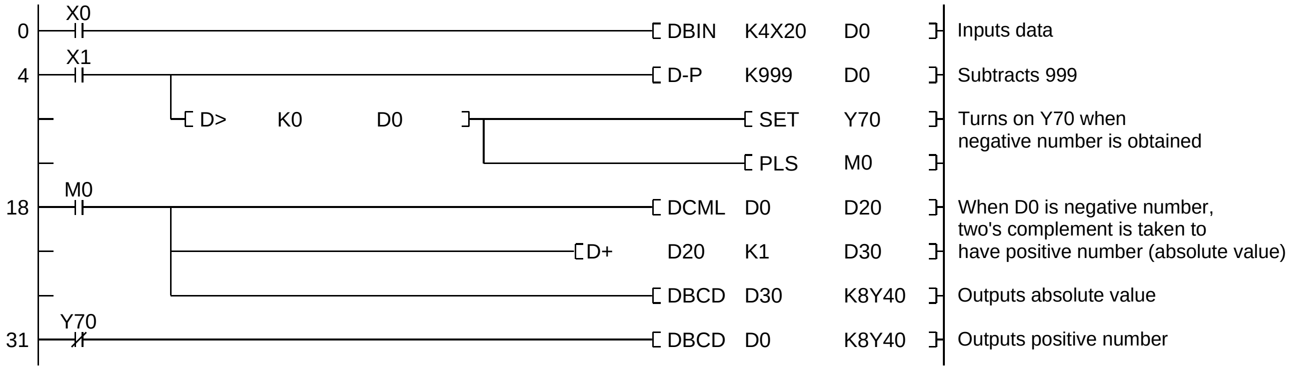
Appendix 4.18 Application example of (D) $\overset{\text{Complement}}{\text{C M L}}$ (P)

The following explains how to obtain absolute values of negative values -32768 or smaller (to -2147483648, 32 bit data).



(Example)
Every time X1 is turned on, 999 is subtracted from a set value and the result is displayed.
When the result value is negative, the output Y70 turns on, and the absolute value of the result is displayed.

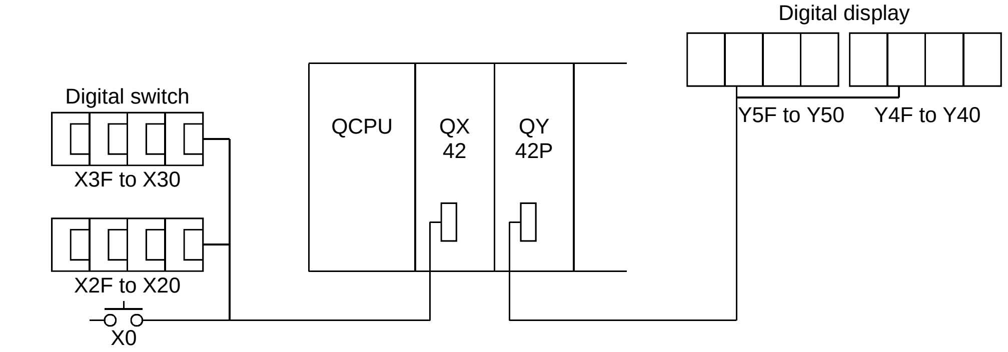


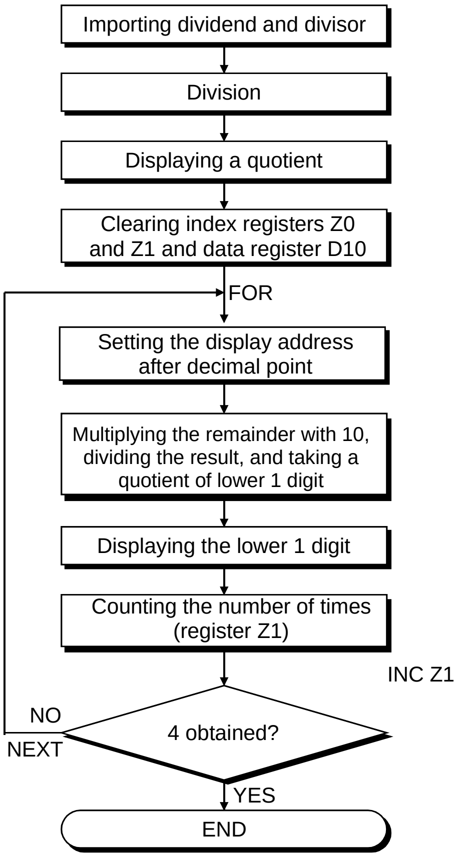


Appendix 4.19 Program showing divided value of 4-digit BIN value to 4 places of decimals

(1) Example 1

The program displays the operation result using a dividend and a divisor which are individually specified in two 4-digit digital switches on two 4-digit displays (integral part and decimal part).



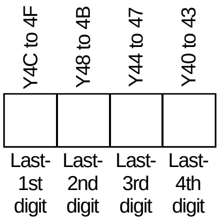

Importing dividend and divisor
Division
Displaying a quotient
Clearing index registers Z0 and Z1 and data register D10
FOR
Setting the display address after decimal point
Multiplying the remainder with 10, dividing the result, and taking a quotient of lower 1 digit
Displaying the lower 1 digit
Counting the number of times (register Z1)
INC Z1
NO
NEXT
4 obtained?
YES
END

Dividend Digital switch X30 to X3F → D0
Divisor Digital switch X20 to X2F → D1

(D0) / (D1) = (D2) ...... (D3)
Quotient Remainder

4 × (Z1) → (D10)	HC-(D10) → (Z0)
1st time 4 × 0 → 0	HC-K0 → HC
2nd time 4 × 1 → 4	HC-K4 → H8
3rd time 4 × 2 → 8	HC-K8 → H4
4th time 4 × 3 → 12	HC-K12 → H0

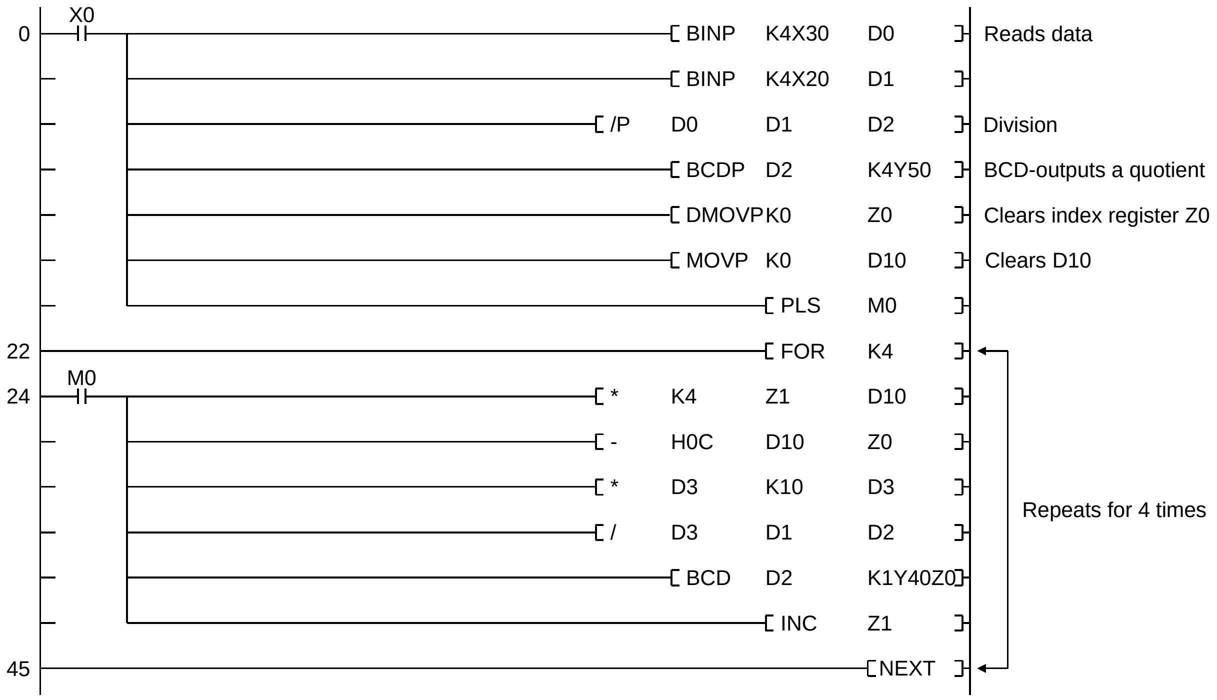
(D3) × 10 → (D3)
(D3) / (D1) = (D2) ...... (D3)



Sequence program of example 1

The FOR-NEXT instruction is executed to divide each decimal place individually and 4 decimal places are displayed in K4Y40.

Project name	QA-5
Program name	MAIN



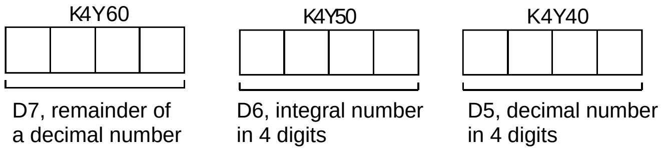
```
    X0
 0 -| |-+-----------------[ BINP   K4X30  D0     ]  Reads data
        +-----------------[ BINP   K4X20  D1     ]
        +-----------------[ /P     D0     D1   D2 ]  Division
        +-----------------[ BCDP   D2     K4Y50  ]  BCD-outputs a quotient
        +-----------------[ DMOVP  K0     Z0     ]  Clears index register Z0
        +-----------------[ MOVP   K0     D10    ]  Clears D10
        +-----------------[ PLS    M0            ]
22 ------------------------[ FOR    K4            ] <-+
    M0                                                 |
24 -| |-+-----------------[ *   K4    Z1    D10   ]   |
        +-----------------[ -   H0C   D10   Z0    ]   |
        +-----------------[ *   D3    K10   D3    ]   |  Repeats for 4 times
        +-----------------[ /   D3    D1    D2    ]   |
        +-----------------[ BCD    D2     K1Y40Z0 ]   |
        +-----------------[ INC    Z1            ]   |
45 ------------------------[NEXT                 ] <-+
```

Executing the INC Z1 instruction adds one to Z1.

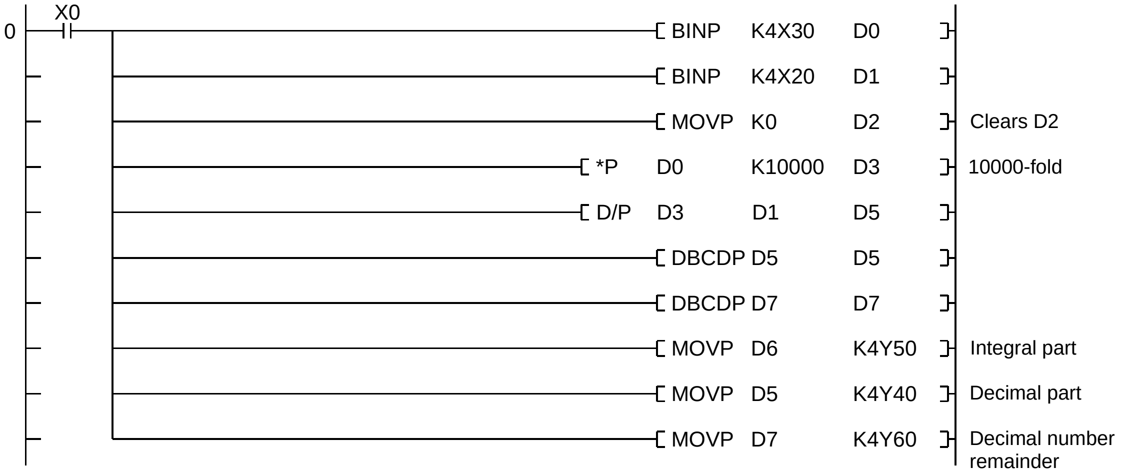
(2) Example 2

In example 2, D0 is divided by D1 to obtain D5 in 4 decimal places.

The dividend D0 is multiplied with 10000. The result of the dividing calculation using this multiplied value is converted to a BCD value and output to an external digital display.


K4Y60
K4Y50
K4Y40
D7, remainder of a decimal number
D6, integral number in 4 digits
D5, decimal number in 4 digits

Project name	QA-6
Program name	MAIN

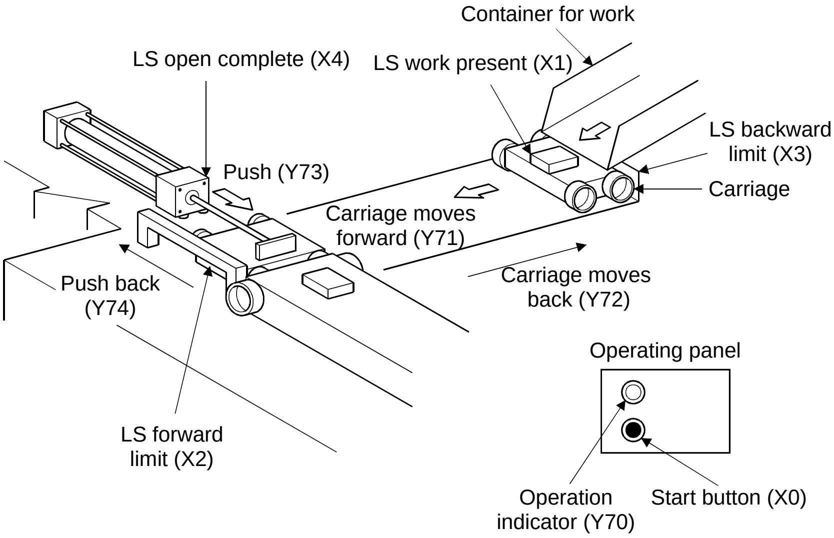


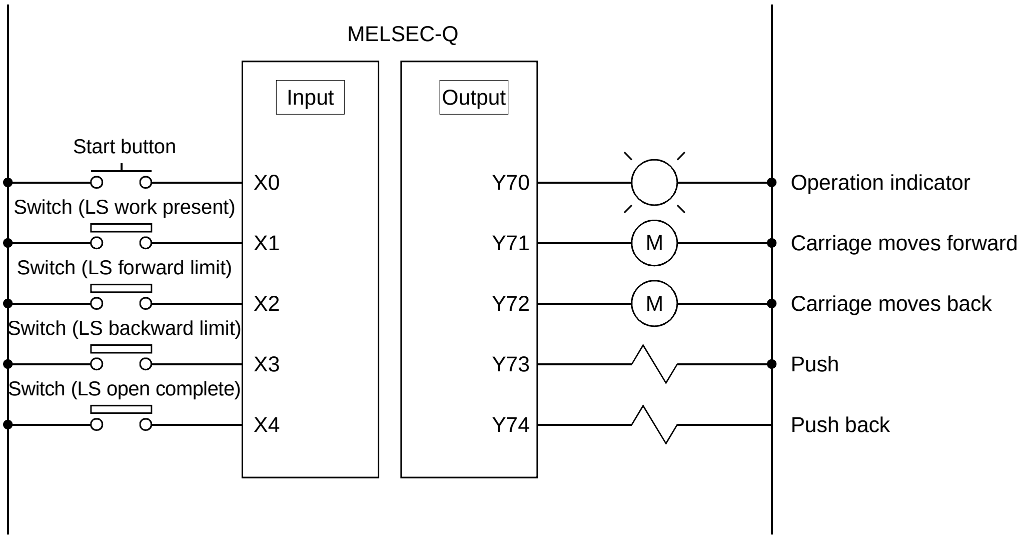
```
0  X0
---| |---+--------------------[ BINP   K4X30   D0     ]
         +--------------------[ BINP   K4X20   D1     ]
         +--------------------[ MOVP   K0      D2     ]  Clears D2
         +---------------[ *P     D0      K10000  D3     ]  10000-fold
         +---------------[ D/P    D3      D1      D5     ]
         +--------------------[ DBCDP  D5      D5     ]
         +--------------------[ DBCDP  D7      D7     ]
         +--------------------[ MOVP   D6      K4Y50  ]  Integral part
         +--------------------[ MOVP   D5      K4Y40  ]  Decimal part
         +--------------------[ MOVP   D7      K4Y60  ]  Decimal number
                                                          remainder
```

Appendix 4.20 Carriage line control

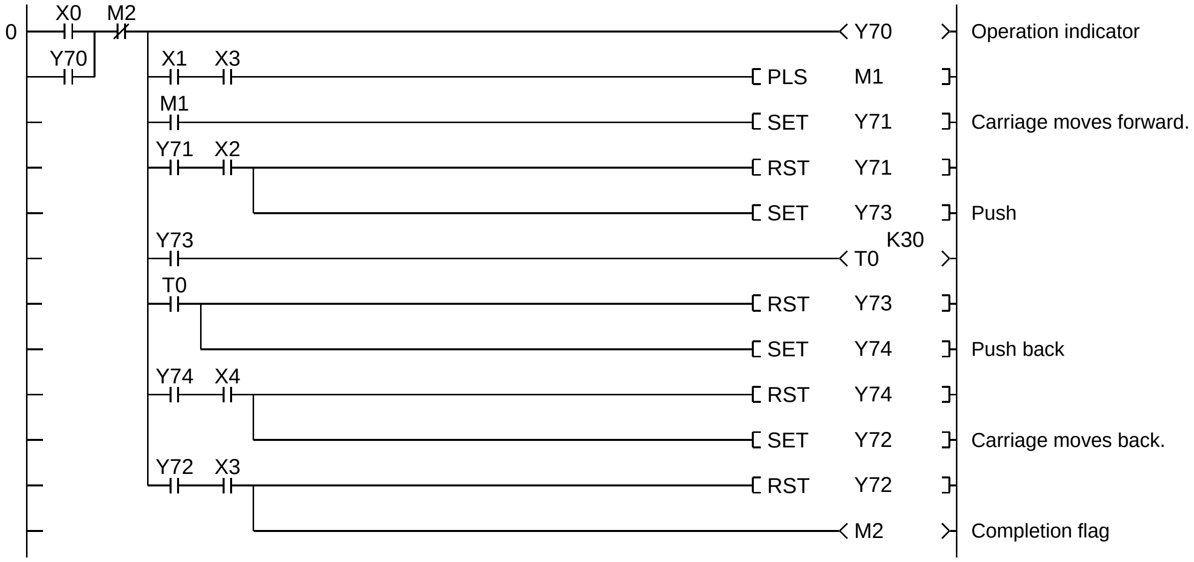
The following is an example of a sequence control using a carriage to convey works (materials).

Series of operations performed in one cycle is as follows; A work is set on the carriage, the carriage moves forward, the carriage stops at the forward limit, the arm pushes the work to the other conveyor side, and the carriage moves back to the backward limit.

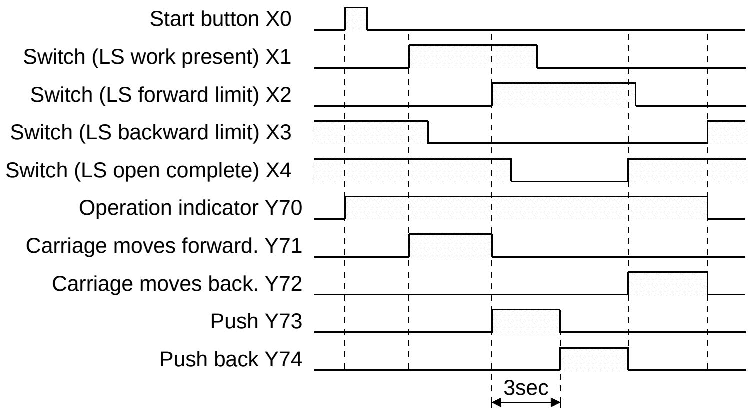




Project name	QA-10
Program name	MAIN

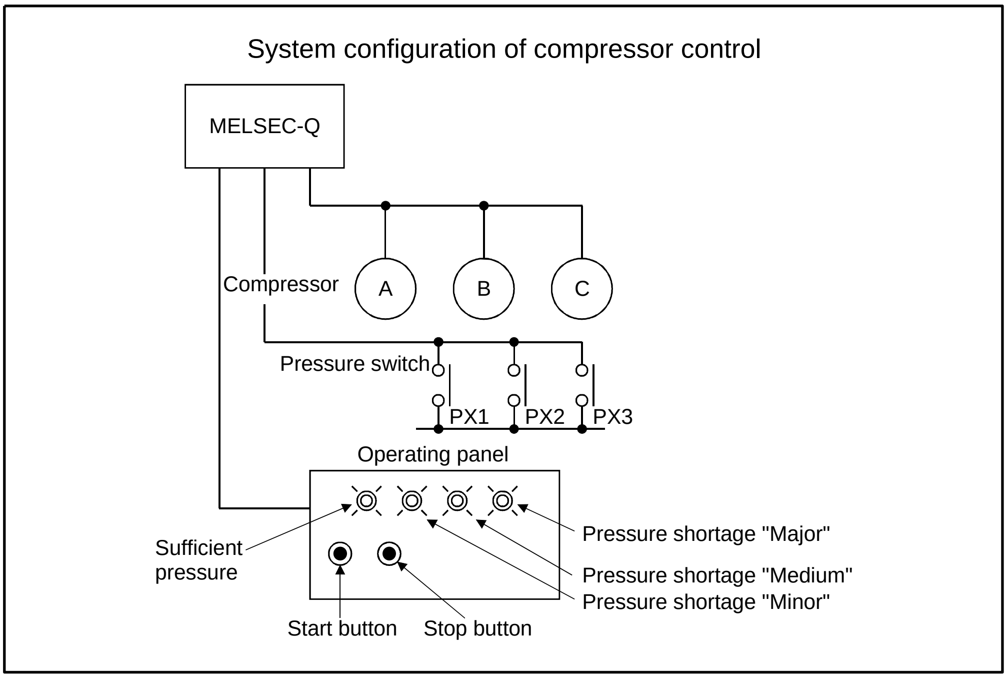


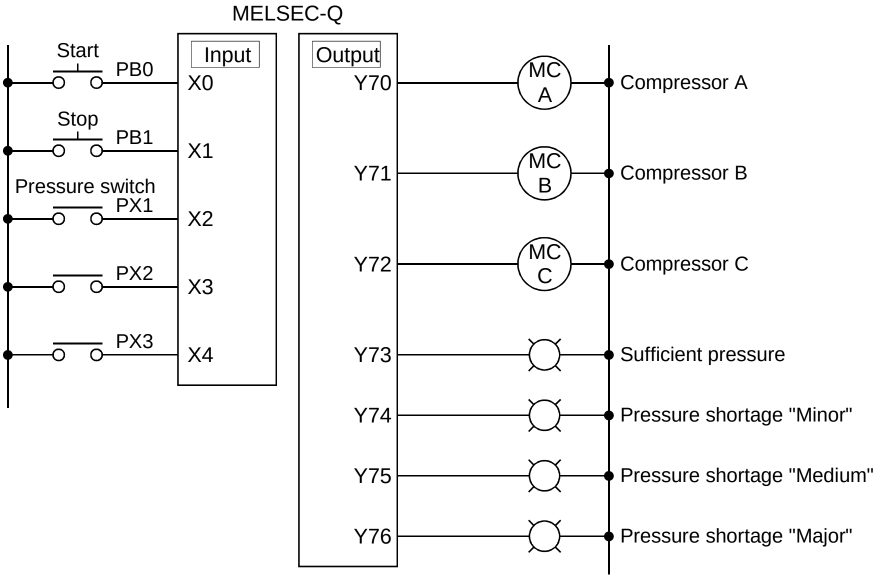
Timing chart



Appendix 4.21 Compressor sequential operation using ring counters

This system provides pressure control using three compressors.
A pressure shortage is detected by the three pressure switches. The number of compressors operating simultaneously depends on the degree of shortage. To equal the number of usages of each compressor, compressors are activated according to the set order.


System configuration of compressor control
MELSEC-Q
Compressor
A
B
C
Pressure switch
PX1
PX2
PX3
Operating panel
Sufficient pressure
Pressure shortage "Major"
Pressure shortage "Medium"
Pressure shortage "Minor"
Start button
Stop button



Operation explanation

(1) The pressure switches (X2, X3, and X4) are initially off. In this state, turning on the start switch (X0) activates the three compressors all together, and when sufficient pressure is obtained (X2, X3, and X4 turn on), the three compressors stop. This is the basic operation of this system.

If all the compressors are at stop with sufficient pressure or the pressure shortage "Minor" is detected (X4 turns off), one compressor is activated to supply pressure until sufficient pressure is obtained.

The compressor activated at this time activates in order from A to C each time compressors are reactivated in reaction to pressure shortage.

Note that the stop switch (X1) is available for stopping compressors at any time.

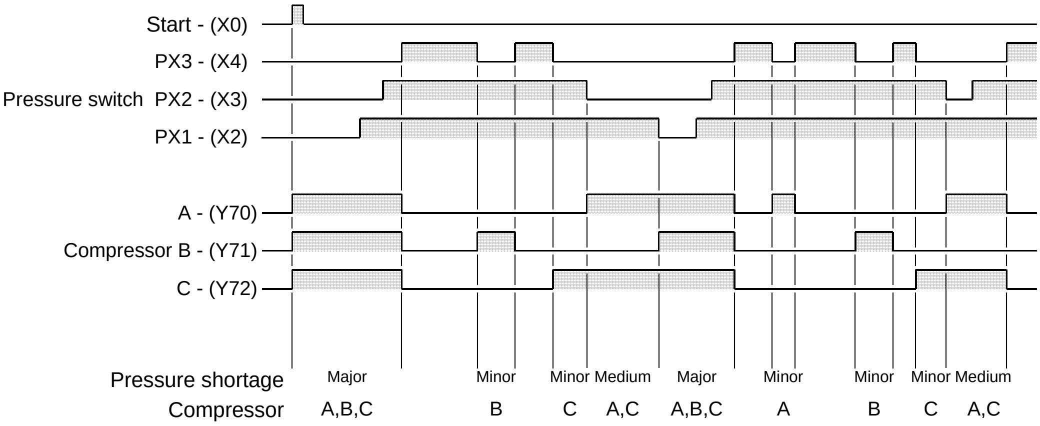
(2) If one compressor does not supply sufficient pressure and the pressure shortage level goes up to "Medium" (X3 turns off), the second compressor is activated to support the first compressor. This second compressor will be compressor C if compressor A has been in operation, A if B has been in operation, and B if C has been in operation.

(3) If two compressors do not supply sufficient pressure and pressure shortage level goes up to "Major" (X2 turns off), the last compressor is also activated.

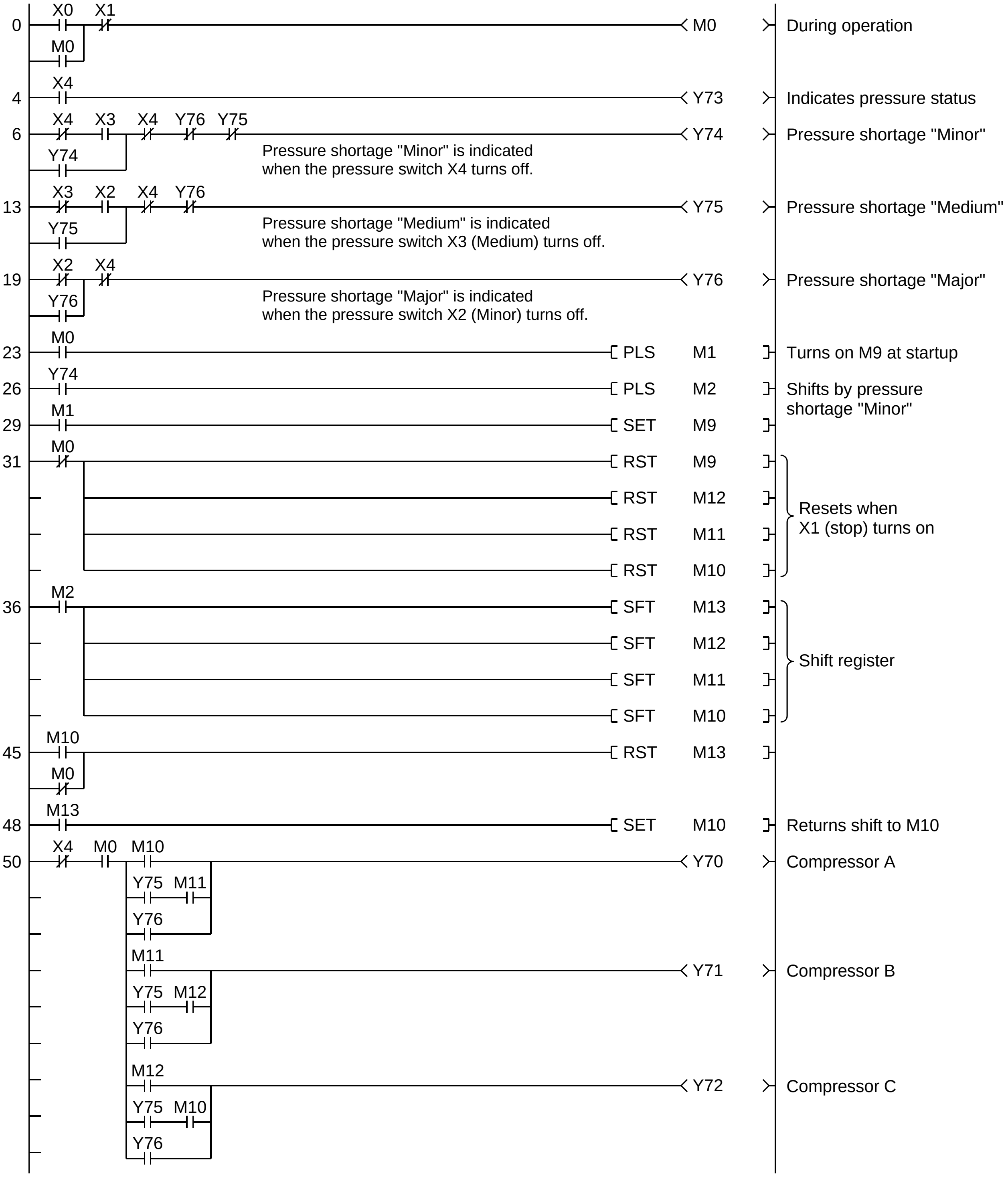
When only one compressor is in operation and pressure shortage level goes from "Minor" to "Major" directly, the rest two compressors are activated simultaneously.

(4) When two or three compressors are in operation, they continue operating together until sufficient pressure is obtained. Then they stop together when sufficient pressure is obtained (X4 turns on).

Timing chart

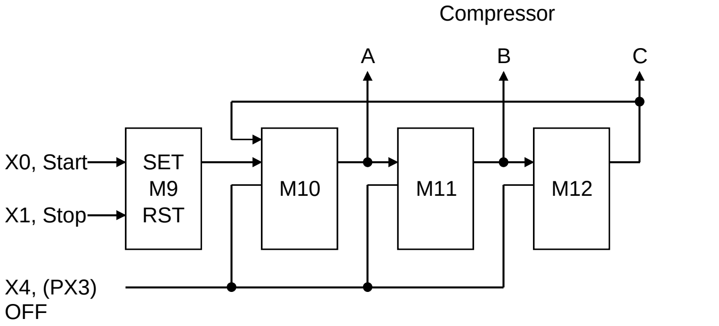


Project name	QA-11
Program name	MAIN

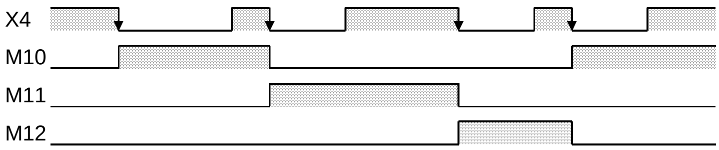

0
X0
X1
M0
M0
During operation
4
X4
Y73
Indicates pressure status
6
X4
X3
X4
Y76
Y75
Y74
Y74
Pressure shortage "Minor"
Pressure shortage "Minor" is indicated
when the pressure switch X4 turns off.
13
X3
X2
X4
Y76
Y75
Y75
Pressure shortage "Medium"
Pressure shortage "Medium" is indicated
when the pressure switch X3 (Medium) turns off.
19
X2
X4
Y76
Y76
Pressure shortage "Major"
Pressure shortage "Major" is indicated
when the pressure switch X2 (Minor) turns off.
23
M0
PLS M1
Turns on M9 at startup
26
Y74
PLS M2
Shifts by pressure shortage "Minor"
29
M1
SET M9
31
M0
RST M9
RST M12
RST M11
RST M10
Resets when X1 (stop) turns on
36
M2
SFT M13
SFT M12
SFT M11
SFT M10
Shift register
45
M10
M0
RST M13
48
M13
SET M10
Returns shift to M10
50
X4
M0
M10
Y75
M11
Y76
Y70
Compressor A
M11
Y75
M12
Y76
Y71
Compressor B
M12
Y75
M10
Y76
Y72
Compressor C

After the basic operation, one compressor is activated in reaction to pressure shortage detected. To use the three compressors equally, they are activated according to the set order. This control is enabled by the 3-stage ring counter (ring-shaped shift registers) M10 to M12.

A shift signal is generated when pressure shortage is detected (X04 switches from on to off).



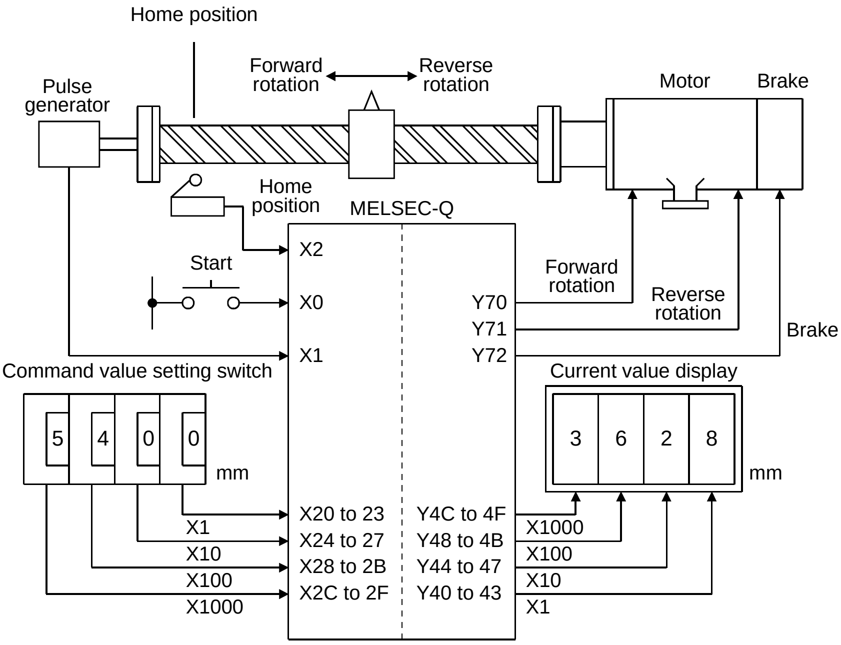
Shift operation



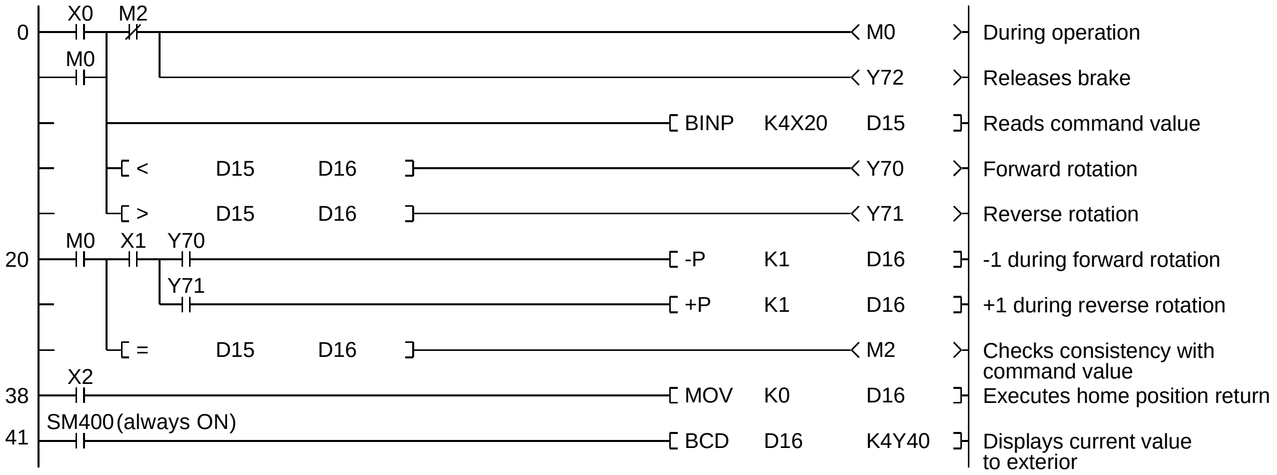
Appendix 4.22 Application example of positioning control

The following is an example of a positioning system with a pulse generator that outputs pulses per motor, brake, and unit of distance.

In this system, a command value is set with the digital switch, and this set command value is compared with the current value at start-up to determine in which direction, forward or reverse, the motor rotates. The current value in the register D16 is subtracted by 1 in forward direction, and incremented by 1 in reverse direction. Positioning is completed when the command value matches the current value. The current value is converted to a BCD value so that current position is represented in 4-digit decimal numbers.

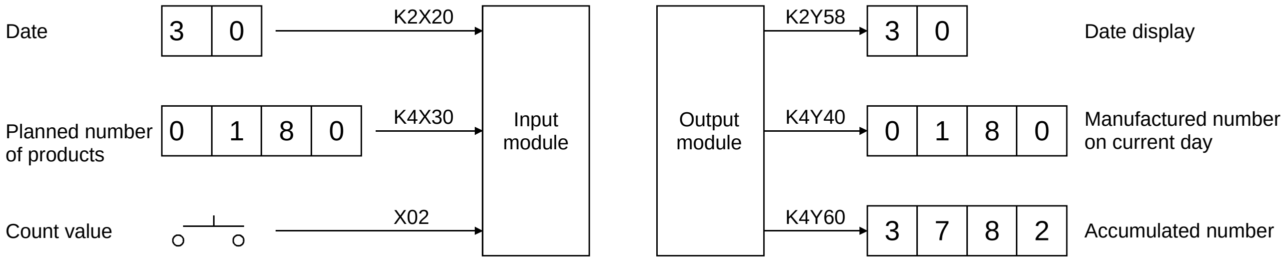


Project name	QA-26
Program name	MAIN



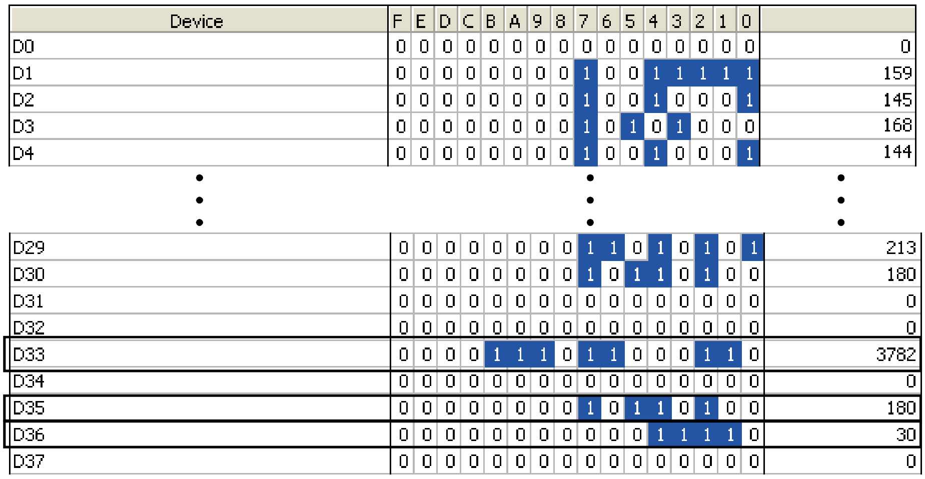
Appendix 4.23 Application example using index Z

(1) The number of manufactured products is counted every day in one month cycle, and the resulting number is stored to the corresponding register of the date (D1 to D31).
(2) The planned number of products to be manufactured is inputted with the external digital switch. Production stops when this number is accomplished.
(3) The date is also specified with the external digital switch.
(4) The accumulated number of products manufactured in the current month as well as the number of manufactured products on the current day is displayed to exterior.



The number of products manufactured on the current day is counted by C5.
The accumulated number of products manufactured is counted by C6.
The date is entered in the index Z to indirectly specify the data register corresponding to the date using D0Z0.
When Z0 is 30, D0Z0 becomes 0 + 30, specifying D30.

[Device/Buffer Memory Batch Monitor screen]



Device	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	
D0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D1	0	0	0	0	0	0	0	0	1	0	0	1	1	1	1	1	159
D2	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	145
D3	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	168
D4	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	144
⋮									⋮								⋮
D29	0	0	0	0	0	0	0	0	1	1	0	1	0	1	0	1	213
D30	0	0	0	0	0	0	0	0	1	0	1	1	0	1	0	0	180
D31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D33	0	0	0	0	1	1	1	0	1	1	0	0	0	1	1	0	3782
D34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D35	0	0	0	0	0	0	0	0	1	0	1	1	0	1	0	0	180
D36	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	30
D37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Stores the resulting number of each day ranging from 1 to 31 in D1 to D31.

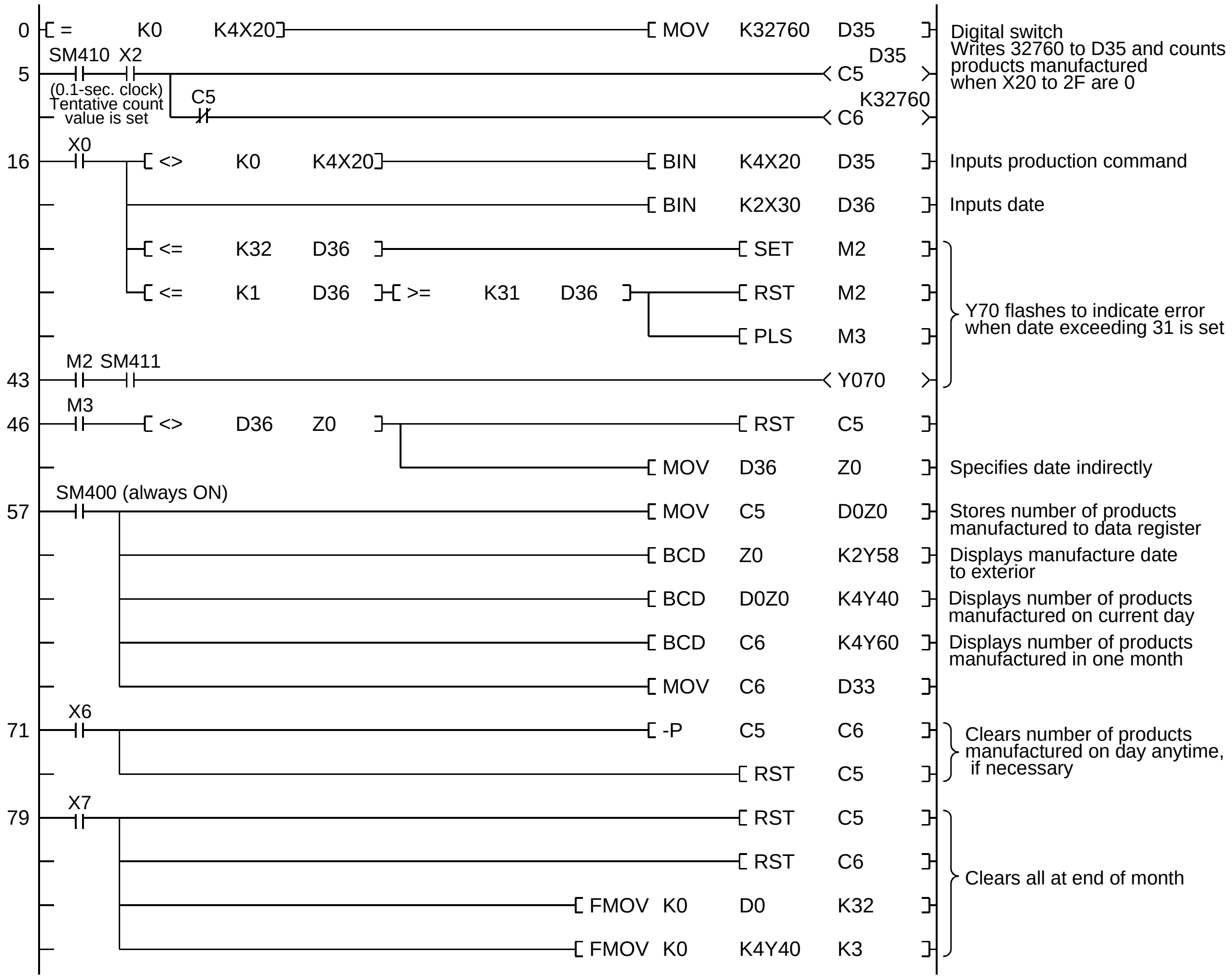
Accumulated number

Planed number of products

Date

Manufacturing results of each day ranging from 1 to 31 are stored in D1 to D31, which are available as production data.

Project name	QA-7
Program name	MAIN



0
[= K0 K4X20]
[MOV K32760 D35]
Digital switch
Writes 32760 to D35 and counts products manufactured when X20 to 2F are 0
5
SM410
X2
(0.1-sec. clock)
Tentative count value is set
D35
(C5)
C5
K32760
(C6)
16
X0
[<> K0 K4X20]
[BIN K4X20 D35]
Inputs production command
[BIN K2X30 D36]
Inputs date
[<= K32 D36]
[SET M2]
[<= K1 D36] [>= K31 D36]
[RST M2]
[PLS M3]
Y70 flashes to indicate error when date exceeding 31 is set
43
M2
SM411
(Y070)
46
M3
[<> D36 Z0]
[RST C5]
[MOV D36 Z0]
Specifies date indirectly
57
SM400 (always ON)
[MOV C5 D0Z0]
Stores number of products manufactured to data register
[BCD Z0 K2Y58]
Displays manufacture date to exterior
[BCD D0Z0 K4Y40]
Displays number of products manufactured on current day
[BCD C6 K4Y60]
Displays number of products manufactured in one month
[MOV C6 D33]
71
X6
[-P C5 C6]
[RST C5]
Clears number of products manufactured on day anytime, if necessary
79
X7
[RST C5]
[RST C6]
[FMOV K0 D0 K32]
[FMOV K0 K4Y40 K3]
Clears all at end of month

FMOV	K0	D0	K32

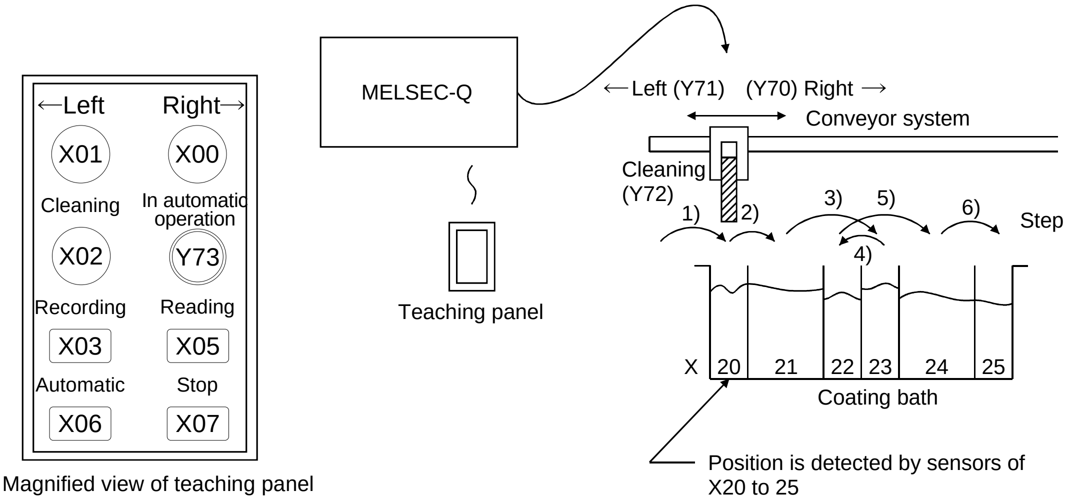
Simultaneously transfers data 0 to D0 to D31.

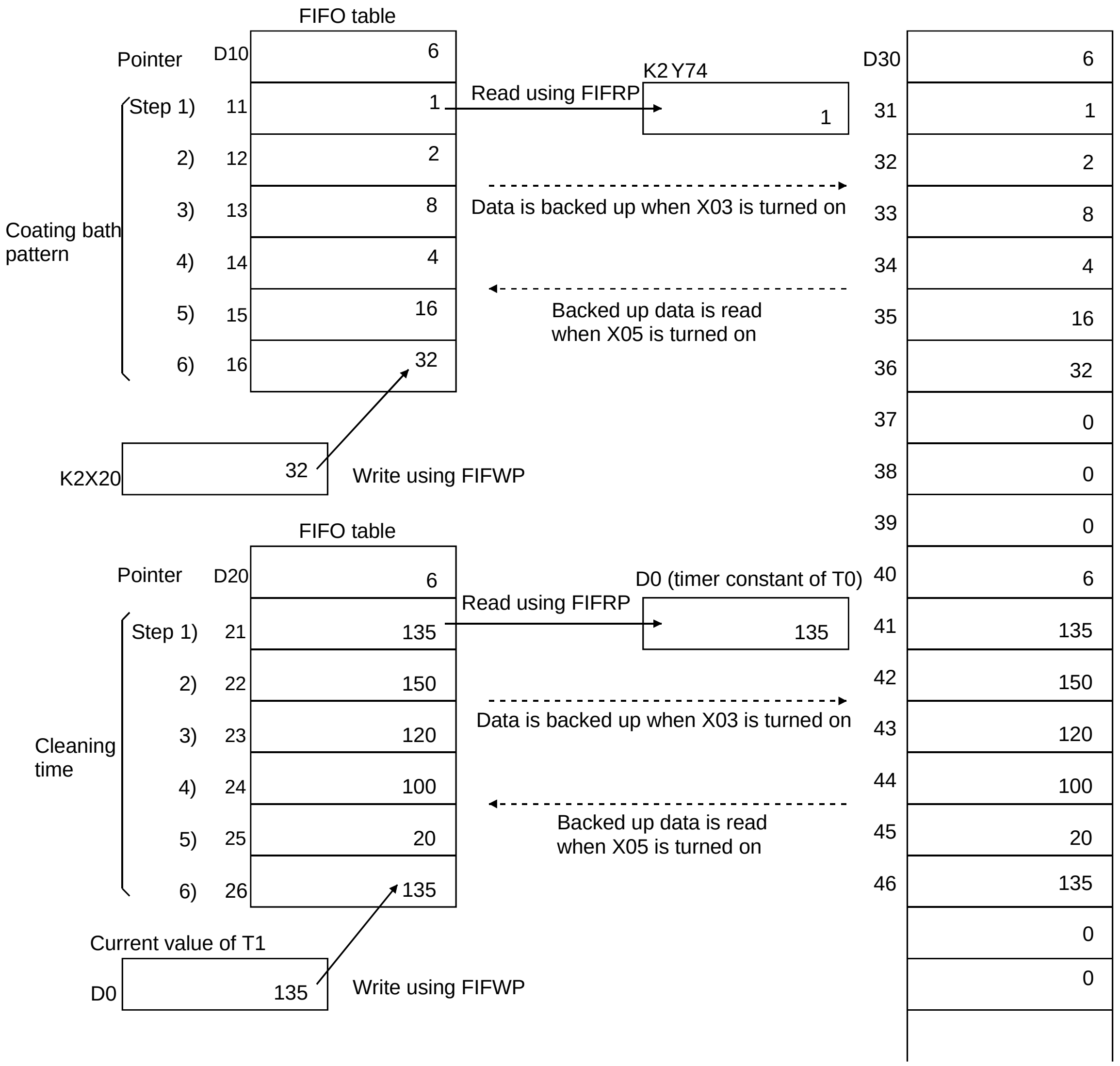
FMOV	K0	K4Y40	K3

Simultaneously transfers data 0 to K4Y40, K4Y50, and K4Y60.

Appendix 4.24 Application example of FIFO instruction

Manual coating work and its working time can be stored and duplicated by machinery later.




Read using FIFRP
K2 Y74
1
Data is backed up when X03 is turned on
Backed up data is read when X05 is turned on
K2X20
32
Write using FIFWP
Read using FIFRP
D0 (timer constant of T0)
135
Data is backed up when X03 is turned on
Backed up data is read when X05 is turned on
Current value of T1
D0
135
Write using FIFWP

FIFO table

Pointer		D10	6
Coating bath pattern	Step 1)	11	1
	2)	12	2
	3)	13	8
	4)	14	4
	5)	15	16
	6)	16	32

FIFO table

Pointer		D20	6
Cleaning time	Step 1)	21	135
	2)	22	150
	3)	23	120
	4)	24	100
	5)	25	20
	6)	26	135

D30	6
31	1
32	2
33	8
34	4
35	16
36	32
37	0
38	0
39	0
40	6
41	135
42	150
43	120
44	100
45	20
46	135
	0
	0

Operation pattern from manual to automatic operation

Teaching panel	Cleaning machine	Coating bath

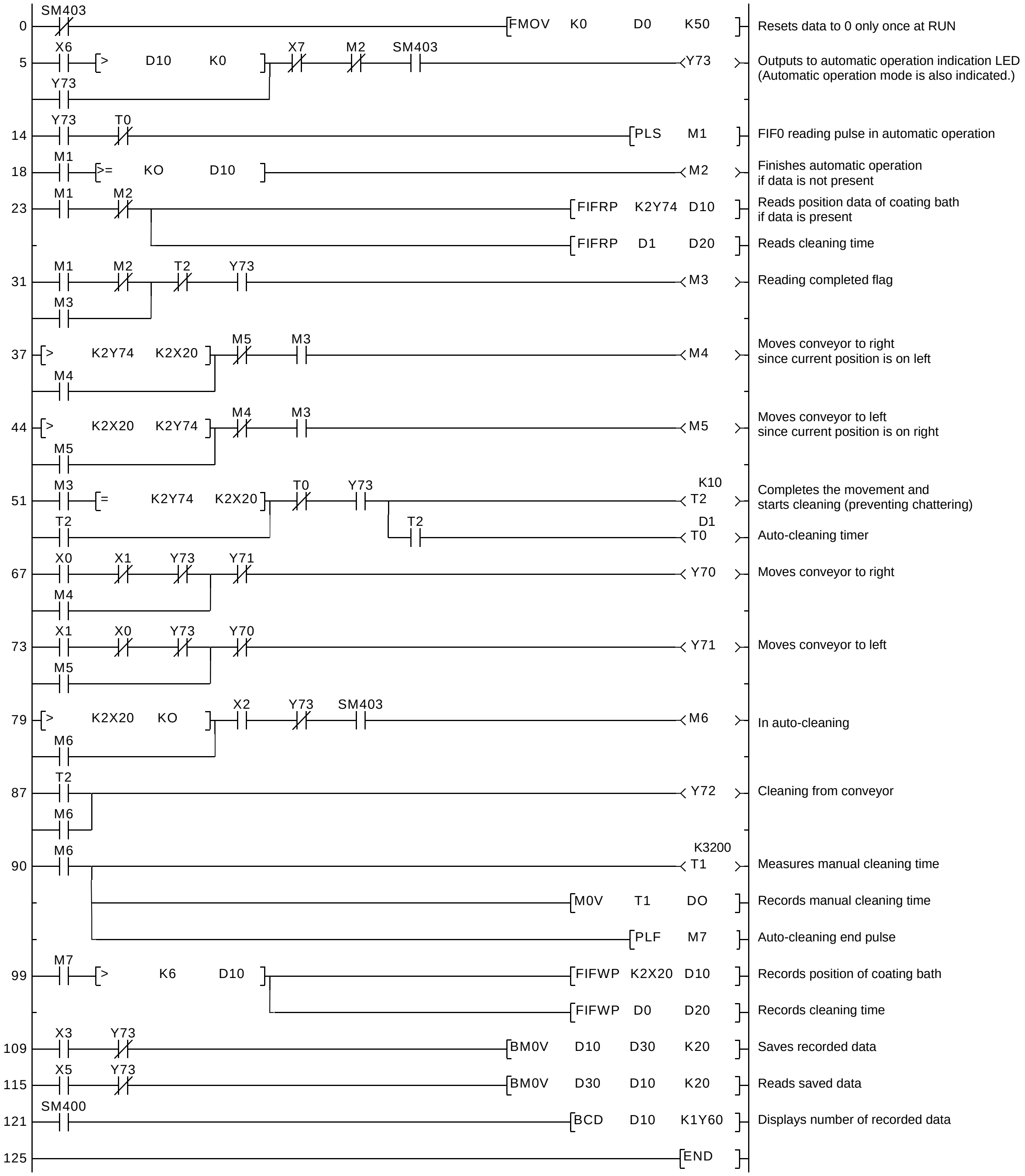
X00 = Manual right moving button
X01 = Manual left moving button
X02 = Manual cleaning button
X03 = Recording data button
X05 = Reading data button
X06 = Automatic operation button
X07 = Operation stop button
Y73 = Automatic operation indication LED
	Y70 = Conveyor, Moving right
	Y71 = Conveyor, Moving left
	Y72 = Conveyor, Cleaning
		X20 = Coating bath-1 (K2X20 = K1)
		X21 = Coating bath-2 (K2X20 = K2)
		X22 = Coating bath-3 (K2X20 = K4)
		X23 = Coating bath-4 (K2X20 = K8)
		X24 = Coating bath-5 (K2X20 = K16)
		X25 = Coating bath-6 (K2X20 = K32)

Start moving to right (X00 = ON)	Moving to right (Y70 = ON) 1)	Standby position (K2X20 = 0) Coating bath-1 (K2X20 = 1)
Stop moving to right (X00 = OFF)	Stop moving (Y70 = OFF)	
Start cleaning (X02 = ON)	Cleaning (Y72 = ON)	
Finish cleaning (X02 = OFF)	Stop cleaning (Y72 = OFF) (A)	
Start moving to right (X00 = ON)	Moving to right (Y70 = ON) 2)	Coating bath-2 (K2X20 = 2)
Stop moving to right (X00 = OFF)	Stop moving (Y70 = OFF)	
Start cleaning (X02 = ON)	Cleaning (Y72 = ON)	
Finish cleaning (X02 = OFF)	Stop cleaning (Y72 = OFF)	
Start moving to right (X00 = ON)	Moving to right (Y70 = ON) 3)	Coating bath-4 (K2X20 = 8)
Stop moving to right (X00 = OFF)	Stop moving (Y70 = OFF)	
Start cleaning (X02 = ON)	Cleaning (Y72 = ON)	
Finish cleaning (X02 = OFF)	Stop cleaning (Y72 = OFF)	
Start moving to left (X01 = ON)	Moving to left (Y71 = ON) 4)	Coating bath-3 (K2X20 = 4)
Stop moving to left (X01 = OFF)	Stop moving (Y71 = OFF)	
Start cleaning (X02 = ON)	Cleaning (Y72 = ON)	
Finish cleaning (X02 = OFF)	Stop cleaning (Y72 = OFF)	
Start moving to right (X00 = ON)	Moving to right (Y70 = ON) 5)	Coating bath-5 (K2X20 = 16)
Stop moving to right (X00 = OFF)	Stop moving (Y70 = OFF)	
Start cleaning (X02 = ON)	Cleaning (Y72 = ON)	
Finish cleaning (X02 = OFF)	Stop cleaning (Y72 = OFF)	
Start moving to right (X00 = ON)	Moving to right (Y70 = ON) 6)	Coating bath-6 (K2X20 = 32)
Stop moving to right (X00 = OFF)	Stop moving (Y70 = OFF)	
Start cleaning (X02 = ON)	Cleaning (Y72 = ON)	
Finish cleaning (X02 = OFF)	Stop cleaning (Y72 = OFF)	
Start automatic operation (X06 = ON ? OFF) Automatic operation indication LED (Y73 = ON)	Moving to left (Y71 = ON) (Starts the automatic operation)	Coating bath-1 (K2X20 = 1)

(A)

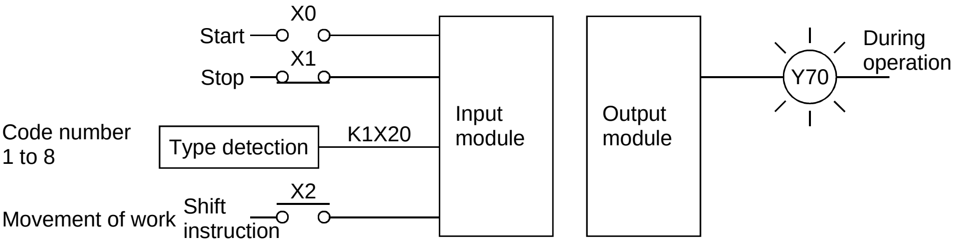
The same operation is repeated from (A).

Project name	QA-9
Program name	MAIN


0 SM403 FMOV K0 D0 K50
Resets data to 0 only once at RUN
5 X6 > D10 K0 X7 M2 SM403 Y73
Y73
Outputs to automatic operation indication LED
(Automatic operation mode is also indicated.)
14 Y73 T0 PLS M1
FIF0 reading pulse in automatic operation
18 M1 >= KO D10 M2
Finishes automatic operation
if data is not present
23 M1 M2 FIFRP K2Y74 D10
Reads position data of coating bath
if data is present
FIFRP D1 D20
Reads cleaning time
31 M1 M2 T2 Y73 M3
M3
Reading completed flag
37 > K2Y74 K2X20 M5 M3 M4
M4
Moves conveyor to right
since current position is on left
44 > K2X20 K2Y74 M4 M3 M5
M5
Moves conveyor to left
since current position is on right
51 M3 = K2Y74 K2X20 T0 Y73 K10 T2
T2
Completes the movement and
starts cleaning (preventing chattering)
T2 D1 T0
Auto-cleaning timer
67 X0 X1 Y73 Y71 Y70
M4
Moves conveyor to right
73 X1 X0 Y73 Y70 Y71
M5
Moves conveyor to left
79 > K2X20 KO X2 Y73 SM403 M6
M6
In auto-cleaning
87 T2 Y72
M6
Cleaning from conveyor
90 M6 K3200 T1
Measures manual cleaning time
M0V T1 DO
Records manual cleaning time
PLF M7
Auto-cleaning end pulse
99 M7 > K6 D10 FIFWP K2X20 D10
Records position of coating bath
FIFWP D0 D20
Records cleaning time
109 X3 Y73 BM0V D10 D30 K20
Saves recorded data
115 X5 Y73 BM0V D30 D10 K20
Reads saved data
121 SM400 BCD D10 K1Y60
Displays number of recorded data
125 END

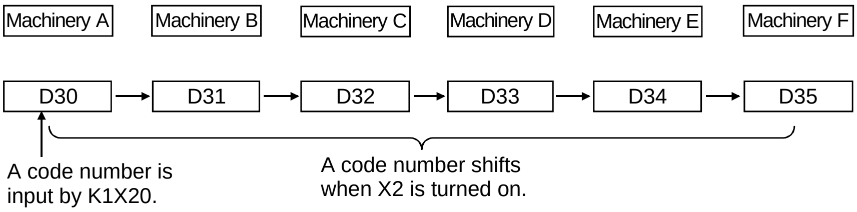
Appendix 4.25 Application example of data shift

Works are conveyed along with their code numbers, and the data register of the processing machinery is analyzed to machine the work according to its code number.

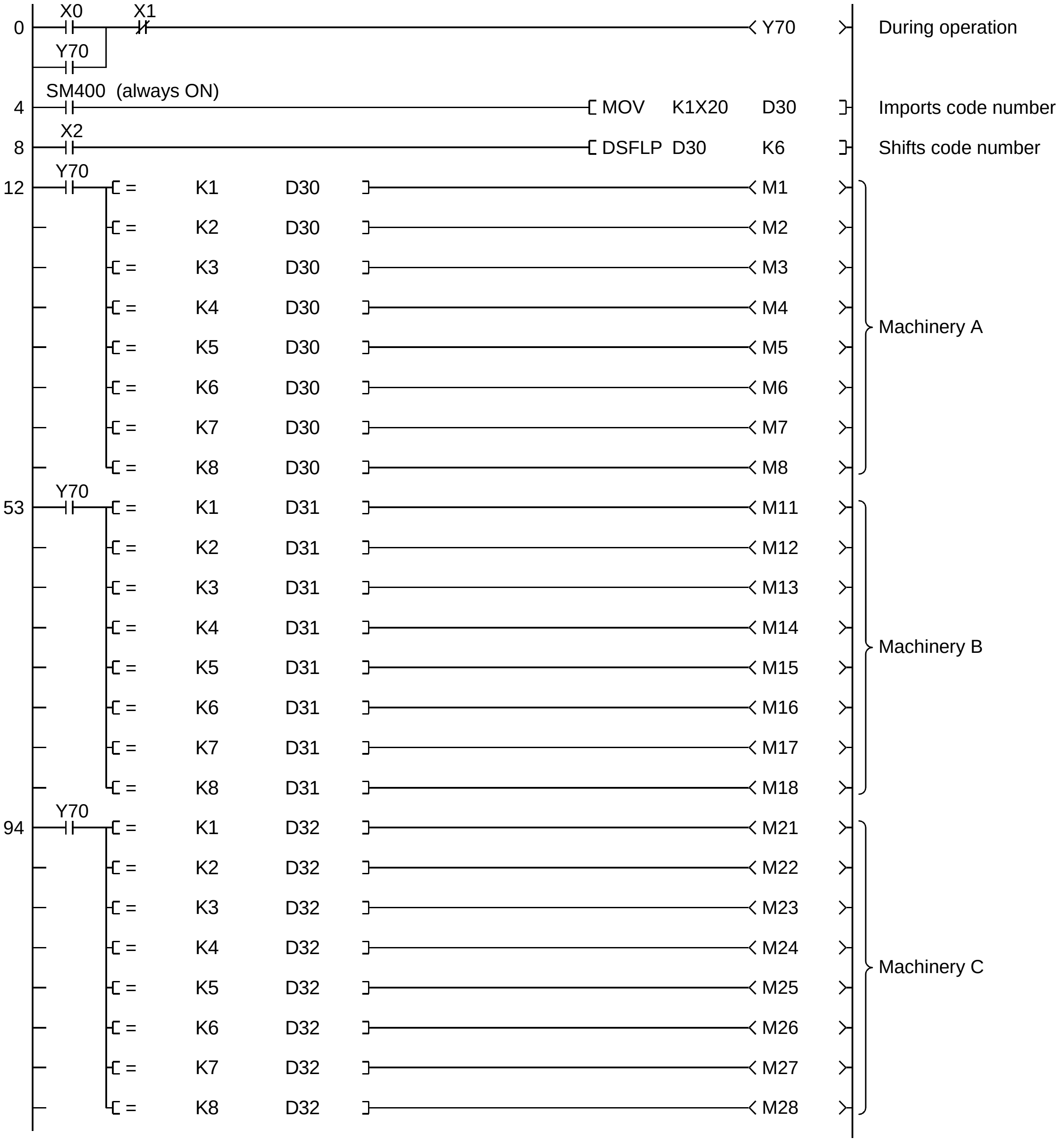


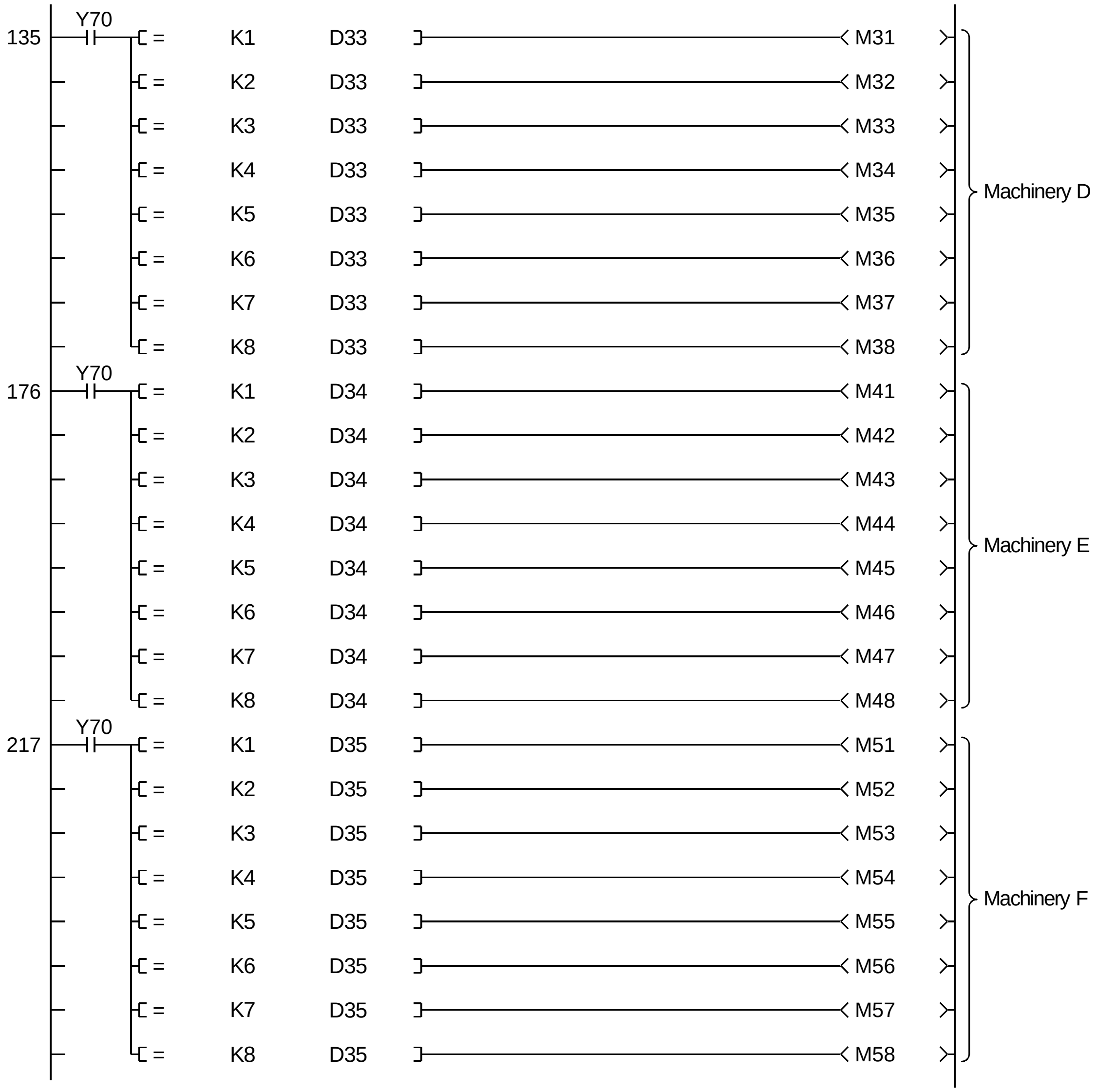
Machinery	Data register	Code 1	Code 2	Code 3	Code 4	Code 5	Code 6	Code 7	Code 8
A	D30	M1	M2	M3	M4	M5	M6	M7	M8
B	D31	M11	M12	M13	M14	M15	M16	M17	M18
C	D32	M21	M22	M23	M24	M25	M26	M27	M28
D	D33	M31	M32	M33	M34	M35	M36	M37	M38
E	D34	M41	M42	M43	M44	M45	M46	M47	M48
F	D35	M51	M52	M53	M54	M55	M56	M57	M58

A code number is stored in the data register, and M corresponding to the stored number is activated to machine the work.



Project name	QA-12
Program name	MAIN

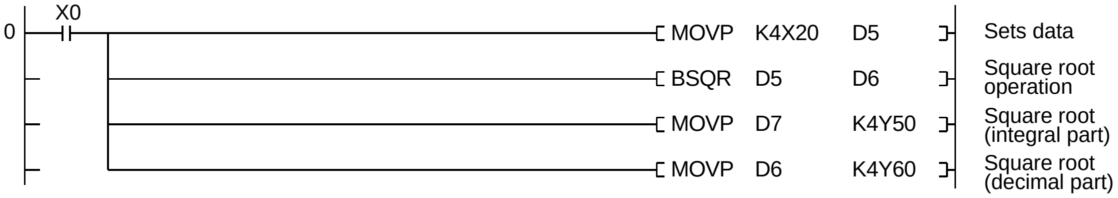



135
Y70
[= K1 D33] (M31)
[= K2 D33] (M32)
[= K3 D33] (M33)
[= K4 D33] (M34)
[= K5 D33] (M35)
[= K6 D33] (M36)
[= K7 D33] (M37)
[= K8 D33] (M38)
Machinery D
176
Y70
[= K1 D34] (M41)
[= K2 D34] (M42)
[= K3 D34] (M43)
[= K4 D34] (M44)
[= K5 D34] (M45)
[= K6 D34] (M46)
[= K7 D34] (M47)
[= K8 D34] (M48)
Machinery E
217
Y70
[= K1 D35] (M51)
[= K2 D35] (M52)
[= K3 D35] (M53)
[= K4 D35] (M54)
[= K5 D35] (M55)
[= K6 D35] (M56)
[= K7 D35] (M57)
[= K8 D35] (M58)
Machinery F

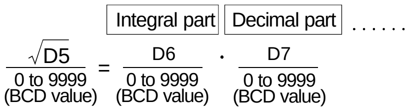
Project name	QA-14
Program name	MAIN

Appendix 4.26 Example of operation program calculating square root of data

The data stored in D5 is calculated to its square root and the result is stored in D6 and D7.



Results of square root operation are stored as follows.



A value in 5th decimal digit is rounded off. Therefore, a value in 4th decimal place has error of ±1.

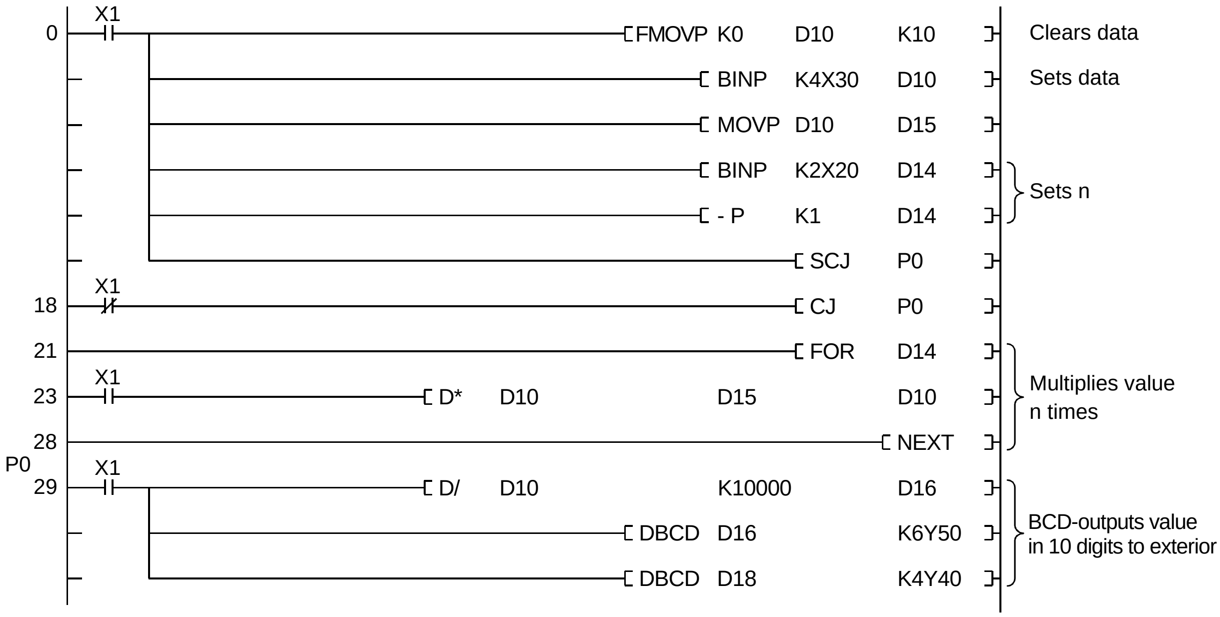
REMARK

QCPUs provide square root operation instructions for data in a real number (floating point) format.

Project name	QA-15
Program name	MAIN

Appendix 4.27 Example of operation program calculating n-th power of data

A value stored in D10 is calculated to its n-th power ("n" is a value stored in D14) and the result is stored in D10.

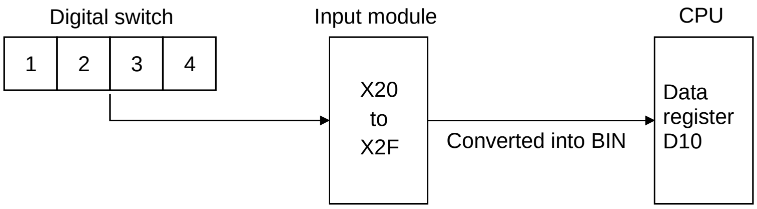


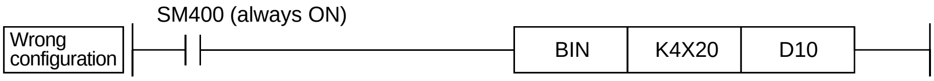
NOTE

An operation error occurs if a value in D10 exceeds 2147483647.

Appendix 4.28 Program using digital switch to import data

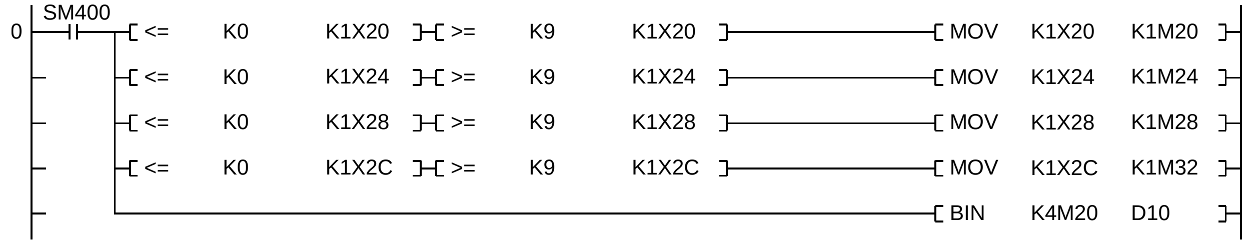
When a set value of the digital switch is always input and stored to D10 of the programmable controller



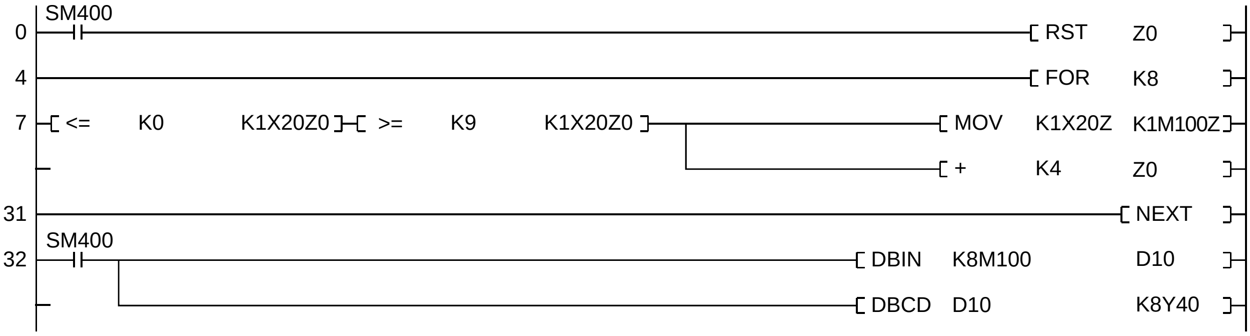


In the above program, changing a value of the digital switch while the programmable controller is in RUN may cause codes other than 0 to 9 depending on the timing of the change, which may cause an operation error of the CPU.
To avoid this, write a program as follows.

(Example 1) For 4 digits of X20 to X2F


0
SM400
[<= K0 K1X20]-[>= K9 K1X20]
[MOV K1X20 K1M20]
[<= K0 K1X24]-[>= K9 K1X24]
[MOV K1X24 K1M24]
[<= K0 K1X28]-[>= K9 K1X28]
[MOV K1X28 K1M28]
[<= K0 K1X2C]-[>= K9 K1X2C]
[MOV K1X2C K1M32]
[BIN K4M20 D10]

(Example 2) For 8 digits of X20 to X3F



Appendix 4.29 Displaying number of faults and fault numbers using fault detection program

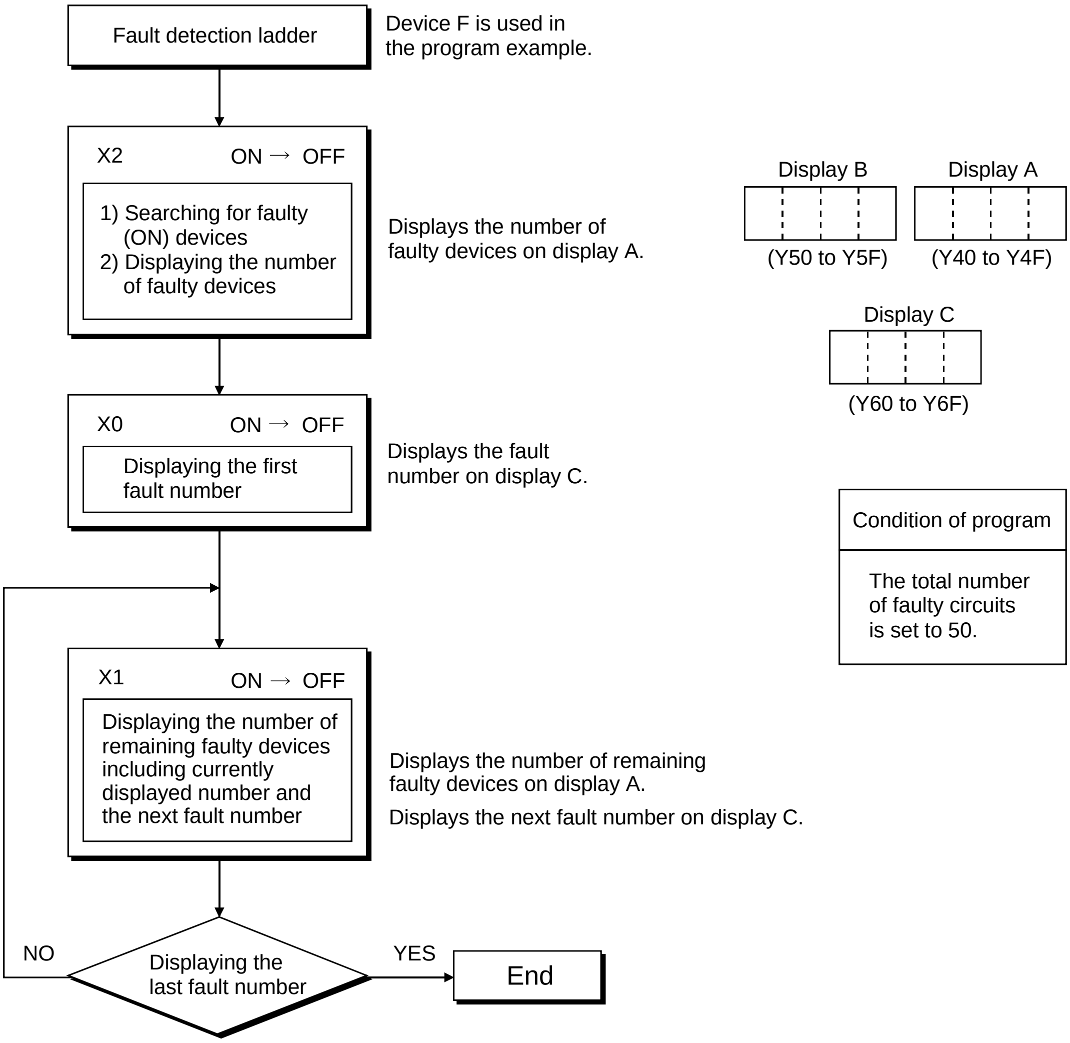
The following program sequentially displays the number of turned-on bit devices (such as X, M, and F) among many bit devices being used continuously, together with their device numbers.

[Application example]

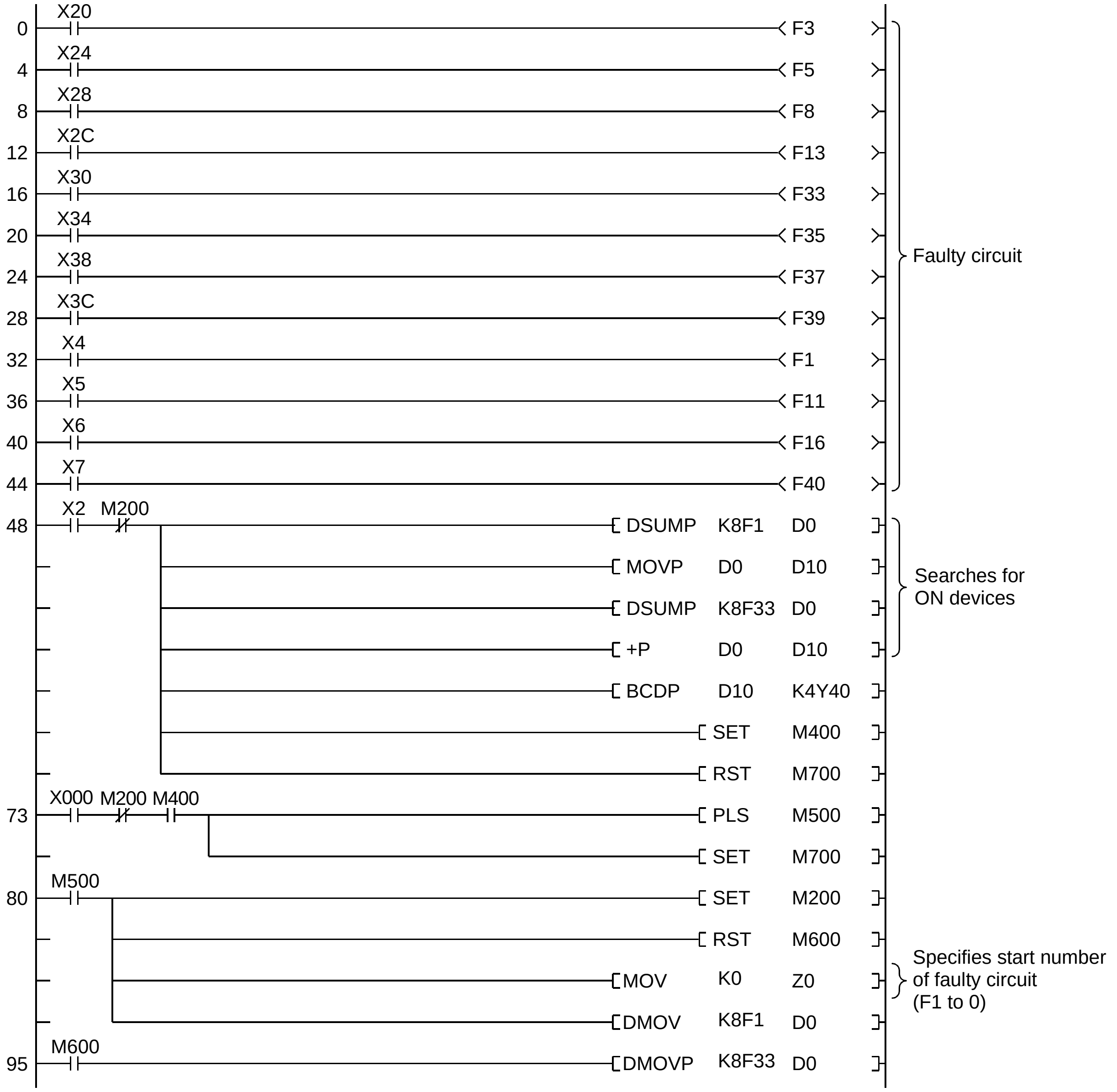
When M or F is used as an output device of a fault detection program, use the following program to obtain a certain fault number from the faults.

[Sequence program flow]

(Operating procedure)


Fault detection ladder
Device F is used in the program example.
X2
ON → OFF
1) Searching for faulty (ON) devices
2) Displaying the number of faulty devices
Displays the number of faulty devices on display A.
X0
ON → OFF
Displaying the first fault number
Displays the fault number on display C.
X1
ON → OFF
Displaying the number of remaining faulty devices including currently displayed number and the next fault number
Displays the number of remaining faulty devices on display A.
Displays the next fault number on display C.
NO
Displaying the last fault number
YES
End
Display B
Display A
(Y50 to Y5F)
(Y40 to Y4F)
Display C
(Y60 to Y6F)
Condition of program
The total number of faulty circuits is set to 50.

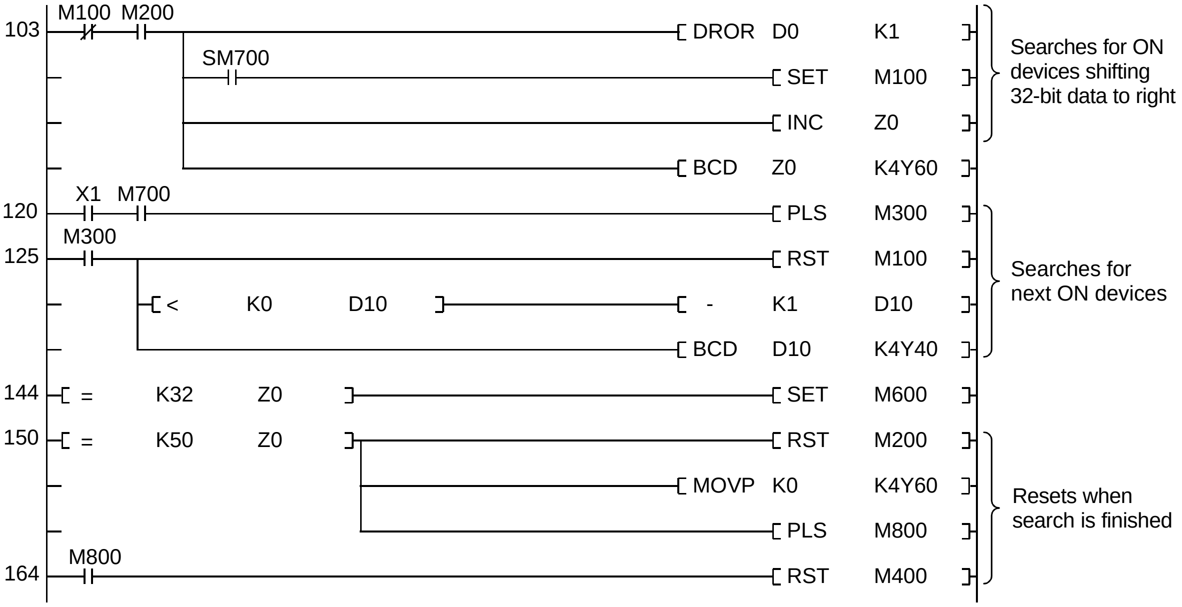
Project name	QA-31
Program name	MAIN



Faulty circuit

Searches for ON devices

Specifies start number of faulty circuit (F1 to 0)



Searches for ON devices shifting 32-bit data to right

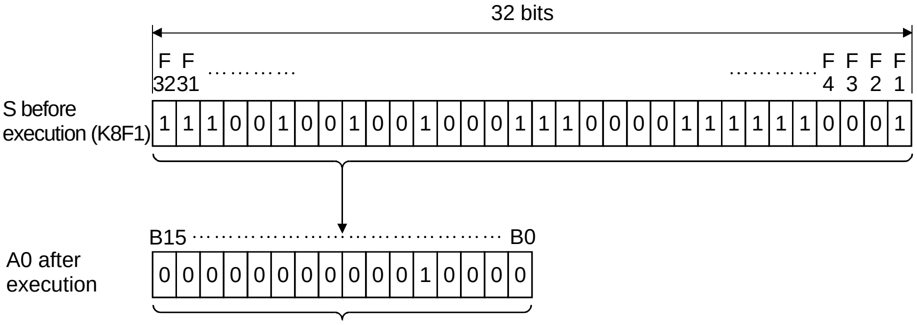
Searches for next ON devices

Resets when search is finished

(1) Searching for ON devices

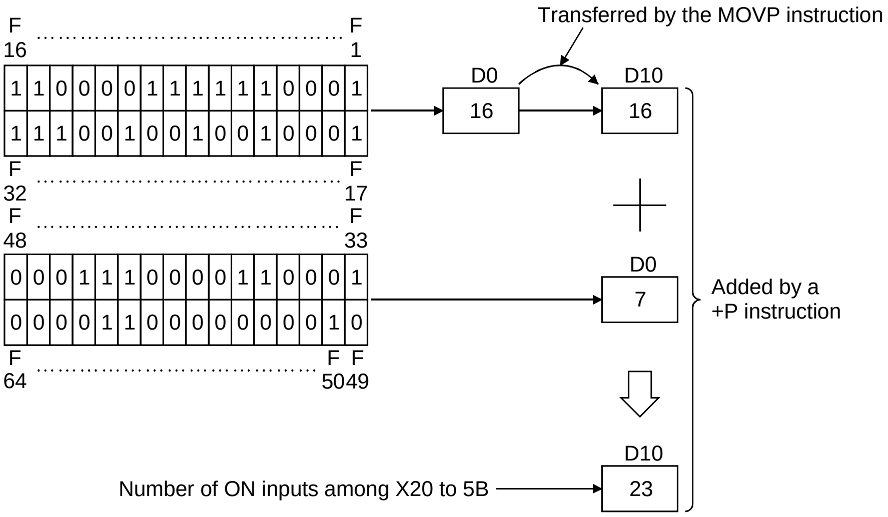
DSUMP	K8F1	D0

DSUMP	K8F33	D0



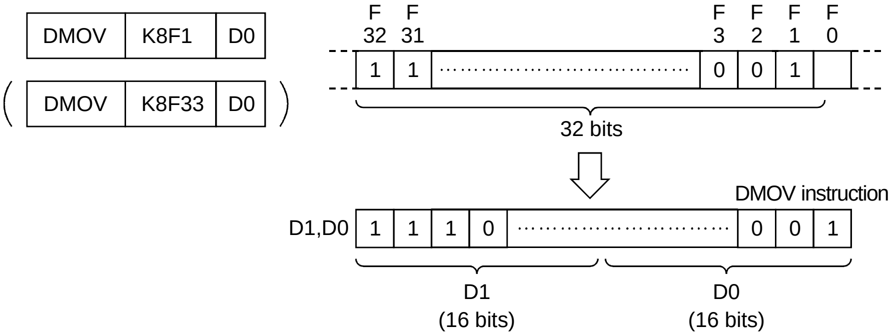
The total number of bits with 1 is stored in BIN.
(in this example, 16)

Turning on X2 stores the number of turned-on bits among F1 to F64 to D10 and display it.

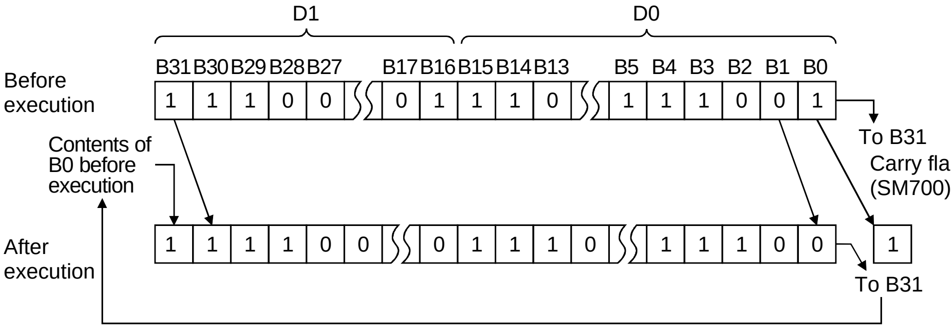


(2) Searching for ON devices shifting 32-bit data to right

DROR	D0	K1



DROR	D0	K1



(a) Turning on X0 sets the above shift data (D0 and D1). After that, the data is shifted right by 1 bit at each scan until a turned-on bit is detected.
When a turned-on bit is detected, shifting stops in that scan (SM700 turns on), and the accumulated number of shifts (equivalent to a device number) is displayed.

(b) Each time X1 is turned on, the next turned-on bit is detected and the detected device number is displayed. At the same time, 1 is subtracted from the number of turned-on bits which have been obtained in advance to display the remaining number of turned-on bits.

Appendix 5 Memory and File to be Handled by CPU Module

- Data to be stored in memories
  The following table lists the data and drive numbers which can be stored in the program memory, standard RAM, standard ROM, and memory card.

Item	CPU module built-in memory			Memory card (RAM)	Memory card (ROM)		File name and extension	Remarks
	Program memory	Standard RAM	Standard ROM	SRAM card	Flash card	ATA card		
	Drive 0*1	Drive 3*1	Drive 4*1	Drive 1*1	Drive 2*1			
Parameter	○	×	○	○	○	○	PARAM.QPA	1 data/drive
Intelligent function module parameter*2	○	×	○	○	○	○	IPARAM.QPA	1 data/drive
Program	⦿	×	○*3	○*4	○*4	○*4	***.QPG	-
Device comment	○*5	×	○*6	○*6	○*6	○*6	***.QCD	-
Device initial value	○	×	○	○	○	○	***.QDI	-
Device data	×	×	○	×	×	×	***.QST	-
File register	×	○*7*8	×	○	○*9	×	***.QDR	-
Local device	×	○*7	×	○	×	×	***.QDL	1 data/CPU module
Sampling trace file	×	○*7	×	○	×	×	***.QTD	-
Error history data	×	×	×	×	×	×	***.QFD	-
Device data storage file	×	×	○	×	×	×	DEVSTORE.QST	-
Module error collection file	×	○	×	×	×	×	IERRLOG.QIE	-
Backup data file	×	×	×	○	○	○	MEMBKUP0.QBP	-
Programmable controller user data	×	×	○	×	×	○*10	***.***	-
User setting system area*11	○	×	×	×	×	×	-	-

⦿: Required, ○: Storable, ×: Not storable

*1: A drive number is used to specify a memory to be written/read by the external device using a sequence program or MC protocol.
Since the memory name is used to specify the target memory in GX Works2, the drive number needs not to be considered.
*2: Store the intelligent function module parameters in the same drive with the parameters.
When they are stored in different drives, the intelligent function module parameters do not become valid.
*3: A program stored in the standard ROM cannot be executed.
Store the program to the program memory before execution.
*4: To execute a program stored in the memory card, make the setting in the Boot File tab of the Q Parameter Setting window.
*5: The device comments cannot be read by instructions in a sequence program.
*6: Reading from a sequence program requires several scans.
*7: Only each one of file register, one local device, and sampling trace file can be stored in the standard RAM.
*8: For the number of storable file registers, refer to QnUCPU User's Manual Function Explanation, Program Fundamentals.
*9: A sequence program allows reading only. No data can be written from the sequence program.
*10: Data can be written or read with the following instructions.
- SP.FREAD (batch-reads data from the specified file in the memory card.)
- SP.FWRITE (batch-writes data to the specified file in the memory card.)

*11: Set an area used by the system.

- Memory capacities and necessity of formatting
  The following tables list the memory capacities and necessity of formatting of each memory.

		Q00UJCPU	Q00UCPU	Q01UCPU	Q02UCPU	Q03UD(E)CPU	Formatting
Program memory		10K steps (40K byte)		15K steps (60K byte)	20K steps (80K byte)	30K steps (120K byte)	*1
Standard ROM		256K byte	512K byte			1024K byte	Unnecessary
Standard RAM		-	128K byte			192K byte	*1
Memory card	SRAM card	-			Q2MEM-1MBS: 1M byte Q2MEM-2MBS: 2M byte Q3MEM-4MBS: 4M byte Q3MEM-8MBS: 8M byte		Necessary (use GX Works2.)
	Flash card	-			Q2MEM-2MBF: 2M byte Q2MEM-4MBF: 4M byte		Unnecessary
	ATA card	-			Q2MEM-8MBA: 8M byte Q2MEM-16MBA: 16M byte Q2MEM-32MBA: 32M byte		Necessary (use GX Works2.)

		Q04UD(E)H CPU	Q06UD(E)H CPU	Q010UD(E)H CPU	Q13UD(E)H CPU	Q20UD(E)H CPU	Q26UD(E)H CPU	Formatting
Program memory		40K steps (160K byte)	60K steps (240K byte)	100K steps (400K byte)	130K steps (520K byte)	200K steps (800K byte)	260K steps (1040K byte)	*1
Standard ROM		1024K byte		2048K byte		4096K byte		Unnecessary
Standard RAM		256K byte	768K byte	1024K byte		1280K byte		*1
Memory card	SRAM card	Q2MEM-1MBS: 1M byte Q2MEM-2MBS: 2M byte Q3MEM-4MBS: 4M byte Q3MEM-8MBS: 8M byte						Necessary (use GX Works2.)
	Flash card	Q2MEM-2MBF: 2M byte Q2MEM-4MBF: 4M byte						Unnecessary
	ATA card	Q2MEM-8MBA: 8M byte Q2MEM-16MBA: 16M byte Q2MEM-32MBA: 32M byte						Necessary (use GX Works2.)

*1: When the memory contents become indefinite in the initial status or due to the end of battery life, the memory is automatically formatted after the programmable controller is powered off and then on or is reset. Make sure to format the memory in GX Works2 before using.

Appendix 6. Comparison with GX Developer (changes)

(1) Supported CPU modules

The following table lists the CPU modules that are supported in GX Works2.

Programmable controller series	Programmable controller type
QCPU (Q mode)	High Performance model QCPU (Q02, Q02H, Q06H, Q12H, Q25H)
	Universal model QCPU (Q00UJ, Q00U, Q01U, Q02U, Q03UD, Q03UDE, Q04UDH, Q04UDEH, Q06UDH, Q06UDEH, Q10UDH, Q10UDEH, Q13UDH, Q13UDEH, Q20UDH, Q20UDEH, Q26UDH, Q26UDEH, Q50UDEH, Q100UDEH)

The following table lists the CPU modules that are not supported in GX Works2.
Use GX Developer for the following CPU modules.

Programmable controller series	Programmable controller type
QCPU (Q mode)	Basic model QCPU (Q00J, Q00, Q01)
	Process CPU (Q02PH, Q06PH, Q12PH, Q25PH)
	Redundant CPU (Q12PRH, Q25PRH)
	Remote I/O master (QJ71LP21, QJ71BR11)
QCPU (motion)	All programmable controller types
QCPU (A mode)	All programmable controller types
QSCPU	All programmable controller types
QnACPU	All programmable controller types
ACPU	All programmable controller types
Motion controller (SCPU)	All programmable controller types
CNC (M6, M7)	All programmable controller types

(2) Unsupported features

The following table lists the features that are not supported in GX Works2.
Use GX Developer, GX Simulator, or GX Configurator for the following features.

Unsupported feature		Alternate S/W
Online function	TEL function	GX Developer GX Simulator
Debug function for ladder program	Monitor condition/Monitor stop condition setting function	
	Scan time measurement function	
	Skip/Parts/Step execution function	
Debug function for ST program	Debug function	
	Breakpoint function	
Intelligent function module programming function	Protocol FB support function	GX Configurator-SC
Intelligent function module debug function	Debug support function	
Online function for positioning module	Trace function	GX Configurator-QP
	System monitor function	
	Test mode function	
Device initial value function	Device memory registration function	GX Developer
	Printing function	
Password function	Password registration function for data in project	GX Developer
Interaction with GX Explorer	Boot by GX Explorer	GX Developer
Interaction with PX Developer	Boot by PX Developer	GX Developer
Interaction with GX Converter	I/O function with GX Converter	GX Developer
MEDOC print format import	Import in MEDOC print format	GX Developer
Online function	Intelligent module diagnostics from system monitor	GX Developer GX Configurator
Sampling trace function	Sampling trace function conditionally on step number	GX Developer

(3) Supported project types

The following table lists the project types that are supported in GX Works2.

Project type	Description
Simple project (without labels)	This is the equivalent of the "Do not use label" project of GX Developer. 1) When a project created in the "Do not use label" of GX Developer is read with GX Works2, the project becomes the Simple project (without labels). 2) When a project created in the Simple project (without labels) of GX Works2 is read with GX Developer, the project becomes the "Do not use label" project.
Simple project (with labels)	This is the equivalent of the "Use label" project of GX Developer. 1) When a project created in "Use label" of GX Developer is read with GX Works2, the project becomes the Simple project (with labels). 2) When a project created in the Simple project (with labels) of GX Works2 is read with GX Developer, the project becomes the "Use label" project.
Structured project	In GX Works2, "structured programming" is available. The structured programming proceeds while creating POUs and combining them (registering tasks in the program file). 1) When a project created in "Use label" with ST of GX Developer is read with GX Works2, the project becomes "Structured Project" 2) The projects created in "Structured Project" of GX Works2 cannot be read with GX Developer.

(a) Using project functions

Before using the project function in GX Works2, review the following precautions.

Function	Description (differences between GX Developer and GX Works2)	
	GX Developer	GX Works2
Protect projects	By installing projects as "monitoring only", the projects can be protected on each personal computer.	By setting projects as "read-only with the "Security" function, project-by-project protection is now available.
Change project types	Project types cannot be changed from "Do not use label" to "Use label".	The following project type changes are now available. 1) From "Simple project (without labels)" to "Simple project (with labels)" 2) From "Simple project (with labels)" to "Structured Project" * Project type cannot be changed directly from "Simple project (without labels)" to "Structured Project".
Read GX Developer format projects	Selecting [Project] → [Open Other Project] can read GX Developer format projects.	
Read GX Configurator-QP format projects	Selecting [Project] → [Intelligent Function Module] → [Import GX Configurator-QP Data] can read GX Configurator-QP format projects.	
Copy data in a project to different projects	It is enabled on the project copy dialog box.	Copy and paste is now available in the Project window.

(4) Programming languages supported by each project type
The following table lists the programming languages that are supported by each project type of GX Works2.

Project type	Supported programming language
Simple project (without labels)	Ladder, SFC (MELSAP3)
Simple project (with labels)	Ladder, SFC (MELSAP3) * Supported program element: label, structure, function block
Structured project	Ladder, SFC (MELSAP3), structured ladder, ST * Supported program element: label, structure, function block, function block, library

The following programming languages are not supported in GX Works2.
Use GX Developer for the following programming languages.

Project type	Supported programming language
List	1) When GX Works2 reads out a program created with lists in GX Developer, it can be displayed or edited in ladder. 2) When GX Developer reads out a program created with ladder in GX Works2, it can be displayed or edited in list.
MELSAP-L	1) When GX Works2 reads out a program created with MELSAP-L in GX Developer, it can be displayed or edited in ladder. 2) When GX Developer reads out a program created with SFC (MELSAP3) in GX Works2, it can be displayed or edited in MELSAP-L.

(a) Using ladder language
Before using the ladder language in GX Works2, review the following precautions.

Function	Description (differences between GX Developer and GX Works2)	
	GX Developer	GX Works2
Program giving devices an alias	It is enabled by the "Alias" function.	Use "Label".
Segment a part of program into POUs (macros)	It is enabled by the "Macro definition/ import" function.	Use "Function Block".
Find/Replace instructions/devices/labels	Find is enabled by directly typing an instruction/device/label in "Read mode".	Pressing the Space key on the ladder editor allows the simple find.
Check use status of device/label	It is enabled by the "Cross Reference List" function and "List of Used Devices" functions.	Select [Find/Replace] → [Cross Reference], or [Find/Replace] → [Device List].
Merge the programs	It is enabled by the "Merge Data" function.	Use copy and paste on the label editor.
Verify	No corresponding function	The Verify Result window clearly shows the following: "unmatched area of the programs", "only verification source contains the program" and "only verification destination contains the program".

(b) Using SFC (MELSAP3) language

Before using the SFC (MELSAP3) language in GX Works2, review the following precautions.

Function	Description (differences between GX Developer and GX Works2)	
	GX Developer	GX Works2
Change block number	It is enabled by the "copy and paste" function in block list.	Each block data is displayed in the Project window, and the block number can be changed in the property of each block data. * Selecting [View] → [Open SFC Blocklist] can display the block list equivalent to that of GX Developer.
Auto scroll	A new block diagram can be opened by block start.	Selecting [View] → [Open Zoom/Start Destination Block] can open it.
Open a start source block by block start	No corresponding function	Selecting [View] → [Back to Start SFC Block] can open it.
Open operation/transition condition programs	Moving a cursor on the SFC diagram can display zoom (operation output /transition condition).	Selecting [View] → [Open Zoom/Start Destination Block] can open it. Or double-clicking while pressing the Ctrl key also can open it.
	Multiple zooms (operation output/transition condition) can be simultaneously displayed. * Changing the "Setting of Zoom Display" option can switch the display in a window in the same way as GX Developer.	

(c) Using labels

Before using labels in GX Works2, review the following precautions.

Function	Description (differences between GX Developer and GX Works2)	
	GX Developer	GX Works2
Check devices automatically assigned to labels	It is enabled by the "Show assigned device" function of label editor.	Check on the ladder editor by selecting [View] → [Device Display].
Import/Export device comments to labels	It is enabled by the "device comment import" function and "device comment export" functions.	Use the copy and paste on the label editor and device comment editor.
Use pointer-type labels	Local pointers are assigned.	Common pointers are now assigned. For projects with labels, 2048 points are set by default in "Common Pointer No." in the "PLC System" tab of PLC Parameter.
Unusable reserved words for label name	The definition of reserved words is different between GX Developer and GX Works2.	

(d) Using function blocks
Before using function blocks in GX Works2, review the following precautions.

Function	Description
Use function blocks created with ladder	Function blocks created with ladder can be used for ladder program, ST program, and SFC program operation outputs. * When using function blocks created with ladder for ST programs, select [Tool] → [Options] → [Compile] → [Basic Setting] → "Enable function block call 'from ladder to Structured Ladder/FBD' and 'from Structured Ladder/FBD or ST to ladder".
Use function blocks created with structured ladder	Function blocks created with structured ladder can be used for ladder programs, structured ladder programs and ST programs.
Use function blocks created with ST	Function blocks created with ST can be used for ladder programs, structured ladder programs, and ST programs. * When using function blocks created with ST for ladder programs, select [Tool] → [Options] → [Compile] → [Basic Setting] → "Enable function block call 'from ladder to Structured Ladder/FBD' and 'from Structured Ladder/FBD or ST to ladder".
When the option "Enable function block call 'from ladder to Structured Ladder/FBD' and 'from Structured Ladder/FBD or ST to ladder" is set	When the VAR_IN_OUT input variable and output variable have different label/device, the input variable value is always equal to the output variable value.

(5) Using device comments
Before using device comments in GX Works2, review the following precautions.

Function	Description (differences between GX Developer and GX Works2)	
	GX Developer	GX Works2
Delete comments of unused devices	It is enabled by the "Delete unused comments" function.	After checking the unused device by selecting [Find/Replace] → [Device List], delete the device comment directly.
Sample comment	Sample comments of the special relay/special register are provided in project format.	Comments of the special relay/special register and intelligent function module can be imported by the "Import from Sample Comment" function on the device comment editor.

(6) Using device memory
Before using the device memory in GX Works2, review the following precautions.

Function	Description
Device memory display	Multiple device ranges can be displayed in a window. * By selecting "All Range" when devices are input, all the device ranges can be displayed in a window in the same way as that of GX Developer.
Copy and past device memory data to Excel	To copy and paste device memory data to Excel, select [Tool] → [Read from Excel File]/[Write to Excel File].

(7) Using device initial values
Before using device initial values in GX Works2, review the following precautions.

Function	Description (differences between GX Developer and GX Works2)	
	GX Developer	GX Works2
Maximum amount of device initial value data to be created	Only one set of data can be created.	Up to 800 sets of data can be created.
Restriction of device number	The device number must be within the maximum points of each programmable controller of devices.	The device number must be within the device setting range of the PLC parameter.
Write to PLC/read from PLC IC memory card write/read	Only 1 data can be read and written.	Selected multiple data can be read and written.

(8) Using online function
Before using the online function in GX Works2, review the following precautions.

Function	Description (differences between GX Developer and GX Works2)	
	GX Developer	GX Works2
Connection destination setting	A project can contain only one set of "connection destination" information.	A project can contain multiple sets of "connection destination" information. To change the "connection destination" information, select "Connection Destination" in the Project window.
Write/Read data to/from intelligent function modules	Data can be written or read to/from CPU modules and intelligent function modules simultaneously.	Selecting [Online] → [Write to PLC] writes data to CPU modules and intelligent function modules simultaneously. Selecting [Online] → [Read from PLC] reads data from CPU modules and intelligent function modules simultaneously.
Write data to the Flash ROM of the CPU module	It is enabled by the "PLC write (Flash ROM)" function.	The "PLC write (Flash ROM)" function is now integrated in the "Write to PLC" function. Select [Online] → [Write to PLC].
Remote operation window PLC diagnostics window System monitor window	Selecting [Online] → [Remote Operation] and [Diagnostics] → [System Monitor]/[PLC Diagnostics] can display the module image and the programmable controller CPU operation status is now easy to see. The remote operation, memory operation, and clock setup can be started from the PLC Diagnostics window.	

Function	Description
Read from PLC	Symbolic information in GX Developer format does not include SFC programs. Read symbolic information on "Simple project (without labels)".
	If symbolic information of GX Developer or GX IEC Developer is read out, the project becomes uncompiled.

(9) Using monitor/debug function
Before using the monitor/debug function in GX Works2, review the following precautions.

Function	Description
Entry device monitor	The "entry device monitor" function is now a docking window as a "watch" function so that it can be displayed without overlapping with the program editor.
	Device/label is now enabled to be entered by dragging and dropping from the program editor and the on/off status of bit devices and current values of word devices can be modified on the monitor window.
Device batch monitoring Buffer batch monitoring	The "device batch monitoring" and "buffer memory batch monitoring" functions are now integrated to realize the same operability.
	The on/off status of bit devices and current values of word devices can be modified on the monitor window.
Monitor and test intelligent function modules	To use the monitoring or test function to FL-net (OPCN-2) interface unit and AS-i master unit, execute the "watch" and "Device/Buffer memory batch monitor" function.

(10) Using printing function
Before using the printing function in GX Works2, review the following precautions.

Function	Description
Additional information print such as statement and device comment	The displayed image is printed or previewed. To print additional information such as a statement and device comment, put the target information on the screen and then select [Project] → [Print Window]/[Print Window Preview].

(11) Copying saved project data
Before copying project data saved in GX Works2, review the following precautions.

Function	Description (differences between GX Developer and GX Works2)	
	GX Developer	GX Works2
Copy saved project data	Saved project data can be copied by copying files under the project name folder.	Copy all the workspace name folders and "workspacelist.xml" created in the same hierarchy as the workspace name folders.

(12) Compatibility with GX Developer
For the compatibility between GX Developer and GX Works2, review the following precautions.

Function	Description
Open projects in other formats	Before opening a GX Developer "Use label" project of which a program and function block have the same name, change the data name in GX Developer.
	Function names of ST language are different between GX Developer and GX Works2. Compile the program and correct errors.
Export projects to GX Developer format file	Applicable projects are the following; 1) Simple project (without labels) 2) Compiled Simple project (with labels) Projects using labels in SFC language are executed.
	The project can be saved in GX Developer format when none of the following is applied. 1) No device is set. 2) The length of the label name exceeds 16 characters. 3) Label name contains a device name or reserved word. 4) An invalid character is used. 5) Data type which is not supported by GX Developer is used. 6) A value which is not constant is used in the constant.
	Data registered to the global label is set as "Auto External" for all the local labels.

(13) Compatibility with GX IEC Developer

For the compatibility between GX IEC Developer and GX Works2, review the following precautions.

Function	Description
Open projects in other formats	Function names of ST language are different between GX IEC Developer and GX Works2. Compile the program and correct errors.
User library	Before using GX IEC Developer user libraries which a password is set to, cancel the password in GX IEC Developer.

(14) Key operation

This section explains the differences of the key operation between GX Developer and GX Works2.

Function			Description	Shortcut key	
				GX Developer	GX Works2
Edit		Read mode	Activates the read mode.	Shift + F2	- (*1)
		Write mode	Activates the write mode.	F2	- (*1)
Find/Replace		Cross reference	Displays the cross reference.	-	
		Device List	Displays the device list.	-	
Convert		Convert (all programs being edited)	Converts all programs being edited.	Ctrl + Alt + F4	-
View		Project data list	Switches display/non-display of the project data list.	Alt + O	-
		Switch between the project data list and window	Switches between the project data list and each window.	Alt + 7	-
		Switch between ladder and list	Switches between the ladder window and list window.	Alt + F1	-
Online	Monitor	Monitor (all the windows)	Monitors ladders of all the opened programs.	Ctrl + F3	-
		Monitor (write mode)	Activates the write mode during ladder monitoring.	Shift + F3	- (*2)
		Stop monitor (all the windows)	Stops the ladder monitoring for all the opened programs.	Ctrl + Alt + F3	-
	Debug	Device test	Turns on or off the device forcibly or modifies the current value.	Alt + 1	-
		Skip execution	Executes selected sequence programs in skip execution.	Alt + 2	-
		Partial execution	Executes sequence programs partially.	Alt + 3	-
		Step execution	Executes the programmable controller CPU in step execution.	Alt + 4	-
		Remote operation	Executes remote operations.	Alt + 6	-

*1: In GX Works2, switching the ladder editor to the read mode/write mode is unnecessary. The ladder can be edited any time.

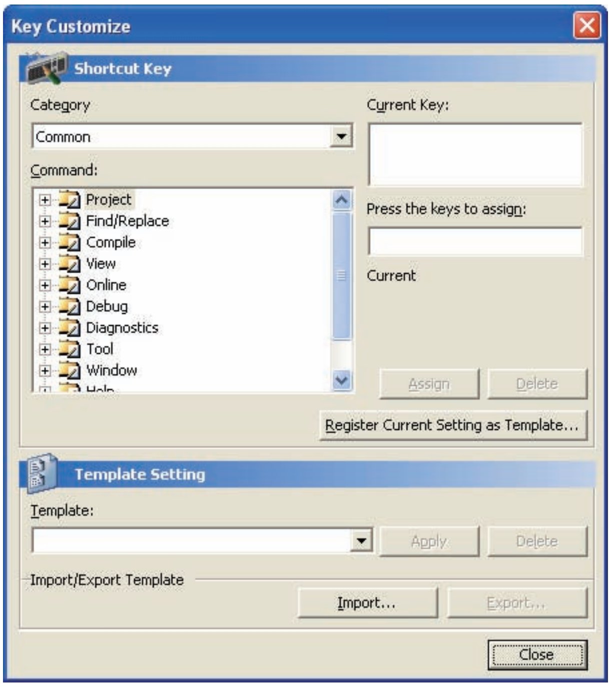
*2: In GX Works2, switching the ladder editor to the monitor (write mode) during the ladder monitoring is unnecessary.
Even during the ladder monitoring, the ladder can be edited and written to the programmable controller in the RUN status.

Appendix 7 Customizing Shortcut Keys

Shortcut keys of each function can be customized.
Customized shortcut keys can be registered as a template and utilized.

Screen display

Select [Tool] → [Key Customize].



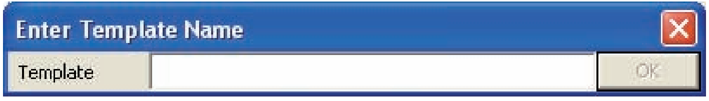
Item		Description
Shortcut Key		-
	Category	Select a category from the group list categorized by window.
	Command	Select a function name whose shortcut key is to be changed.
	Current Key	Displays the shortcut key assigned to the selected command.
	Press the keys to assign	Specify a new shortcut key to be assigned. Pressing a key(s) on the keyboard assigns the key(s). Example) Ctrl + 5
	Current	Displays the menu name to which the entered shortcut key is assigned. When the key is already assigned to another function, the function name is displayed.
Template		Select a template of shortcut keys from the list box. • Default Setting The default setting is set. • GPPA Format Setting The shortcut key setting at ladder programming is changed to the same setting as that at GPPA.

Screen button

Assigns the shortcut key. The assigned shortcut key is displayed in "Current Key".

Deletes the shortcut key selected in "Current Key".

The Enter Template Name screen is displayed.
Register the assigned shortcut keys as a template with a name.
The registered template is displayed in "Template".



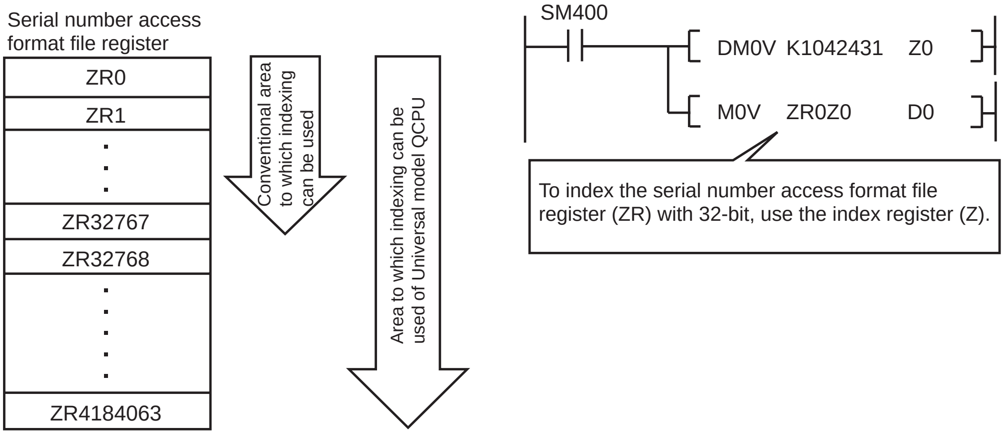
The selected template of shortcut keys is applied.

Deletes a template selected in "Template".

Imports a pre-saved template file (*.gks) and adds it to "Template".

Saves a template selected in "Template" as a template file (*.gks).

In the Universal model QCPU (excludes Q00UJCPU), expanding the index register to 32 bits enables the indexing for all the file register areas.


Serial number access format file register
ZR0
ZR1
ZR32767
ZR32768
ZR4184063
Conventional area to which indexing can be used
Area to which indexing can be used of Universal model QCPU
SM400
DM0V K1042431 Z0
M0V ZR0Z0 D0

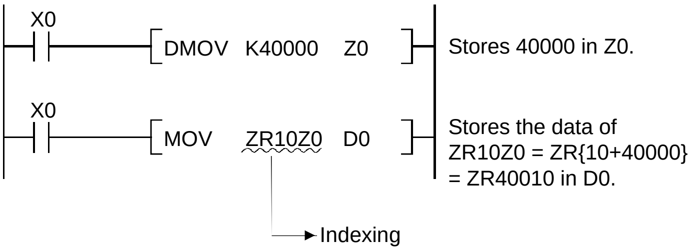
To index the serial number access format file register (ZR) with 32-bit, use the index register (Z).

A method for specifying index registers for 32-bit indexing can be selected from following two methods.

- Specifying the index range used for 32-bit indexing
- Specifying the 32-bit indexing using "ZZ" specification

(1) When specifying the index range used for 32-bit indexing

(a) Each index register can be set between -2147483648 and 2147483647. The following shows an example of indexing.



Stores 40000 in Z0.

Stores the data of ZR10Z0 = ZR{10+40000} = ZR40010 in D0.

(b) Specification method

For indexing with a 32-bit index register, specify the start number of an index register to be used on the Device tab of the PLC parameter setting screen in GX Works2.

POINT

When the start number of the index register used is changed on the Device tab of the PLC parameter setting screen, do not change the parameters only or do not write only the parameters into the programmable controller. Be sure to write the parameters into the programmable controller with the program.
When the parameter is forced to be written into the programmable controller, an error of CAN'T EXE. PRG. occurs. (Error code: 2500)

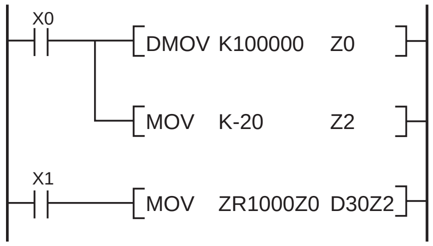
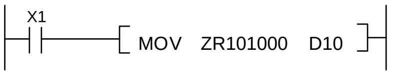
(c) Device for which indexing can be used
Indexing can be used only for the devices shown below.

- ZR: Serial number access format file register
- D: Extended data register
- W: Extended link register

(d) Usable range of index registers
The following table shows the usable range of index registers for indexing with 32-bit index registers.
For indexing with 32-bit index registers, the specified index register (Zn) and the next index register of the specified register (Zn+1) are used. Be sure not to overlap index registers to be used.

Setting value	Index registers to be used	Setting value	Index registers to be used
Z0	Z0, Z1	Z10	Z10, Z11
Z1	Z1, Z2	Z11	Z11, Z12
Z2	Z2, Z3	Z12	Z12, Z13
Z3	Z3, Z4	Z13	Z13, Z14
Z4	Z4, Z5	Z14	Z14, Z15
Z5	Z5, Z6	Z15	Z15, Z16
Z6	Z6, Z7	Z16	Z16, Z17
Z7	Z7, Z8	Z17	Z17, Z18
Z8	Z8, Z9	Z18	Z18, Z19
Z9	Z9, Z10	Z19	Cannot be specified.

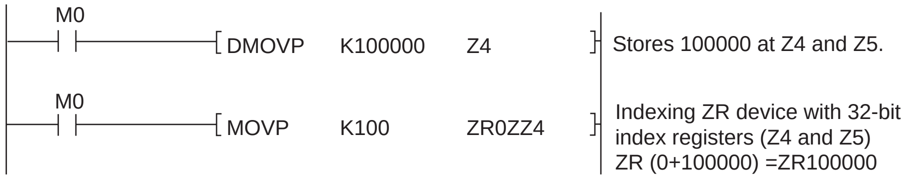
(e) The following shows an example of indexing and the actual process device.
(When Z0 (32-bit) is 100000 and Z2 (16-bit) is -20)

Ladder example	Actual process device
 X0 DMOV K100000 Z0 MOV K-20 Z2 X1 MOV ZR1000Z0 D30Z2	 Description ZR1000Z0···ZR(1000+100000)=ZR101000 D30Z2······ D(30-20)=D10

(2) When specifying the 32-bit indexing using "ZZ" specification

(a) One index register can specify 32-bit indexing using "ZZ" specification such as "ZR0ZZ4".

The following shows the 32-bit indexing with "ZZ" specification.


M0
DMOVP K100000 Z4
Stores 100000 at Z4 and Z5.
M0
MOVP K100 ZR0ZZ4
Indexing ZR device with 32-bit index registers (Z4 and Z5)
ZR (0+100000) =ZR100000

(b) Specification method

For 32-bit indexing using "ZZ" specification, select "Use ZZ" in [Indexing Setting for ZR Device] in the Device tab in PLC parameter setting screen.

(c) Device for which indexing can be used

Indexing can be used only for the devices shown below.

- ZR: Serial number access format file register
- D: Extended data register
- W: Extended link register

(d) Usable range of index registers

The following table shows the usable range of index registers in 32-bit indexing with "ZZ" specification.

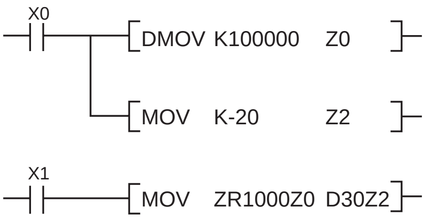
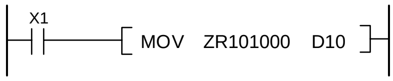
The 32-bit indexing with "ZZ" specification is specified as the format ZRmZZn.

Specifying ZRmZZn enables Zn and Zn+1 of 32-bit values to index the device number of ZRm.

"ZZ" specification*	Index registers to be used	"ZZ" specification*	Index registers to be used
□ZZ0	Z0, Z1	□ZZ10	Z10, Z11
□ZZ1	Z1, Z2	□ZZ11	Z11, Z12
□ZZ2	Z2, Z3	□ZZ12	Z12, Z13
□ZZ3	Z3, Z4	□ZZ13	Z13, Z14
□ZZ4	Z4, Z5	□ZZ14	Z14, Z15
□ZZ5	Z5, Z6	□ZZ15	Z15, Z16
□ZZ6	Z6, Z7	□ZZ16	Z16, Z17
□ZZ7	Z7, Z8	□ZZ17	Z17, Z18
□ZZ8	Z8, Z9	□ZZ18	Z18, Z19
□ZZ9	Z9, Z10	□ZZ19	Cannot be specified.

*: □ indicates a device name (ZR, D, W) for indexing target

(e) The following shows an example of the 32-bit indexing with "ZZ" specification and the actual processing device.
(When Z0 (32-bit) is 100000 and Z2 (16-bit) is -20)

Ladder example	Actual process device
 X0 DMOV K100000 Z0 MOV K-20 Z2 X1 MOV ZR1000Z0 D30Z2	 X1 MOV ZR101000 D10 Description ZR1000Z0···ZR(1000+100000)=ZR101000 D30Z2······ D(30-20)=D10

(f) Available functions for "ZZ" specification
The 32-bit indexing specification with "ZZ" specification applies to the following functions of GX Works2.

No.	Function name and description
1	Specifying devices in program instruction
2	Entry device monitor
3	Device test
4	Device test with conditions
5	Monitor condition setting
6	Sampling trace (Trace point (specifying devices), trace target device)

POINT

ZZn cannot be used alone as a device like "DMOV K100000 ZZ0".
When setting values of index registers to specify 32-bit indexing with "ZZ" specification, set the value of Zn (Z0 to Z19).
ZZn alone cannot be input to each function of GX Works2.

For details, refer to the QnUCPU User's Manual Function Explanation, Program Fundamentals and MELSEC-Q/L Programming Manual (Common Instruction).

Appendix 9　FB

Appendix 9.1　FB

FB is an abbreviation for a Function Block that is designed to convert a ladder block, which is used repeatedly in a sequence program, into a component (FB) to be utilized in a sequence program.

This not only increases the efficiency of program development but also reduces programming mistakes to improve program quality.

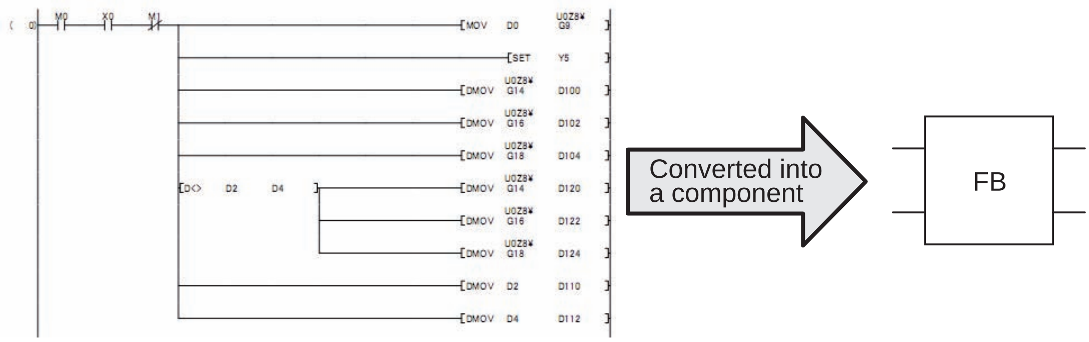


Figure App. 9.1 Converting a sequence program into a component

Appendix 9.1.1　Conversion into components

The following section explains the process to convert a simple program into a component.

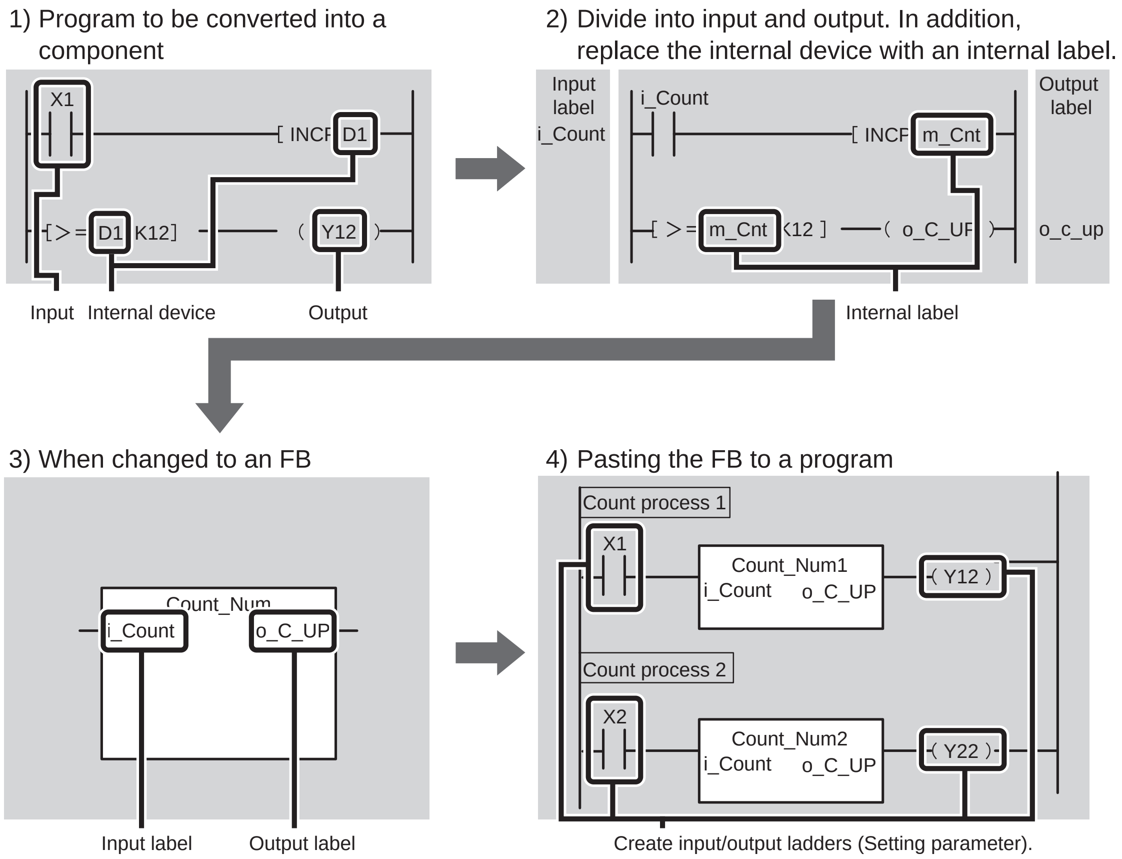


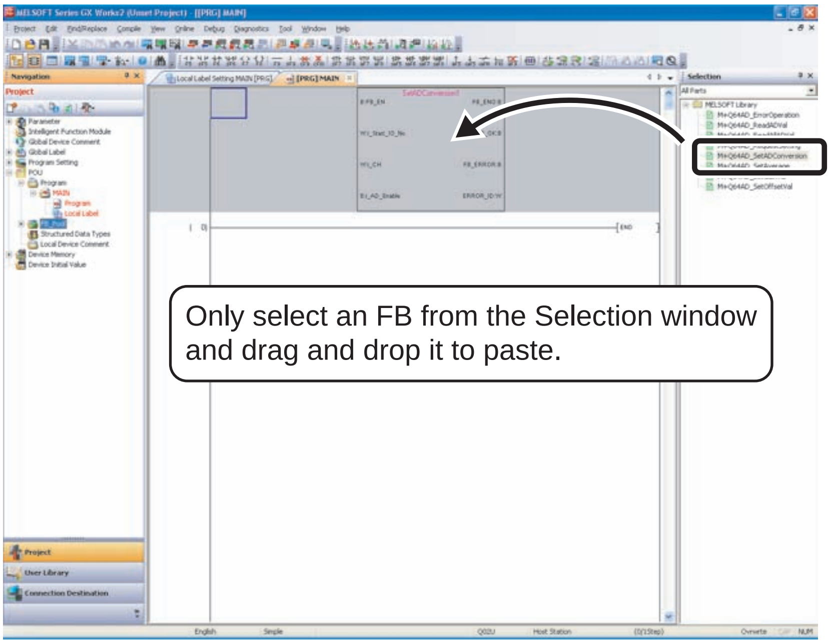
Figure App. 9.2 Flow of conversion into components

Appendix 9.1.2 Advantages of using FBs

This section introduces advantages of creating programs by using FBs.

(1) Easy programming

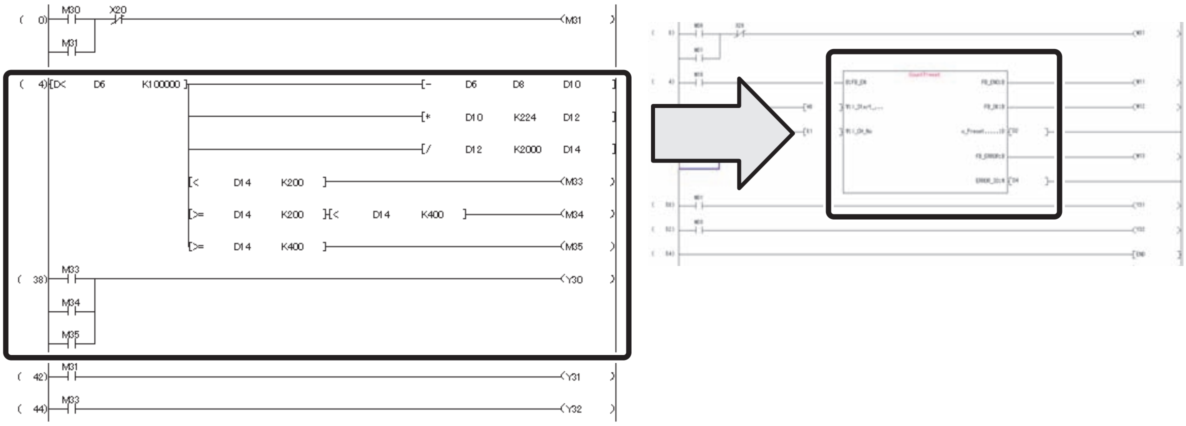
A sequence program can be created simply by pasting FBs. This significantly reduces the program development man-hours. (FB libraries provided by Mitsubishi Electric Corporation. makes programming easier.)



Only select an FB from the Selection window and drag and drop it to paste.

(2) Easy reading

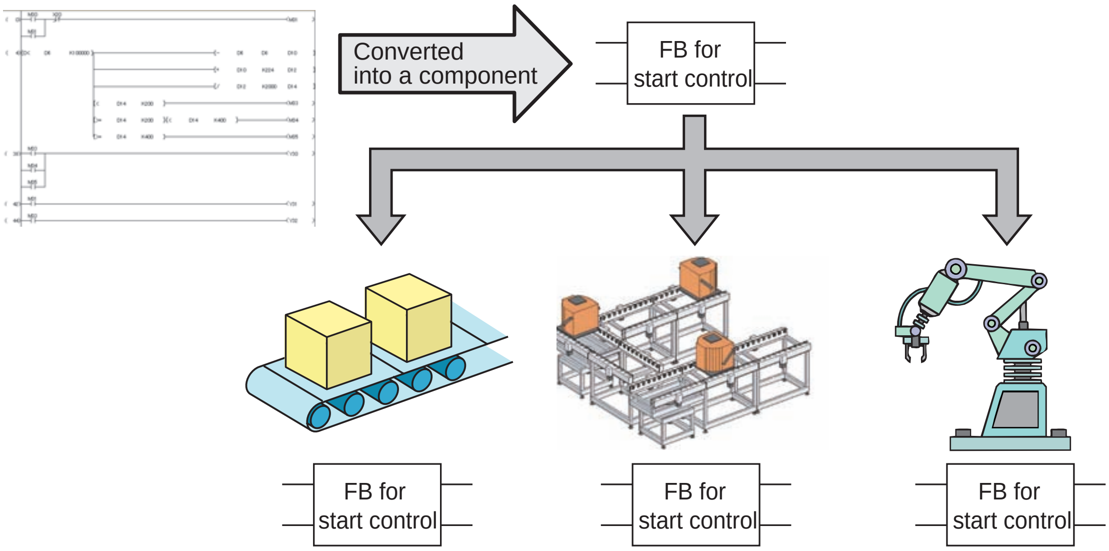
Using an FB creates a simple program with only a "box" (FB), an input, and an output to create an easy-to-read sequence program.



(3) Reusing

Converting a standard program into a component allows the program to be reused any number of times.

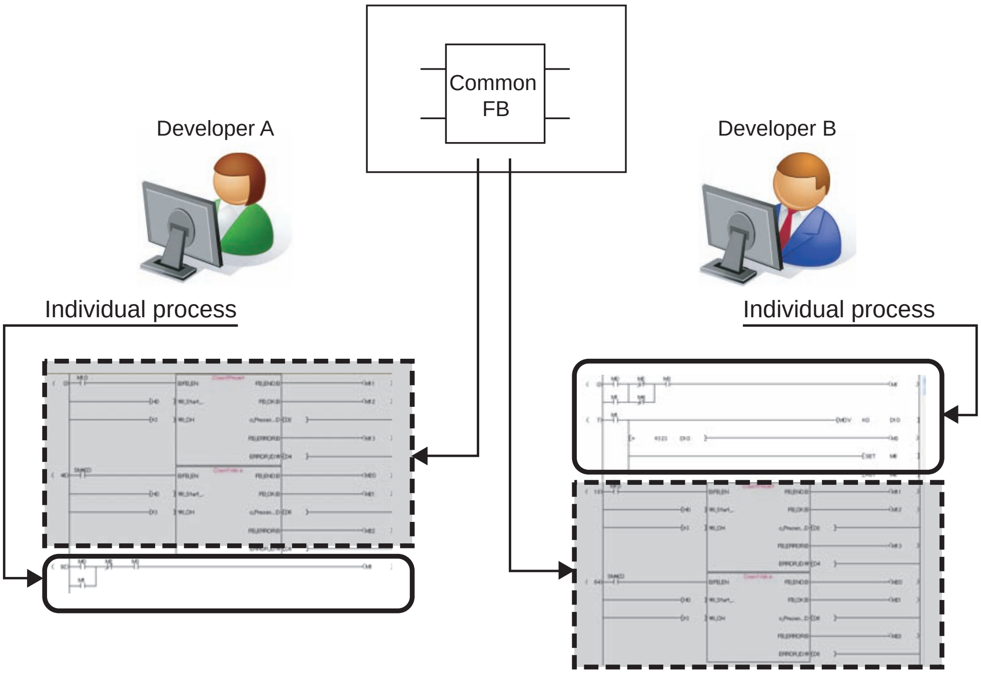
As a result, operations such as copying a sequence program and modifying a device, which have often been required in the past, will be unnecessary.



(4) Improving quality

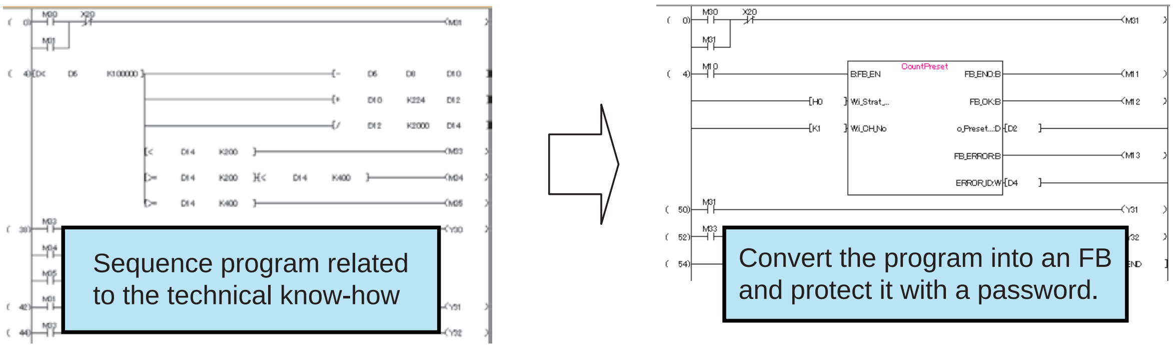
Converting a standard program into a component as an FB to reuse the program allows development of programs of consistent quality, without relying on the technological skill of the program developers.

When developers A and B are developing sequence programs for different devices, using the same FB for the common processing enables the developers to create consistent quality of sequence programs.



(5) Protecting assets

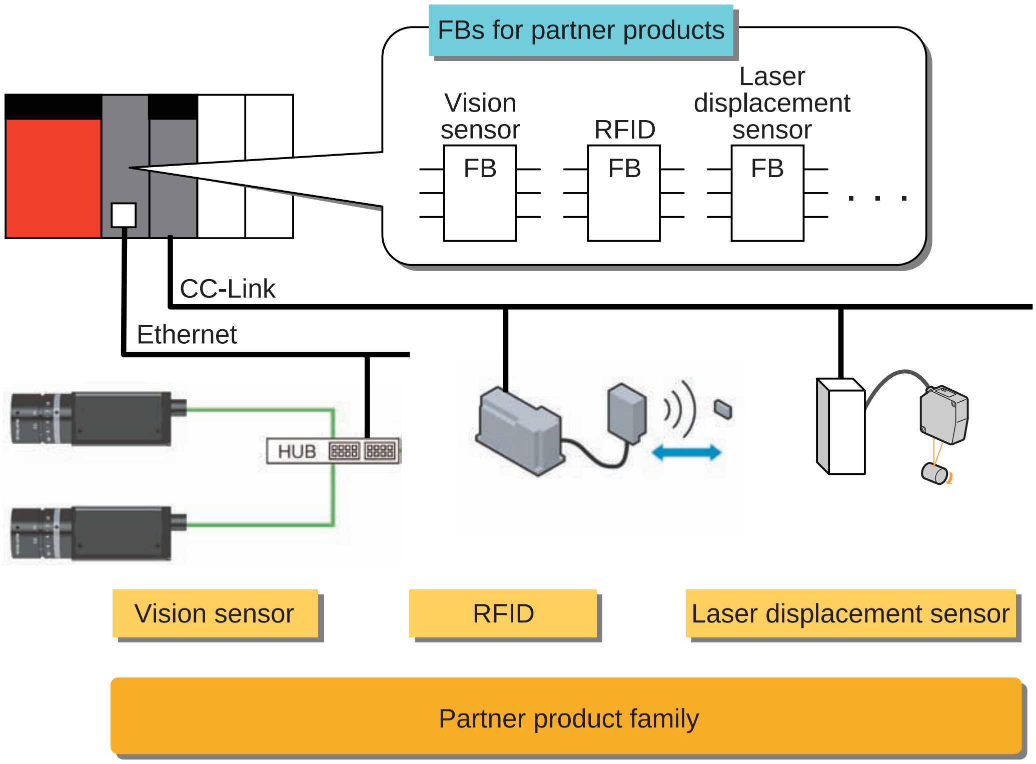
By setting up a block password, the created FB can be protected so that it cannot be viewed.


Sequence program related to the technical know-how
Convert the program into an FB and protect it with a password.
CountPreset

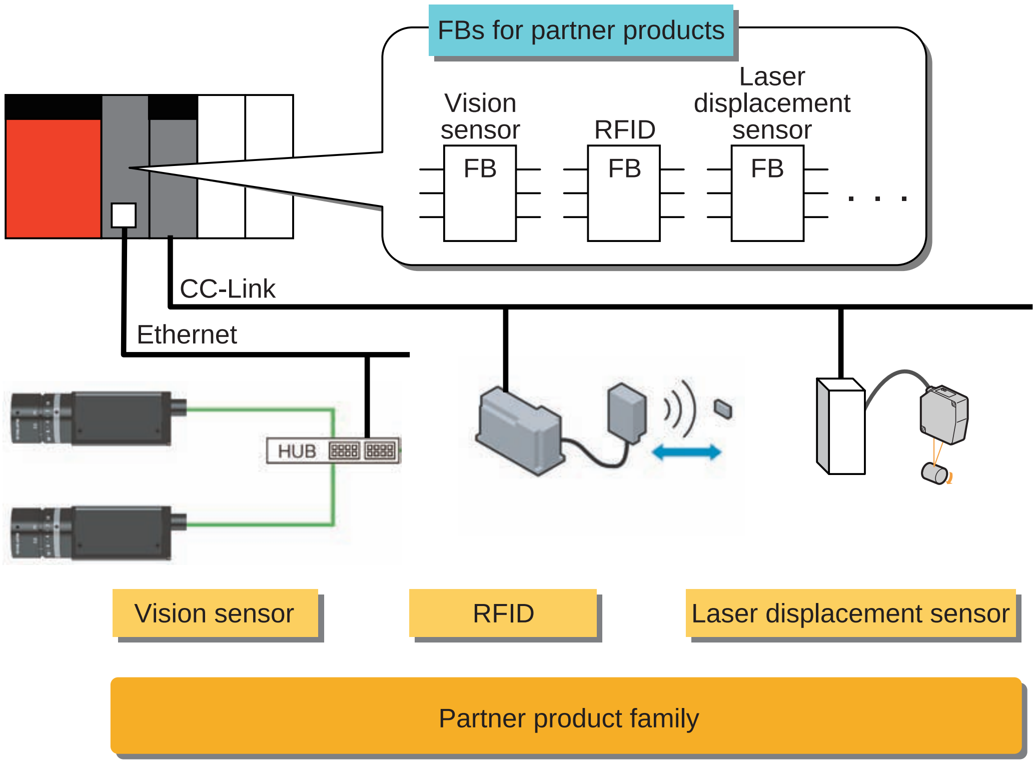
Appendix 9.1.3 FB Libraries

An FB library is a collection of FBs that are usable in GX Works2 (Simple project). Using an FB library enables easy setting and operation of MELSEC-Q/L modules and partner products.

<Example of MELSEC-Q/L module>



<Example of partner product>



(1) FB library lineup
FB libraries include "FBs for MELSEC-Q/L modules" and "FBs for partner products".

(2) How to obtain FB libraries
FB libraries can be obtained from Mitsubishi Electric FA site.

URL http://www.mitsubishielectric.co.jp/fa/index.html

For the procedure to obtain the FB libraries, refer to App. 9.2.2 "Preparations prior to use of FB libraries".

Appendix 9.1.4 Development tool

GX Works2 (Simple project) ver 1.12N or later is required to develop sequence programs using FBs.

POINT

Depending on the FB library, supporting versions of GX Works2 may differ.
For details, refer to the download page of each FB.

Appendix 9.1.5 FB specifications and precautions

The following specifications and precautions must be understood prior to using FBs.

1. An FB cannot be used in another FB.
2. Because an FB specific process is added when an FB is arranged, the number of steps increases when compared to a ladder created without an FB.
3. FBs cannot be used in an interruption program.
4. FBs whose execution does not complete within a scan cannot be used in the FOR to NEXT instruction loops or subroutine programs.

Appendix 9.2 Creating a program by using an FB library

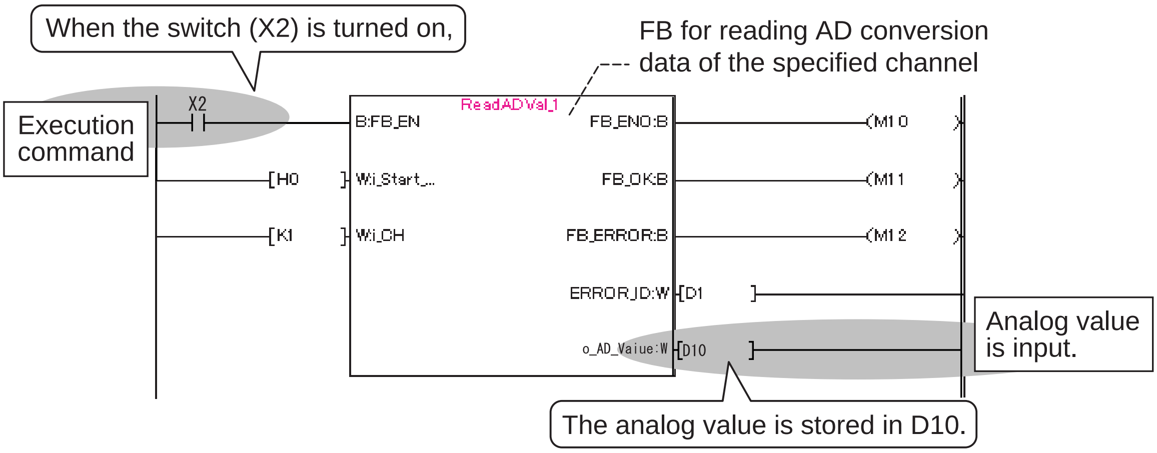
This section explains the procedure to create a program by using an FB library.

Appendx 9.2.1 Programs to be created

This section explains how to use an FB library with an example of importing an analog value from an analog input module.

Example) Reading an analog value to D10 from the analog input module (Q64AD) when the switch (X2) is turned on.

The program can easily be created by using an FB library as follows.



When the switch (X2) is turned on,

Execution command

FB for reading AD conversion data of the specified channel

Analog value is input.

The analog value is stored in D10.

POINT

The FB created by a user is also available other than the FB in the FB library.

For the creation method of a new FB, refer to "MELSOFT GX Works2 FB Quick Start Guide".

Appendix 9.2.2 Preparations prior to use of FB libraries

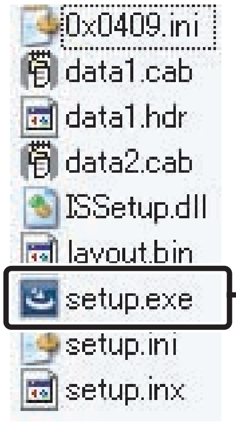
Before using an FB library, contact your distributor to obtain it.
(FB libraries will not be installed when installing GX Works2.)

The following explains operation procedures using the FB library for Q64AD as an example.

1) As the file obtained from your distributor is a zip format file, unzip "q64ad_v100a.zip".

⇩

2) Double-click "setup.exe" in "q64ad_v100a".



0x0409.ini
data1.cab
data1.hdr
data2.cab
ISSetup.dll
layout.bin
setup.exe
setup.ini
setup.inx

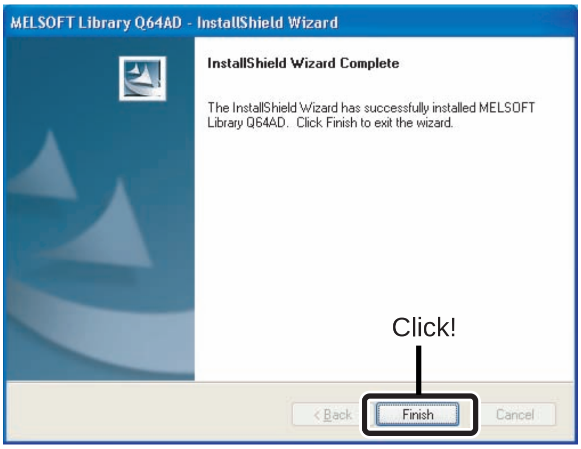
Double-click!

⇩

3) The screen for installation is displayed. Follow the instructions to complete the installation.

⇩

4) The following dialog is displayed when the installation is completed. Click the OK button to close the dialog.



MELSOFT Library Q64AD - InstallShield Wizard

InstallShield Wizard Complete

The InstallShield Wizard has successfully installed MELSOFT Library Q64AD. Click Finish to exit the wizard.

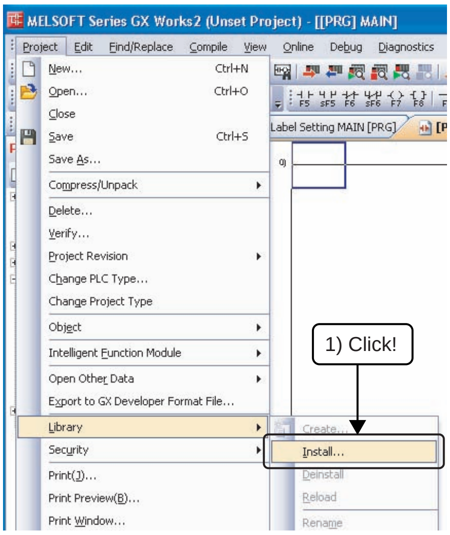
Click!

This completes the preparation prior to use of FB libraries.

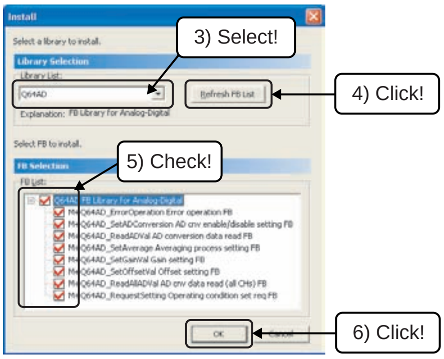
Appendix 9.2.3 Importing an FB library to projects

This section explains how to import an FB library for analog input module (Q64AD) to be pasted to the program into a project.

Create a new project before the following operation.
(refer to section 2.3.2)



1) Click [Project] → [Library] → [Install].


Install
Select a library to install.
Library Selection
Library List:
Q64AD
Refresh FB List
Explanation: FB Library for Analog-Digital
Select FB to install.
FB Selection
FB List:
Q64AD FB Library for Analog-Digital
M+Q64AD_ErrorOperation Error operation FB
M+Q64AD_SetADConversion AD cnv enable/disable setting FB
M+Q64AD_ReadADVal AD conversion data read FB
M+Q64AD_SetAverage Averaging process setting FB
M+Q64AD_SetGainVal Gain setting FB
M+Q64AD_SetOffsetVal Offset setting FB
M+Q64AD_ReadAllADVal AD cnv data read (all CHs) FB
M+Q64AD_RequestSetting Operating condition set req FB
OK
Cancel
3) Select!
4) Click!
5) Check!
6) Click!

2) The Install dialog box is displayed.

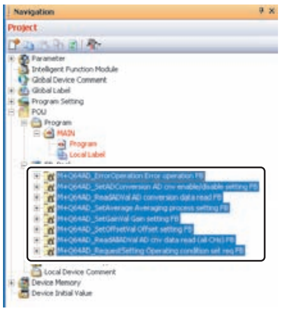
3) Select "Q64AD"from Library List.

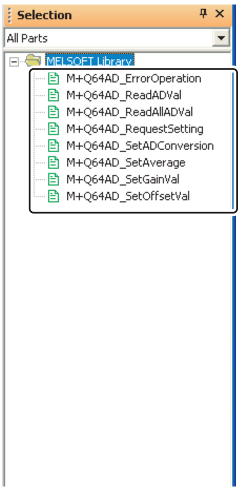
4) Click the Refresh FB List button.

5) Check the library to import.

6) Click the OK button.

7) The imported FBs are displayed under FB_Pool in the Project view and displayed in the Selection window.


Navigation
Project
Parameter
Intelligent Function Module
Global Device Comment
Global Label
Program Setting
POU
Program
MAIN
Program
Local Label
M+Q64AD_ErrorOperation Error operation FB
M+Q64AD_SetADConversion AD cnv enable/disable setting FB
M+Q64AD_ReadADVal AD conversion data read FB
M+Q64AD_SetAverage Averaging process setting FB
M+Q64AD_SetGainVal Gain setting FB
M+Q64AD_SetOffsetVal Offset setting FB
M+Q64AD_ReadAllADVal AD cnv data read (all CHs) FB
M+Q64AD_RequestSetting Operating condition set req FB
Local Device Comment
Device Memory
Device Initial Value

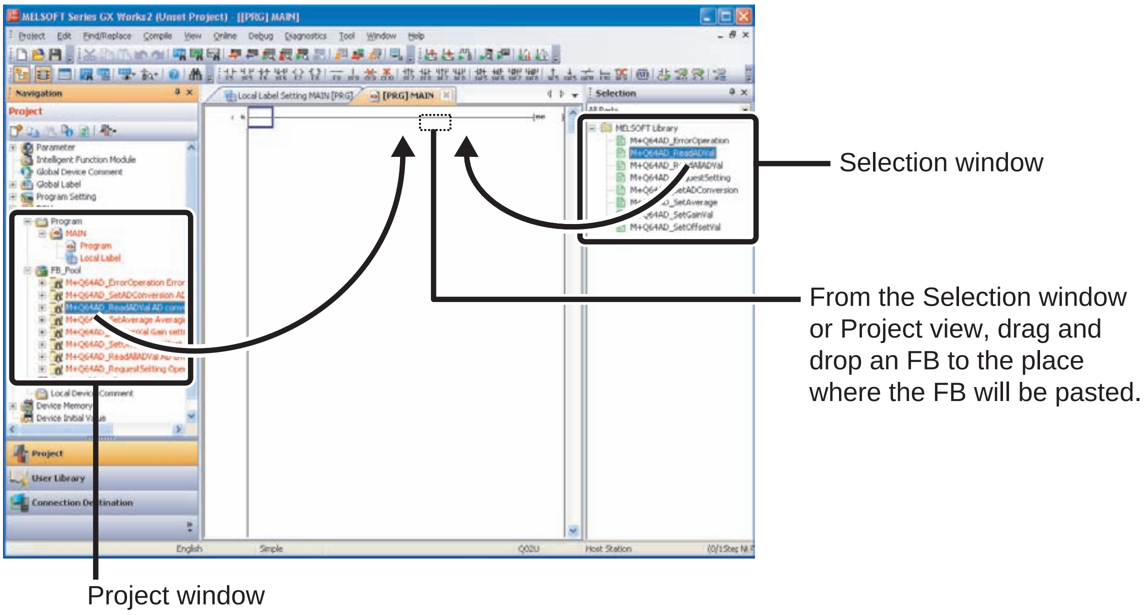


Appendix 9.2.4 Pasting FBs

Drag and drop FBs to be pasted to the program window from the Project view or Selection window. (Drag and drop from the Project view is possible from GX Works2 1.24A or later.)

Operating Procedure

1) Paste "M+Q64AD_ReadADVal" to the program window.



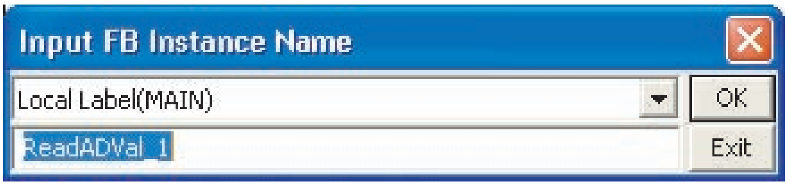
Selection window

From the Selection window or Project view, drag and drop an FB to the place where the FB will be pasted.

Project window

⇩

2) The Input FB Instance Name dialog box is displayed.



Local Label(MAIN)

ReadADVal_1

For details of settings, refer to the next page.

Appendix 9.2.5 Setting names of the pasted FBs

When an FB library is pasted to the program window, a dialog to input a name of the pasted FB (FB instance name) is displayed.

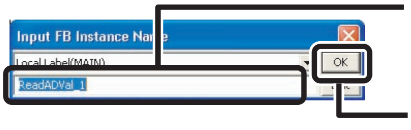
Instance name is a name to distinguish the FB.

A temporary name is automatically set to the instance name. To use the name as it is, close the dialog by clicking OK.
Make sure that the same name does not exist in the same program when changing the name.

In this section, the default is used.

Operating Procedure

1) Enter the FB instance name ("ReadADVal_1" in the example) and click the OK button.

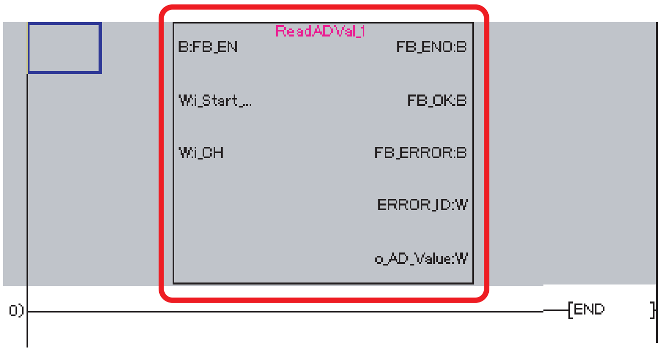


ReadADVal_1

Enter the FB instance name.

Click

2) The FB is pasted to the program window.

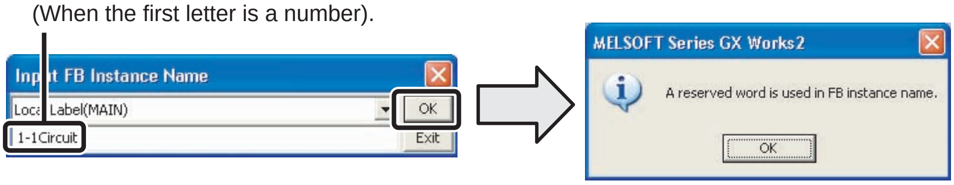


POINT

When entering an instance name, note the following points.
- Case-sensitive
- A number cannot be set for the first letter.
- The maximum number of characters for an instance name is 16.

An error occurs if OK is clicked with the following setting.

(When the first letter is a number).

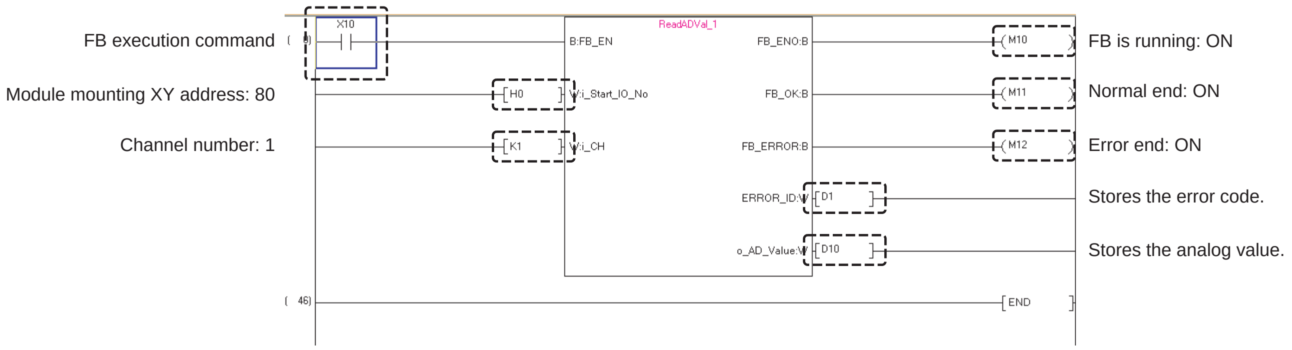


1-1Circuit

A reserved word is used in FB instance name.

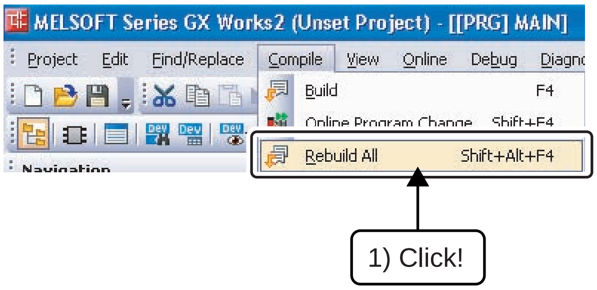
Appendix 9.2.6 Creating input and output ladders

Create the input ladder section and the output ladder section of the pasted FB to complete the program.
Refer to the following figure and enter the information.

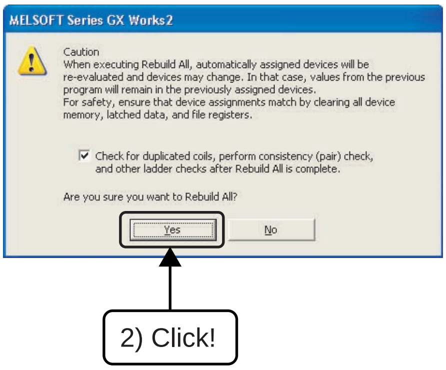


Appendix 9.2.7 Performing conversion/compilation

Conversion/compilation is required to execute the completed program.
The following explains how to convert/compile all programs.


MELSOFT Series GX Works2 (Unset Project) - [[PRG] MAIN]
Project Edit Find/Replace Compile View Online Debug Diagno
Build F4
Online Program Change Shift+F4
Rebuild All Shift+Alt+F4
Navigation
1) Click!

1) Click [Compile] → [Rebuild All].



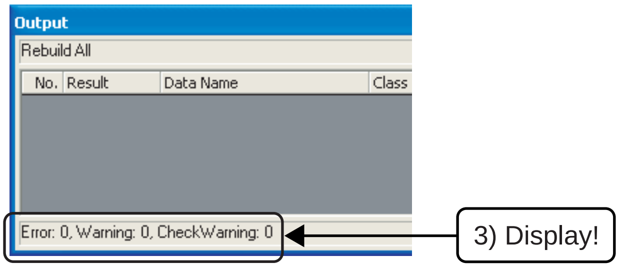
MELSOFT Series GX Works2

Caution
When executing Rebuild All, automatically assigned devices will be re-evaluated and devices may change. In that case, values from the previous program will remain in the previously assigned devices.
For safety, ensure that device assignments match by clearing all device memory, latched data, and file registers.

☑ Check for duplicated coils, perform consistency (pair) check, and other ladder checks after Rebuild All is complete.

Are you sure you want to Rebuild All?

2) The message on the left is displayed. Click the Yes button.



Error: 0, Warning: 0, CheckWarning: 0

3) The compilation result is displayed in the Output window.

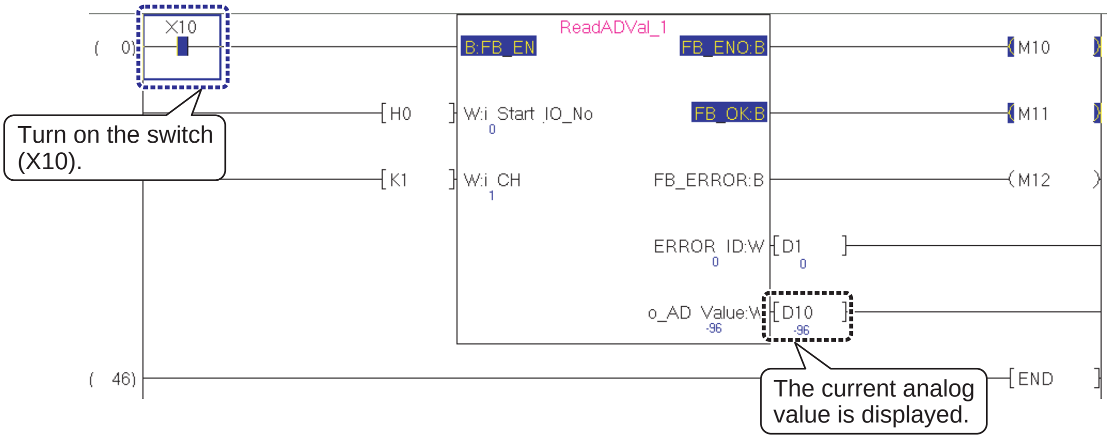
Appendix 9.2.8 Writing sequence programs

For the procedure to write sequence programs, refer to section 2.7 (1) "Writing data to the CPU".

Appendix 9.2.9 Operation check

For the procedure to check the operation of the created program, refer to section 2.8 Monitoring Ladder Program Status.

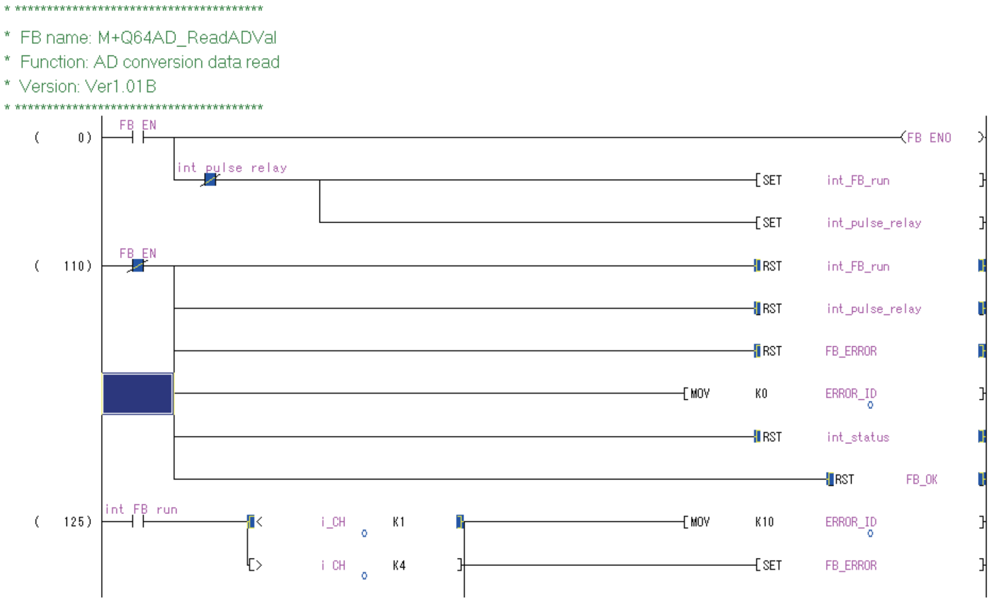
Turn on the switch (X2) and confirm that the analog value is read.



Turn on the switch (X10).

The current analog value is displayed.

Double-clicking the FB in the sequence program on the screen enables monitoring of the sequence program status in the FB.



```
* ****************************************
*  FB name: M+Q64AD_ReadADVal
*  Function: AD conversion data read
*  Version: Ver1.01B
* ****************************************
```

Mitsubishi Programmable Controller Training Manual
Q-series basic course (for GX Works2)

MODEL	SCHOOL-BASIC-GXW2-E
MODEL CODE	13JW55
SH(NA)-081123ENG-A(1210)MEE	

MITSUBISHI ELECTRIC CORPORATION

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

When exported from Japan, this manual does not require application to the Ministry of Economy, Trade and Industry for service transaction permission.

Specifications subject to change without notice.