MITSUBISHI

Mitsubishi Programmable Controller

Training Manual

Q-series advanced course (for GX Works2)



Changes for the Better



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.

- 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-081124ENG-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

NTRODUCTION

CHAPTER 1 OVERVIEW OF QCPU

2- 1 to 2-20

1-1 to 1-6

CHAPTER 2 SYSTEM CONFIGURATION

2.1 Ba	sic System Configuration 2-1
2.1.1	Device configuration
2.1.2	Precautions for system configuration2-4
2.2 Co	nnection with GX Works2
2.2.1	Interface and connection channel 2- 6
	Access range from GX Works2
2.3 Na	me and Appearance of CPU 2-10
2.4 Me	mory System Configuration 2-13
2.4.1	Universal model QCPU module memory configuration 2-13
2.4.2	Memory card application2-15
2.4.3	Handling the memory card 2-16

CHAPTER 3 PERFORMANCE SPECIFICATIONS

3.1	Performance Specifications	
3.2	Device	

CHAPTER 4 BASIC KNOWLEDGE REQUIRED FOR OPERATING GX Works2 4-1 to 4-6

4.1	Screen Configuration in GX Works2 4-1
4.2	Basic Operations of Dialog Box 4- 4
4.3	Ladder Program Creation Method 4- 5

CHAPTER 5 GX Works2 BASIC OPERATIONS (PART 1: SINGLE PROGRAM) 5- 1 to 5-26

5.1 Sys	stem Configuration of Demonstration Machine5-1
5.2 Ba	sic Operation 1 (Operation Before Creating Ladder Programs)
5.2.1	Starting up GX Works2
5.2.2	Selecting programmable controller type and project type (creating a new project)
5.2.3	Creating a program5-5
	Saving a project
5.2.5	Saving a project with another name 5-11
5.3 Bas	sic Operation 2 (Preparation for CPU Operation)5-12
5.3.1	Preparations for starting up CPU
5.4 Bas	sic Operation 3 (Writing Data to Programmable Controller, Monitoring, Modifying Program) 5-19
5.4.1	Writing data to the CPU
5.4.2	Reading data from the CPU
5.4.3	Monitoring 5-22
5.4.4	Modifying a program

3- 1 to 3-16

CHAPTER	R 6 FILE-BASED MANAGEMENT AND PROGRAM EXECUTION MANAGEMENT	6-1 to 6-14
6.1 File	e-Based Management	6- 1
6.1.1	Built-in memory and IC memory card	6-3
6.2 Pro	ogram Execution Management	6-7
	Description of program execution type	
6.2.2	Initial execution type program	6- 9
	Scan execution type program	
6.2.4	Standby type program	6-11
6.2.5	Fixed scan execution type program	6-12
6.2.6	Executing El	6-14

CHAPTER 7 GX Works2 BASIC OPERATIONS (PART 2: MULTIPLE PROGRAMS) 7-1 to 7-14

7.1 Mul	tiple Programs
7.1.1	Creating multiple programs7-1
7.1.2	Creating programs for control7-5
7.1.3	Setting parameters7-7
7.2 Moi	nitor7-10
7.2.1	Program list monitor7-10
7.2.2	Monitor function7-12

8-1 to 4-40

CHAPTER 8 FUNCTIONS OF QCPU

8.1 Maintenance and Debug Functions	
8.1.1 Self-diagnostic function	
8.1.2 System display function	
8.1.3 System protection function	
8.1.4 Password registration function	
8.2 Other Functions	
8.2.1 Constant scan function	
8.2.2 Latch function	
8.2.3 Remote operation function	
8.2.4 Service processing setting	
8.3 Comments Storage Function	
8.4 Appropriate Assignment of Device Points	
8.5 Using File Register	
8.5.1 Preparation for using file register	
8.5.2 Operation check	
8.6 Input Response Speed Change	

CHAPTER 9 PROGRAMMING INTELLIGENT FUNCTION MODULE

\sim	1-	\sim	-20
u -	τO	ч.	- 711

9.1 Co	mmunication with Intelligent Function Module9-1
9.1.1	Various settings with GX Works2
9.1.2	Communications by the intelligent function module device9-4
9.1.3	Communications with the intelligent function module dedicated instruction9-5
9.2 Inte	elligent Function Module System in Demonstration Machine
9.2.1	Creating an exercise program9-6
9.2.2	Switch setting, parameter setting, and auto refresh setting for the intelligent function module
9.2.3	Operation check and monitor test

CHAPTER 10 HOW TO USE MULTIPLE CPU SYSTEM

10- 1 to 10-34

10.1	Ove	erview of Multiple CPU System 10-1
10.2	Diff	erence from Single CPU System 10 -2
10.2	2.1	Mounting position of QCPU/motion CPU
10.2	2.2	I/O number assignment of the multiple CPU system 10-3
10.2	2.3	Communication between QCPUs and modules 10-3
10.2	2.4	Reset and operation for errors 10-4
10.3	Co	mmunication among each QCPU/Motion CPU in Multiple CPU System 10-6
10.3	3.1	CPU shared memory 10-6
10.3	3.2	Communication by auto refresh using CPU shared memory 10-9
10.3	3.3	Communication by auto refresh using multiple CPU high speed transmission area 10-11
10.3	3.4	Communications by the multiple CPU instruction and motion dedicated instruction 10-14
10.4 S	Start	ing up Multiple CPU System 10-15
10.4	4.1	Procedure for starting up the multiple CPU system 10-15
10.4	4.2	System configuration of the demonstration machine 10-17
10.4	4.3	Creating a program for CPU No. 1
10.4	4.4	Parameter setting for CPU No. 1
10.4	4.5	Creating a program for CPU No. 2
10.4	4.6	Parameter setting for CPU No. 2
10.4	4.7	Writing data to the CPU 10-31
10.4	4.8	Operation check 10-33

APPENDIX

Appendix 1 Instruction Tables App 1
Appendix 1.1 Application instructionApp 2
Appendix 1.2 QCPU Instructions App22
Appendix 2 How to Create Ladder Programs with GX Works2 App23
Appendix 3 Offset/Gain Setting App27
Appendix 3.1 Offset/gain setting with GX Works2App27
Appendix 3.2 Offset value and gain value App30
Appendix 4 Specifications of the A/D and D/A Converter ModulesApp34
Appendix 4.1 A/D converter moduleApp34
Appendix 4.2 D/A converter moduleApp38
Appendix 5 Comparison of Timers and Counters App42
Appendix 5.1 Comparison of timers and counters App42
Appendix 5.2 Comparison of counters App43
Appendix 6 Setting device initial values App44
Appendix 6.1 Setting device memories App46
Appendix 6.2 Specifying file names for device initial value App49
Appendix 6.3 Checking the operation of device initial values App50
Appendix 7 Inline Structured Text App51
Appendix 7.1 Editing inline structured text App51
Appendix 7.2 Precautions on using the inline structured text App54
Appendix 8 Battery App55
Appendix 9 Real Number (Floating-point data)App56

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 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)
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) Digital-Analog Converter Module User's Manual SH(NA)-080054
(9) Before Using the Product BCN-P5781
(10) Analog-Digital Converter Module User's Manual SH(NA)-080055
(11) MELSEC-Q/L Programming Manual (Structured Text) SH(NA)-080366

 MEMO

CHAPTER 1 OVERVIEW OF QCPU

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

QCPU has the following features.

(1) Large number of I/O points can be controlled

The Q-Series CPU module supports the following number of actual I/O points which are accessible to the I/O modules mounted on the base unit.

- (a) Basic model QCPU
 - Q00JCPU: 256 points (X/Y0 to FF)
 - Q00CPU, Q01CPU: 1024 points (X/Y0 to 3FF)
 Up to 2048 points (X/Y0 to 7FF) are supported as the I/O device points available for refreshing the remote I/O of CC-Link and link I/O (LX, LY) of
 - the MELSECNET/H.
- (b) High Performance model QCPU One module supports 4096 points (X/Y0 to FFF). Up to 8192 points (X/Y0 to 1FFF) are supported as the I/O device points available for the remote I/O stations in the MELSECNET/H remote I/O network and CC-Link data link.
- (c) Process CPU and redundant CPU
 One module supports 4096 points (X/Y0 to FFF).
 Up to 8192 points (X/Y0 to 1FFF) are supported as the I/O device points available for the remote I/O stations in the MELSECNET/H remote I/O network and CC-Link data link.
- (d) Universal model QCPU
 - Q00UJCPU: 256 points (X/Y0 to FF)
 - Q00UCPU, Q01UCPU: 1024 points (X/Y0 to 3FF)
 - Q02UCPU: 2048 points (X/Y0 to 7FF)
 - Q03UD(E)CPU, Q04UD(E)HCPU, Q06UD(E)HCPU, Q10UD(E)HCPU, Q13UD(E)HCPU, Q20UD(E)HCPU, Q26UD(E)HCPU: 4096 points (X/Y0 to FFF)

Up to 8192 points (X/Y0 to 1FFF) are supported as the I/O device points available for the remote I/O stations in the MELSECNET/H remote I/O network and CC-Link data link.

(2) Lineup corresponding to the program capacity

The following table lists the lineup of CPU available for various program capacity.

CPU module type		Program capacity
Basic model QCPU	Q00(J)CPU	8K steps
Dasic model QCF U	Q01CPU	14K steps
	Q02(H)CPU	28K steps
High Performance	Q06HCPU	60K steps
model QCPU	Q12HCPU	124K steps
	Q25HCPU	252K steps
	Q02PHCPU	28K steps
Process CPU	Q06PHCPU	60K steps
FIDCESS CFU	Q12PHCPU	124K steps
	Q25PHCPU	252K steps
Redundant CPU	Q12PRHCPU	124K steps
Reguliuani CPU	Q25PRHCPU	252K steps
	Q00U(J)CPU	10K steps
	Q01UCPU	15K steps
	Q02UCPU	20K steps
	Q03UD(E)CPU	30K steps
Universal model	Q04UD(E)HCPU	40K steps
QCPU	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

(3) High speed processing

High speed processing has been achieved. (Example: LD instruction)

CPU module type		LD instruction processing speed	
	Q00JCPU	200ns	
Basic model QCPU	Q00CPU	160ns	
	Q01CPU	100ns	
High Performance	Q02CPU	79ns	
model QCPU	Q02HCPU, Q06HCPU, Q12HCPU,		
	Q25HCPU		
Process CPU	Q02PHCPU, Q06PHCPU,	34ns	
Process CPU	Q12PHCPU, Q25PHCPU		
Redundant CPU	Q12PRHCPU, Q25PRHCPU		
	Q00UJCPU	120ns	
	Q00UCPU	80ns	
	Q01UCPU	60ns	
Universal model	Q02UCPU	40ns	
QCPU	Q03UD(E)CPU	20ns	
	Q04UD(E)HCPU, Q06UD(E)HCPU,		
	Q10UD(E)HCPU, Q13UD(E)HCPU,	9.5ns	
	Q20UD(E)HCPU, Q26UD(E)HCPU		

The high-speed system bus for the MELSEC-Q series base unit has achieved faster access to an intelligent function module and link refresh with a network module.

(a) Basic model QCPU

MELSECNET/H link refreshing: 2.2ms/2K words*1

*1: The Q01CPU is used without SB and SW, and the MELSECNET/H network module is mounted on the main base unit.

(b) High Performance model QCPU, Process CPU, Redundant CPU or Universal model QCPU

Access to the intelligent function module: 20μ s/word (approximately 7 times^{*2})

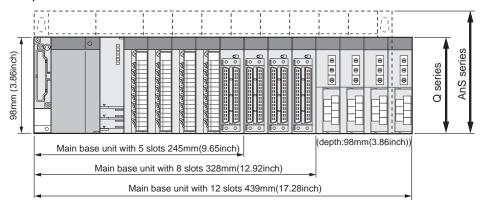
MELSECNET/H link refreshing: 4.6ms/8K words (approximately 4.3 times^{*2})

- *2: These are the values resulted from the following comparison:
 - Comparing Q02HCPU with Q2ASHCPU-S1
 - Comparing Q25PHCPU with Q4ARCPU
 - Comparing Q25PRHCPU with Q4ARCPU
- Increased debugging efficiency through high-speed communication with GX Works2

High-speed communications at maximum 115.2Kbps are available by RS-232 and the time required for writing and reading of programs and monitoring are reduced. Also, the communication time efficiency of debugging is increased. In addition, High Performance model QCPUs (except for the Q02CPU), Process CPUs, Redundant CPUs, and Universal model QCPUs support USB, which enables high-speed communications at 12Mbps.

- (5) AnS/A series I/O modules and special function modules are available The AnS/A series compatible extension base units (QA1S6□B, QA6□B, and QA6ADP+A5□B/A6□B) enable High Performance model QCPU to use the AnS/A series I/O modules and special function modules.
 - * The extension base unit for the A series cannot be used for the Universal model QCPU.
- (6) Saved space by downsizing

The installation space for the Q series is reduced by approximately 60% compared with the AnS series.



(7) Connection of up to 7 extension base units

Up to seven extension base units can be connected to the Q series CPU module.

The whole extension cable length is 13.2m, which enables flexible layout of base units.

(8) Memory extension by memory $card^{*3}$

The QCPU equips the memory card installing connector so that a memory card with the capacity of up to 32M byte can be connected. (The 32M-byte memory card can be connected only when an ATA card is used.) Installing large-capacity memory cards enables large-capacity files to be managed, which allows for the comment setting to all data devices and saving old programs in a memory as correction data.

*3: The Basic model QCPU, Q00(J)CPU, and Q01UCPU do not support memory cards.

POINT

For the High Performance model QCPU, available file register points differs depending on the function version and serial number. For details, refer to the QCPU User's Manual Hardware Design, Maintenance and Inspection.

- (9) Automatic writing to standard ROM^{*4,*5} Parameters and programs of the memory cards can be written to the standard ROM of the CPU module without GX Works2.
 - *4: The Basic model QCPU does not support the following functions. Automatic writing to standard ROM
 - *5: The Universal model QCPU does not support the following function. Automatic writing to standard ROM
- (10) Forced on and off of external I/O^{*6}

Even when the CPU module is running, forced on and off of external input and output is available with GX Works2 regardless of the program execution status. Also, the wiring and operation tests can be conducted without stopping the CPU module by forcibly turning on or off the I/O.

- *6: The Basic model QCPU does not support the following functions. Forced on and off of external I/O
- (11) Remote password setting

When the built-in Ethernet port QCPU, Ethernet module, or serial communication module is externally accessed, an access to the CPU module can be controlled by the remote password.

- (12) Remote I/O network of MELSECNET/H^{*7}
 - A MELSECNET/H remote I/O system can be configured by installing a MELSECNET/H remote master station.
 - *7: The Basic model QCPU does not support the following functions. Remote I/O network of MELSECNET/H

POINT

- The remote password can be set when the built-in Ethernet port QCPU, Ethernet module, or serial communication module of function version B or later is used.
- The MELSECNET/H remote I/O network can be implemented when the MELSECNET/H network module of function version B or later is used.
- (13) Supporting the multiple CPU system The Q series CPU module supports the multiple CPU system. Multiple CPU system can be configured in combination with CPU modules, motion CPUs, PC CPU modules, and C Controller module. For details of the multiple CPU system, refer to the QCPU User's Manual (Multiple CPU System).
- (14) Supporting the redundant power supply systemThe redundant power supply system can be configured with a redundant base unit and redundant power supply modules.The system can continue operation even when one of the power supply modules fails, since the other supplies the power.
- (15) Direct connection to Ethernet^{*8}

The Built-in Ethernet port QCPU module allows for direct connections to Ethernet.

For details of the functions, refer to the QnUCPU User's Manual Communication via Built-in Ethernet Port.

*8: Applicable only to the built-in Ethernet port QCPU.

MEMO)
	J

CHAPTER 2 SYSTEM CONFIGURATION

- 2.1 Basic System Configuration
- 2.1.1 Device configuration

The following figure shows an actual programmable controller configuration.

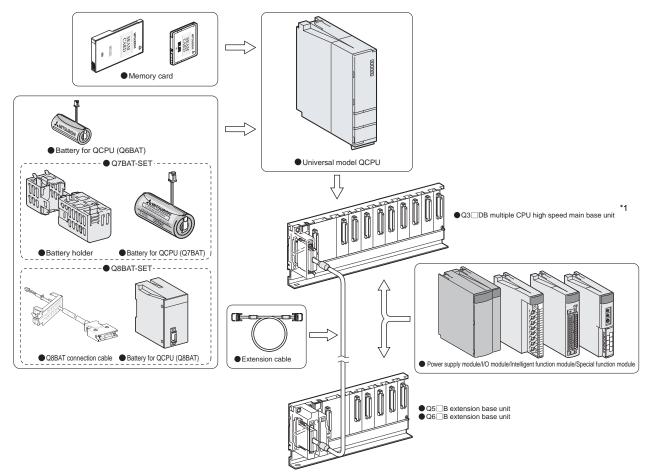
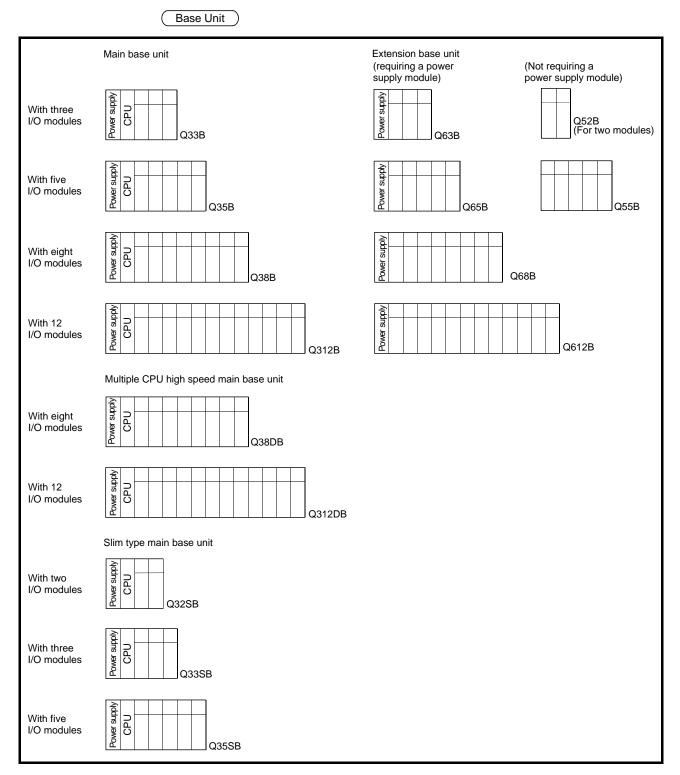


Figure 2.1 Universal model QCPU module configuration (When Q3 DB is used)

*1: The following bases are also available for the Universal model QCPU:

- Q3□B type main base unit
- Q3 RB type redundant power main base unit
- Q3DSB type slim type main 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
Q64PN	100V to 240VAC	5VDC 8.5A
Q61P-D	100V to 240VAC	5VDC 6A
Q61SP	100V to 240VAC	5VDC 2A

(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	
Q04UD(E)HCPU	40K steps		
Q06UD(E)HCPU	60K steps		
Q10UD(E)HCPU	100K steps	9.5ns	4096 points
Q13UD(E)HCPU	130K steps		
Q20UD(E)HCPU	200K steps		
Q26UD(E)HCPU	260K steps		

(I/O Module)

For	I/O points mat	8 points	16 points	32 points	64 points
	120VAC	-	0	-	-
	240VAC	0	-	-	-
odule	24VDC (positive common)	-	0	0	0
Input module	24VDC (high-speed input)	0	-	-	-
<u> </u>	24VDC (negative common)	-	0	0	-
	5/12VDC	-	0	0	0
	Contact output	-	0	-	-
dule	Independent contact output	0	-	-	-
ũ	Triac output	-	0	-	-
Output module	Transistor output (sink)	0	0	0	0
0	Transistor output (source)	-	0	0	-
	I/O mixed	0	-	0	-

2.1.2 Precautions for system configuration

This section explains restrictions for configuring the system with the Q-series CPU module.

- (1) Number of mountable modules
 - (a) The number of mountable modules and supported functions are restricted depending on the module type.

Product name	Model name	Maximum number of modules/units per system
CC-Link IE controller network module ^{*1}	 QJ71GP21-SX QJ71GP21S-SX 	
MELSECNET/H network module	 QJ71LP21 QJ71BR11 QJ71LP21-25 QJ71LP21S-25 QJ71LP21G QJ71NT11B 	Up to 4 modules ^{*2, *3}
Q series Ethernet interface module	 QJ71E71 QJ71E71-B2 QJ71E71-B5 QJ71E71-100 	Up to 4 modules ^{*3}
Q series CC-Link system master/local module	QJ61BT11QJ61BT11N	No restriction ^{*4, *5}
Interrupt module	• QI60	Only 1 module ^{*6}
GOT (Graphic Operation Terminal)	GOT1000 Series (for bus connection only) ^{*7}	Up to 5 units

[When the Universal model QCPU is used]

- *1: Only the CC-Link IE controller network module with the serial number (first five digits) of "09042" or later can be used.
- *2: The number is a total of the CC-Link IE controller network module and MELSECNET/H network module.
- *3: One module is mountable to the one system of the Q00UJCPU, Q00UCPU, and the Q01UCPU, and two modules for the Q02UCPU.
- *4: Available in modules with function version B or later.
- *5: One CPU module can control the following number of modules by setting CC-Link network parameters in GX Works2.
 - Q00UJCPU, Q00UCPU, Q01UCPU: up to 2 modules
 - Q02UCPU: up to 4 modules
 - Q03UD(E)CPU, Q04UD(E)HCPU, Q06UD(E)HCPU, Q10UD(E)HCPU, Q13UD(E)HCPU, Q20UD(E)HCPU, Q26UD(E)HCPU: up to 8 modules

There is no restriction on the number of modules when the parameters are set with the CC-Link dedicated instructions.

For the CC-Link system master/local modules whose parameters can be set by the dedicated instructions, refer to the CC-Link System Master/Local Module User's Manual.

- *6: This number indicates the number of interrupt modules with no interrupt pointer setting. There is no restriction on the number of modules for the interrupt modules with the interrupt pointer setting.
- *7: For the available GOT models, refer to the GOT1000 Series Connection Manual.

 (b) For the GOTs, the GOT1000 series are available (however, Q-mode-compatible operating system and communication driver must be installed).

The Q series bus connection interface module is required for the bus connection.

The GOT800 series, A77GOT, and A64GOT cannot be used. The GOT900 series do not support the Universal model QCPU.

2.2 Connection with GX Works2

2.2.1 Interface and connection channel

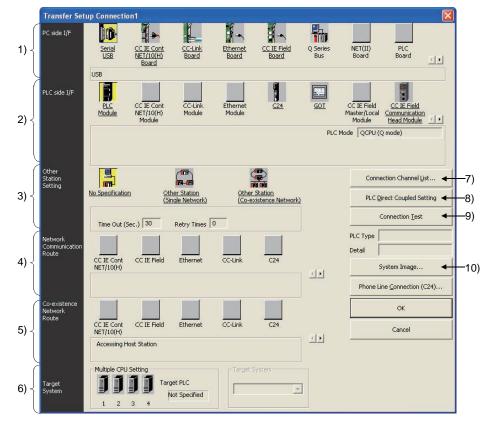
In the Q series, flexible and wide methods are available for connecting the CPU and GX Works2.

For details, refer to the GX Works2 Operating Manual.

GX Works2 has the following two items about the "connection destination".

- Specification of I/F
 PC side I/F or PLC side I/F
- (2) Other station setting and network route

Other Station Setting, Network Communication Route, Co-existence Network Route



The following explains each item of the Transfer Setup screen.

1) PC side I/F

Select the type of the interface on the personal computer side. Double-click each interface to set the details.

2) PLC side I/F

Select the module on the programmable controller side to be connected with the peripheral device.

Double-click each module to set the details.

3) Other Station Setting

Specify the host station or other station.

Double-click each icon to set the details.

- No Specification Select this to access the programmable controller CPU which is directly connected to a personal computer.
- Other Station (Single Network) Select this to access the programmable controller CPU on other station via only one type of network (including a multi-tier system) such as CC-Link, MELSECNET/10(H), CC-Link IE controller network, Q series C24 module and Ethernet.

Since Ethernet is recognized as equivalent to CC-Link IE controller network and MELSECNET/10(H), select "Single Network" for a mixed system in which Ethernet, CC-Link IE controller network, and MELSECNET/10(H) are configured.

- Other Station (Co-existence Network) Select this to access the programmable controller CPU on other station via two types of network. This means the system which consists of two different networks, such as from MELSECNET/10(H) to CC-Link module or from Q series C24 module to MELSECNET/10(H).
- 4) Network Communication Route

Select the network type, network number, station number, and the start I/O number of the network that is routed at an access to the programmable controller CPU on other station. The setting items differ according to the selected network type.

- Co-existence Network Route Select the network type, network number, station number, and the start I/O number of the network to be accessed. The setting items differ according to the selected network type.
- Target System
 In the multiple CPU system, specify the CPU number to be accessed.
- Connection Channel List button
 Displays a list of the types of the connection destination.
 The connection route can be selected from the list.
- 8) PLC Direct Coupled Setting button

This function is useful to change the station specification from "Other Station" to "No Specification".

9) Connection Test button

Tests if the target programmable controller CPU set on the Transfer Setup screen can be accessed properly.

If the test is successful, the model of the target programmable controller CPU module is displayed in the PLC Type column.

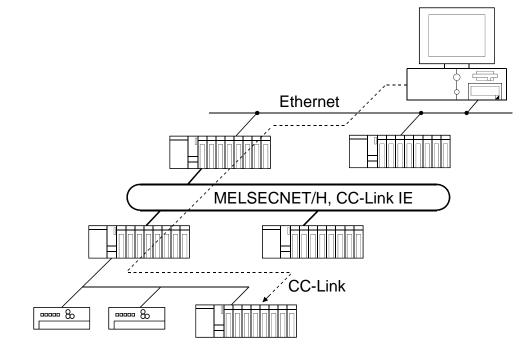
10) System Image button

Displays the connection route in an illustration.

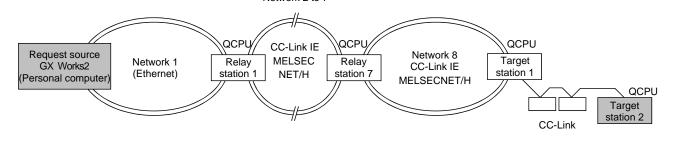
2.2.2 Access range from GX Works2

The seamless communication is established among CC-Link IE, Ethernet, MELSECNET/H, and CC-Link of the Q series.

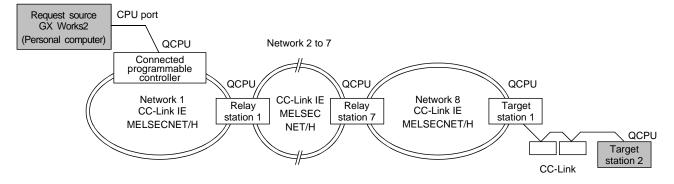
GX Works2 can access a programmable controller via various networks.



 Access example via Ethernet, CC-Link IE, MELSECNET/H, and CC-Link The request source GX Works2 can access up to two target stations. Network 2 to 7

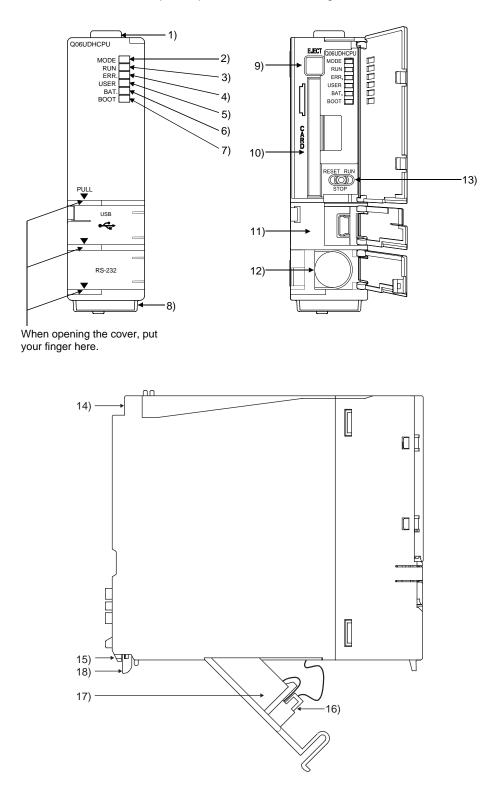


(2) Access example via CC-Link IE, MELSECNET/H, and CC-Link The request source GX Works2 can access up to two target stations.



2.3 Name and Appearance of CPU

This section explains part names and setting of the module.

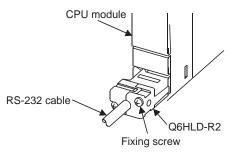


No.	Name	Application
1)	Module fixing hook	Hook used to fix the module to the base unit (Single-operation installation)
2)	MODE LED	Indicates the mode of the CPU module On : Q mode Flash : Device test with conditions is in process. Forced on and off function for external I/O is in process. CPU module change function with memory card is in process.
3)	RUN LED	 Indicates the operation status of the CPU module On : During operation with the RUN/STOP/RESET switch set to "RUN" Off : During stop with the RUN/STOP/RESET switch set to "STOP" When an error which stops operation is detected Flash : Parameters or programs are written when the RUN/STOP/RESET switch is set to "STOP", then the RUN/STOP/RESET switch is set from "STOP" to "RUN" To turn on the RUN LED after writing the program, perform the following operations. Set the RUN/STOP/RESET switch "RUN" → "STOP" → "RUN". Reset the CPU module with the RUN/STOP/RESET switch. Power on the programmable controller again. To turn on the RUN LED after writing the parameters, perform the following operations. Reset the CPU module with the RUN/STOP/RESET switch. Power on the programmable controller again. To turn on the RUN LED after writing the parameters, perform the following operations. Reset the CPU module with the RUN/STOP/RESET switch. Power on the programmable controller again. To turn on the RUN LED after writing the parameters, perform the following operations. Reset the CPU module with the RUN/STOP/RESET switch. Power on the programmable controller again. To turn on the RUN/STOP/RESET switch is set to "RUN" → "STOP" → "RUN" after the parameters are changed, network parameters and intelligent function module parameters are not updated.)
4)	ERR. LED	On : When a self-diagnosis error which does not stop the operation except a battery error is detected (When operation is set to be continued at an error detection in the parameter setting) Off : Normal Flash : When an error which stops operation is detected When the reset operation becomes valid with the RUN/STOP/RESET switch
5)	USER LED	On : When the annunciator is (F) turned on Off : Normal
6)	BAT. LED	On (yellow) : When a battery error occurs due to a battery voltage drop of the memory card Flash (yellow) : When a battery error occurs due to a voltage drop of the CPU module battery On (green) : Turns on for five seconds when the restoration of the data backed up to the standard ROM by the latch data backup is completed. Flash (green) : Flashes when the backup of the data to the standard ROM by the latch data backup is completed. Off : Normal
7)	BOOT LED	On : When the boot operation is started Off : When the boot operation is not being performed
8)	Serial number display	Displays the serial number printed on the rating plate.
9)	Memory card EJECT button	Used to eject the memory card from the CPU module
10)	Memory card installing connector	Connector used to install a memory card to the CPU module

No.	Name	Application
11)	USB connector ^{*1}	Connector for connection with a USB-compatible peripheral device (Connector type miniB) Can be connected with a USB-dedicated cable.
12)	RS-232 connector ^{*1}	Connector for connection with a peripheral device Can be connected with a RS-232 connection cable (QC30R2).
13)	RUN/STOP/RESET switch ^{*2}	 RUN : Executes sequence program operation. STOP : Stops sequence program operation. RESET : Executes hardware reset, operation error reset, and operation initialization etc.
14)	Module fixing screw hole	Hole for the fixing screw to the base unit (M3 \times 12 screw)
15)	Module fixing projection	Projection used to secure the module to the base unit
16)	Battery connector pin	Pins used to connect battery lead wires (Lead wires are disconnected from the connector at the shipping to prevent the battery from consuming.)
17)	Battery	Backup battery for the standard RAM and back-up power function
18)	Module mounting lever	Lever used to mount the module to the base unit

*1: When connecting a cable to the RS-232 connector or USB connector at all times, clamp the cable to prevent a poor connection, moving, and disconnection by unintentional pulling.

The Q6HLD-R2 type connector disconnection prevention holder is provided as a clamp for the RS-232 connector.



*2: Operate the RUN/STOP/RESET switch with a fingertip. To prevent the switch from damage, do not use any tool such as screw driver.

2.4 Memory System Configuration

2.4.1 Universal model QCPU module memory configuration

Program memory RAM (program cache memory) Parameter Program Parameter Program Device comment Device initial value Device initial value Device comment Standard File register Local device ROM Contraction of the second Sampling Parameter Program **4**111 trace file Device comment Device initial value Memory card ROM *1 Programmable controller user data Parameter Program CPU module Device initial value Device comment Storage file used in latch data backup function File register File used in SP.DEVST/S.DEVLD function Standard RAM^{*2} File register Local device Sampling Module error trace file collection file

The memory of universal model QCPU consists of the following block configurations.

*1: A memory card cannot be used for Q00UJCPU, Q00UCPU, and Q01UCPU. *2: Q00UJCPU has no standard RAM.

Program memory:	A memory for storing programs and parameters for 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):	mory card (RAM): A card for storing the file register, local device, device initial value, sampling trace file, and device comments 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

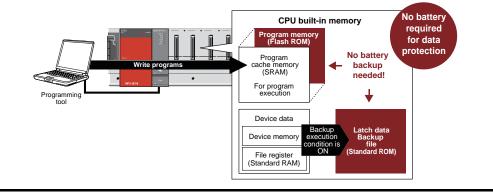
Secure backup by long-term storage

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.



2.4.2 Memory card application

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.

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

(1) SRAM card

File registers in the SRAM card can be written or read by the sequence program.

The SRAM card is used when:

- the number of file registers exceeds the standard RAM capacity, or
- the sampling trace function is used.

When file registers are stored to the SRAM card, they can be written or read by the sequence program up to 4086K points.

(2) Flash card

Write data with GX Works2 and read it by the sequence program. (Data can only be read by the sequence program.) Use the Flash card when changing the data is unnecessary. File registers can be stored up to 2039K points.

(3) ATA card

An ATA card is used for programmable controller user data (general-purpose data).

Programmable controller user data of an ATA card can be accessed by the file access instruction (such as the SP.FWRITE instruction) in a sequence program through a CSV format or binary format.

2.4.3 Handling the memory card

The specifications of the memory card which are available for the QCPU module conform to those of the JEIDA/PCMCIA small programmable controller card. Only one memory card can be installed to the QCPU.

- (1) Memory card specifications
 - (a) SRAM card

Item		Туре			
		Q2MEM-1MBS	Q2MEM-2MBS	Q3MEM-4MBS	Q3MEM-8MBS
Memory capacit after format	ty	1011.5K byte	2034K byte	4078K byte	8172K byte
Storable number of files		255	287	319	
Number of insertions and removals		5000 times			
External	Н	45mm		74mm	
dimensions D		42.8mm			
		3.3mm		8.1mm	
Weight		15g 30g 3		31g	

(b) Flash card

Item		Туре		
		Q2MEM-2MBF	Q2MEM-4MBF	
Memory capacity		2035K byte	4079K byte	
Storable number files	er of	288		
Number of insertions and removals		5000 times		
Number of writing	Number of writings 100000 times) times	
External	Н	45mm		
dimensions	W	42.8mm		
	D	3.3mm		
Weight		15g		

(c) ATA card

ltem		Туре		
nem		Q2MEM-8MBA	Q2MEM-16MBA ^{*2}	Q2MEM-32MBA
Memory capacit after format	ty	7982K byte ^{*1}	15982K byte ^{*1}	31854K byte
Storable numbe files	er of	512 ^{*2}		
Number of insertions and removals		5000 times		
Number of writin	ngs	1000000 times		
External	Н	45mm		
dimensions	W	42.8mm		
	D	3.3mm		
Weight		15g		

*1: The capacity of the ATA cards with the manufacturer control number E or earlier after formatting is as follows.

Manufacturer control number E: Q2MEM-8MBA: 7948K byte, Q2MEM-16MBA: 15948K byte

Manufacturer control number E or earlier: Q2MEM-8MBA: 7940k byte, Q2MEM-16MBA: 15932K byte

*2: Up to 511 files can be stored in the Universal model QCPU.

(2) When using the memory card in the purchased status

(a) Install the enclosed battery.

SRAM card battery

Type Item	Q2MEM-BAT	Q3MEM-BAT	
Classification	Graphite fluoride lithium primary battery	Manganese dioxide lithium primary battery	
Initial voltage (V)	3.0	3.0	
Nominal current (mAh)	48	550	
Battery life when stored	Actually 5 years (room temperature)		
Battery life when used	Refer to the QCPU User's Manual Hardware Design, Maintenance and Inspection		
Application	Power failure backup for SRAM card For Q2MEM-1MBS/Q2MEM-2MBS	Power failure backup for SRAM card For Q3MEM-4MBS/Q3MEM-8MBS	

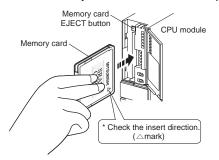
(b) Since the memory card is not formatted in the initial setting, use the card after formatting in GX Works2.

Formatting is unnecessary for Flash cards.

- (3) Installing and removing a memory card
 - (a) For Q2MEM type memory card
 - 1) To install the memory card

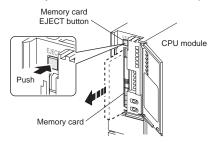
Pay attention to the direction of the memory card when installing it to the CPU module.

Insert the memory card securely into the connector until the projection of the memory card EJECT button appears.

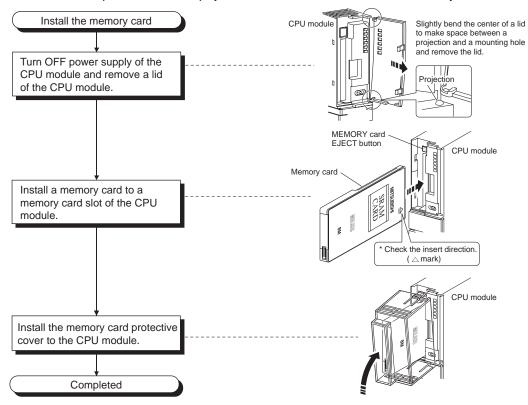


2) To remove the memory card

When removing the memory card from the CPU module, press the memory card EJECT button and pull out the memory card.

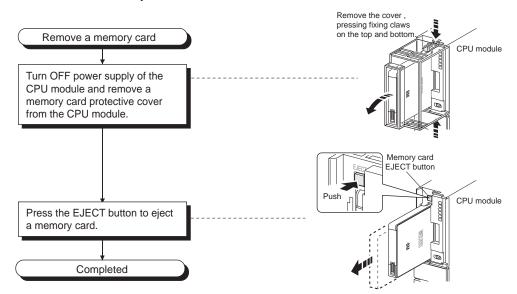


- (b) For Q3MEM type memory card
 - To install the memory card When installing a memory card to the CPU module, follow the following procedures and pay attention to the direction of the memory card.



2) To remove the memory card

When removing the memory card from the CPU module, remove the memory card protective cover and press the EJECT button and pull out the memory card.



(c) To remove the memory card while the power is on

When removing the memory card, confirm that special relays "SM604" and "SM605" are off.

- The memory card cannot be removed when "SM604" is on because the CPU module is using the card.
- Turn off "SM605" when it is on. When both "SM604" and "SM605" are off, remove the memory card according to the following procedure.
- 1) Turn on the special relay "SM609" with the sequence program or the device test of GX Works2.
- 2) Use the monitor function of GX Works2 to check that the special relay "SM600" is turned off.
- Remove the memory card.
 SM600 (Memory card can be used): The system is turned on when the memory card is ready to be used.
 SM604 (memory card is being used): The system is turned on when the CPU module is using the memory card.
 SM605 (memory card installation/removal prohibited): Turned on by the

user to disable a installation/removal of the memory card.

- (d) To install the memory card while the power is on
 - 1) Install the memory card.
 - 2) Use the monitor function of GX Works2 to check that the special relay "SM600" is turned on.

CHAPTER 3 PERFORMANCE SPECIFICATIONS

3.1 Performance Specifications

The following table lists the performance specifications of the Universal model QCPU.

(1) Q00UJCPU, Q00UCPU, Q01UCPU, Q02UCPU, Q03UD(E)CPU

				Uni	iversal model Q	CPU				
	Item		Q00UJCPU	Q00UCPU	Q01UCPU	Q02UCPU	Q03UDCPU			
			QUUUJCFU	QUUUCFU	QUIUCFU	QUZUCFU	Q03UDECPU			
Control meth	od			Stored p	orogram repeat	operation				
I/O control m	node			Refresh mode						
	Convenee			(Direct access I/O is available by specifying direct access I/O (DX□, DY□).) Relay symbol language, logic symbolic language, MELSAP3 (SFC),						
Program language	Sequence of language				tion block, and s					
		ntrol language			-					
Processing speed	LD X0		120ns	80ns	60ns	40ns	20ns			
(sequence instruction)	MOV D0 D	1	240ns	160ns	120ns	80ns	40ns			
Processing speed (redundant function)	(increased	execution time scan time)			-					
Constant sca (function for		ılar scan time)	0.5 to 2000	0.5 to 2000ms (setting available in 0.5ms unit) (setting by parameters)						
Program cap	pacity ^{*1 *2}		10K steps (40K byte)		15K steps (60K byte)	20K steps (80K byte)	30K steps (120K byte)			
	Program m (drive 0)	emory	40K byte 60K byte 80K byte				120K byte			
	Memory ca (drive 1)	rd (RAM)		-			the installed (8M byte max.)			
	Memory ca (drive 2)	rd (ROM)		-		memo (Flash card:	the installed bry card 4M byte max., 2M byte max.)			
Memory *1	Standard R (drive 3)	AM	-		128K byte		192K byte			
capacity '	Standard R (drive 4)	ROM	256K byte		512K byte		1024K byte			
	CPU		-		8K	byte	·			
	CPU shared memory*3 *3 memory*3 CPU high speed transmission area				-		32K byte			

*1: The size unit of the files stored in the memory area differs depending on the CPU module.

For details, refer to the QnUCPU User's Manual (Function Explanation, Program Fundamentals).

 *2: The maximum number of executable sequence steps is obtained by the following formula. (Program capacity) - (File header size (Default: 34 steps))
 For details of the program capacity and files, refer to the QnUCPU User's Manual (Function Explanation, Program Fundamentals).

*3: Data in the CPU shared memory is not latched.

Data in the CPU shared memory is cleared when the programmable controller is powered on or the CPU module is reset.

	ltem				Un	iversal model QC	CPU			
	lte	m		Q00UJCPU	Q00UCPU	Q01UCPU	Q02UCPU	Q03UDCPU Q03UDECPU		
		Program	memory		32	•	64	124		
		Memory ((RAM)	card		-			Q3MEM-8MBS (sed)		
Max. nun	nber of	Memory card	Flash card		-		2	88		
files store	he	(ROM)	ATA card	- 511						
	Standard RAM Standard ROM			-	3 files (each one of the following files: file register file, loca device file, and sampling trace file)					
					1:	28		256		
	am memo	ory			Μ	lax. 100000 time	s ^{*4}			
Number of the stand	lard ROM				Μ	lax. 100000 time	s ^{*5}			
program)	of usable	points or			8192	points (X/Y0 to	1FFF)			
Number of (number of actual I/C	of points a		e to the	256 points (X/Y0 to FF)		points to 3FF)	2048 points (X/Y0 to 7FF)	4096 points (X/Y0 to FFF)		
	Internal relay [M] ^{*6}			8192 points by default (M0 to 8191) (changeable)						
	Latch re	$ay[L]^{*6}$		8192 points by default (L0 to 8191) (changeable)						
	Link rela	ay [B] ^{*6}			8192 points by a	default (B0 to 1Fl	FF) (changeable)		
	Timer [T	-] ^{*6}		The low The measu (Low	h v- and high-spee urement units of -speed timer: 1 t	high-speed timer d timers are spe- the low- and high parameters. o 1000ms, 1ms u	Jeable) (sharing o s) cified by the instr n-speed timers a unit, 100ms by do s unit, 10ms by c	ructions. re set up by efault)		
Number of device points		ve timer [{	ST] ^{*6}	0 point by default (sharing of the low- and high-speed retentive timers) (changeable) The low- and high-speed retentive timers are specified by the instructions. The measurement units of the low- and high-speed retentive timers are set up by parameters. (Low-speed retentive timer: 1 to 1000ms, 1ms unit, 100ms by default) (High-speed retentive timer: 0.1 to 100ms, 0.1ms unit, 10ms by default)						
	Counter	[C] ^{*6}		Normal	counter, 1024 po	pints by default (C0 to 1023) (cha	ngeable)		
	Data reg	gister [D]	6	1	2288 points by a	default (D0 to 12	287) (changeable	e)		
		d data re		-		0 point by defa	ult (changeable)			
	Link reg	ister [W] [*]	6		8192 points by c	lefault (W0 to 1F	FF) (changeable	.)		
		ed link reg	jister [W]	-		0 point by defa	ult (changeable)			
		ator [F] ^{*6}			2048 points by	default (F0 to 204	47) (changeable))		
	Edge re	lay [V] ^{*6}			2048 points by	default (V0 to 204	47) (changeable))		
		ecial relay			2048 points by c	default (SB0 to 7	FF) (changeable)		
	[SW] ^{*6}	ecial regis	ilei		2048 points by d	lefault (SW0 to 7	FF) (changeable	.)		

*4: A single writing operation may not be counted as one.

The number of writing into the program memory can be checked with the special register (SD682 and SD683). *5: A single writing operation may not be counted as one.

The number of writing into the standard ROM can be checked with the special register (SD687 and SD688).

*6: The number of points can be changed within the setting range.

					Uni	versal model Q	CPU		
	Ite	m						Q03UDCPU	
				Q00UJCPU	Q00UCPU	Q01UCPU	Q02UCPU	Q03UDECPU	
						I.		32768 points	
								(R0 to 32767)	
			Standard		32768	3 points (R0 to 3	2767)	Max. 98304	
			RAM	-		6 points by bloc	-	points by	
					Max. 0000		it switching	block	
								switching	
			SRAM				Max 517120	points by block	
			card		_		units of 32768		
			(1M byte)		_		-	0 to 32767)	
			SRAM					points by block	
			card		_			units of 32768	
			(2M byte)		-		-	0 to 32767)	
		[R]	SRAM				points by block		
			card		_		units of 32768		
			(4M byte)		-	-	0 to 32767)		
			SRAM				points by block		
			card				units of 32768		
			(8M byte)		-	-	0 to 32767)		
			(ow byte)				points by block		
			Flash card		-				
			(2M byte)		-		switching in units of 32768 points (R0 to 32767)		
							Max. 2087936 points by bloc		
			Flash card					units of 32768	
Number	File	(4M byte)			_		-	0 to 32767)	
of device	register							98304 points	
points	*7							(ZR0 to	
points			Standard		65536 points (ZR0			98303)	
			RAM	-		switching not re		Block	
					Diookomion		quireu	switching not	
								required	
			SRAM				517120 points (ZR0 to		
			card		-		517120 points (ZR0 to 517119), Block switching no		
			(1M byte)					uired	
			SRAM					oints (ZR0 to	
			card		-			ck switching not	
			(2M byte)				-	uired	
		[ZR]	SRAM					oints (ZR0 to	
			card		-		-	ck switching not	
			(4M byte)					uired	
		(4M byte) SRAM card					-	oints (ZR0 to	
					-			ck switching not	
			(8M byte)					uired	
							-	oints (ZR0 to	
			Flash card		-		-		
			(2M byte)				1041407), Block switching not required		
							required 2087936 points (ZR0 to		
			Flash card		-		-	ck switching not	
			(4M byte)		-		-	uired	
		1					ieq	anou	

*7: When a Flash card is used, only reading is possible. ATA cards cannot be used.

	Item					Universal mo	del QCPU				
		Item						Q03UDCPU			
				Q00UJCPU	Q00UCPU	Q01UCPU	Q02UCPU	Q03UDECPU			
	Step	relay [S] ^{*8}		8192 p	8192 points (S0 to 8191) (the number of device points is fixed.) *9						
	Index	register/ lard devise re	gister [Z]			Max. 20 point					
	Index (32-bi	register [Z] it indexing s device)		-	(Ir	0 points (Z0 (Z) is used in	to 18) double words.)				
Number	Pointe	er [P]		The available ranges of the local The available ranges of the local pointers and common pointers can be			The availa pointers an	points (P0 to 4095), ble ranges of the local d common pointers can et by parameters.			
of device points	vice ints Interrupt pointer [I]			The constant cyclic interval of system interrupt pointers l28 to 31 can be set by parameters.The const system inter can be set (0.5 to 1000ms, 0.5ms unit)Default value l28: 100ms, l29: 40ms,Default value				points (I0 to 255) stant cyclic interval of errupt pointers I28 to 31 set by parameters. 1000ms, 0.5ms unit) alue I28: 100ms, I29: 30: 20ms, I31: 10ms			
	Speci	al relay [SM]				ice points is fixed.)					
		al register [SI	01	2048 points (SD0 to 2047) (the number of device points is fixed.)							
		ion input [FX]	-	16 points (FX0 to F) (the number of device points is fixed.)							
		ion output [F			•			· · ·			
		ion register [F	-	16 points (FY0 to F) (the number of device points is fixed.) 5 points (FD0 to 4) (the number of device points is fixed.)							
Number o		ce tracking wo	-	•	X	,					
Link direc	ct devic	e		Device for accessing the link device directly Dedicated to CC-Link IE controller network and MELSECNET/H Specified form: J□\X□, J□\Y□, J□\W□, J□\B□, J□\SW□, J□\SB□							
Intelligent	t functio	on module de	vice	Device for	accessing the	e buffer memo direc Specified for	tly	lligent function module			
		Data transmis	sion speed		-			100/10Mbps			
		Communicati	1		-			Full-duplex/Half-duplex			
	Transmission method Max. distance between hub and node		method		-			Base band			
0					-			100m			
Specifications of built-in Ethernet port		Max. number of	10BASE-T		-			Cascade connection: Max. four nodes			
CPU module ^{*10}		connectable nodes	10BASE-T		-			Cascade connection: Max. two nodes			
module		Number of connections ^{*11}			-			16 in total for socket communication, MELSOFT connection, and MC protocol, 1 for FTP			

*8: The step relay is a device for the SFC function.

*9: For the Universal model QCPU whose serial number (first five digits) is "10042" or later, the number of device points can be changed to zero.

*10: For the Built-in Ethernet port QCPU

*11: The number is a total of TCP/IP and UDP/IP.

			Un	iversal model QC	PU	
Item		0001110011	000000000	004110011	000110011	Q03UDCPU
		Q00UJCPU	Q00UCPU	Q01UCPU	Q02UCPU	Q03UDECPU
Latab ranga			L0 to 819	91 (8192 points b	y default)	
Latch range		(Latch range c	an be set for B, I	⁻ , V, T, ST, C, D,	and W) (setting I	by parameters)
RUN/PAUSE contact		One contact ca	in be set up in X0	to 1FFF for each	n of RUN and PA	USE (setting by
					Year, month, da	ate, hour, minute,
					second, and c	lay of the week
		Year, month, o	date, hour, minut	e, second, and	(automatic leap	year detection)
		day of the weel	< (automatic leap	year detection)	Accuracy: -2	.96 to +3.74s
Clock function		Accuracy: -2.96	to +3.74s (TYP.	(TYP. +1.42s)/d at 0°C		
		Accuracy: -2.34	to +3.74s (TYP. +	Accuracy: -3	.18 to +3.74s	
		Accuracy: -11.48	B to +2.12s (TYP.	(TYP. +1.50s)/d at 25°C		
				Accuracy: -13.20 to +2.12s		
					(TYP3.54	s)/d at 55°C
		20ms or less				
Allowable momentary po	ower	(100VAC or	Varies	depending on th	e power supply n	nodule.
failure time		more)				
5VDC internal current consumption		0.37A ^{*12}	0.33A		0.23A	0.33A ^{*13}
	Н	98mm	98mm			
External dimensions	W	245mm ^{*14}	27.4mm			
	D	98mm	89.3mm ^{*15}			
Weight	•	0.70kg ^{*14}	0.1	5kg	0.20	kg ^{*15}

*12: The value is for the CPU module and base unit together.

*13: The value is 0.46A for the Q03UDECPU.

*14: The value includes the CPU module, power supply module, and base unit.

*15: The following values are applied for the Q03UDECPU.

External dimensions (D) : 115mm Weight : 0.22kg

3 - 5

(2) Q04UD(E)HCPU, Q06UD(E)HCPU, Q10UD(E)HCPU, Q13UD(E)HCPU, Q20UD(E)HCPU, Q26UD(E)HCPU

					Universal m	odel QCPU				
	Item		Q04UDHCPU	Q06UDHCPU	Q10UDHCPU	Q13UDHCPU	Q20UDHCPU	Q26UDHCPU		
			Q04UDEHCPU	Q06UDEHCPU	Q10UDEHCPU	Q13UDEHCPU	Q20UDEHCPU	Q26UDEHCPU		
Control met	nod				Stored program	repeat operation	l			
I/O control r	node				Refres	n mode				
			(Direct access I/O is available by specifying direct access I/O (DX□, DY□).)							
Program	Sequence language	e control	Relay symbol language, logic symbolic language, MELSAP3 (SFC), MELSAP-L, function block and structured text (ST)							
language	Process of language	control				-				
Processing	LD X0				9.5	ins				
speed (sequence instruction)	MOV D0	D1			19	ns				
Processing speed (redundant function)	Tracking o time (incro time)	execution eased scan			-	-				
Constant sc (function for time)	-	egular scan	0.8	0.5 to 2000ms (setting available in 0.5ms unit) (setting by parameters)						
Program ca	°1 *2							260K steps (1040K byte)		
	Program ((drive 0)	memory						1040K byte		
	Memory c (drive 1)	ard (RAM)		Capacity c	of the installed m	emory card (8M	byte max.)			
	Memory c (drive 2)	ard (ROM)	Capacity of th	e installed mem	ory card (Flash o	card: 4M byte ma	ax., ATA card: 32	2M byte max.)		
	Standard (drive 3)	RAM	256K byte	768K byte	1024	K byte	1280	K byte		
Memory capacity ^{*1}	Standard (drive 4)	ROM	1024	K byte	2048	K byte	4096	K byte		
	QCPU standard CPU memory		8K byte							
	shared memory *3	Multiple CPU high speed transmissi on area			32K	byte				

*1: The size unit of the files stored in the memory area differs depending on the CPU module.

For details, refer to the QnUCPU User's Manual (Function Explanation, Program Fundamentals).

 *2: The maximum number of executable sequence steps is obtained by the following formula. (Program capacity) - (File header size (Default: 34 steps))

For details of the program capacity and files, refer to the QnUCPU User's Manual (Function Explanation, Program Fundamentals). *3: Data in the CPU shared memory is not latched.

Data in the CPU shared memory is cleared when the programmable controller is powered on or the CPU module is reset.

	Item					Universal m	odel QCPU						
		Item		Q04UDHCPU	Q06UDHCPU	Q10UDHCPU	Q13UDHCPU	Q20UDHCPU	Q26UDHCPU				
				Q04UDEHCPU	Q06UDEHCPU	Q10UDEHCPU	Q13UDEHCPU	Q20UDEHCPU	Q26UDEHCPU				
		Program r	memory	12	24		25	2*4					
		Memory c (RAM)	ard		31	9 (when the Q3M	IEM-8MBS is use	ed)					
Max num of fil	nber	Memory	Flash card	288									
stor		card (ROM)	ATA card			5 [.]	11						
		Standard	RAM	3 files (each	3 files (each one of the following files: file register file, local device file, and sampling trace file)								
		Standard	ROM			25	56						
		times of wi	-			Max. 1000	00 times ^{*5}						
		times of wi andard RON	Ū			Max. 1000	00 times ^{*5}						
(nur		I/O device f usable poi	•			8192 points ()	X/Y0 to 1FFF)						
Nun (nur	nber of mber o	I/O points f points acc I/O module		4096 points (X/Y0 to FFF)									
une i		al relay [M]	*7		8192 pc	oints by default (N	10 to 8191) (chan	deable)					
		relay [L] ^{*7}			•	, , , , , , , , , , , , , , , , , , ,	.0 to 8191) (chan	,					
		elay [B] ^{*7}		8192 points by default (B0 to 1FFF) (changeable)									
	Time				The low- and hi asurement units o (Low-speed t	gh-speed timers of the low- and hig imer: 1 to 1000m	able) (sharing of l are specified by t gh-speed timers a s, 1ms unit, 100n is, 0.1ms unit, 10	he instructions. are set up by para ns by default)					
of device points	Reter	ntive timer [ST] ^{*7}	The measure (by default (sharing e low- and high-s ement units of the Low-speed reten	g of the low- and peed retentive tin low- and high-sp tive timer: 1 to 10	high-speed reten ners are specified beed retentive tim 100ms, 1ms unit, 100ms, 0.1ms uni	tive timers) (char d by the instructio lers are set up by 100ms by default	ns. parameters.				
ber	Coun	ter [C] ^{*7}			Normal counter,	1024 points by c	lefault (C0 to 102	3) (changeable)					
Number	Data	register [D]	*7		12288 pc	oints by default (E	00 to 12287) (cha	ngeable)					
	Exter	ided data re	egister [D]			0 point by defau	ult (changeable)						
	Link r	egister [W]	*7		8192 po	ints by default (V	/0 to 1FFF) (char	ngeable)					
*7							ult (changeable)						
	Annu	nciator [F]	7		2048 po	pints by default (F	0 to 2047) (chan	geable)					
	Edge relay [V] ⁷ 2048 points by default (V0 to 2047) (changeable)												
	Link s	special relay	y [SB] ^{*7}	2048 points by default (SB0 to 7FF) (changeable)									
	Link ş [SW]	special regis	ster		2048 po	ints by default (S	W0 to 7FF) (char	ngeable)					

*4: Up to 124 programs can be executed in the CPU module. (The CPU module cannot execute 125 or more programs.)

*5: A single writing operation may not be counted as one.

The number of writing into the program memory can be checked with the special register (SD682 and SD683). *6: A single writing operation may not be counted as one.

The number of writing into the standard ROM can be checked with the special register (SD687 and SD688).

*7: The number of points can be changed within the setting range.

						Universal m	odel QCPU				
	I	tem		Q04UDHCPU	Q06UDHCPU	Q10UDHCPU	Q13UDHCPU	Q20UDHCPU	Q26UDHCPU		
				Q04UDEHCPU	Q06UDEHCPU	Q10UDEHCPU	Q13UDEHCPU	Q20UDEHCPU	Q26UDEHCPU		
			Standard RAM	32768 points (R0 to 32767) Max. 131072 points by block switching	32768 points (R0 to 32767) Max. 393216 points by block switching	•	(R0 to 32767) points by block ching	Max. 655360 p	(R0 to 32767) points by block ching		
			SRAM card (1M byte)	Max.	Max. 517120 points by block switching in units of 32768 points (R0 to 32767)						
		[R]	SRAM card (2M byte)	Max. 1	041408 points b	y block switching	g in units of 3276	8 points (R0 to 3	32767)		
			SRAM card (4M byte)	Max. 2	2087936 points b	y block switching	g in units of 3276	8 points (R0 to 3	32767)		
			SRAM card (8M byte)	Max. 4184064 points by block switching in units of 32768 points (R0 to 32767)							
e points	points		Flash card (2M byte)	Max. 1041408 points by block switching in units of 32768 points (R0 to 32767)							
ⁱ device	File register ^{*8}		Flash card (4M byte)	Max. 2087936 points by block switching in units of 32768 points (R0 to 32767)							
Number of device points	register		Standard RAM	131072 points (ZR0 to 131071) Block switching not required	393216 points (ZR0 to 393215) Block switching not required	(ZR0 to	3 points 524287) Ig not required	(ZR0 to) points 655359) Ig not required		
			SRAM card (1M byte)		517120 points	(ZR0 to 517119), Block switchin	g not required			
		[ZR]	SRAM card (2M byte)		1041408 points	(ZR0 to 104140	07), Block switchi	ng not required			
			SRAM card (4M byte)		2087936 points	(ZR0 to 208793	5), Block switchi	ng not required			
			SRAM card (8M byte)		4184064 points	(ZR0 to 418406	3), Block switchi	ng not required			
			Flash card (2M byte)		1041408 points	(ZR0 to 104140	07), Block switchi	ing not required			
			Flash card (4M byte)		2087936 points	(ZR0 to 208793	5), Block switchi	ing not required			

*8: When a Flash card is used, only reading is possible. ATA cards cannot be used.

					Universal m	nodel QCPU				
	Item		Q04UDHCPU	Q06UDHCPU	Q10UDHCPU	Q13UDHCPU	Q20UDHCPU	Q26UDHCPU		
			Q04UDEHCPU	Q06UDEHCPU	Q10UDEHCPU	Q13UDEHCPU	Q20UDEHCPU	Q26UDEHCPU		
	Step relay [S] ^{*9}		8	3192 points (S0	to 8191) (the nu	mber of device p	oints is fixed.) ^{*1}	0		
	Index register/Si register [Z]	tandard devise	Max. 20 points (Z0 to 19)							
	Index register [Z	<u>[]</u>	Max 10 points (70 to 18)							
	(32-bit indexing	specification of	Max. 10 points (Z0 to 18) (Index register (Z) is used in double words.)							
ıts	ZR device)			(Inde:	k register (Z) is t	ised in double w	ords.)			
Number of device points	Pointer [P]		4096 points (P	0 to 4095), The	available ranges be set by p	of the local poir oarameters.	nters and comm	on pointers can		
f de∖					256 points	(I0 to 255)				
er ol			The constant	cyclic interval of	f system interrup	ot pointers I28 to	31 can be set b	y parameters.		
qur	Interrupt pointer	[1]			(0.5 to 1000m	ns, 0.5ms unit)				
ź				Default value	128: 100ms, 129): 40ms, I30: 20n	ns, I31: 10ms			
	Special relay [SI	M]	2048 points (SM0 to 2047) (the number of device points is fixed.)							
	Special register	[SD]		2048 points (SD	0 to 2047) (the r	number of device	e points is fixed.)		
	Function input [F	FX]		16 points (F)	(0 to F) (the num	nber of device po	pints is fixed.)			
	Function output	[FY]	16 points (FY0 to F) (the number of device points is fixed.)							
	Function registe	r [FD]	5 points (FD0 to 4) (the number of device points is fixed.)							
Num	ber of device tra	cking words	-							
			Device for accessing the link device directly							
Num	ber of device tra	cking words	Dedicated to CC-Link IE controller network and MELSECNET/H							
			Specified form: JDD/XDD, JDD/YDD, JDD/WDD, JDD/BDD, JDD/SWDD, JDD/SBDD							
Intol	ligent function mo	odule device	Device f	or accessing the	e buffer memory	of the intelligent	function module	e directly		
inter					Specified forn	n: UDD\GDD				
t.	Data transm	ission speed			100/1	OMbps				
erne	Communicat	tion mode			Full-duplex	/Half-duplex				
built-in Ethernet	Transmission	n method			Base	band				
lt-in	Transmission	e between hub			10	0m				
	C		100m							
us o	A Max. numbe	r 10BASE-T		Ca	scade connectio	on: Max. four noc	des			
Specifications of	Max. numbe of connectable nodes	100BASE-TX	Cascade connection: Max. two nodes							
	Number of c	onnections ^{*12}	16 in total fo	or socket commu	inication, MELS	OFT connection,	, and MC protoc	ol, 1 for FTP		

*9: The step relay is a device for the SFC function.

*10: For the Universal model QCPU whose serial number (first five digits) is "10042" or later, the number of device points can be changed to zero.

*11: For the Built-in Ethernet port QCPU

*12: The number is a total of TCP/IP and UDP/IP.

				Universal m	odel QCPU			
Item		Q04UDHCPU	Q06UDHCPU	Q10UDHCPU	Q13UDHCPU	Q20UDHCPU	Q26UDHCPU	
		Q04UDEHCPU	Q06UDEHCPU	Q10UDEHCPU	Q13UDEHCPU	Q20UDEHCPU	Q26UDEHCPU	
Latel and a				L0 to 8191 (8192	points by default)			
Latch range		(Lat	ch range can be s	et for B, F, V, T, S	T, C, D, and W) (s	setting by paramet	ers)	
RUN/PAUSE contact		One contac	t can be set up in	X0 to 1FFF for ea	ch of RUN and PA	USE (setting by p	parameters)	
		Year, mont	Year, month, date, hour, minute, second, and day of the week (automatic leap year detection)					
Ole als five ation		Accuracy: -2.96 to +3.74s (TYP. +1.42s)/d at 0°C						
Clock function			Accura	cy: -3.18 to +3.74s	s (TYP. +1.50s)/d a	at 25°C		
			Accura	cy: -13.20 to +2.12	s (TYP3.54s)/d	at 55°C		
Allowable momentary								
power failure time			Varie	s depending on the	e power supply mo	odule.		
5VDC internal current					<u>,</u> *13			
consumption		0.39A ^{*13}						
	н	98mm						
External dimensions	W			27.4	mm			
	D			89.3n				
Weight 0.20kg ^{*14}								

*13: The value is 0.49A for the Q04UDEHCPU, Q06UDEHCPU, Q10UDEHCPU, Q13UDEHCPU, Q20UDEHCPU, and Q26UDEHCPU.

*14: The following values are applied for the Q04UDEHCPU, Q06UDEHCPU, Q10UDEHCPU, Q13UDEHCPU, Q20UDEHCPU, and Q26UDEHCPU.

External dimensions (D): 115mm Weight: 0.22kg

3.2 Device

The following table lists the devices used in QCPU and applicable ranges.

				Default		Setting
Classification	Туре	Device name	Points	Range		range by
			FUILS	Kange		parameters
		Input	8192	X0 to X1FFF	Hexadecimal	
		Output	8192	Y0 to Y1FFF	Hexadecimal	
		Internal relay	8192	M0 to M8191	Decimal	
	Bit	Latch relay	8192	L0 to L8191	Decimal	
	device	Annunciator	2048	F0 to F2047	Decimal	
	device	Edge relay	2048	V0 to V2047	Decimal	Can be
Internal user		Step relay	8192	S0 to S511/block	Decimal	changed
device		Link relay	8192	B0 to B1FFF	Hexadecimal	within 29K
device		Link special relay	2048	SB0 to SB7FF	Hexadecimal	words ^{*3}
		Timer ^{*1}	2048	T0 to T2047	Decimal	words
		Retentive timer ^{*1}	0	(ST0 to ST2047)	Decimal	
	Word	Counter ^{*1}	1024	C0 to C1023	Decimal	
	device	Data register	12288	D0 to D12287	Decimal	
		Link register	8192	W0 to W1FFF	Hexadecimal	
		Link special register	2048	SW0 to SW7FF	Hexadecimal	
	Bit	Function input	16	FX0 to FXF	Hexadecimal	
Internal	device	Function output	16	FY0 to FYF	Hexadecimal	Cannot be
Internal system device	uevice	Special relay	2048	SM0 to SM2047	Decimal	changed.
system device	Bit	Function register	5	FD0 to FD4	Decimal	changeu.
	device	Special register	2048	SD0 to SD2047	Decimal	
		Link input	8192	Jn\X0 to Jn\X1FFF	Hexadecimal	
	Bit	Link output	8192	n\Y0 to Jn\Y1FFF	Hexadecimal	
Link direct	device	Link relay	16384	Jn\B0 to Jn\B3FFF	Hexadecimal	Cannot be
device		Link special relay	512	Jn\SB0 to Jn\SB1FF	Hexadecimal	changed.
	Word	Link register	16384	Jn\W0 to Jn\W3FFF	Hexadecimal	
	device	Link special register	512	Jn\SW0 to Jn\SW1FF	Hexadecimal	
		Intelligent function	65536	Un\G0 to	Decimal	Cannot be
Module	Word	module device	00000	Un\G65535 ^{*2}	Decimal	changed.
access device	device	Multiple CPU shared	14336	U3En\G10000 to	Decimal	Can be
		device ^{*4}	14330	U3En\G24335	Decimal	changed.

Table 3.1 Device list

*1: For the timer, retentive timer, and counter, a bit device is used for contacts and coils, and a word device is used for a current value.

*2: The number of points that can be actually used varies depending on the intelligent function module.

For the number of buffer memory points, refer to the manual for the intelligent function module used.

*3: The number of device points can be changed in the Device tab of the Q Parameter Setting dialog box (the points for input relay, output relay, and step relay cannot be changed). For the Universal model QCPU whose serial number (first five digits) is "10042" or later, the points for step relay can be changed to 0.

*4: Available only in multiple CPU systems.

				Default		Setting	
Classification	Туре	Device name	Points	Range		range by parameters	
Index register/Standard devise register	Word device	Index register/Standard devise register	20	Z0 to Z19	Decimal	Cannot be changed.	
File register ^{*7}	Word device	File register	0	-	-		
Extended data register ^{*7}	Word device	Extended data register	0	-	-	0 to 4086K points ^{*6}	
Extended link register ^{*7}	Word device	Extended link register	0	-	-		
Nesting	-	Nesting	15	N0 to N14	Decimal	Cannot be changed.	
Pointer		Pointer	4096 ^{*8}	P0 to P4095 ^{*9}	Decimal	Cannot be	
Pointer	-	Interrupt pointer	256 ^{*10}	10 to 1255 ^{*11}	Decimal	changed.	
	Bit device	SFC block device	256 ^{*10}	BL0 to BL319 ^{*12}	Decimal		
Other		Network No. specification device	255	J1 to J255	Decimal	Cannot be	
Other	-	I/O No. specification device	-	U0 to UFF, U3E0 to U3E3 ^{*13}	Hexadecimal	changed.	
		Macro instruction argument device	-	VD0 to VD□	Hexadecimal		
		Decimal constant		K-2147483648 to	K2147483647		
		Hexadecimal constant	H0 to HFFFFFFF				
				Single-precision floa	ating point data:		
Constant	-	Real number	I	E ± 1.17549435 - 38 to E			
		constant		Double-precision floa	ting point data ^{*5}	:	
			E ± 2.2250	738585072014 - 308 to E	± 1.7976931348	3623157 + 308	
		Character string constant		"ABC", "	123"		

*5: Up to 15 digits can be entered in GX Works2.

*6: Indicates the total number of points for the file register, extended data register (D), and extended link register (W).

*7: Not available for the Q00UJCPU.

*8: The points are 512 points for the Q00UJCPU, Q00UCPU, and Q01UCPU.

*9: The range is from P0 to P511 for the Q00UJCPU, Q00UCPU, and Q01UCPU.

*10: The points are 128 points for the Q00UJCPU, Q00UCPU, and Q01UCPU.

*11: The range is from I0 to I127 for the Q00UJCPU, Q00UCPU, and Q01UCPU.

*12: The range is from BL0 to BL127 for the Q00UJCPU, Q00UCPU, and Q01UCPU.

*13: The range is from U0 to UF for the Q00UJCPU, from U0 to U3F or from U3E0 to U3E2 for the Q00UCPU and Q01UCPU, and from U0 to U7F or from U3E0 to U3E2 for the Q02UCPU.

Since device types are the same as those of the MELSEC-QnA series, the advantage of easy program creation is succeeded.

The following explains devices which are unique to the MELSEC-Q series.

(1) Special relay/Special register (SM/SD)

The special relay and special register are used for writing or reading data between the QCPU and user program.

The special relay/special register (SM/SD) include the following:

- SM1: Self-diagnostic error
- SM52: Battery low
- (2) Retentive timer (ST)

The device name is expressed as ST to be distinguished from the normal timer. <Example> OUT ST100 K500

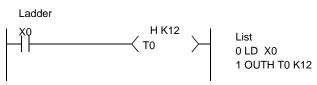
(3) Low-speed timer/High-speed timer (T)

The measurement unit can be changed. To change the setting, use the parameter.

In addition, the low-speed timer and the high-speed timer can be distinguished in a program.

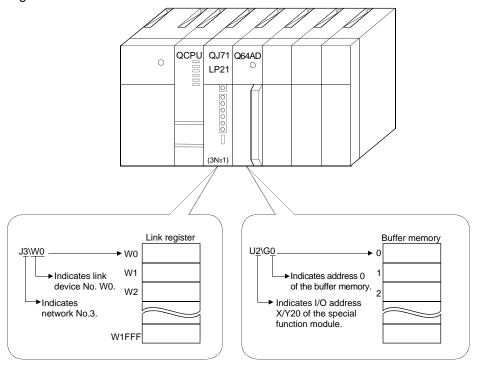
<Example> Low-speed timer: OUT T200 K12

High-speed timer: OUTH T200 K12



(4) Intelligent function module device $(U\Box \GD)$

The intelligent function module device of the intelligent function module or special function module can be accessed directly from the QCPU as a data register.



(5) File register (R/ZR)

The file register is for extending the data register. For the file register, use the standard RAM or the memory card (SRAM, Flash card).

The following explains the maximum capacity of the file register.

When using the standard RAM

The following table shows the maximum points of the file register data that can be stored in the standard RAM.

However, if the standard RAM is used for an application other than file registers, available points are decreased.

CPU module	Points
Q00UCPU, Q01UCPU, Q02UCPU	64K
Q03UDCPU, Q03UDECPU	96K
Q04UDHCPU, Q04UDEHCPU	128K
Q06UDHCPU, Q06UDEHCPU	384K
Q10UDHCPU, Q10UDEHCPU, Q13UDHCPU,	512K
Q13UDEHCPU	512K
Q20UDHCPU, Q20UDEHCPU, Q26UDHCPU,	640K
Q26UDEHCPU	040K

When using an SRAM card

Up to 4086K points can be stored in one file.

Since one block consists of 32K words, up to 128 blocks can be stored. Note that the number of points or blocks that can be added depends on the capacity of the programs and device comments stored in the memory card.

When using a Flash card

Up to 2039K points can be stored in one file.

Since one block consists of 32K words, up to 64 blocks can be stored.

Note that the number of points or blocks that can be added depends on the memory card capacity and the capacity of the programs and device comments stored in the memory card.

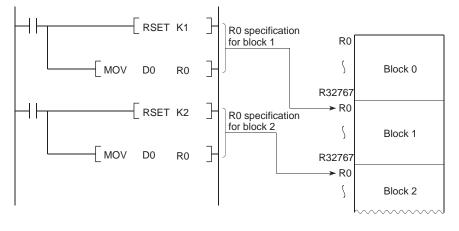
The large-capacity data can be accessed with a block unit of 32K words or the whole file register in series.

(a) Block switching method

The used file register points are divided and specified in units of 32K points (R0 to R32767).

When multiple blocks are used, the desired block is specified with the block number in the RSET instruction.

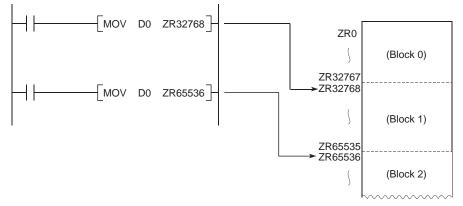
Each block has a specification range of R0 to R32767.



(b) Serial number access method

A file register whose capacity is exceeding 32K points can be specified with consecutive device numbers.

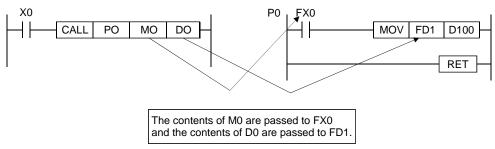
Multiple blocks of a file register can be used as a continuous file register. This type of the device is expressed as "ZR".



(6) Function devices (FX, FY, FD)

Function devices are used in subroutine programs with argument passing. Data are read or written between such subroutine programs and calling programs, using function devices.

- Each device is used as below.
 - $FX \rightarrow Bit$ condition input by a subroutine
 - $FY \rightarrow Bit$ output condition
 - $FD \rightarrow I/O$ data condition



(7) Step relay (S)

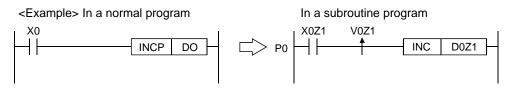
This relay is dedicated to SFC for indicating an active status of each step. In sequence programs, a step relay can be specified with the block.

 $(Example) \quad BL2\S1 \ ... \ Specifies \ the \ step \ relay \ 1 \ of \ the \ block \ No. \ 2.$

(8) Edge relay (V)

The edge relay is for generating pulses in the repeatedly executed programs such as subroutine programs and interrupt programs.

Using this device makes the pulse generation instructions such as a subroutine and the interrupt program easier to be used.



(9) Link direct device $(J\Box \setminus \Box)$

The link direct device is for the direct access to the link device in a CC-Link IE Controller Network module or MELSECNET/H module.

Using this device shortens the transmission time of the link device. Also, the link range which is not set by the network refresh parameter is accessible.

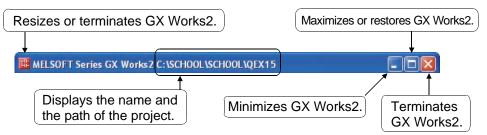
CHAPTER 4 BASIC KNOWLEDGE REQUIRED FOR OPERATING GX Works2

4.1 Screen Configuration in GX Works2

(1) Title bar		2) Menu b	par
MELSOFT Series GX Works2 C:\SCHOOL\SCHOOL	OFX15		
Eroject Edit Eind/Replace Compile View Online	ebug Diagnostics <u>T</u> ool <u>W</u> indow <u>H</u> elp	+	
: D ≥ P 0	월] 🥦 🖳 🖼 📮 💭 👧 🛃 🌄 🎇 🕌 🕌 -		
		Network Parameter - MELSECNET/C	4) Tab
Project			
Contraction of the second			
😑 🚯 Network Parameter	rork Parameter - MELSECNET/CC IE/Ethernet Mod	lule Configuration	
Ethernet / CC IE / MELSEC	M-	dule 1 Module	
Intelligent Function Module	contents display area	7) Edit screen (work window)
Program Setting			
A Program X8 X8 X9	5		(11 K3)
E Gal Device Comment			
Device Memory			(M1)
Project XOE			(M2)
User Library			except 2
Connection Destination	ew selection area	8) Output window	(177)
*	M1		К1000
Output		+	÷×
Change the Language No. Result Data Name Class	Content		Error Code
Error: 0			
Error: 0	Unlabeled	Q06UDEH Host Station	0/45Step Ovrwite CAP NUM ;;
English		Q06UDEH Host Station	0/45Step Ovrwrte CAP NUM
		Q06UDEH Host Station	0/45Step Ovrwrte CAP NUM
English			0/455tep Ovrwrte CAP NUM
English	Ar	roject) [[PRG]W Qnine Debug	0/45Step Ovrwite CAP NUM
English	MELSOFT Series GX Works2 (Untitled P Brosk Lak End/Replace Comple Yew Bew Ctrl+N Popen Ctrl+N	roject) - [[PRG]W	0/45Step Ovwwrte CAP NUM
English	MELSOFT Series GX WorksZ (Untilled P Projek Edt End/Replace Comple Yew End/Replace Comple	roject) [[PRG]]V Online Debug	0/45Step Ovrwrte CAP NUM .::
English	Ar MELSOFT Series GX Works2 (Untitled P Proje, Edt Eind/Replace Comple Yew Dew Ctri+N Save Ctri+S Save Ctri+S	oject) [[PRG]]V Online Debug	0/455tep Ovrwrte CAP NUM
English	Ar MELSOFT Series GX Works2 (Untitled P Proe Gate End/Replace Comple Year Dem Ctrl+N Spen Ctrl+N Cose Cose Cose Corl+S	roject) [[PRG]¥ Qnine Debug] 그 가 나 가 나 가 가 가 가 가 가 가 가 가 가 가 가 가 가 가	0/45Step Ovrwite CAP NUM
English	AT MELSOFT Series GX Works 2 (Untitled P Profession Edge End/Replace Comple View Dem Ctrl+N Sore Ctrl+S Save & Cogpress/Unpack Verfy	roject) [[PRG]¥ Qnine Debug] 그 가 나 가 나 가 가 가 가 가 가 가 가 가 가 가 가 가 가 가	0/45Step Ovrwrte CAP NUM
English	AT	roject) [[PRG]¥ Qnine Debug] 그 가 나 가 나 가 가 가 가 가 가 가 가 가 가 가 가 가 가 가	0/455kep Ovrwrte CAP NUM
English	AT MELSOFT Series GX Works2 (Untilled P MELSOFT Series GX Works2 (Un	roject) [[PRG]¥ Qnine Debug] 그 가 나 가 나 가 가 가 가 가 가 가 가 가 가 가 가 가 가 가	0/45Step Ovrwrte CAP NUM
English	AT	roject) [[PRG]¥ Qnine Debug] 그 가 나 가 나 가 가 가 가 가 가 가 가 가 가 가 가 가 가 가	0/45Step Ovrwrte CAP NUM .::
English	AT	roject) [[PRG]¥ Qnine Debug] 그 가 나 가 나 가 가 가 가 가 가 가 가 가 가 가 가 가 가 가	0/45Step Ovrwete CRP NUM
English	AT MELSOFT Series GX Works 2 (Untitled P Projek Eds End/Replace Comple View Dem Ctrl+N Gose Gose Save & Cogpress/Unpack Delete Yerfy Projek Revision Change Project Type Object Intelligent Eurotion Module	ojeci) [[PRG]¥ Qnine Debus] J H H H H PRG]Write MAI	0/45Step Ovrwrte CGP NUM
English	AT MELSOFT Series GX Works2 (Untitled P Profe & Edt End/Replace Comple Vew Profe & Edt End/Replace Comple Vew Dem Ctrl+N Gose Ctrl+O Gose Ctrl+O Gose Ctrl+O Gose Ctrl+O Cagress(Urpack Profec Revision Change Project Type Object Tabellgent Eunction Module Eguort to GX Developer Format File Lbrary Security	roject) [[PRG]¥ Qnine Debug] 그 가 나 가 나 가 가 가 가 가 가 가 가 가 가 가 가 가 가 가	0/455tep Ovrwrte CAP NUM
English	AT	ojeci) [[PRG]¥ Qnine Debus] J H H H H PRG]Write MAI	0/45Step Ovrwite CAP NUM
English	AT MELSOFT Series GX Works 2 (Untilled P Profix Eds End/Replace Comple Yew Bew Cri+N Gose Save Cri+S Save C	ojeci) [[PRG]¥ Qnine Debus] J H H H H PRG]Write MAI	0/45SRep Ovrwite CAP NUM .::
English	AT MELSOFT Series GX Works 2 (Untilled P Profix Set End/Replace Comple Yew Bew CritHN Gose Save CritHo Gose Save CritHo Gose Coggress/Unpack Pelete Yerfy Projek Revision Change Project Type Object Intelligent Eurotion Module Egoot to GX Developer Format File Urary Security Prink Window Preview Prink Window Previe	ojeci) [[PRG]¥ Qnine Debus] J H H H H PRG]Write MAI	0/45SRp Ovrwite GAP NUM .::
English	AT	ojeci) [[PRG]¥ Qnine Debus] J J J H H H H PRG]Write MAI	0/45Step Ovrwite CAP NUM j
English	AT	ojeci) [[PRG]¥ Qnine Debus] J J J H H H H PRG]Write MAI	0/455tep Ovrwite CAP NUM j
English	AT	ojeci) [[PRG]¥ Qnine Debus] J J J H H H H PRG]Write MAI	0/45Step Ovrwite CAP NUM

1) Title bar

Title bar displays the name of the active project.



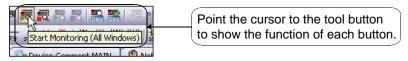
2) Menu bar

Menu bar is a most frequently used item to operate GX Works2. Click the menu bar to select a variety of functions from the drop-down menu.

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help

3) Toolbar

Toolbar equips buttons to easily access the commonly-used functions. This enables a quicker operation.



4) Tab

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

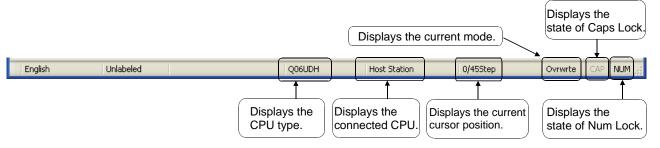
- View contents display area
 View contents display area displays the contents of the currently selected view.
- View selection area
 View selection area allows selection of the view to be displayed.
- 7) Edit screen (work window)

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

 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.



10) Drop-down menu

Drop-down menu displays the names of the available functions in GX Works2. Selecting the function with " > " on the right end displays the related drop-down menu.

Clicking the function with "...." after the name displays the setting dialog box.

	Displays a dialog box.	New Ctrl+N Open Ctrl+O Glose	đ	[PRG]Write MAIN 1 Step ×]
New Project Project Type: Simple Project PLC Series:	OK Cancel	Delete Verify Project Rev Change PLC Type Change Project Type	p-c	down menu.
QCPU (Q mode)	•	Object Intelligent Function Module		Rename
PLC Type:	•	Open Other Data Export to GX Developer Format File	•	A REAL PROPERTY AND A REAL
Language:		Library Security		Paste Ctrl+Shift+V Set as Default Connection
Ladder		Print(]) Print Preview(B) Print Window Print Window Preview		Property Alt+Enter

4.2 Basic Operations of Dialog Box

This section explains the screen configuration of the dialog box.

Q Parameter Setting	
	Device I/O Assignment Multiple CPU Setting Serial Communication
Low s 7) Tab ms (1ms1000ms) High Speed 10.00 ms (0.01ms100ms)	Common Pointer No. P \checkmark 4) Text box
RUN + AUSE Contacts RUN X (X0X1FFF) PAUSE X (X0X1FFF) Latch Data Backup Operation Valid Contact	System Interrupt Settings Interrupt Counter Start No. C (0-766) Fixed Scan Interval I28 [100.0 ms (0.5ms-1000ms)
Device Name	I29 40.0 ms (0.5ms~1000ms) I30 20.0 ms (0.5ms~1000ms) I31 10.0 ms (0.5ms~1000ms) High Speed Interrupt Settings
Previous State Recalculate(Output is 1 scan later) Floating Point Anthmetic Processing Perform Internal anthmetic operations in double precision	Interrupt Program / Fixed Scan Program Setting High Speed Execution A-PLC Compatibility Setting Use special relay / special register from SM/SD 1000
Intelligent Function Module Setting Interrupt Pointer Setting Module Synchronization Synchronize intelligent module's pulse up Pulit-in CC-Unik Setting	Service Processing Setting Execute the process as the scan 10 % time proceeds Specify service process time ms (0.2ms-1000ms) Specify service process time ms (0.2ms-1000ms) Specify service process Times (110 Times) execution counts
(*1)Setting should be set as same when using multiple CPU.	C Execute it while waiting for constant scan setting PLC Module Change Setting PLC Module Change Setting 1) Command button
Print Window Print Window Preview Acknowle	edge XY Assignment Default Check End Cancel

Zoom		
Magnification –		
2	6) Spin b	ох
C <u>5</u> 0%		
Specify Auto	90 🐳	% Columns
ОК		Cancel

1) Command button	
-------------------	--

Command buttons include OK and Cancel Click each button to execute its operation.

2) Checkbox

Click \Box to put \boxdot in the box to execute the operation.

3) List box

After the selection list is displayed by clicking **_**, click an item to select.

4) Text box

Enter letters in a text box with the keyboard. Only numbers can be entered depending on the text box type.

- 5) Radio button Click O of an item to select.
- 6) Spin box

Values can be entered directly or changed by the <a>button. To enter a value directly in a spin box, click <a>button, then enter the value with the keyboard. When changing a value by clicking the <a>button, <a>button, <a>button, <a>button, <a>button, <a>button. Clicking <a>button, <a>button, <a>button, <a>button.

Tab
 Clicking ______ switches the screens in which setting items are displayed.

4.3 Ladder Program Creation Method

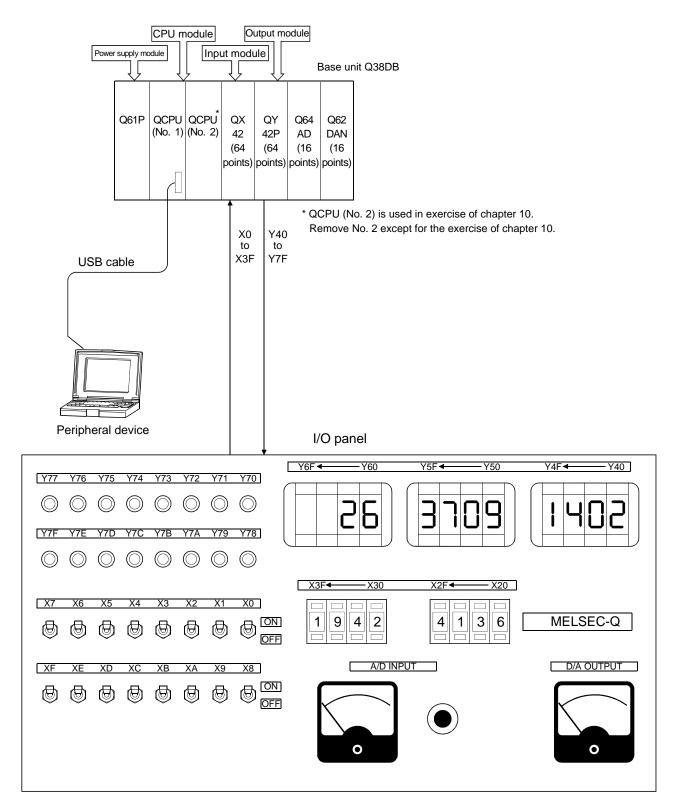
Refer to appendix 2 about the ladder program creation method in GX Works2.

 MEMO		

CHAPTER 5 GX Works2 BASIC OPERATIONS (PART 1: SINGLE PROGRAM)

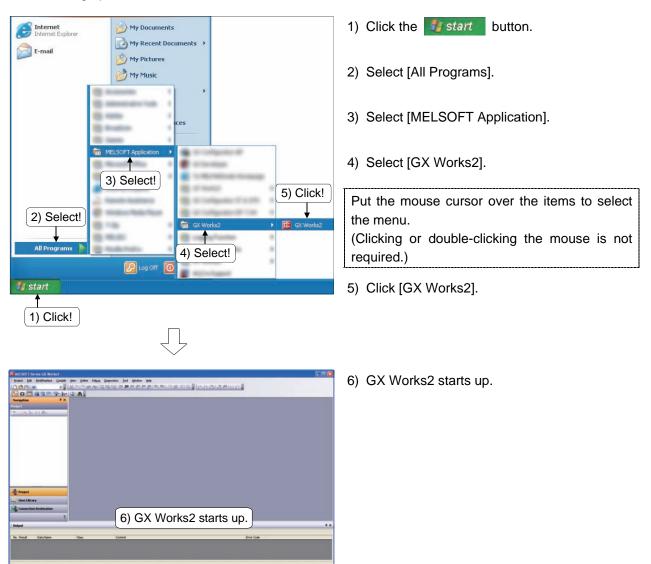
5.1 System Configuration of Demonstration Machine

The following figure shows the system configuration to be used in the exercise.



5.2 Basic Operation 1 (Operation Before Creating Ladder Programs)

5.2.1 Starting up GX Works2

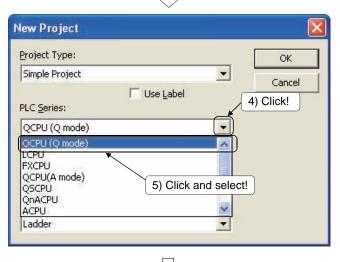


5.2.2 Selecting programmable controller type and project type (creating a new project)



1) Click \square on the toolbar or select [Project] \rightarrow [New Project] (Ctrl + N).

New Project 2) Select! Project Type: Simple Project -Cancel Simple Projec Structured Project * 3) Click and select! QCPU (Q mode) PLC Type: Q06UDH -Language: Ladder -



(To the next page)

- 2) Click the "Project Type" list button.
- The "Project Type" list is displayed. Select "Simple Project".

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

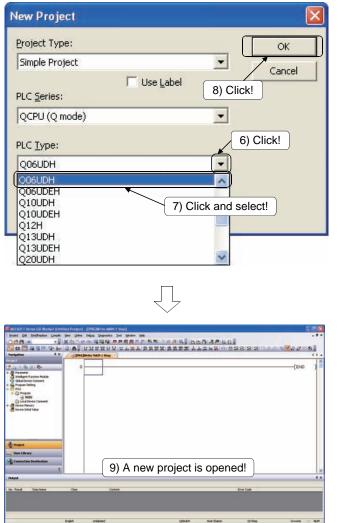
Structured project:

The structured project creates programs with a structured programming.

- 4) Click the "PLC Series" list button.
- 5) The "PLC Series" list is displayed. Select "QCPU (Q mode)".

(From the previous page)





- 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.

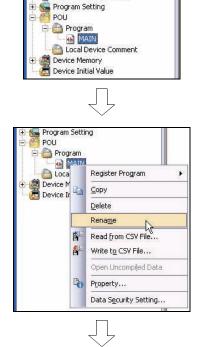
5.2.3 Creating a program

Create a program.

(1) A program to be created

Create a program which displays a counter value counted by a special relay SM411 (0.2-sec. clock) of a programmable controller CPU on the LED of the demonstration machine.

- (2) Devices to be used
 - Y40 to Y4F ... For displaying the counter value
- (3) Program Project name Applied 1 Program name PR1 (CO K10 SM411 (0.2-sec. clock) 0 ┥┟ Counts at 0.2-second intervals. SM400 (always ON) Digital display 5 \neg BCD CO K4Y40 Displays the counter value in 7 segments. Y4F to Y40 C0 9 RST C0 } Resets the count. 4 | D i END 14 Displays the current value of C0
 - (4) Changing a program name Change a program name.
- 1) Right-click "MAIN" in the project data list.

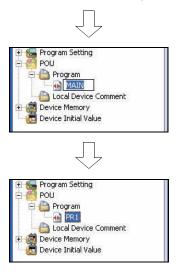


(To the next page)

🚱 Global Device Comment

2) Click [Rename].

(From the previous page)



3) Enter "PR1" as the data name.

4) The program name is changed from "MAIN" to "PR1".

(5) Creating procedure for the program

Create a sequence program with ladder symbols.

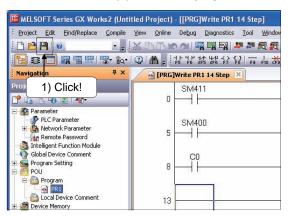
This section explains the creation procedure using the tool buttons.

	REMARK	
	 Right-clicking on the ladder c editing or searching operation Useful operations Insert row: Shift + (Insert) 	reation screen displays a menu for various n. Delete row: Shift + Delete
	Insert column: Ctrl + Inse	
0 Inter Symbol NOTI- Senti		 Click : on the toolbar to open the Enter Symbol window. Enter "SM411". If any other button is pressed by mistake, click the Exit button.
		2) Click the OK button to confirm the entry.
SM411		3) The entered symbol $(\neg \downarrow \vdash)$ is displayed.
0		 4) Click the
		5) Click the OK button.
SM411	(C0 K10)	6) The entered symbol ($_{(C0} \stackrel{K10}{\longrightarrow}$) is displayed.
0 Inter Symbol	(END)	 Click the toolbar, and enter "SM400".
		8) Click the OK button.
	(c0 ^{K10})	9) The entered symbol $(\stackrel{\scriptscriptstyle{SM400}}{\dashv})$ is displayed.
Conversion Bhildes to 4-dgt B Conversion Bhildes to 4-dgt B Conver	(END) (END) (END)	10) Enter "BCD C0 K4Y40".
	Ţ	11) Click the OK button.
(То	the next page)	

(From the previous page)	
SM411 (C0) SM400 [BCD C0 к4/40) I [BCD C0 к4/40) I [END [END)	12) The entered symbol (⊣BCD CO K4Y40⊣) is displayed.
	13) Click the 🔀 button on the toolbar, and enter "C0".
, , , , , , , , , , , , , , , , , , ,	14) Click the OK button.
SM411 (C3) SM400 (8CD C0 K440)	15) The entered symbol $(^{\circ}_{\vdash})$ is displayed.
	16) Click the 🙀 button on the toolbar, and enter "RST C0".
$\overline{\bigcup}$	17) Click the OK button.
9M411 	18) The procedure is finished.
SM400 	
C0 [AST C0]	
	19) Click the 🗾 button on the toolbar to convert the symbol.
0 SM411 (C0 ^{K10})	20) The symbol is converted.
S [BCD C0 K4Y40] C0	
8 (RST C0) 13 (END)	
LEND]	

5.2.4 Saving a project

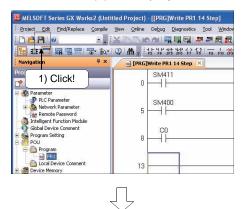
(1) Modified project



1) Click e on the toolbar or select [Project] \rightarrow [Save] (Ctrl + S).

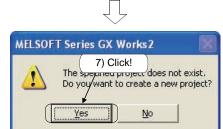
POINT	
	- tion of the project data g the created project, create a folder of "SCHOOL" directly in C drive.

(2) New project



(Only when a newly-created project is saved)

Browse
2) Specify the location to store the project!
4) Set a project name!
3) Set a workspace name!
SCHOOL
Applied 1
e as necessary!
Switch the window bu-dickide this button prmat Project when you want to ((MELSOFT Navigato (MELSOFT Navigato



POINT

Workspace

Workspace enables GX Works2 to manage several projects with one name.

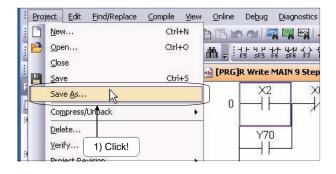
- When the save destination exists When the save destination (workspace and project) exists, the folder where the workspace is saved can be specified in "Workspace/Project List".
 Number of the characters for a workspace name, project name, and title Specify a workspace name, project name, and title 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.

1) Click \square on the toolbar or select [Project] \rightarrow [Save] (Ctrl + S).

- 2) Specify the location to store the project.
- Set a workspace name.
 (Set the name to "SCHOOL".)
- 4) Set a project name.(Set the name to "Applied 1".)
- 5) Set a title as necessary.
- 6) Click the Save button to accept the entry.

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

5.2.5 Saving a project with another name



Save Location:	
C:\SCHOOL\SCHOOL	Browse
Workspace/Project List:	†
Project	PLC Type 2) Specify the location to
E	store the project! space list
C Applied 1	Q06UDH
	4) Set a project name!
	3) Set a workspace name!
<u>W</u> orkspace Name:	SCHOOL
Project Name:	Applied 2
fitle:	
✓ Include revisions	
	e as necessary!
5) 0 - +	



1) Click [Project] \rightarrow [Save as].

- 2) Specify the location to store the project.
- Set a workspace name.
 (Set the name to "SCHOOL".)
- 4) Set a project name.(Set the name to "Applied 2".)
- 5) Set a title as necessary.
- 6) Click the Save button to accept the entry.

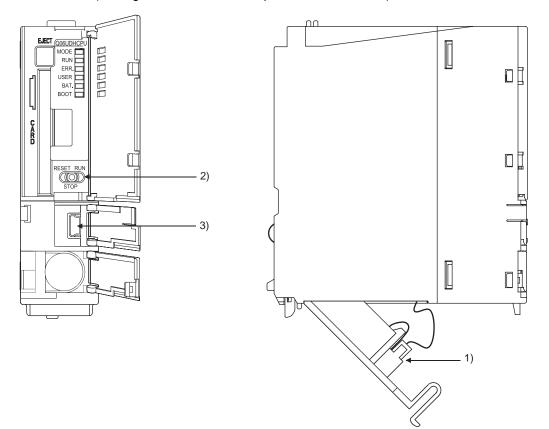
7) Click the Yes button. The new project is saved. 5.3 Basic Operation 2 (Preparation for CPU Operation)

Write the program created in section 5.2.3 to the CPU after the preparation of setting switches and internal clock. Operate the program after writing to monitor and test.

5.3.1 Preparations for starting up CPU

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

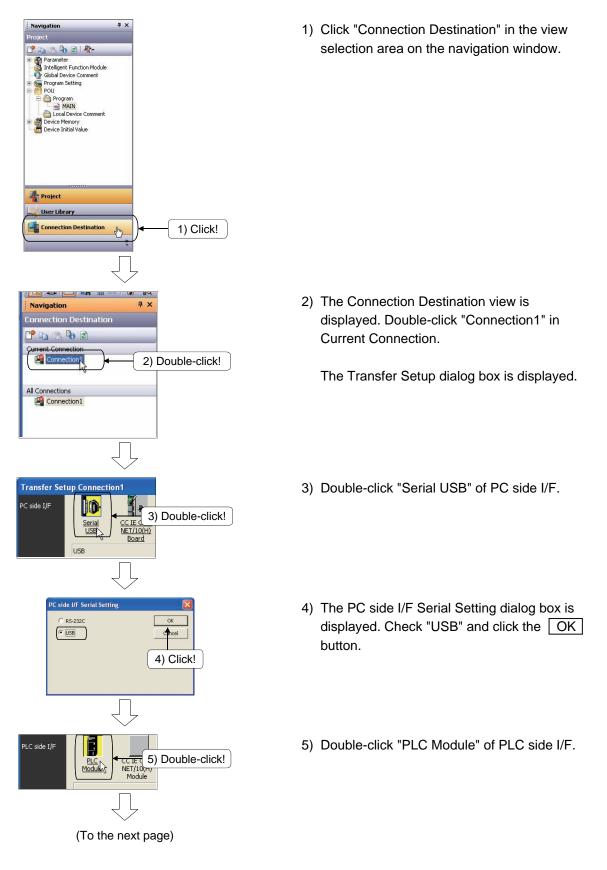
Connect or set the connectors and the switches of (1) to (3) shown below. (The figures below are example of Q06UDHCPU.)

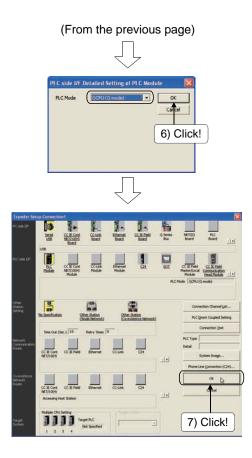


- Connecting a battery Connect the battery since the lead wire of the battery connector is disconnected at the factory shipment.
- (2) Setting the switches Set the RUN/STOP/RESET switch to the STOP position.
- (3) Connecting the USB cable

(4) Setting the connection destination

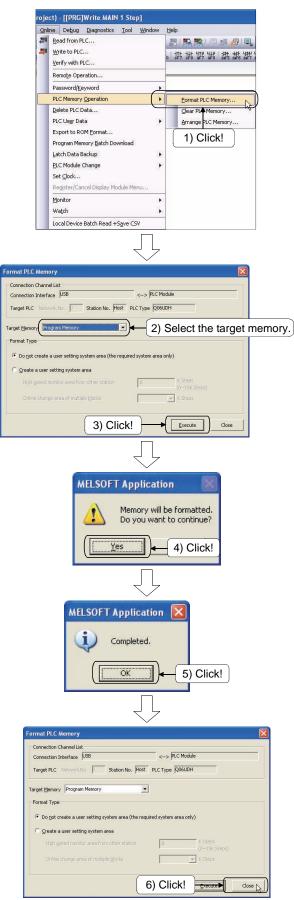
This section explains how to set the connection destination for accessing the programmable controller CPU.





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



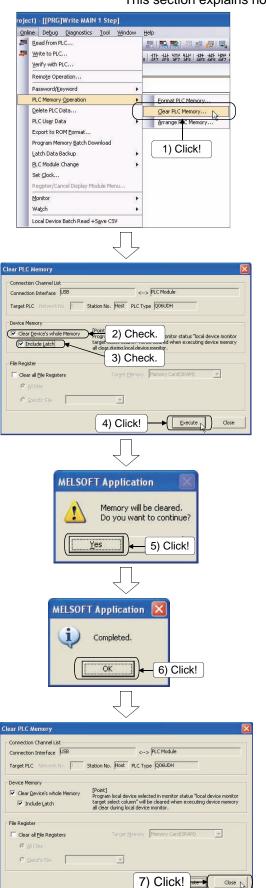
 Click [Online] → [PLC Memory Operation] → [Format PLC Memory].

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

4) Click the Yes button to start formatting.

- 5) When format is completed, the dialog box on the left is displayed. Click the OK button.
- 6) Click the Close button to close the dialog box.

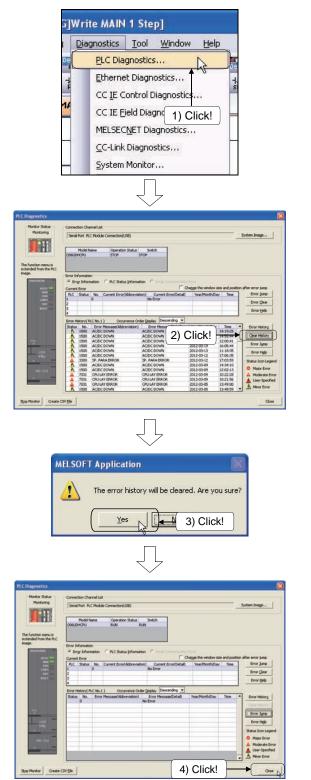
(6) Clearing all the device memory from the CPU This section explains how to clear the device memory of the QCPU.



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

- 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.
- 5) Click the Yes button to clear the latch device.
- When the clearing the latch device is completed, the dialog box on the left is displayed. Click the OK button.
- 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] \rightarrow [PLC Diagnostics].

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

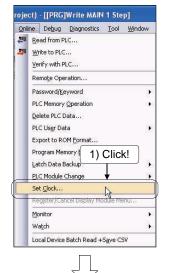
- The confirmation dialog box is displayed. Click the Yes button.
- 4) Click the Close button to close the dialog box.

(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.



<--> PLC Module

3

Get Time from PC

Specify Execution Target

3) Click!

¥

Execute

Close

Station No. Host PLC Type Q06UDH

12

9

2) Enter time!

÷

Connection Channel List

Target PLC

•

Connection Interface USB

 Sun Mon Tue Wed Thu
 Fri
 Sat

 29
 30
 1
 2
 3
 4
 5

 6
 7
 8
 9
 10
 11
 12

 13
 14
 15
 16
 17
 18
 19

 20
 21
 22
 23
 24
 25
 26

 27
 34
 5
 6
 7
 8
 9

C Today: 5/28/2012

5/28/2012

May, 2012 🛛 🕨

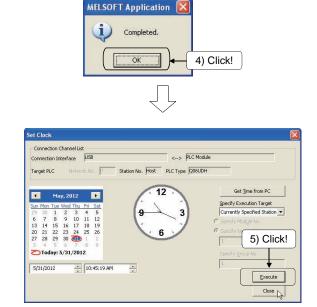
2:51:00 PM

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

- 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.

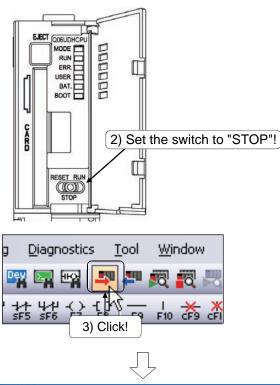
When the time in the personal computer is correct, clock can be set easily by clicking the Get Time from PC button.

- 4) The dialog box on the left is displayed. Click the OK button.
- 5) Click the Close button to close the dialog box.



- 5.4 Basic Operation 3 (Writing Data to Programmable Controller, Monitoring, Modifying Program)
- 5.4.1 Writing data to the CPU

Write the sequence program created in section 5.2.3 to QCPU.



- Suppose that the ladder program (sequence program) has been created in section 5.2.3 with GX Works2 to proceed to the next step.
- 2) Set the RUN/STOP/RESET switch on the CPU to STOP.
- 3) Click and on the toolbar or click [Online] \rightarrow [Write to PLC].

	Bead @ Write	C Yerry	C Delete		_
PLC Module	ligent Punction Module	ecution Target C	 Select pro 	ograms and	
Title			paramete	rs by clicking	a
Edit Data	Parameter+Program	Select All	on data!		9
	mn/Data Name	Title Targ		Larger memory	Size
- Intitled Project)		100			
PLC Data	e Elle)	6	\mathbf{K}	Program Memory/D	
- FR MAIN	n r wej	L L	2012/05/28 15:5	5:05	2176 Bytes
- Parameter	Remote Password/Switch Setting		2012/05/28 15:5	LAR.	464 Dytes
Gobal Device Co		0			404 09103
COMMENT	- 2	9	Decal 2012/05/28 15:5	5:05	
T MAIN			2012/05/28 15:5	106	
Nacasano Calificat	No Setting / Already Set)	Set if it is needed(Free Volume 245,760	5) Click	:! Refresh
Writing Size					
			112 A.M.	Execut	te Close
Writing Size 2,640Bytes	k PLC User Data Wri	te Title Porm		Execut Execut Arrange PLC Memory	Close

(To the next page)

- 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.

(From the previous page	e)
\Box	
Vrite to PLC	
	4/4
	100/100%
Parameter Write : Completed Boot File Write : Completed Remote Password Write : Completed Program (PR1) Write : Completed Write to PLC : Completed	3
Boot File Write : Completed Remote Password Write : Completed Program (PR1) Write : Completed	2
Boot File Write : Completed Remote Password Write : Completed Program (PR1) Write : Completed Write to PLC : Completed	automatically
Boot File Write : Completed Remote Password Write : Completed Program (PR1) Write : Completed	automatically. – 7) Click!
Boot File Write : Completed Remote Password Write : Completed Program (PR1) Write : Completed Write to PLC : Completed	
Boot File Write : Completed Remote Password Write : Completed Program (PR1) Write : Completed Write to PLC : Completed	

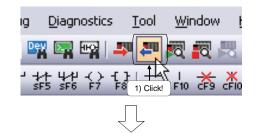
6) The progress dialog box is displayed.

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

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

medion Channel List							
rial Port PLC Module Conn	ection(USB)						System Image
THE REAL PROPERTY AND INCOME.							
	Beed @ Write	C Yerf	Y	C De	lete		
PLC Module 1 1 Inte	Iomt Fonction Nodule	cution Target D	ata()	1 10	Yes)		
				_			
			÷				
Edit Data	Earameter+Program	Select <u>A</u>	Callo	el All Sele	ctions		
	nejData Nane	Title	Target	Detai	Last Change	Target Memory	528
Applied 2							
PLC Data	(mil)			Certif		Program Memory/D	
FBPR1	ire)				2012/05/01 00:4	0.01	2204 Dytes
- D Parameter					2012/00/31 00:4	1999 (64.07 87025
D PLC/Network	Remote Password/Switch Setting		8		2012/05/31 08:4	2002	464 Dytes
Gobal Device Co	reverse	1			and a determined as a		in the second
COMMENT	and the second se	1 1	0	Decal	2012/05/31 08:4	3:03	
- Device Memory				Detai			
E MAIN					2012/05/31 08:4	3:03	
							(
							8) Clic
Necessary Setting	Io Setting / Already Set)	Set & K is needs	KK THE	Sections 1	Heready set 1		(0) 0
Witting Size						Use Volume	
2,660Dytes					243,092	2,660Dytes	Refrest
							_
ted Functions <<						5	ecute Close
the Contract						<u></u>	
	1 111	-	673		111.787	A	
					1		
		10			100 C		

5.4.2 Reading data from the CPU





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

Yes to all

No

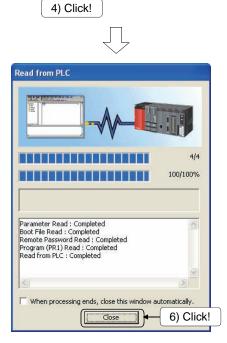
AELSOFT Application

Yes

1

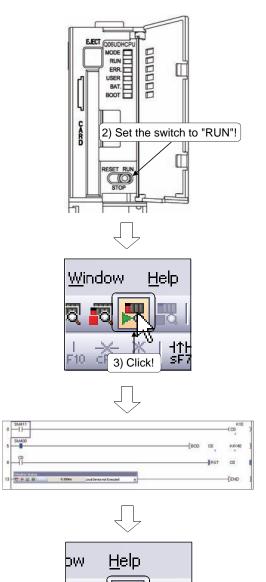
- Click I on the toolbar or click [Online]
 → [Read from PLC].
- From the "PLC Module" tab, click to select the program and parameter to read from the CPU. Or click Parameter + Program to select the target program and parameter.
- 3) Click Execute to accept the selection.

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



- 5) The progress dialog box is displayed.
- 6) The message "Completed" is displayed when the reading is completed. Click Close.

5.4.3 Monitoring

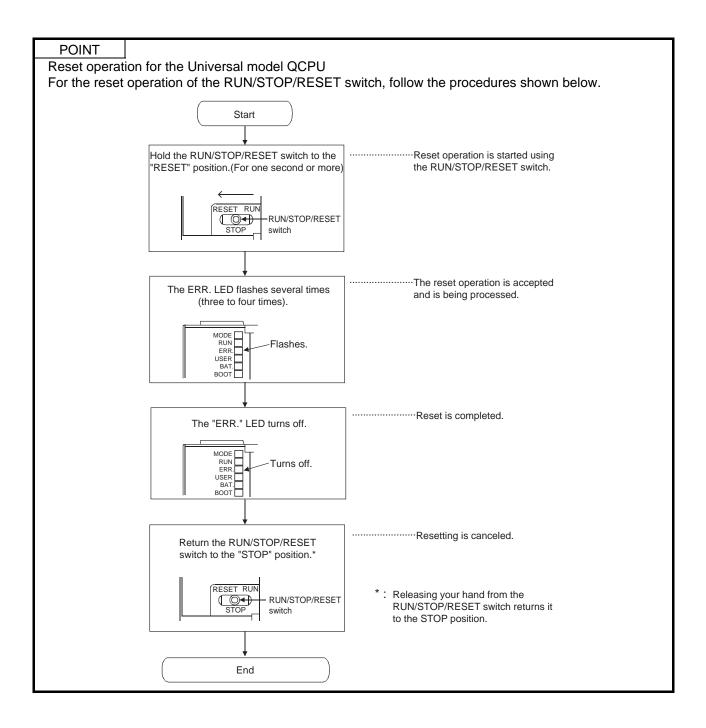


Dev

5) Click!

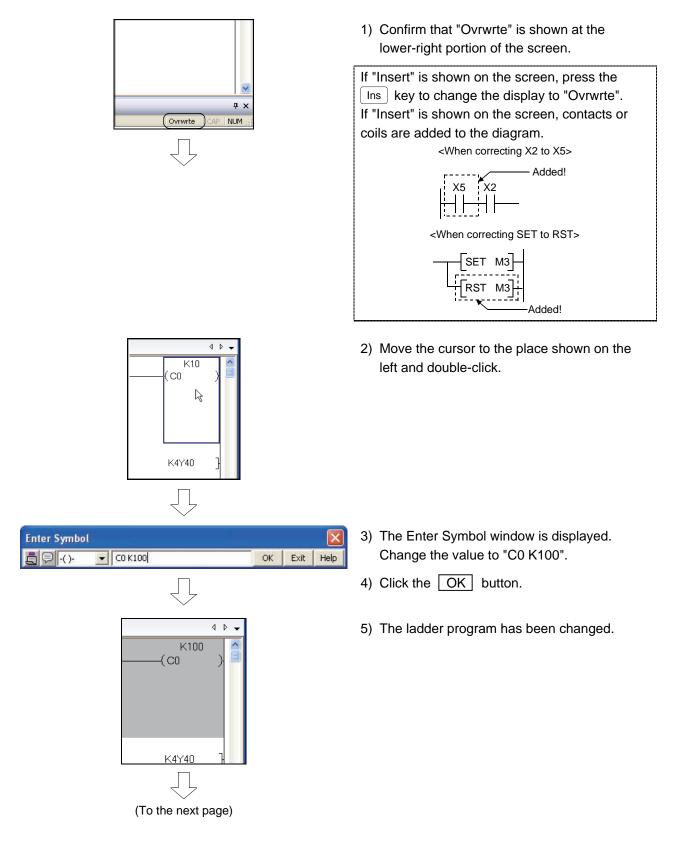
- Suppose that the ladder program (sequence program) has been written into the programmable controller CPU to proceed to the next step.
- Perform the reset operation with the RUN/STOP/RESET switch of the CPU and set the switch to RUN.
 - * For the reset operation, refer to POINT in the next page.
- Click
 [™] on the toolbar or click [Online] → [Monitor] → [Start Monitoring].

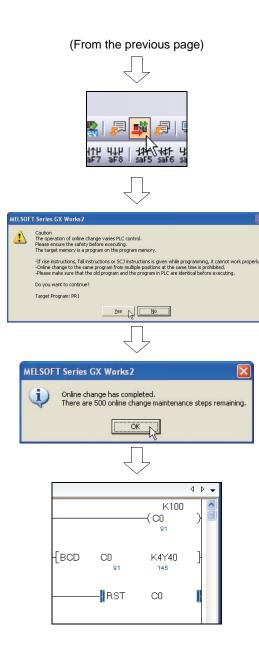
- 4) The Monitor Status dialog box is displayed and the ladder monitor is started.
- To stop the ladder monitor, click
 Image: Boost on the ladder or click [Online] → [Monitor] → [Stop Monitoring].



Modify the program written to the programmable controller CPU in section 5.4.1 while the programmable controller is running.

Modify the setting value K10 to K100 as an example.





- 6) Click the 🗾 button on the toolbar.
- The dialog box on the left is displayed. Click the Yes button to execute the online program change.
- 8) Click the OK button.
- 9) C0 is counted from 0 to 99.The BCD display of the demonstration machine is also counted from 0 to 99.
- 10) Monitor the program.

 MEMO		
)

CHAPTER 6 FILE-BASED MANAGEMENT AND PROGRAM EXECUTION MANAGEMENT

6.1 File-Based Management

In the ACPU, all processes are developed in one program. In the QCPU, a program is composed of several files, which enables the program development divided by functions and processes for several developers.

- (1) Advantages of file-based management
 - (a) Programs can be stored to a memory in file basis. Therefore, even when a file is added or changed corresponding to a partial change of a program, other files are not affected.
 - (b) Names and time can be added to programs and data for management.
 - (c) Each file can be write-protected.

(2)	File	type	and	the	data	to b	e store	ed

The following	table	shows	the	types	of	data	which	can	be	stored	in	each	
memory.													

	CPU r	module built-in m	nemory	Memory card (RAM)	Memory ca	ard (ROM)	File name and		
Item	Program memory	Standard RAM	Standard ROM	SRAM card	Flash card	ATA card	extension	Remarks	
	Drive 0 ^{*1}	Drive 3 ^{*1}	Drive 4 ^{*1}	Drive 1 ^{*1}	Drive	e 2 ^{*1}			
Parameter	0	×	0	0	0	0	PARAM.QPA	1 data/drive	
Intelligent function module parameter ^{*2}	0	×	0	0	0	0	IPARAM.QPA	1 data/drive	
Program	۲	×	0 ^{*3}	0 ^{*4}	0*4	0*4	***.QPG	-	
Device comment	O ^{*5}	×	0 ^{*6}	O ^{*6}	O ^{*6}	0*6	***.QCD	-	
Device initial value	0	×	0	0	0	0	***.QDI	-	
Device data	×	×	0	×	×	×	***.QST	-	
File register	×	0*7*8	×	0	O ^{*9}	×	***.QDR	-	
Local device	×	0*7	×	0	×	×	***.QDL	1 data/CPU module	
Sampling trace file	×	0*7	×	0	×	×	***.QTD	-	
Error history data	×	×	×	×	×	×	***.QFD	-	
Device data storage file	×	×	0	×	×	×	DEVSTORE. QST	-	
Module error collection file	×	0	×	×	×	×	IERRLOG. QIE	-	
Backup data file	×	×	×	0	0	0	MEMBKUP0. QBP	-	
Programmable controller user data	×	х	0	×	×	0 ^{*10}	*** ***	-	
User setting system area	0	×	×	×	×	×	-	-	

•: Required, O: Storable, X: 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 PLC parameter.

*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.

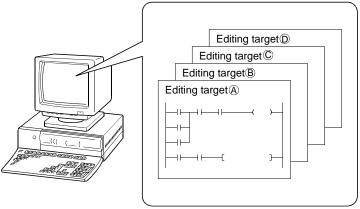
*10: Data can be written or read with the following instructions.

• 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.

(3) Editing multiple programs at the same time In GX Works2, multiple programs can be edited at the same time.



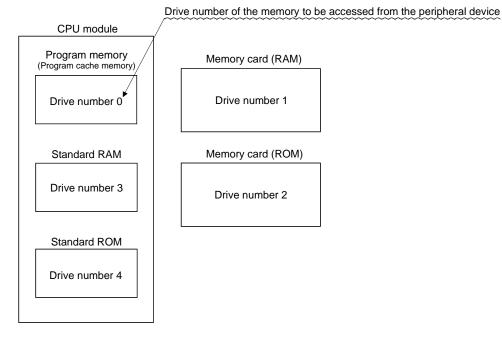
Multiple programs can be edited at the same time.

6.1.1 Built-in memory and IC memory card

Memories to store files include the following two types; built-in memory of the QCPU and memory card.

The built-in memories include the following three types; program memory, standard RAM, and standard ROM.

Memory cards include the following three types; SRAM card, Flash card, and ATA card.



The tables on the next page list the memory capacities and necessity of formatting of each memory in the QCPU.

		Q00UJCPU	Q00UCPU	Q01UCPU	Q02UCPU	Q03UD(E)0	CPU	Form	atting	
Program		10K	steps	15K steps	20K steps	30K step	s	*1		
memory		(40K	byte)	(60K byte)	(80K byte)	(120K byt	e)	1		
Standard	ROM	256K byte		512K byte 1024K byte				Unnece	ssary	
Standard	RAM	-		128K byte		192K byt	:e *	*1		
					Q2MEM-	1MBS: 1M byt	e			
	SRAM				Q2MEM-	2MBF: 2M byt	e N	Necessa	ary (use	
	card		-		Q3MEM-	4MBS: 4M byt	e (GX Wor	ks2.)	
					Q3MEM-	8MBS: 8M byt	e			
Memory	Flash				Q2MEM-	2MBF: 2M byt	e ,			
card	card		-		Q2MEM-	4MBF: 4M byt	e	Unnecessary		
	A.T.A				Q2MEM-	8MBA: 8M byt	e			
	ATA		-		Q2MEM-1	6MBA: 16M b	vte.		ary (use	
	card				Q2MEM-3	2MBA: 32M b	yte C	GX Wor	KSZ.)	
		Q04UD(E)	Q06UD(E)	Q010UD(E)	Q13UD(E)	Q20UD(E)		JD(E)	Formatting	
		HCPU	HCPU	HCPU	HCPU	HCPU	HC	PU		
Program		40K steps	60K steps	100K steps	130K steps	200K steps		steps	*1	
memory		(160K byte)	(240K byte)	(400K byte)	(520K byte)	(800K byte)	(1040ł	K byte)	•	
Standard	ROM	1024	K byte	2048	K byte	4096	< byte		Unnecessar	
Standard	RAM	256K byte	768K byte	1024	K byte	1280	K byte		*1	
				BS: 1M byte				Necessary		
	SRAM			Q2MEM-2M	BS: 2M byte				(use GX	
	card			Q3MEM-4M	BS: 4M byte					

	card	Q3MEM-4MBS: 4M byte Q3MEM-8MBS: 8M byte	(use GX Works2.)		
Memory card	Flash	Q2MEM-2MBF: 2M byte	Unnecessary		
	card	Q2MEM-4MBF: 4M byte			
	ATA card	Q2MEM-8MBA: 8M byte	Necessary		
		Q2MEM-16MBA: 16M byte	(use GX		
		Q2MEM-32MBA: 32M byte	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.

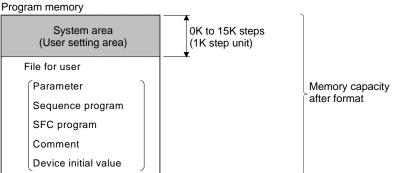
(1) Built-in memory

This section explains a memory map of the program memory built in a QCPU and capacities of the built-in memories (program memory, standard RAM, and standard ROM).

Memory map

The program memory stores files as follows.





Use the user setting area of a system file for a communication with serial communication modules and for monitoring from other station on the network.

Registering the user area enables a high-speed monitoring from other station on the network through the serial communication. (Refer to section 8.1.3 (2))

POINT

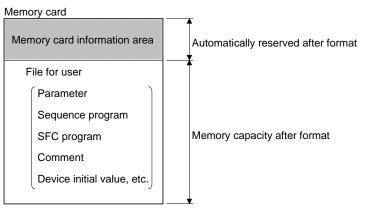
- (1) When using a QCPU for the first time after purchase, format the program memory and standard RAM.
- (2) A program is stored in units of file which is equivalent to a step consisting of 4 bytes.
- (3) Up to 124 programs can be executed in a QCPU.
 - Capacities and the number of storable files of the program memory, standard RAM, and standard ROM Refer to section 3.1.

(2) Memory card

This section explains a memory map and capacity of memory cards for a QCPU.

Memory map

A memory card stores files as follows.



POINT

- (1) When using a memory card (SRAM, ATA) for the first time after purchase, format it with a peripheral device.
- (2) A program is stored to the IC memory card in units of file which is equivalent to 128 steps individually consisting of 512 bytes.

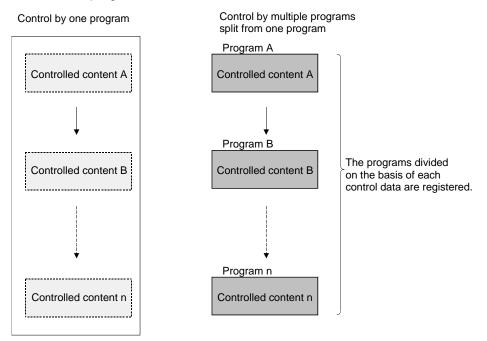
• Capacities and the number of storable files of memory cards

Memory card model name	Memory capacity	Number of storable files				
Q2MEM-1MBS	1011.5K byte	255				
Q2MEM-2MBS	2034K byte	287				
Q3MEM-4MBS	4078K byte	319				
Q3MEM-8MBS	8172K byte	319				
Q2MEM-2MBF	2035K byte	288				
Q2MEM-4MBF	4079K byte	288				
Q2MEM-8MBA	7982K byte	512				
Q2MEM-16MBA	15982K byte	512				
Q2MEM-32MBA	31854K byte	512				

- 6.2 Program Execution Management
- 6.2.1 Description of program execution type

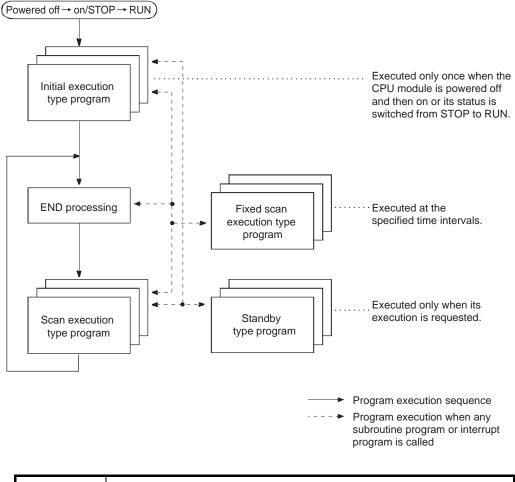
The program execution in the QCPU can be managed in two methods; by integrating controlled contents into one program (conventional method) and by dividing a program into multiple programs by controlled contents.

When dividing a program into multiple programs, set "execution type" in the program setting of the PLC parameter setting in GX Works2 to specify the execution mode for each divided program.



The QCPU executes each divided program according to each execution type; "Initial execution type", "Scan execution type", "Standby type", and "Fixed scan execution type" set by parameters.

The following figure shows the program sequence after the QCPU is powered on or its status is changed from STOP to RUN.

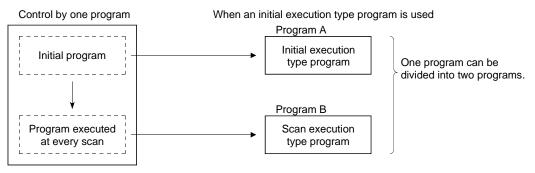


POINT

Use the initial execution type program, standby type program, and the fixed scan execution type program as necessary.

6.2.2 Initial execution type program

- (1) Initial execution type program
 - (a) Initial execution type program is executed only once when the CPU is powered on or its operating status is changed from STOP to RUN.
 - (b) Set the execution type to "Initial" in the program setting of the PLC parameter.
 - (c) Initial execution type program is available for the program which is not necessary to be executed from the next scan after executed once.

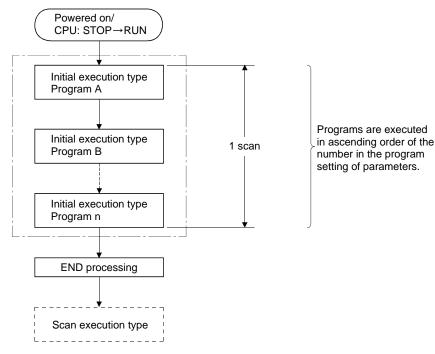


(2) Execution of multiple initial execution type programs

When two or more initial execution type programs exist, the programs are executed in ascending order of the numbers in the program setting of the PLC parameter.

(3) END Processing

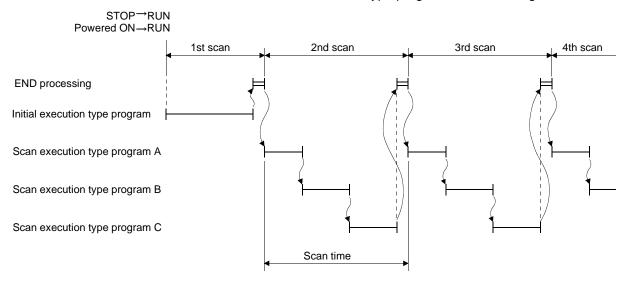
After all initial execution type programs are executed, the END processing is executed. In the next scan or later, "scan execution type program" is executed.



6.2.3 Scan execution type program

- (1) Scan execution type program
 - (a) Scan execution type program is executed once for every scan after the scan in which the initial execution type program is executed.
 - (b) Set the execution type to "Scan" in the program setting of the PLC parameter.
- (2) Execution of multiple scan execution type programs When two or more scan execution type programs exist, the programs are executed in ascending order of the numbers in the program setting of the PLC parameter.
- (3) END Processing

After all scan execution type programs are executed, the END processing is executed and the first scan execution type program is executed again.

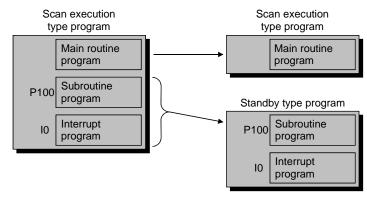


When constant scanning is set
 When the constant scanning is set, the scan execution type programs are executed according to the set constant scan time.
 (Refer to section 8.3.1)

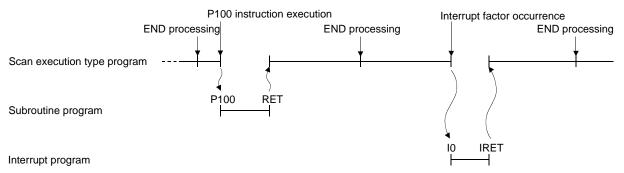
6.2.4 Standby type program

- (1) Standby type program
 - (a) Standby type program is executed only when its execution is requested.
 - (b) Standby type program has the following applications.
 - 1) Program library
 - 2) Program type change
- (2) Program library
 - (a) Standby type program is used for managing subroutine programs and interrupt programs separately from a main routine program.

Multiple subroutine programs and interrupt programs can be created and managed in a single standby type program.



(b) After the execution of a standby type program, the CPU module re-executes the program that called a program in the standby type program. The following figure shows the operation when the subroutine and interrupt programs in the standby type program are executed.



(3) Program type change

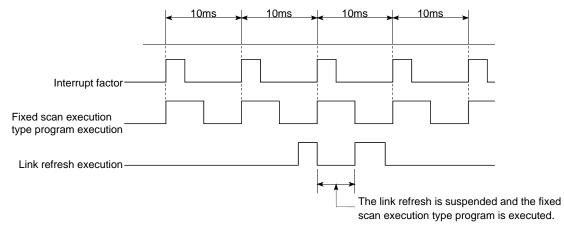
Standby type program is used to create and store programs available in all systems. Only required programs will be executed.

A program set as a standby type program in the PLC parameter can be changed to a scan execution type program and executed in the sequence program.

Use the PSCAN, PSTOP, and POFF instructions to change a program execution type in the QCPU.

6.2.5 Fixed scan execution type program

- (1) Fixed scan execution type program
 - (a) Fixed scan execution type program is an interrupt program executed at specified time intervals.
 Interrupts in units of files are available without interrupt pointers or the IRET instruction.
- (2) Execution of fixed scan execution type program
 - (a) When two or more fixed scan execution type programs exist, a fixed scan execution type program which reaches the specified time is executed.
 When two or more fixed scan execution type programs reach the specified time at the same timing, the programs are executed in ascending order of the numbers in the program setting of the PLC parameter.
 - (b) When a fixed scan execution type program and an interrupt program (I28 to I31) reach the specified time at the same timing, the interrupt program is executed first.
 - (c) When the execution condition is established during the network refresh, the network refresh is interrupted and an interrupt program is executed. Therefore, note that even though the "Block data assurance per station" setting is set in the CC-Link IE or MELSECNET/H network, this setting is invalid when a device set as a refresh target is used in the fixed scan execution type program.



(d) Execution during END processing

When the execution condition is established during the constant scan execution or the waiting time of the END instruction, a fixed scan execution type program is executed.

(3) High-speed execution setting and overhead time of fixed scan execution type program

When fixed scan execution type programs are executed, the following processes are executed.

· Saving and restoring the index register data

· Saving and restoring block numbers of the file register in use

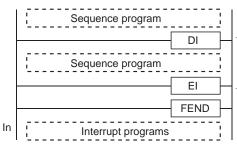
Selecting "High Speed Execution" of Interrupt Program / Fixed Scan Program Setting in the PLC system setting of the PLC parameter does not execute the processes above.

Therefore, the overhead time of the fixed scan execution type programs can be shortened.

For details, refer to the QnUCPU User's Manual (Function Explanation, Program Fundamentals).

The EI instruction is used to clear the interrupt disable state resulting from the execution of the DI instruction, and to create a state in which the interrupt program specified by the interrupt pointer number certified by the IMASK instruction can be executed.

When the IMASK instruction is not executed, I32 to I47 are disabled.



Even if an interrupt factor occurs during the execution of the sequence program between the DI and EI instructions, execution of the interrupt program is suspended until the processing of the sequence program between the DI and EI instructions is completed.

POINT

Terms

- (1) DI
 - The DI instruction disables the execution of an interrupt program until the EI instruction has been executed, even if a start cause for the interrupt program occurs.
 - The program enters a DI state when the power is turned on or the CPU module is reset.
- (2) MASK

The IMASK instruction enables or disables the execution of the interrupt program marked by the specified interrupt pointer by using the bit pattern of 16 points from the device specified by the start number (BIN 16 bits) of the device where the interrupt mask data is stored.

- 1 (ON) ... Interrupt program execution enabled
- 0 (OFF) ... Interrupt program execution disabled

Refer to the MELSEC-Q/L Programming Manual Common Instruction for more details.

CHAPTER 7 GX Works2 BASIC OPERATIONS (PART 2: MULTIPLE PROGRAMS)

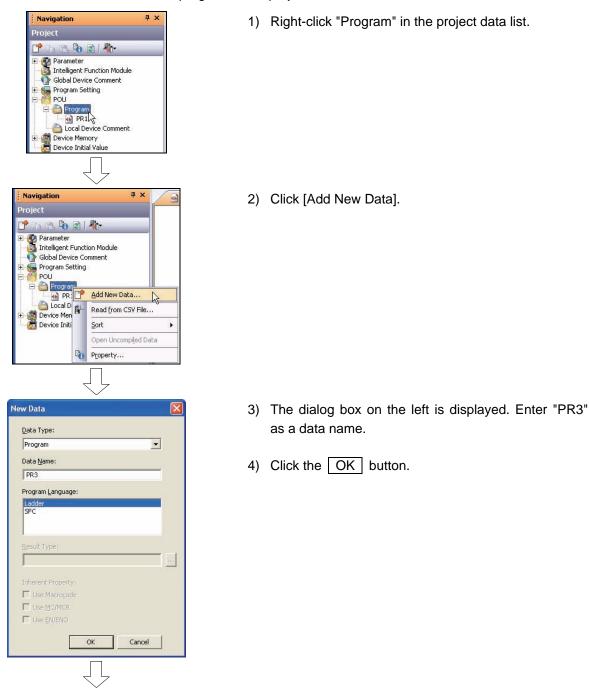
7.1 Multiple Programs

Execute the sequence program created by separate files according to the order set by parameters.

7.1.1 Creating multiple programs

Create multiple programs in the ladder mode.

(1) Procedure for adding program Add a new program in the project.



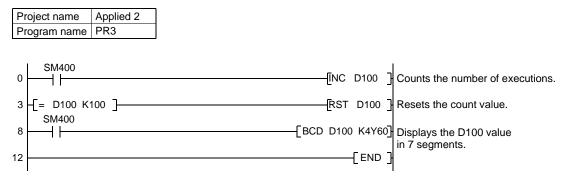
(To the next page)

(From the previous pag	ge)
Navigation	Ψ×
Project	
📑 🗈 🔁 🖬 🛃 👫	
Parameter Intelligent Function Module Global Device Comment Program Setting POU Program PR1 PR3 Local Device Comment Device Memory Device Initial Value	

5) The program "PR3" is newly created.

- (2) Creating a fixed scan execution type program
 - (a) A program to be created Create a program which counts the number of its executions and displays the count value on the 7-segment display.
 - (b) Devices to be used
 - D100-------Register for counting the number of program executions
 - Y60 to Y6F 7-segment display of the D100 value
 - (c) Program

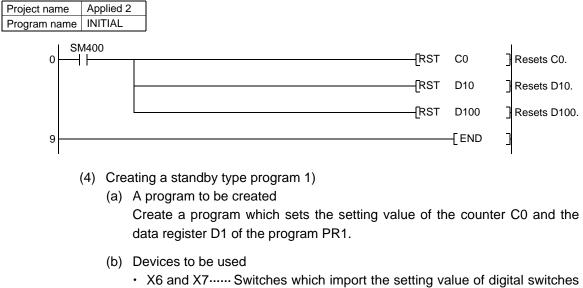
For the creating procedure of a program, refer to section 5.2.3.



(3) Creating an initial execution type program

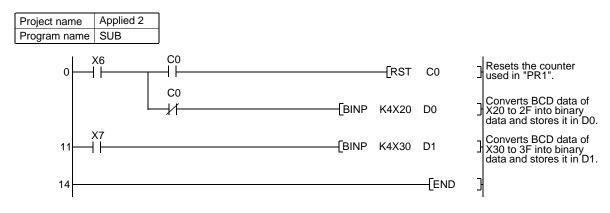
- (a) A program to be created
 Create a program which initializes the register used for PR2 and PR3.
- (b) Devices to be used
 - · C0, D10, and D100 ... Targets to be initialized
- (c) Program

For the creating procedure of a program, refer to section 5.2.3.



- when the program is changed from the standby type to the scan execution type
- Commune Resets the counter used in "PR1"
- D0 and D1 Registers for storing binary data
- (c) Program

For the creating procedure of a program, refer to section 5.2.3.



- (5) Creating a stand by type program 2)
 - (a) A program to be created
 Create a program which counts the number of its executions.
 - (b) Devices to be used
 - D0------ Register for counting the number of program executions
 - Y50 to Y5F 7-segment display of the D10 value
 - P0 ------ Pointer for loop
 - (c) Program

For the creating procedure of a program, refer to section 5.2.3.

Project Prograr		Appli PR2	ed 2	\square										
Flogial	II Hallie	FNZ]										
	0[<	D1	0	K10000)]					-[INC	D10	נ	Counts the number of executions.
	5 =	D1	0	K10000)]					-[RST	D10	ב	Resets the count value (D10).
	10	SM400								-[BCD	D10	K4Y50)]	Displays the D10 value in 7 segments.
for loop	13	SM400								-[моv	K0	D11]	Resets the register for loop (D11).
P0	16	SM400									-[INC	D11]	Counter of register for loop
	20 -	< D'	11	K1000]					-[CJ	P0]	Condition jump for loop
	25											-END)]	
				(b) (c)	sca De •) •)	an execut vices to b X4 X5 Sgram	e use e use e e fi	be. d Switch w executio Switch w rom Y40	/hich shift n type /hich shift	s the p s the 7	orogra 7-segr	m type	e to th	/ (for 4 digits
	ect nam gram na		oplied R1	12										
	0 5	×4 - -{>	D0	K0			SM4 ²	11		[PS0	CAN "	DQ		ches SUB to the execution type.
	13	X5 —↓∕F							[BCD	C0	K4Y4	0	Displ	ays the counter segments.
	16	X5 -							—_[BCD	D1	K4Y4	• ·	Displ	ays the setting value "SUB" in 7 segments.
	19)							[RS	г Со		Rese	ts the counter.
	24										[E	ND]	

7.1.2 Creating programs for control

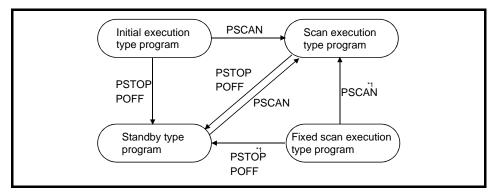
Create programs which control the operating status of the programs created in previous pages.

(1) Program control instructions

Program control instructions change the execution type of the programs while a programmable controller CPU is running.

Program control instructions are the following three types.

- PSCAN instruction
- POFF instruction
- PSTOP instruction



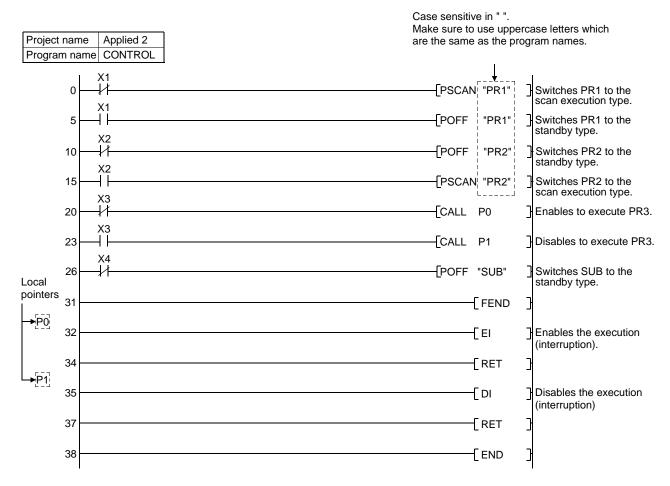
*1: Once the fixed scan execution type program is changed to other execution type, it cannot be returned to the fixed scan execution type.

The POFF instruction is used to securely turn off the external output (Y) when the program is changed to the standby type.

POINT Program execution type changes by program control instructions are executed to the program which is read in the program memory. The program execution type of the program which is not read in the program memory from the memory card cannot be changed.

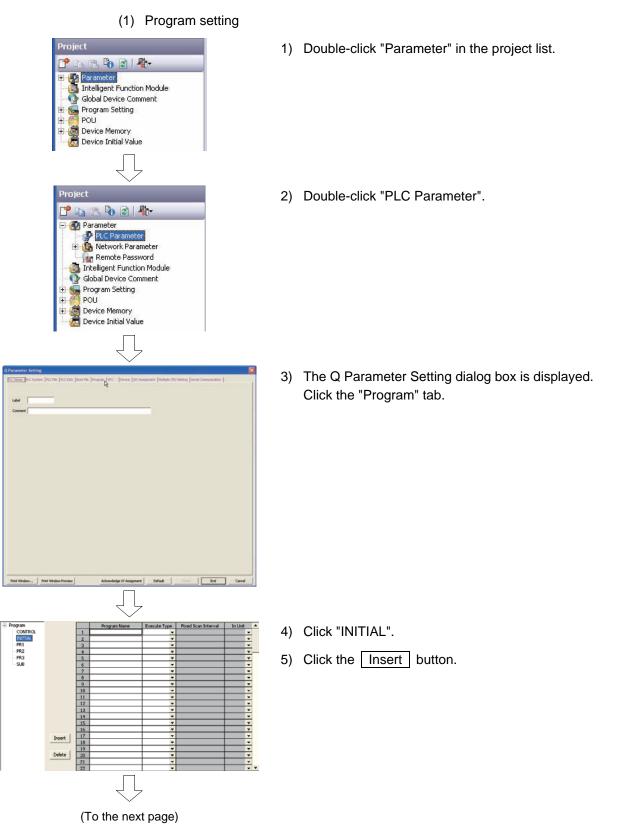
- (2) Programs for control
 - (a) A program to be created
 - 1) Turning on the switch X1 switches PR1 from the scan execution type to the standby type.
 - 2) Turning on the switch X2 switches PR2 from the standby type to the scan execution type.
 - Turning on the switch X3 disables the PR3 execution. (The EI instruction must be executed first to execute the fixed scan execution type program.)
 - 4) Turning on the switch X4 switches SUB from the standby type to the scan execution type.

- (b) Devices to be used
 - X1 (ON)...... Switches PR1 to the standby type
 - (OFF)..... Switches PR1 to the scan execution type
 - X2 (ON)...... Switches PR2 to the scan execution type
 - (OFF)..... Switches PR2 to the standby type
 - X3 (ON)..... Disables PR3 execution
 - (OFF)..... Enables PR3 execution
 - X4 (ON)...... Switches SUB to the scan execution type (OFF)..... Switches SUB to the standby type
 - P0 and P1.... Local pointers
- (c) Program



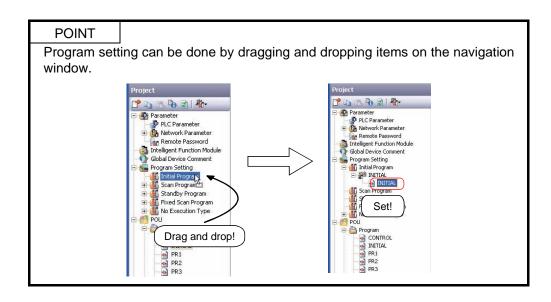
7.1.3 Setting parameters

Set parameters which control the created sequence programs and write them with the programs to the programmable controller CPU.



7 - 7

(From the previous page)	
⊡ Program Program Name Execute Type Fixed Scan Interval In Unit -CONTROL 1 INITIAL Scan ▼ ▼ -INITIAL 2 ▼ ▼ ▼ ▼ -FR1 3 ▼ ▼ ▼ -FR2 4 ▼ ▼ ▼	 Click the second cell of the Program Name column.
5,6 6 · · · · ·	7) Click "SUB".
	8) Click the Insert button.
Program Name Execute Type Fixed Scan Interval In Unit 1 INITIAL Scan • 2 SUB Scan • 3 CONTROL Scan • 4 PR1 Scan • 5 PR2 Scan • 6 PR3 Scan •	 9) Perform the step 6) to 8) repeatedly to set the following contents. a) The third cell → "CONTROL" b) The forth cell → "PR1" c) The fifth cell → "PR2" d) The sixth cell → "PR3"
Program Name Execute Type Fixed Scan Interval In Unit 1 INITIAL Initial • 2 SUB Wait • 3 CONTROL Scan • 4 PR1 Fixed \$San • 5 PR2 Scan •	10) Click 💽 of Execution Type of "INITIAL" and select "Initial".
Program Name Execute Type Fixed Scan Interval In Unit 1 INITIAL Initial • 2 SUB Wait • 3 CONTROL Scan • 4 PR1 Scan • 5 PR2 Wait • 6 PR3 Fixed Scan •	 11) Set the following contents with procedure 10). a) "SUB" and "PR2" → "Wait" b) "PR3" → "Fixed Scan"
Program Name Execute Type Fixed Scan Interval In Unit 1 INITIAL Initial • • 2 SUB Wait • • 3 CONTROL Scan • • 4 PR1 Scan • • 5 PR2 Wait • • 6 PR3 Fixed Scan • 1	12) Enter "1" to "Fixed Scan Interval" of "PR3" and click



(2) PLC system setting

Click the "PLC System" tab on the Q Parameter Setting dialog box. Set "0" to Points Occupied by Empty Slot.

Low Speed 100 ms (Inst-1000ms) High Speed 10.00 ms (0.01ms-100ms)	Common Pointer No. P After (0-4095)
RN X (30-33FFF) FAUSE X (30-33FFF)	Poets Coupled by Empty Slat (*1) 0 Poets System Detempt Settings Detempt Coupled Stat No. C 0-555
tch Data Backup Operation Vald Contact Device Name	Pued Scal Interval 128 100.0 ms (0.5ms-1000ms)
Remote Reset	129 40.0 ms (0.5m-5000ms) 120 20.0 ms (0.5m-1000ms) 121 10.0 ms (0.5m-1000ms)
Previous State Recalculate(Output is 1 scan later)	Interrupt Program / Prived Scan Program Setting
ang fan Albenn Barna Perlan sterna athetis gestore trouble proses	A-PLC Consultability Setting Use special relay / special register from SH(3D 1000
Interrupt Ponter Setting Contemporation Contemporatio Contemporation Contemporation Contemporation Contemporati	Service Processing Setting ho Execute the process as the scan 10 %
	Specify service process time ms (0.2ms-1000ms) Specify service process mesoution counts mesoution counts Times (1-15 Times) Times (1-15 Times)
Setting should be set as same when using multiple CPU.	FLC Module Change Setting FLC Module Change Setting

After the setting of (1) and (2), click the End button and click [Project] \rightarrow [Save] to save the contents.

(3) Writing the data to the programmable controller CPU

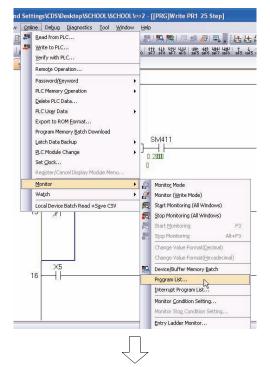
Write the created programs (PR1, PR2, PR3, INITIAL, SUB, and CONTROL) and parameters to the programmable controller CPU. (Refer to section 5.4.1)

7.2 Monitor

7.2.1 Program list monitor

Monitor the execution condition of the multiple programs ("Applied 2") created in section 7.1.

Before monitoring, reset and run the CPU.



1) Click [Online] \rightarrow [Monitor] \rightarrow [Program List].

- Program List Monitor X Entire Scan Time Detail of Scan Time for Scan Execution Program(ms) 0.200 Monitoring Time(ms) Total Scan Time(ms) Scan 200 0.500 END Processing Time(ms) 0.300 Initial 0.200 Low Speed Program(ms) 0.000 0.000 Constant Wait(ms) 0.000 Low Speed Execution Status of Programs Execution Scan Time(ms) Execution Count 🔺 Program 1 SUB Standby 0.000 2 22553 CONTROL 0,200 53731 3 Scan 4 PR1 Scan 0.000 51221 PR2 Standby 0.000 13642 5 6 PR3 Fixed Scar 51 8 9 In this case, scan time display does not change because the actual program processing time is fast.
 - The Program List Monitor dialog box is displayed. Turn on or off the switches X1, X2, X3, and X4 to check that the execution statuses change.
 - · The change of execution status
 - X1 : OFF Switches PR1 to the scan execution type
 - : ON Switches PR1 to the standby type
 - X2 : OFF Switches PR2 to the standby type
 - : ON Switches PR2 to the scan execution type
 - X3 : OFF Enables PR3 execution
 - : ON Disables PR3 execution
 - X4 : OFF Switches SUB to the standby type
 - : ON Switches SUB to the scan execution type

(1) "Entire Scan Time"

Displays the total scan time of each program type and monitoring time set in the "PLC RAS" tab of the PLC parameter in the project data list.

- (a) "Monitoring Time"
 Displays each monitoring time of the scan program and initial program.
 When the scan time exceeds these time, a watchdog timer error is displayed on the CPU.
- (b) "Total Scan Time"

Displays the total time of the scan program and initial program each. For the scan program, the END processing time is included.

- (2) "Detail of Scan Time for Scan Execution" Displays the detail of the scan time.
 - (a) "Program"
 Displays the total execution time of scan programs.
 - (b) "END Processing Time" Displays the END processing time.
 - (c) "Constant Wait"
 Displays the waiting time at a constant scan execution.
- (3) "Execution Status of Programs"

Displays the execution status of programs specified in the "Program" tab of the PLC parameter in the project data list.

- (a) "Program"Displays the program names in the input order in the PLC parameter.
- (b) "Execution" Displays the execution type of the programs specified in the PLC parameter.
- (c) "Scan Time"

Displays the actual scan time (current value). The display is "0.000ms" in the program stop (standby) status.

- * The scan time display does not change when the actual program processing time is short.
- (d) "Execution Count"

Displays the number of executions, counted from 0 at the point of starting count. (After reaching 65536, the count returns to 0.) The execution count is stored even after the program stops.

7.2.2 Monitor function

The monitor function reads the statuses of programs and devices in a CPU from GX Works2.

Operation Check with an Example

 Monitoring of indexing devices and comparison instructions <Program>

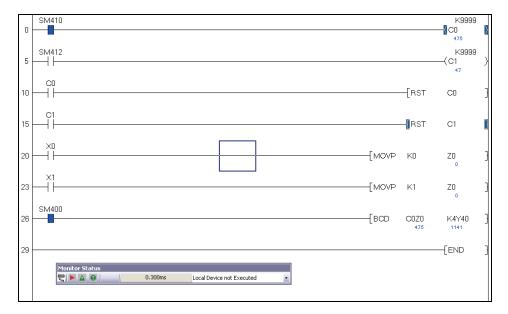
Project name	Applied 3					
Program name	INDEXMOD					
SM41 SM41 SM47 C0 	12	[моч [моч [вср	P K1	<pre></pre>	≻ ₁₉₉ ≻ } } }	

<Operation>

- Read the program above and write it with parameters to the programmable controller.
- Reset and run the CPU.
- Click 🛃 to monitor the program.

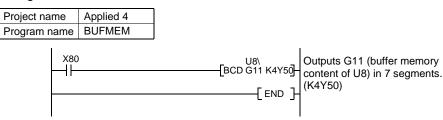
<Description>

When X0 is turned on, indexing is performed as C0Z0 = C (0 + 0) = C0. When X1 is turned on, indexing is performed as C0Z0 = C (0 + 1) = C1. The condition establishment of the comparison instruction can be monitored.



(2) Buffer memory monitor of the intelligent function module

<Program>



<Operation>

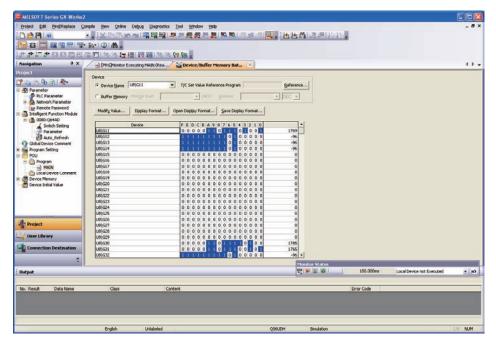
- Read the program above and write it with parameters to the programmable controller.
- Reset and run the CPU.
- Click 🛃 to monitor the program.
- Click [Online] → [Monitor] → [Change Value Format(Hexadecimal)] to change the display format to the hexadecimal notation.
- Click [Online] → [Monitor] → [Device/Buffer Memory Batch].
 Set the contents as "Device Name: U8¥G11", "Monitor Format: Bit and Word", "Display: 16bit Integer" and "Value: DEC" then click [OK].
 (Monitoring is also available in [Watch].)

<Description>

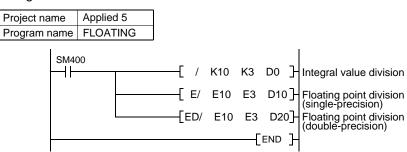
Turn on the READY signal X80 of Q64AD to display the content of the buffer memory address 11 (the value after the A/D conversion) on the 7-segment display with the program and to check that the buffer memory monitor is available with GX Works2.

<Note>

When the A/D conversion value is negative, an operation error occurs by the BCD instruction execution. As the prevention method, adjust the offset value of the A/D converter module.



 (3) Monitoring real number (floating decimal point) data <Program>



<Operation>

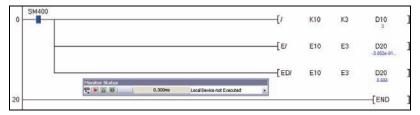
- Read the program above and write it with parameters to the programmable controller.
- Reset and run the CPU.

<Description>

Check that the results of the division are different.

In integral numbers: 10/3 = 3 remainder 1. In floating decimal points: 10/3 = 3.33333...

The data in floating decimal point format can be monitored directly with GX Works2.



Selecting [Online] \rightarrow [Monitor] \rightarrow [Device/Buffer Memory Batch] displays the following display.

Display: 16bit Integer

D0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	3
D1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Display: Real Number (32Bit)

D10	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1	1	3.3333333
D11	0	1	0	0	0	0	0	0	0	1	0	1	0	1	0	1	

Display: Real Number (64Bit)

D20	1	(D	1	0		0	1	0)	1	0	1	0	0	1	1 3.333333333333333333
D21	1		D	1	0		0	1	0		1	0	1	0	0	1	
D22	1	(D	1	0		0	1	0		1	0	1	0	0	1	
D23	0			0	0	0	0	0	0) (0	0	0	0	1	

POINT

The real number data includes the single-precision floating-point data and double-precision floating-point data.

- 1) Two word devices are used for the real number data of the single-precision floating point data.
- 2) Four word devices are used for the real number data of double-precision floating point data.

For how to use the real number data, refer to appendix 9.

CHAPTER 8 FUNCTIONS OF QCPU

8.1 Maintenance and Debug Functions

Maintenance function list

14	Description
Item	Description
Watchdog timer function	Monitors operation delays caused by program error or hardware failure of the CPU.
Self-diagnostic function	Self-diagnoses the CPU module to see whether an error exists or not.
Error history function	Stores the result of self-diagnostics to the memory as error history data.
System display function	Displays the names of the CPU, I/O modules, and intelligent function module and the system configuration of the I/O address from GX Works2.
System protection function	Allows or prohibits writing/reading data to/from each file in the QCPU module.
Password registration function	Restricts operations from the peripheral device to the memory in the CPU.
Remote password function	Selects the restrictions for the external access via Ethernet or serial communication.
LED and LED indicator	Displays operating status of the CPU module with the LEDs on the front of the CPU module or with the LED indicator.
Display of LED	Indicates whether the CPU operation is normal or not.
Display of LED indicator	Displays messages at an error.
Latch data backup to standard ROM function	Backs up latch data such as device data and error history to the standard ROM without a battery.
CPU module change function with memory card	Backs up data in the CPU module to a memory card and restores the backup data to another CPU module.
Module error collection function	Collects errors caused in the intelligent function modules in the CPU module.

The QCPU has useful functions for system maintenance. The following table lists the maintenance functions.

For the details on the functions which require the operation of a peripheral device, refer to the GX Works2 Operating Manual.

Debug function list

The QCPU has useful debug functions. The following table lists the debug function.

Item	Description
Monitor function	Reads the status of programs and devices of the CPU from a peripheral device.
Online change function	Writes programs when the CPU module is in the RUN status.
Execution time measurement function	Displays the processing time of the program which is being executed.
Program monitor list	Displays the processing time of the program which is being executed.
Interrupt program monitor list	Displays the number of executions of interrupt programs.
Sampling trace function	Continuously collects the specified device data with the QCPU at a preset timing.
Device test function	Forcibly changes the current value of a word device or the on/off status of a bit device in a program.
Debug function from multiple	Enables simultaneous debugging by multiple peripheral
peripheral tools	devices.

For the details on the operation method of each function, refer to the GX Works2 Operating Manual.

8.1.1 Self-diagnostic function

This function allows the CPU module to diagnose itself to check for errors.

- This function aims to provide preventive measures and will prevent malfunction of the CPU module.
 When an error occurs at power-on or during the RUN status of the programmable controller, the self-diagnostic function detects and displays the error to stop the programmable controller operation.
- (2) The QCPU stores the detected error as an error code to a special register SD0 and illuminates the ERR.LED. When multiple errors occur, the error code of the latest error is stored to SD0.
- (3) Up to 100 of the latest errors are backed up with a battery-backup even when the power is turned off. Error history can be checked by clicking [Diagnostics] → [PLC Diagnostics] in GX Works2.
- (4) When an error is detected by the self-diagnostic function, select one of the following two CPU operations.
 - Mode that stops CPU module operation
 When an error is detected, the CPU module stops all external outputs of the
 module for which "Error Time Output Mode" is set to "Clear" in "Detailed
 Setting" of the I/O assignment setting of the PLC parameter. (Outputs (Y) in
 the device memory are held).
 Note that the external outputs of the module for which "Error Time Output

Mode" is set to "Hold" are held. (Outputs (Y) in the device memory are held).

 Mode that continues CPU module operation When an error is detected, the CPU module operation executes programs other than the one (instruction) where an error occurred.

Whether to continue or stop an operation can be selected in "Operation Mode When There Is an Error" of the "PLC RAS" tab of the PLC parameter in the project data list.

- 1) Computation Error
- 2) Expanded Command Error
- 3) Fuse Blown
- 4) Module Verify Error
- 5) Intelligent Module Program Execution Error
- 6) File Access Error
- 7) Memory Card Operation Error
- 8) External Power Supply OFF
 - (In the initial setting, "Stop" is set for all errors.)

As an example, when "Module Verify Error" is set to "Continue", the operation is continued from the I/O address before the error.

Error whether to continue or stop an operation can be selected in "Detailed Setting" in the I/O assignment setting of the PLC parameter

- Intelligent function module error
- (5) For error items which can be set to "Stop" or "Continue" in the "PLC RAS" tab of the PLC parameter in the project data list, an error for which "Continue" is set does not stop the operation even when the corresponding error occurs.

- (6) For the following errors, whether to execute an error check can be selected in the "PLC RAS" tab of the PLC parameter in the project data list.
 - 1) Carry Out Battery Check
 - 2) Carry Out Fuse Blown Check
 - 3) Verify Module
 - 4) Check Device Range at Indexing
 - 5) Diagnose Redundant Power Supply System
 - (In the initial setting, all the items set to execute the error check.)

Without the error check, errors are not detected. Therefore, the processing time of the END instruction can be shortened.

8.1.2 System display function

The system configuration (module name, number of I/O occupied points, and I/O address) of the connected host station can be checked from the peripheral device with the system monitor function.

(1) Starting up the system monitor

Click [Diagnostics] \rightarrow [System Monitor].

	Status		Çor	nection	Channe	luit										
C	H.	Monitoring	5	ierial Por	t PLCN	Iodule Connect	ion(USB)							System	Image	
lain Bi	1940								Op	eration to Selected Mod	ule					
M	in Base									Main Base						
	_			-	_	_				Line Looke						
1/0 A	ŧ.	0000 0040 0	000 0090							Slot CPU						
		in the late	-		0 0											
	2		8 8							QOBUDHCPU						
				1												
	125									tailed Information 115		tion 1	Diagnostic	. 1		etar [
	1000			-			_			and a set of the set o						
en le	Ecrimatic	with the second s					Muchile	Informa	Non Lich	(Main Base)						
100	1		Power	Base	11256	Instalad			1.5	11-22-22-22-2	10255	Park	meter	1/0	Network No.	Master
				Type	Sixts	Installed Modules	Status	Base- Skit	Series	Model Name	Point				Station No.	
lase	Module	Base Model Name	Supply									Type	Point	Address		PLC
ase	Module	Main Base	Supply Exist	Q	0	+				Power		Power	Pont	Address	*	PLC
ase	Module	Main Base Extension Base1			0	*	-	CPU	۰. ۹	QOSLIDHICPU	:	Power CPU				
ase	Module	Main Base Extension Base1 Extension Base2			0	•		- CPU 0-0		QOSUDHCPU Empty	•	Power CPU Empty	OPoint	•	•	
950	Module	Main Base Extension Base1 Extension Base2 Extension Base3			0	•		- CPU 0-0 0-1	Q.	QOSLIDHCPU Empty QX42	64Point	Power CPU Empty Input	OPoint 64Point		•	•
ase	Module	Main Base Extension Base1 Extension Base2 Extension Base3 Extension Base4			0	•		- CPU 0-0 0-1 0-2	٩.	QOSLOHCPU Empty QX42 QY429	64Point	Power CPU Empty Input Output	OPoint 64Point 64Point	0000 0040	•	•
ase	Module	Main Base Extension Base1 Extension Base2 Extension Base3			0	•		- CPU 0-0 0-1	Q . Q	QOSLIDHCPU Empty QX42	64Point	Power CPU Empty Input Output	OPoint 64Point	• • 0000 0040 0040	•	•
956	Module	Main Base Extension Base1 Extension Base2 Extension Base3 Extension Base4			0	•		• CPU 0-0 0-1 0-2 0-3 0-4	Q . Q Q	QOSLOHCPU Empty QX42 QY429	64Point	Power CPU Empty Input Output Intelli	OPoint 64Point 64Point	• • 0000 0040 0000		•
956	Module	Main Base Extension Base1 Extension Base2 Extension Base3 Extension Base4 Extension Base5			0	•		• CPU 0-0 0-1 0-2 0-3	9.000	QOSLOHCPU Empty QX42 QY429 Q64AD	64Point 64Point 16Point	Power CPU Empty Input Output Intelli	OPoint 64Point 64Point 16Point	- 0000 0040 0090	•	•
	Module	Main Base Extension Base1 Extension Base2 Extension Base3 Extension Base4 Extension Base5 Extension Base6			0	•		• CPU 0-0 0-1 0-2 0-3 0-4	9.0000	006LDHCPU Empty 0042 01429 064AD 066DAN	64Point 64Point 16Point 16Point	Power CPU Empty Input Output Intell. Intell.	OPoint 64Point 64Point 16Point 16Point	• 0000 0040 0090 •	•	•
Jase	Module	Main Base Extension Base1 Extension Base3 Extension Base4 Extension Base5 Extension Base6 Extension Base6		Q	0	•		* CPU 0-0 0-1 0-2 0-3 0-4 0-4 0-5	9 9999	QOSLDHCPU Empty QX42 QY42P Q64AD Q62DAN Empty	64Point 64Point 16Point	Power CPU Empty Input Output Intell. Intell. Empty	OPoint 64Point 64Point 16Point 16Point 0Point	• 0000 0040 0090 •	•	•
		Main Base Extension Base1 Extension Base3 Extension Base4 Extension Base5 Extension Base6 Extension Base6		Q	0	•		* CPU 0-0 0-1 0-2 0-3 0-4 0-4 0-5	9 9999	QOSLDHCPU Empty QX42 QY42P Q64AD Q62DAN Empty	64Point 64Point 16Point	Power CPU Empty Input Output Intell. Intell. Empty	OPoint 64Point 64Point 16Point 16Point 0Point	* 0000 0040 0090 0090 *	•	
gend O a	na	Main Base Extension Base1 Extension Base3 Extension Base4 Extension Base4 Extension Base5 Extension Base5 Extension Base7 Titlese	Exist	Q	0 Re Error	•		* CPU 0-0 0-1 0-2 0-3 0-4 0-5 0-6	a . a a a a	QOSLEHCPU Empty QN42 QN42P Q642AD Q642AD Q642AN Empty Empty	64Point 64Point 16Point 16Point	Power CPU Empty Input Output Intell. Intell. Empty Empty	OPoint 64Point 64Point 16Point 16Point 0Point 0Point	* 0000 0040 0090 0090 *	• • • •	
gend O a		Main Base Extension Base1 Extension Base3 Extension Base4 Extension Base4 Extension Base5 Extension Base5 Extension Base7 Titlese	East	Q Antonio Modera	ste Error	•		* CPU 0-0 0-1 0-2 0-3 0-4 0-5 0-6	a . a a a a	QOSLEHCPU Empty QN42 QN42P Q642AD Q642AD Q642AN Empty Empty	64Point 64Point 16Point 16Point	Power CPU Empty Input Output Intell. Intell. Empty Empty	OPoint 64Point 64Point 16Point 16Point 0Point 0Point	* 0000 0040 0090 0090 *	• • • •	••••••
	na	Main Base Extension Base1 Extension Base3 Extension Base4 Extension Base4 Extension Base5 Extension Base5 Extension Base7 Titlese	East	Q Antonio Modera	ste Error	•		* CPU 0-0 0-1 0-2 0-3 0-4 0-5 0-6	a . a a a a	QOSLEHCPU Empty QN42 QN42P Q642AD Q642AD Q642AN Empty Empty	64Point 64Point 16Point 16Point	Power CPU Empty Input Output Intell. Intell. Empty Empty	OPoint 64Point 64Point 16Point 16Point 0Point 0Point	* 0000 0040 0090 0090 *	• • • •	•

(2) PLC Diagnostics

Selecting a CPU and clicking the Diagnostics button displays the execution status of the CPU and error history.

	Connec	tion Cha	nnel List –							
Monitoring	Serial	Port PL	C Module C	Connection(USB)					S	ystem Image
	_	Model N	ame	Operation Status	Switch					
	QUOOD	HCPU		RUN R	RUN					
e function menu is tended from the PLC age.	Error In	formatio	n							
MODE	 Erry Eurrent 	or Inforn : Error	nation (PLC Status Informat	tion C Serial Comm		Error ange the window si	ze and posi	tion a	after error jump
RUN ERR.	PLC	Status	No. Cu	urrent Error(Abbreviat	ion) Current Erro	r(Detail)	Year/Month/Da	/ Time		Error Jump
USER	1		0		No Error					Error Clear
	2		R MONON ALON		****			an varanana		Error Gear
	3 4		3 3		2					Error Help
L .		ictory (D	LC No.1)	Occurrence Orc	ler Display Descend	lina 💌				
	Status			essage(Abbreviation)	son Erspicit		Year/Month/Day	Time		Error History
	Δ	7031	CPULAY		CPU LAY ERROR	205007	2012-05-31	13:39:55		Litor history
	A	7031	CPULAY		CPU LAY ERROR		2012-05-31	13:39:10		Clear History
PULL	A	7031	CPULAY	ERROR	CPU LAY ERROR		2012-05-31	13:36:15		
		4100	OPERATI	ON ERROR	OPERATION ERROR	2	2012-05-31	13:26:44		Error Jymp
USB		4100	OPERATI	ON ERROR	OPERATION ERROF	0	2012-05-31	13:18:38		Error Help
	A	3300	SP. PARA	ERROR	SP. PARA ERROR		2012-05-30	16:20:14		Lifer Hgip
-	A	3300	SP. PARA	AERROR	SP. PARA ERROR		2012-05-30	16:19:04		Status Icon Legen
	A	3300	SP. PARA	ERROR	SP. PARA ERROR		2012-05-30	16:18:54		and the second sec
RS-232	A	3300	SP. PARA	AERROR	SP. PARA ERROR		2012-05-30	16:13:19		Major Error
	A	2401	FILE SET	ERROR	FILE SET ERROR		2012-05-29	17:18:17		A Moderate Erro
-	A	2401	FILE SET	ERROR	FILE SET ERROR		2012-05-29	16:46:47		A User-Specified
	A	2401	FILE SET	ERROR	FILE SET ERROR		2012-05-29	16:46:46		Minor Error
		2401	FILE SET		FILE SET ERROR		2012-05-29	16:32:12		

(3) Detailed information of modules

Select a module to check the detailed information.

fonitor Sta	tus	Cor	nection	Channel	List										
	Monitoring	5	ierial Por	t PLC N	lodule Connect	ion(USB)							System	Image	
Main Base Main Bi 1/O Adr.			1	Ú					eration to Selected Mod. Main Base Slot CPU QOBUDHCPU	ke 14 Enforma		Diagnostics			
ase Inform	abion List					Module	Informat		(Main Base)	1001204		Patrona.		2002.0959527.9	
Base Mor	dule Base Model Name	Power Supply	Base Type	Slots	Installed Modules	Status	Base- Slot	Series	Model Name	Point	Para Type	Point		Network No. Station No.	Master PLC
			0	0	4	1.1.1		1000	Power		Power				
	Main Base	Exist													
	Extension Base1	Exist					CPU	Q	QOSLIDHICPU		CPU			3 • 3	•
	Extension Base1 Extension Base2	Dat					0-0		Q05UDHCPU Empty	•	CPU Empty	OPoint			
	Extension Base1 Extension Base2 Extension Base3	Exist					0-0 0-1	Q.	Q06UDHCPU Empty QX42	64Point	CPU Empty Input	OPoint 64Point		•	1
	Extension Base2 Extension Base2 Extension Base3 Extension Base4	Exist					0-0 0-1 0-2	• • • • • • • • • • • • • • • • • • •	QOSLIDHICPU Empty QX42 QY42P	64Point 64Point	CPU Empty Input Output	OPoint 64Point 64Point	0000	•	:
	Extension Base1 Extension Base2 Extension Base3 Extension Base4 Extension Base5	Exist					0-0 0-1 0-2 0-3		QOSLIDHICPU Empty QX42 QY42P Q44AD	64Point 64Point 16Point	CPU Empty Input Output Intelli	OPoint 64Point 64Point 16Point	0000 0040 0000	•	•••••
	Extension Base1 Extension Base2 Extension Base3 Extension Base4 Extension Base5 Extension Base6	Dat	•				0-0 0-1 0-2 0-3 0-4		QOSLIDHCPU Empty QX42 QY429 QS4AD QS4DAN	64Point 64Point 16Point 16Point	CPU Empty Input Output Intell. Intell.	OPoint 64Point 64Point 16Point 16Point	• 0000 0040 0000 0090	•	••••
	Extension Base1 Extension Base2 Extension Base3 Extension Base4 Extension Base5 Extension Base6 Extension Base7	Dag					0-0 0-1 0-2 0-3 0-4 0-5	. 0000	QOSLEHCPU Empty QX42 QX42 Q44AD Q44AD Q42DAN Empty	64Point 64Point 16Point 16Point	CPU Empty Input Output Intell. Intell. Empty	OPoint 64Point 64Point 16Point 16Point 0Point	0000 0040 0090	•	
eral	Extension Base1 Extension Base2 Extension Base3 Extension Base4 Extension Base5 Extension Base6	Dag	4Modu				0-0 0-1 0-2 0-3 0-4 0-5 0-6	. 0000	QOSLOHICPU Empty QX42 QX42 Q44AD Q62DAN Empty Empty	64Point 64Point 16Point 16Point	CPU Empty Input Output Intell. Empty Empty	OPoint 64Point 64Point 16Point 16Point 0Point 0Point	0000 0040 0090 0090	* • • •	
egend	Extension Base1 Extension Base2 Extension Base3 Extension Base4 Extension Base5 Extension Base5 Extension Base7 10 me						0-0 0-1 0-2 0-3 0-4 0-5	. 0000	QOSLEHCPU Empty QX42 QX42 Q44AD Q44AD Q42DAN Empty	64Point 64Point 16Point 16Point	CPU Empty Input Output Intell. Intell. Empty	OPoint 64Point 64Point 16Point 16Point 0Point	0000 0040 0090 0090	•	
egend Minor	Extension Base1 Extension Base2 Extension Base3 Extension Base5 Extension Base5 Extension Base7 Interes	-	Modera	te Error	ornet.		0-0 0-1 0-2 0-3 0-4 0-5 0-6	. 0000	QOSLOHICPU Empty QX42 QX42 Q44AD Q62DAN Empty Empty	64Point 64Point 16Point 16Point	CPU Empty Input Output Intell. Empty Empty	OPoint 64Point 64Point 16Point 16Point 0Point 0Point	0000 0040 0090 0090	* • • •	• • • • •

Click the Detailed Information button.

NOT HE OF	Status		Co	nnection	Channel	lList										
k	H.	Monitoring	E.	Serial Po	t PLCN	fodule Connect	ion(USB)							System	Image	
lain Bi									O\$	veration to Selected M	odule					
Me	in Base									Main Base						
I/O A	dr.	0000 0040 0	000 0090		-					Slot 3						
				1	1 1					064AD						
										Noven						
	1			- 10-	a. a				1 2	2007 I. I.	2.24	1.1		12		- 1
	-		唐 唐	2					De	notemprotection	H/W Informa	/son	Bagnostics			1674
							0.2020	1		15						
ase Ir	formatio	n List			_		Module		tion List	(Main Base)						
		Base Model Name	Power	Base	Slots	Installed Modules	Status	Base- Slot	Series	Model Name	Point		moter		Network No.	Master
Base	Module	base Picces name	Supply	Type		modules	10000000	SIOC				Type	Point	Address	Station No.	PLC
lase		Main Base	Supply Exist	Q	8	4		Slot		Power		Power	Point	Address	Station No.	PLC
lace	57724111				8	1 Modules		cru	q	Power Q06UDHCPU	*		:	•		
lace		Main Base Extension Base1 Extension Base2			8	4		CPU 0-0		QOSUDHCPU Empty	:	Power CPU Empty	OPoint			100
Lace		Main Base Extension Base1 Extension Base2 Extension Base3			8	4		CPU 0-0 0-1	Q .	QOSUDHCPU Empty QU42	•	Power CPU Empty	OPoint 64Point		•	•
lace		Main Base Extension Base1 Extension Base2 Extension Base3 Extension Base4			8	4		CPU 0-0 0-1 0-2	Q . Q Q	Q05UDHCPU Empty QX42 QY42P	64Point 64Point	Power CPU Empty Input Output	OPoint 64Point 64Point	0000	•	
lace		Main Base Extension Base1 Extension Base2 Extension Base3 Extension Base4 Extension Base5			8	1		CPU 0-0 0-1 0-2 0-3	9.000	QOBUDHCPU Empty QVH2 QVH2P Q64AD	64Point 64Point 16Point	Power CPU Empty Input Output Intell.	OPoint 64Point 64Point 16Point	- - - - - - - - - - - - - - - - - - -	•	
Bace		Main Base Extension Base1 Extension Base2 Extension Base3 Extension Base5 Extension Base5 Extension Base6			8	Modules		CPU 0-0 0-1 0-2 0-3 0-4	Q . Q Q	QOSLIDHCPU Empty QVH2 QVH2P Q64AD Q62DAN	64Point 64Point	Power CPU Empty Input Output Intell.	OPoint 64Point 64Point 16Point 16Point	0000 0040 0080 0090		•
		Main Base Extension Base1 Extension Base2 Extension Base3 Extension Base4 Extension Base5		Q	8	Modules		CPU 0-0 0-1 0-2 0-3 0-4 0-5	9.000	QOSLIDHCPU Empty QH42 QH42P Q64AD Q62DAN Empty	64Point 64Point 16Point	Power CPU Empty Input Output Intell. Empty	CPoint 64Point 64Point 16Point 16Point 0Point	0000 0040 0080 0090	•	
lace		Main Base Extension Base1 Extension Base2 Extension Base3 Extension Base5 Extension Base5 Extension Base6			8	4		CPU 0-0 0-1 0-2 0-3 0-4 0-5 0-6	9 9999	QOSUDHCPU Empty QH2 QH2 QH42P Q64AD Q62DAN Empty Empty	64Point 64Point 16Point 16Point	Power CPU Empty Input Output Intell. Empty Empty	CPoint 64Point 64Point 16Point 16Point 0Point 0Point	0000 0040 0080 0090	•	•
teral		Main Base Extension Base1 Extension Base2 Extension Base4 Extension Base4 Extension Base5 Extension Base5 Extension Base5		Q	8	1		CPU 0-0 0-1 0-2 0-3 0-4 0-5	9.0000	QOSLIDHCPU Empty QH42 QH42P Q64AD Q62DAN Empty	64Point 64Point 16Point 16Point	Power CPU Empty Input Output Intell. Empty	CPoint 64Point 64Point 16Point 16Point 0Point	0000 0040 0080 0090	•	•
ngoria		Man Base Extension Base1 Extension Base2 Extension Base4 Extension Base5 Extension Base5 Extension Base5 Extension Base5 Extension Base7	Exist	Q.		•		CPU 0-0 0-1 0-2 0-3 0-4 0-5 0-6	9 9999	QOSUDHCPU Empty QH2 QH2 QH42P Q64AD Q62DAN Empty Empty	64Point 64Point 16Point 16Point	Power CPU Empty Input Output Intell. Empty Empty	CPoint 64Point 64Point 16Point 16Point 0Point 0Point	0000 0040 0080 0090	•	•
	inor	Main Base Extension Base1 Extension Base2 Extension Base4 Extension Base4 Extension Base5 Extension Base5 Extension Base5	Exist	Q Enerst	ite Error			CPU 0-0 0-1 0-2 0-3 0-4 0-5 0-6	9 9999	QOSUDHCPU Empty QH2 QH2 QH42P Q64AD Q62DAN Empty Empty	64Point 64Point 16Point 16Point	Power CPU Empty Input Output Intell. Empty Empty	CPoint 64Point 64Point 16Point 16Point 0Point 0Point	0000 0040 0080 0090	•	•

The following dialog box is displayed.

Module's Detailed Information		
Monitor Status Monitoring	Module Model Name I/O Address Mount Position Product Information Production Number	064AD 0000 0000 0000 Main Base 3 Slot 061200000000-C
	Module Information Module Access Status of External Power Supply Fuse Biown Status Status of I/O Address Verify I/O Clear / Hold Setting Noise Filter Setting	Possible
Error Information	L - Error and Solution	F=
No Error Gear Error No. Error	History r Code	
Cisplay Format	Solution:	
The error history is sequentially disp an old error. The latest error is displic the bottom line.	layed from ayed at	Close

Module Model Name Q64/ Display Format (• HEX	AD C <u>D</u> EC	Product Dé Information Dé	51220000000000-<	1					
	O DEC								
		H/W SW Information							
	H/W SV	H/W SW Information							
Item Va	lue I	tem Value	Item 1 2 3 4 5 5	Value: 0000 0000 0000 0000 0000					
	Item V	Item Value	Item Value Item Value	1 2 3 4					

The status of the module LEDs and switch setting can be displayed by clicking the H/W Information button.

(4) Operation practice

<--> PLC Module

Format PLC Memory

Connection Channel List

Connection Interface USB

Target Memory Program Memory

Create a user setting system area
 High speed monitor area from other station

Format Type

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

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

-

0

K Steps (0--15K Steps)

Execute

Close

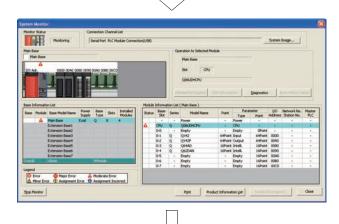
This section explains how to check the cause of the error which is caused intentionally here with the PLC diagnostics function from the system monitor.

1) Stop the CPU.

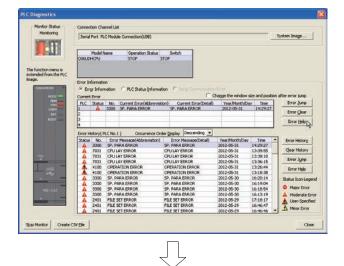
X

- Click [Online] → [PLC Memory Operation] → [Format PLC Memory] to format the program memory. (Refer to section 5.3.1 (5))
- 3) Reset the CPU.

Error occurs.



- Click [Diagnostics] → [System Monitor] and check the module where the error occurs in the System Monitor dialog box.
- 5) After checking the error in the CPU, select "Q06UDH" and click the Diagnostics button.



6) After checking "3300 SP.PARA ERROR" in "Current Error", click the Error Help button.

(To the next page)

	(From the previo	Jus page)
or Help		
Ton		
3300	SP. PARA ERROR SP. PA	RAERROR
Explanati	on	
The start	1/O number in the intelligent function modu actual 1/O number.	le parameter set on peripherals differs
Troublesł	notina	
	e parameter setting.	

 The detail of the error code is displayed. Check the cause and the solution.

8.1.3 System protection function

The QCPU has protection functions (system protection) to prevent programs from being modified by a user other than the designer.

Protection target	File to be protected	Description	Method	Remark
In units of memory cards	All files	Prohibits writing to a memory card.	Turn on the write protect switch of a memory card.	-
In units of files	Program Device comment Device initial value	Changes the attribute for each file to either of the following. Read/write prohibition Write prohibition	Change the attribute for each file in the password registration.	-

Control instruction, read/write, and writing in the table above indicates the following contents.

Item	Description
Control instruction	Operation instruction for the CPU by the remote operation (such as remote RUN and remote STOP)
Read/write	Operations of reading or writing programs
Write	Operations regarding writing programs

8.1.4 Password registration function

A password prohibits reading and writing (overwriting) data such as programs and comments in the QCPU module using a peripheral device.

The password can be registered in units of files of the specified memory (program memory, standard ROM, memory card (RAM), and memory card (ROM)). The following two operations can be prohibited.

- Reading and writing a file
- Writing a file (Reading is available.)

When a password is registered to a file, the file cannot be operated from the peripheral device unless the password is canceled.

To cancel the password, click [Online] \rightarrow [Password/Keyword] \rightarrow [Disable]. Or cancel the password when executing the limited operation.

(1) Registering a password

Click [Online] \rightarrow [Password/Keyword] \rightarrow [New] in GX Works2.

For the details of the operation method of each function, refer to the GX Works2 Operating Manual.

	10		
Target <u>M</u> emor	Program Memor	ry/Device Memory	
Data Type	Data Name	Registration	Registration Conditions
Program	CONTROL	🐖 Register	Write Protection
Program	INITIAL		
Program	PR1		N
Program	PR2		6
Program	PR3		
Program	SUB		

Each item is explained as follows.

- (a) Target Memory
 Set a memory of a file with which a password is registered.
- (b) Data Type Displays the data types of the files with which passwords can be registered.
- (c) Data Name Displays the data name of the files with which passwords can be registered.
- (d) Registration
 Displays the password registration status of the files written in the CPU.
 "Image: "Image: Image: Im
- (e) Registration Conditions Set conditions of the files with which passwords are registered.
 - Read/Write Protection Reading or writing is unavailable unless the password is canceled.
 - Write Protection Writing (overwriting) is unavailable unless the password is canceled. Reading is available.

POINT

- (1) Passwords can be registered only for programs, device comments, and device initial values.
- (2) The passwords registered in the CPU cannot be read by the peripheral device.
- (3) To cancel the password registration, click [Online] \rightarrow [Password/Keyword] \rightarrow [Delete].

8.2 Other Functions

Function list

The following table lists other functions.

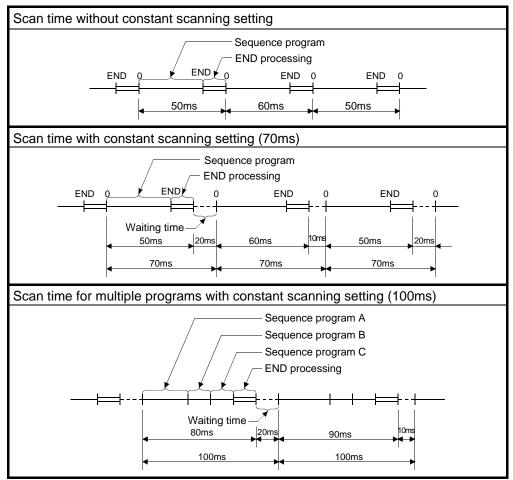
	Item	Description
Co	nstant scan function	Executes a program in a set time interval regardless of its scan time.
Lat	tch function	Holds the device data even at power-off or reset.
	ttery life-prolonging	Extends the life of a battery by holding only clock data.
wh	tput status setting function en the status changed m STOP to RUN	Sets the output (Y) status (outputting the same status prior to STOP or clearing the status) when the operating status of the CPU module is switched from STOP to RUN.
Clo	ock function	Executes the internal clock of the CPU.
Re	mote operation function	Operates the QCPU externally.
	Remote RUN/STOP	Runs or stops the CPU.
	Remote STEP-RUN	Executes the CPU in step execution.
	Remote PAUSE	Stops the CPU temporarily.
	Remote RESET	Resets the CPU.
	Remote latch clear	Clears the latch data in the CPU.
	Relationship between remote operation and CPU	The relationship between the RUN/STOP switch setting for the CPU and the remote operation is explained.
Se	rvice processing setting	Specifies the service processing count or time to be executed in the END processing.
inte	ritch setting for the elligent function module oporting QCPU	Configures settings for intelligent function modules. (Refer to manuals of intelligent function modules for setting details.) \rightarrow Refer to section 9.2.3.
Re	sponse time change for	Changes the response time of the input module supporting
	e input module supporting CPU	QCPU to 1ms, 5ms, 10ms, 20ms, or 70ms. (Default: 10ms) \rightarrow Refer to section 10.7

For the details on the functions which require the operation of a peripheral device, refer to the GX Works2 Operating Manual.

8.2.1 Constant scan function

(1) Constant scan

The scan time of the QCPU varies since the processing time differs depending on the execution status of instructions used in the sequence program. The constant scan function repeatedly executes sequence programs keeping their scan time constant.



Constant scanning operation

For details, refer to the QnUCPU User's Manual (Function Explanation, Program Fundamentals).

POINT

Set a constant scanning time in the "PLC RAS" tab of the PLC parameter in the project data list in GX Works2. Set the constant scanning time within the following range. (WDT setting time) > (Constant scanning setting time) > (CPU maximum scan time)

8.2.2 Latch function

The latch function holds data in each device of the CPU module when:

- the CPU module is powered off and then on,
- the CPU module is reset, or
- power failure occurs exceeding the allowable momentary power failure time.

Data in each device of the CPU module is cleared and back to its default (bit device: off, word device: 0) without the latch function.

Program operation is the same, regardless of the latch status.

(1) Application

This function can be used to hold the data, such as the number of manufactured products, the number of fault products, and address, and to continue the control even when a power failure exceeding the allowable momentary power failure time occurs during the sequential control.

- (2) Devices that can be latched
 - (a) The following devices can be latched.
 - Latch relay
 - Link relay
 - Annunciator
 - Edge relay
 - Timer
 - Retentive timer
 - Counter
 - Data register
 - Link register

POINT

When the battery life-prolonging function is set, the latch relay cannot be latched.

Battery life-prolonging function

This function extends the battery life installed in the CPU module by setting the data to be held to only the clock data.

All data other than the clock data are initialized when the CPU module is powered off or is reset.

For details of the battery life-prolonging function, refer to the QnUCPU User's Manual (Function Explanation, Program Fundamentals).

(b) Set the latch range in the "PLC RAS" tab of the PLC parameter in the project data list in GX Works2.

The latch range setting has the range where the latch clear key becomes valid and the range where the key becomes invalid.

For the latch range of each device, refer to the QnUCPU User's Manual (Function Explanation, Program Fundamentals).

8.2.3 Remote operation function

The remote operation can change the operating status of the CPU module externally (with GX Works2, external devices with the MC protocol, link dedicated instructions of CC-Link IE controller network modules or MELSECNET/H module, or remote contacts).

Remote operation	RUN	STOP	PAUSE	RESET	Latch clear
RUN	RUN	STOP	PAUSE	Operation disabled	Operation disabled
STOP	STOP	STOP	STOP	RESET	Latch clear

The remote operations as follows can be executed with the QCPU.

(1) Remote RUN/STOP

Set the RUN/STOP switch to RUN when executing the remote RUN/STOP. The remote RUN/STOP has two methods for the execution.

(a) Operation from GX Works2 Use the remote RUN/STOP command from GX Works2. Execute the remote STOP, then execute the remote RUN.

Water 10 Improvement (PSCH1 'PR1'' Improvement (PSCH1 'PR1'' (PSCH1 'PR1'') Improvement (PSCH1 'PR1'') (PSCH1 'PR1'')	ther tile ary						
Statute Image: Statute		16	-11		PSON	1962*	1
Name Name <th< td=""><td></td><td>10</td><td></td><td></td><td></td><td>1992</td><td>1</td></th<>		10				1992	1
Name Nam Name Name	Terris Menery		Het	1			
Structure 1000 Structure	NUA 81 82 83	4	BCRidde Darge Set gols	*	{PORF	TRI	1
Company Company <t< td=""><td>G Bibalbena Caneer. Propert lating RDU</td><td></td><td>FLC use Sets Expert to RCM (send</td><td>•</td><td></td><td></td><td></td></t<>	G Bibalbena Caneer. Propert lating RDU		FLC use Sets Expert to RCM (send	•			
Arrighter A.R. Contractor	di facesta	: e	Passard(proof RCNews; Question		[PSCAN	1981	1
	D D B B B W W.		generalite			<u>aan</u>	

Ĺ

USB

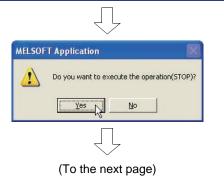
Target PLC

1) Click [Online] \rightarrow [Remote Operation].

ote Operation Connection Channel List Connection Interface <--> PLC Module Station No. Host PLC Type Q06UDH Q06UDHCPU Specify Execution Target Currently Specified Station 💌 R Operation C RUN • STOP C PAUSE Latch Clear RESET Remove Memory $\underline{\subset}$ ard Execute

Close

- 2) The dialog box is displayed. Select "STOP" in "Operation".
- 3) Click the Execute button.



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

(From the previous page)
MELSOFT Application
The operation(STOP) has been completed.
СК
Lemote Operation
Connection Channel Ust Cennection Noterface 08
COEUDHICPU Statum No. Host FLC Type (200404
Currently Specified Station
USER
BOOT C STOP
C BADE C Latch Clear C RESET
C Renove Menory Card
PULL Cente Menory NAC Central T Siggs Flow Execute
Had Door
\square
MELSOFT Application
Do you want to execute the operation(RUN)?
MELSOFT Application
The operation(RUN) has been completed.
COK D
Remote Operation
Connection Oternel List Connection Niterfece J38 <-> (R.C. Module
Target RLC Technol. No.
COEUDHCPU
ERR.
BAT. Common BOOT Carge
/ Puse ∩ unon Onen / Pesse:
C Remove Memory Card
PULL Device Nervoy Net Cleared Signal Play
Hold V

•

- 5) When the remote STOP is completed, the dialog box is displayed. Click the OK button.
- 6) Select "RUN" in "Operation".
- 7) Click the Execute button.

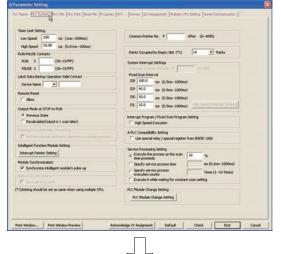
- 8) The confirmation dialog box is displayed. Click the Yes button.
- 9) When the remote RUN is completed, the dialog box is displayed. Click the OK button. Confirm that the CPU is running with the demonstration machine.
- 10) Click the Close button to close the dialog box.

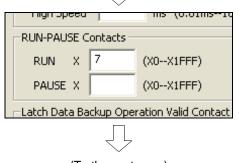
Close 2

(b) Method with the remote RUN contact

Set the remote RUN contact (X) with a parameter. Turning on the contact turns the operation status of the CPU to STOP (turning off runs the CPU).

Ψ× Navigation Project 📫 🗈 🗞 🗞 🖻 👫 Parameter Intelligen Sunction Module + 🚯 Global Device Comment Program Setting 🖻 🛅 Program MAIN Navigation **Ψ**× Project 📑 🗈 🔁 🖻 👘 🛃 Parameter PLC Par 🚯 Network Paranter 🙀 Remote Password Intelligent Function Module Global Device Comment Program Setting POU Program Ė MAIN





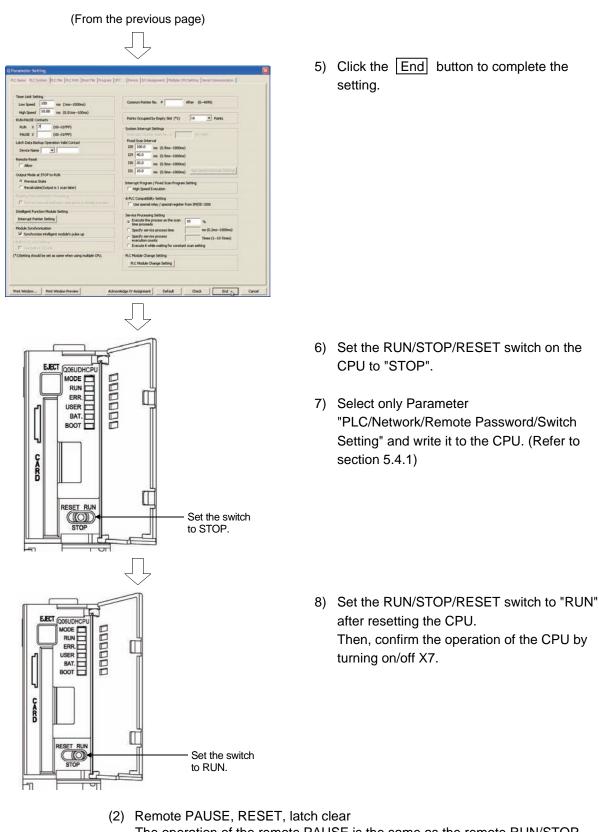
(To the next page)

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

2) Double-click "PLC Parameter".

 The Q Parameter Setting dialog box is displayed. Click the "PLC System" tab.

 To set the RUN contact to X7, enter "7" in the RUN column of RUN-PAUSE Contacts.



- Remote PAUSE, RESET, latch clear
 The operation of the remote PAUSE is the same as the remote RUN/STOP described before.
 For the remote RESET and remote latch clear, the CPU module is needed to be set in the STOP status by the remote STOP operation.
 - * Before the remote RESET is executed, the remote RESET is needed to be allowed in the PLC parameter setting.

8.2.4 Service processing setting

This function can set the number of times and time of the service processing executed in the END processing by parameters.

This function also improves the response of communication with a peripheral device and restrains the increase of the scan time due to the service processing.

This enables the configuration of optimal service processing environment for the system.

```
POINT
```

The service processing indicates the communication service processing with the peripheral device (such as GX Works2) and intelligent function module. However, the link refreshing processing, such as with the CC-Link IE controller network module, MELSECNET/H module, and CC-Link system master local module, are not included.

Using the COM instruction enables the same service processing as the END processing even during the program execution.

Therefore, the high-speed service processing response is available even when the scan time is long.

(1) Parameter setting

Set the parameters in the PLC system setting of the PLC parameter.

mer Limit Setting Law Speed 100 ms (1ms-1000ms) High Speed 10.00 ms (0.01ms-100ms)	Common Pointer No. P Alter (0-4095)
UN-PAUSE Contacts	Points Occupied by Empty Slot (*1) 16 Points
RUN X (X0-XIFFF) PAUSE X (X0-XIFFF)	System Interrupt Settings Interrupt Countre Start No. C (0-768)
Latch Data Backup Operation Valid Contact Device Name	Fixed Scan Interval 123 100.0 ms (0.5ms-1000ms)
Remote Reset	129 40.0 ms (0.5ms-1000ms) 130 20.0 ms (0.5ms-1000ms) 131 10.0 ms (0.5ms-1000ms)
Output: Mode at STOP to RUN Previous State Recalculate(Output is 1 scan later)	T31 10.0 ms (0.5ms-1000ms) High Speed Internal Setting Internut Program / Fixed Scan Program Setting Internut Speed Execution
Cloating Forst Anometic Processing Perform Internal antimetic operations in double precision	A FLC Compatibility Setting Use special relay / special register from SM(SD 1000
Intelligent Function Module Setting	
Interrupt Pointer Setting	Service Processing Setting
Module Synchronization Synchronize intelligent module's pulse up	Specify service process time Specify service process Timer (1, 10 Timer)
Buit-in CC-Link Setting T Use built-in CC-Link	execution counts imes (1-10 mines) Execute it while waiting for constant scan setting
*1)Setting should be set as same when using multiple CPU,	PLC Module Change Setting PLC Module Change Setting

To execute the service processing, select any of the parameter items in the following table.

The setting value of deselected parameter cannot be entered. (Default: Execute the process as the scan time proceeds = 10%)

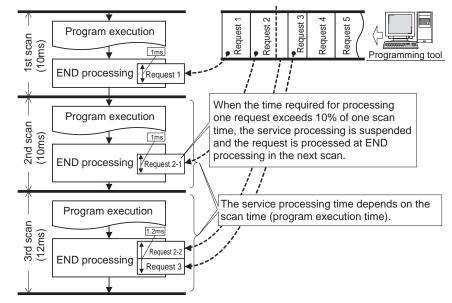
Item	Description	Setting range	Remark
Execute the process as the scan time proceeds	Set the percentage of service processing for one scan.	• Range: 1 to 99% • Unit: 1%	Default when selected = 10%
Specify service process time	Set the time of service processing for one scan.	• Range: 0.2 to 1000ms • Unit: 0.1ms	Default when selected = 0.2ms
Specify service process execution counts	Set the number of times for service processing for one scan.	 Range: 1 to 10 times Unit: 1 time	Default when selected = 1 time
Execute it while waiting for constant scan setting	Set whether to execute service processing during a waiting time for constant scanning setting.	-	Even when the waiting time is 0.2ms or less, the service processing time (0.2ms) is added to the scan time at the service processing execution.

(2) Operations for service processing setting

Operations for each service processing setting is described below. The operation of "Execute the process as the scan time proceeds" is described here.

For other operations, refer to the QnUCPU User's Manual (Function Explanation, Program Fundamentals).

(a) Operation when 10% is set



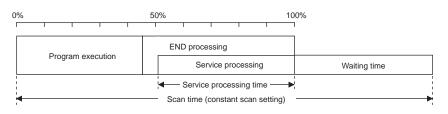
POINT

When no request data for service processing exists, the END processing becomes faster for the request processing time. (The CPU module does not wait for requests.)

(b) Operation for constant scanning setting

The percentage calculation is executed for the time in which the waiting time for the constant scanning is subtracted from the scan time, not executed for the scan time.

Ex. Operation of when "percentage of service processing = 50%"



(3) Precautions

The following explains precautions for the service processing setting.

- (a) For the following functions, scan time exceeds the specified time during service processing even if the service processing time specification is set.
 - Online program change
 - Change T/C setting
 - Local device monitor
 - Program memory backup
 - Writing/reading data to/from a file register (The scan time is increased when the write or read size is large.)
 - Writing/reading data to/from the buffer memory of the intelligent function module (The scan time is increased when the write or read size is large.)
 - Access to a network module
 - Diagnostic functions (CC IE Control diagnostics, MELSECNET diagnostics, Ethernet diagnostics, CC-Link/ CC-Link/LT diagnostics)
 - 2) Monitor function (Module access device, Link direct device)
- (b) Note that the scan time may be increased much longer if the CPU module receives multiple requests simultaneously while the many service processing count specifications are set.
- (c) When the service processing time is set much shorter than the scan time, the response performance of the service processing decreases significantly.

Set the service processing time considering the time-out time of the peripheral device.

(d) An error of -20µs to +30µs occurs between the actual processing time and the set service processing time.

8.3 Comments Storage Function

The QCPU can store various types of comments. This improves the operability of the CPU and makes the program easier to be read for users other than the program designer.

The following table lists each type of the comment which can be stored to the QCPU.

Item	Function
PLC name	Give a name for the CPU to be used.
Title	Give a title for each project.
Data name	Change the data name of the opened project.
Property	Give a title or comment for each data.
Device comment	Gives a comment or label to the device to be used in a program.
Statement/note	Give a statement/note for each step number, P or I pointer.
Device initial value	Cives a comment for the device initial value file
comment	Gives a comment for the device initial value file.

For the details on the setting method of each function, refer to the GX Works2 Operating Manual.

(1) PLC name

Giving a comment to a CPU enables easy confirmation of the target CPU when GX Works2 accesses the QCPU.

Set a label and comment as a PLC name. Set a PLC name in the "PLC Name" tab of the PLC parameter in the project data list.

Q Paramete	r Setting	×
PLC Name	PLC System PLC File PLC RAS Boot File Program SFC Device 1/O Assignment Multiple CPU Setting Serial Communication	
Label	System	
Comment	Convey system	

The following table lists the setting contents.

Item	Description	Setting range	Default
Label	Set a label for the CPU.	Up to 10 characters	Blank
Comment	Set a comment for the CPU.	Up to 64 characters	DIALIK

(2) Title (project index)

Give a title to a project to identify its contents.

Set a title when saving a project with a different name, and the title is stored in the created project.

Save Location:	
C:(SCHOOL)	Browse
Workspace/Project List:	
Workspace	
SCHOOL	
Workspace Name:	50400.
-	SOHOOL Acceled
Broject Name:	
Broject Name: Ijde:	Applied
Workspace Name: Broject Name: IBle: IPlaude revisions	Applied
Broject Name: Ijde:	Applied Control program
Project Name: [Itle: IV Include revisions	Applied

POINT

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.

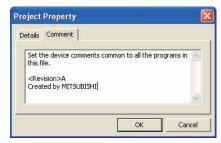
(3) Data name

On the opened project window, select the data name to change. Right-click and click [Rename] to change the data name.

(4) Property

The properties for a folder, parameter, and program are displayed. Also, giving titles and comments for each data is available.

etails Comm	ent	
Data Name	Applied 2	
Title	Common comments to all programs	
Last Change	5/31/2012 4:38:33 PM	



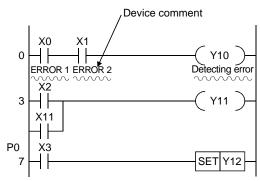
[How to display screen] [Project] \rightarrow [Object] \rightarrow [Property]

[Screen items]

Item Description		
Data Name	Displays the data name.	
Title	Set a title (index) for the data.	
(Up to 128 characters are applicable for a project, and 32 for other data		
Last Change Displays the date when the data was updated.		
Comment Set a comment for the data. (Up to 5120 characters are applicable.)		

(5) Device comment

A device can be displayed with comments, to make the program easier to be read.



(a) Set the device comments on the ladder creation screen.
 Click and double-click at the position where to set the device comments.

Device/Label	Device/Label Comment	Preview	ОК
X0	ERROR1	ERROR1	
			Cancel

- The devices for which a comment can be given are shown below. Device name: X, Y, M, L, F, SM, B, SB, V, T (current value), C (current value), ST (current value), D, SD, W, SW, R, ZR, P, I, U□\G□, J□\X, J□\Y, J□\B, J□\SB, J□\W, J□\SW, BL□\S, BL□\TR (when the comments for P and I are used
 - BLD\S, BLD\TR (when the comments for P and I are used as pointers for a subroutine program and interrupt program, the comments are not displayed. To display the comment, use a pointer statement. (Refer to (6))

(b) When device comment files are written in the CPU, select a valid device comment file with the parameter to use the comment with the application instruction (COMRD).

Set the setting in the "PLC File" tab of the PLC parameter in the project data list.

ile Register	Device Initial Value
Not Used	Not Used
C Use the same file name as the program	C Use the same file name as the program
Corresponding Memory	Corresponding Memory
C Use the following file	C Use the following file
Corresponding Memory	Corresponding Memory
File Name	File Name
Capacity K Points	
(1K4086K Points)	File for Local Device
🔲 Transfer to Standard ROM at Latch data backup o	
Following settings are available in device setting	C Use the following file
when select "Use the following file" and specify capacit -Change of latch(2) of file register.	Corresponding Memory
-Assignment to expanded data register/expanded link register of part of file register area.	File Name
Comment File Used in a Command	File used for SP.DEVST/S.DEVLD Instruction
Not Used	Not Used
C Use the same file name as the program	C Use the following file
Corresponding Memory	Corresponding Memory
C Use the following file	File Name
Corresponding Memory	Capacity K Points
File Name	(1K512K Points)

The following lists the setting contents.

1) Not Used

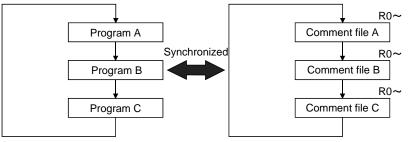
No file register is set by the parameter.

Set this when using a sequence program to specify the comment file to use.

Use the QCDSET instruction to specify the comment file to use. (For details of the QCDSET instruction, refer to the MELSEC-Q/L Programming Manual Common Instruction.)

2) Use the same file name as the program Specify the drive of the memory card.

Every time an execution program is changed, the valid comment file is also changed.

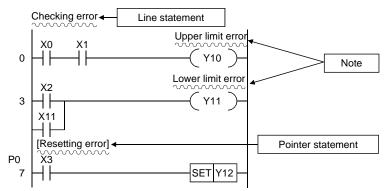


3) Use the following file

Only the comment file specified with the drive and file name is valid.

(6) Statement/note

The statement/note is given for each program step, P, or I pointer to make the program easier to be read.



(a) Set the statement/note on the ladder creation screen. Click 2/2 and double-click at the position where to set the statement/note.

inter Line S	tatement	
• In <u>P</u> LC	Initial setting	ок
C In Periphe	ral Display in Navigation Window	Exit

- (b) Applications of each comment are as follows.
 - Line statement Describes the meaning and application of a ladder block organized by each function.
 - Pointer statement

Describes the meaning and application of each program corresponding to the pointer set to the start of the subroutine program and interrupt program.

Note

Describes the meaning and application of each ladder block.

(7) Device initial value comment

Give a comment for a device initial value file to identify its content. The device initial value comment is stored in the device initial value file. Configure the setting on the Device Initial Value screen in the project data list.

- 12	Points	Start	End	Comment	-
2			8		
3			2		
ŧ					
5					
5					
3			· ·		
3			2		
0			9		
1			0		
2			Q		
3			0		
4			S		
5			S		
6	2		2		-
ı mu ettir		evice Memory D	mory Diversion	vill be write to PLC. hen the device initial range set	ting is chang
C F	Points/Start			Device Memory Div	ersion

8.4 Appropriate Assignment of Device Points

In the ACPU, the device points are fixed.

In the QCPU, the device points to be used can be assigned according to a system appropriately. The following table lists the description.

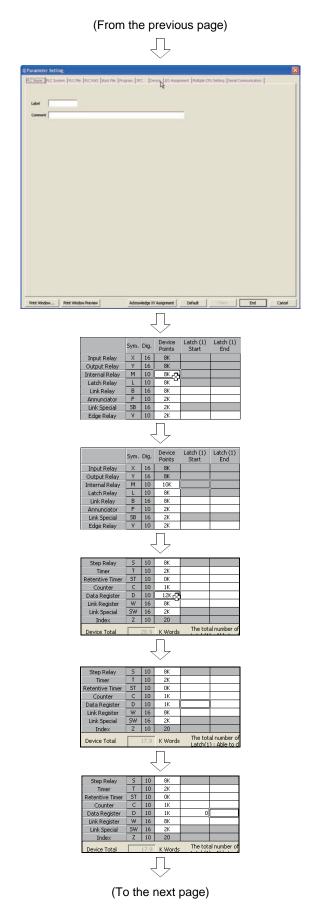
Item	Description	Setting range	Default value
Number of device points	Set the number of each internal device points.	Up to 32K points for one device can be set within the range of 29K words in total (except for the devices X, Y, S, SB, and SW).	X : 8K points (Fixed) Y : 8K points (Fixed) M : 8K points L : 8K points B : 8K points F : 2K points SB : 2K points S : 8K points T : 2K points S : 8K points C : 1K points D : 12K points W : 8K points SW : 2K points
Latch range (1) (Latch clear operation enable range)	Set a latch range where the data can be cleared by the latch clear operation.	Only one range can be set for each device.	Blank
Latch range (2) (Latch clear operation disable range)	Set a latch range where the data cannot be cleared by the latch clear operation.	Only one range can be set for each device.	Blank

 The following explains how to change the internal relay M to 10K points and the data register D to 1K points (D0 to D500: latch clear operation enable range, D501 to D1023: latch clear operation disable range).



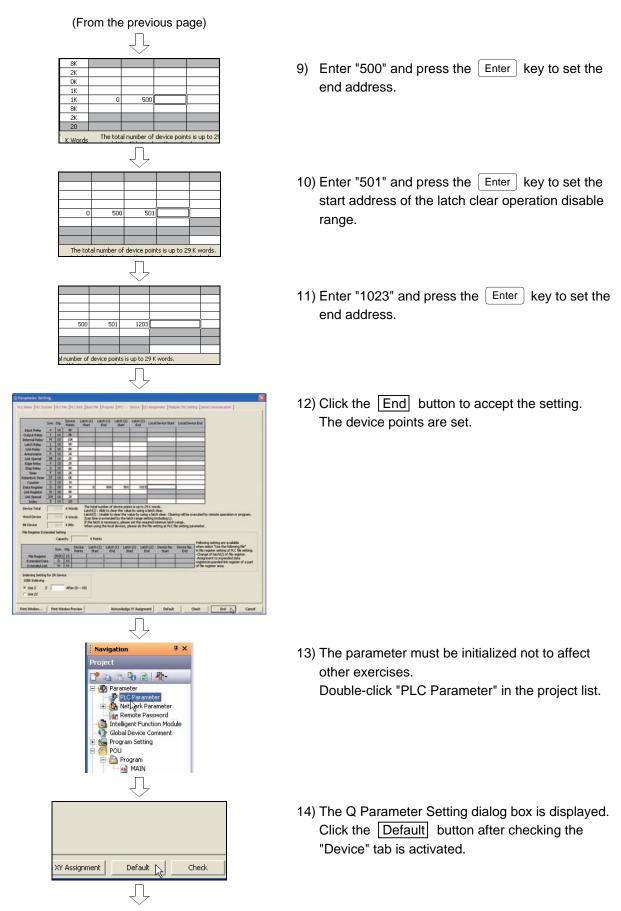
1) Double-click "Parameter" in the project list.

2) Double-click "PLC Parameter".



 The Q Parameter Setting dialog box is displayed. Click the "Device" tab.

- The screen is switched. Click the Internal relay (M) column of Device Points to move the cursor to the column.
- 5) Enter "10K" and press the Enter key.
- 6) Click the Data Register (D) column of Device Points to move the cursor to the column.
- 7) Enter "1K" and press the Enter key.
- 8) The cursor is moved to the Data Register column of Latch (1) Start. Enter "0" and press the Enter key.



(To the next page)

(From the previous page)	
MELSOFT Series GX Works2	15) Click the Yes button.
Setting default data. Is that OK?	
<u>Yes</u> <u>N</u> o	
0 Parameter Setting	
RCName [RCSystem [RCSNE] [RCNAS [Ibot File [Program]97C] Device [10 Assignment [Multiple CPU Setting [Senid Communication]	10) Click the End button.
and the Device Latch(1) Latch(1) Latch(2) Latch(2) studying that Long from the	The device setting is returned to the default.
Input Relay X 16 0K End Start End	The device county is retained to the deladar
Output Relay Y 16 BK	
Leth Palay L 10 0K	
Urk Special 38 16 2X Edge Reley V 10 2X	
Step Relay 5 10 M Teser 7 10 2X	
Retentive Timer SI 10 0X Counter C 10 1X Data Registry D 10 1X	
Link Register W 16 BK	
Index Z 10 20 The total number of device points is up to 29 K words.	
Word Device K Words K controls is dear the value by using a latch clear. Clearing will be executed by remote operation or program. K controls is controls in controls on program. K controls is controls in controls on program.	
BR Device K Bits K Bits When using the local devices, please do the file setting at PLC file setting parameter.	
Fie Register Extended Setting Capacity Capacity KPonts Following setting are evaluate	
Sym. Dg. Device Latch (1) Latch (2) Latch (2) Latch (2) Device No. Device No. When select "Use the following file" Points Start End Start End Start End Start End in file register setting of PLC file setting.	
Extended Data D 10	
Extended Link W 16 of 78 register area.	
3288 tridesing # Use 2 2 # Wher (0 - 18)	
(* Use Z I After (0 – 10) (* Use ZZ	
Priet Window Priet Window Preview Addrowledge XY Assignment Default Check End L Cancel	

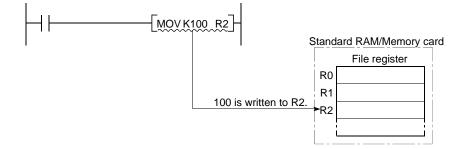
POINT

(Internal user) When the device assignment range is changed, the sequence program created with the parameter before the change cannot be used as it is. After the change, the sequence program and parameter must be written to the CPU again.

8.5 Using File Register

The file register is a register extended from the data register (D). Normally, the file register is used with the standard RAM or memory card.

The file register is stored in the standard RAM of the QCPU or a memory card installed on the QCPU in a file format.



Accesses available for the file register vary for each memory.

	Access method	Standard RAM	SRAM card	Flash card
Writing progr	am	0	0	×
Reading prog	gram	0	0	0
Writing device	e memory to programmable controller	0	0	×
Reading dev	ice memory from programmable controller	0	0	0
	Online test operation from GX Works2	0	0	×
	Writing data to programmable controller with GX Works2	0	0	×
Data modification	Writing data to programmable controller with GX Works2 (flash ROM)	×	×	0
	Batch-writing with a serial communication module	0	0	×
	Writing data with GOT1000 series	0	0	×
	Random write command from GOT1000 series	0	0	×

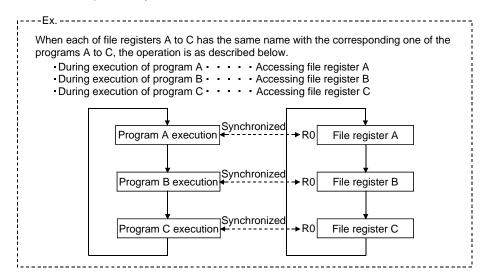
8.5.1 Preparation for using file register

(1) Selecting the file register

Set the file register to be used in a memory card in the sequence program in the "PLC File" tab of the PLC parameter in GX Works2.

- (a) When selecting "Not Used"
 Set this when using a sequence program to specify the file register to use.
 Use the QDRSET instruction to specify the comment file to use.
- (b) When selecting "Use the same file name as the program" Set this when using the file register with the same file name as the sequence program independently. When the program is switched, the name of the file register is automatically changed to the same name as the program.

This is useful when the file register is used for one program as a local device independently.



(c) When selecting "Use the following file"

Set this when one file register is to be shared by all programs. With "Corresponding Memory", "File Name", and "Capacity" set corresponding to the file register to be used, a file of the file register specified by the parameter is created when the QCPU starts to run.

In this exercise, set the file register in the standard RAM.

Personal Sector ELLER State Journe (report (re	Deex [10 support [Maps DV berry [Dearton-search]		1)	Click t
Beinger Parallel Paralle	Period * # Station * # Station * # Station * Generative Mary * Generative Mary * Generative Mary * Markon * Warkali * Generative Mary * Gen			Setting
Ret Vielen. Ret Vielen Reven Abrai		and		
File Register C Not Used C Use the same file name Corresponding Memory	as the program]	2)	Select set the Corres
 Use the following file Corresponding Memory File Name 	Standard RAM (Drive 3)	1		File Na Capac
Capacity	2 K Points (1K4086K Points) ROM at Latch data backup operatio	n.		
Following settings are avail when select "Use the follov -Change of latch(2) of file i	able in device setting ving file" and specify capacity, register, data register/expanded link			
	$\overline{\Box}$			
			3)	Click t
Check	End Cancel			

1) Click the "PLC File" tab on the Q Parameter Setting dialog box.

 2) Select "Use the following file" of "File Register" and set the following items.
 Corresponding Memory: "Standard RAM (Drive 3)" File Name: "FILEREG" (Example)
 Capacity: "2" (Example)

- 3) Click the End button.
- (2) Registration to the QCPU
 - (a) When "Not Used" or "Use the same file name as the program" is selected in the "PLC File" tab of the PLC parameter setting in GX Works2, the file register files are required to be registered to the QCPU. The registration is not required when "Use the following file" is selected.
 - (b) To register the file register files to the QCPU, set the file name and file register size in the "PLC File" tab of the PLC parameter in GX Works2 and write it to the QCPU.

Set the file register size is from ZR0 in units of 1K points (1024 points).

POINT

Precautions for when the file register is unregistered or exceeds the registered size

- (1) When the file of the file register is not registered Writing/reading data to/from the file register causes the "OPERATION ERROR" (error code: 4101).
- (2) Writing/reading data to/from the file register exceeding the registered size (points)

Writing/reading data causes the "OPERATION ERROR" (error code: 4101).

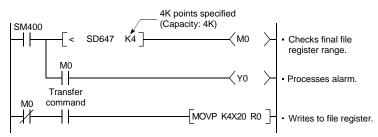
- (3) Register size check
 - (a) When writing/reading data to/from the file register, check the file register size so that data can be written or read within the size (points) set for the CPU module.
 - (b) The available file register size can be checked in the File register capacity area (SD647).

The file register size in units of 1K points is stored in SD647. (The points 1K or less are rounded off.)

- (c) File register size checking procedure
 - 1) Check the file register size to be used.
 - 2) Check that the total file register size set in SD647 on the sequence program is sufficient for the points to be used.

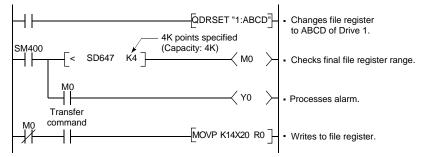
[Program example 1]

When checking the file register range used at the beginning of each program



[Program example 2]

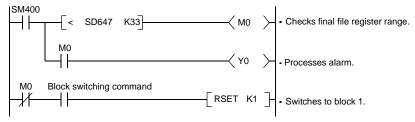
When checking the file register range used after execution of the QDRSET instruction



*: When a file of the file register is switched, the file register size of the currently selected file is stored in SD647.

[Program example 3]

When switching a block



*: Before switching the file register block with the RSET instruction, confirm that the block after the switching has the size of 1K points or more.

(File register size) > [32K points \times (Switching block No.) + 1K points]

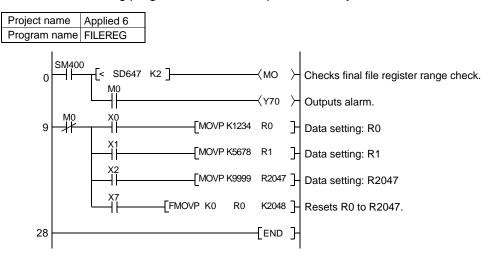
(4) Memory card format

In this exercise, store the file register in the standard RAM. Therefore, format the standard RAM as preparation.

Format PLC Memory	1) Click [Online] \rightarrow [PLC Memory Operation] \rightarrow
Connection Channel List	[Format PLC Memory] of GX Works2, and select
Connection Interface USB <> PLC Module	
Target PLC Network No. 0 Station No. Host PLC Type Q06UDH	"Standard RAM" for "Target Memory" and click the
Target Memory	Execute button.
Format Type	
O not create a user setting system area (the required system area only)	
C Greate a user setting system area	
High speed monitor area from other station 0 K Steps	
Online change area of multiple blocks	
Unime change area or mulique glocis	
Execute Close	
Ţ,	
MELSOFT Application	2) Click the Yes button.
Memory will be formatted.	
Do you want to continue?	
Format PLC Memory	3) Click the Close button.
Connection Channel List	blick the blobe button.
Connection Interface USB <> PLC Module	
Target PLC Network No. J 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)	
C Greate a user setting system area	
High speed monitor area from other station 0 ()-15K Steps	
Online change area of multiple blocks	
Execute Close	

8.5.2 Operation check

(1) Creating a program for operation checkCreate the following program to check the operation easily.



(2) Program setting

Configure the setting as shown below in the "Program" tab of the PLC parameter in the project data list.

	Program Name	Execute Type	Fixed Scan Interval	In Unit 🔺
1	MAIN	Scan 👻		•
2		-		-
3		-		•
4		-		•
5		-		-
6		-		-
7		-		-

(3) Writing data to the programmable controller Write the program (MAIN) and parameter to the CPU.

POINT

Select "Program Memory" for the target memory when writing the program and parameter to the programmable controller.

(4) Operation check

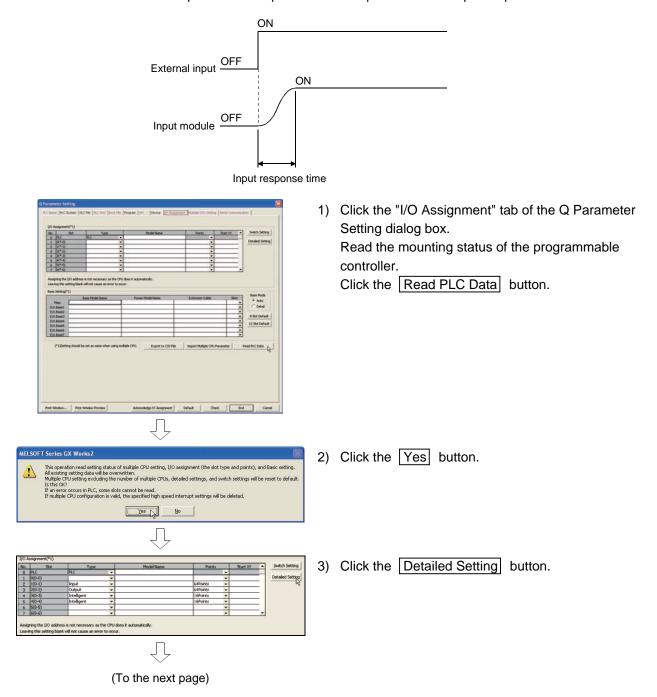
Run the CPU to check the operation on the ladder monitor.

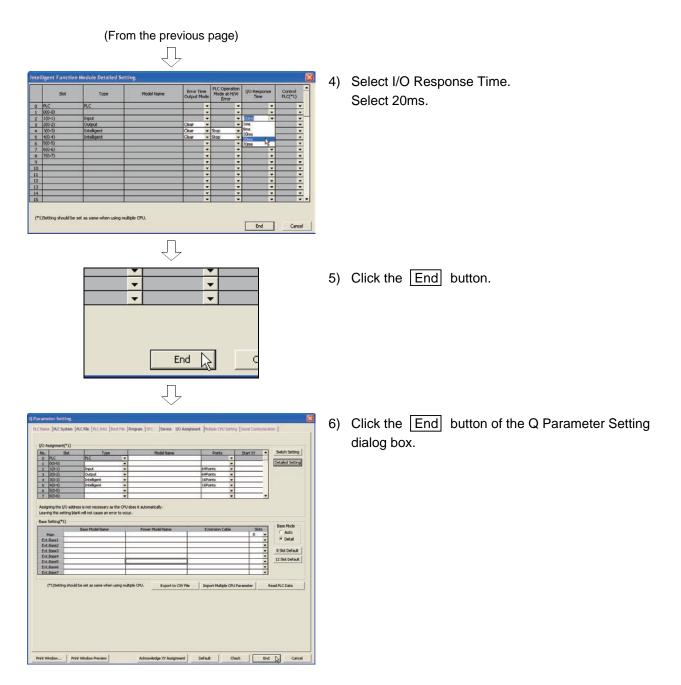
- \bullet The register R is set when X0, X1, and X2 are turned on.
- The register R is reset when X7 is turned on.

In addition, check the value set when X0, X1, and X2 are turned on is not cleared by the reset or latch clear of the CPU.

8.6 Input Response Speed Change

For the input, high-speed input, and I/O combined module, the input response speed can be set individually with the parameter setting of GX Works2. The input module imports external inputs in the set input response time.





Write the PLC parameter set above with the program to the CPU.

CHAPTER 9 PROGRAMMING INTELLIGENT FUNCTION MODULE

On 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.

Name	Number of I/O occupied points	Function	Module curr consumpti	
Analog-digital converter module (Q64AD)	16 points	Input module that converts; 0 to 20mA \rightarrow 0 to 4000 (in standard resolution mode), 0 to ±10V \rightarrow 0 to ±4000	5VDC	0.63A
		(in standard resolution mode)		0.03A
Digital-analog converter module (Q62DAN)	16 points	Output module that converts; 0 to $4000 \rightarrow 0$ to $20mA$ (in standard resolution mode), 0 to $\pm 4000 \rightarrow 0$ to $\pm 10V$	5VDC 24VDC	0.33A
		(in standard resolution mode)	(0.12A

 Table 9.1
 Example of intelligent function module

9.1 Communication with Intelligent Function Module

The following table shows the communication methods between a QCPU and an intelligent function module.

Table 9.2	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 • Auto refresh setting	Use GX Works2.
Device initial value	(Set data in auto refresh settings is stored to the intelligent function module parameter on a QCPU.) 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 (UD\GD)	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.

9.1.1 Various settings with GX Works2

(1) Switch setting, parameter setting, and auto refresh setting of the intelligent function module

Configuring the switch setting, parameter setting, and auto refresh setting of the intelligent function module enables writing or reading data without creating a communication program which communicates with the intelligent function module.

Also, various settings are available regardless of the buffer memory address of the intelligent function module.

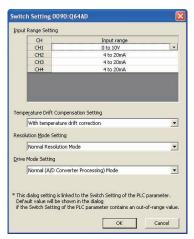
(2) Setting with GX Works2

The following explains the switch setting, parameter setting, and auto refresh setting of the A/D converter module Q64AD.

(a) Switch setting

Configure the switch setting of the intelligent function module. This setting is reflected in the I/O assignment setting of the PLC parameter.

[Switch setting screen]



The set switch setting data is stored in the intelligent function module.

(b) Parameter setting

The initial setting of Q64AD includes the following four types.

- A/D conversion enable/disable setting
- Sampling/averaging processing specification
- Time average/number of times average specification
- Average time/average number of times specification

Configure the initial setting of Q64AD in the following screen.

[Initial setting screen]

Item	Ott	012	00	OH
c setting	Set the A/D conversion s	ystem.		
O conversion enable/disable Itting	0:Enable	1:Disable	1:Disable	1:Disable
mpling/Averaging process Iting	0:Sampling Processing	0.Sampling Processing	0:Sampling Processing	0.5ampling Processing
erage time/Average number of es specification	0:Count Average	0:Count Average	0:Count Average	0:Count Average
verage time/average umber of times	0 Times	0 Times	0 Times	0 Times
verage time/average	0 Times	0 Times	0 Times	0. Times

The set parameter setting data is stored in the intelligent function module.

(c) Auto refresh setting

In the auto refresh setting, set the QCPU-side device for storing the following data.

- Digital output from Q64AD
- Maximum and minimum values of Q64AD
- Error code

Configure the auto refresh setting of Q64AD in the following screen.

[Auto refresh setting screen]

Item		CH1	CH2	СНЗ	CH4
Transfer to CPU			r memory is transmitte	ed to the specified devi	ce.
 Digital output valu Maximum value 	le	D10			
Minimum value					
		200			
Error code		[D20			

The set auto refresh setting data is stored in the intelligent function module.

9.1.2 Communications by the intelligent function module device

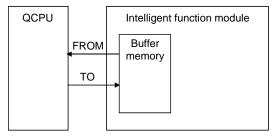
- Intelligent function module device (U□\G□)
 The intelligent function module device represents the buffer memory of the intelligent function module as one of the QCPU module devices.
 Both of the following are available; directly reading the data stored in the buffer memory of the intelligent function module and writing data to the buffer memory.
- (2) Difference from the FROM/TO instruction Since the intelligent function module device can be recognized as a device of the QCPU, reading data from the intelligent function module can be processed with one instruction.

The processing speed is the total of the instruction execution time and the access time with the intelligent function module.

(3) FROM/TO instruction

The FROM instruction stores data read from the buffer memory of the intelligent function module to the specified device.

The TO instruction writes data stored in the specified device to the buffer memory of the intelligent function module.



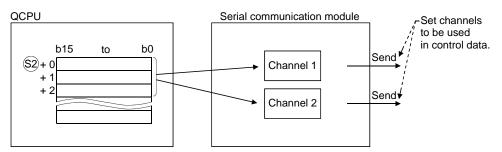
POINT

When reading the data of the intelligent function module for several times in a program, use the FROM instruction at one place and store the data to the data register instead of using the intelligent function module device for every time. The reason is that the intelligent function module accesses the intelligent function module every instruction execution, which extends the scan time of the program.

9.1.3 Communications with the intelligent function module dedicated instruction

This instruction enables easy programming for using functions of the intelligent function module.

For example, the serial communication module dedicated instruction (OUTPUT instruction) allows a data communication in a user-optional message format by nonprocedural protocol.

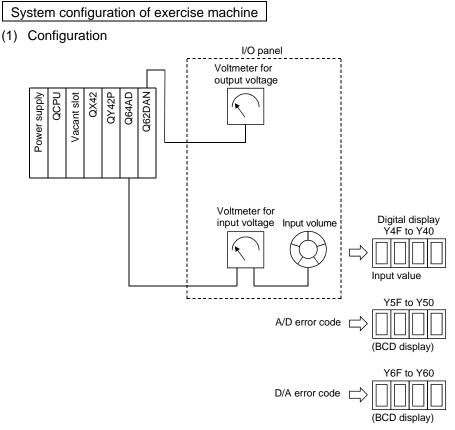


For the intelligent function module dedicated instruction and the completion device, refer to the user's manual of the intelligent function module to be used.

REMARK					
The following table lists the communication timings in the	e commu	unicatior	metho	ds mentione	ed above.
		Co	mmunic	ation timing	
Communication methods with the intelligent function module	Power on	QCPU reset	STOP → RUN	Instruction execution	END processing
Initial setting	0	0	0		
Auto refresh setting					0
Device initial value	0	0	0		
Program created with the FROM/TO instruction				0	
Program created with the intelligent function module device				0	
Program created with the intelligent function module dedicated instruction				0	

9.2 Intelligent Function Module System in Demonstration Machine

9.2.1 Creating an exercise program



(2) Program conditions

<A/D conversion>

The Q64AD executes a sampling processing on analog voltages through CH1, and converts the analog values to digital values.

When the A/D error occurs, the error code is output to the 7-segment display.

- (a) Initial setting
 - A/D conversion enable channel : CH1
- (b) Devices used by users
 - A/D error code reset signal : X0
 - Digital conversion value input from
 - CH1 of the A/D converter module : D10, Y40 to Y4F
 - A/D error code display : D20, Y50 to Y5F

<D/A conversion>

The Q62DAN reduces the digital conversion value of the Q64AD to half and outputs the value from CH1.

When the D/A error occurs, the error code is output to the 7-segment display. (a) Initial setting

Y6F

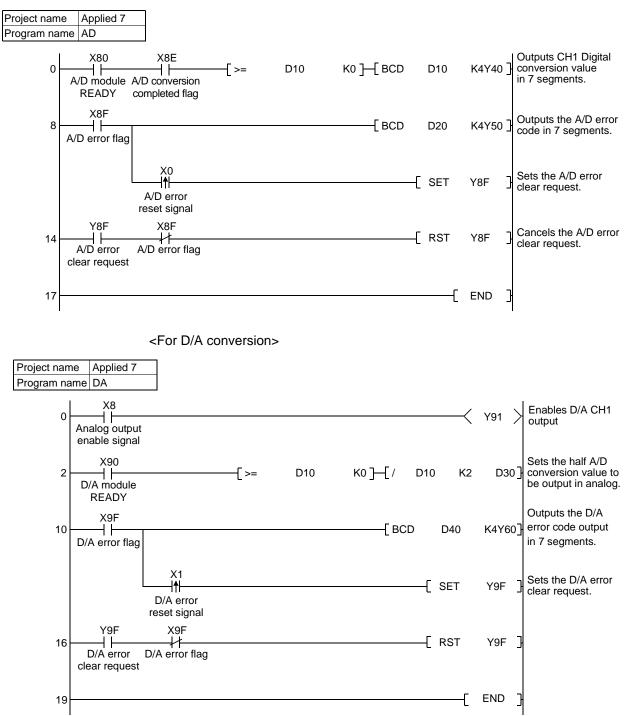
- D/A analog output enable channel : CH1
- (b) Devices used by users

 D/A error code reset signal 	: X1
 Digital output enable signa 	: X8
 Digital value output from CH1 	
of the D/A converter module	: D30
 D/A error code display 	: D40, Y60 to

(3) A program to be created

Create the following programs to check operation easily.

<For A/D conversion>

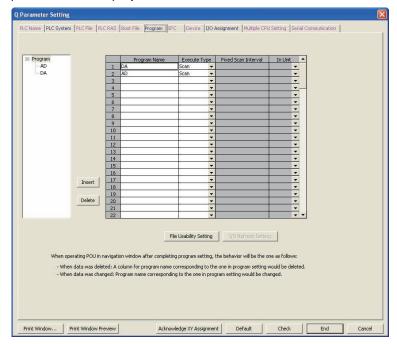


REMARK

When the A/D conversion value is negative, an operation error occurs at the BCD instruction execution. (Refer to section 8.1.1 (2))

(4) Program setting

Configure the setting as shown below in the "Program" tab of the PLC parameter in the project data list.



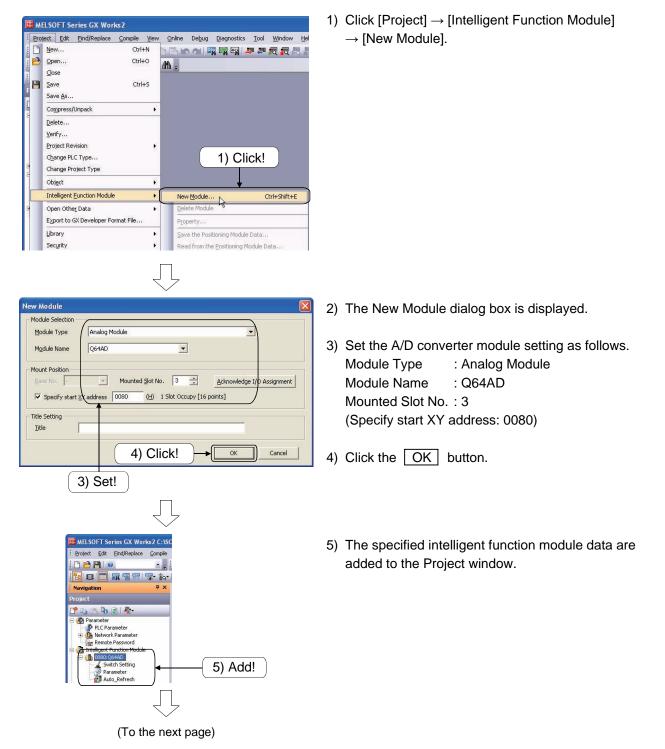
9.2.2 Switch setting, parameter setting, and auto refresh setting for the intelligent function module

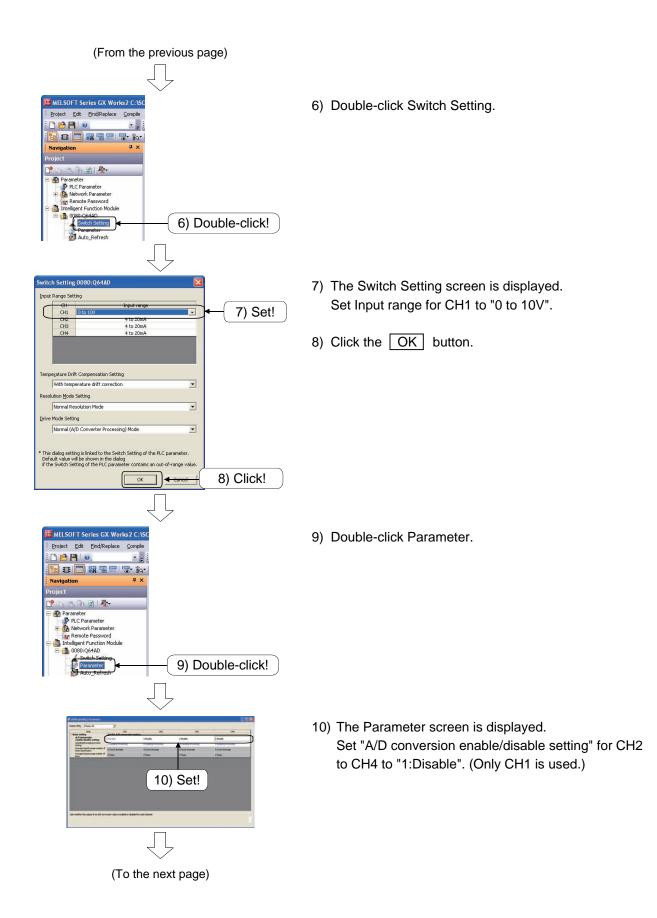
For Q series, the switch setting for the intelligent function module is configured in the I/O assignment settings of GX Works2.

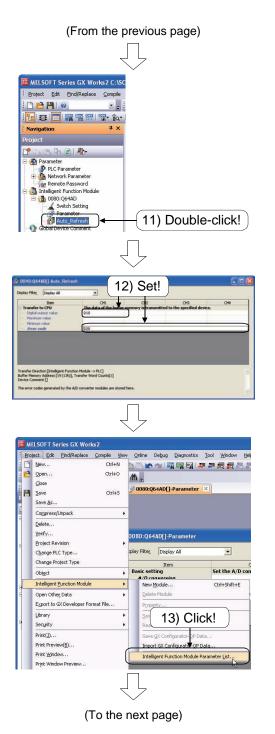
The intelligent function module switches consist of switches 1 to 5 and are set with 16-bit data.

All the default settings of the switches 1 to 5 are 0.

- Adding and setting the 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.
 - (a) Adding and setting method for Q64AD







11) Double-click Auto_Refresh.

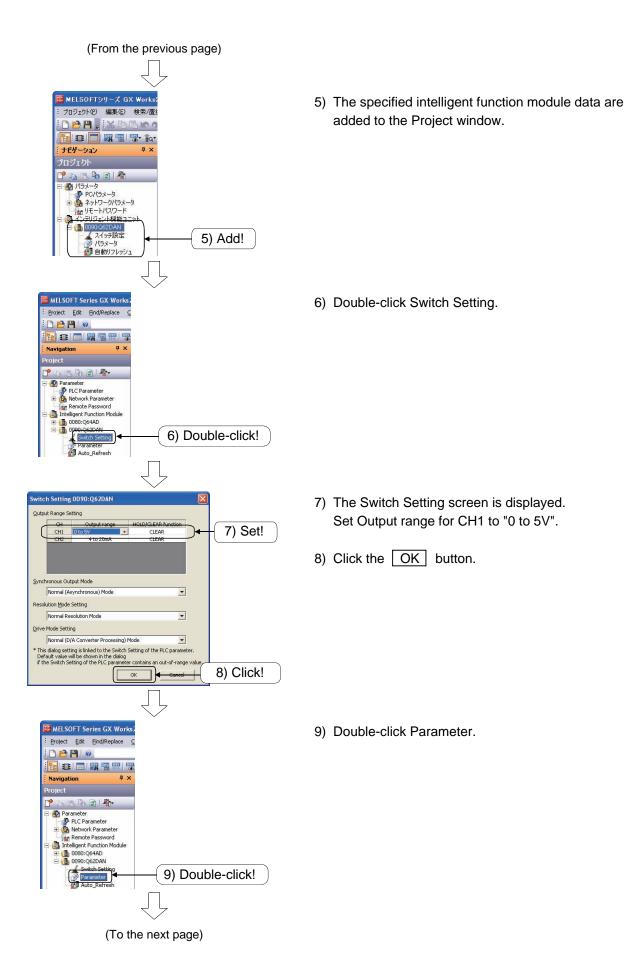
- The Auto_Refresh screen is displayed. Set Digital output value for CH1 to "D10" and Error code for CH1 to "D20".
- 13) Click [Project] \rightarrow [Intelligent Function Module] \rightarrow [Intelligent Function Module Parameter List].

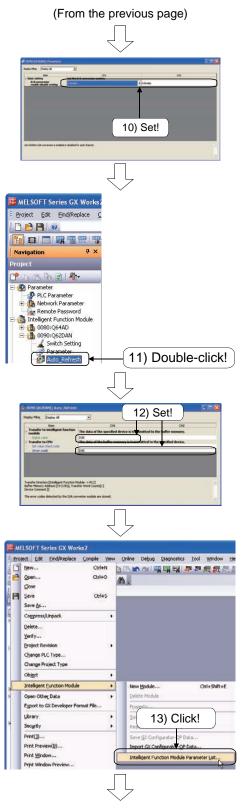
XY Address Module Name Initialization(Count) Auto Refresh(Count) 0080 Q64AD Image: Setting Exist(2) Image: Setting Exist(2)	
Q64AD (✓Setting Exist(2)) ✓Setting Exist(1)	
Explanation Confirm setting status of the intelligent function module, and switch valid/invalid(*) of intelligent function module parameter if necessary.	

- 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.

(b) Adding and setting method for Q62DAN

MELSOFT Series GX Works2	1) Click [Project] \rightarrow [Intelligent Function Module]
Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Hell Wew Ctrl+N Open Ctrl+O Close Save Ctrl+S Save Ctrl+S Save As Compress/Unpack Delete Verify Project Revision Change PLC Type	→ [New Module].
Intelligent Euroction Module New Module Ctrl+Shift+E Gopen Other Data Delete Module Export to GX Developer Format File Property Library Save the Positioning Module Data Security Read from the Positioning Module Data	
New Module Selection	2) The New Module dialog box is displayed.
Module Type Mgdule Name Q62DAN Mount Position Base No. Mounted Slot No. 4 Acknowledge I/D Assignment Specify start & address 0090 (H) 1 Slot Occupy [16 points] Title Setting Itle	 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! Cancel	4) Click the OK button.
3) Set!	
(To the next page)	





(To the next page)

- The Parameter screen is displayed.
 Set "D/A conversion enable/disable setting" for CH1 to "0:Enable". (Only CH1 is used.)
- 11) Double-click Auto_Refresh.

- The Auto_Refresh screen is displayed.
 Set Digital value for CH1 to "D30" and Error code for CH1 to "D40".
- 13) Click [Project] → [Intelligent Function Module]
 → [Intelligent Function Module Parameter List].

	tion Module Parameter :			
XY Address	Module Name	Initialization(Count)	Auto Refresh(Count)	^
0080	Q64AD	Setting Exist(2)	Setting Exist(1)	
0090	Q62DAN	Setting Exist(1)	Setting Exist(2)	
Explanation				
intelligent fu (*Checked	nction module paramete items will be created as	ent function module, and swi r if necessary. intelligent function module p r Setting Count Total		

- 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.

NOTE

Number of parameter settings for intelligent function modules

The number of parameter settings for intelligent function modules (initial setting and auto refresh) is limited according to the programmable controller CPUs and intelligent function modules to be used. Be sure to set the total number of parameter settings of the intelligent function modules within the maximum number of parameter settings of the programmable controller CPUs.

• Limit for the number of parameter settings of the programmable controller CPUs The following table lists the available number of parameter settings for the initial setting and auto refresh that can be set on the programmable controller CPUs.

Application target of	intelligent function module	Max. number of parameter settings			
Application target of		Initial setting	Auto refresh		
	Q00UJ, Q00U, Q01U, Q02U	2048	1024		
	Q03UD, Q03UDE,				
	Q04UDH, Q04UDEH,		2048		
Universal model QCPU	Q06UDH, Q06UDEH,	4096			
Universal model QUI U	Q10UDH, Q10UDEH,				
	Q13UDH, Q13UDEH,				
	Q20UDH, Q20UDEH,				
	Q26UDH, Q26UDEH				

• Limit for the number of parameter settings of the intelligent function modules The following table lists the available number of parameter settings for the initial setting and auto refresh that can be set on the intelligent function modules.

Module type	Module model name	Initial setting (fixed)	Auto refresh (max. number of auto refreshes)
	Q64AD	2	13
Analog module	Q64DAN	1	5
/	Module type Analog module	Module type name	Module type name Initial setting (fixed) Analog module Q64AD 2

For details, refer to the GX Works2 Version 1 Operating Manual (Intelligent Function Module).

REMAR	кк						
Module sv	vitch setting item						
	vitch setting item for	or C					
		ЛС					
			Setting item				
•	Input range setting		Analog input range	Input range setting value			
Switch 1	ПППн		4 to 20mA	Он			
	CH4CH3CH2CH1		0 to 20mA	1н			
			1 to 5V	2н			
	Input range setting		0 to 5V	3н			
Switch 2			-10 to 10V	<u>4н</u>			
Switch 2	∣ЦЦЦЦн		0 to 10V	5H			
	CH8CH7CH6CH5		User range setting	Fн			
Switch 3			Not used				
Switch 4		ור					
	H 00H : With temperature drift correction 01 to FFH : Without temperature drift correction 0H : Normal resolution mode 1 to FH : High resolution mode 0H : Normal mode (A/D conversion processing) 1 to FH : Offset/gain setting mode						
Switch 5			0: Fixed				
Module sw	vitch setting item fo	or C					
Setting item							
Outlate 4	Output range setting		Analog output range	Output range setting value			
Switch 1	ПППП		4 to 20mA	Он			
			0 to 20mA	1н			
	014013012011		1 to 5V	2н			
	Output range		0 to 5V	3н			
	setting		-10 to 10V	4H			
Switch 2	СН8СН7СН6СН5		User range setting	Fн			
Switch 3			CH4CH3CH2CH1 H				
			HOLD/CLEAR function s	etting			
			0H : CLEAR				
			1 to FH : HOLD				
Switch 4][
			01 to FF _H : Synchronized 0H : Normal resolution 1 to FH : High resolution mo 0H : Normal mode (D/A con	mode ode iversion processing)			
<u> </u>			1 to FH : Offset/gain setting mod	de			
Switch 5			0: Fixed				

9.2.3 Operation check and monitor test

(1) Operation check

Run the CPU after resetting to validate the written parameters, then check the operation. (Keep the switch of the CPU No. 2 to STOP.)

• Change input voltages for the A/D converter module with the volume on the demonstration machine.

The digital conversion value is displayed in the 7-segment display (Y40 to Y4F).

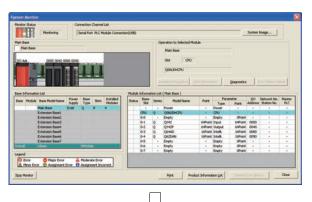
• When X8 is turned on, the D/A OUTPUT voltmeter displays the voltage value that the D/A converter module outputs.

The displayed value is quarter of that of the A/D INPUT voltmeter since the A/D input range is 0 to 10V and the D/A output range is 0 to 5V, and the digital conversion value is operated to half in the CPU.

- When an error occurs, find the cause following (2) Monitor/test described below to solve the error.
- (2) Monitor/test

This section explains how to check the status of the A/D converter module.

(a) Checking method with GX Works2 system monitor



 Participation
 Consume Danseliul

 Participation
 Seerification
 Wenter Image

 Participation
 Seerification
 Wenter Image

 Participation
 Participation
 Wenter Image

 Participation
 Participation
 Wenter Image

 Participation
 Participation
 Participation

 Participation
 Partici

(To the next page)

2) Select "Q64AD" of the slot 3 and click the Detailed Information button.

1) Click [Diagnostics] \rightarrow [System Monitor].

(Or double-click "Q64AD".)

dule's Detailed I	Information	~		
	Monitoring	Nodale Nodel Name 1/O Address Mourt Position Product Information Production Number	Q644D 0080 Man Dase 3 Sot 0612200000000000-C	
		Hodde Information Hodde Access Status of External Power Supply Power Biom Status Status of IJO Address Verify U/O Clear / Hold Setting Noise Pitter Setting Isola Type	Possible In: Agree In: In:	
Error Information	HW Information	Remote Password Setting Status		
Latest Error Code No Error	Update gror History	Contental		8
Deplay Pomet	No. Error Code	Soldard		-
0.0.000	A new part of the second secon	-		

 The information of the selected module (information of Q64AD here) is displayed.

Click the H/W Information button.

Monitoring	Module Model Name (264AD		Product Information 06122000000000-C		
	Display Format				
VW LED Information Rem Value 2 0000	Ren Value	- H/W SW Info	Value	Rem 1 2 3 4 5	Value 0000 0000 0000 0000 0000
pp Monitor [Close

4) The LED status is displayed.

No.	LED name	Status
1	RUN LED	0000н : Off
2	ERROR LED	0001н : On

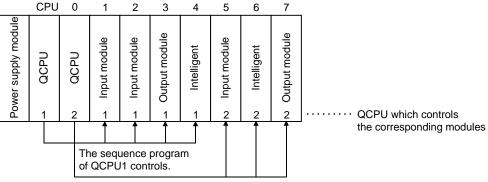
I

CHAPTER 10 HOW TO USE MULTIPLE CPU SYSTEM

10.1 Overview of Multiple CPU System

A multiple CPU system consists of more than one QCPU/motion CPU (up to 4 modules) which is mounted on a main base unit in order to control I/O modules and intelligent function modules separately.

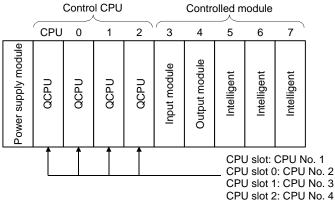
Using more than one QCPU/motion CPU can distribute load produced by the high-load processing and each processing.



The sequence program of QCPU2 controls.

Set each module as follows in the multiple CPU system.

Control CPU	:	QCPU which controls I/O modules and intelligent function modules
Controlled modules	:	I/O modules and intelligent function modules controlled by the control CPU
Non-controlled modules	:	Modules controlled by other CPUs
CPU number	:	Number for identifying multiple QCPUs/motion CPUs mounted on the main base unit. The number 1 is allocated to the CPU slot, and the numbers 2, 3, and 4 are allocated in series.



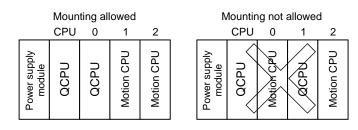
POINT	
 Use the intell CPU system. Intelligent fur 	es can be used in the multiple CPU system. igent function module with the function version B in the multiple action modules with the function version A can be used in the system when the CPU No. 1 is set as the control CPU.

10.2 Difference from Single CPU System

This section explains the differences between the single CPU system and the multiple CPU system.

10.2.1 Mounting position of QCPU/motion CPU

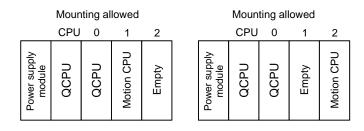
- (1) Up to four QCPUs can be sequentially mounted from the CPU slot (the right-side slot of the power supply module) to the slot No. 2 of the main base unit.
- (2) Motion CPUs (Q17n and Q17nH) can be mounted together on the right-side slots of the QCPU.
 - a) Note that the mountable motion CPUs differ according to the model of the Universal model QCPU. For details, refer to chapter 3 of the QCPU User's Manual (Multiple CPU System).
 - b) The High Performance Model QCPUs or Process CPUs cannot be mounted on the right-side slot of the motion CPUs.



(3) Empty slots can be reserved for future addition of a QCPU/motion CPU (Q17nD). Select the number of CPU modules including an empty slot and set the type of the right-end slot to "PLC (Empty)" in the I/O assignment setting of the PLC parameter.

For the Universal model QCPU (QnUCPU), "PLC (Empty)" can be set between the CPU modules.

Therefore, when a CPU module is added to the system, the CPU No. is not changed, and changing the program is unnecessary.

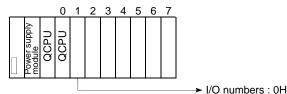


10.2.2 I/O number assignment of the multiple CPU system

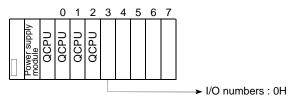
The multiple CPU system is different from the single CPU system in the position (slot) of I/O number 0H.

However, the concept of the order of allocating I/O numbers for the extension base unit, I/O numbers for each slot and empty slots is the same for both types.

- (1) Position of I/O number "0H"
 - (a) The number of slots set with the multiple CPU setting of the PLC parameter are occupied by QCPU/motion CPU in the multiple CPU system.
 - (b) I/O modules and intelligent function modules are mounted from the right of the slots occupied by QCPU/motion CPU modules.
 - (c) The I/O number for an I/O module or intelligent function module mounted to the next slot to the slot occupied by QCPU/motion CPU is set as "0H" and consecutive numbers are allocated sequentially to the right.
 - 1) QCPU: Two CPU modules are mounted



2) QCPU: Four CPU modules are mounted



- 10.2.3 Communication between QCPUs and modules
 - (1) Communication to controlled module/non-controlled modules

The I/O module and intelligent function module controlled by the host CPU can be controlled as well as the single CPU system.

The following table shows the accessibilities to the controlled/non-controlled module.

Access targ	ot	Controlled CPU	Non-controlled CPU		
Access target		Controlled Cr O	Disabled	Enabled	
Input (X)	0 ×			0	
Output (Y) Read		0	×	0 ^{*1}	
	Write	0	×	×	
Buffer memory Read		0	0 ^{*1}	0 ^{*1}	
Duller memory	Write	0	×	×	

The accessibilities for motion CPUs are different from programmable controller CPU, and *1 in the table applies to them.

For details, refer to section 2.1.3 of the MOTION CONTROLLER Programming Manual (COMMON) [Q173D(S)CPU/Q172D(S)CPU] (IB-0300134).

(2) Communication among each QCPU

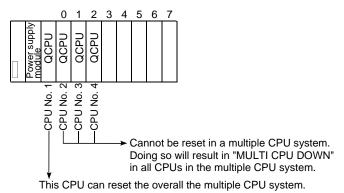
In the multiple CPU system, the following communications are available among each QCPU.

- Auto refresh of the device data between each QCPU/motion CPU with the parameter setting for the multiple CPU system
- Data transfer between QCPUs/motion CPUs with multiple CPU dedicated instructions
- Control instruction from a QCPU to a motion CPU with motion dedicated instructions
- Writing/reading of device data from a QCPU to a motion CPU with the multiple CPU transmission dedicated instruction
- 10.2.4 Reset and operation for errors
 - (1) How to reset a system

The entire multiple CPU system can be reset by resetting the QCPU No. 1. Resetting QCPU No. 1 resets all QCPUs, I/O modules, and intelligent function modules.

If a stop error occurs in any of the QCPU on the multiple CPU system, reset QCPU No. 1 or restart the programmable controller (power supply $ON \rightarrow OFF \rightarrow ON$) for recovery.

(Recovery is impossible by resetting the QCPUs/motion CPUs No. 2 to 4 with the stop error.)



POINT

 It is impossible to reset the QCPUs/motion CPUs of No. 2 to 4 individually in the multiple CPU system.
 If any of QCPUs/motion CPUs is reset during operation of the multiple CPU system, a "MULTI CPU DOWN" (error code: 7000) error occurs in other CPUs, and the entire multiple CPU system stops.
 However, depending on the timing when any of QCPUs/motion CPUs of No. 2 to 4 is reset, an error other than the "MULTI CPU DOWN" may stop other QCPUs.
 A "MULTI CPU DOWN" (error code: 7000) error occurs if QCPUs No. 2 to 4 are reset, regardless of the setting of the operation mode (All station stop by stop error of PLC2 to 4) in the multiple CPU setting of the PLC parameter. (2) Operation for errors

The entire system behaves differently depending whether a stop error occurs in CPU No. 1 or any of CPU No. 2 to 4 in the multiple CPU system.

(a) When a stop error occurs at QCPU No. 1

When a stop error occurs at CPU No. 1, a "MULTI CPU DOWN" (error code: 7000) error occurs at the other QCPUs/motion CPUs No. 2 to 4 and the multiple CPU system is stopped.

Follow the procedures below to restore the system.

- 1) Confirm the error cause with the PLC diagnostics.
- 2) Remove the error cause.
- 3) Reset the QCPU No.1 or restart the power of the programmable controller.

All QCPUs on the entire multiple CPU system are reset and are restored when the QCPU No. 1 is reset or the power of the CPU is reapplied.

(b) When a stop error occurs at QCPUs No. 2 to 4

Set whether to stop the entire system or not in the event of a stop error at the QCPU No. 2 to 4 in "Operation Mode" in the multiple CPU setting of GX Works2.

Follow the procedures below to restore the system.

- 1) Confirm the error-detected CPU No. and error cause in the PLC diagnostics.
- 2) Remove the error cause.
- 3) Reset the QCPU No. 1 or restart the power of the programmable controller.

All QCPUs on the entire multiple CPU system are reset and are restored when the QCPU No. 1 is reset or the power of the CPU is reapplied.

10.3 Communication among each QCPU/Motion CPU in Multiple CPU System

In the multiple CPU system, the following is available;

- Data transfer among each CPU module by the auto refresh of the CPU shared memory
- Reading the CPU shared memory between QCPUs, and the motion CPU shared memory from the QCPU with the multiple CPU dedicated instructions
- Control instruction from a QCPU to a motion CPU with the motion dedicated instructions
- Writing/reading of device data from a QCPU to a motion CPU with the multiple CPU transmission dedicated instructions

10.3.1 CPU shared memory

The CPU shared memory is for transferring data between QCPUs and its capacity is 24336 words of 0H to 5F0FH.

The CPU shared memory consists of five areas; "Host CPU operation information area", "Restricted system area", "Auto refresh area", "User setting area", and "Multiple CPU high speed transmission area".

Configuring the auto refresh setting for the CPU shared memory allows for using the area from 800H to the end point for auto refresh as the auto refresh area.

The start address of the user setting area is next to the end address in the auto refresh area.

When the points for auto refresh is 18 (11 μ), the auto refresh area is 800 μ to 811 μ and the user setting area is 812 μ or later.

The following figure shows the CPU shared memory configuration and the accessibility in the sequence program.

				Host CPU		Other CPUs	
			CPU shared memory	Write	Read	Write	Read
(Он) to (1FFн)	G0 to G511	QCPU standard	Host CPU operation information area	×	0	×	0
(200н) to (7FFн)	to to		System area	×	×	×	0
(800н)			Auto refresh area	×	×	×	×
to (FFFн)	to G4095		User setting area	0	0	×	0
(1000н) to (270Fн)	G4096 to G9999	U	se-prohibited area ^{*1}	×	×	×	×
(2710н) to (5F0Fн)	G10000 to Max. G24335	Multiple CPU high speed transmission area ¹¹		0	0	×	0

○: Communication allowed, ×: Communication not allowed

*1: The Q00UCPU, Q01UCPU, and Q02UCPU do not have the use-prohibited area and the multiple CPU high speed transmission area.

- (1) Host CPU operation information area
 - (a) Information stored in the host CPU operation information area
 - The following information is stored in the host CPU operation information area in the multiple CPU system.^{*1}

They all remain as 0 and do not change in the single CPU system.

*1: For the motion CPU, 5H to 1CH of the host CPU operation information area are not used. When 5H to 1CH of the host CPU operation information area is read from the motion CPU, they are read as "0."

CPU shared memory address	Name	Description	Explanation	Corresponding special register
он	Information availability	Information availability flag	 The area to confirm if information is stored in the host CPU's operation information area (1H to 1FH) or not. O: Information is not stored in the host CPU's operation information area. 1: Information is stored in the host CPU's operation information area. 	-
1H	Diagnostic error	Diagnostic error number	Stores an error number identified during diagnostics in binary.	SD0
2H			Stores the year and month that the error number was stored in the CPU shared memory's 1H address with two digits of the BCD code.	SD1
ЗН	Time the diagnosis error occurred	Time the diagnosis error occurred	Stores the day and time that the error number was stored in the CPU shared memory's 1H address with two digits of the BCD code.	SD2
4H			Stores the minutes and seconds that the error number was stored in the CPU shared memory's 1H address with two digits of the BCD code.	SD3
5H	Error information identification code	Error information identification code	Stores an identification code to determine what error information has been stored in the common error information and individual error information.	SD4
6H to 10H	Common error information	Common error information	Stores the common information corresponding to the error number identified during diagnostic.	SD5 to SD15
11H to 1BH	Individual error information	Individual error information	Stores the individual information corresponding to the error number identified during diagnostic.	SD16 to SD26
1CH	Empty	-	Cannot be used	-
1DH	Switch status	CPU switch status	Stores the CPU module switch status	SD200
1EH	LED status	CPU-LED status	Stores the CPU module's LED bit pattern.	SD201
1FH	CPU operation status	CPU operation status	Stores the CPU module's operation status	SD203

List of host CPU operation information areas

(b) Reading of host CPU operation information area

Other QCPU can use the FROM instruction or multiple CPU area device (U3En\G) to read data from the host CPU operation information area of the host CPU.

However, because there is a delay in data updating, use the read data for monitoring purposes.

(2) Restricted system area

This area is used by the system of the QCPU (OS).

The OS uses this area when the multiple CPU transmission dedicated instruction is executed.

(3) Auto refresh area

This area is used when the multiple CPU system is automatically refreshed. The points from the address next to the last address in the restricted system area are used for auto refresh.

(4) User setting area

This area is for communication between CPU modules.

The points after the ones used for the auto refresh area are used.

(An area including the auto refresh area can be used as the user setting area when auto refresh is not executed.)

(5) QCPU standard area

This area is for the Universal model QCPU to communicate with other CPUs (High Performance QCPU or Process CPU) in a multiple CPU system. This area includes "Host CPU operation information area", "System area", "Auto refresh area", and "User setting area". For each area, refer to (1) to (4).

(6) Multiple CPU high speed transmission area

This area is for communication with other CPU modules in the multiple CPU system using the Universal model QCPU.

The Multiple CPU high speed transmission area includes "auto refresh area" and "user setting area."

(a) Auto refresh area

The area is used when the multiple CPU system is automatically refreshed.

(b) User setting area This area is for storing data to be sent to other CPU modules by the program.

Address for CPU shared memory is 10000H or later.

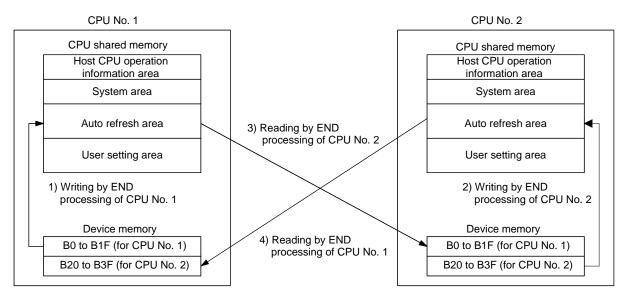
10.3.2 Communication by auto refresh using CPU shared memory

(1) Communication using auto refresh

The auto refresh of the CPU shared memory is executed automatically at the QCPU/motion CPU END processing for the data transfer between each CPU in the multiple CPU system.

As device memory data of other CPUs are automatically read by the auto refresh function, the host CPU can use those device data.

Example) Operation when CPU No. 1 executes auto refresh of 32 points for B0 to B1F, and when CPU No. 2 executes auto refresh of 32 points for B20 to B3F



The processes performed during CPU No. 1 END process

- 1): Transfers B0 to B1F transmission device data for CPU No. 1 to the host CPU shared memory's auto refresh area.
- 4): Transfers data in the CPU No. 2 CPU shared memory's auto refresh area to B20 to B3F in the host CPU.
- The processes performed during CPU No. 2 END process
 - 2): Transfers B20 to B3F transmission device data of CPU No. 2 to the CPU shared memory's auto refresh area.
 - 3): Transfers data in CPU No. 1 CPU shared memory's auto refresh area to B0 to B1F in CPU No. 2.

(2) Executing auto refresh

Auto refresh is executed when the QCPU/motion CPU is in the RUN, STOP or PAUSE status.

Auto refresh cannot be executed when a stop error has been triggered in the QCPU/motion CPU.

If a stop error occurs on one module, the other modules without any error save the data prior to the stop error being triggered

For example, if a stop error occurs in CPU No. 2 when B20 is on, the B20 in CPU No. 1 remains on as shown in Example) in (1).

(3) Settings for auto refresh

The setting for the points to be transmitted by each CPU and the device in which the data is to be stored (the device that executes auto refresh) is configured in the multiple CPU setting of the PLC parameter in GX Works2.

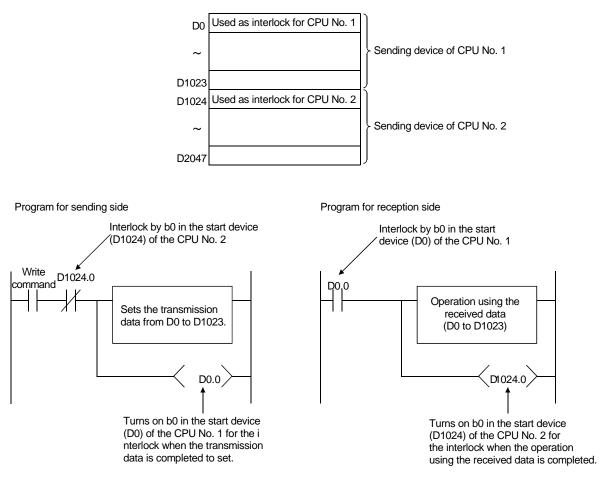
(4) Interlock method for communication by auto refresh

The old data and the new data may be mixed in each CPU due to the timing of a refresh for the host CPU and of reading data from the other CPU.

To execute auto refresh, create an interlock program which uses the start device of devices to be refreshed of each CPU as shown in the following figures. In addition, be careful not to use the data stored in the other CPU when the old data and the new data is mixed.

For example, the following figure shows the program example for the QCPU when the auto refresh setting in the multiple CPU setting is made as follows.

- PLC Side Device : D0
- Send points of CPU No. 1 : 1024 points (D0 to D1023)
- Send points for CPU No. 2 : 1024 points (D1024 to D2047)



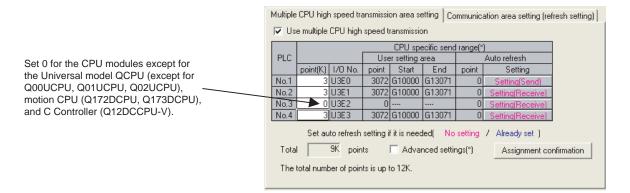
10.3.3 Communication by auto refresh using multiple CPU high speed transmission area

The communication by auto refresh using the multiple CPU high speed transmission area can be executed only when the following conditions are all met.

- The multiple CPU high speed main base unit (Q38DB or Q312DB) is used.
- The Universal model QCPU (except the Q00UCPU, Q01UCPU, Q02UCPU) is used as the CPU No. 1.
- At least two of Universal model QCPUs (except the Q00UCPU, Q01UCPU, and Q02UCPU) and/or motion CPUs (Q172DCPU or Q173DCPU) are used.
- C Controller module (Q12DCCPU-V) is used.

Communication using the multiple CPU high speed transmission area by auto refresh cannot be made with CPU modules other than Universal model QCPUs (except the Q00UCPU, Q01UCPU, and Q02UCPU), C Controller module (Q12DCCPU-V), and Motion CPUs (Q172DCPU and Q173DCPU) mounted on the multiple CPU high speed main base unit.

When any of these modules is mounted on the multiple CPU high speed main base unit, set 0 to the relevant CPU by the "point" field in "CPU specific send range" of "Multiple CPU high speed communication area setting".



(1) Overview of auto refresh

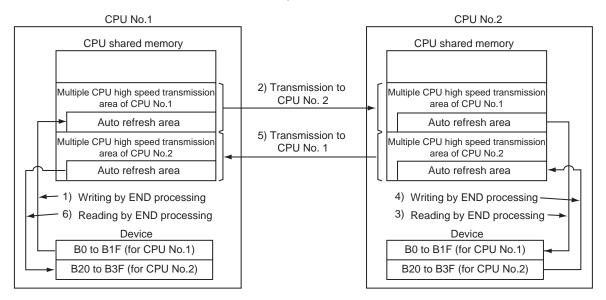
The auto refresh is a communication method using the auto refresh area of the multiple CPU high speed transmission area in the CPU shared memory.

The data written to the auto refresh area of the multiple CPU high speed transmission area is sent to that of the other CPUs in a certain cycle (multiple CPU high speed transmission cycle).

Setting the multiple CPU setting in the PLC parameter allows to automatically read/ write data among all CPUs in the multiple CPU system.

Since device data of other CPUs can be automatically read by the auto refresh function, the host CPU can also use them as those of host CPU.

The following figure shows an overview of operations when CPU No. 1 executes auto refresh of 32 points for B0 to B1F, and when CPU No. 2 executes auto refresh of 32 points for B20 to B3F.



Procedure for the CPU No. 2 to read device data of the CPU No. 1

- 1): Transfers data in B0 to B1F to the auto refresh area of the host CPU at END processing of a CPU No. 1.
- Sends data in the multiple CPU high speed transmission area of CPU No. 1 to CPU No. 2.
- 3): Transfers the received data to B0 to B1F at END processing of CPU No. 2.

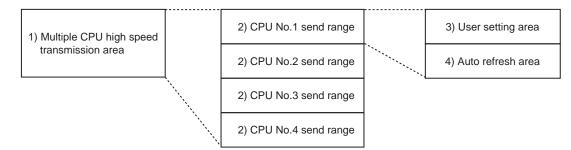
Procedure for the CPU No. 1 to read device data of the CPU No. 2

- 4): Transfers data in B20 to B3F to the auto refresh area of the host CPU at END processing of CPU No. 2.
- 5): Sends data in the multiple CPU high speed transmission area of CPU No. 2 to CPU No. 1.
- 6): Transfers the received data to B20 to B3F at END processing of CPU No. 1.

(2) Executing auto refresh

Auto refresh is executed when the QCPU is in the RUN, STOP or PAUSE status.

(3) Memory configuration of the multiple CPU high speed transmission area The following explains the memory configuration of the multiple CPU high speed transmission area of the CPU shared memory that is used in the multiple CPU high speed transmission function.



No.	Name	Description	Si	ze
NO.	Name	Description	Setting range	Setting unit
1)	Multiple CPU high speed	Area for data transmission between each CPU modules in the multiple CPU system	0 to 14K	1K word
,	transmission area	 The area up to 14K word is divided by each CPU module that constitutes the multiple CPU system 	words	
2)	CPU No. n send area n (n = 1 to 4)	 Area to store the send data of the each CPU module Sends the data stored in the send area of the host CPU to the other CPUs. Other CPU send area stores the data received from the other CPUs. 	0 to 14K words	1K word
3)	User setting area	 Area for data communication with other CPUs using the multiple CPU area device. Can be accessed by the user program using the multiple CPU area device. 	0 to 14K words	2 words
4)	Auto refresh area	 Area for communicating device data with other CPUs by the communication using auto refresh 	0 to 14K words	2 words

POINT

When the COM instruction is used in the sequence program, auto refresh can be executed automatically at the execution of the COM instruction. However, the scan time is prolonged due to the processing time for auto refresh. For details of the COM instruction, refer to the MELSEC-Q/L Programming Manual Common Instruction.

(4) Settings required for auto refresh

To execute auto refresh, setting the number of points to be sent from each CPU module and a device for storing data (device for executing auto refresh) in the multiple CPU setting of the PLC parameter is required.

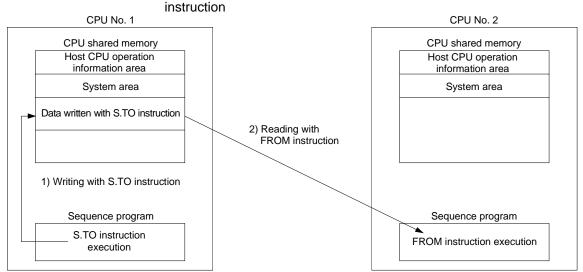
- 10.3.4 Communications by the multiple CPU instruction and motion dedicated instruction
 - (1) Communications by the multiple CPU instruction/intelligent function module device

The QCPU in the multiple CPU system can access the CPU shared memory of QCPU/motion CPU with the S.TO/FROM instruction. Also, the Universal model QCPU can write/read device data to/from another

Universal model QCPU with the multiple CPU high-speed transmission dedicated instruction.

The S.TO instruction is used to write data of the host CPU to the CPU shared memory and the FROM instruction of other CPU is used to read the data. Unlike auto refresh of the CPU shared memory, directly reading the data at instruction execution is available.

Example) When the data written with the S.TO instruction to the CPU shared memory of QCPU No. 1 is read to QCPU No. 2 with the FROM



The processes of QCPU No. 1

1) Writes data in the user setting area of QCPU No. 1 with the S.TO instruction.

The processes of QCPU No. 2

2) Reads the data stored in the user setting area of QCPU No. 1 to the specified device with the FROM instruction.

For details of the S.TO instruction/FROM instruction, refer to the following manual:

MELSEC-Q/L Programming Manual Common Instruction

POINT

Since motion CPUs do not have the S.TO instruction, FROM instruction, and intelligent function module, these instructions are not used for communication between QCPUs and motion CPUs.

For communication between QCPUs and motion CPUs, use auto refresh of the CPU shared memory and the multiple CPU transmission dedicated instruction.

(2) Communication by the motion dedicated instruction The multiple CPU transmission dedicated instruction and multiple CPU high-speed transmission dedicated instruction allows writing data to motion CPUs and reading data via a QCPU. For details and the use of the motion dedicated instruction and multiple CPU

For details and the use of the motion dedicated instruction and multiple CPU transmission dedicated instruction, refer to the Motion CPU Programming Manual.

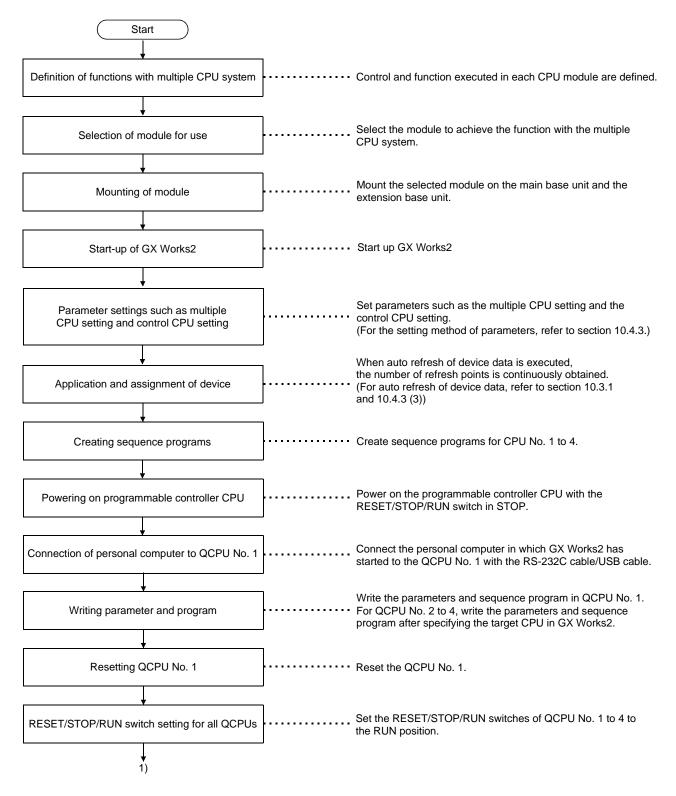
						0	0
POINT							
Control	instructio	ons from a	motion C	PU to c	other motion C	PU is not al	lowed.
 Data tr 	ransfers	with the	multiple	CPU	transmission	dedicated	instructior

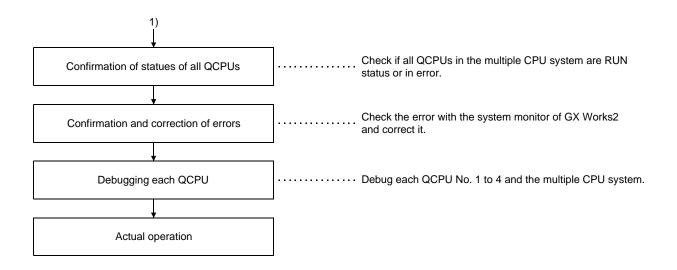
• Data transfers with the multiple CPU transmission dedicated instruction between QCPUs, motion CPU and QCPU, and motion CPUs are not allowed.

10.4 Starting up Multiple CPU System

This Chapter explains the standard start-up procedures for the multiple CPU system.

10.4.1 Procedure for starting up the multiple CPU system

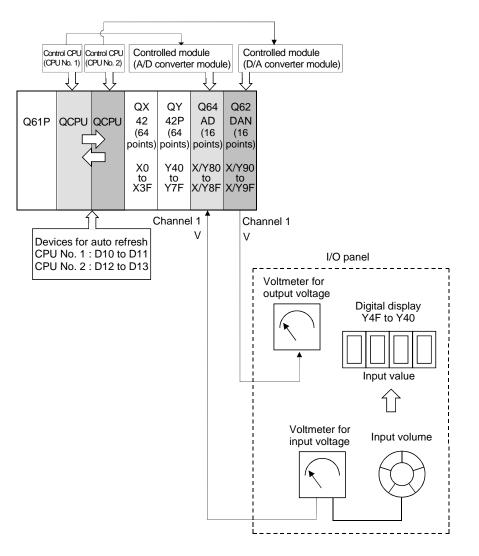




10.4.2 System configuration of the demonstration machine

In order to explain the operation overview of the multiple CPU system, a simple example is used for an exercise.

The following figure shows a system in which the multiple CPUs (CPU No. 1 and No. 2) control different intelligent function modules (A/D and D/A converter module) and transfer the data in the intelligent function modules between the two CPUs.



10.4.3 Creating a program for CPU No. 1

Create a program for checking the operation of the multiple CPU.

Generally, in the multiple CPU system using the auto refresh setting (between CPUs), the old data and the new data may be mixed when multiple data is communicated between CPUs.

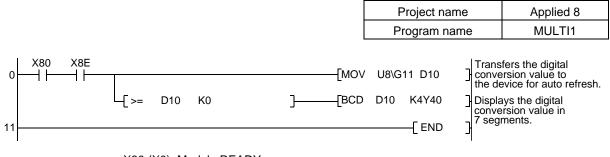
The interlock shown in 10.3.1(4) is required to solve the problem.

In this exercise, the simple program as below is used since the communication data is 1 word which requires no interlock.

Sequence program for QCPU No. 1

The sequence program executes a sampling processing on analog voltages input through CH1 of Q64AD, and then converts the analog values to digital values.

The converted digital value is stored in the device for auto refresh (D10).



X80 (X0): Module READY

X8E (XE): A/D conversion completed flag

U8\G11 (Un\G11): Digital output value from CH1

The multiple CPU system requires the following setting, which is unnecessary for the single CPU system, in the PLC parameter.

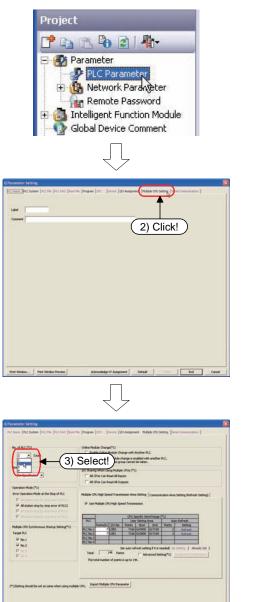
- No. of PLC: Set the number of QCPUs in the multiple CPU system mounted on the main base unit.
- Control PLC: Set the CPU which controls the mounted module.
- Refresh setting: Set the points sent by each CPU and the device for storing data for auto refresh for the device data.

(When auto refresh is not executed, this setting is not required.)

POINT

• The parameter (multiple CPU setting, PLC system (No. of empty slot), and I/O assignment) written into the CPU needs to be the same in all the QCPUs/motion CPUs used in the multiple CPU system.

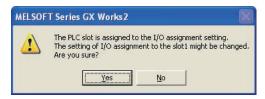
(1) Number of CPUs setting



1) Double-click "PLC Parameter" in the project list of GX Works2.

2) The Q parameter Setting dialog box is displayed. Click the "Multiple CPU Setting" tab.

- 3) Set "No. of PLC" to "2".
- 4) The message below is displayed. Click the Yes button.



Cresk Del

a of FLC (*1) the mobile charge is studied with Fer Specification ... ii mi ii mi 1) Click! talam ... Pres we Selad Cled Did to Refresh Setti PLC No.1 PLC No.2 fresh Device --- Sh V(PLC No.1) 2) Set! Settable Points Available start devices are X,Y,M,L,B,D,W,R,JR,SM,SD,SB,SW Word is used for points. Every 2 points are counted as a set. (*1)50 ng should be set as same when using multiple CPU Check End Cancel

Refre	h Setting					
No.1	PLC NO.2					
lefresh I	bevice Sh	ared Memory(P	LC No.2)			
		Auto Refn	esh	1	CPU Specific	Send Range (U3E1
No	Build Cash	Sat	End	-	Sat	51
1	2	DIZ	D13	-	G17166	617167
2				_		
			-			
3				-		
3						
3 4 5			7 31 54	tl N		
3 4 5 6	_		🤇 3) Se	et!)	-	

POINT

For auto refresh, the following devices can be used.

- Device available for send range: X, Y, M, L, B, D, W, R, ZR, SM, SD, SB, SW
- Device available for receive range: X, Y, M, L, B, D, W, R, ZR

1) Specify the device to be used in the CPU shared memory.

Click the Refresh button in the row of PLC No. 1 in the Multiple CPU High Speed Transmission Area Setting tab.

2) Set the device for storing the data transmitted from CPU No. 1 to the other CPU as follows.

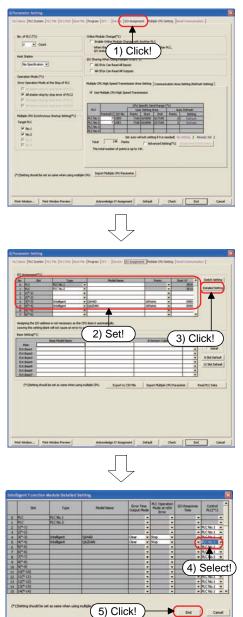
Setting No.1: 2 points, D10

 Click the PLC No.2 tab and set the device for storing the data which CPU No. 1 receives from CPU No. 2 as follows.

Setting No.1: 2 points, D12

(2) Auto refresh setting

(3) Control CPU settings



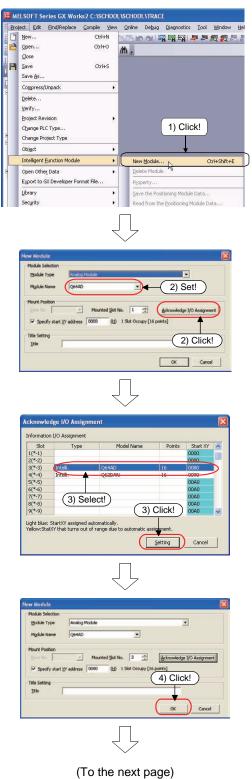
1) Click the "I/O Assignment tab" on the Q Parameter Setting dialog box.

2) Set the I/O assignment to the slots which mount Q62AD and Q62DAN.

Slot	Туре	Model Name	Points	Start XY
3(*-3)	Intelligent	Q64AD	16Points	0080 (Hex.)
4(*-4)	Intelligent	Q62DAN	16Points	0090 (Hex.)

- 3) Click the Detailed Setting button to close the parameter setting screen.
- The Intelligent Function Module Detailed Setting dialog box is displayed. Set "Control PLC" for Q62DAN of the slot "4(*-4)" to "PLC No.2"
- 5) Click the End button to close the parameter setting screen.

- (4) Intelligent function module data setting
 - Set the analog module as described in chapter 9. In this section, set Q64AD which is the controlled module for CPU No. 1.



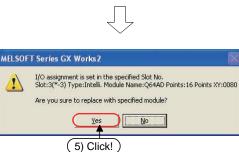
1) Click [Project] \rightarrow [Intelligent Function Module] \rightarrow [New Module].

2) Select "Q64AD" for Module Name. Click the Acknowledge I/O Assignment button.

3) The Acknowledge I/O Assignment dialog box is displayed. Select "Q64AD" and click the Setting button.

 The New Module dialog box is displayed again. Click the OK button.

(From the previous page)



Switch Setting 0080:Q64AD

Input Range Setting CH CH1

CH2 CH3

CH4

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

- 6) Set the switch setting, parameter, and the auto refresh setting as described in chapter 9.
- In the Switch Setting screen, set Input range for CH1 to "0 to 10V".
- 8) In the Parameter screen, set CH1 as follows.

Sampling/Averaging process setting: 1:Average processing Average time/Average number of times: 40 Times

9) In the Auto_Refresh screen, set nothing.



Input rang 0 to 10V

4 to DmA

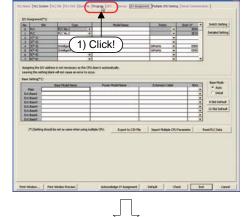
7) Set!

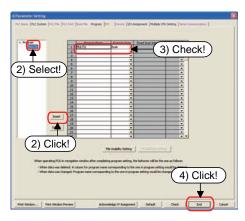
Tame Transfer to CPU	Off	OC ensury is transmitted to the specified i	00	0#
Deptational part of the Mediana calue	The data of the buller of	ensary is transmitted to the specified i	area.	
Home - dat				
Erst ode		<i></i>		
	•			
	(9) Blan	K)		

(5) Program setting

Set the created program as an execution program to prevent a CPU parameter error when multiple programs are included in the same CPU (this setting is optional).

1) Click the "Program" tab on the Q Parameter Setting dialog box.





- 2) Select the program "MULTI1" and click the Insert button.
- Check the program "MULTI1" is set to Scan in Execute Type.
- 4) Click the End button.

Save the created and set program/parameter above.						
	Project name	Applied 8				
	Program name	MULTI1				

10.4.5 Creating a program for CPU No. 2

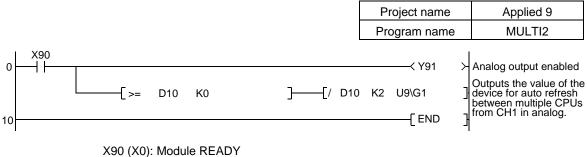
Create a program for QCPU No. 2 in the similar way for QCPU No. 1.

In this exercise, the simple program as below is used since the communication data is 1 word which requires no interlock.

In addition, attach the program to be created here to a different project from the project of the QCPU No. 1.

Sequence program for QCPU No. 2

The sequence program reduces the digital conversion value stored in the device of the QCPU (No. 1) for auto refresh (D10) to half and converts the value into the analog signal from CH1 of Q62DAN.



X90 (X0): Module READY U9\G1 (Un\G1): CH1 Digital value Y91 (Y1): CH1 Output enable/disable flag

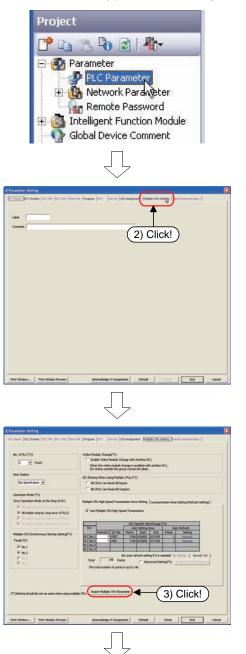
10.4.6 Parameter setting for CPU No. 2

The related items to the multiple CPU system set for CPU No. 1 can be used for the CPU No. 2 with "Import Multiple CPU Parameter" in GX Works2 without re-entering.

POINT

• The parameter (multiple CPU setting, PLC system (No. of empty slot), and I/O assignment) written into the CPU needs to be the same in all the QCPUs/motion CPUs used in the multiple CPU system.

(1) Parameter import



 Double-click "PLC Parameter" in the project list of GX Works2.

2) The Q parameter Setting dialog box is displayed. Click the "Multiple CPU Setting" tab.

3) Click the Import Multiple CPU Parameter button.

Open			
Workspace Location:			
C:\SCHOOL\SCHOOL		Bro	owse
Workspace/Project List:	DICT		
Project Applied 4	Q06UDH	Title	
Applied 5	QOGUDH		
Applied 2	Q06UDH Q06UDH		
Applied 8	QUEUDH		
Applied 9	QOGUDH		
<	101		
(F) Calaati			
(5) Select!	SCHOOL		
Project Name:	Applied 8		
jitle:			
	'		
	(6) Click!)-	Open	Cancel
-	<u> </u>		Calicer
Open a Single File Format F	Switch the win roject when you war	dow by clicking this button It to use single file format pr igator does not support this	oject.
	(MELSOFT Na)	rigator does not support this	s format.)
	2 5		
	\sim		
In the second second	COLLD	2 C	
Import Multipl	e CPU Paramet	er	
	PU parameters are		
all the following p	parameters are ove	erwritten.	
-I/O Ass	ignment Setting		
(I/O Ass	ignment, Base Sett	ing)	
-PLC Sys	tem Setting		
(Points C	Occupied by Empty	Slot)	
and the second	CPU Setting		
Execute the	e multiple CPU para	meter diversion?	
· · · · · ·	Yes	No	
L		NU	
	2 4		
	\sim		
Parameter Setting			8
NUMBER NG System NC/10 NUME Sold	in (Program (1970), (1944)) (U) Autogram	et Multile OV Setting [Seculations and	4 j
N MACTO			71
(8)	Check!)	when R.C. diled with another R.C.	
Hull Station	10 States Man Lang Multi- Old (*1)		
Per Specification 💌	Al Chis Can Read Al Imputs Al Chis Can Read Al Imputs Al Chis Can Read Al Outputs		
Operation Made (*1)			
Drox Operation Mode at the Drop of PLC	Rubple (PUHigh Speed Transmission Area	Setting Communication Area Setting (Refred	Setting
9 Alteria and by the even of RCL 17 Alteria and the second RCL	 Concerning of Children Speed Travers 		
Filmments committee	H.C Uber	millis Seral Range (*1) Setting Anna Auto Rahmah	
Halade CPU Synchrones Startus Selling*1) Target PLC	Pucteral 73/25 120 Marcs	auer End Pares Setting	
P No.1 P No.2	RCNL3 RCNL4		
P MIZ	total 14 Ports	ends setting if it is manufactif. The Service of Alex (*** Advanced Setting(*13)	ady Set 3
- P 1011	The total number of points is up to 14	C	
			ald
Collecting should be set as some of an upper solds	e OU. [Input Midgle OU Farameter]	(9) Cli	ck!
Pet Wedos Pet Windos Presies	Advandudge 11 Ausgement	Default Ows Def	Canal

- 4) The "Open" dialog box is displayed.
- 5) Select the project to import multiple CPU parameters from.

Work space name: SCHOOLProject name: Applied 8

6) Click the Open button.

7) The Import Multiple CPU Parameter dialog box is displayed. Click the Yes (import execution) button.

- The Q Parameter Setting dialog box is displayed again.
 Check No. of PLC is changed from 1 set in the operation procedure 3) to 2.
- 9) Click the End button.

ameter Setting	
new (MCNews (NCTH (NCTA) (Inch	Program
	4
n #RC(*I)	Orites Tedak Overge*1) Utatis Overge Tedak Overge with Another R.C.
2 . Curt (10	0) Click!) and and and and a state of the
at Salar	J) CIICK! Jack Bach
Per Specification •	F 4 Dia Caniteat All Input
- Contraction of the second se	C Al Cha Centreal Al Outputs
Severation Music (*1) Inno Operation Music at the Stop of PLC	Multiple CPU High Speed Transmission Area Setting Communication Area Setting (Rafreds Setting)
The same and is manual (6.1)	2 Une Multiple (200 High Typerd Transmission
9 All dation stop by stop error of PLC2	
F HARMER & BARTER	NC User Setting Area Auto Review
Natale CPU Syndromes (Kartus Selling*1)	HLC Liker Setting Area Auto Indredit Participal 2018a Parts Bart End Parts Setting PLCNa II (2018a Parts Table (2000) 557768 2 Endedit
laget PLC	PLC No. 2 7 JURO 7046 S20000 SUT148 2 fortunt
Photo Photo	PLC No. 4
(P (s))	Set auto-onlinety anticege in a second and the learning of a basely list of a
10 (c) (The table number of posts is up to 140.
	A CONTRACT OF A
Stating should be set as uses when using multipl	in CHU. [Input Hullpin CHU Parameter]
Sating should be set as same often using multip	n ORL [Input Huliph OUTeranete
Sletting should be set as some offer using sullip	n (HL) (İlgəri Məlgili (HL) Fərəndər)
	e (OL) (Signet Malgin (OL) Fargurenter)
California Post Window Province	
f Window	
f Window	Alternational for frances of the second
f Window	Atomining of Kanggawa (Make Diritory (Section of Section of Sectio
f Window	Atomining of Kanggawa (Make Diritory (Section of Section of Sectio
f Window	Atomitique for fragment (bala) Out (bet) Const Program (C) (beta (balance) (bala) OU Strag (beta (b
e sette factore meter factore	Atomickey tri foregoine (Debalt Order Ord
e sette factore meter factore	Atomicking til fangement • Pageer (nr. (sono (10) Analyseent (Packer Oth Delta) (sono (100) Analyseent (• Call (10) Ana
e sette factore meter factore	Attendenty of Kanggood (Date) On (String Section concess)
etetalen: net tinden former meter Setting time (PC2)ener (PC2) (PC2) (pc2) (pc2)ener (PC2) (pc2) (pc2) (pc2) (pc2)ener (pc2) (pc2) (pc2) (pc2) (pc2) (pc2)ener (pc2) (pc2	Atomiting IT fangunger Control (12) Check!
needed letting. The face bases (Incluse (Incluse Incluse Incluse Incluse (Incluse Incluse Inc	Atomickiji ti folgovel (blak) Ovel (bl.) b. Parjen (r.)
events forting	Atomiting IT fangunese Debalt Other Debalt Othe
events ketter: no factores factores 1) Select1	Atomiting of daysers Dated Div (Hing Sectioneerses . Pages rc. sec 10 Augusta Patho Div (Hing Sectioneerses . Pages rc. sec 10 Augusta Patho Div (Hing Sectioneerses

13) Click!

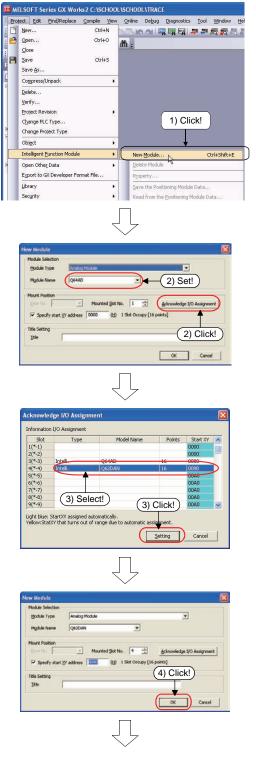
11) Click!

- Pres W

10) Click "Program" tab on the Q Parameter Setting dialog box. (Although this procedure is optional, it prevents a CPU error.)

- 11) Select the program "MULTI2" and click the Insert button.
- 12) Check the program "MULTI2" is set to Scan in Execute Type.
- 13) Click the End button.

(2) Intelligent function module data setting Set the analog module as described in chapter 9. In this section, set Q62DAN.



(To the next page)

1) Click [Project] \rightarrow [Intelligent Function Module] \rightarrow [New Module].

2) Select "Q64DAN" for Module Name. Click the Acknowledge I/O Assignment button.

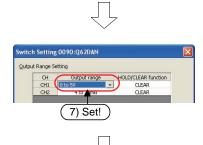
 The Acknowledge I/O Assignment dialog box is displayed. Select "Q64DAN" and click the Setting button.

4) The New Module dialog box is displayed again. Click the OK button.

(From the previous page)

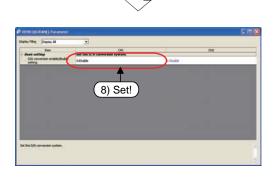


MELSOFT Series GX Works 2 I/O assignment is set in the specified Slot No. Sickt (1-4) Type:Intell. Module Name:(620AN Points:16 Points XY:0090 Are you sure to replace with specified module? Yes No (5) Click!



- 5) The message on the left is displayed. Click the Yes button.
- 6) Set the switch setting, parameter, and the auto refresh setting as described in chapter 9.
- In the Switch Setting screen, set Input range for CH1 to "0 to 5V".
- 8) In the Parameter screen, set CH1 as follows.

D/A conversion enable/disable setting: 0:Enable



- ODDOCACANY | Auto Staffresh

 Ord

 O
- 9) In the Auto_Refresh screen, set nothing.

Save the created and set program/parameter above.					
	Project name	Applied 9			
	Program name	MULTI2			
1					

CRO

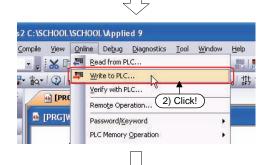
(1) Check!)

]]]] *****

Write the created sequence program and parameter setting in each QCPU. Set the RUN/STOP/RESET switches of CPUs (both No. 1 and No. 2) to "STOP".

Write the created program	for QCPU No. 1 with GX Works2.
	(Program name) MULTI1

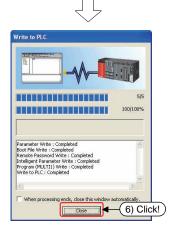
1) Check Target System is "PLC No.1" in the Transfer Setup screen.



2) Click [Online] \rightarrow [Write to PLC].



- 3) The Online Data Operation dialog box is displayed.
- 4) Click the Parameter + Program button.
- 5) Click the Execute button.



6) When the writing to QCPU No. 1 is completed, the dialog box on the left is displayed. Click the Close button.

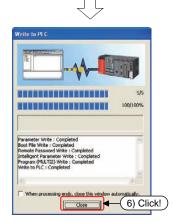
(2) Writing data into QCPU No. 2

Write the created program for QCPU No. 2 with GX Works2. (Project name) Applied 9, (Program name) MULTI2

- 1000 Target PLC PLC No. (1) Set!) 2 C:\SCHOOL\SCHOOL\Applied 9 View Online Debug Diagnostics Compile Tool Window Help Pead from PLC. X -Write to PLC.. HTP SF7 1 a . Verify with PLC.. EPRC (2) Click!) Remote Operation... \min [PRG]W Password/Keyword . PLC Memory Operation ۲
- Control Data Stands
 Section 2011

 Simplify Including Stands
 Section 2012

 Simpl



- 1) Check Target System is "PLC No.2" in the Transfer Setup screen.
- 2) Click [Online] \rightarrow [Write to PLC].

- 3) The Online Data Operation dialog box is displayed.
- 4) Click the Parameter + Program button.
- 5) Click the Execute button.

6) When the writing to QCPU No. 2 is completed, the dialog box on the left is displayed. Click the Close button.

10.4.8 Operation check

Check the operation with the following procedure.

- (1) Reset of the multiple CPU
 - 1) Reset the QCPU No. 1.
 - 2) Set RUN/STOP/RESET switches of QCPUs No. 1 and No. 2 to RUN.
- (2) Operation check
 - 1) Change input voltages for the A/D converter module with the volume on the demonstration machine.

The digital conversion value is displayed in the 7-segment display (Y40 to Y4F).

2) The D/A OUTPUT voltmeter displays the voltage value that the D/A conversion module outputs.

The displayed value is quarter of that of the A/D INPUT voltmeter since the A/D input range is 0 to 10V and the D/A output range is 0 to 5V, and the digital conversion value is operated to half in the CPU.

D ———	
-	

APPENDIX

Appendix 1 Instruction Tables

For instructions related to SFC, refer to the MELSEC-Q/L/QnA Programming Manual (SFC) SH-080041.

Appendix 1.1 Application instruction

(1) Logical operation instructions

Category	Instruction symbol	Symbol	Processing details	Execution condition	Number of basic steps	Subset
	WAND	WAND S D	\cdot (D) \land (S) \rightarrow (D)		3	
Logical AND operation	WANDP	WANDP S D			5	•
	WAND	WAND S1 S2 D	· (S1) ∧(S2) → (D)		4	•
	WANDP	WANDP S1 S2 D			4	*2
	DAND	- DAND S D -	$\cdot (D+1,D) \land (S+1,S) \rightarrow (D+1,D)$		*4	
	DANDP	DANDP S D			*1	
	DAND	DAND S1 S2 D	· (S1+1,S1) ∧ (S2+1,S2) → (D+1,D)		*4	•
	DANDP	DANDP S1 S2 D			*1	*2
	BKAND	BKAND S1 S2 D n	(S1) (S2) (D)		-	
	BKANDP	BKANDP S1 S2 D n			5	
	WOR	WOR S D	\cdot (D) \bigvee (S) \rightarrow (D)		3	
	WORP	WORP S D			3	•
	WOR	- WOR S1 S2 D -	$(S1) \bigvee (S2) \rightarrow (D)$		4	•
	WORP	WORP S1 S2 D			4	*2
Logical OR	DOR	DOR SD	$\cdot (D+1,D) \searrow (S+1,S) \rightarrow (D+1,D)$		*1	
operation	DORP	DORP S D			1	
	DOR	DOR S1 S2 D	\cdot (S1+1,S1) \bigvee (S2+1,S2) \rightarrow (D+1,D)		*1	•
	DORP	DORP S1 S2 D			1 "1	*2
	BKOR	BKOR S1 S2 D n	(S1) (S2) (D) ▲ ■ ■ ■ ■ ■ ■ ■		_	
	BKORP	BKORP S1 S2 D n			5	

REMARK

*1: The number of steps varies depending the device and CPU type to be used.

Device	Number of step		
Device		QnACPU	
 Word device: Internal device (except for the file register ZR) Bit device: Devices whose device Nos. are multiples of 16, whose digit designation is K8, and which use no Indexing 	6	6	
Constant: No limitations			
Devices other than above	4		

*2: Only QCPU supports the subset.

Category	Instruction symbol	Symbol	Processing details	Execution condition	Number of basic steps	Subset
	WXOR	WXOR SD-	$(D) \rightarrow (D) \rightarrow (D)$		3	
	WXORP	WXORP S D			5	Ū
	WXOR	WXOR S1 S2 D	\cdot (S1) \checkmark (S2) \rightarrow (D)		4	•
	WXORP	WXORP S1 S2 D			4	*2
Exclusive	DXOR	DXOR S D	$(D+1,D) \rightarrowtail (S+1,S) \rightarrow (D+1,D)$		*1	
OR	DXORP	- DXORP S D-			1	•
	DXOR	DXOR S1 S2 D	$(S1+1,S1) \rightarrow (S2+1,S2) \rightarrow (D+1,D)$		*1	•
	DXORP	DXORP S1 S2 D			1	*2
	BKXOR	BKXOR S1 S2 D n	(S1) (S2) (D) ■ ■ ■ ■ ↑		F	
	BKXORP	BKXORP S1 S2 D n			5	
	WXNR	WXNR SD	$\overline{(D)} (S) \rightarrow (D)$		3	•
	WXNRP	WXNRP S D			3	
	WXNR	WXNR S1 S2 D	$\overline{(S1)} (S2) \rightarrow (D)$		4	•
NON	WXNRP	WXNRP S1 S2 D			4	*2
	DXNR	DXNR SD	$\cdot \overline{(D+1,D)} \underbrace{\rightarrowtail} (S+1,S) \to (D+1,D)$		*1	
logical	DXNRP	DXNRP S D			1	
sum	DXNR	DXNR S1 S2 D	$\cdot \overline{(S1+1,S1)} \underbrace{ (S2+1,S2)}_{\to} (D+1,D)$		*1	•
	DXNRP	DXNRP S1 S2 D			I	*2
	BKXNOR	BKXNOR S1 S2 D n	(S1) (S2) (D) ▲		5	
	BKXNORP	BKXNORP S1 S2 D n			5	

REMARK

*1: The number of steps varies depending the device and CPU type to be used.

Device		of steps
		QnACPU
Word device: Internal device (except for the file register ZR)		
· Bit device: Devices whose device Nos. are multiples of 16, whose	6	
digit designation is K8, and which use no Indexing	0	6
Constant: No limitations		
Devices other than above	4	

*2: Only QCPU supports the subset.

Category	Instruction symbol	Symbol	Processing details	Execution condition	Number of basic steps	Subset
	ROR	ROR D n	b15 (D) b0 SM700		3	
Right	RORP	RORP D n	Right rotation (n bit)		5	·
rotation	RCR	RCR D n	b15 (D) b0 SM700		- 3	
	RCRP	RCRP D n	Right rotation (n bit)		5	·
	ROL	ROL D n	SM700 b15 (D) b0		3	
Left	ROLP	ROLP D n	Left rotation (n bit)		0	·
rotation	RCL	RCL D n	SM700 b15 (D) b0		3	
	RCLP	RCLP D n			5	·
	DROR	DROR D n	(D+1) (D) b31 to b16 b15 to b0 SM700		- 3	
Right	DRORP	DRORP D n	Right rotation (n bit)		5	·
rotation	DRCR	DRCR D n	(D+1) (D) b31 to b16 b15 to b0 SM700		3	
	DRCRP	DRCRP D n	Right rotation (n bit)		5	
	DROL	DROL D n	(D+1) (D) SM700 b31 to b16 b15 to b0		2	
Left	DROLP	DROLP D n	Left rotation (n bit)		3	•
rotation	DRCL	DRCL D n	(D+1) (D) SM700 b31 to b16 b15 to b0		3	
	DRCLP	- DRCLP D n	Left rotation (n bit)		5	J

(2) Rotation instructions

Category	Instruction symbol	Symbol	Processing details	Execution condition	Number of basic steps	Subset
	SFR	- SFR D n -	b15 bn b0			
n-bit shift -	SFRP	- SFRP D n -	b15 b0 SM700		3	•
	SFL	- SFL D n -	b15 bn b0			
	SFLP	SFLP D n	SM700 b15 60		3	•
	BSFR	BSFR D n	(D)			
	BSFRP	BSFRP D n	SM700		3	
1-bit shift	BSFL	BSFL D n	(D)		- 3	
	BSFLP	BSFLP D n	SM700 SM700		3	
	DSFR	DSFR D n	(D)		- 3	
1-word	DSFRP	DSFRP D n			3	
shift	DSFL	– DSFL D n	(D)			
	DSFLP	- DSFLP D n			3	•

(3) Shift instructions

(4) Bit processing instructions

Category	Instruction symbol	Symbol	Processing details	Execution condition	Number of basic steps	Subset
	BSET	BSET D n	(D) b15 bn b0		2	
Bit	BSETP	BSETP D n	▲1		3	•
set/reset	BRST	BRST D n	(D) b15 bn b0			
-	BRSTP	BRSTP D n			3	•

Category	Instruction symbol	Symbol	Processing details	Execution condition	Number of basic steps	Subset
	TEST	TEST S1 S2 D	(S1) b15 to b0 (D)		4	
	TESTP	TESTP S1 S2 D	Bit specified by (S2)		-	
Bit tests	DTEST	DTEST S1 S2 D	(S1) b31 to b0 (D)			
	DTESTP	DTESTP S1 S2 D	Bit specified by (S2)			
Batch reset of	BKRST	BKRST S n	(S) ON (S) OFF OFF OFF		2	
bit devices	BKRSTP	BKRSTP S n	ON ON OFF		3	

(5)	Data processing	instructions
-----	-----------------	--------------

Category	Instruction symbol	Symbol	Processing details	Execution condition	Number of basic steps	Subset
Data	SER	- SER S1 S2 D n	(S1) (S2)		5	
	SERP	- SERP S1 S2 D n	→(D) : Position of match (D+1) : Number of matches			
searches	DSER	- DSER S1 S2 D n	32 bits (S1) (S2)			
	DSERP	DSERP S1 S2 D n	(D) : Position of match (D+1) : Number of matches		5	
	SUM	- SUM S D-	(S) b15 b0		3	
Bit	SUMP	SUMP S D	(D) : Number of bits where 1 is set		3	
checks	DSUM	- DSUM S D-	(S+1) (S)		3	•
	DSUMP	DSUMP S D	► (D) : Number of bits where 1 is set		5	•
Decede	DECO	DECO S D n	$8 \rightarrow 256 \text{ decode}$ (S) Decode (D)			
Decode	DECOP	DECOP S D n	$(S) \xrightarrow{\text{Decode}}		4	
E	ENCO	ENCO SDn	$256 \rightarrow 8 \text{ decode}$ (S)			
Encode	ENCOP	- ENCOP S D n	$ \begin{array}{c} \hline \\ \hline \\ \\ \hline \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ $		4	

Category	Instruction symbol	Symbol		Processing details	Execution condition	Number of basic steps	Subset
7-segment decode	SEG SEGP	- SEG S D - SEGP S D -		b3 to bO (S) 7SEG (D) 7		3	•
	DIS DISP	- DIS S D n - DISP S D n -	•	Separates 16-bit data specified by (S) into 4-bit units, and stores it at the lower 4 bits of n points from (D). ($n \le 4$)		4	
Separating - and linking	UNI UNIP	UNI SDn UNIP SDn	•	Links the lower 4 bits of n points from the device specified by (S) and stores it at the device specified by (D). ($n \le 4$)		4	
	NDIS NDISP	- NDIS S1 D S2-	•	Separates the data in the devices specified by (S1) into bits specified by the devices from (S2), and stores them to the devices		-	
		- NUNI S1 D S2	•	 following the device specified by (D). Links the data in the devices following the device specified by (S1) with bits specified by the devices from (S2), and 		4	
	NUNIP	- NUNIP S1 D S2 - WTOB S D n -		stores them to the devices following the device specified by (D). Breaks n points of 16-bit data from the			
	WTOB WTOBP	- WTOB S D n WTOBP S D n -		device specified by (S) into 8-bit units, and stores them to the device specified by (D).		-	
	BTOW BTOWP	BTOW S D n BTOWP S D n	•	Links the lower 8 bits of 16-bit data of n points from the device specified by (S) into 16-bit units, and stores them to the device BTOWP specified by (D).		4	
	MAX MAXP	- MAX S D n - MAXP S D n -	•	Searches for the data of n points from the device specified by (S) in 16-bit units, and stores the maximum value to the device specified by (D).		-	
	MIN	- MIN S D n	•	 Searches for the data of n points from the device specified by (S) in 16-bit units, and stores the minimum value to the device 		4	
Search	MINP DMAX	- MINP S D n - - DMAX S D n -	•	specified by (D). Searches for the data of $2 \times n$ points from the device specified by (S) in 32-bit			
	DMAXP	DMAXP S D n		units, and stores the maximum value to the device specified by (D).		4	
	DMIN	- DMIN S D n	•	Searches for the data of 2 \times n points from the device specified by (S) in 32-bit units, and stores the minimum value to			
	DMINP	_ DMINP S D n _		the device specified by (D).			

Category	Instruction symbol	Symbol	Processing details	Execution condition	Number of basic steps	Subset
	SORT	 SORT S1 n S2 D1 D2 S2 : Number of comparisons for single run D1 : Device to be turned on at completion of sort D2 : For system use 	 Sorts data of n points from device specified by (S1) in 16-bit units. (n × (n-1)/2 scans are required) 		0	
Sort	DSORT	DSORT S1 n S2 D1 D2 S2 : Number of comparisons for single run D1 : Device to be turned on at completion of sort D2 : For system use	 Sorts data of 2 × n points from device specified by (S1) in 32-bit units. (n × (n-1)/2 scans are required) 		6	
	WSUM	WSUM S D n	 Adds 16-bit BIN data of n points from the device 			
Total value	WSUMP	WSUMP S D n	specified by (S), and stores it to the device specified by (D).		4	
calculations	DWSUM	- DWSUM S D n	 Adds 32-bit BIN data of n points from the device 			
	DWSUMP	- DWSUMP S D n	specified by (S), and stores it to the device specified by (D).			

(6) Structure creation instructions

Category	Instruction symbol	Symbol	Processing details	Execution e condition R	basic steps Subset
Number of	FOR NEXT	FOR n NEXT	Executes n times between the FOR and NEXT.	2	_
repeats	BREAK	BREAK D Pn	Forcibly ends the execution of the FOR to NEXT cycle and jumps		
	BREAKP	BREAKP D Pn	pointer Pn.Executes subroutine program Pn when		
	CALL	CALL Pn S1 to Sn	the input condition is met.		
	CALLP	CALLP Pn S1 to Sn	(S1 to Sn are arguments sent to subroutine program. $0 \le n \le 5$)	2+	n
	RET	RET	Returns from the subroutine program	1	
	FCALL	FCALL Pn S1 to Sn	Executes non-execution processing of the subroutine program of Pn when	×1	
	FCALLP	FCALLP Pn S1 to Sn	input conditions have not been met.		11
Subroutine program calls	ECALL	ECALL * Pn S1 to Sn *: Program name	 Executes subroutine program Pn in the specified program name when the 		>
	ECALLP	ECALLP * Pn S1 to Sn *: Program name	input condition is met. (S1 to Sn are arguments sent to subroutine program. $0 \le n \le 5$)	3+	n
	EFCALL	EFCALL * Pn S1 to Sn *: Program name	 Executes non-execution processing of subroutine program Pn in the specified 	*2	<u>,</u>
	EFCALLP	EFCALLP * Pn S1 to Sn *: Program name	program name when input conditions have not been not met.	3+	n
	СОМ	СОМ	 Executes link refreshing and general data processing. 	1	
	іх	IX S Device indexing ladder	 Executes indexing for individual devices used in the device indexing 	2	
Fixed	IXEND	IXEND	ladder.	1	
indexing	IXDEV	IXDEV	Stores the indexing value used for indexing executed between the IX	1	
	IXSET	Specifies index value	and IXEND to the device specified by D or later.	3	

*1: n indicates number of arguments for the subroutine program.

*2: n indicates the total number of arguments used in the subroutine program and program name steps.

The number of program name steps is calculated as "number of characters in the program/2" (decimal fraction is rounded up).

Category	Instruction symbol	Symbol	Processing details	Number of Subset
Table processing	FIFW	- FIFW SD-	(S) (D) Pointer Pointer + 1	
	FIFWP	- FIFWP SD-	Device at pointer + 1	3
	FIFR	- FIFR SD-	(S) Pointer Pointer - 1 (D)	
	FIFRP	- FIFRP SD-		
	FPOP	FPOP SD-	(S) Pointer Pointer - 1 (D)	
	FPOPP	FPOPP S D	Device at pointer + 1	
	FINS	- FINS SDn-	(S) (D) Pointer Pointer + 1	
	FINSP	FINS SDn	Specified by n	4
	FDEL	- FDEL SDn-	(S) Pointer Pointer -1 (D)	
	FDELP	- FDELP S D n	Specified by n	4

(7) Table operation instructions

(8) Buffer memory access instructions

Category	Instruction symbol	Symbol	Processing details	Execution condition	Number of basic steps	Subset
	FROM	FROM n1 n2 D n3	Reads data in 16-bit units from a special function module.		- 5	
Data	FROMP	- FROMP n1 n2 D n3-			5	
read	DFRO	DFRO n1 n2 D n3	Reads data in 32-bit units from a special function module.		5	
	DFROP	DFROP n1 n2 D n3			5	
	то	- TO n1 n2 S n3-	· Writes data in 16-bit units to a special		5	
Data	ТОР	- TOP n1 n2 S n3-	function module.		5	
write	DTO	DTO n1 n2 S n3	• Writes data in 32-bit units to a special		_	
	DTOP	DTOP n1 n2 S n3	function module.		- 5	

(9) Display instructions

Category	Instruction symbol	Symbol	Processing details	Execution condition	Number of basic steps	Subset
	PR	* SM701: OFF - PR S D	 Outputs ASCII code of 8 points (16 characters) from the device specified by (S) to the output module. 			
ASCII print	PR	* SM701: ON PR S D	 Outputs ASCII code from device specified by (S) to 00H to the output module. 		3	
	PRC	PRC SD	 Converts comments of the device specified by (S) into ASCII code and outputs them to the output module. 			
Display	LED	- LED S-	 Outputs ASCII code of 8 points (16 characters) from the device specified by (S) to the LED indicator at the front of the CPU. 	ſ	2	
	LEDC	LEDC S	 Outputs the comment from the device specified by (S) to the LED indicator at the front of the CPU. 			
Reset	LEDR	- LEDR -	 Resets the annunciator and LED indicator display. 		1	

	1			1		
Category	Instruction symbol	Symbol	Processing details	Execution condition	Number of basic steps	Subset
Checks	CHKST	- CHKST	 The CHK instruction is executed when CHKST is executed. Jumps the execution to the step following the CHK instruction when CHKST is not executed. 		1	
	СНК	CHK Check condition	 During normal conditions → SM80: OFF, SD80: 0 During abnormal conditions → SM80: ON, SD80: Failure No. 			
	CHKCIR	- CHKCIR -	 Starts update in ladder pattern to be checked with the CHK instruction. 		1	
	CHKEND	- CHKEND -	 Ends update in ladder pattern to be checked with the CHK instruction. 		I	
Status	SLT	– SLT –	Executes the status latch.			
latch	SLTR	- SLTR -	 Resets the status latch to enable the re-execution. 		1	
Sampling	STRA	- STRA -	Sets a trigger for the sampling trace.			
trace	STRAR	- STRAR	 Resets the sampling trace to enable the re-execution. 		1	
	PTRA	- PTRA -	Sets a trigger for the program trace.	_		
Program	PTRAR	- PTRAR	Resets the program trace to enable the re-execution.		1	
trace	PTRAEXE	- PTRAEXE -	 Executes the program trace. 		- 1	
	PTRAEXEP	PTRAEXEP				

(10) Debugging and failure diagnosis instructions

(11) Character string processing instructions

Category	Instruction symbol	Symbol	Processing details Execution condition	Number of basic steps	Subset
BIN	BINDA BINDAP	- BINDA S D - BINDAP S D	Converts a 1-word BIN value specified by (S) into a 5-digit decimal ASCII value, and stores it to the word device specified by (D).	3	
↓ Decimal ASCII	DBINDA DBINDAP	- DBINDA S D - DBINDAP S D	Converts a 2-word BIN value specified by (S) into a 10-digit decimal ASCII value, and stores it to word devices following the word device specified by (D).	3	
BIN ↓	BINHA BINHAP	- BINHA S D - BINHAP S D	Converts a 1-word BIN value specified by (S) into a 4-digit hexadecimal ASCII value, and stores it to word devices following the word device specified by (D).	3	
ASCII	DBINHA DBINHAP	- DBINHA S D - DBINHAP S D	Converts a 2-word BIN value specified by (S) into an 8-digit hexadecimal ASCII value, and stores it to word devices following the word device specified by (D).	3	
BCD ↓	BCDDA BCDDAP	-BCDDA SD- BCDDAP SD-	Converts a 1-word BCD value specified by (S) to a 4-digit decimal ASCII value, and stores it to word devices following the word device specified by (D).	3	
Decimal ASCII	DBCDDA DBCDDAP	- DBCDDA S D - DBCDDAP S D	Converts a 2-word BCD value specified by (S) to an 8-digit decimal ASCII value, and stores it to word devices following the word device specified by (D).	3	
Decimal ASCII	DABIN DABINP	- DABIN S D - DABINP S D	Converts a 5-digit decimal ASCII value specified by (S) into a 1-word BIN value, and stores it to a word device specified by (D).	3	
↓ BIN	DDABIN DDABINP	- DDABIN S D - DDABINP S D	Converts a 10-digit decimal ASCII value specified by (S) into a 2-word BIN value, and stores it to a word device specified by (D).	3	
Hexadeci mal	HABIN HABINP	HABIN S D HABINP S D	Converts a 4-digit hexadecimal ASCII value specified by (S) into a 1-word BIN value, and stores it to a word device specified by (D).	3	
ASCII ↓ BIN	DHABIN DHABINP	- DHABIN S D - DHABINP S D	Converts an 8-digit hexadecimal ASCII value specified by (S) into a 2-word BIN value, and stores it to a word device specified by (D).	3	

Category	Instruction symbol	Symbol		Processing details	Execution condition	Number of basic steps	Subset
Decimal ASCII	DABCD DABCDP	DABCD S D DABCDP S D	•	Converts a 4-digit decimal ASCII value specified by (S) into a 1-word BCD value, and stores it to a word device specified by (D).		3	
↓ BCD	DDABCD DDABCDP	- DDABCD S D - - DDABCDP S D -	•	Converts an 8-digit decimal ASCII value specified by (S) into a 2-word BCD value, and stores it to a word device specified by (D).		3	
Device comment read operation	COMRD COMRDP	COMRD S D	•	Stores a comment of the device specified by (S) to a device specified by (D).		3	
Character string length detection	LEN LENP	LEN S D LENP S D	•	• Stores the data length (number of characters) of character string in the device specified by (S) to a device specified by (D).		3	
BIN	STR STRP	- STR S1 S2 D -	•	Converts a 1-word BIN value specified by (S2) into a decimal character string with the total number of digits and the number of decimal fraction digits specified by (S1) and		4	
↓ Decimal character string		- DSTR S1 S2 D -	•	stores them to a device specified by (D). Converts a 2-word BIN value specified by (S2) into a decimal character string with the total number of digits and the number of		4	
	DSTRP	- DSTRP S1 S2 D		decimal fraction digits specified by (S1) and stores them to a device specified by (D).			
V Decimal	VAL VALP	- VAL S D1 D2-	-	 Converts a character string including decimal point specified by (S) into a 1-word BIN value and the number of decimal fraction digits, and stores them to 		4	
character string ↓ BIN	DVAL	- DVAL S D1 D2-	•	 devices specified by (D1) and (D2). Converts a character string including decimal point specified by (S) into a 2-word BIN value and the number of decimal fraction digits, and stores them to devices specified by (D1) and (D2). 		4	
	DVALP	- DVALP S D1 D2-					
Floating decimal point ↓	ESTR	ESTR S1 S2 D	.	Converts the floating decimal point data specified by (S) into a character string,		4	
string	ESTRP	- ESTRP S1 S2 D		and stores it to devices specified by (D).			
↓ Floating	EVAL EVALP	- EVAL S D-	•	Converts the character string specified by (S) to floating decimal point data, and stores it in devices specified by (D).		3	
	ASC	ASC S D n	•	Converts 1-word BIN values of the devices following the device specified by (S) into hexadecimal ASCII, and stores n		4	
	ASCP	ASCP S D n		characters of them to word devices following the device specified by (D).			
↓ Hexadecimal	HEX	HEX SDn HEXP SDn	•	 Converts n hexadecimal ASCII characters of the devices following the device specified by (S) into BIN values, and stores them to the 		4	
BIN	HEXP	HEXP S D n		devices following the device specified by (D).			

Category	Instruction symbol	Symbol		Processing details	Execution condition	Number of basic steps	Subset
	RIGHT RIGHTP	RIGHT S D n RIGHTP S D n	cha	res n characters from the end of a rracter string specified by (S) to the rice specified by (D).			
	LEFT LEFTP	- LEFT S D n - LEFTP S D n -	cha	res n characters from the start tracter string specified by (S) to the rice specified by (D)		4	
	MIDR	MIDR S1 D S2		res the specified number of characters ecified by (S1) from the position			
-	MIDRP	MIDRP S1 D S2	spe by (cified by (S2) to the device specified (D).		4	
	MIDW	MIDW S1 D S2	strir	res the specified number of character ng specified by (S1) from the position		-	
	MIDWP	MIDWP S1 D S2	spe (D).	cified by the device (S2) specified by			
	INSTR	INSTR S1 S2 D n	• Searches for character string (S1) from the nth character of character string (S2),		5		
	INSTRP	INSTRP S1 S2 D n		and stores matched positions to (D).		5	
Floating decimal point	EMOD	EMOD S1 S2 D	of (nverts the floating decimal point data S1) into BCD data with the number of simal fraction digits specified by (S2) ,		4	
↓ BCD	EMODP	EMODP S1 S2 D	decimal fraction digits specified by (S2) , and stores them to the device specified by (D).				
BCD ↓ Floating	EREXP	EREXP S1S2 D	Converts BCD data of (S1) into the floating decimal point data with the number of decimal fraction digits		4		
decimal point	EREXPP	EREXPP S1 S2 D	spe	cified by (S2), and stores them to the vice specified by (D).			

Number of basic steps Subset Instruction Execution Category Symbol Processing details condition symbol SIN S D SIN _ · Sin(S+1,S) → (D+1,D) 3 S D SINP SINP COS COS SD-· Cos(S+1,S) → (D+1,D) 3 COSP S D ſ COSP TAN SD-Trigonometric TAN _ · Tan(S+1,S) → (D+1,D) 3 functions TANP TANP _ S D -(floating ASIN SD-ASIN · Sin⁻¹(S+1,S) → (D+1,D) decimal point 3 data) ASINP S D -ASINP _ S D ACOS ACOS · Cos⁻¹(S+1,S) → (D+1,D) 3 S D -ACOSP ACOSP - ATAN S D -ATAN · Tan⁻¹(S+1,S) → (D+1,D) 3 ATANP S D ATANP _ RAD ____ RAD SD-• (S+1, S) → (D+1, D) Angles 3 Conversion from angle to radian RADP S D -RADP _ Radians _ DEG S D -DEG • (S+1, S) → (D+1, D) 3 conversion Conversion from radian to angle DEGP SD-DEGP ____ SQR SQR SD-· √(S+1,S) → (D+1,D) 3 Square root SQRP 1 SQRP SD-____ _ EXP SD-EXP Exponential · e^(S+1,S) → (D+1,D) 3 operations EXPP S D -EXPP LOG LOG _ SD-· Log e(S+1,S) → (D+1,D) Natural 3 logarithms LOGP S D LOGP Random RND D • Generates a random number (from 0 to RND less than 32767) and stores it to the number RNDP D RNDP device specified by (D). generation 2 Random SRND D • Updates random number series SRND according to the 16-bit BIN data stored in number series D

(12) Special function instructions

the device specified by (S).

SRNDP

SRNDP

update

f

Category	Instruction symbol	Symbol	Processing details	Execution condition	Number of basic steps	Subset
	BSQR	BSQR S D	• $\sqrt{(S)}$ (D) +0 Integral part		- 3	
Square root	BSQRP	BSQRP S D	+1 Decimal part		5	
Square root	BDSQR	BDSQR S D	• $\sqrt{(S+1,S)} \longrightarrow (D) + 0$ Integral part		3	
	BDSQRP	BDSQRP S D	+1 Decimal part		3	
	BSIN	BSIN S D	• Sin(S) → (D) +0 Sign +1 Integral part		3	
	BSINP	BSINP S D	+1 Integral part +2 Decimal part		3	
	BCOS	BCOS S D	• Cos(S) → (D) +0 Sign +1 Integral part		3	
	BCOSP	BCOSP S D	+1 Integral part +2 Decimal part		3	
	BTAN	BTAN S D	• Tan(S) → (D) +0 Sign +1 Integral part		3	
Trigonometric	BTANP	BTANP S D	+2 Decimal part		3	
functions	BASIN	BASIN S D	• Sin ⁻¹ (S) → (D) +0 Sign +1 Integral part		3	
	BASINP	BASINP S D	+1 Integral part +2 Decimal part		3	
	BACOS	BACOS S D	• $\cos^{-1}(S) \longrightarrow (D) + 0$ Sign			
-	BACOSP	BACOSP S D	+1 Integral part +2 Decimal part		3	
	BATAN	BATAN S D	• $Tan^{-1}(S) \longrightarrow (D) + 0$ Sign			
	BATANP	BATANP S D	+1 Integral part +2 Decimal part		3	

Category	Instruction symbol	Symbol	Processing details	Execution condition	Number of basic steps	Subset
	LIMIT	- LIMIT S1 S2 S3 D	 When (S3) < (S1) Stores value of (S1) to (D) When (S1) ≤ (S3) ≤ (S2) Stores value of (S3) to (D) 		5	
Uppor	LIMITP	LIMITP S1 S2 S3 D	 When (S2) < (S3) Stores value of (S2) to (D) 			
Upper and lower limit controls	DLIMIT	- DLIMIT S1 S2 S3 D-	 When ((S3) + 1, (S3)) < ((S1) + 1, S1) Stores value of ((S1) + 1, (S1)) to ((D) + 1, (D)) When ((S1) + 1, (S1)) ≤ ((S3) + 1, (S3)) < (S2 + 1, S2) 		5	
	DLIMITP	- DLIMITP S1 S2 S3 D	 Stores value of ((S3) + 1, (S3)) to ((D) + 1, (D)) When ((S2), (S2) + 1) < ((S3), (S3) + 1) Stores value of ((S2) + 1, (S2)) to ((D) + 1, (D)) 			
		- BAND S1 S2 S3 D - BANDP S1 S2 S3 D	 When (S1) ≤ (S3) ≤ (S2) 0 → (D) When (S3) < (S1) (S3) - (S1) → (D) 		5	
Dead band controls	BANDP DBAND	- DBAND S1 S2 S3 D	 When (S2) < (S3) (S3) - (S2) → (D) When ((S1) + 1, (S1)) ≤ ((S3) + 1, (S3)) ≤ (S2 + 1, S2) 0 → ((D) + 1, (D)) When ((S3) + 1, (S3)) < ((S1) + 1, (S1)) ((S3) + 1, (S3)) - ((S1) + 1, (S1)) → ((D) 		. 5	
	DBANDP	- DBANDP S1 S2 S3 D	• $((S3) + 1, (S3)) - ((S1) + 1, (S1)) - ((D))$ • When $((S2) + 1, (S2)) < ((S3) + 1, (S3))$ $((S3) + 1, (S3)) - ((S2) + 1, (S2)) \rightarrow ((D))$ + 1, (D))			
	ZONE	- ZONE S1 S2 S3 D	 When (S3) = 0 0 → (D) When (S3) > 0 (S3) + (S2) → (D) 		5	
Zone controls	ZONEP DZONE	- ZONEP S1 S2 S3 D - - DZONE S1 S2 S3 D -	• When $(S3) > 0 (S3)$ $(S1) → (D)$ • When $((S3) + 1, (S3)) = 0$ $0 → ((D) + 1, (D))$ • When $((S3) + 1, (S3)) > 0$ $((S3) + 1, (S3)) + ((S2) + 1, (S2))$			
	DZONEP	- DZONEP S1 S2 S3 D-	→((D) + 1, (D)) • When ((S3) + 1, (S3)) < 0 ((S3) + 1, (S3)) + ((S1) + 1, (S1)) →((D) + 1, (D))		5	

(13) Data control instructions

(14) Switching instructions

Category	Instruction symbol	Symbol	Processing details	Execution condition	Number of basic steps	Subset
Block number	RSET	RSET S	Changes the extension file register block number to the number specified by (S).		2	
switching	RSETP	RSETP S			_	
	QDRSET	QDRSET File name	 Sets file names to be used as file 		*	
File set	QDRSETP	QDRSETP File name	registers.		2+n	
1 116 261	QCDSET	QCDSET File name	 Sets file names to be used as comment 		*	
	QCDSETP	QCDSETP File name	files.		2+n	

*: n ([number of file name characters]/2) indicates a step (decimal fraction is rounded up)

Category	Instruction symbol	Symbol	Processing details	Execution condition	Number of basic steps	Subset
	DATERD	- DATERD D	• (Clock element) →(D)+0 Year +1 Month +2 Day +3 Hour		2	
Read/write	DATERDP	- DATERDP D-	+4 Minute +5 Second +6 Day of week		2	
-	DATEWR	- DATEWR S	• (D)+0 Year +1 Month +2 Day +3 Hour		2	
	DATEWRP	DATEWRP S	+4 Minute +5 Second +6 Day of week		2	
	DATE+	DATE+ S1 S2 D	(S1) (S2) (D) Hour Hour Hour		4	
Clock data addition/su	DATE+P	DATE+P S1 S2 D	$\begin{array}{c c} \mbox{Minute} & \mbox{+} & \mbox{Minute} & \mbox{Minute} \\ \mbox{Second} & \mbox{Second} & \mbox{Second} \end{array} \xrightarrow{\mbox{Minute} \\ \mbox{Second} & \mbox{Second} \end{array}$		4	
btraction	DATE-	DATE - S1 S2 D	(S1) (S2) (D) Hour Hour		4	
	DATE-P	DATE-P S1 S2 D	$\begin{array}{c c} \mbox{Minute} & - & \mbox{Minute} \\ \mbox{Second} & & \mbox{Second} \end{array} \rightarrow & \mbox{Minute} \\ \hline \mbox{Second} & & \mbox{Second} \end{array}$		-	
	SECOND	- SECOND S D	(S) (D) Hour Second (lower)			
Clock data	SECONDP	SECONDP S D	Minute → Second (upper) Second		3	
translation	HOUR	HOUR S D	(S) (D) Second (lower) Hour		5	
	HOURP	HOURP S D	Second (upper) → Minute Second Second			

(15) Clock instructions

(16) Peripheral device instructions

Category	Instruction symbol	Symbol	Processing details	Execution condition	Number of basic steps	Subset
Input/output to peripheral	MSG	MSG S	 Stores the message specified by (S) to the QnACPU. This message is displayed on the peripheral device. 		2	
device	PKEY	PKEY D	 Stores the data input from the peripheral device to the device specified by (D). 		2	

(17) Program instructions

Category	Instruction symbol	Symbol	Processing details	Execution condition	Number of basic steps	Subset
	PSTOP	- PSTOP Program name	Sets the specified program to the standby		*	
	PSTOPP	- PSTOPP Program name	 status. Turns off the coil of the OUT instruction of the specified program, and sets the 		2+n	
	POFF	- POFF Program name-			*	
	POFFP	- POFFP Program name-	program to the standby status.		2+n	
	PSCAN	- PSCAN Program name-	Registers the specified program as a scan		*	
	PSCANP	- PSCANP Program name	execution type.		2+n	
	PLOW	- PLOW Program name	Registers the specified program as a		*	
	PLOWP	- PLOWP Program name	low-speed execution type.		2+n	

*: n ([number of file name characters]/2) indicates a step (decimal fraction is rounded up).

Category	Instruction symbol	Symbol	Processing details	Execution condition	Number of basic steps	Subset
WDT reset	WDT	- WDT -	Resets the watchdog timer during the		1	
WD1 leset	WDTP	WDTP -	execution of the sequence program.		I	
Timing clock	DUTY	DUTY n1 n2 D	(D) ↑ 1 scan 12 scan SM420 to SM424, SM430 to SM434		4	
	ZRRDB	ZRRDB n D	0 Lower 8 bits 1 Upper 8 bits 2 Lower 8 bits ZR1		3	
Direct	ZRRDBP	ZRRDBP n D	$\begin{array}{c c} 3 & Upper 8 bits \\ n & 8 bits \\ \hline \end{array} \rightarrow (D) \\ \hline \end{array}$		Ū	
read/write operations	ZRWRB	ZRWRB n S	(S) 0 Lower 8 bits 1 Upper 8 bits 2 Lower 8 bits 2 Lower 8 bits 2 ZR1		3	
in 1-byte units	ZRWRBP	ZRWRBP n S	$ \begin{array}{c} 3 \\ n \\ \hline 8 \\ \hline $		Ŭ	
units	ADRSET	ADRSET S D	(S) →(D) Indirect address		3	
	ADRSET	ADRSETP S D	of specified device Device name		5	
Numerical key input from keyboard	KEY	KEY S n D1 D2	 Imports ASCII data to 8 points of input unit specified by (S), converts it into hexadecimal value, and stores it to devices following the device specified by (D1). 		5	
Batch save	ZPUSH	ZPUSH D	 Saves the contents of index registers Z0 to Z15 to devices following the 			
of index register	ZPUSHP	ZPUSHP D	device specified by (D).			
Batch recovery of	ZPOP	ZPOP D	 Reads the data stored in the devices following the device specified by (D) 		2	
index register	ZPOPP	ZPOPP D	to the index registers Z0 to Z15.			
Batch write to	EROMWR	EROMWR S D1 n D2	• Writes data to the file register of		5	
E ² PROM	EROMWRP EROMWRP S D1		E ² PROM in batch.		5	

(18) Other instructions

						_
Category	Instruction symbol	Symbol	Processing details	Execution condition	Number of basic steps	Subset
Reading module	UNIRD	UNIRD n1 D n2	 Reads the module information stored in the device starting from the I/O number specified by (n) by the points specified 		4	
information	UNIRDP	UNIRDP n1 D n2	by (n2), and stores it to devices following the device specified by (D).			
Trace set	TRACE	TRACE	 Stores the trace data set with the peripheral device of the number of set times set when SM800, SM801, and SM802 are turned on to the trace file of the IC memory card. 		1	
Trace reset	TRACER	- TRACER -	 Resets the data set with the TRACE instruction. 		1	
Writing data to the specified file	SP.FWRITE		 Writes data to the specified file. 		11	
Reading data from specified file	SP.FREAD	- SP.FREAD U0 80 D0 81 82 D1	 Reads data from the specified file. 		11	
Loading program from memory	PLOADP	- PLOADP S D-	 Transfers the program stored in a memory card or standard memory (other than drive 0) to drive 0 and sets the program to the standby status. 		3	
Unloading program from program memory	PUNLOADP	- PUNLOADP S D	 Deletes the standby program stored in the standard memory (drive 0). 		3	
Load + Unload	PSWAPP	- PSWAPP S1 S2 D	 Deletes the standby program stored in the standard memory (drive 0) specified by (S1). Then, transfers the program stored in a memory card or standard memory (other than drive 0) specified by (S2) to drive 0 and sets it to the standby status. 		4	
High-speed block transfer	RBMOV	RBMOV S D n	 Transfers n points of 16-bit data from the device specified by (S) to n points of the devices starting from the device 		4	
of file register	RBMOVP	RBMOVP S D n	of the devices starting from the device specified by (D).			
Writing to host CPU	S.TO	-S.TO n1 n2 n3 n4 D	Writes device data of the host station		5	
shared memory	S.TOP	-S.TOP n1 n2 n3 n4 D	to the host CPU shared memory area.		5	
Reading from other CPU	FROM	-FROM n1 n2 D n3	 Reads the device data from the other CPU shared memories, and stores 		5	
shared memory	FROMP	-FROMP n1 n2 D n3-	the data in the host station.		5	
CPU shared memory auto refresh	СОМ	СОМ	 Executes the auto refreshing of the intelligent function module, general data, and multiple CPU shared memory. 		1	

Appendix 1.2 QCPU Instructions

Appendix 2 How to Create Ladder Programs with GX Works2

Ladder program creation methods include the following five methods.

- 1. Inputting list expressions (mnemonic language) with the keyboard on the ladder program creation screen
- 2. Using the tool buttons on the toolbar
- 3. Using the function keys
- 4. Using the items of the menu bar
- 5. Directly inputting devices (only for I/O devices)

The following table shows the operations of each method.

Ex.	List expression	Tool button	Function	Menu bar
×1 ⊣¦⊢	LD X1 → Enter	Click $\ddagger f = 0$ "X1" in the device instruction input field \rightarrow Click the OK button.	F5 → "X1" in the device instruction input field → Enter	$\begin{array}{l} [Edit] \rightarrow [Ladder Symbol] \\ \rightarrow [Open Contact] \rightarrow \\ "X1" in the device \\ instruction input field \rightarrow \\ \hline Enter \end{array}$
×1 –∤∕ –	LDI X1 → Enter	Click $\implies \rightarrow$ "X1" in the device instruction input field \rightarrow Click the OK button.	Shift + F5 \rightarrow "X1" in the device instruction input field \rightarrow Enter	$\begin{array}{l} [Edit] \rightarrow [Ladder Symbol] \\ \rightarrow [Close Contact] \rightarrow \\ "X1" in the device \\ instruction input field \rightarrow \\ \hline \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \$
×1 L -	OR X1 → Enter	Click $\begin{array}{c} \begin{array}{c} \begin{array}{c} \\ \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \begin{array}{c} \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \begin{array}{c} \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \begin{array}{c} \end{array} \\ \begin{array}{c} \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \begin{array}{c} \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} $	$\boxed{F6}$ → "X1" in the device instruction input field → \boxed{Enter}	$\begin{array}{l} [Edit] \rightarrow [Ladder Symbol] \\ \rightarrow [Open Branch] \rightarrow "X1" \\ in the device instruction \\ input field \rightarrow \hline Enter \end{array}$
	ORI X1 → Enter	Click $\implies \rightarrow$ "X1" in the device instruction input field \rightarrow Click the OK button.	Shift + F6 \rightarrow "X1" in the device instruction input field \rightarrow Enter	$\begin{array}{l} [Edit] \rightarrow [Ladder Symbol] \\ \rightarrow [Close Branch] \rightarrow "X1" \\ in the device instruction \\ input field \rightarrow \hline \hline Enter \end{array}$
Ŏ	OUT Y1 → Enter	Click $\overleftrightarrow{P} \rightarrow "Y1"$ in the device instruction input field \rightarrow Click the OK button.	$\boxed{F7} \rightarrow "Y1"$ in the device instruction input field $\rightarrow \boxed{Enter}$	$\begin{array}{l} [Edit] \rightarrow [Ladder Symbol] \\ \rightarrow [Coil] \rightarrow "Y1" in the \\ device instruction input \\ field \rightarrow \boxed{Enter} \end{array}$
	MOV K1 D0 → Enter	Click $\overrightarrow{\mathbb{H}} \rightarrow "MOV K1$ D0" in the device instruction input field \rightarrow Click the OK button.	F8 → "MOV K1 D0" in the device instruction input field → Enter	$\begin{array}{l} [{\sf Edit}] \rightarrow [{\sf Ladder Symbol}] \\ \rightarrow [{\sf Application} \\ {\sf Instruction}] \rightarrow "{\sf Y1}" \mbox{ in the} \\ {\sf device instruction input} \\ {\sf field} \rightarrow \end{tabular}$
×1 ⊣∰–	LDP X1 → Enter	Click $\implies \rightarrow$ "X1" in the device instruction input field \rightarrow Click the OK button.	Shift + F7 \rightarrow "X1" in the device instruction input field \rightarrow Enter	$\begin{array}{l} [Edit] \rightarrow [Ladder Symbol] \\ \rightarrow [Pulse Contact \\ Symbol] \rightarrow [Rising Pulse] \\ \rightarrow "X1" in the device \\ instruction input field \rightarrow \\ \hline Enter \end{array}$
×1 	LDPI X1 → Enter	Click $\implies \rightarrow$ "X1" in the device instruction input field \rightarrow Click the OK button.	Shift + Alt + F5 → "X1" in the device instruction input field → Enter	$\begin{array}{l} [Edit] \rightarrow [Ladder Symbol] \\ \rightarrow [Pulse Contact \\ Symbol] \rightarrow [Rising Pulse \\ Close] \rightarrow "X1" in the \\ device instruction input \\ field \rightarrow \boxed{Enter} \end{array}$

Ex.	List expression	Tool button	Function	Menu bar
×1 –↓	LDF X1 → Enter	Click $\implies \rightarrow$ "X1" in the device instruction input field \rightarrow Click the OK button.	Shift + F8 \rightarrow "X1" in the device instruction input field \rightarrow Enter	$\begin{array}{l} [{\sf Edit}] \rightarrow [{\sf Ladder Symbol}] \\ \rightarrow [{\sf Pulse Contact} \\ {\sf Symbol}] \rightarrow [{\sf Falling} \\ {\sf Pulse}] \rightarrow "{\sf X1"} \mbox{ in the} \\ {\sf device instruction input} \\ {\sf field} \rightarrow \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ $
×1	LDFI X1 → Enter	Click $\implies \rightarrow$ "X1" in the device instruction input field \rightarrow Click the OK button.	Shift + Alt + $F6$ \rightarrow "X1" in the device instruction input field \rightarrow Enter	$\begin{array}{l} [Edit] \rightarrow [Ladder Symbol] \\ \rightarrow [Pulse Contact \\ Symbol] \rightarrow [Falling Pulse \\ Close] \rightarrow "X1" in the \\ device instruction input \\ field \rightarrow \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ $
	ORP X1 → Enter	Click $\textcircled{W} \rightarrow "X1"$ in the device instruction input field \rightarrow Click the OK button.	Alt + $F7 \rightarrow "X1"$ in the device instruction input field \rightarrow Enter	$\begin{array}{l} [{\sf Edit}] \rightarrow [{\sf Ladder Symbol}] \\ \rightarrow [{\sf Pulse Contact} \\ {\sf Symbol}] \rightarrow [{\sf Rising Pulse} \\ {\sf Branch}] \rightarrow "{\sf X1}" \mbox{ in the} \\ {\sf device instruction input} \\ {\sf field} \rightarrow \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ $
	ORPI X1 → Enter	Click $\textcircled{W} \rightarrow "X1"$ in the device instruction input field \rightarrow Click the OK button.	Shift + Alt + F7 \rightarrow "X1" in the device instruction input field \rightarrow Enter	$\begin{array}{l} [Edit] \rightarrow [Ladder Symbol] \\ \rightarrow [Pulse Contact \\ Symbol] \rightarrow [Rising Pulse \\ Close Branch] \rightarrow "X1" in \\ the device instruction \\ input field \rightarrow \boxed{Enter} \end{array}$
×1 └╢┤	ORF X1 → Enter	Click $\blacksquare \rightarrow$ "X1" in the device instruction input field \rightarrow Click the OK button.	Alt + F8 \rightarrow "X1" in the device instruction input field \rightarrow Enter	$\begin{array}{l} [Edit] \to [Ladder Symbol] \\ \to [Pulse Contact \\ Symbol] \to [Falling Pulse \\ Branch] \to "X1" \text{ in the} \\ device instruction input \\ field \to \boxed{Enter} \end{array}$
	ORFI X1 → Enter	Click $\implies \rightarrow$ "X1" in the device instruction input field \rightarrow Click the OK button.	Shift + Alt + $F8$ \rightarrow "X1" in the device instruction input field \rightarrow Enter	$\begin{array}{l} [Edit] \rightarrow [Ladder Symbol] \\ \rightarrow [Pulse Contact \\ Symbol] \rightarrow [Falling Pulse \\ Close Branch] \rightarrow "X1" in \\ the device instruction \\ input field \rightarrow \end{tabular}$
1	EGP V0 → Enter	Click $\stackrel{\uparrow}{afs} \rightarrow$ "V0" in the device instruction input field \rightarrow Click the OK button.	Alt + $F5 \rightarrow "V0"$ in the device instruction input field \rightarrow Enter	$\begin{array}{l} [Edit] \rightarrow [Ladder Symbol] \\ \rightarrow [Operation Result \\ Rising Pulse] \rightarrow "V0" in \\ the device instruction \\ input field \rightarrow \boxed{Enter} \end{array}$
t	EGF V0 → Enter	Click $rightarrow = 0$ "V0" in the device instruction input field \rightarrow Click the OK button.	Ctrl+Alt+F5 \rightarrow "V0" in the deviceinstruction input field \rightarrow Enter	$\begin{array}{l} [Edit] \rightarrow [Ladder Symbol] \\ \rightarrow [Operation Result \\ Falling Pulse] \rightarrow "V0" in \\ the device instruction \\ input field \rightarrow \boxed{Enter} \end{array}$
	INV → Enter	Click $\overrightarrow{\text{Gree}} \rightarrow \text{Click the}$ OK button.	$\begin{array}{ c c }\hline Ctrl + & Alt \\ \hline \\ \hline \\ \hline \\ \hline \\ Enter \end{array} + & F10 \\ \hline \\ $	$\begin{array}{l} [Edit] \to [Ladder Symbol] \\ \to [Invert Operation \\ Results] \to \boxed{Enter} \end{array}$

		Tool button	Function	Manuhar
Ex.	List expression	Tool button	Function	Menu bar
Horizontal line	-	Click $\boxed{FS} \rightarrow$ Input the number of lines to be written in the device instruction input field. \rightarrow Click the OK button.	F9 → Input the number of lines to be written in the device instruction input field. → Enter	$ \begin{array}{l} [Edit] \to [Ladder Symbol] \\ \to [Horizontal Line] \to \\ \\ Input the number of lines \\ to be written in the device \\ \\ instruction input field. \to \\ \hline \end{array} \end{array} $
Vertical line	-	Click \downarrow_{10} \rightarrow Input the number of lines to be written in the device instruction input field. \rightarrow Click the OK button.	F10 → Input the number of lines to be written in the device instruction input field. → Enter	$\begin{array}{l} [\text{Edit}] \rightarrow [\text{Ladder Symbol}] \\ \rightarrow [\text{Vertical Line}] \rightarrow \text{Input} \\ \text{the number of lines to be} \\ \text{written in the device} \\ \text{instruction input field.} \rightarrow \\ \hline \text{Enter} \end{array}$
Rule line	-	Click $\downarrow = 0$ → Move the cursor to the position where a line is to be written and drag the cursor.	Alt + $F10 \rightarrow Move$ the cursor to the position where a line is to be written and drag the cursor.	$[\text{Edit}] \rightarrow [\text{Edit Line}] \rightarrow$ Move the cursor to the position where a line is written and drag the cursor.
Delete vertical line	-	Click $\overrightarrow{abs} \rightarrow$ Input the number of lines to be deleted in the device instruction input field. \rightarrow Click the OK button.	Ctrl + F9 → Inputthe number of lines to bedeleted in the deviceinstruction input field. →Enter	$ \begin{array}{l} [Edit] \to [Ladder Symbol] \\ \to [Delete Horizontal \\ Line] \to Input \ the number \\ of lines to be deleted in \\ the device instruction \\ input field. \to \boxed{Enter} \end{array} $
Delete rule line	-	Click $\overleftarrow{bm} \rightarrow$ Input the number of lines to be deleted in the device instruction input field. \rightarrow Click the OK button.	Ctrl + F10 → Inputthe number of lines to bedeleted in the deviceinstruction input field. →Enter	$\begin{array}{l} [{\rm Edit}] \rightarrow [{\rm Ladder \ Symbol}] \\ \rightarrow [{\rm Delete \ Vertical \ Line}] \\ \rightarrow {\rm Input \ the \ number \ of} \\ {\rm lines \ to \ be \ deleted \ in \ the} \\ {\rm device \ instruction \ input \ field.} \rightarrow {\rm \ Enter} \end{array}$
Delete horizontal line	-	Click $\boxed{36} \rightarrow$ Move the cursor to the position where lines are to be deleted and drag the cursor.	Alt + F9 \rightarrow Move the cursor to the position where lines are to be deleted and drag the cursor.	$[Edit] \rightarrow [Delete Line] \rightarrow$ Move the cursor to the position where lines are to be deleted and drag the cursor.
POINT				
Input and deletion		s can also be input and	deleted with Ctrl +	an arrow key.
Horizontal lines		uously from the cursor by the same method.)	position with Ctrl +	Shift + an arrow
				(Y300)
0		Ctrl + Shift	* →	-[end]
	×10 ×20 			-(7300)
0				-[END]

The following shows how to input the instructions for the low-speed timer, high-speed timer, retentive timer, and edge relay.

(1) Low-speed timer	(4) Edge relay
Enter Symbol	Enter Symbol
OUT TO K100 OK Exit Help	EPG VO OK Exit Help
Enter Symbol	Enter Symbol
CK Exit Help	VO OK Exit Help
(2) High-speed timer	
Enter Symbol	
OUTH TO K100 OK Exit Help	
Enter Symbol	
-() H TO K100 OK Exit Help	
(3) High-speed retentive timer	
Enter Symbol	
OUTH STO K100 OK Exit Help	
Enter Symbol	
Exit Help	

To input the retentive timer, set the device points on the device setting screen of the PLC parameter.

In the A/D and D/A converter modules which handle analog signals, adjust the offset/gain to secure the signal accuracy as necessary. (For the offset/gain, refer to appendix. 3.2.)

In the A/D or D/A converter modules of Q-series, the offset/gain settings can be executed in the GX Works2.

The A/D converter module (Q64AD) is used as an example in the following. The setting procedure is the same for the D/A converter module (Q62DAN). For details, refer to the following manuals.

- Analog-Digital Converter Module User's Manual SH-080055
- Digital-Analog Converter Module User's Manual SH-080054

Appendix 3.1 Offset/gain setting with GX Works2

- Relevant Manipulation
 Relevant Manipulation
 Relevant Manipulation
 Relevant Manipulation

 Relevant Manipulation
 Relevant Manipulation
 Relevant Manipulation
 Relevant Manipulation
- (1) Switching the mode to the offset/gain setting mode
 - 1) Click the "I/O Assignment" tab of the PLC parameter.
 - 2) Select "Q64AD" and click the Switch Setting button.

- Image: Section of the sectio
- 3) Set the bit 4 of the switch 4 to a value except "0".

In this example, set the switch 4 to "1000".

Write the PLC parameter set above to the CPU and reset the CPU.

Write the parameters with the following procedure to the second CPU installed to the demonstration machine.
(When only one CPU is used, the following procedure is not needed.)
1) Change the connection destination to the second CPU. (Refer to section 10.4.5 (1).)
2) Write the parameter setting to the QCPU.

 After completion of the writing, set the connection destination to the first CPU again. (Refer to section 10.4.5 (1).)

Resetting switches the mode to the offset/gain setting mode, and the RUN LED on Q64AD flashes.

- te MULTIZ 11 S 1) Click [Tool] \rightarrow [Intelligent Function Module Tool] \rightarrow 法法约 月月 脸边。 [Analog Module] \rightarrow [Offset/Gain Setting]. Options Eary Cu (191 U9\ G1 -1/ D10 K2 Confirm Hemory Size ... Herge Dy Set TEL Data/Co END 1) Click! Dhernet Adapter Module Configura Byth-in L/O Module Tool Q61LD Default Setting Counter Module 75AD75 Boots ing Module ; in Module Module Selection (Offset/Gain Setting) 2) The Module Selection (Offset/Gain Setting) screen Module Selection is displayed. Select "Q64AD (0080)" and click the OK button. Q64AD 2) Select! 2) Click! OK Cancel 3) The message on the left is displayed. Click the 1 Caution - A/D conversion will be cancelled when switching over to offset/gain setting in - In case of error occurrence at the target module, the error will be deared with Yes button. No 3) Click! 4) The Offset/Gain Setting screen is displayed. Set offset/gain settings Error Code Target Module 0080:Q64AD Adjust the offset values and the gain values on the screen. Offset/Gain Setting Offset Status Gain Status Channel Selection Offset Setting Г сн<u>1</u> Г сн<u>г</u> Gain Setting Г CHE CH8 Please select a target channel for the offset/gain setting and press "Offset Setting" or "Gain Setting", Pressing "Close" registers to the module. Close
- (2) Switching the screen to the Offset/Gain Setting screen

(3) Specifying channels

Select the check box on the "Channel Selection" column to specify channels for which the offset or gain is set.

- (4) Applying current or voltage Apply current or voltage to the module.
- (5) Executing the offset/gain setting

For each of the channels specified in (3), click the Offset Setting button when executing the offset setting, or click the Gain Setting button when executing the gain setting.

(6) Switching the mode to the normal mode After completion of the setting, click the OK button. The following message is displayed.

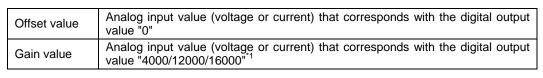
ELSOF	T Series GX Works2
(i)	The mode will be switch over to the offset/gain mode to normal mode.
V	Caution - The mode will not be switched over to normal mode when the offset/gain mode is selected in the switch setting.
	OK

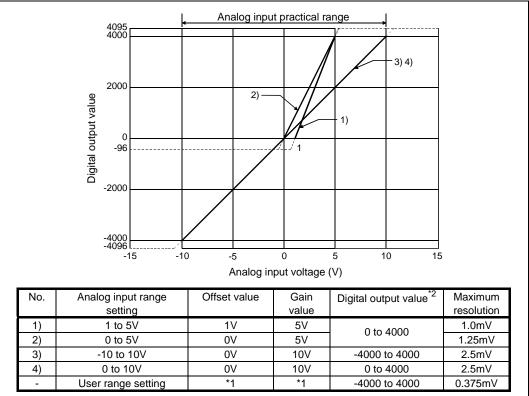
POINT

When an error code is displayed in the setting, the error detail and the solution							
can be confirmed by clicking the button on the right of the error code							
display area. Error codes can be cleared by clicking the Error Clear							
button.							

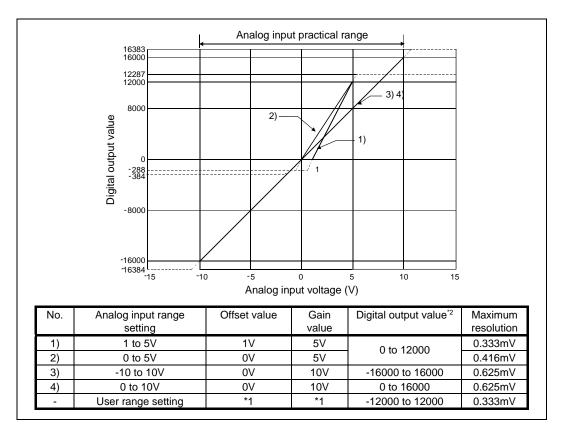
Appendix 3.2 Offset value and gain value

(1) I/O characteristic of the A/D conversion	on
--	----





Voltage input characteristic in standard resolution mode



Voltage input characteristic in high resolution mode

POINT

- (1) Do not input an analog voltage out of the range between -15 to 15V. The elements may be damaged.
- (2) Set the offset/gain values for the user range setting *1 within a range in which the following conditions are met.

{ (Gain value) - (Offset value) } > A

<Value A>

Standard resolution mode	High resolution mode		
1.5V	4.0V		

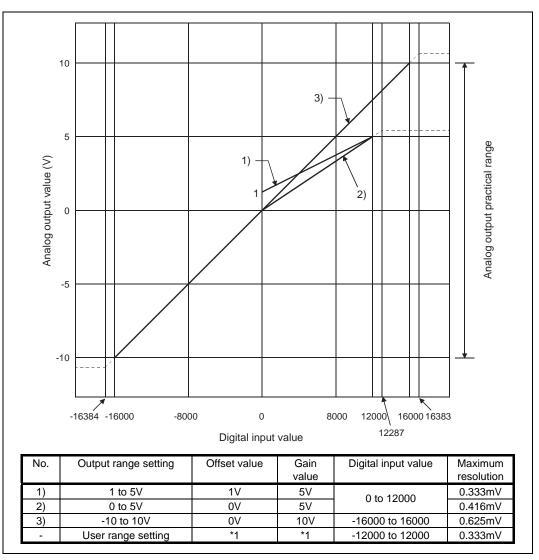
(3) When an analog value that exceeds the range for the digital output value *2 is input, the digital output value will be fixed at the maximum or minimum value.

Analog input range	Standard res	solution mode	High resolution mode		
setting	Minimum	Maximum	Minimum	Maximum	
1 to 5V	-96		-288	12287	
0 to 5V	-90		-200		
-10 to 10V	-4096	4095	-16384		
0 to 10V	-96		-384	10303	
User range setting	-4096		-12288	12287	

Offset value	Э	valu	ie "0"	-		at corresponds with	
Gain value		Ana valu	llog output va le "4000/1200	alue (voltage or 00/16000"	current) th	at corresponds with	the digital inpu
			1				
						,	
1	0						
						3)	
	5						g
S				1) —	~ /		Analog output practical range
) en				,			tica
Analog output value (V)				1	7/	2)	orac
tput	0 -				\vee		ort b
out							outp
alog							bo
Ana							vnal
-	-5						
	J						
-1	0	/	1				_
	-4096	÷ -40	.000 -2	2000 0		2000 4000 4	095
				Digital inpu	it value		
No.	Out	put ra	nge setting	Offset value	Gain	Digital input value	Maximum
			-14		value		resolution
1)			o 5V	1V	5V	0 to 4000	1.0mV
2)			o 5V to 10V	0V	5V	4000 to 4000	1.25mV 2.5mV
3)	He		ige setting	0V *1	10V *1	-4000 to 4000 -4000 to 4000	2.5mV 0.75mV
	03	orial	igo sounig				0.70111

(2) I/O characteristic of the D/A conversion

Voltage output characteristic in standard resolution mode



Voltage output characteristic in high resolution mode

POINT

Set the offset/gain values for the user range setting *1 within a range in which the following conditions are met.

- (a) The setting range is -10 to 10V.
- (b) { (Gain value) (Offset value) } > A

<Value A>

Standard resolution mode	High resolution mode
3.0V	4.0V

Appendix 4 Specifications of the A/D and D/A Converter Modules

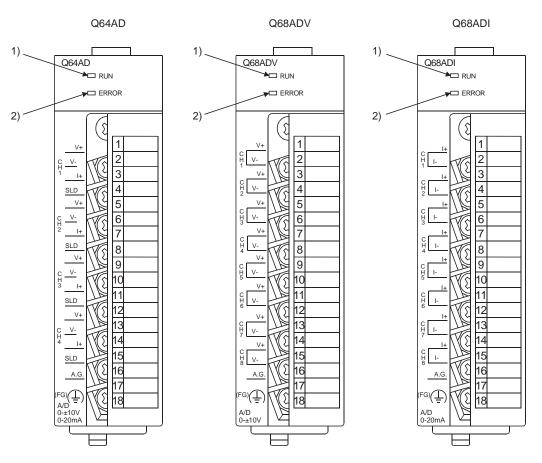
Appendix 4.1 A/D converter module

For details, refer to the Analog-Digital Converter Module User's Manual SH-080055.

Mo	odel name		Q64AD		Q	68ADV			Q68ADI	
Analog input po	oints	4 point	s (4 channels	6)	8 points	s (8 channels)		8 points (8	channels)
	Voltage	-	-10 to 10VDC	C (Input re	sistance value	1MΩ)			-	
Analog input Current		0 to 20mADC (Input resistance value 250Ω)		stance		-		0 to 20mADC (Input resistance value 250Ω)		•
Digital output		16-bit signed binary (standard resolution mode: -4096 to 4095, high resolution mode: -12288 to 12287, -16384 to 16383)							384 to 16383)	
				St	andard resolut	ion mode		Hig	h resolution	mode
		Analog	input range	Digital	output value	Maximum resolution	1 1 100	ital ou	utput value	Maximum resolution
			0 to 10V			2.5mV		0 to	16000	0.625mV
			0 to 5V	0	to 4000	1.25mV		0 to	12000	0.416mV
I/O characterist	ics.	Voltage	1 to 5V	-		1.0mV				0.333mV
Maximum resol	,		-10 to 10V		00 to 4000	2.5mV	2.5mV -16000 to 16000 0.625mV			
			User range setting	-400	10 10 4000	0.375mV	-1	2000	to 12000	
			0 to 20mA	0	to 4000	5µA		0 to	12000	1.66µA
		Current	4 to 20mA			4µA				1.33µA
			User range setting	-400	00 to 4000	1.37µA	-1	2000	to 12000	1.33µA
				Chart			1	Link		a a da
			_		dard resolution	mode	Ambio		resolution n	node
		٨٣			emperature 0 55°C		Amble	nbient temperature 0 to 55°C		
			alog range t	With	Without	Ambient temperature	With		Without	Ambient temperature
				temperature	temperature	25±5℃	tempera			25±5°C
				drift compensation	drift compensation		drift compens		drift compensation	
			0 to 10V	componication			±0.3		±0.4%	±0.1%
			-10 to		±0.4%		git*)	(±64 digit*)	(±16 digit*)	
Accuracy) (alta sa	10V	±0.3%						
(Accuracy for m digital output va		Voltage	0 to 5V 1 to 5V							
uigitai output va	alue)		User							
			range					±0.4% (±48 digit*)		
				(±12 digit*)	(±16 digit*)				0.494	
			0 to 20mA						±0.1% (±12 digit*)	
			2011A 4 to				(±00 u	ອາ /	(± 10 digit)	(212 Gigit)
		Current	20mA							
			User							
			range setting							
			Jotting		1	I	*	Diait	indicates a c	digital value.
						/channel		0		0
Conversion spe	ed	(When the te	emperature d		nsation functio ardless of the					g 160µs will be
Absolute maxim	num input				Voltage: ±15V					
Insulation methe	od	Betwee	n the I/O terr		programmable Between chanr			ply: F	hotocoupler	insulation
Occupied points						points				
Connection term						erminal block	(
Applicable wire	size				0.3 to	0.75mm ²				
Applicable solde terminal	erless		R1.	25-3 (A sc	Iderless termir		e canno	ot be ι	used.)	
Internal current consumption (5			0.63A			0.64A			0.6	4A
Weight			0.18kg		(0.19kg			0.19	Эkg

(1) Performance specifications

(2) Names of parts



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 have been set to the switch 5 on the intelligent function module.

(3) List of I/O signals

Signal direction	n: CPU \leftarrow A/D converter module	Signal direction: Cl	$PU \rightarrow A/D$ converter module
Device No. (input)	Signal name	Device No. (output)	Signal name
X0	Module READY	Y0	
X1	Temperature drift compensation flag	Y1	
X2		Y2	
Х3		Y3	
X4	Use prohibited ^{*1}	Y4	Use prohibited ^{*1}
X5	Ose prohibited	Y5	
X6		Y6	
X7		Y7	
X8	High resolution mode status flag	Y8	
Х9	Operating condition setting completed flag	Y9	Operating condition setting request
XA	Offset/gain setting mode flag	YA	User range writing request
ХВ	Channel change completed flag	YB	Channel change request
XC	Use prohibited ^{*1}	YC	Use prohibited ^{*1}
XD	Maximum value/minimum value reset	YD	Maximum value/minimum
	completed flag	עז	value reset request
XE	A/D conversion completed flag	YE	Use prohibited ^{*1}
XF	Error flag	YF	Error clear request

List of I/O signals of the A/D converter module

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.

(4) Buffer memory

The following explanation is mentioned based on the Q68ADV and Q68ADI with 8-channel analog input (CH1 to CH8).

Add	ress			Add	ress		
Hexa deci mal	Deci mal	Description	R/W ^{*2}	Hexa deci mal	Deci mal	Description	
0н	0	A/D conversion enable/disable setting	R/W	18H	24		
1н	1	CH1 Average time/average number of times	R/W	19н	25		
2н	2	CH2 Average time/average number of times	R/W	1Ан	26	System area	
Зн	3	CH3 Average time/average number of times	R/W	1Вн	27		
4н	4	CH4 Average time/average number of times	R/W	1Сн	28		
5н	5	CH5 Average time/average number of times ^{*1}	R/W	1Dн	29		
6н	6	CH6 Average time/average number of times ^{*1}	R/W	1Ен	30	CH1 Maximum value	R/W
7н	7	CH7 Average time/average number of times ^{*1}	R/W	1Fн	31	CH1 Minimum value	R/W
8н	8	CH8 Average time/average number of times ^{*1}	R/W	20н	32	CH2 Maximum value	R/W
9н	9	Averaging process setting	R/W	21н	33	CH2 Minimum value	R/W
Ан	10	A/D conversion completed flag	R	22н	34	CH3 Maximum value	R/W
Вн	11	CH1 Digital output value	R	23н	35	CH3 Minimum value	R/W
Сн	12	CH2 Digital output value	R	24н	36	CH4 Maximum value	R/W
Dн	13	CH3 Digital output value	R	25н	37	CH4 Minimum value	R/W
Ен	14	CH4 Digital output value	R	26н	38	CH5 Maximum value ^{*1}	R/W
Fн	15	CH5 Digital output value ^{*1}	R	27н	39	CH5 Minimum value ^{*1}	R/W
10н	16	CH6 Digital output value ^{*1}	R	28н	40	CH6 Maximum value ^{*1}	R/W
11н	17	CH7 Digital output value ^{*1}	R	29н	41	CH6 Minimum value ^{*1}	R/W
12н	18	CH8 Digital output value ^{*1}	R	2Ан	42	CH7 Maximum value ^{*1}	R/W
13н	19	Error code	R/W	2Вн	43	CH7 Minimum value ^{*1}	R/W
14н	20	Setting range (CH1 to CH4)	R	2Сн	44	CH8 Maximum value ^{*1}	R/W
15н	21	Setting range (CH5 to CH8) ^{*1}	R	2Dн	45	CH8 Minimum value ^{*1}	R/W
16н	22	Offset/gain setting mode Offset specification	R/W				
17н	23	Offset/gain setting mode Gain specification	R/W				

Buffer memory assignment of the A/D converter module

*1: For Q64AD, the buffer memory areas for CH5 to CH8 are system areas.

*2: Indicates whether reading from and writing to a sequence program are enabled.

R : Read enabled

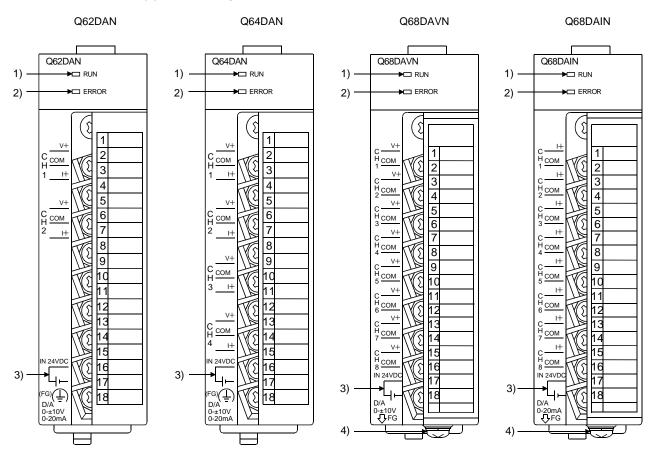
W : Write enabled

Appendix 4.2 D/A converter module

For the details, refer to the Digital-Analog Converter Module User's Manual SH-080054.

Model name		Q62	DAN	Q64DAN	Q68D	Q68DAVN		Q68DAIN	
Analog output poir	nts	2 points (2	2 points (2 channels) 4 points (4 channels) 8 points (8 channels) 16-bit signed binary						
Digital input		(at a stand set a		16-bit sig	ned binary	40000 1- 44	0007 4	00041-40000	
3 1 1	Voltage	(standard resolution mode: -4096 to 4095, high resolution mode: -12288 to 12287, -16384 to 16383) -10 to 10VDC (External load resistance value: 1kΩ to 1MΩ)							
Analog output	Current		```	load resistance value:	- -		- 0 to 20mADC (External load resistance value: 0Ω to 600Ω)		
				Other developments	Cara an a da	L L'arte a			
		Analog	output range	Standard resolu		High re	esolutior	n mode Maximum	
		Analog		Digital input value	Maximum resolution	Digital input	t value	resolution	
			0 to 5V	0 to 4000	1.25mV	0 to 120	000	0.416mV	
1/O oborostaristica	Maximum	Voltogo	1 to 5V	1	1.0mV	40000 1- 4	10000	0.333mV	
I/O characteristics resolution	, Maximum	Voltage	-10 to 10 User rang		2.5mV	-16000 to 1	16000	0.625mV	
10301011011			setting	-4000 10 4000	0.75mV	-12000 to 1	12000	0.333mV	
			0 to 20m/	A	5µA			1.66µA	
		Cummant	4 to 20m/		4µA	0 to 120	000	1.33µA	
		Current	User rang setting		1.5µA	-12000 to 1	12000	0.83µA	
			oottiing						
Accuracy (Accuracy for	Ambient temperatu re 25±5°C	Within ±0.1 % (Voltage: ±10mV, Current: ±20µA)							
maximum analog output value)	Ambient temperatu re 0 to 55°C			Within ±0.3% (Voltage:	±30mV, Curre	nt: ±60µA)			
Conversion speed			80µs/channel						
Absolute	Voltage	±12V			channel				
movimum outout	0				channel				
maximum output	Current		21n	±12V	channel			- 21mA	
				±12V nA Ava	- ilable				
Output short circui		Between th	ne I/O termina	±12V nA Ava I and programmable con Betwe	ilable troller power s een output cha	upply : Pho nnels : No	insulatio	er insulation	
Output short circui		Between th	ne I/O termina	±12V nA Ava Il and programmable con Betwe een external supply powe	ilable troller power s een output cha r and analog c	upply : Pho nnels : No	insulatio	er insulation	
Output short circui Insulation method Occupied points	t protection	Between th	ne I/O termina	±12V nA Ava Il and programmable con Betwe een external supply powe 16 p	ilable troller power s een output cha	upply : Pho nnels : No	insulatio	er insulation	
Output short circui Insulation method Occupied points Connection termin	al	Between th	ne I/O termina	±12V nA Ava Il and programmable con Betwe een external supply powe 16 p 18-point te	ilable troller power s een output cha r and analog c ooints rrminal block	upply : Pho nnels : No	insulatio	er insulation	
maximum output Output short circui Insulation method Occupied points Connection termin Applicable wire siz	al		ne I/O termina Betwe solderless ter	±12V nA Ava Il and programmable con Betwe een external supply powe 16 p 18-point te 0.3 to 0	ilable troller power s een output cha r and analog c ooints rminal block 0.75mm ² FG terminal Other termir	upply : Pho nnels : No putput : Tra	: R1.25- RAV1.2 V1.25- : R1.25- : R1.25- termina	er insulation in r insulation 3, 1.25-YS3, 25-3,	
Output short circui Insulation method Occupied points Connection termin Applicable wire siz	al	R1.25-3 (A	ne I/O termina Betwe solderless ter	±12V nA Ava Il and programmable con Betwe een external supply powe 16 p 18-point te 0.3 to 0 minal with sleeve 24VDC +2	- ilable troller power s een output cha r and analog c oints rminal block .75mm ² FG terminal Other termir 20%, -15%	upply : Pho nnels : No putput : Tran	: R1.25- RAV1.2 V1.25- : R1.25- : R1.25- termina	er insulation in r insulation 3, 1.25-YS3, 25-3, YS3A 3 (A solderless al with sleeve	
Output short circui Insulation method Occupied points Connection termin Applicable wire siz	al ze ess terminal	R1.25-3 (A cannot be u	ne I/O termina Betwe solderless ter sed.)	±12V nA Ava Il and programmable con Betwe een external supply powe 16 p 18-point te 0.3 to 0 minal with sleeve 24VDC +2 Ripple, spike 50	- ilable troller power s een output cha r and analog c ioints rminal block .75mm ² FG terminal Other termir 20%, -15% 00mVP-P or le	upply : Pho nnels : No putput : Tran nals than FG	: R1.25- RAV1.2 V1.25- : R1.25- termina cannot	er insulation n r insulation 3, 1.25-YS3, 25-3, YS3A 3 (A solderless al with sleeve be used.)	
Output short circui Insulation method Occupied points Connection termin Applicable wire siz	al ze ess terminal	R1.25-3 (A	ne I/O termina Betwe solderless ter sed.) rent: 2.5A,	±12V nA Ava Il and programmable con Betwe een external supply powe 16 p 18-point te 0.3 to 0 minal with sleeve 24VDC +2	- ilable troller power s een output cha r and analog c oints rminal block .75mm ² FG terminal Other termir 20%, -15%	upply : Pho nnels : No putput : Tran nals than FG ss ent: 2.5A,	: R1.25- RAV1.2 V1.25- : R1.25- termina cannot	er insulation n r insulation 3, 1.25-YS3, 25-3, YS3A 3 (A solderless al with sleeve be used.) current: 2.5A, thin 230µs	
Output short circui Insulation method Occupied points Connection termin Applicable wire siz Applicable solderle	al ess terminal	R1.25-3 (A cannot be u Inrush curr	solderless ter sed.)	±12V nA Ava I and programmable con Betwee een external supply power 16 p 18-point te 0.3 to 0 minal with sleeve 24VDC +: Ripple, spike 50 Inrush current: 2.5A,	- ilable troller power s een output cha r and analog c ooints rminal block .75mm ² FG terminal Other termir 20%, -15% 00mVP-P or le Inrush curr	upply : Pho nnels : No putput : Train nals than FG ss ent: 2.5A, 230µs	: R1.25- RAV1.2 V1.25- : R1.25- termina cannot	er insulation n r insulation 3, 1.25-YS3, 25-3, YS3A 3 (A solderless al with sleeve be used.) current: 2.5A,	
Output short circui Insulation method Occupied points Connection termin Applicable wire siz	al ess terminal	R1.25-3 (A cannot be u Inrush curr within	solderless ter sed.)	±12V nA Ava I and programmable con Betwee een external supply power 16 p 18-point te 0.3 to 0 minal with sleeve 24VDC +2 Ripple, spike 50 Inrush current: 2.5A, within 260µs	- ilable troller power s een output cha r and analog c joints rminal block 7.5mm ² FG terminal Other termir 20%, -15% 00mVP-P or le Inrush curr within 2	upply : Pho nnels : No putput : Tran nals than FG ss ent: 2.5A, 230µs DA BA	: R1.25- RAV1.2 V1.25- : R1.25- termina cannot	er insulation n r insulation 3, 1.25-YS3, 25-3, YS3A 3 (A solderless al with sleeve be used.) current: 2.5A, thin 230µs	

(2) Names of parts



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 the operation 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
4)	FG terminal	Frame ground terminal

(3) List of I/O signals

Signal direction	D/A converter module \rightarrow CPU module	Signal direction	CPU module \rightarrow D/A converter module
Device No.	Signal name	Device No.	Signal name
X0	Module READY	Y0	Use prohibited ^{*1}
X1		Y1	CH1 Output enable/disable flag
X2		Y2	CH2 Output enable/disable flag
Х3		Y3 ^{*2}	CH3 Output enable/disable flag
X4	Use prohibited ^{*1}	Y4 ^{*2}	CH4 Output enable/disable flag
X5		Y5	CH5 Output enable/disable flag
X6		Y6	CH6 Output enable/disable flag
X7		Y7	CH7 Output enable/disable flag
X8	High resolution mode status flag	Y8	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
ХВ	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

List of I/O signals of the D/A converter module

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.

(4) Buffer memory

The following explanation is mentioned based on the Q64DAN with 4-channel analog output (CH1 to CH4).

Addr	ess	Name	Default ^{*2}	Read/write ^{*3}	
Hexadecimal	Decimal	Name	Delault	Read/write	
Он	0	D/A conversion enable/disable	Q62DAN : Зн Q64DAN : Fн	R/W	
1н	1	CH1 Digital value	0	R/W	
2н	2	CH2 Digital value	0		
3н	3	CH3 Digital value ^{*1}	0	R/W	
4н	4	CH4 Digital value ^{*1}	0	R/W	
5н	5		-	-	
6н	6		-	-	
7н	7		-	-	
8н	8	System area	-	-	
9н	9		-	-	
Ан	10		-	-	
Вн	11	CH1 Set value check code	0	R	
Сн	12	CH2 Set value check code	0	R	
Dн	13	CH3 Set value check code ^{*1}	0	R	
Ен	14	CH4 Set value check code ^{*1}	0	R	
Fн	15		-	-	
10н	16	Sustem erec	-	-	
11н	17	System area	-	-	
12н	18		-	-	
13н	19	Error code	0	R/W	
14н	20	Setting range	0	R	
15н	21	System area	-	-	
16н	22	Offset/gain setting mode Offset specification	0	R/W	
17н	23	Offset/gain setting mode Gain specification	0	R/W	
18н	24	Offset/gain adjustment value specification	0	R/W	

Buffer memory assignment of the D/A converter module

*1: For Q62DAN, the buffer memory areas for CH3 and CH4 are system areas.

*2: This is the initial value set after the power is turned on or the programmable controller CPU is reset.

*3: Indicates whether reading from and writing to a sequence program are enabled. R: Read enabled

W: Write enabled

Appendix 5 Comparison of Timers and Counters

Appendix 5.1 Comparison of timers and counters

Fund	ction	QCPU/QnACPU	AnUCPU	AnACPU	AnNCPU	
Low-speed	Measurement unit	 100ms (default value) Can be changed in the range of 1 to 100ms (parameter) 	• Fixed at 100ms			
timer	Specification method					
	Measurement unit	 10ms (default value) Can be changed in the range of 1 to 100ms (parameter) 	• Fixed at 10ms			
High-speed timer	Specification method	High-speed timer specified H K100 H K100	* High-speed time parameters.	{т2	00 00	
	Measurement unit	 Same measurement unit as the low-speed timer 	• Fixed at 10ms			
Retentive timer	Specification method					
	Measurement unit	 Same measurement unit as the high-speed timer 				
		High-speed timer specified High-speed timer setting: Conducted by sequence program.	• None			
Setting range for set values		• 1 to 32767	• 1 to 32767			
Processing for set value 0		Momentarily ON	• Infinity (does no	t time out)		
	Contact	 Enabled (only Z0 and Z1 are usable) 	Enabled		Disabled	
Indexing	Coil	Enabled (only Z0 and Z1 are usable)	Disabled		Disabled	
maching	Set value	• Disabled	Disabled		Disabled	
	Current value	Enabled (Z0 to Z15 are usable)	Enabled		Enabled	
Update process value Contact ON/OFF	ing for current	When OUT Tn instruction is executed	When END processing is executed			
	Processing					

Comparison of timers

(1) Cautions on using timers

Q/QnACPU updates the current value of timers and turns on or off the contacts of them at the execution of the OUT $T\Box$ instruction.

Therefore, if (Current value) \geq (Set value) when the timer coil is turned on, the contact of that timer is turned on.

When creating a program in which the operation of the timer contact triggers the operation of another timer, set the timer that operates later first.

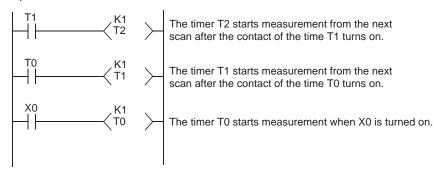
In the following cases, all the timers turns on at the same scan if the program is created in the order the timers operates.

• When the set value is smaller than a scan time with high-speed timers

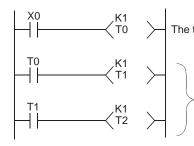
• When the set value is "1" with low-speed timers

Example

• For timers T0 to T2, the program is created in the order the timer operates later.



• For timers T0 to T2, the program is created in the order of timer operation.



The timer T0 starts measurement when X0 is turned on.

The contacts of the timers T1 and T2 turn on when the contact of the timer T0 turns on.

Appendix 5.2 Comparison of counters

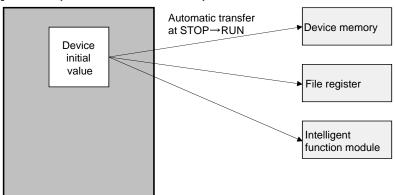
Function		QCPU/QnACPU	AnUCPU	AnACPU	AnNCPU
Specification method					
	Contact	• Enabled (only Z0 and Z1 are usable)	Enabled	Enabled	
la da sia a	Coil	Enabled (only Z0 and Z1 are usable)	Enabled	Enabled	
Indexing	Set value	• Disabled	Disabled	Disabled	
	Current value	Enabled (Z0 to Z15 are usable)	Enabled	Enabled	
Update processing for current value		When OUT Tn instruction is executed	When END pro	ocessing is exect	uted
Contact ON/C	OFF processing				

Comparison of counters

Appendix 6 Setting device initial values

The device initial value set in the peripheral device beforehand can be automatically transferred to the device memory, file register, and the intelligent function module when the status changed from STOP to RUN. When the device initial values are set, the initial setting program is unnecessary.

Program memory, Standard ROM, and Memory card



Required Setting

To use the setting function of the device initial values, set "Device Memory" and "Device Initial Value" in the project data list.

The device initial value file reflecting the setting needs to be written to be stored in the program memory, the standard ROM, or the memory card.

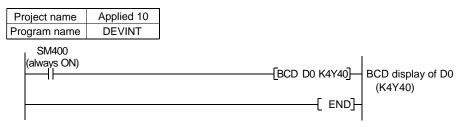
Devices Where the Initial Values can be Set

Device	Applica	Device name	Applica	Device	Applica	Device	Applica	Device	Applica
name	bility	Device name	bility	name	bility	name	bility	name	bility
Х	×	T (Contact)	×	FD	×	SZ	×	U□\G	0
Y	×	T (Coil)	×	В	×	S	×	J□\X	×
М	×	T (Current value)	0	SB	×	TR	×	J□\Y	×
L	×	C (Contact)	×	W	0	BL	×	J□\B	×
F	×	C (Coil)	×	SW	0	U	×	J⊡∖SB	×
SM	×	C (Current value)	0	G	×	J	×	J⊡\W	0
FX	×	ST (Contact)	×	R	0	ZR	0	J⊡∖SW	0
FY	×	ST (Coil)	×	Р	×			BL⊡∖S	×
V	×	ST (Current value)	0	I	×			BL⊡\TR	×
DX	×	D	0	Ν	×				
DY	×	SD	0	Z	×				

The following table shows the list of devices where the device initial values can be set.

Preparations for the Initial Value Setting Operation Check

(1) Create the following program to check operation easily.



(2) Configure the setting in the "Program" tab of the PLC parameter in the project data list as shown below.

	Program Name	Execute Type	Fixed Scan Interval	In Unit	٠
1	DEVINT	Scan 👻		-	
2		-		+	
3		-		•	
4		-		-	
5		-		-	
6		-		-	
7		-		-	

Appendix 6.1 Setting device memories

 Set device initial value to a device memory. Set D0 to "1234" in the following procedure.

Sort Property
Data Type: Device Memory Data Name: MAIN Program Language: Result Type: Inherent Property:
Data Type: Device Memory Data Name: MAIN Program Language: Result Type: Inherent Property:
Device Memory
Data Name: MAIN T Program Language: Result Type:
MAIN Program Language: Result Type: Inherent Property:
Result Type:
Inherent Property:
Inherent Property:
Inherent Property:
Inherent Property:
Use MC/MCR Use EN/ENO
L BREMINO
OK Cancel
Device Homeon MAIN
Device Memory MAIN
+0 +1
Device Memory MAIN
+0 +1
DO
Device Memory MAIN
+0 +1
D0 0
Į Ļ
(To the next page)

1) Right-click "Device Memory" in the project data list and click [Add New Data].

2) The dialog box on the left is displayed. Click the OK button.In this example, set "MAIN" as Data Name.

- 3) Select a cell where a device value is set.
- 4) Enter "D0" to the cell.
- 5) Press the Enter key. The device "D0" is set in the device number display area.

previous	
+0 1234	41 +1
	mory MA

- 6) Click the cell D0 and enter "1234". Press the Enter key.
 The initial value is set to D0.
- (2) Set a device range of the device initial value.In this example, set D0 to D9 as the device initial value.
- Parameter Marameter Intelligent Function Module 0 B Global Device Comment Program Setting 3 POU Program DEVINT Local Device Comment Device Memory 🗎 MAIN Add New Data... vice Initial Value Ľ <u>S</u>ort 6 Property... New Data Data Type: Device Initial Value • Data <u>N</u>ame: MAIN -🔲 Use Macrogode Use MC/MCR Use EN/ENO Cancel OK Device Initial Value MAIN Points Start End 1 10 D0 D9 2 3 4 (To the next page)
- 1) Right-click "Device Initial Value" and click [Add New Data].

2) The dialog box on the left is displayed. Click the OK button.In this example, set "MAIN" as Data Name.

3) The dialog box on the left is displayed. Enter "D0" for Start, and "D9" for End.

Ind Comment 10 00 00	
10 D0 D9 Image and the device data that is set above will be write to PLC. execute Device Memory Diversion 'operation when the device initial range setting is change Method Device Memory Diversion Device Memory Diversion OK Cancel Diversion Diversion Diversion Diversion Diversion Diversion Diversion Diversion Diversion Diversion Diversion Diversion Diversion Diversin	
Execute Device Memory Diversion operation when the device initial range setting is change Alethod Device Memory Diversion Device Memory Diversion Ex/Start Mann Print Pregiew Print Setting OK Cancel FT Application The device initial value range setting currently will be decide, and the initial value data will be read newly after clearing. After it execute, this operation cannot be canceled because the setting is decided. Do you want to divert device memory? Yes No MELSOFT Application MELSOFT Application MELSOFT Application MELSOFT Application MELSOFT Application	
Execute Device Memory Diversion operation when the device initial range setting is change Alethod Device Memory Diversion Device Memory Diversion Ex/Start Mann Print Pregiew Print Setting OK Cancel FT Application The device initial value range setting currently will be decide, and the initial value data will be read newly after clearing. After it execute, this operation cannot be canceled because the setting is decided. Do you want to divert device memory? Yes No MELSOFT Application MELSOFT Application MELSOFT Application MELSOFT Application MELSOFT Application	
Execute Device Memory Diversion operation when the device initial range setting is change Alethod Device Memory Diversion Device Memory Diversion Ex/Start Mann Print Pregiew Print Setting OK Cancel FT Application The device initial value range setting currently will be decide, and the initial value data will be read newly after clearing. After it execute, this operation cannot be canceled because the setting is decided. Do you want to divert device memory? Yes No MELSOFT Application MELSOFT Application MELSOFT Application MELSOFT Application MELSOFT Application	
Execute Device Memory Diversion operation when the device initial range setting is change Alethod Device Memory Diversion Device Memory Diversion Ex/Start Mann Print Pregiew Print Setting OK Cancel FT Application The device initial value range setting currently will be decide, and the initial value data will be read newly after clearing. After it execute, this operation cannot be canceled because the setting is decided. Do you want to divert device memory? Yes No MELSOFT Application MELSOFT Application MELSOFT Application MELSOFT Application MELSOFT Application	
Execute Device Memory Diversion operation when the device initial range setting is change Alethod Device Memory Diversion Device Memory Diversion Ex/Start Mann Print Pregiew Print Setting OK Cancel FT Application The device initial value range setting currently will be decide, and the initial value data will be read newly after clearing. After it execute, this operation cannot be canceled because the setting is decided. Do you want to divert device memory? Yes No MELSOFT Application MELSOFT Application MELSOFT Application MELSOFT Application MELSOFT Application	
Execute Device Memory Diversion operation when the device initial range setting is change Alethod Device Memory Diversion Device Memory Diversion Ex/Start Mann Print Pregiew Print Setting OK Cancel FT Application The device initial value range setting currently will be decide, and the initial value data will be read newly after clearing. After it execute, this operation cannot be canceled because the setting is decided. Do you want to divert device memory? Yes No MELSOFT Application MELSOFT Application MELSOFT Application MELSOFT Application MELSOFT Application	
Execute Device Memory Diversion operation when the device initial range setting is change Alethod Device Memory Diversion Device Memory Diversion Ex/Start Mann Print Pregiew Print Setting OK Cancel FT Application The device initial value range setting currently will be decide, and the initial value data will be read newly after clearing. After it execute, this operation cannot be canceled because the setting is decided. Do you want to divert device memory? Yes No MELSOFT Application MELSOFT Application MELSOFT Application MELSOFT Application MELSOFT Application	
Execute Device Memory Diversion operation when the device initial range setting is change Alethod Device Memory Diversion Device Memory Diversion Ex/Start Mann Print Pregiew Print Setting OK Cancel FT Application The device initial value range setting currently will be decide, and the initial value data will be read newly after clearing. After it execute, this operation cannot be canceled because the setting is decided. Do you want to divert device memory? Yes No MELSOFT Application MELSOFT Application MELSOFT Application MELSOFT Application MELSOFT Application	
Execute Device Memory Diversion operation when the device initial range setting is change Alethod Device Memory Diversion Device Memory Diversion Ex/Start Mann Print Pregiew Print Setting OK Cancel FT Application The device initial value range setting currently will be decide, and the initial value data will be read newly after clearing. After it execute, this operation cannot be canceled because the setting is decided. Do you want to divert device memory? Yes No MELSOFT Application MELSOFT Application MELSOFT Application MELSOFT Application MELSOFT Application	
Execute Device Memory Diversion operation when the device initial range setting is change Alethod Device Memory Diversion Device Memory Diversion Ex/Start Mann Print Pregiew Print Setting OK Cancel FT Application The device initial value range setting currently will be decide, and the initial value data will be read newly after clearing. After it execute, this operation cannot be canceled because the setting is decided. Do you want to divert device memory? Yes No MELSOFT Application MELSOFT Application MELSOFT Application MELSOFT Application MELSOFT Application	Į
tyEnd Device Memory Prints Setting OK Cancel	ged.
Gend For Diversion ts/Start MAIN Print Preview Print Setting OK Cancel FT Application The device initial value range setting currently will be decide, and the initial value data will be read newly after clearing. After it execute, this operation cannot be canceled because the setting is decided. Do you want to divert device memory? Yes No MELSOFT Application MELSOFT Application titlat Value MAIN total value MAIN	
FT Application The device initial value range setting currently will be decide, and the initial value data will be read newly after clearing. After it execute, this operation cannot be canceled because the setting is decided. Do you want to divert device memory? Yes MELSOFT Application Image: Completed. Image: Completed. <t< td=""><td></td></t<>	
FT Application The device initial value range setting currently will be decide, and the initial value data will be read newly after clearing. After it execute, this operation cannot be canceled because the setting is decided. Do you want to divert device memory? Yes MELSOFT Application Image: Completed. Image: Completed. <t< td=""><td>1</td></t<>	1
The device initial value range setting currently will be decide, and the initial value data will be read newly after clearing. After it execute, this operation cannot be canceled because the setting is decided. Do you want to divert device memory? Yes No Yes No WELSOFT Application Image: Completed. OK OK OK Image: Completed. OK Image: Completed. Image: Completed. Image: Completed. Image: Completed. <td>-</td>	-
The device initial value range setting currently will be decide, and the initial value data will be read newly after clearing. After it execute, this operation cannot be canceled because the setting is decided. Do you want to divert device memory? Yes No Yes No WELSOFT Application Image: Completed. OK OK OK Image: Completed. OK Image: Completed. Image: Completed. Image: Completed. Image: Completed. <td></td>	
The device initial value range setting currently will be decide, and the initial value data will be read newly after clearing. After it execute, this operation cannot be canceled because the setting is decided. Do you want to divert device memory? Yes No Yes No WELSOFT Application Image: Completed. OK OK OK Image: Completed. OK Image: Completed. Image: Completed. Image: Completed. Image: Completed. <td></td>	
The device initial value range setting currently will be decide, and the initial value data will be read newly after clearing. After it execute, this operation cannot be canceled because the setting is decided. Do you want to divert device memory? Yes No Yes No WELSOFT Application Image: Completed. OK OK OK Image: Completed. OK Image: Completed. Image: Completed. Image: Completed. Image: Completed. <td>X</td>	X
and the initial value data will be read newly after clearing. After it execute, this operation cannot be canceled because the setting is decided. Do you want to divert device memory? Yes No MELSOFT Application X MELSOFT Application X Completed. OK IIII Value MAIN Points Start End Comment A	
After it execute, this operation cannot be canceled because the setting is decided. Do you want to divert device memory? Yes No MELSOFT Application MELSOFT Application Image: Completed. OK Image: Completed. Image: Completed. Image: Completed. Image: Co	
Do you want to divert device memory?	
Yes No MELSOFT Application Image: Completed. Image: Completed. Image: Completed.	
MELSOFT Application Completed. Complete	
Completed.	
Completed.	
Completed.	
Completed.	
Itial Value MAIN	
Points Start End Comment 🔺	
•	
range and the device data that is set above will be write to PLC.	
execute 'Device Memory Diversion' operation when the device initial range setting is change Nethod Device Memory Diversion	
Device Memory	ed.
ts/Start MAIN C/Lnd for Diversion ts/Start MAIN Cevice Memory Diversion	ed.
	ed.

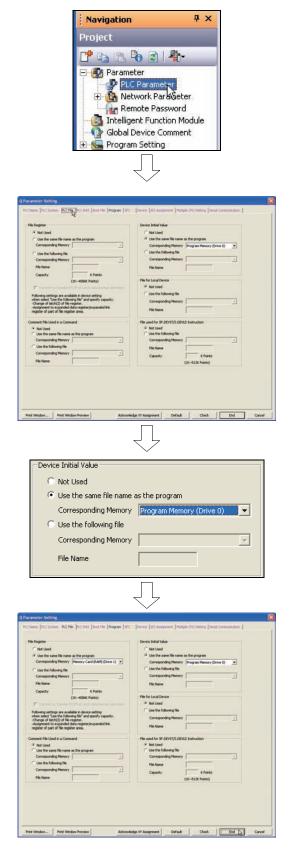
- 4) Select the device memory "MAIN", where the device initial value is set.
- 5) Click the Device Memory Diversion button.

 The dialog box on the left is displayed. Click the OK button.

- 7) Click the OK button.
- 8) Click the OK button to close the dialog box.

Appendix 6.2 Specifying file names for device initial value

Set a file to be used as a device initial value file in the "PLC File" tab of the PLC parameter in the project data list.



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

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

- The screen is switched. Click "Use the same file as the program" in "Device Initial Value".
- 4) Confirm that "Program Memory (Drive 0)" is displayed.
- 5) Click the End button to accept the setting.

Appendix 6.3 Checking the operation of device initial values

(1) Writing data to the CPU

Write programs, parameters and device initial values to the CPU.

ine Data Operation							
onnection Channel List							
erial Port PLC Module Connect	ion(USB)						System Image
 C Re	ad ⊙ <u>W</u> rite	⊂ <u>V</u> er	ify	CD	elete		
PLC Module	nt Function Module Exec	ution Target I	Data(N	lo 1	Yes)		
itle							
🔠 Edit Data	Parameter+Program	Select All	Cano	el All Sel	ections		
Module Name,	/Data Name	Title	Target	Detail	Last Change	Target Memory	Size
- C Applied 10						Duran Managulo	27
- C Data	a)			Detail		Program Memory/D	10
DEVINT	c)			Datai	2010/01/27 13:46:29	12	2164 Bytes
- O Parameter				1	2010/01/21 10/10/25		21010700
	note Password/Switch Setting				2009/11/05 16:19:27		484 Bytes
Global Device Comm				1			and the second second
E Device Memory				Detail			
📖 👼 MAIN					2012/06/01 15:33:27	1	
🖃 👼 Device Initial Value							
📖 👼 MAIN					2012/06/01 15:44:03	8	76 Bytes
Necessary Setting(No : Writing Size 2,724Bytes	Setting / Already Set) S	5et if it is need	led(No :			se Volume 3,028Bytes	Refres <u>h</u>
lated Eunctions <<		5	1	1		Exe	cute Close
Remote Set Clock	PLC User Data Write	Title I	Format Pl			range PLC Memory	

POINT

In this section, the device initial value setting file also can be written to the CPU simultaneously with the parameters and programs.

Reason: The device initial value setting file uses the same target memory area. (In this section, the program memory is used for all.)

Generally, the target memory area for the device initial value setting file is different from the one for parameter and program files. Each of them can be written into each target memory separately.

(2) Operation check

When the RUN/STOP/RESET switch on the CPU is switched from STOP to RUN, or at the timing of the power-on, the values of the device initial value setting file are transferred to the CPU device.

In this section, the initial set value of the register D0, "1234" is displayed on the LED (K4Y40) of the demonstration machine.

POINT

When the latch range overlaps with the device initial value, the device initial value is given priority.

Appendix 7 Inline Structured Text

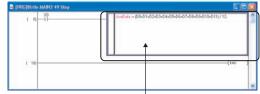
Inline structured text is a function to edit/monitor a program by creating an inline structured text box that displays an ST program at the coil instruction area on the ladder editor of the project with labels.

With this function, a numeric value operation or a character string process can be easily created in the ladder program.

Select "Enable function block call 'from ladder to Structured Ladder/FBD or ST' and 'from Structured Ladder/FBD or ST to ladder'" under [Tool] \rightarrow [Options] \rightarrow [Compile] \rightarrow [Basic Setting] to use the inline structured text function.

< Using ladder >			
🖨 [PRG]Write MABH 44 Step			E 18 6
		10	Allows]
	[+	00	ALC:NO.
	[*	D1	Addies 1
<u></u>	[+	D2	Addres 1
	[*	D3	AbDes]
	[*	D4	Addres]
	[*	05	A804+]
	[+	D6	Alfan]
	[+	D7	Addres 1
<u>.</u>	{*	De	AbDes]
	[+	D9	AllCus]
	{*	D10	Alles]
	-{+	D11	A80++]
(40)	{1 *d50=	к12	Antes 1
45			-[END]

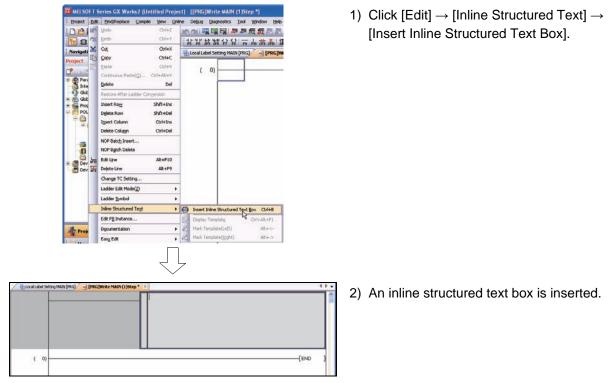
< Using Inline structured text >



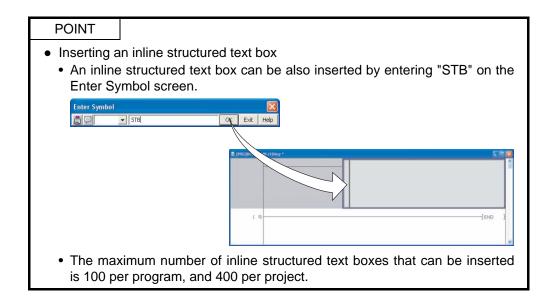
Inline structured text box (STB)

Appendix 7.1 Editing inline structured text





App. - 51



(2) Editing an inline structured text program Double-click the inline structured text box.

The editing method of the inline structured text is the same as that of the ST language.

For editing programs in the ST language, refer to the GX Works2 Version 1 Operating Manual Structured Project and MELSEC-Q/L/F Structured Programming Manual (Fundamentals).

(3) Deleting an inline structured text box



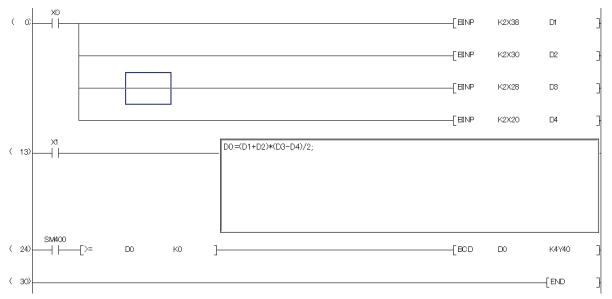
1) Select an inline structured text box to be deleted.

∏Write MAIN 25 Step	
(0)	
(18)	END-

2) Click [Edit] \rightarrow [Delete].

One ladder block containing the inline structured text box is deleted.

(4) Program example of the inline structured text



* Program example without the inline structured text



Appendix 7.2 Precautions on using the inline structured text

- (1) Precautions on creating ladder programs
 - One inline structured text box can be created for one ladder block.
 - Both an FB and an inline structured text box cannot be used in a ladder block.
 - When the creation of an inline structured text box is attempted at the contact instruction area, an inline structured text box is created at the coil instruction area.
 - A ladder program cannot be edited if an unconverted inline structured text program exists on the ladder editor. Edit a program after converting it.
- (2) Precautions on editing in an inline structured text box
 - A maximum number of characters that can be entered is 2048.
 - Up to 23 local labels can be used in an inline structured text box. (Excluding constants)
 - The following data type labels cannot be used.
 - Counter
 - Timer
 - Retentive timer
 - Pointer
 - Structure
 - Array
 - Function block
 - Lower-case device names cannot be used as labels regardless of the option setting.
 - Instructions cannot be entered using the Selection window.
 - The template function of the ST program cannot be used.
 - Label candidates are not displayed.
 - The editing status cannot be recovered to the previous status by the operation such as clicking [Edit] → [Undo].
- (3) Verifying a program containing an inline structured text

When a project or project revision is verified with "Program" selected, an inline structured text is not the verification target. When verifying a program containing an inline structured text, select "Program File" to verify.

- (4) Finding or replacing a program containing an inline structured text The inside of an inline structured text box are not the target of the following Find/Replace functions.
 - Replace String
 - Find Device/Replace Device
 - Find Instruction/Replace Instruction
 - Find Contact or Coil
 - Change Open/Close Contact
 - Device Batch Replace

When searching the inside of an inline structured text box, use Cross Reference or Device List.

For Cross Reference and Device List, refer to the GX Works2 Version 1 Operating Manual Common.

(5) Copying an inline structured text box

When copying an inline structured text box, select a ladder block including the left side of the left power rail. Contacts of a ladder block containing an inline structured text box only or an inline structured text box only cannot be copied. A ladder block containing an unconverted inline structured text box also cannot be copied. Copy a ladder block after converting the program.

(6) Jumping to an inline structured text box during monitoring At search during monitoring or verification for an inline structured text box, the cursor is moved to the target position, but the range is not selected.

Appendix 8 Battery

Install a battery (Q6BAT, Q7BAT, or Q8BAT) in the CPU module to hold data in the program memory, standard RAM, and latch devices even if power failure occurs.

The following table shows the specifications of the batteries used for the CPU module.

ltem			Туре				
nem	Q6BAT	Q7BAT	Q8BAT				
Classification	Manganese dioxide li	thium primary battery	Manganese dioxide lithium primary battery (assembled battery)				
Initial voltage			3.0V				
Nominal current	1800mAh	5000mAh	18000mAh (1800mAh $ imes$ 10 pieces)				
Battery life when stored		Actually 5 years	(room temperature)				
Battery life when used	Refer to the Q	CPU User's Manual Hard	dware Design, Maintenance and Inspection.				
Lithium content	0.49kg	1.52kg 4.9kg					
Application	For data retention of	the program memory, sta	andard RAM, and latch device during power failure				
Accessory	-	Battery holder ^{*1}	Q8BAT connection cable ^{*2}				

*1: Included only when the Q7BAT-SET is purchased.

*2: Included only when the Q8BAT-SET is purchased.

The real number data includes the single-precision floating-point data and double-precision floating-point data.

- (1) Single-precision floating-point data
 - (a) Internal representation of real numbers

Internal representation of real numbers used in the CPU module is shown below.

Real number data can be represented as follows, using two word devices.

[Sign] 1. [Mantissa] × 2^[Exponent]

The bit configuration and the meaning of each bit are described below.

b31 b30 to	b23 b22	to	b16	b15		to		b0
b31 b30 to b2	23				b22 to	b0		
Sign Exponent (8	3 bits)			N	/lantissa (2	23 bits)		

- Sign...The most significant bit, b31, is the sign bit.
 Positive, 1: Negative
- 2) Exponent

The 8 bits, b23 to b30, represent the exponent n of 2ⁿ.

The following shows the exponent n according to the binary values in b23 to b30.

b23 to b30	FFн	FЕн	FDн	5		81н	80н	7Fн	7Ен	(\$	02н	01н	00н
n	Not used	127	126		>	2	1	0	-1		$\langle \rangle$	-125	-126	Not used

3) Mantissa

Each of the 23 bits, b0 to b22, represents the "XXXXXX..." portion when the data is represented in binary, "1.XXXXXX...".

(b) Calculation example

The following shows the calculation examples. (The "X" in (nnnnn)X indicates the numeral system used.)

1) When "10" is stored

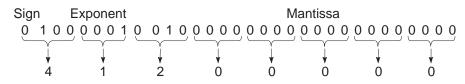
 $(10)_{10} \rightarrow (1010)_2 \rightarrow (1.010000.... \times 2^3)_2$

 Sign:
 Positive $\rightarrow 0$

 Exponent:
 $3 \rightarrow 82H \rightarrow (10000010)2$

 Mantissa:
 (010 00000 00000 000000)2

In this case, the value will be encoded as 41200000H.

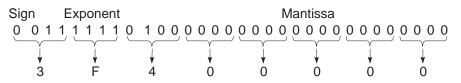


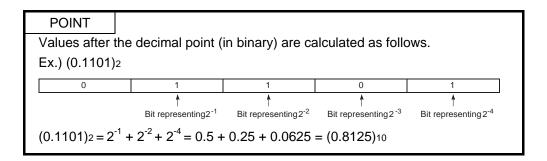
2) When "0.75" is stored

 $(0.75)_{10} \rightarrow (0.11)_2 \rightarrow (1.100.... \times 2^{-1})_2$

Sign:	Positive $\rightarrow 0$
Exponent:	$-1 \rightarrow 7EH \rightarrow (01111110)_2$
Mantissa:	(100 00000 00000 00000 00000)2

In this case, the value will be encoded as 3F400000H.





- (2) Double-precision floating-point data
 - (a) Internal representation of real numbers

Internal representation of real numbers used in the CPU module is shown below.

Real number data can be represented as follows, using four word devices.

[Sign] 1. [Mantissa] × 2^[Exponent]

The bit configuration and the meaning of each bit are described below.

							3	
b63 b62	to	b52 b51	to	b16	b15		to	b0
$\smile \smile$	~~~~							
b63 h	52 to b6	2				b0 to b51		

b63 b52 to b62 Sign Exponent (11 bits) b0 to b51 Mantissa (52 bits)

- 1) Sign ... The most significant bit, b63, is the sign bit.
 - 0: Positive, 1: Negative
- 2) Exponent
 - The 11 bits, b52 to b62, represent the exponent n of 2^n .

The following shows the exponent according to the binary values in b52 to b62.

b52 to b62	7FFн	7FEн	7FDH	(\$	400н	3FFн	3FEн	3FDH	3FCн	(\$	02н	01н	00н
n	Not used	1023	1022		\langle	1	0	-1	-2	-3		\langle	-1021	-1022	Not used

3) Mantissa

Each of the 52 bits, b0 to b51, represents the "XXXXXX..." portion when the data is represented in binary, "1.XXXXXX...".

(b) Calculation example

The following shows the calculation examples. (The "X" in (nnnnn)X indicates the numeral system used.)

1) When "10" is stored

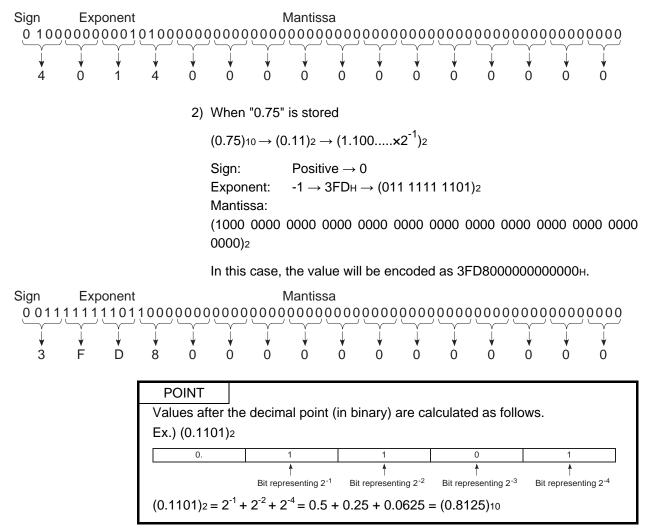
 $(10)_{10} \rightarrow (1010)_2 \rightarrow (1.010000.... \times 2^3)_2$

```
        Sign:
        Positive \rightarrow 0

        Exponent:
        3 \rightarrow 401 \text{H} \rightarrow (100\ 0000\ 0001)_2

        Mantissa:
        (0100\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 000\ 0000\ 000\ 000\ 000\ 000\ 000\ 000\ 000\ 000\ 000\ 000\ 000\ 000\ 000\ 000\ 000\ 000\ 000\ 000\ 000\ 000\ 000\ 000\ 000\ 000\ 000\ 000\ 000\ 000\ 000\ 000\ 000\ 000\ 000\ 000\ 000\ 000\ 000\ 000\ 000\ 000\ 000\ 000\ 000\ 000\ 00\ 000\ 000\ 000\ 00\ 00\ 000\ 000\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00
```

In this case, the value will be encoded as 401400000000000.



Mitsubishi Programmable Controller Training Manual Q-series advanced course (for GX Works2)

MODEL SCHOOL-APPLI-GXW2-E

13JW56

MODEL CODE

SH(NA)-081124ENG-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.