

MELSEC Q series

Programmable Controller

User's Manual
(Function Explanation,
Program Fundamentals)

QnUCPU

● SAFETY PRECAUTIONS ●

(Read these precautions before using this product.)

Before using this product, please read this manual and the relevant manuals carefully and pay full attention to safety to handle the product correctly.

In this manual, the safety precautions are classified into two levels: "⚠ WARNING" and "⚠ CAUTION".



Indicates that incorrect handling may cause hazardous conditions, resulting in death or severe injury.



Indicates that incorrect handling may cause hazardous conditions, resulting in minor or moderate injury or property damage.

Under some circumstances, failure to observe the precautions given under "⚠ CAUTION" may lead to serious consequences.

Observe the precautions of both levels because they are important for personal and system safety. Make sure that the end users read this manual and then keep the manual in a safe place for future reference.

[Design Precautions]



WARNING

- Configure safety circuits external to the programmable controller to ensure that the entire system operates safely even when a fault occurs in the external power supply or the programmable controller. Failure to do so may result in an accident due to an incorrect output or malfunction.

- (1) Configure external safety circuits, such as an emergency stop circuit, protection circuit, and protective interlock circuit for forward/reverse operation or upper/lower limit positioning.
- (2) The programmable controller stops its operation upon detection of the following status, and the output status of the system will be as shown below.

Status	Output
Overcurrent or overvoltage protection of the power supply module is activated.	All outputs are turned off.
The CPU module detects an error such as a watchdog timer error by the self-diagnostic function.	All outputs are held or turned off according to the parameter setting.

All outputs may turn on when an error occurs in the part, such as I/O control part, where the CPU module cannot detect any error. To ensure safety operation in such a case, provide a safety mechanism or a fail-safe circuit external to the programmable controller. For a fail-safe circuit example, refer to Chapter 10 LOADING AND INSTALLATION in the QCPU User's Manual (Hardware Design, Maintenance and Inspection).

- (3) Outputs may remain on or off due to a failure of an output module relay or transistor. Configure an external circuit for monitoring output signals that could cause a serious accident.

[Design Precautions]



WARNING

- In an output module, when a load current exceeding the rated current or an overcurrent caused by a load short-circuit flows for a long time, it may cause smoke and fire. To prevent this, configure an external safety circuit, such as a fuse.
- Configure a circuit so that the programmable controller is turned on first and then the external power supply.
If the external power supply is turned on first, an accident may occur due to an incorrect output or malfunction.
- For the operating status of each station after a communication failure, refer to relevant manuals for the network.
Incorrect output or malfunction due to a communication failure may result in an accident.
- When changing data of the running programmable controller from a peripheral connected to the CPU module or from a personal computer connected to an intelligent function module, configure an interlock circuit in the sequence program to ensure that the entire system will always operate safely. For program modification and operating status change, read relevant manuals carefully and ensure the safety before operation.
Especially, in the case of a control from an external device to a remote programmable controller, immediate action cannot be taken for a problem on the programmable controller due to a communication failure.
To prevent this, configure an interlock circuit in the sequence program, and determine corrective actions to be taken between the external device and CPU module in case of a communication failure.



CAUTION

- Do not install the control lines or communication cables together with the main circuit lines or power cables.
Keep a distance of 100mm (3.94 inches) or more between them.
Failure to do so may result in malfunction due to noise.
- When a device such as a lamp, heater, or solenoid valve is controlled through an output module, a large current (approximately ten times greater than normal) may flow when the output is turned from off to on.
Take measures such as replacing the module with one having a sufficient current rating.
- After the CPU module is powered on or is reset, the time taken to enter the RUN status varies depending on the system configuration, parameter settings, and/or program size. Design circuits so that the entire system will always operate safely, regardless of the time.

[Installation Precautions]



CAUTION

- Use the programmable controller in an environment that meets the general specifications in the QCPU User's Manual (Hardware Design, Maintenance and Inspection).
Failure to do so may result in electric shock, fire, malfunction, or damage to or deterioration of the product.
- To mount the module, while pressing the module mounting lever located in the lower part of the module, fully insert the module fixing projection(s) into the hole(s) in the base unit and press the module until it snaps into place.
Incorrect mounting may cause malfunction, failure or drop of the module.
When using the programmable controller in an environment of frequent vibrations, fix the module with a screw.
Tighten the screw within the specified torque range.
Undertightening can cause drop of the screw, short circuit or malfunction.
Overtightening can damage the screw and/or module, resulting in drop, short circuit, or malfunction.
- When using an extension cable, connect it to the extension cable connector of the base unit securely.
Check the connection for looseness.
Poor contact may cause incorrect input or output.
- When using a memory card, fully insert it into the memory card slot.
Check that it is inserted completely.
Poor contact may cause malfunction.
- Shut off the external power supply for the system in all phases before mounting or removing the module. Failure to do so may result in damage to the product.
A module can be replaced online (while power is on) on any MELSECNET/H remote I/O station or in the system where a CPU module supporting the online module change function is used.
Note that there are restrictions on the modules that can be replaced online, and each module has its predetermined replacement procedure.
For details, refer to the relevant sections in the QCPU User's Manual (Hardware Design, Maintenance and Inspection) and in the manual for the corresponding module.
- Do not directly touch any conductive part of the module.
Doing so can cause malfunction or failure of the module.
- When using a Motion CPU module and modules designed for motion control, check that the combinations of these modules are correct before applying power.
The modules may be damaged if the combination is incorrect.
For details, refer to the user's manual for the Motion CPU module.

[Wiring Precautions]



WARNING

- Shut off the external power supply for the system in all phases before wiring.
Failure to do so may result in electric shock or damage to the product.
- After wiring, attach the included terminal cover to the module before turning it on for operation.
Failure to do so may result in electric shock.



CAUTION

- Ground the FG and LG terminals to the protective ground conductor dedicated to the programmable controller.
Failure to do so may result in electric shock or malfunction.
- Use applicable solderless terminals and tighten them within the specified torque range. If any spade solderless terminal is used, it may be disconnected when the terminal screw comes loose, resulting in failure.
- Check the rated voltage and terminal layout before wiring to the module, and connect the cables correctly.
Connecting a power supply with a different voltage rating or incorrect wiring may cause a fire or failure.
- Connectors for external connection must be crimped or pressed with the tool specified by the manufacturer, or must be correctly soldered.
Incomplete connections could result in short circuit, fire, or malfunction.
- Tighten the terminal screw within the specified torque range.
Undertightening can cause short circuit, fire, or malfunction.
Overtightening can damage the screw and/or module, resulting in drop, short circuit, or malfunction.
- Prevent foreign matter such as dust or wire chips from entering the module.
Such foreign matter can cause a fire, failure, or malfunction.

[Wiring Precautions]



WARNING

- A protective film is attached to the top of the module to prevent foreign matter, such as wire chips, from entering the module during wiring.
Do not remove the film during wiring.
Remove it for heat dissipation before system operation.
- Mitsubishi programmable controllers must be installed in control panels.
Connect the main power supply to the power supply module in the control panel through a relay terminal block.
Wiring and replacement of a power supply module must be performed by maintenance personnel who is familiar with protection against electric shock. (For wiring methods, refer to the QCPU User's Manual (Hardware Design, Maintenance and Inspection)).

[Startup and Maintenance Precautions]



WARNING

- Do not touch any terminal while power is on.
Doing so will cause electric shock.
- Correctly connect the battery connector.
Do not charge, disassemble, heat, short-circuit, solder, or throw the battery into the fire.
Doing so will cause the battery to produce heat, explode, or ignite, resulting in injury and fire.
- Shut off the external power supply for the system in all phases before cleaning the module or retightening the terminal screws or module fixing screws.
Failure to do so may result in electric shock.
Undertightening the terminal screws can cause short circuit or malfunction.
Overtightening can damage the screw and/or module, resulting in drop, short circuit, or malfunction.

[Startup and Maintenance Precautions]

CAUTION

- Before performing online operations (especially, program modification, forced output, and operation status change) for the running CPU module from the peripheral connected, read relevant manuals carefully and ensure the safety.
Improper operation may damage machines or cause accidents.
- Do not disassemble or modify the modules.
Doing so may cause failure, malfunction, injury, or a fire.
- Use any radio communication device such as a cellular phone or PHS (Personal Handy-phone System) more than 25cm (9.85 inches) away in all directions from the programmable controller.
Failure to do so may cause malfunction.
- Shut off the external power supply for the system in all phases before mounting or removing the module. Failure to do so may cause the module to fail or malfunction.
A module can be replaced online (while power is on) on any MELSECNET/H remote I/O station or in the system where a CPU module supporting the online module change function is used.
Note that there are restrictions on the modules that can be replaced online, and each module has its predetermined replacement procedure.
For details, refer to the relevant sections in the QCPU User's Manual (Hardware Design, Maintenance and Inspection) and in the manual for the corresponding module.
- After the first use of the product, do not mount/remove the module to/from the base unit, and the terminal block to/from the module more than 50 times (IEC 61131-2 compliant) respectively.
Exceeding the limit of 50 times may cause malfunction.
- Do not drop or apply shock to the battery to be installed in the module.
Doing so may damage the battery, causing the battery fluid to leak inside the battery.
If the battery is dropped or any shock is applied to it, dispose of it without using.
- Before handling the module, touch a grounded metal object to discharge the static electricity from the human body.
Failure to do so may cause the module to fail or malfunction.

[Disposal Precautions]



CAUTION

- When disposing of this product, treat it as industrial waste.
When disposing of batteries, separate them from other wastes according to the local regulations.
(For details of the battery directive in EU member states, refer to the QCPU User's Manual (Hardware Design, Maintenance and Inspection).)

[Transportation Precautions]



CAUTION

- When transporting lithium batteries, follow the transportation regulations.
(For details of the regulated models, refer to the QCPU User's Manual (Hardware Design, Maintenance and Inspection).)

● CONDITIONS OF USE FOR THE PRODUCT ●

- (1) Mitsubishi programmable controller ("the PRODUCT") shall be used in conditions;
- i) where any problem, fault or failure occurring in the PRODUCT, if any, shall not lead to any major or serious accident; and
 - ii) where the backup and fail-safe function are systematically or automatically provided outside of the PRODUCT for the case of any problem, fault or failure occurring in the PRODUCT.

- (2) The PRODUCT has been designed and manufactured for the purpose of being used in general industries.

MITSUBISHI SHALL HAVE NO RESPONSIBILITY OR LIABILITY (INCLUDING, BUT NOT LIMITED TO ANY AND ALL RESPONSIBILITY OR LIABILITY BASED ON CONTRACT, WARRANTY, TORT, PRODUCT LIABILITY) FOR ANY INJURY OR DEATH TO PERSONS OR LOSS OR DAMAGE TO PROPERTY CAUSED BY the PRODUCT THAT ARE OPERATED OR USED IN APPLICATION NOT INTENDED OR EXCLUDED BY INSTRUCTIONS, PRECAUTIONS, OR WARNING CONTAINED IN MITSUBISHI'S USER, INSTRUCTION AND/OR SAFETY MANUALS, TECHNICAL BULLETINS AND GUIDELINES FOR the PRODUCT.

("Prohibited Application")

Prohibited Applications include, but not limited to, the use of the PRODUCT in;

- Nuclear Power Plants and any other power plants operated by Power companies, and/or any other cases in which the public could be affected if any problem or fault occurs in the PRODUCT.
- Railway companies or Public service purposes, and/or any other cases in which establishment of a special quality assurance system is required by the Purchaser or End User.
- Aircraft or Aerospace, Medical applications, Train equipment, transport equipment such as Elevator and Escalator, Incineration and Fuel devices, Vehicles, Manned transportation, Equipment for Recreation and Amusement, and Safety devices, handling of Nuclear or Hazardous Materials or Chemicals, Mining and Drilling, and/or other applications where there is a significant risk of injury to the public or property.

Notwithstanding the above, restrictions Mitsubishi may in its sole discretion, authorize use of the PRODUCT in one or more of the Prohibited Applications, provided that the usage of the PRODUCT is limited only for the specific applications agreed to by Mitsubishi and provided further that no special quality assurance or fail-safe, redundant or other safety features which exceed the general specifications of the PRODUCTS are required. For details, please contact the Mitsubishi representative in your region.

REVISIONS

*The manual number is given on the bottom left of the back cover.

Print date	Manual number	Revision
Dec., 2008	SH(NA)-080807ENG-A	First edition
Mar., 2009	SH(NA)-080807ENG-B	Revision because of function addition to Built-in Ethernet port QCPU (first five digits of the serial number is "11012" or later) Partial correction SAFETY PRECAUTIONS, INTRODUCTION, MANUALS, MANUAL PAGE ORGANIZATION, GENERIC TERMS AND ABBREVIATIONS, Section 1.3, 1.6, 2.2.2, 2.2.3, 2.3, 2.3.3, 2.3.4, 2.4, CHAPTER 3, Section 3.3, CHAPTER 4, Section 4.1.2, 4.2.2, 4.2.3, 5.1.1, 5.1.3, 5.1.6, 5.1.7, 5.1.8, 5.1.10, 6.1, 6.3, 6.4, 6.5, 6.6.1, 6.6.5, 6.11.1, 6.11.3, 6.11.4, 6.12.1, 6.13.3, 6.14, 6.15, 6.15.1, 6.15.2, 6.16, 6.17, 6.18, 6.20, 6.28, 6.30, 7.1.2, 8.2, 8.3, 9.2, 9.2.5, 9.2.11, 9.7.4, 9.11, 9.14.2, 11.5, Appendix 1, Appendix 2, Appendix 3.1, Appendix 3.3.2, Appendix 4 Addition Appendix 3.4.2
Jul., 2009	SH(NA)-080807ENG-C	Revision because of function addition to the Universal model QCPU (first five digits of the serial number is "11043" or later) Partial correction Section 5.1.1, 5.1.5, 5.3.3, 6.1, 6.18, 6.21.2, 6.28, 8.1, 9.2.10, 9.6.1, 10.1.3, 12.2, Appendix 1, Appendix 2, Appendix 3.1.2, Appendix 3.2 Addition Section 6.31, 6.32
Nov., 2009	SH(NA)-080807ENG-D	Partial correction SAFETY PRECAUTIONS, Section 4.2.2, 9.6.1, 9.7.4, Appendix 3.1.1, Appendix 3.1.2 Addition CONDITIONS OF USE FOR THE PRODUCT

Japanese manual version SH-080802-F

This manual 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 manual.

© 2008 MITSUBISHI ELECTRIC CORPORATION

*The manual number is given on the bottom left of the back cover.

Print date	Manual number	Revision
Apr., 2010	SH(NA)-080807ENG-E	Revision because of function addition to the Universal model QCPU (first five digits of the serial number is "12012" or later) Model addition Q50UDEHCPU, Q100UDEHCPU Partial correction INTRODUCTION, MANUALS, GENERIC TERMS AND ABBREVIATIONS, Section 1.5, 2.1, 2.2.1, 2.2.3, 2.3.4, 2.4, 3.1, 3.5, 3.7, 3.8, 3.8.1, 3.8.2, 4.2.1, 5.1.1, 5.1.3, 5.3.3, 5.3.4, 6.3, 6.5, 6.6, 6.6.1, 6.6.2, 6.6.3, 6.6.4, 6.9, 6.11.2, 6.11.3, 6.12.2, 6.12.3, 6.13.1, 6.14, 6.15, 6.23, 6.24, 6.26, 6.29, 6.30, 6.30.1, 6.31, 7.1.4, 7.1.5, 8.1, 8.2, 9.1, 9.2, 9.2.1, 9.2.5, 9.2.7, 9.2.8, 9.2.12, 9.2.13, 9.2.14, 9.3.1, 9.4, 9.5.1, 9.6.1, 9.6.2, 9.7, 9.7.2, 9.7.4, 9.8, 9.9, 9.10, 9.12.1, 9.14.2, 10.1.1, 10.1.2, 10.1.3, 11.1, 11.3, 11.4, 11.5, Appendix 1, Appendix 2, Appendix 3.1.1, Appendix 3.1.2, Appendix 3.3.1, Appendix 3.5.1, Appendix 3.5.2, Appendix 4 Addition Section 6.33
Aug., 2010	SH(NA)-080807ENG-F	Revision on the new functions of the Universal model QCPU whose serial number (first 5 digits) is "12052" or later Partial correction SAFETY PRECAUTIONS, Section 2.1, 2.4.4, 5.1.8, 6.1, 6.30.1, 6.30.2, 8.1, 9.1, 9.2, 9.2.1, 10.1.2, 10.1.3, Appendix 2, Appendix 4 Addition Section 6.34

Memo

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

INTRODUCTION

This manual describes the memory maps, functions, programs, I/O number assignment, and devices of the Universal model QCPU.

Before using this product, please read this manual and the relevant manuals carefully and develop familiarity with the functions and performance of the Q series programmable controller to handle the product correctly.

■ Relevant CPU module

CPU module	Model
Universal model QCPU	Q00UJCPU, Q00UCPU, Q01UCPU, Q02UCPU, Q03UD(E)CPU, Q04UD(E)HCPU, Q06UD(E)HCPU, Q10UD(E)HCPU, Q13UD(E)HCPU, Q20UD(E)HCPU, Q26UD(E)HCPU, Q50UDEHCPU, Q100UDEHCPU

Remark

This manual does not describe the specifications of the power supply modules, base units, extension cables, memory cards, and batteries.

For details, refer to the following.

 QCPU User's Manual (Hardware Design, Maintenance and Inspection)

For multiple CPU systems, refer to the following.

 QCPU User's Manual (Multiple CPU System)

CONTENTS

SAFETY PRECAUTIONS.....	A - 1
CONDITIONS OF USE FOR THE PRODUCT	A - 9
REVISIONS	A - 10
INTRODUCTION	A - 13
MANUALS	A - 20
MANUAL PAGE ORGANIZATION	A - 22
GENERIC TERMS AND ABBREVIATIONS	A - 23

CHAPTER1 OVERVIEW	1-1 to 1-15

1.1 Processing Order in the CPU Module	1 - 1
1.2 Storing and Executing Programs	1 - 2
1.3 Structured Programming	1 - 3
1.4 Devices and Instructions Useful for Programming	1 - 6
1.5 Features.....	1 - 11
1.6 Checking Serial Number and Function Version	1 - 14

CHAPTER2 SEQUENCE PROGRAMS	2-1 to 2-43

2.1 Sequence Program Overview	2 - 1
2.2 Sequence Program Configuration.....	2 - 3
2.2.1 Main routine program	2 - 4
2.2.2 Subroutine program	2 - 5
2.2.3 Interrupt program	2 - 6
2.3 Settings When Program is Divided	2 - 13
2.3.1 Initial execution type program	2 - 17
2.3.2 Scan execution type program	2 - 20
2.3.3 Stand-by type program	2 - 21
2.3.4 Fixed scan execution type program	2 - 26
2.3.5 Changing the program execution type	2 - 30
2.4 Data Used in Sequence Programs	2 - 32
2.4.1 BIN (Binary Code)	2 - 35
2.4.2 HEX (Hexadecimal).....	2 - 37
2.4.3 BCD (Binary-coded Decimal)	2 - 38
2.4.4 Real number (Floating-point data).....	2 - 39
2.4.5 Character string data.....	2 - 43

CHAPTER3 CPU MODULE OPERATION	3-1 to 3-14

3.1 Initial Processing.....	3 - 1
3.2 I/O Refresh (Refresh Processing with Input/Output Modules)	3 - 2
3.3 Program Operation	3 - 2
3.4 END Processing.....	3 - 2
3.5 Operation Processing in the RUN,STOP, or PAUSE Status	3 - 3
3.6 Operation Processing during Momentary Power Failure	3 - 5
3.7 Data Clear Processing	3 - 6

3.8	I/O Processing and Response Delay	3 - 8
3.8.1	Refresh mode	3 - 9
3.8.2	Direct mode	3 - 12

CHAPTER4	ASSIGNMENT OF BASE UNIT AND I/O NUMBER	4-1 to 4-17

4.1	Base Unit Assignment	4 - 1
4.1.1	Base mode	4 - 1
4.1.2	Base unit assignment setting	4 - 2
4.2	I/O Number Assignment	4 - 4
4.2.1	Concept of I/O number assignment	4 - 5
4.2.2	Setting I/O numbers	4 - 8
4.2.3	I/O number setting example	4 - 13
4.2.4	Checking I/O numbers	4 - 17

CHAPTER5	MEMORIES AND FILES USED FOR CPU MODULE	5-1 to 5-38

5.1	Memories Used for CPU Module	5 - 1
5.1.1	Memory composition and storable data	5 - 1
5.1.2	Program memory	5 - 5
5.1.3	Program cache memory	5 - 8
5.1.4	Standard ROM	5 - 11
5.1.5	Standard RAM	5 - 12
5.1.6	Memory card	5 - 14
5.1.7	Writing to the Flash card by GX Developer	5 - 18
5.1.8	Operating the program in the memory card (boot operation)	5 - 23
5.1.9	Details of written files	5 - 27
5.1.10	Specifying valid parameters (parameter-valid drive setting)	5 - 28
5.2	Program File Structure	5 - 30
5.3	File Operations by GX Developer and Handling Precautions	5 - 32
5.3.1	File operations	5 - 32
5.3.2	Precautions for handling files	5 - 33
5.3.3	File size	5 - 34
5.3.4	Units of file sizes	5 - 36

CHAPTER6	FUNCTIONS	6-1 to 6-173

6.1	Function List	6 - 1
6.2	Constant Scan	6 - 5
6.3	Latch Function	6 - 8
6.4	Output Mode at Operating Status Change (STOP to RUN)	6 - 11
6.5	Clock Function	6 - 14
6.6	Remote Operation	6 - 19
6.6.1	Remote RUN/STOP	6 - 19
6.6.2	Remote PAUSE	6 - 22
6.6.3	Remote RESET	6 - 24

6.6.4	Remote latch clear	6 - 26
6.6.5	Relationship between remote operation and RUN/STOP status of the CPU module	6 - 27
6.7	Q Series-compatible Module Input Response Time Selection (I/O Response Time)	6 - 28
6.8	Error Time Output Mode Setting	6 - 30
6.9	H/W Error Time PLC Operation Mode Setting	6 - 31
6.10	Intelligent Function Module Switch Setting	6 - 32
6.11	Monitor Function	6 - 34
6.11.1	Monitor condition setting	6 - 35
6.11.2	Local device monitor/test	6 - 41
6.11.3	External input/output forced on/off	6 - 44
6.11.4	Executorial conditioned device test	6 - 49
6.12	Writing Programs While CPU Module is in RUN Status	6 - 59
6.12.1	Online change (ladder mode)	6 - 59
6.12.2	Online change (files)	6 - 62
6.12.3	Precautions for online change	6 - 64
6.13	Execution Time Measurement	6 - 70
6.13.1	Program monitor list	6 - 70
6.13.2	Interrupt program monitor list	6 - 73
6.13.3	Scan time measurement	6 - 74
6.14	Sampling Trace Function	6 - 78
6.15	Debug Function from Multiple GX Developers	6 - 93
6.15.1	Simultaneous monitoring from multiple GX Developers function	6 - 94
6.15.2	Online change function from multiple GX Developers	6 - 96
6.16	Watchdog Timer (WDT)	6 - 98
6.17	Self-diagnostic Function	6 - 100
6.17.1	LEDs indicating errors	6 - 107
6.17.2	Error clear	6 - 107
6.18	Error History	6 - 108
6.19	System Protection	6 - 109
6.19.1	Password registration	6 - 109
6.19.2	Remote password	6 - 111
6.20	System Display of CPU Module with GX Developer	6 - 116
6.21	LED Indication	6 - 119
6.21.1	Methods for turning off the LEDs	6 - 119
6.21.2	LED indication priority	6 - 120
6.22	Interrupt from Intelligent Function Module	6 - 122
6.23	Serial Communication Function	6 - 123
6.24	Service Processing	6 - 131
6.24.1	Service processing setting	6 - 131
6.25	Initial Device Value	6 - 138
6.26	Battery Life-prolonging Function	6 - 142
6.27	Memory Check Function	6 - 143
6.28	Latch Data Backup to Standard ROM Function	6 - 144
6.29	Writing/Reading Device Data to/from Standard ROM	6 - 149
6.30	CPU Module Change Function with Memory Card	6 - 150
6.30.1	Backup function to memory card	6 - 153

6.30.2	Backup data restoration function.....	6 - 161
6.31	Module model name read	6 - 165
6.32	Module error collection.....	6 - 166
6.33	Local Device Batch Read Function.....	6 - 170
6.34	Send Points Expansion Function (CC-Link IE Controller Network Module).....	6 - 172

CHAPTER7 COMMUNICATIONS WITH INTELLIGENT FUNCTION MODULE	7-1 to 7-9

7.1	Communications between CPU Module and Intelligent Function Module	7 - 1
7.1.1	Initial setting and auto refresh setting by GX Configurator	7 - 2
7.1.2	Initial setting by initial device value	7 - 5
7.1.3	Communications with the FROM and TO instructions	7 - 5
7.1.4	Communications using the intelligent function module device	7 - 6
7.1.5	Communications using the intelligent function module dedicated instruction	7 - 8

CHAPTER8 PARAMETERS	8-1 to 8-27

8.1	PLC Parameters	8 - 2
8.2	Network Parameters	8 - 21
8.3	Remote Password.....	8 - 27

CHAPTER9 DEVICES	9-1 to 9-96

9.1	Device List	9 - 1
9.2	Internal User Device	9 - 3
9.2.1	Input (X)	9 - 6
9.2.2	Output (Y).....	9 - 8
9.2.3	Internal relay (M)	9 - 9
9.2.4	Latch relay (L)	9 - 10
9.2.5	Annunciator (F)	9 - 11
9.2.6	Edge relay (V)	9 - 15
9.2.7	Link relay (B).....	9 - 16
9.2.8	Link special relay (SB)	9 - 18
9.2.9	Step relay (S)	9 - 19
9.2.10	Timer (T).....	9 - 20
9.2.11	Counter (C)	9 - 27
9.2.12	Data register (D).....	9 - 31
9.2.13	Link register (W).....	9 - 32
9.2.14	Link special register (SW)	9 - 34
9.3	Internal System Devices	9 - 35
9.3.1	Function devices (FX, FY, FD)	9 - 35
9.3.2	Special relay (SM)	9 - 38
9.3.3	Special register (SD)	9 - 39
9.4	Link Direct Device (JL).....	9 - 40
9.5	Module Access Devices	9 - 44

9.5.1	Intelligent function module device (U\G)	9 - 44
9.5.2	Cyclic transmission area device (U3En\G)	9 - 46
9.6	Index Register (Z)/Standard Device Resister (Z)	9 - 47
9.6.1	Index register (Z)	9 - 47
9.6.2	Standard device register (Z)	9 - 49
9.6.3	Switching from the scan execution type program to the interrupt/fixed scan execution type program	9 - 50
9.7	File Register (R)	9 - 54
9.7.1	File register data storage location	9 - 55
9.7.2	File register size	9 - 55
9.7.3	Differences in available accesses by storage memory	9 - 56
9.7.4	Registration procedure for the file register	9 - 57
9.7.5	Specification methods of the file register	9 - 62
9.7.6	Precautions for using the file register	9 - 63
9.8	Extended Data Register (D) and Extended Link Register (W)	9 - 65
9.9	Nesting (N)	9 - 71
9.10	Pointer (P)	9 - 72
9.10.1	Local pointer	9 - 74
9.10.2	Common pointer	9 - 76
9.11	Interrupt Pointer(I)	9 - 78
9.11.1	List of interrupt pointer numbers and interrupt factors	9 - 79
9.12	Other Devices	9 - 80
9.12.1	SFC block device (BL)	9 - 80
9.12.2	Network No. specification device (J)	9 - 80
9.12.3	I/O No. specification device (U)	9 - 81
9.12.4	Macro instruction argument device (VD)	9 - 82
9.13	Constants	9 - 83
9.13.1	Decimal constant (K)	9 - 83
9.13.2	Hexadecimal constant (H)	9 - 83
9.13.3	Real number (E)	9 - 84
9.13.4	Character string (" ")	9 - 85
9.14	Convenient Usage of Devices	9 - 86
9.14.1	Global device	9 - 86
9.14.2	Local device	9 - 87

CHAPTER10 CPU MODULE PROCESSING TIME

10-1 to 10-19

10.1	Scan Time	10 - 1
10.1.1	Scan time structure	10 - 1
10.1.2	Time required for each processing included in scan time	10 - 2
10.1.3	Factors that increase the scan time	10 - 11

CHAPTER11 PROCEDURES FOR WRITING PROGRAM TO CPU MODULE

11-1 to 11-10

11.1	Items to be Considered for Creating Programs	11 - 1

11.2 Hardware Check	11 - 3
11.3 Procedure for Writing One Program	11 - 5
11.4 Procedure for Writing Multiple Programs	11 - 7
11.5 Procedure for Boot Operation	11 - 10

CHAPTER12 SPECIAL RELAY LIST AND SPECIAL REGISTER LIST	12-1 to 12-126

12.1 Special Relay List.....	12 - 1
12.2 Special Register List	12 - 47

APPENDICES	App-1 to App-72

Appendix 1 List of Parameter Numbers	App- 1
Appendix 2 Functions Added or Changed by Version Upgrade.....	App- 6
Appendix 3 Method of Replacing Basic Model QCPU or High Performance Model QCPU with Universal Model QCPU	App- 8
Appendix 3.1 Replacement Precautions	App- 8
Appendix 3.1.1 Replacing Basic model QCPU with Universal model QCPU	App- 8
Appendix 3.1.2 Replacing High Performance model QCPU with Universal model QCPU	App- 12
Appendix 3.2 Applicable devices and software	App- 19
Appendix 3.3 Instructions	App- 23
Appendix 3.3.1 Instructions not supported in the Universal model QCPU and replacing methods.....	App- 23
Appendix 3.3.2 Replacing programs using multiple CPU transmission dedicated instructions	App- 25
Appendix 3.3.3 Program replacement examples	App- 26
Appendix 3.4 Functions.....	App- 40
Appendix 3.4.1 Floating-point operation instructions	App- 40
Appendix 3.4.2 Error check processing for floating-point data comparison instructions	App- 48
Appendix 3.4.3 Range check processing for index-modified devices	App- 52
Appendix 3.4.4 Device latch function	App- 55
Appendix 3.4.5 File usability setting.....	App- 57
Appendix 3.4.6 Parameter-valid drive and boot file setting	App- 60
Appendix 3.4.7 External input/output forced on/off function	App- 63
Appendix 3.5 Special Relay and Special Register.....	App- 67
Appendix 3.5.1 Special relay list	App- 67
Appendix 3.5.2 Special register list.....	App- 70
Appendix 4 Device Point Assignment Sheet.....	App- 72

INDEX	Index-1 to Index-5

MANUALS

To understand the main specifications, functions, and usage of the CPU module, refer to the basic manuals.
Read other manuals as well when using a different type of CPU module and its functions.
Order each manual as needed, referring to the following list.

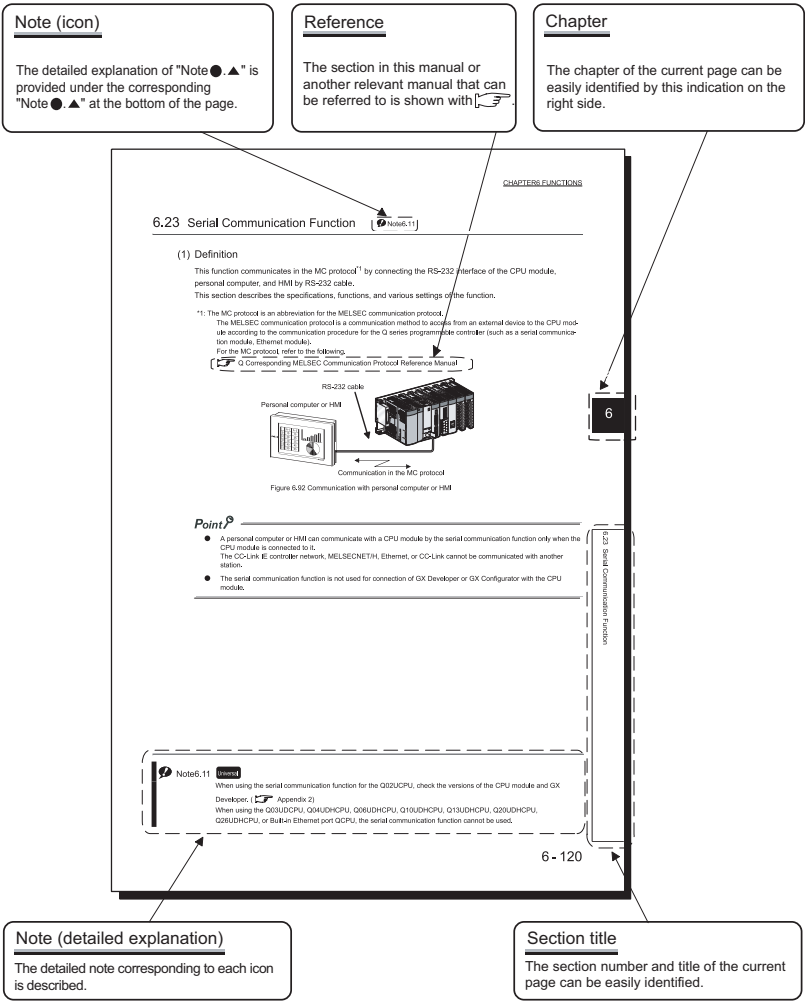
● :Basic manual, ○ :Other CPU module manuals

Manual name < Manual number (model code) >	Description	Manual type
● User's manual		
QCPU User's Manual (Hardware Design, Maintenance and Inspection) < SH-080483ENG (13JR73) >	Specifications of the hardware (CPU modules, power supply modules, base units, extension cables, and memory cards), system maintenance and inspection, troubleshooting, and error codes	●
QnUCPU User's Manual (Function Explanation, Program Fundamentals) < SH-080807ENG (13JZ27) >	Functions, methods, and devices for programming	●
QCPU User's Manual (Multiple CPU System) < SH-080485ENG (13JR75) >	Information on multiple CPU system configuration (system configuration, I/O numbers, communication between CPU modules, and communication with the input/output modules and intelligent function modules)	○
QnUCPU User's Manual (Communication via Built-in Ethernet Port) < SH-080811ENG (13JZ29) >	Functions for the communication via built-in Ethernet port of the CPU module	○
● Programming manual		
MELSEC-Q/L Programming Manual (Common Instruction) < SH-080809ENG (13JW10) >	How to use sequence instructions, basic instructions, and application instructions	●
MELSEC-Q/L/QnA Programming Manual (SFC) < SH-080041 (13JF60) >	System configuration, performance specifications, functions, programming, debugging, and error codes for SFC (MELSAP3) programs	○
MELSEC-Q/L Programming Manual (MELSAP-L) < SH-080076 (13JF61) >	Programming methods, specifications, and functions for SFC (MELSAP-L) programs	○
MELSEC-Q/L Programming Manual (Structured Text) < SH-080366E (13JF68) >	Programming methods using structured languages	○
MELSEC-Q/L/QnA Programming Manual (PID Control Instructions) < SH-080040 (13JF59) >	Dedicated instructions for PID control	○

Other relevant manuals

Manual name	Description
CC-Link IE Controller Network Reference Manual < SH-080668ENG (13JV16) >	Specifications, procedures and settings before system operation, parameter setting, programming, and troubleshooting of the CC-Link IE controller network module
MELSEC-Q CC-Link IE Field Network Master/Local Module User's Manual < SH-080917ENG (13JZ47) >	Specifications, procedures and settings before system operation, parameter setting, programming, and troubleshooting of the CC-Link IE field network module
Q Corresponding MELSECNET/H Network System Reference Manual (PLC to PLC network) < SH-080049 (13JF92) >	Specifications, procedures and settings before system operation, parameter setting, programming, and troubleshooting of a MELSECNET/H network system (PLC to PLC network)
Q Corresponding MELSECNET/H Network System Reference Manual (Remote I/O network) < SH-080124 (13JF96) >	Specifications, procedures and settings before system operation, parameter setting, programming, and troubleshooting of a MELSECNET/H network system (remote I/O network)
Q Corresponding Ethernet Interface Module User's Manual (Basic) < SH-080009 (13JL88) >	Specifications, procedures for data communication with external devices, line connection (open/close), fixed buffer communication, random access buffer communication, and troubleshooting of the Ethernet module
Q Corresponding Ethernet Interface Module User's Manual (Application) < SH-080010 (13JL89) >	E-mail function, programmable controller CPU status monitoring function, communication via MELSECNET/H or MELSECNET/10, communication using the data link instructions, and file transfer function (FTP server) of the Ethernet module
CC-Link System Master/Local Module User's Manual < SH-080394E (13JR64) >	System configuration, performance specifications, functions, handling, wiring, and troubleshooting of the QJ61BT11N
Q Corresponding Serial Communication Module User's Manual (Basic) < SH-080006 (13JL86) >	Overview, system configuration, specifications, procedures before operation, basic data communication method with external devices, maintenance and inspection, and troubleshooting for using the serial communication module
MELSEC-Q/L Serial Communication Module User's Manual (Application) < SH-080007 (13JL87) >	Special functions (specifications, usage, and settings and data communication method with external devices of the serial communication module
MELSEC-Q/L MELSEC Communication Protocol Reference Manual < SH-080008 (13JF89) >	Communication method using the MC protocol, which reads/writes data to/from the CPU module via the serial communication module or Ethernet module
GX Developer Version 8 Operating Manual < SH-080373E (13JU41) >	Operating methods of GX Developer, such as programming and printout
GX Works2 Version1 Operating Manual (Common) < SH-080779ENG (13JU63) >	Operating methods of GX Works2, such as programming and printout

MANUAL PAGE ORGANIZATION



* cining page components, and differs from an actual page.

Icons	Description
Universal model QCPU	
Universal	Icons indicate that specifications described on the page contain some precautions.

In addition, this manual uses the following types of explanations.

Point

In addition to description of the page, notes or functions that require special attention are described here.

Remark

The reference related to the page or useful information are described here.

GENERIC TERMS AND ABBREVIATIONS

Unless otherwise specified, this manual uses the following generic terms and abbreviations.

*□ indicates a part of the model or version.

(Example): Q33B, Q35B, Q38B, Q312B → Q3□B

Generic term/abbreviation	Description
■ Series	
Q series	Abbreviation for Mitsubishi MELSEC-Q series programmable controller
■ CPU module type	
CPU module	Generic term for the Universal model QCPU
Universal model QCPU	Generic term for the Q00UJCPU, Q00UCPU, Q01UCPU, Q02UCPU, Q03UDHCPU, Q04UDHCPU, Q06UDHCPU, Q10UDHCPU, Q13UDHCPU, Q20UDHCPU, Q26UDHCPU, Q03UDECPU, Q04UDEHCPU, Q06UDEHCPU, Q10UDEHCPU, Q13UDEHCPU, Q20UDEHCPU, Q26UDEHCPU, Q50UDEHCPU and Q100UDEHCPU
Built-in Ethernet port QCPU	Generic term for the Q03UDECPU, Q04UDEHCPU, Q06UDEHCPU, Q10UDEHCPU, Q13UDEHCPU, Q20UDEHCPU, Q26UDEHCPU, Q50UDEHCPU and Q100UDEHCPU
Motion CPU	Generic term for Mitsubishi motion controllers, Q172CPUN, Q173CPUN, Q172HCPU, Q173HCPU, Q172CPUN-T, Q173CPUN-T, Q172HCPU-T, Q173HCPU-T, Q172DCPU, and Q173DCPU
PC CPU module	Generic term for MELSEC-Q series PC CPU module, PPC-CPU852(MS)-512, manufactured by CONTEC Co., Ltd.
C Controller module	Generic term for the Q06CCPU-V-H01, Q06CCPU-V, Q06CCPU-V-B, and Q12DCCPU-V C Controller modules
■ CPU module model	
QnUD(H)CPU	Generic term for the Q03UDHCPU, Q04UDHCPU, Q06UDHCPU, Q10UDHCPU, Q13UDHCPU, Q20UDHCPU, and Q26UDHCPU
■ Base unit type	
Base unit	Generic term for the main base unit, extension base unit, slim type main base unit, redundant power main base unit, redundant power extension base unit, and multiple CPU high speed main base unit
Main base unit	Generic term for the Q3□B, Q3□SB, Q3□RB, and Q3□DB
Extension base unit	Generic term for the Q5□B, Q6□B, and Q6□RB
Slim type main base unit	Another name for the Q3□SB
Redundant power main base unit	Another name for the Q3□RB
Redundant power extension base unit	Another name for the Q6□RB
Multiple CPU high speed main base unit	Another name for the Q3□DB
■ Base unit model	
Q3□B	Generic term for the Q33B, Q35B, Q38B, and Q312B main base units
Q3□SB	Generic term for the Q32SB, Q33SB, and Q35SB slim type main base units
Q3□RB	Another name for the Q38RB main base unit for redundant power supply system
Q3□DB	Generic term for the Q38DB and Q312DB multiple CPU high speed main base units
Q5□B	Generic term for the Q52B and Q55B extension base units
Q6□B	Generic term for the Q63B, Q65B, Q68B, and Q612B extension base units

Generic term/abbreviation	Description
Q6□RB	Another name for the Q68RB extension base unit for redundant power supply system
■ Power supply module	
Power supply module	Generic term for the Q series power supply module, slim type power supply module, and redundant power supply module
Q series power supply module	Generic term for the Q61P-A1, Q61P-A2, Q61P, Q61P-D, Q62P, Q63P, Q64P, and Q64PN power supply modules
Slim type power supply module	Abbreviation for the Q61SP slim type power supply module
Redundant power supply module	Generic term for the Q63RP and Q64RP power supply modules for redundant power supply
■ Network module	
CC-Link IE module	Generic term for the CC-Link IE controller network and CC-Link IE field network
MELSECNET/H module	Abbreviation for the MELSECNET/H network module
Ethernet module	Abbreviation for the Ethernet network module
CC-Link module	Abbreviation for the CC-Link system master/local module
■ Network	
CC-Link IE	Generic term for the CC-Link IE controller network and CC-Link IE field network
MELSECNET/H	Abbreviation for the MELSECNET/H network system
Ethernet	Abbreviation for the Ethernet network system
CC-Link	Abbreviation for the Control & Communication Link
■ Memory card	
Memory card	Generic term for the SRAM card, Flash card, and ATA card
SRAM card	Generic term for the Q2MEM-1MBS, Q2MEM-2MBS, Q3MEM-4MBS, and Q3MEM-8MBS SRAM cards
Flash card	Generic term for the Q2MEM-2MBF and Q2MEM-4MBF Flash cards
ATA card	Generic term for the Q2MEM-8MBA, Q2MEM-16MBA, and Q2MEM-32MBA ATA cards
■ Others	
GX Developer	Product name for MELSEC PLC software package
GX Works2	
Extension cable	Generic term for the QC05B, QC06B, QC12B, QC30B, QC50B, and QC100B extension cables
Battery	Generic term for the Q6BAT, Q7BAT, and Q8BAT CPU module batteries, Q2MEM-BAT SRAM card battery, and Q3MEM-BAT SRAM card battery
GOT	Generic term for Mitsubishi Graphic Operation Terminal, GOT-A*** series, GOT-F*** series, and GOT1000 series

CHAPTER1 OVERVIEW

The CPU module performs sequence control by executing programs.

This chapter describes the processing order in the CPU module, locations where the created programs are stored, and devices and instructions useful for programming.

1.1 Processing Order in the CPU Module

The CPU module performs processing in the following order.

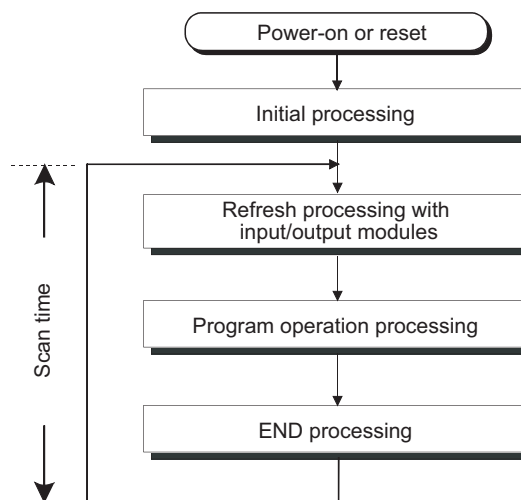


Figure 1.1 Processing order in the CPU module

(1) Initial processing (👉 Section 3.1)

The CPU module performs preprocessing required for program operations.

The preprocessing is performed only once when the module is powered on or reset.

(2) Refresh processing with input and output modules (👉 Section 3.2)

The CPU module takes on/off data from the input module or intelligent function module and outputs on/off data to the output module or intelligent function module.

(3) Program operation processing (👉 Section 3.3)

The CPU module sequentially executes the program stored in the module from the step 0 to the END or FEND instruction.

(4) END processing (👉 Section 3.4)

The CPU module performs refresh processing with network modules or communicates with external devices.

1.2 Storing and Executing Programs

This section describes where to store and how to execute the programs in the CPU module.

(1) Programming

Programs are created with GX Developer.

For details of program configuration and execution conditions, refer to CHAPTER 2.

(2) Storing programs

Created programs and set parameters are stored in the following memories of the CPU module.

(☞ Section 5.1)

- Program memory
- Standard ROM (parameters only)
- Memory card

(3) Executing programs

The CPU module executes the programs stored in the program memory.

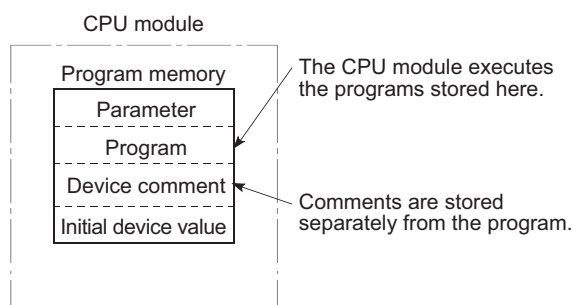


Figure 1.2 Executing programs

To execute the programs stored in a memory card, the programs need to be booted to the program memory

(☞ Section 5.1.8) when the CPU module is powered off and then on or reset.

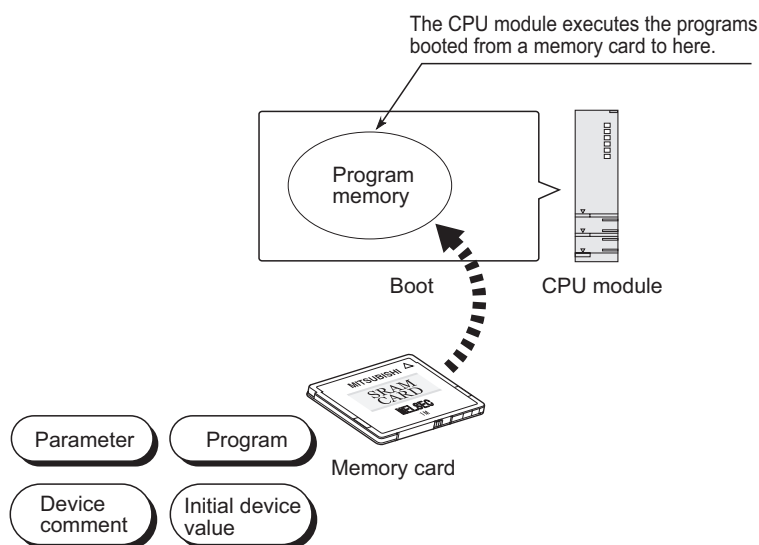


Figure 1.3 Executing programs stored in a memory card

1.3 Structured Programming

The programs to be executed in the CPU module can be structured in the following two ways.

- In one program
- By dividing into multiple files

(1) Structuring in one program

Structured programming is available by creating one program as a collection of three program sections: main routine program (☞ Section 2.2.1), subroutine program (☞ Section 2.2.2), and interrupt program (☞ Section 2.2.3).

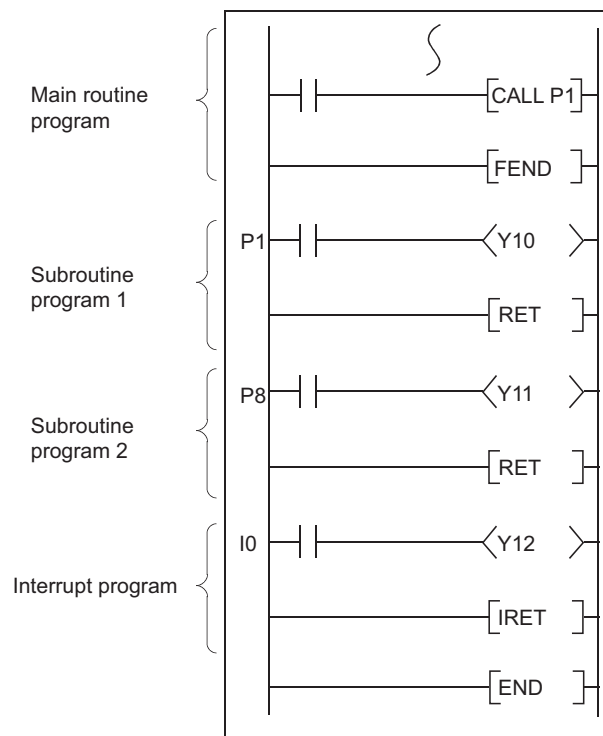


Figure 1.4 Structuring in one program

(2) Structuring by dividing into multiple files

A program is stored in a file.

Changing the file name allows the CPU module to store multiple programs.

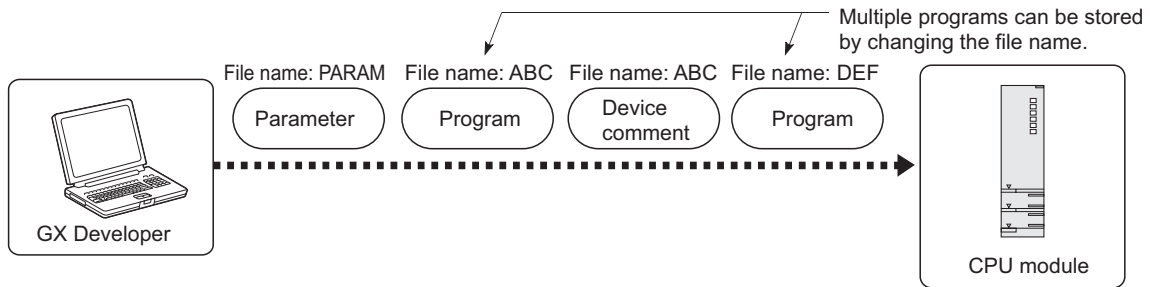


Figure 1.5 Structuring by dividing into multiple files

Dividing into multiple files according to the processes or functions enables simultaneous programming by two or more designers. Managing the files separately eases reuse and utilization to other programs.

Structured programming is efficient in this way because only the corresponding file needs to be modified or debugged in case of change in the specifications.

(a) Dividing into multiple files according to the processes^{*1}

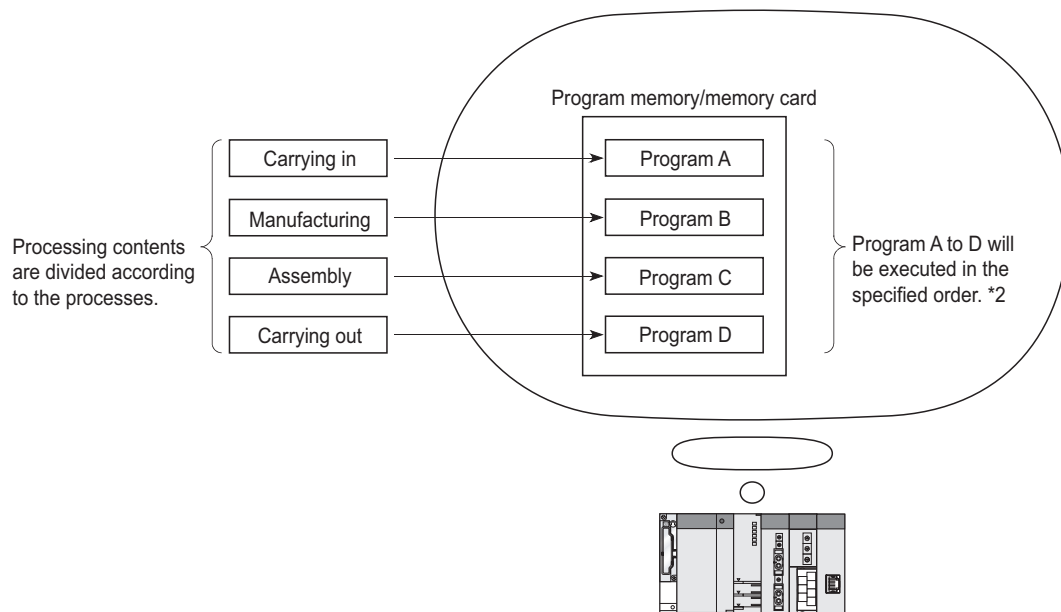
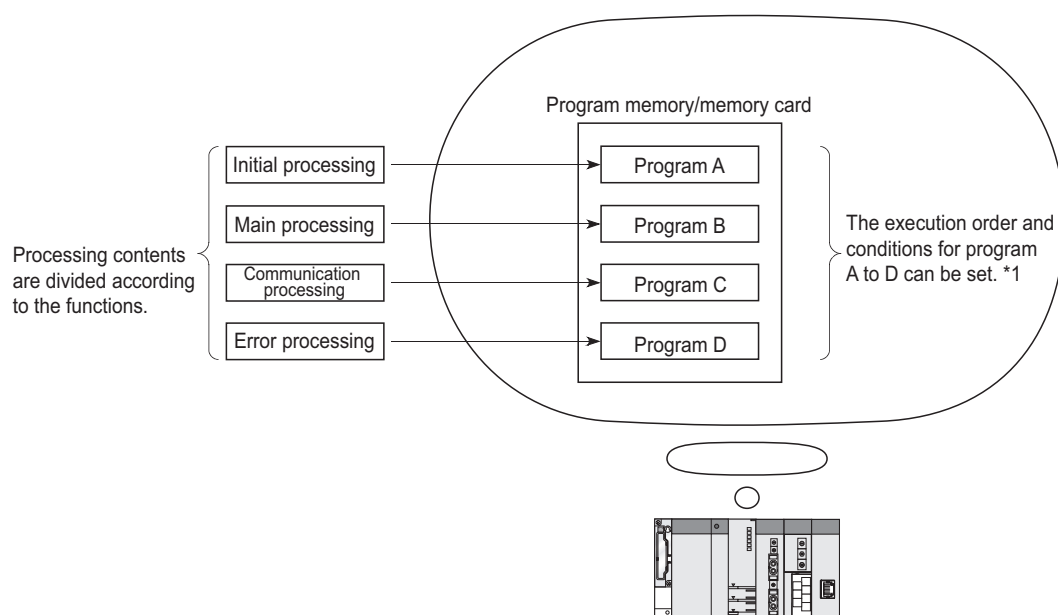


Figure 1.6 Dividing into multiple files according to the processes

^{*1}: The processing contents divided according to the processes can further be divided and managed according to the functions.

^{*2}: The execution order can be set in the Program tab of the PLC parameter dialog box. (☞ Section 2.3(2)).

(b) Dividing into multiple files according to the functions**Figure 1.7 Dividing into multiple files according to the functions**

*1: The execution order and conditions can be set in the Program tab of the PLC parameter dialog box. (☞ Section 2.3(2)).

1.4 Devices and Instructions Useful for Programming

The CPU module is provided with devices and instructions useful for programming.
This section describes the outline of these devices and instructions.

(1) Various ways of device specification

(a) Using each bit of a word device as a contact or coil

By specifying a bit of a word device, the bit can be used as a contact or coil.

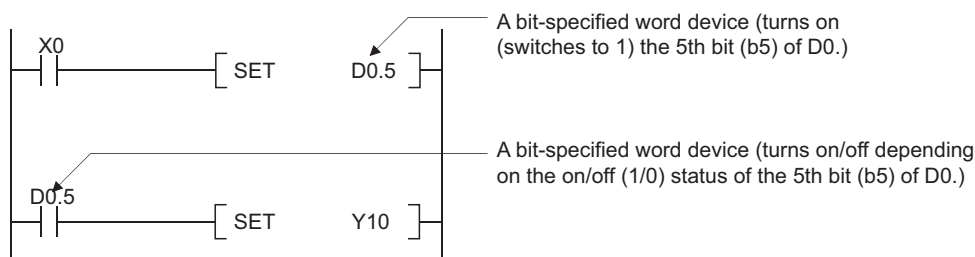


Figure 1.8 Specifying a bit of a word device

(b) Easy direct processing in units of one point

Use of the direct access input (DX ) and direct access output (DY ) enables easy direct processing (in units of one point) in the program. (➡ Section 3.8.2)

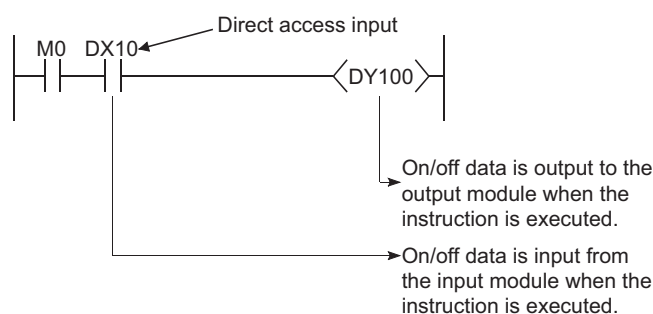


Figure 1.9 Direct processing in units of one point

(c) No input pulse conversion required by using a differential contact

Pulse conversion processing for inputs is no longer required with the use of a differential contacts ( and ).

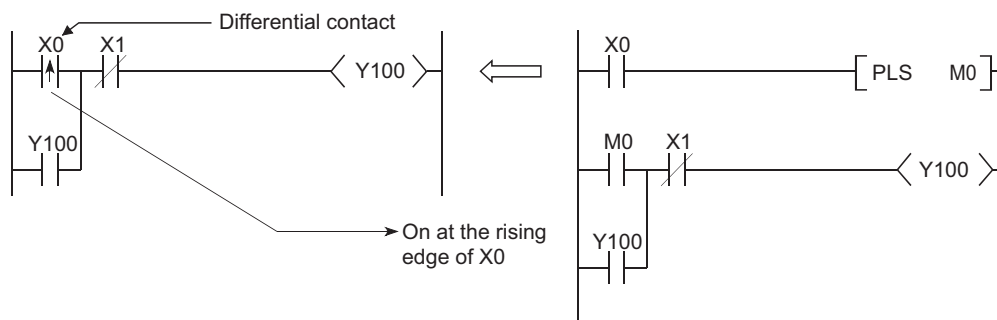


Figure 1.10 Use of a differential contact

(d) Direct access to the buffer memory of the intelligent function module

The buffer memory of the intelligent function module can be used as a device area in a program.

( Section 9.5.1)

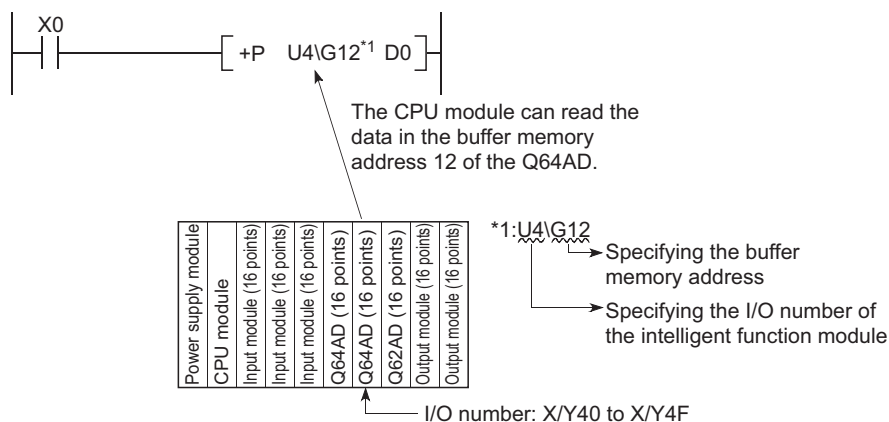


Figure 1.11 Direct access to the buffer memory of the intelligent function module

(e) Direct access to the link devices

The link devices (LX, LY, LB, LW, SB, or SW) in network modules can be directly accessed without the refresh setting. ( Section 9.4)

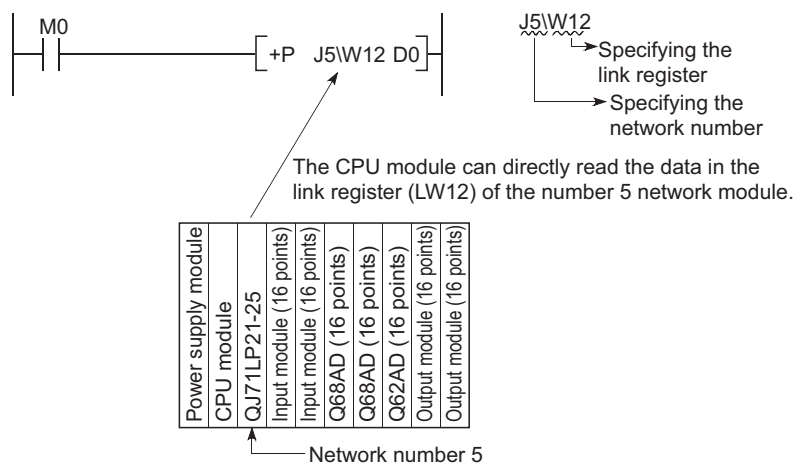


Figure 1.12 Direct access to the link devices

(2) Structural description of programs

Use of the index register and edge relay enables easy structured programming including the pulse conversion processing. (Section 9.2.6)

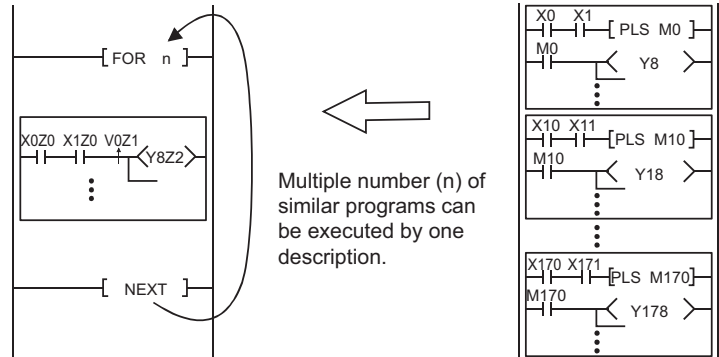


Figure 1.13 Structured programming including the pulse conversion processing

(3) Easy data processing

(a) Using real numbers and character string constants without conversion

Real numbers (floating-point data) and character string constants can be used without conversion for programming.

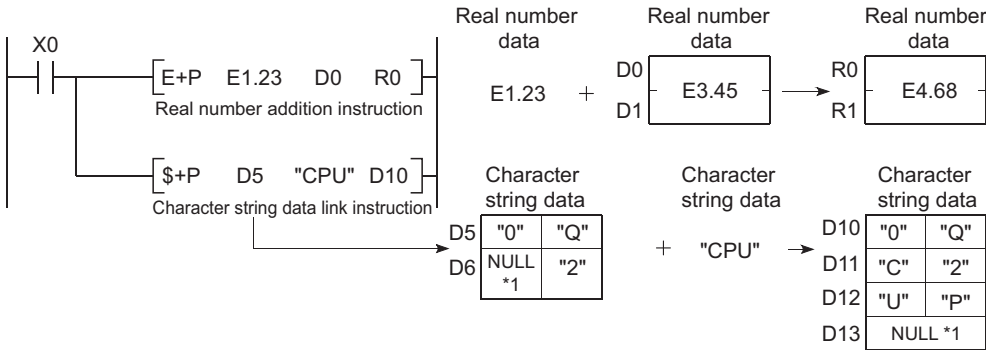


Figure 1.14 Using real numbers and character string constants

*1: The NULL character represents "00H" (end of character strings).

(b) High-speed processing of bulk data

Extension of the data table operation instructions, such as the data table processing instruction, allows high-speed processing of bulk data.

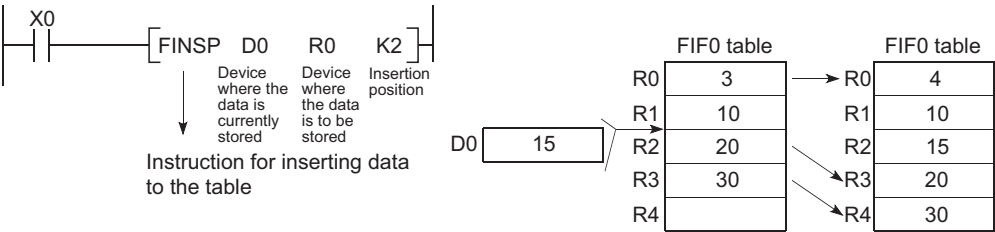


Figure 1.15 Data processing with the table processing instruction

(4) Flexible management of subroutine programs

(a) Subroutine program sharing

The number of steps in a program can be reduced by sharing subroutine programs. In addition, creating and managing programs become easier.

Subroutine programs can be created within the same program and called. Subroutine programs in other programs can also be called by using the common pointer.

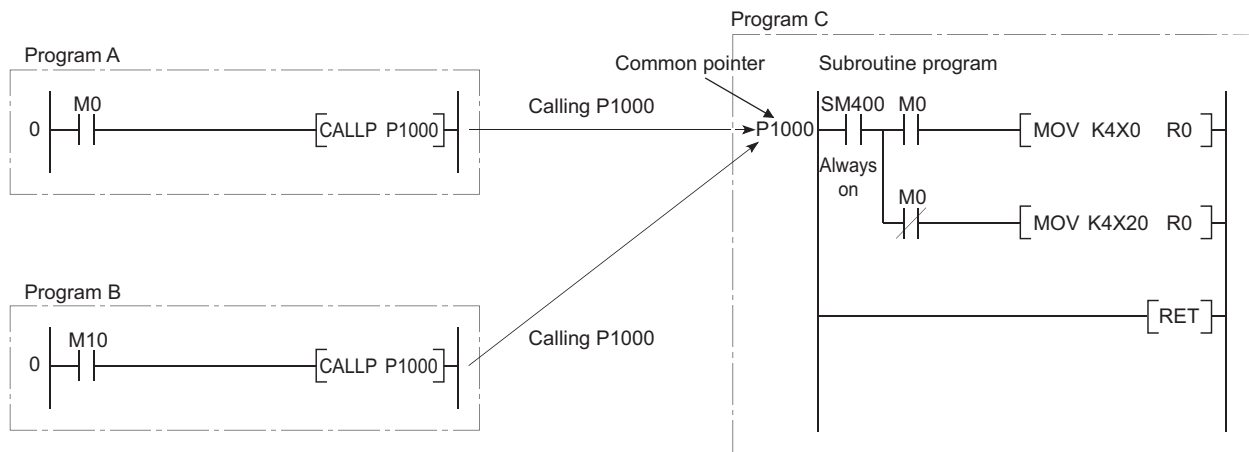


Figure 1.16 Subroutine program sharing

(b) Subroutine call instruction with argument passing

Subroutine program that is called more than one time can be created easily.

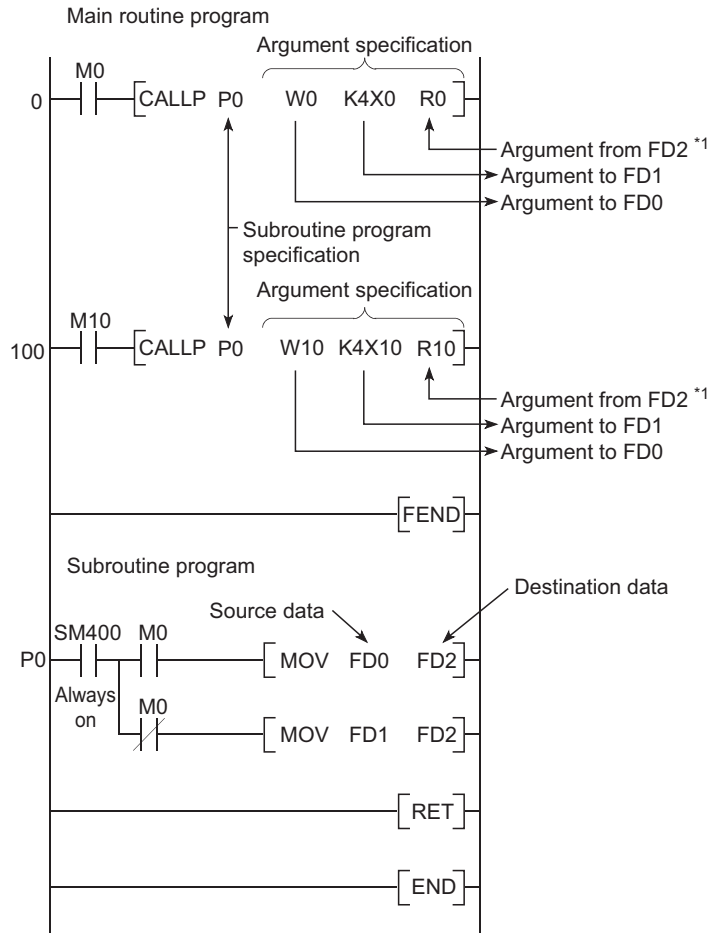


Figure 1.17 Calling subroutine program with argument passing

*1: For input and output conditions for an argument, refer to Section 9.3.1.

1.5 Features

This section describes the features specific to the Universal model QCPU.

(1) High-speed processing more than ever

The processing time required for the basic instructions, floating-point operations, and accesses to the file register becomes shorter than the existing Q series CPU modules.

Use of a standard device register (Z)^{*1} achieves high-speed processing between register operations (transfer instruction).

*1: An index register used between register operations is designated as a standard device register. (☞ Section 9.6.2)

(2) Large-capacity file register

The file register whose capacity is 896K points at maximum (4086K points at maximum when a memory card is used)^{*2} can be set inside the CPU module.

*2: For the Q100UDEHCPU only

(3) Use of double-precision floating-point operation instructions

The double-precision floating-point operation instructions (64-bit instructions) are available as well as the existing single-precision floating-point operation instructions. (☞ Section 2.4.4)

This enables more accurate analog control and positioning control.

(4) Using the file register area as the data register and link register

The file register (ZR) area can be used as an pseudo extended area^{*3} of the data register (D) and link register (W). (☞ Section 9.8)

Programming using the extended data register (D) or extended link register (W) whose capacity is 896K points at maximum (4086K points at maximum when a memory card is used)^{*4} is available in addition to the internal user device.

*3: Extended areas of the data register (D) and link register (W) are called the extended data register (D) and extended link register (W), respectively.

*4: For the Q100UDEHCPU only

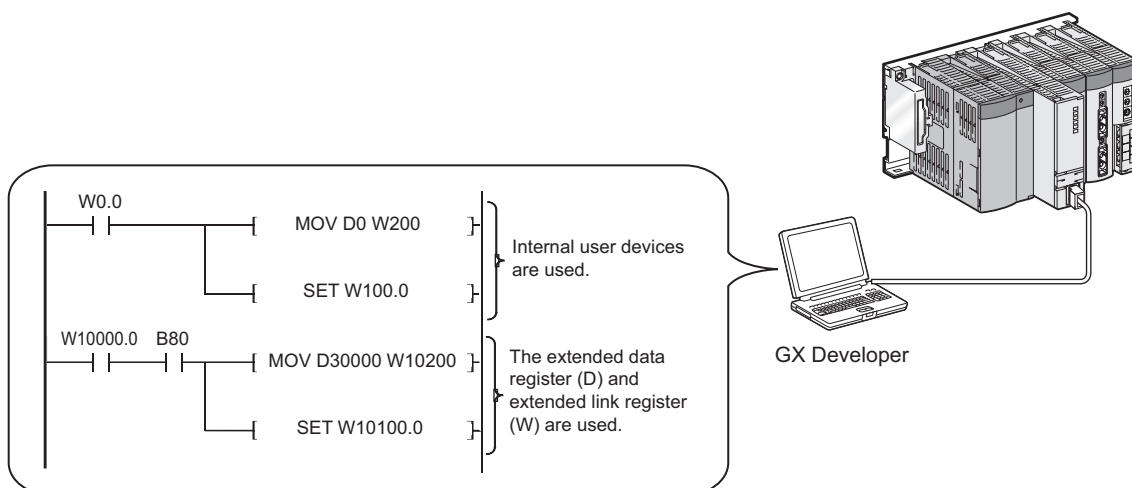


Figure 1.18 Extended data register (D) and extended link register (W)

(5) 32-bit index modification

Since the index modification range is expanded to 32 bits, index modification for the entire file register areas is possible. (☞ Section 9.6.1)

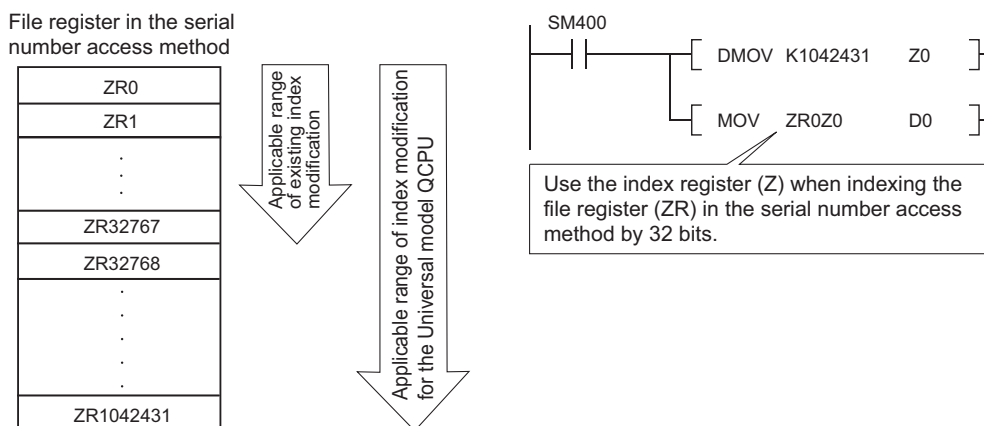


Figure 1.19 32-bit index modification

(6) Communication via the built-in Ethernet port of the CPU module

The Built-in Ethernet port QCPU can communicate with MELSOFT, GOT, and external devices using the built-in Ethernet port of the module.

☞ QnUCPU User's Manual (Communication via Built-in Ethernet Port)

(7) Communication with the personal computer and HMI by the serial communication function (☞ Section 6.23)

The Q00UJCPU, Q00UCPU, Q01UCPU, and Q02UCPU can communicate using the MELSEC communication protocol (hereafter, MC protocol) by connecting a RS-232 interface and personal computer or HMI.

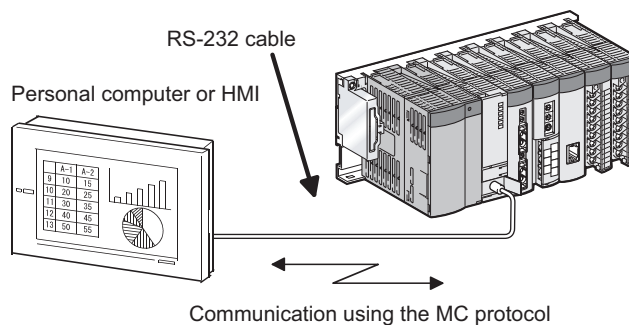


Figure 1.20 Communication with the personal computer or HMI

Point

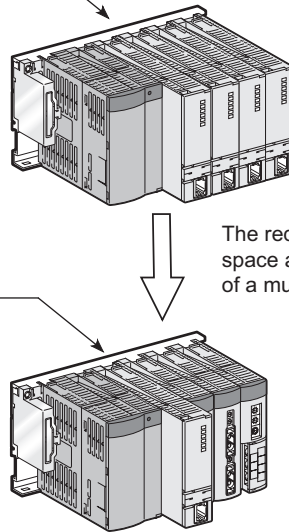
The CPU module functions are added at the update of serial number of CPU module or GX Developer version. For functions added by the update, refer to Appendix 2.

(8) Storage of up to 1000k step programs

Up to 1000k step programs can be stored in one CPU module, for the Q100UDEHCPU. Therefore, if a multiple CPU system is configured due to low program capacity, a system can be configured by one CPU module. In addition, the maintainability of the program is enhanced since data communication programs in the multiple CPU systems are not required.

Multiple CPU system configuration
(Four Q26UDEHCPUs)

Q100UDEHCPU



The reduction of system cost and saving space are available since the configuration of a multiple CPU system is not required.

Figure 1.21 The changes from a multiple CPU system configuration to a system configuration with a single CPU module

1.6 Checking Serial Number and Function Version

The serial number and function version of the CPU module can be checked on the rating plate, on the front of the module, and on the System monitor screen in GX Developer.

(1) Checking on the rating plate

The rating plate is located on the side of the CPU module.

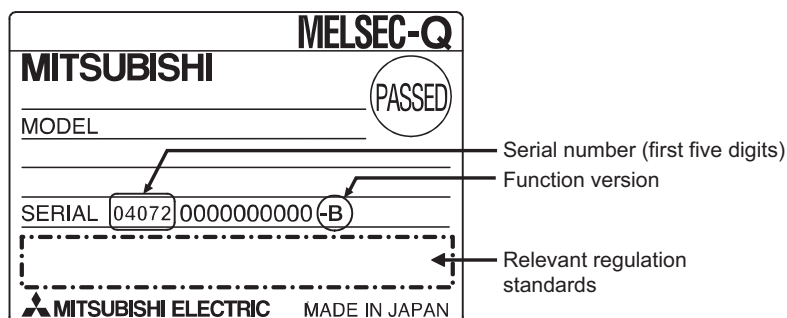


Figure 1.22 Rating plate

(2) Checking on the front of the module

The serial number on the rating plate is printed on the front (at the bottom) of the module. This applies only to the CPU module manufactured in mid-September, 2007 or later.

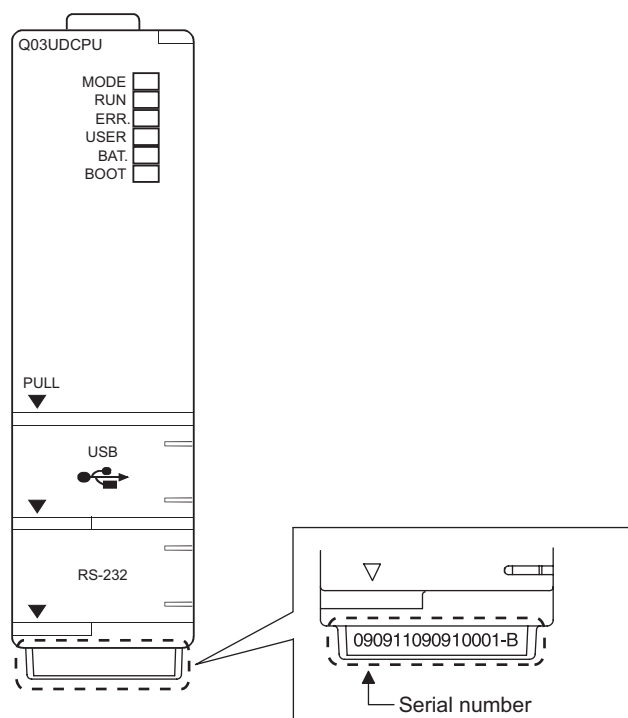


Figure 1.23 Front of the module

(3) Checking on the System monitor (Product Information List) screen

To open the screen for checking the serial number and function version, select [Diagnostics] → [System monitor] and click the  button in GX Developer.

On the same screen, the serial number and function version of intelligent function modules can also be checked.

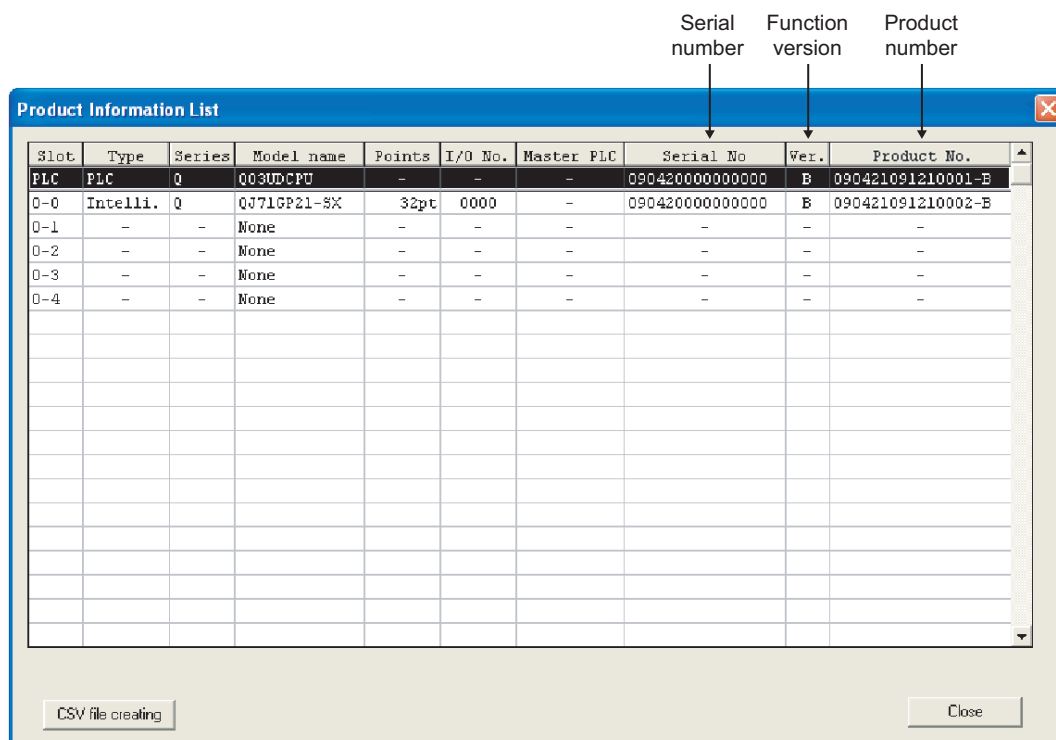


Figure 1.24 System monitor (Product Information List) screen

[Serial number, function version, and product number]

- The serial number of the module is displayed in the "Serial No." column.
- The function version of the module is displayed in the "Ver." column.
- The serial number (product number) printed on the rating plate of the module is displayed in the "Product No." column.

Note that "-" is displayed for the module that does not support the product number display.

Point

The serial number displayed on the Product Information List screen of GX Developer may differ from that on the rating plate and on the front of the module.

- The serial number on the rating plate and on the front of the module indicates the management information of the product.
- The serial number displayed on the Product Information List screen indicates the functional information of the product.

The functional information of the product will be updated when a function is added.

CHAPTER2 SEQUENCE PROGRAMS

2.1 Sequence Program Overview

(1) Definition

Sequence program is one of the programs that can be executed in the CPU module.

A sequence program consists of instructions, such as sequence instructions, basic instruction, and application instruction.

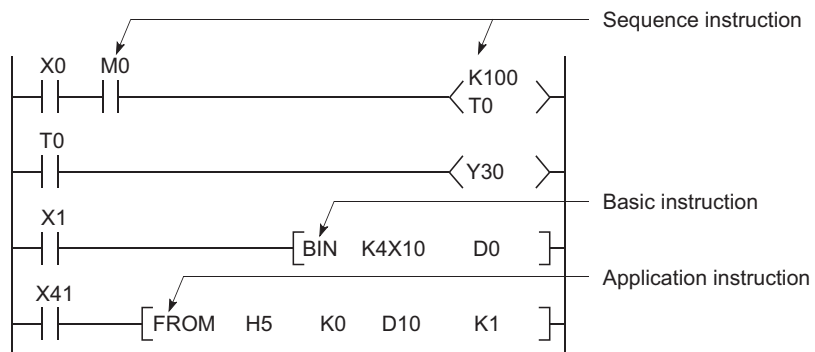


Figure 2.1 Sequence program

Point

Data indicating the execution status of an operation in a sequence program step is referred to as "signal flow".

Remark

For the instructions used in sequence programs, refer to the following.

➡ MELSEC-Q/L Programming Manual (Common Instruction)

(2) Programming method

There are two programming modes for sequence programs.

- Ladder mode
- List mode

(a) Ladder mode

Ladder mode is a mode based on the concept of sequential control by relay circuits. A program in ladder mode is similar to a schematic for a set of relay circuits. Programming in units of ladder blocks is available. A ladder block, which starts from the left rail and ends at the right rail, is the minimum unit for operating sequence programs.

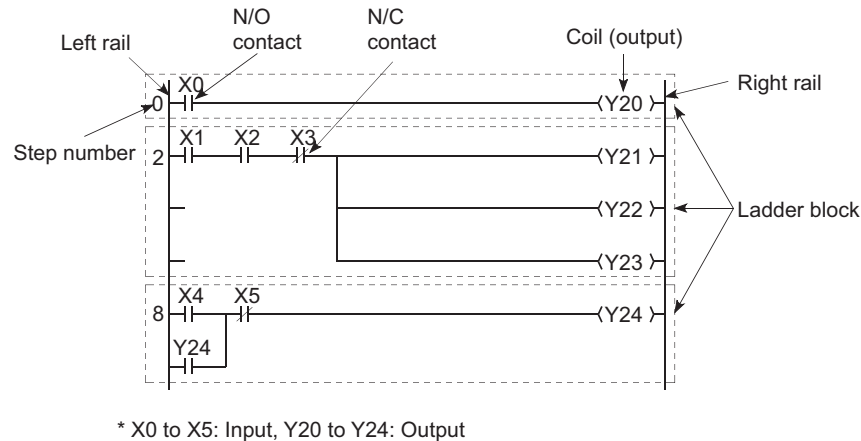


Figure 2.2 Ladder mode

(b) List mode

List mode uses dedicated instructions for programming. The symbols, such as contacts and coils, used in ladder mode are replaced with dedicated instructions. The following instructions are used for normally open contacts, normally closed contacts, and coils.

- Normally open contact ... LD, AND, OR
- Normally closed contact ... LDI, ANI, ORI
- Coil ... OUT

Remark

There are two other types of programs that can be executed in the CPU module: SFC programs and ST programs. For details, refer to the following.

- ➡ MELSEC-Q/L/QnA Programming Manual (SFC)
- ➡ MELSEC-Q/L Programming Manual (MELSAP-L)
- ➡ MELSEC-Q/L Programming Manual (Structured Text)

(3) Sequence program operation

A sequence program is sequentially operated from the step 0 to the END or FEND instruction.

In ladder mode, a sequence program is operated from left to right and top to bottom in a ladder block.

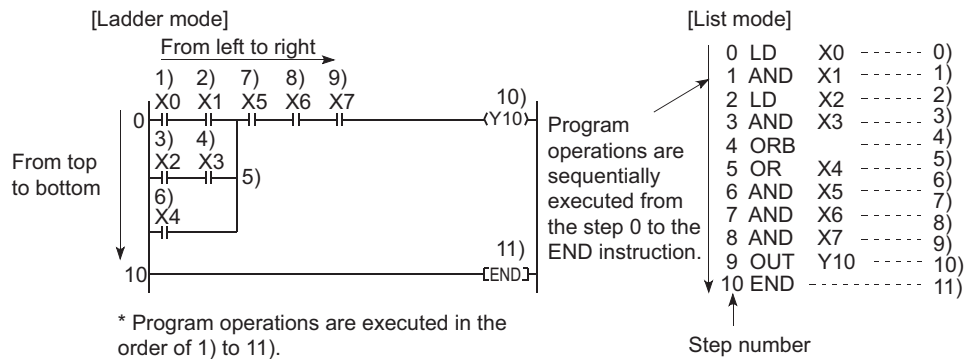


Figure 2.3 Comparison between ladder mode and list mode

2.2 Sequence Program Configuration

Sequence programs are classified into the following three types.

- Main routine program
- Subroutine program
- Interrupt program

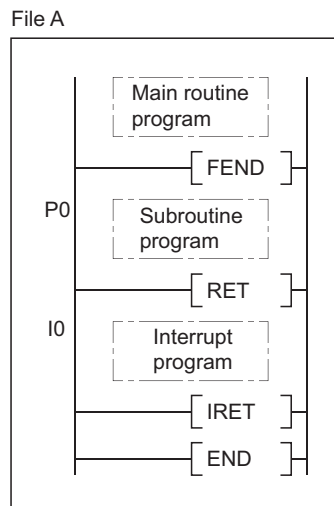


Figure 2.4 Sequence program classification

2.2.1 Main routine program

(1) Definition

Main routine program is an entire program from the step 0 to the END or FEND instruction.

(2) Program operation

A main routine program executes its operations from the step 0 to the END or FEND instruction and then performs END processing.
After the END processing, the program restarts its operations from the step 0.

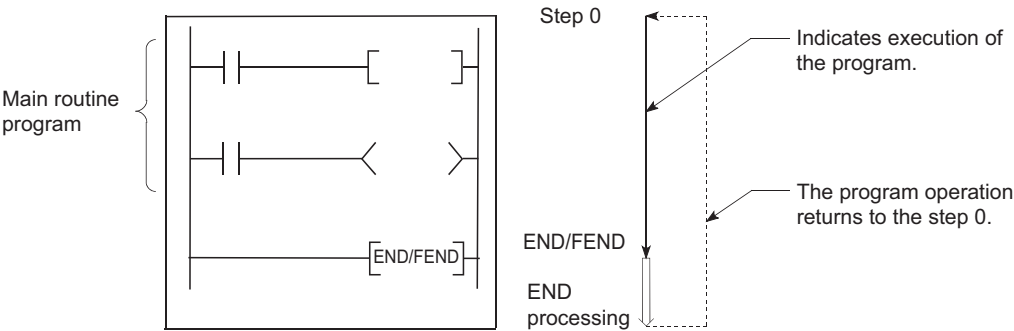


Figure 2.5 Main routine program

Point

When multiple programs are executed, the main routine program operation after execution of the END or FEND instruction varies depending on the preset execution conditions. (☞ Section 2.3.1)

Remark

For details of the END and FEND instructions, refer to the following.
☞ MELSEC-Q/L Programming Manual (Common Instruction)

2.2.2 Subroutine program

(1) Definition

Subroutine program is a program from a pointer (P□) to the RET instruction.

This program is executed only when it is called by a subroutine program call instruction (such as CALL(P), FCALL(P)) from a main routine program.

(2) Application

- Programming a program which is executed two or more times in one scan as a subroutine program can reduce the number of steps in the entire program.
- Programming a program which is executed only when a certain condition is satisfied as a subroutine program can shorten the scan time.

(3) Programming of subroutine programs

Create subroutine programs between the FEND and END instructions in the main routine program.

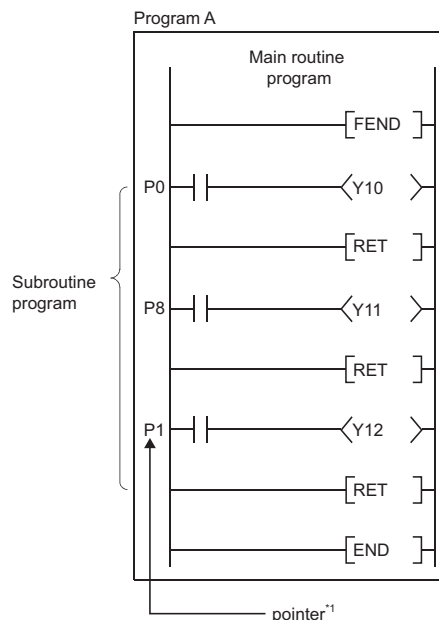


Figure 2.6 Programming location of subroutine programs

*1: The pointer numbers do not need to be specified in ascending order.

Point

Subroutine programs can be managed as one separate program (stand-by type program).

(☞ Section 2.2.3)

Remark

Subroutine programs can be configured with the nesting. (☞ Section 9.9)

2.2.3 Interrupt program

(1) Definition

Interrupt program is a program from an interrupt pointer (I□) to the IRET instruction.

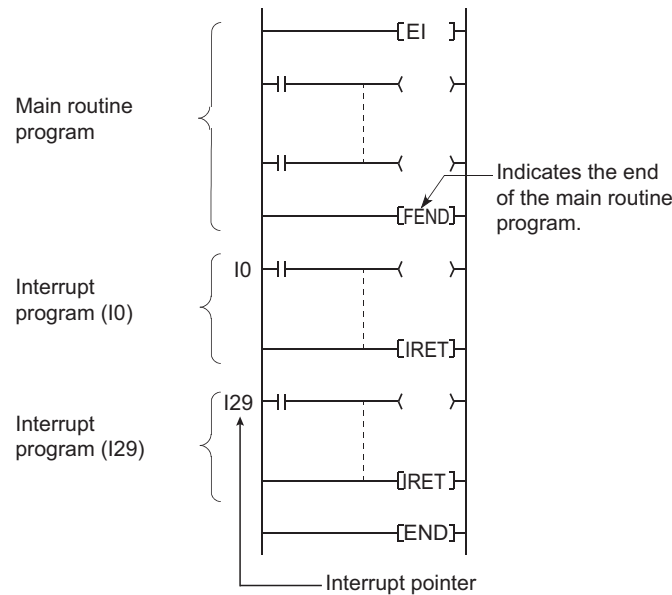


Figure 2.7 Interrupt program

The interrupt pointer (I□) number varies depending on the interrupt factor. (☞ Section 9.11)
When an interrupt factor occurs, the interrupt program of the interrupt pointer number corresponding to that factor is executed. (Interrupt programs are executed only when the corresponding interrupt factor occurs.)

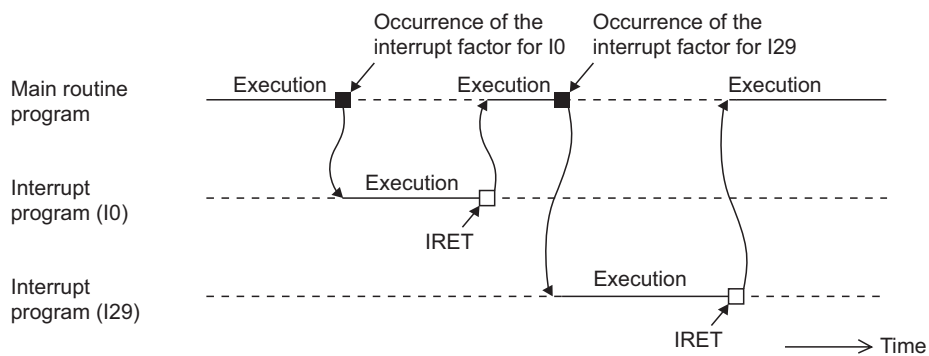
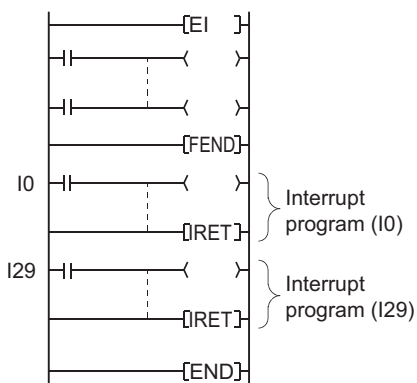


Figure 2.8 Interrupt program execution timing

Only one interrupt program can be created with one interrupt pointer number.



Remark

For details of the interrupt factors and interrupt pointers, refer to Section 9.11.

(2) Programming of interrupt programs

Create interrupt programs between the FEND and END instructions in the main routine program.

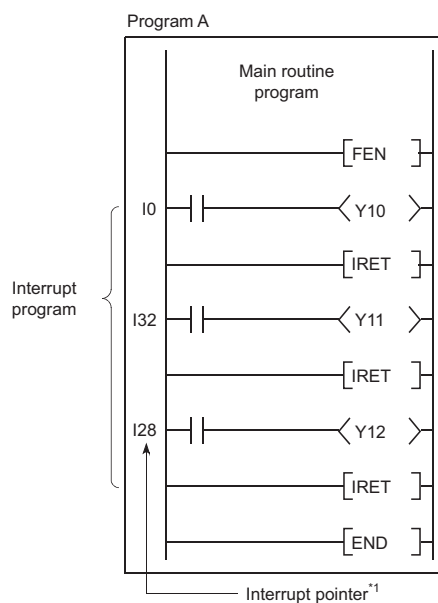


Figure 2.9 Programming location of interrupt programs

*1: The pointer numbers do not need to be specified in ascending order.

Interrupt programs can be managed as one separate program (stand-by type program). (See Section 2.3.3)

(a) Before executing an interrupt program

Before executing the interrupt programs of the interrupt pointers I0 to I15, I28 to I31, I45, and I50 to I255, enable interrupts with the EI instruction.

Remark

For details of the EI instruction, refer to the following.

☞ MELSEC-Q/L Programming Manual (Common Instruction)

(b) Restrictions on programming**1) PLS and PLF instructions**

The PLS and PLF instructions perform off processing in the next scan of which the instruction is executed.

Therefore, the device which is turned on by the instruction remains on until the same instruction is reexecuted.

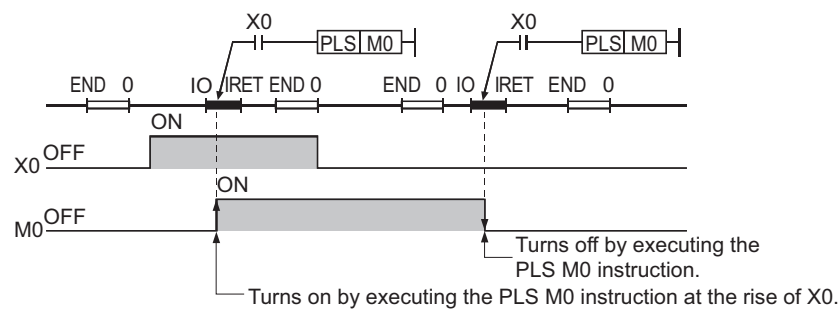


Figure 2.10 Device turned on by the PLS instruction in the interrupt program

2) EI and DI instructions

During execution of an interrupt program, interrupts are disabled (DI) so that any other interrupt processing will not be executed.

Do not execute the EI or DI instruction during interrupt program execution.

3) Timer (T) and counter (C)

Do not use the timer (T) and counter (C) in interrupt programs.

If more than one interrupts occur in one scan, the timer (T) and counter (C) in the interrupt program cannot measure the time correctly.

The OUT C instruction status causes the counter (C) measure the number of interrupts incorrectly.

4) Instructions not available in interrupt programs

Refer to sections corresponding to each instruction in the following.

☞ MELSEC-Q/L Programming Manual (Common Instruction)

(3) Operation when an interrupt factor occurs

There are restrictions on interrupt programs depending on the interrupt factor occurrence timing.

(a) When an interrupt factor occurs before the interrupt program execution status is enabled

The CPU module stores the interrupt factor occurred.

As soon as the interrupt program execution status is enabled, the CPU module executes the interrupt program corresponding to the stored interrupt factor.

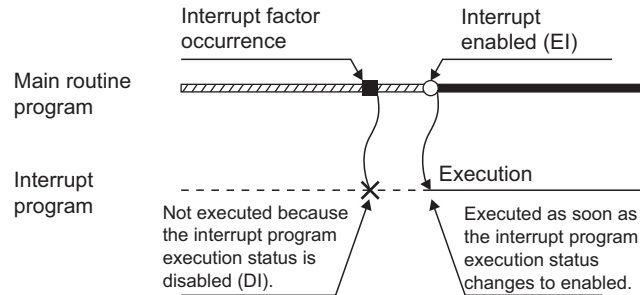


Figure 2.11 When an interrupt factor occurs before interrupts are enabled

When the same interrupt factor occurs more than one time before the interrupt program execution status is enabled, the interrupt factors of I0 to I15, I28 to I31, I45, I50 to I255, and fixed scan execution type programs are stored only once.

Note that all interrupt factors occurred are discarded if they are masked by the IMASK instruction.

Remark

For details of the IMASK instruction, refer to the following.

➞ MELSEC-Q/L Programming Manual (Common Instruction)

(b) When an interrupt factor occurs in the STOP or PAUSE status

The CPU module executes the interrupt program as soon as the interrupt program execution status is enabled after the CPU module status is changed to RUN.

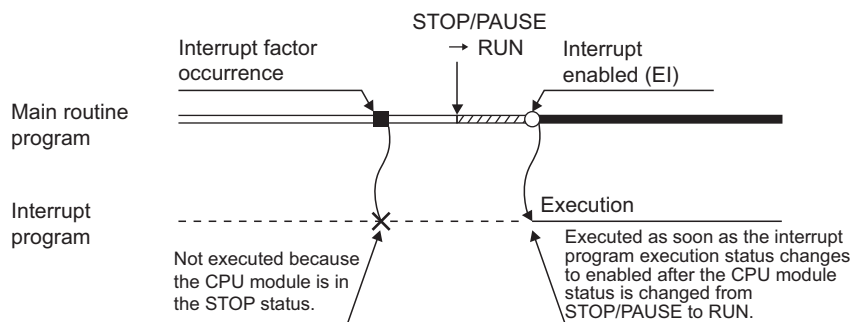


Figure 2.12 When an interrupt factor occurs in the STOP or PAUSE status

(c) When multiple interrupt factors occur simultaneously in the interrupt program execution enabled status

The interrupt programs are executed in the order of interrupt pointers (I□) with high priority.

(☞ Section 9.11.1)

Other interrupt programs have to wait until processing of the interrupt program being executed is completed.

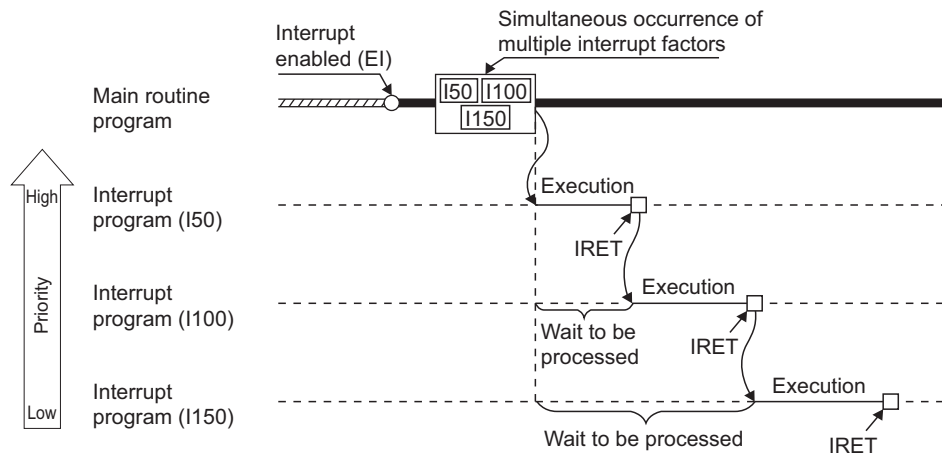


Figure 2.13 When multiple interrupt factors occur simultaneously

(d) When the same interrupt factor as that of the interrupt program being executed occurs

When the same interrupt factor as that of the interrupt program being executed occurs more than one time before completion of interrupt program processing, the interrupt factors of I0 to I15, I45, and I50 to I255 are stored only once, and then the interrupt program corresponding to each stored interrupt factor is executed after completion of current interrupt program execution.

The interrupt factors of I28 to I31 and fixed scan execution type programs are all stored, and then all the interrupt programs corresponding to interrupt factors are executed after completion of current interrupt program execution.

(e) When an interrupt factor occurs during link refresh

The link refresh is suspended and an interrupt program is executed.

Even if the Block data assurance per station setting is enabled in the CC-Link IE or MELSECNET/H network, this setting does not work when a device set as a refresh target is used in the interrupt program.

In the interrupt program, do not use any refresh target device.

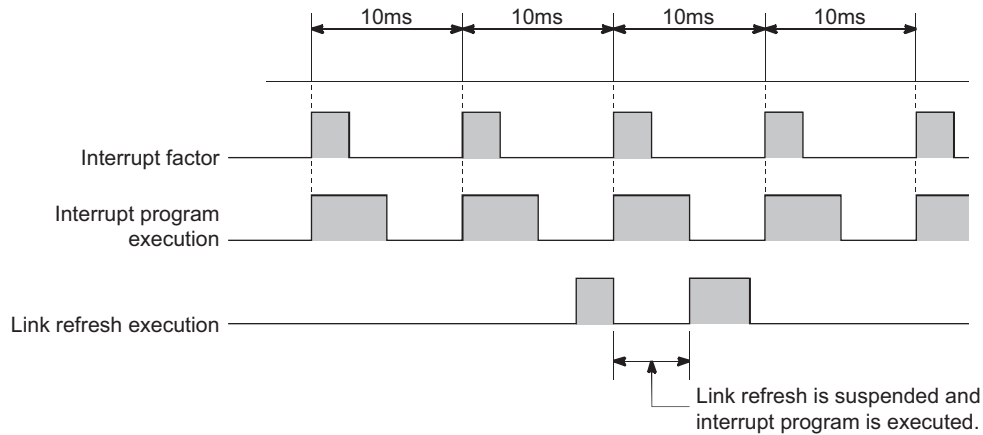


Figure 2.14 When an interrupt factor occurs during link refresh

Remark

For the Block data assurance per station setting, refer to the following.

 Reference manual for each network module

(f) Interrupt during END processing

When the constant scan function is used and an interrupt factor occurs during the waiting time in END processing, an interrupt program corresponding to the interrupt factor is executed.

(g) When an interrupt factor occurs during access to another module

When an interrupt factor occurs during access to another module (during service processing or instruction processing), the interrupt program becomes standby status until the service processing or the instruction in execution is completed.

To shorten the wait time of the interrupt, reduce the amount of data that access to other modules.

(4) Processing at program execution type change

When the program execution type is changed from the scan execution type to the interrupt, the CPU module saves and restores the following data. (☞ Section 9.6.3)

- Data in the index register
- File register block number

Whether to save and restore the data above can be set in the PLC parameter dialog box.

If the data is not saved or restored, the overhead time of the corresponding interrupt program can be shortened.

(☞ Section 10.1.2)

(5) Restrictions

(a) When the same device is used

During execution of an instruction in a main routine program, an interrupt program may be executed, suspending the processing of the instruction being executed.

If the same device is used for the main routine program and interrupt program, device data may become inconsistent. In this case, take the following measures to prevent device data inconsistency.

1) Moving device data to another device

Do not directly specify the device where the data is written by the interrupt program in the main routine program. Use the data in another device by moving the data with the transfer instruction.

2) Disabling interrupts with the DI instruction

Disable interrupts with the DI instruction if instructions that may cause inconvenience for the main routine program are used.

However, interrupts do not occur during access to the device of the corresponding argument of the instruction. For this reason, data inconsistency will not occur in units of arguments.

2.3 Settings When Program is Divided

When one sequence program is divided into multiple programs, execution conditions, such as executing a program only once at start-up or executing a program at fixed intervals, can be set for each program.

(1) Control by multiple programs dividing one program

The CPU module can store multiple programs divided on the basis of each control unit. This enables programming of one sequence program by two or more designers.

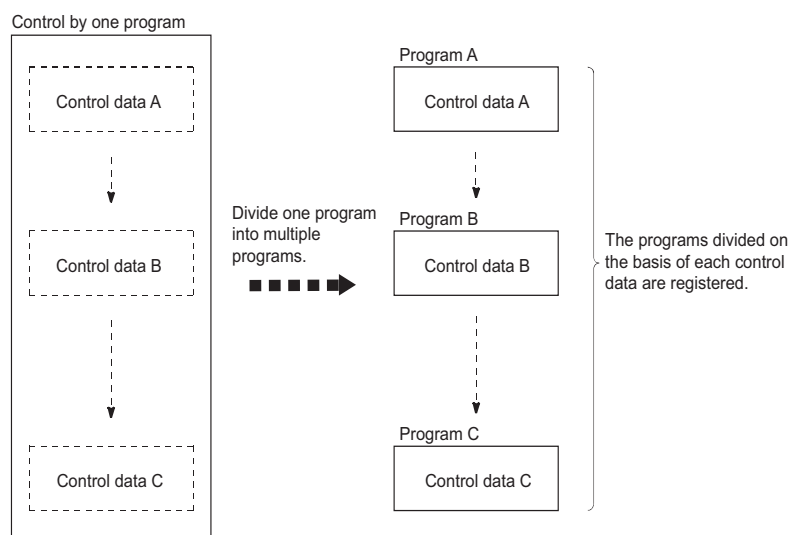


Figure 2.15 Control by multiple programs

(2) Settings required for execution of multiple programs

To execute multiple programs in the CPU module, names (file names) and execution conditions of the programs must be set.

Set them in the Program tab of the PLC parameter dialog box.

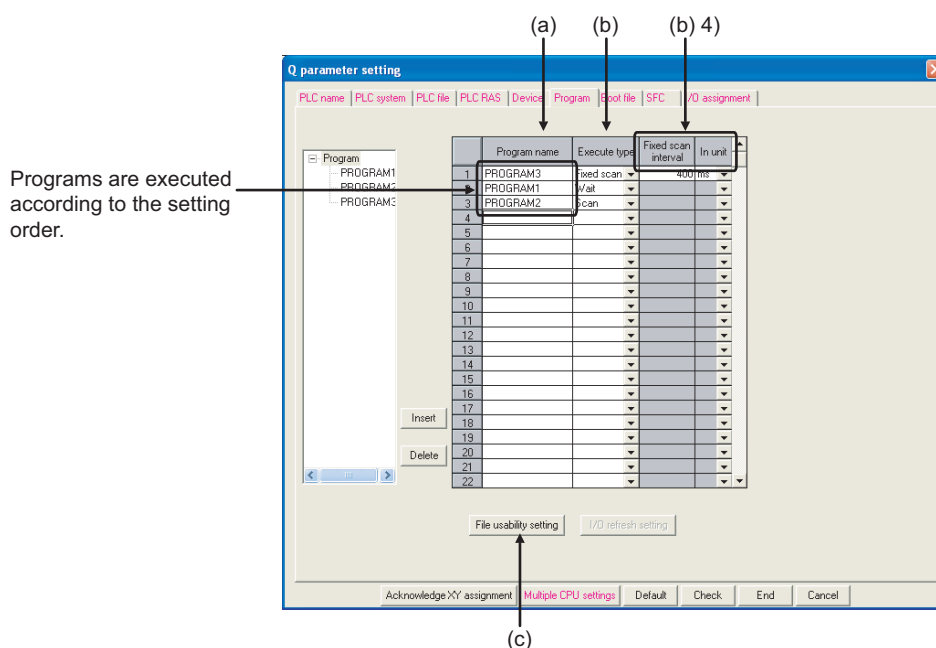


Figure 2.16 Program setting

(a) Program name

Enter the name (file name) of the program to be executed in the CPU module.

(b) Execute type

Select an execution type of the program set under "Program name".

The CPU module executes programs whose execution type has been set here according to the setting order.

1) Initial execution type ("Initial")

This program is executed only once when the CPU module is powered on or its status is switched from STOP to RUN. (☞ Section 2.3.1)

2) Scan execution type ("Scan")

This program is executed once in every scan, starting in the next scan of which the initial execution type program is executed and later. (☞ Section 2.3.2)

3) Stand-by type ("Wait")

This program is executed only when its execution is requested. (☞ Section 2.3.3)

4) Fixed scan execution type ("Fixed scan")

This program is executed at time intervals specified with fixed scan interval and unit.
(☞ Section 2.3.4)

- Fixed scan interval ("Fixed scan interval")

Enter the execution interval of fixed scan execution type program.

The setting range varies depending on the setting unit.

- When the unit is "ms" : 0.5 to 999.5ms (in increments of 0.5ms)
- When the unit is "s" : 1 to 60s (in increments of 1s)

- Unit ("In unit")

Select the unit ("ms" or "s") of the fixed scan interval.

(c) **File usability setting** Note2.1

For each program, determine whether to use the file specified for the local device in the PLC file tab of the PLC parameter dialog box.

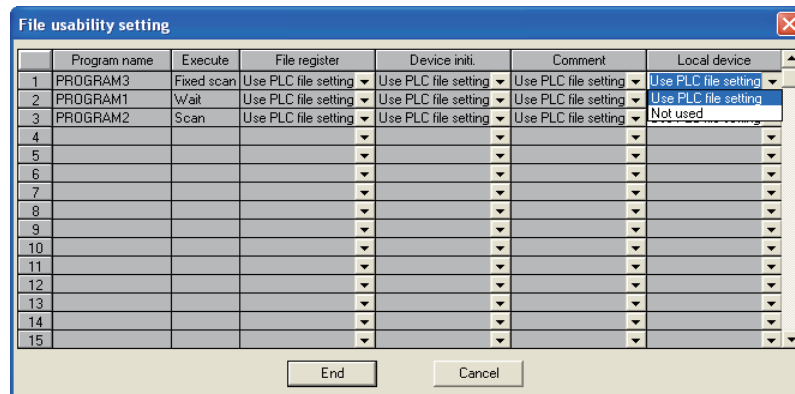


Figure 2.17 File usability setting

The default is set to "Use PLC file setting".

When "Not used" is selected, data in the local device is not saved or restored when the program execution type is changed.

 Note2.1

Universal

The Q00UJCPU does not support the file usability setting.

When using the file usability setting for the Q02UCPU, Q03UDCPU, Q04UDHCPU, Q06UDHCPU, Q13UDHCPU, or Q26UDHCPU, check the versions of the CPU module and GX Developer. ( Appendix 2)

(3) Program sequence in the CPU module

Figure 2.18 shows the program sequence after the CPU module is powered on or its operating status is changed from STOP to RUN.

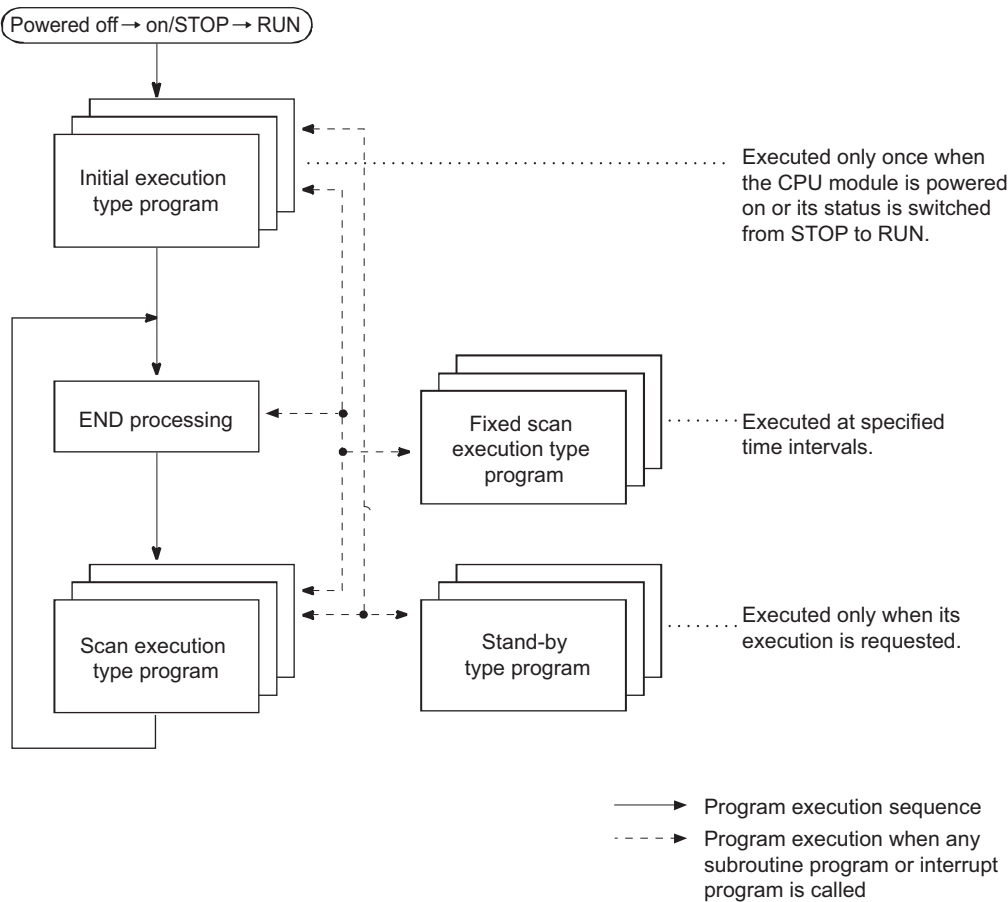


Figure 2.18 Program sequence

Point Use initial execution type program, stand-by type program, and fixed scan execution type program as required.

2.3.1 Initial execution type program

(1) Definition

Initial execution type program is executed only once when the CPU module is powered on or its operating status is changed from STOP to RUN.

This type of program can be used as a program that need not be executed from the next scan and later once it is executed, like initial processing to an intelligent function module.

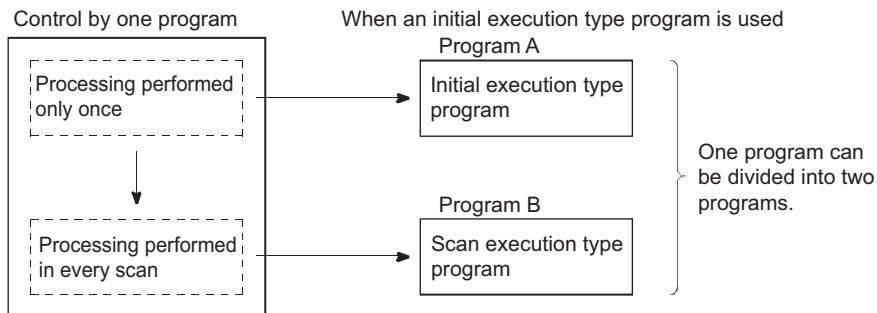


Figure 2.19 When processing performed only once is separated as an initial execution type program

(2) Processing

(a) Execution order

After completion of all the initial execution type program execution, END processing is performed. In the next scan and later, scan execution type programs are executed.

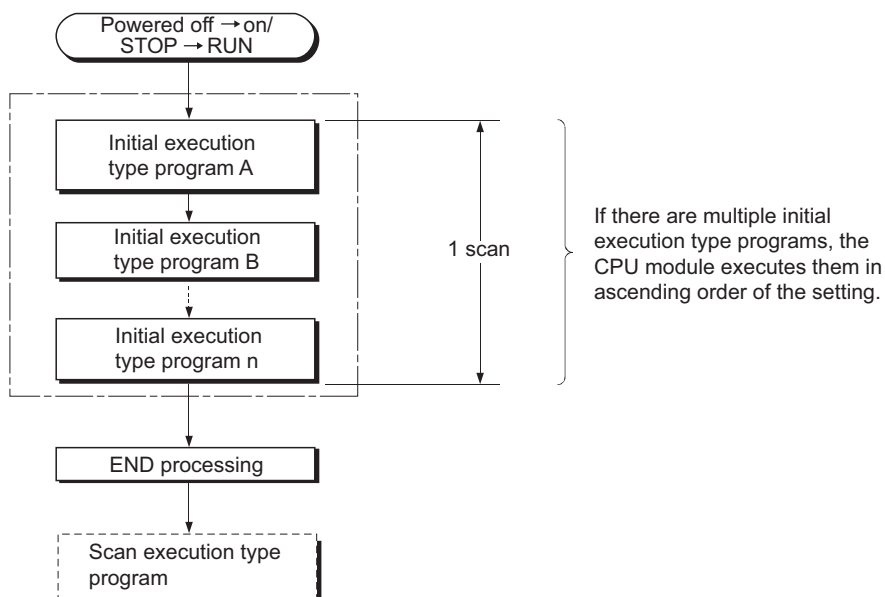


Figure 2.20 Execution order of the initial execution type programs

(b) Initial scan time

Initial scan time is the execution time of initial execution type program.
When multiple programs are executed, the initial scan time will be the time required for completing all the initial execution type program execution.

1) Initial scan time storage location

The CPU module measures the initial scan time and stores it into the special register (SD522 and SD523).
The initial scan time can be checked by monitoring SD522 and SD523.

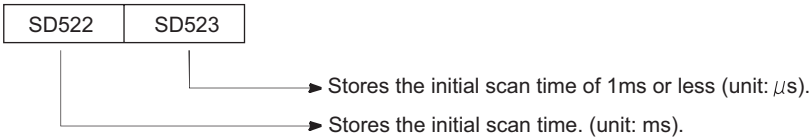


Figure 2.21 Initial scan time storage location

Example If the stored values in SD522 and SD523 are 3 and 400 respectively, the initial scan time is 3.4ms.

2) Accuracy and measurement of the initial scan time

Accuracy of the initial scan time stored in the special register is $\pm 0.1\text{ms}$.
Even if the WDT instruction (instruction that resets the watchdog timer) is executed in the sequence program, the measurement of the initial scan time continues.

3) Execution of an interrupt program or fixed scan execution type program

When an interrupt program or fixed scan execution type program is executed before completion of the initial execution type program execution, the execution time of the executed program will be added to the initial scan time.

(3) Precautions on programming

Initial execution type programs do not support the instructions that require several scans (instructions with completion device).

Example SEND, RECV, and similar instructions

(4) Initial execution monitoring time setting

Initial execution monitoring time is a timer for monitoring initial scan time.

Set a time value in the PLC RAS tab of the PLC parameter dialog box.

The setting range is 10 to 2000ms (in increments of 10ms).

No default value is set.

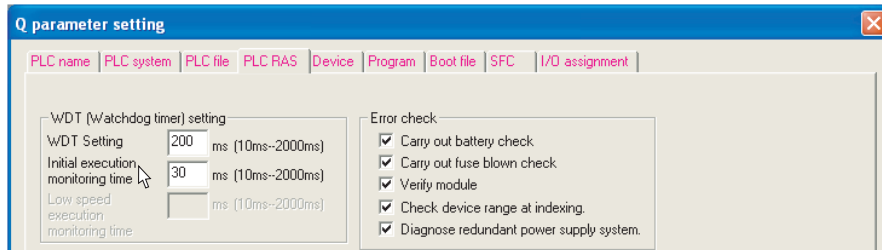


Figure 2.22 PLC RAS setting (Initial execution monitoring time)

(a) When the initial scan time exceeds the preset initial execution monitoring time

"WDT ERROR" occurs and the CPU module stops program operations.

Set a time value so that the initial execution monitoring time becomes longer than actual initial scan time.

Point

An error of the measurement value is 10ms for the initial execution monitoring time setting.

If the initial execution monitoring time (t) parameter is set to 10ms, "WDT ERROR" occurs when actual initial scan time is within the range of $10\text{ms} < t < 20\text{ms}$.

2.3.2 Scan execution type program

(1) Definition

Scan execution type program is executed once in every scan, starting in the next scan of which the initial execution type program is executed and later.

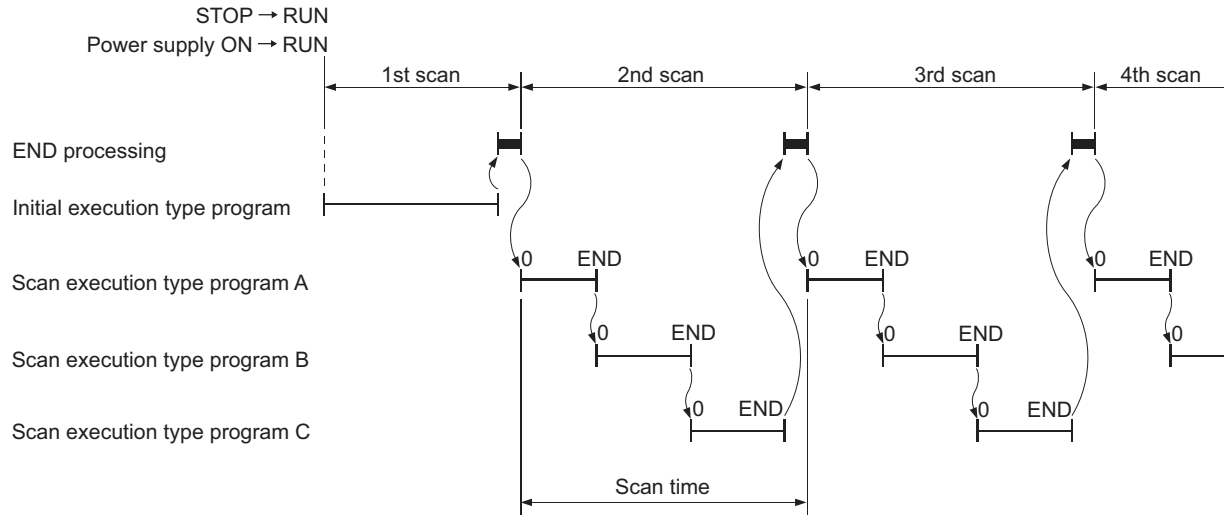


Figure 2.23 Execution order of the scan execution type programs

(2) Processing

When multiple scan execution type programs are executed, the scan time will be the time required for completing all the scan execution type program execution.

If an interrupt program or fixed scan execution type program is executed, execution time of the executed program will be added to the scan time.

2.3.3 Stand-by type program

(1) Definition

Stand-by type program is executed only when its execution is requested.

This type of program can be changed to any desired execution type by a sequence program instruction.

(2) Application

(a) Program library

Stand-by type program is used as a program library, a collection of subroutine programs and/or interrupt programs, and managed separately from a main routine program.

Multiple subroutine programs and/or interrupt programs can be created and managed in a single stand-by type program.

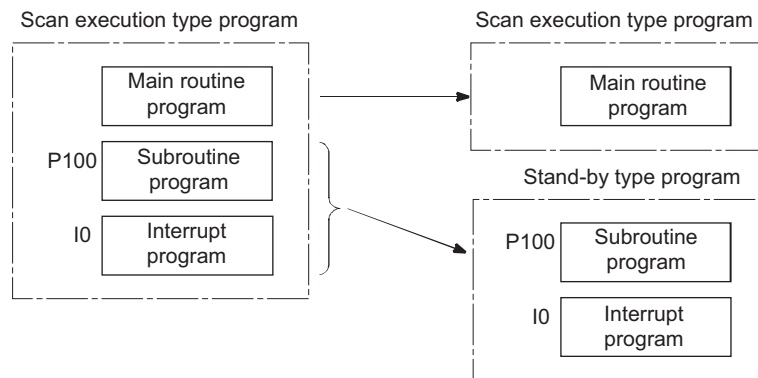


Figure 2.24 Program library using a stand-by type program

(b) Program type change

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

For example, a program preset as a stand-by ("Wait") type program in the PLC parameter dialog box can be changed to a scan execution type program and executed in the sequence program.

(3) Execution method

Execute stand-by type programs in either of the following methods.

- Create subroutine and/or interrupt programs in a stand-by type program and call them using a pointer or when an interrupt occurs.
- Change a stand-by type program to any other execution type using instructions.

(a) Creating subroutine and/or interrupt programs in a single stand-by type program

When creating subroutine and/or interrupt programs in a single stand-by type program, start the program from the step 0.

The FEND instruction used in creation of a subroutine or interrupt program is not required after a main routine program.

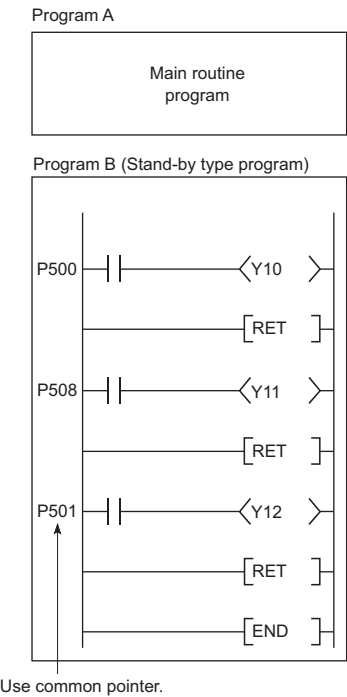


Figure 2.25 Creating subroutine programs in a single stand-by type program

1) Executing a subroutine program and interrupt program in a stand-by type program

After execution of the stand-by type program, the CPU module reexecutes the program that called a program in the stand-by type program.

Figure 2.26 shows the operation when the subroutine and interrupt programs in the stand-by type program are executed.

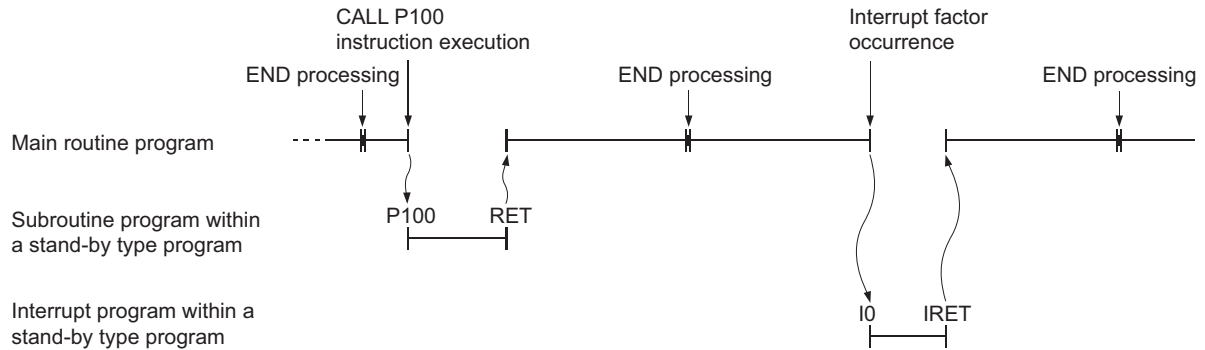


Figure 2.26 Operation when the subroutine and interrupt programs in the stand-by type program are executed

Point

- For restrictions on programming of subroutine and interrupt programs, refer to the following.
 - Subroutine program:  Section 2.2.2
 - Interrupt program:  Section 2.3.2
- Use common pointers. ( Section 9.10.2)
If local pointers are used, subroutine programs in a stand-by type program cannot be executed from any other program.

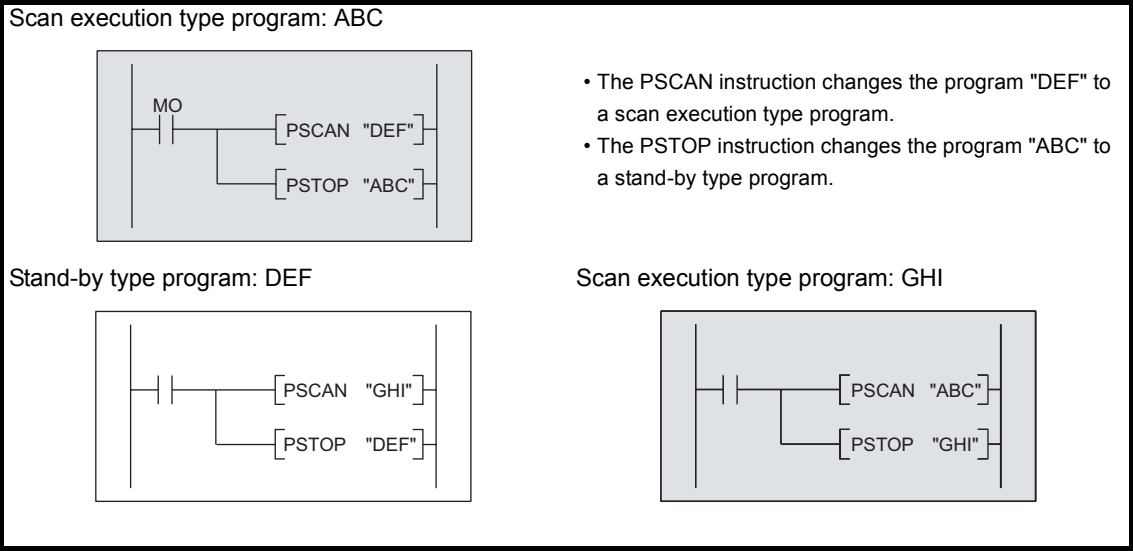
(b) Changing the program execution type using instructions

Use the PSCAN, PSTOP, or POFF instruction to change a program execution type.

1) Changing the execution type (in the case of scan execution type program)

- Set the programs "ABC" and "GHI" as scan execution type programs and the program "DEF" as a stand-by type program.
- When the condition is established (the internal relay (M0) in Figure 2.27 turns on), the program "DEF" is changed to a scan execution type program and the program "ABC" to a stand-by type program.

[Before execution of the PSCAN and PSTOP instructions]



↓ When M0 turns on

[After execution of the PSCAN and PSTOP instructions]

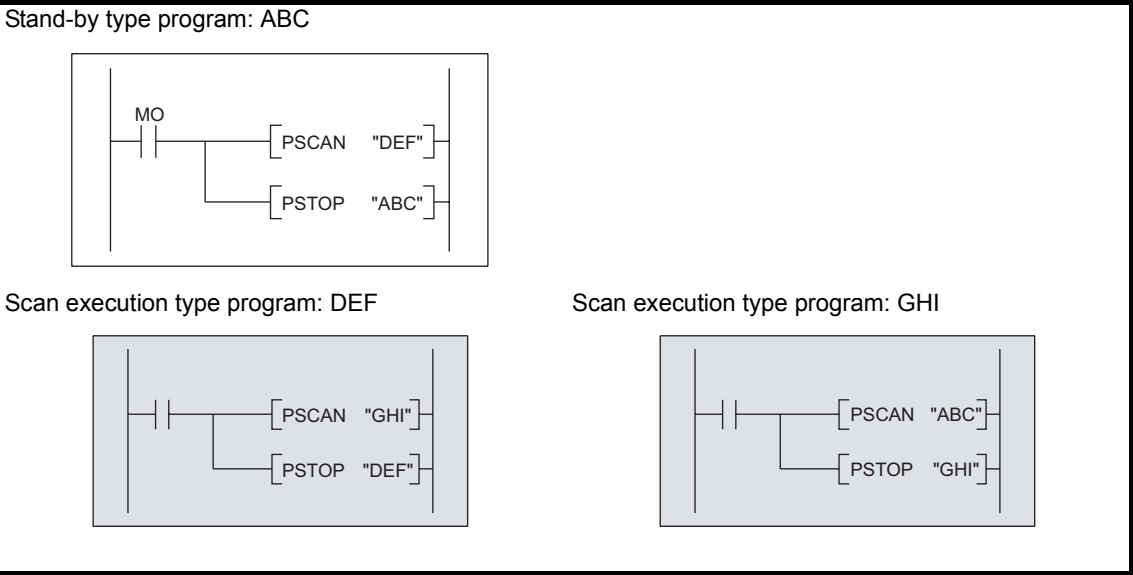


Figure 2.27 Example of changing the execution type (in the case of scan execution type program)

2) Execution type change timing

The program execution type is changed in END processing.

Therefore, the execution type will not be changed in the middle of program execution.

If different types are set to the same program in the same scan, the program will be changed to the type specified by the last instruction executed.

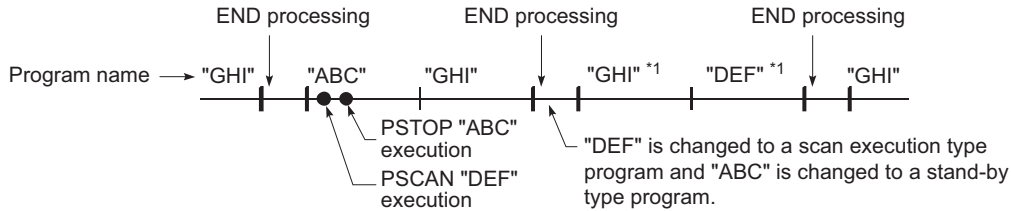


Figure 2.28 Execution type change timing

*1: The programs "GHI" and "DEF" are executed in the order set in the Program tab of the PLC parameter dialog box.

(4) Precautions on programming

(a) Unavailable devices

Unavailable devices depend on the program type (subroutine program or interrupt program) or the execution type changed by an instruction.

(b) Use of local devices

For execution of a subroutine program using a local device, refer to Section 9.14.2.

2.3.4 Fixed scan execution type program

(1) Definition

Fixed scan execution type program is a program executed at specified time intervals.
This type of programs, unlike interrupt programs, can be interrupted in units of files without interrupt pointers or the IRET instruction.
For the restrictions on programming, refer to Section 2.2.3(2)(b).
(The restrictions on programming are the same as those for interrupt programs.)

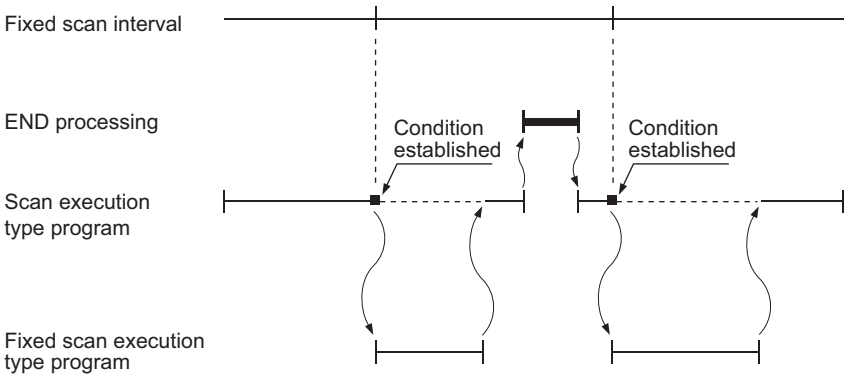


Figure 2.29 Execution of a fixed scan execution type program

Point

To execute a fixed scan execution type program, execute the EI instruction in the initial execution type program or scan execution type program to enable interrupts.

(2) Processing

(a) When two or more fixed scan execution type programs exist

Each fixed scan execution type program is executed at specified time intervals.

If two or more fixed scan execution type programs reach the specified time at the same timing, programs will be executed in ascending order of the numbers set in the Program tab of the PLC parameter dialog box.

(b) When both fixed scan execution type program and interrupt program exist

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 will be given priority.

(c) When the execution condition is established during link refresh

The link refresh is suspended and a fixed scan execution type program is executed.

Even if the Block data assurance per station setting is enabled in the CC-Link IE or MELSECNET/H network, this setting does not work when a device set as a refresh target is used in the fixed scan execution type program.

In the fixed scan execution type program, do not use any refresh target device.

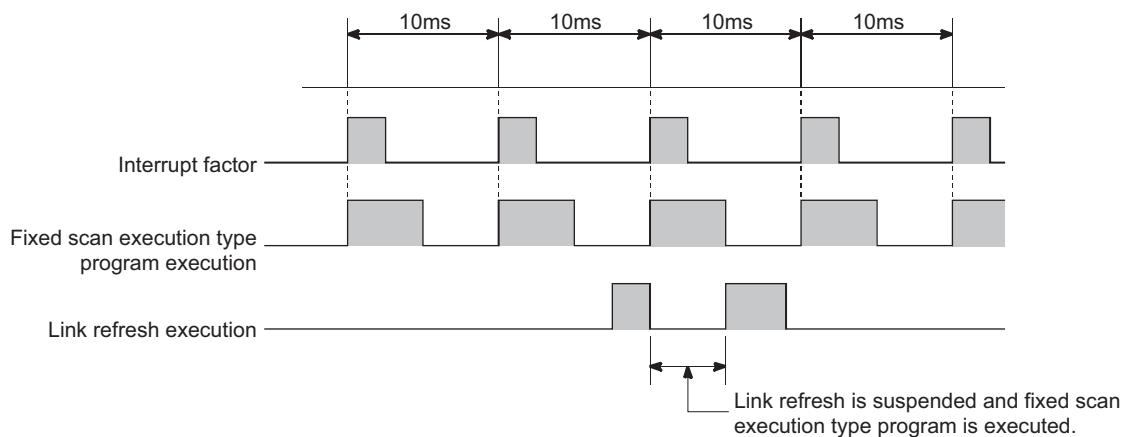


Figure 2.30 When the execution condition is established during link refresh

Remark

For the Block data assurance per station setting, refer to the following.

☞ Reference manual for each network module

(d) When the execution condition is established during END processing

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

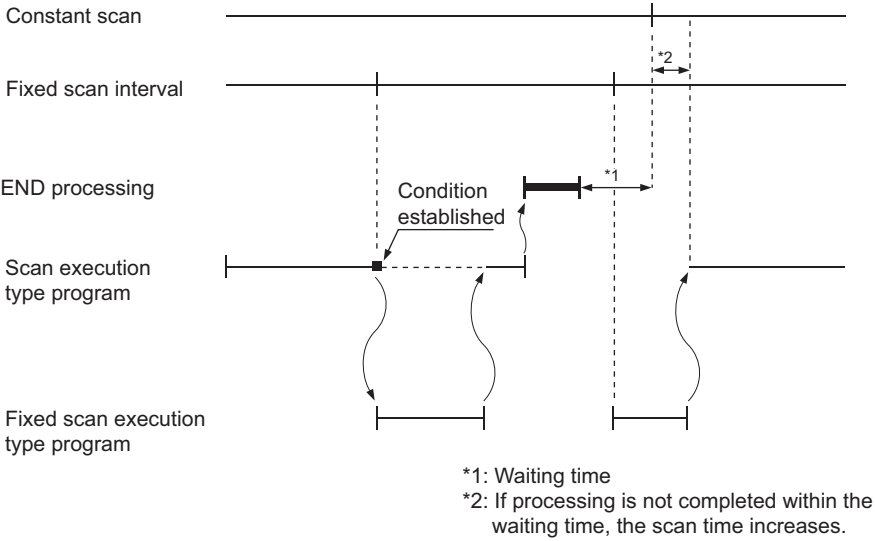


Figure 2.31 When the execution condition is established during the waiting time

(3) Processing at program execution type change

For how to save and restore data in the index register when the program execution type is changed, refer to Section 2.2.3(4). (The method is the same as that for interrupt programs.)

(4) Precautions

(a) Execution interval of a fixed scan execution type program

Execution interval of a fixed scan execution type program may increase from the preset interval depending on the time set for disabling interrupts by the DI instruction (interrupt disabled time). If the interrupt disabled time by the DI instruction becomes too long, use an interrupt program by fixed scan interrupt (I28 to I31) instead of a fixed scan execution type program.

Highest common factor of fixed scan execution interval ^{*1} < Interrupt disabled time ... Condition 1)

^{*1}: This is the highest common factor of execution interval set to multiple fixed scan execution type programs

When the condition 1) is satisfied, the actual execution interval of a fixed scan execution type program may increase from the preset interval by the time shown in the expression below.

$$\frac{\text{Interrupt disabled time}}{\text{Highest common factor of scan execution interval}} \times \text{Fixed scan execution interval set to the corresponding program}$$

The following shows an example of the increase in execution time of a fixed scan execution type program.

Example

- Fixed scan execution interval ... 10ms, 5ms, 1ms, 0.5ms
- Highest common factor of fixed scan execution interval ... 0.5ms
- Interrupt disabled time (DI) ... 5ms
- (Interrupt enabled time (EI) ... less than 0.5ms)

With the settings above, the condition 1) will be 0.5ms < 5ms.

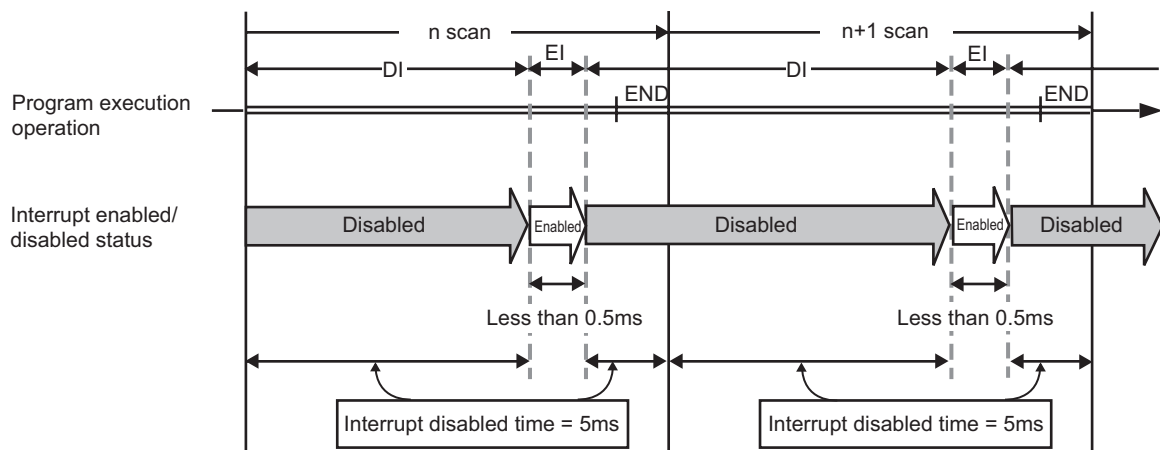


Figure 2.32 Program execution and interrupt enabled/disabled status

The execution time of a fixed scan execution type program whose execution interval is set to 10ms increases 100ms ($5 \div 0.5 \times 10 = 100$) at the most.

2.3.5 Changing the program execution type

(1) Changing the execution type using instructions

(a) Instructions used to change the execution type

The execution type of sequence programs can be changed using instructions even during execution.

Use the PSCAN, PSTOP, or POFF instruction to change the execution type.

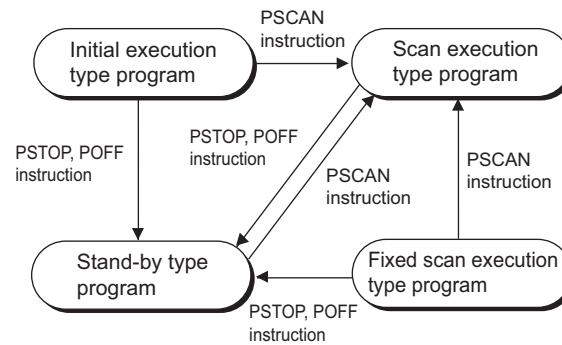


Figure 2.33 Pattern of execution type change

Table2.1 Timing of execution type change

Execution type before change	Instruction		
	PSCAN	PSTOP	POFF
Scan execution type	Remains unchanged.	Changes to the stand-by type.	Turns off outputs in the next scan. Changes to the stand-by type in two scans later.
Initial execution type	Changes to the scan execution type.	Changes to the stand-by type.	Turns off outputs in the next scan. Changes to the stand-by type in two scans later.
Stand-by type	Changes to the scan execution type.	Remains unchanged.	No processing
Fixed scan execution type	Changes to the scan execution type.	Changes to the stand-by type.	Turns off outputs in the next scan. Changes to the stand-by type in two scans later.

Point

Once the fixed scan execution type program is changed to another execution type, the type cannot be returned to the fixed scan execution type.

(b) Execution type change example

In a control program, a stand-by type program matching the preset condition is changed to a scan execution type program in the course of program execution.

An unused scan execution type program can also be changed to a stand-by type program.

Figure 2.34 shows the case where the execution type of the stand-by type programs "ABC", "DEF", "GHI", and "JKL" are changed in the control program.

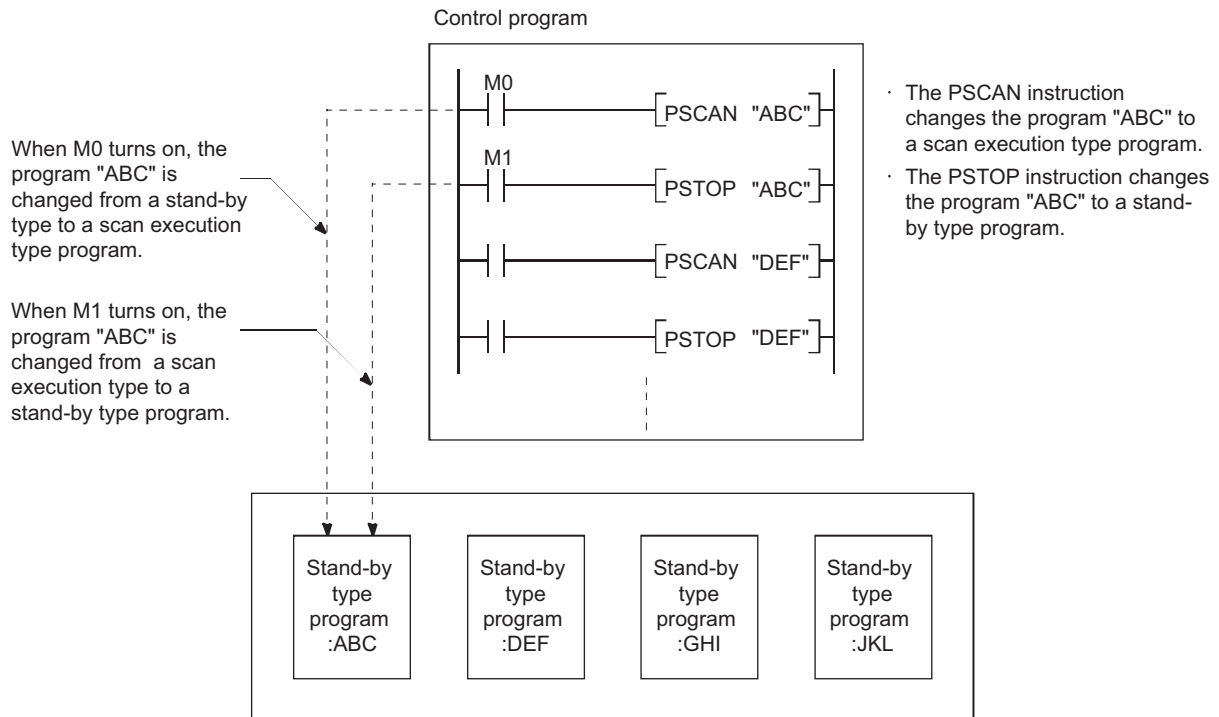


Figure 2.34 Execution type change example

(2) Changing the execution type from the Program monitor list screen

The execution type of programs can be changed on the screen opened by selecting [Online] → [Monitor] → [Program monitor list]. (☞ Section 6.13.1)

2.4 Data Used in Sequence Programs

The CPU module represents numeric and alphabetic data using two symbols (states): 0 (off) and 1 (on).

Data represented using these two symbols is called binary number (BIN).

The CPU module can also use hexadecimal (HEX) (each hexadecimal digit represents four binary bits), binary-coded decimal (BCD), or real numbers.

Table2.2 shows the numeric representations of BIN, HEX, BCD, and DEC (decimal).

Table2.2 Numeric representations of BIN, HEX,BCD,and DEC

DEC (Decimal)	HEX (Hexadecimal)	BIN (Binary)				BCD (Binary-coded decimal)			
0	0				0				0
1	1				1				1
2	2				10				10
3	3				11				11
.	.				.				.
.	.				.				.
.	.				.				.
9	9				1001				1001
10	A				1010			1	0000
11	B				1011			1	0001
12	C				1100			1	0010
13	D				1101			1	0011
14	E				1110			1	0100
15	F				1111			1	0101
16	10			1	0000			1	0110
17	11			1	0001			1	0111
.	.				.				.
.	.				.				.
.	.				.				.
47	2F			10	1111			100	0111
.	.								
.	.								
.	.								
32766	7FFE	0111	1111	1111	1110	-			
32767	7FFF	0111	1111	1111	1111	-			
-32768	8000	1000	0000	0000	0000	1000	0000	0000	0000
-32767	8001	1000	0000	0000	0001	1000	0000	0000	0001
.	.								
.	.								
.	.								
-2	FFFE	1111	1111	1111	1110	-			
-1	FFFF	1111	1111	1111	1111	-			

(1) Inputting numeric values externally to the CPU module

When setting a numeric value to the CPU module externally using a digital switch, BCD (binary-coded decimal) can be used as DEC (decimal) by the method given in (b).

(a) Numeric values used inside the CPU module

The CPU module performs program operations in binary.

If the value set in binary-coded decimal is used without conversion, the CPU module performs program operations regarding the set value as binary.

Therefore, the program operations are not performed correctly. (☞ (b) below)

(b) Using any numeric data regardless of the data type

To convert the data set in binary-coded decimal into binary, which can be used in the CPU module, use the BIN instruction.

The BIN instruction allows the CPU module to use any external numeric data regardless of the data type.

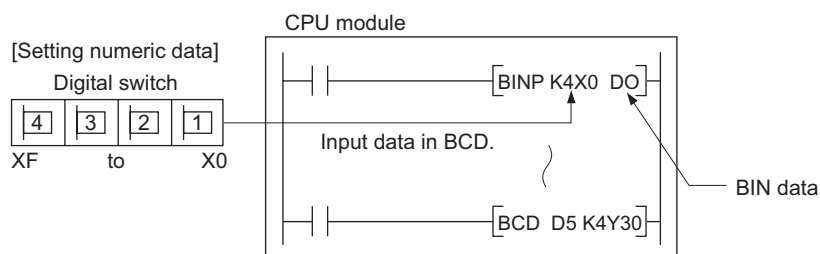


Figure 2.35 Inputting data from a digital switch to the CPU module

Remark

For details of the BIN instruction, refer to the following.

☞ MELSEC-Q/L Programming Manual (Common Instruction)

(2) Outputting numeric values externally from the CPU module

When externally displaying numeric values operated in the CPU module, a digital indicator can be used.

(a) Outputting numeric values

The CPU module performs program operations in binary.

If the binary values used in the CPU module are output to a digital indicator, the indicator does not show the values correctly.

To convert the data set in binary into binary-coded decimal, which can be used in the external indicator, use the BCD instruction.

The BCD instruction allows the external indicator to display values in decimal.

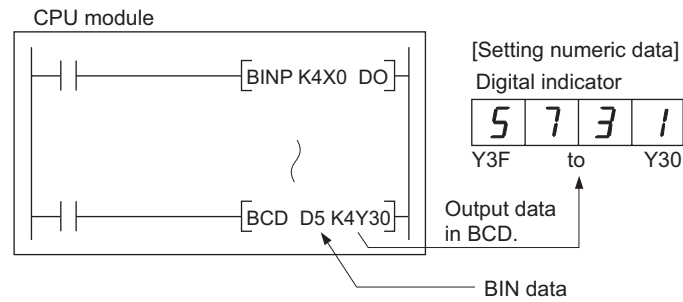


Figure 2.36 Display of operation results in the CPU module by a digital indicator

Remark
For details of the BCD instruction, refer to the following.
☞ MELSEC-Q/L Programming Manual (Common Instruction)
.....

2.4.1 BIN (Binary Code)

(1) Definition

Binary is a numeral system that represents numeric values using two symbols, 0 (off) and 1 (on).

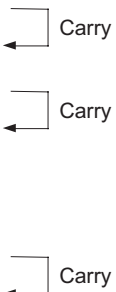
Decimal notation uses the symbols 0 through 9. When the symbols for the first digit are exhausted (a digit reaches 9), the next-higher digit (to the left) is incremented, and counting starts over at 0.

In binary notation, only the symbols 0 and 1 are used. After a digit reaches 1, an increment resets it to 0 and the next digit (to the left) is incremented. (The numeric value becomes 10, which is equal to 2 in decimal.)

Table2.3 shows the numeric representations in BIN and DEC.

Table2.3 Numeric representations in BIN and DEC

DEC (Decimal)	BIN (Binary)
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011



(2) Numeric representation in BIN

(a) Bit configuration of BIN used in the CPU module

Each register (such as the data register, link register) in the CPU module consists of 16 bits.

(b) Numeric data available in the CPU module

Each register in the CPU module can store numeric values in the range of -32768 to 32767.

Figure 2.37 shows the numeric representations for registers.

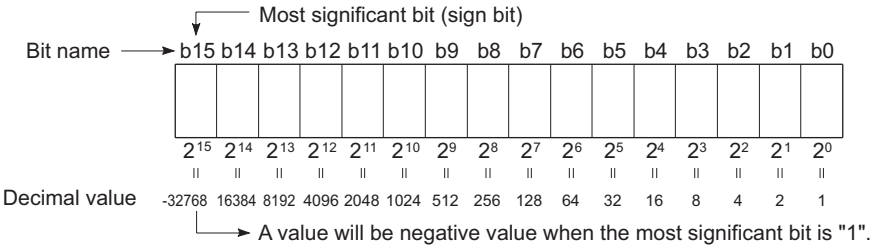


Figure 2.37 Numeric representations for registers in the CPU module

Point

- A numeric value of 2^n is assigned for each bit of registers.
Note that an unsigned binary number (0 to 65535) cannot be used in the most significant bit position since the most significant bit is a sign bit.
- The most significant bit is "0"...Positive
 - The most significant bit is "1"...Negative

2.4.2 HEX (Hexadecimal)

(1) Definition

Hexadecimal (HEX) is a numeral system that represents four binary bits as one digit.

With four binary bits, sixteen different numeric values, 0 to 15, can be represented.

Hexadecimal notation uses 16 symbols to represent numeric values 0 to 15 in one digit, the symbols 0 to 9 to represent values zero to nine, and A_H to F_H to represent values ten to fifteen. After a digit reaches F_H, the next-higher digit (to the left) is incremented.

Table 2.4 shows the numeric representations in BIN, HEX, and DEC.

Table 2.4 Numeric representations in BIN, HEX, and DEC

DEC (Decimal)	HEX (Hexadecimal)	BIN (Binary)	
0	0		0
1	1		1
2	2		10
3	3		11
.	.		.
.	.		.
.	.		.
9	9		1001
10	A		1010
11	B		1011
12	C		1100
13	D		1101
14	E		1110
15	F		1111
16	10	1	0000
17	11	1	0001
.	.		.
.	.		.
.	.		.
47	2F	10	1111

Carry

(2) Numeric representation in HEX

Each register (such as the data register, link register) in the CPU module consists of 16 bits.

In the 16-bit configuration register, 0 to FFFF_H can be specified in hexadecimal.

2.4.3 BCD (Binary-coded Decimal)

(1) Definition

BCD is a numeral system that uses four binary bits to represent the decimal digits 0 through 9. The difference from hexadecimal is that BCD does not use letters A to F. Table2.5 shows the numeric representations in BIN, BCD, and DEC.

Table2.5 Numeric representations in BIN,BCD,and DEC

DEC (Decimal)	BIN (Binary)	BCD (Binary-coded Decimal)	
0	0000		0
1	0001		1
2	0010		10
3	0011		11
4	0100		100
5	0101		101
6	0110		110
7	0111		111
8	1000		1000
9	1001		1001
10	1010	1	0000
11	1011	1	0001
12	1100	1	0010

Carry

(2) Numeric representation in BCD

Each register (such as the data register, link register) in the CPU module consists of 16 bits. Therefore, the numeric values can be stored in each register are those in the range between 0 to 9999 in BCD.

2.4.4 Real number (Floating-point data)

There are two types of real number data: single-precision floating-point data and double-precision floating-point data.

(1) Single-precision floating-point data

(a) Internal representation

Internal representation of real numbers used in the CPU module is given below.

Real number data can be represented as follows, using two word devices.

$$[\text{Sign}] \ 1. [\text{Mantissa}] \times 2^{[\text{Exponent}]}$$

The bit configuration and the meaning of each bit are described below.

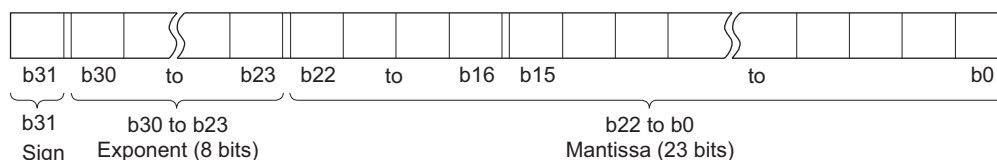


Figure 2.38 Bit configuration of real number data

1) Sign

The most significant bit, b31, is the sign bit.

0: Positive

1: Negative

2) Exponent

The 8 bits, b23 to b30, represent the excess n of 2^n .

The following shows the excess n according to the binary values in b23 to b30.

b23 to b30	FFH	FEH	FDH		81H	80H	7FH	7EH		02H	01H	00H
n	Not used	127	126		2	1	0	-1		-125	-126	Not used

Figure 2.39 Relation between the exponent and excess n

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

Calculation examples are shown below. (The "X" in (nnnnnn) x indicates the numeral system used.)

1) Storing "10"

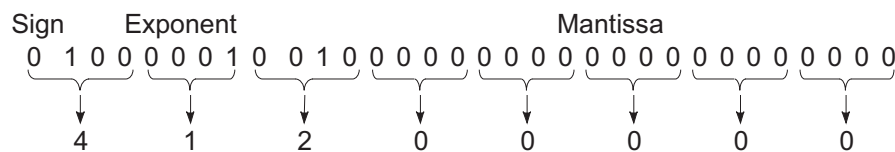
$$(10)_{10} \rightarrow (1010)_2 \rightarrow (1.010000..... \times 2^3)_2$$

Sign: Positive $\rightarrow 0$

Exponent: 3 $\rightarrow 8H \rightarrow (10000010)_2$

Mantissa: (010 00000 00000 00000 00000)₂

In this case, the value will be encoded as 41200000H.

**2) Storing "0.75"**

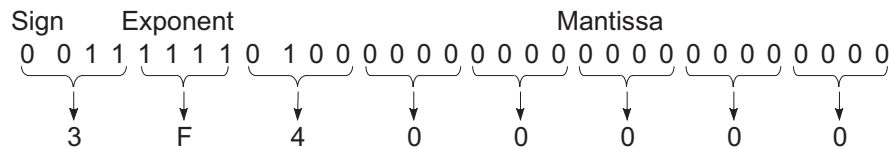
$$(0.75)_{10} \rightarrow (0.11)_2 \rightarrow (1.100..... \times 2^{-1})_2$$

Sign: Positive $\rightarrow 0$

Exponent: -1 $\rightarrow 7EH \rightarrow (01111110)_2$

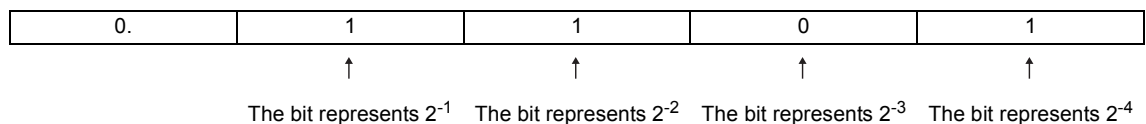
Mantissa: (100 00000 00000 00000 00000)₂

In this case, the value will be encoded as 3F400000H.

**Point**

Values after the decimal point (in binary) is calculated as follows.

Example $(0.1101)_2$



$$(0.1101)_2 = 2^{-1} + 2^{-2} + 2^{-4} = 0.5 + 0.25 + 0.0625 = (0.8125)_{10}$$

(2) Double-precision floating-point data

(a) Internal representation

Real number data used in the CPU module is internally represented as follows, using four word devices.

$$[\text{Sign}] \ 1. [\text{Mantissa}] \times 2^{[\text{Exponent}]}$$

The bit configuration and the meaning of each bit are described below.

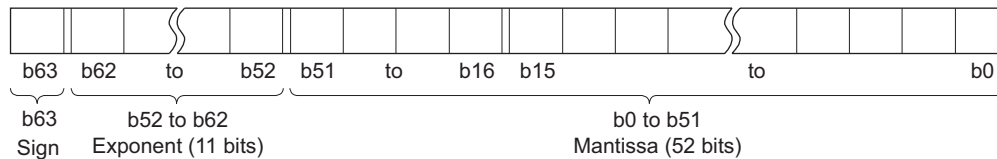


Figure 2.40 Bit configuration of real number data

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 excess n of 2^n .

The following shows the excess n according to the binary values in b52 to b62.

b52 to b62	7FF _H	7FE _H	7FD _H			400 _H	3FF _H	3FE _H	3FD _H	3FC _H			02 _H	01 _H	00 _H
n	Not used	1023	1022			1	1	0	-1	-2			-1021	-1022	Not used

Figure 2.41 Relation between the exponent and excess n

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

Calculation examples are shown below. (The "X" in (nnnnnn) x indicates the numeral system used.)

1) Storing "10"

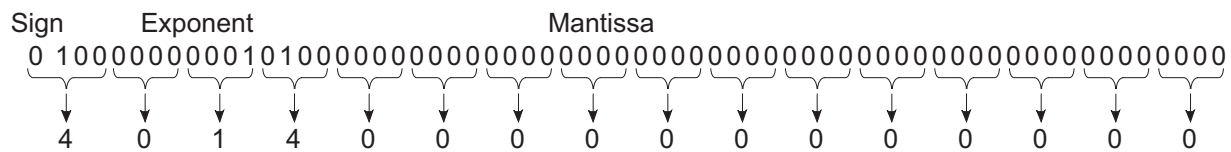
$$(10)_{10} \rightarrow (1010)_2 \rightarrow (1.010000..... \times 2^3)_2$$

Sign: Positive $\rightarrow 0$

Exponent: $3 \rightarrow 401_{10} \rightarrow (100\ 0000\ 0001)_2$

Mantissa: (0100 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000)₂

In this case, the value will be encoded as 4014000000000000H.



2) Storing "0.75"

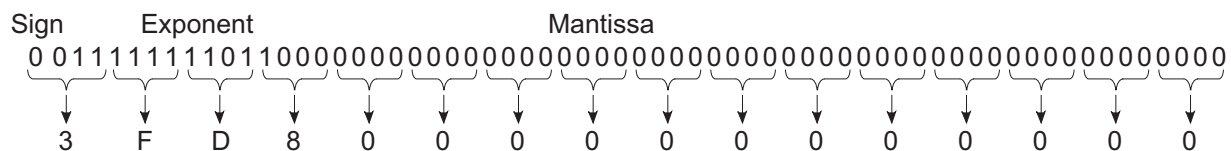
$$(0.75)_{10} \rightarrow (0.11)_2 \rightarrow (1.100\dots \times 2^{-1})_2$$

Sign: Positive $\rightarrow 0$

Exponent: $-1 \rightarrow 3\text{FDH} \rightarrow (011\ 1111\ 1101)_2$

Mantissa: (1000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000)2

In this case, the value will be encoded as 3FD8000000000000_H.



Point

Values after the decimal point (in binary) is calculated as follows.

Example $(0.1101)_2$

0.	1	1	0	1
	↑	↑	↑	↑

The bit represents 2^{-1} The bit represents 2^{-2} The bit represents 2^{-3} The bit represents 2^{-4}

$$(0.1101)_2 = 2^{-1} + 2^{-2} + 2^{-4} = 0.5 + 0.25 + 0.0625 = (0.8125)_{10}$$

2.4.5 Character string data

(1) Definition

The CPU module uses ASCII code data.

(2) ASCII code character strings

Table2.6 lists the ASCII code character strings.

"00H" (NULL code) in Table2.6 is used at the end of a character string as a terminator.

Table2.6 ASCII code character strings

									0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	
									0	0	0	0	1	1	1	1	0	0	0	1	0	1	1	1	1	
									0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	1	
									0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1	
b8	b7	b6	b5	b4	b3	b2	b1	Low	Column																	
									0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F		
				0	0	0	0	0	NULL		(SP)	0	@	P	`	p										
				0	0	0	1	1			!	1	A	Q	a	q										
				0	0	1	0	2			"	2	B	R	b	r										
				0	0	1	1	3			#	3	C	S	c	s										
				0	1	0	0	4			\$	4	D	T	d	t										
				0	1	0	1	5			%	5	E	U	e	u										
				0	1	1	0	6			&	6	F	V	f	v										
				0	1	1	1	7			'	7	G	W	g	w										
				1	0	0	0	8			(	8	H	X	h	x										
				1	0	0	1	9			)	9	I	Y	i	y										
				1	0	1	0	A			*	:	J	Z	j	z										
				1	0	1	1	B			+	;	K	[	k	{										
				1	1	0	0	C			(Comma) ,	<	L	\	l											
				1	1	0	1	D			(Minus) -	=	M	]	m	}										
				1	1	1	0	E			(Period) .	>	N	^	n	—										
				1	1	1	1	F			/	?	O	(Under line) _	o											

CHAPTER3 CPU MODULE OPERATION

This chapter describes operation of the CPU module.

3.1 Initial Processing

The CPU module performs preprocessing required for sequence program operations.

The preprocessing is executed only once when any of the operations described in Table3.1 is performed to the CPU module.

When initial processing is completed, the CPU module will be placed in the operation status set by the RUN/STOP/RESET switch. (☞ Section 3.5)

Table3.1 Initial processing list

Initial processing item	CPU module status		
	Powered-on	Reset	Changed from STOP to RUN ^{*1}
The I/O module initialization	○	○	×
Boot from a memory card	○	○	×
PLC parameter check	○	○	○
Multiple CPU system parameter consistency check	○	○	○
Initialization of devices outside the latch range (bit device: off, word device: 0)	○	○	×
Automatic I/O number assignment of mounted modules	○	○	○
CC-Link IE and MELSECNET/H information setting	○	○	×
Intelligent function module switch setting	○	○	×
CC-Link information setting	○	○	×
Ethernet information setting	○	○	×
Initial device value setting	○	○	○
Serial communication function setting	○	○	×

○ : Performed, × : Not performed

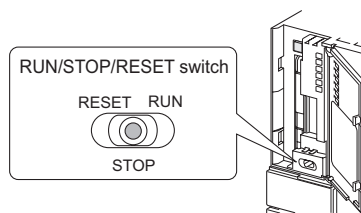
*1: The operation indicates that the status is changed back to RUN without resetting the module after any parameter or program was changed in the STOP status.

(The RUN/STOP/RESET switch is set from STOP to RUN (the RUN LED will flash), then back to STOP and to RUN again.)

Note that the PLS □ P instruction (instruction for pulse conversion) may not be executed properly with the above operation. This is because the previous information may not be inherited depending on the program changes.

Point

If any parameter or program is changed in the STOP status, reset the CPU module using the RUN/STOP/RESET switch.



3.2 I/O Refresh (Refresh Processing with Input/Output Modules)

The CPU module performs the following before sequence program operations.

- On/off data input from the input module or intelligent function module to the CPU module
- On/off data output from the CPU module to the output module or intelligent function module

When the constant scan time is set, I/O refresh is performed after the constant scan waiting time has elapsed. (I/O refresh is performed at each constant scan cycle.)

3.3 Program Operation

The CPU module sequentially executes the program stored in the module from the step 0 to the END or FEND instruction. (➡ CHAPTER 2)

3.4 END Processing

The CPU module performs refresh processing with network modules and communication with external devices.

END processing includes the following.

- Refresh with network modules (➡ CHAPTER 10)
- Auto refresh with intelligent function module (➡ Section 7.1.1)
- Intelligent function module dedicated instruction processing (➡ CHAPTER 10)
- Device data latch processing (➡ Section 6.3, CHAPTER 10)
- Service processing (➡ Section 6.24, CHAPTER 10)
- Watchdog timer reset (➡ Section 6.16)
- Auto refresh between multiple CPU modules (➡ QCPU User's Manual (Multiple CPU System))
- Device data collection using the sampling trace function (only when trace point is set to every scan (after END instruction execution)) (➡ Section 6.14)
- Self-diagnostics processing (➡ Section 6.17)
- Special relay/special register value setting (only for those that should be set during END processing) (➡ CHAPTER 12)

Point

When the constant scan function (➡ Section 6.2) is used, the results of processing performed in END processing are held for the period between after END processing is completed and until the next scan starts.

3.5 Operation Processing in the RUN,STOP, or PAUSE Status

There are three types of operating status of the CPU module.

- RUN status
- STOP status
- PAUSE status

This section describes program operation processing in the CPU module based on its operating status.

(1) Operation processing in the RUN status

RUN status is a status where sequence program operations are repeatedly performed in a loop between the step 0 and the END (FEND) instruction.

(a) Output status when entering the RUN status

The CPU module outputs either of the following according to the output mode parameter setting when its status is changed to RUN. (☞ Section 6.4)

- Output (Y) status saved immediately before entering the STOP status
- Result of operations performed for one scan after entering the RUN status

(b) Processing time required before operations

The processing time required for the CPU module to start sequence program operations after its operating status is changed from STOP to RUN varies depending on the system configuration and/or parameter settings. (It takes one to three seconds normally.)

(2) Operation processing in the STOP status

STOP status is a status where sequence program operations are stopped by the RUN/STOP/RESET switch or the remote STOP function (☞ Section 6.6.1).

The CPU module status will be changed to STOP when a stop error occurs.

(a) Output status when entering the STOP status

When entering the STOP status, the CPU module saves data in the output (Y) and turns off all outputs.

The device memory other than that of the output (Y) will be held.

(3) Operation processing in the PAUSE status

PAUSE status is a status where sequence program operations are stopped by the remote PAUSE function (☞ Section 6.6.2) after operations are performed for one scan, holding the output and device memory status.

(4) Operation processing in the CPU module when switch operation is performed

Table3.2 Operation processing when switch operation is performed

RUN/STOP status	CPU module operation processing			
	Sequence program operation processing	External output	Device memory	
			M,L,S,T,C,D	Y
RUN → STOP	The CPU module executes the program until the END instruction and stops.	The CPU module saves the output (Y) status immediately before its status is changed to STOP and turns off all the outputs.	The CPU module holds the device memory status immediately before its status is changed to STOP.	The CPU module saves the output (Y) status immediately before its status is changed to STOP and turns off all the outputs.
STOP → RUN	The CPU module executes the program from the step 0.	The CPU module outputs data according to the output mode parameter setting. ( Section 6.4)	The CPU module holds the device memory status immediately before its status is changed to STOP. Note that the CPU module uses initial device values if those values are preset. Local device data are cleared.	The CPU module outputs data according to the output mode parameter setting. ( Section 6.4)

Point

The CPU module performs the following in any of the RUN, STOP, or PAUSE status.

- Refresh processing with I/O modules
- Refresh processing with network modules
- Auto refresh processing with intelligent function modules
- Self-diagnostics processing
- Service processing
- Intelligent function module dedicated instruction processing (completion processing only)
- Operation processing of Multiple CPU high speed transmission function
- Setting values for special relay/special register (only for those that should be set during END processing)

Even if the CPU module is in the STOP or PAUSE status, the following operations can be executed.

- I/O monitor or test operation from GX Developer
- Read/write data from/to external devices using the MC protocol
- Communication with other stations using CC-Link IE or MELSECNET/H
- Communication with CC-Link remote stations

3.6 Operation Processing during Momentary Power Failure

When the input voltage supplied to the power supply module drops below the specified range, the CPU module detects a momentary power failure and performs the following operation.

(1) When a momentary power failure occurs for a period shorter than the allowable power failure time

The CPU module registers error data and suspends the operation processing.

The CPU module, however, continues measurement in the timer device and holds the output status.

(a) When resume start is specified for the SFC program

Data in the system is saved.

(b) When power is recovered after a momentary power failure

The CPU module restarts its operation processing.

(c) Watchdog timer (WDT) measurement during a momentary power failure

Even if operation processing is suspended due to a momentary power failure, the CPU module continues the measurement of the watchdog timer (WDT).

For example, when the WDT setting of PLC parameter is 200ms and the scan time is 190ms, if a momentary power failure occurs for 15ms, "WDT ERROR" occurs.

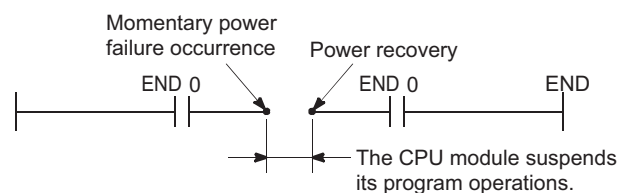


Figure 3.1 Operation processing during a momentary power failure

(2) When a momentary power failure occurs for a period longer than the allowable power failure time

The CPU module starts its operations initially.

Operation processing will be the same as that when any of the following is performed.

- Programmable controller is powered on.
- The CPU module is reset by the RUN/STOP/RESET switch.
- The CPU module is reset by GX Developer (the remote reset operation).

Point

- In a redundant power supply system, the CPU module does not suspend its operations if a momentary power failure occurs in either of the power supply modules. However, if a momentary power failure occurs under the condition where the power is supplied to only one of the power supply modules, operations are suspended.
- Information of a momentary power failure occurred in a redundant power supply system will be stored in SM1782 to SM1783 and SD1782 to SD1783. On the other hand, information of a momentary power failure occurred in a single power supply system will be stored in SM53 and SD53. (➡ CHAPTER 12)

3.7 Data Clear Processing

This section describes how to clear data in the CPU module and the setting required for the latch data clear.

(1) Clearing data in the CPU module

Data in the CPU module are cleared when the reset operation (by the RUN/STOP/RESET switch or by powering the module off and then on) is performed.

However, data in (a) below cannot be cleared by the reset operation.

(a) Data that cannot be cleared by the reset operation

- Data in the program memory
- Data in the standard ROM
- Data in a memory card
- Data in latch-specified devices ( (2) in this section)
- Data in the file register

(b) Clearing data that cannot be cleared by the reset operation

1) Data in the program memory

Data can be cleared by:

- selecting the "Clear program memory" checkbox in the Boot file tab of the PLC parameter dialog box, or
- selecting [Online] → [Delete PLC data] in GX Developer.

2) Data in the standard ROM

Data can be cleared automatically when the data is written to the standard ROM.

3) Data in a memory card

Data can be cleared by selecting [Online] → [Delete PLC data] in GX Developer.

4) Data in latch-specified devices

Refer to (2) in this section.

5) Data in the file register

Data can be cleared by:

- resetting devices with the RST instruction,
- transferring K0 with the MOV or FMOV instruction, or
 MELSEC-Q/L Programming Manual (Common Instruction)
- executing "Clear all file registers" from the screen opened by selecting [Online] → [Clear PLC memory] in GX Developer.

(2) Latch specification of devices

Set a latch range for each latch-target device in the Device tab of the PLC parameter dialog box.

( Section 6.3(5))

(a) Latch range setting

Two kinds of latch range can be set by GX Developer.

1) Latch clear operation enable range ("Latch (1) start/end")

Data in this latch range can be cleared with the remote latch clear operation.

2) Latch clear operation disable range ("Latch (2) start/end")

Data in this latch range cannot be cleared with the remote latch clear operation.

(b) Clearing device data set in the latch clear operation enable range

Clear data with the remote latch clear operation ( Section 6.6.4).

(c) Clearing device data set in latch clear operation disable range

Clear data by:

- resetting devices with the RST instruction,
- transferring K0 with the MOV or FMOV instruction, or
 MELSEC-Q/L Programming Manual (Common Instruction)
- executing "Clear device's whole memory (including latch)" from the screen opened by selecting [Online] → [Clear PLC memory] in GX Developer.

Point

Latching devices increases the scan time.

When latching a device, consider the increase in the scan time. ( Section 10.1.2(9))

Remark

For the operation of GX Developer, refer to the following.

 GX Developer Version 8 Operating Manual

3.8 I/O Processing and Response Delay

The CPU module performs I/O processing in the refresh mode.

Using the direct access input/output in a sequence program, however, allows the CPU module to perform I/O processing in the direct mode at the time of each instruction execution.

This section describes these I/O processing modes of the CPU module and response delays.

(a) Refresh mode(Section 3.8.1)

Refresh mode is a mode for the CPU module to access input/output modules and perform I/O processing collectively before the start of sequence program operations.

(b) Direct mode(Section 3.8.2)

Direct mode is a mode for the CPU module to access input/output modules and perform I/O processing at the timing when each instruction is executed in a sequence program.

To access input/output modules in the direct mode, use the direct access input or direct access output in a sequence program.

(1) Differences between refresh mode and direct mode

The direct mode directly accesses input/output modules at execution of an instruction. Therefore, data input is faster than in refresh mode.

Processing time required for each instruction, however, is longer.

Table3.3 shows the availability of the refresh mode and the direct mode for each input and output.

Table3.3 Availability of modes

Item	Refresh mode	direct mode
Input/output modules	Available	Available
Input/output of intelligent function modules		
Remote input/output in CC-Link IE, MELSECNET/H, or CC-Link	Available	Not available

3.8.1 Refresh mode

(1) Definition

Refresh mode is a mode for the CPU module to access input/output modules and perform I/O processing collectively before the start of sequence program operations.

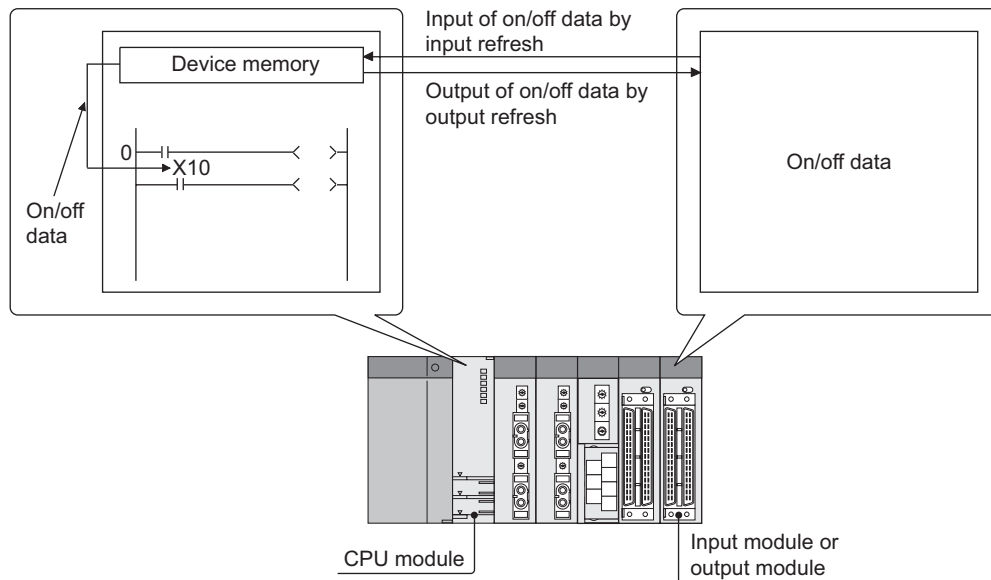


Figure 3.2 Refresh mode

(2) Input

On/off data of an input module are batch-input to the area for communication with the input module in the CPU module before the start of sequence program operations.

The CPU module performs sequence program operations using the on/off data stored in the input (X) device memory.

(3) Output

The operation results of the sequence program is output to the output (Y) device memory in the CPU module every time program operation is performed. Then, the CPU module batch-outputs the on/off data in the output (Y) device memory to an output module before the start of sequence program operations.

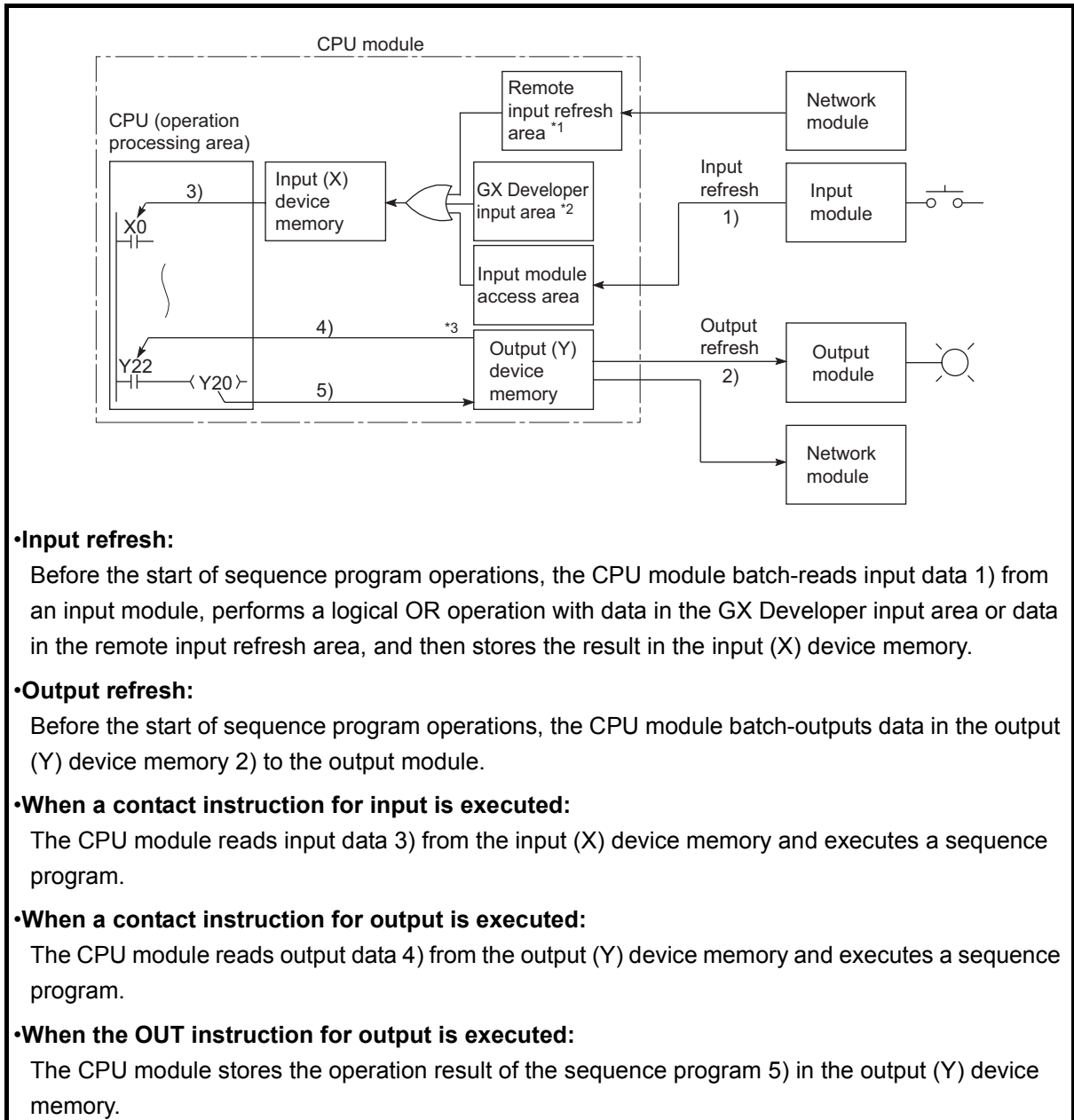


Figure 3.3 I/O data flow in refresh mode

*1: The remote input refresh area indicates the area to be used when auto refresh is set to the input (X) in the CC-Link IE, MELSECNET/H, or CC-Link.

Data in the remote input refresh area will be refreshed automatically during END processing.

*2: Data in the GX Developer input area can be turned on/off by the following operation.

- Test operation by GX Developer
- Writing data from a network module

*3: Data in the output (Y) device memory can be turned on/off by the following operation.

- Test operation by GX Developer
- Refresh via CC-Link IE or MELSECNET/H
- Writing data from an external device using the MC protocol
- Auto refresh via CC-Link

(4) Response delay

An output response which corresponds to the status change in the input module delays for two scans (maximum) depending on the on timing of an external contact.

Examples

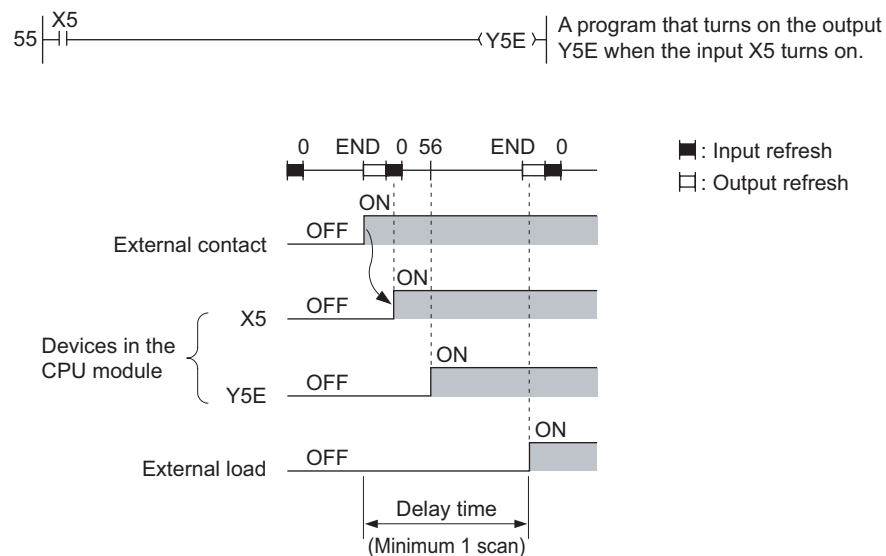


Figure 3.4 Y5E turns on the earliest

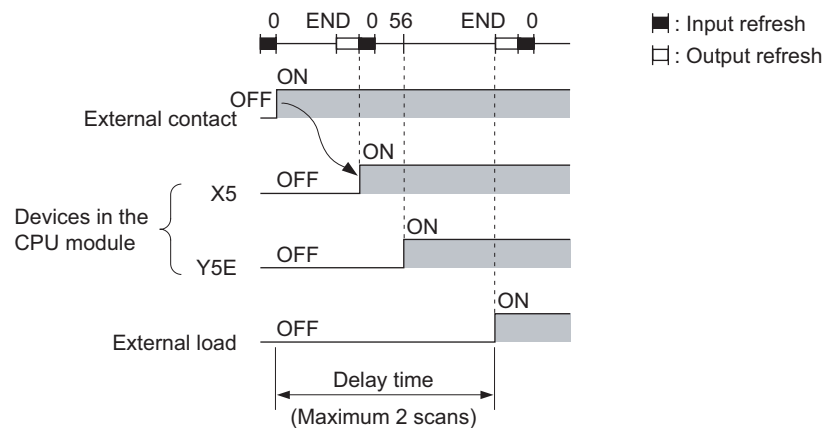


Figure 3.5 Y5E turns on the latest

3.8.2 Direct mode

(1) Definition

The direct mode is a mode for the CPU module to access input/output modules and performs I/O processing at the timing when each instruction is executed in a sequence program.

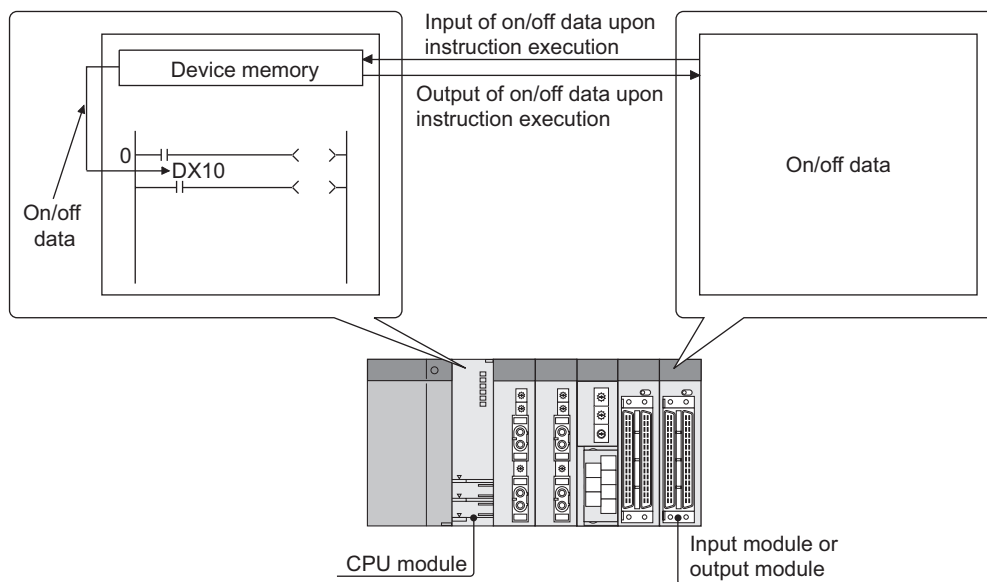


Figure 3.6 Direct mode

With this mode, the CPU module uses the direct access input (DX) and direct access output (DY) to perform I/O processing.

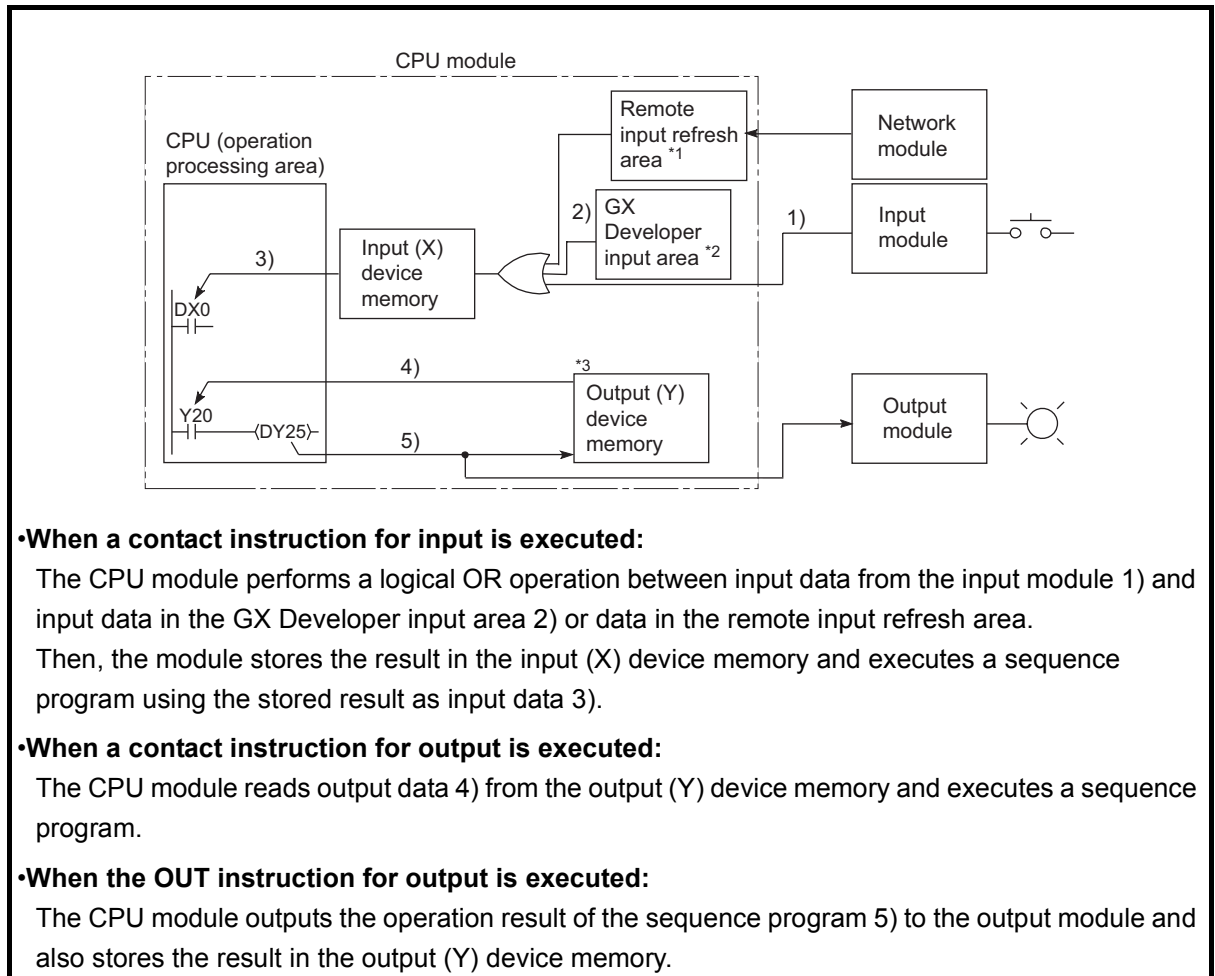


Figure 3.7 I/O data flow in direct mode

- *1: The remote input refresh area indicates the area to be used when auto refresh is set to the input (X) in the CC-Link IE, MELSECNET/H, or CC-Link.
Data in the remote input refresh area will be refreshed automatically during END processing.
- *2: Data in the GX Developer input area can be turned on/off by the following operation.
- Test operation by GX Developer
 - Writing data from a network module
- *3: Data in the output (Y) device memory can be turned on/off by the following operation.
- Test operation by GX Developer
 - Refresh via CC-Link IE or MELSECNET/H
 - Writing data from an external device using the MC protocol
 - Auto refresh via CC-Link

(2) Response delay

An output response which corresponds to the status change in the input module delays for one scan (maximum) depending on the on timing of an external contact.

Examples

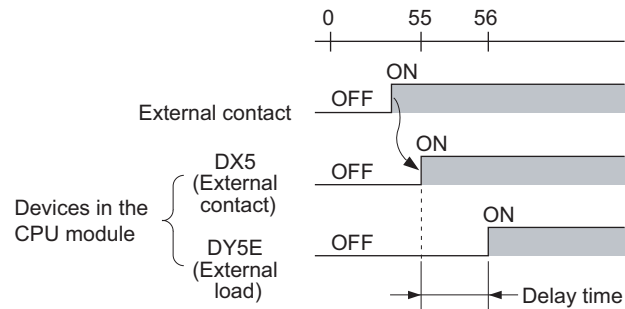
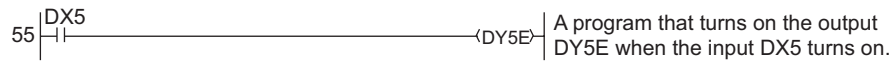


Figure 3.8 DY5E turns on the earliest

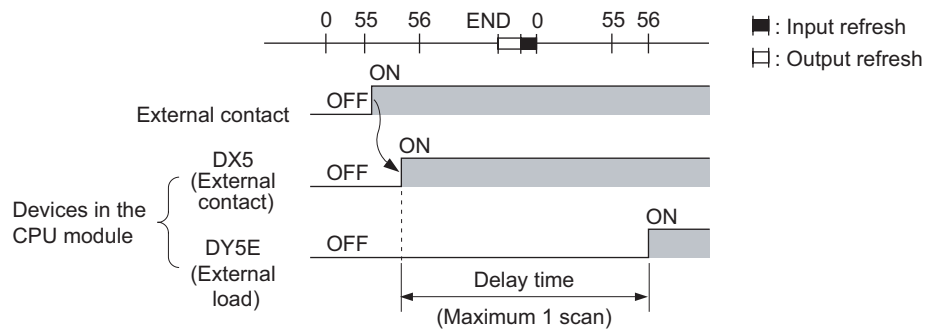


Figure 3.9 DY5E turns on the latest

CHAPTER4 ASSIGNMENT OF BASE UNIT AND I/O NUMBER

This chapter describes the base unit and I/O number assignment required for the CPU module to communicate data with I/O modules and/or intelligent function modules.

4.1 Base Unit Assignment

4

4.1.1 Base mode

Use this mode when assigning the number of available slots to the main base unit and extension base units. The following two modes are available.

- Auto mode
- Detail mode

(1) Auto mode

Use this mode when assigning the number of slots equal to that on the base unit used.

(2) Detail mode

Use the detail mode when assigning the number of slots for each base unit.

Any number of slots can be assigned irrespective of the actual number of slots on the base unit to be used.

(a) Setting the number of slots greater than the actual one

Slots are occupied by the number of slots set.

The slots after actually used ones are regarded as empty slots.

For example, three slots will be the empty slots when a 5-slot base unit is used and the number of available slots are set to eight.

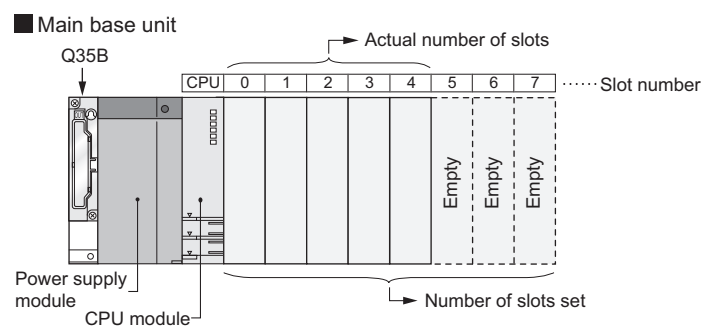


Figure 4.1 Setting the number of slots greater than the actual one

The number of points for the empty slots will be either value set on the PLC system tab, or on the I/O assignment tab in the PLC parameter dialog box. (The default is 16 points.)

(b) Setting the number of slots smaller than the actual one

Set the smaller number than the actual number of slots when slots with no module mounted need not be recognized.

For example, four slots from the right end of the base unit will be the prohibited slots when using a 12-slot base unit and setting the number of available slots to eight. (Mounting a module on a prohibited slot causes "SP.UNIT LAY ERR.")

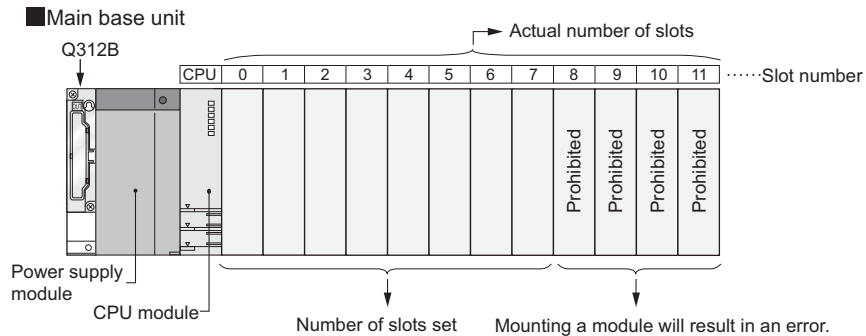


Figure 4.2 Setting the number of slots smaller than the actual one

4.1.2 Base unit assignment setting

Set base units on the I/O assignment tab of the PLC parameter dialog box.

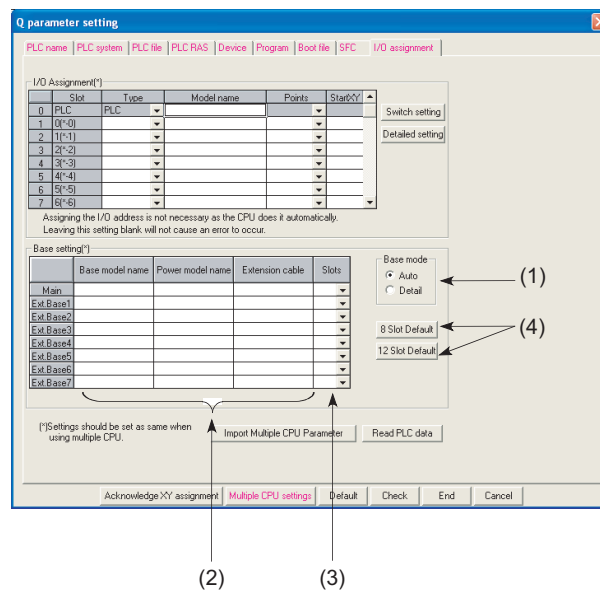


Figure 4.3 I/O assignment setting

(1) Auto/Detail

Select the mode for the base unit assignment either from auto mode or detail mode.

(2) Base model name, Power model name, and Extension cable

Enter the model names of mounted base units, power supply modules, and extension cables to be used within 16 characters for user reference or when printing out parameters.

CPU modules do not use the entered model names.

(3) Slots

When "Detail" is set, select the number of slots on the base unit to use from the following.

2 (2 slots), 3 (3 slots), 5 (5 slots), 8 (8 slots), 10 (10 slots), or 12 (12 slots)

(4) 8 Slot Default/12 Slot Default

When "Detail" is set, select either of these items for batch-setting the base units to the specified number of slots.

Point

- In auto mode, when any extension base number is skipped at the setting using the base number setting connector, an empty extension base cannot be reserved.
To reserve empty extension bases for future extension, select detail mode.
- In detail mode, set the number of slots to all base units used.
Failure to do so may result in incorrect I/O assignment setting.

4.2 I/O Number Assignment

The I/O number indicates addresses used for sequence programs in the following cases.

- Input of on/off data to the CPU module
- Output of on/off data from the CPU module to the external device

(1) Input and output of on/off data

The input (X) is used to input on/off data to the CPU module, and the output (Y) is used to output on/off data from the CPU module.

(2) I/O number representation

The I/O numbers are represented in hexadecimal.

When a 16-point I/O module is used, the I/O number for each slot will be 16 point-sequence number from $\square\square 0$ to $\square\square F$ as shown in Figure 4.4.

"X" and "Y" is prefixed to the I/O number of input modules and the I/O number of output modules, respectively.

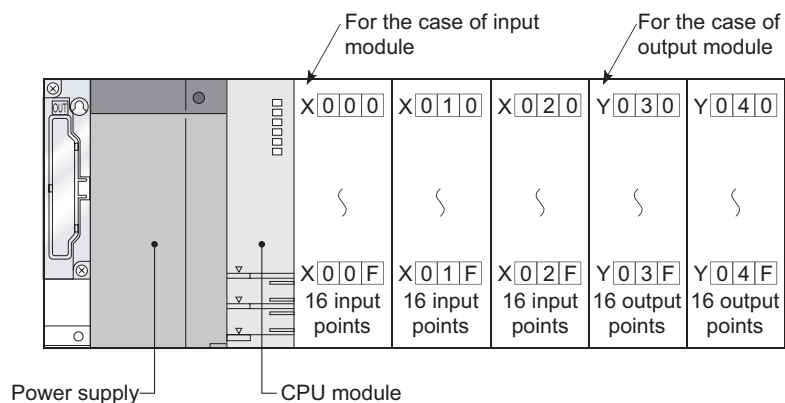


Figure 4.4 I/O numbers

4.2.1 Concept of I/O number assignment

The CPU module assigns I/O numbers at power on or reset, according to the I/O assignment setting.

(1) I/O number assignment

The Figure 4.5 shows an example of I/O number assignment to base units in the system where the CPU module is mounted on the main base unit.

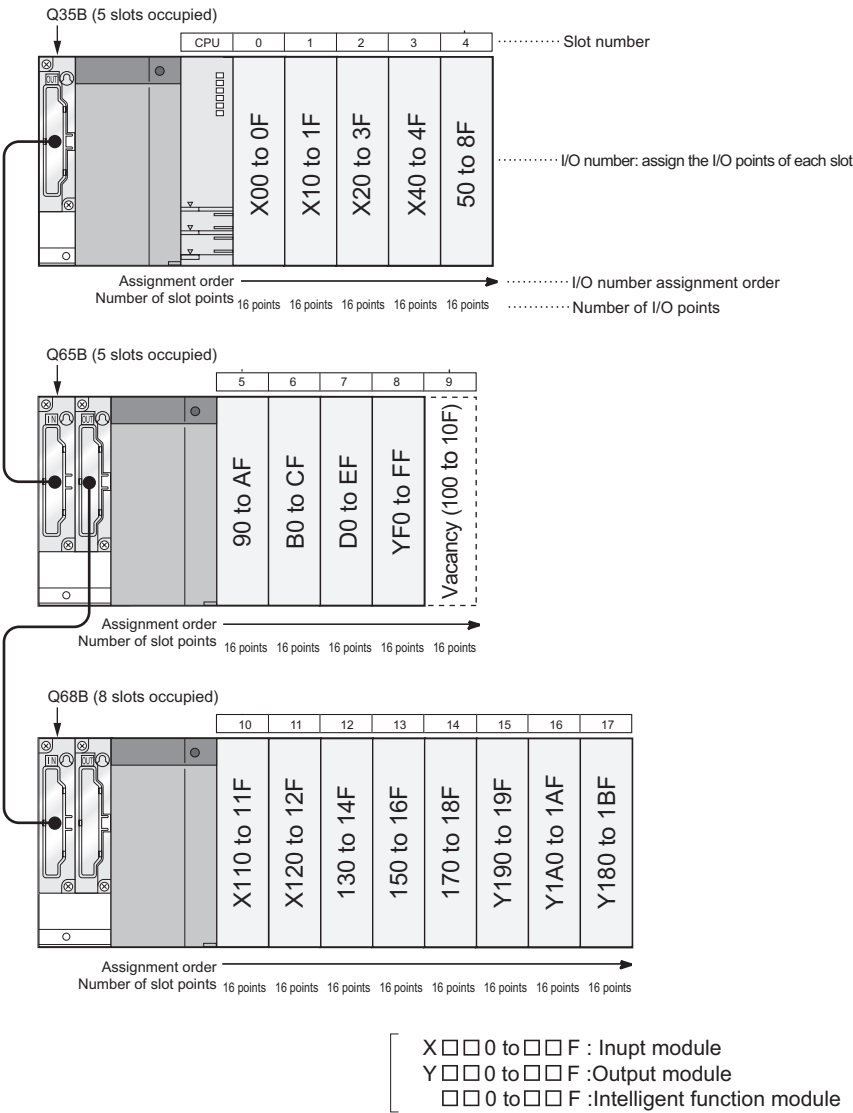


Figure 4.5 I/O number assignment example

(a) Assignment order

For the main base unit, the I/O numbers are assigned to the modules from left to right in a sequential order, starting from 0H assigned to the module on the right of the CPU module. For extension base units, the I/O numbers are continued from the last number of the I/O number of the main base unit.

(b) I/O number of each slot

Each slot on the base unit occupies I/O numbers by the number of I/O points of the mounted modules.

(2) I/O assignment on a remote I/O stations

The devices of input (X) and output (Y) in the CPU module can be assigned to I/O modules and intelligent function modules, which allows to control the modules in the remote I/O system such as CC-Link IE field network, MELSECNET/H remote I/O network and CC-Link.

Also, the devices of input (X) and output (Y) to be refreshed (on the CPU module side) can be used for the master/local modules of CC Link IE field network and the link I/O (LX, LY) of MELSECNET/H modules.

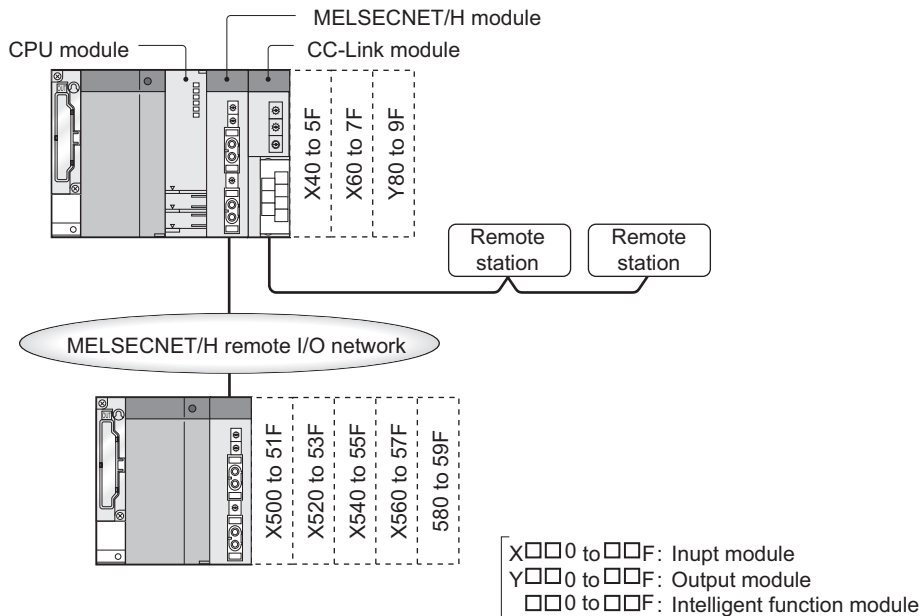


Figure 4.6 I/O number assignment on the remote station

(a) I/O numbers available on remote I/O stations

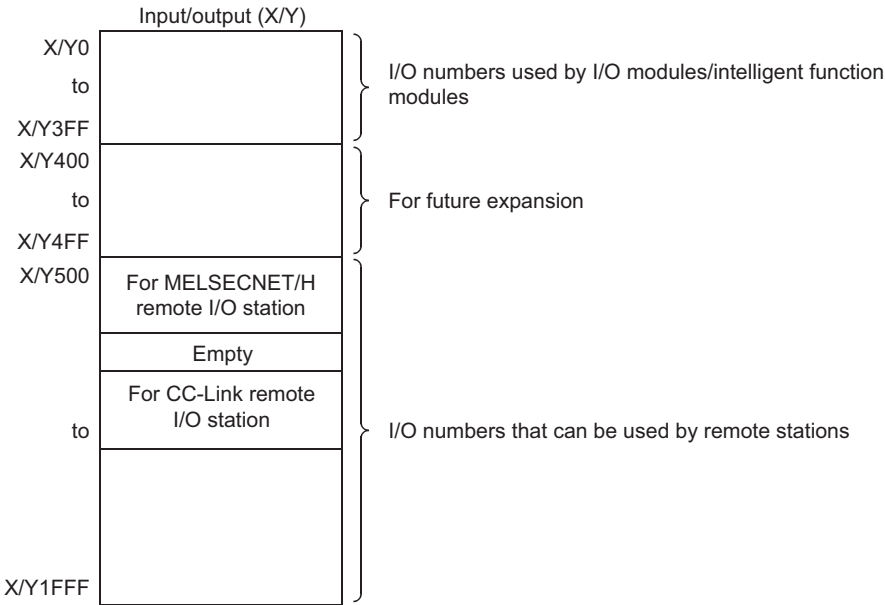
When the input (X) and output (Y) of the CPU module are used for the I/O numbers in the remote station, assign the I/O numbers later than those used for the I/O modules and intelligent function modules on the CPU module side.

Example When X/Y0 to X/Y3FF (1024 points) are used for the I/O modules and intelligent function modules on the CPU module side, X/Y400 and later can be used in the remote stations.

(b) Precautions for using remote station I/O numbers

1) Setting for future extension

When the input (X) and output (Y) of the CPU module are used for the I/O numbers on the remote station, consider future extension of I/O modules and/or intelligent function modules on the CPU module side.



When X/Y0 to 3FF (1024 points) are used by I/O modules and/or intelligent function modules and X/Y400 to 4FF (256 points) are secured for future extension

Figure 4.7 Assignment example for remote station of I/O number

2) When using CC-Link IE field network, MELSECNET/H and CC-Link are used

Do not duplicate the I/O numbers to be refreshed target (CPU module side device) of MELSECNET/H and CC-Link IE field network, as well as to the CC-Link remote I/O system station.

Point

- When network parameter setting has not been made in the CC-Link system, X/Y 1000 to 17FF (2048 points) are assigned to the CC-Link system master/local modules of lower numbers.
- There are no restrictions on the I/O number assignment order for the CC-Link IE field networks, MELSECNET/H remote I/O networks, CC-Link, or other networks.
- Free space can be provided in the area for the CC-Link IE field network remote station, MELSECNET/H remote I/O station, and CC-Link remote station.

4.2.2 Setting I/O numbers

Set the I/O number on the I/O assignment tab.

(1) Purpose of I/O number assignment

(a) Reserving points for future module changes

The number of points can be flexibly set so that the I/O number modification can be avoided when changing the current module to another in the future.

For example, 32 points can be assigned for future use to the slot where an input module with 16 points is currently mounted.

(b) Preventing I/O numbers from changing

The change in the I/O numbers can be prevented when an I/O module or intelligent function module, whose occupied I/O points are other than 16, is removed due to failure.

(c) Changing the I/O numbers to those used in the program

When the I/O numbers used in the actual system differ from those in the designed program, the I/O numbers of each module on the base unit can be changed to the ones in the designed program.

Point

- If any of the I/O modules whose number of I/O points are other than 16 fails without I/O assignment setting, the I/O numbers assigned following to the failed module may change, leading to a malfunction. For this reason, making the I/O assignment setting is recommended.
- I/O assignment setting allows the following settings as well.
 - Input response time (I/O response time) ( Section 6.7)
 - Error time output mode ( Section 6.8)
 - CPU module operation during a hardware error of intelligent function modules ( Section 6.9)
 - Switch setting of intelligent function modules and interrupt modules ( Section 6.10)

The I/O assignment is required for the input response time and switch settings.

(2) I/O assignment

The I/O assignment is set on the I/O assignment tab of the PLC parameter dialog box.
On the I/O assignment tab, the following items can be set for each slot on the base unit.

- "Type" (module type)
- "Points" (I/O points)
- "Start XY" (start I/O number)

For example, to change the I/O number of the specified slot, setting is allowed only to the number of points.
For other items that are not set, settings are completed based on the installation status of the base unit.

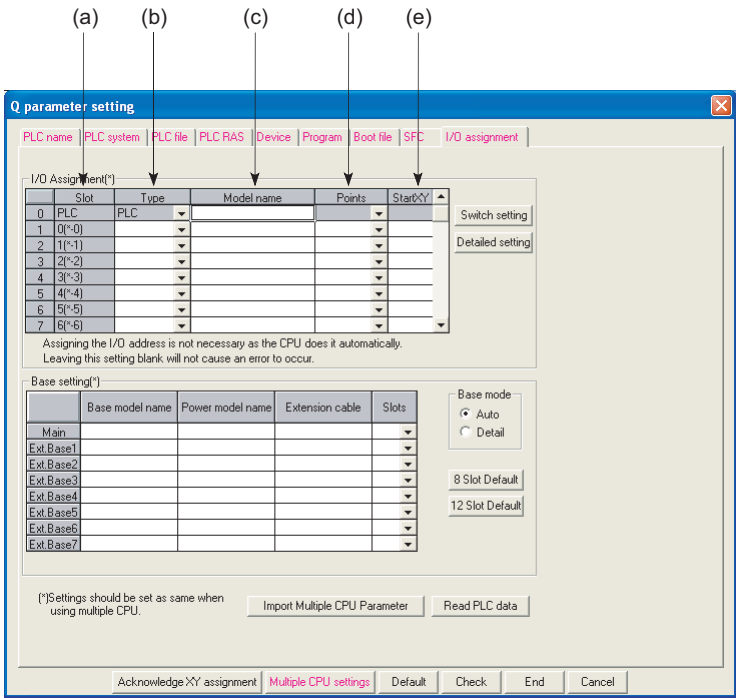


Figure 4.8 I/O assignment

(a) Slot

The slot number and location of the slot are displayed.
When the base unit is set in Auto mode, the base unit number is indicated in "**", and the slot number is counted from slot 0 of the main base unit.

(b) Type

Select the type of the mounted module from the followings:

- Empty (empty slot)
- Input (input module)
- Hi input (high-speed input module)
- Output (output module)
- I/O Mix (I/O combined module)
- Intelli. (intelligent function module)
- Interrupt (interrupt module)

If the type is not specified, the type of the actually mounted module is used.

(c) Type

Enter the model names of mounted modules within 16 characters.

CPU modules do not use entered model names. (Use the entered model names for user reference.)

(d) Points

When changing the number of I/O points for each slot, the selections are as follows.

0 points, 16 points, 32 points, 48 points, 64 points, 128 points, 256 points, 512 points, 1024 points

If the number of points is not selected, the points of the actually mounted module is used.

For empty slots, the points that are set on the PLC system tab of the PLC parameter dialog box is assigned.

(Default: 16 points)

(e) Start XY

When changing the I/O number for each slot, enter a new start I/O number.

If start XY is not specified for a slot, the I/O number continuing from the last number of the current setting is assigned.

(3) Precautions

(a) Type setting

The type set to the I/O assignment tab must be the same as that of the mounted module.

Setting a different type may cause incorrect operation.

For the intelligent function module, the I/O points must also be the same in addition to the I/O assignment setting.

Table4.1 shows the operations when the mounted module type differs from the one in the I/O assignment tab.

Table4.1 Incorrect operation when the module type differs

Mounted module	I/O assignment setting	Result
Input module, output module, I/O combined module	• Intelli. • Interrupt	Error (SP.UNIT.LAY.ERR.)
Intelligent function module	• Input • Hi. input • Output • I/O combined	Error (SP.UNIT.LAY.ERR.)
Empty slot	• Input • Hi. input • Output • I/O combined • Intelli. • Interrupt	Empty slot
All modules	• Empty	Empty slot
Other combinations	-	Error does not occur but incorrect operation may be caused. Or, error (PARAMETER ERROR (error code: 3000)).

(b) I/O points of slots

The number of I/O points for each slot selected in the I/O assignment tab is set in priority to those of mounted modules.

1) When the preset number of I/O points is less than those of mounted I/O modules

The available points for the mounted I/O module will be decreased.

For example, when the number of I/O points is set to 16 points in the I/O assignment setting of PLC parameter to the slot where a 32-point input module is mounted, the second half 16 points of the 32-point input module becomes unavailable.

2) When the preset number of I/O points exceeds those of mounted I/O modules

The exceeded number of points will not be used in I/O modules.

3) Last I/O number

Set the last I/O number within the I/O point range.

Failure to do so causes an error ("SP. UNIT LAY ERR."). ("****" is displayed as an I/O address on the System monitor screen of GX Developer.)

4) When setting 0 points in empty slots

Setting "Empty" and "0" to type and points respectively even occupies one slot.

To set any slots unoccupied after the specific slot number, set the number of slots in detail mode. (☞
Section 4.1.1)

(c) Start XY setting

When the start XY has not been entered, the CPU module automatically assigns it. The CPU module automatically assigns the start XY if it is not set. For this reason, the start XY setting of each slot may be duplicated with the one assigned by the CPU module in the case of 1) or 2) below.

- 1) Start XY values are not in the correct order.
- 2) Slots with and without the start XY setting (automatically assigned slot) are mixed

An example of start XY duplication is given in Figure 4.9 below.

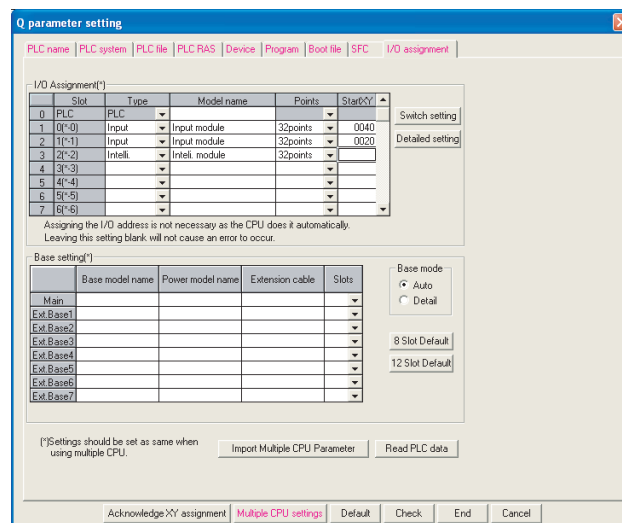


Figure 4.9 I/O assignment setting with start XY duplication

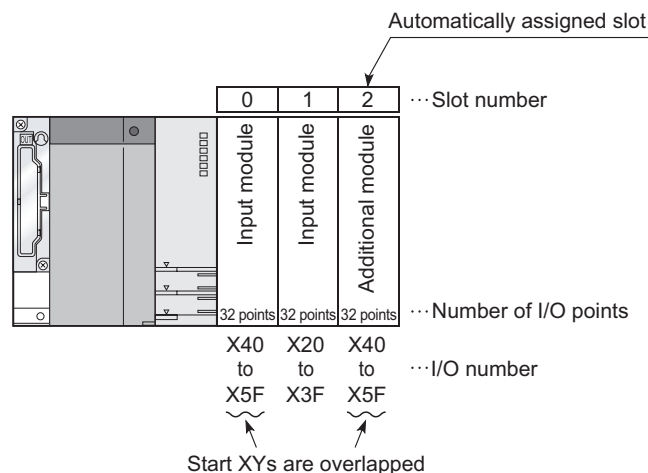


Figure 4.10 Start XY set by above (Figure 4.9) I/O assignment

Do not set duplicated start XY for each slot.

Specify start XY in the additional module to prevent the duplication of start XY. (Example: Input "0060" to "start XY" in slot2.)

Duplication of start XY will result in "SP. UNIT LAY ERR." (An error occurs even the module is not mounted.)

4.2.3 I/O number setting example

I/O number setting examples are provided as follows.

(1) Changing the number of points of an empty slot from 16 to 32

Reserve 32 points for the currently empty slot (Slot 3) so that the I/O numbers of Slot No. 4 and later do not change when a 32-point input module is mounted there in the future.

(a) System configuration and I/O number assignment before I/O assignment using GX

Developer

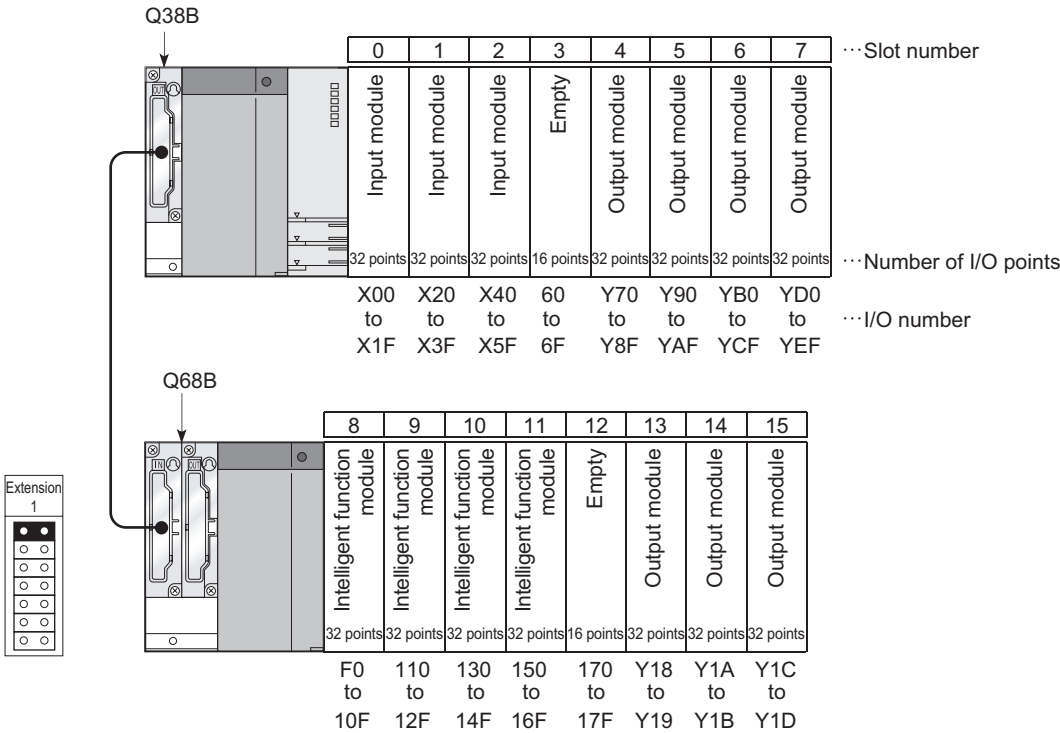


Figure 4.11 I/O number assignment (Before changing points of the empty slot)

(b) I/O assignment

Select "32 points" for the number of I/O points of Slot 3 in the I/O assignment setting of PLC parameter in GX Developer.

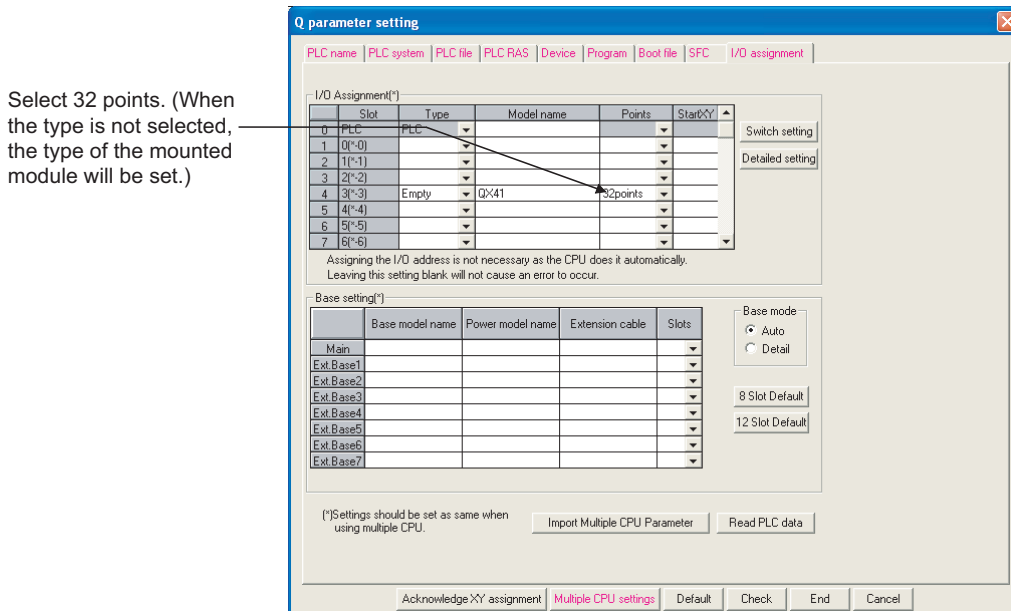


Figure 4.12 I/O assignment setting (When changing points of Slot 3)

(c) I/O number assignment after the I/O assignment using GX Developer

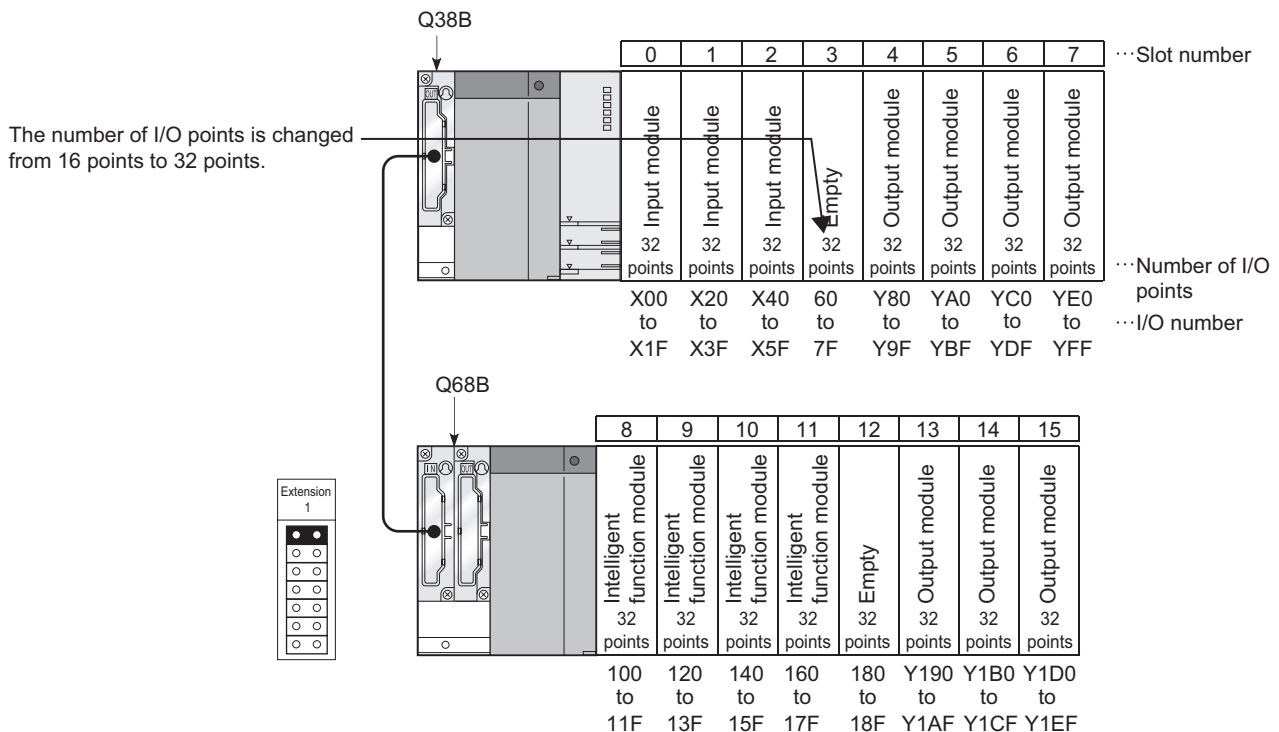


Figure 4.13 I/O numbers after I/O assignment (After changing points of the empty slot)

(2) Changing the I/O number of an empty slot

Change the I/O number of the currently empty slot (Slot 3) to X200 through 21F so that the I/O numbers of Slot 4 and later do not change when a 32-point input module is mounted there in the future.

(a) System configuration and I/O number assignment before the I/O assignment using GX Developer

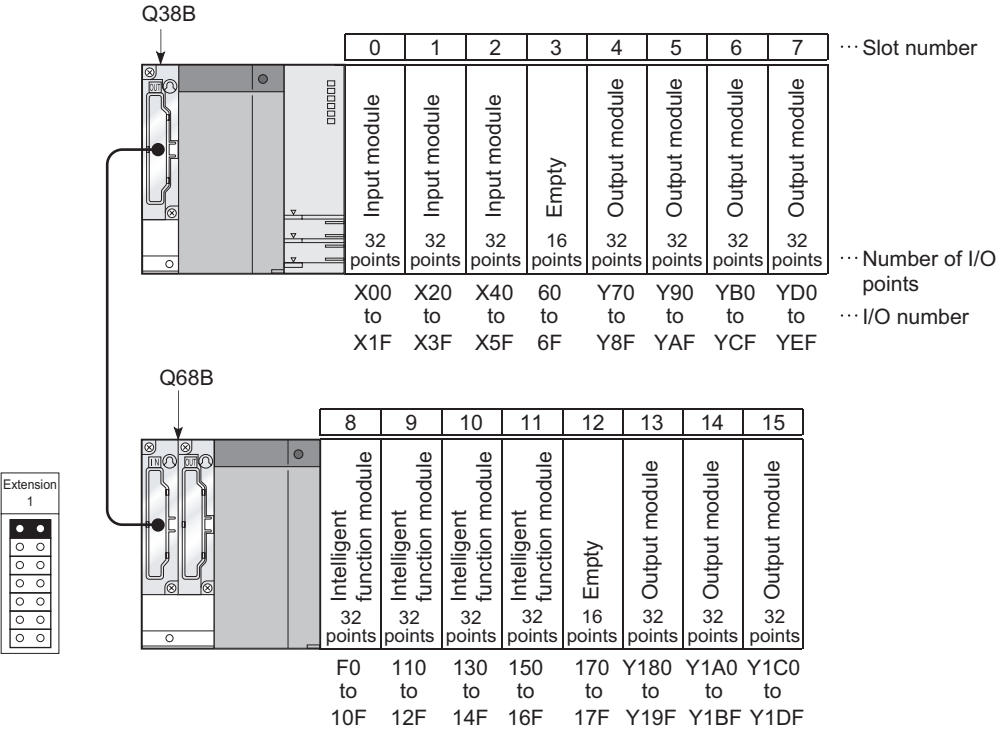


Figure 4.14 I/O number assignment (Before changing slot I/O numbers)

(b) I/O assignment

Set "200" for the start XY of Slot 3 and "70" to Slot 4 in the I/O assignment setting of PLC parameter in GX Developer.

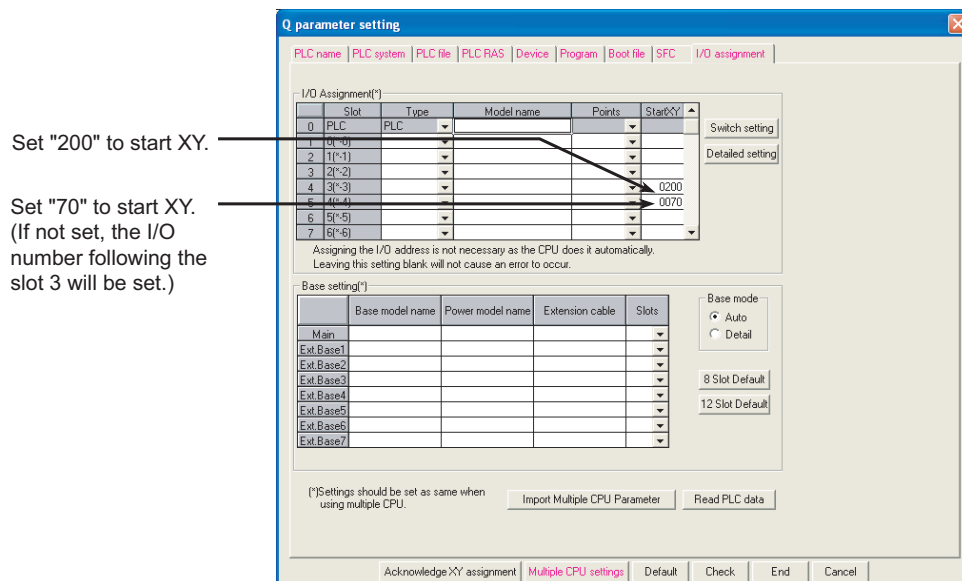


Figure 4.15 I/O assignment setting (When changing I/O numbers of Slot 3)

(c) I/O number assignment after the I/O assignment using GX Developer

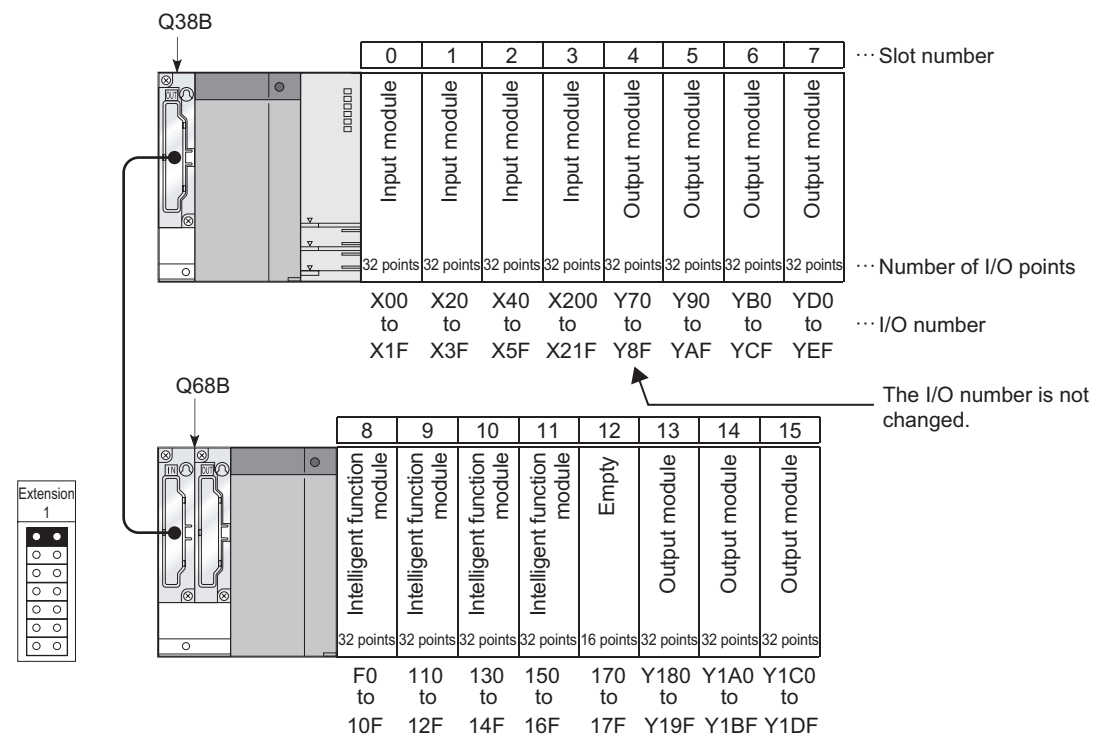


Figure 4.16 I/O number assignment (After changing slot I/O numbers)

4.2.4 Checking I/O numbers

Information of mounted modules and their I/O numbers can be checked on the System monitor screen of GX Developer. (☞ Section 6.20)

CHAPTER5 MEMORIES AND FILES USED FOR CPU MODULE

5.1 Memories Used for CPU Module

5.1.1 Memory composition and storable data

This section describes the memories used for the Universal model QCPU and data that can be stored in the memories.

(1) Memory composition

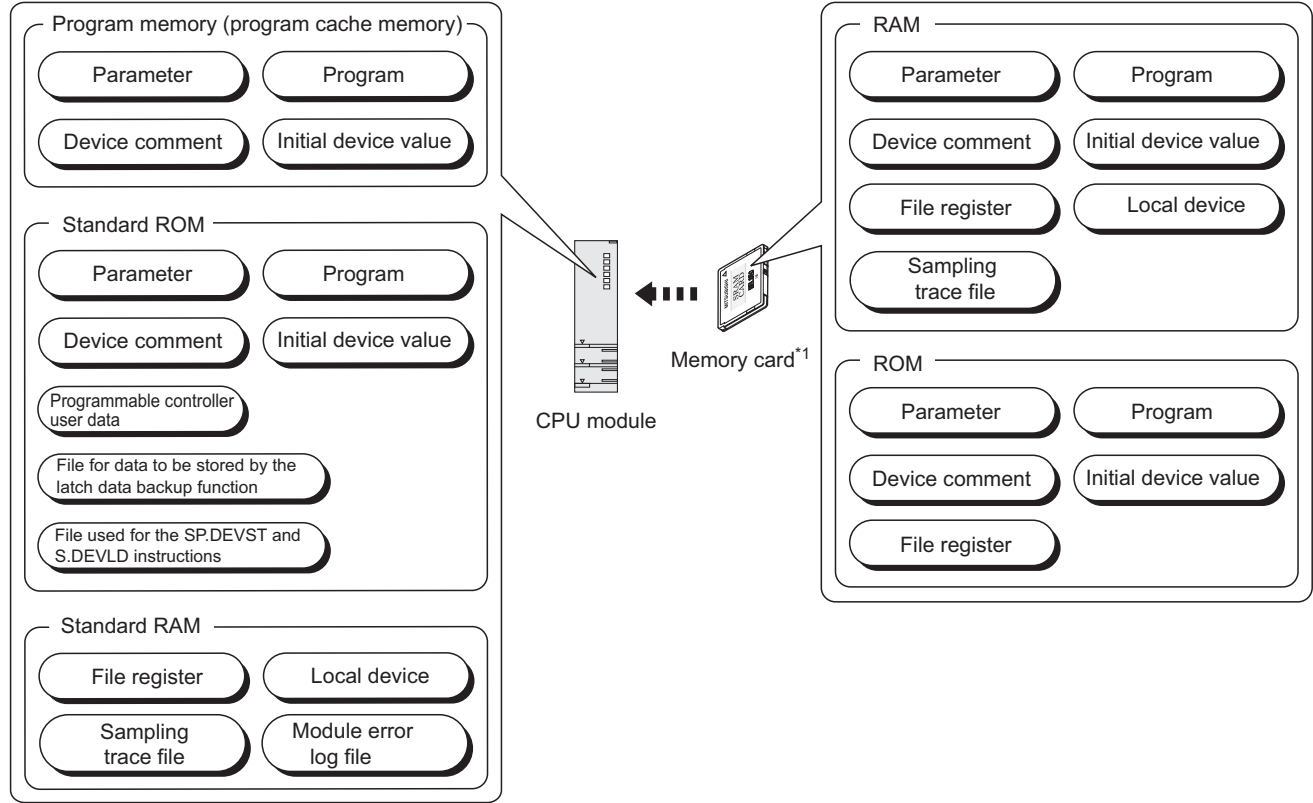


Figure 5.1 Memory composition of the Universal model QCPU

*1: The Q00UJCPU, Q00UCPU, and Q01UCPU cannot be used with a memory card.

*2: The Q00UJCPU does not have the standard RAM.

(a) Program memory (👉 Section 5.1.2)

This memory is for storing programs and parameters for CPU module operation.

The CPU module transfers a program from the program memory to the program cache memory for operation.

(👉 Section 5.1.3)

(b) Standard ROM (👉 Section 5.1.4)

This memory is for storing data such as device comments and PLC user data.

(c) Standard RAM (👉 Section 5.1.5)

This memory is for using file registers, local devices, and sampling trace files without a memory card.

In addition, the module error collection files are also stored.

(d) Memory card (👉 Section 5.1.6)

This memory is for expansion of built-in memory in the CPU module.

Three types of memory cards are available: SRAM card, Flash card, and ATA card.

(2) Data that can be stored in each memory

Table5.1 provides the data that can be stored in each memory.

Table5.1 Data that can be stored in each memory

Item	CPU module built-in memory			Memory card (RAM)	Memory card (ROM)		File name and extension	Remarks
	Program memory	Standard RAM	Standard ROM	SRAM card	Flash card	ATA card		
	Drive 0 *1	Drive 3 *1	Drive 4 *1	Drive 1 *1	Drive 2 *1			
Parameter	○	×	○	○	○	○	PARAM.QPA	1 data/drive
Intelligent function module parameter*2	○	×	○	○	○	○	IPARAM.QPA	1 data/drive
Program	◎	×	○ *3	○ *4	○ *4	○ *4	***.QPG	-
Device comment	○ *5	×	○ *6	○ *6	○ *6	○ *6	***.QCD	-
Initial device value	○	×	○	○	○	○	***.QDI	-
Device data	×	×	○	×	×	×	***.QST	-
File register	×	○ *7*8	×	○	○ *9	×	***.QDR	-
Local device	×	○ *7	×	○	×	×	***.QDL	1 data/CPU module
Sampling trace file	×	○ *7	×	○	×	×	***.QTD	-
Error history data	×	×	×	×	×	×	***.QFD	-
Device data storage file	×	×	○	×	×	×	DEVSTORE.QST	-
Module error collection file	×	○	×	×	×	×	IERRLOG.QIE	-
Backup data file	×	×	×	○	○	○	MEMBKUP0.QBP	-
Programmable controller user data	×	×	○	×	×	○ *10	***.***	-
Label program*12	○	×	○	○	○	○	PROJINFO.CAB*13	-
User setting system area*11	○	×	×	×	×	×	-	-

◎ : Required, ○ : Storable, × : Not storable

*1: A drive number is used to specify a memory to be written/read by the external device using a sequence program or MC protocol.

Since the memory name is used to specify the target memory by GX Developer, 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 dialog box.

*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/or sampling trace file can be stored in the standard RAM.

*8: For the number of storable file register points, refer to Section 9.7.

*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. (☞ Section 5.1.2(2)(b))

*12: Data that stores the label program configuration (For GX Works2, "source information" is described instead.)

For the label program, refer to the following.

☞ GX Developer Version8 Operating Manual

*13: For GX Works2, SRCINFOM.CAB (simple project) and SRCINFOI.CAB (structured project) are described instead.

(3) Memory capacities and necessity of formatting

Table5.2 provides the memory capacities and necessity of formatting of each memory.

Format a memory requiring formatting by GX Developer beforehand.

Table5.2 Memory capacities and necessity of formatting

		Q00UJCPU	Q00UCPU	Q01UCPU	Q02UCPU	Q03UDCPU, Q03UDECPU	Q04UDHCPU, Q04UDEHCPU	Q06UDHCPU, Q06UDEHCPU	Formatting
Program memory		40K bytes (10K steps)	40K bytes (10K steps)	60K bytes (15K steps)	80K bytes (20K steps)	120K bytes (30K steps)	160K bytes (40K steps)	240K bytes (60K steps)	Necessary
Standard ROM		256K bytes	512K bytes	512K bytes	512K bytes	1024K bytes	1024K bytes	1024K bytes	Unnecessary
Standard RAM		-	128K bytes	128K bytes	128K bytes	192K bytes	256K bytes	768K bytes	Necessary ^{*1}
Memory card	SRAM card	-			Q2MEM-1MBS: 1M byte Q2MEM-2MBS: 2M bytes Q3MEM-4MBS: 4M bytes Q3MEM-8MBS: 8M bytes				Necessary (Use GX Developer.)
	Flash card	-			Q2MEM-2MBF: 2M bytes Q2MEM-4MBF: 4M bytes				Unnecessary
	ATA card	-			Q2MEM-8MBA: 8M bytes Q2MEM-16MBA: 16M bytes Q2MEM-32MBA: 32M bytes				Necessary (Use GX Developer.)

		Q10UDHCPU, Q10UDEHCPU	Q13UDHCPU, Q13UDEHCPU	Q20UDHCPU, Q20UDEHCPU	Q26UDHCPU, Q26UDEHCPU	Q50UDEHCPU	Q100UDEHCPU	Formatting
Program memory		400K bytes (100K steps)	520K bytes (130K steps)	800K bytes (200K steps)	1040K bytes (260K steps)	2000K bytes (500K steps)	4000K bytes (1000K steps)	Necessary
Standard ROM		2048K bytes	2048K bytes	4096K bytes	4096K bytes	8192K bytes	16384K bytes	Unnecessary
Standard RAM		1024K bytes	1024K bytes	1280K bytes	1280K bytes	1536K bytes	1792K bytes	Necessary ^{*1}
Memory card	SRAM card				Q2MEM-1MBS: 1M byte Q2MEM-2MBS: 2M bytes Q3MEM-4MBS: 4M bytes Q3MEM-8MBS: 8M bytes			Necessary (Use GX Developer.)
	Flash card				Q2MEM-2MBF: 2M bytes Q2MEM-4MBF: 4M bytes			Unnecessary
	ATA card				Q2MEM-8MBA: 8M bytes Q2MEM-16MBA: 16M bytes Q2MEM-32MBA: 32M bytes			Necessary (Use GX Developer.)

^{*1}: When the memory contents become indefinite in initial status or due to the end of battery life, the memory is automatically formatted after the CPU module is powered off and then on or is reset.

Point

- When files are written to each memory, the unit of stored file size depends on the target CPU module and memory area. (Section 5.3.4)
- In memory capacity calculation, 1 step is equal to 4 bytes.

5.1.2 Program memory

(1) Definition

This memory is for storing programs and parameters for CPU module operation.

The CPU module transfers a program from the program memory to the program cache memory for operation. (☞ Section 5.1.3)

Point

If the total size of data to be stored exceeds the program memory capacity:

- reduce the user setting system area, or
- transfer data other than programs to the standard ROM or memory card.

(2) Before using the program memory

Format the program memory by GX Developer.

(a) Formatting

Select [Online] → [Format PLC memory] in GX Developer.

Select "Program memory/Device memory" in "Target memory".

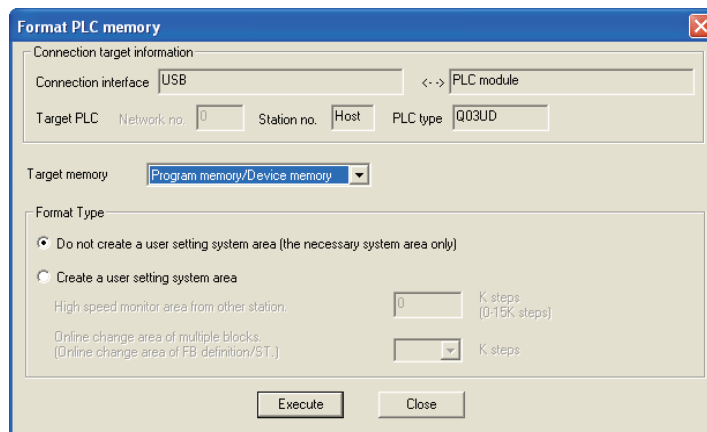


Figure 5.2 Formatting the program memory

(b) Creating a user setting system area

When formatting a program memory, set the capacity of user setting system area.

1) Do not create a user setting system area (the necessary system area only)

The user setting system area is not created during formatting.

2) Create a user setting system area

The user setting system area is created during formatting.

Table 5.3 provides the type of user setting system area.

Table 5.3 Type of user setting system area

System area	Description
High speed monitor area from other station.	Setting this area increases the monitoring speed in the GX Developer connected to such as a serial communication module. When using RS-232 and USB together with GX Developer, this area is used to register monitor data from GX Developer connected to such as a serial communication module.

Point

When a user setting system area is created, the available area reduces by the number of steps created in the area.

(c) Checking the memory capacity after formatting

Select [Online] → [Read from PLC] in GX Developer.

- 1) Select "Program memory/Device memory" in "Target memory" on the Read from PLC screen.
- 2) Click the **Free space volume** button.
- 3) The memory capacity appears in "Total free space volume".

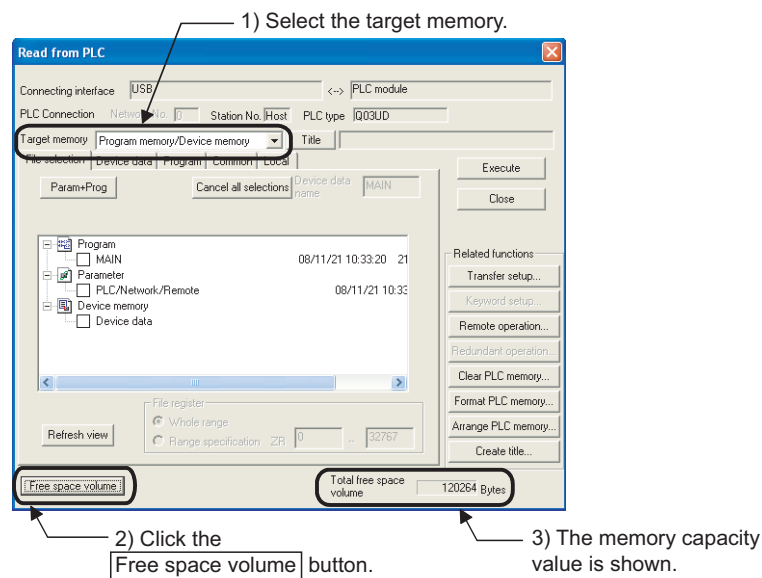


Figure 5.3 Procedure for checking the memory capacity

(3) Writing to the program memory

Select [Online] → [Write to PLC] in GX Developer.

Select "Program memory/Device memory" in "Target memory" on the Write to PLC screen.

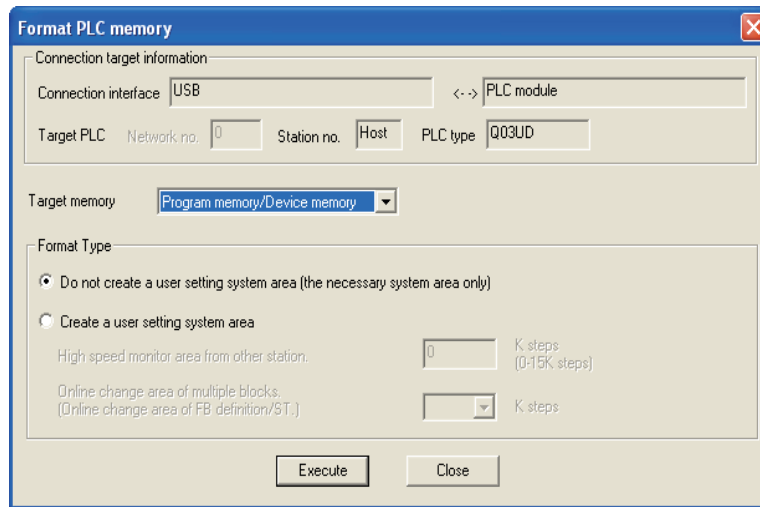


Figure 5.4 Write to PLC screen

Point

The file size has its minimum unit. (☞ Section 5.3.4)

The occupied memory capacity may be greater than the actual file size.

Note that as the number of files increases, the difference between the occupied memory capacity and the actual file size increases.

5.1.3 Program cache memory

(1) Definition

This memory is for program operation.

The CPU module transfers a program from the program memory to the program cache memory for operation.

The program is transferred during:

- initial processing after the CPU module is powered on, or
- initial processing after the CPU module is reset.

Figure 5.5 provides the flow of program operations.

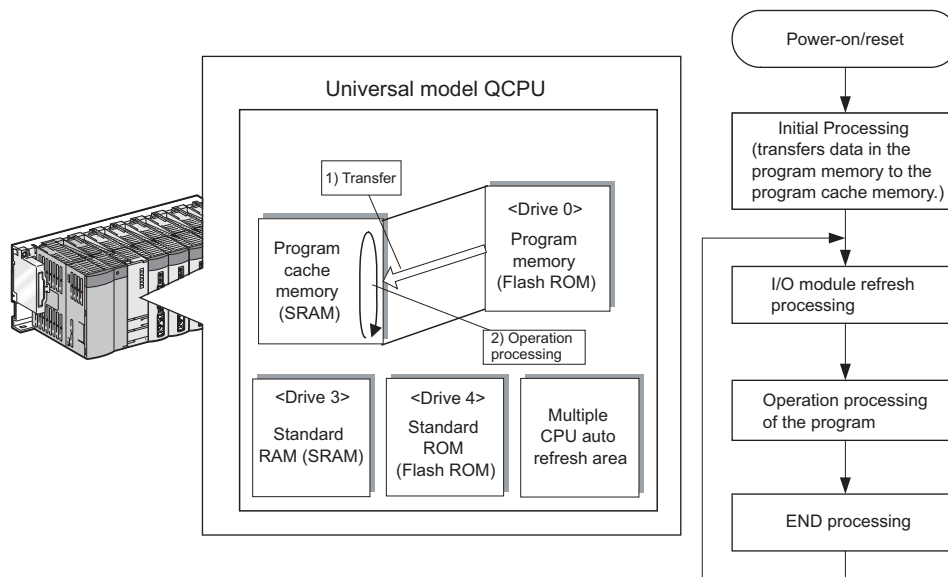


Figure 5.5 Flow of program operation

(2) Writing a program

When writing data from GX Developer, programs and parameters are written to the program cache memory in the CPU module. After the completion of the writing, the data are transferred to the program memory.

Figure 5.6 provides the flow of writing a program.

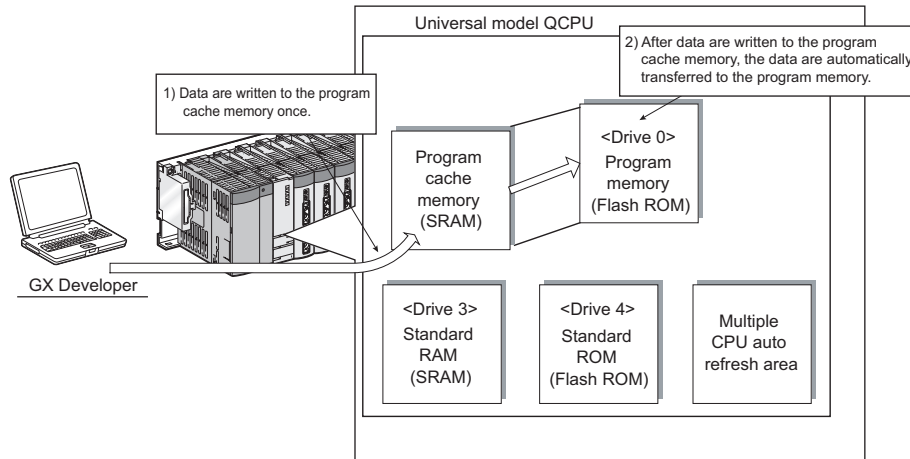


Figure 5.6 Flow of writing a program

(3) Transferring data to the program memory by GX Developer

Data can also be batch-transferred from the program cache memory to the program memory by selecting [Online] → [Program memory batch transfer] in GX Developer.

Figure 5.4 provides the transfer time required when data are batch-transferred to the program memory while the CPU module is in the RUN status.

Table 5.4 Transfer time of program memory batch-transfer

CPU module	Transfer time
Q00UJCPU, Q00UCPU, Q01UCPU, Q02UCPU	$T_s \times 320 + 4.8 \text{ (s)}$
Q03UDCPU, Q04UDHCPU, Q06UDHCPU, Q03UDECPU, Q04UDEHCPU, Q06UDEHCPU	$T_s \times 260 + 4.7 \text{ (s)}$
Q10UDHCPU, Q10UDEHCPU	$T_s \times 439 + 6.2 \text{ (s)}$
Q13UDHCPU, Q13UDEHCPU	$T_s \times 600 + 8.0 \text{ (s)}$
Q20UDHCPU, Q20UDEHCPU	$T_s \times 839 + 11.4 \text{ (s)}$
Q26UDHCPU, Q26UDEHCPU	$T_s \times 1100 + 15.0 \text{ (s)}$
Q50UDEHCPU	$T_s \times 2450 + 17.0 \text{ (s)}$
Q100UDEHCPU	$T_s \times 4550 + 9.0 \text{ (s)}$

T_s : scan time (s)

(4) Checking status during data transfer to the program memory

The status during data transfer to the program memory can be checked either in the progress screen of GX Developer or by the special relay and special register.

(a) Checking the status in the progress screen

Figure 5.7 provides the progress screen of GX Developer.

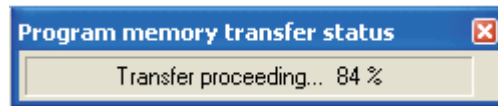


Figure 5.7 Screen showing status of data transfer to the program memory

(b) Checking the status by the special relay and special register

The status can be checked by SM681 and SD681.

(5) Checking whether data are transferred to the program memory Note5.1

Whether data are transferred from the program cache memory to the program memory can be checked by SM165.



Note5.1

Universal

When checking the transfer status to the program memory in the Q02UCPU, Q03UDCPU, Q04UDHCPU, or Q06UDHCPU, check the versions of the CPU module and GX Developer. ( Appendix 2)

5.1.4 Standard ROM

(1) Definition

This memory is for storing data such as parameters and programs.

(2) Checking the memory capacity

Select [Online] → [Read from PLC] in GX Developer.

- 1) Select "Standard ROM" in "Target memory" on the Read from PLC screen.
- 2) Click the **Free space volume** button.
- 3) The memory capacity appears in "Total free space volume".

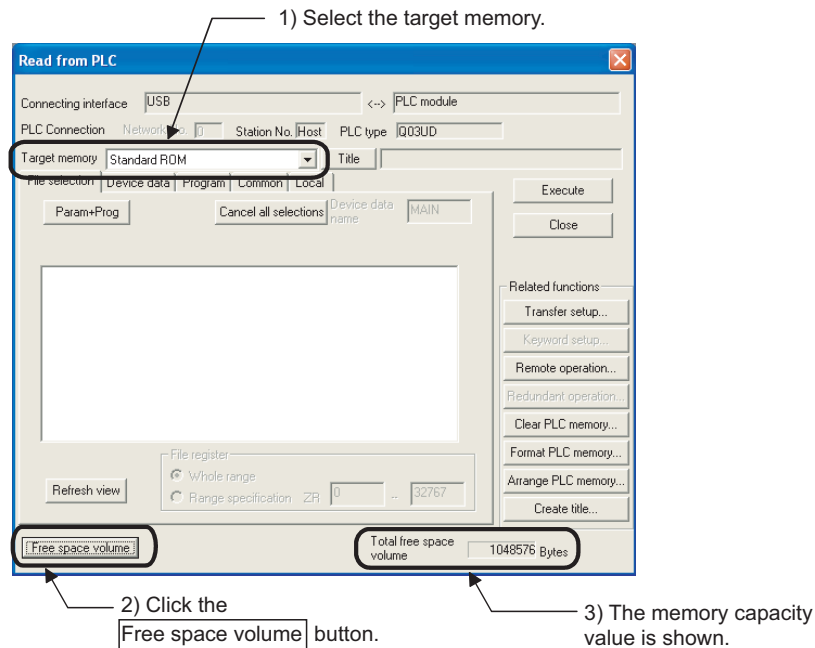


Figure 5.8 Procedure for checking the memory capacity

(3) Writing to the standard ROM

Select [Online] → [Write to PLC] in GX Developer.

Select "Standard ROM" in "Target memory".

Point

The file size has its minimum unit. (☞ Section 5.3.4)

The occupied memory capacity may be greater than the actual file size.

Note that as the number of files increases, the difference between the occupied memory capacity and the actual file size increases.

5.1.5 Standard RAM Note5.2

(1) Definition

This memory is for using file registers, local devices, and sampling trace files without a memory card.
Storing the file registers in the standard RAM allows fast access as data registers do.
The memory is also used for storing a module error collection file.

Point

If the size of files to be stored exceeds the standard RAM capacity:

- store the files in the memory card, or
- reduce the number of points of the file register, local device, or sampling trace.

Note when file registers are stored in the memory card, the access speed will be slower than when they are stored in the standard RAM.

(2) Before using the standard RAM

Format the standard RAM by GX Developer.

(a) Formatting

Select [Online] → [Format PLC memory] in GX Developer.
Select "Standard RAM" in "Target memory".

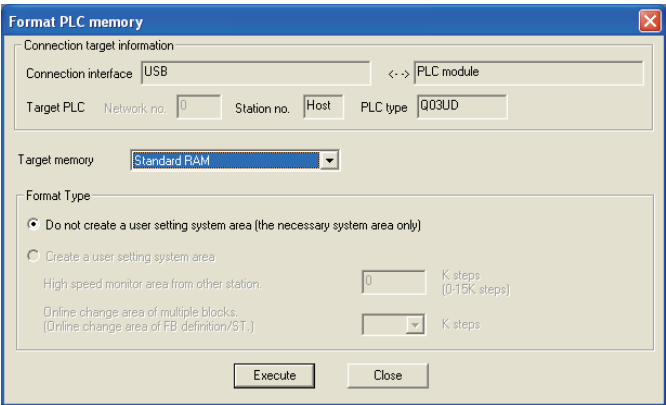


Figure 5.9 Formatting the standard RAM



Note5.2

Universal

The Q00UJCPU does not have the standard RAM.

(b) Checking the memory capacity after formatting

Select [Online] → [Read from PLC] in GX Developer.

- 1) Select "Standard RAM" in "Target memory" on the Read from PLC screen.
- 2) Click the **Free space volume** button.
- 3) The memory capacity appears in "Total free space volume".

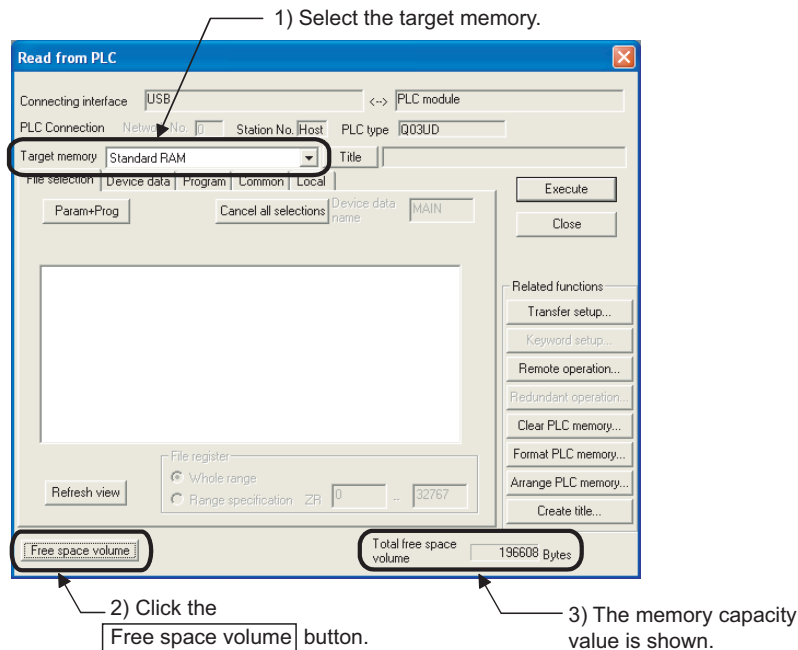


Figure 5.10 Procedure for checking the memory capacity

(3) Writing to the standard RAM

Select [Online] → [Write to PLC] in GX Developer.

Select "Standard RAM" in "Target memory" on the Write to PLC screen.

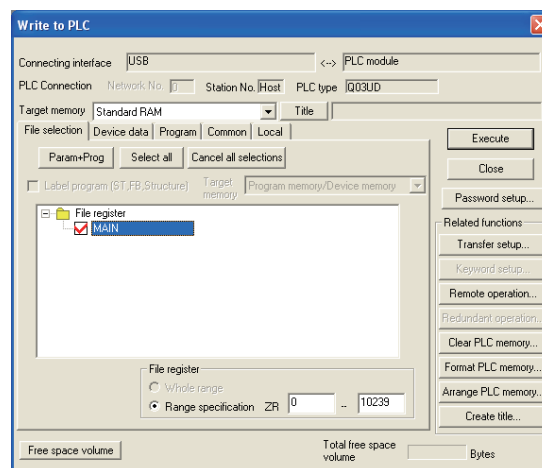


Figure 5.11 Write to PLC screen

Point

The file size has its minimum unit. (☞ Section 5.3.4)

The occupied memory capacity may be greater than the actual file size.

Note that as the number of files increases, the difference between the occupied memory capacity and the actual file size increases.

5.1.6 Memory card Note5.3

(1) Definition

This memory is for expansion of a memory in the CPU module.

The following three types are available:

- SRAM card
- Flash card
- ATA card

(a) 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 register points is greater than the standard RAM capacity, or
- the sampling trace function is used. ( Section 6.14)

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

(b) Flash card

Write data by GX Developer and read it by the sequence program. (Data can be read only by the sequence program.)

Use the Flash card when data are not changed.

File registers can be stored up to 2039K points.

(c) ATA card

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

With the file access instruction (such as the SP. FWRITE instruction) in the sequence program, access the programmable controller user data in the ATA card in CSV format/binary format.



Note5.3

Universal

The Q00UJCPU, Q00UCPU, and Q01UCPU cannot be used with a memory card.

(2) Before using the SRAM card or ATA card

Format the SRAM card or ATA card by GX Developer.

(a) Formatting

Select [Online] → [Format PLC memory] in GX Developer.

- When formatting the SRAM card, select "Memory card (RAM)" in "Target memory".
- When formatting the ATA card, select "Memory card (ROM)" in "Target memory".

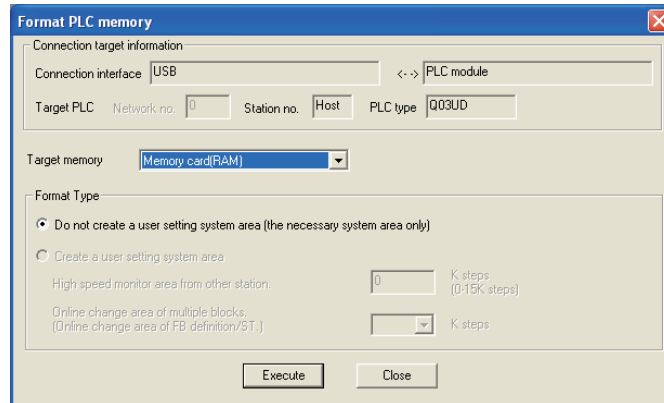


Figure 5.12 Formatting the SRAM card or ATA card

Point

- Use only GX Developer to format the ATA card.
If formatting the ATA card by such as the formatting function of Microsoft® Windows®, the card may not be used with the CPU module.
- When formatting the SRAM card or ATA card, the memory card information area is automatically secured. Therefore, the card capacity reduces by the area size.

Remark

Formatting is not required for the Flash card.

(b) Checking the memory capacity after formatting

Select [Online] → [Read from PLC] in GX Developer.

- 1) Select "Memory card (RAM)" or "Memory card (ROM)" in "Target memory" on the Read from PLC screen.
- 2) Click the **Free space volume** button.
- 3) The memory capacity appears in "Total free space volume".

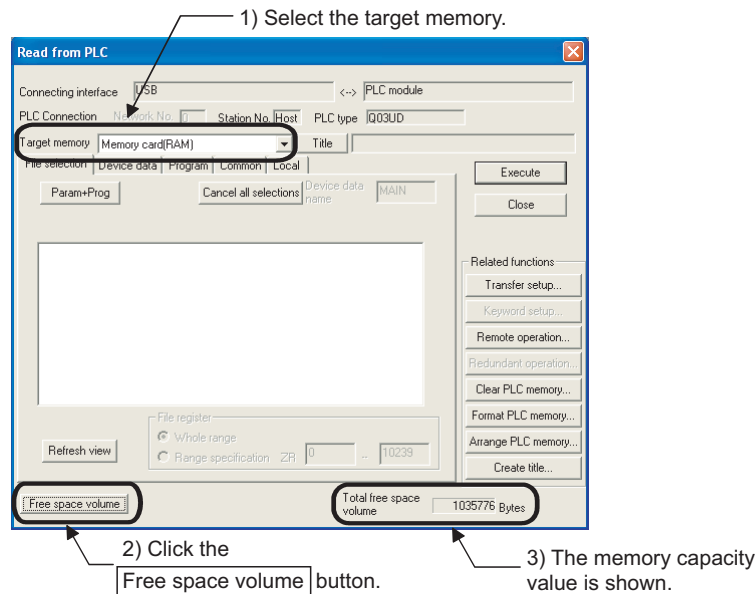


Figure 5.13 Procedure for checking the memory capacity

(3) Writing to the memory card

The following describes the operations before writing and the methods for writing.

(a) Writing to the SRAM card or the ATA card

Select [Online] → [Write to PLC] in GX Developer.

- When writing data to the SRAM card, select "Memory card (RAM)" in "Target memory" on the Write to PLC screen.
- When writing data to the ATA card, select "Memory card (ROM)" in "Target memory" on the Write to PLC screen.

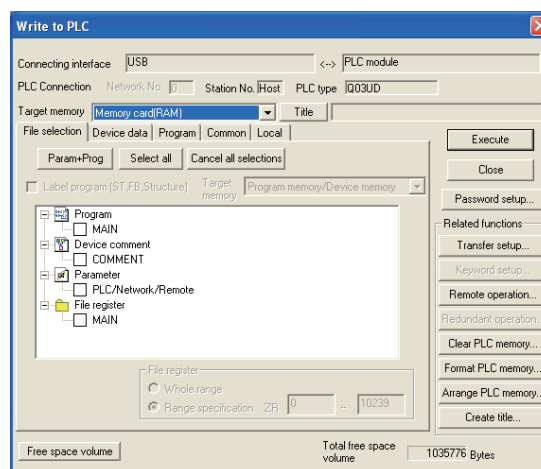


Figure 5.14 Write to PLC screen

(b) Writing to the Flash card

The following two methods are available.

- Writing by "Write the program memory to ROM" (☞ Section 5.1.7(1)(a))
- Writing by "Write to PLC (Flash ROM)" (☞ Section 5.1.7(1)(b))

Point

The file size has its minimum unit. (☞ Section 5.3.4)

The occupied memory capacity may be greater than the actual file size.

Note that as the number of files increases, the difference between the occupied memory capacity and the actual file size increases.

(4) How to use the program stored in the memory card

Boot the program stored in the memory card to the program memory before its execution. (☞ Section 5.1.8)

5.1.7 Writing to the Flash card by GX Developer

(1) Methods for writing data to the Flash card and applications

Figure 5.15 provides the methods for writing data to the Flash card.

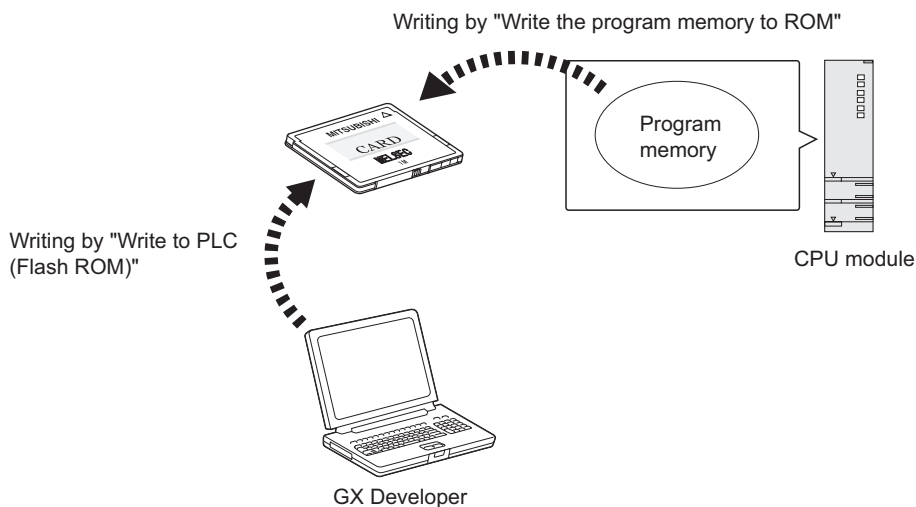


Figure 5.15 Methods for writing data to the Flash card

(a) Writing by “Write the program memory to ROM”

Data in the program memory are batch-written to the Flash card.

To batch-write the data, select [Online] → [Write to PLC (Flash ROM)] → [Write the program memory to ROM] in GX Developer.

Use this method when:

- debugging the program in the program memory and writing the debugged program to the Flash card without change for boot operation (☞ Section 5.1.8), or
- storing the data in the program memory to the Flash card without battery backup.

(b) Writing by “Write to PLC (Flash ROM)”

Files specified by GX Developer are batch-written to the Flash card.

To batch-write the files, select [Online] → [Write to PLC (Flash ROM)] → [Write to PLC (Flash ROM)] in GX Developer.

Use this method when:

- storing parameters, initial device values, or device comments whose size exceeds the program memory capacity to the Flash card, or
- using file registers stored in the Flash card.

(2) Writing to the Flash card

The following describes the operations before writing and the methods for writing.

(a) Before writing

Check the following.

1) Preparing files to be written

Writing a file to the Flash card automatically deletes all files stored in the Flash card.

Also write all files same as the stored files together.

2) Boot operation

When storing parameters to the Flash card at boot operation, make the boot file setting described in Section 5.1.8.

(b) Writing procedure

The following describes a procedure for writing a file to the Flash card.

1) Procedure for [Write the program memory to ROM] in GX Developer

- Select [Online] → [Write to PLC (Flash ROM)] → [Write the program memory to ROM].
- The program memory data into ROM screen appears.

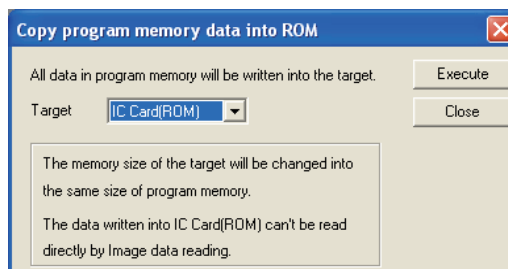


Figure 5.16 Copy program memory data into ROM screen

- Select the memory to be written in "Target" to write the program memory file to the Flash card.

Point

- When files are written by "Write the program memory to ROM", the capacity of used target memory is equal to that of used program memory.
To fully use the capacity of the target memory, write the files by "Write to PLC (Flash ROM)".
- When writing data that cannot be stored to the program memory (file register) to the Flash card, write it by "Write to PLC (Flash ROM)".

2) Procedure using [Write to PLC (Flash ROM)] in GX Developer

- Select [Online] → [Write to PLC (Flash ROM)] → [Write to PLC (Flash ROM)].
- The Write to PLC (Flash ROM) screen appears.

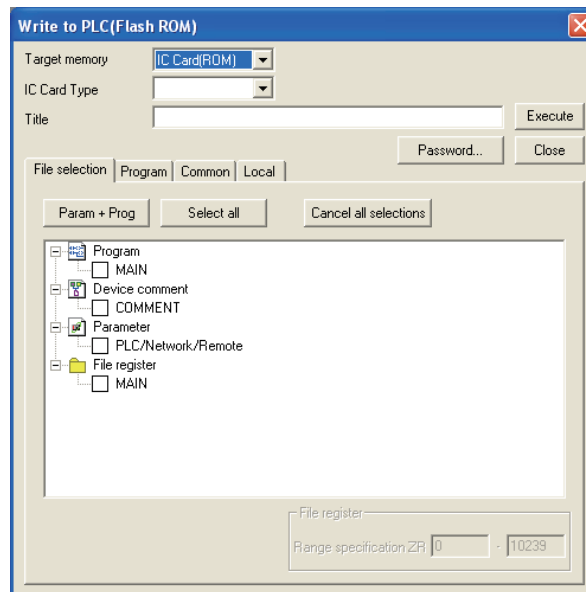


Figure 5.17 Write to PLC (Flash ROM) screen

- Select the target memory.
- Select a file to be written and write it to the Flash card.

(3) Adding or changing a file in the Flash card

Add or change the file by either of the following methods (stored files cannot be added or changed directly).

(a) When writing files using [Write the program memory to ROM] in GX Developer

- To read all files in the program memory, select [Online] → [Read from PLC].
- Add or change the read files.
- Write the added or changed files to the program memory.
- Select [Online] → [Write to PLC (Flash ROM)] → [Write the program memory to ROM] and write the files to the Flash card.

(b) When writing files using [Write to PLC (Flash ROM)] in GX Developer

- To read all files in the Flash card, select [Online] → [Read from PLC].
- Add or change the read files.
- Select [Online] → [Write to PLC (Flash ROM)] → [Write to PLC (Flash ROM)] and write the files to the Flash card.

(4) Precautions

(a) Setting the communication time check period in GX Developer

Since writing a file to the Flash card takes time, set "Check at communication time" in GX Developer to 60 seconds or longer.

If the set time is short, GX Developer may time out.

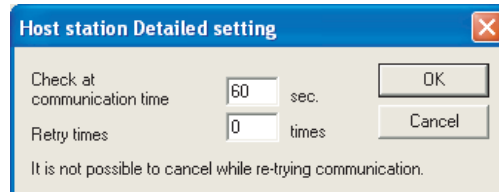


Figure 5.18 Check at communication time setting

(b) Writing a file from GX Developer in another station via CC-Link

Since writing a file to the Flash card takes time, make the CPU monitoring time setting (SW000A) of CC-Link to 60 seconds or longer.

(The default value of 90 seconds can be used.)

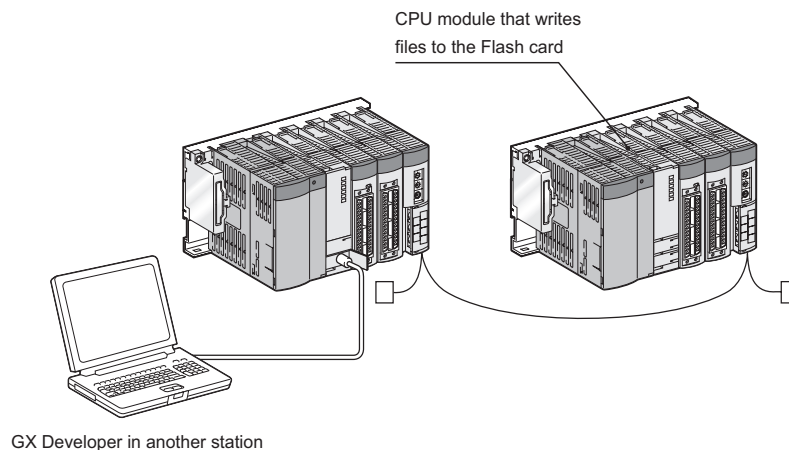


Figure 5.19 Writing a file from GX Developer in another station(via CC-Link)

Remark

For settings of the communication time check period and CPU monitoring time, refer to the following.

GX Developer Version8 Operating Manual

(c) Time required for “Write to PLC (Flash ROM)”

Using "Write to PLC (Flash ROM)" writes data to the entire space in the Flash card.

Therefore, even if a program having the small number of steps is written to the Flash card, the processing takes time.

(Writing using the Q2MEM-4MBF at a communication speed of 115.2Kbps with an RS-232 interface requires 14 minutes.)

When writing data to the Flash card, increase the transmission speed or use an USB.

Communication time takes time when "Write to PLC (Flash ROM)" is executed from another station.

(d) Online change (Only “Write to PLC (Flash ROM)” can be used.)

"Write to PLC (Flash ROM)" can be executed during the RUN status except the following cases.

Execute "Write to PLC (Flash ROM)" after setting the RUN/STOP/RESET switch to STOP when:

- 1) File registers stored in the Flash card are used in the sequence program, or
- 2) although "Not used" is set to the file registers by the PLC parameter dialog box, they are used in the sequence program.

If "Write to PLC (Flash ROM)" is executed during the RUN status in either above situations of 1) or 2), an error may occur, resulting the CPU module to stop.

(e) Writing/reading data during execution of “Write to PLC (Flash ROM)”

While "Write to PLC (Flash ROM)" is executed, data cannot be read from/written to other modules.

Therefore, time-out may occur in other modules.

(f) When “Write to PLC (Flash ROM)” is executed during the STOP status

Do not set the RUN/STOP/RESET switch to RUN during the writing.

The CPU module may not enter the RUN status normally during execution of "Write to PLC (Flash ROM)".

Set the RUN/STOP/RESET switch to RUN after the writing.

5.1.8 Operating the program in the memory card (boot operation)

This section describes methods for operating the program stored in the memory card.

(1) Operating the program in the memory card

To execute the boot operation, set the names of files to be booted in the Boot file tab of the PLC parameter dialog box. (☞ (3) in this section, (4))

The program stored in the memory card, whose file name is specified in the Boot file tab, is booted with the program memory after the CPU module is powered off and then on or is reset.

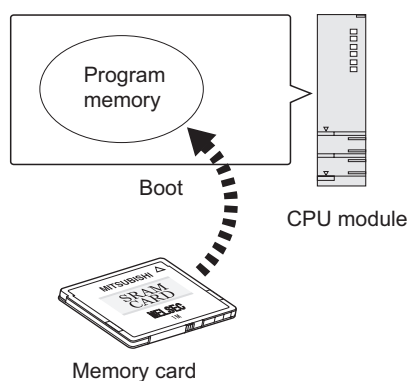


Figure 5.20 Boot operation

(2) Bootable files, transfer source, and transfer destination

Table Table5.5 provides the combinations of bootable file, transfer source, and transfer destination.

Table5.5 Combinations of bootable file, transfer source, and transfer destination

File name	Transfer source	Transfer destination	
		Program memory	Standard ROM
Parameter	Memory card	○	○
Sequence program		○	×
Device comment		○	○
Initial device value		○	○
Label program		○	○

○ : Bootable, × : Cannot be booted.

(3) Procedure before boot operation

The following explains the procedures to store the files to be booted in the memory card and then start boot operation.

(a) Creating a program

Create a program.

(b) Boot file setting

Set the names of files to be booted to the program memory in the Boot file tab of the PLC parameter dialog box.

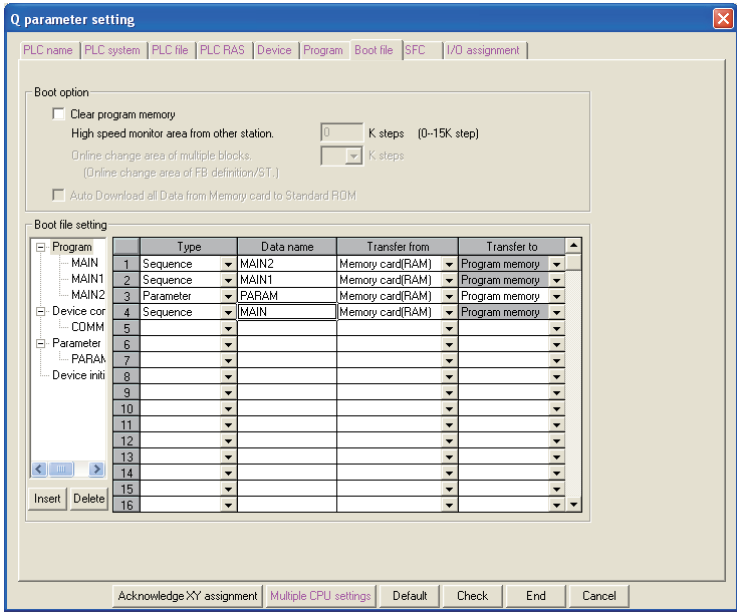


Figure 5.21 Boot file tab

(c) Mounting the memory card

Mount the memory card to the CPU module.

(d) Writing to the memory card

Write the parameters and programs set in the Boot file tab to the memory card.

(e) Executing the program

Set the RUN/STOP/RESET switch to RESET.

The BOOT LED turns on after a boot from the specified memory is completed.

(f) Checking whether a boot is normally completed

The following status indicates normal completion of boot operation.

- The BOOT LED turns on.
- The special relay (SM660) turns on.
- The data written to the transfer source memory and the data in the program memory are found the same by verification made by selecting [Online] → [Verify with PLC] in GX Developer.

(4) Operation for stopping boot operation

To stop boot operation and operate the CPU module by the parameters and program files written to the program memory, perform the following operations.

- 1) Remove the memory card and write parameters without boot file setting to the program memory by GX Developer.
- 2) Power on the programmable controller again or reset the CPU module.

(5) Boot operation precautions

(a) Storage location of parameters

- Store the parameters set in the Boot file tab of the PLC parameter dialog box to the memory card.
If the parameters set in the Boot file tab are stored in the program memory or standard ROM, the CPU module ignores the parameter settings. (☞ Section 5.1.10)
- When both of the following status 1) and 2) occurs, the CPU module is not operated by the parameters in the memory card, but by the parameters in the program memory.
 - 1) There are parameters in the program memory.
 - 2) The parameters in the memory card are not set in the Boot file tab.

(b) Online change in boot operation

Even if a program in the program memory is overwritten in the RUN status by boot operation, the change is not updated to the original program in the memory card.

Therefore, write the same file whose program was written to the program memory to the memory card (ROM) by Write to PLC (Flash ROM). (☞ Section 5.1.7)

(c) Maximum number of settable boot files

Set the maximum number of settable boot files in the Boot file tab of the PLC parameter dialog box so that it may be the same with the number of files storable to the program memory.

However, the number of boot files reduces by 1 when:

- a heading is set, or
- the parameters set in the Boot file tab of the PLC parameter dialog box and stored in the memory card is booted.

(d) Boot operation when the ATA card is used

When data are booted from the ATA card, the processing time of maximum 200ms may be required per 1K step (4K bytes).

(e) When data in the program memory are changed after the CPU module is powered off and then on or is reset

If the program memory data are changed after the sequence program is written to the program memory and the CPU module is powered off and then on or is reset, a boot operation may be active.

While the BOOT LED on the front of the CPU module is on, the boot operation is active. Refer to (4) in this section and stop the boot operation.

(f) Size after a boot from the memory card

The size unit of a file stored in each memory differs between the memory card and the program memory.

( Section 5.3.4)

Therefore, note that files transferred from the memory card to the program memory differ in memory capacity between before and after the transfer.

(g) Program written to the memory card

Set the programmable controller type (model name of the CPU module) for the program written to the memory card (program set in the Boot file tab) and the model name of the CPU module to be booted to the same.

5.1.9 Details of written files

For each file written to the CPU module, its name, size, and created date and time set at the file creation are appended.

The file is displayed on the Read from PLC screen, opened by selecting [Online] → [Read from PLC] in GX Developer, as shown below.

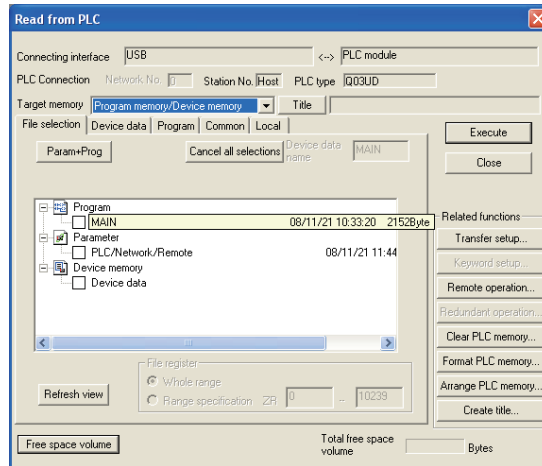


Figure 5.22 Displaying the details of a file

(a) File name

1) File name structure and file specification

Each file name is composed of a name (up to 8 characters in one byte/4 characters in double bytes) and an extension (3 characters).

Create a file name with upper-case characters only.

An extension is automatically appended according to the type set when the file was created.

2) Characters that cannot be used for a file name

The following reserved words for Microsoft® Windows® cannot be used as a file name.

COM1 to COM9, PRN, LPT1 to LPT9, NULL, AUX, CLOCK\$, CON

3) How to specify a file name in the sequence program

Since the sequence program is not case-sensitive, the file can be named by both upper-case and lower-case characters.

*Both "ABC" and "abc" are treated as "ABC".

In double-byte characters, an upper-case character and lower-case character are distinguished. Name a file by an upper-case character.

*"ABC" and "abc" are distinguished.

(b) Date and time

The date and time when a file was written to the CPU module is shown.

The date and time are appended according to the clock set on the GX Developer (personal computer) side.

(c) Size

The file size when the file was written from GX Developer to the CPU module is shown in units of bytes (To display the latest data, select [Online] → [Read from PLC] and click the **Refresh view** button). At least 64 bytes (136 bytes for a program) are added to the file created by an user except a file register file.

(☞ Section 5.3.3)

5.1.10 Specifying valid parameters (parameter-valid drive setting)

Drives (memories) storing valid parameters are automatically specified by the system. The valid parameters are determined by the priority of the drives where parameters are stored. Settings by an user is unnecessary.

(1) Priority of the parameter-valid drives

Table5.6 provides the priority of the drives where parameters are stored.

Write parameters to be validated to the higher priority drives.

Table5.6 Priority of the drives

Priority		Parameter drive
High ↑ ↓ Low	1	Drive 0 (program memory)
	2	Drive 1 (memory card RAM)
	3	Drive 2 (memory card ROM)
	4	Drive 3 (standard ROM)

Figure 5.23 provides the operation flowchart for the CPU module to select parameter drives.

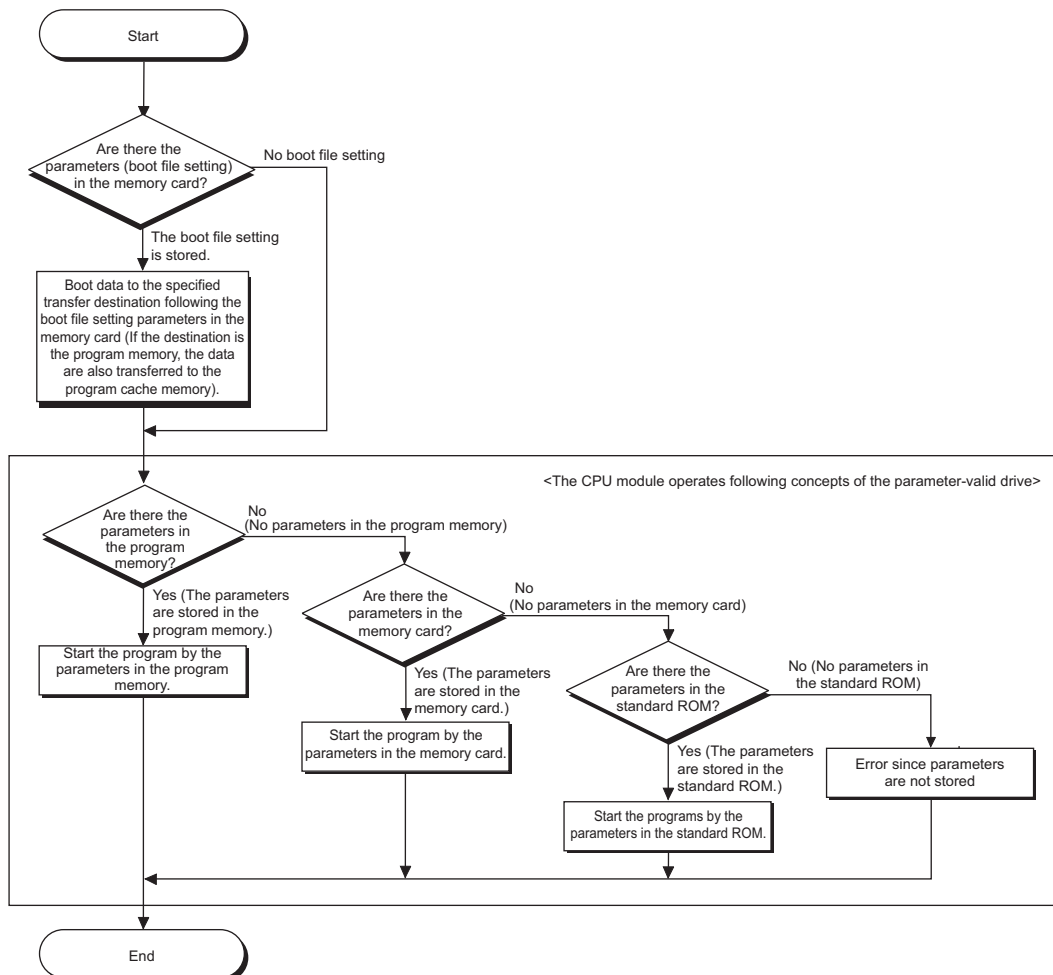


Figure 5.23 Flowchart to specify drives storing parameters

(2) When to determine valid parameters

The CPU module automatically searches for parameters in the following timing and operates by the settings of the parameters stored in the drives when:

- the CPU module is powered off and then on, or
- it is reset.

When storing parameters by Write to PLC in GX Developer, the timing for validating the parameters depends on the drive that stores the parameters.

(a) When parameters are stored to the drive different from the one that stores the parameters in operation

The parameters are validated according to the priority set to the drive after the CPU module is powered off and then on or is reset.

(b) When parameters are stored to the drive same as the one that stores the parameters in operation

Only the setting made in the Device tab of the PLC parameter dialog box is validated after "Write to PLC" is performed.

To validate all parameter settings, power off and then on or reset the CPU module.

5.2 Program File Structure

A program file consists of a file header, execution program, and reserved area for online change.

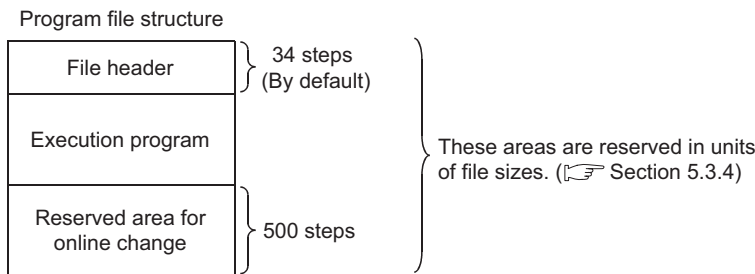


Figure 5.24 Program file structure

(1) Details of each structure

The capacity of the programs stored in the CPU module program memory is the total of above three areas.

(a) File header

This area stores the name, size, and created date of files.
The file header size ranges from 25 to 35 steps (100 to 140 bytes) depending on the setting made in the Device tab of the PLC parameter dialog box (34 steps is set by default).

(b) Execution program

This area stores the created program.

(c) Reserved area for online change

This area is used when the number of steps is increased after writing data in the RUN status from GX Developer.
When such operation is performed, the remaining reserved area for online change is shown.

1) Default

500 steps (2000 bytes) is set by default.

2) Setting change

To change the number of steps, select [Online] → [Write to PLC] → <<Program>> tab in GX Developer.
This setting can be made when data are written in the RUN status. (☞ Section 6.12.1)

(2) Displaying the program capacity on the GX Developer screen

During programming by GX Developer, the program size (total of the file header size and the number of steps in the created program) is displayed by the number of steps as shown in Figure 5.25.

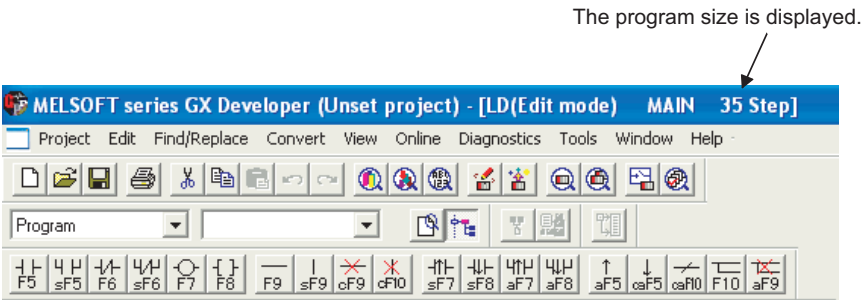
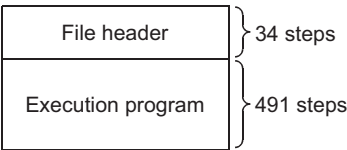


Figure 5.25 Displaying the program size

Point

- The program size displayed during programming by GX Developer is the total of the file header and execution program, and the capacity of the reserved area for online change (500 steps) is not included.

Example The size of program having the execution program area of 491 steps is displayed on the GX Developer screen as shown below (The file header is fixed to 34 steps).



Display on the GX Developer screen:
34 steps + 491 steps = 525 steps

Figure 5.26 File status displayed on the GX Developer screen

- Since files are stored in units of file sizes in the program memory, the program size displayed during programming by GX Developer may differ from the program file size in the CPU module.
(☞ Section 5.3.3)

5.3 File Operations by GX Developer and Handling Precautions

5.3.1 File operations

Table5.7 shows the functions can be performed to files stored in the program memory, standard ROM, and memory card by the online functions of GX Developer.

However, the executable operations depends on the password registration setting by GX Developer and CPU module status (RUN/STOP).

Table5.7 File operations executable from GX Developer

File operation	Operation detail	Operability		
		While write protection password is set to the file	While write/read protection password is set to the file	While the CPU module is in the RUN status
Read from PLC	Reads a file from the target memory.	○	△	○
Write to PLC	Writes a file to the program memory or SRAM card.	△	△	○
	Writes a file to the standard ROM.	△	△	○
Verify with PLC	Verifies the file in the target memory and the file of GX Developer.	△	△	○
Write the program memory to ROM	Batch-writes files stored in the program memory to the Flash card.	○	○	○
Write to PLC (Flash ROM)	Batch-writes specified files from GX Developer to the Flash card.	△	△	○
Format PLC memory	Formats a memory.	○	○	×
Arrange PLC memory	Rearranges files stored in a memory at random.	○	○	×
Online change (ladder mode)	Writes data changed in the ladder mode to the program memory.	△	△	○

○ : Executable, △ : Executable when entered password matches, × : Cannot be executed

5.3.2 Precautions for handling files

(1) Power-off or reset at file operation

When the CPU module is powered off or reset at file operation, files in each memory are held (for a memory card, when the CPU module where a memory card has been mounted before power-off is powered on).

Point

When the programmable controller is powered off during an operation in which a file is moved, the data in operation are held in the internal memory of the CPU module.
The held data are recovered at power-on.
To hold the internal memory data, battery backup is required.

(2) Concurrent writing from multiple GX Developers to one file

GX Developers other than the one writing data to a file cannot access the file until the writing ends.

Also, GX Developers other than the one accessing a file cannot write data to the file until the access ends.

Therefore, write data to one file from multiple GX Developers one by one.

(3) Concurrent access from multiple GX Developers to different files

Except the currently viewed GX Developer, maximum 10 GX Developers can access to different files in one CPU module concurrently.

5.3.3 File size

The size of a file used for the CPU module depends on the file type.

When a file is written to the memory area, the unit of the stored file depends on the CPU module and memory area to be written. (☞ Section 5.3.4)

When using the program memory, standard RAM, standard ROM, or memory card, calculate the rough size of each file with reference to Table5.8.

Table5.8 Calculation of file size

Function	Rough file size (unit: Byte)																							
Drive heading	72																							
Parameter	Default: 464 (can be increased by parameter setting.)*¹ Reference Boot setting → 84 + (18 × (number of files))*² • With CC-Link IE controller network setting → • 72 + (total parameter sizes of each module) + (size of the routing setting) + (size of the data link transfer setting) • Parameter size of each module: Up to 10368 (when only LB/LW(1) is set, 1826 + 16 × (number of refresh transfer points)) • Routing setting size: 6 + 8 × (number of routing settings) • Size of data link transfer setting: 6 + 12 × (number of transfer settings) + 86 × (number of modules) • With CC-Link IE Field network setting → • 72 + (total parameter sizes of each module) + (size of the routing setting) + (size of the data link transfer setting) • Parameter size of each module: Up to 13074 • Routing setting size: 6 + 8 × (number of routing settings) • Size of data link transfer setting: 6 + 16 × (RX number of transfer settings) + 16 × (RW number of transfer settings) • With the MELSECNET/H setting → Increase up to 6180/module • With the Ethernet setting → Increase up to 922/module • With the CC-Link setting → Increase up to the values in the following table (The values indicate an increment of each module). <table><tr><th rowspan="2">CC-Link setting</th><th colspan="3">Mode setting</th></tr><tr><th>Ver.1 mode</th><th>Ver.2 mode</th><th>Ver.2 additional mode</th></tr><tr><td>1st module</td><td>550 bytes</td><td>572 bytes</td><td>624 bytes</td></tr><tr><td>2nd to 4th modules</td><td>536 bytes</td><td>558 bytes</td><td>610 bytes</td></tr><tr><td>5th module</td><td>550 bytes</td><td>566 bytes</td><td>618 bytes</td></tr><tr><td>6th to 8th modules</td><td>536 bytes</td><td>558 bytes</td><td>610 bytes</td></tr></table> • With the remote password setting →92 + (number of target modules × 10), increase up to 172/module	CC-Link setting	Mode setting			Ver.1 mode	Ver.2 mode	Ver.2 additional mode	1st module	550 bytes	572 bytes	624 bytes	2nd to 4th modules	536 bytes	558 bytes	610 bytes	5th module	550 bytes	566 bytes	618 bytes	6th to 8th modules	536 bytes	558 bytes	610 bytes
CC-Link setting	Mode setting																							
	Ver.1 mode	Ver.2 mode	Ver.2 additional mode																					
1st module	550 bytes	572 bytes	624 bytes																					
2nd to 4th modules	536 bytes	558 bytes	610 bytes																					
5th module	550 bytes	566 bytes	618 bytes																					
6th to 8th modules	536 bytes	558 bytes	610 bytes																					
Sequence program	148* ³ + (4 × ((number of steps) + (number of steps of reserved area for online change)))																							
Device comment	74 + 8 + (total comment data size of each device) Comment data size per device = 10 + 10250 × a + 40 × b • a: Quotient of ((number of device points)/256) • b: remainder of ((number of device points)/256))																							
Initial device value	66 + 44 × n + 2 × (total number of device points set to the initial device value) + 8 n: number of settings of the initial device value																							
User setting area	Setting value at formatting (0 to 15K)																							
File register* ⁶	2 × (number of file register points)																							

(To the next page)

Table5.8 Calculation of file size(continued)

Function	Rough file capacity (unit: Byte)
Sampling trace file* ⁶	$362 + (\text{number of word device points} + \text{number of bit device points}) \times 12 + (N1 + N2 + N3 + \text{number of word device points} \times 2 + (\text{number of bit device points}/16) \times 2) \times \text{the number of traces (total number of executions)}^{*4}$ - Apply the following values to N1 to N3 according to the items set in "Trace additional information" of the Trace condition settings screen. (☞ Section 6.14(4)(b)) - N1: When "Time" is set, apply "4". - N2: When "Step no." is set, apply "10". - N3: When "Program name" is set, apply "8".
Device data backup file	Setting value at formatting (2 to 1024K)
Module error collection file* ⁸	$76 + (64 \times (\text{value set for the number of storable errors}))$
Local device* ⁶	$72 + 6 \times (\text{set device type}) + 2 \times ((\text{total number of M and V points})/16 + (\text{number of D points}) + 18 \times (\text{total number of T, ST, and C points})/16) \times (\text{number of programs where local devices are used})^{*5}$ - M, V, D, T, ST, and C indicate the following set devices. - M: internal relay - V: edge relay - D: data register - T: timer - ST: retentive timer - C: counter - Z: index register*⁷

*1: The value is adjusted by the system so that the total bytes with the addition of the network parameter setting may be multiple of four.

*2: The value is adjusted by the system so that the bytes may be multiple of four.

*3: 148 is set by default (This value depends on the parameter setting).

*4: After the decimal point of a value found by the number of bit device points/16 is rounded up.

*5: For the Universal model QCPU, if the serial number (first five digits) is "10011" or earlier, apply the number of execution programs.

*6: The Q00UJCPU does not support this function.

*7: When using the index register as a local device for the Q03UDCPU, Q04UDHCPU, Q06UDHCPU, Q13UDHCPU, Q26UDHCPU, or Built-in Ethernet port QCPU, check the versions of the CPU module and GX Developer.

(☞ Appendix 2)

*8: When using the module error collection, check the versions of the CPU module and GX Works2.

(☞ Appendix 2)

Remark

For calculation example of memory capacity, refer to Section 5.3.4.

5.3.4 Units of file sizes

(1) Definition

When a file is written to the memory area, the unit of the stored file depends on the CPU module and memory area to be written. This unit is referred to as a file size unit.

(a) File size unit for each memory area

The following table shows the file size unit depending on the CPU module and memory area to be written.

Table5.9 File size unit for each CPU module and memory area

CPU module model	Memory area	
	Program memory, standard ROM, Flash card ^{*1}	Standard RAM
Q00UJCPU	Program memory: 1 step/4 bytes Standard ROM: 64 steps/256 bytes	-
Q00UCPU, Q01UCPU	Program memory: 1 step/4 bytes Standard ROM: 128 steps/512 bytes	512 bytes
Q02UCPU, Q03UDCPU, Q04UDHCPU, Q06UDHCPU, Q03UDECPU, Q04UDEHCPU, Q06UDEHCPU	Program memory: 1 step/4 bytes Standard ROM, Flash card: 128 steps/512 bytes	
Q10UDHCPU, Q10UDEHCPU, Q13UDHCPU, Q13UDEHCPU	Program memory: 1 step/4 bytes Standard ROM: 256 steps/1024 bytes Flash card: 128 steps/512 bytes	
Q20UDHCPU, Q20UDEHCPU, Q26UDHCPU, Q26UDEHCPU, Q50UDEHCPU, Q100UDEHCPU	Program memory: 1 step/4 bytes Standard ROM: 512 steps/2048 bytes Flash card: 128 steps/512 bytes	

^{*1}: The file size unit of the Flash card is applied when a file is written to the Flash card by [Online] → [Write the program memory to ROM] in GX Developer. (☞ Section 5.1.7(1)(a))

(b) File size unit for each memory card

Table5.10 File size unit for each memory card

Type	Memory card model	File size unit (cluster size)
SRAM card	Q2MEM-1MBS	512 bytes
	Q2MEM-2MBS	1024 bytes
	Q3MEM-4MBS	1024 bytes
	Q3MEM-8MBS	4096 bytes
Flash card ^{*1}	Q2MEM-2MBF	1024 bytes
	Q2MEM-4MBF	1024 bytes
ATA card	Q2MEM-8MBA	4096 bytes
	Q2MEM-16MBA	4096 bytes
	Q2MEM-32MBA	2048 bytes

^{*1}: The file size unit of the Flash card is applied when:

•file is written to the Flash card by [Online] → [Write to PLC (Flash ROM)] in GX Developer.

(☞ Section 5.1.7(1)(b))

•file is written to the Flash card by GX Developer without accessing the CPU module

(2) Calculation example of memory capacity

The following shows an calculation example of memory capacity when the parameters and sequence program are written to the program memory.

(a) Conditions

- 1) CPU module to be written: Q26UDHCPU
- 2) Writing file

Table5.11 File sizes

File name	File size*1
PARAM.QPA (parameter file)	464 bytes
MAIN.QPG (sequence program)	525 steps/2100 bytes*2

*1: For the file size, refer to (1)(a) in this section.

*2: This indicates the program size (file header + execution program) displayed on the GX Developer screen.

(☞ Section 5.2)

- 3) Reserved area for online change: 500 steps/2000 bytes

(b) Memory capacity calculation

The memory capacity is calculated in units of file sizes of the CPU module to be written.

The file size unit of the Q26UDHCPU in this example is 1 step/4 bytes.(☞ (1) in this section)

1) Calculation of parameter file size

Since the parameter file size is 464 bytes, 116 steps/464 bytes is occupied in the program memory.

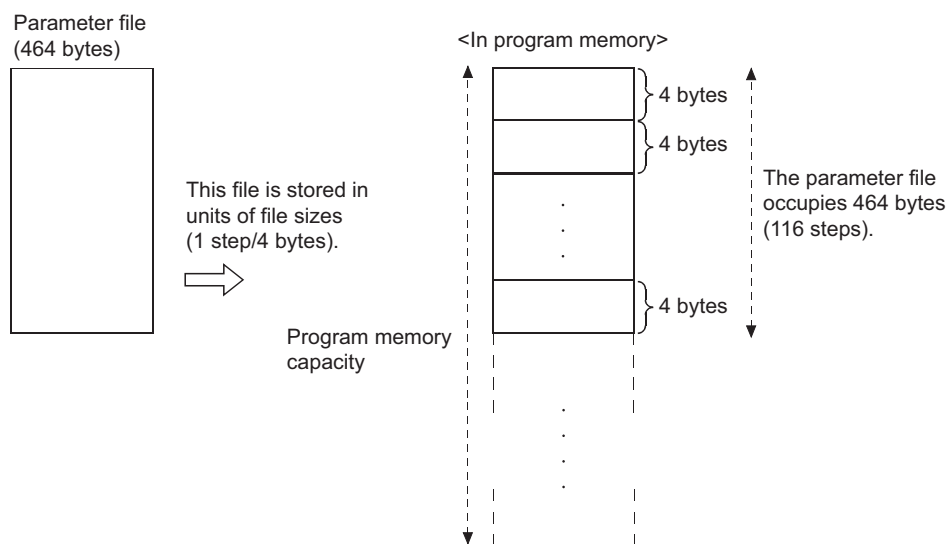


Figure 5.27 Occupation in units of the file size (parameter file)

2) Calculation of program size

The program size is found by the formula: sequence program size + reserved area for online change.

Since a program is stored in units of file sizes (1 step), only the amount equal to the program size is occupied.

3) Result

The calculation results of the memory capacities are as shown below.

Table5.12 Calculation results of memory capacities

File name	File size		Memory capacity
PARAM.QPA	464 bytes		116 steps (464 bytes)
MAIN.QPG	Sequence program size	525 steps	1025 steps (4100 bytes)
	Reserved area for online change	500 steps	
	Total	1025 steps	
Total memory capacity			1141 steps (4564 bytes)

CHAPTER6 FUNCTIONS

This chapter describes the functions of the Universal model QCPU.

6.1 Function List

Table6.1 lists the functions of the Universal model QCPU.

Table6.1 Function list

Item	Description	Q00UJ CPU	Q00UCPU, Q01UCPU	Q02U CPU	QnUD(H) CPU	Built-in Ethernet port QCPU	Reference
Boot operation	Transfers data stored on a memory card to the program memory or the standard ROM at power-on or reset.	×	×	○	○	○	Section 5.1.8
Constant scan	Executes a program in a set time interval regardless of its scan time.	○	○	○	○	○	Section 6.2
Latch function	Holds the device data even at power OFF or reset.	○	○	○	○	○	Section 6.3
Output status selection when the status changed from STOP to RUN	Selects the output (Y) status (outputting the same status prior to STOP or clearing the status) when the CPU module status is switched from STOP to RUN.	○	○	○	○	○	Section 6.4
Clock function	Reads the internal clock data of the CPU module to use it for time management.	○	○	○	○	○	Section 6.5
Remote RUN/STOP	Runs or stops the program operations in the CPU module externally.	○	○	○	○	○	Section 6.6.1
Remote PAUSE	Stops the program operations in the CPU module externally, holding the status of outputs (Y).	○	○	○	○	○	Section 6.6.2
Remote RESET	Resets the CPU module externally when the CPU module is in a STOP status.	○	○	○	○	○	Section 6.6.3
Remote latch clear	Clears the latch data in the CPU module when the CPU module is in a STOP status.	○	○	○	○	○	Section 6.6.4
Input response time selection	Selects input response time values for the Q series-compatible input modules, I/O combined modules, high-speed input modules, and interrupt modules.	○	○	○	○	○	Section 6.7
Error time output mode setting	Sets whether to clear or retain the output to the Q series compatible output modules, I/O combined modules, intelligent function modules, and interrupt modules at the time of a stop error of the CPU module.	○	○	○	○	○	Section 6.8
H/W error time PLC operation mode setting	Sets whether to stop or continue operations in the CPU module when a hardware error has occurred in an intelligent function module or interrupt module.	○	○	○	○	○	Section 6.9

○: Supported, △: Partly supported, ×: Not supported

(To the next page)

Table6.1 Function list (Continued)

Item	Description	Q00UJ CPU	Q00UCPU, Q01UCPU	Q02U CPU	QnUD(H) CPU	Built-in Ethernet port QCPU	Reference
Intelligent function module switch setting	Makes settings for the intelligent function modules and interrupt modules. (Refer to manuals of intelligent function modules and interrupt modules for setting details.)	○	○	○	○	○	Section 6.10
Monitor function	Reads the status of programs and devices in the CPU module by GX Developer.	○	○	○	○	○	Section 6.11
Monitor condition setting	Specifies the monitoring timing of the CPU module with device condition or step number.	×	×	△ *1	△ *1	○	Section 6.11.1
Local device monitor/test	Monitors and/or tests the local devices of the specified program by GX Developer.	×	○	○	△ *1	○	Section 6.11.2
External input/output forced on/off	Forcibly turns on/off the external input/output of the CPU module by GX Developer.	○	○	△ *1	△ *1	○	Section 6.11.3
Executorial conditioned device test	Changes a device value within the specified step of a sequence program.	○	○	△ *1	△ *1	○	Section 6.11.4
Online change	Writes programs when the CPU module is in the RUN status.	○	○	○	○	○	Section 6.12
Program monitor list	Displays the scan time and execution status of the program being executed.	○	○	○	○	○	Section 6.13.1
Interrupt program monitor list	Displays the number of executions of interrupt programs.	○	○	○	○	○	Section 6.13.2
Scan time measurement	Measures the execution time of the area specified by the steps in a program.	○	○	△ *1	△ *1	○	Section 6.13.3
Sampling trace function	Continuously samples the specified device data at a preset timing.	×	○	○	○	○	Section 6.14
Debug function from multiple GX Developers	Enables simultaneous debugging by multiple GX Developers.	○	○	○	○	○	Section 6.15
Watchdog timer	Monitors operational delays caused by hardware failure or program error of the CPU module.	○	○	○	○	○	Section 6.16
Self-diagnostic function	Self-diagnoses the CPU module to see whether an error exists or not.	○	○	○	○	○	Section 6.17
Error history	Stores the result of self-diagnostics to the memory as error history data.	○	○	○	○	○	Section 6.18
System protection	Prevents the programs from being modified from GX Developer, serial communication module, and Ethernet module.	○	○	○	○	○	Section 6.19

○: Supported, △: Partly supported, ×: Not supported

(To the next page)

Table6.1 Function list (Continued)

Item	Description	Q00UJ CPU	Q00UCPU, Q01UCPU	Q02U CPU	QnUD(H) CPU	Built-in Ethernet port QCPU	Reference
Password registration	Prohibits writing/reading data to/from each file in the CPU module using GX Developer.	○	○	○	○	○	Section 6.19.1
Remote password	Prevents unauthorized access from external devices such serial communication module and Ethernet module.	○	○	○	○	○	Section 6.19.2
System display	Monitors the system configuration using GX Developer.	○	○	○	○	○	Section 6.20
LED indication	Displays the operating status of the CPU module with LEDs on the front of the module.	○	○	○	○	○	Section 6.21
LED indication priority	Sets whether to indicate an error with LED according to the priority of each error.	○	○	○	○	○	Section 6.21.2
Interrupt from intelligent function module	Executes an interrupt program at the time of interrupt request from the intelligent function module.	○	○	○	○	○	Section 6.22
Serial communication function	Connects the RS-232 interface of the CPU module and the personnel computer or HMI with RS-232 cable and communicates in the MC protocol.	○	○	△ ^{*1}	×	×	Section 6.23
Service processing setting	Specifies the service processing count or time to be executed in END processing.	○	○	○	○	○	Section 6.24.1
Initial device value	Registers data used in programs with devices and the buffer memories of the intelligent function modules and special function modules without programs.	○	○	○	○	○	Section 6.25
Battery life-prolonging function	Prolongs a battery life by holding only clock data in the battery.	○	○	○	○	○	Section 6.26
Memory check function	Checks whether data in the memories of the CPU module are not changed due to excessive electric noise.	○	○	○	○	○	Section 6.27
Latch data backup to standard ROM function	Backs up latch data such as device data and error history without using a battery.	○	○	○	○	○	Section 6.28
Writing/reading device data to/from standard ROM	Writes/reads device data to/from standard ROM.	○	○	○	○	○	Section 6.29
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.	×	×	△ ^{*1}	△ ^{*1}	△ ^{*1}	Section 6.30

○: Supported, △: Partly supported, ×: Not supported

(To the next page)

Table6.1 Function list (Continued)

Item	Description	Q00UJ CPU	Q00UCPU, Q01UCPU	Q02U CPU	QnUD(H) CPU	Built-in Ethernet port QCPU	Reference
Module model name read	Reads the model name of a module on a base unit.	△ *1	△ *1	△ *1	△ *1	△ *1	Section 6.31
Module error collection	Collects errors occurred in the connected intelligent function modules in the CPU module.	△ *1	△ *1	△ *1	△ *1	△ *1	Section 6.32
Local device batch read function	Batch-reads local device data in the CPU module and stores them in a CSV file.	×	△ *1	△ *1	△ *1	△ *1	Section 6.33
Send points expansion function (CC-Link IE controller network module)	Expands the maximum link points per station using a CC-Link IE controller network module.	△ *1	△ *1	△ *1	△ *1	△ *1	Section 6.34

*1: Availability depends on the version of the CPU module. (☞ Appendix 2)

6.2 Constant Scan

(1) Definition

Scan time of the CPU module is not constant because the processing time varies depending on the execution status of instructions used in a sequence program.

This function allows sequence programs to be executed repeatedly, maintaining its scan time constant.

(2) Application

I/O refresh is performed before every sequence program execution.

This function is used to maintain I/O refresh intervals constant even if the execution time of each sequence program differs.

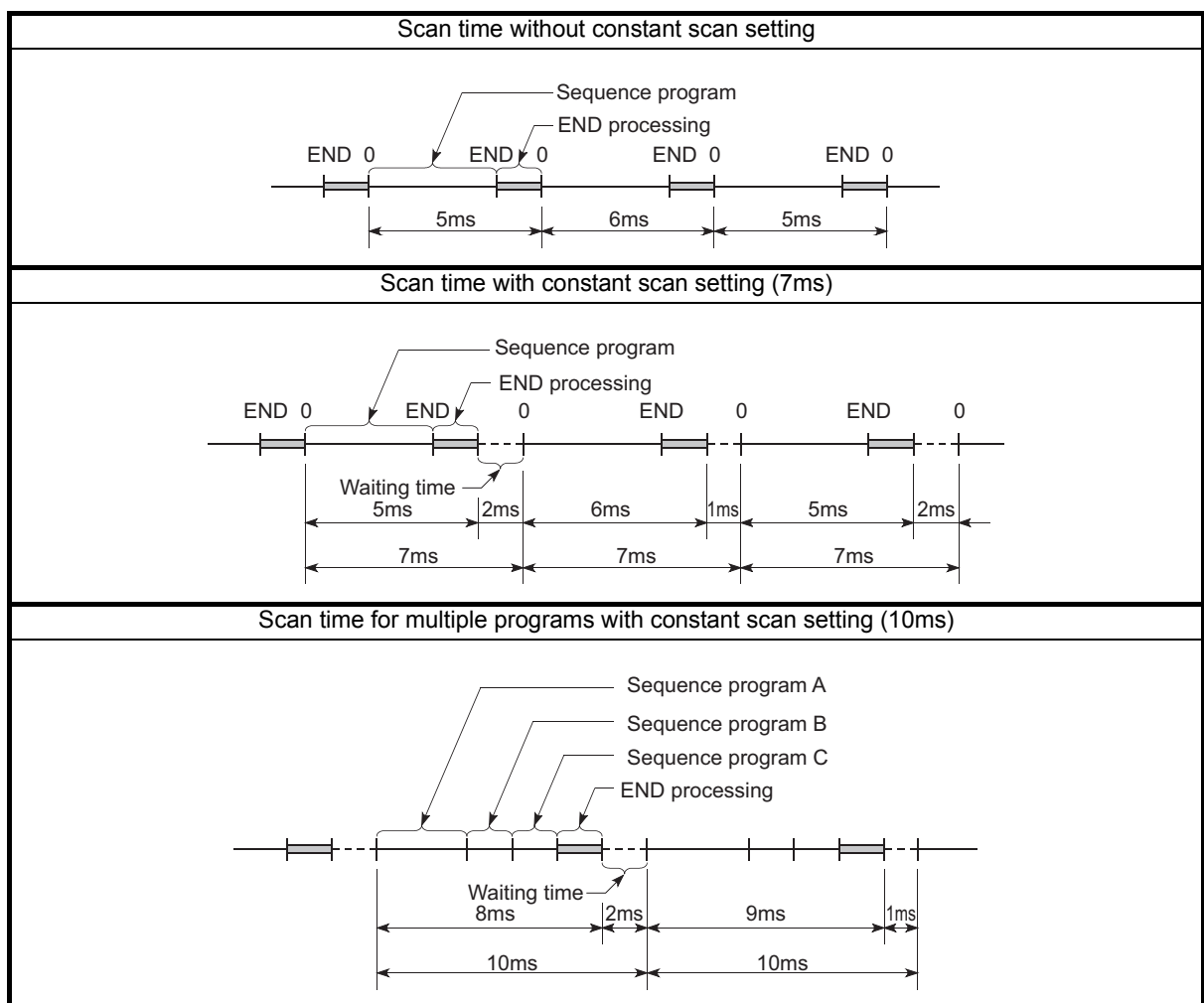


Figure 6.1 Constant scan operation

(3) Constant scan time setting

Set a constant scan time value in the PLC RAS tab of the PLC parameter dialog box.
The setting range is 0.5 to 2000ms (in increments of 0.5ms).
When not executing the constant scan function, leave the constant scan time setting box blank.

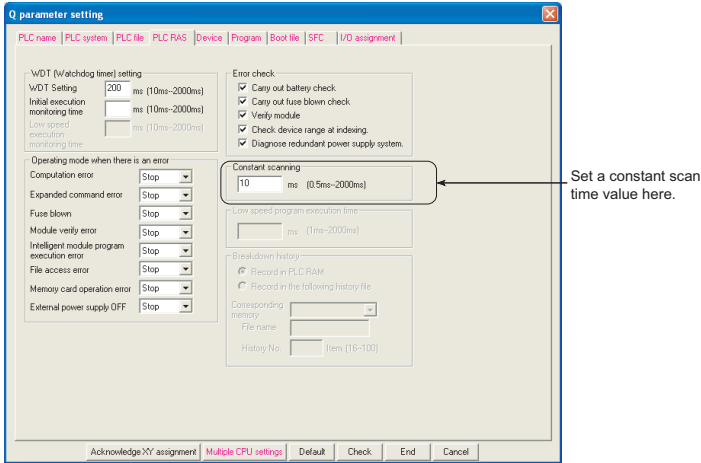


Figure 6.2 When the constant scan time is set to 10ms

(a) Condition

The constant scan time needs to satisfy the following relational expression.

$$(WDT \text{ setting time}) > (Constant \text{ scan setting time}) > (Sequence \text{ program maximum scan time})$$

If the sequence program scan time is longer than the constant scan setting time, the CPU module detects “PRG. TIME OVER” (error code: 5010).
In this case, the constant scan setting will be ignored and the sequence program scan time will be applied.

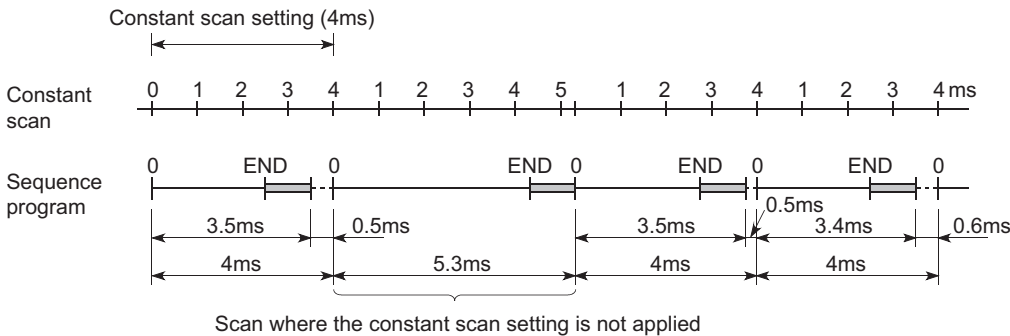


Figure 6.3 Operation when the scan time is longer than the constant scan setting time

If the sequence program scan time is longer than the WDT setting time, the CPU module detects “WDT ERROR”.
In this case, the program execution will be stopped.

(4) Waiting time from when END processing is executed until next scan starts

Sequence program processing is stopped during the waiting time from when END processing of a sequence program is executed until next scan starts.

(a) When an interrupt factor occurs during waiting time

Either of the following programs is executed.

- Interrupt program
- Fixed scan execution type program

(b) When a service processing parameter is set

The communication service processing with peripherals (such as GX Developer) and intelligent function modules is enabled during the waiting time by setting a service processing parameter. (➡ Section 6.24.1)

(5) Constant scan accuracy

The constant scan accuracy is 0.01ms.

However, when any program indicated below is being executed, the constant scan time may increase.

(a) Interrupt program or fixed scan execution type program

Interrupts are disabled while an interrupt program or fixed scan execution type program is executed.

Even if the constant scan time runs out during execution of an interrupt program or fixed scan execution type program, the constant scan cannot be finished.

In this case, the constant scan time may increase by the execution time of the program executed.

6.3 Latch Function

(1) Definition

This 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 set back to its default (bit device: off, word device: 0) without using the latch function.

(2) Application

This function is used to hold the data managed by sequential control and continue control operation especially when the CPU module is powered off and then on.

(3) Program operation when the latch function is use

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

(4) Devices that can be latched

The following devices can be latched.(By default, only the latch relay is enabled for latch.)

- Latch relay (L)
- Link relay (B)
- Annunciator (F)
- Edge relay (V)
- Timer (T)
- Retentive timer (ST)
- Counter (C)
- Data register (D)
- Link register (W)

Point

When the battery life-prolonging function ( Section 6.26) is set, the latch relay cannot be latched.

(5) Latch range setting

Set a latch range in the Device tab of the PLC parameter dialog box.

There are two types of latch range settings: the latch clear operation enable range setting (Latch (1)) and the latch clear operation disable range setting (Latch (2)).

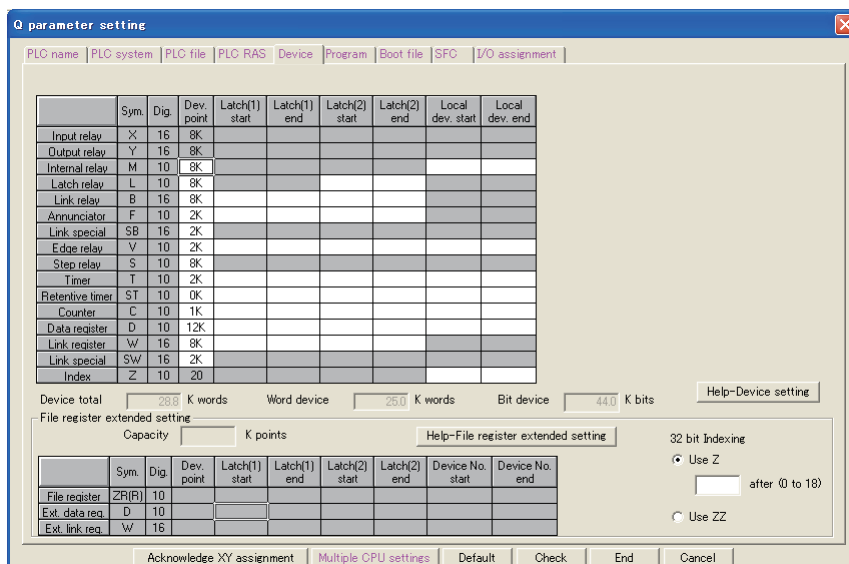


Figure 6.4 Latch range setting

Point

The latch range of the file register (ZR), extended data register (D), and extended link register (W) can also be set.

After selecting "Use the following file." and setting the capacity of the file register in the PLC file tab of the PLC parameter dialog box, set a latch range of the file register (ZR), extended data register (D), and extended link register (W). (For the extended data register (D) and extended link register (W), assign a part of the file register area.)

If "Use the same file name as the program." is selected in the PLC file tab of the PLC parameter dialog box, a latch range of the file register (ZR), extended data register (D), and extended link register (W) cannot be set. (All data in the file register will be latched.)

The data outside the latch range will be cleared when the CPU module is powered off and then on or is reset.

The file register points set in the PLC file tab is displayed.

Set a latch range.

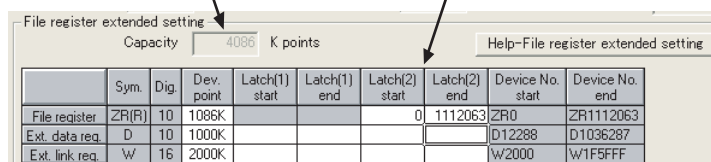


Figure 6.5 Latch range setting

When the file register file to be used is switched with the QDRSET instruction, the latch range setting of the file register will be invalid.

After switching, regardless of the latch range setting, whole range of the file register will be latched.

(6) Device data latch method and influence on the scan time

Data latch processing is performed during END processing.

For this reason, the scan time increases.

Consider an influence on the scan time when latching devices. (☞ Section 10.1.2)

Point

To shorten the scan time due to latch, minimize the number of latch points (latch (1) setting, latch (2) setting, and latch relay (L)) as much as possible.

The number of latch points can be reduced by performing the following.

- Move data to be latched to the file register.
- Store device data that is less frequently updated in the standard ROM with the SP.DEVST instruction. (The device data stored in the standard ROM can be read with the S(P).DEVLD instruction.)

☞ MELSEC-Q/L Programming Manual (Common Instruction)

(7) Device data latch clear

Table6.2 shows the status of device data when the latch clear operation is performed.

Table6.2 Status when the latch clear operation is performed

Latch setting	Status of data
Device data without latch setting	Cleared
Device data in the "Latch (1)" range	Cleared
Device data in the "Latch (2)" range	Held ^{*1}

*1: For the clearing method, refer to Section 3.7.

Point

Data in the file register (R or ZR) will not be cleared by the latch clear operation.

To clear data in the file register (R or ZR), perform data clear operation by a sequence program or GX Developer.

☞ Section 9.7.3

(8) Precautions

(a) When a local device or initial device value is specified

Device data cannot be latched even if the device has been latch-specified,

(b) Use of battery

Device data in the latch range are held with the battery installed to the CPU module.

- Even for the boot operation, the battery is required to latch device data.
- Note that if the battery connector is disconnected from the connector of the CPU module while the power supply for the programmable controller is off, device data in the latch range will not be held and will become undefined.

6.4 Output Mode at Operating Status Change (STOP to RUN)

(1) Definition

When the operating status is changed from RUN to STOP, the CPU module internally stores the outputs (Y) in the RUN status and then turns off all the outputs (Y).

The status of the outputs (Y) when the operating status of the CPU module is changed back from STOP to RUN can be selected from the following two options in the parameter setting in GX Developer.

- Output the output (Y) status prior to STOP. ("Previous state")
- Clear the output (Y) status. ("Recalculate (output is 1 scan later)")

(2) Application

This function is used to determine the status of outputs (whether to resume the outputs from the previous status or not) when the operating status is changed from STOP to RUN in the holding circuit.



Figure 6.6 Holding circuit

- When outputting the output (Y) status prior to STOP

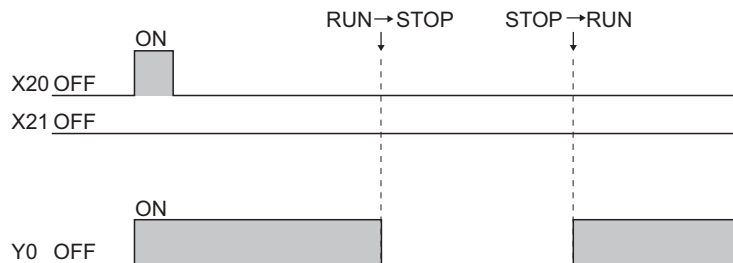


Figure 6.7 Timing chart when the parameter is set to "Previous state"

- When clearing the output (Y) status

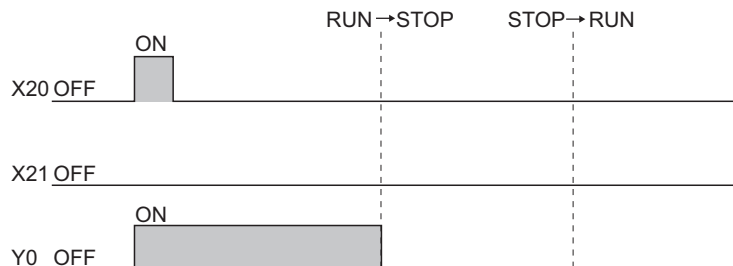


Figure 6.8 Timing chart when the parameter is set to "Recalculate (output is 1 scan later)"

(3) Operation when the operating status is changed from STOP to RUN

(a) Previous state (Default)

The CPU module outputs the output (Y) status immediately before changing to the STOP status and then performs sequence program operations.

(b) Recalculate (output is 1 scan later)

All outputs are turned off.

The CPU module outputs the output (Y) status after sequence program operations are completed. For the operation of the CPU module when the output (Y) status is forcibly turned on in the STOP status, refer to (5) in this section.

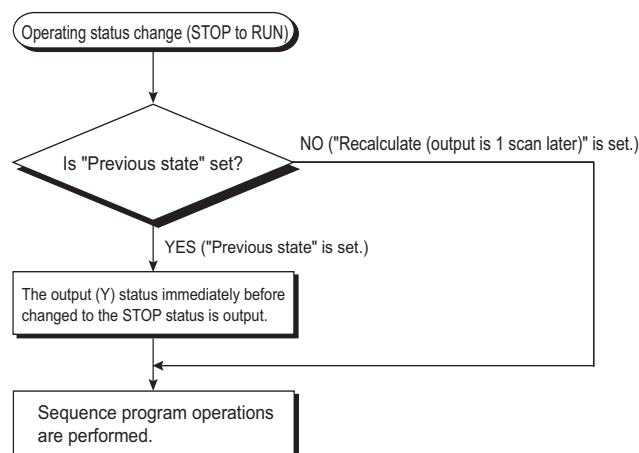


Figure 6.9 Operation when the operating status is changed from STOP to RUN

(4) Setting the output mode when the operating status is changed from STOP to RUN

Set the output mode when the operating status is changed from STOP to RUN in the PLC system tab of the PLC parameter dialog box.

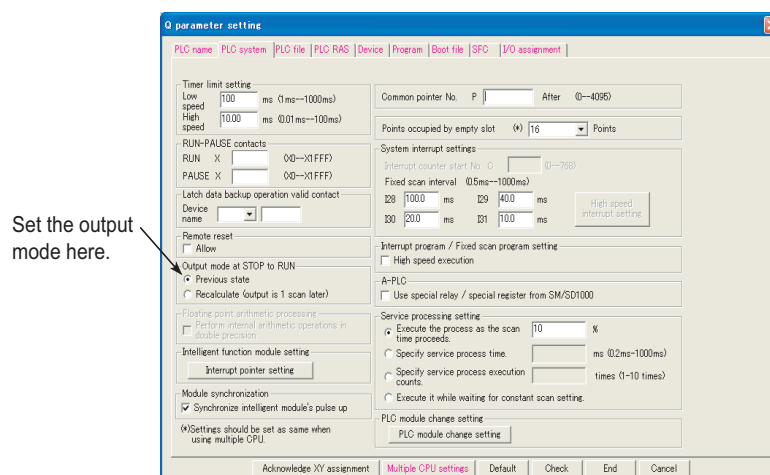


Figure 6.10 PLC system setting

(5) Precautions

Table6.3 shows the output status of the CPU module when the operating status is changed from STOP to RUN after the outputs (Y) are forcibly turned on in the STOP status.

Table6.3 Output status when the operating status is changed from STOP to RUN after the output forced on operation is performed

Output mode ("Output mode at STOP to RUN") selected	Output status
Previous state	The output status prior to STOP is output. Even if the outputs are forcibly turned on, the on status is not held if the output status prior to STOP was off.
Recalculate (output is 1 scan later)	The on status is held and output.

6.5 Clock Function

(1) Definition

This function reads the internal clock data of the CPU module by a sequence program and uses it for time management.

The clock data is used for time management required for some functions in the system, such as storing date into the error history.

Remark

The Built-in Ethernet port QCPU can set the time in the CPU module automatically by using the time setting function (SNTP client).

 QnUCPU User's Manual (Communication via Built-in Ethernet Port)

(2) Clock operation at power off and momentary power failure

Clock operation continues by the internal battery of the CPU module even when the programmable controller is powered off or power failure occurs exceeding the allowable momentary power failure time.

(3) Clock data

Table6.4 shows the details of clock data, which is used internally in the CPU module.

Table6.4 Clock data details

Data name	Description	
Year	Four digits ^{*1} (from 1980 to 2079)	
Month	1 to 12	
Day	1 to 31 (Automatic leap year detection)	
Hour	0 to 23 (24 hours)	
Minute	0 to 59	
Second	0 to 59	
Day of the week	0	Sunday
	1	Monday
	2	Tuesday
	3	Wednesday
	4	Thursday
	5	Friday
	6	Saturday
1/1000 seconds ^{*2}	0 to 999	

*1: Storing in SD213 for the first two digits and SD210 for the last two digits of the year

*2: Use only the expansion clock data read (S(P).DATERD) to read.

 MELSEC-Q/L Programming Manual (Common Instruction)

(4) Changing and reading clock data

(a) Changing clock data

Clock data can be changed either by GX Developer or a program.

1) Changing clock data by GX Developer

Select [Online] → [Set time] to open the Set time screen and change the clock data.

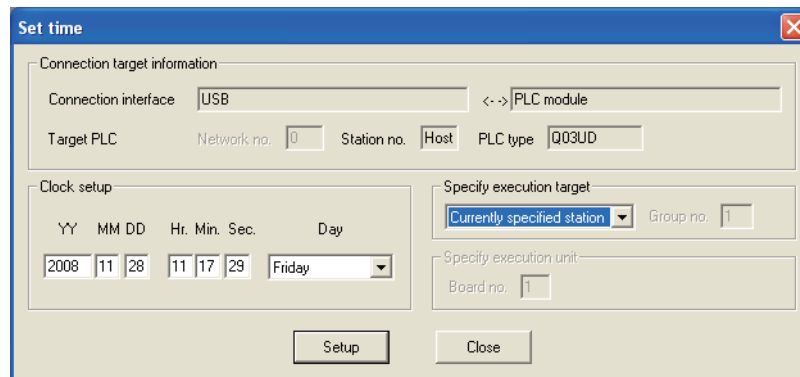


Figure 6.11 Set time screen

2) Changing clock data by a program

Use the DATEWR instruction (instruction for writing clock data) to change the clock data.

Figure 6.12 shows a program for writing the set clock data to D0 to D6.

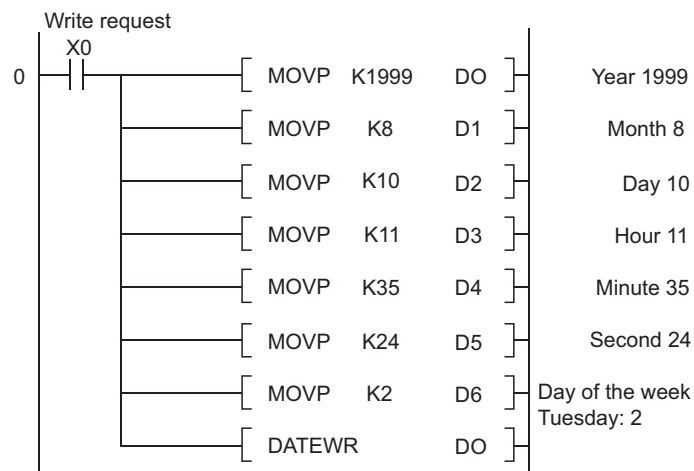


Figure 6.12 Program example for writing clock data

For details of the DATEWR instruction, refer to the following.

 MELSEC-Q/L Programming Manual (Common Instruction)

Point

- When clock data is changed, the clock of 1/1000 second is reset to 0.
- Year data settable by GX Developer is up to 2037.

(b) Reading clock data

To read clock data to the data register, use either of the following instructions in the program.

- DATERD (instruction for reading clock data)
- S(P).DATERD (instruction for reading extended clock data)

Figure 6.13 shows a program for storing the clock data read with the DATERD instruction to D10 to D16.

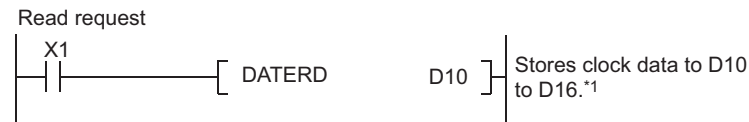


Figure 6.13 Program example for storing clock data

*1: Figure 6.14 shows the clock data stored in D10 to D16.

D10	2004	Year (four digits)
D11	4	Month
D12	1	Date
D13	11	Hour
D14	35	Minute
D15	24	Second
D16	2	Day of the week

(3) in this section

Figure 6.14 Clock data stored

For details of the DATERD and S(P).DATERD instructions, refer to the following.

 MELSEC-Q/L Programming Manual (Common Instruction)

Point

Clock data can also be written or read by the special relay (SM210 to SM213) and special register (SD210 to SD213). For details of the special relay and special register, refer to CHAPTER 12.

(5) Precautions

(a) Initial clock data setting

No clock data is set at the factory.

Clock data is required for some functions of the CPU module used in the system, such as error history storage, or for intelligent function modules.

Before using the CPU module for the first time, set the time correctly.

(b) Clock data correction

If a part of the clock data is corrected, rewrite the entire clock data to the CPU module.

(c) Clock data setting range

When changing clock data, write data within the range given in (3) in this section.

If data outside of clock range is written to the CPU module, the clock function does not operate normally.

However, the CPU module does not detect an error if the clock data is out of the range.

Table6.5 Example of clock data

	Write operation to the CPU module	CPU module operation
February 30	Executed	An error is not detected.
32 of month 13	Not executed	When the DATEWR instruction is executed, "OPERATION ERROR" (error code: 4100) is detected. When SM210 is turned on, SM211 turns on.

(d) Use for clock data of 1/1000 sec.

1) Function that clock data of 1/1000 sec. can be used

Only the following instructions can use the clock data of 1/1000 sec.

- S(P).DATERD
- S(P).DATE+
- S(P).DATE-

Other instructions cannot use the clock data of 1/1000 sec.

(Such as for reading data by SM/SD, storing the time of error occurrence as error history data, reading data by GX Developer, or reading data by dedicated instructions of other modules.)

2) When clock data is changed

When clock data is changed by GX Developer or instructions (including dedicated instruction of other modules), the clock of 1/1000 sec. is reset to 0.

(6) Clock data accuracy

Accuracy of the clock data varies depending on the ambient temperature as shown below.

Table6.6 Clock data accuracy

Ambient temperature (°C)	Accuracy (Day difference, S)
0	-2.96 to +3.74 (TYP.+1.42)
+ 25	-3.18 to +3.74 (TYP.+1.50)
+ 55	-13.20 to +2.12 (TYP.-3.54)

(7) Clock data comparison

To compare clock data in a sequence program, read the clock data with the DATERD instruction (instruction for reading clock data).

Since the DATERD instruction reads the year data in four digits, the data can be compared by the comparison instruction without any modifications.

6.6 Remote Operation

Remote operation allows to change the operating status of the CPU module externally (by GX Developer or external devices using the MC protocol, with link dedicated instructions of the CC-Link IE module or MELSECNET/H module, or using remote contacts).

There are four types of remote operations:

- Remote RUN/STOP :  Section 6.6.1
- Remote PAUSE :  Section 6.6.2
- Remote RESET :  Section 6.6.3
- Remote latch clear :  Section 6.6.4

6.6.1 Remote RUN/STOP

(1) Definition

This operation changes the operating status of the CPU module externally to RUN or STOP, keeping the RUN/STOP/RESET switch of the CPU module in the RUN position.

(2) Application

This operation is useful to run or stop the CPU module remotely when:

- the CPU module is inaccessible, or
- the CPU module is in a control panel.

(3) Program operation

The program operation will be as follows when the remote RUN/STOP operation is performed.

(a) Remote STOP

The CPU module executes a program until the END instruction and changes its operating status to STOP.

(b) Remote RUN

The CPU module changes its operating status to RUN and executes a program from the step 0. (The remote RUN operation must be performed to the CPU module whose operating status has been changed to STOP by the remote STOP operation.)

(4) Executing method

There are three methods for performing the remote RUN/STOP operation.

- Using a RUN contact
- By GX Developer or an external device using the MC protocol
- With link dedicated instructions of the CC-Link IE module or MELSECNET/H module

(a) Using a RUN contact

Set a RUN contact in the PLC system tab of the PLC parameter dialog box.

The settable device range is X0 to 1FFF.

The remote RUN/STOP operation can be performed by turning on/off the set RUN contact.

- When the RUN contact is turned off, the CPU module status changes to RUN.
- When the RUN contact is turned on, the CPU module status changes to STOP.

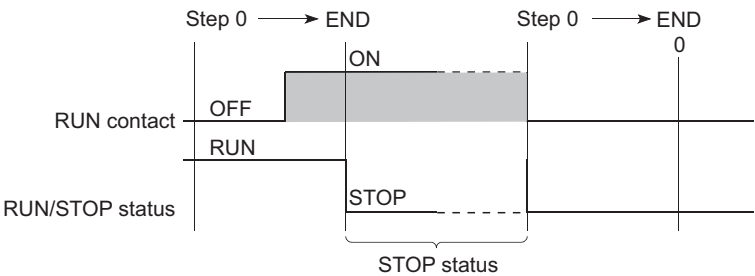


Figure 6.15 Remote RUN/STOP using a RUN contact

(b) By GX Developer or an external device using the MC protocol

Select [Online] → [Remote operation] in GX Developer.

To perform the remote RUN/STOP operation from an external device, use the MC protocol command.

 MELSEC-Q/L MELSEC Communication Protocol Reference Manual

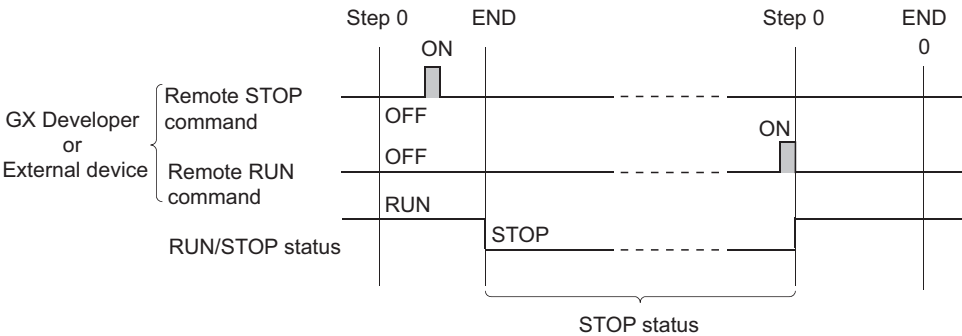


Figure 6.16 Remote RUN/STOP by GX Developer or an external device

(c) With link dedicated instructions of the CC-Link IE module or MELSECNET/H module

The remote RUN/STOP operation by link dedicated instructions of the CC-Link IE module or MELSECNET/H module can change the RUN/STOP status of the CPU module.

For details, refer to the following.

 Reference manual of each network module

(5) Precautions

Pay attention to the following since the STOP status is given priority over the RUN status.

(a) Timing of changing to the STOP status

The operating status of the CPU module is changed to STOP when the remote STOP operation is performed from any one of the following: RUN contact, GX Developer, or an external device using the MC protocol.

(b) When changing the status back to RUN

To change the operating status back to RUN after the CPU module status was changed to STOP by the remote STOP operation, perform the remote RUN operation in the same order for the remote STOP operation.

Point

- The definitions of the RUN/STOP status are described below.
 - RUN status...A status where sequence program operations are repeatedly performed in a loop between the step 0 and the END or FEND instruction.
 - STOP status...A status where sequence program operations are stopped. All outputs (Y) are turned off.

 - After being reset, the operating status of the CPU module becomes the one set with the RUN/STOP/RESET switch.
-

6.6.2 Remote PAUSE

(1) Definition

This operation changes the operating status of the CPU module externally to PAUSE, keeping the RUN/STOP/RESET switch of the CPU module in the RUN position.

PAUSE status is a status where sequence program operations in the CPU module are stopped, holding the status (on or off) of all outputs (Y).

(2) Application

This operation is useful, especially during the process control, to hold the on status of outputs (Y) even after the operating status of the CPU module is switched from RUN to STOP.

(3) Executing method

There are two methods for performing the remote PAUSE operation.

- Using a PAUSE contact
- By GX Developer or an external device using the MC protocol

(a) Using a PAUSE contact

Set a PAUSE contact in the PLC system tab of the PLC parameter dialog box.

The settable device range is X0 to 1FFF.

- The PAUSE contact (SM204) turns on during END processing of the scan where both the PAUSE contact and PAUSE enable coil (SM206) turn on.
The CPU module executes one more scan until the END instruction after the scan where the PAUSE contact turns on, and then changes its operating status to PAUSE. In the PAUSE status, the program operations are stopped.
- When the PAUSE contact or SM206 is turned off, the PAUSE status will be canceled and the CPU module will restart the sequence program operation from the step 0.

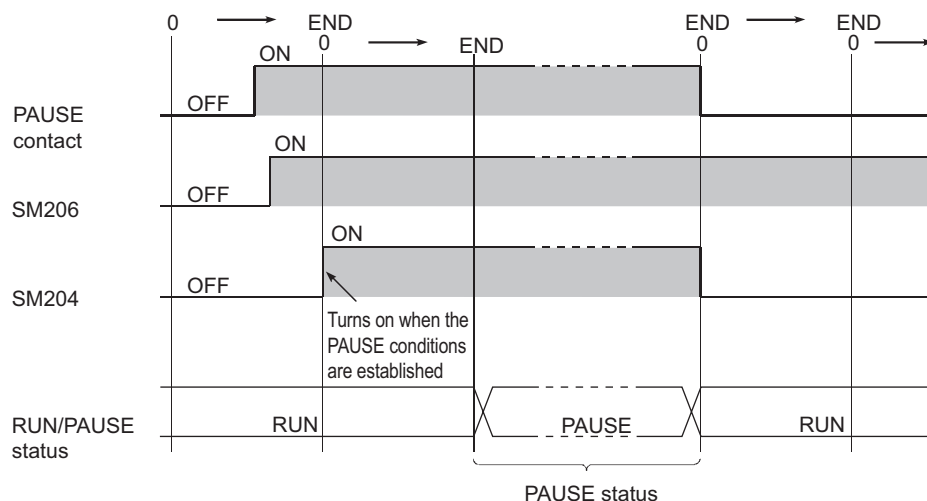


Figure 6.17 Remote PAUSE using a PAUSE contact



Setting of only a PAUSE contact is not allowed.
When setting a PAUSE contact, set a RUN contact as well.

(b) By GX Developer or an external device using the MC protocol

Select [Online] → [Remote operation] in GX Developer.

To perform the remote PAUSE operation from an external device, use the MC protocol command.

📖 MELSEC-Q/L MELSEC Communication Protocol Reference Manual

- The PAUSE contact (SM204) turns on during END processing of the scan where the remote PAUSE command is executed.
The CPU module executes one more scan until the END instruction after the scan where the PAUSE contact turns on, and then changes its operating status to PAUSE. In the PAUSE status, the program operations are stopped.
- Upon execution of the remote RUN command, the CPU module will restart the sequence program operations from the step 0.

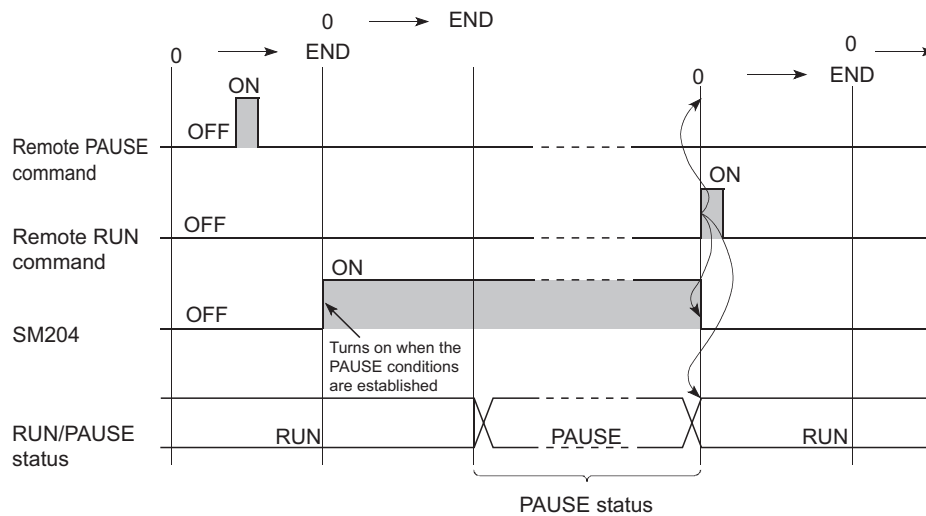


Figure 6.18 Remote PAUSE by GX Developer or an external device

(4) Precautions

(a) When forcibly keeping output status

To forcibly keep the output status (on or off) in the PAUSE status, provide an interlock with the PAUSE contact (SM204).

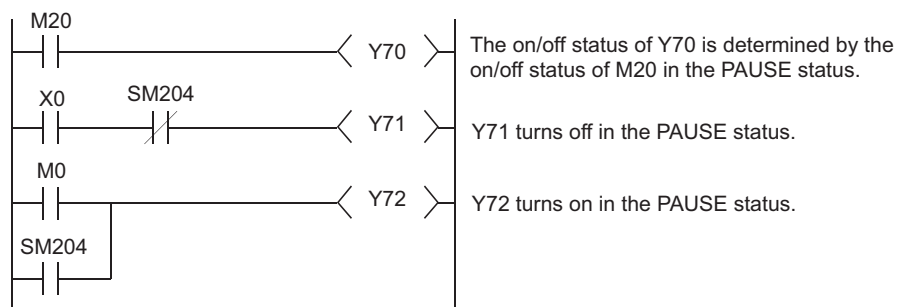


Figure 6.19 Program example for forcibly keeping output status in the PAUSE status

6.6.3 Remote RESET

(1) Definition

This operation resets the CPU module externally when the CPU module is in the STOP status.

Even if the RUN/STOP/RESET switch is in the RUN position, this operation can be performed when the module is stopped due to an error detected by the self-diagnostics function.

(2) Application

This operation is useful to reset the CPU module remotely when an error occurs in the CPU module placed in an inaccessible location.

(3) Executing method

This operation can only be performed by GX Developer or an external device using the MC protocol.

- Select [Online] → [Remote operation] in GX Developer.
- To perform the remote RESET operation from an external device, use the MC protocol command.

 MELSEC-Q/L MELSEC Communication Protocol Reference Manual

Point

Before performing the remote RESET operation, select the "Allow" checkbox for the remote RESET operation in the PLC system tab of the PLC parameter dialog box, and then write the parameter setting to the CPU module.

Without the preset parameter setting, the operation cannot be performed.

Select the checkbox here to enable the remote RESET operation.

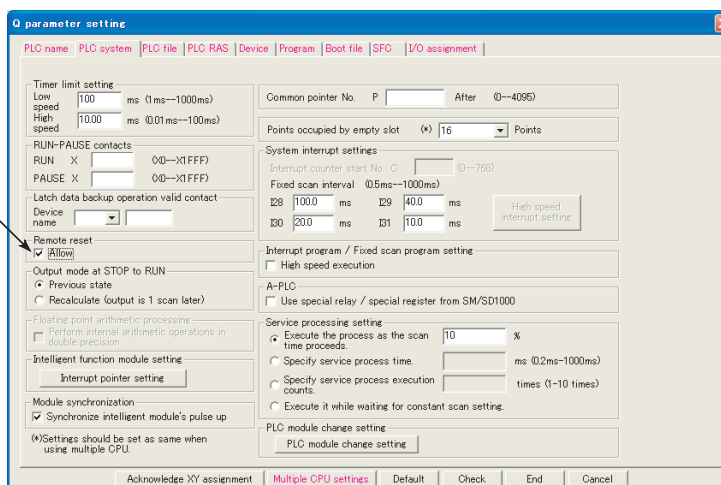


Figure 6.20 Parameter setting required for the remote RESET operation

(4) Precautions

(a) Remote RESET in the RUN status

When the CPU module is in the RUN status, the remote RESET operation cannot be performed.
To perform the operation, change the operating status of the CPU module to STOP by the remote STOP or similar operation.

(b) Status after reset processing

After reset processing of the remote RESET operation is completed, the CPU module will be placed in the operating status set by the RUN/STOP/RESET switch.

If the RUN/STOP/RESET switch is set to STOP, the CPU module will be in the STOP status. If the switch is set to RUN, the CPU module will be in the RUN status.

Point

- If the remote RESET operation is performed to the CPU module which is stopped due to an error, note that the CPU module will be placed in the operating status set by the RUN/STOP/RESET switch after reset processing is completed.

 - If the CPU module cannot be reset by the remote RESET operation from GX Developer, check that the "Allow" checkbox for the remote RESET operation in the PLC system tab of the PLC parameter dialog box is selected.
If the checkbox is not selected, the CPU module cannot be reset even after the remote RESET processing from GX Developer is completed.
-

(c) When an error occurs due to noise

Note that the CPU module may not be reset by the remote RESET operation from GX Developer.
In this case, reset the CPU module using the RUN/STOP/RESET switch or power off and then on the CPU module.

6.6.4 Remote latch clear

(1) Definition

This function resets the latched device data from GX Developer or an external device when the CPU module is in the STOP status.

(2) Application

This function is useful in the following cases if used together with the remote RUN/STOP operation.

- When the CPU module is inaccessible
- To clear latched device data in the CPU module in a control panel externally

(3) Executing method

This operation can only be performed by GX Developer or an external device using the MC protocol.

- Select [Online] → [Remote operation] in GX Developer.
- To execute the remote latch clear operation from an external device, use the MC protocol command.

 MELSEC-Q/L MELSEC Communication Protocol Reference Manual

To perform the remote latch clear operation, follow the following steps.

- 1) Change the operating status of the CPU module to STOP by the remote STOP operation.
- 2) Clear the latched device data in the CPU module by the remote latch clear operation.
- 3) After remote latch clear processing is completed, perform the remote RUN operation to return the operating status to RUN.

(4) Precautions

(a) Latch clear in the RUN status

The latch clear operation cannot be performed when the CPU module is in the RUN status.

(b) Latch clear enabled range

There are two kinds of latch range can be set in the Device tab of the PLC parameter dialog box: latch clear operation enable and disable range.

Remote latch clear operation resets only the data set in the "Latch (1)" (latch clear operation enable range).

For the method for resetting the device data in the latch clear operation disable range, refer to Section 3.7(2)(c).

(c) Devices that are reset by the remote latch clear operation

Devices that are not latched are also reset when the remote latch clear operation is performed.

6.6.5 Relationship between remote operation and RUN/STOP status of the CPU module

(1) Relationship between remote operation and RUN/STOP status of the CPU module

Table6.7 shows the operating status of the CPU module according to the combination of remote operation and RUN/STOP status.

Table6.7 Operating status of the CPU module

RUN/STOP status	Remote operation				
	RUN ^{*1}	STOP	PAUSE ^{*2}	RESET ^{*3}	Latch clear
RUN	RUN	STOP	PAUSE	Operation disabled ^{*4}	Operation disabled ^{*4}
STOP	STOP	STOP	STOP	RESET ^{*5}	Latch clear

*1: When performing the operation using a RUN contact, "RUN-PAUSE contact" must be set in the PLC system tab of the PLC parameter dialog box.

*2: When performing the operation using a PAUSE contact, "RUN-PAUSE contact" must be set in the PLC system tab of the PLC parameter dialog box.

In addition, the PAUSE enable coil (SM206) must be turned on.

*3: The "Allow" checkbox for the remote RESET operation must be selected in the PLC system tab of the PLC parameter dialog box.

*4: The remote RESET and remote latch clear operations are enabled if the CPU module status is changed to STOP by the remote STOP operation.

*5: The STOP status includes a case where the CPU module is stopped due to an error.

(2) Remote operations from a single GX Developer

When remote operations are performed from a single GX Developer, the operating status of the CPU module will be the status of the last remote operation performed.

(3) Remote operations from multiple GX Developers

Any remote operation from other GX Developers via other stations cannot be performed to the CPU module where remote operations are being performed from GX Developer connected.

To perform any remote operations from another GX Developer, cancel the remote operation by performing the remote RUN operation from the same GX Developer that is performing the current remote operation.

For example, even if the remote STOP or RUN operation is performed from another GX Developer to the CPU module where the remote PAUSE operation has been performed by GX Developer connected, the CPU module remains in the PAUSE status.

Once after the remote RUN operation is performed from the same GX Developer that is performing the remote PAUSE operation, remote operations from another GX Developer will be enabled.

6.7 Q Series-compatible Module Input Response Time Selection (I/O Response Time)

(1) Definition

This function changes the input response time for each Q series-compatible module.

Table6.8 shows the modules available for input response time change and selectable time settings.

Table6.8 Modules available for input response time change

Module name	Type	Settable time setting
Input module	"Input"	1ms, 5ms, 10ms, 20ms, 70ms
I/O combined module	"I/O Mix"	(Default: 10ms)
High-speed input module	"Hi Input"	0.1ms, 0.2ms, 0.4ms, 0.6ms, 1ms
Interrupt module	"Interrupt"	(Default: 0.2ms)

The Q series-compatible modules in the table above take in external inputs within the set time here.

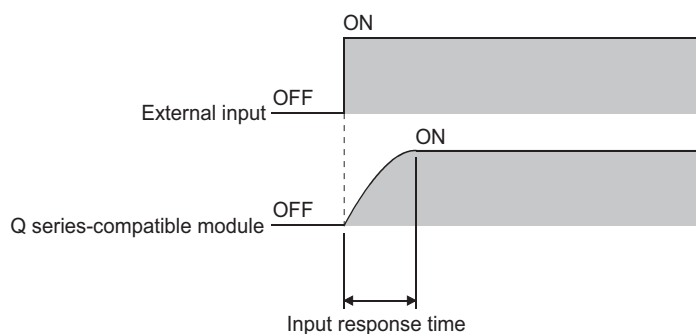


Figure 6.21 Input response time

(2) Input response time setting

Set input response time values in the I/O assignment tab of the PLC parameter dialog box.

- 1) Make I/O assignment for the target module.
- 2) Click the **Detailed setting** button.
- 3) On the screen opened, select an input response time value ("I/O response time").

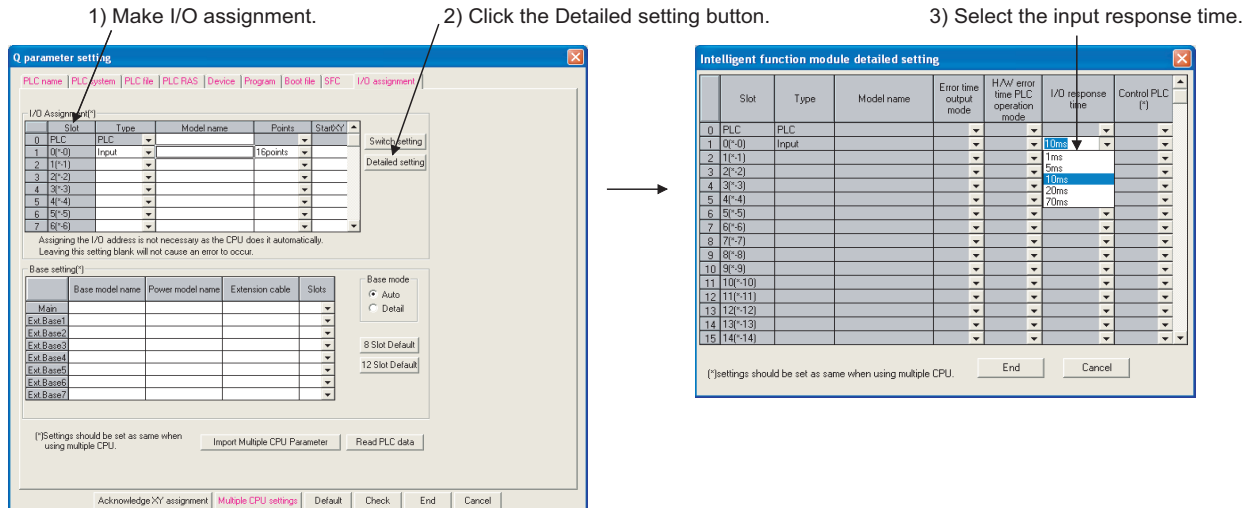


Figure 6.22 Input response time setting

(3) Precautions

(a) When input response time is shortened

The shorter the input response time is, the more the CPU module is susceptible to noise. Consider the operating environment when setting input response time values.

(b) Enabling the setting

The input response time setting will be enabled when:

- the CPU module is powered off and then on, or
- the CPU module is reset.

6.8 Error Time Output Mode Setting

(1) Definition

This function determines the output mode (clear or hold) from the CPU module to the Q series-compatible output modules, I/O combined modules, intelligent function modules, and/or interrupt module when a stop error occurs in the CPU module.

(2) Error time output mode setting

Set the error time output mode in the I/O assignment tab of the PLC parameter dialog box.

- 1) Make I/O assignment for the target module.
- 2) Click the **Detailed setting** button.
- 3) On the screen opened, select "Clear" or "Hold". (Default: "Clear")

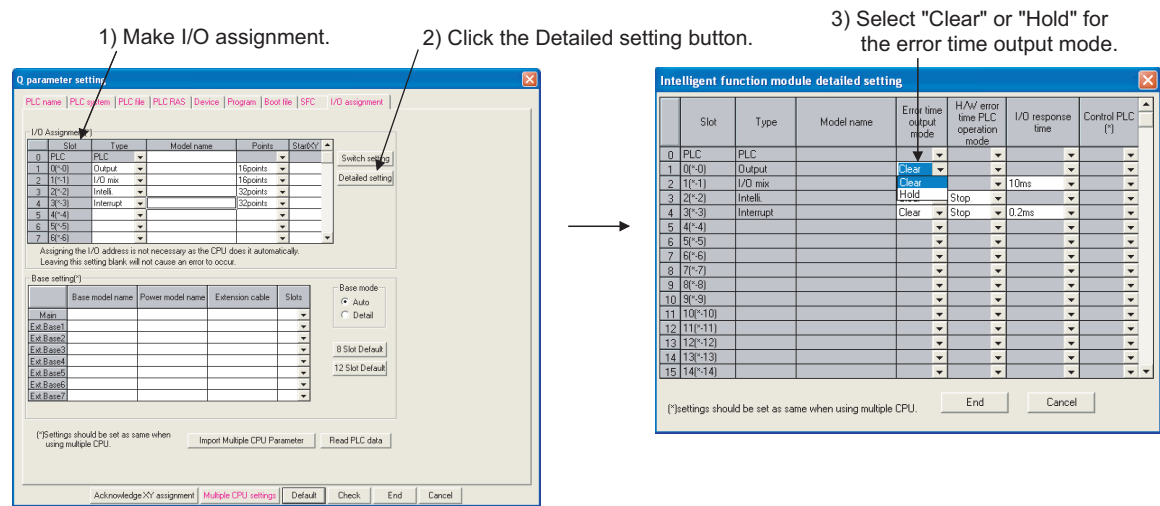


Figure 6.23 Error time output mode setting

(3) Precautions

The error time output setting will be enabled when:

- the CPU module is powered off and then on, or
- the CPU module is reset.

Failure to perform either of the operations above after changing the error time output mode setting will result in "PARAMETER ERROR" (error code: 3000).

6.9 H/W Error Time PLC Operation Mode Setting

(1) Definition

This setting determines whether to stop or continue the CPU module operation when a hardware error (CPU module detects SP.UNIT DOWN) occurs in the intelligent function module or the interrupt module.

(2) H/W error time PLC operation mode setting

Set the H/W error time PLC operation mode in the I/O assignment tab of the PLC parameter dialog box.

- 1) Make I/O assignment for the target module.
- 2) Click the **Detailed setting** button.
- 3) On the screen opened, select "Stop" or "Continue". (Default: "Stop")

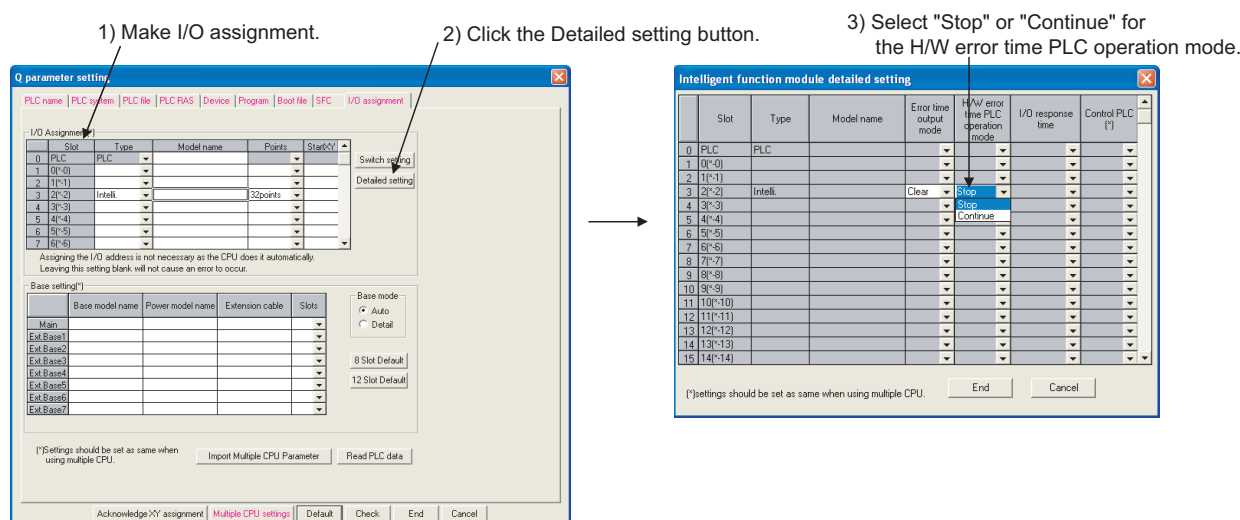


Figure 6.24 H/W error time PLC operation setting

(3) Precautions

The H/W error time PLC operation setting will be enabled when:

- the CPU module is powered off and then on, or
- the CPU module is reset.

6.10 Intelligent Function Module Switch Setting

(1) Definition

This function sets the switches of each Q series-compatible intelligent function module and interrupt module in GX Developer.

(2) Writing the switch settings

The switch settings will be written from the CPU module to each intelligent function module and interrupt module when:

- the CPU module is powered off and then on, or
- the CPU module is reset.

2) The settings are written when the CPU module is powered off and then on or reset.

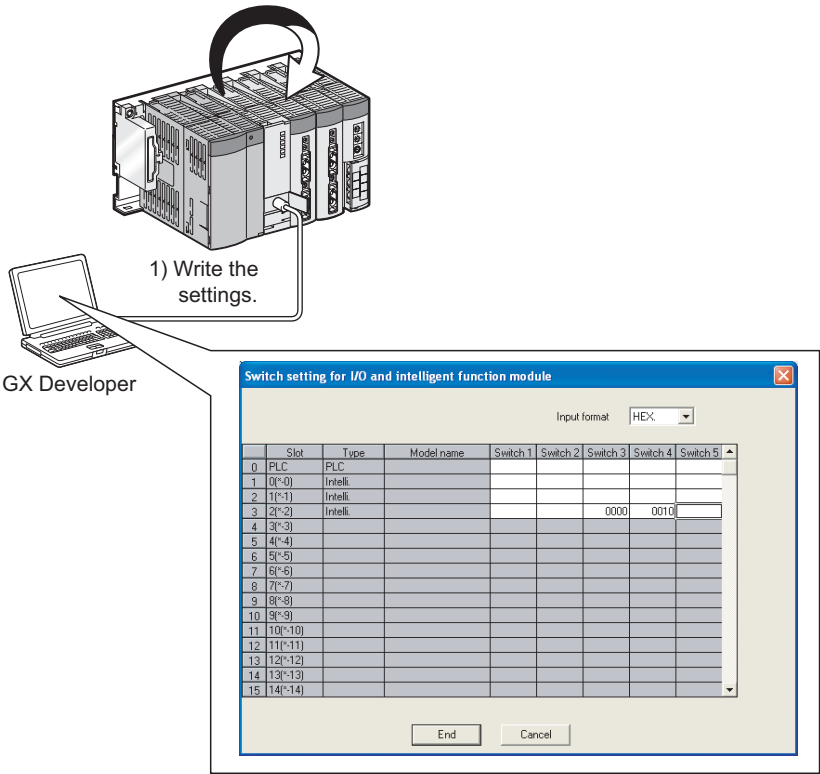


Figure 6.25 Writing the switch settings to each intelligent function module

(3) Switch settings

Set the switch details for each intelligent function module and interrupt module in the I/O assignment tab of the PLC parameter dialog box.

- 1) Make I/O assignment for the target module.
- 2) Click the **Switch setting** button.
- 3) Set the switch details for each module.

- 1) Make I/O assignment.
- 2) Click the Switch setting button.

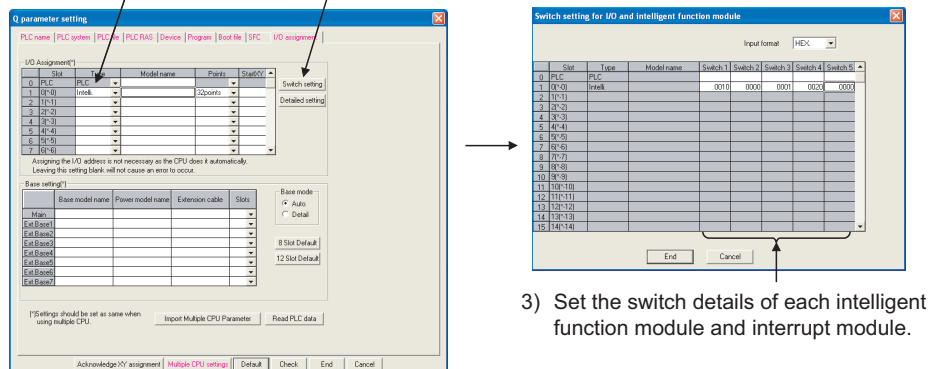


Figure 6.26 Switch settings

(4) Precautions

(a) Switch setting details of each module

For the switch setting details of each intelligent function module or interrupt module, refer to the manual for the intelligent function module or interrupt module used.

(b) Enabling the setting

The switch settings of each intelligent function module or interrupt module will be enabled when:

- the CPU module is powered off and then on, or
- the CPU module is reset.

6.11 Monitor Function

(1) Definition

This function reads program and device data in the CPU module, and intelligent function module status using GX Developer.

Table6.9 List of monitor functions and availability

Monitor function	Availability					Reference
	Q00UJ CPU	Q00UCPU, Q01UCPU	Q02UCPU	QnUD(H) CPU	Built-in Ethernet port QCPU	
Ladder monitor	○	○	○	○	○	GX Developer Version 8 Operating Manual
Device/buffer memory batch monitor	○	○	○	○	○	
Entry data monitor	○	○	○	○	○	
Device test	○	○	○	○	○	
Entry ladder monitor	○	○	○	○	○	
Monitor condition setting	×	×	△ ^{*1}	△ ^{*1}	○	Section 6.11.1
Local device monitor/test	×	○	○	○	○	Section 6.11.2
External input/output forced on/off	○	○	△ ^{*1}	△ ^{*1}	○	Section 6.11.3
Executorial conditioned device test	○	○	△ ^{*1}	△ ^{*1}	○	Section 6.11.4

○ : Available, △ : Available with restrictions, × : Not available

*1: Availability depends on the version of the CPU module. (☞ Appendix 2)

(2) Monitor request timing and displayed data

The CPU module processes monitor requests from GX Developer during END processing. For this reason, the data in the CPU module at the time of END processing will be displayed in GX Developer.

(3) Monitor with monitor condition settings

By setting the monitor condition in GX Developer during debugging, the program operation status in the CPU module can be monitored under the specified condition. Besides, the monitoring status under the specified condition can be held by setting the monitoring stop condition.

(4) Local device monitor

If multiple programs are executed and local devices are used, data in local devices of each program can also be monitored.

6.11.1 Monitor condition setting Note6.1

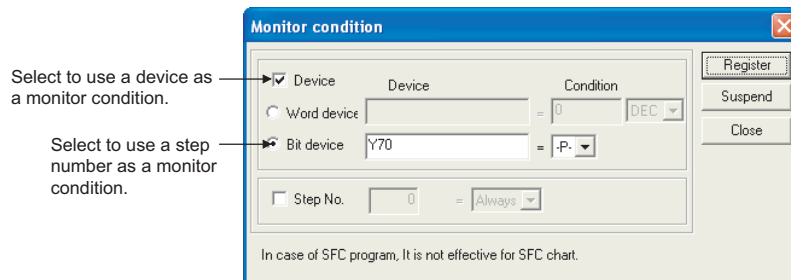
This function is used to monitor data in the CPU module under the specified condition.

(1) Monitor condition setting for ladder monitor

Switch GX Developer into monitor mode.

Select [Online] → [Monitor] → [Monitor condition setup] to open the Monitor condition screen.

Set the condition as shown below to monitor data on the rising edge of Y70.



Monitoring starts when the condition is established.
(An operation status when the condition is established is displayed.)

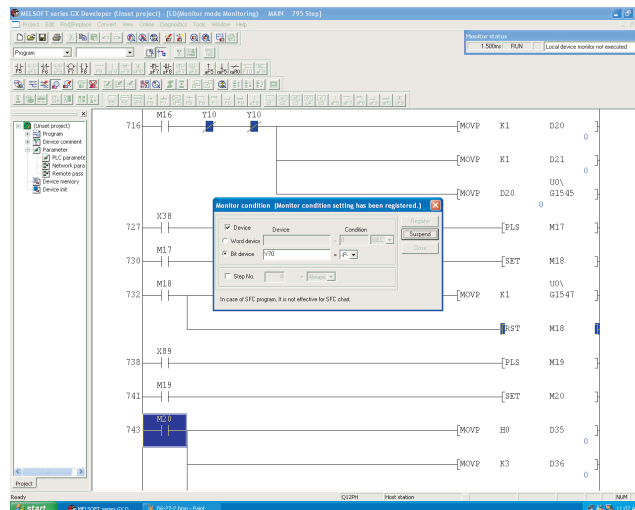


Figure 6.27 Monitor condition setting screen

(a) When only a step number is specified

Monitor data is collected when the status immediately before execution of the specified step becomes the specified status.

The following status can be specified.

- When the operation of the specified step changes from the non-execution status to the execution status: <-P->
- When the operation of the specified step changes from the execution status to the non-execution status: <-F->
- Always only when the operation of the specified step is in execution: <ON>
- Always only when the operation of the specified step is in non-execution: <OFF>
- Always regardless of the status of the operation of the specified step: <Always>

Note6.1

Universal

The Q00JCPU, Q00UCPU, and Q01UCPU do not support the monitor condition setting.

When setting monitor condition for the Q02UCPU, Q03UDCPU, Q04UDHCPU, Q06UDHCPU, Q13UDHCPU, or

Q26UDHCPU, check the versions of the CPU module and GX Developer. ( Appendix 2)



- If a step between the AND/OR blocks is specified as a monitor condition, monitor data is collected when the status previous to execution of the specified step is specified by the LD instruction.
The monitor timing depends on the step specified as a monitor condition.
The following shows examples of monitoring when the step 2 is on (Step No. [2] = <ON>).

- When the step 2 is connected by the AND instruction:
In Figure 6.28, the monitor execution condition is established when both X0 and X1 are on.

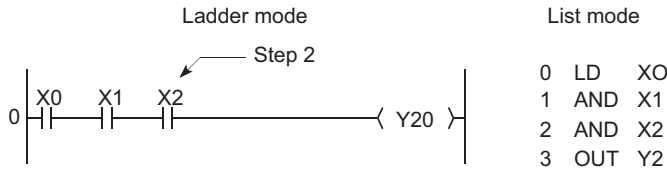


Figure 6.28 When the step 2 is connected by the AND instruction

- When the step 2 is connected in the middle of the AND/OR block:
In Figure 6.29, the monitor execution condition is established when X1 turns on. (The on/off status of X0 does not affect the establishment of the monitor execution condition.)

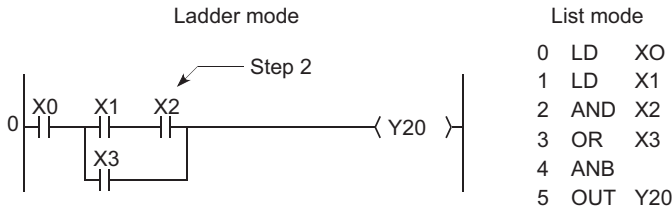


Figure 6.29 When the step 2 is connected in the middle of the AND/OR block

- If the start of a ladder block other than the step 0 is specified for the step number as a detailed condition, monitor data is collected when the execution status of the instruction immediately before execution becomes the specified status.
If (Step No. [2] = <ON>) is specified in the following ladder, monitor data is collected when OUT Y10 turns on.

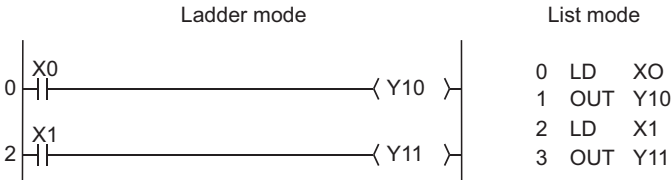


Figure 6.30 When the start of a ladder block other than the step 0 is specified for the step number

- Be sure to set the condition of the step set as step No.0 to "Always".

(b) When only a device is specified

Either word device or bit device can be specified.

1) When a word device is specified

Monitor data is collected when the current value of the specified word device becomes the specified value.
Enter the current value (in decimal or hexadecimal).

2) When a bit device is specified

Monitor data is collected when the execution status of the specified bit device becomes the specified status.
Select the execution condition (on the rising edge or falling edge).

(c) When a step number and device are specified

Monitor data is collected when the status previous to execution of the specified status or the status (current value) of the specified bit device (word device) becomes the specified value.

Point

When "Step No.[100]=<-P->, Word device [D1]=[K5]" is specified as an execution condition, a monitor execution condition is established on the rising edge of the step 100 and also D1=5.



Figure 6.31 When the rising edge of the step 100 and D1=5 are specified

The monitor interval of GX Developer depends on the processing speed of GX Developer.

For the monitor execution conditions established at the interval shorter than the monitor interval of GX Developer, monitor is executed only when the monitor execution condition is established at the monitor timing of GX Developer.

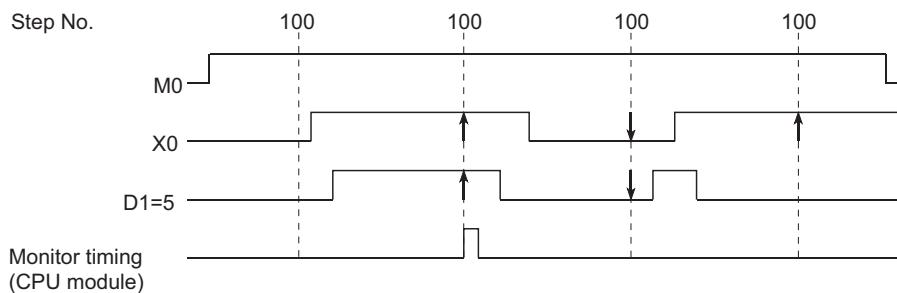


Figure 6.32 Monitor timing of GX Developer

(2) Monitor stop condition setting

Set a monitor stop condition on the screen opened by selecting [Online] → [Monitor] → [Monitor stop condition setup].

Set the condition as shown in Figure 6.33 to stop a monitor operation on the rising edged of Y71.

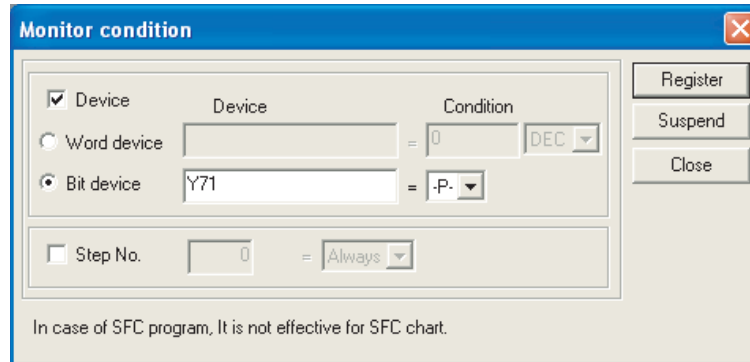


Figure 6.33 Monitor stop condition screen

(a) When a device is specified

Either word device or bit device can be specified.

1) When a word device is specified

A monitor operation is stopped when the current value of the specified word device becomes the specified value.

Enter the current value (in decimal, hexadecimal, 16-bit integer, 32-bit integer, or real number).

2) When a bit device is specified

A monitor operation is stopped when the execution status of the specified bit device becomes the specified status.

Select the execution condition (on the rising edge or falling edge).

(b) When a step number is specified

A monitor operation is stopped when the execution status of the step specified as a monitor condition becomes the specified status.

The following status can be specified.

- When the operation of the specified step changes from the non-execution status to the execution status: <-P->
- When the operation of the specified step changes from the execution status to the non-execution status: <-F->
- Always only when the operation of the specified step is in execution: <ON>
- Always only when the operation of the specified step is in non-execution: <OFF>
- Always regardless of the status of the operation of the specified step: <Always>

If a step number is not specified, a monitor operation is stopped after END processing of the CPU module.

(3) Precautions

(a) Files to be monitored

When monitor conditions are set, GX Developer monitors the file displayed on the screen.

Select [Online] → [Read from PLC] in GX Developer and read data from the CPU module so that the file name in the CPU module to be monitored matches the file named displayed on the screen of GX Developer.

(b) No file register setting

If the file register is monitored when there is no file register used, "FFFFH" is displayed.

(c) Device assignment

For monitor operation, device assignment in the CPU module and GX Developer must be the same.

(d) Monitoring the buffer memory of an intelligent function module

When monitoring the buffer memory of an intelligent function module, the scan time increases for the same reason for execution of the FROM/TO instructions.

(e) Monitoring by multiple users

When multiple users are performing monitoring at the same time, pay attention to the following.

- High speed monitor can be performed by increasing 1K step per monitor file of other stations in the system area when formatting the program memory or setting a parameter in the Boot file tab of the PLC parameter dialog box.
- Up to 15 stations can be set as the station monitor file, but the program space will be reduced.
- If the monitor condition or monitor stop condition is set, only one user can perform monitoring.

(f) Setting a monitor stop condition

A monitor stop condition can be set only in the ladder monitor.

(g) Specifying the same device as a condition

When specifying the same device as a monitor condition or monitor stop condition, set the on/off status as well.

(h) Specifying a step number as a monitor condition

If an instruction in the specified step is not executed in such cases described below, the monitor condition will not be established.

- The specified step is skipped with the CJ, SCJ, or JMP instruction.
- The specified step is the END instruction and never be executed because the FEND instruction also exists in the program.

(i) During monitor condition registration

Do not reset the CPU module while monitoring conditions are being registered.

(j) Monitor operation with monitor condition setting

When monitor operation with monitor condition setting is performed, other applications on the same personal computer cannot execute any online function using the same route for the monitor operation. The following applications must be noted.

- GX Developer
- Application using MX Component
- MX Sheet

If any online function is executed by other applications using the same route for the monitor operation, the following situations may occur.

- No response is returned from the CPU module for the online function executed. (An online communication function time-out occurs.)
- The CPU module detects an error (error code: 4109) for the online function executed.
- Even when the monitor condition is established in the CPU module, monitoring results cannot be updated for the monitor operation with monitor condition setting.

6.11.2 Local device monitor/test Note6.2

This operation is useful for debugging a program, monitoring local devices ( Section 9.14.2) in the program monitored by GX Developer.

(1) Monitoring a local device

Table6.10 shows the monitor operation when the CPU module executes three programs "A", "B", and "C" and D0 to D99 are set as a local device.

(Three programs are to be executed in the order of A→B→C→(END processing)→A→B....)

Table6.10 Data displayed when three programs are executed

Setting	Monitored device	
	D0 (Local device)	D100 (Global device)
Local device monitor is set	The D0 value in the specified program (local device for a program) is monitored.*1	The D100 value after execution of the specified program is monitored.*2
Local device monitor is not set	The D0 value after execution of the program "C" is monitored.	The D100 value after execution of the program "C" is monitored.

*1: When "Not used" is set for "Local device" in File usability setting of the Program tab, the D0 value after execution of the specified program is monitored.

*2: For Built-in Ethernet port QCPU whose serial number (first 5 digits) is "11013" to "12051", when the local device is monitored from built-in Ethernet port, the D100 value after execution of the program "C" is monitored.



Note6.2

Universal

The Q00UJCPU does not support the use of local devices.

When local devices are set to be monitored and the program "B" is displayed for monitoring, the local device(s) used in the program "B" can be monitored.

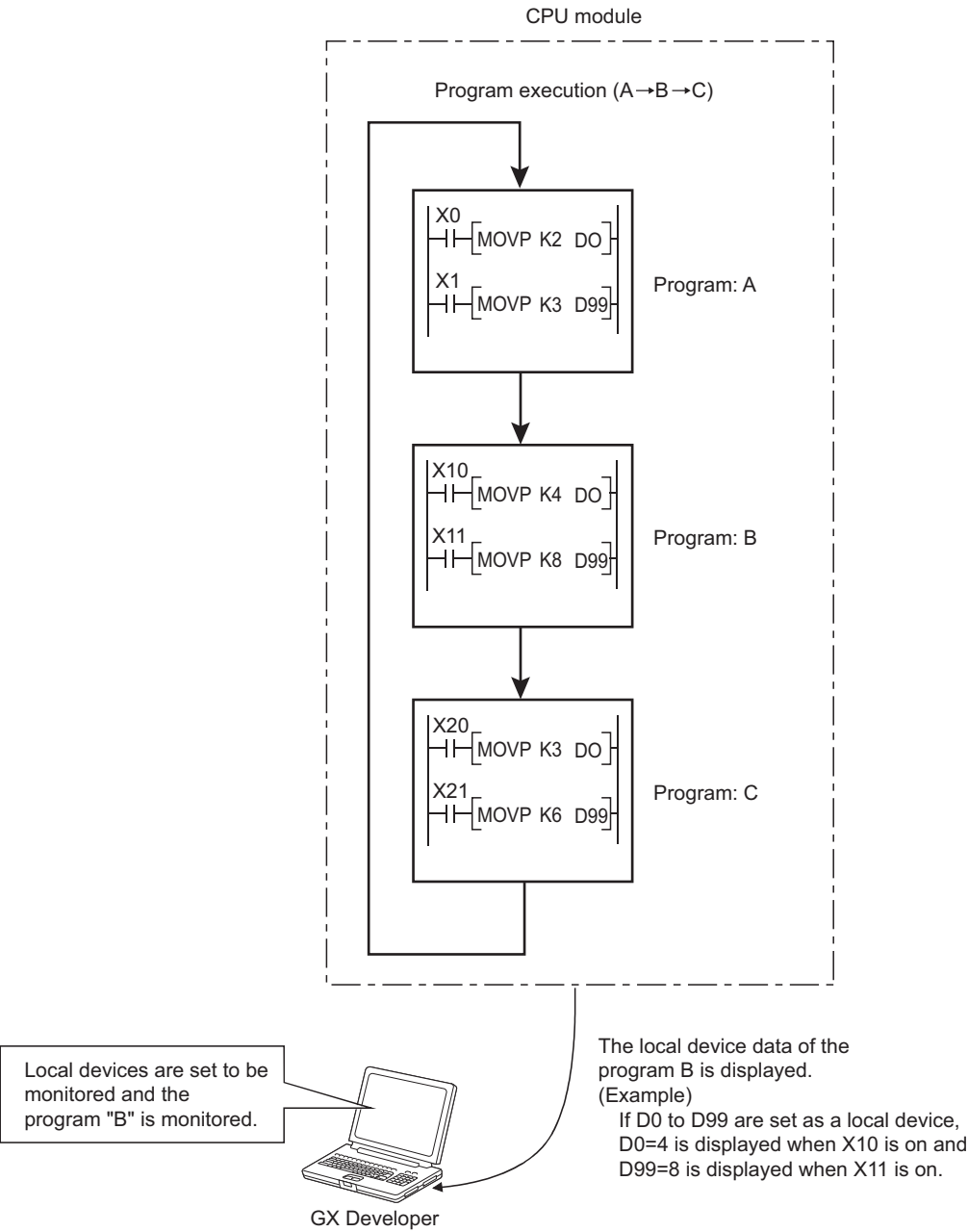


Figure 6.34 Local device monitor example

(2) Monitoring procedure

The following shows the local device monitoring procedure.

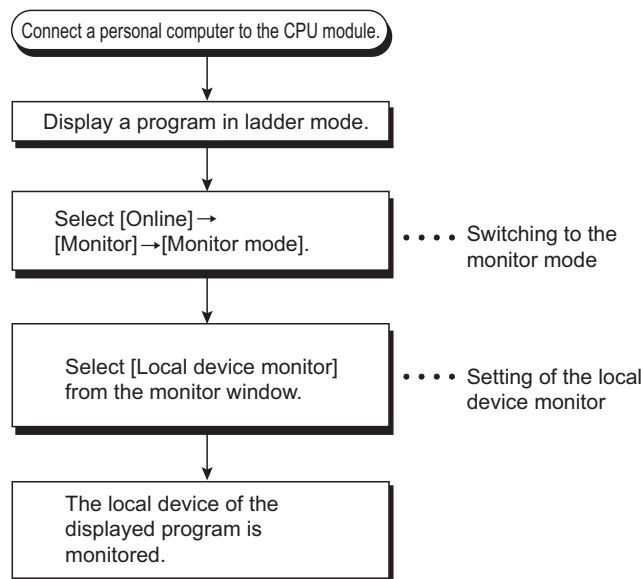


Figure 6.35 Local device monitoring procedure

(3) Precautions

(a) Local devices that can be monitored/tested by a single GX Developer

A single GX Developer can monitor or test local devices in one program at a time. Local devices in multiple programs cannot be monitored or tested simultaneously.

(b) Number of programs that can be monitored/tested

Local devices in 16 programs can be monitored or tested simultaneously from multiple GX Developers connected to the RS-232 interface of the CPU module or the serial communication module.

(c) Monitoring local devices in a stand-by type program

When local devices in a stand-by type program are monitored, data in local devices are saved and restored. For this reason, the scan time increases. (☞ Section 9.14.2)

(d) Monitoring local devices in a fixed scan execution type program

When local devices in a fixed scan execution type program are monitored, data in local devices cannot be acquired and "0" is displayed.

(e) Clearing the PLC memory during local device monitoring

Local devices of monitoring programs are the target for the PLC memory clear operation during local device monitoring.

6.11.3 External input/output forced on/off Note6.3

The external input/output can forcibly be turned on/off on the screen opened by selecting [Online] → [Debug] → [Forced input output registration/cancellation] in GX Developer.

The information registered for forced on/off can be cancelled by an operation from GX Developer.

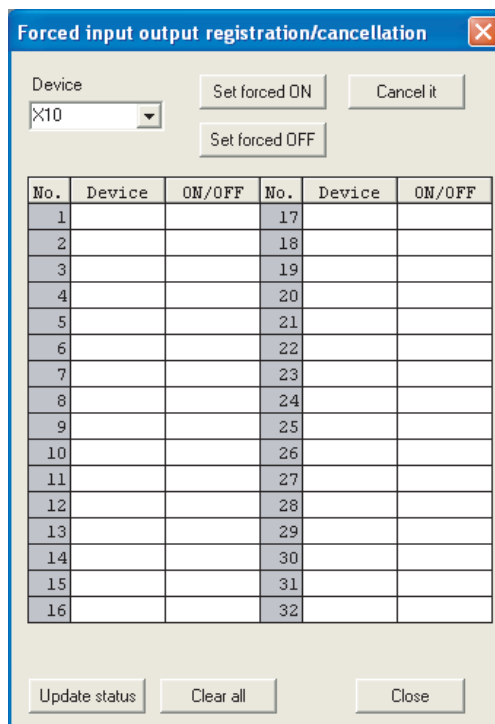


Figure 6.36 Forced input output registration/cancellation screen

(1) Input/output operation when a forced on/off operation is performed

There are three kinds of forced on/off operations: forced on ("Set forced ON"), forced off ("Set forced OFF"), and forced on/off cancellation ("Cancel it").

Table6.11 shows the CPU module operation when a forced on/off operation is performed.

Table6.11 Input/output operation when a forced on/off operation is performed

Operation	Input (X) operation	Output (Y) operation
Forced on/off cancellation (no operation)	The CPU module performs sequence program operations using external inputs.	The CPU module outputs the results of sequence program operations externally.
Forced on	The CPU module performs sequence program operations using inputs forcibly turned on.	The CPU module outputs "on" externally regardless of the results of sequence program operations.
Forced off	The CPU module performs sequence program operations using inputs forcibly turned off.	The CPU module outputs "off" externally regardless of the results of sequence program operations.



Note6.3

Universal

When performing an external input/output forced on/off operation for the Q02UCPU, Q03UDCPU, Q04UDHCPU, Q06UDHCPU, Q13UDHCPU, or Q26UDHCPU, check the versions of the CPU module and GX Developer. ( Appendix 2)

Figure 6.37 shows the input/output operation when a forced on/off operation is performed.

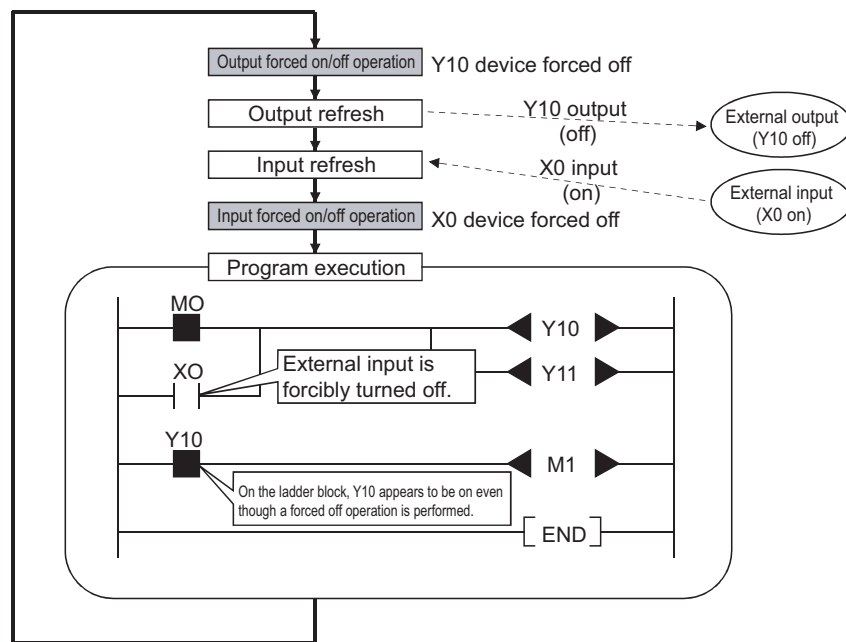


Figure 6.37 Input/output operation when a forced on/off operation is performed

(2) Specifications

(a) CPU module status where input/output can forcibly be turned on/off

Forced on/off can be registered regardless of the operating status (RUN/STOP) of the CPU module.

Note, however, that only input can be forcibly turned on/off during a stop error.

The CPU module outputs on/off data only to Y device.

(b) Registerable devices

Forced on/of can be registered as many as the number of I/O device points in the CPU module.

(c) Target input/output

The following input/output are targeted for a forced on/off operation.

- Input(X) and output (Y) of modules mounted on the base unit
- Input(X) and output (Y) of the CPU module to be refreshed from LX/LY of a CC-Link IE controller network module or MELSECNET/H module
- Input(X) and output (Y) of the CPU module to be refreshed from RX/RX of a CC-Link IE Field network master/local module or CC-Link module

When forcibly turning on/off the devices outside the above refresh ranges (for example, empty slots), only input/output in the CPU module device memory are turned on/off and the results are not output externally.



In multiple CPU systems, inputs and outputs of control modules can forcibly turned on/off.

Even when inputs and outputs of non-control modules are registered for forced on/off, the input/output devices in other CPU modules and inputs and outputs of modules controlled by other CPU modules cannot be forcibly turned on/off. (The input/output devices in the host CPU module can forcibly turned on/off.)

(d) Cancelling on/off registration data

The registered forced ON/OFF data can be canceled by GX Developer.

Once the registered data is canceled, the status of the forced on/off registered devices will be as follows.

Table6.12 Status of devices after forced on/off registration data is canceled

Forced on/off registered device		Sequence program operations (on/off) performed	Sequence program operations (on/off) not performed
Input	Input from modules mounted on the base unit	Uses the on/off status input from modules.	
	Input of the CPU module to be refreshed from LX of a CC-Link IE controller network module or MELSECNET/H module	Used the on/off status refreshed via CC-Link IE controller network module or MELSECNET/H module.	
	Input of the CPU module to be refreshed from RX of a CC-Link IE Field network master/local module or CC-Link module	Uses the on/off status refreshed via CC-Link IE Field network master/local module or CC-Link.	
	Input other than above (outside of the refresh range)	Uses the results of sequence program operations.	Holds the forced on/off status.
Output	Output from modules mounted on the base unit	Outputs the results of sequence program operations.	Holds the registered on/off status.
	Output of the CPU module to be refreshed from LY of a CC-Link IE controller network module or MELSECNET/H module		
	Output of the CPU module to be refreshed from RY of a CC-Link IE Field network master/local module or CC-Link module		
	Output other than above (outside of the refresh range)	Outputs the results of sequence program operations. (The results are not output externally.)	Holds the forced on/off status.

Forced on/off setting can be cleared by:

- powering off and then on the CPU module,
- resetting the CPU module by the RUN/STOP/RESET switch, or
- resetting the CPU module by the remote RESET operation.

(e) External input/output forced on/off timing

Table6.13 shows the external input/output forced on/off timing.

Table6.13 Forced on/off timing

Refresh area	Input	Output
Input and output of modules mounted on the base unit	• During END processing (input refresh) • At execution of the COM instruction (input refresh) • At execution of an instruction using direct access input (DX) (LD, LDI, AND, ANI, OR, ORI, LDP, LDF, ANDP, ANDF, ORP, ORF) • At execution of the RFS or MTR instruction • At execution of an instruction used for a system interrupt (UDCNT1, UDCNT2, SPD)	• During END processing (output refresh) • At execution of the COM instruction (output refresh) • At execution of an instruction using direct access input (DX) (OUT, SET, DELTA, RST, PLS, PLF, FF, MC, SFT) • At execution of the RFS or MTR instruction • At execution of an instruction used for a system interrupt (PLSY, PWM)
Input and output of the CPU module to be refreshed from LX/LY of a CC-Link IE controller network module or MELSECNET/H module	• During END processing (refresh via CC-Link IE controller network or MELSECNET/H) • At execution of the COM instruction • At execution of the ZCOM instruction	
Input and output of the CPU module to be refreshed from RX/RX of a CC-Link IE Field network master/local module or CC-Link module	• During END processing (auto refresh) • At execution of the COM instruction (auto refresh) • At execution of the ZCOM instruction (auto refresh)	

(f) Number of registerable devices

Forced on/off can be registered for 32 devices in total.

(g) When output Y contact is used in a sequence program

On/off operations in a sequence program are given priority.

(h) Checking forced on/off registration status

Forced on/off execution status can be checked by:

- reading the forced on/off registration status by GX Developer,
- flashing of the MODE LED (green), (The MODE LED flashes in green when at least one forced on/off is registered.) or
- the on status of the 1st bit in SD840 (Debug function usage).

Point

- The MODE LED also flashes in green when the executorial conditioned device test function is used. To check the registration status using the MODE LED, check the status of the executorial conditioned device test function as well. (☞ Section 6.11.4)
- When using SD840 to check the registration or cancellation status, remind that SD840 is used to check the status of the executorial conditioned device test function as well.

(i) Forcibly turning input or output on/off from multiple GX Developers

Forced on/off can be registered to a single CPU module from multiple GX Developers connected via network. In this case, the last registration will be effective.

For this reason, the forced on/off status which is different from the status actually registered in the CPU module may be displayed on the screen of GX Developer that registered forced on/off earlier.

When the forced on/off registration is performed from multiple GX Developers, click the "Update status" button to update the registered data and execute the function.

(3) Operating procedure

Operating procedure is described below.

- To register forced on/off for a device, select [Online] → [Debug] → [Forced input output registration/cancellation] in GX Developer.
- On the screen opened, specify a device and click the "Set forced ON" or "Set forced OFF" button.

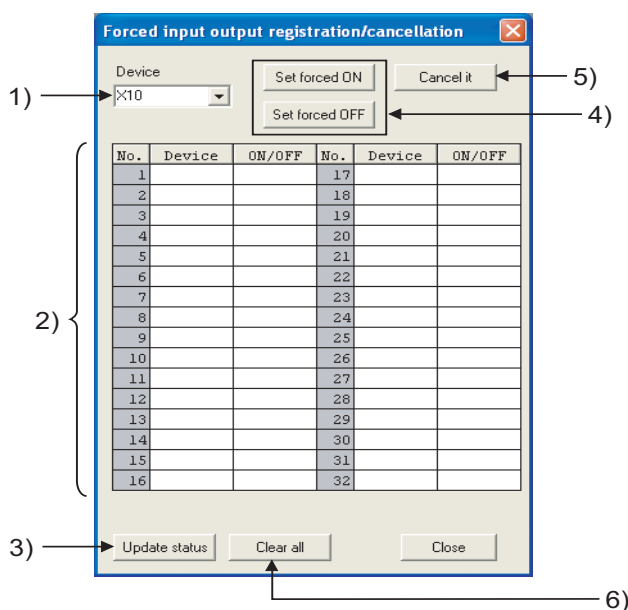


Figure 6.38 Forced input output registration/cancellation screen

Table 6.14 Items on the Forced input output registration/cancellation screen

No.	Item	Description
1)	Device	Select the I/O number for which forced on/off is to be registered or cancelled.
2)	Registration status display area	Displays the forced on/off registration status.
3)	Update status	Reads the forced on/off registration status from the CPU module.
4)	Set forced ON/OFF	Registers forced on/off for a device specified.
5)	Cancel it	Cancels forced on/off registered for the device specified.
6)	Clear all	Cancels all forced on/off registration.

6.11.4 Executorial conditioned device test Note6.4

This function changes a device value within the specified step of a program.

This enables debugging of the specified ladder block without modifying the program.*1

*1: The executorial conditioned device test is not available for the SFC program.

(1) Operation of the executorial conditioned device test

A device value will be changed based on the registration data once after the executorial conditioned device test setting is registered.

The changed device value becomes effective in the ladder blocks of the specified step number and later.

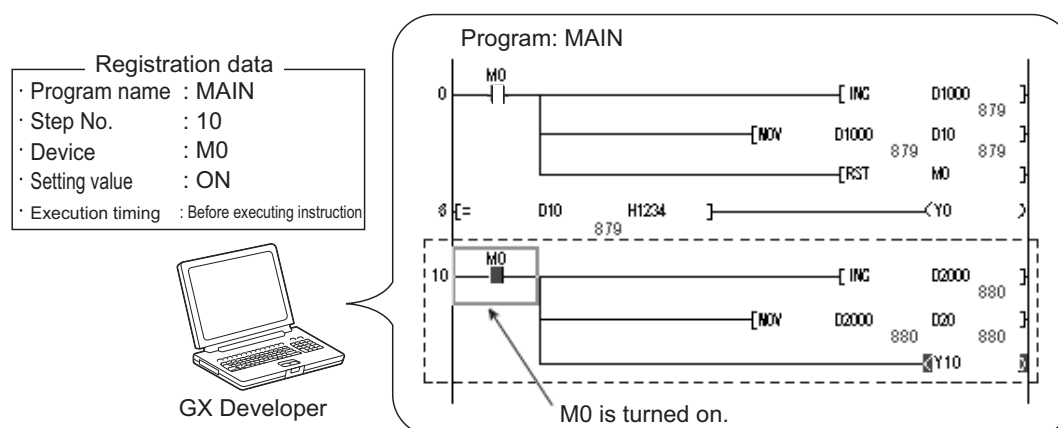
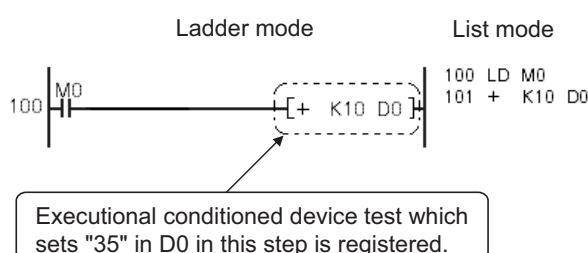


Figure 6.39 Operation of the executorial conditioned device test

Note that a device value is changed within the specified step regardless of an execution status of the instruction in the specified step.

<Program example>



<Operation>

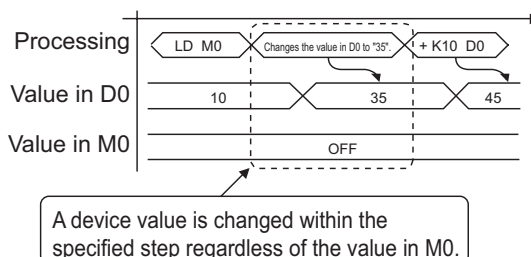


Figure 6.40 Operation example of the executorial conditioned device test

 Note6.4

Universal

When using the executorial conditioned device test in the Q02UDCPU, Q03UDCPU, Q04UDHCPU, Q06UDHCPU, Q13UDHCPU, or Q26UDHCPU, check the versions of the CPU module and GX Developer. ( Appendix 2)

(2) Available devices and number of settable devices

Table6.15 Available devices and number of settable devices

Type	Available device	Number of settable devices
Bit device	X, Y, M, L, B, F, SB, V, SM, T (contact), ST (contact), C (contact), J ₀ \X, J ₀ \Y, J ₀ \B, J ₀ \SB, FX, FY, DX, and DY	Up to 32 (in total)
Word device	T (current value), ST (current value), C (current value), D* ¹ , SD, W* ² , SW, R, ZR, Z, U ₀ \G ₀ , U3E ₀ \G ₀ , J ₀ \W ₀ , J ₀ \SW ₀ , and FD	
	Digit-specified bit device: X, Y, M, L, F, SM, V, B, SB, J ₀ \X, J ₀ \Y, J ₀ \B, and J ₀ \SB Indirect specification (@D0): D, SD, W, SW, R, and ZR (devices specified with @)	

*1: The extended data register (D) is included.

*2: The extended link register (W) is included.

(3) How to check the execution status

The execution status of registered executional conditioned device test can be checked in three different ways:

- By the flash of the MODE LED in green
- By the on status of the first bit of SD840 (Debug function usage)
- By the display on the screen for checking the registration status in GX Developer

( (4)(d) in this section)

Point

- The MODE LED also flashes in green when the external input/output forced on/off function is used. To check the execution status using the MODE LED, check the status of the external input/output forced on/off function as well. ( Section 6.11.3)
- When using SD840 to check the execution status, remind that SD840 is used to check the status of the external input/output forced on/off function as well.

(4) Operating instructions

(a) Registering executional conditioned device test settings

Select the registration target step number on the program editing screen in GX Developer. Then, select [Online] → [Debug] → [Executional conditioned device test] → [Register executional conditioned device test].

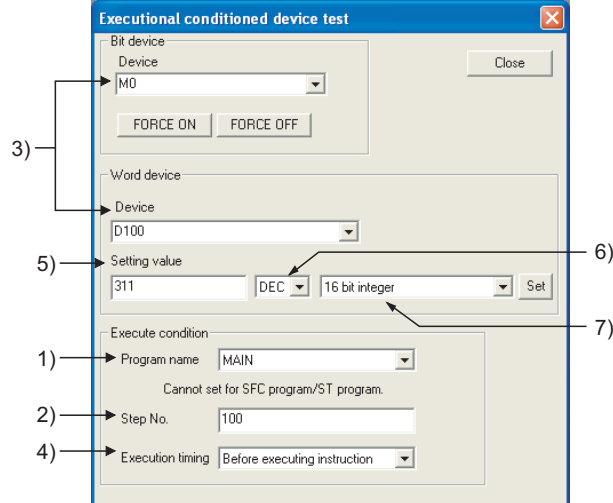


Figure 6.41 Screen for registering executional conditioned device test settings

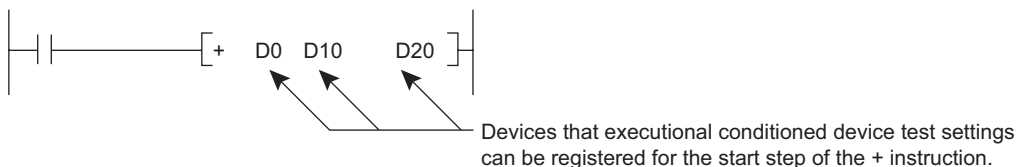
Table 6.16 Items on the screen for registering executional conditioned device test settings

No.	Item	Setting range	
		For bit device	For word device
1)	Program name	Name of a program registered in the CPU module	
2)	Step No.	0 to (Step number for the END instruction)	
3)	Device	Device listed in (2) in this section	
4)	Execution timing	Before executing instruction / After executing instruction	
5)	Setting value	- (Not specified)	Setting range corresponding to each data type
6)	DEC / HEX	- (Not specified)	DEC (decimal) / HEX (hexadecimal)
7)	Data type ^{*1}	- (Not specified)	16 bit integer / 32 bit integer / Real number (single precision) / Real number (Double precision)

*1: When a word device which is index-modified or indirectly-specified is set, only "16 bit integer" can be specified.

1) Multiple executional conditioned device test registrations for the same step number

Multiple executional conditioned device test settings can be registered for one step number.



However, if multiple executional conditioned device test settings with same device name and same execution timing are registered for the same step number, the registration data will be overwritten. (Even though the same device is specified, if the execution timing differs, two settings can be registered for one step.)

Point

- When setting a word device with a different data type, a device is regarded as the same device.

Example When a word device is set in the order of "D100 (16 bit integer)" and then "D100 (Real number (single precision))", "D100 (Real number (single precision))" is registered.

- When setting a device with a different modification method (such as a bit-specified word device, digit-specified bit device, or index-modified device), a device is regarded as a different device.

Example When a word device is set in the order of "D100.F" and then "D100Z0 (Real number (single precision))", both devices are registered.

2) Step to be specified for executional conditioned device test registration

Any step number (0 to step number for the END instruction) in a sequence program can be specified.

Point

Be sure to specify the start step of each instruction.

6

3) Execution timing

Timing of changing a device value can be specified. A device value can be changed either before or after an instruction of the specified step is executed. Figure 6.42 shows the module operation based on the execution timing.

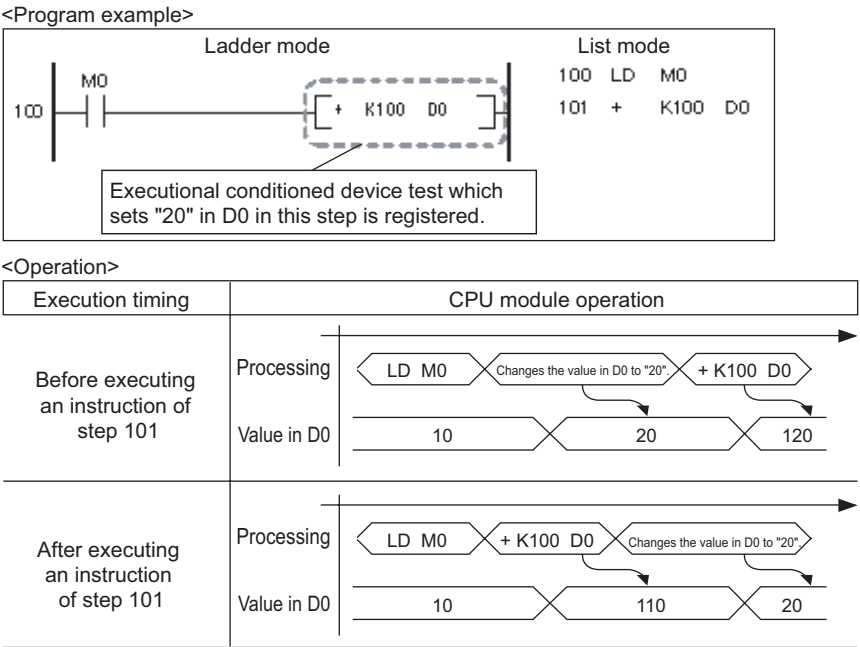


Figure 6.42 CPU module operation based on execution timing

6.11 Monitor Function
6.11.4 Executional conditioned device test

Note that there may be a case where a device value will not be changed depending on the execution timing even though the specified step is executed.

The following instructions need to be noted when registering executional conditioned device test settings.

- Instructions that do not change device values^{*1}

A device value will not be changed by executing the executional conditioned device test when the execution timing has been set to "After executing instruction", specifying the step for instructions that do not execute the next step, such as branch instructions.

*1: If the execution condition of an instruction is not satisfied, a device value will be changed based on the registration data.

Table6.17 lists the instructions that do not change device values.

Table6.17 Instructions that do not change device values

No.	Classification	Instruction	Operation
1	Stop	STOP	• When the execution condition for an instruction is satisfied A device value will not be changed even when the specified step is executed.
2	Jump	CJ	
3		SCJ	
4		GOEND	
5	Repeat (Loop)	BREAK(P)	• When execution condition for an instruction is not satisfied A device value will be changed after the specified step is executed.
6	Subroutine program call	CALL(P)	
7		FCALL(P)	
8		ECALL(P)	
9		EFCALL(P)	
10		XCALL	
11	End	FEND	A device value will not be changed even when the specified step is executed.
12	Jump	JMP	
13	Return from subroutine program	RET	
14	Return from interrupt program	IRET	

- FOR/NEXT instructions

When the executional conditioned device test setting is registered specifying the step for the FOR or NEXT instruction, timing of device value change is different from the timing when steps for other instructions are specified.

Table6.18 shows the device value change timing based on the execution timing.

Table6.18 Device value change timing when a step for the FOR or NEXT instruction is specified

Instruction of the specified step	Execution timing setting	
	Before executing instruction	After executing instruction
FOR	Executed once before the start of loop processing.	Executed once after the start of loop processing. (Executed before the operation of the program between the FOR and NEXT instructions.)
NEXT	Executed every loop processing. (Executed after the operation of the program between the FOR and NEXT instructions.)	Executed once after the start of loop processing.

- END instruction

When the executional conditioned device test setting is registered specifying the step for the END instruction, the execution timing is restricted to "Before executing instruction" only. If "After executing instruction" is set, the CPU module returns a registration error to GX Developer.

4) Number of settings that can be registered simultaneously in one scan

Eight executional conditioned device test settings can be registered into the CPU module simultaneously in one scan.

When nine or more executional conditioned device test settings are to be registered simultaneously by GX Developer, they will be registered over multiple scans.

(b) Disabling executional conditioned device test settings

Select the disabling target step number on the program editing screen in GX Developer. Then, select [Online] → [Debug] → [Executional conditioned device test] → [Check/disable executional conditioned device test].

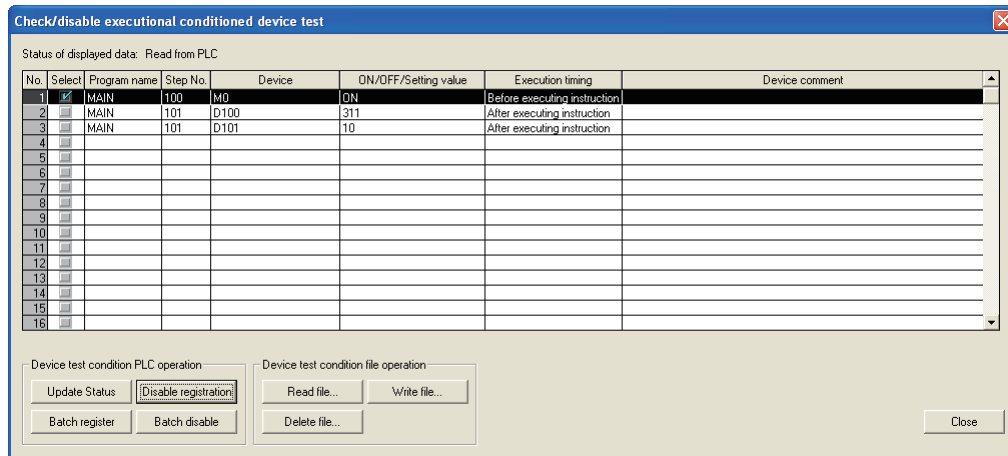


Figure 6.43 Screen for checking/disabling executional conditioned device test settings

Table 6.19 Items on the screen for checking/disabling executional conditioned device test settings

Item	Setting range	
	For bit device	For word device
Program name	Name of a program registered in the CPU module	
Step No.	0 to (Step No. for the END instruction)	
Device	Device listed in (2) in this section	
Execution timing	Before executing instruction/After executing instruction	

1) Number of settings that can be disabled simultaneously in one scan

Eight executional conditioned device test settings can be disabled simultaneously in one scan.

When nine or more executional conditioned device test settings are to be disabled simultaneously by GX Developer, they will be disabled over multiple scans.

(c) Batch-disabling executional conditioned device test settings

Select [Online] → [Debug] → [Executional conditioned device test] → [Batch disable executional conditioned device test] in GX Developer.

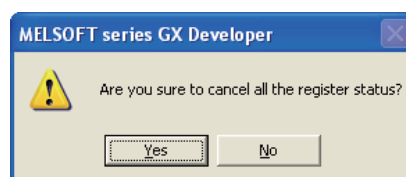


Figure 6.44 Screen for batch-disabling executional conditioned device test settings

(d) Checking executional conditioned device test settings

Select [Online] → [Debug] → [Executional conditioned device test] → [Check/disable executional conditioned device test].

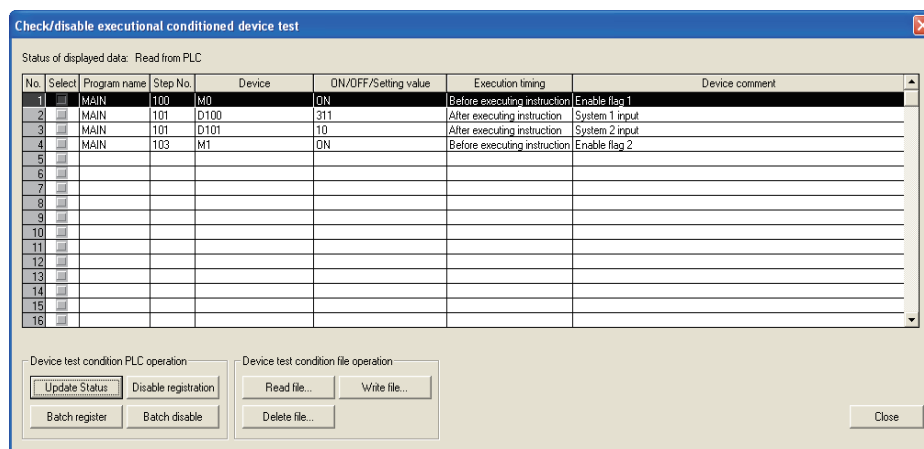


Figure 6.45 Screen for checking executional conditioned device test settings

Remark

1. For the operating instruction of checking or disabling executional conditioned device test settings, refer to the following.
GX Developer Version 8 Operating Manual
2. Usage of the executional conditioned device test can be checked in the special register (SD840).

(e) Cases where executional conditioned device test settings cannot be registered or disabled

In the following cases, executional conditioned device test setting cannot be registered or disabled.
When multiple settings are to be registered, no setting can be registered if any of the settings applies to the cases below.

1) Executional conditioned device test settings cannot be registered

- Specified program does not exist.
- Specified step does not exist.
- Specified device does not exist.
- The number of registered executional conditioned device test settings exceeds 32.

2) Executional conditioned device test settings cannot be disabled

- Specified program does not exist.
- Specified step does not exist.
- Specified device does not exist.
- No executional conditioned device test setting has been registered.

(5) Precautions

(a) Operations from multiple GX Developers

Executorial conditioned device test setting can be registered in the same CPU module from multiple GX Developers connected via network.

Note, however, that if multiple executorial conditioned device test settings are registered with the same device name in the same step, the registration data will be overwritten.

When registering executorial conditioned device data setting from multiple GX Developers update the data first by clicking the **Read file** button, and then register the setting.

(b) Priority

If any of the following functions is set in the same step number that is specified by the executorial conditioned device test setting, the executorial conditioned device test is given the priority to other functions.

- Monitor condition setting
- Sampling trace function (trace point)
- Sampling trace function (trigger point)
- Scan time measurement (start step)
- Scan time measurement (end step)

(c) Disabling executorial conditioned device test settings

Executorial conditioned device test setting can be disabled by any of the following operations, in addition to the operation by GX Developer.

- Powering off and then on the CPU module
- Resetting the CPU module
- Writing program files to the program memory while the CPU module is in the STOP status
- Clearing the program memory while the CPU module is in the STOP status
- Formatting the program memory while the CPU module is in the STOP status

(d) Writing data to the running CPU module

The CPU module operation will be as follows if the executorial conditioned device test and the online change function are executed simultaneously.

1) When the executorial conditioned device test is executed during execution of the online change function

The online change function completes normally. However, the executorial conditioned device test cannot be executed.

The following message box will appear. Execute the executorial conditioned device test again after the online change has completed.

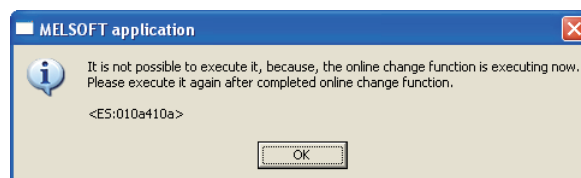


Figure 6.46 Message box (when the executorial conditioned device test is executed during execution of the online change function)

2) When the online change function is executed during execution of the executorial conditioned device test

The Online change function completes normally. If any executorial conditioned device test setting has been registered in the program to be changed online, the corresponding setting will be disabled.

( (5)(e) in this section)

(e) Online change of the CPU module with executorial conditioned device test registration

1) Online change (ladder mode)

If any executorial conditioned device test setting has been registered in the ladder block to be changed online, the CPU module disables the corresponding setting.

Example 1) Step numbers of registrations 1 to 3 are specified in the executorial conditioned device test settings. When the ladder block including the registration 2 is changed online, the registration 2 is disabled during execution of the online change function. Since the registrations 1 and 3 are not included in the change target program, they are not disabled.

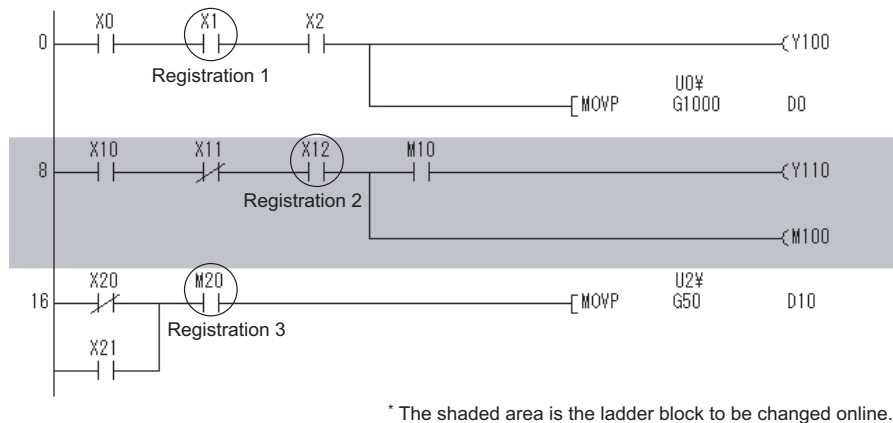


Figure 6.47 Adding a ladder block online

Example 2) When multiple ladder blocks are to be changed online, ladder blocks between the change target ladder blocks will be included in the change target. For this reason, if the online change function is executed as shown in Figure 6.48, all registrations 1 to 3 are disabled.

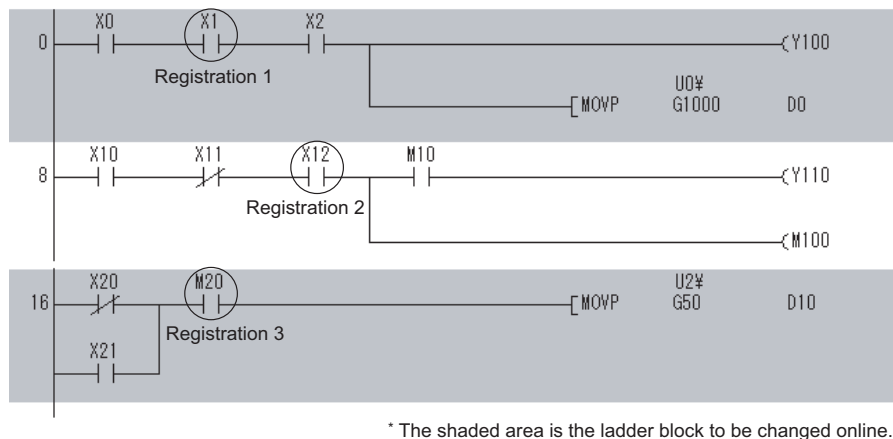


Figure 6.48 Changing multiple ladder blocks online

Example 3) When a ladder block is to be added online, the executional conditioned device test setting included in the ladder block followed after the added ladder block will be disabled. For this reason, if the online change function is executed as shown in Figure 6.49, the registration 2 is disabled.

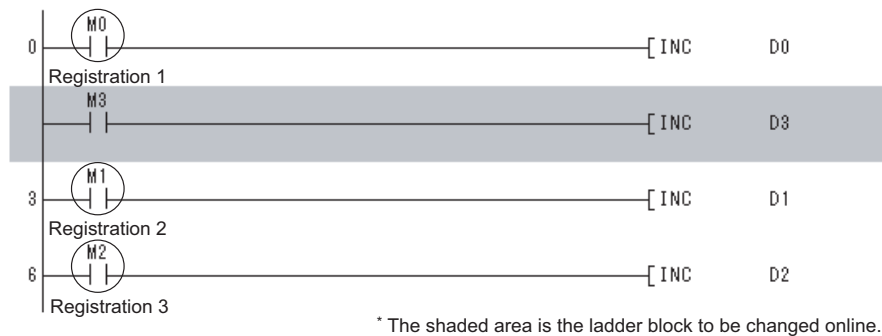


Figure 6.49 Adding a ladder block online

2) Online change (files)

All executional conditioned device test settings registered to the program in the online change target file are disabled.

(f) Precautions for specifying an index-modified device

If an index-modified device name is specified to register the executional conditioned device test setting, the CPU module does not check whether the specified device is within the setting range.

If the index-modified device is out of the device range or on the boundary of devices, a device value will not be changed within the specified step.

(g) Precautions for specifying an indirectly-specified device

If indirectly-specified device name is specified to register the executional conditioned device test setting, the CPU module does not check whether the specified device is within the setting range.

If the indirectly-specified device is out of the device range or on the boundary of devices, a device value will not be changed within the specified step.

(h) Precautions for specifying the file register

If the file register is specified to register the executional conditioned device test setting, the CPU module does not check the file register file assignment and the file register number range.

A file register value will not be changed within the specified step in the following cases.

- The file register file is not assigned.
- The specified file register number is out of the file register range.

6.12 Writing Programs While CPU Module is in RUN Status

There are two ways of writing programs in the RUN status.

- Online change (ladder mode) :  Section 6.12.1
- Online change (files) :  Section 6.12.2

Data can also be written in the RUN status using a pointer. ( Section 6.15.2)

6.12.1 Online change (ladder mode)

(1) Definition

This function writes programs to the CPU module in the RUN status.

This function enables the program to be changed without stopping the program operation in the CPU module.

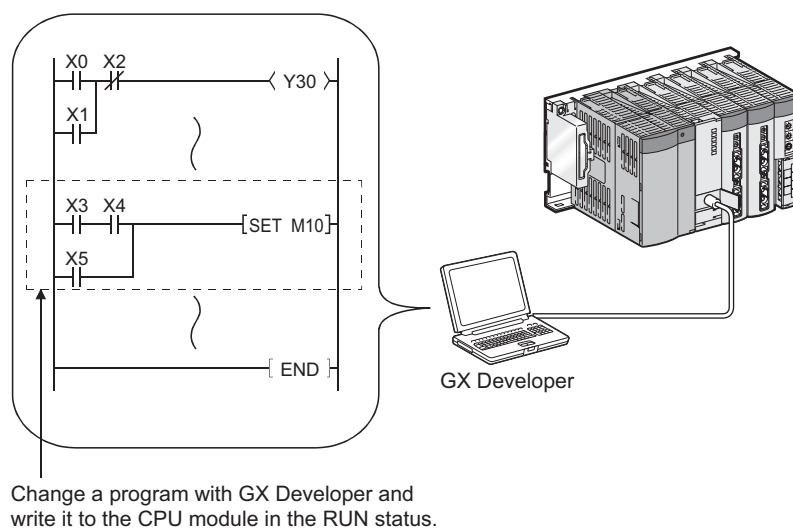


Figure 6.50 Outline of online change (ladder mode)

This function also can write programs by GX Developer connected to another station on the network.

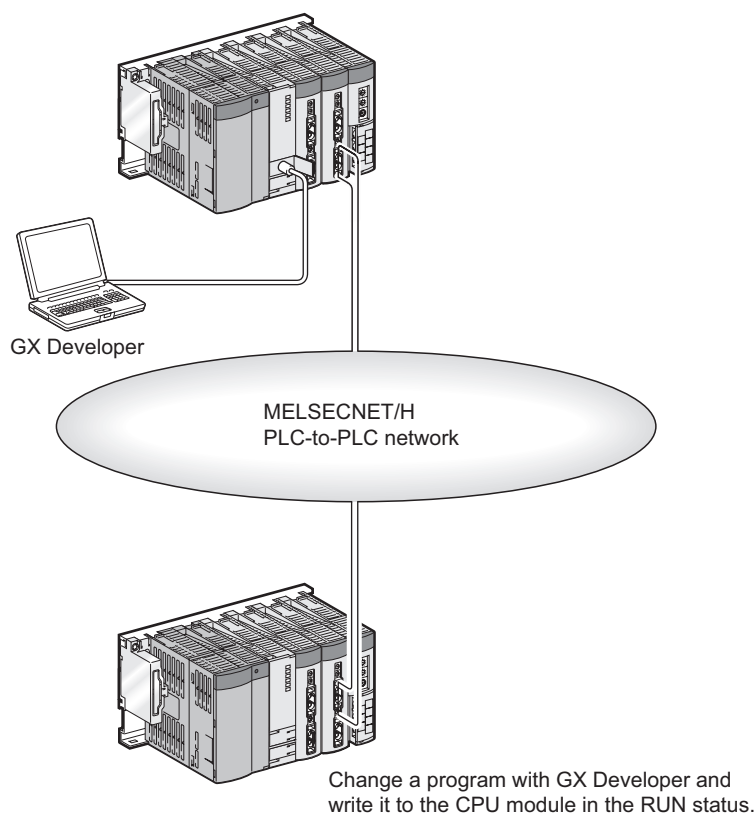


Figure 6.51 Outline of online change via network

(2) Memory for online change

A program cache memory (program memory) is available.

(3) Number of steps that can be batch-written by online change

Up to 512 steps can be batch-written.

(4) Changing the reserved area for online change

A program file has an area designated as reserved area for online change to support the online change that changes program file size.

The following provides precautions when changing the size of reserved area for online change.

(a) Size of a program file

The size of a program file is addition of created program size and reserved area for online change.

(b) When program file size is increased from the secured capacity

If the size secured for the program file (size including reserved area for online change) is exceeded after a program is written in the RUN status, the reserved area for online change can be re-set before the writing if the user memory area has space.

(c) Increase in the scan time

The scan time is increased when reserved area for online change is re-set when programs are written in the RUN status.

For increase in the scan time, refer to Section 10.1.3.

(5) Operations prohibited when programs are written to the CPU module in the RUN status, T/C setting value is changed, or data are transferred from a program cache memory to a program memory

Refer to Section 6.12.3(2).

(6) Instructions that do not operate normally when programs are written to the CPU module in the RUN status

Refer to Section 6.12.3(3).

6.12.2 Online change (files)

(1) Definition

This function batch-writes files shown in Table 6.20 to the CPU module in the RUN status by online operation from GX Developer.

Table6.20 Files that can be written to the CPU module in the RUN status

File name	CPU module built-in memory			Memory card (RAM)	Memory card (ROM)	
	Program memory	Standard RAM	Standard ROM	SRAM card	Flash card	ATA card
Parameter	×	○	×	×	×	×
Intelligent function module parameter	×	×	×	×	×	×
Program	○	×	○	○	×	×
Device comment	○	×	△ ^{*1}	△ ^{*1}	×	△ ^{*1}
Initial device value	×	×	×	×	×	×
File register	×	△ ^{*1}	×	△ ^{*1}	×	×
Local device	×	×	×	×	×	×
Sampling trace file	×	○	×	○	×	×
Programmable controller user data	×	×	○	×	×	○

○ : Can be written, △ : Partially restricted, × : Cannot be written

*1: The file can be written if not being accessed by a sequence program.

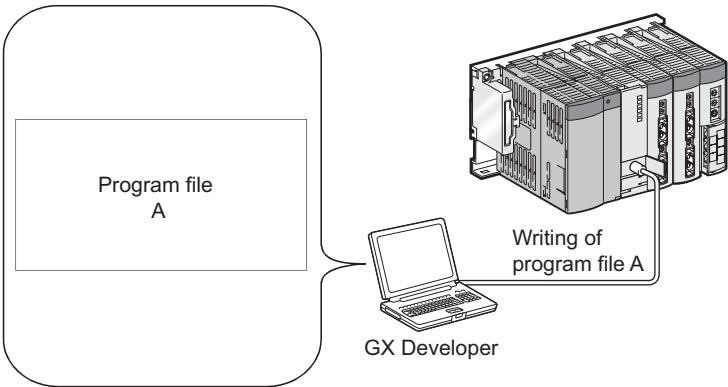


Figure 6.52 Outline of online change (files)

(2) Availability

(a) For the Q00UJCPU, Q00UCPU, and Q01UCPU

The function cannot be performed in the following cases.

- A program memory does not have enough area for storing a program file to be written.
- A program memory stores the maximum number of files can be stored.

(b) For the Q02UCPU, QnUD(H)CPU, and Built-in Ethernet port QCPU

Files can be written in the RUN status, regardless of free space in the program memory and the number of files to be stored.

(3) Increase in the scan time

The scan time increases when a program file is written to the CPU module in the RUN status.

For increase in the scan time, refer to Section 10.1.3.

(4) Online change(files) from multiple locations

Do not simultaneously write files to one CPU module in the RUN status from multiple locations.

Doing so may delete program files.

(5) Online change (files) of SFC programs

SFC programs cannot be written in units of files to the CPU module in the RUN status.

(6) Operations prohibited when programs are written to the CPU module in the RUN status, T/C setting value is changed, or data are transferred from a program cache memory to a program memory

Refer to Section 6.12.3(2).

(7) Instructions that do not operate normally when files are written to the CPU module in the RUN status

Refer to Section 6.12.3(3).

6.12.3 Precautions for online change

The following shows precautions for online change.

(1) Online change during boot operation

When data are written to the CPU module in the RUN status during boot operation, the status of boot source program is not changed.

(2) Operations prohibited when programs are written to the CPU module in the RUN status, T/C setting value is changed, or data are transferred from a program cache memory to a program memory

Do not perform the following operations.

(a) Power-off or reset

The following operations are not normally completed if they are performed during online change, TC setting value change, or data transfer from the program cache memory to the program memory.

If performed, write the data to the CPU module again.

- Power-off
- Reset

(b) Operations from GX Developer

The following operations cannot be performed during online change, TC setting value change, or data transfer from the program cache memory to the program memory. If performed, an error is displayed on GX Developer. Perform the following operations after online change, TC setting value change, or data transfer from the program cache memory to the program memory.

(Ladder mode, writing files during RUN, and function block)

TC setting value change

Data transfer to the program memory

- Online change (ladder mode), online change (files)
- TC setting value change
- Data transfer to the program memory  Note6.5
- Write to PLC (Flash ROM)



Note6.5

Universal

In the universal model QCPUs with the first five digits of the serial number "12012" or later transferable to the program memory during online change.

(3) Instructions do not operate normally during online change

When data are written to the CPU module in the RUN status, the following instructions do not operate normally.

- Rise instruction
- SCJ instruction

(a) Rise instruction

The rise instruction is not executed when the instruction is in the data written to the CPU module in the RUN status, even if the execution condition (off → on) is met.

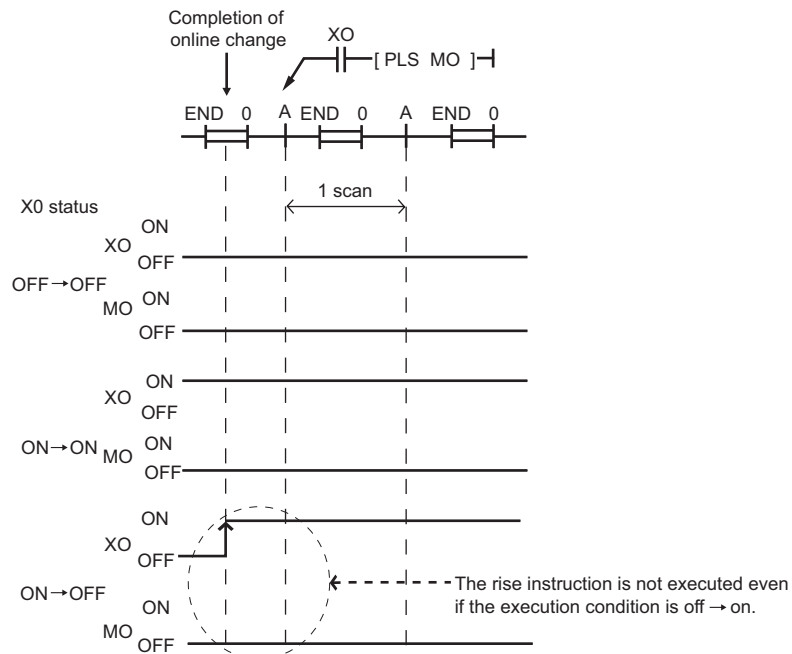


Figure 6.53 Operation of the rise instruction

The corresponding rise instructions are PLS and □P.

(b) SCJ instruction

When the SCJ instruction is in the data written to the CPU module in the RUN status and the execution condition is on at completion of the writing, a jump to the specified pointer is made without a wait of one scan.

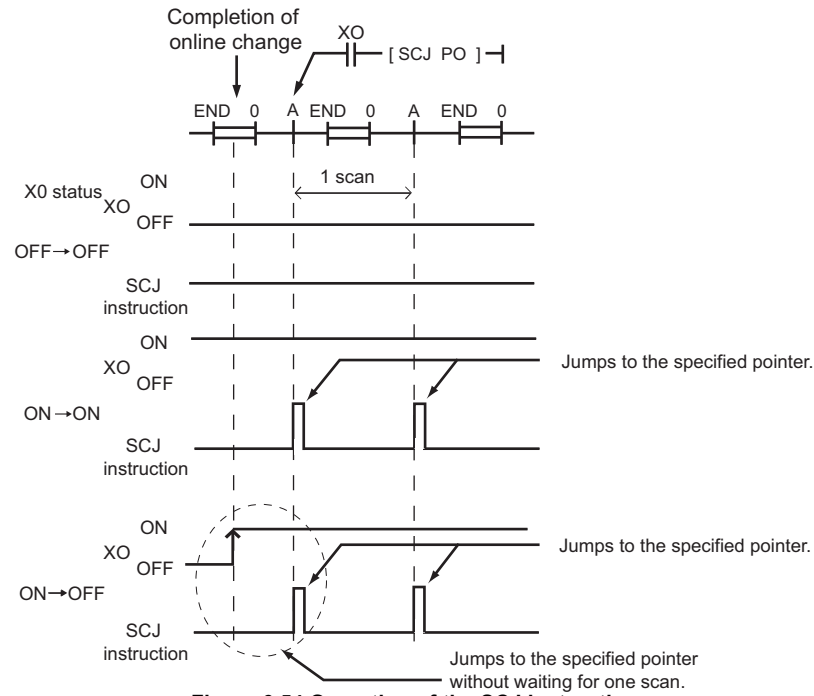


Figure 6.54 Operation of the SCJ instruction

With "Trailing edge instructions are not executed" not selected for the optional setting of GX Developer, the fall instruction is executed when the instruction is in the data written to the CPU module in the RUN status, even if the execution condition (on → off) is not met. (Same operation as the high performance model QCPU)

When this option is not selected, trailing edge instructions whose execution condition are "off" will be executed. (It is selected by default.)

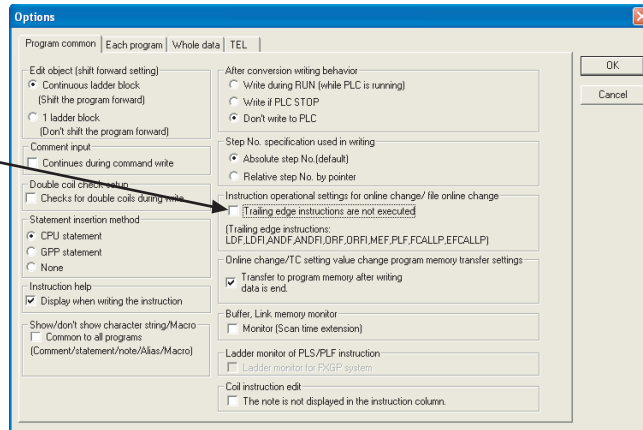


Figure 6.55 Options screen

The corresponding fall instructions are LDF, ANDF, ORF, MEF, PLF, FCALLP, and EFCALLP. The following describes the operation with and without "Trailing edge instructions are not executed" selected.

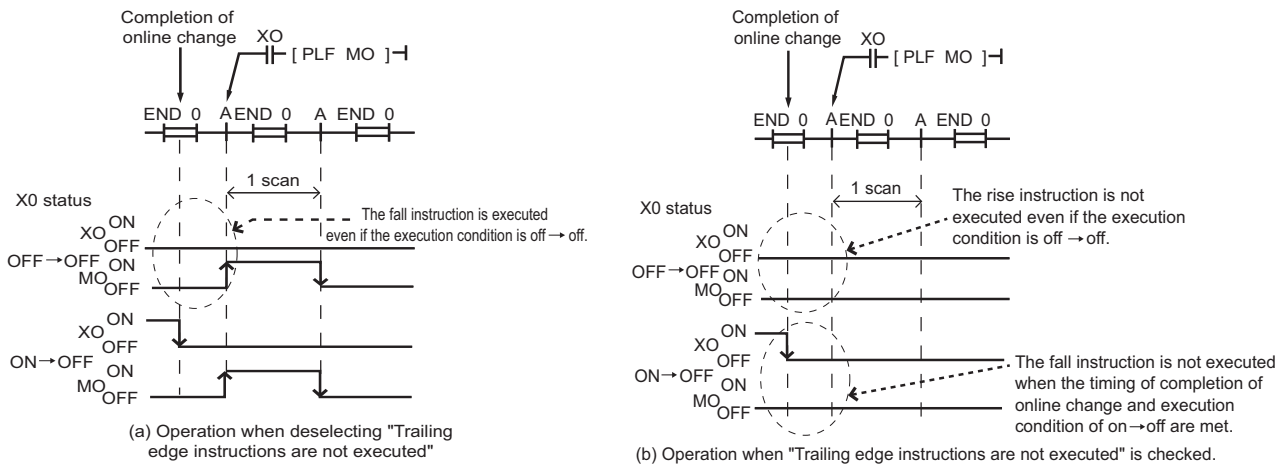


Figure 6.56 Operation comparison of the fall instruction

(4) Writing to the program memory during online change and T/C setting value change

Contents changed due to online change and TC setting value change are automatically transferred to the program memory simultaneously when the data are written to the program cache memory.

The time required for writing data to the CPU module in the RUN status and changing T/C setting value is increased due to automatic transfer of the program memory by the time shown in Table6.21.

Table6.21 Time increased due to automatic transfer of data to the program memory

CPU module	Transfer time
Q00UJCPU,Q00UCPU,Q01UCPU,Q02UCPU	$T_s \times 320 + 4.8 \text{ (s)}$
Q03UDCPU,Q04UDHCPU,Q06UDHCPU,Q03UDECPU, Q04UDEHCPU,Q06UDEHCPU	$T_s \times 260 + 4.7 \text{ (s)}$
Q10UDHCPU,Q10UDEHCPU	$T_s \times 439 + 6.2 \text{ (s)}$
Q13UDHCPU,Q13UDEHCPU	$T_s \times 600 + 8.0 \text{ (s)}$
Q20UDHCPU,Q20UDEHCPU	$T_s \times 839 + 11.4 \text{ (s)}$
Q26UDHCPU,Q26UDEHCPU	$T_s \times 1100 + 15.0 \text{ (s)}$
Q50UDEHCPU	$T_s \times 2450 + 17.0 \text{ (s)}$
Q100UDEHCPU	$T_s \times 4550 + 9.0 \text{ (s)}$

T_s : Scan time (s)

Since the number of writes to the program memory (Flash ROM) is limited (up to 100,000 times), set the automatic transfer to the program memory to be disabled when data are written to the CPU module in the RUN status and changing T/C setting value frequently.

Automatic transfer to the program memory can be set to be disabled in the Options screen of GX Developer.

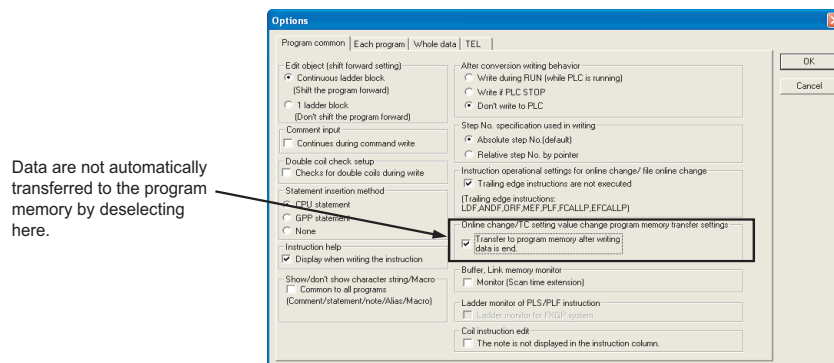
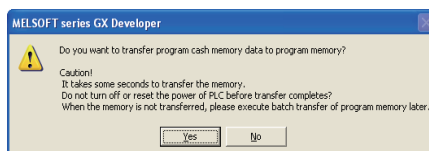


Figure 6.57 Online change/TC setting value change program memory transfer settings

When the automatic transfer is set to be disabled, the following message appears after online change.



Selecting "Yes" transfers data to the program memory.

Selecting "No" does not transfer data to the program memory.

When selecting "No", execute "Program memory batch transfer" by GX Developer.

Program transfer status can be checked by the special relay (SM165). Note6.6

When SM165 is on, the program memory batch transfer has not completed.

When SM165 is off, the program memory batch transfer has completed.

Note6.6

Universal

When checking the batch transfer status to the program memory with the Q02UCPU, Q03UDCPU, Q04UDHCPU, or Q06UDHCPU, check the versions of the CPU module and GX Developer. (Appendix 2)

6.13 Execution Time Measurement

(1) Definition

This function displays the processing time of the program being executed.

(2) Applications and types

This function can be used to know the effect of processing time of each program on the total scan time when the system is adjusted.

There are following three types.

- Program monitor list :  Section 6.13.1
- Interrupt program list monitor :  Section 6.13.2
- Scan time measurement :  Section 6.13.3

6.13.1 Program monitor list

(1) Definition

This function displays the processing time of the program being executed.

The scan time, number of execution times, and processing time by item can be displayed for each program.

(2) Execution

Selecting [Online] → [Monitor] → [Program monitor list] displays the Program monitor list screen.*1

Figure 6.58 shows an example of executing the program monitor list.

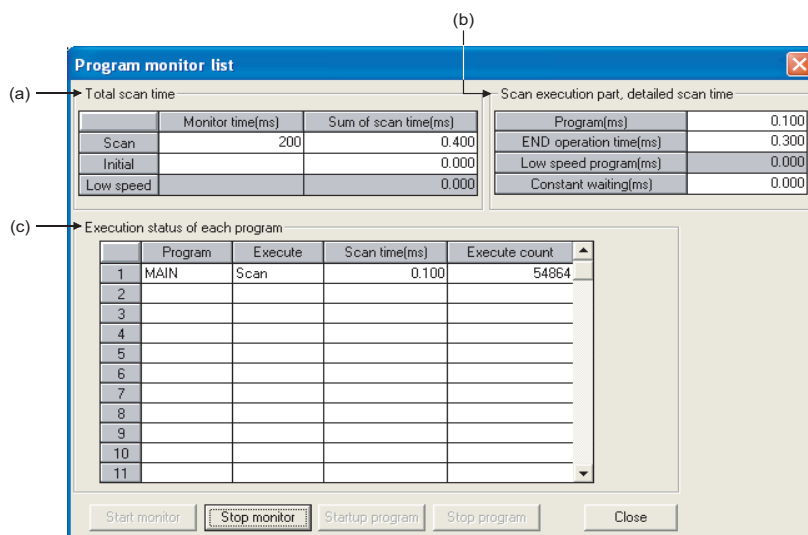


Figure 6.58 Program monitor list screen

*1: During execution of a fixed scan execution type program, the scan time of the fixed scan execution type program is not displayed.

"-" is displayed in the Scan time column.

(a) Total Scan Time

The monitoring time set in "WDT (Watchdog timer) setting" of the PLC RAS tab of the PLC parameter dialog box and total scan time for each program type during execution by the CPU module are displayed.

1) Monitor time

The monitoring time of each program is displayed.

If the scan time exceeds this time, the CPU module detects "WDT ERROR".

2) Sum of scan time

The total time of each item in "Scan execution part, detailed scan time" is displayed.

When constant scan time is set, the constant scan time is displayed.

(b) Scan execution part, detailed scan time

The details of the scan time are displayed.

1) Program

The total execution time of the scan execution type program is displayed.

2) END operation time

The END processing time is displayed.

3) Low speed program

Since a low-speed execution type program is not used "0.000" is displayed.

4) Constant waiting

The constant scan waiting time is displayed when the constant scan time is set.

(c) Execution status of each program

The execution status of a program selected at the program tab of the PLC parameter dialog box is displayed.

1) Program

The program name is displayed in the order set in the PLC parameter dialog box.

2) Execution

The program type set in the PLC parameter dialog box is displayed.

3) Scan time

The actual scan time (current value) is displayed.

When a program is in stop (standby) status, the scan time is displayed as 0.000 ms.

4) Execute count

The number of execution times of programs before monitoring is displayed, setting the measurement start as "0". The number of execution times is displayed up to 65535 and returns to 0 when the 65536 is measured.

The execution times is held even when the program is stopped.

Remark

When the POFF instruction is executed, a non-execution processing is performed for one scan.
The number of execution times displayed is the addition of the execution times of the non-execution processing.

For details of the POFF instruction, refer to the following.

 MELSEC-Q/L Programming Manual (Common Instruction)

(3) Program start and stop

Program cannot be started and stopped from the Program list monitor screen.

(4) Precautions

The scan time of a fixed scan execution type program is not displayed during its execution.

"-" is displayed in the Scan time column.

6.13.2 Interrupt program monitor list

(1) Definition

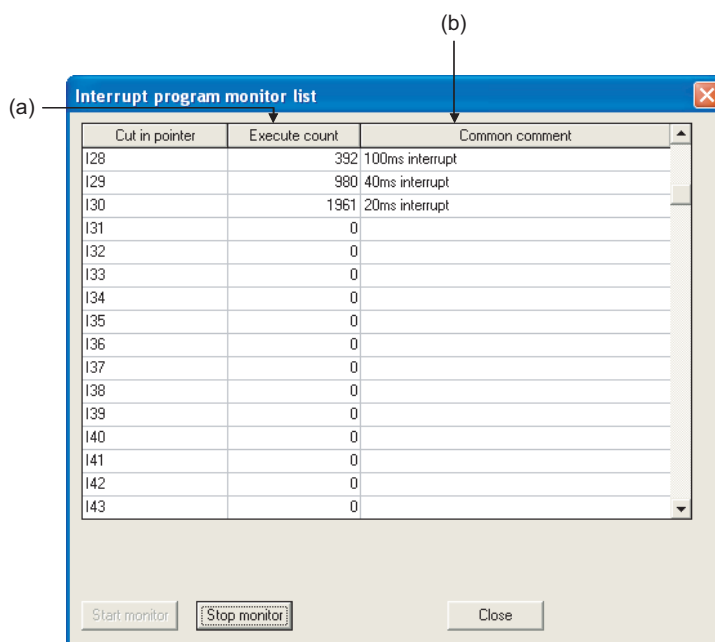
This function displays the number of executions of an interrupt program.

This function is used to check the execution status of the interrupt program.

(2) Execution

Selecting [Online] → [Monitor] → [Interrupt program monitor list] of GX Developer displays the Interrupt program monitor list screen.

Figure Figure 6.59 shows an execution example of the interrupt program monitor list.



Cut in pointer	Execute count	Common comment
I28	392	100ms interrupt
I29	980	40ms interrupt
I30	1961	20ms interrupt
I31	0	
I32	0	
I33	0	
I34	0	
I35	0	
I36	0	
I37	0	
I38	0	
I39	0	
I40	0	
I41	0	
I42	0	
I43	0	

Figure 6.59 Interrupt program monitor list screen

(a) Execute count

The number of executions of an interrupt program is displayed.

This function starts counting after the CPU module is in the RUN status.

When the counting reaches 65536 times, it is reset to 0.

(b) Common comment

Device comments created to an interrupt pointer is displayed.

6.13.3 Scan time measurement Note6.7

(1) Definition

This function displays the processing time of set program section during ladder monitoring.
The time required for the subroutine and interrupt programs can be measured.

(2) Range specification of scan time measurement

There are following two types for specifying a scan time measurement range.

- Setting on the ladder monitor screen
- Setting on the scan time measurement screen

(3) When the subroutine program call instruction is in the measurement range

When the subroutine program call instruction (CALL) is in the range of scan time measurement, the scan time includes the time required for processing a subroutine program.

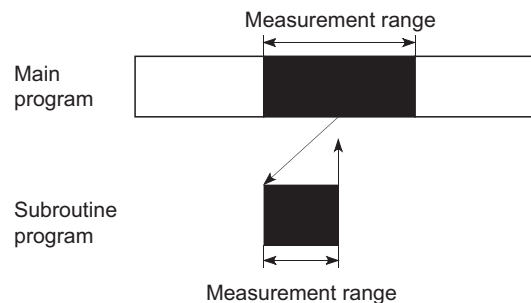


Figure 6.60 When subroutine program is in measurement range

(4) When interrupt programs/fixed scan execution type programs are executed in the scan time measurement range

The execution time of interrupt programs and fixed scan execution type programs are added.



Note6.7

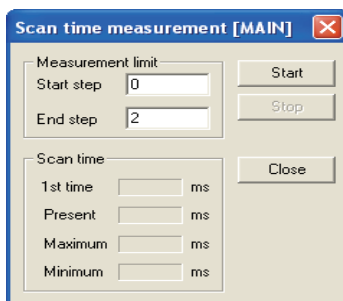
Universal

When using the scan time measurement for the Q02UCPU, Q03UDCPU, Q04UDHCPU, Q06UDHCPU, Q13UDHCPU, and Q26UDHCPU, check the versions of CPU module and GX Developer. ( Appendix 2)

(5) Execution

Measure the scan time by the following procedure.

- Display the start of the ladder program where scan time is measured in GX Developer and set the monitor mode.
- Select [Online] → [Monitor] → [Scan time measurement] to open the Scan time measurement screen.



- Enter the start and end steps and click the **Start** button.

Example When the start step is 52 and the end step is 105

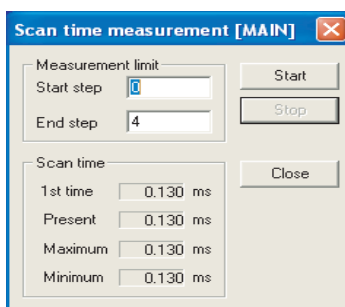


Figure 6.61 Scan time measurement screen

Remark

When displaying the Scan time measurement screen after specifying the scan time measurement range in the monitor mode, the start and end steps are set in the specified range.
To specify the range, press the "Shift" key and click the mouse (The specified part is inverted).

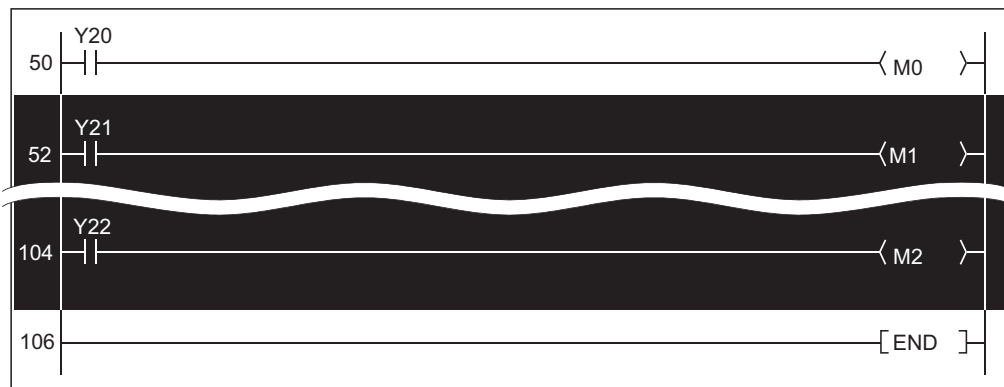


Figure 6.62 Measurement range specification

(6) Precautions

(a) Measurement range setting

Set the measurement range so that "Start step < End step" is satisfied.

(b) Minimum unit of measurement time

The minimum unit of measurement time is 0.01ms.

If the measurement time is less than 0.01ms, 0.000ms is displayed.

(c) When between the FOR and NEXT instructions is specified

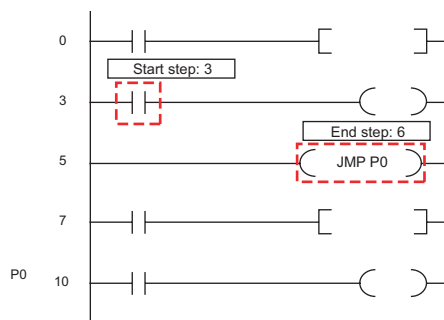
The execution time of one scan between the specified steps is displayed.

(d) When scan time cannot be measured

Scan time cannot be updated on the Scan time measurement screen in the following cases.

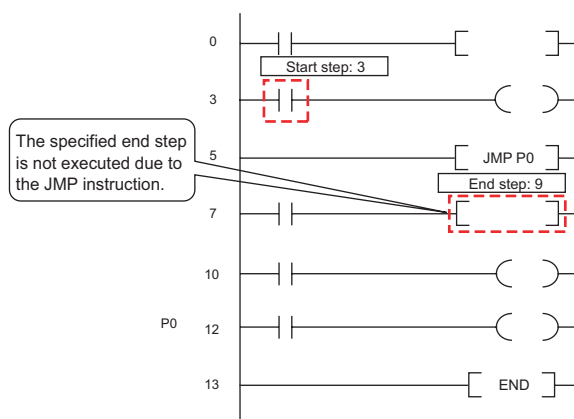
- When the branch instruction is specified the to end step

Example The JMP instruction is specified to the end step.



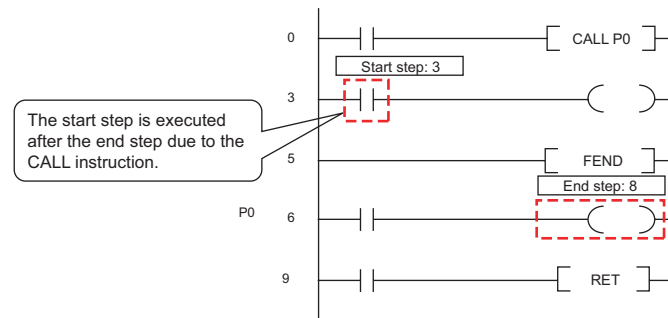
- When only the start step is executed

Example The specified end step is not executed by the JMP instruction.



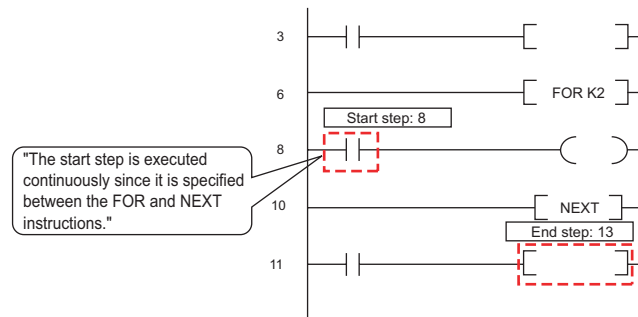
- When the end step is executed before the start step

Example The start step is specified as the next step of the CALL instruction and the end step is specified in a subroutine program executed by the CALL instruction.



- When the start step is executed continuously

Example Only the start step is specified between the FOR and NEXT instructions.



6.14 Sampling Trace Function Note6.8

(1) Definition

This function samples the data of the specified device at a preset timing and at a preset interval (sampling cycle), and then stores the trace results in the sampling trace file.

(2) Application

This function is useful to check the change of the device data used in the program during debugging at a preset timing.

In addition, this function is used to read the device data at trigger condition establishment.

(3) Sampling trace file

This file stores the trace setting necessary for executing the function and trace results.

Point

Sampling trace file can be stored only in the Standard RAM or SRAM card. ( Section 5.1.1)

(4) Sampling trace operation

(a) Operation of the CPU module

When a sampling trace trigger is issued by GX Developer, the CPU module executes traces for the preset number of times.

The number of traces will be a value of which the number of bytes for the sampling trace area divided by the number of bytes of the specified device ($N1 + N2 + N3 + \text{word device points} \times 2 + (\text{bit device points}/16) \times 2$).^{*1 *2}

*1: Round up the result of "bit device points/16" in the expression to the right of the decimal point.

*2: Add the following values to N1 to N3 according to the items selected under the trace additional information of the trace condition setting.

•N1: When "Time(sec)" is selected, add "4".

•N2: When "Step no." is selected, add "10".

•N3: When "Program name" is selected, add "8".

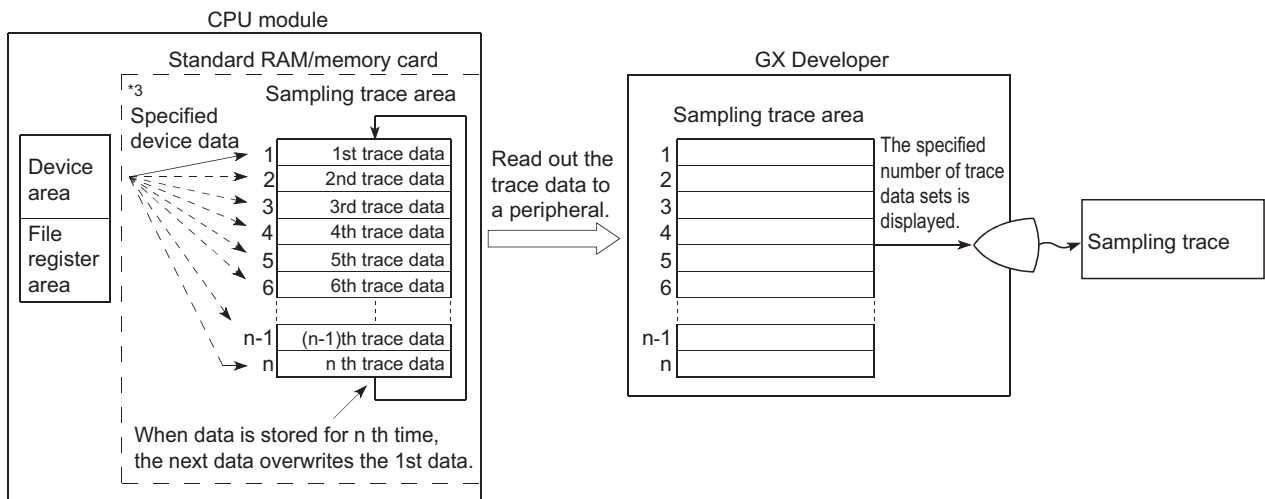


Figure 6.63 Sampling trace operation

*3: When the trigger is issued, the CPU module samples data for the preset number of times and latches the data in the sampling trace area.

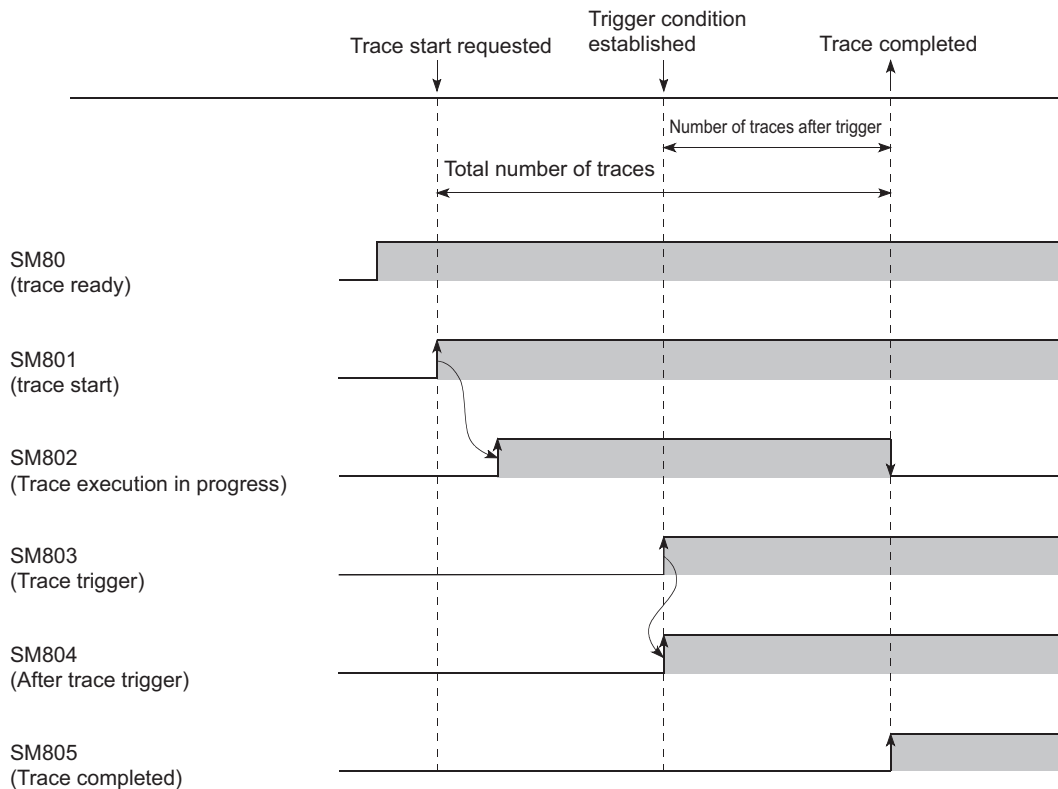
(b) Operation of the special relay**1) When the sampling trace is executed normally**

The execution status of the sampling trace can be checked in the special relay listed in Table6.22.

Table6.22 Execution status of the sampling trace

Number	Name	Description
SM800	Trace preparation	Turns on when the trace setting in GX Developer is written to the CPU module. The relay is used to check the sampling trace enable status.
SM801	Trace start	Turns on when the sampling trace is started.
SM802	Trace execution in progress	Turns on during sampling trace execution. The relay is used to check the sampling trace execution status.
SM803	Trace trigger	A trigger turns on upon the status change of the relay (off → on).
SM804	After trace trigger	Turns on when any of the following condition is established. • A trigger is issued by GX Developer. • The TRACE instruction is executed. • SM803 turns on. • Detailed setting (Device and Step No.) The relay is used to check the trigger condition establishment status.
SM805	Trace completed	Turns on when the sampling trace is completed.
SM826	Trace error	Turns on when an error occurs during sampling trace execution.

Figure 6.64 shows the operation flow chart of the special relay for sampling trace execution.

**Figure 6.64 Operation flow chart of the special relay (for sampling trace execution)**

2) When the sampling trace is interrupted

If SM801 (Trace start) is turned off during sampling trace, execution of the sampling trace will be interrupted. When the sampling trace is interrupted, the trace count is cleared.

The sampling trace restarts by turning on SM801.

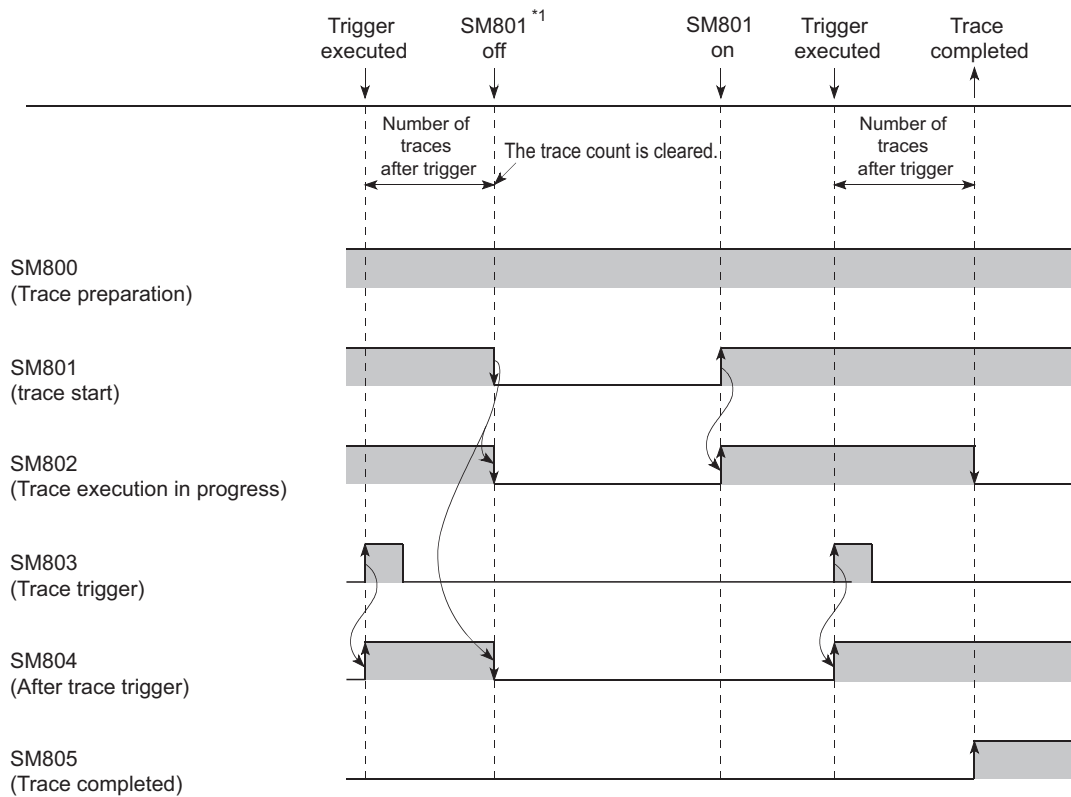


Figure 6.65 Operation flow chart of the special relay (for sampling trace interruption)

*1: SM801 also turns off when the sampling trace is interrupted by GX Developer.

(5) Operating procedure

Select [Online] → [Trace] → [Sampling trace...] in GX Developer.

On the screen opened, select the method for operating the sampling trace.

- "Wizard setting/execution"
( GX Developer Version 8 Operating Manual)
- "Individual setting/execution"
( (5)(a) in this section)

(a) Setting "Trace data (setting + result) storage" and "Trace execution method"

On the screen opened, set the trace data storage location and trace execution method.

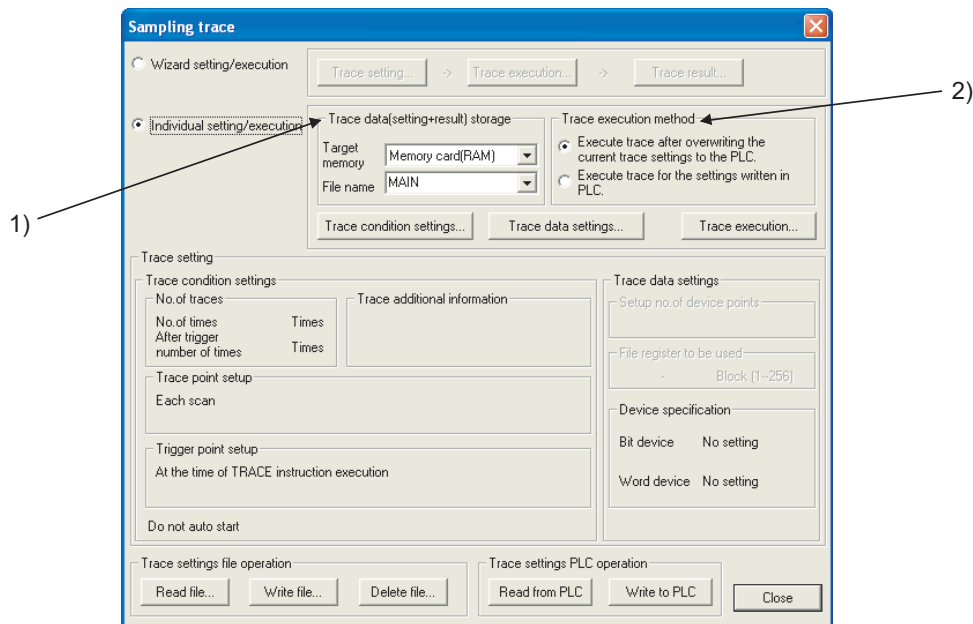


Figure 6.66 Sampling trace screen

1) Trace data (setting + result) storage

Select the memory for storing the trace data and the file for writing the trace conditions.

Select either "Standard RAM" or "Memory card (RAM)" for the target memory setting.

Trace results will be stored in the memory set here under the selected file name.

2) Trace execution method

Select either of the following trace execution method.

- "Execute trace after overwriting the current trace setting to the PLC":
The CPU module executes the sampling trace after the trace settings are overwritten to the existing sampling trace file.
- "Execute trace for the settings written in PLC":
The CPU module executes the sampling trace with the trace settings in the sampling trace file selected for "Trace data (setting + result) storage".

(b) Setting trace conditions

Set trace conditions on the screen opened by clicking the **Trace condition setting** button on the screen shown in Figure 6.66.

On the Trace condition settings screen, set the following items.

- Number of traces ("No. of traces", "After trigger number of times")
- Trace point setting ("Trace point setup")
- Trigger point setting ("Trigger point setup")
- Additional information ("Trace additional informaton")
- Auto start setting ("Auto start trace")

Figure 6.67 Trace condition settings screen

1) No. of traces

There are two items need to be set: "No. of times" and "After trigger number of times".

- No. of times: Select the number of executions from the start to the end of sampling trace.
- After trigger number of times: Select the number of executions from the trigger point to the end of sampling trace.

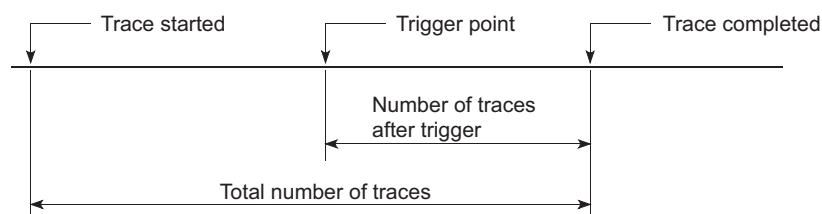


Figure 6.68 Relationship between two setting items

Set the numbers for each items within the following setting range.

$$(\text{"After trigger number of times"}) \leq (\text{"No. of times"}) \leq (8192)$$

2) Trace point setup

Select the timing for collecting trace data from the items listed in Table6.23.

Table6.23 Trace point setup item

Item	Description													
Each scan	Collects trace data during END processing of each scan.													
Interval	Collects trace data at specified time intervals.													
Each multiple CPU high speed communication cycle*1	Collects trace data in a cycle of 0.88m × specified time intervals. Since data is collected at the timing of interrupt program (I45) execution, trace data can be collected only when the following conditions are all satisfied. • The multiple CPU high-speed main base unit (Q3 □ DB) is used. • The multiple CPU system where two or more CPU modules are used and the multiple CPU high speed transmission function is set. • Interrupt pointer (I45) exists in a program. • Interrupts are enabled and interrupt mask of I45 is cancelled.													
Detail	A trace point (device and/or step number) needs to be set. The following devices can be set as a trace point. • Bit device: X(DX), Y(DY), M, L, F, SM, V, B, SB, T(contact), ST(contact), C(contact), FX, FY, J □ \X, J □ \Y, J □ \B, J □ \SB, BL □ \S • Word device: T(current value), ST(current value), C(current value), D*1, SD, W*2, SW, R, Z, ZR, FD, U □ \G, J □ \W, J □ \SW, U3E □ \G The following modifications are available for the above devices. • Digit specification of bit device • Bit specification of word device • Indirect specification of word device • Index modification When the set conditions are met, data collection is performed. The following shows the conditions of setting items. <table><tr><th>Device</th><th>Conditional formula</th><th>Description</th></tr><tr><td rowspan="2">Bit device</td><td>↑</td><td>Data are collected on the rising edge of the specified device/label.</td></tr><tr><td>↓</td><td>Data are collected on the falling edge of the specified device/label.</td></tr><tr><td rowspan="2">Word device</td><td>When values match</td><td>Data are collected when the current value of the specified device/label is equal to the condition value.</td></tr><tr><td>When values are changed</td><td>Data are collected when the current value of the specified device/label is changed.</td></tr></table> The collection timing for trace data when "Step No." is selected is the same as when setting the monitor conditions. (☞ Section 6.11.1)	Device	Conditional formula	Description	Bit device	↑	Data are collected on the rising edge of the specified device/label.	↓	Data are collected on the falling edge of the specified device/label.	Word device	When values match	Data are collected when the current value of the specified device/label is equal to the condition value.	When values are changed	Data are collected when the current value of the specified device/label is changed.
Device	Conditional formula	Description												
Bit device	↑	Data are collected on the rising edge of the specified device/label.												
	↓	Data are collected on the falling edge of the specified device/label.												
Word device	When values match	Data are collected when the current value of the specified device/label is equal to the condition value.												
	When values are changed	Data are collected when the current value of the specified device/label is changed.												

*1: The extended data register (D) is included.

*2: The extended link register (W) is included.

3) Trace additional information

Set the information added for each trace.

Select one or more items from the following. (If not necessary, do not select any item.)

- Time(sec.): Stores the time when the trace was executed.
- Step no.: Stores the step number where the trace was executed.
- Program name: Stores the program name where the trace was executed.

4) Trigger point setup

Select the trigger point from the items listed in Table6.24.

Table6.24 Trigger point setup item

Item	Description												
At the time of TRACE instuction execution	The time of execution of the TRACE instruction is set as a trigger.												
At the time of trigger operation from GX Developer	The time when a trigger is issued by GX Developer is set as a trigger.												
Detail	A trace point (device and/or step number) needs to be set. The following devices can be set as a trace point. • Bit device: X(DX), Y(DY), M, L, F, SM, V, B, SB, T(contact), ST(contact), C(contact), FX, FY • Word device: T(current value), ST(current value), C(current value), D^{*1}, SD, W^{*2}, SW, R, ZR												
	The following modification is available for the above devices. • Bit specification of word device												
	Indirectly-specified devices cannot be set.												
	A trigger point is set as the timing of when the set conditions are met. The following shows the conditions of setting items.												
	<table><tr><th>Device</th><th>Conditional formula</th><th>Description</th></tr><tr><td rowspan="2">Bit device</td><td>↑</td><td>A trigger occurs at the rising edge of the specified device/label.</td></tr><tr><td>↓</td><td>A trigger occurs at the falling edge of the specified device/label.</td></tr><tr><td rowspan="2">Word device</td><td>When values match</td><td>A trigger occurs when the current value of the specified device/label is equal to the condition value.</td></tr><tr><td>When writing into devices</td><td>A trigger occurs when a value is written into the specified device/label.</td></tr></table>	Device	Conditional formula	Description	Bit device	↑	A trigger occurs at the rising edge of the specified device/label.	↓	A trigger occurs at the falling edge of the specified device/label.	Word device	When values match	A trigger occurs when the current value of the specified device/label is equal to the condition value.	When writing into devices
Device	Conditional formula	Description											
Bit device	↑	A trigger occurs at the rising edge of the specified device/label.											
	↓	A trigger occurs at the falling edge of the specified device/label.											
Word device	When values match	A trigger occurs when the current value of the specified device/label is equal to the condition value.											
	When writing into devices	A trigger occurs when a value is written into the specified device/label.											
	The collection timing for trace data when "Step No." is selected is the same as when setting the monitor conditions. (☞ Section 6.11.1)												

*1: The extended data register (D) is included.

*2: The extended link register (W) is included.

5) Auto start trace Note6.9

When the checkbox is selected, the sampling trace will be started automatically when:

- the CPU module is powered off and then on after the trace settings are written, or
- the CPU module status is changed from STOP to RUN.

Point

When the CPU module is powered off and then on, reset, or the CPU module status is changed from STOP to RUN to rewritten the sampling trace settings, the settings cannot be rewritten to the CPU module regardless of the status of SM829 (Forced registration specification of trace setting) if the trigger condition has already been established. In this case, the CPU module will be in the status where the sampling trace setting registration has been canceled. SM800 (Trace preparation) turns off.



Note6.9

Universal

When using the sampling trace auto start function for the Q02UCPU, Q03UDCPU, Q04UDHCPU, Q06UDHCPU, Q13UDHCPU, or Q26UDHCPU, check the versions of the CPU module and GX Developer. ( Appendix 2)

(c) Setting trace data

Set trace data on the screen opened by clicking the **Trace data setting** button on the screen shown in Figure 6.66.

Table6.25 shows the devices can be set as trace data.

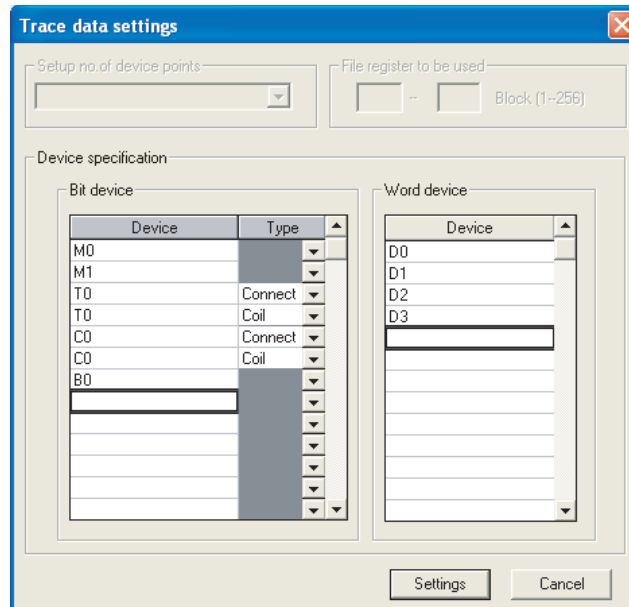


Figure 6.69 Trace data settings screen

Table6.25 Devices can be set as trace data

Item	Description
Bit device	The following bit devices can be set up to 50 points. X, DX, Y, DY, M, L, F, SM, V, B, SB, T(contact), T(coil), ST(contact), ST(coil), C(contact), C(coil), J □ \X, J □ \Y, J □ \B, J □ \SB, BL □ \S
Word device	The following word devices can be set up to 50 points. T (current value), ST (current value), C (current value), D^{*1}, SD, W^{*2}, SW, R, Z, ZR, U □ \G, J □ \W, J □ \SW, U3E □ \G The following modifications are available for the above devices. • Digit specification of bit device • Bit specification of word device • Index modification Indirectly-specified devices cannot be set.

*1: The extended data register (D) is included.

*2: The extended link register (W) is included.

(d) Writing the trace condition settings and trace data settings

Write the set trace conditions and trace data to the memory selected as a sampling trace file for "Trace data (setting + result) storage".

Click the **Write to PLC** button on the screen shown in Figure 6.66 to write the settings.

Point

When storing the sampling trace file into a memory card (SRAM card), more than one sampling trace files can be stored by changing the file name.

For the standard RAM, only one sampling trace file can be stored.

When multiple sampling trace files are used, use the memory card (SRAM card).

(e) Executing the sampling trace

Click the **Trace execution** button on the screen shown in Figure 6.66 to open the Execute sampling trace screen.

Select an item shown in Table 6.26 and click the **Execute** button.

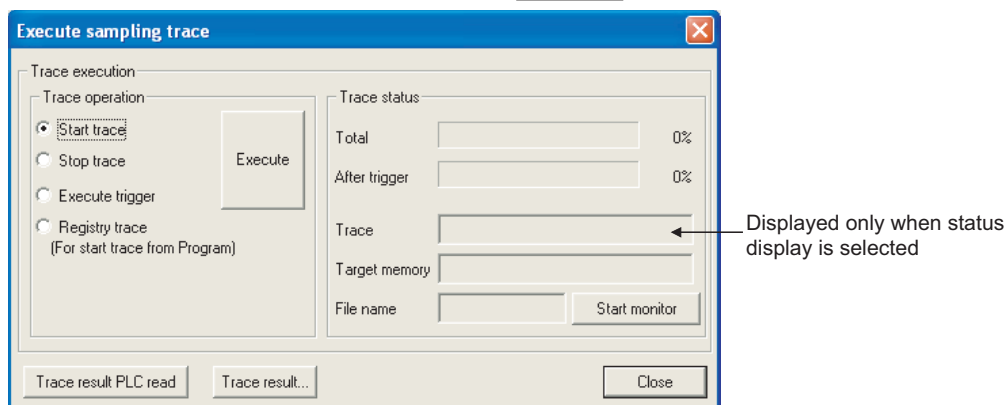


Figure 6.70 Execute sampling trace screen

Table 6.26 Trace point setting

Item	Description
Start trace	Starts the function, and starts counting the number of sampling trace executions.
Stop trace	Stops the function, and clears the total sampling trace execution count and the execution count after trigger. (To restart the function, select "Start trace" again.)
Execute trigger	Executes a trigger, and starts counting the number of sampling trace executions after trigger. The function will be ended when the trace execution count after trigger reaches the preset count.
Registry trace (For start trace from Program)	Registers trace data when a program is executed.

(f) Displaying trace results

Read trace results from the CPU module and display the data.

- 1) Click the **Trace result PLC read** button on the screen shown in Figure 6.70 to read trace results.
- 2) Click the **Trace result** button on the same screen to display the trace results read.

The trace results shows the on/off status of each bit device for every sampling cycle and the current value of each word device.

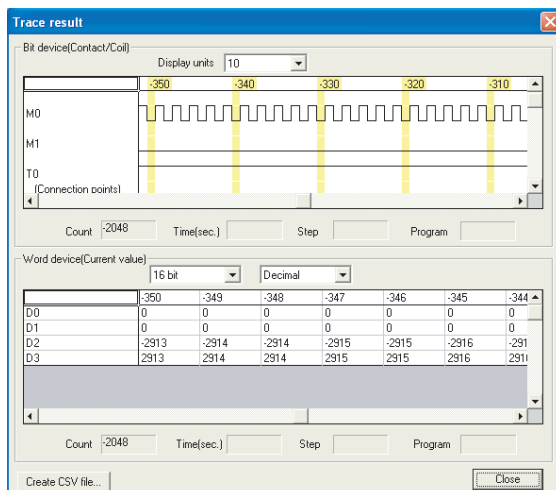


Figure 6.71 Trace result screen

Point

Specified devices are read when the condition selected in "Trigger point setup" (trigger condition) is established. Therefore, when devices are sampled in every scan and the sampling is finished by trigger operation from a peripheral, the data is sampled twice because timing of the sampling is the same with that of the establishment of trigger condition.

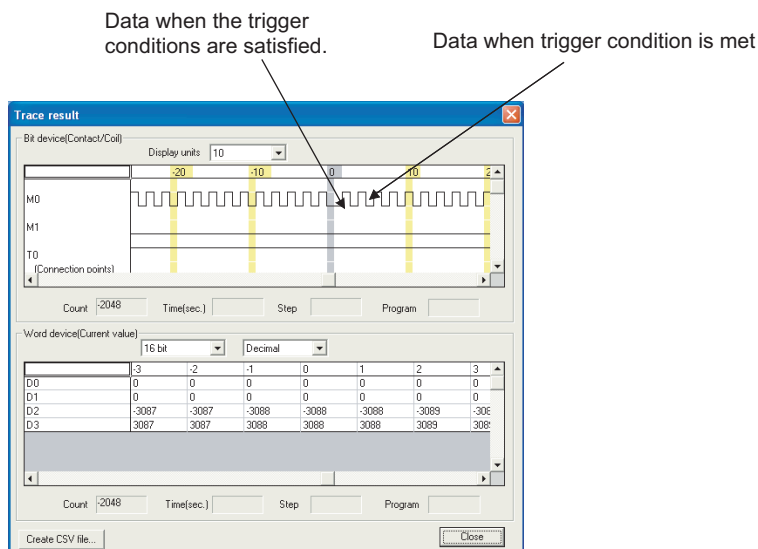


Figure 6.72 Trace result

(6) Method for clearing trace execution status

The trace execution status can be cleared by latch clear using the remote latch clear operation.

( Section 6.6.4)

To perform the sampling trace again after latch clear, select "Start trace" or "Registry trace".

(7) Precautions

(a) Areas where sampling trace can be performed

The sampling trace can be performed from other stations on the network or serial communication module.

However, it cannot be performed from multiple devices simultaneously. It can be performed from one device to the CPU module.

(b) Holding and clearing the trace setting

The trace setting (sampling trace file) registered with the CPU module is latched.

Even if the CPU module is powered off and then on or is reset, the sampling trace can be performed again with the trace setting at registration.

However, the previous trace result cannot be read.

Also in the following cases, even when the trigger condition of the sampling trace is established, the latched trace setting will be cleared since the condition is not recognized as the trigger condition (SM800 (Trace preparation) turns off).

Register the trace setting again with GX Developer.

- 1) When selecting "Standard RAM" in "Target memory", configuring the setting that changes the local device size in the standard RAM*1, writing parameters to the CPU module, and then performing any of the following operations

- The CPU module is powered off and then on
- The CPU module is reset.
- The CPU module is set from STOP to RUN.

*1: The operation includes when a local device is created.

- 2) When selecting "Standard RAM" in "Target memory" and the sampling trace file is corrupt, either of the following operations were performed.

- The CPU module is powered off and then on.
- The CPU module is reset.

Point

To keep the trace result to the personal computer even after configuring the setting that changes the local device size, perform the following operations.

- Click the Trace result PLC read button on the screen shown in Figure 6.70 to read the trace result to the personal computer.
- Click the Trace result button on the screen shown in Figure 6.70 to display the trace result.
- Click the Create CSV file button on the screen shown in Figure 6.71 to save the trace result in CSV format.

3) When selecting "Memory card (RAM)" in "Target memory" while the SRAM card where the sampling trace file has been registered is not mounted, either of the following operations were performed.

- The CPU module is powered off and then on.
- The CPU module is reset.

4) When selecting "Memory card (RAM)" in "Target memory" and the sampling trace file is corrupt, either of the following operations were performed.

- The CPU module is powered off and then on.
- The CPU module is reset.

(c) Reading trace result in the STOP status

The trace result cannot be read while the CPU module is in the STOP status.

When reading the trace result, read it while the CPU module is in the RUN status.

(d) Sampling trace registration while the trigger condition is established

Even if the trigger condition is established, the sampling trace setting can be registered by the following procedure.

- 1) Turn on SM829 (Forced registration specification of trace setting).
- 2) Select "Individual setting/execution" in the screen shown in Figure 6.66.
- 3) Click the Trace execution button, select "Registry trace (For start trace from Program)", and click the Execute button.

Point

- When the sampling trace is registered by the procedures above (d) 1) to 3), start the sampling trace in status where the trigger condition is not established.
If the trigger condition has been established when the sampling trace starts, the trigger may not be normally executed.
- When selecting "Wizard setting/execution" in the Sampling trace screen shown in (d) 2), and starting the sampling trace, the trigger may not be normally executed, although the sampling trace setting can be registered.

(e) When a file register is selected as a specified device by the detail setting of trace conditions

When a file register is selected as a specified device by the detail setting of trace point setting and trigger point setting, do not change the block numbers of file register file and file register after trace registration.

Trace data may not be normally sampled.

(f) Trace point setting

When setting the trace point setting per each time or per multiple CPU high speed transmission cycle, pay attention to the sampling interval and sampling processing time for one sampling since the sampling trace is performed as interrupt processing.

If the sampling processing time for one sampling is long, "WDT ERROR" may be detected.

(g) Performing sampling trace during execution of another sampling trace

The first sampling trace is performed normally. The second sampling trace cannot be performed.

(h) Executing online change

When sampling trace and online change are performed simultaneously, they operate as follows.

1) Performing sampling trace during online change

- The trace point or trigger point is specified by the step number:
The online change is completed normally but the sampling trace is not performed.
- The trace point and trigger point are specified by except the step number:
Both the online change and sampling trace can be performed.

2) Performing online change during execution of sampling trace

- The trace point or trigger point is specified by the step number:
The sampling trace is suspended but the online change is normally performed.
- The trace point and trigger point are specified by except the step number:
Both the online change and sampling trace can be performed.

6.15 Debug Function from Multiple GX Developers

(1) Definition

This function allows debugs from multiple GX Developers connected to such as a CPU module or serial communication module.

When files are divided according to the processes or functions, this function can be used when multiple GX Developers debug different files.

(2) Description

The Table6.27 shows the combinations of the debug functions which can be executed from multiple GX Developers.

Table6.27 Combinations of the debug functions

Functions in execution	Functions executed later							
	Ladder monitor, Device batch monitor, Entry data monitor, Entry ladder monitor, Local device monitor	Program monitor list	Interrupt program monitor list	Monitor condition setting	Online change	Scan time measurement	Sampling trace	Executorial conditioned device test
Ladder monitor, Device batch monitor, Entry data monitor, Entry ladder monitor, Local device monitor	○	○	○	○	○	○	○	○
Program monitor list	○	×	○	○	○	○	○	○
Interrupt program monitor list	○	○	×	○	○	○	○	○
Monitor condition setup	○	○	○	×	△ ^{*1}	○	○	○
Online change	○	○	○	×	×	×	×	×
Scan time measurement	○	○	○	○	×	×	○	○
Sampling trace	○	○	○	○	△ ^{*2}	○	×	○
Executorial conditioned device test	○	○	○	○	△ ^{*3}	○	○	○

○ : Can be simultaneously performed, △ : Partially restricted, × : Can not be simultaneously performed

*1: Cannot be performed simultaneously when the step number, or the step number and device number, are set in the monitor condition.

*2: Cannot be performed simultaneously unless the step number is set to the trace point or the trigger point.

*3: Cannot be performed simultaneously in the following cases.

- The data to be changed online includes the registration of an executorial conditioned device test.
- When adding a ladder block by online change, registration of an executorial conditioned device test is included in the ladder block immediately after the one where the ladder block is to be added.
- The program to be changed online includes registration of an executorial conditioned device test.

6.15.1 Simultaneous monitoring from multiple GX Developers function

(1) Definition

This function allows simultaneous monitoring from multiple GX Developers connected to such as a CPU module or serial communication module.

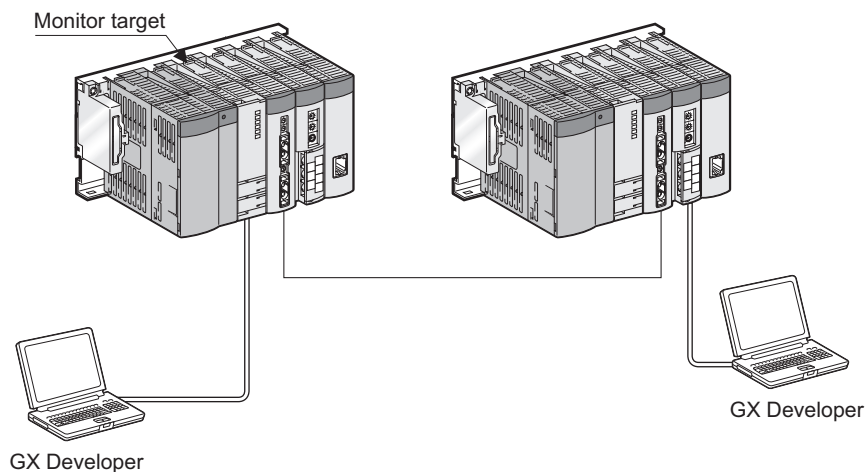


Figure 6.73 Simultaneous monitoring

Creating a user setting system area allows high-speed monitoring from multiple GX Developers (Setting a monitoring file for the host station is unnecessary).

(2) Setting for simultaneous monitoring from multiple GX Developers

Create a user setting system area in the following procedure.

- Select [Online] → [Format PLC memory] in GX Developer to open the screen shown in Figure 6.74.
- Select "Program memory/Device memory" in "Target Memory".
- Select "Create a user setting system area" in "Format Type".
- Set the number of steps for the system area (in increments of: 1K step).

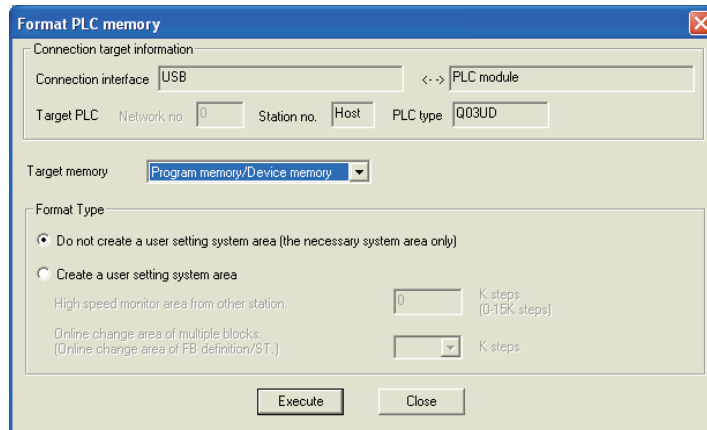


Figure 6.74 System area setting (when 1K step is set)

Table 6.28 shows the maximum number of steps settable in the system area.
1K step is available for a monitoring file from another station.

Table 6.28 Maximum size of steps settable in the system area

Maximum size of settable step	System area for monitoring from another station
Maximum 15K steps	Maximum 15

(3) Precautions

(a) Monitor condition setting

The monitor conditions can be set from one GX Developer.

(b) Necessity of system area setting

Although multiple GX Developers in other stations can simultaneously monitor a CPU module without the user setting system area, the monitor speed will be slow.

Since the system area is set in the program memory, the area for storing programs reduces by the size of set system area.

(c) The number of GX Developers for which high-speed monitoring can be set

The number of GX Developers that can simultaneously monitor a CPU module at high-speed is "the number of user setting system areas (the number of K steps) + 1".

For example, when user setting system area of 15K steps is created, maximum 16 GX Developers can simultaneously monitor a CPU module at high-speed.

6.15.2 Online change function from multiple GX Developers

(1) Definition

This function allows multiple GX Developers to perform online change to one file or different files.

- Online change to one file:
Select "Relative step No. by pointer".
- Online change to different files:
- The writing can be executed without selecting "Relative step No. by pointer".

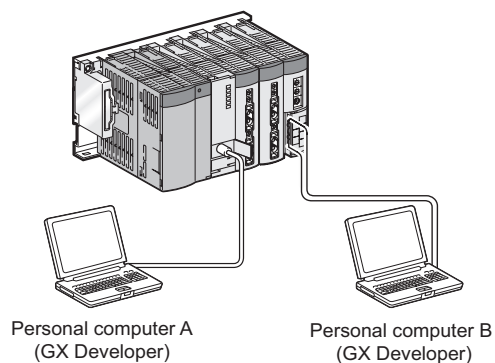


Figure 6.75 Simultaneous online change from multiple GX Developers

(2) Operating procedure for performing online change to one file

Select [Tools] → [Options] → <Program common> tab in GX Developer.→

Set a pointer for online change beforehand.

(a) Setting “After conversion writing behavior” and “Step No. specification used in writing”

Set them as follows:

- 1) Select "Write during RUN (while PLC is running)" in "After conversion writing behavior".
- 2) Select “Absolute step No. (default)” or “Relative step No. by pointer” in “Step No. specification used in writing”.

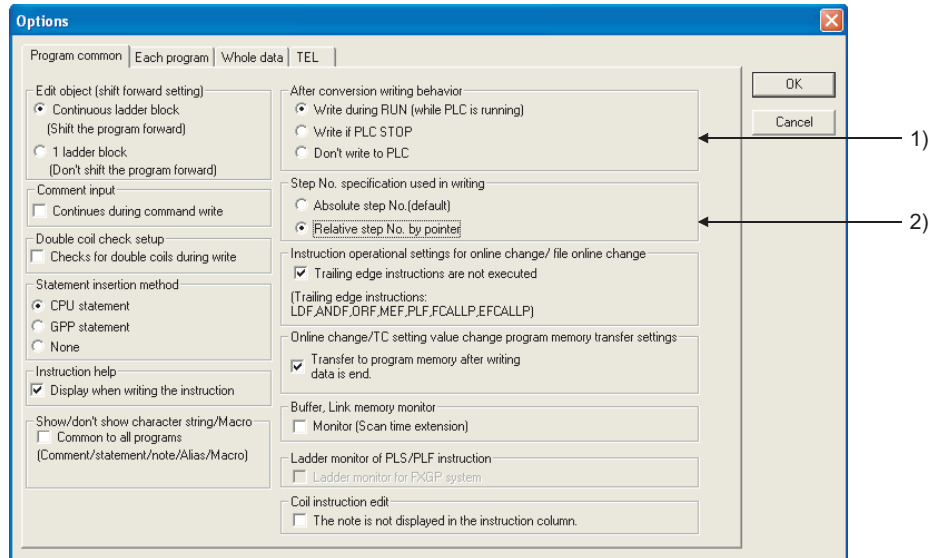


Figure 6.76 Options screen

(b) Performing online change

Display the ladder including the specified pointer and write the changed ladder during RUN.

(3) Precautions

Precautions for online change from multiple GX Developers are the same as those for usual Write during RUN

( Section 6.12.3).

6.16 Watchdog Timer (WDT)

(1) Definition

This function serves as an CPU module internal timer to detect errors of CPU module hardware and sequence programs.

(2) Setting and resetting

(a) Setting

The watchdog timer setting can be changed in the PLC RAS setting of PLC parameter.

The default is set to 200ms.

The setting range is 10 to 2000ms (in increments of: 10ms).

(b) Reset

The CPU module resets the watchdog timer during END processing.

- The watchdog timer does not time up when the CPU module operates normally and the END/FEND instruction is executed within the setting value of watchdog timer.
- The watchdog timer times up when the scan time of the sequence program is extended and the END/FEND instruction could not be executed within the setting value of watchdog timer due to the hardware failure of the CPU module or execution of an interrupt program/fixed scan execution type program.

(3) When the watchdog timer times up

“WDT ERROR” is detected and the following status occurs:

- 1) The CPU module turns off all outputs.
- 2) The RUN LED on the front of the CPU module turns off and the ERR. LED starts flashing.
- 3) SM1 turns on and the error codes 5000 and 5001 are stored in SD0.

(4) Precautions

(a) Watchdog timer error

An error is observed within the range of 0 to 10ms.

Set a watchdog timer while considering such an error.

(b) Resetting a watchdog timer when a program is repeatedly executed between the FOR and NEXT instructions

The watchdog timer can be reset by executing the WDT instruction in the sequence program.
To avoid the time up of watchdog timer while a program is repeatedly executed between the FOR and NEXT instructions, reset the watchdog timer by the WDT instruction.

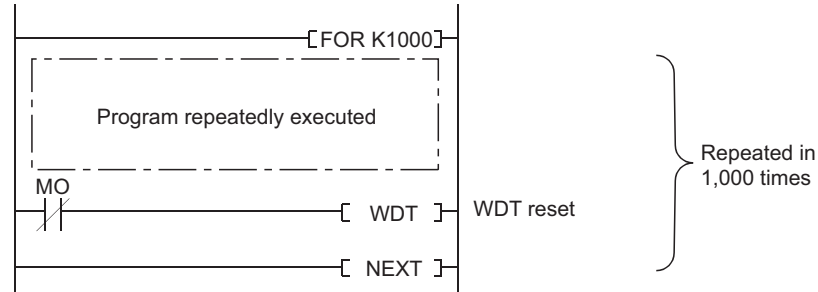


Figure 6.77 Resetting a watchdog timer when the program is executed between the FOR and NEXT instructions

(c) Scan time when using the WDT instruction

The scan time value is not reset even if the watchdog timer is reset in the sequence program.
The scan time is measured up to the END instruction.

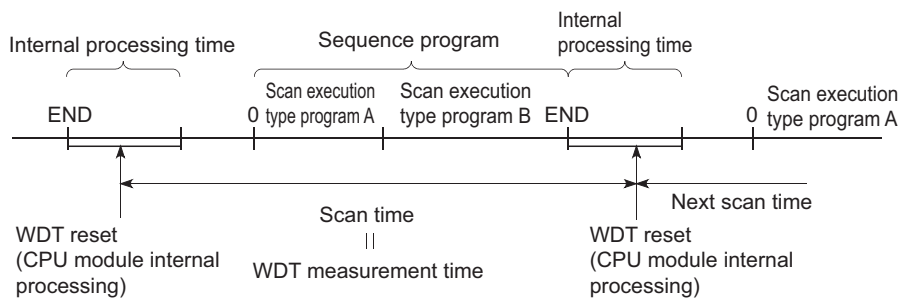


Figure 6.78 Watchdog timer reset

Point

- A scan time is time required for the CPU module to operate the sequence program from step 0 and return to the step 0 in the sequence program with the same file name.
- The scan time depends on the execution status of the following:
 - Instructions used in the program
 - Interrupt program and fixed scan execution type program

To keep the same scan time in every scan, use the constant scan function. (☞ Section 6.2)

6.17 Self-diagnostic Function

(1) Definition

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

This function aims to preventive measures and prevention of malfunction of the CPU module.

(2) Self-diagnostic timing

When an error occurs at power-on or during the RUN or STOP status of the CPU module, the error is detected and displayed by the self-diagnostic function, and the CPU module stops an operation.

Note that errors cannot be detected by the function depending on error status or an instruction executed.

When the operation is not stopped by the function, configure a safety circuit external to the programmable controller so that the entire system operates safely.

(3) Checking errors

(a) LED status

When the CPU module detects an error, the ERR. LED turns on.

(b) Storage location of error information and error check

When the CPU module detects an error, the special relays (SM0, SM1) turn on and the error information (error code) are stored in the special register (SD0).

When several errors are detected, the latest error code is stored in SD0.

Use the special relays and special register in a program as an interlock for the programmable controller and mechanical system.

(4) Checking error history

To check the latest error code, select [Diagnostics] → [PLC diagnostics] → "Error log" in GX Developer.

The error data history are backed up using a battery even after the programmable controller is powered off.

(5) CPU module operation at error detection

(a) Mode at error detection

When an error is detected by the self-diagnostic function, the CPU module enters either of the following modes.

1) Mode that stops CPU module operation

When an error is detected, the CPU module stops an operation and turns off all external outputs of the module set to "Clear" in "Error time output mode" in "Detailed setting" of the I/O assignment tab of the PLC parameter dialog box (Outputs (Y) in the device memory are held).

Note that the external outputs of the module set to "Hold" in "Error time output mode" are held (Outputs (Y) in the device memory are held).

2) Mode that continues CPU module operation

When an error is detected, the CPU module operates programs other than the one (instruction) where an error occurred.

(b) Errors whether to continue or stop an operation can be selected

Whether to continue or stop an operation can be selected in the following errors.

1) Errors whether to continue or stop an operation can be selected in the PLC RAS tab of the PLC parameter dialog box

- Computation error (including SFC program)
- Expanded command error (setting for future extension)
- Fuse blown
- Module verify error
- Intelligent module program execution error
- File access error
- Memory card operation error
- External power supply OFF (setting for future extension)

For example, when "Module verify error" is set to "Continue", an operation is continued from the I/O number before an error.

For details of errors, refer to "Self-diagnostics list". (Fig. 7) in this section)

2) Error whether to continue or stop an operation can be selected in "Detailed setting" in the I/O assignment tab of the PLC parameter dialog box

- Intelligent function module error

(6) Error check options

Whether to check the following errors or not can be selected in the PLC RAS tab of the PLC parameter dialog box (All the options are selected (executed) by default).

- 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.  Note6.10

**Note6.10****Universal**

When setting the diagnostic function of the redundant power supply system for the Q02UCPU, Q03UDCPU, Q04UDHCPU, Q06UDHCPU, Q13UDHCPU, or Q26UDHCPU, check the versions of the CPU module and GX Developer. ( Appendix 2)

(7) Self-diagnostics list

The following table shows the self-diagnostics performed by the CPU module.

To check the error messages in the "Error message" column of Table6.29, select [Diagnostics] → [PLC diagnostics] of GX Developer.

Table6.29 Self-diagnostics list

Diagnostics	Error message	Diagnostic timing	CPU module status	LED status		Q00UJ CPU	Q00U CPU, Q01U CPU	Q02U CPU	QnUD(H) CPU	Built-in Ethernet port QCPU
				RUN	ERR.					
Hardware failure	CPU error	MAIN CPU DOWN	• Always	Stop	Off	Flashing	○	○	○	○
	END instruction not executed	END NOT EXECUTE	• Execution of the END instruction	Stop	Off	Flashing	○	○	○	○
	SFC program execution error	SFCP. END ERROR	• Execution of a SFC program	Stop	Off	Flashing	○	○	○	○
	RAM check	RAM ERROR	• Power-on/reset	Stop	Off	Flashing	○	○	○	○
	Operation circuit check	OPE.CIRCUIT ERR.	• Power-on/reset • Execution of the END instruction	Stop	Off	Flashing	○	○	○	○
	Fuse blown *1 *2	FUSE BREAK OFF	• Always	Stop/continue	Off/on	Flashing/on	○	○	○	○
	I/O interrupt error	I/O INT. ERROR	• Occurrence of an interrupt	Stop	Off	Flashing	○	○	○	○
	LAN controller failure	LAN CTRL. DOWN	• Power-on/reset	Stop	Off	Flashing	○	○	○	○
	Intelligent function module error *1	SP.UNIT DOWN	• Power-on/reset • Execution of the FROM/TO instructions • Execution of the intelligent function module dedicated instruction • Execution of the END instruction	Stop/continue	Off/on	Flashing/on	○	○	○	○
	Control bus error	CONTROL-BUS ERR.	• Power-on • Execution of END processing • Execution of the FROM/TO instructions • Execution of the intelligent function module dedicated instruction • Always	Stop	Off	Flashing	○	○	○	○
	Momentary power failure	AC/DC DOWN	• Always	Continue	On	Off	○	○	○	○
	Multiple CPU high speed bus error	MULTI-C.BUS ERR.	• Power-on/reset	Stop	Off	Flashing	×	×	×	○

○ : Self-diagnostics is performed. × : Self-diagnostics is not performed.

(To the next page)

Table6.29 Self-diagnostics list (continued)

Diagnostics		Error message	Diagnostic timing	CPU module status	LED status		Q00UJ CPU	Q00U CPU, Q01U CPU	Q02U CPU	QnUD(H) CPU	Built-in Ethernet port QCPU
					RUN	ERR.					
Handling error	Voltage drop of power supply for redundant base unit	SINGLE PS. DOWN	• Always	Continue	On	On	×	○	○ *5	○ *5	○
	Redundant power supply module failure	SINGLE PS. ERROR	• Always	Continue	On	On	×	○	○ *5	○ *5	○
	Module verification ^{*1*2}	UNIT VERIFY ERR.	• Execution of the END instruction	Stop/continue	Off/on	Flashing/on	○	○	○	○	○
	Base assignment error	BASE LAY ERROR	• Power-on/reset	Stop	Off	Flashing	○	○	○	○	○
	Intelligent function module assignment error	SP.UNIT LAY ERR.	• Power-on/reset • Switching from STOP to RUN	Stop	Off	Flashing	○	○	○	○	○
	Intelligent program execution error ^{*1}	SP.UNIT ERROR	• Execution of the FROM/TO instructions	Stop/continue	Off/on	Flashing/on	○	○	○	○	○
	Intelligent function module version error	SP.UNIT VER.ERR	• Power-on/reset	Stop	Off	Flashing	○	○	○	○	○
	No parameter	MISSING PARA.	• Power-on/reset • Switching from STOP to RUN	Stop	Off	Flashing	○	○	○	○	○
	Boot error	BOOT ERROR	• Power-on/reset	Stop	Off	Flashing	○	○	○	○	○
	Backup error	RESTORE ERROR	• Power-on/reset	Stop	Off	Flashing	○	○	○	○	○
	Memory card operation error ^{*1}	ICM.OPE. ERROR	• Mounting/removal of the memory card	Stop/continue	Off/on	Flashing/on	×	×	○	○	○
	File setting error	FILE SET ERROR	• Power-on/reset • Writing to programmable controller	Stop	Off	Flashing	○	○	○	○	○
	File access error ^{*1}	FILE OPE. ERROR	• Execution of an instruction	Stop/continue	Off/on	Flashing/on	○	○	○	○	○
	Instruction execution disabled	CAN&T EXE.PRG.	• Power-on/reset • Switching from STOP to RUN	Stop	Off	Flashing	○	○	○	○	○
Parameter error	Parameter setting check	PARAMETER ERROR	• Power-on/reset • Switching from STOP to RUN • Writing to programmable controller	Stop	Off	Flashing	○	○	○	○	○
	Link parameter error	LINK PARA.ERROR	• Power-on/reset • Switching from STOP to RUN	Stop	Off	Flashing	○	○	○	○	○

○ : Self-diagnostics is performed. × : Self-diagnostics is not performed.

(To the next page)

Table6.29 Self-diagnostics list (continued)

Diagnostics	Error message	Diagnostic timing	CPU module status	LED status		Q00UJ CPU	Q00U CPU, Q01U CPU	Q02U CPU	QnUD(H) CPU	Built-in Ethernet port QCPU
				RUN	ERR.					
SFC parameter error	SFC PARA.ERROR	- Switching from STOP to RUN - Writing to programmable controller	Stop	Off	Flashing	○	○	○	○	○
Intelligent function module parameter error	SP.PARA.ERROR	Power-on/reset	Stop	Off	Flashing	○	○	○	○	○
Password error	REMOTE PASS.ERR	- Power-on/reset - Switching from STOP to RUN	Stop	Off	Flashing	○	○	○	○	○
Instruction code check	INSTRUCT. CODE ERR	- Power-on/reset - Switching from STOP to RUN - Execution of an instruction	Stop	Off	Flashing	○	○	○	○	○
No END instruction	MISSING END INS.	- Power-on/reset - Switching from STOP to RUN	Stop	Off	Flashing	○	○	○	○	○
Pointer setting error	CANëT SET(P)	- Power-on/reset - Switching from STOP to RUN	Stop	Off	Flashing	○	○	○	○	○
	CANëT SET(I)	- Power-on/reset - Switching from STOP to RUN	Stop	Off	Flashing	○	○	○	○	○
Program error	Operation error*1 *4	OPERATION ERROR	• Execution of an instruction	Stop/continue	Off/on	Flashing/on	○	○	○	○
	FOR to NEXT instructions structure error	FOR NEXT ERROR	• Execution of an instruction	Stop	Off	Flashing	○	○	○	○
	CALL to RET instructions structure error	CANëT EXECUTE(P)	• Execution of an instruction	Stop	Off	Flashing	○	○	○	○
	Interrupt program error	CANëT EXECUTE(I)	• Execution of an instruction	Stop	Off	Flashing	○	○	○	○
	Instruction execution disabled	INST. FORMAT ERR.	• Execution of an instruction	Stop	Off	Flashing	○	○	○	○
	Dedicated instruction of multiple CPU high speed bus error	MULTI COM.ERROR	• Execution of an instruction	Stop	Off	Flashing	×	×	×	○
	SFC block configuration error	CANëT SET(BL)	• Switching from STOP to RUN	Stop	Off	Flashing	○	○	○	○
	SFC step configuration error	CANëT SET(S)	• Switching from STOP to RUN	Stop	Off	Flashing	○	○	○	○
	SFC execution error	SFC EXE. ERROR	• Switching from STOP to RUN	Stop	Off	Flashing	○	○	○	○

○ : Self-diagnostics is performed. × : Self-diagnostics is not performed

(To the next page)

Table6.29 Self-diagnostics list (continued)

Diagnostics		Error message	Diagnostic timing	CPU module status	LED status		Q00UJ CPU	Q00U CPU, Q01U CPU	Q02U CPU	QnUD(H) CPU	Built-in Ethernet port QCPU
					RUN	ERR.					
Program error	SFC syntax error	SFCP. FORMAT ERR.	• Switching from STOP to RUN	Stop	Off	Flashing	○	○	○	○	○
	SFC block execution error	BLOCK EXE.ERROR	• Execution of an instruction	Stop	Off	Flashing	○	○	○	○	○
	SFC step execution error	STEP EXE.ERROR	• Execution of an instruction	Stop	Off	Flashing	○	○	○	○	○
CPU error	Watchdog error supervision	WDT ERROR	• Always	Stop	Off	Flashing	○	○	○	○	○
	Program time-out	PRG.TIME OVER	• Always	Continue	On	On	○	○	○	○	○
Multiple CPU systems error	Another CPU major error	MULTI CPU DOWN	• Always • Power-on/reset	Stop	Off	Flashing	×	○	○	○	○
	Multiple CPU systems execution error	MULTI EXE.ERROR	• Power-on/reset	Stop	Off	Flashing	×	○	○	○	○
	Multiple CPU systems consistency error	CPU LAY. ERROR	• Power-on/reset	Stop	Off	Flashing	×	○	○	○	○
	Another CPU minor error	MULTI CPU ERROR	• Always	Continue	On	On	×	○	○	○	○
File diagnostic check		INCORRECT FILE	• Power-on/reset • Switching from STOP to RUN • Writing to programmable controller	Stop	Off	Off	○	○	○	○	○
Annunciator check		F****	• Execution of an instruction	Continue	On	USER LED turns on.	○	○	○	○	○

○ : Self-diagnostics is performed. × : Self-diagnostics is not performed

*1: The operating status can be changed to "Continue" with the parameter setting of GX Developer The default is set to "Stop".

*2: This option can be set "not checked" (set "checked" by default) with the parameter setting of GX Developer. Also, this option is not checked while SM251 is on.

*3: This option can be set "not checked" (set "checked" by default) with the parameter setting of GX Developer.

*4: This error includes an operation error when device range check is made at index modification.

*5: The CPU module having the serial number (first five digits) is "10042" or later can be diagnosed.

6.17.1 LEDs indicating errors

When an error occurs, the LEDs on the front of the CPU module turns on/flashes. (👉 Section 6.21)

6.17.2 Error clear

The CPU module can clear an error by a program if the error does not stop program operation.

(1) Procedures for error clear

Clear an error by the following procedures.

- Resolve the error cause.
- Store the code of the error to be cleared in the special register SD50.
- Turn off and then on the special relay SM50.
- The error is cleared.

(a) Procedures for error clear in case of multiple errors

When the latest error (error stored in the special register SD0) is cleared, error information stored in special relays and special registers (SM0, SM1, SM5, SM16, SD0 to SD26) are cleared and therefore information on errors that have not been cleared cannot be obtained from the special relays and special registers.

For errors that have not been cleared, obtain the past errors from the error history (👉 Section 6.18) and clear the errors.

(2) Status after error clear

When the CPU module is recovered by clearing an error, the special relay, special register, and LEDs affected by the error return to the status before the error.

If the same error occurs after clearing the error, it is logged in the error history again.

(3) Clear of annunciator

When multiple annunciators are detected, only the first detected "F" is cleared. (👉 Section 9.2.5)

Point 🔑

- When an error is cleared by storing the code of the error to be cleared in SD50, the last digit of the code number is ignored.

Example When the error code 2410, 2411, or 2412 occurs, clearing the error by storing 2412 in SD50 also clears the error codes 2410 and 2411.

- If the CPU module is not an error cause, the error cannot be resolved by using the special relay (SM50) and special register (SD50).

Example Since "SP. UNIT DOWN" indicates an error occurred to the Q bus, the error cannot be resolved by using the special relay (SM50) and special register (SD50).

To resolve the error cause, refer to the following.

👉 QCPU User's Manual (Hardware Design, Maintenance and Inspecting)

6.18 Error History

This function stores an error detected by the self-diagnostic function and the detection time as an error history in a memory.
Select [Diagnostics] → [PLC diagnostics] of GX Developer to check the history.

Point

The detection time is based on the clock in the CPU module. Make sure to set the correct time before the first use of the CPU module. (☞ Section 6.5)

(1) Storage area

All stored logs are saved to the storage memory for error history of the CPU module.

Table6.30 Storage area for error history file

Storage area	Number of storable logs
System memory in CPU module*1	Up to 100*2

*1: The memory is managed inside the system.
*2: When the number of storable logs are exceeded, the latest error log is stored by deletion of the oldest error log.

(2) How to clear error history

To clear the storage memory for error history and error history file, select [Diagnostics] → [PLC diagnostics] in GX Developer and click the Clear log button.
This method clears all data stored in the storage memory for error history of the CPU module and error history file in a memory card.

6.19 System Protection

The CPU module has protection functions (system protection) to prevent programs being modified by a third party other than the designer with GX Developer or serial communication module.

Table 6.31 System protection types

Protection target	File that can be protected	Description	Method	Valid timing	Reference
In units of memory cards ^{*1}	All files	Prohibits writing to a memory card.	Turn on the write protect switch of a memory card.	Always	-
In units of files	• Program • Device comment • Initial device value	• Changes the attribute for each file to either of the following. • Read/write (reading/writing to programs) prohibition • Write prohibition (operations regarding writing such as writing to programs and tests)	Make setting in the Password registration screen.	Always	Section 6.19.1

^{*1}: The Q00UJCPU, Q00UCPU, and Q01UCPU do not support the system protection in units of memory cards since they cannot use the memory card.

6.19.1 Password registration

This function prohibits reading and writing data such as a programs and device comments in the CPU module with GX Developer.

(1) Valid password range

The password can be registered with program, device comment, and initial device values files in the specified memory (program memory, standard ROM, and memory card).

(2) Operations that can be prohibited

The following two operations can be prohibited.

- Reading and writing a file
- Writing a file

When a password has been registered with a file, GX Developer cannot operate the file unless entered password does not match with the registered password.

(3) Setting method

Select [Online] → [Password setup] or click the Password setup button on the Write to PLC screen in GX Developer.

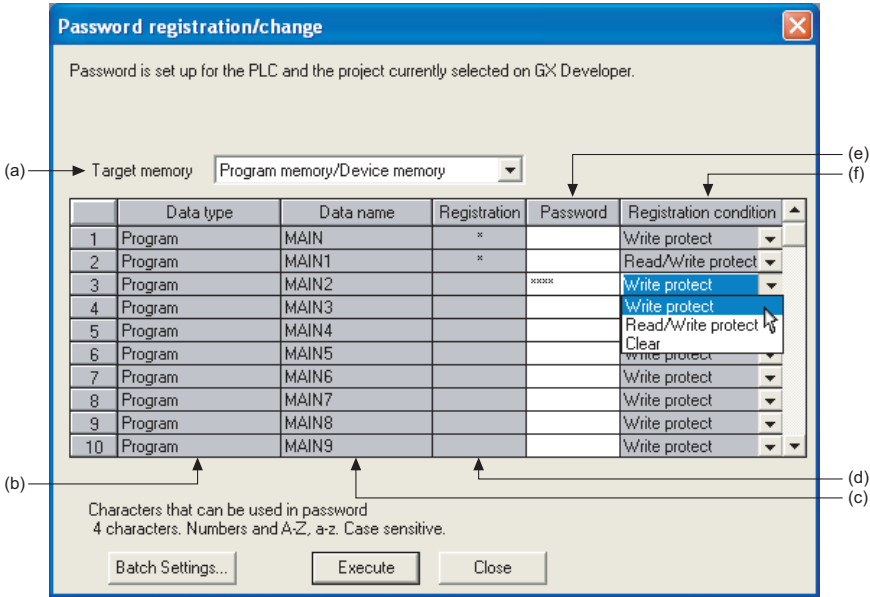


Figure 6.79 Password registration/change screen

(a) Target memory

Select a memory storing a file where a password is to be registered.

(b) Data type

Displays the type of a file stored in the target memory.

(c) Data name

Displays the name of a file stored in the target memory.

(d) Registration

Displays "*" when a password has been set to the target file.

(e) Password

Enter current password or a password to be registered.

The Registration Condition column can be set after setting a password.

(f) Registration Condition

1) Write protect

Writing to a password-protected file is prohibited (reading is allowed).

2) Read/Write protect

Reading and writing to a password-protected file is prohibited.

3) Clear

The set password is cleared (entering the current password is necessary).

(4) Precautions

(a) Password management

A password registered with a file cannot be read from the file.

Forgetting the registered password disables the following operations.

- Program memory or memory card: Format PLC memory
- Standard ROM: Batch-writing

Make sure to record the registered password and store the recording paper.

(b) Operations that overwrite a file independent of the password registration setting

The following operations overwrite a file in the target drive (program memory, standard ROM) independent of the password registration setting.

- Boot from a memory card
- Replacement of the CPU modules with a memory card (backup data restoration function)

6.19.2 Remote password

(1) Definition

This function prevents unauthorized remote access to the CPU module.

If a remote password has been set and the CPU module is remotely accessed, entering a remote password is required.

(2) Settable modules and the number of settable modules

Table6.32 shows the modules for which the remote password can be set and the number of settable modules.

Table6.32 Settable modules and the number of settable modules

Settable module	Number of settable modules
Ethernet module	4
Serial communication module	8
Modem interface module	
Built-in Ethernet port QCPU	1

Point

- The number of settable modules in the above table indicates the number of modules for which the remote password can be set, not the number of mountable modules in the system using the CPU module.
- For the number of mountable modules in the system, refer to the following.
 QCPU User's Manual (Hardware Design, Maintenance and Inspection)
- For details of the remote password of each module, refer to the following.
 Manual for each module

(3) Flow from remote password setting to reflection of the password

Set a remote password ((5) in this section) and then write it to the CPU module.

The remote password is transferred to the target module when the CPU module is powered off and then on or is reset. ((2) in this section)

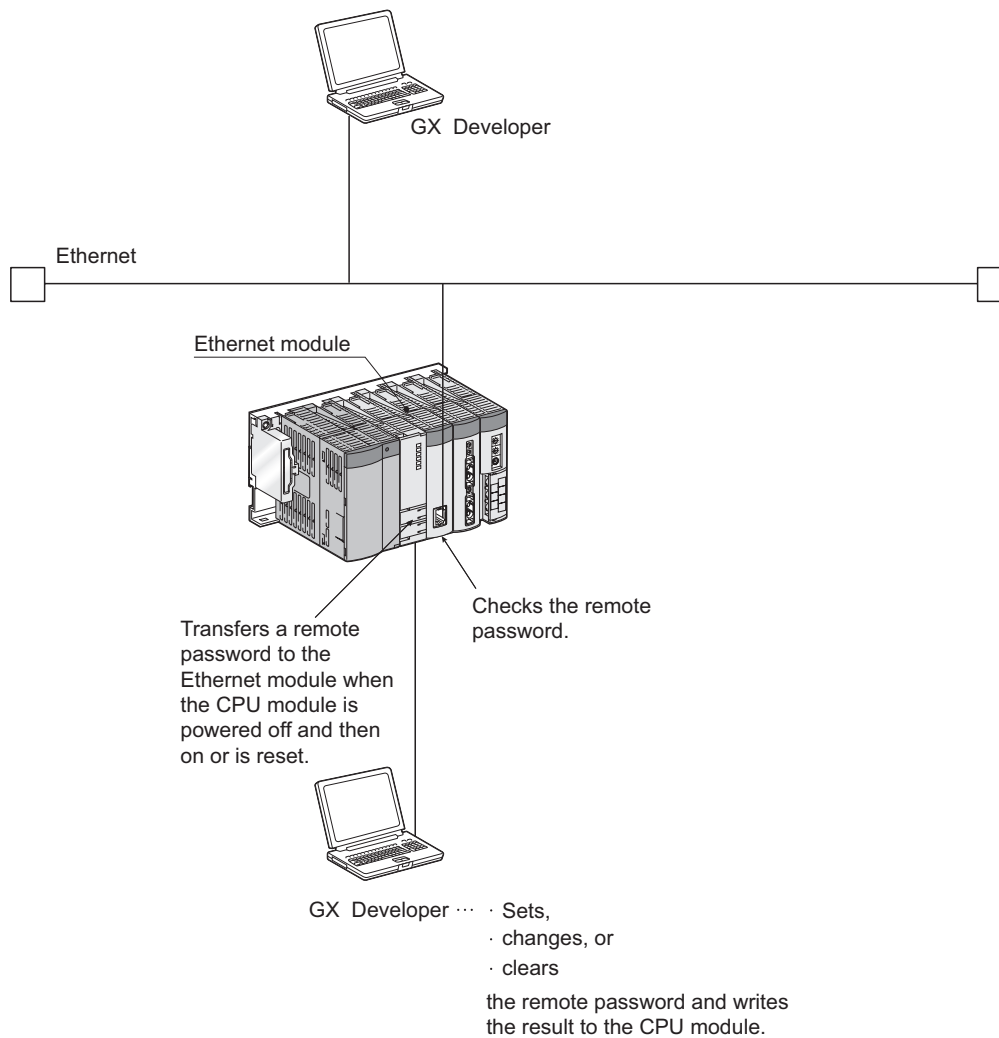


Figure 6.80 Outline of a remote password

(4) Remote password lock/unlock

Unlock the remote password of a serial communication module via a modem or the Ethernet module via Ethernet.
When entered remote password matches with the registered password, the module can access the CPU module.

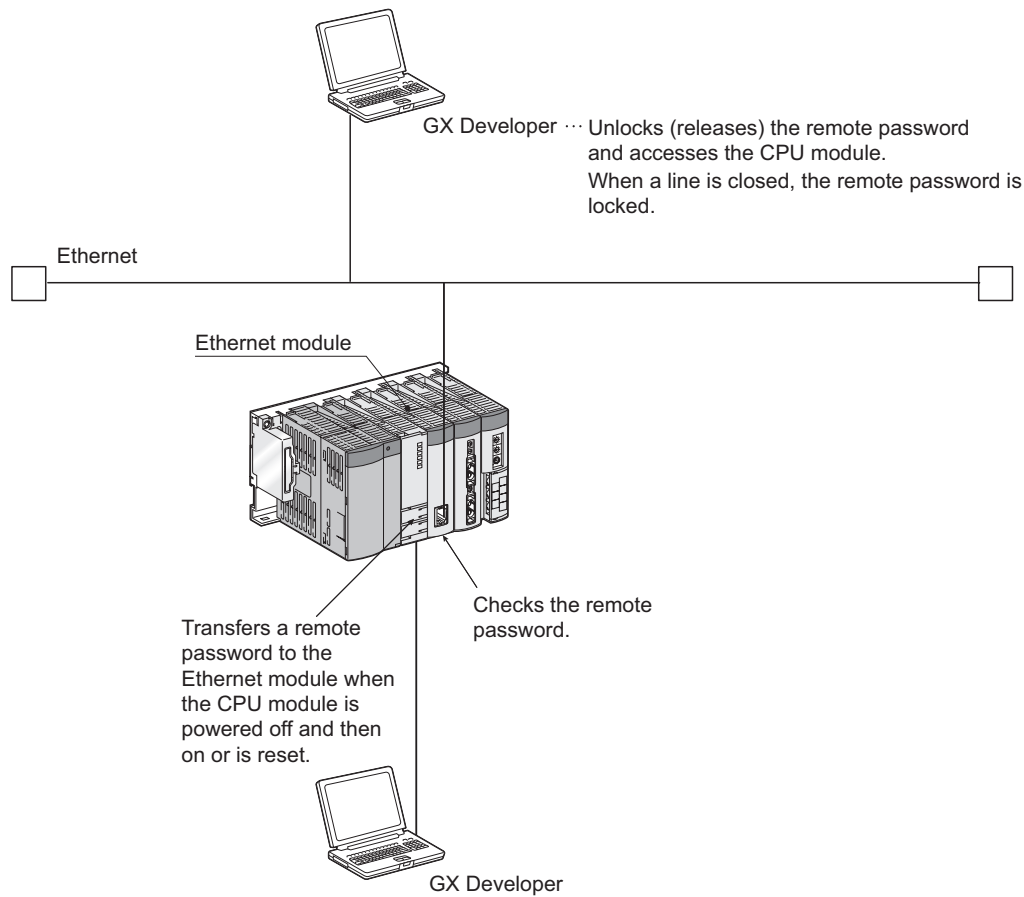


Figure 6.81 Outline of locking/unlocking a remote password with an Ethernet module

(5) Procedures for setting/changing/clearing a remote password

(a) Setting a remote password

- In the project data list of GX Developer, select [Parameter] → [Remote pass].

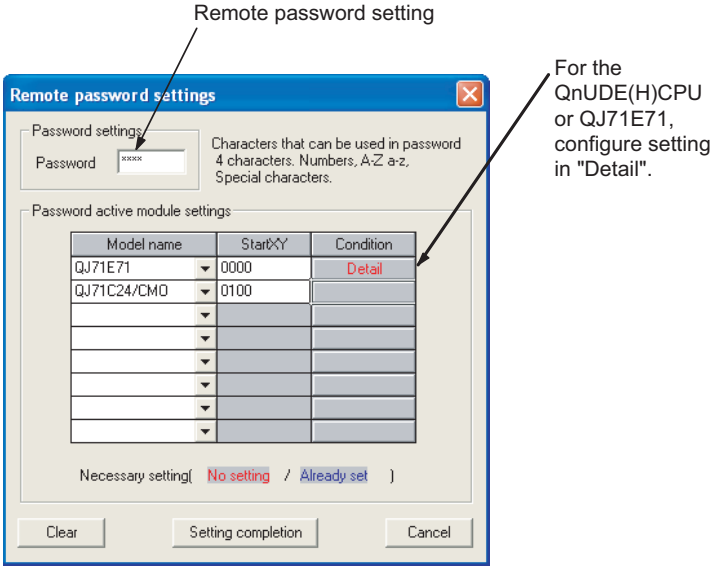


Figure 6.82 Remote password settings screen

Table6.33 Setting items on the Remote password settings screen

Item		Description	Setting range/option
Password settings		Enter a remote password.	Four characters or less (alphanumeric characters, special symbols)
Password active module settings	Model name	Select a model name.	• QnUDE(H)CPU • QJ71E71 • QJ71C24/CMO
	Start X/Y	Set the start address of the module.	0000 _H to 0FE0 _H
Detail		-	-
User connection No.		Select user connection No.	Connection 1 to Connection 16
System connection	Auto open UDP port	Select a valid port of the remote password.	-
	FTP transmission port (TCP/IP)		
	GX Developer transmission port (TCP/IP)		
	GX Developer transmission port (UDP/IP)		
	HTTP port		

- Connect GX Developer to the CPU module.
- Write a set remote password to the CPU module.
- In multiple CPU systems, write a remote password to the control CPU of the module to which the remote password is to be set.

After setting a remote password, store the parameters to the valid drive ( Section 5.1.10).

(b) Changing a remote password

Change set password in the Remote password settings screen and write a new password to the CPU module.

(c) Clearing a remote password

- To delete set remote password, click the button in the Remote password settings screen.
- Write a remote password with GX Developer.

6.20 System Display of CPU Module with GX Developer

When the CPU module is connected to GX Developer, this function can check the following items of the modules on the base unit in the System Monitor screen.

- Installed status
- Parameter status
- Module's Detailed Information
- Product information

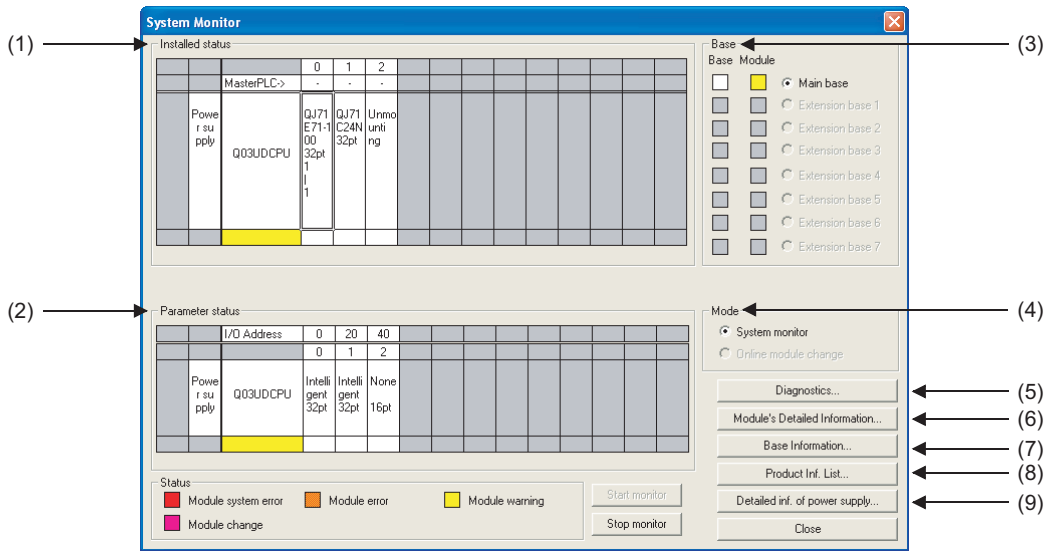


Figure 6.83 System Monitor screen

(1) Installed status

The control CPU, model name, and the number of points of the modules on the selected base unit can be checked.

"Unmounting" is displayed in the field of a number of slot where a module is not mounted.

The model name of a module on a slot to which "Empty" is set in the I/O assignment tab of the PLC parameter dialog box is not displayed.

When using a redundant base unit, mounted status of the power supply module is also displayed.

Note6.11

(2) Parameter status

The I/O number, type, and the number of points of the module on each slot of the selected base unit can be displayed.

If the Parameter status displays "0" for an empty slot or assignment error, it indicates that the I/O assignment setting of the PLC parameter dialog box and the mounted status differs. Therefore, change the I/O assignment setting so that it may match with the mounted status.

When using a redundant base unit, mounted status of the power supply module is also displayed. Note6.11



Note6.11

Universal

Before monitoring the Q02UCPU, Q03UDCPU, Q04UDHCPU, Q06UDHCPU, Q13UDHCPU, or Q26UDHCPU, check the versions of the CPU module and GX Developer. (Appendix 2)

(3) Base

The status of the base unit and modules on the base unit can be checked.

When there is even one faulty module, the "Module" field color changes according to the status described at the bottom of the screen.

(4) Mode

The mode cannot be selected since modules cannot be replaced online.

(5) Diagnostics

Click this button to check an error and status of the selected module.

(6) Module's Detailed Information

Click this button to check the details of the selected module.

For details of the intelligent function module, refer to the manual for the intelligent function module used.

(7) Base Information

The "Overall Information" and "Base Information" can be checked.

(a) Overall Information

The number of base units and the number of modules on the base units can be checked.

(b) Base Information

The name, the number of slots, type, and the number of modules of the selected base unit can be checked.

(8) Product Inf. List

Individual information (Type, Series, Model name, Points, I/O No., Master PLC, Serial No., Ver., and Product No.) of the mounted CPU module, I/O modules, and intelligent function module can be checked.

[illegible]

Figure 6.84 Product Information List screen

(9) Detailed information of power supply module

This screen displays "ON/OFF status", "Error existence", and "Number of momentary power failures" of the power supply module.
This screen can be displayed when using the power supply module supporting a redundant base unit and this screen.

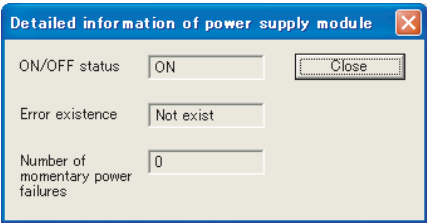


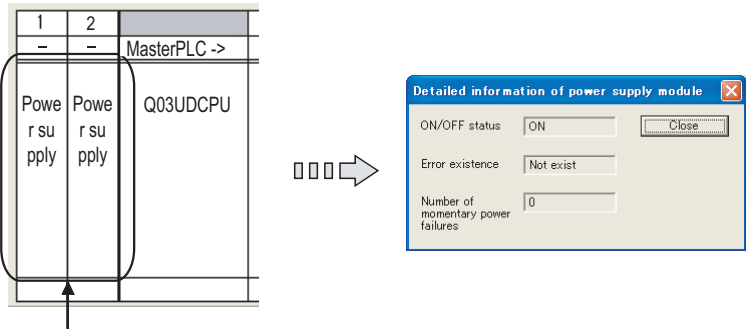
Figure 6.85 Detailed information of power supply module screen

Table6.34 Description of the Detailed information of power supply module screen

Item	Description
ON/OFF status	Displays the status of an input power supply to the redundant power supply module.
Error existence	Displays whether a failure (error) occurs in the redundant power supply module.
Number of momentary power failures	- When the redundant power supply module of redundant main base unit is selected Displays the number of momentary power failures of the redundant power supply module on the redundant main base unit (display range: 0 to 65535). - When the redundant power supply module of redundant extension base unit is selected Displays "-" and the number of momentary power failures is not counted.

Point

- In multiple CPU systems, the Detailed information of power supply module screen can be displayed:
 - when GX Developer is connected to CPU No. 1, or
 - when serial numbers (first five digits) of all CPU modules are "07032" or later.
- Double-clicking the area of power supply module in "Installed status" can also display the screen.



Double-clicking this area displays the Detailed information of power supply module screen.

Remark

For details of the System Monitor screen of GX Developer, refer to the following.
GX Developer Version 8 Operating Manual

6.21 LED Indication

Operating status of the CPU module can be checked by the LEDs on the front of the CPU module.
For details of LED indications, refer to the following.

 QCPU User's Manual (Hardware Design, Maintenance and Inspection)

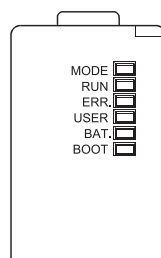


Figure 6.86 LEDs on the front of the CPU module

6.21.1 Methods for turning off the LEDs

(1) Methods

The LEDs can be turned off by the following operations (except for reset operation).

Table 6.35 Methods for turning off the LEDs

Method for turning off the LED	Relevant LED			
	ERR.	USER	BAT.	BOOT
Execute the LEDR instruction after resolving the error.	○	○	○	×
After resolving the error, clear the error by the special relay SM50 and special register SD50 ^{*1} (operation continuation error only).	○	○	○	×
Turn off the LED by the special relay SM202 and special register SD202. ^{*1}	×	○	×	○

○ : Valid, × : Invalid

*1: Description of special relays and special registers

- SM50 : Clears an error of the error code stored in SD50 when the CPU module is powered off and then on.
- SD50 : Stores an code of an error to be cleared.
: For details of error codes, refer to the following.
 QCPU User's Manual (Hardware Design, Maintenance and Inspection)
- SM202 : Turns off the LED corresponding to each bit of SD202 when the CPU module is powered off and then on.
- SD202 : Set an LED to be turned off.

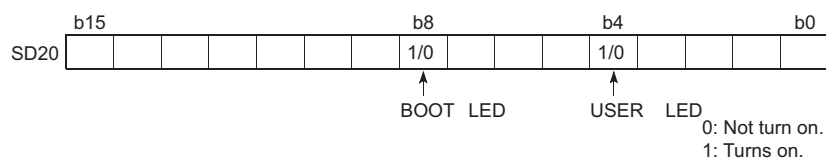


Figure 6.87 Bit structure of the special register SD202

Configure setting to turn off each LED as follows:

- Turning off both the Boot LED and USER LED: SD202 = 110_H
- Turning off only the BOOT LED: SD202 = 100_H
- Turning off only the USER LED: SD202 = 10_H

(2) Methods for not turning on the ERR. LED, USER LED, and BAT. LED

There is a priority in indications of the ERR.LED, USER LED, and BAT.LED. (☞ Section 6.21.2)
When an cause number of an LED is deleted in the priority, the LED will not turn on even if an error with the cause number occurs.

6.21.2 LED indication priority

This section describes a priority for error messages stored in the LED display data (SD220 to SD227) in case of an error.

(1) Displayed error messages and their priorities

- In case of multiple errors, the error messages are displayed with the following conditions.
- A stop error is always set to the LED display data (SD220 to SD227).
 - An operation continuation error is displayed according to the priority cause number described in this section.
Whether to indicate an error according to its priority using LED can be selected. (Set the priority using special registers, SD207 to SD209.)
 - When errors having the same priority occur simultaneously, the error detected first is displayed.

The priority is determined with the special registers SD207 to SD209 as follows.

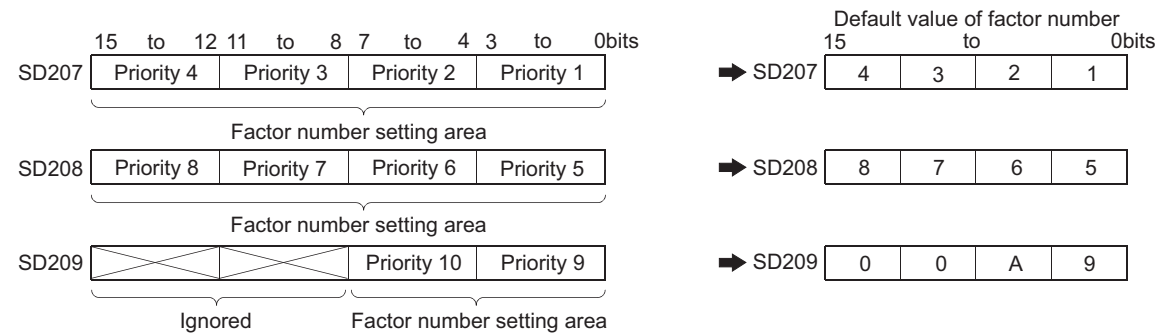


Figure 6.88 Special registers and bit structure ragarding a priority

(2) Priorities and cause numbers

The following table shows the description and priority of the cause numbers set to the special registers SD207 to SD209.

Table6.36 List of cause numbers and priorities

Priority	Cause number (hexadecimal)	Displayed error message	Remarks
1	1	• AC/DC DOWN	• Power-off
2	2	• UNIT VERIFY ERR. • FUSE BREAK OFF • SP.UNIT ERROR • SP.UNIT DOWN	• I/O module verification error • Fuse blown • Intelligent function module verification error
3	3	• OPERATION ERROR • SFCP OPE.ERROR • SFCP EXE.ERROR	• Operation error • SFC instruction operation error • SFC program execution error
4	4	• ICM.OPE.ERROR* ¹ • FILE OPE.ERROR • FLASH ROM ERROR	• Memory card operation error • File access error • Flash ROM access count over error
5	5	• PRG.TIME OVER • MULTI CPU ERROR* ²	• Constant scan setting time-out error • Another CPU error in multiple CPU systems
6	6	-	—
7	7	• Annunciator	—
8	8	-	—
9	9	• BATTERY ERROR	—
10	A	-	—

*1: The Q00UJCPU, Q00UCPU, and Q01UCPU cannot display the error message.

*2: The Q00UJCPU cannot display the error message.

Point

- To remain the LED off even in case of an error, set the cause number setting area (each 4 bits) of SD207 to SD209 that stores the corresponding cause number to "0".

Example To remain the ERR. LED off even when a fuse blown error is detected, set the cause number setting area where the cause number "2" is stored to "0".

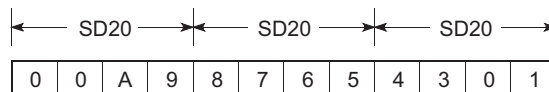


Figure 6.89 Cause numbers stored in SD207 to SD209

Because the cause number "2" is not set, the ERR.LED remains off even if a fuse blown is detected.

In this case, even if another error with the cause number "2" (I/O module verification error or intelligent function module verification error) is detected, the ERR.LED remains off.

- If "0" is set to the cause number setting area (setting that does not turn on the LED), SM0 (Diagnostic errors) and SM1 (Self-diagnostic error) turn on, and the error code is stored to SD0 (Diagnostic errors).

6.22 Interrupt from Intelligent Function Module

The CPU module can execute an interrupt program (I □) by the interrupt request from the intelligent function module. For example, the serial communication module can receive data by an interrupt program when the following data communication functions are executed.

- Data reception during communication by nonprocedural protocol
- Data reception during communication by bidirectional protocol

Using an interrupt program enables a CPU module to receive data quickly.

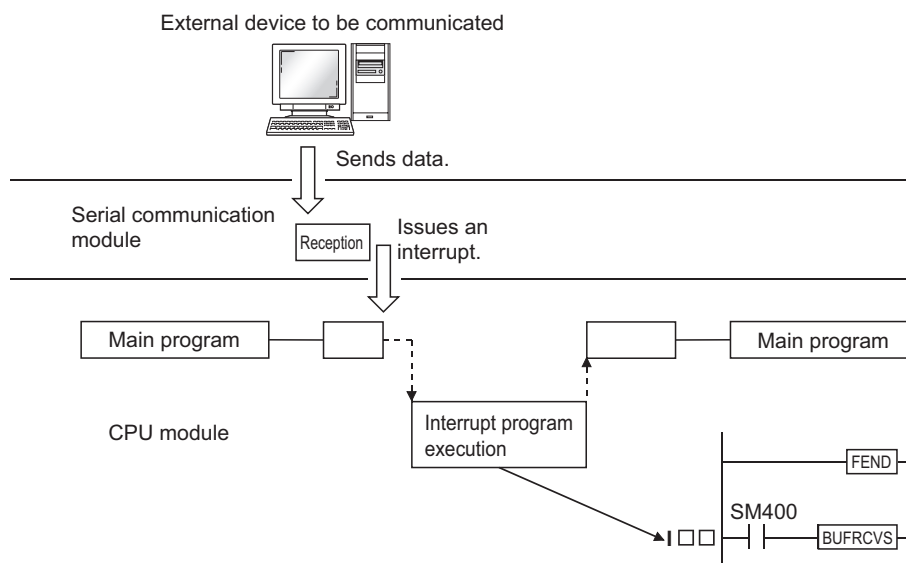


Figure 6.90 Interrupt from a serial communication module

(1) Setting an interrupt from the intelligent function module

To execute an interrupt program by an interrupt from the intelligent function module, select "Interrupt pointer setting" in "Intelligent function module setting" of the PLC system tab in the PLC parameter dialog box. Configure system setting at the intelligent function module is also required.

For execution of an interrupt program by an interrupt from the intelligent function module, refer to the following.

☞ Manual for the intelligent function module used

Remark

For the numbers of interrupt pointers available for an interrupt from the intelligent function module, refer to Section 9.11.

(1) Definition

This function communicates in the MC protocol*¹ by connecting the RS-232 interface of the CPU module, personal computer, and HMI by RS-232 cable.

This section describes the specifications, functions, and various settings of the function.

*1: The MC protocol is an abbreviation for the MELSEC communication protocol.

The MELSEC communication protocol is a communication method to access from an external device to the CPU module according to the communication procedure for the Q series programmable controller (such as a serial communication module, Ethernet module).

For the MC protocol, refer to the following.

 MELSEC-Q/L MELSEC Communication Protocol Reference Manual

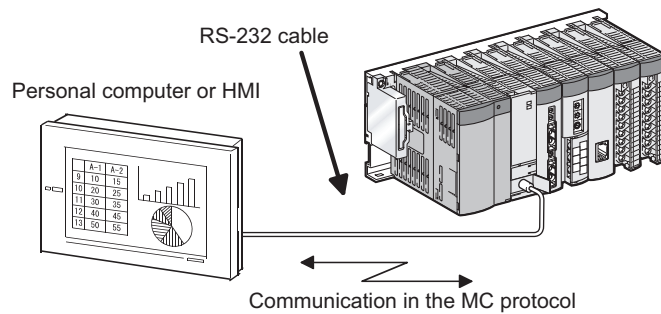


Figure 6.91 Communication with personal computer or HMI

Point

- A personal computer or HMI can communicate with a CPU module by the serial communication function only when the CPU module is connected to it.
The CC-Link IE, MELSECNET/H, Ethernet, or CC-Link cannot be communicated with another station.
- The serial communication function is not used for connection of GX Developer or GX Configurator with the CPU module.



Note6.12

Universal

When using the serial communication function for the Q02UCPU, check the versions of the CPU module and GX Developer. ( Appendix 2)

When using the Q03UDCPU, Q04UDHCPU, Q06UDHCPU, Q10UDHCPU, Q13UDHCPU, Q20UDHCPU, Q26UDHCPU, or Built-in Ethernet port QCPU, the serial communication function cannot be used.

(2) Specifications

(a) Transmission specifications

Table6.37 shows the transmission specifications of RS-232 for the serial communication function of the CPU module.

Check that the specifications of the personal computer and HMI match those of Table6.37 before using the function.

Table6.37 Transmission specifications of the serial communication function

Item	Default	Setting range
Communication method	Full-duplex communication	-
Synchronization method	Asynchronous method	-
Transmission speed ^{*1}	19.2kbps	9.6kbps, 19.2kbps, 38.4kbps, 57.6kbps, 115.2kbps
Data format	• Start bit: 1 • Data bit: 8 • Parity bit: Odd • Stop bit: 1	-
MC protocol format ^{*2} (automatic detection)	• Format 4 (ASCII) • Format 5 (binary)	-
Frame ^{*2}	• QnA-compatible 3C frame • QnA-compatible 4C frame	-
Transmission control	DTR/DSR control	-
Sum check ^{*1}	Checked	Checked/not checked
Transmission wait time ^{*1}	No waiting time	No waiting time, 10ms to 150ms (in increments of 10ms)
RUN write setting ^{*1}	Prohibited (deselected)	Permit (checked), prohibited (deselected)
Overall cable distance	15m	-

*1: Can be set in the PLC parameter dialog box.

*2: Table6.38 shows the relationship between the MC protocol formats and frames.

Table6.38 Relationship between the MC protocol formats and frames

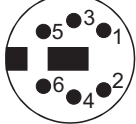
Function		Format 4	Format 5
Communication in ASCII code	QnA-compatible 3C frame	○	×
	QnA-compatible 4C frame	○	×
Communication in binary code	QnA-compatible 4C frame	×	○

○ : Available, × : Unavailable

(b) RS-232 connector specifications

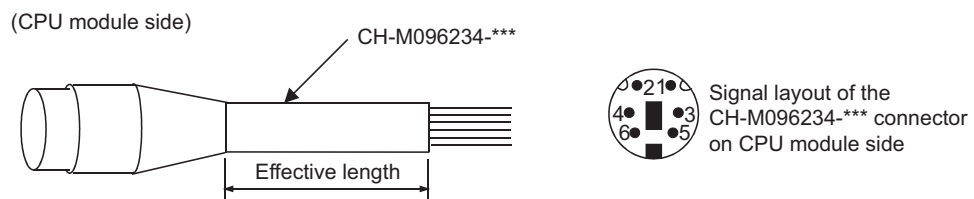
Table6.39 shows the specifications of the RS-232 connector for the CPU module.

Table6.39 RS-232 connector specifications

Appearance	Pin number	Signal	Signal name
 Mini-DIN 6 pins (female)	1	RD(RXD)	Receive data
	2	SD(TXD)	Send data
	3	SG	Signal ground
	4	-	-
	5	DSR(DR)	Data setting ready
	6	DTR(ER)	Data terminal ready

(c) RS-232 cable

- The following RS-232 cable can be used for connecting the CPU module to the personal computer or HMI.
- QC30R2 (cable length: 3m)
- CH-M096234-*** (manufactured by CHUGAI Co., Ltd.)
- Cable with a Mini-DIN connector on one side and without connector on the other side
- *** indicates a cable length, which can be lengthened up to 15m in units of 0.1m.



Pin number	1	2	3	4	5	6	Metal shell
Signal	RD	SD	SG	-	DR	ER	
Wire core	Red	Black	Green/ white	-	Yellow	Brown	Shield

Figure 6.92 Effective length and signal layout of RS-232 cable

(3) Functions

Table6.40 shows the MC protocol commands that can be executed by the serial communication function.

Table6.40 MC protocol commands supported by the serial communication function

Function			Command	Processing	Number of processing points
Device memory	Batch read	In units of bits	0401(00 □ 1)	Reads bit devices in units of 1 point.	ASCII: 3584 points BIN: 7168 points
		In units of words	0401(00 □ 0)	Reads bit devices in units of 16 points. Reads word devices in units of 1 point.	480 words (7680 points) 480 points
	Batch write ^{*1}	In units of bits	1401(00 □ 1)	Writes bit devices in units of 1 point.	ASCII: 3584 points BIN: 7168 points
		In units of words	1401(00 □ 0)	Writes bit devices in units of 16 points. Writes word devices in units of 1 point.	480 words (7680 points) 480 points
	Random read	In units of words	0403(00 □ 0)	Reads bit devices in units of 16 points or 32 points by specifying the device or device number at random.	96 points
				Reads word devices in units of 1 point or 2 points by specifying the device or device number at random.	
	Test ^{*1} (random write)	In units of bits	1402(00 □ 1)	Sets/resets bit devices in units of 1 point by specifying the device or device number at random.	94 points
		In units of words	1402(00 □ 0)	Sets/resets bit devices in units of 16 points or 32 points by specifying the device or device number at random.	*2
				Writes word devices in units of 1 point or 2 points by specifying the device or device number at random.	
	Monitor registration	In units of words	0801(00 □ 0)	Registers bit devices to be monitored in units of 16 points or 32 points.	96 points
				Registers word devices to be monitored in units of 1 point or 2 points.	96 points
	Monitor	In units of words	0802(00 □ 0)	Monitors devices registered for monitoring.	Number of monitor registration points

*1: When writing data while the CPU module is in the RUN status, set "RUN write setting" to "Permit".

*2: Set the number of processing points within the range of the following calculation formula.

$$(\text{number of word access points}) \times 12 + (\text{number of double word access points}) \times 14 \leq 960$$

- One point of a bit device corresponds to 16 bits for word access or to 32 bits for double word access.
- One point of a word device corresponds to one word for word access or to two words for double word access.

(4) Accessible devices

Table6.41 shows accessible devices by the serial communication function.

Table6.41 Accessible devices by the serial communication function

Category	Device		Device code	Device number range*1 (default value)		Write	Read	
Internal system device	Function input		FX ^{*2}	000000 to 00000F	Hexadecimal	×	×	
	Function output		FY ^{*2}	000000 to 00000F	Hexadecimal			
	Function register		FD	000000 to 000004	Decimal	○	○	
	Special relay		SM	000000 to 002047	Decimal			
	Special register		SD	000000 to 002047	Decimal			
Internal user device	Input		X	000000 to 001FFF	Hexadecimal			
	Output		Y	000000 to 001FFF	Hexadecimal			
	Internal relay		M	000000 to 008191	Decimal			
	Latch relay		L	000000 to 008191	Decimal			
	Annunciator		F	000000 to 002047	Decimal			
	Edge relay		V	000000 to 002047	Decimal			
	Link relay		B	000000 to 001FFF	Hexadecimal			
	Data register		D	000000 to 012287	Decimal			
	Link register		W	000000 to 001FFF	Hexadecimal			
	Timer	Contact	TS	000000 to 002047	Decimal			
		Coil	TC					
		Current value	TN					
	Retentive timer	Contact	SS	000000 to 002047	Decimal			
		Coil	SC					
		Current value	SN					
	Counter	Contact	CS	000000 to 001023	Decimal			
		Coil	CC					
		Current value	CN					
	Link special relay		SB	000000 to 0007FF	Hexadecimal			×
	Link special register		SW	000000 to 0007FF	Hexadecimal			
	Step relay		S	000000 to 008191	Decimal	○		
	Direct input		DX	000000 to 000FFF	Hexadecimal			
	Direct output		DY	000000 to 000FFF	Hexadecimal			
Index register	Index register	Z	000000 to 000019	Decimal				
File register ^{*3}	File register	R	000000 to 032767	Decimal				
		ZR	000000 to 3FD7FF	Hexadecimal				
Extended data register ^{*3}	Extended data register	D	000000 to 093183	Decimal	○			
Extended link register ^{*3}	Extended link register	W	000000 to 016BFF	Hexadecimal				

○ : Reading/writing are enabled. × : Writing is prohibited.

*1: After changing the number of device points of the CPU module with GX Developer, use the devices within the new device number range.
Decimal and hexadecimal indicate whether a command specified for the MC protocol is decimal or hexadecimal.

*2: From 000005 to 00000F of FX and FY will be indefinite value.

*3: These registers are not supported by the Q00UJCPU.

(5) Setting of transmission specifications

Set Transmission speed, Sum check, Transmission wait time, and Run write setting of the serial communication function in the Serial tab of the PLC parameter dialog box.

- Select "Use serial communication" in communication with the personal computer or HMI.
- Set Transmission speed, Sum check, Transmission wait time, and Run write setting in the tab.

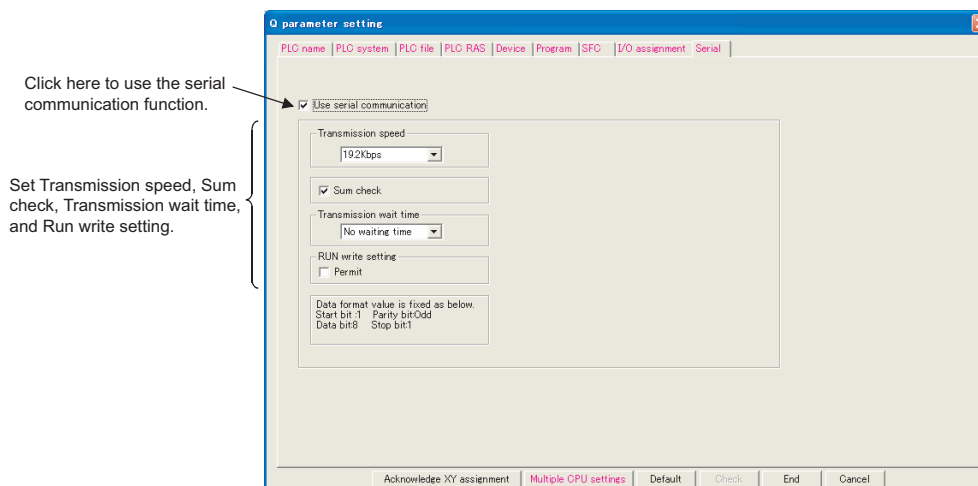


Figure 6.93 Serial tab

(6) Precautions

(a) Switching connection to GX Developer during communication with such as HMI

The CPU module can switch connection to GX Developer during communication with the personal computer or HMI with the serial communication function.

However, the personal computer or HMI in communication by the serial communication function results in a communication error.

For startup methods of the personal computer and HMI when they are reconnected to the CPU module, refer to the manual for used device.

(b) Transmission speed set in the Transfer Setup screen

When "Use serial communication" is selected, the transmission speed set in the Transfer setup screen of GX Developer is ignored.

Point

The data set in the "Serial" tab is valid when:

- the CPU module is powered on, or
- the CPU module is reset.

(c) Communication error

If any of the following status is met, responses are not returned and therefore communication cannot be made.
Review the transmission frame.

- 1) The serial communication function is set not to be used.
- 2) Communication is made at different transmission speed and data format.
- 3) A frame to be sent has no correct starting end or terminal.
 - 3C frame format 4: ENQ/CR + LF
 - 4C frame format 4: ENQ/CR + LF
 - 4C frame format 5*¹: DLE+STX/DLE + ETX

*1: When "Sumcheck enable" is set, the sumcheck code is included.

- 4) The frame identification number of a frame to be sent is incorrect.
- 5) The number of transmission bytes is under the header part size.

(7) Error codes during communication with the serial communication function

Table6.42 shows the error codes, error description, and corrective actions sent from the CPU module to the external device when an error occurs during communication with the serial communication function.

Table6.42 Error codes sent from the CPU module to external device

Error code (hexadecimal)	Error item	Description	Corrective action
4000H to 4FFFH	-	Error detected by the CPU module (error occurred by other than the serial communication function)	Refer to the QCPU User's Manual (Hardware Design, Maintenance and Inspection), and take corrective action.
7155H	Unregistered monitor error	A monitor request was given before monitor registration.	Give a monitor request after registering a device to be monitored.
7157H	Request target specification error	Other than the Q00UJCPU, Q00UCPU, Q01UCPU, and Q02UCPU was selected as a request module and route.	Check that the transmission message address is Q00UJCPU, Q00UCPU, Q01UCPU, Q02UCPU, modify the setting and communicate again.
7164H	Request data error	The requested data or device specification method is wrong.	Check the sent message/requested data of the external device, correct it, and restart communication.
7167H	Disabled during RUN	A write command was specified while online change is disabled.	Enable the online change and restart communication.
7E40H	Command error	A subcommand or a command that does not exist is specified.	Check and correct the sent message of the external device and restart communication.
7E41H	Data length error	The number of points specified for random write/read exceeds the number of points enabled for communication.	Check and correct the sent message of the external device and restart communication.
7E42H	Data count error	The requested number of points exceeds the range of the command.	Check and correct the sent message of the external device and restart communication.
7E43H	Device error	The device specified does not exist. The device specified cannot be specified by the corresponding command.	Check and correct the sent message of the external device and restart communication.
7E47H	Continuous request error	The next request was received before the response message was returned.	Do not give continuous requests from the external device. Match the monitoring time of timer 1 with the time-out time of the external device.
7F21H	Receive header section error	The command (frame) section specified is in error. The ASCII code received cannot be converted into binary.	Check and correct the sent message of the external device and restart communication.
7F22H	Command error	The command or device specified does not exist.	Check and correct the sent message of the external device and restart communication.
7F23H	MC protocol message error	The data (such as ETX, CR+LF) specified after the character part does not exist or in error.	Check and correct the sent message of the external device and restart communication.
7F24H	Sumcheck error	The calculated sumcheck does not match the received sumcheck.	Review the sumcheck of external device.
7F67H	Overrun error	The next data was received before the CPU module completed receive processing.	Reduce the communication speed and restart communication. Check the CPU module for momentary power failure. (For the CPU module, use the special register SD53 to check.) When an momentary power failure occurs, remove its cause.
7F68H	Framing error	The stop bit setting does not match.	Match the setting of the CPU module with that of the external device.
7F69H	Parity error	The parity bit setting does not match.	Match the setting of the CPU module with that of the external device.

6.24 Service Processing

6.24.1 Service processing setting

(1) Definition

This function allows to set the time and the number of times of service processing performed at END processing by parameters.

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

This achieves the configuration of service processing environment optimum for the system.

Point

The service processing refers to communication service processing with a peripheral (such as GX Developer) and intelligent function modules. Link refresh processing such as CC-Link IE module, MELSECNET/H module, and CC-Link Safety system master/local module are not included.

Using the COM instruction enables service processing during program execution of same performance with service processing at END processing. Therefore, the high-speed service processing response can be performed even if the scan time is long.

(2) Parameter setting

Set the parameters in the PLC system tab of the PLC parameter dialog box.

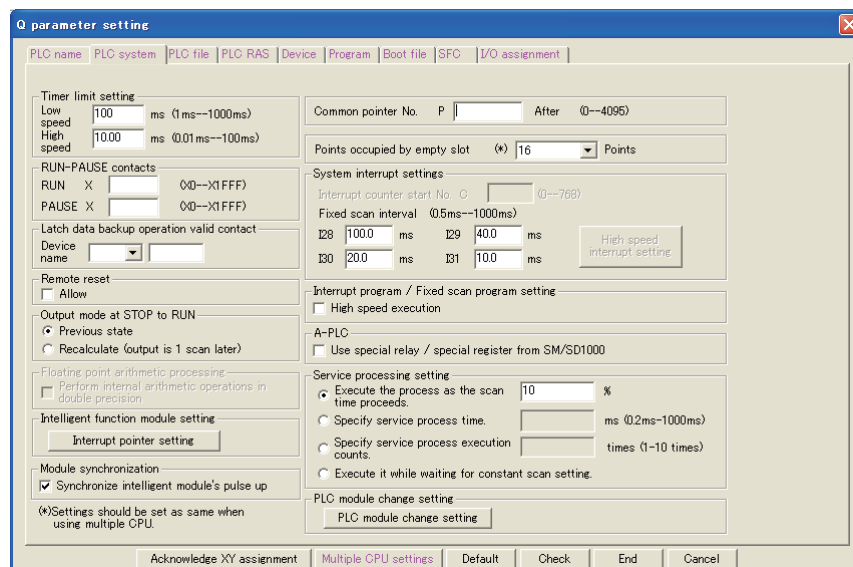


Figure 6.94 Parameter setting screen

To perform the service processing, select any of the parameter items in Table6.43.

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

Table6.43 Parameter items

Item	Description	Setting range	Remarks
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: 0.2ms to 1000ms - Unit: 0.1ms	Default when selected =0.2ms
Specify service process execution counts.	Set the number of 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 perform service processing during waiting time for constant scan setting.	-	Even when the waiting time is 0.2ms or less, the service processing time (0.2ms) will be added to the scan time at service processing execution.

(3) Operations for service processing setting

Operations for each service processing setting is described below.

(a) Operation when "Execute the process as the scan time proceeds." is selected

1) Operation when 10% is set

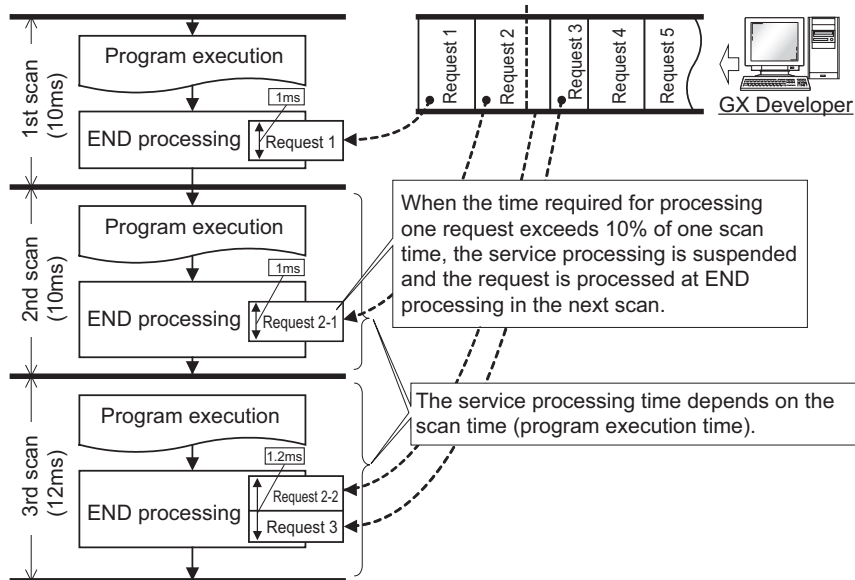


Figure 6.95 Operation when 10% is set

Point

If no request data for service processing exists, END processing speeds up by the request processing time. (The CPU module does not wait for requests.)

2) Operation for constant scan setting

The calculation of the service processing time is a calculation of the percentage of the time excluding the waiting time of the constant scan from the scan time, not a calculation of the percentage of the scan time.

Example Operation when 50% is set

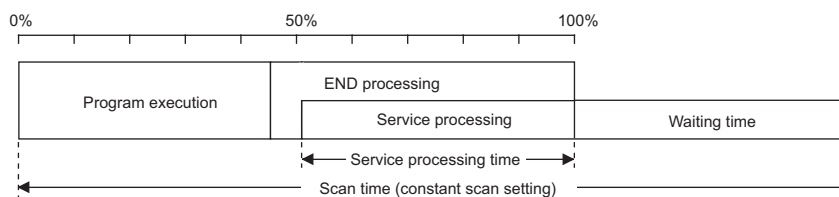


Figure 6.96 Operation for constant scan setting

Point

When setting the constant scan, selecting "Execute it while waiting for constant scan setting." can perform the service processing efficiently. (☞ (3)(d) in this section)

(b) Operation when "Specify service process execution counts." is selected

1) Operation when 1 time is set

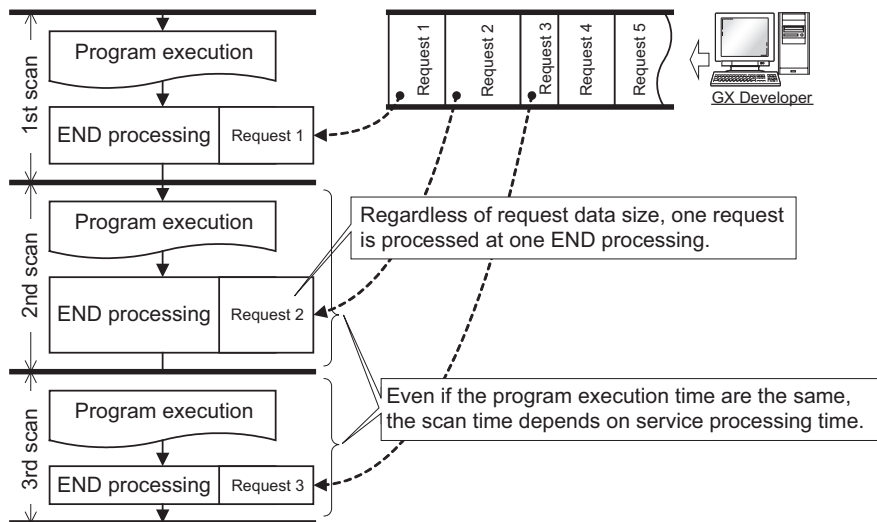


Figure 6.97 Operation when 1 time is set

2) Operation when 2 times is set

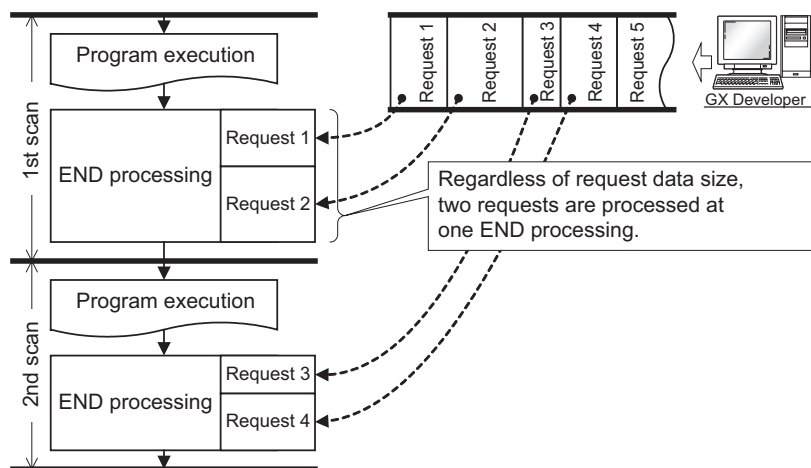


Figure 6.98 Operation when 2 times is set

Point

- When several devices are connected to one CPU module, each device requests service processing. When the CPU module receives requests from several devices simultaneously, a single END processing can accept several requests simultaneously if the service processing count is set to the number of connected devices. This improves response performance. (Note that the scan time increases by the service processing time.)
- If no request data exists when setting the service processing count, END processing speeds up by the request processing time. (The CPU module does not wait for requests.)

(c) Operation when "Specify service process time." is selected

1) Operation when 0.5ms is set

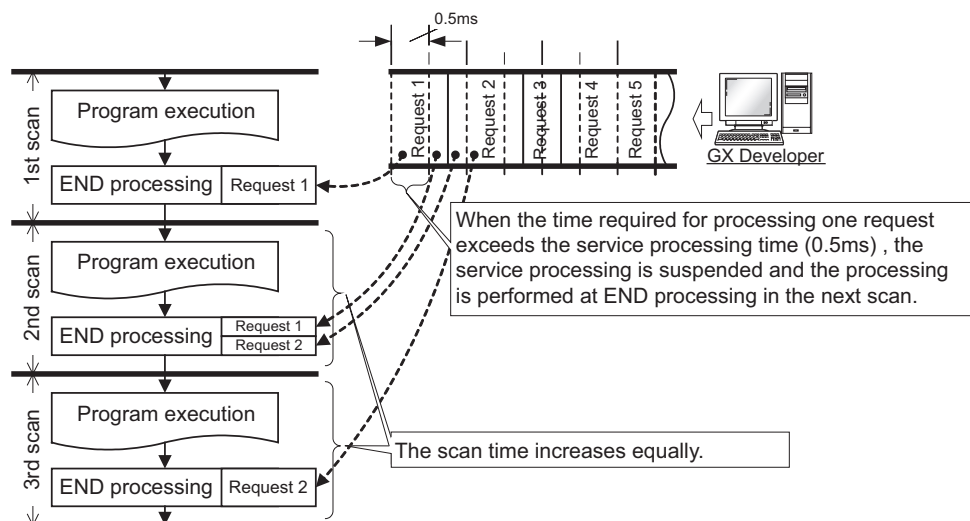


Figure 6.99 Operation when 0.5ms is set

2) Operation when 1ms is set

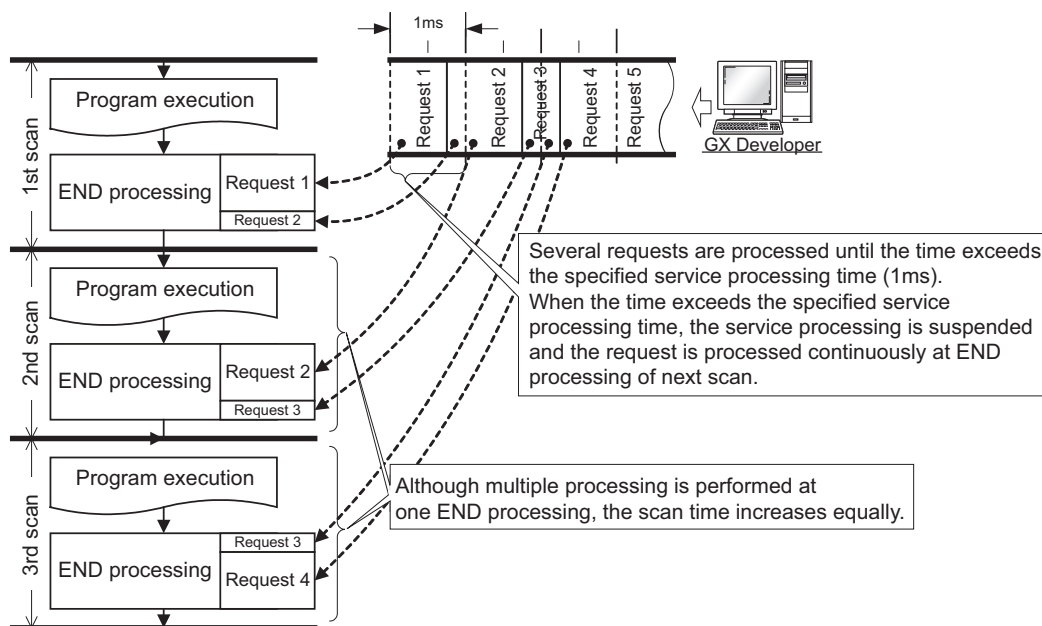


Figure 6.100 Operation when 1ms is set

Point

If no request data exists when setting the service processing time, END processing speeds up by the request processing time. (The CPU module does not wait for requests.)

(d) Operation when "Execute it while waiting for constant scan setting." is selected

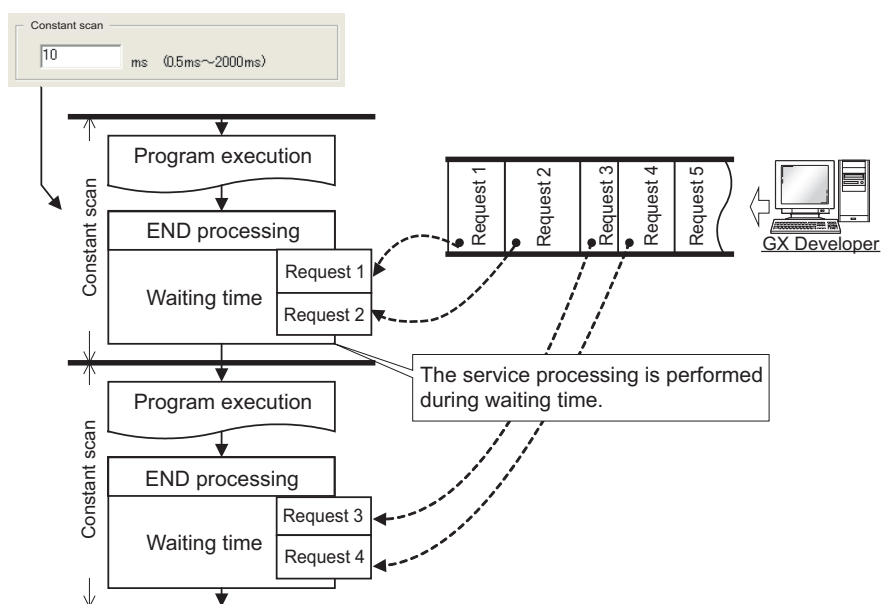
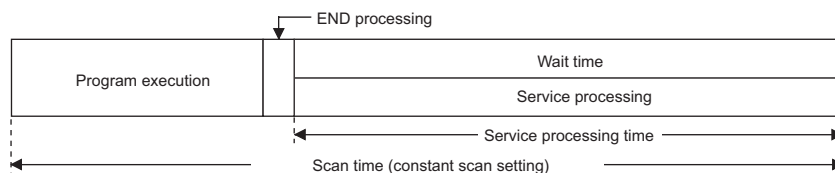


Figure 6.101 Operation when "Execute it while waiting for constant scan setting." is selected

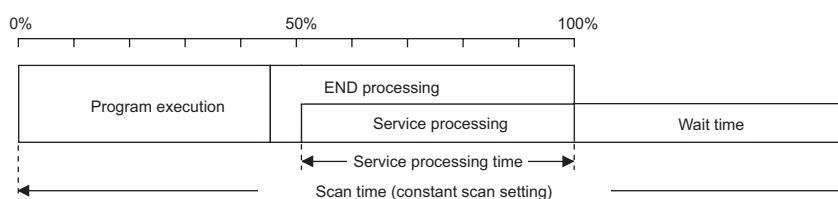
Point

- When setting the constant scan, selecting "Execute it while waiting for constant scan setting." can perform the service processing efficiently.

- When "Execute it while waiting for constant scan setting." is selected



- When "Execute the process as the scan time proceeds." is selected (50% is set.)



- Even when there is no waiting time, the service processing (0.2ms) is performed. Therefore, when the waiting time is less than 0.2ms, the constant scan time may be exceeded.

(4) Precautions

The following describes precautions when the service processing setting is configured.

- 1) For the following functions, scan time will be increased longer than the specified time during service processing even if the service processing time specification is set.
 - Online change
 - Change T/C setting
 - Local device monitor
 - Program memory backup
 - Writing to/reading from a file register (The scan time will be increased when the write or read size is large.)
 - Writing to/reading from the buffer memory of the intelligent function module (The scan time will be increased when the write or read size is large.)
 - Access to a network module
 - a) Diagnostic functions (CC IE Control diagnostics, CC IE Field diagnostics, MELSECNET diagnostics, Ethernet diagnostics, CC-Link/ CC-Link/LT diagnostics)
 - b) Monitor function (Module access device, Link direct device)
- 2) Note that the scan time will be increased much longer if the CPU module receives multiple requests simultaneously while the service processing count specification is set many.
- 3) The response performance of service processing significantly reduces in the following cases. Set service processing time considering the time-out time of the peripheral.
 - Service processing time is set much shorter than the scan time.
 - Setting "Execute it while waiting for constant scan setting" results in increase in the scan time and decrease in the service processing time.
- 4) An error of $-20\ \mu\text{s}$ to $+30\ \mu\text{s}$ occurs between the actual processing time and the set service processing time.

6.25 Initial Device Value

(1) Definition

This function registers data used in a program to the device or the buffer memory of the intelligent function module without a program.

(2) Application

Using an initial device value can omit device data setting program by initial processing program.

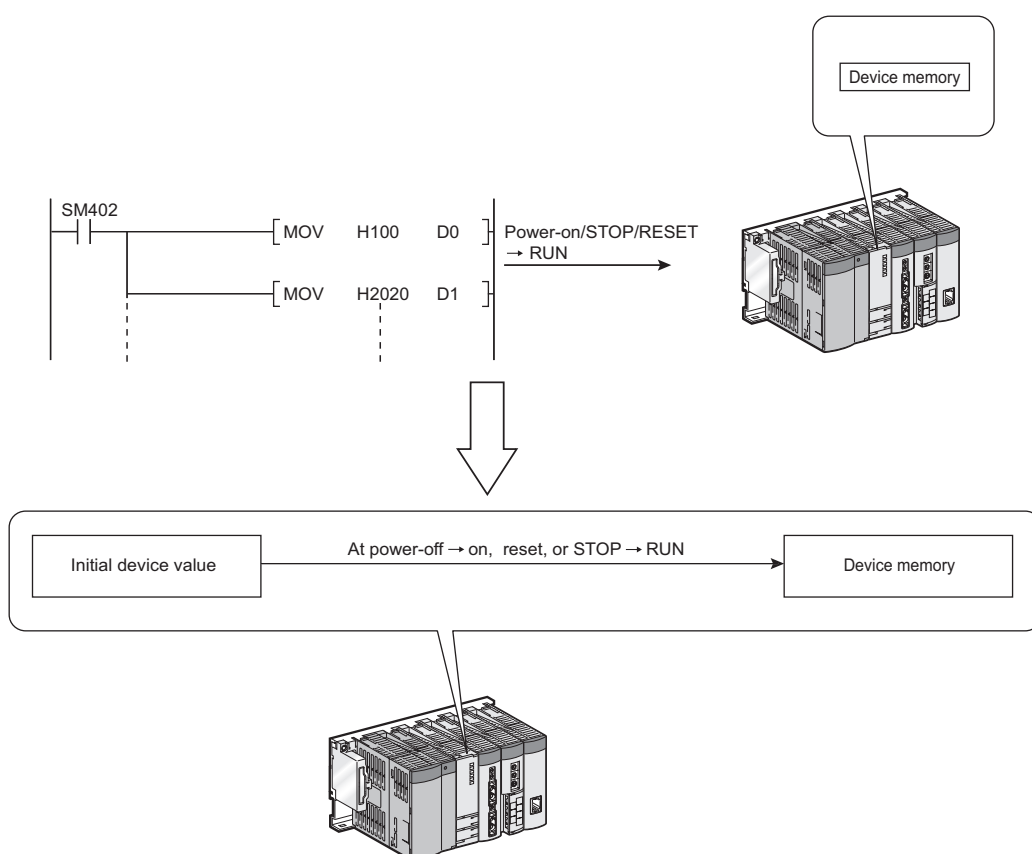


Figure 6.102 Data setting by initial program

(3) Timing when initial device values are written to the specified device

The CPU module writes data in the specified initial device value file to the specified device or the buffer memory of the intelligent function module when the CPU module is powered off and then on, is reset, or is set to the STOP status and then the RUN status.

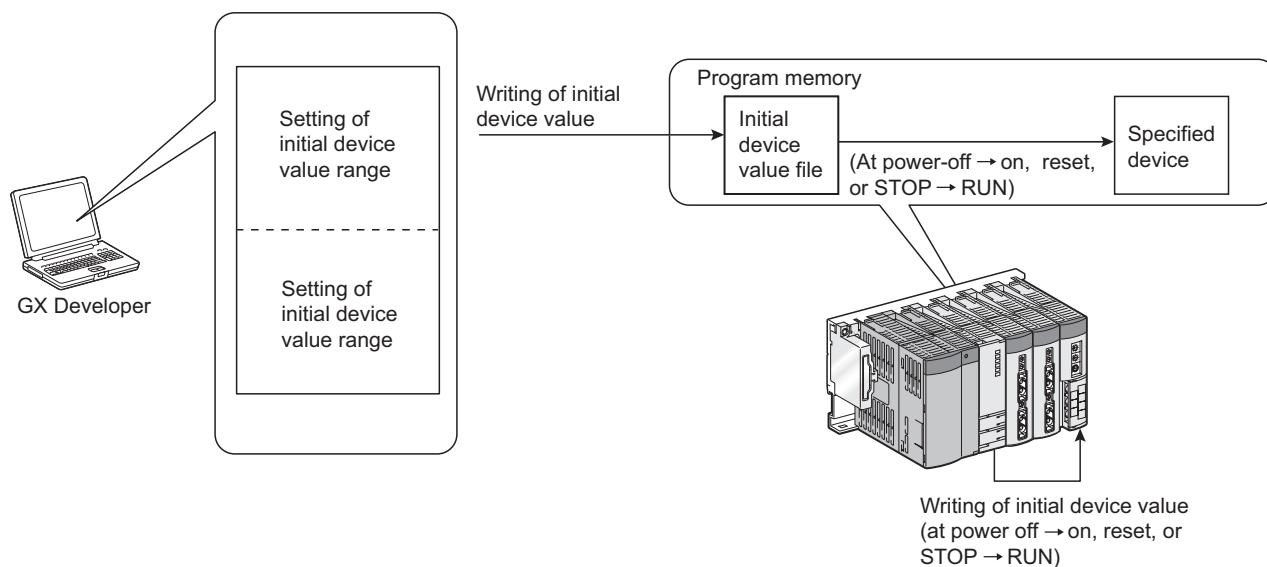


Figure 6.103 Flow of writing initial device value

(4) Devices that can be used^{*1}

The following shows devices that can be used for initial device value.

- Current timer value (T)
- Current retentive timer value (ST)
- Current counter value (C)
- Data register (D)^{*2}
- Special register (SD)
- Link register (W)^{*3}
- Link special register (SW)
- File register (R)
- File register (ZR)
- Intelligent function module device (U□\G□)
- Link direct device (J□\W□, J□\SW□)

^{*1}: For available ranges, refer to Section 9.1.

^{*2}: The extended data register (D) is included.

^{*3}: The extended link register (W) is included.

(5) Procedures and settings for using initial device values

To use initial device values, create initial device value data with GX Developer beforehand, and store the data as a initial device value file in the program memory, standard ROM, or memory card of the CPU module.

- Add an initial device value data to the project data list of GX Developer.
The Device initialization range setting screen appears. Set the initial device value range.
The number of settable points is up to 8000 points per range setting.
- Add device memory data in the project data list of GX Developer.
The device memory screen appears. Set the initial device value data within the initial device value range set above.

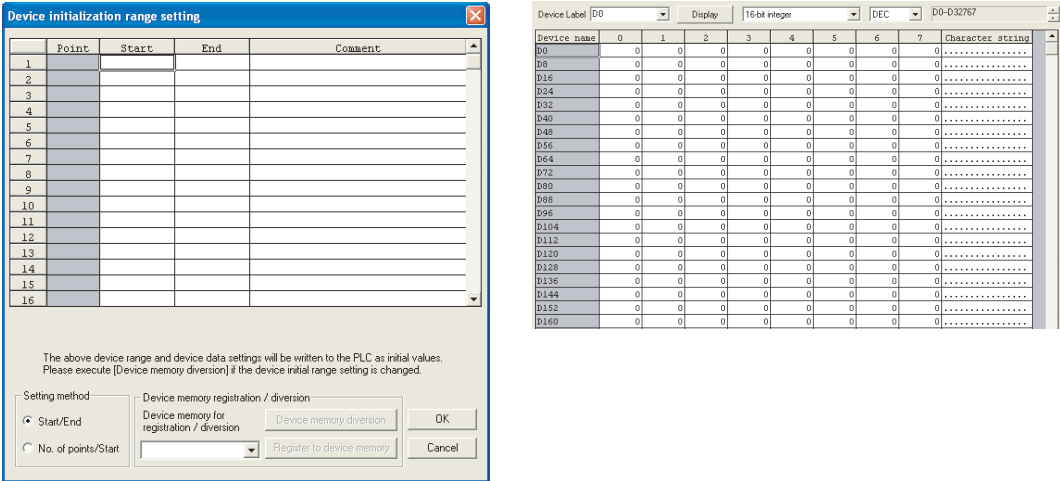


Figure 6.104 Device initialization range setting screen and device memory screen

Point

When changing the setting on the Device initialization range setting screen, always execute "Device memory registration/diversion".
For details of the "Device memory registration/diversion", refer to the following.

 GX Developer Version 8 Operating Manual

- Select the name of a file where the initial device value data are stored in the PLC file tab of the PLC parameter dialog box.

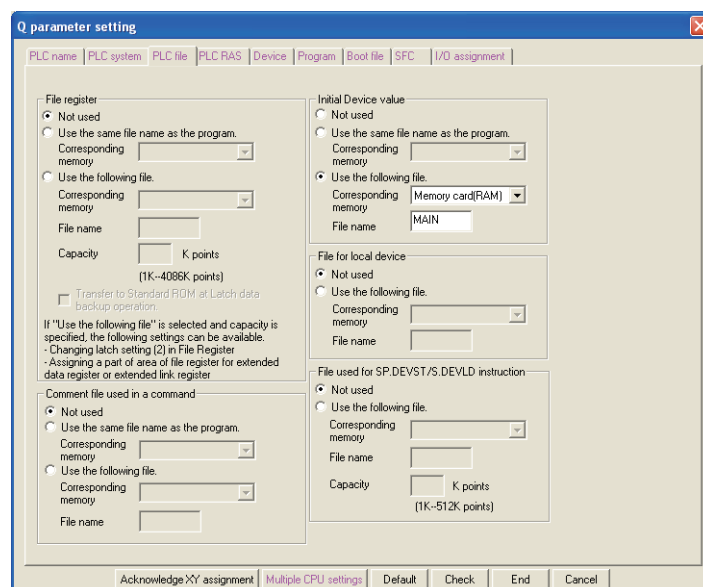


Figure 6.105 PLC file tab

- Write the set initial device value and parameters to the CPU module.

(6) Precautions

(a) When initial device value and latch range are overlapped

In that case, initial device value takes priority. Therefore, the latch range data will be overwritten to the initial device value data after the CPU module is powered off and then on.

(b) Area disabling the initial device value setting when the CPU module is set from STOP to RUN

The initial device value are also reflected when the CPU module is set from STOP to RUN.

For an area where an initial device value is not to be set when the CPU module is set from STOP to RUN (data that are set when the CPU module is powered off and then on and changed by a program), the initial device value cannot be used.

Use an instruction such as the MOV instruction in the main routine program so that the initial device values will be set to the specified devices.

Use the TO instruction to write data to the buffer memory of the intelligent function module.

(c) Devices that require module synchronization setting

When setting the following devices in the Device initialization range setting screen, set "Module synchronization" in the PLC system tab of the PLC parameter dialog box.

If the setting is not configured, the initial device values may not be set to the target module properly.

- Intelligent function module device (U□\G□)
- Link direct device (J□\W□, J□\SW□)

Remark

For details of the initial device value range setting, setting of the initial device value data, and writing of the initial device values to the CPU module, refer to the following.

GX Developer Version 8 Operating Manual

6.26 Battery Life-prolonging Function

(1) Definition

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

This function initializes all data other than the clock data when the CPU module is powered off or is reset.

Table6.44 Initialization details

Data held by a battery		Description
Error history		The number of error history data is initialized to zero.
Latch device (L)		Cleared to zero.
Device in the latch range		Cleared to zero.
Standard RAM		Formatted (cleared to zero).
File register assigned to the standard RAM	Set "Use the same file name as the program."	A file is deleted.
	Set "Use the following file".	A file is deleted (A file is recreated at power-on or reset (Data is cleared to zero..))

(2) Setting

Set the function in the I/O assignment tab of the PLC parameter dialog box.

- 1) Configure the I/O assignment setting.
- 2) Click the **Switch setting** button.
- 3) Enter 0001_H to the switch 3 of the slot where the CPU module is mounted (When entering the value to a slot in another station, the value is ignored).

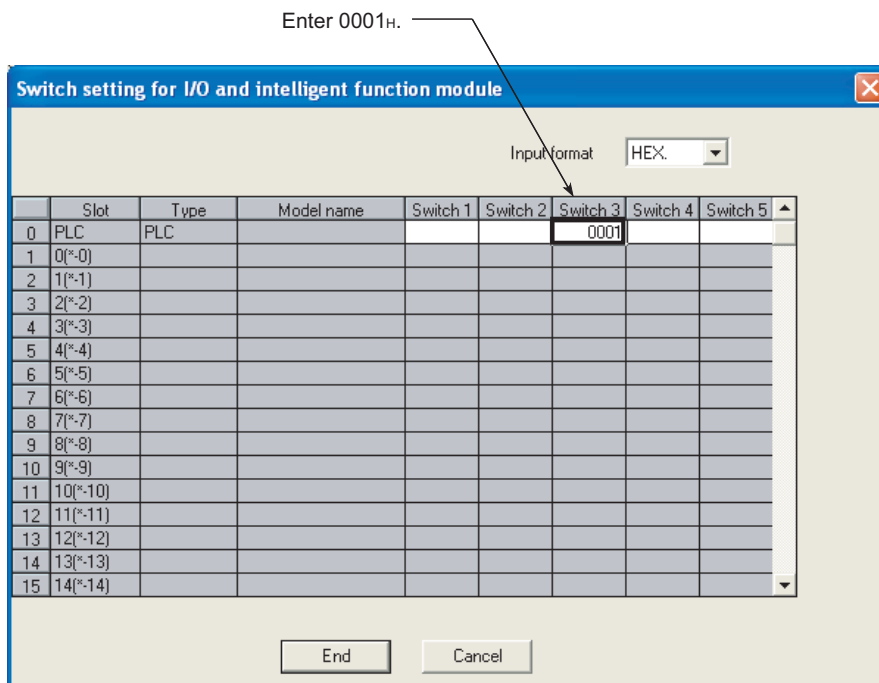


Figure 6.106 Switch setting screen

(3) Battery life

For the life of battery installed in the CPU module when the battery life-prolonging function is used, refer to the following.

☞ QCPU User's Manual (Hardware Design, Maintenance and Inspection)

6.27 Memory Check Function

This function checks whether data in the memories of the CPU module are not changed due to such as excessive electric noise.

Since the CPU module automatically checks a memory, setting for enabling this function is unnecessary.

This function does not require processing time.

(1) Data to be checked

(a) Program

The program during execution is compared with the user program written to the program memory.

If they do not match, a stop error, "RAM ERROR" (error code: 1160) is detected.

(b) Parameter

The parameters are compared with the ones written to the parameter-valid drive.

(c) Device memory

If the devices cannot be read, a stop error, "RAM ERROR" (error code: 1161) is detected.

(2) Execution timing

1) Program: At program execution

2) Parameter:

- When the CPU module is powered off and then on
- When the CPU module is reset
- When the CPU module is set from STOP to RUN after data are written to it

3) Device memory: When device data are read

6.28 Latch Data Backup to Standard ROM Function

(1) Definition

This function holds (backs up) latch data, such as device data and error history, to the standard ROM without using a battery when the system is stopped for a long period. This function helps to extend battery life.

Remark

When this function is performed, the battery life-prolonging function is enabled regardless of the parameter setting for the battery life-prolonging function.

Also, the battery life-prolonging function is switched to be disabled after backup data are restored.

The setting status of the battery life-prolonging function can be checked by SD119 (battery life-prolonging factor).

For details of the battery life-prolonging function, refer to Section 6.26.

(2) Data to be backed up

Table6.45 shows data to be backed up.

Table6.45 Data to be backed up

Data to be backed up	Description	Remarks
Device data	- Internal user device (M, L, B, F, V, T, ST, C, D, W) - Index register (Z)/standard device register (Z) - File register (R, ZR) - Extended data register (D)^{*1} - Extended link register (W)^{*1}	The data of the file register, extended data register (D), and extended link register (W) are backed up only when the file register in the standard RAM is used
Error history	Error history data immediately before the latch data backup function to standard ROM function is performed	-
Module error collection file	Information on errors occurred in the connected intelligent function modules	The data are backed up independent of the setting of the module error collection.
SFC program continuation start information	Information to start the SFC program continuously	-
Trace setting (Sampling trace file)	Trace condition settings and Trace data settings created by the sampling trace function.	Data cannot be backed up in the following cases. - Trace setting is made in a memory card. - Trace setting is not registered with the CPU module.

*1: The data are backed up only when the Universal model QCPU having the serial number (first five digits) is "10042" or later is used.

When backing up the data in the file register, extended data register (D), and extended link register (W), pay attention to the following.

- The data are backed up only when the file register in the standard RAM is set to be used.
- Select "Transfer to Standard ROM at Latch data backup operation." on the PLC file tab of the PLC parameter dialog box.

Selecting this option starts backup.

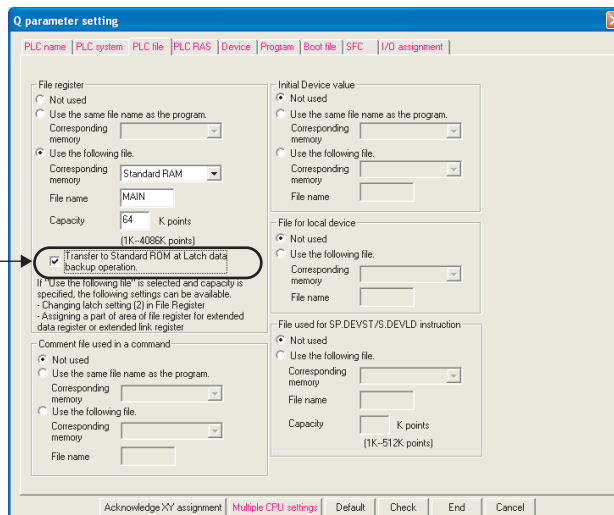


Figure 6.107 PLC file tab

(3) File size for storing backup data

The system automatically creates storage file for backup data when the latch data backup function to standard ROM is performed.

Table6.46 shows the size of storage file.

Table6.46 Sizes of files to be stored

Data to be backed up		File size
Device data	Internal user device (M, L, B, F, V, T, ST, C, D, W) Index register (Z)/standard device register (Z)	• Serial number (first five digits) is "11042" or earlier: 110890 bytes* ¹ (When the default device assignment is set)* ²
Error history		• Serial number (first five digits) is "11043" or later: 117310 bytes* ⁵ (When the default device assignment is set)* ²
SFC program continuation start information		
Module error collection file (stored in system memory.)		
Module error collection file (stored in standard RAM.)		
File register (R, ZR)* ³ , extended data register (D)* ³ , extended link register (W)* ³		64 + 2 × Number of file register points
ÉgTrace setting (sampling trace file)* ⁴		16 + Sampling trace file size

*1: The file size of the Q00UJCPU, Q00UCPU, Q01UCPU, and Q02UCPU is 87210 bytes.

*2: The size depends on the parameter setting.

*3: Storage files are created only when data in the file register, extended data register (D), and extended link register (W) are backed up.

*4: Storage files are created only when the trace registration has been made.

*5: The file size of the Q00UJCPU, Q00UCPU, and Q01UCPU is 89790 bytes and that of the Q02UCPU is 93630 bytes.

(4) Execution method

(a) Execution by contacts

1) Setting method

Set "Latch data backup operation valid contact" in the PLC system tab of the PLC parameter dialog box (Devices X, M, or B can be selected).

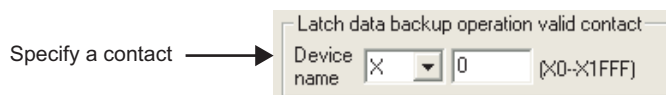


Figure 6.108 Setting screen of latch data backup start contact to standard ROM

2) Execution method

Backup starts at the rise of a contact (off → on).

After backup, the BAT.LED of the CPU module flashes (green), indicating that the CPU module is in the standby status ready to be powered off.

3) Precautions

- a) Since data to be backed up is the data when a contact is on (END processing), the CPU module will not become the RUN status until the CPU module is powered on again or is reset after this operation.
- b) Since the status of latch data backup start contact to the standard ROM is checked at execution of the END instruction, data are not backed up even if a contact is turned on → off → on, or off → on → off in one scan.
- c) In the following cases, data are not backed up unless the latch data backup start contact to the standard ROM is turned off and then on again.
 - The latch data backup start contact to the standard ROM is set to X, and the CPU module is powered off and then on or is reset after backup by turning off and then on the contact.
 - The latch data backup start contact to the standard ROM is set to M or B, and data are backed up by turning off and then on the contact.

(b) Execution by remote operation

1) Execution method

Select [Online] → [Latch data backup operation] of GX Developer.

After backup, the BAT.LED of the CPU module flashes (green), indicating that the CPU module is in the standby status ready to be powered off.

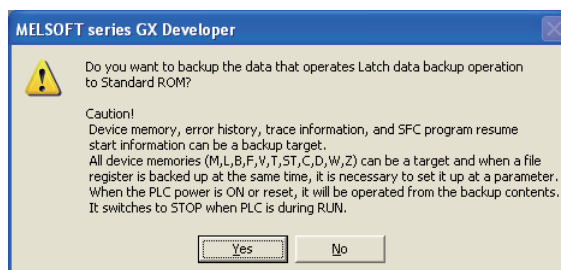


Figure 6.109 Remote operation execution screen

Data to be backed up is the data at the execution of remote operation.

(5) Restoring backup data

The backup data is automatically restored by the following operations.

- At power-off → on of the CPU module
- At reset

Whether to restore data once after backup or per above operation is set by SM676 (Specification of restore repeated execution).

Table 6.47 Status of SM676 and restoration operation

Status of SM676 at backup operation	Restoration operation
SM676 is off.	Data are restored once when the CPU module is powered off and then on or is reset after backup.
SM676 is on.	Data are restored whenever the CPU module is powered off and then on or is reset after backup. Data are repeatedly restored until the backup data are deleted or the latch data are backed up next time.

After backup data are restored, the BAT.LED on the CPU module turns on (green) for five seconds.

Point

If the number of device points configured in the parameter setting and the number of device points at the time of backup are different, "RESTORE ERROR" (error code: 2220) is detected when the backup data are restored, and data restoration is not normally completed.

(Data will be restored again when the CPU module is powered off and then on or is reset in the next time.)

Perform any of the following operations to normally complete restoring data.

- Return the status of data when the parameters are backed up.
- Delete the backup data.
- Backup data again.

(6) Deleting backup data

Select [Online] → [Latch data backup operation] of GX Developer (Set the CPU module to STOP since data cannot be deleted in the RUN status).

Also, information of special registers (SD671 to SD675) can be initialized (cleared to 0) by deleting the backup data.

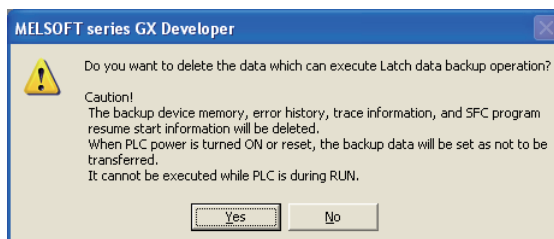


Figure 6.110 Screen for deleting backup data

(7) Checking with special relays and special registers

The status of execution of latch data backup to the standard ROM or restoration operation can be checked by SM671, SM676, SD671 to SD679.

(8) Precautions

The following provides precautions for backing up latch data.

- 1) Do not power off or reset the CPU module during backup of latch data. If performed, "RESTORE ERROR" (error code: 2221) is detected, and backup data is not restored (The backup data are deleted).
- 2) Even if the backup data exists, the initial device value has a priority over the backup data when the initial device value is set.
Therefore, the device where the initial device value setting is configured is overwritten by the device data of the initial device value after reflecting the backup data.
- 3) Even if the latch device or latch range setting is used, backup data has a priority. Therefore, even when data of latch device and latch range setting are changed after backup, it is overwritten by data backed up when the CPU module is powered off and then on or is reset.
- 4) Devices where local device range setting is configured are not backed up. They are initialized (cleared to 0) when the CPU module is powered off and then on or is reset.
- 5) When the number of writes to the standard ROM exceeds 100,000 times ("FLASH ROM ERROR" (error code: 1610) is detected), data may not be normally backed up.
- 6) Backup data cannot be deleted unless the data are deleted or the storage location memory (standard ROM) for the backup data is formatted.
- 7) The following operations cannot be performed during latch data backup. Perform them after the backup operation.
If performed, an error is displayed on the GX Developer screen.
 - Format PLC memory (standard ROM only)
 - Latch data backup by remote operation
 - Online change (ladder mode, files, function block)
 - Write the program memory to ROM
 - Write to PLC (Flash ROM)

6.29 Writing/Reading Device Data to/from Standard ROM

(1) Definition

This function writes device data to the standard ROM.
Writing the fixed values for operation and operation results to the standard ROM can prevent losing data due to low battery. Also, timing of writing to the standard ROM can be set by an instruction.

(2) Execution method

Device data are written to the standard ROM by the SP.DEVST instruction.
The device data written to the standard ROM is read to the specified device by the S(P).DEVLD instruction.

(3) Devices that can be written to the standard ROM

Table6.48 shows the devices whose data can be written to the standard ROM.

Table6.48 Devices that can be written to the standard ROM and storage location

Device	Storage location
X,Y,M,L,B,F,V,T,ST,C,D*1,W*2,SM,SD,SB,SW,R,ZR	Standard ROM(device data storage location)

*1: The extended data register (D) is included.
*2: The extended link register (W) is included.

(4) Setting method

The area for storing the device data is set in the standard ROM by the PLC file tab of the PLC parameter dialog box.

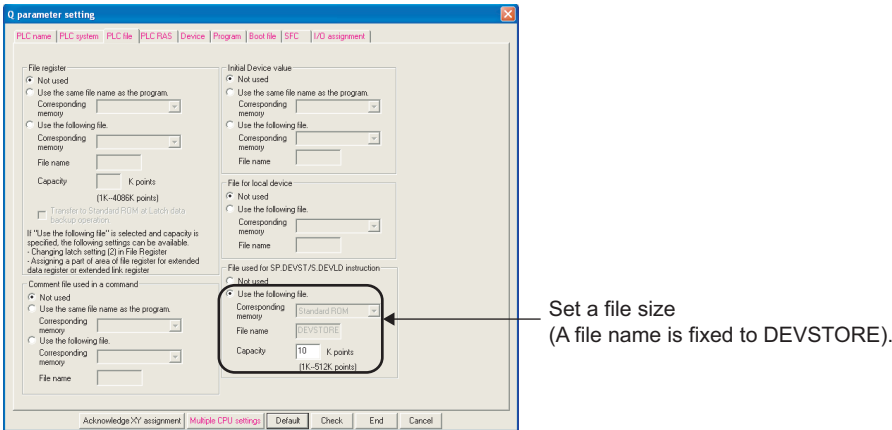


Figure 6.111 Screen for setting a file for storing device data

Remark
For details of the instructions, refer to the following.
➡ MELSEC-Q/L Programming Manual (Common Instruction)
.....

6.30 CPU Module Change Function with Memory Card Note6.13

(1) Definition

This function backs up data in the CPU module to a memory card and restores the backup data to another CPU module.

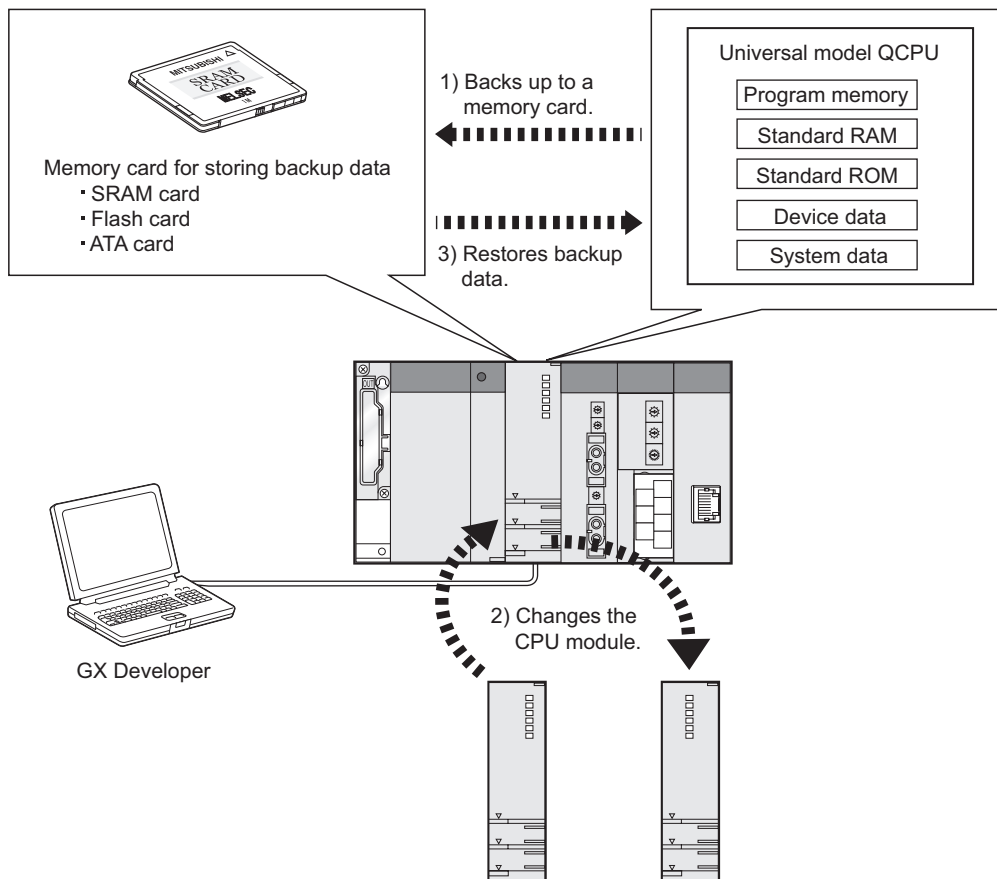


Figure 6.112 CPU module change function with memory card

This function consists of the following two functions.

- Backup function to memory card:  Section 6.30.1
- Backup data restoration function:  Section 6.30.2



Note6.13

Universal

The Q00UJCPU, Q00UCPU, and Q01UCPU do not support the CPU module change function with memory card. When using the function for the Q02UCPU, Q03UDCPU, Q04UDHCPU, Q06UDHCPU, Q13UDHCPU, Q26UDHCPU, or Built-in Ethernet port QCPU, check the versions of the CPU module and GX Developer. ( Appendix 2)

(2) Backup data file

After data are backed up, a backup data file "MEMBKUP0.QBP" is created in a memory card.

Only one backup data file can be stored to a memory card.

When data are backed up to a memory card containing a backup data file again, the stored backup data file is overwritten. When a memory card contains another file, a backup data file can be stored in the memory card without deleting the stored file. *1 *2

*1: After restoration, if the parameters are stored in the memory card but not in the CPU module, the CPU module operates by the parameter settings in the memory card.

*2: When data are backed up to the Flash card, only a backup data file can be stored.

The backup data file can be deleted by "Delete PLC data" of GX Developer.

(3) Backup data

(a) Backup data selection

Table6.49 shows data to be backed up.

Table6.49 Data to be backed up

Backup data (drive)	Description	Backup selection by the user
Program memory (drive 0)	All data in the program memory (drive 0) *1	Selectable
Standard RAM (drive 3)	All data in the standard RAM (drive 3)	
Standard ROM (drive 4)	All data in the standard ROM (drive 4)	
Device data*2	Internal user device (L, B, F, V, T, ST, C, D, W)	Not selectable (Data are automatically backed up by the system.)
System data	Data managed by the system (such as error history)	

*1: Data in the program cache memory are backed up.

*2: Only the latch relay (L) and devices to which the latch range setting can be configured are backed up.

(b) Maximum backup data size

Table6.50 shows the maximum size of data that can be backed up.

Table6.50 Maximum size of data that can be backed up

Backup target data (drive)	Q02UCPU	Q03UD/ Q03UDECPU	Q04UDH/ Q04UDEHCPU	Q06UDH/ Q06UDEHCPU	Q10UDH/ Q10UDEHCPU
Program memory (drive 0)	82	124	164	244	408
Standard RAM (drive 3)	130	194	258	770	1026
Standard ROM (drive 4)	516	1032	1032	1032	2056
Device data	128	128	128	128	128
System data	41	64	64	64	64
Total (maximum)	897	1542	1646	2238	3682

Backup target data (drive)	Q13UDH/ Q13UDEHCPU	Q20UDH/ Q20UDEHCPU	Q26UDH/ Q26UDEHCPU	Q50UDEHCPU	Q100UDEHCPU
Program memory (drive 0)	528	808	1048	2008	4008
Standard RAM (drive 3)	1026	1282	1282	1538	1794
Standard ROM (drive 4)	2056	4104	4104	8200	16392
Device data	128	128	128	128	128
System data	64	64	64	64	64
Total (maximum)	3802	6386	6626	11938	22386

(Unit: K byte)

The backup data size can be checked by the following methods.

- Execution screen of GX Developer (☞ Section 6.30.1(1)(b))
- SD698 and SD699(☞ CHAPTER 12)*¹

*1: Can check the data size after backup starts.

6.30.1 Backup function to memory card

This function can save data in the CPU module to a memory card.

If a memory card is used in running system, data can be backed up by replacing the current memory card with the one for storing a backup data.

(1) Methods

(a) Method with contacts

1) Setting

To back up data with contacts, turn on the device set in the "PLC module change setting" screen in the PLC system tab of the PLC parameter dialog box.

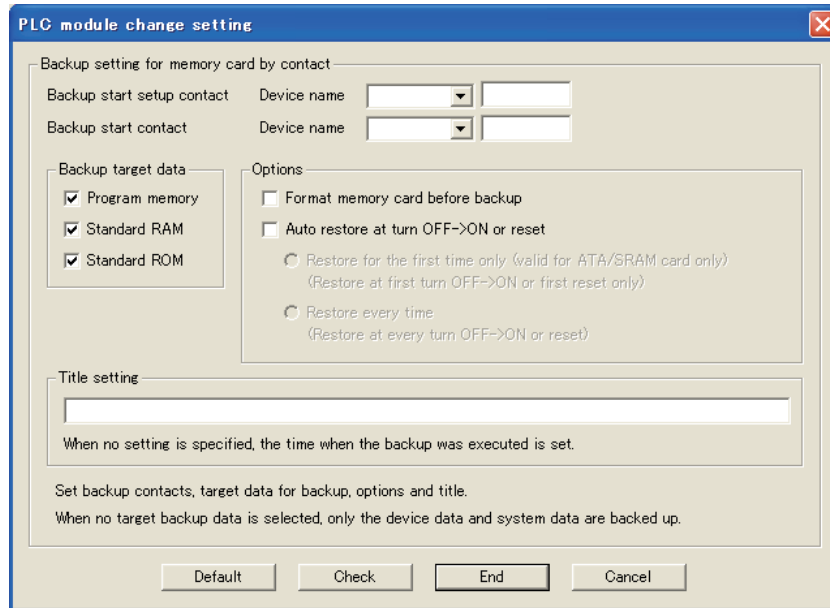


Figure 6.113 PLC module change setting screen

Table6.51 Setting items

Item	Description	Setting range	Default
Backup start setup contact*1	At the rise of the selected device among the items shown in the right column, backup is ready to start.	Available devices*2 • X (0 to 1FFF) • M (0 to 8191)*3 • B (0 to 1FFF)*3	-
Backup start contact	At the rise of the selected device among the items shown in the right column, backup enters execution status.		
Backup target data	Select data to be backed up stored in any of memories shown in right column.*4	• Program memory (drive 0) • Standard RAM (drive 3) • Standard ROM (drive 4)	All drives are backed up.
Format memory card before backup	Select whether to format a memory card before backup.	Formatted Not formatted	Not formatted
Title setting*5	Set a title appended to backup data stored in a memory card.	32 characters (16 in two-byte characters)	Set the title of the current time. (Example) If data are backed up at 12 p.m. on October 1, 2008, "200810011200" is set.

*1: The CPU module enters the STOP status at the rise of the backup start setup contact, the backup start contact cannot be turned on in the sequence program.

*2: The backup start setup contact and backup start contact cannot be set to the same device.

*3: This indicates the default number of points. The setting range when the internal user device is set to the maximum number of points (60K points) is M(0 to 61439) and B(0 to 0EFFF).

*4: If data to be backed up is deselected in "Backup target data", only device data and system data are backed up.

*5: The title is used for identifying backup data.

Settings for the title of the backup data stored in a memory card can be checked on the Delete PLC data screen of GX Developer.

2) Operating procedure

Turn on in the order of the backup start setup contact and the backup start contact. Data are not backed up when only the backup start contact is on.

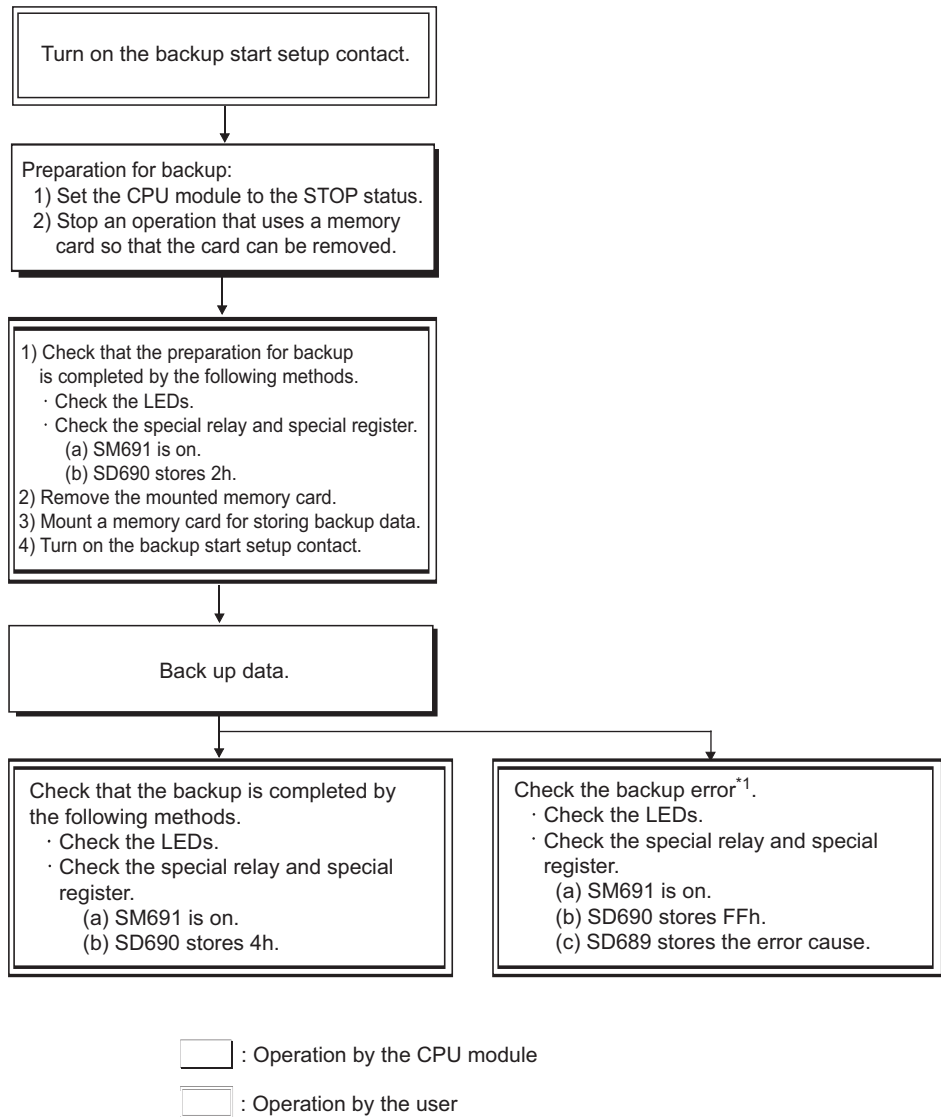


Figure 6.114 Backup procedure using contacts

*1: Since the SM691 (Backup start preparation status flag) is on, data can be backed up again by turning off and then on the backup start contact.

3) Operation of contacts

Operations of the backup start setup contact, backup start contact, SM691 (Backup start preparation status flag), and SD690 (Backup status) at backup are shown in Figure 6.115.

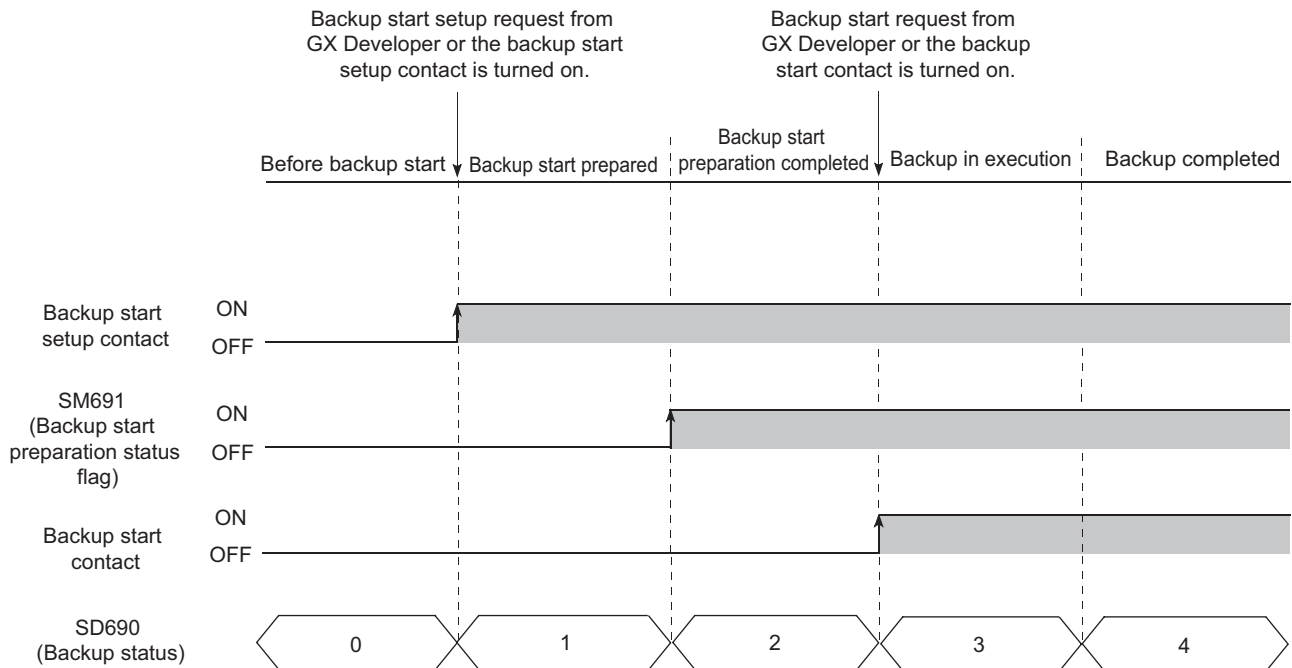


Figure 6.115 Operation of contacts

If the backup start contact is turned on while the value in SD690 is 0_H (Before backup start) or 1_H (Backup start prepared), data are not backed up.

If the backup start contact is on before the value in SD690 becomes 2_H (Backup start preparation completed), turning off and then on the backup start contact again while the value in SD690 is 2_H (Backup start preparation completed) starts backup.

(b) Backup from GX Developer

Select [Online] → [PLC module change] → [Create backup data...].

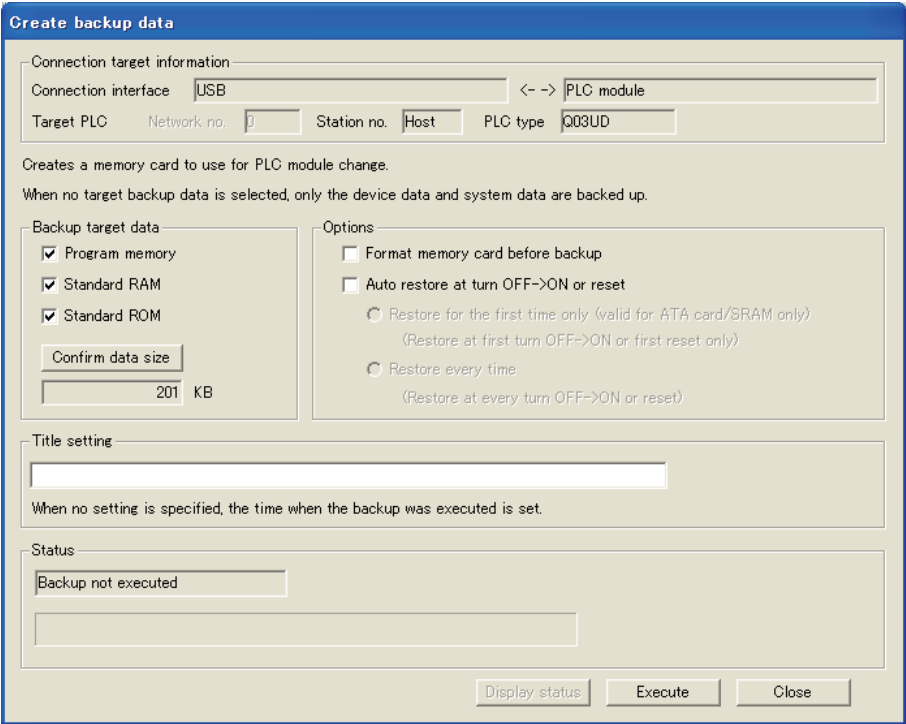


Figure 6.116 Create backup data screen

For details of the operating procedure by the GX Developer wizard, refer to the following.

 GX Developer Version 8 Operating Manual

Point

Clicking the **Confirm data size** button on the screen shown in Figure 6.116 displays the size of backup data created at backup.
Data size can be checked by GX Developer connected to the CPU module.
This function is available regardless of the operating status of the CPU module (RUN, STOP, PAUSE) or mounting status of a memory card.

(2) Operations of backup to a memory card

(a) Mounting/removal of a memory card

In a system using a memory card, the memory card can be changed/removed after preparation for backup has been completed (Turning on SM609 (Memory card remove/insert enable flag) is unnecessary).

After preparation for backup has been completed, the CPU module turns off SM604 (Memory card in-use flag).

(b) Status of backup to a memory card

Table6.52 shows backup status.

Table6.52 Backup status

Status	Description	Value in SD690
Before backup start	Backup is not started.	0H
Backup start prepared	A memory card can be mounted/removed.	1H
Backup start preparation completed	Set data to be backed up.	2H
Backup in execution	Backup is in execution.	3H
Backup completed	Backup is normally completed.	4H
Backup error	Backup failed due to an error.	FFH

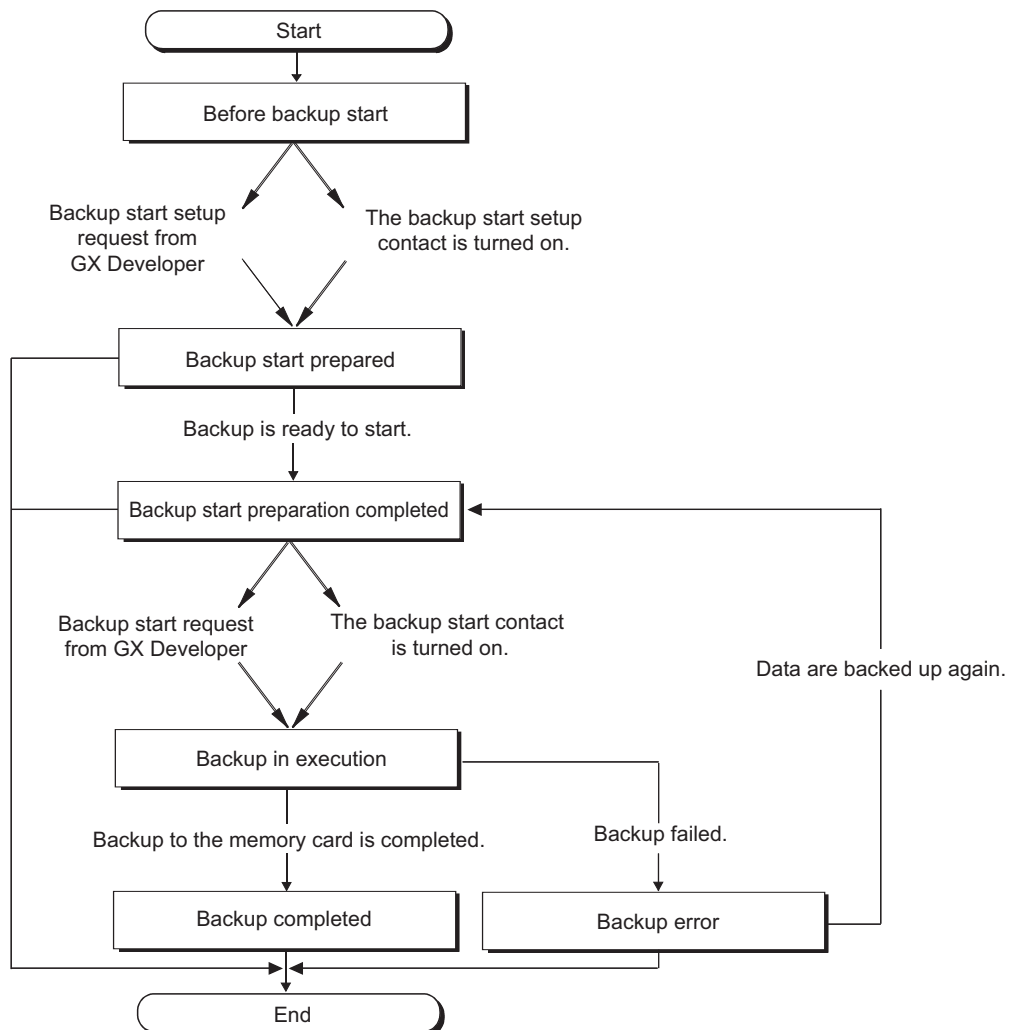


Figure 6.117 Flow of status change at backup

(c) Operations of special relays and special register*1

Figure 6.118 shows the operations of the SM609 (Memory card remove/insert enable flag), SM691 (Backup start preparation status flag), and SD690 (Backup status).

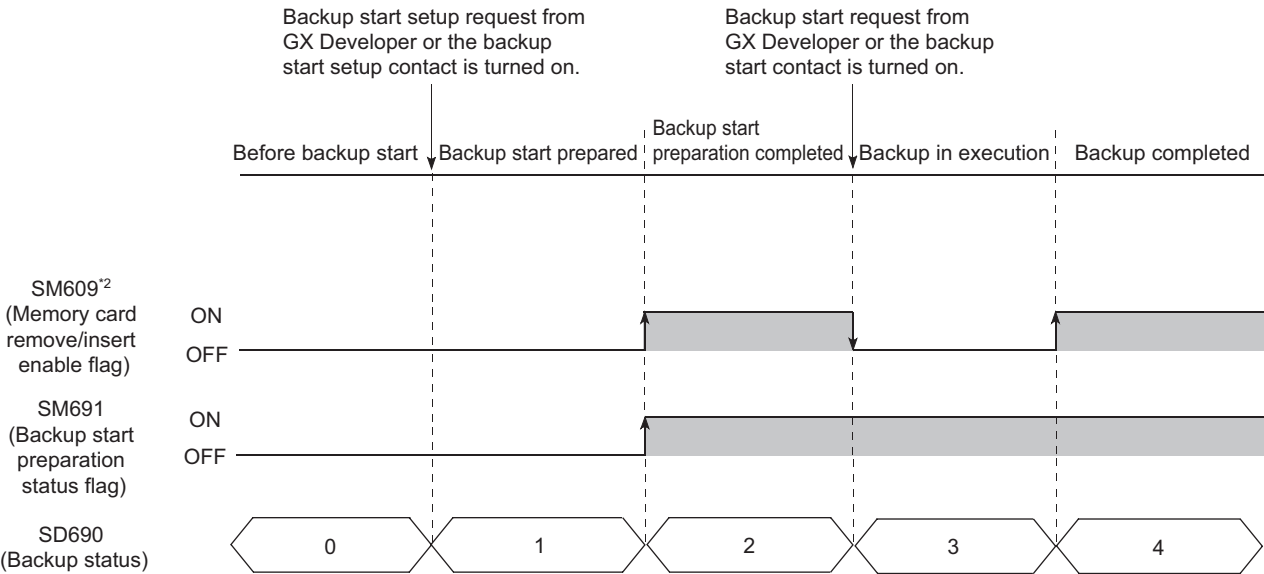


Figure 6.118 Special relays and special register at backup

*1: For details of the special relays and special register used for backup, refer to CHAPTER 12.
*2: This special relay is turned on/off by the system.

(d) Operating status of the CPU module

After END processing where the backup start setup request is accepted, the CPU module changes the operating status to STOP at backup. Therefore, data can be backed up regardless of the operating status. After backup, power off and then on the CPU module or reset the CPU module.(If the CPU module is not powered off and then on or is not reset but set the RESET/STOP/RUN switch to RUN, the CPU module remains in the STOP status).

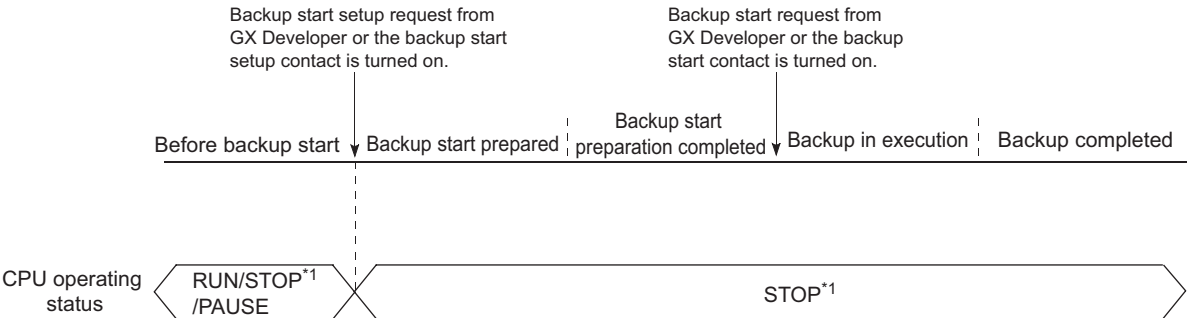


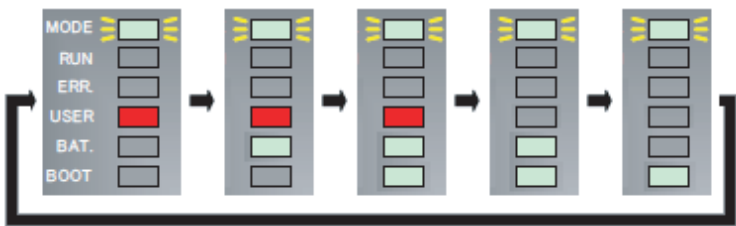
Figure 6.119 Operating status of the CPU module at backup

*1: The status includes a stop error.

(3) LEDs indicating backup status

The LEDs on the front of the CPU module indicate backup status.

Table6.53 LEDs indicating backup status

Value stored in SD690	Backup status	LED indication
2H	Backup start preparation completed	MODE: Flash (green), BAT.: Flash (green)
3H	Backup in execution	The color changes at intervals of 800ms as follows.  ■ : Flashing (green) ■ : On (red) ■ : On (green)
4H	Backup completed	MODE: Flash (green), BAT.: Flash (green), BOOT: Flash (green)
FFH	Backup error	MODE: Flash (green), USER: MODE: Flash (red), BAT.: Flash (green)

(4) Error cause of backup

When backup is not normally completed, an diagnostic error is not detected. In that case, the error cause is stored in SD689(Backup error factor) and the error response is returned to GX Developer.

Table6.54 Error cause of restoration

Value stored in SD689	Error response number	Error cause
100H ^{*1}	41FEH ^{*2}	Backup started with no memory card inserted.
200H	-	Size of data to be backed up exceeds the capacity of mounted memory card.
300H	-	Write protection has been set to the memory card.
400H	-	Writing to the memory card was not normally completed.
500H	-	Reading from a drive storing backup data was not normally completed (program memory read error).
503H	-	Reading from a drive storing backup data was not normally completed (standard RAM read error).
504H	-	Reading from a drive storing backup data was not normally completed (standard ROM read error).
510H	-	Reading from a drive storing backup data was not normally completed (system data read error).
600H ^{*1}	4335H ^{*2}	Backup preparation started while latch data were backed up to the standard ROM.
601H ^{*1}	410AH ^{*2}	Backup preparation started while data were written in the RUN status.
602H ^{*1}	4336H ^{*2}	Backup preparation started while a FTP client connected a FTP to the CPU module in FTP connection.
-	4082H ^{*3} , 4330H ^{*4}	Backup preparation or backup started while another backup was in execution.
-	4333H ^{*2}	Backup started while the CPU module was in "Before backup start" (SD690 = 0).

*1: Only when data are backed up with contacts

*2: Only when data are backed up from GX Developer

*3: Only when data area backed up from another boot source.

*4: Only when data area backed up from the same boot source.

(5) Functions that cannot be performed during backup

Table6.55 shows functions that cannot be performed during backup.

Table6.55 Functions that cannot be performed during backup

Category	Function	Category	Function
Drive operation	Format PLC memory	Trace	Sampling trace registration
	Write the program memory to ROM	Remote operation	Remote latch clear
	Program memory batch transfer	File operation	Write to PLC
	Arrange PLC memory		Delete PLC data
Online change	Online change (ladder mode)		Write PLC user data
	Online change (files)		Delete PLC user data
	Online change (multiple blocks)		Password registration
	Change TC setting		Latch data backup operation
Monitor	Monitor condition setup	FTP function	For all operations and commands
Debug	Executorial conditioned device test		

(6) Precautions

(a) Do not perform the following operations during backup.

- Mounting/removal of a memory card
- Power-off of the CPU module
- Reset

(b) Even if booting parameters backed up with contacts for the Universal model QCPU having the serial number (first five digits) is "10101" or earlier, the system ignores the parameters.

In this case, even if the backup start setup contact or backup start contact set by the parameters is turned on, the operating status of the CPU module does not change (a diagnostic error is not detected).

(c) After backup is ready to start, the following functions stop (do not resume the operations even after the backup).

- Refresh of network modules
- Data link transfer
- Auto refresh of intelligent function modules
- Auto refresh of CPU shared memory

6.30.2 Backup data restoration function

This function restores data backed up to a memory card to the CPU module.

(1) Methods

(a) Restoration from GX Developer

Select [Online] → [PLC module change] → [Restore...].

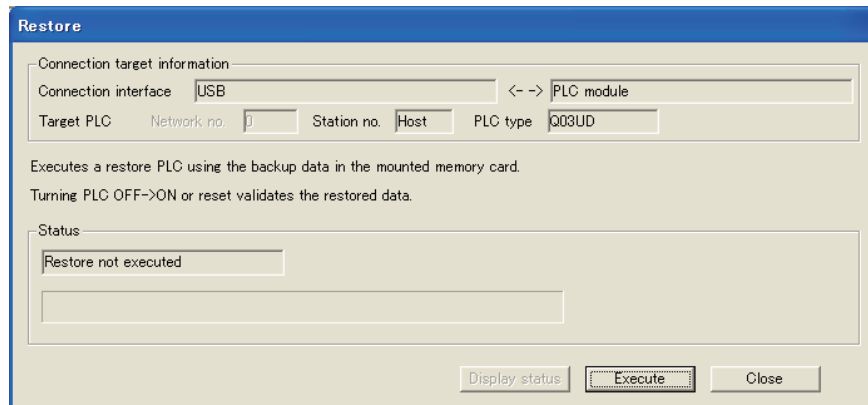


Figure 6.120 Restore screen

Select "Yes" in the screen appearing after clicking "Execute" in the above screen and power off and then on or reset the CPU module. Restored data will become valid.

(b) Automatic restoration

Select "Auto restore at turn OFF → ON" in "PLC module change setting" in the PLC system tab of the PLC parameter dialog box. (☞ Section 6.30.1(1)(a))

After data are backed up and the CPU module is powered off and then on or is reset, restoration automatically starts.

Also, items shown in Table 6.56 can be set by the option setting.

Table 6.56 Option setting items

Option setting item	The CPU module is powered off and then on or is reset	
	First time	After the first time
Restore for the first time only *1 *2	Restored	Not restored (The CPU module operates the memory card as usual.) *3
Restore every time	Restored	Restored

*1: This setting is valid only when backup data are stored to the ATA card or SRAM card.

Note if the write protect switch of the SRAM card is set to be valid (write protection), the setting does not become valid and the data are restored even after the first time ("RESTORE ERROR" (error code: 2226) occurs).

*2: When using the FLASH card, restoration can be performed even after the first time.

*3: Restoration after the first time can be performed from GX Developer.

Data are restored at initial processing after the CPU module is powered off and then on or is reset.

After restoration, powering off and then on the CPU module or resetting the CPU module again is not required to change the operating status to the one set with the RUN/STOP/RESET switch.

If a memory card needs to be changed after restoration, turn on SM609 (Memory card remove/insert enable flag) and change the memory card after SM600 (Memory card usable flags) turns off.

(2) Operation for restoring backup data

Figure 6.121 shows operation for restoration.

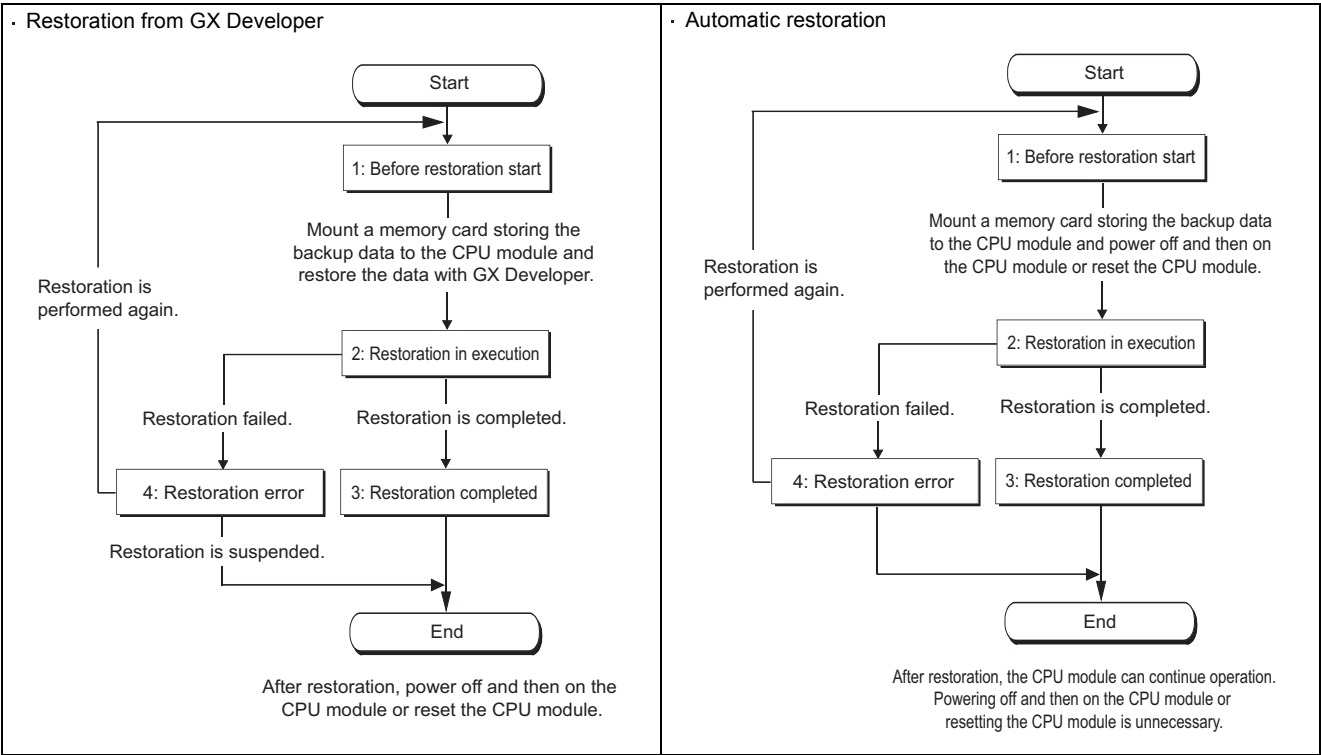


Figure 6.121 Operation for restoration

A value indicating restoration status is stored in SD693 (Restoration status).

Table6.57 Status at restoration

Status	Description	Value stored in SD693
Before restoration start	Restoration is not performed.	0H
Restoration in execution	Restoration is in execution.	1H
Restoration completed	Restoration is normally completed.	2H
Restoration error	Restoration failed due to an error.	FFH

(3) Operation of special relay and special register*1

Figure 6.122 shows the operation of SM692 (Restoration complete flag) at restoration.

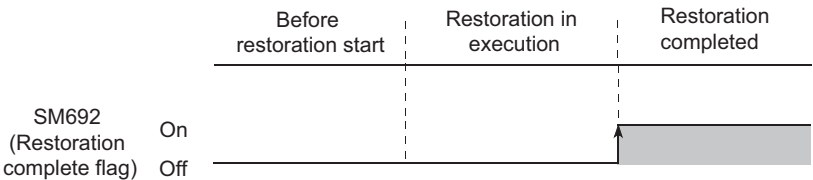


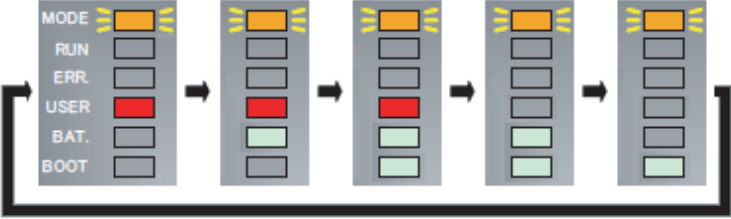
Figure 6.122 Operation of the special relay

*1: For details of special relays and special registers for restoration, refer to CHAPTER 12.

(4) LEDs indicating restoration status

The LEDs on the front of the CPU module indicate restoration status.

Table6.58 LEDs indicating restoration status

Value stored in SD693	Restoration status	LED indication
0H	Before restoration start	MODE: On (green)
1H	Restoration in execution	The color changes at intervals of 800ms as follows.   : Flashing (orange)  : On (red)  : On (green)
2H	Restoration completed	- Restoration from GX Developer MODE: Flashing (orange), BAT.: Flashing (green), BOOT: Flashing (green) - Automatic restoration MODE: On (green)
FFH	Restoration error	- Restoration from GX Developer MODE: Flashing (orange), USER: Flashing (red), BAT.: Flashing (green) - Automatic restoration MODE: On (green), ERR: Flashing (red)

(5) Error cause of restoration

When restoration is not normally completed, an diagnostic error is not detected. In that case, the error cause is stored in SD692 (Restoration error factor) and the error response is returned to GX Developer.

Table6.59 Error cause of restoration

Value stored in SD692	Error response number	Error cause
800H	-	The CPU module where data are restored is different model with the one where the backup source data are stored.
801H	-	- Backup data files do not match. - Reading of backup data from a memory card was not normally completed.
810H	-	Writing of backup data to the restoration destination drive was not normally completed.
-	4335H ^{*1}	Restoration started while latch data were backed up to the standard ROM.
-	410AH ^{*1}	Restoration started while data were written in the RUN status.
-	4336H ^{*1}	Restoration started while a FTP client connected a FTP to the CPU module in FTP connection.
-	4330H ^{*1}	Restoration started while another restoration was in execution.
-	41FEH ^{*1}	Restoration started while a memory card was not mounted.

*1: Only when data are restored from GX Developer

When automatic restoration is not normally completed, "RESTORE ERROR" (error code: 2225 to 2227) is detected.

Table6.60 Error when automatic restoration is not normally completed

Error code	Error message	Error cause
2225	RESTORE ERROR	The CPU module where data are restored is different model with the one where the backup source data are stored.
2226		• Backup data file is corrupt (The contents of backup data file do not match with the check code). • Reading of backup data from a memory card was not normally completed. • The write protect switch of the SRAM card is valid (write protection), and therefore "Restore for the first time only" cannot be set to valid.
2227		Writing of backup data to the restoration destination drive was not normally completed.

Remark

For details of error codes, refer to the following.

 QCPU User's Manual (Hardware Design, Maintenance and Inspection)

(6) Functions that cannot be performed during restoration

Functions that cannot be performed during restoration are the same as the ones that cannot be performed during backup. ( Section 6.30.1(5))

(7) Precautions

(a) Do not perform the following operations during restoration.

- Mounting/removal of a memory card
- Power-off of the CPU module
- Reset

(b) Even if mounting a memory card storing backup data file to the Universal model QCPU having the serial number (first five digits) is "10101" or earlier and powering off and then on the CPU module or reset the CPU module, the system ignores the backup data file and starts operation (a diagnostic error is not detected).

(c) Do not use a memory card where the boot file setting parameter is stored. If used, data in the memory card are overwritten by the boot file setting even restoration is performed.

(d) If restoration is started from GX Developer, the following functions stop (do not resume the operations even after the restoration).

- Refresh of network modules
- Data link transfer
- Auto refresh of intelligent function modules
- Auto refresh of CPU shared memory

(1) Definition

This function reads the model name of a module on a base unit.
The mounted module is identified in a ladder program and processing according to the module can be performed.

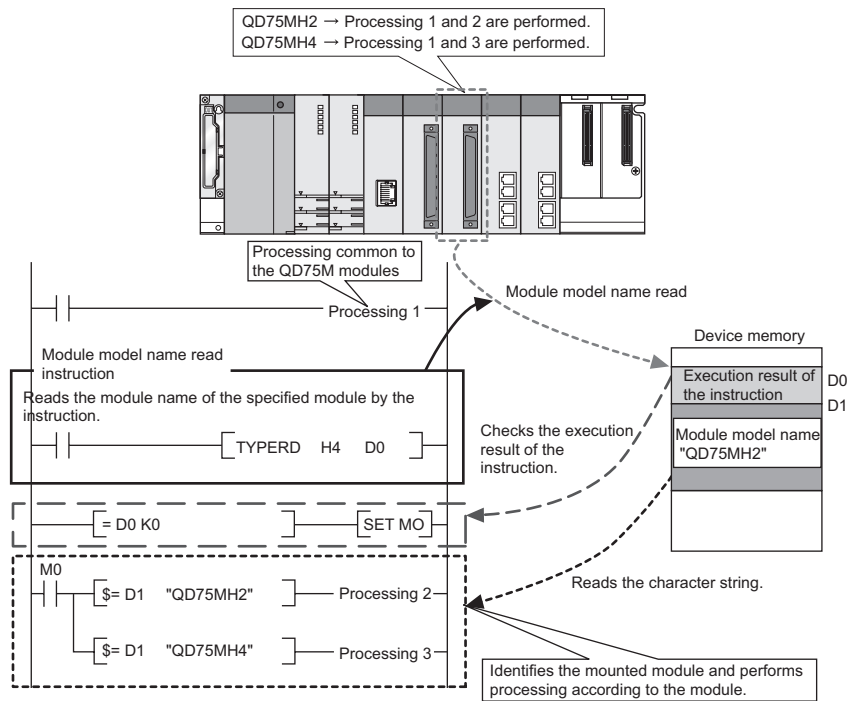


Figure 6.123 Operation of the module model name read

(2) Execution method

Use the TYPERD instruction.

Remark

For details of the instruction, refer to the following.

 MELSEC-Q/L Programming Manual (Common Instruction)

 Note6.14

Universal

When using the module model name read function, check the versions of the CPU module and GX Developer.

( Appendix 2)

6.32 Module error collection Note6.15

(1) Definition

This function collects errors occurred in the connected intelligent function modules in the CPU module. By storing the errors in a memory that can hold data in the event of a power failure, the errors can be held even after power-off or reset.

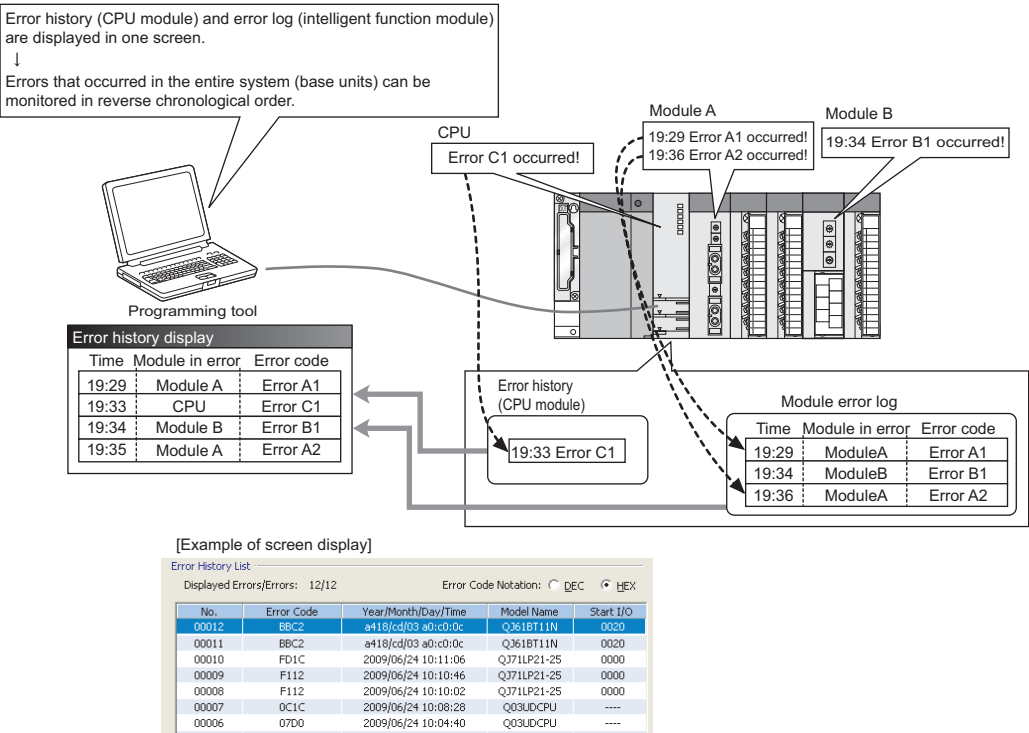


Figure 6.124 Operation of module error collection

(2) Supported modules

The CPU module collects errors occurred in the connected intelligent function modules*1. The CPU module does not collect the errors of intelligent function modules on other stations in the network.

*1: Indicates intelligent function modules that support this function. For supported module versions, refer to the manual for each module.

(3) Timing when module errors are collected

Module errors are not collected during execution of a program such as the COM instruction but collected in END processing.

 Note6.15 **Universal**

When using the module error collection, check the versions of the CPU module and GX Works2. ( Appendix 2) GX Developer cannot display module errors.

(4) Storing module errors

The module errors can be stored either the system memory^{*1} or the standard RAM.
The errors are stored separately from error history (CPU module) data.

^{*1}: The memory is managed inside the system.

Table6.61 Storage locations and the number of storable module errors

CPU module	System memory	Standard RAM
Q00UJCPU	40 (Fixed)	-
Q00UCPU, Q01UCPU	40 (Fixed)	1000
Q02UCPU, QnUD(H)CPU, Built-in Ethernet port QCPU	100 (Fixed)	1000

(5) Setting method

Select "Collect error histories of intelligent function modules" in "Module Error History Collection (Intelligent Function Module)" in the PLC RAS tab of the PLC parameter dialog box.

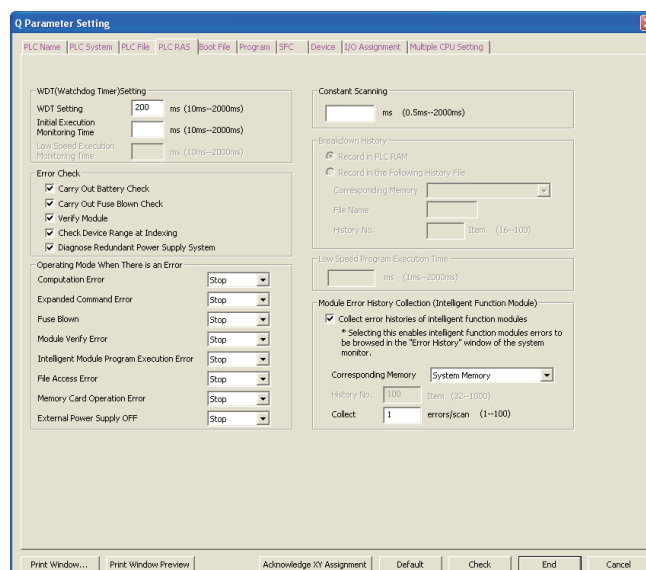


Figure 6.125 PLC RAS

Table6.62 PLC RAS setting list

Item	Description	Setting range	Default
Corresponding Memory	Select a storage location.	- System Memory - Standard RAM^{*1, *2}	System Memory
History No.	Set the number of collected errors only when the errors are stored in the standard RAM.	32 to 1000	40/100
Collection No.	Set the number of collected errors in one scan. ^{*3}	- Stored in system memory: 1 to 100 - Stored in standard RAM: 1 to 128	1

^{*1}: When a sampling trace file is stored to the standard RAM, powering off and then on or resetting the CPU module will delete the file.

^{*2}: The battery consumption may be increased.

QCPU User's Manual (Hardware Design, Maintenance and Inspection)

^{*3}: If collected module errors are frequently lost, set a greater value to "Collection No.". The recommended value is the number of intelligent function modules that support this function.

Parameter settings are enabled to the CPU module when:

- the CPU module is powered off and then on or
- the CPU module is reset.

(6) Monitoring module errors

Collected module errors can be monitored in the Error History screen opened by selecting [Diagnostics] → [System Monitor] → [System Error History] of GX Works2.

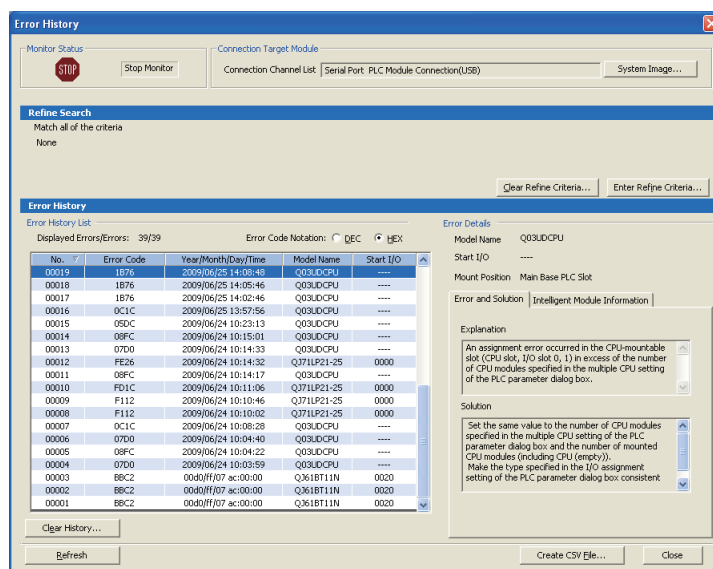


Figure 6.126 Error History screen

Table 6.63 Description of Error History List

Item	Description	Remarks
Error Code ^{*1}	Displays error code numbers.	-
Year/Month/Day/Time ^{*2}	Displays the year, month, day, hour, minute, and second when an error occurred.	The year can be displayed within the range of 1980 to 2079.
Model Name	Displays a module model name.	-
Start I/O	Displays the start I/O number of a module in error.	-

*1: For details of error codes, refer to the manual for the intelligent function module.

*2: If an error occurred during initial processing, its occurrence time may be stored as "0000/00/00 00:00:00" in the module error collection file. In this case, the error is not displayed in correct order in Error History List.

Point

- The Error History screen can be displayed by selecting a module figure in the System Monitor screen and clicking the **Error History Detail** button.
In this case, only the errors of the selected module is displayed.
☞ GX Works2 Version1 Operating Manual (Common)
- Errors are not displayed for modules that do not support the module error collection function.
- Errors may not be displayed when they occur successively.

(7) Clearing module error history

Clear module error history by clicking the **Clear History** in the Error History screen.

Note that error information on each intelligent function module displayed in the Module's Detailed Information screen is not cleared.

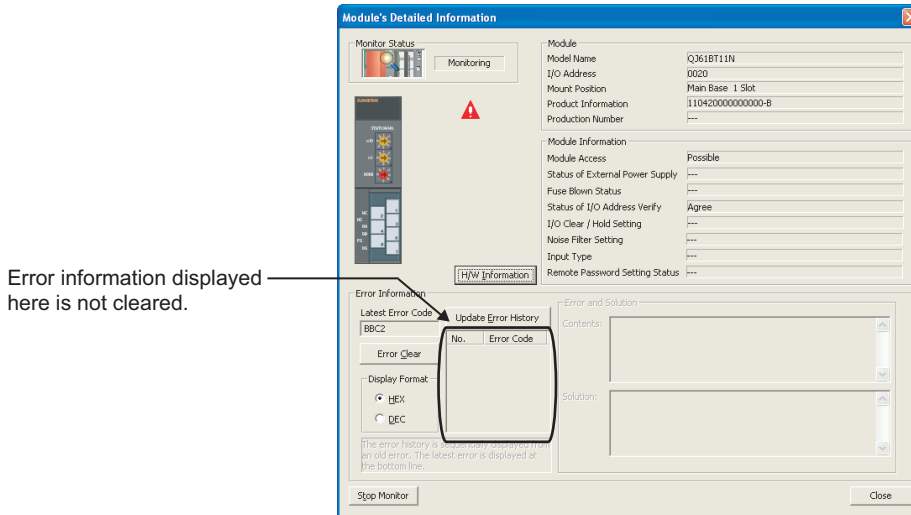


Figure 6.127 Module's Detailed Information screen

Point

The module error history is cleared when the standard RAM is formatted.

Note that a module error collection file cannot be deleted since it is automatically created after the CPU module is powered off and then on or is reset. To delete the file, clear the setting and then format the standard RAM.

(8) Precautions

(a) Using the CPU module change function with memory card

Backing up or restoring data will stop collecting module errors.

6.33 Local Device Batch Read Function Note6.16

(1) Definition

This function reads the contents of the CPU module local device in a batch, and allows them to be saved in CSV files.

Therefore, all the contents of local devices can be saved in CSV files.

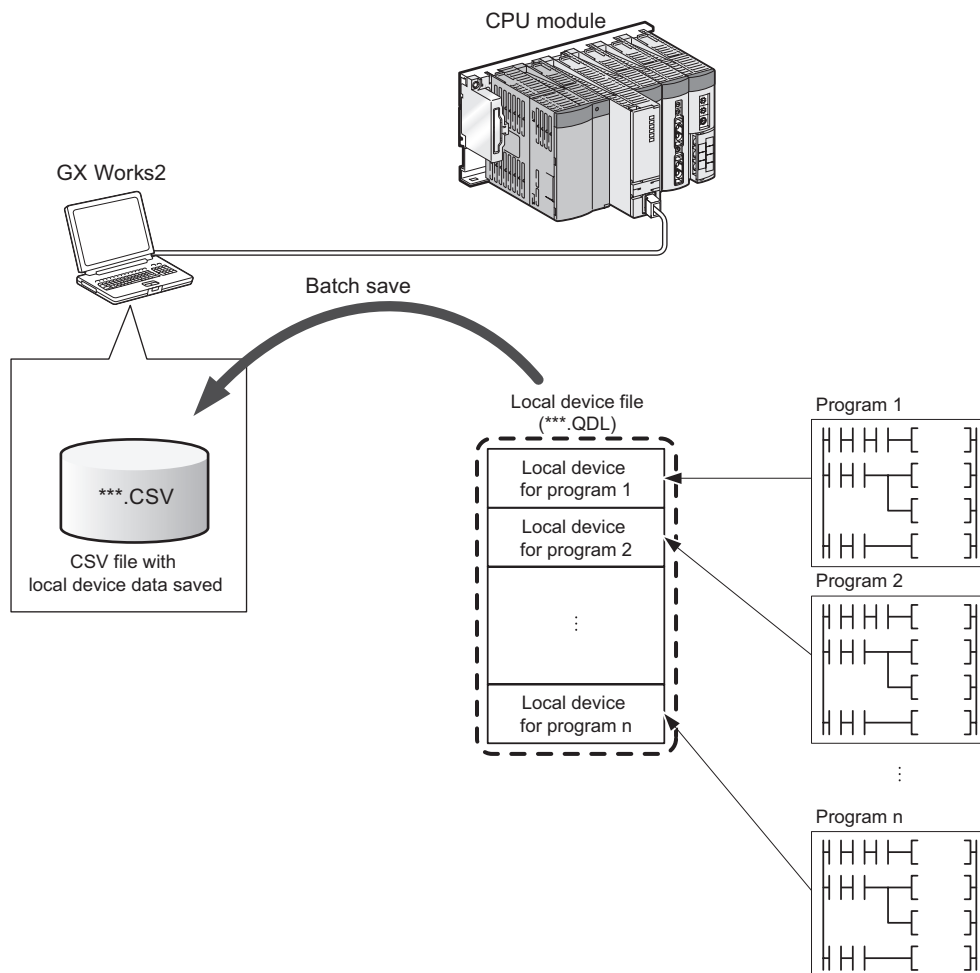


Figure 6.128 Operation of the local device batch read function



Note6.16

Universal

Before using the local device batch read function, confirm the CPU module and GX Works2 versions.

( Appendix 2)

The local device batch read function cannot be used in Q00UJCPU.

(2) Operating method

Open the local device batch read function from GX Works2 [Online] → [Local Device Batch Read + CSV Save]

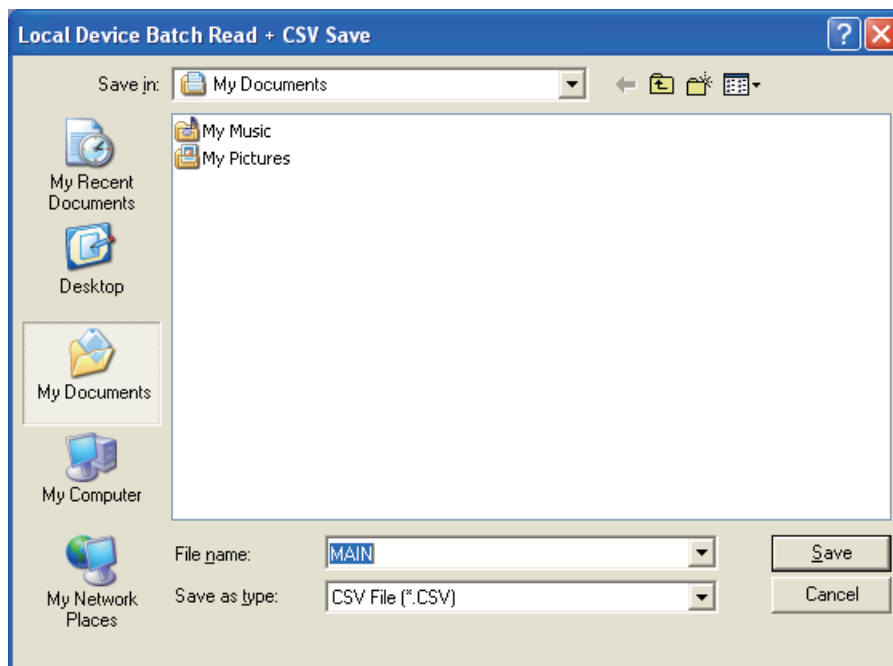


Figure 6.129 Local Device Batch Read + CSV Save screen

(3) CSV file contents and format

For the contents and format of CSV files, refer to the following.

☞ GX Works2 Version1 Operating Manual (Common)

(4) Precautions for using the local device batch read function

(a) When local device files do not exist

The local device batch read function cannot be executed.

(b) When the CPU module is in RUN status

If the local device batch read function is executed when the CPU module is in RUN status, the reading process is divided into multiple scans, and the read data are also divided into the number of scans.

Therefore, this function is recommended to use when the CPU module is in STOP or PAUSE status (when the device data do not change).

6.34 Send Points Expansion Function (CC-Link IE Controller Network Module) Note6.17

(1) Definition

This function expands the maximum link points per station using a CC-Link IE controller network module.

Cyclic transfer can be performed up to 32k points for link relay (LB) and 128k points for link register (LW).

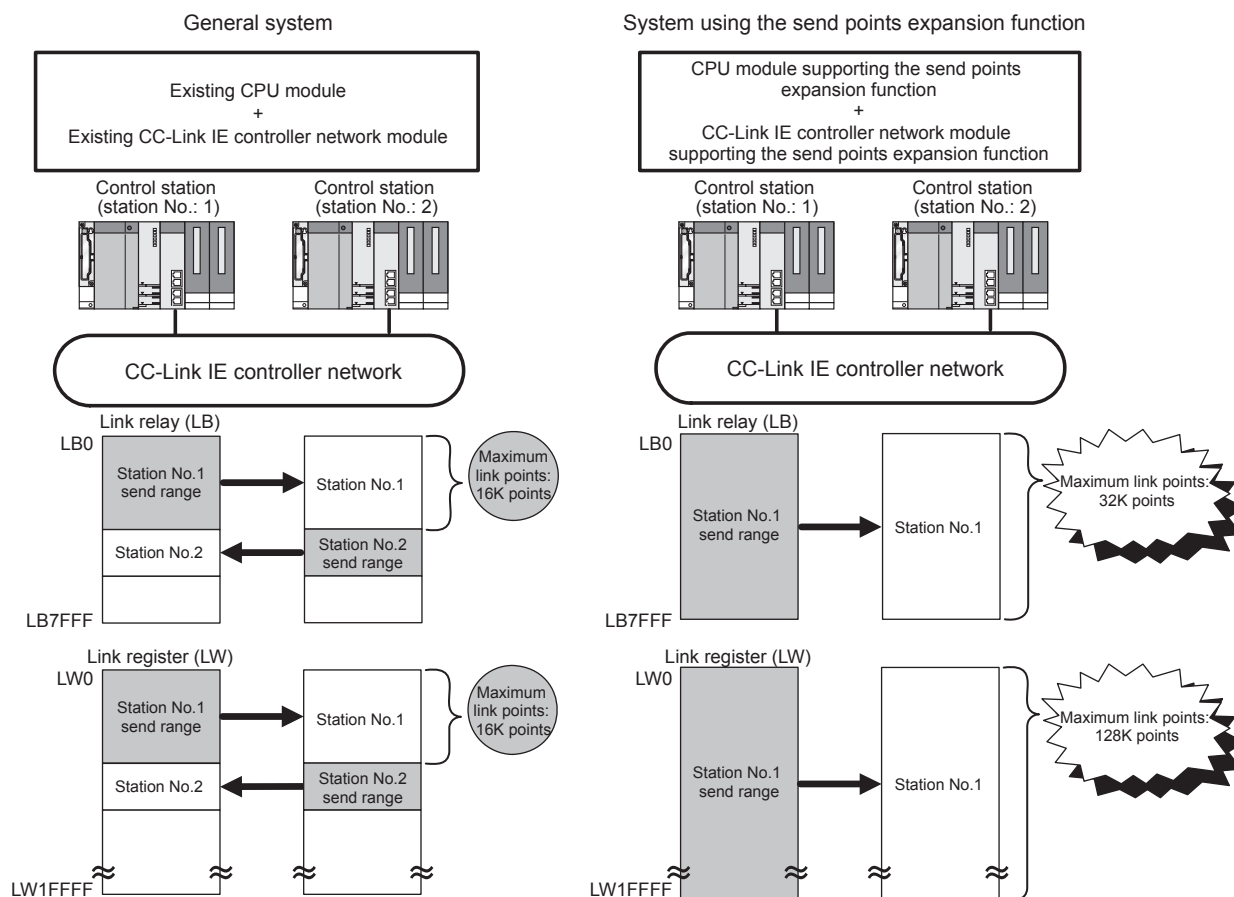


Figure 6.130 Send Points Expansion Function (CC-Link IE Controller Network Module)



Note6.17 **Universal**

When using the send points expansion function (CC-Link IE controller network module), check the versions of the CPU module and GX Works2. ( Appendix 2)

(2) Settings

This function can be used by setting the following network parameters with GX Works2.

- Network Type
- Network Range Assignment
- Refresh Parameters

For network parameter settings, refer to the following.

 CC-Link IE Controller Network Reference Manual

(3) Precautions

(a) Operation in boot operation

If the parameters for this function is stored on a memory card and the parameters are transferred to the CPU module that does not support this function, "LINK PARA. ERROR" (error code: 3102) will occur.

(b) Operation when backup data are restored from a memory card or GOT

If the parameters for this function backed up to a memory card or GOT are restored to the CPU module that does not support this function, "LINK PARA. ERROR" (error code: 3102) will occur.

CHAPTER7 COMMUNICATIONS WITH INTELLIGENT FUNCTION MODULE

(1) Intelligent function module

The intelligent function module allows the CPU module to process analog quantity and high speed pulses that cannot be processed by the I/O modules.

For example, the analog-digital conversion module, one of the intelligent function modules, uses analog quantity by converting it into a digital value.

(2) Communication with intelligent function module

The intelligent function module is equipped with a memory (buffer memory) to store the data taken in from or output to external devices.

The CPU modules writes or reads data to or from the buffer memory of the intelligent function module.

7.1 Communications between CPU Module and Intelligent Function Module

The following table shows the communication methods between the CPU module and intelligent function modules and the communication timing.

Table7.1 Communication methods with intelligent function module and the communication timing

Communication methods with intelligent function module		Communication timing					Reference
		Power off → on	CPU module is reset	STOP → RUN ^{*2}	Instruction execution	END processing	
GX Configurator	Initial setting	○	○	○	-	-	Section 7.1.1
	Auto refresh setting	-	-	-	-	○	
Initial device value		○	○	○	-	-	Section 7.1.2
FROM or TO instruction ^{*1}		-	-	-	○	-	Section 7.1.3
Intelligent function module device ^{*1}		-	-	-	○	-	Section 7.1.4
Intelligent function module dedicated instruction ^{*1}		-	-	-	○	-	Section 7.1.5

○ : Executed, - : Not executed

*1: Indicates the program that uses the intelligent function module device, FROM or TO instruction, or the intelligent function module dedicated instruction.

*2: The RUN/STOP/RESET switch is set from STOP → RUN (RUN LED flashes) → STOP → RUN.

Point

- Data, such as initial settings used for communications with intelligent function module, are stored in the CPU module. For the storage location, refer to Section 5.1.
- Power off and then on or reset the CPU module to make the initial setting configured in GX Configurator taken effect.

7.1.1 Initial setting and auto refresh setting by GX Configurator

The initial setting and auto refresh setting can be made by adding in GX Configurator that is supported by the intelligent function module to GX Developer.

After the initial and auto refresh settings, data can be read or written without creating a program for communications with intelligent function modules.

(1) Starting GX Configurator

Select [Tools] → [Intelligent function module utility] → [Start] in GX Developer.

(2) Setting in GX Configurator

An example of initial setting and auto refresh setting for the A/D conversion module Q64AD is used in this section.

(a) Initial setting

The initial setting for the Q64AD offers the following four types.

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

Make the initial settings of the Q64AD in the Initial setting screen in GX Configurator as shown in Figure 7.1.

Setting item	Setting value
CH1 A/D conversion enable/disable setting	Enable
CH1 Sampling process/averaging process setting	Sampling
CH1 Time/number of times specifying	Number of times
CH1 Average time/average number of times setting (Setting range) Time: 2 to 5000 ms Number of times: 4 to 62500 times	4
CH2 A/D conversion enable/disable setting	Enable
CH2 Sampling process/averaging process setting	Sampling
CH2 Time/number of times specifying	Number of times

Details:
Decimal input
Setting range:
4 - 62500

Make text file End setup Cancel

Figure 7.1 Initial setting screen

The initial setting data set in this screen are stored into the intelligent function module parameters of the CPU module.

(b) Auto refresh setting

The CPU module devices for storing the following data can be set in the Auto refresh setting screen.

- Digital output of the Q64AD
- Maximum and minimum values of the Q64AD
- Error codes

Make auto refresh settings of the Q64AD in the Auto refresh setting screen in GX Configurator as shown in Figure 7.2.

Setting item	Module side Buffer size	Module side Transfer word count	Transfer direction	PLC side Device
CH1 Digital output value	1	1	->	
CH2 Digital output value	1	1	->	
CH3 Digital output value	1	1	->	
CH4 Digital output value	1	1	->	
CH1 Maximum value	1	1	->	
CH1 Minimum value	1	1	->	
CH2 Maximum value	1	1	->	
CH2 Minimum value	1	1	->	
CH3 Maximum value	1	1	->	

Figure 7.2 Auto refresh setting screen

The auto refresh setting data set in this screen are stored into the intelligent function module parameters of the CPU module.

(3) Limitation on the number of parameter settings

Limitations are placed on the number of parameters (initial setting and auto refresh setting) set in GX Configurator.

When multiple intelligent function modules are mounted, make setting in GX Configurator so that the number of parameter settings for all intelligent function modules may not exceed the limitation shown in Table7.2.

Table7.2 Maximum number of parameters set in GX Configurator

CPU module	Maximum number of parameter settings	
	Initial setting	Auto refresh setting
Q00UJCPU, Q00UCPU, Q01UCPU	512	256
Q02UCPU	2048	1024
Q03UDCPU, Q04UDHCPU, Q06UDHCPU, Q10UDHCPU, Q13UDHCPU, Q20UDHCPU, Q26UDHCPU, Q03UDECPU, Q04UDEHCPU, Q06UDEHCPU, Q10UDEHCPU, Q13UDEHCPU, Q20UDEHCPU, Q26UDEHCPU	4096	2048

One line in the Auto refresh screen is counted as one parameter setting.

This one line is counted as one parameter setting.
Blanks are not counted.
Add the number of lines on this screen to those of other intelligent function modules.

Figure 7.3 Counting the number of parameter settings

Remark
For details of GX Configurator, refer to the manual for the intelligent function module used.
.....

7.1.2 Initial setting by initial device value

(1) Initial device value

Using an initial device value (☞ Section 6.25) allows the initial setting of the intelligent function module without a program.

The set initial device values are written from the CPU module to the intelligent function module when the CPU module is powered off and then on, is reset, or set from STOP to RUN.

(2) Setting initial device values

Use GX Developer to set the following.

- Set the device data of the intelligent function module (☞ Section 9.5.1) used as the initial device value to the device memory.
- In the initial device value setting, specify the device range of the intelligent function module used as the initial device value.

7.1.3 Communications with the FROM and TO instructions

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.

Remark

- For details of the FROM and TO instructions, refer to the following.
☞ QCPU Programming Manual (Common Instructions)
- For details of the buffer memory of the intelligent function module, refer to the following.
☞ Manual for the intelligent function module used

7.1.4 Communications using the intelligent function module device

(1) Intelligent function module device

The intelligent function module device (Figure 7.4) represents the buffer memory of the intelligent function module as one of the CPU module devices.

The data stored in the buffer memory of the intelligent function module can be treated by the sequence instruction as well as the device memory.

A programming example is given below for the case where "100" is written to the buffer memory of an address 0 of the intelligent function module whose I/O number is X/Y20 to X/Y2F.

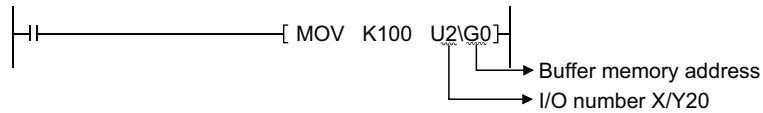


Figure 7.4 Use of the intelligent function module device

(2) Difference from the FROM and TO instructions

The intelligent function module device processes data read from the intelligent function module with one instruction as the device can be treated as one of the CPU module devices.

A program example is given below for the case where data read from the intelligent function module are added and then the result is stored in D2.

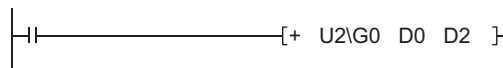


Figure 7.5 Application example of the intelligent function module device

This reduces the number of steps in whole program.

The instruction processing time indicates the time required for instruction execution and accessing the intelligent function module.

Point

The intelligent function module device accesses the intelligent function module every time when an instruction is executed. When writing or reading buffer memory data using multiple intelligent function module devices in a sequence program, write or read the data with the FROM or TO instruction in one location of the program.

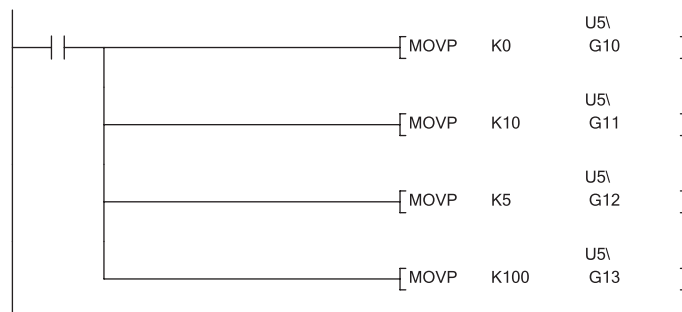


Figure 7.6 Writing using multiple intelligent function module devices

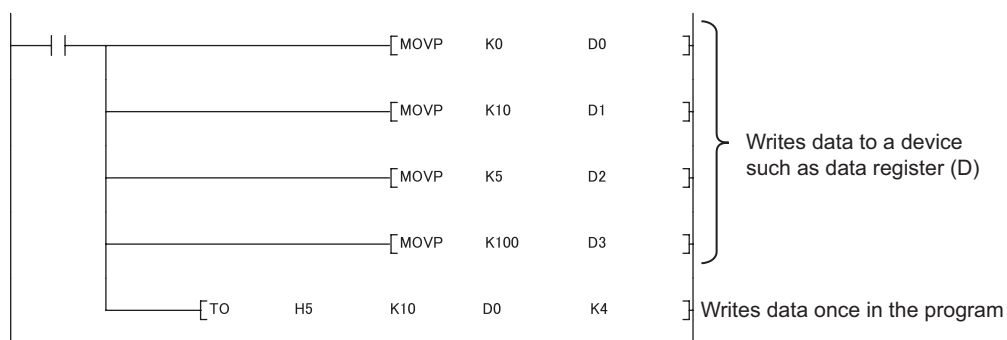


Figure 7.7 Writing using the TO instruction

7.1.5 Communications using the intelligent function module dedicated instruction

(1) Intelligent function module dedicated instruction

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

(a) Example with the serial communication module dedicated instruction (OUTPUT instruction)

The OUTPUT instruction allows communications with external device by nonprocedural protocol regardless of the buffer memory address of the serial communication module.

[Transmission by the OUTPUT instruction]

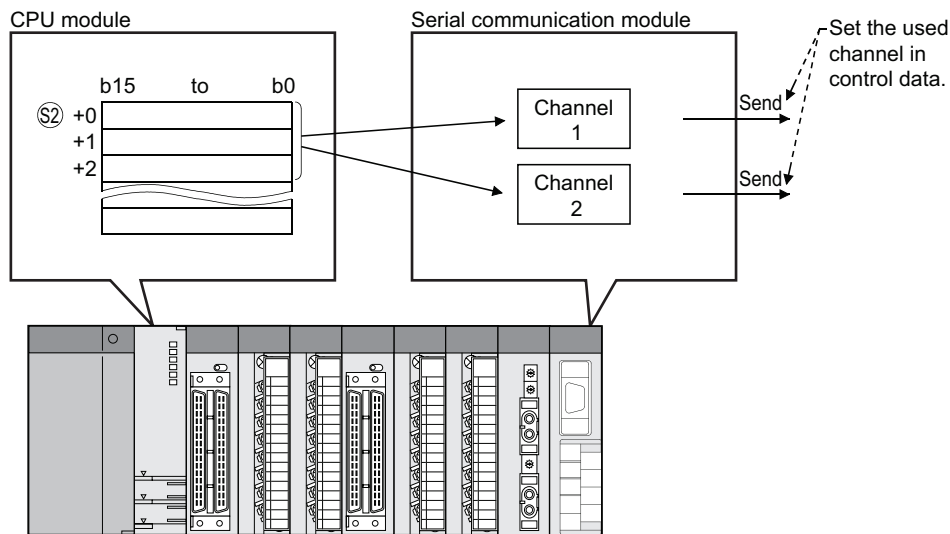


Figure 7.8 Communications with external devices using the OUTPUT instruction

(2) Processing of the intelligent function module dedicated instruction

Some of intelligent function module dedicated instructions can specify the completion device.

This completion device turns on for one scan when an instruction execution is completed.

When using multiple intelligent function module dedicated instructions to one intelligent function module, execute the dedicated instructions one by one after the completion device turns on.

(3) Precautions

(a) When the CPU module is set from RUN to STOP before the completion device turns on

When a intelligent function module dedicated instruction is executed and the CPU module is set from RUN to STOP before the completion device turns on, the completion device will not turn on until the CPU module is set to RUN and then finishes one scan.

(b) Available range

The intelligent function module dedicated instructions are not applicable to the intelligent function modules mounted on the remote I/O stations of CC-Link IE field network or on the remote I/O stations of MELSECNET/H.

The instructions are applicable to the intelligent function modules mounted on the base unit or extension base units.

Remark

For intelligent function module dedicated instructions and completion devices, refer to the following.

 Manual for the intelligent function module used

CHAPTER8 PARAMETERS

This chapter describes the parameters required to be set for configuring a programmable controller system.

(1) Parameter types

The following parameters are provided for CPU module setting.

- PLC parameters (☞ Section 8.1)

These parameters are set when a programmable controller is used.

- Network parameters (☞ Section 8.2)

These parameters are set when a programmable controller module is used in combination with any of CC-Link IE controller network modules, MELSECNET/H modules, Ethernet modules, and CC-Link modules.

- Remote password (☞ Section 8.3)

Parameters are set when the remote password function is used for an Ethernet module, serial communication module, modem interface module, or Built-in Ethernet port QCPU.

(2) Parameter setting method

Use GX Developer.

For the setting details, refer to the following.

☞ GX Developer Version 8 Operating Manual

Point

Grayed-out (unselectable) parameter settings in GX Developer are not available since the corresponding function is not supported.

Remark

Each parameter number shown in the tables in this chapter is stored in the special register (SD16 to SD26) when an error occurs in parameter setting.
Identify the parameter error location from the parameter number.

For the list of the special register (SD16 to SD26), refer to CHAPTER 12.
For the parameter updating procedure, refer to CHAPTER 11.

8.1 PLC Parameters

This section provides the list of PLC parameters and describes parameter details.

(1) PLC name

A label and a comment for the CPU module are set.

The settings will be displayed in the list for the find CPU function.  Note8.1

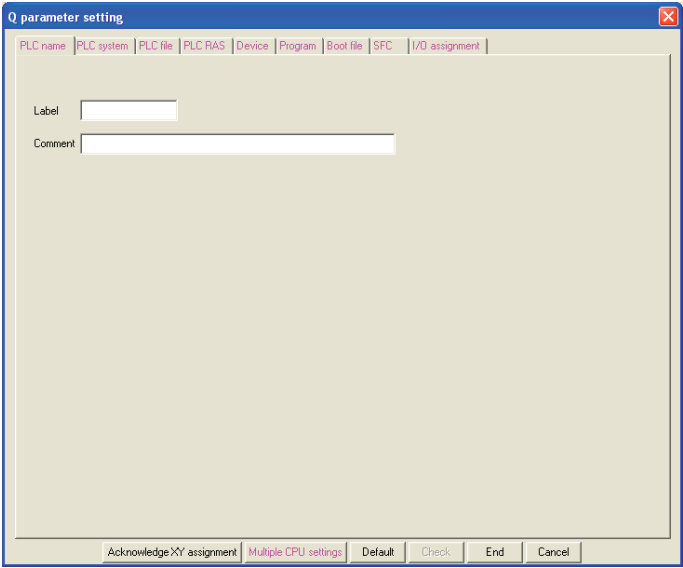


Figure 8.1 PLC name

Table8.1 PLC name setting list

Item	Parameter No.	Description	Setting range	Default	Reference
Label	0000 _H	Set a label (name, application) for the CPU module.	Up to 10 characters	Blank	-
Comment	0001 _H	Set a comment for the CPU module label.	Up to 64 characters	Blank	-

 Note8.1  Universal

The Universal model QCPUs other than the Built-in Ethernet port QCPU do not support the find CPU function.

(2) PLC system

Parameters required for use of the CPU module are set.

Figure 8.2 PLC system

Table 8.2 PLC system setting list

Item		Parameter No.	Description	Setting range	Default	Reference
Timer limit setting	Low speed	1000 _H	Set the time limit for the low speed timer or high speed timer.	1ms to 1000ms (in increments of 1ms)	100ms	Section 9.2.10
	High speed			0.01ms to 100.0ms (in increments of 0.01ms)	10.0ms	Section 9.2.10
RUN-PAUSE conctcs	RUN	1001 _H	Set the contacts that control RUN/PAUSE of the CPU module. Setting of only the PAUSE contact is not allowed.	X0 to 1FFF	Blank	Section 6.6.1
	PAUSE					Section 6.6.2
Latch data backup operation valid contact		1014 _H	Set the valid contact device No. for backup of latch data to the standard ROM.	X, M, B	Blank	Section 6.28
Remote reset		1002 _H	Select whether to allow the remote reset from GX Developer.	Selected/deselected	Deselected	Section 6.6.3
Output mode at STOP to RUN		1003 _H	Set the status of the outputs (Y) when the operating status is switched from STOP to RUN.	Previous state, Recalculate (output is 1 scan later)	Previous state	Section 6.4
Intelligent function module setting (Interrupt pointer setting)		100A _H	Assign the interrupt pointers (I50 to I255) and set the start I/O number and start SI number of each intelligent function module.	- Start I/O No. - Start SI No. - I50 to 255	Blank	Section 9.11
Module synchronization		100C _H	Select whether to synchronize CPU module startup with intelligent function module startup.	Selected/deselected	Deselected	-

(To the next page)

Table8.2 PLC system setting list (continued)

Item		Parameter No.	Description	Setting range	Default	Reference
Common pointer No.		1005 _H	Set the start number of common pointers.	P0 to 4095	Blank	Section 9.10.2
Points occupied by empty slot		1007 _H	Set the number of points for empty slots on the main/extension base units.	0, 16, 32, 64, 128, 256, 512, or 1024 points	16 points	Section 3.2.2
System interrupt settings	Fixed scan interval (n: 28 to 31)	1008 _H	Set each execution interval for the interrupt pointers (I28 to I31).	0.5ms to 1000ms (in increments of 0.5ms)	• I28: 100.0ms • I29: 40.0ms • I30: 20.0ms • I31: 10.0ms	Section 9.11
Interrupt program/Fixed scan program setting		1008 _H	Enable or disable high speed execution of interrupt programs or fixed scan execution type programs.	Selected/deselected	Deselected	Section 2.2.3, 2.3.4
Service processing setting		1013 _H	Select any of the following options. • Execute the process as the scan time proceeds. • Specify service process time. • Specify service process execution counts. • Execute it while waiting for constant scan time setting.	• 1 to 99% (in increments of 1%) • 1 to 10 (in increments of 1 time) • 0.2 to 1000ms (in increments of 0.1ms) • Blank	Execute the process as the scan time proceeds.: 10%	Section 6.24.1
PLC module change setting (PLC module change setting)		1017 _H	Set items required when performing the CPU module change with memory card function.	• Backup start setup contact • Backup start contact • Backup target data • Title setting	Blank	Section 6.30

(3) PLC file

Parameters required for the files used in the CPU module are set.

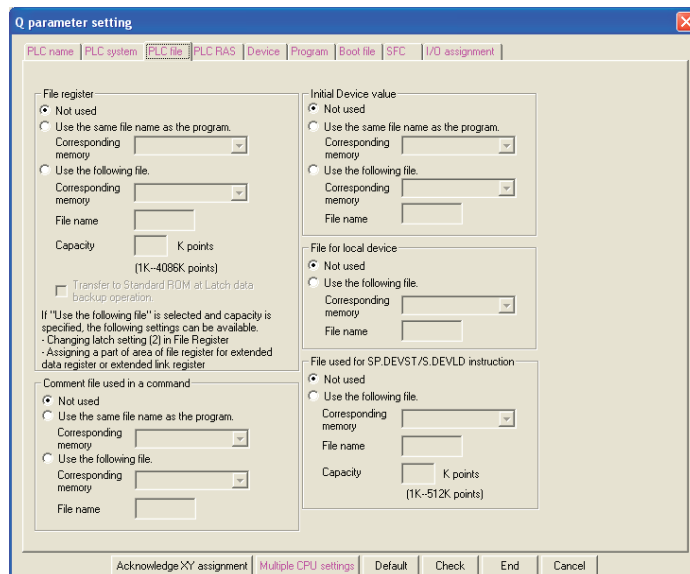


Figure 8.3 PLC file

Table8.3 PLC file setting list

Item	Parameter No.	Description	Setting range	Default	Reference
File register ^{*1}	1100 _H	Set a file for the file register used in the program.	- Not used - Use the same file name as the program. - Use the following file.	Not used	Section 9.7
Transfer to Standard ROM at Latch data backup operation ^{*1}	1104 _H	Select whether to batch-transfer the data in the file register at the time of latch data backup to the standard ROM.	Checked/unchecked	Not selectable	Section 6.27
Comment file used in a command	1101 _H	Set a file for device comments used in the program.	- Not used - Use the same file name as the program. - Use the following file.	Not used	-
Initial Device value	1102 _H	Set a file for initial values of the devices used for the CPU module.	- Not used - Use the same file name as the program. - Use the following file.	Not used	Section 6.25
File for local device ^{*1}	1103 _H	Set a file for local devices used in the program.	- Not used - Use the following file.	Not used	Section 9.14.2
File use for SP.DEVST/S.DEVLD instruction	1105 _H	Set a device data file used for writing to or reading from the standard ROM.	- Not used - Use the following file.	Not used	Section 6.28

^{*1}: Not available for the Q00UJCPU.

(4) PLC RAS (1)

Parameters required for performing the RAS functions are set.

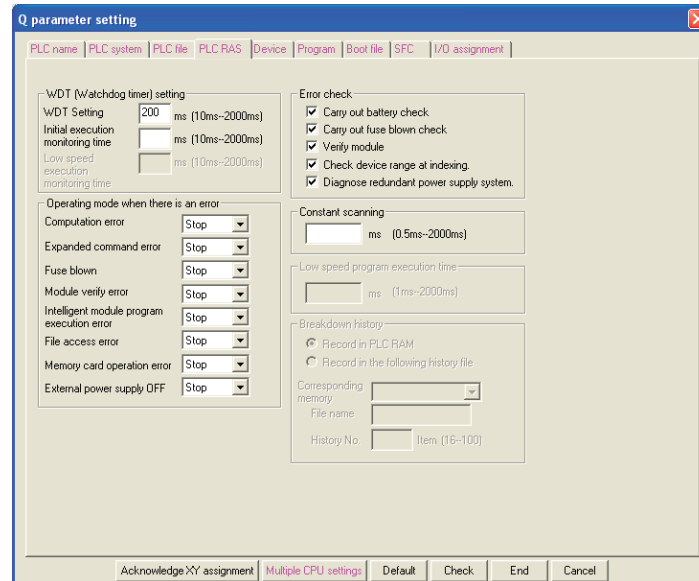


Figure 8.4 PLC RAS (1)

Table8.4 PLC RAS (1) setting list

Item		Parameter No.	Description	Setting range	Default	Reference
WDT (Watchdog timer) setting	WDT Setting	3000 _H	Set a watchdog timer value for the CPU module.	10ms to 2000ms (in increments of 10ms)	200ms	Section 6.16
	Initial execution monitoring time		Set a watchdog timer value in the case of using an initial execution type program.	10ms to 2000ms (in increments of 10ms)	Blank	Section 2.3.1
Operating mode when there is an error	Computation error	3002 _H	Set the operation mode of the CPU module when an error is detected.	Stop/Continue	Stop	Section 6.17
	Expanded command error*1					
	Fuse blown					
	Module verify error					
	Intelligent module program execution error					
	File access error					
	Memory card operation error*3					
	External power supply OFF*1					
Error check	Carry out battery check	3001 _H	Enable or disable detection of the specified error.	Selected/deselected	Deselected	Section 6.17
	Carry out fuse blown check					
	Verify module					
	Check device range at indexing.					
	Diagnose redundant power supply system.*2					
Constant scanning		3003 _H	Set a constant scan time value.	0.5ms to 2000ms (in increments of 0.5ms)	Blank	Section 6.2

*1: These items are provided for future expansion.

*2: When selecting this item for the Q02UCPU, Q03UDCPU, Q04UDHCPU, Q06UDHCPU, Q13UDHCPU, or Q26UDHCPU, check the versions of the CPU module and GX Developer. (Appendix 2)

*3: Not available for the Q00UJCPU, Q00UCPU, and Q01UCPU.

(5) PLC RAS (2) Note8.2

Parameters required for performing the RAS functions are set.

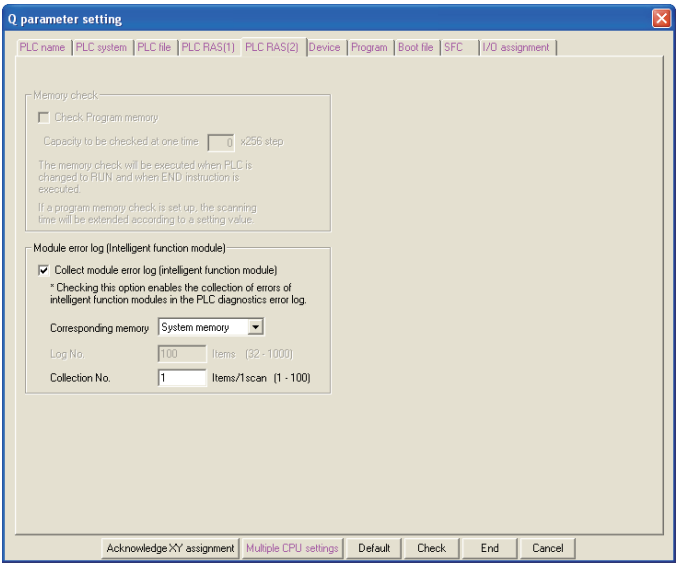


Figure 8.5 PLC RAS (2)

Table8.5 PLC RAS (2) setting list

Item		Parameter No.	Description	Setting range	Default	Reference
Module error log (Intelligent function module)	Collect module error log (intelligent function module)	300A _H	Set whether to collect module errors.	Selected/deselected	Selected	Section 6.32
	Corresponding memory		Select a storage location.	• System memory • Standard RAM	System memory	
	Log No.		Set the number of collected errors only when the errors are stored in the standard RAM.	32 to 1000	40/100	
	Collection No.		Set the number of collected errors in one scan.	• Stored in system memory: 1 to 100 • Stored in standard RAM: 1 to 128	1	

 Note8.2 

The setting screen differs between GX Works2 and GX Developer. For the setting of the module error log function, refer to Section 6.32(5)

(6) Device

Number of points, latch range, and local device range are set for each device.

Figure 8.6 Device

Table8.6 Device setting list

Item	Parameter No.	Description	Setting range	Default	Reference
Device points ^{*1}	2000 _H	Set the number of device points that is appropriate to the system.	X (8K), Y (8K), and S (8K)^{*3} are fixed. Setting is available within the range of 29K words in total, including the above fixed points (1.5K words). • One device: Up to 32K points • Total bit device points: Up to 64K points	• X: 8K • Y: 8K • M: 8K • L: 8K^{*2} • B: 8K • F: 2K • SB: 2K • V: 2K • S: 8K^{*3 *7} • T: 2K • ST: 0K • C: 1K • D: 12K • W: 8K • SW: 2K	Section 9.1
Latch (1) start/end (Latch clear valid) ^{*2}	2001 _H	Set a latch range (start and end device numbers), which can be cleared by remote latch clear operation.	Setting is available for only one range for each of B, F, V, T, ST, C, D, and W devices.	Blank	Section 3.4, 6.3
Latch (2) start/end (Latch clear invalid) ^{*2}	2002 _H	Set a latch range (start and end device numbers), which cannot be cleared by remote latch clear operation.	Setting is available for only one range for each of L, B, F, V, T, ST, C, D, and W devices.	Blank	Section 3.4, 6.3
Local device start/end ^{*6}	2003 _H	Set a range (start and end device numbers), which is used for a local device.	Setting is available for only one range for each of M, V, T, ST, C, D, and Z devices. ^{*5}	Blank	Section 9.14.2

(To the next page)

Table8.6 Device setting list (continued)

Item	Parameter No.	Description	Setting range	Default	Reference
File register extended setting ^{*6}	2000 _H	Set points for the file register (ZR), extended data register (D), and extended link register (W).	Point assignment to the file register (ZR), extended data register (D), or extended link register (W). Assign part of the points of the file register to the extended data register and extended link register.	Blank	Section 9.1, 9.8
	2004 _H	Set a latch range (start and end device numbers), which can be cleared by remote latch clear operation.	Each latch range for the file register (ZR), extended data register (D), or extended link register (W).	Blank	Section 3.4, 6.3
	2005 _H	Set a latch range (start and end device numbers), which cannot be cleared by remote latch clear operation.	Each latch range for the file register (ZR), extended data register (D), or extended link register (W).	Blank	Section 3.4, 6.3
	2000 _H	Select Z or ZZ device for 32-bit indexing.	Z0 to Z18 (when using Z device)	Use Z	MELSEC-Q/L Programming Manual (Common Instruction)

*1: When changing the device points, new setting must not exceed the refresh ranges of network modules or the auto refresh ranges of intelligent function modules.

If a new device point setting exceeds the corresponding device range, the data may be written to another device or an error may occur.

*2: When a device latch range is set, the scan time increases.

When latching a device, consider the increase in the scan time. (➡ Section 10.1.2)

*3: The points for the step relay can be changed to 0 if the serial number (first five digits) of the Universal model QCPU is "10042" or later.

*4: Available only when the serial number (first five digits) of the Universal model QCPU is "10042" or later.

*5: When using the index register as a local device for the Q03UDCPU, Q04UDHCPU, Q06UDHCPU, Q13UDHCPU, Q26UDHCPU, or

Built-in Ethernet port QCPU, check the versions of the CPU module and GX Developer. (➡ Appendix 2)

*6: Not available for the Q00UJCPU.

*7: For the Universal model QCPU whose serial number (first five digits) is "12052" or later, a step relay can be set in increments of 1k point and up to the following points. (➡ Appendix 2)

- Q00UJCPU, Q00UCPU, Q01UCPU, and Q02UCPU: 8192 points
- Universal model QCPU other than the Q00UJCPU, Q00UCPU, Q01UCPU, and Q02UCPU: 16384 points

(7) Program

File names and execution types (execution conditions) are set for each program when two or more programs are written to the CPU module.

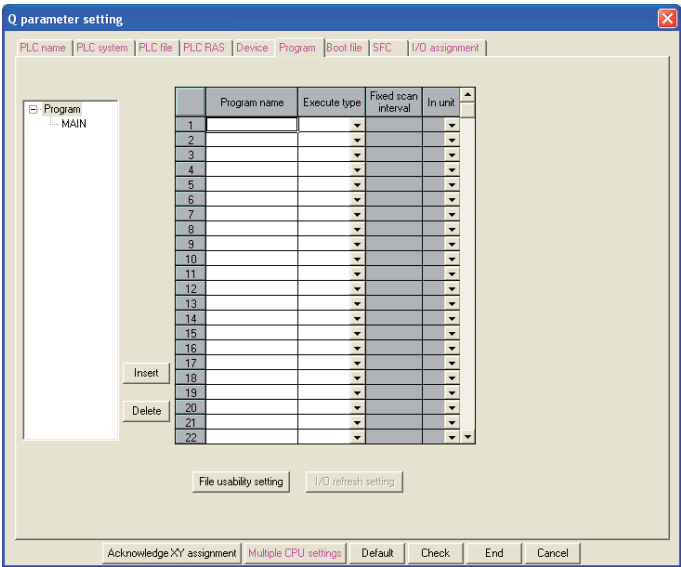


Figure 8.7 Program

Table8.7 Program setting list

Item	Parameter No.	Description	Setting range	Default	Reference
Program setting	7000H	When writing two or more programs to the CPU module, set a file name and execution type (execution condition) of each program. Also, set a fixed scan interval (execution interval of the fixed scan execution type program).	• Program name • Execute type (fixed scan interval when the fixed scan execution is selected) • File usability setting*1	Blank	Section 2.3

*1: Available for local devices only.
When using the file usability setting, check the versions of the CPU module and GX Developer. (Appendix 2)
The setting is not available for the Q00UJCPU.

(8) Boot file

Parameters required for a boot from a memory card are set.

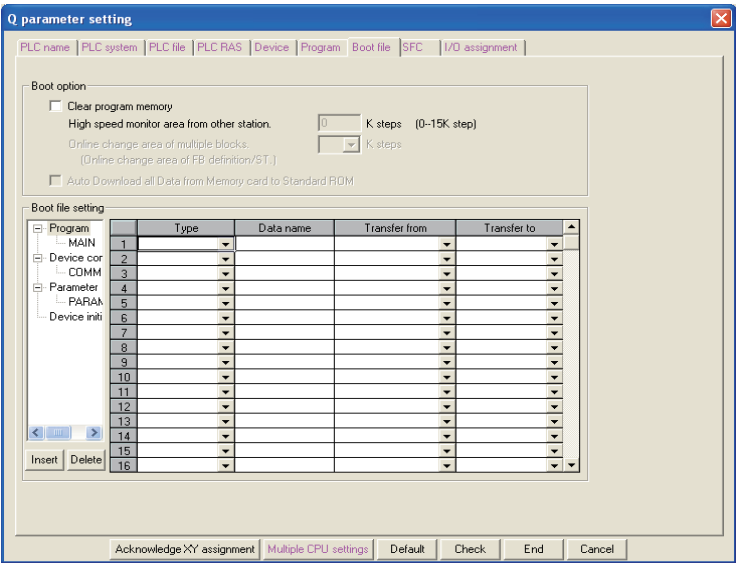


Figure 8.8 Boot file

Table8.8 Boot file setting list

Item		Parameter No.	Description	Setting range	Default	Reference
Boot option	Clear program memory	7000 _H	Select whether to clear the program memory at the time of boot.	Selected/deselected	Deselected	Section 5.1.8
Boot file setting			Set the type and data name of the boot file, and transfer source drive for boot operation.	Type, "Data name, and Transfer from (the transfer target drive (Transfer to) is automatically set in the program memory.)	Blank	

(9) SFC

The mode and conditions for starting an SFC program, and the output mode in the case of a block stop are set.

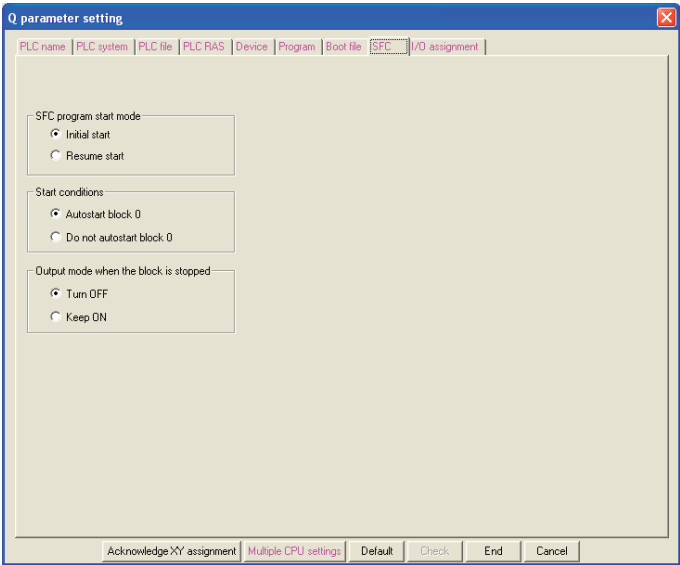


Figure 8.9 SFC

Table8.9 SFC setting list

Item	Parameter No.	Description	Setting range	Default	Reference
SFC program start mode	8002 _H	Set the mode and conditions for starring an SFC program, and also set the output mode in case a program block is stopped.	MELSEC-Q/L/QnA Programming Manual (SFC)	Initial start	-
Start conditions	8003 _H			Autostart block 0	
Output mode when the block is stopped	8006 _H			Turn OFF	

(10)I/O assignment

The mounting status of each module in the system is set.

Q parameter setting

PLC name | PLC system | PLC file | PLC RAS | Device | Program | Boot file | SFC | I/O assignment

I/O Assignment(*)

Slot	Type	Model name	Points	StartXY
0	PLC			
1	0(*-0)			
2	1(*-1)			
3	2(*-2)			
4	3(*-3)			
5	4(*-4)			
6	5(*-5)			
7	6(*-6)			

Assigning the I/O address is not necessary as the CPU does it automatically.
Leaving this setting blank will not cause an error to occur.

Base setting(*)

	Base model name	Power model name	Extension cable	Slots
Main				
Ext.Base1				
Ext.Base2				
Ext.Base3				
Ext.Base4				
Ext.Base5				
Ext.Base6				
Ext.Base7				

Base mode:
☒ Auto
☐ Detail

8 Slot Default
12 Slot Default

(*)Settings should be set as same when using multiple CPU.

Import Multiple CPU Parameter Read PLC data

Acknowledge XY assignment Multiple CPU settings Default Check End Cancel

Figure 8.10 I/O assignment

Table8.10 I/O assignment setting list

Item		Parameter No.	Description	Setting range	Default	Reference
I/O Assignment	Type	0400 _H	Set the type of the mounted module.	- CPU No.2 to No.4: No.n/Empty (Set "CPU (Empty)" for the slot where no CPU module is mounted.) - Empty, Input, Hi input, Output, Intelli., I/O mix, or Interrupt	Blank	Section 4.2.2
	Model name		Set the model name of the mounted module. (Entered at user's discretion. Do not use the one for the CPU module.)	Up to 16 characters		
	Points		Set the number of points assigned to each slot.	0, 16, 32, 48, 64, 128, 256, 512, or 1024 points		
	Start XY		Set the start I/O number of each slot.	0 _H to FF0 _H		
Base setting	Base model name	0401 _H	Set the model name of the main base unit or extension base unit. (Entered at user's discretion. Do not use the one for the CPU module.)	Up to 16 characters	Blank	Section 4.1.2
	Power model name		Set the model name of the power supply module on the main base unit or extension base unit. (Entered at user's discretion. Do not use the one for the CPU module.)	Up to 16 characters		
	Extension cable		Set the extension cable name. (Entered at user's discretion. Do not use the one for the CPU module.)	Up to 16 characters		
	Slots		Set the number of slots of the main base unit or extension base unit. This setting must be done for all base units.	2, 3, 5, 8, 10, or 12		

(To the next page)

Table8.10 I/O assignment setting list (continued)

Item		Parameter No.	Description	Setting range	Default	Reference
Switch setting		0407 _H	Set various switches of an intelligent function module.	Refer to the manual for the intelligent function module used.	Blank	Section 6.10
Detailed setting	Error time output mode	0403 _H	Set whether to clear or hold the output in case of a stop error in the control CPU.	Clear/Hold	Clear	Section 6.8
	H/W error time PLC operation mode	4004 _H	Set whether to stop or continue the operation of the control CPU in case of a hardware failure of the intelligent function module.	Stop/Continue	Stop	Section 6.9
	I/O response time	0405 _H	Set a response time for the input module, high-speed input module, I/O combined module, or interrupt module.	- Input or I/O mix: 1ms, 5ms, 10ms, 20ms, or 70ms - Hi input or Interrupt: 0.1ms, 0.2ms, 0.4ms, 0.6ms, or 1ms	- Input or I/O mix: 10ms - Hi input or Interrupt: 0.2ms	Section 6.7
	Control PLC	0406 _H	Set the control CPU for the input/output modules and intelligent function module.	No.1, No.2, No.3, or No.4	No.1	QCPU User's Manual (Multiple CPU System)

(11) Serial Note8.3

The transmission speed, sum check, transmission wait time, and RUN write setting for using the serial communication function of the CPU module are set.

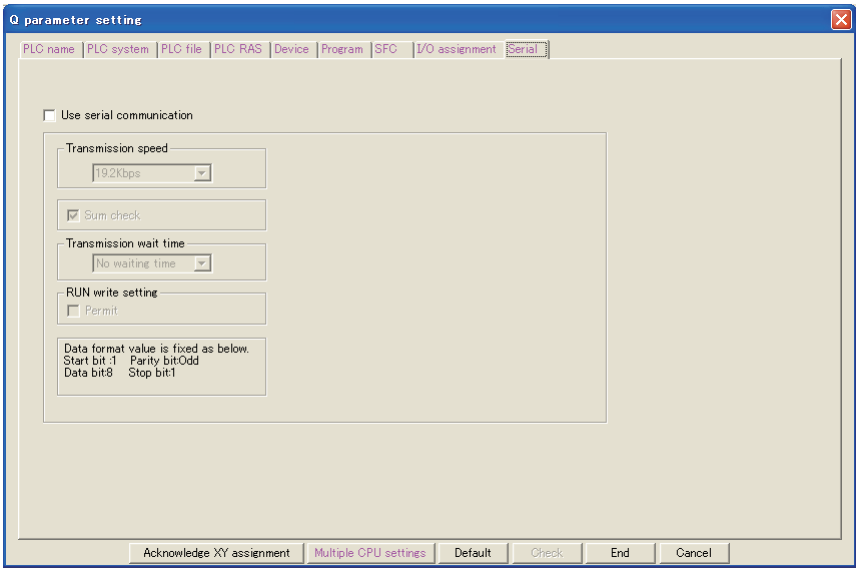


Figure 8.11 Serial

Table8.11 Serial setting list

Item	Parameter No.	Description	Setting range	Default	Reference
Use serial communication	100EH	Select the item when using the serial communication function.	Selected/deselected	Deselected	Section 6.23
Transmission speed		Set a transmission speed for data communication with the external device.	9.6kbps, 19.2kbps, 38.4kbps, 57.6kbps, 115.2kbps	19.2kbps	
Sum check		Set whether to add a sum check code to a message sent or received when using the serial communication function, according to the specifications of the external device.	Selected/deselected	Selected	
Transmission wait time		Set a period of waiting time on the Basic model QCPU side in case the CPU module cannot receive data immediately after the external device sends data.	No waiting time/10ms to 150ms (in increments of 10ms)	No waiting time	
RUN write setting		Enable or disable writing of data from the external device to the running CPU module.	Selected/deselected	Deselected	



Note8.3

Universal

When using the serial communication function for the Q02UCPU, check the versions of the CPU module and GX Developer. ( Appendix 2)
The Q03UDCPU, Q04UDHCPU, Q06UDHCPU, Q10UDHCPU, Q13UDHCPU, Q20UDHCPU, Q26UDHCPU, and Built-in Ethernet port QCPU do not support the serial communication function.

(12) Acknowledge XY assignment

The parameters set in the I/O assignment, Ethernet/CC IE/MELSECNET setting, and CC-Link setting can be confirmed.

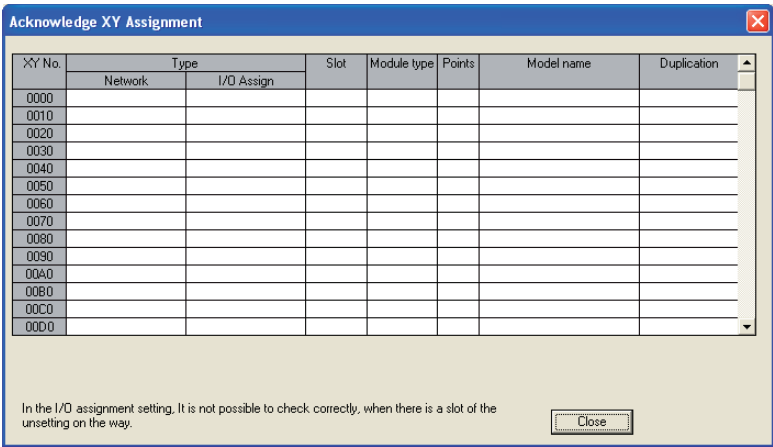


Figure 8.12 Acknowledge XY assignment

Table8.12 Acknowledge X/Y assignment list

Item	Parameter No.	Description	Setting range	Default	Reference
X/Y assignment	-	The data set in the I/O assignment, Ethernet/CC IE/ MELSECNET setting, and CC-Link setting can be checked.	-	-	-

(13) Multiple CPU settings Note8.4

Parameters required for configuring a multiple CPU system are set.

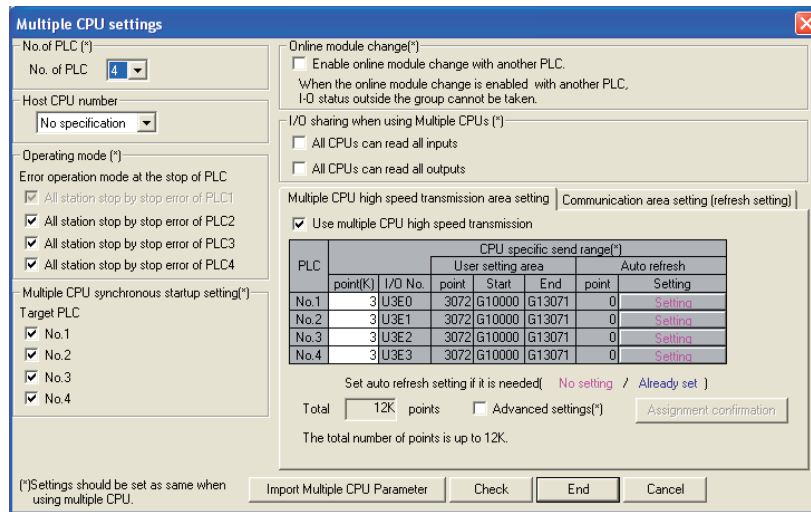


Figure 8.13 Multiple CPU settings

Table8.13 Multiple CPU setting list

Item	Parameter No.	Description	Setting range	Default	Reference
No. of PLC	0E00 _H	Set the number of CPU modules used in a multiple CPU system.	1 to 4	1	QCPU User's Manual (Multiple CPU System)
Host CPU number ^{*1}	E00C _H	Set a CPU number for which the multiple CPU setting parameters are set. (Set the number of the connected CPU module.)	PLC No.1 to No.4	Blank	
Operating mode	0E01 _H	Select the multiple CPU system operation to be performed in case a stop error occurs in any of CPU No.2 to No.4. When CPU No.1 results in a stop error, the multiple CPU system stops. (Fixed)	Selected/ deselected	All items selected	
Multiple CPU synchronous startup setting ^{*1}	E00B _H	Enable or disable synchronous startup of the CPU modules on the multiple CPU system.	Mp.1 to No.4	All items selected	

(To the next page)

Table8.13 Multiple CPU setting list (continued)

Item			Parameter No.	Description	Setting range	Default	Reference
Online module change* ¹			E006 _H	Enable or disable the online module change in the multiple CPU system. (When enabled, the CPU module cannot read the I/O data outside the specified group.)	Selected/deselected	Deselected	QQCPU Useris Manual (Multiple CPU System)
I/O sharing when using Multiple CPUs		All CPUs can read all inputs	0E04 _H	Select whether to read the input data of the input modules or intelligent function modules controlled by another CPU.	Selected/deselected	Deselected	
		All CPUs can read all outputs		Select whether to read the output data of the output modules controlled by another CPU.	Selected/deselected	Deselected	
Multiple CPU high speed transmission area setting* ¹	CPU specific send range		E008 _H	Set the size of the multiple CPU high-speed transmission area that is assigned to each CPU module of the multiple CPU system.	Simple setting: 0 to 12K (in increments of 1K points) Advanced setting: 0 to 16K (in increments of 0.5K points)	3K points	
	Auto refresh setting	Refresh setting	E009 _H E00A _H	Set an area range for data communication performed with the auto refresh function in the user area of the multiple CPU high-speed transmission area.	Available devices* ² : X, Y, M, L, B, D, W, R, ZR, SM, SD, SB, and SW	-	
Communication area setting (refresh setting)			E002 _H E003 _H	In the multiple CPU system, data are transferred by auto refresh among respective CPU modules. Set the devices to be written or read and their points.	[Set starting devices for each CPU] Selected/deselected	Deselected	
					[CPU specific send range] 0 to 2048 points (in increments of 2 points) per CPU Up to 8K points (8192 points) per system	Blank	
					[PLC side device] B, M, Y, D, W, R, or ZR Occupies the device of the points set for the send range and starting from the specified device number. • One point in the send range equals 16 points in B, M, or Y. • One point in the send range equals one point in D, W, R, or ZR.		

*1: Not available for the Q00UCPU, Q01UCPU, and Q02UCPU.

*2: SM, SD, SB, and SW are valid only when they are selected as send devices.

(14) Built-in Ethernet port Note8.5

Parameters required for use of the built-in Ethernet port are set.

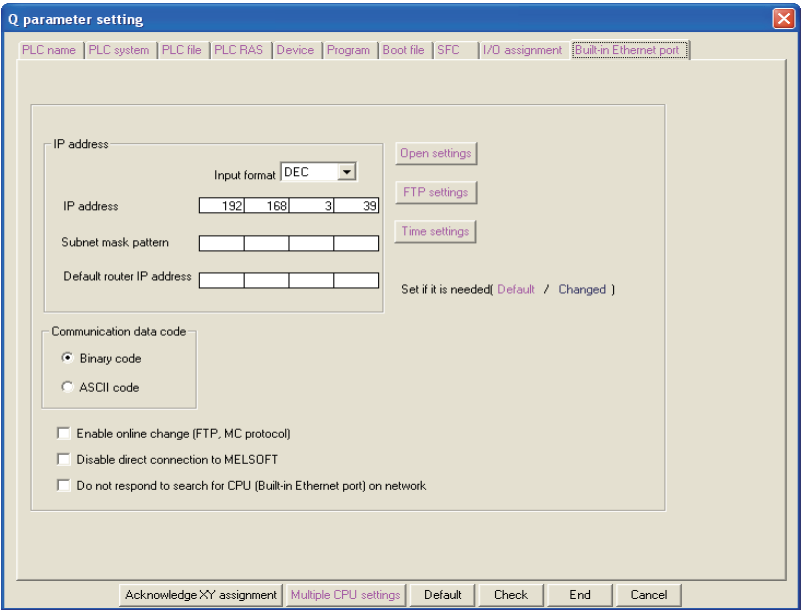


Figure 8.14 Built-in Ethernet port

Table8.14 Built-in Ethernet port setting list

Item	Parameter No.	Description	Setting range	Default	Reference
IP address	1016 _H	• IP address: Enter the IP address of the QnUDE(H)CPU. • Subnet mask pattern: Enter the subnet mask pattern when using a router. • Default router IP address: Enter the IP address of the router.	• IP address: 0.0.0.1 to 223.255.255.254 (00000001_H to 0DFFFFFFE_H) • Subnet mask pattern: Blank or 192.0.0.0 to 255.255.255.252 (0C000000_H to 0FFFFFFC_H) • Default router IP address: Blank or 0.0.0.1 to 223.255.255.254 (00000001_H to 0DFFFFFFE_H)	• IP address: 192.168.3.39 • Subnet mask pattern: Blank • Default router IP address: Blank	QnUCPU User's Manual(Com munication via Built-in Ethernet Port)
Communication data code		Select the code for MC protocol communication.	Binary code/ASCII code	Binary code	

(To the next page)

Table8.14 Built-in Ethernet port setting list (continued)

Item	Parameter No.	Description	Setting range	Default	Reference
Open settings	1016 _H	Set data when using MC protocol or socket communication.	-	Blank	QnUCPU User's Manual (Communication via Built-in Ethernet Port)
FTP settings		Set data when using the file transfer function (FTP).	-	Blank	
Time settings		Set data when using the time setting function.	-	Blank	
Enable online change (FTP, MC protocol)		Enable or disable writing data in devices or files to the running CPU module when MC protocol or FTP is used.	Selected/deselected	Deselected	
Disable direct connection to MELSOFT		Enable or disable direct connection to MELSOFT. To enhance the security with the remote password setting, check it to disable it.	Selected/deselected	Deselected	
Do not respond to search for CPU (Built-in Ethernet port) on network		Checking this box disables response to the find CPU function of the MELSOFT connection. To enhance the security, check it to disable this.	Selected/deselected	Deselected	

8.2 Network Parameters

This section provides the list of network parameters and describes parameter details.

■ Symbols, M and N, used in the "Parameter No." column

M and N in "Parameter No." in this section denote the following:

- N: Indicates the module number.
- M: Indicates the network type.

Table8.15 For CC-Link IE and MELSECNET/H ( (1), (2), (3) in this section)

M	Network type
1H	CC IE Control (Control station), MELSECNET/H mode (Control station), MELSECNET/H Extended mode (Control station), MELSECNET/10 mode (Control station)
2H	CC IE Control (Normal station), MELSECNET/H mode (Normal station), MELSECNET/H Extended mode (Normal station), MELSECNET/10 mode (Normal station)
5H	MELSECNET/H (Remote master station)
8H	CC IE Field (master station)
9H	CC IE Field (local station)
AH	MELSECNET/H (Standby station)
BH	MELSECNET/H mode multiplexed remote I/O network master station
DH	MELSECNET/H mode multiplexed remote I/O network sub-master station (when no parameter is set)
EH	MELSECNET/H mode multiplexed remote I/O network sub-master station (when parameters are set)

Table8.16 For CC-Link setting ( (5) in this section)

M	Network type
0H	Master station
1H	Local station
2H	Standby master station

(1) CC-Link IE controller network setting

Network parameters for the CC-Link IE controller network are set.

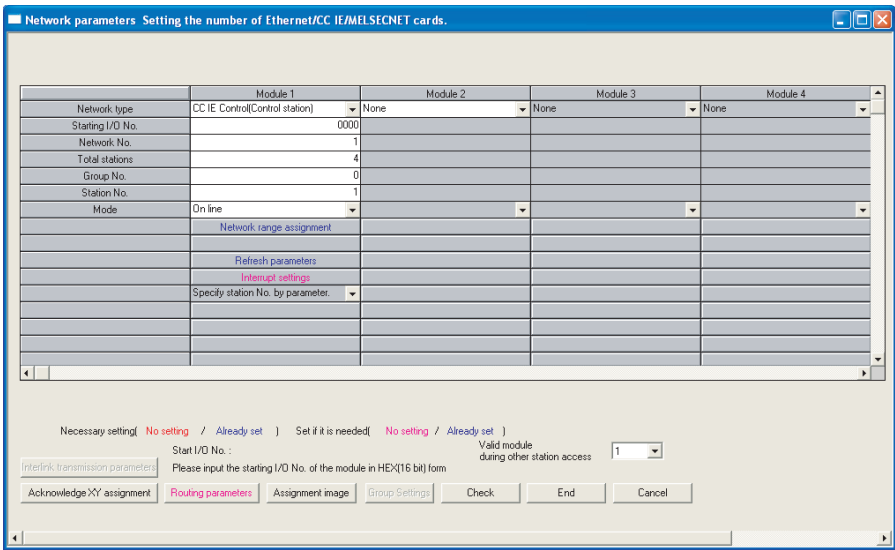


Figure 8.15 Setting the number of Ethernet/CC IE/MELSECNET cards (CC-Link IE controller network setting)

Table8.17 CC-Link IE controller network setting list

Item	Parameter No.	Description	Setting range	Default	Reference
Number of modules on CC-Link IE controller network	A000 _H	Set network parameters for the CC-Link IE controller network.	Refer to the manual for the CC-Link IE controller network.	-	-
Starting I/O No.	ANM0 _H				
Network No.					
Total stations					
Station No.					
Group No.	0Amn _H				
Mode	ANM0 _H				
Refresh parameters	ANM1 _H				
Common parameters	ANM2 _H				
Station inherent parameters	ANM3 _H				
Interlink transmission parameters	A002 _H				
Routing parameters	5003 _H				

(2) CC-Link IE Field network setting

Network parameters for the CC-Link IE Field network are set.

Figure 8.16 Setting the number of Ethernet/CC IE/MELSECNET cards (CC-Link IE Field network setting)

Table8.18 CC-Link IE Field network setting list

Item	Parameter No.	Description	Setting range	Default	Reference
Network Type	A080 _H	Set network parameters for the CC-Link IE Field network.	Refer to the manual for the CC-Link IE Field network.	-	-
Start I/O No.	ANM0 _H				
Network No.					
Total Stations					
Station No.					
Mode	ANM0 _H				
Refresh Parameters	ANM1 _H				
Network Configuration Setting	ANM2 _H				
Network Operation Setting					
Interrupt Setting	ANM3 _H				
Interlink Transmission Parameters	A082 _H				
Routing Parameters	5003 _H				

(3) MELSECNET/H setting

Network parameters for MELSECNET/H are set.

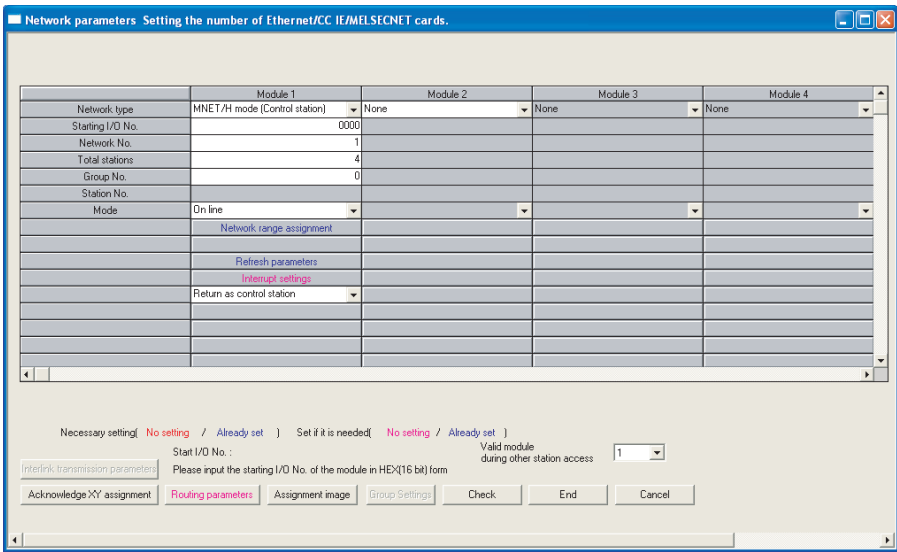


Figure 8.17 Setting the number of Ethernet/CC IE/MELSECNET cards (MELSECNET/H setting)

Table8.19 MELSECNET/H setting list

Item	Parameter No.	Description	Setting range	Default	Reference
Number of modules on MELSECNET/H	5000 _H	Set MELSECNET/H network parameters.	Refer to the manual for the Q series-compatible MELSECNET/H.	-	-
Starting I/O No.	5NM0 _H				
Network No.					
Total stations					
Group No.	05mn _H				
Mode	5NM0 _H				
Refresh parameters	5NM1 _H				
Common parameters	5NM2 _H				
Station inherent parameters	5NM3 _H				
Common parameters 2	5NMA _H				
Station inherent parameters 2	5NMB _H				
Interrupt settings					
Valid module during other station access	5001 _H				
Interlink transmission parameters	5002 _H				
Routing parameters	5003 _H				

(4) Ethernet setting

Network parameters for Ethernet are set.

Network parameters Setting the number of Ethernet/CC IE/MELSECNET cards.

	Module 1	Module 2	Module 3	Module 4
Network type	Ethernet	None	None	None
Starting I/O No.	0020			
Network No.	1			
Total stations				
Group No.	1			
Station No.	1			
Mode	On line			
Operational settings				
Initial settings				
Open settings				
Router relay parameter				
Station No. <-> IP information				
FTP Parameters				
E-mail settings				
Interrupt settings				

Necessary setting: No setting / Already set Set if it is needed: No setting / Already set

Start I/O No.: Please input the starting I/O No. of the module in HEX(16 bit) form

Valid module during other station access: 1

Initial transmission parameters

Acknowledge XY assignment Routing parameters Assignment image Group Settings Check End Cancel

Figure 8.18 Setting the number of Ethernet/CC IE/MELSECNET cards (Ethernet setting)

Table8.20 Ethernet setting list

Item	Parameter No.	Description	Setting range	Default	Reference
Number of modules on Ethernet	9000 _H	Set Ethernet network parameters.	Refer to the manual for the Q series-compatible Ethernet.	-	-
Starting I/ONo.	9N00 _H				
Network No.					
Group No.					
Station No.					
Operational settings					
Initial settings	9N01 _H				
Open settings	9N02 _H				
Router relay parameter	9N03 _H				
Station No.<->IP information	9N05 _H				
FTP Parameters	9N06 _H				
E-mail settings	9N07 _H				
News setting	9N08 _H				
Interrupt settings	9N09 _H				
Valid module during other station access	5001 _H				
Routing parameter	9N04 _H				

(5) CC-Link setting

Parameters for CC-Link are set.

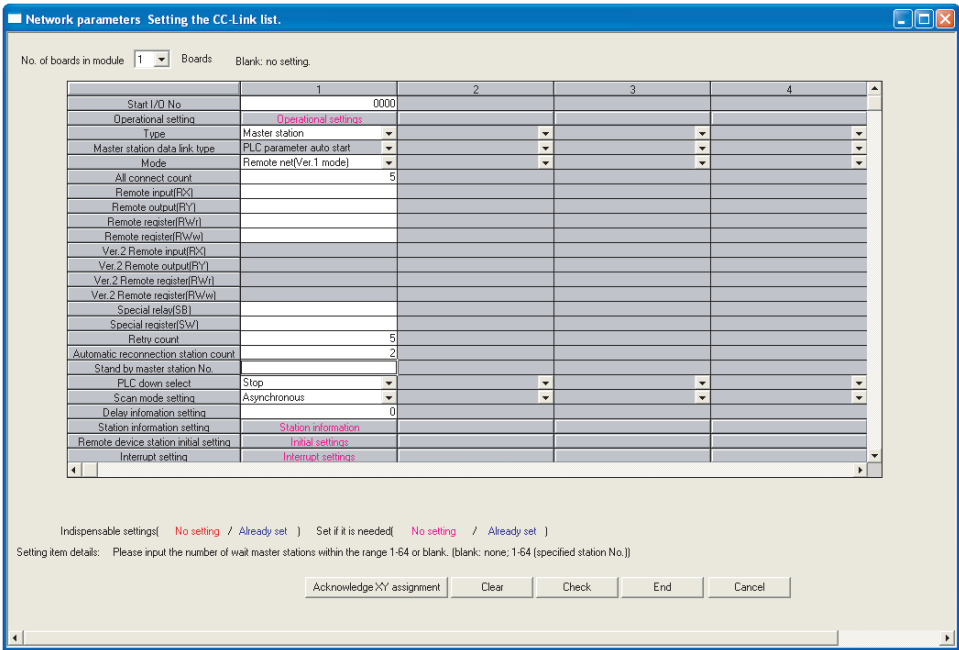


Figure 8.19 Setting the CC-Link list

Table8.21 CC-Link setting list

Item	Parameter No.	Description	Setting range	Default	Reference
Number of modules	C000 _H	Set CC-Link parameters.	Refer to the manual for CC-Link.	-	
Type					
Start I/O No.	CNM2 _H				
Operational setting					
All connect count					
Remote input (RX)	CNM1 _H				
Remote output (RY)					
Remote register (RWr)					
Remote register (RWw)					
Ver.2 Remote input (RX)					
Ver.2 Remote output (RY)					
Ver.2 Remote register (RWr)					
Ver.2 Remote register (RWw)	CNM2 _H				
Special relay (SB)					
Special register (SW)					
Retry count					
Automatic reconnection station count					
Standby master station No.					
PLC down select					
Scan mode setting					
Delay information setting					
Station information setting					
Remote device station initial setting					
Interrupt setting					

8.3 Remote Password

This section provides the list of parameters for use of remote password and describes parameter details.

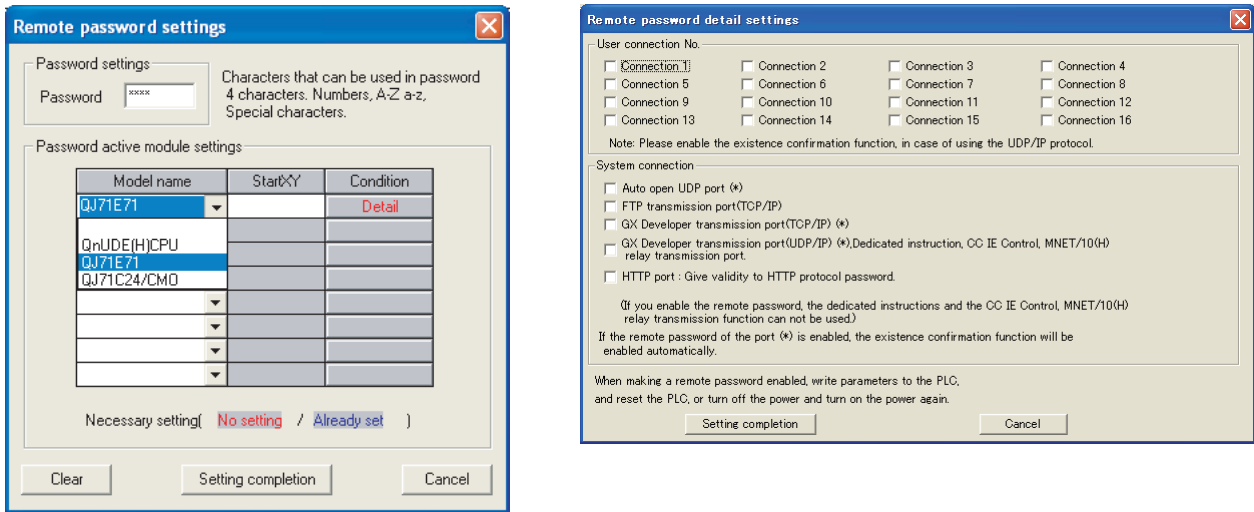


Figure 8.20 Remote password settings dialog box

A remote password is set for an Ethernet module, serial communication module, modem interface module, or Built-in Ethernet port QCPU.

Table8.22 Remote password setting list

Item		Parameter No.	Description	Setting range	Default	Reference
Password settings		-	Enter a remote password.	Four characters or less (alphanumeric characters, special symbols)	-	
Password active module settings	Model name	-	Select a model name of the module for which the remote password set to the CPU module is checked.	- QnUDE(H)CPU - QJ71E71 - QJ71C24/CMO	-	- QnUDE(H)CPU: QnUCPU User's Manual (Communication via Built-in Ethernet Port) - QJ71E71: Ethernet module manual - QJ71C24: Serial communication module manual - QJ71CMO: Modem interface module manual
	Start XY	-	Set the start address of the module for which the remote password is checked.	0000 _H to 0FE0 _H	-	
Detail		-	Set details of the remote password for the QJ71E71.	-	-	
System connection		-	Select a valid port of the remote password for system connection.	Specify a valid port of the remote password. - Auto open UDP port - FTP transmission port (TCP/IP) - GX Developer transmission port (TCP/IP) - GX Developer transmission port (UDP/IP), Dedicated instruction, CC IE Control, MNET/10(H) relay transmission port. - HTTP port	-	

CHAPTER9 DEVICES

This chapter describes the devices that can be used in the CPU module.

9.1 Device List

Table 9.1 lists the names and data ranges of the devices that can be used in the CPU module.

Table9.1 Device list

Classification	Type	Device name	Default			Parameter-set range	Reference	
			Points	Range				
Internal user device	Bit device	Input	8192	X0 to X1FFF	Hexadecimal	Can be changed within 29K words *3	Section 9.2.1	
		Output	8192	Y0 to Y1FFF	Hexadecimal		Section 9.2.2	
		Internal relay	8192	M0 to M8191	Decimal		Section 9.2.3	
		Latch relay	8192	L0 to L8191	Decimal		Section 9.2.4	
		Annunciator	2048	F0 to F2047	Decimal		Section 9.2.5	
		Edge relay	2048	V0 to V2047	Decimal		Section 9.2.6	
		Step relay	8192	S0 to S511/block	Decimal		Section 9.2.9	
		Link relay	8192	B0 to B1FFF	Hexadecimal		Section 9.2.7	
		Link special relay	2048	SB0 to SB7FF	Hexadecimal		Section 9.2.8	
	Word device	Timer*1	2048	T0 to T2047	Decimal		Section 9.2.10	
		Retentive timer*1	0	(ST0 to ST2047)	Decimal			
		Counter*1	1024	C0 to C1023	Decimal			Section 9.2.11
		Data register	12288	D0 to D12287	Decimal			Section 9.2.12
		Link register	8192	W0 to W1FFF	Hexadecimal			Section 9.2.13
		Link special register	2048	SW0 to SW7FF	Hexadecimal			Section 9.2.14
Internal system device	Bit device	Function input	16	FX0 to FXF	Cannot be changed	Section 9.3.1		
		Function output	16	FY0 to FYF		Section 9.3.1		
		Special relay	2048	SM0 to SM2047		Section 9.3.2		
	Word device	Function register	5	FD0 to FD4		Section 9.3.1		
		Special register	2048	SD0 to SD2047		Section 9.3.2		
Link direct device	Bit device	Link input	16384*14	Jn\X0 to Jn\X3FFF*15	Cannot be changed	Section 9.4		
		Link output	16384*14	Jn\Y0 to Jn\Y3FFF*15				
		Link relay	32768	Jn\B0 to Jn\B7FFF				
		Link special relay	512	Jn\SB0 to Jn\SB1FF				
	Word device	Link register	131072	Jn\W0 to Jn\W1FFFF				
		Link special register	512	Jn\SW0 to Jn\SW1FF				
Module access device	Word device	Intelligent function module device	65536	Un\G0 to Un\G65535*2	Cannot be changed	Section 9.5		
		Cyclic transmission area device *4	14336	U3En\G10000 to U3En\G24335	Can be changed			
Index register or standard device register	Word device	Index register or standard device register	20	Z0 to Z19	Decimal	Cannot be changed	Section 9.6	

(To the next page)

Table9.1 Device list (continued)

Classification	Type	Device name	Default			Parameter-set range	Reference
			Points	Range			
File register ^{*7}	Word device	File register	0	-	-	0 to 4086K points ^{*6}	Section 9.7
Extended data register ^{*7}	Word device	Extended data register	0 ^{*16}	-	-		Section 9.8
Extended link register ^{*7}	Word device	Extended link register	0	-	-		Section 9.8
Nesting	-	Nesting	15	N0 to N14	Decimal	Cannot be changed	Section 9.9
Pointer	-	Pointer	4096 ^{*8 *17}	P0 to P4095 ^{*9 *18}	Decimal	Cannot be changed	Section 9.10
		Interrupt pointer	256 ^{*10}	I0 to I255 ^{*11}	Decimal		Section 9.11
Other	Bit device	SFC block device	320 ^{*10}	BL0 to BL319 ^{*12}	Decimal	Cannot be changed	Section 9.12
	-	Network No. specification device	255	J1 to J255	Decimal		Section 9.12.1
		I/O No. specification device	-	U0 to UFF, U3E0 to U3E3E3 ^{*13}	Hexadecimal		Section 9.12.3
		Macro instruction argument device	-	VD0 to VD □	Hexadecimal		Section 9.12.4
Constant	-	Decimal constant	K-2147483648 to K2147483647				Section 9.13.1
		Hexadecimal constant	H0 to HFFFFFFFF				Section 9.13.2
		Real number constant	Single-precision floating-point data: E ± 1.17549435 — 38 to E ± 3.40282347 + 38				Section 9.13.3
			Double-precision floating-point data ^{*5} : E ± 2.2250738585072014 — 308 to E ± 1.7976931348623157 + 308				Section 9.13.3
		Character string constant	Up to 32 characters such as "ABC" or "123", etc.				Section 9.13.4

*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 present 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 setting range can be changed (except input, output, and step relay) in the Device tab of the PLC parameter dialog box.

The step relay can be changed to 0 points for the Universal model QCPU whose serial number (first five digits) is "10042" or later.

(Appendix 2) For the Universal model QCPU whose serial number (first five digits) is "12052" or later, a step relay can be set in increments of 1k point and up to the following points. (Appendix 2)

- Q00UJCPU, Q00UCPU, Q01UCPU, and Q02UCPU: 8192 points
- Universal model QCPU other than the Q00UJCPU, Q00UCPU, Q01UCPU, and Q02UCPU: 16384 points

*4: Available only in multiple CPU systems.

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

*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: "512 points" for the Q00UJCPU, Q00UCPU, and Q01UCPU.

*9: "P0 to P511" for the Q00UJCPU, Q00UCPU, and Q01UCPU.

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

*11: "I0 to I127" for the Q00UJCPU, Q00UCPU, and Q01UCPU.

*12: "BL0 to BL127" for the Q00UJCPU, Q00UCPU, and Q01UCPU.

*13: "U0 to UF" for the Q00UJCPU, "U0 to U3F" or "U3E0 to U3E3E2" for the Q00UCPU and Q01UCPU, and "U0 to U7F" or "U3E0 to U3E3E2" for the Q02UCPU.

*14: "8192 points" for the universal model QCPUs with the first 5 digits of the serial number older than "12011".

*15: "Jn\ X/Y0 to 1FFF" for the universal model QCPUs with the first 5 digits of the serial number older than "12011".

*16: "128k points" for the Q50UDEHCPU and Q100UDEHCPU.

*17: "8192 points" for the Q50UDEHCPU and Q100UDEHCPU.

*18: "P0 to 8191" for the Q50UDEHCPU and Q100UDEHCPU.

9.2 Internal User Device

(1) Definition

Internal user devices can be used for various user applications.

(2) Points for internal user devices

The default values can be changed in the Device tab of the PLC parameter dialog box.

However, the points for the input (X), output (Y), and step relay (S)  Note9.1  Note9.3 cannot be changed.

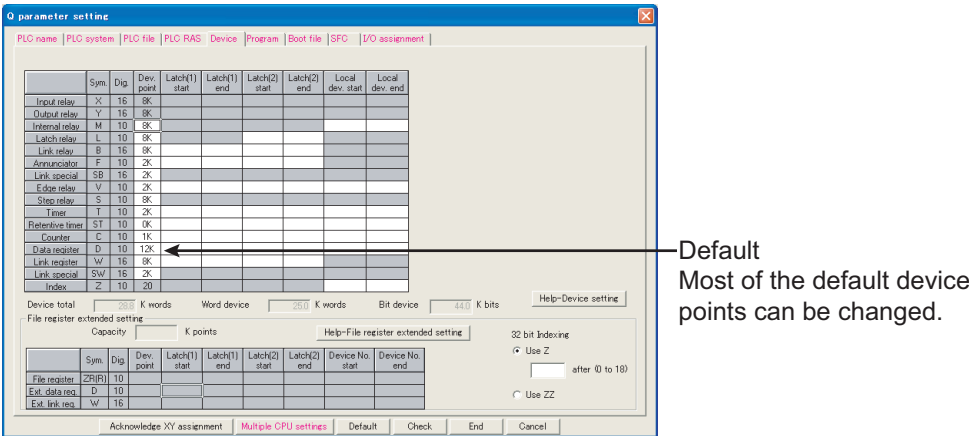


Figure 9.1 Device tab of the PLC parameter dialog box

When changing device points, note the following.

- Set each device in increments of 16 points.
- The maximum number of points for a bit device is 32K.

For an internal relay and link relay, the maximum number of points can be set up to 60K.  Note9.2

- Up to 29K words can be set for total internal user devices.
- One point set for the timer, retentive timer, or counter must be regarded as two points (one for coil and one for contact).



Note9.1

Universal

The points for the step relay (S) can be changed to 0 if the serial number (first five digits) of the Universal model QCPU is "10042" or later.



Note9.2

Universal

This applies to the Universal model QCPU when the first five digits of the serial No. is "10042" or later.



Note9.3

Universal

For the Universal model QCPU whose serial number (first five digits) is "12052" or later, a step relay can be set in increments of 1k point and up to the following points. (Appendix 2)

- Q00UJCPU, Q00UCPU, Q01UCPU, and Q02UCPU: 8192 points
- Universal model QCPU other than the Q00UJCPU, Q00UCPU, Q01UCPU, and Q02UCPU: 16384 points

- When changing device points, the following refresh ranges must not exceed the corresponding device ranges.

- Refresh range of network module
- Auto refresh range of intelligent function module

If device points are set exceeding the corresponding device range, data may be written to any other device or an error may occur.

- The total number of points for the internal relay, latch relay, annunciator, edge relay, link relay, link special relay, step relay, timer, retentive timer, and counter must be set within the following range.

- Serial number (first five digits) of the Universal model QCPU is "10041" or earlier: Up to 64K points
- Serial number (first five digits) of the Universal model QCPU is "10042" or later: Not limited (Up to 29 words for internal user devices)

(3) Memory size

Set the internal user devices so that the following condition is satisfied.

(Bit device size) + (Timer, retentive timer, and counter sizes) + (Word device size) ≤ 29K words

(a) Bit device

For bit devices, 16 points are calculated as one word.

$$(\text{Bit device size}) = \frac{(X+Y+M+L+B+F+SB+V+S)}{16} \text{ Words}$$

(b) Timer (T), retentive timer (ST), and counter (C)

For the timer (T), retentive timer (ST), and counter (C), 16 points are calculated as 18 words.

$$(\text{Timer, retentive timer, or counter size}) = \frac{(T+ST+C)}{16} \times 18 \text{ Words}$$

(c) Word device

For the data register (D), link register (W), and link special register (SW), 16 points are calculated as 16 words.

$$(\text{Word device size}) = \frac{(D+W+SW)}{16} \times 16 \text{ Words}$$

If the points for the internal user devices have been changed, the following files created with the previous parameters can no longer be used.

- Sequence program files
- SFC program files
- ST program files

In such a case, read the above program files out from the CPU module to GX Developer, and then write them back again to the CPU module.

(4) Device point assignment example

Table9.2 shows a device point assignment example.

Table9.2 uses the same format as the device point assignment sheet shown in Appendix 4.

Table9.2 Device point assignment example

Device name	Symbol	Numeric notation	Number of device point ^{*2}		Restriction check					
			Points	Range	Size (words) ^{*3}	Points (bits) ^{*2}				
Input relay ^{*1}	X	Hexadecimal	8K (8192)	X0000 to X1FFF	/16	512	×	1	8192	
Output relay ^{*1}	Y	Hexadecimal	8K (8192)	Y0000 to Y1FFF	/16	512	×	1	8192	
Internal relay	M	Decimal	16K (16384)	M0 to M16383	/16	1024	×	1	16384	
Latch relay	L	Decimal	4K (4096)	L0 to L4095	/16	256	×	1	4096	
Link relay	B	Hexadecimal	4K (4096)	B0000 to B0FFF	/16	256	×	1	4096	
Annunciator	F	Decimal	1K (1024)	F0 to F1023	/16	64	×	1	1024	
Link special relay	SB	Hexadecimal	2K (2048)	SB0000 to SB07FF	/16	128	×	1	2048	
Edge relay	V	Decimal	1K (1024)	V0 to V1023	/16	64	×	1	1024	
Step relay ^{*1}	S	Decimal	8K (8192)	S0 to S8191	/16	512	×	1	8192	
Timer	T	Decimal	2K (2048)	T0 to T2047	×	$\frac{18}{16}$	2304	×	2	4096
Retentive timer	ST	Decimal	2K (2048)	ST0 to ST2047	×	$\frac{18}{16}$	2304	×	2	4096
Counter	C	Decimal	1K (1024)	C0 to C1023	×	$\frac{18}{16}$	1152	×	2	2048
Data register	D	Decimal	14K (14336)	D0 to D14335	×	1	14336			-
Link register	W	Hexadecimal	4K (4096)	W0000 to W4095	×	1	4096			-
Link special register	SW	Hexadecimal	2K (2048)	SW0000 to SW07FF	×	1	2048			-
Total							29568 (29696 or less)			63488 (65536 or less)

*1: The points are fixed for the system. (Cannot be changed)

The points for the step relay can be changed to 0 if the Universal model QCPU whose serial number (first five digits) is "10042" or later. For the Universal model QCPU whose serial number (first five digits) is "12052" or later, a step relay can be set in increments of 1k point and up to the following points. (Appendix 2)

- Q00UJCPU, Q00UCPU, Q01UCPU, and Q02UCPU: 8192 points
- Universal model QCPU other than the Q00UJCPU, Q00UCPU, Q01UCPU, and Q02UCPU: 16384 points

*2: Up to 32K points can be set for each device.

However, up to 60K points can be set for each device of the internal relay and link relay if the Universal model QCPU whose serial number (first five digits) is "10042" or later,

*3: Enter the values multiplied (or divided) by the number shown in the Size (words) column.

9.2.1 Input (X)

(1) Definition

The input (X) is used to send commands or data to the CPU module from external devices such as push-button switches, selector switches, limit switches, and digital switches.

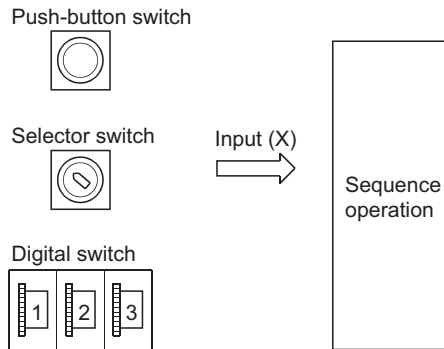


Figure 9.2 Commands from external devices to a CPU module

(2) Concept of input (X)

One input point is assumed to be a virtual relay X_n in the CPU module. Programs use the normally open or closed contact of X_n .

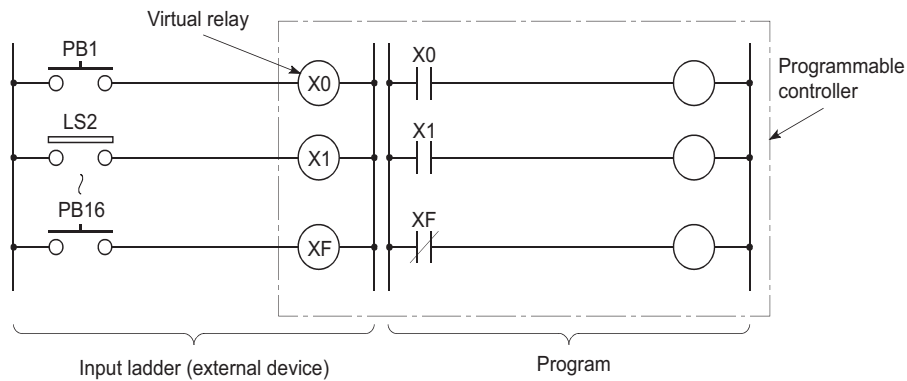


Figure 9.3 Concept of input (X)

(3) Allowable number of normally open or closed contacts

There are no restrictions on the number of normally open or closed contacts of X_n used in a program, as long as the program capacity is not exceeded.

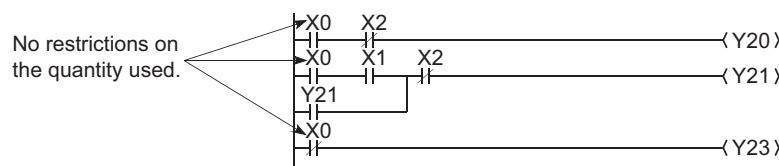


Figure 9.4 Input (X) used in a program

Point

- When debugging a program, the input (X) can be set to on or off by the following:
 - Device test in GX Developer
 - OUT Xn instruction

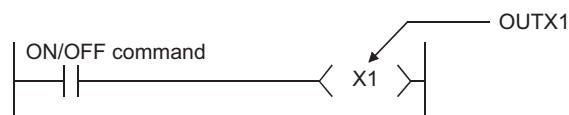


Figure 9.5 Input (X) on/off with the OUT Xn instruction

- The input (X) can also be used for the following.
 - Refresh target device (CPU module side) of RX in CC-Link IE Field network or CC-Link
 - Refresh target device (CPU module side) of CC-Link IE controller network or MELSECNET/H

9.2.2 Output (Y)

(1) Definition

The output (Y) is used to output control results on programs to external devices such as signal lamps, digital displays, electromagnetic switches (contactors), or solenoids.

Data can be output to the outside like using a normally open contact.

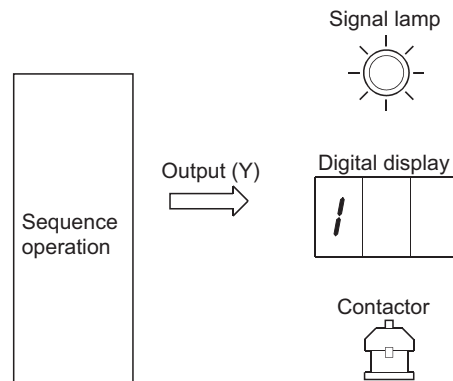


Figure 9.6 Output from a CPU module to external devices

(2) Allowable number of normally open or closed contacts

There are no restrictions on the number of normally open or closed contacts of Y_n used in a program, as long as the program capacity is not exceeded.

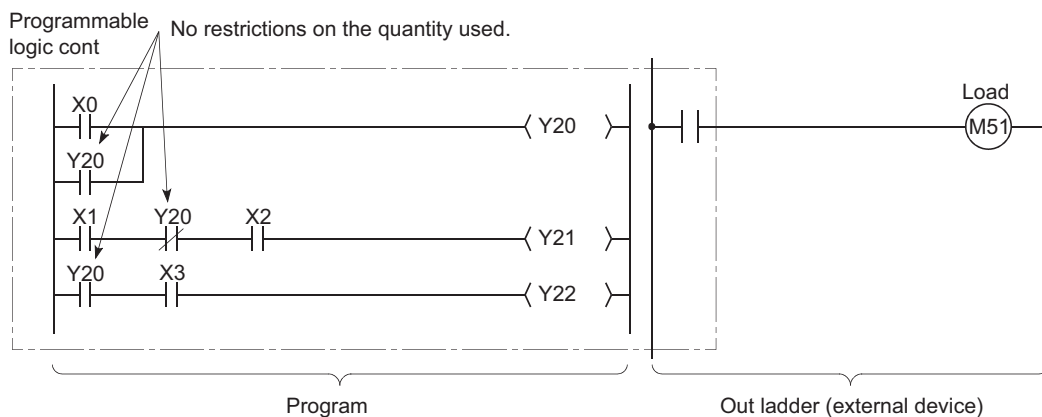


Figure 9.7 Output (Y) used in a program

(3) Using the output (Y) as the internal relay (M)

The output (Y) corresponding to the slots for input modules or empty slots can be utilized as the internal relay (M).

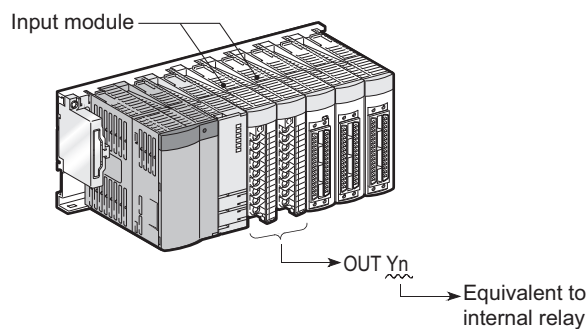


Figure 9.8 Using as the internal relay

9.2.3 Internal relay (M)

(1) Definition

The internal relay (M) is a device for auxiliary relays used in the CPU module.

All of the internal relay are set to off in the following cases:

- When the CPU module is powered off and then on
- When the CPU module is reset
- When latch clear is executed (☞ Section 6.3)

(2) Latch (data retention during power failure)

The internal relay cannot be latched.

(3) Allowable number of normally open or closed contacts

There are no restrictions on the number of normally open or closed contacts used in a program, as long as the program capacity is not exceeded.

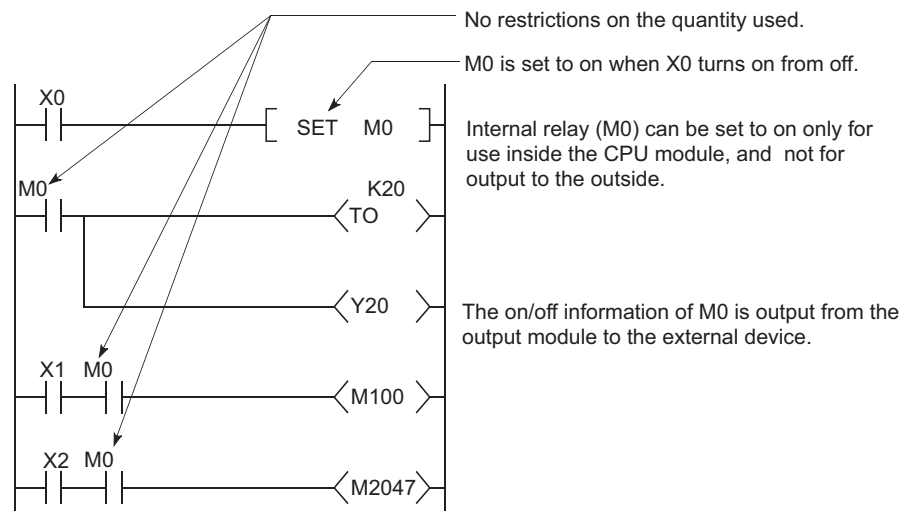


Figure 9.9 Internal relay (M) used in a program

(4) Method for external output

The output (Y) is used to output sequence program operation results to external devices.

Point

Use the latch relay (L) when latch (data retention during power failure) is required.

(☞ Section 9.2.4)

9.2.4 Latch relay (L)

(1) Definition

The latch relay (L) is a device for auxiliary relays that can be latched inside the CPU module.

Latch relay data are retained by batteries in the CPU module during power failure.

Operation results (on/off information) immediately before the following will be also retained.

- Powering off and then on the CPU module
- Resetting the CPU module

(2) Latch relay clear

The latch relay is turned off by the latch clear operation. (☞ Section 3.7)

However, the latch relay set in "Latch (2) start/end" in the Device tab of the PLC parameter dialog box cannot be turned off even by the remote latch clear function.

(3) Allowable number of normally open or closed contacts

There are no restrictions on the number of normally open or closed contacts used in a program, as long as the program capacity is not exceeded.

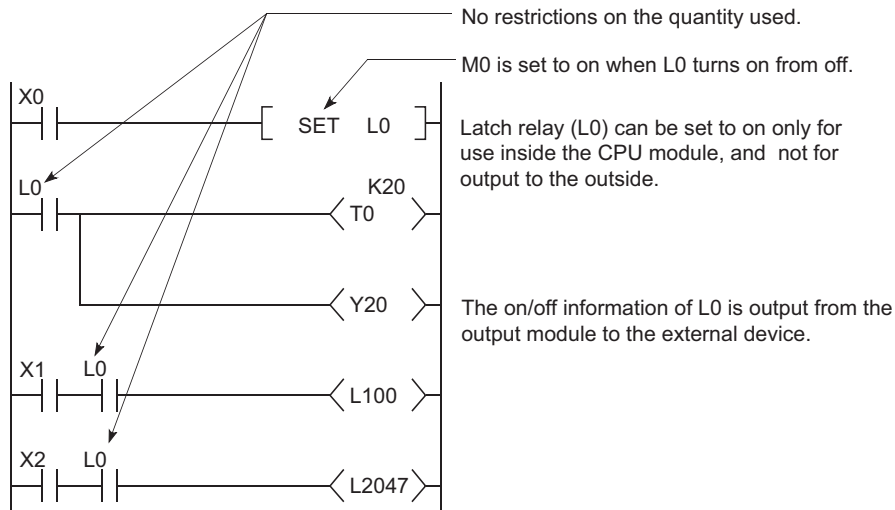


Figure 9.10 Latch relay

Point

Scan time is prolonged when the latch relay (L) is used. Reducing the points of latch relay (L) can reduce the prolonging scan time. (☞ Section 10.1.2)

(4) Method for external output

The output (Y) is used to output sequence program operation results to external devices.

Point

- If latch is not required, use the internal relay (M). (☞ Section 9.2.3)
- The latch clear invalid area is set in the Device setting of PLC parameter. (☞ Section 6.3)

9.2.5 Annunciator (F)

(1) Definition

The annunciator (F) is an internal relay which can be effectively used in fault detection programs for user-created system.

(2) Special relay and special register after annunciator ON

When the annunciator is turned on, the special relay (SM62) is set to on, and the numbers and quantity of the annunciator numbers are stored in the special register (SD62 to SD79).

- Special relay : SM62 ... Turns on even if only one of the annunciator number areas is turned on.
- Special register : SD62 ... Stores the number of the annunciator that was turned on first.
SD63 ... Stores the quantity of the annunciator number areas that are on.
SD64 to SD79 ... Stores annunciator numbers in the order of turning on.
(The same annunciator number is stored in SD62 and SD64.)

The annunciator number stored in SD62 is also registered in the error history area.

(3) Applications of the annunciator

Using the annunciator in a fault detection program allows check for a system fault and identification of the fault (annunciator number) by monitoring the special register (SD62 to SD79) when the special relay (SM62) turns on.

Example In this program, when annunciator (F5) is turned on, the corresponding annunciator number is output to the outside.

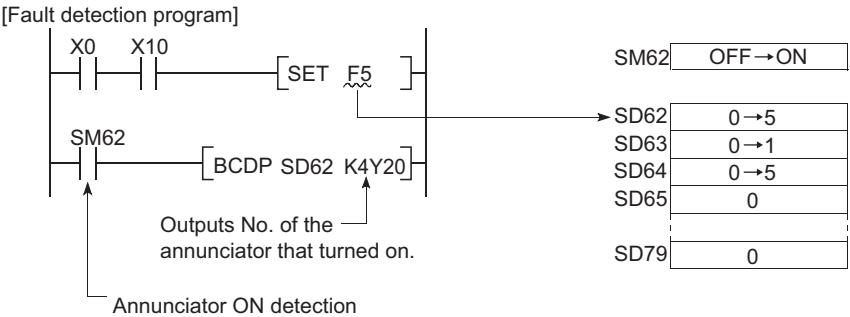


Figure 9.11 Detecting and saving the annunciator ON

(4) Allowable number of normally open or closed contacts

There are no restrictions on the number of normally open or closed contacts used in a program, as long as the program capacity is not exceeded.

(5) Turning on the annunciator and processing

(a) Turning on the annunciator

The following instructions can be used.

1) SET F□ instruction

The SET F□ instruction can be used to turn on the annunciator only on the leading edge (off to on) of an input condition.

Even if the input condition turns off, the annunciator is held on.

Using many annunciator numbers can shorten scan time more than using the OUT F□ instruction.

2) OUT F□ instruction

The OUT F□ instruction can be also used to turn on or off the annunciator. However, since the processing is performed for every scan, the scan time is longer than the case of using the SET F□ instruction.

In addition, execution of the RST F□, LEDR, or BKRST instruction is required after the annunciator is turned off with the OUT F□ instruction.

Therefore, use of the SET F□ instruction is recommended.

Point

If the annunciator is turned on with any instruction other than SET F□ and OUT F□ (for example, the MOV instruction), the same operation as the internal relay (M) is performed.

The ON information is not stored in SM62, and annunciator numbers are not stored in SD62 and SD64 to SD79.

(b) Processing after annunciator on

1) Data stored in the special register (SD62 to SD79)

- Turned-on annunciator numbers are stored in SD64 to SD79 in order.
- The annunciator number in SD64 is stored in SD62.
- SD63 value is incremented by "1".

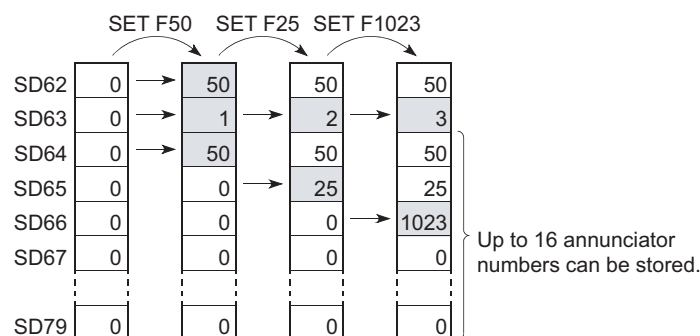


Figure 9.12 Processing after annunciator ON

2) Processing on the CPU

The USER LED on the front side turns on (red).

3) On/off setting for the LED

Whether to turn on the USER.LED or not when the annunciator is turned on can be set in the LED indication setting. (☞ Section 6.21.2).

(6) Turning off the annunciator and processing

(a) Turning off the annunciator

The following instructions can be used.

1) RST F□ instruction

This is used to turn off the annunciator number that was turned on with the SET F□ instruction.

2) LEDR instruction

This is used to turn off the annunciator number stored in SD62 and SD64.

3) BKRST instruction

This is used to turn off all of the annunciator numbers within the specified range.

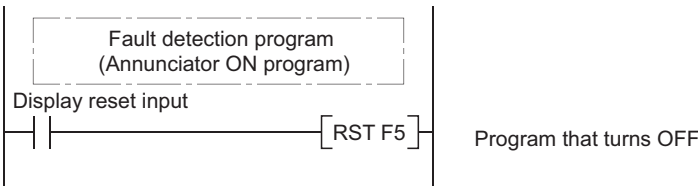
4) OUT F□ instruction

One annunciator number can be turned on or off with the same instruction.

However, even if an annunciator number is turned off with the OUT F□ instruction, the off processing described in (6)(b) in this section is not performed.

If the annunciator is turned off with the OUT F□ instruction, execution of the RST F□, LEDR, or BKRST instruction is required.

- Turning off annunciator 5 (F5)



- Turning off all of the turned-on annunciator numbers

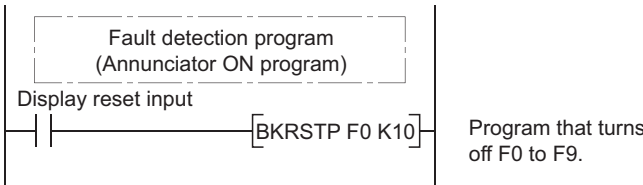


Figure 9.13 Example of turning off the annunciator

Remark

For details of each instruction, refer to the following.

 MELSEC-Q/L Programming Manual (Common Instruction)

.....

(b) Processing after annunciator off

1) Data stored in the special register (SD62 to SD79) after execution of the LEDR instruction

- The annunciator number in SD64 is deleted, and the other annunciator numbers in the register addressed SD65 and after are shifted accordingly.
- The annunciator number in SD64 is stored into SD62.
- SD63 value is decremented by "1".
- If the SD63 value is changed to "0", SM62 is turned off.

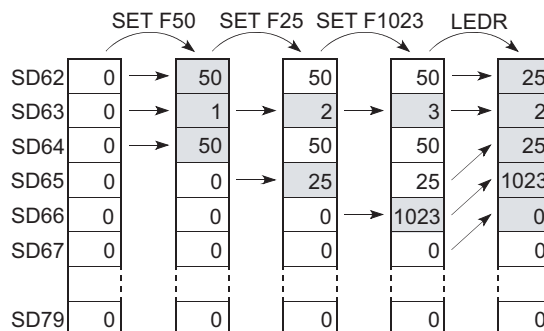


Figure 9.14 Processing after annunciator OFF (when the LEDR instruction executed)

2) Data stored in the special register (SD62 to SD79) when the annunciator is turned off with the RST F□ or BKRST instruction

- The annunciator number specified with the RST or BKRST instruction is deleted, and the other annunciator numbers in the register addressed SD65 and after are shifted accordingly.
- If the existing annunciator number in SD64 is turned off, a new annunciator number stored in SD64 will be stored in SD62.
- SD63 value is decremented by "1".
- If the SD63 value is changed to "0", SM62 is turned off.

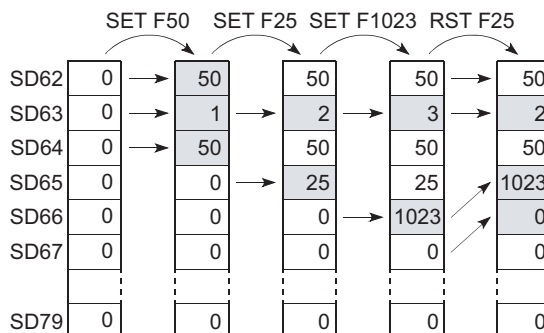


Figure 9.15 Processing after annunciator OFF (when the RST F□ instruction is executed)

3) LED indication

When all of the annunciator numbers in SD64 to SD79 turn off, the LED that was turned on by turn-on of the annunciator will turn off. (☞ (5)(b) in this section)

Point

If the LEDR instruction is executed while the annunciator is on and at the same time the operation continuation error that has higher priority (☞ Section 6.21.1) than the annunciator has occurred, the LEDR instruction clears the higher priority error. Because of this, the annunciator is not turned off by execution of the LEDR instruction.

To turn off the annunciator with the LEDR instruction, remove the error whose priority is higher than that of the annunciator.

9.2.6 Edge relay (V)

(1) Definition

The edge relay (V) is a device in which the on/off information from the beginning of the ladder block. Contacts only can be used. (Coils cannot be used).

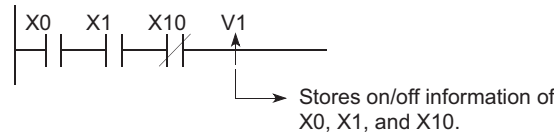
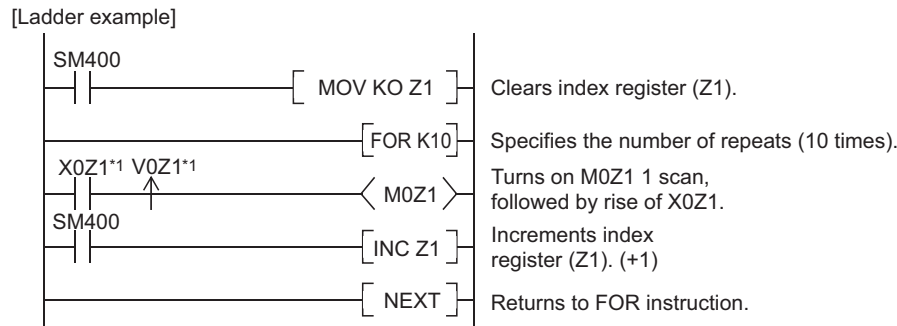


Figure 9.16 Edge relay

(2) Applications of the edge relay

The edge relay can be utilized to detect the leading edge (off to on) in programs configured using index modification.



*1: The on/off information for X0Z1 is stored in the V0Z1 edge relay.
For example, the on/off information of X0 is stored in V0, and that of X1 is stored in V1.

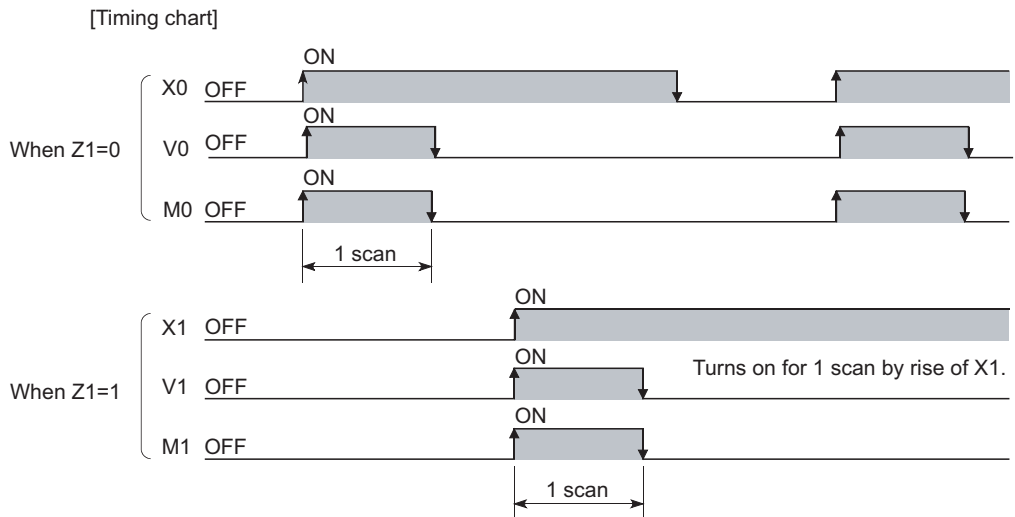


Figure 9.17 Application example and timing chart of the edge relay

(3) Precautions

The edge relay of the same number cannot be set only once in a program.

9.2.7 Link relay (B)

(1) Definition

The link relay (B) is a relay on the CPU module side, and it is used for refreshing the link relay (LB) data of another module such as a MELSECNET/H network module to the CPU module or refreshing the CPU module data to the link relay (LB) of the MELSECNET/H network module.

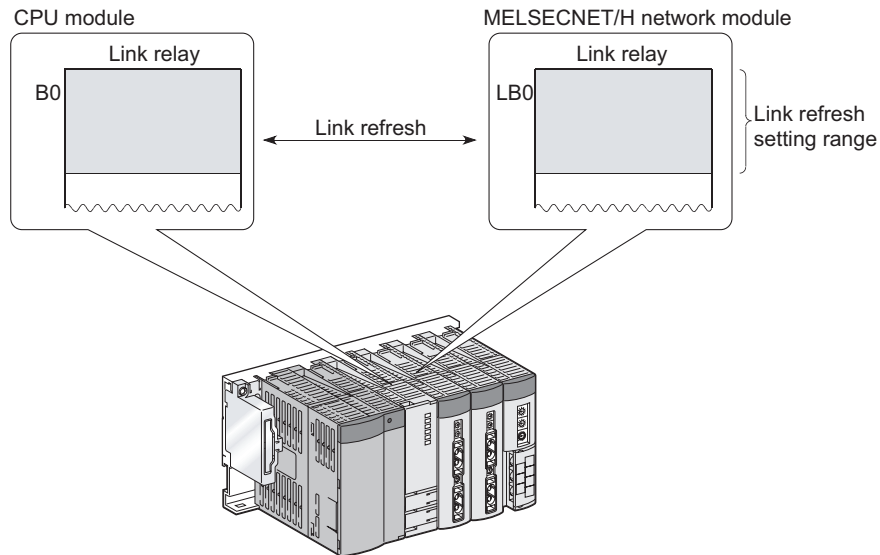


Figure 9.18 Link refresh

(2) Allowable number of normally open or closed contacts

There are no restrictions on the number of normally open or closed contacts used in a program, as long as the program capacity is not exceeded.

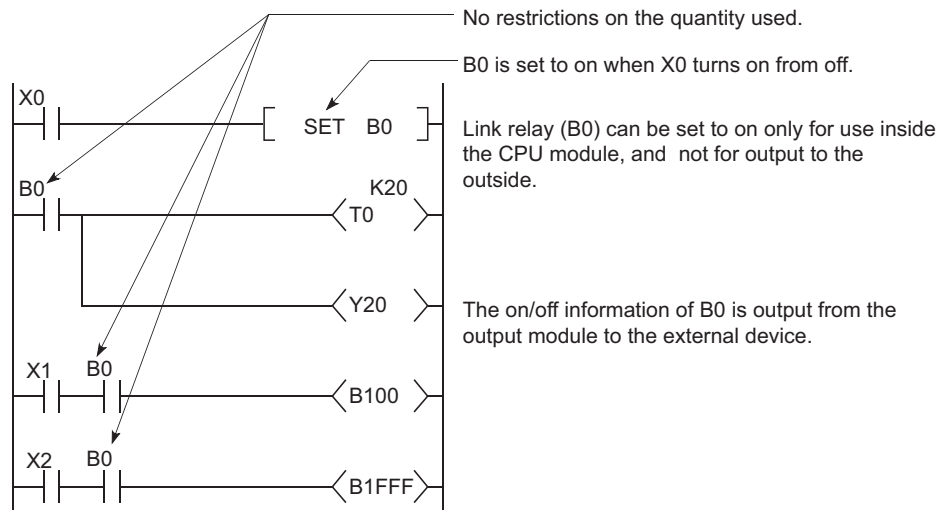


Figure 9.19 Link relay

(3) Using the link relay in the network system

- Network parameters must be set.
- The link relay range areas that is not set by network parameters (not used for a network system such as a MELSECNET/H network) can be used as the internal relay or latch relay.
- Link relay range where no latch is performed.....Internal relay
 - Link relay range where latch is performed.....Latch relay

Point  To use the link device in the network module exceeding link relay points of the CPU module (default: 8192 points), change the link relay points in the Device tab of the PLC Parameter dialog box.

Remark
For network parameters, refer to the following.
 Reference manual for each network module
.....

9.2.8 Link special relay (SB)

(1) Definition

The Link special relay (SB) is a relay that indicates various communication status and detected errors of intelligent function modules such as CC-Link IE module or MELSECNET/H module.

Each of this device area is turned on or off according to a factor occurred during data link.

The communication status and errors on the network can be confirmed by monitoring the link special relay (SB).

(2) Number of link special relay points

The points for the link special relay in the CPU module is 2048. (SB0 to SB7FF).

However, the points can be changed in the Device tab of the PLC parameter dialog box. (☞ Section 9.1)

To an intelligent function module that has a link special relay, such as a CC-Link IE module or MELSECNET/H module, 512 points are assigned.

Assigning the link special relay as shown in Figure 9.20 allows refresh of the CC-Link link special relay (SB) to the link special relay (SB) of the CPU module.

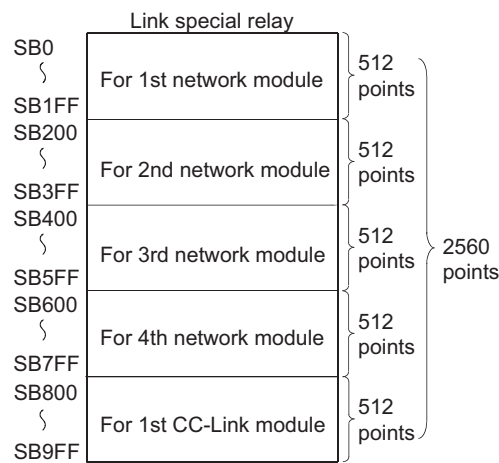


Figure 9.20 Points of the link special relay

Remark
For details of the link special relay, refer to the manual for an intelligent function module that has the link special relay.
.....

9.2.9 Step relay (S)

This device is provided for SFC programs.

Point

Because the step relay is a device exclusively used for SFC programs, it cannot be used as an internal relay in the sequence program.
If used, an SFC error will occur, and the system may go down.

Remark

For use of the step relay, refer to the following.
 MELSEC-Q/L/QnA Programming Manual (SFC)

9.2.10 Timer (T)

(1) Definition

Time counting starts when a coil is turned on, and it times out and the contact turns on when the current value reaches the set value.

The timer is of an incremental type.

(2) Timer types

Timers are mainly classified into the following two types.

- 1) Timer of which value is set to 0 when the coil is turned off.
- 2) Retentive timer that holds the current value even if the coil is turned off.

Also, low-speed and high-speed timers are included in timer 1), and low-speed and high-speed retentive timers are included in timer 2).

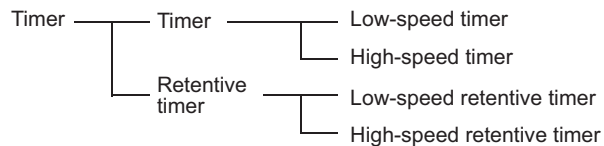


Figure 9.21 Timer types

(3) Specification of the timer

The same device is used for the low- and high-speed timers, and the type is determined according to the instruction used.

Example For the OUT T0 instruction, the low-speed timer is specified, and for the OUTH T0 instruction, the high-speed timer is specified.

The same device is used for the low- and high-speed retentive timers, and the type is determined according to the instruction used.

Example For the OUT ST0 instruction, the low-speed retentive timer is specified, and for the OUTH ST0 instruction, the high-speed retentive timer is specified.

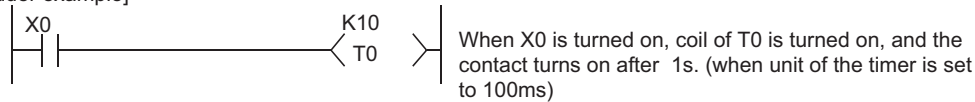
(4) Low-speed timer

(a) Definition

This type of timer measures time in increments of 1 to 1000ms.

The timer starts time measurement when its coil is turned on, and when it times out, the contact is turned on. If the timer's coil is turned off, the current value is changed to "0" and the contact is turned off.

[Ladder example]



[Timing chart]

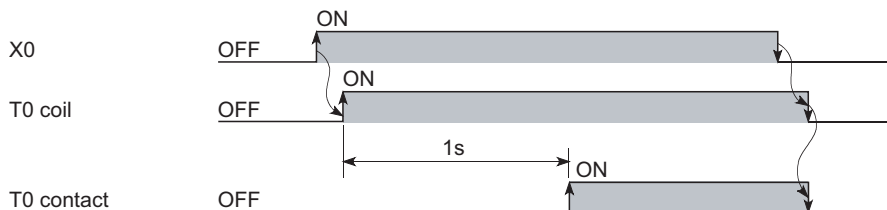


Figure 9.22 Ladder example and timing chart of a low-speed timer

(b) time increment setting

The time increment is set in the PLC system tab of the PLC parameter dialog box.

The default is 100ms, and it can be changed in increments of 1ms.

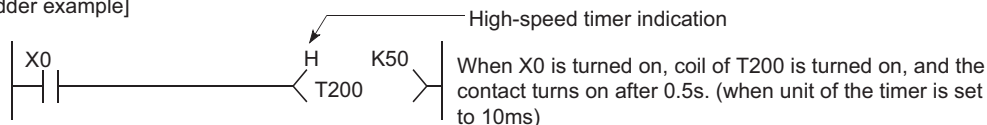
(5) High-speed timer

(a) Definition

This type of timer measures time in increments of 0.01 to 100ms.

The timer starts time measurement when its coil is turned on, and when it times out, the contact is turned on. If the timer's coil is turned off, the current value is changed to "0" and the contact is turned off.

[Ladder example]



[Timing chart]

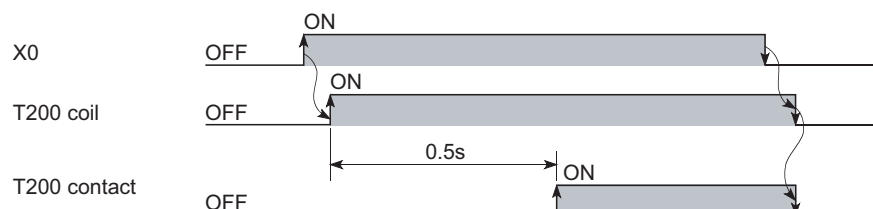


Figure 9.23 Ladder example and timing chart of a high-speed timer

(b) Time increment setting

The time increment is set in the PLC system tab of the PLC parameter dialog box.

The default is 10.0ms, and it can be changed in increments of 0.01ms.

(7) Timer processing and accuracy

(a) Processing

When the OUT T□ or OUT ST□ instruction is executed, the on/off switching of the timer coil, current value update, and on/off switching of the contact are performed.
In the END processing, the current timer value is not updated and the contact is not turned on/off.

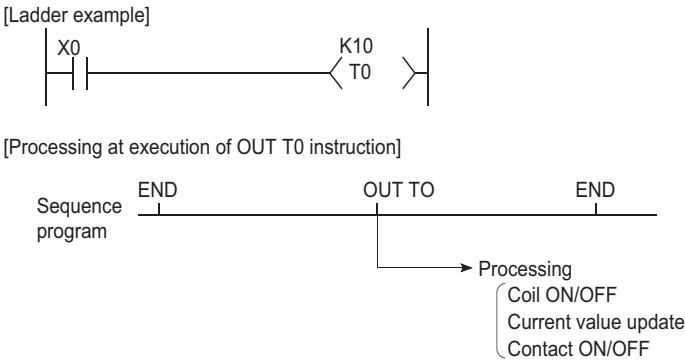


Figure 9.25 Processing at execution of the OUT T0 instruction

(b) Accuracy

The value obtained by the END instruction is added to the current value when the OUT T□ or OUT ST□ instruction is executed.

The current value is not updated while the timer coil is off even if the OUT T□ or OUT ST□ instruction is executed.

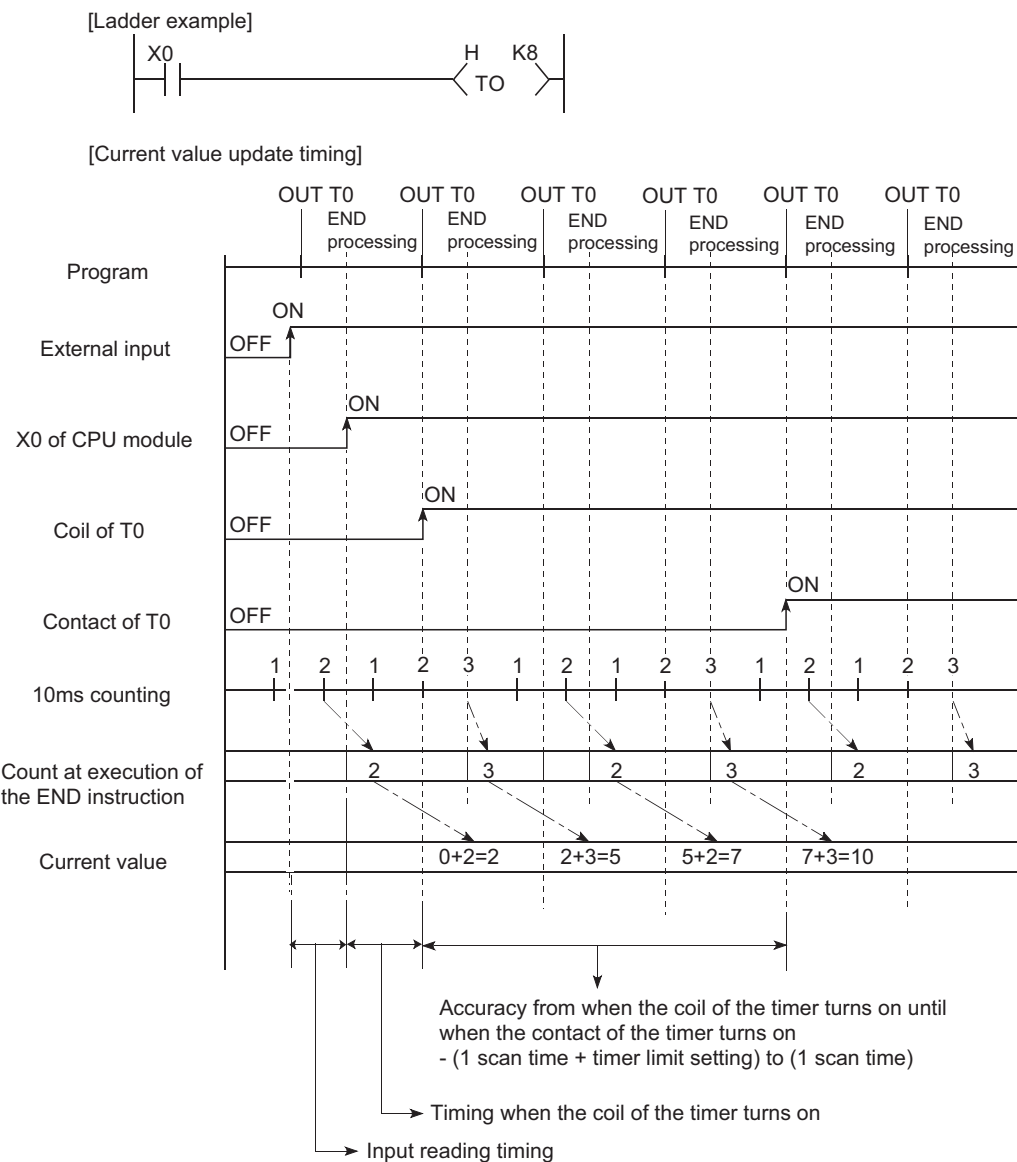


Figure 9.26 Timer accuracy (in the case of 10ms)

Accuracy of the timer response that is from reading input (X) to output the data is up to "2-scan time + timer limit setting".

(8) Precautions for using timers

(a) Use of the same timer

Do not use the OUT T□ instruction that describes the same timer more than once within one scan.

If this occurs, the current timer value will be updated by each OUT T□ instruction execution, resulting in incorrect time measurement.

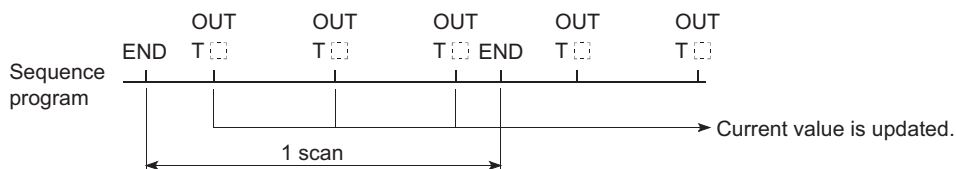


Figure 9.27 When the same timer is used

(b) When the timer is not executed in every scan

While a coil of a timer (for example, T1) is on, do not make the OUT T1 instruction jumped to any other part with another instruction such as CJ.

If jump of the OUT T□ instruction has occurred, the current timer value is not updated.

Also, if a timer exists in a subroutine program, execute a subroutine call including the OUT T1 instruction once in each scan while the coil of the timer (for example, T1) is on.

Failure to do so will not update the current timer value.

(c) Programs that cannot use timers

Timers cannot be used in interrupt programs and fixed scan execution programs.

(d) When the set value is 0

The contact turns on when the OUT T□ instruction is executed.

(e) When the set value is 1

When the timer limit setting value is greater than the scan time value, the count value of the timer becomes 1 at execution of the END instruction in a scan where the count value reaches to the timer limit setting value.

When the coil of the timer turns on in the scan next to the one where the count value becomes 1, the current timer value becomes 1. In this case, the timer times up in the step and the contact turns on.

If the contact of the timer (setting value = 1) turns on in a short time, change the timer with small timer limit setting value and set a greater value to the contact.

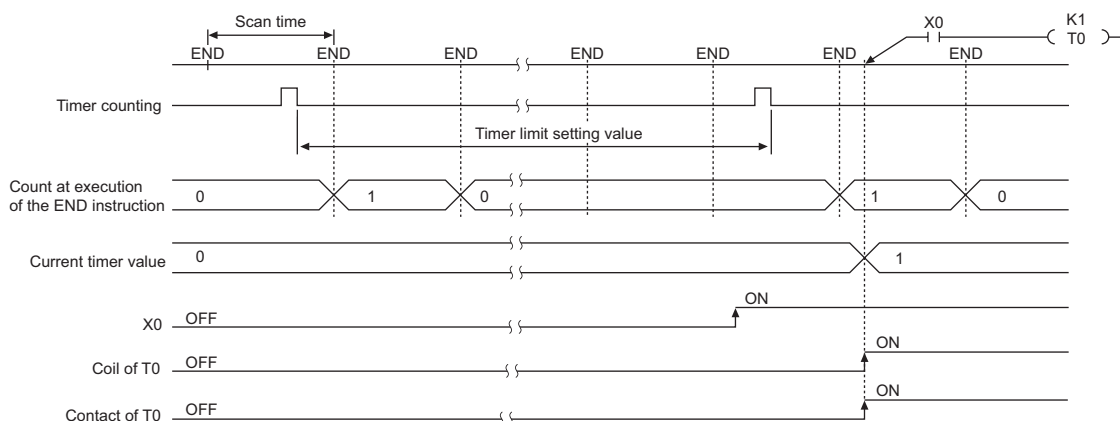


Figure 9.28 Operation when the set value is 1

(f) When the set value is changed after time-out

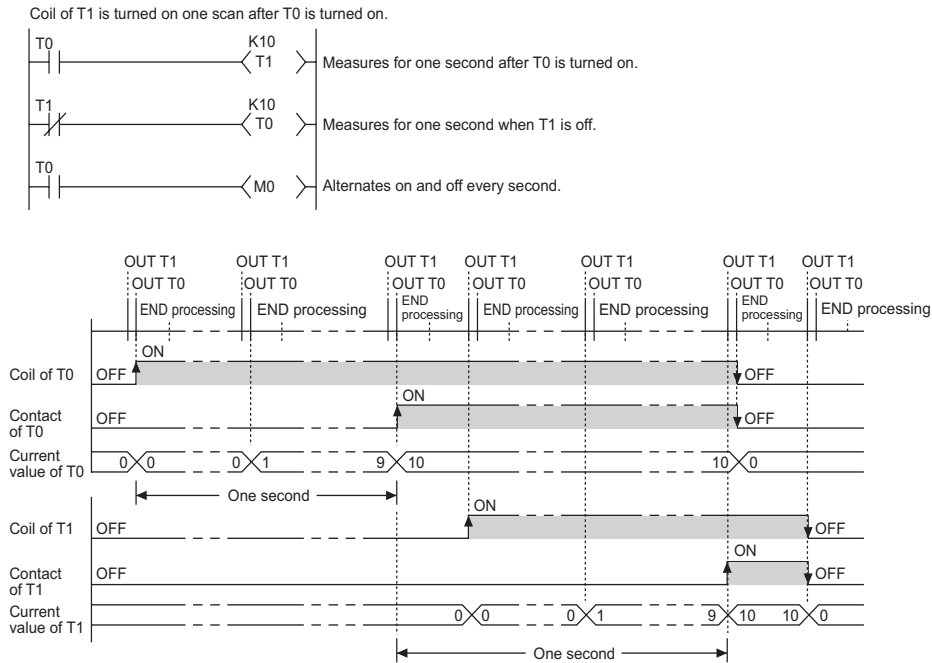
Even if the set value is changed to a larger value after time-out of the timer, the timer remains timed-out and does not start the operation.

(g) When using multiple timers

When using multiple timers to update the respective current values at execution of each OUT T□ instruction, pay attention to the ladder sequence.

For example, to create an on/off ladder using two timers, refer to examples shown in Figure 9.29.

[Correct ladder example]



[Incorrect ladder example]

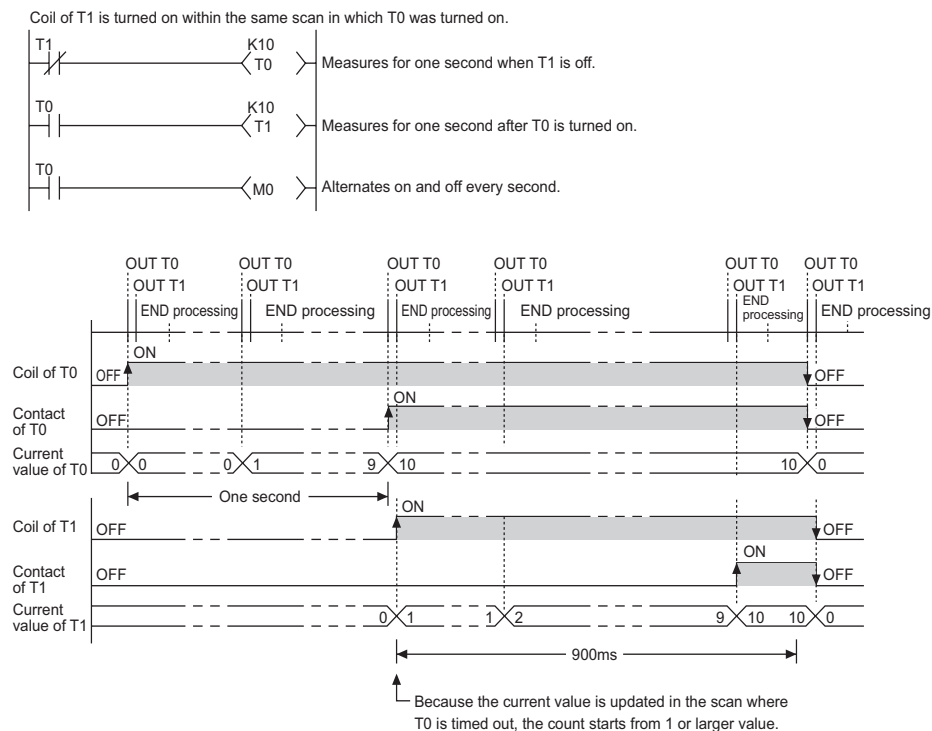


Figure 9.29 On/off ladder using two timers

9.2.11 Counter (C)

(1) Definition

The counter (C) is a device that counts the number of rises for input conditions in sequence programs. When the count value matches the set value, the counting stops and its contact is turned on. The counter is of an incremental type.

(2) Counter type

The following counter is available.

- Counter that counts the number of rises for input conditions in sequence programs

(3) Counting

(a) When the OUT C□ instruction is executed

The coil of the counter is turned on/off, the current value is updated (the count value + 1), and the contact is turned on.

In the END processing, the current counter value is not updated and the contact is not turned on.

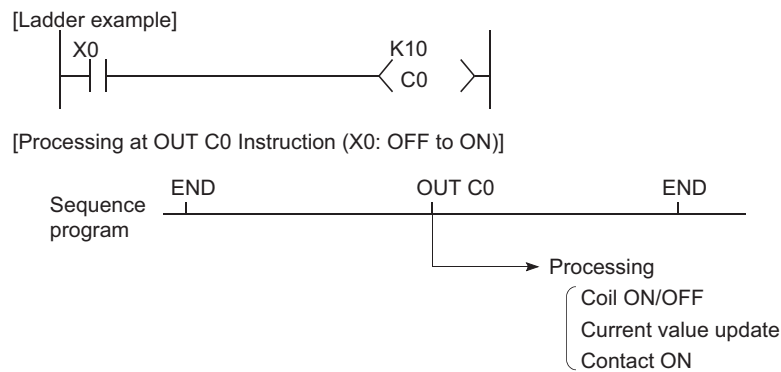


Figure 9.30 Execution and processing of the OUT C□ instruction

(b) Current value update (count value + 1)

The current value is updated (count value + 1) at the leading edge (OFF → ON) of the OUT C□ instruction.

The current value is not updated while the coil is off, or when it remains on or turns off from on by the OUT C□ instruction.

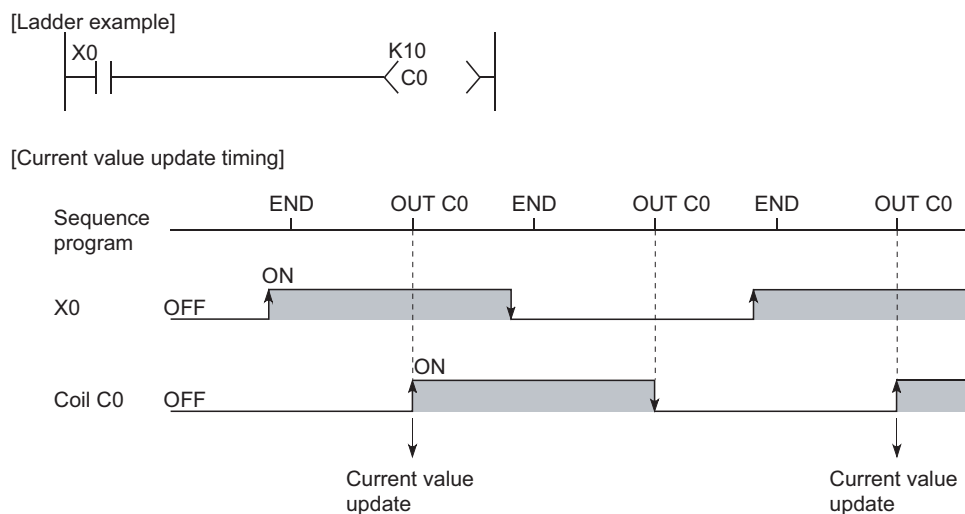


Figure 9.31 Current value update timing

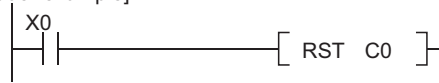
(c) Resetting the counter

The current counter value is not cleared even if the OUT C□ instruction is turned off.

To clear the current value and to turn off the contact of the counter, use the RST C□ instruction.

At the time of execution of the RST C□ instruction, the counter value is cleared, and the contact is also turned off.

[Ladder example]



[Counter reset timing]

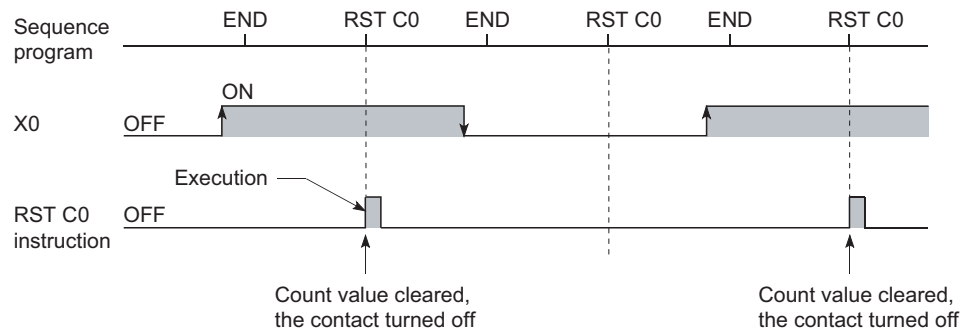


Figure 9.32 Resetting the counter

1) Precautions for resetting the counter

Execution of the RST C $\square$ instruction also turns off the coil of C $\square$.

If the execution condition for the OUT C $\square$ instruction is still ON after execution of the RST C $\square$ instruction, turn on the coil of C $\square$ at execution of the OUT C $\square$ instruction and update the current value (count value + 1).

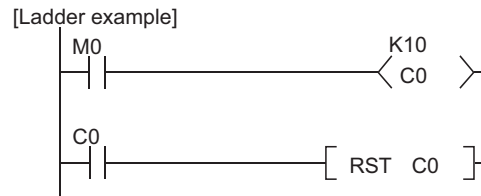


Figure 9.33 Counter resetting ladder example

In the above ladder example, when M0 turns on from off, the coil of C0 turns on, updating the current value.

When C0 reaches the preset value finally, the contact of C0 turns on, and execution of the RST C0 instruction clears the current value of C0. At this time, the coil of C0 also turns off.

If M0 is still on in the next scan, the current value is updated since the coil of C0 turns on from off at execution of the OUT C0 instruction. (The current value is changed to 1.)

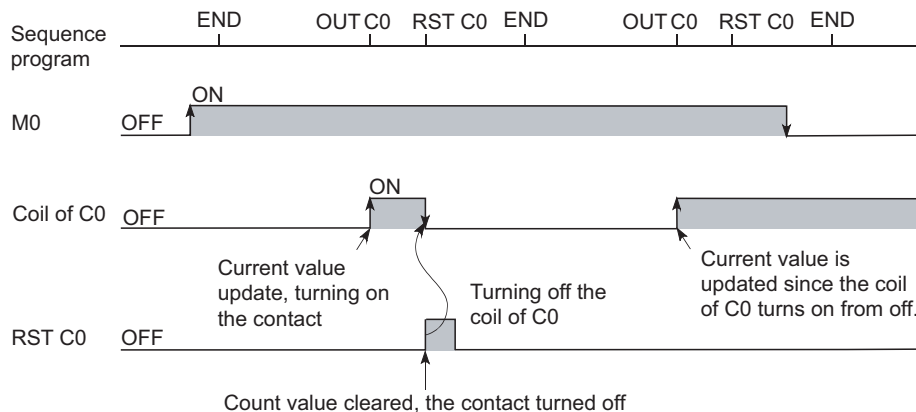


Figure 9.34 Current value update timing

To prevent the above, it is recommended to add a normally closed contact of the OUT C0 instruction execution to the condition for the RST C0 instruction execution so that the coil of C0 does not turn off while the execution condition (M0) of the OUT C0 instruction is on.

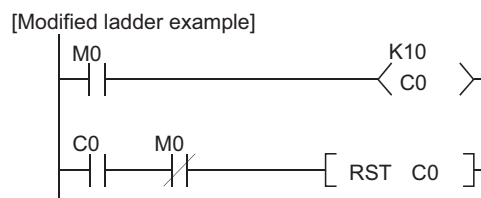


Figure 9.35 Counter resetting ladder example (recommended)

(d) Maximum counting speed

The counter can count only when the on/off time of the input condition is longer than the execution interval of the corresponding OUT C□ instruction.

The maximum counting speed is calculated by the following expression:

$$\text{Maximum counting speed (Cmax)} = \frac{n}{100} \times \frac{1}{T} [\text{times/s}]$$

n: Duty (%) *1
T: Execution interval of the OUT C□ instruction (sec)

*1: Duty (n) is the ON-OFF time ratio of count input signal, and is expressed as a percentage value.

$$\text{When } T1 \geq T2, n = \frac{T2}{T1+T2} \times 100\%$$

$$\text{When } T1 < T2, n = \frac{T1}{T1+T2} \times 100\%$$



Figure 9.36 Duty ratio

Point

The maximum counting speed can be increased by placing multiple counters within one scan.

At this time, use the direct access input (DX□) (see Section 3.8.2) for the counter input signal.

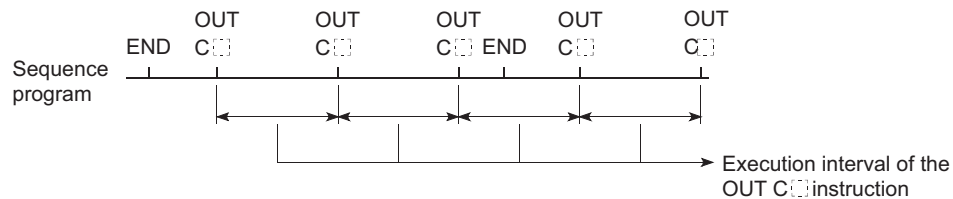


Figure 9.37 Ladder example for increasing the maximum counting speed

(4) Precautions

(a) When counting processing is suspended

If an interrupt occurs during execution of the processing shown below, counting is suspended until the execution of each processing is completed.

- Each instruction on the sequence program
- Interrupt program
- Fixed scan execution type program

Upon completion of the processing, the counting restarts.

However, if the same interrupt occurs again during each processing, these interrupts are counted as once.

9.2.12 Data register (D)

(1) Definition

The data register (D) is a memory in which numeric data (-32768 to 32767, or 0000H to FFFFH) can be stored.

(2) Bit structure of the data register

(a) Bit structure and read/write unit

One point of the data register consists of 16 bits, and data can be read or written in units of 16 bits.

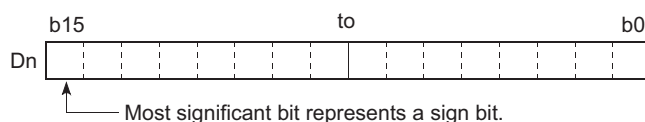


Figure 9.38 Bit structure of the data register

Point

Data register data are handled as signed data.

In the case of the hexadecimal notation, 0000H to FFFFH can be stored. However, because the most significant bit represents a sign bit, decimal values that can be specified are -32768 to 32767.

(b) When using a 32-bit instruction for the data register

For a 32-bit instruction, two consecutive points of the data register (D_n and D_{n+1}) are the target of the processing.

The lower 16 bits correspond to the data register number (D_n) specified in the sequence program, and the higher 16 bits correspond to the specified data register number + 1.

Example When D12 is specified in the DMOV instruction, D12 represents the lower 16 bits and D13 represents the higher 16 bits.

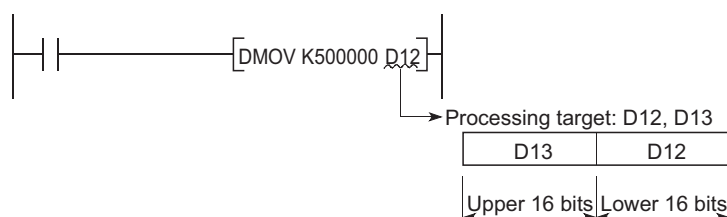


Figure 9.39 Data transfer with a 32-bit instruction and storage location

Data of -2147483648 to 2147483647 or 00000000H to FFFFFFFFH can be stored in a two-point area of the data register. (The most significant bit in a 32-bit structure is a sign bit.)

(3) Retention of stored data

The data stored in the data register are held until other different data are stored.

Note that the stored data are initialized when the CPU module is powered off or reset.

9.2.13 Link register (W)

(1) Definition

The link register (W) is a memory in the CPU module, which is refreshed with link register (LW) data of an intelligent function module such as a MELSECNET/H network module.

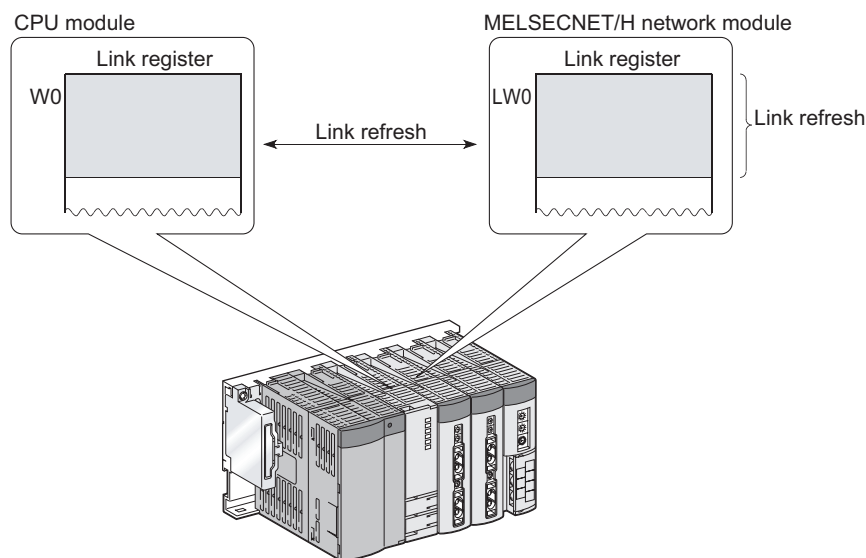


Figure 9.40 Link refresh

In the link register, numeric data (-32768 to 32767, or 0000H to FFFFH) are stored.

(2) Bit structure of the link register

(a) Bit structure and read/write unit

One point of the link register consists of 16 bits, and data can be read or written in units of 16 bits.

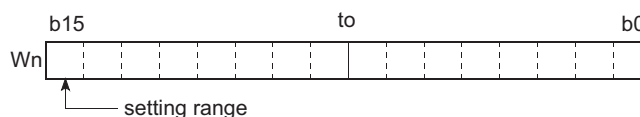


Figure 9.41 Bit structure of the link register

Point

- Link register data are handled as signed data.
In the case of the hexadecimal notation, 0000H to FFFFH can be stored. However, because the most significant bit represents a sign bit, decimal values that can be specified are -32768 to 32767.
- The area range in the link register, which is not used for a MELSECNET/H network module, can be used as a data register.

(b) When using a 32-bit instruction for the link register

For a 32-bit instruction, two consecutive points of the data register (Wn and Wn+1) are the target of the processing.
The lower 16 bits correspond to the link register number (Wn) specified in the sequence program, and the higher 16 bits correspond to the specified link register number + 1.

Example When W12 is specified in the DMOV instruction, W12 represents the lower 16 bits and D13 represents the higher 16 bits.

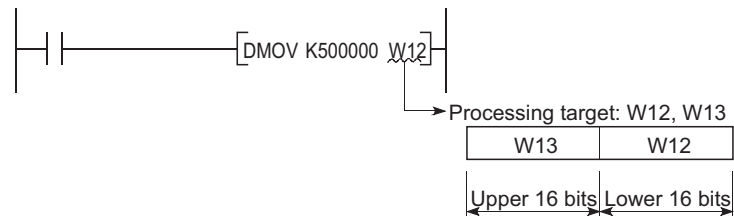


Figure 9.42 Data transfer with a 32-bit instruction and storage location

Data of -2147483648 to 2147483647 or 00000000H to FFFFFFFFH can be stored in a two-point area of the link register. (The most significant bit in a 32-bit structure is a sign bit.)

(3) Retention of stored data

The data stored in the link register are held until other different data are stored.
Note that the stored data are initialized when the CPU module is powered off or reset.

Point

To use the link device in the network module exceeding link register points of the CPU module (default: 8192 points), change the link register points in the Device tab of the PLC Parameter dialog box.

(4) Using the link register in a network system

Network parameters must be set.
The area range that is not set by network parameters can be used as a data register.

Remark

For the network parameters, refer to the following.
Reference manual for each network module

9.2.14 Link special register (SW)

(1) Definition

The link special register (SW) is used to store communication status data and error data of intelligent function modules, such as CC-Link IE module and MELSECNET/H module.
Because the data link information is stored as numeric data, error locations and causes can be checked by monitoring the link special register.

(2) Number of link special register points

The points for the link special register in the CPU module are 2048 (SW0 to SW7FF).
However, the number of points can be changed in the Device tab of the PLC parameter dialog box.
(☞ Section 9.1)

The points for an intelligent function module such as a CC-Link IE module or a MELSECNET/H module are 512.
By assigning it as shown in Figure 9.43, data of the link special register (SW) in a CC-Link module can be also refreshed to the link special register (SW) in the CPU module.

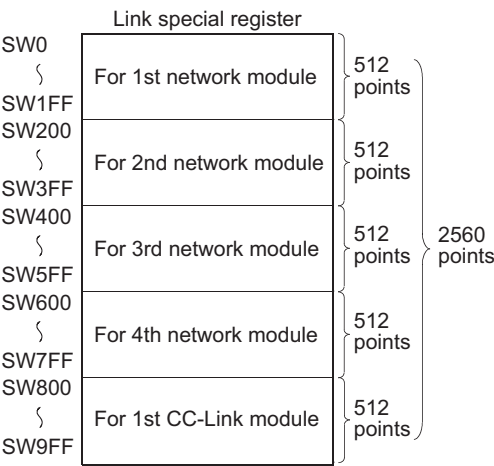


Figure 9.43 Points of the link special register

Remark
For details of the link special register, refer to the manual for each intelligent function module that has the link special register.
.....

9.3 Internal System Devices

Internal system devices are provided for system operations.

The allocations and sizes of internal system devices are fixed, and cannot be changed by the user.

9.3.1 Function devices (FX, FY, FD)

(1) Definition

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.

Example When FX0, FY1, and FD2 are used in a subroutine program, and if X0, M0, and D0 are specified with a subroutine program call instruction, on/off data of X0 and FY1 are passed to FX0 and M0 respectively, and D0 data are passed to FD2.

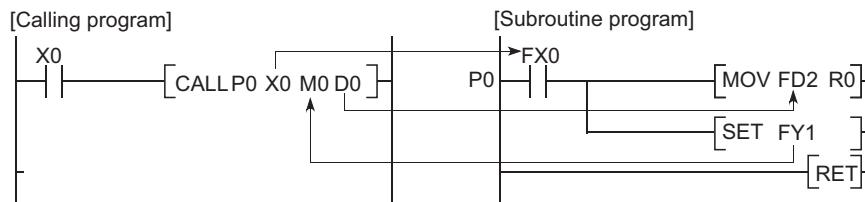


Figure 9.44 Application example of function devices

(2) Applications of function devices

Because a device in each calling program can be determined by using a function device for subroutine programs, the same subroutine program can be used without considering other calling programs.

(3) Types of function devices

The following three types of function devices are available.

- Function input (FX)
- Function output (FY)
- Function register (FD)

(a) Function input (FX)

- The function input is used to pass on/off data to a subroutine program.
- Bit data specified by a subroutine call instruction with argument passing are fetched into a subroutine program and they are used for operations.
- All bit devices for the CPU module can be used.

(b) Function output (FY)

- The function output is used for passing an operation result (on/off data) in a subroutine program to a calling program.
- An operation result is stored in the device specified in the subroutine program with argument passing.
- All bit devices except for input devices of the CPU module (X and DX) can be used.

(c) Function register (FD)

- The function register is used for data writing or reading between a subroutine program and a calling program.
- The CPU module auto-detects the input or output conditions of the function register.
Source data are input data of the subroutine program.
Destination data are output data from the subroutine program.
- The function register of one point can occupy up to four words.
Note that, however, the number of words used differs depending on the instruction in the subroutine program.

1) A one-word instruction uses one word only.

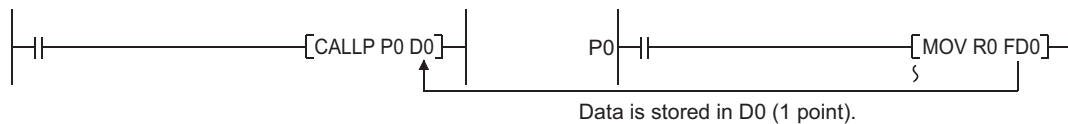


Figure 9.45 When the function register of one point occupies one word

2) two-word instruction uses two words.

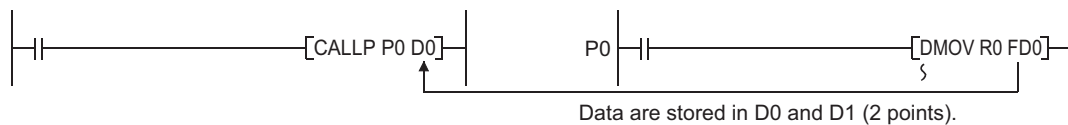


Figure 9.46 When the function register of one point occupies two words

3) At a destination using 32-bit multiplication or division, four words are used.



Figure 9.47 When the function register of one point occupies four words

- Word devices of the CPU module can be used.

Point

In subroutine programs with argument passing, do not use any devices that are used by the function register. If this occurs, function register values will not be normally passed to the calling program.

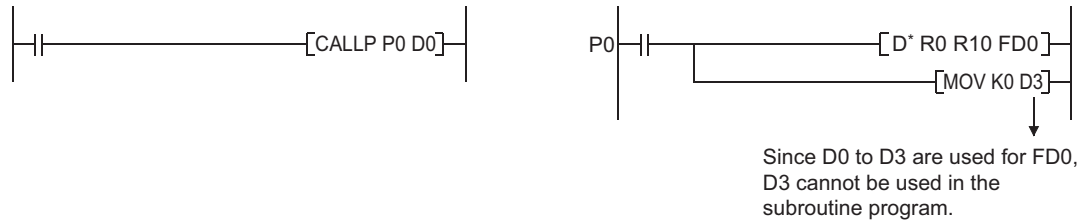


Figure 9.48 Ladder example in which use of devices is not allowed in a subroutine program with argument passing

Remark
For use of function devices, refer to the following.
☞ MELSEC-Q/L Programming Manual (Common Instruction)
.....

9.3.2 Special relay (SM)

(1) Definition

The special relay (SM) is an internal relay of which details are specified inside the programmable controller, and the CPU module status data are stored in this special relay.

(2) Special relay classifications

Table9.3 shows special relay classifications.

Table9.3 Special relay classification list

Classification	Special relay
Diagnostics information	SM0 to SM199
Serial communication function	SM100 to SM115 * ¹
System information	SM200 to SM399
System clocks and counters	SM400 to SM499
Scan information, I/O refresh	SM500 to SM599
Drive information	SM600 to SM699
Instruction related	SM700 to SM799
Debugging	SM800 to SM899
Latch area	SM900 to SM999
A → QnA conversion compatibility	SM1000 to SM1299 * ² * ³
Redundant power supply module information	SM1780 to SM1799* ³

*1: Available only for the Q00UJCPU, Q00UCPU, Q01UCPU, and Q02UCPU. (☞ Section 6.23)

*2: Valid only when "Use special relay/special register from SM/SD1000" is selected in the PLC system tab of the PLC parameter dialog box.

*3: Applicability differs depending on the CPU module version. (☞ Appendix 2)

Remark

For details of the special relay, refer to CHAPTER 12.

9.3.3 Special register (SD)

(1) Definition

The special register (SD) is an internal relay of which details are specified inside the programmable controller, and the CPU module status data (such as error diagnostics or system information) are stored in this special register.

(2) Special register classifications

Table9.4 shows special register classifications.

Table9.4 Special register classification list

Classification	Special register
Diagnostics information	SD0 to SD199
Serial communication function	SD100 to SD111 ^{*1}
System information	SD200 to SD399
System clocks and counters	SD400 to SD499
Scan information	SD500 to SD599
Drive information	SD600 to SD699
Instruction related	SD700 to SD799
Debugging	SD800 to SD899
Latch area	SD900 to SD929
Fuse blown module	SD1300 to SD1399
Check of I/O modules	SD1400 to SD1499
A → QnA conversion compatibility	SD1000 to SD1299 ^{*2 *3}
Redundant power supply module information	SD1780 to SD1799 ^{*3}

*1: Available only for the Q00UJCPU, Q00UCPU, Q01UCPU, and Q02UCPU. (☞ Section 6.23)

*2: Valid only when "Use special relay/special register from SM/SD1000" is selected in the PLC system tab of the PLC parameter dialog box.

*3: Applicability differs depending on the CPU module version. (☞ Appendix 2)

Remark

For details of the special register, refer to CHAPTER 12.

9.4 Link Direct Device (J□\□)

(1) Definition

The link direct device is a device for direct access to the link device in a CC-Link IE controller network module, CC-Link IE Field network master/local module or MELSECNET/H module.

This CPU module can directly write data to or read data from the link device in each network module using sequence programs, regardless of the link refresh of the CPU module.

(2) Specification method and application example

(a) Specification method

1) CC-Link IE controller network module, MELSECNET/H module

Specify a network number and a device number.

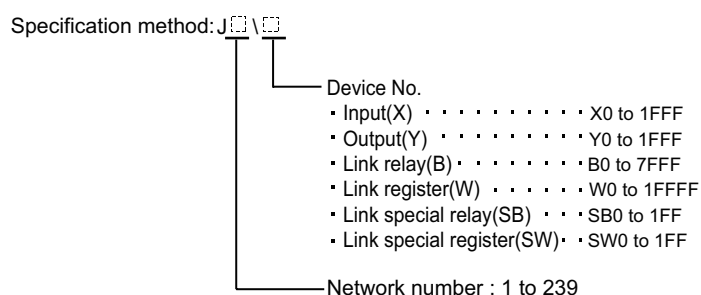


Figure 9.49 Specification method

2) CC-Link IE field network master/local module

Specify a network number and a device number.

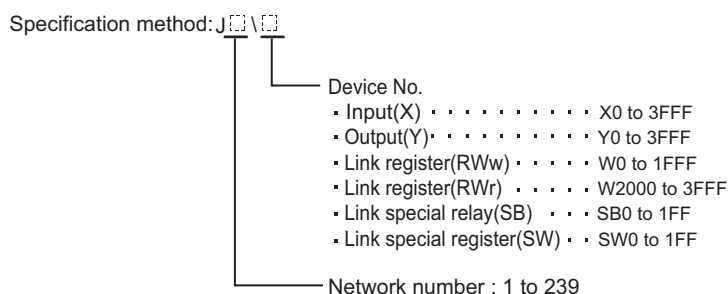


Figure 9.50 Specification method

(b) Application example

For link register 10 (W10) of network number 2, "J2\W10" must be used.

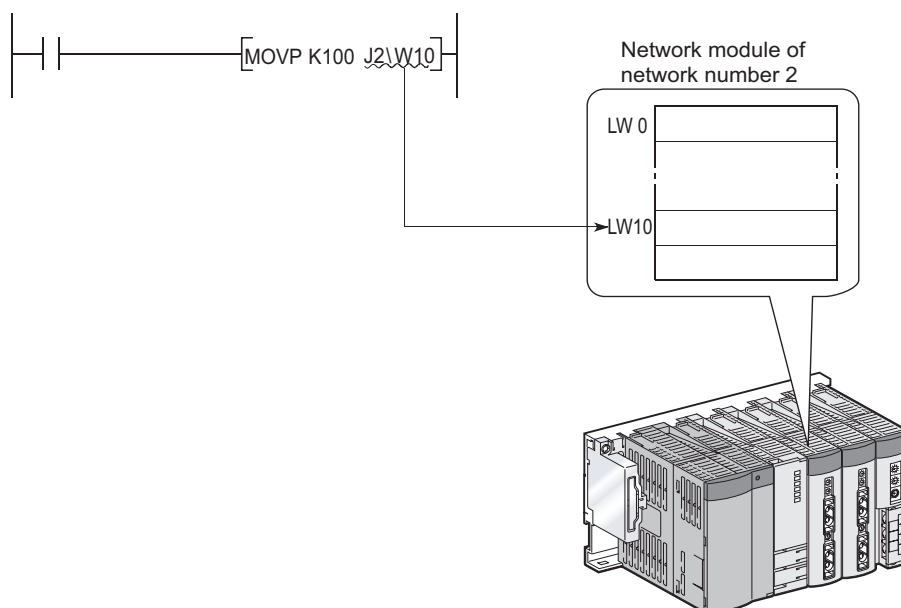


Figure 9.51 Application example

For a bit device (X, Y, B, or SB), the digit must be specified.

Example J1\K1X0, J10\K4B0

(3) Specification range

A link device that is not set in the Network parameter dialog box can be specified.

(a) Writing

- The write range must be within the link device send range that is set by common parameters on Network parameter setting dialog box, and it must be outside the refresh range set by network refresh parameters.

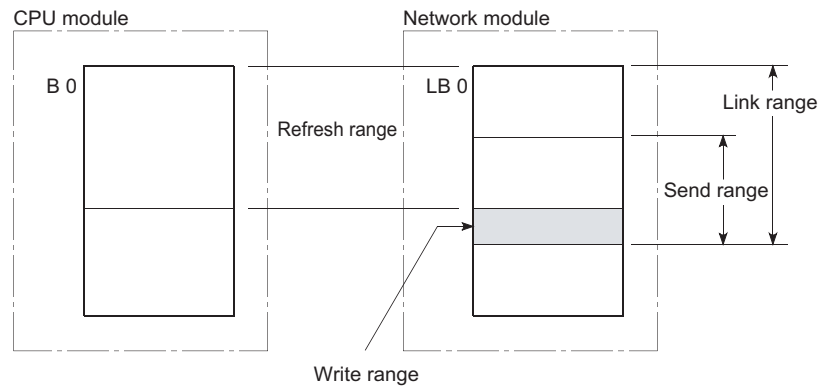


Figure 9.52 Write range of a link direct device

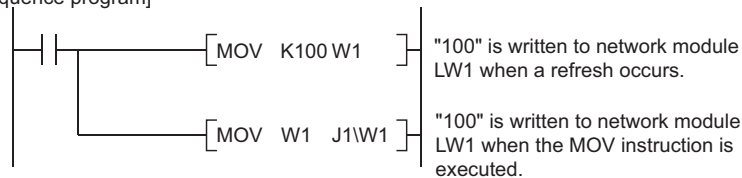
- Although writing can be done to a refresh range portion (specified by refresh parameters) within the link device range, the link module's link device data will be overwritten when a refresh occurs. When writing data by using a link direct device, write the same data to the relevant devices on the CPU module side, which are set by refresh parameters.

[Refresh parameter settings]

1) Network number: 1

2) CPU module (W0 to W3F) ↔ Network module (LW0 to LW3F)

[Sequence program]



[Writing timing]

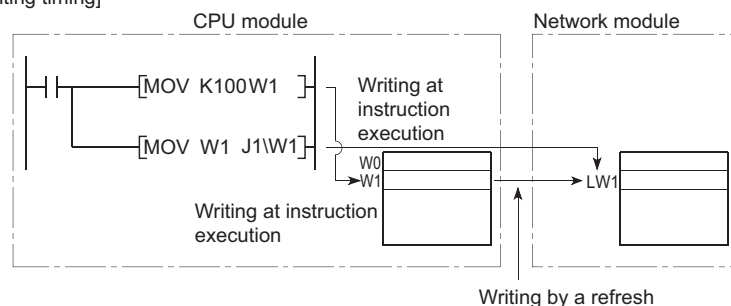


Figure 9.53 Writing to the link device set within the refresh range

- If data are written to another station's write range using a link direct device, the data will be overwritten with other data that are received from the corresponding station.

(b) Reading

The link device ranges of network modules can be read.

Point

Writing or reading data by using a link direct device is allowed for only one network module that is on the same network. If two or more network modules are mounted on the same network, a network module with the lowest slot number is the target of writing or reading by the link direct device. For example, if network modules set as station numbers 1 and 2 are mounted on network number 1 as shown in Figure 9.54, station number 2 is the target of the link direct device.

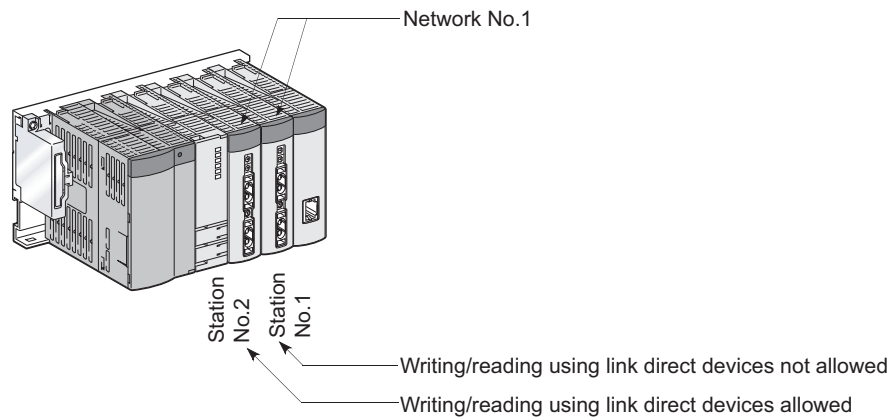


Figure 9.54 When two or more network module are mounted on the same network

(4) Differences between link direct devices and link refresh

Table9.5 Differences between link direct devices and link refresh

Item		Link direct device	Link refresh
Description on programs	Link relay	J□□K4B0 or higher	B0 or higher
	Link register	J□□W0 or higher	W0 or higher
	Link special relay	J□□K4SB0 or higher	SB0 or higher
	Link special register	J□□SW0 or higher	SW0 or higher
Number of steps		2 steps	1 step
Range of network module access		J□□□0 to J□□□3FFF	Range specified by refresh parameters
Guaranteed access data integrity		2 Word (32-bit) units	

Remark

For network parameters, common parameters, and network refresh parameters, refer to the following.

- Details:
 ☞ Network manual for each network module
- Setting method:
 ☞ GX Developer Version 8 Operating Manual

.....

9.5 Module Access Devices

9.5.1 Intelligent function module device (U□\G□)

(1) Definition

The intelligent function module device allows direct access from the CPU module to the buffer memories of the intelligent function modules which are mounted on the main and extension base units.

(2) Specification method and application example

(a) Specification method

Specify the I/O number and buffer memory address of the intelligent function module.

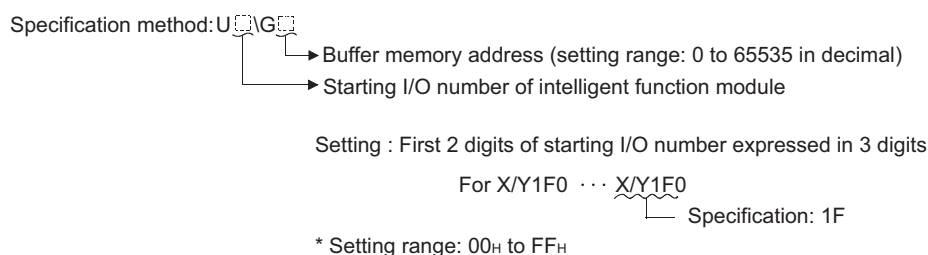


Figure 9.55 Specification method

(b) Application example

When the Q64AD analog-digital converter module is mounted in the position of I/O number 020 (X/Y020 to X/Y02F), to store digital output values of CH.1 to CH.4 into D0 to D3 accordingly, specify the device as shown in Figure 9.55.

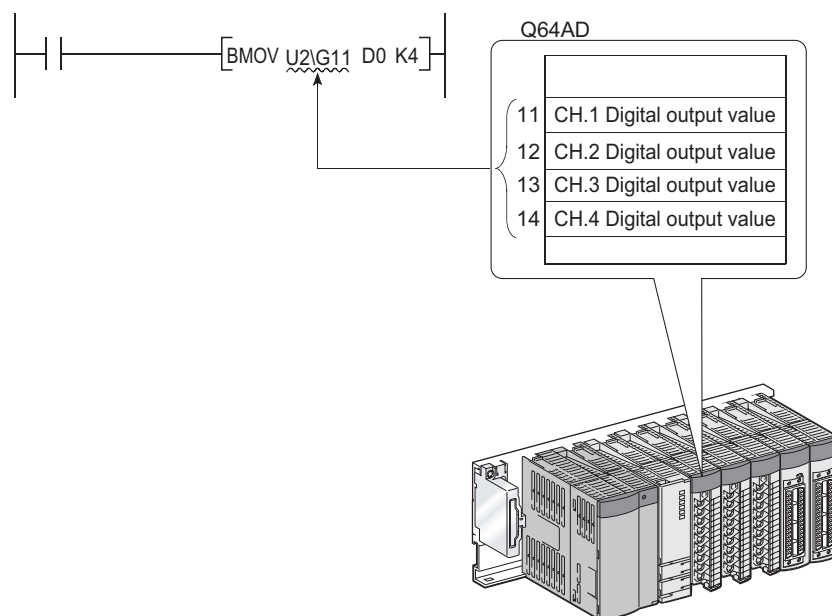


Figure 9.56 Application example



If the intelligent function module device is used, device comments can be attached to the buffer memory.

GX Developer Version 8 Operating Manual

(3) Processing speed

The processing speed of the intelligent function module device is as follows:

- The processing speed of writing or reading using the intelligent function module device is slightly higher compared with the case of using the FROM or TO instruction.

Example "MOV U2\G11 D0"

- When reading from the buffer memory of an intelligent function module and another processing with one instruction, totalize the processing speed of the FROM or TO instruction and the other instruction.

Example "+ U2\G11 D0 D10"

Point

Instead of using the intelligent function module device in the sequence program twice or more to write or read buffer memory data, using the FROM or TO instruction once in one place can increase the processing speed.

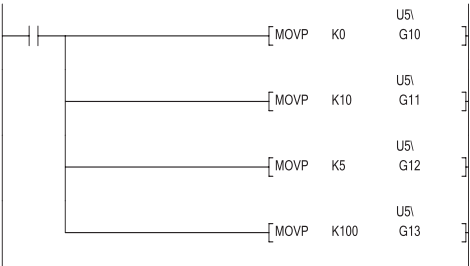


Figure 9.57 Writing data using the intelligent function module device multiple times

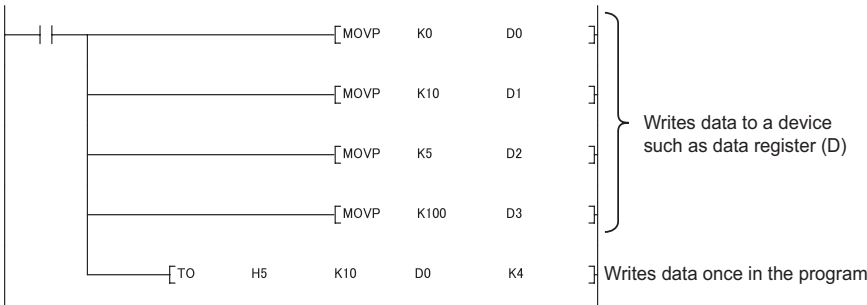


Figure 9.58 Writing data using the TO instruction once in the program

Remark

1. For buffer memory addresses and applications, refer to the manual for each intelligent function module/special function module.
2. For the FROM and TO instructions, refer to the following.
 MELSEC-Q/L Programming Manual (Common Instruction)

9.5.2 Cyclic transmission area device (U3En\G□)

(1) Definition

The cyclic transmission area device is used to access the CPU shared memory of each CPU module in a multiple CPU system.

(2) Features

- The transfer speed is higher than the case of using the write (S.TO or TO) or read (FROM) instruction to the CPU shared memory, resulting in reduced programming steps.
- Using the cyclic transmission area device allows bit manipulation.
- By setting device comments for the cyclic transmission area device, program readability is increased.
- Because information on the CPU shared memory can be directly specified as an argument of the instruction, no interlock device is required.

(3) Specification method

Specify the I/O number of the CPU module and the CPU shared memory address.

Specification method: U3En\G□

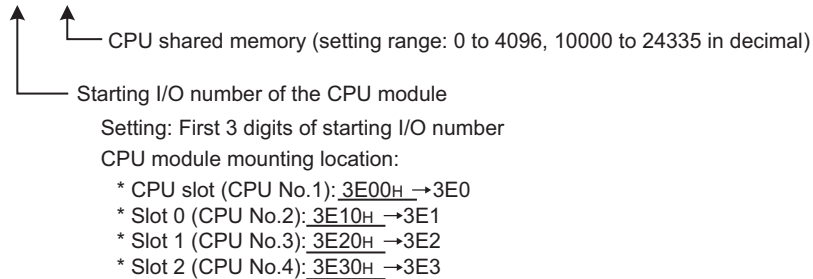


Figure 9.59 Specification method

Remark

For details of the cyclic transmission area device, refer to the following.

☞ QCPU User's Manual (Multiple CPU System)

9.6 Index Register (Z)/Standard Device Resister (Z)

9.6.1 Index register (Z)

(1) Definition

The index register is used for indirect specification (index modification) in sequence programs. Index modification uses one point of the index register.

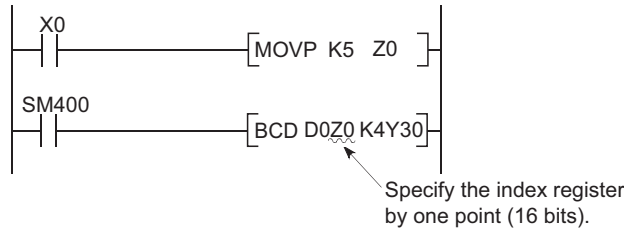


Figure 9.60 Index register

The index register has 20 points (Z0 to Z19).

(2) Bit structure of the index register

(a) Bit structure and read/write unit

One point of the index register consists of 16 bits, and data can be read or written in units of 16 bits.

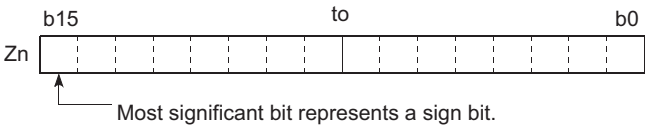


Figure 9.61 Bit structure of the index register

Point

Index register data are handled as signed data.
In the case of the hexadecimal notation, 0000H to FFFFH can be stored. However, because the most significant bit represents a sign bit, decimal values that can be specified are -32768 to 32767.

(b) When using the index register for a 32-bit instruction

The processing target is Z_n and Z_{n+1} .

The lower 16 bits correspond to the specified index register number (Z_n), and the higher 16 bits correspond to the specified index register number + 1.

Example When Z2 is specified in the DMOV instruction, Z2 represents the lower 16 bits and Z3 represents the higher 16 bits. (The most significant bit in a 32-bit structure is a sign bit.)

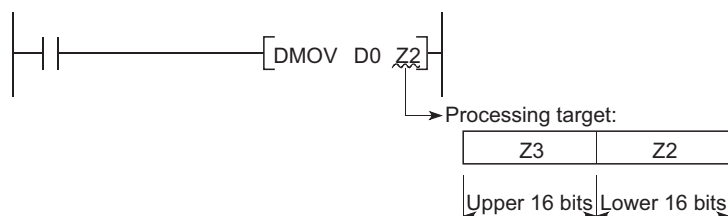


Figure 9.62 Data transfer with a 32-bit instruction and storage location

(3) When using 32-bit index modification

For the file register (ZR), extended data register (D), extended link register (W) using the serial number access method, 32-bit index modification using two points of the index register is available.

The following two kinds of methods can be used to specify the index register.

- Specify the range used for 32-bit index modification.
- Specify the 32-bit index modification using "ZZ". Note9.3

Remark

For details and precautions of index modification using the index register, refer to the following.

MELSEC-Q/L Programming Manual (Common Instruction)



Note9.3

Universal

When specifying the 32-bit index modification using "ZZ" for the Q02UCPU, Q03UDCPU, Q04UDHCPU, Q06UDHCPU, Q13UDHCPU, or Q26UDHCPU, check the versions of the CPU module and GX Developer.

(Appendix 2)

9.6.2 Standard device register (Z)

(1) Definition

By using the index register between register operations, operations can be executed at a higher speed.
The index register used in this case is called the standard device register.

(2) Device number

Since the standard device register is the same device as the index register, pay attention not to use the same device number when using the index modification.

Remark

.....

For operation processing and processing time of the standard device register, refer to the following.

 MELSEC-Q/L Programming Manual (Common Instruction)

.....

9.6.3 Switching from the scan execution type program to the interrupt/fixed scan execution type program

The CPU module performs the following when switching from the scan execution type program to the interrupt/fixed scan execution type program.

- Saving and restoring the index register data
- Saving and restoring block numbers of the file register

(1) Setting for saving and restoration

Saving and restoration setting can be enabled in the PLC system tab of the PLC parameter dialog box.

To disable writing to the index register in the interrupt/fixed scan execution type program, select "High speed execution" in the Interrupt program/Fixed scan program setting area.

If this setting is enabled, the program will switch faster than before.

The screenshot shows the 'Q parameter setting' dialog box with the 'PLC system' tab selected. The 'Timer limit setting' section has 'Low speed' at 100 ms and 'High speed' at 10.00 ms. The 'RUN-PAUSE contacts' section has 'RUN' and 'PAUSE' both set to X. The 'Latch data backup operation valid contact' section has 'Device name' set to X. The 'Remote reset' section has 'Allow' checked. The 'Output mode at STOP to RUN' section has 'Previous state' selected. The 'Floating point arithmetic processing' section has 'Perform internal arithmetic operations in double precision' checked. The 'Intelligent function module setting' section has 'Interrupt pointer setting' button. The 'Module synchronization' section has 'Synchronize intelligent module's pulse up' checked. The 'Service processing setting' section has 'Execute the process as the scan time proceeds' selected. The 'PLC module change setting' section has 'PLC module change setting' button. The 'System interrupt settings' section has 'Interrupt counter start No.' set to 0 and 'Fixed scan interval' set to 0.5ms. The 'High speed interrupt setting' button is highlighted.

Figure 9.63 Interrupt/fixed scan program setting

(2) Processing of the index register

(a) When "High-speed execution" is not selected

1) When switching from the scan execution type program to the interrupt/fixed scan execution type program

The CPU module saves index register values in the scan execution type program, and passes them to the interrupt/fixed scan execution type program.

2) When switching from the interrupt/fixed scan execution type program to the scan execution type program

The CPU module restores the saved index register values.

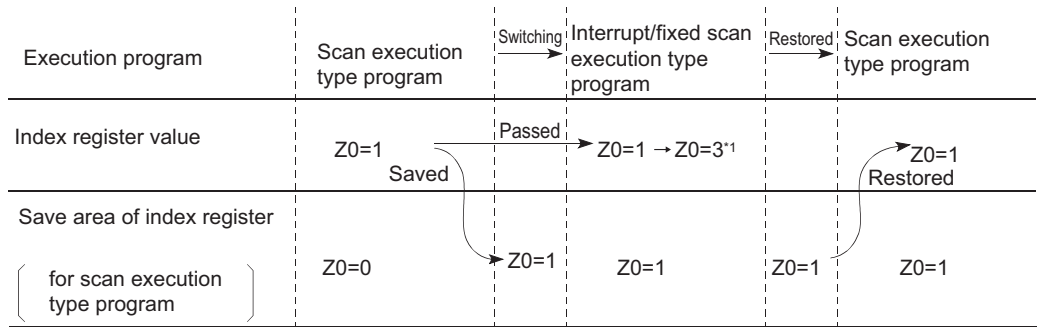


Figure 9.64 Saving and restoring index register data (when "High speed execution" is not selected)

*1: The Z0 value is changed to 3 in the interrupt program.

Point

To pass index register values from the interrupt/fixed scan execution type program to the scan execution type program, use word devices.

9.6 Index Register (Z)/Standard Device Resister (Z)
9.6.3 Switching from the scan execution type program to the interrupt/fixed scan execution type program

(b) When "High-speed execution" is selected

1) When switching from the scan execution type program to the interrupt/fixed scan execution type program

The CPU module does not save/restore any index register values.

2) When switching from the interrupt/fixed scan execution type program to the scan execution type program

If data are written to the index register by the interrupt/fixed scan execution type program, the values of the index register used in the scan execution type program will be corrupted.

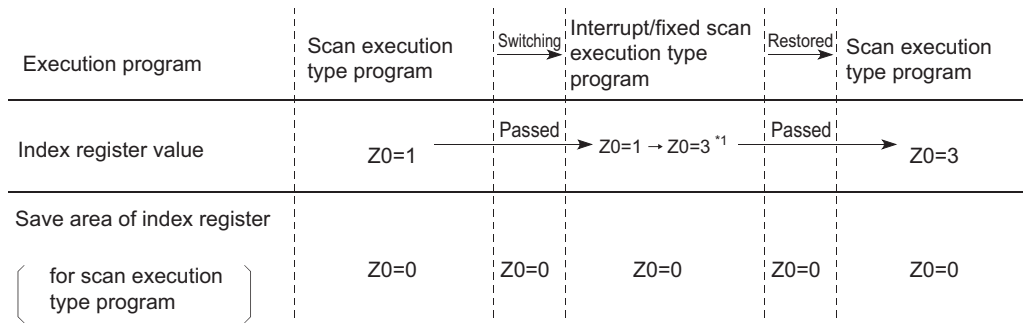


Figure 9.65 Saving and restoring index register data (when "High speed execution" is selected)

*1: The Z0 value is changed to 3 in the interrupt program.

When writing data to the index register by the interrupt/fixed scan execution type program, use the ZPUSH or ZPOP instruction to save and restore the data.

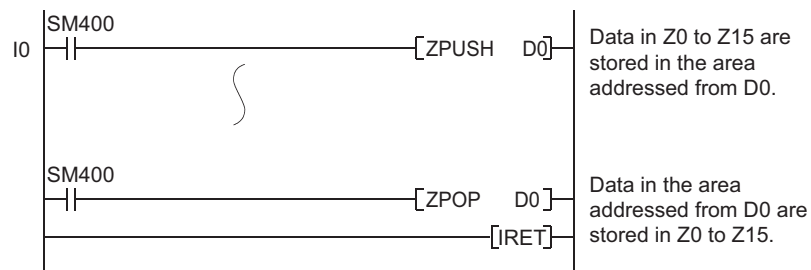


Figure 9.66 Writing data to the index register in the interrupt/fixed scan execution type program

(3) Processing of file register's block numbers

(a) When switching from the scan execution type program to the interrupt/fixed scan execution type program

The CPU module saves the file register block numbers in the scan execution type program, and passes them to the interrupt/fixed scan execution type program.

(b) When switching from the interrupt/fixed scan execution type program to the scan execution type program

The CPU module restores the saved block numbers of the file register.

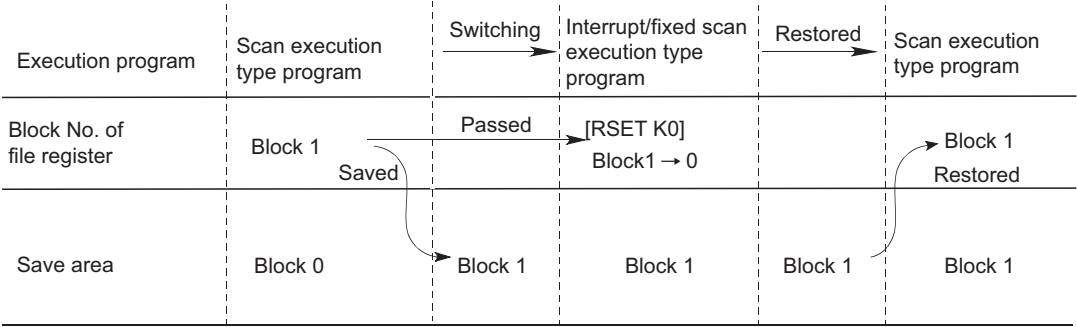


Figure 9.67 Saving and restoring file register's block numbers

9.7 File Register (R) Note9.4

(1) Definition

The file register (R) is a device provided for extending the data register.

The file register can be used at the same processing speed as the data register.

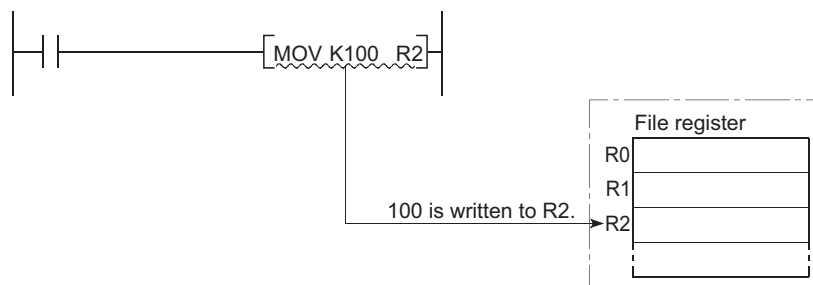


Figure 9.68 Writing to the file register

(2) Bit structure of the file register

(a) Bit structure and read/write unit

One point of the file register consists of 16 bits, and data can be read or written in units of 16 bits.

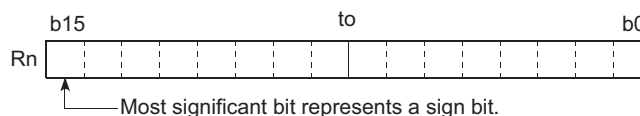


Figure 9.69 Bit structure of the file register

(b) When using a 32-bit instruction for the file register

The processing target is R_n and R_{n+1} .

The lower 16 bits correspond to the file register number (R_n) specified in the sequence program, and the higher 16 bits correspond to the specified file register number + 1.

For example, when R2 is specified in the DMOV instruction, R2 represents the lower 16 bits and R3 represents the higher 16 bits.

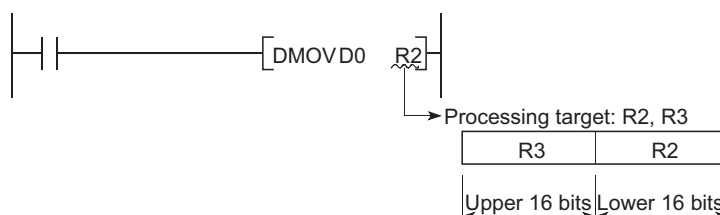


Figure 9.70 Data transfer with a 32-bit instruction and storage location

Data of -2147483648 to 2147483647 or 00000000H to FFFFFFFFH can be stored in a two-point area of the file register. (The most significant bit in a 32-bit structure is a sign bit.)

(3) Clearing the file register

The file register contents are backed up by the battery built in the CPU module, and they are held if the CPU module is powered off or reset.

(Not initialized even if the latch is cleared.*1)

*1: The latch range of the file register can be set in the Device tab of the PLC parameter dialog box. (☞ Section 6.3)

To initialize the file register contents, perform the data clear operation in the sequence program or GX Developer.

(a) When clearing by the sequence program

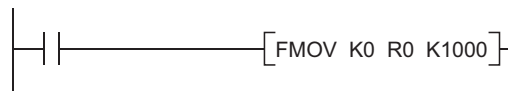


Figure 9.71 Example of clearing the file register R0 to R999

(b) When clearing by GX Developer

Select [Online] → [Clear PLC memory] in GX Developer and clear the data by selecting "Clear all file registers".

9.7.1 File register data storage location

There are the following three kinds of memory media for storing file register data.

- Standard RAM
- SRAM card
- Flash card

9.7.2 File register size

(1) When using the standard RAM

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

Note that, however, if the standard RAM is used for an application other than file registers, available points are decreased. (☞ Section 5.1.10)

Table9.6 File register size

CPU module	Points
Q00UCPU, Q01UCPU, Q02UCPU	64K
Q03UDCPU, Q03UDECPU	96K
Q04UDHCPU, Q04UDEHCPU	128K
Q06UDHCPU, Q06UDEHCPU	384K
Q10UDHCPU, Q10UDEHCPU, Q13UDHCPU, Q13UDEHCPU	512K
Q20UDHCPU, Q20UDEHCPU, Q26UDHCPU, Q26UDEHCPU	640K
Q50UDEHCPU	768K
Q100UDEHCPU	896K

(2) 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 size of the programs and device comments stored in the memory card.

(3) 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 size of the programs and device comments stored in the memory card.

Remark

For the memory cards available for the CPU module, refer to Section 5.1.1.

9.7.3 Differences in available accesses by storage memory

Accesses available for the file register vary for each memory.

Table9.7 Differences in accesses available for the file register

Access		Standard RAM	SRAM card	Flash card
Program writing		○	○	×
Program reading		○	○	○
Writing device memory to programmable controller		○	○	×
Reading device memory from programmable controller		○	○	○
Data modification	Online test operation from GX Developer	○	○	×
	Writing to programmable controller by GX Developer	○	○	×
	Writing to programmable controller by GX Developer (flash ROM)	×	×	○
	Batch write by a serial communication module	○	○	×
	Device data writing from GOT1000 series	○	○	×
	Random write command from GOT1000 series	○	○	×

9.7.4 Registration procedure for the file register

To use a file register, register the file of the file register to the CPU module in the following steps.

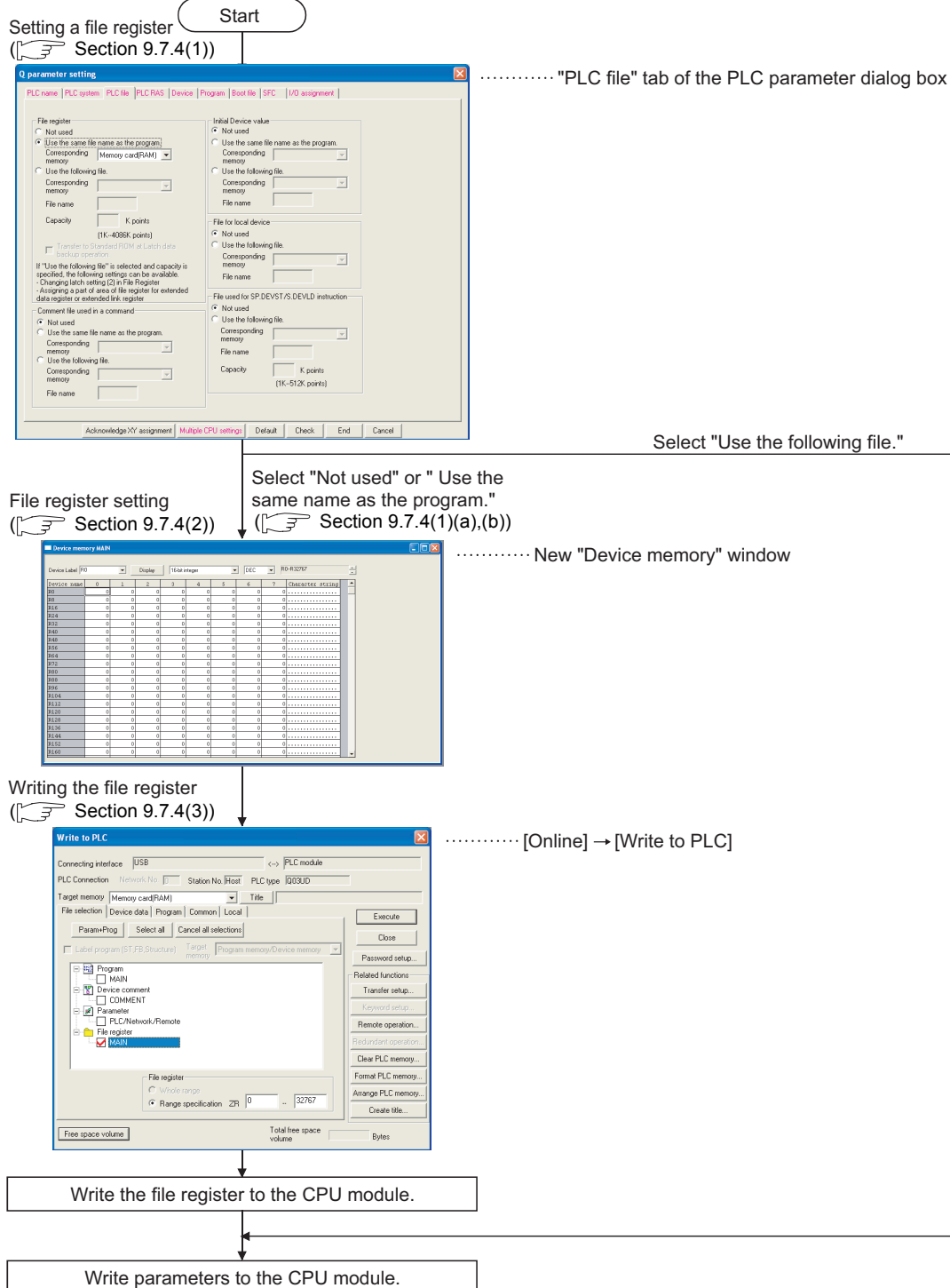


Figure 9.72 Registration procedure for the file register

(1) Setting the file register

In the PLC file tab of the PLC parameter dialog box, specify the standard RAM or a memory card to use the file register in the sequence program.

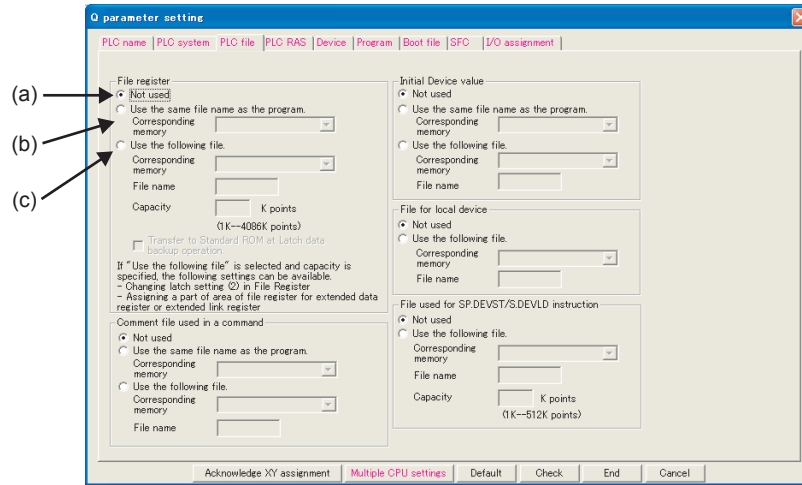


Figure 9.73 File register setting

(a) Not used

Select this in the following cases.

- When not using any file register
- When specifying a file register used in the sequence program (the QDRSET instruction is used for specification.)

(b) Use the same file name as the program.

Select this when executing the file register with the same file name as the sequence program.

1) When the program is changed

The file name of the file register is automatically changed to the same name as the program.

This feature is useful if the file register is exclusively used for one program as a local device.

Example When each of file registers A to C has the same name with the corresponding one of the program A to C, the operation is as described below.

- During execution of program A: Accessing file register A
- During execution of program B: Accessing file register B
- During execution of program C: Accessing file register C

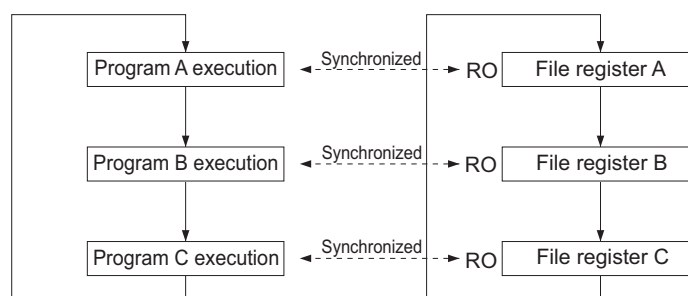


Figure 9.74 When the program is changed

2) Point setting for file registers

Select [Online] → [Write to PLC] in GX Developer and set the number of file register points.

Point

- Only one file register can be created in the standard RAM.
To create more than one, use a SRAM or Flash card.
- With some instructions, file registers set for respective programs cannot be specified.
For details, refer to the pages describing devices available for each instruction in the following manual.
☞ QCPU Programming Manual (Common Instructions)

(c) Use the following file.

Select this when one file register is to be shared by all execution programs.

Specify "Corresponding memory", "File name", and "Capacity" and write these parameters to the CPU module to create a file for the file register.

If the capacity is not specified, note the following.

- When the specified file register file is stored in the specified drive, the file is used. (The capacity is the same as that of the stored file register file.)
- If the file register file with the specified file name is not found on the specified drive, "PARAMETER ERROR" (error code: 3002) will occur.
- For use of an ATA card, "Memory card (ROM) cannot be selected for "Corresponding memory".
(File register data cannot be stored in ATA cards.)
Selecting "Memory card (ROM)" for "Corresponding memory" and writing the settings to the CPU module will result in "PARAMETER ERROR" (error code: 3002).

(2) File register setting

In a new device memory window, set data for the specified file register.

Device Label	R0	Display	16-bit integer	DEC	R0-R32767				
Device name	0	1	2	3	4	5	6	7	Character string
R0	0	0	0	0	0	0	0	0	
R8	0	0	0	0	0	0	0	0	
R16	0	0	0	0	0	0	0	0	
R24	0	0	0	0	0	0	0	0	
R32	0	0	0	0	0	0	0	0	
R40	0	0	0	0	0	0	0	0	
R48	0	0	0	0	0	0	0	0	
R56	0	0	0	0	0	0	0	0	
R64	0	0	0	0	0	0	0	0	
R72	0	0	0	0	0	0	0	0	
R80	0	0	0	0	0	0	0	0	
R88	0	0	0	0	0	0	0	0	
R96	0	0	0	0	0	0	0	0	
R104	0	0	0	0	0	0	0	0	
R112	0	0	0	0	0	0	0	0	
R120	0	0	0	0	0	0	0	0	

Figure 9.75 Device memory window

(a) Devices

Setting R_n (R0 in the case shown above) and clicking the Display button will display the file register list.

(b) Data setting

Enter data that are set for the file register.

This step is not needed when you specify only the capacity of file register.

(3) Registering the file register file to the CPU module

When either of the following is selected in the PLC file tab of the PLC parameter dialog box, the file register file must be registered to the CPU module.

- Not used. (when specifying a file register used in the sequence program) (☞ (1)(a) in this section)
- Use the same name with the program. (☞ (1)(b) in this section)

File register files can be registered to the CPU module by selecting [Online] → [Write to PLC] in GX Developer.

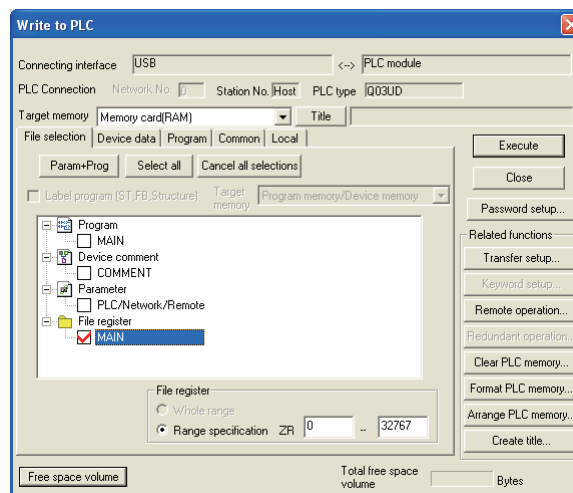


Figure 9.76 File register file registration

(a) Target memory

Select the Standard RAM, Memory card (RAM), or Memory card (ROM) from this list box.

When using the same file name as that of the program, register the file register to the memory specified in the PLC File tab of the PLC parameter dialog box.

(b) Selecting a file register file

By selecting a memory for the file register, file names of the set file registers are displayed. Select a file register file.

(c) Setting the size and file name

Set the size of the file register and a file name of the file register file to be written to the CPU module.

1) File register size of the CPU module

The file register size can be set in increments of one point.

Note that each file size is ensured in units of 256 points.

Even if a file register is not specified from ZR0, the created file will have an assignment from ZR0 to the last number.

For example, if the write range of a file register is specified to be ZR1000 to ZR1791, the created file register file will have an assignment from ZR0 to ZR1791.

However, because the data in ZR0 to ZR999 are unreliable, specify the file register from ZR0.

The size of the file register is checked in the units of 1K points. Therefore, the file register size must be specified from R0 in the units of 1K points.

(d) Writing to the CPU module

The file register file of the specified points is registered to the specified memory of the CPU module.

9.7.5 Specification methods of the file register

(1) Block switching method

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

If 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.

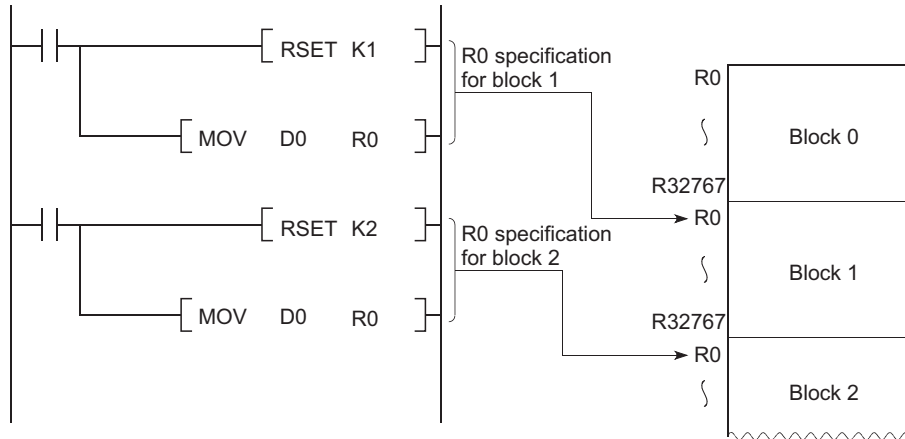


Figure 9.77 Block switching method

(2) Serial number access method

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

Multiple blocks of a file register can be used as a continuous file register.

This kind of device is expressed as "ZR".

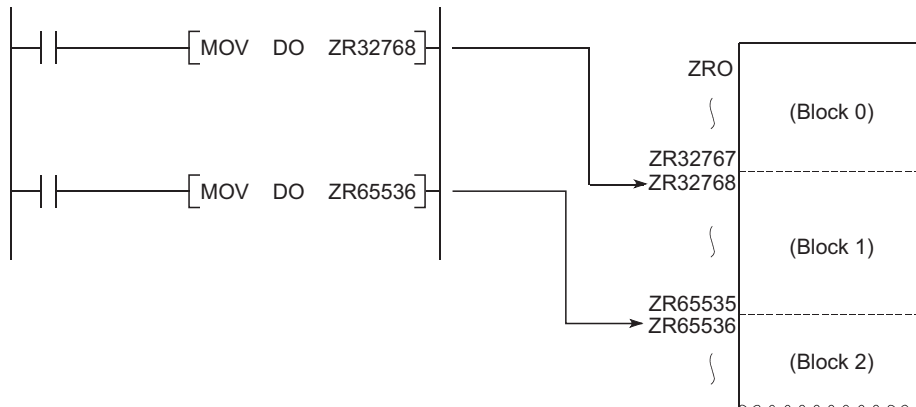


Figure 9.78 Serial number access method



The block numbers and ZR device points that can be specified vary depending on the following.

- Storage location of the file register (☞ Section 9.7.1)
- File register size (☞ Section 9.7.2)

9.7.6 Precautions for using the file register

(1) No registration or use of an invalid file register number

(a) When the file of the file register has not been registered

Writing to or reading from the file register will result in "OPERATION ERROR" (error code: 4101).

(b) When writing to or reading from the file register exceeding the registered size (points)

"OPERATION ERROR" (error code: 4101) will occur.

(2) File register size check

When writing to or reading 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.

(a) Checking the file register size

The file register size can be checked in the File register capacity area (SD647). *1

The file register size data in units of 1K points is stored in this SD647.

*1: If a file register file is switched to another, the size of the currently selected file register file is stored in SD647.

Point

The remainder after dividing the file register size by 1K points is discarded.

To ensure an accurate "range of use" check, specify the file register setting in units of 1K points (1024 points).

(b) Checking timing

- In a program using any file register, check the file register size at step 0.
- After execution of the file register file switching instruction (QDRSET), check the file register size.
- Before executing the file register block switching instruction (RSET), confirm that space of 1K points or more can be ensured after the switching.

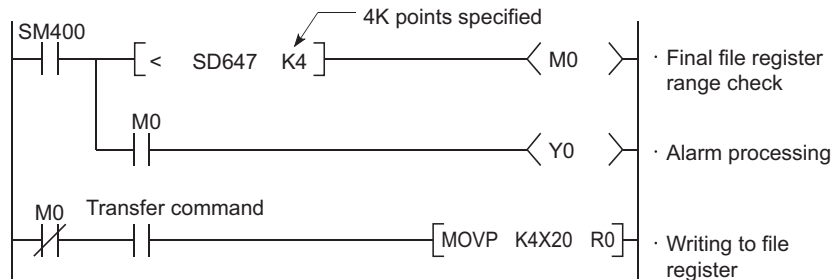
$$(\text{File register size}) > [32\text{K points} \times (\text{Switching block No}) + \{1\text{K points}\}]$$

(c) File register size checking procedure

- Check the file register size used for each sequence program.
- Check the total file register size set in SD647 on the sequence program to see if there are sufficient number of points to be used or not.

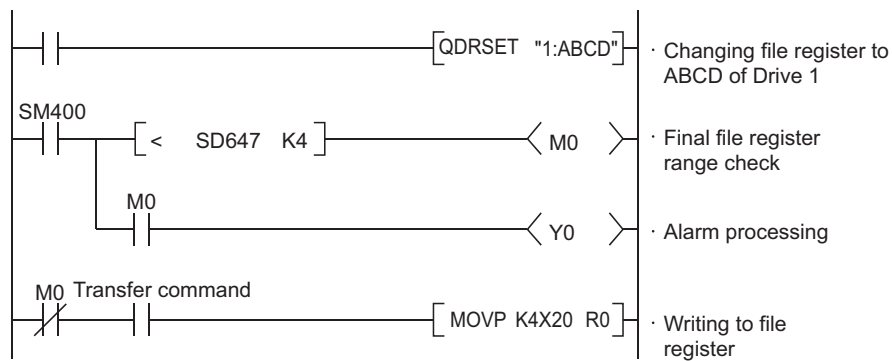
[Program example 1]

The file register range of use is checked at the beginning of each program.



[Program example 2]

The file register range of use is checked after execution of the QDRSET instruction.



[Program example 3]

When a block is switched to another:

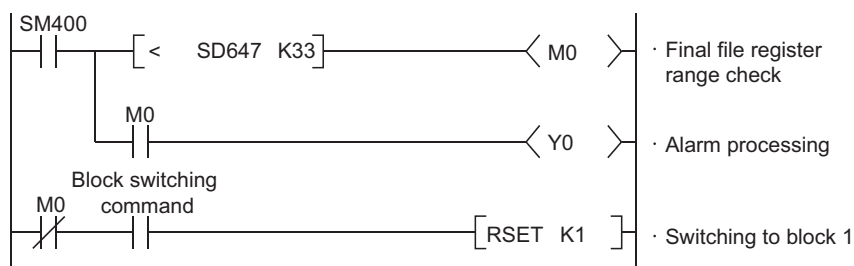


Figure 9.79 Program examples of file register checking

(3) Deleting a file register file

To delete an unnecessary file register file, select [Online] → [Delete PLC data] in GX Developer.

9.8 Extended Data Register (D) and Extended Link Register (W)

Note9.5

(1) Definition

The extended data register (D) and extended link register (W) are devices for using the large-capacity file register (ZR) area as an extended area of the data register (D) and link register (W). These devices can be programmed as the data register (D) and link register (W) together with the file register (ZR) area.

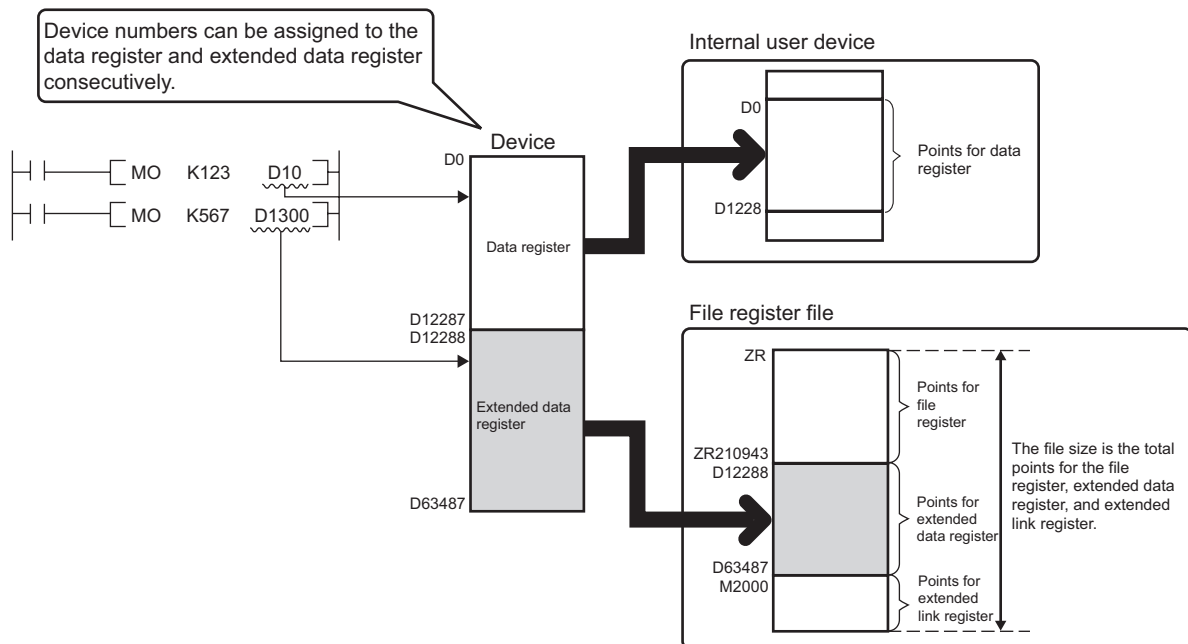


Figure 9.80 Operation of the extended data register (D) and extended link register (W)

Note9.5

Universal

For the Q00JCPU, the extended data register (D) and extended link register (W) cannot be used.
 For using them for the Q02JCPU, Q03UDCPU, Q04UDHCPU, Q06UDHCPU, Q13UDHCPU, Q26UDHCPU, and
 Built-in Ethernet port QCPU, check the version of GX Developer. (Appendix 2)

(2) Device numbers

Device numbers for the extended data register (D) and extended link register (W) can be assigned consecutively after those for the internal user devices, data register (D) and link register (W).

Point

- Even though device numbers are consecutively assigned, there is no physical area contiguity between the data register (D) (internal user device) and the extended data register (D), and between the link register (W) (internal user device) and the extended link register (W).
To use them as one contiguous area, set the points for the data register (D) and link register (W) (internal user device) to "0" in the Device tab of the PLC parameter dialog box, and use only the extended data register (D) and extended link register (W).
 - When using the file register (ZR), extended data register (D) and extended link register (W) with the auto refresh setting enabled in GX Configurator, the points set in the File register extended setting in the Device tab of the PLC parameter dialog box must not be exceeded.
-

(3) Setting method

Since the extended data register (D) and extended link register (W) use the file register area, data must be set for both the file register setting and the device setting.

(a) File register setting

Select "Use the following file." in the PLC file tab of the PLC parameter dialog box, and enter data in the boxes indicated in Figure 9.81.

The "Use the same file name as the program." option is not selectable.

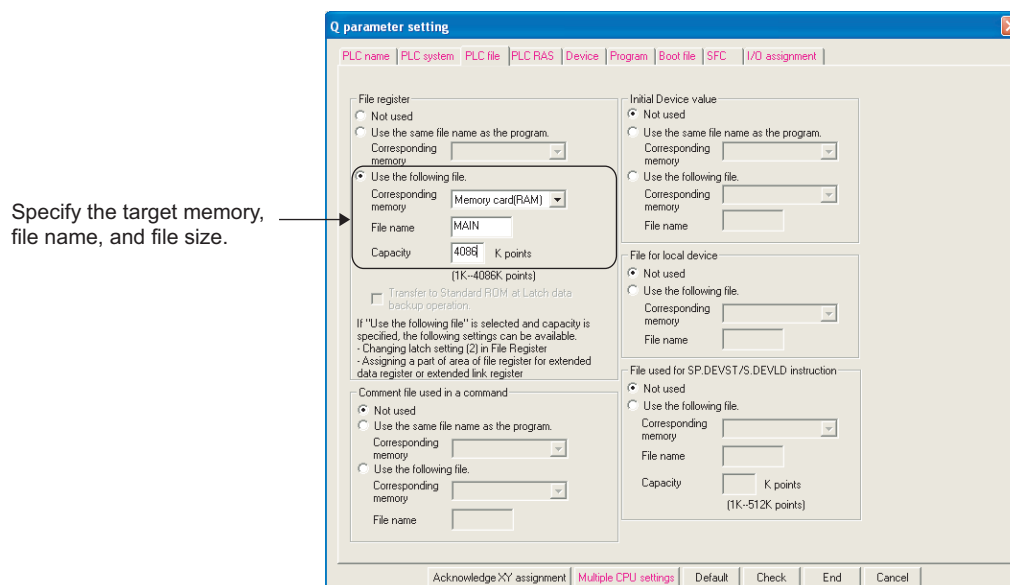


Figure 9.81 Dialog box for file register point setting

Table9.8 File register setting item list

Item	Corresponding memory		File name	Capacity ^{*1}	Remarks
"Use the following file"	Memory card (RAM) ^{*4}		Any name	1 to 4086K points ^{*2}	-
	Memory card (ROM) ^{*4}		Any name	1 to 2039K points ^{*3}	Read only
	Standard RAM	Q00UCPU, Q01UCPU, Q02UCPU	Any name	1 to 64K points	-
		Q03UDCPU, Q03UDECPU		1 to 96K points	
		Q04UDHCPU, Q04UDEHCPU		1 to 128K points	
		Q06UDHCPU, Q06UDEHCPU		1 to 384K points	
		Q10UDHCPU, Q10UDEHCPU, Q13UDHCPU, Q13UDEHCPU		1 to 512K points	
		Q20UDHCPU, Q20UDEHCPU, Q26UDHCPU, Q26UDEHCPU		1 to 640K points	
		Q50UDEHCPU		1 to 768K points	
		Q100UDEHCPU		1 to 896K points	

^{*1}: The total points of the file register (ZR), extended data register (D), and extended link register (W).

^{*2}: This is the maximum number of points when an SRAM card (8M bytes) is used.

^{*3}: This is the maximum number of points when a Flash card (4M bytes) is used.

^{*4}: For the Q00UCPU and Q01UCPU, no memory card can be used.

(b) Device setting

Set each number of points for the file register (ZR), extended data register (D), and extended link register (W) in the File register extended setting in the Device tab of the PLC parameter dialog box.

Assign a part of the points set for the file register (ZR) in the PLC file tab to the extended data register (D) and extended link register (W).

If data are to be latched, specify the latch range.

- Latch (2) of the file register (ZR)
- Latch (1) and (2) of the extended data register (D)
- Latch (1) and (2) of the extended link register (W)

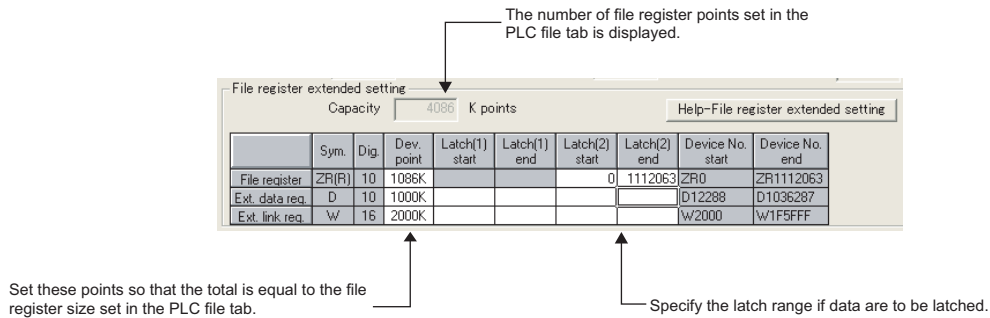


Figure 9.82 File register extended setting

Once the points for the extended data register (D) and extended link register (W) is set, areas for these devices are reserved in the file register file.

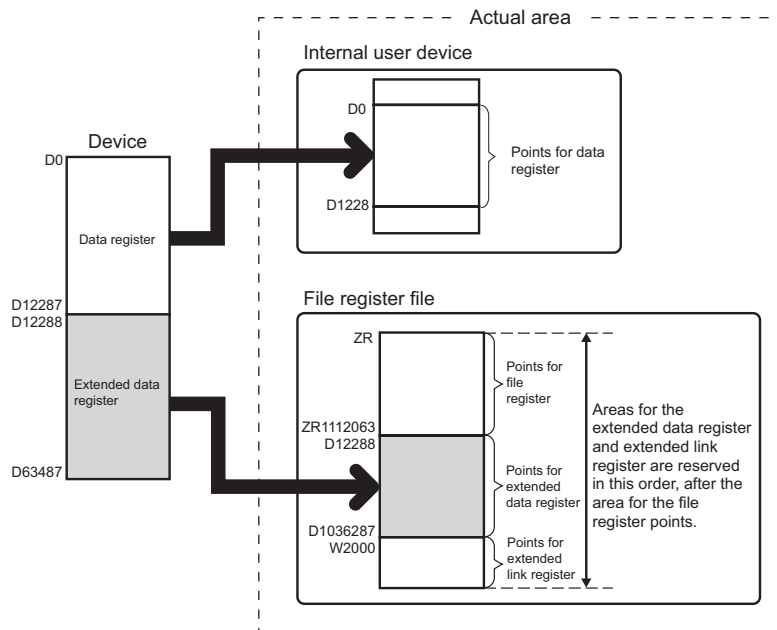


Figure 9.83 Areas for the extended data register (D) and extended link register (W)

(4) Checking the points by the special register

The points for each of the file register (ZR), extended data register (D), and extended link register (W) can be checked in the following special register areas.

- SD306, SD307: File register (ZR)
- SD308, SD309: Extended data register (D)
- SD310, SD311: Extended link register (W)

(5) Precautions

For use of the extended data register (D) and extended link register (W), pay attention to the following.

- 1) When the extended data register (D) and extended link register (W) are specified, the values of the following items will be the same as those for the file register (ZR).
 - Number of program steps
 - Instruction processing time
 - Processing time of auto refresh with network modules
 - Processing time of auto refresh with intelligent function modules
 - Processing time of auto refresh between CPU shared memories
- 2) The file register size cannot be changed in the RUN status.
- 3) The file register cannot be switched to another using the QDRSET instruction. ("OPERATION ERROR" (error code: 4100))
- 4) Set the refresh ranges for the following auto refresh properly so that each refresh range does not cross over the boundary between the internal user device and the extended data register (D) or extended link register (W).
 - Auto refresh with network modules
 - Auto refresh with intelligent function modules
 - Auto refresh between CPU shared memories
- 5) Set the following properly so that each specification does not cross over the boundary between the internal user device and the extended data register (D) or extended link register (W).
 - Index modification
 - Indirect specification
 - Specification for instructions that use block data ^{*1}

*1: Block data means the following:

- Data used in instructions, such as FMOV, BMOV, and BK+, which treat more than one word for operation.
- Control data, composed of two or more words, which are specified in instructions, such as SP.FWRITE and SP.FREAD.
- Data in a 32-bit or greater format (binary 32 bits, real number, indirect address of a device)

Remark

For details on the index modification and indirect specification with the extended data register (D) and extended link register (W), refer to the following.

 MELSEC-Q/L Programming Manual (Common Instruction)

- 6) To access the extended data register (D) or extended link register (W) from a module that does not support the use of these devices, device numbers need to be specified with those of the file register (ZR).
Calculation formulas for obtaining device numbers of the file register (ZR) to be specified to access the extended data register (D) and extended link register (W) and calculation examples are described below.

Table9.9 Calculation formulas for obtaining device numbers of the file register (ZR) to be specified to access the extended data register (D) and extended link register (W) *1

Item	Calculation formula
Device number of the file register (ZR) used to access the extended data register (D)	$ED_{ZN} = ZR_C + (ED_N - D_C)$
Device number of the file register (ZR) used to access the extended link register (W)	$EW_{ZN} = ZR_C + ED_C + (EW_N - W_C)$

*1: Variables in the table indicate the following:

- ZR_C : Points of the file register (ZR)
- ED_{ZN} : Device number of the file register (ZR) used to access the extended data register (D)
- ED_N : Access target device number of the extended data register (D)
- D_C : Points of the data register (D)
- ED_C : Points of the extended data register (D)
- EW_{ZN} : Device number of the file register (ZR) used to access the extended link register (W)
- EW_N : Access target device number of the extended link register (W) (hexadecimal)
- W_C : Points of the link register (W)

[Calculation example]

- D_C : Points of the data register (D) ... 12288 points
- W_C : Points of the link register (W) ... 8192 points
- ZR_C : Points of the file register (ZR) ... 2048 points
- ED_C : Points of the extended data register (D) ... 2048 points

1) Device number of the file register (ZR) used to access D13000

$$ED_{ZN} = 2048 + (13000 - 12288) = 2760$$

2) Device number of the file register (ZR) used to access W2100

$$EW_{ZN} = 2048 + 2048 + (2100_H - 8192) = 2048 + 2048 + (8448 - 8192) = 4352$$

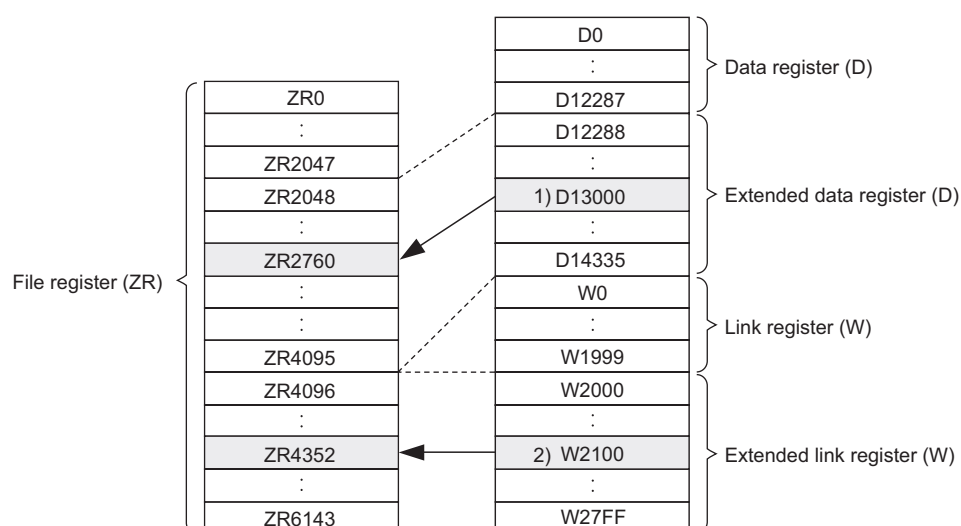


Figure 9.84 Access to the extended data register (D) and extended link register (W)

9.9 Nesting (N)

(1) Definition

Nesting (N) is a device used in the master control instructions (MC and MCR instructions) to program operation conditions in a nesting structure.

(2) Specification method using master control instructions

The master control instruction opens or closes a common ladder gate to switch the ladder of a sequence program efficiently.

Specify the nesting (N) in ascending order (in order of N0 to N14), starting from the outside of the nesting structure.

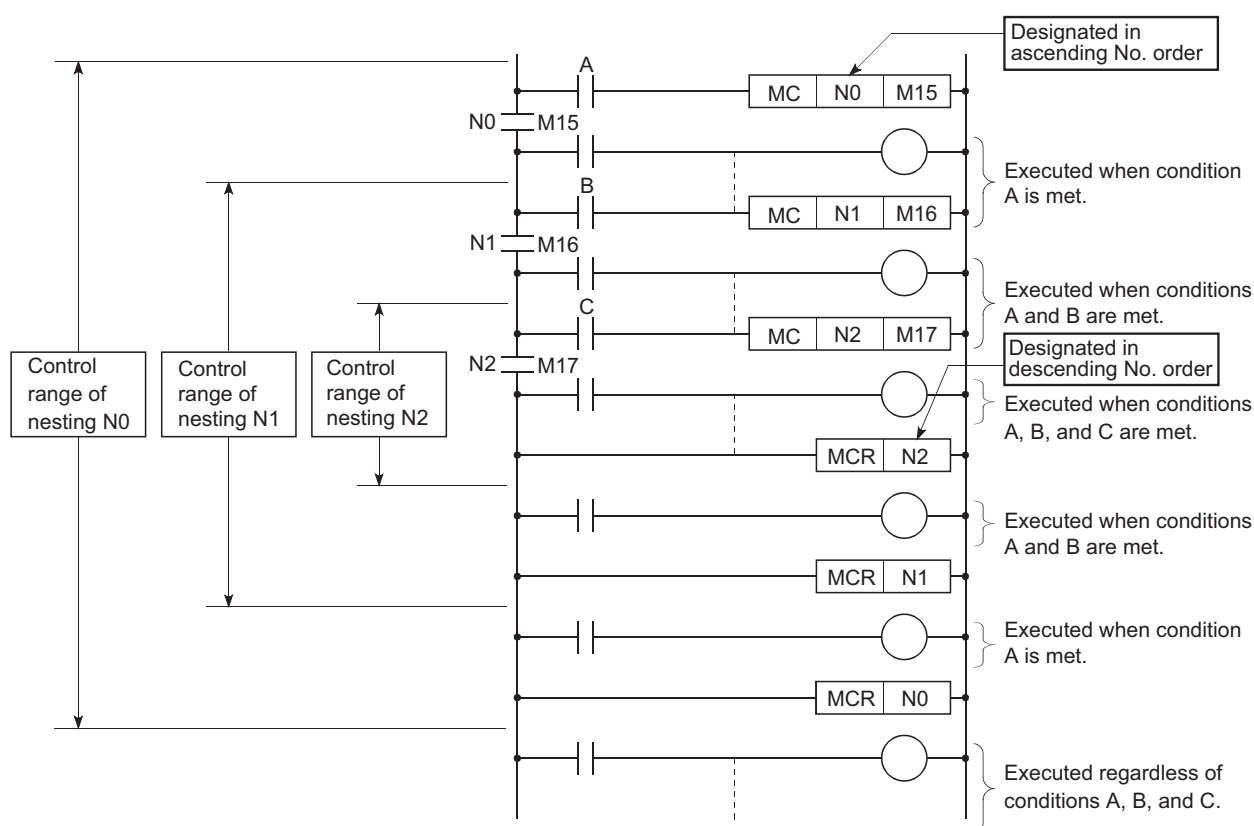


Figure 9.85 Programming example using the nesting

Remark

For use of the nesting, refer to the following.

☞ MELSEC-Q/L Programming Manual (Common Instruction)

9.10 Pointer (P)

(1) Definition

The pointer (P) is a device used in jump instructions (CJ, SCJ, or JMP) or subroutine call instructions (such as CALL).

(2) Applications

Pointers can be used in the following applications.

- Specification of the jump destination in a jump instruction (CJ, SCJ, or JMP) and a label (start address of the jump destination)
- Specification of the call destination of a subroutine call instruction (CALL or CALLP) and a label (start address of the subroutine program)

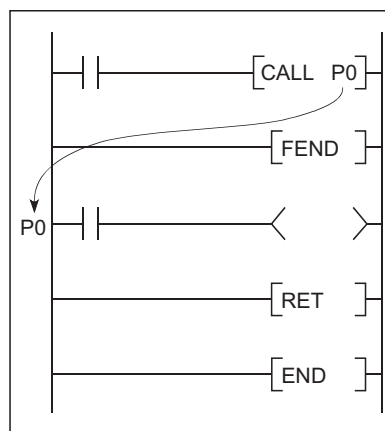


Figure 9.86 Program using a pointer

(3) Pointer types

There are the following two different pointer types.

- Local pointer (👉 Section 9.10.1):
The pointer used independently in each program
- Common pointer (👉 Section 9.10.2):
The pointer that can be called in all running programs by the subroutine call instruction.

(4) Number of available pointer points

The following shows the available number of points.

Table9.10 File register setting item list

CPU module	Points
Q00UJCPU, Q00UCPU, Q01UCPU	512 points
Q02UCPU, Q03UD(E)CPU, Q04UD(E)HCPU, Q06UD(E)HCPU, Q10UD(E)HCPU, Q13UD(E)HCPU, Q20UD(E)HCPU, Q26UD(E)HCPU	4096 points
Q50UDEHCPU, Q100UDEHCPU	8192 points

Remark
For the jump instructions and subroutine call instructions, refer to the following.
 MELSEC-Q/L Programming Manual (Common Instruction)
.....

9.10.1 Local pointer

(1) Definition

The local pointer is a pointer that can be used independently in jump instructions and subroutine call instructions in each program.

The same pointer number can be used in respective programs.

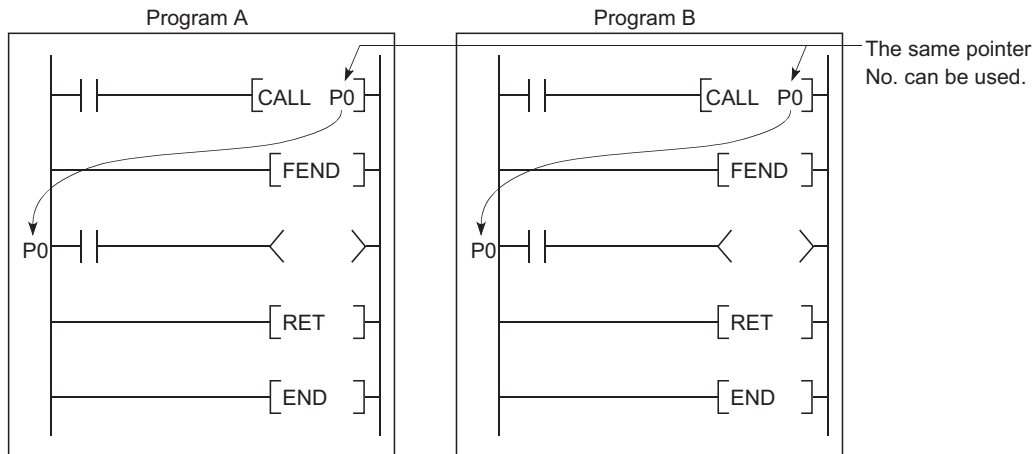


Figure 9.87 Using the same pointer in respective programs (local pointer)

(2) Number of local pointer points

The local pointer can be divided for use of all the programs stored in the program memory.

The local pointer number ranges from P0 to the highest number of the local pointer in use. (The CPU module's system computes the number of points used.)

Even if only P99 is used in a program, for example, the number of points used will be 100, which is from P0 to P99.

For using the local pointer for several programs, use the pointers in ascending order starting from P0 in each program.

Example The total is 600 points when the pointer is used as shown below.

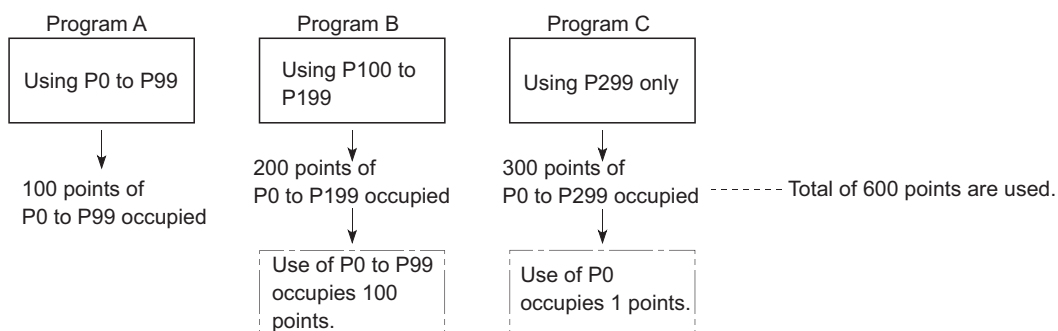


Figure 9.88 Concept of the local pointer points

(3) Precautions for using the local pointer

(a) Program where the local pointer is described

- A jump from another program is not allowed.
- jump instructions and sub-routine CALL instructions.
- Use the ECALL instruction from another program when calling a subroutine program in a program file that contains any local pointer.

(b) Total number of local pointer points

If the total number of pointers (in all programs) exceeds 4096 points, a "Pointer configuration error" (error code: 4020) occurs.

9.10.2 Common pointer

(1) Definition

The common pointer is used to call subroutine programs from all programs that are being executed.

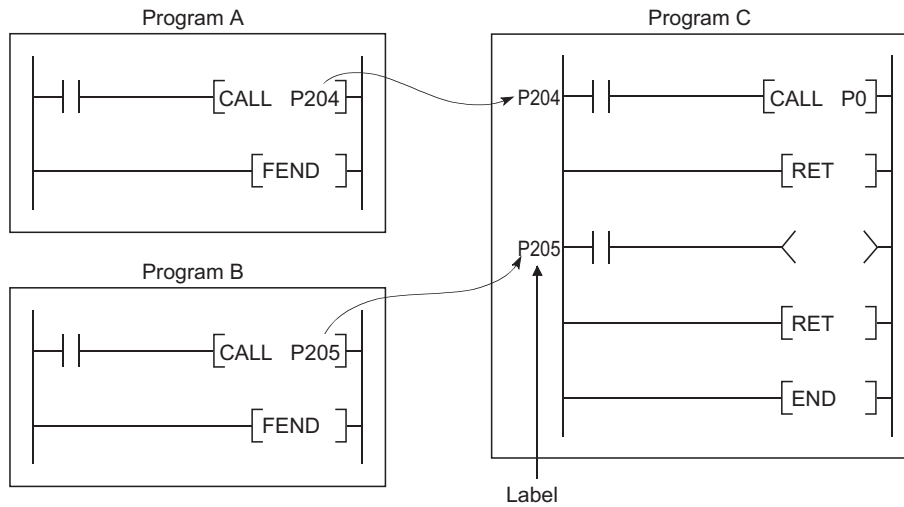


Figure 9.89 Calling pointers in another program (common pointer)

(2) Common pointer range

In the PLC system tab of the PLC parameter dialog box, set the start number for the common pointer.

The common pointer range is from the specified pointer number to P4095.

However, the pointer number that can be entered here is a number higher than the total points used for the local pointer.

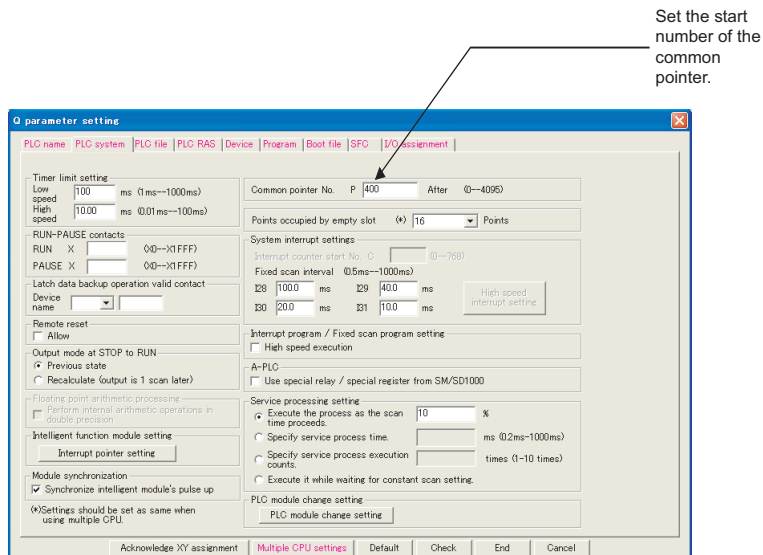


Figure 9.90 Dialog box for setting the common pointer

If a total of 400 points are used in three programs (100 points in each of Program A and Program B, and 200 points in Program C), for example, P400 and higher numbers can be set for the common pointer.

(3) Precautions

- 1) The same pointer number cannot be used as a label.
Doing so will result in a "Pointer configuration error" (error code: 4021).
- 2) If the total number of the local pointer points used in several programs exceeds the start number of the common pointer, a "Pointer configuration error" (error code: 4020) will occur.

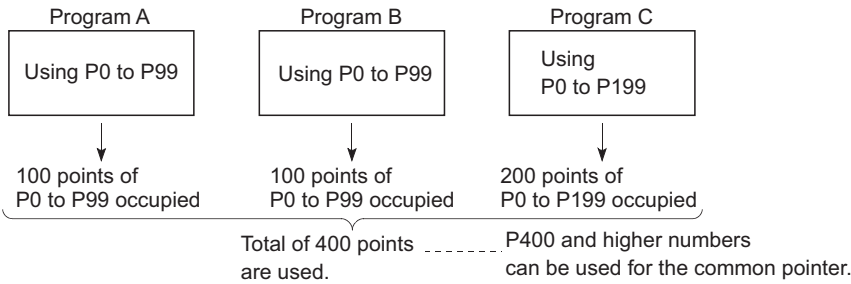


Figure 9.91 Concept of the common pointer range

Point

The jump instructions are not capable of executing a jump to the common pointer in other programs.
Use the common pointer with subroutine call instructions only.

9.11.1 List of interrupt pointer numbers and interrupt factors

The list of interrupt pointer numbers and interrupt factors are shown below.

Table9.12 List of interrupt pointer numbers and interrupt factors

I No.	Interrupt factor	Priority	I No.	Interrupt factor	Priority
I0	Interrupt by interrupt module (QI60)	1st point	I32 to I44	-	N/A
I1		2nd point			
I2		3rd point			
I3		4th point			
I4		5th point			
I5		6th point			
I6		7th point			
I7		8th point			
I8		9th point			
I9		10th point			
I10		11th point			
I11		12th point			
I12		13th point			
I13		14th point			
I14		15th point			
I15		16th point			
I16 to I27	-	N/A	-	N/A	-
I28 ^{*5}	Interrupt by internal timer ^{*1}	100ms	I50 to I255	Intelligent function module interrupt ^{*3 *5} / Interrupt by interrupt module (QI60)	Specify the intelligent function module or interrupt module (QI60) with a parameter. 22 to 227
I29 ^{*5}		40ms			
I30 ^{*5}		20ms			
I31 ^{*5}		10ms			

*1: The time-limit value of the internal timer is set by default.

In the PLC system tab of the PLC parameter dialog box, the value can be changed within the range of 0.5ms to 1000ms in increments of 0.5ms.

*2: This is available for multiple CPU system configuration.

*3: To use the intelligent function module interrupt, the intelligent function module setting (interrupt pointer setting) is required in the PLC system tab of the PLC parameter dialog box. (For interrupt from an intelligent function module, refer to Section 6.22.

*4: I50 has the highest priority (priority 22), and I255 has the lowest priority (priority 227).

*5: When an interrupt occurs, even if no interrupt pointer exists on the program, CANIT EXECUTE(I) (error code: 4220) does not occur.

9.12 Other Devices

9.12.1 SFC block device (BL)

The SFC block is used to check that the specified block in the SFC program is activated.

Remark

For use of the SFC block device, refer to the following.

☞ MELSEC-Q/L/QnA Programming Manual (SFC)

9.12.2 Network No. specification device (J)

(1) Definition

The network No. specification device is used to specify the network number in the link dedicated instructions.

(2) Specification method

In the link dedicated instruction, this device is specified as shown in Figure 9.93.

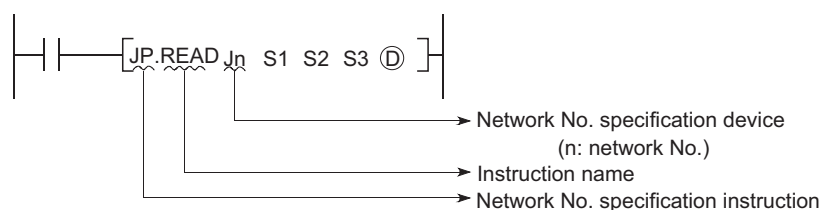


Figure 9.93 How to use the network No. specification device

Remark

For details of the link dedicated instructions, refer to the following.

☞ Reference manual for each network module

9.12.3 I/O No. specification device (U)

(1) Definition

The I/O No. specification device is used to specify I/O numbers in the intelligent function module dedicated instructions.

(2) Specification method

In the intelligent function module dedicated instruction, this device is specified as shown in Figure 9.94.

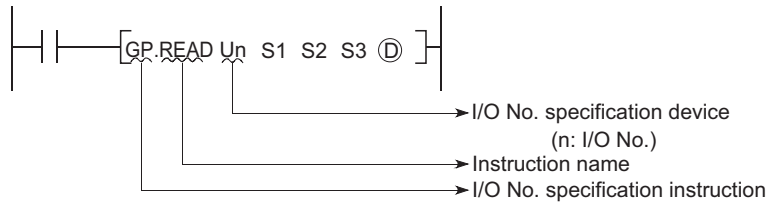


Figure 9.94 How to use the I/O No. specification device

Remark
 For details of the intelligent function module dedicated instructions, refer to the following.
 ➞ Manual for the intelligent function module used

9.12.4 Macro instruction argument device (VD)

(1) Definition

The macro instruction argument device (VD) is used with ladders registered as macros.

When a VD□ setting is specified, the value is converted to the specified device when the macro instruction is executed.

(2) Specification method

Among the devices used in the ladders registered as macros in GX Developer, specify a device used for VD.

When using macro instructions in the sequence program, specify devices that correspond to the macro instruction argument devices used in the macro registration ladders in ascending order.

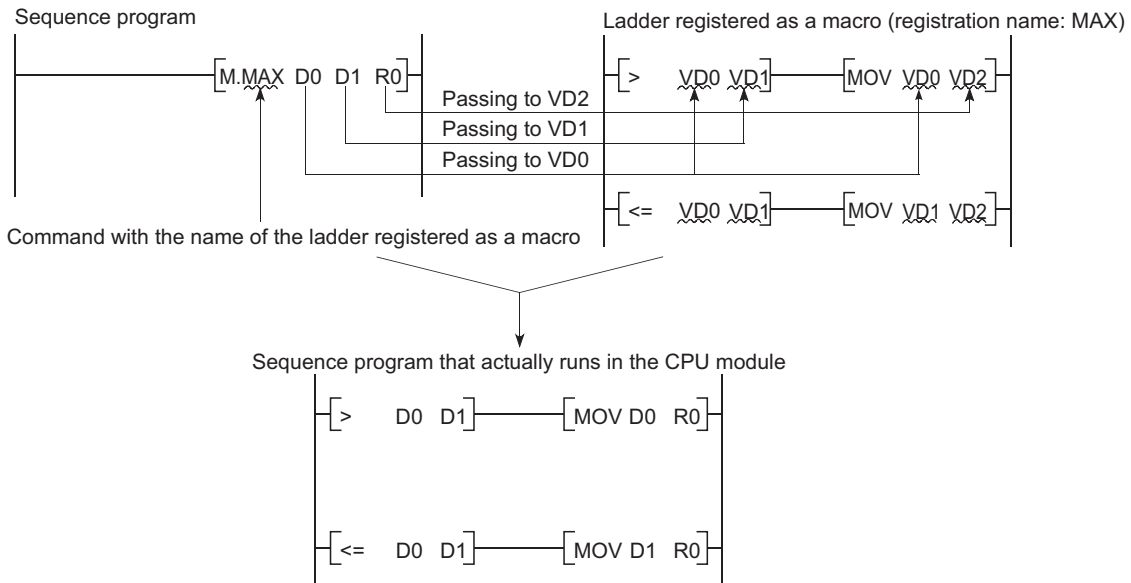


Figure 9.95 Macro instruction argument device specification

Point

- With the macro instruction argument device, VD0 to VD9 can be used in one macro registration ladder.
- The GX Developer read mode provides an option to view a program in macro instruction format. To change the display, select [View] → [Display macro instruction format].

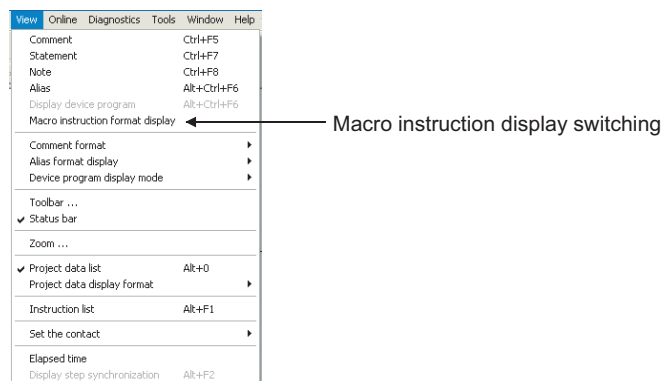


Figure 9.96 Setting for macro instruction display

9.13 Constants

9.13.1 Decimal constant (K)

(1) Definition

The decimal constant (K) is used to specify decimal data in sequence programs.

Specify it as K□□□ (example: K1234) in sequence programs.

In the CPU module, data are stored in binary (BIN). (☞ Section 2.4.1)

(2) Specification range

The specification ranges for decimal constants are as follows:

- When using word data (16-bit data) K-32768 to K32767
- When using 2-word data (32-bit data) K-2147483648 to K2147483647

Point

The most significant bit represents a sign bit.

9.13.2 Hexadecimal constant (H)

(1) Definition

The hexadecimal constant (H) is a device for specifying hexadecimal or BCD data in sequence programs.
(For BCD data, each digit of a hexadecimal number is specified with 0 to 9.)

In sequence programs, specify it as H□□□ (example: H1234). (☞ Section 2.4.2)

(2) Specification range

The specification ranges for hexadecimal constants are as follows:

- When using word data (16-bit data) ... H0 to HFFFF (For BCD data, H0 to H9999)
- When using 2-word data (32-bit data) ... H0 to HFFFFFFFF (For BCD data, H0 to H99999999)

9.13.3 Real number (E)

(1) Definition

The real number (E) is a device used to specify real numbers in sequence programs.

In sequence programs, specify it as E□□□ (example: E1.234). (☞ Section 2.4.4)



Figure 9.97 Real number specification

(2) Specification range

(a) Real number setting range

- For single-precision floating-point data
 $-2^{128} < \text{Device} \leq -2^{-126}, 0, 2^{-126} \leq \text{Device} < 2^{128}$
- For double-precision floating-point data
 $-2^{1024} < \text{Device} \leq -2^{-1022}, 0, 2^{-1022} \leq \text{Device} < 2^{1024}$

(b) When an overflow or underflow has occurred

Table9.13 shows the operation of the CPU module when an overflow or underflow has occurred during arithmetic operation.

Table9.13 When an overflow or underflow has occurred

Overflow	Underflow
OPERATION ERROR (error code: 4141)	Turned to 0 without any error

(c) When a special value^{*1} is input

If operation is performed with input data that contains a special value, "OPERATION ERROR" (error code: 4140) occurs.

*1: The special values are -0, unnormalized numbers, nonnumeric characters, and $\pm \infty$.

(3) Specification method

Real numbers can be specified in sequence programs by the following expressions.

- Normal expression •• A numeric value can be specified as it is.
Example 10.2345 can be specified as E10.2345.
- Exponential expression •• A numeric value is specified by (Value) $\times 10^n$.

Example 1234 is specified as E1.234 + 3.^{*1}

*1: + 3 represents 10^3 in E1.234 + 3.

9.13.4 Character string (" ")

(1) Definition

The character string is a device used to specify a character string in sequence program.
Characters enclosed in quotation marks (example: "ABCD1234") are specified.

(2) Available characters

All ASCII code characters can be used in character strings.
The CPU module distinguishes between upper and lower case characters.

(3) Number of specified characters

A string from the specified character to the NUL code (00H) is one unit.
Note that, however, up to 32 characters can be specified for an instruction using a character string, such as \$MOV.

9.14 Convenient Usage of Devices

When multiple programs are executed in the CPU module, each program can be executed independently by specifying an internal user device as a local device.

Devices of the CPU module are classified into the following two types:

- Global device that can be shared by multiple programs that are being executed.
- Local device that is used independently for each program.

9.14.1 Global device

Programs being executed in the CPU module can share the global device.

Global device data are stored in the device memory of the CPU module, and can be shared by all programs.

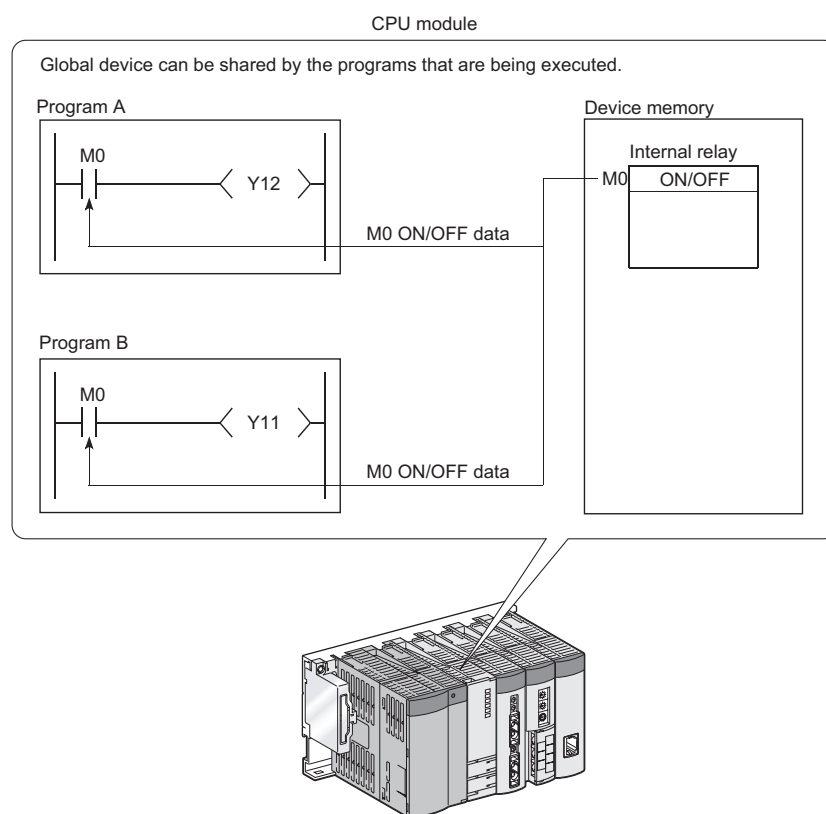


Figure 9.98 Using a global device



- All of the devices that have not been set as local devices (☞ Section 9.14.2) are global devices.
- For execution of multiple programs, the range to be shared by all programs and the range to be used independently by each program (☞ Section 9.14.2) must be specified in advance.

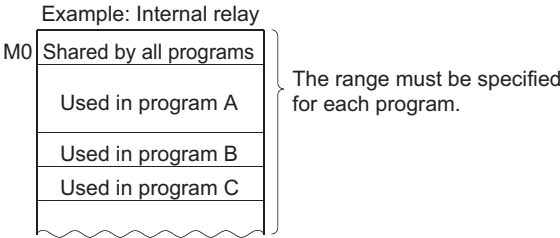


Figure 9.99 Range specification for a device

9.14.2 Local device Note9.6

The local device is a device that can be used independently for each program.
Using local devices allows programming of multiple independently-executed programs without considering other programs.
Note that local device data can be stored in the standard RAM and a memory card (SRAM) only.

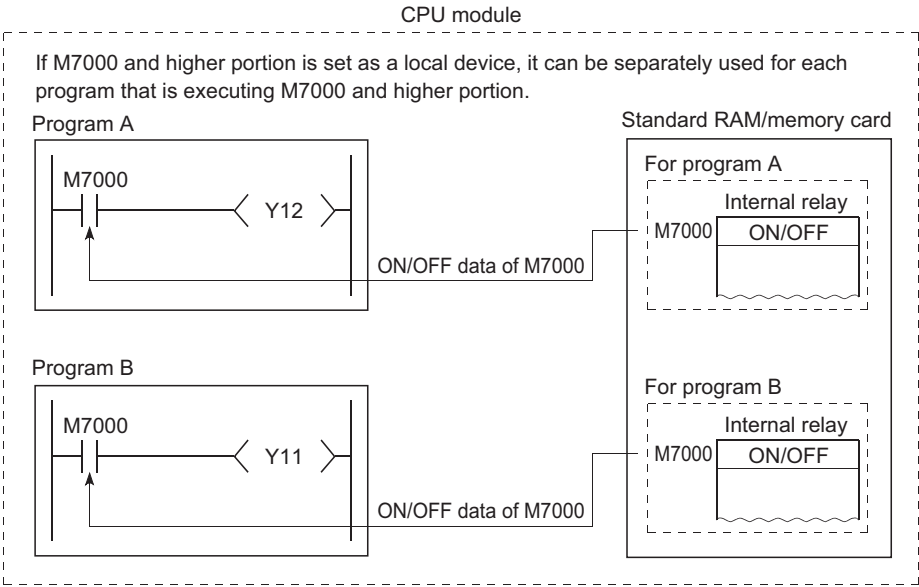


Figure 9.100 Using local devices



Note9.6

Universal

For the Q00UJCPU, local devices cannot be used.

(1) Devices that can be used as local devices

The following devices can be used as local devices.

- Internal relay (M)
- Edge relay (V)
- Timer (T, ST)
- Counter (C)
- Data register (D)
- Index register (Z)  Note9.7

(2) Saving and restoring a local device file

When some programs use a local device, respective local device file data in the standard RAM or a memory card (SRAM) are exchanged with the device memory data of the CPU module after execution of each program.

For this reason, the scan time increases by the time spent for data exchange.

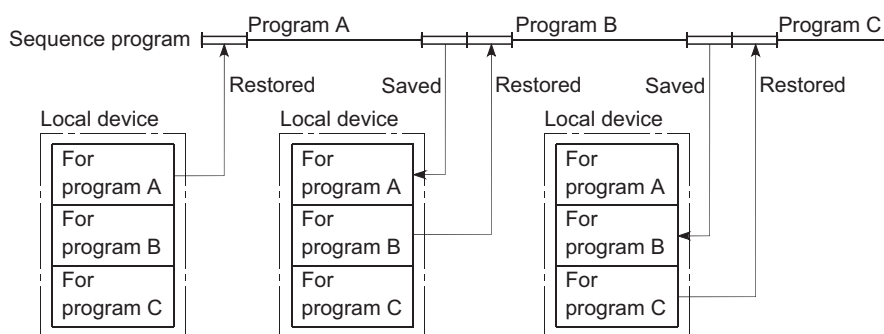


Figure 9.101 Saving and restoring local device files

Remark

1. There are some instructions for which a local device cannot be specified.
For details, refer to the pages describing devices available for each instruction in the following manual.
 MELSEC-Q/L Programming Manual (Common Instruction)
2. For the concept of the number of words used for the local devices, refer to Section 9.2.

Note9.7

Universal

When using the index register as a local device for the Q02UCPU, Q03UDCPU, Q04UDHCPU, Q06UDHCPU, Q13UDHCPU, Q26UDHCPU, and Built-in Ethernet port QCPU, check the versions of the CPU module and GX Developer. ( Appendix 2)

(3) Local device setting

(a) Setting the local device range

In the Device tab of the PLC parameter dialog box, set the range that is used as a local device.

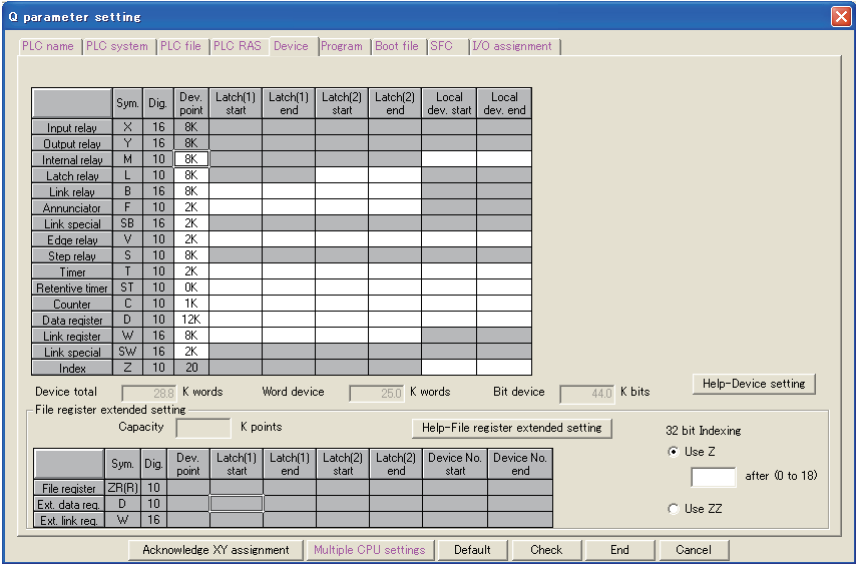


Figure 9.102 Device

Note that the local device range is common to all programs, and cannot be changed for each program. For example, if a local device range is specified as M0 to M100, this range setting applies to all programs that use the local device.

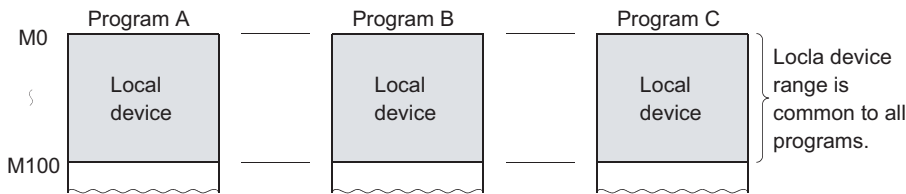


Figure 9.103 Local device range

Point

- The 32-bit index modification range must not overlap with the local device setting range of the index register. If overlapped, 32-bit index modification values will be written over the local device values.
- When CPU module parameters which contains local device setting of the index register are read out from the GX Developer that does not support the setting, all of the index register data will be read out as global device data.

(b) Setting the drive and file name

After setting the local device range, set a memory for storing the local device file and a file name in the PLC file tab of the PLC parameter dialog box.

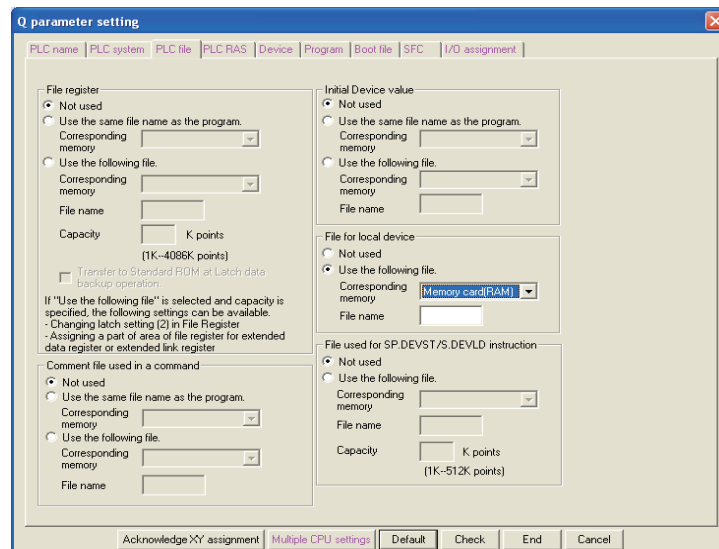


Figure 9.104 PLC file

(c) Writing the setting data

Write the data set in (a) and (b) to the CPU module.

Select [Online] → [Write to PLC] in GX Developer.

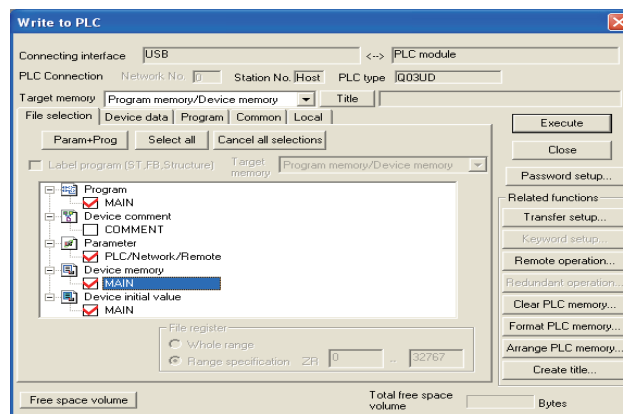


Figure 9.105 Device memory writing

Point

- If the size setting of the local device in the standard RAM is changed with a sampling trace file stored in the standard RAM, the sampling trace file is cleared.
To save the trace results in your personal computer, perform the following operations.
 - 1) Click the **Trace result PLC read** button on the Sampling trace dialog box to read the trace result into the personal computer. (☞ Section 6.14(5)(e))
 - 2) Click the **Trace result** button to display the trace result.
 - 3) Click the **Create CSV file** button to store the trace results in CSV format.
- All of the devices that have not been set as local devices are global devices.

(4) Setting of whether to use a local device (for each program) Note9.8

Use of the local device can be set for each program, and this function can reduce the scan time.

Also, since the area for saving and restoring data is not required for the programs not using a local device, the local device file size can be reduced.

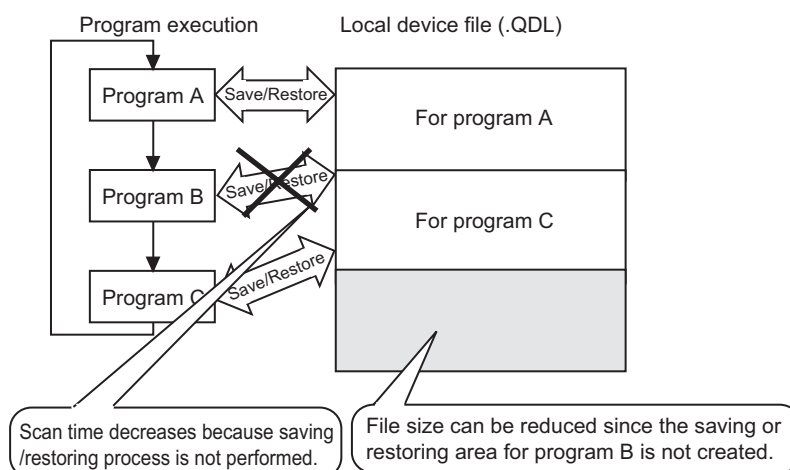


Figure 9.106 Save area configuration of a local device file



Note9.8

Universal

For setting a local device for each program in the Q02UCPU, Q03UDCPU, Q04UDHCPU, Q06UDHCPU, Q13UDHCPU, and Q26UDHCPU, check the versions of the CPU module and GX Developer. (☞ Appendix 2)

(a) Setting method

In addition to the setting in (3) in this section, set the following.

Select the File usability setting button in the Program tab of the PLC parameter dialog box, and specify the programs that use the local device.

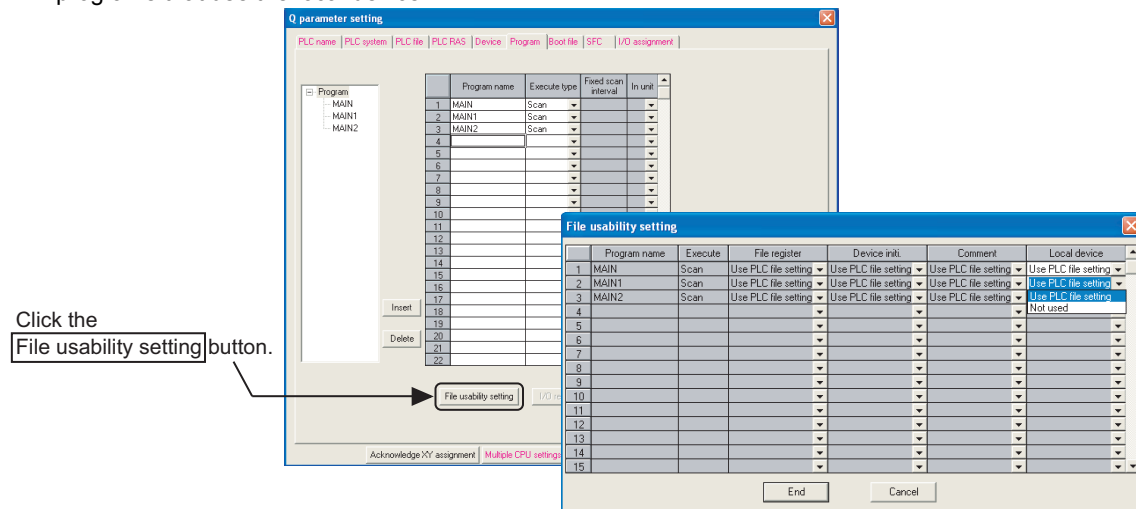


Figure 9.107 File usability setting dialog box

(b) Precautions

1) Change of the local device

Do not change or refer to the local device in a program for which the local device is set to "Not used". Even if the local device is changed in such a program, the changed data will not be held.

2) Conditions for creating a local device file

Creation of a local device file depends on the PLC parameter settings.

Table9.14 shows the conditions for creating a local device file.

Table9.14 Conditions for creating a local device file

PLC parameter setting			File creation	Error detection
PLC file setting	Device setting ^{*1}	File usability setting		
Set	Set	Use PLC file setting	○	-
		Not used	○	-
	Not set	Use PLC file setting	×	-
		Not used	×	-
Not set	Set	Use PLC file setting	×	PARAMETER ERROR (error code: 3000)
		Not used	×	-
	Not set	Use PLC file setting	×	-
		Not used	×	-

○ : Created, × : Not created

^{*1}: Indicates the local device range setting in the Device tab.

(5) Using the local device corresponding to the file where a subroutine program is stored

When executing a subroutine program, you can utilize the local device corresponding to the file where the subroutine program is stored.

Use of the relevant local device is set by ON/OFF of SM776.

Table9.15 Local device switching by ON/OFF of the special relay (SM776)

SM776	Operation
OFF	Perform operations with the local device that corresponds to the source file of the subroutine program.
ON	Perform operations with the local device that corresponds to the file where the subroutine program is stored.

(a) When SM776 is off

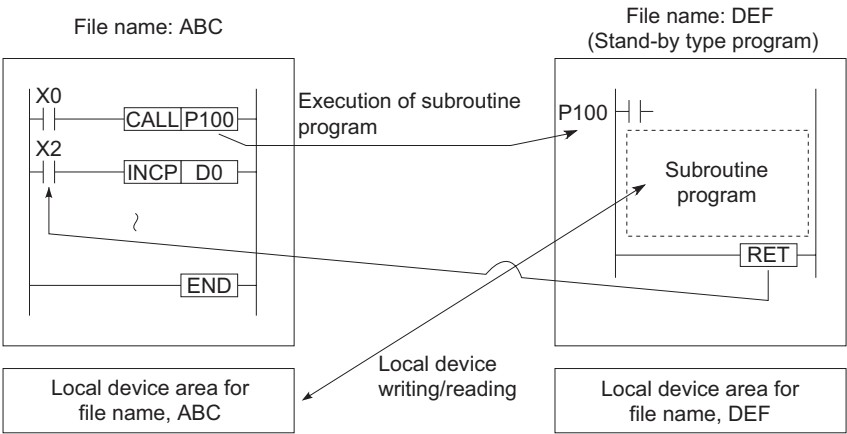


Figure 9.108 When SM776 is off

(b) When SM776 is on

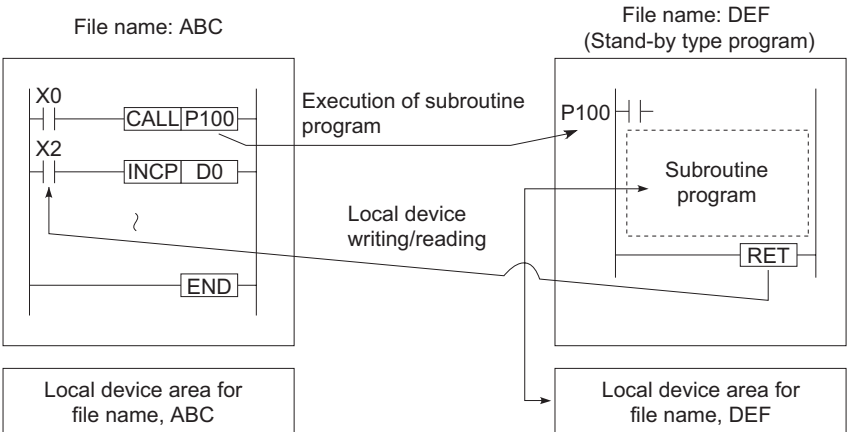


Figure 9.109 When SM776 is on

(c) Precautions

- When SM776 is on, local device data are read out when a subroutine program is called, and the data are saved after execution of the RET instruction.
Because of this, the scan time is increased if one subroutine program is executed with SM776 set to on.
- The on/off status of SM776 is set for each CPU module.
It cannot be set for each file.
- If the on/off status of SM776 is changed during sequence program execution, control is implemented according to the information after the change.

Remark

For details of SM776, refer to CHAPTER 12.

(6) When executing an interrupt/fixed scan execution type program

When executing an interrupt/fixed scan execution type program, you can utilize the local device corresponding to the file where the program is stored.

Use of the relevant local device is set by ON/OFF of SM777. *1

*1: The index register set as the local device uses the local device area for the program executed before the interrupt/fixed scan execution type program, regardless of the on/off status of SM777.

Table9.16 Local device switching by ON/OFF of the special relay (SM777)

SM777	Operation
OFF	Perform operations with the local device that corresponds to the program executed before the interrupt/fixed scan execution type program.
ON	Perform operations with the local device that corresponds to the program file where the interrupt/fixed scan execution type program is stored.

(a) When SM777 is off

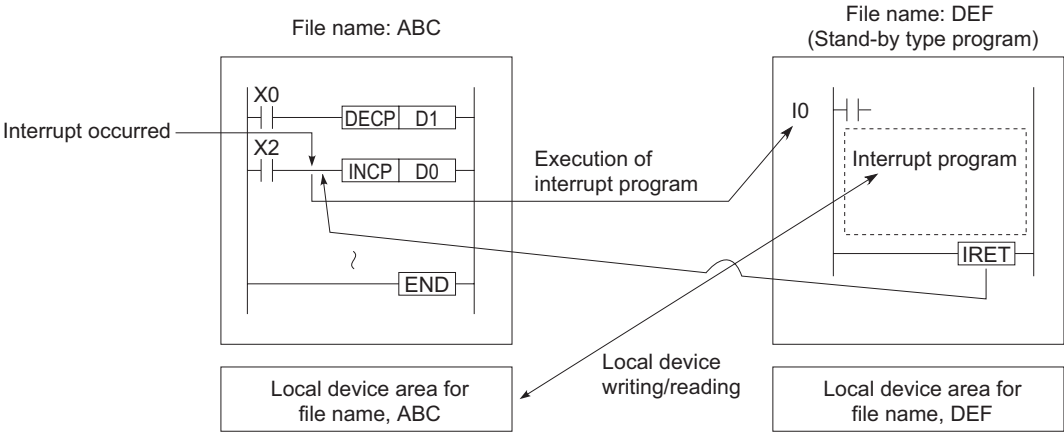


Figure 9.110 When SM777 is off

(b) When SM777 is on

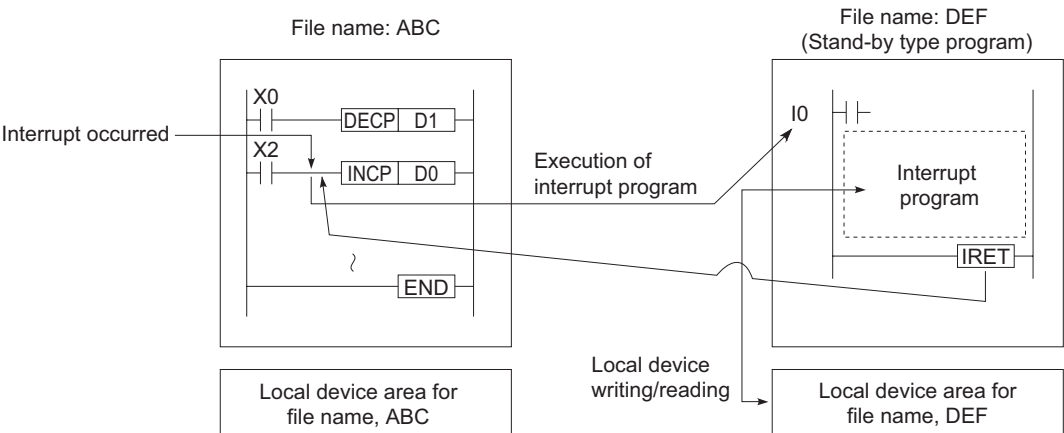


Figure 9.111 When SM777 is on

(c) Precautions

- When SM777 is on, local device data are read out before execution of an interrupt/fixed scan execution type program, and the data are saved after execution of the IRET instruction.
Because of this, the scan time is increased if one interrupt/fixed scan execution type program is executed with SM777 set to on.
- The on/off status of SM777 is set for each CPU module.
It cannot be set for each file.
- If the on/off status of SM777 is changed during sequence program execution, control is implemented according to the information after the change.

Remark

For details of SM777, refer to CHAPTER 12.

(7) Clearing local device data

Local device data is cleared by either of the following:

- When the CPU module is powered off and then on or is reset
- When the CPU module status is changed from STOP to RUN

Local device data cannot be cleared from GX Developer.

CHAPTER10 CPU MODULE PROCESSING TIME

This chapter describes the CPU module processing time.

10.1 Scan Time

This section describes the scan time structures and CPU module processing time.

10.1.1 Scan time structure

A CPU module sequentially performs the following processing in the RUN status.

Scan time is the time required for all processing and executions to be performed.

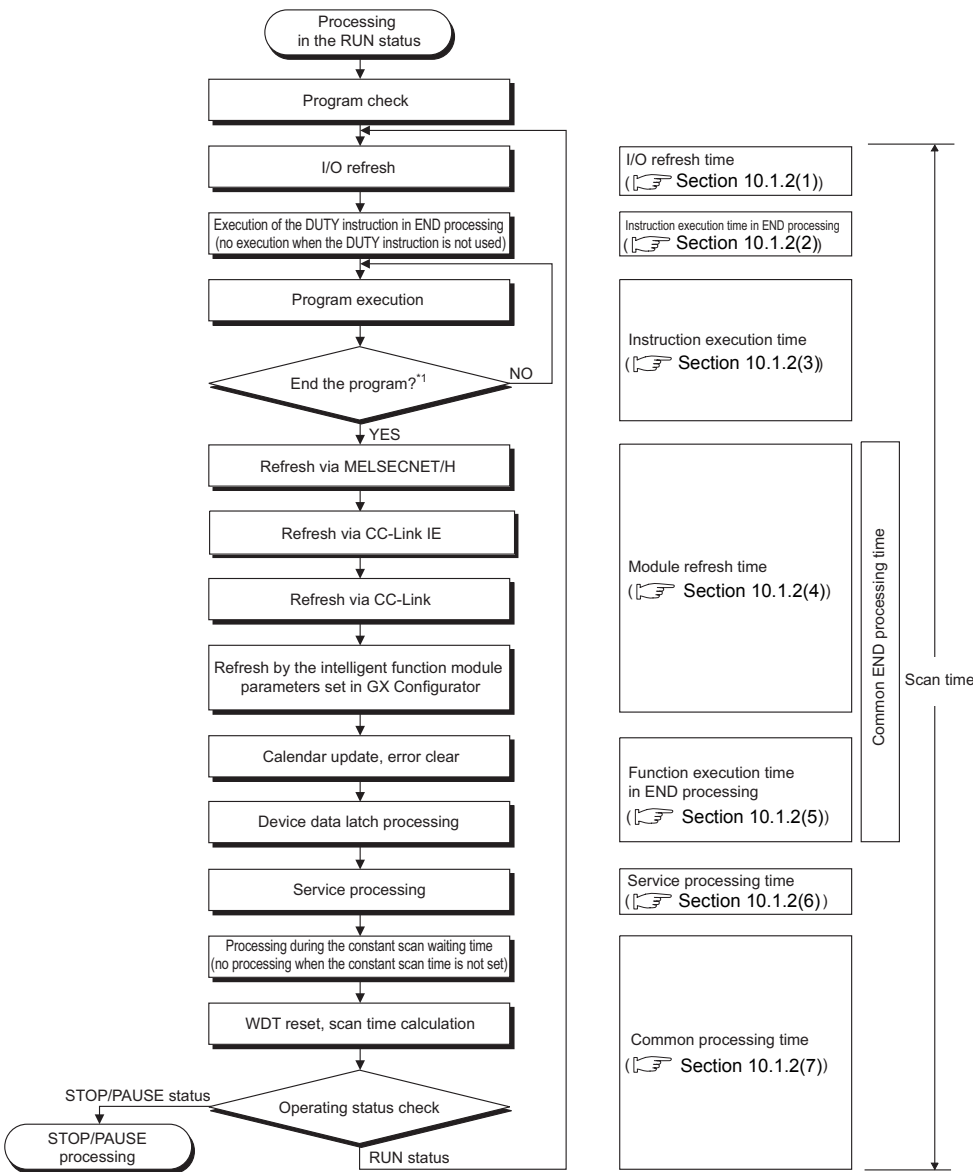


Figure 10.1 Scan time structure

*1: End of a program indicates the timing when the END, GOEND, FEND, or STOP instruction is executed.

(1) How to check scan time

The CPU module measures current, minimum, and maximum values of the scan time.

The scan time can be checked by monitoring the special register (SD520, SD521, and SD524 to SD527).

Accuracy of each stored scan time is $\pm 0.1\text{ms}$.

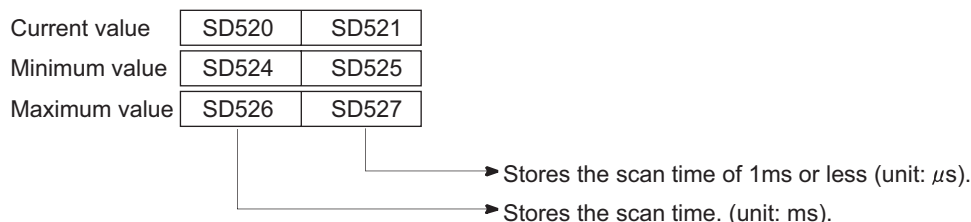


Figure 10.2 Scan time storage location

Example If the stored values in SD520 and SD521 are 3 and 400 respectively, the scan time is 3.4ms.

10.1.2 Time required for each processing included in scan time

This section describes how to calculate the processing and execution time described in Section 10.1.1.

(1) I/O refresh time

The I/O refresh time is time required for refreshing I/O data to/from the following modules mounted on the main base unit and extension base units.

- Input module
- Output module
- Intelligent function module

■ Calculation method

Use the following expression to calculate the I/O refresh time.

$$(\text{I/O refresh time}) = (\text{number of input points}/16) \times N1 + (\text{number of output points}/16) \times N2$$

For N1 and N2, refer to Table10.1.

Table10.1 I/O refresh time

CPU module	Q3□B, Q3□SB, Q3□RB, Q3□DB		Q5□B, Q6□B, Q6□RB	
	N1	N2	N1	N2
Q00UJCPU, Q00UCPU, Q01UCPU	1.8 μs	1.0 μs	2.6 μs	1.9 μs
Q02UCPU	1.5 μs	1.1 μs	2.4 μs	1.9 μs
Q03UDCPU, Q04UDHCPU, Q06UDHCPU, Q10UDHCPU, Q13UDHCPU, Q20UDHCPU, Q26UDHCPU, Q03UDECPU, Q04UDEHCPU, Q06UDEHCPU, Q10UDEHCPU, Q13UDEHCPU, Q20UDEHCPU, Q26UDEHCPU, Q50UDEHCPU, Q100UDEHCPU	1.5 μs	1.0 μs	2.3 μs	1.8 μs

(2) Instruction execution time in END processing

This is the processing time of the DUTY instruction in END processing.

The user timing clock (SM420 to 424 and SM430 to SM434) specified with the DUTY instruction is turned on/off during the END processing.

Table10.2 Instruction execution time in END processing

CPU module	Processing time in END processing	
	When set to 1	When set to 5
Q00UJCPU, Q00UCPU, Q01UCPU	0.0120ms	0.0140ms
Q02UCPU	0.0050ms	0.0550ms
Q03UDCPU, Q03UDECPU	0.0043ms	0.0046ms
Q04UDHCPU, Q06UDHCPU, Q10UDHCPU, Q13UDHCPU, Q20UDHCPU, Q26UDHCPU, Q04UDEHCPU, Q06UDEHCPU, Q10UDEHCPU, Q13UDEHCPU, Q20UDEHCPU, Q26UDEHCPU, Q50UDEHCPU, Q100UDEHCPU	0.0041ms	0.0045ms

(3) Instruction execution time

The instruction execution time is the time required for all instructions used in the program to be executed.

Remark

For the processing time required for each instruction, refer to the following.

 MELSEC-Q/L Programming Manual (Common Instruction)

(a) Overhead time at execution of interrupt and fixed scan execution type programs

When calculating instruction execution time, add the overhead time given in the following table to the instruction execution time, which is described in (3).

Two kinds of overhead time (pre-start and program-end) need to be added to interrupt programs.

Table10.3 Pre-start overhead time for interrupt programs

CPU module	Fixed scan interrupt (I28 to I31)		Multiple CPU synchronous interrupt (I45)		Interrupt*1 (I0 to I15) from QI60 or interrupt (I50 to I127) from the intelligent function module	
	Without high-speed start	With high-speed start	Without high-speed start	With high-speed start	Without high-speed start	With high-speed start
Q00UJCPU, Q00UCPU, Q01UCPU	55 μ s	35 μ s	---	---	76 μ s	55 μ s
Q02UCPU	48 μ s	17 μ s	---	---	60 μ s	31 μ s
Q03UDCPU, Q03UDECPU	47 μ s	17 μ s	46 μ s	16 μ s	54 μ s	22 μ s
Q04UDHCPU, Q06UDHCPU, Q10UDHCPU, Q13UDHCPU, Q20UDHCPU, Q26UDHCPU, Q04UDEHCPU, Q06UDEHCPU, Q10UDEHCPU, Q13UDEHCPU, Q20UDEHCPU, Q26UDEHCPU	46 μ s	16 μ s	44 μ s	14 μ s	52 μ s	22 μ s
Q50UDEHCPU, Q100UDEHCPU	73 μ s	16 μ s	71 μ s	14 μ s	79 μ s	22 μ s

*1: Indicates the value when the QI60 is mounted on the slot 0 of the main base unit.

Table10.4 Program-end overhead time for interrupt programs

CPU module	Without high-speed start	With high-speed start
Q00UJCPU, Q00UCPU, Q01UCPU	28 μ s	15 μ s
Q02UCPU	26 μ s	7 μ s
Q03UDCPU, Q03UDECPU	26 μ s	7 μ s
Q04UDHCPU, Q06UDHCPU, Q10UDHCPU, Q13UDHCPU, Q20UDHCPU, Q26UDHCPU, Q04UDEHCPU, Q06UDEHCPU, Q10UDEHCPU, Q13UDEHCPU, Q20UDEHCPU, Q26UDEHCPU	26 μ s	7 μ s
Q50UDEHCPU, Q100UDEHCPU	44 μ s	7 μ s

Table10.5 Overhead time for fixed scan execution type programs

CPU module	Without high-speed start	With high-speed start
Q00UJCPU, Q00UCPU, Q01UCPU	92 μ s	60 μ s
Q02UCPU	73 μ s	25 μ s
Q03UDCPU, Q03UDECPU	73 μ s	24 μ s
Q04UDHCPU, Q06UDHCPU, Q10UDHCPU, Q13UDHCPU, Q20UDHCPU, Q26UDHCPU, Q04UDEHCPU, Q06UDEHCPU, Q10UDEHCPU, Q13UDEHCPU, Q20UDEHCPU, Q26UDEHCPU	72 μ s	23 μ s
Q50UDEHCPU, Q100UDEHCPU	117 μ s	23 μ s

1) Overhead time when local devices in the interrupt program are enabled

When SM777 (Enable/disable local device in interrupt program) is turned on, the time given in Table10.6 and Table10.7 will be added to the overhead time given in Table10.6 and Table10.7.

Each n, N1, N2, and N3 in the table indicates the following:

- n: Number of local device points (unit: K words)
- N1: Number of devices that specified a local device
- N2: Number of word device points that specified a local device
- N3: Number of bit device points that specified a local device

Table10.6 When a local device file in the standard RAM is used

CPU module	Additional time to the pre-start overhead time for interrupt programs (Table10.3)	Additional time to the program-end overhead time for interrupt programs (Table10.4)
Q00UCPU, Q01UCPU	$(13.2 \times N1) + (0.22 \times (N2 + (N3 \div 16))) + 210 \mu s$	$(8 \times N1) + (0.23 \times (N2 + (N3 \div 16))) + 30 \mu s$
Q02UCPU	$(13.2 \times N1) + (0.22 \times (N2 + (N3 \div 16))) + 210 \mu s$	$(8 \times N1) + (0.22 \times (N2 + (N3 \div 16))) + 30 \mu s$
Q03UDCPU, Q03UDECPU	$(8 \times N1) + (0.22 \times (N2 + (N3 \div 16))) + 80 \mu s$	$(8 \times N1) + (0.22 \times (N2 + (N3 \div 16))) + 20 \mu s$
Q04UDHCPU, Q06UDHCPU, Q10UDHCPU, Q13UDHCPU, Q20UDHCPU, Q26UDHCPU, Q04UDEHCPU, Q06UDEHCPU, Q10UDEHCPU, Q13UDEHCPU, Q20UDEHCPU, Q26UDEHCPU, Q50UDEHCPU, Q100UDEHCPU	$(8 \times N1) + (0.10 \times (N2 + (N3 \div 16))) + 80 \mu s$	$(8 \times N1) + (0.10 \times (N2 + (N3 \div 16))) + 20 \mu s$

Table10.7 When a local device file in a SRAM card is used

CPU module	Additional time to the pre-start overhead time for interrupt programs (Table 10.3)	Additional time to the program-end overhead time for interrupt programs (Table 10.4)
Q02UCPU	$(16 \times N1) + (0.43 \times (N2 + (N3 \div 16))) + 260 \mu s$	$(16 \times N1) + (0.43 \times (N2 + (N3 \div 16))) + 60 \mu s$
Q03UDCPU, Q03UDECPU	$(12 \times N1) + (0.43 \times (N2 + (N3 \div 16))) + 100 \mu s$	$(12 \times N1) + (0.43 \times (N2 + (N3 \div 16))) + 20 \mu s$
Q04UDHCPU, Q06UDHCPU, Q10UDHCPU, Q13UDHCPU, Q20UDHCPU, Q26UDHCPU, Q04UDEHCPU, Q06UDEHCPU, Q10UDEHCPU, Q13UDEHCPU, Q20UDEHCPU, Q26UDEHCPU, Q50UDEHCPU, Q100UDEHCPU	$(12 \times N1) + (0.40 \times (N2 + (N3 \div 16))) + 100 \mu s$	$(12 \times N1) + (0.40 \times (N2 + (N3 \div 16))) + 20 \mu s$

(4) Module refresh time

Module refresh time is the total time required for the CPU module to refresh data with CC-Link IE, MELSECNET/H, and CC-Link.

(a) Refresh via CC-Link IE

This is the time required for refreshing data between link devices in a CC-Link IE module and devices in the CPU module.

(b) Refresh via MELSECNET/H

This is the time required for refreshing data between link devices in a MELSECNET/H module and devices in the CPU module.

(c) Auto refresh via CC-Link

This is the time required for refreshing data between a CC-Link system master/local module and CPU module.

Remark

For each refresh time, refer to the following.



Reference manual of each network module

(d) Auto refresh with an intelligent function module

This is the time required for refreshing data between the buffer memory of an intelligent function module and devices in the CPU module.

Use intelligent function module utility package (GX Configurator) for auto refresh settings.

■ Calculation method

Use the following expression to calculate the auto refresh time with an intelligent function module.

$$(\text{Refresh time}) = \text{KN1} + \text{KN2} \times (\text{number of refresh points})$$

For KN1 and KN2, use the values given in Table10.8 and Table10.9.

Table10.8 When an intelligent function module is mounted on the main base unit

CPU module	KN1	KN2
Q00UJCPU, Q00UCPU, Q01UCPU	96.3 μs	6.7 μs
Q02UCPU	23 μs	6 μs
Q03UDCPU, Q03UDECPU	6 μs	5 μs
Q04UDHCPU, Q06UDHCPU, Q10UDHCPU, Q13UDHCPU, Q20UDHCPU, Q26UDHCPU, Q04UDEHCPU, Q06UDEHCPU, Q10UDEHCPU, Q13UDEHCPU, Q20UDEHCPU, Q26UDEHCPU, Q50UDEHCPU, Q100UDEHCPU	4 μs	5 μs

Table10.9 When an intelligent function module is mounted on the extension base unit

CPU module	KN1	KN2
Q00UJCPU	79.7 μs	8.9 μs
Q00UCPU, Q01UCPU	79.7 μs	8.1 μs
Q02UCPU	45 μs	7.0 μs
Q03UDCPU, Q03UDECPU	7 μs	6.0 μs
Q04UDHCPU, Q06UDHCPU, Q10UDHCPU, Q13UDHCPU, Q20UDHCPU, Q26UDHCPU, Q04UDEHCPU, Q06UDEHCPU, Q10UDEHCPU, Q13UDEHCPU, Q20UDEHCPU, Q26UDEHCPU, Q50UDEHCPU, Q100UDEHCPU	5 μs	6 μs

Example When the number of auto refresh points for the analog-digital converter module (Q64AD) is 4 points (when the module is mounted on the Q26UDHCPU main base unit)

$$0.029 \text{ (ms)} = 0.005 + 0.006 \times 4$$

(5) Function execution time in END processing

This is the time required for updating calendar or clearing error in END processing.

(a) Calendar update processing time

When the clock data set request (SM210 changes from off to on) or the clock data read request (SM213 turns on) is issued, the processing time for changing or reading the clock data is required in END processing.

Table10.10 Calendar update processing time

CPU module	Processing time in END processing	
	When the clock data set request is issued	When the clock data read request is issued
Q00UJCPU, Q00UCPU, Q01UCPU	0.040ms	0.019ms
Q02UCPU	0.040ms	0.019ms
Q03UDCPU, Q04UDHCPU, Q06UDHCPU, Q10UDHCPU, Q13UDHCPU, Q20UDHCPU, Q26UDHCPU, Q03UDECPU, Q04UDEHCPU, Q06UDEHCPU, Q10UDEHCPU, Q13UDEHCPU, Q20UDEHCPU, Q26UDEHCPU, Q50UDEHCPU, Q100UDEHCPU	0.011ms	0.004ms

(b) Error clear processing time

Upon the rising edge (changes from off to on) of SM50 (Error reset), the processing time for clearing the continuation error stored in SD50 is required.

Table10.11 Error clear processing time

CPU module	Processing time in END processing	
	When the error is cleared (the one detected by the annunciator)	When the error is cleared
Q00UJCPU, Q00UCPU, Q01UCPU	0.185ms	0.185ms
Q02UCPU	0.180ms	0.175ms
Q03UDCPU, Q03UDECPU	0.068ms	0.062ms
Q04UDHCPU, Q06UDHCPU, Q10UDHCPU, Q13UDHCPU, Q20UDHCPU, Q26UDHCPU, Q04UDEHCPU, Q06UDEHCPU, Q10UDEHCPU, Q13UDEHCPU, Q20UDEHCPU, Q26UDEHCPU, Q50UDEHCPU, Q100UDEHCPU	0.065ms	0.062ms

(6) Device data latch processing time

When the latch range is set in the Device tab of the PLC parameter dialog box^{*1 *2 *3}, the processing time shown in Table10.12 is required.

Each N1, N2, and N3 in the table indicates the following:

- N1: Number of devices specified to be latched (Count the latch range (1) and the latch range (2) as different devices.)
- N2: Number of bit device points specified to be latched
- N3: Number of word device points specified to be latched

Table10.12 Processing time when the latch range is set

CPU module	Processing time
Q00UJCPU, Q00UCPU, Q01UCPU	$(4.4 \times N1) + (0.12 \times (N2 \div 16 + N3)) \mu s$
Q02UCPU	$(4.0 \times N1) + (0.12 \times (N2 \div 16 + N3)) \mu s$
Q03UDCPU, Q03UDECPU	$(3.0 \times N1) + (0.12 \times (N2 \div 16 + N3)) \mu s$
Q04UDHCPU, Q06UDHCPU, Q10UDHCPU, Q13UDHCPU, Q20UDHCPU, Q26UDHCPU, Q04UDEHCPU, Q06UDEHCPU, Q10UDEHCPU, Q13UDEHCPU, Q20UDEHCPU, Q26UDEHCPU, Q50UDEHCPU, Q100UDEHCPU	$(3.0 \times N1) + (0.05 \times (N2 \div 16 + N3)) \mu s$

*1: When setting the latch range of the timer (T), retentive timer (ST), and counter (C), one point for word device and two points for bit device are occupied per point.

*2: The case where the points are set for the latch relay (L) is included.

*3: The scan time will not increase if the latch range is set for the file register (R, ZR), extended data register (D), or extended link register (W).

Point

To shorten the scan time, minimize the number of latch points.

The number of latch points can be reduced by performing the following.

- Move data to be latched to the file register.
- Store device data that is updated less frequently to the standard ROM with the SP.DEVST instruction. (The device data stored in the standard ROM can be read with the S(P).DEVLD instruction.)



MELSEC-Q/L Programming Manual (Common Instruction)

(7) Service processing time

Service processing is the communication processing with GX Developer and external devices. When monitoring device data, reading programs, and setting monitor conditions in GX Developer, the processing time shown in Table10.13 or Table10.14 is required.

Table10.13 Processing time to monitor device data and read programs

CPU module	Processing time*1	
	Monitoring device data (Data register: 32 points)	Reading programs (10K step)
Q00UJCPU, Q00UCPU, Q01UCPU	1.60ms	3.70ms
Q02UCPU	1.00ms	1.55ms
Q03UDCPU, Q04UDHCPU, Q06UDHCPU, Q10UDHCPU, Q13UDHCPU, Q20UDHCPU, Q26UDHCPU	0.35ms	0.95ms
Q03UDECPU, Q04UDEHCPU, Q06UDEHCPU, Q10UDEHCPU, Q13UDEHCPU, Q20UDEHCPU, Q26UDEHCPU, Q50UDEHCPU, Q100UDEHCPU	0.90ms	1.10ms

*1: The time in the table is for the case where the service processing count is set to one.

Table10.14 Processing time to set monitor conditions

CPU module	Processing time	
	Specified step match	Specified device match
Q02UCPU	0.03ms	0.04ms
Q03UDCPU, Q04UDHCPU, Q06UDHCPU, Q10UDHCPU, Q13UDHCPU, Q20UDHCPU, Q26UDHCPU, Q03UDECPU, Q04UDEHCPU, Q06UDEHCPU, Q10UDEHCPU, Q13UDEHCPU, Q20UDEHCPU, Q26UDEHCPU	0.01ms	0.03ms

(8) Common processing time

The CPU module performs common processing by the system. The common processing time shown in Table10.15 is required.

Table10.15 Common processing time

CPU module	Processing time
Q00UJCPU, Q00UCPU, Q01UCPU	0.25ms
Q02UCPU	0.17ms
Q03UDCPU	0.13ms
Q04UDHCPU, Q06UDHCPU, Q10UDHCPU, Q13UDHCPU, Q20UDHCPU, Q26UDHCPU	0.10ms
Q03UDECPU	0.22ms
Q04UDEHCPU, Q06UDEHCPU, Q10UDEHCPU, Q13UDEHCPU, Q20UDEHCPU, Q26UDEHCPU, Q50UDEHCPU, Q100UDEHCPU	0.18ms

(9) Multiple CPU high speed transmission processing time

This is the processing time required for data transmission between the CPU modules when the multiple CPU high speed transmission function is used.

For the multiple CPU high speed transmission processing time, refer to the following.

 QCPU User's Manual (Multiple CPU System)

10.1.3 Factors that increase the scan time

When executing any of the functions or operations described in this section, add the given processing time to the time value calculated in Section 10.1.2.

(1) Sampling trace

When the sampling trace function ( Section 6.14) is executed, the processing time shown in Table10.16 is required.

Table10.16 Processing time (when 50 points of the internal relay (for bit device) and 50 points of the data register (for word device) are set as sampling trace data)

	CPU module	Processing time
Standard RAM	Q00UCPU, Q01UCPU	0.12ms
	Q02UCPU	0.09ms
	Q03UDCPU, Q03UDECPU	0.07ms ^{*1}
	Q04UDHCPU, Q06UDHCPU, Q10UDHCPU, Q13UDHCPU, Q20UDHCPU, Q26UDHCPU, Q04UDEHCPU, Q06UDEHCPU, Q10UDEHCPU, Q13UDEHCPU, Q20UDEHCPU, Q26UDEHCPU	0.06ms ^{*1}
	Q50UDEHCPU, Q100UDEHCPU	0.06ms
SRAM card	Q02UCPU	0.16ms
	Q03UDCPU, Q03UDECPU	0.08ms ^{*1}
	Q04UDHCPU, Q06UDHCPU, Q10UDHCPU, Q13UDHCPU, Q20UDHCPU, Q26UDHCPU, Q04UDEHCPU, Q06UDEHCPU, Q10UDEHCPU, Q13UDEHCPU, Q20UDEHCPU, Q26UDEHCPU	0.06ms ^{*1}
	Q50UDEHCPU, Q100UDEHCPU	0.06ms

*1: When specifying devices in the trigger point setting, the scan time may be increased by up to 0.163 μ s per instruction.

(2) Use of local devices

When local devices are used, the processing time shown in Table10.17 is required.

Each n, N1, N2, N3, and N4 in the table indicates the following:

- n: Number of programs using a local device^{*1}
- N1: Number of devices that specified a local device
- N2: Number of word device points that specified a local device (Except index register)
- N3: Number of bit device points that specified a local device
- N4: Number of index register points that were specified as a local device

Table10.17 Processing time (when local devices are used)

	CPU module	Processing time
Standard RAM	Q00UCPU, Q01UCPU	$((16.0 \times N1) + (0.23 \times (N2 + (N3 \div 16)))) + (1.49 \times N4) + 98.3) \times n + 92.0 \mu s$
	Q02UCPU	$((24.0 \times N1) + (0.23 \times (N2 + (N3 \div 16)))) + (1.57 \times N4) + 108) \times n + 59.0 \mu s$
	Q03UDCPU, Q03UDECPU	$((8.0 \times N1) + (0.22 \times (N2 + (N3 \div 16)))) + (0.65 \times N4) + 38.0) \times n + 14.2 \mu s$
	Q04UDHCPU, Q06UDHCPU, Q04UDEHCPU, Q06UDEHCPU	$((8.0 \times N1) + (0.10 \times (N2 + (N3 \div 16)))) + (0.47 \times N4) + 35.5) \times n + 12.7 \mu s$
	Q10UDHCPU, Q13UDHCPU, Q20UDHCPU, Q26UDHCPU, Q10UDEHCPU, Q13UDEHCPU, Q20UDEHCPU, Q26UDEHCPU, Q50UDEHCPU, Q100UDEHCPU	$((8.0 \times N1) + (0.10 \times (N2 + (N3 \div 16)))) + (0.68 \times N4) + 35.5) \times n + 17.3 \mu s$
SRAM card	Q02UCPU	$((24.0 \times N1) + (0.43 \times (N2 + (N3 \div 16)))) + (1.40 \times N4) + 66.0) \times n + 83.0 \mu s$
	Q03UDCPU, Q03UDECPU	$((12.0 \times N1) + (0.43 \times (N2 + (N3 \div 16)))) + (0.68 \times N4) + 41.0) \times n + 17.0 \mu s$
	Q04UDHCPU, Q06UDHCPU, Q04UDEHCPU, Q06UDEHCPU	$((12.0 \times N1) + (0.40 \times (N2 + (N3 \div 16)))) + (0.59 \times N4) + 38.5) \times n + 17.0 \mu s$
	Q10UDHCPU, Q13UDHCPU, Q20UDHCPU, Q26UDHCPU, Q10UDEHCPU, Q13UDEHCPU, Q20UDEHCPU, Q26UDEHCPU, Q50UDEHCPU, Q100UDEHCPU	$((12.0 \times N1) + (0.40 \times (N2 + (N3 \div 16)))) + (0.79 \times N4) + 44.7) \times n + 12.1 \mu s$

*1: If the serial number (first five digits) of the Universal model QCPU is "10011" or earlier, "n" indicates the number of executed programs.

(a) When local devices in a subroutine program are enabled

When SM776 (Enable/disable local device at CALL) is turned on, the processing time shown in Table10.18 or Table10.19 is required for each subroutine call.

Each n, N1, N2, N3, and N4 in the table indicates the following:

- n: Number of local device points (unit: K words)
- N1: Number of devices that specified a local device
- N2: Number of word device points that specified a local device (Except index register)
- N3: Number of bit device points that specified a local device
- N4: Number of index register points that were specified as a local device

Table10.18 Processing time (when a local device file in the standard RAM is used)

CPU module	Processing time when a subroutine program in the same file is called	Processing time when a subroutine program in a different file is called
Q00UCPU, Q01UCPU	0.00 μ s	$(20.3 \times N1) + (0.760 \times (N2 + (N3 \div 16))) + (4.47 \times N4) + 257.0 \mu$ s
Q02UCPU	0.00 μ s	$(20.3 \times N1) + (0.760 \times (N2 + (N3 \div 16))) + (4.71 \times N4) + 257.0 \mu$ s
Q03UDCPU, Q03UDECPU	0.00 μ s	$(16 \times N1) + (0.44 \times (N2 + (N3 \div 16))) + (1.30 \times N4) + 100 \mu$ s
Q04UDHCPU, Q06UDHCPU, Q04UDEHCPU, Q06UDEHCPU	0.00 μ s	$(16 \times N1) + (0.20 \times (N2 + (N3 \div 16))) + (0.94 \times N4) + 100 \mu$ s
Q10UDHCPU, Q13UDHCPU, Q20UDHCPU, Q26UDHCPU, Q10UDEHCPU, Q13UDEHCPU, Q20UDEHCPU, Q26UDEHCPU, Q50UDEHCPU, Q100UDEHCPU	0.00 μ s	$(16 \times N1) + (0.20 \times (N2 + (N3 \div 16))) + (1.36 \times N4) + 100 \mu$ s

Table10.19 Processing time (when a local device file in a SRAM card is used)

CPU module	Processing time when a subroutine program in the same file is called	Processing time when a subroutine program in a different file is called
Q02UCPU	0.00 μ s	$(20.3 \times N1) + (0.760 \times (N2 + (N3 \div 16))) + (2.80 \times N4) + 257.0 \mu$ s
Q03UDCPU, Q03UDECPU	0.00 μ s	$(24 \times N1) + (0.86 \times (N2 + (N3 \div 16))) + (1.36 \times N4) + 120 \mu$ s
Q04UDHCPU, Q06UDHCPU, Q04UDEHCPU, Q06UDEHCPU	0.00 μ s	$(24 \times N1) + (0.80 \times (N2 + (N3 \div 16))) + (1.18 \times N4) + 100 \mu$ s
Q10UDHCPU, Q13UDHCPU, Q20UDHCPU, Q26UDHCPU, Q10UDEHCPU, Q13UDEHCPU, Q20UDEHCPU, Q26UDEHCPU, Q50UDEHCPU, Q100UDEHCPU	0.00 μ s	$(24 \times N1) + (0.80 \times (N2 + (N3 \div 16))) + (1.58 \times N4) + 120 \mu$ s

(3) Execution of multiple programs

When multiple programs are executed, the processing time shown in Table10.20 is required for each program.

Table10.20 Processing time for each program (when multiple programs are executed)

CPU module	Processing time
Q00UJCPU, Q00UCPU, Q01UCPU	$0.053 \times n^{*1}$ ms
Q02UCPU	$0.04 \times n^{*1}$ ms
Q03UDCPU, Q03UDECPU	$0.02 \times n^{*1}$ ms
Q04UDHCPU, Q06UDHCPU, Q10UDHCPU, Q13UDHCPU, Q20UDHCPU, Q26UDHCPU, Q04UDEHCPU, Q06UDEHCPU, Q10UDEHCPU, Q13UDEHCPU, Q20UDEHCPU, Q26UDEHCPU, Q50UDEHCPU, Q100UDEHCPU	$0.02 \times n^{*1}$ ms

*1: "n" indicates the number of program files.

(4) Removal and insertion of a memory card

When a memory card is removed or inserted, the processing time shown in Table10.21 is required only for one scan where a memory card is removed or inserted.

Table10.21 Processing time (when a memory card is removed or inserted)

CPU module	Processing time	
	When a memory card is inserted	When a memory card is removed
Q02UCPU	0.7ms	0.2ms
Q03UDCPU, Q04UDHCPU, Q06UDHCPU, Q10UDHCPU, Q13UDHCPU, Q20UDHCPU, Q26UDHCPU, Q03UDECPU, Q04UDEHCPU, Q06UDEHCPU, Q10UDEHCPU, Q13UDEHCPU, Q20UDEHCPU, Q26UDEHCPU, Q50UDEHCPU, Q100UDEHCPU	0.6ms	0.1ms

(5) Use of the file register

When "Use the same file name as the program." is selected in the PLC file tab of the PLC parameter dialog box, the processing time shown in Table10.22 is required.

When "Use the following file." is selected, the scan time will not be increased.

Table10.22 Processing time (when the file register is used)

	CPU module	Processing time
Standard RAM	Q00UCPU, Q01UCPU	0.135ms
	Q02UCPU	0.082ms
	Q03UDCPU, Q03UDECPU	0.043ms
	Q04UDHCPU, Q06UDHCPU, Q10UDHCPU, Q13UDHCPU, Q20UDHCPU, Q26UDHCPU, Q04UDEHCPU, Q06UDEHCPU, Q10UDEHCPU, Q13UDEHCPU, Q20UDEHCPU, Q26UDEHCPU, Q50UDEHCPU, Q100UDEHCPU	0.041ms
SRAM card	Q02UCPU	$0.11 \times n^{*1}$ ms
	Q03UDCPU, Q03UDECPU	$0.06 \times n^{*1}$ ms
	Q04UDHCPU, Q06UDHCPU, Q10UDHCPU, Q13UDHCPU, Q20UDHCPU, Q26UDHCPU, Q04UDEHCPU, Q06UDEHCPU, Q10UDEHCPU, Q13UDEHCPU, Q20UDEHCPU, Q26UDEHCPU, Q50UDEHCPU, Q100UDEHCPU	$0.06 \times n^{*1}$ ms

*1: "n" indicates the number of program files.

(6) Online change

When data is written to the running CPU module, the processing time described below is required.

(a) Online change (ladder mode)

When a program in the running CPU module is changed in ladder mode, the processing time shown in Table10.23 is required.*¹

*1: The time in the table is for the case where the service processing count is set to one.

Table10.23 Processing time (online change (ladder mode))

CPU module	Processing time	
	The reserved area for online change is not changed.	The reserved area for online change is re-set.
Q00UJCPU, Q00UCPU, Q01UCPU	Up to 2.1ms	Up to 2.1ms
Q02UCPU	Up to 1.3ms	Up to 1.3ms
Q03UDCPU, Q03UDECPU	Up to 1.0ms	Up to 1.0ms
Q04UDHCPU, Q06UDHCPU, Q10UDHCPU, Q13UDHCPU, Q20UDHCPU, Q26UDHCPU, Q04UDEHCPU, Q06UDEHCPU, Q10UDEHCPU, Q13UDEHCPU, Q20UDEHCPU, Q26UDEHCPU, Q50UDEHCPU, Q100UDEHCPU	Up to 0.7ms	Up to 0.7ms

(b) Online change (files)

When a file is written to the running CPU module, the processing time shown in Table10.24 is required.*¹

*1: The time in the table is for the case where the service processing count is set to one.

Table10.24 Processing time (online change (files))

CPU module	Processing time	
	Scan time = 2ms	Scan time = 20ms
Q00UJCPU	Up to 4.00ms	Up to 6.20ms
Q00UCPU	Up to 3.50ms	Up to 5.80ms
Q01UCPU	Up to 3.50ms	Up to 5.60ms
Q02UCPU	Up to 4.80ms	Up to 4.80ms
Q03UDCPU, Q03UDECPU	Up to 3.75ms	Up to 3.75ms
Q04UDHCPU, Q06UDHCPU, Q10UDHCPU, Q13UDHCPU, Q20UDHCPU, Q26UDHCPU, Q04UDEHCPU, Q06UDEHCPU, Q10UDEHCPU, Q13UDEHCPU, Q20UDEHCPU, Q26UDEHCPU, Q50UDEHCPU, Q100UDEHCPU	Up to 3.70ms	Up to 3.70ms

(7) Non-group output status read

In multiple CPU systems, the scan time increases when “All CPUs can read all outputs” is selected in the Multiple CPU settings screen of the PLC parameter dialog box.

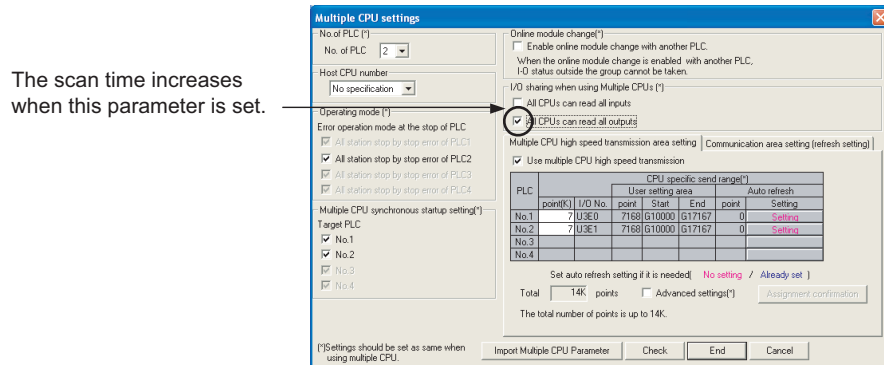


Figure 10.3 Multiple CPU settings screen

(8) Scan time measurement

When the scan time is measured by GX Developer, the processing time shown in Table10.25 is required (Section 6.13.3)

Table10.25 Processing time (when the scan time is measured)

CPU module	Processing time
Q00UJCPU, Q00UCPU, Q01UCPU	$179.5 + 5.8 \times \text{number of branch instructions } \mu\text{s}^{*1}$
Q02UCPU	$40.0 + 3.0 \times \text{number of branch instructions } \mu\text{s}^{*1}$
Q03UDCPU, Q04UDHCPU, Q06UDHCPU, Q10UDHCPU, Q13UDHCPU, Q20UDHCPU, Q26UDHCPU, Q03UDECPU, Q04UDEHCPU, Q06UDEHCPU, Q10UDEHCPU, Q13UDEHCPU, Q20UDEHCPU, Q26UDEHCPU, Q50UDEHCPU, Q100UDEHCPU	$40.0 + 1.5 \times \text{number of branch instructions } \mu\text{s}^{*1}$

*1: The number of the branch instructions is a total of the following instructions, which are executed during the scan time measurement.

- Pointer branch instruction : CJ, SCJ, JMP
- Subroutine program call instruction: CALL(P), FCALL(P), ECALL(P), EFCALL(P), XCALL(P), RET

(9) Monitor condition setting

When setting monitor conditions on GX Developer (Section 6.11.1), the processing time shown in Table10.26 occurs.

Table10.26 Processing time when monitor condition setting is executed

CPU module	Processing time	
	Specified step matches the execution condition	Specified device matches the execution condition (Device D matches the condition)
Q02UCPU	30 μs	40 μs
Q03UDCPU, Q03UDECPU, Q04UDHCPU, Q06UDHCPU, Q10UDHCPU, Q13UDHCPU, Q20UDHCPU, Q26UDHCPU, Q04UDEHCPU, Q06UDEHCPU, Q10UDEHCPU, Q13UDEHCPU, Q20UDEHCPU, Q26UDEHCPU	10 μs	30 μs

(10)Time taken to collect module errors

When using the module error collection, the scan time increases by the time found by the following calculation formula.

■ Calculation formula

$$\text{Collection time} = N1 + N2 \times (\text{Number of module errors collected in one scan})$$

The following table shows N1 and N2 values.

Table10.27 N1 values

CPU module	Mounted base unit	
	Main base unit	Extension base unit
Q00UJCPU	175 μ s	190 μ s
Q00UCPU, Q01UCPU	145 μ s	190 μ s
Q02UCPU	145 μ s	185 μ s
Q03UDCPU, Q04UDHCPU, Q06UDHCPU, Q10UDHCPU, Q13UDHCPU, Q20UDHCPU, Q26UDHCPU, Q03UDECPU, Q04UDEHCPU, Q06UDEHCPU, Q10UDEHCPU, Q13UDEHCPU, Q20UDEHCPU, Q26UDEHCPU, Q50UDEHCPU, Q100UDEHCPU	15 μ s	15 μ s

Table10.28 N2 values

CPU module	Mounted base unit	
	Main base unit	Extension base unit
Q00UJCPU, Q00UCPU, Q01UCPU	120 μ s	140 μ s
Q02UCPU	90 μ s	105 μ s
Q03UDCPU, Q04UDHCPU, Q06UDHCPU, Q10UDHCPU, Q13UDHCPU, Q20UDHCPU, Q26UDHCPU, Q03UDECPU, Q04UDEHCPU, Q06UDEHCPU, Q10UDEHCPU, Q13UDEHCPU, Q20UDEHCPU, Q26UDEHCPU, Q50UDEHCPU, Q100UDEHCPU	70 μ s	100 μ s

(11) Batch transfer of data to the program memory

When data in the program cache memory is batch-transferred to the program memory by GX Developer, the processing time shown in Table10.29 is required.*1

*1: The time in the table is for the case where the service processing count is set to one.

Table10.29 Processing time (when data is batch-transferred to the program memory)

CPU module	Processing time	
	Scan time = 2ms	Scan time = 20ms
Q00UJCPU, Q00UCPU, Q01UCPU	1.90ms	4.90ms
Q02UCPU	1.55ms	4.50ms
Q03UDCPU	1.10ms	3.65ms
Q04UDHCPU, Q06UDHCPU, Q10UDHCPU, Q13UDHCPU, Q20UDHCPU, Q26UDHCPU	1.05ms	3.65ms
Q04UDEHCPU, Q06UDEHCPU, Q10UDEHCPU, Q13UDEHCPU, Q20UDEHCPU, Q26UDEHCPU, Q50UDEHCPU, Q100UDEHCPU	1.40ms	4.00ms

(12) Diagnostics of the redundant power supply system

When the "Diagnose redundant power supply system." is selected on the PLC RAS tab of the PLC parameter dialog box, the processing time shown in Table10.30 is required.

Table10.30 Processing time (when the diagnostics for the redundant power supply system is enabled)

CPU module	Processing time	
	With a power supply module failure*1	Without a power supply module failure*1
Q00UCPU	125 μ s	165 μ s
Q01UCPU	125 μ s	135 μ s
Q02UCPU	90 μ s	90 μ s
Q03UDCPU, Q04UDHCPU, Q06UDHCPU, Q10UDHCPU, Q13UDHCPU, Q20UDHCPU, Q26UDHCPU, Q03UDECPU, Q04UDEHCPU, Q06UDEHCPU, Q10UDEHCPU, Q13UDEHCPU, Q20UDEHCPU, Q26UDEHCPU, Q50UDEHCPU, Q100UDEHCPU	43 μ s	52 μ s

*1: A power supply module failure indicates any of the following.

- The redundant power supply module has failed.
- Power for the redundant power supply module is turned off.
- The redundant power supply module is not mounted.

(13)A -PLC Compatibility Setting

When "A-PLC Compatibility Setting" is enabled in the PLC System tab of the PLC Parameter dialog box, the processing time shown in Table10.31 occurs.

Table10.31 Processing time of A-PLC compatibility setting

CPU module	Processing time
Q00UJCPU, Q00UCPU, Q01UCPU	95 μ s
Q02UCPU	90 μ s
Q03UDCPU, Q03UDECPU, Q04UDHCPU, Q06UDHCPU, Q10UDHCPU, Q13UDHCPU, Q20UDHCPU, Q26UDHCPU, Q04UDEHCPU, Q06UDEHCPU, Q10UDEHCPU, Q13UDEHCPU, Q20UDEHCPU, Q26UDEHCPU, Q50UDEHCPU, Q100UDEHCPU	34 μ s

CHAPTER11 PROCEDURES FOR WRITING PROGRAM TO CPU MODULE

This chapter describes procedures for writing a program created by GX Developer to the CPU module.

Remark

For procedures for starting the CPU module, refer to the following.

☞ QCPU User's Manual (Hardware Design, Maintenance and Inspection)

11.1 Items to be Considered for Creating Programs

To create a program, the number of device points, size, and file name of the program must be predetermined.

(1) Program size

Check whether the total size of programs and parameters are within the program size executable in the CPU module used. (☞ Section 5.3.3)

Table11.1 provides the program size executable in each CPU module.

Table11.1 Program size

CPU module	Program size
Q00UJCPU, Q00UCPU	10K steps (40K bytes)
Q01UCPU	15K steps (60K bytes)
Q02UCPU	20K steps (80K bytes)
Q03UDCPU, Q03UDECPU	30K steps (120K bytes)
Q04UDHCPU, Q04UDEHCPU	40K steps (160K bytes)
Q06UDHCPU, Q06UDEHCPU	60K steps (240K bytes)
Q10UDHCPU, Q10UDEHCPU	100K steps (400K bytes)
Q13UDHCPU, Q13UDEHCPU	130K steps (520K bytes)
Q20UDHCPU, Q20UDEHCPU	200K steps (800K bytes)
Q26UDHCPU, Q26UDEHCPU	260K steps (1040K bytes)
Q50UDEHCPU	500K steps (2000K bytes)
Q100UDEHCPU	1000K steps (4000K bytes)

Whether parameters are stored to the program memory, standard ROM, or memory card can be set.

If the program size above is required for only programs, store the parameters in the standard ROM or memory card.

(2) Determining a unit for structuring the programs

When creating multiple programs, determine a unit (process/function) for structuring the programs.

(3) Setting the execution conditions for programs to be created

When executing multiple programs, set their execution conditions to each program. (☞ Section 2.3)
Without the setting, the programs cannot be executed.

(4) Setting the applications of devices and the number of device points

Consider the applications of devices and the number of device points used in the program.
(☞ CHAPTER 9)

(5) Setting the initial device value

Set data necessary as an initial value to the device memory and the buffer memory of the intelligent function module. (☞ Section 6.25)

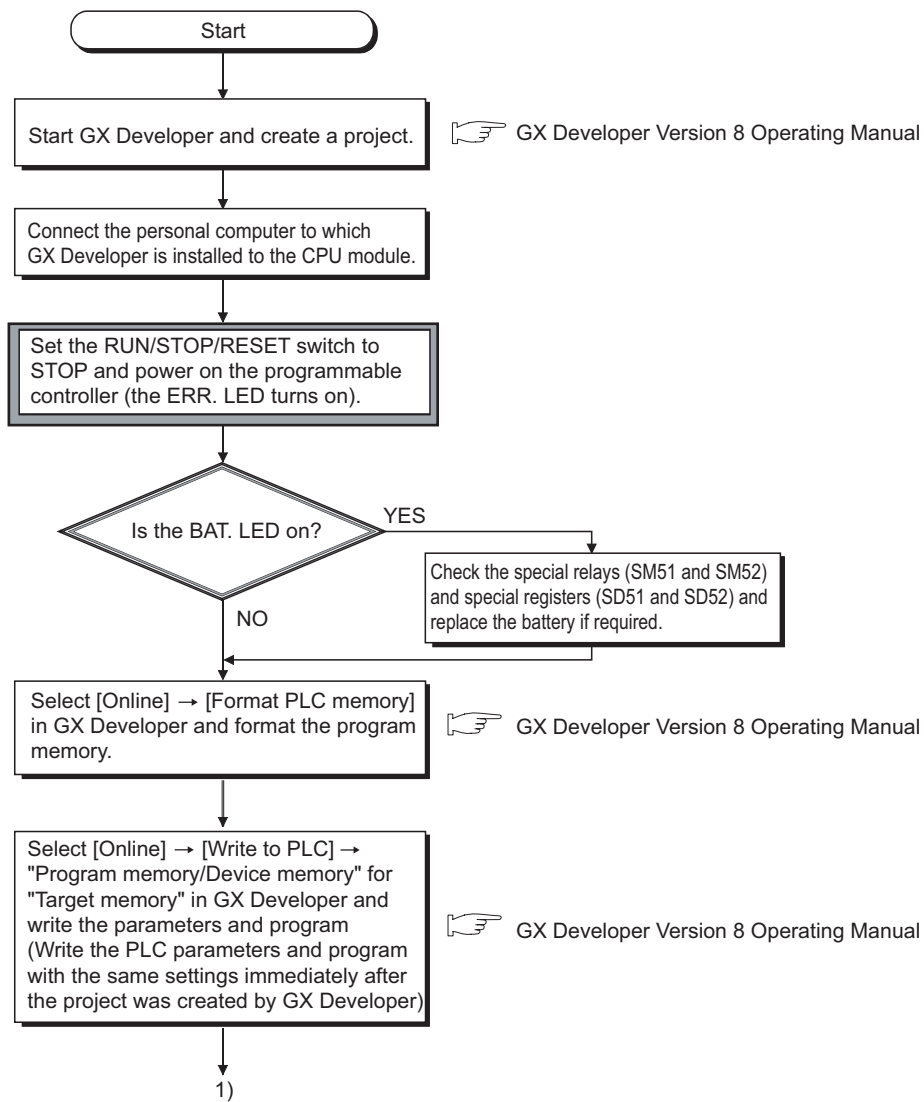
(6) Setting boot operation

When storing a program to the memory card, execute the program after boot operation.
For boot operation, make the setting in the Boot file tab of the PLC parameter dialog box.
(☞ Section 5.1.8, Section 5.1.10)

11.2 Hardware Check

This section describes a procedure for checking hardware before writing a created program.

In the following procedure,  indicates an operation on the CPU module side.



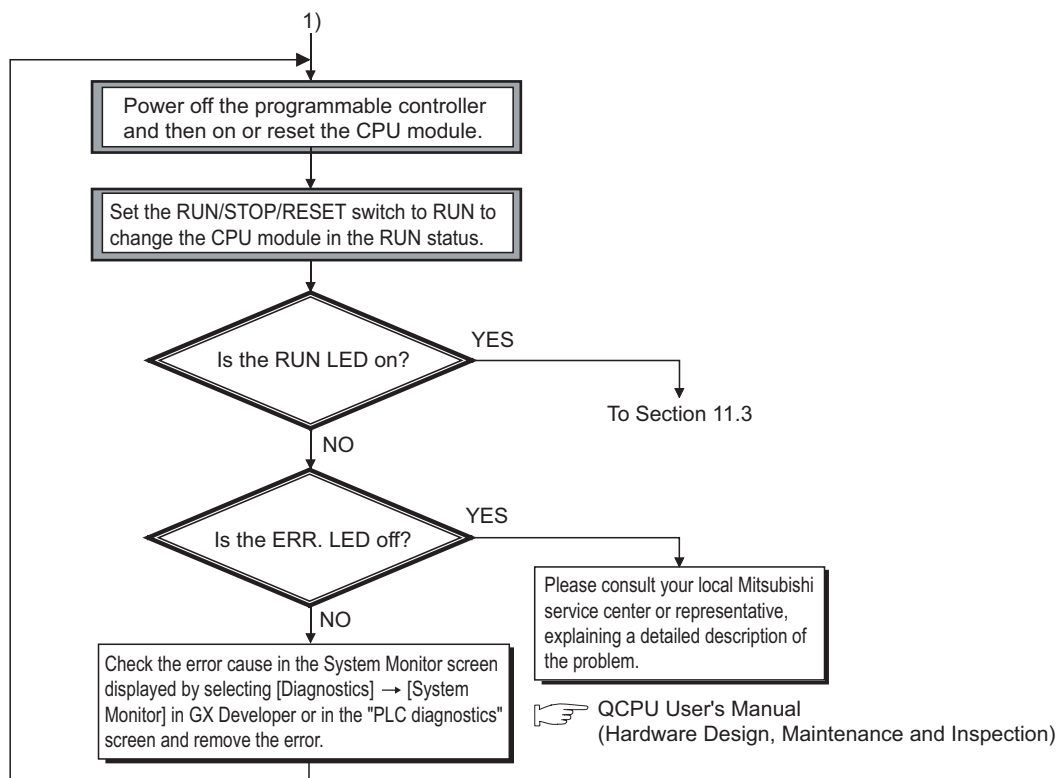


Figure 11.1 Hardware check flowchart

Remark

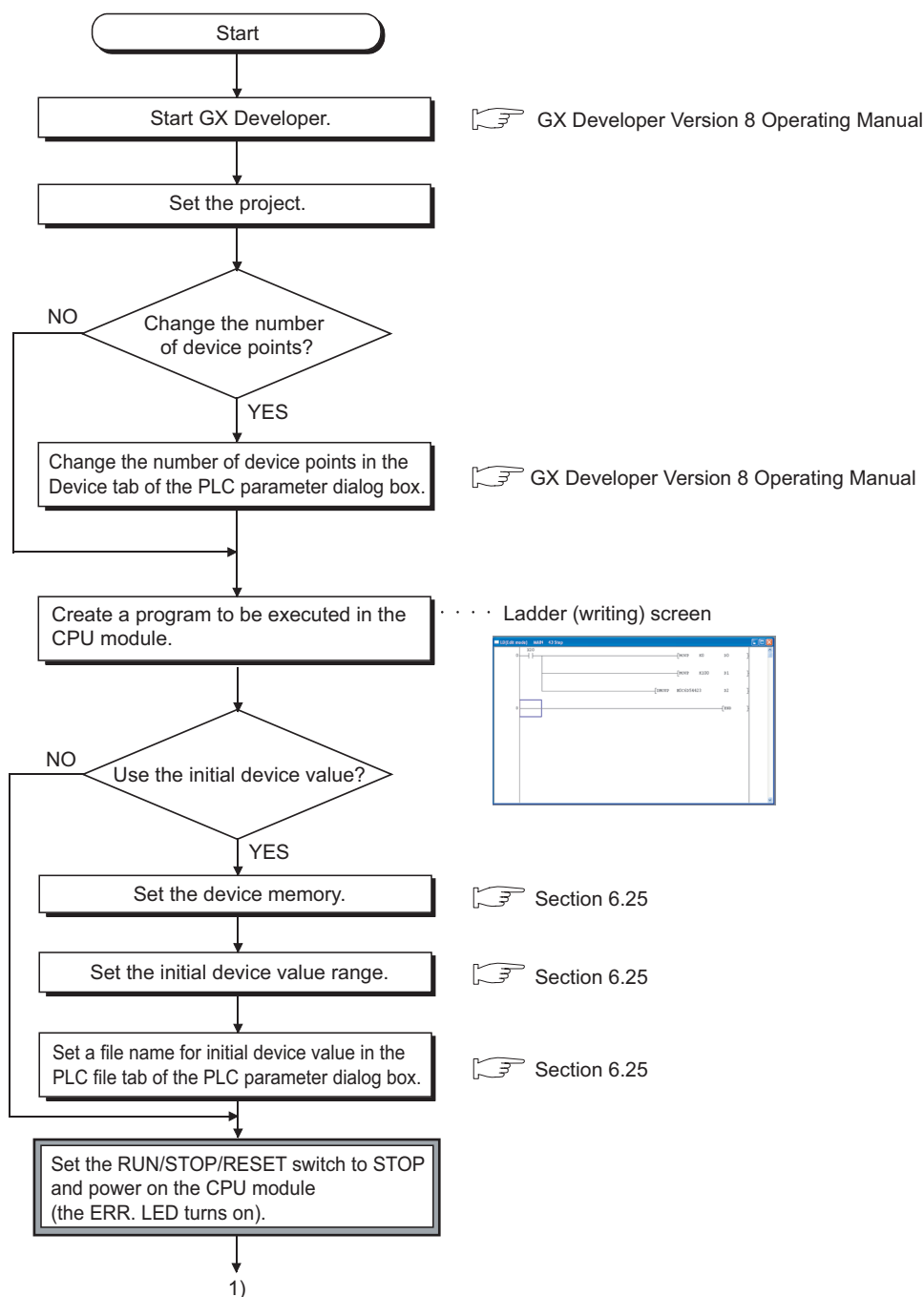
For installation and mounting procedures of the CPU module, refer to the following.

☞ QCPU User's Manual (Hardware Design, Maintenance and Inspection)

11.3 Procedure for Writing One Program

This section describes a procedure for writing a program to the program memory. (☞ Section 5.1.2)
Follow the procedure below and then the procedure provided in Section 11.5 before storing the program in the memory card for boot operation.

In the following procedure,  indicates an operation on the CPU module side.



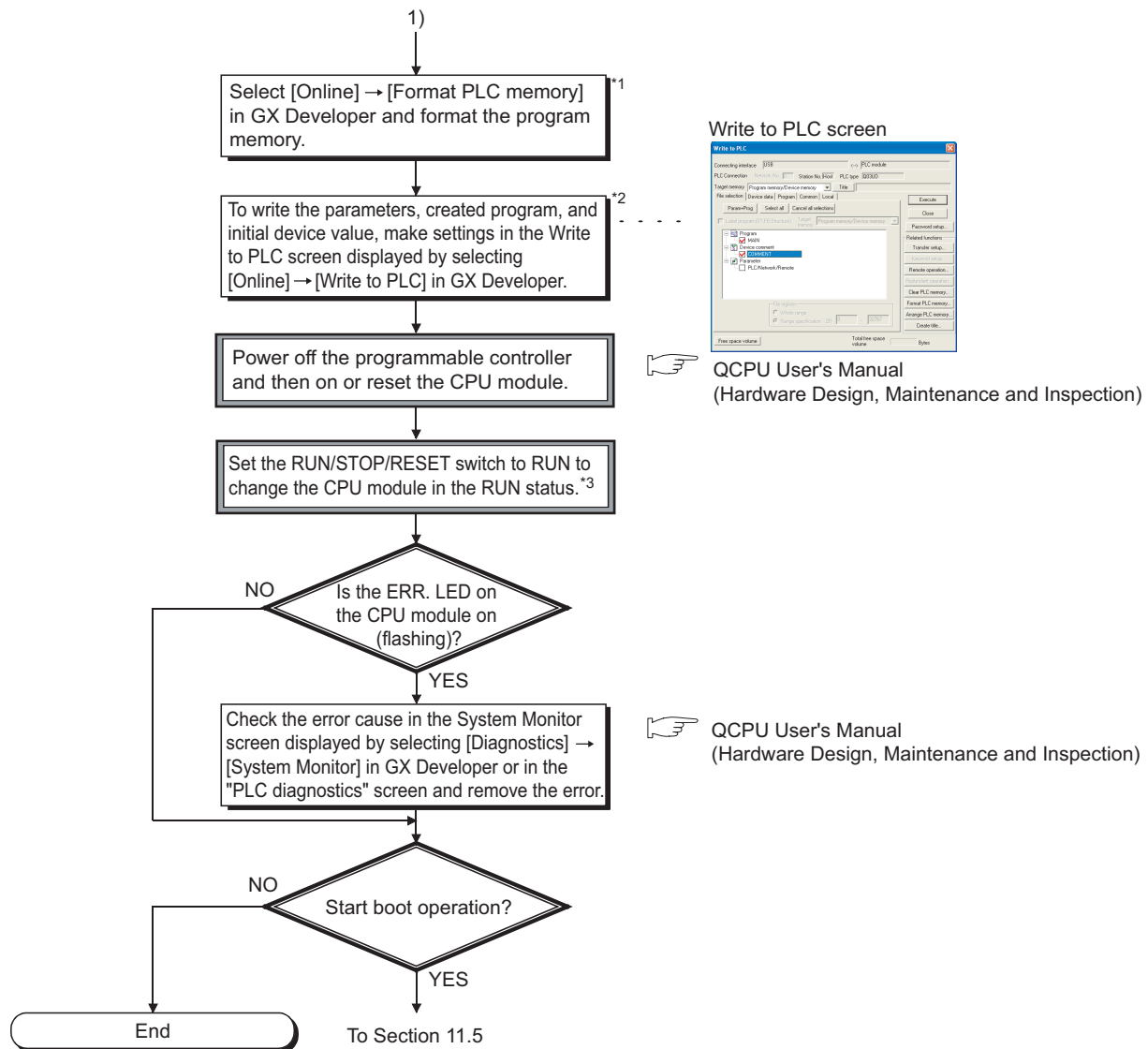


Figure 11.2 Flowchart for writing one program

*1: When storing the file register and initial device value to the standard RAM or memory card (except the Flash card), format the memory used.

*2: Write each data in the memory as shown below:

- Program:Program memory
- Parameter:Memory set in the parameter-valid drive
- Initial device value:Memory set in the PLC file tab of the PLC parameter dialog box

*3: In CPU modules with large program memory capacity, it may take time to turn to the RUN status.

Point

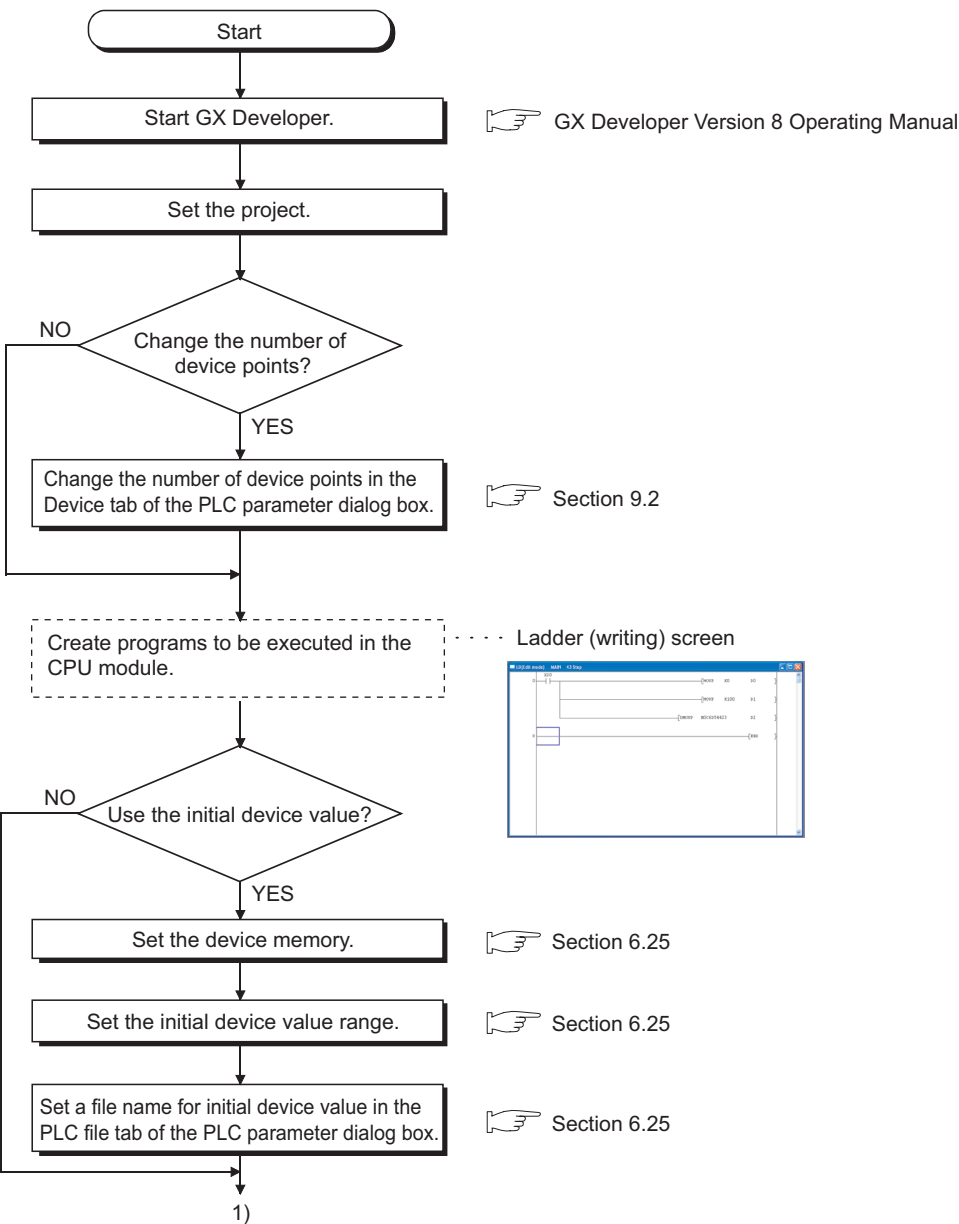
When the Universal model QCPU No.1 is the control CPU in the MELSECNET/H system, the Universal model QCPU can communicate with the Universal model QCPU where MISSING PARA. has been detected via the MELSECNET/H (for the network number 1 only).

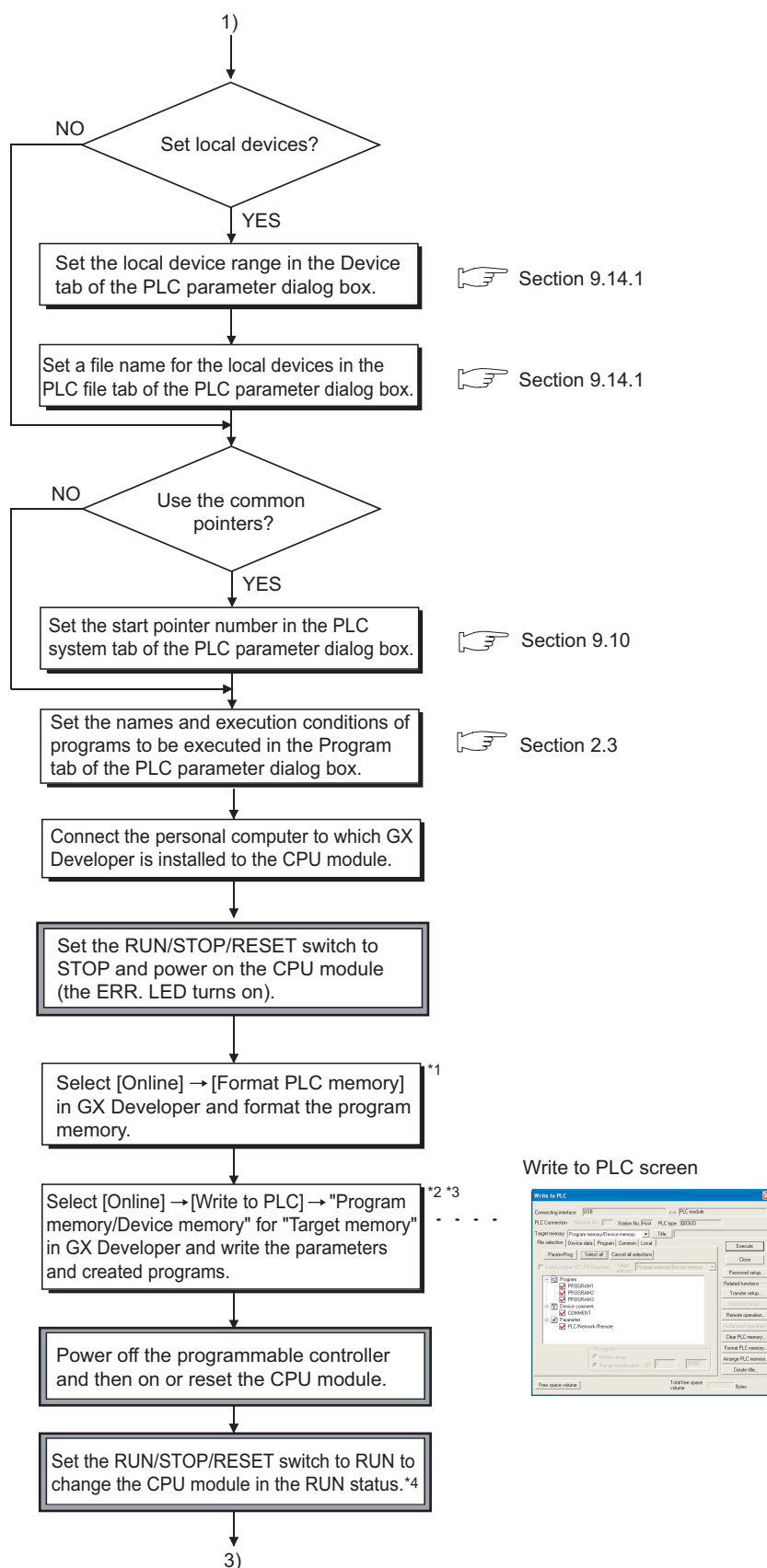
11.4 Procedure for Writing Multiple Programs

This section describes a procedure for writing multiple programs to the program memory. (☞ Section 5.1.2)

Follow the procedure below and then the procedure provided in Section 11.5 before storing the programs in the memory card for boot operation.

In the following procedure, ☐ indicates an operation on the CPU module side.





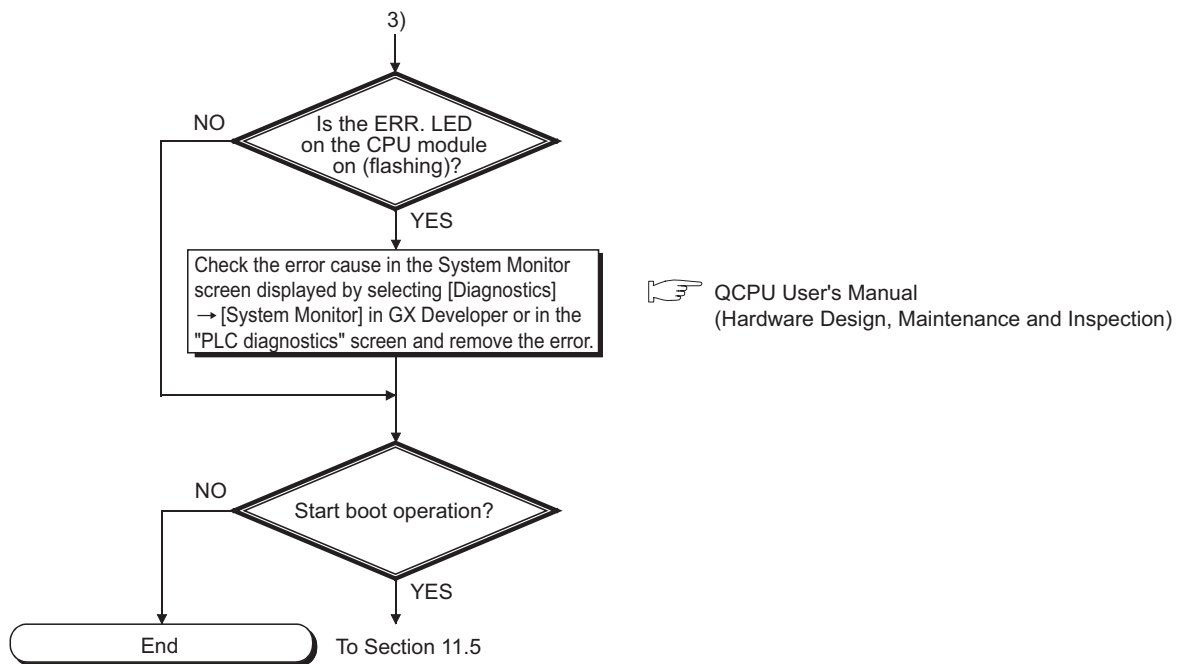


Figure 11.3 Flowchart for writing multiple programs

- *1: When storing the file register and initial device value to the standard RAM or memory card (except the Flash card), format the memory used.
- *2: Write each data in the memory as shown below:
 - Program:Program memory
 - Parameter:Memory set in the parameter-valid drive
 - Initial device value:Memory set in the PLC file tab of the PLC parameter dialog box
- *3: When executing the same number of programs of the maximum storable number of files in the program memory, write data other than programs (such as parameter files) into memories other than the program memory.
- *4: In CPU modules with large program memory capacity, it may take time to turn to the RUN status.

Point

When the Universal model QCPU No.1 is the control CPU in the MELSECNET/H system, the Universal model QCPU can communicate with the Universal model QCPU where MISSING PARA. has been detected via the MELSECNET/H (for the network number 1 only).

11.5 Procedure for Boot Operation

This section describes a procedure for boot operation.

In the following procedure,  indicates an operation on the CPU module side.

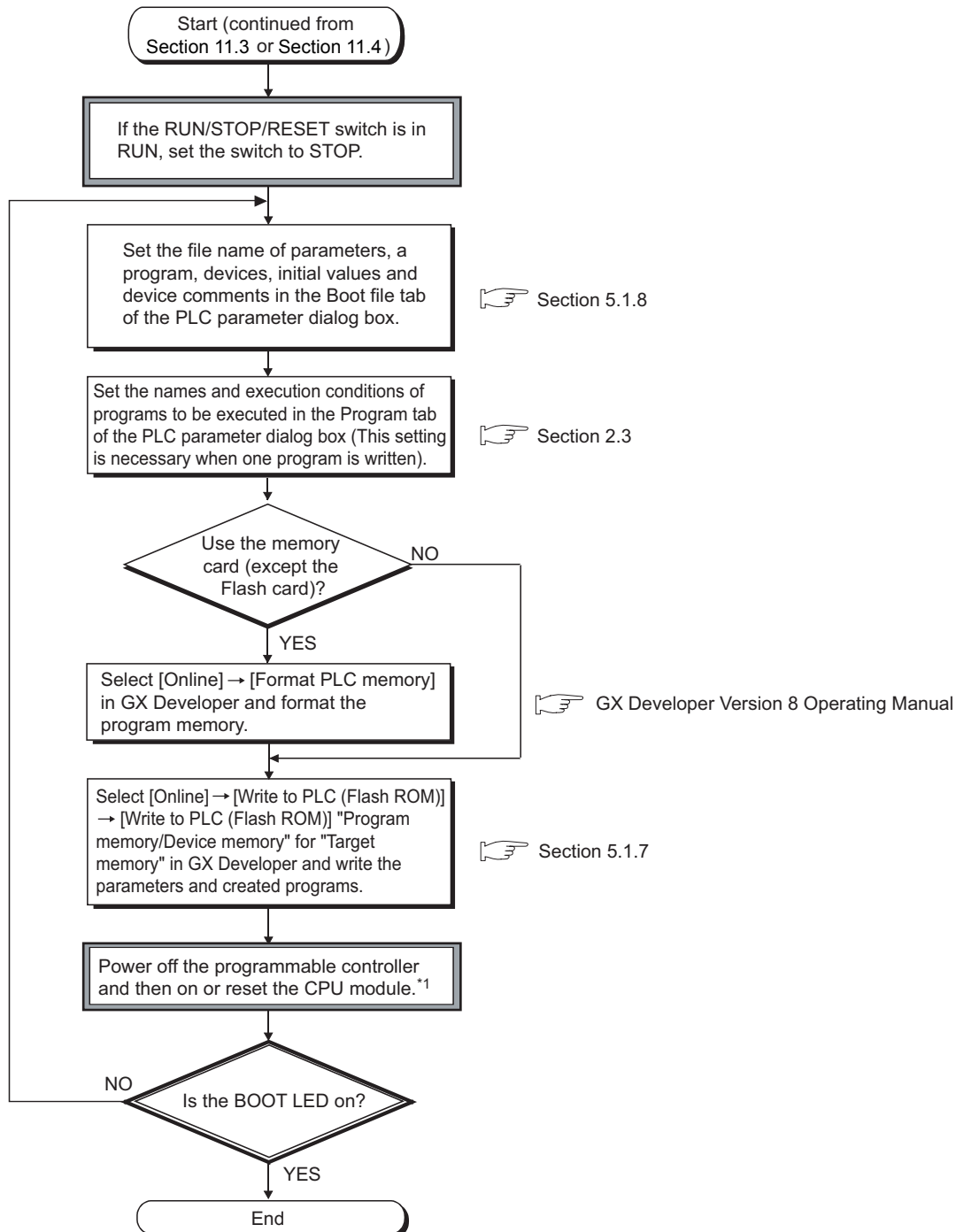


Figure 11.4 Flowchart for boot operation

*1: In CPU modules with large program memory capacity, it may take time to turn to the RUN status.

CHAPTER12 SPECIAL RELAY LIST AND SPECIAL REGISTER LIST

12.1 Special Relay List

The special relay (SM) is an internal relay whose application is fixed in the programmable controller. For this reason, the special relay cannot be used in the same way as other internal relays are used in sequence programs. However, the bit of the special relay can be turned on or off as needed to control the CPU module.

The following table shows how to read the special relay list.

Item	Description
Number	Special relay number
Name	Special relay name
Meaning	Contents of special relay
Explanation	Detailed description of special relay
Set by (When Set)	Set side and set timing of special relay <Set by> • S: Set by system • U: Set by user (using a program, programming tool, GOT, or test operation from other external devices) • S/U: Set by both system and user <When Set> The following shows the set timing when the special relay is set by system. • Every END processing: Set during every END processing • Initial: Set during initial processing (after power-on or status change from STOP to RUN) • Status change: Set when the operating status is changed • Error: Set if an error occurs • Instruction execution: Set when an instruction is executed • Request: Set when requested by a user (using the special relay) • When system is switched: Set when the system is switched (between the control system and the standby system)
Corresponding CPU	CPU module supporting the special relay • QCPU: All the Q series CPU modules • Q00J/Q00/Q01: Basic model QCPU • Qn(H): High Performance model QCPU • QnPH: Process CPU • QnPRH: Redundant CPU • QnU: Universal model QCPU • Q00UJ/Q00U/Q01U: Q00UJCPU, Q00UCPU, and Q01UCPU • LCPU: All the L series CPU modules • CPU module model: Only the specified model (Example: Q02UCPU, L26CPU-BT)
Corresponding ACPU M9□□□	• Special relay (M9□□□) supported by the ACPUs ("M9□□□ format change" indicates the one whose application has been changed. Incompatible with the Q00J/Q00/Q01 and QnPRH.) • "New" indicates the one added for the QCPU or LCPU.

For details on the following items, refer to the following.

- For network related items:  Manuals for each network module
- For SFC programs:  MELSEC-Q/L/QnA Programming Manual (SFC)

Point

Do not change the values of special relay set by system using a program or by test operation. Doing so may result in system down or communication failure.

(1) Diagnostic information

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU M9□□□	Corresponding CPU
SM0	Diagnostic errors	OFF : No error ON : Error	- This relay turns on if an error is detected by diagnostics. (Also turns on if an error is detected by an annunciator or the CHK instruction.) - This relay remains on even after the system returns to normal.	S (Error)	New	Qn(H) QnPH QnPRH
			- This relay turns on if an error is detected by diagnostics. (Also turns on if an error is detected by an annunciator.) - This relay remains on even after the system returns to normal.			Q00J/Q00/Q01 QnU LCPU
SM1	Self-diagnostic error	OFF : No self-diagnosis errors ON : Self-diagnosis	- This relay turns on if an error is detected by self-diagnostics. (Remains off if an error is detected by an annunciator or the CHK instruction.) - This relay remains on even after the system returns to normal.		M9008	Qn(H) QnPH QnPRH
			- This relay turns on if an error is detected by self-diagnostics. (Remains off if an error is detected by an annunciator.) - This relay remains on even after the system returns to normal.		New	Q00J/Q00/Q01 QnU LCPU
SM5	Error common information	OFF : No error common information ON : Error common information	This relay turns on if error common information data exists when SM0 turns on.			QCPU LCPU
SM16	Error individual information	OFF : No error individual information ON : Error individual information	This relay turns on if error individual information data exists when SM0 turns on.			
SM50	Error reset	OFF→ON: Error reset	Conducts error reset operation	U		
SM51	Battery low latch	OFF : Normal ON : Battery low	- This relay turns on if the battery voltage of the CPU module or the memory card drops below the rated value. - This relay remains on even after the battery voltage returns to normal. - Synchronizes with the BAT. LED.	S (Error)	M9007	Qn(H) QnPH QnPRH QnU LCPU
			- This relay turns on if the battery voltage of the CPU module drops below the rated value. - This relay remains on even after the battery voltage returns to normal. - Synchronous with the ERR. LED.		New	Q00J/Q00/Q01
SM52	Battery low	OFF : Normal ON : Battery low	This relay has the same specifications as those of SM51 except that this relay turns off after the battery voltage returns to normal.		M9006	QCPU LCPU

Number	Name	Meaning	Explanation		Set by (When Set)	Corres- ponding ACPU M9□□□	Corresponding CPU	
SM53	AC/DC DOWN detection	OFF : AC/DC DOWN not detected	This relay turns on if a momentary power failure within 20ms occurs during use of an AC power supply module. This relay is reset when the CPU module is powered off and then on.		S (Error)	M9005	QCPU	
		ON : AC/DC DOWN detected	This relay turns on if a momentary power failure within 10ms occurs during use of an AC power supply module. This relay is reset when the CPU module is powered off and then on.				LCPU	
SM53	AC/DC DOWN detection	OFF : AC/DC DOWN not detected ON : AC/DC DOWN detected	This relay turns on if a momentary power failure within 10ms occurs during use of a DC power supply module. This relay is reset when the CPU module is powered off and then on.			M9005	QCPU LCPU	
SM56	Operation error	OFF : Normal ON : Operation error	• This relay turns on if an operation error occurs. • This relay remains on even after the system returns to normal.			M9011		
SM60	Blown fuse detection	OFF : Normal ON : Module with blown fuse	• Turns on if there is at least one output module whose fuse has blown. • This relay remains on even after the system returns to normal. • Output modules on remote I/O stations are also checked.			M9000	QCPU	
SM61	I/O module verify error	OFF : Normal ON : Error	• This relay turns on if the status of the I/O module differs from that registered at power-on. • This relay remains on even after the system returns to normal. • I/O modules on remote I/O stations are also checked.			M9002	QCPU LCPU	
SM62	Annunciator detection	OFF : Not detected ON : Detected	This relay turns on if at least one annunciator (F) turns on.			M9009		
SM80	CHK detection	OFF : Not detected ON : Detected	• This relay turns on if an error is detected by the CHK instruction. • This relay remains on even after the system returns to normal.			S (Instruction execution)	New	Qn(H) QnPH QnPRH
SM84	Error clear	OFF→ON: Error clear	This relay is turned on to clear an error set to SD84 and SD85.					LCPU
SM90	Startup of monitoring timer for step transition (Enabled only when SFC program exists)	OFF : Not started (monitoring timer reset) ON : Started (monitoring timer started)	Corresponds to SD90	• Goes ON when measurement of step transition monitoring timer is commenced. • Resets step transition monitoring timer when it goes OFF.		U	M9108	Qn(H) QnPH QnPRH
SM91			Corresponds to SD91		M9109			
SM92			Corresponds to SD92		M9110			
SM93			Corresponds to SD93		M9111			
SM94			Corresponds to SD94		M9112			
SM95			Corresponds to SD95		M9113			
SM96			Corresponds to SD96		M9114			
SM97			Corresponds to SD97		New			
SM98			Corresponds to SD98					
SM99			Corresponds to SD99					

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU M9□□□	Corresponding CPU
SM100	Serial communication function using flag	OFF : Serial communication function is not used. ON : Serial communication function is used.	Stores the information of the serial communication setting configured in the PLC parameter dialog box.	S (Power-on or reset)	New	Q00/Q01 Q00UJ/Q00U/ Q01U Q02UCPU* ²
SM101	Communication protocol status flag	OFF : Programming tool ON : MC protocol communication device	This relay stores whether a device communicating via RS-232 interface is a programming tool or MC protocol communication device.	S (RS232 communication)		Q00/Q01 Q00UJ/Q00U/ Q01U Q02UCPU* ²
		Communication with programming tool	This relay is always off.(communication with a programming tool)			LCPU
SM110	Protocol error	OFF : Normal ON : Abnormal	- Turns on if a failed protocol was used to make communication in the serial communication function. - This relay remains on even after the protocol returns to normal.	S (Error)		Q00/Q01 Q00UJ/Q00U/ Q01U Q02UCPU* ²
SM111	Communication status	OFF : Normal ON : Abnormal	- Turns on if the mode used to make communication was different from the mode set in the serial communication function. - This relay remains on even after the system returns to normal.			
SM112	Error information clear	ON : Cleared	This relay is turned on to clear error codes stored in SM110, SM111, SD110, and SD111. The error codes are cleared when this relay is turned on.	U		
SM113	Overrun error	OFF : Normal ON : Abnormal	This relay turns on if an overrun error occurs in communication using the serial communication function.	S (Error)		
SM114	Parity error	OFF : Normal ON : Abnormal	This relay turns on if a parity error occurs in communication using the serial communication function.			
SM115	Framing error	OFF : Normal ON : Abnormal	This relay turns on if a flaming error occurs in communication using the serial communication function.			
SM165	Program memory batch transfer execution status	OFF : Completed ON : Not being executed or Not completed	- This relay turns on when data are written to the program cache memory. - This relay turns off when program memory batch transfer is completed. - This relay remains on when data written to the program cache memory are not batch-transferred to the program memory.	S (Status change)		QnU* ¹ LCPU

*1 The relevant modules are as follows:

- The Universal model QCPU whose serial number (first five digits) is "10012" or later.
- Q13UDHCPU, Q26UDHCPU

*2 The module whose first 5 digits of serial No. is "10102" or later.

(2) System information

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU M9□□□	Corres- ponding CPU
SM202	LED OFF command	OFF→ON : LED OFF	When this relay turns on from off, the LED corresponding to each bit in SD202 turns off.	U	New	Qn(H) QnPH QnPRH QnU LCPU
SM203	STOP contact	STOP status	This relay is on when the module is in the STOP status.	S (Status change)	M9042	QCPU LCPU
SM204	PAUSE contact	PAUSE status	This relay is on when the module is in the PAUSE status.		M9041	
SM206	PAUSE enable coil	OFF : PAUSE disabled ON : PAUSE enabled	PAUSE status is entered if this relay is ON when the PAUSE contact goes ON	U	M9040	
SM210	Clock data set request	OFF : Ignored ON : Set request	Clock data stored in SD210 to SD213 are written to the CPU module after the END instruction execution in the scan where this relay is turned on.		M9025	
SM211	Clock data error	OFF : No error ON : Error	This relay turns on if an error occurs in the clock data (SD210 to SD213), and is off while there is no error.	S (Request)	M9026	
SM213	Clock data read request	OFF : Ignored ON : Read request	This relay is turned on to read clock data and store them as BCD values into SD210 to SD213.	U	M9028	
SM220	CPU No.1 preparation completed	OFF : CPU No.1 preparation uncompleted ON : CPU No.1 preparation completed	Turns on when an access to CPU No.1 from another CPU becomes possible after power-on or reset operation. This relay is used as an interlock for accessing CPU No.1 when the multiple CPU synchronous setting is set to asynchronous.	S (When status changed)	New	QCPU
SM221	CPU No.2 preparation completed	OFF : CPU No.2 preparation uncompleted ON : CPU No.2 preparation completed	Turns on when an access to CPU No.2 from another CPU becomes possible after power-on or reset operation. This relay is used as an interlock for accessing CPU No.2 when the multiple CPU synchronous setting is set to asynchronous.			QnU*7
SM222	CPU No.3 preparation completed	OFF : CPU No.3 preparation uncompleted ON : CPU No.3 preparation completed	Turns on when an access to CPU No.3 from another CPU becomes possible after power-on or reset operation. This relay is used as an interlock for accessing CPU No.3 when the multiple CPU synchronous setting is set to asynchronous.			
SM223	CPU No.4 preparation completed	OFF : CPU No.4 preparation uncompleted ON : CPU No.4 preparation completed	Turns on when an access to CPU No.4 from another CPU becomes possible after power-on or reset operation. This relay is used as an interlock for accessing CPU No.4 when the multiple CPU synchronous setting is set to asynchronous.			QnU*5
SM235	Online module change flag	OFF : Online module change is not in progress ON : Online module change in progress	This relay is on during online module change. (for host CPU)	S (During online module change)		QnPH
SM236	Online module change complete flag	OFF : Online module change incomplete ON : Online module change complete	- This relay is on only for one scan after completion of online module change. - This relay can be used only in the scan execution type program. (for host CPU)	S (When online module change is complete)		

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU M9□□□	Corres- ponding CPU
SM237	Device range check inhibit flag	OFF : Device range checked ON : Device range not checked	Selects whether to check a device range during execution of the BMOV, FMOV or DFMov instruction (only when the conditions for subset processing are established).	U		QnU* ⁶ LCPu
SM240	No. 1 CPU reset flag	OFF : No. 1 CPU reset cancel ON : No. 1 CPU resetting	• Turns off when CPU No.1 is reset. • Turns on while CPU No.1 is being reset (including the case where the CPU module is removed from the base unit). The other CPUs are also put in reset status.	S (Status change)	New	Q00/Q01* ¹ Qn(H)* ¹ QnPH QnU* ⁷
	No. 1 CPU reset flag	Reset status	• This relay is always off. (reset status)			LCPu
SM241	No. 2 CPU reset flag	OFF : No. 2 CPU reset cancel ON : No. 2 CPU resetting	• Turns off when CPU No.2 is reset. • Turns on while CPU No.2 is being reset (including the case where the CPU module is removed from the base unit). "MULTI CPU DOWN" (error code: 7000) is detected on the other CPUs.			Q00/Q01* ¹ Qn(H)* ¹ QnPH QnU* ⁷
SM242	No. 3 CPU reset flag	OFF : No. 3 CPU reset cancel ON : No. 3 CPU resetting	• Turns off when CPU No.3 is reset. • Turns on while CPU No.3 is being reset (including the case where the CPU module is removed from the base unit). "MULTI CPU DOWN" (error code: 7000) is detected on the other CPUs.			Qn(H)* ¹ QnPH QnU* ⁵
SM243	No. 4 CPU reset flag	OFF : No. 4 CPU reset cancel ON : No. 4 CPU resetting	• Turns off when CPU No.4 is reset. • Turns on while CPU No.4 is being reset (including the case where the CPU module is removed from the base unit). "MULTI CPU DOWN" (error code: 7000) is detected on the other CPUs.			Q00/Q01* ¹ Qn(H)* ¹ QnPH QnU* ⁷ LCPu
SM244	No. 1 CPU error flag	OFF : No. 1 CPU normal ON : No. 1 CPU during stop error	• This relay is off when CPU No.1 is normal (including the case where a continuation error has occurred). • This relay is on when CPU No.1 has a stop error.			Q00/Q01* ¹ Qn(H)* ¹ QnPH QnU* ⁷ LCPu
SM245	No. 2 CPU error flag	OFF : No. 2 CPU normal ON : No. 2 CPU during stop error	• This relay is on when CPU No.2 is normal (including the case where a continuation error has occurred). • This relay is on when CPU No.2 has a stop error.			Q00/Q01* ¹ Qn(H)* ¹ QnPH QnU* ⁷
SM246	No. 3 CPU error flag	OFF : No. 3 CPU normal ON : No. 3 CPU during stop error	• This relay is off when CPU No.3 is normal (including the case where a continuation error has occurred). • This relay is on when CPU No.3 has a stop error.			Qn(H)* ¹ QnPH QnU* ⁵
SM247	No. 4 CPU error flag	OFF : No. 4 CPU normal ON : No. 4 CPU during stop error	• This relay is off when CPU No.4 is normal (including the case where a continuation error has occurred). • This relay is on when CPU No.4 has a stop error.			

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU M9□□□	Corres- ponding CPU
SM250	Max. loaded I/O read	OFF : Ignored ON : Read	When this relay turns on from off, the largest I/O number among those of the mounted modules is read into SD250.			
SM254	All stations refresh command	OFF : Refresh arrival station ON : Refresh all stations	- Effective for the batch refresh (also effective for the low speed cyclic) - Designate whether to receive arrival stations only or to receive all slave stations in the MELSECNET/H. - Effective for the batch refresh (also effective for the low speed cyclic) - Designate whether to receive arrival stations only or to receive all slave stations in the CC-Link IE controller network .	U		Qn(H) QnPH QnPRH
SM254	All stations refresh command	OFF : Refresh arrival station ON : Refresh all stations	- Effective for the batch refresh (also effective for the low speed cyclic) - Specify whether to receive only arrival station or all stations in the MELSECNET/H or CC-Link IE controller network.	U		QnU
SM255	MELSECNET/10, MELSECNET/H module 1 information	OFF : Operative network ON : Standby network	Turns on when it belongs to the standby network. (If no specification has been made, it is set to the operative network.)	S (Initial)	New	Qn(H) QnPH QnPRH
SM256		OFF : Reads ON : Does not read	For refresh in the direction from the link module to the CPU module (B, W, etc.), specify whether to read data from the link module or not.	U		
SM257		OFF : Writes ON : Does not write	For refresh in the direction from the CPU module to the link module (B, W, etc.), specify whether to write data to the link module or not.	U		
SM260	MELSECNET/10, MELSECNET/H module 2 information	OFF : Operative network ON : Standby network	Turns on when it belongs to the standby network. (If no specification has been made, it is set to the operative network.)	S (Initial)		
SM261		OFF : Reads ON : Does not read	For refresh in the direction from the link module to the CPU module (B, W, etc.), specify whether to read data from the link module or not.	U		
SM262		OFF : Writes ON : Does not write	For refresh in the direction from the CPU module to the link module (B, W, etc.), specify whether to write data to the link module or not.	U		
SM265	MELSECNET/10, MELSECNET/H module 3 information	OFF : Operative network ON : Standby network	Turns on when it belongs to the standby network. (If no specification has been made, it is set to the operative network.)	S (Initial)		
SM266		OFF : Reads ON : Does not read	For refresh in the direction from the link module to the CPU module (B, W, etc.), specify whether to read data from the link module or not.	U		
SM267		OFF : Writes ON : Does not write	For refresh in the direction from the CPU module to the link module (B, W, etc.), specify whether to write data to the link module or not.	U		
SM270	MELSECNET/10, MELSECNET/H module 4 information	OFF : Operative network ON : Standby network	Turns on when it belongs to the standby network. (If no specification has been made, it is set to the operative network.)	S (Initial)		
SM271		OFF : Reads ON : Does not read	For refresh in the direction from the link module to the CPU module (B, W, etc.), specify whether to read data from the link module or not.	U		
SM272		OFF : Writes ON : Does not write	For refresh in the direction from the CPU module to the link module (B, W, etc.), specify whether to write data to the link module or not.	U		
SM280	CC-Link error	OFF : Normal ON : Error	Turns on if a CC-Link error is detected in any of the CC-Link modules installed. Turns off when the condition returns to normal.	S (Status change)		

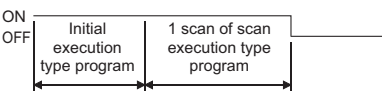
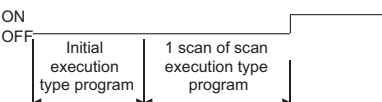
Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU M9□□□	Corres- ponding CPU
SM310	RS-232 adapter	OFF : Not connected ON : Connected	This relay stores information on whether an RS-232 adapter is connected or not. Connection of an RS-232 adapter is checked at the time of initialization, and if it is connected, this relay turns on. The on/off status set during initialization is held until the power is turned off and on again or the system is reset.	S (Initial)	New	LCPU
SM315	Communication reserved time delay enable/disable flag	OFF : Without delay ON : With delay	- This flag is enabled when the time reserved for communication processing is set in SD315. - Turns ON to delay the END processing by the time set in SD315 in order to perform communication processing. (The scan time increases by the period set in SD315.) - Turns OFF to perform the END processing without a delay of the time set in SD315 when there is no communication processing. (Defaults to OFF)	U		Q00J/Q00/ Q01
SM319	Automatic CC-Link start	OFF : Not activated ON : Activated	- This relay indicates whether the CC-Link module is started and all the data are refreshed by the automatic CC-Link start function. - This relay is on when all the data are refreshed by the automatic CC-Link start function. - Then the automatic CC-Link start function is not activated, or when the refresh device range is insufficient, this relay is turned off. (If the refresh device range set for the automatic CC-Link start function is insufficient, all of the refresh is stopped.)	S (Initial processing and status change)		LCPU
SM320	Presence/absence of SFC program	OFF : SFC program absent ON : SFC program present	- This relay is on if an SFC program is registered. - This relay turns off if no SFC program is registered.	S (Initial)	M9100	Q00J/Q00/ Q01*1 Qn(H) QnPH QnPRH QnU LCPU
SM321	Start/stop SFC program	OFF : SFC program not executed (stop) ON : SFC program executed (start)	- The same value as in SM320 is set as the initial value. (This relay turns on when an SFC program is registered.) - Turning off this relay stops SFC program execution. - Turning on this relay restarts SFC program execution.	S (Initial)/U	M9101 format change	
SM322	SFC program start status	OFF : Initial start ON : Resume start	In the SFC setting of the PLC Parameter dialog box, Initial start is set for the SFC program start mode. - At initial start: OFF - At continued start: ON		M9102 format change	
SM323	Presence/absence of continuous transition for entire block	OFF : Continuous transition not effective ON : Continuous transition effective	Set the presence/absence of continuous transition for the block where "Continuous transition bit" of the SFC data device has not been set.	U	M9103	
SM324	Continuous transition prevention flag	OFF : When transition is executed ON : When no transition	- This relay is off while the module is in the continuous transition mode or during continuous transition, and is on when continuous transition is not executed. - This relay is always on while the CPU module is operating not in the continuous transition mode.	S (Instruction execution)	M9104	
				S (Status change)	New	Q00J/Q00/ Q01*1 Qn(H) QnPH QnPRH QnU

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU M9□□□	Corres- ponding CPU
SM325	Output mode at block stop	OFF : OFF ON : Preserves	Select whether the coil outputs of the active steps are held or not at the time of a block stop. • The initial value is set to off when the output mode at a block stop is off in the parameter setting, and it is set to on when the coil outputs are set to be held. • When this relay is turned off, all coil outputs are turned off. • When this relay is turned on, the coil output state is held.	S (Initial)/U	M9196	Q00J/Q00/ Q01* ¹ Qn(H) QnPH QnPRH QnU LCPU
SM326	SFC device clear mode	OFF : Clear device ON : Preserves device	Select the device status at the time of switching from STOP to program write, and then to RUN. (All devices except the step relay)	U	New	
SM327	Output during end step execution	OFF : Hold step output turned OFF (cleared) ON : Hold step output held	If this relay is off, the coil output turns off when the step held after transition (SC, SE, or ST) reaches the end step.	S (Initial)/U		Qn(H) QnPH QnPRH QnU LCPU
						Q00J/Q00/ Q01* ¹
SM328	Clear processing mode when end step is reached	OFF : Clear processing is performed. ON : Clear processing is not performed.	Select whether clear processing will be performed or not if active steps other than the ones being held exist in the block when the end step is reached. • When this relay turns OFF, all active steps are forcibly terminated to terminate the block. • When this relay is ON, the execution of the block is continued as-is. • If active steps other than the ones being held do not exist when the end step is reached, the steps being held are terminated to terminate the block.	U		Q00J/Q00/ Q01* ¹ QnU LCPU
SM329	Online change (inactive block) status flag	OFF : Not executed ON : Being executed	This relay is on while online change (inactive block) is executed.	S (Status change)		QnU* ⁸
SM330	Operation mode for low speed execution type program	OFF : Asynchronous mode ON : Synchronous mode	Select whether the low speed execution type program will be executed in the asynchronous mode or in the synchronous mode. • Asynchronous mode (this relay is turned off.) The operation of the low-speed execution type program is performed continuously within an excess time. • Synchronous mode (this relay is turned on.) The operation of the low-speed execution type program is not performed continuously, but performed from the next scan, even if there is excess time.	U		Qn(H) QnPH
SM331	Normal SFC program execution status	OFF : Not executed ON : Being executed	• This relay stores the information on whether the normal SFC program is in execution or not. • Used as an interlock for execution of the SFC control instruction.	S (Status change)		Qn(H)* ³ QnPH* ⁴ QnPRH
SM332	Program execution management SFC program execution status	OFF : Not executed ON : Being executed	• This relay stores the information on whether the SFC program for program execution management is in execution or not. • Used as an interlock for execution of the SFC control instruction.			
SM390	Access execution flag	ON indicates completion of intelligent function module access	• This relay stores the status information on the intelligent function module access instruction that was just executed. (This data is overwritten if the intelligent function module access instruction is executed again.) • Used by the user in a program as a completion bit.			Qn(H) QnPH QnPRH

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU M9□□□	Corres- ponding CPU
SM391	GINT instruction execution completion flag	OFF : Not executed ON : Execution completed	Stores the execution status of the S(P).GINT instruction. • Turns off before execution of the instruction. • Turns on after completion of the instruction.	S (Instruction execution)	New	QnU

- *1 This applies to the CPU of function version B or later.
 *2 The module whose first 5 digits of serial No. is "09012" or later.
 *3 The module whose first 5 digits of serial No. is "04122" or later.
 *4 The module whose first 5 digits of serial No. is "07032" or later.
 *5 The Universal model QCPU except the Q00UJCPU, Q00UCPU, Q01UCPU, and Q02UCPU.
 *6 The relevant modules are as follows:
 • The Universal model QCPU whose serial number (first five digits) is "10012" or later.
 • Q13UDHCPU, Q26UDHCPU
 *7 The Universal model QCPU except the Q00UJCPU.
 *8 This applies when the first five digits of the serial number is "12052" or later.

(3) System clock/counter

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU M9□□□	Corresponding CPU	
SM400	Always ON	ON _____ OFF	This relay is always on.	S (Every END processing)	M9036	QCPU LCPU	
SM401	Always OFF	ON OFF _____	This relay is always off.		M9037		
SM402	After RUN, ON for 1 scan only	ON _____ OFF ← 1 scan	• This relay turns on for one scan after the CPU module enters the RUN status. • This relay can be used only in a scan execution type program. • When an initial execution type program is used, this relay turns off at the END processing of the scan execution type program in the first scan after the CPU module enters the RUN status. 		M9038	Qn(H) QnPH QnPRH QnU LCPU	
			This relay turns on for one scan after the CPU module enters the RUN status.		New	Q00J/Q00/Q01	
SM403	After RUN, OFF for 1 scan only	ON ← 1 scan OFF _____	• This relay turns off for one scan after the CPU module enters the RUN status. • This relay can be used only in a scan execution type program. • When an initial execution type program is used, this relay turns on at the END processing of the scan execution type program in the first scan after the CPU module enters the RUN status. 		M9039	Qn(H) QnPH QnPRH QnU LCPU	
			This relay turns off for one scan after the CPU module enters the RUN status.		New	Q00J/Q00/Q01	
SM404	Low speed execution type program ON for 1 scan only after RUN	ON _____ OFF ← 1 scan	• This relay turns on for one scan after the CPU module enters the RUN status. • This relay can be used only in a low-speed execution type program.	S (Status change)		Qn(H) QnPH	
SM405	Low speed execution type program After RUN, OFF for 1 scan only	ON ← 1 scan OFF _____	• This relay turns off for one scan after the CPU module enters the RUN status. • This relay can be used only in a low-speed execution type program.				
SM409	0.01 second clock	0.005s _____ 0.005s	• This relay repeatedly turns on and off at 5-ms interval. • This relay starts with off at power-on or reset of the CPU module. (Note if the specified time has elapsed, on/off status will change even during program execution.)			Qn(H) QnPH QnPRH QnU LCPU	

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU M9□□□	Corresponding CPU
SM410	0.1 second clock		- This relay repeatedly turns on and off at the specified interval. - This relay starts with off at power-on or reset of the CPU module. (Note if the specified time has elapsed, on/off status will change even during program execution.)	S (Status change)	M9030	QCPU LCPU
SM411	0.2 second clock				M9031	
SM412	1 second clock				M9032	
SM413	2 second clock				M9033	
SM414	2n second clock				M9034 format change	
SM415	2n (ms) clock		- This relay repeatedly turns on and off at the interval specified in SD415 (unit: ms). - This relay starts with off at power-on or reset of the CPU module. (Note if the specified time has elapsed, on/off status will change even during program execution.)		New	Qn(H) QnPH QnPRH QnU LCPU
SM420	User timing clock No.0		- This relay repeatedly turns on and off at the specified scan intervals. - This relay starts with off at power-on or reset of the CPU module. (For the redundant CPU, however, this relay will become always off after system switching.) - The on/off scan intervals are set by the DUTY instruction. - n1: On scan interval - n2: Off scan interval	S (Every END processing)	M9020	QCPU LCPU
SM421	User timing clock No.1				M9021	
SM422	User timing clock No.2				M9022	
SM423	User timing clock No.3				M9023	
SM424	User timing clock No.4				M9024	
SM430	User timing clock No.5	For use with SM420 to SM424 low speed programs			New	Qn(H) QnPH
SM431	User timing clock No.6					
SM432	User timing clock No.7					
SM433	User timing clock No.8					
SM434	User timing clock No.9					

(4) Scan information

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU M9□□□	Corresponding CPU
SM510	Low speed program execution flag	OFF : Completed or not executed ON : Execution under way.	This relay is on while a low-speed execution type program is being executed.	S (Every END processing)	New	Qn(H) QnPH
SM551	Reads module service interval	OFF : Ignored ON : Read	When this relay is turned on, the service interval of the module specified by SD550 is read to SD551 and SD552.	U	New	Qn(H) QnPH QnPRH

(5) I/O refresh

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU M9□□□	Corresponding CPU
SM580	Program to program I/O refresh	OFF : Not refreshed ON : Refreshed	When this relay is turned on, I/O refresh is performed after execution of the first program, and then the next program is executed. When a sequence program and a SFC program are to be executed, the sequence program is executed, I/O refresh is performed, and then the SFC program is executed.	U	New	Q00J/Q00/Q01* ¹

*1 This applies to the CPU of function version B or later.

(6) Drive information

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU M9□□□	Corresponding CPU
SM600	Memory card usable flags	OFF : Unusable ON : Use enabled	This relay turns on when the memory card becomes ready for use.	S (Status change)	New	Qn(H) QnPH QnPRH QnU*1
			This relay turns on when the SD memory card becomes ready for use. (This relay turns on when a compatible SD memory card is inserted and set to be enabled with the SD memory card lock switch.)			LCPU
SM601	Memory card protect flag	OFF : No protect ON : Protect	This relay is on while the write-protect switch of the memory card is on.			Qn(H) QnPH QnPRH QnU*1 LCPU
SM602	Drive 1 flag	OFF : No drive 1 ON : Drive 1 present	This relay is on while a RAM is being inserted.			Qn(H) QnPH QnPRH QnU*1
SM603	Drive 2 flag	OFF : No drive 2 ON : Drive 2 present	This relay is on while a ROM is being inserted.			Qn(H) QnPH QnPRH QnU*1
			This relay is on while a SD memory card is being inserted. (This relay is on while a SD memory card is being inserted, regardless of the availability and the type of the card.)			LCPU
SM604	Memory card in- use flag	OFF : Not used ON : In use	This relay is on while a memory card is being used.	S (Status change)	New	Qn(H) QnPH QnPRH QnU*1 LCPU
SM605	Memory card remove/insert prohibit flag	OFF : Remove/insert enabled ON : Remove/insert prohibited	This relay is turned on to disable the insertion and removal of a memory card.	U		Qn(H) QnPH QnPRH QnU*1
			This relay is turned on to disable the insertion and removal of a memory card. (This relay turns on when a compatible SD memory card is inserted and set to be enabled with the SD memory card lock switch. This relay does not turn on while "ICM.OPE.ERROR" occurs.)	S (Status change)		LCPU
SM606	SD memory card forced disable instruction	OFF : SD memory card forced disable cancel instruction ON : SD memory card forced disable instruction	- This relay is turned on to execute the SD memory card forced disable instruction. When there are any functions accessing to an SD memory card, the process of disablement is held until it is completed. - This relay is turned off to cancel the SD memory card forced disable instruction.	U		LCPU
SM607	SD memory card forced disable status flag	OFF : Not being disabled by SD memory card forced disable instruction ON : Being disabled by SD memory card forced disable instruction	- This relay turns on when an SD memory card is disabled by turning on SM606 (SD memory card forced disable instruction). - This relay turns off when the forced disable status of SD memory card is canceled by turning off SM606 (SD memory card forced disable instruction).	S (Status change)		LCPU

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU M9□□□	Corresponding CPU
SM609	Memory card remove/insert enable flag	OFF : Remove/insert prohibited ON : Remove/insert enabled	• This relay is turned on to enable the insertion and removal of a memory card. • Turned OFF by the system after the memory card is removed. • This relay can be used while both SM604 and SM605 are off.	S/U	New	Qn(H) QnPH QnPRH QnU ^{*1}
SM620	Drive 3/4 usable flags	OFF : Unusable ON : Use enabled	This relay is always on.	S (Initial)		QCPU LCPU
SM621	Drives 3 and 4 protection flag	OFF : Not protected ON : Protected	This relay is always off.			Q00J/Q00/Q01 Qn(H) QnPH QnPRH QnU ^{*2} LCPU
SM622	Drive 3 flag	OFF : No drive 3 ON : Drive 3 present	This relay is always on.			QCPU LCPU
SM623	Drive 4 flag	OFF : No drive 4 ON : Drive 4 present	This relay is always on.			Qn(H) QnPH QnPRH QnU LCPU
SM624	Drive 3/4 in-use flag	OFF : Not used ON : In use	This relay is on while a file stored in the drive 3 (standard RAM) or the drive 4 (standard ROM) is being used.	S (Status change)		Q00J/Q00/Q01 Qn(H) QnPH QnPRH QnU ^{*2} LCPU
SM640	File register use	OFF : File register not used ON : File register in use	This relay is on while a file register is being used.			Qn(H) QnPH QnPRH QnU LCPU
SM650	Comment use	OFF : File register not used ON : File register in use	This relay is on while a comment file is being used.			Qn(H) QnPH QnPRH
SM660	Boot operation	OFF : Internal memory execution ON : Boot operation in progress	• This relay is on during boot operation. • This relay turns off when the boot specification switch is turned off.			Q00J/Q00/Q01 QnU ^{*3} LCPU
		OFF : Program memory execution ON : Boot operation in progress	This relay is on during boot operation.			QnU LCPU
SM671	Latch data backup to standard ROM completion flag	OFF : Not completed ON : Completed	• This relay turns on when latch data backup to the standard ROM is completed. • Time when the backup is completed is stored in SD672 or later.			QnU LCPU

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU M9□□□	Corresponding CPU
SM672	Memory card file register access range flag	OFF : Within access range ON : Outside access range	- This relay turns on when an area outside a file register range in a memory card is accessed. (This relay is set at END processing.) - This relay is reset from a program.	S/U	New	Qn(H) QnPH QnPRH
SM675	Error completion of latch data backup to standard ROM	OFF : No Error ON : Error	- This relay turns on if latch data backup to the standard ROM is not completed. - This relay turns off when the backup is completed.	S		QnU LCPU
SM676	Specification of restration repeated execution	OFF : Not specified ON : Specified	- When latch data are backed up while this relay is on, the backup data will be restored at every power-on of the CPU module. - The backup data will be restored at every power-on until the latch data are deleted or the latch data are backed up again.	U		
SM680	Program memory write error	ON : Write error OFF : Write not executed/normal	This relay turns on if a write error is detected during writing to the program memory (flash ROM). This relay turns off when a write command is given.	S (At write)		
SM681	Program memory writing flag	ON : During writing OFF : Write not executed	This relay is on during writing to the program memory (flash ROM) and turns off when the writing is completed.			
SM682	Program memory overwrite count error flag	ON : Overwrite count is 100,000 or more OFF : Overwrite count is less than 100,000	This relay turns on when overwrite count of the program memory (flash ROM) reaches to 100,000. (It is necessary to change CPU module.)			
SM685	Standard ROM write error	ON : Write error OFF : Write not executed/normal	This relay turns on if a write error is detected during writing to the standard ROM (flash ROM). This relay turns off when a write command is given.			
SM686	Standard ROM writing flag	ON : During overwriting OFF : Overwrite not executed	This relay is on during writing to the standard ROM (flash ROM) and turns off when the writing is completed.			
SM687	Standard ROM overwrite count error flag	ON : Overwrite count is 100,000 or more OFF : Overwrite count is less than 100,000	This relay turns on when overwrite count of the standard ROM (flash ROM) reaches to 100,000. (It is necessary to change CPU module.)			
SM691	Backup start preparation status flag	OFF : Backup start preparation not completed ON : Backup start preparation completed	Turns on when the backup preparation is completed.	S (Status change)		QnU*1 LCPU
SM692	Restoration complete flag	OFF : Restoration not completed ON : Restoration completed	This relay turns on when backup data in a memory card has been restored.			

*1 The modules whose serial number (first five digits) is "10102" or later are the relevant models. (Except the Q00UJCPU, Q00UCPU, and Q01UCPU)

*2 The Universal model QCPU except the Q00UJCPU.

*3 The Universal model QCPU except the Q00UJCPU, Q00UCPU, and Q01UCPU.

(7) Instruction-related relay

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU M9□□□	Corresponding CPU	
SM700	Carry flag	OFF : Carry OFF ON : Carry ON	Carry flag used in application instruction	S (Instruction execution)	M9012	QCPU LCPU	
SM701	Number of output characters selection	OFF : Output until NULL code encountered ON : 16 characters output	Used for the PR, PRC, BINDA, DBINDA, BINHA, DBINHA, BCDDA, DBCDDA, or COMRD instruction	U	M9049	Qn(H) QnPH QnPRH QnU LCPU	
SM702	Search method	OFF : Search next ON : 2-part search	- Designates method to be used by search instruction. - Data must be arranged for 2-part search.		QCPU LCPU		
SM703	Sort order	OFF : Ascending order ON : Descending order	The sort instruction is used to designate whether data should be sorted in ascending order or in descending order.				
SM704	Block comparison	OFF : Non-match found ON : All match	This relay turns on when all data conditions are met for the BKCMP instruction. This relay turns on when all data conditions are met for the DBKCMP instruction.	S (Instruction execution)	New	QnU*2 LCPU	
SM709	DT/TM instruction improper data detection flag	OFF : Improper data not detected ON : Improper data detected	This relay turns on when the data to be compared by the DT or TM instruction cannot be recognized as date or time data, when the device (three words) to be compared is exceeding the specified device range.	S (Instruction execution)/U			
SM710	CHK instruction priority ranking flag	OFF : Conditions priority ON : Pattern priority	- Remains as originally set when OFF. - Priority for the CHK instruction is changed when on.	S (Instruction execution)		Qn(H) QnPH QnPRH	
SM715	EI flag	OFF : During DI ON : During EI	This relay is on while the EI instruction is being executed.			QCPU LCPU	
SM716	Block comparison (Except an interrupt program)	OFF : Mismatch found ON : No mismatch	This relay turns on when all data conditions are met for the DBKCMP instruction. (Initial execution type program and scan execution type program or standby type program executed from initial execution type program or scan execution type program)			QnU*2 LCPU	
SM717	Block comparison (Interrupt program)	OFF : Mismatch found ON : No mismatch	This relay turns on when all data conditions are met for the DBKCMP instruction. (Interrupt program, fixed scan execution type program, or standby type program executed from interrupt program or fixed scan execution type program)				
SM718	Block comparison (Interrupt program (I45))	OFF : Mismatch found ON : No mismatch	This relay turns on when all data conditions are met for the DBKCMP instruction. (Interrupt program (I45) or standby type program that was executed from interrupt program (I45))			QnU*3	
SM720	Comment read completion flag	OFF : Comment read not completed ON : Comment read completed	This relay turns on only during first scan after the processing of the COMRD or PRC instruction is completed. This relay turns on only during first scan after the processing of the COMRD instruction is completed.	S (Status change)		Qn(H) QnPH	
							QnPRH QnU LCPU

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU M9□□□	Corresponding CPU
SM721	File being accessed	OFF : File not accessed ON : File being accessed	This relay is on while a file is being accessed by the SP. FWRITE, SP. FREAD, COMRD, PRC, or LEDC instruction.	S (Status change)	New	Qn(H) QnPH
			This relay is on while a file is being accessed by the SP. FWRITE, SP. FREAD, COMRD, or LEDC instruction.			Qn(H) QnPH QnPRH
			This relay is on while a file is being accessed by the SP. FWRITE, SP. FREAD, COMRD, or SP.DEVST instruction.			QnU
			- This relay is on while a file is being accessed by the SP. FWRITE, SP. FREAD, COMRD, or SP.DEVST instruction. - This relay is on while a SD memory card or the standard ROM is being accessed.			LCPU
			This relay is on while an ATA card or the standard ROM is being accessed.			QnU* ⁴
			This relay is on while the S(P).SFCSCOMR or the S(P).SFCTCOMR instruction is executed.			QnU* ¹¹
SM722	BIN/DBIN instruction error disabling flag	OFF : Error detection performed ON : Error detection not performed	Turned ON when "OPERATION ERROR" is suppressed for BIN or DBIN instruction.	U		QCPU LCPU
SM734	XCALL instruction execution condition designation	OFF : Not executed by execution condition risen ON : Executed by execution condition risen	- During OFF, XCALL instructions will not be executed even if execution condition is risen. - During ON, XCALL instructions will be executed when execution condition is risen.			Qn(H)* ⁴
SM735	SFC comment readout instruction in execution flag	OFF : SFC comment readout instruction is inactivated. ON : SFC comment readout instruction is activating.	This relay turns on while a SFC step comment readout instruction (S(P).SFCSCOMR) or SFC transmission condition comment readout instruction (S(P).SFCTCOMR) is being executed.	S (Status change)		Qn(H)* ⁵ QnPH* ⁶ QnPRH* ⁶ QnU* ¹¹
SM738	MSG instruction reception flag	OFF : Instruction not executed ON : Instruction execution	This relay turns on when the MSG instruction is executed.	S (Instruction execution)		Qn(H) QnPRH
SM740	Display unit availability flag	OFF : Not usable ON : Usable	This relay is on while the display unit can be used.	S (Initial/Status change)	LCPU	
SM750	Scaling instruction search method setting	OFF : Search next ON : 2-part search	Determines a search method when the scaling instruction is executed.	U	QnU* ² LCPU	
SM774	PID bumpless processing (for complete derivative)	OFF : Matched ON : Not matched	Specifies whether to match the set value (SV) with the process value (PV) or not in the manual mode.		Q00J/Q00/Q01* ¹¹ Qn(H) QnPRH QnU LCPU	

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU M9□□□	Corresponding CPU
SM775	Selection of refresh processing during COM/CCOM instruction execution	OFF : Performs link refresh ON : Performs no link refresh	Select whether link refresh processing will be performed or not when only communication with the CPU module is made at the execution of the COM instruction.	U		Q00J/Q00/Q01 Qn(H) QnPH
		OFF : Performs refresh processes other than an I/O refresh ON : Performs refresh set by SD778	Select whether to perform refresh processes other than an I/O refresh set by SD778 when the COM or CCOM instruction is executed.			Q00J/Q00/Q01* ¹ Qn(H)* ⁷ QnPH* ⁴ QnPRH QnU LCPU
SM776	Enable/disable local device at CALL	OFF : Local device disabled ON : Local device enabled	Set whether the local device of the subroutine program called at execution of the CALL instruction is valid or invalid.			Qn(H) QnPH QnPRH QnU* ¹⁰ LCPU
SM777	Enable/disable local device in interrupt program	OFF : Local device disabled ON : Local device enabled	Set whether the local device at execution of the interrupt program is valid or invalid.			
SM794	PID bumpless processing(for incomplete derivative)	OFF : Matched ON : Not matched	Specifies whether to match the set value (SV) with the process value (PV) or not in the manual mode.			Q00J/Q00/Q01* ¹ Qn(H)* ⁸ QnPRH QnU LCPU
SM796	Block information using multiple CPU high-speed transmission dedicated instruction (for CPU No.1)	OFF : Block is secured ON : Block set by SD796 cannot be secured	This relay turns on when the number of the remaining blocks in the dedicated instruction transmission area used for the multiple CPU high-speed transmission dedicated instruction (target CPU= CPU No.1) is less than the number of blocks specified in SD796. This relay is on when an instruction is executed, and is off while an END processing is being executed or when free space is available in the area.	S (When instruction/END processing executed)	New	QnU* ⁹
SM797	Block information using multiple CPU high-speed transmission dedicated instruction (for CPU No.2)	OFF : Block is secured ON : Block set by SD797 cannot be secured	This relay turns on when the number of the remaining blocks in the dedicated instruction transmission area used for the multiple CPU high-speed transmission dedicated instruction (target CPU= CPU No.2) is less than the number of blocks specified in SD797. This relay is on when an instruction is executed, and is off while an END processing is being executed or when free space is available in the area.			
SM798	Block information using multiple CPU high-speed transmission dedicated instruction (for CPU No.3)	OFF : Block is secured ON : Block set by SD798 cannot be secured	This relay turns on when the number of the remaining blocks in the dedicated instruction transmission area used for the multiple CPU high-speed transmission dedicated instruction (target CPU= CPU No.3) is less than the number of blocks specified in SD798. This relay is on when an instruction is executed, and is off while an END processing is being executed or when free space is available in the area.			

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU M9□□□	Corresponding CPU
SM799	Block information using multiple CPU high-speed transmission dedicated instruction (for CPU No.4)	OFF : Block is secured ON : Block set by SD799 cannot be secured	This relay turns on when the number of the remaining blocks in the dedicated instruction transmission area used for the multiple CPU high-speed transmission dedicated instruction (target CPU= CPU No.) is less than the number of blocks specified in SD799. This relay is on when an instruction is executed, and is off while an END processing is being executed or when free space is available in the area.	S (When instruction/END processing executed)	New	QnU ^{*9}

- *1 This applies to the CPU module of function version B or later.
- *2 The relevant modules are as follows:
 •The Universal model QCPU whose serial number (first five digits) is "10102" or later.
 •Q00UJCPU, Q00UCPU, Q01UCPU
- *3 The relevant modules are as follows:
 •The Universal model QCPU whose serial number (first five digits) is "10102" or later.
 •Q00UCPU, Q01UCPU
- *4 The module whose first 5 digits of serial No. is "07032" or later.
- *5 The module whose first 5 digits of serial No. is "06082" or later.
- *6 The module whose first 5 digits of serial No. is "07012" or later.
- *7 The module whose first 5 digits of serial No. is "04012" or later.
- *8 The module whose first 5 digits of serial No. is "05032" or later.
- *9 The Universal model QCPU except the Q00UJCPU, Q00UCPU, Q01UCPU, and Q02UCPU.
- *10 The Universal model QCPU except the Q00UJCPU.
- *11 This applies when the first five digits of the serial number is "12052" or later.

(8) Debugging

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU M9□□□	Corresponding CPU
SM800	Trace preparation	OFF : Not ready ON : Ready	Turns on when the trace preparation is completed.	S (Status change)	New	Qn(H) QnPH QnPRH QnU* ¹ LCPU
SM801	Trace start	OFF : Suspend ON : Start	- When this relay is turned on, a trace is started. - When this relay is turned off, a trace is stopped. (The related special relays will all turn off)	U	M9047	
SM802	Trace execution in progress	OFF : Suspend ON : Start	This relay is on while a trace is being executed.	S (Status change)	M9046	
SM803	Trace trigger	OFF→ON: Start	- This relay turns on when the specified trigger condition is met. - This relay is turned on to meet the trigger condition.	S (Status change)/U	New	
SM804	After trace trigger	OFF : Not after trigger ON : After trigger	Turns on after trace is triggered.	S (Status change)	New	
SM805	Trace completed	OFF : Not completed ON : End	This relay turns on when a trace is completed.		M9043	
SM826	Trace error	OFF : Normal ON : Errors	This relay turns on if an error occurs during trace.		New	QnU* ¹ LCPU
SM829	Forced registration specification of trace setting	ON : Forced registration enabled OFF : Forced registration disabled	When this relay is turned on and a sampling trace setting is registered using a programming tool, the sampling trace setting can be registered with the CPU module even when the trigger condition has been met.	U		
SM841	Auto logging	ON : Auto logging OFF : No auto logging	This relay is on while auto logging is being executed. This relay turns off when auto logging is completed and the SD memory card lock switch is slid toward the module top to stop access to the SD memory card.	S (Status change)		LCPU

*1 The Universal model QCPU except the Q00UJCPU.

(9) Conversion from A series to Q or L series

The special relay (M9000 to M9255) for ACPU corresponds to the special relay (SM1000 to SM1255) for QCPU or LCPU after the A to Q/L conversion. (Note that the Basic model QCPU and Redundant CPU do not support the A to Q/L conversion.) All bits in this area of the special relay are turned on or off by system (cannot be turned on or off by user using a program). To turn on or off the bit by user, correct the program using the special relay for QCPU or LCPU. The special relay (M9084, M9200 to M9255), however, includes the areas that can be turned on or off by user. For those areas, the bit can be turned on or off by user in the converted special relay (SM1084, SM1200 to SM1255) as well. For details on the special relay for ACPU, refer to the following.



User's manual for the CPU module used



Type MELSECNET, MELSECNET/B Data Link System Reference Manual

Point

To use the converted special relay in the High Performance model QCPU, Process CPU, Universal model QCPU, or LCPU, check "Use special relay/special register from SM/SD1000" under "A-PLC Compatibility Setting".

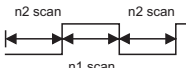
Project window ⇨ [Parameter] ⇨ [PLC Parameter] ⇨ [PLC System]

Note that the processing time will increase when the converted special relay is used.

[How to read the Special Relay for Modification column]

- If the special relay number for QCPU or LCPU is provided, correct the program using it.
- ☐ means that the converted special relay can be used.
- ☒ means that the special relay cannot be used in QCPU or LCPU.

ACPU Special Relay	Special Relay after Conversion	Special Relay for Modification	Name	Meaning	Details	Corresponding CPU
M9000	SM1000	-	Fuse blown	OFF : Normal ON : Module with blown fuse	• Turns on if there is at least one output module whose fuse has blown. • This relay remains on even after the condition returns to normal. • Output modules on remote I/O stations are also checked for blown fuse.	Qn(H) QnPH QnU*1
M9002	SM1002	-	I/O module verify error	OFF : Normal ON : Error	• This relay turns on if the status of the I/O module differs from that registered at power-on. • This relay remains on even after the system returns to normal. • I/O modules on remote I/O stations are also checked. • This relay is reset only when SD1116 to SD1123 are reset.	Qn(H) QnPH QnU*1 LCPU
M9005	SM1005	-	AC DOWN detection	OFF : AC DOWN not detected ON : AC DOWN detected	• This relay turns on if a momentary power failure within 20ms occurs during use of an AC power supply module. • This relay is reset when the CPU module is powered off and then on.	Qn(H) QnPH QnU*1
					• This relay turns on if a momentary power failure within 10ms occurs when using an AC power supply module. • This relay is reset when the CPU module is powered off and then on.	LCPU
					• This relay turns on if a momentary power failure within 10ms occurs during use of a DC power supply module. • This relay is reset when the CPU module is powered off and then on.	Qn(H) QnPH QnU*1 LCPU

ACPU Special Relay	Special Relay after Conversion	Special Relay for Modification	Name	Meaning	Details	Corres- ponding CPU
M9006	SM1006	-	Battery low	OFF : Normal ON : Battery low	- This relay turns on when the battery voltage drops to or below the specified. - It turns off when the battery voltage returns to normal.	Qn(H) QnPH QnU ^{*1} LCPU
M9007	SM1007	-	Battery low latch	OFF : Normal ON : Battery low	- This relay turns on when the battery voltage drops to or below the specified. - This relay remains on even after the battery voltage returns to normal.	
M9008	SM1008	SM1	Self-diagnosis error	OFF : No error ON : Error	This relay turns on if an error is detected by self-diagnostics.	
M9009	SM1009	SM62	Annunciator detection	OFF : No F number detected ON : F number detected	- This relay turns on when the OUT F or SET F instruction is executed. - It turns off when the SD1124 value is cleared to zero.	
M9011	SM1011	SM56	Operation error flag	OFF : No error ON : Error	- This relay turns on when an operation error occurs during execution of an application instruction. - This relay remains on even after the system returns to normal.	Qn(H) QnPH QnU ^{*1}
M9012	SM1012	SM700	Carry flag	OFF : Carry OFF ON : Carry ON	Carry flag used in application instruction.	Qn(H) QnPH
M9016	SM1016	×	Data memory clear flag	OFF : Ignored ON : Output cleared	When SM1016 turns on and remote RUN mode is activated from a computer, all the data memory including the latch range (except for the special relay and special register) is cleared.	
M9017	SM1017	×	Data memory clear flag	OFF : Ignored ON : Output cleared	When SM1017 turns on and remote RUN mode is activated from a computer, all the data memory that is not latched (except for the special relay and special register) is cleared.	
M9020	SM1020	-	User timing clock No.0		- This relay repeatedly turns on and off at the specified scan intervals. - When the CPU module is powered on or reset, this relay is set to on from off to start the clock. Set the intervals of on/off by DUTY instruction.  •n1: On scan interval •n2: Off scan interval When SM1020 to SM1024 are specified for the DUTY instruction in programs, if the CPU type is changed from the High Performance model QCPU or Process CPU to the Universal model QCPU or LCPU, they are replaced with SM420 to SM424. (For the Universal model QCPU and LCPU, SM1020 to SM1024 cannot be specified.)	Qn(H) QnPH QnU ^{*1} LCPU
M9021	SM1021	-	User timing clock No.1			
M9022	SM1022	-	User timing clock No.2			
M9023	SM1023	-	User timing clock No.3			
M9024	SM1024	-	User timing clock No.4			
M9025	SM1025	-	Clock data set request	OFF : Ignored ON : Set request present used	Clock data stored in SD1025 to SD1028 are written to the CPU module after the END instruction execution in the scan where SM1025 is turned on.	Qn(H) QnPH QnU ^{*1} LCPU
M9026	SM1026	-	Clock data error	OFF : No error ON : Error	This relay turns on if an error occurs in the clock data (SD1025 to SD1028), and is off while there is no error.	
M9028	SM1028	-	Clock data read request	OFF : Ignored ON : Read request	This relay is turned on to read clock data and store them as BCD values into SD1025 to SD1028.	
M9029	SM1029	×	Batch processing of data communications requests	OFF : Batch processing not conducted ON : Batch processing conducted	- When this relay is turned on in the program, all the data communication requests accepted during one scan are processed in the END processing of that scan. - The batch processing of data communication requests can be turned on or off during running. - The default is OFF (processed one at a time for each END processing in the order in which data communication requests are accepted).	Qn(H) QnPH

ACPU Special Relay	Special Relay after Conversion	Special Relay for Modification	Name	Meaning	Details	Corresponding CPU
M9030	SM1030	-	0.1 second clock		0.1-, 0.2-, 1-, and 2-second clocks are generated. - The relay turns on or off not for each scan, but also during a scan if the time has elapsed. - When the CPU module is powered on or reset, this relay is set to on from off to start the clock.	
M9031	SM1031	-	0.2 second clock			
M9032	SM1032	-	1 second clock			
M9033	SM1033	-	2 second clock			
M9034	SM1034	-	2n minute clock(1 minute clock)*2		- This relay repeatedly turns on and off according to the number of seconds specified in SD414. (Default: n = 30) - The relay turns on or off not for each scan, but also during a scan if the time has elapsed. - When the CPU module is powered on or reset, this relay is set to on from off to start the clock.	Qn(H) QnPH QnU*1 LCPU
M9036	SM1036	-	Always ON	ON	- This relay is used for initialization or as a dummy contact of application instructions in the program. - SM1036 and SM1037 are turned on or off regardless of the key switch setting on the front face of the CPU module. The states of SM1038 and SM1039 change depending on the key switch setting. When it is set to STOP, the relay is off. When it is set to other than STOP, SM1038 is on for one scan only and SM1039 is off for one scan only.	
M9037	SM1037	-	Always OFF	OFF		
M9038	SM1038	-	ON for 1 scan only after RUN	ON		
M9039	SM1039	-	RUN flag(After RUN, OFF for 1 scan only)	OFF		
M9040	SM1040	SM206	PAUSE enable coil	OFF : PAUSE disabled ON : PAUSE enabled	This relay is on when the CPU module is in PAUSE status or when the PAUSE contact is on.	Qn(H) QnPH
M9041	SM1041	SM204	PAUSE status contact	OFF : PAUSE not in effect ON : PAUSE in effect		Qn(H) QnPH QnU*1 LCPU
M9042	SM1042	SM203	STOP status contact	OFF : STOP not in effect ON : STOP in effect	This relay turns on when the RUN key switch or RUN/STOP switch is set to STOP.	
M9043	SM1043	SM805	Sampling trace completed	OFF : Sampling trace in progress ON : Sampling trace completed	This relay turns on after execution of the TRACE instruction and upon completion of sampling trace performed the number of times preset by the parameter. Reset when TRACER instruction is executed.	
M9045	SM1045	×	Watchdog timer (WDT) reset	OFF : Does not reset WDT ON : Resets WDT	If SM1045 is turned on, the watchdog timer is reset when the ZCOM instruction and batch processing of data communication requests are executed. (Use this when scan time exceeds 200ms.)	Qn(H) QnPH

ACPU Special Relay	Special Relay after Conversion	Special Relay for Modification	Name	Meaning	Details	Corres- ponding CPU
M9046	SM1046	SM802	Sampling trace	OFF : Trace not in progress ON : Trace in progress	This relay is on during execution of sampling trace.	Qn(H) QnPH QnU*1 LCPU
M9047	SM1047	SM801	Sampling trace preparations	OFF : Sampling trace suspended ON : Sampling trace started	Sampling trace is not executed unless SM1047 is turned ON. Sampling trace is cancelled when SM1047 turns off.	Qn(H) QnPH
M9049	SM1049	SM701	Switching the number of output characters	OFF : Output until NULL code encountered ON : 16 characters output	- When SM1049 is off, characters up to NULL (00_H) code are output. - When SM1049 is ON, ASCII codes of 16 characters are output.	
M9051	SM1051	×	CHG instruction execution disable	OFF : Enabled ON : Disable	- Switched ON to disable the CHG instruction. - Turn this on when requesting program transfer. It is automatically turned off upon completion of the transfer.	
M9052	SM1052	×	SEG instruction switch	OFF : 7SEG segment display ON : I/O partial refresh	When SM1052 is on, the SEG instruction is used as an I/O part refresh instruction. When SM1052 is off, the SEG instruction is used as a 7-SEG display instruction.	
M9056	SM1056	×	Main side P, I set request	OFF : Other than when P, I set being requested ON : P, I set being requested	While a program is running, upon completion of transfer of another program (for example, a subprogram when the main program is running), a P and I set request is turned on. This relay automatically turns off upon completion of P and I setting.	
M9057	SM1057	×	Sub side P, I set request	OFF : Other than when P, I set being requested ON : P, I set being requested		
M9058	SM1058	×	Main side P, I set completion	Momentarily ON at P, I set completion	This relay turns on for a moment upon completion of P and I setting, and immediately turns off.	
M9059	SM1059	×	Sub program P, I set completion	Momentarily ON at P, I set completion		
M9060	SM1060	×	Sub program 2 P, I set request	OFF : Other than when P, I set being requested ON : P, I set being requested	While a program is running, upon completion of transfer of another program (for example, a subprogram when the main program is running), a P and I set request is turned on. This relay automatically turns off upon completion of P and I setting.	
M9061	SM1061	×	Sub program 3 P, I set request	OFF : Other than when P, I set being requested ON : P, I set being requested		
M9070	SM1070	×	A8UPU/ A8PUJrequired search time*3	OFF : Read time not shortened ON : Read time shortened	When this is turned on, the search time in the A8UPU/A8PUJ can be shortened. (In this case, the scan time is extended by 10%.)	
M9084	SM1084	×	Error check	OFF : Error check executed ON : No error check	This relay sets whether or not to check the following errors at the time of the END instruction processing (for setting of the END instruction processing time). - Check for fuse blown. - Check of battery - Collation check of I/O module	
M9091	SM1091	×	Operation error details flag	OFF : No error ON : Error	- This relay turns on when the detail factor of the operation error is stored into SD1091. - This relay remains on even after the condition returns to normal.	

ACPU Special Relay	Special Relay after Conversion	Special Relay for Modification	Name	Meaning	Details	Corres- ponding CPU
M9100	SM1100	SM320	Presence/ absence of SFC program	OFF : SFC programs not used ON : SFC programs used	This relay is on when an SFC program has been registered, and is off when no program is registered.	Qn(H) QnPH
M9101	SM1101	SM321	Start/stop SFC program	OFF : SFC programs stop ON : SFC programs start	• The same value as in SM1100 is set as the initial value. (This relay turns on when an SFC program is registered.) • This relay is turned off to stop SFC program execution. • This relay is turned on to resume the SFC program execution.	
M9102	SM1102	SM322	SFC program start status	OFF : Initial start ON : Resume start	In the SFC setting of the PLC parameter dialog box, Initial start is set for the SFC program start mode. • At initial start: OFF • At continue start: ON	
M9103	SM1103	SM323	Presence/ absence of continuous transition	OFF : Continuous transition not effective ON : Continuous transition effective	Set whether to enable or disable continuous transition for the blocks where "continuous transition bit" of the SFC information device is not set.	
M9104	SM1104	SM324	Continuous transition suspension flag	OFF : When transition is completed ON : When no transition	• This relay is off during operation in the continuous transition mode or during continuous transition, and is on while continuous transition is not performed. • This relay is always on while the CPU module is operating not in the continuous transition mode.	
M9108	SM1108	SM90	Step transition monitoring timer start (equivalent of SD90)	OFF : Monitoring timer reset ON : Monitoring timer reset start	The relay turns on when measurement by the step transition monitoring timer is started. The step transition monitoring timer is reset when the relay turns off.	
M9109	SM1109	SM91	Step transition monitoring timer start (equivalent of SD91)			
M9110	SM1110	SM92	Step transition monitoring timer start (equivalent of SD92)			
M9111	SM1111	SM93	Step transition monitoring timer start (equivalent of SD93)			
M9112	SM1112	SM94	Step transition monitoring timer start (equivalent of SD94)			
M9113	SM1113	SM95	Step transition monitoring timer start (equivalent of SD95)			
M9114	SM1114	SM96	Step transition monitoring timer start (equivalent of SD96)			
M9196	SM1196	SM325	Operation output at block stop	OFF : Coil output OFF ON : Coil output ON	Selects the operation output when block stop is executed. • On: Retains the on or off status of the coil used in the operation output of the step, which was being executed at the time of block stop. • Off::Turns off all the coil outputs. (Operation output by the SET instruction is retained regardless of the on/off status of SM1196.)	

ACPU Special Relay	Special Relay after Conversion	Special Relay for Modification	Name	Meaning	Details	Corres- ponding CPU															
M9197	SM1197	×	Switch between blown fuse and I/O module verification error display	<table><tr><td>SM1197</td><td>SM1198</td><td>I/O numbers to be displayed</td></tr><tr><td>OFF</td><td>OFF</td><td>X/Y0 to 7F0</td></tr><tr><td>ON</td><td>OFF</td><td>X/Y800 to FF0</td></tr><tr><td>OFF</td><td>ON</td><td>X/Y1000 to 17F0</td></tr><tr><td>ON</td><td>ON</td><td>X/Y1800 to 1FF0</td></tr></table>	SM1197	SM1198	I/O numbers to be displayed	OFF	OFF	X/Y0 to 7F0	ON	OFF	X/Y800 to FF0	OFF	ON	X/Y1000 to 17F0	ON	ON	X/Y1800 to 1FF0	Switches I/O numbers between the fuse-blown module registers (SD1100 to SD1107) and I/O module verify error registers (SD1116 to SD1123) according to the on/off combination of SM1197 and SM1198.	Qn(H) QnPH
SM1197	SM1198	I/O numbers to be displayed																			
OFF	OFF	X/Y0 to 7F0																			
ON	OFF	X/Y800 to FF0																			
OFF	ON	X/Y1000 to 17F0																			
ON	ON	X/Y1800 to 1FF0																			
M9198	SM1198	×																			
M9199	SM1199	×	Data recovery of online sampling trace/status latch	OFF : Data recovery disabled ON : Data recovery enabled	- Recovers the setting data stored in the CPU module at restart when sampling trace/status latch is executed. - Turn this on to re-execute the sampling trace or status latch. (Rewriting data using the programming tool is not required.)																

- *1 The relevant modules are as follows:
- The Universal model QCPU whose serial number (first five digits) is "10102" or later.
 - Q00UJCPU, Q00UCPU, Q01UCPU
- *2 1 minute clock indicates the name of the special relay (M9034) of the ACP.
- *3 The A8UPU/A8PUJ is not available for the QCPU/LCPU.

(10) Built-in Ethernet port QCPU and built-in Ethernet function

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU M9□□□	Corresponding CPU
SM1270	Time setting function (SNTP client) execution	OFF : No time setting function (SNTP client) execution ON : Time setting function (SNTP client) execution	This relay is turned on to perform the time setting function (SNTP client). (Turns on only when "Use" has been set for the time setting function in the time setting parameter.)	U	New	QnU*1 LCPU
SM1273	Remote password mismatch count clear	OFF : Normal ON : Clear	This relay is turned on to clear the accumulated number of mismatched remote password entries (SD979 to SD999).			QnU*2
SM1292	IP address storage area write request	OFF : Ignored ON : Write request	The IP address setting stored in SD1292 to SD1297 are written to the IP address storage area (flash ROM) of the CPU module when the END instruction is executed in the scan where this relay is turned on.			
SM1293	IP address storage area write completion	OFF : Not completed ON : Completed	• This relay turns on when writing to the IP address storage area (flash ROM) is completed. • This relay turns off when the END instruction is executed in the scan where SM1292 is turned off.	S (Status change)		
SM1294	IP address storage area write error	OFF : Normal ON : Error	• This relay turns on when writing to the IP address storage area (flash ROM) fails. • This relay turns off when the END instruction is executed in the scan where SM1292 is turned off.			
SM1295	IP address storage area clear request	OFF : Ignored ON : Clear request	The IP address storage area (flash ROM) is cleared when the END instruction is executed in the scan where this relay is turned on.	U		
SM1296	IP address storage area clear completion	OFF : Not completed ON : Completed	• This relay turns on when clearing the IP address storage area (flash ROM) is completed. • This relay turns off when the END instruction is executed in the scan where SM1295 is turned off.	S (Status change)		
SM1297	IP address storage area clear error	OFF : Normal ON : Error	• This relay turns on when clearing the IP address storage area (flash ROM) fails. • This relay turns off when the END instruction is executed in the scan where SM1295 is turned off.			

*1 This applies to the Built-in Ethernet port QCPU.

*2 This applies to the built-in Ethernet port QCPU whose first five digits of serial No. is "11082" or later.

(11) Process control instruction

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU M9□□□	Corresponding CPU
SM1500	Hold mode	OFF : No-hold	Specifies whether or not to hold the output value when a range over occurs for the S.IN instruction range check.	U	New	QnPH QnPRH
SM1501		ON : Hold	Specifies whether or not the output value is held when a range over occurs for the S.OUT instruction range check.			

(12) Redundant system (host system CPU information *1)

The special relay (SM1510 to SM1599) is valid only for redundant systems. All bits are set to off for stand-alone systems.

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU M9□□□	Corres- ponding CPU												
SM1510	Operation mode	OFF : Redundant system backup mode, stand-alone system ON : Redundant system separate mode	This relay is on while the system is operating in the separate mode.	S (Every END processing)	New	QnPRH												
SM1511	System A identification flag	• Distinguishes between system A and system B. • The flag status does not change even if the tracking cable is disconnected. <table><tr><td></td><td>System A</td><td>System B</td><td>If TRK. CABLE ERR. (error code: 6210) occurred (Unknown)</td></tr><tr><td>SM1511</td><td>ON</td><td>OFF</td><td>OFF</td></tr><tr><td>SM1512</td><td>OFF</td><td>ON</td><td>OFF</td></tr></table>					System A	System B	If TRK. CABLE ERR. (error code: 6210) occurred (Unknown)	SM1511	ON	OFF	OFF	SM1512	OFF	ON	OFF	S (Initial)
	System A			System B			If TRK. CABLE ERR. (error code: 6210) occurred (Unknown)											
SM1511	ON			OFF			OFF											
SM1512	OFF	ON	OFF															
SM1512	System B identification flag																	
SM1513	Debug mode status flag	OFF : Not in debug mode ON : Debug mode	This relay is on while the system is operating in the debug mode.															

Number	Name	Meaning	Explanation			Set by (When Set)	Corres- ponding ACPU M9□□□	Corres- ponding CPU														
SM1515	Control system judgment flag	• Indicates operation system status. • The flag status does not change even if the tracking cable is disconnected. <table><tr><td></td><td>Control system</td><td>Standby system</td><td colspan="2">If TRK. CABLE ERR. (error code: 6210) occurred (Unknown)</td></tr><tr><td>SM1515</td><td>ON</td><td>OFF</td><td colspan="2">OFF</td></tr><tr><td>SM1516</td><td>OFF</td><td>ON</td><td colspan="2">OFF</td></tr></table>		Control system	Standby system	If TRK. CABLE ERR. (error code: 6210) occurred (Unknown)		SM1515	ON	OFF	OFF		SM1516	OFF	ON	OFF		S (Status change)				
	Control system		Standby system	If TRK. CABLE ERR. (error code: 6210) occurred (Unknown)																		
SM1515	ON		OFF	OFF																		
SM1516	OFF		ON	OFF																		
SM1516	Standby system judgment flag																					
SM1517	CPU module startup status	OFF : Power supply on startup ON : Operation system switch start up	Turns on when the CPU module is started up by the system switching (switching from the standby system to the control system). Remains OFF when the standby system is switched to the control system by a power-ON startup.			S (Status change)																
SM1518	Standby system to control system switching status flag	ON 	• This relay turns on during one scan after the standby system was switched to the control system. • This relay can be used only in a scan execution type program.			S (Every END processing)																
SM1519	Previous Control System Identification Flag	ON 	When the previous control system is System B, this relay turns on during one scan in System A, following the RUN state after both Systems A and B were simultaneously turned on or were reset.																			
SM1520	Data tracking transfer trigger specification	OFF : No trigger ON : Trigger	SM1520	Block 1	• When data is transferred based on the tracking setting of the Redundant parameter dialog box, the target block is specified as trigger. • When "Do auto forward Tracking block No.1" is selected for the tracking setting, SM1520 is turned on by the system at power-on or when the system is switched from STOP to RUN. In other cases, SM1520 to SM1583 are turned on by the user.	S (initial)/U			New	QnPRH												
SM1521			SM1521	Block 2																		
SM1522			SM1522	Block 3																		
SM1523			SM1523	Block 4																		
SM1524			SM1524	Block 5																		
SM1525			SM1525	Block 6																		
SM1526			SM1526	Block 7																		
SM1527			SM1527	Block 8																		
SM1528			SM1528	Block 9																		
SM1529			SM1529	Block 10																		
SM1530			SM1530	Block 11																		
SM1531			SM1531	Block 12																		
SM1532			SM1532	Block 13																		
SM1533			SM1533	Block 14																		
SM1534			SM1534	Block 15																		
SM1535			SM1535	Block 16																		
SM1536			SM1536	Block 17																		
SM1537			SM1537	Block 18																		
SM1538			SM1538	Block 19																		
SM1539			SM1539	Block 20																		
SM1540			SM1540	Block 21																		
SM1541			SM1541	Block 22																		
SM1542			SM1542	Block 23																		
SM1543			SM1543	Block 24																		
SM1544			SM1544	Block 25																		
SM1545			SM1545	Block 26																		
SM1546			SM1546	Block 27																		
SM1547			SM1547	Block 28																		
SM1548			SM1548	Block 29																		
SM1549			SM1549	Block 30																		
SM1550			SM1550	Block 31																		
SM1551			SM1551	Block 32																		

Number	Name	Meaning	Explanation			Set by (When Set)	Corres- ponding ACPU M9□□□	Corres- ponding CPU
SM1552	Data tracking transfer trigger specification	OFF : No trigger ON : Trigger	SM1552	Block 33	- When data is transferred based on the tracking setting of the Redundant parameter dialog box, the target block is specified as trigger. - When "Do auto forward Tracking block No.1" is selected for the tracking setting, SM1520 is turned on by the system at power-on or when the system is switched from STOP to RUN.In other cases, SM1520 to SM1583 are turned on by the user.	S (initial)/U	New	QnPRH
SM1553			SM1553	Block 34				
SM1554			SM1554	Block 35				
SM1555			SM1555	Block 36				
SM1556			SM1556	Block 37				
SM1557			SM1557	Block 38				
SM1558			SM1558	Block 39				
SM1559			SM1559	Block 40				
SM1560			SM1560	Block 41				
SM1561			SM1561	Block 42				
SM1562			SM1562	Block 43				
SM1563			SM1563	Block 44				
SM1564			SM1564	Block 45				
SM1565			SM1565	Block 46				
SM1566			SM1566	Block 47				
SM1567			SM1567	Block 48				
SM1568			SM1568	Block 49				
SM1569			SM1569	Block 50				
SM1570			SM1570	Block 51				
SM1571			SM1571	Block 52				
SM1572			SM1572	Block 53				
SM1573			SM1573	Block 54				
SM1574			SM1574	Block 55				
SM1575			SM1575	Block 56				
SM1576			SM1576	Block 57				
SM1577			SM1577	Block 58				
SM1578			SM1578	Block 59				
SM1579			SM1579	Block 60				
SM1581	SM1581	Block 61						
SM1582	SM1582	Block 62						
SM1583	SM1583	Block 63						
SM1590	System switching enable/disable flag from network module	OFF : System switching request issuing module absent ON : System switching request issuing module present	Turns on when a system switching request is issued from the network module. The module No. that issued system switching can be checked by SD1590. Turns off when all bits of SD1590 are off.			S (Every END processing)	New	QnPRH
SM1591	Standby system error detection disable flag at system switching	ON : Error is not detected by new standby system at system switching OFF : Error is detected by new standby system at system switching	This flag is used when switching the system in any of the following sources to determine whether to detect "STANDBY" (error code 6210) in the new standby system: [Reason(s) for system switching] - System switching with a programming tool - System switching using dedicated instruction - System switching by the intelligent function module			U		
SM1592	Enable/disable user system switching	OFF : Disable user system switching ON : Enable user system switching	This relay stores whether to enable manual switching using a programming tool or the system switching instruction (SP.CONTSW).					

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU M9□□□	Corres- ponding CPU
SM1593	Setting to access extension base unit of standby system CPU	OFF : Error ON : Ignored	This relay sets the behavior of the system after the standby CPU in the separate mode accessed the buffer memory of an intelligent function module mounted on an extension base unit. • OFF: "OPERATION ERROR" (error code: 4112) is returned. • ON: No processing	U	New	QnPRH
SM1595	Memory copy to other system start flag	OFF : Start memory copy ON : No memory copy initiated	When SM1595 is turned on from off, memory copying from the control system to the standby system starts. Note that memory copy does not start even after SM1595 was turned on from off if the I/O No. of the copy destination (standby system CPU module: 3D1 _H) is not stored in SD1595.			
SM1596	Memory copy to other system status flag	OFF : Memory copy not executed ON : Memory copy executed	• This relay is on during memory copy from the control system to the standby system. • This relay turns off when memory copy is complete.	S (Starting to copy/finish)		
SM1597	Memory copy to other system completion flag	OFF : Memory copy not completed ON : Memory copy completed	• This relay turns upon completion of memory copy from the control system to the standby system.	S (finish)/U		
SM1598	Copy contents of standard ROM during memory copy	OFF : Copy standard ROM data ON : Standard ROM data is not copied	• If set to on by user, the standard ROM data is not copied to the other system while memory copy is executing.	U		

*1 The information of the host CPU module is stored.

(13) Redundant system (other system CPU information^{*1})

The special relay (SM1600 to SM1649) is valid when the redundant system is in backup mode and is invalid in separate mode. All bits are set to off for stand-alone systems.

Number	Name	Meaning	Explanation	Set by (When Set)	Corresponding Host SM□□ ^{*2}	Corresponding CPU
SM1600	Other system error flag	OFF : No error ON : Error	- This relay turns on if an error is detected by error check for redundant system. (This relay turns on when any of the SD1600 bits turns on.) - This relay turns off when an error is cleared.	S (Every END processing)	-	QnPRH
SM1610	Other system diagnostics error	OFF : No error ON : Error	- This relay turns on if a diagnostic error occurs in the CPU module in the other system. (Also turns off when an annunciator turns on and when an error is detected by the CHK instruction.) - The SM0 status for the CPU module in the other system is reflected.		SM0	
SM1611	Other systems self diagnostics error.	OFF : No self diagnostics error occurred ON : Self diagnostics error occurred	- This relay turns on if a self-diagnostics error occurred in the CPU module in the other system. (Excluding error detections by an annunciator and the CHK instruction.) - The SM1 status for the CPU module in the other system is reflected.		SM1	
SM1615	Other system common error information	OFF : No common error information present ON : Common error information present	- This relay turns on when there is error common information data for an error occurred in the CPU module in the other system. - The SM5 status for the CPU module in the other system is reflected.		SM5	
SM1626	Error individual information for other systems	OFF : No individual error information present ON : Individual error information present	- This relay turns on when there is error individual information for an error occurred in the CPU module in the other system. - The SM16 status for the CPU module in the other system is reflected.		SM16	
SM1649	Standby system cancel error flag	OFF to ON: Cancels error of standby system	This relay is turned on from off to clear a continuation error occurred in the standby system. Use SD1649 to specify the error code of the error to be canceled.	U	-	

*1 Stores other system CPU diagnostic information and system information.

*2 This shows the special relay (SM□□) for the host system CPU.

(14) Redundant system (tracking information)

The special relay (SM1700 to SM1799) is valid when the redundant system is in backup mode or in separate mode. All bits are set to off for stand-alone systems.

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU M9□□□	Corres- ponding CPU
SM1700	Transfer trigger completion flag	OFF : Transfer not completed ON : Transfer completed	This relay remains on for one scan upon completion of a transfer for any of the blocks 1 to 64.	S (Status change)	New	QnPRH
SM1709	Manual system switching disable/enable setting during online program change redundant tracking	ON : Manual system switching enabled (Disable canceled) OFF : Manual system switching disabled	- This relay is turned from off to on to enable the user to switch a system during online program change for redundancy. After the manual system switching disable status is canceled, the system automatically turns off SM1709. - A system can be switched even a online program change for redundancy is being performed and regardless of the status of this relay, if the reason for the switching is any of the following: - Power-off - Reset - Hardware failure - CPU stop error - The system switching disable status can also be canceled by this relay during the following states. - Multiple-block online program change redundant tracking execution status - File batch online program change redundant tracking execution status	S (Request)/U		
SM1710	Transfer tracking data during online program change enable flag	OFF : No device tracking ON : Transfer device memory	- This relay specifies whether to execute a tracking transfer for the following control data during online program change for redundancy. - Device memory (Including SMs and SDs that automatically execute a tracking transfer) - PIDINIT information, S.PIDINIT information, SFC information - SM1710 can be also used to specify whether to enable a tracking transfer while multiple-block online program change redundant tracking and while file batch online program change redundant tracking. - SM1710 is transferred from the control system to the standby system by tracking transfer.	U		

Number	Name	Meaning	Explanation			Set by (When Set)	Corres- ponding ACPU M9□□□	Corres- ponding CPU
SM1712	Transfer trigger completion flag	OFF : Transfer uncompleted ON : Transfer completed	SM1712	Block 1	This relay turns on only during one scan upon completion of a transfer for the relevant block.	S (Status change)	New	QnPRH
SM1713			SM1713	Block 2				
SM1714			SM1714	Block 3				
SM1715			SM1715	Block 4				
SM1716			SM1716	Block 5				
SM1717			SM1717	Block 6				
SM1718			SM1718	Block 7				
SM1719			SM1719	Block 8				
SM1720			SM1720	Block 9				
SM1721			SM1721	Block 10				
SM1722			SM1722	Block 11				
SM1723			SM1723	Block 12				
SM1724			SM1724	Block 13				
SM1725			SM1725	Block 14				
SM1726			SM1726	Block 15				
SM1727			SM1727	Block 16				
SM1728			SM1728	Block 17				
SM1729			SM1729	Block 18				
SM1730			SM1730	Block 19				
SM1731			SM1731	Block 20				
SM1732			SM1732	Block 21				
SM1733			SM1733	Block 22				
SM1734			SM1734	Block 23				
SM1735			SM1735	Block 24				
SM1736			SM1736	Block 25				
SM1737			SM1737	Block 26				
SM1738			SM1738	Block 27				
SM1739			SM1739	Block 28				
SM1740			SM1740	Block 29				
SM1741			SM1741	Block 30				
SM1742			SM1742	Block 31				
SM1743			SM1743	Block 32				
SM1744			SM1744	Block 33				
SM1745			SM1745	Block 34				
SM1746			SM1746	Block 35				
SM1747			SM1747	Block 36				
SM1748			SM1748	Block 37				
SM1749			SM1749	Block 38				
SM1750			SM1750	Block 39				
SM1751			SM1751	Block 40				
SM1752			SM1752	Block 41				
SM1753			SM1753	Block 42				
SM1754			SM1754	Block 43				
SM1755			SM1755	Block 44				
SM1756			SM1756	Block 45				
SM1757			SM1757	Block 46				
SM1758			SM1758	Block 47				
SM1759			SM1759	Block 48				
SM1760			SM1760	Block 49				

Number	Name	Meaning	Explanation			Set by (When Set)	Corres- ponding ACPU M9□□□	Corres- ponding CPU
SM1761	Transfer trigger completion flag	OFF : Transfer uncompleted ON : Transfer completed	SM1761	Block 50	This relay turns on only for one scan upon completion of a transfer for the relevant block.	S (Status change)	New	QnPRH
SM1762			SM1762	Block 51				
SM1763			SM1763	Block 52				
SM1764			SM1764	Block 53				
SM1765			SM1765	Block 54				
SM1766			SM1766	Block 55				
SM1767			SM1767	Block 56				
SM1768			SM1768	Block 57				
SM1769			SM1769	Block 58				
SM1770			SM1770	Block 59				
SM1771			SM1771	Block 60				
SM1772			SM1772	Block 61				
SM1773			SM1773	Block 62				
SM1774			SM1774	Block 63				
SM1775			SM1775	Block 64				

(15) Redundant power supply module information

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU M9□□□	Corresponding CPU
SM1780	Power supply off detection flag	OFF : No redundant power supply module with input power OFF detected ON : Redundant power supply module with input power OFF detected	• Turns ON when one or more redundant power supply modules with input power OFF are detected. • Turns on if any of SD1780 bits is on. • Turns off if all bits of SD1780 are off. • This relay turns off when the main base unit is not the redundant main base unit (Q38RB). • When the multiple CPU system is configured, the flags are stored only to the CPU No.1.	S (Every END processing)	New	Qn(H)* ³ QnPH* ³ QnPRH QnU* ⁴
SM1781	Power supply failure detection flag	OFF : No faulty redundant power supply module detected ON : Faulty redundant power supply module detected	• Turns ON when one or more faulty redundant power supply modules are detected. • Turns on if any of SD1781 bits is on. • Turns off if all bits of SD1781 are off. • This relay turns off when the main base unit is not the redundant main base unit (Q38RB). • When the multiple CPU system is configured, the flags are stored only to the CPU No.1.			
SM1782	Momentary power failure detection flag for power supply 1 * ¹	OFF : No momentary power failure detected ON : Momentary power failure detected	• Turns ON when a momentary power failure of the input power supply to the power supply 1 or 2 is detected one or more times. After turning on, this relay remains on even if the power supply recovers from the momentary power failure. • Turns off the flags (SM1782 and SM1783) of the power supply 1 and 2 when the CPU module starts. • When the input power to one of the redundant power supply modules turns OFF the corresponding flag turns OFF. • This relay turns off when the main base unit is not the redundant main base unit (Q38RB). • When the multiple CPU system is configured, the flags are stored only to the CPU No.1.			
SM1783	Momentary power failure detection flag for power supply 2 * ²					

*1: The "power supply 1" indicates the redundant power supply module mounted on the POWER 1 slot of the redundant base unit (Q38RB/Q68RB/Q65WRB).

*2: The "power supply 2" indicates the redundant power supply module mounted on the POWER 2 slot of the redundant base unit (Q38RB/Q68RB/Q65WRB).

*3: The module whose first 5 digits of serial No. is "04012" or later.

However, for the multiple CPU system configuration, this applies to all CPU modules whose first 5 digits of serial No. are "07032" or later.

*4: The module whose first 5 digits of serial No. is "10042" or later.

(16) Built-in I/O function

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU M9□□□	Corresponding CPU
SM1840	Axis 1 busy	OFF : Not busy ON : Busy	• This relay turns on when positioning control, OPR control, JOG operation, or absolute position restoration is started. This relay turns off when each control is completed. In positioning control, this relay turns off when the axis 1 decelerates and stops, and then "dwell time" elapsed. (This relay remains on while positioning control is being performed.) • This relay turns off when each control is ended due to such as an error or stop operation.	S (Every END processing)	New	LCPU
SM1841	Axis 1 positioning completion	OFF : Not completed ON : Completed	• This relay turns on when OPR control, position control, or absolute position restoration is completed. • This relay turns off when OPR control, positioning control, absolute position restoration, or JOG operation is started. • This relay remains off when JOG operation is completed. • This relay remains off when position control is stopped.	S (Instruction execution/ Status change)		
SM1842	Axis 1 OPR request	OFF : Machine OPR control completed ON : Machine OPR control started	• This relay turns on when the CPU module is powered on, is reset, or is set from STOP to RUN; or the drive unit ready signal turns off; or machine OPR control is started. • This relay turns off when machine OPR control is completed.	S (Every END processing)		
SM1843	Axis 1 OPR completion	OFF : Not completed ON : Completed	• This relay turns on when machine OPR control is completed. • This relay turns off when OPR control, positioning control, absolute position restoration, or JOG operation is started; or the CPU module is set from STOP to RUN; or the drive unit ready signal turns off.	S (Instruction execution/ Status change)		
SM1844	Axis 1 speed 0	OFF : Operating at speed other than 0 ON : Operating at speed 0	• This relay turns on when JOG operation or speed control in speed/position switching control set at a speed of "0" is started. • This relay turns on when speed is changed with a new speed value of "0", and turns off when speed is changed with a new speed value other than "0". • This relay turns off when SM1840 turns off.	S (Every END processing)		
SM1845	Axis 1 error	OFF : No error ON : Error	• This relay turns on if an error occurs. • The present error can be checked by SD1845. • This relay is turned off by turning on SM1850.			
SM1846	Axis 1 warning	OFF : No warning ON : Warning	• This relay turns on if a warning occurs. • The present warning can be checked by SD1846. • This relay is turned off by turning on SM1850.			
SM1847	Axis 1 start in busy status	OFF : No start attempted in busy status ON : Start attempted in busy status	• This relay turns on when positioning control, OPR control, JOG operation, or absolute position restoration is attempted while the axis 1 is in the busy status. The executed start instruction will be ignored. • This relay is reset by the user.	S (Instruction execution) /U		

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU M9□□□	Corresponding CPU
SM1848	Axis 1 start instruction	OFF : Not executed ON : Being executed	- This relay turns on when positioning control by the start instruction (IPPSTRT1(P), IPDSTRT1(P), IPSIMUL(P), IPABRST1), JOG operation by the JOG start instruction (IPJOG1), or OPR control by the OPR start instruction (IOPPR1(P)) is started. - This relay turns off when positioning control, OPR control, or JOG operation is completed.	S (Instruction execution/ Status change)		
SM1850	Axis 1 error reset	OFF → ON: Resets the Axis 1 error. OFF : Clears the reset status.	- Turning on this relay will turn off SM1845 and SM1846 and will clear the SD1845 and SD1846 values to "0". - Even if this relay is turned on, SM1845 will not turn off and the SD1845 value will not be cleared to "0" until SM1840 turns off.	U		
SM1851	Axis 1 OPR request off	OFF → ON: Axis 1 OPR request OFF : Cleared	Turning on this relay will forcibly turn off SM1842.			
SM1852	Axis 1 speed/ position switching	OFF : Disabled ON : Enabled	This relay stores whether to enable switching from speed control to position control in speed/position switching control.			
SM1860	Axis 2 busy	OFF : Not busy ON : Busy	- This relay turns on when positioning control, OPR control, JOG operation, or absolute position restoration is started. This relay turns off when each control is completed. In positioning control, this relay turns off when the axis 2 decelerates and stops, and then "dwell time" elapsed. (This relay remains on while positioning control is being performed.) - This relay turns off when each control is ended due to such as an error or stop operation.	S (Every END processing)	New	LCPU
SM1861	Axis 2 positioning completion	OFF : Not completed ON : Completed	- This relay turns on when OPR control, position control, or absolute position restoration is completed. - This relay turns off when OPR control, positioning control, absolute position restoration, or JOG operation is started. - This relay remains off when JOG operation is completed. - This relay remains off when position control is stopped.	S (Instruction execution/ Status change)		
SM1862	Axis 2 OPR request	OFF : Machine OPR control completed ON : Machine OPR control started	- This relay turns on when the CPU module is powered on, is reset, or is set from STOP to RUN; or the drive unit ready signal turns off; or machine OPR control is started. - This relay turns off when machine OPR control is completed.	S (Every END processing)		

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU M9□□□	Corresponding CPU		
SM1863	Axis 2 OPR completion	OFF : Not completed ON : Completed	• This relay turns on when machine OPR control is completed. • This relay turns off when OPR control, positioning control, absolute position restoration, or JOG operation is started; or the CPU module is set from STOP to RUN; or the drive unit ready signal turns off.	S (Instruction execution/ Status change)				
SM1864	Axis 2 speed 0	OFF : Operating at speed other than 0 ON : Operating at speed 0	• This relay turns on when JOG operation or speed control in speed/position switching control set at a speed of "0" is started. • This relay turns on when speed is changed with a new speed value of "0", and turns off when speed is changed with a new speed value other than "0". • This relay turns off when SM1860 turns off.	S (Every END processing)				
SM1865	Axis 2 error	OFF : No error ON : Error	• This relay turns on if an error occurs. • The present error can be checked by SD1865. • This relay is turned off by turning on SM1870.					
SM1866	Axis 2 warning	OFF : No warning ON : Warning	• This relay turns on if a warning occurs. • The present warning can be checked by SD1866. • This relay is turned off by turning on SM1870.					
SM1867	Axis 2 start in busy status	OFF : No start attempted in busy status ON : Start attempted in busy status	• This relay turns on when positioning control, OPR control, JOG operation, or absolute position restoration is attempted while the axis 2 is in the busy status. The executed start instruction will be ignored. • This relay is reset by the user.	S (Instruction execution) /U			New	LCPU
SM1868	Axis 2 start instruction	OFF : Not executed ON : Being executed	• This relay turns on when positioning control by the start instruction (IPPSTR2(P), IPDSTR2(P), IPSIMUL(P), IPABRST2), JOG operation by the JOG start instruction (IPJOG2), or OPR control by the OPR start instruction (IOPR2(P)) is started. • This relay turns off when positioning control, OPR control, or JOG operation is completed.	S (Instruction execution/ Status change)				
SM1870	Axis 2 error reset	OFF → ON: Resets the Axis 2 error. OFF : Clears the reset status.	• Turning on this relay will turn off SM1865 and SM1866 and will clear the SD1865 and SD1866 values to "0". • Even if this relay is turned on, SM1865 will not turn off and the SD1865 value will not be cleared to "0" until SM1860 turns off.	U				
SM1871	Axis 2 OPR request off	OFF→ON: Axis 2 OPR request OFF : Cleared	Turning on this relay will forcibly turn off SM1862.					
SM1872	Axis 2 speed/ position switching	OFF : Disabled ON : Enabled	This relay stores whether to enable switching from speed control to position control in speed/position switching control.					

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU M9□□□	Corresponding CPU
SM1880	CH1 counter value greater (No.1)	OFF : Coincidence point (No.1) or smaller ON : Greater than coincidence point (No.1)	• This relay turns on when "current value of CH1 > coincidence output No.1 point setting value" is met. • This relay turns off when "current value of CH1 ≤ coincidence output No.1 point setting value" is met.	S (Every END processing)	New	LCPU
SM1881	CH1 counter value coincidence (No.1)	OFF : Not detected ON : Detected	• This relay turns on when "current value of CH1 = coincidence output No.1 point setting value" is met. • This relay is turned off by turning on CH1 coincidence signal No.1 reset command.	S (Status change/ Every END processing)		
SM1882	CH1 counter value smaller (No.1)	OFF : Coincidence point (No.1) or greater ON : Smaller than coincidence point (No.1)	• This relay turns on when "current value of CH1 < coincidence output No.1 point setting value" is met. • This relay turns off when "current value of CH1 ≥ coincidence output No.1 point setting value" is met.	S (Every END processing)		
SM1883	CH1 counter value greater (No.2)	OFF : Coincidence point (No.2) or smaller ON : Greater than coincidence point (No.2)	• This relay turns on when "current value of CH1 > coincidence output No.2 point setting value" is met. • This relay turns off when "current value of CH1 ≤ coincidence output No.2 point setting value" is met.			
SM1884	CH1 counter value coincidence (No.2)	OFF : Not detected ON : Detected	• This relay turns on when "current value of CH1 = coincidence output No.2 point setting value" is met. • This relay is turned off by turning on CH1 coincidence signal No.2 reset command.	S (Status change/ Every END processing)		
SM1885	CH1 counter value smaller (No.2)	OFF : Coincidence point (No.2) or greater ON : Smaller than coincidence point (No.2)	• This relay turns on when "current value of CH1 < coincidence output No.2 point setting value" is met. • This relay turns off when "current value of CH1 ≥ coincidence output No.2 point setting value" is met.	S (Every END processing)		
SM1886	CH1 external preset (phase Z) request detection	OFF : Not detected ON : Detected	• This relay turns on when a preset request by phase Z (preset) terminal of CH1 is detected. • This relay is turned off by turning on CH1 external preset (phase Z) request detection clear command.			
SM1887	CH1 error	OFF : No error ON : Error	• This relay turns on if the CH1 error occurs. • This relay turns off when an error cause is removed and CH1 error reset command is turned on.			
SM1888	CH1 warning	OFF : No warning ON : Warning	• This relay turns on if a warning occurs in CH1. • This relay turns off when a warning cause is removed and CH1 error reset command is turned on.			

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU M9□□□	Corresponding CPU
SM1890	CH1 coincidence signal No.1 reset command	Resets CH1 counter value coincidence No.1.	- This relay is turned on to reset CH1 counter value coincidence No.1. - The command is valid while this relay is on. - The on time must be held for at least 2ms.	U	New	LCPU
SM1891	CH1 coincidence signal No.2 reset command	Resets CH1 counter value coincidence No.2.	- This relay is turned on to reset CH1 counter value coincidence No.2. - The command is valid while this relay is on. - The on time must be held for at least 2ms.			
SM1892	CH1 coincidence output enable command	Controls outputs from CH1 coincidence output No.1 and No.2 terminals.	- This relay is turned on to perform coincidence output from CH1 coincidence output No.1 and CH1 coincidence output No.2 terminals. - The command is valid while this relay is on.			
SM1893	CH1 preset command	Presets the counter value.	- This relay is turned on to preset the counter value. - The command is valid at the rise of this relay (off → on). - The on and off time must be held for at least 2ms.			
SM1894	CH1 count down command	Counts down pulses.	- This relay is turned on to count down pulses. - The command is valid while the Pulse input mode is either 1-phase multiple of n or 1-phase multiple of n (A phase only). - The command is valid while this relay is on.			
SM1895	CH1 count enable command	Starts counting.	- This relay is turned on to start counting. - The command is valid while this relay is on.			
SM1896	CH1 counter function selection start command	Starts the selected counter function.	- This relay is turned on to start the selected counter function. - When the count disabling function is selected, the command is valid while this relay is on. - When the latch counter function or the sampling counter function is selected, the command is valid at the rise of this relay (off → on). The on time must be held for at least 2ms. - When the count disabling/preset function or the latch counter/preset function is selected, the command is invalid.			
SM1897	CH1 external preset (phase Z) request detection reset command	Resets CH1 external preset (phase Z) request detection.	- This relay is turned on to reset CH1 external preset (phase Z) request detection. - The command is valid at the rise of this relay (off → on). - The on and off time must be held for at least 2ms.			
SM1898	CH1 pulse measurement start command	Starts pulse measurement.	- This relay is turned on to measure pulses. - The command is valid while this relay is on.			
SM1899	CH1 error reset command	Resets the CH1 error.	- This relay is turned on to reset the CH1 error. - The command is valid at the rise of this relay (off → on). - The on and off time must be held for at least 2ms.			

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU M9□□□	Corresponding CPU
SM1900	CH2 counter value greater (No.1)	OFF : Coincidence point (No.1) or smaller ON : Greater than coincidence point (No.1)	• This relay turns on when "current value of CH2 > coincidence output No.1 point setting value" is met. • This relay turns off when "current value of CH2 ≤ coincidence output No.1 point setting value" is met.	S (Every END processing)	New	LCPU
SM1901	CH2 counter value coincidence (No.1)	OFF : Not detected ON : Detected	• This relay turns on when "current value of CH2 = coincidence output No.1 point setting value" is met. • This relay is turned off by turning on CH2 coincidence signal No.1 reset command.	S (Status change/ Every END processing)		
SM1902	CH2 counter value smaller (No.1)	OFF : Coincidence point (No.1) or greater ON : Smaller than coincidence point (No.1)	• This relay turns on when "current value of CH2 < coincidence output No.1 point setting value" is met. • This relay turns off when "current value of CH2 ≥ coincidence output No.1 point setting value" is met.	S (Every END processing)		
SM1903	CH2 counter value greater (No.2)	OFF : Coincidence point (No.2) or smaller ON : Greater than coincidence point (No.2)	• This relay turns on when "current value of CH2 > coincidence output No.2 point setting value" is met. • This relay turns off when "current value of CH2 ≤ coincidence output No.2 point setting value" is met.			
SM1904	CH2 counter value coincidence (No.2)	OFF : Not detected ON : Detected	• This relay turns on when "current value of CH2 = coincidence output No.2 point setting value" is met. • This relay is turned off by turning on CH2 coincidence signal No.2 reset command.	S (Status change/ Every END processing)		
SM1905	CH2 counter value smaller (No.2)	OFF : Coincidence point (No.2) or greater ON : Smaller than coincidence point (No.2)	• This relay turns on when "current value of CH2 < coincidence output No.2 point setting value" is met. • This relay turns off when "current value of CH2 ≥ coincidence output No.2 point setting value" is met.	S (Every END processing)		
SM1906	CH2 external preset (phase Z) request detection	OFF : Not detected ON : Detected	• This relay turns on when a preset request by phase Z (preset) terminal of CH2 is detected. • This relay is turned off by turning on CH2 external preset (phase Z) request detection clear command.			
SM1907	CH2 Error	OFF : No error ON : Error	• This relay turns on if the CH2 error occurs. • This relay turns off when an error cause is removed and CH2 error reset command is turned on.			
SM1908	CH2 warning	OFF : No warning ON : Warning	• This relay turns on if a warning occurs in CH2. • This relay turns off when a warning cause is removed and CH2 error reset command is turned on.			

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU M9□□□	Corresponding CPU
SM1910	CH2 coincidence signal No.1 reset command	Resets CH2 counter value coincidence No.1.	- This relay is turned on to reset CH2 counter value coincidence No.1. - The command is valid while this relay is on. - The on time must be held for at least 2ms.	U	New	LCPU
SM1911	CH2 coincidence signal No.2 reset command	Resets CH2 counter value coincidence No.2.	- This relay is turned on to reset CH2 counter value coincidence No.2. - The command is valid while this relay is on. - The on time must be held for at least 2ms.			
SM1912	CH2 coincidence output enable command	Controls outputs from CH2 coincidence output No.1 and No.2 terminals.	- This relay is turned on to perform coincidence output from CH2 coincidence output No.1 and CH2 coincidence output No.2 terminals. - The command is valid while this relay is on.			
SM1913	CH2 preset command	Presets the counter value.	- This relay is turned on to preset the counter value. - The command is valid at the rise of this relay (off → on). - The on and off time must be held for at least 2ms.			
SM1914	CH2 count down command	Counts down pulses.	- This relay is turned on to count down pulses. - The command is valid while the Pulse input mode is either 1-phase multiple of n or 1-phase multiple of n (A phase only). - The command is valid while this relay is on.			
SM1915	CH2 count enable command	Starts counting.	- This relay is turned on to start counting. - The command is valid while this relay is on.			
SM1916	CH2 counter function selection start command	Starts the selected counter function.	- This relay is turned on to start the selected counter function. - When the count disabling function is selected, the command is valid while this relay is on. - When the latch counter function or the sampling counter function is selected, the command is valid at the rise of this relay (off → on). The on time must be held for at least 2ms. - When the count disabling/preset function or the latch counter/preset function is selected, the command is invalid.			
SM1917	CH2 external preset (phase Z) request detection reset command	Resets CH2 external preset (phase Z) request detection.	- This relay is turned on to reset CH2 external preset (phase Z) request detection. - The command is valid at the rise of this relay (off → on). - The on and off time must be held for at least 2ms.			
SM1918	CH2 pulse measurement start command	Starts pulse measurement.	- This relay is turned on to measure pulses. - The command is valid while this relay is on.			
SM1919	CH2 error reset command	Resets the CH2 error.	- This relay is turned on to reset the CH2 error. - The command is valid at the rise of this relay (off → on). - The on and off time must be held for at least 2ms.			

(17) Data logging

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU M9□□□	Corresponding CPU
SM1940	Data logging setting No.1 Data logging preparation	OFF : Not ready ON : Ready	This relay turns on when the system is ready for data logging. This relay remains on even after data logging is suspended. However, this relay turns off when data logging is stopped.	S (Initial)	New	LCPU
SM1941	Data logging setting No.1 Data logging start	OFF : Pause ON : Start	This relay is turned on to start data logging and is turned off to suspend data logging. (The related special relays will all turn off.)	U		
SM1942	Data logging setting No.1 Data logging collection	OFF : Not being collected ON : Being collected	This relay is on while data logging is being collected.	S (Status change)		
SM1943	Data logging setting No.1 Data logging end	OFF : Not ended ON : Ended	This relay turns on when data logging is ended. [Continuous is set for Logging type] The corresponding bit turns on when data logging is ended after data have been written by the number of storable files (Stop is set for Operation occurring when number of saved files is exceeded). [Trigger is set for Logging type] The corresponding bit turns on when the trigger condition is met, data are collected by the number of set times, and then the data are written to the SD memory card. This relay also turns on if an error occurs during data logging (except data logging error occurred by the execution of online change).			
SM1944	Data logging setting No.1 Data logging trigger	OFF→ON: Triggered	• This relay turns on when the specified trigger condition is met. • This relay is turned on to meet the trigger condition.			
SM1945	Data logging setting No.1 After data logging trigger	OFF : Not triggered ON : Triggered	This relay turns on after trigger logging is triggered. This relay remains on even after data logging is completed. This relay turns off when trigger logging is suspended or stopped.	S (Status change)		
SM1946	Data logging setting No.1 Data logging error	OFF : No error ON : Error	This relay turns on if a data logging error occurs. This relay is turned off by the registration of the setting or a stop command from LCPU Logging Configuration Tool.	S (Error)		
SM1947	Data logging setting No.1 Data storage in SD memory card	OFF : Not stored ON : Being stored	This relay is on while buffer memory data are being stored to a SD memory card by data logging.	S (Status change)		

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU M9□□□	Corresponding CPU
SM1950 to SM1957	Data logging setting No.2	Same as in data logging setting No.1	Same as in data logging setting No.1 (SM1940 to SM1947)	Same as in data logging setting No.1	New	LCPUCPU
SM1960 to SM1967	Data logging setting No.3					
SM1970 to SM1977	Data logging setting No.4					
SM1980 to SM1987	Data logging setting No.5					
SM1990 to SM1997	Data logging setting No.6					
SM2000 to SM2007	Data logging setting No.7					
SM2010 to SM2017	Data logging setting No.8					
SM2020 to SM2027	Data logging setting No.9					
SM2030 to SM2037	Data logging setting No.10					

12.2 Special Register List

The special register (SD) is an internal register whose application is fixed in the programmable controller. For this reason, the special register cannot be used in the same way as other internal registers are used in sequence programs. However, data can be written to the special register to control the CPU module as needed. Data is stored in binary format if not specified.

The following table shows how to read the special register list.

Item	Description
Number	Special register number
Name	Special register name
Meaning	Contents of special register
Explanation	Detailed description of special register
Set by (When Set)	Set side and set timing of special register <Set by> • S: Set by system • U: Set by user (using a program, programming tool, GOT, or test operation from other external devices) • S/U: Set by both system and user <When Set> The following shows the set timing when the special register is set by system. • Every END processing: Set during every END processing • Initial: Set during initial processing (after power-on or status change from STOP to RUN) • Status change: Set when the operating status is changed • Error: Set if an error occurs • Instruction execution: Set when an instruction is executed • Request: Set when requested by a user (using the special relay) • When condition occurs: Set when the condition is triggered • When system is switched: Set when the system is switched (between the control system and the standby system) • When RUN/STOP/RESET switch changed: Set when the RUN/STOP/RESET switch is changed • Card removal: Set when a memory card is inserted or removed • At write: Set when data is written to the CPU module
Corresponding CPU	CPU module supporting the special register • QCPU: All the Q series CPU modules • Q00J/Q00/Q01: Basic model QCPU • Qn(H): High Performance model QCPU • QnPH: Process CPU • QnPRH: Redundant CPU • QnU: Universal model QCPU • Q00UJ/Q00U/Q01U: Q00UJCPU, Q00UCPU, and Q01UCPU • LCPU: All the L series CPU modules • CPU module model: Only the specified model (Example: Q02UCPU, L26CPU-BT)
Corresponding ACPU D9□□□	• Special register (D9□□□) supported by the ACPU ("D9□□□ format change" indicates the one whose application has been changed. Incompatible with the Q00J/Q00/Q01 and QnPRH.) • "New" indicates the one added for the QCPU or LCPU.

For details on the following items, refer to the following.

- For network related items:  Manuals for each network module
- For SFC programs:  MELSEC-Q/L/QnA Programming Manual (SFC)

Point

Do not change the values of special register set by system using a program or by test operation. Doing so may result in system down or communication failure.

(1) Diagnostic information

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU
SD0	Diagnostic errors	Diagnosis error code	- This register stores the error code of an error detected by diagnostics. - Contents identical to latest fault history information.	S (Error)	D9008 format change	QCPU LCPU
SD1	Clock time for diagnosis error occurrence	Clock time for diagnosis error occurrence	This register stores the year (last two digits) and the month when the SD0 data is updated in 4-digit BCD. b15 to b8 b7 to b0 Year (0 to 99) Month (1 to 12) (Example) October, 1995 9510_H		New	
SD2			This register stores the day and the hour when the SD0 data is updated in 4-digit BCD. b15 to b8 b7 to b0 Day (1 to 31) Hour (0 to 23) (Example) 10 a.m. on 25th 2510_H			
SD3			This register stores the minute and the second when the SD0 data is updated in 4-digit BCD. b15 to b8 b7 to b0 Minutes (0 to 59) Seconds (0 to 59) (Example) 35 min. 48 sec. 3548_H			

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU
SD4	Error information categories	Error information category code	- Error information is stored in Error common information (SD5 to SD15) and Error individual information (SD16 to SD26). - This register stores a category code indicating an error information type. b15to b8 b7to b0 Individual information category codesCommon information category codes - The common information category codes store the following codes: 0: No error 1: Module No. (Slot No./CPU No./base No.)*1*2 2: File name/drive name 3: Time (value set) 4: Program error location 5: Reason(s) for system switching (for Redundant CPU) 6: Reason(s) for tracking size excess error (for Redundant CPU) 7: Base No./power supply No. (Except for the Universal model QCPU and LCPU having the serial No. (first five digits) of "10041" or earlier) 8: Tracking transmission data classification (for Redundant CPU) *1 For a multiple CPU system, the module No. or CPU No. is stored according to an error. (To determine whether a storage value is a module No. or CPU No., refer to each error code.) CPU No. 1: 1, CPU No. 2: 2, CPU No. 3: 3, CPU No. 4: 4 *2 For the LCPU, only a slot No. is stored. - The individual information category codes store the following codes: 0: No error 1: (Empty) 2: File name/drive name 3: Time (value actually measured) 4: Program error location 5: Parameter number 6: Annunciator (F) No. 7: CHK instruction failure No. (Except for the Basic model QCPU, Universal model QCPU, and LCPU.) 8: Reason(s) for system switching failure (for Redundant CPU) 9: Failure information (for LCPU) 12: File diagnostic information (for Universal model QCPU and LCPU) 13: Parameter No./CPU No. (for Universal model QCPU)	S (Error)	New	QCPU LCPU

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU																						
SD5	Error common information	Error common information	- This register stores common information corresponding to the error code stored in SD0. - The following ten types of information are stored here: - The error common information type can be determined by "common information category code" stored in SD4. (Values stored in "common information category code" correspond to the following 1) to 8).) 1) Module No. <table border="1"><thead><tr><th>Number</th><th>Meaning</th></tr></thead><tbody><tr><td>SD5</td><td>Slot No./CPU No./Base No.*1, *2, *3,*4,*5</td></tr><tr><td>SD6</td><td>I/O No. *6</td></tr><tr><td>SD7</td><td rowspan="9">(Empty)</td></tr><tr><td>SD8</td></tr><tr><td>SD9</td></tr><tr><td>SD10</td></tr><tr><td>SD11</td></tr><tr><td>SD12</td></tr><tr><td>SD13</td></tr><tr><td>SD14</td></tr><tr><td>SD15</td></tr></tbody></table> *1 For a multiple CPU system, the module No. or CPU No. is stored according to an error. (To determine whether a storage value is a module No. or CPU No., refer to each error code.) CPU No. 1: 1, CPU No. 2: 2, CPU No. 3: 3, CPU No. 4: 4 *2 If a fuse has been blown or an I/O module verification error occurs in a module on the MELSECNET/H remote I/O station, the network No. is stored in the upper 8 bits and the station No. is stored in the lower 8 bits. To determine a fuse-blown module or a module where an I/O module verification error occurs, check the I/O No. *3: If an instruction is executed to the Basic model QCPU on the slot where the module cannot be mounted, "255" is stored in SD5. *4 The definitions of the base No. and slot No. are as follows: [Base No.] This number indicates a base unit on which the CPU module is mounted. <table border="1"><thead><tr><th>Base No.</th><th>Definition</th></tr></thead><tbody><tr><td>0</td><td>Indicates the main base unit where a CPU module is mounted.</td></tr><tr><td>1 to 7</td><td>Indicates the extension base unit. The stage number setting made by the stage number setting connector on the extension base unit is the base No. When stage number setting is extension 1: Base No. = 1 When stage number setting is extension 7: Base No. = 7</td></tr></tbody></table> [Slot No.] This number is used to identify the slot No. of a module where an error occurs. The "0" I/O slot (slot on the right of the CPU slot) on the main base unit is defined as "Slot No. = 0". The slot Nos. are assigned in sequence numbers in order of the main base unit and then the first extension base unit to 7th extension base unit. When the number of slots on base units has been set in the I/O assignment tab of the PLC Parameter dialog box, the slot Nos. are assigned by the number of set slots. *5 If a module is not mounted on any slots as set, FF_H is stored. *6 If FFFF_H is stored in SD6 (I/O No.), this indicates that the I/O No. cannot be identified due to such as overlap of an I/O No. in the I/O assignment setting of the PLC Parameter dialog box. In this case, identify the error location using SD5. (To the next page)	Number	Meaning	SD5	Slot No./CPU No./Base No.*1, *2, *3,*4,*5	SD6	I/O No. *6	SD7	(Empty)	SD8	SD9	SD10	SD11	SD12	SD13	SD14	SD15	Base No.	Definition	0	Indicates the main base unit where a CPU module is mounted.	1 to 7	Indicates the extension base unit. The stage number setting made by the stage number setting connector on the extension base unit is the base No. When stage number setting is extension 1: Base No. = 1 When stage number setting is extension 7: Base No. = 7	S (Error)	New	QCPU LCP
Number				Meaning																								
SD5				Slot No./CPU No./Base No.*1, *2, *3,*4,*5																								
SD6				I/O No. *6																								
SD7				(Empty)																								
SD8																												
SD9																												
SD10																												
SD11																												
SD12																												
SD13																												
SD14																												
SD15																												
Base No.				Definition																								
0				Indicates the main base unit where a CPU module is mounted.																								
1 to 7	Indicates the extension base unit. The stage number setting made by the stage number setting connector on the extension base unit is the base No. When stage number setting is extension 1: Base No. = 1 When stage number setting is extension 7: Base No. = 7																											
SD6																												
SD7																												
SD8																												
SD9																												
SD10																												
SD11																												
SD12																												
SD13																												
SD14																												
SD15																												

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU																																			
SD5	Error common information	Error common information	2) File name/drive name	S (Error)	New	QCPU LCPU																																			
SD6																																									
SD7																																									
SD8																																									
SD9																																									
SD10																																									
SD11																																									
SD12																																									
SD13																																									
SD14																																									
SD15																																									
<table><tr><th>Number</th><th>Meaning</th></tr><tr><td>SD5</td><td>Drive</td></tr><tr><td>SD6</td><td rowspan="4">File name (ASCII code: 8 characters)</td></tr><tr><td>SD7</td></tr><tr><td>SD8</td></tr><tr><td>SD9</td></tr><tr><td>SD10</td><td>Extension *7</td><td>2EH(.)</td></tr><tr><td>SD11</td><td>(ASCII code: 3 characters)</td><td></td></tr><tr><td>SD12</td><td rowspan="4">(Empty)</td><td></td></tr><tr><td>SD13</td></tr><tr><td>SD14</td></tr><tr><td>SD15</td></tr></table> (Example) File name = ABCDEFGH. IJK b15 to b8 b7 to b0 <table><tr><td>42H(B)</td><td>41H(A)</td></tr><tr><td>44H(D)</td><td>43H(C)</td></tr><tr><td>46H(F)</td><td>45H(E)</td></tr><tr><td>48H(H)</td><td>47H(G)</td></tr><tr><td>49H(I)</td><td>2EH(.)</td></tr><tr><td>4BH(K)</td><td>4AH(J)</td></tr></table>							Number	Meaning	SD5	Drive	SD6	File name (ASCII code: 8 characters)	SD7	SD8	SD9	SD10	Extension *7	2EH(.)	SD11	(ASCII code: 3 characters)		SD12	(Empty)		SD13	SD14	SD15	42H(B)	41H(A)	44H(D)	43H(C)	46H(F)	45H(E)	48H(H)	47H(G)	49H(I)	2EH(.)	4BH(K)	4AH(J)		
Number			Meaning																																						
SD5			Drive																																						
SD6			File name (ASCII code: 8 characters)																																						
SD7																																									
SD8																																									
SD9																																									
SD10	Extension *7	2EH(.)																																							
SD11	(ASCII code: 3 characters)																																								
SD12	(Empty)																																								
SD13																																									
SD14																																									
SD15																																									
42H(B)	41H(A)																																								
44H(D)	43H(C)																																								
46H(F)	45H(E)																																								
48H(H)	47H(G)																																								
49H(I)	2EH(.)																																								
4BH(K)	4AH(J)																																								
3) Time (value set)																																									
<table><tr><th>Number</th><th>Meaning</th></tr><tr><td>SD5</td><td>Time : 1μs units (0 to 999μs)</td></tr><tr><td>SD6</td><td>Time : 1ms units (0 to 65535ms)</td></tr><tr><td>SD7</td><td rowspan="9">(Empty)</td></tr><tr><td>SD8</td></tr><tr><td>SD9</td></tr><tr><td>SD10</td></tr><tr><td>SD11</td></tr><tr><td>SD12</td></tr><tr><td>SD13</td></tr><tr><td>SD14</td></tr><tr><td>SD15</td></tr></table>				Number	Meaning	SD5	Time : 1μs units (0 to 999μs)	SD6	Time : 1ms units (0 to 65535ms)	SD7	(Empty)	SD8	SD9	SD10	SD11	SD12	SD13	SD14	SD15																						
Number	Meaning																																								
SD5	Time : 1μs units (0 to 999μs)																																								
SD6	Time : 1ms units (0 to 65535ms)																																								
SD7	(Empty)																																								
SD8																																									
SD9																																									
SD10																																									
SD11																																									
SD12																																									
SD13																																									
SD14																																									
SD15																																									
4) Program error location																																									
<table><tr><th>Number</th><th>Meaning</th></tr><tr><td>SD5</td><td rowspan="4">File name (ASCII code: 8 characters)</td></tr><tr><td>SD6</td></tr><tr><td>SD7</td></tr><tr><td>SD8</td></tr><tr><td>SD9</td><td>Extension *7</td><td>2EH(.)</td></tr><tr><td>SD10</td><td>(ASCII code: 3 characters)</td><td></td></tr><tr><td>SD11</td><td>Pattern *8</td><td></td></tr><tr><td>SD12</td><td>Block No.</td><td></td></tr><tr><td>SD13</td><td>Step No./transition condition</td><td></td></tr><tr><td>SD14</td><td>Sequence step No. (L)</td><td></td></tr><tr><td>SD15</td><td>Sequence step No. (H)</td><td></td></tr></table>				Number	Meaning	SD5	File name (ASCII code: 8 characters)	SD6	SD7	SD8	SD9	Extension *7	2EH(.)	SD10	(ASCII code: 3 characters)		SD11	Pattern *8		SD12	Block No.		SD13	Step No./transition condition		SD14	Sequence step No. (L)		SD15	Sequence step No. (H)											
Number	Meaning																																								
SD5	File name (ASCII code: 8 characters)																																								
SD6																																									
SD7																																									
SD8																																									
SD9	Extension *7	2EH(.)																																							
SD10	(ASCII code: 3 characters)																																								
SD11	Pattern *8																																								
SD12	Block No.																																								
SD13	Step No./transition condition																																								
SD14	Sequence step No. (L)																																								
SD15	Sequence step No. (H)																																								
*7 For extension names, refer to the following table.																																									
*8 The description of the bit pattern is as follows:																																									
<table><tr><td>15</td><td>14</td><td>to</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>← (Bit number)</td></tr><tr><td>0</td><td>0</td><td>to</td><td>0</td><td>0</td><td>*</td><td>*</td><td>*</td><td></td></tr></table> (Not used) SFC block designation present (1)/absent (0) SFC step designation present (1)/absent (0) SFC transition designation present (1)/absent (0)				15	14	to	4	3	2	1	0	← (Bit number)	0	0	to	0	0	*	*	*																					
15	14	to	4	3	2	1	0	← (Bit number)																																	
0	0	to	0	0	*	*	*																																		
(To the next page)																																									

*7: The extension names are shown below.

SD _n Higher 8 bits	SD _{n+1}		Extension Name	File Type
	Lower 8 bits	Higher 8 bits		
51H	50H	41H	QPA	Parameters
51H	50H	47H	QPG	Program
51H	43H	44H	QCD	Device comment
51H	44H	49H	QDI	Initial device value
51H	44H	52H	QDR	File register
51H	44H	4CH	QDL	Local device (For High Performance model QCPU, Process CPU, Redundant CPU, Universal model QCPU, and LCPU)
51H	54H	44H	QTD	Sampling trace data (For High Performance model QCPU, Process CPU, Redundant CPU, Universal model QCPU, and LCPU)
51H	46H	44H	QFD	Breakdown history data (For High Performance model QCPU, Process CPU, and Redundant CPU)
51H	53H	54H	QST	SP.DEVST/S.DEVLD instruction file (for Universal model QCPU and LCPU)

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU																																																																																																																			
SD5	Error common information	Error common information	5) Reason(s) for system switching	S (Error)	New	QnPRH																																																																																																																			
SD6																																																																																																																									
SD7																																																																																																																									
SD8																																																																																																																									
SD9																																																																																																																									
SD10																																																																																																																									
SD11																																																																																																																									
SD12																																																																																																																									
SD13																																																																																																																									
SD14																																																																																																																									
SD15																																																																																																																									
<table><tr><th>Number</th><th>Meaning</th></tr><tr><td>SD5</td><td>System switching cause *9</td></tr><tr><td>SD6</td><td>Control system switching instruction argument</td></tr><tr><td>SD7</td><td rowspan="9">(Empty)</td></tr><tr><td>SD8</td></tr><tr><td>SD9</td></tr><tr><td>SD10</td></tr><tr><td>SD11</td></tr><tr><td>SD12</td></tr><tr><td>SD13</td></tr><tr><td>SD14</td></tr><tr><td>SD15</td></tr></table>			Number				Meaning	SD5	System switching cause *9	SD6	Control system switching instruction argument	SD7	(Empty)	SD8	SD9	SD10	SD11	SD12	SD13	SD14	SD15																																																																																																				
Number			Meaning																																																																																																																						
SD5			System switching cause *9																																																																																																																						
SD6			Control system switching instruction argument																																																																																																																						
SD7	(Empty)																																																																																																																								
SD8																																																																																																																									
SD9																																																																																																																									
SD10																																																																																																																									
SD11																																																																																																																									
SD12																																																																																																																									
SD13																																																																																																																									
SD14																																																																																																																									
SD15																																																																																																																									
*9 The following shows the description. 0 : No system switching condition (default) 1 : Power-OFF, reset, hardware failure, watchdog timer error 2 : Stop error (except watchdog timer error) 3 : System switching request by network module 16 : Control system switching instruction 17 : Control system switching request from GX Developer																																																																																																																									
6) Reason(s) for tracking size excess error																																																																																																																									
The following shows block Nos. when data size that can be tracked (100K) is exceeded in the bit pattern of the corresponding special relay.																																																																																																																									
<table><tr><td></td><td>b15</td><td>b14</td><td>b13</td><td>b12</td><td>b11</td><td>b10</td><td>b9</td><td>b8</td><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr><tr><td>SD5</td><td>1 (SM1535) (Block16)</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1 (SM1528) (Block9)</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1 (SM1520) (Block1)</td></tr><tr><td>SD6</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>SD7</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>SD8</td><td>1 (SM1583) (Block64)</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1 (SM1568) (Block49)</td></tr><tr><td>SD9</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>SD15</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>				b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	SD5	1 (SM1535) (Block16)	0	0	0	0	0	0	1 (SM1528) (Block9)	0	0	0	0	0	0	0	1 (SM1520) (Block1)	SD6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SD7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SD8	1 (SM1583) (Block64)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1 (SM1568) (Block49)	SD9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SD15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0																																																																																																									
SD5	1 (SM1535) (Block16)	0	0	0	0	0	0	1 (SM1528) (Block9)	0	0	0	0	0	0	0	1 (SM1520) (Block1)																																																																																																									
SD6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																																																																																																									
SD7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																																																																																																									
SD8	1 (SM1583) (Block64)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1 (SM1568) (Block49)																																																																																																									
SD9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																																																																																																									
SD15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																																																																																																									
(To the next page)																																																																																																																									

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU																																																																																														
SD5	Error common information	Error common information	7) Base No./power supply No. <table><tr><th>Number</th><th>Meaning</th></tr><tr><td>SD5</td><td>Base No.</td></tr><tr><td>SD6</td><td>Power supply No.</td></tr><tr><td>SD7</td><td rowspan="9">(Empty)</td></tr><tr><td>SD8</td></tr><tr><td>SD9</td></tr><tr><td>SD10</td></tr><tr><td>SD11</td></tr><tr><td>SD12</td></tr><tr><td>SD13</td></tr><tr><td>SD14</td></tr><tr><td>SD15</td></tr></table> 1: Power supply 1 fault 2: Power supply 2 fault "Power supply module 1": Redundant power supply module mounted on POWER 1 slot of redundant module 1": base unit (Q38RB, Q68RB, Q65WRB) "Power supply module 2": Redundant power supply module mounted on POWER 2 slot of redundant module 2": base unit (Q38RB, Q68RB, Q65WRB)	Number	Meaning	SD5	Base No.	SD6	Power supply No.	SD7	(Empty)	SD8	SD9	SD10	SD11	SD12	SD13	SD14	SD15	S (Error)	D9□□□	Qn(H) ^{*1} QnPH ^{*1} QnPRH QnU ^{*2}																																																																														
Number				Meaning																																																																																																
SD5				Base No.																																																																																																
SD6				Power supply No.																																																																																																
SD7				(Empty)																																																																																																
SD8																																																																																																				
SD9																																																																																																				
SD10																																																																																																				
SD11																																																																																																				
SD12																																																																																																				
SD13																																																																																																				
SD14																																																																																																				
SD15																																																																																																				
SD6																																																																																																				
SD7																																																																																																				
SD8																																																																																																				
SD9																																																																																																				
SD10																																																																																																				
SD11																																																																																																				
SD12																																																																																																				
SD13																																																																																																				
SD14																																																																																																				
SD15																																																																																																				
SD13																																																																																																				
SD14																																																																																																				
SD15																																																																																																				
8) Tracking transmission data classification This register stores a data type during tracking. <table><tr><th>Number</th><th>Meaning</th></tr><tr><td>SD5</td><td>Data type *10</td></tr><tr><td>SD6</td><td rowspan="10">(Empty)</td></tr><tr><td>SD7</td></tr><tr><td>SD8</td></tr><tr><td>SD9</td></tr><tr><td>SD10</td></tr><tr><td>SD11</td></tr><tr><td>SD12</td></tr><tr><td>SD13</td></tr><tr><td>SD14</td></tr><tr><td>SD15</td></tr></table> *10 The description of the data type is as follows: <table><tr><td>b15</td><td>b14 to b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td><td></td></tr><tr><td colspan="8">0</td><td>Each bit 0: Not sent 1: Being sent</td></tr><tr><td colspan="8"></td><td>→ Device data</td></tr><tr><td colspan="8"></td><td>→ Signal flow</td></tr><tr><td colspan="8"></td><td>→ PIDINIT/S. PIDINIT instruction data</td></tr><tr><td colspan="8"></td><td>→ SFC execution data</td></tr><tr><td colspan="8"></td><td>→ System switching request</td></tr><tr><td colspan="8"></td><td>→ Operation mode change request</td></tr><tr><td colspan="8"></td><td>→ System data</td></tr></table>			Number	Meaning	SD5	Data type *10	SD6	(Empty)	SD7	SD8	SD9	SD10	SD11	SD12	SD13	SD14	SD15	b15	b14 to b6	b5	b4	b3	b2	b1	b0		0								Each bit 0: Not sent 1: Being sent									→ Device data									→ Signal flow									→ PIDINIT/S. PIDINIT instruction data									→ SFC execution data									→ System switching request									→ Operation mode change request									→ System data	New	QnPRH
Number	Meaning																																																																																																			
SD5	Data type *10																																																																																																			
SD6	(Empty)																																																																																																			
SD7																																																																																																				
SD8																																																																																																				
SD9																																																																																																				
SD10																																																																																																				
SD11																																																																																																				
SD12																																																																																																				
SD13																																																																																																				
SD14																																																																																																				
SD15																																																																																																				
b15	b14 to b6	b5	b4	b3	b2	b1	b0																																																																																													
0								Each bit 0: Not sent 1: Being sent																																																																																												
								→ Device data																																																																																												
								→ Signal flow																																																																																												
								→ PIDINIT/S. PIDINIT instruction data																																																																																												
								→ SFC execution data																																																																																												
								→ System switching request																																																																																												
								→ Operation mode change request																																																																																												
								→ System data																																																																																												

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU																																																																																			
SD16	Error individual information	Error individual information	- This register stores individual information corresponding to the error code stored in SD0. - There are the following eight different types of information are stored. - The error individual information type can be determined by "individual information category code" stored in SD4. (Values stored in "individual information category code" correspond to the following 1) to 8), 12), and 13).) 1) (Empty) 2) File name/drive name <table><tr><th>Number</th><th>Meaning</th></tr><tr><td>SD16</td><td>Drive</td></tr><tr><td>SD17</td><td rowspan="4">File name (ASCII code: 8 characters)</td></tr><tr><td>SD18</td></tr><tr><td>SD19</td></tr><tr><td>SD20</td></tr><tr><td>SD21</td><td>Extension *6</td><td>2EH(.)</td></tr><tr><td>SD22</td><td>(ASCII code: 3 characters)</td><td></td></tr><tr><td>SD23</td><td rowspan="4">(Empty)</td><td></td></tr><tr><td>SD24</td></tr><tr><td>SD25</td></tr><tr><td>SD26</td></tr></table> (Example) File name = ABCDEF GH. IJK b15 to b8 b7 to b0 42H(B) 41H(A) 44H(D) 43H(C) 46H(F) 45H(E) 48H(H) 47H(G) 49H(I) 2EH(.) 4BH(K) 4AH(J) 3) Time (value actually measured) <table><tr><th>Number</th><th>Meaning</th></tr><tr><td>SD16</td><td>Time : 1 μs units (0 to 999 μs)</td></tr><tr><td>SD17</td><td>Time : 1ms units (0 to 65535ms)</td></tr><tr><td>SD18</td><td rowspan="9">(Empty)</td></tr><tr><td>SD19</td></tr><tr><td>SD20</td></tr><tr><td>SD21</td></tr><tr><td>SD22</td></tr><tr><td>SD23</td></tr><tr><td>SD24</td></tr><tr><td>SD25</td></tr><tr><td>SD26</td></tr></table> 4) Program error location <table><tr><th>Number</th><th>Meaning</th></tr><tr><td>SD16</td><td rowspan="4">File name (ASCII code: 8 characters)</td></tr><tr><td>SD17</td></tr><tr><td>SD18</td></tr><tr><td>SD19</td></tr><tr><td>SD20</td><td>Extension *6</td><td>2EH(.)</td></tr><tr><td>SD21</td><td>(ASCII code: 3 characters)</td><td></td></tr><tr><td>SD22</td><td>Pattern *7</td><td></td></tr><tr><td>SD23</td><td>Block No.</td><td></td></tr><tr><td>SD24</td><td>Step No./transition No.</td><td></td></tr><tr><td>SD25</td><td>Sequence step No. (L)</td><td></td></tr><tr><td>SD26</td><td>Sequence step No. (H)</td><td></td></tr></table> *6 For extension names, refer to the table on Page 51. *7 The description of the bit pattern is as follows: <table><tr><td>15</td><td>14</td><td>to</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td></td></tr><tr><td>0</td><td>0</td><td>to</td><td>0</td><td>0</td><td>*</td><td>*</td><td>*</td><td>← (Bit number)</td></tr></table> (Not used) SFC block designation present (1)/absent (0) SFC step designation present (1)/absent (0) SFC transition designation present (1)/absent (0)	Number	Meaning	SD16	Drive	SD17	File name (ASCII code: 8 characters)	SD18	SD19	SD20	SD21	Extension *6	2EH(.)	SD22	(ASCII code: 3 characters)		SD23	(Empty)		SD24	SD25	SD26	Number	Meaning	SD16	Time : 1 μs units (0 to 999 μs)	SD17	Time : 1ms units (0 to 65535ms)	SD18	(Empty)	SD19	SD20	SD21	SD22	SD23	SD24	SD25	SD26	Number	Meaning	SD16	File name (ASCII code: 8 characters)	SD17	SD18	SD19	SD20	Extension *6	2EH(.)	SD21	(ASCII code: 3 characters)		SD22	Pattern *7		SD23	Block No.		SD24	Step No./transition No.		SD25	Sequence step No. (L)		SD26	Sequence step No. (H)		15	14	to	4	3	2	1	0		0	0	to	0	0	*	*	*	← (Bit number)	S (Error)	New	QCPU LCPU
Number				Meaning																																																																																					
SD16				Drive																																																																																					
SD17				File name (ASCII code: 8 characters)																																																																																					
SD18																																																																																									
SD19																																																																																									
SD20																																																																																									
SD21				Extension *6	2EH(.)																																																																																				
SD22				(ASCII code: 3 characters)																																																																																					
SD23				(Empty)																																																																																					
SD24																																																																																									
SD25																																																																																									
SD26																																																																																									
Number	Meaning																																																																																								
SD16	Time : 1 μs units (0 to 999 μs)																																																																																								
SD17	Time : 1ms units (0 to 65535ms)																																																																																								
SD18	(Empty)																																																																																								
SD19																																																																																									
SD20																																																																																									
SD21																																																																																									
SD22																																																																																									
SD23																																																																																									
SD24																																																																																									
SD25																																																																																									
SD26																																																																																									
Number	Meaning																																																																																								
SD16	File name (ASCII code: 8 characters)																																																																																								
SD17																																																																																									
SD18																																																																																									
SD19																																																																																									
SD20	Extension *6	2EH(.)																																																																																							
SD21	(ASCII code: 3 characters)																																																																																								
SD22	Pattern *7																																																																																								
SD23	Block No.																																																																																								
SD24	Step No./transition No.																																																																																								
SD25	Sequence step No. (L)																																																																																								
SD26	Sequence step No. (H)																																																																																								
15	14	to	4	3	2	1	0																																																																																		
0	0	to	0	0	*	*	*	← (Bit number)																																																																																	
SD17																																																																																									
SD18																																																																																									
SD19																																																																																									
SD20																																																																																									
SD21																																																																																									
SD22																																																																																									
SD23																																																																																									
SD24																																																																																									
SD25																																																																																									
SD26																																																																																									

(To the next page)

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU																					
SD26	Error individual information	Error individual information	5) Parameter No. 6) Annunciator number / 7) CHK instruction malfunction number <table><tr><th>Number</th><th>Meaning</th><th>Number</th><th>Meaning</th></tr><tr><td>SD16</td><td>Parameter No. *8</td><td>SD16</td><td>No.</td></tr><tr><td>SD17</td><td rowspan="11">(Empty)</td><td>SD17</td><td rowspan="11">(Empty)</td></tr><tr><td>SD18</td></tr><tr><td>SD19</td></tr><tr><td>SD20</td></tr><tr><td>SD21</td></tr><tr><td>SD22</td></tr><tr><td>SD23</td></tr><tr><td>SD24</td></tr><tr><td>SD25</td></tr><tr><td>SD26</td></tr></table> *8 For details of the parameter No., refer to the following:  User's Manual (Function Explanation, Program Fundamentals) for the CPU module used	Number	Meaning	Number	Meaning	SD16	Parameter No. *8	SD16	No.	SD17	(Empty)	SD17	(Empty)	SD18	SD19	SD20	SD21	SD22	SD23	SD24	SD25	SD26	S (Error)	New	QCPU LCPU
			Number	Meaning	Number	Meaning																					
SD16	Parameter No. *8	SD16	No.																								
SD17	(Empty)	SD17	(Empty)																								
SD18																											
SD19																											
SD20																											
SD21																											
SD22																											
SD23																											
SD24																											
SD25																											
SD26																											
8) Reason(s) for system switching failure <table><tr><th>Number</th><th>Meaning</th></tr><tr><td>SD16</td><td>System switching prohibition condition *9</td></tr><tr><td>SD17</td><td rowspan="10">(Empty)</td></tr><tr><td>SD18</td></tr><tr><td>SD19</td></tr><tr><td>SD20</td></tr><tr><td>SD21</td></tr><tr><td>SD22</td></tr><tr><td>SD23</td></tr><tr><td>SD24</td></tr><tr><td>SD25</td></tr><tr><td>SD26</td></tr></table> *9 The following shows the description. 0 : Normal switching completion (default) 1 : Tracking cable fault (cable removal, cable fault, internal circuit fault, hardware fault) 2 : Hardware failure, power OFF, reset or watchdog timer error occurring in standby system 3 : Hardware failure, power OFF, reset or watchdog timer error occurring in control system 4 : Preparing for tracking 5 : Time limit exceeded 6 : Standby system is in stop error (except watchdog timer error) 7 : Operation differs between two systems (in backup mode only) 8 : During memory copy from control system to standby system 9 : Online program change 10 : Error detected by network module of standby system 11 : System switching being executed 12 : Online module change in progress		Number		Meaning	SD16	System switching prohibition condition *9	SD17	(Empty)	SD18	SD19	SD20	SD21	SD22	SD23	SD24	SD25	SD26	QnPRH									
Number	Meaning																										
SD16	System switching prohibition condition *9																										
SD17	(Empty)																										
SD18																											
SD19																											
SD20																											
SD21																											
SD22																											
SD23																											
SD24																											
SD25																											
SD26																											

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU																																
SD26	Error individual information	Error individual information	9) Failure information <table><tr><th>Number</th><th>Meaning</th></tr><tr><td>SD16</td><td>Failure information 1</td></tr><tr><td>SD17</td><td>Failure information 2</td></tr><tr><td>SD18</td><td>Failure information 3</td></tr><tr><td>SD19</td><td>Failure information 4</td></tr><tr><td>SD20</td><td>Failure information 5</td></tr><tr><td>SD21</td><td>Failure information 6</td></tr><tr><td>SD22</td><td>Failure information 7</td></tr><tr><td>SD23</td><td>Failure information 8</td></tr><tr><td>SD24</td><td>Failure information 9</td></tr><tr><td>SD25</td><td>Failure information 10</td></tr><tr><td>SD26</td><td>Failure information 11</td></tr></table>	Number	Meaning	SD16	Failure information 1	SD17	Failure information 2	SD18	Failure information 3	SD19	Failure information 4	SD20	Failure information 5	SD21	Failure information 6	SD22	Failure information 7	SD23	Failure information 8	SD24	Failure information 9	SD25	Failure information 10	SD26	Failure information 11	S (Error)	New	LCPU								
			Number	Meaning																																		
			SD16	Failure information 1																																		
SD17	Failure information 2																																					
SD18	Failure information 3																																					
SD19	Failure information 4																																					
SD20	Failure information 5																																					
SD21	Failure information 6																																					
SD22	Failure information 7																																					
SD23	Failure information 8																																					
SD24	Failure information 9																																					
SD25	Failure information 10																																					
SD26	Failure information 11																																					
12) File diagnostic information <table><tr><td>SD16</td><td>Failure information1 (H)</td><td>drive No.(L)</td></tr><tr><td>SD17</td><td colspan="2" rowspan="4">File name (ASCII: 8 characters)</td></tr><tr><td>SD18</td></tr><tr><td>SD19</td></tr><tr><td>SD20</td></tr><tr><td>SD21</td><td>EXTension *6</td><td>2EH(.)</td></tr><tr><td>SD22</td><td colspan="2">(ASCII; 3 characters)</td></tr><tr><td>SD23</td><td colspan="2">Failure information 2 (CRC value that is read)</td></tr><tr><td>SD24</td><td colspan="2">Failure information 3 (CRC value that is calculated)</td></tr><tr><td>SD25</td><td colspan="2"></td></tr><tr><td>SD26</td><td colspan="2"></td></tr></table>	SD16	Failure information1 (H)	drive No.(L)	SD17	File name (ASCII: 8 characters)		SD18	SD19	SD20	SD21	EXTension *6	2EH(.)	SD22	(ASCII; 3 characters)		SD23	Failure information 2 (CRC value that is read)		SD24	Failure information 3 (CRC value that is calculated)		SD25			SD26			U	QnU LCPU									
SD16	Failure information1 (H)	drive No.(L)																																				
SD17	File name (ASCII: 8 characters)																																					
SD18																																						
SD19																																						
SD20																																						
SD21	EXTension *6	2EH(.)																																				
SD22	(ASCII; 3 characters)																																					
SD23	Failure information 2 (CRC value that is read)																																					
SD24	Failure information 3 (CRC value that is calculated)																																					
SD25																																						
SD26																																						
13) Parameter No./CPU No. <table><tr><th>Number</th><th>Meaning</th></tr><tr><td>SD16</td><td>Parameter No.*16</td></tr><tr><td>SD17</td><td>CPU No. (1 to 4)</td></tr><tr><td>SD18</td><td rowspan="9">(Empty)</td></tr><tr><td>SD19</td></tr><tr><td>SD20</td></tr><tr><td>SD21</td></tr><tr><td>SD22</td></tr><tr><td>SD23</td></tr><tr><td>SD24</td></tr><tr><td>SD25</td></tr><tr><td>SD26</td></tr></table> *16 For details of the parameter No., refer to the following:  User's Manual (Function Explanation, Program Fundamentals) for the CPU module used	Number	Meaning	SD16	Parameter No.*16	SD17	CPU No. (1 to 4)	SD18	(Empty)	SD19	SD20	SD21	SD22	SD23	SD24	SD25	SD26	QnU																					
Number	Meaning																																					
SD16	Parameter No.*16																																					
SD17	CPU No. (1 to 4)																																					
SD18	(Empty)																																					
SD19																																						
SD20																																						
SD21																																						
SD22																																						
SD23																																						
SD24																																						
SD25																																						
SD26																																						
SD50	Error reset	Error number that performs error reset	Stores error number that performs error reset			QCPU LCPU																																
SD51	Battery low latch	Bit pattern indicating where battery voltage drop occurred	• If a battery voltage drops, the corresponding bit stores "1" (turns on). • This register remains on even after the battery voltage returns to normal. <table><tr><td>b15</td><td>to</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td><td></td></tr><tr><td colspan="4">0</td><td></td><td></td><td></td></tr><tr><td colspan="4"></td><td></td><td></td><td rowspan="3">} *1</td></tr><tr><td colspan="4"></td><td></td><td></td></tr><tr><td colspan="4"></td><td></td><td></td></tr></table> → Battery error for CPU module → SRAM card battery alarm → SRAM card battery error *1 These bits are not available for the Basic model QCPU and LCPU. • If an alarm occurs, data can be held within the time specified for battery low. • The error indicates full discharge of a battery.	b15	to	b3	b2	b1	b0		0													} *1													S (Error)	QCPU LCPU
b15	to	b3	b2	b1	b0																																	
0																																						
						} *1																																

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU
SD81	Continuation error cause	Continuation error cause	This register stores a continuation error cause b15 b12 b11 b8 b7 b4 b3 b0 SD81			

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU
SD100	Transmission speed storage area	Stores the transmission speed specified in the serial communication setting.	96 : 9.6kbps,192 : 19.2kbps,384 : 38.4kbps, 576 : 57.6kbps,1152 : 115.2kbps	S (Power-ON or reset)	New	Q00/Q01 Q00UJ Q00U Q01U Q02UCPU*4
SD101	Communication setting storage area	Stores the communication setting specified in the serial communication setting.	b15to b6b5b4b3to b0 Since this area is reserved for a system, storage data are variable. Write during RUN setting 0: Disabled 1: Enabled Since this area is reserved for a system, storage data are variable. Sumcheck presence 0: Absent 1: Present			
SD102	Transmission wait time storage area	Stores the transmission wait time specified in the serial communication setting.	0: No waiting time 10 to 150: Waiting time (unit: ms) Default: 0			
SD105	CH1 transmission speed setting (RS-232)	Stores a transmission speed set in a programming tool.	96 : 9600bps, 192 : 19.2kbps,384 : 38.4kbps, 576 : 57.6kbps, 1152 : 115.2kbps This register holds a value stored in RS-232 connection even after communication in other than RS-232 connection is started. (When no connection is made, this register stores "1152".)	S		Qn(H) QnPH QnPRH QnU*3 LCPU
SD110	Data sending result storage area	Stores the data sending result when the serial communication function is used.	Stores the error code at the time of sending data.	S (Error)		Q00/Q01 Q00UJ Q00U Q01U Q02UCPU*4
SD111	Data receiving result storage area	Stores the data receiving result when the serial communication function is used.	Stores the error code at the time of receiving data.			
SD118	Amount of battery consumption	Amount of battery consumption	This register stores a battery consumption rate. [Value range] • 1 or 2: Q00UJCPU, Q00UCPU, Q01UCPU, Q02UCPU, Q03UD(E)CPU, Q04UD(E)HCPU, L02CPU • 1 to 3: Q06UD(E)HCPU, L26CPU-BT • 1 to 4: Q10UD(E)HCPU, Q20UD(E)HCPU, Q13UD(E)HCPU, Q26UD(E)HCPU • 1 to 5: Q50UDEHCPU, Q100UDEHCPU	S (Status change)		QnU LCPU

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU																																																																				
SD119	Battery life-prolonging factor	Battery life-prolonging factor	This register stores a value indicating a cause that has the battery life-prolonging function enabled. While this register is other than "0", the battery life-prolonging function is enabled. 0:No factor 1:Factor b15 to b2 b1 b0 Fixed to 0 b0: CPU switch setting b1: Backup in execution by latch data backup function (to standard ROM)	S (Status change)	New	QnU LCPU																																																																				
SD130	Fuse blown module	Bit pattern in units of 16 points, indicating the modules whose fuses have blown 0: No blown fuse 1: Blown fuse present	• This register stores the number of a fuse-blown output module (in units of 16 points) in the following bit pattern. (When module numbers have been set by the parameter, the parameter-set numbers are stored.) <table><tr><td></td><td>b15</td><td>b14</td><td>b13</td><td>b12</td><td>b11</td><td>b10</td><td>b9</td><td>b8</td><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr><tr><td>SD130</td><td>0</td><td>0</td><td>0</td><td>1 (YC0)</td><td>0</td><td>0</td><td>0</td><td>1 (Y90)</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>SD131</td><td>1 (Y1F0)</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1 (Y1A0)</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>SD137</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1 (Y7B0)</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1 (Y730)</td><td>0</td><td>0</td><td>0</td></tr></table> Indicates fuse blow.			b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	SD130	0	0	0	1 (YC0)	0	0	0	1 (Y90)	0	0	0	0	0	0	0	0	SD131	1 (Y1F0)	0	0	0	0	1 (Y1A0)	0	0	0	0	0	0	0	0	0	0	SD137	0	0	0	0	1 (Y7B0)	0	0	0	0	0	0	0	1 (Y730)	0	0	0	S (Error)	Q00J/Q00/ Q01
			b15	b14		b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0																																																							
SD130			0	0		0	1 (YC0)	0	0	0	1 (Y90)	0	0	0	0	0	0	0	0																																																							
SD131			1 (Y1F0)	0		0	0	0	1 (Y1A0)	0	0	0	0	0	0	0	0	0	0																																																							
SD137			0	0		0	0	1 (Y7B0)	0	0	0	0	0	0	0	1 (Y730)	0	0	0																																																							
SD131																																																																										
SD132																																																																										
SD133																																																																										
SD134																																																																										
SD135																																																																										
SD136																																																																										
SD137																																																																										
SD150	I/O module verify error	Bit pattern, in units of 16 points, indicating the modules with verify errors. 0: No I/O verify errors 1: I/O verify error present	• If the status of the I/O module changes from that obtained at power-on, the module No. (unit: 16 points) is stored in the following bit pattern. (When I/O module numbers have been set by the parameter, the parameter-set numbers are stored.) <table><tr><td></td><td>b15</td><td>b14</td><td>b13</td><td>b12</td><td>b11</td><td>b10</td><td>b9</td><td>b8</td><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr><tr><td>SD150</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1 (X₀^Y)</td></tr><tr><td>SD151</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1 (X₁^Y)</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>SD157</td><td>0</td><td>1 (X₇^Y)</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table> Indicates an I/O module verify error.		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	SD150	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1 (X ₀ ^Y)	SD151	0	0	0	0	0	0	1 (X ₁ ^Y)	0	0	0	0	0	0	0	0	0	SD157	0	1 (X ₇ ^Y)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	S (Error)	Q00J/Q00/ Q01	
			b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0																																																								
SD150			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1 (X ₀ ^Y)																																																								
SD151			0	0	0	0	0	0	1 (X ₁ ^Y)	0	0	0	0	0	0	0	0	0																																																								
SD157			0	1 (X ₇ ^Y)	0	0	0	0	0	0	0	0	0	0	0	0	0	0																																																								
SD151																																																																										
SD152																																																																										
SD153																																																																										
SD154																																																																										
SD155																																																																										
SD156																																																																										
SD157																																																																										

*1 The module whose first 5 digits of serial No. is "07032" or later.

*2 The module whose first 5 digits of serial No. is "10042" or later.

*3 This applies to Universal model QCPUs except for the Built-in Ethernet port QCPU.

*4 The module whose first 5 digits of serial No. is "10102" or later.

(2) System information

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU
SD200	Status of switch	Status of CPU switch	• This register stores the status of the CPU module switches in the following bit pattern. b15 to b12 b11 to b8 b7 to b4 b3 to b0 3)Empty2)1) 1): CPU switch status 0: RUN 1: STOP 2: L.CLR 2): Memory card switch Always OFF 3): DIP switch b8 through b12 correspond to SW1 through SW5 of system setting switch 1. 0: OFF, 1: ON. b13 through b15 are empty.	S (Every END processing)	New	Qn(H) QnPH QnPRH
			This register stores the status of the CPU module switches in the following bit pattern. b15 to b8 b7 to b4 b3 to b0 Empty2)1) 1): CPU switch status 0: RUN 1: STOP 2): Memory card switch Always OFF			Q00J/Q00/ Q01
			This register stores the status of the CPU module switches in the following bit pattern. b15 to b8 b7 to b4 b3 to b0 Empty2)1) 1): CPU switch status 0: RUN 1: STOP 2): Memory card switch Always OFF	S (when RUN/ STOP/RESET switch changed)		QnU
			This register stores the status of the CPU module switches in the following bit pattern. b15 to b6 b5 b4 b3 to b0 Empty2)1) 1): CPU switch status 0: RUN 1: STOP 2): SD memory card switch 0: Not usable 1: Usable			LCPU

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU
SD201	LED status	Status of CPU-LED	- This register stores the LED status information on the CPU module in the following bit pattern. 0 is off, 1 is on, and 2 is flicker. 1): RUN, 2): ERR., 3): USER^{*1}, 4): BAT.^{*1}, 5): BOOT^{*1}, 6): Empty^{*1}, 7): Empty^{*1}, 8): MODE^{*1} (0: Off, 1: Green, 2: Orange) ^{*1} For the Basic model QCPU, 3) to 8) are left empty.	S (Status change)	New	Q00J/Q00/ Q01 Qn(H) QnPH QnPRH
			- This register stores the LED status information on the CPU module in the following bit pattern. 0 is off, 1 is on, and 2 is flicker. 1): RUN, 2): ERR., 3): USER, 4): BAT., 5): BOOT^{*1}, 6): Empty, 7): Empty, 8): MODE ^{*1} For the Q00UJCPU, Q00UCPU, and Q01UCPU, 5) is left empty.			QnU
			- This register stores the LED status information on the CPU module in the following bit pattern. 0 is off, 1 is on, and 2 is flicker. 1): RUN, 2): ERR., 3): USER, 4): BAT., 5): Empty, 6): Empty, 7): I/O ERR., 8): MODE			LCPU
SD202	LED off command	Bit pattern of LED that is turned off	- By specifying the LEDs in this register and turning SM202 from off to on, the LEDs can be turned off. The USER and BOOT^{*1} LEDs can be specified. - The LED to be turned off can be specified in the following bit pattern. (Setting "1" turns off the LED and setting "0" does not turn off the LED.) ^{*1} For the Q00UJCPU, Q00UCPU, and Q01UCPU, the BOOT LED cannot be specified.	U		Qn(H) QnPH QnPRH QnU
			- By specifying the LEDs in this register and turning SM202 from off to on, the LEDs can be turned off. The USER LED can be specified. - The LED to be turned off can be specified in the following bit pattern. ((Setting "1" turns off the LED and setting "0" does not turn off the LED.)			LCPU

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU				
SD203	Operating status of CPU	Operating status of CPU	This register stores the operating status of the CPU module in the following bit pattern. b15 to b12 b11 to b8 b7 to b4 b3 to b0 2) 1) <table><tr><td>1): Operating status of CPU</td><td>0: RUN 2: STOP 3: PAUSE</td></tr><tr><td>2): STOP/PAUSE cause *1</td><td>0: Switch 1: Remote contact 2: Remote operation from programming tool/ serial communication, etc. 3: Internal program instruction 4: Error</td></tr></table> *1 The item detected first is stored. (However, for the Universal model QCPU and LCPU, the latest cause after operation status change is stored.)	1): Operating status of CPU	0: RUN 2: STOP 3: PAUSE	2): STOP/PAUSE cause *1	0: Switch 1: Remote contact 2: Remote operation from programming tool/ serial communication, etc. 3: Internal program instruction 4: Error	S (Every END processing)	D9015 format change	QCPU LCPU
1): Operating status of CPU	0: RUN 2: STOP 3: PAUSE									
2): STOP/PAUSE cause *1	0: Switch 1: Remote contact 2: Remote operation from programming tool/ serial communication, etc. 3: Internal program instruction 4: Error									

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU
SD204	LED display color	CPU-LED display color	The LED display color of the LED status shown in SD201 1) to 8). b15 b12 b11 b8 b7 b4 b3 b0 1)RUN LED 0: OFF 1: Green 2)ERR. LED 0: OFF 1: Red 3)USER LED 0: OFF 1: Red 4)BAT. LED 0: OFF 1: Yellow 2: Green 5)BOOT LED *1 0: OFF 1: Green 6)Empty 7)Empty 8)MODE LED 0: OFF 1: Green <td rowspan="2">S (Status change)</td> <td rowspan="2">New</td> <td>QnU</td>	S (Status change)	New	QnU
			The LED display color of the LED status shown in SD201 1) to 8). b15 b12 b11 b8 b7 b4 b3 b0 1)RUN LED 0: OFF 1: Green 2)ERROR LED 0: OFF 1: Red 3)USER LED 0: OFF 1: Red 4)BAT. LED 0: OFF 1: Yellow 2: Green 5)Empty 6)Empty 7)I/O ERR. LED 0: OFF 1: Red 8)MODE LED 0: OFF 1: Green <td>LCPU</td>			LCPU

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU															
SD207	LED display priority ranking	Priorities 1 to 4	- The priority of the LED indication in the case of an error is set by a cause number. (For the Basic model QCPU, only the annunciator (cause number 7) is available.) - For the Universal model QCPU and LCPU, specify whether to enable or disable LED indication of the error that has priority when an error occurs. - The setting areas for priorities are as follows:	U	D9038	Q00J/ Q00/ Q01 * 1 Qn(H) QnPH QnPRH QnU LCPU															
SD208		Priorities 5 to 8	b15 to b12b11 to b8b7 to b4b3 to b0 <table><tr><td>SD207</td><td>Priority 4</td><td>Priority 3</td><td>Priority 2</td><td>Priority 1</td></tr><tr><td>SD208</td><td>Priority 8</td><td>Priority 7</td><td>Priority 6</td><td>Priority 5</td></tr><tr><td>SD209</td><td>Priority 12</td><td>Priority 11</td><td>Priority 10</td><td>Priority 9</td></tr></table> (Priority 11 is valid when Redundant CPU is used.) (Priority 12 is valid when LCPU is used.)		SD207		Priority 4	Priority 3	Priority 2	Priority 1	SD208	Priority 8	Priority 7	Priority 6	Priority 5	SD209	Priority 12	Priority 11	Priority 10	Priority 9	D9039 format change
SD207		Priority 4	Priority 3		Priority 2		Priority 1														
SD208	Priority 8	Priority 7	Priority 6	Priority 5																	
SD209	Priority 12	Priority 11	Priority 10	Priority 9																	
SD209	Priorities 9 to 12	[Default value] SD207 = 4321_H (0000_H for Basic model QCPU) SD208 = 8765_H (0700_H for Basic model QCPU) (0765_H for Redundant CPU) SD209 = 00A9_H (0000_H for Basic model QCPU) (0B09_H for Redundant CPU, CBA9_H for LCPU) - No indication if "0" is stored. - For the Basic model QCPU, the ERR. LED lights up upon turn-on of the annunciator, if "7" is stored in any of Priorities 1 to 11. - For the Basic model QCPU, the ERR. LED does not light up upon turn-on of the annunciator, if "7" is not stored in any of Priorities 1 to 11. However, even if "7" is stored, the error, which makes CPU module operation stop (including the error that is specified with a parameter), will be indicated with the LED without condition.	New																		

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU																								
SD210	Clock data	Clock data (year, month)	This register stores the year (last two digits) and month in BCD as shown below. b15 to b12b11 to b8b7 to b4b3 to b0 YearMonth Example: July, 1993 9307_H	S(Request) /U	D9025	QCPU LCPU																								
SD211	Clock data	Clock data (day, hour)	This register stores the day and hour in BCD as shown below. b15 to b12b11 to b8b7 to b4b3 to b0 DayHour Example: 31st, 10 a.m. 3110_H		D9026																									
SD212	Clock data	Clock data (minute, second)	This register stores the minute and the second in BCD as shown below. b15 to b12b11 to b8b7 to b4b3 to b0 MinuteSecond Example: 35 min, 48 s 3548_H		D9027																									
SD213	Clock data	Clock data (higher digits of year, day of week)	This register stores the year (first two digits) and day of week in BCD as shown below. b15 to b12b11 to b8b7 to b4b3 to b0 Higher digits of year (19 or 20)Day of the week Example: 1993, Friday 1905_H <table><tr><td>0</td><td>Sunday</td></tr><tr><td>1</td><td>Monday</td></tr><tr><td>2</td><td>Tuesday</td></tr><tr><td>3</td><td>Wednesday</td></tr><tr><td>4</td><td>Thursday</td></tr><tr><td>5</td><td>Friday</td></tr><tr><td>6</td><td>Saturday</td></tr></table>		0		Sunday	1	Monday	2	Tuesday	3	Wednesday	4	Thursday	5	Friday	6	Saturday	D9028										
0	Sunday																													
1	Monday																													
2	Tuesday																													
3	Wednesday																													
4	Thursday																													
5	Friday																													
6	Saturday																													
SD220	LED display data	LED display data	• LED display ASCII data (16 characters) stored here. For the Basic model QCPU, an error message (up to 16 ASCII characters) is stored. (Including a message for the case the annunciator is on) b15 to b8b7 to b0 <table><tr><td>SD220</td><td>15th character from the right</td><td>16th character from the right</td></tr><tr><td>SD221</td><td>13th character from the right</td><td>14th character from the right</td></tr><tr><td>SD222</td><td>11th character from the right</td><td>12th character from the right</td></tr><tr><td>SD223</td><td>9th character from the right</td><td>10th character from the right</td></tr><tr><td>SD224</td><td>7th character from the right</td><td>8th character from the right</td></tr><tr><td>SD225</td><td>5th character from the right</td><td>6th character from the right</td></tr><tr><td>SD226</td><td>3rd character from the right</td><td>4th character from the right</td></tr><tr><td>SD227</td><td>1st character from the right</td><td>2nd character from the right</td></tr></table> • For the Basic model QCPU, Universal model QCPU or LCPU, HMI data at the time of CHK instruction execution are not stored.	SD220	15th character from the right	16th character from the right	SD221	13th character from the right	14th character from the right	SD222	11th character from the right	12th character from the right	SD223	9th character from the right	10th character from the right	SD224	7th character from the right	8th character from the right	SD225	5th character from the right	6th character from the right	SD226	3rd character from the right	4th character from the right	SD227	1st character from the right	2nd character from the right	S (When changed)	New	QCPU LCPU
SD220				15th character from the right	16th character from the right																									
SD221				13th character from the right	14th character from the right																									
SD222				11th character from the right	12th character from the right																									
SD223				9th character from the right	10th character from the right																									
SD224				7th character from the right	8th character from the right																									
SD225				5th character from the right	6th character from the right																									
SD226	3rd character from the right	4th character from the right																												
SD227	1st character from the right	2nd character from the right																												
SD221																														
SD222																														
SD223																														
SD224																														
SD225																														
SD226																														
SD227																														
SD235	Module to which online module change is being performed	The header I/O number of the module to which online module change is being performed /10 _H	10 _H is added to the value of the header I/O number of which the online module change is being performed.	S (During online module change)		QnPH QnPRH																								

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU
SD240	Base mode	0: Automatic mode 1: Detail mode	Stores the base mode.	S (Initial)	New	QCPU
SD241	Extension stage number	0: Main base only 1 to 7: Extension stage number	Stores the maximum number of the extension bases being installed.			
SD242	A/Q base differentiation	Base type differentiation 0: QA**B is installed (A mode) 1: Q**B is installed (Q mode)				Qn(H) QnPH QnPRH
	Installed Q base presence/absence	Base type differentiation 0: Base not installed 1: Q**B is installed				Q00J/Q00/ Q01
	Installed Q base presence/absence	Base type differentiation 0: Base not installed 1: Q**B is installed	• For the Q00UJCPU, the bits for the third to seventh extension bases are fixed to "0". • For the Q00UCPU, Q01UCPU, and Q02UCPU, the bits for the fifth to seventh extension bases are fixed to "0".			QnU

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU										
SD243	No. of base slots	No. of base slots	- The number of slots used is stored in the area corresponding to each base unit as shown below. b15 to b12b11 to b8b7 to b4b3 to b0 <table><tr><td>SD243</td><td>Extension 3</td><td>Extension 2</td><td>Extension 1</td><td>Main</td></tr><tr><td>SD244</td><td>Extension 7</td><td>Extension 6</td><td>Extension 5</td><td>Extension 4</td></tr></table> - For the Q00UJCPU, the bits for the third to seventh extension bases are fixed to "0". - For the Q00UCPU, Q01UCPU, and Q02UCPU, the bits for the fifth to seventh extension bases are fixed to "0".	SD243	Extension 3	Extension 2	Extension 1	Main	SD244	Extension 7	Extension 6	Extension 5	Extension 4	S (Initial)		Qn(H) QnPH QnPRH QnU
SD243			Extension 3	Extension 2	Extension 1	Main										
SD244	Extension 7		Extension 6	Extension 5	Extension 4											
SD244																
SD243	No. of base slots (Operation status)	The number of slots used is stored in the area corresponding to each base unit as shown below. (The number of slots set in the parameter setting.) b15 to b12b11 to b8b7 to b4b3 to b0 <table><tr><td>SD243</td><td>Extension 3</td><td>Extension 2</td><td>Extension 1</td><td>Main</td></tr><tr><td>SD244</td><td>Fixed to 0</td><td>Fixed to 0</td><td>Fixed to 0</td><td>Extension 4</td></tr></table>	SD243	Extension 3	Extension 2	Extension 1	Main	SD244	Fixed to 0	Fixed to 0	Fixed to 0	Extension 4		Q00J/Q00/ Q01		
SD243		Extension 3	Extension 2	Extension 1	Main											
SD244	Fixed to 0	Fixed to 0	Fixed to 0	Extension 4												
SD244																
SD245	No. of base slots (Mounting status)		The number of slots where modules are actually mounted is stored in the area corresponding to each base unit as shown below. b15 to b12b11 to b8b7 to b4b3 to b0 <table><tr><td>SD245</td><td>Extension 3</td><td>Extension 2</td><td>Extension 1</td><td>Main</td></tr><tr><td>SD246</td><td>Fixed to 0</td><td>Fixed to 0</td><td>Fixed to 0</td><td>Extension 4</td></tr></table>	SD245	Extension 3	Extension 2	Extension 1	Main	SD246	Fixed to 0	Fixed to 0	Fixed to 0	Extension 4		New	Q00J/Q00 /Q01*1
SD245			Extension 3	Extension 2	Extension 1	Main										
SD246	Fixed to 0		Fixed to 0	Fixed to 0	Extension 4											
SD246																
SD250	Loaded maximum I/O	Loaded maximum I/O No	When SM250 is turned on from off, the first two digits of the number, which is the last I/O number of the mounted modules plus 1, are stored.	S (Request END)		Qn(H) QnPH QnPRH										
			The first two digits of the number, which is the last I/O number of the mounted modules plus 1, are stored.			Q00J/Q00/ Q01 QnU LCPU										
SD254	MELSECNET /10. MELSECNET /H information	Number of modules installed	Indicates the number of mounted MELSECNET/10 modules or MELSECNET/H modules.	S (Initial)		QCPU										
SD255		I/O No.	Indicates the I/O number of mounted MELSECNET/10 module or MELSECNET/H module.													
SD256		Network No.	Indicates the network No. of mounted MELSECNET/10 module or MELSECNET/H module.													
SD257		Group number	Indicates the group No. of mounted MELSECNET/10 module or MELSECNET/H module.													
SD258		Station No.	Indicates the station No. of mounted MELSECNET/10 module or MELSECNET/H module.				Qn(H) QnPH QnPRH QnU*2									
SD259		Standby informa- tion	In the case of standby stations, the module number of the standby station is stored. (1 to 4)													
SD260 to SD264		Information from 2nd module	Data configuration is the same as that of the 1st module (SD255 to SD259).					Qn(H) QnPH QnPRH QnU*3								
SD265 to SD269		Information from 3rd module	Data configuration is the same as that of the 1st module (SD255 to SD259).													
SD270 to SD274	Information from 4th module	Data configuration is the same as that of the 1st module (SD255 to SD259).														

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU
SD280	CC-Link error	Error detection status	This register stores error detection status in the following bit pattern. Information of 3 Information of 2 Information of 1 b15 to b12 b11 to b8 b7 to b4 b3 to b0 Empty 1st module 2nd module 3rd module 4th module 1): When Xn0 of a mounted CC-Link module turns on, the corresponding bit is set to 1 (on). 2): When either Xn1 or XnF of a mounted CC-Link module turns off, the corresponding bit is set to 1 (on). 3) When a mounted CC-Link module is not able to communicate with the CPU module, the corresponding bit is set to 1 (on). The above modules are numbered in order of the head I/O numbers. (However, the one where parameter setting has not been made is not counted.)	S (Error)	New	Qn(H) QnPH QnPRH
SD281			This register stores error detection status in the following bit pattern. Information of 3 Information of 2 Information of 1 b15 to b12 b11 to b8 b7 to b4 b3 to b0 Empty 5st module 6nd module 7rd module 8th module 1): When Xn0 of a mounted CC-Link module turns on, the corresponding bit is set to 1 (on). 2): When either Xn1 or XnF of the mounted CC-Link module turns off, the corresponding bit is set to 1 (on). 3): When a mounted CC-Link module is not able to communicate with the CPU module, the corresponding bit is set to 1 (on). The above modules are numbered in order of the head I/O numbers. (However, the one where parameter setting has not been made is not counted.)			Qn(H) ^{*4} QnPH ^{*4} QnPRH ^{*5}
SD286	Device assignment	Points assigned to M (for extension)	- The number of points assigned to M is stored with 32 bits. - The number of 32k or less points can be assigned to M.	S (Initial)		QnU ^{*6} LCPU
SD287		Points assigned to B (for extension)	- The number of points assigned to B is stored with 32 bits. - The number of 32k or less points can be assigned to B.			
SD288						
SD289						

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU
SD290	Device assignment (Same as parameter contents)	Number of points assigned for X	Stores the number of points currently set for X devices.	S (Initial)	New	QCPU LCPU
SD291		Number of points assigned for Y	Stores the number of points currently set for Y devices.			
SD292		Number of points assigned for M	Stores the number of points currently set for M devices.			
SD293		Number of points assigned for L	Stores the number of points currently set for L devices.			
SD294		Number of points assigned for B	Stores the number of points currently set for B devices.			
SD295		Number of points assigned for F	Stores the number of points currently set for F devices.			
SD296		Number of points assigned for SB	Stores the number of points currently set for SB devices.			
SD297		Number of points assigned for V	Stores the number of points currently set for V devices.			
SD298		Number of points assigned for S	Stores the number of points currently set for S devices.			
SD299		Number of points assigned for T	Stores the number of points currently set for T devices.			
SD300		Number of points assigned for ST	Stores the number of points currently set for ST devices.			
SD301		Number of points assigned for C	Stores the number of points currently set for C devices.			
SD302		Number of points assigned for D	Stores the number of points currently set for D devices.			
SD303		Number of points assigned for W	Stores the number of points currently set for W devices.			
SD304		Number of points assigned for SW	Stores the number of points currently set for SW devices.			

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU	
SD305	Device assignment (Index register)	16 bit modification Number of points assigned for Z	Stores the number of points of index register (Z) used for the 16-bit modification area. (Depending on the index modification setting for ZR in the parameter setting.)	S (Initial)		QnU LCPU	
SD306	Device assignment (Same as parameter contents)	Number of points assigned for ZR (for extension)	The number of points for ZR is stored (except the number of points of extended data register (D) and extended link register (W)). The number of points assigned to ZR is stored into this register only when 1k point or more is set for the extended data register (D) or extended link register (W).			QnU*7 LCPU	
SD307							
SD308	Device assignment (assignment including the number of points set to the extended data register (D) and extended link register (W))	Number of points assigned for D (for inside + for extension)	The total points of the data register (D) in the internal device memory area and the extended data register (D) are stored as a 32-bit binary value.				
SD309							
SD310							
SD311		Number of points assigned for W (for inside + for extension)	The total points of the link register (W) in the internal device memory area and the extended link register (W) are stored as a 32-bit binary value.				
SD315	Time reserved for communication processing	Time reserved for communication processing	• This register specifies the amount of processing time for communication with a programming tool or another module. • The greater the value specified is, the shorter the response time for communication with another (such as a programming tool or serial communication module) is. However, scan time will increase by the specified time. • Setting range: 1 to 100ms A setting outside the above range is regarded as no setting.	U	New	Q00J/Q00/ Q01 Qn(H) QnPH QnPRH	
SD329	Online change (inactive block) target block number	SFC block number	• While online change (inactive block) is executed (SM329 is on.), this register stores the target SFC block number. • In other than the above status, this register stores FFFF_H.	S (Status change)		QnU*8	
SD340	Ethernet information	No. of modules installed	Indicates the number of mounted Ethernet module.	S (Initial)		QCPU	
SD341		Information of 1st module	I/O No.				Indicates I/O No. of mounted Ethernet module
SD342			Network No.				Indicates network No. of mounted Ethernet module
SD343			Group No.				Indicates group No. of the mounted Ethernet module.
SD344			Station No		Indicates station No. of mounted Ethernet module		
SD345 to SD346			Empty		Empty (The IP address of the 1st Ethernet module is stored in the buffer memory.)	Qn(H) QnPH QnPRH QnU*2	
SD347			Empty		Empty (An error code of the 1st Ethernet module is read with the ERRRD instruction.)		
SD348 to SD354	Ethernet information	Information from 2nd module	Data configuration is the same as that of the 1st module (SD341 to SD347).		Qn(H) QnPH QnPRH QnU*3		
SD355 to SD361		Information from 3rd module	Data configuration is the same as that of the 1st module (SD341 to SD347).				
SD362 to SD368		Information from 4th module	Data configuration is the same as that of the 1st module (SD341 to SD347).				

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU
SD380	Ethernet instruction reception status	Instruction reception status of 1st module	b15b8b7b6b5b4b3b2b1b0 0 Not used Instruction reception status of channel 1 Instruction reception status of channel 2 Instruction reception status of channel 3 Instruction reception status of channel 4 Instruction reception status of channel 5 Instruction reception status of channel 6 Instruction reception status of channel 7 Instruction reception status of channel 8 ON: Received (Channel is being used.) OFF: Not received (Channel is not used.)	S (Instruction execution)	New	QnPRH
SD381		Instruction reception status of 2nd module	Data configuration is the same as that of the 1st module (SD380).			
SD382		Instruction reception status of 3rd module	Data configuration is the same as that of the 1st module (SD380).			
SD383		Instruction reception status of 4th module	Data configuration is the same as that of the 1st module (SD380).			

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU
SD393	Multiple CPU system information	Number of multiple CPUs	The number of CPU modules that comprise the multiple CPU system is stored. (1 to 3, Empty also included)	S (Initial)	New	Q00/Q01 ^{*1} QnU
SD394		CPU mounting information	This register stores information on the CPU module types of CPU No.1 to No.3 and whether or not the CPU modules are mounted.			Q00/Q01 ^{*1}
SD395		Multiple CPU number	In a multiple CPU system configuration, the CPU number of the host CPU is stored. CPU No. 1: 1, CPU No. 2: 2, CPU No. 3: 3, CPU No. 4: 4	S (END processing error)	New	Q00/Q01 ^{*1} Qn(H) ^{*1} QnPH QnU
SD396		No. 1 CPU operation status	The operation information of each CPU No. is stored. (The information on the number of multiple CPUs indicated in SD393 is stored.)			Q00/Q01 ^{*1} QnU
SD397		No. 2 CPU operation status				Q00/Q01 ^{*1} QnU ^{*7}
SD398		No. 3 CPU operation status				
SD399		No. 4 CPU operation statu				QnU ^{*3}

*1 Function version is B or later.

*2 The Universal model QCPU except the Q00UJCPU, Q00UCPU, and Q01UCPU.

*3 The Universal model QCPU except the Q00UJCPU, Q00UCPU, Q01UCPU, and Q02UCPU.

*4 The module whose first 5 digits of serial No. is "08032" or later.

*5 The module whose first 5 digits of serial No. is "09012" or later.

*6 The module whose first 5 digits of serial No. is "10042" or later.

*7 The Universal model QCPU except the Q00UJCPU.

*8 This applies when the first five digits of the serial number is "12052" or later.

(3) System clock/counter

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU	
SD412	1 second counter	Number of counts in 1-second units	- This register is incremented by 1 for each second after the CPU module is set to RUN. - Count repeats from 0 to 32767 to -32768 to 0	S (Status change)	D9022	QCPU LCPU	
SD414	2n second clock setting	2n second clock units	- Stores value n of 2n second clock (Default is 30) - Setting can be made between 1 and 32767	U	New		
SD415	2nms clock setting	2nms clock units	- Stores value n of 2nms clock (Default is 30) - Setting can be made between 1 and 32767			S (Every END processing)	
SD420	Scan counter	Number of counts in each scan	- This register is incremented by 1 for each scan of a scan execution type program after the CPU module is set to RUN. (Not incremented for each scan of an initial execution type program.) - Count repeats from 0 to 32767 to -32768 to 0	S (Every END processing)			Q00J/Q00/Q01
			- This register is incremented by 1 for each scan after the CPU module is set to RUN. - Count repeats from 0 to 32767 to -32768 to 0				
SD430	Low speed scan counter	Number of counts in each scan	- This register is incremented by 1 for each scan of a low-speed execution type program after the CPU module is set to RUN. - Count repeats from 0 to 32767 to -32768 to 0 - Used only for low speed execution type programs				

(4) Scan information

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU
SD500	Execution program No.	Program No. in execution	Program number of program currently being executed is stored as BIN value.	S (Status change)	New	Qn(H) QnPH QnPRH QnU LCPU
SD510	Low speed excution type program No.	Low speed execution type program No. in execution	- Program number of low speed excution type program No. currently being executed is stored as BIN value. - Enabled only when SM510 is ON.			Qn(H) QnPH
SD520	Current scan time	Current scan time (ms value)	This register stores the current scan time. (The time is measured in increments of 100μs (in icrements of 1μs for the Universal model QCPU and LCPU).) SD520: Stores a ms value (storage range: 0 to 65535). SD521: Stores a μs value (storage range: 0 to 900 (0 to 999 for the Universal model QCPU and LCPU)). Example: When the current scan time is 23.6ms, the following values are stored: - SD520 = 23 - SD521 = 600	S (Every END processing)	D9018 format change	QCPU LCPU
SD521		Current scan time (μs value)				
SD522	Initial scan time	Initial scan time (ms value)	This register stores the scan time of an initial execution type program. (The time is measured in increments of 100μs (in icrements of 1μs for the Universal model QCPU and LCPU).) SD522: Stores a ms value (storage range: 0 to 65535). SD523: Stores a μs value (storage range: 0 to 900 (0 to 999 for the Universal model QCPU and LCPU)).	S (First END processing)	New	Qn(H) QnPH QnPRH QnU LCPU
SD523		Initial scan time (μs value)				
SD524	Minimum scan time	Minimum scan time (ms value)	This register stores the minimum scan time. (The time is measured in increments of 100μs.) SD524: Stores a ms value (storage range: 0 to 65535). SD525: Stores a μs value (storage range: 0 to 900)			Q00J/Q00/ Q01
SD525		Minimum scan time (μs value)				
SD526	Maximum scan time	Maximum scan time (ms value)	This register stores the maximum scan time. (The time is measured in increments of 100μs.) SD526: Stores a ms value (storage range: 0 to 65535). SD527: Stores a μs value (storage range: 0 to 900)			Q00J/Q00/ Q01
SD527		Maximum scan time (μs value)				
SD524	Minimum scan time	Minimum scan time (ms value)	This register stores the minimum scan time except that of an initial execution type program. (The time is measured in increments of 100μs (in icrements of 1μs for the Universal model QCPU and LCPU).) SD524: Stores a ms value (storage range: 0 to 65535). SD525: Stores a μs value (storage range: 0 to 900 (0 to 999 for the Universal model QCPU and LCPU)).	S (Every END processing)	D9017 format change	Qn(H) QnPH QnPRH QnU LCPU
SD525		Minimum scan time (μs value)			New	
SD526	Maximum scan time	Maximum scan time (ms value)	This register stores the maximum scan time excluding the scan time of an initial execution type program. (The time is measured in increments of 100μs (in icrements of 1μs for the Universal model QCPU and LCPU).) SD526: Stores a ms value (storage range: 0 to 65535). SD527: Stores a μs value (storage range: 0 to 900 (0 to 999 for the Universal model QCPU and LCPU)).		D9019 format change	Qn(H) QnPH QnPRH QnU LCPU
SD527		Maximum scan time (μs value)			New	

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU
SD528	Current scan time for low speed execution type programs	Current scan time (ms value)	This register stores the current scan time of a low-speed execution type program.(The time is measured in increments of 100μs.) SD528: Stores a ms value (storage range: 0 to 65535) SD529: Stores a μs value (storage range: 0 to 900)	S (Every END processing)	New	Qn(H) QnPH
SD529		Current scan time (μs value)				
SD532	Minimum scan time (ms value)	This register stores the minimum scan time of a low-speed execution type program.(The time is measured in increments of 100μs.) SD532: Stores a ms value (storage range: 0 to 65535) SD533: Stores a μs value (storage range: 0 to 900)				
SD533	Minimum scan time (μs value)					
SD534	Maximum scan time for low speed execution type programs	Maximum scan time (ms value)	This register stores the maximum scan time excluding the time taken to the first scan of a low-speed execution type program. (The time is measured in increments of 100μs.) SD534: Stores a ms value (storage range: 0 to 65535) SD535: Stores a μs value (storage range: 0 to 900)			
SD535		Maximum scan time (μs value)				
SD540	END processing time	END processing time (ms value)	Stores the time from the end of a scan program to the start of the next scan. (The time is measured in increments of 100μs.) SD540: Stores a ms value (storage range: 0 to 65535) SD541: Stores a μs value (storage range: 0 to 900)			Q00J/Q00/ Q01
SD541		END processing time (μs value)				
SD540	END processing time	END processing time (ms value)	Stores the time from the end of a scan execution type program to the start of the next scan. (The time is measured in increments of 100μs (in icrements of 1μs for the Universal model QCPU and LCPU).) SD540: Stores a ms value (storage range: 0 to 65535) SD541: Stores a μs value (storage range: 0 to 900 (0 to 999 for the Universal model QCPU and LCPU)).			Qn(H) QnPH QnPRH QnU LCPU
SD541		END processing time (μs value)				
SD542	Constant scan wait time	Constant scan wait time (ms value)	This register stores wait time for constant scan. (The time is measured in increments of 100μs (in icrements of 1μs for the Universal model QCPU and LCPU).) SD542: Stores a ms value (storage range: 0 to 65535) SD543: Stores a μs value (storage range: 0 to 900 (0 to 999 for the Universal model QCPU and LCPU)).			QCPU LCPU
SD543		Constant scan wait time (μs value)				

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU	
SD544	Cumulative execution time for low speed	Cumulative execution time for low speed execution type programs (ms value)	- Stores the cumulative execution time of a low-speed execution type program. (The time is measured in increments of 100μs.) SD544: Stores a ms value (storage range: 0 to 65535) SD545: Stores a μs value (storage range: 0 to 900) - Cleared to 0 after the end of one scan of a low-speed execution type program.	S (Every END processing)	New	Qn(H) QnPH	
SD545	execution type programs	Cumulative execution time for low speed execution type programs (μs value)					
SD546	Execution time for low speed	Execution time for low speed execution type programs (ms value)	- Stores the execution time of a low-speed execution type program in one scan. (The time is measured in increments of 100μs.) SD546: Stores a ms value (storage range: 0 to 65535) SD547: Stores a μs value (storage range: 0 to 900) - Stored every scan.				
SD547	execution type programs	Execution time for low speed execution type programs (μs value)					
SD548	Scan program execution time	Scan program execution time (ms value)	- Stores the execution time of a scan program in one scan. (The time is measured in increments of 100μs (in increments of 1μs for the Universal model QCPU and LCPU).) SD548: Stores a ms value (storage range: 0 to 65535) SD549: Stores a μs value (storage range: 0 to 900 (0 to 999 for the Universal model QCPU and LCPU)). - Stored every scan.			Q00J/Q00/ Q01 QnU LCPU	
SD549		Scan program execution time (μs value)					
SD548	Scan execution type program execution time	Scan execution type program execution time (ms value)	- Stores the execution time of a scan execution type program in one scan. (The time is measured in increments of 100μs.) SD548: Stores a ms value (storage range: 0 to 65535) SD549: Stores a μs value (storage range: 0 to 900) - Stored every scan.				Qn(H) QnPH QnPRH
SD549		Scan execution type program execution time (μs value)					
SD550	Service interval measuremen t module	Unit/module No.	Sets I/O number for module that measures service interval.	U			
SD551	Service interval time	Module service interval (ms value)	This register stores the service interval of a module specified by SD550 when SM551 is turned on. (The time is measured in increments of 100μs.) SD551: Stores a ms value (storage range: 0 to 65535) SD552: Stores a μs value (storage range: 0 to 900)	S (Request)			
SD552		Module service interval (μs value)					

(5) Drive information

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU		
SD600	Memory card typs	Memory card typs	This register stores a value indicating the type of used memory card in the following bit pattern. b15 to b8 b7 to b4 b3 to b0 0 Drive 1 (RAM) type *1 0: Does not exist 1: SRAM card Drive 2 (ROM) type *1 0: Does not exist (1: SRAM) 2: ATA card 3: Flash card *1: For the Q00UJCPU, Q00UCPU, and Q01UCPU, the drive 1 (RAM) type and drive 2 (ROM) type are fixed at "0".	S (Initial and card removal)		Qn(H) QnPH QnPRH QnU		
			This register stores a value indicating the type of used memory card in the following bit pattern. b15 to b8 b7 to b4 b3 to b0 0 Drive 1 (RAM) type 0: Does not exist (Fixed to 0) Drive 2 (ROM) type 0: Does not exist 4: SD memory card			LCPU		
SD602	Drive 1 (Memory card RAM) capacity	Drive 1 capacity	This register stores the drive 1 storage capacity (unit: 1K byte).(Free space value after formatting is stored.)		New	Qn(H) QnPH QnPRH QnU*2		
SD603	Drive 2 (Memory card ROM) capacity	Drive 2 capacity	This register stores the drive 2 storage capacity (unit: 1K byte).*1 *1 For the Q2MEM-8MBA, a value stored to this register depends on the product control number of the ATA card. For details, refer to the following.  User's Manual (Hardware Design, Maintenance and Inspection) for the CPU module used					
SD604	Memory card use conditions	Memory card use conditions	This register stores the usage status of a memory card in the following bit pattern. (Each bit is on while the memory card is being used.) <table><tr><td>b0 : Boot operation (QBT) b1 : Parameters (QPA) b2 : Device comments (QCD) b3 : Device initial value (QDI) b4 : File register (QDR) b5 : Sampling trace (QTD) b6 : Not used b7 : Not used</td><td>b8 : Not used b9 : CPU fault history (QFD) b10 : Not used b11 : Local device (QDL) b12 : Not used b13 : Not used b14 : Not used b15 : Not used</td></tr></table>	b0 : Boot operation (QBT) b1 : Parameters (QPA) b2 : Device comments (QCD) b3 : Device initial value (QDI) b4 : File register (QDR) b5 : Sampling trace (QTD) b6 : Not used b7 : Not used	b8 : Not used b9 : CPU fault history (QFD) b10 : Not used b11 : Local device (QDL) b12 : Not used b13 : Not used b14 : Not used b15 : Not used	S (Status change)		Qn(H) QnPH QnPRH
b0 : Boot operation (QBT) b1 : Parameters (QPA) b2 : Device comments (QCD) b3 : Device initial value (QDI) b4 : File register (QDR) b5 : Sampling trace (QTD) b6 : Not used b7 : Not used	b8 : Not used b9 : CPU fault history (QFD) b10 : Not used b11 : Local device (QDL) b12 : Not used b13 : Not used b14 : Not used b15 : Not used							

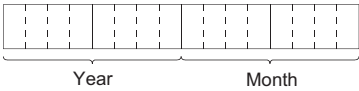
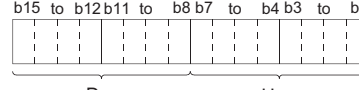
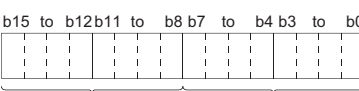
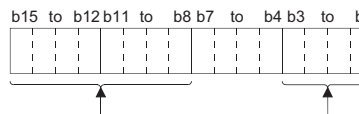
Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU
SD620	Drive 3/4 typs	Drive 3/4 typs	This register stores the usage status of drives 3 and 4 in the following bit pattern. Drive 3 (Standard RAM) type 0: Absent 1: Present Drive 4 (Standard ROM) type "3 (FLASH ROM)"	S (Initial)	New	Q00J/Q00/ Q01
			This register stores the usage status of drives 3 and 4 in the following bit pattern. Drive 3 (Standard RAM) type Fixed to 1 *1 Drive 4 (Standard ROM) type Fixed to 3 *1 For the Q00UJCPU, the drive 3 (Standard RAM) type is fixed at "0".			Qn(H) QnPH QnPRH QnU LCPU
SD622	Drive 3 (Standard RAM) capacity	Drive 3 capacity	This register stores the drive 3 storage capacity (unit: 1K byte).	S (Initial)	New	Q00J/Q00/ Q01
			This register stores the drive 3 storage capacity (unit: 1K byte). (Free space value after formatting is stored.)			Qn(H) QnPH QnPRH QnU LCPU
SD623	Drive 4 (Standard ROM) capacity	Drive 4 capacity	This register stores the drive 4 storage capacity (unit: 1K byte).			Q00J/Q00/ Q01
			This register stores the drive 4 storage capacity (unit: 1K byte). (Free space value after formatting is stored.)			Qn(H) QnPH QnPRH QnU LCPU

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU																
SD624	Drive 3/4 use conditions	Drive 3/4 use conditions	This register stores the usage status of the drives 3 and 4 in the following bit pattern. b15 to b5 b4 to b0 0 0 0 0 0 0 Boot operation (QBT) 0: Not used 1: In use File register (QDR) 0: Not used 1: In use	S (Status change)	New	Q00J/Q00/ Q01																
			This register stores the usage status of the drives 3 and 4 in the following bit pattern. (Each bit is on while the corresponding drive is being used.) <table><tr><td>b0 : Boot operation (QBT)</td><td>b8 : Not used</td></tr><tr><td>b1 : Parameters (QPA)</td><td>b9 : CPU fault history (QFD)</td></tr><tr><td>b2 : Device comments (QCD)</td><td>b10 : SFC trace (QTS)</td></tr><tr><td>b3 : Device initial value (QDI)</td><td>b11 : Local device (QDL)</td></tr><tr><td>b4 : File register (QDR)</td><td>b12 : Not used</td></tr><tr><td>b5 : Sampling trace (QTD)</td><td>b13 : Not used</td></tr><tr><td>b6 : Not used</td><td>b14 : Not used</td></tr><tr><td>b7 : Not used</td><td>b15 : Not used</td></tr></table>			b0 : Boot operation (QBT)	b8 : Not used	b1 : Parameters (QPA)	b9 : CPU fault history (QFD)	b2 : Device comments (QCD)	b10 : SFC trace (QTS)	b3 : Device initial value (QDI)	b11 : Local device (QDL)	b4 : File register (QDR)	b12 : Not used	b5 : Sampling trace (QTD)	b13 : Not used	b6 : Not used	b14 : Not used	b7 : Not used	b15 : Not used	Qn(H) QnPH QnPRH
			b0 : Boot operation (QBT)			b8 : Not used																
			b1 : Parameters (QPA)			b9 : CPU fault history (QFD)																
b2 : Device comments (QCD)	b10 : SFC trace (QTS)																					
b3 : Device initial value (QDI)	b11 : Local device (QDL)																					
b4 : File register (QDR)	b12 : Not used																					
b5 : Sampling trace (QTD)	b13 : Not used																					
b6 : Not used	b14 : Not used																					
b7 : Not used	b15 : Not used																					
This register stores the usage status of the drives 3 and 4 in the following bit pattern. (Each bit is on while the corresponding drive is being used.) <table><tr><td>b0 : Not used</td><td>b8 : Module error log *2</td></tr><tr><td>b1 : Parameters (QPA)</td><td>b9 : Not used</td></tr><tr><td>b2 : Device comments (QCD)</td><td>b10 : Not used</td></tr><tr><td>b3 : Device initial value (QDI) *1</td><td>b11 : Local device (QDL)</td></tr><tr><td>b4 : File register (QDR)</td><td>b12 : Not used</td></tr><tr><td>b5 : Sampling trace (QTD)</td><td>b13 : Not used</td></tr><tr><td>b6 : Not used</td><td>b14 : Not used</td></tr><tr><td>b7 : Not used</td><td>b15 : Not used</td></tr></table> *1 This bit turns on when the writing of initial device values is started and turns off at the completion. *2 This bit can be used when the first five digits of the serial No. is "11043" or later.	b0 : Not used	b8 : Module error log *2	b1 : Parameters (QPA)	b9 : Not used	b2 : Device comments (QCD)	b10 : Not used	b3 : Device initial value (QDI) *1	b11 : Local device (QDL)	b4 : File register (QDR)	b12 : Not used	b5 : Sampling trace (QTD)	b13 : Not used	b6 : Not used	b14 : Not used	b7 : Not used	b15 : Not used	QnU					
b0 : Not used	b8 : Module error log *2																					
b1 : Parameters (QPA)	b9 : Not used																					
b2 : Device comments (QCD)	b10 : Not used																					
b3 : Device initial value (QDI) *1	b11 : Local device (QDL)																					
b4 : File register (QDR)	b12 : Not used																					
b5 : Sampling trace (QTD)	b13 : Not used																					
b6 : Not used	b14 : Not used																					
b7 : Not used	b15 : Not used																					
This register stores the usage status of the drives 3 and 4 in the following bit pattern. (Each bit is on while the corresponding drive is being used.) <table><tr><td>b0 : Not used</td><td>b8 : Module error log</td></tr><tr><td>b1 : Parameters (QPA)</td><td>b9 : Not used</td></tr><tr><td>b2 : Device comments (QCD)</td><td>b10 : Not used</td></tr><tr><td>b3 : Device initial value (QDI) *1</td><td>b11 : Local device (QDL)</td></tr><tr><td>b4 : File register (QDR)</td><td>b12 : Not used</td></tr><tr><td>b5 : Sampling trace (QTD)</td><td>b13 : Data logging setting (QLG) *2</td></tr><tr><td>b6 : Not used</td><td>b14 : Not used</td></tr><tr><td>b7 : Not used</td><td>b15 : Not used</td></tr></table> *1 This bit turns on when the writing of initial device values is started and turns off at the completion. *2 This bit turns on when data logging setting is registered and turns off at the completion or stop of data logging.	b0 : Not used	b8 : Module error log	b1 : Parameters (QPA)	b9 : Not used	b2 : Device comments (QCD)	b10 : Not used	b3 : Device initial value (QDI) *1	b11 : Local device (QDL)	b4 : File register (QDR)	b12 : Not used	b5 : Sampling trace (QTD)	b13 : Data logging setting (QLG) *2	b6 : Not used	b14 : Not used	b7 : Not used	b15 : Not used	LCPU					
b0 : Not used	b8 : Module error log																					
b1 : Parameters (QPA)	b9 : Not used																					
b2 : Device comments (QCD)	b10 : Not used																					
b3 : Device initial value (QDI) *1	b11 : Local device (QDL)																					
b4 : File register (QDR)	b12 : Not used																					
b5 : Sampling trace (QTD)	b13 : Data logging setting (QLG) *2																					
b6 : Not used	b14 : Not used																					
b7 : Not used	b15 : Not used																					

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU																																																	
SD640	File register drive	Drive number:	This register stores the number of a drive storing a file register.*1 *1 For the LCPU, this register is fixed at drive 3.	S (Status change) *4	New	Q00J/Q00/ Q01 Qn(H) QnPH QnPRH QnU*3 LCPU																																																	
SD641	File register file name	File register file name	This register stores the file name of a file register (MAIN.QDR) in ASCII code. <table><tr><td></td><td>b15</td><td>to</td><td>b8</td><td>b7</td><td>to</td><td>b0</td></tr><tr><td>SD641</td><td colspan="3">2nd character (A)</td><td colspan="3">1st character (M)</td></tr><tr><td>SD642</td><td colspan="3">4th character (N)</td><td colspan="3">3rd character (I)</td></tr><tr><td>SD643</td><td colspan="3">6th character ()</td><td colspan="3">5th character ()</td></tr><tr><td>SD644</td><td colspan="3">8th character ()</td><td colspan="3">7th character ()</td></tr><tr><td>SD645</td><td colspan="3">1st character of the extension (Q)</td><td colspan="3">2EH(.)</td></tr><tr><td>SD646</td><td colspan="3">3rd character of the extension (R)</td><td colspan="3">2nd character of the extension (D)</td></tr></table>			b15	to	b8	b7	to	b0	SD641	2nd character (A)			1st character (M)			SD642	4th character (N)			3rd character (I)			SD643	6th character ()			5th character ()			SD644	8th character ()			7th character ()			SD645	1st character of the extension (Q)			2EH(.)			SD646	3rd character of the extension (R)			2nd character of the extension (D)			S (Initial)	Q00J/Q00/ Q01
			b15	to		b8	b7	to	b0																																														
SD641			2nd character (A)			1st character (M)																																																	
SD642			4th character (N)			3rd character (I)																																																	
SD643			6th character ()			5th character ()																																																	
SD644			8th character ()			7th character ()																																																	
SD645	1st character of the extension (Q)			2EH(.)																																																			
SD646	3rd character of the extension (R)			2nd character of the extension (D)																																																			
SD642																																																							
SD643																																																							
SD644			This register stores the file name of the file register selected by the parameter or the QDRSET instruction in ASCII code (with an extension). <table><tr><td></td><td>b15</td><td>to</td><td>b8</td><td>b7</td><td>to</td><td>b0</td></tr><tr><td>SD641</td><td colspan="3">2nd character</td><td colspan="3">1st character</td></tr><tr><td>SD642</td><td colspan="3">4th character</td><td colspan="3">3rd character</td></tr><tr><td>SD643</td><td colspan="3">6th character</td><td colspan="3">5th character</td></tr><tr><td>SD644</td><td colspan="3">8th character</td><td colspan="3">7th character</td></tr><tr><td>SD645</td><td colspan="3">1st character of extension</td><td colspan="3">2EH(.)</td></tr><tr><td>SD646</td><td colspan="3">3rd character of the extension</td><td colspan="3">2nd character of the extension</td></tr></table>		b15	to	b8	b7	to	b0	SD641	2nd character			1st character			SD642	4th character			3rd character			SD643	6th character			5th character			SD644	8th character			7th character			SD645	1st character of extension			2EH(.)			SD646	3rd character of the extension			2nd character of the extension			S (Status change)	Qn(H) QnPH QnPRH QnU*3	
	b15	to	b8	b7	to	b0																																																	
SD641	2nd character			1st character																																																			
SD642	4th character			3rd character																																																			
SD643	6th character			5th character																																																			
SD644	8th character			7th character																																																			
SD645	1st character of extension			2EH(.)																																																			
SD646	3rd character of the extension			2nd character of the extension																																																			
SD645																																																							
SD646			This register stores the file name of the file register selected by the parameter in ASCII code (with an extension). <table><tr><td></td><td>b15</td><td>to</td><td>b8</td><td>b7</td><td>to</td><td>b0</td></tr><tr><td>SD641</td><td colspan="3">2nd character</td><td colspan="3">1st character</td></tr><tr><td>SD642</td><td colspan="3">4th character</td><td colspan="3">3rd character</td></tr><tr><td>SD643</td><td colspan="3">6th character</td><td colspan="3">5th character</td></tr><tr><td>SD644</td><td colspan="3">8th character</td><td colspan="3">7th character</td></tr><tr><td>SD645</td><td colspan="3">1st character of extension</td><td colspan="3">2EH(.)</td></tr><tr><td>SD646</td><td colspan="3">3rd character of the extension</td><td colspan="3">2nd character of the extension</td></tr></table>		b15	to	b8	b7	to	b0	SD641	2nd character			1st character			SD642	4th character			3rd character			SD643	6th character			5th character			SD644	8th character			7th character			SD645	1st character of extension			2EH(.)			SD646	3rd character of the extension			2nd character of the extension				LCPU	
	b15	to	b8	b7	to	b0																																																	
SD641	2nd character			1st character																																																			
SD642	4th character			3rd character																																																			
SD643	6th character			5th character																																																			
SD644	8th character			7th character																																																			
SD645	1st character of extension			2EH(.)																																																			
SD646	3rd character of the extension			2nd character of the extension																																																			
SD647	File register capacity	File register capacity	This register stores the data size of the selected file register (unit: 1K word).	S (Status change) S (Initial)	New	Qn(H) QnPH QnPRH QnU*3 LCPU																																																	
						Q00J/Q00/ Q01																																																	
SD648	File register block number	File register block number	This register stores the block number of the selected file register.	S (Status change) *4	D9035	Q00J/Q00/ Q01 Qn(H) QnPH QnPRH QnU*3 LCPU																																																	

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU																		
SD650	Comment drive	Comment drive number	This register stores the drive number of the comment selected by the parameter or the QCDSET instruction.	S (Status change)		Qn(H) QnPH QnPRH QnU LCPU																		
SD651	Comment file name	Comment file name	This register stores the file name of the comment selected by the parameter or the QCDSET instruction in ASCII code (with an extension).																					
SD652																								
SD653																								
SD654																								
SD655																								
SD656																								
			<table><tr><td></td><td>b15 to b8</td><td>b7 to b0</td></tr><tr><td>SD651</td><td>2nd character</td><td>1st character</td></tr><tr><td>SD652</td><td>4th character</td><td>3rd character</td></tr><tr><td>SD653</td><td>6th character</td><td>5th character</td></tr><tr><td>SD654</td><td>8th character</td><td>7th character</td></tr><tr><td>SD655</td><td>1st character of the extension</td><td>2EH(.)</td></tr><tr><td>SD656</td><td>3rd character of the extension</td><td>2nd character of the extension</td></tr></table>					b15 to b8	b7 to b0	SD651	2nd character	1st character	SD652	4th character	3rd character	SD653	6th character	5th character	SD654	8th character	7th character	SD655	1st character of the extension	2EH(.)
	b15 to b8	b7 to b0																						
SD651	2nd character	1st character																						
SD652	4th character	3rd character																						
SD653	6th character	5th character																						
SD654	8th character	7th character																						
SD655	1st character of the extension	2EH(.)																						
SD656	3rd character of the extension	2nd character of the extension																						
SD660	Boot operation designation file	Boot designation file drive number	This register stores the number of a drive where the boot designation file (*.QBT) has been stored.	S (Initial)	New	Qn(H) QnPH QnPRH QnU*2 LCPU																		
SD661		File name of boot designation file	This register stores the name of a boot designation file (*.QBT) in ASCII code (with an extension).																					
SD662																								
SD663																								
SD664																								
SD665																								
SD666																								
			<table><tr><td></td><td>b15 to b8</td><td>b7 to b0</td></tr><tr><td>SD661</td><td>2nd character</td><td>1st character</td></tr><tr><td>SD662</td><td>4th character</td><td>3rd character</td></tr><tr><td>SD663</td><td>6th character</td><td>5th character</td></tr><tr><td>SD664</td><td>8th character</td><td>7th character</td></tr><tr><td>SD665</td><td>1st character of the extension</td><td>2EH(.)</td></tr><tr><td>SD666</td><td>3rd character of the extension</td><td>2nd character of the extension</td></tr></table>				b15 to b8	b7 to b0	SD661	2nd character	1st character	SD662	4th character	3rd character	SD663	6th character	5th character	SD664	8th character	7th character	SD665	1st character of the extension	2EH(.)	SD666
	b15 to b8	b7 to b0																						
SD661	2nd character	1st character																						
SD662	4th character	3rd character																						
SD663	6th character	5th character																						
SD664	8th character	7th character																						
SD665	1st character of the extension	2EH(.)																						
SD666	3rd character of the extension	2nd character of the extension																						
SD670	Parameter enable drive information	Parameter enable drive No.	This register stores the number of a drive where valid parameters have been stored.*1 • 0: Drive 0 (program memory) • 1: Drive 1 (SRAM card) • 2: Drive 2 (Flash card/ATA card) • 4: Drive 4 (standard ROM) *1 For the Q00UJCPU, Q00UCPU, and Q01UCPU, only drives 0 and 4 are parameter-valid drives.			QnU																		
			This register stores the number of a drive where valid parameters have been stored. • 0: Drive 0 (program memory) • 2: Drive 2 (SD memory card) • 4: Drive 4 (standard ROM)			LCPU																		

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU																																																	
SD671	Status of latch data backup function	Status display	This register stores the execution status of latch data backup in the following bit pattern. <table><tr><th colspan="2">Status</th><th>Presence/ absence of backup data</th><th>Restore operation at turning power supply ON from OFF</th></tr><tr><td>0</td><td>No backup data</td><td>Absent</td><td>Restoring not executed</td></tr><tr><td>1</td><td>Restore ready completion</td><td rowspan="4">Present</td><td>Restoring executed when turning power supply ON from OFF the following time</td></tr><tr><td>2</td><td>Restore execution completion *1</td><td>Restoring not executed</td></tr><tr><td>3</td><td>Backup execution wait *2</td><td>Restoring not executed</td></tr><tr><td>4</td><td>Restore repeated execution ready completion</td><td>Restoring executed when turning power supply ON from OFF</td></tr></table> *1 Indicates status immediately after restoration. *2 Indicates status after the CPU module is powered off and then on while the CPU module is in the "2: Restore execution completion" status.	Status		Presence/ absence of backup data	Restore operation at turning power supply ON from OFF	0	No backup data	Absent	Restoring not executed	1	Restore ready completion	Present	Restoring executed when turning power supply ON from OFF the following time	2	Restore execution completion *1	Restoring not executed	3	Backup execution wait *2	Restoring not executed	4	Restore repeated execution ready completion	Restoring executed when turning power supply ON from OFF	S (Status change)																														
Status		Presence/ absence of backup data	Restore operation at turning power supply ON from OFF																																																				
0	No backup data	Absent	Restoring not executed																																																				
1	Restore ready completion	Present	Restoring executed when turning power supply ON from OFF the following time																																																				
2	Restore execution completion *1		Restoring not executed																																																				
3	Backup execution wait *2		Restoring not executed																																																				
4	Restore repeated execution ready completion		Restoring executed when turning power supply ON from OFF																																																				
SD672	Backup information	Backup time (Year and month)	This register stores the year (last two digits) and the month when data were backed up in 2-digit BCD. <table><tr><td>b15</td><td>to</td><td>b12</td><td>b11</td><td>to</td><td>b8</td><td>b7</td><td>to</td><td>b4</td><td>b3</td><td>to</td><td>b0</td><td>Example:</td></tr><tr><td>□</td><td></td><td>□</td><td>□</td><td></td><td>□</td><td>□</td><td></td><td>□</td><td>□</td><td></td><td>□</td><td>July, 1993</td></tr><tr><td colspan="6">Year</td><td colspan="6">Month</td><td>9307_H</td></tr></table>	b15	to	b12	b11	to	b8	b7	to	b4	b3	to	b0	Example:	□		□	□		□	□		□	□		□	July, 1993	Year						Month						9307 _H	S (At write)	New	QnU LCPU										
b15		to	b12	b11	to	b8	b7	to	b4	b3	to	b0	Example:																																										
□			□	□		□	□		□	□		□	July, 1993																																										
Year						Month						9307 _H																																											
SD673	Backup time (Day and hour)	This register stores the day and the hour when data were backed up in 2-digit BCD. <table><tr><td>b15</td><td>to</td><td>b12</td><td>b11</td><td>to</td><td>b8</td><td>b7</td><td>to</td><td>b4</td><td>b3</td><td>to</td><td>b0</td><td>Example:</td></tr><tr><td>□</td><td></td><td>□</td><td>□</td><td></td><td>□</td><td>□</td><td></td><td>□</td><td>□</td><td></td><td>□</td><td>31st, 10 a.m.</td></tr><tr><td colspan="6">Day</td><td colspan="6">Hour</td><td>3110_H</td></tr></table>	b15	to	b12	b11	to	b8	b7	to	b4	b3	to	b0	Example:	□		□	□		□	□		□	□		□	31st, 10 a.m.	Day						Hour						3110 _H														
b15	to	b12	b11	to	b8	b7	to	b4	b3	to	b0	Example:																																											
□		□	□		□	□		□	□		□	31st, 10 a.m.																																											
Day						Hour						3110 _H																																											
SD674	Backup time (Minute and second)	This register stores the minute and the second when data were backed up in 2-digit BCD. <table><tr><td>b15</td><td>to</td><td>b12</td><td>b11</td><td>to</td><td>b8</td><td>b7</td><td>to</td><td>b4</td><td>b3</td><td>to</td><td>b0</td><td>Example:</td></tr><tr><td>□</td><td></td><td>□</td><td>□</td><td></td><td>□</td><td>□</td><td></td><td>□</td><td>□</td><td></td><td>□</td><td>35 min., 48 sec.</td></tr><tr><td colspan="6">Minute</td><td colspan="6">Second</td><td>3548_H</td></tr></table>	b15	to	b12	b11	to	b8	b7	to	b4	b3	to	b0	Example:	□		□	□		□	□		□	□		□	35 min., 48 sec.	Minute						Second						3548 _H														
b15	to	b12	b11	to	b8	b7	to	b4	b3	to	b0	Example:																																											
□		□	□		□	□		□	□		□	35 min., 48 sec.																																											
Minute						Second						3548 _H																																											
SD675	Backup time (Year and day of week)	This register stores the year (first two digits) and the day of the week when data were backed up in BCD. <table><tr><td>b15</td><td>to</td><td>b12</td><td>b11</td><td>to</td><td>b8</td><td>b7</td><td>to</td><td>b4</td><td>b3</td><td>to</td><td>b0</td><td>Example:</td></tr><tr><td>□</td><td></td><td>□</td><td>□</td><td></td><td>□</td><td>□</td><td></td><td>□</td><td>□</td><td></td><td>□</td><td>1993, Friday</td></tr><tr><td colspan="6">Higher digits of year (0 to 99)</td><td colspan="6">Day of the week</td><td>1905_H</td></tr></table> <table><tr><td>0</td><td>Sunday</td></tr><tr><td>1</td><td>Monday</td></tr><tr><td>2</td><td>Tuesday</td></tr><tr><td>3</td><td>Wednesday</td></tr><tr><td>4</td><td>Thursday</td></tr><tr><td>5</td><td>Friday</td></tr><tr><td>6</td><td>Saturday</td></tr></table>	b15	to	b12	b11	to	b8	b7	to	b4	b3	to	b0	Example:	□		□	□		□	□		□	□		□	1993, Friday	Higher digits of year (0 to 99)						Day of the week						1905 _H	0	Sunday	1	Monday	2	Tuesday	3	Wednesday	4	Thursday	5	Friday	6	Saturday
b15	to	b12	b11	to	b8	b7	to	b4	b3	to	b0	Example:																																											
□		□	□		□	□		□	□		□	1993, Friday																																											
Higher digits of year (0 to 99)						Day of the week						1905 _H																																											
0	Sunday																																																						
1	Monday																																																						
2	Tuesday																																																						
3	Wednesday																																																						
4	Thursday																																																						
5	Friday																																																						
6	Saturday																																																						

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU												
SD676	Backup data restration information	Restore time (Year and month)	This register stores the year (last two digits) and the month when data were restored in 2-digit BCD. b15 to b12 b11 to b8 b7 to b4 b3 to b0 Example:  July, 1993 9307_H	S (Initial)	New	QnU LCPU												
SD677		Restore time (Day and time)	This register stores the day and the hour when data were restored in 2-digit BCD. b15 to b12 b11 to b8 b7 to b4 b3 to b0 Example:  31st, 10 a.m. 3110_H															
SD678		Restore time (Minute and second)	This register stores the minute and the second when data were restored in 2-digit BCD. b15 to b12 b11 to b8 b7 to b4 b3 to b0 Example:  35 min., 48 sec. 3548_H															
SD679		Restore time (Year and day of week)	This register stores the year (first two digits) and the day of the week when data were restored in BCD. b15 to b12 b11 to b8 b7 to b4 b3 to b0 Example:  1993, Friday 1905_H Higher digits of year (0 to 99) <table><tr><th colspan="2">Day of the week</th></tr><tr><td>0</td><td>Sunday</td></tr><tr><td>1</td><td>Monday</td></tr><tr><td>2</td><td>Tuesday</td></tr><tr><td>3</td><td>Wednesday</td></tr><tr><td>4</td><td>Thursday</td></tr><tr><td>5</td><td>Friday</td></tr><tr><td>6</td><td>Saturday</td></tr></table>				Day of the week		0	Sunday	1	Monday	2	Tuesday	3	Wednesday	4	Thursday
Day of the week																		
0	Sunday																	
1	Monday																	
2	Tuesday																	
3	Wednesday																	
4	Thursday																	
5	Friday																	
6	Saturday																	

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU
SD681	Program memory write (transfer) status	Write (transfer) status display (percentage)	This register stores the progress of writing (transfer) to the program memory (flash ROM) in percentage (0 to 100%). (When a write (transfer) command is given, "0" is stored in this register.)	S (At write)	New	QnU LCPU
SD682	Program memory write count index	Write count index up to present	This register stores the index value of write count of the program memory (flash ROM)*1 up to the present in 32-bit binary. When the index value exceeds 100 thousand times, "FLASH ROM ERROR" (error code: 1610) occurs. (The index value will be counted even after it exceeds 100 thousand.)			
SD683			*1 The write count does not equal to the index value. (Since the maximum write count of the flash ROM has been increased by the system, 1 is added about every two writing operations.)			
SD686	Standard ROM write (transfer) status	Write (transfer) status display (percentage)	This register stores the progress of writing (transfer) to the standard ROM (flash ROM) in percentage (0 to 100%). When a write (transfer) command is given, "0" is stored in this register.			
SD687	Standard ROM write count index	Write count index up to present	This register stores the index value of write count of the standard ROM (flash ROM)*1 up to the present in 32-bit binary. When the index value exceeds 100 thousand times, "FLASH ROM ERROR" (error code: 1610) occurs. (The index value will be counted even after it exceeds 100 thousand.)			
SD688			*1 The write count does not equal to the index value. (Since the maximum write count of the flash ROM has been increased by the system, 1 is added to the index value when the total write data size after the previous count-up reaches about 1M byte.)			

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU
SD689	Backup error factor	Backup error factor	This register stores the cause of an error that occurred during backup. • 0_H: No error • 100_H: Memory card not inserted • 200_H: Backup data size exceeded • 300_H: Memory card write-protect setting • 400_H: Memory card write error • 500_H: Backup data read error (program memory) • 503_H: Backup data read error (standard RAM) • 504_H: Backup data read error (standard ROM) • 510_H: Backup data read error (system data) • 600_H: Backup preparation was performed while latch data was being backed up to the standard ROM. • 601_H: Backup preparation was performed during online change. • 602_H: Backup preparation was performed while a FTP client connected to the CPU module in FTP connection is present.	S (Error)	New	QnU*1 LCPU
SD690	Backup status	Backup status	Stores the current backup status. • 0: Before backup • 1: Being prepared • 2: Ready • 3: Being executed • 4: Completed • FF: Backup error	S (Status change)		
SD691	Backup execution status	Backup execution status display (percentage)	• This register stores the progress of backup to the memory card in percentage (0 to 100%). • "0" is stored at the start of backup.			
SD692	Restoration error factor	Factor of error occurred in restoration	Stores the cause of an error that occurred in restoration. • 800_H: The CPU module model name does not match. • 801_H: The backup data file does not match or the reading of backup data from the memory card is not completed. • 810_H: Writing backup data to the restoration drive is not completed.	S (Error)		

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU
SD693	Restoration status	Current restoration status	Stores the current restoration status. • 0: Before restoration • 1: Being executed • 2: Completed • FF: Restoration error (In automatic restoration, "0: Before restoration" is stored at the completion of restoration.)	S (Status change)	New	QnU* ¹ LCPU
SD694	Restoration execution status	Restoration execution status display (Percentage)	• This register stores the progress of restoration to the CPU module in percentage (0 to 100%). • "0" is stored at the start of restoration. In automatic restoration, "0: Before restoration" is stored at the completion of restoration.			
SD695	Specification of writing to standard ROM instruction count	Specification of writing to standard ROM instruction count	• This register stores the maximum number of executions of the writing to standard ROM instruction (SP.DEVST) per day. • When the number of executions of the writing to standard ROM instruction exceeds the number of times set by SD695, "OPERATION ERROR" (error code: 4113) occurs. • The setting range of this register is 1 to 32767. If "0" or a value outside the range has been set, "OPERATION ERROR" (error code: 4113) occurs at execution of the writing to standard ROM instruction.	U		QnU LCPU
SD696	Available memory in memory card	Available memory in memory card	This register stores a free space value in a memory card in 32-bit binary.	S (Backup in operation)		QnU* ¹
SD697						
SD696	Free memory card space at backup	Free memory card space at backup (lower bits)	This register stores a free space value in a SD memory card if the free space is insufficient for storing the backup data and resulting in a backup error. (unit: byte) This register is cleared to "0" when backup is completed.		LCPU	
SD697		Free memory card space at backup (upper bits)				
SD698	Backup data capacity	Backup data size (lower bits)	This register stores backup data size in 32-bit binary.		QnU* ¹ LCPU	
SD699		Backup data size (upper bits)				

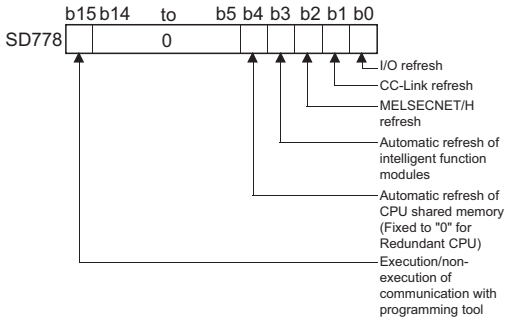
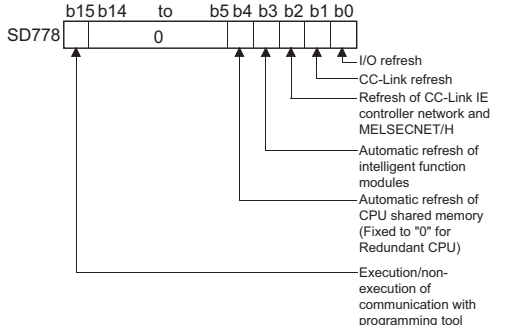
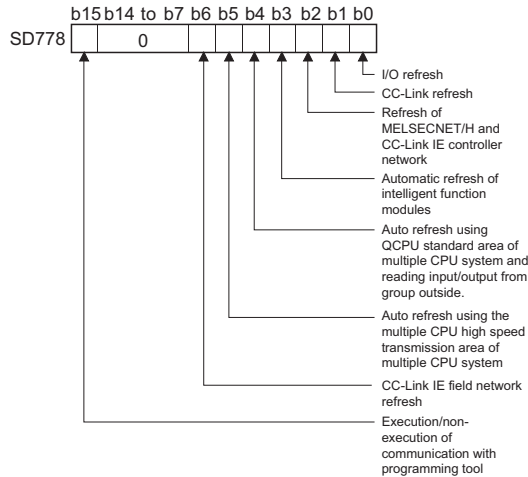
- *1 The module whose first 5 digits of serial No. is "10102" or later. (Except the Q00UJCPU, Q00UCPU, and Q01UCPU)
 *2 The Universal model QCPU except the Q00UJCPU, Q00UCPU, and Q01UCPU.
 *3 The Universal model QCPU except the Q00UJCPU.
 *4 On the Basic model QCPU, data is set at STOP to RUN or RSET instruction execution after parameter execution.

(6) Instruction-related register

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU																				
SD705	Mask pattern	Mask pattern	Turning SM705 during a block operation enables all data in the block to be processed to masked values according to the mask patterns stored in SD705 (in SD705 and SD706 for double word data).	U	New	Q00J/Q00/ Q01 Qn(H) QnPH QnPRH																				
SD706																										
SD715	IMASK instruction mask pattern	Mask pattern	The mask patterns masked by the IMASK instruction are stored as follows. <table><tr><td></td><td>b15</td><td></td><td>b1</td><td>b0</td></tr><tr><td>SD715</td><td>I15</td><td>to</td><td>I1</td><td>I0</td></tr><tr><td>SD716</td><td>I31</td><td>to</td><td>I17</td><td>I16</td></tr><tr><td>SD717</td><td>I47</td><td>to</td><td>I33</td><td>I32</td></tr></table>			b15		b1	b0	SD715	I15	to	I1	I0	SD716	I31	to	I17	I16	SD717	I47	to	I33	I32	S (During execution)	QCPU LCPU
				b15			b1	b0																		
SD715				I15		to	I1	I0																		
SD716	I31	to	I17	I16																						
SD717	I47	to	I33	I32																						
SD716																										
SD717																										
SD718	Accumulator	Accumulator	For use as replacement for accumulators used in A series programs.	S/U																						
SD719																										
SD720	Program No. designation for PLOADP instruction	Program No. designation for PLOADP instruction	This register stores the program number of the program to read it with the PLOADP instruction.(Specified range: 1 to 124)	U		Qn(H) QnPH																				

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU																																																																																																																																																																																																																																							
SD738	Message storage	Message storage	This register stores the message specified by the MSG instruction. <table><tr><td></td><td>b15</td><td>to</td><td>b8</td><td>b7</td><td>to</td><td>b0</td></tr><tr><td>SD738</td><td>2nd character</td><td></td><td></td><td>1st character</td><td></td><td></td></tr><tr><td>SD739</td><td>4th character</td><td></td><td></td><td>3rd character</td><td></td><td></td></tr><tr><td>SD740</td><td>6th character</td><td></td><td></td><td>5th character</td><td></td><td></td></tr><tr><td>SD741</td><td>8th character</td><td></td><td></td><td>7th character</td><td></td><td></td></tr><tr><td>SD742</td><td>10th character</td><td></td><td></td><td>9th character</td><td></td><td></td></tr><tr><td>SD743</td><td>12th character</td><td></td><td></td><td>11th character</td><td></td><td></td></tr><tr><td>SD744</td><td>14th character</td><td></td><td></td><td>13th character</td><td></td><td></td></tr><tr><td>SD745</td><td>16th character</td><td></td><td></td><td>15th character</td><td></td><td></td></tr><tr><td>SD746</td><td>18th character</td><td></td><td></td><td>17th character</td><td></td><td></td></tr><tr><td>SD747</td><td>20th character</td><td></td><td></td><td>19th character</td><td></td><td></td></tr><tr><td>SD748</td><td>22nd character</td><td></td><td></td><td>21st character</td><td></td><td></td></tr><tr><td>SD749</td><td>24th character</td><td></td><td></td><td>23rd character</td><td></td><td></td></tr><tr><td>SD750</td><td>26th character</td><td></td><td></td><td>25th character</td><td></td><td></td></tr><tr><td>SD751</td><td>28th character</td><td></td><td></td><td>27th character</td><td></td><td></td></tr><tr><td>SD752</td><td>30th character</td><td></td><td></td><td>29th character</td><td></td><td></td></tr><tr><td>SD753</td><td>32nd character</td><td></td><td></td><td>31st character</td><td></td><td></td></tr><tr><td>SD754</td><td>34th character</td><td></td><td></td><td>33rd character</td><td></td><td></td></tr><tr><td>SD755</td><td>36th character</td><td></td><td></td><td>35th character</td><td></td><td></td></tr><tr><td>SD756</td><td>38th character</td><td></td><td></td><td>37th character</td><td></td><td></td></tr><tr><td>SD757</td><td>40th character</td><td></td><td></td><td>39th character</td><td></td><td></td></tr><tr><td>SD758</td><td>42nd character</td><td></td><td></td><td>41st character</td><td></td><td></td></tr><tr><td>SD759</td><td>44th character</td><td></td><td></td><td>43rd character</td><td></td><td></td></tr><tr><td>SD760</td><td>46th character</td><td></td><td></td><td>45th character</td><td></td><td></td></tr><tr><td>SD761</td><td>48th character</td><td></td><td></td><td>47th character</td><td></td><td></td></tr><tr><td>SD762</td><td>50th character</td><td></td><td></td><td>49th character</td><td></td><td></td></tr><tr><td>SD763</td><td>52nd character</td><td></td><td></td><td>51st character</td><td></td><td></td></tr><tr><td>SD764</td><td>54th character</td><td></td><td></td><td>53rd character</td><td></td><td></td></tr><tr><td>SD765</td><td>56th character</td><td></td><td></td><td>55th character</td><td></td><td></td></tr><tr><td>SD766</td><td>58th character</td><td></td><td></td><td>57th character</td><td></td><td></td></tr><tr><td>SD767</td><td>60th character</td><td></td><td></td><td>59th character</td><td></td><td></td></tr><tr><td>SD768</td><td>62nd character</td><td></td><td></td><td>61st character</td><td></td><td></td></tr><tr><td>SD769</td><td>64th character</td><td></td><td></td><td>63rd character</td><td></td><td></td></tr></table>		b15	to	b8	b7	to	b0	SD738	2nd character			1st character			SD739	4th character			3rd character			SD740	6th character			5th character			SD741	8th character			7th character			SD742	10th character			9th character			SD743	12th character			11th character			SD744	14th character			13th character			SD745	16th character			15th character			SD746	18th character			17th character			SD747	20th character			19th character			SD748	22nd character			21st character			SD749	24th character			23rd character			SD750	26th character			25th character			SD751	28th character			27th character			SD752	30th character			29th character			SD753	32nd character			31st character			SD754	34th character			33rd character			SD755	36th character			35th character			SD756	38th character			37th character			SD757	40th character			39th character			SD758	42nd character			41st character			SD759	44th character			43rd character			SD760	46th character			45th character			SD761	48th character			47th character			SD762	50th character			49th character			SD763	52nd character			51st character			SD764	54th character			53rd character			SD765	56th character			55th character			SD766	58th character			57th character			SD767	60th character			59th character			SD768	62nd character			61st character			SD769	64th character			63rd character			S (During execution)	New	Qn(H)
				b15	to	b8	b7	to	b0																																																																																																																																																																																																																																				
SD738				2nd character			1st character																																																																																																																																																																																																																																						
SD739				4th character			3rd character																																																																																																																																																																																																																																						
SD740				6th character			5th character																																																																																																																																																																																																																																						
SD741				8th character			7th character																																																																																																																																																																																																																																						
SD742				10th character			9th character																																																																																																																																																																																																																																						
SD743				12th character			11th character																																																																																																																																																																																																																																						
SD744				14th character			13th character																																																																																																																																																																																																																																						
SD745				16th character			15th character																																																																																																																																																																																																																																						
SD746				18th character			17th character																																																																																																																																																																																																																																						
SD747				20th character			19th character																																																																																																																																																																																																																																						
SD748				22nd character			21st character																																																																																																																																																																																																																																						
SD749				24th character			23rd character																																																																																																																																																																																																																																						
SD750				26th character			25th character																																																																																																																																																																																																																																						
SD751				28th character			27th character																																																																																																																																																																																																																																						
SD752				30th character			29th character																																																																																																																																																																																																																																						
SD753				32nd character			31st character																																																																																																																																																																																																																																						
SD754				34th character			33rd character																																																																																																																																																																																																																																						
SD755				36th character			35th character																																																																																																																																																																																																																																						
SD756				38th character			37th character																																																																																																																																																																																																																																						
SD757				40th character			39th character																																																																																																																																																																																																																																						
SD758				42nd character			41st character																																																																																																																																																																																																																																						
SD759				44th character			43rd character																																																																																																																																																																																																																																						
SD760				46th character			45th character																																																																																																																																																																																																																																						
SD761				48th character			47th character																																																																																																																																																																																																																																						
SD762				50th character			49th character																																																																																																																																																																																																																																						
SD763				52nd character			51st character																																																																																																																																																																																																																																						
SD764				54th character			53rd character																																																																																																																																																																																																																																						
SD765				56th character			55th character																																																																																																																																																																																																																																						
SD766	58th character			57th character																																																																																																																																																																																																																																									
SD767	60th character			59th character																																																																																																																																																																																																																																									
SD768	62nd character			61st character																																																																																																																																																																																																																																									
SD769	64th character			63rd character																																																																																																																																																																																																																																									
SD739																																																																																																																																																																																																																																													
SD740																																																																																																																																																																																																																																													
SD741																																																																																																																																																																																																																																													
SD742																																																																																																																																																																																																																																													
SD743																																																																																																																																																																																																																																													
SD744																																																																																																																																																																																																																																													
SD745																																																																																																																																																																																																																																													
SD746																																																																																																																																																																																																																																													
SD747																																																																																																																																																																																																																																													
SD748																																																																																																																																																																																																																																													
SD749																																																																																																																																																																																																																																													
SD750																																																																																																																																																																																																																																													
SD751																																																																																																																																																																																																																																													
SD752																																																																																																																																																																																																																																													
SD753																																																																																																																																																																																																																																													
SD754																																																																																																																																																																																																																																													
SD755																																																																																																																																																																																																																																													
SD756																																																																																																																																																																																																																																													
SD757																																																																																																																																																																																																																																													
SD758																																																																																																																																																																																																																																													
SD759																																																																																																																																																																																																																																													
SD760																																																																																																																																																																																																																																													
SD761																																																																																																																																																																																																																																													
SD762																																																																																																																																																																																																																																													
SD763																																																																																																																																																																																																																																													
SD764																																																																																																																																																																																																																																													
SD765																																																																																																																																																																																																																																													
SD766																																																																																																																																																																																																																																													
SD767																																																																																																																																																																																																																																													
SD768																																																																																																																																																																																																																																													
SD769																																																																																																																																																																																																																																													

SD774	PID limit setting (for complete derivative)	0: With limit 1: Without limit	This register stores the limit of each PID loop as shown below. <table><tr><td></td><td>b15</td><td>to</td><td>b8</td><td>b7</td><td>to</td><td>b1</td><td>b0</td></tr><tr><td>SD774</td><td></td><td></td><td></td><td>Loop8</td><td>to</td><td>Loop2</td><td>Loop1</td></tr></table>		b15	to	b8	b7	to	b1	b0	SD774				Loop8	to	Loop2	Loop1	U		Q00J/Q00 /Q01*1	
			b15	to	b8	b7	to	b1	b0														
SD774						Loop8	to	Loop2	Loop1														
SD774 and SD775	This register stores the limit of each PID loop as shown below. <table><tr><td></td><td>b15</td><td></td><td></td><td>b1</td><td></td><td>b0</td></tr><tr><td>SD774</td><td>Loop16</td><td>to</td><td></td><td>Loop2</td><td></td><td>Loop1</td></tr><tr><td>SD775</td><td>Loop32</td><td>to</td><td></td><td>Loop18</td><td></td><td>Loop17</td></tr></table>		b15			b1		b0	SD774	Loop16	to		Loop2		Loop1	SD775	Loop32	to		Loop18		Loop17	Qn(H) QnPRH QnU LCPU
	b15			b1		b0																	
SD774	Loop16	to		Loop2		Loop1																	
SD775	Loop32	to		Loop18		Loop17																	

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU
SD778	Refresh processing selection when the COM/CCOM instruction is executed	b0 to b14: (Default: 0) 0: Do not refresh 1: Refresh b15 bit 0: Communication with CPU module is executed 1: Communication with CPU module is nonexecuted	- Selects whether or not the data is refreshed when the COM instruction is executed. - Designation of SD778 is made valid when SM775 turns ON.  - Refresh between multiple CPUs by the COM instruction is performed under the following conditions. Data reception from another CPU: When b4 of SD778 is "1" Data transmission from host CPU: When b15 of SD778 is "0"	U	New	Q00J/Q00 /Q01* ¹ Qn(H)* ²
			- Selects whether or not the data is refreshed when the COM instruction is executed. - Designation of SD778 is made valid when SM775 turns ON.  - Refresh between multiple CPUs by the COM instruction is performed under the following conditions. Data reception from another CPU: When b4 of SD778 is "1" Data transmission from host CPU: When b15 of SD778 is "0" - When b2 of SD778 is 1, both the CC-Link IE controller network and MELSECNET/H perform a refresh. Therefore, when refresh point is large, processing time for the COM instruction is extended.			Qn(H)* ⁴ QnPH* ³ QnPRH
			- Selects whether or not the data is refreshed when the COM, CCOM instruction is executed. - Designation of SD778 is made valid when SM775 turns ON. 			QnU

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU
SD778	Refresh processing selection when the COM/CCOM instruction is executed	b0 , b1, b3, b14 (Default: 0) 0: Do not refresh 1: Refresh b15 bit 0: Communication with peripheral device is executed 1: Communication with peripheral device is nonexecuted	• Selects whether or not the data is refreshed when the COM, CCOM instruction is executed. • Designation of SD778 is made valid when SM775 turns ON. SD778 b15b14to b4b3b2b1b0 0 I/O refresh CC-Link refresh Fixed to 0 Automatic refresh of intelligent function modules Fixed to 0 Communication with display unit Execution/non-execution of communication with programming tool	U		LCPU
SD781 to SD785	Mask pattern of IMASK instruction	Mask pattern	The mask patterns masked by the IMASK instruction are stored as follows. SD781 b15 I63 to b1 I49 b0 I48 SD782 I79 to I65 I64 to SD785 I127 to I113 I112	S (During execution)	New	Q00J/Q00/Q01
SD781 to SD793			The mask patterns masked by the IMASK instruction are stored as follows. *1 SD781 b15 I63 to b1 I49 b0 I48 SD782 I79 to I65 I64 to SD793 I255 to I241 I240 *1: The Q00UJCPU, Q00UCPU, and Q01UCPU cannot use SD786 to SD793.			Qn(H) QnPH QnPRH QnU LCPU
SD794	PID limit setting (for incomplete derivative)	0: With limit 1: Without limit	This register stores the limit of each PID loop as shown below. SD794 b15to b8 b7 b1 b0 Loop8 to Loop2 Loop1	U		Q00J/Q00/Q01*1
SD794 to SD795			This register stores the limit of each PID loop as shown below. SD794 b15 Loop16 to b1 Loop2 b0 Loop1 SD795 Loop32 to Loop18 Loop17		Qn(H)*4 QnPRH QnU LCPU	

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU
SD796	Maximum number of blocks used for the multiple CPU high-speed transmission dedicated instruction setting (for CPU No.1)	Maximum number of blocks range for dedicated instructions Range: 1 to 7 (Default: 2 Or when setting other than 1 to 7, the register operates as 7). ^{*6}	Specifies the maximum number of blocks used for the multiple CPU high-speed transmission dedicated instruction (target CPU=CPU No.1). When the multiple CPU high-speed transmission dedicated instruction is executed to the CPU No.1, and the number of empty blocks of the dedicated instruction transmission area is less than the setting value of this register, SM796 is turned ON, which is used as the interlock signal for consecutive execution of the multiple CPU high-speed transmission dedicated instruction.	U (At 1 scan after RUN)	New	QnU ^{*5}
SD797	Maximum number of blocks used for the multiple CPU high-speed transmission dedicated instruction setting (for CPU No.2)		Specifies the maximum number of blocks used for the multiple CPU high-speed transmission dedicated instruction (target CPU=CPU No.2). When the multiple CPU high-speed transmission dedicated instruction is executed to the CPU No.2, and the number of empty blocks of the dedicated instruction transmission area is less than the setting value of this register, SM797 is turned ON, which is used as the interlock signal for consecutive execution of the multiple CPU high-speed transmission dedicated instruction.			
SD798	Maximum number of blocks used for the multiple CPU high-speed transmission dedicated instruction setting (for CPU No.3)		Specifies the maximum number of blocks used for the multiple CPU high-speed transmission dedicated instruction (target CPU=CPU No.3). When the multiple CPU high-speed transmission dedicated instruction is executed to the CPU No.3, and the number of empty blocks of the dedicated instruction transmission area is less than the setting value of this register, SM798 is turned ON, which is used as the interlock signal for consecutive execution of the multiple CPU high-speed transmission dedicated instruction.			
SD799	Maximum number of blocks used for the multiple CPU high-speed transmission dedicated instruction setting (for CPU No.4)		Specifies the maximum number of blocks used for the multiple CPU high-speed transmission dedicated instruction (target CPU=CPU No.4). When the multiple CPU high-speed transmission dedicated instruction is executed to the CPU No.4, and the number of empty blocks of the dedicated instruction transmission area is less than the setting value of this register, SM799 is turned ON, which is used as the interlock signal for consecutive execution of the multiple CPU high-speed transmission dedicated instruction.			

*1 Function version is B or later.

*2 The module whose first 5 digits of serial No. is "04012" or later.

*3 The module whose first 5 digits of serial No. is "07032" or later.

*4 The module whose first 5 digits of serial No. is "09012" or later.

*5 The Universal model QCPU except the Q00UJCPU, Q00UCPU, Q01UCPU, and Q02UCPU.

*6 The range is from 1 to 9 for the Q03UDCPU, Q04UDCPU, and Q06UDHCP whose first 5 digits of serial number is "10012" or earlier.

(Default: 2 Or when setting other than 1 to 9, the register operates as 9).

12.2 Special Register List

Number	Name	Name	Explanation	Set by (When Set)	Corresponding ACPU D9□□□	Corresponding CPU
SD952	History of memory copy from control system to standby system	Latest status of memory copy from control system to standby system	This register stores a value indicating the completion status of the latest memory data copy from the control system to the standby system. 1) The value same as the SD1596 value is stored at completion or abend of the memory data copy from the control system to the standby system. 2) Since data have been backed up in case of power failure, this register holds the value indicating the latest memory data copy status from the control system to the standby system. 3) This register is cleared to 0 by latch clear.	S (Status change)	New	QnPRH

*1: This applies to the Built-in Ethernet port QCPU.

(9) Remote password count

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU
SD979	Direct MELSOFT connection	Count of unlock processing failures	This register stores the number of mismatched password entries. Range: 0 to 0FFFE _H (0FFFF _H when the range is exceeded)	S (Status change)	New	QnU*1 LCPU
SD980 to SD995	Connection 1 to 16					
SD997	MELSOFT connection using UDP port					
SD998	MELSOFT connection using TCP port					
SD999	FTP communication port					

*1: This applies to the Built-in Ethernet port QCPU.

(10) Conversion from A series to Q or L series

The special register (D9000 to D9255) for ACPUs corresponds to the special register (SD1000 to SD1255) for QCPUs or LCPUs after the A to Q/L conversion. (Note that the Basic model QCPU and Redundant CPU do not support the A to Q/L conversion.)

All data in this area of the special register are set by system (cannot be set by user using a program). To set data by user, correct the program using the special register for QCPU or LCPU. The special register (D9200 to D9255), however, includes the areas that can be set by user. For those areas, data can be set by user in the converted special register (SD1200 to SD1255) as well. For details on the special register for ACPUs, refer to the following.

 User's manual for the CPU module used

 Type MELSECNET, MELSECNET/B Data Link System Reference Manual

Point

To use the converted special register in the High Performance model QCPU, Process CPU, Universal model QCPU, or LCPU, check "Use special relay/special register from SM/SD1000" under "A-PLC Compatibility Setting".

Project window ⇨ [Parameter] ⇨ [PLC Parameter] ⇨ [PLC System]

Note that the processing time will increase when the converted special register is used.

[How to read the Special Register for Modification column]

- If the special register number for QCPU or LCPU is provided, correct the program using it.
- ☐ means that the converted special register can be used.
- ☒ means that the special register cannot be used in QCPU or LCPU.

ACPU Special Register	Special Register after Conversion	Special Register for Modification	Name	Meaning	Details	Corres- ponding CPU																																								
D9000	SD1000	-	Fuse blown	Number of module with blown fuse	- If a module with blown fuse is detected, the lowest first I/O number of the module is stored in hexadecimal. (Example: If a fuse blown is occurred in the output module with output number Y50 to Y6F, "50" is stored in hexadecimal.) To monitor the number by a programming tool, monitor in hexadecimal.(This register is cleared when contents in SD1100 to SD1107 are all reset to "0".) - Output modules on remote I/O stations are also checked for blown fuse.	Qn(H) QnPH QnU*1																																								
D9001	SD1001	-	Fuse blown	Number of module with blown fuse	- If any fuse is blown, this register stores a number corresponding to each setting switch number or to slot number of the base unit. <table><tr><th colspan="2">AJ02 I/O module</th><th colspan="2">Extension base unit</th></tr><tr><th>Setting switch</th><th>Stored data</th><th>Base unit slot No.</th><th>Stored data</th></tr><tr><td>0</td><td>0</td><td>0</td><td>4</td></tr><tr><td>1</td><td>1</td><td>1</td><td>5</td></tr><tr><td>2</td><td>2</td><td>2</td><td>6</td></tr><tr><td>3</td><td>3</td><td>3</td><td>7</td></tr><tr><td>4</td><td>4</td><td></td><td></td></tr><tr><td>5</td><td>5</td><td></td><td></td></tr><tr><td>6</td><td>6</td><td></td><td></td></tr><tr><td>7</td><td>7</td><td></td><td></td></tr></table> - For the remote I/O station, the value of (module I/O No./10_H) + 1 is stored.	AJ02 I/O module		Extension base unit		Setting switch	Stored data	Base unit slot No.	Stored data	0	0	0	4	1	1	1	5	2	2	2	6	3	3	3	7	4	4			5	5			6	6			7	7			Qn(H) QnPH
AJ02 I/O module		Extension base unit																																												
Setting switch	Stored data	Base unit slot No.	Stored data																																											
0	0	0	4																																											
1	1	1	5																																											
2	2	2	6																																											
3	3	3	7																																											
4	4																																													
5	5																																													
6	6																																													
7	7																																													
D9002	SD1002	-	I/O module verify error	I/O module verify error module number	- If the status of the I/O modules changes from that obtained at power-on, the lowest first I/O number of the module is stored in hexadecimal.(Example: If a module verification error is occurred on the output module with output numbers Y50 to Y6F, "50" is stored in hexadecimal.) To monitor the number by a programming tool, monitor in hexadecimal.(This register is cleared when contents in SD1116 to SD1123 are all reset to "0".) - I/O module verification is conducted on I/O modules on remote I/O stations.	Qn(H) QnPH QnU*1 LCPU																																								
D9005	SD1005	-	AC DOWN counter	Number of times for AC DOWN	- A value stored in this register is incremented by one whenever the input voltage falls to or below 85% (AC power) or 65% (DC power) of the rating during operation of the CPU module. - The counter starts the routine: counts up from 0 to 32767, then counts down to -32768 and then again counts up to 0.	Qn(H) QnPH QnU*1 LCPU																																								
D9008	SD1008	SD0	Self-diagnostic error	Self-diagnostic error number	This register stores the error code of an error detected by self-diagnostics.	Qn(H) QnPH QnU*1 LCPU																																								
D9009	SD1009	SD62	Annunciator detection	F number at which external failure has occurred	- When any of F0 to F2047 (default device setting) is turned on by the OUT F or SET F instruction, the F number that has been detected earliest among the F numbers that have turned on is stored in BIN code. - SD1009 can be cleared by RST F or LEDR instruction. If another F number has been detected, the clearing of SD1009 causes the next number to be stored in SD1009.																																									

ACPU Special Register	Special Register after Conversion	Special Register for Modification	Name	Meaning	Details	Corresponding CPU
D9010	SD1010	×	Error step	Step number at which operation error has occurred.	If an operation error occurred during execution of an application instruction, the number of the step having the error is stored. The contents of SD1010 are updated upon every operation error.	Qn(H) QnPH
D9011	SD1011	×		Step number at which operation error has occurred.	If an operation error occurred during execution of an application instruction, the number of the step having the error is stored. Because the step number is stored in SD1011 when SM1011 turns from off to on, the data in SD1011 are not updated unless SM1011 is cleared by a user program	
D9014	SD1014	×	I/O control mode	I/O control mode number	The I/O control mode that has been set is returned in any of the following numbers. • 0: Both input and output in direct mode • 1: Input in refresh mode, output in direct mode • 3: Both input and output in refresh mode	
D9015	SD1015	SD203	Operating status of CPU	Operating status of CPU	Operation status of a CPU module is stored as shown below. Remote RUN/STOP by computer 0 RUN 1 STOP 2 PAUSE*1 CPU module key switch 0 RUN 1 STOP 2 PAUSE*1 3 STEP RUN (Remains the same in remote RUN/STOP mode.) Status in program 0 Except below 1 STOP Instruction execution Remote RUN/STOP by parameter setting 0 RUN 1 STOP 2 PAUSE*1 *1 For the High Performance model QCPU and Process CPU, if the CPU module is running and SM1040 is off, the CPU module remains in the RUN status even though it is set to the PAUSE status.	Qn(H) QnPH QnU*1 LCPU

ACPU Special Register	Special Register after Conversion	Special Register for Modification	Name	Meaning	Details	Corresponding CPU
D9016	SD1016	×	Program number	0: Main program (ROM) 1: Main program (RAM) 2: Subprogram 1 (RAM) 3: Subprogram 2 (RAM) 4: Subprogram 3 (RAM) 5: Subprogram 1 (ROM) 6: Subprogram 2 (ROM) 7: Subprogram 3 (ROM) 8: Main program (E ² PROM) 9: Subprogram 1 (E ² PROM) A: Subprogram 2 (E ² PROM) B: Subprogram 3 (E ² PROM)	This register stores any of the values from 0 to B, indicating which program is currently running.	Qn(H) QnPH
D9017	SD1017	SD524	Scan time	Minimum scan time (10 ms units)	If a scan time value is smaller than the value in SD1017, the SD1017 value is updated in the END processing. Therefore the minimum value of scan time is stored in SD1017.	Qn(H) QnPH QnU ^{*1} LCPU
D9018	SD1018	SD520		Scan time (10 ms units)	This register stores a scan time in every END processing.	
D9019	SD1019	SD526		Maximum scan time (10 ms units)	If a scan time value is greater than the value in SD1019, the SD1019 value is updated in END processing. Therefore the maximum value of scan time is stored in SD1019.	
D9020	SD1020	×	Constant scan	Constant scan time (User sets in 10 ms units)	This register stores an interval value in units of 10ms to run a program at regular intervals. • 0: No constant scan function • 1 to 200: Constant scan function available (executing at a interval of setting value × 10ms)	Qn(H) QnPH
D9021	SD1021	-	Scan time	Scan time (1 ms units)	This register stores scan time in every END processing.	Qn(H) QnPH QnU ^{*1} LCPU
D9022	SD1022	SD412	Count in units of 1s.	Count in units of 1s.	• The value is incremented by one every second after RUN. • The counter starts the routine: counts up from 0 to 32767, then counts down to -32768 and then again counts up to 0.	

ACPU Special Register	Special Register after Conversion	Special Register for Modification	Name	Meaning	Details	Corres- ponding CPU															
D9025	SD1025	-	Clock data	Clock data (year, month)	This register stores the last two digits of the year and the month in BCD as shown below. b15 to b12b11 to b8b7 to b4b3 to b0 Year Month Example: 1987, July H8707	Qn(H) QnPH QnU*1 LCPU															
D9026	SD1026	-	Clock data	Clock data (day, hour)	This register stores the day and the hour in BCD as shown below. b15 to b12b11 to b8b7 to b4b3 to b0 Day Hour Example: 31st, 10 a.m. H3110																
D9027	SD1027	-	Clock data	Clock data (minute, second)	This register stores the minute and the second in BCD as shown below. b15 to b12b11 to b8b7 to b4b3 to b0 Minute Second Example: 35 min, 48 sec. H3548																
D9028	SD1028	-	Clock data	Clock data (day of week)	This register stores the day of the week in BCD as shown below. b15 to b12b11 to b8b7 to b4b3 to b0 Always set "0" Example: Friday H0005 <table><tr><th colspan="2">Day of the week</th></tr><tr><td>0</td><td>Sunday</td></tr><tr><td>1</td><td>Monday</td></tr><tr><td>2</td><td>Tuesday</td></tr><tr><td>3</td><td>Wednesday</td></tr><tr><td>4</td><td>Thursday</td></tr><tr><td>5</td><td>Friday</td></tr><tr><td>6</td><td>Saturday</td></tr></table>		Day of the week		0	Sunday	1	Monday	2	Tuesday	3	Wednesday	4	Thursday	5	Friday	6
Day of the week																					
0	Sunday																				
1	Monday																				
2	Tuesday																				
3	Wednesday																				
4	Thursday																				
5	Friday																				
6	Saturday																				
D9035	SD1035	SD648	Extension file register	Use block No.	Stores the block No. of the extension file register being used in BCD code.	Qn(H) QnPH															
D9036	SD1036	×	Extension file registerfor designation of device number	Device number when individual devices from extension file register are directly accessed	Designate the device number for the extension file register for direct read and write in 2 words at SD1036 and SD1037 in BIN data.																
D9037	SD1037	×			Use consecutive numbers beginning with R0 of block No. 1 to designate device numbers.																

SD1037,SD1036

Device No. (BIN data)

0 to 16383

16384 to

to

Extension file register

Block No.1 area

Block No.2 area

ACPU Special Register	Special Register after Conversion	Special Register for Modification	Name	Meaning	Details	Corres-ponding CPU														
D9038	SD1038	SD207	LED display priority ranking	Priorities 1 to 4	- This register stores priority of errors to be indicated by the ERROR LED (on or flash) by using cause numbers. - Configuration of the priority setting areas is as shown below. <table><tr><td>b15 to b12</td><td>b11 to b8</td><td>b7 to b4</td><td>b3 to b0</td></tr><tr><td>SD207</td><td>Priority 4</td><td>Priority 3</td><td>Priority 2</td><td>Priority 1</td></tr><tr><td>SD208</td><td colspan="2">Priority 7</td><td>Priority 6</td><td>Priority 5</td></tr></table> - For details, refer to the following. User's manual of the CPU module used Type ACPU/QCPU-A (A Mode) Programming Manual (Fundamentals)	b15 to b12	b11 to b8	b7 to b4	b3 to b0	SD207	Priority 4	Priority 3	Priority 2	Priority 1	SD208	Priority 7		Priority 6	Priority 5	
b15 to b12	b11 to b8	b7 to b4		b3 to b0																
SD207	Priority 4	Priority 3	Priority 2	Priority 1																
SD208	Priority 7		Priority 6	Priority 5																
D9039	SD1039	SD208	Priorities 5 to 7																	
D9044	SD1044	×	For sampling trace	Step or time during sampling trace	To operate the STRA or STRAR instruction of a sampling trace by turning on or off SM803 with a programming tool, use the value stored in SD1044 as the sampling trace condition. - When "Each scan" is selected: 0 - When a timing is specified: setting value (Unit: 10ms)	Qn(H) QnPH														
D9049	SD1049	×	Work area for SFC	Block number of extension file register	- This register stores the block No. of the extended file register used as a work area for executing the SFC program. - This register stores "0" when SM320 is off and when empty area of 16K bytes or smaller is used (16K byte or less is too small to be used as block No.1 for an extended file register).															
D9050	SD1050	×	SFC program error number	Error code generated by SFC program	This register stores an error code of the error occurred in the SFC program. 0 : No error 80: SFC program parameter error 81: SFC code error 82: Number of steps of simultaneous execution exceeded 83: Block start error 84: SFC program operation error															
D9051	SD1051	×	Error block	Block number where error occurred	This register stores the number of the block in the SFC program where an error occurred. For error 83, the number of the block where the program was started is stored.															
D9052	SD1052	×	Error step	Step number where error occurred	- This register stores the number of the step in the SFC program where error 83 occurred. - For error 80, 81, and 82, "0" is stored. - For error 83, the block starting step number is stored.															
D9053	SD1053	×	Error transition	Transition condition number where error occurred	This register stores the number of the transition condition in the SFC program where error code 84 occurred. For error codes 80, 81, 82, and 83, "0" is stored.															
D9054	SD1054	×	Error sequence step	Sequence step number where error occurred	This register stores the sequence step number of transfer condition and operation output in the SFC program where error 84 occurred.															
D9055	SD1055	SD812	Status latch execution step number	Status latch execution step number	- This register stores the number of the step where a status latch was executed. - When a status latch was executed in a main sequence program, the step No. is stored. - When a status latch was executed in a SFC program, the block number and step number are stored. <table><tr><td>Block No. (BIN)</td><td>Step No. (BIN)</td></tr><tr><td>← Upper 8 bits →</td><td>← Lower 8 bits →</td></tr></table>		Block No. (BIN)	Step No. (BIN)	← Upper 8 bits →	← Lower 8 bits →										
Block No. (BIN)	Step No. (BIN)																			
← Upper 8 bits →	← Lower 8 bits →																			

ACPU Special Register	Special Register after Conversion	Special Register for Modification	Name	Meaning	Details	Corres- ponding CPU
D9072	SD1072	×	PLC communication check	Data check of serial communication module	The serial communication module automatically reads and writes data in a single loopback test to perform communication check.	Qn(H) QnPH
D9085	SD1085	×	Register for setting time check value	1 s to 65535 s	Sets the time check time of the data link instructions (ZNRD, ZNWR) for the MELSECNET/10. • Setting range: 1s to 65535s (1 to 65535) • Unit: second • Default: 10s (If 0 has been set)	
D9090	SD1090	×	Microcomputer subroutine input data area start device number	Depends on microcomputer package.	For details, refer to the following. Manual for respective microcomputer package	
D9091	SD1091	×	Detailed error code	Self-diagnosis detailed error code	This register stores description of the error cause of an instruction error.	Qn(H) QnPH QnU*1 LCPU
D9094	SD1094	SD251	Head I/O number of I/O module to be replaced	Head I/O number of I/O module to be replaced	This register stores the first two digits of the start I/O number of an I/O module, which is to be removed and mounted online (with power on). Example) Input module with I/O No. X2F0 → H2F	Qn(H) QnPH
D9095	SD1095	SD200	DIP switch information	DIP switch information	This register stores a status of the DIP switch of the CPU module in the following format. • 0: OFF • 1: ON D9095 b15 to b5 b4 b3 b2 b1 b0 0 <	

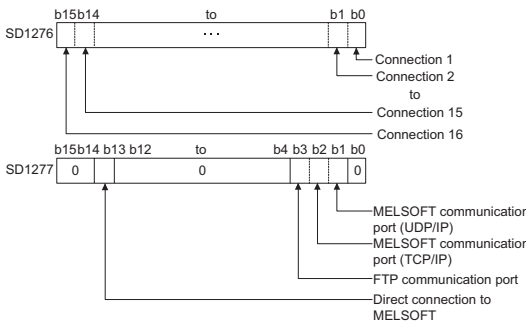
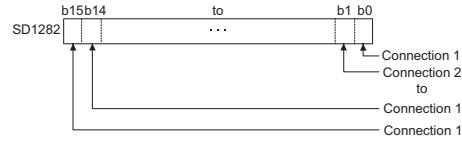
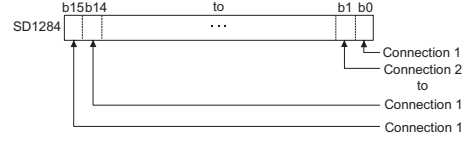
ACPU Special Register	Special Register after Conversion	Special Register for Modification	Name	Meaning	Details	Corresponding CPU
D9108	SD1108	-	Step transfer monitoring timer setting	Timer setting valve and the F number at time out	b15 to b8 b7 to b0 ↑ F number setting (02 to 255) ↑ Timer time limit setting (1 to 255s (1s units)) Turning on any of registers SM1108 to SM1114 activates a monitoring timer. If the transition condition for the step is not established before the time-out time, the annunciator (F) turns on.	Qn(H) QnPH
D9109	SD1109					
D9110	SD1110					
D9111	SD1111					
D9112	SD1112					
D9113	SD1113					
D9114	SD1114					
D9116	SD1116	-	I/O module verification error	Bit pattern, in units of 16 points, indicating the modules with verification errors	b15 b14 b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 XY 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 XY 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 XY 0 Indicates an I/O module verify error I/O module verification is conducted on I/O modules on remote I/O stations. (If normal status is restored, clear is not performed. Therefore, it is required to perform clear by user program.)	
D9117	SD1117					
D9118	SD1118					
D9119	SD1119					
D9120	SD1120					
D9121	SD1121					
D9122	SD1122					
D9123	SD1123					
D9124	SD1124	SD63	Number of annunciator detections	Number of annunciator detections	When any of F0 to F2047 (default device setting) is turned on by the SET F instruction, a value in SD1124 is incremented by one (up to a maximum of 16). When the RST F or LEDR instruction is executed, it is decremented by one.	
D9125	SD1125	SD64	Annunciator detection number	Annunciator detection number	SET SET SET SET RST SET SET SET SET SET SET SET SET LEDR F50 F25 F99 F25 F15 F70 F65 F38 F110 F151 F210 0 50 50 50 50 50 50 50 50 50 50 50 50 99 0 1 2 3 2 3 4 5 6 7 8 8 8 0 50 50 50 50 50 50 50 50 50 50 50 50 99 0 0 25 25 99 99 99 99 99 99 99 99 99 15 0 0 0 99 0 15 15 15 15 15 15 15 70 0 0 0 0 0 0 70 70 70 70 70 70 65 0 0 0 0 0 0 0 65 65 65 65 65 38 0 0 0 0 0 0 0 0 38 38 38 38 110 0 0 0 0 0 0 0 0 0 110 110 110 151 0 0 0 0 0 0 0 0 0 0 151 151 210	Qn(H) QnPH QnU*1 LCPU
D9126	SD1126	SD65				
D9127	SD1127	SD66				
D9128	SD1128	SD67				
D9129	SD1129	SD68				
D9130	SD1130	SD69				
D9131	SD1131	SD70				
D9132	SD1132	SD71				

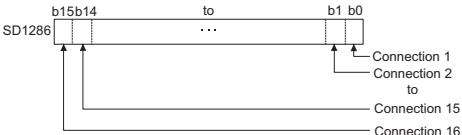
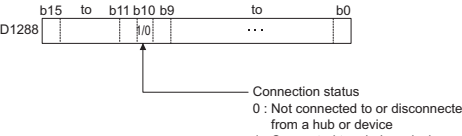
*1: The relevant modules are as follows:

- The Universal model QCPU whose serial number (first five digits) is "10102" or later.
- Q00UJCPU, Q00UCPU, Q01UCPU

(11) Built-in Ethernet port QCPU and built-in Ethernet function

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU																					
SD1260	IP address in- use	IP address (lower digits)	This register stores an IP address of the built-in Ethernet port.	S (Initial)		QnU* ³																					
SD1261		IP address (upper digits)																									
SD1262		Subnet mask pattern (lower digits)	• This register stores a subnet mask pattern of the built-in Ethernet port. • When a subnet mask pattern is not set, "0" is stored.																								
SD1263		Subnet mask pattern (upper digits)																									
SD1264		Default router IP address (lower digits)	• This register stores a default router IP address of the built-in Ethernet port. • When a default router IP address is not stored, "0" is stored.																								
SD1265		Default router IP address (upper digits)																									
SD1270	Operation result	Stores operation result.	This register stores the operation result of the time setting function. • 0: Not executed • 1: Success 0FFFF _H : Failure	S (Status change)	New	QnU* ¹ LCPU																					
SD1271	Execution time	Stores time acquired with time setting function.	This register stores the year (last two digits) and the month that the time setting function was executed in 2-digit BCD. b15 to b12b11 to b8b7 to b4b3 to b0 <table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table> Example: July, 1993 9307_H YearMonth																								
SD1272			This register stores the day and the hour that the time setting function was executed in a 2-digit BCD. b15 to b12b11 to b8b7 to b4b3 to b0 <table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table> Example: 31st, 10 a.m. 3110_H DayHour																								
SD1273	This register stores the minute and the second that the time setting function was executed in a 2-digit BCD. b15 to b12b11 to b8b7 to b4b3 to b0 <table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table> Example: 35 min., 48 sec. 3548_H MinuteSecond																										
SD1274			This register stores the year (first two digits) and the day of the week that the time setting was executed in 2-digit BCD. b15 to b12b11 to b8b7 to b4b3 to b0 <table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table> Example: 1993, Friday 1905_H Higher digits of year (0 to 99) <table><tr><th>Day of the week</th></tr><tr><td>0 Sunday</td></tr><tr><td>1 Monday</td></tr><tr><td>2 Tuesday</td></tr><tr><td>3 Wednesday</td></tr><tr><td>4 Thursday</td></tr><tr><td>5 Friday</td></tr><tr><td>6 Saturday</td></tr></table>																	Day of the week	0 Sunday	1 Monday	2 Tuesday	3 Wednesday	4 Thursday	5 Friday	6 Saturday
Day of the week																											
0 Sunday																											
1 Monday																											
2 Tuesday																											
3 Wednesday																											
4 Thursday																											
5 Friday																											
6 Saturday																											

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU
SD1275	Required response time	Stores time required for clock time aquisition.	This register stores the time required for a clock value to be set on the CPU after being sent to the SNTP server. • Range: A0 to 0FFFE _H (Unit: ms) 0FFFF _H when the above limit is exceeded.			
SD1276	Forced connection invalidation	Specifies forced connection invalidation.	This register is specified to forcibly invalidate a connection by a program. After being specified invalid, the connection stops communication and does not respond.(When a remote password is used and frequent unlock processing errors occur on a connection, this register is useful for temporarily disabling access.)  • 0: Valid (default) • 1: Invalid	U		QnU*1 LCPU
SD1277						
SD1282	Open completion signal	Stores open completion status	This register stores the open completion status of a socket communication connection. (when the open method is socket communication). Bits for connections other than that of socket communication are always "0".  • 0: Open processing is not completed. • 1: Open processing is completed.	S (Status change)	New	QnU*2 LCPU
SD1284	Open request signal	Stores open request status	This register stores the open request status of a socket communication connection. Bits for connections other than that of socket communication are always "0".  • 0: No open request • 1: In open request			

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU
SD1286	Reception status signal	Stores reception status	This register stores the receive status of a socket communication connection. Bits for connections other than that of socket communication are always "0".  • TCP (standard receive mode) • 0: Data not received • 1: Data received • TCP (fixed-length receive mode) • 0: Data not received or received data size is not the size of receive buffer. • 1: Received data size reached to the receive buffer size. • UDP • 0: Data not received • 1: Data received	S (Status change)		QnU* ² LCPU
SD1288	Built-in Ethernet port connection status	Stores connection status of built- in Ethernet port	This register stores a connection status of the built-in Ethernet port.  Connection status 0 : Not connected to or disconnected from a hub or device 1 : Connected to a hub or device		New	
SD1292	IP address setting	IP address (lower digits)	• Specify an IP address to be stored in the IP address storage area (flash ROM). Range: 00000001_H to DFFFFFFE_H (0.0.0.1 to 223.255.255.254)	S (Status change) /U		QnU* ³
SD1293		IP address (upper digits)	• When writing to or clearing the IP address storage area (flash ROM) is completed, the values of the IP address stored in the IP address storage area (flash ROM) are stored.			
SD1294		Subnet mask pattern (lower digits)	• Specify a subnet mask pattern to be stored in the IP address storage area (flash ROM). Range: C0000000_H to FFFFFFFC_H (192.0.0.0 to 255.255.255.252), 00000000_H (blank)			
SD1295		Subnet mask pattern (upper digits)	• When writing to or clearing the IP address storage area (flash ROM) is completed, the values of the subnet mask pattern stored in the IP address storage area (flash ROM) are stored.			
SD1296		Default router IP address (lower digits)	• Specify a default router IP address to be stored in the IP address storage area (flash ROM). Range: 00000001_H to DFFFFFFE_H (0.0.0.1 to 223.255.255.254), 00000000_H (blank)			
SD1297		Default router IP address (upper digits)	• When writing to or clearing the IP address storage area (flash ROM) is completed, the values of the default router IP address stored in the IP address storage area (flash ROM) are stored.			

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU
SD1298	IP address storage area write error factor	Stores error factor when failing to write to IP address storage area	This register stores an error factor occurred when writing to the IP address storage area (flash ROM). (Links with SM1294.) • 0 _H : No error • 100 _H : The values of SD1292 to SD1297 are out of the setting range. • 200 _H : Write error • 300 _H : Writing is not available because other function is being executed. • 400 _H : Writing is not available because the IP address storage area is being cleared	S (Status change)	New	QnU* ³
SD1299	IP address storage area clear error factor	Stores error factor when failing to clear IP address storage area	This register stores an error factor occurred when clearing the IP address storage area (flash ROM). (Links with SM1297.) • 0 _H : No error • 200 _H : Clear error • 300 _H : Clearing is not available because other function is being executed. • 400 _H : Clearing is not available because the IP address storage area is being written.			

*1 This applies to the Built-in Ethernet port QCPU.

*2 The built-in Ethernet port QCPU whose serial number (first five digits) is "11012" or later is targeted.

*3 This applies to the built-in Ethernet port QCPU whose first five digits of serial No. is "11082" or later.

(12) Fuse blown module

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU				
SD1300	Fuse blown module	Bit pattern in units of 16 points, indicating the modules whose fuses have blown 0: No blown fuse 1: Blown fuse present	- The number of an output module whose fuse has blown is stored in the following bit pattern (in units of 16 points). (If the module numbers are set by parameter, the parameter-set numbers are stored.) - The status of the blown fuse of an output module on a remote station is also detected. b15 b14 b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0 SD1300 0 0 0 1 (YCD) 0 0 0 1 (YBO) 0 0 0 0 0 0 0 0 SD1301 1 (Y1F0) 0 0 0 0 1 (Y1A0) 0 0 0 0 0 0 0 0 0 SD1331 0 0 0 0 1 (Y1F30) (Y1F30) 0 0 0 0 0 0 0 1 (Y1F30) (Y1F30) 0 0 0 Indicates fuse blow - Not cleared even if the blown fuse is replaced with a new one. The numbers are cleared by clearing the error.	S (Error)	D9100	Qn(H) QnPH QnPRH QnU				
SD1301					D9101					
SD1302					D9102					
SD1303					D9103					
SD1304					D9104					
SD1305					D9105					
SD1306					D9106					
SD1307					D9107					
SD1308									New	
SD1309 to SD1330										
SD1331										

(13) I/O module verification

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU																																																																				
SD1400	I/O module verify error	Bit pattern, in units of 16 points, indicating the modules with verification errors. 0: No I/O verification errors 1: I/O verification error present	• If the status of the I/O module changes from that obtained at power-on, the module No. is stored in the following bit pattern.(If the I/O numbers are set by parameter, the parameter-set numbers are stored.) • Also detects I/O module information. <table><tr><td></td><td>b15</td><td>b14</td><td>b13</td><td>b12</td><td>b11</td><td>b10</td><td>b9</td><td>b8</td><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr><tr><td>SD1400</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1(X,Y) 0</td></tr><tr><td>SD1401</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1(X,Y) 190</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>SD1431</td><td>0</td><td>1(X,Y) YFE9</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table> Indicates an I/O module verification error • Not cleared even if the blown fuse is replaced with a new one. This flag is cleared by error resetting operation.		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	SD1400	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1(X,Y) 0	SD1401	0	0	0	0	0	0	1(X,Y) 190	0	0	0	0	0	0	0	0	0	SD1431	0	1(X,Y) YFE9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	S (Error)	D9116	Qn(H) QnPH QnPRH QnU LCPU
				b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0																																																							
SD1400				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1(X,Y) 0																																																							
SD1401				0	0	0	0	0	0	1(X,Y) 190	0	0	0	0	0	0	0	0	0																																																							
SD1431				0	1(X,Y) YFE9	0	0	0	0	0	0	0	0	0	0	0	0	0	0																																																							
SD1401				D9117																																																																						
SD1402				D9118																																																																						
SD1403				D9119																																																																						
SD1404				D9120																																																																						
SD1405				D9121																																																																						
SD1406	D9122																																																																									
SD1407	D9123																																																																									
SD1408	New																																																																									
SD1409 to SD1430																																																																										
SD1431																																																																										

(14) Process control instruction

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU									
SD1500 SD1501	Basic period	Basic period time	Set the basic period (1 second units) use for the process control instruction using floating point data. Floating points data = <table><tr><td>SD1501</td><td>SD1500</td></tr></table>	SD1501	SD1500	U	New	QnPH							
SD1501	SD1500														
SD1502	Process control instruction detail error code	Process control instruction detail error code	This register indicates the details of an error occurred by executing a process control instruction.	S (Error)											
SD1503	Process control instruction generated error location	Process control instruction generated error location	Shows the error process block that occurred in the process control instruction.												
SD1506 SD1507	Dummy device	Dummy device	Used to specify dummy devices by a process control instruction.	U	QnPH QnPRH										
SD1508	Function availability selection for process control instruction	b0 Bumpless function availability setting for the S.PIDP instruction 0: Enabled 1: Disabled (Default: 0)	This register stores whether to enable functions for process control instructions. SD1508 <table><tr><td>b15</td><td>b14</td><td>to</td><td>b2</td><td>b1</td><td>b0</td></tr><tr><td>0</td><td></td><td>0</td><td></td><td>0</td><td>1/0</td></tr></table> ↑ Bumpless function availability for the S.PIDP instruction			b15	b14	to	b2	b1	b0	0		0	
b15	b14	to	b2	b1	b0										
0		0		0	1/0										

(15) Redundant system (host system CPU information *1)

The special register (SD1510 to SD1599) is valid only for redundant systems. All bits are set to "0" for stand-alone systems.

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU
SD1585	Redundant system LED status	4 LED states • BACKUP • CONTROL • SYSTEM A • SYSTEM B	The LED status of BACKUP, CONTROL, SYSTEM A, or SYSTEM B is stored in the following format: b15 to b10 b9 b8 b7 b6 b5 b4 b3 b2 to b0 SYSTEM B 0: Off 1: On 2: Flicker SYSTEM A 0: Off 1: On 2: Flicker BACKUP 0: Off 1: On (red) 2: Flicker (red) 3: On (green) 4: Flicker (green) CONTROL 0: Off 1: On	S (Status change)	New	QnPRH

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU
SD1588	Reason(s) for system switching	Reason(s) for system switching that occurred in host station	Stores the reason(s) for system switching on the host system. The following values are stored corresponding to the methods for system switching: This register is initialized with zero (0) stored when the system is powered on from off or is reset. • 0: Initial value (control system has never been switched) • 1: Power off, Reset, H/W failure, WDT error • 2: CPU stop error (except WDT) • 3: System switching request from network module • 16: System switching dedicated instruction • 17: System switching request from a programming tool	S (when condition occurs)	○	QnPRH
SD1589	Reason(s) for system switching failure conditions	Reason(s) for system switching failure No	• If a system switching is failed, any of the following value is stored in this register. • 0: System switching complete (default) • 1: Tracking cable is not connected, tracking cable failure, or internal circuit failure • 2: H/W failure, power-off, reset, watchdog timer error on the standby system • 3: H/W failure, power-off, reset, WDT error on the control system • 4: Preparing tracking communication • 5: Communication timeout • 6: Stop error on the standby system (except for watchdog timer error) • 7: Operation differs between both systems (detected only in the back up mode) • 8: During memory copy from control system to standby system • 9: Performing program online change • 10: Detecting a failure of network module on the standby system • 11: System is being switched • Resets to "0" when host system is powered on. • Zero is stored in this register upon completion of system switching.	S (when system is switched)	○	
SD1590	Network module head address, which requested system switching	Network module head address, which requested system switching	• When system switching is requested from a network module in the host system, the bit corresponding to the module that received the request turns on. Each bit 0:OFF 1:ON Module 0: CPU module is invalid as it is 2-slot model Module 1: Module on right side of CPU module to Module11: Module at rightmost end of 12-slot base (Q312B) • The system turns off the bit after the error is removed by a user. • For the number for modules where system switching is requested from a network module in other system, refer to SD1690.	S (Error/Status change)	New	
SD1595	Memory copy target I/O number	Memory copy target I/O number	• Before SM1595 is turned from off to on, the I/O No. of the memory copy destination (Standby system CPU module: 3D1_H) is stored in this register.	U		
SD1596	Memory copy status	Memory copy status	• Stores the execution result of Memory copy function. • 0: Memory copy is complete • 4241_H: Standby system power supply off • 4242_H: Tracking cable is disconnected or is damaged • 4247_H: Memory copy is being executed • 4248_H: Unsupported memory copy destination I/O number	S (Status change)		

*1 The information of the host CPU module is stored.

(16) Redundant system (other system CPU information*1)

The special register (SD1600 to SD1650) is valid when the redundant system is in backup mode and is invalid in separate mode. The special register (SD1651 to SD1690) is valid when the redundant system is in backup mode or in separate mode. All bits in SD1600 to SD1690 are set to "0" for stand-alone systems.

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU SD□□*2	Corres- ponding CPU
SD1600	System error information	System error information	- If an error is detected by the error check for redundant system, the corresponding bit shown below turns on. That bit turns OFF when the error is cleared after that. - Each bit 0: OFF 1: ON - Tracking cable is not connected or damaged - Power-OFF, reset, watchdog timer error or hardware failure occurred in other system - Other system stop error (except watchdog timer error) - Bit turns on when failing to connect with other system. The following causes are shown below: • Tracking H/W failure • Host system WDT error • Cannot recognize other system therefore causing error - If any of b0, b1, b2 and b15 is on, the other bits are off. - In the debug mode, b0, b1, b2 and b15 are all off.	S (Every END processing)		
SD1601	System switching results	System switching results	Reason(s) for system switching is stored. - When a system is switched, the reason for system switching is stored in SD1601 of both systems. - This register is initialized with zero (0) stored when the system is powered on from off or is reset. - The following shows the values stored in this register. 0: Initial value (control system has never been switched) 1: Power-off, reset, H/W failure, or watchdog timer error *1 2: Stop error (except for watchdog timer error) 3: A system switching request from network module 16: Control system switching instruction 17: System switching request from a programming tool *1: When the system is switched upon the power-off or reset of the control system, "1" is not stored in SD1601 of the new standby system.	S (when system is switched)		QnPRH
SD1602	System switching dedicated instruction parameter	System switching dedicated instruction parameter	- This register stores the argument to the instruction when a system is switched by the SP.CONTSW instruction. (The argument for the SP.CONTSW instruction is stored in SD1602 of both systems upon system switching.) - SD1602 is only valid when "16" is stored in SD1601. - SD1602 is updated only when a system is switched by the control system switching instruction.			
SD1610	Other system diagnostic error	Diagnostic error code	- This register stores an error code for the error occurred on other system. - The value in SD0 of the CPU module on other system is reflected.	S (Every END processing)	SD0	
SD1611	Other system diagnostic error occurrence time	Diagnostic error occurrence time	- Stores the date and time when diagnostics error occurred corresponding to error code stored in SD1610. - Data format is the same as SD1 to SD3. - The values in SD1 to SD03 of the CPU module on other system are reflected.	S (Every END processing)	SD1 to SD3	
SD1612						
SD1613						

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU SD□□'2	Corres- ponding CPU
SD1614	Other system error information category	Error information category code	- This register stores the category code of error information and individual information of the error that occurred on other system. - Data format is the same as SD4. - The value in SD4 of the CPU module on other system is reflected.	S (Every END processing)	SD4	QnPRH
SD1615 to SD1625	Other system error common information	Error common information	- Stores the common information corresponding to the error code stored in this system CPU. - Data composition is the same as SD5 to SD15. - The values in SD5 to SD15 of the CPU module on other system are reflected.		SD5 to SD15	
SD1626 to SD1636	Other system error individual information	Error individual information	- Stores the individual information corresponding to the error code stored in this system CPU. - Data composition is the same as SD16 to SD26. - The values in SD16 to SD26 of the CPU module on other system are reflected.		SD16 to SD26	
SD1649	Standby system error cancel command	Error code of the error to be cleared	- This register stores the error code of the error to clear by clearing a standby system error. - The standby system error is cleared by turning SM1649 from off to on after storing the error code of the error to clear. - The least significant digit (ones place) of the error code in this register is ignored. (The errors corresponding to error codes 4100 to 4109 are cleared by storing 4100 in this register.)		-	
SD1650	Other system operating information	Other system operating information	This register stores the operating status of the CPU module on the other system in the following bit pattern. When communications with other systems are disabled or the system is in the debug mode, "00FFH" is stored. b15 to b8 SD1650 0 b7 to b4 b3 to b0 0: No error 1: Continue error 2: Stop error F: Communication with other system disabled *1 * 1 Communication with other system disabled, debug mode 0: RUN 2: STOP 3: PAUSE F: Communication with other system disabled *1 Note: Communications are disabled in the following states. - The other system is powered off or is being reset. - A hardware failure has occurred on the host or the other system. - A watchdog timer error has occurred on the host or the other system. - Tracking cable is not connected. Tracking cable is broken or failed.			

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU SD□□*2	Corres- ponding CPU
SD1690	Network module head address, which requested system switching on host (control) system	Network module head address, which requested system switching on host (control) system	• Stores head address of network module which a system switch request was initiated, using the following format. Each bit 0:OFF 1:ON b15 to b11 to b1 b0 SD1690 0 0/1 ... 0/1 0 Module 0: CPU module is invalid as it is 2-slot model Module 1: Module on right side of CPU module to Module11: Module at rightmost end of 12-slot base (Q312B) • Turns off automatically by system, after network error is reset by user. • To find the number for the module where system switching is requested from a network module in the host system, refer to SD1590.	S (Every END processing)	-	QnPRH

*1 The diagnostic information and system information of other CPU modules are stored.

*2 Shows the special register (SD□□) for the host system CPU.

(17) Redundant system (tracking information)

The special register (SD1700 to SD1779) is valid only for redundant systems. All bits are set to "0" for stand-alone systems.

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU
SD1700	Tracking error detection count	Tracking error detection count	- A value in this register is incremented by one upon tracking error. - The counter starts the routine: counts up from 0 to 32767, then counts down to -32767 and then again counts up to 0.	S (Error)		
SD1710	Waiting time for online program change (standby system)	Waiting time for online program change (standby system)	- This register stores the waiting time required for starting the online program change in the standby system after completion of that in the control system. The value is specified in units of seconds. - If online program change is not requested even after it is completed in the control system, the CPU modules in both of the system determine that it is a failure of an online program change for redundancy. In this case, both system CPU modules resume the consistency check for the systems that have been on hold during the online program change. Also, the control system is set to accept another request of online program change for redundancy. - When both systems are powered on, 90 seconds are set to SD1710 as the default value. - Set the value within the range 90 to 3600 seconds. When the setting is 0 to 89 seconds, it is regarded as 90 seconds for operation. If the setting is outside the allowed range, it is regarded other than 0 to 3600 seconds for operation. - The waiting time for a start of online program change to the standby system CPU module is checked according to the SD1710 setting during online change of multiple blocks and online change of batch of files for redundancy.	S (Initial) /U	New	QnPRH

(18) Redundant power supply module information

The special register (SD1780 to SD1789) is valid only for redundant power supply systems. All bits are set to "0" for stand-alone power supply systems.

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU
SD1780	Power supply off detection status	Power supply off detection status	- This register stores status of the redundant power supply module (Q63RP and Q64RP) with input power off, in the following bit pattern. Input power OFF detection status of power supply 2^{*1} Input power OFF detection status of power supply 1^{*1} Each bit b15 to b8 b7 to b0 0: Input power ON status/ No redundant power supply module 1: Input power OFF status SD1780 Main base unit 1st extension base : 7th extension base Main base unit 1st extension base : 7th extension base - When the main base unit is not the redundant power main base unit (Q38RB), "0" is stored. - When configuring multiple CPU, the status is stored to 1st CPU module.	S (Every END processing)	New	Qn(H) ^{*2} QnPH ^{*2} QnPRH QnU ^{*3}

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU
SD1781	Power supply failure detection status	Power supply failure detection status	- This register stores failure detection status of the redundant power supply module (Q63RP and Q64RP) in the following bit pattern.(After a failure is detected on a redundant power supply module, the bit corresponding to the failed module turns to "0" upon turning off the module.) - When the main base unit is not the redundant power main base unit (Q38RB), "0" is stored. - When configuring multiple CPU, the status is stored to 1st CPU module.	S (Every END processing)	New	Qn(H) ^{*2} QnPH ^{*2} QnPRH QnU ^{*3}
SD1782	Momentary power failure detection counter for power supply 1 ^{*1}	Momentary power failure detection count for power supply 1	- Counts the number of times of momentary power failure of the power supply 1/2. - Monitors the status of the power supply 1/ 2 mounted on the redundant power main base unit (Q38RB) and counts the number of times of momentary power failure.			
SD1783	Momentary power failure detection counter for power supply 2 ^{*1}	Momentary power failure detection count for power supply 2	- Status of power supply 1/power supply 2 mounted on the redundant extension base unit is not monitored. - When the CPU module starts, the counter of the power supply 1/ 2 is cleared to 0. - If the input power to one of the redundant power supply modules is turned OFF, the corresponding counter is cleared to 0. - The counter is incremented by one upon momentary power failure on the power supply 1 or 2.(The counter repeats increment and decrement of the value; 0 → 32767 → - 32768 → 0 .(The value is displayed within the range of 0 to 65535 in the system monitor screen of programming tool.)) - Stores 0 when the main base unit is not the redundant power main base unit (Q38RB). - When configuring multiple CPU, the status is stored to 1st CPU module.			

*1 The "power supply 1" indicates the redundant power supply module mounted on the POWER 1 slot of the redundant base unit (Q38RB/68RB/Q65WRB).

The "power supply 2" indicates the redundant power supply module mounted on the POWER 2 slot of the redundant base unit (Q38RB/68RB/Q65WRB).

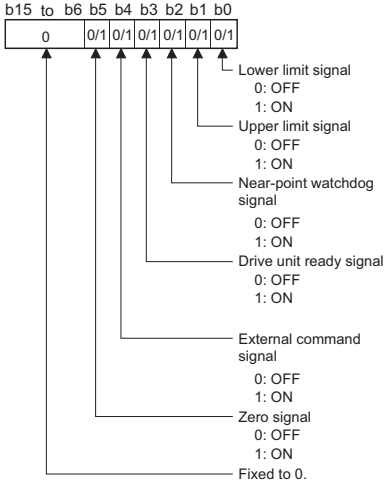
*2 The module whose first 5 digits of serial No. is "07032" or later. (However, for the multiple CPU system configuration, this applies to all CPU modules whose first 5 digits of serial No. are "07032" or later.)

*3 The module whose first 5 digits of serial No. is "10042" or later.

(19)Built-in I/O function

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU
SD1830 SD1831 SD1832	Functions selected for input signals	Functions selected for input signals	This register stores values indicating functions assigned for input signals. Each signal has 4-bit data. Values are stored according to functions as follows: • 0: General-purpose input function • 1: Interrupt input function • 2: Pulse catch function • 3: Positioning function • 4: High-speed counter function			
SD1833						
SD1834						
SD1835						
SD1834 SD1835	Functions selected for output signals	Functions selected for output signals	This register stores values indicating functions assigned for output signals. Each signal has 4-bit data. Values are stored according to functions as follows: • 0: General-purpose output function • 3: Positioning function • 4: High-speed counter function	S (Every END processing)	New	LCPU
SD1836			This register stores the operating status of the positioning function and high-speed counter function. Operating status of positioning and high-speed counter functions			

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU
SD1840	Axis 1 current feed value	Current feed value	This register stores the current position value when the position where OPR control is completed is set as a base point. "0" is stored at power-on or reset of the CPU module. An OP address is stored at the completion of machine OPR control. This register is cleared to "0" when speed control in speed/position switching control is started. - When the current feed value is changed, the value after current value change is stored. - The current position read from a servo amplifier is stored at the completion of absolute position restoration.*1*2 *1 Range: -2147483648 to 2147483647 pulses *2 Since the internal update cycle of the storage value is 1ms, the information of the current feed value may be older than the actual command position by 1ms at maximum depending on the refresh timing at END processing.	S (Every END processing)	New	LCPU
SD1841						
SD1842	Axis 1 current speed	Current speed	This register stores the current speed. (Fractions are not stored. If the current speed is slower than 1 pulse/s, "0" may be displayed.) *1*2 *1 Range: 0 to 200000 pulses *2 Since the internal update cycle of the storage value is 1ms, the information of the current speed value may be older than the actual command position by 1ms at maximum depending on the refresh timing at END processing.			
SD1843						
SD1844	Axis 1 axis operation status	Axis operation status	This register stores the axis operating status. -1: Error occurring 0: Standing by 1: Stopped 2: In JOG operation 3: In OPR 4: In position control 5: In speed-position control (speed) (speed control in speed/position switching control) 6: In speed-position control (position) (position control in speed/position switching control) 7: Decelerating (axis stop ON) 8: Decelerating (JOG start OFF) 9: In high-speed OPR 10: In speed control 11: Analyzing			
SD1845	Axis 1 error code	Error code	- This register stores the error code of the present axis error. - If another error occurs while an axis error occurs, the latest error code is not stored. - This register is cleared to "0" by turning on SM1850.			
SD1846	Axis 1 warning code	Warning code	- This register stores the warning code of the present axis warning. - If another axis warning occurs, the latest warning code is written to this register. - This register is cleared to "0" by turning on SM1850.			

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU
SD1847	Axis 1 external I/O signal	External I/O signals	- This register stores the on/off status of external I/O signals. - When an OPR method with the OPR retry function (Near-point dog method, Count 1, Count 2) is performed, a value indicating the status of the upper limit signal or the lower limit signal is stored to the external command signal.*1  *1 When an OPR method with the OPR retry function is performed even once, a value indicating the status of the upper limit signal or the lower limit signal will be stored until when speed/position switching control is started.	S (Every END processing)	New	LCPU
SD1848	Axis 1 movement amount after near-point dog ON	Movement amount after near-point dog ON	- This register stores "0" when machine OPR control is started. - After machine OPR control is started, this register stores a travel distance from the point where the near-point watchdog signal turns on (The point is set to "0".) to the point where machine OPR control is completed.*1 - When an OPR method is set to Stopper 3, this register always stores "0". *1 Range: 0 to 2147483647 pulses			
SD1849						
SD1850	Axis 1 Data No. of positioning being executed	Data No. of positioning being executed	- This register stores the data No. of positioning being executed. (A storage value will be held until the next control starts.) "0" is stored when JOG operation or machine OPR control is started. "1" is stored when high-speed OPR control is started. "1" is stored when positioning control is started by the IPDSTRT1 or IPDSTRT2 instruction. - If an error occurs at the start of positioning control, the previous value will be held.			

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU
SD1860	Axis 2 current feed value	Current feed value	This register stores the current position value when the position where OPR control is completed is set as a base point. "0" is stored at power-on or reset of the CPU module. An OP address is stored at the completion of machine OPR control. This register is cleared to "0" when speed control in speed/position switching control is started. - When the current feed value is changed, the value after current value change is stored. - The current position read from a servo amplifier is stored at the completion of absolute position restoration.*1*2 *1 Range: -2147483648 to 2147483647 pulses *2 Since the internal update cycle of the storage value is 1ms, the information of the current feed value may be older than the actual command position by 1ms at maximum depending on the refresh timing at END processing.	S (Every END processing)	New	LCPU
SD1861						
SD1862	Axis 2 current speed	Current speed	This register stores the current speed. (Fractions are not stored. If the current speed is slower than 1 pulse/s, "0" may be displayed.) *1*2 *1 Range: 0 to 200000 pulses *2 Since the internal update cycle of the storage value is 1ms, the information of the current speed value may be older than the actual command position by 1ms at maximum depending on the refresh timing at END processing.			
SD1863						
SD1864	Axis 2 axis operation status	Axis operation status	This register stores the axis operating status. -1: Error occurring 0: Standing by 1: Stopped 2: In JOG operation 3: In OPR 4: In position control 5: In speed-position control (speed) (speed control in speed/position switching control) 6: In speed-position control (position) (position control in speed/position switching control) 7: Decelerating (axis stop ON) 8: Decelerating (JOG start OFF) 9: In high-speed OPR 10: In speed control 11: Analyzing			
SD1865	Axis 2 error code	Error code	- This register stores the error code of the present axis error. - If another error occurs while an axis error occurs, the latest error code is not stored. - This register is cleared to "0" by turning on SM1870.			
SD1866	Axis 2 warning code	Warning code	- This register stores the warning code of the present axis warning. - If another axis warning occurs, the latest warning code is written to this register. - This register is cleared to "0" by turning on SM1870.			

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU														
SD1867	Axis 2 external I/O signals	External I/O signals	- This register stores the on/off status of external I/O signals. - When an OPR method with the OPR retry function (Near-point dog method, Count 1, Count 2) is performed, a value indicating the status of the upper limit signal or the lower limit signal is stored to the external command signal.**1 <table border="1" style="margin: auto;"> <tr> <td>b15 to b6</td> <td>b5</td> <td>b4</td> <td>b3</td> <td>b2</td> <td>b1</td> <td>b0</td> </tr> <tr> <td>0</td> <td>0/1</td> <td>0/1</td> <td>0/1</td> <td>0/1</td> <td>0/1</td> <td>0/1</td> </tr> </table> Lower limit signal 0: OFF 1: ON Upper limit signal 0: OFF 1: ON Near-point watchdog signal 0: OFF 1: ON Drive unit ready signal 0: OFF 1: ON External command signal 0: OFF 1: ON Zero signal 0: OFF 1: ON Fixed to 0. *1 When an OPR method with the OPR retry function is performed even once, a value indicating the status of the upper limit signal or the lower limit signal will be stored until when speed/position switching control is started.	b15 to b6	b5	b4	b3	b2	b1	b0	0	0/1	0/1	0/1	0/1	0/1	0/1	S (Every END processing)	New	LCPU
b15 to b6	b5	b4	b3	b2	b1	b0														
0	0/1	0/1	0/1	0/1	0/1	0/1														

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU																		
SD1868	Axis 2 movement amount after near-point dog ON	Movement amount after near-point dog ON	- This register stores "0" when machine OPR control is started. - After machine OPR control is started, this register stores a travel distance from the point where the near-point watchdog signal turns on (The point is set to "0".) to the point where machine OPR control is completed.*1 - When an OPR method is set to Stopper 3, this register always stores "0". *1 Range: 0 to 2147483647 pulses	S (Every END processing)																				
SD1869																								
SD1870	Axis 2 Data No. of positioning being executed	Data No. of positioning being executed	- This register stores the data No. of positioning being executed. (A storage value will be held until the next control starts.) "0" is stored when JOG operation or machine OPR control is started. "1" is stored when high-speed OPR control is started. "1" is stored when positioning control is started by the IPDSTR1 or IPDSTR2 instruction. - If an error occurs at the start of positioning control, the previous value will be held.																					
SD1880	CH1 current value	CH1 current value	- This register stores the current counter value of CH1 at END processing. - When the ICCNTRD1 instruction is executed, this register is updated by the current value at that moment. The current value is updated at END processing and by the ICCNTRD1 instruction only when Normal Mode is set for Operation Mode Setting (high-speed counter function parameter). The range of a value that can be read is from -2147483648 to 2147483647.	S (Every END processing/ Instruction execution)	New	LCPU																		
SD1881																								
SD1882	CH1 status monitor	CH1 status monitor	This register stores each status of CH1. <table><tr><td>b15 to b8</td><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr><tr><td>0</td><td>0/1</td><td>0/1</td><td>0/1</td><td>0/1</td><td>0/1</td><td>0/1</td><td>0/1</td><td>0/1</td></tr></table> - Addition/subtraction 0: Addition 1: Subtraction - Underflow detection flag 0: Not detected 1: Detected - Overflow detection flag 0: Not detected 1: Detected - Sampling flag 0: Not operating 1: Operating - Frequency measurement flag 0: Not operating 1: Operating - Rotation speed measurement flag 0: Not operating 1: Operating - Pulse measurement flag 0: Not operating 1: Operating - PWM output flag 0: Not operating 1: Operating - Fixed to 0.	b15 to b8	b7	b6	b5	b4	b3	b2	b1	b0	0	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	S (Every END processing)		
b15 to b8	b7	b6	b5	b4	b3	b2	b1	b0																
0	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1																

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU
SD1883	CH1 external I/O status monitor	CH1 external I/O status monitor	- This register stores a value indicating the external I/O signal status of CH1. - Unused signal status is fixed at off. - When Normal Mode is set for Operation Mode Setting (high-speed counter function parameter), a value according to the setting configured for Function Input Logic Setting (high-speed counter function parameter) is stored in the function input status. Therefore, when a voltage is applied to the function input terminal while Negative logic is set for Function input logic setting, this register turns off. - When other than A Phase/B Phase is selected for Count Source Selection (high-speed counter function parameter), the phase A input status and phase B input status are fixed at off.	S (Every END processing)	New	LCPU

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU
SD1884	CH1 operation mode monitor	CH1 operation mode monitor	This register stores a value indicating the operation mode for high-speed counter of CH1 set by the parameter. • 0: Not used • 1: Normal mode • 2: Frequency measurement mode • 3: Rotation speed measurement mode • 4: Pulse measurement mode • 5: PWM output mode	S (Every END processing)	New	LCPU
SD1885	CH1 counter type monitor	CH1 counter type monitor	• This register stores a value indicating the counter type for high-speed counter of CH1 set by the parameter. • Counter selection is disabled (fixed at "0") when a value stored to CH1 operation mode monitor (SD1884) is other than "1" (normal mode). • 0: Linear counter • 1: Ring counter			
SD1886	CH1 selected counter function	CH1 selected counter function	• This register stores a value indicating the selected counter function for high-speed counter of CH1 set by the parameter. • Counter selection is disabled (fixed at "0") when a value stored to CH1 operation mode monitor (SD1884) is other than "1" (normal mode). • 0: Count disabling function • 1: Latch counter function • 2: Sampling counter function • 3: Count disabling/preset function • 4: Latch counter/preset function			
SD1887	CH1 error code	CH1 error code	This register stores the error code of an error occurred in CH1.			
SD1888	CH1 warning code	CH1 warning code	This register stores the warning code of a warning occurred in CH1.	S (Every END processing/ Instruction execution)		
SD1900	CH2 current value	CH2 current value	• This register stores the current counter value of CH2 at END processing. • When the ICCNTRD2 instruction is executed, this register is updated by the current value at that moment.			
SD1901			The current value is updated at END processing and by the ICCNTRD2 instruction only when Normal Mode is set for Operation Mode Setting (high-speed counter function parameter). The range of a value that can be read is from -2147483648 to 2147483647.			

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU
SD1902	CH2 status monitor	CH2 status monitor	This register stores each status of CH2.			
SD1903	CH2 external I/O status monitor	CH2 external I/O status monitor	• This register stores a value indicating the external I/O signal status of CH2. • Unused signal status is fixed at off. • When Normal Mode is set for Operation Mode Setting (high-speed counter function parameter), a value according to the setting configured for Function Input Logic Setting (high-speed counter function parameter) is stored in the function input status. Therefore, when a voltage is applied to the function input terminal while Negative logic is set for Function input logic setting, this register turns off. • When other than A Phase/B Phase is selected for Count Source Selection (high-speed counter function parameter), the phase A input status and phase B input status are fixed at off.	S (Every END processing)	New	LCPU

Number	Name	Meaning	Explanation	Set by (When Set)	Corres- ponding ACPU D9□□□	Corres- ponding CPU
SD1904	CH2 operation mode monitor	CH2 operation mode monitor	This register stores a value indicating the operation mode for high-speed counter of CH2 set by the parameter. • 0: Unused • 1: Normal mode • 2: Frequency measurement mode • 3: Rotation speed measurement mode • 4: Pulse measurement mode • 5: PWM output mode	S (Every END processing)	New	LCPU
SD1905	CH2 counter type monitor	CH2 counter type monitor	• This register stores a value indicating the counter type for high-speed counter of CH2 set by the parameter. • Counter selection is disabled (fixed at "0") when a value stored to CH2 operation mode monitor (SD1904) is other than "1" (normal mode). • 0: Linear counter • 1: Ring counter			
SD1906	CH2 selected counter function	CH2 selected counter function	• This register stores a value indicating the selected counter function for high-speed counter of CH2 set by the parameter. • Counter selection is disabled (fixed at "0") when a value stored to CH2 operation mode monitor (SD1904) is other than "1" (normal mode). • 0: Count disabling function • 1: Latch counter function • 2: Sampling counter function • 3: Count disabling/preset function • 4: Latch counter/preset function			
SD1907	CH2 error code	CH2 error code	This register stores the error code of an error occurred in CH2.			
SD1908	CH2 warning code	CH2 warning code	This register stores the warning code of a warning occurred in CH2.			

(20) Data logging

Number	Name	Meaning	Explanation	Set by (When Set)	Correspo nding ACPU D9□□□	Corresponding CPU
SD1940	Data logging setting No.1 Latest file No.	Latest file No.	This register stores the latest file number. This register is cleared to "0" by a stop command from LCPUCPU Logging Configuration Tool.	S (Status change)		
SD1941						
SD1942	Data logging setting No.1 Oldest file No.	Oldest file No.	This register stores the oldest file number. This register is cleared to "0" by a stop command from LCPUCPU Logging Configuration Tool.			
SD1943						
SD1944	Data logging setting No.1 Free buffer space	Free buffer space	This register stores free buffer space (unit: 1K byte). If the value is small, processing overflow may occur. For trigger logging, this register stores the buffer size until when data are collected by the number of records after trigger. This register is cleared to "0" by a stop command from LCPUCPU Logging Configuration Tool.			
SD1945	Data logging setting No.1 Processing timeout count	Number of times processing overflow occurred	This register stores the number of times that data logging processing overflow occurred. If an overflow occurs, some data may not be collected. When the storage value reaches to 65535, count is resumed from "0". If Stop is specified for Operation occurring when number of saved files is exceeded, processing overflow may occur from when data collection by the number of specified storage files is completed and until when data logging is stopped. This register is cleared to "0" by the registration of the setting or a stop command from LCPUCPU Logging Configuration Tool.	S (Error)	New	LCPUCPU
SD1946	Data logging setting No.1 Data logging error cause	Data logging error cause	This register stores the cause of an error that occurred during data logging. • 0: No error • Other than 0: For values stored at error occurrence, refer to the errors that occurs in data logging described in the following. ( MELSEC-L CPU Module User's Manual (Data Logging Function)) This register is cleared to "0" by the registration of the setting or a stop command from LCPUCPU Logging Configuration Tool.			

Number	Name	Meaning	Explanation	Set by (When Set)	Corresponding ACPU D9□□□	Corresponding CPU
SD1950 to SD1956	Data logging setting No.2	Same as in data logging setting No.1	Same as in data logging setting No.1 (SD1940 to SD1946)	Same as in data logging setting No.1	New	LCPUCPU
SD1960 to SD1966	Data logging setting No.3					
SD1970 to SD1976	Data logging setting No.4					
SD1980 to SD1986	Data logging setting No.5					
SD1990 to SD1996	Data logging setting No.6					
SD2000 to SD2006	Data logging setting No.7					
SD2010 to SD2016	Data logging setting No.8					
SD2020 to SD2026	Data logging setting No.9					
SD2030 to SD2036	Data logging setting No.10					

APPENDICES

Appendix 1 List of Parameter Numbers

Each parameter number will be stored in the special register (SD16 to SD26) when an error occurs in the parameter settings.

TableApp.1 lists the parameter items and corresponding parameter numbers.

For explanation of M and N shown in the "Parameter No." column, refer to Section 8.2.

TableApp.1 List of parameter numbers

Item		Parameter No.	Reference
Label		0000 _H	Section 8.1(1)
Comment		0001 _H	
I/O assignment	Type	0400 _H	Section 4.2.2, 8.1 (9)
	Model name		
	Points		
	Start XY (Start I/O number)		
Base setting	Base model name	0401 _H	Section 4.2.1, Section 8.1(10)
	Power model name		
	Extension cable		
	Slots		
Detailed setting	Error time output mode	0403 _H	Section 6.8, Section 8.1(10)
	I/O response time	0405 _H	Section 6.8, Section 8.1(10)
	Control PLC	0406 _H	Section 8.1 (9), QCPU User's Manual (Multiple CPU System)
Switch setting		0409 _H	Section 8.1(9)
No. of PLC		0E00 _H	Section 8.1 (12), QCPU User's Manual (Multiple CPU System)
Operating mode		0E01 _H	
I/O sharing when using Multiple CPUs	All CPUs can read all inputs	0E04 _H	
	All CPUs can read all outputs		
Timer limit setting	Low speed	1000 _H	Section 8.1(2)
	High speed		
RUN-PAUSE contacts	RUN	1001 _H	Section 6.6.1, Section 8.1(2)
	PAUSE		Section 6.6.2, Section 8.1(2)
Remote reset		1002 _H	Section 6.6.3, Section 8.1(2)
Output mode at STOP to RUN		1003 _H	Section 6.4, Section 8.1(2)
Common pointer No.		1005 _H	Section 9.10.2, Section 8.1(2)
Points occupied by empty slot		1007 _H	Section 4.4.2, Section 8.1(2)
Interrupt program/Fixed scan program setting		1008 _H	Section 2.2.3, 2.3.4, 8.1 (2)
System interrupt settings	Fixed scan interval (n : 28 to 31)		Section 8.1(2)
Intelligent function module setting (Interrupt pointer setting)		100A _H	Section 9.11, Section 8.1(2)
Module synchronization		100C _H	Section 8.1(2)

(To the next page)

TableApp.1 List of parameter numbers (continued)

Item		Parameter No.	Reference
Use serial communication		100EH	Section 6.23, Section 8.1(11)
Transmission speed			
Sum check			
Transmission wait time			
RUN write setting			
Service processing setting		1013H	Section 6.24.1, Section 8.1(2)
Latch data backup operation valid contact		1014H	Section 6.28, Section 8.1(2)
File register		1100H	Section 9.7, Section 8.1(3)
Comment file used in a command		1101H	Section 8.1(3)
Initial Device value		1102H	Section 6.25, Section 8.1(3)
File for local device		1103H	Section 9.14.2, Section 8.1(3)
Transfer to Standard ROM at Latch data backup operation		1104H	Section 6.28, Section 8.1(3)
File used for SP.DEVST/S.DEVLD instruction		1105H	Section 6.29, Section 8.1(6)
Device points		2000H	Section 9.1, Section 8.1(6)
Latch (1) start/end		2001H	Section 3.4, 6.3, Section 8.1(3)
Latch (2) start/end		2002H	
Local device start/end		2003H	Section 9.14.2, Section 8.1(6)
File register extended setting	Device points	2000H	Section 9.8, Section 8.1(6)
	Latch (1) start/end	2004H	
	Latch (2) start/end	2005H	
	32 bit Indexing	2000H	Section 8.1(6)
WDT (watchdog timer) setting	WDT setting	3000H	Section 6.16, Section 8.1(4)
	Initial execution monitoring time		Section 2.3.1, Section 8.1(4)
Error check	Carry out battery check	3001H	
	Carry out fuse blown check		
	Verify module		
	Check device range at indexing		
	Diagnose redundant power supply system		
Operating mode when there is an error	Computation error	3002H	Section 6.17, Section 8.1(4)
	Expanded command error		
	Fuse blown		
	Module verify error		
	Intelligent module program execution error		
	File access error		
	Memory card operation error		
	External power supply OFF		
Constant scanning		3003H	Section 6.2, Section 8.1(4)
Module error collection setting		300AH	Section 6.32, Section 8.1(5)
Detailed setting	H/W error time PLC operation mode	4004H	Section 6.9, Section 8.1(10)

(To the next page)

APPENDIX

Appendix 1 List of Parameter Numbers

TableApp.1 List of parameter numbers (continued)

Item		Parameter No.	Reference
Number of modules on MELSECNET/H		5000 _H	Section 8.2(3)
Valid module during other station access		5001 _H	
Interlink transmission parameters		5002 _H	
Routing parameters		5003 _H	
Starting I/O No.		5NM0 _H	
Network No.			
Total stations			
Mode		5NM0 _H	
Refresh parameters		5NM1 _H	
Common parameters		5NM2 _H	
Station inherent parameters		5NM3 _H	
Sub-master parameters		5NM5 _H	
Common parameters 2		5NMA _H	
Station inherent parameters 2		5NMB _H	
Interrupt settings			
Program		7000 _H	Section 2.3, Section 8.1(7)
Boot option	Clear program memory	7000 _H	Section 5.1.2, 5.1.8, Section 8.1(8)
	Auto Download all Data from Memory card to Standard ROM		
Boot file setting			
SFC program start mode		8002 _H	Section 8.1(9)
Start conditions		8003 _H	
Output mode when the block is stopped		8006 _H	
Number of modules on Ethernet		9000 _H	Section 8.2(4)
Starting I/O No.		9N00 _H	
Network No.			
Group No.			
Station No.			
Operational settings			
Initial settings		9N01 _H	
Open settings		9N02 _H	
Router relay parameter		9N03 _H	
Station No.<->IP information		9N05 _H	
FTP Parameters		9N06 _H	
E-mail settings		9N07 _H	
	News setting	9N08 _H	
Interrupt settings		9N09 _H	
Routing parameters		9N04 _H	

(To the next page)

TableApp.1 List of parameter numbers (continued)

Item	Parameter No.	Reference
Number of modules on CC-Link IE controller network	A000 _H	Section 8.2(1), (2), (3)
Interlink transmission parameters	A002 _H	
Routing parameters	A003 _H	
Network type	A080 _H	
Interlink transmission parameters	A082 _H	
Starting I/O No.	ANM0 _H	
Network No.		
Total stations		
Station No.		
Mode	ANM0 _H	
Refresh parameters	ANM1 _H	
Common parameters	ANM2 _H	
Network configuration settings		
Network operation settings		
Station inherent parameters	ANM3 _H	
Interrupt settings		
Number of modules	C000 _H	Section 8.2(5)
Remote input (RX)	CNM1 _H	
Remote output (RY)		
Remote register (RW _r)		
Remote register (RW _w)		
Ver.2 Remote input (RX)		
Ver.2 Remote output (RY)		
Ver.2 Remote register (RW _r)		
Ver.2 Remote register (RW _w)		
Special relay (SB)		
Special register (SW)		
Operational setting	CNM2 _H	
All connect count		
Retry count		
Automatic reconnection station count		
Standby master station No.		
PLC down select		
Scan mode setting		
Delay information setting		
Station information setting		
Remote device station initial setting		
Interrupt setting		

(To the next page)

APPEN-
DIX

Appendix 1 List of Parameter Numbers

TableApp.1 List of parameter numbers (continued)

Item		Parameter No.	Reference
Communication area setting (refresh setting)		E002 _H , E003 _H	Section 8.1 (12), QCPU User's Manual (Multiple CPU System)
Online module change		E006 _H	
Refresh parameter detailed device specification		E007 _H	
Multiple CPU high speed transmission area setting	CPU specific send range	E008 _H	
	Auto refresh setting	E009 _H	
Multiple CPU synchronous startup setting		E00B _H	
Host CPU number		E00C _H	

Appendix 2 Functions Added or Changed by Version Upgrade

The Universal model QCPU is upgraded when some functions are added or specifications are changed.
Therefore, the functions and specifications differ depending on the function version and serial number.

(1) Functions added and supported CPU module and GX Developer versions

TableApp.2 Functions added and supported CPU module and GX Developer versions

Function	Function version	First 5 digits of serial No.	Supported GX Developer
Use of the PC CPU module* ¹ (☞ QCPU User's Manual (Multiple CPU System))	B	"09072" or later	—
Setting of whether to use local device for each program (☞ Section 9.14.2)		"10012" or later	Version 8.62Q or later
Program batch transfer execution status (SM165) (☞ Section 5.1.3)			—
Multiple CPU high speed transmission dedicated instruction* ¹ (☞ MELSEC-Q/L Programming Manual (Common Instruction))			
Amount of battery consumption display (☞ QCPU User's Manual (Hardware Design, Maintenance and Inspection))			
Bit device extension (☞ Section 9.2)		"10042" or later	Version 8.68W or later
Executorial conditioned device test (☞ Section 6.11.4)			
Auto start sampling trace function* ¹ (☞ Section 6.14)			
CC-Link IE group cyclic transmission function (☞ CC-Link IE Controller Network Reference Manual)			
Scan time measurement (☞ Section 6.13.3)			
External input/output forced on/off (☞ Section 6.11.3)			
Monitor condition setting* ¹ (☞ Section 6.11.3)			
Redundant power supply system-compatible* ¹ (☞ Section 6.20)			
32-bit index modification using "ZZ" (☞ MELSEC-Q/L Programming Manual (Common Instruction))			
Extended data register (D) and extended link register (W)* ¹ * ² (☞ Section 9.8)		"09042" or later* ¹	Version 8.70Y or later
Serial Communication Function* ¹ (☞ Section 6.23)		"10102" or later	Version 8.78G or later
CPU module change function with memory card* ¹ (☞ Section 6.30)			
Local device setting of the index register* ¹ (☞ Section 9.14.2)			
Communication using the A-compatible 1C/1E frame (MC protocol)* ³ * ⁴ (☞ MELSEC-Q/L MELSEC Communication Protocol Reference Manual)		"11012" or later	Version 8.82L or later
Socket communication function (☞ QnUCPU User's Manual (Communication via Built-in Ethernet Port))* ¹			
Module model name read (☞ Section 6.31)		"11082" or later	—
IP address change function (☞ QnUCPU User's Manual (Communication via Built-in Ethernet Port))* ¹			—

— : Not related to GX Developer

APPENDIX

Appendix 2 Functions Added or Changed by Version Upgrade

*1: Some models do not support the function. For details, refer to the corresponding reference.

*2: Data of the extended data register (D) and extended link register (W) can be retained in the standard ROM by using the latch data backup to standard ROM function (☞ Section 6.28) if the serial number (first five digits) of the Universal model QCPU is "10042" or later.

*3: Communication using the A-compatible 1E frame is available only via any Ethernet module.

If the module is connected to the built-in Ethernet port of the CPU module, this function is not available.

*4: Communication using the A-compatible 1C frame is available only via any serial communication module.

If the module is connected to the built-in RS-232 interface of the CPU module, this function is not available.

(2) Functions added and supported CPU module and GX Works2 versions

TableApp.3 Functions added and supported CPU module and GX Works2 versions

Function	Function version	First 5 digits of serial No.	Supported GX Works2
Module error collection function *1 *3	B	"11043" or later	Version 1.09K or later
IP address change function (☞ QnUCPU User's Manual (Communication via Built-in Ethernet Port)) *1		"11082" or later	—
Local device batch read function (☞ Section 6.33) *1		"12012" or later	Version 1.31H or later
(☞ MELSEC-Q CC-Link IE Field Network Master/Local Module User's Manual)		*2	*2
Send points expansion function (CC-Link IE controller network module) (☞ Section 6.34) *3		"12052" or later	Version 1.34L or later
Online change of inactive blocks (SFC) (☞ MELSEC-Q/L/QnA Programming Manual (SFC)) *1			
Expansion of SFC step relay points (☞ Section 9.2)			
Operation mode setting at SFC double block START (☞ MELSEC-Q/L/QnA Programming Manual (SFC)) *1			
SFC comment readout instruction (☞ MELSEC-Q/L/QnA Programming Manual (SFC)) *1			
Data up to 10238 bytes can be exchanged with the SP.SOCSND/S(P).SOCRCV(S)/S(P).SOCRDATA instructions (☞ QnUCPU User's Manual (Communication via Built-in Ethernet Port)) *1			—

— : Not related to GX Works2

*1: Some models do not support the function. For details, refer to the corresponding reference.

*2: For each version, refer to the following.

☞ MELSEC-Q CC-Link IE Field Network Master/Local Module User's Manual

*3: For the versions of the intelligent function modules that support the function, refer to the manual for the intelligent function module used.

Appendix 3 Method of Replacing Basic Model QCPU or High Performance Model QCPU with Universal Model QCPU

Appendix 3.1 Replacement Precautions

This section describes precautions for replacing the Basic model QCPU or High Performance model QCPU with the Universal model QCPU and the replacement methods.

Appendix 3.1.1 Replacing Basic model QCPU with Universal model QCPU

(1) System configuration

TableApp.4 Precautions for replacement and replacement methods (System configuration)

Item	Precautions	Replacement method	Reference
GOT	GOT900 series cannot be connected.	Use GOT1000 series.	---
Applicable products and software	Products and software compatible with the Universal model QCPU must be used.	Products need to be replaced for the compatibility with the Universal model QCPU and software need to be upgraded for the communication with the Universal model QCPU are described in Appendix 3.2.	Appendix 3.2
Multiple CPU system	To configure a multiple CPU system, CPU modules compatible with the Universal model QCPU must be used.	CPU modules compatible with the Universal model QCPU are described in Appendix 3.2.	Appendix 3.2

(2) Program

TableApp.5 Precautions for replacement and replacement methods (Program)

Item	Precautions	Replacement method	Reference
Language and instruction	Some instructions are not supported.	Replace the instructions not supported in the Universal model QCPU are described in Appendix 3.3.	Appendix 3.3
Floating-point operation	When using the floating-point data comparison instructions, LDE[], ANDE[], ORE[], LDED[], ANDED[], and ORED[], if the comparison source data are -0, nonnumeric, unnormalized number, or $\pm \infty$, "OPERATION ERROR" (error code: 4101) is detected. ([] indicates one of the followings; =, <, >, <=, >=, <., >.)	When the floating-point data comparison instructions are used, modify the program as described in Appendix 3.4.2.	Section 2.4.4, Appendix 3.4.2
Device range check at index modification	When a device number exceeds a setting range due to index modification, "OPERATION ERROR" (error code: 4101) is detected.	Deselect the "Check device range at indexing" checkbox in the PLC RAS tab of the PLC parameter dialog box so that checking is not performed.	Section 6.17, Appendix 3.4.3

(To the next page)

TableApp.4 Precautions for replacement and replacement methods (Program) (Continued)

Item	Precautions	Replacement method	Reference
Latch setting	If latch ranges of internal user devices are specified, the processing time is added in proportion to the device points set to be latched.	The latch function of the Universal model QCPU is enhanced. (1) Large-capacity file register (R, ZR) (2) Writing/reading device data to the standard ROM (SP.DEVST and S(P).DEVLD instructions) (3) Latch range specification of internal devices Change the latch method to the method described in (1) to (3) above according to the application.	Section 6.3, Appendix 3.4.4
Interrupt counter	Interrupt counter is not supported.	Check the numbers of executions for interrupt programs on the Interrupt program monitor list screen of GX Developer.	Section 6.13.2, 9.2.11
SCJ instruction	When the SCJ instruction is used in the Universal model QCPU, the AND SM400 (or NOP instruction) needs to be inserted immediately before the SCJ instruction.	Insert the AND SM400 (or NOP instruction) immediately before the SCJ instruction when the SCJ instruction is used.	Section 6.5.1 in the MELSEC-Q/L Programming Manual (Common Instruction)
JP/GP.SREAD and JP/GP.SWRITE instructions	Use of the completion notification device of the target station specified in D3 when the SREAD and SWRITE instructions are used is available. (The Basic model QCPU ignores the device specified in D3.)	To get the same operation as the Basic model QCPU, omit D3 or use the READ instruction instead of the SREAD instruction.	Section 8.3.2 in the MELSEC-Q/L Programming Manual (Common Instruction)
ZPUSH instruction	The number of index registers is increased to 20 for the Universal model QCPU. The area for saving the data in the index register with the ZPUSH instruction is increased as well.	Increase the save area used for the ZPUSH instruction as needed.	Section 7.18.8 in the MELSEC-Q/L Programming Manual (Common Instruction)
Use of the annunciator (SET F□ and OUT F□ instructions)	When the annunciator is turned on by the SET F□ or OUT F□ instruction, the USER LED turns on. (The ERR.LED does not turn on.)	---	---
I/O refresh between programs	I/O refresh between programs cannot be executed.	Execute I/O refresh at the start or end of each program with the RFS or COM instruction. (When the COM instruction is used, I/O refresh to be executed can be specified in SD778 by turning on SM774.)	---
SM/SD	Usage of a part of the special relay and special register is different.	Replace the corresponding special relay and special register as described in Appendix 3.5.	Appendix 3.5
Multiple CPU system	The start address for the user setting area (auto refresh) in the CPU shared memory is changed.	If the user setting area in the CPU shared memory is specified in the program, change the address for the user setting area by performing an operation for replacing a device in GX Developer. (Example: "MOV D0 U3E0\G192" → "MOV D0U3E0\G2048")	---

(To the next page)

TableApp.4 Precautions for replacement and replacement methods (Program) (Continued)

Item	Precautions	Replacement method	Reference
File register	To use the file register, capacity setting is required.	Set the capacity of the file register used in the PLC file tab of the PLC parameter.	Section 8.1
SFC program	The following settings are required for using SFC programs. • Program setting (when both sequence programs and SFC programs exist.) • Common pointer No. setting (to execute the CALL instruction from SFC programs)	• Set program details in the Program tab of the PLC parameter dialog box. • Enter a common pointer number in the PLC system of the PLC parameter dialog box.	

(3) Drive and file

TableApp.6 Precautions for replacement and replacement methods (Drive and file)

Item	Precautions	Replacement method	Reference
Boot file setting	The boot file setting is not supported.	Since the Universal model QCPU holds the data in the program memory even when the battery voltage drops, the boot file setting is not necessary. Move files with the boot setting (from the standard ROM to the program memory) to the program memory.	Section 5.1.8

(4) External communication (Service processing)

TableApp.7 Precautions for replacement and replacement methods (External communication (Service processing))

Item	Precautions	Replacement method	Reference
Time reserved for communication processing	The time reserved for communication processing (SM315/SD315) is not supported.	Set service processing time in the PLC system tab of the PLC parameter dialog box.	Section 6.24.1
MC protocol	The following commands cannot specify monitoring conditions. • Randomly reading data in units of word (Command: 0403) • Device memory monitoring (Command: 0801) The applicable frame types are as follows: • QnA-compatible 3C/4C frame • QnA-compatible 3E frame • 4E frame	---	MELSEC-Q/L MELSEC Communication Protocol Reference Manual

(5) Battery installation position

TableApp.8 Precautions for replacement and replacement methods (Battery installation position)

Item	Precautions	Replacement method	Reference
Battery installation position	The battery replacement method is different. The battery installation position varies depending on the model. • Q00JCPU, Q00CPU, Q01CPU ...On the front of the module. • Q00UJCPU, Q00UCPU, Q01UCPU ...At the bottom of the module.	For the battery replacement method, refer to the in the Reference column.	Section 7.2 in the QCPU User's Manual (Hardware Design, Maintenance and Inspection)

(6) Program size

TableApp.9 Precautions for replacement and replacement methods (Program size)

Item	Precautions	Replacement method	Reference
Program size	Data in the program memory of the Basic model QCPU may exceed the size of the program memory of the Universal model QCPU.	Store parameter and device comment files in the standard ROM.	---

Appendix 3.1.2 Replacing High Performance model QCPU with Universal model QCPU

(1) System configuration

TableApp.10 Precautions for replacement and replacement methods (System configuration)

Item	Precautions	Replacement method	Reference
Use of AnS/A series module	AnS/A series modules are not supported.	Use Q series modules.	---
GOT	GOT900 series cannot be connected.	Use GOT1000 series.	---
Programming tool connection	Applicable USB cables are different. • High Performance model QCPU ... A-B type • Universal model QCPU ... A-miniB type	Use USB cables of A-miniB type. Or, use USB conversion adapters of B-miniB type.	---
Applicable products and software	Products and softwares compatible with the Universal model QCPU must be used.	Products need to be replaced for the compatibility with the Universal model QCPU and software need to be upgraded for the communication with the Universal model QCPU are described in Appendix 3.2.	Appendix 3.2
Multiple CPU system	To configure a multiple CPU system, CPU modules compatible with the Universal model QCPU must be used.	CPU modules compatible with the Universal model QCPU are described in Appendix 3.2.	Appendix 3.2
	In a multiple CPU system using the Motion CPU, an existing auto refresh area and user setting area cannot be used for data communication with the Motion CPU.	For data communication with the Motion CPU, use an auto refresh area and user setting area in the multiple CPU high-speed transmission area.	Chapter 4 in the QCPU User's Manual (Multiple CPU System)
Redundant power supply system ^{*1}	In a redundant power supply system using the Q38RB redundant power supply main base unit and the Q68RB redundant power supply extension base unit, the status of the power supply module cannot be stored in the special relay and special register (SM1780 to SM1783/SD1780 to SD1783). The status cannot be displayed on the system monitor screen.	Check the status of the power supply module by the LED on the front of the module.	Section 5.3 in the QCPU User's Manual (Hardware Design, Maintenance and Inspection)
MELSECNET/H	The simple dual-structured network function is not supported.	---	Section 7.7 in the Q Corresponding MELSECNET/H Network System Reference Manual (PLC to PLC network)

^{*1}: The serial number (first five digits) of the applicable module (Q02UCPU, Q03UDCPU, Q04UDHCPU, Q06UDHCPU, Q13UDHCPU, and Q26UDHCPU) must be "10041" or earlier.

APPENDIX

Appendix 3 Method of Replacing Basic Model QCPU or High Performance Model QCPU with Universal Model QCPU
Appendix 3.1 Replacement Precautions

(2) Program

TableApp.11 Precautions for replacement and replacement methods (Program)

Item	Precautions	Replacement method	Reference
Language and instruction	Some instructions are not supported.	Replace the instructions not supported in the Universal model QCPU are described in Appendix 3.3.	Appendix 3.3
Floating-point operation	The Universal model QCPU performs program operations of floating-point data in single precision.	Instructions for floating-point double-precision operations are added for the Universal model QCPU. Replace the instructions if floating-point double-precision operations are required, as described in Appendix 3.4.	Section 2.4.4, Appendix 3.4, Appendix 3.4.2
	When using the floating-point data comparison instructions, LDE[], ANDE[], ORE[], LDED[], ANDED[], and ORED[], if the comparison source data are -0, nonnumeric, unnormalized number, or $\pm \infty$, "OPERATION ERROR" (error code: 4101) is detected. ([] indicates one of the followings; =, <, <=, >=, <, >.)	When the floating-point data comparison instructions are used, modify the program as described in Appendix 3.4.2.	
Device range check at index modification	When a device number exceeds a setting range due to index modification, "OPERATION ERROR" (error code: 4101) is detected.	Deselect the "Check device range at indexing" checkbox in the PLC RAS tab of the PLC parameter dialog box so that checking is not performed.	Section 6.17, Appendix 3.4.3
Program execution type	Low-speed execution type programs are not supported.	Use scan execution type programs or fixed scan execution type programs.	Section 2.3
	A program execution type cannot be changed by remote operation.	Use instructions for switching program execution types, such as PSTOP, POFF, and PSCAN.	Section 2.3
Latch setting	If latch ranges of internal user devices are specified, the processing time is added in proportion to the device points set to be latched. (For example, if 8K points are latched for the latch relay (L), the processing time of 28.6 μ s is required.)	The latch function of the Universal model QCPU is enhanced. (1) Large-capacity file register (R, ZR) (2) Writing/reading device data to the standard ROM (SP.DEVST and S(P).DEVLD instructions) (3) Latch range specification of internal devices Change the latch method to the method described in (1) to (3) above according to the application.	Section 6.3, Appendix 3.4.4

(To the next page)

TableApp.9 Precautions for replacement and replacement methods (Program) (Continued)

Item	Precautions	Replacement method	Reference
Interrupt program	The interrupt pointer (I49) for the high-speed interrupt function is not supported.	Consider the use of interrupt pointers for fixed scan interrupt (I28 to I31).	Section 6.13.2, 9.2.11
	Interrupt counter is not supported.	Check the numbers of executions for interrupt programs on the Interrupt program monitor list screen of GX Developer.	
	The interrupt pointers (I32 to I40) due to an error are not supported.	---	Section 9.11
SCJ instruction	When the SCJ instruction is used in the Universal model QCPU, the AND SM400 (or NOP instruction) needs to be inserted immediately before the SCJ instruction.	Insert the AND SM400 (or NOP instruction) immediately before the SCJ instruction when the SCJ instruction is used.	Section 6.5.1 in the MELSEC-Q/L Programming Manual (Common Instruction)
ZPUSH instruction	The number of index registers is increased to 20 for the Universal model QCPU. The area for saving the data in the index register with the ZPUSH instruction is increased as well.	Increase the save areas used for the ZPUSH instruction as needed.	Section 7.18.8 in the MELSEC-Q/L Programming Manual (Common Instruction)
File usability setting for each program	The following file usability setting for each program is not available.* ¹ • File register • Initial device value • Comment	When file usability is set, modify the program as described in Appendix 3.4.	Section 2.3, Appendix 3.4.5
I/O refresh setting for each program	I/O refresh setting for each program is not available.	Use the RFS instruction if I/O refresh setting for each program is required.	• Section 2.3 • MELSEC-Q/L Programming Manual (Common Instruction)
SM/SD	Usage of a part of the special relay and special register is different.	Replace the corresponding special relay and special register as described in Appendix 3.5.	Appendix 3.5
	A series-compatible special relay and special register are not supported. (SM1000 to SM1255/SD1000 to SD1255)	Using GX Developer, A series-compatible special relay and special register can be replaced with the Universal model QCPU-compatible special relay and special register. Note, however, that the ones which are not compatible with the Universal model QCPU are replaced with SM1255 and SD1255. Modify programs as needed.	Chapter 12
Processing time	Scan time and each processing time are different.	Modify programs as needed, checking the processing timing.	---

* 1: Even the local device file usability setting is not available for the Q02UCPU, Q03UDCPU, Q04UDHCPU, and Q06UDHCPU if the serial number (first five digits) is "10011" or earlier.

APPENDIX

Appendix 3 Method of Replacing Basic Model QCPU or High Performance Model QCPU with Universal Model QCPU
Appendix 3.1 Replacement Precautions

(3) Drive and file

TableApp.12 Precautions for replacement and replacement methods (Drive and file)

Item	Precautions	Replacement method	Reference
Boot file setting	Files in the standard ROM cannot be booted to the program memory.	Since the Universal model QCPU holds the data in the program memory even when the battery voltage drops, the boot file setting is not necessary. Move files with the boot setting (from the standard ROM to the program memory) to the program memory.	Section 5.1.8, 5.1.10, Appendix 3.4.6
	Booting operation is different.	Replacement method when the parameter-valid drive and the boot file setting are set in the High Performance model QCPU is described in Appendix 3.4.6.	
Automatic all data write from memory card to standard ROM	The setting method of this function is different.	In the Boot file tab of the PLC parameter dialog box, select "standard ROM" for the transfer destination. Note, however, that the transfer destination of "program" is fixed to "program memory". (Setting by DIP switches is not necessary.)	Section 5.1.8

(4) External communication

TableApp.13 Precautions for replacement and replacement methods (External communication)

Item	Precautions	Replacement method	Reference
Module service interval time read	The module service interval time cannot be read.	---	Section 6.24.1
MC protocol	The following frame types cannot be used when accessing the Universal model QCPU.*1 • A-compatible 1C frame • A-compatible 1E frame	Use the frame types below. • QnA-compatible 2C/3C/4C frame • QnA-compatible 3E frame • 4E frame	MELSEC-Q/L MELSEC Communication Protocol Reference Manual
	The following commands cannot specify monitoring conditions. • Randomly reading data in units of word (Command: 0403) • Device memory monitoring (Command: 0801) The applicable frame types are as follows: • QnA-compatible 3C/4C frame • QnA-compatible 3E frame • 4E frame	---	

*1: The serial number (first five digits) of the applicable module (Q02UCPU, Q03UDCPU, Q04UDHCPU, Q06UDHCPU, Q13UDHCPU, and Q26UDHCPU) must be "10101" or earlier.

(5) Diagnostic function

TableApp.14 Precautions for replacement and replacement methods (Diagnostic function)

Item	Precautions	Replacement method	Reference
Error history	Error history data cannot be stored in the memory card.	The Universal model QCPU can store history data by the number of storable history data in a memory card (100) to the system memory.	Section 6.18
LED indication priority setting	LED indication priority cannot be set. Only LED indication setting at error occurrence is supported.	---	Section 6.21.2

(6) Debugging

TableApp.15 Precautions for replacement and replacement methods (Debugging)

Item	Precautions	Replacement method	Reference
Monitor ^{*3}	The monitoring condition cannot be set.	Use the sampling trace function for checking device data under the specified monitoring condition. With this function, changes of the specified device data can be recorded at the following timings: • at execution of the specified step • on the rising/falling edge of bit devices • when the value of word devices coincide with the setting value • at every specified time (settable range: 1 to 5000ms)	Section 6.11.1, 6.14
Scan time measurement by GX Developer ^{*3}	Time required for executing a part of the program cannot be measured using the scan time measurement function. ^{*1}	Calculate the time using instruction processing time described in the manual.	• Section 6.13.3 • Appendix 1.4 in the MELSEC-Q/L Programming Manual (Common Instruction)
External input/output forced on/off ^{*3}	The external input/output forced on/off function is not supported. ^{*2}	The function can be replaced with the programs described in Appendix 3.4.7. Note, however, that replacement method described does not apply in the following cases: • Input and output targeted for forced on/off are referred to or changed using the direct input device (DX) and direct output device (DY). • Input and output targeted for forced on/off are referred to or changed within an interrupt program.	Section 6.11.3, Appendix 3.4.7

* 1: Scan time of each program can be checked on the Program monitor list screen.

* 2: Device test by GX Developer can be performed.

* 3: The serial number (first five digits) of the applicable module (Q02UCPU, Q03UDCPU, Q04UDHCPU, Q06UDHCPU, Q13UDHCPU, and Q26UDHCPU) must be "10041" or earlier.

(7) Switch on the front of the CPU module

TableApp.16 Precautions for replacement and replacement methods (Switch on the front of the CPU module)

Item	Precautions	Replacement method	Reference
Switch on the front of the CPU module	The operation method with the RESET/RUN/STOP switch is modified.	The RESET/STOP/RUN switch of the Universal model QCPU can be used for the reset operation of the CPU module and switching between the STOP and RUN status.	Section 4.4 in the QCPU User's Manual (Hardware Design, Maintenance and Inspection)
	Latch data cannot be cleared by the switch.	To clear latch data, use the remote latch clear operation of GX Developer.	Section 4.4 in the QCPU User's Manual (Hardware Design, Maintenance and Inspection)
	The system protect cannot be set by the switch.	Data in the files can be protected by setting a password for each file. Password for each file can be registered with GX Developer.	Section 6.19
	The parameter-valid drive setting is not necessary.	The Universal model QCPU automatically determines the parameter-valid drive. Change the setting as described in Appendix 3.4.5 when the parameter-valid drive is set to other than the program memory in the High Performance model QCPU.	• Appendix 3.4.6 • Section 4.4 in the QCPU User's Manual (Hardware Design, Maintenance and Inspection)

(8) SFC

TableApp.17 Precautions for replacement and replacement methods (SFC)

Item	Precautions	Replacement method	Reference
Step transition monitoring timer	The step transition monitoring timer is not supported.	Change the program as described in Appendix 3.1 in the manual in the Reference column.	Section 4.6 and Appendix 3.1 in the MELSEC-Q/L/QnA Programming Manual (SFC)
SFC operation mode setting	The periodic execution block setting is not supported.	Change the program as described in Appendix 3.2 in the manual in the Reference column.	Section 4.7.4 and Appendix 3.2 in the MELSEC-Q/L/QnA Programming Manual (SFC)
	An operation mode at double block START cannot be selected. (Fixed to "WAIT".)* ¹	---	Section 4.7.5 in the MELSEC-Q/L/QnA Programming Manual (SFC)
	An operation mode at transition to active step cannot be selected. (Fixed to "TRANSFER".)	---	Section 4.7.6 in the MELSEC-Q/L/QnA Programming Manual (SFC)
SFC program for program execution management	SFC programs for program execution management are not supported.	---	Section 5.2.3 in the MELSEC-Q/L/QnA Programming Manual (SFC)
SFC control instruction	Some SFC control instructions are not supported.	SFC control instructions not supported in the Universal model QCPU and replacing methods are described in Appendix 3.3.	• Appendix 3.3 • Section 4.4 in the MELSEC-Q/L/QnA Programming Manual (SFC)
SFC comment readout instruction	The following SFC comment readout instructions are not supported. • S(P).SFCSCOMR (SFC step comment readout instruction) • S(P).SFCTCOMR (SFC transition condition comment readout instruction)	---	Section 4.8 in the MELSEC-Q/L/QnA Programming Manual (SFC)
Method of SFC program change	SFC program files cannot be written to the running CPU module. (Programs in SFC Figure can be changed online.)	• Write program data to the CPU module after changing the Universal model QCPU status to STOP. • An inactive block in an SFC program can be changed by online change of inactive block.*²	Section 6.6 in the MELSEC-Q/L/QnA Programming Manual (SFC)

*1: This applies to the Q00UJCPU, Q00UCPU, Q01UCPU, and Q02UCPU and the Universal model QCPU whose serial number (first five digits) is "12051" or earlier.

*2: This operation is available for the Universal model QCPU other than the Q00UJCPU, Q00UCPU, Q01UCPU, and Q02UCPU and whose serial number (first five digits) is "12052" or later.

APPENDIX

Appendix 3 Method of Replacing Basic Model QCPU or High Performance Model QCPU with Universal Model QCPU
Appendix 3.1 Replacement Precautions

Appendix 3.2 Applicable devices and software

(1) Products need to be replaced for the compatibility with the Universal model QCPU

The following tables show products need to be replaced for the compatibility with the Universal model QCPU. (As for products not listed in the tables below, replacement is not required.)

TableApp.18 Product need to be replaced (Communication module)

Product	Model	Serial number (first five digits) of the product compatible with the Universal model QCPU*2	
		Used with Q02U/Q03UD/Q04UDH/Q06UDHCPU	Used with Q00UJ/Q00U/Q01UCPU/Q10UDH/Q13UDH/Q20UDH/Q26UDHCPU, or Built-in Ethernet port QCPU
Web server module*1	• QJ71WS96	"09042" or later.	"10012" or later.
MES interface module	• QJ71MES96	"09042" or later.	"10012" or later.

*1: The Universal model QCPU does not operate normally when the Web server module on which GX RemoteService-I is installed is used.

*2: The Universal model QCPU does not operate normally when a product not compatible with the Universal model QCPU is used.

TableApp.19 Product need to be replaced (Personal computer board)

Product	Model	Dedicated software package version compatible with the Universal model QCPU*1		
		Used with Q02U/Q03UD/Q04UDH/Q06UDHCPU	Used with Q13UDH/Q26UDHCPU	Used with Q00UJ/Q00U/Q01U/Q10UDH/Q20UDHCPU, or Built-in Ethernet port QCPU
CC-Link IE controller network interface board	• Q80BD-J71GP21-SX • Q80BD-J71GP21S-SX	No restrictions	Version 1.03D or later	Version 1.06G or later
MELSECNET/H interface board	SI/QSI/H-PCF optical cable	• Q80BD-J71LP21-25 • Q80BD-J71LP21S-25	Version 15R or later	Version 18U or later
		• Q81BD-J71LP21-25	Version 19V or later	Version 19V or later
	GI optical cable	• Q80BD-J71LP21G	Version 15R or later	Version 18U or later
	Coaxial cable	• Q80BD-J71BR11		
CC-Link system master/local interface board	• Q80BD-J61BT11N	Version 1.02C or later	Version 1.05F or later	Version 1.07H or later
	• Q81BD-J61BT11	Version 1.06G or later	Version 1.06G or later	

*1: No restrictions on the board itself.

TableApp.20 Product need to be replaced (GOT)

Product	Model	OS version provided with GT Designer2 compatible with the Universal model QCPU*1				
		Used with Q00UJ/Q00U/Q01U/Q10UDH/Q20UDHCPU	Used with Q02U/Q03UD/Q04UDH/Q06UDHCPU	Used with Q13UDH/Q26UDHCPU	Used with Q03UDE/Q04UDEH/Q06UDEH/Q13UDEH/Q26UDEHCPU	Used with Q10UDEH/Q20UDEHCPU
GOT1000	GT15□□-□□	Version 2.91V or later	Version 2.60N or later	Version 2.76E or later	Version 2.81K or later	Version 2.91V or later
	GT11□□-□□					
	GT10□□-□□	Version 2.91V or later	Version 2.76E or later	Version 2.76E or later	---	---

*1: No restrictions on GOT itself.

TableApp.21 Product need to be replaced (Network module and serial communication module)

Product	Model	Module version compatible with the Universal model QCPU	
		Used with Q00UJ/Q00U/Q01U/Q02U/ Q03UD/Q04UDH/Q06UDH/Q10UDH/ Q13UDH/Q20UDH/Q26UDHCPU	Used with Built-in Ethernet port QCPU
MELSECNET/H model	• QJ71LP21-25 • QJ71LP21S-25 • QJ71LP21G • QJ71BR11	No restrictions	Some restrictions depending on use conditions ^{*1}
Serial communication module	• QJ71C24N • QJ71C24N-R2 • QJ71C24N-R4		Serial number (first five digits) "10042" or later
Modem interface module	• QJ71CMON		

*1: The serial number (first five digits) of the MELSECNET/H module must be "10042" or later if all conditions 1) to 4) described below are satisfied.

- 1) A multiple CPU system including Built-in Ethernet port QCPU is configured.
- 2) GX Developer or GOT is connected to an Ethernet port of the Built-in Ethernet port QCPU.
- 3) GX Developer or GOT accesses the CPU module on another station via the MELSECNET/H module controlled by another CPU.
- 4) The access target on another station is A/QnA series CPU module.

APPENDIX

(2) CPU modules that can configure a multiple CPU system with the Universal model QCPU

CPU modules that can configure a multiple CPU system with the Universal model QCPU are shown below.

(a) For the QnUD(H)CPU or Built-in Ethernet port QCPU

TableApp.22 CPU module that can configure a multiple CPU system with the QnUD(H)CPU or Built-in Ethernet port QCPU

CPU module	Model	Applicable version			Restrictions
		Configured with Q03UD/Q04UDH/Q06UDHCPU	Configured with Q13UDH/Q26UDH, Q03UDE/Q04UDEH, Q06UDEH/Q13UDEH/Q26UDEHCPU	Configured with Q10UDH/Q20UDH/Q10UDEH/Q20UDEHCPU	
Motion CPU	- Q172DCPU - Q173DCPU	No restrictions			Use only the multiple CPU high-speed main base unit (Q3□ DB) as a main base unit.
PC CPU module	PPC-CPU852(MS)	Driver S/W (PPC-DRV-02) version 1.01 or later	Driver S/W (PPC-DRV-02) version 1.02 or later	Driver S/W (PPC-DRV-02) version 1.03 or later	---
C Controller module	- Q06CCPU-V - Q06CCPU-V-B	No restrictions	Serial number (first five digits) "10012" or later	Serial number (first five digits) "10102" or later.	---
	Q12DCCPU-V	No restrictions			---
High Performance model QCPU	- Q02CPU - Q02HCPU - Q06HCPU - Q12HCPU - Q25HCPU	Function version B or later			---
Process CPU	- Q02PHCPU - Q06PHCPU - Q12PHCPU - Q25PHCPU	No restrictions			---

(b) For the Q00UCPU, Q01UCPU, or Q02UCPU

TableApp.23 CPU module that can configure a multiple CPU system with Q00UCPU, Q01UCPU, or Q02UCPU

CPU module	Model	Applicable version		Restrictions
		Configured with Q00U/Q01UCPU	Configured with Q02UCPU	
Motion CPU	- Q172CPUN(-T) - Q173CPUN(-T) - Q172HCPU(-T) - Q173HCPU(-T)	No restrictions		The multiple CPU high-speed main base unit (Q3□ DB) cannot be used as a main base unit.
PC CPU module	PPC-CPU852(MS)	Driver S/W (PPC-DRV-02) version 1.03 or later	Driver S/W (PPC-DRV-02) version 1.01 or later	---
C Controller module	- Q06CCPU-V - Q06CCPU-V-B	Serial number (first five digits) "10102" or later	No restrictions	---
	Q12DCCPU-V	No restrictions		---

(3) Software need to be upgraded for the compatibility with the Universal model QCPU

The following table shows software need to be upgraded for the communication with the Universal model QCPU.

(As for software not listed in the table below, version upgrade is not required.)

The latest version can be downloaded from the MELFANSweb.

TableApp.24 Software need to be upgraded for the compatibility with the Universal model QCPU

Product	Model	Version compatible with the Universal model QCPU			
		Used with Q02U/ Q03UD/Q04UDH/ Q06UDHCPU	Used with Q13UDH/ Q26UDHCPU	Used with Q03UDE/ Q04UDEH/Q06UDEH/ Q13UDEH/ Q26UDEHCPU	Used with Q00UJ/Q00U/ Q01U/Q10UDH/ Q20UDH/Q10UDEH/ Q20UDEHCPU
GX Developer	SW8D5C-GPPW-E	Version 8.48A or later	Version 8.62Q or later	Version 8.68W or later	Version 8.78G or later
GX Configurator-AD	SW2D5C-QADU-E	Version 2.05F or later ^{*1}	Version 2.05F or later ^{*2}	Version 2.05F or later ^{*3}	Version 2.05F or later ^{*4}
GX Configurator-DA	SW2D5C-QDAU-E	Version 2.06G or later ^{*1}	Version 2.06G or later ^{*2}	Version 2.06G or later ^{*3}	Version 2.06G or later ^{*4}
GX Configurator-SC	SW2D5C-QSCU-E	Version 2.12N or later ^{*1}	Version 2.12N or later ^{*2}	Version 2.17T or later ^{*3}	Version 2.17T or later ^{*4}
GX Configurator-CT	SW0D5C-QCTU-E	Version 1.25AB or later ^{*1}	Version 1.25AB or later ^{*2}	Version 1.25AB or later ^{*3}	Version 1.25AB or later ^{*4}
GX Configurator-TI	SW1D5C-QTIU-E	Version 1.24AA or later ^{*1}	Version 1.24AA or later ^{*2}	Version 1.24AA or later ^{*3}	Version 1.24AA or later ^{*4}
GX Configurator-TC	SW0D5C-QTCU-E	Version 1.23Z or later ^{*1}	Version 1.23Z or later ^{*2}	Version 1.23Z or later ^{*3}	Version 1.23Z or later ^{*4}
GX Configurator-FL	SW0D5C-QFLU-E	Version 1.23Z or later ^{*1}	Version 1.23Z or later ^{*2}	Version 1.23Z or later ^{*3}	Version 1.23Z or later ^{*4}
GX Configurator-QP	SW2D5C-QD75P-E	Version 2.25B or later	Version 2.29F or later	Version 2.30G or later ^{*5}	Version 2.32J or later
GX Configurator-PT	SW1D5C-QPTU-E	Version 1.23Z or later ^{*1}	Version 1.23Z or later ^{*2}	Version 1.23Z or later ^{*3}	Version 1.23Z or later ^{*4}
GX Configurator-AS	SW1D5C-QASU-E	Version 1.21X or later ^{*1}	Version 1.21X or later ^{*2}	Version 1.21X or later ^{*3}	Version 1.21X or later ^{*4}
GX Configurator-MB	SW1D5C-QMBU-E	Version 1.08J or later ^{*1}	Version 1.08J or later ^{*2}	Version 1.08J or later ^{*3}	Version 1.08J or later ^{*4}
GX Configurator-DN	SW1D5C-QDNU-E	Version 1.23Z or later ^{*1}	Version 1.23Z or later ^{*2}	Version 1.24AA or later ^{*3}	Version 1.24AA or later ^{*4}
MX Component	SW3D5C-ACT-E	Version 3.09K or later	Version 3.10L or later	Version 3.11M or later	Version 3.12N or later
GX Simulator	SW7D5C-LLT-J	Version 7.23Z or later ^{*4}	Version 7.23Z or later ^{*4}	Version 7.23Z or later ^{*4}	Version 7.23Z or later ^{*4}

*1: The software can be used by installing GX Developer Version 8.48A or later.

*2: The software can be used by installing GX Developer Version 8.62Q or later.

*3: The software can be used by installing GX Developer Version 8.68W or later.

*4: The software can be used by installing GX Developer Version 8.78G or later.

*5: GX Configurator-QP Version 2.29F can be used when connected via USB.

(4) Software not supported in the Universal model QCPU

The following table shows software not supported in the Universal model QCPU.

TableApp.25 Software not supported in the Universal model QCPU

Product	Model
GX Explorer	SW $\square$ D5C-EXP-E
GX Converter	SW $\square$ D5C-CNVW-E

Appendix 3.3 Instructions

Appendix 3.3.1 Instructions not supported in the Universal model QCPU and replacing methods

The Universal model QCPU does not support instructions listed in the TableApp.25. and TableApp.26. Instructions need to be replaced using replacing methods described in the tables. (If no instruction in the list is used, replacement is not required.)

TableApp.26 Instructions not supported in the Universal model QCPU and replacing methods

Symbol	Instruction	Replacing method	Reference	
IX	Index modification of entire ladder	Instructions can be replaced using a replacement program.	Appendix 3.3.3 (1)	
IXEND				
IXDEV	Modification value specification in index modification of entire ladder	Change the program so that the device offset values specified by the IXSET instruction are directly set to the index modification table using the MOV instruction.	Appendix 3.3.3 (2)	
IXSET				
PR	Print ASCII code instruction	- It is recommended to use GOT as an ASCII code display device. ASCII codes stored in devices are directly displayed as characters on GOT. - Instructions can be replaced using a replacement program.	Appendix 3.3.3 (3)	
PRC	Print comment instruction	- It is recommended to use GOT as an ASCII code display device.Device comments can be displayed on GOT. - Comment data can be output to a display device in the replacement program of the PR instruction after reading data using the reading device comment data instruction (COMRD(P)).		
CHKST	Specific format failure check instruction	Instructions can be replaced using a replacement program.	Appendix 3.3.3 (4)	
CHK				
CHKCIR	Format change instruction for CHK instruction	Failure detection ladder patterns can be changed in a replacement program.		
CHKEND				
PLOW	Program low-speed execution registration instruction	- Use the PSCAN instruction instead of this instruction when low-speed execution type programs are replaced with scan execution type programs. - No instruction can be used if low-speed execution type programs are replaced with fixed scan execution type programs.	---	
PCHK	Program execution status check instruction	Check a program execution status on the Program monitor list screen of GX Developer. For details, refer to Section 6.13.1 in this manual.	---	
KEY	Numerical key input instruction	- It is recommended to use GOT as a numeral input device. - Instructions can be replaced using a replacement program.	Appendix 3.3.3 (5)	
PLOADP	Load program from memory card	Store all programs to be executed in the program memory. The Universal model QCPU can neither add programs to the program memory nor change them with other programs during RUN. If the capacity of the program memory is not enough, store parameters, device comments, and initial device values in the program memory into the standard ROM or a memory card instead.	---	
PUNLOADP	Unload program from memory card			
PSWAPP	Load + Unload			

TableApp.27 SFC control instructions not supported in the Universal model QCPU and replacing methods

Symbol	Instruction	Replacing method
LD TRn	Forced transition check instruction	When the programmable controller type is changed, these instructions are converted into SM1255. Modify programs as needed.
AND TRn		
OR TRn		
LDI TRn		
ANDI TRn		
ORI TRn		
LD BLm\TRn		
AND BLm\TRn		
OR BLm\TRn		
LDI BLm\TRn		
ANDI BLm\TRn		
ORI BLm\TRn		
SCHG(D)	Active step change instruction	Refer to Appendix 3 "Restrictions on Basic Model QCPU, Universal Model QCPU, and LCPU and Alternative Methods" in the MELSEC-Q/L/QnA Programming Manual (SFC).
SET TRn	Transition control instruction	Refer to Appendix 3 "Restrictions on Basic Model QCPU, Universal Model QCPU, and LCPU and Alternative Methods" in the MELSEC-Q/L/QnA Programming Manual (SFC).
SET BLm\TRn		
RST TRn		
RST BLm\TRn		
BRSET(S)	Block switching instruction	When the programmable controller type is changed, these instructions are converted into SM1255. Modify programs as needed.

APPENDIX

Appendix 3.3.2 Replacing programs using multiple CPU transmission dedicated instructions

(1) Replacing the module with the QnUD(H)CPU or Built-in Ethernet port QCPU

TableApp.27 shows instructions need to be replaced and corresponding alternative instructions. For the specifications of each instruction, refer to the manuals for the Motion CPU.

TableApp.28 SFC instructions not supported in the QnUD(H)CPU and Built-in Ethernet port QCPU and thier alternatives

Symbol	Instruction description	Symbol of alternative instruction
S(P).DDWR	Write other CPU device data into host CPU	D(P).DDWR
S(P).DDRDR	Read other CPU device data into host CPU	D(P).DDRDR
S(P).SFCS	Request of motion SFC program startup	D(P).SFCS
S(P).SVST	Request of servo program startup	D(P).SVST
S(P).CHGA	Current value change of halted axis/synchronized encoder/cam axis	D(P).CHGA
S(P).CHGV	Axis speed change during positioning and JOG operation	D(P).CHGV
S(P).CHGT	Torque control value change during operation and suspension in real mode	D(P).CHGT
S(P).GINT	Request of other CPU interrupt program startup	D(P).GINT

(2) Replacing the module with the Q00UCPU, Q01UCPU, or Q02UCPU

The Q00UCPU, Q01UCPU, and Q02UCPU support the same multiple CPU transmission dedicated instructions used in the Basic model QCPU.

The alternative instructions in TableApp.27 are not available for these CPU modules.

Appendix 3.3.3 Program replacement examples

This section shows program replacement examples for the instructions of which replacement programs are available in Appendix 3.3. (Skip this section if instructions listed in Appendix 3.3.1 are not used.)

(1) Replacement example of the IX and IXEND instructions

Since index registers are saved using the ZPUSH instruction, a 23-word index register save area is required.

(a) Example of device assignment

TableApp.29 Example of device assignment

Before replacement		After replacement	
Application	Device	Application	Device
Index modification table	D100 to D115	Index modification table	D100 to D115
		Index register save area	D200 to D222

APPEN-
DIX

If the device numbers in the example above are used for other applications, assign unused device numbers instead.

(b) Program before replacement

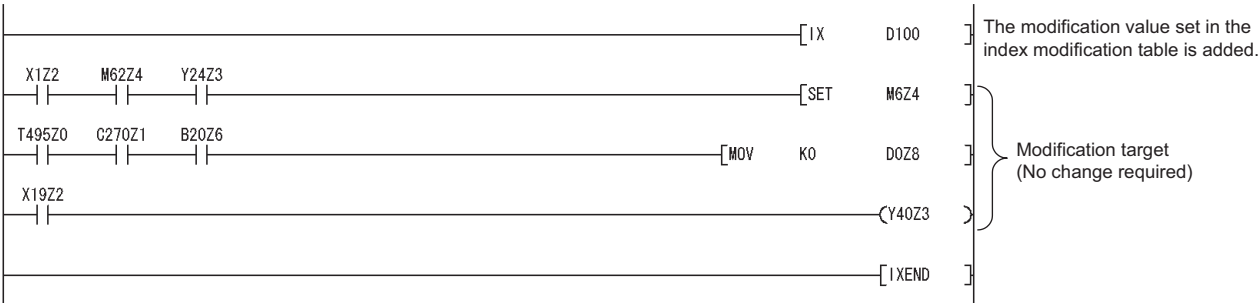


Figure App.1 Sample program

Appendix 3 Method of Replacing Basic Model QCPU or High Performance Model QCPU with Universal Model QCPU
Appendix 3.3 Instructions

(c) Program after replacement

- Replace the IX instruction with the ZPUSH instruction and the processing for setting the contents of index modification table to index registers.
- Replace the IXEND instruction with the ZPOP instruction.

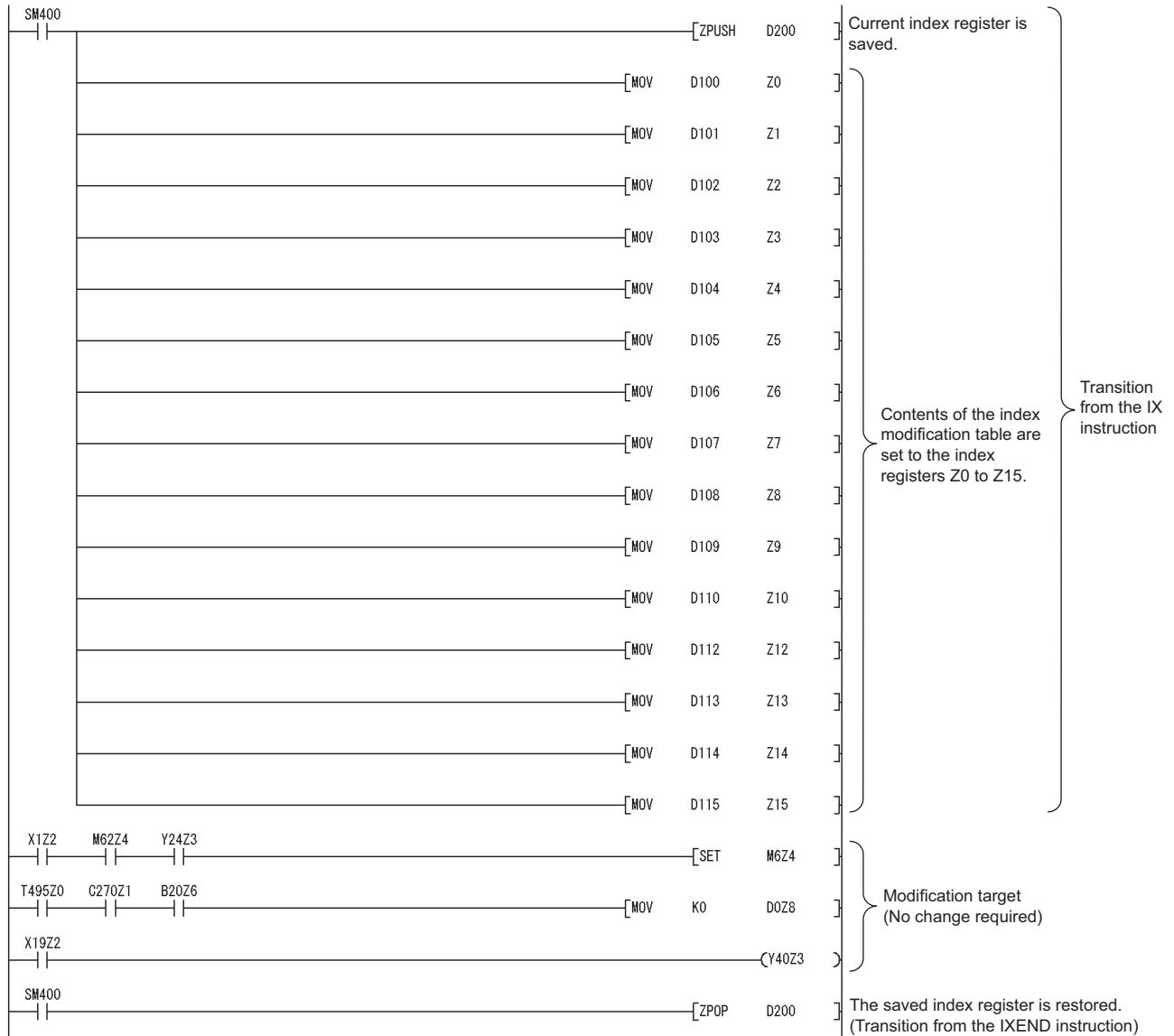


Figure App.2 Sample program

(2) Replacement example of the IXDEV and IXSET instructions

Change the program so that the device offset value specified by the contacts between the IXDEV and the IXSET instructions are directly set to the index modification table using the MOV instruction.

For the devices whose device offset value is not specified by the IXDEV and IXSET instructions, set the device offset value to 0 in the program after replacement.

Figure App.3 shows how the device offset value is set in the program before and after replacement.

Device offset specification		Index modification table	
Timer	T□ ├─┤	→	(D)+0
Counter	C□ ├─┤	→	(D)+1
Input *1	X□ ├─┤	→	(D)+2
Output *1	Y□ ├─┤	→	(D)+3
Internal relay	M□ ├─┤	→	(D)+4
Latch relay	L□ ├─┤	→	(D)+5
Edge relay	V□ ├─┤	→	(D)+6
Link relay *1	B□ ├─┤	→	(D)+7
Data register	D□.XX ├─┤	→	(D)+8
Link register *1	W□.XX ├─┤	→	(D)+9
File register	R□.XX ├─┤	→	(D)+10
Intelligent function module device *2	U□\G□.XX ├─┤	Start I/O number →	(D)+11
		Buffer memory address →	(D)+12
Link direct device *3	J□\B□ ├─┤	→	(D)+13
File register (through number)	ZR□.XX ├─┤	→	(D)+14
Pointer	— IXSET P□ —	→	(D)+15

Figure App.3 Device offset value specification and index modification table

*1: Device numbers are represented in hexadecimal. Use hexadecimal constants (H□□) when setting values in the index modification table.

*2: Start I/O numbers (U□□) are represented in hexadecimal. Use hexadecimal constants (H□□) when setting values in the index modification table.

*3: Devices B, W, X, or Y can be specified following J\□□. Set device numbers for B, W, X, and Y as device offset values of each device in the index modification table.

For example, if "J10\Y220" is specified by the IXDEV and IXSET instructions, set "K10" in (D)+13 and "H220" in (D)+3 in the replacement program. ((D) indicates the start device in the index modification table.)

(a) Program before replacement

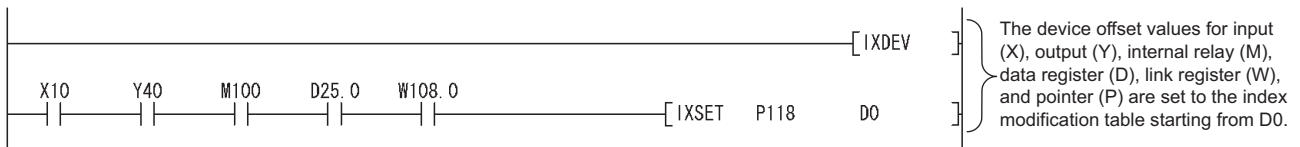


Figure App.4 Sample program

(b) Program after replacement

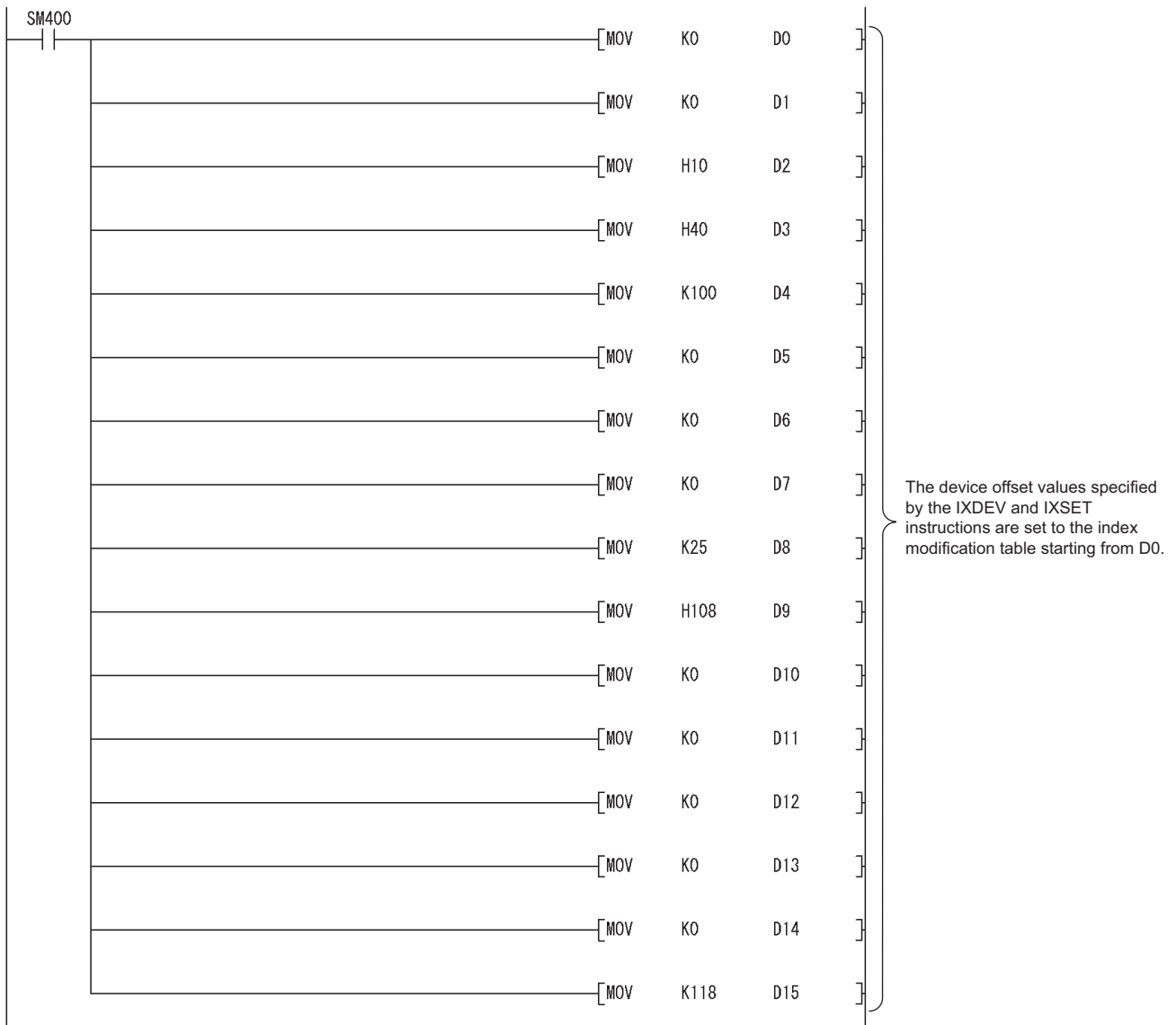


Figure App.5 Sample program

(3) Replacement example of the PR instruction

The number of output characters can be switched by the on/off status of SM701.

(a) Example of device assignment

TableApp.30 Example of device assignment

Before replacemen		After replacement	
Application	Device	Application	Device
Output string	D0 to D3	Output string	D0 to D3
ASCII code output signal	Y100 to Y107	ASCII code output signal	Y100 to Y107
Strobe signal	Y108	Strobe signal	Y108
In-execution flag	Y109	In-execution flag	Y109
		Output string storage address (BIN32)	D20 to D21
		Output string storage address (BIN32) (Used for sub-routine programs and interrupt programs)	D200 to D201
		Number of output characters	D202
		Output module start Y number	D203
		Character extraction position	D204
		Number of extracted characters	D205
		String output status value	D206
		Result of string extraction by the MIDR instruction	D207
		String output in-execution flag	M200
		For index modification	Z0

If the device numbers in the example above are used for other applications, assign unused device numbers instead.

(b) Program before replacement

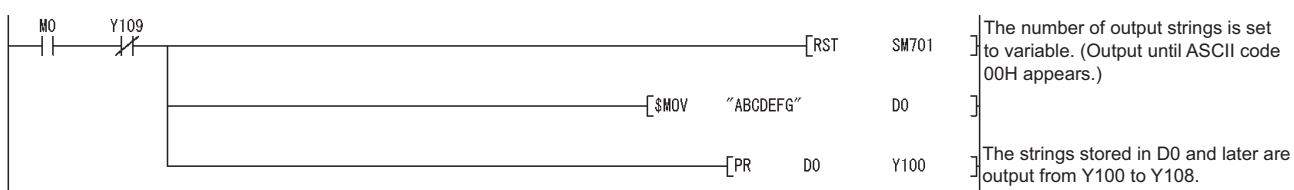


Figure App.6 Sample program

(c) Program after replacement

In the sequence program after replacement, three programs are required as shown below.

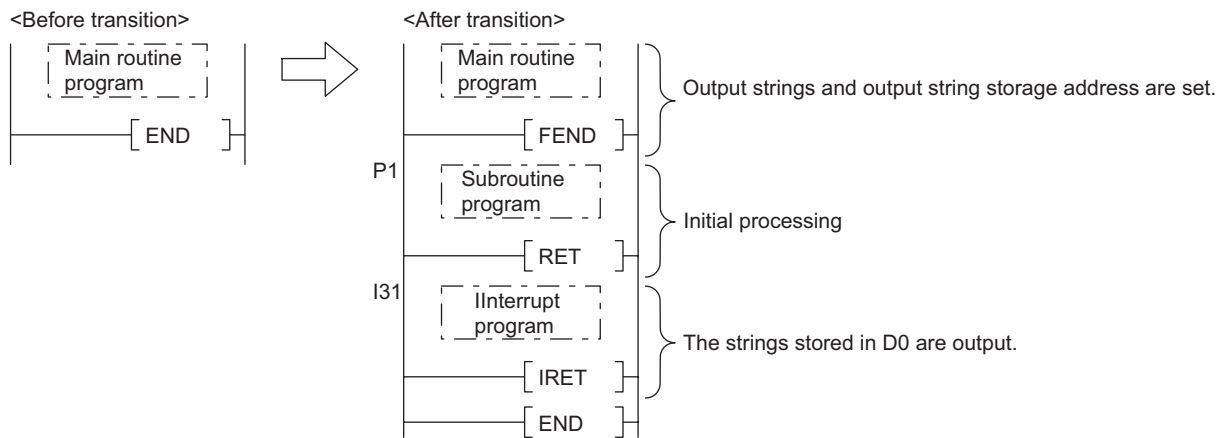


Figure App.7 Program execution

1) Main routine program

- Replace the PR instruction with the CALL instruction so that a subroutine program is called.
- Output string storage device ("D0" in the program below) cannot be specified directly with the CALL instruction.
- Use the ADRSET instruction to acquire the indirect address for the CALL instruction.
- Y device ("Y100" in the program before replacement shown in (b)) cannot be specified directly as output Y number with the CALL instruction. Specify the output Y number in integer.
- An interrupt program is used to output character codes via the output module. Enable the execution of interrupt programs using the EI instruction.

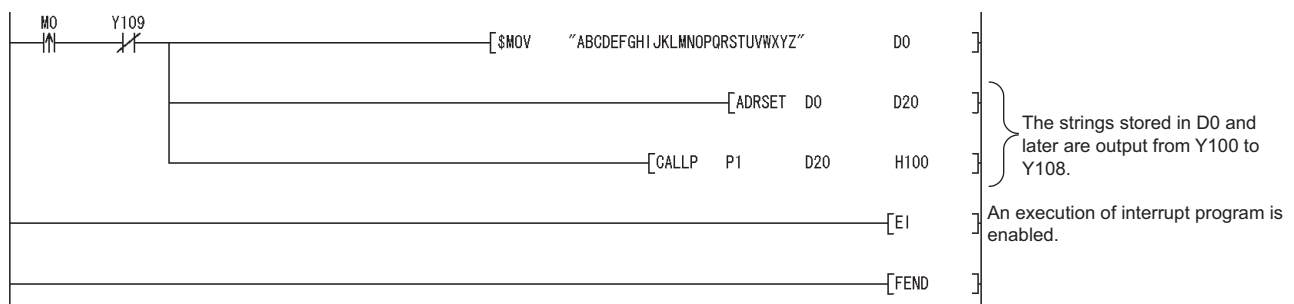


Figure App.8 Sample program

2) Subroutine program

- In the subroutine program, the data for outputting ASCII codes using a fixed scan interrupt program (10ms) are set to work devices. Also, the flag for activating the processing in the fixed scan interrupt program is turned on.
- Specify the following arguments for the subroutine program.

First argument	Output string storage address	(Input)
Second argument	Output module start Y number	(Input)

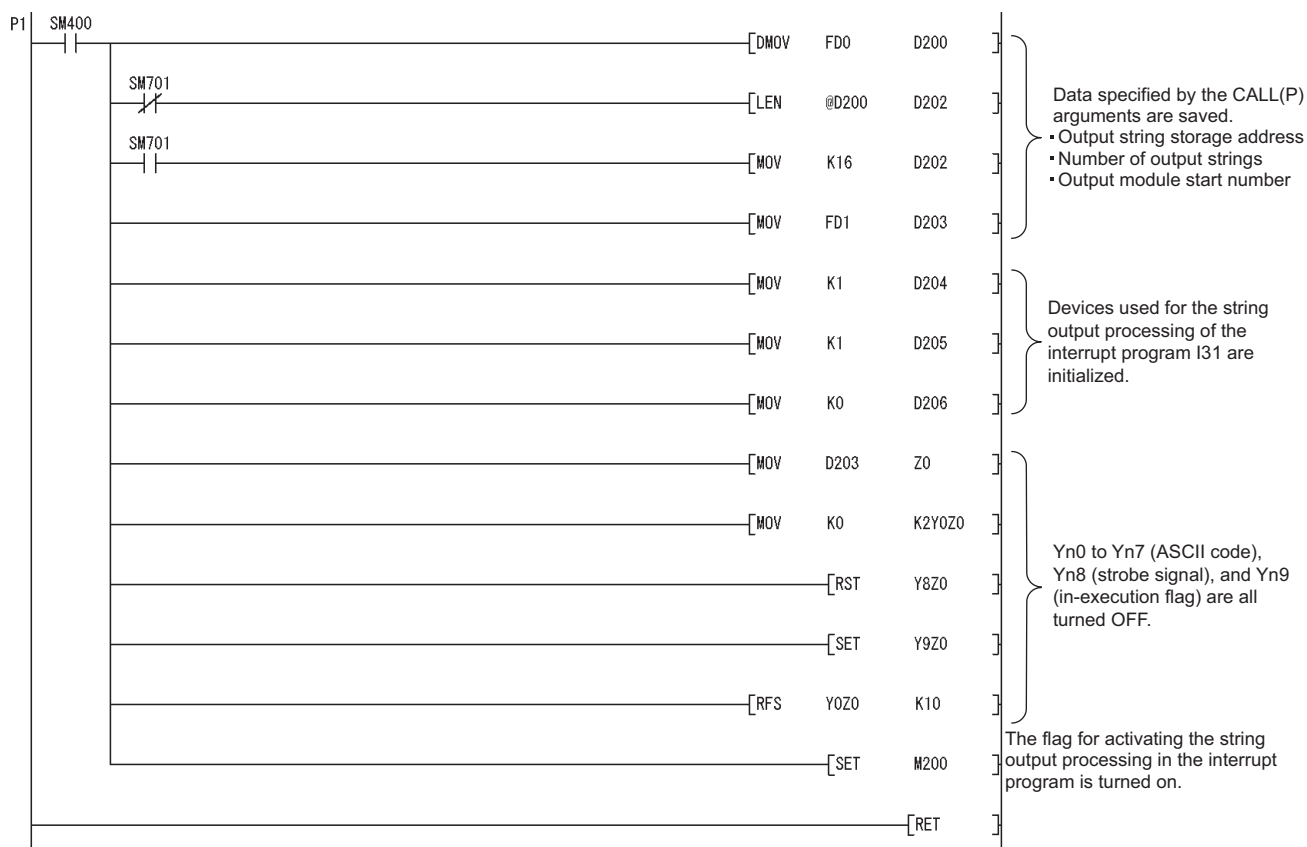


Figure App.9 Sample program

3) Interrupt program

The following processing is added to a fixed scan interrupt program (10ms).

The fixed scan interrupt program outputs ASCII codes from the output module and controls the strobe signal.

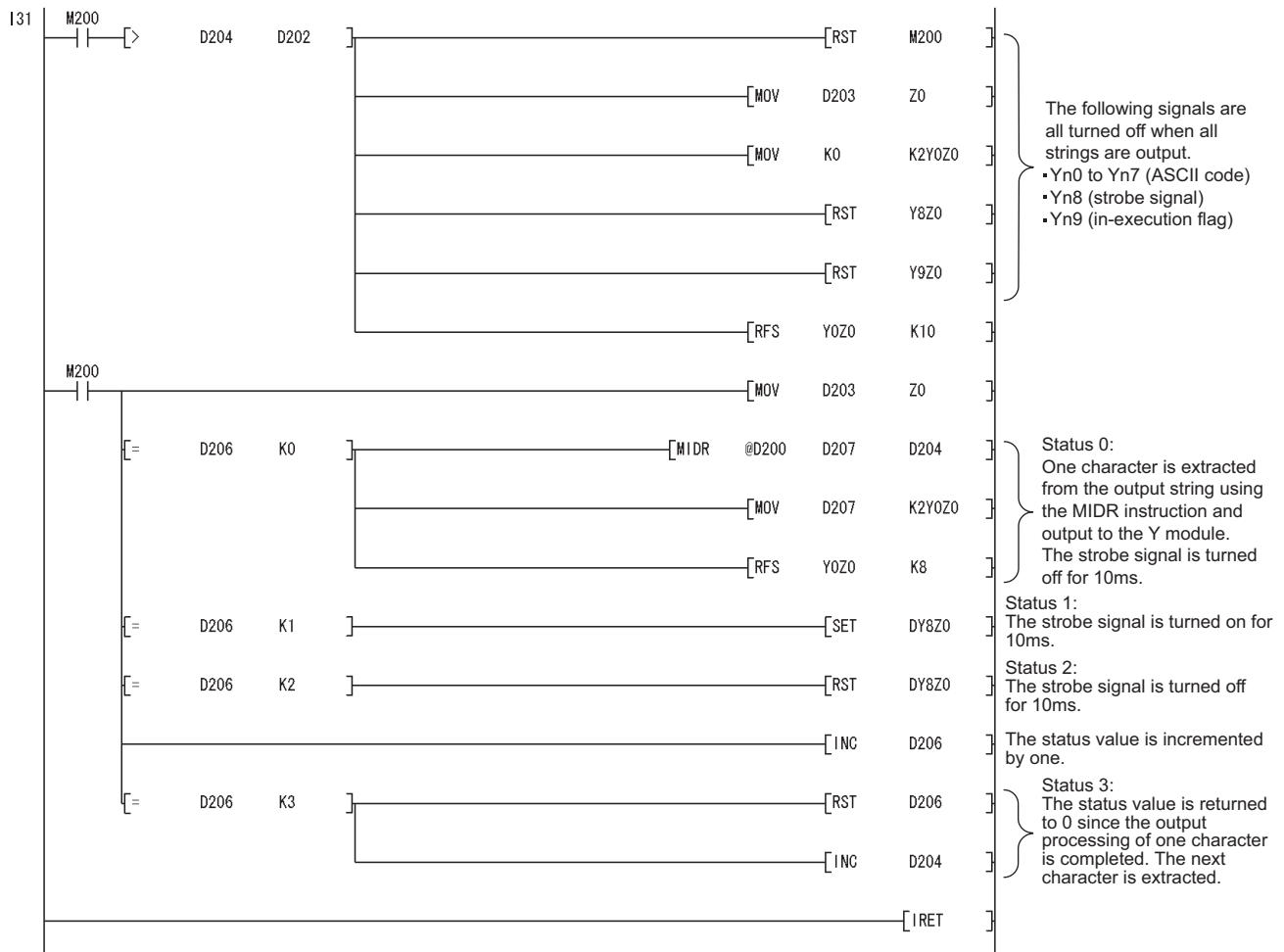


Figure App.10 Sample program

(4) Replacement example of the CHKST and CHK instructions

In the example below, if the replacement program for the CHKST and CHK instructions detects a failure, a failure number (contact number + coil number) is stored in D200 and the annunciator F200 is turned on.

(a) Example of device assignment

TableApp.31 Example of device assignment

Before replacement		After replacement	
Application	Device	Application	Device
Advance end detection sensor input 1	X100	Advance end detection sensor input 1	X100
Retract end detection sensor input 1	X101	Retract end detection sensor input 1	X101
Advance end detection sensor input 2	X102	Advance end detection sensor input 2	X102
Retract end detection sensor input 2	X103	Retract end detection sensor input 2	X103
Advance end detection sensor input 3	X104	Advance end detection sensor input 3	X104
Retract end detection sensor input 3	X105	Retract end detection sensor input 3	X105
Advance end detection sensor input 4	X106	Advance end detection sensor input 4	X106
Retract end detection sensor input 4	X107	Retract end detection sensor input 4	X107
Failure detection output 1	Y100	Failure detection output 1	Y100
Failure detection output 2	Y102	Failure detection output 2	Y102
Failure detection output 3	Y104	Failure detection output 3	Y104
Failure detection output 4	Y106	Failure detection output 4	Y106
		Coil number (failure type detected)	D100
		Contact number	D101
		Failure number	D200
		Failure detection display	F200
		For index modification	Z0

If the device numbers in the example above are used for other applications, assign unused device numbers instead.

When the advance end detection sensor input performs a failure detection of X_n, assign device numbers for the retract end detection sensor input and the failure detection output as described below.

Advance end detection sensor input	X _n
Retract end detection sensor input	X _{n+1}
Failure detection output	Y _n

(b) Program before replacement

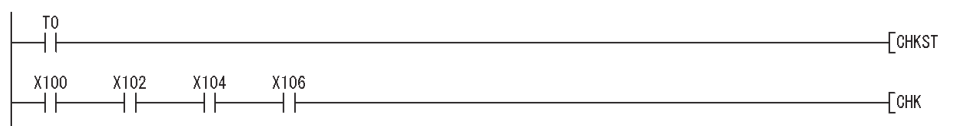


Figure App.11 Sample program

(c) Program after replacement

In the sequence program after replacement, two programs are required as shown below.

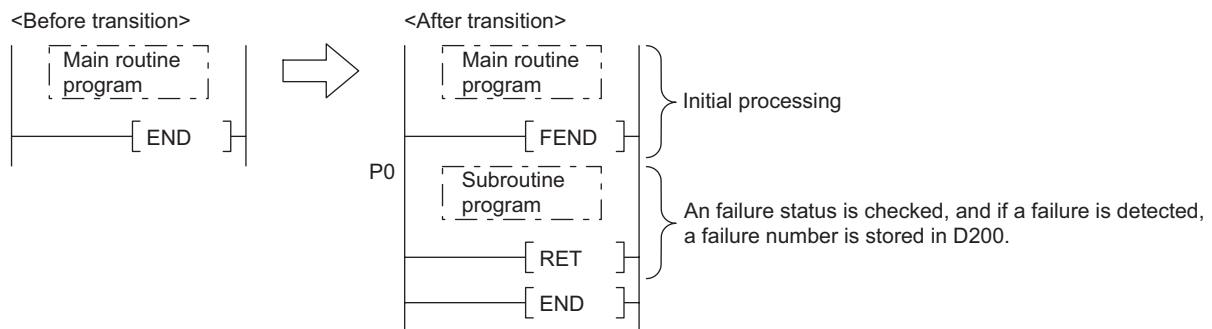


Figure App.12 Program execution

1) Main routine program

- Replace the CHKST and CHK instructions with the CALL instructions so that a subroutine program is called.
- One CALL instruction is required for each device specified as check condition in front of the CHK instruction. (In the program before replacement shown in (b), four CALL instructions need to be added since there are four check conditions in front of the CHK instruction.)
- Device number and contact number of X devices (check condition) are specified in each CALL instruction.
- Contact number is used to display failure number when a failure is detected.



Figure App.13 Sample program

2) Subroutine program

- In the subroutine program, an failure status is checked using a failure detection ladder pattern.
- If a failure is detected, a failure number is stored in D200 and the annunciator F200 is turned on.
- Specify the following arguments for the subroutine program.

First argument	Device number of X device targeted for failure check	(Input)
Second argument	Contact number of X device targeted for failure check	(Input)

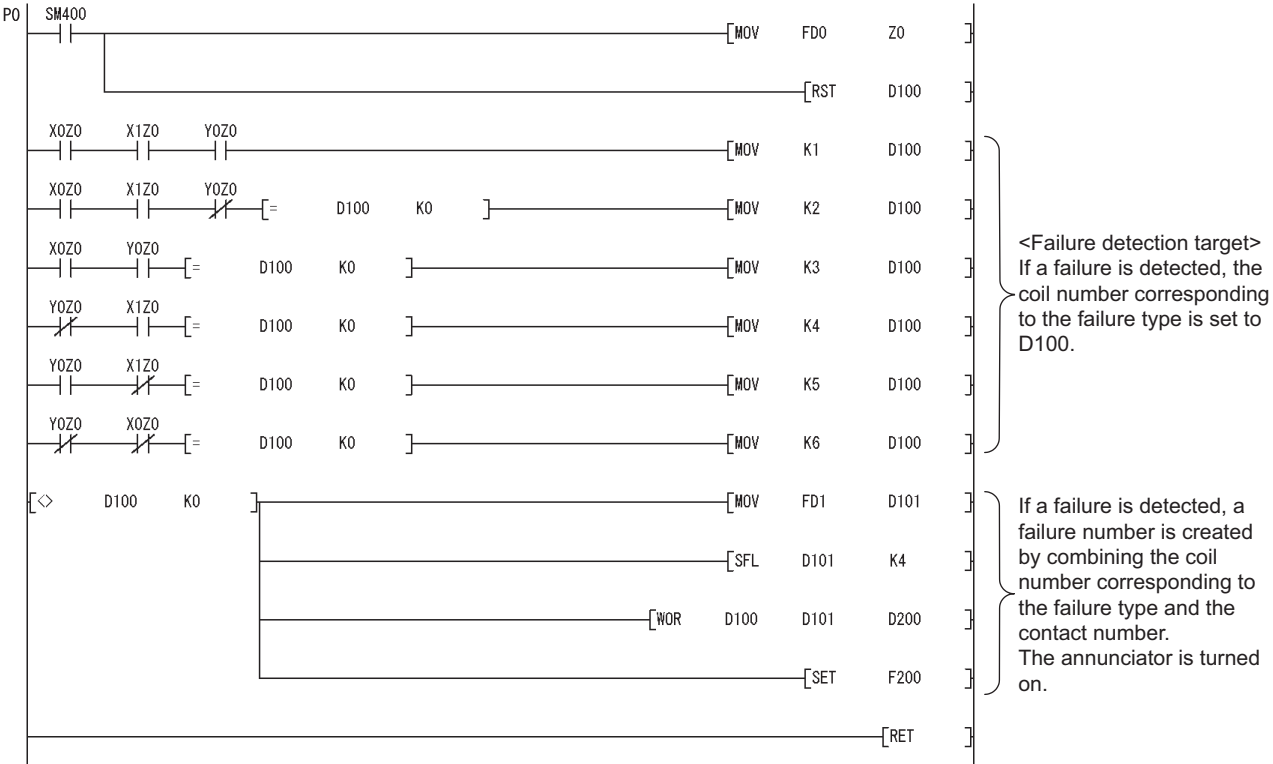


Figure App.14 Sample program

(d) Replacement method when failure detection ladder patterns are changed by the CHKCIR and CHKEND instructions

Failure detection ladder patterns can be changed in the subroutine program described in (c).

(5) Replacement example of the KEY instruction

(a) Example of device assignment

TableApp.32 Example of device assignment

Before replacemen		After replacement	
Application	Device	Application	Device
Numeric input execution instruction	M0	Numeric input execution instruction	M0
Input complete flag	M1	Input complete flag	M1
Input data area	D200 to D203	Input data area	D200 to D203
ASCII code input signal	X100 to X107	ASCII code input signal	X100 to X107
Strobe signal	X108	Strobe signal	X108
		Input data area address (BIN32)	D210 to D211
		(Input data area + 0) address (BIN32)	D212 to D213
		(Input data area + 1) address (BIN32)	D214 to D215
		(Input data area + 2) address (BIN32)	D216 to D217
		For shifting input data	D218
		For converting input data	D219 to D220

If the device numbers in the example above are used for other applications, assign unused device numbers instead.

(b) Program before replacement



Figure App.15 Sample program

(c) Program after replacement

In the sequence program after replacement, two programs are required as shown below.

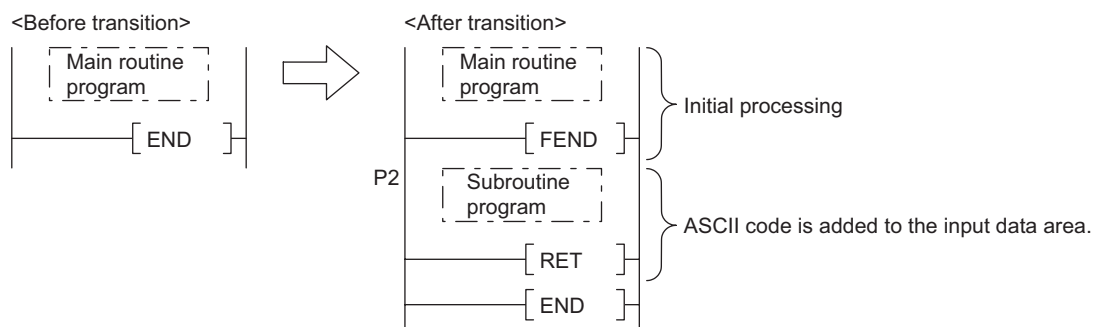


Figure App.16 Program execution

APPEN-
DIX

1) Main routing program

- Set "0" in the input data area on the rising edge of the execution instruction ("M0" in the program below) and initialize the program.
- Execute the CALL instruction on every rising edge of the strobe signal ("X108" in the program below) so that a subroutine program is called.
- In the subroutine program, input codes are added to the input data area and the completion status is checked.
- Pass the following data to the subroutine program at execution of the CALL instruction.
 - 1) ASCII code input values from the input module (Xn0 to Xn7)
 - 2) Number of digits to be input
 - 3) Indirect address of the input data area (Use the ADRSET instruction to acquire the indirect address for the input data area.)
 - 4) Bit devices to be turned on when input is completed

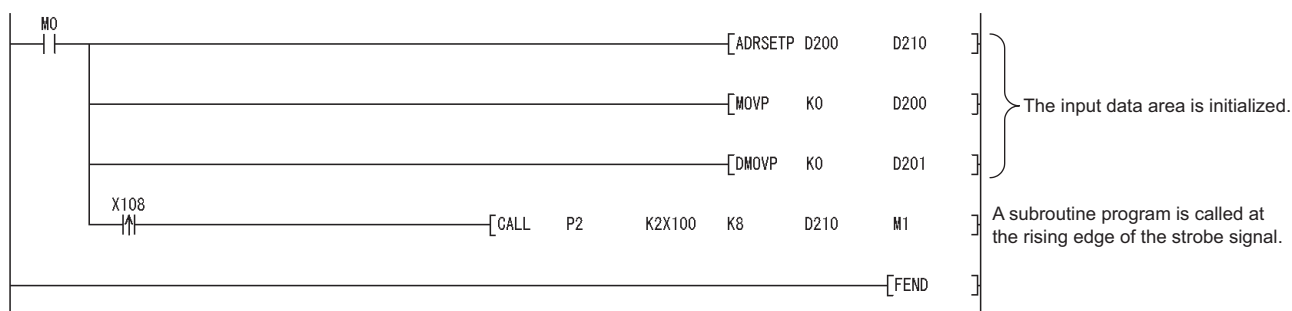


Figure App.17 Sample program

2) Subroutine program

- In the subroutine program, ASCII codes specified by an argument are added to the input data area and the completion status is checked.
- Specify the following arguments for the subroutine program.

First argument	ASCII code input from the input module (K2Xn)	(Input)
Second argument	Number of digits to be input	(Input)
Third argument	Indirect address of the input data area	(Input)
Fourth argument	Bit device to be turned on when input is completed	(Output)

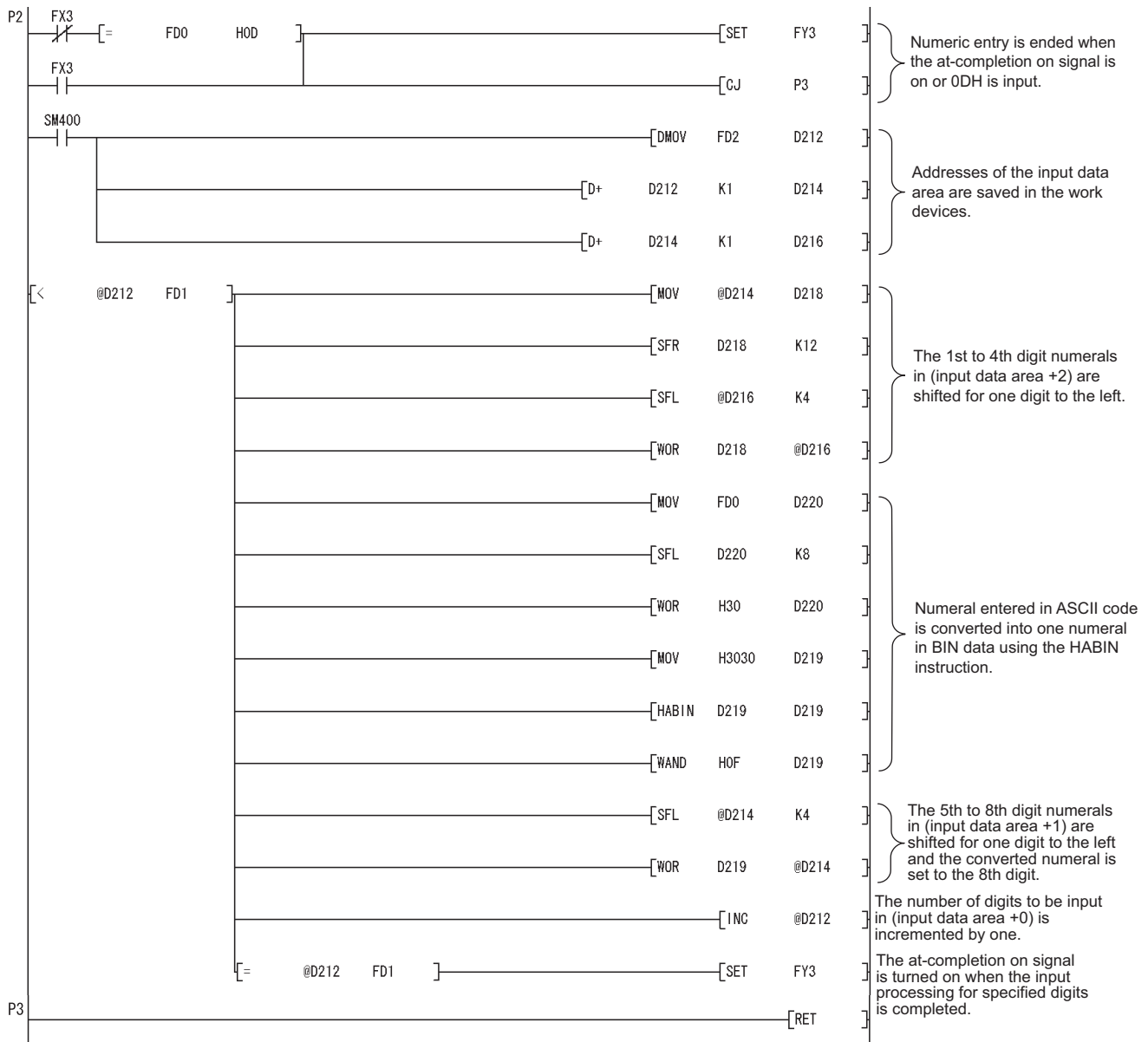


Figure App.18 Sample program

Appendix 3.4 Functions

Appendix 3.4.1 Floating-point operation instructions

(1) Differences between High Performance model QCPU and Universal model QCPU

(a) High Performance model QCPU

The High Performance model QCPU can perform only the single-precision floating-point operation instructions. Note, however, that internal operation processing can be performed in double precision by selecting the item shown below (default: selected).

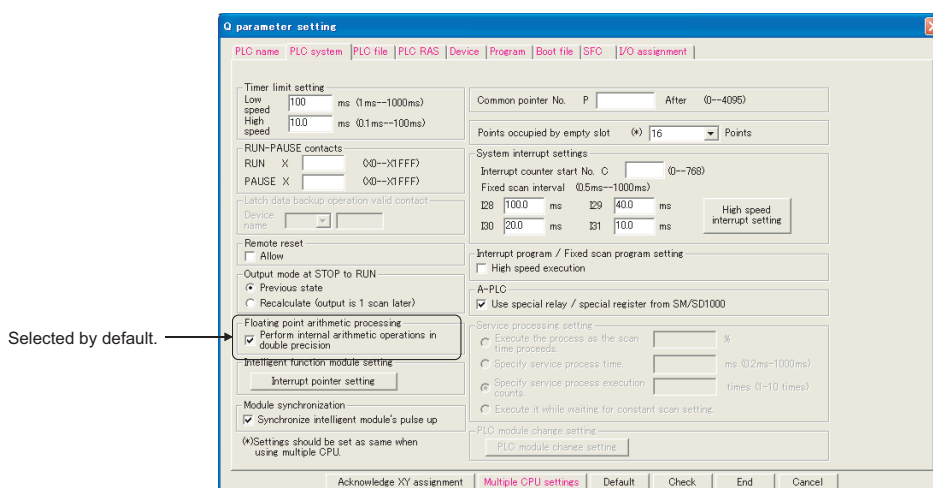


Figure App.19 PLC system tab

(b) Universal model QCPU

The Universal model QCPU supports the double-precision floating-point operation instructions.

The operation can be performed either in single precision or double precision depending on the data. Therefore, the "Perform internal arithmetic operations in double precision" item in the PLC system tab of the PLC parameter dialog box cannot be selected.

Because of this new function, operation results (both in single precision and double precision) slightly differ between the High Performance model QCPU and the Universal model QCPU if "Perform internal arithmetic operations in double precision" is selected in the High Performance model QCPU.

If higher accuracy is required in floating-point operations, replace the floating-point operation instructions as described in (4).

However, if six or less digits are used as significant digits for the floating-point operation instructions, replacement is not necessary. The single-precision floating-point operation results in the Universal model QCPU can be used as they are in the system. When not replacing instructions, make sure that it does not cause any problems in the system.

(2) Floating-point operation instructions for the Universal model QCPU

TableApp.31 lists floating-point operation instructions for the Universal model QCPU.

Specifications of the single-precision floating-point operation instructions are compatible with those for the High Performance model QCPU.

TableApp.33 List of floating-point operation instructions supported in the Universal model QCPU

Instruction name		Instruction symbol		Remarks
		Single-precision floating-point data	Double-precision floating-point data	
Comparison	Floating-point data comparison	LDE[]	LEDED[]	[] indicates one of the followings; <>, =, <, >, <=, >=
		ANDE[]	ANDED[]	
		ORE[]	ORED[]	
Data transfer	Floating-point data transfer	EMOV(P)	EDMOV(P)	---
Four arithmetic operation	Floating-point data addition	E+(P)	ED+(P)	---
	Floating-point data subtraction	E-(P)	ED-(P)	
	Floating-point data multiplication	E*(P)	ED*(P)	
	Floating-point data division	E/(P)	ED/(P)	
Data conversion	Conversion from BIN 16-bit data to floating-point data	FLT(P)	FLTD(P)	---
	Conversion from BIN 32-bit data to floating-point data	DFLT(P)	DFLTD(P)	
	Conversion from floating-point data to BIN 16-bit data	INT(P)	INTD(P)	
	Conversion from floating-point data to BIN 32-bit data	DINT(P)	DINTD(P)	
	Floating-point sign inversion	ENEG(P)	EDNEG(P)	
Special function	SIN operation	SIN(P)	SIND(P)	---
	COS operation	COS(P)	COSD(P)	
	TAN operation	TAN(P)	TAND(P)	
	SIN-1 operation	ASIN(P)	ASIND(P)	
	COS-1 operation	ACOS(P)	ACOSD(P)	
	TAN-1 operation	ATAN(P)	ATAND(P)	
	Conversion from angle to radian	RAD(P)	RADD(P)	
	Conversion from radian to angle	DEG(P)	DEGD(P)	
	Square root	SQR(P)	SQRD(P)	
	Exponential operation	EXP(P)	EXPD(P)	
	Natural logarithm operation	LOG(P)	LOGD(P)	

Floating-point data can be converted mutually between single precision and double precision using instructions in TableApp.34.

TableApp.34 Floating-point data conversion instructions (single precision↔double precision)

Instruction name	Instruction symbol
Single precision to double precision conversion	ECON(P)
Double precision to single precision conversion	EDCON(P)

(3) Advantages and disadvantages when using the double-precision floating-point data of the Universal model QCPU

TableApp.33 shows the advantages and disadvantages when executing the double-precision floating-point operation instructions in the Universal model QCPU.

If higher accuracy is required in floating-point operations, it is recommended to replace the instructions with the double-precision floating-point operation instructions.

TableApp.35 Advantages and disadvantages when using the double-precision floating-point operation instructions

Advantage	Disadvantage
The results are more accurate than those of the single-precision floating-point operation instructions.	The instruction processing speed is slower than that of the single-precision floating-point operation instructions. *1 Double-precision floating-operation data use twice as many word device points as single-precision floating-operation data.

*1: The processing speed of the double-precision floating-point operation instructions in the Universal model QCPU is higher than that of floating-point operation instructions using internal double-precision operations in the High Performance model QCPU.

TableApp.34 shows the comparison between single-precision and double precision floating-point data.

TableApp.36 Comparison between single-precision and double precision floating-point data

Item		Single-precision floating-point data	Double-precision floating-point data
Word point required for data retention		2 words	4 words
Setting range		$-2^{128} < N \leq -2^{126}, 0,$ $2^{-126} \leq N < 2^{128}$	$-2^{1024} < N \leq -2^{1022}, 0,$ $2^{-1022} \leq N < 2^{1024}$
Precision (number of bits)	Mantissa	23 bits	52 bits
	Exponent	8 bits	11 bits
	Sign	1 bits	1 bits
Instruction processing speed (Q04UDHCPU/ Q06UDHCPU) (Maximum)	Data comparison (Conductive status) (LDE>= / LDED>=)	5.5 μ s	9.0 μ s
	Data transfer (EMOV/EDMOV)	0.019 μ s	5.0 μ s
	Addition (3 devices) (E+ / ED+)	0.0665 μ s	9.2 μ s
	SIN operation (SIN/SIND)	5.7 μ s	13.8 μ s

(4) Replacing the High Performance model QCPU with the Universal model QCPU

(a) Replacing all single-precision floating-point operation instructions with double-precision floating-point operation instructions

Single-precision floating-point data occupy two points of word device per data.

On the other hand, four points are required per double-precision floating-point data.

Therefore, all device numbers for storing floating-point data need to be reassigned.

Example Replacing the floating-point operation $[A \times B+C]$ (Changing all floating-point data into double precision.)

1) Device assignment

TableApp.37 Device assignment

Before replacement			After replacement		
Application	Device	Data type	Application	Device	Data type
Data A	D0 to D1	Floating-point data (single precision)	Data A	D0 to D3	Floating-point data (double precision)
Data B	D2 to D3		Data B	D4 to D7	
Data C	D4 to D5		Data C	D8 to D11	
Result	D6 to D7		Result	D12 to D15	

2) Program before replacement

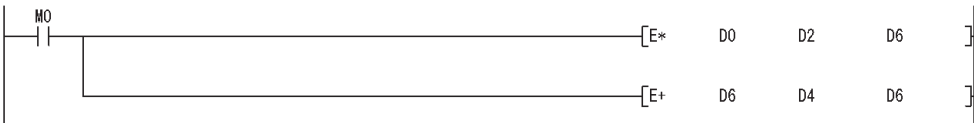


Figure App.20 Sample program

3) Program after replacement

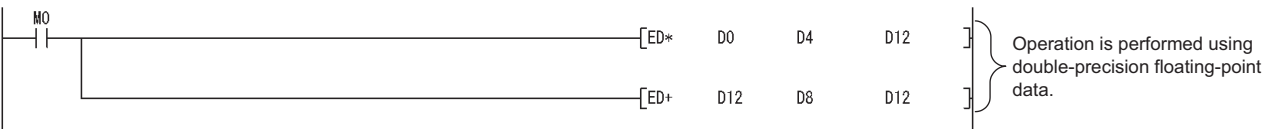


Figure App.21 Sample program

(b) **Replacing a part of floating-point operation instructions with double-precision floating-point operation instructions**

Only operations that require high accuracy are replaced with double-precision floating-point operation instructions.

Using the ECON and EDCON instructions, convert floating-point data mutually between single precision and double-precision. The flow of a replacement program is as follows:

- Data required for operations are converted from single precision to double precision using the ECON instruction.
- Operations are performed in double precision using the double-precision floating-point operation instructions.
- Operation results are converted from double precision to single precision using the EDCON instruction.

A program example that floating-point data are converted mutually between single precision and double precision before and after operations is shown below.

Example

Replacing the floating-point operation $[A \times B + C]$ (Using the ECON and EDCON instructions)

1) Device assignment

TableApp.38 Device assignment

Before replacement		
Application	Device	Data type
Data A	D0 to D1	Floating-point data (single precision)
Data B	D2 to D3	
Data C	D4 to D5	
Result	D6 to D7	

After replacement		
Application	Device	Data type
Data A	D0 to D1	Floating-point data (single precision)
Data B	D2 to D3	
Data C	D4 to D5	
Result	D6 to D7	
Data A	D10 to D13	Floating-point data (double precision)
Data B	D14 to D17	
Data C	D18 to D21	
Result	D22 to D25	

2) Program before replacement

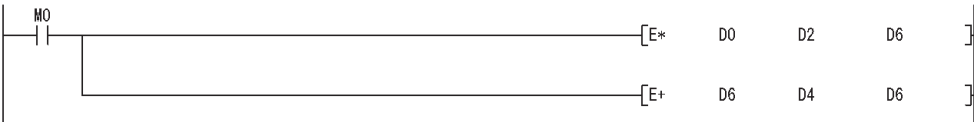


Figure App.22 Sample program

3) Program after replacement

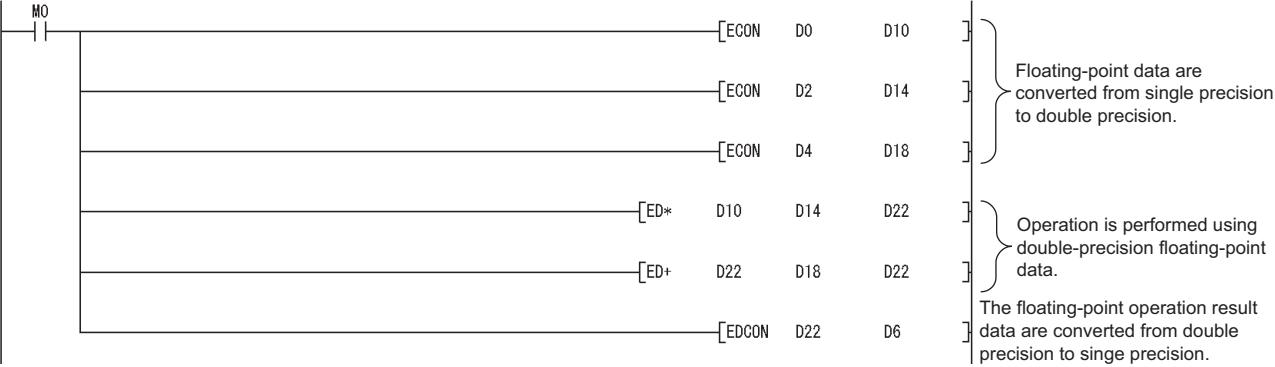


Figure App.23 Sample program

(c) Replacing a part of floating-point operation instructions with double-precision floating-point operation instructions using subroutine programs

The flow of a replacement program described in (b) can be regarded as one subroutine program.

Create a subroutine program for each floating-point operation instruction and then replace the original floating-point operation instructions with the CALL(P) instruction so that the corresponding subroutine program is called.

With this method, changes in the program are minimized, but the processing for calling subroutine programs increases the scan time.

In addition, since conversions from double precision to single precision are performed for each instruction, rounding-off errors generated during operations are larger than those in the replacement program described in (b).

Example Replacing the floating-point operation $[A \times B + C]$ (Using a subroutine program)

1) Device assignment

TableApp.39 Device assignment

Before replacement		
Application	Device	Data type
Data A	D0 to D1	Floating-point data (single precision)
Data B	D2 to D3	
Data C	D4 to D5	
Result	D6 to D7	

After replacement		
Application	Device	Data type
Data A	D0 to D1	Floating-point data (single precision)
Data B	D2 to D3	
Data C	D4 to D5	
Result	D6 to D7	
Subroutine input data 1	D900 to D903	Floating-point data (double precision)
Subroutine input data 2	D904 to D907	
Subroutine operation result	D908 to D911	

2) Program before replacement

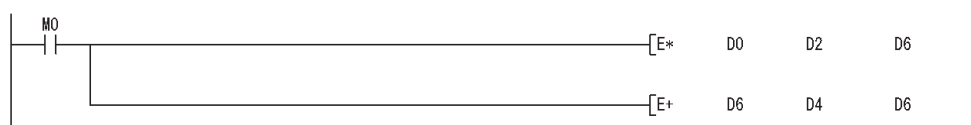


Figure App.24 Sample program

3) Program after replacement

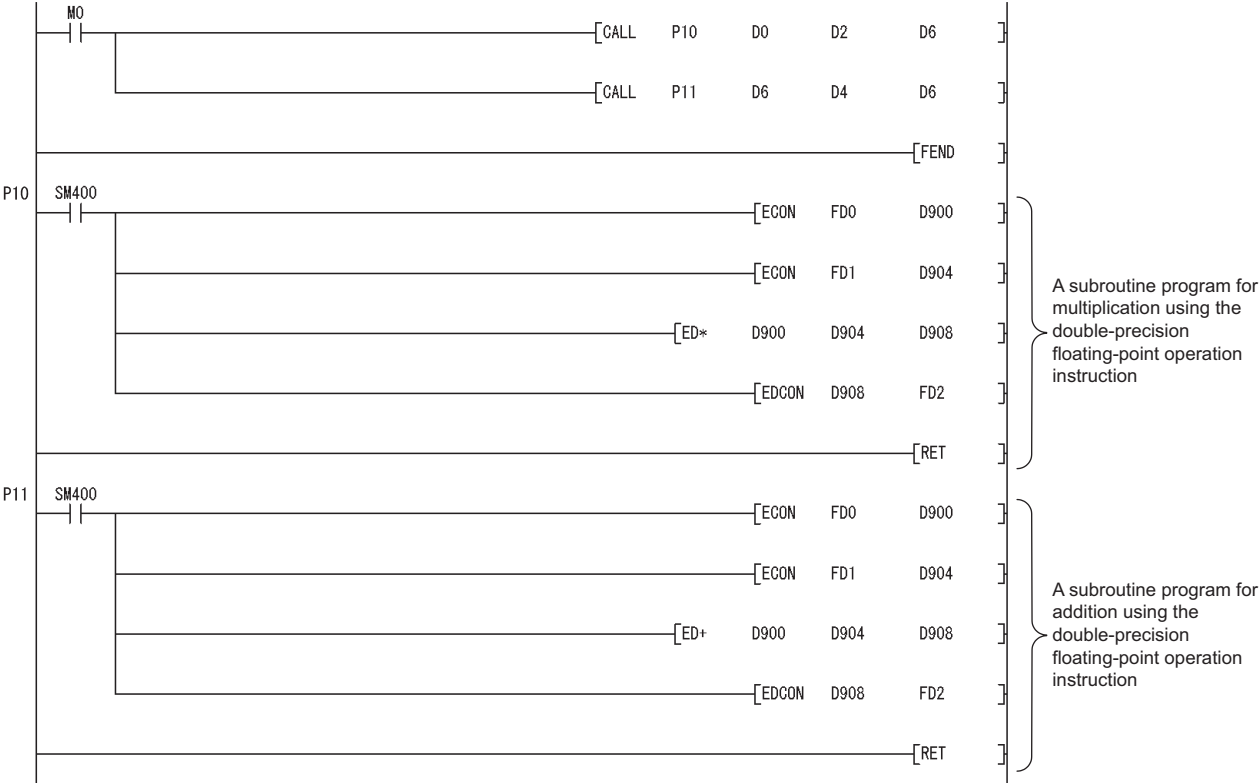


Figure App.25 Sample program

Appendix 3.4.2 Error check processing for floating-point data comparison instructions

(1) Input data check

Error check processing for floating-point data comparison instructions performed in the Universal model QCPU are enhanced. Input of a "special value" (-0, nonnumeric, unnormalized number, or ±∞) is checked, and if those special values are input, the CPU module detects "OPERATION ERROR" (error code: 4140).

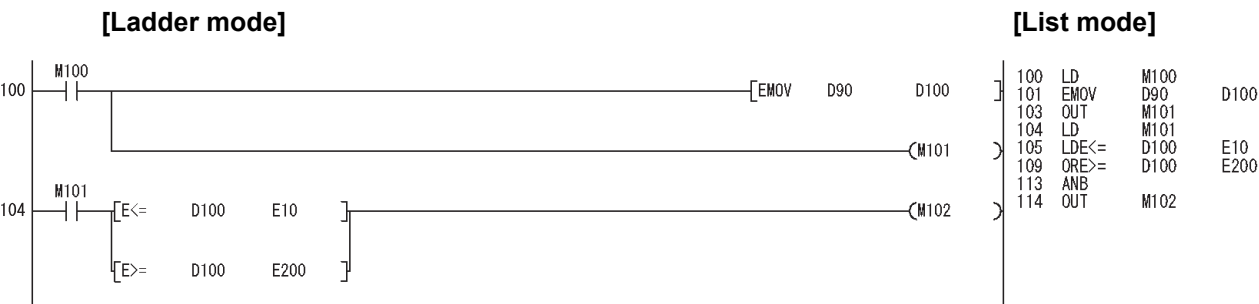
When the LDE□, ANDE□, ORE□, LDED□, ANDED□, and/or ORED□ instructions (□ indicates one of the followings; =, <>, <, >, <=, >=) are used in the program, "OPERATION ERROR" (error code: 4140) may be detected if invalid floating-point data exist. This occurs even when interlocks are provided using the valid data flags (the signal which shows the floating-point validity).

Invalid floating-point data are not stored in the result of operations performed in the Universal model QCPU. Those invalid data are considered to be stored in the following cases:

- The same device is used for storing floating-point data and other data, such as binary values, BCD values, and strings.
 - Use different devices for storing floating-point data and data other than floating-point data.
- Floating-point data externally written are invalid.
 - Take measures on the external-source side so that valid data are written.

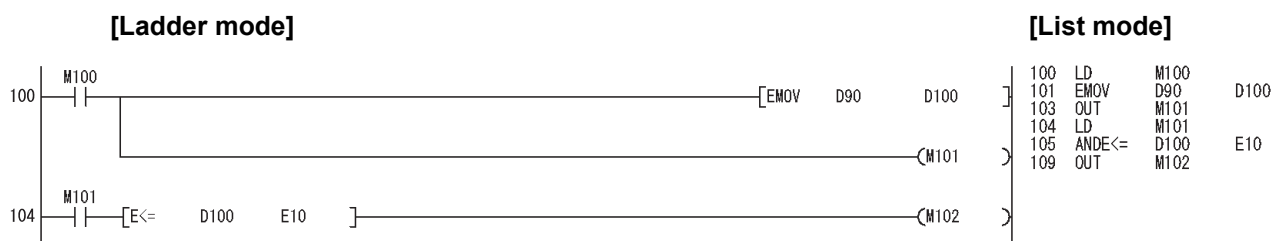
If an error occurs in the floating-point data comparison instructions, take appropriate measures to remove error causes described above.

Example 1) Detecting "OPERATION ERROR" (error code: 4140) in the LDE□ instruction



In the ladder block starting from the step 104, the floating-point data comparison instructions of the step 105 and 109 are not executed when the M101 (valid data flag) is off. However, the LDE<= instruction of the step 105 and the ORE>= instruction of the step 109 are executed regardless of the execution result of the LD instruction of the step 104 in the program above. Therefore, even when the M101 is off, "OPERATION ERROR" (error code: 4140) will be detected in the LDE<= instruction of the step 105 if a "special value" is stored in D100. For the method of avoiding "OPERATION ERROR", refer to(2) in this section.

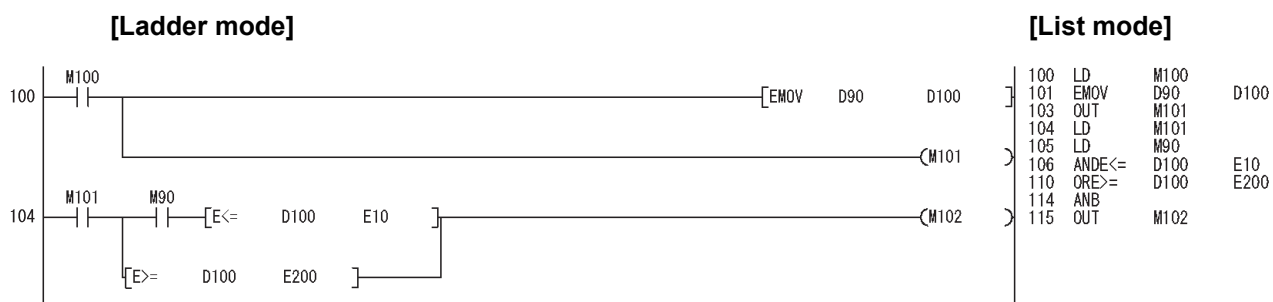
Example 2) Not detecting "OPERATION ERROR" (error code: 4140) in the ANDE<= instruction



In the ladder block starting from the step 104, the ANDE<= instruction of the step 105 is not executed when the M101 (valid data flag) is off.

The ANDE<= instruction of the step 105 is not executed when the M101 is off in the LD instruction of the step 104 in the program above. Therefore, when the M101 is off, "OPERATION ERROR" (error code: 4140) will not be detected even if a "special value" is stored in D100.

Example 3) Detecting "OPERATION ERROR" (error code: 4140) in the ANDE<= instruction



In the ladder block starting from the step 104, the ANDE<= instruction of the step 106 and the ORG>= instruction of the step 110 are not executed when the M101 (valid data flag) is off.

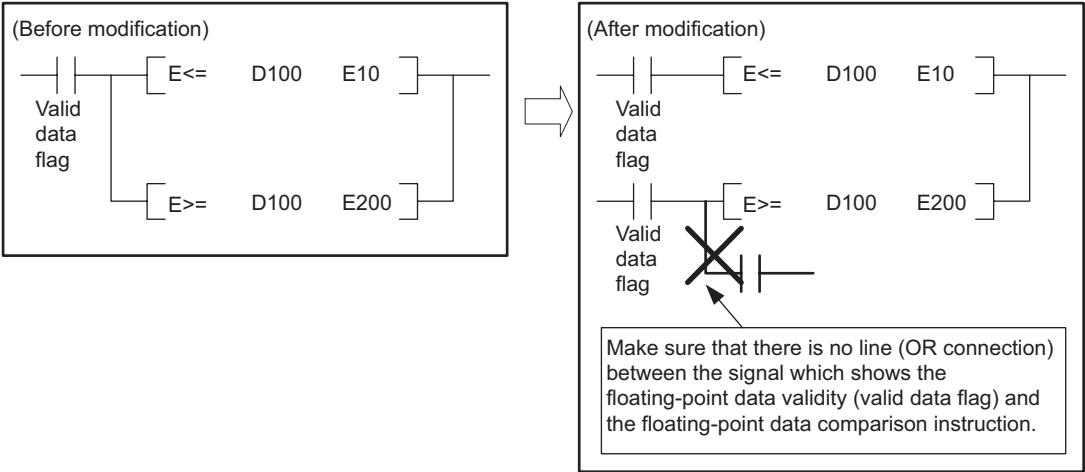
However, if the M90 is on in the LD instruction of the step 105, the ANDE<= instruction of the step 106 is executed. Therefore, even when the M101 is off, "OPERATION ERROR" (error code: 4140) will be detected in the ANDE<= instruction of the step 106 if the M90 is on and a "special value" is stored in D100.

For the method of avoiding "OPERATION ERROR", refer to (2) in this section.

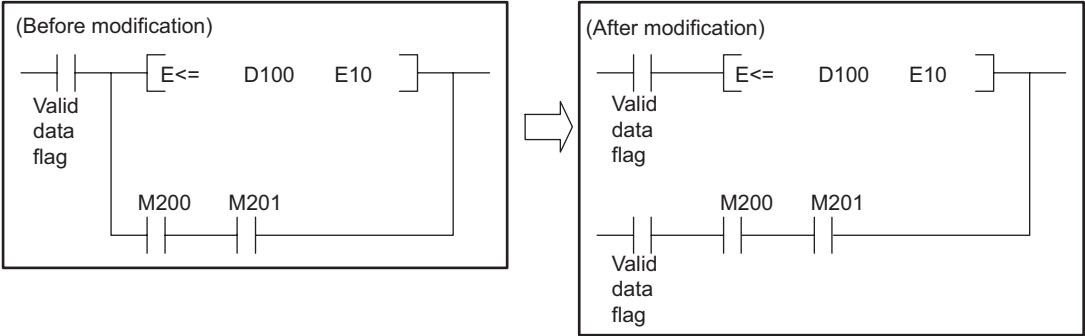
(2) Method of avoiding “OPERATION ERROR” (error code: 4140) in the floating-point data comparison instructions

As shown in the modification examples below, connect the contacts of valid data flag in series for each floating-point data comparison instruction. (Use AND connection for connecting the contact of the valid data flag and the floating-point data comparison instruction.)
Make sure that there is no line (OR connection) between the valid data flag and the floating-point data comparison instruction.

<Modification example 1>

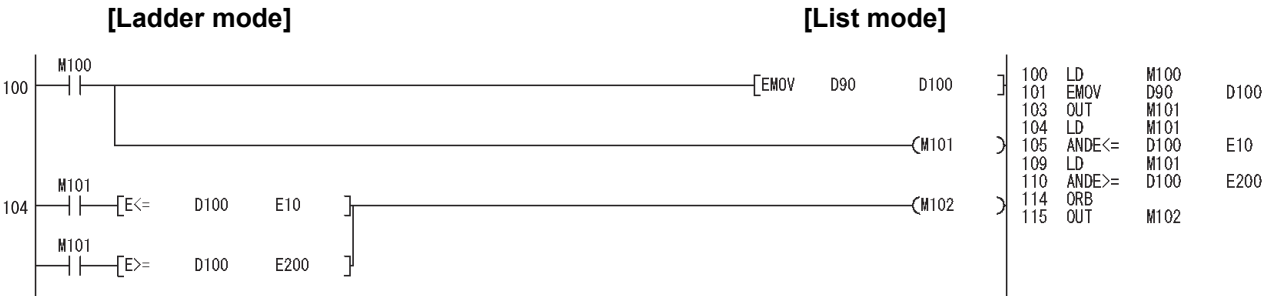


<Modification example 2>

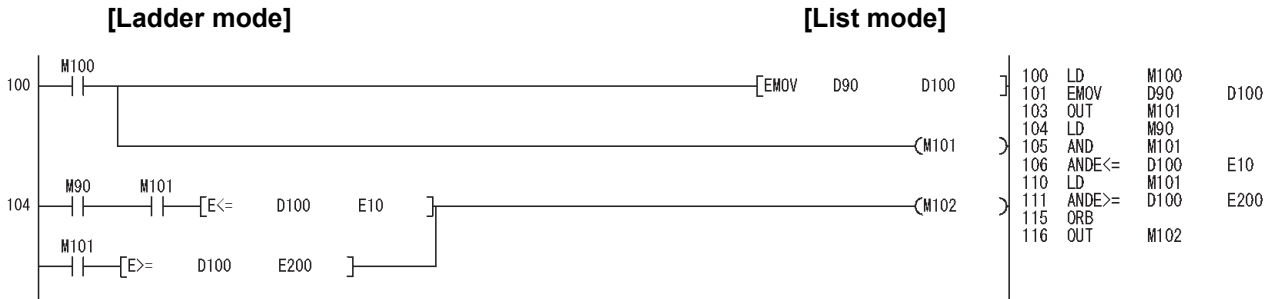


Program examples after modification for Example 1) and 3) in (1) are shown below.

Example 4) Program after modification for Example 1) ("OPERATION ERROR" (error code: 4140) is no longer detected.)



Example 5) Program after modification for Example 3) ("OPERATION ERROR" (error code: 4140) is no longer detected.)

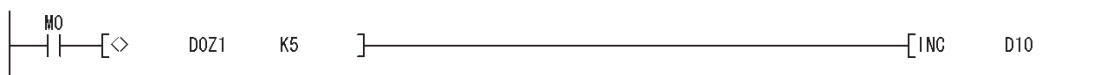


Appendix 3.4.3 Range check processing for index-modified devices

(1) Device range check

Error check processing at index modification of devices has been enhanced for the Universal model QCPU. Each index-modified device range is checked, and if the check target device is outside the device range before index modification, the CPU module detects "OPERATION ERROR" (error code: 4101).

Example 1) Detecting "OPERATION ERROR" (error code: 4101) by error check processing at index modification of devices



In Example 1), when the contact (M0) is on and the value, -1 or less, is specified in Z1, the device D0Z1 is included in the C device range, exceeding the D device range, as shown in Figure App.26.

As a result, "OPERATION ERROR" (error code: 4101) will be detected.

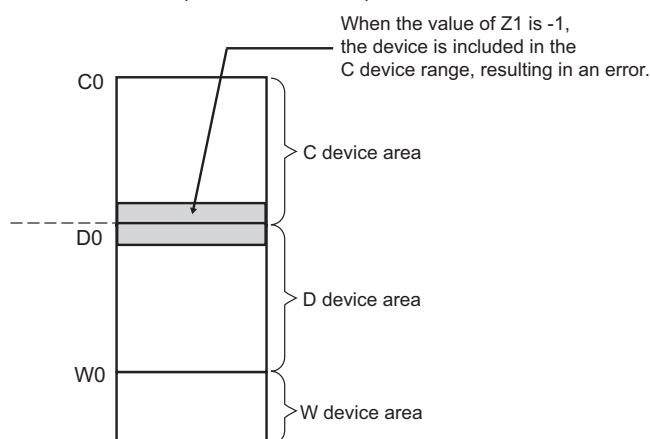
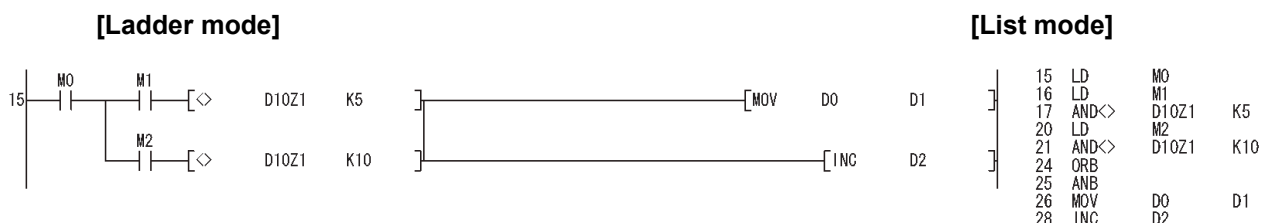


Figure App.26 Device D0Z1 when the value of Z1 is -1

When an error is detected, check the index modification value (value of Z1 in the above example) and remove the error cause.

Examples of the cases where an error is detected and not detected are shown below.

Example 2) Detecting "OPERATION ERROR" (error code: 4101)



In Example 2, in the ladder block starting from the step 15, the AND<> instruction of the step 17 or 21 is supposed to be not executed when M0 (valid data flag) is off.

However, since the LD instruction which is always executed is used in the step 16 and 20, the AND<> instruction of the step 17 or 21 is executed regardless of the execution status of the LD instruction in the step 15 when M1 or M2 is on.

For this reason, even when M0 is off, if the D10Z1 value is outside the D device range, "OPERATION ERROR" (error code: 4101) will be detected in the AND<> instruction of the step 17.

Note that the step 26 (MOV D0 D1) and the step 28 (INC D2) are not executed.

For the method of avoiding "OPERATION ERROR" (error code: 4101), refer to (2) in this section.

Example 3) Not detecting "OPERATION ERROR" (error code: 4101)



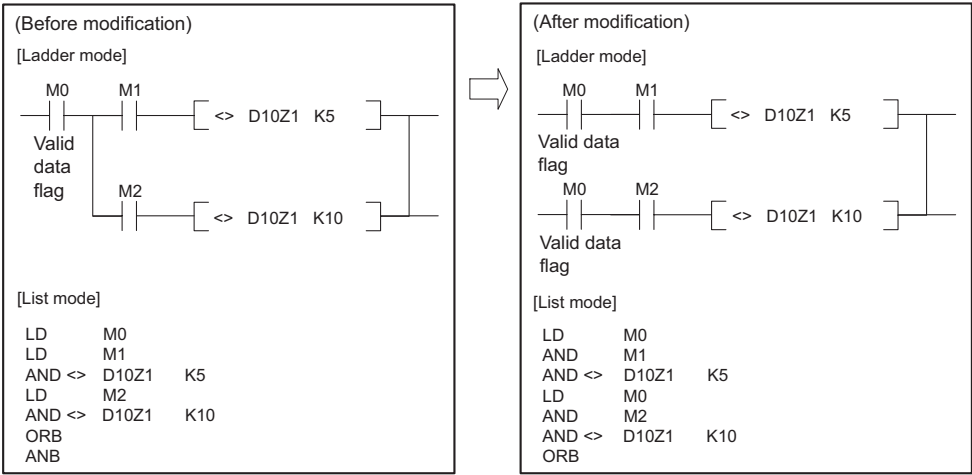
In Example 3, the AND<> instruction of the step 16 is not executed when M0 (valid data flag) of the step 15 is off. For this reason, "OPERATION ERROR" (error code: 4101) will not be detected no matter what the D10Z1 value is.

(2) Method of avoiding "OPERATION ERROR" (error code: 4101)

When the index-modified device range does not need to be checked, use the method 1).
When the index-modified device range needs to be checked, use the method 2).

- 1) Deselect the "Check device range at indexing" item in the PLC RAS tab of the PLC parameter dialog box so that the index-modified device range will not be checked.
- 2) As shown in the modification examples below, connect the contacts of valid data flag in series for each instruction that checks the index-modified device range.

<Modification example>



In the program before modification (on the left), the instruction immediately before the AND<> instruction is regarded as the LD instruction. However, in the program after modification (on the right), the same instruction will be regarded as the AND instruction.

In the program after modification, only when both contacts of M0 and M1 (or M2) turn on, the AND<> instruction is executed. As a result, no error will be detected during index-modified device range check processing.

Appendix 3.4.4 Device latch function

(1) Overview

The device latch function*1 for the Universal model QCPU is more enhanced compared to that for the Basic model QCPU and High performance model QCPU.

This section describes the enhanced device latch function in the Universal model QCPU.

*1: The latch function is used to hold device data even when the CPU module is powered off or reset.

(2) Device data latch methods

Device data of the Universal model QCPU can be latched by:

- using the large-capacity file register (R, ZR),
- writing/reading device data to the standard ROM (with the SP.DEVST and S(P).DEVLD instructions), or
- specifying a latch range of internal user devices.

(3) Details of each latch method

(a) Large-capacity file register (R, ZR)

Data in the file register can be latched by batteries.

File register size is larger and processing speed is higher in the Universal model QCPU, compared to the Basic model QCPU and High Performance model QCPU.

To latch a lot of data (many device points), use of the file register is effective.

TableApp.40 shows the file register size available for each CPU module.

TableApp.40 File register size available for each CPU module

Model	File register (R, ZR) size in the standard RAM
Q00UCPU,Q01UCPU,Q02UCPU	64K points
Q03UDCPU,Q03UDECPU	96K points
Q04UDHCPU,Q04UDEHCPU	128K points
Q06UDHCPU,Q06UDEHCPU	384K points
Q10UDHCPU,Q13UDHCPU,Q10UDEHCPU,Q13UDEHCPU	512K points
Q20UDHCPU,Q20UDHCPU,Q26UDEHCPU,Q26UDEHCPU	640K points

(b) Writing/reading device data to the standard ROM (SP.DEVST and S(P).DEVLD instructions)

Device data of the Universal model QCPU can be latched using the SP.DEVST and S (P).DEVLD instructions (instructions for writing/reading data to/from the standard ROM).

Utilizing the standard ROM allows data backup without batteries.

This method is effective for latching data that will be updated less frequently.

(c) Specifying the latch range of internal user devices

Device data of the Universal model QCPU can be latched by specifying a latch range of internal user devices in the same way as for the Basic model QCPU and High Performance model QCPU.

The ranges can be set in the Device tab of the PLC parameter dialog box.

Internal user devices that can be latched are as follows:

- Latch relay (L)
- Link relay (B)
- Annunciator (F)
- Edge relay (V)
- Timer (T)
- Retentive timer (ST)
- Counter (C)
- Data register (D)
- Link register (W)

Point

- If latch ranges of internal user devices are specified in the Universal model QCPU, the processing time will be added to the scan time in proportion to the device points set to be latched. (For example, if 8K points are latched for the latch relay (L), the scan time will be 28.6 μ s.) To shorten the scan time, remove unnecessary latch device points to minimize the latch range.
- The scan time will not increase when a latch range of the file register (R, ZR) is specified.

(4) How to shorten the scan time

When data to be latched are stored in the file register (R, ZR), the processing time is shorter than that for latching internal user device.

Example Reducing the latch points of the data register (D) from 8K points to 2K points, and using the file register (ZR) instead (when the Q06UDHCPU is used)

TableApp.41 Differences between before and after moving latch points of the data register (D) to the file register (ZR)

Item		Before	After
Latch points of the data register (D)		8192 (8K) points	2048 (2K) points (6K points are moved to the file register.)
Number of devices in the program	Data register (D) (Latch range)	400	100
	File register (ZR) (Standard RAM)	0	300
Additional scan time		0.41ms	0.13ms ^{*1}
Number of steps increased		---	300 steps

*1: Time indicates the time required additionally when the file register is stored in the standard RAM.

Appendix 3.4.5 File usability setting

(1) Differences between High Performance model QCPU and Universal model QCPU

(a) High Performance model QCPU

In the High Performance model QCPU, file usability ("Use PLC file setting" or "Not used") of the following files can be set for each program on the screen opened by clicking the "File usability setting" button on the Program tab of the PLC parameter dialog box.

- File register
- Initial device value
- Comment
- Local device

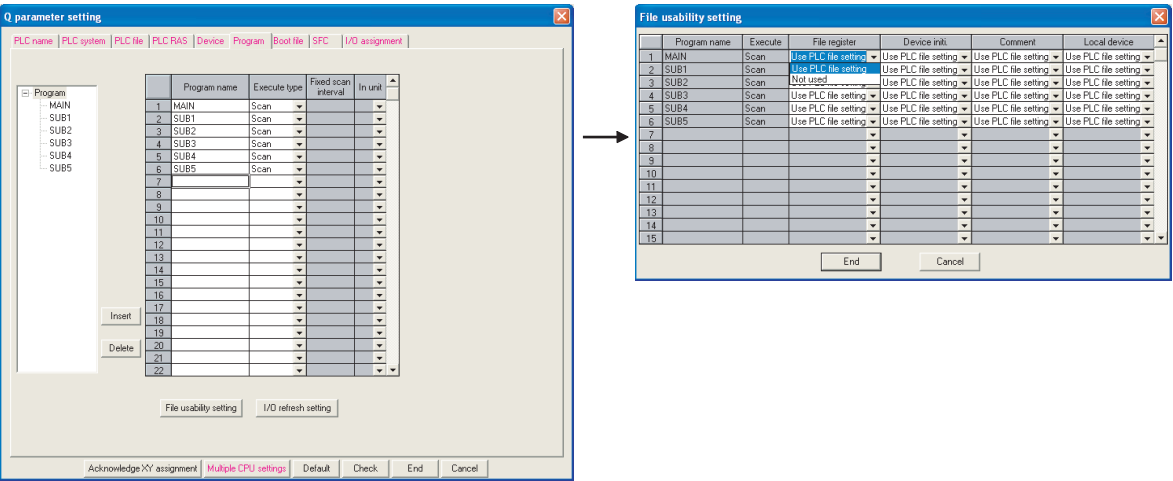


Figure App.27 Program tab

(b) Universal model QCPU

In the Universal model QCPU, file usability of the following files*1 cannot be set for each program on the screen opened by clicking the "File usability setting" button on the Program tab of the PLC parameter dialog box.

- File register
- Initial device value
- Comment

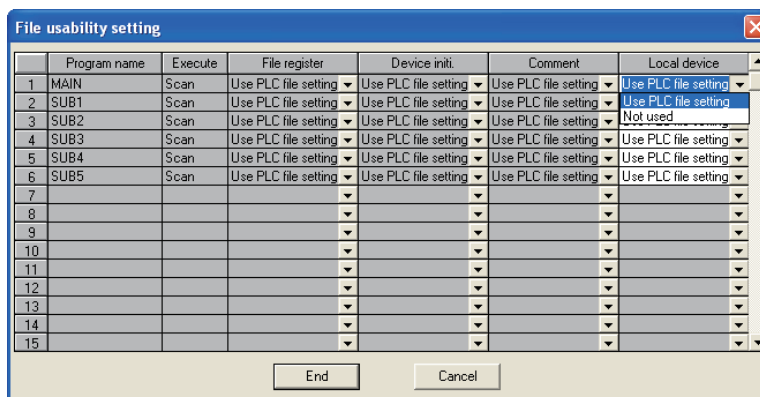


Figure App.28 File usability setting screen

*1: Even file usability of local device file cannot be set if the serial number (first five digits) of the Q02UCPU, Q03UDCPU, Q04UDHCPU, or Q06UDHCPU is "10011" or earlier.

If the local device is set to be used in the PLC file tab of the PLC parameter dialog box in the High Performance model QCPU, all the programs use the local device in the Universal model QCPU after replacement.

When the file usability setting is set in the High Performance model QCPU, change the setting as described below.

APPEN-
DIX

(2) Method of replacing High Performance model QCPU with Universal model QCPU

Replacement method varies depending on the settings in the PLC file tab of the PLC parameter dialog box.

TableApp.42 Replacement method

Setting in the PLC file tab	Setting in Universal model QCPU									
"Not used." is selected.	No change in parameter settings is required. Operation of the Universal model QCPU is the same regardless of the file usability setting in the High Performance model QCPU.									
"Use the same file name as the program." is selected.	When file usability is set to "Not used." in the High Performance model QCPU, delete the corresponding program file (file register, initial device value, or comment), which uses the same name as the program, from the target memory. The Universal model QCPU executes a program without using a program file if no program file that uses the same name as the program exists in the target memory. High Performance model QCPU PLC parameter setting PLC file setting File register setting Use the same file name as the program. (Target memory: Memory card (RAM)) Program setting <table><tr><td rowspan="3">File usability setting</td><th>Program name</th><th>File register</th></tr><tr><td>MAIN</td><td>Use PLC file setting</td></tr><tr><td>SUB1</td><td>Not used</td></tr><tr><td>SUB2</td><td>Not used</td></tr></table> SRAM card MAIN SUB1 SUB2 File register File register File register Programs do not use file registers 'SUB1' and 'SUB2' according to the "File usability setting". Universal model QCPU PLC parameter setting PLC file setting File register setting Use the same file name as the program. (Target memory: Memory card (RAM)) SRAM card MAIN SUB1 SUB2 File register File register File register File registers 'SUB1' and 'SUB2' shall be deleted so that the programs 'SUB1' and 'SUB2' do not use them.	File usability setting	Program name	File register	MAIN	Use PLC file setting	SUB1	Not used	SUB2	Not used
File usability setting	Program name		File register							
	MAIN		Use PLC file setting							
	SUB1	Not used								
SUB2	Not used									
"Use the following file." is selected.	No change in parameter settings is required. Operation of the Universal model QCPU is the same regardless of the file usability setting in the High Performance model QCPU.									

Appendix 3.4.6 Parameter-valid drive and boot file setting

(1) Differences between High Performance model QCPU and Universal model QCPU

(a) High Performance model QCPU

The parameter-valid drive is specified by the switches on the front panel of the High Performance model QCPU.

(b) Universal model QCPU

The Universal model QCPU automatically determines the parameter-valid drive, depending on the existence of parameters in the drive (program memory, memory card, or standard ROM).

Therefore, when replacing the High Performance model QCPU with the Universal model QCPU, changing the boot file setting for parameter and/or moving files to another drive may be required.

When replacing the module, change the setting as described below.

(2) Method of replacing High Performance model QCPU with Universal model QCPU

(a) When the parameter-valid drive is set to the standard ROM in the High Performance model QCPU

TableApp.43 When the parameter-valid drive is set to the standard ROM

Setting in High Performance model QCPU	Setting in Universal model QCPU										
Setting in the Boot file tab of the PLC parameter dialog box											
No boot file setting	Change the setting so that the Universal model QCPU can refer to the parameters in the standard ROM. • Changes in parameter settings are not required. • Delete parameters that exist in the program memory and memory card.*2										
Settings in the Boot file tab <table border="1"> <tr> <th>Type</th><th>Transfer from</th><th>Transfer to</th></tr> <tr> <td>Program</td><td>Standard ROM</td><td>Program memory</td></tr> </table> (No boot file setting for parameters)	Type	Transfer from	Transfer to	Program	Standard ROM	Program memory	Change the setting so that programs are stored in the program memory in the first place, instead of booting from the standard ROM. • Delete all settings for parameter in the Boot file tab of the PLC parameter dialog box. • Delete parameters that exist in the program memory and memory card.*2 • Move the programs with boot setting into the program memory from the standard ROM.*1				
Type	Transfer from	Transfer to									
Program	Standard ROM	Program memory									
Settings in the Boot file tab <table border="1"> <tr> <th>Type</th><th>Transfer from</th><th>Transfer to</th></tr> <tr> <td>Program</td><td>Standard ROM</td><td>Program memory</td></tr> <tr> <td>Parameter</td><td>Standard ROM</td><td>Program memory</td></tr> </table>	Type	Transfer from	Transfer to	Program	Standard ROM	Program memory	Parameter	Standard ROM	Program memory	Change the setting so that programs and parameters are stored in the program memory in the first place, instead of booting from the standard ROM. • Move the programs and parameters with boot setting into the program memory from the standard ROM.*1 • Delete all settings for parameter in the Boot file tab of the PLC parameter dialog box.	
Type	Transfer from	Transfer to									
Program	Standard ROM	Program memory									
Parameter	Standard ROM	Program memory									
Settings in the Boot file tab <table border="1"> <tr> <th>Type</th><th>Transfer from</th><th>Transfer to</th></tr> <tr> <td>Program</td><td>Memory card</td><td>Program memory</td></tr> </table> (No boot file setting for parameters)	Type	Transfer from	Transfer to	Program	Memory card	Program memory	Change the setting so that the Universal model QCPU can refer to the parameters in the memory card, and programs are booted from the memory card to the program memory. • Move the parameters in the standard ROM into the memory card. • Make setting so that programs are booted from the memory card to the program memory in the Boot file tab of the PLC parameter dialog box.*3				
Type	Transfer from	Transfer to									
Program	Memory card	Program memory									

(To the next page)

TableApp.43 When the parameter-valid drive is set to the standard ROM (Continued)

Setting in High Performance model QCPU			Setting in Universal model QCPU
Setting in the Boot file tab of the PLC parameter dialog box			
Settings in the Boot file tab			Change the setting so that the Universal model QCPU can refer to the parameters in the memory card, and programs and parameters are booted from the memory card to the program memory. • Move the parameters in the standard ROM into the memory card. • Make setting so that programs and parameters are booted from the memory card to the program memory in the Boot file tab of the PLC parameter dialog box.*3
Type	Transfer from	Transfer to	
Program	Memory card	Program memory	
Parameter	Memory card	Program memory	
Settings in the Boot file tab			Delete all settings for data other than programs and parameters in the boot file setting. Since these data can be used even not stored in the program memory, it is not necessary to transfer them to the program memory. Or, change the setting so that they are stored in the program memory in the first place. • Delete all settings for data other than programs and parameters in the Boot file tab of the PLC parameter dialog box. • Move the data other than programs and parameters into the program memory as needed.
Type	Transfer from	Transfer to	
(Data other than program and parameter)	Memory card	Program memory	
Settings in the Boot file tab			(Data other than program and parameter indicate initial device value, device comment, and label program.)
Type	Transfer from	Transfer to	
(Data other than program and parameter)	Standard ROM	Program memory	

*1: Since the Universal model QCPU holds the data in the program memory even when the battery voltage drops, the boot file setting is not necessary.

*2: The Universal model QCPU searches for parameters in order of in the program memory, in the memory card, and in the standard ROM. Then, the module uses the parameters found first.
If parameters exist in the program memory or the memory card, the Universal model QCPU cannot use the parameters in the standard ROM.

*3: The Universal model QCPU ignores the boot file setting for parameters in the standard ROM.

(b) When the parameter-valid drive is set to the memory card (RAM) or memory card (ROM) in the High Performance model QCPU

TableApp.44 When the parameter-valid drive is set to the memory card (RAM) or memory card (ROM)

Setting in High Performance model QCPU			Setting in Universal model QCPU									
Setting in the Boot file tab of the PLC parameter dialog box												
No boot file setting			Change the setting so that the Universal model QCPU can refer to the parameters in the memory card. • Changes in parameter settings are not required. • Delete parameters that exist in the program memory.*²									
Settings in the Boot file tab <table><tr><td>Type</td><td>Transfer from</td><td>Transfer to</td></tr><tr><td>Program</td><td>Memory card</td><td>Program memory</td></tr></table> (No boot file setting for parameters)			Type	Transfer from	Transfer to	Program	Memory card	Program memory	Change the setting so that the Universal model QCPU can refer to the parameters in the memory card. • Changes in parameter settings are not required. • Delete parameters that exist in the program memory.*²			
Type	Transfer from	Transfer to										
Program	Memory card	Program memory										
Settings in the Boot file tab <table><tr><td>Type</td><td>Transfer from</td><td>Transfer to</td></tr><tr><td>Program</td><td>Memory card</td><td>Program memory</td></tr><tr><td>Parameter</td><td>Memory card</td><td>Program memory</td></tr></table>			Type	Transfer from	Transfer to	Program	Memory card	Program memory	Parameter	Memory card	Program memory	No changes are required.
Type	Transfer from	Transfer to										
Program	Memory card	Program memory										
Parameter	Memory card	Program memory										
Settings in the Boot file tab <table><tr><td>Type</td><td>Transfer from</td><td>Transfer to</td></tr><tr><td>Program</td><td>Standard ROM</td><td>Program memory</td></tr></table> (No boot file setting for parameters)			Type	Transfer from	Transfer to	Program	Standard ROM	Program memory	Change the setting so that programs are stored in the program memory in the first place, instead of booting from the standard ROM. • Move the programs targeted for booting from the standard ROM into the program memory.*¹ • Delete all settings for program in the Boot file tab of the PLC parameter dialog box. • Delete parameters that exist in the program memory.*²			
Type	Transfer from	Transfer to										
Program	Standard ROM	Program memory										
Settings in the Boot file tab <table><tr><td>Type</td><td>Transfer from</td><td>Transfer to</td></tr><tr><td>Program</td><td>Standard ROM</td><td>Program memory</td></tr><tr><td>Parameter</td><td>Memory card</td><td>Program memory</td></tr></table>			Type	Transfer from	Transfer to	Program	Standard ROM	Program memory	Parameter	Memory card	Program memory	Change the setting so that programs are stored in the program memory in the first place, instead of booting from the standard ROM. • Move the programs targeted for booting from the standard ROM into the program memory.*¹ • Delete all settings for program in the Boot file tab of the PLC parameter dialog box.
Type	Transfer from	Transfer to										
Program	Standard ROM	Program memory										
Parameter	Memory card	Program memory										
Settings in the Boot file tab <table><tr><td>Type</td><td>Transfer from</td><td>Transfer to</td></tr><tr><td>(Data other than program and parameter)</td><td>Memory card</td><td>Program memory</td></tr></table>			Type	Transfer from	Transfer to	(Data other than program and parameter)	Memory card	Program memory	Delete all settings for data other than programs and parameters in the boot file setting. Since these data can be used even not stored in the program memory, it is not necessary to transfer them to the program memory. Or, change the setting so that they are stored in the program memory in the first place. • Delete all settings for data other than programs and parameters in the Boot file tab of the PLC parameter dialog box. • Move the data other than programs and parameters into the program memory as needed.			
Type	Transfer from	Transfer to										
(Data other than program and parameter)	Memory card	Program memory										
Settings in the Boot file tab <table><tr><td>Type</td><td>Transfer from</td><td>Transfer to</td></tr><tr><td>(Data other than program and parameter)</td><td>Standard ROM</td><td>Program memory</td></tr></table> (Data other than program and parameter indicate initial device value, device comment, and label program.)			Type	Transfer from	Transfer to	(Data other than program and parameter)	Standard ROM	Program memory				
Type	Transfer from	Transfer to										
(Data other than program and parameter)	Standard ROM	Program memory										

*1: Since the Universal model QCPU holds the data in the program memory even when the battery voltage drops, the boot file setting is not necessary.

*2: The Universal model QCPU searches for parameters in order of in the program memory, in the memory card, and in the standard ROM. Then, the module uses the parameters found first.
If parameters exist in the program memory or the memory card, the Universal model QCPU cannot use the parameters in the standard ROM.

Appendix 3.4.7 External input/output forced on/off function

(1) Differences between High Performance model QCPU and Universal model QCPU

(a) High Performance model QCPU

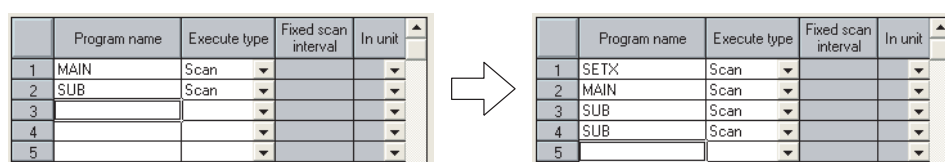
External input/output can be forcibly turned on/off on the screen opened by selecting [Online] → [Debug] → [Forced input output registration/cancellation] in GX Developer.

(b) Universal model QCPU

The external input/output forced on/off function is not supported in the Universal model QCPU.  NoteApp.1
External input/output can be forcibly turned on/off by using the replacement program described below.

(2) Method of replacing High Performance model QCPU with Universal model QCPU

As shown in Figure App.29, add programs, "SETX" and "SETY", in the Program tab of the PLC parameter dialog box.



	Program name	Execute type	Fixed scan interval	In unit
1	MAIN	Scan		
2	SUB	Scan		
3				
4				
5				

	Program name	Execute type	Fixed scan interval	In unit
1	SETX	Scan		
2	MAIN	Scan		
3	SUB	Scan		
4	SUB	Scan		
5				

Figure App.29 Modification in the Program tab of the PLC parameter setting

The following table shows the program setting of the "SETX" and "SETY".

TableApp.45 Program setting of "SETX" and "SETY"

Program name	Execution type	Position where program is added
SETX	Scan	Start of Program setting (No.1)
SETY	Scan	End of Program setting

NoteApp.1 **Universal**

When performing an external input/output forced on/off operation for the Q02UCPU, Q03UDCPU, Q04UDHCPU, Q06UDHCPU, Q13UDHCPU, or Q26UDHCPU, check the versions of the CPU module and GX Developer.

( Appendix 2)

Example) **Forcibly turning X40, X77, and X7A on, and X41 and Y7B off**

The programs, "SETX" and "SETY", turns on or off the X and Y devices, which have been registered for forced on/off using the external input/output forced on/off function, at each scan using the SET and RST instructions.

High Performance model QCPU

Forced input output registration/cancellation

Device

Set forced ON

Cancel it

Set forced OFF

No.	Device	ON/OFF	No.	Device	ON/OFF
➤ 1	X40	ON	17		
➤ 2	X41	OFF	18		
➤ 3	Y77	ON	19		
➤ 4	Y7A	ON	20		
➤ 5	Y7B	OFF	21		
6			22		
7			23		
8			24		
9			25		
10			26		
11			27		
12			28		
13			29		
14			30		
15			31		
16			32		

Update status

Clear all

Close

Universal model QCPU

• Program example of "SETX"

M100

CALL

P10

FEND

P10

SM400

SET

X40

RST

X41

RET

➤ (1)

➤ (2)

• Program example of "SETY"

M100

CALL

P11

FEND

P11

SM400

SET

Y77

SET

Y7A

RST

Y7B

RET

➤ (3)

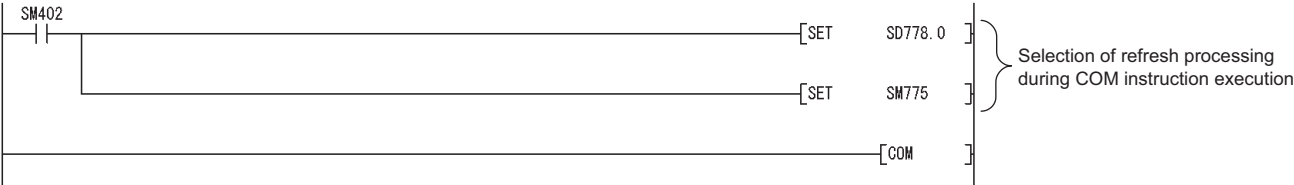
➤ (4)

➤ (5)

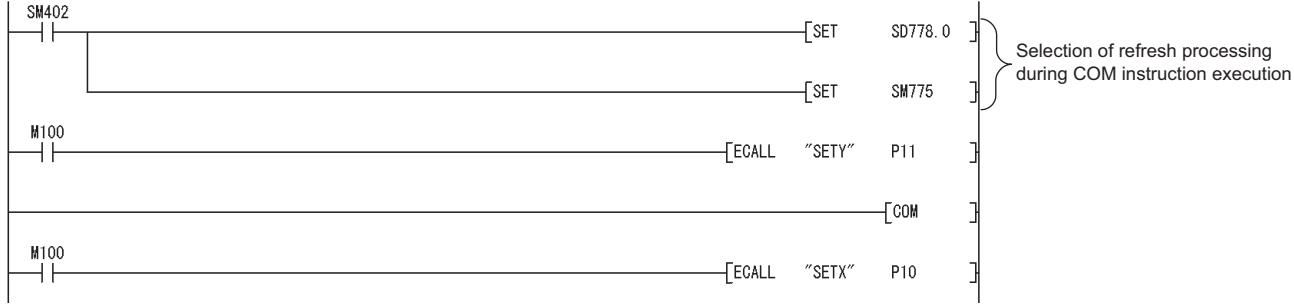
(3) Replacing the COM instruction

If the COM instruction is used, add subroutine calls for P10 and P11 before and after the COM instruction. (P10 and P11 are pointers shown in the program examples in (2).) When SM775 is on (Executes refresh set by SD778) and also the 0 bit of SD778 is off (Do not execute I/O refresh), replacement of the instruction is not necessary.

(a) Program before replacement



(b) Program after replacement

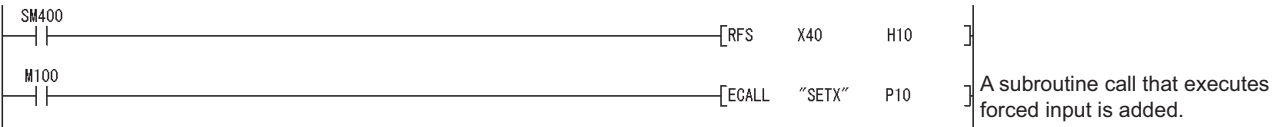


(4) Replacing the RFS instruction

If any I/O numbers targeted for forced on/off are included in the partial refresh range specified by the RFS instruction, add subroutine calls for P10 and P11 before and after the RFS instruction. (P10 and P11 are pointers shown in the program examples in (2).)
If no I/O number targeted for forced on/off is included, addition of subroutine calls for P10 and P11 is not necessary.

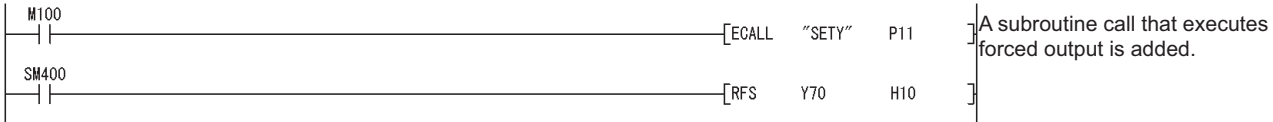
(a) When partial refresh for input (X) is executed by the RFS instruction

Add a subroutine call that executes forced input after the RFS instruction.



(b) When partial refresh for output (Y) is executed by the RFS instruction

Add a subroutine call that executes forced output before the RFS instruction.



(5) Restrictions

- Replacements described in (2) to (4) do not apply in the following cases.
- Input and output targeted for forced on/off are referred to or changed using the direct input device (DX) and direct output device (DY).
 - Input and output targeted for forced on/off are referred to or changed within an interrupt program.

Appendix 3.5 Special Relay and Special Register

The Universal model QCPU does not support the use of the special relay and special register described in Appendix 3.5.1 and Appendix 3.5.2.

Replace them using the method described in the table or delete the corresponding sections.

Appendix 3.5.1 Special relay list

TableApp.42 lists the special relay not supported in the Universal model QCPU and measures to be taken.

TableApp.46 Special relay not supported by the Universal model QCPU and measures

Number	Name/Description		Measures
SM80	CHK detection		The Universal model QCPU does not support the CHK instruction. For the replacing method of the CHK instruction, refer to Appendix 3.3.3.
SM91	Step transition monitoring timer start		The Universal model QCPU does not support the step transition monitoring timer function. For the replacing method of this function, refer to Appendix 3 "Restrictions on Basic Model QCPU, Universal Model QCPU, and LCPU and Alternative Methods" in the MELSEC-Q/L/ QnA Programming Manual (SFC).
SM92			
SM93			
SM94			
SM95			
SM96			
SM97			
SM98			
SM99			
SM250	Largest mounted I/O number read		Operation of SD250 is not necessary. The Universal model QCPU always stores the largest mounted I/O number in SD250. Delete the corresponding sections.
SM255	MELSECNET/H module 1 information	Indicates operative network or standby network	These are special relays for the simple dual-structured network function. Since the Universal model QCPU does not support this function, there is no application for these special relays. Delete the corresponding sections.
SM256		At refresh from link module to CPU, selects whether to read data from the link module.	
SM257		At refresh from CPU to link module, selects whether to write data to the link module.	
SM260	MELSECNET/H module 2 information	Indicates operative network or standby network	
SM261		At refresh from link module to CPU, selects whether to read data from the link module.	
SM262		At refresh from CPU to link module, selects whether to write data to the link module.	
SM265	MELSECNET/H module 3 information	Indicates operative network or standby network	
SM266		At refresh from link module to CPU, selects whether to read data from the link module.	
SM267		At refresh from CPU to link module, selects whether to write data to the link module.	
SM270	MELSECNET/H module 4 information	Indicates operative network or standby network	
SM271		At refresh from link module to CPU, selects whether to read data from the link module.	
SM272		At refresh from CPU to link module, selects whether to write data to the link module.	

(To the next page)

TableApp.46 Special relay not supported in the Universal model QCPU and measures (Continued)

Number	Name/Description	Measures
SM280	CC-Link error	Replace the relay with the I/O signals (Xn0, Xn1, and XnF) of the mounted CC-Link module.
SM315	Communication reserved time delay enable/disable flag	Set a service processing time value in the PLC system tab of the PLC parameter dialog box.
SM330	Operation mode for low-speed execution type program	The Universal model QCPU does not support low-speed execution type programs. Delete the corresponding sections.
SM331	Normal SFC program execution status	The Universal model QCPU supports only normal SFC programs. Delete SM331 and SM332, which are used as interlocks, or replace them with SM321.
SM332	Program execution management SFC program execution status	
SM390	Access execution flag	Modify the program that Module ready signal (Xn) is used as an interlock, according to sample programs described in the manual for each module.
SM404	ON for only 1 scan after RUN of low-speed execution type programs	The Universal model QCPU does not support low-speed execution type programs. Delete the corresponding sections or replace them with the special relays for scan execution type programs (SM402 and SM403).
SM405	Off for only 1 scan after RUN of low-speed execution type programs	
SM430	User timing clock No.5 (for low-speed execution type programs)	The Universal model QCPU does not support low-speed execution type programs. Delete the corresponding sections or replace them with the special relays for scan execution type programs (SM420 and SM424).
SM431	User timing clock No.6 (for low-speed execution type programs)	
SM432	User timing clock No.7 (for low-speed execution type programs)	
SM433	User timing clock No.8 (for low-speed execution type programs)	
SM434	User timing clock No.9 (for low-speed execution type programs)	
SM510	Low-speed execution type program executing flag	The Universal model QCPU does not support low-speed execution type programs. Delete the corresponding sections.
SM551	Module service interval time read	The Universal model QCPU does not support the service interval measurement function. Delete the corresponding sections.
SM580	Program to program I/O refresh	Perform I/O refresh at the start or end of each program with the RFS or COM instruction.
SM660	Boot operation	Move files with a boot setting (from the standard ROM or a memory card to the program memory) to the program memory.
SM672	Memory card file register access range flag	When outside the range of the file register in the memory card is accessed, the Universal model QCPU detects "OPERATION ERROR" (error code: 4101). Programming for detecting errors using this special relay is not necessary. Delete the corresponding sections.
SM710	CHK instruction priority flag	The Universal model QCPU does not support the CHK instruction. For the replacing method of the CHK instruction, refer to Appendix 3.3.
SM734	XCALL instruction execution condition specification	The Universal model QCPU executes the XCALL instruction on the rising edge of execution condition as well. There is no application for this special relay. Delete the corresponding sections.
SM735	SFC comment readout instruction in-execution flag	The Universal model QCPU does not support the following instructions: • SFC step comment readout instruction (S(P).SFCSOMR) • SFC transition condition comment readout instruction (S(P).SFCTOMR) Delete the corresponding sections.

(To the next page)

APPENDIX

Appendix 3 Method of Replacing Basic Model QCPU or High Performance Model QCPU with Universal Model QCPU
Appendix 3.5 Special Relay and Special Register

TableApp.46 Special relay not supported in the Universal model QCPU and measures (Continued)

Number	Name/Description	Measures
SM1780*1	Power supply off detection flag	The Universal model QCPU does not store redundant power supply system information in SM1780 to SM1783. Delete the corresponding sections. (SM1780 to SM1783 are always off.)
SM1781*1	Power supply failure detection flag	
SM1782*1	Momentary power failure detection flag for power supply 1	
SM1783*1	Momentary power failure detection flag for power supply 2	

*1: The special relay can be used if the serial number (first five digits) of the Universal model QCPU is "10042" or later.

Appendix 3.5.2 Special register list

TableApp.43 lists the special register not supported in the Universal model QCPU and measures to be taken

TableApp.47 Special register not supported in the Universal model QCPU and measures

Number	Name/Description	Measures
SD80	CHK number	The Universal model QCPU does not support the CHK instruction. For the replacing method of the CHK instruction, refer to Appendix 3.3.
SD90	Step transition monitoring timer setting value	The Universal model QCPU does not support the step transition monitoring timer function. For the replacing method of this function, refer to Appendix 3 "Restrictions on Basic Model QCPU, Universal Model QCPU, and LCPU and Alternative Methods" in the MELSEC-Q/L/QnA Programming Manual (SFC).
SD91		
SD92		
SD93		
SD94		
SD95		
SD96		
SD97		
SD98		
SD99		
SD130 to SD137	Fuse blown module	Replace SD130 to SD137 with SD1300 to SD1307.
SD150 to SD157	I/O module verify error	Replace SD150 to SD157 with SD1400 to SD1407.
SD245	No. of base slots (Mounting status)	Replace SD245 and SD246 with SD243 and SD244, respectively.
SD246		
SD280	CC-Link error	Replace these registers with the I/O signals (Xn0, Xn1, and XnF) of the mounted CC-Link module.
SD281		
SD315	Time reserved for communication processing	Service processing setting is available for the Universal model QCPU on the PLC system setting tab of the PLC parameter dialog box. Select "Specify service process time." for the service processing setting parameter and set the service processing time. Other setting methods can be selected as well.
SD394	CPU mounting information	• Check the type and model of other CPU modules mounted on the System monitor screen of GX Developer. • Check the mounting status of other CPU modules in SD396 to SD398.
SD430	Low-speed scan counter	The Universal model QCPU does not support low-speed execution type programs. Delete the corresponding sections or replace them with the special register for scan execution type programs (SD420).
SD510	Low-speed execution type program number	The Universal model QCPU does not support low-speed execution type programs. Delete the corresponding sections or replace them with the special register for scan execution type programs (SD500).
SD528	Current scan time for low-speed execution type programs	The Universal model QCPU does not support low-speed execution type programs. Delete the corresponding sections or replace them with the special register for scan execution type programs (SD520 and SD521).
SD529		
SD532	Minimum scan time for low-speed execution type programs	The Universal model QCPU does not support low-speed execution type programs. Delete the corresponding sections or replace them with the special registers for scan execution type programs (SD524 to SD527).
SD533		
SD534	Maximum scan time for low-speed execution type programs	
SD535		
SD544	Cumulative execution time for low-speed execution type programs	The Universal model QCPU does not support low-speed execution type programs. Delete the corresponding sections.
SD545		
SD546	Execution time for low-speed execution type programs	The Universal model QCPU does not support low-speed execution type programs. Delete the corresponding sections.
SD547		

(To the next page)

APPENDIX

Appendix 3 Method of Replacing Basic Model QCPU or High Performance Model QCPU with Universal Model QCPU
Appendix 3.5 Special Relay and Special Register

TableApp.47 Special register not supported in the Universal model QCPU and measures (Continued)

Number	Name/Description	Measures
SD550	Service interval measurement module	The Universal model QCPU does not support the service interval measurement function. Delete the corresponding sections.
SD551	Service interval time	
SD552		
SD720	Program No. specification for PLAODP instruction	The Universal model QCPU does not support the PLAODP instruction. Delete the corresponding sections.
SD1780 *1	Power supply off detection status	The Universal model QCPU does not store redundant power supply system information in SD1780 to SD1783. Delete the corresponding sections. (SD1780 to SD1783 are always off.)
SD1781 *1	Power supply failure detection status	
SD1782 *1	Momentary power failure detection counter for power supply 1	
SD1783 *1	Momentary power failure detection counter for power supply 1	

*1: The special register can be used if the serial number (first five digits) of the Universal model QCPU is "10042" or later.

Appendix 4 Device Point Assignment Sheet

TableApp.48 Device point assignment sheet

Device name	Symbol	Numeric notation	Number of device point ^{*2}		Restriction check		
			Points	Range	Size (words) ^{*3}	Points (bits) ^{*2}	
Input relay ^{*1}	X	Hexadecimal	8K (8192)	X0000 to X1FFF	/16	512	× 1 8192
Output relay ^{*1}	Y	Hexadecimal	8K (8192)	Y0000 to Y1FFF	/16	512	× 1 8192
Internal relay	M	Decimal	K ()	M0 to	/16		× 1
Latch relay	L	Decimal	K ()	L0 to	/16		× 1
Link relay	B	Hexadecimal	K ()	B0000 to	/16		× 1
Annunciator	F	Decimal	K ()	F0 to	/16		× 1
Link special relay	SB	Hexadecimal	K ()	SB0000 to	/16		× 1
Edge relay	V	Decimal	K ()	V0 to	/16		× 1
Step relay ^{*1}	S	Decimal	8K (8192)	S0 to S8191	/16	512	× 1 8192
Timer	T	Decimal	K ()	T0 to	× $\frac{18}{16}$		× 2
Retentive timer	ST	Decimal	K ()	ST0 to	× $\frac{18}{16}$		× 2
Counter	C	Decimal	K ()	C0 to	× $\frac{18}{16}$		× 2
Data register	D	Decimal	K ()	D0 to	× 1		
Link register	W	Hexadecimal	K ()	W0000 to	× 1		
Link special register	SW	Hexadecimal	K ()	SW0000 to	× 1		
Total					(29696 or less)		(65536 or less)

*1: The points are fixed for the system. (Cannot be changed)

The points for the step relay can be changed to 0 if the Universal model QCPU whose serial number (first five digits) is "10042" or later. For the Universal model QCPU whose serial number (first five digits) is "12052" or later, a step relay can be set in increments of 1k point and up to the following points. (Appendix 2)

- Q00UJCPU, Q00UCPU, Q01UCPU, and Q02UCPU: 8192 points
- Universal model QCPU other than the Q00UJCPU, Q00UCPU, Q01UCPU, and Q02UCPU: 16384 points

*2: Up to 32K points can be set for each device.

However, up to 60K points can be set for each device of the internal relay and link relay if the Universal model QCPU whose serial number (first five digits) is "10042" or later,

*3: Enter the values multiplied (or divided) by the number shown in the Size (words) column.

INDEX

[A]

Acknowledge XY assignment	8-16
Annunciator (F)	
Processing after annunciator off	9-14
Processing after annunciator on	9-12
ASCII code	2-43
Auto mode	4-1
Auto refresh setting	7-2

[B]

B (Link relay)	9-16
Backup data restoration function	6-161
Backup function to memory card	6-153
Base unit	A-23
Battery life-prolonging function	6-142
BCD (Binary-coded decimal)	2-38
BIN (Binary code)	2-35
BL (SFC block device)	9-80
Boot operation	5-23
Built-in Ethernet port of the CPU module	1-12
Built-in Ethernet port setting	8-19

[C]

C (Counter)	9-27
CC-Link	A-24
Changing clock data	6-15
Changing the program execution type	2-30
Character string	9-85
Checking the function version	1-14
Checking the serial number	1-14
Clearing	
Clearing data in the latch relay	9-10
Clearing local device data	9-96
Clearing the counter value	9-28
Clearing the error history data	6-108
Clearing the file register data	9-55
Clearing the retentive timer value	9-22
Remote latch clear	6-26
Clock function	6-14
Common pointer	9-76
Communications with the FROM and TO instructions	7-5
Constant scan	6-5
Counter (C)	
Counting	9-27
Resetting the counter	9-28
CPU module change function with memory card	6-150
Cyclic transmission area device (U3En\G□)	9-46

[D]

D (Data register)	9-31

Data clear processing	3-6
Data register (D)	9-31
Debug execution	
Operation procedure	6-95
Debug function from multiple GX Developers	6-93
Decimal constant (K)	9-83
Detail mode	4-1
Device list	9-1
Device point assignment sheet	App-72
Device setting	8-8
Direct access input	3-12
Direct access input (DX)	3-12
Direct access output	3-12
Direct access output (DY)	3-12
Direct mode	3-12
Double-precision floating-point data	2-41
Duty	9-30
DX (Direct access input)	3-12
DY (Direct access output)	3-12

[E]

E (Real number)	9-84
Edge relay (V)	9-15
END processing	3-2
Error	
Clearing errors	6-107
Error time output mode setting	6-30
Operating mode when there is an error	8-6
Error history	6-108
Ethernet	A-24
Execution time measurement	6-70
Executorial conditioned device test	6-49
Extended data register (D)	9-65
Extended link register (W)	9-65
Extension base unit	A-23
External input/output forced on/off	6-44

[F]

F (Annunciator)	9-11
FD (Function register)	9-36
File	5-30
File register	
Accesses available for the file register	9-56
Block switching method	9-62
Clearing the file register data	9-55
Serial number access method	9-62
File size	5-34
File usability setting	2-15
Fixed scan execution type program	2-26
Floating-point data	2-39
Forced on/off	6-44
Formatting	
Formatting the ATA card	5-15
Formatting the SRAM card	5-15

Memory card clear	5-15
Function devices (FX, FY, FD)	9-35
Function list	6-1
FX (Function input)	9-35
FY (Function output)	9-36

[G]

Global device	9-86
GOT	A-24
GX Configurator	7-2
GX Developer	A-24

[H]

H (Hexadecimal constant)	9-83
H/W error time PLC operation mode	6-31
H/W error time PLC operation mode setting	6-31
Hardware check	11-3
HEX (Hexadecimal)	2-37
Hexadecimal constant (H)	9-83
High-speed timer (T)	9-21

[I]

I (Interrupt pointer)	9-78
I/O assignment	4-8
I/O assignment on remote I/O stations	4-6
I/O No. specification device (U)	9-81
I/O number	4-1
I/O number assignment	4-1
I/O processing and response delay	3-8
I/O response time	6-28
Index register (Z)	
Restoring the index register data	9-50
Saving the index register data	9-50
Initial device value	6-138
Initial execution monitoring time	2-19
Initial execution type program	2-17
Initial processing	3-1
Initial setting	7-2,7-5
Input (X)	9-6
Intelligent function module	
Intelligent function module dedicated instructions	7-8
Intelligent function module device (U□\G□)	9-44
Intelligent function module switch setting	6-32
Interrupt form intelligent function module	6-122
Intelligent function module setting	8-3
Internal relay (M)	9-9
Internal system device	9-35
Internal user device	9-3
Interrupt	
Interrupt form intelligent function module	6-122
Interrupt module	6-28
Interrupt pointer (I)	9-78
Interrupt pointer setting	8-3
Interrupt program	2-6

Interrupt program monitor list	6-73
List of interrupt factors	9-79
Interrupt program/Fixed scan program setting	8-4

[J]

J (Network No. specification device)	9-80
J□\□ (Link direct device)	9-40

[K]

K (Decimal constant)	9-83

[L]

L (Latch relay)	9-10
Ladder mode	2-2,6-59
Latch	6-8
Latch clear	3-7
Latch clear operation disable range (Latch (2) start/end)	3-7
Latch clear operation enable range (Latch (1) start/end)	3-7
Latch data backup to standard ROM function	6-144
Latch function	6-8
Latch relay (L)	9-10
LED	
Methods for turning off the LEDs	6-119
Link direct device (J□\□)	9-40
Link register (W)	9-32
Link relay (B)	9-16
Link special register (SW)	9-34
Link special relay (SB)	9-18
List mode	2-2
Local device	9-87
Clearing local device data	9-96
Local device monitor	6-41
Local pointer	9-74
Low-speed timer (T)	9-21

[M]

M (Internal relay)	9-9
Macro instruction argument device (VD)	9-82
Main base unit	A-23
Main routine program	2-4
Maximum counting speed	9-30
MELSECNET/H	A-24
Memory capacities of files	5-34
Memory capacity	5-34
Memory card	5-14
Memory check function	6-143
Memory composition	5-1
Module service interval	6-131
Momentary power failure	3-5
Monitor	6-34
Monitor condition setting	6-35
Motion CPU	A-23

Multiple CPU high speed transmission area setting	8-18
Multiple CPU settings	8-17
Multiple CPU synchronous interrupt	9-78

[N]

N (Nesting)	9-71
Nesting	9-71
Network No. specification device (J)	9-80
Network parameters	8-21

[O]

Online change	6-59
Online change (files)	6-62
Online change (ladder mode)	6-59
Online change from multiple GX Developers	6-96
Output mode at STOP to RUN	6-12

[P]

P (Pointer)	9-72
Parameters	
Network parameters	8-21
PLC	8-2
Password	6-109
PAUSE status	3-3
PC CPU module	A-23
PC RAS (1) setting	8-6
PLC file setting	8-5
PLC name setting	8-2
PLC parameters	8-2
Pointer	
Common pointer	9-76
Interrupt pointer	9-78
Local pointer	9-74
Points occupied by empty slot	8-4
Procedure	
Procedure for boot run	5-24
Procedure for writing a program	5-19
Program cache memory	5-8
Program monitor list	6-70
Program setting	8-10
Protection	
Password registration	6-109
Remote password	6-111

[Q]

Q series power supply module	A-24
Q3□B	A-23
Q3□DB	A-23
Q3□RB	A-23
Q3□SB	A-23
Q5□B	A-23
Q6□B	A-23
Q6□RB	A-24
QI60	6-28

QnUD(H)CPU	A-23

[R]

R (File register)	9-54
Reading clock data	6-16
Redundant power extension base unit	A-23
Redundant power main base unit	A-23
Refresh mode	3-9
Remote latch clear	6-26
Remote operation	6-19
Remote password	6-111
Remote PAUSE	6-22
Remote RESET	6-24
Remote RUN/STOP	6-19
RUN status	3-3

[S]

S (Step relay)	9-19
Sampling trace	6-78
SB (Link special relay)	9-18
Scan execution type program	2-20
Scan time	2-18
Scan time accuracy	2-18
Scan time measurement	6-74
SD (Special register)	9-38
Self-diagnostic function	6-100
Serial communication function	6-123
Serial setting	8-15
Service processing	6-131
Service processing setting	6-131
Setting in GX Configurator	7-2
SFC block device (BL)	9-80
SFC setting	8-12
Single-precision floating-point data	2-39
Size (file capacity)	5-34
Slim type main base unit	A-23
Slim type power supply module	A-24
SM (Special relay)	9-38
Special register (SD)	9-38
Special relay (SM)	9-38
ST (Retentive timer)	9-22
Standard device register (Z)	9-49
Standard RAM	5-12
Standard ROM	5-11
Stand-by type program	2-21
Step relay (S)	9-19
STOP status	3-3
Subroutine program	2-5
SW (Link special register)	9-34
Switch setting	
Intelligent function module switch setting	6-32
System protection	6-109

[T]

T (Timer)	9-20
Timer (T)	

Accuracy	9-24
Processing	9-23
Timer limit setting	8-3
Transferring data to the program memory.	5-9

[U]

U (I/O No. specification device)	9-81
U3En\G□ (Cyclic transmission area device)	9-46
Units	5-36
Units of file sizes	5-36
U□\G□ (Intelligent function module device)	9-44
User setting system area.	5-5

[V]

V (Edge relay)	9-15
VD (Macro instruction argument device)	9-82

[W]

W (Link register)	9-32
Watchdog timer (WDT)	6-98
WDT (Watchdog timer)	6-98
WDT setting.	6-98
Write the program memory to ROM	5-18
Write to PLC (Flash ROM).	5-20
Writing	
Writing data to the memory card	5-16
Writing data to the program memory	5-7
Writing data to the standard RAM	5-13
Writing data to the standard ROM	5-11
Writing/reading device data to/from standard ROM	6-149

[X]

X (Input).	9-6

[Y]

Y (Output)	9-8

[Z]

Z (Index register)	9-47
Z (Standard device register)	9-49
ZR (Serial number access method of file register)	9-62

Memo

[illegible]

Warranty

Please confirm the following product warranty details before using this product.

1. Gratis Warranty Term and Gratis Warranty Range

If any faults or defects (hereinafter "Failure") found to be the responsibility of Mitsubishi occurs during use of the product within the gratis warranty term, the product shall be repaired at no cost via the sales representative or Mitsubishi Service Company.

However, if repairs are required onsite at domestic or overseas location, expenses to send an engineer will be solely at the customer's discretion. Mitsubishi shall not be held responsible for any re-commissioning, maintenance, or testing on-site that involves replacement of the failed module.

[Gratis Warranty Term]

The gratis warranty term of the product shall be for one year after the date of purchase or delivery to a designated place.

Note that after manufacture and shipment from Mitsubishi, the maximum distribution period shall be six (6) months, and the longest gratis warranty term after manufacturing shall be eighteen (18) months. The gratis warranty term of repair parts shall not exceed the gratis warranty term before repairs.

[Gratis Warranty Range]

- (1) The range shall be limited to normal use within the usage state, usage methods and usage environment, etc., which follow the conditions and precautions, etc., given in the instruction manual, user's manual and caution labels on the product.
- (2) Even within the gratis warranty term, repairs shall be charged for in the following cases.
 1. Failure occurring from inappropriate storage or handling, carelessness or negligence by the user. Failure caused by the user's hardware or software design.
 2. Failure caused by unapproved modifications, etc., to the product by the user.
 3. When the Mitsubishi product is assembled into a user's device, Failure that could have been avoided if functions or structures, judged as necessary in the legal safety measures the user's device is subject to or as necessary by industry standards, had been provided.
 4. Failure that could have been avoided if consumable parts (battery, backlight, fuse, etc.) designated in the instruction manual had been correctly serviced or replaced.
 5. Failure caused by external irresistible forces such as fires or abnormal voltages, and Failure caused by force majeure such as earthquakes, lightning, wind and water damage.
 6. Failure caused by reasons unpredictable by scientific technology standards at time of shipment from Mitsubishi.
 7. Any other failure found not to be the responsibility of Mitsubishi or that admitted not to be so by the user.

2. Onerous repair term after discontinuation of production

- (1) Mitsubishi shall accept onerous product repairs for seven (7) years after production of the product is discontinued.

Discontinuation of production shall be notified with Mitsubishi Technical Bulletins, etc.

- (2) Product supply (including repair parts) is not available after production is discontinued.

3. Overseas service

Overseas, repairs shall be accepted by Mitsubishi's local overseas FA Center. Note that the repair conditions at each FA Center may differ.

4. Exclusion of loss in opportunity and secondary loss from warranty liability

Regardless of the gratis warranty term, Mitsubishi shall not be liable for compensation of damages caused by any cause found not to be the responsibility of Mitsubishi, loss in opportunity, lost profits incurred to the user by Failures of Mitsubishi products, special damages and secondary damages whether foreseeable or not, compensation for accidents, and compensation for damages to products other than Mitsubishi products, replacement by the user, maintenance of on-site equipment, start-up test run and other tasks.

5. Changes in product specifications

The specifications given in the catalogs, manuals or technical documents are subject to change without prior notice.

Microsoft, Windows, Windows NT, Windows Vista are registered trademarks of Microsoft Corporation in the United States and other countries.

Pentium and Celeron are trademarks of Intel Corporation in the United States and other countries.

Ethernet is a trademark of Xerox Co., Ltd. in the United States.

CompactFlash is a trademark of SanDisk Corporation.

VxWorks, Tornado, WindPower, WindSh and WindView are registered trademarks of Wind River Systems, Inc.

Other company names and product names used in this document are trademarks or registered trademarks of respective owners.

QnUCPU User's Manual

Function Explanation, Program Fundamentals

MODEL	QNUCPU-U-KP-E
MODEL CODE	13JZ27
SH(NA)-080807ENG-F(1008)MEE	



HEAD OFFICE : TOKYO BUILDING, 2-7-3 MARUNOUCHI, CHIYODA-KU, TOKYO 100-8310, JAPAN
NAGOYA WORKS : 1-14, YADA-MINAMI 5-CHOME, HIGASHI-KU, NAGOYA, JAPAN

When exported from Japan, this manual does not require application to the
Ministry of Economy, Trade and Industry for service transaction permission.

Specifications subject to change without notice.

HEADQUARTERS

MITSUBISHI ELECTRIC EUROPE B.V. EUROPE
German Branch
Gothaer Straße 8
D-40880 Ratingen
Phone: +49 (0)2102 / 486-0
Fax: +49 (0)2102 / 486-1120

MITSUBISHI ELECTRIC EUROPE B.V. CZECH REPUBLIC
Czech Branch
Avenir Business Park, Radlická 714/113a
CZ-158 00 Praha 5
Phone: +420 - 251 551 470
Fax: +420 - 251-551-471

MITSUBISHI ELECTRIC EUROPE B.V. FRANCE
French Branch
25, Boulevard des Bouvets
F-92741 Nanterre Cedex
Phone: +33 (0)1 / 55 68 55 68
Fax: +33 (0)1 / 55 68 57 57

MITSUBISHI ELECTRIC EUROPE B.V. IRELAND
Irish Branch
Westgate Business Park, Ballymount
IRL-Dublin 24
Phone: +353 (0)1 4198800
Fax: +353 (0)1 4198890

MITSUBISHI ELECTRIC EUROPE B.V. ITALY
Italian Branch
Viale Colleoni 7
I-20041 Agrate Brianza (MB)
Phone: +39 039 / 60 53 1
Fax: +39 039 / 60 53 312

MITSUBISHI ELECTRIC EUROPE B.V. POLAND
Poland Branch
Krakowska 50
PL-32-083 Balice
Phone: +48 (0)12 / 630 47 00
Fax: +48 (0)12 / 630 47 01

MITSUBISHI ELECTRIC EUROPE B.V. SPAIN
Spanish Branch
Carretera de Rubí 76-80
E-08190 Sant Cugat del Vallés (Barcelona)
Phone: 902 131121 // +34 935653131
Fax: +34 935891579

MITSUBISHI ELECTRIC EUROPE B.V. UK
UK Branch
Travellers Lane
UK-Hatfield, Herts. AL10 8XB
Phone: +44 (0)1707 / 27 61 00
Fax: +44 (0)1707 / 27 86 95

MITSUBISHI ELECTRIC CORPORATION JAPAN
Office Tower "Z" 14 F
8-12, 1 chome, Harumi Chuo-Ku
Tokyo 104-6212
Phone: +81 3 622 160 60
Fax: +81 3 622 160 75

MITSUBISHI ELECTRIC AUTOMATION, Inc. USA
500 Corporate Woods Parkway
Vernon Hills, IL 60061
Phone: +1 847 478 21 00
Fax: +1 847 478 22 53

EUROPEAN REPRESENTATIVES

AUSTRIA
GEVA
Wiener Straße 89
AT-2500 Baden
Phone: +43 (0)2252 / 85 55 20
Fax: +43 (0)2252 / 488 60

BELARUS
TEHNIKON
Oktyabrskaya 16/5, Off. 703-711
BY-220030 Minsk
Phone: +375 (0)17 / 210 46 26
Fax: +375 (0)17 / 210 46 26

BELGIUM
ESCO DRIVES & AUTOMATION
Culliganlaan 3
BE-1831 Diegem
Phone: +32 (0)2 / 717 64 30
Fax: +32 (0)2 / 717 64 31

BELGIUM
Koning & Hartman b.v.
Woluwelaan 31
BE-1800 Vilvoorde
Phone: +32 (0)2 / 257 02 40
Fax: +32 (0)2 / 257 02 49

BOSNIA AND HERZEGOVINA
INEA BH d.o.o.
Aleja Lipa 56
BA-71000 Sarajevo
Phone: +387 (0)33 / 921 164
Fax: +387 (0)33 / 524 539

BULGARIA
AKHNATON
4 Andrej Ljapchev Blvd. Pb 21
BG-1756 Sofia
Phone: +359 (0)2 / 817 6004
Fax: +359 (0)2 / 97 44 06 1

CROATIA
INEA CR d.o.o.
Losinjska 4 a
HR-10000 Zagreb
Phone: +385 (0)1 / 36 940 -01/-02/-03
Fax: +385 (0)1 / 36 940 -03

CZECH REPUBLIC
AutoCont C.S. s.r.o.
Technologická 374/6
CZ-708 00 Ostrava-Pustkovec
Phone: +420 595 691 150
Fax: +420 595 691 199

CZECH REPUBLIC
B:ELECTRIC, s.r.o.
Mladoboleslavská 812
CZ-197 00 Praha 19 - Kbely
Phone: +420 286 850 848, +420 724 317 975
Fax: +420 286 850 850

DENMARK
Beijer Electronics A/S
Lykkegårdsvej 17
DK-4000 Roskilde
Phone: +45 (0)46 / 75 76 66
Fax: +45 (0)46 / 75 56 26

ESTONIA
Beijer Electronics Eesti OÜ
Pärnu mnt.160i
EE-11317 Tallinn
Phone: +372 (0)6 / 51 81 40
Fax: +372 (0)6 / 51 81 49

FINLAND
Beijer Electronics OY
Peltoie 37
FIN-28400 Ulvila
Phone: +358 (0)207 / 463 540
Fax: +358 (0)207 / 463 541

GREECE
UTECO A.B.E.E.
5, Mavrogenous Str.
GR-18542 Piraeus
Phone: +30 211 / 1206 900
Fax: +30 211 / 1206 999

HUNGARY
MELTRADE Kft.
Fertő utca 14.
HU-1107 Budapest
Phone: +36 (0)1 / 431-9726
Fax: +36 (0)1 / 431-9727

LATVIA
Beijer Electronics SIA
Ritausmas iela 23
LV-1058 Riga
Phone: +371 (0)784 / 2280
Fax: +371 (0)784 / 2281

LITHUANIA
Beijer Electronics UAB
Savanoriu Pr. 187
LT-02300 Vilnius
Phone: +370 (0)5 / 232 3101
Fax: +370 (0)5 / 232 2980

EUROPEAN REPRESENTATIVES

MALTA
ALFATRADE Ltd.
99, Paola Hill
Malta- Paola PLA 1702
Phone: +356 (0)21 / 697 816
Fax: +356 (0)21 / 697 817

MOLDOVA
INTEHSIS srl
bld. Traian 23/1
MD-2060 Kishinev
Phone: +373 (0)22 / 66 4242
Fax: +373 (0)22 / 66 4280

NETHERLANDS
HIFLEX AUTOM. TECHNIEK B.V.
Wolweverstraat 22
NL-2984 CD Ridderkerk
Phone: +31 (0)180 - 46 60 04
Fax: +31 (0)180 - 44 23 55

NETHERLANDS
Koning & Hartman b.v.
Haarlerbergweg 21-23
NL-1101 CH Amsterdam
Phone: +31 (0)20 / 587 76 00
Fax: +31 (0)20 / 587 76 05

NORWAY
Beijer Electronics AS
Postboks 487
NO-3002 Drammen
Phone: +47 (0)32 / 24 30 00
Fax: +47 (0)32 / 84 85 77

ROMANIA
Sirius Trading & Services srl
Aleea Lacul Morii Nr. 3
RO-060841 Bucuresti, Sector 6
Phone: +40 (0)21 / 430 40 06
Fax: +40 (0)21 / 430 40 02

SERBIA
Craft Con. & Engineering d.o.o.
Bulevar Svetog Cara Konstantina 80-86
SER-18106 Nis
Phone: +381 (0)18 / 292-24-4/5
Fax: +381 (0)18 / 292-24-4/5

SERBIA
INEA SR d.o.o.
Izletnicka 10
SER-113000 Smederevo
Phone: +381 (0)26 / 617 163
Fax: +381 (0)26 / 617 163

SLOVAKIA
AutoCont Control s.r.o.
Radlinského 47
SK-02601 Dolny Kubin
Phone: +421 (0)43 / 5868210
Fax: +421 (0)43 / 5868210

SLOVAKIA
CS MTrade Slovensko, s.r.o.
Vajanskeho 58
SK-92101 Piestany
Phone: +421 (0)33 / 7742 760
Fax: +421 (0)33 / 7735 144

SLOVENIA
INEA d.o.o.
Stegne 11
SI-1000 Ljubljana
Phone: +386 (0)1 / 513 8100
Fax: +386 (0)1 / 513 8170

SWEDEN
Beijer Electronics AB
Box 426
SE-20124 Malmö
Phone: +46 (0)40 / 35 86 00
Fax: +46 (0)40 / 93 23 01

SWITZERLAND
Omni Ray AG
Im Schörl 5
CH-8600 Dübendorf
Phone: +41 (0)44 / 802 28 80
Fax: +41 (0)44 / 802 28 28

TURKEY
GTS
Bayraktar Bulvarı Nutuk Sok. No:5
TR-34775 Yukarı Dudullu-Ümraniye-İSTANBUL
Phone: +90 (0)216 526 39 90
Fax: +90 (0)216 526 3995

UKRAINE
CSC Automation Ltd.
4-B, M. Raskovoyi St.
UA-02660 Kiev
Phone: +380 (0)44 / 494 33 55
Fax: +380 (0)44 / 494-33-66

EURASIAN REPRESENTATIVES

KAZAKHSTAN
Kazpromautomatiks Ltd.
Mustafina Str. 7/2
KAZ-470046 Karaganda
Phone: +7 7212 / 50 11 50
Fax: +7 7212 / 50 11 50

MIDDLE EAST REPRESENTATIVES

ISRAEL
ILAN & GAVISH Ltd.
24 Shenkar Str., Kiryat Arie
IL-49001 Petah-Tiqva
Phone: +972 (0)3 / 922 18 24
Fax: +972 (0)3 / 924 0761

ISRAEL
TEXEL ELECTRONICS Ltd.
2 Ha'umanut, P.O.B. 6272
IL-42160 Netanya
Phone: +972 (0)9 / 863 39 80
Fax: +972 (0)9 / 885 24 30

LEBANON
CEG INTERNATIONAL
Cebaco Center/Block A Autostrade DORA
Lebanon - Beirut
Phone: +961 (0)1 / 240 430
Fax: +961 (0)1 / 240 438

AFRICAN REPRESENTATIVE

SOUTH AFRICA
CBI Ltd.
Private Bag 2016
ZA-1600 Isando
Phone: +27 (0)11 / 977 0770
Fax: +27 (0)11 / 977 0761