

MITSUBISHI

Mitsubishi Programmable Controller

MELSEC **Q** series

QnUCPU User's Manual

Communication via Built-in Ethernet Port

QSERIES

-Q03UDVCPU	-Q06UDEHCPU	-Q26UDVCPU
-Q03UDECPU	-Q10UDEHCPU	-Q26UDEHCPU
-Q04UDVCPU	-Q13UDVCPU	-Q50UDEHCPU
-Q04UDEHCPU	-Q13UDEHCPU	-Q100UDEHCPU
-Q06UDVCPU	-Q20UDEHCPU	

MODEL

● 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".



WARNING

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



CAUTION

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.

	Q/L series module	AnS/A series module
Overcurrent or overvoltage protection of the power supply module is activated.	All outputs are turned off	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 are turned off

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 Appendix 8 General Safety Requirements 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, when a remote programmable controller is controlled by an external device, immediate action cannot be taken if a problem occurs in 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.

[Design Precautions]

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 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.
- When using an SD memory card, fully insert it into the SD memory card slot.
Check that it is inserted completely.
Poor contact may cause malfunction.
- Securely insert an extended SRAM cassette into the cassette connector of a CPU module.
After insertion, close the cassette cover to prevent the cassette from coming off.
Failure to do so may cause malfunction.
- Shut off the external power supply (all phases) used in the system before mounting or removing a 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 parts and electronic components of the module, memory card, SD memory card, or extended SRAM cassette.
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 (all phases) used in the system before installation and 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.

[Wiring Precautions]

CAUTION

- Individually ground the FG and LG terminals of the programmable controller with a ground resistance of 100Ω or less. 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.
- Securely connect the connector to the module. Failure to do so may cause malfunction.
- 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.
- Do not install the control lines or communication cables together with the main circuit lines or power cables. Keep a distance of 100mm or more between them. Failure to do so may result in malfunction due to noise.
- Place the cables in a duct or clamp them. If not, dangling cable may swing or inadvertently be pulled, resulting in damage to the module or cables or malfunction due to poor contact.
- Check the interface type and correctly connect the cable. Incorrect wiring (connecting the cable to an incorrect interface) may cause failure of the module and external device.
- 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.
- 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.

[Wiring Precautions]

CAUTION

- When disconnecting the cable from the module, do not pull the cable by the cable part.
For the cable with connector, hold the connector part of the cable.
For the cable connected to the terminal block, loosen the terminal screw.
Pulling the cable connected to the module may result in malfunction or damage to the module or cable.
- 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 or malfunction.
- Correctly connect the battery connector.
Do not charge, disassemble, heat, short-circuit, solder, or throw the battery into the fire. Also, do not expose it to liquid or strong shock.
Doing so will cause the battery to produce heat, explode, ignite, or leak, resulting in injury and fire.
- Shut off the external power supply (all phases) used in the system before cleaning the module or retightening the terminal screws, connector screws, or module fixing screws.
Failure to do so may result in electric shock or cause the module to fail 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.

[Startup and Maintenance Precautions]

CAUTION

- Shut off the external power supply (all phases) used in the system before mounting or removing a 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, and do not insert/remove the extended SRAM cassette to/from the CPU module more than 50 times (IEC 61131-2 compliant) respectively.
Exceeding the limit of 50 times may cause malfunction.
- After the first use of the product, do not insert/remove the SD memory card to/from the CPU module more than 500 times. Exceeding the limit 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.

INTRODUCTION

This manual describes the function of the Universal model QCPU using Ethernet communication.

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.


When applying the program examples introduced in this manual to the actual system, ensure the applicability and confirm that it will not cause system control problems.

■ Relevant CPU module

CPU module	Model
Built-in Ethernet port QCPU	Q03UDVCP, Q03UDECP, Q04UDVCP, Q04UDEHCP, Q06UDVCP, Q06UDEHCP, Q10UDEHCP, Q13UDVCP, Q13UDEHCP, Q20UDEHCP, Q26UDVCP, Q26UDEHCP, Q50UDEHCP, Q100UDEHCP

Remark

This manual does not describe any functions other than the functions of CPU module using Ethernet communication. For details of functions other than the functions of CPU module using Ethernet communication, refer to the following manual.

 QnUCPU User's Manual (Function Explanation, Program Fundamentals)

Memo

CONTENTS

SAFETY PRECAUTIONS	1
CONDITIONS OF USE FOR THE PRODUCT	7
INTRODUCTION	8
MANUALS	13
MANUAL PAGE ORGANIZATION	15
TERMS	19
<hr/>	
CHAPTER 1 OVERVIEW	20
<hr/>	
1.1 Features	20
<hr/>	
CHAPTER 2 COMMUNICATION SPECIFICATIONS	22
<hr/>	
CHAPTER 3 CONNECTION OF PROGRAMMING TOOLS AND GOT	24
<hr/>	
3.1 Setting for the CPU Module	25
3.2 Setting for the Programming Tool	27
3.3 Searching CPU Modules on the Network	29
3.4 Communication via Routers	30
3.5 Precautions	31
<hr/>	
CHAPTER 4 DIRECT CONNECTION TO PROGRAMMING TOOL (SIMPLE CONNECTION)	33
<hr/>	
4.1 Setting Method	34
4.2 Precautions	35
<hr/>	
CHAPTER 5 MC PROTOCOL COMMUNICATION	36
<hr/>	
5.1 Setting Method	37
5.2 MC Protocol Commands	39
5.2.1 Command list	39
5.2.2 Available devices	41
5.3 Precautions	44
5.4 Error codes, end codes, and abnormal codes in MC protocol communication	46
<hr/>	
CHAPTER 6 DATA COMMUNICATIONS USING THE PREDEFINED PROTOCOL	49
<hr/>	
6.1 Specifications	50
6.2 Setting Method	51
6.3 Setting Items of Predefined Protocol Support Function	56
6.3.1 Communication type	56
6.3.2 Packet elements set for "Packet Setting"	56
6.4 Predefined Protocol Function Instructions	57
6.4.1 Executing the predefined protocol (SP.ECPRTCL)	57
6.5 Execution Conditions of Predefined Protocol Function	58
6.6 Operation Image and Data Structure of the Predefined Protocol Function	58
6.7 Precautions	59

CHAPTER 7 SOCKET COMMUNICATION FUNCTION 60

7.1	Communication Using TCP	62
7.2	Communication Using UDP	71
7.3	Precautions for the Socket Communication Function	77
7.4	Socket Communication Function Instructions	79
7.4.1	Establishing a connection (SP.SOCOPEN)	80
7.4.2	Disconnecting a connection (SP.SOCCLOSE)	84
7.4.3	Reading out received data in the END processing (SP.SOCRCV)	87
7.4.4	Reading out received data during instruction execution (S.SOCRCVS)	91
7.4.5	Sending data (SP.SOCSND)	94
7.4.6	Reading out connection information (SP.SOCCINF)	98
7.4.7	Changing the target of a connection (UDP/IP) (SP.SOCCSET)	101
7.4.8	Changing the receive mode of a connection (SP.SOCRMODE)	103
7.4.9	Socket communication receive data read (S(P).SOCRDATA)	107

CHAPTER 8 TIME SETTING FUNCTION (SNTP CLIENT) 109

8.1	Setting Method	110
8.2	Precautions	111

CHAPTER 9 FILE TRANSFER FUNCTION (FTP) 112

9.1	Setting for FTP Communication	113
9.2	Files Transferable Using FTP	118
9.3	Files That Can Be Deleted Using FTP	119
9.4	FTP Commands	120
9.4.1	List of FTP commands	120
9.4.2	How to specify an FTP command	122
9.4.3	Details of FTP commands	123
9.5	Precautions	132

CHAPTER 10 REMOTE PASSWORD 134

10.1	Communication Using Remote Password	135
10.2	Remote Password Setting	136
10.3	Precautions	138
10.4	Detection of Unauthorized Access and Actions	139

CHAPTER 11 IP ADDRESS CHANGE FUNCTION 140

11.1	IP Address of the Built-in Ethernet Port	141
11.2	How to Use the Function	142
11.2.1	Write operation	142
11.2.2	Clear operation	145
11.3	Checking the IP Address	148
11.4	Precautions	148

CHAPTER 12 IP PACKET TRANSFER FUNCTION	150
APPENDICES	152
Appendix 1 Operation Processing Time for Each Instruction.	152
Appendix 2 Port Numbers Used by Built-in Ethernet Port QCPU	154
Appendix 3 Added and Changed Functions.	155
Appendix 4 Specifications Comparison with Ethernet Module	156
INDEX	162
INSTRUCTION INDEX	164
REVISIONS	165
WARRANTY	167

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. The manuals related to this product are listed below. Please place an order as needed.

●: Basic manual, ○: Other CPU module manuals/Use them to utilize functions.

(1) CPU module user's manual

Manual name <manual number (model code)>	Description	Manual type
QCPU User's Manual (Hardware Design, Maintenance and Inspection) <SH-080483ENG (13JR73)>	Specifications of the CPU modules, power supply modules, base units, extension cables, memory cards, SD memory cards, extended SRAM cassettes, and batteries, information on how to establish a system, maintenance and inspection, and troubleshooting	●
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 building multiple CPU systems (system configurations, I/O numbers, communications between CPU modules, and communications with I/O modules and intelligent function modules)	○
QnUDVCPULCPU User's Manual (Data Logging Function) <SH-080893ENG (13JZ39)>	Detailed description of the data logging function of the CPU module	○

(2) Programming manual

Manual name <manual number (model code)>	Description	Manual type
MELSEC-Q/L Programming Manual (Common Instruction) <SH-080809ENG, 13JW10>	Usage of common instructions, such as sequence instructions, basic instructions, and application instructions	●
MELSEC-Q/L/QnA Programming Manual (SFC) <SH-080041, 13JF60>	System configuration, specifications, functions, programming, and error codes for SFC (MELSAP3) programs	○
MELSEC-Q/L Programming Manual (MELSAP-L) <SH-080076, 13JF61>	System configuration, specifications, functions, programming, and error codes for SFC (MELSAP-L) programs	○
MELSEC-Q/L Programming Manual (Structured Text) <SH-080366E, 13JF68>	System configuration and programming using structured text language	○
MELSEC-Q/L/QnA Programming Manual (PID Control Instructions) <SH-080040, 13JF59>	Dedicated instructions for PID control	○
MELSEC-Q Programming/Structured Programming Manual (Process Control Instructions) <SH-080316E, 13JF67>	Dedicated instructions for process control	○

(3) Operating manual

Manual name <manual number (model code)>	Description	Manual type
GX Works2 Version 1 Operating Manual (Common) <SH-080779ENG, 13JU63>	System configuration, parameter settings, and online operations of GX Works2, which are common to Simple projects and Structured projects	●
GX Developer Version 8 Operating Manual <SH-080373E, 13JU41>	Operating methods of GX Developer, such as programming, printing, monitoring, and debugging	○

MANUAL PAGE ORGANIZATION

In this manual, pages are organized and the symbols are used as shown below. The following page illustration is for explanation purpose only, and is different from the actual pages.

"" is used for screen names and items.

1. shows operating procedures.

☞ shows mouse operations.*1

[] is used for items in the menu bar and the project window.

Ex. shows setting or operating examples.

📖 shows reference manuals.

📄 shows reference pages.

The chapter of the current page is shown.

The section of the current page is shown.

Point shows notes that requires attention.

Remark shows useful information.

*1 The mouse operation example is provided below. (For GX Works2)


Menu bar

Ex. ☞ [Online] ☞ [Write to PLC...]
Select [Online] on the menu bar, and then select [Write to PLC...].

A window selected in the view selection area is displayed.

Ex. ☞ Project window ☞ [Parameter]
☞ [PLC Parameter]
Select [Project] from the view selection area to open the Project window.
In the Project window, expand [Parameter] and select [PLC Parameter].

View selection area

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

Pages describing instructions are organized as shown below.

The following page illustrations are for explanation purpose only, and are different from the actual pages.

CHAPTER 6 SOCKET COMMUNICATION FUNCTION

6.4.2 Disconnecting a connection (SPSOCCLOSE)

Execution condition of the instruction

Structure of the instruction in the ladder mode

Shows the devices applicable to the instruction

Setting data	Internal device		R, ZR	JCIQ		UIC/QC	Zn	Consta st K, H	Others
	Bit	Word		Bit	Word				
①	—	○	○	—	—	—	—	○	—
②	—	△ ¹⁾	△ ¹⁾	—	—	—	—	—	—
③	△ ¹⁾	—	△ ¹⁾	—	—	—	—	—	—

*1. File registers set for each local device or program cannot be used.

Setting side

User : Device value is set by the user.
System: Device value is set by the CPU module.

Descriptions of setting data and data type

Setting data	Description	Set by	Data type
LD	Dummy	—	Character string
①	Connection number (Setting range: 1 to 16)	User	BIN 16-bit
②	Start number of the device from which control data are stored	System	Device name
③	Start number of the device which turns on for one scan upon completion of the instruction ③+1 also turns on when failed.	System	Bit

Descriptions of control data (if any)

Device	Item	Description	Setting range	Set by
③+0	System area	—	—	—
③+1	Completion status	Completion status is stored 0000: Completed Other than 0000: Failed (Error code)	—	System

63

Detailed descriptions of the instruction

(3) Function

This instruction closes a connection specified in ①. (Disconnection of a connection)

The result of the SPSOCCLOSE instruction can be checked with the completion device, ③ + 0 and ③ + 1.

- Completion device ③ + 0
 - Turns on in the END processing of a scan after completion of the SPSOCCLOSE instruction, and turns off in the next END processing.
- Completion device ③ + 1
 - Turns on or off according to the result of the SPSOCCLOSE instruction.

State	Description
When completed	Remains off.
When failed	Turns on in the END processing of a scan after completion of the SPSOCCLOSE instruction, and turns off in the next END processing.

Conditions for the error and error codes

For the errors not described in this manual, refer to the following.

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

(4) Error

A detection of an operation error turns on the Error flag (SM0) and a corresponding error code is stored in SDO when:

- The connection number specified for ① is other than 1 to 16. (Error code: 4101)
- The device numbers specified for ② and ③ exceed the device point range. (Error code: 4101)
- An invalid device is specified. (Error code: 4004)

(5) Program example

When M2000 is turned on or when the connected device disconnects connection No. 1, connection No. 1 is disconnected by the following program.

- Device used

Device number	Application
SD1282	Open completion signal
SD1284	Open request signal
D200	SPSOCCLOSE instruction control data
M200	SPSOCCLOSE instruction completion device





• Program

```

LD M2000 SD1282
LD M2000 SD1284
M2000 SPSOCCLOSE "01" K1 D200 M200
M200 M200 M210
M210 M210 M202
M202 M202 M203
M203 M203 M210
M210 M210 M210
END
  
```

64


- Instructions can be executed under the following conditions.

Execution condition	Any time	During on	On the falling edge	During off	On the falling edge
Symbol	No symbol				

- The following devices can be used.

Setting data	Internal device (system, user)		File register	Link direct device J□\□ ^{*4}		Intelligent function module U□\G□	Index register Zn	Constant ^{*5}	Others ^{*5}
	Bit	Word		Bit	Word				
Applicable device ^{*1}	X, Y, M, L, SM, F, B, SB, FX, FY ^{*2}	T, ST, C, D, W, SD, SW, FD, @□ ^{*2*3}	R, ZR	J□\X, J□\Y, J□\B, J□\SB	J□\W, J□\SW	U□\G□	Z	K, H, E, \$	P, I, J, U, DX, DY, N, BL, TR, BL\S, V

*1 For details on each device, refer to the following.

 QnUCPU User's Manual (Function Explanation, Program Fundamentals)

*2 FX and FY can be used for bit data only, and FD for word data only.

*3 When T, ST, and C are used for instructions other than the following instructions, they can be used for word data only (but cannot be used for bit data).

[Instructions available for bit data] LD, LDI, AND, ANI, OR, ORI, LDP, LDF, ANDP, ANDF, ORP, ORF, OUT, and RST

*4 Available for CC-Link IE, MELSECNET/H, and MELSECNET/10

*5 In the "Constant" and "Others" columns, a device(s) that can be set for each instruction is shown.

- The following data types can be used.

Data type	Description
Bit	Bit data or the start number of bit data
BIN 16-bit	16-bit binary data or the start number of word device
BIN 32-bit	32-bit binary data or the start number of double-word device
BCD 4-digit	Four-digit binary-coded decimal data
BCD 8-digit	Eight-digit binary-coded decimal data
Real number	Floating-point data
Character string	Character string data
Device name	Device name data

TERMS

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

*□ represents the model or version.

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

Generic term/abbreviation	Description
■ Series	
Q series	A generic term for the Mitsubishi MELSEC-Q series programmable controller
■ CPU module type	
CPU module	A generic term for the Universal model QCPU
Universal model QCPU	A generic term for the Q00UJCPU, Q00UCPU, Q01UCPU, Q02UCPU, Q03UDCPU, Q03UDVCPU, Q03UDECPU, Q04UDHCPU, Q04UDVCPU, Q04UDEHCPU, Q06UDHCPU, Q06UDVCPU, Q06UDEHCPU, Q10UDHCPU, Q10UDEHCPU, Q13UDHCPU, Q13UDVCPU, Q13UDEHCPU, Q20UDHCPU, Q20UDEHCPU, Q26UDHCPU, Q26UDVCPU, Q26UDEHCPU, Q50UDEHCPU, and Q100UDEHCPU
Built-in Ethernet port QCPU	A generic term for the Q03UDVCPU, Q03UDECPU, Q04UDVCPU, Q04UDEHCPU, Q06UDVCPU, Q06UDEHCPU, Q10UDEHCPU, Q13UDVCPU, Q13UDEHCPU, Q20UDEHCPU, Q26UDVCPU, Q26UDEHCPU, Q50UDEHCPU, and Q100UDEHCPU
High-speed Universal model QCPU	A generic term for the Q03UDVCPU, Q04UDVCPU, Q06UDVCPU, Q13UDVCPU, and Q26UDVCPU
■ CPU module model	
QnUDVCPU	A generic term for the Q03UDVCPU, Q04UDVCPU, Q06UDVCPU, Q13UDVCPU, and Q26UDVCPU
QnUDE(H)CPU	A generic term for the Q03UDECPU, Q04UDEHCPU, Q06UDEHCPU, Q10UDEHCPU, Q13UDEHCPU, Q20UDEHCPU, Q26UDEHCPU, Q50UDEHCPU, and Q100UDEHCPU
■ Network module	
CC-Link IE module	A generic term for the CC-Link IE Controller Network module and CC-Link IE Field Network module
MELSECNET/H module	The abbreviation for the MELSECNET/H network module
Ethernet module	The abbreviation for the Ethernet interface module
CC-Link module	The abbreviation for the CC-Link system master/local module
■ Network	
CC-Link IE	A generic term for CC-Link IE Controller Network and CC-Link IE Field Network
MELSECNET/H	The abbreviation for the MELSECNET/H network system
■ Software package	
Programming tool	A generic term for GX Works2 and GX Developer
GX Works2	The product name of the software package for the MELSEC programmable controllers
GX Developer	
■ Others	
GOT	A generic term for Mitsubishi Graphic Operation Terminal, GOT-A*** series, GOT-F*** series, and GOT1000 series

CHAPTER 1 OVERVIEW

1.1 Features

The features specific to the Built-in Ethernet port QCPU are described below.

(1) Connection of programming tools and GOTs (👉 Page 24, CHAPTER 3)

- The Find CPU function makes it possible to find the Built-in Ethernet port QCPU connected to the same hub as programming tool and displays a list.
- MELSOFT connection allows access through routers in an environment such as a corporate LAN.

(2) Direct connection to a programming tool (simple connection) (👉 Page 33, CHAPTER 4)

The CPU module can be directly connected to a programming tool with a single Ethernet cable only, without using a hub (simple connection). In direct connections, communications can be performed simply by the transfer setup, without setting IP addresses.

(3) MC protocol communication (👉 Page 36, CHAPTER 5)

From an external device such as a personal computer or HMI, device data of the CPU module can be read or written, and this allows CPU module operation monitoring, data analysis, and production control.

(4) Data communications using the predefined protocol (👉 Page 49, CHAPTER 6)

The predefined protocol function sends and receives packets predefined by using GX Works2, enabling easy communications with external devices (such as measuring instruments and bar code readers). Protocol can be either selected from the prepared predefined protocol library, or created and edited by users.

(5) Socket communication function (👉 Page 60, CHAPTER 7)

By using instructions dedicated to socket communication, any data can be transferred from and to the external devices connected through Ethernet using TCP or UDP.

(6) Time setting function (SNTP client) (👉 Page 109, CHAPTER 8)

- Automatic time setting of the CPU module can reduce the maintenance cost for time setting.
- By sharing the same clock data among CPU modules connected to Ethernet via their built-in Ethernet ports, the order of errors between processes can be traced, facilitating problem solving.
- Since the automatic time setting is enabled upon power-on of the CPU module, operations can be started based on accurate clock data.

(7) File transfer function (FTP) (☞ Page 112, CHAPTER 9)

Each of the files stored in the CPU module can be read or written from the interfacing device with the FTP client function, and a large amount of data can be easily transferred.

(8) Remote password (☞ Page 134, CHAPTER 10)

Remote password setting can prevent unauthorized access from the outside and enhance the security of the system.

(9) IP address change function (☞ Page 140, CHAPTER 11)

The IP address of the built-in Ethernet port can be changed from a GOT, not in the Built-in Ethernet Port Setting of the PLC Parameter.

(10) IP packet transfer function (☞ Page 150, CHAPTER 12)

Communications can be performed with a device which supports the following IP addresses, which have been specified via a CC-Link IE Controller Network module or CC-Link IE Field Network module, using a protocol such as the FTP or HTTP via a built-in Ethernet port from an Ethernet device such as a personal computer.

- External devices on CC-Link IE Controller Network or CC-Link IE Field Network
- External devices on the Ethernet network, which are connected through the built-in Ethernet ports

Point 

Some functions have been added by the upgrade of the serial numbers of CPU modules or the programming tool. For the list of functions added by the upgrade, refer to Page 155, Appendix 3.

CHAPTER 2 COMMUNICATION SPECIFICATIONS

The following are the communication specifications of the built-in Ethernet port of the CPU module.

Item		Specification
Transmission specifications	Data transfer speed	100 or 10 Mbps
	Communication mode	Full-duplex or half-duplex
	Transmission method	Base band
	Maximum distance between hub and node	100 m
	Maximum number of nodes/connection	10BASE-T
100BASE-TX		Cascade connection: Up to two ^{*2}
Number of connections	TCP/IP	Total of 16 for socket communication, MELSOFT connection, MC protocol, and predefined protocol. One for FTP
	UDP/IP	
Connection cable ^{*1}	10BASE-T	Ethernet cable of category 3 or higher (STP/UTP cable) ^{*3}
	100BASE-TX	Ethernet cable of category 5 or higher (STP cable)

- *1 Straight cables can be used.
When the CPU module is directly connected to GOT with Ethernet cable, a cross cable of Category 5e or lower can also be used.
- *2 This number applies when a repeater hub is used.
When using a switching hub, check the number of cascaded stages with the manufacturer of the hub to be used.
- *3 Use of STP cables is recommended in an environment with noise.

Hubs with 10BASE-T or 100BASE-TX ports^{*4} can be used.
Up to 16 external devices can access one CPU module at the same time.

- *4 The ports must comply with the IEEE802.3 10BASE-T or IEEE802.3 100BASE-TX standards.

Point

- When connected to a hub, the CPU module determines the cable used (10BASE-T or 100BASE-TX) and the communication mode (full-duplex or half-duplex) according to the hub.
Set the hub into the half-duplex mode if the hub that does not have the auto-negotiation function.
- The operation of commercial devices used for the following applications is not guaranteed. Check the operation before using the module.
 - Internet (general public line)
(Internet-access service offered by an Internet service provider or a telecommunications carrier)
 - Firewall device(s)
 - Broadband router(s)
 - Wireless LAN
- If Ethernet communication is performed with "Specify service process execution counts" selected for "Service processing setting" in the PLC system tab of PLC parameter, a scan time increases by time for service processing (approximately 500ms).
To reduce it to 500ms or less, select an item other than "Specify service process execution counts".
(Example: Select "Specify service process time" and then enter a time value.)
- If broadcast storm occurs in the network, scan time may be increased.
- If the destination device of the CPU module does not respond due to power-off or other reasons, Ethernet communication of the CPU module may delay up to 500ms.

Remark

TCP and UDP are defined as follows:

- TCP (Transmission Control Protocol)

In communications among programmable controllers and networked devices, this protocol establishes a connection between port numbers of the two devices to perform reliable data communications.

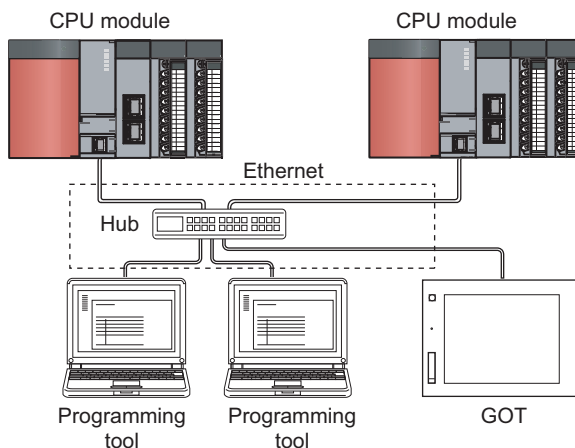
- UDP (User Datagram Protocol)

This is a connectionless protocol and thereby its speed is faster than that of TCP. However, the reliability in data communications is low. (Data may be lost or not be received in correct order.) Note that simultaneous broadcast is available.

Select an appropriate protocol, considering the specifications of the external device and the characteristics of the above protocols.

CHAPTER 3 CONNECTION OF PROGRAMMING TOOLS AND GOT

This chapter explains how to connect the CPU module to a programming tool or GOT.

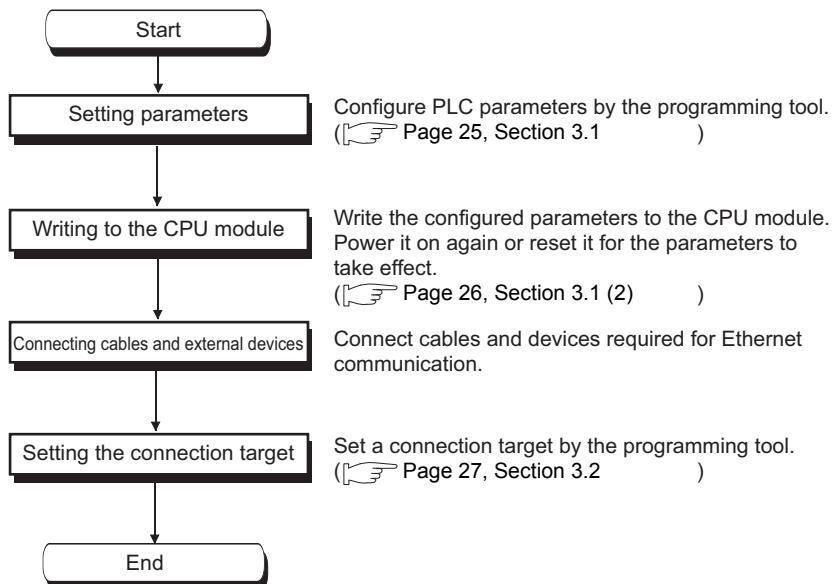


Point

A CPU module can be directly connected with the programming tool using one Ethernet cable. In direct connection (simple connection), the module and the tool can communicate with each other without each other's IP address in mind.

(☞ Page 33, CHAPTER 4)

To start Ethernet communication, perform the following steps.



For the GOT setting, refer to the following manual.

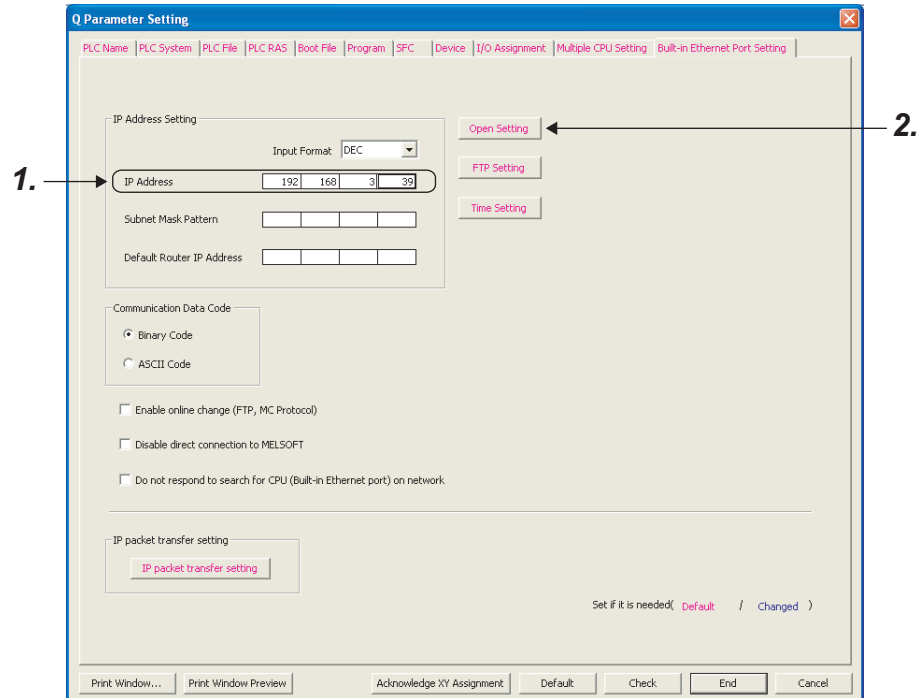
📖 GOT1000 Series Connection Manual (Mitsubishi Products)

3.1 Setting for the CPU Module

(1) PLC parameter setting

Select the "Built-in Ethernet Port Setting" tab and set the parameters.

Project window ⇨ [Parameter] ⇨ [PLC Parameter] ⇨ [Built-in Ethernet Port Setting]

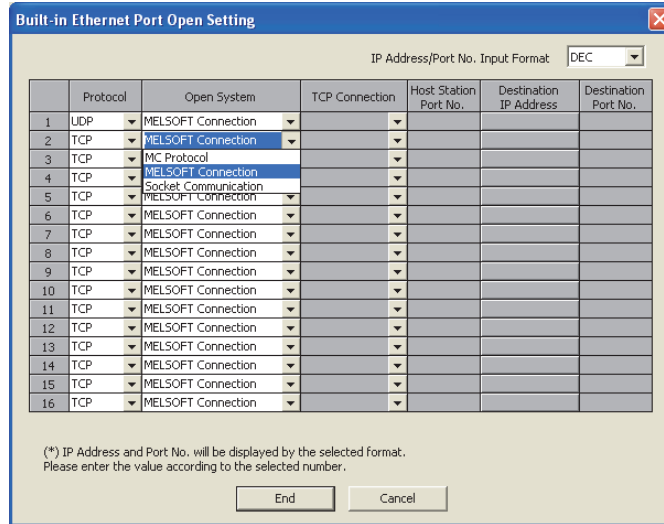


1. Set the IP address of the CPU module.

2. Set MELSOFT connection.

Project window ⇨ [Parameter] ⇨ [PLC Parameter] ⇨ [Built-in Ethernet Port Setting]

⇨  button



	Protocol	Open System	TCP Connection	Host Station Port No.	Destination IP Address	Destination Port No.
1	UDP	MELSOFT Connection				
2	TCP	MELSOFT Connection				
3	TCP	MC Protocol				
4	TCP	MELSOFT Connection				
5	TCP	Socket Communication				
6	TCP	MELSOFT Connection				
7	TCP	MELSOFT Connection				
8	TCP	MELSOFT Connection				
9	TCP	MELSOFT Connection				
10	TCP	MELSOFT Connection				
11	TCP	MELSOFT Connection				
12	TCP	MELSOFT Connection				
13	TCP	MELSOFT Connection				
14	TCP	MELSOFT Connection				
15	TCP	MELSOFT Connection				
16	TCP	MELSOFT Connection				

(*) IP Address and Port No. will be displayed by the selected format.
Please enter the value according to the selected number.

End Cancel

Item	Setting
Protocol	Select "TCP" or "UDP" depending on the connected device.
Open System	Select "MELSOFT Connection".

(2) Writing to the CPU module

From the "Write to PLC" window, write the parameter settings to the CPU module.

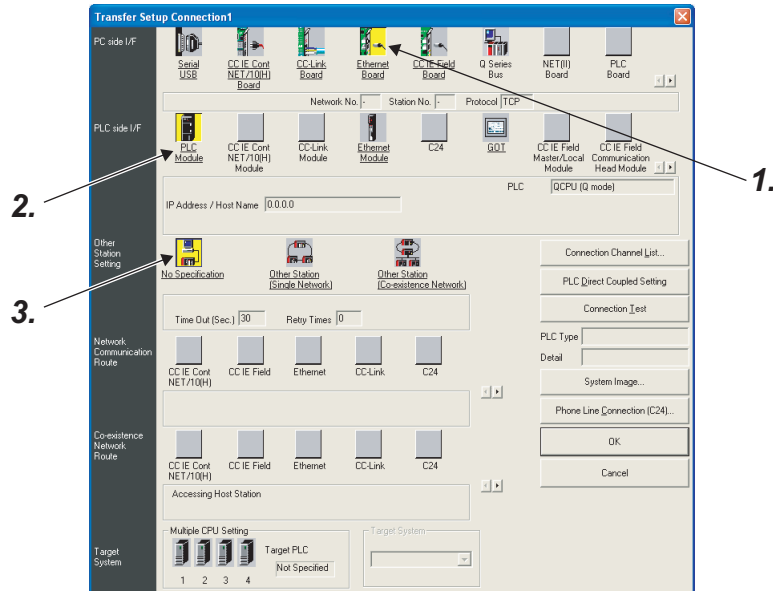
 [Online] ⇨ [Write to PLC]

After writing the parameters to the CPU module, power off and on or reset the CPU module to enable the parameters.


3.2 Setting for the Programming Tool

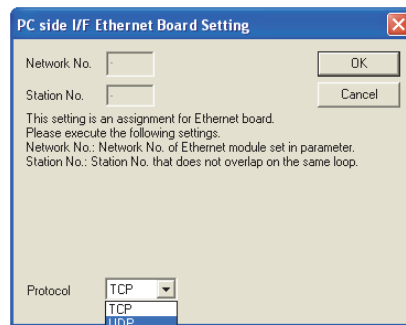
Configure the settings in the "Transfer Setup" window.

 Connection Destination window ⇨ [Connection1]



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

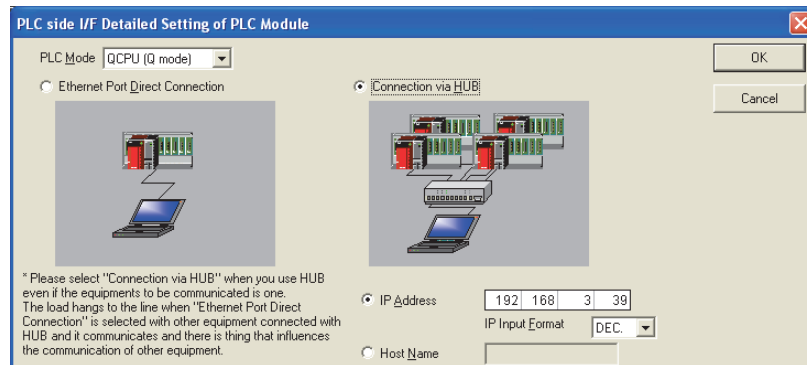
In the "PC side I/F Ethernet Board Setting" window, select a "TCP" or "UDP" protocol. Select the same protocol as the one set in the Open Setting window. ( Page 25, Section 3.1)



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

Enter the IP address or host name of the CPU module in the "PLC side I/F Detailed Setting of PLC Module" window, as shown below.

(For the host name, enter the name set in the Microsoft® Windows® hosts file.)

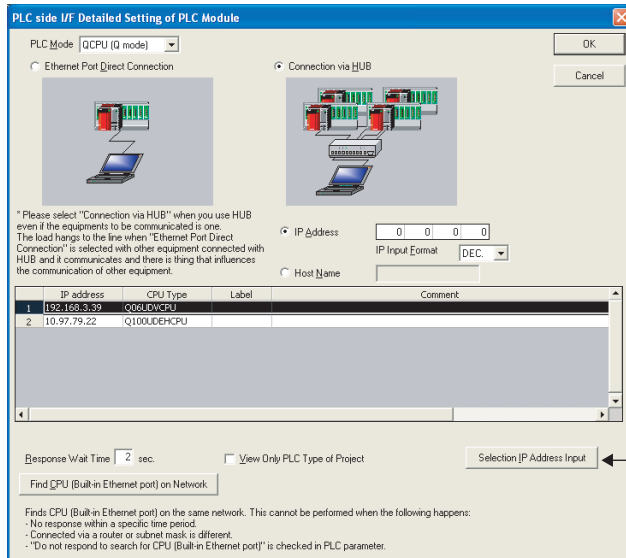


3. Set "Other Station Setting".

Select an item appropriate to the operating environment.

3.3 Searching CPU Modules on the Network

In a configuration using a hub, clicking **Find CPU (Built-in Ethernet port) on Network** in the "PLC side I/F Detailed Setting of PLC Module" window will start searching for CPU modules connected to the hub where the programming tool is also connected, and display a list of them.



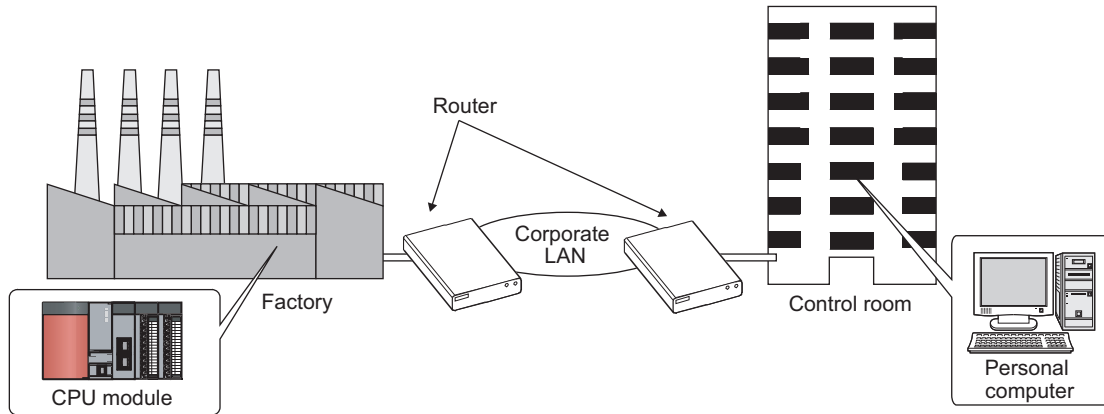
Clicking this button will automatically enter the IP address of the CPU module.

- CPU modules connected to cascaded hubs are also searched and a list of them is displayed.
- CPU modules connected via a router cannot be searched.
- Some CPU modules connected via wireless LAN may not be found since Ethernet communication may not be stable due to packet loss.
- If multiple CPU modules with the same IP address are found in the list, check the IP address parameters for the CPU modules. Starting communication with the IP address duplicated will cause a communication error.
- Appropriate CPU modules may not be found if a heavy load for service processing is applied. Increase the response waiting time value in the "Find CPU (Built-in Ethernet port)" window, or the service processing time value in the Service processing setting tab of PLC parameter.
- By selecting the option shown below in the Built-in Ethernet port tab of PLC parameter, the Find CPU function can be disabled and the system does not respond to a search request on the network.



3.4 Communication via Routers

From the built-in Ethernet port, access is available via routers on a network such as a corporate LAN.*1



- *1 For some functions as shown below, communications via a router are not available.
- Searching CPU modules on the network
 - Simultaneous broadcast in socket communication

For access via a router, follow the instruction in the step 1 on Page 25, Section 3.1 (1) to set the subnet mask pattern and the default router IP address in addition to the IP address.

Project window ⇔ [Parameter] ⇔ [PLC Parameter] ⇔ [Built-in Ethernet Port Setting]

Set the subnet mask pattern and default router IP address. →

IP Address Setting				
Input Format	DEC			
IP Address	192	168	3	39
Subnet Mask Pattern	255	255	255	0
Default Router IP Address	192	168	0	254

Open Setting
FTP Setting
Time Setting

3.5 Precautions

(1) IP address duplication

Check that the IP address is not duplicated when configuring a network or connecting a new device to a network. If the IP address is duplicated, a device may communicate with the wrong device.

Check for the IP address duplication in the following ways.

- Check for the IP address duplication with the find CPU function.
- Disconnect the device from the line and send ping to the IP address of the disconnected device.
Having a response means the IP address duplication.

(2) KeepAlive check

When the protocol is set to TCP, KeepAlive check is performed. (Checking for a response to a KeepAlive ACK message)

An alive check message is sent five seconds after reception of the last message from the connected device to check if the device returns a response or not. If no response is received, the alive check message will be resent at intervals of five seconds. When no response is received for 45 seconds, the connected device is regarded as non-existent and the connection is disconnected. If the connected device does not support the TCP KeepAlive function, the connection may be disconnected.

(3) Connections exceeding the setting

Do not exceed the number of connections set for the open settings parameters. Establishing too many TCP connections from a personal computer may cause the following states, depending on the application.

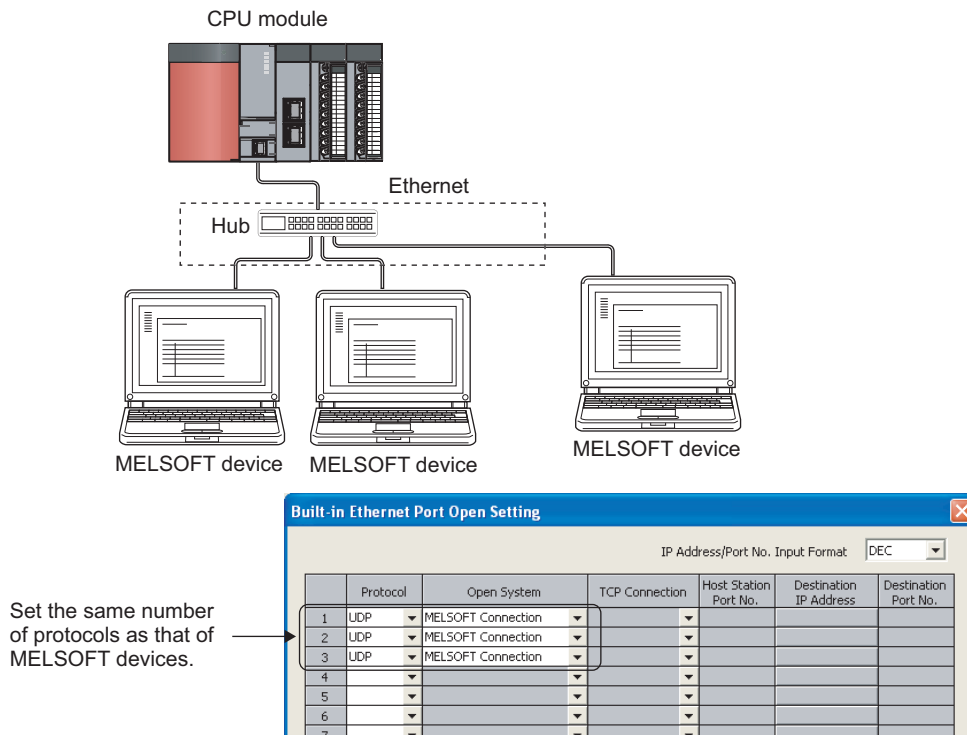
- Time before timeout error detection is increased.
- An unexpected timeout error occurs in any of the communicating devices.

(4) Retransmission on TCP connection

If no ACK response is returned from the other end of a TCP connection, the ACK will be resent six times, starting in 0.3 seconds after the first transmission, and then 0.6, 1.2, 2.4, 4.8, and 9.6 seconds. When no TCP ACK response is returned within 19.2 seconds after the last retransmission, the device is regarded as faulty and the connection is disconnected. (As a result, the connection is disconnected in total of 38.1 seconds.)

(5) MELSOFT connection over TCP or UDP

For TCP or UDP communications with multiple MELSOFT devices, set the same number of protocols as that of the connected MELSOFT devices in the setting of the PLC parameter.



Point

When all MELSOFT devices start communicating at the same time, devices may fail to communicate because of the congestion of communications. In such a case, schedule the timing for when each device starts communicating so that the communication congestion will not occur. When using GOTs, for example, set different rise time and time-out values in the GOTs.

(6) Sampling trace

When the function has been executed using the programming tool via a built-in Ethernet port, stop the function before powering off or resetting the CPU module.

(7) Remote STOP or remote PAUSE

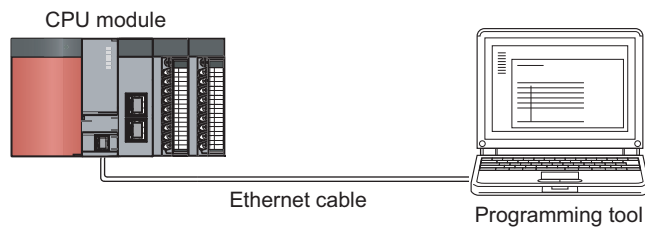
When remote STOP or remote PAUSE has been executed using the programming tool via a built-in Ethernet port, perform the following operations before powering off or resetting the CPU module.

- Remote RUN
- Remote RESET

CHAPTER 4 DIRECT CONNECTION TO PROGRAMMING TOOL (SIMPLE CONNECTION)

The CPU module can be directly connected to the programming tool with an Ethernet cable, without using a hub (simple connection).

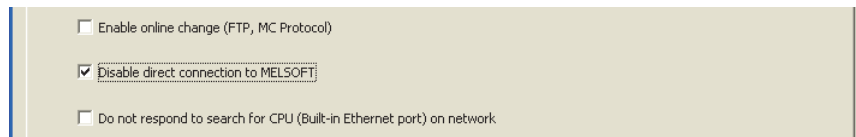
For direct connection, the IP address and host name need not be specified in the connection target setting. (Simultaneous broadcast is used.)



Point


An Ethernet cable used for direct connection will be longer compared with the case of using a USB cable. This can cause an unauthorized connection from a remote location.

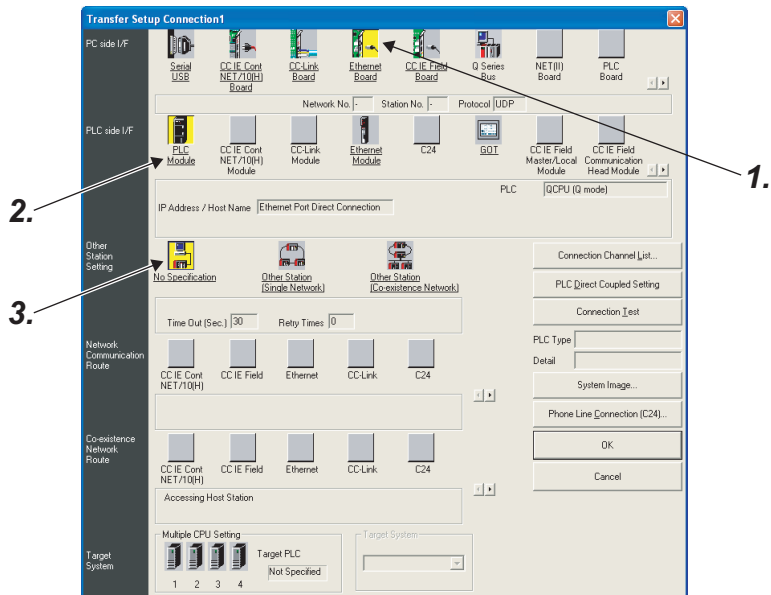
Unauthorized connections can be prevented by selecting the following option in the Built-in Ethernet port tab of the PLC parameter window.



4.1 Setting Method

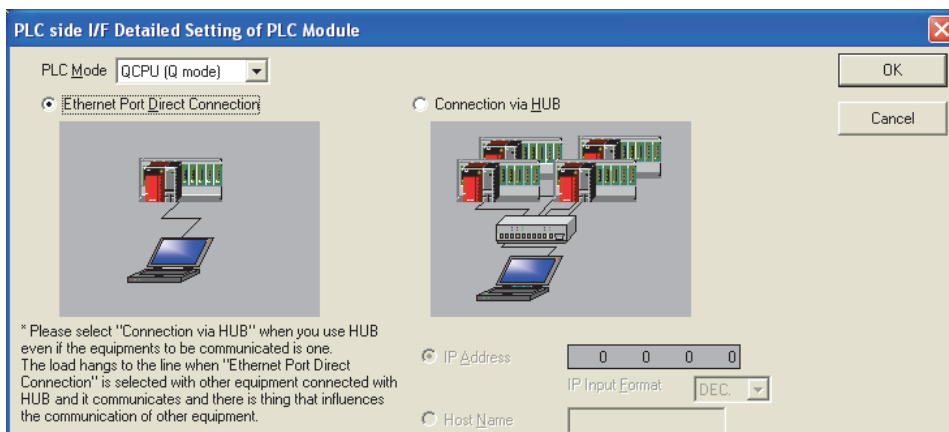
Set the items on the Transfer Setup window.

 Connection Destination window ⇔ [Connection1]



1. Select "Ethernet Board" for "PC side I/F".
2. Select "PLC Module" for "PLC side I/F".

In the "PLC side IF Detailed Setting of PLC Module" window, select the Ethernet Port Direct Connection checkbox as shown below.



3. Complete setting of "Other Station Setting".

Select an item appropriate to the operating environment.

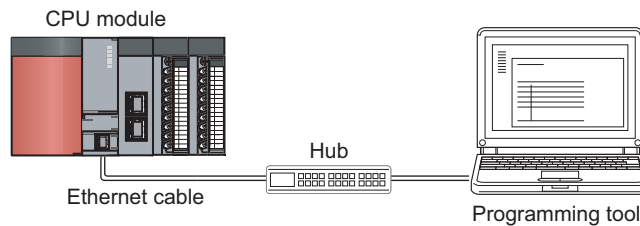
4.2 Precautions

(1) Connection to LAN line

When connecting the CPU module to a LAN line, do not set direct connection. Doing so will apply a load to the LAN line and adversely affect communications with other external devices.

(2) Indirect connection

- Do not set up direct connection when a CPU module is connected to an external device in a one-to-one basis using a hub as shown below.



- When two or more Ethernet ports are enabled in the network connections setting on the personal computer, communication by direct connection is not possible. In the setting, leave only one Ethernet port enabled for direct connection and disable other Ethernet ports.

(3) Conditions that disallow direct connection

When any of the following conditions is met, communication by direct connection may not be available. In that case, check the setting of the CPU module and/or personal computer.

- In the CPU module IP address bits, the bits corresponding to "0" in the personal computer subnet mask are all ON or all OFF.

Ex.	CPU module IP address	:	64.	64.	255.	255
	Personal computer IP address	:	64.	64.	1.	1
	Personal computer subnet mask	:	255.	255.	0.	0

- In the CPU module IP address bits, the bits corresponding to the host address of the class in the personal computer IP address are all ON or all OFF.

Ex.	CPU module IP address	:	64.	64.	255.	255
	Personal computer IP address	:	192.	168.	0.	1
	Personal computer subnet mask	:	255.	0.	0.	0

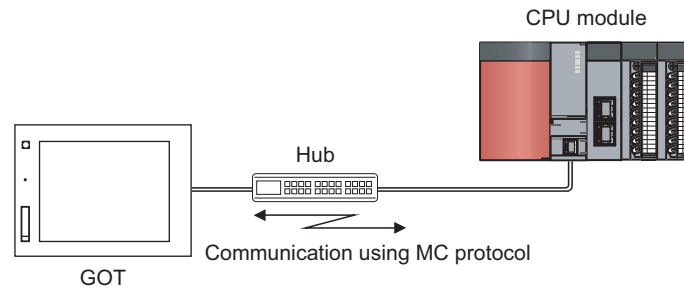
Remark

- The IP address pattern for each class is as follows.
Class A: 0.x.x.x to 127.x.x.x Class B: 128.x.x.x to 191.x.x.x Class C: 192.x.x.x to 223.x.x.x
- The host address for each class is the part shown with "0".
Class A: 255. 0. 0. 0 Class B: 255.255. 0. 0 Class C: 255.255.255. 0

CHAPTER 5 MC PROTOCOL COMMUNICATION

The built-in Ethernet port allows MC protocol communication. From an peripheral device such as a personal computer or HMI, device data of the CPU module can be read or written using MC protocol. Monitoring of CPU module operation, data analysis, and production control are available on a personal computer or HMI by these device data reading and writing.

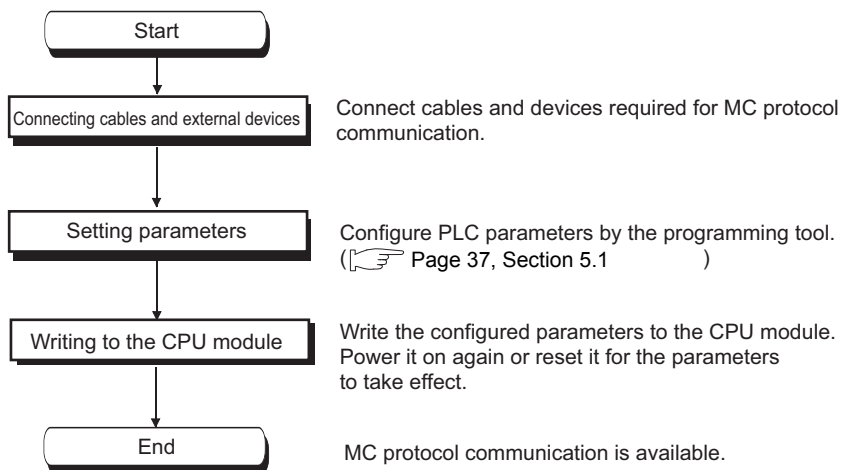
Besides, the remote password function can prevent unauthorized access from outside of the system. (☞ Page 134, CHAPTER 10)



Point

From the peripheral device such as a personal computer or HMI, only the CPU module connected can communicate using MC protocol.
An access to a CPU on another station via CC-Link network is not allowed.

To start MC protocol communication, perform the following steps.



For the MC protocol communication, refer to the following manual.

📖 MELSEC-Q/L MELSEC Communication Protocol Reference Manual

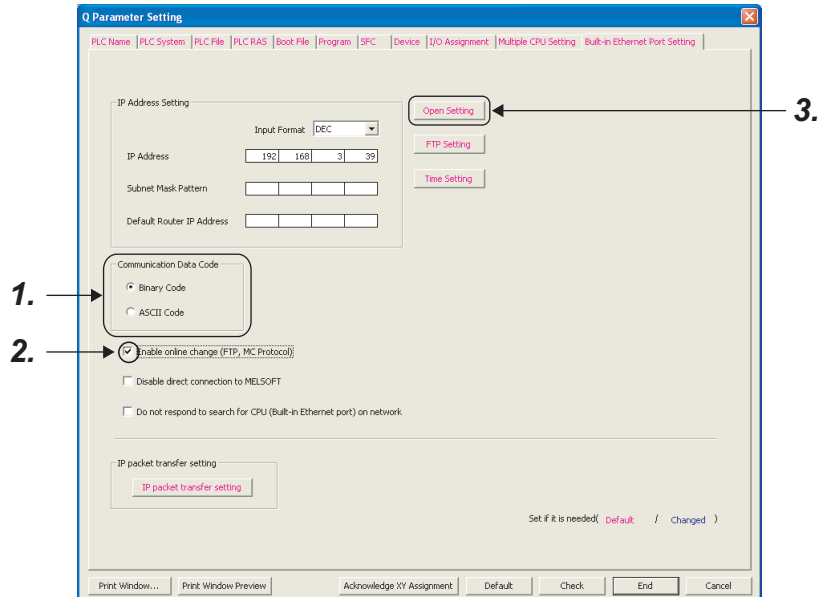
Remark

Access through routers is also available. When configuring the settings for it, set the subnet mask pattern and default router IP address. (☞ Page 30, Section 3.4)

5.1 Setting Method

Setting for communication using the MC protocol is described below.

Project window ⇨ [Parameter] ⇨ [PLC Parameter] ⇨ [Built-in Ethernet Port Setting]

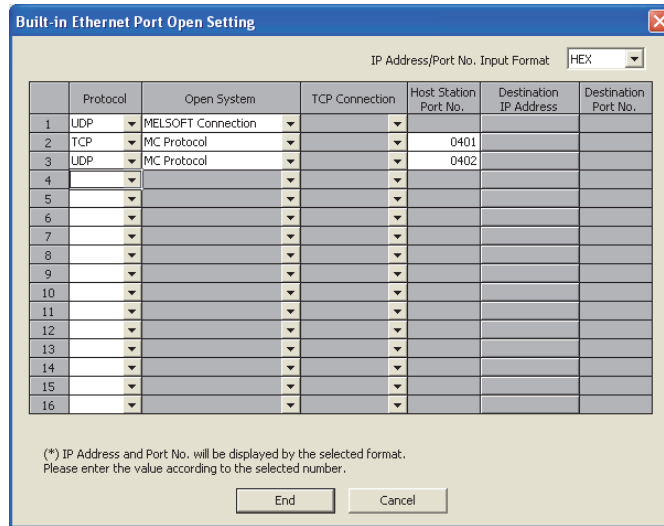


1. Select Binary or ASCII code as the communication data code used for MC protocol.
2. Select the "Enable online change (FTP, MC Protocol)" checkbox to enable data to be written to the CPU module even in the RUN state.

3. Set connections used for MC protocol communication.

Project window ⇨ [Parameter] ⇨ [PLC Parameter] ⇨ [Built-in Ethernet Port Setting]

⇨  button



	Protocol	Open System	TCP Connection	Host Station Port No.	Destination IP Address	Destination Port No.
1	UDP	MELSOFT Connection				
2	TCP	MC Protocol		0401		
3	UDP	MC Protocol		0402		
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						

(*) IP Address and Port No. will be displayed by the selected format.
Please enter the value according to the selected number.

End Cancel

Item	Description
Protocol	Select TCP or UDP depending on the target device.
Open System	Select "MC Protocol".
Host Station Port No.	Set the port number of the host station. (Setting range: 0401 _H to 1387 _H , 1392 _H to FFFE _H)*1

*1 Do not specify 1388_H to 1391_H because these ports are used by the system. (☞ Page 154, Appendix 2)

Point

When the "Enable online change (FTP, MC protocol)" setting is disabled, if the CPU module in the RUN state receives a data write request from the target device, data will not be written and a NAK message will be returned.

5.2 MC Protocol Commands

5.2.1 Command list

The following commands are executable for MC protocol communication of the CPU module.

(1) When QnA-compatible 3E frame is used

○ : Available, × : N/A

Function		Command (Subcommand)	Description	Number of processed points	CPU module state			
					STOP	RUN		
						Write enabled	Write disabled	
Device memory	Batch read	In units of bits	0401 (0001)	Reads bit devices in units of one point.	ASCII: 3584 points BIN: 7168 points			
		In units of words	0401 (0000)	Reads bit devices in units of 16 points.	960 words (15360 points)	○	○	○
				Reads word devices in units of one point.	960 points			
	Batch write	In units of bits	1401 (0001)	Writes bit devices in units of one point.	ASCII: 3584 points BIN: 7168 points			
		In units of words	1401 (0000)	Writes bit devices in units of 16 points.	960 words (15360 points)	○	○	×
				Writes word devices in units of one point.	960 points			
	Random read* ¹	In units of words	0403 (0000)	Reads bit devices in units of 16 or 32 points by randomly specifying the target.	192 points	○	○	○
				Reads word devices in units of one or two points by randomly specifying the target.				
	Test (Random write)	In units of bits	1402 (0001)	Sets or resets bit devices in units of one point by randomly specifying the target.	188 points	○	○	○
				Sets or resets bit devices in units of 16 or 32 points by randomly specifying the target.				
		In units of words * ¹	1402 (0000)	Writes word devices in units of one or two points by randomly specifying the target.	* ⁴	○	○	×
	Monitor registration * ¹ * ² * ³	In units of words	0801 (0000)	Registers bit devices to be monitored in units of 16 or 32 points.	192 points	○	○	○
Registers word devices to be monitored in units of one or two points.								
Monitor	In units of words	0802 (0000)	Monitors the devices registered.	Number of registered points	○	○	○	
Remote password	Unlock	1630 (0000)	Specifies a remote password to unlock the locked state.	-	○	○	○	
	Lock	1631 (0000)	Specifies a remote password to lock the unlocked state.	-	○	○	○	

*¹ Devices, TS, TC, SS, SC, CS, and CC cannot be specified in units of words.

Specifying any of these for monitor registration will cause an error (4032_H) at the time of monitoring execution.

*² For monitor registration, monitoring conditions cannot be set.

*³ Do not execute monitor registration from multiple devices. If executed, the last monitor registration takes effect.

*⁴ Set the number of processed points so that the following condition is satisfied.

(Number of word access points) × 12 + (Number of double-word access points) × 14 ≤ 1920

For bit devices, one point is regarded as 16 bits in word access and 32 bits in double-word access.

For word devices, one point is regarded as one word in word access, and two words in double-word access.

(2) When A-compatible 1E frame is used

○ : Available, × : N/A

Function		Command /response type	Description	Number of processed points	Status of CPU module			
					STOP	RUN		
						Write enabled	Write disabled	
Device memory	Batch read	In units of bits	00 _H	Reads bit devices in units of one point.	256 points	○	○	○
		In units of words	01 _H	Reads bit devices in units of 16 points.	128 words (2048 points)			
				Reads word devices in units of one point.	256 points			
	Batch write	In units of bits	02 _H	Writes bit devices in units of one point.	256 points	○	○	×
		In units of words	03 _H	Writes bit devices in units of 16 points.	40 words (640 points)			
				Writes word devices in units of one point.	256 points			
	Test (Random write)	In units of bits	04 _H	Sets/resets bit devices in units of one point by randomly specifying a device or device number.	80 points	○	○	×
		In units of words *1	05 _H	Sets/resets bit devices in units of 16 points by randomly specifying a device or device number.	40 words (640 points)			
				Writes word devices in units of one points by randomly specifying a device or device number.	40 points			
	Monitor data registration	In units of bits	06 _H	Registers bit devices to be monitored in units of one points.	40 points	○	○	○
		In units of words *1	07 _H	Registers bit devices to be monitored in units of 16 point.	20 words (320 points)			
				Registers word devices to be monitored in units of one point.	20 points			
	Monitor	In units of bits	08 _H	Monitors devices with monitor data registered.	Number of registered points	○	○	○
		In units of words *1	09 _H					

*1 Devices such as TS, TC, CS, and CC cannot be specified in units of words. If specified, an error (4032_H) occurs during monitoring.

5.2.2 Available devices

The following table lists the devices available in the commands used for MC protocol communication.

(1) When QnA-compatible 3E frame is used

Classification	Device	Device code ^{*1}		Device number range			
		ASCII	Binary				
Internal user device	Input	X*	9C _H	The number range of a device in a CPU module, which is accessed to, can be specified.	Hexadecimal		
	Output	Y*	9D _H		Hexadecimal		
	Internal relay	M*	90 _H	The number range of a device in a CPU module, which is accessed to, can be specified.	Decimal		
	Latch relay	L*	92 _H		Decimal		
	Annunciator	F*	93 _H		Decimal		
	Edge relay	V*	94 _H		Decimal		
	Link relay	B*	A0 _H		Hexadecimal		
	Data register	D*	A8 _H		Decimal		
	Link register	W*	B4 _H		Hexadecimal		
	Timer	Contact	TS		C1 _H	The number range of a device in a CPU module, which is accessed to, can be specified.	Decimal
		Coil	TC		C0 _H		
		Current value	TN		C2 _H		
	Retentive timer	Contact	SS	C7 _H	Note that the access to a local device is not possible.	Decimal	
		Coil	SC	C6 _H			
		Current value	SN	C8 _H			
	Counter	Contact	CS	C4 _H	The number range of a device in a CPU module, which is accessed to, can be specified.	Decimal	
		Coil	CC	C3 _H			
		Current value	CN	C5 _H			
		Link special relay	SB	A1 _H	Hexadecimal		
		Link special register	SW	B5 _H	Hexadecimal		
	Step relay	S*	98 _H	Decimal			
	Direct input ^{*2}	DX	A2 _H	Hexadecimal			
	Direct output ^{*2}	DY	A3 _H	Hexadecimal			
Internal system device	Function input	-	-	Cannot be accessed.	Hexadecimal		
	Function output	-	-		Hexadecimal		
	Function register	-	-		Decimal		
	Special relay	SM	91 _H	The number range of a device in a CPU module, which is accessed to, can be specified.	Decimal		
	Special register	SD	A9 _H	Note that the access to a local device is not possible.	Decimal		
Index register		Z*	CC _H	Decimal			
File register		R*	AF _H	The number range of a device in a CPU module, which is accessed to, can be specified.	Decimal		
		ZR	B0 _H		Decimal		
Extended data register		D*	A8 _H	<ul style="list-style-type: none"> • Binary: Within the device number range of the CPU module accessed • ASCII: 000000 to 999999 (up to 976.6K points) 	Decimal		
Extended link register		W*	B4 _H	The number range of a device in a CPU module, which is accessed to, can be specified.	Hexadecimal		

1 This is a code specified in MC protocol messages. When communicating data in ASCII code, specify the code in two characters. If the code consists of only one character, add "" (ASCII code: 2A_H) or a space (ASCII code: 20_H) after the character.

*2 Devices of DX/DY1000 or later are not available. Use X/Y devices to access devices of X/Y1000 or later.

(2) When A-compatible 1E frame is used

Classification	Device	Device code		Device range	Device number
		ASCII	Binary		
internal user device	Input	5820 (35 _H /38 _H /32 _H /30 _H)	58 _H /20 _H	X0 to X7FF	0000 _H to 07FF _H
	Output	5920 (35 _H /39 _H /32 _H /30 _H)	59 _H /20 _H	Y0 to Y7FF	0000 _H to 07FF _H
	Internal relay	4D20 (34 _H /44 _H /32 _H /30 _H)	4D _H /20 _H	<ul style="list-style-type: none"> • M0 to M8191 • M9000 to M9255 (SM1000 to SM1255)* ¹ Note, however, that local devices cannot be accessed.	<ul style="list-style-type: none"> • 0000_H to 1FFF_H • 2328_H to 2427_H
	Latch relay	-	-	Cannot be accessed.	
	Annunciator	4620 (34 _H /36 _H /32 _H /30 _H)	46 _H /20 _H	F0 to F2047	0000 _H to 07FF _H
	Edge relay	-	-	Cannot be accessed.	
	Link relay	4220 (34 _H /32 _H /32 _H /30 _H)	42 _H /20 _H	B0 to BFFF	0000 _H to 0FFF _H
	Data register	4420 (34 _H /34 _H /32 _H /30 _H)	42 _H /20 _H	<ul style="list-style-type: none"> • D0 to D6143 • D9000 to D9255 (SD1000 to SD1255)* ¹ Note, however, that local devices cannot be accessed.	<ul style="list-style-type: none"> • 0000_H to 17FF_H • 2328_H to 2427_H
Link register	5720 (35 _H /37 _H /32 _H /30 _H)	57 _H /20 _H	W0 to WFFF	0000 _H to 0FFF _H	

Classification	Device		Device code		Device range	Device number
			ASCII	Binary		
internal user device	Timer	Contact	5453 (35 _H /34 _H /35 _H /33 _H)	54 _H /53 _H	T0 to T2047 Note, however, that local devices cannot be accessed.	0000 _H to 07FF _H
		Coil	5443 (35 _H /34 _H /34 _H /33 _H)	54 _H /43 _H		
		Current value	544E (35 _H /34 _H /34 _H /45 _H)	54 _H /4E _H		
	Retentive timer	Contact	-	-	Cannot be accessed.	
		Coil	-	-		
		Current value	-	-		
	Counter	Contact	4353 (34 _H /33 _H /35 _H /33 _H)	43 _H /53 _H	C0 to C1023 Note, however, that local devices cannot be accessed.	0000 _H to 03FF _H
		Coil	4343 (34 _H /33 _H /34 _H /33 _H)	43 _H /43 _H		
		Current value	434E (34 _H /33 _H /34 _H 45 _H)	43 _H /4E _H		
	Link special relay		-	-	Cannot be accessed.	
	Link special register		-	-		
	Step relay		-	-		
	Direct input		-	-		
	Direct output		-	-		
	Function input		-	-		
Internal system device	Function output		-	-	Cannot be accessed.	
	Function register		-	-		
	Special relay		-	-		
	Special register		-	-		
	Index register		-	-		
File register* ²		5220 (35 _H /32 _H /32 _H /30 _H)	52 _H /20 _H	R0 to R8191 Note that local devices cannot be accessed.	0000 _H to 1FFF _H	
Extended data register		4420 (34 _H /34 _H /32 _H /30 _H)	42 _H /20 _H	• D0 to D6143 • D9000 to D9255 (SD1000 to SD1255)* ¹	• 0000 _H to 17FF _H • 2328 _H to 2427 _H	
Extended link register		5720 (35 _H /37 _H /32 _H /30 _H)	57 _H /20 _H	W0 to WFFF	0000 _H to 0FFF _H	

*1 When M9000 to M9255 (D9000 to D9255) are specified, specify SM1000 to SM1255 (SD1000 to SD1255). Make sure to check the checkbox in the "A-PLC Compatibility Setting" field in the PLC system tab of the PLC parameters.

*2 Available for the CPU modules with the following serial number (first five digits).

- QnUDE(H)CPU: "14112" or later
- QnUDVCPU: "15043" or later

5.3 Precautions

(1) Number of devices

Only the external devices whose open system is set to "MC Protocol" can be connected concurrently using MC protocol.

 Project window ⇨ [Parameter] ⇨ [PLC Parameter] ⇨ [Built-in Ethernet Port Setting]

⇨  button

(2) Data communication frame

The following table lists the availability of data communication frames.

Communication frame	Availability
4E frame	×
QnA-compatible 3E frame	○
A-compatible 1E frame	○ ^{*1}

*1 Available for the QnUDE(H)CPU with a serial number (first five digits) of "13102" or later

(3) Access range

- Only the connected CPU module can be accessed. Accessing another module will cause an error.
- In a multiple CPU system, other CPU modules not connected to Ethernet cannot be accessed.
- Accessing a CPU module on another station in CC-Link IE, MELSECNET/H, Ethernet or CC-Link via a connected CPU module is not possible.

(4) When UDP is selected for Protocol

- If a new request message is sent to a UDP port after the previous request message is sent to the same port and before no response is returned, the new request message will be discarded.
- Setting the same host station port number for multiple UDP connections is regarded as one setting. For communication with multiple devices using the same host station port number, select TCP.

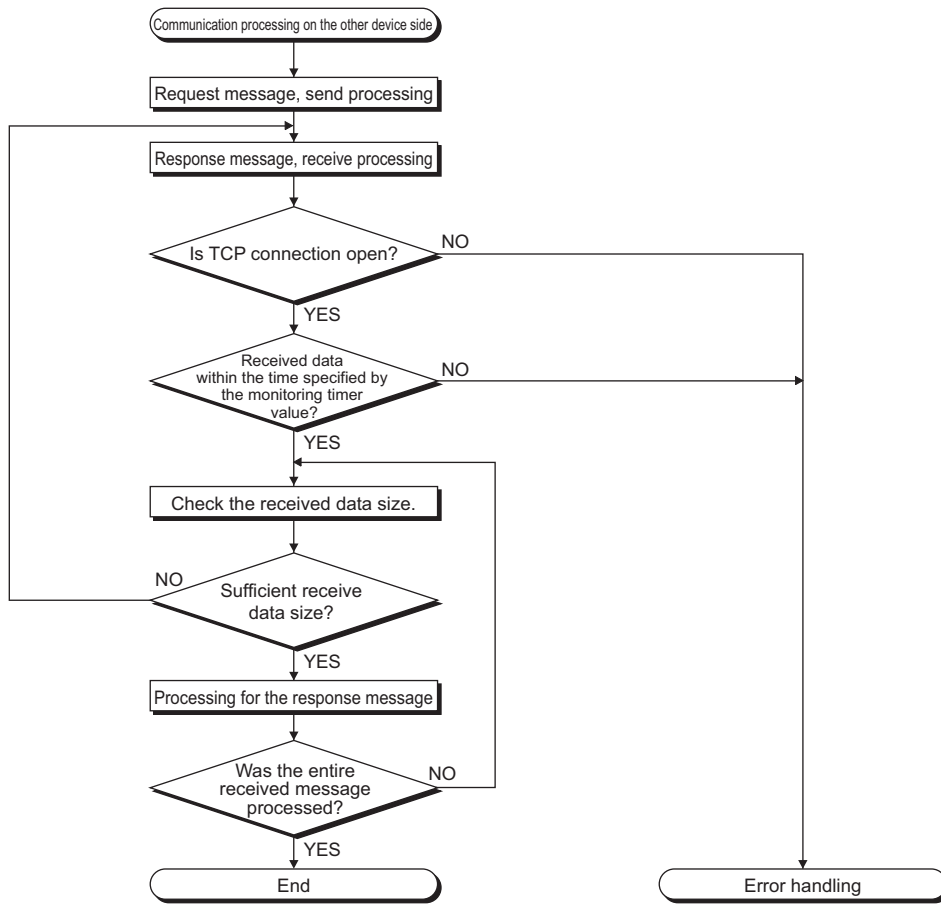
(5) File access during MC communication

The CPU module will perform file access processing prior to Ethernet communication processing. Because of this, processing of the MC protocol function may be delayed if a file is accessed by FTP or a programming tool during use of the MC protocol function.

When accessing a file while response time monitoring is performed on the connected device with the MC protocol function, add the time required for file access to the monitoring time.

(6) Receiving a response message

The following shows an example of receive processing on the other device side.




Point

For Ethernet communication, TCP socket functions are used inside personal computers. The functions do not have boundary concept. Therefore, if the sender sent data by calling the "send" function once, the receiver needs to call the "recv" function once or more times to read out the data. ("send" does not correspond to "recv" on the one-to-one basis.) For this reason, the processing shown above is always required on the program of the receiving device. Note that, if the "recv" function is used in blocking mode, data may be read by calling the function once.

5.4 Error codes, end codes, and abnormal codes in MC protocol communication

When an error occurs during MC protocol communication, an error code is sent from the CPU module to the external device. The following table lists error codes, error descriptions, and corrective actions.

(1) When QnA-compatible 3E frame is used

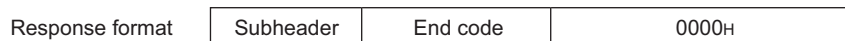
Error code (Hexadecimal)	Description	Corrective action
4000 _H to 4FFF _H	Errors detected by the CPU module (Errors occurred in other than MC protocol communication)	Refer to the following manual.  QCPU User's Manual(Hardware Design, Maintenance and Inspection)
0055 _H	Although online change is disabled, the connected device requested the RUN-state CPU module for data writing.	<ul style="list-style-type: none"> • Before enabling online change, write the data. • Change the CPU module state to STOP and write the data.
C050 _H	When "Communication Data Code" is set to ASCII Code, ASCII code data that cannot be converted to binary were received.	<ul style="list-style-type: none"> • Select Binary Code for "Communication Data Code", and restart the CPU module. • Correct the send data of the connected device and resend the data.
C051 _H to C054 _H	The number of read or write points is outside the allowable range.	Correct the number of read or write points, and resend the data to the CPU module.
C056 _H	The read or write request exceeds the maximum address.	Correct the start address or the number of read or write points, and resend the data to the CPU module. (The maximum address must not be exceeded.)
C058 _H	The request data length after ASCII-to-binary conversion does not match the data size of the character area (a part of text data).	Check and correct the text data or the request data length of the header, and resend the data to the CPU module.
C059 _H	<ul style="list-style-type: none"> • The command and/or subcommand are specified incorrectly. • The CPU module does not support the command and/or subcommand. 	<ul style="list-style-type: none"> • Check the request. • Use commands and/or subcommands supported by the CPU module.
C05B _H	The CPU module cannot read data from or write data to the specified device.	Check the device to be read or written.
C05C _H	The request data is incorrect. (e.g. reading or writing data in units of bits from or to a word device)	Correct the request data and resend it to the CPU module. (e.g. subcommand correction)
C05D _H	No monitor registration	Perform monitor registration before monitoring.
C05F _H	The request cannot be executed to the CPU module.	<ul style="list-style-type: none"> • Correct the network number, PC number, request destination module I/O number, or request destination module station number. • Correct the read/write request data.
C060 _H	The request data is incorrect. (ex. incorrect specification of data for bit devices)	Correct the request data and resend it to the CPU module.
C061 _H	The request data length does not match the number of data in the character area (a part of text data).	Check and correct the text data or the request data length of the header, and resend the data to the CPU module.
C06F _H	The CPU module received a request message in ASCII format when "Communication Data Code" is set to Binary Code, or received it in binary format when the setting is set to ASCII Code. (This error code is only registered to the error history, and no abnormal response is returned.)	<ul style="list-style-type: none"> • Send a request message that matches the "Communication Data Code" setting. • Change the "Communication Data Code" setting so that it will match the request message.
C070 _H	The device memory extension cannot be specified for the target station.	Read data from or write data to the device memory without specifying the extension.
C0B5 _H	The CPU module cannot handle the data specified.	<ul style="list-style-type: none"> • Correct the request data. • Stop the current request.
C200 _H	The remote password is incorrect.	Correct the remote password, and unlock and lock the remote password function again.

Error code (Hexadecimal)	Description	Corrective action
C201 _H	The port used for communication is locked with the remote password. Or, because of the remote password lock status with "Communication Data Code" set to ASCII Code, the subcommand and later part cannot be converted to a binary code.	Unlock the remote password before communication.
C204 _H	The connected device is different from the one that requested for unlock processing of the remote password.	From the device that requested the unlock processing, request for lock processing of the remote password.

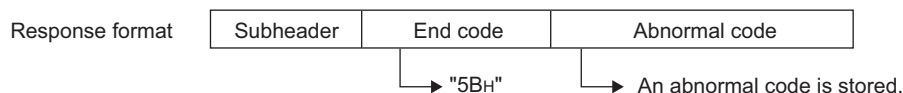
(2) When A-compatible 1E frame is used

An end code and abnormal code are added to a response in MC protocol communication that uses A-compatible 1E frames.

- ◆ When the end code is a value other than "5B_H"



- ◆ When the end code is "5B_H"




(a) End codes

End code	Description	Corrective action
00 _H	Data is communicated successfully.	-
54 _H	When the communication data code setting is set to ASCII code in the Built-in Ethernet port QCPU, ASCII code data that cannot be converted to binary code was received.	Correct the send data on the external device side.
55 _H	When the setting for online change is disabled on the Built-in Ethernet port tab of PLC parameter in GX Developer, an external device requested online change to the CPU module.	<ul style="list-style-type: none"> • Enable online change and write data. • Change the status of the CPU module to STOP and write data.
56 _H	An external device specified the incorrect device.	Specify the device correctly.
57 _H	<ul style="list-style-type: none"> • The number of points for a command specified by an external device exceeds the maximum number of processing points (number of points can be processed per communication) for each processing. • The number of points specified from the start address (start device number) exceeds the maximum address (maximum device number) for each processing. • The length of a command message is too short to analyze. 	<ul style="list-style-type: none"> • Correct the number of points specified or the start address (start device number). • Check the command.
	Monitoring was requested without monitor data being registered.	Register the monitor data.
58 _H	The start address (start device number) of a command specified by an external device exceeds the setting range.	Correct the start address within the setting range for each processing.
	The file register (R) cannot be specified.	Check the device.
	<ul style="list-style-type: none"> • A word device is specified in a command for bit devices. • In a command for word devices, the start number of a bit device is specified by a value other than a multiple of 16. 	Correct the command or the specified device.
5B _H	The CPU module cannot process requests from an external device.	Identify an error location according to the abnormal code and correct the error.

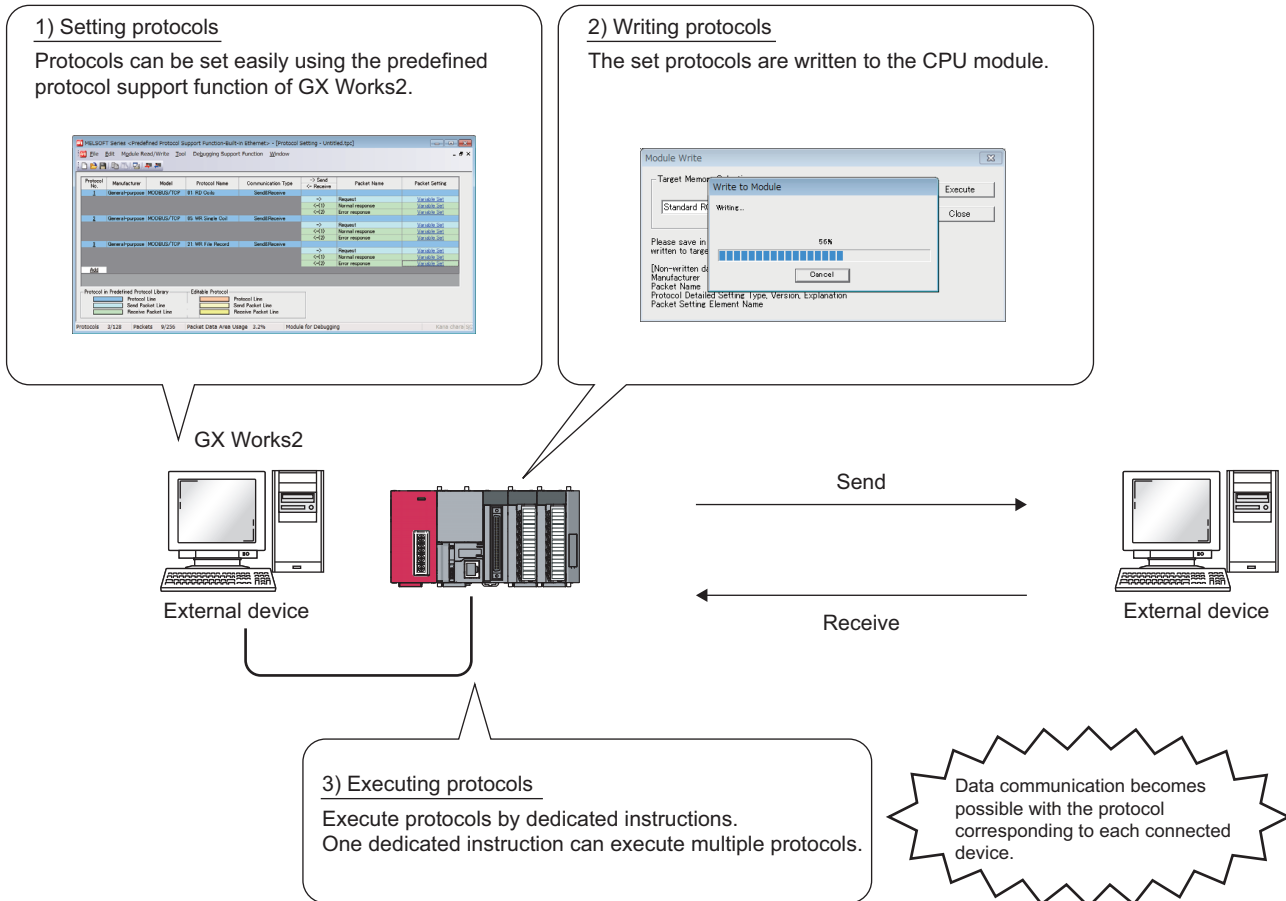
(b) Abnormal codes

If the end code is "5B_H," an abnormal code is attached immediately after the code. The following table lists the abnormal codes to be attached, error details, and required actions.


Abnormal code	Description	Corrective action
4000 _H to 4FFF _H	CPU detected error (Error that occurred in other than communication using the MC protocol)	Refer to the following manual.  QCPU User's Manual (Hardware Design, Maintenance and Inspection)
C080 _H	PC number of another station (01 _H to 80 _H) is specified.	Specify the PC number of own station (FF _H).
C201 _H	The communication target port is in the remote password locked status. Or, when the communication data code setting is set to ASCII code, any data of subcommands and later cannot be converted to binary code since the remote password is locked.	Perform remote password unlock processing before communication.

CHAPTER 6 DATA COMMUNICATIONS USING THE PREDEFINED PROTOCOL Note 6.1

The predefined protocol function sends and receives packets predefined by using GX Works2, enabling easy communications with external devices (such as measuring instruments and bar code readers). Protocols can be either selected from the prepared predefined protocol library, or created and edited by users.



6

 **Note 6.1** Universal

The predefined protocol function can only be used for the QnUDVCPU.
 When using the predefined protocol function, check the versions of the CPU module and GX Works2.
 (👉 Page 155, Appendix 3)

6.1 Specifications

The following table lists the specifications.

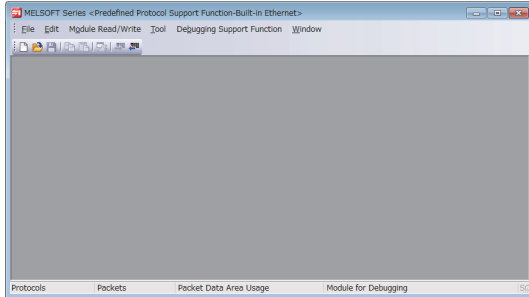
Item		Description
Protocol setting data	Number of protocols ^{*1}	Up to 128
	Number of packets ^{*2}	Up to 256
	Packet area data size ^{*3}	Up to 12288 bytes
Available connection		Connection No.1 to No.16
Protocol execution method		SPECPRCTL instruction
Communication data code		Binary code
Length of data that can be sent or received at a time		Up to 2046 bytes

- *1 Number of protocols specified as the protocol setting data
- *2 Total number of packets set to each protocol
- *3 Size as a sum of all packets

6.2 Setting Method

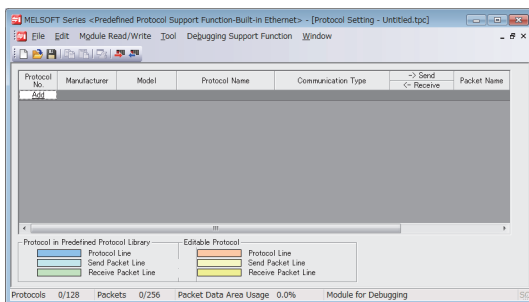
The setting required for using the predefined protocol function is described below.

1. Open the "Predefined Protocol Support Function" window.



[Tool] ⇨ [Predefined Protocol Support Function] ⇨ [Built-in Ethernet]

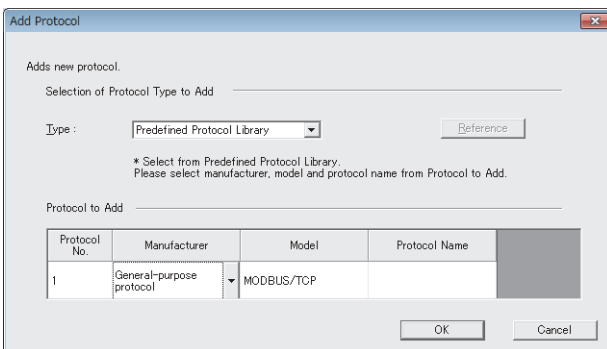
2. Create a new file.



[File] ⇨ [New...] ⇨ "Add"

3. In the "Add Protocol" window, select "Predefined Protocol Library" or "Add New".

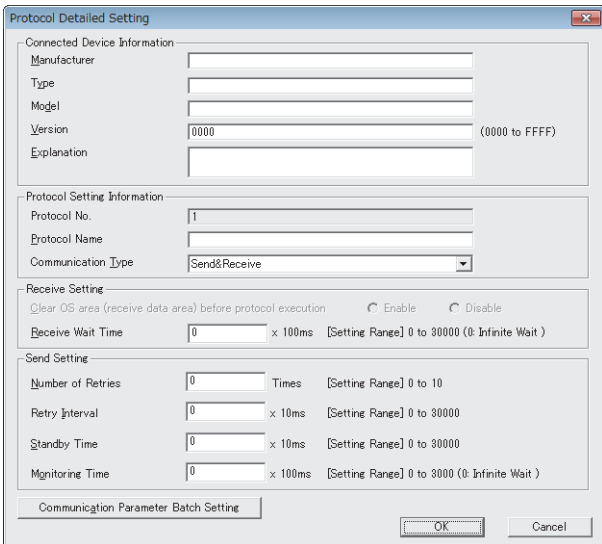
When "Predefined Protocol Library" is selected, select a desired protocol from the library registered in GX Works2.



Point

Select "User Protocol Library" in the "Add Protocol" window to read the protocol stored by the user. For details of the user protocol library, refer to the following.

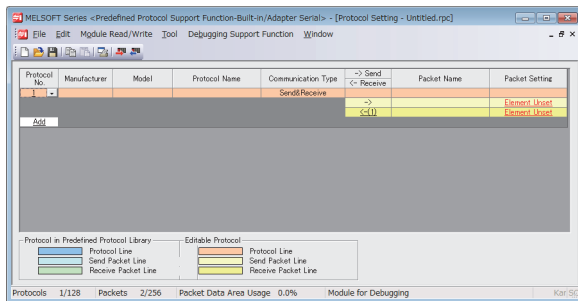
GX Works2 Version 1 Operating Manual (Intelligent Function Module)



4. Set the items required for data communications.

In the "Protocol Detailed Setting" window, set the communication parameters for the protocol.

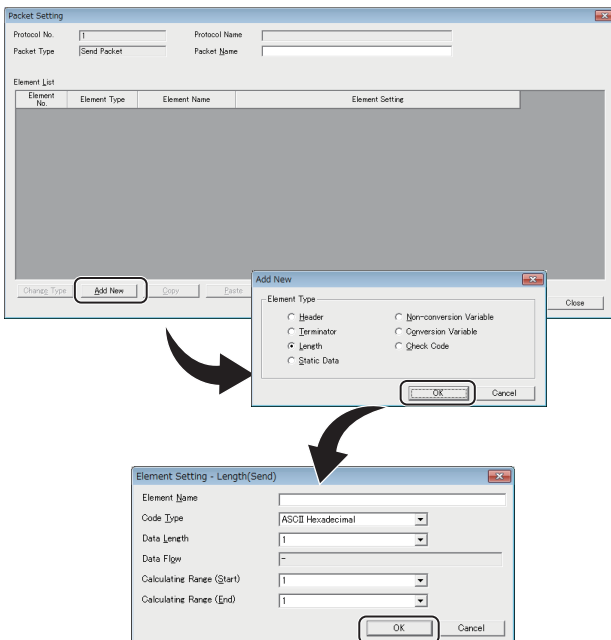
- ☞ "Protocol Setting" window ⇒ Select protocol ⇒ [Edit] ⇒ [Protocol Detailed Setting]

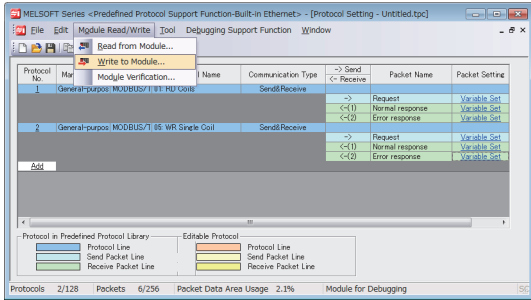


5. Set the packet configuration.

In the "Packet Setting" window, set the configuration of packets to be sent or received.

- ☞ "Protocol Setting" window ⇒ "Variable Unset" or "Element Unset"





6. Write the protocol setting data to the CPU module.

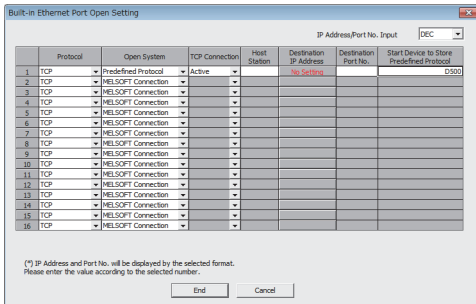
In the "Writing Protocol Setting" window, specify the write-target drive in the CPU module and write the protocol setting data.*1

[Module Read/Write] ⇨ [Write to Module]

- *1 The written protocol setting data will be enabled when
- the CPU module is powered on or is reset, or
 - SM1355 (Predefined protocol setting check request) is turned on.

Point

- To enable the written protocol setting data without powering off or resetting the CPU module, turn on SM1355 (Predefined protocol setting check request). Before enabling the protocol setting data, stop the CPU module and check that no instruction is being executed. Depending on the on-timing of SM1355 (Predefined protocol setting check request), the instruction being executed may end abnormally.

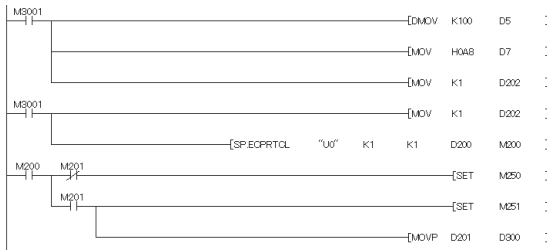


7. Set the connection for connecting the predefined protocol.

[PLC Parameter] ⇨ [Built-in Ethernet Port Setting] ⇨ [Open Setting]

Item	Description
Protocol	Select "TCP" or "UDP".
Open System	Select "Predefined Protocol".
TCP Connection	When "Protocol" is "TCP", select the connection method. (☞ Page 62, Section 7.1)
Host Station	Set the port number of the CPU module. Setting range: 0001 _H to 1387 _H , 1392 _H to FFFE _H (1 to 4999, 5010 to 65534)*1
Destination IP Address	Set the IP address of the external device. Setting range: 0.0.0.1 to 223.255.255.254
Destination Port No.	Set the port number of the external device. • Setting range: 0001 _H to FFFE _H (1 to 65534) (for TCP) • Setting range: 0001 _H to FFFF _H (1 to 65534/65535) (for UDP)
Start Device to Store Predefined Protocol	Set the start device number for storing the protocol execution status, received data verification result, number of protocol executions, and protocol cancellation status. (☞ Page 55, Section 6.2 (1)) The area of 19 words starting from the specified device is used. Available devices: D, W, R, ZR

*1 Do not specify 1388_H to 1391_H (5000 to 5009) because these ports are used by the system. (☞ Page 154, Appendix 2)

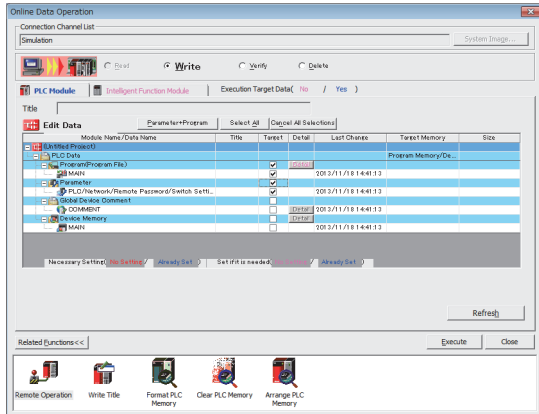


8. Create a program for starting data communications.

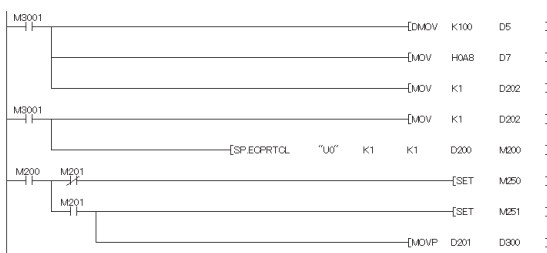
- In the case of Active open, create the program in which SP.SOCOPEN instruction is used to establish a connection. (☞ Page 80, Section 7.4.1)
- To activate the predefined protocol, the SP.ECPRTCL instruction is used. (☞ Page 57, Section 6.4.1)

9. Write the program to the CPU module.

[Online] ⇄ [Write to PLC]



10. Execute the program written to the CPU module by using the SP.ECPRTCL instruction.



For the protocol setting method, refer to the following.

GX Works2 Version 1 Operating Manual (Intelligent Function Module)

(1) Start device to store the predefined protocol operation status

In the open setting, set the start device where the predefined protocol operation status is to be stored. The following information is stored in the area of 19 words starting from the selected device.

Start Device to Store Predefined Protocol (offset)	Name	Description*1
+0	Protocol execution status	The execution status of the predefined protocol function is stored. (Default: 0) <ul style="list-style-type: none"> • 0: Not executed • 1: Waiting for data to be sent • 2: Data being sent • 3: Waiting for data to be received • 5: Execution completed
+1	Received data verification result (received packet No.1)	The verification result of received packet No.1 is stored. (Default: 0) (☞ Page 55, Section 6.2 (1) (a)) <ul style="list-style-type: none"> • b0 to b7: Element No. where the verification result did not match • b8 to b15: The cause of mismatch (verification result code)
+2 to 16	Received data verification result (received packet No.2 to 16)	The bit configuration is the same as the received packet No.1.
+17	Number of protocol executions	The number of protocol executions is stored. (Default: 0) <ul style="list-style-type: none"> • 0: No execution • 1 to 65535: Number of executions (The value remains the same after 65535.)
+18	Protocol cancellation specification	The protocol cancellation request status is stored. (Default: 0) <ul style="list-style-type: none"> • 0: No cancellation requested • 1: Cancellation requested (set by users) • 2: Cancellation completed (set by the system)

*1 Even after data communications by the predefined protocol function (executed by the SP.ECPRTCL instruction) is completed, the stored values are held.

(a) Received data verification result

The following information is stored in the Received data verification result.

- Element No. where the verification result did not match (b0 to b7)

Stored value	Description
0	Verification matched
1 to 32	Element No. where the verification result did not match
FF _H	Verification not performed

- The cause of mismatch (verification result code) (b8 to b15)

Stored value	Description	Cause
00 _H	Normal	—
01 _H	Insufficient receive data	The total packet size of receive data is smaller than that set in protocol data.
10 _H	Data not matched	The receive data do not match the value set in protocol data.
11 _H	ASCII-Binary conversion error	When "ASCII Hexadecimal" is set in Code Type, data not in ASCII code are received.
12 _H	Data length error	The received Length value exceeds 2046 bytes.
30 _H	Data length size error	The Length value received from the external device does not match the actual length of received data.
FF _H	Verification not performed	—

6.3 Setting Items of Predefined Protocol Support Function

6.3.1 Communication type

There are three protocol communication types: "Send Only", "Receive Only", and "Send & Receive".

For details of the protocol communication types, refer to the following.

 Q Corresponding Ethernet Interface Module User's Manual (Basic)

6.3.2 Packet elements set for "Packet Setting"

The packet elements include "Length", "Static data", "Non-conversion Variable", and "Non-verified Reception". Up to 32 elements can be set in a single packet.

The protocol setting method and packet elements are the same as those of the Ethernet interface module, except for some differences.

 Q Corresponding Ethernet Interface Module User's Manual (Basic)

The following are the differences.

- Read the E71 (Ethernet interface module) as the CPU module in the manual.
- Buffer memory cannot be set as a data storage area for non-conversion variable.

6.4 Predefined Protocol Function Instructions

The predefined protocol function instructions are provided for the CPU module to use the predefined protocol function. This section describes the predefined protocol function instructions.

Instruction	Description	Reference
SP.SOCOPEN	Establishes a connection with the external device. If an Active open is performed by the CPU module, the SP.SOCOPEN instruction is used to establish a connection with the external device.	Page 62, Section 7.1 Page 80, Section 7.4.1
SP.SOCCLOSE	Closes a connection with the external device.	Page 84, Section 7.4.2
SP.SOCCINF	Reads out connection information.	Page 98, Section 7.4.6
SP.SOCCSET	Changes the target of the connection for UDP/IP communications.	Page 101, Section 7.4.7
SP.ECPRTCL	Executes the configured predefined protocol.	Page 57, Section 6.4.1

Point


If the instruction has a completion device, do not change any data (such as control data and request data) specified for the instruction until the execution is completed.

6.4.1 Executing the predefined protocol (SP.ECPRTCL)

This instruction sends and receives packets set by the predefined protocol support function of GX Works2. Usage of the SP.ECPRTCL instruction is the same as that of the GP.ECPRTCL instruction for the Ethernet interface module, except for some differences. For the GP.ECPRTCL instruction, refer to the following.

 Q Corresponding Ethernet Interface Module User's Manual (Basic)

The following are the differences.

Item	Differences
Setting data Un	U0 (dummy) needs to be specified. Since this is a dummy, no change is required when replacing from the GP.ECPRTCL instruction of the Ethernet interface module.
Setting data n1	The connection No. set in the "Open Setting" of the "Built-in Ethernet Port Setting" tab need to be specified.
Protocol execution status	The protocol execution status can be checked by "Start Device to Store Predefined Protocol" in the "Open Setting" of the "Built-in Ethernet Port Setting" tab.
Pairing open	The pairing open setting is not available in the CPU module. The CPU module sends and receives data using one connection.
Cancellation of protocol execution	Protocol execution can be cancelled by "Start Device to Store Predefined Protocol" (offset+18) set in the "Open Setting" of the "Built-in Ethernet Port Setting" tab.  Page 55, Section 6.2 (1)

6.5 Execution Conditions of Predefined Protocol Function

The predefined protocol function can be executed when the protocol setting data is enabled and SM1354 (Predefined protocol ready) is on. The written protocol setting data will be enabled when

- the CPU module is powered on or is reset, or
- SM1355 (Predefined protocol setting check request) is turned on

SM1354 is used as an interlock signal to execute the protocol.

(1) Operation of SM1354

SM1354 (Predefined protocol ready) is turned on when the protocol setting data is normal, and remains off when the protocol setting data is abnormal or not written.

If the protocol setting data is abnormal, the following occurs.

- The error details are stored in SD1359 to SD1362 (Predefined protocol setting data error information).
- The error code is stored in SD1381 (Predefined protocol function error code).

Registration of the protocol setting data can be checked in the following special register areas.

- SD1363 (Number of protocols registered)
- SD1365 to SD1372 (Protocol registration status)

(2) Changing the protocol setting data without powering off or resetting the CPU module

The protocol setting data can be changed by turning on SM1355 (Predefined protocol setting check request).

After SM1355 turns on, the following occurs.

- SM1354 (Predefined protocol ready) turns off.
- If the protocol setting data is normal, SM1354 turns on, and SM1355 turns off.
- If the protocol setting data is abnormal, SM1354 does not turn on, and SM1355 turns off. The error information is stored in SD1359 to SD1362 (Predefined protocol setting data error information) and SD1381 (Predefined protocol function error code).

6.6 Operation Image and Data Structure of the Predefined Protocol Function

For operation image and data structure of the predefined protocol function, refer to the following. (Read the E71 (Ethernet interface module) as the CPU module.)

 Q Corresponding Ethernet Interface Module User's Manual (Basic)

6.7 Precautions

(1) Send/receive packet

- When a message larger than 1460 bytes is sent from the external device using TCP, a packet may be divided. As a result, data may not be received as one message with the predefined protocol function. Keep the message less than 1460 bytes or use UDP.
- When data is received using TCP, separate data blocks that are sent successively may be combined at the receiving side. Therefore, handshaking is required prior to communications.

(2) Predefined protocol setting

Power on or reset the CPU module, or turn on SM1355 (Predefined protocol setting check request) to enable the written protocol setting data. If the written data are not enabled, the data may be changed unintentionally when the CPU module is powered on or is reset.

(3) Device data

Data set by the predefined protocol function are communicated during the service processing. Therefore, if the COM instruction is executed while b15 of SD778 (Refresh processing selection when the COM/CCOM instruction is executed) is on, data communications are also performed at the execution of the COM instruction.

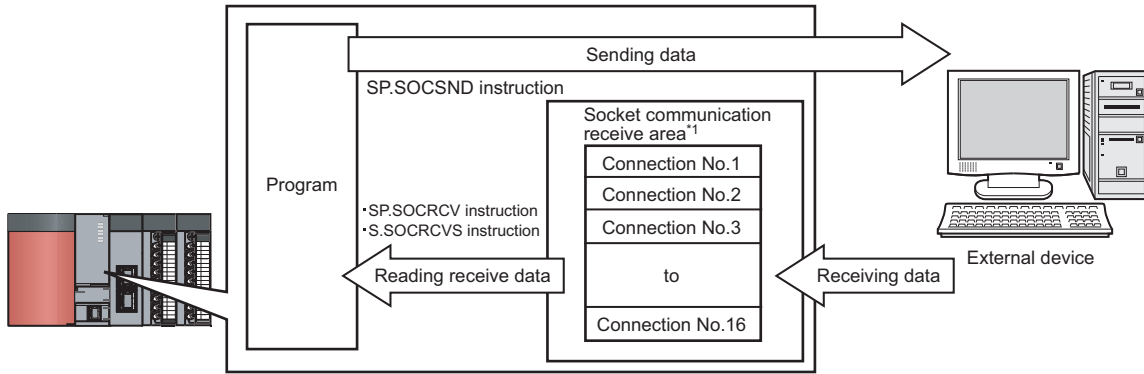
(4) Connection with different open systems

Even the connection for which the open system is set to "Socket Communication" in parameter can be reopened as the predefined protocol after the connection is closed. Similarly, the connection for which the open system is set to "Predefined Protocol" in parameter can be reopened as the socket communication after the connection is closed.

However, if the connection for which the open system is set to "Socket Communication" in parameter is reopened as the predefined protocol after completion of closing, the status storage and protocol cancellation are not possible since the start device to store the predefined protocol operation status cannot be specified.



CHAPTER 7 SOCKET COMMUNICATION FUNCTION Note 7.1

The socket communication function allows data communications with the devices on Ethernet by TCP or UDP using various dedicated instructions.




*1 The area is used for storing data received from the connected open devices.

Remark

- For dedicated instructions used for the socket communication function, refer to:  Page 79, Section 7.4
- Access through routers is also available (except for simultaneous broadcast). When configuring the settings for it, set the subnet mask pattern and default router IP address. ( Page 30, Section 3.4)

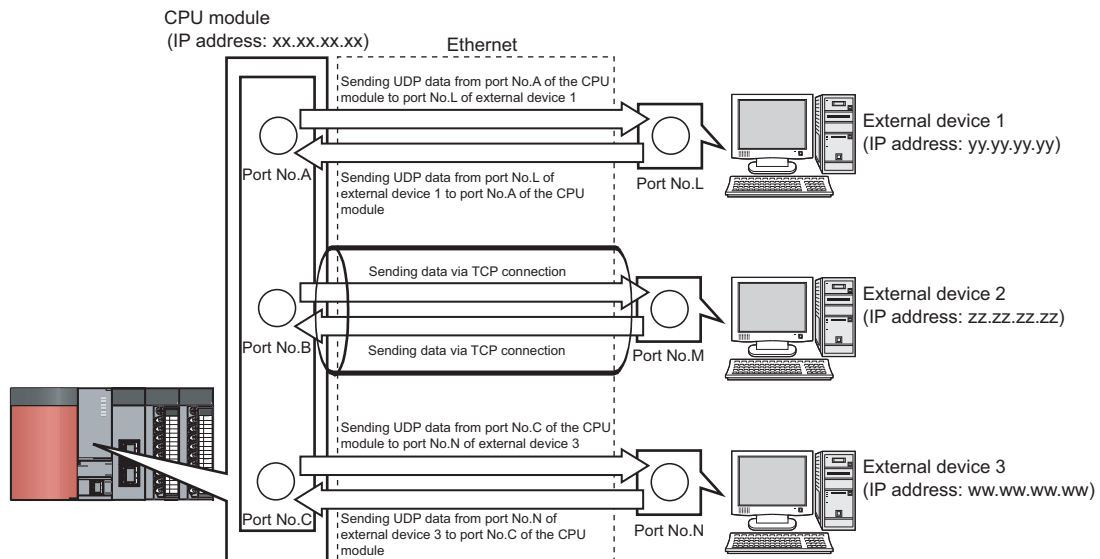
Note 7.1 Universal

Before using the socket communication function for the QnUDE(H)CPU, check the versions of the CPU module and programming tool. ( Page 155, Appendix 3)

(1) Port numbers

In socket communication, port numbers are used to identify respective communications and thereby multiple communications are available both on TCP and UDP.

- For sending
 - Specify the port number of the CPU module from which data are sent, and the port number of the destination device.
- For receiving
 - Specify the port number of the CPU module, and read out the data sent to the port.



7.1 Communication Using TCP

TCP (Transmission Control Protocol) establishes a connection to a device with a port number, and performs reliable data communications.

To perform socket communication using TCP, confirm the following in advance.

- IP address and port number of the target device
- IP address and port number of the CPU module
- Which side will open a connection, the target device or CPU module?
(Active open or Passive open)

(1) TCP connection

There are two types of open operation for TCP connection: Active open and Passive open.

Firstly, the device waiting for a TCP connection performs a Passive open at the specified port.

The other device performs an Active open by specifying the port number of the device which is waiting in Passive open state.

Through the above process, a TCP connection is established and communication is available.

(a) Active open

Active open is a TCP connection method, which actively opens a connection to the device that is passively waiting for a TCP connection.

(b) Passive open

The following two types of Passive open methods are available for TCP connection.

Type of TCP connection	Description
Unpassive	Allows a connection regardless of the IP address and port number of the connected device. (The IP address and port number of the device connected can be acquired using the SP.SOCCINF instruction.)
Fullpassive	Allows a connection to the device only when the specified IP address and port number are met. A connection made by another device that does not have the specified IP address and port number is automatically disconnected before communication.

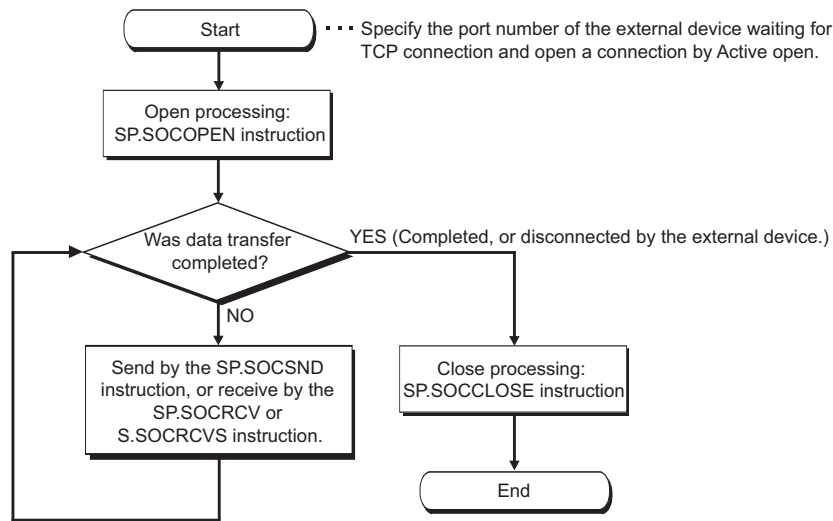
Remark

The expressions of Active and Passive opens may vary according to the device.

- Active open: TCP connection initiating device, client, connecting side, etc.
- Passive open: TCP connection waiting device, server, listening side, etc.

(2) Program example for Active open

The following shows a communication flow of an Active open.

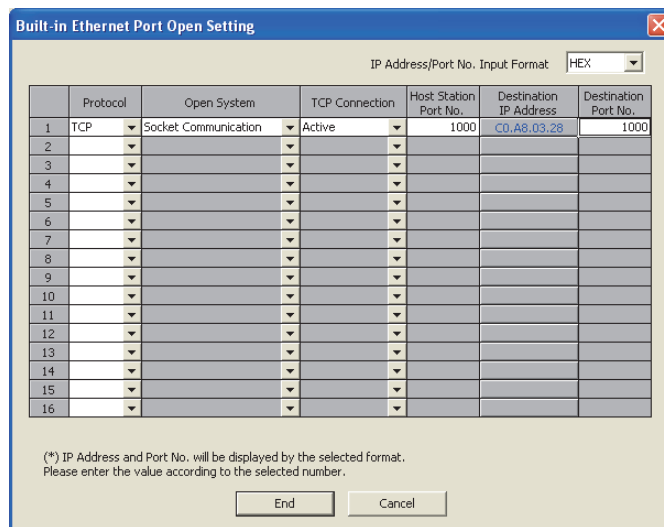


(a) Parameter setting

The following parameters are set for the sample program.

Project window ⇨ [Parameter] ⇨ [PLC Parameter] ⇨ [Built-in Ethernet Port Setting]

⇨ button



Item	Setting
Protocol	TCP
Open System	Socket Communication
TCP Connection	Active
Host Station Port No.	1000 _H (Setting range: 0001 _H to 1387 _H , 1392 _H to FFFE _H) ^{*1}
Destination IP Address	192.168.3.40 (Setting range: 0.0.0.1 to 223.255.255.254)
Destination Port No.	1000 _H (Setting range: 0001 _H to FFFE _H)

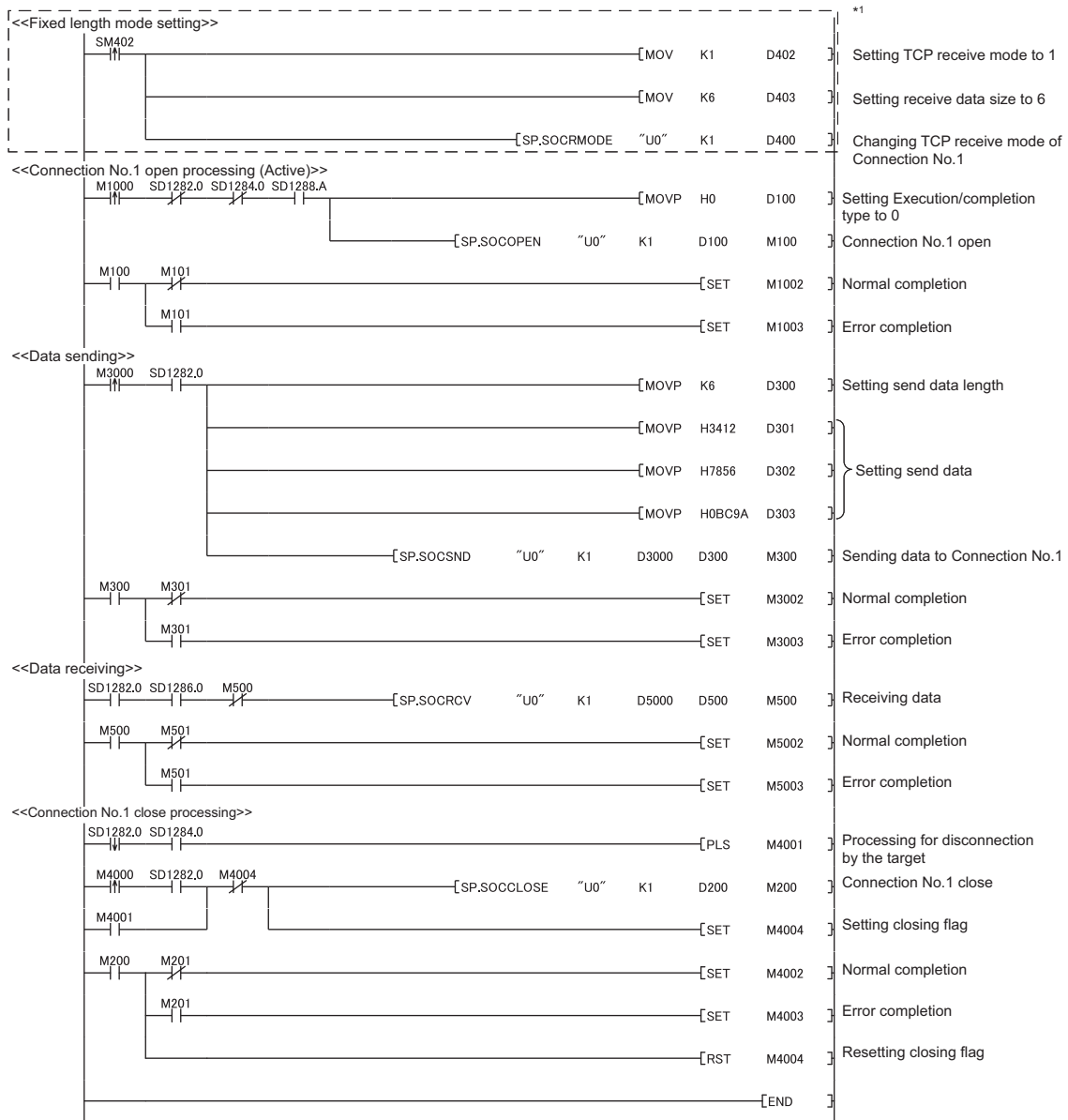
*1 Do not specify 1388_H to 1391_H because these ports are used by the system. (☞ Page 154, Appendix 2)

(b) Devices used in the program

The following table lists the device numbers and applications used in the sample program.

Device number	Application
M1000	Open direction
D100 to D109	SP.SOCOPEN instruction control data
M100 and M101	SP.SOCOPEN instruction completion device
M1002	Normal open indication
M1003	Open error indication
M3000	Send direction
D3000 and D3001	SP.SOCSND instruction control data
M300 and M301	SP.SOCSND instruction completion device
D300 to D303	Send data length and send data (6 bytes of 12 _H , 34 _H , 56 _H , 78 _H , 9A _H , and BC _H)
M3002	Normal send indication
M3003	Send error indication
M4000	Close direction
M4001	Disconnection by the other device
SD1282	Open completion signal
SD1284	Open request signal
SD1286	Receive state signal
SD1288	Connection state signal
D200 and D201	SP.SOCCLOSE instruction control data
M200 and M201	SP.SOCCLOSE instruction completion device
M4002	Normal close indication
M4003	Close error indication
M4004	Closing flag
D400 to D403	SP.SOCRMODE instruction control data
D5000 and D5001	SP.SOCRVC instruction control data
M500 and M501	SP.SOCRVC instruction completion device
D500 and higher	Receive data length and receive data
M5002	Normal receive indication
M5003	Receive error indication

(c) Sample program

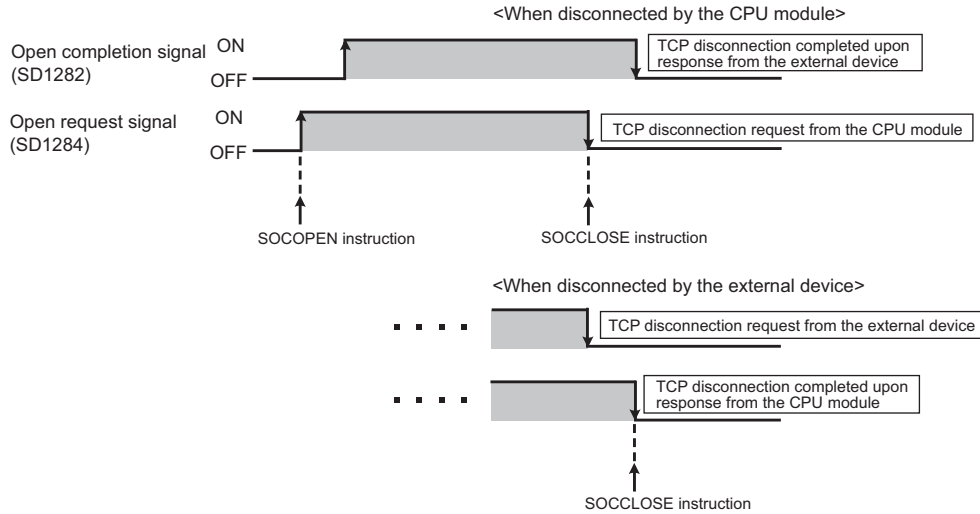


*1 There are two kinds of TCP receive modes: TCP standard receive mode and TCP fixed-length receive mode. For fixing the data size, run the program enclosed by a dotted line. (It can be omitted when the data size is not fixed.)
 For the TCP receive mode, refer to the section for the SP.SOCRMODE instruction. (Page 103, Section 7.4.8)

(d) Precaution for Active open communication

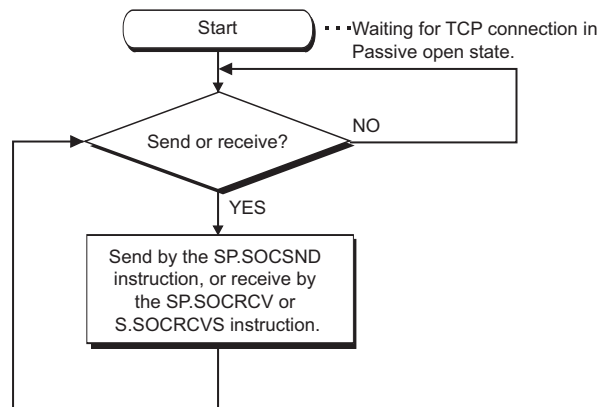
Configure an interlock circuit using the Open completion signal (SD1282) and Open request signal (SD1284) in the program.

The following chart shows on/off timings of the Open completion signal and Open request signal.




(3) Program example for Passive open

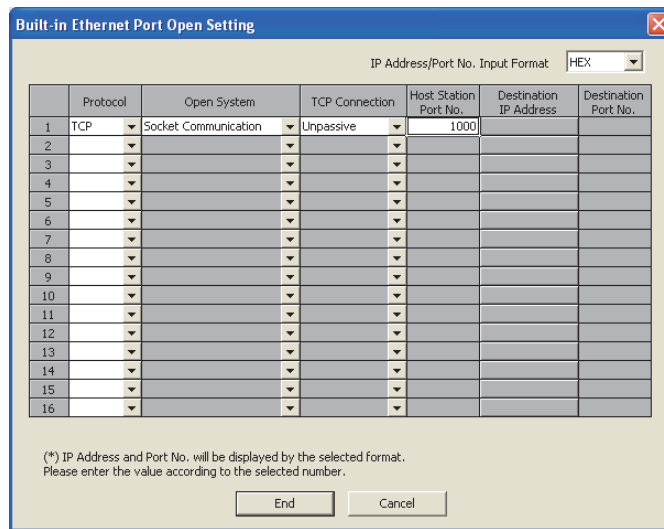
The following shows a communication flow of a Passive open.



(a) Parameter setting

The following parameters are set for the sample program.

Project window ⇨ [Parameter] ⇨ [PLC Parameter] ⇨ [Built-in Ethernet Port Setting] ⇨  button



Item	Setting
Protocol	TCP
Open System	Socket Communication
TCP Connection	Unpassive
Host Station Port No.	1000 _H (Setting range: 0001 _H to 1387 _H , 1392 _H to FFFE _H) ^{*2}
Destination IP Address	Blank ^{*1} (Setting range: 0.0.0.1 to 223.255.255.254)
Destination Port No.	Blank ^{*1} (Setting range: 0001 _H to FFFE _H)

*1 When "Fullpassive" is selected for "TCP Connection", a value must be set.

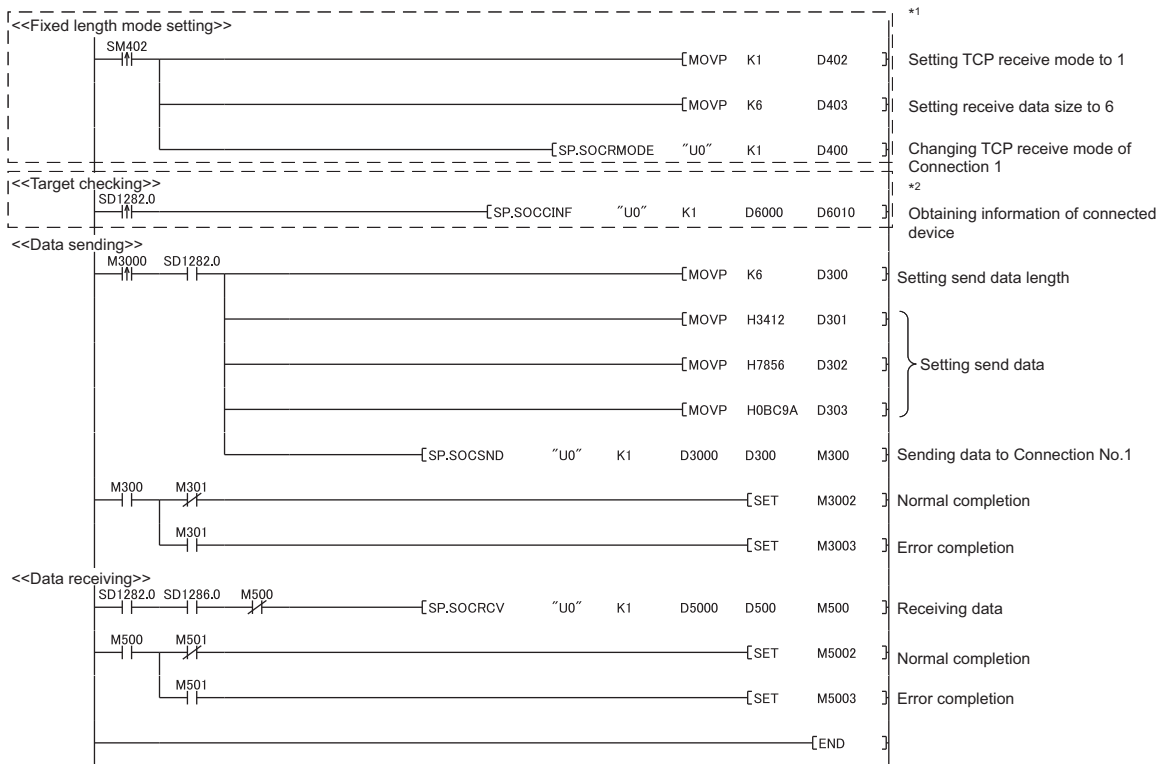
*2 Do not specify 1388_H to 1391_H because these ports are used by the system. (☞ Page 154, Appendix 2)

(b) Devices used in the program

The following table lists the device numbers and applications used in the sample program.

Device number	Application
M3000	Send direction
D3000 and D3001	SP.SOCSND instruction control data
M300 and M301	SP.SOCSND instruction completion device
D300 to D303	Send data length and send data (6 bytes of 12 _H , 34 _H , 56 _H , 78 _H , 9A _H , and BC _H)
M3002	Normal send indication
M3003	Send error indication
D400 to D403	SP.SOCRMODE instruction control data
SD1282	Open completion signal
SD1286	Receive state signal
D5000 and D5001	SP.SOCRCV instruction control data
M500 and M501	SP.SOCRCV instruction completion device
D500 and higher	Receive data length and receive data
M5002	Normal receive indication
M5003	Receive error indication
D6000 and D6001	SP.SOCCINF instruction control data
D6010 to D6014	SP.SOCCINF instruction completion device

(c) Sample program

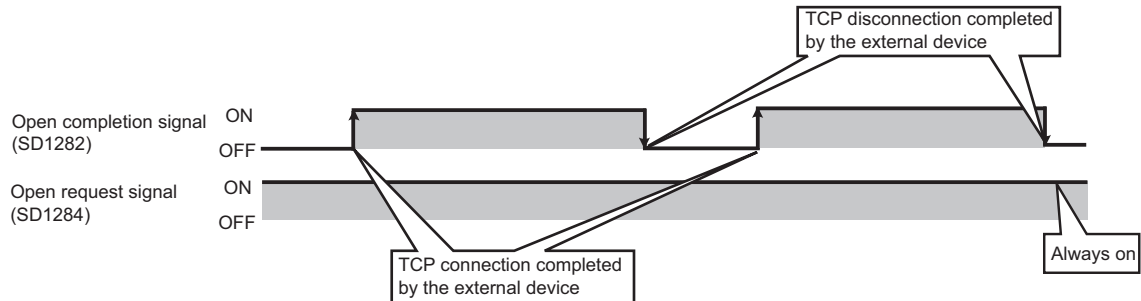




- *1 There are two kinds of TCP receive modes: TCP standard receive mode and TCP fixed-length receive mode. For fixing the data size, run the program enclosed by a dotted line. (It can be omitted when the data size is not fixed.) For the TCP receive mode, refer to the section for the SP.SOCRMODE instruction. (Page 103, Section 7.4.8)
- *2 For acquiring information of the device connected on TCP, run the program enclosed by a dotted line. (It can be omitted when the information acquisition is not needed.)

(d) Precaution for Passive open communication

- Configure an interlock circuit using the Open completion signal (SD1282) and Open request signal (SD1284) in the program.

The following chart shows on/off timings of the Open completion signal and Open request signal.



- When a device establishes a connection by Passive open, the IP address and port number of the connected device can be acquired using the SP.SOCCINF instruction.
For the SP.SOCCINF instruction, refer to:  Page 98, Section 7.4.6
- On TCP, one connection is established with one target device.
To communicate with multiple devices from one port number, prepare the same number of connections as that of the target devices.
A connection that exceeds the preset number of connections will be disconnected immediately.
- Do not accept a connection from a device until the CPU module is placed in the wait-for-open state.
If a TCP connection request is received before entering the wait-for-open state after completion of CPU startup, the request will be recognized as an error, and a forced close message for the connection will be returned to the interfacing device.
In this case, wait until the CPU state is changed to the wait-for-open state and then retry the connection from the device.
- Do not execute the SP.SOCCLOSE instruction in a program.
Doing so will disable data transfer since the Open completion signal and Open request signal of the corresponding connection turn off for close processing.
To reopen a closed connection, execute the SP.SOCOPEN instruction.
For the SP.SOCOPEN instruction, refer to:  Page 80, Section 7.4.1

7.2 Communication Using UDP

UDP (User Datagram Protocol) is a simple protocol that does not perform data sequencing and retransmission. To perform socket communication using UDP, confirm the following in advance.

- IP address and port number of the target device
- IP address and port number of the CPU module

(1) Program example

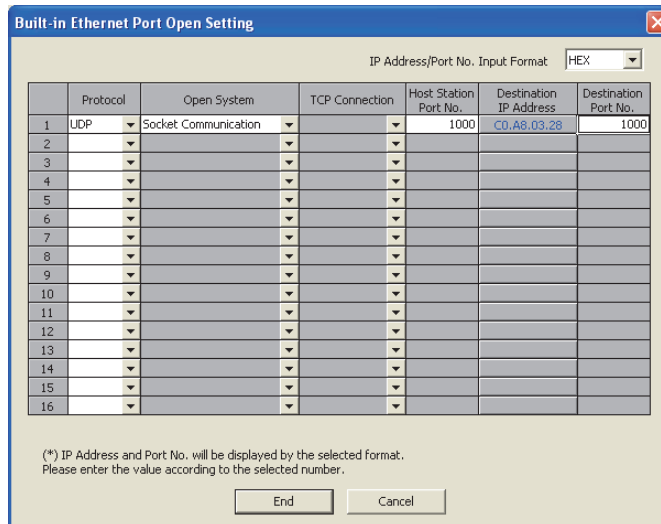
This section provides a program example for communication using UDP.

(a) Parameter setting

The following parameters are set for the sample program.

Project window ⇒ [Parameter] ⇒ [PLC Parameter] ⇒ [Built-in Ethernet Port Setting]

⇒  button



	Protocol	Open System	TCP Connection	Host Station Port No.	Destination IP Address	Destination Port No.
1	UDP	Socket Communication		1000	C0.A8.03.28	1000
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						

(*) IP Address and Port No. will be displayed by the selected format.
Please enter the value according to the selected number.

Item	Setting
Protocol	UDP
Open System	Socket Communication
TCP Connection	Blank
Host Station Port No.	1000 _H (Setting range: 0001 _H to 1387 _H and 1392 _H to FFFE _H)* ¹
Destination IP Address	192.168.3.40 (Setting range: 0.0.0.1 to 223.255.255.254/255.255.255.255)
Destination Port No.	1000 _H (Setting range: 0001 _H to FFFE _H /FFFF _H)

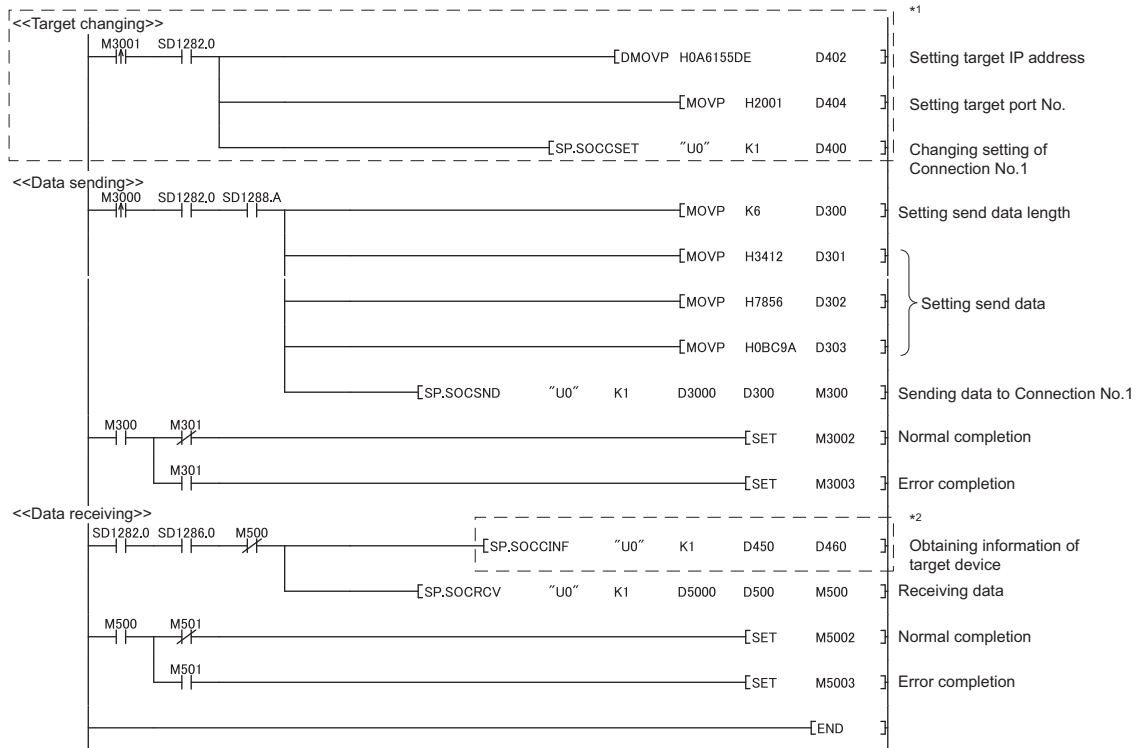
*1 Do not specify 1388_H to 1391_H because these ports are used by the system. (☞ Page 154, Appendix 2)

(b) Devices used in the program

The following table lists the device numbers and applications used in the sample program.

Device number	Application
M3000	Send direction
D3000 and D3001	SP.SOCSND instruction control data
M300 and M301	SP.SOCSND instruction completion device
D300 and D303	Send data length and send data (6 bytes of 12 _H , 34 _H , 56 _H , 78 _H , 9A _H , and BC _H)
M3002	Normal send indication
M3003	Send error indication
D5000 to D5001	SP.SOCRCV instruction control data
M500 to M501	SP.SOCRCV instruction completion device
SD1282	Open completion signal
SD1286	Receive state signal
SD1288	Connection state signal
M3001	Target change direction
D500 and higher	Receive data length and receive data
M5002	Normal receive indication
M5003	Receive error indication
D400 to D404	SP.SOCCSET instruction control data
D450 to D451	SP.SOCCINF instruction control data
D460 to D464	SP.SOCCINF instruction connection information

(c) Sample program

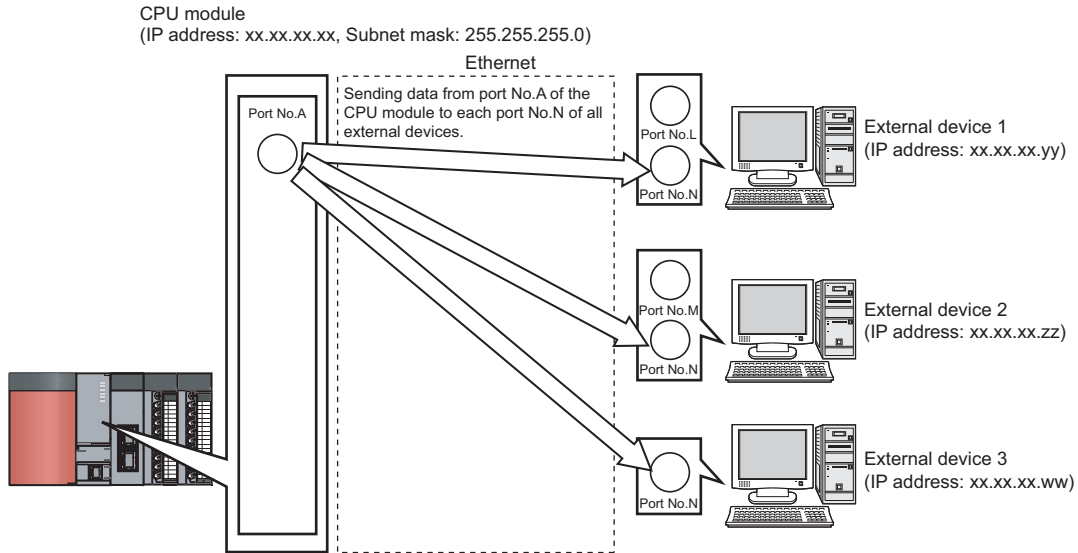


- *1 For changing the target device, run the program enclosed by a dotted line. (It can be omitted when the communication target is not changed.) For details, refer to the section of the SP.SOCCSET instruction. (Page 101, Section 7.4.7)
- *2 For acquiring information of the target device connected on UDP, run the program enclosed by a dotted line. (It can be omitted when the information acquisition is not needed.)

(2) Simultaneous broadcast

For simultaneous broadcast using UDP, set the parameters as follows.

- Destination IP Address: $FF.FF.FF.FF_H$
- Destination Port No.: $FFFF_H$



Destination IP Address	Destination Port No.	Receiving	Sending
Other than $FF.FF.FF.FF_H$	Other than $FFFF_H$	Receives only the data sent from the specified IP address and port No. among the entire data sent to the host station port No.	Sends data from the host station port No. to the specified IP address and port No.
Other than $FF.FF.FF.FF_H$	$FFFF_H$	Receives the data sent from all ports of the specified IP address among the entire data sent to the host station port No.	N/A
$FF.FF.FF.FF_H$	Other than $FFFF_H$	Receives the data sent from the specified port No. of all the IP addresses among the entire data sent to the host station port No.	Sends data to the port No. specified in the settings for simultaneous broadcast.
$FF.FF.FF.FF_H$	$FFFF_H$	Receives all data that have sent to the host station port No.	N/A

(3) Precautions


(a) Use of UDP

Data may be lost, or may arrive out of order.
Consider the application of TCP if any problem is expected.

(b) Sending and receiving data

Data sending process may be terminated even if the communication line between the CPU module and target device is not connected due to a reason such as cable disconnection.
To avoid this, it is recommended to provide communication procedures at the user's discretion.


(c) Changing the target

Use the SP.SOCCSET instruction to change the target device.
For the SP.SOCCSET instruction, refer to:  Page 101, Section 7.4.7

(d) Open completion signal and Open request signal

Once UDP is selected for a connection, the Open completion signal and Open request signal of the connection are always on.


(e) SP.SOCCLOSE instruction

Do not execute the SP.SOCCLOSE instruction in the program.
Doing so will disable data transfer since the Open completion signal and Open request signal of the corresponding connection turn off for close processing.
To reopen the closed connection, execute the SP.SOCOPEN instruction.
For the SP.SOCOPEN instruction, refer to:  Page 80, Section 7.4.1

(f) Simultaneous broadcast targets

With simultaneous broadcast, data can be sent to the devices which are connected to the same hub of the CPU module, and to those connected to the cascaded hub(s).
Data cannot be received from the devices connected through routers.

(g) Receiving data using simultaneous broadcast

When data are received through a connection of simultaneous broadcast, the IP address and port number of the sender can be acquired by the SP.SOCCINF instruction.
For the SP.SOCCINF instruction, refer to:  Page 98, Section 7.4.6

(h) Connection of simultaneous broadcast

Data cannot be sent when $FFFF_H$ is specified for the port number of the transmission target.
To send data, specify a value other than $FFFF_H$.

(i) Destination IP address of the message transferred by simultaneous broadcast

Use a CPU module IP address of which all the bits corresponding to the host address are on.

When the subnet mask pattern is specified, apply the pattern before using the above mentioned IP address.

Ex.	IP address of the CPU module side	:	64.	168.	3.	39
	Subnet mask pattern	:	None			
	IP address of simultaneous broadcast	:	64.	255.	255.	255

Ex.	IP address of the CPU module side	:	64.	168.	3.	39
	Subnet mask pattern	:	255.	255.	255.	0
	IP address of simultaneous broadcast	:	64.	168.	3.	255

7.3 Precautions for the Socket Communication Function

This section provides other precautions for the socket communication function.

(1) Port number

Host station port number, 0001_H to 03FF_H, are assigned for reserved port numbers (WELL KNOWN PORT NUMBERS) and F000_H to FFFE_H are for other communication functions. Therefore, using 0400_H to 1387_H and 1392_H to EFFF_H is recommended.

Do not specify 1388_H to 1391_H because these ports are used by the system. (☞ Page 154, Appendix 2)

Do not specify 0014_H and 0015_H for the socket communication function when using the FTP function.

Do not specify 007B_H for the socket communication function when using the time setting function (SNTP).

Do not specify F000_H to FFFE_H for the socket communication function when using the data logging file transfer function.

(2) Reading out received data

Read out received data when the Receive state signal (SD1286) has turned on.

Communication via the built-in Ethernet port may be affected if a considerable amount of received data have not been read out for a long time.

(3) Conditions for closing

In TCP communications, even if no close request is sent from the connected device, the Open completion signal will turn off to close the connection in the following cases.

- Alive check is timed out.
- Forced close is received from the connected device.

(4) Elements for TCP connection

The following four elements control TCP connections, and only one connection can be established with a unique setting for these elements. To use multiple TCP connections at the same time, at least one of the four elements must be different.

- IP address of the CPU module
- Port number of the CPU module
- IP address of the target device
- Port number of the target device

(5) Reestablishment of the same connection

Allow eight seconds or more before reestablishing a connection of the same target IP address, the same host station port number, and the same target port number after closing it.

If the reestablishment is time-critical, it is recommended to change the host station port number on the Active open side.

(6) Precautions for file access during communication

The CPU module will perform file access processing prior to Ethernet communication processing. Because of this, processing of the socket communication function may be delayed if a file is accessed by FTP or a programming tool during the processing.

When accessing a file while response time monitoring is performed on the connected device with the socket communication function, add the time required for file access to the monitoring time.

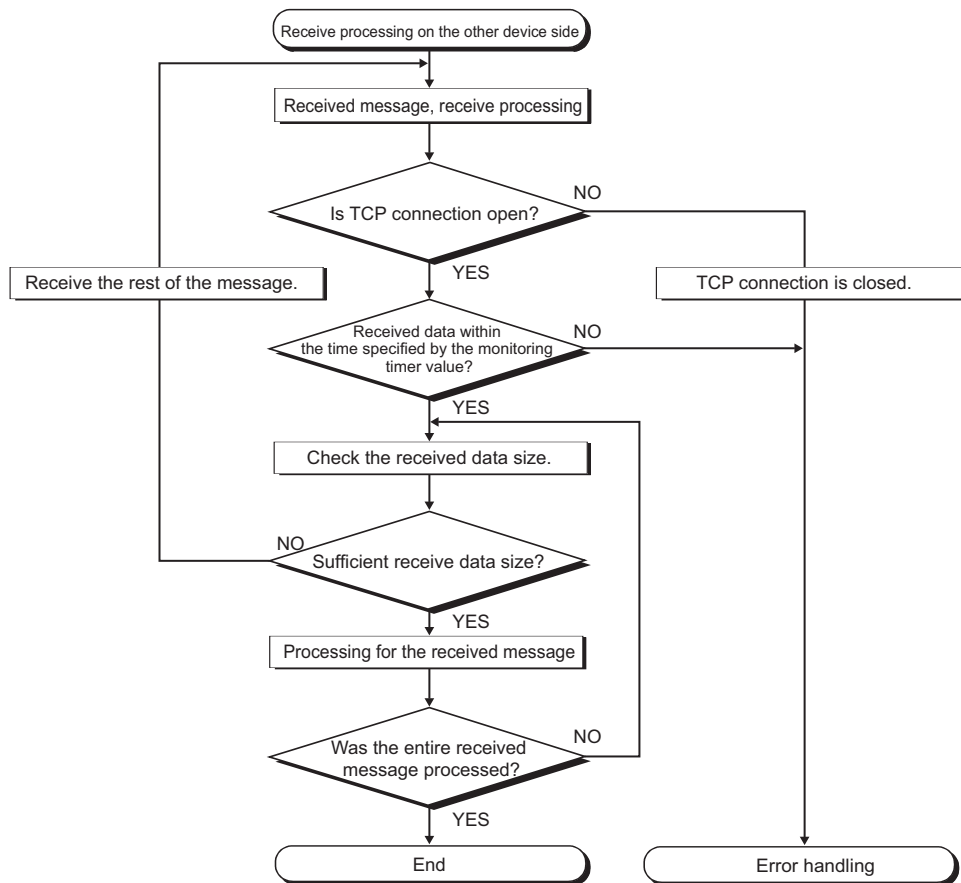
(7) Checking receive data length

Since no delimiter is provided for TCP communication data, on the receiving end, separate data blocks that are sent continuously may be combined, or data sent all at once may be segmented.

The receive data length must be confirmed on the receiving side as necessary.

When data are received on the CPU side and the data length is determined, the fixed-length mode is recommended.

When receiving data on the target device side, confirm the receive data length as shown below.



(8) If an error (error code: 41A0_H) has occurred

In TCP communications, if an error (error code: 41A0_H) occurs at the sender, part of send data may have been sent. Therefore, if the data are sent again after the error (error code: 41A0_H), close the connection to discard the data. Then open a connection again, and send the data again.

7.4 Socket Communication Function Instructions



The socket communication function instructions are provided for the CPU module to use the socket communication function.

This section explains the socket communication function instructions.

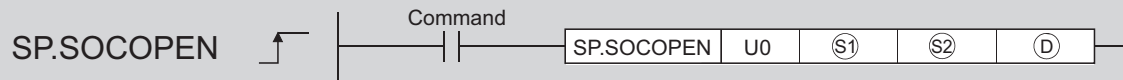
The following is a list of the instructions.

Instruction	Description	Reference
SP.SOCOPEN	Establishes a connection.	Page 80, Section 7.4.1
SP.SOCCLOSE	Closes a connection.	Page 84, Section 7.4.2
SP.SOCRCV	Reads out received data (in END processing).	Page 87, Section 7.4.3
S.SOCRCVS	Reads out received data (upon instruction execution).	Page 91, Section 7.4.4
SP.SOCSND	Sends data.	Page 94, Section 7.4.5
SP.SOCCINF	Reads out connection information.	Page 98, Section 7.4.6
SP.SOCCSET	Changes the target of the connection for UDP/IP communication.	Page 101, Section 7.4.7
SP.SOCRMODE	Changes receive mode of the connection.	Page 103, Section 7.4.8
S(P).SOCRDATA	Reads out data from the socket communication receive data area.	Page 107, Section 7.4.9

Point

- For configuration of data communication using the socket communication function, refer to:  Page 62, Section 7.1, Page 71, Section 7.2
- If the instruction has a completion device, do not change any data, such as control data and request data, that are specified for the instruction until the execution of the instruction is completed.
- Do not execute any socket communication function instruction in an interrupt program.
- For error codes, refer to the following.
 QCPU User's Manual (Hardware Design, Maintenance and Inspection)

7.4.1 Establishing a connection (SP.SOCOPEN)



Setting data	Internal device		R, ZR	J□\□		U□\G□	Zn	Constant K, H	Others
	Bit	Word		Bit	Word				
S1	-	○	○			-		○	-
S2	-	△*1	△*1			-		-	-
D	△*1	-	△*1			-		-	-

*1 File registers set for each local device or program cannot be used.

(1) Setting data

Setting data	Description	Set by*2	Data type
U0	Dummy	-	Character string
S1	Connection number (Setting range: 1 to 16)	User	BIN 16-bit
S2	Start number of the device from which control data are stored	User, system	Device name
D	Start number of the device which turns on for one scan upon completion of the instruction D+1 also turns on when failed.	System	Bit

*2 The "Set by" column indicates the following.

User: The data must be set before executing the SP.SOCOPEN instruction.

System: The CPU module stores the execution result of the SP.SOCOPEN instruction.

Point

When replacing the ZP.OPEN instruction (Ethernet module dedicated instruction), dummy argument can be used in a Built-in Ethernet port QCPU instruction without rewriting.

(2) Control data

Device	Item	Description	Setting range	Set by ^{*3}
Ⓢ2+0	Execution/completion type	Specify which settings are used to open a connection, parameter settings configured by a programming tool or control data settings Ⓢ2+2 to Ⓢ2+9. 0000 _H : Connection is opened according to the settings set in "Open settings" of PLC parameter. 8000 _H : Connection is opened according to the values specified for control data Ⓢ2+2 to Ⓢ2+9.	0000 _H 8000 _H	User
Ⓢ2+1	Completion status	Completion status is stored 0000 _H : Completed Other than 0000 _H : Failed (Error code)	-	System
Ⓢ2+2	Application setting area	<div style="text-align: center;"> </div> <p>[1] Communication method (protocol) 0: TCP/IP 1: UDP/IP [2] Socket communication and predefined protocol procedure 1: Nonprocedural (fixed) [3] Predefined protocol setting 0: Predefined protocol function is not used. (Socket communication function is used.) 1: Predefined protocol function is used. [4] Open system 00: Active open or UDP/IP 10: Unpassive open 11: Fullpassive open</p>	As described in the left column	User
Ⓢ2+3	Host station port number	Specify the port number of the host station.	1 _H to 1387 _H 1392 _H to FFFE _H	
Ⓢ2+4 Ⓢ2+5	Target device IP address ^{*4}	Specify the IP address of the target device.	1 _H to FFFFFFFF _H (FFFFFFF _H : Simultaneous broadcast)	
Ⓢ2+6	Target device port number ^{*4}	Specify the port number of the target device.	1 _H to FFFF _H (FFFF _H : Simultaneous broadcast)	
Ⓢ2+7 to Ⓢ2+9	-	Use prohibited	-	System

*3 The "Set by" column indicates the following.

User: The data must be set before executing the SP.SOCOPEN instruction.

System: The CPU module stores the execution result of the SP.SOCOPEN instruction.

*4 For the Unpassive open, the IP address and port number of the target device are ignored.

*5 Using 0400_H to 1387_H and 1392_H to EFFF_H is recommended because the host station port numbers, 0001_H to 03FF_H, are assigned for generally reserved port numbers (WELL KNOWN PORT NUMBERS), and F000_H to FFFE_H are used for other communication functions. Do not specify 1388_H to 1391_H because these ports are used by the system. (Page 154, Appendix 2)

(3) Function

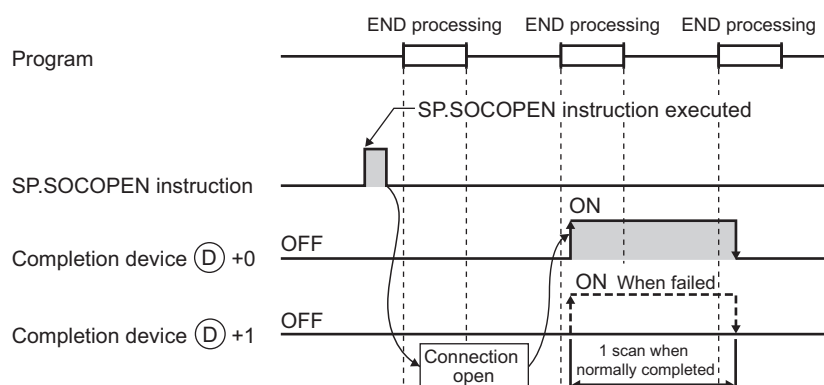
This instruction opens a connection specified in S1 .

The set values used for the open processing is selected in $\text{S2}+0$.

The result of the SP.SOCOPEN instruction can be checked with the completion device, $\text{D}+0$ and $\text{D}+1$.

- Completion device $\text{D}+0$
Turns on in the END processing of the scan after completion of the SP.SOCOPEN instruction, and turns off in the next END processing.
- Completion device $\text{D}+1$
Turns on or off according to the result of the SP.SOCOPEN instruction.

State	Description
When completed	Remains off.
When failed	Turns on in the END processing of the scan after the SP.SOCOPEN instruction is completed and turns off in the next END processing.



- A connection with no parameters (no protocol is specified) can be opened. In this case, specify 8000_H for $\text{S2}+0$ and configure open settings in $\text{S2}+2$ to $\text{S2}+9$.

(4) Error

A detection of an operation error turns on the Error flag (SM0) and a corresponding error code is stored in SD0 when:

- The instruction is executed in the QnUDE(H)CPU with a serial number (first five digits) of "11011" or earlier or a CPU module other than the Built-in Ethernet port QCPU. (Error code: 4002)
- The connection number specified for S1 is other than 1 to 16. (Error code: 4101)
- The device numbers specified for S2 and D exceed the device point range. (Error code: 4101)
- An invalid device is specified. (Error code: 4004)

(5) Program examples

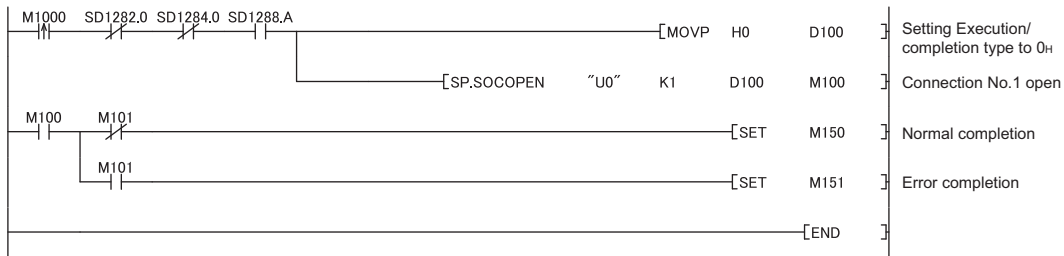
(a) Opening a connection using parameter settings

When M1000 is turned on, connection No.1 is opened using the parameters set in "Open settings" of PLC parameter.

- Devices used

Device number	Application
SD1282	Open completion signal
SD1284	Open request signal
SD1288	Connection state signal
D100	SP.SOCOPEN instruction control data
M100	SP.SOCOPEN instruction completion device

- Program



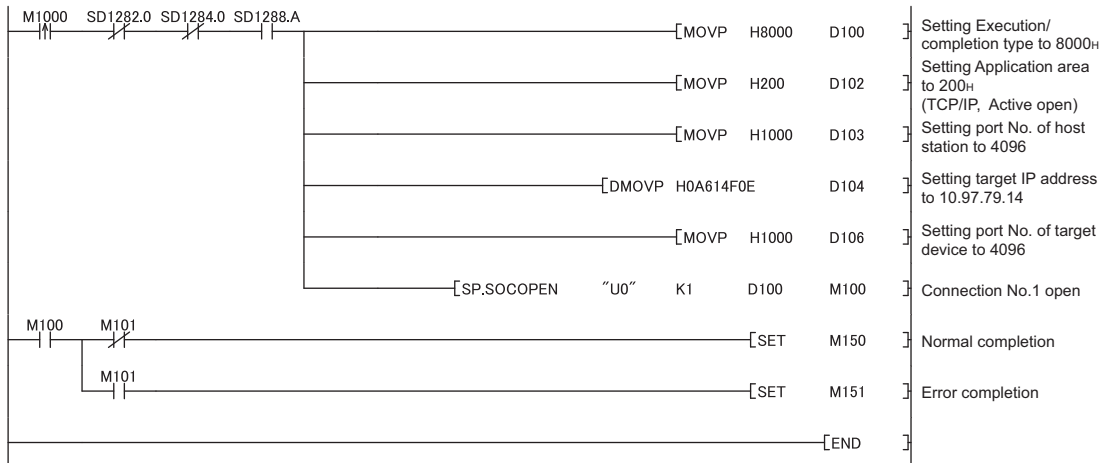
(b) Opening a connection using control data settings

When M1000 is turned on, connection No.1 is opened using control data.

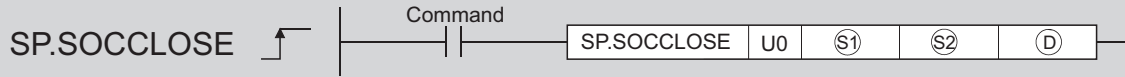
- Devices used

Device number	Application
SD1282	Open completion signal
SD1284	Open request signal
SD1288	Connection state signal
D100	SP.SOCOPEN instruction control data
M100	SP.SOCOPEN instruction completion device

- Program



7.4.2 Disconnecting a connection (SP.SOCCLOSE)



Setting data	Internal device		R, ZR	J□\□		U□\G□	Zn	Constant K, H	Others
	Bit	Word		Bit	Word				
Ⓢ1	-	○	○			-		○	-
Ⓢ2	-	△*1	△*1			-		-	-
Ⓓ	△*1	-	△*1			-		-	-

*1 File registers set for each local device or program cannot be used.

(1) Setting data

Setting data	Description	Set by*2	Data type
U0	Dummy	-	Character string
Ⓢ1	Connection number (Setting range: 1 to 16)	User	BIN 16-bit
Ⓢ2	Start number of the device from which control data are stored	System	Device name
Ⓓ	Start number of the device which turns on for one scan upon completion of the instruction Ⓓ+1 also turns on when failed.		Bit

*2 The "Set by" column indicates the following.

User: The data must be set before executing the SP.SOCCLOSE instruction.

System: The CPU module stores the execution result of the SP.SOCCLOSE instruction.

Point

When replacing the ZP.CLOSE instruction (Ethernet module dedicated instruction), dummy argument can be used in a Built-in Ethernet port QCPU instruction without rewriting.

(2) Control data

Device	Item	Description	Setting range	Set by*3
Ⓢ2+0	System area	-	-	-
Ⓢ2+1	Completion status	Completion status is stored 0000 _H : Completed Other than 0000 _H : Failed (Error code)	-	System

*3 The "Set by" column indicates the following.

System: The CPU module stores the execution result of the SP.SOCCLOSE instruction.

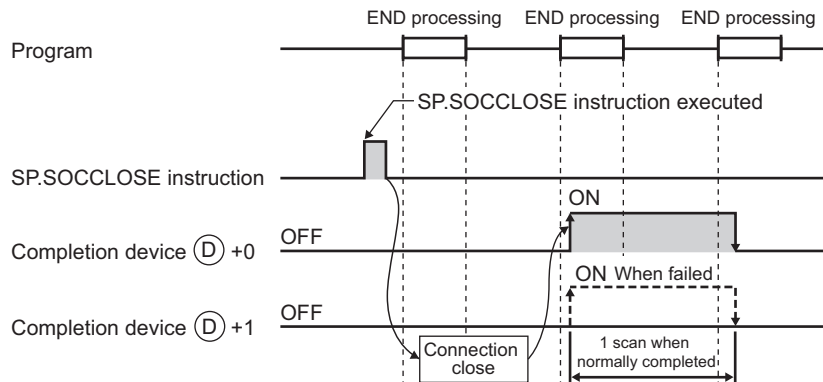
(3) Function

This instruction closes a connection specified in S1 . (Disconnection of a connection)

The result of the SP.SOCCLOSE instruction can be checked with the completion device, $\text{D}+0$ and $\text{D}+1$.

- Completion device $\text{D}+0$
Turns on in the END processing of a scan after completion of the SP.SOCCLOSE instruction, and turns off in the next END processing.
- Completion device $\text{D}+1$
Turns on or off according to the result of the SP.SOCCLOSE instruction.

State	Description
When completed	Remains off.
When failed	Turns on in the END processing of a scan after completion of the SP.SOCCLOSE instruction, and turns off in the next END processing.



(4) Error

A detection of an operation error turns on the Error flag (SM0) and a corresponding error code is stored in SD0 when:

- The instruction is executed in the QnUDE(H)CPU with a serial number (first five digits) of "11011" or earlier or a CPU module other than the Built-in Ethernet port QCPU. (Error code: 4002)
- The connection number specified for S1 is other than 1 to 16. (Error code: 4101)
- The device numbers specified for S2 and D exceed the device point range. (Error code: 4101)
- An invalid device is specified. (Error code: 4004)

Remark
Do not use Passive open for execution of the SP.SOCCLOSE instruction. Doing so will turn off the Open completion signal and Open request signal of the connection and cause close processing, which disables data transfer.
.....

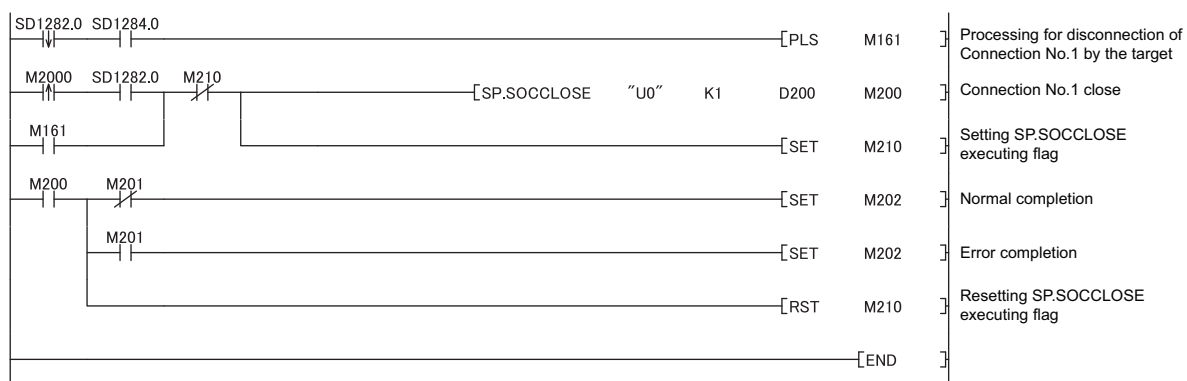
(5) Program example

When M2000 is turned on or when the connected device disconnects connection No.1, connection No.1 is disconnected by the following program.

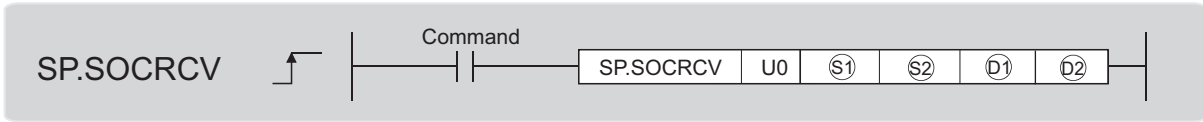
- Devices used

Device number	Application
SD1282	Open completion signal
SD1284	Open request signal
D200	SP.SOCCKLOSE instruction control data
M200	SP.SOCCKLOSE instruction completion device

- Program



7.4.3 Reading out received data in the END processing (SP.SOCRCV)



Setting data	Internal device		R, ZR	J□□		U□\G□	Zn	Constant K, H	Others
	Bit	Word		Bit	Word				
S1	-	○	○			-		○	-
S2	-	△ *1	△ *1			-		-	-
D1	-	△ *1	△ *1			-		-	-
D2	△ *1	-	△ *1			-		-	-

*1 File registers set for each local device or program cannot be used.

(1) Setting data

Setting data	Description	Set by*2	Data type
U0	Dummy	-	Character string
S1	Connection number (Setting range: 1 to 16)	User	BIN 16-bit
S2	Start number of the device from which control data are specified	System	Device name
D1	Start number of the device from which receive data are stored		
D2	Start number of the device which turns on for one scan upon completion of the instruction D2+1 also turns on when failed.		Bit

*2 The "Set by" column indicates the following.

User: The data must be set before executing the SP.SOCRCV instruction.

System: The CPU module stores the execution result of the SP.SOCRCV instruction.

Point

When replacing the ZP.BUFRVCV instruction (Ethernet module dedicated instruction), dummy argument can be used in a Built-in Ethernet port QCPU instruction without rewriting.

(2) Control data

Device	Item	Description	Setting range	Set by ^{*3}
Ⓢ2+0	System area	-	-	-
Ⓢ2+1	Completion status	Completion status is stored. 0000 _H : Completed Other than 0000 _H : Failed (Error code)	-	System
Ⓣ0+0	Receive data length	The length of the data which were read out from the Socket communication receive data area is stored (in bytes).	0 to 10238 ^{*4}	
Ⓣ0+1 to Ⓣ0+n	Receive data	The data which were read out from the Socket communication receive data area are stored in order.	-	

*3 The "Set by" column indicates the following.

System: The CPU module stores the execution result of the SP.SOCRCV instruction.

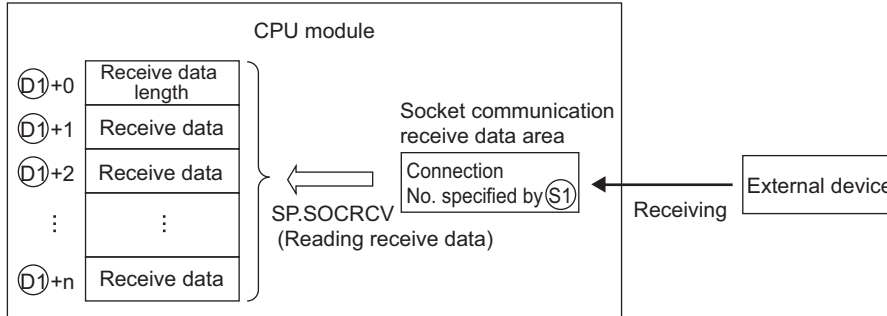
*4 0 to 2046 for the QnUDE(H)CPU with a serial number (first five digits) of "12051" or earlier

Point

- Receive data size is 2046 bytes by default. To receive data over 2046 bytes, change the receive data size with the SP.SOCRMODE instruction.
- When the SP.SOCRCV instruction is executed, data are read from Socket communication receive data area at END processing. Therefore, executing the SP.SOCRCV instruction will increase the scan time.
- If the CPU module receives odd-byte data, invalid data is stored to the high byte of the device where the last received data is stored.

(3) Function

This instruction reads out received data of the connection specified in (S1) from the Socket communication receive data area in the END processing after execution of the SP.SOCRCV instruction.

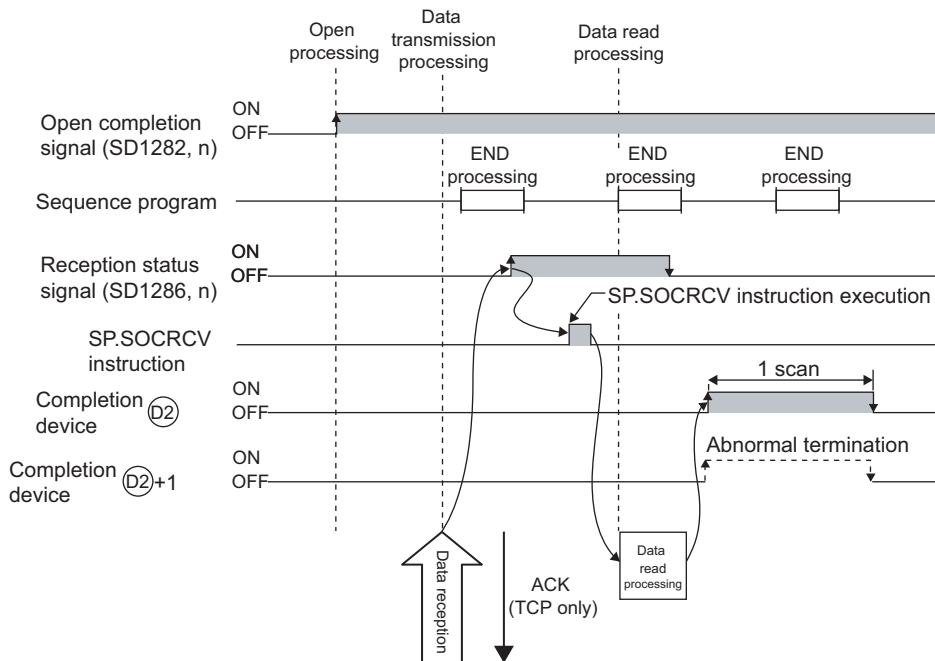


The result of the SP.SOCRCV instruction can be checked with the completion device (D2)+0 and (D2)+1.

- Completion device (D2)+0
Turns on in the END processing of the scan after completion of the SP.SOCRCV instruction, and turns off in the next END processing.
- Completion device (D2)+1
Turns on or off according to the result of the SP.SOCRCV instruction.

State	Description
When completed	Remains off
When failed	Turns on in the END processing of the scan after completion of the SP.SOCRCV instruction, and turns off in the next END processing.

The following figure shows the timing of reception processing with the SP.SOCRCV instruction.



(4) Error

A detection of an operation error turns on the Error flag (SM0) and a corresponding error code is stored in SD0 when:

- The instruction is executed in the QnUDE(H)CPU with a serial number (first five digits) of "11011" or earlier or a CPU module other than the Built-in Ethernet port QCPU. (Error code: 4002)
- The connection number specified for (S1) is other than 1 to 16. (Error code: 4101)
- The received data size exceeds the size of the receive data storage device. (Error code: 4101)
- The device numbers specified for (S2), (D1), and (D2) exceed the device point range. (Error code: 4101)
- An invalid device is specified. (Error code: 4004)

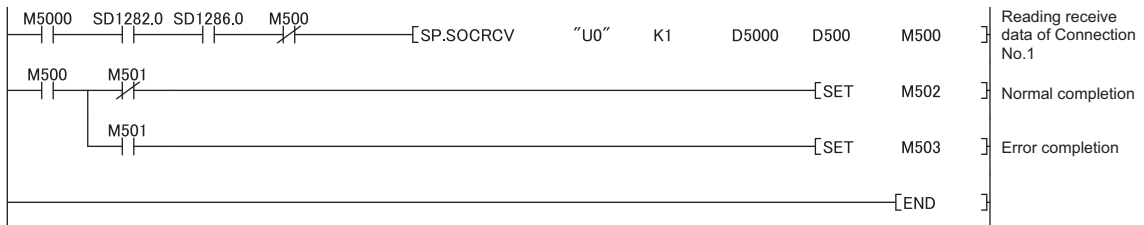
(5) Program example

When M5000 is turned on, data received from the connected device are read out.

- Devices used

Device number	Application
SD1282	Open completion signal
SD1286	Receive state signal
D5000	SP.SOCRCV instruction control data
D500	Receive data length and receive data storage location
M500	SP.SOCRCV instruction completion device

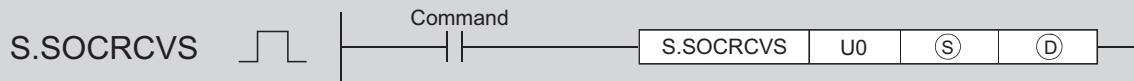
- Program



Point

- To avoid receiving large amounts of data, the volume of received data can be limited by setting the receive data size using the SP.SOCRMODE instruction.
- Consecutively sent data can be consecutively read out by connecting the completion device of the SP.SOCRCV instruction to the execution command at the normally closed contact.

7.4.4 Reading out received data during instruction execution (S.SOCRCVS)



Setting data	Internal device		R, ZR	J□□		U□\G□	Zn	Constant K, H	Others
	Bit	Word		Bit	Word				
Ⓢ	-	○	○			-		○	-
Ⓓ	-	○	○			-		-	-

(1) Setting data

Setting data	Description	Set by ^{*1}	Data type
U0	Dummy	-	Character string
Ⓢ	Connection number (Setting range: 1 to 16)	User	BIN 16-bit
Ⓓ	Start number of the device from which received data are stored	System	Device name

*1 The "Set by" column indicates the following.

User: The data must be set before executing the S.SOCRCVS instruction.

System: The CPU module stores the execution result of the S.SOCRCVS instruction.

Point

When replacing the Z.BUFRCVS instruction (Ethernet module dedicated instruction), dummy argument can be used in a Built-in Ethernet port QCPU instruction without rewriting.

(2) Control data

Device	Item	Description	Setting range	Set by ^{*2}
Ⓓ+0	Receive data length	The length of the data which were read out from the Socket communication receive data area is stored (in bytes).	0 to 10238 ^{*3}	System
Ⓓ+1 to Ⓓ+n	Receive data	The data which were read out from the Socket communication receive data area are stored in ascending order of addresses.	-	

*2 The "Set by" column indicates the following.

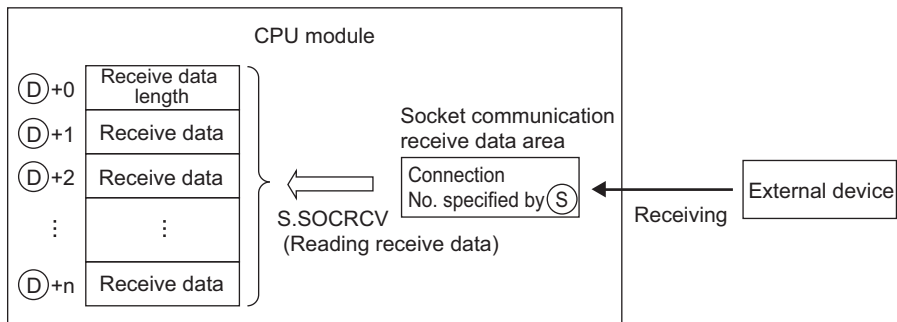
System: The CPU module stores the execution result of the S.SOCRCVS instruction.

*3 0 to 2046 for the QnUDE(H)CPU with a serial number (first five digits) of "12051" or earlier

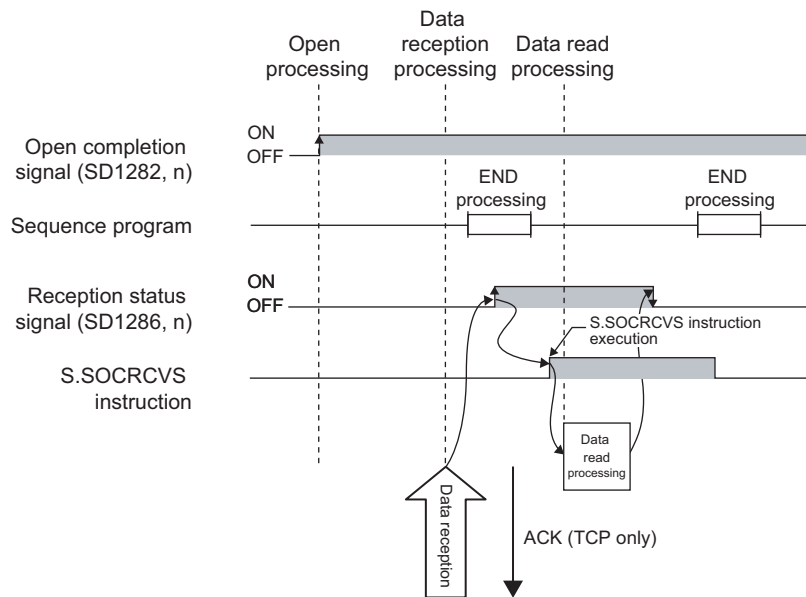
- Receive data size is 2046 bytes by default. To receive data over 2046 bytes, change the receive data size with the SP.SOCRMODE instruction.
- If the CPU module receives odd-byte data, invalid data is stored to the high byte of the device where the last received data is stored.

(3) Function

This instruction reads out received data of the connection specified in (S) from the Socket communication receive data area.



The following figure shows the timing of reception processing with the S.SOCRCVS instruction.



(4) Error

A detection of an operation error turns on the Error flag (SM0) and a corresponding error code is stored in SD0 when:

- The instruction is executed in the QnUDE(H)CPU with a serial number (first five digits) of "11011" or earlier or a CPU module other than the Built-in Ethernet port QCPU. (Error code: 4002)
- The connection number specified for ⑤ is other than 1 to 16. (Error code: 4101)
- The received data size exceeds the size of the receive data storage device. (Error code: 4101)
- The device number specified for ④ exceeds the device point range. (Error code: 4101)
- An invalid device is specified. (Error code: 4004)

(5) Precaution

Do not use both this and SP.SOCRCV instructions when reading out receive data of the same connection.

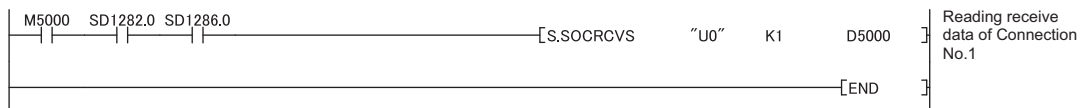
(6) Program example

When M5000 is turned on, data received from the connected device are read out.

- Devices used

Device number	Application
SD1282	Open completion signal
SD1286	Receive state signal
D5000	Receive data length and receive data storage location

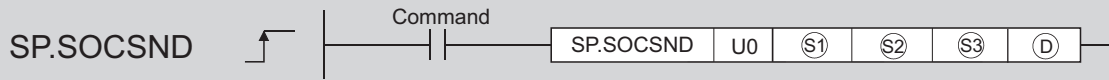
- Program



Point

- To avoid receiving large amounts of data, the volume of received data can be limited by setting the receive data size using the SP.SOCRMODE instruction.
- Data reception can be speeded up by placing a receive processing program at the beginning of multiple programs.

7.4.5 Sending data (SP.SOCSND)



Setting data	Internal device		R, ZR	J□\□		U□\G□	Zn	Constant K, H	Others
	Bit	Word		Bit	Word				
Ⓢ1	-	○	○			-		○	-
Ⓢ2	-	△*1	△*1			-		-	-
Ⓢ3	-	○	○			-		-	-
Ⓓ	△*1	-	△*1			-		-	-

*1 File registers set for each local device or program cannot be used.

(1) Setting data

Setting data	Description	Set by*2	Data type
U0	Dummy	-	Character string
Ⓢ1	Connection number (Setting range: 1 to 16)	User	BIN 16-bit
Ⓢ2	Start number of the device where control data are specified	System	Device name
Ⓢ3	Start number of the device from which send data are stored	User	
Ⓓ	Start number of the device which turns on for one scan upon completion of the instruction Ⓓ+1 also turns on when failed.	System	Bit

*2 The "Set by" column indicates the following.
 User: The data must be set before executing the SP.SOCSND instruction.
 System: The CPU module stores the execution result of the SP.SOCSND instruction.

Point

When replacing the ZP.BUFSND instruction (Ethernet module dedicated instruction), dummy argument can be used in a Built-in Ethernet port QCPU instruction without rewriting.

(2) Control data

Device	Item	Description	Setting range	Set by ^{*3}
Ⓢ2+0	System area	-	-	-
Ⓢ2+1	Completion status	Completion status is stored. 0000 _H : Completed Other than 0000 _H : Failed (Error code)	-	System
Ⓢ3+0	Send data length	The length of send data is specified (in bytes).	1 to 10238 ^{*4}	User
Ⓢ3+1 to Ⓢ3+n	Send data	Send data are specified.	-	

*3 The "Set by" column indicates the following.

User: The data must be set before executing the SP.SOCSND instruction.

System: The CPU module stores the execution result of the SP.SOCSND instruction.

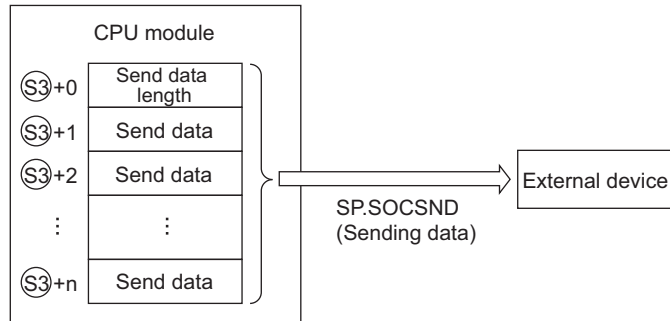
*4 1 to 2046 for the QnUDE(H)CPU with a serial number (first five digits) of "12051" or earlier

Point

For TCP, set the send data length within the maximum window size of the target device (receive buffer of TCP). Data whose size exceeds the maximum window size cannot be sent.

(3) Function

This instruction sends data set in $\textcircled{S}3$ to the target device of the connection specified by $\textcircled{S}3$.

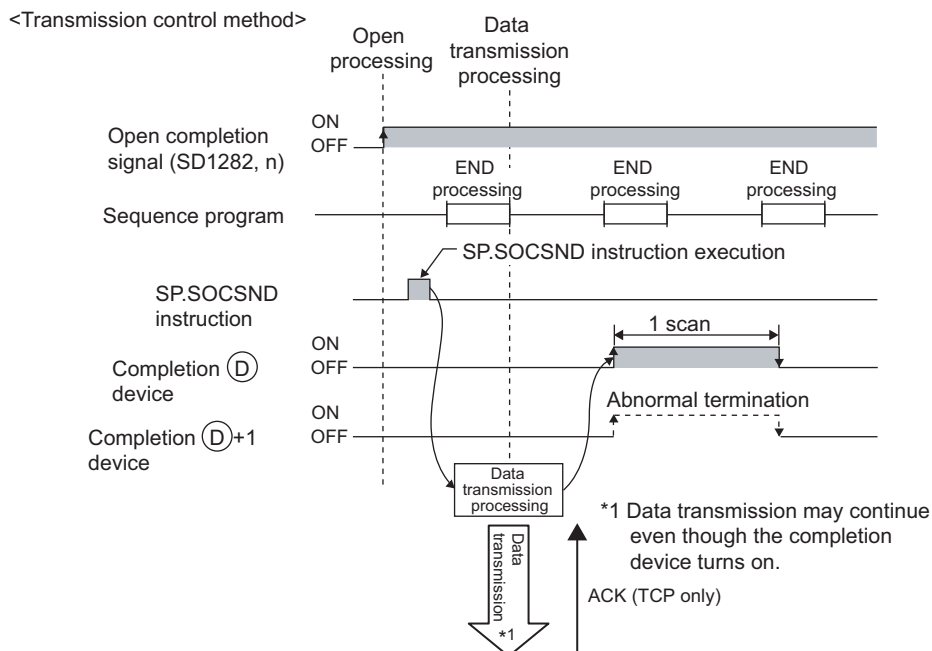


The result of the SP.SOCSND instruction can be checked with the completion device, $\textcircled{D}+0$ and $\textcircled{D}+1$.

- Completion device $\textcircled{D}+0$
Turns on in the END processing of the scan after completion of the SP.SOCSND instruction, and turns off in the next END processing.
- Completion device $\textcircled{D}+1$
Turns on or off according to the result of the SP.SOCSND instruction.

State	Description
When completed	Remains off.
When failed	Turns on in the END processing of the scan after completion of the SP.SOCSND instruction, and turns off in the next END processing.

The following figure shows the timing of reception processing with the SP.SOCRCV instruction.



(4) Error

A detection of an operation error turns on the Error flag (SM0) and a corresponding error code is stored in SD0 when:

- The instruction is executed in the QnUDE(H)CPU with a serial number (first five digits) of "11011" or earlier or a CPU module other than the Built-in Ethernet port QCPU. (Error code: 4002)
- The connection number specified for (S1) is other than 1 to 16. (Error code: 4101)
- The device numbers specified for (S2), (S3), and (D) exceed the device point range. (Error code: 4101)
- An invalid device is specified. (Error code: 4004)

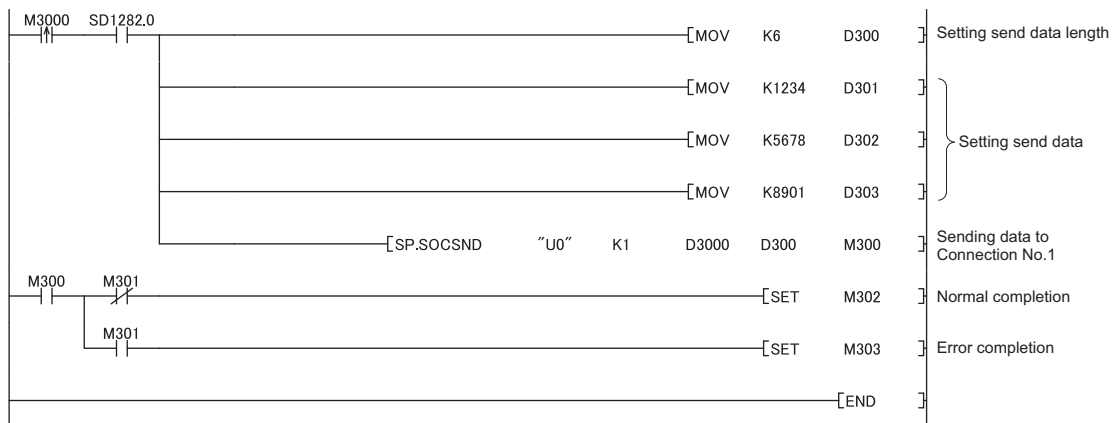
(5) Program example

When M3000 is turned on, data (1234, 5678, and 8901) are sent to the target device using the socket communication function.

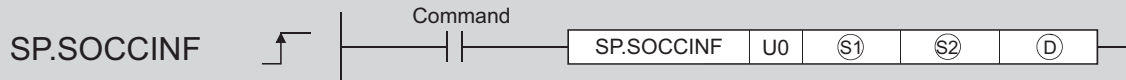
- Devices used

Device number	Application
SD1282	Open completion signal
D3000	SP.SOCSND instruction control data
D300	Send data length and send data storage location
M300	SP.SOCSND instruction completion device

- Program



7.4.6 Reading out connection information (SP.SOCCINF)



Setting data	Internal device		R, ZR	J□\□		U□\G□	Zn	Constant K, H	Others
	Bit	Word		Bit	Word				
Ⓢ1	-	○	○			-		○	-
Ⓢ2	-	○	○			-		-	-
Ⓓ	-	○	○			-		-	-

(1) Setting data

Setting data	Description	Set by ^{*1}	Data type
U0	Dummy	-	Character string
Ⓢ1	Connection number (Setting range: 1 to 16)	User	BIN 16-bit
Ⓢ2	Start number of the device from which control data are stored	System	Device name
Ⓓ	Start number of the device from which connection information is stored		

*1 The "Set by" column indicates the following.

User: The data must be set before executing the SP.SOCCINF instruction.

System: The CPU module stores the execution result of the SP.SOCCINF instruction.

(2) Control data

Device	Item	Description	Setting range	Set by*2
Ⓢ2+0	System area	-	-	-
Ⓢ2+1	Completion status	Completion status is stored. 0000 _H : Completed Other than 0000 _H : Failed (Error code)	-	System
Ⓓ+0 Ⓓ+1	Target device IP address	IP address of the target device is stored.	1 _H to FFFFFFFF _H *4 *5	
Ⓓ+2	Target device port number	Port number of the target device is stored.	1 _H to FFFF _H *4 *6	
Ⓓ+3	Host station port number	Port number of the host station is stored.	1 _H to 1387 _H 1392 _H to FFFE _H *3 *4	
Ⓓ+4	Application setting area	<div style="text-align: center;"> </div> <p>[1] Communication method (protocol) 0: TCP/IP 1: UDP/IP</p> <p>[2] Socket communication procedure 1: Nonprocedural (fixed)</p> <p>[3] Open system 00: Active open or UDP/IP 10: Unpassive open 11: Fullpassive open</p>	*4	

*2 The "Set by" column indicates the following.

System: The CPU module stores the execution result of the SP.SOCCINF instruction.

*3 Using 0400_H to 1387_H and 1392_H to EFFF_H is recommended because the host station port numbers, 0001_H to 03FF_H, are assigned for generally reserved port numbers (WELL KNOWN PORT NUMBERS), and F000_H to FFFE_H are used for other communication functions. Do not specify 1388_H to 1391_H because these ports are used by the system. (Page 154, Appendix 2)

*4 When the item is performed by the unopened connection, 0_H is returned.

*5 When the instruction is performed to the connection set the target device IP address to FFFFFFFF_H (simultaneous broadcast), the source IP address of the received data is returned. In this case, perform the instruction when Receive state signal (SD1286) is on. When the instruction is performed before the data are received, FFFFFFFF_H is returned.

*6 When the instruction is performed to the connection set the target device port number to FFFF_H (simultaneous broadcast), the source port number of the received data is returned. In this case, perform the instruction when Receive state signal (SD1286) is on. When the instruction is performed before the data are received, FFFF_H is returned.

(3) Function

This instruction reads out connection information specified in (S1).

(4) Error

A detection of an operation error turns on the Error flag (SM0) and a corresponding error code is stored in SD0 when:

- The instruction is executed in the QnUDE(H)CPU with a serial number (first five digits) of "11011" or earlier or a CPU module other than the Built-in Ethernet port QCPU. (Error code: 4002)
- The connection number specified for (S1) is other than 1 to 16. (Error code: 4101)
- The device numbers specified for (S2) and (D) exceed the device point range. (Error code: 4101)
- An invalid device is specified. (Error code: 4004)

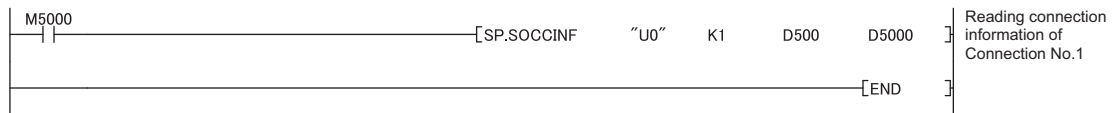
(5) Program example

When M5000 is turned on, connection information of connection No.1 is read out.

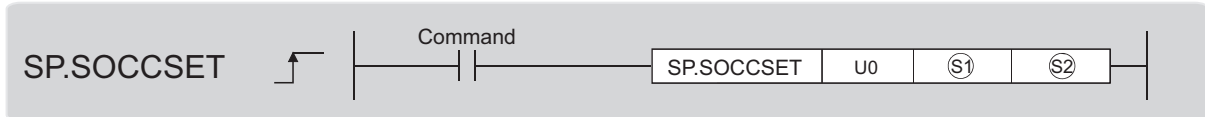
- Devices used

Device number	Application
D500	SP.SOCSND instruction control data
D5000	Storage location of connection information

- Program



7.4.7 Changing the target of a connection (UDP/IP) (SP.SOCCSET)



Setting data	Internal device		R, ZR	J□□		U□\G□	Zn	Constant K, H	Others
	Bit	Word		Bit	Word				
S1	-	○	○			-		○	-
S2	-	○	○			-		-	-

(1) Setting data

Setting data	Description	Set by ^{*1}	Data type
U0	Dummy	-	Character string
S1	Connection number (Setting range: 1 to 16)	User	BIN 16-bit
S2	Start number of the device from which control data are stored	User, System	Device name

*1 The "Set by" column indicates the following.

User: The data must be set before executing the SP.SOCCSET instruction.

System: The CPU module stores the execution result of the SP.SOCCSET instruction.

(2) Control data

Device	Item	Description	Setting range	Set by ^{*2}
S2+0	System area	-	-	-
S2+1	Completion status	Completion status is stored. 0000 _H : Completed Other than 0000 _H : Failed (Error code)	-	System
S2+2 S2+3	Target device IP address	IP address of the target device is specified.	1 _H to FFFFFFFF _H (FFFFFFF _H : Simultaneous broadcast)	User
S2+4	Target device port number	Port number of the target device is specified.	1 _H to FFFF _H (FFFF _H : Simultaneous broadcast)	

*2 The "Set by" column indicates the following.

User: The data must be set before executing the SP.SOCCSET instruction.

System: The CPU module stores execution result of the SP.SOCCSET instruction.

(3) Function

This instruction changes the IP address and port number of the target device of which connection is specified in ①.

(Note that this instruction is available only for UDP/IP communications.)

Point

- The target device can be changed without closing the connection by using the SP.SOCCSET instruction.
- The set values take effect at the following SP.SOCCSET instruction execution timing:
 - When data exist in the socket communication receive data area: After execution of the SP.SOCRCV or S.SOCRCVS instruction, only once after execution of the SP.SOCCSET instruction
 - When no data exists in the socket communication receive data area: After execution of the SP.SOCCSET instruction

(4) Error

A detection of an operation error turns on the Error flag (SM0) and a corresponding error code is stored in SD0 when:

- The instruction is executed in the QnUDE(H)CPU with a serial number (first five digits) of "11011" or earlier or a CPU module other than the Built-in Ethernet port QCPU. (Error code: 4002)
- The connection number specified for ① is other than 1 to 16. (Error code: 4101)
- The device number specified for ② exceeds the device point range. (Error code: 4101)
- An invalid device is specified. (Error code: 4004)

(5) Precaution

Do not change the target device using the SP.SOCCSET instruction during execution of the SP.SOCSND instruction.

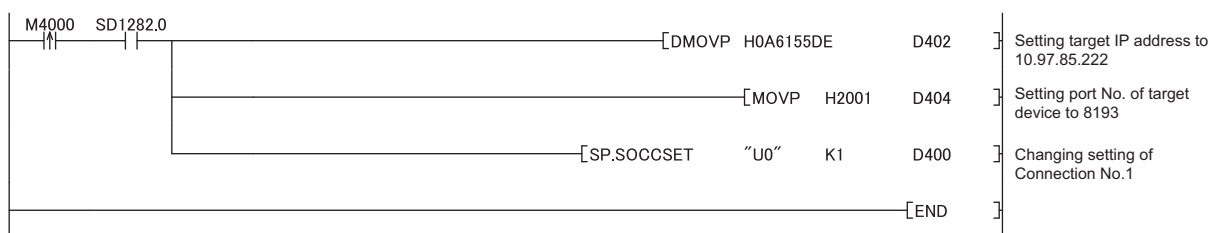
(6) Program example

When M4000 is turned on, the connection destination (IP address and port number of the target device) of connection No.1, which is open, is changed.

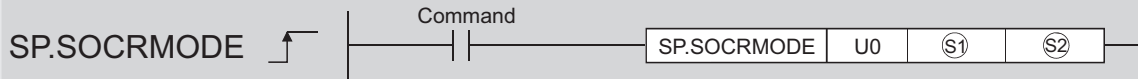
- Devices used

Device number	Application
SD1282	Open completion signal
D400	SP.SOCCSET instruction control data

- Program



7.4.8 Changing the receive mode of a connection (SP.SOCRMODE)



Setting data	Internal device		R, ZR	J□□		U□\G□	Zn	Constant K, H	Others
	Bit	Word		Bit	Word				
Ⓢ1	-	○	○			-		○	-
Ⓢ2	-	○	○			-		-	-

(1) Setting data

Setting data	Description	Set by ^{*1}	Data type
U0	Dummy	-	Character string
Ⓢ1	Connection number (Setting range: 1 to 16)	User	BIN 16-bit
Ⓢ2	Start number of the device from which control data are stored	User, System	Device name

*1 The "Set by" column indicates the following.

User: The data must be set before executing the SP.SOCRMODE instruction.

System: The CPU module stores the execution result of the SP.SOCRMODE instruction.

(2) Control data

Device	Item	Description	Setting range	Set by ^{*3}
Ⓢ2+0	System area	-	-	-
Ⓢ2+1	Completion status	Completion status is stored. 0000 _H : Completed Other than 0000 _H : Failed (Error code)	-	System
Ⓢ2+2	TCP receive mode ^{*2}	TCP receive mode is stored. 0: TCP standard receive mode 1: TCP fixed-length receive mode	0 or 1	User
Ⓢ2+3	Receive data size	Receive data size in the socket communication is stored (in bytes).	1 to 10238 ^{*4}	

*2 Invalid for connections in UDP communications.

*3 The "Set by" column indicates the following.

User: The data must be set before executing the SP.SOCRMODE instruction.

System: The CPU module stores the execution result of the SP.SOCRMODE instruction.

*4 1 to 2046 for the QnUDE(H)CPU with a serial number (first five digits) of "12051" or earlier

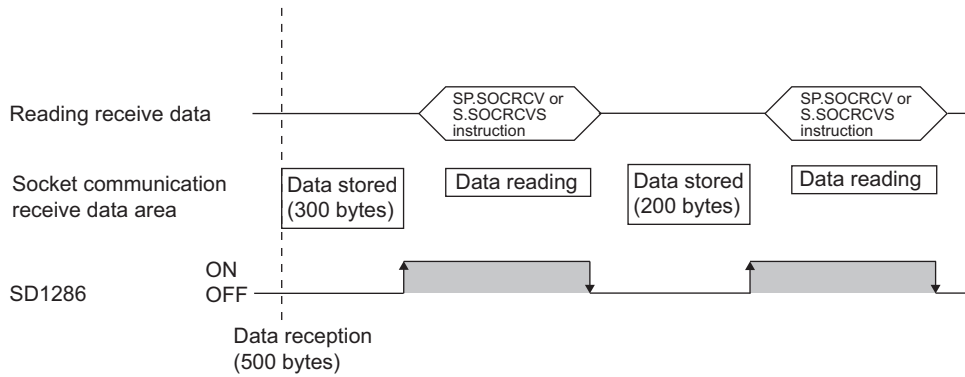
(3) Function

This instruction changes the TCP receive mode and receive data size of the connection specified in ⑤1. The mode is changed as specified in ⑤2+2. (This instruction is invalid for UDP connections.)

(a) TCP standard receive mode

When data are received, they are stored in the Socket communication receive data area, and SD1286 turns on. If data are received exceeding the specified receive data size, the excess data are read out the next time.

Ex. The receive data size is set to 300 bytes, and 500-byte data are received.



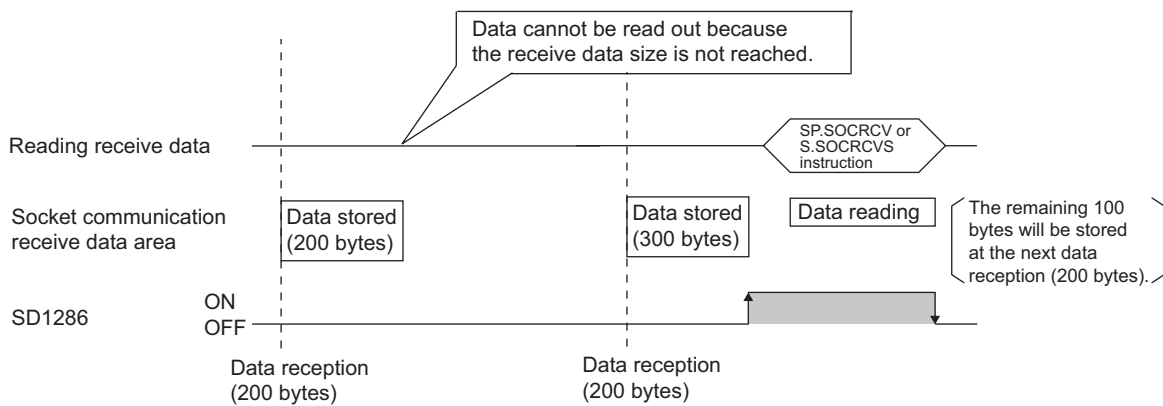
(b) TCP fixed-length receive mode

When data are received, they are stored in the Socket communication receive data area, and SD1286 (Receive state signal) turns on. However, if the size of the data does not reach the specified receive data size, SD1286 does not turn on.

Data reception is repeated and once the data size reaches to the specified size, SD1286 turns on.

If data are received exceeding the specified receive data size, the excess data are read out the next time.

Ex. The receive data size is set to 300 bytes, and 200-byte data are consecutively received.



Point

- **Effective use of devices**
Devices can be effectively used by setting the receive data size to less than 1024 words while the default size of the receive data storing devices for the SP.SOCRCV and S.SOCRCVS instructions is 1024 words.
- **Prevention of receive data fragmentation**
Data from the connected device may be fragmented depending on the line type. To prevent this, the receive data size can be specified in the TCP fixed-length receive mode.
- **Prevention of receive data binding**
Separately sent data may be combined depending on the interfacing device due to a delay in receive processing of the program.
To prevent this, the receive data size can be specified in the TCP fixed-length receive mode.

Remark

- The set values will take effect at the following timing of the SP.SOCRMODE instruction execution.
- Before opening: Values take effect after opening a connection.
 - When there are data in the Socket communication receive data area:
Values take effect after execution of the SP.SOCRCV or S.SOCRCVS instruction once after the S.SOCRMODE instruction.
 - When there is no data in the Socket communication receive data area:
Values take effect after the SP.SOCRMODE instruction is executed.

(4) Error

A detection of an operation error turns on the Error flag (SM0) and a corresponding error code is stored in SD0 when:

- The instruction is executed in the QnUDE(H)CPU with a serial number (first five digits) of "11011" or earlier or a CPU module other than the Built-in Ethernet port QCPU. (Error code: 4002)
- The connection number specified for ⑤① is other than 1 to 16. (Error code: 4101)
- The device number specified for ⑤② exceeds the device point range. (Error code: 4101)
- Invalid device is specified. (Error code: 4004)

Remark

Even when the Receive state signal is not on in the TCP fixed-length receive mode, the data received at the point can be read out with the SP.SOCRDATA instruction. This allows you to check whether the data sent from the connected device is adequate in size.

(5) Program example

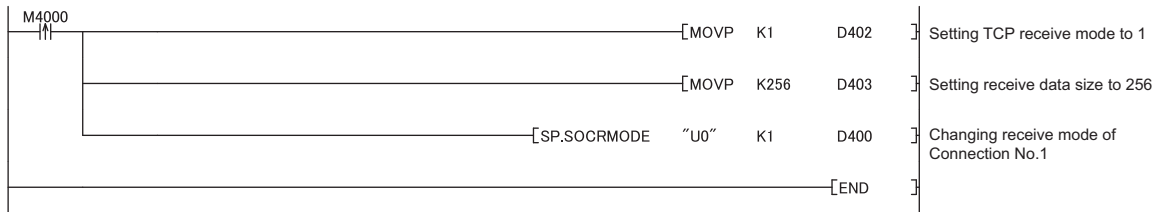
When M4000 is turned on, connection No.1 is set to the TCP fixed-length receive mode and the receive data size is set to 256 bytes.

After execution of the instruction and when the receive data size of connection No.1 reaches 256 bytes, the Receive state signal is turned on.

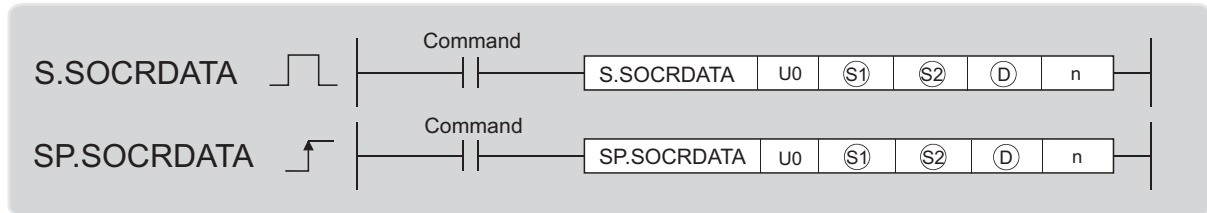
- Device used

Device number	Application
D400	SP.SOCRMODE instruction control data

- Program



7.4.9 Socket communication receive data read (S(P).SOCRDATA)



Setting data	Internal device		R, ZR	J□\□		U□\G□	Zn	Constant K, H	Others
	Bit	Word		Bit	Word				
Ⓢ1	-	○	○			-		○	-
Ⓢ2	-	○	○			-		-	-
Ⓓ	-	○	○			-		-	-
n	-	○	○			-		○	-

(1) Setting data

Setting data	Description	Set by ^{*1}	Data type
U0	Dummy	-	Character string
Ⓢ1	Connection number (Setting range: 1 to 16)	User	BIN 16-bit
Ⓢ2	Start number of the device from which control data are stored		Device name
Ⓓ	Start number of the device where read data are stored		
n	Number of read data (1 to 5120 words ^{*2})		BIN 16-bit

*1 The "Set by" column indicates the following.

User: The data must be set before executing the S(P).SOCRDATA instruction.

*2 1 to 1024 for the QnUDE(H)CPU with a serial number (first five digits) of "12051" or earlier

(2) Control data

Device	Item	Description	Setting range	Set by ^{*2}
Ⓢ2+0	System area	-	-	-
Ⓢ2+1	Completion status	Completion status is stored. 0000 _H : Completed Other than 0000 _H : Failed (Error code)	-	System

*2 The "Set by" column indicates the following.

System: The CPU module stores the execution result of the S(P).SOCRDATA instruction.

(3) Function

This instruction reads out the data of the amount specified for n from the Socket communication receive data area of which connection is specified in ⑤, and stores them in the device specified in ④ or higher. No processing is performed when the number of read data (n) is 0.

Point

- Data of the receive data length can be read out by setting the number of read data to one word. This allows change of the device storing receive data, upon execution of the SP.SOCRCV or S.SOCRCVS instruction.
- In the following order, based on the data currently received, the size of the data received the next time can be specified.
 1. Check the data currently received using the S(P).SOCRDATA instruction.
 2. Specify the size of the data to be received the next time using the SP.SOCRMODE instruction.
 3. Read out the data currently received using the SP.SOCRCV or S.SOCRCVS instruction.

Remark

- Even if the S(P).SOCRDATA instruction is executed, the next receive data will not be stored in the Socket communication receive data area because the area is not cleared and the Receive state signal does not change.
- To update the receive data, read out the data using the SP.SOCRCV or S.SOCRCVS instruction.

(4) Error

A detection of an operation error turns on the Error flag (SM0) and a corresponding error code is stored in SD0 when:

- The instruction is executed in the QnUDE(H)CPU with a serial number (first five digits) of "11011" or earlier or a CPU module other than the Built-in Ethernet port QCPU. (Error code: 4002)
- The connection number specified for ⑤ is other than 1 to 16. (Error code: 4101)
- The device numbers specified for ②, ④, and n1 exceed the device point range. (Error code: 4101)
- An invalid device is specified. (Error code: 4004)

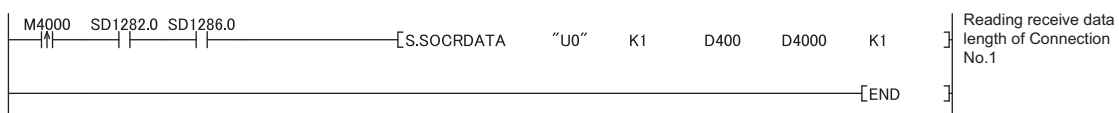
(5) Program example

When M4000 is turned on, the receive data length of connection No.1 is read out.

- Devices used

Device number	Application
SD1282	Open completion signal
SD1286	Receive state signal
D400	S.SOCRDATA instruction control data
D4000	Storage location where data are read out
K1	Number of read data (one word)

- Program



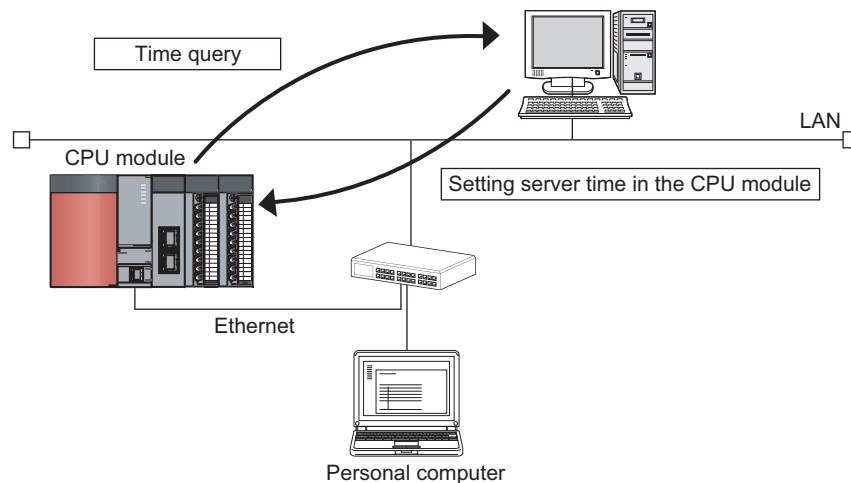
CHAPTER 8 TIME SETTING FUNCTION (SNTP CLIENT)

The CPU module collects time information from a time information server on LAN, and automatically sets its own time. With this time setting function, the CPU module queries the server for time information at the specified timing, and can set the time information sent from the server as its own clock data.

The time setting can be performed at the following timing.

- When the CPU module is powered off and then on, or is reset
- At the specified time intervals (Execution interval)
- At the specified clock time (Execution time)
- According to the special relay state ^{*1}

*1 Time is set when SM1270 is turned on for one scan.



Point

- Confirm the connection of the hub or the interfacing device in advance when setting time at the timing of CPU module power-on or reset.
- The time setting results can be checked with the special register (SD1270 to SD1275).
- During execution of the time setting function, other time setting operations are ignored.

Remark

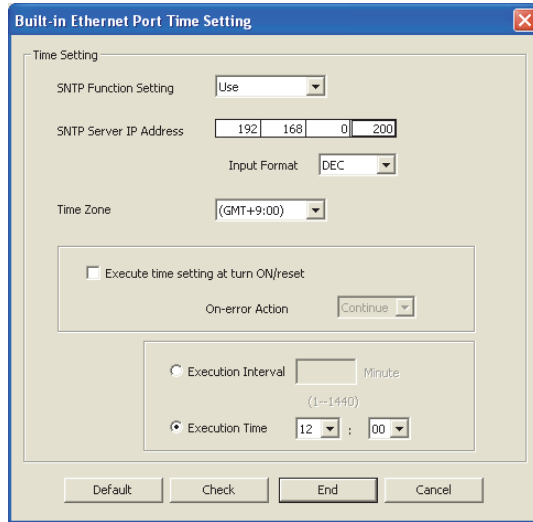
Access through routers is also available. When configuring the settings for it, set the subnet mask pattern and default router IP address. (☞ Page 30, Section 3.4)

8.1 Setting Method

Configure the time settings in the Built-in Ethernet port tab of the PLC parameter window.

Project window ⇨ [Parameter] ⇨ [PLC Parameter] ⇨ [Built-in Ethernet Port Setting]

⇨  button



Item	Description	Setting range
SNTP Function Setting	Select whether to use this function or not.	Used or Not used
SNTP Server IP Address	Specify the IP address of the SNTP server.	0.0.0.1 to 223.255.255.254
Time Zone	Specify a time zone in which the time is to be synchronized. Japan Standard Time "GMT+9:00" is set by default.	(GMT-12:00 to GMT+13:00)
Execute time setting at turn ON/reset	Select whether to execute the time setting function upon power-on or reset of the CPU module.	-
At Error Occurrence	Select whether to stop or continue the time setting when an error is detected upon power-on or reset of the CPU module.	Continue or Stop
Execution Interval ^{*2}	Select this when executing the time setting function at fixed time intervals.	1 to 1440 (min.)
Execution Time ^{*2}	Select this when executing the time setting function at a specified time (in increments of 30 minutes).	00:00 to 23:30

*2 Either of these two options must be selected.

8.2 Precautions

(1) Communication timeout

A communication timeout occurs when 20 seconds have elapsed without receiving any response after sending a time query.

At the time of a communication timeout, the value in SD1270 is FFFF_H.

(2) Time information server

When the time setting function is used, an SNTP server (time information server) is required on the LAN.

(3) Delay resulted from the time required for communication

A delay occurs and affects the set time as a result of the time spent for communication with the SNTP server computer. For a high-accuracy time setting, specify an SNTP server computer that is networked as close to the CPU module as possible.

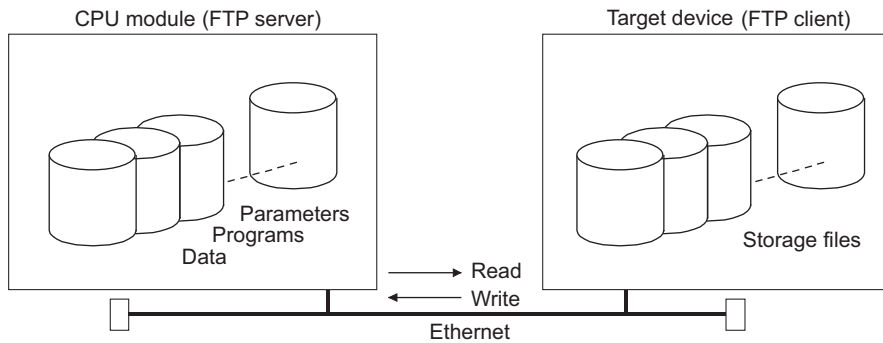
(4) For multiple CPU system configuration

In a multiple CPU system, enable the time setting function of only the Built-in Ethernet port QCPU No.1. When a CPU module other than the Built-in Ethernet port QCPU No.1 is enabled, the clock data of the Built-in Ethernet port QCPU No.1 is automatically set.

CHAPTER 9 FILE TRANSFER FUNCTION (FTP)

The CPU module supports the server function for FTP (File Transfer Protocol) which is a protocol designed for file transfer to or from the connected device.

The device with the FTP client function can directly access any files located in the CPU module.



The following operations can be performed between the connected device with the FTP client function and the CPU module.

(a) Reading files from the CPU module (download)

The files in the CPU module can be stored in the connected device.

(b) Writing files to the CPU module (upload)

The files stored in the connected device can be registered to the CPU module.

(c) Browsing the names of the files in the CPU module

The files registered to the CPU module can be checked from the connected device.

Point

In a multiple CPU system, only the CPU module connected with an Ethernet cable can transfer files.

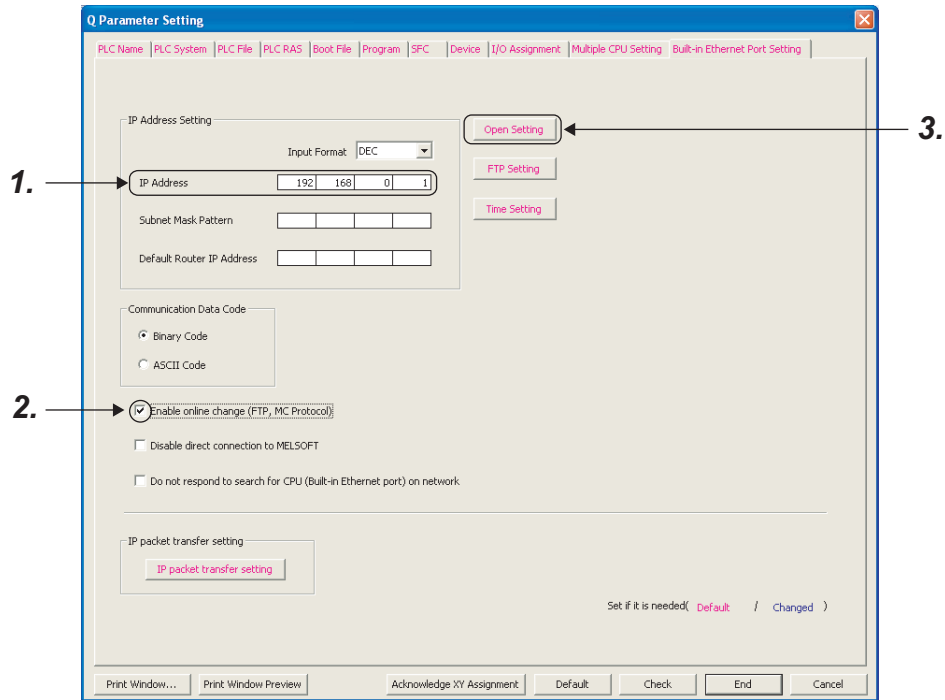
Remark

Access through routers is also available. When configuring the settings for it, set the subnet mask and default router IP address. (Page 30, Section 3.4)

9.1 Setting for FTP Communication

(1) Operation on the CPU module side

Project window ⇒ [Parameter] ⇒ [PLC Parameter] ⇒ [Built-in Ethernet Port Setting]



1. Set the IP address of the CPU module.
2. Select "Enable online change (FTP, MC Protocol)" when data need to be written even while the CPU module is in RUN state.

3. Configure the FTP settings.

Item	Description
FTP	Select "Used".
Login Name	Set a log-in name used for file transfer (login) request from an external device. [Default value] • QnUDVCPU: "MELSEC" • QnUDE(H)CPU: "QNUDECPU"
Password	Set an FTP password used for file transfer request from an external device. To change the password, enter both the current password and a new password for confirmation. [Default value] • QnUDVCPU: "MELSEC" • QnUDE(H)CPU: "QNUDECPU"
Command Input Monitoring Timer	Set a time for monitoring command input performed by the CPU module. When no command is input within the set period of time, the FTP connection is disconnected. (Setting range : 1 to 32767 (× 500ms)) Set a time value larger than the time required for file transfer.

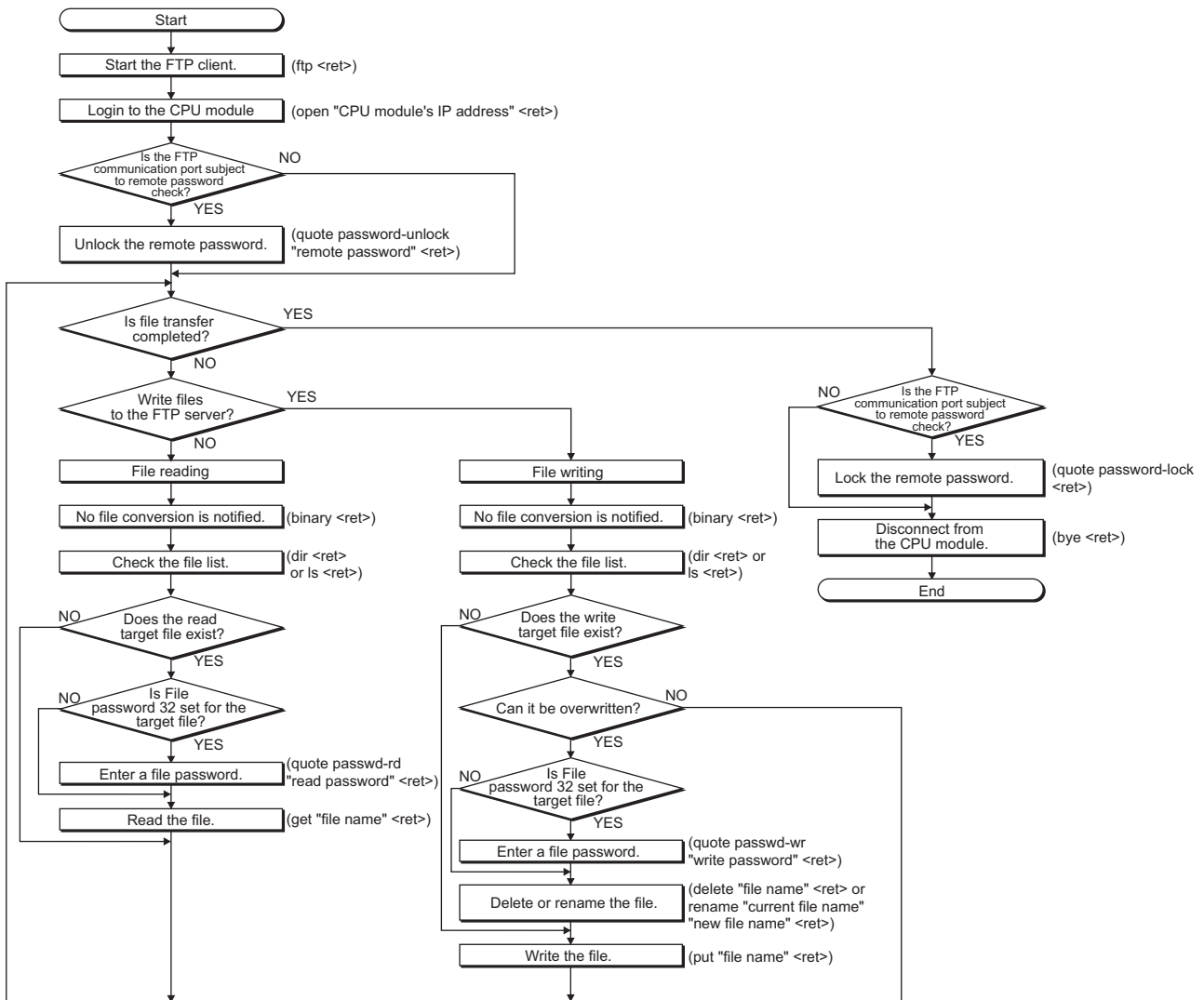
(2) Operation on the connected device (FTP client) side

The following describes the procedure and processing on the connected device side, which is required for using the FTP server function of the CPU module.

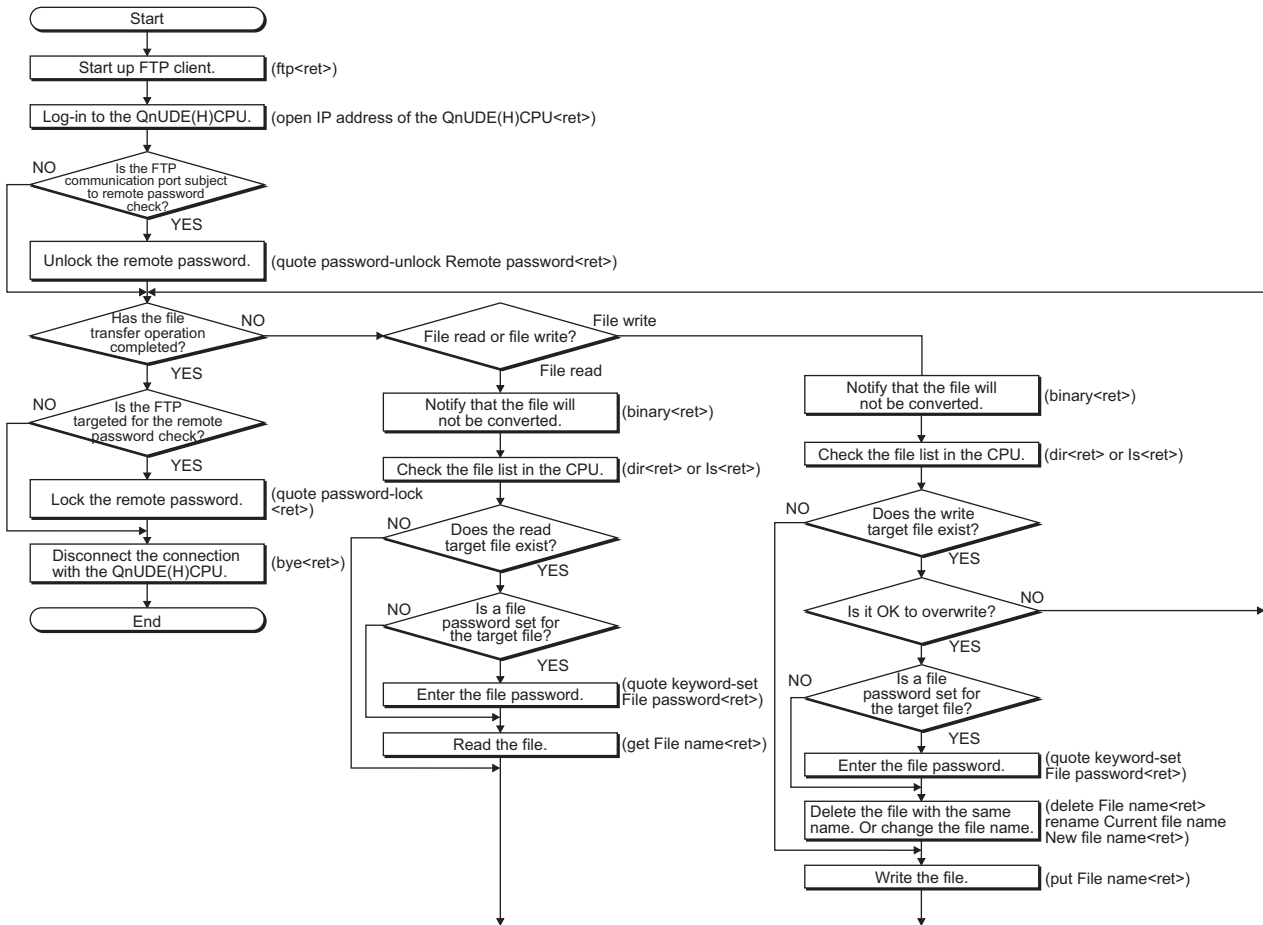
Various FTP commands and how to enter each of them are also shown.

(<ret> indicates an input of the CR, Enter, or Return key.)

(a) For the High-speed Universal model QCPU



(b) For the QnUDE(H)CPU



(c) Logging in to the CPU module

The following explains the operation flow from starting an FTP session until log-in to the CPU module.

Ex. Start up the FTP from the MS-DOS prompt of Microsoft® Windows®.

```

ex Command Prompt - ftp 10.97.14.254
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS>CD\
C:\>ftp
1. → ftp> open 10.97.14.254
2. → Connected to 10.97.14.254
220 QnUDE(H)CPU FTP server ready.
3. → User (10.97.14.254:(none)):CPU
331 Password required.
4. → Password:
230 User logged in.
ftp>
  
```

1. Start an FTP session. (FTP <ret>)
2. Open a connection to the FTP server (open "IP address of the CPU module" <ret>)
3. Specify the login name (Login name <ret>)
4. Specify the password (Password <ret>)

(d) Locking and unlocking a remote password

If the FTP communication port is specified as a remote password check target in the remote password settings, unlock the remote password using the following command.

(quote password-unlock "Remote password" <ret>)

When terminating the operation, lock the unlocked remote password using the following command.

(quote password-lock <ret>)

Point

If the FTP communication port is specified as a remote password check target, any other commands cannot be used until the remote password is unlocked.

(e) Entering a file password

When a file password has been set in the target file, the file password has to be entered using the following command before accessing the file.

CPU module	Command
QnUDVCPU	Write password (quote passwd-wr "write password" <ret>)
	Read password (quote passwd-rd "read password" <ret>)
QnUDE(H)CPU	File password (quote keyword-set "file password" <ret>)


9.2 Files Transferable Using FTP

The following table lists the files that can be transferred using the file transfer function.

○ :Available, × :N/A

File type	Program memory *2*3	Standard RAM	Standard ROM*4	SRAM card	Flash card	ATA card	SD memory card*4	File name or extension
	Drive 0	Drive 3	Drive 4	Drive 1	Drive 2			
Parameter	○	×	×	×	×	×	×	PARAM.QPA
Intelligent function module parameter	○	×	×	×	×	×	×	IPARAM.QPA
Program	○	×	×	×	×	×	×	.QPG
Device comment	○	×	×	×	×	×	×	.QCD
Device initial value	○	×	×	×	×	×	×	.QDI
File register	×	○	×	○	○	×	×	.QDR
Local device	×	×	×	×	×	×	×	.QDL
Sampling trace file	×	×	×	×	×	×	×	.QTD
Programmable controller user data*1	×	×	○	×	×	○	○	Any file name
Source information	×	×	×	×	×	×	×	*5
Drive heading	×	×	×	×	×	×	×	QN.DAT
Device data storage file	×	×	×	×	×	×	×	DEVSTORE.QST
Module error log file	×	×	×	×	×	×	×	IERRLOG.QIE
Boot setting file	×	×	×	×	×	×	×	AUTOEXEC.QBT
Remote password	×	×	×	×	×	×	×	00000000.QTM
Latch data backup file	×	×	×	×	×	×	×	LCHDAT00.QBP
Backup data file	×	×	×	×	×	×	×	MEMBKUP0.QBP
Data logging setting file	×	×	×	×	×	×	×	LOGCOM.QLG, LOG01.QLG to LOG10.QLG
Data logging file	×	×	×	×	×	×	○	.CSV
Predefined protocol setting file	×	×	×	×	×	×	×	ECPRTCL.QPT

*1 The SP.FWRITE or SP.FREAD instruction in the program will write or read the file respectively. For details, refer to the following manual.

 MELSEC-Q/L Programming Manual (Common Instruction)

*2 Files can be written to the program memory only when the CPU module is in the STOP state.

*3 Write destination is the program cache memory.

The user should back up files using the "pm-write" command as required. ( Page 120, Section 9.4.1)

*4 The following number of files can be stored using FTP.

[Maximum number of files storable on the drive - 1]

*5 Except for the High-speed Universal model QCPU, file names are SRCINF1M.CAB and SRCINF2M.CAB for Simple projects (with labels), and SRCINF11.CAB and SRCINF21.CAB for Structured projects.

For the High-speed Universal model QCPU, file names are SRCINF1M.C32 and SRCINF2M.C32 for Simple projects (with labels), and SRCINF11.C32 and SRCINF21.C32 for Structured projects.

9.3 Files That Can Be Deleted Using FTP

The following table lists the files that can be deleted using the file transfer function.

○: Deletable, △: Not deletable only in RUN state, ×: Not deletable, -: Not writable

File type	Program memory ^{*2*3}	Standard RAM	Standard ROM ^{*4}	SRAM card	Flash card	ATA card	SD memory card ^{*4}	File name or extension
	Drive 0	Drive 3	Drive 4	Drive 1	Drive 2			
Parameter	△	○ ^{*6}	○	○	○	○	○	PARAM.QPA
Intelligent function module parameter	△	○ ^{*6}	○	○	○	○	○	IPARAM.QPA
Program	△	○ ^{*6}	○	○	○	○	○	.QPG
Device comment	△	○ ^{*6}	○	○	○	○	○	.QCD
Device initial value	△	○ ^{*6}	○	○	○	○	○	.QDI
File register	-	○	-	○	○	-	-	.QDR
Local device	-	○	-	○	-	-	-	.QDL
Sampling trace file	-	○	-	○	-	-	-	.QTD
Programmable controller user data ^{*1}	-	-	○	-	-	○	○	Any file name
Source information	△	○	○	○	○	○	○	^{*5}
Drive heading	△	○ ^{*6}	○	○	○	○	○	QN.DAT
Device data storage file	-	-	△	-	-	-	-	DEVSTORE.QST
Module error log file	-	×	-	-	-	-	-	IERRLOG.QIE
Boot setting file	△	○ ^{*6}	○	○	○	○	○	AUTOEXEC.QBT
Remote password	△	○ ^{*6}	○	○	○	○	○	00000000.QTM
Latch data backup file	-	-	○	-	-	-	-	LCHDAT00.QBP
Backup data file	-	-	-	○	○	○	○	MEMBKUP0.QBP
Data logging setting file	-	-	○	-	-	-	○	LOGCOM.QLG, LOG01.QLG to LOG10.QLG
Data logging file	-	-	-	-	-	-	○	.CSV
Predefined protocol setting file	×	×	○	×	×	×	○	ECPRTCL.QPT

*1 The SP.FWRITE or SP.FREAD instruction in the program will write or read the file respectively. For details, refer to the following manual.

 MELSEC-Q/L Programming Manual (Common Instruction)

*2 Files can be written to the program memory only when the CPU module is in the STOP state.

*3 Write destination is the program cache memory.

The user should back up files using the "pm-write" command as required. ( Page 120, Section 9.4.1)

*4 The following number of files, which can be written using FTP, can be stored.

[Maximum number of files storable on the drive - 1]

*5 Except for the High-speed Universal model QCPU, file names are SRCINF1M.CAB and SRCINF2M.CAB for Simple projects (with labels), and SRCINF1I.CAB and SRCINF2I.CAB for Structured projects.

For the High-speed Universal model QCPU, file names are SRCINF1M.C32 and SRCINF2M.C32 for Simple projects (with labels), and SRCINF1I.C32 and SRCINF2I.C32 for Structured projects.

*6 For the QnUDE(H)CPU, this file cannot be stored.



When the online change settings is disabled in the "Built-in Ethernet Port Setting" tab of PLC Parameter, deleting a file in RUN state will cause an error.


9.4 FTP Commands

9.4.1 List of FTP commands

○ : Available, × : N/A

Command	Function	CPU module state			Remote password	
		STOP	RUN		Unlocked *2	Locked *2
			Write enabled *1	Write disabled *1		
binary ^{*5}	Notifies the FTP server of file transfer without conversion.	○	○	○	○	×
bye	Disconnects the line to the FTP server and terminates the session.	○	○	○	○	○
close	Disconnects the line to the FTP server.	○	○	○	○	○
delete ^{*6}	Deletes a file in the CPU module. ^{*4}	○	○	×	○	×
dir	Displays file information of the CPU module. ^{*4}	○	○	○	○	×
get ^{*6}	Reads a file from the CPU module. ^{*4}	○	○	○	○	×
ls	Displays file names of the files stored in the CPU module. ^{*4}	○	○	○	○	×
mdelete ^{*6}	Deletes file(s) stored in the CPU module. ^{*4}	○	○	×	○	×
mdir	Stores file information of the CPU module into a file. ^{*4}	○	○	○	○	×
mget ^{*6}	Reads file(s) from the CPU module. ^{*4}	○	○	○	○	×
mls	Stores CPU module's file names into a file. ^{*4}	○	○	○	○	×
mput ^{*6}	Writes file(s) to the CPU module.	○	○	×	○	×
open	Connects to the FTP server.	○	○	○	○	○
put ^{*6}	Writes a file to the CPU module.	○	○	×	○	×
pwd	Displays the current directory of the CPU module.	○	○	○	○	×
quit	Disconnects the line to the FTP server and terminates the connection.	○	○	○	○	○
quote	Sends an FTP server subcommand. ^{*3}	○	○	○	○	○
rename ^{*6}	Changes a CPU module file name. ^{*4}	○	○	×	○	×
user	Inputs the user name and password of the CPU module.	○	○	○	○	○

*1 Whether the online change settings is enabled or not is indicated in the "Built-in Ethernet Port" tab in the PLC Parameter window.

*2 These indicate whether the remote password can be used or not when the FTP communication port is specified as a remote password check target in the remote password setting. For remote passwords, refer to:  Page 134, CHAPTER 10

*3 On the next page, subcommands available with the "quote" command are shown.

*4 Each of these commands can include a folder name in the file specification.

*5 This command is set automatically in the CPU module. Therefore, a file transfer is coded in binary regardless of the "Communication Data Code" of the "Built-in Ethernet Port Setting" in PLC Parameter.

*6 The command cannot be used for access control target files in the file access control by security key.

The following table lists the subcommands available with the command, "quote".

○ : Available, × : N/A

Subcommand	Function	CPU module state			Remote password	
		STOP	RUN		Unlocked	Locked
			Write enabled	Write disabled		
change ^{*5}	Displays or changes the CPU module file attribute. ^{*2}	○	○	×	○	×
keyword-set ^{*3}	Sets/displays/clears the Built-in Ethernet port QCPU file access password.	○	○	○	○	×
password-lock	Locks an unlocked remote password.	○	○	○	○	× ^{*1}
password-unlock	Unlocks an locked remote password.	○	○	○	○	○
status	Displays the operation information of the CPU module.	○	○	○	○	×
run	Changes the CPU module state to RUN.	○	○	○	○	×
stop	Changes the CPU module state to STOP.	○	○	○	○	×
pm-write	Writes data to the program memory.	○	×	×	○	×
passwd-rd ^{*4}	Sets, displays, or clears file password 32 (read password).	○	○	○	○	×
passwd-wr ^{*4}	Sets, displays, or clears file password 32 (write password).	○	○	○	○	×

- *1 Even if the subcommand is executed, the remote password remains locked with no error occurred.
- *2 This command can include a folder name in the file specification.
- *3 The subcommand cannot be used for the QnUDVCPU.
- *4 The subcommand cannot be used for the QnUDE(H)CPU.
- *5 The subcommand cannot be used for access control target files in the file access control by security key.

9.4.2 How to specify an FTP command

This section explains how to specify a file with an FTP command on the FTP client (connected device), which is supported by the CPU module.

(1) File specification

A file can be specified for an FTP command on the FTP client side as follows:

- For CPU modules, each file is specified using a drive name and a file name.*²
- When specifying a file in the CPU module using the FTP function, specify the target file in the following order.

[Specification format] Drive name:\Folder name*¹\File name.Extension

[Example]

3:\MAINSEQ1.QDR (other than drive 2)

2:\LOGGING\LOG01\00000001\LOG01_00000001.CSV (drive 2)


[Specification details] Refer to (a) and (b) below.

*¹ Only for a file in drive 2, the folder name can be specified.



*² Use "\" as a delimiter.

(a) Drive name (drive No.)

Specify the drive name of the file transfer target memory.

For the CPU module's memories and drive names, refer to:  Page 118, Section 9.2

(b) Folder name, file name, and extension

- Specify a folder name and a file name according to the rules described in the following manual.
 QnUCPU User's Manual (Function Explanation, Program Fundamentals)
- Set an extension predetermined by the CPU module.  Page 118, Section 9.2
- For an FTP command which can be used for multiple files, specify the file name and extension using a wild card character (* or ?).
 - *: Indicates all files having any characters (including no character) from the position where "*" is used.
 - ?: Indicates all files having any characters (including no character) at the position where "?" is used. (Multiple "?" can be used.)

Some FTP clients have other restrictions on the characters that can be used for file names.

Point

The part enclosed with brackets in the specification format can be omitted.

9.4.3 Details of FTP commands

Details of the FTP commands on the FTP client side, which are supported by the CPU module, and how to use each of them are described below.

Point

Note that some FTP commands may not function as described in this manual, depending on the FTP application used on the FTP client side.

Check the functions and operation methods, referring to the manual(s) for the FTP client.

(1) FTP server support commands

●binary

[Function] Notifies the FTP server of file transfer without conversion.
Neither return codes nor kanji codes are converted.
This command is set automatically in the CPU module.

[Specification format] binary (abbreviated to "bin")

●bye

[Function] Disconnects the line to the FTP server, and terminates the FTP session.

[Specification format] bye

[Identical command] quit

●close

[Function] Disconnects the line to the FTP server.

[Specification format] close

●delete

[Function] Deletes a file stored in the CPU module.

[Specification format] delete "file path name"

[Example] When deleting a file stored in an SD memory card:
delete 2:\MAINSEQ1.USR

[Similar command] mdelete

●dir

[Function] Displays the names, creation dates, and sizes of the files stored in the CPU module.

[Specification format] dir [drive name:\]

[Example] When displaying the detailed information of the files stored in an SD memory card:
dir 2:\

[Similar command] ls

●get

[Function]	Reads a file from the CPU module.
[Specification format]	get "source file path name" [destination file path name]
[Example 1]	When reading a file stored in the standard RAM and saving it under the same file name: get 3:\MAINSEQ1.QDR
[Example 2]	When reading a file stored in the standard RAM and saving it under a different file name: get 3:\SEQ1BAK.QDR \SEQ\SEQ10LD.QDR
[Note]	<ul style="list-style-type: none">• When no destination file path name (FTP client side) is specified, the file is saved in the FTP client using the same source file name (CPU module side).• The transfer destination is on the current directory where FTP is started up and connected to the server.

●ls

[Function]	Displays the names of the files stored in the CPU module.
[Specification format]	ls [drive name:]
[Example]	When displaying the names of the files stored in an SD memory card: ls 2:\
[Similar command]	dir

●mdelete

[Function]	Deletes a file stored in the CPU module. To delete multiple files, specify the file name and extension within the file path name using a wild card character (* or ?).
[Specification format]	mdelete "file path name" (abbreviated to "mdel")
[Example]	When deleting all the files whose extensions are "QPG" from the program memory: mdelete 0:*.QPG
[Similar command]	delete

●mdir

[Function]	Saves detailed information (file names, creation dates, and sizes) of the files stored in the CPU module as log data into a file on the FTP client side.
[Specification format]	mdir "source drive name" :\"destination file path name"
[Example]	When saving detailed information of the files stored in an SD memory card into the S990901.LOG file: mdir 2:\ S990901.LOG
[Note]	<ul style="list-style-type: none">• Type ":" immediately after the source drive name.• Specify a source drive name when specifying a destination file path name (FTP client side).• Without a destination file path name, the file is saved using a file name determined by the FTP application on the FTP client side.• The transfer destination is on the current directory where FTP is started up and connected to the server.
[Similar command]	mls

●mget

[Function]	Reads out a file from the CPU module. To read out multiple files, specify the file name and extension within the file path name using a wild card (* and/or ?). When reading multiple files, reception is checked for each file transfer.
[Specification format]	mget "file path name"
[Example]	When reading all the files whose extensions are "USR" among the files stored in an SD memory card: mget 2:*.USR
[Note]	The file read out is saved on the FTP client side under the same file name. The storage destination is on the current directory where FTP is started up and connected to the server.

●mls

[Function]	Stores the names of the files in the CPU module as log data into a file on the FTP client side.
[Specification format]	mls "source drive name":\ "destination file path name"
[Example]	When storing the names of the files in an SD memory card into the S990901F.LOG file: mls 2:\ S990901F.LOG
[Note]	<ul style="list-style-type: none"> • Type "\" immediately after the source drive name. • Specify a source drive name when specifying a destination file path name (FTP client side). • Without a destination file path name, the file is stored using a file name determined by the FTP application on the FTP client side. • The transfer destination is on the current directory where FTP is started up and connected to the server.
[Similar command]	mdir

●mput

[Function]	Writes a file to the CPU module. To write multiple files, specify the file name and extension within the file path name using a wild card (* or ?). When writing multiple files, transmission is checked for each file transfer.
[Specification format]	mput "source file path name"
[Example]	When writing all the files whose extensions are "USR": mput *.USR
[Note]	The storage destination file name is the same as that on the FTP client side. The transfer destination is the memory in which current parameter files are stored.

●open

[Function]	Connects to the FTP server by specifying the host name or IP address and port number of the FTP server.
[Specification format]	open "host name" [port number] open "IP address" [port number] <ul style="list-style-type: none"> • Host name: Host name set in the Microsoft® Windows® hosts file • IP address: CPU module IP address • Port number: Port number to be used
[Example 1]	When connecting to the FTP server by specifying a host name: open HOST
[Example 2]	When connecting to the FTP server by specifying an IP address: open 192.0.1.254
[Note]	Connection is also possible by specifying an IP address at startup of FTP.

●put

[Function]	Writes a file to the CPU module.
[Specification format]	put "source file path name" [destination file path name]
[Example 1]	When writing the MAINSEQ1.QDR file to the standard RAM with the same file name: put MAINSEQ1.QDR 3:\MAINSEQ1.QDR
[Example 2]	When writing the MAINSEQ.QDR file to the standard RAM with a different file name: put MAINSEQ.QDR 3:\MAINSEQ1.QDR
[Note]	<ul style="list-style-type: none">• If no directory is specified for the source file path name (FTP client side), the file on the current directory where FTP is started up and connected to the server is written.• When no destination file path name (FTP server side) is specified, the file is saved in the memory in which current parameter files are stored.

●pwd

[Function]	Displays the current directory name of the CPU module.
[Specification format]	pwd
[Note]	"\" is displayed as the execution result of the "pwd" command.

●quit

[Function]	Disconnects the line from the FTP server and terminates the FTP session.
[Specification format]	quit
[Identical command]	bye

●quote

[Function]	Sends an FTP server subcommand (a subcommand dedicated to CPU modules).
[Specification format]	quote
[Example]	quote password-lock
[Note]	Only CPU module dedicated subcommands can be specified. Refer to (2).

●rename

[Function]	Renames a CPU module file.
[Specification format]	rename "old file path name" "new file path name" (abbreviated to "ren")
[Example]	When renaming a file stored in the standard RAM: rename 3:\MAINSEQ1.QDR 3:\SEQ1OLD.QDR
[Note]	Either of the following response codes is displayed upon completion. 350 Need more info. 250 Rename successful.

●user

[Function]	Inputs the user name and password of the connected FTP server.
[Specification format]	user "user name" [FTP password] <ul style="list-style-type: none">• User name: Login name set with a CPU module parameter• FTP password: FTP password set with a CPU module parameter
[Example 1]	When specifying a user name: user CPU
[Example 2]	When specifying a user name and password: user CPU CPU

(2) CPU-module-dedicated subcommands

The CPU-module-dedicated subcommands affixed to an FTP command, "quote," are described below.

●change

[Function]	Displays or changes the attribute of a file stored in the CPU module.
[Specification format 1]	When displaying the file attribute: quote change "file path name" Either of the following is displayed as an execution result upon completion. <ul style="list-style-type: none"> • When the specified file is read-only: ----- R • When the specified file is writable and readable: ----- W
[Specification format 2]	When changing the file attribute: quote change "file path name" "attribute" Use either of the following to specify the attribute. <ul style="list-style-type: none"> • To change it to a read-only file: r • To change it to a writable and readable file: w
[Example 1]	When displaying the attribute of the file stored in the standard RAM: quote change 3:\MAINSEQ1.QDR
[Example 2]	When changing the attribute of the file stored in the standard RAM: quote change 3:\MAINSEQ1.QDR r

●keyword-set

[Function]	Sets a file password registered in the file transfer target file into the Built-in Ethernet port QCPU.*1 Or, displays/clears the password set for FTP settings in parameter.
[Specification format]	quote keyword-set [File password] <ul style="list-style-type: none"> • File password :Specify the file password registered in the Built-in Ethernet port QCPU file. To clear the set file password, specify "*****". One of the following is displayed as the execution result upon normal completion. <ul style="list-style-type: none"> • When setting a file password : 200 Command successful • When displaying a file password: 200 Keyword is "File password" • When clearing a file password : 200 Command successful
[Example 1]	When setting the password (1234) quote keyword-set 1234
[Example 2]	When displaying the password currently set for FTP settings in parameter quote keyword-set
[Example 3]	When clearing the password currently set for FTP settings in parameter quote keyword-set ****
[Note]	<ul style="list-style-type: none"> • One file password can be set for the FTP settings in Built-in Ethernet port QCPU parameter. When the file targeted for file transfer changes, re-set the file password of the target file when a file password is also registered for the change target file. • When logging in to the Built-in Ethernet port QCPU, the file password is initialized (cleared) to "*****".

*1 : This command is used only when a file password is registered in the file transfer target file. The QnUDE(H)CPU checks a file password when a specified file is accessed.

●password-unlock

[Function]	Specify the remote password set in the CPU module to unlock the password.* ²
[Specification format]	quote password-unlock [remote password] <ul style="list-style-type: none">• Remote password: Specify the remote password set with a parameter for the CPU module. The following is displayed as an execution result upon completion. 200 Command Okey The following is displayed if the entered remote password does not match the setting. 556 Password Error The following is displayed if another command is requested before unlock processing of the remote password. 555 Password Locked
[Example]	When specifying a remote password (1234): quote password-unlock 1234
[Note]	<ul style="list-style-type: none">• The remote password is locked when you log in if the remote password check is enabled for the FTP communication port.• By executing this command before various FTP operations, the password is unlocked, allowing file operations of the CPU module.• Unlock processing will be completed if the remote password is unlocked when the remote password check is disabled for the FTP communication port.

*2 Use this command only when the FTP communication port is specified as a remote password check target.

●password-lock

[Function]	Locks the remote password set in the CPU module.* ³
[Specification format]	quote password-lock The following is displayed as an execution result upon completion. 200 Command Okey
[Example]	When locking the remote password: quote password-lock

*3 Use this command only when the FTP communication port is specified as a remote password check target.

●run

[Function]	Changes the CPU module state to RUN. (Remote RUN.) At this time, device memory clear can be specified.
[Specification format]	quote run [mode [clear mode]] <ul style="list-style-type: none"> • Mode: Specify whether to force remote RUN or not. <ul style="list-style-type: none"> 0: Normal RUN (default) 1: Forced RUN • Clear mode: <ul style="list-style-type: none"> Specify the CPU module device memory clear (initialization) processing performed when the operation starts by executing remote RUN. 0 : Do not clear device (default) 1 : Clear other than the latch range 2 : Clear all including the latch range <p>The following message is displayed as an execution result upon completion.</p> <p>200 Command successful</p>
[Example 1]	When executing remote RUN, with "Normal RUN" and "Do not clear device" specified: quote run
[Example 2]	When executing remote RUN, with "Forced RUN" and "Do not clear device" specified: quote run 1
[Example 3]	When executing remote RUN, with "Forced RUN" and "Clear other than the latch range": specified quote run 1 1
[Note]	<ul style="list-style-type: none"> • Forced RUN should be used only to force remote RUN to the CPU module from another device when a device that executed remote STOP to the CPU module have a problem and cannot perform remote RUN. With Normal RUN, the CPU module state cannot be changed from STOP/PAUSE to RUN if the previous state is set by a different device. • Specify the clear mode at the start of an operation, according to the system arrangement. After completing the specified clear processing, the CPU module runs in accordance with the PLC parameter settings ("Initial Device value" setting on the PLC file tab).

●status

[Function]	Displays information of the CPU module operation. This command is used to check the information on the CPU module operation before transferring a file to the CPU module.
[Specification format]	quote status One of the following is displayed as an execution result upon completion. <ul style="list-style-type: none"> • When the CPU module is in RUN state: "RUN" • When the CPU module is in STOP state: "STOP" • When the CPU module is in PAUSE state: "PAUSE"

●stop

[Function]	Changes the CPU module state to STOP (remote STOP).
[Specification format]	quote stop The following message is displayed as an execution result upon completion. 200 Command successful
[Note]	Before writing data to the program memory, set the CPU module into the STOP state using this command.

●pm-write

- [Function] Transfers program cache memory to the program memory.
- [Specification format] quote pm-write
The following message is displayed as an execution result upon completion.
200 Command successful
- [Note] Before writing data, set the CPU module to STOP.

●passwd-rd

- [Function] Set the read password (file password 32) that has been registered in a target file to transfer in the CPU module.
Displays and clears the read password that has been set in the CPU module.
Use this command only when a read password has been registered in the target file to transfer.
The CPU module checks a password when a specified file is accessed.
- [Specification format 1] When setting a read password in the CPU module
quote passwd-rd "read password"
The following message is displayed as an execution result at the normal end.
• 200 Command successful
- [Specification format 2] When displaying the read password that has been set in the CPU module
quote passwd-rd
Any of the following is displayed as an execution result at the normal end.
• When the command is executed with a read password set: 200 Read-password is "read password."
• When the command is executed with no read passwords set: 200 Read-password is not set.
- [Specification format 3] When clearing the read password that has been set in the CPU module
quote passwd-rd c or quote passwd-rd C
The following message is displayed as an execution result at the normal end.
• 200 Command successful
- [Note] • A read password can be set for the CPU module.
When the target file to transfer is changed, re-set a read password for a new target file if the read password has been registered in the new file.
• Re-set a read password before accessing the target file to transfer because the password that has been set using the command is initialized or cleared upon login to the CPU module.

●passwd-wr

[Function]	<p>Sets the write password (file password 32) that has been registered in a target file to transfer in the CPU module.</p> <p>Displays and clears the write password that has been set in the CPU module.</p> <p>Use this command only when a write password has been registered in the target file to transfer.</p> <p>The CPU module checks a password when a specified file is accessed.</p>
[Specification format 1]	<p>When setting a write password in the CPU module</p> <pre>quote passwd-wr "write password"</pre> <p>The following message is displayed as an execution result at the normal end.</p> <ul style="list-style-type: none"> • 200 Command successful
[Specification format 2]	<p>When displaying the write password that has been set in the CPU module</p> <pre>quote passwd-wr</pre> <p>Any of the following messages is displayed as an execution result at the normal end.</p> <ul style="list-style-type: none"> • When the command is executed with a write password set: 200 Write-password is "write password." • When the command is executed with no write passwords set: 200 Write-password is not set.
[Specification format 3]	<p>When clearing the write password that has been set in the CPU module</p> <pre>quote passwd-wr c or quote passwd-wr C</pre> <p>The following message is displayed as an execution result at the normal end.</p> <ul style="list-style-type: none"> • 200 Command successful
[Note]	<ul style="list-style-type: none"> • A write password can be set for the CPU module. <p>When the target file to transfer is changed, re-set a write password for a new target file if the write password has been registered in the new file.</p> <ul style="list-style-type: none"> • Re-set a write password before accessing the target file to transfer because the password that has been set using the command is initialized or cleared upon login to the CPU module.


9.5 Precautions

(1) FTP clients

- Some FTP clients may have FTP command specifications different from those described in this manual. In such a case, check the functions and operation methods, referring to the manuals for the FTP client.
- An FTP operation from Microsoft® Internet Explorer is not allowed. If it is attempted, an error will occur in Internet Explorer.


(2) CPU module side processing

- Files only in the drives of the host station CPU module can be accessed.
- Files may be in failure. Stop accessing the memory card or SD memory card before turning off the power supply, resetting the module, or unloading the memory card or SD memory card. By using SM606 (SD memory card forced disable instruction), accesses to the SD memory card can be disabled at a time. For details on the SD memory card forced disable instruction, refer to the following.

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

- During file access, do not operate the file from a peripheral such as a programming tool. (Also do not perform online operations such as online change, scan time measurement, registration of the step number set for a monitoring condition). If the file is operated during operation of the FTP function, an error may occur in the peripheral. Perform the processing suspended due to an error again after performing the FTP function.
- An error will occur if an FTP client tries to connect while the backup or restoration using a memory card is being performed for the CPU module change function. Execute the connection again after the end of the backup or restoration. An error will also occur if the backup or restoration is performed while an FTP client is being connected. Execute the backup or restoration again after disconnecting the FTP client.

(3) Communication processing

- If a timeout error occurs during file transfer, the TCP connection will be closed (disconnected). To restart the file transfer, log in to the CPU module once again from the FTP client.
- For each FTP connection, the existence of the target device is checked. For details, refer to:  Page 31, Section 3.5
- The processing time for file transfer depends on the factors such as the Ethernet line congestion, the number of connections simultaneously used (communication processing of other connections), and the system configuration.
- Only one FTP client can log in to a CPU module at the same time. If another FTP client attempts to connect to a CPU module that is already in the logged-in state, the connection cannot be established, resulting in an error.
- An attempt to execute another communication (MELSOFT connection or MC protocol) using UDP during file transfer using FTP may cause an error such as a timeout error. Execute it after completion of the file transfer, or use TCP.

(4) File writing

- An existing file cannot be overwritten and saved.
Either delete a file with the file delete command (delete or mdelete) or rename the file with the file rename command (rename), and then write the file.
- Writing is not allowed to read-only files and files locked by a function other than FTP. If attempted, a write error occurs.
- When a memory card or SD memory card is protected, a file cannot be transferred (written) into the card. If the file transfer is executed, an error occurs.
- When a file is written, a temporary file (FTP_I***.TMP) is created automatically. This file is renamed to the destination file name upon completion. However, if power failure or reset of the CPU module occurs during the file writing, this temporary file may remain. If this occurs, delete the temporary file.
- Before writing or deleting data to the file register in the standard RAM, set the CPU module to STOP.
- When the file register in the standard RAM is set as an auto refresh device, do not perform any writing or deletion to the corresponding drive.
- When writing a large file to a memory card or SD memory card, set the CPU module to STOP. If writing is performed in the RUN status, a communication error may occur.



(5) File deletion

- The timing for deleting files must be determined by the user, considering the overall system including the CPU module and programming tool.
- When a memory card or SD memory card is protected, no files can be deleted. An error occurs if a file is deleted.

(6) Password for FTP

When you forgot the FTP password, set the FTP parameters again by the following steps.

- 1. Read out the parameters from the CPU module to the programming tool.**
- 2. In the FTP Parameter Setting window, click the "Default" button to return all the FTP parameters to default values.**

 Project window ⇒ [Parameter] ⇒ [PLC Parameter] ⇒ [Built-in Ethernet Port Setting]
⇒  button

- 3. Configure the FTP parameter settings again.**
- 4. Write the parameters to the CPU module from the "Write to PLC" window.**

 [Online] ⇒ [Write to PLC]

- 5. Power off and then on or reset the CPU module to enable the parameters.**

CHAPTER 10 REMOTE PASSWORD

A remote password is checked when a connection is requested for the following.

- Communication using a programming tool
- Communication using MC protocol
- File transfer (FTP)

Point

The remote password function is one of the preventive methods against unauthorized access (e.g. destruction of data and programs) from external devices.

However, this function cannot completely prevent unauthorized access.

Other preventive measures should be taken at users' discretion if security of the programmable controller system needs to be maintained against unauthorized access from external devices. We cannot be held responsibility for any problems caused by unauthorized access.

[Examples of measures against unauthorized access]

- Install a firewall.
 - Set up a personal computer as a relay station, and control the relay of communication data using an application program.
 - Set up an external device that can control access rights as a relay station.
For devices that can control access rights, please consult your network service provider or networking equipment vendors.
-

10.1 Communication Using Remote Password

Communication is performed in the order described below when a remote password is set for the CPU module.

(1) Allowing access (unlock processing)

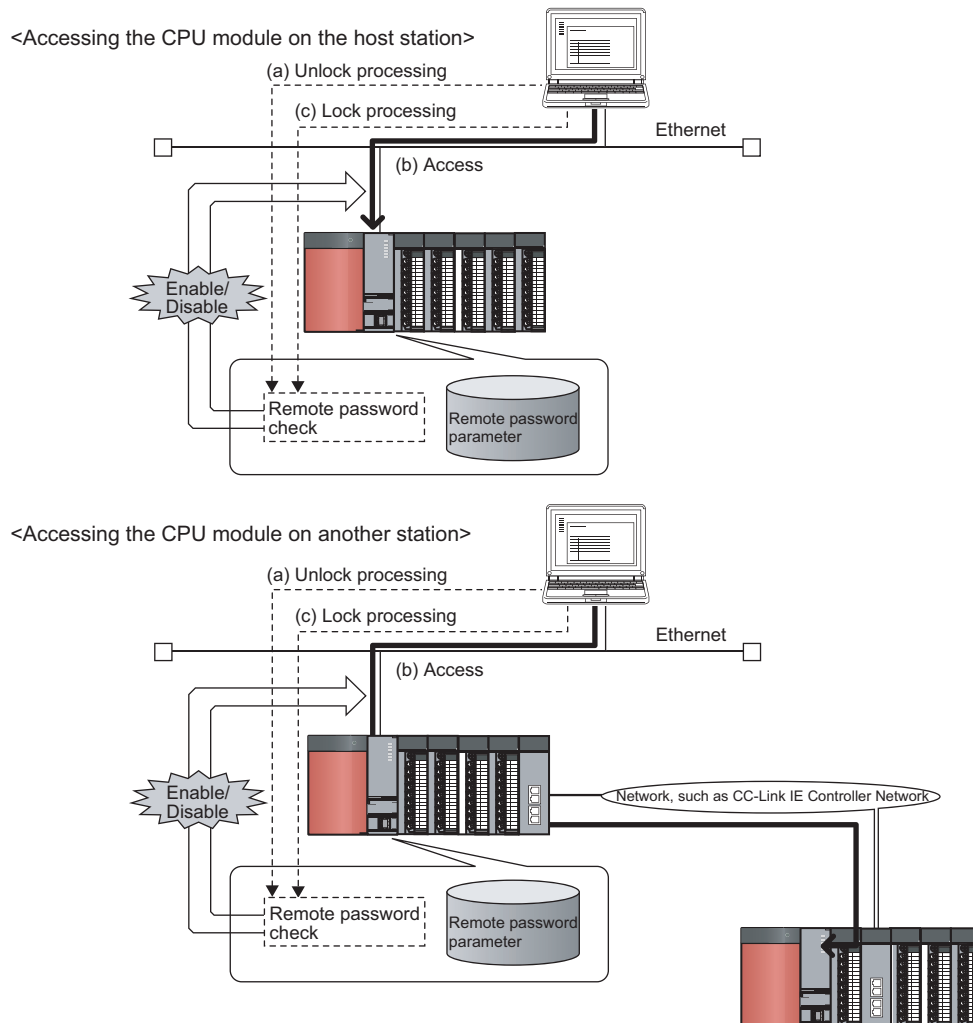
On a communication device such as a personal computer, unlock the remote password set for the CPU module. If it is not unlocked, an error will occur on the connected device because the CPU module will prohibit any access.

(2) Access processing

Access the CPU module after completion of the remote password unlock processing.

(3) Prohibiting access (lock processing)

When terminating access from the personal computer, lock the remote password to prohibit an access from any other personal computers.

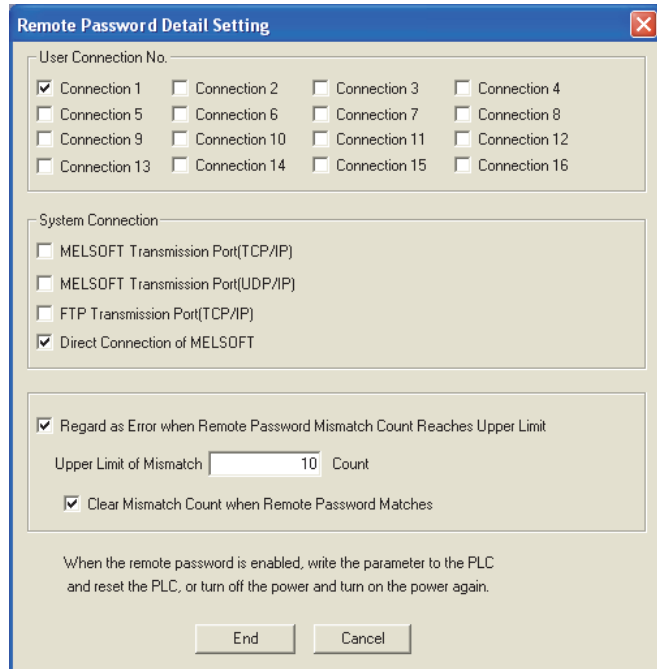
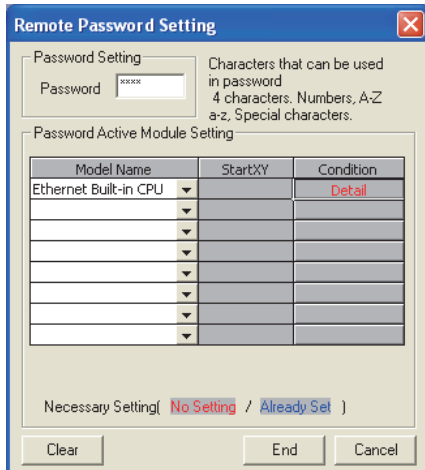



10.2 Remote Password Setting

(1) Setting a remote password

Set a remote password and a target connection in the programming tool, and write the data to the CPU module.

 Project window ⇨ [Parameter] ⇨ [Remote Password]



Item	Description	Setting range	
Password Setting	Enter a remote password to be set for the CPU module.*1	Up to four characters	
Password Active Module Setting	Model Name	Select a CPU model to enable the remote password for the built-in Ethernet port of the CPU module. Only one CPU model is selectable.	
	Condition	Click this to display the "Remote Password Detail Setting" window.	
User Connection No.*5	Select one of them when the remote password is to be enabled for the built-in Ethernet port. (Setting of an unused connection or MELSOFT connection is ignored.)	Select the checkboxes appropriate to the target connection.	
System Connection*6	MELSOFT Transmission Port (TCP/IP)*2		Select one of them when the remote password is to be enabled for the built-in Ethernet port.
	MELSOFT Transmission Port (UDP/IP)*2*3		
	FTP Transmission Port (TCP/IP)		
	Direct Connection of MELSOFT*4		
Regard as Error when Remote Password Mismatch Count Reaches Upper Limit	Select this when enabling this operation. (Useful for detecting unauthorized access)  Page 139, Section 10.4)		
Upper Limit of Mismatch	Specify the maximum number of mismatches.	1 to 65535	
Clear Mismatch Count when Remote Password Matches	Select the checkbox when this operation is to be performed.	-	

- *1 One-byte alphanumeric and special characters can be used for remote password entry.
(Case-sensitive)
- *2 To enable the remote password for the port for which the open system is set to "MELSOFT Connection" in PLC Parameter, select the following checkbox.
When Protocol is set to "TCP" → "MELSOFT Transmission Port (TCP/IP)".
When Protocol is set to "UDP" → "MELSOFT Transmission Port (UDP/IP)".
- *3 When connecting the CPU module and a GOT via Ethernet, do not select "MELSOFT Transmission Port (UDP/IP)".
- *4 Select this checkbox to enable the remote password for the CPU module that is directly connected to the programming tool using the built-in Ethernet port. (☞ Page 33, CHAPTER 4)
- *5 User connection is for users for communications such as MC protocol communications and communications using fixed buffers.
- *6 System connection is used by the system for communications such as FTP communications and MELSOFT communications (TCP/IP, UDP/IP).

(2) Writing to the CPU module

Write the remote password to the CPU module from the "Write to PLC" window.

☞ [Online] ⇄ [Write to PLC]

After writing the parameters to the CPU module, power off and then on or reset the CPU module to enable the parameters.

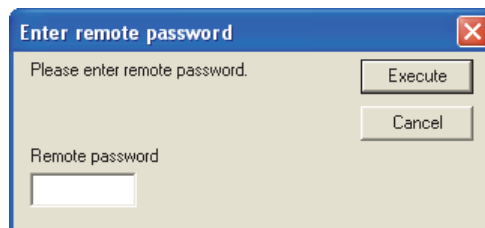
(3) Unlocking or locking the remote password

The remote password is unlocked or locked from an external device such as a personal computer, as described below.

(a) When using MELSOFT connection

Enter a remote password in the following window that appears during communication.

When the remote password is entered, the programming tool performs unlock processing and then accesses the CPU module



(b) When using MC protocol

Use commands dedicated to MC protocol. (☞ Page 39, Section 5.2.1)

(c) When using the FTP function

Use the "password-lock" and "password-unlock" commands. (☞ Page 120, Section 9.4.1)

10.3 Precautions

(1) When a remote password is set for UDP connections

- Determine a target device before data communication. (The communication target needs to be determined because, after unlocking of the remote password, communication is available with any other devices.)
- At the end of data communication, always lock the remote password.
(If the lock processing is not performed, the unlock state is held until a timeout occurs. No communication for 10 minutes causes a timeout, and the CPU module automatically performs lock processing.)

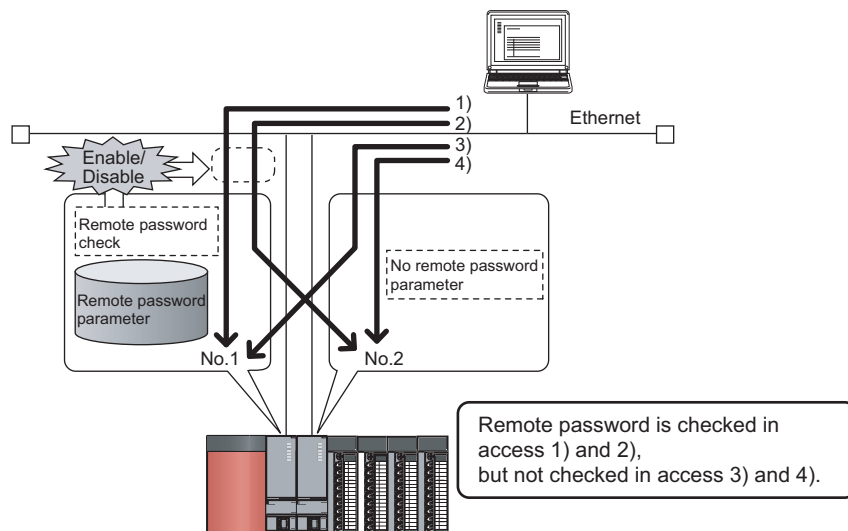
To prevent unauthorized access using the remote password setting, it is recommended to set all connection protocols to TCP/IP and disable direct connection with the parameter.

(2) When a TCP/IP connection is closed before lock processing

The CPU module automatically performs lock processing.

(3) Remote password valid range

The remote password is valid only for access from the Built-in Ethernet port QCPU for which the parameter settings were made. When multiple CPU modules are used in a multiple CPU system, set a remote password for each setting target CPU module respectively.



10.4 Detection of Unauthorized Access and Actions

When the remote password mismatch count reaches the upper limit in unlock processing, "REMOTE PASS FAIL" (error code: 2700) is detected.

If this occurs, unauthorized access from the outside of the system can be considered as a cause of the error.


Take the following actions as needed.

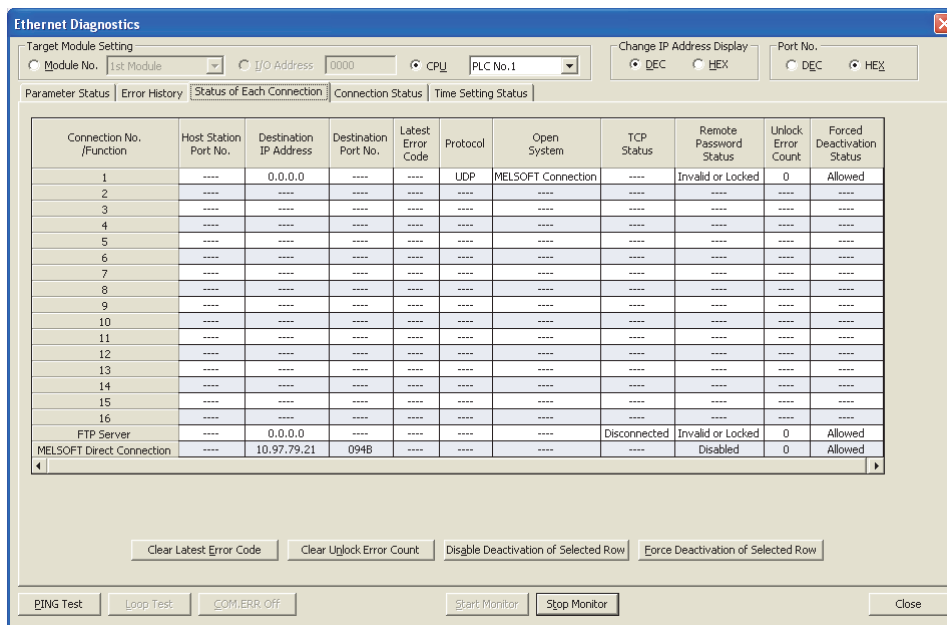
1. Monitor the Remote password count (SD979 to SD999) and identify the connection of which remote mismatch count has reached the upper limit in unlock processing.

2. Stop the communication by disabling the connection as shown below.

- Select the connection in the "Ethernet diagnostics" window, and force it to be deactivated.

( GX Works2 Version 1 Operating Manual (Common))

 [Diagnostics] ⇨ [Ethernet Diagnostics] ⇨ "Status of Each Connection"



- Turn on the force deactivation flag of the connection in the special register (SD1276, SD1277).

3. Clear the "REMOTE PASS FAIL" error (error code: 2700).

The remote password count (SD979 to SD999) is also cleared.

4. Inform your system administrator that the number of unlock processing failures exceeded the limit, and take appropriate actions.

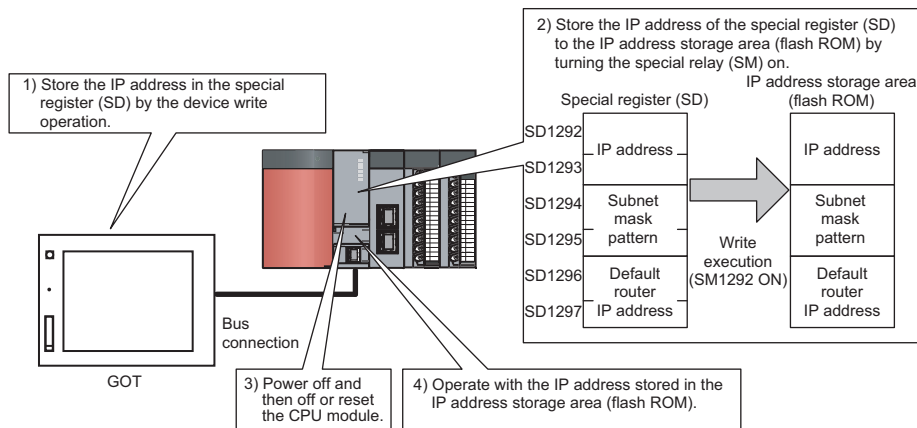
Point

If the error is detected due to frequent incorrect typing by authorized users, prevent this by the following operations.

- Enable "Clear Mismatch Count when Remote Password Matches" in the "Remote Password Detail Setting" window.
- Clear the accumulated count of remote password mismatches using the special relay (SM1273).


CHAPTER 11 IP ADDRESS CHANGE FUNCTION Note 11.1

The IP address of a built-in Ethernet port can be changed without changing the built-in Ethernet port settings of PLC parameters, by storing the values in special relays and special registers.




Point

This function can also be used by changing values for special relay and special register from GOT. For details of the special relay and special register used for this function, refer to the lists of the special relay and special register in the following.

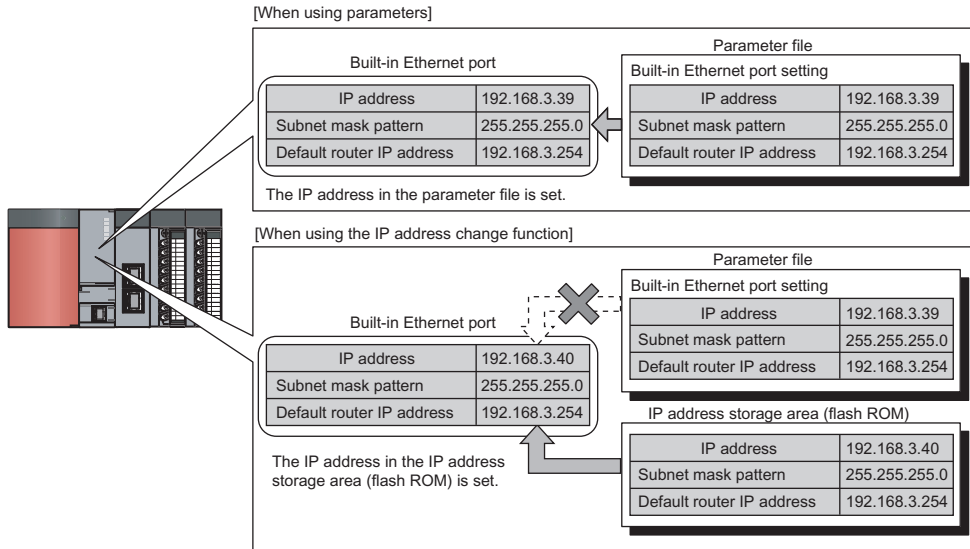
 QCPU User's Manual (Hardware Design, Maintenance and Inspection)

Note 11.1 Universal

To use the IP address change function for the QnUDE(H)CPU, check the version of the CPU module. ( Page 155, Appendix 3)

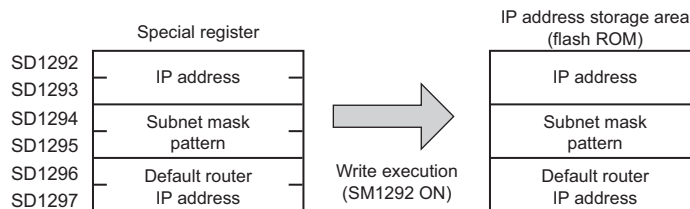
11.1 IP Address of the Built-in Ethernet Port

For IP address of the built-in Ethernet port, a value of the built-in Ethernet port setting of the PLC Parameter is set at the initial processing of the CPU module. When this function is used, the value stored in the IP address storage area (flash ROM) will be set to the IP address of the built-in Ethernet port which is set during the initial processing of the CPU module, instead of the value set in the parameter.



(1) Write and clear operation to the IP address storage area (flash ROM)

Write a value of IP address to the IP address storage area (flash ROM). The special relay and special register are used to perform the write and clear operations.



(2) Execution timing of writing to and clearing the IP address storage area (flash ROM)

Write and clear processing are executed in the END processing. Therefore, the scan time is increased during execution.

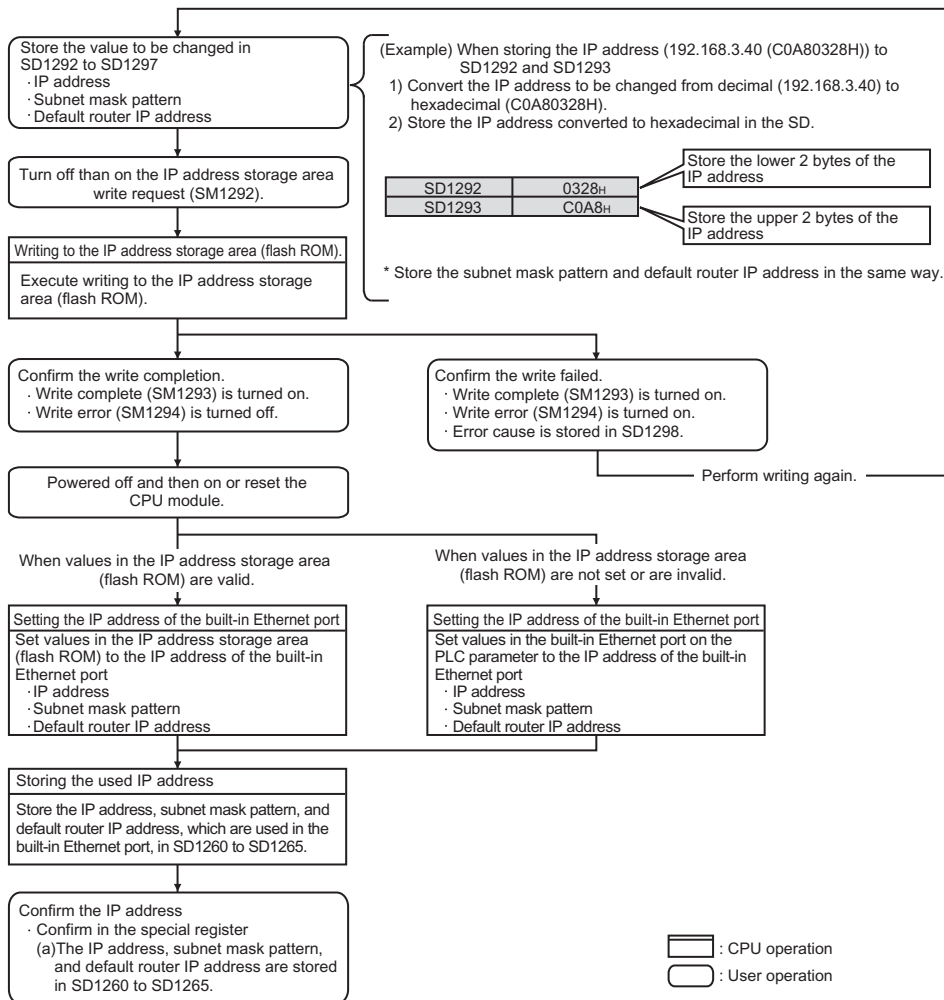
11.2 How to Use the Function

11.2.1 Write operation

This operation can be performed by storing the IP address to be changed in SD1292 to SD1297 and turning off and on SM1292 (IP address storage area write request).

(1) Operating procedure

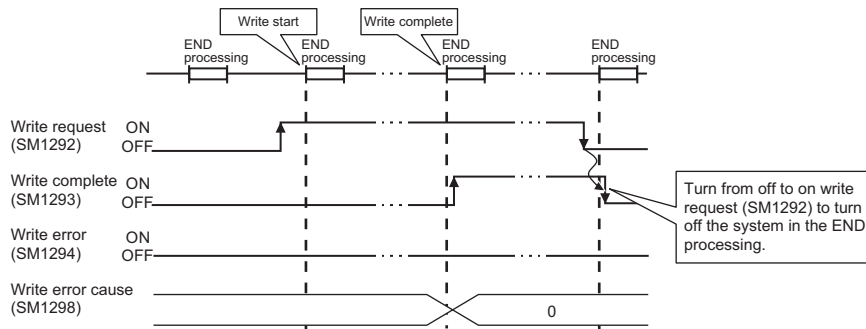
The following shows the write operation flow.



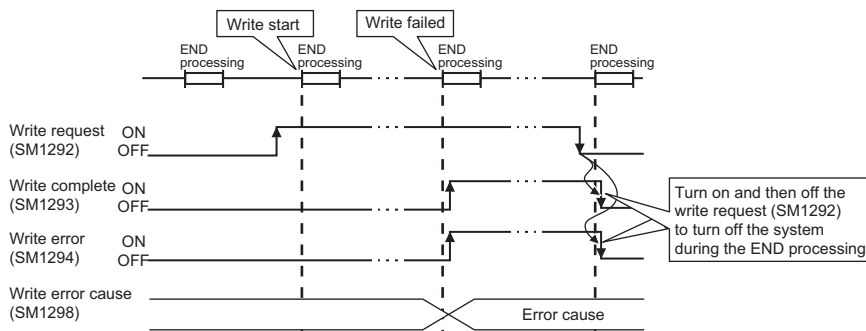
(2) Operations of special relay and special register

The following shows the operations of special relay and special register during the operation of writing to the IP address storage area (flash ROM).

(a) Operations of special relay and special register during the operation of writing to the IP address storage area (flash ROM)



(b) Operations of special relay and special register when the operation of writing to the IP address storage area (flash ROM) failed



(3) Cause of failure to write to the IP address storage area (flash ROM)

If the processing of writing to the IP address storage area (flash ROM) is not completed, the error cause is stored in SD1298 (IP address storage area write error factor).

Value in SD1298	Error cause
100 _H	The value in SD1292 to SD1297 is outside the setting range.
200 _H	An error occurred during writing.
300 _H	Writing cannot be performed since the following functions are being executed. <ul style="list-style-type: none"> • Online Program Change • Export to ROM format • Write to PLC (flash ROM)
400 _H	Writing is started during clear processing execution.

11.2 How to Use the Function
11.2.1 Write operation

(4) Program example

The following shows a program example of writing to the IP address storage area (flash ROM).

(a) Devices used in programs

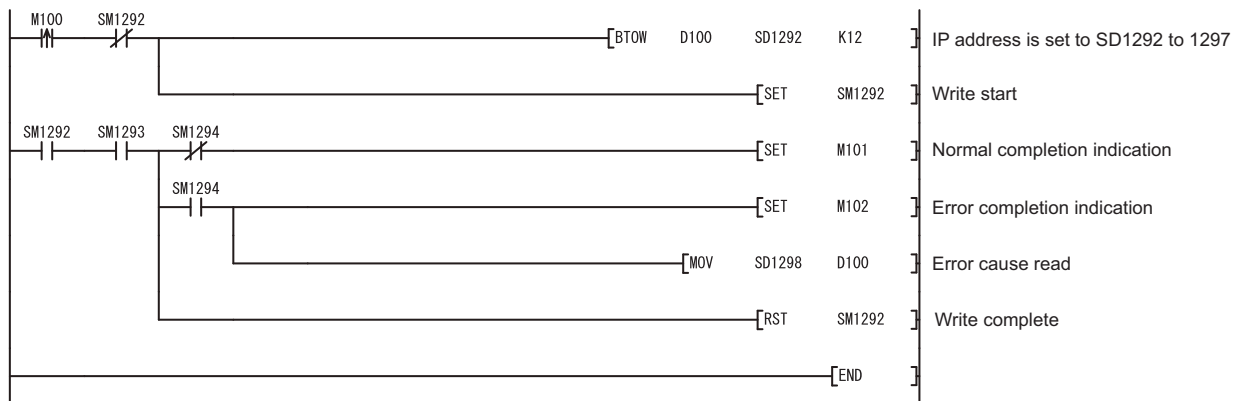
Device number	Application	Device number	Application
M100	Write Instruction	SM1293	IP address storage area write complete
D100 to D103*1	IP address to be changed	SM1294	IP address storage area write error
D104 to D107*1	Subnet mask pattern to be changed	M101	Normal write completion indication
D108 to D111*1	Default router IP address to be changed	M102	Abnormal write completion indication
SD1292 to SD1297	IP address setting	SD1298	IP address storage area write error factor
SM1292	IP address storage area write request	D100	Write error factor display

*1 Set as follows.

Example When setting the IP address 192.168.3.40 in D100 to D103

D100	40
D101	3
D102	168
D103	192

(b) Sample program

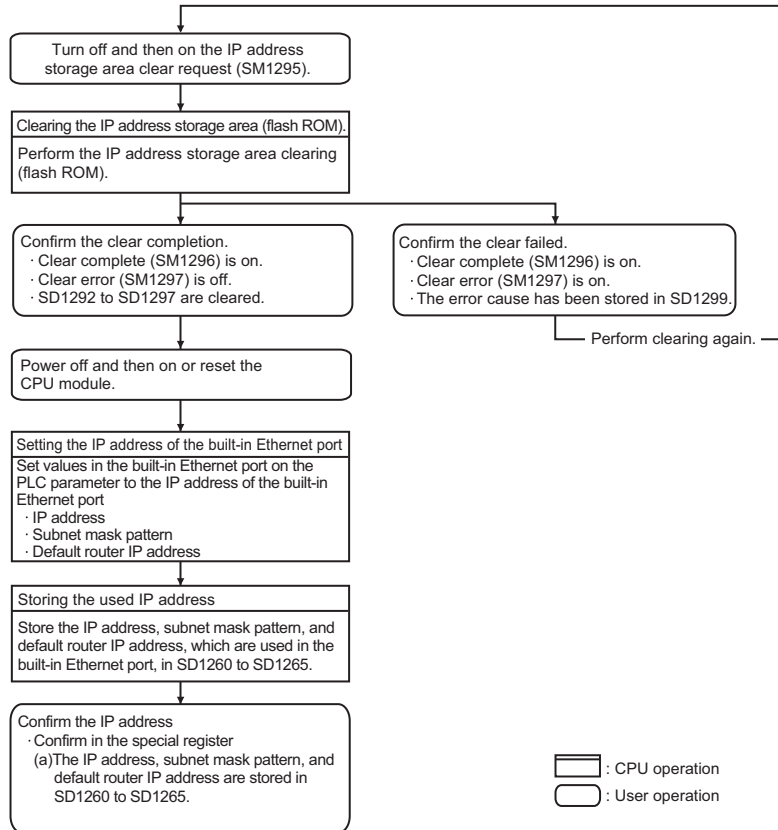


11.2.2 Clear operation

This operation can be performed by turning off and on SM1295 (IP address storage area clear request).

(1) Operating procedure

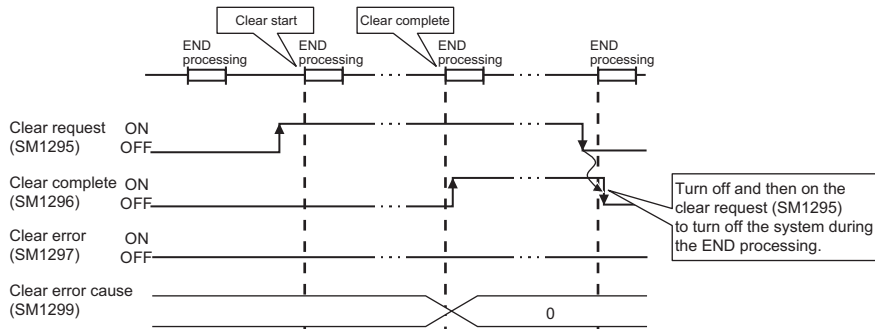
The following shows the clear operation flow.



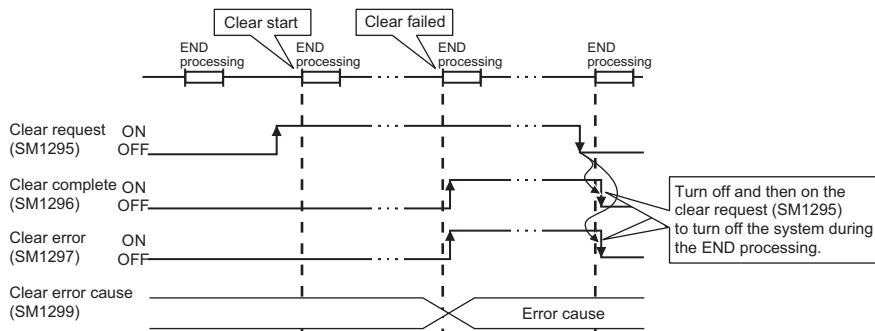
(2) Operations of special relay and special register

The following shows the operations of special relay and special register during the operation of clearing the IP address storage area (flash ROM).

(a) Operations of special relay and special register during the operation of clearing the IP address storage area (flash ROM)



(b) Operations of special relay and special register when the operation of clearing the IP address storage area (flash ROM) failed



(3) Cause of failure to clear the IP address storage area (flash ROM)

If the processing of clearing the IP address storage area (flash ROM) is not completed, the error cause is stored in SD1299 (IP address storage area clear error factor).

Value in SD1299	Error cause
200 _H	An error occurred during clearing.
300 _H	Clearing cannot be performed since the following functions are being executed. <ul style="list-style-type: none"> • Online Program Change • Export to ROM format • Write to PLC (flash ROM)
400 _H	Clearing is started during write processing execution.

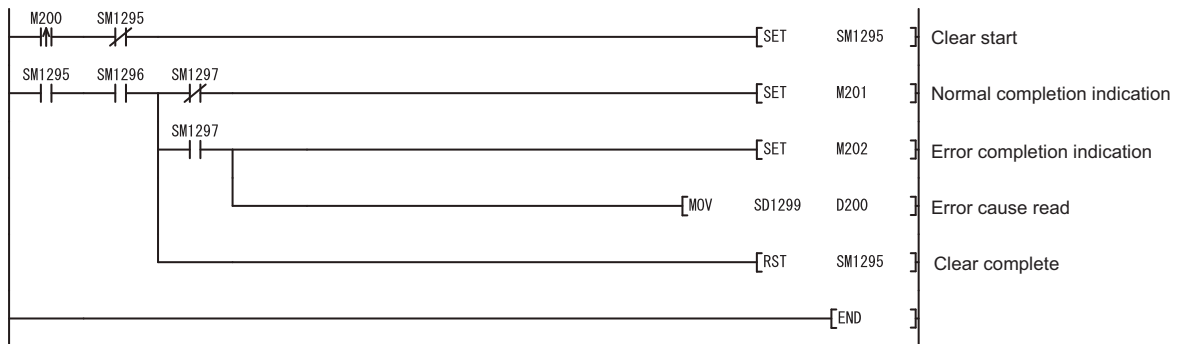
(4) Program example

The following shows a program example of clearing the IP address storage area (flash ROM).

(a) Devices used in programs

Device number	Application	Device number	Application
M200	Clear instruction	M201	Normal clear completion indication
SM1295	IP address storage area clear request	M202	Abnormal clear completion indication
SM1296	IP address storage area clear completion	SD1299	IP address storage area clear error factor
SM1297	IP address storage area clear error	D200	Clear error factor display

(b) Sample program



11.3 Checking the IP Address

(1) Checking using the Ethernet diagnostics

The IP address of the built-in Ethernet port can be checked with the Ethernet diagnostics. For details of the Ethernet diagnostics, refer to the following.

 GX Works2 Version 1 Operating Manual (Common)

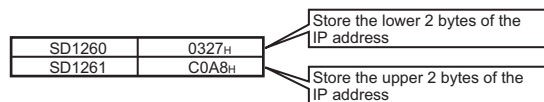
(2) Checking by the special relay and special register

The IP address of the built-in Ethernet port can be checked using the special relay and special register.

- IP address: SD1260 to SD1261
- Subnet mask pattern: SD1262 to SD1263
- Default router IP address: SD1264 to SD1265

*1 Stored as follows.

Example When IP address is 192.168.3.39 (C0A80327_H)



11.4 Precautions

The following describes precautions for the IP address change function.

(1) Power-off and reset operations

Do not power off and reset the CPU module while writing to or clearing the IP address storage area (flash ROM). Values may not be reflected to the IP address storage area (flash ROM). Check that SM1293 (IP address storage area write complete) and SM1296 (IP address storage area clear complete) are started before powering off and resetting the CPU module.

(2) IP address of parameters

For IP address of the built-in Ethernet port, values in the IP address storage area (flash ROM) take priority over values of parameters. To use the IP address specified by parameters, clear the value in the IP address storage area (flash ROM).

(3) Duplicated IP addresses

When changing an IP address, check that the changed IP address is not the same as that of other devices. If IP addresses are duplicated, communication may be performed with a wrong device. The duplicated status can be checked by either of the following methods.

- By using the Find CPU function
- By disconnecting an external device from the network and performing a PING test to the IP address of the disconnected external device (The IP address is duplicated if a response message is returned.)

(4) Functions that cannot be performed during write processing and clear processing

The following functions cannot be executed during the write or clear processing to the IP address storage area. Do not execute the functions before the processing is completed. Doing so may cause an error.

If operation of writing to or clearing the IP address storage area (flash ROM) is executed during the execution of the following functions, the write or clear operation is processed as an error.

- Online Program Change
- Export to ROM format
- Write to PLC (flash ROM)

(5) Execution timing of write processing and clear processing

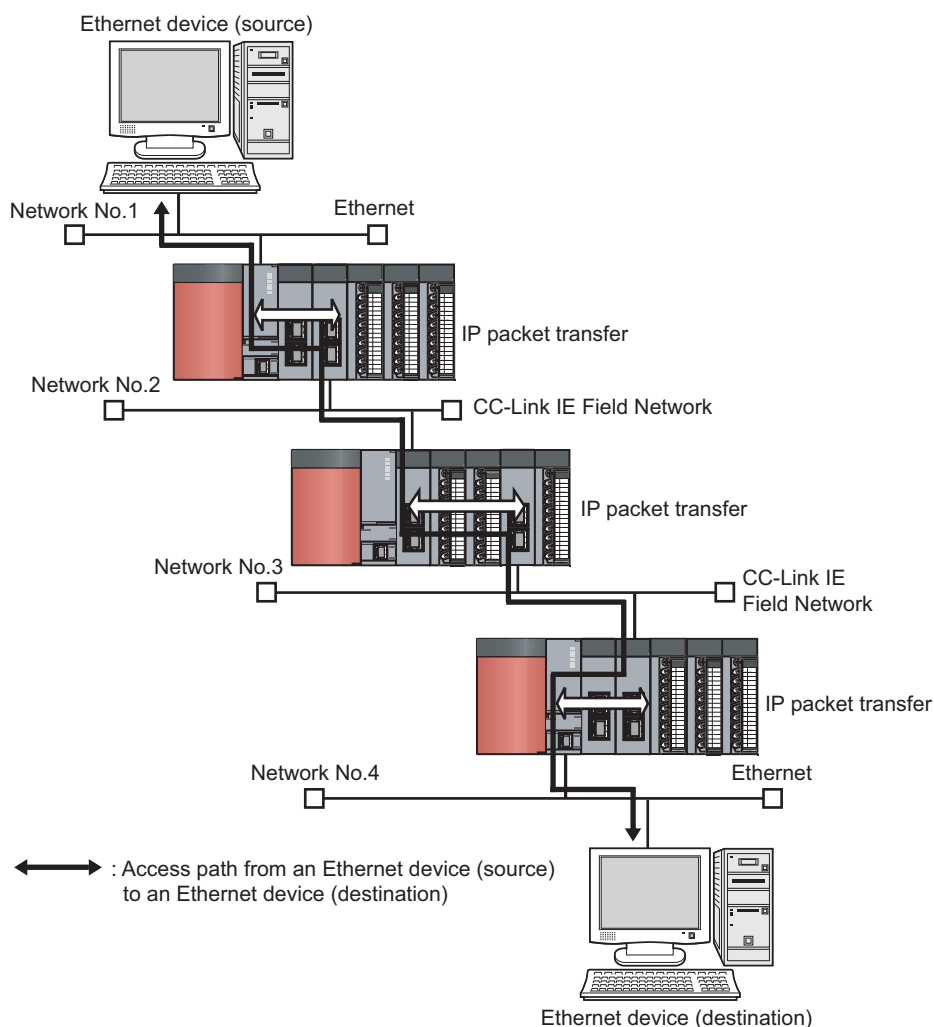
- Since the status on SM1292 (IP address storage area write request) and SM1295 (IP address storage area clear request) are checked in the END processing, processing of writing to or clearing the IP address storage area (flash ROM) cannot be executed if a contact is turned off and on, or turning on and off in one scan.
- If SM1292 (IP address storage area write request) is turned off and on again while writing to the IP address storage area (flash ROM), the write processing executed earlier is completed properly and the processing executed later is ignored. (The same occurs for the clear operation.)
- If SM1295 (IP address storage area clear request) is turned off and on while writing to the IP address storage area (flash ROM), the clear operation is processed as an error. (The same occurs when the write operation is executed during the clear processing execution.)
- If SM1292 (IP address storage area write request) and SM1295 (IP address storage area clear request) are turned off and on in one scan, the write operation is prior to be executed, and the clear operation is processed as an error.

CHAPTER 12 IP PACKET TRANSFER

FUNCTION Note 12.1

Communications can be performed with a device which supports the following IP addresses, which have been specified via a CC-Link IE Controller Network module or CC-Link IE Field Network module, using a protocol such as the FTP or HTTP via a built-in Ethernet port from an Ethernet device such as a personal computer.


- External devices on CC-Link IE Controller Network or CC-Link IE Field Network
- External devices on the Ethernet network, which are connected through the built-in Ethernet ports



Point



This function is supported only by GX Works2. (It is not supported by GX Developer.)

Note 12.1 **Universal**

Check the versions of the CPU module and GX Works2 when using the IP packet transfer function for the QnUDE(H)CPU. ( Page 155, Appendix 3)

(1) How to use

For the settings of IP packet transfer function or how to use the function, refer to the following.

-  Manual for the CC-link IE Controller Network module used
-  Manual for the CC-link IE Field Network module used

(2) Precautions

- The data that are communicated using the IP packet transfer function are communicated separately by the following processing.
 - Service processing by a sequence scan of a CPU module
 - Link scan on CC-Link IE Controller Network
 - Link scan on CC-Link IE Field Network

The above processing causes communication speeds to be slower than Ethernet lines. Manuals for the CC-Link IE Controller Network module or CC-Link IE Field Network module describe targeted communication speeds when the IP packet transfer function is used.


- Broadcast communication and multicast communication cannot be performed with the IP packet transfer function. Perform unicast communication (communication with one request destination specified).
- If the application time-out of the request source device occurs due to a heavy communication load on the path that an IP packet takes, measure the response time using the ping command from the request source device and adjust the application timeout time.
- Data size must be within 1460 bytes when the ping command is used.
- UDP provides less reliable data communications compared with TCP and thus data may be lost or arrive out of order. Use TCP communications if a problem occurs with the UDP communications.

APPENDICES

Appendix 1 Operation Processing Time for Each Instruction

The following table lists the processing time of instructions described in this manual.

For details on the processing time, refer to the following.

 MELSEC-Q/L Programming Manual (Common Instruction)

Type	Instruction	Condition		Processing time(μs)			
				QnUDVCPU		QnUDE(H)CPU	
				Minimum	Maximum	Minimum	Maximum
Instructions for the socket communication function	SP.SOCOPEN	TCP	Active	14.900	34.800	18.500	40.900
			Unpassive	14.900	32.500	18.400	40.900
			Fullpassive	14.900	32.400	18.400	40.900
		UDP	14.900	34.600	18.400	40.900	
	SP.SOCCLOSE	TCP	From the host CPU	14.600	34.100	18.500	40.100
			From the connected device	14.600	34.000	18.500	40.000
		UDP	14.600	33.400	18.900	40.100	
	SP.SOCCRCV	TCP	Minimum data volume (1 byte)	6.400	25.000	17.000	39.100
			Maximum data volume (2046 bytes)	6.300	24.900	17.500	39.100
			Maximum data volume (10238 bytes)	6.200	24.700	17.500	39.100
		UDP	Minimum data volume (1 byte)	6.200	25.000	17.100	39.100
			Maximum data volume (2046 bytes)	6.300	25.000	17.500	39.100
			Maximum data volume (10238 bytes)	6.300	24.900	17.500	39.100
	S.SOCCRCVS	TCP	Minimum data volume (1 byte)	14.000	36.600	12.300	29.100
			Maximum data volume (2046 bytes)	37.900	66.700	243.400	259.100
			Maximum data volume (10238 bytes)	149.100	190.400	1168.600	1185.300
		UDP	Minimum data volume (1 byte)	14.200	36.500	12.800	30.100
			Maximum data volume (2046 bytes)	38.100	69.100	243.400	259.100
			Maximum data volume (10238 bytes)	153.800	191.800	1167.600	1185.300
	SP.SOCSND	TCP	Minimum data volume (1 byte)	11.700	34.500	18.900	43.100
			Maximum data volume (2046 bytes)	41.100	75.900	290.000	313.700
			Maximum data volume (10238 bytes)	177.600	235.800	1367.600	1407.300
		UDP	Minimum data volume (1 byte)	11.400	35.400	18.900	43.100
			Maximum data volume (2046 bytes)	41.800	76.800	290.000	313.700
			Maximum data volume (10238 bytes)	189.800	235.900	1367.600	1407.300
	SP.SOCCINF		-	4.900	20.800	12.700	32.200
	SP.SOCCSET		-	4.200	19.200	10.700	29.200
SP.SOCCRMODE	Standard mode → Fixed-length mode		8.500	19.400	9.700	27.200	
	Fixed-length mode → Standard mode		8.300	19.400	9.700	27.200	
SP.SOCCRDATA	Minimum data volume (1 word)		4.400	19.200	9.700	27.200	
	Maximum data volume (1024 words)		28.100	51.800	241.700	258.200	
	Maximum data volume (5120 words)		144.200	173.800	1168.600	1184.300	

Type	Instruction	Condition	Processing time(μ s)			
			QnUDVCPU		QnUDE(H)CPU	
			Minimum	Maximum	Minimum	Maximum
Instruction for the predefined protocol function	SP.ECPRTCL	-	15.900	50.300	-	-

A

Appendix 2 Port Numbers Used by Built-in Ethernet Port QCPU

Do not specify the following port numbers, because these numbers are used by the system.

Port number	Application
1388 _H (5000)	For future extension (For Ethernet modules, this port number is used for "Auto Open UDP Port".)
1389 _H (5001)	For future extension (For Ethernet modules, this port number is used for "MELSOFT application transmission port (TCP/UDP), data link instructions".)
138A _H (5002)	For future extension (For Ethernet modules, this port number is used for "MELSOFT application transmission port (TCP/IP)".)
138B _H (5003) to 138D _H (5005)	For future extension
138E _H (5006)	MELSOFT communication port (UDP/IP)
138F _H (5007)	MELSOFT communication port (TCP/IP)
1390 _H (5008)	MELSOFT direct connection port
1391 _H (5009)	For future extension

Appendix 3 Added and Changed Functions

The following table lists added or modified functions in the CPU module and programming tool for built-in Ethernet port communications and the corresponding serial numbers of CPU modules and software versions of GX Works2 software.

×: Not available, -: A function not related to the programming tool

Added function	Corresponding function version	The first five digits of the corresponding serial number	The corresponding version of the programming tool	
			GX Works2	GX Developer
Socket communication function (☞ Page 60, CHAPTER 7)	B	"11012" or later	1.11M or later	8.78G or later
IP address change function (☞ Page 140, CHAPTER 11)		"11082" or later	-	-
Data up to 10238 bytes can be exchanged with the SP.SOCSND/S(P).SOCRCV(S)/S(P).SOCRDATA instructions (☞ Page 60, CHAPTER 7)		"12052" or later		
A-compatible IE frame for communication using the MC protocol (☞ Page 44, Section 5.3)		"13102" or later		
IP packet transfer function (Available for CC-Link IE Field Network) (☞ Page 150, CHAPTER 12) ^{*2}		"14022" or later	1.77F or later	×
IP packet transfer function (Available for CC-Link IE Controller Network) (☞ Page 150, CHAPTER 12) ^{*2}			1.98C or later	
File registers available for the A-compatible 1E frame in the MC protocol (☞ Page 42, Section 5.2.2 (2)) ^{*1}		*3	-	-
Predefined protocol function (☞ Page 49, CHAPTER 6)		"15103" or later	1.501X or later	×

*1 This function cannot be used for some models. For the availability of the function, refer to each reference.

*2 For the versions of the intelligent function modules that support the function, refer to the manual for the intelligent function module used.

*3 Available for the CPU modules with the following serial number (first five digits).

- QnUDE(H)CPU: "14112" or later
- QnUDVCP: "15043" or later

Appendix 4 Specifications Comparison with Ethernet Module

(1) Specifications comparison with the Ethernet module

The following table lists the comparison of specifications between the Built-in Ethernet port QCPU and the Ethernet module (QJ71E71-100).

○ : Available, △ : Available but partially restricted, × : Not available

Item		Description	Availability	
			Built-in Ethernet port QCPU	QJ71E71-100
MC protocol communication	4E frame	<ul style="list-style-type: none"> Reads/writes data in the CPU module from/to an external device. A frame format that can receive multiple request messages at a time. 	×	○
	QnA-compatible 3E frame	Reading/Writing data in device memory	○ ^{*1*8}	○
		Other	×	○
	A-compatible 1E frame	<ul style="list-style-type: none"> Reads/writes data in the CPU module from/to an external device. A frame format that is compatible with A series E71. 	△ ^{*11}	○
Fixed buffer communication	With procedure ("Procedure exist")	Sends/receives any data between the CPU module and an external device using the fixed buffer of the Ethernet module.	×	○
	Without procedure ("No procedure")		△ ^{*9}	○
Random access buffer communication		Reads/writes data from/in the random access buffer of the Ethernet module from/to multiple external devices.	×	○
E-mail function		Sends/receives data by e-mail. <ul style="list-style-type: none"> Sending/receiving e-mail by the CPU module Sending/receiving e-mail using the CPU module monitoring function (the automatic notification function) of the Ethernet module 	×	○
Communication using data link instructions		Reads/writes data in the CPU module on another station via Ethernet using data link instructions.	×	○
File transfer (FTP server function)		Reads/writes data in the CPU module in file units from/to an external device using FTP commands.	○ ^{*2}	○
Web function		Communicates CPU module information (a state of the CPU module or a device value) with a personal computer or a programmable controller in remote locations through Internet.	×	○
Communications relayed through CC-Link IE Controller Network, CC-Link IE Field Network, MELSECNET/H, and/or MELSECNET/10		Communicates data over multiple networks in the system where an Ethernet network and other networks co-exist, or where data are communicated over multiple Ethernet networks.	×	○
Router relay function		Communicates data via a router or gateway. (The router relay function is not a function by which the Ethernet module works as a router.)	○ ^{*3}	○
Send frame setting	Ethernet (V2.0)	Sends data using the frame format selected for the Ethernet header of the data link layer.	○	○
	IEEE802.3		×	○

Item		Description	Availability	
			Built-in Ethernet port QCPU	QJ71E71-100
Alive check function (alive check of an external device)	Check with Ping ("Use the Ping")	Checks the connection status of an external device by sending a Ping message (ICMP Echo) to an external device. Closes the corresponding connection if no response message is received.	×	○
	Check with KeepAlive ("Use the KeepAlive")	Checks the connection status of an external device by sending an ACK message, which notifies an open status of the connection opened using the TCP protocol to an external device.	○*4	○
Pairing open		Enables data communications using two connections with opening of one port, by pairing the receiving connection with the sending connection.	×	○
Communication using automatic open UDP port setting		Enables communications without performing open/close processing after the station, in which an Ethernet module is mounted, is started up.	×	○
Remote password check		Prevents unauthorized access to the CPU module by users in remote locations.	○	○
Simultaneous broadcast		Enables simultaneous broadcast to all Ethernet module mounted stations within the same Ethernet network, when fixed buffer communications are performed without procedure on the condition that UDP/IP is used.	△*10	○
Connection to MELSOFT products or GOT		Enables the connection to a MELSOFT product (such as a programming tool and MX Component) or GOT	○	○
Find CPU function		Finds the CPU modules connected to the same hub as GX Developer, and displays a list.	○	×
Time setting function (SNTP client)		Collects time information from the time information server and sets time in the CPU module automatically.	○	×
User connection		Connection which is used by user during communications using the MC protocol or the fixed buffer. Can be used as MELSOFT communication port of system connection by setting parameters. Up to 16 connections can be used.	○	○
System connection	Auto open UDP port	Connection which is used by the system only.	×	○
	FTP transmission port		○	○
	MELSOFT transmission port (UDP/IP)		○*5	○*6
	MELSOFT transmission port (TCP/IP)		○*5	○*6*7
	HTTP port		×	○
	MELSOFT direct connection		○	×
IP packet transfer function		Communicates data (using FTP or HTTP) through the built-in Ethernet ports from an Ethernet device (such as a personal computer) to the following IP-compatible devices connected via a CC-Link IE Controller Network or CC-Link IE Field Network module. • External devices on CC-Link IE Controller Network or CC-Link IE Field Network • External devices on the Ethernet network, which are connected through the built-in Ethernet ports	○	×
Communications using SLMP		Enables an external device to read/write data from/to the SLMP compatible device connected to the shared network with the Ethernet module. In addition, enables an external device to read/write data from/to a device in the CPU module connected to the Ethernet module.	×	○

A

Appendix 4 Specifications Comparison with Ethernet Module

Item	Description	Availability	
		Built-in Ethernet port QCPU	QJ71E71-100
Data communications using the predefined protocol	Enables the Ethernet module to send/receive data to/from an external device by using the protocol for the external device. The external device side protocol can be easily selected, or created/edited from the Predefined Protocol Library of GX Works2.	○ ^{*11}	○

- *1 Available commands are limited. (☞ Page 39, Section 5.2)
- *2 The "quote cpuchg" command cannot be used. (☞ Page 120, Section 9.4)
- *3 Only the default router can be specified.
- *4 Settings are fixed to the following: Interval timer: 5 seconds, Resend timer: 8 times.
- *5 Up to 16 devices can be connected by setting "MELSOFT Connection" for user connections in PLC parameter.
- *6 The MELSOFT transmission port corresponds to the GX Developer transmission port of QJ71E71-100.
- *7 Up to 17 devices can be connected (including one system connection) by setting "MELSOFT Connection" for user connections in the network parameter.
- *8 For processing on the external device side, refer to Page 159, Appendix 4 (2).
- *9 Executable with the socket communication function. Refer to (3) in this section for the differences.
Check the versions of the CPU module and programming tool before using the function. (☞ Page 155, Appendix 3)
- *10 Executable with the socket communication function.
Check the versions of the CPU module and programming tool before using the function. (☞ Page 155, Appendix 3)
- *11 Check the versions of the CPU module and programming tool before using the function. (☞ Page 155, Appendix 3)

Remark

For Ethernet modules, refer to the following.

📖 Q Corresponding Ethernet Interface Module User's Manual (Basic)

.....

(2) Differences on MC protocol functions between the Built-in Ethernet port QCPU and Ethernet module

Item	QJ71E71-100	Built-in Ethernet port QCPU	Consequence of a communication with the Built-in Ethernet port QCPU	Action
Data sending method on TCP when the response message size exceeds 1460 bytes (TCP Maximum Segment Size Option transmission)	Data sending method is selectable. (Default: "Disable TCP Maximum Segment Size Option transmission")	Data sending method is fixed to "Enable TCP Maximum Segment Size Option transmission" and cannot be changed.	If the response message size exceeds 1460 bytes, the message split by the external device may not be read correctly.	Perform the procedure described in Page 45, Section 5.3 (6) so that the external device can process split data.
Wait time for receiving entire message (from the first message to the last message) when a request message is split and sent	1 to 16383.5 seconds. (Default: 30 seconds) (The time value can be changed in "Response monitoring timer" under "Timer setting" of GX Developer.)	Fixed to one second. (A request message is discarded if the next part of the split message cannot be received within one second.)	If each part of the split request message is not sent within one second, a response message is not returned and a communication timeout occurs in the external device.	Retry communication from the external device. If a communication timeout frequently occurs, reduce the load of the external device or Ethernet network.
Operation when request messages are consecutively sent to one connection	Even when one connection consecutively receives request messages, each request message can be processed.	When one connection receives another request message before responding a request message, the second message is discarded.	If request messages are consecutively sent to one connection, response messages are not returned and a communication timeout may occur in the external device.	Check that the external device receives a response message before sending next request message. (Do not consecutively send request messages from the external device.)

A

(3) Differences between the socket communication and the nonprocedural communication using a fixed buffer of the Ethernet module

Item	QJ71E71-100	Built-in Ethernet port QCPU	Consequence of a communication with the Built-in Ethernet port QCPU	Action
Instruction name	ZP.OPEN ZP.CLOSE ZP.BUFRVCV Z.BUFRCVS ZP.BUFSND	SP.SOCOPEN SP.SOCCLOSE SP.SOCRCV S.SOCRCVS SP.SOCSND	-	Replace the instruction name.
Pairing open not necessary	When sending or receiving data using one connection, two connections are occupied by the pairing open setting.	When sending or receiving data using one connection, pairing setting is not used.	-	Set only one connection by parameter. When connection No. of the instruction is the same as the second connection No. of pairing open, replace it with the first connection No.
Automation of UDP and TCP-Full/Unpassive open	Select whether UDP and TCP-Full/Unpassive open is performed by the initial timing setting parameter automatically or by an instruction.	UDP and TCP-Full/Unpassive open is automatically performed.	-	Delete the instructions for open and close for UDP and TCP-Full/Unpassive.
Data sending method on TCP when the message size exceeds 1460 bytes (TCP Maximum Segment Size Option transmission)	Select whether to enable TCP Maximum Segment Size Option transmission in the buffer memory. (Default: "Disable TCP Maximum Segment Size Option transmission")	"Enable TCP Maximum Segment Size Option transmission"	When communications with the QJ71E71-100 are used for a CPU module which performs TCP communications with an external device with the message size over 1460 bytes, the external device may not correctly read split data.	Perform the procedure indicated in Page 78, Section 7.3 (7) so that the external device can process split data.
Connection information acquisition and setting methods	Performed by reading from or writing to the buffer memory.	Performed using socket function instructions.	-	Replace the information acquisition and setting methods with the SP.SOCCINF or SP.SOCCSET instruction.
Activation of an interrupt program in data receiving	An interrupt program can be activated during data receiving.	An interrupt program cannot be activated during data receiving.	-	Program the data receiving processing at the beginning of the scan program.
Host station port number	The following numbers cannot be used as host station port number. 1388 _H to 138A _H (5000 to 5002)	The following numbers cannot be used as host station port number. 1388 _H to 1391 _H (5000 to 5009)	-	Change the port number.
Specification of alive check	Select whether to perform alive check of TCP/IP and UDP/IP using parameters.	Alive check for TCP/IP is performed by default. Alive check for UDP/IP is not available.	-	As described in the left columns.
Ethernet address specification using the OPEN instruction	Ethernet address (MAC address) of the external device can be specified using the ZP.OPEN instruction.	Ethernet address (MAC address) of the external device cannot be specified.	-	Specify "0" for the Ethernet address. (No specification is required. The Ethernet address is automatically acquired for communication.)
Maximum communication data size	2046 bytes	<ul style="list-style-type: none"> Serial number (first 5 digits) is "12051" or earlier: 2046 bytes Serial number (first 5 digits) is "12052" or later: 10238 bytes 	-	-

Memo

A

Appendix 4 Specifications Comparison with Ethernet Module

INDEX

A	
Active open	62
B	
binary	123
Built-in Ethernet port QCPU	19
bye	123
C	
change	127
Class	35
close	123
CPU module	19
CPU-module-dedicated subcommands	127
D	
Data communication frame	44
delete	123
dir	123
Direct connection	33
E	
Ethernet cable	22
Ethernet communication	24
F	
File transfer function	112
FTP	112
FTP commands	120
FTP server support commands	123
Fullpassive	62
G	
get	124
GOT	19,24
GX Developer	19
H	
High-speed Universal model QCPU	19
Host name	28
Hub	22
I	
IP packet transfer function	150
K	
KeepAlive	31
keyword-set	127
L	
Lock processing	135
ls	124
M	
MC protocol	36
mdelete	124
mdir	124
mget	125
mls	125
mput	125
N	
NAK message	38
O	
open	125
P	
Passive open	62
password-lock	128
password-unlock	128
pm-write	130
put	126
pwd	126
Q	
Q series	19
QnUDE(H)CPU	19
QnUDVCPU	19
quit	126
quote	126
R	
recv function	45
rename	126
ret	115
Routers	30
run	129
S	
Searching CPU modules	29
Simple connection	33
Simultaneous broadcast	74
SNTP client	109
Socket communication function instructions	79
status	129
stop	129
Subcommand	121

T

TCP.....	23
TCP fixed-length receive mode.....	104
TCP socket functions.....	45
TCP standard receive mode.....	104
Time setting function.....	109

U

UDP.....	23
Universal model QCPU.....	19
Unlock processing.....	135
Unpassive.....	62
user.....	126

W

Wild card.....	122
----------------	-----



INSTRUCTION INDEX

S

S.SOCRCVS	91
S(P).SOCRDATA.	107
SP.ECPRTCL	57
SP.SOCCINF	98
SP.SOCCLOSE.	84
SP.SOCCSET	101
SP.SOCOPEN.	80
SP.SOCRCV	87
SP.SOCRMODE	103
SP.SOCSND	94

REVISIONS

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

Print date	*Manual number	Revision
December 2008	SH(NA)-080811ENG-A	First edition
March 2009	SH(NA)-080811ENG-B	Revisions involving function addition to the Built-in Ethernet port QCPU (support for the serial number (first five digits) of "11012" and later) <div style="border: 1px solid black; padding: 2px; display: inline-block;">Correction</div> SAFETY PRECAUTIONS, MANUAL PAGE ORGANIZATION, GENERIC TERMS AND ABBREVIATIONS, Section 1.1, CHAPTER 2, 3, Appendix 1 <div style="border: 1px solid black; padding: 2px; display: inline-block;">Addition</div> Section 3.4, CHAPTER 4, Appendix 2, 3 <div style="border: 1px solid black; padding: 2px; display: inline-block;">Change of section No.</div> Section 3.4 → Section 3.5, Section 3.5 → Section 3.6, Section 3.6 → Section 3.7
April 2010	SH(NA)-080811ENG-C	Revisions involving function addition to the Built-in Ethernet port QCPU (support for the serial number (first five digits) of "11082" and later) <div style="border: 1px solid black; padding: 2px; display: inline-block;">Model addition</div> Q50UDEHCPU, Q100UDEHCPU <div style="border: 1px solid black; padding: 2px; display: inline-block;">Correction</div> SAFETY PRECAUTIONS, MANUAL PAGE ORGANIZATION, GENERIC TERMS AND ABBREVIATIONS, Section 1.1, CHAPTER 2, CHAPTER 3, Section 3.1.2, 3.1.4, 3.1.5, 3.2, 3.2.1, 3.2.2, 3.3, 3.3.3, 3.3.4, 3.3.5, 3.4, 3.4.1, 3.4.2, 3.4.3, 3.5, 3.5.1, 3.5.2, 3.6, 3.6.1, 3.6.2, 3.6.3, 3.6.5, 3.6.6, 3.7.2, CHAPTER 4, Section 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, 4.8, 4.9, Appendix 1, Appendix 2, Appendix 3 <div style="border: 1px solid black; padding: 2px; display: inline-block;">Addition</div> Section 3.1.3, 3.8, 3.8.1, 3.8.2, 3.8.3, 3.8.4
August 2010	SH(NA)-080811ENG-D	Revision on the new functions of the Universal model QCPU with a serial number (first 5 digits) of "12052" or later <div style="border: 1px solid black; padding: 2px; display: inline-block;">Correction</div> SAFETY PRECAUTIONS, CHAPTER 2, Section 3.6.2, 3.6.6, 4.3, 4.4, 4.5, 4.8, 4.9, Appendix 1, Appendix 2
July 2011	SH(NA)-080811ENG-E	<div style="border: 1px solid black; padding: 2px; display: inline-block;">Correction</div> SAFETY PRECAUTIONS, CHAPTER 2, Section 3.3.1, 3.3.3, 3.4.1, 3.4.2, 3.4.3, 4.1, 4.6, Appendix 1 <div style="border: 1px solid black; padding: 2px; display: inline-block;">Addition</div> Appendix 4
October 2011	SH(NA)-080811ENG-F	Revision on the new functions of the Universal model QCPU with a serial number (first five digits) of "13102" or later <div style="border: 1px solid black; padding: 2px; display: inline-block;">Correction</div> MANUAL PAGE ORGANIZATION, Section 3.1.1, 3.1.2, 3.1.4, 3.2.1, 3.3.2, 3.3.3, 3.3.4, 3.3.5, 3.4.1, 3.4.2, 3.6.4, 3.7.4, 3.8, CHAPTER 4, Appendix 1, Appendix 2
February 2012	SH(NA)-080811ENG-G	Revision on the new functions of the Universal model QCPU with a serial number (first 5 digits) of "14022" or later <div style="border: 1px solid black; padding: 2px; display: inline-block;">Correction</div> Section 1.1, CHAPTER 3, Section 3.3.5, Appendix 1, Appendix 2, Appendix 4 <div style="border: 1px solid black; padding: 2px; display: inline-block;">Addition</div> Section 3.9
May 2012	SH(NA)-080811ENG-H	<div style="border: 1px solid black; padding: 2px; display: inline-block;">Correction</div> Section 4.2

Print date	*Manual number	Revision
February 2013	SH(NA)-080811ENG-I	Overall revision due to addition of the Universal model QCPU and the changed manual layout <div style="border: 1px solid black; display: inline-block; padding: 2px;">Model addition</div> Q03UDVCPU, Q04UDVCPU, Q06UDVCPU, Q13UDVCPU, Q26UDVCPU
September 2013	SH(NA)-080811ENG-J	Revision on the new functions of the Universal model QCPU with a serial number (first 5 digits) of "15043" or later <div style="border: 1px solid black; display: inline-block; padding: 2px;">Correction</div> Section 5.2.2, 8.1, 8.2, 8.4.1, Appendix 3
January 2014	SH(NA)-080811ENG-K	Revision on the new functions of the Universal model QCPU with a serial number (first five digits) of "15103" or later <div style="border: 1px solid black; display: inline-block; padding: 2px;">Partial correction</div> Section 1.1, Chapter 2, Section 7.3, 7.4.1, 9.2, 9.3, Chapter 10, Section 11.4, Appendix 1, Appendix 3, Appendix 4 <div style="border: 1px solid black; display: inline-block; padding: 2px;">Addition</div> Chapter 6
February 2014	SH(NA)-080811ENG-L	<div style="border: 1px solid black; display: inline-block; padding: 2px;">Partial correction</div> Section 7.4.1

Japanese manual version SH-080806-N

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

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 Vista, Windows NT, Windows XP, Windows Server, Visio, Excel, PowerPoint, Visual Basic, Visual C++, and Access are either registered trademarks or trademarks of Microsoft Corporation in the United States, Japan, and other countries.

Intel, Pentium, and Celeron are either registered trademarks or trademarks of Intel Corporation in the United States and other countries.

Ethernet is a registered trademark of Xerox Corp.

The SD and SDHC logos are either registered trademarks or trademarks of SD-3C, LLC.

All other company names and product names used in this manual are either trademarks or registered trademarks of their respective companies.



QnUCPU User's Manual

Communication via Built-in Ethernet Port

MODEL	QNUDEHCPU-U-ET-E
MODEL CODE	13JZ29
SH(NA)-080811ENG-L(1402)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.