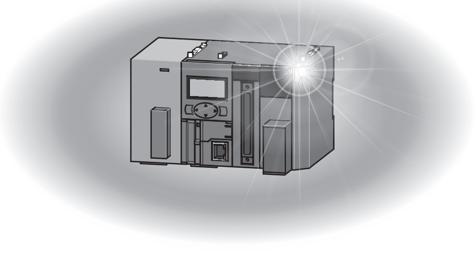
MITSUBISHI ELECTRIC

Programmable Controller

MELSEC *L* *series*

MELSEC-L CPU Module User's Manual (Built-In Ethernet Function)

-L02CPU
-L02CPU-P
-L06CPU
-L06CPU-P
-L26CPU
-L26CPU-P
-L26CPU-BT
-L26CPU-PBT



●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) Emergency stop circuits, protection circuits, and protective interlock circuits for conflicting operations (such as forward/reverse rotations or upper/lower limit positioning) must be configured external to the programmable controller.
  - (2) Machine OPR (Original Point Return) of the positioning function is controlled by two kinds of data: an OPR direction and an OPR speed. Deceleration starts when the near-point dog signal turns on. If an incorrect OPR direction is set, motion control may continue without deceleration. To prevent machine damage caused by this, configure an interlock circuit external to the programmable controller.
  - (3) When the CPU module detects an error during control by the positioning function, the motion slows down and stops.
  - (4) When the programmable controller detects an abnormal condition, it stops the operation and all outputs are:
    - Turned off if the overcurrent or overvoltage protection of the power supply module is activated.
    - Held or turned off according to the parameter setting if the self-diagnostic function of the CPU module detects an error such as a watchdog timer error.
  - (5) All outputs may be turned on if an error occurs in a part, such as an 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 "General Safety Requirements" in the MELSEC-L CPU Module User's Manual (Hardware Design, Maintenance and Inspection).
  - (6) Outputs may remain on or off due to a failure of a component such as a transistor in an output circuit. Configure an external circuit for monitoring output signals that could cause a serious accident.
- ● In an output circuit, 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 each network. Incorrect output or malfunction due to a communication failure may result in an accident.
- ● To prevent the malfunction of the programmable controller system due to harmful e-mails, take preventive measures (such as antivirus measures) so that the mail server for this module does not receive harmful e-mails.
- ● To maintain the safety of the programmable controller system against unauthorized access from external devices via the network, take appropriate measures. To maintain the safety against unauthorized access via the Internet, take measures such as installing a firewall.

[Design Precautions]

⚠WARNING

- When changing data from a peripheral device connected to the CPU module during operation, configure an interlock circuit in the program to ensure that the entire system will always operate safely. For other forms of control (such as program modification or operating status change) of a running programmable controller, read the relevant manuals carefully and ensure that the operation is safe before proceeding. 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 program, and determine corrective actions to be taken between the external device and CPU module in case of a communication failure.
- An absolute position restoration by the positioning function may turn off the servo-on signal (servo off) for approximately 20ms, and the motor may run unexpectedly. If this causes a problem, provide an electromagnetic brake to lock the motor during absolute position restoration.

[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 or more between them. Failure to do so may result in malfunction due to noise.
- During control of an inductive load such as a lamp, heater, or solenoid valve, a large current (approximately ten times greater than normal) may flow when the output is turned from off to on. Therefore, use a module that has 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]

⚠WARNING

- 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 electric shock or cause the module to fail or malfunction.

[Installation Precautions]

⚠CAUTION

- Use the programmable controller in an environment that meets the general specifications in the MELSEC-L CPU Module 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 interconnect modules, engage the respective connectors and securely lock the module joint levers until they click. Incorrect interconnection may cause malfunction, failure, or drop of the module.
- Do not directly touch any conductive parts and electronic components of the module. Doing so can cause malfunction or failure of the module.
- Securely connect an extension cable to the connectors of a branch module and an extension module. After connections, check that the cable 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.
- Do not directly touch any conductive parts and electronic components of the module or SD memory card. Doing so can cause malfunction or failure of the module.

[Wiring Precautions]

⚠WARNING

- Shut off the external power supply (all phases) used in the system before wiring. Failure to do so may result in electric shock or cause the module to fail or malfunction.
- After installation and 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 100 ohms 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 a terminal block screw comes loose, resulting in failure.
- Check the rated voltage and terminal layout before wiring to the module, and connect the cables correctly. Connecting a power supply with a different voltage rating or incorrect wiring may cause a fire or failure.
- Connectors for external devices must be crimped or pressed with the tool specified by the manufacturer, or must be correctly soldered. Incomplete connections may cause short circuit, fire, or malfunction.
- Tighten the terminal block screws 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.
- 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.
- 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.
- To use the high-speed counter function, ground the shield cable on the encoder side (relay box). Always ground the FG and LG terminals to the protective ground conductor. Failure to do so may cause malfunction.
- 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 qualified maintenance personnel with knowledge of protection against electric shock.
  For wiring methods, refer to the MELSEC-L CPU Module 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 block screws. Failure to do so may result in electric shock.

[Startup and Maintenance Precautions]

⚠CAUTION

- Before performing online operations (especially, program modification, forced output, and operating 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 module. 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 away in all directions from the programmable controller. 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 cause the module to fail or malfunction.
- Tighten the terminal block screws within the specified torque range. Undertightening can cause drop of the component or wire, short circuit, or malfunction. Overtightening can damage the screw and/or module, resulting in drop, short circuit, or malfunction.
- After the first use of the product (module, display unit, and terminal block), the number of connections/disconnections is limited to 50 times (in accordance with IEC 61131-2). Exceeding the limit may cause malfunction.
- After the first use of the SD memory card, do not insert/remove the memory card 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 conducting object such as a grounded metal to discharge the static electricity from the human body. Failure to do so may cause the module to fail or malfunction.
- Before testing the operation by the positioning function, set a low speed value for the speed limit parameter so that the operation can be stopped immediately upon occurrence of a hazardous condition.

[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 on battery regulations in EU member states, refer to the MELSEC-L CPU Module User's Manual (Hardware Design, Maintenance and Inspection).)

[Transportation Precautions]

⚠CAUTION

- When transporting lithium batteries, follow the transportation regulations. (For details on the regulated models, refer to the MELSEC-L CPU Module 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

Thank you for purchasing the Mitsubishi MELSEC-L series programmable controllers.
This manual describes the functions of the CPU module and programming necessary for 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 MELSEC-L series programmable controller to handle the product correctly.
When applying the program examples introduced in this manual to an actual system, ensure the applicability and confirm that it will not cause system control problems.

Please make sure that the end users read this manual.

■ Relevant CPU modules

CPU module	Model
LCPU	L02CPU, L02CPU-P, L06CPU, L06CPU-P, L26CPU, L26CPU-P, L26CPU-BT, and L26CPU-PBT

Remark

- This manual describes only the functions of CPU module using Ethernet communication.
  For other CPU module functions, refer to the following.
  📖 MELSEC-L CPU Module User's Manual (Function Explanation, Program Fundamentals)
  📖 MELSEC-L CPU Module User's Manual (Built-In I/O Function)
  📖 QnUDVCPU/LCPU User's Manual (Data Logging Function)
- The L02SCPU and L02SCPU-P do not support the built-in Ethernet function.

RELEVANT MANUALS

(1) CPU module user's manual

Manual name <manual number (model code)>	Description
MELSEC-L CPU Module User's Manual (Hardware Design, Maintenance and Inspection) <SH-080890ENG, 13JZ36>	Specifications of the CPU modules, power supply modules, display unit, branch module, extension module, SD memory cards, and batteries, information on how to establish a system, maintenance and inspection, and troubleshooting
MELSEC-L CPU Module User's Manual (Function Explanation, Program Fundamentals) <SH-080889ENG, 13JZ35>	Functions and devices of the CPU module, and programming
MELSEC-L CPU Module User's Manual (Built-In I/O Function) <SH-080892ENG, 13JZ38>	The general-purpose I/O function, interrupt input function, pulse catch function, positioning function, and high-speed counter function of the CPU module
QnUDVCPU/LCPU User's Manual (Data Logging Function) <SH-080893ENG, 13JZ39>	The data logging function of the CPU module

(2) Programming manual

Manual name <manual number (model code)>	Description
MELSEC-Q/L Programming Manual (Common Instruction) <SH-080809ENG, 13JW10>	Detailed description and usage of instructions used in programs

(3) Operating manual

Manual name <manual number (model code)>	Description
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

(4) I/O module and intelligent function module manual

Manual name <manual number (model code)>	Description
MELSEC Communication Protocol Reference Manual <SH-080008, 13JF89>	Details of MELSEC communication protocol (MC protocol) that is used for data communication between a target device and a CPU module

CONTENTS

SAFETY PRECAUTIONS 1
CONDITIONS OF USE FOR THE PRODUCT 7
INTRODUCTION 8
RELEVANT MANUALS 9
MANUAL PAGE ORGANIZATION 14
TERMS 17

CHAPTER 1 OVERVIEW 18

CHAPTER 2 COMMUNICATION SPECIFICATIONS 20

CHAPTER 3 CONNECTION WITH PROGRAMMING TOOL AND GOT 22

3.1 Setting Required for the CPU Module 23
3.2 Setting Required for the Programming Tool 25
3.3 Searching CPU Modules on the Network 27
3.4 Communication via Routers 28
3.5 Precautions 29

CHAPTER 4 DIRECT CONNECTION TO PROGRAMMING TOOL (SIMPLE CONNECTION) 31

4.1 Setting Method 32
4.2 Precautions 33

CHAPTER 5 MC PROTOCOL COMMUNICATION 34

5.1 Sending a Command from an External Device to the CPU Module 34
5.1.1 Setting Method 36
5.1.2 MC Protocol Commands 38
5.1.3 Precautions 40
5.1.4 Error Codes for MC Protocol Communication 42
5.2 Sending a Command from the CPU Module to an External Device 43
5.2.1 Sending an SLMP frame (SP.SLMPSND) 44

CHAPTER 6 DATA COMMUNICATIONS USING THE PREDEFINED PROTOCOL 53

6.1 Specifications 54
6.2 Setting Method 55
6.3 Setting Items of Predefined Protocol Support Function 60
6.3.1 Communication type 60
6.3.2 Packet elements set for "Packet Setting" 60
6.4 Predefined Protocol Function Instructions 61
6.4.1 Executing the predefined protocol (SP.ECPRTCL) 61
6.5 Execution Conditions of Predefined Protocol Function 62
6.6 Operation Image and Data Structure of the Predefined Protocol Function 62
6.7 Precautions 63

CHAPTER 7 SOCKET COMMUNICATION FUNCTION 64

7.1 Communication Using TCP 66

7.2 Communication Using UDP 77

7.3 Precautions for the Socket Communication Function. 83

7.4 Socket Communication Function Instructions. 85

7.4.1 Establishing a connection (SP.SOCOPEN) 86

7.4.2 Disconnecting a connection (SP.SOCCLOSE) 92

7.4.3 Reading out received data in the END processing (SP.SOCRCV) 95

7.4.4 Reading out received data during instruction execution (S.SOCRCVS) 99

7.4.5 Sending data (SP.SOCSND) 102

7.4.6 Reading out connection information (SP.SOCCINF) 106

7.4.7 Changing the target of a connection (UDP/IP) (SP.SOCCSET). 109

7.4.8 Changing the receive mode of a connection (SP.SOCRMODE) 111

7.4.9 Socket communication receive data read (S(P).SOCRDATA) 114

CHAPTER 8 TIME SETTING FUNCTION (SNTP CLIENT) 116

8.1 Setting Method 117

8.2 Precautions. 118

CHAPTER 9 FILE TRANSFER FUNCTION (FTP SERVER) 119

9.1 Setting for FTP Communication 120

9.2 Files That Can Be Transferred Using FTP 123

9.3 Files That Can Be Deleted Using FTP 125

9.4 FTP Commands 127

9.4.1 List of FTP commands 127

9.4.2 How to specify an FTP command 129

9.4.3 Details of FTP commands 130

9.5 Precautions. 137

CHAPTER 10 FILE TRANSFER FUNCTION (FTP CLIENT) 139

10.1 Transferrable Files 140

10.2 Setting Method 141

10.2.1 Parameter setting. 141

10.2.2 Settings at programming 143

10.3 File Transfer Function Instruction 146

10.3.1 FTP client file transmission (SP.FTPPUT) 146

10.3.2 FTP client file acquisition (SP.FTPGET) 152

10.4 Diagnostics 157

10.5 Precautions. 158

CHAPTER 11 REMOTE PASSWORD 159

11.1 Communication Using Remote Password 160

11.2 Remote Password Setting 161

11.3 Precautions. 163

11.4 Detection of Unauthorized Access and Actions 164

CHAPTER 12 SIMPLE PLC COMMUNICATION FUNCTION 165

12.1 Setting Method 166

12.2 Program to Check Communications. 177

12.3 Diagnostics 178

12.4 Errors Related to the Simple PLC Communication Function 178

12.5 Precautions. 179

CHAPTER 13 IP ADDRESS CHANGE FUNCTION 181

13.1 IP Address of the Built-in Ethernet Port 182

13.2 How to Use the Function 183

13.2.1 Write operation. 183

13.2.2 Clear operation. 186

13.3 Checking the IP Address 189

13.4 Precautions. 190

CHAPTER 14 IP PACKET TRANSFER FUNCTION 191

CHAPTER 15 E-MAIL SEND/RECEIVE FUNCTION 193

15.1 E-mail Specifications. 194

15.2 Setting Method 195

15.2.1 Parameter setting. 195

15.2.2 Settings at programming 198

15.3 Function Details 199

15.3.1 Function to read/write device when e-mail is received. 199

15.3.2 Send destination e-mail address edit function 203

15.4 E-mail Send/Receive Function Instructions 204

15.4.1 Send e-mail (SP.MLSEND) 204

15.4.2 Send e-mail (SP.MLRECV) 208

15.4.3 E-mail address registration/deletion (SP.MLOPEADR) 213

15.4.4 E-mail address acquisition (SP.MLGETADR). 216

15.5 Diagnostics 218

APPENDICES 219

Appendix 1 Operation Processing Time for Each Instruction. 219

Appendix 2 Port Numbers Used by Built-in Ethernet Port LCPU 221

Appendix 3 Added and Changed Functions. 222

Appendix 4 Performance List of Simple PLC Communication Function 223

Appendix 5 Specifications Comparison with Ethernet Module 225

INDEX 231

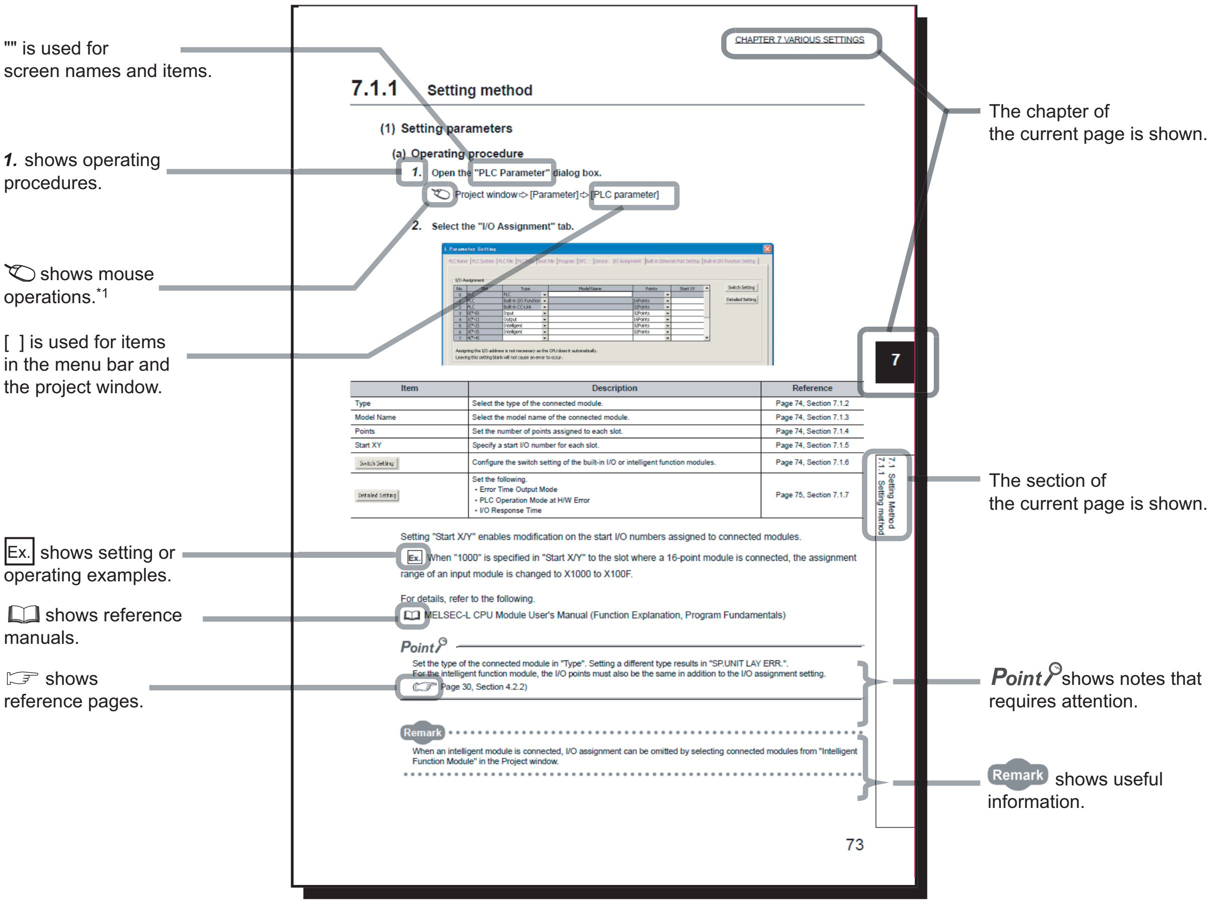
INSTRUCTION INDEX 233

REVISIONS 235
WARRANTY 237
TRADEMARKS 238

MANUAL PAGE ORGANIZATION

In this manual, pages are organized and the symbols are used as shown below.

The following illustration is for explanation purpose only, and should not be referred to as an actual documentation.



"" 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.

CHAPTER 7 VARIOUS SETTINGS

7.1.1 Setting method

(1) Setting parameters

(a) Operating procedure

1. Open the "PLC Parameter" dialog box.

🖱 Project window ⇨ [Parameter] ⇨ [PLC parameter]

2. Select the "I/O Assignment" tab.

Item	Description	Reference
Type	Select the type of the connected module.	Page 74, Section 7.1.2
Model Name	Select the model name of the connected module.	Page 74, Section 7.1.3
Points	Set the number of points assigned to each slot.	Page 74, Section 7.1.4
Start XY	Specify a start I/O number for each slot.	Page 74, Section 7.1.5
Switch Setting	Configure the switch setting of the built-in I/O or intelligent function modules.	Page 74, Section 7.1.6
Detailed Setting	Set the following. • Error Time Output Mode • PLC Operation Mode at H/W Error • I/O Response Time	Page 75, Section 7.1.7

Setting "Start X/Y" enables modification on the start I/O numbers assigned to connected modules.

Ex. When "1000" is specified in "Start X/Y" to the slot where a 16-point module is connected, the assignment range of an input module is changed to X1000 to X100F.

For details, refer to the following.

📖 MELSEC-L CPU Module User's Manual (Function Explanation, Program Fundamentals)

Point

Set the type of the connected module in "Type". Setting a different type results in "SP.UNIT LAY ERR.".
For the intelligent function module, the I/O points must also be the same in addition to the I/O assignment setting.
(☞ Page 30, Section 4.2.2)

Remark

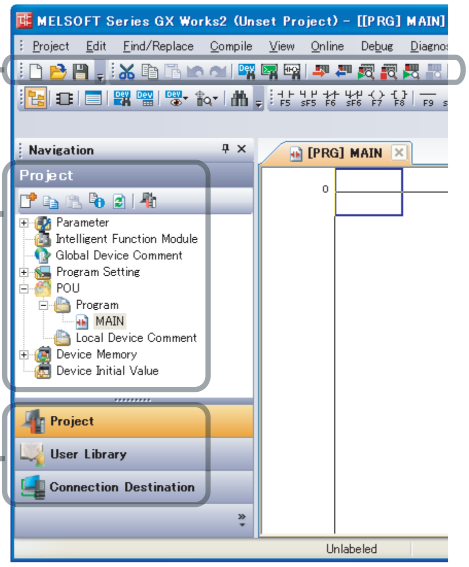
When an intelligent module is connected, I/O assignment can be omitted by selecting connected modules from "Intelligent Function Module" in the Project window.

7.1 Setting Method 7.1.1 Setting method

7

73

[*1] The mouse operation example (for GX Works2) is provided below.



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]

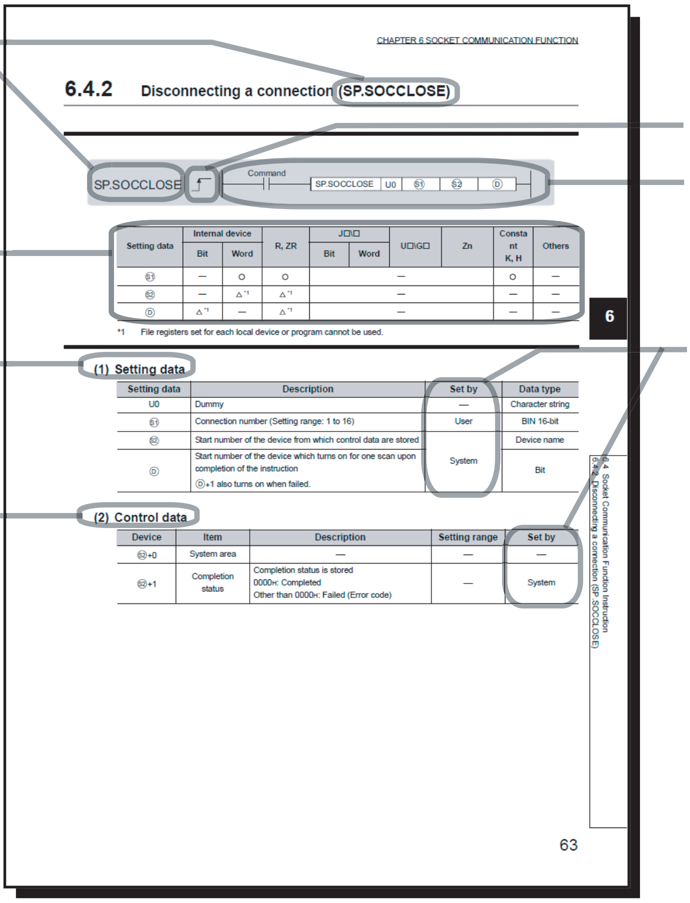
Select [Project] from the view selection area to open the Project window.

In the Project window, expand [Parameter] and select [PLC Parameter].

View selection area

Pages describing instructions are organized as shown below.

The following illustration is for explanation purpose only, and should not be referred to as an actual documentation.



Instruction name

○ shows the devices applicable to the instruction

Descriptions of setting data and data type

Descriptions of control data (if any)

Execution condition of the instruction

Structure of the instruction in the ladder mode

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

CHAPTER 6 SOCKET COMMUNICATION FUNCTION

6.4.2 Disconnecting a connection (SP.SOCCLOSE)

SP.SOCCLOSE ↑

```
  Command
---| |-------[SP.SOCCLOSE | U0 | Ⓢ1 | Ⓢ2 | Ⓓ ]-
```

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.

6

(1) Setting data

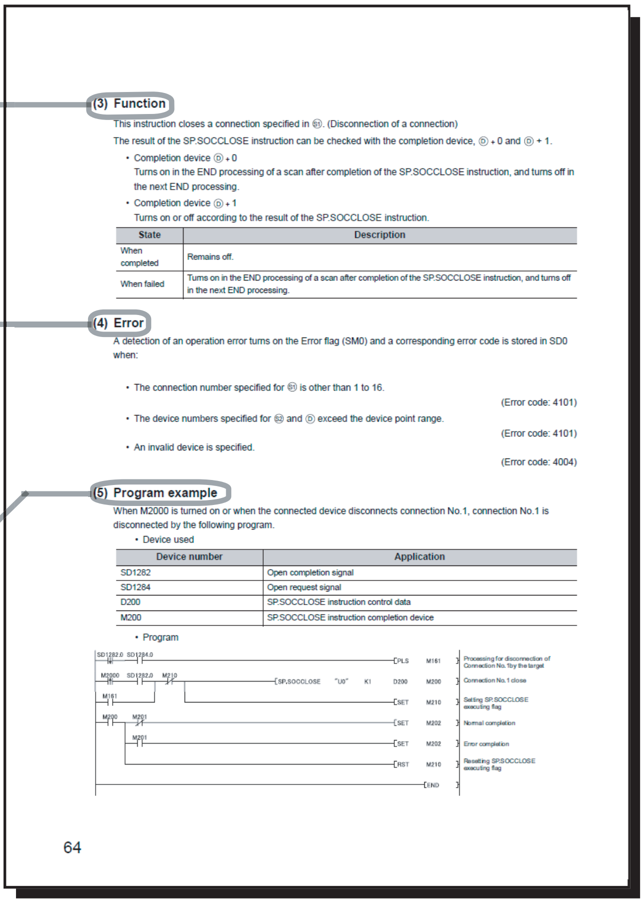
Setting data	Description	Set by	Data type
U0	Dummy	—	Character string
Ⓢ1	Connection number (Setting range: 1 to 16)	User	BIN 16-bit
Ⓢ2	Start number of the device from which control data are stored	System	Device name
Ⓓ	Start number of the device which turns on for one scan upon completion of the instruction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

(2) Control data

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

6.4 Socket Communication Function Instruction
6.4.2 Disconnecting a connection (SP.SOCCLOSE)

63



Detailed descriptions of the instruction

Conditions for the error and error codes
For the errors not described in this manual, refer to the following.
MELSEC-Q/L Programming Manual (Common Instruction)

Simple program example(s) and descriptions of the devices used

(3) Function

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

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

- Completion device Ⓓ + 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 Ⓓ + 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 connection number specified for Ⓢ1 is other than 1 to 16.
  (Error code: 4101)
- The device numbers specified for Ⓢ2 and Ⓓ 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

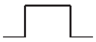
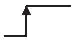
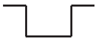
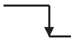
Device number	Application
SD1282	Open completion signal
SD1284	Open request signal
D200	SP.SOCCLOSE instruction control data
M200	SP.SOCCLOSE instruction completion device

- Program

```
SD1282.0 SD1284.0
---|↓|------| |------------------------[PLS  M161                    ]  Processing for disconnection of
                                                                         Connection No. 1by the target
M2000    SD1282.0  M210
---|↑|------| |-----|/|----[SP.SOCCLOSE "U0" K1 D200 M200           ]  Connection No. 1 close
M161                    |
---| |-----------       |-------------[SET  M210                     ]  Setting SP.SOCCLOSE
                                                                         executing flag
M200     M201
---| |-----|/|-------------------------[SET  M202                    ]  Normal completion
         M201
        |--| |--------------------------[SET  M202                   ]  Error completion
        |
        |-------------------------------[RST  M210                   ]  Resetting SP.SOCCLOSE
                                                                         executing flag
----------------------------------------------------------[END       ]
```

64

- Instructions can be executed under the following conditions.

Execution condition	Any time	During on	On the rising 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□\□		Intelligent function module device U□\G□	Index register Zn	Constant [*3]	Others [*3]
	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, @□	R, ZR	—		U□\G□	Z	K, H, E, $	P, I, J, U, D, X, DY, N, BL, TR, BL\S, V

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

MELSEC-L CPU Module 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] 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 terms.

Term	Description
Branch module	The abbreviation for the MELSEC-L series branch module
Built-in Ethernet port LCPU	A generic term for the L02CPU, L02CPU-P, L06CPU, L06CPU-P, L26CPU, L26CPU-P, L26CPU-BT, and L26CPU-PBT
CPU module	The abbreviation for the MELSEC-L series CPU module
Display unit	A liquid crystal display to be attached to the CPU module
END cover	A cover to be attached to the right side of the rightmost MELSEC-L series module
Extension cable	The abbreviation for the MELSEC-L series extension cable
Extension module	The abbreviation for the MELSEC-L series extension module
External device	A GOT, measuring instrument, ID module, bar code reader, adjuster, and other CPU modules connected with a CPU module for the data communication
FTP	The abbreviation for File Transfer Protocol, which is a standard network protocol used to exchange files
GX Works2	The product name of the software package for the MELSEC programmable controllers
GX Developer	
LCPU	Another term for the MELSEC-L series CPU module
MC protocol	The abbreviation for the MELSEC communication protocol, a protocol to access a CPU module from a target device in the Ethernet or serial communication
Power supply module	The abbreviation for the MELSEC-L series power supply module
Programming tool	A generic term for GX Works2 and GX Developer
SD memory card	Secure Digital Memory Card, which is a flash memory device. The NZ1MEM-2GBSD, NZ1MEM-4GBSD, NZ1MEM-8GBSD, NZ1MEM-16GBSD, L1MEM-2GBSD, and L1MEM-4GBSD are available.
SNTP	The abbreviation for Simple Network Time Protocol, which is a protocol for synchronizing the clocks of computer systems over a TCP/IP based network

CHAPTER 1 OVERVIEW

The following describes the built-in Ethernet function of the CPU module.

(1) Connection with programming tool and GOT (☞ Page 22, CHAPTER 3)

- By using a hub, the CPU module can be connected to multiple programming tools and GOTs. Up to 16 external devices can be connected to a single CPU module at the same time.
- When CPU modules and a programming tool are connected to the same hub, these CPU modules can be searched from the programming tool. Displayed search results include IP addresses so that any of them can be specified.
- MELSOFT connection allows access through routers in an environment such as a corporate LAN.

(2) Direct connection to programming tool (simple connection) (☞ Page 31, 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).
For direct connection, the IP address and host name need not be specified in the transfer setup.

(3) MC protocol communication (☞ Page 34, CHAPTER 5)

From an external device such as a personal computer or HMI, device data of the CPU module can be read or written using MC protocol commands.
In addition, MC protocol messages (QnA-compatible 3E frame) can be sent from the CPU module to external devices connected on the Ethernet network.

(4) Communications using the predefined protocol (☞ Page 53, 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 64, 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 116, 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 server) (☞ Page 119, 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) File transfer function (FTP client) (☞ Page 139, CHAPTER 10)

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

(9) Remote password (☞ Page 159, CHAPTER 11)

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

(10)Simple PLC communication function (☞ Page 165, CHAPTER 12)

Device data can be communicated between the CPU modules connected with Ethernet cable without programming.

(11)IP packet transfer function (☞ Page 191, CHAPTER 14)

Communications can be performed with a device which supports the following IP addresses, which have been specified via a 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 Field Network
- External devices on the Ethernet network, which are connected through the built-in Ethernet ports

(12)E-mail send/receive function (☞ Page 193, CHAPTER 15)

E-mail can be sent and received between the CPU module and electronic devices such as mobile phones and personal computers.

In addition, writing an MC protocol command in the message body of incoming mail enables device reading and writing.

CHAPTER 2 COMMUNICATION SPECIFICATIONS

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

Item			Specifications
Transmission specifications	Data transfer speed		100/10Mbps
	Communication mode		Full-duplex or half-duplex
	Transmission method		Base band
	Maximum distance between hub and node		100m
	Maximum number of nodes/connection	10BASE-T	Cascade connection: Up to four*2
		100BASE-TX	Cascade connection: Up to two*2
Number of connections	TCP/IP		A total of 16 connections of socket communications, MELSOFT connection, MC protocol, predefined protocol, FTP client, and e-mail*4 1 connection for FTP server
	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 the 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.

*4 One setting of FTP client or e-mail uses two consecutive connections.

Hubs with 10BASE-T or 100BASE-TX ports*5 can be used.
Up to 16 external devices can access one CPU module at the same time.

*5 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

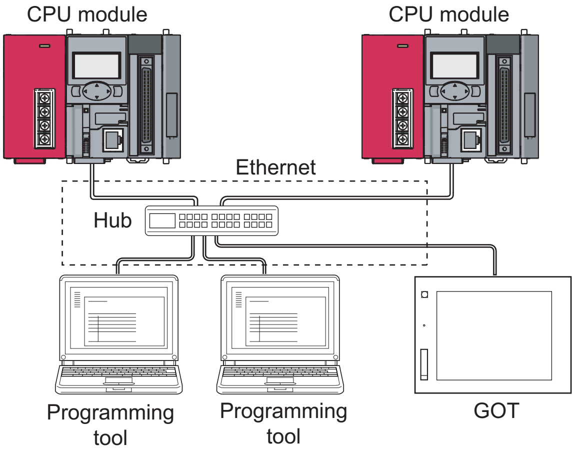
TCP and UDP are defined as follows:

- 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.

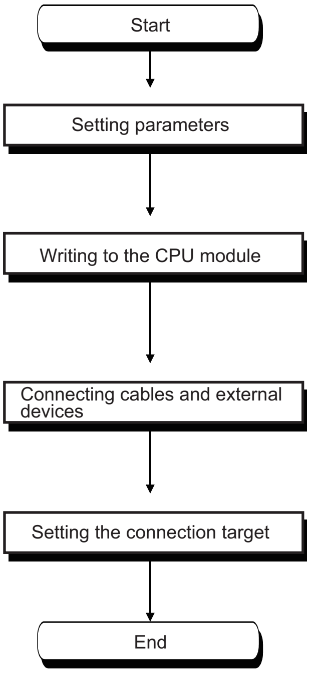
Select an appropriate protocol, considering the specifications of the external device and the characteristics of the above protocols.

CHAPTER 3 CONNECTION WITH PROGRAMMING TOOL AND GOT

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



To start Ethernet communication, perform the following steps.


Start
Setting parameters
Writing to the CPU module
Connecting cables and external devices
Setting the connection target
End

Set PLC parameters using the programming tool.
(☞ Page 23, Section 3.1 (1))

Write the set parameters to the CPU module. Power off and on or reset the CPU module to enable the parameters.
(☞ Page 24, Section 3.1 (2))

Connect cables and devices required for Ethernet communication.

Set a connection target by the programming tool.
(☞ Page 25, Section 3.2)

For the GOT setting, refer to the following.

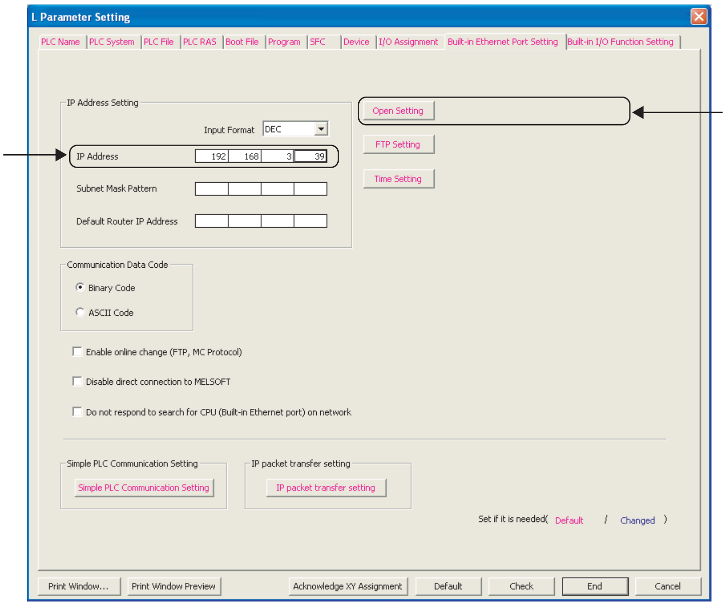
📖 Manual for the GOT used

3.1 Setting Required 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]



Parameter Setting

PLC Name | PLC System | PLC File | PLC RAS | Boot File | Program | SFC | Device | I/O Assignment | Built-in Ethernet Port Setting | Built-in I/O Function Setting

IP Address Setting
Input Format: DEC
IP Address: 192 . 168 . 3 . 39
Subnet Mask Pattern
Default Router IP Address

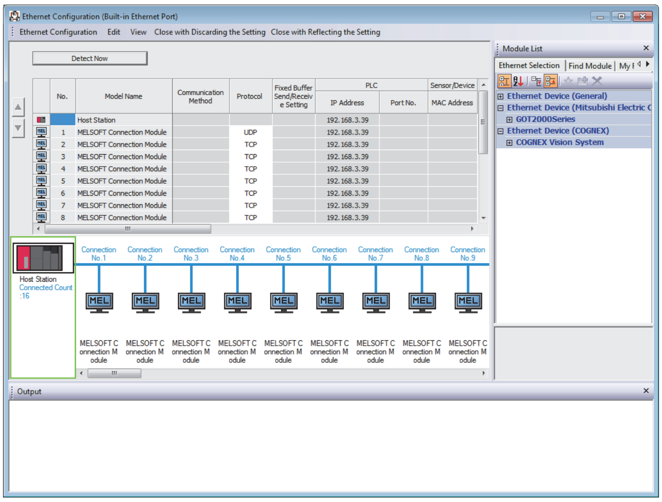
Communication Data Code
● Binary Code
○ ASCII Code

□ Enable online change (FTP, MC Protocol)
□ Disable direct connection to MELSOFT
□ Do not respond to search for CPU (Built-in Ethernet port) on network

1.
2.
Open Setting
FTP Setting
Time Setting
Simple PLC Communication Setting
Simple PLC Communication Setting
IP packet transfer setting
IP packet transfer setting
Set if it is needed(Default / Changed)
Print Window...
Print Window Preview
Acknowledge XY Assignment
Default
Check
End
Cancel

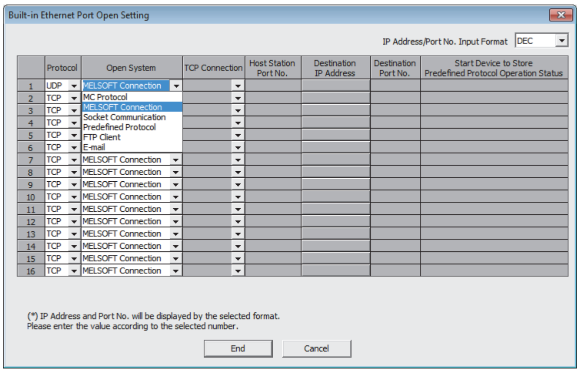
1. **Set the IP address of the CPU module.**
2. **Set MELSOFT connection.**

Project window ➪ [Parameter] ➪ [PLC Parameter] ➪ [Built-in Ethernet Port Setting] ➪ [Ethernet Conf.] or [Open Setting] button



Ethernet Configuration (Built-in Ethernet Port)

No.	Model Name	Communication Method	Protocol	Fixed Buffer Send/Receive Setting	PLC IP Address	PLC Port No.	Sensor/Device MAC Address
	Host Station				192.168.3.39		
1	MELSOFT Connection Module		UDP		192.168.3.39		
2	MELSOFT Connection Module		TCP		192.168.3.39		
3	MELSOFT Connection Module		TCP		192.168.3.39		
4	MELSOFT Connection Module		TCP		192.168.3.39		
5	MELSOFT Connection Module		TCP		192.168.3.39		
6	MELSOFT Connection Module		TCP		192.168.3.39		
7	MELSOFT Connection Module		TCP		192.168.3.39		
8	MELSOFT Connection Module		TCP		192.168.3.39		



Built-in Ethernet Port Open Setting

IP Address/Port No. Input Format: DEC

	Protocol	Open System	TCP Connection	Host Station Port No.	Destination IP Address	Destination Port No.	Start Device to Store Predefined Protocol Operation Status
1	UDP	MELSOFT Connection					
2	TCP	MC Protocol					
3	TCP	MELSOFT Connection					
4	TCP	Socket Communication					
5	TCP	Predefined Protocol					
6	TCP	FTP Client E-mail					
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.

- For the "Ethernet Conf.", drag and drop the "MELSOFT Connection Module" from "Module List" to the left side on the window. Select a protocol from "Protocol" depending on the target device.
- For the "Open Setting"

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

(2) Writing parameters to the CPU module

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

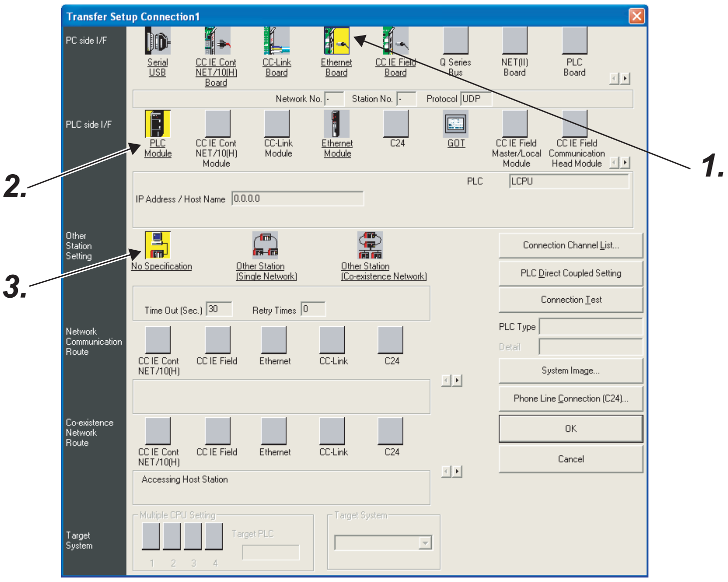
[Online] ⇨ [Write to PLC]

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

3.2 Setting Required for the Programming Tool

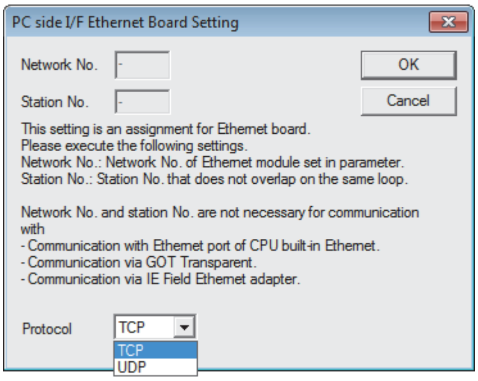
Configure the settings in the "Transfer Setup" window.

Connection Destination window ➪ [Connection1]



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

In the "PC side IF Ethernet Board Setting" window, select a "TCP" or "UDP" protocol. Select the same protocol as the one set in the "Ethernet Conf." or "Open Setting" window. (☞ Page 23, Section 3.1)



Network No. -

Station No. -

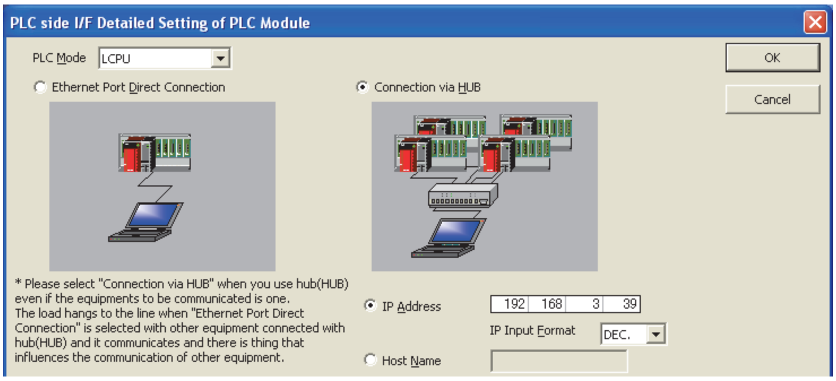
This setting is an assignment for Ethernet board.
Please execute the following settings.
Network No.: Network No. of Ethernet module set in parameter.
Station No.: Station No. that does not overlap on the same loop.

Network No. and station No. are not necessary for communication with
- Communication with Ethernet port of CPU built-in Ethernet.
- Communication via GOT Transparent.
- Communication via IE Field Ethernet adapter.

Protocol TCP
TCP
UDP

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.)



PLC side I/F Detailed Setting of PLC Module

PLC Mode: LCPU

○ Ethernet Port Direct Connection

● Connection via HUB

* Please select "Connection via HUB" when you use hub(HUB) even if the equipments to be communicated is one.
The load hangs to the line when "Ethernet Port Direct Connection" is selected with other equipment connected with hub(HUB) and it communicates and there is thing that influences the communication of other equipment.

● IP Address: 192 . 168 . 3 . 39

IP Input Format: DEC.

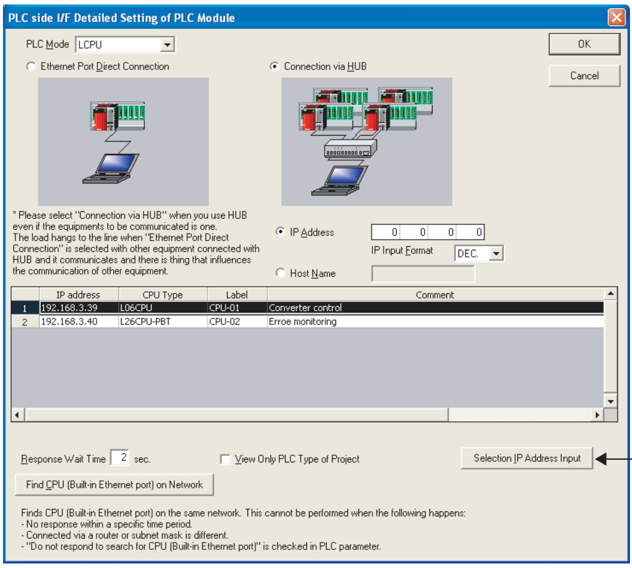
○ Host Name:

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.



PLC side I/F Detailed Setting of PLC Module

PLC Mode: LCPU

○ Ethernet Port Direct Connection

● Connection via HUB

* Please select "Connection via HUB" when you use HUB even if the equipments to be communicated is one.
The load hangs to the line when "Ethernet Port Direct Connection" is selected with other equipment connected with HUB and it communicates and there is thing that influences the communication of other equipment.

● IP Address: 0 0 0 0

IP Input Format: DEC.

○ Host Name

	IP address	CPU Type	Label	Comment
1	192.168.3.39	L06CPU	CPU-01	Converter control
2	192.168.3.40	L26CPU-PBT	CPU-02	Erroe monitoring

Response Wait Time: 2 sec.

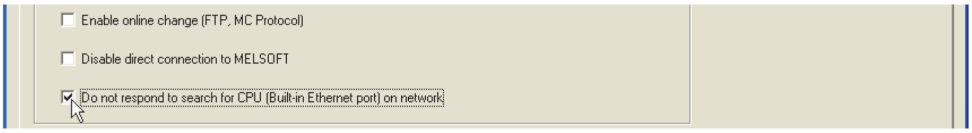
□ View Only PLC Type of Project

Finds CPU (Built-in Ethernet port) on the same network. This cannot be performed when the following happens:
- No response within a specific time period.
- Connected via a router or subnet mask is different.
- "Do not respond to search for CPU (Built-in Ethernet port)" is checked in PLC parameter.

OK
Cancel
Selection IP Address Input
Find CPU (Built-in Ethernet port) on Network

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 item shown below in the "Built-in Ethernet Port Setting" tab of PLC parameter, the Find CPU function can be disabled and the system does not respond to a search request on the network.



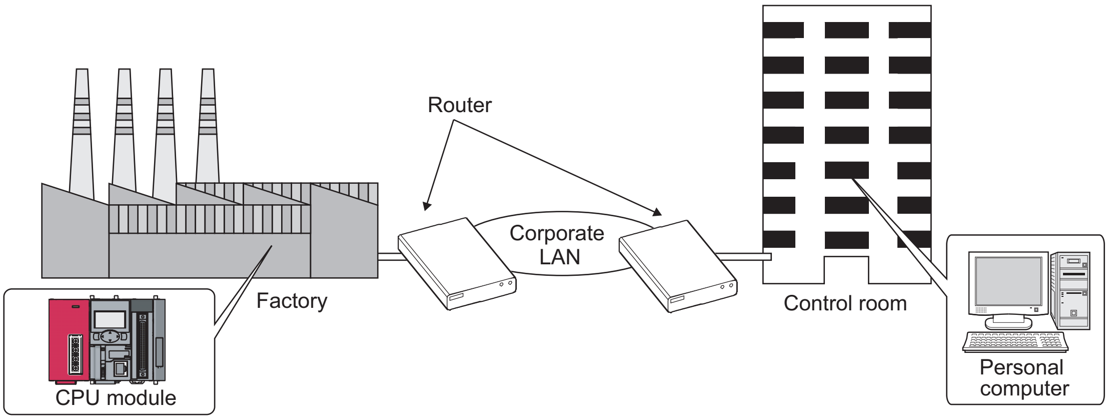
□ Enable online change (FTP, MC Protocol)

□ Disable direct connection to MELSOFT

☑ Do not respond to search for CPU (Built-in Ethernet port) on network

3.4 Communication via Routers

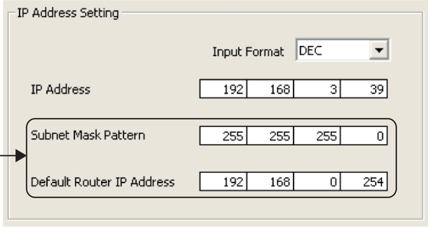
From the built-in Ethernet port, access is available via routers on a network such as a corporate LAN.[*1]



*1 The following functions do not support the communication via routers.
· 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 23, Section 3.1) 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]



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

Set the subnet mask pattern and default router IP address.

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. (A response to a KeepAlive ACK message is checked.) 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 "Ethernet Conf." or "Open Setting" of parameters. Establishing too many TCP connections from a personal computer may cause the following states, depending on the application.

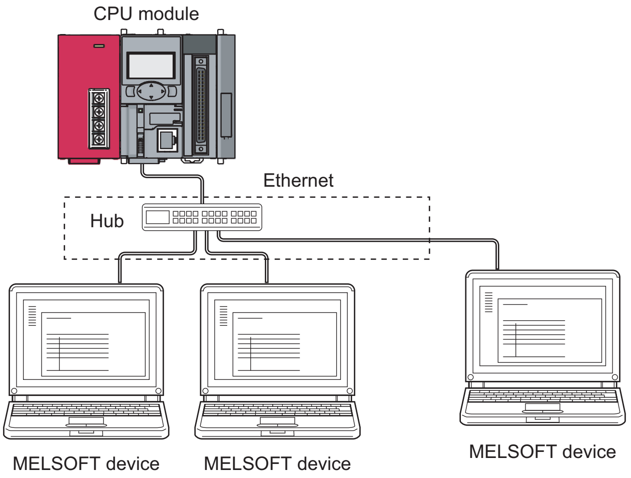
- Time before timeout error detection is increased.
- An unexpected timeout error occurs in any of the communicating devices.

(4) Retransmission on TCP connection

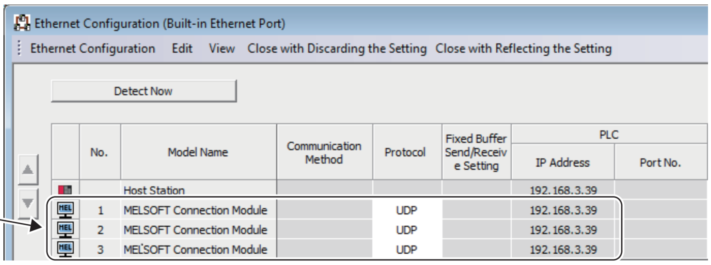
If no ACK response is returned from the other end of a TCP connection, the ACK will be resent six times, starting in 0.3 seconds after the first transmission, and then 0.6, 1.2, 2.4, 4.8, and 9.6 seconds. When no TCP ACK response is returned within 19.2 seconds after the last retransmission, the device 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 connections as that of the connected MELSOFT devices in PLC parameter.



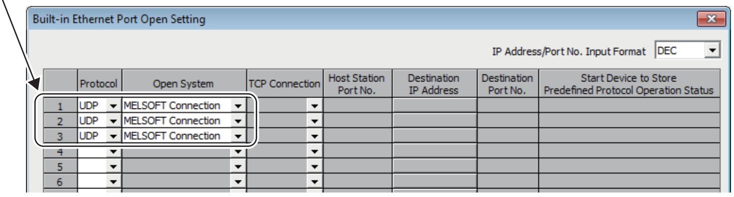
Set the same number of protocols as that of MELSOFT devices.



Ethernet Configuration (Built-in Ethernet Port)

Ethernet Configuration
Edit
View
Close with Discarding the Setting
Close with Reflecting the Setting
Detect Now

	No.	Model Name	Communication Method	Protocol	Fixed Buffer Send/Receive Setting	PLC	
						IP Address	Port No.
		Host Station				192.168.3.39	
	1	MELSOFT Connection Module		UDP		192.168.3.39	
	2	MELSOFT Connection Module		UDP		192.168.3.39	
	3	MELSOFT Connection Module		UDP		192.168.3.39	



Built-in Ethernet Port Open Setting

IP Address/Port No. Input Format: DEC

	Protocol	Open System	TCP Connection	Host Station Port No.	Destination IP Address	Destination Port No.	Start Device to Store Predefined Protocol Operation Status
1	UDP	MELSOFT Connection					
2	UDP	MELSOFT Connection					
3	UDP	MELSOFT Connection					
4							
5							
6							

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

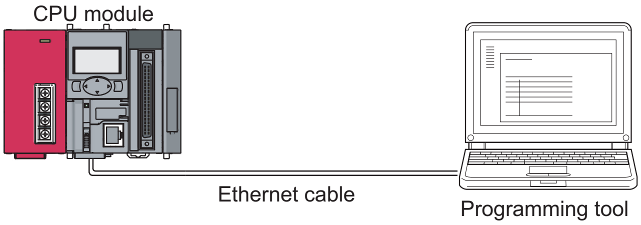
When remote STOP or remote PAUSE has been executed using the programming tool via a built-in Ethernet port, perform the following operations before powering off or resetting the CPU module.

- 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).

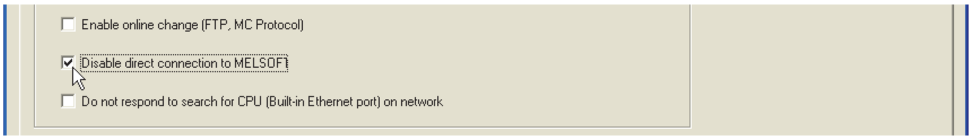
For direct connection, the IP address and host name need not be specified in the connection target setting. (Simultaneous broadcast is used.)



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 PLC parameter.



□ Enable online change (FTP, MC Protocol)

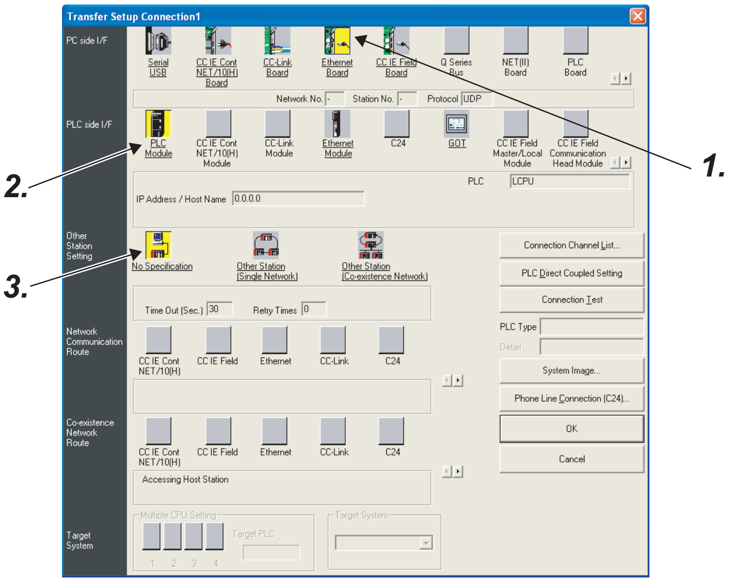
☑ Disable direct connection to MELSOFT

□ Do not respond to search for CPU (Built-in Ethernet port) on network

4.1 Setting Method

Set the items on the "Transfer Setup" window.

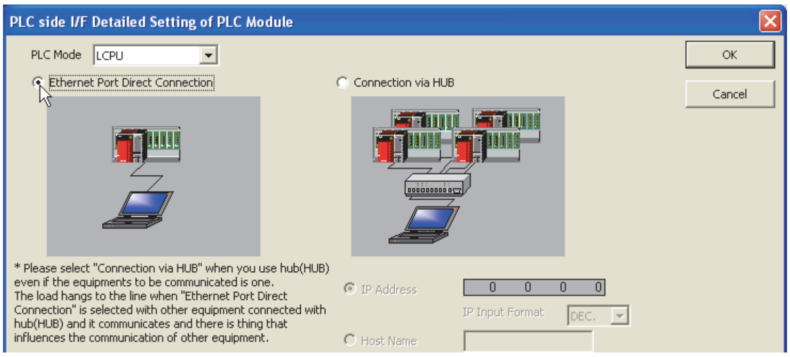
Connection Destination window ➪ [Connection1]



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.



* Please select "Connection via HUB" when you use hub(HUB) even if the equipments to be communicated is one. The load hangs to the line when "Ethernet Port Direct Connection" is selected with other equipment connected with hub(HUB) and it communicates and there is thing that influences the communication of other equipment.

3. Complete setting of "Other Station Setting".

Select an item appropriate to the operating environment.

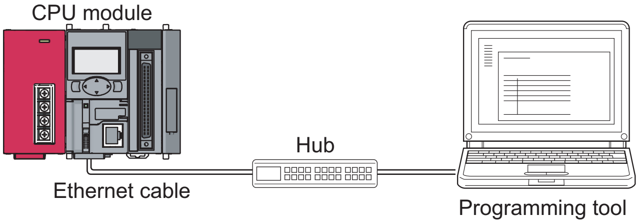
4.2 Precautions

(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.

Example:	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.

Example:	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

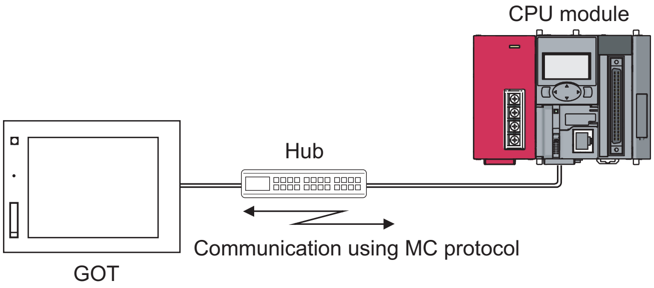
● The host address for each class is the part shown with "0".
Class A: 255. 0. 0. 0, Class B: 255.255. 0. 0, Class C: 255.255.255. 0

CHAPTER 5 MC PROTOCOL COMMUNICATION

The built-in Ethernet port allows MC protocol communication.

5.1 Sending a Command from an External Device to the CPU Module

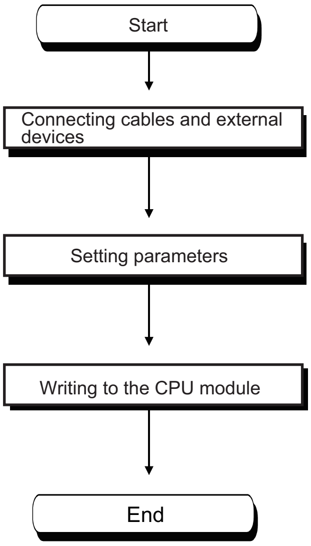
From an external device such as a personal computer or HMI, device data of the CPU module can be read or written using MC protocol. Monitoring of CPU module operation, data analysis, and production control are available on a personal computer or HMI by these device data reading and writing. Besides, the remote password function can prevent unauthorized access from outside of the system. (Page 159, CHAPTER 11)



Point

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

To start MC protocol communication, perform the following steps.


Start
Connecting cables and external devices
Setting parameters
Writing to the CPU module
End

Connect cables and devices required for MC protocol communication.

Set PLC parameters using the programming tool. (Page 36, Section 5.1.1)

Write the set parameters to the CPU module. Power off and on or reset the CPU module to enable the parameters.

MC protocol communication is available.

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

MELSEC Communication Protocol Reference Manual

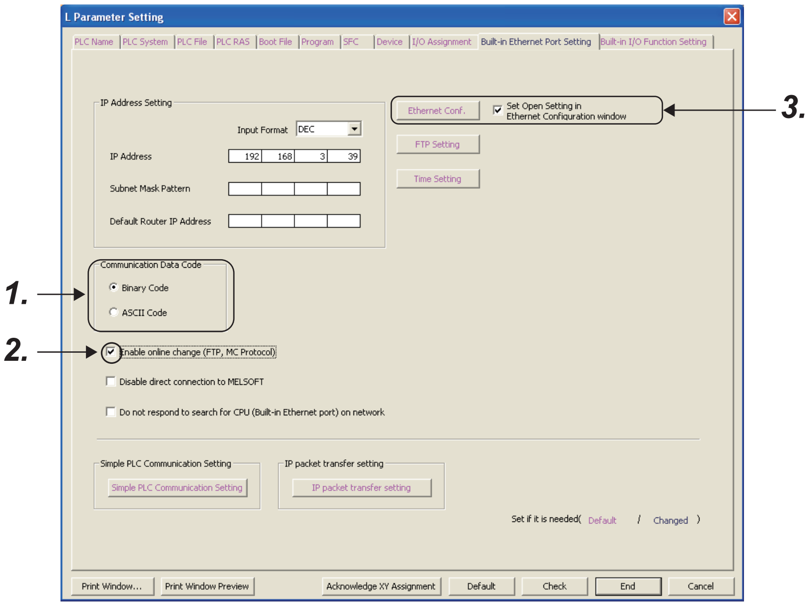
Remark

Access through routers is also available. When configuring the settings for it, set the subnet mask pattern and default router IP address. (☞ Page 28, Section 3.4)

5.1.1 Setting Method

Setting for communication using the MC protocol is described below.

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



L Parameter Setting

IP Address Setting

Input Format: DEC

IP Address: 192 . 168 . 3 . 39

Subnet Mask Pattern:

Default Router IP Address:

Communication Data Code

● Binary Code

○ ASCII Code

☑ Enable online change (FTP, MC Protocol)

□ Disable direct connection to MELSOFT

□ Do not respond to search for CPU (Built-in Ethernet port) on network

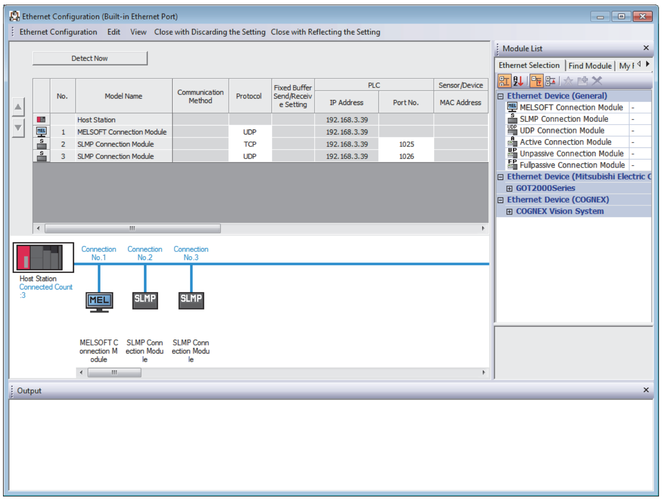
☑ Set Open Setting in Ethernet Configuration window

PLC Name
PLC System
PLC File
PLC RAS
Boot File
Program
SFC
Device
I/O Assignment
Built-in Ethernet Port Setting
Built-in I/O Function Setting
Ethernet Conf.
FTP Setting
Time Setting
Simple PLC Communication Setting
Simple PLC Communication Setting
IP packet transfer setting
IP packet transfer setting
Set if it is needed(Default / Changed)
Print Window...
Print Window Preview
Acknowledge XY Assignment
Default
Check
End
Cancel
1.
2.
3.

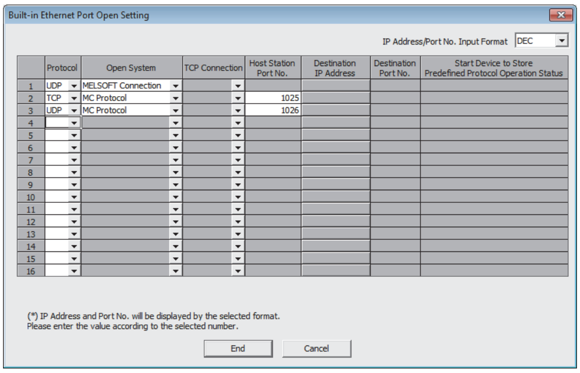
1. **Select Binary or ASCII code as the communication data code used for MC protocol.**
2. **Select the "Enable online change (FTP, MC Protocol)" checkbox to enable data to be written to the CPU module even in the RUN state.**

3. Set connections used for MC protocol communication.

Project window ➪ [Parameter] ➪ [PLC Parameter] ➪ [Built-in Ethernet Port Setting] ➪ [Ethernet Conf.] or [Open Setting] button



No.	Model Name	Communication Method	Protocol	Fixed Buffer Send/Receive Setting	PLC		Sensor/Device
					IP Address	Port No.	MAC Address
	Host Station				192.168.3.39		
1	MELSOFT Connection Module		UDP		192.168.3.39		
2	SLMP Connection Module		TCP		192.168.3.39	1025	
3	SLMP Connection Module		UDP		192.168.3.39	1026	



IP Address/Port No. Input Format: DEC

	Protocol	Open System	TCP Connection	Host Station Port No.	Destination IP Address	Destination Port No.	Start Device to Store Predefined Protocol Operation Status
1	UDP	MELSOFT Connection					
2	TCP	MC Protocol		1025			
3	UDP	MC Protocol		1026			
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.

- For "Ethernet Conf.", drag and drop "SLMP Connection Module" from "Module List" to the left side on the window. Select a protocol from "Protocol" depending on the target device. Set the port number of the host station in "Port No.". (Setting range: 1025 to 4999, 5010 to 65534) Do not specify 5000 to 5009 because these ports are used by the system. (☞ Page 221, Appendix 2)

- For the "Open Setting"

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 $FFFE_H$ (1025 to 4999, 5010 to 65534))[*1]

[*1] Do not specify 1388_H to 1391_H (5000 to 5009) because these ports are used by the system. (☞ Page 221, 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 an NAK message will be returned.

5.1.2 MC Protocol Commands

(1) Command list

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

○: Available, ×: N/A

Function			Command (Subcommand)*1	Description	Number of processed points	CPU module status		
						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*2	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	○	○	×
		In units of words*2	1402 (0000)	Sets or resets bit devices in units of 16 or 32 points by randomly specifying the target.	*5			
				Writes word devices in units of one or two points by randomly specifying the target.				
	Monitor registration*2*3*4	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 These commands are for QnA-compatible 3E frames.

*2 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.

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

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

*5 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) Applicable devices

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

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. Note that the access to a local device is not possible.	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	$C1_H$		Decimal
		Coil	TC	$C0_H$		
		Current value	TN	$C2_H$		
	Retentive timer	Contact	SS	$C7_H$		Decimal
		Coil	SC	$C6_H$		
		Current value	SN	$C8_H$		
	Counter	Contact	CS	$C4_H$		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	$A2_H$	The number range of a device in a CPU module, which is accessed to, can be specified.	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	$A9_H$		Decimal
Index register			Z*	CC_H	The number range of a device in a CPU module, which is accessed to, can be specified. Note that the access to a local device is not possible.	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*	$A8_H$		Decimal
Extended link register			W*	$B4_H$		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] For the L02CPU and L02CPU-P, devices of DX/DY400 or later number cannot be used. Use X/Y devices to access devices of X/Y400 or later. For the L06CPU, L06CPU-P, L26CPU, L26CPU-P, L26CPU-BT, and L26CPU-PBT, devices of DX/DY1000 or later number cannot be used. Use X/Y devices to access devices of X/Y1000 or later.

5.1.3 Precautions

(1) Number of devices

Only the external devices set in "Ethernet Conf." or "Open Setting" can be connected concurrently using MC protocol.

🖱 Project window ⇨ [Parameter] ⇨ [PLC Parameter] ⇨ [Built-in Ethernet Port Setting] ⇨ [Ethernet Conf.] or [Open Setting] button

(2) Data communication frame

The QnA-compatible 3E frames only are applicable to CPU modules.

(3) Access range

- Only the connected CPU module can be accessed. Accessing another module will cause an error.
- Accessing another station on a network such as CC-Link is not allowed via the connected CPU module.

(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.

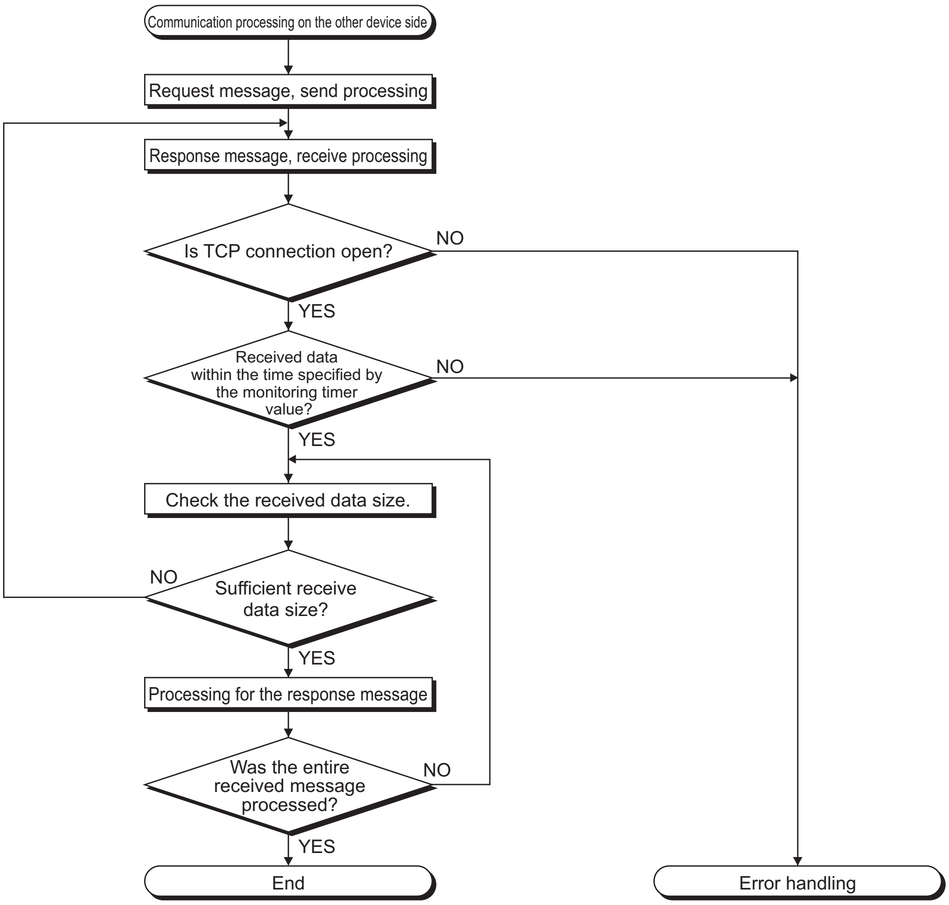
(5) File access during MC communication

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.


Communication processing on the other device side
Request message, send processing
Response message, receive processing
Is TCP connection open?
NO
YES
Received data within the time specified by the monitoring timer value?
NO
YES
Check the received data size.
NO
Sufficient receive data size?
YES
Processing for the response message
Was the entire received message processed?
NO
YES
End
Error handling

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.

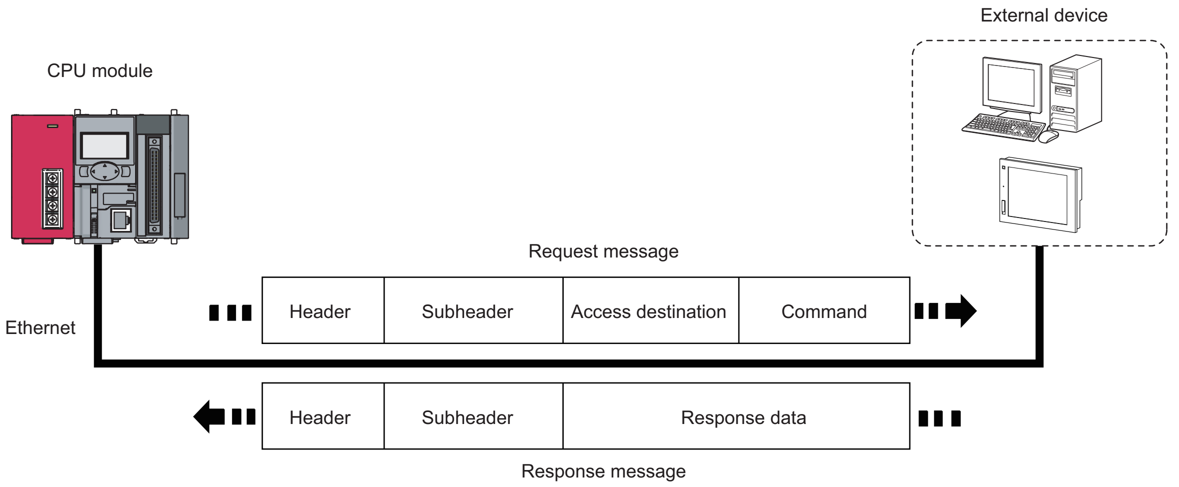
5.1.4 Error Codes for MC Protocol Communication

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 actions to be taken.

Error code (Hexadecimal)	Description	Action
4000_H to $4FFF_H$	Errors detected by the CPU module (Errors occurred in other than MC protocol communication)	Refer to the following. 📖 MELSEC-L CPU Module 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.	• Before enabling online change, write the data. • Change the operating status of the CPU module 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.	• 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.
$C059_H$	• The command and/or subcommand are specified incorrectly. • The CPU module does not support the command and/or subcommand.	• 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.	• 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.)	• 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.	• 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.
$C201_H$	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.

5.2 Sending a Command from the CPU Module to an External Device

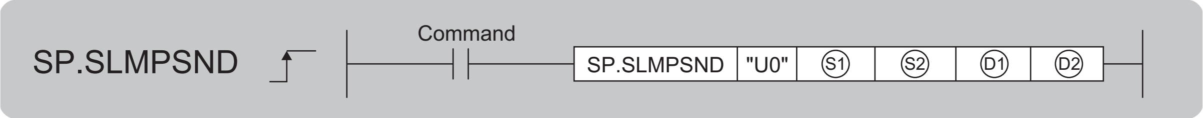
MC protocol messages (QnA-compatible 3E frame) can be sent from the CPU module to external devices on the Ethernet network.
To send the messages, use the SLMP frame send instruction.



Remark

Before sending the messages from the CPU module to external devices by using the SLMP frame send instruction, check the version of the CPU module used. (☞ Page 222, Appendix 3)

5.2.1 Sending an SLMP frame (SP.SLMPSND)



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

[*1] Local devices cannot be used.

(1) Setting data

Setting data	Description	Set by[*1]	Data type
U0	Dummy	—	Character string
S1	Start number of the device from which control data are stored	User, system	Device name
S2	Start number of the device from which a request frame is stored	User	Device name
D1	Start number of the device from which a response frame is stored	System	Device name
D2	Start number of the device which turns on for one scan upon completion of the instruction D2+1 also turns on when the instruction is completed with an error.	System	Bit

[*1] The "Set by" column indicates the following.
User: The data must be set before the SP.SLMPSND instruction is executed.
System: The CPU module stores the execution result of the SP.SLMPSND instruction.

(2) Control data

Device	Item	Description	Setting range	Set by[*1]
Ⓢ1+0	Execution/error completion type	b15 b7 b0 1 0 [2] 0 [1] [1] Execution type • 0: Without arrival check (The instruction is regarded as completed when a request message is sent from the host station.)[*2] • 1: With arrival check (The instruction is regarded as completed when a response message is received from the external device.) [2] Error completion type Specify whether to set data at completion with an error. • Ⓢ10: Do not set data in Ⓢ1+13 and later at completion with an error. (Clear Ⓢ1+13 or later.) • 1: Set data in Ⓢ1+13 or later at completion with an error.	—	User
Ⓢ1+1	Completion status	The completion status is stored. 0000_H: Completed normally Other than 0000_H: Completed with an error (error code)	—	System
Ⓢ1+2	Host station channel No.	Fixed to 1	1	User
Ⓢ1+3	IP address of external device (third and fourth octets)	Specify the IP address (third and fourth octets) of external device. b15 b8 b7 b0 3 4 3, 4: Indicates the octets of the IP address.	00000001_H to $FFFFFFFE_H$ (both Ⓢ1+3 and Ⓢ1+4 together) (1 to 4294967294)	User
Ⓢ1+4	IP address of external device (first and second octets)	Specify the IP address (first and second octets) of external device. b15 b8 b7 b0 1 2 1, 2: Indicates the octets of the IP address.		User
Ⓢ1+5	Target device port number	Specify the port number of the target device.	1 to 65534 (1 to $FFFE_H$)	User
Ⓢ1+6	Request destination network number	Fixed to 0000_H	0000_H	User
Ⓢ1+7	Request destination station number	Fixed to $00FF_H$	$00FF_H$	User
Ⓢ1+8	Request destination module I/O number	Specify an access target module. • $03FF_H$: Host station/control CPU • $03E0_H$: Multiple CPU No.1 • $03E1_H$: Multiple CPU No.2 • $03E2_H$: Multiple CPU No.3 • $03E3_H$: Multiple CPU No.4 • $03D0_H$: Control system CPU • $03D1_H$: Standby system CPU • $03D2_H$: System A CPU • $03D3_H$: System B CPU	$03FF_H$, $03E0_H$ to $03E3_H$, $03D0_H$ to $03D3_H$	User

Device	Item	Description	Setting range	Set by[*1]
Ⓢ1+9	Request destination multidrop station number	Fixed to 0000_H	0000_H	User
Ⓢ1+10	Number of resends	The device becomes effective when the execution type specified by Ⓢ1+0 is "1: With arrival check". ■**Before instruction execution** Specify the number of resends to be performed if the instruction is not completed within the monitoring time specified by Ⓢ1+11. • Setting range: 0 to 15 ■**At completion of instruction** The number of resends performed (result) is stored. • Setting range: 0 to 15	0 to 15	User/system
Ⓢ1+11	Arrival monitoring time	Specify the monitoring time until completion of processing. If processing is not completed within the monitoring time, the request is resent for the number of resends specified in Ⓢ1+10. 0: 10 seconds 1 to 32767: 1 to 32767 seconds	0 to 32767	User
Ⓢ1+12	Clock setting flag	The validity status (valid or invalid) of the data in Ⓢ1+13 and later is stored. The data in Ⓢ1+13 and later is cleared when the instruction is completed successfully. 0: Invalid 1: Valid	—	System
Ⓢ1+13	Clock data	Upper 8 bits: Month (01_H to 12_H) Lower 8 bits: Last two digits of the year (00_H to 99_H)	—	System
Ⓢ1+14		Upper 8 bits: Hour (00_H to 23_H) Lower 8 bits: Day (01_H to 31_H)		
Ⓢ1+15		Upper 8 bits: Second (00_H to 59_H) Lower 8 bits: Minute (00_H to 59_H)		
Ⓢ1+16		Upper 8 bits: First two digits of the year (00_H to 99_H) Lower 8 bits: Day of the week (00_H (Sun) to 06_H (Sat))		
Ⓢ1+17	IP address of error detected device (third and fourth octets)	The IP address (third and fourth octets) of the device where an error was detected is stored. b15 b8 b7 b0 | 3 | 4 | 3, 4: Indicates the octets of the IP address.	—	System
Ⓢ1+18	IP address of error detected device (first and second octets)	The IP address (first and second octets) of the device where an error was detected is stored. b15 b8 b7 b0 | 1 | 2 | 1, 2: Indicates the octets of the IP address.		

[*1] The "Set by" column indicates the following.
User: The data must be set before the SP.SLMPSND instruction is executed.
System: The CPU module stores the execution result of the SP.SLMPSND instruction.

[*2] If Ⓢ1+0 is set to "0: Without arrival check", receive data is not set. Set 0 in Ⓢ1+0 in the following cases:
· When a command that does not return a response message is used
· When a response message is not referred to

(3) Request frame

Device	Item	Description	Setting range	Set by[*1]
S2+0	Request data length	Specify the data length from the monitoring timer to the request data. (In units of bytes)	1 to 2000	User
S2+1	Monitoring timer	This timer sets the waiting time for the external device that received a request message to wait for the response after it issued a processing request to the access destination. (Unit: Increments of 250ms) 0: Infinite wait 1 to 65535: 1 to 65535 × 250ms	0 to 65535	User
S2+2 to S2+n	Request data	The request data of the MC protocol message is stored.	—	User

*1 The "Set by" column indicates the following.
User: The data must be set before the SP.SLMPSND instruction is executed.

(4) Response frame

Device	Item	Description	Setting range	Set by[*1]
D1+0	Response data length	The data length from the end code to the response data is stored. (In units of bytes)	2 to 2000	System
D1+1	End code	The result of command processing is stored. In normal end, 0 is stored. In abnormal end, an error code set by the external device is stored.	—	System
D1+2 to D1+n	Response data	Execution results for the request data are set. (Some commands do not return response data.)	—	System

*1 The "Set by" column indicates the following.
System: The CPU module stores the execution result of the SP.SLMPSND instruction.

(5) Function

- This instruction sends the request frame in the device specified by S2 and later to the external device specified by the external device IP address in the control data. When a response message is received from the external device, it is stored in the device specified by D1.

The following figures show the request data and the response data in normal/abnormal end.

■ Request data

Header	Subheader	Request destination network No.	Request destination station No.	Request destination module I/O No.	Request destination multidrop station No.	Request data length	Monitoring timer	Request data	Footer
S1+3 to S1+5		S1+6	S1+7	S1+8	S1+9	S2+0	S2+1	S2+2 to S2+n	

Specified by S1 (except some devices)

Specified by S2

■ Response data

When completed normally

Header	Subheader	Request destination network No.	Request destination station No.	Request destination module I/O No.	Request destination multidrop station No.	Response data length	End code	Response data	Footer
						D1+0	D1+1	D1+2 to D1 +n	

Stored in D1

When completed with an error

Header	Subheader	Request destination network No.	Request destination station No.	Request destination module I/O No.	Request destination multidrop station No.	Response data length	End code	Network No. (responding station)	Station No. (responding station)	Request destination module I/O No.	Request destination multidrop station No.	Command	Subcommand	Footer
						D1+0	D1+1							

Error information

Stored in D1

- The SP.SLMPSND instruction communicates using UDP. Set the external device to use UDP.
- The SP.SLMPSND instruction communicates in binary code. Match the setting of the external device also with the binary code.
- When the setting is configured in "Simple PLC Communication Setting" of the PLC parameter, the SP.SLMPSND instruction cannot be executed.

- The result of the SP.SLMPSND instruction can be checked with the completion device, Ⓓ2+0 and Ⓓ2+1.

• Completion device Ⓓ2+0

This device turns on at the END processing of the scan in which the SP.SLMPSND instruction is completed and turns off at the next END processing.

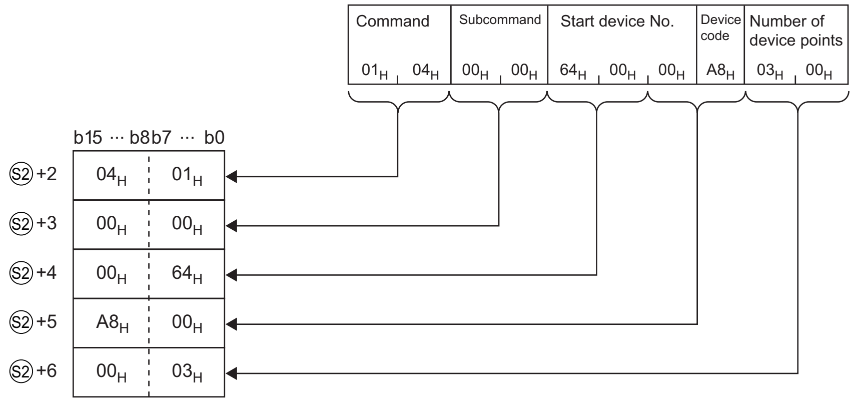
• Completion device Ⓓ2+1

This device turns on or off according to the result of the SP.SLMPSND instruction.

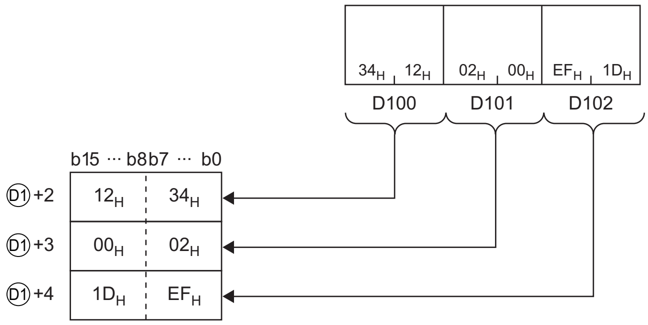
State	Description
When completed normally	Remains off.
When completed with an error	Turns on at the END processing of scan in which the SP.SLMPSND instruction is completed and turns off at the next END processing. When the instruction is completed with an error, an error code is stored in the completion status of Ⓢ1+1.

Ex. Sending "Batch read in word units (command: 0401_H)" which reads the value in D100 to D102

■ Request data



■ Response data



(6) Error

Each of the following events results in an operation error, and the error flag (SM0) turns on and an error code is stored in SD0.

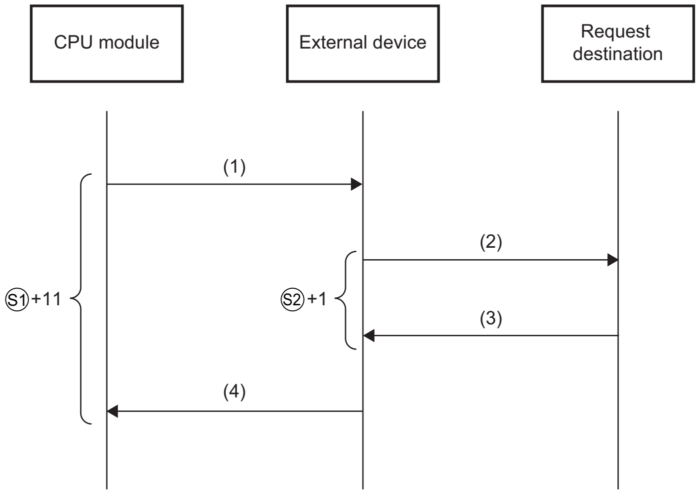
- An instruction is used in a CPU module with a serial number (first five digits) of "18111" or earlier.
  (Error code: 4002)
- The value specified by S1+2 as the host station channel number is outside the specified range.
  (Error code: 4101)
- The value specified by S2+0 as the request data length is out of the range of 1 to 2000.
  (Error code: 4100)
- The device specified by S1, S2, D1, or D2 exceeds the range of the number of device points.
  (Error code: 4101)
- A device which cannot be specified is specified.
  (Error code: 4004)
- An instruction has been executed when the simple PLC communication is set in the PLC parameter.
  (Error code: 4161)

When the instruction completes with an error, the completion status indication device D2+1 turns on and an error code is stored in the completion status S1+1. For the error code stored in the completion status S1+1, refer to the following.

📖 MELSEC-L CPU Module User's Manual (Hardware Design, Maintenance and Inspection)

(7) Precautions

- Do not execute multiple SP.SLMPSND instructions concurrently. Multiple SP.SLMPSND instructions specifying the same channel cannot be used concurrently.
- Specify the arrival monitoring time (Ⓢ1+11) of the control data and monitoring timer (Ⓢ2+1) of the request frame so that the arrival monitoring time ≥ monitoring timer.



(1)Request message

(2)Processing request from external device to request destination

(3)Processing response from request destination to external device

(4)Response message

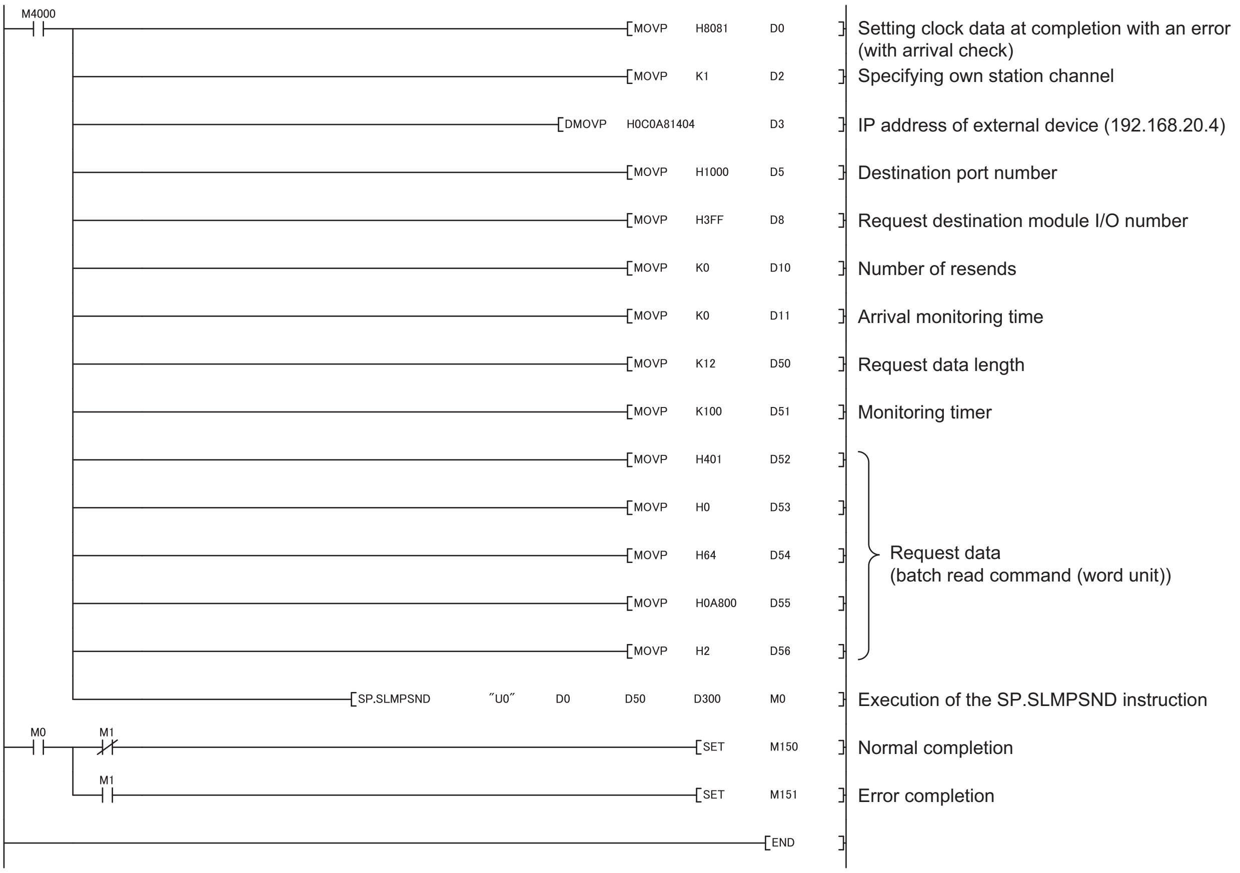
Point

The SP.SLMPSND instruction is successfully completed even if the external device returns an abnormal response. When the SP.SLMPSND instruction is completed successfully, whether the response is normal or abnormal can be identified by the end code of the response frame. When an abnormal response is returned, check the manual of the external device being used and take corrective action.

- An available port number F000$_H$ to FFFE$_H$ is used for the SP.SLMPSND instruction as a host station port number. Therefore, during the execution of the SP.SLMPSND instruction, do not specify F000$_H$ to FFFEH as a host station port number for other functions such as the SP.SOCOPEN instruction and the iQ Sensor Solution function of the built-in Ethernet port. If these numbers are set, the function may not be completed properly.

(8) Program example

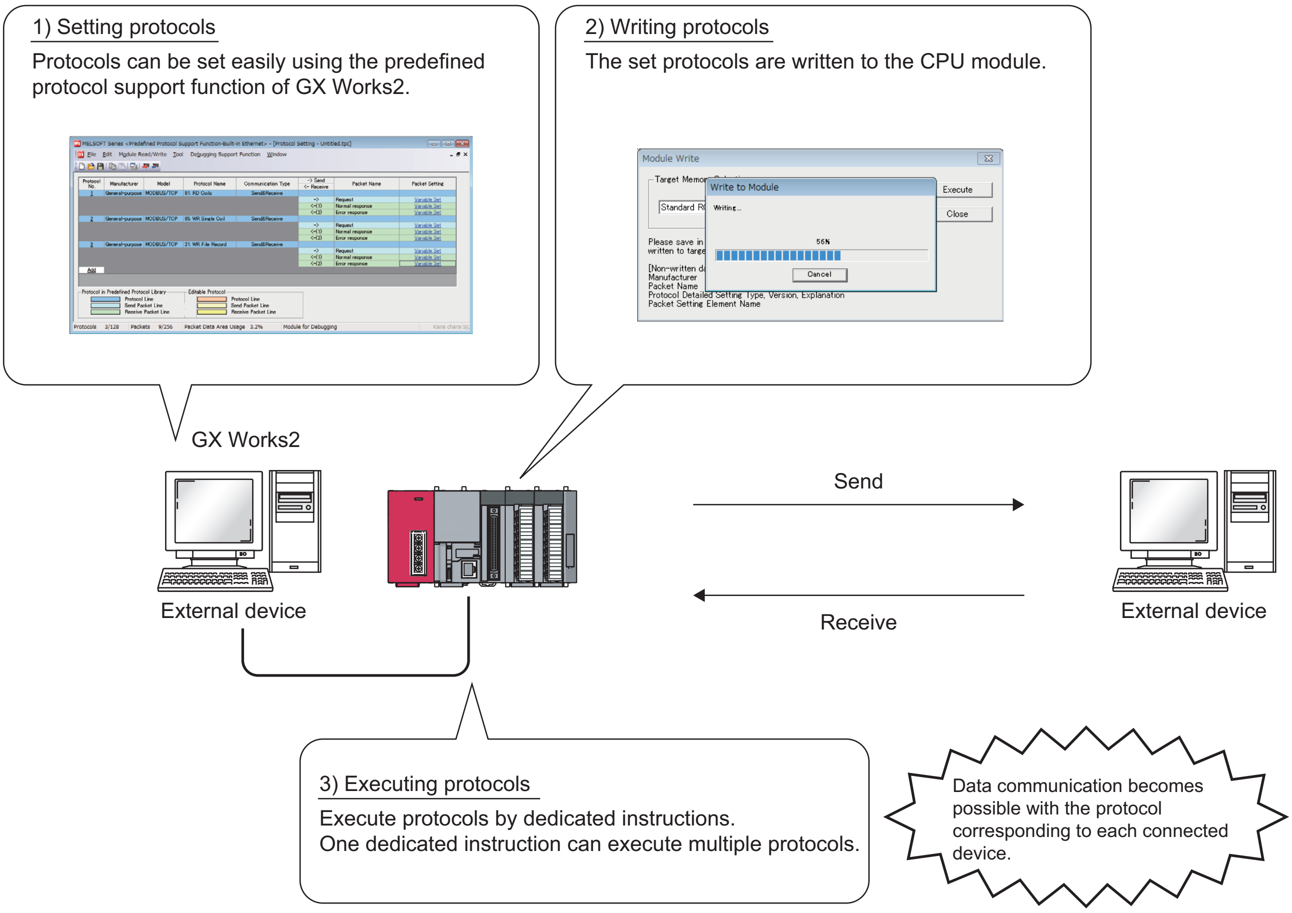
When M4000 is turned on, this sample program sends "Batch read in word units (command: 0401_H)" to the CPU module on another station for which the IP address has been set to 192.168.20.4, and reads the values in D100 to D101 (two words).



```
M4000
[MOVP   H8081        D0  ]  Setting clock data at completion with an error
                            (with arrival check)
[MOVP   K1           D2  ]  Specifying own station channel
[DMOVP  H0C0A81404   D3  ]  IP address of external device (192.168.20.4)
[MOVP   H1000        D5  ]  Destination port number
[MOVP   H3FF         D8  ]  Request destination module I/O number
[MOVP   K0           D10 ]  Number of resends
[MOVP   K0           D11 ]  Arrival monitoring time
[MOVP   K12          D50 ]  Request data length
[MOVP   K100         D51 ]  Monitoring timer
[MOVP   H401         D52 ]
[MOVP   H0           D53 ]
[MOVP   H64          D54 ]  Request data
[MOVP   H0A800       D55 ]  (batch read command (word unit))
[MOVP   H2           D56 ]
[SP.SLMPSND  "U0"  D0  D50  D300  M0 ]  Execution of the SP.SLMPSND instruction
M0  M1(NC)
[SET    M150 ]  Normal completion
    M1
[SET    M151 ]  Error completion
[END ]
```

CHAPTER 6 DATA COMMUNICATIONS USING THE PREDEFINED PROTOCOL

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.



1) Setting protocols

Protocols can be set easily using the predefined protocol support function of GX Works2.

2) Writing protocols

The set protocols are written to the CPU module.

3) Executing protocols

Execute protocols by dedicated instructions.
One dedicated instruction can execute multiple protocols.

Data communication becomes possible with the protocol corresponding to each connected device.

Remark

When using the predefined protocol function, check the versions of the CPU module and GX Works2. (☞ Page 222, 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		SP.ECPRTCL 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

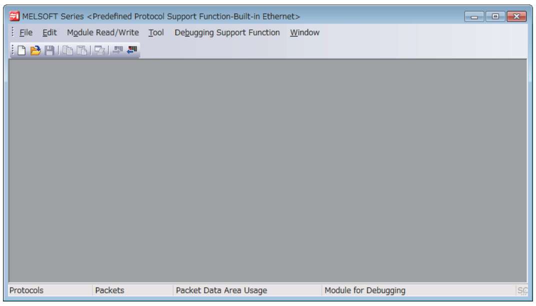
*3 Size as a sum of all packets

6.2 Setting Method

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

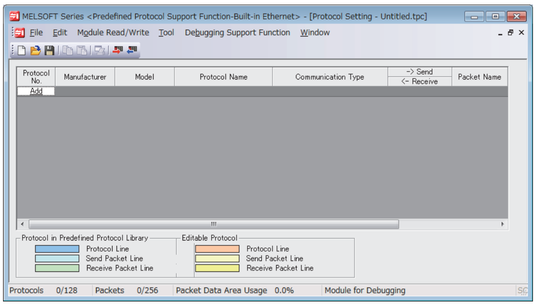
1. Open the "Predefined Protocol Support Function" window.

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


MELSOFT Series <Predefined Protocol Support Function-Built-in Ethernet>
File Edit Module Read/Write Tool Debugging Support Function Window
Protocols
Packets
Packet Data Area Usage
Module for Debugging

2. Create a new file.

[File] ➪ [New] ➪ "Add"

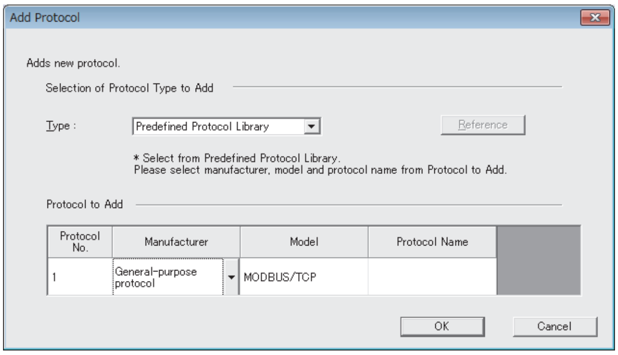

MELSOFT Series <Predefined Protocol Support Function-Built-in Ethernet> - [Protocol Setting - Untitled.tpc]
File Edit Module Read/Write Tool Debugging Support Function Window

Protocol No.	Manufacturer	Model	Protocol Name	Communication Type	-> Send <- Receive	Packet Name
Add						

Protocol in Predefined Protocol Library
Protocol Line
Send Packet Line
Receive Packet Line
Editable Protocol
Protocol Line
Send Packet Line
Receive Packet Line
Protocols 0/128
Packets 0/256
Packet Data Area Usage 0.0%
Module for Debugging

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.



Adds new protocol.

Selection of Protocol Type to Add

Type : Predefined Protocol Library

* Select from Predefined Protocol Library.
Please select manufacturer, model and protocol name from Protocol to Add.

Protocol to Add

Protocol No.	Manufacturer	Model	Protocol Name
1	General-purpose protocol	MODBUS/TCP	

Point

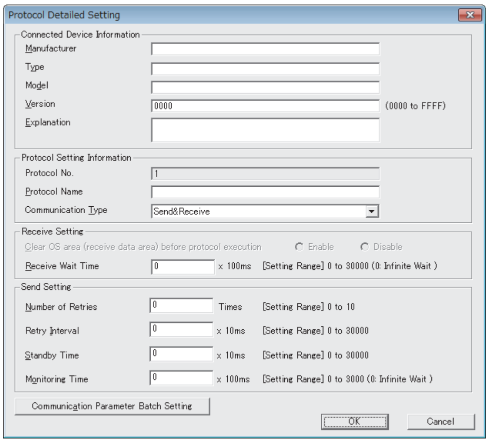
Select "User Protocol Library" in the "Add Protocol" window to read the protocol stored by the user. For details on the user protocol library, refer to the following.

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]


Protocol Detailed Setting
Communication Parameter Batch Setting
OK
Cancel

Connected Device Information
Manufacturer
Type
Model
Version 0000 (0000 to FFFF)
Explanation

Protocol Setting Information
Protocol No. 1
Protocol Name
Communication Type Send&Receive

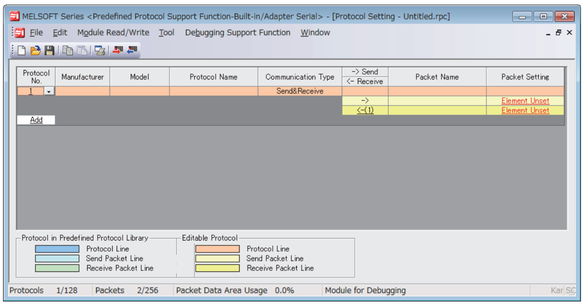
Receive Setting
Clear OS area (receive data area) before protocol execution ○ Enable ○ Disable
Receive Wait Time 0 x 100ms [Setting Range] 0 to 30000 (0: Infinite Wait)

Send Setting
Number of Retries 0 Times [Setting Range] 0 to 10
Retry Interval 0 x 10ms [Setting Range] 0 to 30000
Standby Time 0 x 10ms [Setting Range] 0 to 30000
Monitoring Time 0 x 100ms [Setting Range] 0 to 3000 (0: Infinite Wait)

5. Set the packet configuration.

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

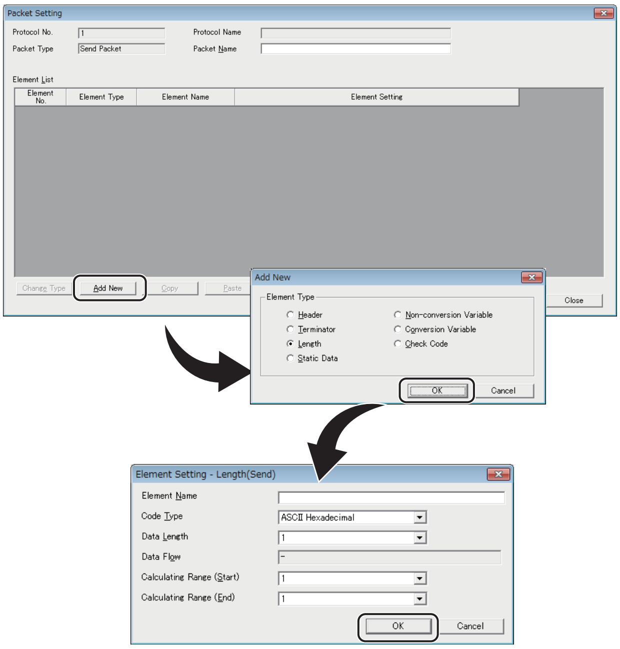
"Protocol Setting" window" ➪ "Variable Unset" or "Element Unset"


MELSOFT Series <Predefined Protocol Support Function-Built-in/Adapter Serial> - [Protocol Setting - Untitled.rpc]

Protocol No.	Manufacturer	Model	Protocol Name	Communication Type	-> Send <- Receive	Packet Name	Packet Setting
1				Send&Receive			
					->		Element Unset
					<-(1)		Element Unset

Protocols 1/128 Packets 2/256 Packet Data Area Usage 0.0% Module for Debugging

↓


Packet Setting

Protocol No. 1
Packet Type Send Packet
Protocol Name
Packet Name

Element List

Element No.	Element Type	Element Name	Element Setting

Add New

Element Type
- ○ Header
- ○ Terminator
- ● Length
- ○ Static Data
- ○ Non-conversion Variable
- ○ Conversion Variable
- ○ Check Code

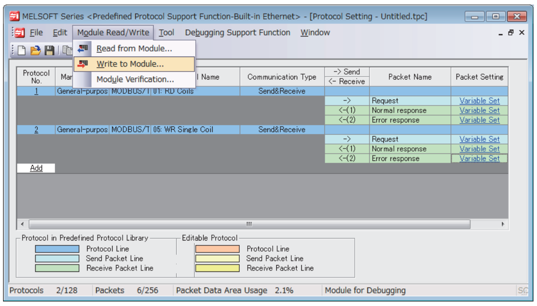
Element Setting - Length(Send)

Element Name
Code Type: ASCII Hexadecimal
Data Length: 1
Data Flow: -
Calculating Range (Start): 1
Calculating Range (End): 1

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]


MELSOFT Series <Predefined Protocol Support Function-Built-in Ethernet> - [Protocol Setting - Untitled.tpc]

Protocol No.	Mar		l Name	Communication Type	-> Send <- Receive	Packet Name	Packet Setting
1	General-purpos	MODBUS/T	01: RD Coils	Send&Receive			
					->	Request	Variable Set
					<-(1)	Normal response	Variable Set
					<-(2)	Error response	Variable Set
2	General-purpos	MODBUS/T	05: WR Single Coil	Send&Receive			
					->	Request	Variable Set
					<-(1)	Normal response	Variable Set
					<-(2)	Error response	Variable Set

Protocols 2/128 Packets 6/256 Packet Data Area Usage 2.1% Module for Debugging

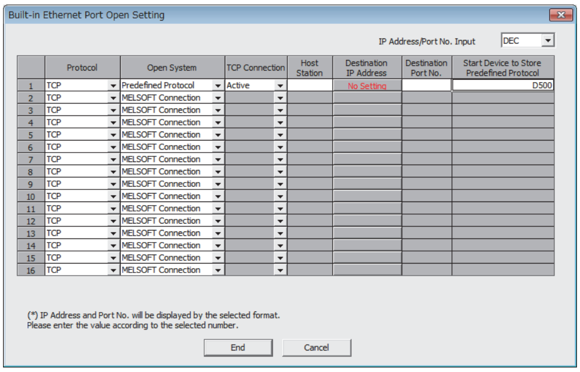
*1 The written protocol setting data will be enabled when
- the CPU module is powered on or is reset, or
- SM1355 (Predefined protocol setting check request) is turned on.

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]



Built-in Ethernet Port Open Setting

IP Address/Port No. Input: DEC

	Protocol	Open System	TCP Connection	Host Station	Destination IP Address	Destination Port No.	Start Device to Store Predefined Protocol
1	TCP	Predefined Protocol	Active		No Setting		D500
2	TCP	MELSOFT Connection					
3	TCP	MELSOFT Connection					
4	TCP	MELSOFT Connection					
5	TCP	MELSOFT Connection					
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.

Item	Description
Protocol	Select "TCP" or "UDP".
Open System	Select "Predefined Protocol".
TCP Connection	When "Protocol" is "TCP", select the connection method. (☞ Page 66, 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 $FFFE_H$/$FFFF_H$ (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 59, Section 6.2 (1)) The area of 19 words starting from the specified device is used. Available devices: D, W, R, ZR

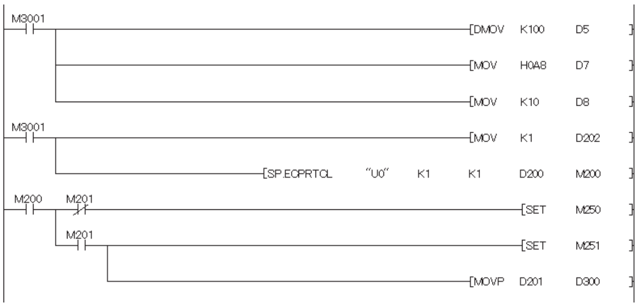
[*1] Do not specify 1388_H to 1391_H (5000 to 5009) because these ports are used by the system. (☞ Page 221, Appendix 2)

Point

When using the predefined protocol function, open setting in the Ethernet configuration window is not available.
Do not select the "Set Open Setting in Ethernet Configuration Window" checkbox in "Built-in Ethernet Port Setting" tab.

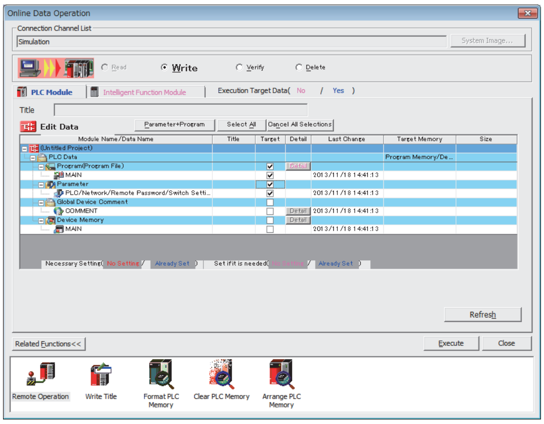
8. **Create a program for starting data communications.**

In the case of Active open, create the program in which SP.SOCOPEN instruction is used to establish a connection. (☞ Page 86, Section 7.4.1)
To activate the predefined protocol, the SP.ECPRTCL instruction is used. (☞ Page 61, Section 6.4.1)



9. **Write the program to the CPU module.**

⇨ [Online] ⇨ [Write to PLC]


Online Data Operation

Connection Channel List: Simulation

○ Read ● Write ○ Verify ○ Delete

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

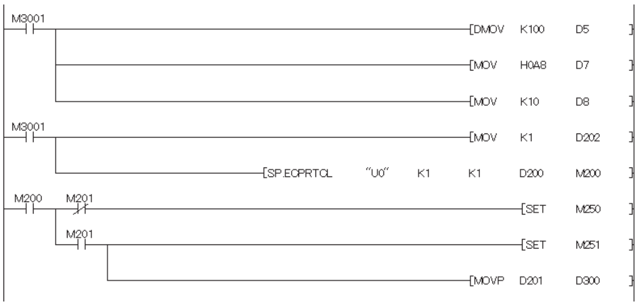
Title:

Edit Data — Parameter+Program | Select All | Cancel All Selections

Module Name/Data Name	Title	Target	Detail	Last Change	Target Memory	Size
(Untitled Project)						
PLC Data					Program Memory/De...	
Program(Program File)		☑	Detail			
MAIN		☑		2013/11/18 14:41:13		
Parameter		☑				
PLC/Network/Remote Password/Switch Setti...		☑		2013/11/18 14:41:13		
Global Device Comment		□				
COMMENT		□	Detail	2013/11/18 14:41:13		
Device Memory		□	Detail			
MAIN		□		2013/11/18 14:41:13		

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

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) • 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 59, Section 6.2 (1) (a)) • 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) • 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) • 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 on the protocol communication types, refer to the following.

📖 MELSEC-L 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. 📖 MELSEC-L 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 66, Section 7.1 Page 86, Section 7.4.1
SP.SOCCLOSE	Closes a connection with the external device.	Page 92, Section 7.4.2
SP.SOCCINF	Reads out connection information.	Page 106, Section 7.4.6
SP.SOCCSET	Changes the target of the connection for UDP/IP communications.	Page 109, Section 7.4.7
SP.ECPRTCL	Executes the configured predefined protocol.	Page 61, 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.

MELSEC-L 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 59, 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.)

📖 MELSEC-L 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

(a) To use two predefined protocol functions simultaneously (Ethernet or RS-232/422/485)

Store the predefined protocol setting files in the same drive. If the setting files are stored separately in the SD memory card and the standard ROM, the setting stored in the SD memory card is enabled while the setting stored in the standard ROM is disabled.
To enable the predefined protocol setting stored in the standard ROM, format the SD memory card or delete the predefined protocol setting stored in the SD memory card. Then, enable the setting.
The predefined protocol setting can be checked and deleted by operating the PLC user data.

🖱 [Online] ⇨ [PLC User Data] ⇨ [Read] or [Delete]

(b) After writing the 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

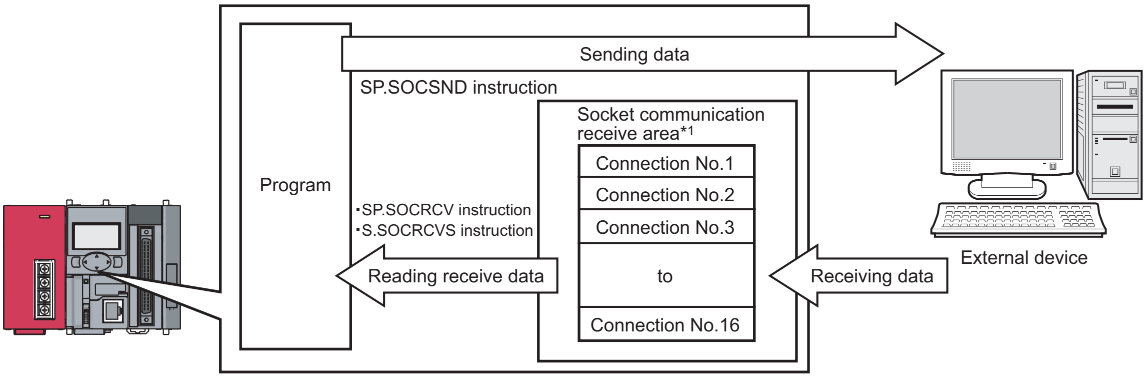
Data set by the predefined protocol function are communicated during the service processing. Therefore, if COM instruction is executed while b15 of SD778 (Refresh processing selection when the COM/CCOM instruction is executed) is on, data communications are also performed at the execution of the COM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

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.

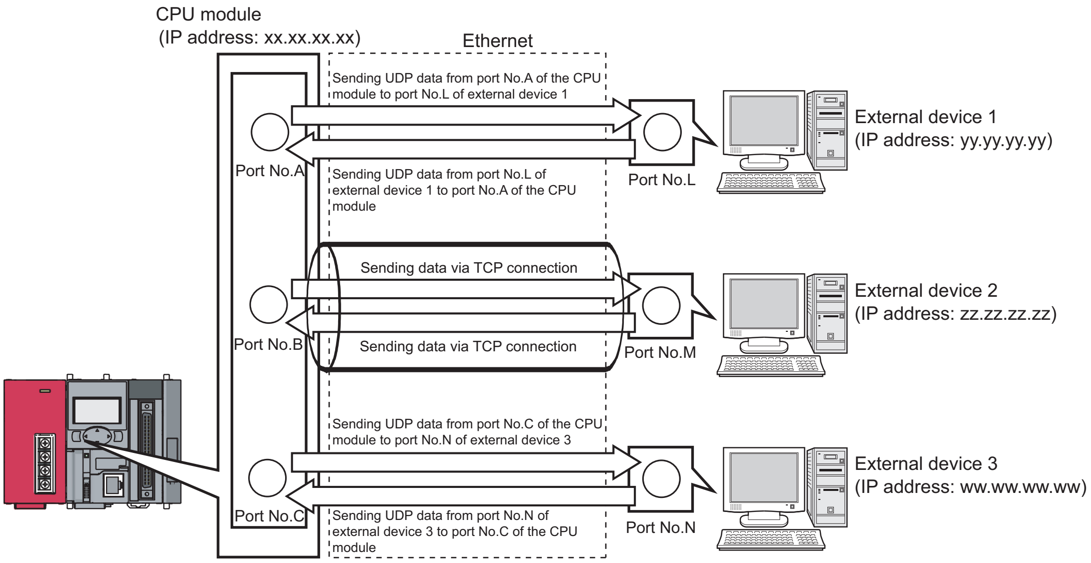
Remark

- For dedicated instructions used for the socket communication function, refer to Page 85, 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 28, Section 3.4)

(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.


CPU module
(IP address: xx.xx.xx.xx)
Ethernet
Sending UDP data from port No.A of the CPU module to port No.L of external device 1
Port No.A
Sending UDP data from port No.L of external device 1 to port No.A of the CPU module
Port No.L
External device 1
(IP address: yy.yy.yy.yy)
Sending data via TCP connection
Port No.B
Sending data via TCP connection
Port No.M
External device 2
(IP address: zz.zz.zz.zz)
Sending UDP data from port No.C of the CPU module to port No.N of external device 3
Port No.C
Sending UDP data from port No.N of external device 3 to port No.C of the CPU module
Port No.N
External device 3
(IP address: ww.ww.ww.ww)

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.

TCP connection method	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.

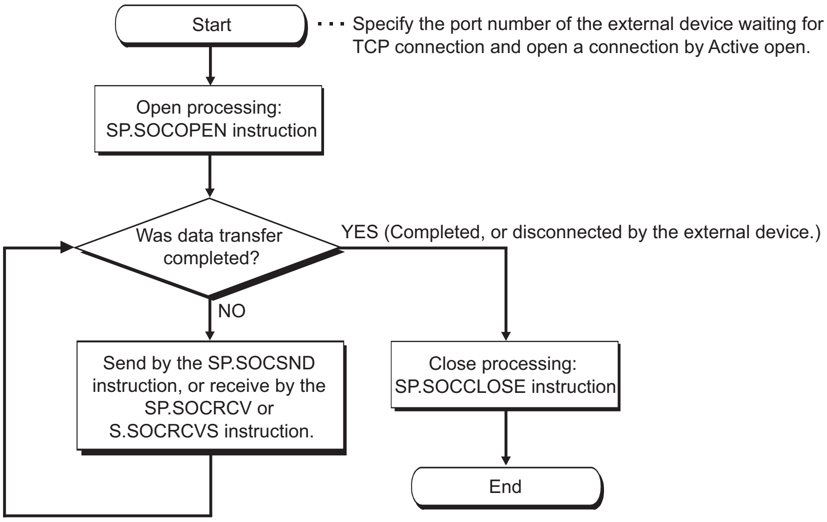
Remark

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

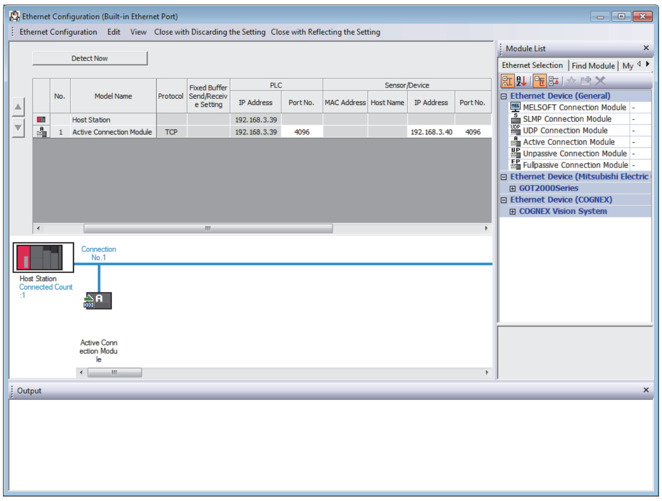
The following shows a communication flow of an Active open.


Start
· · · Specify the port number of the external device waiting for TCP connection and open a connection by Active open.
Open processing: SP.SOCOPEN instruction
Was data transfer completed?
YES (Completed, or disconnected by the external device.)
NO
Send by the SP.SOCSND instruction, or receive by the SP.SOCRCV or S.SOCRCVS instruction.
Close processing: SP.SOCCLOSE instruction
End

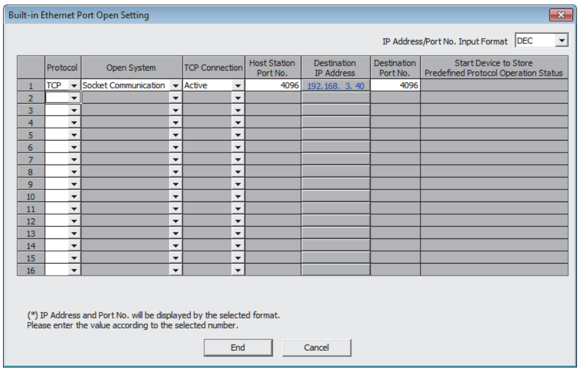
(a) Parameter setting

The following parameters are set for the sample program.

Project window ➪ [Parameter] ➪ [PLC Parameter] ➪ [Built-in Ethernet Port Setting] ➪ [Ethernet Conf.] or [Open Setting] button



No.	Model Name	Protocol	Fixed Buffer Send/Receive Setting	PLC		Sensor/Device			
				IP Address	Port No.	MAC Address	Host Name	IP Address	Port No.
	Host Station			192.168.3.39					
1	Active Connection Module	TCP		192.168.3.39	4096			192.168.3.40	4096



IP Address/Port No. Input Format: DEC

	Protocol	Open System	TCP Connection	Host Station Port No.	Destination IP Address	Destination Port No.	Start Device to Store Predefined Protocol Operation Status
1	TCP	Socket Communication	Active	4096	192.168. 3. 40	4096	
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.

- For "Ethernet Conf.", drag and drop "Active Connection Module" from "Module List" to the left side on the window. Set the port numbers and IP address as mentioned below.

Item		Description
PLC	Port No.	4096 (Setting range: 1 to 4999, 5010 to 65534) Do not specify 5000 to 5009 because these ports are used by the system. (☞ Page 221, Appendix 2)
Sensor/device	IP Address	192.168.3.40 (Setting range: 0.0.0.1 to 223.255.255.254)
	Port No.	4096 (Setting range: 1 to 65534)

- For the "Open Setting"

Item	Description
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 to 4999, 5010 to 65534)) [*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 to 65534))

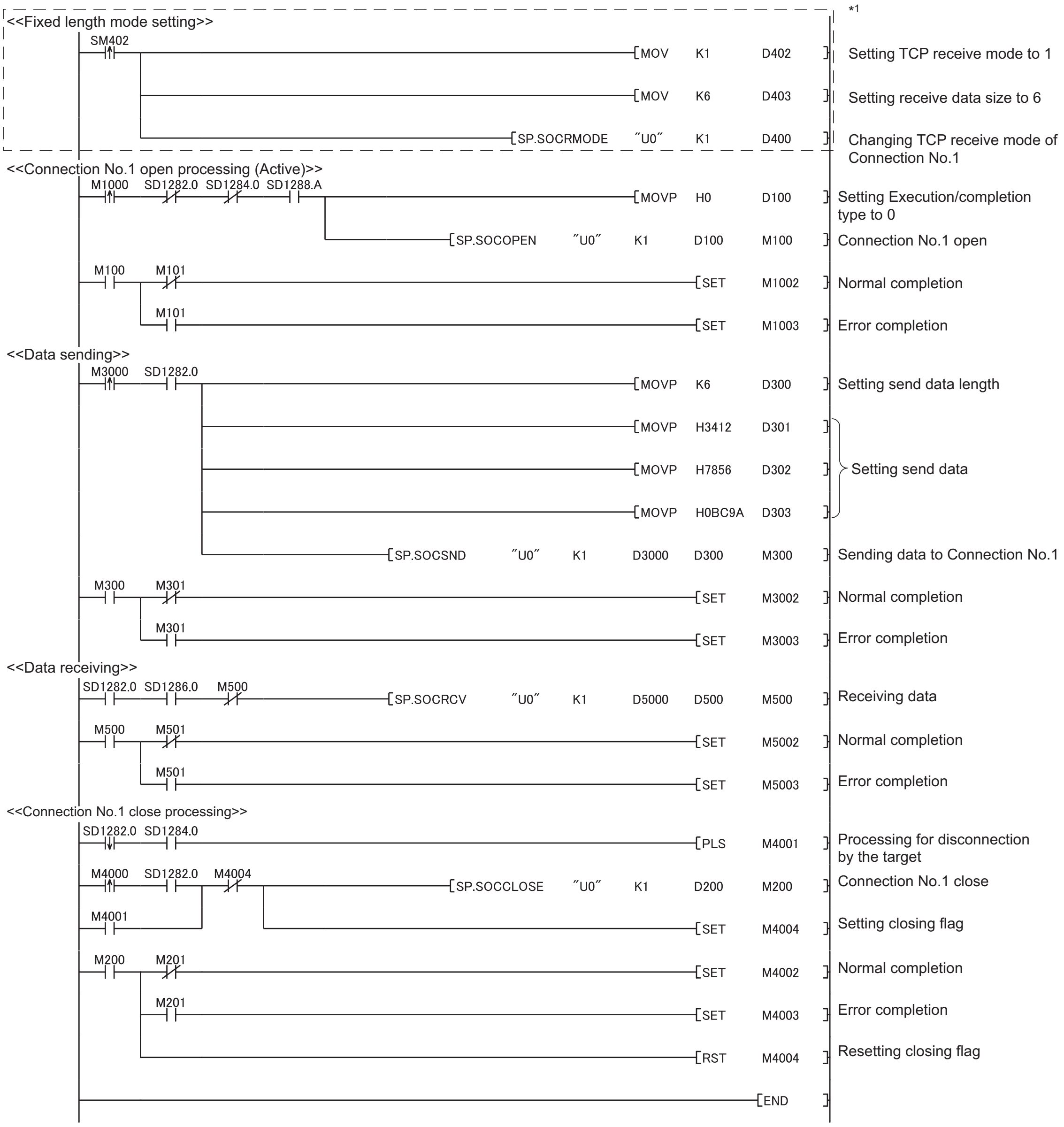
*1 Do not specify 1388$_H$ to 1391$_H$ (5000 to 5009) because these ports are used by the system. (☞ Page 221, Appendix 2)

(b) Devices used in programs

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.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

(c) Sample program

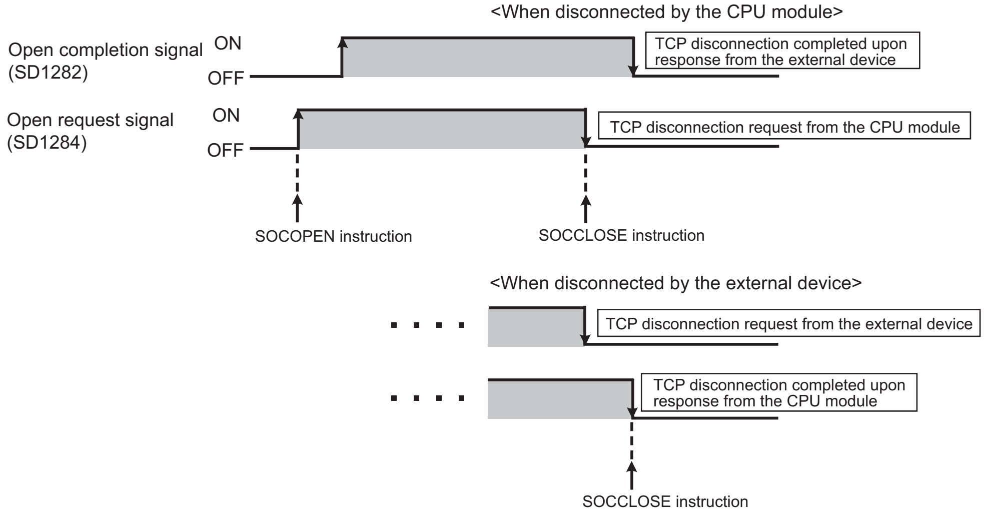


*1 There are two kinds of TCP receive modes: TCP standard receive mode and TCP fixed-length receive mode. For fixing the data size, run the program enclosed by a dotted line.
(It can be omitted when the data size is not fixed.)
For the TCP receive mode, refer to the section for the SP.SOCRMODE instruction. (☞ Page 111, Section 7.4.8)

(d) Precaution for Active open communication

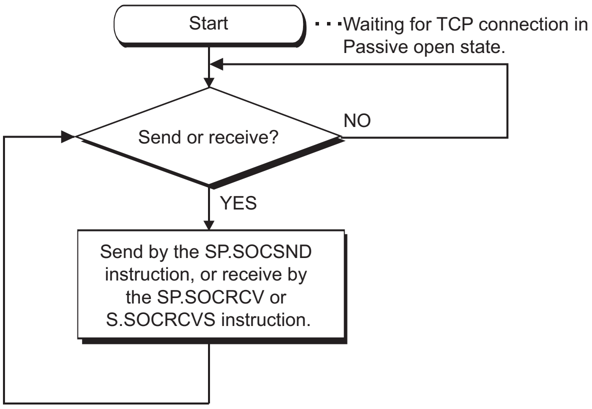
Configure an interlock circuit using the Open completion signal (SD1282) and Open request signal (SD1284) in the program.

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


<When disconnected by the CPU module>
Open completion signal (SD1282)
ON
OFF
TCP disconnection completed upon response from the external device
Open request signal (SD1284)
ON
OFF
TCP disconnection request from the CPU module
SOCOPEN instruction
SOCCLOSE instruction
<When disconnected by the external device>
TCP disconnection request from the external device
TCP disconnection completed upon response from the CPU module
SOCCLOSE instruction

(3) Program example for Passive open

The following shows a communication flow of a Passive open.

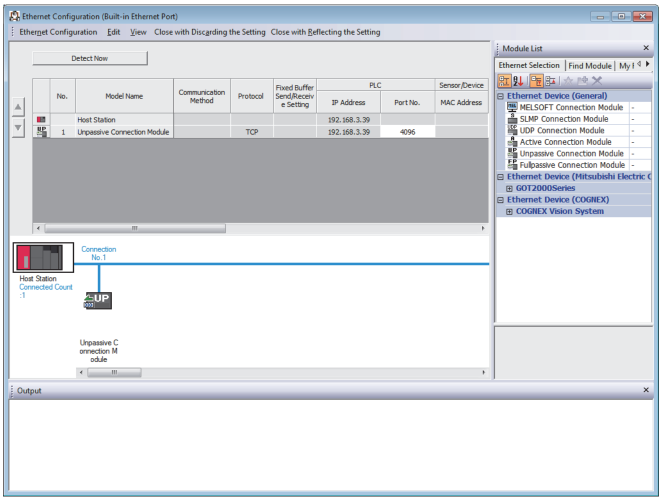


Start
· · ·Waiting for TCP connection in Passive open state.
Send or receive?
NO
YES
Send by the SP.SOCSND instruction, or receive by the SP.SOCRCV or S.SOCRCVS instruction.

(a) Parameter setting

The following parameters are set for the sample program.

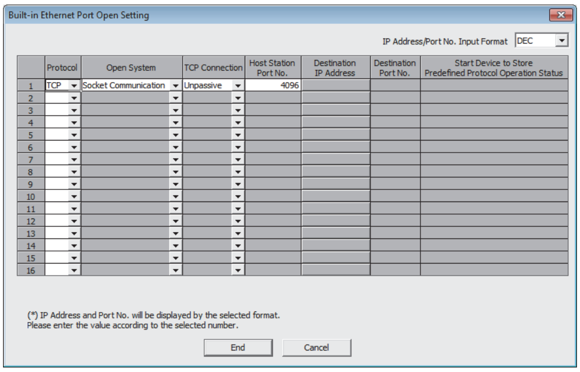
Project window ➪ [Parameter] ➪ [PLC Parameter] ➪ [Built-in Ethernet Port Setting] ➪ [Ethernet Conf.] or [Open Setting] button



Ethernet Configuration (Built-in Ethernet Port)
Ethernet Configuration
Edit
View
Close with Discarding the Setting
Close with Reflecting the Setting
Detect Now
Module List
Ethernet Selection
Find Module
Connection No.1
Host Station
Connected Count :1
UP
Unpassive Connection Module
Output

	No.	Model Name	Communication Method	Protocol	Fixed Buffer Send/Receive Setting	PLC		Sensor/Device
						IP Address	Port No.	MAC Address
		Host Station				192.168.3.39		
	1	Unpassive Connection Module		TCP		192.168.3.39	4096	

Module List:
- Ethernet Device (General)
  - MELSOFT Connection Module -
  - SLMP Connection Module -
  - UDP Connection Module -
  - Active Connection Module -
  - Unpassive Connection Module -
  - Fullpassive Connection Module -
- Ethernet Device (Mitsubishi Electric C
  - GOT2000Series
- Ethernet Device (COGNEX)
  - COGNEX Vision System



IP Address/Port No. Input Format: DEC

	Protocol	Open System	TCP Connection	Host Station Port No.	Destination IP Address	Destination Port No.	Start Device to Store Predefined Protocol Operation Status
1	TCP	Socket Communication	Unpassive	4096			
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.

- For "Ethernet Conf.", drag and drop "Unpassive Connection Module" or "Fullpassive Connection Module" from "Module List" to the left side on the window. Set the port numbers and IP address as mentioned below.

Item		Description
PLC	Port No.	4096 (Setting range: 1 to 4999, 5010 to 65534) Do not specify 5000 to 5009 because these ports are used by the system. (☞ Page 221, Appendix 2)
Sensor/device	IP Address	Blank When "Fullpassive Connection Module" is selected, a value must be set. (Setting range: 0.0.0.1 to 223.255.255.254)
	Port No.	Blank When "Fullpassive Connection Module" is selected, a value must be set. (Setting range: 1 to 65534)

- For the "Open Setting"

Item	Description
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 $FFFE_H$ (1 to 4999, 5010 to 65534))[*1]
Destination IP Address	Blank. When "Fullpassive" is selected for "TCP Connection", a value must be set. (Setting range: 0.0.0.1 to 223.255.255.254)
Destination Port No.	Blank. When "Fullpassive" is selected for "TCP Connection", a value must be set. (Setting range: 0001_H to $FFFE_H$ (1 to 65534))

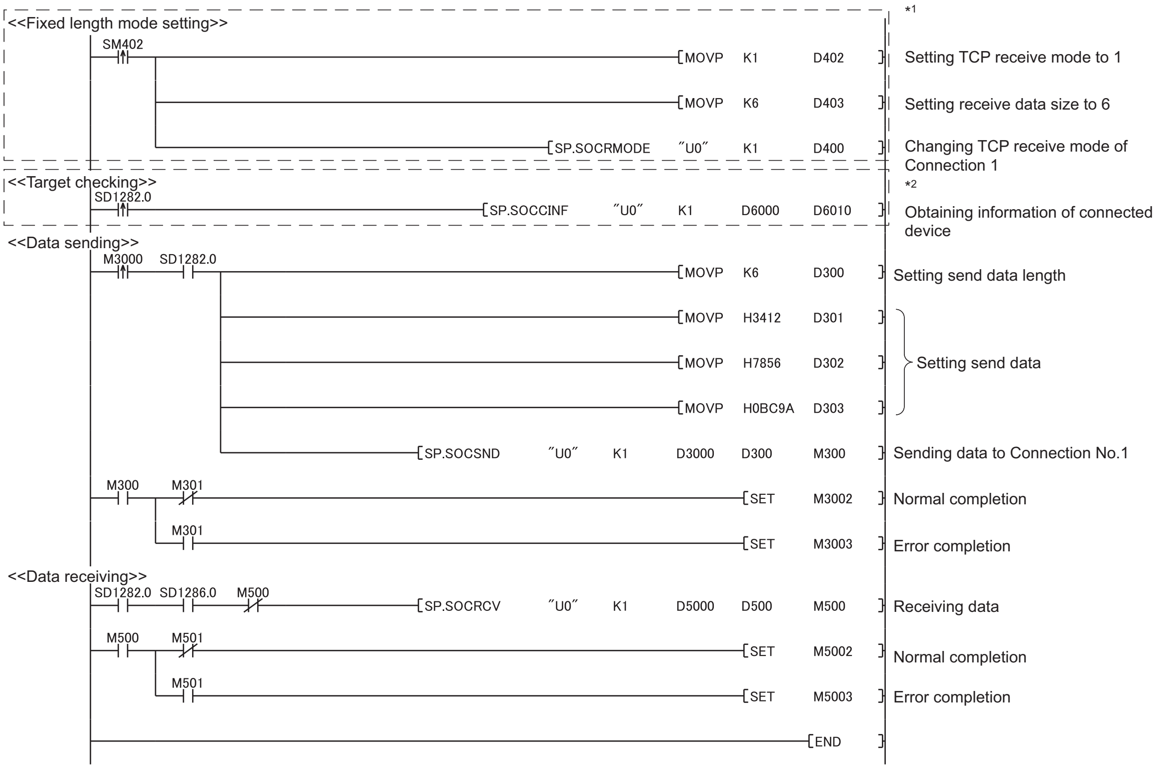
*1 Do not specify 1388_H to 1391_H (5000 to 5009) because these ports are used by the system. (☞ Page 221, Appendix 2)

(b) Devices used in programs

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 connection information

(c) Sample program



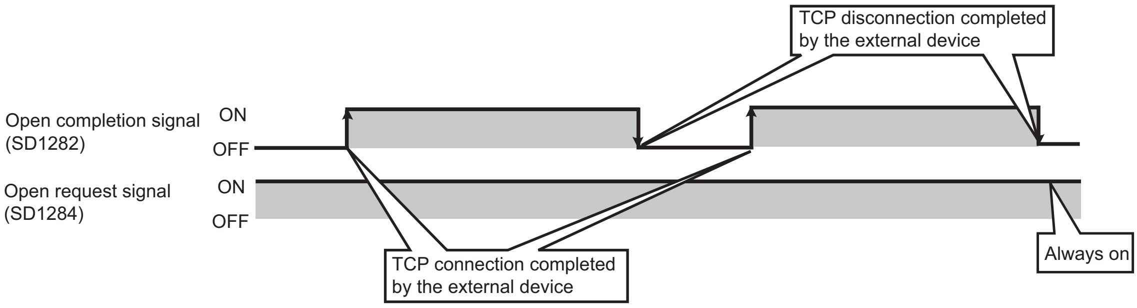
*1 There are two kinds of TCP receive modes: TCP standard receive mode and TCP fixed-length receive mode.
For fixing the data size, run the program enclosed by a dotted line.
(It can be omitted when the data size is not fixed.)

For the TCP receive mode, refer to the section for the SP.SOCRMODE instruction. (☞ Page 111, 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.


TCP disconnection completed by the external device
Open completion signal (SD1282)
ON
OFF
Open request signal (SD1284)
ON
OFF
TCP connection completed by the external device
Always on

- 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 106, 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 86, 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

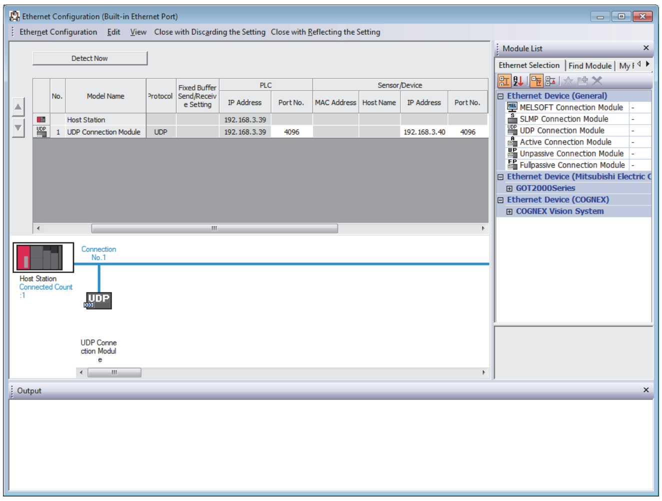
(1) Program example

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] ⇨ [Ethernet Conf.] or [Open Setting] button

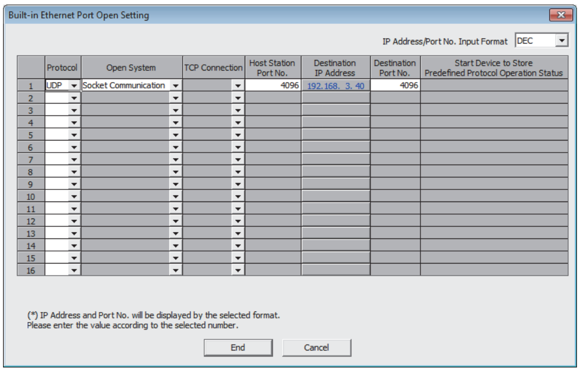


Ethernet Configuration (Built-in Ethernet Port)

Ethernet Configuration Edit View Close with Discarding the Setting Close with Reflecting the Setting

Detect Now

	No.	Model Name	Protocol	Fixed Buffer Send/Receive Setting	PLC		Sensor/Device			
					IP Address	Port No.	MAC Address	Host Name	IP Address	Port No.
		Host Station			192.168.3.39					
UDP	1	UDP Connection Module	UDP		192.168.3.39	4096			192.168.3.40	4096



Built-in Ethernet Port Open Setting

IP Address/Port No. Input Format: DEC

	Protocol	Open System	TCP Connection	Host Station Port No.	Destination IP Address	Destination Port No.	Start Device to Store Predefined Protocol Operation Status
1	UDP	Socket Communication		4096	192.168. 3. 40	4096	
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.

- For "Ethernet Conf.", drag and drop "UDP Connection Module" from "Module List" to the left side on the window. Set the port numbers and IP address as mentioned below.

Item		Description
PLC	Port No.	4096 (Setting range: 1 to 4999, 5010 to 65534) Do not specify 5000 to 5009 because these ports are used by the system. (☞ Page 221, Appendix 2)
Sensor/device	IP Address	192.168.3.40 (Setting range: 0.0.0.1 to 223.255.255.254/255.255.255.255)
	Port No.	4096 (Setting range: 1 to 65534/ 65535)

- For the "Open Setting"

Item	Description
Protocol	UDP
Open System	Socket Communication
TCP Connection	Blank
Host Station Port No.	1000_H (Setting range: 0001_H to 1387_H, 1392_H to $FFFE_H$ (1 to 4999, 5010 to 65534)) *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 to 65534/65535))

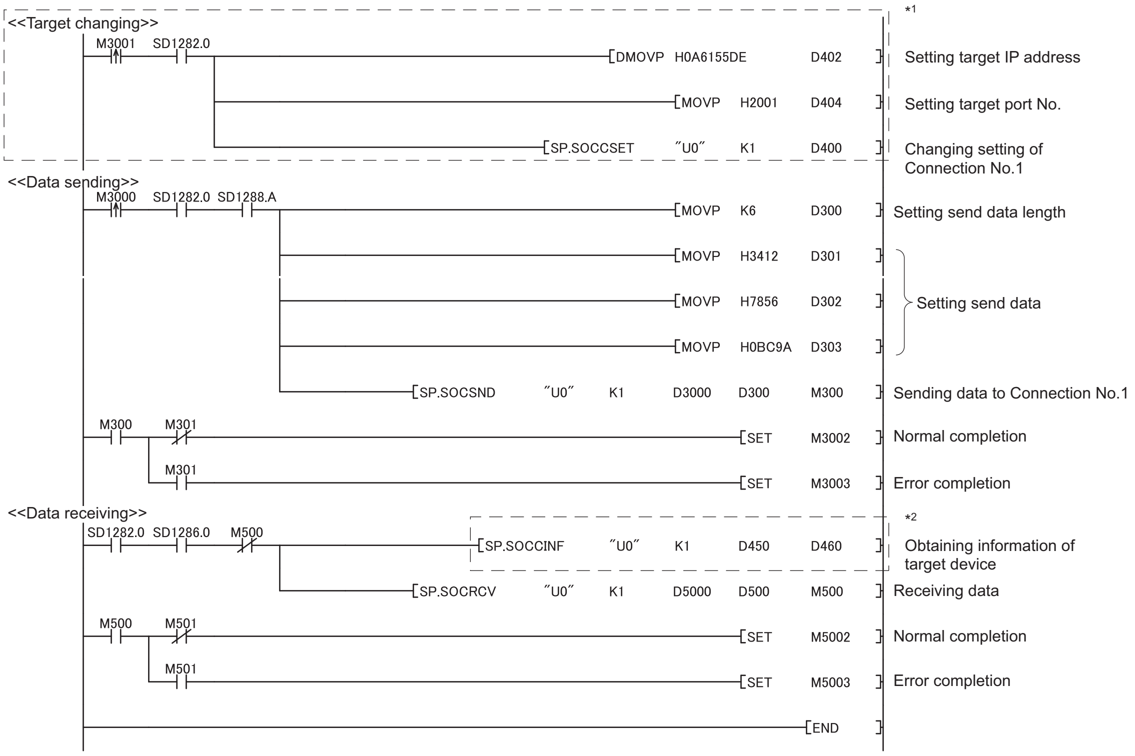
*1 Do not specify 1388_H to 1391_H (5000 to 5009) because these ports are used by the system. (☞ Page 221, Appendix 2)

(b) Devices used in programs

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
D5000 and D5001	SP.SOCRCV instruction control data
M500 and 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 and D451	SP.SOCCINF instruction control data
D460 to D464	SP.SOCCINF instruction connection information

(c) Sample program



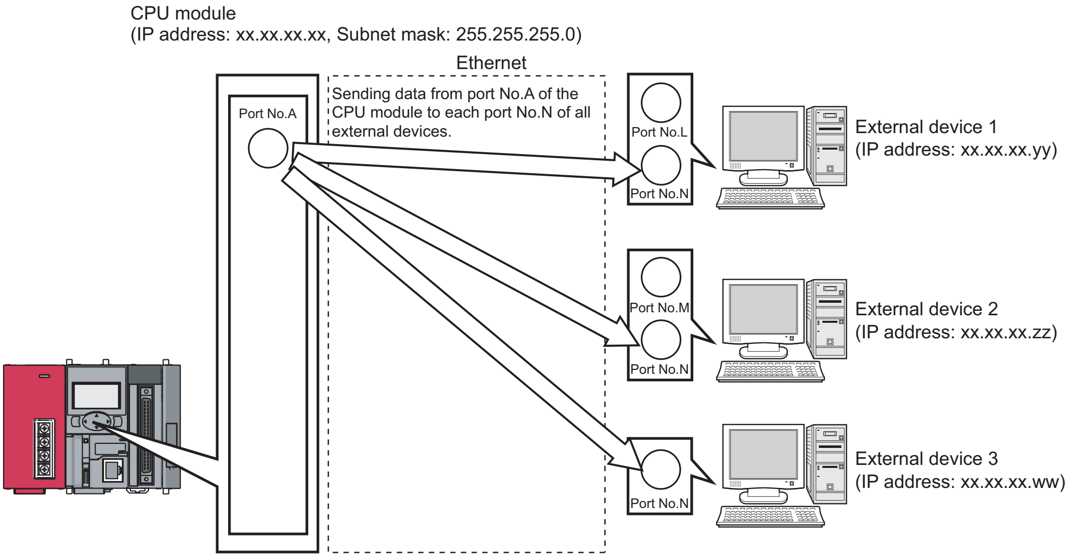
*1 For changing the target device, run the program enclosed by a dotted line. (It can be omitted when the communication target is not changed.) For details, refer to the section of the SP.SOCCSET instruction. (☞ Page 109, Section 7.4.7)

*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.: $FFFF_H$


CPU module
(IP address: xx.xx.xx.xx, Subnet mask: 255.255.255.0)
Ethernet
Port No.A
Sending data from port No.A of the CPU module to each port No.N of all external devices.
Port No.L
Port No.N
External device 1
(IP address: xx.xx.xx.yy)
Port No.M
Port No.N
External device 2
(IP address: xx.xx.xx.zz)
Port No.N
External device 3
(IP address: xx.xx.xx.ww)

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 109, 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 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 86, 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 106, 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 musk pattern is specified, apply the pattern before using the above mentioned IP address.

Example:	CPU module IP address	:	64.	168.	3.	39
	Subnet mask pattern	:	None			
	IP address of simultaneous broadcast	:	64.	255.	255.	255
Example:	CPU module IP address	:	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 221, Appendix 2)
Do not specify 0014_H and 0015_H for the socket communications function when using the file transfer function (FTP server).
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.
Do not specify $B0B5_H$ and $F000_H$ to $FFFE_H$ for the socket communication function when using the iQ Sensor Solution 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.

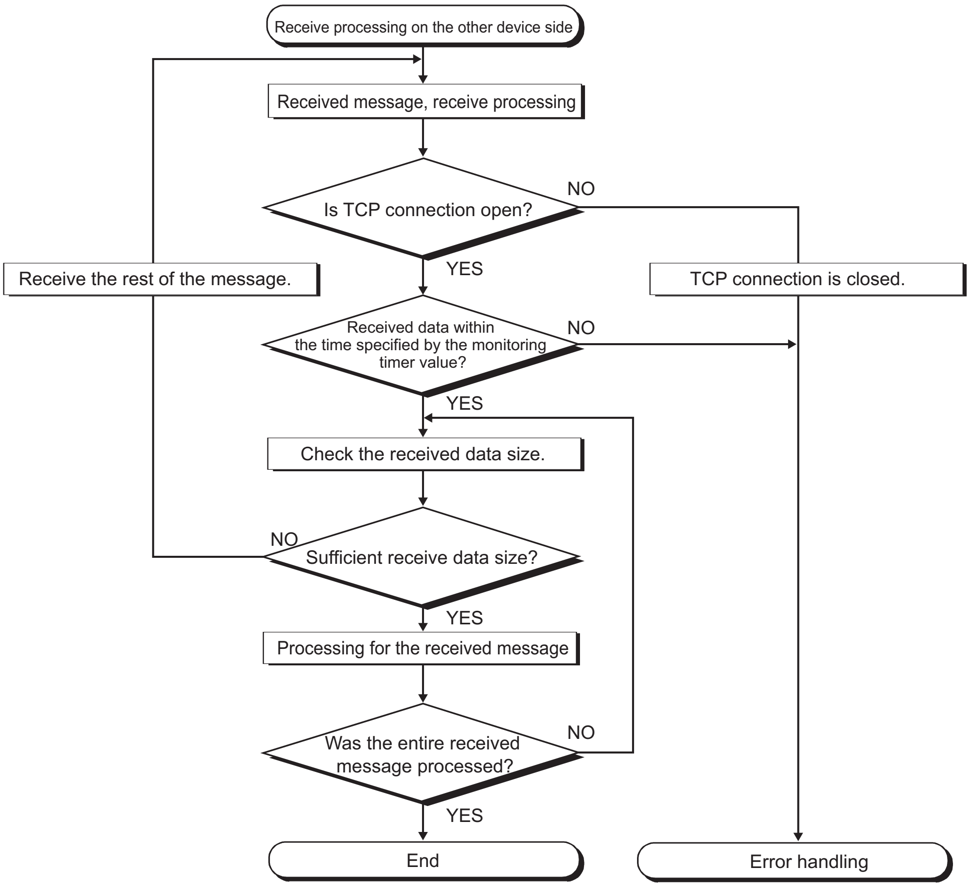
When accessing a file while response time monitoring is performed on the connected device with the socket communication function, add the time required for file access to the monitoring time.

(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 module 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.


Receive processing on the other device side
Received message, receive processing
Is TCP connection open?
NO
YES
Receive the rest of the message.
TCP connection is closed.
Received data within the time specified by the monitoring timer value?
NO
YES
Check the received data size.
NO
Sufficient receive data size?
YES
Processing for the received message
Was the entire received message processed?
NO
YES
End
Error handling

(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 describes the socket communication function instructions.

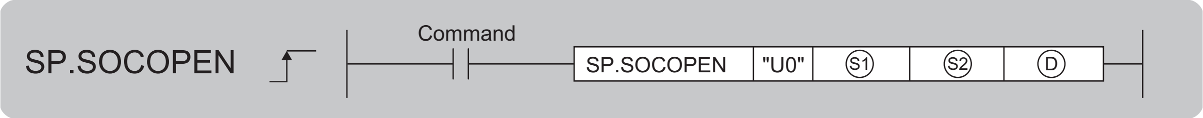
The following is a list of the instructions.

Instruction	Description	Reference
SP.SOCOPEN	Establishes a connection.	Page 86, Section 7.4.1
SP.SOCCLOSE	Closes a connection.	Page 92, Section 7.4.2
SP.SOCRCV	Reads out received data (in END processing).	Page 95, Section 7.4.3
S.SOCRCVS	Reads out received data (upon instruction execution).	Page 99, Section 7.4.4
SP.SOCSND	Sends data.	Page 102, Section 7.4.5
SP.SOCCINF	Reads out connection information.	Page 106, Section 7.4.6
SP.SOCCSET	Changes the target of the connection for UDP/IP communication.	Page 109, Section 7.4.7
SP.SOCRMODE	Changes receive mode of the connection.	Page 111, Section 7.4.8
S(P).SOCRDATA	Reads out data from the socket communication receive data area.	Page 114, Section 7.4.9

Point

- For configuration of data communication using the socket communication function, refer to Page 66, Section 7.1, Page 77, 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.
  MELSEC-Q/L Programming Manual (Common Instruction)

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				
Ⓢ1	—	○	○	—				○	—
Ⓢ2	—	○[*1]	○[*1]	—				—	—
Ⓓ	○[*1]	○ (Except for T, ST, C)	○[*1]	—				—	—

*1 Local devices cannot be used.

(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
Ⓓ	Start number of the device which turns on for one scan upon completion of the instruction ⒹⒹ+1 also turns on when failed.	System	Bit

*1 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.

(2) Control data

Device	Item	Description	Setting range	Set by[*1]
Ⓢ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 "Ethernet Conf." or "Open Setting" of PLC parameter. The control data Ⓢ2+2 to Ⓢ2+9 do not need to be set. 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 normally Other than 0000_H: Failed (Error code)	—	System
Ⓢ2+2	Application setting area	b15 b14 b13 to b11 b10 b9 b8 b7 to b0 Ⓢ2+2 [4] 0 [3] [2] [1] 0 [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	Refer to 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 $FFFE_H$[*3]	
Ⓢ2+4 Ⓢ2+5	Target device IP address[*2]	Specify the IP address of the target device.	1_H to $FFFFFFFF_H$ ($FFFFFFFF_H$: Simultaneous broadcast)	
Ⓢ2+6	Target device port number[*2]	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

*1 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.

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

*3 Because 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, 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 221, Appendix 2)

(3) Function

This instruction opens a connection specified in Ⓢ1.

The set values used for the open processing is selected in Ⓢ2+0.

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

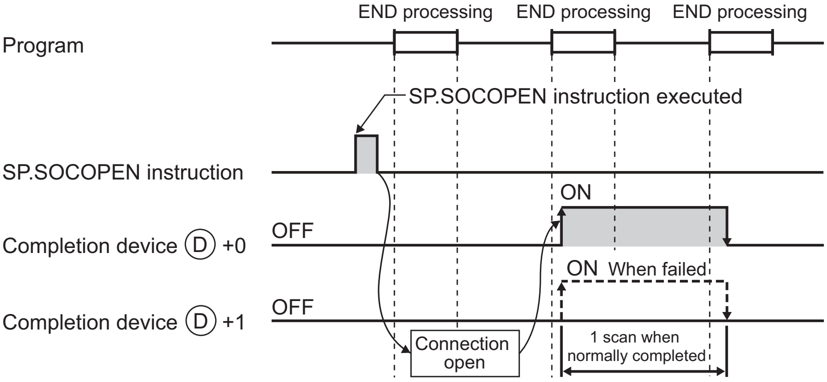
- Completion device Ⓓ+0

This device turns on during END processing of the scan where the SP.SOCOPEN instruction completes, and turns off during the next END processing.

- Completion device Ⓓ+1

This device turns on or off according to the result of the SP.SOCOPEN instruction.

State	Description
When completed normally	Remains off.
When failed	Turns on during END processing of the scan where the SP.SOCOPEN instruction completes, and turns off during the next END processing.



- A connection with no parameters (no protocol is specified) can be opened. In this case, specify 8000_H for Ⓢ2+0 and configure open settings in Ⓢ2+2 to Ⓢ2+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 connection number specified for Ⓢ1 is other than 1 to 16.

(Error code: 4101)

- The device numbers specified for Ⓢ2 and Ⓓ exceed the device point range.

(Error code: 4101)

- An invalid device is specified.

(Error code: 4004)

(5) Program examples

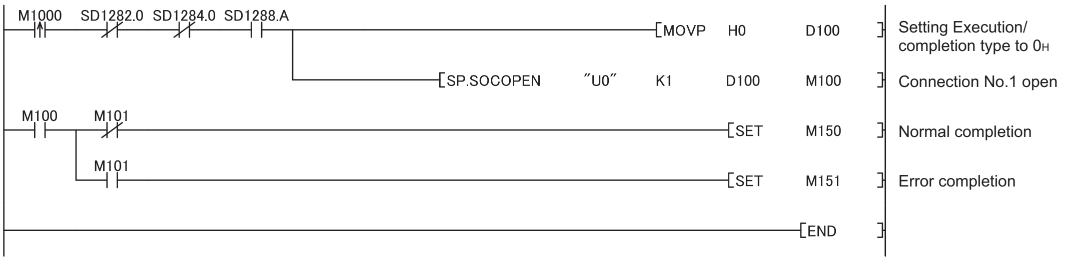
(a) Opening a connection using parameter settings

When M1000 is turned on, connection No.1 is opened using the parameters set in "Ethernet Conf." or "Open Setting" 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



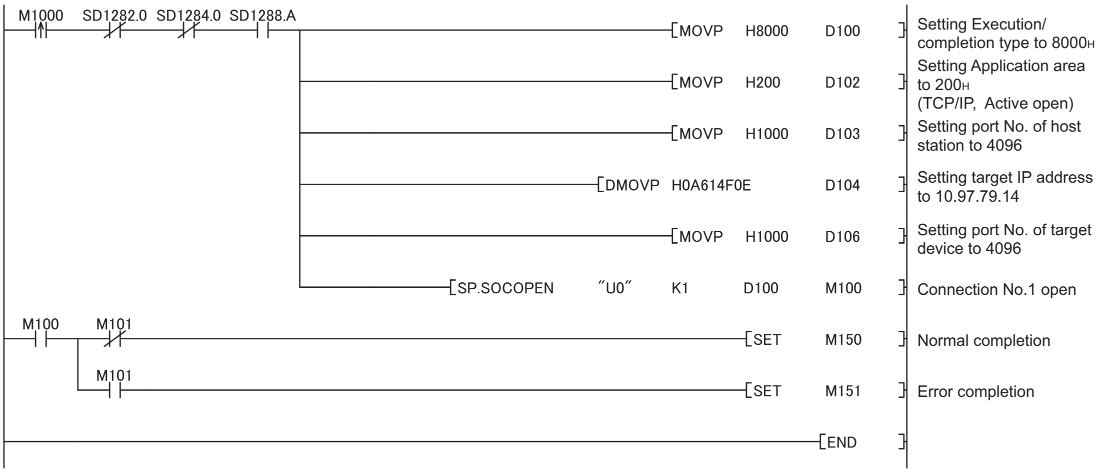
(b) Opening a connection using control data settings

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

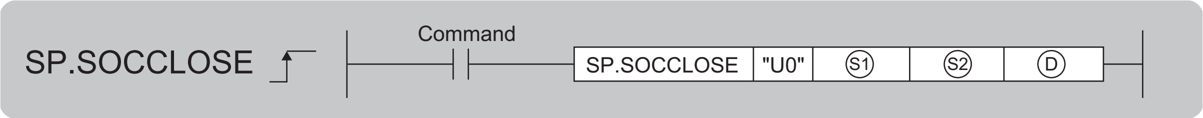
- Devices used

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


M1000
SD1282.0
SD1284.0
SD1288.A
[MOVP H8000 D100]
Setting Execution/ completion type to 8000H
[MOVP H200 D102]
Setting Application area to 200H (TCP/IP, Active open)
[MOVP H1000 D103]
Setting port No. of host station to 4096
[DMOVP H0A614F0E D104]
Setting target IP address to 10.97.79.14
[MOVP H1000 D106]
Setting port No. of target device to 4096
[SP.SOCOPEN "U0" K1 D100 M100]
Connection No.1 open
M100
M101
[SET M150]
Normal completion
M101
[SET M151]
Error completion
[END]

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	○ (Except for T, ST, C)	○*1	—				—	—

*1 Local devices cannot be used.

(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 which turns on for one scan upon completion of the instruction ⒹⒹ+1 also turns on when failed.		Bit

*1 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.

(2) Control data

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

*1 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 Ⓢ1. (Disconnection of a connection)

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

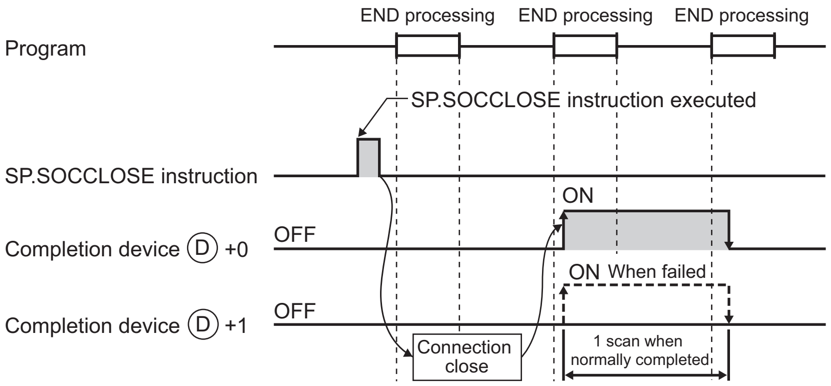
- Completion device Ⓓ+0

This device turns on during END processing of the scan where the SP.SOCCLOSE instruction completes, and turns off during the next END processing.

- Completion device Ⓓ+1

This device turns on or off according to the result of the SP.SOCCLOSE instruction.

State	Description
When completed normally	Remains off.
When failed	Turns on during END processing of the scan where the SP.SOCCLOSE instruction completes, and turns off during 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 connection number specified for Ⓢ1 is other than 1 to 16.

(Error code: 4101)

- The device numbers specified for Ⓢ2 and Ⓓ 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.

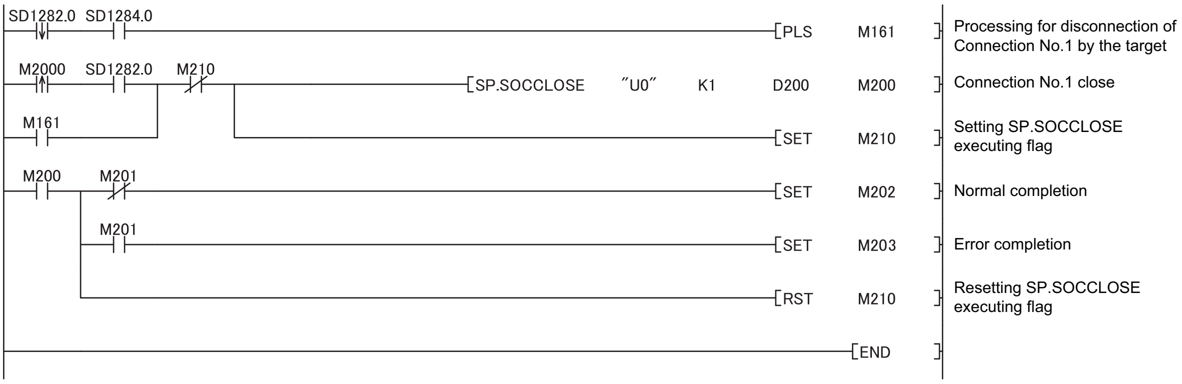
(5) Program example

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

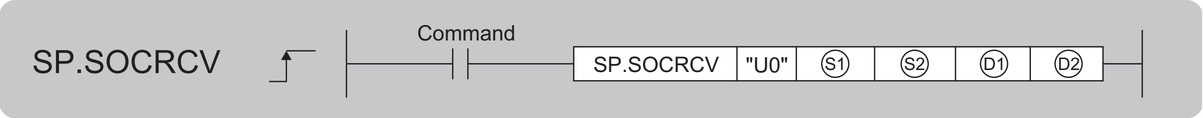
- Devices used

Device number	Application
SD1282	Open completion signal
SD1284	Open request signal
D200	SP.SOCCLOSE instruction control data
M200	SP.SOCCLOSE 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]	○ (Except for T, ST, C)	○[*1]	—				—	—

*1 Local devices cannot be used.

(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 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 D2+1 also turns on when failed.		Bit

*1 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.

(2) Control data

Device	Item	Description	Setting range	Set by[*1]
S2+0	System area	—	—	—
S2+1	Completion status	Completion status is stored 0000_H: Completed normally Other than 0000_H: Failed (Error code)	—	System
D1+0	Receive data length	The length of the data which were read out from the Socket communication receive data area is stored. (Number of bytes)	0 to 2046	
D1+1 to D1+n	Receive data	The data which were read out from the Socket communication receive data area are stored in order.	—	

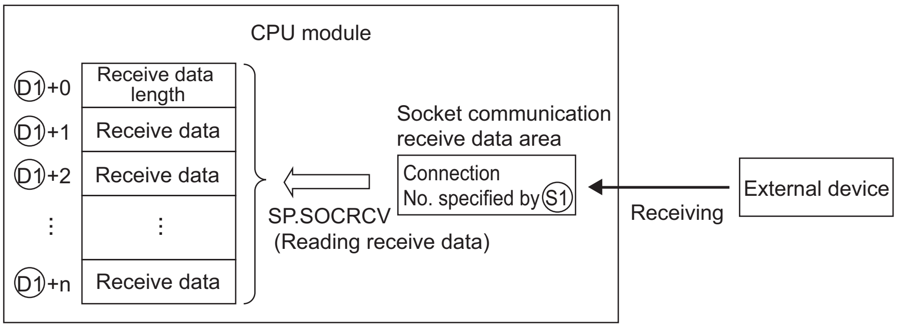
[*1] The "Set by" column indicates the following.
System: The CPU module stores the execution result of the SP.SOCRCV instruction.

Point

- 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

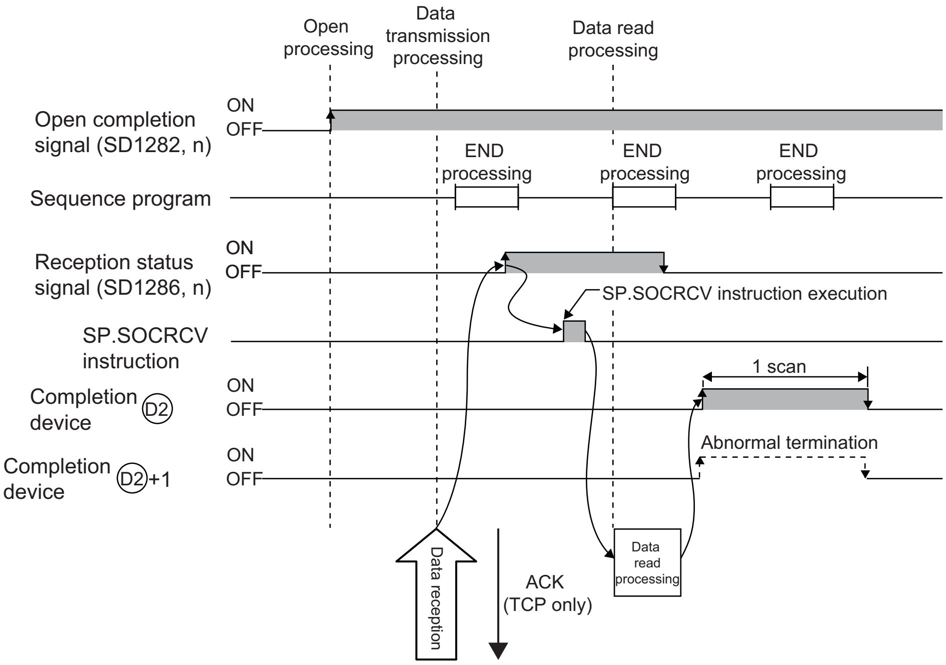
This device turns on during END processing of the scan where the SP.SOCRCV instruction completes, and turns off during the next END processing.

- Completion device D2+1

This device turns on or off according to the result of the SP.SOCRCV instruction.

State	Description
When completed normally	Remains off.
When failed	Turns on during END processing of the scan where the SP.SOCRCV instruction completes, and turns off during 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 connection number specified for Ⓢ1 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 Ⓢ2, Ⓓ1, and Ⓓ2 exceed the device point range.
  (Error code: 4101)
- An invalid device is specified.
  (Error code: 4004)

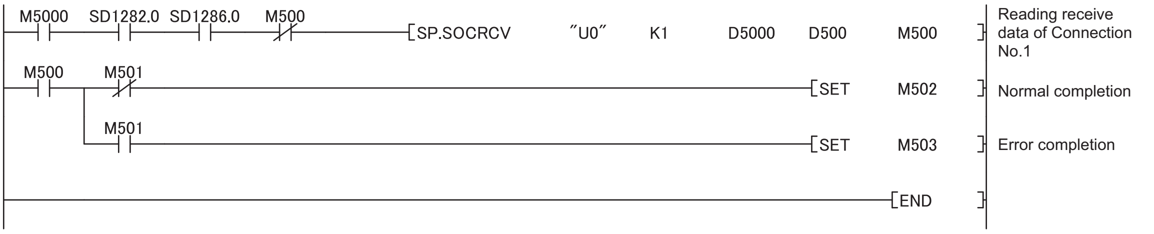
(5) Program example

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

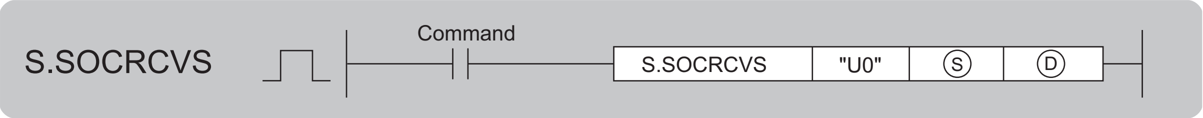
- Program



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 receive 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.

(2) Control data

Device	Item	Description	Setting range	Set by[*1]
Ⓓ+0	Receive data length	The length of the data which were read out from the Socket communication receive data area is stored. (Number of bytes)	0 to 2046	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.	—	

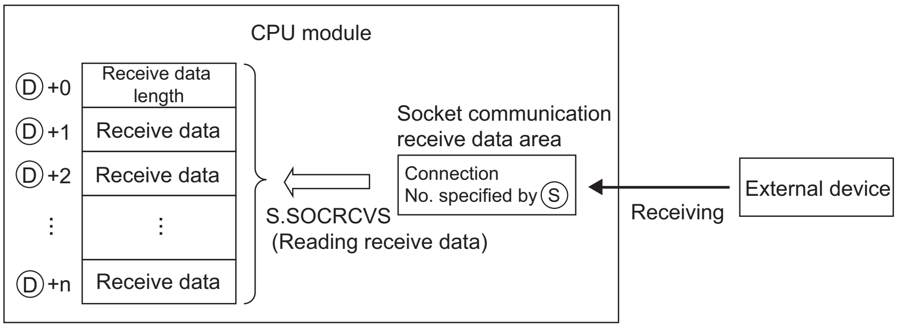
*1 The "Set by" column indicates the following.
System: The CPU module stores the execution result of the S.SOCRCVS instruction.

Point

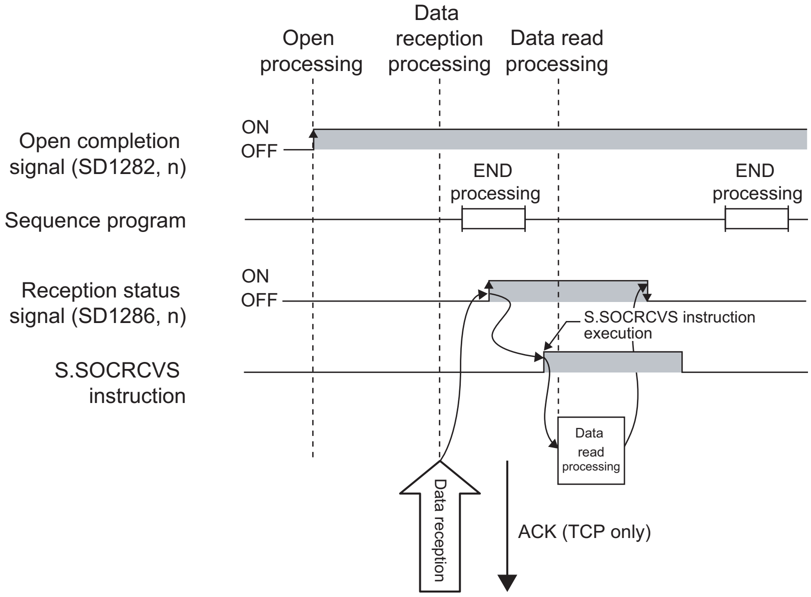
If the CPU module receives odd-byte data, invalid data is stored to the high byte of the device where the last received data is stored.

(3) Function

This instruction reads out received data of the connection specified in Ⓢ 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 connection number specified for Ⓢ 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 Ⓓ 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.

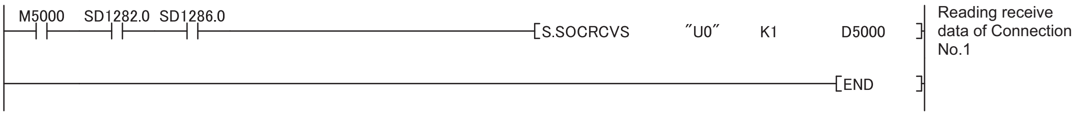
(6) Program example

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

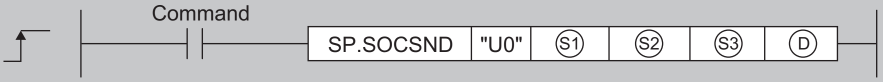


Point

- ● 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)

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]	○ (Except for T, ST, C)	○[*1]	—				—	—

*1 Local devices cannot be used.

(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 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

*1 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.

(2) Control data

Device	Item	Description	Setting range	Set by[*1]
S2+0	System area	—	—	—
S2+1	Completion status	Completion status is stored 0000_H: Completed normally Other than 0000_H: Failed (Error code)	—	System
S3+0	Send data length	The length of send data is specified (in bytes).	1 to 2046	User
S3+1 to S3+n	Send data	Send data are specified.	—	

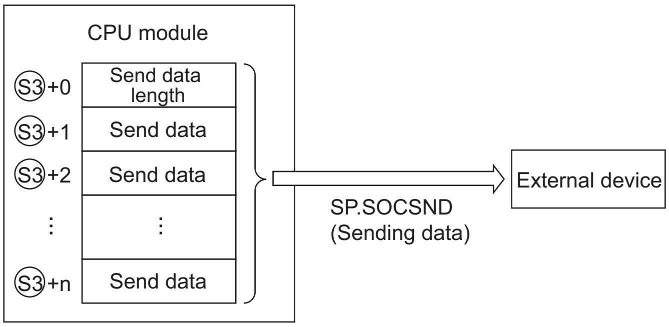
[*1] The "Set by" column indicates the following.
User: The data must be set before executing the SP.SOCSND instruction.
System: The CPU module stores the execution result of the SP.SOCSND instruction.

Point

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 S3 to the external device of the connection specified by S1.



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

- Completion device D+0

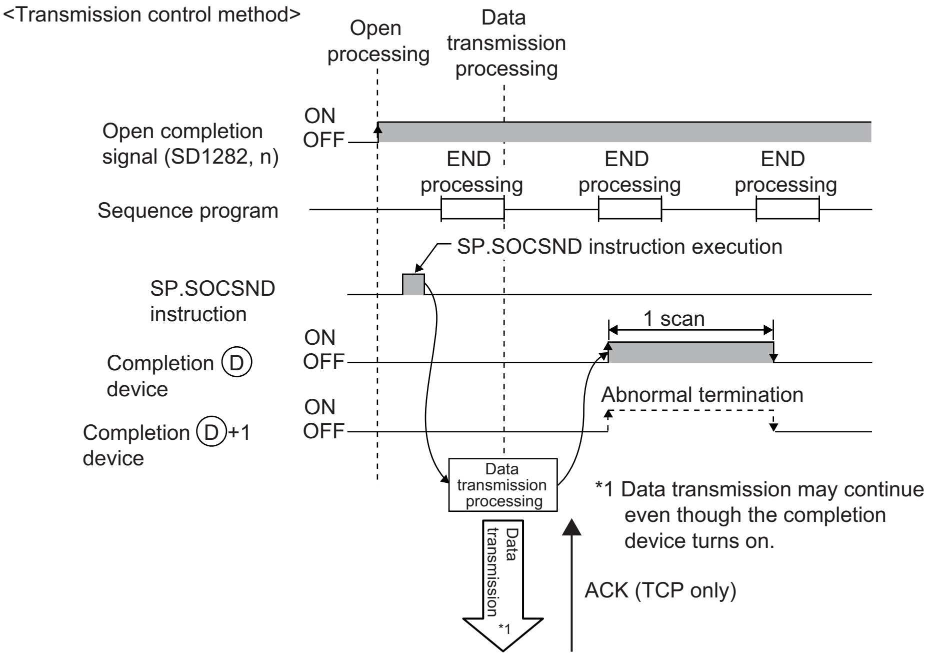
This device turns on during END processing of the scan where the SP.SOCSND instruction completes, and turns off during the next END processing.

- Completion device D+1

This device turns on or off according to the result of the SP.SOCSND instruction.

State	Description
When completed normally	Remains off.
When failed	Turns on during END processing of the scan where the SP.SOCSND instruction completes, and turns off during the next END processing.

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


<Transmission control method>
Open processing
Data transmission processing
Open completion signal (SD1282, n)
ON
OFF
END processing
END processing
END processing
Sequence program
SP.SOCSND instruction execution
SP.SOCSND instruction
1 scan
ON
OFF
Completion Ⓓ device
Abnormal termination
ON
OFF
Completion Ⓓ+1 device
Data transmission processing
*1 Data transmission may continue even though the completion device turns on.
Data transmission *1
ACK (TCP only)

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

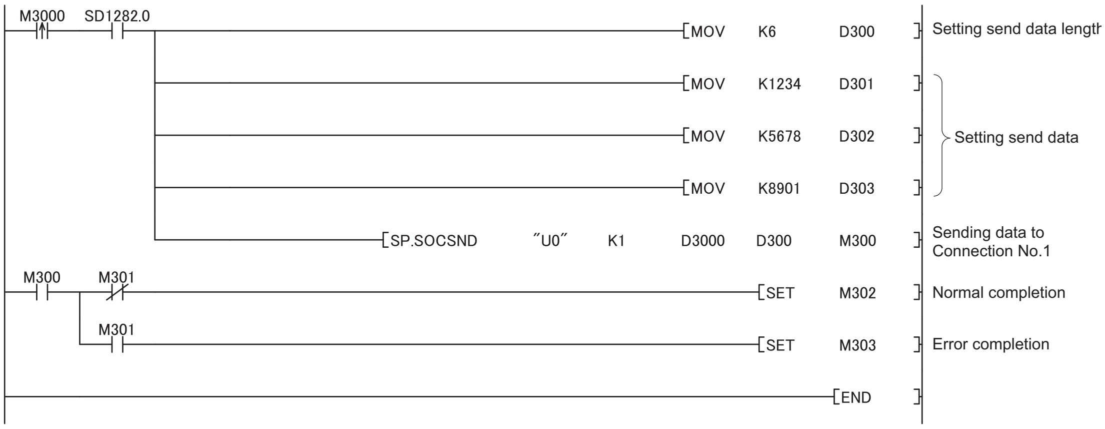
(5) Program example

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

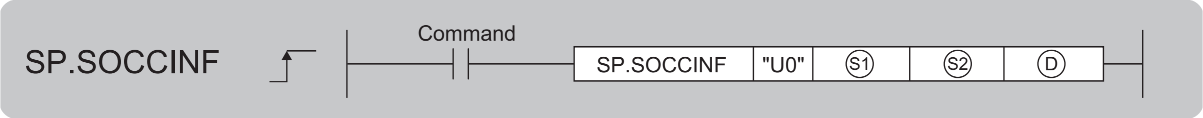
- Devices used

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)



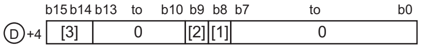
Setting data	Internal device		R, ZR	J□\□		U□\G□	Zn	Constant K, H	Others
	Bit	Word		Bit	Word				
Ⓢ1	—	○	○	—				○	—
Ⓢ2	—	○	○	—				—	—
Ⓓ	—	○	○	—				—	—

(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^{*1}
Ⓢ2+0	System area	—	—	—
Ⓢ2+1	Completion status	Completion status is stored 0000_H: Completed normally 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 $FFFFFFFF_H$^{*3*4}	
Ⓓ+2	Target device port number	Port number of the target device is stored.	1_H to $FFFF_H$^{*3*5}	
Ⓓ+3	Host station port number	Port number of the host station is stored.	1_H to 1387_H 1392_H to $FFFE_H$^{*2*3}	
Ⓓ+4	Application setting area	 [1] Communication method (protocol) 0: TCP/IP 1: UDP/IP [2] Socket communication procedure 1: Nonprocedural (fixed) [3] Open system 00: Active open or UDP/IP 10: Unpassive open 11: Fullpassive open	*3	

*1 The "Set by" column indicates the following.
System: The CPU module stores the execution result of the SP.SOCCINF instruction.

*2 Because 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, 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 221, Appendix 2)

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

*4 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.

*5 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 Ⓢ1.

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

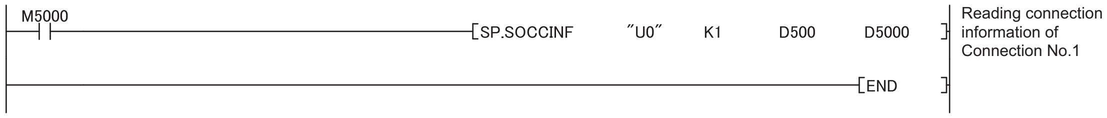
(5) Program example

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

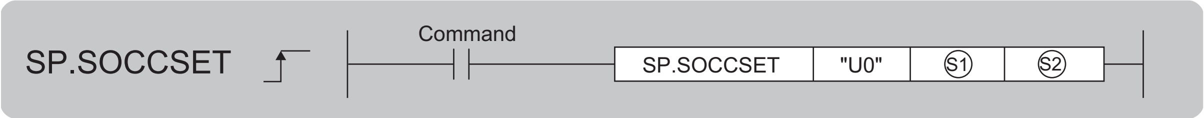
- Devices used

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

- Program


M5000
SP.SOCCINF "U0" K1 D500 D5000
Reading connection information of Connection No.1
END

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				
Ⓢ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.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[*1]
Ⓢ2+0	System area	—	—	—
Ⓢ2+1	Completion status	Completion status is stored 0000_H: Completed normally Other than 0000_H: Failed (Error code)	—	System
Ⓢ2+2 Ⓢ2+3	Target device IP address	Specify the IP address of the target device.	1_H to $FFFFFFFF_H$ ($FFFFFFFF_H$: Simultaneous broadcast)	User
Ⓢ2+4	Target device port number	Specify the port number of the target device.	1_H to $FFFF_H$ ($FFFF_H$: Simultaneous broadcast)	

[*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.

(3) Function

This instruction changes the IP address and port number of the target device of the connection 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.
- ● After the SP.SOCCSET instruction is executed, data communications can be performed with the changed target device.

(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 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.

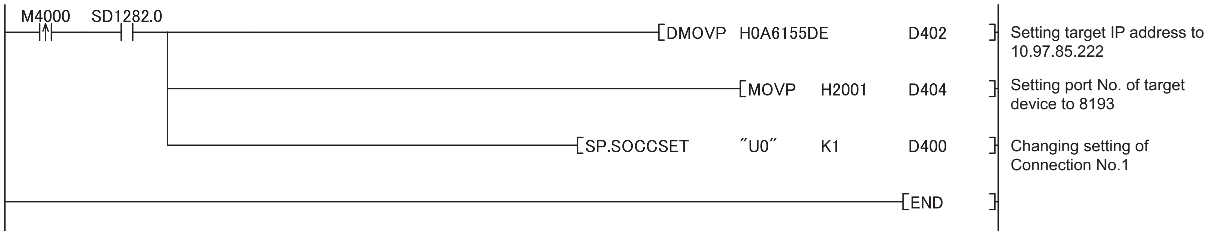
(6) Program example

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.

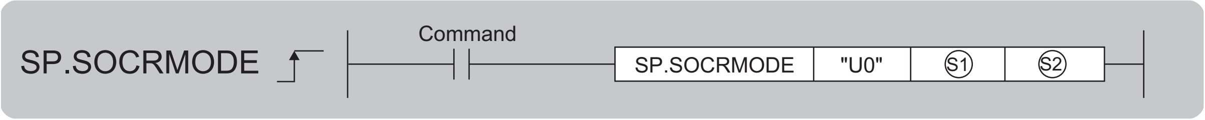
- Devices used

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

- Program


M4000
SD1282.0
DMOVP H0A6155DE D402
Setting target IP address to 10.97.85.222
MOVP H2001 D404
Setting port No. of target device to 8193
SP.SOCCSET ″U0″ K1 D400
Changing setting of Connection No.1
END

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[*2]
Ⓢ2+0	System area	—	—	—
Ⓢ2+1	Completion status	Completion status is stored 0000_H: Completed normally Other than 0000_H: Failed (Error code)	—	System
Ⓢ2+2	TCP receive mode[*1]	TCP receive mode is stored. 0: TCP standard receive mode 1: TCP fixed-length receive mode	0, 1	User
Ⓢ2+3	Receive data size	Receive data size in the socket communication is stored (in bytes).	1 to 2046	

*1 Invalid for connections in UDP communications.
*2 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.

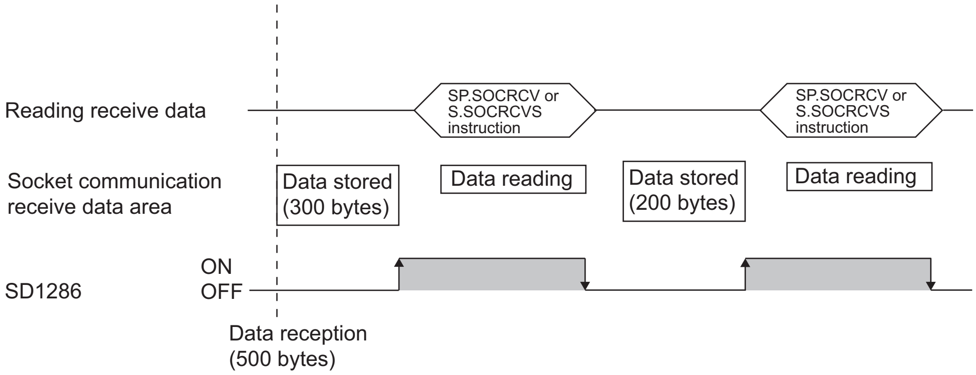
(3) Function

This instruction changes the TCP receive mode and receive data size of the connection specified in Ⓢ1. The mode is changed as specified in Ⓢ2+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.


Reading receive data
SP.SOCRCV or S.SOCRCVS instruction
SP.SOCRCV or S.SOCRCVS instruction
Socket communication receive data area
Data stored (300 bytes)
Data reading
Data stored (200 bytes)
Data reading
SD1286
ON
OFF
Data reception (500 bytes)

(b) TCP fixed-length receive mode

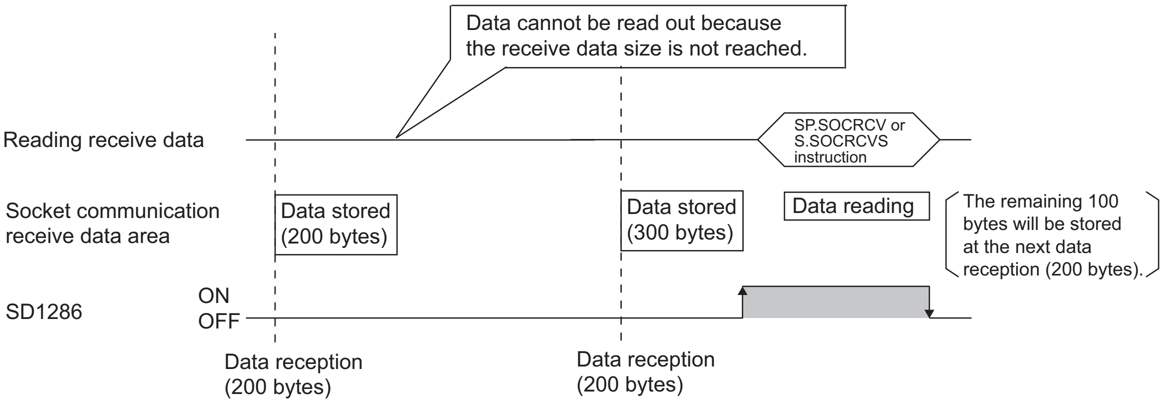
When data are received, they are stored in the Socket communication receive data area, and SD1286 (Receive state signal) turns on.

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.


Data cannot be read out because the receive data size is not reached.
Reading receive data
SP.SOCRCV or S.SOCRCVS instruction
Socket communication receive data area
Data stored (200 bytes)
Data stored (300 bytes)
Data reading
The remaining 100 bytes will be stored at the next data reception (200 bytes).
SD1286
ON
OFF
Data reception (200 bytes)
Data reception (200 bytes)

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 be enabled at the following timing of the SP.SOCRMODE instruction execution.

- Before opening: Values will be enabled after opening a connection.
- When there are data in the Socket communication receive data area: Values will be enabled 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 will be enabled 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 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)

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.

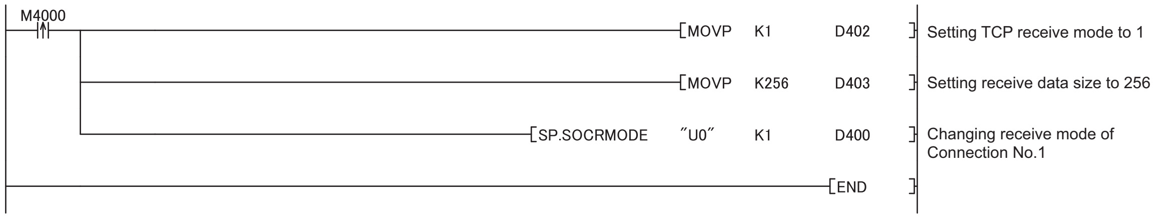
(5) Program example

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.

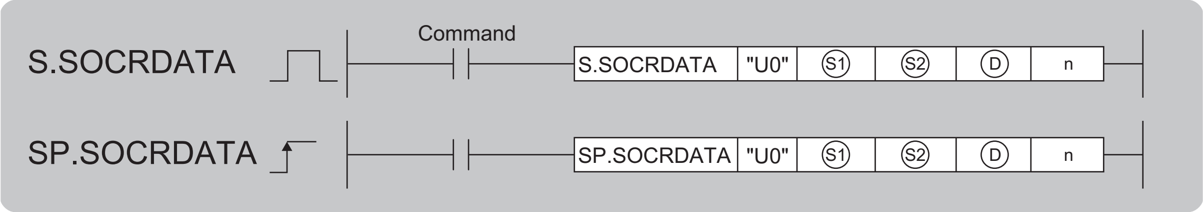
- Devices used

Device number	Application
D400	SP.SOCRMODE instruction control data

- Program


M4000
[MOVP K1 D402]
Setting TCP receive mode to 1
[MOVP K256 D403]
Setting receive data size to 256
[SP.SOCRMODE "U0" K1 D400]
Changing receive mode of Connection No.1
[END]

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 1024 words)		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) Control data

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

[*1] 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 the connection specified in Ⓢ1, and stores them in the device specified in Ⓓ 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 connection number specified for Ⓢ1 is other than 1 to 16.
  (Error code: 4101)
- The device numbers specified for Ⓢ2, Ⓓ, and n exceed the device point range.
  (Error code: 4101)
- An invalid device is specified.
  (Error code: 4004)
- When the value of the device designated by "n" exceeds 1024
  (Error code: 4100)

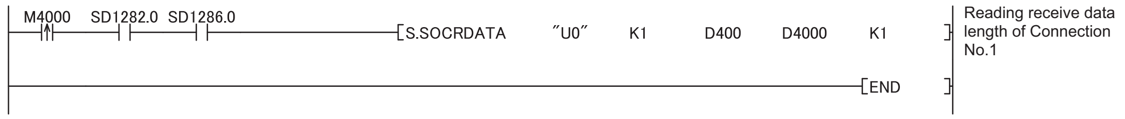
(5) Program example

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



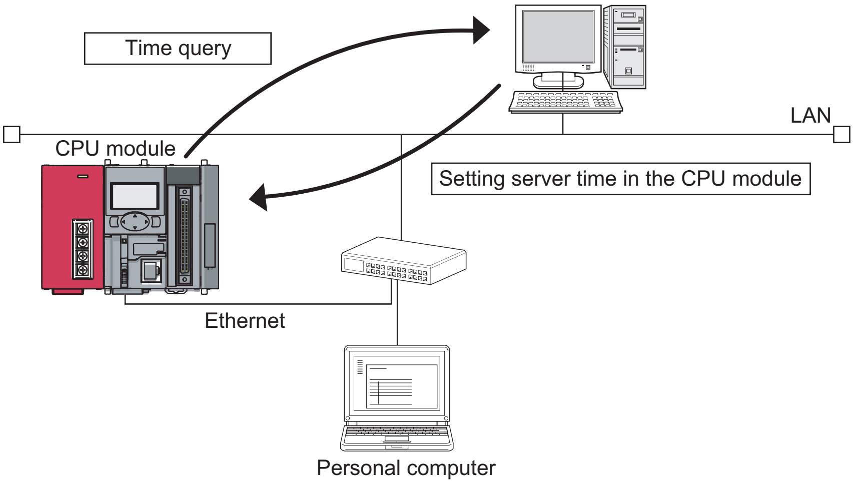
CHAPTER 8 TIME SETTING FUNCTION (SNTP CLIENT)

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 performed at the following timing.

- When the programmable controller is powered off and on, or the CPU module 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.

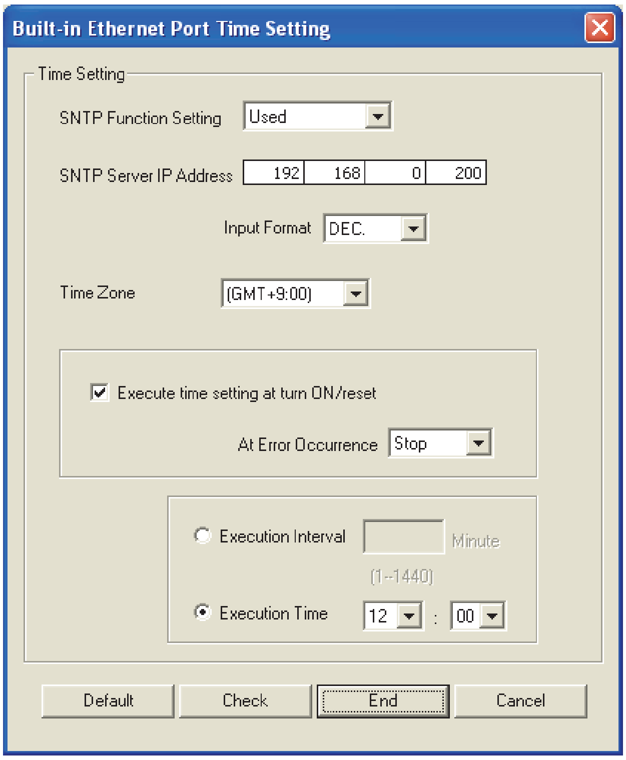
Remark

Access through routers is also available. When configuring the settings for it, set the subnet mask pattern and default router IP address. (☞ Page 28, Section 3.4)

8.1 Setting Method

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

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



Built-in Ethernet Port Time Setting

Time Setting

SNTP Function Setting: Used

SNTP Server IP Address: 192 . 168 . 0 . 200

Input Format: DEC.

Time Zone: (GMT+9:00)

☑ Execute time setting at turn ON/reset

At Error Occurrence: Stop

○ Execution Interval [ ] Minute (1--1440)

● Execution Time 12 : 00

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*1	Select this when executing the time setting function at fixed time intervals.	1 to 1440 (min.)
Execution Time*1	Select this when executing the time setting function at a specified time. (in increments of 30 minutes)	00:00 to 23:30

*1 Either of these two options must be selected.

8.2 Precautions

(1) Communication timeout

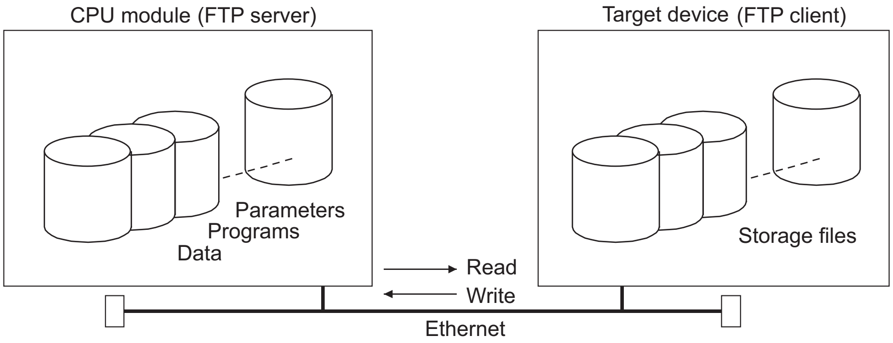
A communication timeout occurs when 20 seconds have elapsed without receiving any response after sending a time query. At the time of a communication timeout, the value in SD1270 is FFFF_H.

(2) 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.

CHAPTER 9 FILE TRANSFER FUNCTION (FTP SERVER)

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.

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

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

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

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

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

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

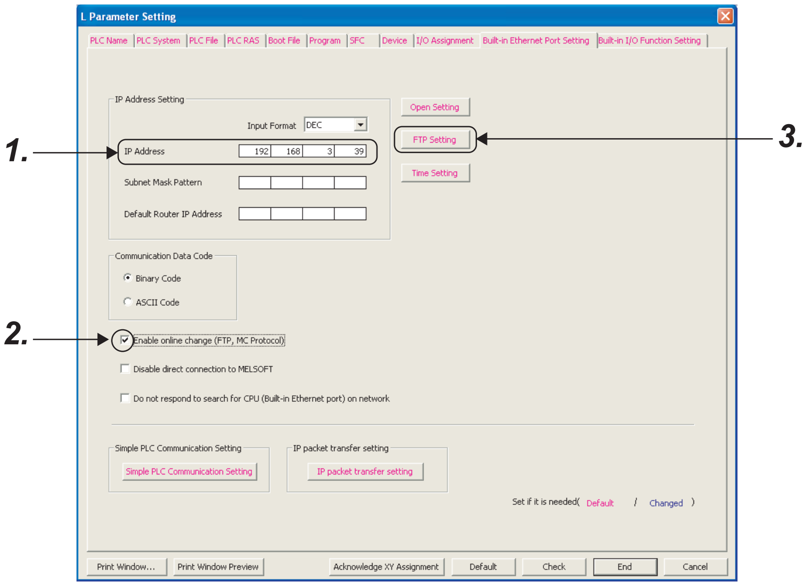
Remark

Access via routers is also possible. For the access, set the subnet mask pattern and the default router IP address. (☞ Page 28, 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]



L Parameter Setting

IP Address Setting

Input Format: DEC

IP Address: 192 . 168 . 3 . 39

Subnet Mask Pattern

Default Router IP Address

Communication Data Code

● Binary Code

○ ASCII Code

☑ Enable online change (FTP, MC Protocol)

□ Disable direct connection to MELSOFT

□ Do not respond to search for CPU (Built-in Ethernet port) on network

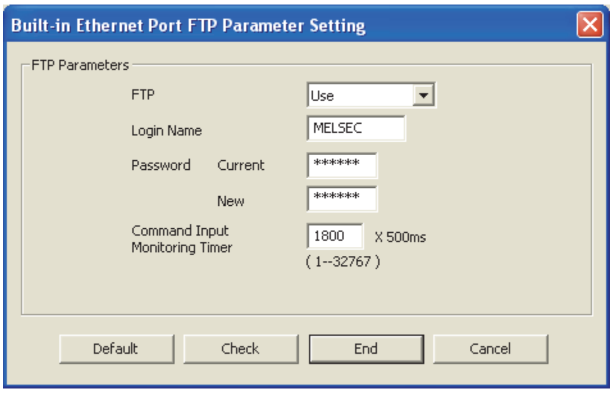
Set if it is needed(Default / Changed)

1.
2.
3.
PLC Name
PLC System
PLC File
PLC RAS
Boot File
Program
SFC
Device
I/O Assignment
Built-in Ethernet Port Setting
Built-in I/O Function Setting
Open Setting
FTP Setting
Time Setting
Simple PLC Communication Setting
Simple PLC Communication Setting
IP packet transfer setting
IP packet transfer setting
Print Window...
Print Window Preview
Acknowledge XY Assignment
Default
Check
End
Cancel

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.**



Built-in Ethernet Port FTP Parameter Setting

FTP Parameters

FTP: Use

Login Name: MELSEC

Password Current: ******

New: ******

Command Input Monitoring Timer: 1800 X 500ms (1--32767)

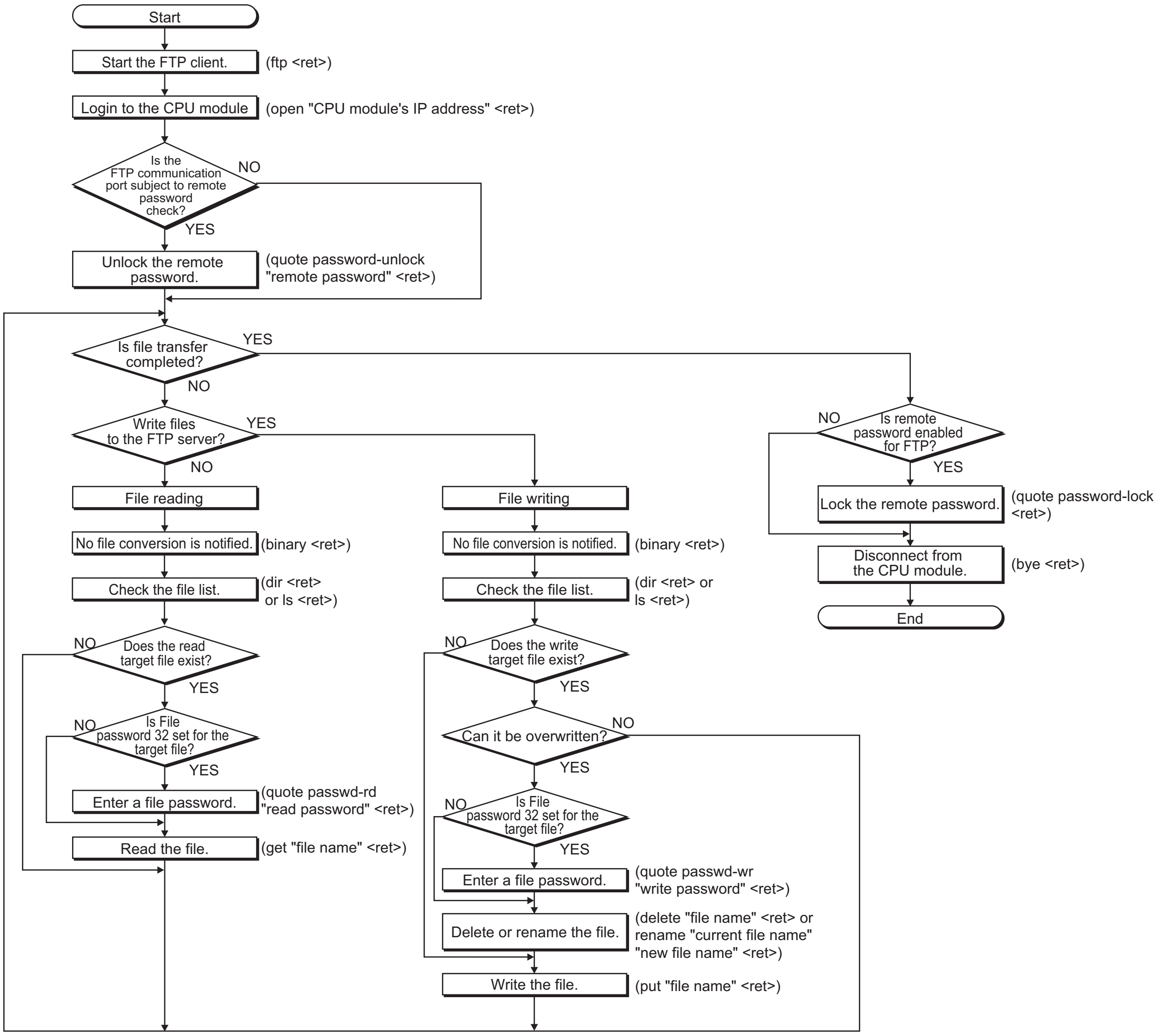
Item	Description
FTP	Select "Use".
Login Name	Set a log-in name used for file transfer (login) request from an external device. The default value is "MELSEC".
Password	Set an FTP password used for file transfer request from an external device. To change the password, enter both the current password and a new password for confirmation. The default value is "MELSEC".
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.

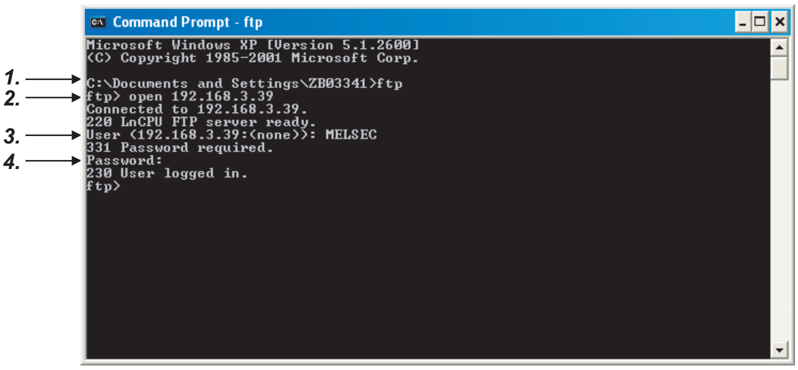
(<ret> indicates an input of the CR, Enter, or Return key.)


Start
Start the FTP client.
(ftp <ret>)
Login to the CPU module
(open "CPU module's IP address" <ret>)
Is the FTP communication port subject to remote password check?
NO
YES
Unlock the remote password.
(quote password-unlock "remote password" <ret>)
Is file transfer completed?
YES
NO
Write files to the FTP server?
YES
NO
File reading
No file conversion is notified.
(binary <ret>)
Check the file list.
(dir <ret> or ls <ret>)
Does the read target file exist?
NO
YES
Is File password 32 set for the target file?
NO
YES
Enter a file password.
(quote passwd-rd "read password" <ret>)
Read the file.
(get "file name" <ret>)
File writing
No file conversion is notified.
(binary <ret>)
Check the file list.
(dir <ret> or ls <ret>)
Does the write target file exist?
NO
YES
Can it be overwritten?
NO
YES
Is File password 32 set for the target file?
NO
YES
Enter a file password.
(quote passwd-wr "write password" <ret>)
Delete or rename the file.
(delete "file name" <ret> or rename "current file name" "new file name" <ret>)
Write the file.
(put "file name" <ret>)
Is remote password enabled for FTP?
NO
YES
Lock the remote password.
(quote password-lock <ret>)
Disconnect from the CPU module.
(bye <ret>)
End

(a) Logging in to the CPU module

The following describes 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®.



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\ZB03341>ftp
ftp> open 192.168.3.39
Connected to 192.168.3.39.
220 LnCPU FTP server ready.
User (192.168.3.39:(none)): MELSEC
331 Password required.
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>)**

(b) 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.

(c) Entering file password 32

When file password 32 (write or read password) is set for the target file, the password must be entered with the following command before accessing the file.

- Write password (quote passwd-wr "write password" <ret>)
- Read password (quote passwd-rd "read password" <ret>)

9.2 Files That Can Be Transferred 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]	SD memory card[*4]	File name and extension (Any name can be specified for ***.)
	Drive 0	Drive 3	Drive 4	Drive 2	
Parameter	○	×	×	×	PARAM.QPA
Intelligent function module parameter	○	×	×	×	IPARAM.QPA
Program[*7]	○	×	×	×	***.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.QBK
Backup data file	×	×	×	×	MEMBKUP0.QBP
Data logging setting file	×	×	×	×	LOGCOM.QLG, LOG01.QLG to LOG10.QLG
Data logging file	×	×	×	○	***.CSV
Menu definition file	×	×	×	×	MENUDEF.QDF
Predefined protocol setting file	×	×	×	×	ECPRTCL.QPT, CPRTCL.QPT
System file for project data batch save/load	×	×	×	○	SVLDINF.QSL
Device data file	×	×	×	○	DEVDATA.QDT
Backup file for sampling trace setting	×	×	×	×	TRCDAT00.QBK
Data logging information backup for power failure	×	×	×	×	LOGDAT00.QBK
System file for the iQ Sensor Solution function (data backup/restoration)	×	×	×	○	SSBRINF.QSI
Backup data file for the iQ Sensor Solution function (data backup/restoration)	×	×	×	○	—[*6]

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

📖 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 127, 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 File names are SRCINF1M.C32, SRCINF2M.C32 for Simple projects (with labels), and SRCINF1I.C32, SRCINF2I.C32 for Structured projects.

*6 The file name depends on the connection type of the data backup/restoration. (📖 iQ Sensor Solution Reference Manual)

*7 When the block password for which "Execution Program Protection Setting" is enabled is set, files cannot be transferred by using the command "get", "mget", or "rename".

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 in RUN state, ×: Not deletable, —: Not writable

File type	Program memory*2*3	Standard RAM	Standard ROM*4	SD memory card*4	File name and extension (Any name can be specified for ***.)
	Drive 0	Drive 3	Drive 4	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.QBK
Backup data file	—	—	—	○	MEMBKUP0.QBP
Data logging setting file	—	—	○	○	LOGCOM.QLG, LOG01.QLG to LOG10.QLG
Data logging file	—	—	—	○	***.CSV
Menu definition file	—	—	○	○	MENUDEF.QDF
Predefined protocol setting file	—	—	○	○	ECPRTCL.QPT, CPRTCL.QPT
System file for project data batch save/load	—	—	—	○	SVLDINF.QSL
Device data file	—	—	—	○	DEVDATA.QDT
Backup file for sampling trace setting	—	—	—	○	TRCDAT00.QBK
Data logging information backup for power failure	—	—	—	○	LOGDAT00.QBK
System file for the iQ Sensor Solution function (data backup/restoration)	—	—	—	○	SSBRINF.QSI
Backup data file for the iQ Sensor Solution function (data backup/restoration)	—	—	—	○	—*6

*1 The SP.FWRITE or SP.FREAD instruction in the program will write or read the file.
For details, refer to the following.
📖 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 127, 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 File names are SRCINF1M.C32, SRCINF2M.C32 for Simple projects (with labels), and SRCINF1I.C32, SRCINF2I.C32 for Structured projects.

*6 The file name depends on the connection type of the data backup/restoration. (📖 iQ Sensor Solution Reference Manual)

Point

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 status			Remote password	
		STOP	RUN		Unlocked*2	Locked*2
			Write enabled*1	Write disabled*1		
binary*5	Notifies that the file will be transferred without conversion.	○	○	○	○	×
bye	Closes and ends the connection with the FTP server.	○	○	○	○	○
close	Closes the connection with the FTP server.	○	○	○	○	○
delete	Deletes a file in the CPU module.*4	○	○	×	○	×
dir	Displays file information of the CPU module.*4	○	○	○	○	×
get	Reads a file from the CPU module.*4	○	○	○	○	×
ls	Displays the names of the files stored in the CPU module.*4	○	○	○	○	×
mdelete	Deletes a file in the CPU module.*4	○	○	×	○	×
mdir	Stores the CPU module file information in the file.*4	○	○	○	○	×
mget	Reads a file from the CPU module.*4	○	○	○	○	×
mls	Stores the CPU module file name in the file.*4	○	○	○	○	×
mput	Writes a file to the CPU module.	○	○	×	○	×
open	Connects to the FTP server.	○	○	○	○	○
put	Writes a file to the CPU module.	○	○	×	○	×
pwd	Displays the current directory of the CPU module.	○	○	○	○	×
quit	Closes and ends the connection with the FTP server.	○	○	○	○	○
quote	Sends an FTP server subcommand.*3	○	○	○	○	○
rename	Changes a CPU module file name.*4	○	○	×	○	×
user	Inputs the user name and password for the CPU module.	○	○	○	○	○

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

*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 details on the remote password, refer to Page 159, CHAPTER 11.

*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" in the "Built-in Ethernet Port Setting" tab of PLC parameter.

(1) Subcommands usable with the "quote" command

○: Available, ×: N/A

Subcommand	Function	CPU module status			Remote password	
		STOP	RUN		Unlocked	Locked
			Write enabled	Write disabled		
change	Displays or changes the attribute of a file stored in the CPU module.[2]	○	○	×	○	×
password-lock	Changes the remote password from the unlock state to the lock state.	○	○	○	○	×[1]
password-unlock	Changes the remote password from the lock state to the unlock state.	○	○	○	○	○
status	Displays the operating status of the CPU module.	○	○	○	○	×
run	Sets the CPU module to the RUN state.	○	○	○	○	×
stop	Sets the CPU module to the STOP state.	○	○	○	○	×
pm-write	Writes the program memory.	○	×	×	○	×
passwd-rd	Sets, displays, or clears the file password 32 (read password).	○	○	○	○	×
passwd-wr	Sets, displays, or clears the 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.

9.4.2 How to specify an FTP command

This section describes 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] The file name can be specified only when specifying a file stored in drive 2.
[*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 memory and drive names, refer to Page 123, 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 MELSEC-L CPU Module User's Manual (Function Explanation/Program Fundamentals).
- Set an extension predetermined by the CPU module. ☞ Page 123, 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 ?).

*	All files with the arbitrary character string (including none) are targeted from the position specified with "*".
?	All files with the arbitrary character string (including none) are targeted from the position specified with "?". ("?" can be used multiple times.)

(Depending on the FTP client, there may be additional restrictions to the characters that can be used for the file name.)

Point

The section enclosed in square brackets [ ] in the specification format can be omitted.

9.4.3 Details of FTP commands

This section describes the FTP commands on the FTP client side supported by the CPU module, and the methods of using those commands.

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 that the file will be transferred without conversion. The return code and kanji codes are also not converted. This command is set automatically in the CPU module.
	Specification format	binary (abbreviation: bin)
bye	Function	Closes the connection with the FTP and quits the FTP.
	Specification format	bye
	Identical command	quit
close	Function	Closes the connection with 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 the SD memory card: delete 2:\MAINSEQ1.USR
	Similar command	mdelete
dir	Function	Displays the file name, date of creation, and volume of files stored in the CPU module.
	Specification format	dir [drive name:\]
	Example	When displaying the detailed information of the files stored in the 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	• 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 the SD memory card: ls 2:\
	Similar command	dir
mdelet	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" (abbreviation: mdel)
	Example	When deleting all the files whose extensions are "QPG" from the program memory: mdelete 0:*.QPG
	Similar command	delete
mdir	Function	Stores the detailed information (file name, date of creation, volume) of the files stored in the CPU module in the FTP client side file as log data.
	Specification format	mdir "source drive name":\"destination file path name"
	Example	When storing the detailed information of the files stored n the SD memory card into the S990901.LOG file: mdir 2:\ S990901.LOG
	Note	• Always specify "\" immediately after the source drive name. • Always specify the source drive name when specifying the destination file path name (FTP client side). • If the destination file path name is not specified, the file is stored with the file name determined by the FTP client's FTP application. • The transfer destination is on the current directory where FTP is started up and connected to the server.
	Similar command	mls
mget	Function	Reads a file from the CPU module. To read multiple files, specify the file name and extension within the file path name using a wild card character (* 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 the SD memory card: mget 2:*.USR
	Note	The file read out is stored 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 name of the file stored in the CPU module in the FTP client side file as log data.
	Specification format	mls "source drive name":\"destination file path name"
	Example	When storing the names of files stored in the SD memory card into the S990901F.LOG file: mls 2:\ S990901F.LOG
	Note	• Always specify "\" immediately after the source drive name. • Always specify the source drive name when specifying the destination file path name (FTP client side). • If the destination file path name is not specified, the file is stored with the file name determined by the FTP client's FTP application. • 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 character (* 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	Specifies the host name or IP address and port number on the FTP server side, and connects with the FTP server.
	Specification format	open "host name" [port number] open "IP address" [port number] • Host name: Host name set with Microsoft® Windows® hosts file • IP address: IP address of the CPU module side • Port number: Port number to be used
	Example 1	When specifying the host name and connecting to the FTP server open HOST
	Example 2	When specifying the IP address and connecting to the FTP server open 192.0.1.254
	Note	The IP address can be specified to create a connection when starting the 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	• If the directory is not specified with 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 stored in the memory in which current parameter files are stored.

pwd	Function	Displays the current directory of the CPU module.
	Specification format	pwd
	Note	"\" is displayed as the execution result of the "pwd" command.
quit	Function	Closes the connection with the FTP and quits the FTP.
	Specification format	quit
	Identical command	bye
quote	Function	Sends an FTP server subcommand (CPU module dedicated subcommand).
	Specification format	quote
	Example	quote password-lock
	Note	Only the CPU module dedicated subcommands can be specified. Refer to (2) in this section.
rename	Function	Changes a CPU module file name.
	Specification format	rename "old file path name" "new file path name" (abbreviation: ren)
	Example	When changing the name of a file stored in the standard RAM: rename 3:\MAINSEQ1.QDR 3:\SEQ1OLD.QDR
	Note	The following response code is displayed when the file is successfully renamed. 350 Need more info. 250 Rename successful.
user	Function	Inputs the user name and password for the connected FTP server.
	Specification format	user "user name" [FTP password] • User name: Login name set in parameter of the CPU module • FTP password: FTP password set in parameter of the CPU module
	Example 1	When specifying the user name user CPU
	Example 2	When specifying the user name and password user CPU CPU

(2) CPU module dedicated subcommands

The following table lists the CPU module dedicated subcommands that are added to the "quote" command and sent.

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 the execution result when the processing completes successfully. • 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. • 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 a file stored in the standard RAM: quote change 3:\MAINSEQ1.QDR
	Example 2	When changing the attribute of a file stored in the standard RAM: quote change 3:\MAINSEQ1.QDR r
password-unlock	Function	Specifies the remote password set in the CPU module and unlocks the password.[*1]
	Specification format	quote password-unlock [remote password] • Remote password: Specify the remote password set in parameter of the CPU module. The following is displayed as the execution result when the processing completes successfully. 200 Command 0key The following is displayed if the remote passwords do not match. 556 Password Error The following is displayed if another command is requested before the remote password is unlocked. 555 Password Locked
	Example	When specifying a remote password (1234): quote password-unlock 1234
	Note	• If the FTP communication port is specified as a remote password check target when logging in, the password is locked. • By executing this command before various FTP operations, the password is unlocked, allowing file operations of the CPU module. • If the FTP communication port is not specified as a remote password check target, Unlock processing will be completed if the remote password is unlocked.

[*1] 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.[*2]
	Specification format	quote password-lock The following is displayed as the execution result when the processing completes successfully. 200 Command 0key
	Example	When locking the remote password: quote password-lock

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

run	Function	Sets the CPU module to the RUN state (remote RUN). At this time, device memory clear can be specified.
	Specification format	quote run [mode [clear mode]] • Mode: Specify whether to forcibly execute remote RUN or not. 0: Normal RUN (default) 1: Forced RUN • 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. The following message is displayed as the execution result when the processing completes successfully. 200 Command successful
	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	• 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 operating status of the CPU module cannot be changed from STOP/PAUSE to RUN if the status has been 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 setting ("Initial Device value" setting on the "PLC File" tab).
status	Function	Displays the operating status of the CPU module. This command is used to check the operating status of the CPU module when transferring files to the CPU module.
	Specification format	quote status One of the following is displayed as the execution result when the processing completes successfully. • 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	Sets the CPU module to the STOP state (remote STOP).
	Specification format	quote stop The following message is displayed as the execution result when the processing completes successfully. 200 Command successful
	Note	Before writing data to the program memory, set the CPU module to the STOP state using this command.

pm-write	Function	Transfers data in the program cache memory to the program memory.
	Specification format	quote pm-write The following message is displayed as the execution result when the processing completes successfully. 200 Command successful
	Note	Before writing data, set the CPU module to STOP.
passwd-rd	Function	Sets, displays, or clears the read password (file password 32) registered in the file transfer target file. The CPU module checks the read password when accessing the transfer target file.
	Specification format	quote passwd-rd [read password] One of the following is displayed as the execution result when the processing completes successfully. • When setting a read password: 200 Command successful • When displaying the read password: 200 Read-password is [read password] • When clearing the read password: 200 Command successful • When displaying the read password with no password set: 200 Read-password is not set.
	Example 1	When setting a read password (ABCD1234@efgh): quote passwd-rd ABCD1234@efgh
	Example 2	When displaying the read password that is currently set to FTP: quote passwd-rd
	Example 3	When clearing the read password that is currently set to FTP: quote passwd-rd c, or quote passwd-rd C
	Note	• One read password can be set to FTP of the CPU module. When the transfer target file is changed, set the read password of the target file again if a read password is preset to the new target file. • When the CPU module is logged in, the read password is initialized (cleared).
passwd-wr	Function	Sets, displays, or clears the write password (file password 32) registered in the file transfer target file. The CPU module checks the write password when accessing the transfer target file.
	Specification format	quote passwd-wr [write password] One of the following is displayed as the execution result when the processing completes successfully. • When setting a write password: 200 Command successful • When displaying the write password: 200 Write-password is [write password] • When clearing the write password: 200 Command successful • When displaying the write password with no password set: 200 Write-password is not set.
	Example 1	When setting a write password (1234@ABCDefgh): quote passwd-wr 1234@ABCDefgh
	Example 2	When displaying the write password that is currently set to FTP: quote passwd-wr
	Example 3	When clearing the write password that is currently set to FTP: quote passwd-wr c, or quote passwd-wr C
	Note	• One write password can be set to FTP of the CPU module. When the transfer target file is changed, set the write password of the target file again if a write password is preset to the new target file. • When the CPU module is logged in, the write password is initialized (cleared).

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.
- FTP operations are not possible from Microsoft® Internet Explorer. If attempted, an error will occur in Internet Explorer.

(2) Processing on CPU module side

- Files only in the drives of the host station CPU module can be accessed.
- Do not power off or reset the CPU module, or do not remove the SD memory card during file access. Doing so may corrupt the file. Stop accessing to the SD memory card before performing these operations. All of SD memory card operations can be disabled by SM606 (SD memory card forced disable instruction). For the forced disablement of the SD memory card, refer to the following.

  📖 MELSEC-L CPU Module 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, scam 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 the FTP function execution is completed.
- If an access is attempted from an FTP client during backup or restoration for the CPU module change function using the SD memory card, an error will occur. Reexecute it after completion of the backup or restoration. An error will also occur if backup or restoration is performed during access from an FTP client. In that case, reexecute it after disconnecting the FTP client.

(3) Communication processing

- If a timeout error occurs during file transfer, the TCP connection will be closed. 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 29, Section 3.5.
- The processing time for file transfer differs depending 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 the 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) Writing files

- An existing file cannot be overwritten. Delete or rename an existing file with the file delete command (delete, mdelete) or the file rename command (rename) before writing files.
- A read-only file or a file locked by the function except for FTP cannot be written. If attempted, a write error occurs.
- A file cannot be transferred when the SD memory card used is protected. If attempted, a write error occurs.
- A temporary file (FTP_I***.TMP) is automatically created at a file writing. The file is renamed the write target file name when a writing is completed, however, the temporary file may remain if the system is powered off or the CPU module is reset during a file writing. In this case, 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 an SD memory card, set the CPU module to STOP. If writing is performed in RUN state, a communication error may occur.

(5) Deleting files

- The timing for deleting files must be determined by the user, considering the overall system including the CPU module and programming tool.
- Files cannot be deleted if the SD memory card is write-protected. If attempted, an error occurs.

(6) FTP password

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. **Set the FTP parameters again.**

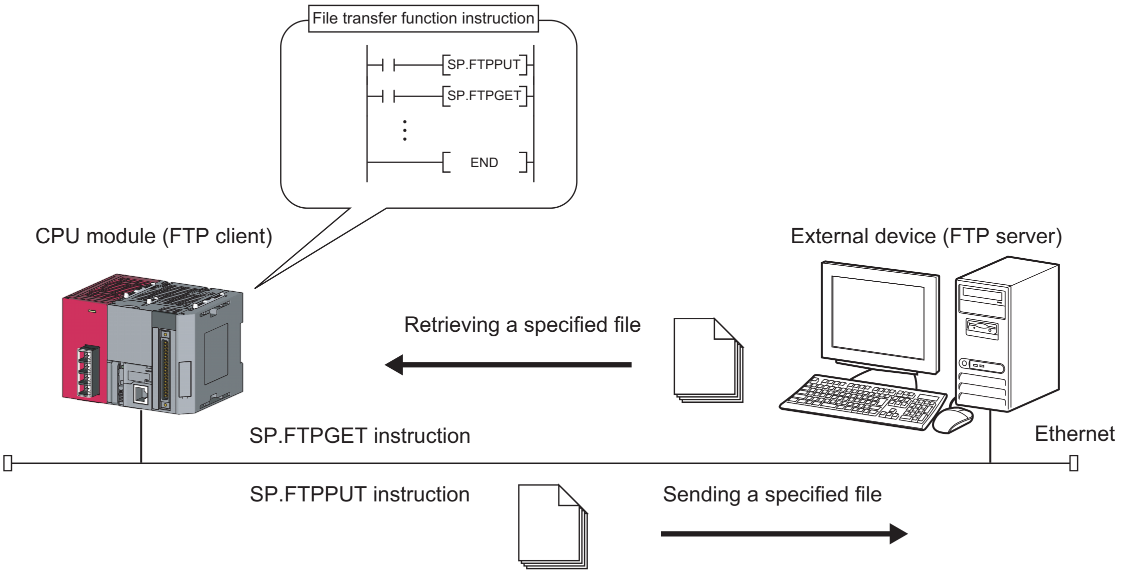
4. **Write the set parameters to the CPU module.**

⇨ [Online] ⇨ [Write to PLC]

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

CHAPTER 10 FILE TRANSFER FUNCTION (FTP CLIENT)

The CPU module becomes an FTP client and can execute file transfer with the FTP server connected to Ethernet using the file transfer function instruction.*1*2



*1 An FTP server is required for using the file transfer function (FTP client). For details of servers, refer to server manuals.
*2 The FTP server of which the operation is ensured by Mitsubishi Electric Corporation are shown below.

FTP server	Compatible OS
Microsoft® Internet Information Services(IIS)	Microsoft® Windows® 7

Remark

When using the file transfer function (FTP client), check the versions of the CPU module and GX Works2. (☞ Page 222, Appendix 3)

10.1 Transferrable Files

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

○: Available, ×: N/A

File type*1	Program memory	Standard RAM	Standard ROM	SD memory card	File name and extension (Any name can be specified for ***.)
	Drive 0	Drive 3	Drive 4	Drive 2	
Parameter	○*2	×	○*2	×	PARAM.QPA
Intelligent function module parameter	○*2	×	○*2	×	IPARAM.QPA
Program	○*2	×	×	×	***.QPG
Device comment	○*2	×	○*2	×	***.QCD
Device initial value	○*2	×	○*2	×	***.QDI
File register	×	○*2	×	×	***.QDR
Local device	×	×	×	×	***.QDL
Sampling trace file	×	○*2	×	×	***.QTD
Programmable controller user data	×	×	○	○	***.CSV ***.BIN
Source information	○*2	×	○	×	***.C32*3
Drive heading	×	×	×	×	Qn.DAT
Device data storage file	×	×	○	×	DEVSTORE.QST
Module error log file	×	○*2	×	×	IERRLOG.QIE
Boot setting file	×	×	×	×	AUTOEXEC.QBT
Remote password	×	×	×	×	00000000.QTM
Latch data backup file	×	×	×	×	LCHDAT00.QBK
Backup data file	×	×	×	×	MEMBKUP0.QBP
Data logging setting file	×	×	×	×	LOGCOM.QLG LOG01.QLG
Data logging file	×	×	×	○	***.CSV
Predefined protocol setting file	×	×	○*2	○	ECPRTCL.QPT, CPRTCL.QPT
Menu definition file	×	×	×	×	MENUDEF.QDF
System file for project data batch save/load	×	×	×	○	SVLDINF.QSL
Device data file	×	×	×	○	DEVDATA.QDT
Sampling trace setting backup file	×	×	×	×	TRCDAT00.QBK
Power failure backup for data logging information	×	×	×	×	LOGDAT00.QBK
System file for the iQ Sensor Solution function (data backup/restoration)	×	×	×	○	SSBRINF.QSI
Backup data file for the iQ Sensor Solution function (data backup/restoration)	×	×	×	○	—*4

*1 Up to 4G bytes of data can be transferred in one file.

*2 The program memory, standard RAM, and standard ROM files of the CPU module can be transferred only from the FTP client to the FTP server.

*3 The file names and the number of files stored in source information varies depending on the type and version of the programming tool used in the label program.

*4 The file name depends on the connection type of the data backup/restoration. (📖 iQ Sensor Solution Reference Manual)

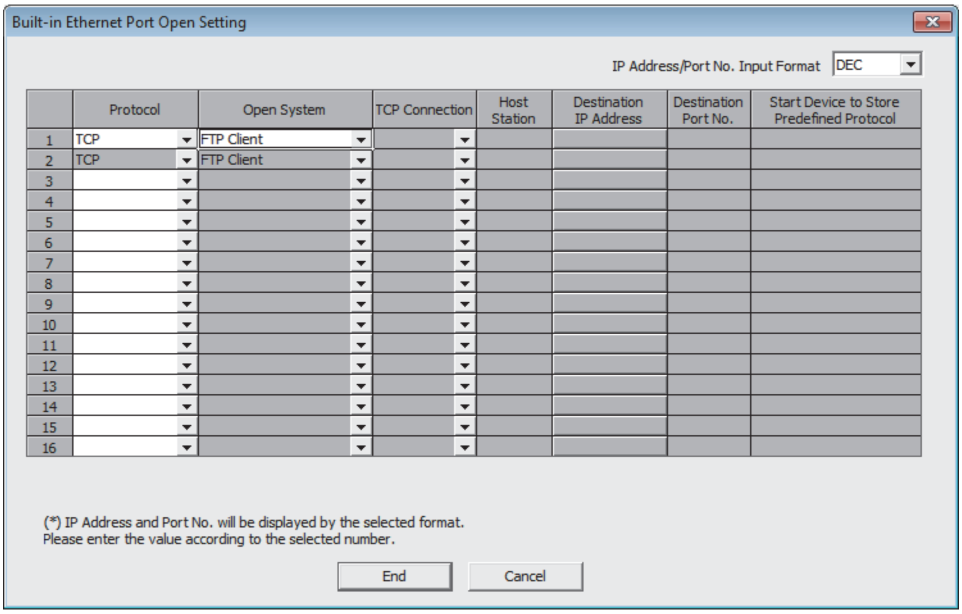
10.2 Setting Method

10.2.1 Parameter setting

(1) Open setting

Set the connection established with the FTP client.

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



Built-in Ethernet Port Open Setting

IP Address/Port No. Input Format: DEC

	Protocol	Open System	TCP Connection	Host Station	Destination IP Address	Destination Port No.	Start Device to Store Predefined Protocol
1	TCP	FTP Client					
2	TCP	FTP Client					
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	Description
Protocol	Select "TCP".
Open System	Select "FTP Client".

This function uses two connections. If "FTP Client" is selected in the open system, the open system of the next connection No. is also changed to "FTP Client".

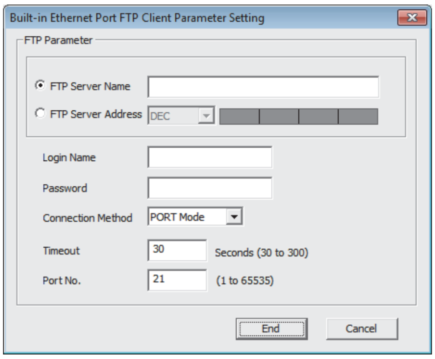
(Note that "FTP Client" cannot be selected in connection No.16.)

"FTP Client" can be set in open system only in one connection.

(2) FTP client setting

Set FTP client parameters.

Project window ➪ [Parameter] ➪ [PLC Parameter] ➪ [Built-in Ethernet Port Setting] ➪ [FTP Client Setting]



Built-in Ethernet Port FTP Client Parameter Setting

FTP Parameter

● FTP Server Name

○ FTP Server Address DEC

Login Name

Password

Connection Method PORT Mode

Timeout 30 Seconds (30 to 300)

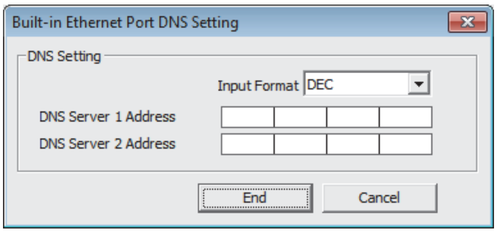
Port No. 21 (1 to 65535)

Item	Description	Setting range	Default
FTP server name	Set the destination FTP server name (domain name).	1 to 256 one-byte alphanumeric characters and symbols	—
FTP server address	Set the IP address of the destination FTP server. The IP address can be specified with decimal or hexadecimal numbers.	0.0.0.1 to 223.255.255.254 (00.00.00.01 to DF.FF.FF.FE)	—
Login Name	Set the login name used to log in to the FTP server.	1 to 32 one-byte alphanumeric characters and symbols	—
Password	Set the password used to log in to the FTP server.	1 to 32 one-byte alphanumeric characters and symbols	—
Connection Method	Set the FTP server connection method.	• PORT Mode • RASV Mode	PORT Mode
Timeout	Set the FTP server timeout time.	30 to 300 seconds	30 seconds
Port No.	Set the port No. used to control the FTP server.	1 to 65535	21

(3) DNS setting

Set the IP address of the DNS server.

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



Built-in Ethernet Port DNS Setting

DNS Setting

Input Format DEC

DNS Server 1 Address

DNS Server 2 Address

Item	Description	Setting range	Default
DNS Server 1 Address	Set the IP address of DNS server 1. The IP address can be specified with decimal or hexadecimal numbers.	0.0.0.1 to 223.255.255.254 (00.00.00.01 to DF.FF.FF.FE)	—
DNS Server 2 Address	Set the IP address of DNS server 2. The IP address can be specified with decimal or hexadecimal numbers.	0.0.0.1 to 223.255.255.254 (00.00.00.01 to DF.FF.FF.FE)	—

10.2.2 Settings at programming

This function executes the file transfer function using the following instructions.

Instruction	Description	Reference
SP.FTPPUT	Sends the specified file to the FTP server.	Page 146, Section 10.3.1
SP.FTPGET	Obtains the specified file from the FTP server.	Page 152, Section 10.3.2

(1) Wild card specification

For an FTP command which can be used for multiple files, specify the file name and extension using a wild card character (* or ?).

Symbol	Description
*	All file names each including an arbitrary character string (including none) from the position where "*" is specified are targeted.
?	All file names each including an arbitrary character (excluding none) at the position where "?" is specified are targeted. ("?" can be used multiple times.)

The following specification methods result in an error.

- Two or more asterisks "*" are used in the file name (before the period) or extension. (Example: "*abc*.csv")
- "*" and "?" are both used in the file name (before the period) or extension. (Example: "*ab?.csv")

The number of files that can be transferred with the wild card specification is determined by the total size of the file names of the specified files.

Specified files can be transferred when the number of these files and the total size of the file names satisfy the following expression.

If a file transfer function instruction is executed when the following expression is not satisfied, the instruction is completed with an error.

$$(Fi+NM) + 1 < 65536$$

N: Total number of files that match the wild card specification
Fi: Total size of the file names that match the wild card specification
M: 9 bytes (fixed)

Point

- If an error occurs in sending or receiving a file included in the target files during execution of the file transfer function instruction, the transfer is stopped at the time of error occurrence. (Remaining files are not transferred.)
If an error occurs during file transfer, re-execute the file transfer.
- If the same file name exists in the destination while "Do not overwrite" is specified in the file overwrite setting, none of the specified files including the relevant file are transferred.
- A file whose file path exceeds 255 characters is not transferred even if it satisfies the wild card specification conditions.

(2) Automatic creation of folder

If file transfer is executed by specifying a folder that does not exist as the transfer destination, the specified folder is automatically created in the destination and files are transferred to the folder.

(3) File overwrite specification

Specify the operation to be performed during file transfer if a file with the same name as one to be transferred exists in the transfer destination.
Specify it in the argument of the file transfer function instruction.

Specification item	Description
Do not overwrite	The file with the same name is not transferred. (Completed with an error)
Overwrite all	The relevant file in the destination is overwritten by the transferred file.
Overwrite only updated file	The time stamp of the destination file is compared with that of the transfer source. • If the time stamp of the destination is later: File transfer is not performed. • If the time stamp is the same or the time stamp of the transfer destination is older: File transfer is performed. (Overwritten)

Point

● Time stamp of transfer file
The time stamp of the transfer file may not be obtained depending on the FTP server. Be careful, in this case, the transfer file is always overwritten even when "Overwrite only updated file" is specified in the file overwrite setting.

● CPU module (FTP client) time and FTP server time
Execute the time setting function (SNTP client) to match the CPU module (FTP client) time and FTP server time. If the time is inconsistent, time stamps cannot be compared correctly even if "Overwrite only updated file" is selected in the file overwrite setting.

● SD1256 (number of files to be processed) and SD1257 (number of files processed completely)
The total number of files to be processed is stored in SD1256 and the number of files which have been processed is stored in SD1257. However, while "Overwrite only updated file" is selected in the file overwrite setting, the number of files which have been judged as not targets for file transfer is included in SD1256 and SD1257.

(4) Specification of measures in case of power off

Specify whether to take measures against power off during file transfer.
If the CPU module is powered off or reset during file transfer while the same file exists in the transfer destination, the transfer target file may be overwritten incompletely.
If "Take measures against power off" is set, the transfer file is transferred to the temporary file (FTPCLI_I.TMP) and, after completion of transfer, the file name is changed to the transfer target file name.
This prevents the transfer target file from being overwritten by an incomplete file even If the CPU module is powered off or reset during file transfer.
Specify it in the argument of the file transfer function instruction.

Specification item	Description
Take measures against power off	A temporary file is created during file transfer.
No measures against power off	A temporary file is not created during file transfer.

If the CPU module is powered off or reset during file transfer, an unnecessary file (temporary file or incomplete transfer target file) may remain in the transfer destination. The following discusses the deletion of this file.

- During file transfer to the FTP server (SP.FTPPUT instruction): Access the FTP server and directly delete the unnecessary file remaining in the FTP server.
- During file transfer to the FTP server (SP.FTPGET instruction): Unnecessary files are deleted as follows.
    - Temporary file: The temporary file remaining in the FTP client is automatically deleted when the next file transfer to the same directory begins.
    - Incomplete transfer target file The incomplete transfer target file remaining in the FTP client is overwritten when the next file transfer to the same file is performed. (This is applicable when "Overwrite all" or "Overwrite only updated file" is selected in the file overwrite specification.)

Point

Suppose "Take measures against power off" is specified and the same file as the transfer target file exists in the transfer destination. If the size of free space in the transfer destination is smaller than the size of the transfer file (temporary file), and error occurs even when "Overwrite all" or "Overwrite only updated file" is specified in the file overwrite setting.

(5) Special relay and special register

The following table lists the special relay and special register used with this function.

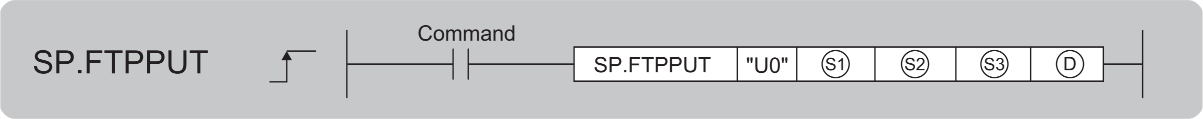
For details, refer to the following.

MELSEC-L CPU Module User's Manual (Hardware Design, Maintenance and Inspection)

Number	Name	Description
SM1298	FTP client connection status	This relay turns on when the connection with the FTP server is established. The relay turns off when the connection with the FTP server is cut off (disconnection).
SD1256	Total number of files to be processed	This register stores the total number of files to be processed by the file transfer function (FTP client). The data is stored at the start of the file transfer (END processing).
SD1257	Number of files processed	This register stores the number of files processed by the file transfer function (FTP client). The data is cleared to 0 at the start of the transfer processing. The data is incremented by one every time processing of a file is completed.

10.3 File Transfer Function Instruction

10.3.1 FTP client file transmission (SP.FTPPUT)



Setting data	Internal device		R, ZR	J□\□		U□\G□	Zn	Constant $	Others
	Bit	Word		Bit	Word				
(S1)	—	○*1	○*1	—				—	—
(S2)	—	○	○	—				○	—
(S3)	—	○	○	—				○	—
(D)	○*1	—	○*1	—				—	—

*1 Local devices cannot be used.

(1) Setting data

Setting data	Description	Set by	Data type
U0	Dummy	—	Character string
(S1)	Start number of the device from which control data are stored	User, system	Device name
(S2)	String data of the transfer source file path, or start number of the device in which string data is stored	User	Character string
(S3)	String data of the transfer source directory path, or start number of the device in which string data is stored	User	Character string
(D)	Start number of the device which turns on for one scan upon completion of the instruction (D)+1 also turns on when the instruction is completed with an error.	System	Bit

(2) Control data

Device	Item	Description	Setting range	Set by
S1+0	Execution type	b15 … b4 b3 b2 b1 b0 S1+0: 0 | [2] | 0 | [1] [1] File overwrite setting Specify the operation to be performed if a file with the same name exists in the transfer destination. • 00: Do not overwrite • 01: Overwrite all • 10: Overwrite only updated file [2] Specification of measures in case of power off Specify whether to take measures against power off. • 0: Take measures against power off • 1: No measures against power off	0000_H to $000A_H$	User
S1+1	Completion status	Completion status is stored • 0000_H: Completed normally • Other than 0000_H: Completed with an error (error code)	—	System
S1+2	Total number of files to be processed	The total number of files to be processed is stored.	—	System
S1+3	Number of files processed	The number of files processed completely is stored.	—	System

(3) Function

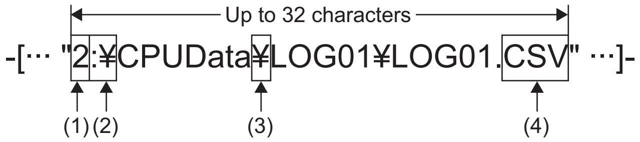
- The SP/.FTPPUT instruction sends the specified file from the CPU module to the FTP server.
- S2S3The function connects to the FTP server specified by "FTP Client Setting", and sends the file specified by S2 to the directory specified by S3. The function releases the connection from the FTP server after file transfer.
- The function stores the total number of transfer files in "Total number of files to be processed" of S1+2 and the number of files which has been transferred completely in "Number of files processed" of S1+3.

Point

If the values of S1+2 and S1+3 are not the same at completion of the instruction, the file transfer failed. Re-execute the instruction.

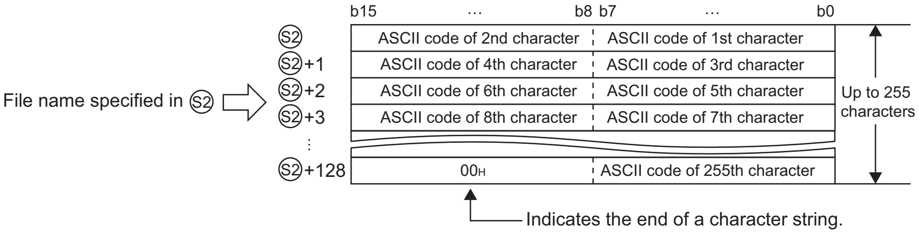
(a) Ⓢ2 (transfer file path)

- In Ⓢ2, specify the character string beginning from the drive No. (0, 2, 3, or 4) through to the file name including an extension. Use one-byte "\" or "/" for the delimiters of directory and file names. Up to 32 one-byte characters can be used in the specified string. However, up to 255 characters can be used by specifying a device number. (Up to 246 characters can be set for the path excluding the file name.)



(1) Drive numbers that can be specified are 0, 2, 3, or 4.
(2) Use ":\" or ":/" for the delimiter of the drive number.
(3) Use "\" or "/" for the delimiters of directory and file names.
(4) The specified file name should include an extension.

- The character strings that can be specified in Ⓢ2 include one-byte letters, digits, symbols, kana characters, and two-byte characters (Shift JIS code).
- The method of device setting in Ⓢ2 is as follows.

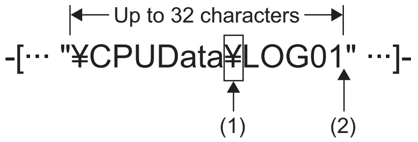

File name specified in Ⓢ2
Up to 255 characters
Indicates the end of a character string.

	b15 ··· b8	b7 ··· b0
Ⓢ2	ASCII code of 2nd character	ASCII code of 1st character
Ⓢ2+1	ASCII code of 4th character	ASCII code of 3rd character
Ⓢ2+2	ASCII code of 6th character	ASCII code of 5th character
Ⓢ2+3	ASCII code of 8th character	ASCII code of 7th character
⋮		
Ⓢ2+128	00H	ASCII code of 255th character

- Wild cards (*, ?) can be used in file names in Ⓢ2. (☞ Page 143, Section 10.2.2 (1))

(b) Ⓢ3 (transfer destination directory path)

- In Ⓢ3, specify the character string indicating the file transfer destination directory in the FTP server. (Any file name in the transfer destination cannot be specified.)
- When using the FTP server function of the CPU module in place of the FTP server, specify the character string beginning from the drive number.
- Use one-byte "\" or "/" for the delimiters of directory names. The delimiter at the end of string can be omitted. If the delimiter is omitted, "\" is assumed at the end.
- Up to 32 one-byte characters can be used in the specified string. However, by specifying with a device number, up to 255 characters including the file name of Ⓢ2 (up to 254 characters if the trailing delimiter is omitted) can be set. The characters that can be used are the same as Ⓢ2.



(1) Use "\" or "/" for the delimiters of directory names.
(2) The delimiter at the end of string can be omitted.

- The character string specified in Ⓢ3 is the relative path from the home directory of the FTP server. If NULL is specified or only "00_H" is specified for the device, it shows directly under the home directory of the FTP server. For details, follow the FTP server specifications.
- If the same file exists in the transfer destination, operation is performed according to the settings in b0 and b1 of Ⓢ1+0. If the same file exists in the transfer destination while "Do not overwrite" is specified, the SP.FTPPUT instruction is completed with an error.

Point

● Two-byte characters of file name
When a file whose file name contains a two-byte character or two-byte code character is transferred, set the kanji code type of the FTP server to Shift JIS.
If the kanji code types do not match between the FTP server and client, file names may not be identified. To prevent this problem, assign a file name with one-byte ASCII code characters to the file transferred to the FTP client by the SP.FTPGET instruction. Do not change the name of the file, which has been transferred to the FTP server using the SP.FTPPUT instruction, in the FTP server.

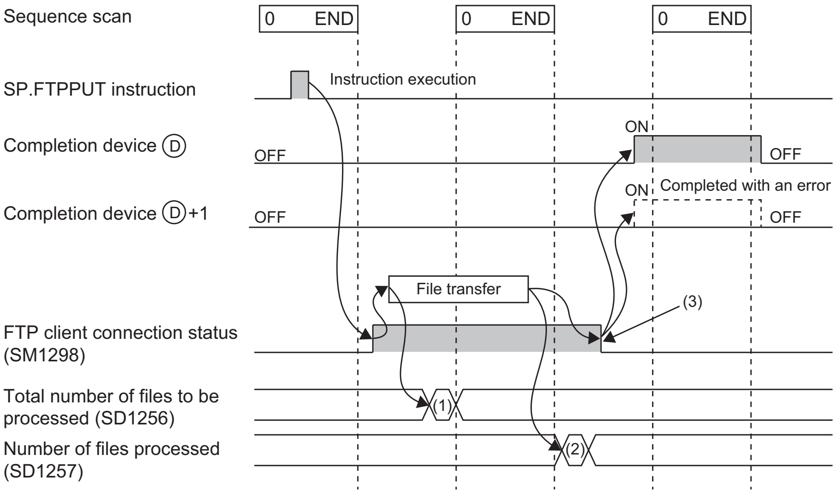
● Distinction of upper-case and lower-case characters
The FTP client (CPU module) does not distinguish between one-byte upper-case and lower-case letters, except in drive 2 (SD memory card), and therefore the specification with lower-case characters is handled the same as the specification with upper-case characters. Upper-case and lower-case characters are distinguished in drive 2.

(c) Instruction execution

- The result of the SP.FTPPUT instruction can be checked with the completion device, Ⓓ+0 and Ⓓ+1.
  - Completion device Ⓓ+0: This device turns on during END processing of the scan where the SP.FTPPUT instruction completes, and turns off during the next END processing.
  - Completion device Ⓓ+1: This device turns on or off according to the result of the SP.FTPPUT instruction.

State	Description
When completed normally	Remains off.
When completed with an error	Turns on during END processing of the scan where the SP.FTPPUT instruction completes, and turns off during the next END processing. When the instruction is completed with an error, an error code is stored in the completion status of Ⓢ1+1.

- The SP.FTPPUT instruction runtime operation is as follows.



(1) The total number of files to be processed is stored.
(2) When transfer is completed successfully, the number of completely processed files is incremented.
(3) When all files have been transferred completely, this status turns off.

- SM1298 (FTP client connection status) turns on during connection to the FTP server and turns off when the connection is released.
- During execution of the SP.FTPPUT instruction, SM721 (file accessing flag) turns on.

(4) Error

Each of the following events results in an operation error, and the error flag (SM0) turns on and an error code is stored in SD0.

- An instruction is used in a CPU module with a serial number (first five digits) of "16111" or earlier.

  (Error code: 4002)

- A device which cannot be specified is specified.

  (Error code: 4004)

- One of the following types of data is specified.

  (Error code: 4100)

  - Data which cannot be handled by the instruction is specified in Ⓢ2 or Ⓢ3.
  - When a wild card is used in the file name specified in Ⓢ2, the number of characters making up the storage path of a matching file excluding the file name exceeds 246.
  - The number of characters of the string stored in the device specified by Ⓢ2 exceeds 255.
  - The number of characters obtained by subtracting those of the file name from the string stored in the device specified by Ⓢ2 exceeds 246.
  - Two or more asterisks "*" are used in the file name (before the period) or extension specified in Ⓢ2.
  - Ⓢ2"*" and "?" are both used in the file name (before the period) or extension specified in Ⓢ2.
  - The character string specified in Ⓢ3 includes the number of characters outside the specification range.
  - The delimiter of the drive number of Ⓢ2 is neither ":\" nor ":/".
  - The file name specified by Ⓢ2 has a file extension not permitted for file transfer.
  - No file name is specified in Ⓢ2.

- The device specified by Ⓢ1 or Ⓓ exceeds the range of the number of device points.

  (Error code: 4101)

- "00_H" does not exist in the range from the device number specified by Ⓢ2 or Ⓢ3 to the relevant device.

  (Error code: 4101)

- The SP.FTPPUT instruction is executed with "FTP client" not set in the open method of the PC parameter.

  (Error code: 4160)

(5) Precautions

- When executing the SP.FTPPUT instruction, turn on SM721 (file accessing flag) for interlocking. When SM721 is on, the SP.FTPPUT instruction performs no processing.
- When executing the SP.FTPPUT instruction by specifying an SD memory card as the transfer source or transfer destination, turn on SM606 (SD memory card forced disable instruction) for interlocking. When SM606 is on, the SP.FTPPUT instruction performs no processing.
- When access is restricted by file password 32, the SP.FTPPUT instruction is completed with an error.
- The SP.FTPPUT and SP.FTPGET instructions cannot be executed simultaneously. A subsequent instruction is ignored and not executed until the execution of the preceding instruction is completed.
- When a block password is set to the transfer source file and "Execution Program Protection Setting" is enabled for the password, the SP.FTPPUT instruction is completed with an error.

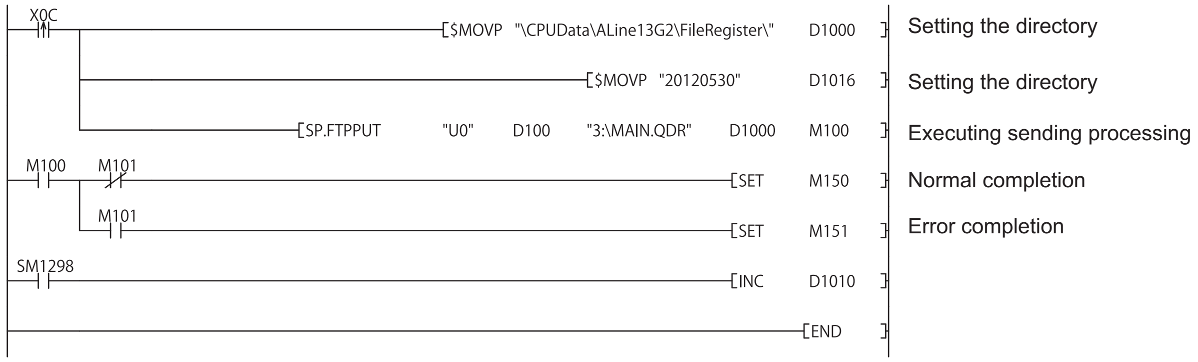
(6) Program example

When XC is turned on, this sample program sends the file register (MAIN.QDR) stored in the standard RAM to the "\CPUData\ALine13G2\FileRegister\20120530" directory in the FTP server.

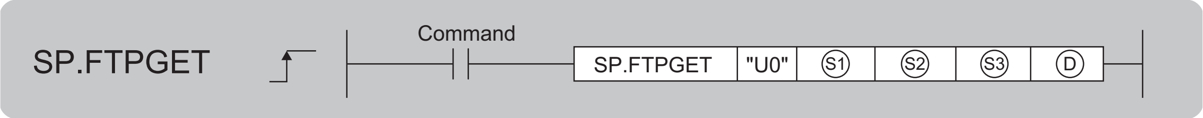
- Devices used

Device number	Application
D100	SP.FTPPUT instruction control data
D1000	Start number of the device in which string data of the transfer destination directory path is stored
M100	SP.FTPPUT instruction completion device
M101	SP.FTPPUT instruction completion device (when completed with an error)
M150	Successful completion indication bit
M151	Unsuccessful completion indication bit
SM1298	FTP client connection status

- Program



10.3.2 FTP client file acquisition (SP.FTPGET)



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

*1 Local devices cannot be used.

(1) Setting data

Setting data	Description	Set by	Data type
U0	Dummy	—	Character string
Ⓢ1	Start number of the device from which control data are stored	User, system	Device name
Ⓢ2	String data of the transfer source file path, or start number of the device in which string data is stored	User	Character string
Ⓢ3	String data of the transfer source directory path, or start number of the device in which string data is stored	User	Character string
Ⓓ	Start number of the device which turns on for one scan upon completion of the instruction Ⓓ+1 also turns on when the instruction is completed with an error.	System	Bit

(2) Control data

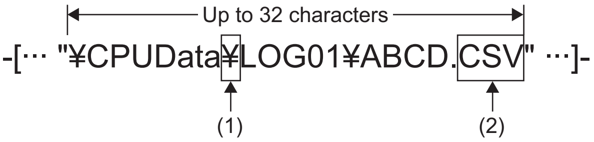
Device	Item	Description	Setting range	Set by
Ⓢ1+0	Execution type	b15 … b4 b3 b2 b1 b0 Ⓢ1+0 | 0 | [2] | 0 | [1] [1] File overwrite setting Specify the operation to be performed if a file with the same name exists in the transfer destination. • 00: Do not overwrite • 01: Overwrite all • 10: Overwrite only updated file [2] Specification of measures in case of power off Specify whether to take measures against power off. • 0: Take measures against power off • 1: No measures against power off	0000_H to $000A_H$	User
Ⓢ1+1	Completion status	Completion status is stored • 0000_H: Completed normally • Other than 0000_H: Completed with an error (error code)	—	System
Ⓢ1+2	Total number of files to be processed	The total number of files to be processed is stored.	—	System
Ⓢ1+3	Number of files processed	The number of files processed completely is stored.	—	System

(3) Function

- The SP.FTPGET instruction gets the specified file from the FTP server to the CPU module.
- Ⓢ2Ⓢ3The function connects to the FTP server specified by "FTP Client Setting", and gets the file specified by Ⓢ2 to the directory specified by Ⓢ3. The function releases the connection from the FTP server after getting the file.
- The function stores the total number of transfer files in "Total number of files to be processed" of Ⓢ1+2 and the number of files which has been transferred completely in "Number of files processed" of Ⓢ1+3.

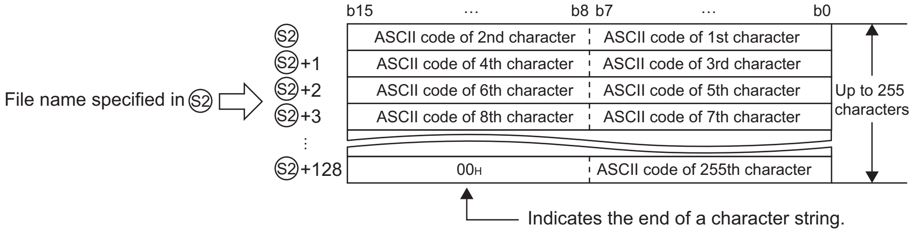
(a) Ⓢ2 (transfer file path)

- In Ⓢ2, specify the character string beginning from the home directory of the FTP server to the file name including an extension.
- When using the FTP server function of the CPU module in place of the FTP server, specify the character string beginning from the drive number.
- Use one-byte "\" or "/" for the delimiters of directory and file names.
- Up to 32 one-byte characters can be used in the specified string. However, up to 255 characters can be used by specifying a device number.



(1) Use ":\" or ":/" for the delimiter.
(2) The specified file name should include an extension.

- The character strings that can be specified in Ⓢ2 include one-byte letters, digits, symbols, kana characters, and two-byte characters (Shift JIS code).
- The method of device setting in Ⓢ2 is as follows.

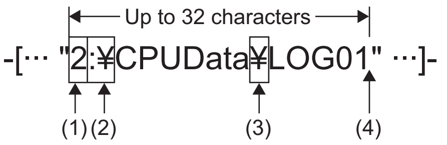

File name specified in Ⓢ2
Up to 255 characters
Indicates the end of a character string.

	b15 ··· b8	b7 ··· b0
Ⓢ2	ASCII code of 2nd character	ASCII code of 1st character
Ⓢ2+1	ASCII code of 4th character	ASCII code of 3rd character
Ⓢ2+2	ASCII code of 6th character	ASCII code of 5th character
Ⓢ2+3	ASCII code of 8th character	ASCII code of 7th character
⋮		
Ⓢ2+128	00H	ASCII code of 255th character

- Wild cards (*, ?) can be used in file names in Ⓢ2. (☞ Page 143, Section 10.2.2 (1))
- The file path in Ⓢ2 is the relative path from the home directory of the FTP server. If only a file name is specified, it shows directly under the home directory of the FTP server. If specifying only a file name, begin with a delimiter. For details, follow the FTP server specifications.

(b) Ⓢ3 (transfer destination directory path)

- In Ⓢ3, specify the character string beginning from the drive No. (2 or 4) through to the transfer destination directory. (Any file name in the transfer destination cannot be specified.)
- Use one-byte "\" or "/" for the delimiters of directory names. The delimiter at the end of string can be omitted. If the delimiter is omitted, "\" is assumed at the end.
- Up to 32 one-byte characters can be used in the specified string. However, by specifying with a device number, up to 247 characters (up to 246 characters if the trailing delimiter is omitted) can be set for the directory path. (The total number of characters of individual file name parts in Ⓢ2 should not exceed 255.) The characters that can be used are the same as Ⓢ2.



(1) Drive numbers that can be specified are 2 and 4.
(2) Use ":\" or ":/" for the delimiter of the drive number.
(3) Use "\" or "/" for the delimiters of directory names.
(4) The delimiter at the end of string can be omitted.

- If the same file exists in the transfer destination, operation is performed according to the settings in b0 and b1 of Ⓢ1+0. If the same file exists in the transfer destination while "Do not overwrite" is specified, the SP.FTPGET instruction is completed with an error.

Point

● Two-byte characters of file name
When a file whose file name contains a two-byte character or two-byte code character is transferred, set the kanji code type of the FTP server to Shift JIS.
If the kanji code types do not match between the FTP server and client, file names may not be identified. To prevent this problem, assign a file name with one-byte ASCII code characters to the file transferred to the FTP client by the SP.FTPGET instruction. Do not change the name of the file, which has been transferred to the FTP server using the SP.FTPPUT instruction, in the FTP server.

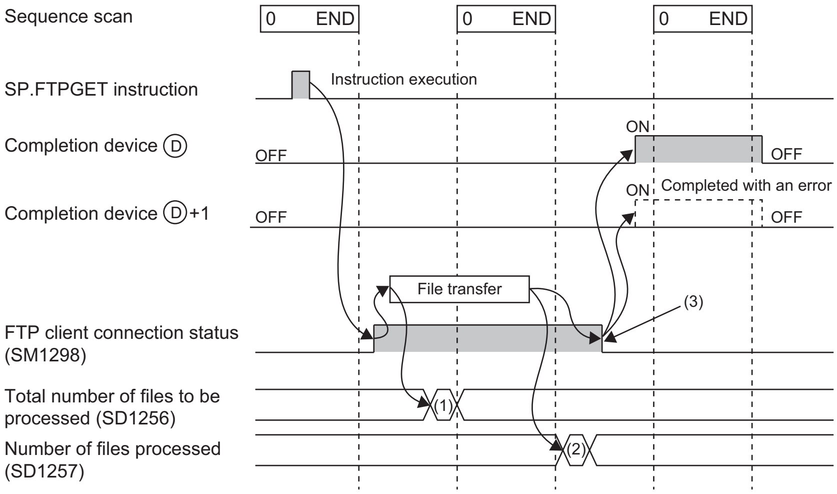
● Distinction of upper-case and lower-case characters
The FTP client (CPU module) does not distinguish between one-byte upper-case and lower-case letters, except in drive 2 (SD memory card), and therefore the specification with lower-case characters is handled the same as the specification with upper-case characters. Upper-case and lower-case characters are distinguished in drive 2.

(c) Instruction execution

- The result of the SP.FTPGET instruction can be checked with the completion device, Ⓓ+0 and Ⓓ+1.
  - Completion device Ⓓ+0: This device turns on during END processing of the scan where the SP.FTPGET instruction completes, and turns off during the next END processing.
  - Completion device Ⓓ+1: This device turns on or off according to the result of the SP.FTPGET instruction.

State	Description
When completed normally	Remains off.
When completed with an error	Turns on during END processing of the scan where the SP.FTPGET instruction completes, and turns off during the next END processing. When the instruction is completed with an error, an error code is stored in the completion status of Ⓢ1+1.

- The SP.FTPGET instruction runtime operation is as follows.



(1) The total number of files to be processed is stored.
(2) When transfer is completed successfully, the number of completely processed files is incremented.
(3) When all files have been transferred completely, this status turns off.

- SM1298 (FTP client connection status) turns on during connection to the FTP server and turns off when the connection is released.
- During execution of the SP.FTPGET instruction, SM721 (file accessing flag) turns on.

(4) Error

Each of the following events results in an operation error, and the error flag (SM0) turns on and an error code is stored in SD0.

- An instruction is used in a CPU module with a serial number (first five digits) of "16111" or earlier.

  (Error code: 4002)
- A device which cannot be specified is specified.

  (Error code: 4004)
- One of the following types of data is specified.

  (Error code: 4100)

  - Data which cannot be handled by the instruction is specified in S2 or S3.
  - The number of characters of the string stored in the device specified by S3 exceeds 246.
  - The total number of characters of individual file name parts in the character strings specified in S3 and S2 exceeds 255.
  - Two or more asterisks "*" are used in the file name (before the period) or extension specified in S2.
  - S2"*" and "?" are both used in the file name (before the period) or extension specified in S2.
  - The character string specified in S3 includes the number of characters outside the specification range.
  - The delimiter of the drive number of S3 is neither ":\" nor ":/".
  - The file name specified by S2 has a file extension not permitted for file transfer.
  - No file name is specified in S2.
- The device specified by S1 or D exceeds the range of the number of device points.

  (Error code: 4101)
- "00_H" does not exist in the range from the device number specified by S2 or S3 to the relevant device.

  (Error code: 4101)
- The SP.FTPGET instruction is executed with "FTP Client" not set in the open method of the PC parameter.

  (Error code: 4160)

(5) Precautions

- When executing the SP.FTPGET instruction, turn on SM721 (file accessing flag) for interlocking. When SM721 is on, the SP.FTPGET instruction performs no processing.
- When executing the SP.FTPGET instruction by specifying an SD memory card as the transfer source or transfer destination, turn on SM606 (SD memory card forced disable instruction) for interlocking. When SM606 is on, the SP.FTPGET instruction performs no processing.
- When access is restricted by file password 32, the SP.FTPGET instruction is completed with an error.
- The SP.FTPPUT and SP.FTPGET instructions cannot be executed simultaneously. A subsequent instruction is ignored and not executed until the execution of the preceding instruction is completed.

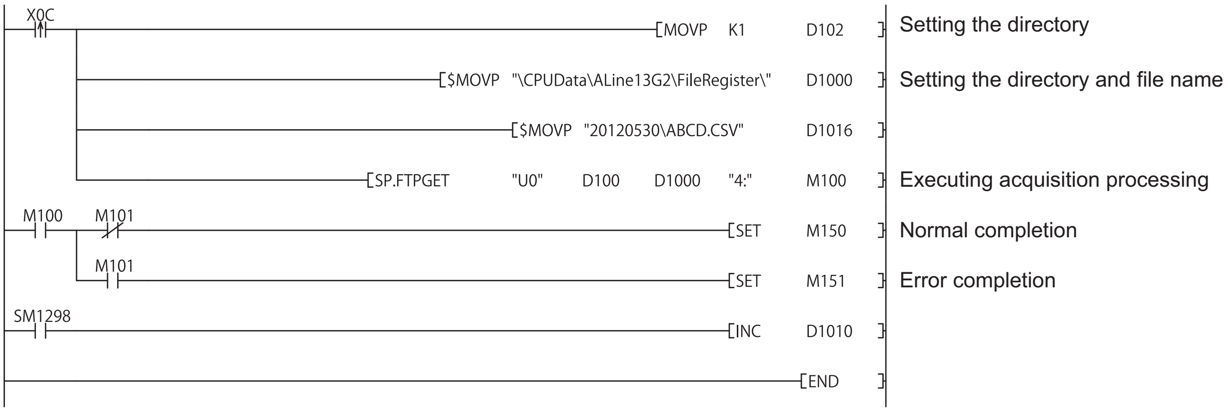
(6) Program example

When XC is turned on, this sample program gets the "ABCD.CSV" file in the "\CPUData\ALine13G2\FileRegister\20120530" directory to the standard ROM.

- Devices used

Device number	Application
D100	SP.FTPGET instruction control data
D1000	Start number of the device in which string data of the get destination directory path is stored
M100	SP.FTPGET instruction completion device
M101	SP.FTPGET instruction completion device (when completed with an error)
M150	Successful completion indication bit
M151	Unsuccessful completion indication bit
SM1298	FTP client connection status

- Program


X0C
[MOVP K1 D102]
Setting the directory
[$MOVP "\CPUData\ALine13G2\FileRegister\" D1000]
Setting the directory and file name
[$MOVP "20120530\ABCD.CSV" D1016]
[SP.FTPGET "U0" D100 D1000 "4:" M100]
Executing acquisition processing
M100
M101
[SET M150]
Normal completion
M101
[SET M151]
Error completion
SM1298
[INC D1010]
[END]

10.4 Diagnostics

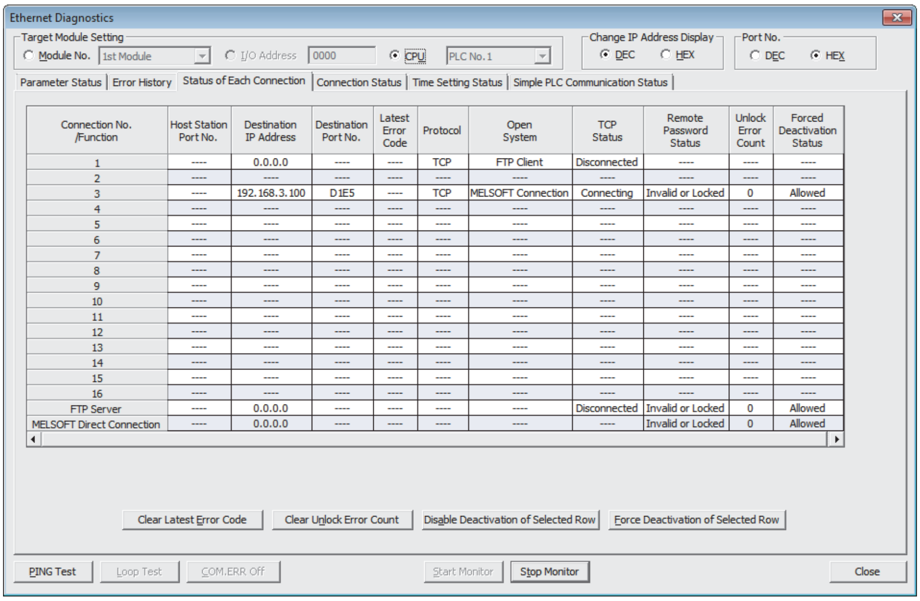
The communication status of the simple PLC communication function can be checked in the "Status of Each Connection" tab in the "Ethernet Diagnostics" window.

[Diagnostics] ⇨ [Ethernet Diagnostics]

For details on Ethernet diagnostics, refer to the following.

📖 GX Works2 Version 1 Operating Manual (Common)

The connection number in which the states and errors of the connections used by this function are stored is the one with the smaller number of the two connections that are set in "FTP Client" in the open setting.



Ethernet Diagnostics

Target Module Setting: ○ Module No. 1st Module ○ I/O Address 0000 ● CPU PLC No.1

Change IP Address Display: ● DEC ○ HEX

Port No.: ○ DEC ● HEX

Parameter Status | Error History | Status of Each Connection | Connection Status | Time Setting Status | Simple PLC Communication Status

Connection No. /Function	Host Station Port No.	Destination IP Address	Destination Port No.	Latest Error Code	Protocol	Open System	TCP Status	Remote Password Status	Unlock Error Count	Forced Deactivation Status
1	----	0.0.0.0	----	----	TCP	FTP Client	Disconnected	----	----	----
2	----	----	----	----	----	----	----	----	----	----
3	----	192.168.3.100	D1E5	----	TCP	MELSOFT Connection	Connecting	Invalid or Locked	0	Allowed
4	----	----	----	----	----	----	----	----	----	----
5	----	----	----	----	----	----	----	----	----	----
6	----	----	----	----	----	----	----	----	----	----
7	----	----	----	----	----	----	----	----	----	----
8	----	----	----	----	----	----	----	----	----	----
9	----	----	----	----	----	----	----	----	----	----
10	----	----	----	----	----	----	----	----	----	----
11	----	----	----	----	----	----	----	----	----	----
12	----	----	----	----	----	----	----	----	----	----
13	----	----	----	----	----	----	----	----	----	----
14	----	----	----	----	----	----	----	----	----	----
15	----	----	----	----	----	----	----	----	----	----
16	----	----	----	----	----	----	----	----	----	----
FTP Server	----	0.0.0.0	----	----	----	----	Disconnected	Invalid or Locked	0	Allowed
MELSOFT Direct Connection	----	0.0.0.0	----	----	----	----	----	Invalid or Locked	0	Allowed

10.5 Precautions

This section provides precautions on using this function.

(1) Precautions on the file transfer function (FTP client)

(a) FTP server connection

When a connection is set up with the FTP server, KeepAlive is used to verify the connection.

An alive check message is sent five seconds after reception of the last message from the FTP server to check if the FTP server returns a response or not. If no response is received, alive check messages are sent at 5-second intervals.

If no response is received for 45 seconds, the FTP server is assumed as not alive and the connection is disconnected.

If the FTP server is not compatible with the TCP KeepAlive function (responding to KeepAlive ACK messages), the connection may be disconnected.

(b) Communication error during file acquisition

If the FTP client file get instruction (SP.FTPGET instruction) is used to get a large file, the load of transfer processing will be increased and a communication error may result.

When transferring a large file, make proper settings of the FTP server, such as maximizing the connection time (timeout time), to prevent a communication error from occurring.

(c) Execution of file transfer function instruction during execution of another function

If one of the following functions is active when the file transfer function instruction is executed, the instruction will be completed with an error.

- Project data batch save/load function
- iQ Sensor Solution function (data backup/restoration)

(d) Execution of another function during execution of file transfer function instruction

If one of the following functions is executed during execution of the file transfer function, an error will result.

- CPU module change function with memory card
- Project data batch save/load function
- iQ Sensor Solution function (data backup/restoration)

(2) Precautions on operations from peripherals

(a) File access

If the file being accessed by this function is accessed by other peripherals, the error processing corresponding to the relevant peripheral may occur.

Do not allow peripherals to access the relevant file while SM721 (file accessing flag) is on.

If the error corresponding to the peripheral does not occur during execution of the FTP client file send instruction (SP.FTPPUT), the file data being transferred to the FTP server is possibly damaged.

CHAPTER 11 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.)

11.1 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.

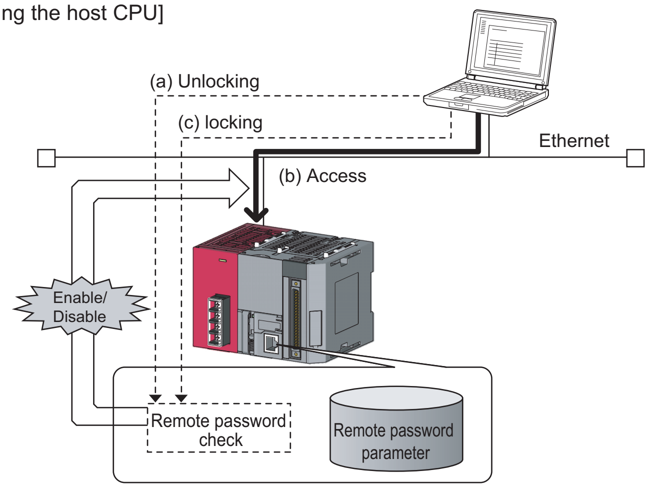
(2) Access processing

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

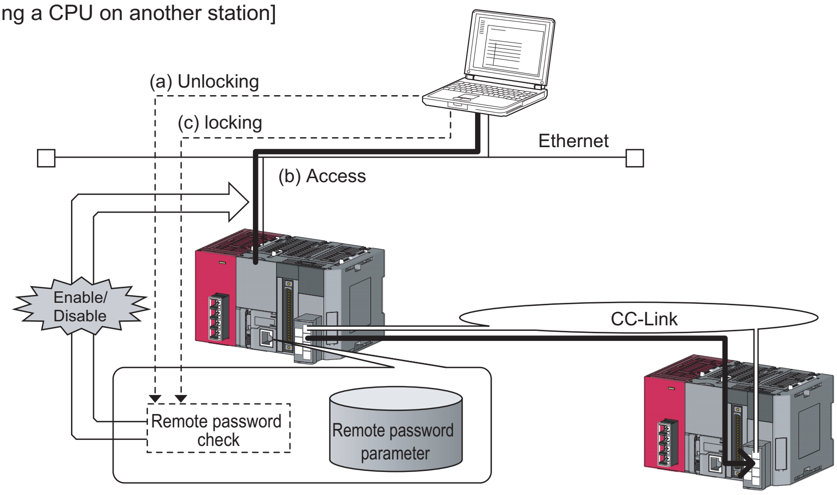
(3) Prohibiting access (lock processing)

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

[Accessing the host CPU]



[Accessing a CPU on another station]

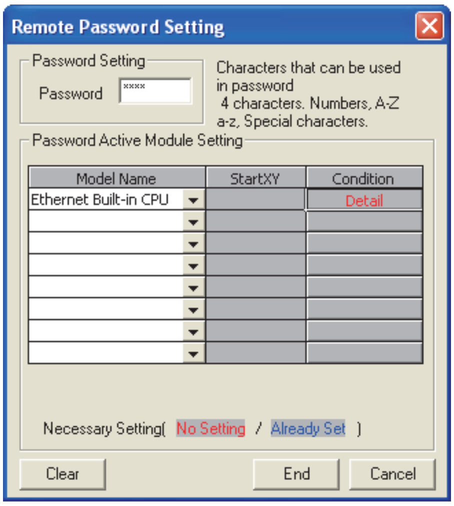


11.2 Remote Password Setting

(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]



Remote Password Setting

Password Setting

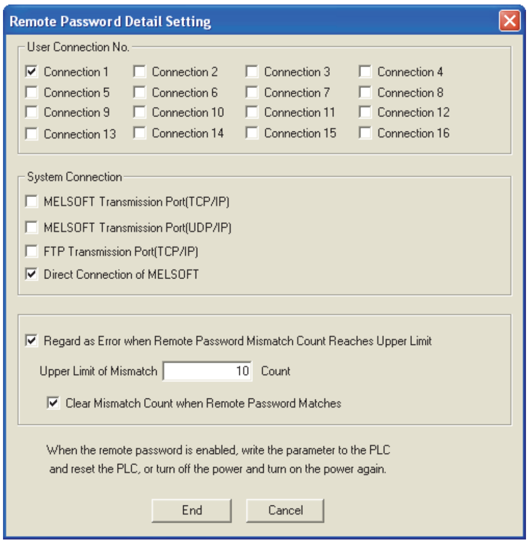
Password ****

Characters that can be used in password
4 characters. Numbers, A-Z a-z, Special characters.

Password Active Module Setting

Model Name	StartXY	Condition
Ethernet Built-in CPU		Detail

Necessary Setting(No Setting / Already Set)



Remote Password Detail Setting

User Connection No.

☑ Connection 1 □ Connection 2 □ Connection 3 □ Connection 4
□ Connection 5 □ Connection 6 □ Connection 7 □ Connection 8
□ Connection 9 □ Connection 10 □ Connection 11 □ Connection 12
□ Connection 13 □ Connection 14 □ Connection 15 □ Connection 16

System Connection

□ MELSOFT Transmission Port(TCP/IP)
□ MELSOFT Transmission Port(UDP/IP)
□ FTP Transmission Port(TCP/IP)
☑ Direct Connection of MELSOFT

☑ Regard as Error when Remote Password Mismatch Count Reaches Upper Limit

Upper Limit of Mismatch 10 Count

☑ Clear Mismatch Count when Remote Password Matches

When the remote password is enabled, write the parameter to the PLC and reset the PLC, or turn off the power and turn on the power again.

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.	Ethernet Built-in CPU
	Condition	Click this to display the "Remote Password Detail Setting" window.	—
User Connection No.[5]	Connection 1 to 16	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 164, Section 11.4)	—
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 31, 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.

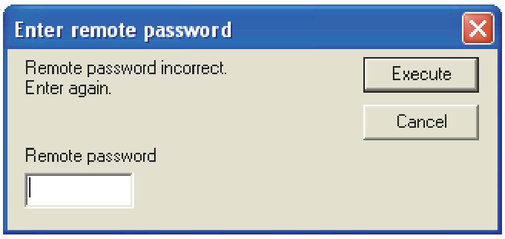
(3) Unlocking or locking the remote password

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



Remote password incorrect.
Enter again.

(b) When using MC protocol

Use commands dedicated to MC protocol. (☞ Page 38, Section 5.1.2)

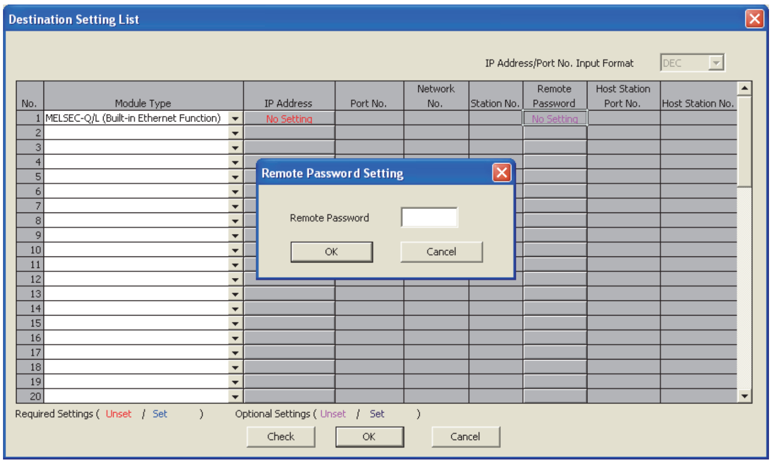
(c) When using the FTP function

Use the "password-lock" and "password-unlock" commands. (☞ Page 127, Section 9.4.1)

(d) When using the simple PLC communication function

Unlock the remote password in the "Destination Setting List" window of Simple PLC Communication Setting.

Project window ⇨ [Parameter] ⇨ [PLC Parameter] ⇨ [Built-in Ethernet Port Setting] ⇨ [Simple PLC Communication Setting] ⇨ [Destination Setting List]



IP Address/Port No. Input Format DEC

No.	Module Type	IP Address	Port No.	Network No.	Station No.	Remote Password	Host Station Port No.	Host Station No.
1	MELSEC-Q/L (Built-in Ethernet Function)	No Setting				No Setting		
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								

Required Settings (Unset / Set) Optional Settings (Unset / Set)

11.3 Precautions

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

- Determine the external device to communicate with and exchange the data. (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.

11.4 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.

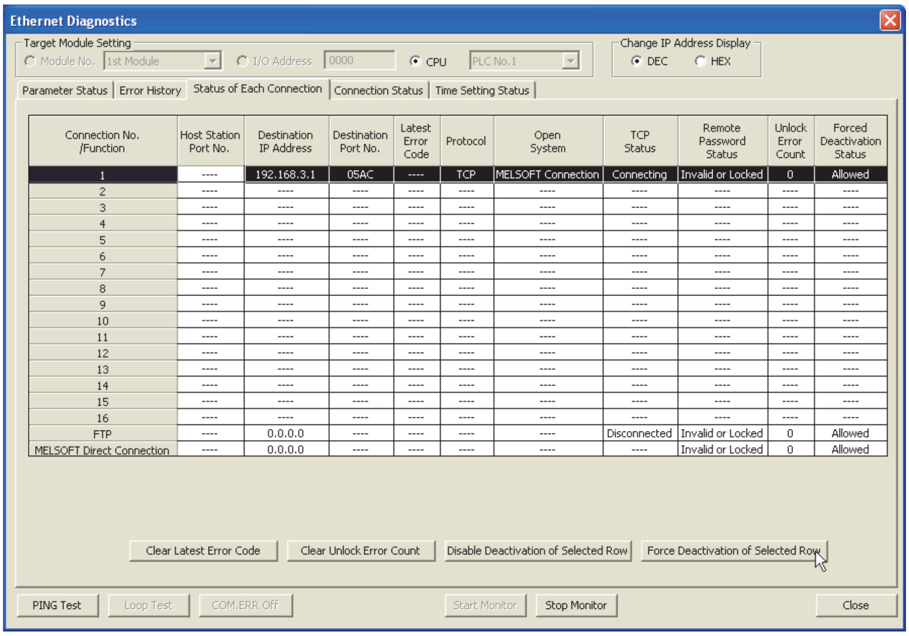
Take the following actions as needed.

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"



Connection No. /Function	Host Station Port No.	Destination IP Address	Destination Port No.	Latest Error Code	Protocol	Open System	TCP Status	Remote Password Status	Unlock Error Count	Forced Deactivation Status
1	----	192.168.3.1	05AC	----	TCP	MELSOFT Connection	Connecting	Invalid or Locked	0	Allowed
2	----	----	----	----	----	----	----	----	----	----
3	----	----	----	----	----	----	----	----	----	----
4	----	----	----	----	----	----	----	----	----	----
5	----	----	----	----	----	----	----	----	----	----
6	----	----	----	----	----	----	----	----	----	----
7	----	----	----	----	----	----	----	----	----	----
8	----	----	----	----	----	----	----	----	----	----
9	----	----	----	----	----	----	----	----	----	----
10	----	----	----	----	----	----	----	----	----	----
11	----	----	----	----	----	----	----	----	----	----
12	----	----	----	----	----	----	----	----	----	----
13	----	----	----	----	----	----	----	----	----	----
14	----	----	----	----	----	----	----	----	----	----
15	----	----	----	----	----	----	----	----	----	----
16	----	----	----	----	----	----	----	----	----	----
FTP	----	0.0.0.0	----	----	----	----	Disconnected	Invalid or Locked	0	Allowed
MELSOFT Direct Connection	----	0.0.0.0	----	----	----	----	----	Invalid or Locked	0	Allowed

- 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).

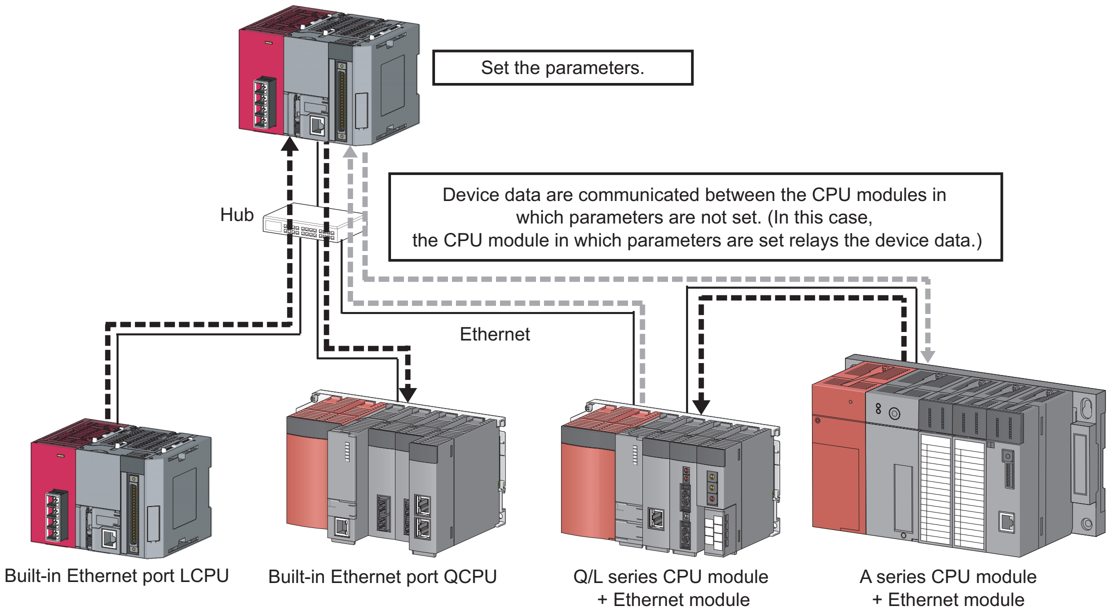
CHAPTER 12 SIMPLE PLC COMMUNICATION FUNCTION

Remark

When using the simple PLC communication function, check the versions of the CPU module and GX Works2. (☞ Page 222, Appendix 3)

This function allows data communications between specified devices at the specified timing just by doing simple settings from a programming tool. Specify one device (transmission source) to one device (transmission destination). The communications are performed between the specified devices.

The communications using this function are not possible when a stop error occurs in the CPU module where the parameters are set.


Set the parameters.
Device data are communicated between the CPU modules in which parameters are not set. (In this case, the CPU module in which parameters are set relays the device data.)
Hub
Ethernet
Built-in Ethernet port LCPU
Built-in Ethernet port QCPU
Q/L series CPU module + Ethernet module
A series CPU module + Ethernet module

Point

The communications using this function are possible only for the CPU modules connected over Ethernet. The following communications are not possible.

- Communication with a CPU module on other stations via CC-Link network
- Communication with a CPU module except the connected Built-in Ethernet port QCPU when multiple CPU system is configured
- Communication with a CPU module that does not control the connected Ethernet module when multiple CPU system is configured

Remark

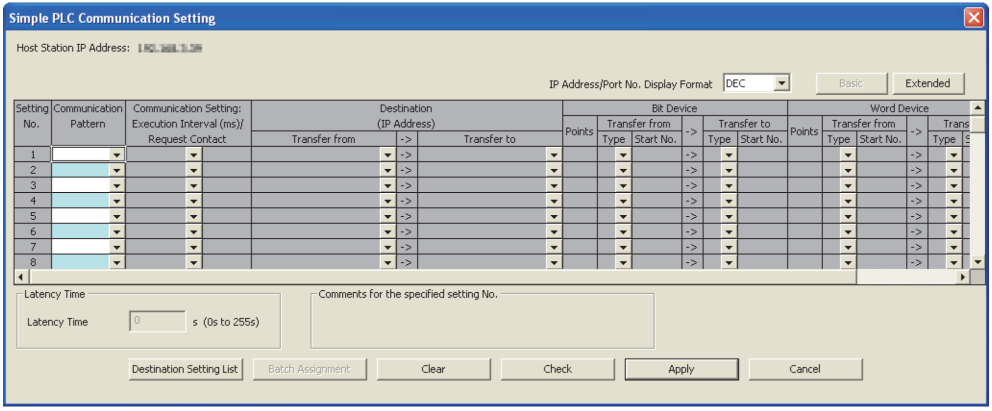
- For an external device, refer to Page 169, Section 12.1 (3) (b).
- Access via routers is also available. For the access, set the subnet mask pattern and the default router IP address. (☞ Page 28, Section 3.4)

12.1 Setting Method

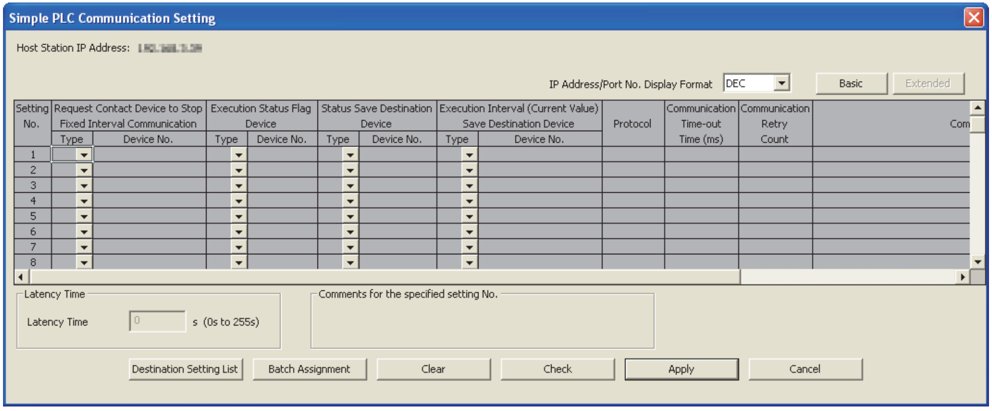
To use this function, configure the settings in "Simple PLC Communication Setting". Up to 64 devices can be set in "Simple PLC Communication Setting".

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

[Base Setting]



[Extended Setting]


Simple PLC Communication Setting
Host Station IP Address: [redacted]
IP Address/Port No. Display Format
DEC
Basic
Extended
Setting No.
Request Contact Device to Stop Fixed Interval Communication
Type
Device No.
Execution Status Flag Device
Status Save Destination Device
Execution Interval (Current Value) Save Destination Device
Protocol
Communication Time-out Time (ms)
Communication Retry Count
Com
1
2
3
4
5
6
7
8
Latency Time
Latency Time
0
s (0s to 255s)
Comments for the specified setting No.
Destination Setting List
Batch Assignment
Clear
Check
Apply
Cancel

Simple PLC Communication Setting consists of Base Setting (essential) and Extended Setting (optional). To switch the window, use the Basic / Extended button on the top right of the window.

(1) Communication Pattern

Select the pattern from the following items.

Setting item	Description
Read	Read the data of the specified destination device (transmission source) to the specified device of the host station (transmission destination).
Write	Write the data of the specified device of the host station (transmission source) to the specified destination device (transmission destination).
Transfer	Read the data of the specified destination device (transmission source) and write it to another specified destination device (transmission destination).[*1]

*1 The device data is not reflected in the CPU module where the parameters are set (the CPU module that relays the data).

(2) Communication Setting: Execution Interval (ms)/Request Contact

Select the communication timing from the following items.

Setting item		Description	Setting range
Fixed	—	Data are communicated between the devices at a specified execution interval.	—
	Execution Interval	Execution interval is set.	10ms to 65535ms (in increments of 1ms)
On Request	—	Data are communicated between the devices only when requested.	—
	Request Contact[*1]	Data are communicated at the rising (off to on) of the device that is specified as Request Contact. The on/off status of Request Contact is confirmed in the END processing.	X, M, B

*1 Request Contact cannot be specified overlapped with the following devices.
- Request Contact Device to Stop Fixed Interval Communication
- Execution Status Flag Device
- Source device when the host station is a transmission destination device
- Request Contact Device to Stop Fixed Interval Communication that is used for other setting No.
- Execution Status Flag Device that is used for other setting No.
- Source device when the host station is a transmission destination device, which is used for other setting No.

Point

- Actual time of execution interval may be longer than the value of the setting because the time is affected by the specified devices or Ethernet line congestion. For the preventive measures, refer to the troubleshooting described in the following manual.

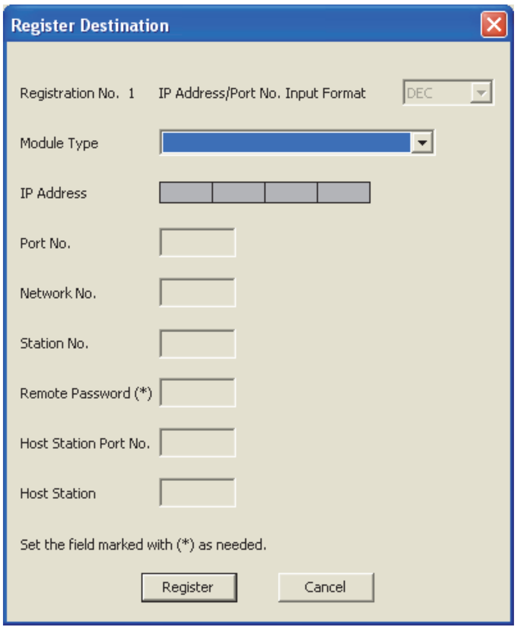
  MELSEC-L CPU Module User's Manual (Hardware Design, Maintenance and Inspection)
- If latency time is set, data communication is started after the latency time has passed. (Page 177, Section 12.1 (12))

(3) Register Destination

(a) Setting of devices for communications

Set the devices for both transmission source and transmission destination. Up to 64 devices can be registered.

Destination Setting List This setting also can be done in the "Destination Setting List" window that will open by clicking the Destination Setting List button in the "Simple PLC Communication Setting" window. Use the "Destination Setting List" window as well to change the setting information.



Register Destination

Registration No. 1

IP Address/Port No. Input Format: DEC

Set the field marked with (*) as needed.

Setting item	Description	Setting range
IP Address/Port No. Input Format	Select the input format of IP address and port No.	DEC/HEX
Module Type	Select the device for communications.	☞ Page 169, Section 12.1 (3) (b)
IP Address	Set the IP address of the device.	0.0.0.1 to 223.255.255.254
Port No.	Set the port No. of the device.	0401_H to 1387_H, 1392_H to $FFFE_H$ (0401_H to $15AC_H$, $15B0_H$ to $FFFE_H$ for communication with FX3)
Network No.	Set the network No. of the device (host station).	1 to 239
Station No.	Set the station No. of the device.	1 to 64
Remote Password	If a remote password is set for the destination device, the communication is possible by entering the password.	4 characters or less (one-byte alphanumeric and special characters)
Host Station Port No.	Set the port No. of the host station.	0401_H to 1387_H, 1392_H to $EFFF_H$ (0401_H to $15AC_H$, $15B0_H$ to $EFFF_H$ for communication with FX3)
Host Station No.	Set the station No. of the host station.*1	1 to 64

*1 Set the station number not to overlap the numbers set for other devices on the same Ethernet network (such as Ethernet module, programming tool, and GOT).

Point

If the L/Q/QnA series modules are used for the simple PLC communication, $F000_H$ to $FFFE_H$ is automatically selected for the port No. of the host station. Therefore, when using the simple PLC communication function, do not set $F000_H$ to $FFFE_H$ to Host Station Port No. for the establishing a connection instruction (SP.SOCOPEN) of the socket communication function. If set, the instruction may not be completed properly.

(b) Module Type list

The following table shows the communication devices and specifications applicable to each module.

Module type	Device for communications	Communication specification
MELSEC-Q/L (Built-in Ethernet Function)	Built-in Ethernet port QCPU	MELSOFT connection
	Built-in Ethernet port LCPU	
MELSEC-Q/L (Ethernet Module)	Basic model QCPU + Ethernet module	
	High Performance model QCPU + Ethernet module	
	Process CPU + Ethernet module	
	Universal model QCPU + Ethernet module	
	Built-in Ethernet port LCPU + Ethernet module	
MELSEC-QnA (Ethernet Module)	QnACPU + Ethernet module	
MELSEC-A (Ethernet Module)/MELSEC-FX3 (Ethernet Block/Adapter)[*1]	ACPU + Ethernet module	MC protocol A-compatible 1E frame (binary code communication)
	High Performance model QCPU (A mode) + Ethernet module	
	FX3U(C)CPU + Ethernet block	
	FX3CPU + Ethernet adapter	

[*1] MELSEC-FX3 (Ethernet Block/Adapter) is compatible with only the CPU module with a serial number (first five digits) of "16042" or later.

(c) Setting for each device for communications

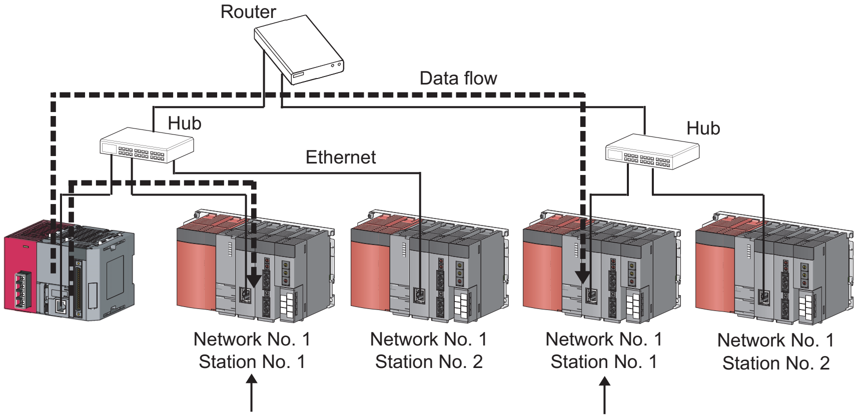
The settings vary depending on the device for communications.

○: Setting required, △: Setting required if the password is set for destination device for communications, ×: Setting not required

Device for communications	Setting item				
	IP Address	Network No.	Station No.	Port No.	Remote Password
MELSEC-Q/L (Built-in Ethernet Function)	○	×	×	×	△
MELSEC-Q/L (Ethernet Module)	○	○[*1*2]	○[*1*2]	×	△
MELSEC-QnA (Ethernet Module)	○	○[*1*2]	○[*1*2]	×	×
MELSEC-A (Ethernet Module)/MELSEC-FX3 (Ethernet Block/Adapter)	○	×	×	○[*1]	×

*1 Setting of the host station is required.

*2 For the "MELSEC-Q/L (Ethernet Module)" and "MELSEC-QnA (Ethernet Module)," settings of these items in "Simple PLC Communication Setting" are required because these modules perform communication processing using the network No. and station No.



The settings must be configured in simple PLC communication setting.

The settings must be configured in simple PLC communication setting.

(4) Settings for destination devices for communications

If following devices are used as the destination devices for communications, the settings of these devices are also required.

Destination device for communications	Necessary setting		
	Parameter	Programming tool	DIP switch on the front of the module
MELSEC-Q/L (Built-in Ethernet Function)	Set the following items (for the number of devices) in the "Open Setting" window.[*1] • Protocol: "UDP" • Open System: "MELSOFT Connection"	—	—
MELSEC-Q/L (Ethernet Module)	Set the following items from the Ethernet setting of Network Parameter. • "Send Frame Setting" of "Operation Setting": "Ethernet(V2.0)" • "Station No.<->IP Information System" of "Station No.<->IP Information": "Automatic Response System"	—	—
MELSEC-QnA (Ethernet Module)	Set the following items from the Ethernet setting of Network Parameter. • "Send Frame Setting" of "Operation Setting": "Ethernet(V2.0)" • "Station No.<->IP Information System" of "Station No.<->IP Information": "Automatic Response System"	• Create an initial processing program and turn off the bits (6 and 7) in the special function setting area (address: 4 (4_H)) of buffer memory. • Turn on the bit 15 in the exchange instruction area during STOP (address: 103 (67_H)) of buffer memory.[*4]	Set the following items. • SW3 (Automatic start up mode setting): ON[*3] • SW7 (CPU exchange timing setting): ON[*2]
MELSEC-A (Ethernet Module)[*5]	—	• Create an initial processing program and do the settings as shown below. · Set the IP address and the port No. referring to (3) in this section. · Set "UDP" for the communication method. • Turn on the bit 15 in the exchange instruction area during STOP (address: 496 ($1F0_H$)) of buffer memory, and the bits corresponding to the connection No. used for bits 0 to 7.[*4]	Set the following items. • SW2 (Data code setting): OFF • SW7 (CPU exchange timing setting): ON[*2]
MELSEC-FX3 (Ethernet Block/ Adapter)[*5]	• Set "Communication Data Code" of "Built-in Ethernet Port Setting" to "Binary Code". • Set the following items (for the number of devices) in the "Open Setting" window. · Protocol: "UDP" · "Open system": "MC Protocol" · Set the IP address and the port No. referring to (3) in this section.	—	—

[1] The module has the setting by default to communicate with one device whose protocol is "UDP".
[2] To execute the simple PLC communication function when the CPU module (destination device for communications) is in RUN state, turn on this switch.
[3] If Ethernet module is in Automatic startup mode, turn on this switch.
[4] To execute the simple PLC communication function when the CPU module (destination device for communications) is in STOP state or in stop error state, turn on the bits.
[5] The module cannot communicate regardless of the setting if a stop error occurs in it.

(5) Applicable devices

The devices that can be specified as transmission source and transmission destination vary depending on the destination device for communications. Bit device and word device can be set together for each setting No.

(a) Number of device points

Set the number of device points to the total number of setting No.1 to 64 (4096 words maximum).

The maximum number of points for each setting No. may differ depending on the communication destination.

· For the MELSEC-FX3 (Ethernet Block/ Adapter)

Transmission source: 96 words maximum (word device 64 points + bit device 512 points)

Transmission destination: 74 words maximum (word device 64 points + bit device 160 points)

· For the cases other than the above

Transmission source/destination: 512 words maximum (word device 256 points + bit device 4096 points)

(b) Devices that can be specified

The following devices can be specified.

Destination device for communications	Type	Applicable device[1]
MELSEC-Q/L (Built-in Ethernet Function), MELSEC-Q/L (Ethernet Module)	Bit device	X, Y, M, L, B, SB, SM
	Word device	D, D (extended data register)[2], W, W (extended link register)[2], R, ZR, SW, SD
MELSEC-QnA (Ethernet Module)	Bit device	X, Y, M, L, B, SB, SM
	Word device	D, W, R, ZR, SW, SD
MELSEC-A (Ethernet Module)	Bit device	X, Y, M[3], B
	Word device	D, W, R
MELSEC-FX3 (Ethernet Block/ Adapter)[4]	Bit device	X, Y, M[3]
	Word device	D, R

[1] The size of a bit device and a word device can be specified in units of 16 points and one point respectively. Use 0 or multiples of 16 to specify the device number of a bit device.
[2] The devices are compatible with the following modules:
· Universal model QCPU with a serial number (first five digits) of "09042" or later (except the Q00UJCPU)
· Built-in Ethernet port LCPU
[3] Use "M9000 + multiples of 16" when specifying the device of M9000 or later.
[4] When B, W is specified for the device, the FX3 side gets an error, disabling communication.

Point

- Do not write any data to the special relay and the special register that are set on the system side. Doing so may cause a system failure or communication failure.
- For I/O processing when X and Y are specified as transmission destination, refer to the following.
  MELSEC-L CPU Module User's Manual (Function Explanation, Program Fundamentals)

(c) Timing of write or read of the device data

The data of a bit device and a word device set for each setting No. are communicated in the END processing.*1 Even so, depending on the destination device for communications, data communications for one setting may not be done completely, resulting in data inconsistency.

Destination device for communications	Device data		Data inconsistency for one setting
	Bit device	Word device	
MELSEC-Q/L (Built-in Ethernet Function)	Data of a bit device and a word device are guaranteed to be communicated by setting unit.		Data inconsistency of a bit device and a word device is the same level because the data of both devices are communicated at the same timing.
MELSEC-Q/L (Ethernet Module)	Data of a bit device are guaranteed to be communicated by setting unit.	Data of a word device are guaranteed to be communicated by setting unit.	If both a bit device and a word device are set for the same setting No., the level of data inconsistency of both devices may be different because the data of each device is communicated at different timing.*2
MELSEC-QnA (Ethernet Module)			
MELSEC-A (Ethernet Module) MELSEC-FX3 (Ethernet Block/ Adapter)	Within the setting range, data of a bit device are guaranteed to be communicated in units of 32 bits (2 words).	Within the setting range, data of a word device are guaranteed to be communicated in units of 2 words.	• If both a bit device and a word device are set for the same setting No., the level of data inconsistency of both devices may be different because the data of each device is communicated at different timing.*2 • If the data of 32 bits (2 words) or larger is communicated, data inconsistency might differ in units of 32 bits (2 words).*3

*1 If the COM instruction is used, the data are written or read not only in the END processing but also during sequence scan (at the execution of the COM instruction).

*2 The order of the write and read is shown below.

- When "Write" is set for Communication Pattern: A word device comes first, then a bit device.
- When "Read" is set for Communication Pattern: A bit device comes first, then a word device.
- When "Transfer" is set for Communication Pattern: The host station reads the data of transmission source in order of a bit device and a word device, and writes them to transmission destination in order of a word device and a bit device.

*3 Because the write and read is performed from the data with smaller device number, use the device with the largest No. to secure the data.

(6) Request Contact Device to Stop Fixed Interval Communication

Select the device from X, M, or B and specify it as Request Contact Device to Stop Fixed Interval Communication.[*1] If "Fixed Interval" is set for Communication Setting, the communications for the specified setting No. temporarily stop when Request Contact Device to Stop Fixed Interval Communication turns on. The communications re-start when Request Contact Device to Stop Fixed Interval Communication turns off. The on/off status is confirmed in the END processing. This stop request is enabled when the communication status is "Communicating" or "Error".

[*1] Request Contact cannot be specified overlapped with the following devices.
· Request Contact
· Execution Status Flag Device
· Source device when the host station is a transmission destination device
· Request Contact that is used for other setting No.
· Execution Status Flag Device that is used for other setting No.
· Source device when the host station is a transmission destination device, which is used for other setting No.

Point

- When communications are stopped by the request to stop fixed interval communication, the data communications at a specified execution interval, which is performed when "Fixed" is set for Communication Setting, is ignored.
- The stop time by the request to stop fixed interval communication is not counted as Execution Interval (Current Value).
- If the communications are stopped by the request to stop fixed interval communication during retry of communication, the retry count is reset to 0.

(7) Execution Status Flag Device

Select the device from X, M, or B and specify it as a device to store the execution status.[*1] The on/off status below indicates the execution status.

- ON: Communicating (the state of "Status" being "3H")
- OFF: Communication stop

[*1] Execution Status Flag Device cannot be specified overlapped with the following devices.
· Request Contact
· Request Contact Device to Stop Fixed Interval Communication
· Source device when the host station is a transmission destination device
· Request Contact that is used for other setting No.
· Request Contact Device to Stop Fixed Interval Communication that is used for other setting No.
· Execution Status Flag Device that is used for other setting No.
· Source device when the host station is a transmission destination device, which is used for other setting No.

Point

For examples of how to use this device, refer to Page 177, Section 12.2.

(8) Status Save Destination Device

Select the device from D, D (extended data register), W, W (extended link register), R, or ZR and specify it as a device to store the communication status.*1 The values listed below indicate the communication status.

Item	Description		
	Communication status	Communication setting	Description
1_H	Preparing	"Fixed" is set.	When waiting for the latency time to pass, or during the time until communications start after the CPU module is powered off and on or reset
		"On Request" is set.	When waiting for the latency time to pass or during the time until communications start after the CPU module is powered off and on or reset (including the waiting time of check if the hub is connected or not)
2_H	Waiting for the request	"On Request" is set.	When waiting for the communication request by Request Contact
3_H	Communicating	"Fixed" is set.	When device data communications at a specified execution interval are possible
		"On Request" is set.	When device data are communicated according to the states of Request Contact
4_H	Communication stop	"Fixed" is set.	When the simple PLC communication function is stopped by the request to stop fixed interval communication
Error Code No.	The corresponding error code is stored.		

*1 Status Save Destination Device cannot be specified overlapped with the following devices.
· Status Save Destination Device
· Execution Interval (Current Value) Save Destination Device
· Source device when the host station is a transmission destination device
· Status Save Destination Device that is used for other setting No.
· Execution Interval (Current Value) Save Destination Device that is used for other setting No.
· Source device when the host station is a transmission destination device, which is used for other setting No.

Point

For examples of how to use this device, refer to Page 177, Section 12.2.

(9) Execution Interval (Current Value) Save Destination Device

Select the device from D, D (extended data register), W, W (extended link register), R, or ZR and specify it as a device to store Execution Interval (Current Value).*1

Item	Description
Execution Interval (Current Value)	If "Fixed" is set for Communication Setting, the actual execution interval of the communications is stored. If the communications have never been performed properly, "0" is stored.

*1 Execution Interval (Current Value) Save Destination Device cannot be specified overlapped with the following devices.
· Status Save Destination Device
· Execution Interval (Current Value) Save Destination Device
· Source device when the host station is a transmission destination device
· Status Save Destination Device that is used for other setting No.
· Execution Interval (Current Value) Save Destination Device that is used for other setting No.
· Source device when the host station is a transmission destination device, which is used for other setting No.

Point

For examples of how to use this device, refer to Page 177, Section 12.2.

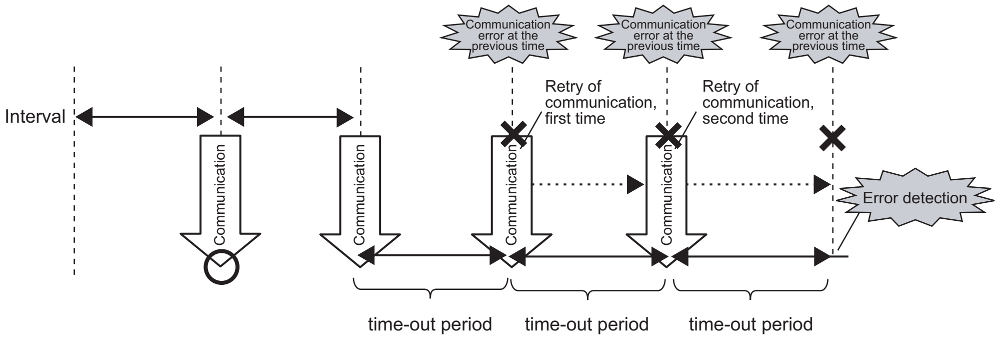
(10)Time-out Time and Retry Count

Setting item	Description	Setting range
Communication Time-out Time	Set the time interval until an error is detected or the retry of communication is performed when the destination device for communications does not respond or communications fail.	10ms to 65535ms (in increments of 1ms)[*1]
Communication Retry Count	Set the number of communication retry to be performed when the destination device for communications does not respond or communications fail.	0 to 255[*2]

[*1] When the destination device for communications is "MELSEC-A (Ethernet Module)/MELSEC-FX3 (Ethernet Block/ Adapter)", the value is fixed at 65535ms.

[*2] When the destination device for communications is "MELSEC-A (Ethernet Module)/MELSEC-FX3 (Ethernet Block/ Adapter)", the value is fixed at 0.

Ex. Error detection timing when 2 is set for Communication Retry Count


Communication error at the previous time
Communication error at the previous time
Communication error at the previous time
Interval
Retry of communication, first time
Retry of communication, second time
Communication
Communication
Communication
Communication
Error detection
time-out period
time-out period
time-out period

Point

The error is detected if the destination device does not respond or communications fail by when the following time has passed: (Communication Retry Count + 1) × Communication Time-out Time

(a) Precautions for retry of communication

- Communications at a specified execution interval (when "Fixed" is set for Communication Setting) or communications according to the on status of Request Contact (when "On Request" is set for Communication Setting) are ignored during retry of communication because that period is regarded as Communication Time-out Time.
- During retry of communication, the data at the time of communication error is sent to the destination device (for writing of device data) and the data at the time of the retry of communication is received (for reading of device data).

(11)Comment

Comment can be set by entering up to 32 one-byte characters (16 double-byte characters).

(12)Latency Time

By setting the latency time, the start timing in the communications can be switched in the following cases.

- To prevent an error due to the overlap of the start timing in the communications
- To start communications after the destination device is ready for the communications

During the latency time, "Preparing" is indicated in Communication status.

Setting item	Description	Setting range
Latency Time	Set the time required to start communications after completion of the CPU module startup.	0s to 255s (in increments of 1s)

Point

Communications at a specified execution interval (when "Fixed" is set for Communication Setting) or communications at the rising (off to on) of Request Contact (when "On Request" is set for Communication Setting) are ignored during the latency time.

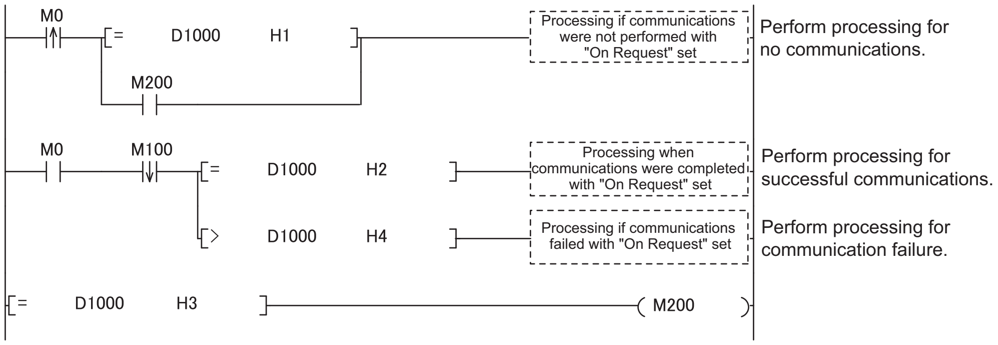
12.2 Program to Check Communications

This section shows the program used to check communications when "On Request" is set for Communication Setting.

(1) Devices used in programs

Device	Application
M0	Request Contact for setting No.1
M100	Execution Status Flag Device for setting No.1
M200	Flag to check if Request Contact turns on (when the status is "Communicating")
D1000	Status Save Destination Device for setting No.1

(2) Program example


M0
[= D1000 H1]
M200
Processing if communications were not performed with "On Request" set
M0
M100
[= D1000 H2]
Processing when communications were completed with "On Request" set
[> D1000 H4]
Processing if communications failed with "On Request" set
[= D1000 H3]
M200

Perform processing for no communications.

Perform processing for successful communications.

Perform processing for communication failure.

12.3 Diagnostics

The communication status of the simple PLC communication function can be checked in the "Simple PLC Communication Status" tab of the "Ethernet Diagnostics" window.

🖱 [Diagnostics] ⇨ [Ethernet Diagnostics]

For details on Ethernet diagnostics, refer to the following.

📖 GX Works2 Version 1 Operating Manual (Common)

12.4 Errors Related to the Simple PLC Communication Function

If an error related to this function occurs, it is not regarded as a diagnostic error in the CPU module. Instead the simple PLC communication status (the corresponding error code) is stored.

(1) Errors caused by the operation status of the CPU module or the operations of each function

The errors are not stored in the history of Ethernet diagnostics.

Error code	Error item	Error timing	Description	Action
4900_H	Other errors	When the parameters are written to the CPU module	After the values in the "Device" setting of PLC parameter of the CPU module, where the simple PLC communication function had been set, were changed, the parameters were written to the CPU module.	Power off and on or reset the CPU module.
4901_H		When an error related to the simple PLC communication function occurs (at corresponding setting No.)	The file register used for the simple PLC communication function became out of range due to the block number change of the file register.	Correct the device number of the file register, and then power off and on or reset the CPU module.
4902_H		When an error related to the simple PLC communication function occurs (at the other setting No.)	The communication was stopped because an error occurred at the other setting No., for which the same destination device of the corresponding setting No. had been set.	Eliminate the error cause.
4903_H		When a stop error occurs	The communication was stopped because a stop error occurred in the CPU module where the simple PLC communication function had been set.	Power off and on or reset the CPU module.

(2) Errors caused by communication failure

The errors are stored in the history of Ethernet diagnostics. The error codes to be stored are the same as those of communication related errors. For error descriptions and the corrective actions, refer to the following.

📖 MELSEC-L CPU Module User's Manual (Hardware Design, Maintenance and Inspection)

(3) Operations after the error

After the occurrence of the error, if the conditions are restored for proper communications, data communications are re-started at the following timing.

- When "Fixed" is set for Communication Setting: At the timing of the execution interval
- When "On Request" is set for Communication Setting: At the rising (off to on) of Request Contact

12.5 Precautions

This section provides precautions for the simple PLC communication function.

(1) The timing when the settings are enabled

The settings for this function are enabled at the following timing.

- When powering off and on the CPU module
- When resetting the CPU module

Even if the values of the settings are changed and the CPU module state is changed from STOP to RUN during execution of this function, the CPU module keeps operating with the previous settings. Note that if the "Device" setting of PLC parameter is changed and the parameters are written to the CPU module, an error (error code: 4900_H) occurs while the parameters are being written.

(2) Communications with other CPU modules

The communications are possible only between the CPU modules connected over Ethernet. (Communications with the other CPU modules cannot be performed through the CPU module connected over Ethernet.)

(3) Device data

If the COM instruction is used, not only the data in the END processing but also the data during sequence scan (at the execution of the COM instruction) are subjected to communications. Therefore, if you want to use exact data in the END processing for the program, conduct following operations.

[Transmission source] For the program, use a different device from the one specified for this function. At the end of the program (just before the END), reflect the data of that device to the one specified for this function.

[Transmission destination] At the beginning of the program, reflect the data of the device specified for this function to the other device, and use that device for the program.

(4) Incomplete data reception and timeout

Because the load of Ethernet communication becomes heavy during the simple PLC communication, if the other communication (MELSOFT connection or MC protocol) is simultaneously performed by using protocol UDP, the data reception of UDP may not be completed, resulting in a timeout error. Therefore, to perform other communications during the simple PLC communication, the communication using TCP is recommended. Also conduct following operations to reduce Ethernet communication load.

- Increase the execution interval of the simple PLC communication.
- Reduce the number of device points for communication.

(5) Communication stop

For a certain setting No., if the device is waiting for the response of the destination device due to power off of the module, cable disconnection, or power off of the hub, communications for the other setting No. might be stopped for 1000ms.

(6) Errors at the destination device

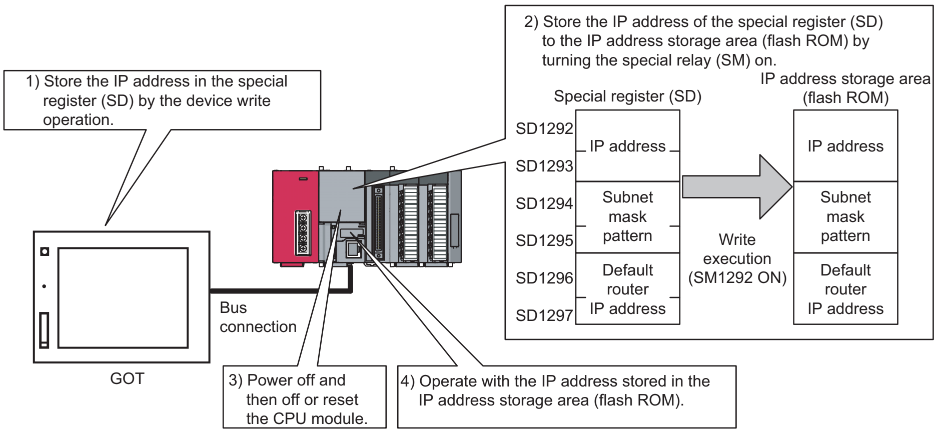
If an error related to this function occurs, following errors might be detected at the destination device.

- When the destination device is Q/L series: Remote password mismatch error
- When the destination device is A series: Device number specification error

Also, an error might be detected depending on the settings of the destination device or the conditions of Ethernet communication.

CHAPTER 13 IP ADDRESS CHANGE FUNCTION

This function changes the IP address of a built-in Ethernet port without changing the built-in Ethernet port settings in PLC parameter, by storing the values in the special relay and special register.


1) Store the IP address in the special register (SD) by the device write operation.
2) Store the IP address of the special register (SD) to the IP address storage area (flash ROM) by turning the special relay (SM) on.
Special register (SD)
IP address storage area (flash ROM)
SD1292
SD1293
SD1294
SD1295
SD1296
SD1297
IP address
Subnet mask pattern
Default router IP address
Write execution (SM1292 ON)
IP address
Subnet mask pattern
Default router IP address
Bus connection
GOT
3) Power off and then off or reset the CPU module.
4) Operate with the IP address stored in the IP address storage area (flash ROM).

Point

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

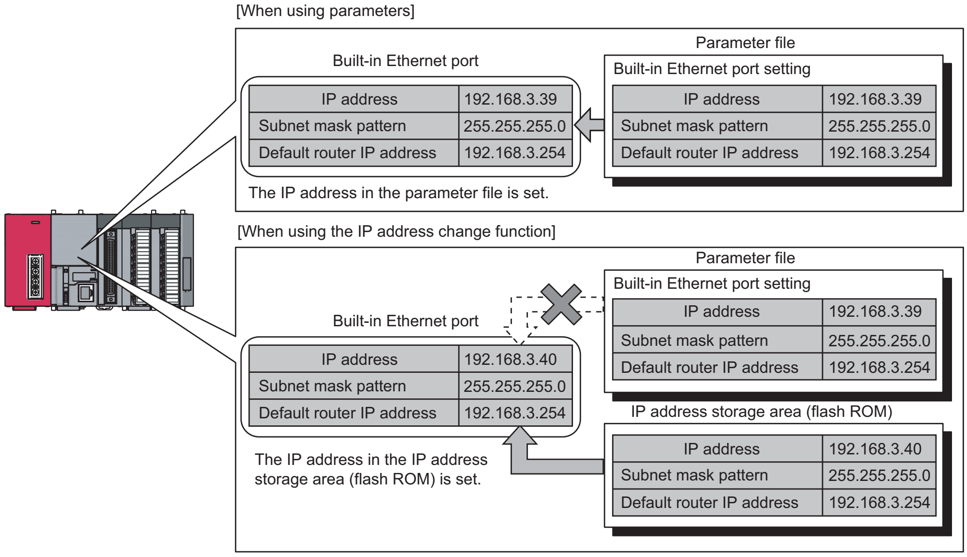
MELSEC-L CPU Module User's Manual (Hardware Design, Maintenance and Inspection)

Remark

Check the version of the CPU module when using the IP address change function. (Page 222, Appendix 3)

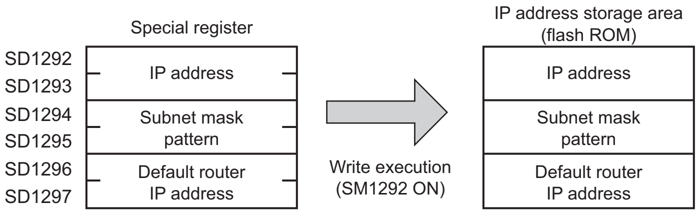
13.1 IP Address of the Built-in Ethernet Port

For IP address of the built-in Ethernet port, a value of the built-in Ethernet port setting in 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.


[When using parameters]
Built-in Ethernet port
IP address | 192.168.3.39
Subnet mask pattern | 255.255.255.0
Default router IP address | 192.168.3.254
Parameter file
Built-in Ethernet port setting
IP address | 192.168.3.39
Subnet mask pattern | 255.255.255.0
Default router IP address | 192.168.3.254
The IP address in the parameter file is set.
[When using the IP address change function]
Parameter file
Built-in Ethernet port setting
IP address | 192.168.3.39
Subnet mask pattern | 255.255.255.0
Default router IP address | 192.168.3.254
Built-in Ethernet port
IP address | 192.168.3.40
Subnet mask pattern | 255.255.255.0
Default router IP address | 192.168.3.254
IP address storage area (flash ROM)
IP address | 192.168.3.40
Subnet mask pattern | 255.255.255.0
Default router IP address | 192.168.3.254
The IP address in the IP address storage area (flash ROM) is set.

(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.


Special register
SD1292
SD1293
IP address
SD1294
SD1295
Subnet mask pattern
SD1296
SD1297
Default router IP address
Write execution (SM1292 ON)
IP address storage area (flash ROM)
IP address
Subnet mask pattern
Default router IP address

(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.

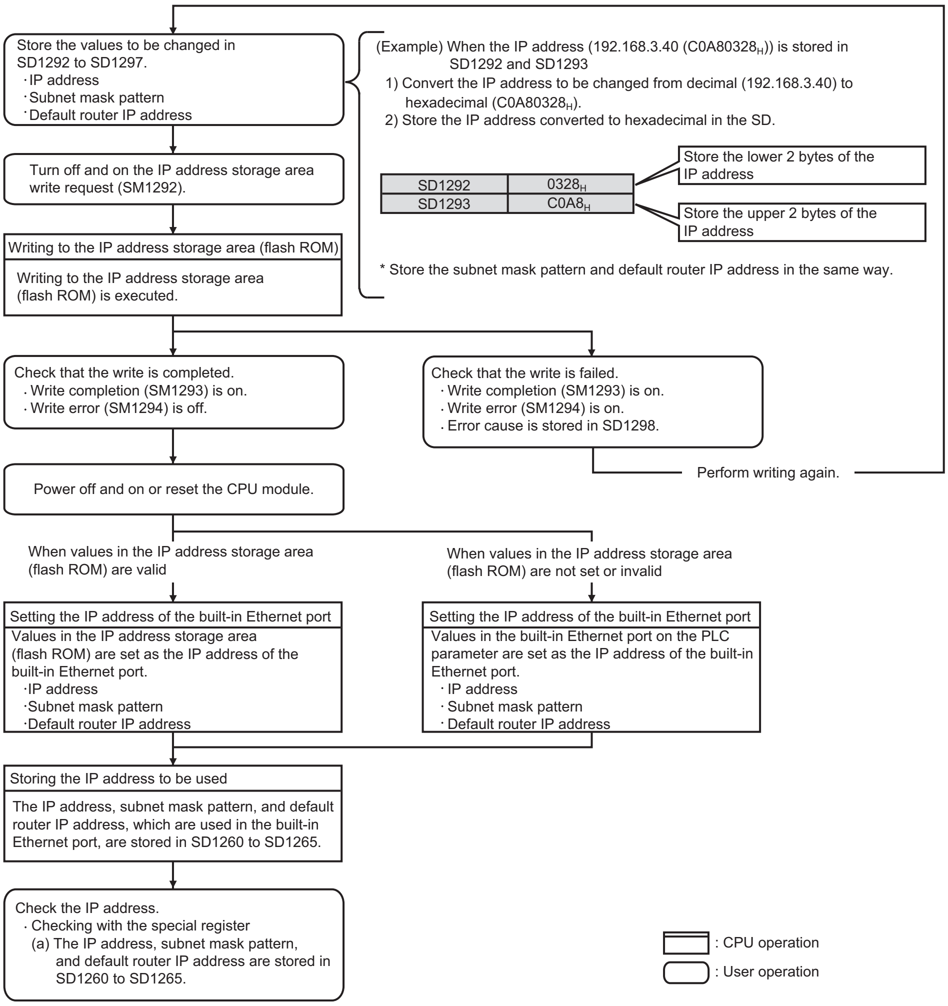
13.2 How to Use the Function

13.2.1 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.



Store the values to be changed in SD1292 to SD1297.
- IP address
- Subnet mask pattern
- Default router IP address

(Example) When the IP address (192.168.3.40 (C0A80328$_H$)) is stored in SD1292 and SD1293
1) Convert the IP address to be changed from decimal (192.168.3.40) to hexadecimal (C0A80328$_H$).
2) Store the IP address converted to hexadecimal in the SD.

SD1292	0328$_H$
SD1293	C0A8$_H$

Store the lower 2 bytes of the IP address
Store the upper 2 bytes of the IP address

* Store the subnet mask pattern and default router IP address in the same way.

Turn off and on the IP address storage area write request (SM1292).

Writing to the IP address storage area (flash ROM)

Writing to the IP address storage area (flash ROM) is executed.

Check that the write is completed.
- Write completion (SM1293) is on.
- Write error (SM1294) is off.

Check that the write is failed.
- Write completion (SM1293) is on.
- Write error (SM1294) is on.
- Error cause is stored in SD1298.

Power off and on or reset the CPU module.

When values in the IP address storage area (flash ROM) are valid

Setting the IP address of the built-in Ethernet port

Values in the IP address storage area (flash ROM) are set as the IP address of the built-in Ethernet port.
- IP address
- Subnet mask pattern
- Default router IP address

When values in the IP address storage area (flash ROM) are not set or invalid

Setting the IP address of the built-in Ethernet port

Values in the built-in Ethernet port on the PLC parameter are set as the IP address of the built-in Ethernet port.
- IP address
- Subnet mask pattern
- Default router IP address

Storing the IP address to be used

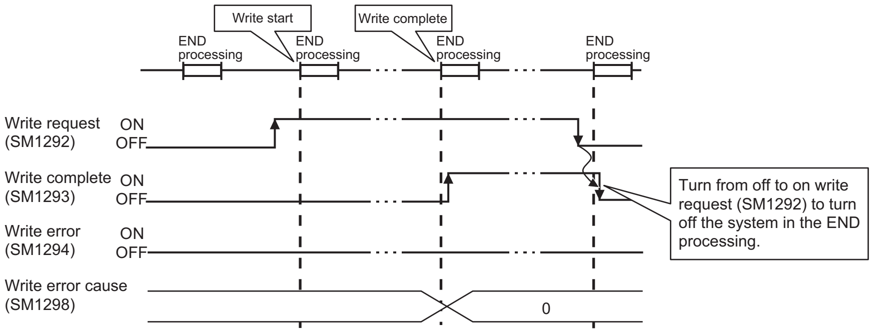
The IP address, subnet mask pattern, and default router IP address, which are used in the built-in Ethernet port, are stored in SD1260 to SD1265.

Check the IP address.
- Checking with the special register
  (a) The IP address, subnet mask pattern, and default router IP address are stored in SD1260 to SD1265.

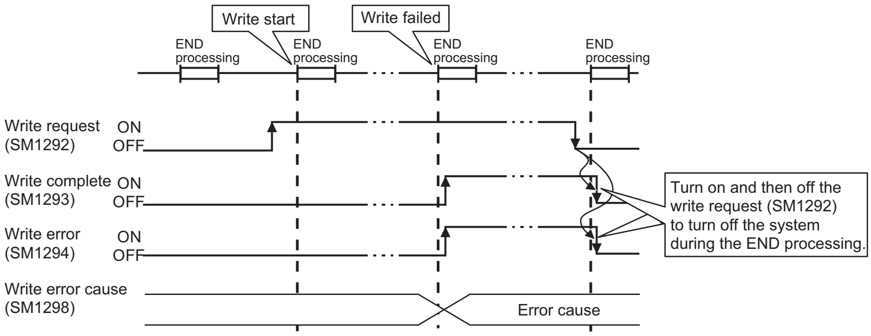
(2) Operations of special relay and special register

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. • Online program change • Copy program memory data into ROM • Write to PLC (flash ROM)
400_H	Writing is started during clear processing execution.

(4) Program example

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

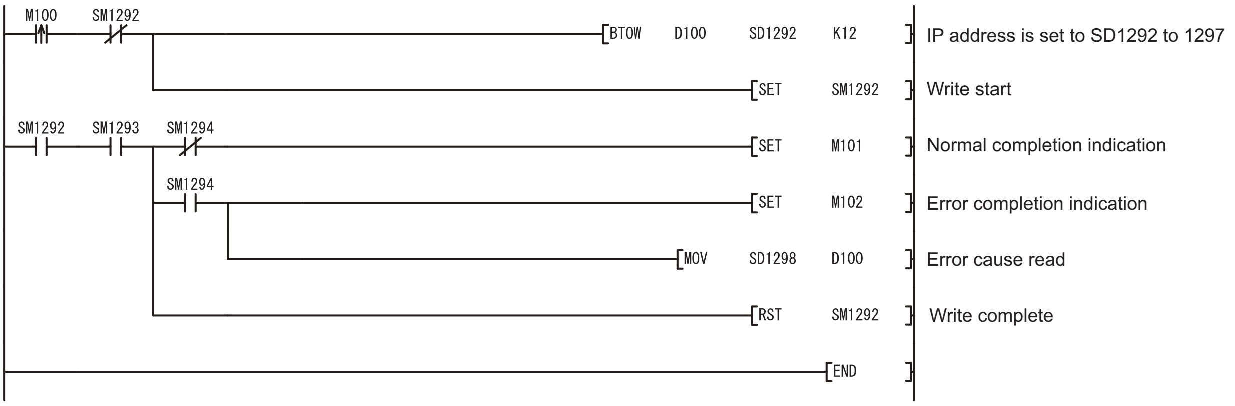
(a) Devices used in programs

Device number	Application	Device number	Application
M100	Write command	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

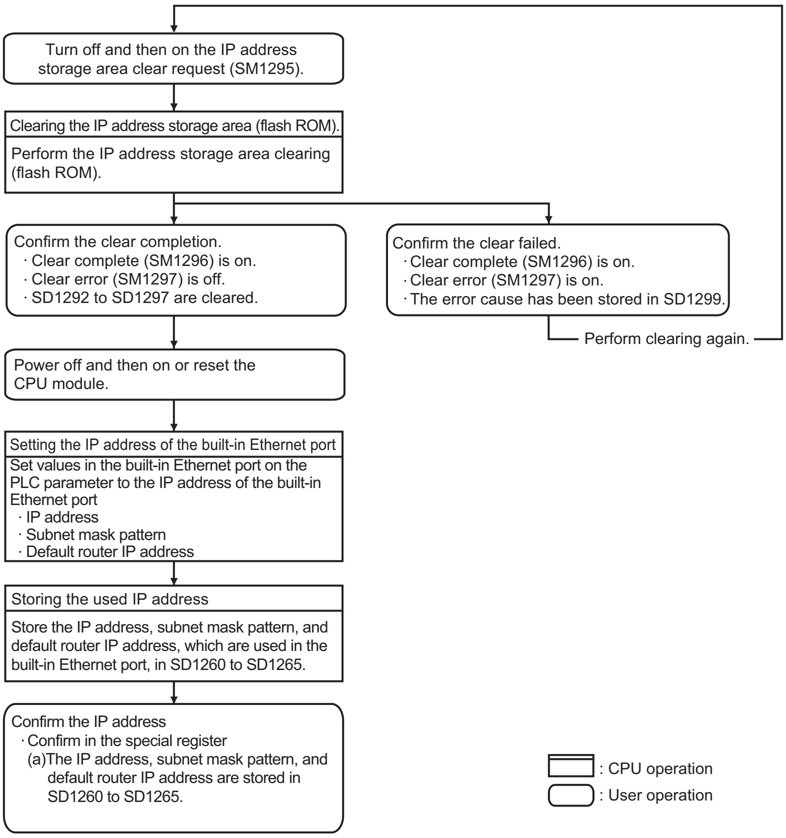


13.2.2 Clear operation

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

(1) Operating procedure

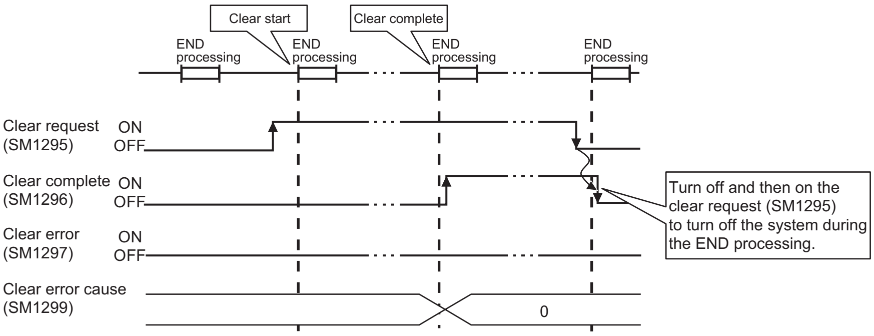
The following shows the clear operation flow.


Turn off and then on the IP address storage area clear request (SM1295).
Clearing the IP address storage area (flash ROM).
Perform the IP address storage area clearing (flash ROM).
Confirm the clear completion.
· Clear complete (SM1296) is on.
· Clear error (SM1297) is off.
· SD1292 to SD1297 are cleared.
Confirm the clear failed.
· Clear complete (SM1296) is on.
· Clear error (SM1297) is on.
· The error cause has been stored in SD1299.
Perform clearing again.
Power off and then on or reset the CPU module.
Setting the IP address of the built-in Ethernet port
Set values in the built-in Ethernet port on the PLC parameter to the IP address of the built-in Ethernet port
· IP address
· Subnet mask pattern
· Default router IP address
Storing the used IP address
Store the IP address, subnet mask pattern, and default router IP address, which are used in the built-in Ethernet port, in SD1260 to SD1265.
Confirm the IP address
· Confirm in the special register
(a)The IP address, subnet mask pattern, and default router IP address are stored in SD1260 to SD1265.
: CPU operation
: User operation

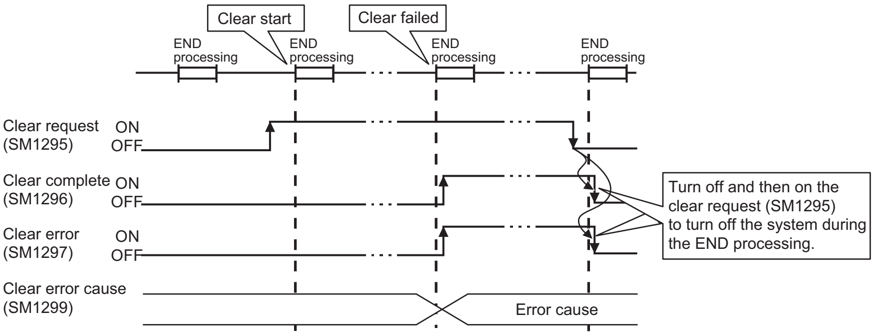
(2) Operations of special relay and special register

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. • Online program change • Copy program memory data into ROM • Write to PLC (flash ROM)
400_H	Clearing is started during write processing execution.

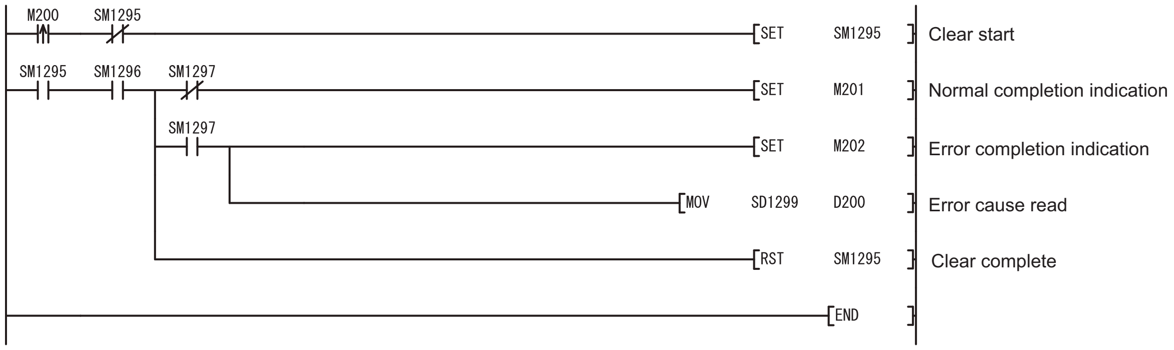
(4) Program example

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 command	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



13.3 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 on the Ethernet diagnostics, refer to the following.

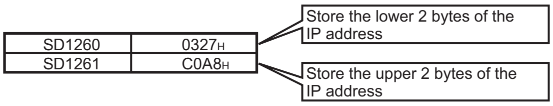
📖 GX Works2 Version 1 Operating Manual (Common)

(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.[*1]

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

[*1] Stored as follows.
Example: When IP address is 192.168.3.39 (C0A80327$_H$)



SD1260	0327H
SD1261	C0A8H

Store the lower 2 bytes of the IP address
Store the upper 2 bytes of the IP address

13.4 Precautions

The following describes precautions for the IP address change function.

(1) Power-off and reset operations

Do not power off or 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) turn on before powering off or 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
- Copy program memory data into ROM
- Write to PLC (flash ROM)

(5) Execution timing of write processing and clear processing

- Since the status of 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.

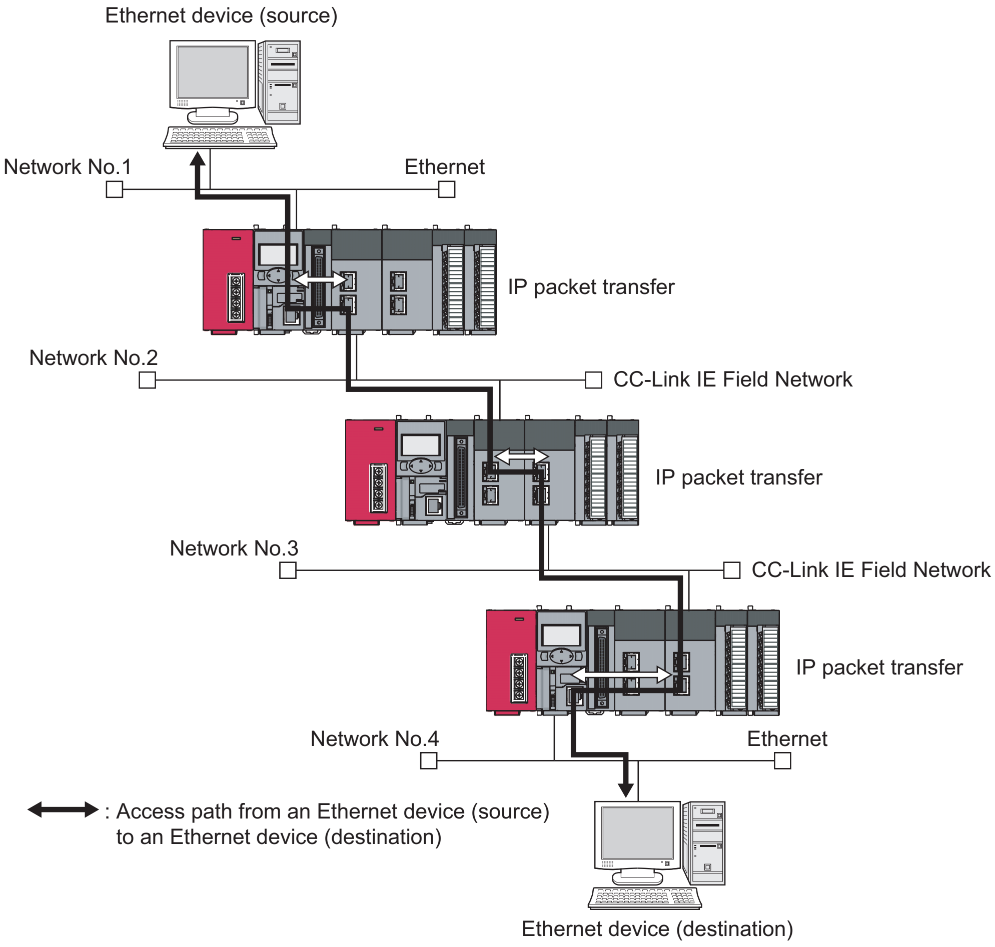
CHAPTER 14 IP PACKET TRANSFER FUNCTION

Remark

When using the IP packet transfer function, check the versions of the CPU module and GX Works2. (☞ Page 222, Appendix 3)

This 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 Field Network module.

- External devices on 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.)

(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 Field Network module used

(2) Precautions

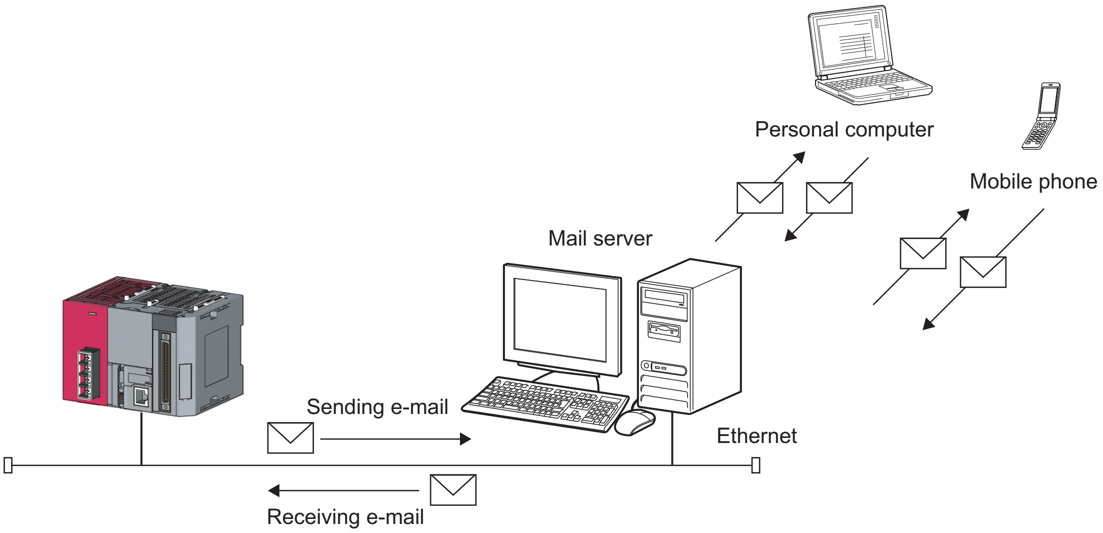
- The data that are communicated using the IP packet transfer function are communicated separately by the following processing.
    - · Service processing by a sequence scan of a CPU module
    - · Link scan on CC-Link IE Field Network

The above processing causes communication speeds to be slower than Ethernet lines. Manuals for the 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.

CHAPTER 15 E-MAIL SEND/RECEIVE FUNCTION

The e-mail send/receive function enables the CPU module to send e-mail to and receive it from electronic devices such as mobile phones and personal computers.



(1) Sending and receiving e-mail

(a) Sending e-mail

After setting the send destination e-mail address, the e-mail send instruction (SP.MLSEND) is executed to send e-mail during END processing.

(b) Receiving e-mail

The e-mail receive instruction (SP.MLRECV) is executed to get e-mail stored in the server to the CPU module. The "Subject + message body" of e-mail is written to the specified device.

(2) Function to read/write device when e-mail is received

Writing an MC protocol command in the message body of e-mail sent to the CPU module from a personal computer or mobile phone enables device reading or writing.

After command execution, e-mail is returned to the send address to report the execution result. (☞ Page 199, Section 15.3.1)

(3) Send destination e-mail address edit function

This function enables a remote mobile phone or personal computer to change the send destination e-mail address. (☞ Page 203, Section 15.3.2)

Remark

When using the e-mail send/receive function, check the versions of the CPU module and GX Works2. (☞ Page 222, Appendix 3)

15.1 E-mail Specifications

The e-mail specifications are listed below.

Item		Specifications
Data size	Attachment	Not permitted
	Message body	4096 words (8192 bytes)
Data transfer method	Send	Sending message body
	Receive	Receiving message body
Incoming mail format		Text format only
Subject		Us-ASCII or ISO-2022-JP (Base 64) format
MIME		Version 1.0
Send (encode)		• Subject: 7 bits • Message body: 7 bits
Receive (decode)		• Subject: 7 bits • Message body: 7 bits
Encryption		None
Compression		None
Communication with e-mail server		SMTP (send server) POP3 (receive server)
Port number		SMTP (send server): 1 to 65535 (variable) POP3 (receive server): 1 to 65535 (variable)
Operation check mailer		Microsoft®Outlook2010 (32-bit version) (Japanese edition)
Authentication function		Provided (SMTP authentication or POP before SMTP)

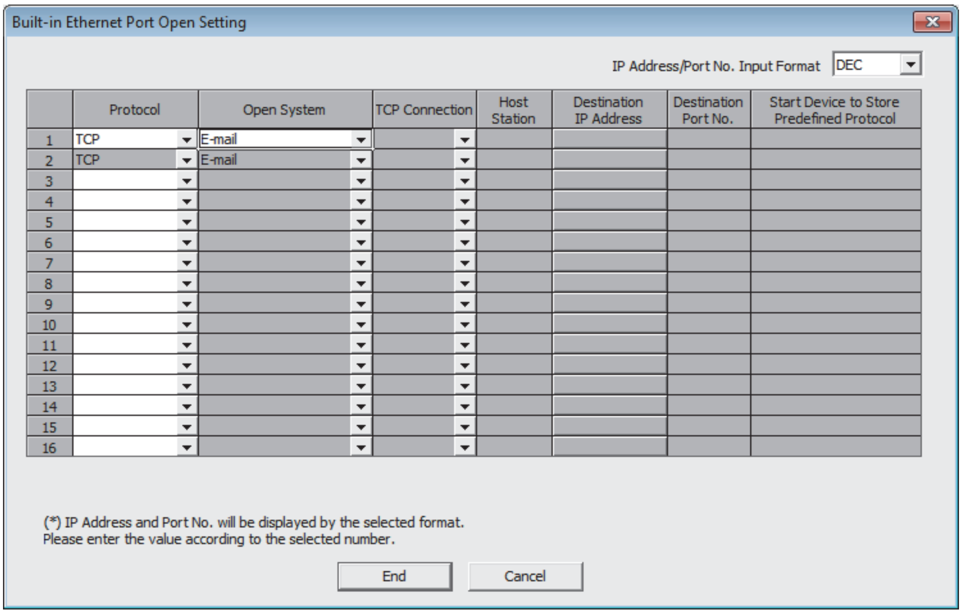
15.2 Setting Method

15.2.1 Parameter setting

(1) Open setting

Set the connections to be used by the e-mail send/receive function.

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



Built-in Ethernet Port Open Setting

IP Address/Port No. Input Format: DEC

	Protocol	Open System	TCP Connection	Host Station	Destination IP Address	Destination Port No.	Start Device to Store Predefined Protocol
1	TCP	E-mail					
2	TCP	E-mail					
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	Description
Protocol	Select "TCP".
Open System	Select "E-mail".

This function uses two connections. If "E-mail" is selected in the Open System, the Open System of the next connection No. is also changed to "E-mail".

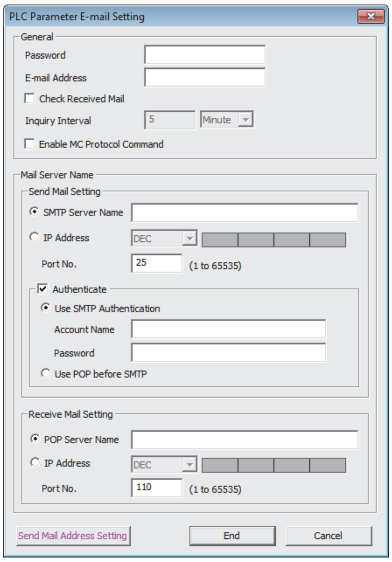
(Note that "E-mail" cannot be selected in connection No.16.)

"E-mail" can be set in Open System only in one connection.

(2) E-mail setting

Set the parameters of the e-mail send/receive function.

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



PLC Parameter E-mail Setting

General

Password

E-mail Address

□ Check Received Mail

Inquiry Interval 5 Minute

□ Enable MC Protocol Command

Mail Server Name

Send Mail Setting

● SMTP Server Name

○ IP Address DEC

Port No. 25 (1 to 65535)

☑ Authenticate

● Use SMTP Authentication

Account Name

Password

○ Use POP before SMTP

Receive Mail Setting

● POP Server Name

○ IP Address DEC

Port No. 110 (1 to 65535)

Item		Description	Setting range	Default
General	Password	Set the password for the mail server.	1 to 64 one-byte alphanumeric characters and symbols	—
	E-mail Address	Set the e-mail address of the CPU module.	1 to 256 one-byte alphanumeric characters and symbols	—
	Check Received Mail	Select this item to periodically query whether e-mail has reached the server. When e-mail has reached, SM1662 (received mail presence flag) turns on so that the CPU module can confirm that there is received mail.	Check or do not check for received mail.	Do not check for received mail.
	Inquiry Interval	Set the query time interval with the time unit for the receive mail server.	30 seconds to 24 hours	5 Minutes
	Enable MC Protocol Command	The command is executed if an MC protocol command is stated in the message body of received mail.	Allow or do not allow to receive MC protocol command mail.	Do not allow to receive MC protocol command mail.
Send Mail Setting	SMTP Server Name	Set the destination SMTP server name.	1 to 256 one-byte alphanumeric characters and symbols	—
	IP Address	Set the IP address of the destination SMTP server. The IP address can be specified with decimal or hexadecimal numbers.	0.0.0.1 to 223.255.255.254 (00.00.00.01 to DF.FF.FF.FE)	—
	Port No.	Set the port No. used to control the SMTP server.	1 to 65535	25
	Authenticate	Select whether to enable the server authentication function.	Enable or disable authentication.	Authenticate
	Authentication system	Select the authentication system.	• Use SMTP Authentication • Use POP before SMTP	Use SMTP Authentication
	Account Name	Set the account name of the SMTP server.	1 to 64 one-byte alphanumeric characters and symbols	—
	Password	Set the password used to log in to the SMTP server.	1 to 64 one-byte alphanumeric characters and symbols	—

Item		Description	Setting range	Default
Receive Mail Setting	POP Server Name	Set the POP server name of the receive destination.	1 to 256 one-byte alphanumeric characters and symbols	—
	IP Address	Set the IP address of the receive destination POP server. The IP address can be specified with decimal or hexadecimal numbers.	0.0.0.1 to 223.255.255.254 (00.00.00.01 to DF.FF.FF.FE)	—
	Port No.	Set the port No. used to control the POP server.	1 to 65535	110
Send Mail Address Setting		Set the e-mail address of the external device to which e-mail is to be sent.	Up to 16 addresses, each consisting of 1 to 256 one-byte alphanumeric characters and symbols	—

Point

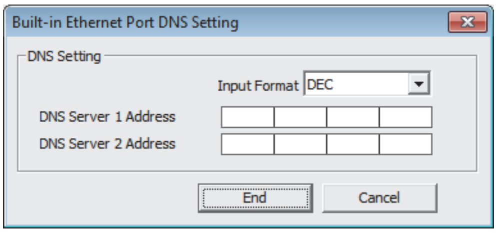
The e-mail addresses that can be set by this function can be stored in the following.

Item	Description	Setting method
PC parameter	• Used to send e-mail to a small number of destinations at a time. • An e-mail address can be registered directly in the send destination e-mail address setting number.	Register in "Send Mail Address Setting" of PC parameters.
CPU module internal memory	• Used to send e-mail to multiple destinations at a time. • Addresses can be dynamically registered or changed from the remote location by sending e-mail stating the MC protocol command to write e-mail addresses to devices.	Register by executing the SP.MLOPEADR instruction.

(3) DNS Setting

Set the IP address of the DNS server.

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



Input Format DEC

Item	Description	Setting range	Default
DNS Server 1 Address	Set the IP address of DNS server 1. The IP address can be specified with decimal or hexadecimal numbers.	0.0.0.1 to 223.255.255.254 (00.00.00.01 to DF.FF.FF.FE)	—
DNS Server 2 Address	Set the IP address of DNS server 2. The IP address can be specified with decimal or hexadecimal numbers.	0.0.0.1 to 223.255.255.254 (00.00.00.01 to DF.FF.FF.FE)	—

15.2.2 Settings at programming

This function sends and receives e-mail using the following instructions.

Instruction	Description	Reference
SP.MLSEND	Sends e-mail to the specified external device.	Page 204, Section 15.4.1
SP.MLRECV	Reads received e-mail from the mail server.	Page 208, Section 15.4.2

(1) Special relay and special register

The following table lists the special relay and special register used with this function.

For details, refer to the following.

📖 MELSEC-L CPU Module User's Manual (Hardware Design, Maintenance and Inspection)

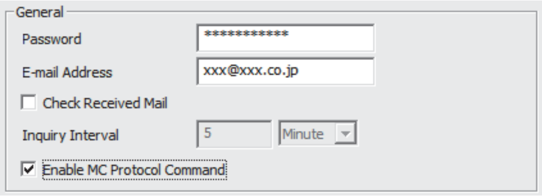
Number	Name	Description
SM1659	MC protocol command being processed flag	• This relay turns on while the MC protocol command is being processed. • If the received mail is an MC protocol command mail, the relay turns off during the END processing of the scan where the command is processed and the return mail is sent.
SM1660	MC protocol command error flag	• This relay turns on when an MC protocol command error is detected. The relay also turns on when the MC protocol command is disabled in parameter. • The relay turns off upon execution of the e-mail receive instruction.
SM1661	E-mail address backup processing flag	• This relay turns on when e-mail address backup starts (when an address is obtained by the e-mail address acquisition instruction (SP.MPGETADR)). • The relay turns off during the END processing after completion of e-mail address backup processing.
SM1662	Incoming mail check flag	• This relay turns on when there is incoming mail on the server when checked. ("Check Received Mail" needs to be set in advance in the e-mail setting.) • The relay turns off during the END processing of the scan where the e-mail receive instruction processing completes.
SD1659	Language information	Set the language information of the content of send e-mail (Subject + message body). • 0 or other than below: English/Japanese (default) • 2: Chinese (simplified Chinese characters)
SD1660	MC protocol command error code	• An error code is stored when an MC protocol command error is detected. An error code is also stored when the MC protocol command is disabled by the parameter. • A value, 0, is set at execution of the e-mail receive instruction.

15.3 Function Details

15.3.1 Function to read/write device when e-mail is received

Writing an MC protocol command in the message body of incoming mail enables device reading and writing. After command execution, e-mail is returned to the send address to report the result of reading.

When using this function, check "Enable MC Protocol Command" in the built-in Ethernet port setting of the PC parameter. (☞ Page 196, Section 15.2.1 (2))



General

Password	************
E-mail Address	xxx@xxx.co.jp

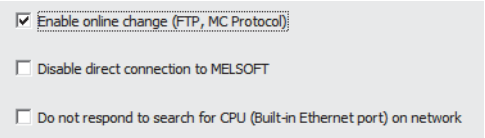
□ Check Received Mail

Inquiry Interval	5	Minute

☑ Enable MC Protocol Command

Check also "Enable online change (FTP, MC Protocol)".

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

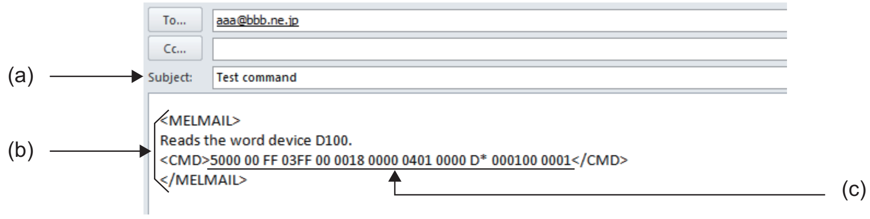


☑ Enable online change (FTP, MC Protocol)

□ Disable direct connection to MELSOFT

□ Do not respond to search for CPU (Built-in Ethernet port) on network

(1) Request e-mail format



To... aaa@bbb.ne.jp

Subject: Test command

```
<MELMAIL>
Reads the word device D100.
<CMD>5000 00 FF 03FF 00 0018 0000 0401 0000 D* 000100 0001</CMD>
</MELMAIL>
```

(a) Subject

Specify any subject.

(b) Message body

State the start/end of e-mail and the start/end of a command by enclosing the relevant item with the following header and footer.

Header/footer name	Header	Footer	Description
Start/end	<MELMAIL>	</MELMAIL>	Indicates the start and end of command e-mail. Anything stated before <MELMAIL> or after </MELMAIL> is invalid.
Command	<CMD>	</CMD>	The data written between the header and footer is assumed as a command.

(c) Data format

Command messages are compatible with the format supported by QnA-compatible 3E frame of the MC protocol used in the CPU module. Only application data needs to be specified; no header (for TCP/IP or UDP/IP) is required.

The frame format is shown below.

Sub-header	Header						Command	Subcommand	Request data part
	Network module number	PC number	Request destination module I/O number	Request destination module station number	Request data length	CPU monitoring timer			
5000_H (Fixed value)	00_H (Fixed value)	FF_H (Fixed value)	$03FF_H$ (Fixed value)	00_H (Fixed value)	Variable value	0000_H (Fixed value)	Variable value	Variable value	Variable value

The applicable commands are as follows.

Type	Operation	Command
Device memory	Batch read	0401
	Batch write	1401
	Random read	0403
	Test (random write)	1402

If a command other than the above (such as monitor) is received, e-mail of abnormal response is returned. (☞ Page 202, Section 15.3.1 (3))

For details of the data format, refer to the following.

📖 MELSEC Communication Protocol Reference Manual

(d) How to use

- Write a start header at the beginning of the message body.
- Write an end header at the end of the message body.
- Anything other than the start/end and command headers and footers can be written in any format.
- Only "alphanumeric characters" and "one-byte spaces" can be input between the command header and footer.
- Only one command can be stated in a single command mail.
- The maximum size of one command is 1024 bytes.
- The data code is fixed to ASCII, and the "Communication Data Code" in the built-in Ethernet port setting is ignored.

(2) Response mail format (normal response)

If the content of the received e-mail satisfies the request e-mail format, the following message is written as a normal response.

(a) Subject

"Re:" is automatically added to the beginning of the command e-mail received by the CPU module.
If adding "Re:" causes the command e-mail to exceed 128 one-byte characters (64 two-byte characters), the last three one-byte characters (two two-byte characters) are deleted.

(b) Message body

State a response message by enclosing it with the following start header and end footer.

Header/footer name	Header	Footer	Description
Start/end	<MELRET>	</MELRET>	Indicates the start and end of MC protocol response message.

(c) Data format

The frame format of the response mail is shown below.

- Ended normally

Sub-header	Header						Response data part
	Network module number	PC number	Request destination module I/O number	Request destination module station number	Response data length	End code	
D000$_H$ (Fixed value)	00$_H$ (Fixed value)	FF$_H$ (Fixed value)	03FF$_H$ (Fixed value)	00$_H$ (Fixed value)	Variable value	0000$_H$ (Fixed value)	Variable value

- Ended with an error

Sub-header	Header						Error information part					
	Network module number	PC number	Request destination module I/O number	Request destination module station number	Request data length	End code	Network module number	PC number	Request destination module I/O number	Request destination module station number	Command	Subcommand
D000$_H$ (Fixed value)	00$_H$ (Fixed value)	FF$_H$ (Fixed value)	03FF$_H$ (Fixed value)	00$_H$ (Fixed value)	Variable value	Variable value*1	00$_H$ (Fixed value)	FF$_H$ (Fixed value)	03FF$_H$ (Fixed value)	00$_H$ (Fixed value)	Variable value	Variable value

*1 An error code is stored in the end code. For details of errors, refer to the following.
☞Page 42, Section 5.1.4

(d) Command examples

A response mail is returned in response to the request mail requesting to read word device D100.

- Ended normally (1995$_H$ is assumed stored in D100.)

Subject	Message body
Re: Test command	<MELRET>D000 00 FF 03FF 00 0014 0000 1995</MELRET>

- Ended with an error

Subject	Message body
Re: Test command	<MELRET>D000 00 FF 03FF 00 0014 C051 00 FF 03FF 00 0401 0000</MELRET>

(3) Response mail format (abnormal response)

If the content of the received e-mail fails to satisfy the request e-mail format, the following message is written as an abnormal response.

(a) Subject

"Re:" is added to the beginning of the received command e-mail.
If adding "Re:" causes the command e-mail to exceed 128 one-byte characters (64 two-byte characters), the last three one-byte characters (two two-byte characters) are deleted.

(b) Message body

The following description is sent as response mail.

Response mail
The command was not able to be executed due to the following errors. Check the contents of e-mail and send it again. • Format error • Data size error • Data count error • Unsupported command error
Since communications using the MC protocol was not permitted, the command was not able to be executed. Check the parameters and send e-mail again.

(c) Precautions

If one of the following command mails is received, an error mail is sent.

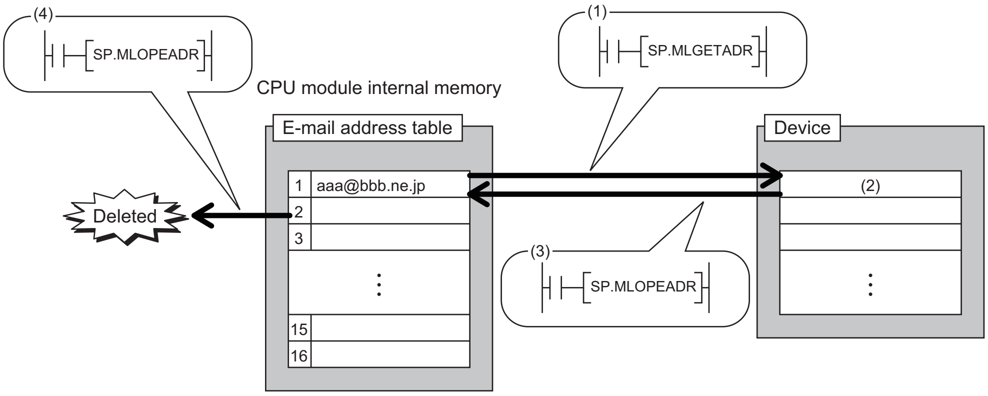
- The start and end header and footer of a command or e-mail are unpaired.
- An abnormal value is stored in the frame subheader.
- The command size exceeds the maximum size (1024 bytes).
- Two or more commands exist in one mail.
- An incompatible command is received.

15.3.2 Send destination e-mail address edit function

This function can edit, register, and delete e-mail addresses in the e-mail address table stored in the internal memory of the CPU module.

Point

Separately from the send e-mail address setting registered by PC parameters, this function targets 16 storage areas in the internal memory of the CPU module.



(1) When editing an e-mail address, issue the SP.MLGETADR instruction to obtain the e-mail address.
(2) Edit the obtained e-mail address.
(3) When registering an e-mail address, issue the SP.MLOPEADR instruction to register the target e-mail address.
(4) When deleting an e-mail address, issue the SP.MLOPEADR instruction to delete the target e-mail address.

The send-destination e-mail address edit function uses the following instructions.

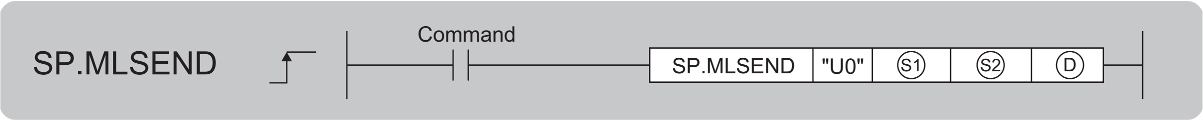
Instruction	Description	Reference
SP.MLOPEADR	Registers or deletes a send-destination e-mail address.	Page 213, Section 15.4.3
SP.MLGETADR	Stores a send destination e-mail address in the specified device.	Page 216, Section 15.4.4

Ex. Operation for changing the e-mail address (aaa@bbb.ne.jp) of e-mail address setting No.1

Step	Operation	E-mail address of setting No.1
1	Send an e-mail with the MC protocol command to instruct the CPU module to write the e-mail address to the device.	aaa@bbb.ne.jp
2	Execute the e-mail receive instruction (SP.MLRECV). (The MC protocol command writes the e-mail address to the specified device.)	aaa@bbb.ne.jp
3	Execute e-mail address table registration/deletion instruction (SP.MLOPEADR).	aaa@bbb.ne.jp,ccc@bbb.ne.jp
4	Execute e-mail address table registration/deletion instruction (SP.MLOPEADR). (Deletes the old e-mail address (aaa@bbb.ne.jp) remaining in setting No.1.)	ccc@bbb.ne.jp

15.4 E-mail Send/Receive Function Instructions

15.4.1 Send e-mail (SP.MLSEND)



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

*1 Local devices cannot be used.

(1) Setting data

Setting data	Description	Set by	Data type
U0	Dummy	—	Character string
Ⓢ1	Start number of the device from which control data are stored	User, system	Device name
Ⓢ2	Start number of the device in which the content of the send e-mail (Subject + message body) is stored	User	Device name
Ⓓ	Start number of the device which turns on for one scan upon completion of the instruction Ⓓ+1 also turns on when the instruction is completed with an error.	System	Bit

(2) Control data

Device	Item	Description	Setting range	Set by
Ⓢ1+0	Completion type, send data format, and destination e-mail address specification	b15 b14 b13 b12 b11 … b8 b7 b6 … b0 Ⓢ1+0 [3] 0 [2] 0 [1] 0 [1] Completion type Set whether clock data setting is required at completion with an error. • 0: Clock data setting not required Clock data at occurrence of an error is not set in Ⓢ1+11 to Ⓢ1+15. • 1: Clock data setting required Clock data at occurrence of an error is set in Ⓢ1+11 to Ⓢ1+15. [2] Send data format*1 Specify the format of the send data. • 1: Binary data [3] Destination e-mail address specification Specify the destination e-mail address used. • 0: E-mail address that is set by the PC parameter • 1: E-mail address registered in CPU internal memory	See the left column.	User

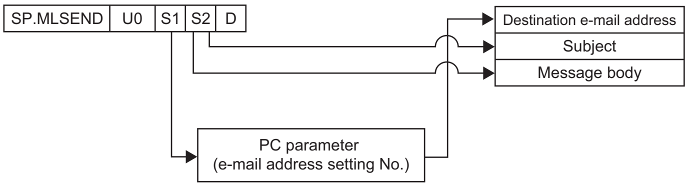
Device	Item	Description	Setting range	Set by
S1+1	Completion status	Completion status is stored. • 0000_H: Completed normally • Other than 0000_H: Failed (error code)	—	System
S1+2	Destination number	Specify the external device to send e-mail to, with the destination e-mail address setting number that is set by the PC parameter or in the CPU internal memory. • 1 to 16: Destination external device setting number	1 to 16	User
S1+3 to S1+8	System area	Use prohibited	—	System
S1+9	Sent data length	Specify the data length (Subject + message body) of the e-mail stored in S2+0 to S2+n. (Subject: 0 to 373, Message body: 1 to 4096) • 1 to 4469: E-mail data length (word)	1 to 4469	User
S1+10	Subject length	Specify the data length of the Subject of the e-mail stored in S2+0 to S2+n. • 0 to 373: Subject data length (word)	0 to 373	User
S1+11	Clock setting flag	The valid/invalid status of the clock data in S1+12 to S1+15 is stored. • 0: Invalid • 1: Valid	—	System
S1+12	Clock data (Set only when abnormal)	Upper 8 bits: Month (01_H to 12_H) Lower 8 bits: Last two digits of year (00_H to 99_H)	—	System
S1+13		Upper 8 bits: Hour (00_H to 23_H) Lower 8 bits: Day (01_H to 31_H)		
S1+14		Upper 8 bits: Second (00_H to 59_H) Lower 8 bits: Minute (00_H to 59_H)		
S1+15		Upper 8 bits: First two digits of year (00_H to 99_H) Lower 8 bits: Day of the week (00_H (Sun) to 06_H (Sat))		

[*1] Set parameters by observing the following.

- Specify the message body in ASCII code in the sequence program.
- Use "$0D0A_H$" and "0000_H" as control codes. ($0D0A_H$: return code (CR+LF), 0000_H: end of message body)
- One-byte katakana characters cannot be used.
- The number of characters of the data written to one line of the message body should be 78 or less.
- Be sure to attach return code CR+LF ($0D0A_H$) to the end of a line of the message body.

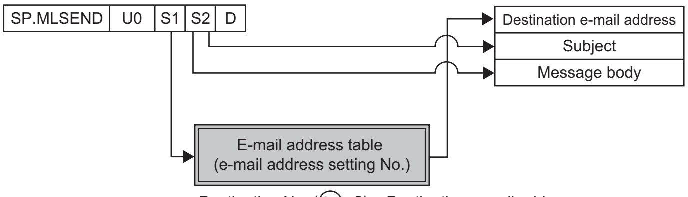
(3) Function

- Sends e-mail to the specified external device.
- S1S1When "E-mail address set by PC parameter" is specified in the destination e-mail address specification in S1+0, e-mail is sent to the e-mail address registered in e-mail address setting No. (S1+2) of the PC parameter.



Destination No. (S1+2) = Destination e-mail address

- Ⓢ1Ⓢ1When "E-mail address stored in CPU internal memory" is specified in the destination e-mail address specification in Ⓢ1+0, e-mail is sent to the e-mail address registered in e-mail address send destination No. (Ⓢ1+2) stored in CPU internal memory.



Destination No. (Ⓢ1+2) = Destination e-mail address

- Specify the content (Subject + message body) of the send e-mail specified in Ⓢ2 in ASCII code.
- The range of Ⓢ2 is determined by Ⓢ1+9 (send data length) and Ⓢ1+10 (Subject length).
- The Subject length specified in Ⓢ1+10 is in word units. Therefore, when the Subject string specified in Ⓢ2 is an odd number, store 00_H (NULL) in upper 8 bits of the end device number.

Ex. Subject is "abcde".

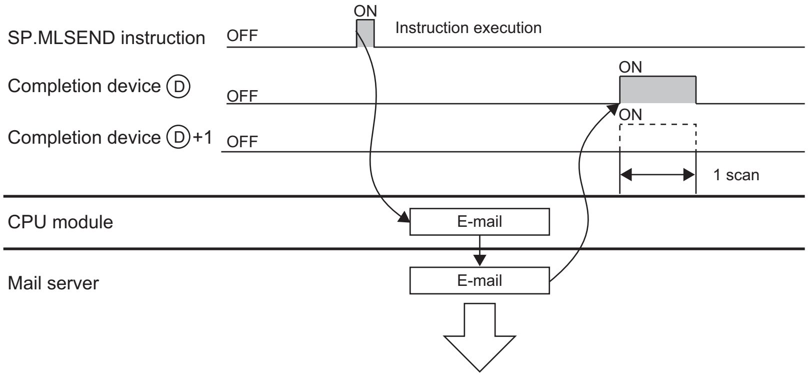
	b15 … b8	b7 … b0
Ⓢ2	62_H (b)	61_H (a)
Ⓢ2+1	64_H (d)	63_H (c)
Ⓢ2+2	00_H (NULL)	65_H (e)
Ⓢ2+3	2nd character in message body	1st character in message body
	⋮	

Subject data length

- The result of the SP.MLSEND instruction can be checked with the completion device, Ⓓ+0 and Ⓓ+1.
  - Completion device Ⓓ+0: This device turns on during END processing of the scan where the SP.MLSEND instruction completes, and turns off during the next END processing.
  - Completion device Ⓓ+1: This device turns on or off according to the result of the SP.MLSEND instruction.

State	Description
When completed normally	Remains off.
When completed with an error	Turns on during END processing of the scan where the SP.MLSEND instruction completes, and turns off during the next END processing. Ⓢ1When completed with an error, an error code is stored in the completion status of Ⓢ1+1. Also, when bit 7 of Ⓢ1+0 is on, clock data is stored in Ⓢ1+12 to Ⓢ1+15.

- The SP.MLSEND instruction runtime operation is as follows.



- If a e-mail address setting number in which no e-mail address is registered is specified, the instruction is completed with an error.

(4) Error

Each of the following events results in an operation error, and the error flag (SM0) turns on and an error code is stored in SD0.

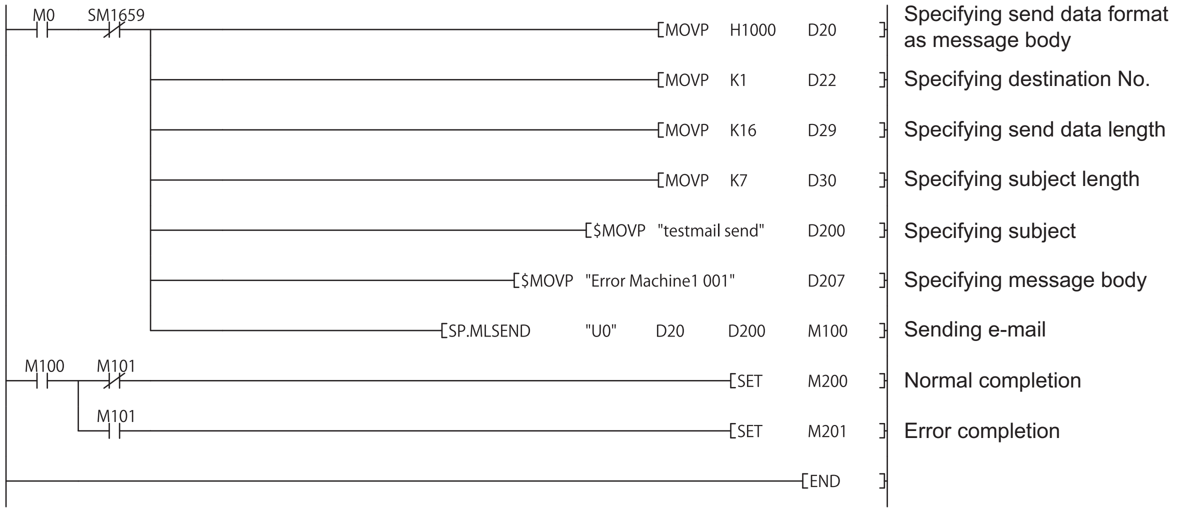
- An instruction is used in a CPU module with a serial number (first five digits) of "16111" or earlier.
  (Error code: 4002)
- The Subject length specified in Ⓢ1+10 exceeds the send data length specified in Ⓢ1+9.
  (Error code: 4100)
- The send data length, message body, or Subject length specified in Ⓢ1+9 or Ⓢ1+10 is out of the specified range.
  (Error code: 4100)
- The device specified by Ⓢ1 or Ⓓ exceeds the range of the number of device points.
  (Error code: 4101)
- The SP.MLSEND instruction is executed with "e-mail function" not set in the open method of the PC parameter.
  (Error code: 4160)

(5) Precautions

- If an instruction is executed while another e-mail send/receive instruction is already being executed or while SM1659 (MC protocol command being processed flag) is on, the instruction executed later performs no processing.
- Set SD1660 (e-mail language information) before instruction execution.
- When executing the SP.MLSEND instruction, check in advance the destination e-mail address. If the external device cannot be found due to an incorrect e-mail address, the SP.MLRECV instruction is executed for the destination mail server to cause an error and confirm the error event.

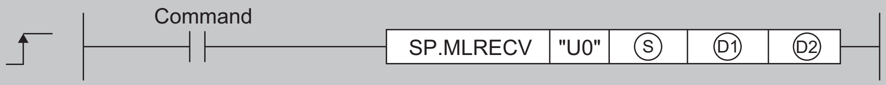
(6) Program example

The sample program sends the e-mail (Subject + message body) stored in D200 and later, based on the control data stored in D20 to D35.


M0
SM1659
[MOVP H1000 D20]
Specifying send data format as message body
[MOVP K1 D22]
Specifying destination No.
[MOVP K16 D29]
Specifying send data length
[MOVP K7 D30]
Specifying subject length
[$MOVP "testmail send" D200]
Specifying subject
[$MOVP "Error Machine1 001" D207]
Specifying message body
[SP.MLSEND "U0" D20 D200 M100]
Sending e-mail
M100
M101
[SET M200]
Normal completion
M101
[SET M201]
Error completion
[END]

15.4.2 Send e-mail (SP.MLRECV)

SP.MLRECV



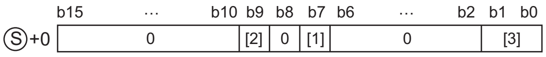
Setting data	Internal device		R, ZR	J□\□		U□\G□	Zn	Constant	Others
	Bit	Word		Bit	Word				
Ⓢ	—	○*1	○*1	—					
Ⓓ1	—	○*1	○*1	—					
Ⓓ2	○*1	—	○*1	—					

*1 Local devices cannot be used.

(1) Setting data

Setting data	Description	Set by	Data type
U0	Dummy	—	Character string
Ⓢ	Start number of the device from which control data are stored	User, system	Device name
Ⓓ1	Start device of own station in which the received e-mail (header + message body) is stored.	System	Device name
Ⓓ2	Start number of the device which turns on for one scan upon completion of the instruction Ⓓ2+1 also turns on when the instruction is completed with an error.	System	Bit

(2) Control data

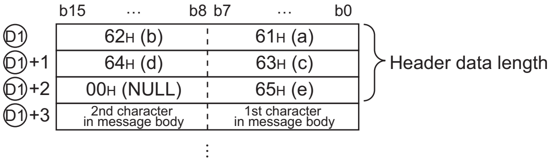
Device	Item	Description	Setting range	Set by
Ⓢ+0	Execution type/completion type	 b15 … b10 b9 b8 b7 b6 … b2 b1 b0 Ⓢ+0 0 [2] 0 [1] 0 [3] [1] Completion type Set whether clock data setting is required at completion with an error. • 0: Clock data setting not required Clock data at occurrence of an error is not set in Ⓢ+11 to Ⓢ+15. • 1: Clock data setting required Clock data at occurrence of an error is set in Ⓢ+11 to Ⓢ+15. [2] Execution type Specify whether to query server storage e-mail after reading the received mail. • 0: Do not query (Received mail information is not read.) • 1: Query (Received mail information is read.) [3] Received mail type Specify the combination type of received mail. • 11: Receive only message body.	See the left column.	User
Ⓢ+1	Completion status	Completion status is stored. • 0000_H: Completed normally • Other than 0000_H: Failed (error code)	—	System

Device	Item	Description	Setting range	Set by
Ⓢ+2	E-mail number to be read	Specify the number of the e-mail to be read when multiple e-mails are received. • 0: First e-mail • 1 to 65535: Specified e-mail	0 to 65535	User
Ⓢ+3 to Ⓢ+8	System area	Use prohibited	—	System
Ⓢ+9	Receive data length	■**Before instruction execution** Specify the data length (header + message body) of the e-mail that can be stored in Ⓓ1+0 to Ⓓ1+n. (Header: 1 to 373, Message body: 1 to 4096) • 0: Follow the data length of the received e-mail. • 1 to 4469: Data length of the received e-mail that can be stored ■**At completion of instruction** The data length (header + message body) of the e-mail stored in Ⓓ1+0 to Ⓓ1+n is stored. • 1 to 4469: Data length of the received e-mail that can be stored	1 to 4469	User
Ⓢ+10	Header length	■**Before instruction execution** Specify the header data length of the e-mail that can be stored in Ⓓ1+0 to Ⓓ1+n. • 0: Follow the header data length of the received mail. • 1 to 373: Header data length of the stored received mail ■**At completion of instruction** The header data length of the e-mail that can be stored in Ⓓ1+0 to Ⓓ1+n is stored. • 1 to 373: Header data length of the stored received mail	0 to 373	User
Ⓢ+11	Clock setting flag	The valid/invalid status of the clock data in Ⓢ+12 to Ⓢ+15 is stored. • 0: Invalid • 1: Valid	—	System
Ⓢ+12	Clock data (Set only when abnormal)	Upper 8 bits: Month (01_H to 12_H) Lower 8 bits: Last two digits of year (00_H to 99_H)	—	System
Ⓢ+13		Upper 8 bits: Hour (00_H to 23_H) Lower 8 bits: Day (01_H to 31_H)		
Ⓢ+14		Upper 8 bits: Second (00_H to 59_H) Lower 8 bits: Minute (00_H to 59_H)		
Ⓢ+15		Upper 8 bits: First two digits of year (00_H to 99_H) Lower 8 bits: Day of the week (00_H (Sun) to 06_H (Sat))		

(3) Function

- Reads received e-mail from the mail server. The received mail is deleted from the mail server.
- If e-mail longer than the receive data length (Ⓢ+9) specified by the control data of the SP.MLRECV is received, the data of the receive data length is stored and the remaining data is ignored.
- The header string of the header + message body stored in Ⓓ1 is an odd number, 00_H (NULL) is stored in the upper 8 bits of the end device number.

Ex. When the header is "abcde"

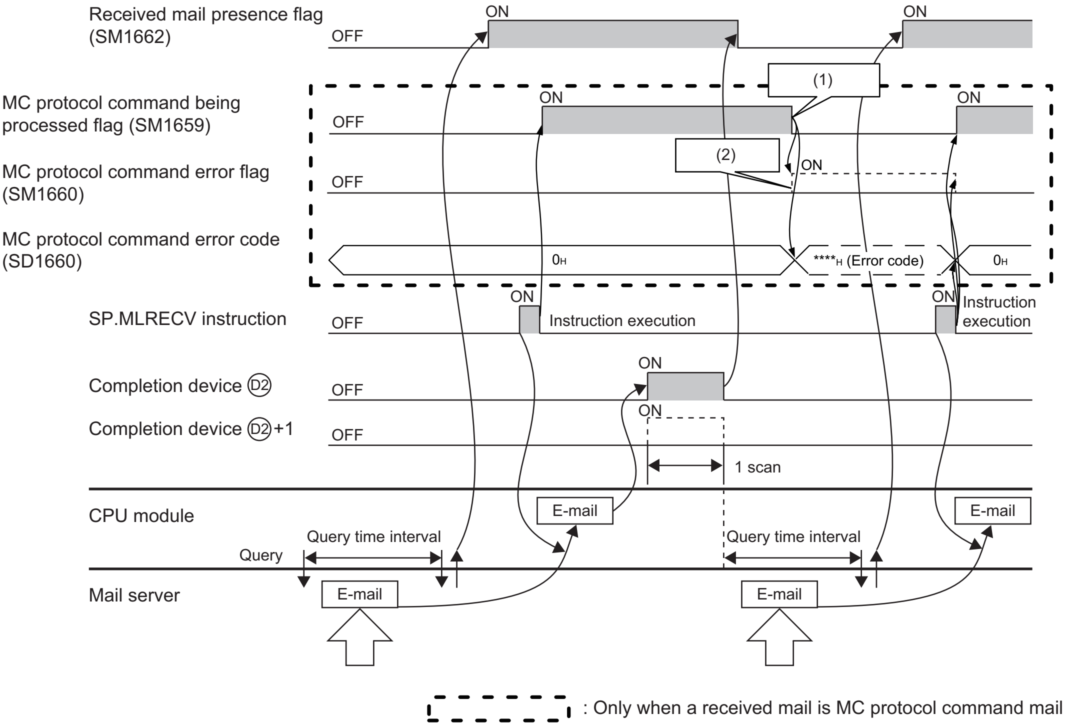


	b15 … b8	b7 … b0	
D1	62H (b)	61H (a)	Header data length
D1+1	64H (d)	63H (c)	
D1+2	00H (NULL)	65H (e)	
D1+3	2nd character in message body	1st character in message body	
⋮			

- When the received mail is an MC protocol command mail, SM1659 (MC protocol command being processed flag) turns on. It turns off during the END processing where the return mail is sent after MC protocol command processing.
- If the SP.MLRECV instruction is executed while SM1662 (received mail presence flag) is off, the instruction is completed with an error.
- When the received mail has an attachment, the attachment is not received but only the message body is received.
- When MC protocol command mail is faulty or the MC protocol command is completed with an error, SM1660 (e-mail function MC protocol command error flag) turns on and an error code is stored in SD1660 (e-mail function MC protocol command error code).
- The result of the SP.MLRECV instruction can be checked with the completion device, D2+0 and D2+1.
  - Completion device D2+0: This device turns on during END processing of the scan where the SP.MLRECV instruction completes, and turns off during the next END processing.
  - Completion device D2+1: This device turns on or off according to the result of the SP.MLRECV instruction.

State	Description
When completed normally	Remains off.
When completed with an error	Turns on during END processing of the scan where the SP.MLRECV instruction completes, and turns off during the next END processing. SWhen completed with an error, an error code is stored in the completion status of S+1. Also, when bit 7 of S+0 is on, clock data is stored in S+12 to S+15.

- The SP.MLRECV instruction runtime operation is as follows.



(1) MC protocol command mail processing completed

(2) This relay turns on when the MC protocol command is completed with an error.

(4) Error

Each of the following events results in an operation error, and the error flag (SM0) turns on and an error code is stored in SD0.

- An instruction is used in a CPU module with a serial number (first five digits) of "16111" or earlier.

  (Error code: 4002)

- The header length specified in Ⓢ+10 exceeds the receive data length specified in Ⓢ+9.

  (Error code: 4100)

- The receive data length, message body, or header length specified in Ⓢ+9 or Ⓢ+10 is out of the specified range.

  (Error code: 4100)

- The device specified by Ⓢ, Ⓓ1, or Ⓓ2 exceeds the range of the number of device points.

  (Error code: 4101)

- The SP.MLRECV instruction is executed with "e-mail function" not set in the open method of the PC parameter.

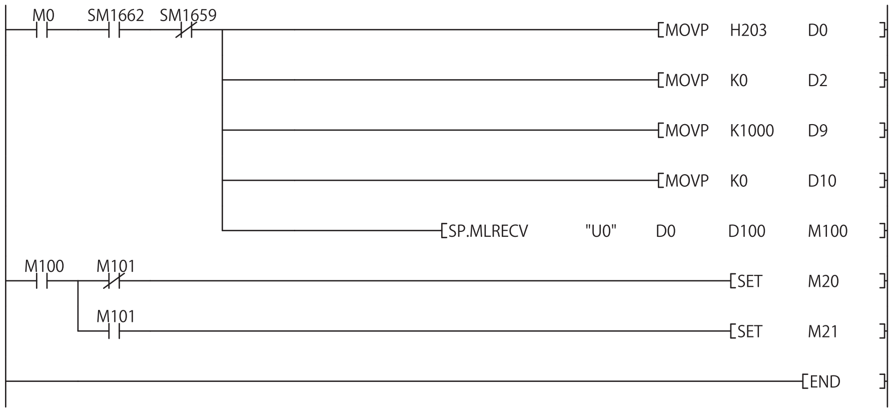
  (Error code: 4160)

(5) Precautions

- If an instruction is executed while another e-mail send/receive instruction is already being executed or while SM1659 (MC protocol command being processed flag) is on, the instruction executed later performs no processing.

(6) Program example

The sample program stores the e-mail (header + message body), which was received based on the control data stored in D0 to D15, in D100 and later.



Querying a server
Specifying received mail type as "Receive only message body"

Specifying to read the first e-mail

Specifying data length of received mail

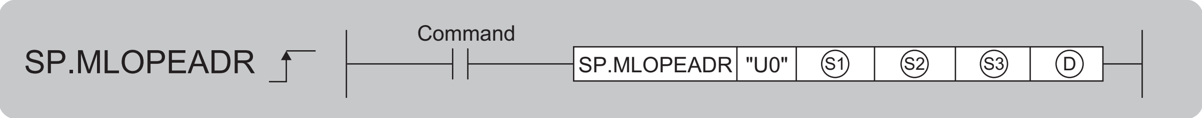
Specifying header data length of received mail

Reading a received mail

Normal completion

Error completion

15.4.3 E-mail address registration/deletion (SP.MLOPEADR)



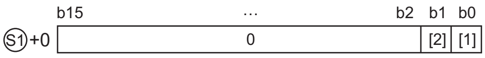
Setting data	Internal device		R, ZR	J□\□		U□\G□	Zn	Constant		Others
	Bit	Word		Bit	Word			K, H	$	
Ⓢ1	—	○*1	○*1	—				—	—	—
Ⓢ2	—	○	○	—				○	—	—
Ⓢ3	—	○	○	—				—	○	—
Ⓓ	○*1	—	○*1	—				—	—	—

*1 Local devices cannot be used.

(1) Setting data

Setting data	Description	Set by	Data type
U0	Dummy	—	Character string
Ⓢ1	Start number of the device from which control data are stored	User, system	Device name
Ⓢ2	Destination e-mail address setting No. (1 to 16)	User	BIN 16-bit
Ⓢ3	E-mail address string or start number (Number of characters specified for e-mail address: 1 to 500 one-byte characters) of the device in which the string is stored	User	Character string
Ⓓ	Start number of the device which turns on for one scan upon completion of the instruction Ⓓ+1 also turns on when the instruction is completed with an error.	System	Bit

(2) Control data

Device	Item	Description	Setting range	Set by
Ⓢ1+0	Operation processing setting area	Specify the operation processing.  b15 … b2 b1 b0 Ⓢ1+0 0 [2] [1] [1] Specify the execution type. • 0: Register • 1: Delete [2] Enable/disable backup. • 0: Enable backup. • 1: Disable backup.	See the left column.	User
Ⓢ1+1	Completion status	Completion status is stored. • 0000$_H$: Completed normally • Other than 0000$_H$: Failed (error code)	—	System

(3) Function

- This instruction registers/deletes the e-mail address specified in S3 in the destination e-mail address setting number specified in S2 of the e-mail address table. (The destination e-mail address setting number of the PC parameter is not applicable.)
- The execution type specified in S1+0 specifies whether to register or delete an e-mail address.
- To either register or delete an e-mail address with the SP.MLOPEADR instruction, the e-mail address needs to be specified in S3.
- The data from the device number specified in S3 to the device number in which "00_H" is stored is registered as an e-mail address.
- The characters that can be used for the e-mail address are "20_H" to "$7E_H$". (However, "$2C_H$" (comma) is excluded.) If any other character is used, the instruction is completed with an error.
- The e-mail address table allows multiple e-mail addresses to be registered in one setting number. If another e-mail address is registered in the setting number in which an e-mail address is already registered, a one-byte comma "," is automatically inserted as a delimiter between the e-mail addresses. (Up to 500 one-byte characters can be registered.)
- If delimiter "," exists in the string specified in S3, the instruction is completed with an error. (Two or more e-mail addresses cannot be registered or deleted simultaneously.)

Point

Note that if an already registered e-mail address is registered during e-mail address registration, two or more identical e-mail addresses come to exist. (Same e-mails by the number of registered e-mail addresses are delivered.)

- If the number of characters that can be registered is exceeded when registering the e-mail address, the instruction is completed with an error.
- If the specified e-mail address is not registered when deleting an e-mail address, the instruction is completed with an error.
- By performing the backup setting in S1+0, the e-mail addresses in the e-mail address table can be held even after the power is turned off. SM1661 is on during backup. The e-mail addresses are backed up to flash ROM and restored automatically during initialization.

Point

Do not perform resetting or turn off the power while SM1661 (e-mail address backup processing flag) is on (during backup). Doing so may corrupt the file or prevent the backup from being reflected. If the backup file is corrupted, it is deleted during restoration.

- The result of the SP.MLOPEADR instruction can be checked with the completion device, Ⓓ+0 and Ⓓ+1.
  - Completion device Ⓓ+0: This device turns on during END processing of the scan where the SP.MLOPEADR instruction completes, and turns off during the next END processing.
  - Completion device Ⓓ+1: This device turns on or off according to the result of the SP.MLOPEADR instruction.

State	Description
When completed normally	Remains off.
When completed with an error	Turns on during END processing of the scan where the SP.MLOPEADR instruction completes, and turns off during the next END processing. S1 When completed with an error, an error code is stored in the completion status of S1+1.

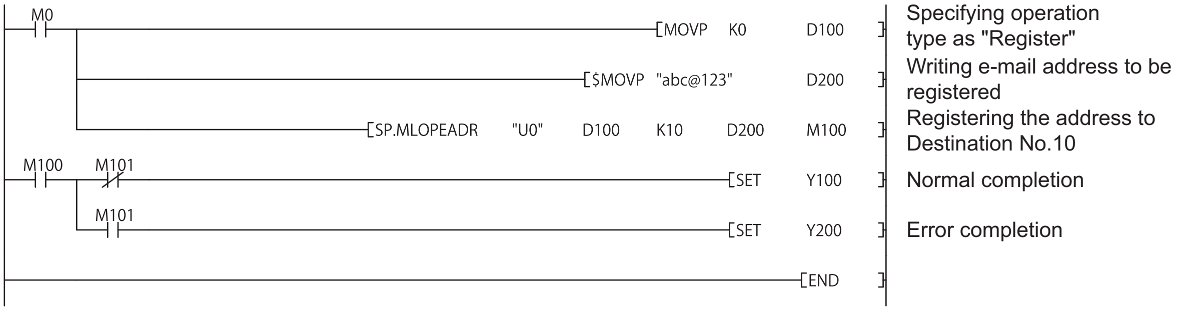
(4) Error

Each of the following events results in an operation error, and the error flag (SM0) turns on and an error code is stored in SD0.

- An instruction is used in a CPU module with a serial number (first five digits) of "16111" or earlier.
  (Error code: 4002)
- A device which cannot be specified is specified.
  (Error code: 4004)
- The device specified by S1 or D exceeds the range of the number of device points.
  (Error code: 4101)
- S2 is other than 1 to 16.
  (Error code: 4101)

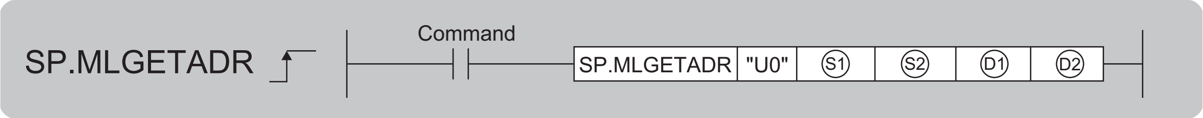
(5) Program example

This sample program stores the control data in D100 and D101, and adds the e-mail address of D200 to destination number 10.



M0
[MOVP K0 D100]
Specifying operation type as "Register"
[$MOVP "abc@123" D200]
Writing e-mail address to be registered
[SP.MLOPEADR "U0" D100 K10 D200 M100]
Registering the address to Destination No.10
M100 M101
[SET Y100]
Normal completion
M101
[SET Y200]
Error completion
[END]

15.4.4 E-mail address acquisition (SP.MLGETADR)



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

*1 Local devices cannot be used.

(1) Setting data

Setting data	Description	Set by	Data type
U0	Dummy	—	Character string
S1	Start number of the device from which control data are stored	User, system	Device name
S2	Destination e-mail address setting No. (1 to 16)	User	BIN 16-bit
D1	Start device of own station to store the obtained e-mail address	System	Device name
D2	Start number of the device which turns on for one scan upon completion of the instruction D2+1 also turns on when the instruction is completed with an error.	System	Bit

(2) Control data

Device	Item	Description	Setting range	Set by
S1+0	Destination e-mail address specification	Specify the destination e-mail address used. • 0: E-mail address that is set by the PLC parameter • 1: E-mail address registered in CPU internal memory	0, 1	User
S1+1	Completion status	Completion status is stored. • 0000_H: Completed normally • Other than 0000_H: Failed (error code)	—	System
S1+2	Device data length	■**Before instruction execution** Device data length that can be stored in D1+0 to D1+n. • 1 or higher: Device data length	1 to 501	User
		■**At completion of instruction** E-mail address data length stored in D1+0 to D1+n • 1 or higher: E-mail address data length + 1 ("NULL" is added to the end of the character string.)	—	System

(3) Function

- This instruction stores the e-mail address stored in the destination e-mail address setting number in Ⓢ2 selected by Ⓢ1+0 in the device specified by Ⓓ1 and later.
- If the e-mail address is not registered in the destination e-mail address setting number, "00_H" is stored in Ⓓ1.
- When two or more registered e-mail addresses are set in one destination setting number, "," is automatically inserted as a delimiter between the addresses.
- "00_H" (NULL) is stored at the end of the obtained character string.
- The result of the SP.MLGETADR instruction can be checked with the completion device, Ⓓ2+0 and Ⓓ2+1.
  - Completion device Ⓓ2+0: This device turns on during END processing of the scan where the SP.MLGETADR instruction completes, and turns off during the next END processing.
  - Completion device Ⓓ2+1: This device turns on or off according to the result of the SP.MLGETADR instruction.

State	Description
When completed normally	Remains off.
When completed with an error	Turns on during END processing of the scan where the SP.MLGETADR instruction completes, and turns off during the next END processing. When the instruction is completed with an error, an error code is stored in the completion status of Ⓢ1+1.

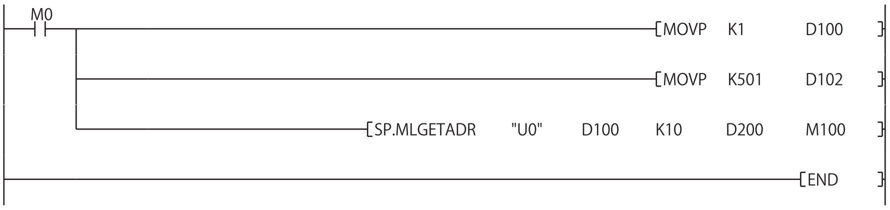
(4) Error

Each of the following events results in an operation error, and the error flag (SM0) turns on and an error code is stored in SD0.

- An instruction is used in a CPU module with a serial number (first five digits) of "16111" or earlier.
  (Error code: 4002)
- A device which cannot be specified is specified.
  (Error code: 4004)
- Ⓢ2 is other than 1 to 16.
  (Error code: 4101)
- The device specified by Ⓢ1, Ⓓ1, or Ⓓ2 exceeds the range of the number of device points.
  (Error code: 4101)

(5) Program example

This sample program stores control data in D100 and gets the e-mail address of destination setting No.10 of the parameter setting to D200.



Specifying destination e-mail address as "E-mail address registered in CPU module internal memory

Specifying device data length

Acquiring e-mail address of Destination No.10

15.5 Diagnostics

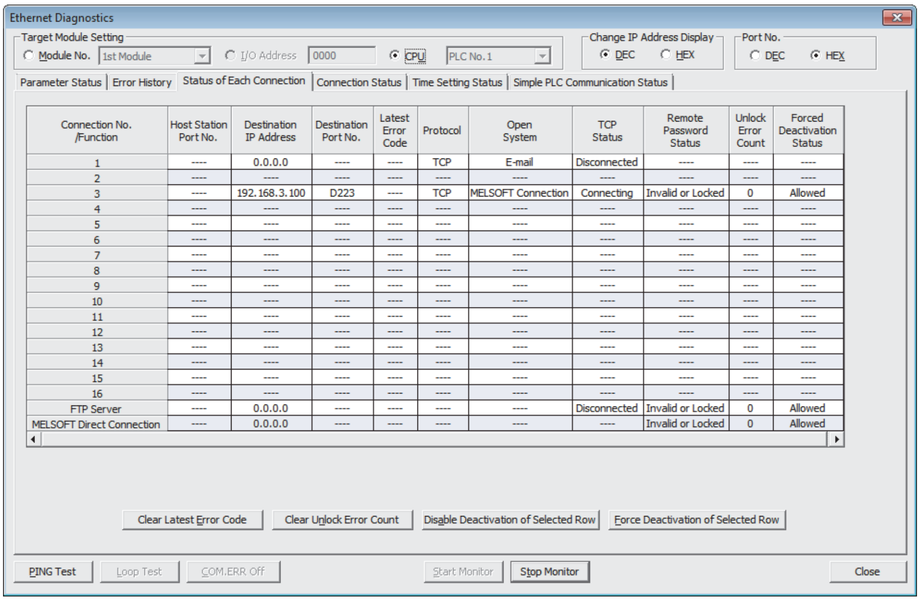
The communication status of the simple PLC communication function can be checked in the "Status of Each Connection" tab in the "Ethernet Diagnostics" window.

[Diagnostics] ⇨ [Ethernet Diagnostics]

For details on Ethernet diagnostics, refer to the following.

📖 GX Works2 Version 1 Operating Manual (Common)

The connection number in which the states and errors of the connections used by this function are stored is the one with the smaller number of the two connections that are set in "E-mail" in the open setting.



Ethernet Diagnostics

Target Module Setting: ○ Module No. 1st Module ○ I/O Address 0000 ● CPU PLC No.1

Change IP Address Display: ● DEC ○ HEX

Port No.: ○ DEC ● HEX

Connection No. /Function	Host Station Port No.	Destination IP Address	Destination Port No.	Latest Error Code	Protocol	Open System	TCP Status	Remote Password Status	Unlock Error Count	Forced Deactivation Status
1	----	0.0.0.0	----	----	TCP	E-mail	Disconnected	----	----	----
2	----	----	----	----	----	----	----	----	----	----
3	----	192.168.3.100	D223	----	TCP	MELSOFT Connection	Connecting	Invalid or Locked	0	Allowed
4	----	----	----	----	----	----	----	----	----	----
5	----	----	----	----	----	----	----	----	----	----
6	----	----	----	----	----	----	----	----	----	----
7	----	----	----	----	----	----	----	----	----	----
8	----	----	----	----	----	----	----	----	----	----
9	----	----	----	----	----	----	----	----	----	----
10	----	----	----	----	----	----	----	----	----	----
11	----	----	----	----	----	----	----	----	----	----
12	----	----	----	----	----	----	----	----	----	----
13	----	----	----	----	----	----	----	----	----	----
14	----	----	----	----	----	----	----	----	----	----
15	----	----	----	----	----	----	----	----	----	----
16	----	----	----	----	----	----	----	----	----	----
FTP Server	----	0.0.0.0	----	----	----	----	Disconnected	Invalid or Locked	0	Allowed
MELSOFT Direct Connection	----	0.0.0.0	----	----	----	----	----	Invalid or Locked	0	Allowed

APPENDICES

Appendix 1 Operation Processing Time for Each Instruction

The table below indicates the operation processing times required for the instructions introduced 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)			
				L02CPU, L02CPU-P		L06CPU, L06CPU-P, L26CPU, L26CPU-P, L26CPU-BT, L26CPU-PBT	
				Minimum	Maximum	Minimum	Maximum
Instructions for the socket communication function	SP.SOCOPEN	TCP	Active	27.20	58.90	23.50	45.50
			Unpassive				
			Fullpassive				
		UDP					
	SP.SOCCLOSE	TCP	From the host CPU	18.90	56.60	16.60	43.80
			From the connected device				
		UDP					
	SP.SOCRCV	TCP	1 byte	16.90	50.70	15.40	39.20
			2046 bytes				
		UDP	1 byte				
			2046 bytes				
	S.SOCRCVS	TCP	1 byte	16.30	46.30	16.00	35.20
			2046 bytes	504.30	532.60	394.20	413.20
		UDP	1 byte	16.30	46.30	16.00	35.20
			2046 bytes	504.30	532.60	394.20	413.20
	SP.SOCSND	TCP	1 byte	45.70	61.30	36.60	48.20
			2046 bytes	379.70	395.30	267.60	278.20
		UDP	1 byte	45.70	61.30	36.60	48.20
			2046 bytes	379.70	395.30	267.60	278.20
	SP.SOCCINF	—		17.10	43.70	14.70	32.30
	SP.SOCCSET	—		13.30	42.90	12.70	31.40
	SP.SOCRMODE	Standard mode → Fixed-length mode		16.00	39.60	13.70	29.30
		Fixed-length mode → Standard mode					
	SP.SOCRDATA	1 word		13.00	35.60	11.30	26.10
		1024 words		500.00	521.60	389.70	404.30
Instruction for the predefined protocol function	SP.ECPRTCL	—		54.00	123.00	43.50	90.50

Type	Instruction	Condition	Processing time (μs)			
			L02CPU, L02CPU-P		L06CPU, L06CPU-P, L26CPU, L26CPU-P, L26CPU-BT, L26CPU-PBT	
			Minimum	Maximum	Minimum	Maximum
File transfer function instructions	SP.FTPPUT	File name + server path string = 32 characters	251.000	256.000	234.000	244.000
	SP.FTPGET	File name + server path string = 32 characters	248.000	252.000	214.000	224.000
E-mail send/receive function instructions	SP.MLSEND	Send data length: 16 words	55.800	60.000	44.300	48.500
	SP.MLRECV	Receive data length: 16 words	54.600	58.700	43.800	47.100
	SP.MLOPEADR	E-mail address registration/deletion registration: 32 characters	51.200	58.700	39.300	45.400
	SP.MLGETADR	E-mail address acquisition	40.900	51.700	33.900	41.700
SLMP frame send instruction	SP.SLMPSND	Batch read in word units (command: 0401_H): Number of read points = 1 point	78.600	86.900	60.200	66.500

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

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 Q series Ethernet modules, this port number is used for "Auto Open UDP Port".)
1389_H (5001)	For future extension (For Q series Ethernet modules, this port number is used for "over UDP/IP and Ethernet module".)
$138A_H$ (5002)	For future extension (For Q series Ethernet modules, this port number is used for "over TCP/IP and Ethernet module".)
$138B_H$ (5003) to $138D_H$ (5005)	For future extension
$138E_H$ (5006)	MELSOFT communication port (over UDP/IP and CPU module)
$138F_H$ (5007)	MELSOFT communication port (over TCP/IP and CPU module)
1390_H (5008)	MELSOFT direct connection port (over CPU module)
1391_H (5009)	For future extension

Appendix 3 Added and Changed Functions

The following table shows the built-in Ethernet function, which is changed or added to the CPU module and GX Works2, as well as the serial number of the CPU module and the version of GX Works 2 that support the function.

Added function	Serial No. (first 5 digits) of CPU module	GX Works2 version	Reference
Simple PLC communication function	"13042" or later	1.62Q or later	Page 165, CHAPTER 12
IP packet transfer function[*1]	"14112" or later	1.98C or later	Page 191, CHAPTER 14
IP packet address change function	"15102" or later	—	Page 181, CHAPTER 13
Predefined protocol function	"15102" or later	1.501X or later	Page 53, CHAPTER 6
Simple PLC communication function (Support by F series FX3)	"16042" or later	1.513K or later	Page 165, CHAPTER 12
File transfer function (FTP client)	"16112" or later	1.525X or later	Page 139, CHAPTER 10
E-mail send/receive function			Page 193, CHAPTER 15
SLMP frame send instruction	"18112" or later	—	Page 43, Section 5.2

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

Appendix 4 Performance List of Simple PLC Communication Function

This section shows the performance of the execution interval (simple PLC communication function). The execution interval varies depending on the number of settings, the number of communication points, the scan time of the CPU module. The interval is also affected by the execution conditions of other functions or Ethernet communication condition.

(1) Condition 1

- Communication Setting: "Fixed Interval"
- Destination device: "MELSEC-Q/L (Built-in Ethernet Function)" = L26CPU-BT
- Sequence scan time (host station): 1ms (the period before the execution of simple PLC communication function)
- Sequence scan time (destination device): 1ms (The destination device means both the transmission source and transmission destination for data transfer.)
- Device data: Bit device = M, Word device = D
- Number of settings: The number of destination devices (Double the number for data transfer.)
- The retry of communication is not performed.

CPU module	Communication pattern	Number of communication points for one setting	Processing time by the number of settings (unit: ms)		
			1	8	16
L02CPU, L02CPU-P	Read	32 words for each*1 (Total 64 words)	10.0	18.1	30.7
		64 words for each*1 (Total 128 words)	10.0	18.2	31.0
		256 words for each*1 (Total 512 words)	10.0	37.8	—
	Write	32 words for each*1 (Total 64 words)	10.0	18.2	31.0
		64 words for each*1 (Total 128 words)	10.0	18.3	31.1
		256 words for each*1 (Total 512 words)	12.0	39.2	—
	Transfer	32 words for each*1 (Total 64 words)	14.0	27.3	48.6
		64 words for each*1 (Total 128 words)	14.0	27.3	48.6
		256 words for each*1 (Total 512 words)	20.0	58.0	—
L06CPU, L06CPU-P, L26CPU, L26CPU-P, L26CPU-BT, L26CPU-PBT	Read	32 words for each*1 (Total 64 words)	10.0	18.1	30.7
		64 words for each*1 (Total 128 words)	10.0	18.2	31.0
		256 words for each*1 (Total 512 words)	10.0	37.8	—
	Write	32 words for each*1 (Total 64 words)	10.0	18.2	31.0
		64 words for each*1 (Total 128 words)	10.0	18.3	31.1
		256 words for each*1 (Total 512 words)	12.0	39.2	—
	Transfer	32 words for each*1 (Total 64 words)	14.0	27.3	48.6
		64 words for each*1 (Total 128 words)	14.0	27.3	48.6
		256 words for each*1 (Total 512 words)	20.0	58.0	—

*1 Number of points for each device (bit device and word device)

(2) Condition 2

- Communication Setting: "Fixed Interval"
- Destination device: "MELSEC-Q/L (Built-in Ethernet Function)" = L26CPU-BT
- Sequence scan time (host station): 10ms (the period before the execution of simple PLC communication function)
- Sequence scan time (destination device): 1ms (The destination device means both the transmission source and transmission destination for data transfer.)
- Device data: Bit device = M, Word device = D
- Number of settings: The number of destination devices (Double the number for data transfer.)
- The retry of communication is not performed.

CPU module	Communication pattern	Number of communication points for one setting	Processing time by the number of settings (unit: ms)		
			1	8	16
L02CPU, L02CPU-P	Read	32 words for each[*1] (Total 64 words)	10.0	17.6	29.8
		64 words for each[*1] (Total 128 words)	11.0	17.9	30.2
		256 words for each[*1] (Total 512 words)	11.0	37.4	—
	Write	32 words for each[*1] (Total 64 words)	10.0	18.1	30.7
		64 words for each[*1] (Total 128 words)	11.0	18.2	31.0
		256 words for each[*1] (Total 512 words)	12.0	39.5	—
	Transfer	32 words for each[*1] (Total 64 words)	13.0	26.7	47.4
		64 words for each[*1] (Total 128 words)	14.0	26.9	47.9
		256 words for each[*1] (Total 512 words)	20.0	58.3	—
L06CPU, L06CPU-P, L26CPU, L26CPU-P, L26CPU-BT, L26CPU-PBT	Read	32 words for each[*1] (Total 64 words)	10.0	17.6	29.8
		64 words for each[*1] (Total 128 words)	11.0	17.9	30.2
		256 words for each[*1] (Total 512 words)	11.0	37.4	—
	Write	32 words for each[*1] (Total 64 words)	10.0	18.1	30.7
		64 words for each[*1] (Total 128 words)	11.0	18.2	31.0
		256 words for each[*1] (Total 512 words)	12.0	39.5	—
	Transfer	32 words for each[*1] (Total 64 words)	13.0	26.7	47.4
		64 words for each[*1] (Total 128 words)	14.0	26.9	47.9
		256 words for each[*1] (Total 512 words)	20.0	58.3	—

[*1] Number of points for each device (bit device and word device)

Appendix 5 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 LCPU and the Ethernet module (LJ71E71-100).

○: Available, △: Restricted, ×: Unavailable

Item			Description	Availability	
				Built-in Ethernet port LCPU	LJ71E71-100
MC protocol communication	4E frame		• 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	Reads/writes data (device) in the CPU module from/to an external device.	○*1*8	○
		Others	Reads/writes data (file) in the CPU module from/to an external device.	×	○
	A-compatible 1E frame		• Reads/writes data in the CPU module from/to an external device. • A frame format that is compatible with A series E71	×	○
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. • 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	△*11*12	○
Communications 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	○
File transfer (FTP client function)			The CPU module becomes an FTP client and can execute file transfer with the FTP server connected to the LAN using the file transfer function instruction.	○*11	×
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 the 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 LCPU	LJ71E71-100
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 connected 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		Searches 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		○	×
Simple PLC communication function		Allows data communications between specified devices at the specified timing just by doing simple settings from a programming tool.	○	×
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 Field Network module. • External devices on 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.	×	○

Item	Description	Availability	
		Built-in Ethernet port LCPU	LJ71E71-100
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]	○
SLMP frame send instruction	MC protocol messages (QnA-compatible 3E frame) can be sent from the CPU module to external devices on the Ethernet network.	○[11]	×

[1] Available commands are limited. (☞ Page 38, Section 5.1.2)

[2] The "quote cpuchg" command cannot be used. (☞ Page 127, 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 LJ71E71-100.

[7] Up to 17 devices can be connected (including one system connection) by setting "MELSOFT Connection" for user connections in network parameter.

[8] For processing on the external device side, refer to Page 228, Appendix 5 (2).

[9] Executable with the socket communication function. Refer to Page 229, Appendix 5 (3) for the differences. Before using the function, check the versions of the CPU module and programming tool used. (☞ Page 222, Appendix 3)

[10] Executable with the socket communication function. Before using the function, check the versions of the CPU module and programming tool used. (☞ Page 222, Appendix 3)

[11] Before using the function, check the versions of the CPU module and programming tool used. (☞ Page 222, Appendix 3)

[12] Executable with the E-mail send/receive function. E-mail cannot be sent by the CPU module monitoring function.

Remark

For Ethernet modules, refer to the following.

📖 MELSEC-L Ethernet Interface Module User's Manual (Basic)

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

Item	LJ71E71-100	Built-in Ethernet port LCPU	Consequence of a communication with the Built-in Ethernet port LCPU	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 41, Section 5.1 (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.)

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

Item	LJ71E71-100	Built-in Ethernet port LCPU	Consequence of a communication with the Built-in Ethernet port LCPU	Action
Instruction name	ZP.OPEN ZP.CLOSE ZP.BUFRCV Z.BUFRCVS ZP.BUFSND	SP.SOCOPEN SP.SOCCLOSE SP.SOCRCV S.SOCRCVS SP.SOCSND	—	Replace the instruction name.
Pairing open not required	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 LJ71E71-100 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 described in Page 84, 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	• Serial number (first 5 digits) is "12051" or earlier: 2046 bytes • Serial number (first 5 digits) is "12052" or later: 10238 bytes	—	—

Memo

INDEX

A

Active open 66

B

binary 130
bye 130

C

change 134
Class 33
close 130
CPU module. 17
CPU module dedicated subcommands 134

D

Data communication frame 40
delete 130
dir 130
Direct connection 31

E

E-mail send/receive function 193
Ethernet cable 20
Ethernet communication 22
External device. 17

F

File transfer function 119,139
FTP client. 139
FTP commands 127
FTP server. 119
FTP server support commands. 130
Fullpassive. 66

G

get. 130
GOT 22

H

Host name 26
Hosts file 26
Hub 20

K

KeepAlive. 29

L

LCPU 17
Lock processing 160
ls 131

M

MC protocol communication. 34
mdelete. 131
mdir 131
mget 131
mls 132
mput 132

N

NAK message 37

O

open 132

P

Passive open 66
passwd-rd 136
passwd-wr 136
password-lock 135
password-unlock 134
pm-write 136
Predefined protocol. 53
Programming tool 17
put 132
pwd. 133

Q

quit 133
quote. 133

R

recv function 41
rename 133
ret. 121
Router. 28
run 135

S

SD memory card 17
Searching CPU modules 27
Simple connection 31
Simultaneous broadcast 80
SNTP 17
SNTP client 116
Socket communication function instructions. 85
status 135
stop. 135
Subcommands 128

T

TCP 21
TCP fixed-length receive mode 112
TCP socket functions 41
TCP standard receive mode 112
Time setting function 116

U

UDP 21
Unlock processing 160
Unpassive 66
user 133

W

Wild card. 129

INSTRUCTION INDEX

S

S.SOCRCVS 99
S(P).SOCRDATA 114
SP.ECPRTCL 61
SP.FTPGET 152
SP.FTPPUT 146
SP.MLGETADR 216
SP.MLOPEADR 213
SP.MLRECV 208
SP.MLSEND 204
SP.SLMPSND 44
SP.SOCCINF 106
SP.SOCCLOSE 92
SP.SOCCSET 109
SP.SOCOPEN 86
SP.SOCRCV 95
SP.SOCRMODE 111
SP.SOCSND 102

Memo

REVISIONS

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

Print date	*Manual number	Description
January 2010	SH(NA)-080891ENG-A	First edition
April 2010	SH(NA)-080891ENG-B	Partial correction Section 8.5
January 2011	SH(NA)-080891ENG-C	Partial correction TERMS, Chapter 2, Section 3.5, 5.1, 6.1, 6.2, 6.3, 6.4.1, 6.4.3, 6.4.4, 6.4.5, 6.4.6, 8.4.2, 8.5 Addition Appendix 2
July 2011	SH(NA)-080891ENG-D	Addition of LCPU models Model addition L02CPU-P, L26CPU-PBT Partial correction INTRODUCTION, Chapter 1, Section 5.2.2, 6.3, 6.4.1, 6.4.6, Appendix 1 Addition Chapter 10, Section 10.1, 10.2 10.3, 10.4, 10.5, Appendix 3, 4
September 2011	SH(NA)-080891ENG-E	Partial correction SAFETY PRECAUTIONS, RELEVANT MANUALS, TERMS, Chapter 2, Section 3.5, 5.2.2
May 2012	SH(NA)-080891ENG-F	Partial correction Section 3.1, 3.5, 5.1, 6.1, 6.2, 8.4.2, 10.1
February 2013	SH(NA)-080891ENG-G	Addition of LCPU models and overall revision because of the model addition Model addition L06CPU, L26CPU
May 2013	SH(NA)-080891ENG-H	Addition of LCPU models Model addition L06CPU-P, L26CPU-P Partial correction INTRODUCTION, TERMS, Section 5.2.2, Appendix 1, 4
October 2013	SH(NA)-080891ENG-I	Partial correction Section 3.1, 3.2, 3.5, 5.1, 5.3, 6.1, 6.2, 6.4.1, 8.1, 10.1

Print date	*Manual number	Description
January 2014	SH(NA)-080891ENG-J	Partial correction Chapter 1, Section 7.4.1, 9.2, 9.3, Appendix 1, 5 Addition Chapter 6, 12
February 2014	SH(NA)-080891ENG-K	Partial correction Section 7.4.1
June 2014	SH(NA)-080891ENG-L	Partial correction Section 11.1, Appendix 3
September 2014	SH(NA)-080891ENG-M	Partial correction Section 7.4.2
December 2014	SH(NA)-080891ENG-N	Partial correction Chapter 1, 2, Section 7.3, 7.4.1 to 7.4.9, Chapter 9, Appendix 1, 3, 5 Addition Chapter 10, 15 Change Chapter 10 to 13 → Chapter 11 to 14
March 2015	SH(NA)-080891ENG-O	Model addition NZ1MEM-2GBSD, NZ1MEM-4GBSD, NZ1MEM-8GBSD, NZ1MEM-16GBSD Partial correction TERMS, Section 7.4.4, 9.2, 10.2
January 2017	SH(NA)-080891ENG-P	Partial correction Chapter 1, 5, Appendix 1, 3, 5 Addition Section 5.2

Japanese manual version SH-080875-Q

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

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 to:

(1) Damages caused by any cause found not to be the responsibility of Mitsubishi.

(2) Loss in opportunity, lost profits incurred to the user by Failures of Mitsubishi products.

(3) Special damages and secondary damages whether foreseeable or not, compensation for accidents, and compensation for damages to products other than Mitsubishi products.

(4) 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.

TRADEMARKS

Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
Ethernet is a registered trademark of Fuji Xerox Co., Ltd. in Japan.
The company names, system names and product names mentioned in this manual are either registered trademarks or trademarks of their respective companies.
In some cases, trademark symbols such as '™' or '®' are not specified in this manual.

SH(NA)-080891ENG-P(1701)MEE
MODEL: LCPU-U-NE-E
MODEL CODE: 13JZ37

MITSUBISHI ELECTRIC CORPORATION

HEAD OFFICE : TOKYO BUILDING, 2-7-3 MARUNOUCHI, CHIYODA-KU, TOKYO 100-8310, JAPAN
NAGOYA WORKS : 1-14 , YADA-MINAMI 5-CHOME , HIGASHI-KU, NAGOYA , JAPAN

When exported from Japan, this manual does not require application to the Ministry of Economy, Trade and Industry for service transaction permission.

Specifications subject to change without notice.