



Programmable Controller

MELSEC iQ-F
series



MELSEC iQ-F



FX5 User's Manual (Ethernet Communication)

SAFETY PRECAUTIONS

(Read these precautions before use.)

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

This manual classifies the safety precautions into two categories: [ 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. |

Depending on the circumstances, procedures indicated by [ CAUTION] may also cause severe injury.

It is important to follow all precautions for personal safety.


Store this manual in a safe place so that it can be read whenever necessary. Always forward it to the end user.

[DESIGN PRECAUTIONS]

WARNING

- Make sure to set up the following safety circuits outside the PLC to ensure safe system operation even during external power supply problems or PLC failure. Otherwise, malfunctions may cause serious accidents.
 - (1) Note that when the CPU module detects an error, such as a watchdog timer error, during self-diagnosis, all outputs are turned off. Also, when an error that cannot be detected by the CPU module occurs in an input/output control block, output control may be disabled. External circuits and mechanisms should be designed to ensure safe machinery operation in such a case.
- Construct an interlock circuit in the program so that the whole system always operates on the safe side before executing the control (for data change) of the PLC in operation. Read the manual thoroughly and ensure complete safety before executing other controls (for program change, parameter change, forcible output and operation status change) of the PLC in operation. Otherwise, the machine may be damaged and accidents may occur due to erroneous operations.
- For the operating status of each station after a communication failure of the network, refer to relevant manuals for the network. Incorrect output or malfunction may result in an accident.
- When executing control (data change) to a running other station programmable controller by connecting the external device to the SLMP compatible device, configure interlock circuits in the program of the other station programmable controller to ensure that the entire system operates safely at any times.

For other controls to a running other station programmable controller (such as program modification or operating status change), read relevant manuals carefully and ensure the safety before the operation. Especially, in the case of a control from an external device to a remote other station programmable controller, immediate action cannot be taken for a problem on the programmable controller due to communication failure. Determine the handling method as a system when communication failure occurs along with configuration of interlock circuit on other station PLC program, by considering external equipment and other station PLC.

- Do not write any data into the "system area" or "write protect area" of the buffer memory in the SLMP compatible device or intelligent function module. Also, do not output (ON) any "use prohibited" signals among the signals which are output to the SLMP compatible device and intelligent function device. Executing data writing to the "system area" or "write protect area", or outputting "use prohibited" signals may cause malfunction of the programmable controller alarm.
 - Do not write any data to the "system area" and "write-protect area" of the buffer memory in the intelligent function module. Executing data writing to the "system area" or "write-protect area" may cause malfunction of the programmable controller alarm. For the "system area" or "write-protect area", refer to  Page 283 List of Buffer Memory Applications and Assignments.
 - If a communication cable is disconnected, the network may be unstable, resulting in a communication failure of multiple stations. Construct an interlock circuit in the program so that the system always operates on the safe side even if communications fail. Incorrect output or malfunction may result in an accident.
-

[SECURITY PRECAUTIONS]

WARNING

- To maintain the security (confidentiality, integrity, and availability) of the programmable controller and the system against unauthorized access, denial-of-service (DoS) attacks, computer viruses, and other cyberattacks from unreliable networks and devices via network, take appropriate measures such as firewalls, virtual private networks (VPNs), and antivirus solutions.
-

[INSTALLATION PRECAUTIONS]

CAUTION

- Connect/disconnect the Ethernet cable by holding the connector and keeping it straight. If the cable connected to the module is pulled, the module or cable may be damaged, or malfunction may be caused by cable contact failure.
-

[WIRING PRECAUTIONS]

WARNING

- Make sure to cut off all phases of the power supply externally before attempting installation or wiring work. Failure to do so may cause electric shock or damage to the product.
 - Make sure to attach the terminal cover, provided as an accessory, before turning on the power or initiating operation after installation or wiring work. Failure to do so may cause electric shock.
-

[WIRING PRECAUTIONS]

CAUTION

- Install module so that excessive force will not be applied to terminal blocks, power connectors, I/O connectors, communication connectors, or communication cables. Failure to do so may result in wire damage/breakage or PLC failure.
 - Do not bundle the power line, control line and communication cables together with or lay them close to the main circuit, high-voltage line, load line or power line. As a guideline, lay the power line, control line and communication cables at least 100 mm away from the main circuit, high-voltage line, load line or power line.
-

[STARTUP AND MAINTENANCE PRECAUTIONS]

WARNING

- Do not touch any terminal while the PLC's power is on. Doing so may cause electric shock or malfunctions.
 - Before cleaning or retightening terminals, cut off all phases of the power supply externally. Failure to do so in the power ON status may cause electric shock.
 - Before modifying the program in operation, forcible output, running or stopping the PLC, read through this manual carefully, and ensure complete safety. An operation error may damage the machinery or cause accidents.
 - Do not change the program in the PLC from two or more peripheral equipment devices at the same time. (i.e. from an engineering tool and a GOT) Doing so may cause destruction or malfunction of the PLC program.
-

[STARTUP AND MAINTENANCE PRECAUTIONS]

CAUTION

- When connecting an external device with a CPU module or intelligent function module to modify data of a running programmable controller, 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, parameter change, forced output, or operating status change) of a running programmable controller, read the relevant manuals carefully and ensure that the operation is safe before proceeding. Improper operation may damage machines or cause accidents.
 - 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.
 - Do not disassemble or modify the PLC. Doing so may cause fire, equipment failures, or malfunctions. For repair, contact your local Mitsubishi Electric representative.
 - After the first use of the SD memory card, do not insert/remove the memory card more than 500 times. 500 times or more may cause malfunction.
 - Turn off the power to the PLC before attaching or detaching the following devices. Failure to do so may cause equipment failures or malfunctions.
 - Peripheral devices, expansion board, expansion adapter, and connector conversion adapter
 - Extension modules and bus conversion module
 - Battery
 - Read relevant manuals carefully and ensure the safety before performing online operations (operation status change) with peripheral devices connected to the running SLMP compatible device or CPU modules of other stations. Improper operation may damage machines or cause accidents.
-

[OPERATION PRECAUTIONS]

CAUTION

- Construct an interlock circuit in the program so that the whole system always operates on the safe side before executing the control (for data change) of the PLC in operation.
Read the manual thoroughly and ensure complete safety before executing other controls (for program change, parameter change, forcible output and operation status change) of the PLC in operation. Otherwise, the machine may be damaged and accidents may occur by erroneous operations.
 - Do not power off the CPU module or reset the CPU module while the setting values in the buffer memory are being written to the flash ROM in the intelligent function module. Doing so will make the data in the flash ROM card undefined. The values need to be set in the buffer memory and written to the flash ROM again. Doing so can cause malfunction or failure of the module.
 - Note that the whole system may not be reset by the RUN/STOP/RESET switch when the CPU module or intelligent function module detects an error, such as a watchdog timer error, during self-diagnosis. In such cases, turn the power off and on again.
-

INTRODUCTION

This manual contains text, diagrams and explanations which will guide the reader in the correct installation, safe use and operation of the Ethernet function of FX5 CPU module and Ethernet module.

It should be read and understood before attempting to install or use the unit. Store this manual in a safe place so that you can read it whenever necessary.

Always forward it to the end user.

Regarding use of this product

- This product has been manufactured as a general-purpose part for general industries, and has not been designed or manufactured to be incorporated in a device or system used in purposes related to human life.
- Before using the product for special purposes such as nuclear power, electric power, aerospace, medicine or passenger movement vehicles, consult Mitsubishi Electric.
- This product has been manufactured under strict quality control. However when installing the product where major accidents or losses could occur if the product fails, install appropriate backup or failsafe functions in the system.

Note

- If in doubt at any stage during the installation of the product, always consult a professional electrical engineer who is qualified and trained in the local and national standards. If in doubt about the operation or use, please consult the nearest Mitsubishi Electric representative.
- Mitsubishi Electric will not accept responsibility for actual use of the product based on these illustrative examples.
- This manual content, specification etc. may be changed, without a notice, for improvement.
- The information in this manual has been carefully checked and is believed to be accurate; however, if you notice a doubtful point, an error, etc., please contact the nearest Mitsubishi Electric representative. When doing so, please provide the manual number given at the end of this manual.

MEMO

CONTENTS

| | |
|--|-----------|
| SAFETY PRECAUTIONS | 1 |
| INTRODUCTION | 6 |
| RELEVANT MANUALS | 12 |
| TERMS | 13 |
| CHAPTER 1 OUTLINE | 16 |
| CHAPTER 2 SPECIFICATIONS | 18 |
| 2.1 Communication Specifications | 18 |
| 2.2 Connection specifications | 20 |
| CHAPTER 3 LIST OF FUNCTIONS | 21 |
| CHAPTER 4 CONNECTION WITH MELSOFT PRODUCT AND GOT | 23 |
| 4.1 Direct Connection with Engineering Tool | 23 |
| Setting method | 24 |
| Precautions | 29 |
| 4.2 Connection via a hub | 30 |
| Setting the Ethernet-equipped module | 31 |
| Engineering Tool Settings | 33 |
| Searching Ethernet-equipped modules on Network | 37 |
| Communication via Router | 39 |
| Precautions | 40 |
| CHAPTER 5 SLMP FUNCTION | 42 |
| 5.1 Specifications | 44 |
| Communication specifications | 44 |
| Link specifications | 45 |
| 5.2 Setting Method | 46 |
| 5.3 SLMP Commands | 48 |
| Command list | 48 |
| Applicable devices | 57 |
| 5.4 Data Communication Process | 61 |
| 5.5 Precautions | 61 |
| CHAPTER 6 PREDEFINED PROTOCOL SUPPORT FUNCTION | 64 |
| 6.1 Data Communication | 65 |
| 6.2 Protocol Communication Type | 70 |
| 6.3 Packet Elements | 71 |
| 6.4 Execution Conditions of Predefined Protocol Communications | 76 |
| 6.5 Example of Predefined Protocol Communications | 77 |
| 6.6 Predefined Protocol Support Function Instruction | 83 |
| Executing the registered protocols | 83 |
| 6.7 Precautions | 89 |
| CHAPTER 7 SOCKET COMMUNICATION FUNCTION | 90 |
| 7.1 Communication Using TCP | 91 |
| Program example | 92 |

| | | |
|---|--|------------|
| 7.2 | Communication Using UDP | 99 |
| | Program example | 100 |
| 7.3 | Precautions | 102 |
| 7.4 | Socket Communication Function Instructions | 104 |
| | Opening a connection | 105 |
| | Disconnecting a connection | 113 |
| | Reading received data in the END processing | 119 |
| | Reading received data in the processing | 123 |
| | Sending data | 126 |
| | Reading connection information | 133 |
| | Reading socket communication receive data | 135 |
| CHAPTER 8 FILE TRANSFER FUNCTION (FTP SERVER) | | 137 |
| 8.1 | Data communication procedures | 137 |
| 8.2 | Files that can be transferred with FTP | 141 |
| 8.3 | FTP command | 141 |
| 8.4 | Precautions | 147 |
| CHAPTER 9 FILE TRANSFER FUNCTION (FTP CLIENT) | | 149 |
| 9.1 | File Transfer Specifications of File Transfer Function (FTP Client) | 149 |
| | Transferable file | 149 |
| 9.2 | Procedure for Executing the File Transfer Function (FTP Client) | 150 |
| | Setting in the external device (FTP server) | 150 |
| | Setting in the CPU module (FTP client) | 151 |
| | Executing a file transfer function instruction | 151 |
| 9.3 | Program Example | 152 |
| | Program example to specify wild cards | 152 |
| | Program example to transfer files one by one | 155 |
| | Example of program for obtaining files from FTP server | 158 |
| 9.4 | Precautions | 161 |
| CHAPTER 10 TIME SETTING FUNCTION (SNTP CLIENT) | | 162 |
| CHAPTER 11 WEB SERVER FUNCTION | | 165 |
| 11.1 | JavaScript Objects | 165 |
| | Common information on JavaScript Objects | 165 |
| | Data block (WSDatblk) | 168 |
| | Level display (WSLevel) | 170 |
| | Figure display (WSFigure) | 172 |
| | Image display (WSPicture) | 174 |
| | Historical graph display (WSHstgrp) | 176 |
| | Write button (WSWrtBtn) | 178 |
| | Logout button (WSLogoutBtn) | 179 |
| 11.2 | CGI Object | 180 |
| | Data specified for CGI object | 180 |
| | Device read CGI | 182 |
| | Device write CGI | 187 |
| 11.3 | Error message | 191 |

| | |
|---|------------|
| CHAPTER 12 SECURITY FUNCTION | 192 |
| 12.1 IP Filter Function | 192 |
| 12.2 Remote Password | 196 |
| Communication using remote password | 196 |
| Remote password setting | 197 |
| Precautions | 199 |
| Detection of unauthorized access and actions | 200 |
| CHAPTER 13 SIMPLE CPU COMMUNICATION FUNCTION | 201 |
| 13.1 Setting Procedure | 203 |
| 13.2 Checking the simple CPU communication status | 217 |
| 13.3 Program Example | 221 |
| 13.4 Precautions | 223 |
| CHAPTER 14 IP ADDRESS CHANGE FUNCTION | 225 |
| 14.1 Overview of the IP address change function | 225 |
| 14.2 IP address to be set for the Ethernet-equipped module | 226 |
| 14.3 Write operation to IP address storage area | 227 |
| IP address storage area write procedure | 227 |
| 14.4 Clear operation to IP address storage area | 229 |
| IP address storage area clear procedure | 229 |
| 14.5 Precautions | 230 |
| CHAPTER 15 TROUBLESHOOTING | 231 |
| 15.1 Checking Errors by LEDs | 231 |
| Error display check | 231 |
| Error information read/clear method | 232 |
| 15.2 Checking Errors by GX Works3 | 233 |
| Ethernet diagnostics | 233 |
| Module diagnostics | 238 |
| Event history function | 240 |
| 15.3 Troubleshooting by Symptom | 241 |
| IP address duplication detection | 241 |
| When using the file transfer function | 242 |
| When the simple CPU communication function is used | 243 |
| 15.4 Error Codes | 245 |
| Error codes of the IP address change function | 245 |
| Error codes of the simple CPU communication function | 246 |
| Error codes of completion status | 248 |
| Error codes of the Ethernet communication | 249 |
| SLMP function error code | 255 |
| Module error code of Ethernet module | 257 |
| 15.5 List of Event Code | 261 |
| 15.6 Troubleshooting Flowchart | 266 |
| Errors during SLMP communication | 267 |
| Errors during file transfer function (FTP server) | 268 |
| Errors during file transfer function (FTP client) | 269 |

| | |
|---|----------------|
| APPENDIX | 270 |
| Appendix 1 List of Special Device Applications and Assignments | 270 |
| Appendix 2 List of Buffer Memory Applications and Assignments | 283 |
| Appendix 3 Processing Time | 287 |
| List of performance of simple CPU communication function | 287 |
| Appendix 4 Added and Changed Functions | 288 |
| INDEX | 291 |
| REVISIONS | 293 |
| WARRANTY | 295 |
| TRADEMARKS | 296 |

RELEVANT MANUALS

| Manual name <manual number> | Description |
|--|---|
| MELSEC iQ-F FX5 User's Manual (Startup) <JY997D58201> | Describes the performance specifications, procedures before operation, and troubleshooting of the FX5 CPU module. |
| MELSEC iQ-F FX5UJ User's Manual (Hardware) <SH-082206ENG> | Describes the details of hardware of the FX5UJ CPU module, including input/output specifications, wiring, installation, and maintenance. |
| MELSEC iQ-F FX5U User's Manual (Hardware) <JY997D55301> | Describes the details of hardware of the FX5U CPU module, including input/output specifications, wiring, installation, and maintenance. |
| MELSEC iQ-F FX5UC User's Manual (Hardware) <JY997D61401> | Describes the details of hardware of the FX5UC CPU module, including input/output specifications, wiring, installation, and maintenance. |
| MELSEC iQ-F FX5 User's Manual (Application) <JY997D55401> | Describes the basic knowledge required for program design, functions of the CPU module, devices/labels, and parameters. |
| MELSEC iQ-F FX5 Programming Manual (Program Design) <JY997D55701> | Describes the specifications of ladder, ST, FBD/LD, and SFC programs, and labels. |
| MELSEC iQ-F FX5 Programming Manual (Instructions, Standard Functions/Function Blocks) <JY997D55801> | Describes the specifications of instructions and functions that can be used in programs. |
| MELSEC iQ-F FX5 User's Manual (Serial Communication) <JY997D55901> | Describes the N:N network, Parallel link, MELSEC Communication protocol, inverter communication, non-protocol communication, and predefined protocol support. |
| MELSEC iQ-F FX5 User's Manual (MELSEC Communication Protocol) <JY997D60801> | Explains methods for the device that is communicating with the CPU module by MC protocol to read and write the data of the CPU module. |
| MELSEC iQ-F FX5 User's Manual (MODBUS Communication) <JY997D56101> | Describes the MODBUS serial communication and MODBUS/TCP communication. |
| MELSEC iQ-F FX5 User's Manual (PROFIBUS) <SH-081910ENG> | Describes the PROFIBUS-DP master module. |
| MELSEC iQ-F FX5 User's Manual (Ethernet Communication) <JY997D56201> (This manual) | Describes the Ethernet communication function of the CPU module built-in and the Ethernet module. |
| MELSEC iQ-F FX5-ENET User's Manual <SH-082026ENG> | Describes the FX5-ENET. |
| MELSEC iQ-F FX5-ENET/IP User's Manual <SH-082027ENG> | Describes the FX5-ENET/IP. |
| MELSEC iQ-F FX5 User's Manual (BACnet) <SH-082218ENG> | BACnet functions of the Ethernet module. |
| MELSEC iQ-F FX5 User's Manual (SLMP) <JY997D56001> | Explains methods for the device that is communicating with the CPU module by SLMP to read and write the data of the CPU module. |
| MELSEC iQ-F FX5 User's Manual (CC-Link IE TSN) <SH-082215ENG> | Describes the CC-Link IE TSN module. |
| MELSEC iQ-F FX5 User's Manual (CC-Link IE) <JY997D64201> | Describes the CC-Link IE field network module. |
| MELSEC iQ-F FX5 User's Manual (CC-Link) <SH-081793ENG> | Describes the CC-Link system master/intelligent device module. |
| MELSEC iQ-F FX5 User's Manual (AnyWireASLINK) <SH-081796ENG> | Describes the AnyWireASLINK system master module. |
| MELSEC iQ-F FX5 User's Manual (Positioning Control - CPU module built-in, High-speed pulse input/output module) <JY997D56301> | Describes the positioning function of the CPU module built-in and the high-speed pulse input/output module. |
| MELSEC iQ-F FX5 User's Manual (Positioning Control - Intelligent function module) <SH-081805ENG> | Describes the positioning module. |
| MELSEC iQ-F FX5 Motion Module/Simple Motion Module User's Manual (Startup) <IB0300251> | Describes the specifications, procedures before operation, system configuration, wiring, and operation examples of the Motion module/Simple Motion module. |
| MELSEC iQ-F FX5 Motion Module/Simple Motion Module User's Manual (Application) <IB0300253> | Describes the functions, input/output signals, buffer memories, parameter settings, programming, and troubleshooting of the Motion module/Simple Motion module. |
| MELSEC iQ-F FX5 Motion Module/Simple Motion Module User's Manual (Advanced Synchronous Control) <IB0300255> | Describes the functions and programming for the synchronous control of the Motion module/Simple Motion module. |
| MELSEC iQ-F FX5 Motion Module User's Manual (CC-Link IE TSN) <IB0300568> | Describes the functions, parameter settings, troubleshooting, and buffer memories of the CC-Link IE TSN network. |

| Manual name <manual number> | Description |
|--|---|
| MELSEC iQ-F FX5 User's Manual (Analog Control - CPU module built-in, Expansion adapter) <JY997D60501> | Describes the analog function of the CPU module built-in and the analog adapter. |
| MELSEC iQ-F FX5 User's Manual (Analog Control - Intelligent function module) <SH-081802ENG> | Describes the analog input module, analog output module, and multiple input module. |
| MELSEC iQ-F FX5 User's Manual (Temperature Control) <SH-081799ENG> | Describes the temperature control module. |
| MELSEC iQ-F FX5 User's Manual (Safety Control) <SH-082078ENG> | Describes the safety extension modules. |
| GX Works3 Operating Manual <SH-081215ENG> | Describes the system configuration, parameter settings, and online operations of GX Works3. |
| Transition from MELSEC FX3G, FX3U, FX3UC Series to MELSEC iQ-F Series Handbook <JY997D66201> | Describes the transition from MELSEC FX3G/FX3U/FX3UC series to MELSEC iQ-F series. |


TERMS

Unless otherwise specified, this manual uses the following terms.

For details on the FX3 devices that can be connected with the FX5, refer to the User's Manual (Hardware) of the CPU module to be used.

| Terms | Description |
|---|--|
| ■Devices | |
| FX5 | Generic term for FX5UJ, FX5U and FX5UC PLCs |
| FX3 | Generic term for FX3S, FX3G, FX3GC, FX3U, and FX3UC PLCs |
| FX5 CPU module | Generic term for FX5UJ CPU module, FX5U CPU module and FX5UC CPU module |
| FX5UJ CPU module | Generic term for FX5UJ-24MR/ES, FX5UJ-24MT/ES, FX5UJ-24MT/ESS, FX5UJ-40MR/ES, FX5UJ-40MT/ES, FX5UJ-40MT/ESS, FX5UJ-60MR/ES, FX5UJ-60MT/ES, and FX5UJ-60MT/ESS |
| FX5U CPU module | Generic term for FX5U-32MR/ES, FX5U-32MT/ES, FX5U-32MT/ESS, FX5U-64MR/ES, FX5U-64MT/ES, FX5U-64MT/ESS, FX5U-80MR/ES, FX5U-80MT/ES, FX5U-80MT/ESS, FX5U-32MR/DS, FX5U-32MT/DS, FX5U-32MT/DSS, FX5U-64MR/DS, FX5U-64MT/DS, FX5U-64MT/DSS, FX5U-80MR/DS, FX5U-80MT/DS, and FX5U-80MT/DSS |
| FX5UC CPU module | Generic term for FX5UC-32MT/D, FX5UC-32MT/DSS, FX5UC-64MT/D, FX5UC-64MT/DSS, FX5UC-96MT/D, FX5UC-96MT/DSS, FX5UC-32MT/DS-TS, FX5UC-32MT/DSS-TS, and FX5UC-32MR/DS-TS |
| Extension module | Generic term for FX5 extension modules, FX3 function modules, Extension modules (extension cable type) and Extension module (extension connector type) |
| FX5 extension module | Generic term for I/O modules, FX5 extension power supply modules, FX5 intelligent function modules, and FX5 safety extension modules |
| FX3 extension module | Generic term for FX3 extension power supply module and FX3 intelligent function module |
| Extension module (extension cable type) | Generic term for Input modules (extension cable type), Output modules (extension cable type), Input/output modules (extension cable type), Powered input/output module, High-speed pulse input/output module, Extension power supply module (extension cable type), Connector conversion module (extension cable type), Intelligent function modules, Safety extension modules, and Bus conversion module (extension cable type) |
| Extension module (extension connector type) | Generic term for Input modules (extension connector type), Output modules (extension connector type), Input/output modules (extension connector type), Extension power supply module (extension connector type), Connector conversion module (extension connector type), and Bus conversion module (extension connector type) |
| I/O module | Generic term for Input modules, Output modules, Input/output modules, Powered input/output modules, and High-speed pulse input/output modules |
| Input module | Generic term for Input modules (extension cable type) and Input modules (extension connector type) |
| Input module (extension cable type) | Generic term for FX5-8EX/ES and FX5-16EX/ES |
| Input module (extension connector type) | Generic term for FX5-C16EX/D, FX5-C16EX/DS, FX5-C32EX/D, FX5-C32EX/DS, and FX5-C32EX/DS-TS |
| Output module | Generic term for Output modules (extension cable type) and Output modules (extension connector type) |
| Output module (extension cable type) | Generic term for FX5-8EYR/ES, FX5-8EYT/ES, FX5-8EYT/ESS, FX5-16EYR/ES, FX5-16EYT/ES, and FX5-16EYT/ESS |
| Output module (extension connector type) | Generic term for FX5-C16EYT/D, FX5-C16EYT/DSS, FX5-C16EYR/D-TS, FX5-C32EYT/D, FX5-C32EYT/DSS, FX5-C32EYT/D-TS, and FX5-C32EYT/DSS-TS |
| Input/output module | Generic term for Input/output modules (extension cable type) and Input/output modules (extension connector type) |
| Input/output module (extension cable type) | Generic term for FX5-16ER/ES, FX5-16ET/ES, and FX5-16ET/ESS |

| Terms | Description |
|--|---|
| Input/output module (extension connector type) | Generic term for FX5-C32ET/D, FX5-C32ET/DSS, FX5-C32ET/DS-TS, and FX5-C32ET/DSS-TS |
| Powered input/output module | Generic term for FX5-32ER/ES, FX5-32ET/ES, FX5-32ET/ESS, FX5-32ER/DS, FX5-32ET/DS, and FX5-32ET/DSS |
| High-speed pulse input/output module | Generic term for FX5-16ET/ES-H and FX5-16ET/ESS-H |
| Extension power supply module | Generic term for FX5 extension power supply module and FX3 extension power supply module |
| FX5 extension power supply module | Generic term for FX5 extension power supply module (extension cable type) and FX5 extension power supply module (extension connector type) |
| FX5 extension power supply module (extension cable type) | Different name for FX5-1PSU-5V |
| FX5 extension power supply module (extension connector type) | Different name for FX5-C1PS-5V |
| FX3 extension power supply module | Different name for FX3U-1PSU-5V |
| Intelligent module | The abbreviation for intelligent function modules |
| Intelligent function module | Generic term for FX5 intelligent function modules and FX3 intelligent function modules |
| FX5 intelligent function module | Generic term for FX5-4AD, FX5-4DA, FX5-8AD, FX5-4LC, FX5-20PG-P, FX5-20PG-D, FX5-40SSC-G, FX5-80SSC-G, FX5-40SSC-S, FX5-80SSC-S, FX5-ENET, FX5-ENET/IP, FX5-CCLGN-MS, FX5-CCLIEF, FX5-CCL-MS, FX5-ASL-M, and FX5-DP-M |
| FX3 intelligent function module | Generic term for FX3U-4AD, FX3U-4DA, FX3U-4LC, FX3U-1PG, FX3U-2HC, FX3U-16CCL-M, FX3U-64CCL, FX3U-128ASL-M, and FX3U-32DP |
| FX5 safety extension module | Generic term for safety main modules and safety expansion modules |
| Safety main module | Different name for FX5-SF-MU4T5 |
| Safety expansion module | Generic term for expansion modules installed to a safety main module |
| Safety input expansion module | Different name for FX5-SF-8DI4 |
| Expansion board | Generic term for board for FX5UJ CPU module and FX5U CPU module |
| Communication board | Generic term for FX5-232-BD, FX5-485-BD, and FX5-422-BD-GOT |
| Expansion adapter | Generic term for adapter for FX5 CPU module |
| Communication adapter | Generic term for FX5-232ADP and FX5-485ADP |
| Analog adapter | Generic term for FX5-4AD-ADP, FX5-4DA-ADP, FX5-4AD-PT-ADP, FX5-4AD-TC-ADP, and FX5-4A-ADP |
| Bus conversion module | Generic term for Bus conversion module (extension cable type) and Bus conversion module (extension connector type) |
| Bus conversion module (extension cable type) | Different name for FX5-CNV-BUS |
| Bus conversion module (extension connector type) | Different name for FX5-CNV-BUSC |
| Connector conversion module | Generic term for Connector conversion module (extension cable type) and Connector conversion module (extension connector type) |
| Connector conversion module (extension cable type) | Different name for FX5-CNV-IF |
| Connector conversion module (extension connector type) | Different name for FX5-CNV-IFC |
| Extended extension cable | Generic term for FX5-30EC and FX5-65EC |
| Connector conversion adapter | Different name for FX5-CNV-BC |
| Battery | Different name for FX3U-32BL |
| SD memory card | Generic term for NZ1MEM-2GBSD, NZ1MEM-4GBSD, NZ1MEM-8GBSD, NZ1MEM-16GBSD, L1MEM-2GBSD and L1MEM-4GBSD SD memory cards Abbreviation of Secure Digital Memory Card. Device that stores data using flash memory. |
| Peripheral device | Generic term for engineering tools and GOTs |
| GOT | Generic term for Mitsubishi Electric Graphic Operation Terminal GOT1000 and GOT2000 series |
| ■Software packages | |
| Engineering tool | The product name of the software package for the MELSEC programmable controllers |
| GX Works3 | The product name of the software package, SWnDND-GXW3, for the MELSEC programmable controllers (The 'n' represents a version.) |
| ■Communication-related | |
| Ethernet-equipped module | A generic term for the following modules when the Ethernet function is used: <ul style="list-style-type: none"> • CPU module • FX5-ENET • FX5-ENET/IP |

| Terms | Description |
|-------------------------------|--|
| Ethernet module | Generic term for FX5-ENET and FX5-ENET/IP |
| TCP | The abbreviation for 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 | The abbreviation for User Datagram Protocol. This is a connectionless protocol and thereby its speed is faster than that of TCP, but less reliable. (Data may be lost or not be received in correct order.) |
| SLMP | The abbreviation for Seamless Message Protocol. A protocol for accessing SLMP-compatible devices and PLCs that are connected to SLMP-compatible devices from external devices. |
| SLMP-compatible device | Generic term for devices that can receive SLMP messages. |
| MC protocol | The abbreviation of the MELSEC communication protocol. A protocol for accessing MC protocol-compatible devices and PLCs that are connected to MC protocol-compatible devices from external devices. |
| MC protocol-compatible device | Generic term for devices that can receive MC protocol messages. |
| MODBUS/TCP | A generic term for the protocol designed to use MODBUS protocol messages on a TCP/IP network. |
| FTP | The abbreviation for File Transfer Protocol. This protocol is used to transfer data files over a network. |
| External device | A generic term for personal computers connected by Ethernet for data communication and other Ethernet-equipped modules. |
| Relay station | A station that includes two or more network modules. Transient transmission is performed through this station to stations on other networks. |
| Buffer memory | Memory areas of Intelligent function modules and SLMP-compatible devices for storing setting values and monitor values. |
| Data logging file | The file which stored data collected by data logging function. |
| Device supporting iQSS | A generic term for a device which supports iQ Sensor Solution For details on iQ Sensor Solution, refer to the following.  iQ Sensor Solution Reference Manual |
| OPEN | Generic term for SP.SOCOPEN instruction and GP.OPEN instruction |
| CLOSE | Generic term for SP.SOCCLOSE instruction and GP.CLOSE instruction |
| SOCRCV | Generic term for SP.SOCRCV instruction and GP.SOCRCV instruction |
| SOCSND | Generic term for SP.SOCSND instruction and GP.SOCSND instruction |

1 OUTLINE

The following describes the Ethernet function of FX5 CPU module and Ethernet module.

Connection with engineering tool and GOT

- The CPU module can be connected to multiple engineering tools and GOT by using hub. Up to 8 external devices can be connected one CPU module at the same time.
- CPU modules connected to the same hub as the engineering tool can be searched and the IP address of the displayed target device can be specified.
- In MELSOFT connection, access through routers in an environment such as a corporate LAN.

Direct connection with engineering tool

The CPU module can be directly connected to the engineering tool with an Ethernet cable, without using a hub. For direct connection, the IP address and host name need not be specified in the transfer setup.

Communication using SLMP

CPU module device data can be read or written from external devices such as a personal computer or GOT, enabling the CPU module operation monitoring, data analysis, and production control.

Predefined protocol support

Data can be exchanged between the external device (such as measuring instrument and bar code reader) and the CPU module following the protocol of the device.

Socket communication

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

MODBUS/TCP communication

By using sequence program, MODBUS devices of the external devices connected through Ethernet can be read/written. For details, refer to [MELSEC iQ-F FX5 User's Manual \(MODBUS Communication\)](#).

File transfer function (FTP server)

Using the dedicated FTP commands enables an external device to read out, write, and delete individual data file.

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.

Time setting function (SNTP client)

Time information is collected from the time information server (SNTP server) connected on the LAN at the specified timing, and the CPU module's time is automatically set.

Web server function

Monitors and diagnoses the CPU module using a Web browser via connected network.

IP filter function

This function identifies IP address of the access source and prevents access by unauthorized IP addresses.

Simple CPU communication function

Allows data communications between specified devices at the specified timing just by setting simple parameters from the engineering tool for the CPU module.

Remote password

Unauthorized access from the outside can be prevented and the security can be enhanced by setting the remote password.

IP Address Change Function

This function is provided to change the IP address of the CPU module by setting the desired IP address to special registers from a peripheral unit or another unit and turning ON a special relay.

This function changes the IP address of the CPU module even if no settings are made in GX Works3 PLC parameters.

CC-Link IE Field Network Basic

Data is periodically communicated between the master station and slave stations using link devices (cyclic transmission).

For details, refer to [MELSEC iQ-F FX5-ENET/IP User's Manual](#).

EtherNet/IP communication

The module can communicate seamlessly with an EtherNet/IP network by using the communication protocol CIP.

For details, refer to [MELSEC iQ-F FX5-ENET/IP User's Manual](#).

BACnet communication function

Service can be executed from each device to communicate data.

For details, refer to [MELSEC iQ-F FX5 User's Manual \(BACnet\)](#).

Automatic detection of connected devices

Detects devices supporting iQSS which are connected to the CPU module (built-in Ethernet port), and automatically displays them on "List of devices" and "Device map area" using an engineering tool.

Communication setting reflection of Ethernet device

Reflects the communication settings (such as IP addresses) in devices supporting iQSS in "Device map area" which are connected over Ethernet.

Sensor parameter read/write

Reads/writes parameters from/to iQSS-compatible devices.

2 SPECIFICATIONS

2.1 Communication Specifications

CPU module

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

| Item | | Specification |
|--------------------------------|-------------------------------|---|
| Transmission specifications | Data transfer speed | 100/10 Mbps |
| | Communication mode | Full-duplex or half-duplex ^{*1} |
| | Interface | RJ45 connector |
| | Transmission method | Base band |
| | Maximum segment length | 100 m (length between hub and node) ^{*2} |
| | Number of cascade connections | 100BASE-TX |
| 10BASE-T | | 4 levels maximum ^{*3} |
| Protocol type | | CC-Link IE field network Basic, MELSOFT connection, SLMP server (3E/1E frame), Socket communication, Predefined protocol support, FTP Server, FTP Client, MODBUS/TCP communication, SNTP client, Web server (HTTP), Simple CPU communication function |
| Number of connections | | Total of 8 connections ^{*4*5} (Up to 8 external devices can access one CPU module at the same time.) |
| Hub ^{*1} | | Hubs with 100BASE-TX or 10BASE-T ports ^{*6} can be used. |
| IP address ^{*7} | | Initial value: 192.168.3.250 |
| Connection cable ^{*8} | 100BASE-TX | Ethernet cable of category 5 or higher (STP cable) |
| | 10BASE-T | Ethernet cable of category 3 or higher (STP cable) |

*1 IEEE802.3x flow control is not supported.

*2 For maximum segment length (length between hubs), consult the manufacturer of the hub used.

*3 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.

*4 The first device for MELSOFT connection is not included in the number of connections. (The second and the following devices are included.)

*5 The CC-Link IE field network Basic, FTP server, FTP client, SNTP client, Web server and simple CPU communication function are not included in the number of connections.

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

*7 If the first octet is 0 or 127, a parameter error (2222H) will occur. (Example: 0.0.0.0, 127.0.0.0, etc.)

*8 A straight cable can be used. If a personal computer or GOT and CPU module are directly connected a cross cable can be used.

Point

- When connected to a hub, the CPU module determines the cable used (100BASE-TX or 10BASE-T) 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.
- 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 get delayed by up to 500 ms.

Precautions


The operation of the following connections is not guaranteed. Check the operation before using the module.


- Connection using internet (general public line) (Internet-access service offered by an Internet service provider or a telecommunications carrier)
- Connection using firewall device(s)
- Connection using broadband router(s)
- Connection using wireless LAN

Ethernet module

This describes the communication specifications of the Ethernet port of the Ethernet module.

For another specifications, refer to the following manuals.

 MELSEC iQ-F FX5-ENET User's Manual

 MELSEC iQ-F FX5-ENET/IP User's Manual

| Item | | Specification |
|--------------------------------|-------------------------------|--|
| Transmission specifications | Data transfer speed | 100/10 Mbps |
| | Communication mode | Full-duplex or half-duplex ^{*1} |
| | Interface | RJ45 connector |
| | Transmission method | Base band |
| | Maximum segment length | 100 m (length between hub and node) ^{*2} |
| | Number of cascade connections | 100BASE-TX |
| 10BASE-T | | 4 levels maximum ^{*3} |
| Protocol type | FX5-ENET | CC-Link IE field network Basic, MELSOFT connection, SLMP server (3E/1E frame), Socket communication, simple CPU communication, BACnet/IP |
| | FX5-ENET/IP | EtherNet/IP communication, MELSOFT connection, SLMP server (3E/1E frame), Socket communication, simple CPU communication, BACnet/IP |
| Number of connections | | Total of 32 connections ^{*4} (Up to 32 external devices can access one Ethernet module at the same time.) |
| Hub ^{*1} | | Hubs with 100BASE-TX or 10BASE-T ports ^{*5} can be used. |
| IP address | | Initial value: 192.168.3.251 |
| Connection cable ^{*6} | 100BASE-TX | Ethernet cable of category 5 or higher (STP cable) |
| | 10BASE-T | Ethernet cable of category 3 or higher (STP cable) |

*1 IEEE802.3x flow control is not supported.

*2 For maximum segment length (length between hubs), consult the manufacturer of the hub used.

*3 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.

*4 The CC-Link IE field network Basic and EtherNet/IP communication are not included in the number of connections. The first device for MELSOFT connection is not included in the number of connections. (The second and the following devices are included.)

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

Hubs with 100BASE-TX ports can be used for CC-Link IE field network Basic and EtherNet/IP communication.

*6 Straight/cross cables can be used.

CC-Link IE field network Basic and EtherNet/IP communication are supported only for 100BASE-TX connection.

Point

- If the destination device of the Ethernet module does not respond due to power off or other reasons, Ethernet communication of the Ethernet module may get delayed by up to 500 ms.
- In the case of general-purpose Ethernet communication, when Ethernet module is connected with a hub, it distinguishes between 100BASE-TX and 10BASE-T and between full-duplex and half-duplex communication modes according to the hub. Set the hub to half-duplex mode if the hub does not have the auto-negotiation function.

Precautions

The operation of the following connections is not guaranteed. Check the operation before using the module.

- Connection using internet (general public line) (Internet-access service offered by an Internet service provider or a telecommunications carrier)
- Connection using firewall device(s)
- Connection using broadband router(s)
- Connection using wireless LAN

2.2 Connection specifications

Ethernet cable

Use one of the cables listed below for the Ethernet cable (100BASE-TX/10BASE-T cable) to connect to the Ethernet port of Ethernet-equipped module.

| Item | Specifications |
|-----------------------|---|
| When using 100BASE-TX | Ethernet cable: Category 5 or higher (STP cable ^{*1}) |
| When using 10BASE-T | Ethernet cable: Category 3 or higher (STP cable ^{*1}) |

*1 Shielded twisted pair cable.

■CPU module

A straight cable can be used. A cross cable can also be used when using direct connection between the personal computer and the built-in Ethernet.

■Ethernet module

A straight/cross cable can be used.

Ethernet cable connection

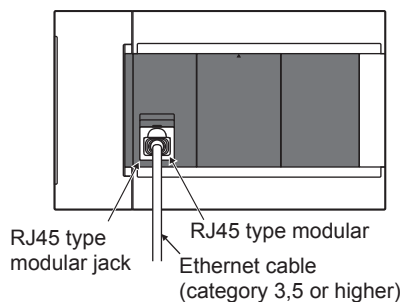
This section describes how to connect the Ethernet to a 100BASE-TX/10BASE-T network.

<Connection procedure>

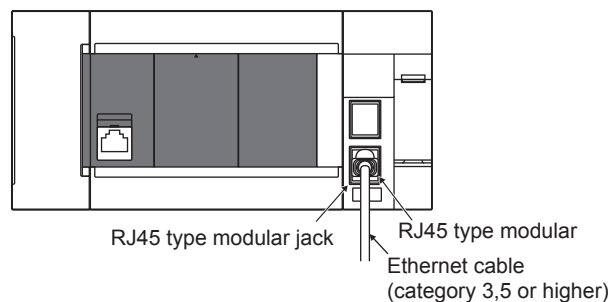
1. Connect the Ethernet cable to a hub.
2. Connect the Ethernet cable to the Ethernet.

The following shows the Ethernet cable connection diagram.

[CPU module]



[Ethernet module]



Point





- When connected to a hub, the Ethernet-equipped module determines the cable used (100BASE-TX or 10BASE-T) and the communication mode (full-duplex or half-duplex) according to the hub (Auto-negotiation function). Set the hub to the half-duplex mode if the hub that does not support the auto-negotiation function.
- When the ground terminal of the Ethernet-equipped module cannot be grounded, the communication line may be closed due to the effects of noise, making it impossible to communicate with other devices.

3 LIST OF FUNCTIONS

The following table shows the list of functions of the built-in Ethernet of the Ethernet-equipped module.

○: Supported, △: Partially supported, —: Not supported

| Function | Outline of system | Supported function | | | | Reference |
|--------------------------------------|---|--------------------|----------------|-----------------|-----------------|--|
| | | CPU module | | Ethernet module | | |
| | | FX5UJ | FX5U/ FX5UC | FX5- ENET | FX5- ENET/IP | |
| Direct connection with MELSOFT | Ethernet-equipped module and MELSOFT product (GX Works3, etc.) are connected by single Ethernet cable without using a hub. Communication is done by simply specifying the connection destination; setting the IP address is not required. | ○ | ○ | ○ | ○ | Page 23 Direct Connection with Engineering Tool |
| MELSOFT connection | Communication with MELSOFT products (GX Works3, etc.) is done within LAN such as company internal LAN. | ○ | ○ | ○ | ○ | Page 30 Connection via a hub |
| Connected module search function | Searches for Ethernet-equipped module connected with personal computer using GX Works3 within the same hub. Acquires IP address by selecting from search results list. | ○ | ○ | ○ | ○ | Page 37 Searching Ethernet-equipped modules on Network |
| MELSOFT diagnosis function | Diagnoses Ethernet port of CPU module and Ethernet module from GX Works3. (Ethernet diagnostics) | ○ | ○ | ○ | ○ | Page 233 Ethernet diagnostics |
| SLMP communication function | Reads and writes PLC data from other device. | △ | ○ | ○ | ○ | Page 42 SLMP FUNCTION |
| Predefined protocol support function | When the predefined protocol support function is used, data can be exchanged with the external device. | ○ | ○ | — | — | Page 64 PREDEFINED PROTOCOL SUPPORT FUNCTION |
| Socket communication function | By using socket communication instructions, any data can be transferred from and to the external devices connected through Ethernet using TCP or UDP. | ○ | ○ | ○ | ○ | Page 90 SOCKET COMMUNICATION FUNCTION |
| MODBUS/TCP communication | By using sequence program, MODBUS devices of the external devices connected through Ethernet can be read/written. | ○ | ○ | — | — | □□MELSEC iQ-F FX5 User's Manual (MODBUS Communication) |
| File transfer function (FTP server) | Using the dedicated FTP commands enables an external device to read out, write, and delete individual data file. | ○ | ○ | — | — | Page 137 FILE TRANSFER FUNCTION (FTP SERVER) |
| 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. | — | ○ | — | — | Page 149 FILE TRANSFER FUNCTION (FTP CLIENT) |
| Time setting function (SNTP client) | Time information is collected from the time information server (SNTP server) connected on the LAN at the specified timing, and the CPU module's time is automatically set. | ○ | ○ | — | — | Page 162 TIME SETTING FUNCTION (SNTP CLIENT) |
| Web server function | Monitors and diagnoses the CPU module using a Web browser via connected network. | △ | ○ | — | — | Page 165 WEB SERVER FUNCTION □□MELSEC iQ-R/MELSEC iQ-F Web Server Function Guide Book |
| IP filter function | This function identifies IP address of the access source and prevents access by unauthorized IP addresses. | ○ | ○ | ○ | ○ | Page 192 IP Filter Function |
| Remote password | Remote password setting can prevent unauthorized access from the outside and enhance the security of the system. | ○ | ○ | — | — | Page 196 Remote Password |
| Simple CPU communication function | Allows data communications between specified devices at the specified timing just by doing simple parameter settings from an engineering tool for the CPU module. | △ | ○ | ○ | ○ | Page 201 SIMPLE CPU COMMUNICATION FUNCTION |

| Function | Outline of system | Supported function | | | | Reference |
|---|---|--------------------|----------------|-----------------|-----------------|--|
| | | CPU module | | Ethernet module | | |
| | | FX5UJ | FX5U/ FX5UC | FX5- ENET | FX5- ENET/IP | |
| IP address change function | This function is provided to change the IP address of the CPU module by setting the desired IP address to special registers from a peripheral unit or another unit and turning ON a special relay. | ○ | ○ | ○ | ○ | Page 225 IP ADDRESS CHANGE FUNCTION |
| CC-Link IE field network Basic | Data is periodically communicated between the master station and slave stations using link devices (cyclic transmission). | ○ | ○ | ○ | — |  CC-Link IE Field Network Basic Reference Manual |
| EtherNet/IP communication | The module can communicate seamlessly with an EtherNet/IP network by using the communication protocol CIP. | — | — | — | ○ |  MELSEC iQ-F FX5-ENET/IP User's Manual |
| Automatic detection of connected devices | Detects devices supporting iQSS which are connected to the CPU module (built-in Ethernet port), and automatically displays them on "List of devices" and "Device map area" using an engineering tool. | ○ | ○ | — | — |  iQ Sensor Solution Reference Manual (SH-081133ENG) |
| Communication setting reflection of Ethernet device | Reflects the communication settings (such as IP addresses) in devices supporting iQSS in "Device map area" which are connected over Ethernet. | ○ | ○ | — | — | |
| Sensor parameter read/write | Reads/writes parameters from/to iQSS-compatible devices. | ○ | ○ | — | — | |
| BACnet function | Uses a PLC system as a BACnet device. | — | — | ○ | ○ |  MELSEC iQ-F FX5 User's Manual (BACnet) |

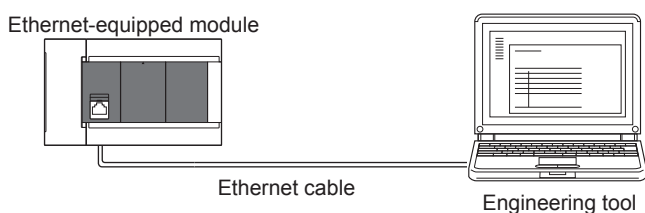
4 CONNECTION WITH MELSOFT PRODUCT AND GOT

This chapter describes the method of communication between the Ethernet-equipped module and MELSOFT Product (engineering tool, MX Component, etc.) or GOT.

4.1 Direct Connection with Engineering Tool

The Ethernet-equipped module can be directly connected to the engineering tool (GX Works3) with an Ethernet cable, without using a hub.

For direct connection, the IP address and host name need not be specified.



4

Point

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

With GX Works3, you can prevent hacking by opting to "Disable" for "Direct Connection with MELSOFT" by the following operation.

[CPU module]

Navigation window⇒[Parameter]⇒[FX5UJCPU] or [FX5UCPU]⇒[Module Parameter]⇒[Ethernet Port]⇒[Application Settings]⇒[Security]

[Ethernet module]

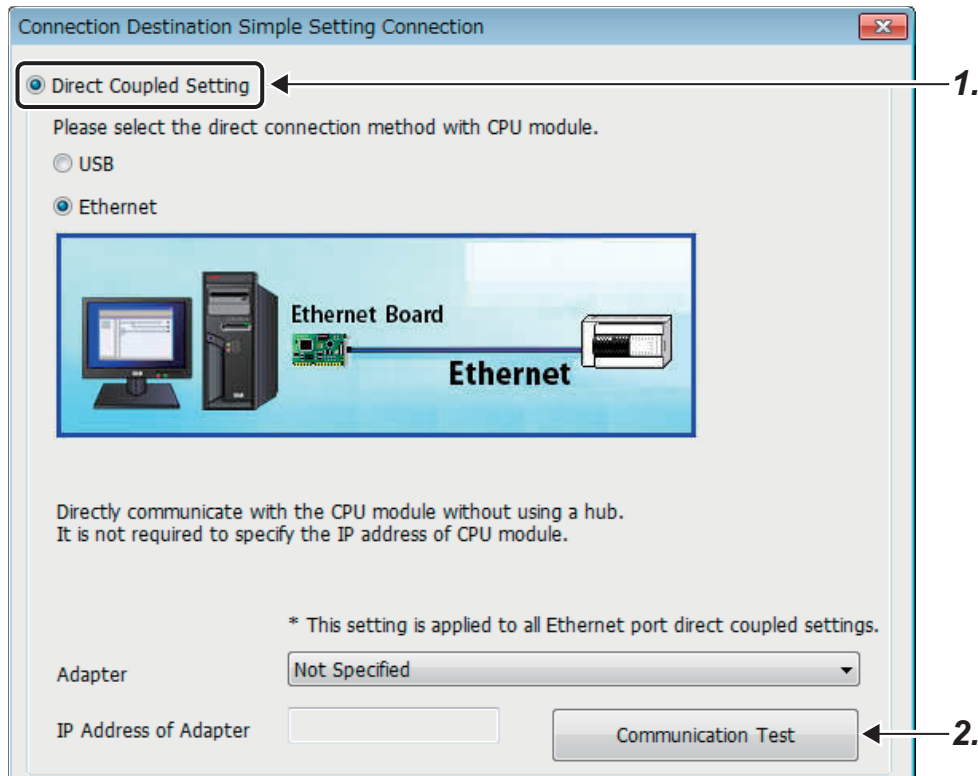
Navigation window⇒[Parameter]⇒[Module Information]⇒[FX5-ENET] or [FX5-ENET/IP]⇒[Application Settings]⇒[Security]

Setting method

Set on the "Specify Connection Destination Connection" screen of GX Works3.

🔗 Online⇒[Current Connection Destination]

Simple setting method



1. Select [Direct Coupled Setting] on the "Connection Destination Simple Setting Connection" screen.

Point

The Ethernet adapter on the personal computer side used for the Ethernet port direct connection can be specified.

Select an item appropriate to the operating environment.

2. Click the [Communication Test] button and check if the connection for the Ethernet-equipped module is possible.

Detailed setting method

■CPU module

Other Connection Method

Select this method if you want to connect to CPU module with a connection method other than the direct coupled setting.

Other Connection Method
(Open the Specify Connection Destination window)

1. On the "Connection Destination Simple Setting Connection" screen, select [Other Connection Method] and click the [Other Connection Method (Open the Specify Connection Destination window)] button.

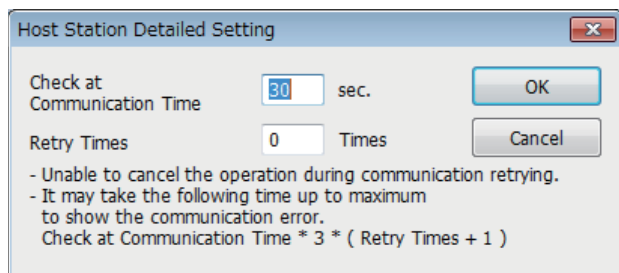
2. Select "Ethernet Board" for "PC side I/F".

3. Select "PLC Module" for "PLC side I/F" and double-click it.

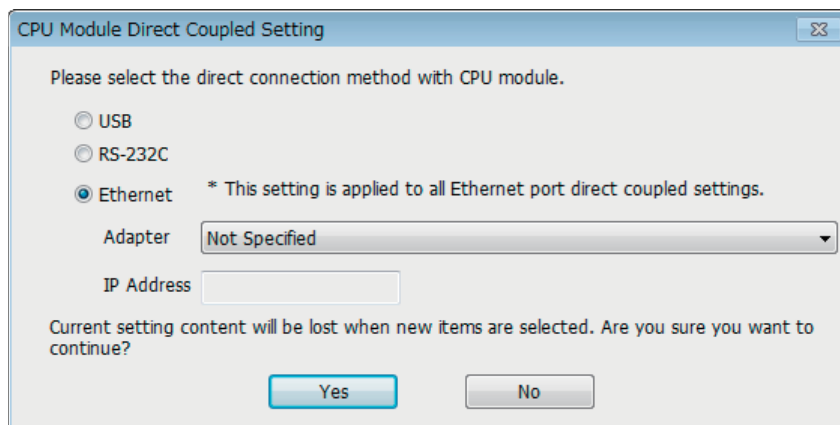
In the "PLC side I/F Detailed Setting of PLC Module" screen, select the "Ethernet Port Direct Connection" as shown below.

* Please select 'Connection via HUB' when you use HUB even if there is only one target device to communicate. If HUB is connected to other devices and also 'Ethernet Port Direct Connection' is selected during communication, the line becomes overloaded. This might affect other devices' communication.

4. Select "No Specification" on "Other Station Setting" and double-click it.
Set the "Other Station Setting" according to the operating environment.



5. Click the [CPU Module Direct Coupled Setting] button.
Select [Ethernet] for the connection method for the CPU module and click the [Yes] button.



Point

The Ethernet adapter on the personal computer side used for the Ethernet port direct connection can be specified.
Select an item appropriate to the operating environment.

6. Click the [Connection Test] button on the "Specify Connection Destination Connection" screen and check if the connection for the CPU module is possible.

■ Ethernet module

Other Connection Method

Select this method if you want to connect to CPU module with a connection method other than the direct coupled setting.

Other Connection Method
(Open the Specify Connection Destination window)

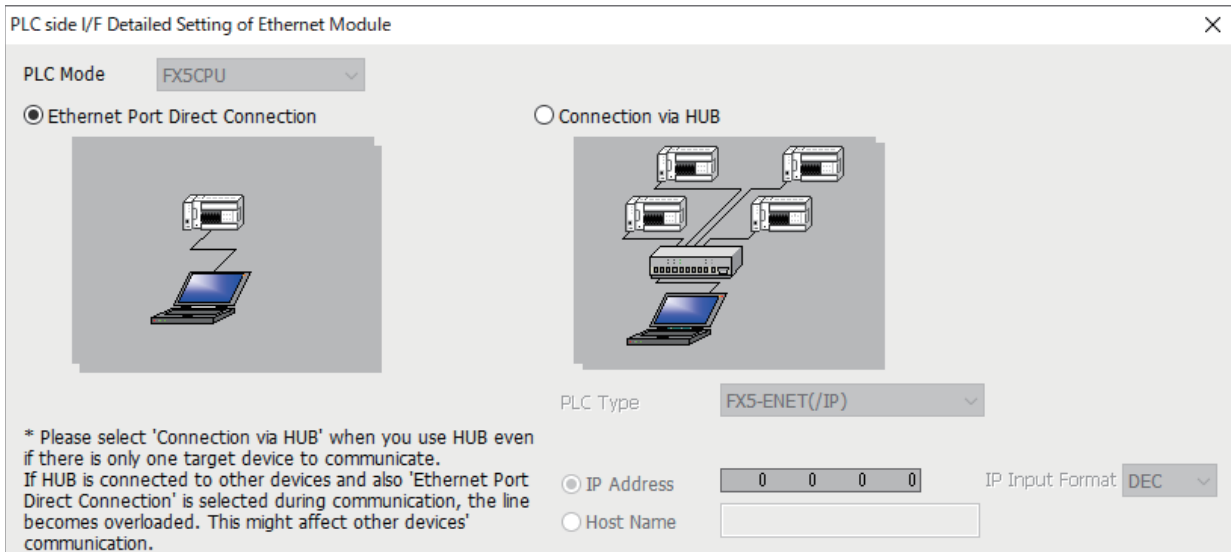
1. On the "Connection Destination Simple Setting Connection" screen, select [Other Connection Method] and click the [Other Connection Method (Open the Specify Connection Destination window)] button.

The screenshot shows the 'Specify Connection Destination Connection' dialog box. It is divided into several sections: 'PC side I/F', 'PLC side I/F', 'Other Station Setting', and 'Network Communication Route'. In the 'PC side I/F' section, the 'Ethernet Board' icon is selected and highlighted with a box and an arrow labeled '2.'. In the 'PLC side I/F' section, the 'Ethernet Module' icon is selected and highlighted with a box and an arrow labeled '3.'. In the 'Other Station Setting' section, the 'Other Station(Single Network)' icon is selected and highlighted with a box and an arrow labeled '4.'. In the 'Network Communication Route' section, the 'Ethernet' icon is selected and highlighted with a box and an arrow labeled '5.'. On the right side of the dialog, there are several buttons: 'Connection Channel List...', 'CPU Module Direct Coupled Setting' (with an arrow labeled '6.'), 'Connection Test', 'PLC Type', and 'System Image...'. At the bottom, there are 'OK' and 'Cancel' buttons.

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

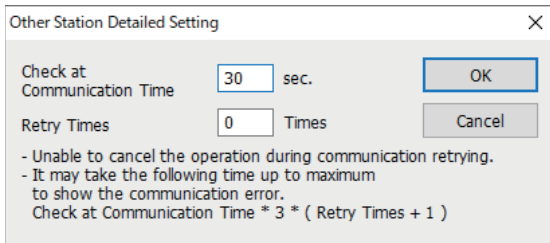
3. Select "Ethernet module" for PLC side I/F, and double-click it.

Select "Ethernet Port Direct Connection" on the "PLC side I/F Detailed Setting of Ethernet Module" screen.



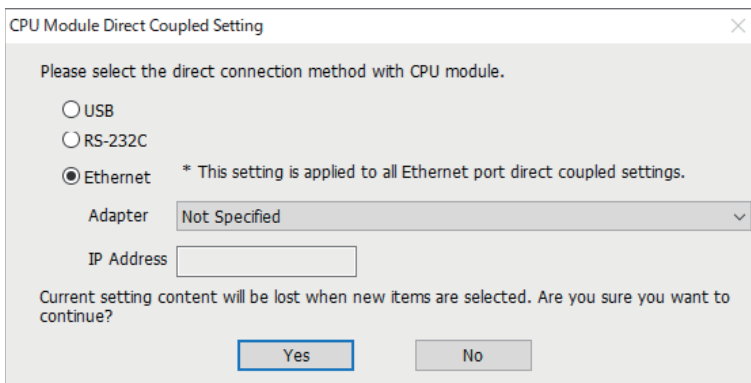
4. Double-click "Other Station (Single Network)" in Other Station Setting.

Set the other station specifications according to the operating environment.



5. Click the [CPU Module Direct Coupled Setting] button.

Select the [Ethernet] for the connection method for the CPU module and click the [Yes] button.



Point

The Ethernet adapter on the personal computer to be used for direct connection of Ethernet port can be specified. Select an item appropriate to the operating environment.

6. Click the [Communication Test] button on the "Specify Connection Destination Connection" screen, and check if the module can be connected with the Ethernet module.

Precautions

Connection to LAN line

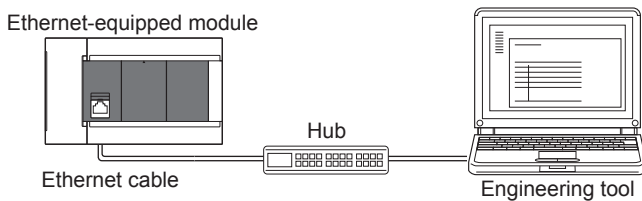
When connecting the CPU module to a LAN line, do not set direct connection. Doing so will apply a load on the LAN line and adversely affect communications with other external devices.

Direct connection through the Ethernet module

For direct connection through the Ethernet module, only one Ethernet port can be used. (For direct connection by using P1, P2 cannot be used. Also, P1 cannot be used when P2 is used.)

Indirect connection

When a Ethernet-equipped module is connected to an external device via a hub, communication cannot be performed by direct connection. (☞ Page 30 Connection via a hub)



Conditions that disallow direct connection

When the following condition is met, it may not be possible to communicate directly. In such case, check the setting of the Ethernet-equipped module and/or personal computer.

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

Ex.

Ethernet-equipped 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 Ethernet-equipped module IP address bits, if the bits corresponding to the host address of the class of the personal computer IP address are all ON or all OFF.

Ex.

Personal computer IP address: 192.168.0.1 ← 192.x.x.x., class C and the host address is the fourth octet.

Personal computer subnet mask: 255.0.0.0

Ethernet-equipped module IP address: 64.64.255.255 ← each bit turns on because of the fourth octet is 255

Point

The IP address 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 portion including "0" as shown below.

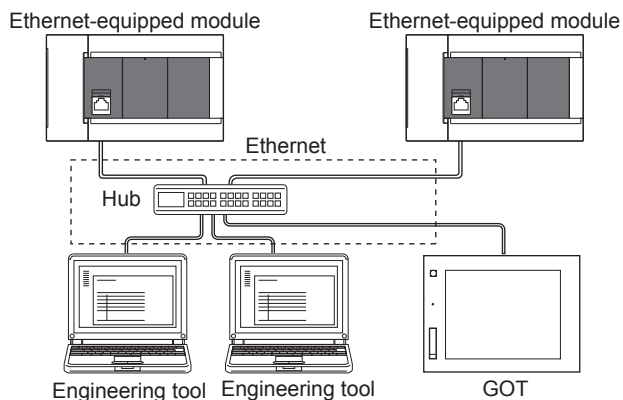
- Class A: 255.0.0.0
- Class B: 255.255.0.0
- Class C: 255.255.255.0

When the communication setting cannot be established

Even if direct connection with the Ethernet adapter of the personal computer is performed, the communication setting may not be established. When the communication setting cannot be established, set the appropriate IP address in the network setting for the personal computer. (GX Works3 Operating Manual)

4.2 Connection via a hub

In case of connection to Ethernet via hub, you must do Ethernet-equipped module settings and MELSOFT Product (engineering tool, etc.) settings or GOT settings.



The flow up to start of Ethernet communication by the connection via a hub is as follows.

1. Setting parameters

Create unit parameters (IP address) with the engineering tool. (Page 31 Setting module parameters)

2. Writing to the Ethernet-equipped module

Directly connect the Ethernet-equipped module and engineering tool, and write the set parameters to the Ethernet-equipped module. Turn power OFF → ON or reset the system to enable the parameters. (Page 32 Writing to the Ethernet-equipped module)

3. Connecting cables and external devices

Connect them via a hub to perform Ethernet communication. (Page 20 Connection specifications)

4. Setting the connection destination

Set connection destination with the engineering tool. (Page 33 Engineering Tool Settings)

For GOT settings, refer to the following manuals.

GOT2000 Series Connection Manual (Mitsubishi Products)

GOT1000 Series Connection Manual (Mitsubishi Products)

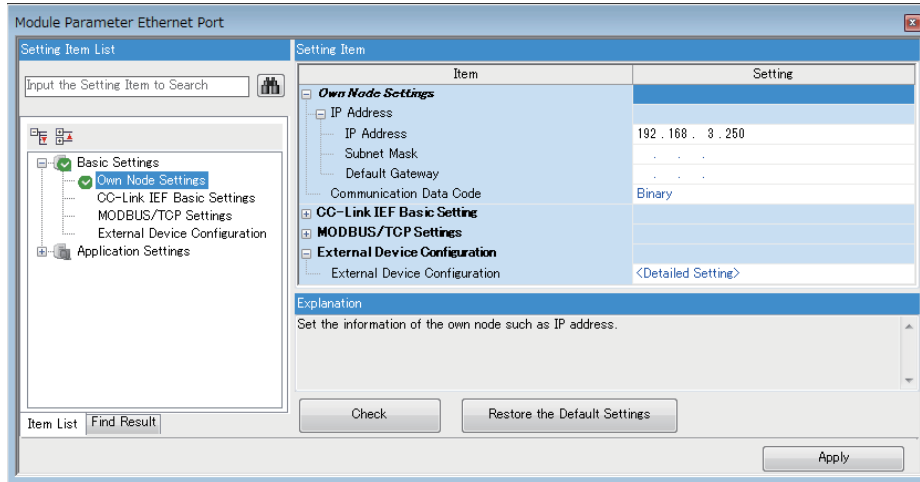
Setting the Ethernet-equipped module

Setting module parameters

■CPU module

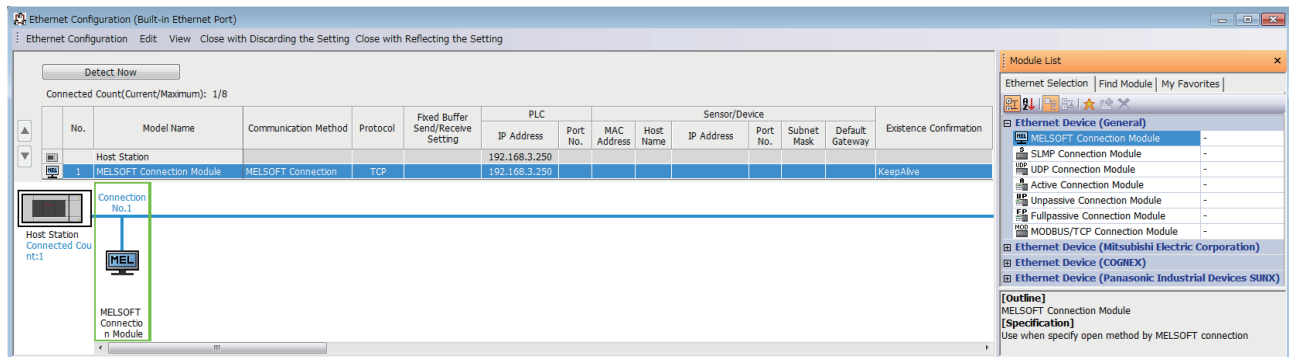
Set the parameters on the "Module parameter Ethernet Port" screen of GX Works3.

Navigation window⇒[Parameter]⇒[FX5UJCPU] or [FX5UCPU]⇒[Module Parameter]⇒[Ethernet Port]⇒[Basic Settings]⇒[Own Node Settings]



1. Set IP address of the CPU module.
2. Set MELSOFT connections.

Navigation window⇒[Parameter]⇒[FX5UJCPU] or [FX5UCPU]⇒[Module Parameter]⇒[Ethernet Port]⇒[Basic Settings]⇒[External Device Configuration]⇒[Detailed Setting]⇒[Ethernet Configuration (Built-in Ethernet Port)] screen

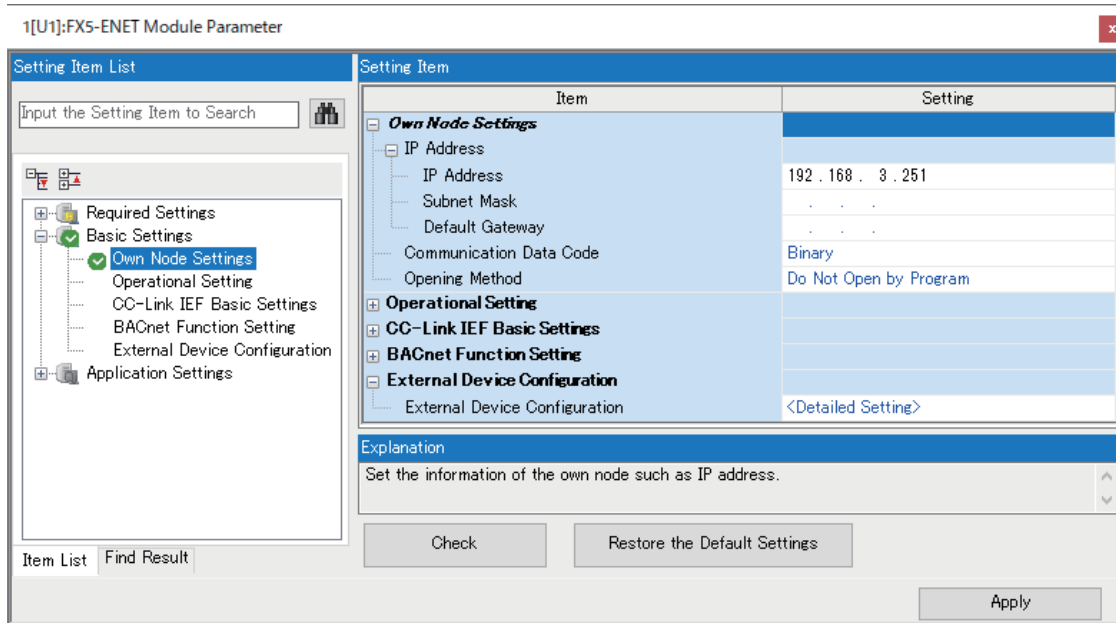


Drag and drop "MELSOFT Connection Module" from the "Module List" to the left side of the screen.

■ Ethernet module

Set the parameters on the “n[Un]: FX5-ENET(/IP) Module Parameter” screen of GX Works3.

Navigation window⇒[Parameter]⇒[Module Information]⇒[FX5-ENET] or [FX5-ENET/IP]⇒[Basic Settings]⇒[Own Node Settings]



1. Set the IP address of the Ethernet module.

Writing to the Ethernet-equipped module

Write the parameters set in the Ethernet-equipped module.

Navigation window⇒[Online]⇒[Write to PLC]

After writing the parameters to the Ethernet-equipped module, turn OFF and ON or reset the power to the CPU module to enable the parameters.

Engineering Tool Settings

Set on the "Specify Connection Destination Connection" screen of GX Works3.

Online⇒[Current Connection Destination]

Other Connection Method

Select this method if you want to connect to CPU module with a connection method other than the direct coupled setting.

Other Connection Method
(Open the Specify Connection Destination window)

CPU module

1. On the "Connection Destination Simple Setting Connection" screen, select [Other Connection Method] and click the [Other Connection Method (Open the Specify Connection Destination window)] button.

Specify Connection Destination Connection

PC side I/F

Serial USB Ethernet Board

PLC side I/F

PLC Module Ethernet Module GOT

Network No. - Station No. - Protocol TCP

IP Address/Host Name 192.168.3.250

PLC Mode FX5CPU

Other Station Setting

No Specification Other Station(Single Network)

Time Out (Sec.) 10 Retry Times 0

Network Communication Route

CC IE TSN Ethernet CC-Link

Connection Channel List...

CPU Module Direct Coupled Setting

Connection Test

PLC Type

System Image...

OK

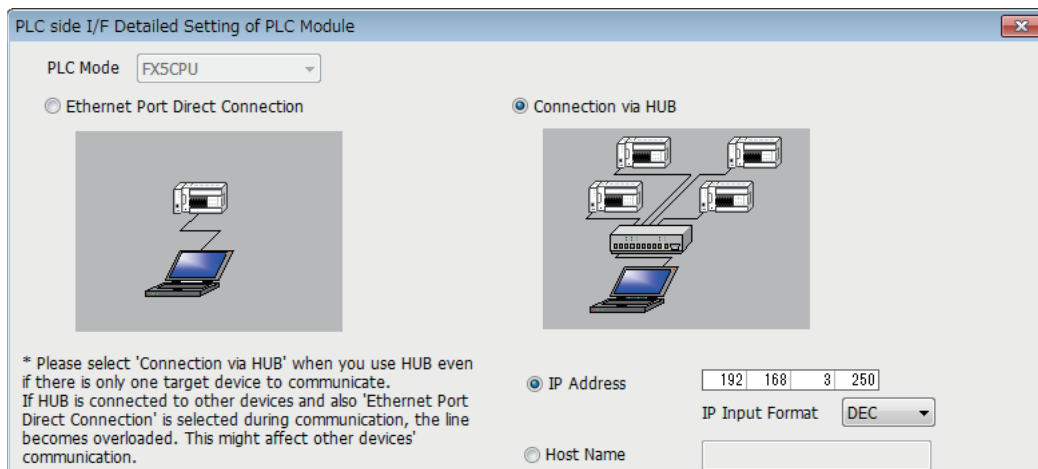
Cancel

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

3. Select "CPU Module" for "PLC side I/F" and double-click it.

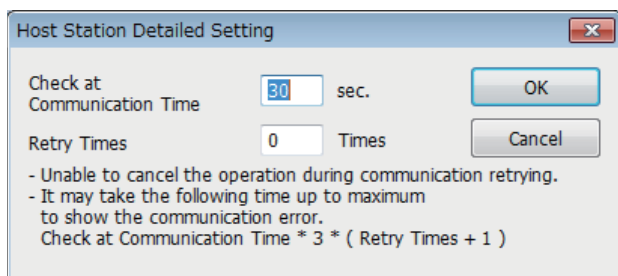
Input the CPU IP address or host name in the "PLC side I/F Detailed Setting of PLC Module" screen as shown in the following figure.

In case of host name, set the name specified in the Microsoft® Windows® hosts file.



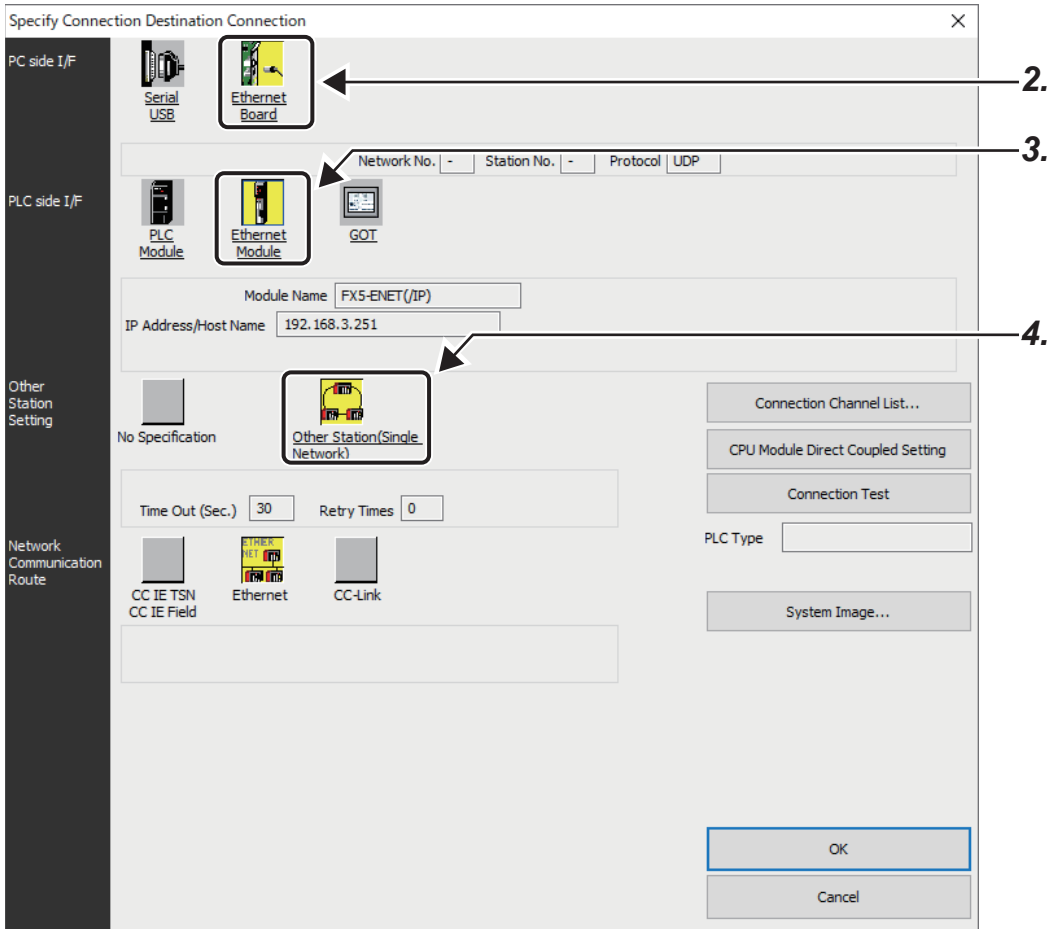
4. Select "No Other Station Specification" on "Other Station Specification" and double-click it.

Set the other station specification according to the operating environment.



Ethernet module

1. On the "Connection Destination Simple Setting Connection" screen, select [Other Connection Method] and click the [Other Connection Method (Open the Specify Connection Destination window)] button.

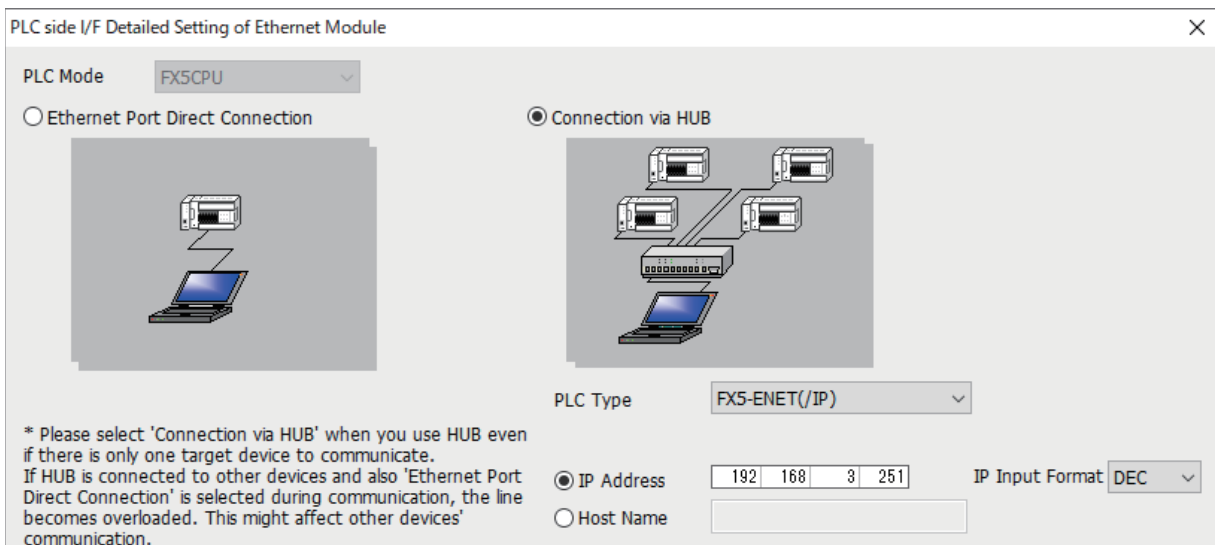


2. Select "Ethernet Board" for "PC side I/F".

3. Select "Ethernet module" for PLC side I/F, and double-click it.

Input the IP address or host name of the Ethernet module in the "PLC side I/F Detailed Setting of Ethernet Module" screen as shown below.

In case of host name, set the name specified in the Microsoft® Windows® hosts file.



4. Double-click “Other Station (Single Network)” in Other Station Setting.
Set the other station specifications according to the operating environment.

Other Station Detailed Setting

Check at Communication Time sec.

Retry Times Times

- Unable to cancel the operation during communication retrying.
- It may take the following time up to maximum to show the communication error.
Check at Communication Time * 3 * (Retry Times + 1)

Searching Ethernet-equipped modules on Network

In the case of GX Works3, with connections using a hub, you can search for and display the list of Ethernet-equipped modules connected to the same hub as personal computer (GX Works3).

[CPU module]

🖱️ Online⇒[Current Connection Destination]⇒Double-click [PLC side I/F Detailed Setting of PLC Module]⇒[Find]

PLC side I/F Detailed Setting of PLC Module

PLC Mode: FX5CPU

Ethernet Port Direct Connection
 Connection via HUB

* Please select 'Connection via HUB' when you use HUB even if there is only one target device to communicate. If HUB is connected to other devices and also 'Ethernet Port Direct Connection' is selected during communication, the line becomes overloaded. This might affect other devices' communication.

IP Address: 0 0 0 0
 IP Input Format: DEC

Host Name

Search for the FX5CPU on network.

Response Wait Time: 2 sec. Display Only CPU Type of Project(V) Find(S)

Selection IP Address Input

Search for FX5CPU on the same network. Unable to search for the following causes:

- No response within a specific time period.
- Connected via a router or subnet mask is different.
- 'No response to search for CPU module on network' is set in module parameter.

| | IP address | CPU Type | Label | Comment |
|---|---------------|----------|-------|---------|
| 1 | 192.168.3.250 | FX5UCPU | | |
| 2 | 192.168.3.251 | FX5UCPU | | |

OK Cancel

[Ethernet module]

🔍 Online⇒[Current Connection Destination]⇒Double-click [PLC side I/F Detailed Setting of Ethernet Module]⇒[Find]

PLC side I/F Detailed Setting of Ethernet Module

PLC Mode: FX5CPU

Ethernet Port Direct Connection

Connection via HUB

PLC Type: FX5-ENET(/IP)

* Please select 'Connection via HUB' when you use HUB even if there is only one target device to communicate. If HUB is connected to other devices and also 'Ethernet Port Direct Connection' is selected during communication, the line becomes overloaded. This might affect other devices' communication.

IP Address: 0 | 0 | 0 | 0 IP Input Format: DEC

Host Name

Search for Ethernet module on network.

Response Wait Time: 2 Seconds Display Only CPU Type of Project(V) Selection IP Address Input Find(S)

Search for Ethernet module on the same network. Unable to search for the following cases:
- No response within a specific time period.
- Connected via a router or subnet mask is different.

| | IP Address | PLC Type | Label | Comment |
|---|---------------|----------|-------|---------|
| 1 | 192.168.3.251 | FX5UCPU | | |
| 2 | 192.168.3.252 | FX5UJCPU | | |

OK Cancel

Target modules

- Ethernet-equipped modules connected to the same hub as GX Works3
- Ethernet-equipped modules connected to cascaded hubs

Point

When a search is performed from the Ethernet module, only the Ethernet-equipped modules of iQ-F Series will be targets.

The module selecting "Do Not Respond" in "Do Not Respond to CPU Module Search" in GX Works3 can be excluded from the target.

[CPU module]

Navigation window⇒[Parameter]⇒[FX5UJCPU] or [FX5UCPU]⇒[Module Parameter]⇒[Ethernet Port]⇒[Application Settings]⇒[Security]

[Ethernet module]

Navigation window⇒[Parameter]⇒[Module Information]⇒[FX5-ENET] or [FX5-ENET/IP]⇒[Application Settings]⇒[Security]

Precautions

If multiple Ethernet-equipped modules with the same IP address are found in the list, check the IP address parameters for the Ethernet-equipped modules. Starting communication with the IP address duplicated will cause a communication error.

Ethernet-equipped modules cannot be searched in the following cases

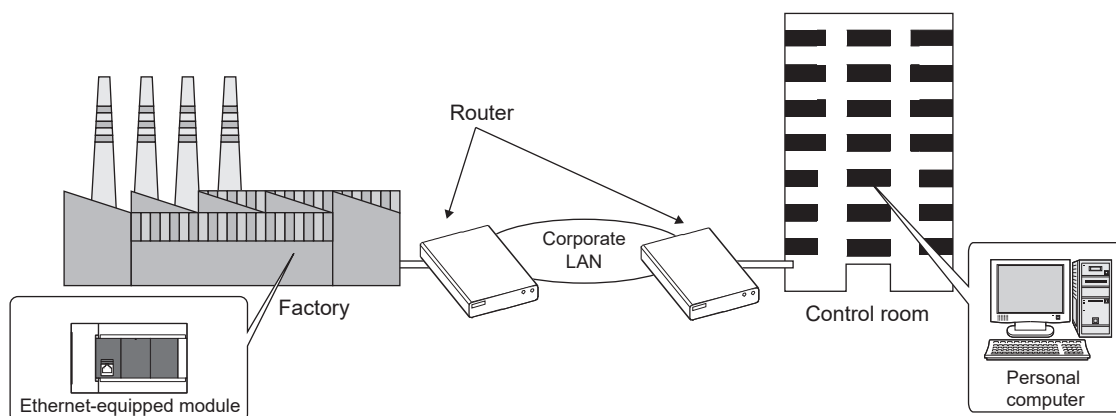
- The Ethernet-equipped module whose address is denied by the IP filter function cannot be searched.
- Ethernet-equipped modules connected via router cannot be searched.
- Some Ethernet-equipped modules connected via wireless LAN may not be found since Ethernet communication may not be stable due to packet loss.
- Appropriate Ethernet-equipped modules may not be found if the service processing load is heavy.

Point

When the appropriate Ethernet-equipped module cannot be searched, increase the response waiting time value, or change the service processing counts in the service processing settings of the CPU parameters.

Communication via Router

Access via routers from built-in Ethernet port is available in an environment such as a corporate LAN.*1



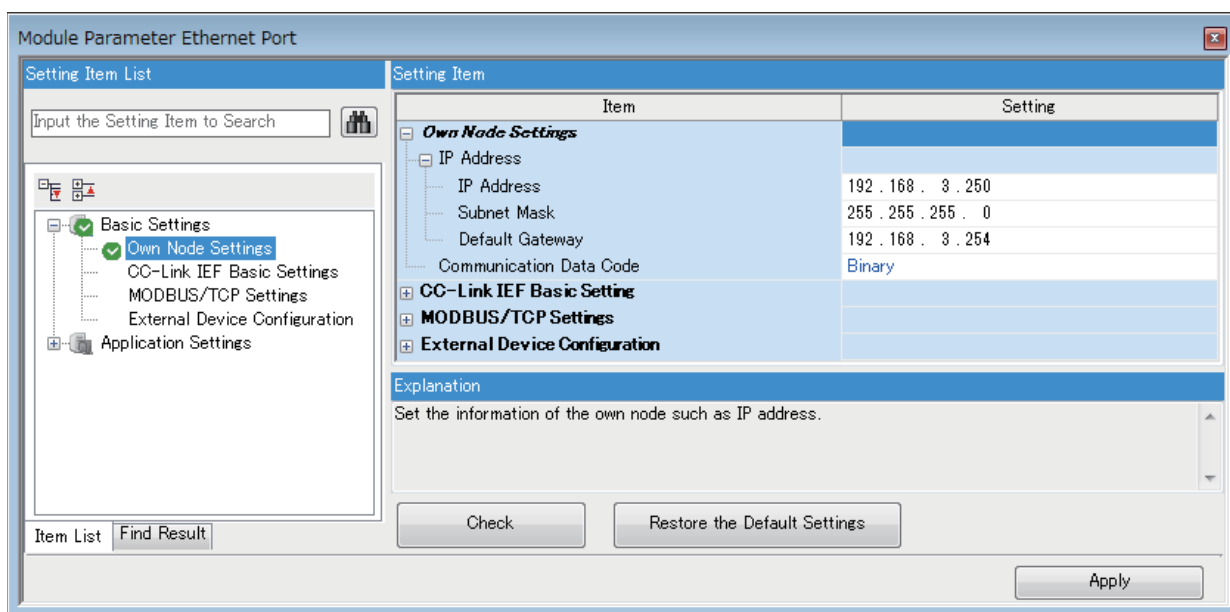
*1 Communication through routers is impossible for some functions. The following functions do not support communication via routers.

- Searching for Ethernet-equipped modules on the network

For access via router, set the subnet mask pattern and default gateway IP address in addition to IP address as per Page 31 Setting module parameters.

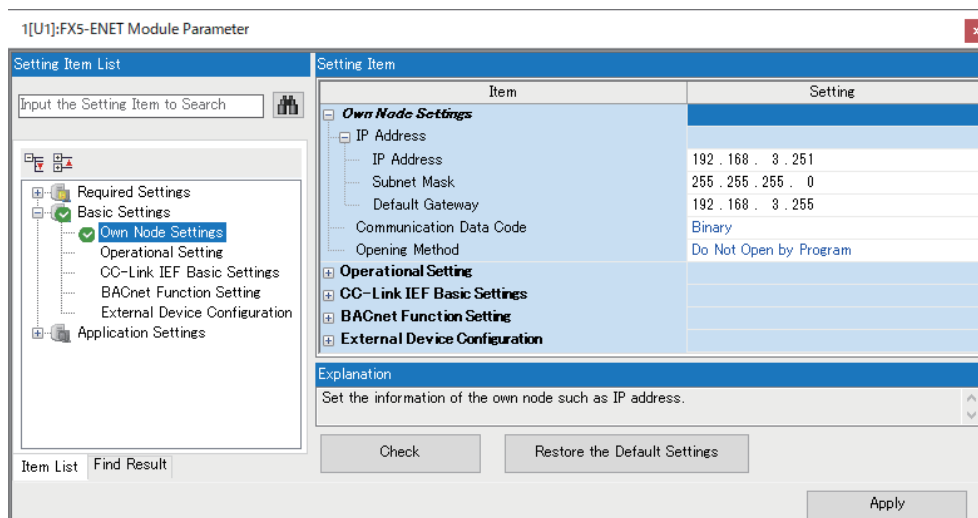
[CPU module]

Navigation window ⇒ [Parameter] ⇒ [FX5UJCPU] or [FX5UCPU] ⇒ [Module Parameter] ⇒ [Ethernet Port] ⇒ [Basic Settings] ⇒ [Own Node Settings]



[Ethernet module]

Navigation window ⇒ [Parameter] ⇒ [Module Information] ⇒ [FX5-ENET] or [FX5-ENET/IP] ⇒ [Basic Settings] ⇒ [Own Node Settings]



Precautions

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 IP address duplication with the connected CPU search function.

KeepAlive check

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

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

If the connected device does not support the TCP KeepAlive function, the connection may be disconnected.

Connections exceeding the setting

Do not exceed the number of connections set in the Ethernet configuration settings of the parameters. If the personal computer makes a number of TCP connections that exceeds the set number, the following state results depending on the application.

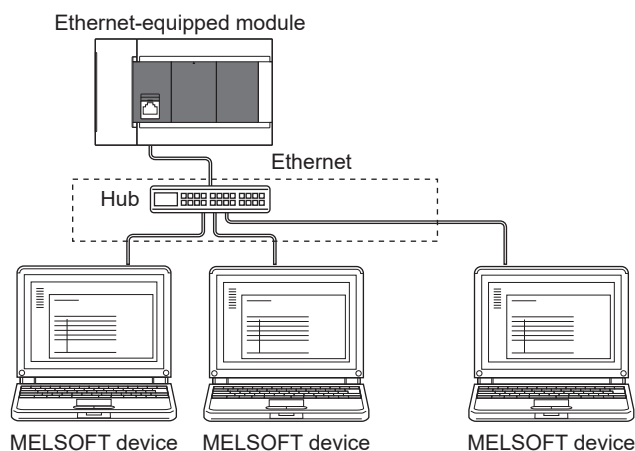
- Timeout error detection time gets extended.
- Unexpected timeout error occurs in any of the communicating devices.

Retransmission in case of TCP connection

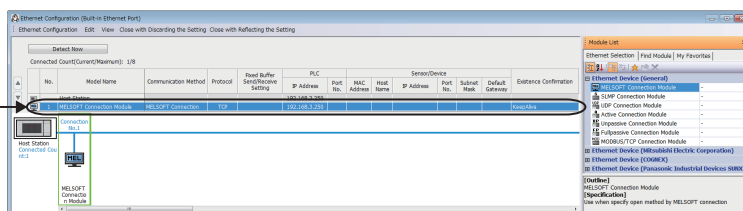
In the TCP connection, if no TCP ACK response is returned from the external device in response to a transmission, resending will be performed. Resending will be performed twelve times, starting 3 seconds after the first transmission, and then 6, 12, 24, 48, 60, and thereafter every 60 seconds. When no TCP ACK response is returned within 60 seconds after the last retransmission, the external device is regarded as faulty and the connection is disconnected. (The connection is disconnected in total of 573 seconds as external device fault.)

TCP MELSOFT connection

In case of TCP communication with multiple MELSOFT devices (GX Works3, etc.), set the same number of MELSOFT devices in the unit parameters.



Set the same number of devices as MELSOFT devices



Point

When all MELSOFT devices start communicating at the same time, devices may fail to communicate because of the congestion in communication. 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.

Remote STOP

When remote STOP is executed using the engineering tool, execute remote RUN before turning OFF the power of the CPU module.

Function that cannot be used via hub

In case of connection via a hub, the Ethernet diagnostic function cannot be used. When using the Ethernet diagnostic function, directly connect the Ethernet-equipped module and GX Works3.

5 SLMP FUNCTION

SLMP (Seamless Message Protocol) is a protocol for accessing SLMP-compatible devices from an external device (such as personal computer or GOT) using TCP or UDP through Ethernet.

SLMP communication can be used for the target device that can send and receive messages in the control procedure of SLMP.

The Ethernet port of the Ethernet-equipped module can be used as an SLMP server. The Ethernet port of the CPU module can be used also as an SLMP client.

The message format of SLMP is 3E/1E frames.

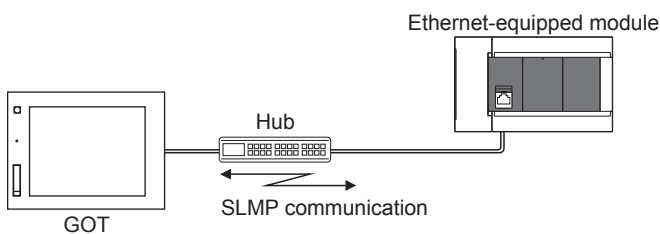
[Server function]

The Ethernet-equipped module performs data processing and data transfer based on a request message (command) from external devices.

[Client function]

Request messages can be sent to external devices and response messages from external devices can be received by dedicated instructions. The SLMP client function can be used only for the CPU module, and the SLMP frame transmission is available only with the 3E frame.

For details on the SLMP function, refer to the MELSEC iQ-F FX5 User's Manual (SLMP).



Point

The message format of each SLMP is the same as that of the following MC protocols frames.

- 3E frame: QnA compatible 3E frame of MC protocol
- 1E frame: A compatible 1E frame of MC protocol

The external devices used with the above MC protocols can be connected to SLMP compatible devices.

Restriction

The 1E frame can be used only for FX5U/FX5UC CPU module and Ethernet module.

Applications

- Ethernet-equipped module device data can be read and written using SLMP (3E/1E frames) from external devices.
- Ethernet-equipped module operation monitoring, data analysis, and production control is possible from external devices by reading and writing device data.
- With the remote password function, unauthorized access from the outside can be prevented. (Page 196 Remote Password)

Data communication procedures

1. Connecting cables and external devices

Make the connections for SLMP communication. (☞ Page 20 Connection specifications)

2. Setting parameters

Configure the module parameters with the engineering tool. (☞ Page 46 Setting Method)

3. Writing to the Ethernet-equipped module

Write the parameters set in the Ethernet-equipped module. Turn power OFF → ON or perform reset to enable the parameters.

4. Initial process state check

After setting the module parameter, check that the Ethernet-equipped module has been normally initialized.

- CPU module: Initial status (SD10683.b0): ON
- Ethernet module: Initial status (Un\G158.b0): ON

5. SLMP communication^{*1}

[Server function]^{*2}

SLMP messages from external devices are received.

[Client function]^{*3}

SLMP messages are sent to external devices. (☞ Page 61 Sending the SLMP frame)

*1 The connection is established/disconnected by the system.

*2 Operation by user is not required.

*3 The SLMP client function can be used only for the CPU module, and the SLMP frame transmission is available only with the 3E frame.

Point

Access through routers is also available. In order to configure this, set the subnet mask pattern and default gateway IP address. (☞ Page 39 Communication via Router)

5.1 Specifications

Communication specifications

Communication by the SLMP function is implemented with the following specifications, and they can be configured in module parameters in the GX Works3.

| Item | | Specification | |
|-----------------------------|-------------------------------|---|--------------------------------------|
| | | CPU module | Ethernet module |
| Transmission specifications | Data transfer speed | 100/10 Mbps | |
| | Communication mode | Full-duplex or half-duplex ^{*1} | |
| | Interface | RJ45 connector | |
| | Transmission method | Base band | |
| | Maximum segment length | 100 m (length between hub and node) ^{*2} | |
| | Number of cascade connections | 100BASE-TX | 2 levels maximum ^{*3} |
| 10BASE-T | | 4 levels maximum ^{*3} | |
| Number of ports | | 1 port | 2 port |
| Number of connections | | 8 connections maximum ^{*4} | 32 connections maximum ^{*5} |

*1 IEEE802.3x flow control is not supported.

*2 For maximum segment length (length between hubs), consult the manufacturer of the hub used.

*3 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.

*4 Maximum of 8 connections including SLMP server, MELSOFT connections, socket communication, MODBUS/TCP communication, and predefined protocol support.

*5 Up to 32 connections for SLMP server, MELSOFT connection, socket communication, simple CPU communication and BACnet


Point

Hubs with 100BASE-TX or 10BASE-T ports can be connected.

A personal computer can also be directly connected without using a hub.

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

Link specifications

The method for calculating the link time of the CPU module is shown below.
 For applicable commands and devices, refer to  Page 48 SLMP Commands.

Point 

The link time between Ethernet modules varies depending on the usage of other intelligent function modules.

Link time

Calculate the minimum processing time for transmission by SLMP with the following formula.

However, the processing time may further increase due to the network load (line congestion), window size of connected devices, the number of simultaneously used connections, and the system configuration. Use the result of this formula as a guideline value of the processing time, when only 1 connection is being used.

- Minimum processing time for communication by SLMP (for batch read, batch write)

$$Tfs = Ke + (Kdt \times Df) + Scr \times \text{number of scans required for processing} + \text{other device ACK processing time}$$

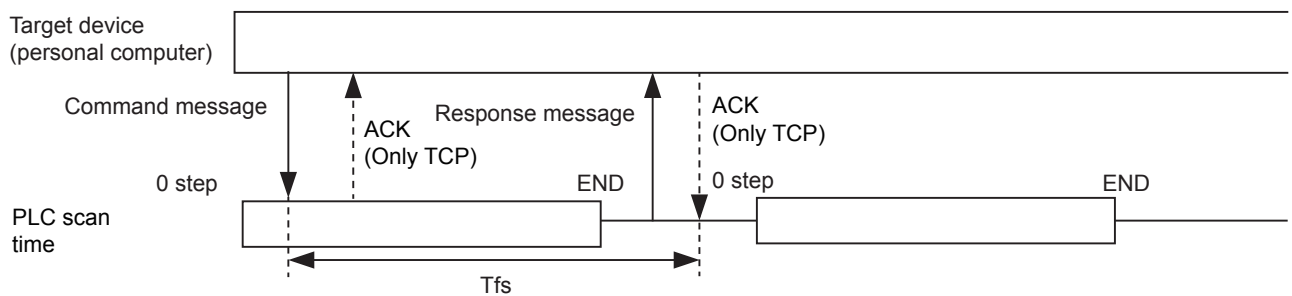
Tfs: The time from when the personal computer receives the request data until the PLC finishes processing (unit: ms)^{*1}

Ke, Kdt: Constants (refer to the table below)

Df: Number of words of requested data + Number of words of response data (application data portion)

Scr: Scan time

*1 The following shows the timing from when the personal computer receives the request data until the PLC finishes processing.



| Communication content | | For TCP/IP communication | | For UDP/IP communication | |
|-----------------------|--|--------------------------|-------|--------------------------|-------|
| | | Ke | Kdt | Ke | Kdt |
| Batch read | When communicating as ASCII code data | 1 | 0.001 | 1 | 0.001 |
| | When communicating as binary code data | 1 | 0.001 | 1 | 0.001 |
| Batch writing | When communicating as ASCII code data | 1 | 0.001 | 1 | 0.001 |
| | When communicating as binary code data | 1 | 0.001 | 1 | 0.001 |

Ex.

[Calculation example 1]

When performing TCP/IP communication with a personal computer and reading 32 points (devices) of data from the own station's data register (D) as binary code data, using SLMP communication, the time from when the computer request data is received until processing is finished (unit: ms)

Connected station scan time is 40 ms.

$$Tfs = 1 + (0.001 \times 32) + 40 \times 1 + \text{other device ACK processing time}$$

[Calculation example 2]

When performing TCP/IP communication with a personal computer and writing 32 points (devices) of data to the own station's data register (D) as binary code data, using SLMP communication, the time from when the computer request data is received until processing is finished (unit: ms)

Connected station scan time is 40 ms.

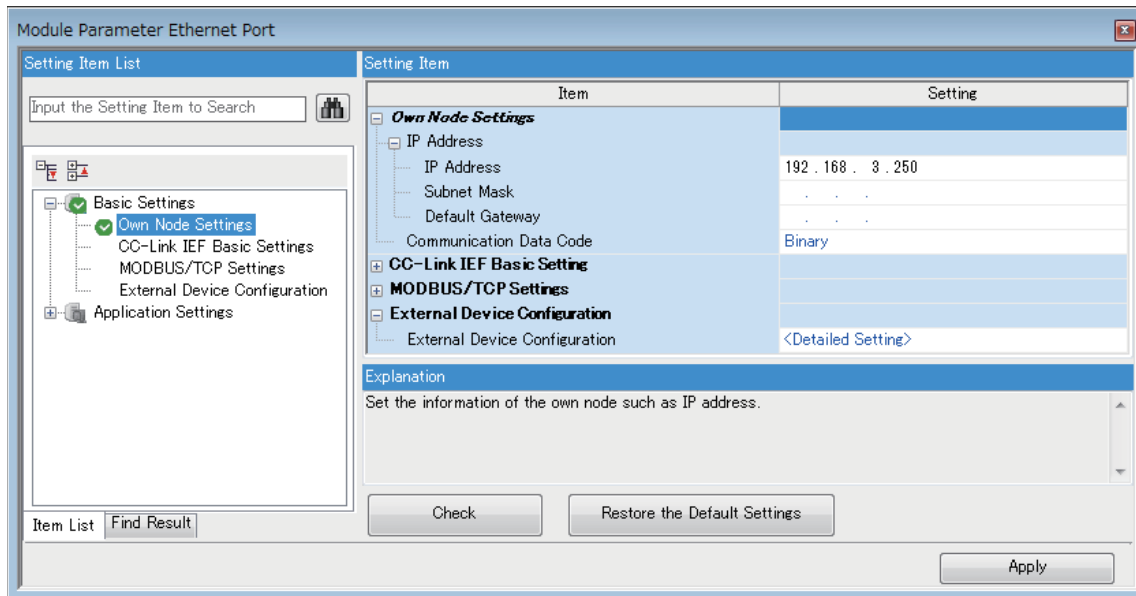
$$Tfs = 1 + (0.001 \times 32) + 40 \times 1 + \text{other device ACK processing time}$$

5.2 Setting Method

The following shows the configuration for communication by SLMP.

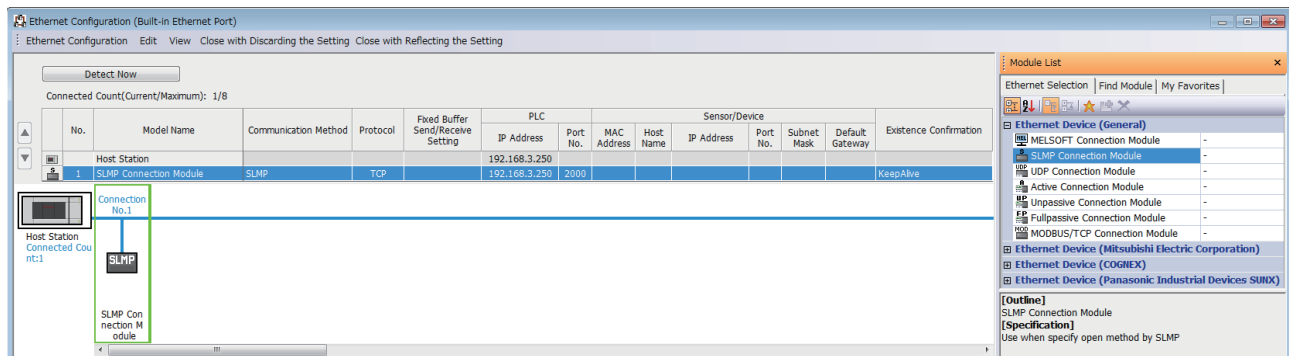
CPU module

Navigation window⇒[Parameter]⇒[FX5UJCPU] or [FX5UCPU]⇒[Module Parameter]⇒[Ethernet Port]⇒[Basic Settings]⇒[Own Node Settings]



1. Under "Own Node Settings", set "IP Address" setting and "Communication Data Code".
2. Configure the connection for the SLMP connection.

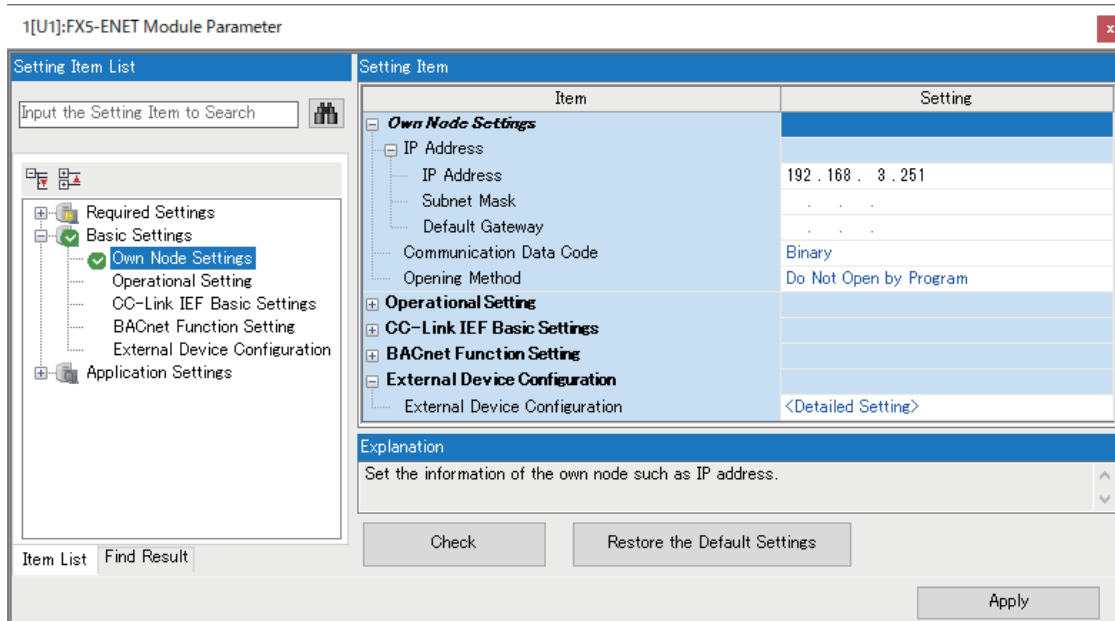
Navigation window⇒[Parameter]⇒[FX5UJCPU] or [FX5UCPU]⇒[Module Parameter]⇒[Ethernet Port]⇒[Basic Settings]⇒[External Device Configuration]⇒[Detailed Setting]⇒[Ethernet Configuration (Built-in Ethernet Port)] screen



3. Drag and drop "SLMP Connection Module" under "Module List" to the left side of the screen. Select protocol (TCP or UDP) that matches the other device in "Protocol". Set the host station port number (setting range: 1 to 5548, 5570 to 65534) for the "Port No." and the IP address of the target device (only UDP). Do not specify 5549 to 5569 for the host station port numbers because these ports are reserved for the system.

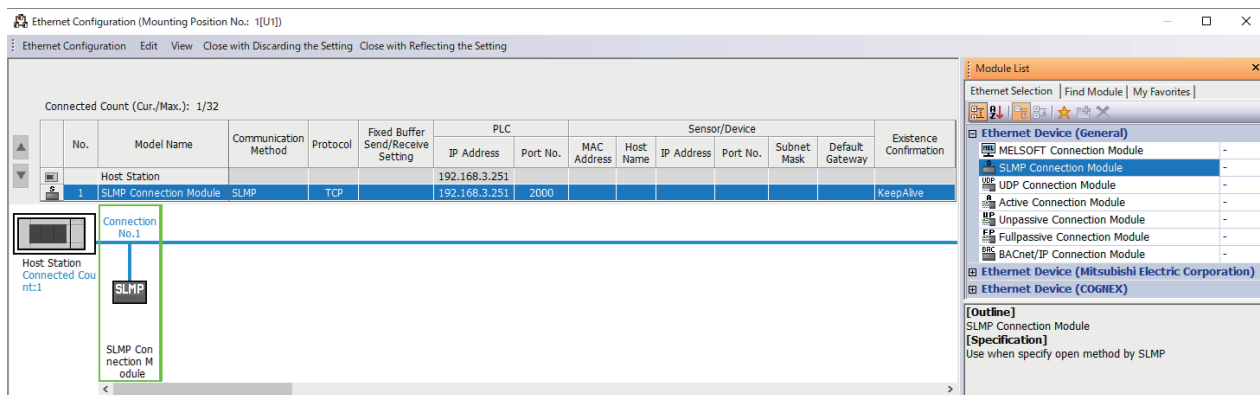
Ethernet module

Navigation window⇒[Parameter]⇒[Module Information]⇒[FX5-ENET] or [FX5-ENET/IP]⇒[Basic Settings]⇒[Own Node Settings]



1. Under "Own Node Settings", set "IP Address" setting items and "Communication Data Code".
2. Configure the connection for the SLMP connection.

Navigation window⇒[Parameter]⇒[Module Information]⇒[FX5-ENET] or [FX5-ENET/IP]⇒[Basic Settings]⇒[External Device Configuration]⇒[Detailed Setting]⇒[Ethernet Configuration (Mounting Position No.: n[Un])] screen



3. Drag and drop "SLMP Connection Module" under "Module List" to the left side of the screen. Select the protocol (TCP or UDP) appropriate to the external device in "Protocol." Set the own station port number (setting range: 1 to 5548 and 5570 to 65534) in "Port No.". Do not specify the port numbers 5549 to 5569 because these ports are used by the system.

5.3 SLMP Commands

For details on the SLMP commands, refer to the MELSEC iQ-F FX5 User's Manual (SLMP).

Command list

The following SLMP commands can be executed on the Ethernet-equipped module.

3E frames

■CPU module

| Name | Command | Sub-commands | Processing content | Number of points processed per communication |
|----------------------|---------|--------------|---|--|
| Device Read (Batch) | 0401H | 0001H | This command reads data from a bit device or word device in units of 1 bit. | ASCII: 1792 points BIN: 3584 points |
| | | 0000H | <ul style="list-style-type: none"> This command reads data from bit devices in units of 16 bits. This command reads data from word devices in units of 1 word. | ASCII: 480 words (7680 points) BIN: 960 words (15360 points) |
| | | 0081H | <ul style="list-style-type: none"> This command reads data from the buffer memory in intelligent function modules in units of 1 bit. This command reads data from devices indirectly specified by index registers in units of 1 bit. | ASCII: 1792 points BIN: 3584 points |
| | | 0080H | <ul style="list-style-type: none"> This command reads data from the buffer memory in intelligent function modules in units of 1 word. This command reads data from devices indirectly specified by index registers in units of 1 word. | ASCII: 480 words (7680 points) BIN: 960 words (15360 points) |
| | | 0083H | <ul style="list-style-type: none"> This command reads data from the buffer memory in intelligent function modules in units of 1 bit. This command reads data from devices indirectly specified by index registers in units of 1 bit. | ASCII: 1792 points BIN: 3584 points |
| | | 0082H | <ul style="list-style-type: none"> This command reads data from the buffer memory in intelligent function modules in units of 1 word. This command reads data from devices indirectly specified by index registers in units of 1 word. | ASCII: 480 words (7680 points) BIN: 960 words (15360 points) |
| Device Write (Batch) | 1401H | 0001H | This command writes data to bit devices in units of 1 bit. | ASCII: 1792 points BIN: 3584 points |
| | | 0000H | <ul style="list-style-type: none"> This command writes data to bit devices in units of 16 bits. This command writes data to word devices in units of 1 word. | ASCII: 480 words (7680 points) BIN: 960 words (15360 points) |
| | | 0081H | <ul style="list-style-type: none"> This command writes data to the buffer memory in intelligent function modules and SLMP-compatible devices in units of 1 bit. Bit devices, word devices, and buffer memory are indirectly specified by index registers. | ASCII: 1792 points BIN: 3584 points |
| | | 0080H | This command writes data to the buffer memory in intelligent function modules and SLMP-compatible devices in units of 1 word (16 bits). | ASCII: 480 words (7680 points) BIN: 960 words (15360 points) |
| | | 0083H | This command writes data to the buffer memory in intelligent function modules and SLMP-compatible devices in units of 1 bit. | ASCII: 1972 points BIN: 3584 points |
| | | 0082H | This command writes data to the buffer memory in intelligent function modules and SLMP-compatible devices in units of 1 word (16 bits). | ASCII: 480 words (7680 points) BIN: 960 words (15360 points) |
| Device Read Random | 0403H | 0000H | This command reads data from word devices in units of 1 word or 2 words by randomly specifying device numbers. | ASCII: (Word access points + double word access points) × 2 ≤ 192 BIN: Word access points + double word access points ≤ 192 |

| Name | Command | Sub-commands | Processing content | Number of points processed per communication |
|---------------------|---------|--------------|---|--|
| Device Read Random | 0403H | 0080H | This command reads data from the buffer memory in intelligent function modules and SLMP-compatible devices in units of 1 word (16 bits). | ASCII: (Word access points + double word access points) × 4 ≤ 192 BIN: (Word access points + double word access points) × 2 ≤ 192 |
| | | 0082H | This command reads data from the buffer memory in intelligent function modules and SLMP-compatible devices in units of 1 word (16 bits). | ASCII: (Word access points + double word access points) × 4 ≤ 192 BIN: (Word access points + double word access points) × 2 ≤ 192 |
| Device Write Random | 1402H | 0001H | This command writes data to bit devices in units of 1 bit by randomly specifying device numbers. | ASCII: 94 points BIN: 188 points |
| | | 0000H | <ul style="list-style-type: none"> This command writes data to bit devices in units of 16 bits by randomly specifying device numbers. This command writes data to word devices in units of 1 word or 2 words by randomly specifying device numbers. | ASCII: ((Word access points) × 12+ (double-word access points) × 14) × 2 ≤ 1920 BIN: (Word access points) × 12+ (double-word access points) × 14 ≤ 1920 |
| | | 0081H | <ul style="list-style-type: none"> This command writes data to the buffer memory in intelligent function modules and SLMP-compatible devices in units of 1 bit. Buffer memory is indirectly specified by index registers. | ASCII: 47 points BIN: 94 points |
| | | 0080H | This command writes data to the buffer memory in intelligent function modules and SLMP-compatible devices in units of 1 word (16 bits) or 2 words. | ASCII: ((Word access points) × 12+ (double-word access points) × 14) × 4 ≤ 1920 BIN: ((Word access points) × 12+ (double-word access points) × 14) × 2 ≤ 1920 |
| | | 0083H | This command writes data to the buffer memory in intelligent function modules and SLMP-compatible devices in units of 1 bit. | ASCII: 47 points BIN: 94 points |
| | | 0082H | This command writes data to the buffer memory in intelligent function modules and SLMP-compatible devices in units of 1 word (16 bits) or 2 words. | ASCII: ((Word access points) × 12+ (double-word access points) × 14) × 4 ≤ 1920 BIN: ((Word access points) × 12+ (double-word access points) × 14) × 2 ≤ 1920 |
| Device Read Block | 0406H | 0000H | With n points of bit devices and word devices as 1 block, this command reads data by randomly specifying multiple blocks. (When bit devices are specified, 1 point is 16 bits.) | ASCII: (Number of word device blocks + number of bit device blocks) × 2 ≤ 120 and (Total points of each blocks of word device + total points of each blocks of bit device) × 2 ≤ 960 BIN: Number of word device blocks + number of bit device blocks ≤ 120 and Total points of each blocks of word device + total points of each blocks of bit device ≤ 960 |

| Name | Command | Sub-commands | Processing content | Number of points processed per communication |
|--------------------|---------|--------------|--|--|
| Device Read Block | 0406H | 0080H | With n points of buffer memory in intelligent function modules and SLMP-compatible devices as 1 block, this command reads data by randomly specifying multiple blocks. (When bit devices are specified, 1 point is 16 bits.) | ASCII: (Number of word device blocks + number of bit device blocks) × 4 ≤ 120 and (Total points of each blocks of word device + total points of each blocks of bit device) × 2 ≤ 960 BIN: (Number of word device blocks + number of bit device blocks) × 2 ≤ 120 and Total points of each blocks of word device + total points of each blocks of bit device ≤ 960 |
| | | 0082H | With n points of buffer memory in intelligent function modules and SLMP-compatible devices as 1 block, this command reads data by randomly specifying multiple blocks. | ASCII: (Number of word device blocks + number of bit device blocks) × 4 ≤ 120 and (Total points of each blocks of word device + total points of each blocks of bit device) × 2 ≤ 960 BIN: (Number of word device blocks + number of bit device blocks) × 2 ≤ 120 and Total points of each blocks of word device + total points of each blocks of bit device ≤ 960 |
| Device Write Block | 1406H | 0000H | With n points of bit devices and word devices as 1 block, this command writes data by randomly specifying multiple blocks. (When bit devices are specified, 1 point is 16 bits.) | ASCII: (Number of word device blocks + number of bit device blocks) × 2 ≤ 120 and ((Number of word device blocks + number of bit device blocks) × 4 + Total points of each blocks of word device + total points of each blocks of bit device) × 2 ≤ 760 BIN: Number of word device blocks + number of bit device blocks ≤ 120 and (Number of word device blocks + number of bit device blocks) × 4 + Total points of each blocks of word device + total points of each blocks of bit device ≤ 760 |
| | | 0080H | With n points of buffer memory in intelligent function modules and SLMP-compatible devices as 1 block, this command writes data by randomly specifying multiple blocks. (When bit devices are specified, 1 point is 16 bits.) | ASCII: (Number of word device blocks + number of bit device blocks) × 4 ≤ 120 and ((Number of word device blocks + number of bit device blocks) × 4 + Total points of each blocks of word device + total points of each blocks of bit device) × 2 ≤ 760 BIN: (Number of word device blocks + number of bit device blocks) × 2 ≤ 120 and (Number of word device blocks + number of bit device blocks) × 4 + Total points of each blocks of word device + total points of each blocks of bit device ≤ 760 |

| Name | Command | Sub-commands | Processing content | Number of points processed per communication |
|--------------------|---------|--------------|---|--|
| Device Write Block | 1406H | 0082H | With n points of buffer memory in intelligent function modules and SLMP-compatible devices as 1 block, this command writes data by randomly specifying multiple blocks. | ASCII: (Number of word device blocks + number of bit device blocks) × 4 ≤ 120 and ((Number of word device blocks + number of bit device blocks) × 4 + Total points of each blocks of word device + total points of each blocks of bit device) × 2 ≤ 760 BIN: (Number of word device blocks + number of bit device blocks) × 2 ≤ 120 and (Number of word device blocks + number of bit device blocks) × 4 + Total points of each blocks of word device + total points of each blocks of bit device ≤ 760 |
| Remote Run | 1001H | 0000H | This command performs a remote RUN request for a device. | — |
| Remote Stop | 1002H | 0000H | This command performs a remote STOP request for a device. | — |
| Remote Pause | 1003H | 0000H | This command performs a remote PAUSE request for a device. | — |
| Remote Latch Clear | 1005H | 0000H | This command performs a remote latch clear request when the device is in the STOP state. | — |
| Remote Reset | 1006H | 0000H | This command performs a remote reset request to reset the device error stop state. | — |
| Read Type Name | 0101H | 0000H | This command reads the processor module name code (processor type) of a device. | — |
| Self-Test | 0619H | 0000H | This command checks if normal communication is possible. | — |
| Clear Error | 1617H | 0001H | This command batch clears all errors and turns off the LED. | — |
| Password Lock | 1631H | 0000H | This command sets to the locked status from the unlocked status by specifying the remote password. (Sets the device to the state where communication is not possible.) | — |
| Password Unlock | 1630H | 0000H | This command sets to the unlocked status from the locked status by specifying the remote password. (Sets the device to the state where communication is possible.) | — |

■ Ethernet module

| Name | Command | Sub-commands | Processing content | Number of points processed per communication |
|----------------------------------|---------|--------------|---|--|
| Device Read (Batch) | 0401H | 0001H | This command reads data from a bit device in units of 1 bit. | ASCII: 1792 points BIN: 3584 points |
| | | 0000H | <ul style="list-style-type: none"> This command reads data from bit devices in units of 16 bits. This command reads data from word devices in units of 1 word. | ASCII: 480 words (7680 points) BIN: 960 words (15360 points) |
| | | 0081H | The CPU devices (bit devices) can be directly accessed. This command performs the same processing as the batch read (command: 0401H) and subcommand: 0001H. | ASCII: 1792 points BIN: 3584 points |
| | | 0080H | <ul style="list-style-type: none"> This command reads data from the buffer memory in intelligent function modules in units of 1 word. The CPU devices (bit devices and word devices) can be directly accessed. This command performs the same processing as the batch read (command: 0401) and subcommand: 0000H. | ASCII: 480 words (7680 points) BIN: 960 words (15360 points) |
| | | 0083H | The CPU devices (bit devices) can be directly accessed. This command performs the same processing as the batch read (command: 0401H) and subcommand: 0001H. | ASCII: 1792 points BIN: 3584 points |
| | | 0082H | <ul style="list-style-type: none"> This command reads data from the buffer memory in intelligent function modules in units of 1 word. The CPU devices (bit devices and word devices) can be directly accessed. This command performs the same processing as the batch read (command: 0401) and subcommand: 0000H. | ASCII: 480 words (7680 points) BIN: 960 words (15360 points) |
| Device Write (Batch) | 1401H | 0001H | This command writes data to bit devices in units of 1 bit. | ASCII: 1792 points BIN: 3584 points |
| | | 0000H | <ul style="list-style-type: none"> This command writes data to bit devices in units of 16 bits. This command writes data to word devices in units of 1 word. | ASCII: 480 words (7680 points) BIN: 949 words (15184 points) |
| | | 0081H | The CPU devices (bit devices) can be directly accessed. This command performs the same processing as the batch write (command: 1401H) and subcommand: 0001H. | ASCII: 1792 points BIN: 3584 points |
| | | 0080H | <ul style="list-style-type: none"> This command writes data to the buffer memory in intelligent function modules and SLMP-compatible devices in units of 1 word (16 bits). The CPU devices (bit devices and word devices) can be directly accessed. This command performs the same processing as the batch write (command: 1401H) and subcommand: 0000H. | ASCII: 480 words (7680 points) BIN: 949 words (15184 points) |
| | | 0083H | The CPU devices (bit devices) can be directly accessed. This command performs the same processing as the batch write (command: 1401H) and subcommand: 0001H. | ASCII: 1972 points BIN: 3584 points |
| | | 0082H | <ul style="list-style-type: none"> This command writes data to the buffer memory in intelligent function modules and SLMP-compatible devices in units of 1 word (16 bits). The CPU devices (bit devices and word devices) can be directly accessed. This command performs the same processing as the batch write (command: 1401H) and subcommand: 0000H. | ASCII: 480 words (7680 points) BIN: 949 words (15184 points) |
| Device Read Random* ¹ | 0403H | 0000H | This command reads data from bit devices and word devices in units of 1 word or 2 words by randomly specifying device numbers. | ASCII: (Word access points + double word access points) × 2 ≤ 192 BIN: Word access points + double word access points ≤ 123 |
| | | 0080H | <ul style="list-style-type: none"> This command reads data from the buffer memory in intelligent function modules in units of 1 word (16 bits) or 2 words. This command reads data from devices indirectly specified by index registers in units of 1 word or 2 words. The CPU devices (bit devices and word devices) can be directly accessed. This command performs the same processing as the random read (command: 0403H) and subcommand: 0000H. | ASCII: (Word access points + double word access points) × 4 ≤ 192 BIN: (Word access points + double word access points) × 2 ≤ 192 |
| | | 0082H | <ul style="list-style-type: none"> This command reads data from the buffer memory in intelligent function modules in units of 1 word (16 bits) or 2 words. This command reads data from devices indirectly specified by index registers in units of 1 word or 2 words. The CPU devices (bit devices and word devices) can be directly accessed. This command performs the same processing as the random read (command: 0403H) and subcommand: 0000H. | ASCII: (Word access points + double word access points) × 4 ≤ 192 BIN: (Word access points + double word access points) × 2 ≤ 192 |

| Name | Command | Sub-commands | Processing content | Number of points processed per communication |
|-----------------------------------|---------|--------------|--|--|
| Device Write Random ^{*2} | 1402H | 0001H | This command writes data to bit devices in units of 1 bit by randomly specifying device numbers. | ASCII: 94 points BIN: 188 points |
| | | 0000H | <ul style="list-style-type: none"> This command writes data to bit devices in units of 16 bits by randomly specifying device numbers. This command writes data to word devices in units of 1 word or 2 words by randomly specifying device numbers. | ASCII: $((\text{Word access points}) \times 12 + (\text{double-word access points}) \times 14) \times 2 \leq 1920$ BIN: $(\text{Word access points}) \times 18 + (\text{double-word access points}) \times 20 \leq 1920$ |
| | | 0081H | The CPU devices (bit devices) can be directly accessed. This command performs the same processing as the random write (command: 1402H) and subcommand: 0001H. | ASCII: 47 points BIN: 94 points |
| | | 0080H | <ul style="list-style-type: none"> This command writes data to the buffer memory in intelligent function modules in units of 1 word (16 bits) or 2 words. This command writes data from devices indirectly specified by index registers in units of 1 word or 2 words. The CPU devices (bit devices and word devices) can be directly accessed. This command performs the same processing as the random write (command: 1402H) and subcommand: 0000H. | ASCII: $((\text{Word access points}) \times 12 + (\text{double-word access points}) \times 14) \times 4 \leq 1920$ BIN: $((\text{Word access points}) \times 12 + (\text{double-word access points}) \times 14) \times 2 \leq 1920$ |
| | | 0083H | The CPU devices (bit devices) can be directly accessed. This command performs the same processing as the random write (command: 1402H) and subcommand: 0001H. | ASCII: 47 points BIN: 94 points |
| | | 0082H | <ul style="list-style-type: none"> This command writes data to the buffer memory in intelligent function modules in units of 1 word (16 bits) or 2 words. This command writes data from devices indirectly specified by index registers in units of 1 word or 2 words. The CPU devices (bit devices and word devices) can be directly accessed. This command performs the same processing as the random write (command: 1402H) and subcommand: 0000H. | ASCII: $((\text{Word access points}) \times 12 + (\text{double-word access points}) \times 14) \times 4 \leq 1920$ BIN: $((\text{Word access points}) \times 12 + (\text{double-word access points}) \times 14) \times 2 \leq 1920$ |
| Device Read Block ^{*3} | 0406H | 0000H | With n points of bit devices and word devices as 1 block, this command reads data by randomly specifying multiple blocks. (When bit devices are specified, 1 point is 16 bits.) | ASCII: $(\text{Number of word device blocks} + \text{number of bit device blocks}) \times 2 \leq 120$ and $(\text{Total points of each blocks of word device} + \text{total points of each blocks of bit device}) \times 2 \leq 960$ BIN: $\text{Number of word device blocks} + \text{number of bit device blocks} \leq 120$ and $\text{Total points of each blocks of word device} + \text{total points of each blocks of bit device} \leq 960$ |

| Name | Command | Sub-commands | Processing content | Number of points processed per communication |
|---------------------------------|---------|--------------|---|--|
| Device Read Block ³ | 0406H | 0080H | <ul style="list-style-type: none"> With n points of buffer memory in intelligent function modules as 1 block, this command reads data by randomly specifying multiple blocks. (When bit devices are specified, 1 point is 16 bits.) This command performs the same processing as the multiple block batch read (command: 0406H) and subcommand: 0000H. | ASCII: (Number of word device blocks + number of bit device blocks) × 4 ≤ 120 and (Total points of each blocks of word device + total points of each blocks of bit device) × 2 ≤ 960 BIN: (Number of word device blocks + number of bit device blocks) × 2 ≤ 120 and Total points of each blocks of word device + total points of each blocks of bit device ≤ 960 |
| | | 0082H | <ul style="list-style-type: none"> With n points of buffer memory in intelligent function modules as 1 block, this command reads data by randomly specifying multiple blocks. This command performs the same processing as the multiple block batch read (command: 0406H) and subcommand: 0000H. | ASCII: (Number of word device blocks + number of bit device blocks) × 4 ≤ 120 and (Total points of each blocks of word device + total points of each blocks of bit device) × 2 ≤ 960 BIN: (Number of word device blocks + number of bit device blocks) × 2 ≤ 120 and Total points of each blocks of word device + total points of each blocks of bit device ≤ 960 |
| Device Write Block ³ | 1406H | 0000H | With n points of bit devices and word devices as 1 block, this command writes data by randomly specifying multiple blocks. (When bit devices are specified, 1 point is 16 bits.) | ASCII: (Number of word device blocks + number of bit device blocks) × 2 ≤ 120 and ((Number of word device blocks + number of bit device blocks) × 4 + Total points of each blocks of word device + total points of each blocks of bit device) × 2 ≤ 770 BIN: Number of word device blocks + number of bit device blocks ≤ 120 and (Number of word device blocks + number of bit device blocks) × 4 + Total points of each blocks of word device + total points of each blocks of bit device ≤ 770 |
| | | 0080H | <ul style="list-style-type: none"> With n points of buffer memory in intelligent function modules as 1 block, this command writes data by randomly specifying multiple blocks. (When bit devices are specified, 1 point is 16 bits.) This command performs the same processing as the multiple block batch write (command: 1406H) and subcommand: 0000H | ASCII: (Number of word device blocks + number of bit device blocks) × 4 ≤ 120 and ((Number of word device blocks + number of bit device blocks) × 4 + Total points of each blocks of word device + total points of each blocks of bit device) × 2 ≤ 770 BIN: (Number of word device blocks + number of bit device blocks) × 2 ≤ 120 and (Number of word device blocks + number of bit device blocks) × 4 + Total points of each blocks of word device + total points of each blocks of bit device ≤ 770 |

| Name | Command | Sub-commands | Processing content | Number of points processed per communication |
|----------------------------------|---------|--------------|---|--|
| Device Write Block ^{*3} | 1406H | 0082H | <ul style="list-style-type: none"> With n points of buffer memory in intelligent function modules as 1 block, this command writes data by randomly specifying multiple blocks. This command performs the same processing as the multiple block batch write (command: 1406H) and subcommand: 0000H | ASCII: (Number of word device blocks + number of bit device blocks) × 4 ≤ 120 and ((Number of word device blocks + number of bit device blocks) × 4 + Total points of each blocks of word device + total points of each blocks of bit device) × 2 ≤ 770 BIN: (Number of word device blocks + number of bit device blocks) × 2 ≤ 120 and (Number of word device blocks + number of bit device blocks) × 4 + Total points of each blocks of word device + total points of each blocks of bit device ≤ 770 |
| Remote Run | 1001H | 0000H | This command performs a remote RUN request for a device. | — |
| Remote Stop | 1002H | 0000H | This command performs a remote STOP request for a device. | — |
| Remote Pause | 1003H | 0000H | This command performs a remote PAUSE request for a device. | — |
| Remote Latch Clear | 1005H | 0000H | This command performs a remote latch clear request when the device is in the STOP state. | — |
| Remote Reset | 1006H | 0000H | This command performs a remote reset request to reset the device error stop state. | — |
| Read Type Name | 0101H | 0000H | This command reads the processor module name code (processor type) of a device. | — |
| Self-Test | 0619H | 0000H | This command checks if normal communication is possible. | — |

*1 The following devices (contacts and coils) cannot be specified.

- Timers (TS and TC)
- Retentive timers (STS and STC)
- Counters (CS and CC)
- Long counters (LCS and LCC)

*2 The following devices (contacts and coils) cannot be specified.

- Timers (TS and TC)
- Retentive timers (STS and STC)
- Counters (CS and CC)

*3 Double-word devices cannot be accessed.

1E frames

| Name | Command | Processing content | Number of points processed per communication |
|----------------------|-------------------|--|--|
| Collective read | 00H | Reads in 1-bit units from bit devices. | 256 points |
| | 01H | Reads in 16-bit units from bit devices.* ¹ | 32 words (512 points) |
| | | Reads in 1-word units from word devices.* ² | 64 points |
| Collective writing | 02H | Writes data in 1-bit units to bit devices. | 160 points |
| | 03H | Writes data in 16-bit units to bit devices.* ¹ | 10 words (160 points) |
| | | Writes data in 1-word units to bit devices.* ^{2*3} | 64 points |
| Test (random write) | 04H | Writes data in 1-bit units to a bit device by randomly specifying the device number. | 80 points |
| | 05H* ⁴ | Writes data in 16-bit units to a bit device by randomly specifying the device number. | 10 words (160 points) |
| | | Writes data in 1-word units to a word device by randomly specifying the device number. | 10 points |
| Remote Run | 13H | Issues a remote RUN/STOP request to the Ethernet-equipped module.* ⁵ | — |
| Remote Stop | 14H | | |
| PC type name reading | 15H | Reads Ethernet-equipped module model code. | — |
| Return | 16H | Returns received characters back to the computer as they are. | 254 bytes |

*1 The head bit device number must be a multiple of 16.

*2 When reading double word data by user program or the like, read/write 32 bits of data at a time. Specification across C199 and C200 is not allowed.

*3 Long counters cannot be specified.

*4 TS (timer (contact)), CS (counter (contact)) and LCS (long timer (contact)) cannot be specified in the Ethernet module.

*5 When the Remote STOP operation is performed, another communication requesting station cannot perform the remote RUN/STOP operation until the communication requesting station that performed the remote STOP operation performs the remote RUN operation. When the power to the CPU module is turned on, off and on in the remote RUN/STOP state, the remote RUN/STOP state is disabled. When the remote RUN/STOP state is disabled, the RUN/STOP switch setting of the Ethernet-equipped module is enabled.

Applicable devices

The following shows the available devices and device number ranges in commands used for the SLMP communication function.

3E frames

With 3E frames, specify the device to access with the "Device code" listed below.

| Classification | Device | Type | Device code* ¹ (Device specification format: Long) | | Device No. | Device compatibility* ² | | |
|----------------------|----------------------|---------------|--|----------------|---|------------------------------------|---------|---|
| | | | ASCII code | Binary code | | | | |
| Internal user device | Input | Bit | X* (X***) | 9CH (9C00H) | Specify in the range of device numbers of the module to access. | * ³ | ○ | |
| | Output | | Y* (Y***) | 9DH (9D00H) | | * ³ | ○ | |
| | Internal relay | | M* (M***) | 90H (9000H) | | Decimal | ○ | |
| | Latching relay | | L* (L***) | 92H (9200H) | | Decimal | ○ | |
| | Annunciator | | F* (F***) | 93H (9300H) | | Decimal | ○ | |
| | Edge relay | | V* (V***) | 94H (9400H) | | Decimal | — | |
| | Link relay | | B* (B***) | A0H (A000H) | | Hexadecimal | ○ | |
| | Step relay | | S* (S***) | 98H (9800H) | | Decimal | ○ | |
| | Data register | | Word | D* (D***) | | A8H (A800H) | Decimal | ○ |
| | Link register | W* (W***) | | B4H (B400H) | | Hexadecimal | ○ | |
| | Timer | Contact | Bit | TS (TS**) | | C1H (C100H) | Decimal | ○ |
| | | Coil | Bit | TC (TC**) | | C0H (C000H) | | ○ |
| | | Current value | Word | TN (TN**) | | C2H (C200H) | | ○ |
| | Long timer | Contact | Bit | — (LTS*) | | 51H (5100H) | Decimal | — |
| | | Coil | Bit | — (LTC*) | | 50H (5000H) | | — |
| | | Current value | Double Word | — (LTN*) | | 52H (5200H) | | — |
| | Retentive timer | Contact | Bit | SS (STS*) | | C7H (C700H) | Decimal | ○ |
| | | Coil | Bit | SC (STC*) | | C6H (C600H) | | ○ |
| | | Current value | Word | SN (STN*) | | C8H (C800H) | | ○ |
| | Long retentive timer | Contact | Bit | — (LSTS) | | 59H (5900H) | Decimal | — |
| Coil | | Bit | — (LSTC) | 58H (5800H) | — | | | |
| Current value | | Double Word | — (LSTN) | 5AH (5A00H) | — | | | |
| Counter | Contact | Bit | CS (CS**) | C4H (C400H) | Decimal | ○ | | |
| | Coil | Bit | CC (CC**) | C3H (C300H) | | ○ | | |

| Classification | Device | | Type | Device code*1 (Device specification format: Long) | | Device No. | | Device compatibility*2 |
|------------------------|-----------------------|---------------|-------------|---|----------------|---|-------------|------------------------|
| | | | | ASCII code | Binary code | | | |
| Internal user device | Counter | Current value | Word | CN (CN**) | C5H (C500H) | Specify in the range of device numbers of the module to access. | Decimal | ○ |
| | Long counter | Contact | Bit | — (LCS*) | 55H (5500H) | | | ○ |
| | | Coil | Bit | — (LCC*) | 54H (5400H) | | | ○ |
| | | Current value | Double Word | — (LCN*) | 56H (5600H) | | | ○ |
| | Link special relay | | Bit | SB (SB**) | A1H (A100H) | | Hexadecimal | ○ |
| | Link special register | | Word | SW (SW**) | B5H (B500H) | | | ○ |
| System device | Special relay | | Bit | SM (SM**) | 91H (9100H) | Specify in the range of device numbers of the module to access. | Decimal | ○ |
| | Special register | | Word | SD (SD**) | A9H (A900H) | | Decimal | ○ |
| | Function input | | Bit | — | — | — | Hexadecimal | — |
| | Function output | | | — | — | | Hexadecimal | — |
| | Function register | | Word | — | — | Decimal | — | |
| Index register | | | 16 bits | Z* (Z***) | CCH (CC00H) | Specify in the range of device numbers of the module to access. | Decimal | ○ |
| | | | 32 bits | LZ (LZ***) | 62H (6200H) | | Decimal | ○ |
| File register | | | Word | R* (R***) | AFH (AF00H) | Specify in the range of device numbers of the module to access. | Decimal | ○ |
| | | | | ZR (ZR**) | B0H (B000H) | | Decimal | — |
| Link direct device*4 | Link input | | Bit | X* (X***) | 9CH (9C00H) | Specify in the range of device numbers of the module to access. | Hexadecimal | — |
| | Link output | | | Y* (Y***) | 9DH (9D00H) | | Hexadecimal | — |
| | Link relay | | | B* (B***) | A0H (A000H) | | Hexadecimal | — |
| | Link special relay | | | SB (SB**) | A1H (A100H) | | Hexadecimal | — |
| | Link register | | Word | W* (W***) | B4H (B400H) | | Hexadecimal | — |
| | Link special register | | | SW (SW**) | B5H (B500H) | | Hexadecimal | — |
| Module access device*4 | Link register | | Word | W* (W***) | B4H (B400H) | Specify in the range of device numbers of the module to access. | Hexadecimal | — |
| | Link special register | | | SW (SW**) | B5H (B500H) | | Hexadecimal | — |
| | Module access device | | | G* (G***) | ABH (AB00H) | | Decimal | ○ |
| Other devices | SFC block device | | Bit | BL (BL**) | DCH (DC00H) | — | Decimal | × |

- *1 [ASCII code]
If the device code is less than the specified character number, add "*" (ASCII code: 2AH) or a space (ASCII code: 20H) after the device code.
[Binary code]
When "Device code" is less than the size specified add "00H" to the end of the device code.
- *2 ○: SLMP-compatible device
—: FX5-incompatible device
×: SLMP-incompatible device
- *3 Depends on the communication data code. See below.
ASCII code (X,Y OCT): Octal
ASCII code (X,Y HEX), Binary code: Hexadecimal
- *4 "Device memory extension specification" for sub-commands must be turned ON (1).

1E frames

With 1E frames, specify the device to access with the "Device code" listed below.

| Classification | Device | Type | Symbol | Device code*1 | | Device No. | | Access*2 | |
|-----------------------|------------------|---------------|--------|---------------|-------------|------------|-------------|----------|---|
| | | | | ASCII code | Binary code | | | | |
| Internal user device | Input | Bit | X | X* | 5820H | 0 to 377 | *3 | ○ | |
| | Output | | Y | Y* | 5920H | 0 to 377 | *3 | ○ | |
| | Internal relay | | M | M* | 4D20H | 0 to 7679 | Decimal | ○ | |
| | Latching relay | | L | L* | 4C20H | — | Decimal | — | |
| | Annunciator | | F | F* | 4620H | — | Decimal | — | |
| | Link relay | | B | B* | 4220H | — | Hexadecimal | — | |
| | Step relay | | S | S* | 5320H | 0 to 4095 | Decimal | ○ | |
| | Data register | Word | D | D* | 4420H | 0 to 7999 | Decimal | ○ | |
| | Link register | | W | W* | 5720H | — | Hexadecimal | — | |
| | Timer | Contact | Bit | TS | TS | 5453H | 0 to 511 | Decimal | ○ |
| | | Coil | Bit | TC | TC | 5443H | — | | — |
| | | Current value | Word | TN | TN | 544EH | 0 to 511 | | ○ |
| | Retentive timer | Contact | Bit | STS | SS | 5353H | — | Decimal | — |
| | | Coil | Bit | STC | SC | 5343H | — | | — |
| | | Current value | Word | STN | SN | 534EH | — | | — |
| | Counter | Contact | Bit | CS | CS | 4353H | 0 to 255 | Decimal | ○ |
| | | Coil | Bit | CC | CC | 4343H | — | | — |
| | | Current value | Word | CN | CN | 434EH | 0 to 255 | | ○ |
| | Long counter | Contact | Bit | LCS | LS | 4C53H | *4 | Decimal | △ |
| Coil | | Bit | LCC | LC | 4C43H | — | — | | |
| Current value | | Double Word | LCN | LN | 4C4EH | *5 | △ | | |
| Link special relay | | Bit | SB | SB | 5342H | — | Hexadecimal | — | |
| Link special register | | Word | SW | SW | 5357H | — | Hexadecimal | — | |
| System device | Special relay | Bit | SM | SM | 534DH | *6 | Decimal | △ | |
| | Special register | Word | SD | SD | 5344H | *7 | Decimal | △ | |
| Index register | | Word | Z | Z* | 5A20H | — | Decimal | — | |
| Long index register | | Double Word | LZ | LZ | 4C5AH | — | Decimal | — | |
| File register | | Word | R | R* | 5220H | 0 to 32767 | Decimal | ○ | |
| Link direct device | Link input | Bit | DX | DX | 4458H | — | Hexadecimal | — | |
| | Link output | | DY | DY | 4459H | — | Hexadecimal | — | |
| Module access device | | Word | G | — | — | — | Decimal | — | |

*1 [ASCII code]

If the device code is less than the specified character number, add a space (ASCII code: 20H) after the device code.

[Binary code]

When "Device code" is less than the size specified add "20H" to the end of the device code.

*2 ○: Accessible to the FX5 CPU module devices

—: Inaccessible to the FX5 CPU module devices

△: Accessible to a specific device of FX5 CPU module (cannot be directly specified)

*3 Depends on the communication data code. See below.

ASCII code (X, Y OCT): octal

ASCII code (X, Y HEX), binary code: hexadecimal

*4 Access LCS0 to LCS55 by specifying CS200 to CS255.

*5 Access LCN0 to LCN55 by specifying CN200 to CN255.

*6 Access SM8000 to SM8511 by specifying M8000 to M8511.

*7 Access SD8000 to SD8511 by specifying D8000 to D8511.

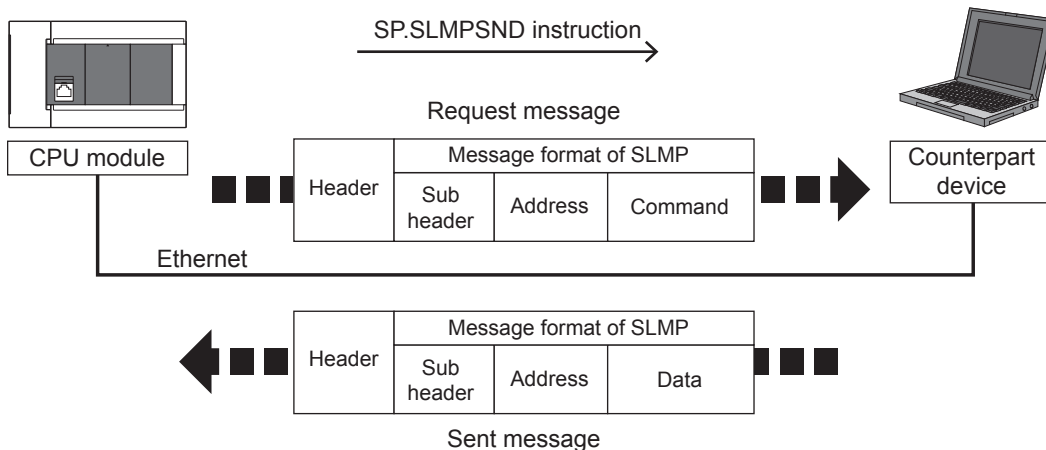
5.4 Data Communication Process

Sending the SLMP frame

When sending a request message from the CPU module to the external device, use the following dedicated instructions. Supported only for 3E frame.

| Instruction symbol | Description |
|--------------------|--|
| SP.SLMPSND | Sends SLMP messages to the SLMP-compatible device. |

Specify the external device and SLMP command to SP.SLMPSND instruction, execute SP.SLMPSND instruction in the program, and the request message is sent from the CPU module to the external device. The response message from the external device is stored to the device specified by SP.SLMPSND instruction.



For details on SP.SLMPSND instruction, refer to [MELSEC iQ-F FX5 Programming Manual \(Instructions, Standard Functions/Function Blocks\)](#)

5.5 Precautions

Checking communication status based on LED display

Check the status of the "SD/RD" LED display at the Ethernet port of the Ethernet-equipped module.

| "SD/RD" LED indicator status | Operation status |
|------------------------------|--------------------------------------|
| Flashing | Data is being sent or received. |
| Off | Data is not being sent nor received. |

The LED flashes brightly when performing SLMP (3E frame) communication normally.

If the LED is not flashing, check the wiring and the communication settings.

Checking communication status based on error code

For the error codes stored in the end code when there is an abnormal end of SLMP (3E frame) communication, refer to

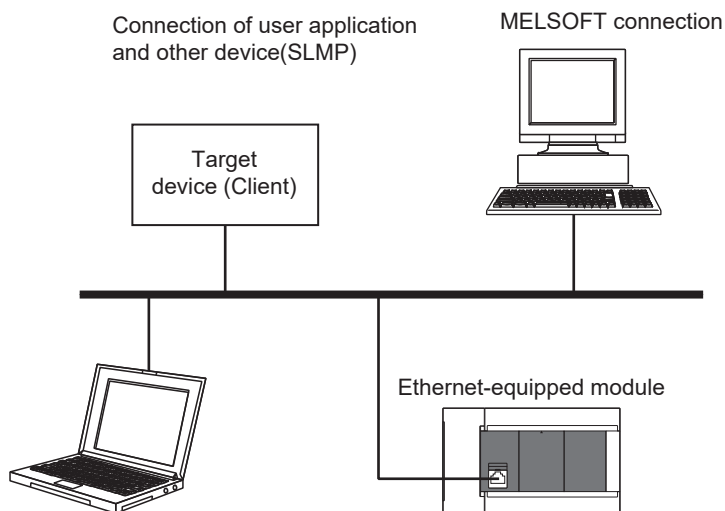
[Page 255 SLMP function error code.](#)

Number of connectable units

One Ethernet-equipped module can be accessed by the following number of external devices at the same time (including socket communication, MELSOFT connections^{*1}, and SLMP).

- CPU module: Up to 8 units
- Ethernet module: Up to 32 units

*1 The first device for MELSOFT connection is not included.



Maintenance
GX Works3, etc. (MELSOFT connection)

For connections with external devices by SLMP, the number of possible simultaneous connections is the number of devices configured in the Ethernet configuration settings only.

Port number

Host station port number, 1 to 1023 (0001H to 03FFH), are assigned for reserved port numbers (WELL KNOWN PORT NUMBERS) and 61440 to 65534 (F000H to FFEH) are for other communication functions. Therefore, using 1024 to 5548, 5570 to 61439 (0400H to 15ACH, 15C2H to EFFFH) is recommended.

Data communication frames

The frames that can be used on the Ethernet-equipped module are the same as 3E/1E frames.

Access range

- Only the connected Ethernet-equipped module can be accessed. Transmissions to other modules will result in an error.
- Communication with other stations such as CC-Link via the connected Ethernet-equipped module cannot be done.

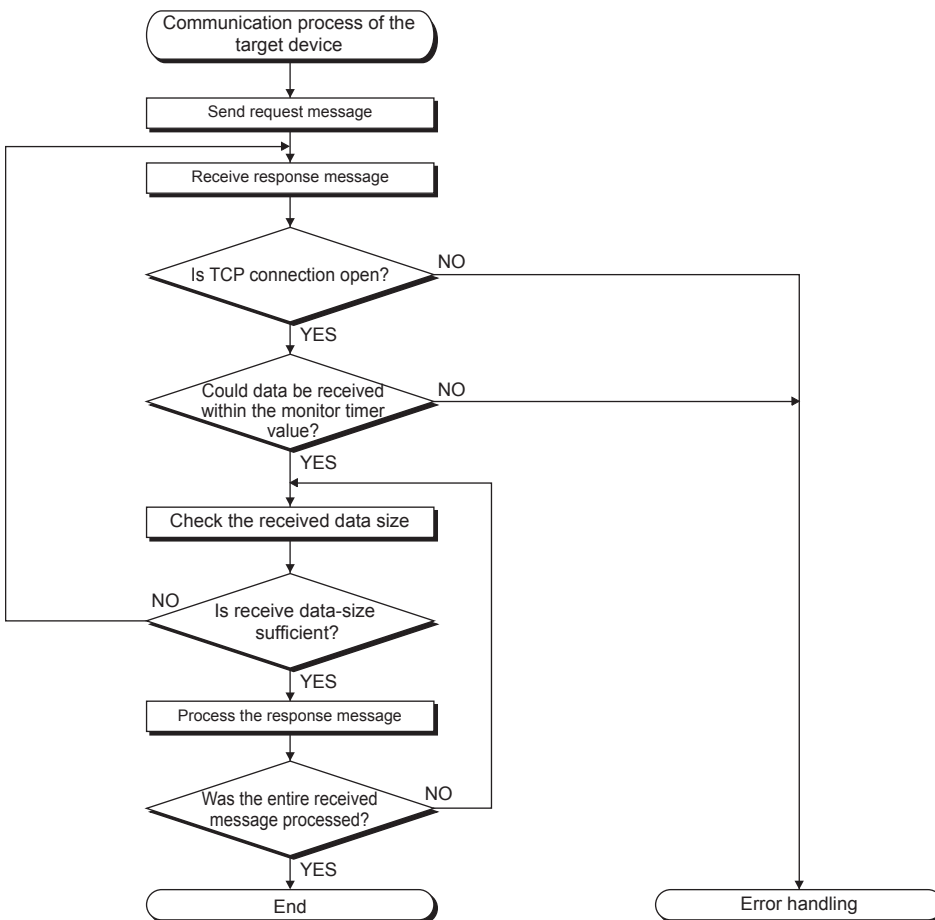
For details of the access range, refer to the MELSEC iQ-F FX5 User's Manual (SLMP).

Precautions when the protocol is set to UDP

- For a single UDP port, if a new request message is sent while waiting for the response message after sending the first request message, the new request message will be discarded.
- When the same local port number has been set multiple times in UDP, the result is the same as if only one has been set. If you want to communicate with multiple external devices using the same local port number, use TCP.

Data reception processing for response messages

The following shows an example of the data reception processing of an external device.



5

Point

For Ethernet communication, TCP socket functions are used inside personal computers.


These functions have no concept of boundaries. When the sender sends data by calling the send function once, the receiver will call the recv function once or more to read that data. (Send and recv do not have a one-to-one correspondence.)

Therefore, the processing shown above is always required in the program of the receiving device.

When the recv function is used with the blocking mode, data may be read by calling the function once.

6 PREDEFINED PROTOCOL SUPPORT FUNCTION

This chapter describes predefined protocol support function (built-in Ethernet).

For details on the predefined protocol support function (serial communication), refer to  MELSEC iQ-F FX5 User's Manual (Serial Communication).

Outline

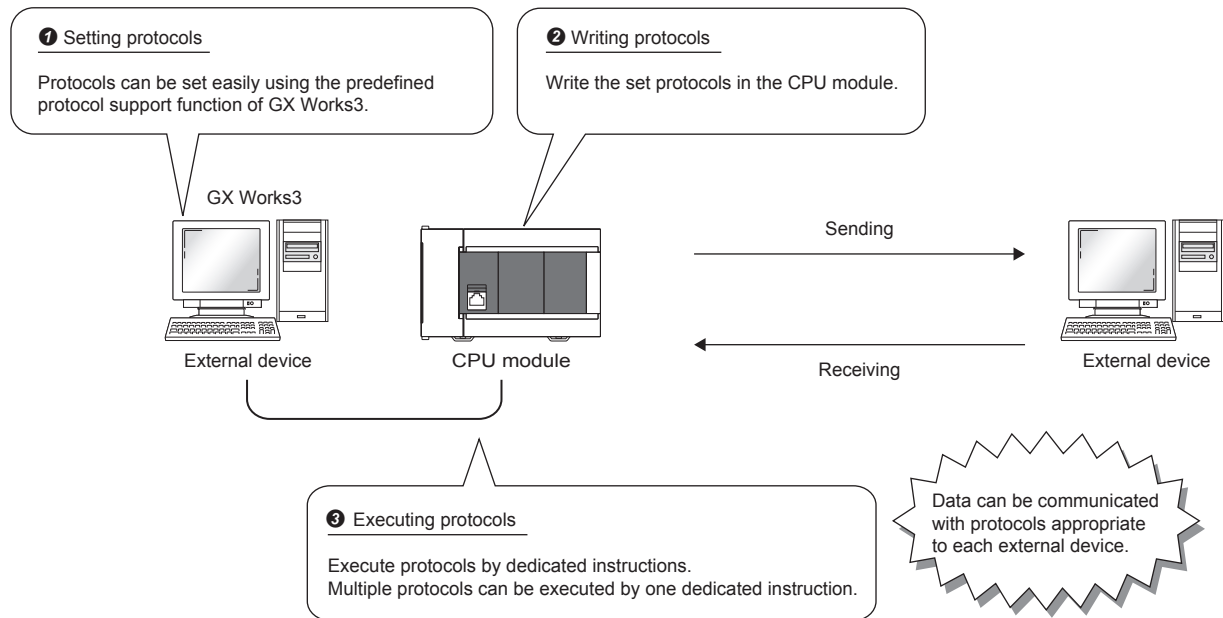
Data can be exchanged between the external device (such as measuring instrument and bar code reader) and the CPU module following the protocol of the device.

Data that varies according to communication session can be handled by incorporating a device or buffer memory into the communication packet.

Sets the protocol required for communication with the external device using the engineering tool.

The protocol can be set by selecting from the predefined protocol library (SLMP, MODBUS/TCP*¹, etc.), or it can be created and edited.

*1 The SLMP and MODBUS/TCP are available only in the client.



Point

The number of protocols and packets that can be registered is as follow.

- Protocols: 64 maximum
- Packets: 128 maximum
- Packet data area size: 6144 bytes maximum

When the number of packets reaches the upper limit, protocols cannot be added even if the number of protocols has not reached the upper limit.

If the packet data area size reaches the upper limit, protocols and packets cannot be added even if the number of protocols and packets has not reached the upper limit.

Applicable connections

The connections Nos. 1 to 8 can be used for communications using the communication protocol support function.

6.1 Data Communication

When the predefined protocol support function is used, data can be exchanged with the external device using the following procedure.

1. Select, create or edit the protocol with the predefined protocol support function, and write the protocol setting data. (☞ Page 65 Creating the protocol setting data)
2. Set the module parameter. (☞ Page 70 Module parameter setting procedure)
3. Write the parameters to the CPU module.
4. Perform the open processing to establish a connection between the CPU module and external device.
5. Execute the protocol with the dedicated instruction (SP.ECPRTCL instruction).
6. Close the connection when communication is finished.

Point

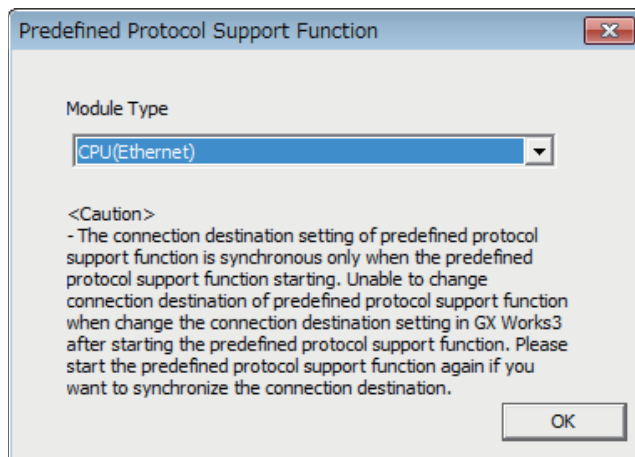
The communication data code is binary code communication regardless of the selected settings.

Creating the protocol setting data

Use the predefined protocol support function to create the protocol setting data.

[Tool] ⇒ [Predefined Protocol Support Function]

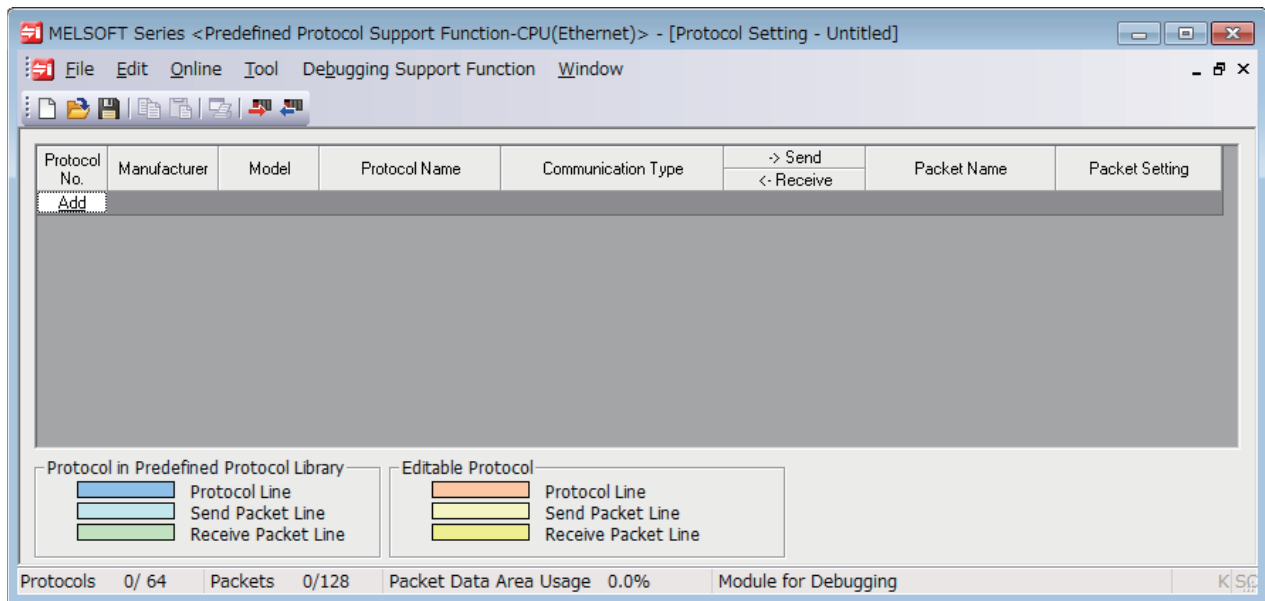
Select the module for which to create the protocol setting data.



■ Newly creating the protocol setting data

Newly create the protocol setting data.


[File] ⇒ [New] ⇒ "Protocol Setting" screen

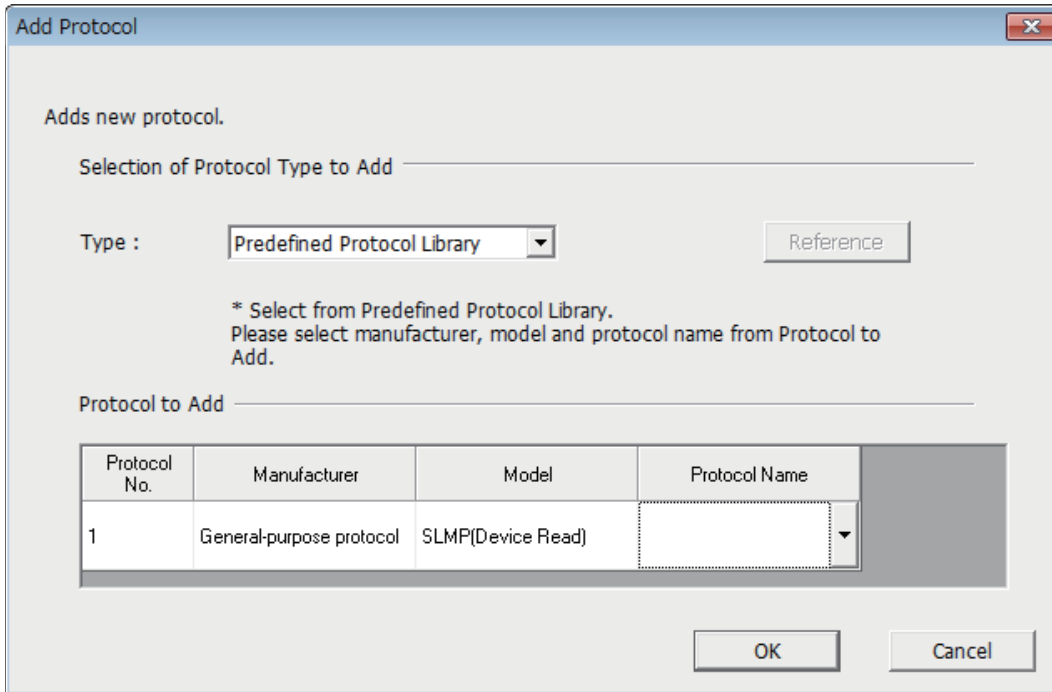


| Item | Description |
|--------------------|--|
| Protocol No. | Displays the protocol number used with the dedicated instruction. |
| Manufacturer | Displays the name of the manufacturer of the device for which the protocol is being set. |
| Model | Displays the model of the protocol to be set. |
| Protocol Name | Displays the name of the protocol to be set. |
| Communication Type | Displays the communication type of the protocol to be set. Send only: Sends one send packet once. Receive only: If there is a matching packet within up to 16 registered and received packets, it is received. Send & receive: After sending one send packet, if there is a matching packet within up to 16 registered and received packets, it is received. |
| ->Send/<-Receive | Displays the packet send direction. ->: For send <-(1) to (16): For receive, the received packet number is displayed in parentheses. |
| Packet Name | Displays the packet name. |
| Packet Setting | Displays the validity of variables in the packet elements and the variable setting state. If Variable Unset, Elements Unset, or Element Error, the protocol is not written to the CPU module. No Variable: When there is no variable in the elements Variable Set: Only when all variables have been set Variable Unset: When there is an unset variable Elements Unset: When there are no elements in an editable protocol Element Error: When elements do not satisfy requirements |

■ Adding protocol

Add protocol.

 [Edit] ⇄ [Add Protocol]



Adds new protocol.

Selection of Protocol Type to Add _____

Type :

* 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 | SLMP(Device Read) | <input type="text"/> |

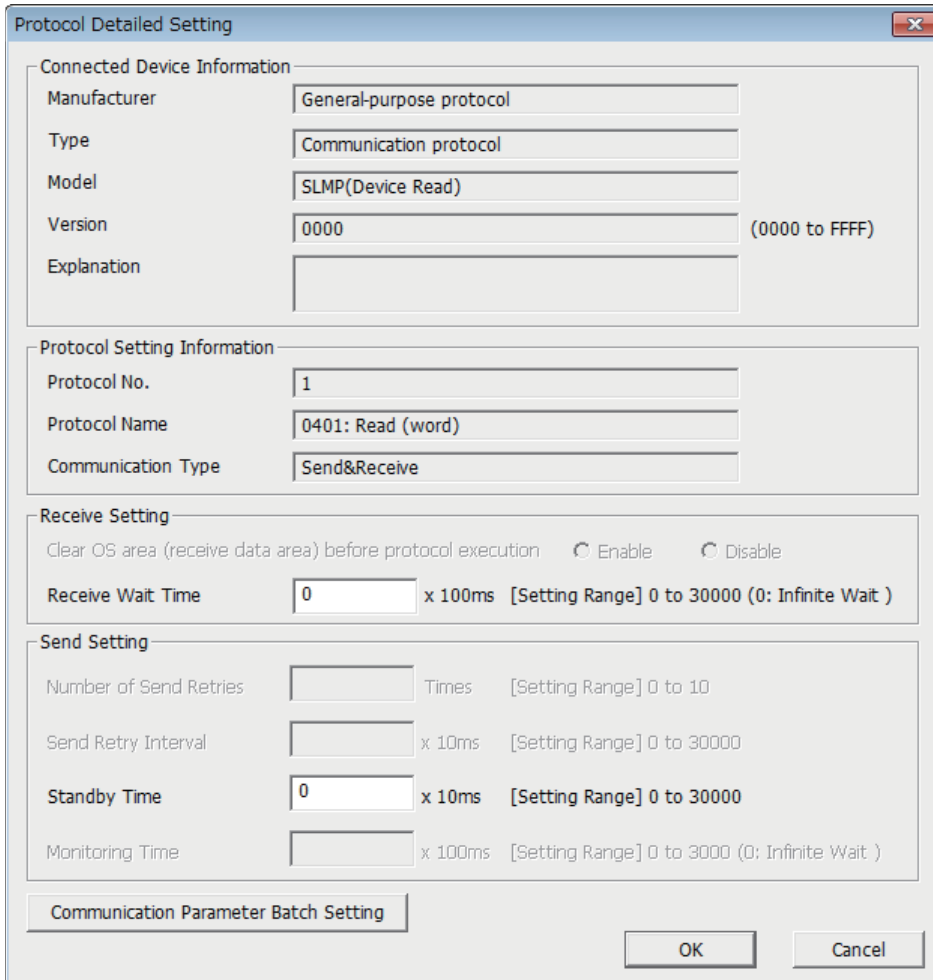
| Item | Description | Setting range |
|-----------------------------|--|---|
| Type | Select the type of protocol to be added. | <ul style="list-style-type: none"> • Predefined Protocol Library • User Protocol Library • Add New |
| Protocol No. | Select the protocol number to be added. | 1 to 64 |
| Manufacturer ^{*1} | Set the maker of the protocol to be added. | — |
| Model ^{*1} | Set the type of protocol to be added. | — |
| Protocol Name ^{*1} | Set the name of the protocol to be added. | — |

*1 The name can be set only when "Predefined Protocol Library" is selected for "Type"

■ Protocol Detailed Setting

Set the protocol send/receive parameters.

 "Protocol Setting" window ⇒ Select a protocol ⇒ [Edit] ⇒ [Protocol Detailed Setting]



Protocol Detailed Setting

Connected Device Information

Manufacturer: General-purpose protocol

Type: Communication protocol

Model: SLMP(Device Read)

Version: 0000 (0000 to FFFF)

Explanation:

Protocol Setting Information

Protocol No.: 1

Protocol Name: 0401: Read (word)

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 Send Retries: Times [Setting Range] 0 to 10

Send Retry Interval: x 10ms [Setting Range] 0 to 30000

Standby Time: 0 x 10ms [Setting Range] 0 to 30000

Monitoring Time: x 100ms [Setting Range] 0 to 3000 (0: Infinite Wait)

Communication Parameter Batch Setting

OK Cancel

| Item | Description | |
|--|--------------------|--|
| Connected Device Information ^{*1} | Manufacturer | Set the protocol maker name. |
| | Type | Set the protocol device type. |
| | Model | Set the protocol model. |
| | Version | Set the protocol device version. |
| | Explanation | Set a description of the protocol device. |
| Protocol Setting Information ^{*1} | Protocol No. | The protocol number for the selected protocol is displayed. |
| | Protocol Name | Set the protocol name. |
| | Communication Type | Type Set the protocol communication type. |
| Receive Setting | Receive Wait Time | Set the time for wait after the module enters the receive data wait state. If communication with the external device is disabled because of a disconnection and matching packet data is not received within the specified time, the module judges that an error has occurred and cancels the receive data wait state. |
| Send Setting | Standby Time | Set the time to wait from when the protocol set for the module enters the execution state to when the data is actually sent. The time for the external device to enter the receive enable state can be adjusted with this in respect to the module's send timing. |

*1 The setting cannot be changed if the protocol was selected from the predefined protocol library.

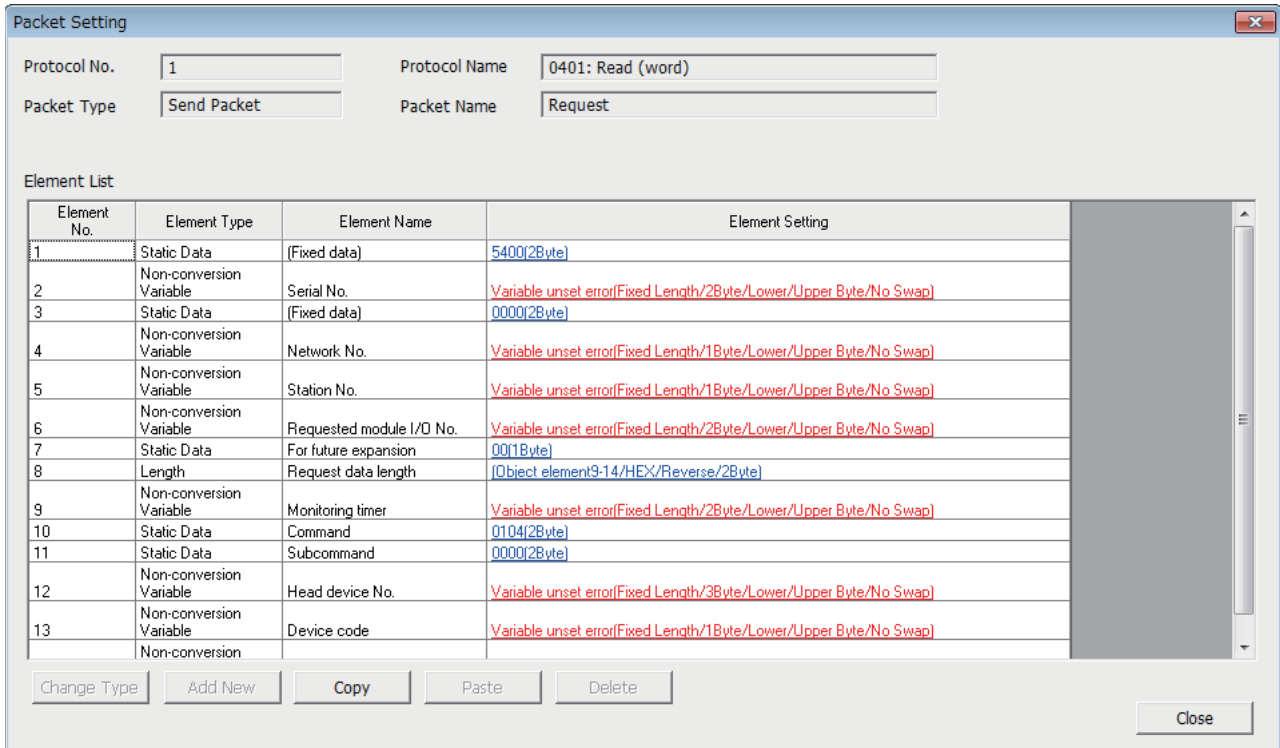
Point

Send/receive parameters can be set for multiple protocols by clicking the [Communication Parameter Batch Setting] button and setting the range of the set protocol numbers, receive settings, and send settings.

■ Packet setting

Set the configuration of the send/receive packets on the "Packet Setting" window.

 "Protocol Setting" window ⇒ Packet to be set



| Element No. | Element Type | Element Name | Element Setting |
|-------------|-------------------------|--------------------------|---|
| 1 | Static Data | (Fixed data) | 5400(2Byte) |
| 2 | Non-conversion Variable | Serial No. | Variable unset error(Fixed Length/2Byte/Lower/Upper Byte/No Swap) |
| 3 | Static Data | (Fixed data) | 0000(2Byte) |
| 4 | Non-conversion Variable | Network No. | Variable unset error(Fixed Length/1Byte/Lower/Upper Byte/No Swap) |
| 5 | Non-conversion Variable | Station No. | Variable unset error(Fixed Length/1Byte/Lower/Upper Byte/No Swap) |
| 6 | Non-conversion Variable | Requested module I/O No. | Variable unset error(Fixed Length/2Byte/Lower/Upper Byte/No Swap) |
| 7 | Static Data | For future expansion | 00(1Byte) |
| 8 | Length | Request data length | (Object element9-14/HEX/Reverse/2Byte) |
| 9 | Non-conversion Variable | Monitoring timer | Variable unset error(Fixed Length/2Byte/Lower/Upper Byte/No Swap) |
| 10 | Static Data | Command | 0104(2Byte) |
| 11 | Static Data | Subcommand | 0000(2Byte) |
| 12 | Non-conversion Variable | Head device No. | Variable unset error(Fixed Length/3Byte/Lower/Upper Byte/No Swap) |
| 13 | Non-conversion Variable | Device code | Variable unset error(Fixed Length/1Byte/Lower/Upper Byte/No Swap) |

The above window opens when "Predefined Protocol Library" is selected on the "Add Protocol" window.


When "Add New" or "User Protocol Library" has been selected, configure the packets with the [Change Type] button and [Add New] button.

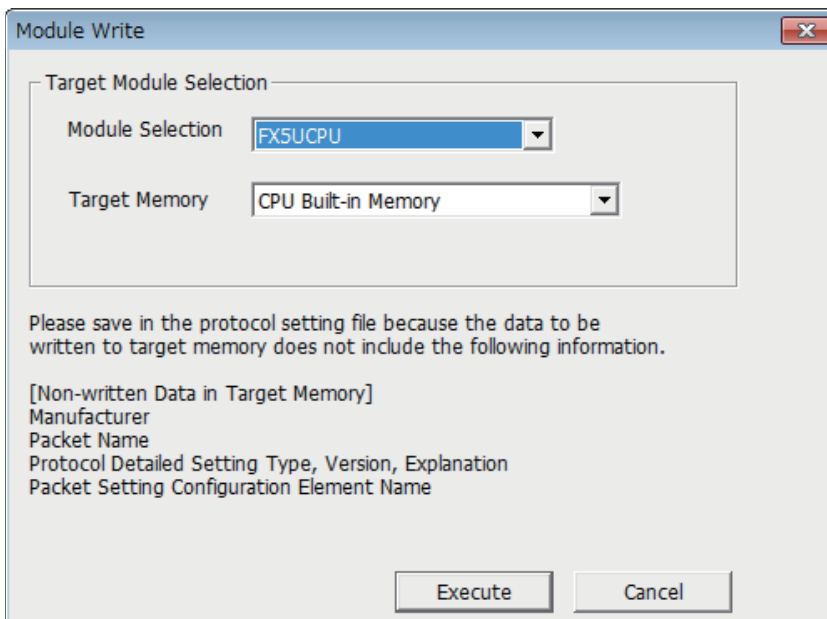
For details on the packet elements, refer to the following.

 Page 71 Packet Elements

■ Writing the protocol setting data

Write the protocol setting data to the CPU module.

 [Online] ⇒ [Write to Module]



Target Module Selection

Module Selection:

Target Memory:

Please save in the protocol setting file because the data to be written to target memory does not include the following information.

[Non-written Data in Target Memory]
 Manufacturer
 Packet Name
 Protocol Detailed Setting Type, Version, Explanation
 Packet Setting Configuration Element Name

Execute Cancel

Select the module and memory into which the protocol data is to be written, and execute write.

The protocol setting data is written into the module extension parameters.

The following data is not written as the protocol setting data so it will not be displayed even when read. However, when the protocol is selected from the predefined protocol library, the following can be displayed.

- Manufacturer
- Packet name
- Type, version, and explanation in the protocol detailed setting
- Element name in packet settings

When the predefined protocol settings are written into multiple target memories, the following operation will take place. The predefined protocol settings written in the SD memory card can be transferred to the CPU built-in memory by using boot operation.

For details on boot operation, refer to the following.

 MELSEC iQ-F FX5 User's Manual (Application)

Module parameter setting procedure

Set "External Device Configuration" under "Basic Settings".

 Page 77 Parameter settings

1. Select the external device to be connected in "Module List" and drag it to "List of devices" or "Device map area".

| External device name | Description |
|-------------------------------|--|
| UDP Connection Module | Select to communicate with the external device using UDP/IP |
| Active Connection Module | Select to perform the open processing to the external device from the CPU module (Active open) and communicate using TCP/IP. |
| Unpassive Connection Module | Select to receive the open processing from a unspecified external device (Unpassive open) and communicate using TCP/IP. |
| Fullpassive Connection Module | Select to receive the open processing from the specified external device (Fullpassive open) and communicate using TCP/IP. |

2. Set "Communication Procedure" for the external device to "Predefined Protocol".

3. Set the other parameters required for communication in the connection.

Applicable dedicated instructions

The dedicated instruction "SP.ECPRTCL" is used in the communication protocol support function (built-in Ethernet).

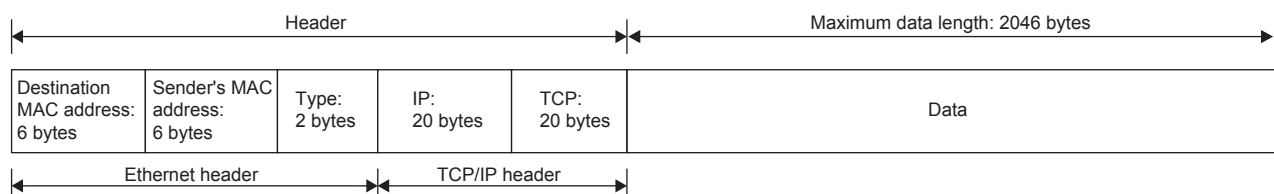
6.2 Protocol Communication Type

The packets sent to the external device when a processing is executed and the external device's receive packets are registered in the protocol.

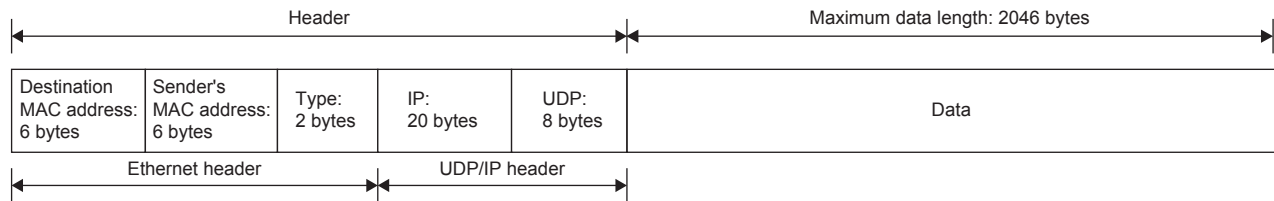
The packet elements set with the predefined protocol support function are the data section of the packets that are actually sent and received.

This section describes an example of the packet configuration.

For TCP/IP



For UDP/IP



With the predefined protocol support function, data is exchanged with the external device with the procedures (communication type) shown below.

| Communication type | Description |
|--------------------|--|
| Send Only | The send packet is sent once. |
| Receive Only | If there is a packet that matches within the maximum of 16 registered receive packets, the packet is received. |
| Send & Receive | After sending the send packets, if there are packets that match the up to 16 registered receive packets, the packets are received. |

6.3 Packet Elements

The packet is created with a combination of packet elements.

Up to 32 elements can be set in one packet. One packet can have a maximum data length of 2046 bytes.

This section describes the details of the packet elements.

Static data

Use when there are specific codes and character strings, such as commands, in the packet.

- When sending: The specified code and character string are sent.
- When receiving: The received data is verified.

Multiple static data elements can be placed anywhere in the data part.

The following table lists the items.

| Item | Description | Remarks |
|---------------|---|--|
| Element Name | Set the element name. | — |
| Code Type | Select a data type of the setting value. ASCII String/ASCII Control Code/HEX | — |
| Setting Value | Set data within 1 to 50 bytes. Code type and setting range are as follows: <ul style="list-style-type: none"> • ASCII String: 20H to 7EH • ASCII Control Code: Control code of 00H to 1FH and 7FH • HEX: Hexadecimal data of 00H to FFH | Setting example ASCII String: "ABC" ASCII Control Code: STX HEX: FFFF |

Length

The length code is used when there is an element that indicates the data length in the packet.

- When sending: Automatically calculates the data length in the specified range, and adds it to the packet.
- When receiving: From the received data, the data (value) corresponding to the length is verified as the specified range's data length.

Length elements can be placed anywhere in the data part.

Multiple length elements can be set placed in one packet.

The following table lists the items.

| Item | Description | Remarks | |
|-------------------|---|--|--|
| Element Name | Set the element name. | — | |
| Code Type | Select the data length type. ASCII hexadecimal/HEX | — | |
| Data Length | Select the data length on the line. The range is 1 to 4 bytes. | — | |
| Data Flow | Forward Direction (Upper byte → Lower byte) | When sending: Sends the calculated length in order from the upper byte. When receiving: Receives the data in order from the upper byte. | This cannot be set if the data length is 1 byte. |
| | Reverse Direction (Lower byte → Upper byte) | When sending: Sends the calculated length in order from the low-order byte. When receiving: Receives the data in order from the low-order byte. | |
| | Byte Swap (by Word) ^{*1} | When sending: Interchanges the bytes in word units and sends the calculated length. When receiving: Interchanges the bytes in word units and receives the data. | |
| Calculating Range | Start | Select the start packet element number for the range to be calculated. The range is 1 to 32. | — |
| | End | Select the end packet element number for the range to be calculated. The range is 1 to 32. | |

*1 This can be selected only when the data length is set to 4 bytes.

Point

- If there are no elements other than length, an element error occurs. (When using length, one or more elements other than length are required.)
- If the calculation result exceeds the number of digits set with "Data Length", the excessive digit value is discarded (invalidated). For example, if Data Length is 2 bytes and the data size calculation results are "123" bytes, the data length will be "23".
- If there is a non-conversion variable (variable length)/non-verified reception (character length variable) after the length, and that section is not included in the length calculating range, arrange the static data immediately after the non-conversion variable/non-verified reception.
- When the code type setting is "ASCII Hexadecimal", a mismatch will occur if a character string other than "0" to "9", "A" to "F", and "a" to "f" is received.
- Use "0" to "9" or "A" to "F" when converting to ASCII characters during send.
- When arranging multiple length elements, none of the length calculating range may overlap.
- When arranging multiple length elements, the previous length calculating range may not exceed the arranged length.
- A length element cannot be arranged at the final position of the packet elements.

Non-conversion variable

Use this to send the CPU module device data as part of the send packet, or to store part of the received packet in the CPU module device.

Multiple non-conversion variable can be arranged in one packet.

The following table lists the items.

| Item | Description | |
|--|--|--|
| Element Name | Set the element name. | |
| Fixed Length/ Variable Length | Fixed Length | The data whose length is fixed is sent and received. |
| | Variable Length | When sending: The data length is specified at the time of the protocol execution and the data is sent. When receiving: The data whose length is variable is received. |
| Data Length/ Maximum Data Length | Set the data length of the send/receive data. (For a variable length, set the maximum data length that can be specified for the data length storage area.) The range is 1 to 2046. | |
| Unit of Stored Data | Lower byte + Upper byte | When sending: Each one word (2 bytes) data in the data storage area is sent in the order of the lower byte to the upper byte. When receiving: The receive data is stored to the data storage area in the order of the lower byte to the upper byte. |
| | Lower Bytes Only | When sending: Each lower byte data in the data storage area is sent. The CPU module ignores the upper byte data. When receiving: The receive data is stored to each lower byte in the data storage area. The CPU module stores 00H in the upper byte. |
| Byte Swap | Disable (Lower → Upper)/ Enable (Upper → Lower) | When sending: When "Enable (Upper → Lower)" is selected, data in the upper byte and lower byte are swapped by one word (2 bytes) and sent. When "Unit of Stored Data" is "Lower Byte + Upper Byte" and "Data Length" is an odd number of bytes, the upper byte is sent at transmission of the last byte. When "Unit of Stored Data" is "Lower Bytes Only" and "Data Length" is an odd number of bytes, data without any byte swap is sent at transmission of the last byte. When receiving: When "Enable (Upper → Lower)" is selected, data in the upper byte and lower byte are swapped by word and sent. When "Unit of Stored Data" is "Lower Byte + Upper Byte" and "Data Length" is an odd number of bytes, the last byte is stored to the upper byte. When "Unit of Stored Data" is "Lower Bytes Only" and "Data Length" is an odd number of bytes, the last byte is stored without any byte swap. |
| Data Storage Area Specification | Specify the start device for storing the variable value. The settable devices are listed below. User device ^{*1,2} <ul style="list-style-type: none"> • Input (X) • Output (Y) • Internal relay (M) • Latch relay (L) • Link relay (B) • Data register (D) • Link register (W) File register ^{*2} <ul style="list-style-type: none"> • File register (R) | |

*1 Do not set local devices.

*2 Set within the device range specified with "Device/Label Memory Area Setting" in "Memory/Device Setting" under "CPU Parameters".

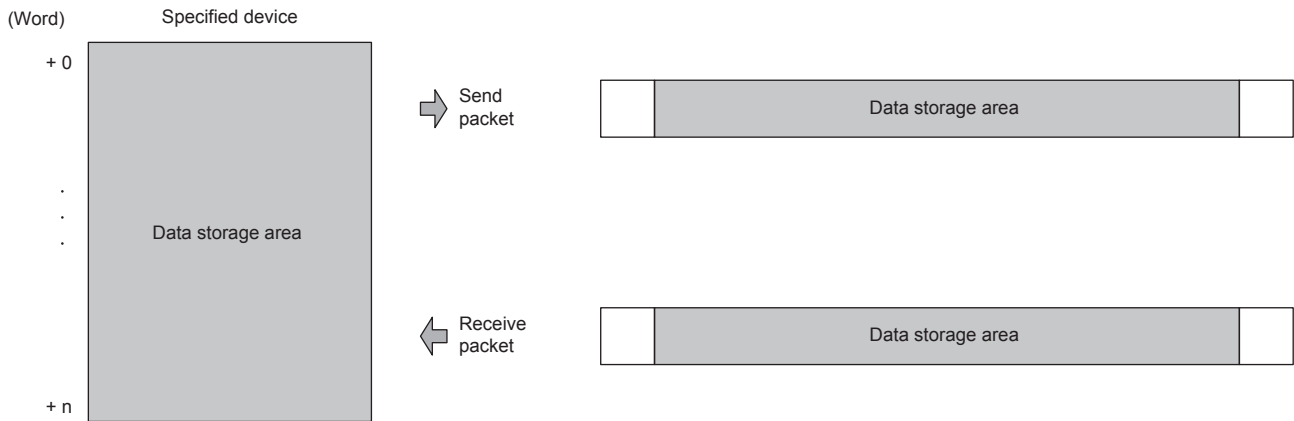
The following figures show the configuration of the data storage area.

■When "Fixed Length/Variable Length" is "Fixed Length"

The area after the device number specified on the "Element Setting" window becomes the data storage area.

The occupied data storage area differs according to the "Unit of Stored Data".

- When "Lower Byte + Upper Byte" is selected, the same size as the data length is occupied. (However, when the data length of a send packet is an odd number, the upper byte (lower byte for "Byte Swap") of the end device is not sent. When the data length of a receive packet is an odd number, the last data is stored with one byte of 00H.)
- When "Lower Bytes Only" is selected, a size double the data length is occupied.



For send packet: Send data is stored by the program

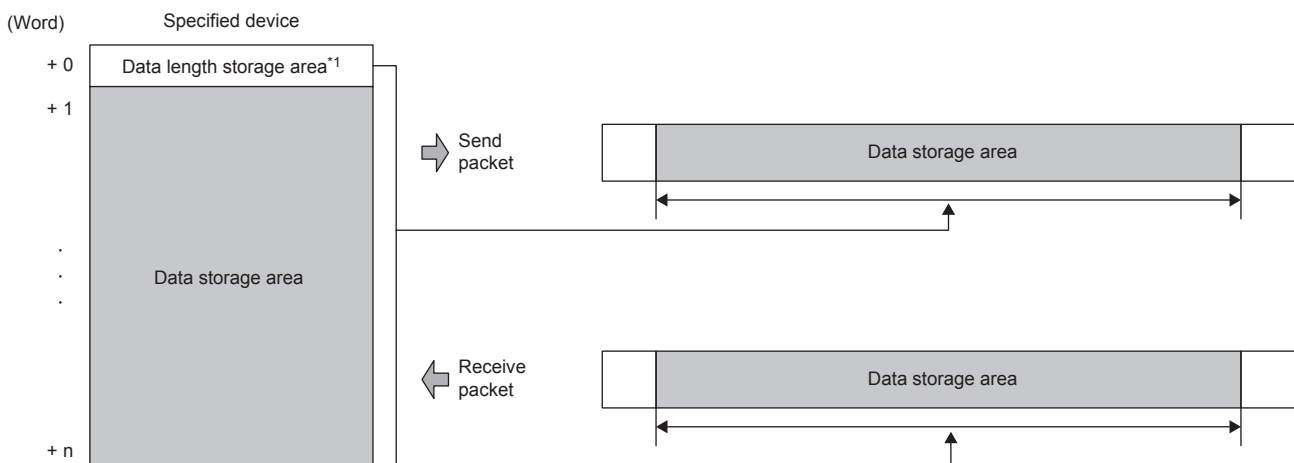
For receive packet: Receive data is stored by the CPU module

■When "Fixed Length/Variable Length" is "Variable Length"

The area after the device number specified on the "Element Setting" window + 1 becomes the data storage area.

The occupied data storage area differs according to the "Unit of Stored Data".

- When "Lower Byte + Upper Byte" is selected, the same size as the data length + one word (length for the data length storage area) are occupied. (However, when the data length of a send packet is an odd number, the upper byte (lower byte for "Byte Swap") of the end device is not sent. When the data length of a receive packet is an odd number, the last data is stored with one byte of 00H.)
- When "Lower Bytes Only" is selected, a size double the data length + one word (for data length storage area) is occupied.



For send packet: Send data is stored by the program

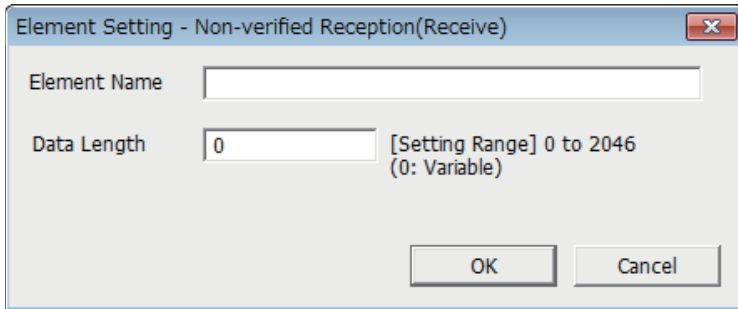
For receive packet: Receive data is stored by the CPU module

*1 The data length unit is byte fixed

When "Fixed Length/Variable Length" is "Variable Length" and the configuration is set as follows, an error occurs:

- An element other than static data is placed behind a non-conversion variable element when non-conversion variable is out of the length calculating range or when there is no length element (except for when nonconversion variable is placed at the end of the packet elements).
- Multiple non-conversion variable elements are placed in the length calculating range, while a length element is not placed.
- A non-conversion variable element is placed before a length element in the length calculating range.

Non-verified reception



Use this when receive data include unnecessary data.

If the receive packet contains non-verified reception, CPU module skims over the specified number of characters.

Multiple non-verified reception elements can be set in one packet.

The following table lists the items.

| Item | Description | |
|--------------|---|---|
| Element Name | Set the element name. | |
| Data Length | 0 (Number of characters variable) | Set when the number of characters that are not verified differs between each communication session. |
| | 1 to 2046 (number of character specification) | Set the number of characters that are not verified. |

When "Data Length" is set to 0, an error will occur if the following layout is used.

- An element other than static data is placed behind a non-verified reception element when non-verified reception is out of the length calculating range or when there is no length element (except for when non-verified reception is placed at the end of the packet elements).
- Multiple non-verified reception elements are placed in the length calculating range, while a length element is not placed.
- A non-verified reception element is placed before a length element in the length calculating range.

6.4 Execution Conditions of Predefined Protocol Communications

The predefined protocol communications can be executed when "Predefined protocol ready (SD10692)" is "1". This section describes the operation of 'Predefined protocol ready' (SD10692).

When the system is powered on or reset

If protocol setting data is written, the CPU module checks the protocol setting data when the system is powered on or reset. If the protocol setting data is normal, the CPU module sets "Predefined protocol ready (SD10692)" to "1" and enables execution of the protocol.

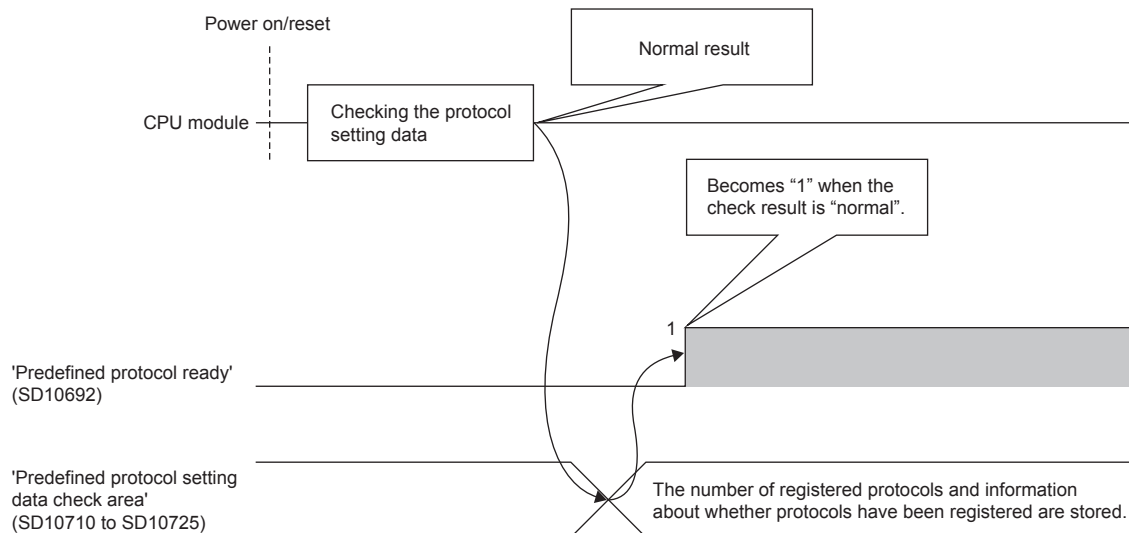
'Predefined protocol ready' (SD10692) is used as the interlock signal for executing the protocol.

If the protocol setting data is abnormal, "Predefined protocol ready (SD10692)" remains "0", and the details of the error are stored in SD10710 to SD10713 in the "Predefined protocol setting data check area".

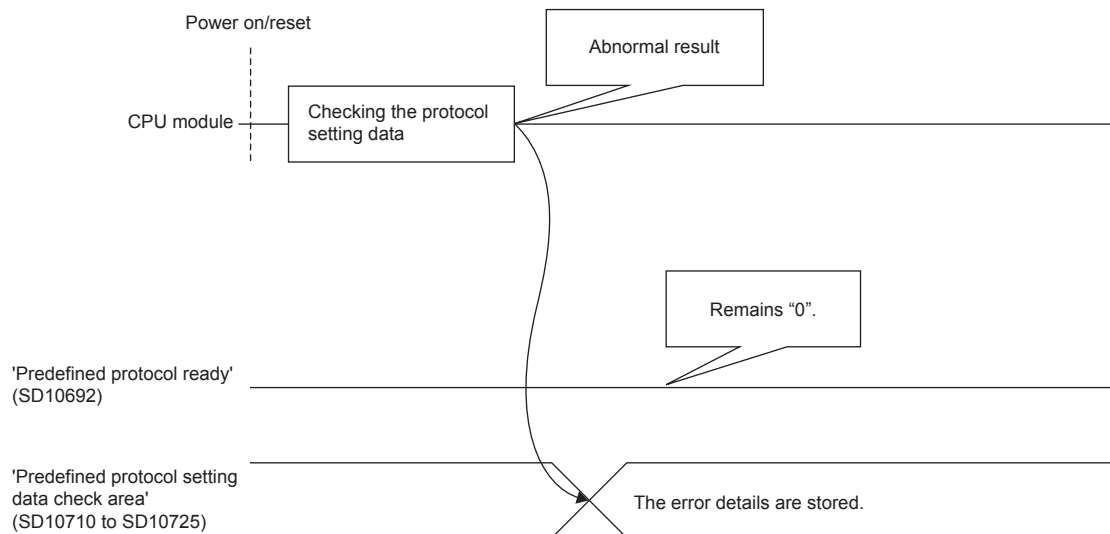
If protocol setting data is not written, the protocol setting data is not checked, and "Predefined protocol ready (SD10692)" remains "0".

Whether the protocol setting data is registered or not can be checked with 'Number of registered predefined protocols' (SD10714) and 'Predefined protocol registration' (SD10722 to SD10725).

■When protocol setting data is normal



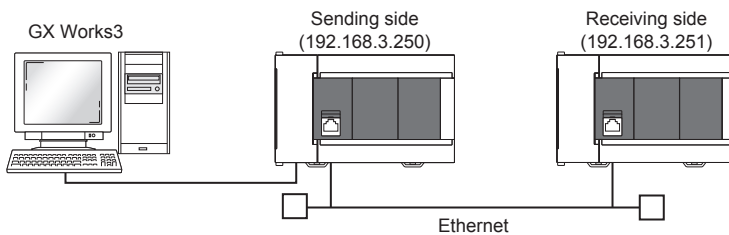
■When protocol setting data is abnormal



6.5 Example of Predefined Protocol Communications

This section describes an example of predefined protocol communications using UDP/IP.

System configuration



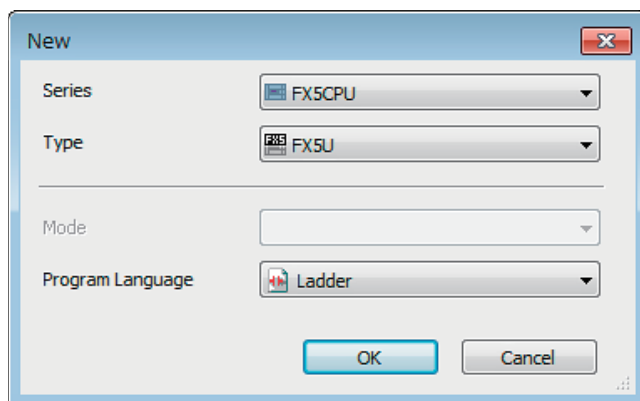
Parameter settings

Connect GX Works3 to the CPU module and set the parameters.

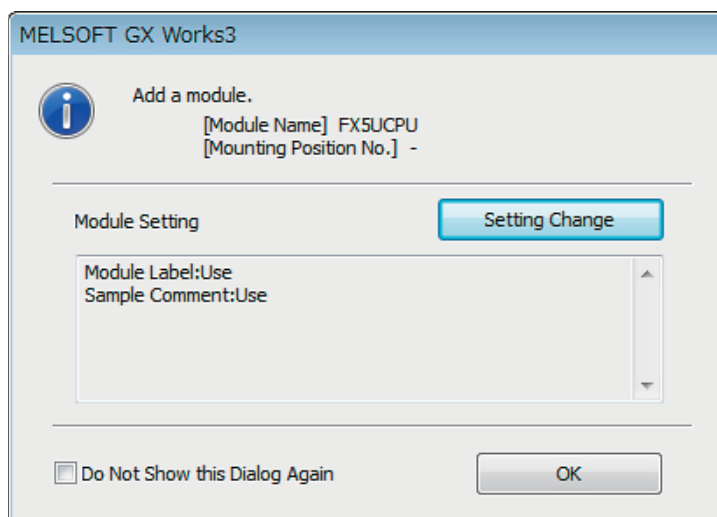
■ Sending side

1. Set the CPU module in the following.

[Project] ⇒ [New]

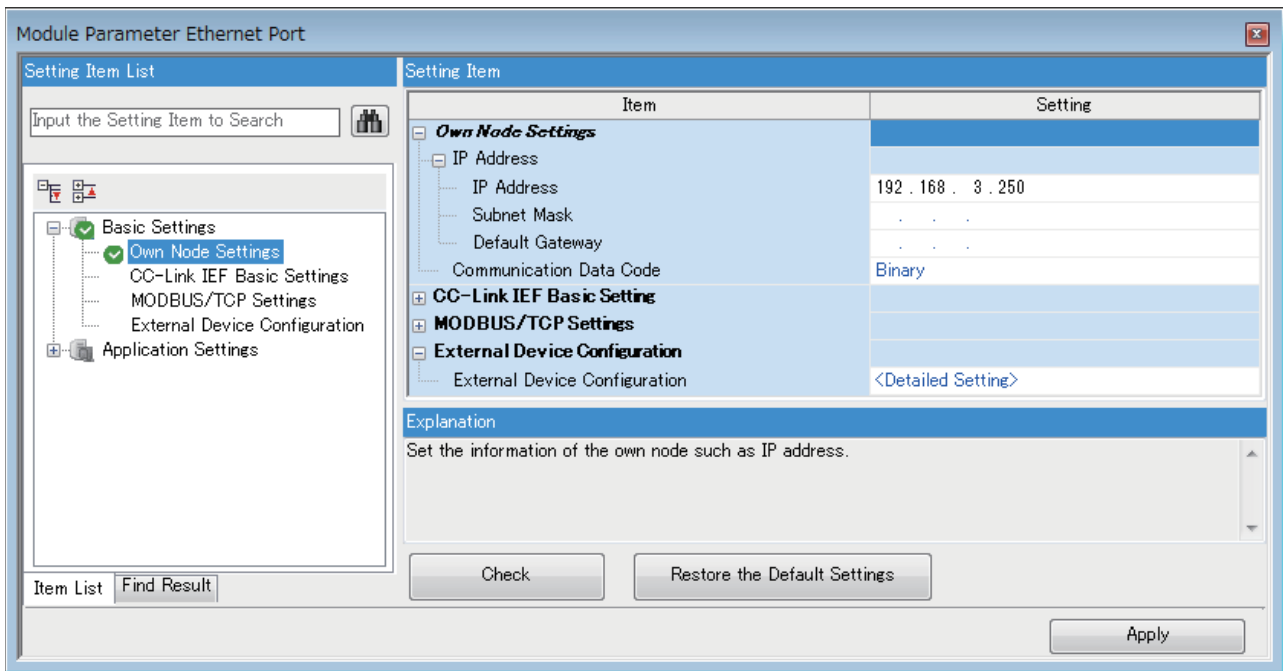


2. Set the module labels and click the [OK] button shown below.



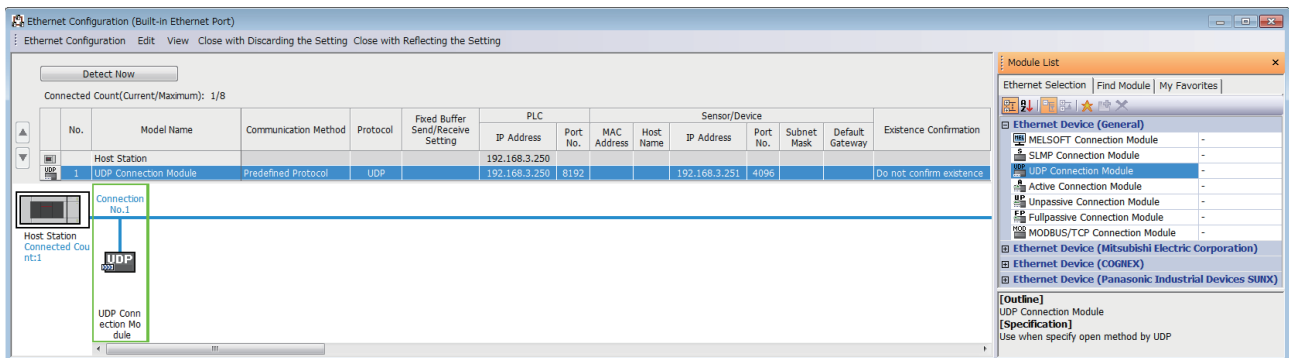
3. Set the "Basic Settings" in the following.

Navigation window⇒[Parameter]⇒[FX5UJCPU] or [FX5UCPU]⇒[Module Parameter]⇒[Ethernet Port]⇒[Basic Settings]



4. Set the external device configuration in the following.

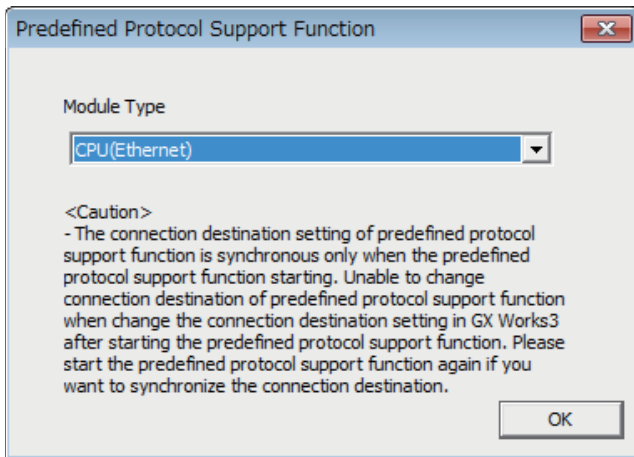
Navigation window⇒[Parameter]⇒[FX5UJCPU] or [FX5UCPU]⇒[Module Parameter]⇒[Ethernet Port]⇒[Basic Settings]⇒[External Device Configuration]



5. Start the predefined protocol support function.

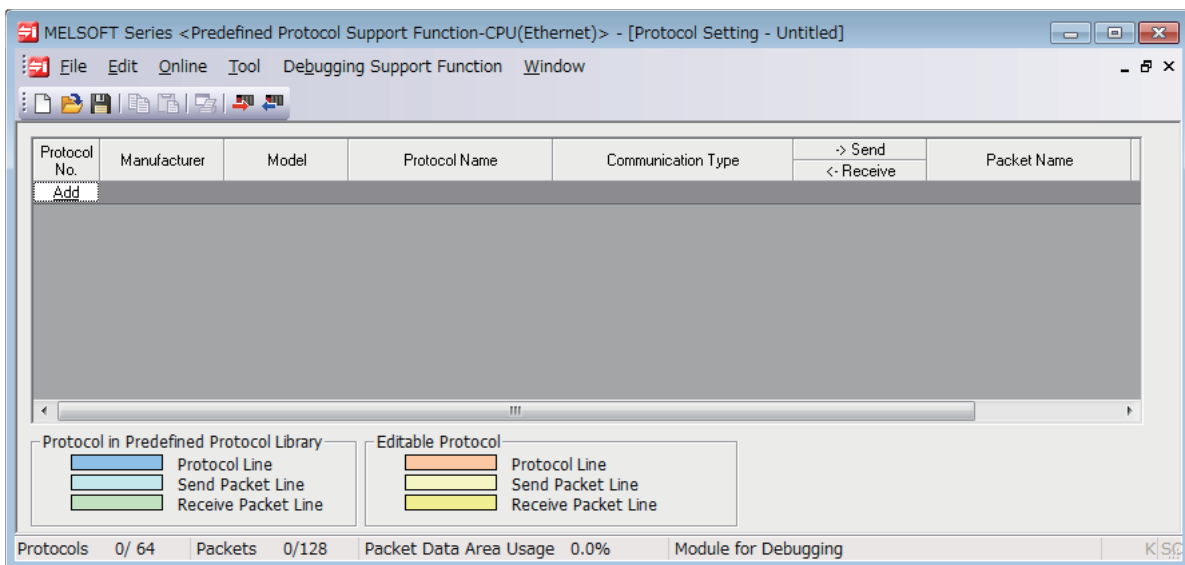
[Tool] ⇒ [Predefined Protocol Support Function]

6. Select "CPU(Ethernet)" for "Module Type" and click the [OK] button.



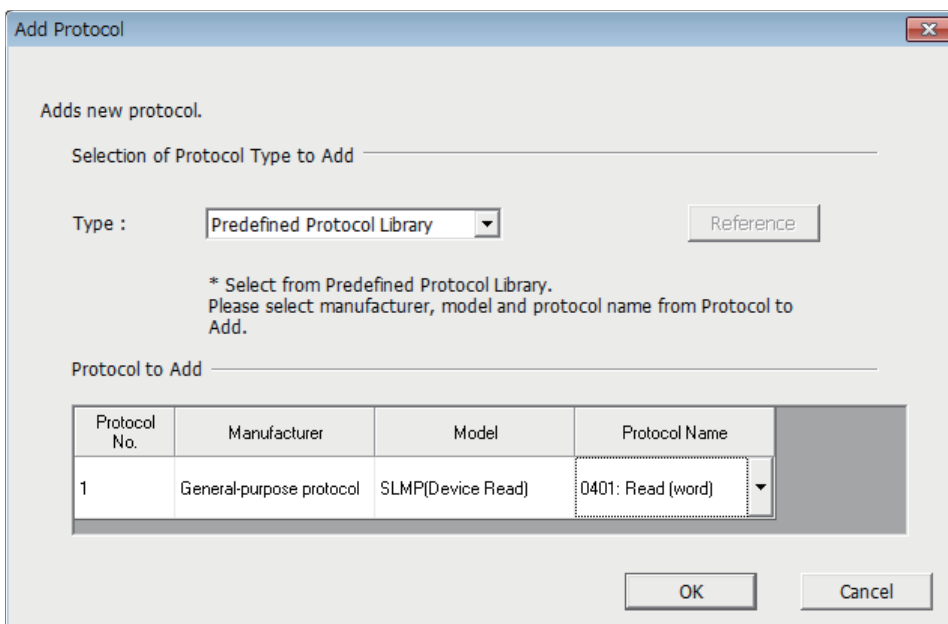
7. Newly create the protocol setting.

[File] ⇨ [New]




8. Set a protocol in the following.

[Edit] ⇨ [Add Protocol]



9. Set each packet in the following.

 "Protocol Setting" window ⇒ Packet to be set

- Request

Packet Setting

Protocol No. Protocol Name

Packet Type Packet Name

Element List

| Element No. | Element Type | Element Name | Element Setting |
|-------------|-------------------------|--------------------------|--|
| 1 | Static Data | (Fixed data) | 5400(2Byte) |
| 2 | Non-conversion Variable | Serial No. | [D0-D0](Fixed Length/2Byte/Lower/Upper Byte/No Swap) |
| 3 | Static Data | (Fixed data) | 0000(2Byte) |
| 4 | Non-conversion Variable | Network No. | [D1-D1](Fixed Length/1Byte/Lower/Upper Byte/No Swap) |
| 5 | Non-conversion Variable | Station No. | [D2-D2](Fixed Length/1Byte/Lower/Upper Byte/No Swap) |
| 6 | Non-conversion Variable | Requested module I/O No. | [D3-D3](Fixed Length/2Byte/Lower/Upper Byte/No Swap) |
| 7 | Static Data | For future expansion | 00(1Byte) |
| 8 | Length | Request data length | [Object element9-14/HEX/Reverse/2Byte] |
| 9 | Non-conversion Variable | Monitoring timer | [D4-D4](Fixed Length/2Byte/Lower/Upper Byte/No Swap) |
| 10 | Static Data | Command | 0104(2Byte) |
| 11 | Static Data | Subcommand | 0000(2Byte) |
| 12 | Non-conversion Variable | Head device No. | [D5-D5](Fixed Length/3Byte/Lower/Upper Byte/No Swap) |
| 13 | Non-conversion Variable | Device code | [D7-D7](Fixed Length/1Byte/Lower/Upper Byte/No Swap) |

Change Type Add New Copy Paste Delete

Close

- Normal response

Packet Setting

Protocol No. Protocol Name

Packet Type Packet Name

Packet No.

Element List

| Element No. | Element Type | Element Name | Element Setting |
|-------------|-------------------------|--------------------------|--|
| 1 | Static Data | (Fixed data) | D400(2Byte) |
| 2 | Non-conversion Variable | Serial No. | [D9-D9](Fixed Length/2Byte/Lower/Upper Byte/No Swap) |
| 3 | Static Data | (Fixed data) | 0000(2Byte) |
| 4 | Non-conversion Variable | Network No. | [D10-D10](Fixed Length/1Byte/Lower/Upper Byte/No Swap) |
| 5 | Non-conversion Variable | Station No. | [D11-D11](Fixed Length/1Byte/Lower/Upper Byte/No Swap) |
| 6 | Non-conversion Variable | Requested module I/O No. | [D12-D12](Fixed Length/2Byte/Lower/Upper Byte/No Swap) |
| 7 | Static Data | For future expansion | 00(1Byte) |
| 8 | Length | Response data length | [Object element9-10/HEX/Reverse/2Byte] |
| 9 | Static Data | End code | 0000(2Byte) |
| 10 | Non-conversion Variable | Response data | [D13][D14-D973](Variable Length/1920Byte/Lower/Upper Byte/No Swap) |

Change Type Add New Copy Paste Delete

Close

- Error response

Packet Setting

Protocol No. Protocol Name

Packet Type Packet Name

Packet No.

Element List

| Element No. | Element Type | Element Name | Element Setting |
|-------------|-------------------------|--------------------------|--|
| 1 | Static Data | (Fixed data) | D400(2Byte) |
| 2 | Non-conversion Variable | Serial No. | [D974-D974](Fixed Length/2Byte/Lower/Upper Byte/No Swap) |
| 3 | Static Data | (Fixed data) | 0000(2Byte) |
| 4 | Non-conversion Variable | Network No. | [D975-D975](Fixed Length/1Byte/Lower/Upper Byte/No Swap) |
| 5 | Non-conversion Variable | Station No. | [D976-D976](Fixed Length/1Byte/Lower/Upper Byte/No Swap) |
| 6 | Non-conversion Variable | Requested module I/O No. | [D977-D977](Fixed Length/2Byte/Lower/Upper Byte/No Swap) |
| 7 | Static Data | For future expansion | 00(1Byte) |
| 8 | Length | Response data length | [Object element9-15/HEX/Reverse/2Byte] |
| 9 | Non-conversion Variable | End code | [D978-D978](Fixed Length/2Byte/Lower/Upper Byte/No Swap) |
| 10 | Non-conversion Variable | Network No. | [D979-D979](Fixed Length/1Byte/Lower/Upper Byte/No Swap) |
| 11 | Non-conversion Variable | Station No. | [D980-D980](Fixed Length/1Byte/Lower/Upper Byte/No Swap) |
| 12 | Non-conversion Variable | Requested module I/O No. | [D981-D981](Fixed Length/2Byte/Lower/Upper Byte/No Swap) |
| 13 | Static Data | For future expansion | 00(1Byte) |

Change Type Add New Copy Paste Delete

Close

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

[Online] ⇄ [Write to Module]

Module Write

Target Module Selection

Module Selection

Target Memory

Please save in the protocol setting file because the data to be written to target memory does not include the following information.

[Non-written Data in Target Memory]
 Manufacturer
 Packet Name
 Protocol Detailed Setting Type, Version, Explanation
 Packet Setting Configuration Element Name

Execute Cancel

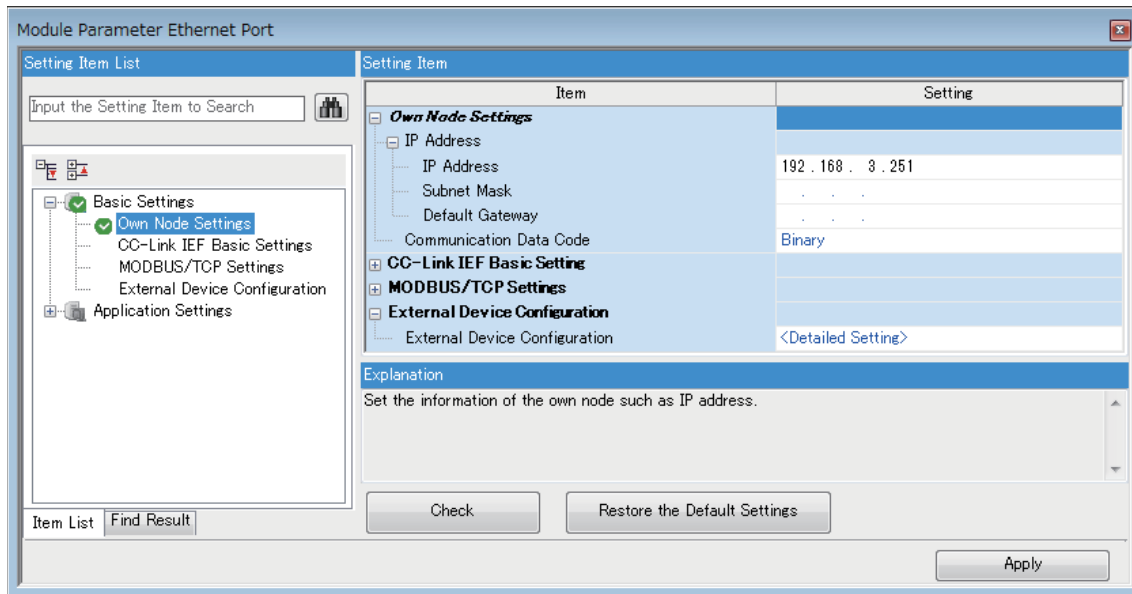
11. Write the set parameters to the CPU module. Then reset the CPU module or power off and on the system.

[Online] ⇄ [Write to PLC]

■Receiving side

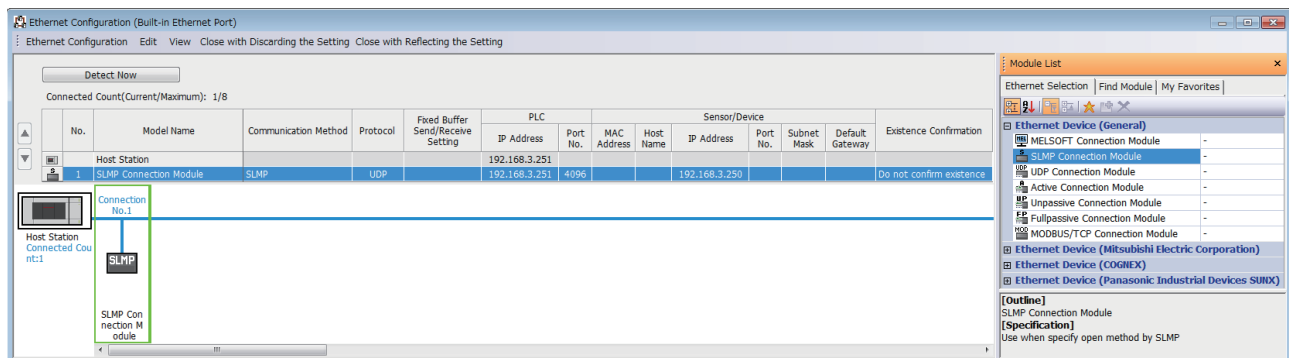
1. Set the CPU module and add the module labels of the CPU module. The setting method of the CPU module and addition method of the module label are the same as those of when setting the sending side. (☞ Page 77 Sending side)
2. Set the "Basic Settings" in the following.

☞ Navigation window⇒[Parameter]⇒[FX5UJCPU] or [FX5UCPU]⇒[Module Parameter]⇒[Ethernet Port]⇒[Basic Settings]



3. Set the external device configuration in the following.

☞ Navigation window⇒[Parameter]⇒[FX5UJCPU] or [FX5UCPU]⇒[Module Parameter]⇒[Ethernet Port]⇒[Basic Settings]⇒[External Device Configuration]⇒[Detailed Setting]⇒[Ethernet Configuration (Built-in Ethernet Port)] screen



4. Write the set parameters to the CPU module. Then reset the CPU module or power off and on the system.

☞ [Online] ⇒ [Write to PLC]

6.6 Predefined Protocol Support Function Instruction

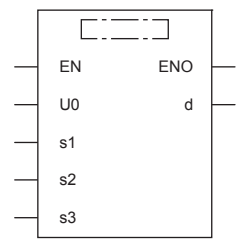
Executing the registered protocols

SP.ECPRTCL

This instruction executes the communication protocol registered using the engineering tool.

| Ladder diagram | Structured text |
|----------------|---|
| | <pre>ENO:=SP_ECPRTCL(EN,U0,s1,s2,s3,d);</pre> |

FBD/LD



("SP_ECPRTCL" enters □.)

Setting data

■ Descriptions, ranges, and data types

| Operand | Description | Range | Data type | Data type (label) |
|-------------------|---|---------------------------------|------------------------|---|
| (U) ^{*1} | Dummy (Input the character string [U0].) | — | Character string | ANYSTRING_SINGLE |
| (s1) | Connection number | 1 to 8 | 16-bit unsigned binary | ANY16 |
| (s2) | Number of protocols to be executed continuously | 1 to 8 | 16-bit unsigned binary | ANY16 |
| (s3) | Head device number for storing the control data | Refer to Control data (Page 84) | Word | ANY16_ARRAY (Number of elements: 18) |
| (d) | Head device number which turns ON when the execution of the instruction is completed and remains on for 1 scan. If the instruction is completed with an error, (d)+1 is also turned on. | — | Bit | ANYBIT_ARRAY (Number of elements: 2) |
| EN | Execution condition | — | Bit | BOOL |
| ENO | Execution result | — | Bit | BOOL |

*1 In the case of the ST language and the FBD/LD language, U displays as U0.

■ Applicable devices

| Operand | Bit | Word | | | Double word | | Indirect specification | Constant | | | Others |
|---------|-----------------------------|---------------------------|-------|---|-------------|----|------------------------|----------|---|----|--------|
| | X, Y, M, L, SM, F, B, SB, S | T, ST, C, D, W, SD, SW, R | U□\G□ | Z | LC | LZ | | K, H | E | \$ | |
| (U) | — | ○ | — | — | — | — | ○ | — | — | ○ | — |
| (s1) | ○ | ○ | — | — | — | — | ○ | ○ | — | — | — |
| (s2) | ○ | ○ | — | — | — | — | ○ | ○ | — | — | — |
| (s3) | ○ | ○ | — | — | — | — | ○ | — | — | — | — |
| (d) | ○ | ○ ^{*1} | — | — | — | — | — | — | — | — | — |

*1 T, ST, C cannot be used.

■Control data

| Device | Item | Description | Setting range | Set by*1 |
|---------|---|--|---------------|----------|
| (s3)+0 | Resulting number of executed protocols | The number of protocols executed by the SP.ECPRTCL instruction is stored. Any protocol where an error occurred is also included in the execution number. If the setting of setting data or control data contains an error, "0" is stored. | 0, 1 to 8 | System |
| (s3)+1 | Completion status | The completion status is stored upon completion of the instruction. When two or more protocols are executed, the execution result of the protocol executed last is stored. <ul style="list-style-type: none"> • 0: Normal completion • Other than 0: Error completion (error code) | — | |
| (s3)+2 | Execution protocol number 1 | Specify the number of the protocol to be executed first. | 1 to 64 | User |
| (s3)+3 | Execution protocol number 2 | Specify the number of the protocol to be executed second. | 0, 1 to 64 | |
| (s3)+4 | Execution protocol number 3 | Specify the number of the protocol to be executed third. | 0, 1 to 64 | |
| (s3)+5 | Execution protocol number 4 | Specify the number of the protocol to be executed fourth. | 0, 1 to 64 | |
| (s3)+6 | Execution protocol number 5 | Specify the number of the protocol to be executed fifth. | 0, 1 to 64 | |
| (s3)+7 | Execution protocol number 6 | Specify the number of the protocol to be executed sixth. | 0, 1 to 64 | |
| (s3)+8 | Execution protocol number 7 | Specify the number of the protocol to be executed seventh. | 0, 1 to 64 | |
| (s3)+9 | Execution protocol number 8 | Specify the number of the protocol to be executed eighth. | 0, 1 to 64 | |
| (s3)+10 | Collation match Receive packet number 1 | If receiving is included in the communication type of the protocol that has been executed first, the receive packet number successful in collation match is stored. If the communication type is "receive only", "0" is stored. If an error occurs during execution of the first protocol, "0" is stored. | 0, 1 to 16 | System |
| (s3)+11 | Collation match Receive packet number 2 | If receiving is included in the communication type of the protocol that has been executed second, the receive packet number successful in collation match is stored. If the communication type is "receive only", "0" is stored. If an error occurs during execution of the second protocol, "0" is stored. If the number of protocols executed is less than 2, "0" is stored. | 0, 1 to 16 | |
| (s3)+12 | Collation match Receive packet number 3 | If receiving is included in the communication type of the protocol that has been executed third, the receive packet number successful in collation match is stored. If the communication type is "receive only", "0" is stored. If an error occurs during execution of the third protocol, "0" is stored. If the number of protocols executed is less than 3, "0" is stored. | 0, 1 to 16 | |
| (s3)+13 | Collation match Receive packet number 4 | If receiving is included in the communication type of the protocol that has been executed fourth, the receive packet number successful in collation match is stored. If the communication type is "receive only", "0" is stored. If an error occurs during execution of the fourth protocol, "0" is stored. If the number of protocols executed is less than 4, "0" is stored. | 0, 1 to 16 | |
| (s3)+14 | Collation match Receive packet number 5 | If receiving is included in the communication type of the protocol that has been executed fifth, the receive packet number successful in collation match is stored. If the communication type is "receive only", "0" is stored. If an error occurs during execution of the fifth protocol, "0" is stored. If the number of protocols executed is less than 5, "0" is stored. | 0, 1 to 16 | |
| (s3)+15 | Collation match Receive packet number 6 | If receiving is included in the communication type of the protocol that has been executed sixth, the receive packet number successful in collation match is stored. If the communication type is "receive only", "0" is stored. If an error occurs during execution of the sixth protocol, "0" is stored. If the number of protocols executed is less than 6, "0" is stored. | 0, 1 to 16 | |
| (s3)+16 | Collation match Receive packet number 7 | If receiving is included in the communication type of the protocol that has been executed seventh, the receive packet number successful in collation match is stored. If the communication type is "receive only", "0" is stored. If an error occurs during execution of the seventh protocol, "0" is stored. If the number of protocols executed is less than 7, "0" is stored. | 0, 1 to 16 | |
| (s3)+17 | Collation match Receive packet number 8 | If receiving is included in the communication type of the protocol that has been executed eighth, the receive packet number successful in collation match is stored. If the communication type is "receive only", "0" is stored. If an error occurs during execution of the eighth protocol, "0" is stored. If the number of protocols executed is less than 8, "0" is stored. | 0, 1 to 16 | |

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

Processing details

This instruction executes the protocol registered using the engineering tool. Using the connection specified by (s1), the instruction executes the protocol in accordance with the control data stored in the device specified by (s3) and later. The instruction continuously executes as many protocols as specified by (s2) (a maximum of 8 protocols) at one time.

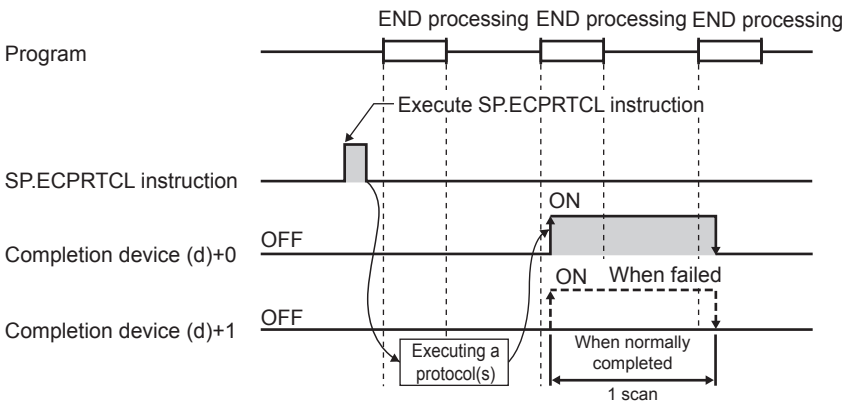
The number of executed protocols is stored in the device specified by (s3)+0.

The completion of the SP.ECPRTCL instruction can be checked using the completion devices (d)+0 and (d)+1.

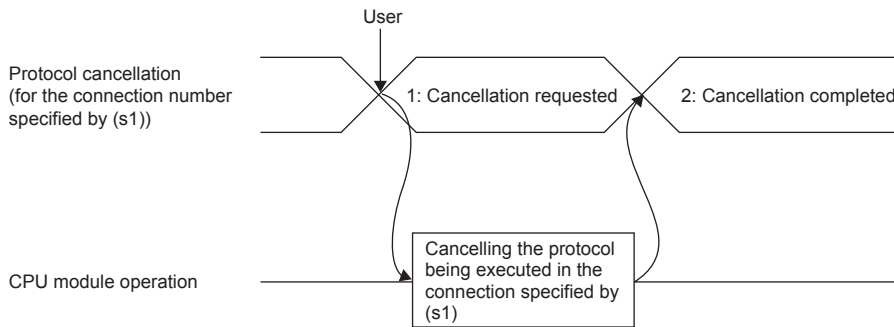
- Completion device (d)+0: Turns ON during the END processing for the scan in which the SP.ECPRTCL instruction is completed, and turns OFF during the next END processing.
- Completion device (d)+1: Turns ON or OFF depending on the status when the SP.ECPRTCL instruction is completed.

| Status | Description |
|------------------------------|--|
| When completed normally | The device does not change (remains OFF). |
| When completed with an error | The device turns ON during the END processing for the scan in which the SP.ECPRTCL instruction is completed, and turns OFF during the next END processing. |

- The following figure shows the SP.ECPRTCL instruction execution timing.



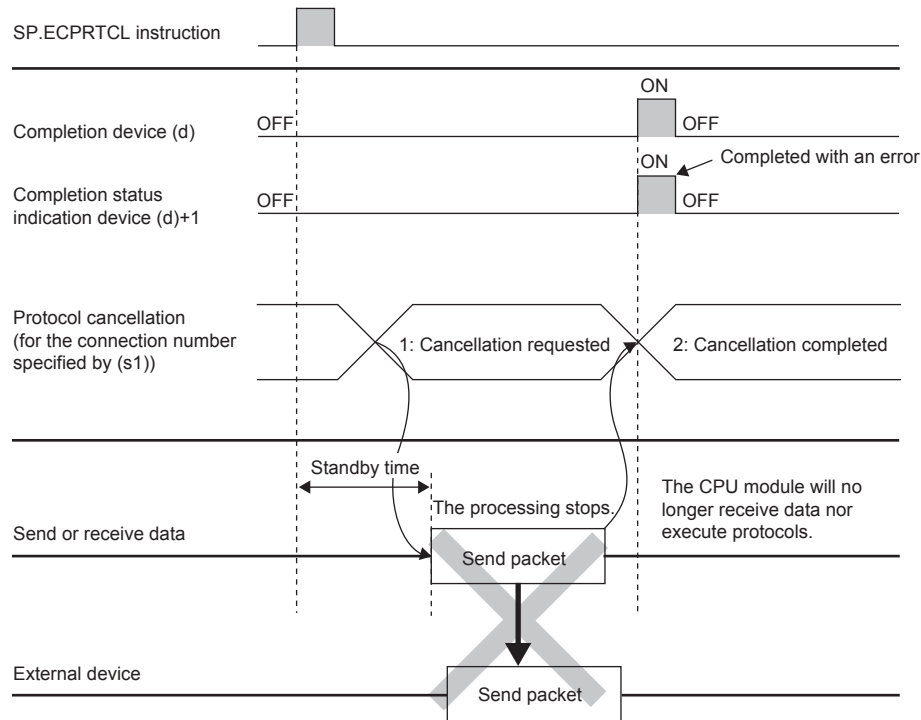
- Protocol execution can be canceled by setting a protocol cancel request. The protocol cancel request is specified in the predefined protocol support function execution status check area (SD10740 to SD10899).



• The following figure shows the protocol cancel operations timing.

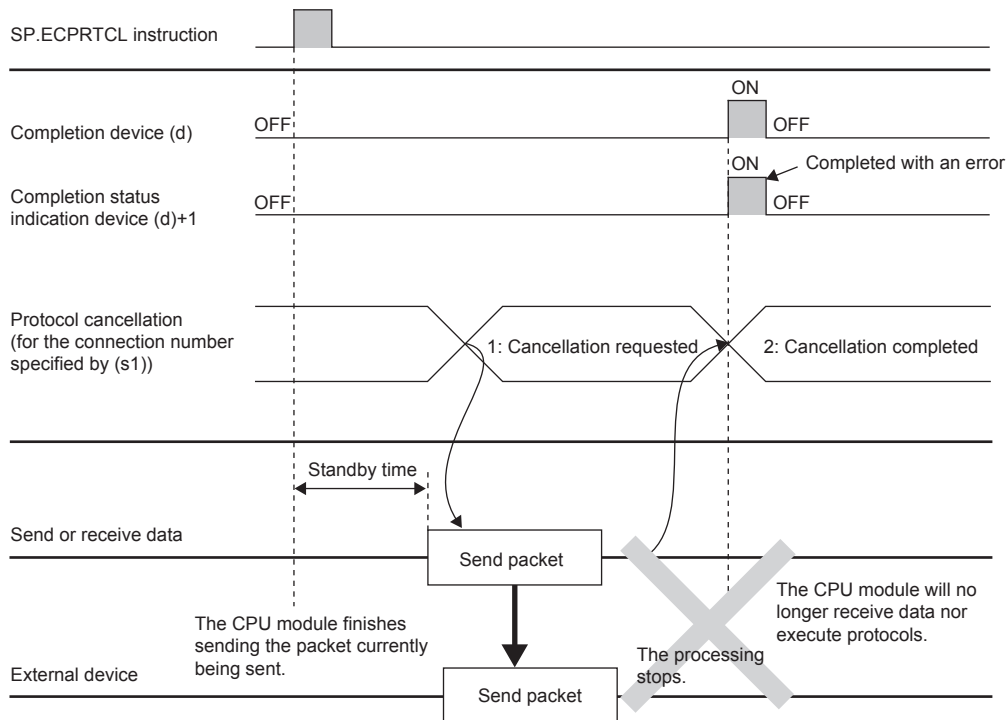
■ If a cancel request is issued before transmission

The following figure shows the operation when the protocol execution status is "1: Waiting for transmission".



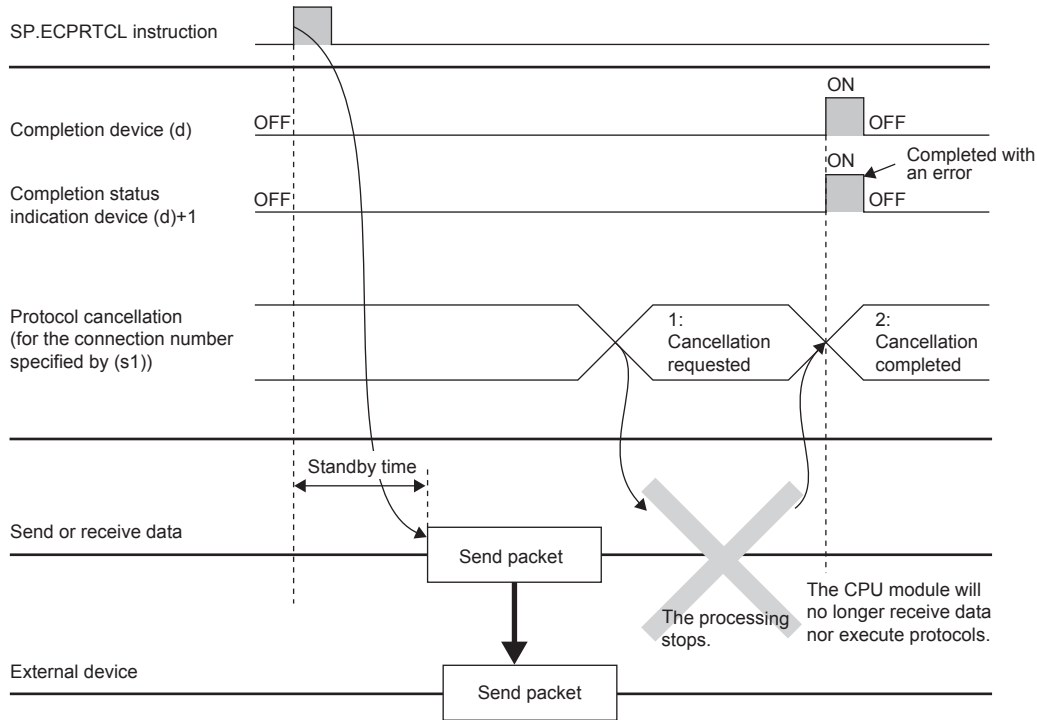
■ If a cancel request is issued before completion of transmission

The following figure shows the operation when transmission has not been completed while the protocol execution status is "2: Sending".



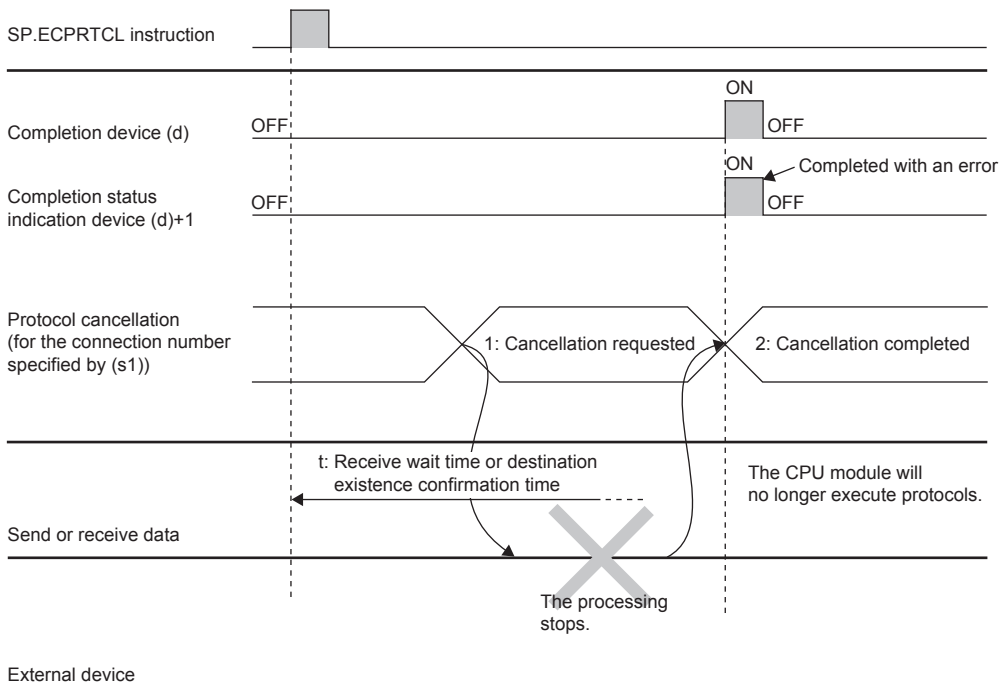
■ If a cancel request is issued upon completion of transmission

The following figure shows the operation when transmission has been completed while the protocol execution status is "2: Sending".



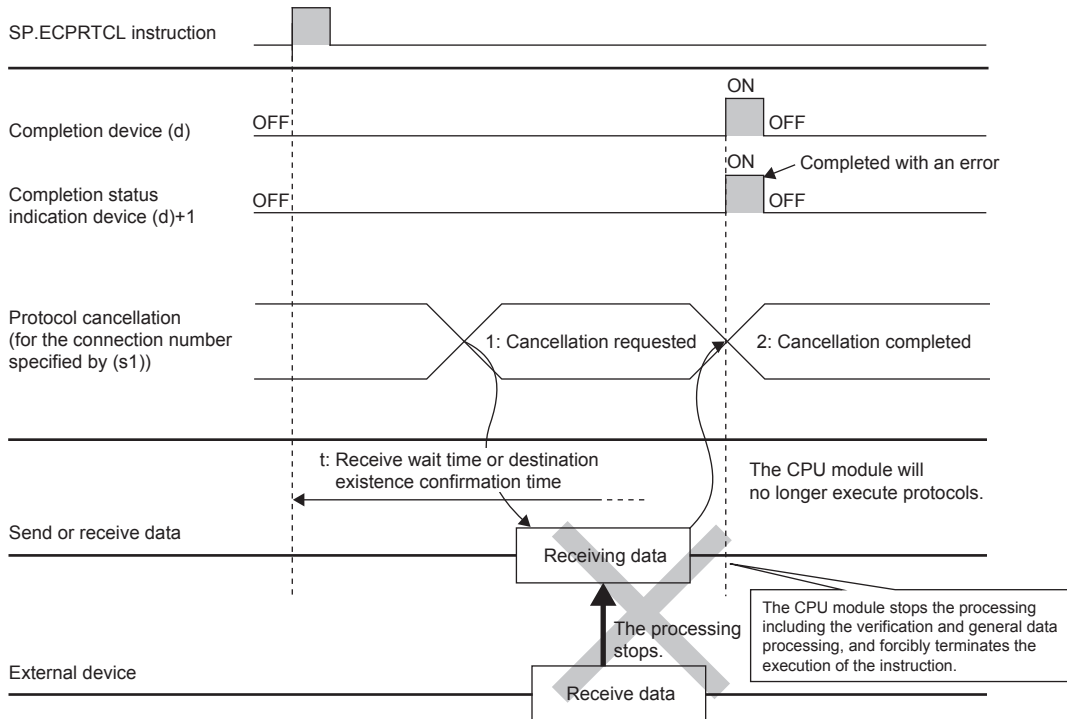
■ If a cancel request is issued while waiting for reception

The following figure shows the operation when the protocol execution status is "3: Waiting for data reception".



■ If a cancel request is issued during receiving

The following figure shows the operation when the protocol execution status is "4: Receiving".



Precautions

- If an error occurs in the mth protocol while multiple protocols are being executed, the instruction does not execute the "m+1"th protocol and after and is completed with an error.
- The connections for which the SP.ECPRTCL instruction can be executed are only those for which "Communication protocol" is specified for the communication means.
- If a cancel request is received during execution of the mth protocol while multiple protocols are executed continuously, following is stored in (s3).

| Device | Item | Description |
|----------|--|--|
| (s3)+0 | Resulting number of executed protocols | The executed protocol number. |
| (s3)+1 | Completion status | The error codes. |
| (s3)+10 | Collation match Receive packet number 1 | The receive packet number successful in collation match for the already executed protocol. |
| ⋮ | ⋮ | |
| (s3)+m+8 | Collation match Receive packet number m-1 | |

- If same instructions are executed for the same connection, the subsequent instruction is ignored and is not executed until the preceding instruction is completed.
- The SP.ECPRTCL instruction itself does not open/close a connection and therefore the SP.SOCOPEN/SP.SOCCLOSE instructions need to be used to open/close the connection.

☞ Refer to the Page 105 SP.SOCOPEN and Page 113 SP.SOCCLOSE

Operation error

| Error code (SD0/SD8067) | Description |
|-------------------------|--|
| 2820H | The device used exceeded the specified range. |
| 2821H | The device used to store data are overlapping. |
| 2822H | Device that cannot be specified is specified. |
| 3405H | The input data was out of range. |

6.7 Precautions

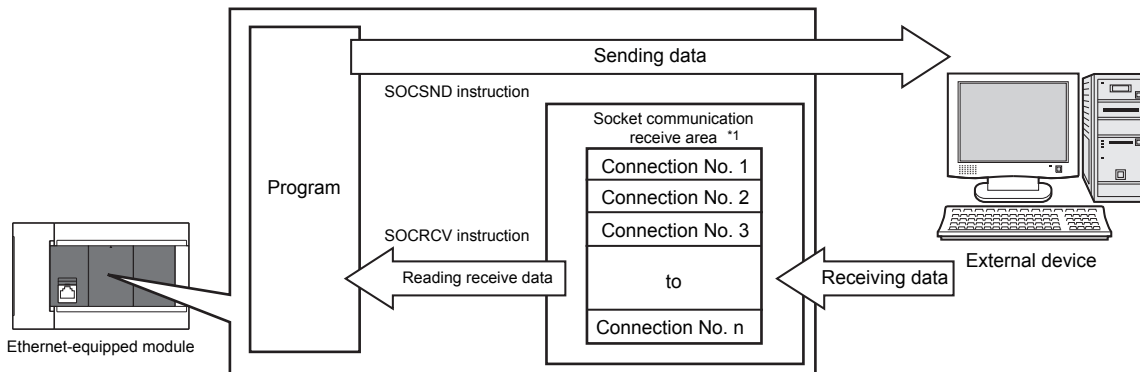
This section provides other precautions for the predefined protocol support function.

Port number

Host station port number, 1 to 1023 (0001H to 03FFH), are assigned for reserved port numbers (WELL KNOWN PORT NUMBERS) and 61440 to 65534 (F000H to FFFEH) are for other communication functions. Therefore, using 1024 to 5548, 5570 to 61439 (0400H to 15ACH, 15C2H to EFFFH) is recommended.

7 SOCKET COMMUNICATION FUNCTION

The socket communication function allows data communication 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.
 CPU module: Connection No.1 to No.8
 Ethernet module: Connection No.1 to No.32

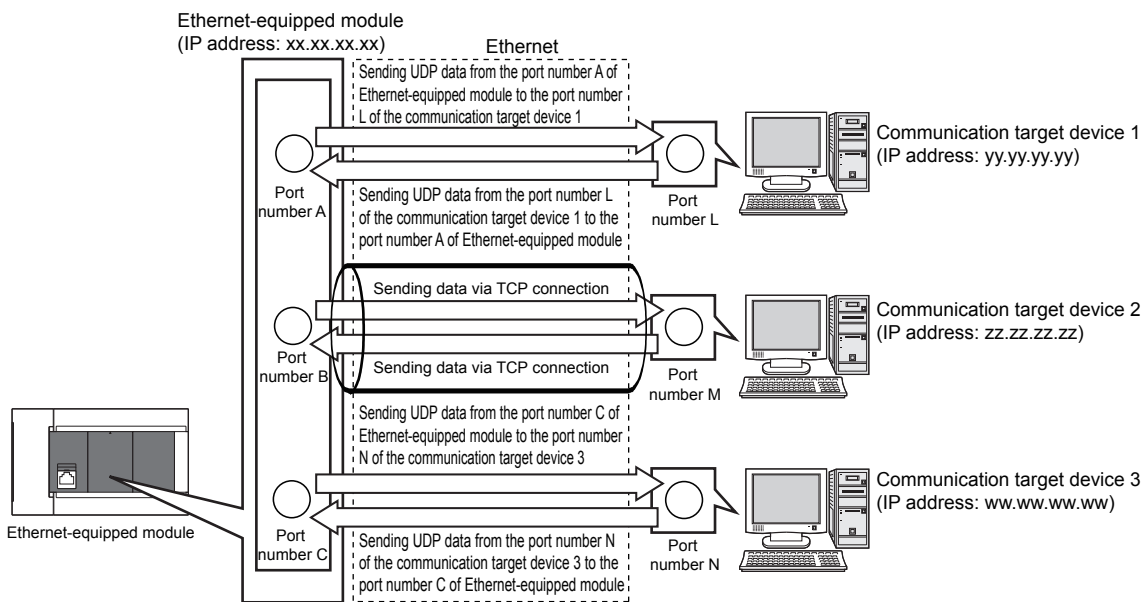
Point

- For dedicated instructions used for the socket communication function, refer to [Page 104 Socket Communication Function Instructions](#).
- Access through routers is also available. When configuring the settings set the subnet mask pattern and default gateway IP address. ([Page 39 Communication via Router](#))

Port numbers

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

- For sending: Specify the port number of the Ethernet-equipped module from which data is sent, and the port number of the destination device.
- For receiving: Specify the port number of the Ethernet-equipped module, and read the data sent to the port.



7.1 Communication Using TCP

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

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 Ethernet-equipped module
- Which side will open a connection, the target device or Ethernet-equipped module? (Active open or Passive open)

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.

■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.

■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 CPU module can acquire the IP address and port number of the connected target device 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. |

Point

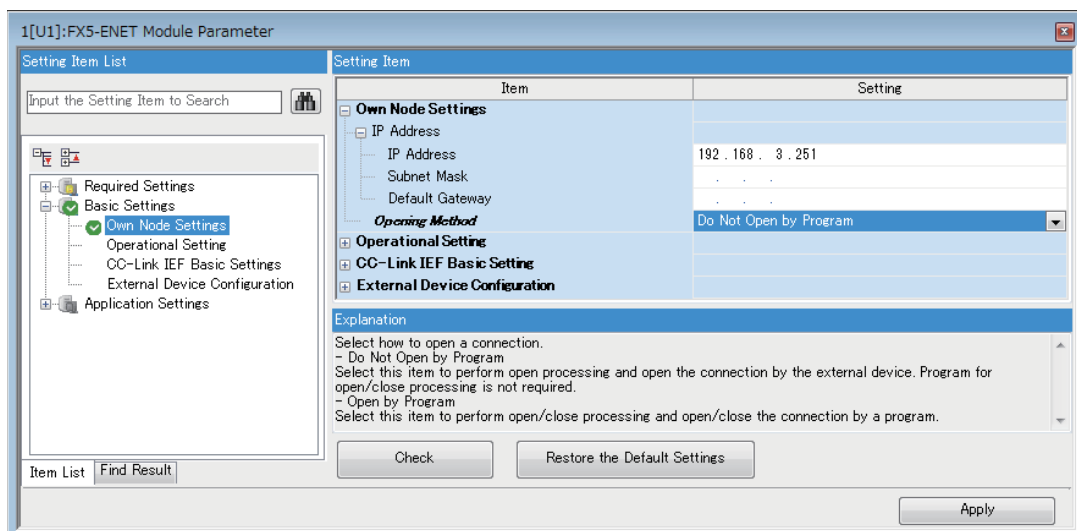
The expressions of Active and Passive opens may vary according to the device.

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

Open/close processing procedure

The open/close processing procedure for the Ethernet module in the Passive open mode varies depending on the "Opening Method".

Navigation window⇒[Parameter]⇒[Module Information]⇒[FX5-ENET] or [FX5-ENET/IP]⇒[Basic Settings]⇒[Own Node Settings]



■When "Do Not Open by Program" is set

The Ethernet module is constantly in the open standby state, so the connection is established when Active open is initiated by the external device. This eliminates the need for an open processing program on the Ethernet module side.

Point

When the close processing is executed with a dedicated instruction from the Ethernet module, even if "Do Not Open by Program" is set, the connection will not return to the open request standby state after the close processing completes.

■When "Open by Program" is set

Before the open request is received from the external device, the Ethernet module must execute the GP.OPEN instruction and enter the open standby state. Data can be sent and received after the open processing completes normally.

Point

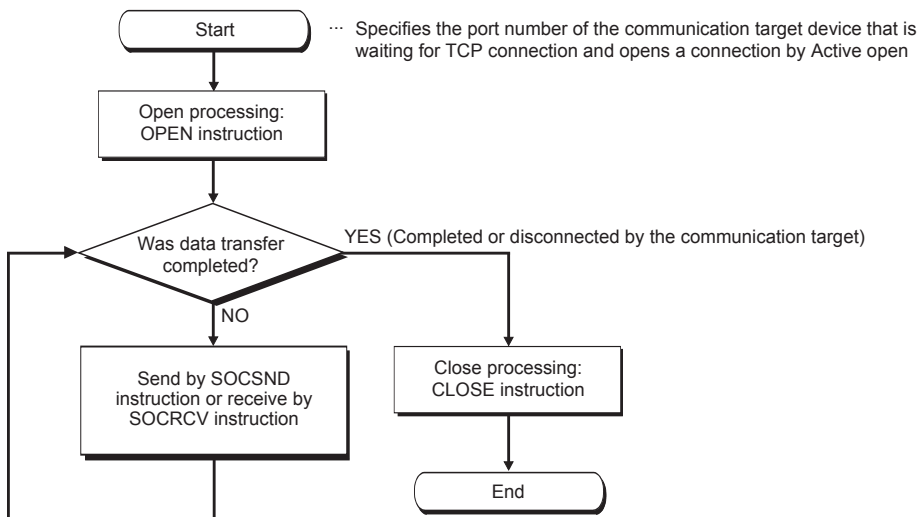
- Change the connection setting before executing the GP.OPEN instruction.
- After the open processing is executed, the open request cannot be canceled until the open processing completes. Execute the close processing (GP.CLOSE instruction) after open completes.

Program example

This section provides a program example for communication using TCP.

Program example for Active open

The following shows the communication flow of an Active open.

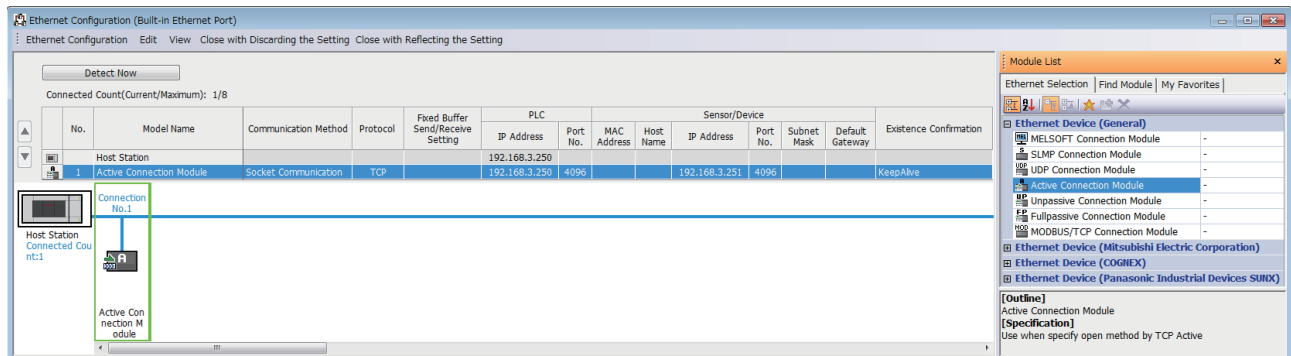


Parameter setting

The following parameters are set for the sample program.

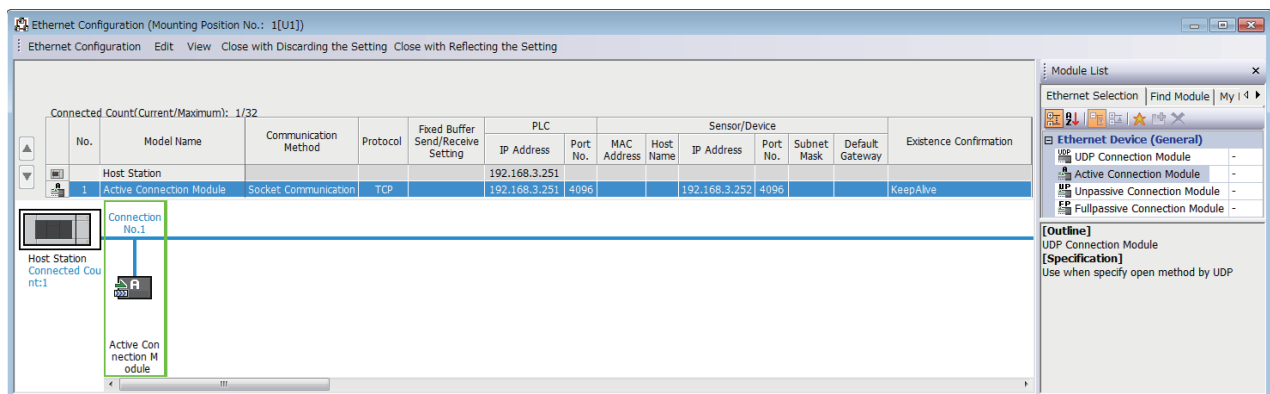
[CPU module]

Navigation window⇒[Parameter]⇒[FX5UJCPU] or [FX5UCPU]⇒[Module Parameter]⇒[Ethernet Port]⇒[Basic Settings]⇒[External Device Configuration]⇒[Detailed Setting]⇒[Ethernet Configuration (Built-in Ethernet Port)] screen



[Ethernet module]

Navigation window⇒[Parameter]⇒[Module Information]⇒[FX5-ENET] or [FX5-ENET/IP]⇒[Basic Settings]⇒[External Device Configuration]⇒[Ethernet Configuration (Mounting Position No.: n[Un])] screen



- Drag and drop the "Active Connection Module" from "Module List" to the left side on the window. Execute the settings as mentioned below.

| Item | Description | |
|---------------|-------------|---|
| PLC | Port No. | 4096 (Setting range: 1 to 5548, 5670 to 65534) Do not specify 5549 to 5569 because these ports are used by the system. |
| Sensor/Device | IP Address | 192.168.3.251 (Setting range: 0.0.0.1 to 223.255.255.254) |
| | Port No. | 4096 (Setting range: 1 to 65534) |

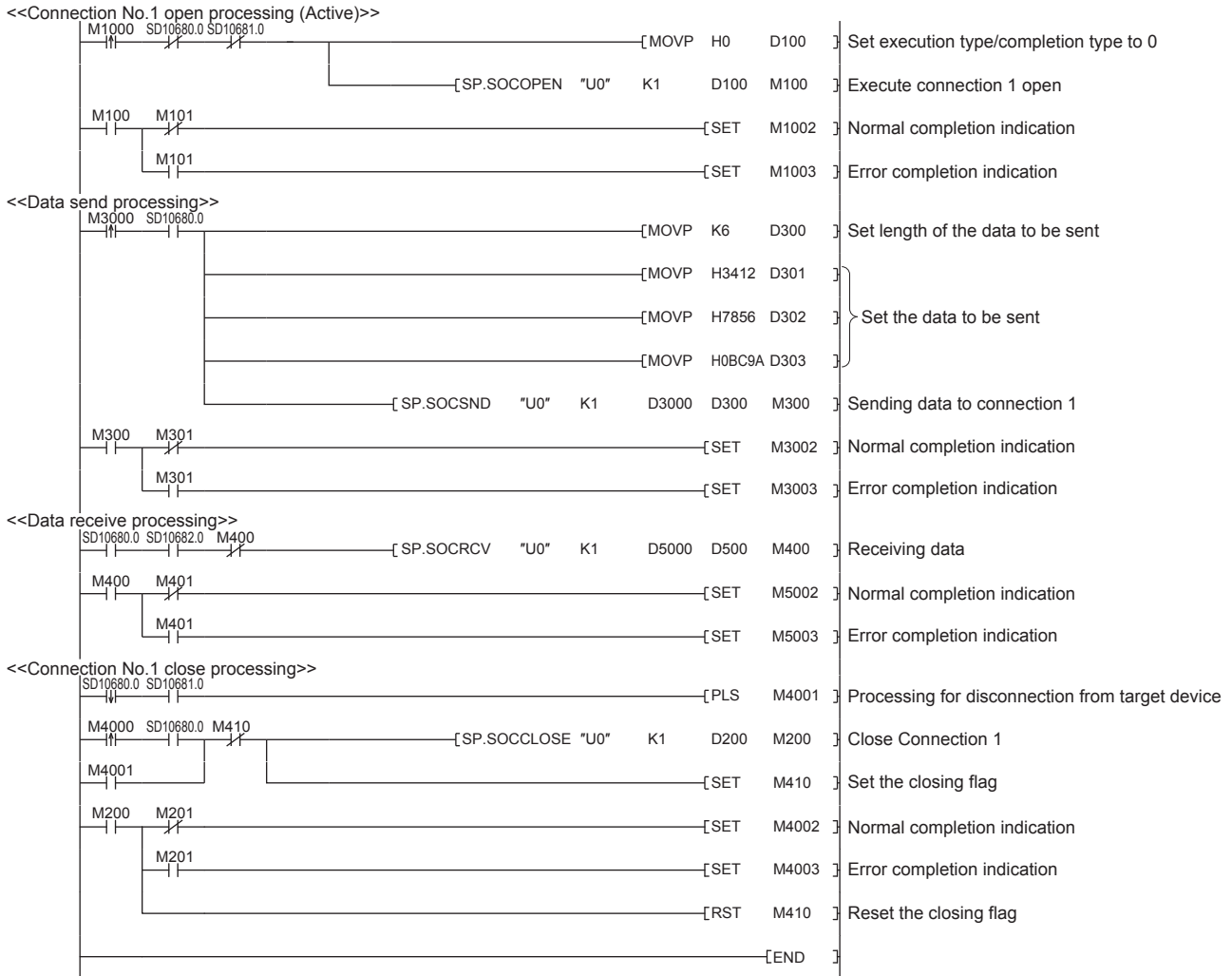
■Devices used in the sample program

The following table lists the device numbers used in the sample program and their applications. The devices used when the CPU module is used are shown below.

| Device No. | Application |
|-----------------|---|
| M1000 | Open instruction |
| D100 to D109 | SP.SOCOPEN instruction control data |
| M100 | SP.SOCOPEN instruction normal completion device |
| M101 | SP.SOCOPEN instruction error completion device |
| M1002 | Normal open indication |
| M1003 | Open error indication |
| M3000 | Send instruction |
| D3000 and D3001 | SP.SOCSND instruction control data |
| M300 | SP.SOCSND instruction normal completion device |
| M301 | SP.SOCSND instruction error completion device |
| D300 to D303 | Send data length and send data (6 bytes of 12H, 34H, 56H, 78H, 9AH, BCH) |
| M3002 | Normal send indication |
| M3003 | Send error indication |
| M4000 | Close instruction |
| M4001 | Disconnection by the other device |
| SD10680 | Open completion signal |
| SD10681 | Open request signal |
| SD10682 | Receive state signal |
| D200 and D201 | SP.SOCCLOSE instruction control data |
| M200 | SP.SOCCLOSE instruction normal completion device |
| M201 | SP.SOCCLOSE instruction error completion device |
| M4002 | Normal close indication |
| M4003 | Close error indication |
| M410 | Closing flag |
| D5000 and D5001 | SP.SOCRCV instruction control data |
| M400 | SP.SOCRCV instruction normal completion device |
| M401 | SP.SOCRCV instruction error completion device |
| D500 and higher | Received data length and received data |
| M5002 | Normal receive indication |
| M5003 | Receive error indication |

Sample program

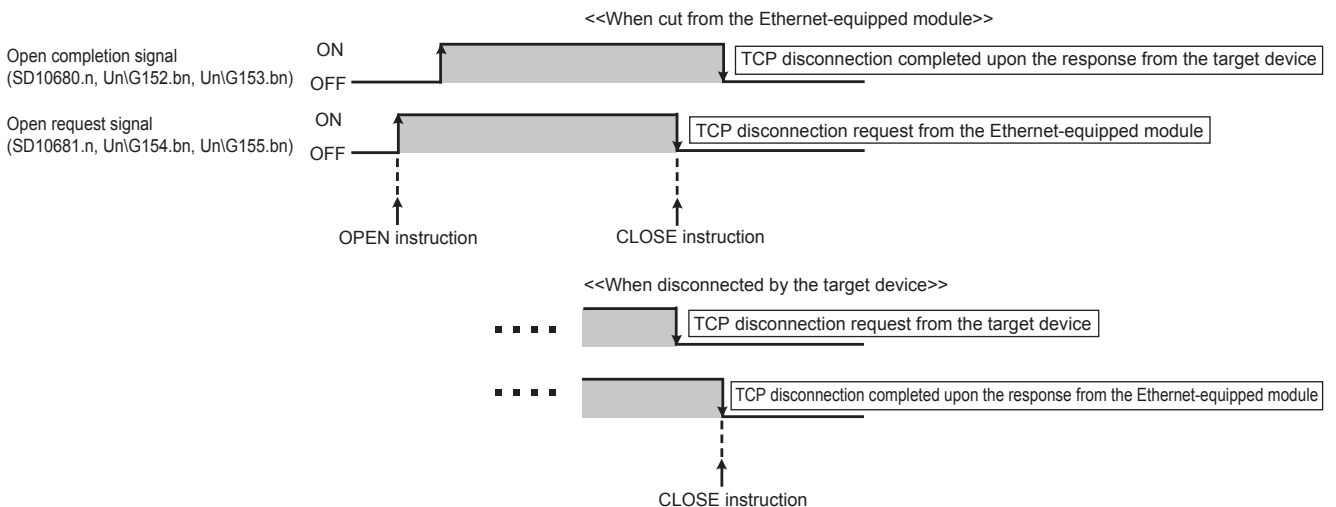
A program example using the CPU module is shown below.



Precautions for Active open communication

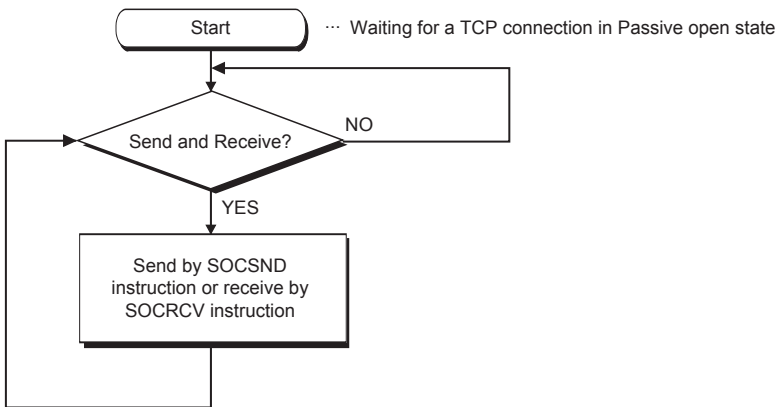
Configure an interlock circuit using the Open completion signal (SD10680.n, UnG152.bn, UnG153.bn) and Open request signal (SD10681.n, UnG154.bn, UnG155.bn) in the program.

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



Program example for Passive open

The following shows the communication flow of a Passive open.

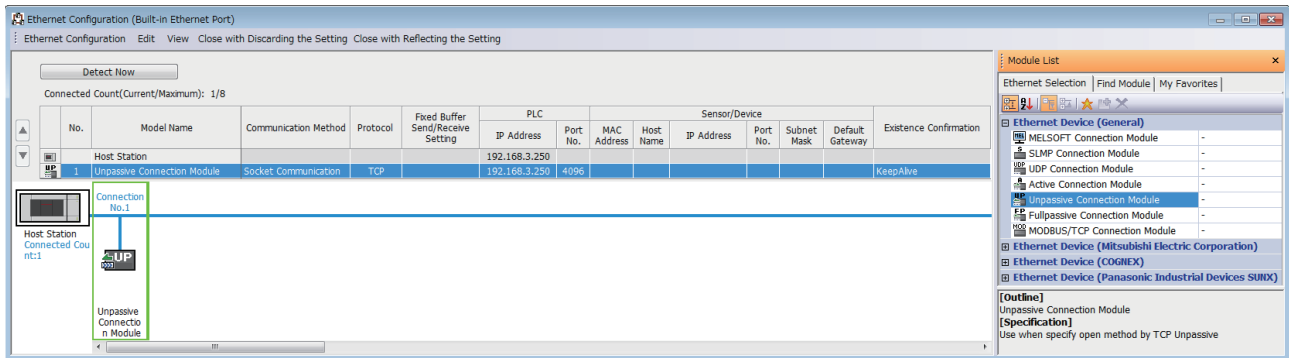


Parameter setting

The following parameters are set for the sample program.

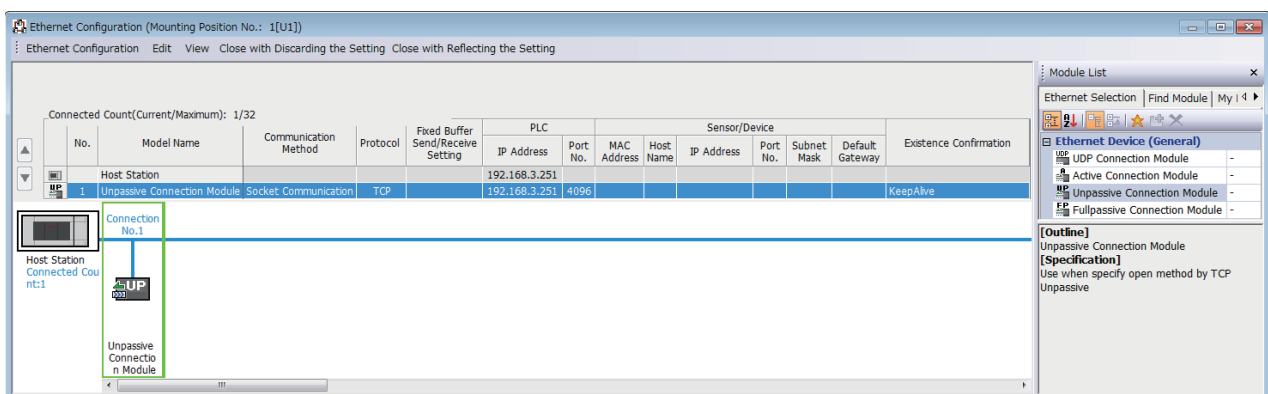
[CPU module]

Navigation window ⇒ [Parameter] ⇒ [FX5UJCPU] or [FX5UCPU] ⇒ [Module Parameter] ⇒ [Ethernet Port] ⇒ [Basic Settings] ⇒ [External Device Configuration] ⇒ [Detailed Setting] ⇒ [Ethernet Configuration (Built-in Ethernet Port)] screen



[Ethernet module]

Navigation window ⇒ [Parameter] ⇒ [Module Information] ⇒ [FX5-ENET] or [FX5-ENET/IP] ⇒ [Basic Settings] ⇒ [External Device Configuration] ⇒ [Ethernet Configuration (Mounting Position No.: n[Un])] screen



- Drag and drop the "Unpassive Connection Module" or "Fullpassive Connection Module" from "Module List" to the left side on the window. Execute the settings as mentioned below.

| Item | | Description |
|---------------|------------|---|
| PLC | Port No. | 4096 (Setting range: 1 to 5548, 5670 to 65534) Do not specify 5549 to 5569 because these ports are used by the system. |
| Sensor/Device | IP Address | Blank When "General Socket 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 "General Socket Fullpassive Connection Module" is selected, a value must be set. (Setting range: 1 to 65534) |

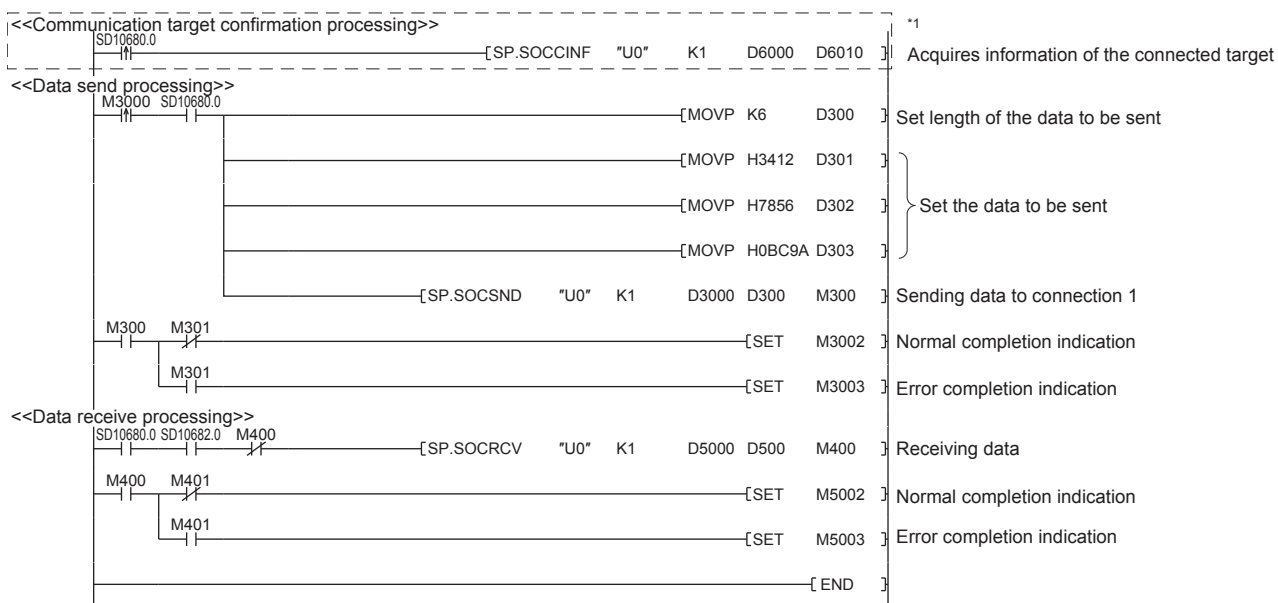
■Devices used in the sample program

The following table lists the device numbers used in the sample program and their applications. The devices used when the CPU module is used are shown below.

| Device No. | Application |
|-----------------|---|
| M3000 | Send instruction |
| D3000 and D3001 | SP.SOCSND instruction control data |
| M300 | SP.SOCSND instruction normal completion device |
| M301 | SP.SOCSND instruction error completion device |
| D300 to D303 | Send data length and send data (6 bytes of 12H, 34H, 56H, 78H, 9AH, BCH) |
| M3002 | Normal send indication |
| M3003 | Send error indication |
| SD10680 | Open completion signal |
| SD10682 | Receive state signal |
| D5000 and D5001 | SP.SOCRCV instruction control data |
| M400 | SP.SOCRCV instruction normal completion device |
| M401 | SP.SOCRCV instruction error completion device |
| D500 and higher | Received data length and received 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 |

■Sample program

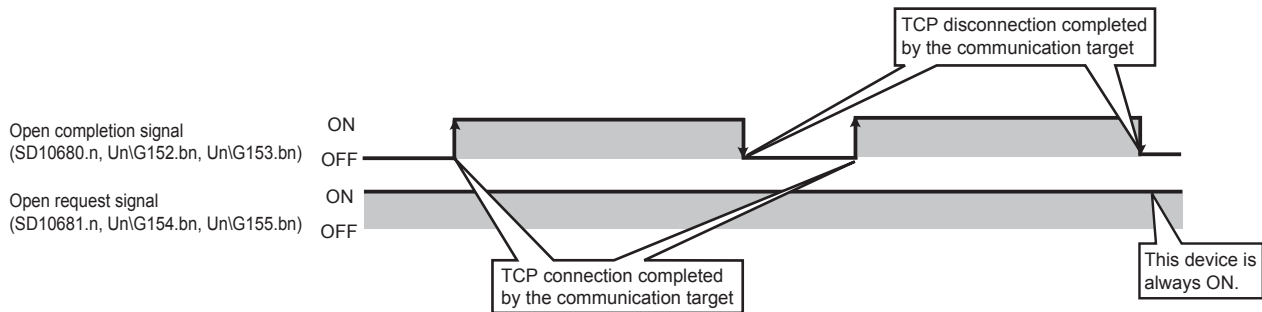
A program example using the CPU module is shown below.



*1 For acquiring information of the device connected with TCP, run the program enclosed by the dotted line.
(It can be omitted when the information acquisition is not needed.)

■ Precautions for Passive open communication

- Configure an interlock circuit using the Open completion signal (SD10680.n, Un\G152.bn, Un\G153.bn) and Open request signal (SD10681.n, Un\G154.bn, Un\G155.bn) in the program. The following chart shows on/off timings of the Open completion signal and Open request signal.



- When a communication target is connected in the Passive open mode, the CPU module can acquire the IP address and port number of the connected communication target using the SP.SOCCINF instruction.
- 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 the number of 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 Ethernet-equipped 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 Ethernet-equipped 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 state of Ethernet-equipped module side is changed to the wait-for-open state and then retry the connection from the device.
- Do not execute the CLOSE 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 the closed connection, execute the OPEN instruction.

7.2 Communication Using UDP

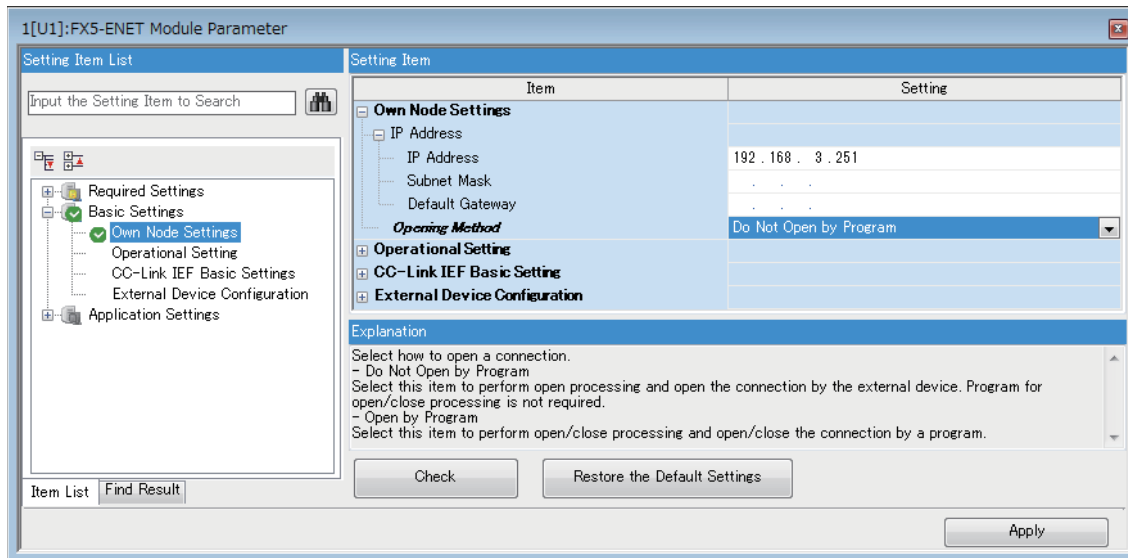
UDP (User Datagram Protocol) is a simple protocol that does not perform data sequencing and retransmission. To perform socket communication using UDP, confirm the following in advance.

- IP address and port number of the target device
- IP address and port number of the Ethernet-equipped module

Open/close processing procedure

The open/close processing procedure for the Ethernet module varies depending on the opening method setting.

Navigation window⇒[Parameter]⇒[Module Information]⇒[FX5-ENET] or [FX5-ENET/IP]⇒[Basic Settings]⇒[Own Node Settings]



■When "Do Not Open by Program" is set

After the Ethernet module mounted station starts up, the UDP/IP communications setting connection automatically opens, and data send/receive is enabled. Program for open/close processing is not required.



When the close processing is executed with a dedicated instruction from the Ethernet module, even if "Do Not Open by Program" is set, the close processing after the disconnection with the external device must be executed by the program.

■When "Open by Program" is set

Before the open/close request is received from the external device, the Ethernet module must execute the GP.OPEN/ GP.CLOSE instruction and enter the open/close standby state. Data can be sent and received after the open processing completes normally.

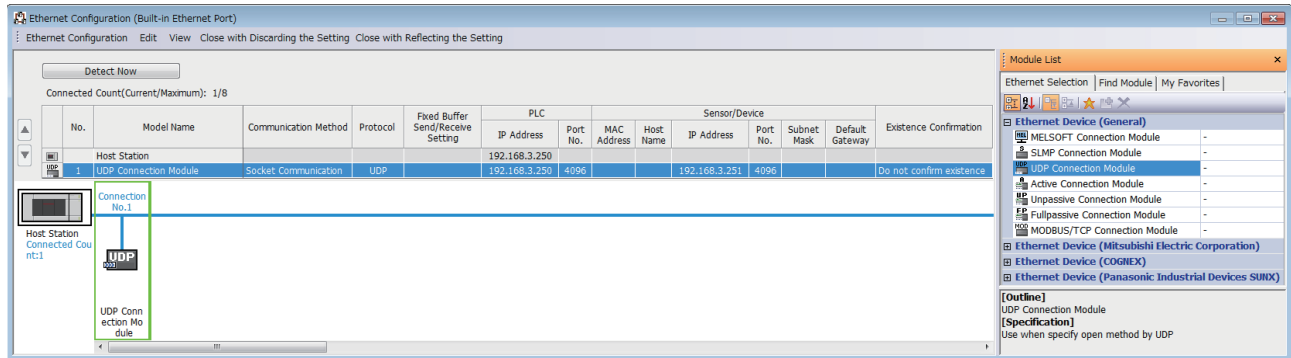
Program example

This section provides a program example for communication using UDP.

Parameter setting

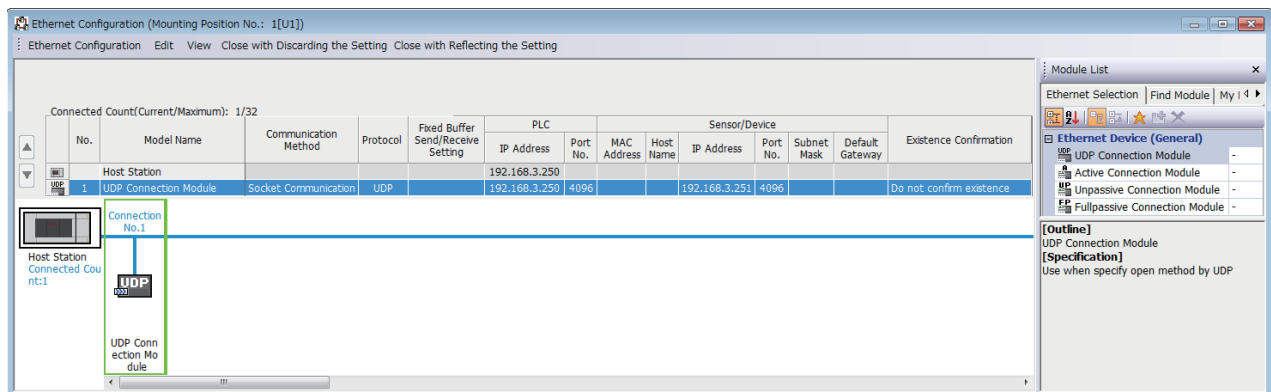
[CPU module]

Navigation window⇒[Parameter]⇒[FX5UJCPU] or [FX5UCPU]⇒[Module Parameter]⇒[Ethernet Port]⇒[Basic Settings]⇒[External Device Configuration]⇒[Detailed Setting]⇒[Ethernet Configuration (Built-in Ethernet Port)] screen



[Ethernet module]

Navigation window⇒[Parameter]⇒[Module Information]⇒[FX5-ENET] or [FX5-ENET/IP]⇒[Basic Settings]⇒[External Device Configuration]⇒[Ethernet Configuration (Mounting Position No.: n[Un])] screen



- Drag and drop the "UDP Connection Equipment" from "Module List" to the left side on the window. Execute the settings as mentioned below.

| Item | | Description |
|---------------|------------|---|
| PLC | Port No. | 4096 (Setting range: 1 to 5548, 5670 to 65534) Do not specify 5549 to 5569 because these ports are used by the system. |
| Sensor/Device | IP Address | 192.168.3.251 (Setting range: 0.0.0.1 to 223.255.255.254) |
| | Port No. | 4096 (Setting range: 1 to 65534) |

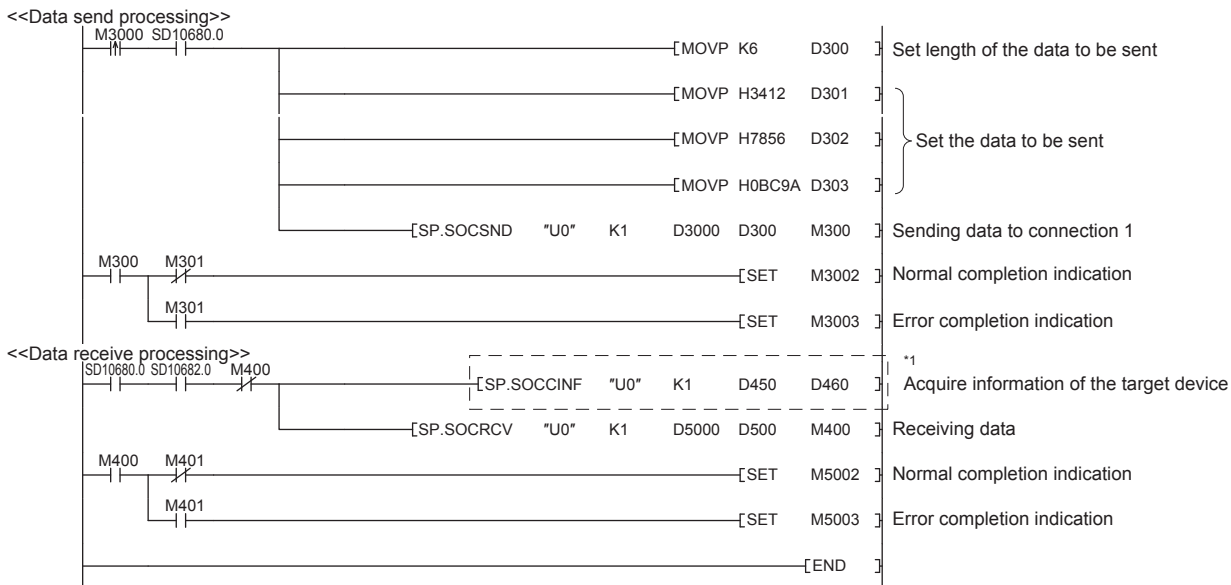
Devices used in the sample program

The following table lists the device numbers used in the sample program and their applications. The devices used when the CPU module is used are shown below.

| Device No. | Application |
|-----------------|--|
| M3000 | Send instruction |
| D3000 and D3001 | SP.SOCSND instruction control data |
| M300 | SP.SOCSND instruction normal completion device |
| M301 | SP.SOCSND instruction error completion device |
| D300 to D303 | Receive data length and send data (6 bytes of 12H, 34H, 56H, 78H, 9AH, BCH) |
| M3002 | Normal send indication |
| M3003 | Send error indication |
| D5000 and D5001 | SP.SOCRCV instruction control data |
| M400 | SP.SOCRCV instruction normal completion device |
| M401 | SP.SOCRCV instruction error completion device |
| SD10680 | Open completion signal |
| SD10682 | Receive state signal |
| M3001 | Target change direction |
| D500 and higher | Receive data length and receive data |
| M5002 | Normal receive indication |
| M5003 | Receive error indication |
| D450 and D451 | SP.SOCCINF instruction control data |
| D460 to D464 | SP.SOCCINF instruction connection information |

Sample program

A program example using the CPU module is shown below.



*1 For acquiring information of the target device connected on UDP, run the program enclosed by the dotted line.
(It can be omitted when the information acquisition is not needed.)

Precautions

■Use of UDP

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

■Sending and receiving data

Data sending process may complete normally 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 procedure at the user's discretion.

■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.

■CLOSE instruction

Do not execute the CLOSE instruction in the program.

Doing so will disable data transfer since the Open completion signal and Open request signal of the corresponding connection turn off for close processing.

To reopen the closed connection, execute the OPEN instruction.

For the OPEN instruction, refer to  Page 105 Opening a connection.

7.3 Precautions

This section provides other precautions for the socket communication function.

Port number

Host station port number, 1 to 1023 (0001H to 03FFH), are assigned for reserved port numbers (WELL KNOWN PORT NUMBERS) and 61440 to 65534 (F000H to FFFEH) are for other communication functions. Therefore, using 1024 to 5548, 5570 to 61439 (0400H to 15ACH, 15C2H to EFFFH) is recommended.

Do not specify 5549 to 5569 (15ADH to 15C1H) because these ports are used by the system.

Do not specify 45237 (B0B5H) and 61440 to 65534 (F000H to FFFEH) for the socket communication function when using the iQ Sensor Solution-compatible function.

When using the following functions, do not specify the port number reserved for the socket communication function.

- File transfer function (FTP server): 20 (14H), 21 (15H)
- Web server function: 80 (50H)*1
- Time setting function (SNTP client): 123 (7BH)
- SLMP function: 61440 (F000H), 61441 (F001H)
- CC-Link IE field network Basic: 61450 (F00AH)

*1 Port No. can be changed. (Default: 80)

Reading received data

Read received data when the Receive state signal (SD10682.n) or Socket communications receive status signal (Un\G156.n to Un\G157.n) has turned on.

Communication via the Ethernet-equipped module may be affected if a considerable amount of received data has not been read for a long time.

Conditions for closing

In TCP communication, 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.

Elements of 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 Ethernet-equipped module
- Port number of the Ethernet-equipped module
- IP address of the target device
- Port number of the target device

Reestablishment of the same connection

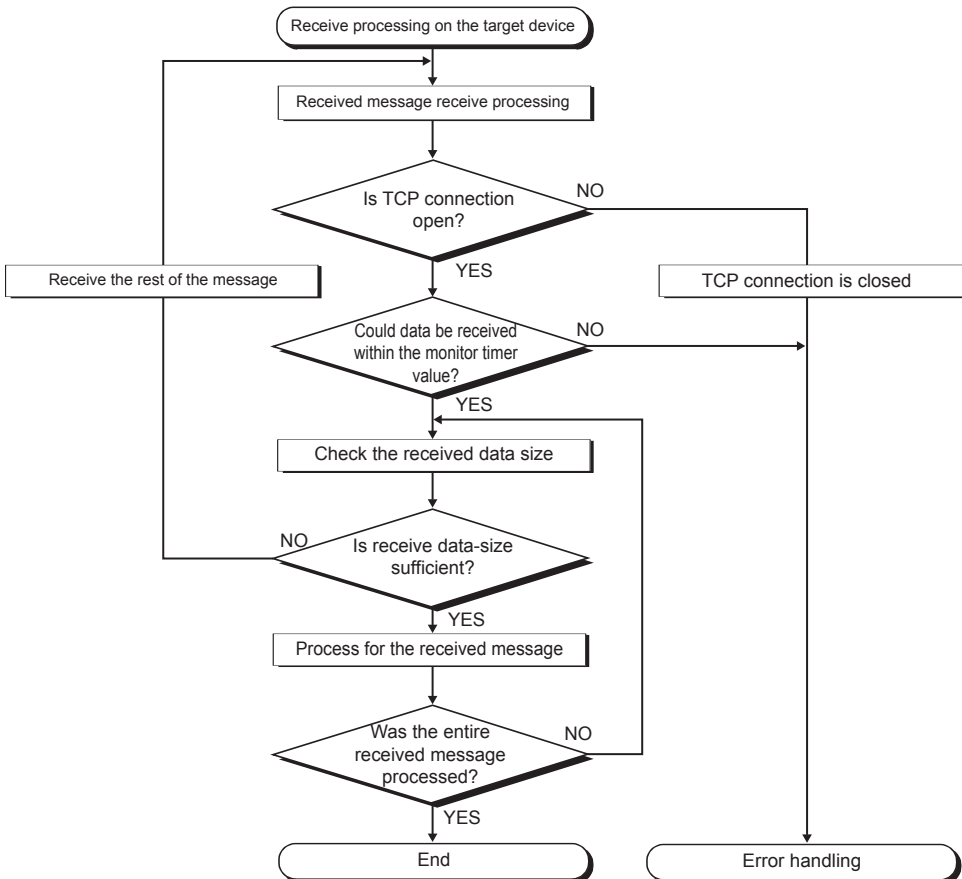
Allow 500 ms or more before reestablishing a connection of the same target IP address, the same host station port number, and the same target port number after closing it.

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

Checking receive data length

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

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



7.4 Socket Communication Function Instructions

The socket communication function instructions are provided for the Ethernet-equipped module to use the socket communication function.

This section explains the socket communication function instructions.

The following is a list of the instructions.





CPU module dedicated instruction

| Instruction | Description | Reference |
|--------------|---|-----------------------|
| SP.SOCOPEN | Establishes a connection. | Page 105 SP.SOCOPEN |
| SP.SOCCLOSE | Closes a connection. | Page 113 SP.SOCCLOSE |
| SP.SOCRCV | Reads the data received (Read at END processing). | Page 119 SP.SOCRCV |
| SP.SOCSND | Sends data. | Page 126 SP.SOCSND |
| SP.SOCCINF | Reads connection information. | Page 133 SP.SOCCINF |
| S(P).SOCDATA | Reads data from the socket communication receive data area. | Page 135 S(P).SOCDATA |

Ethernet module dedicated instruction

| Instruction | Description | Reference |
|-------------|---------------------------|--------------------|
| GP.OPEN | Establishes a connection. | Page 109 GP.OPEN |
| GP.CLOSE | Closes a connection. | Page 116 GP.CLOSE |
| GP.SOCRCV | Reads the data received. | Page 123 GP.SOCRCV |
| GP.SOCSND | Sends data. | Page 130 GP.SOCSND |

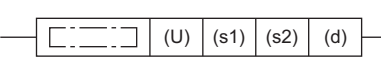
Point

- For configuration of data communication using the socket communication function, refer to  Page 91 Communication Using TCP and  Page 99 Communication Using UDP.
- 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  Page 245 Error Codes or  MELSEC iQ-F FX5 User's Manual (Application).

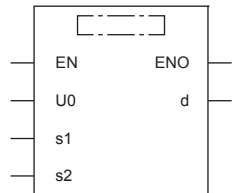
Opening a connection

SP.SOCOPEN

Establishes a connection. (CPU module dedicated instruction)

| Ladder diagram | Structured text |
|---|--|
|  | <pre>ENO:=SP_SOCOPEN(EN,U0,s1,s2,d);</pre> |


FBD/LD



("SP_SOCOPEN" enters □.)

Setting data

■ Descriptions, ranges, and data types

| Operand | Description | Range | Data type | Data type (Label) |
|-------------------|--|--|------------------------|---|
| (U) ^{*1} | Dummy (Input the character string ['U0'].) | — | Character string | __*2 (ANYSTRING_SINGLE) |
| (s1) | Connection number | 1 to 8 | 16-bit unsigned binary | ANY16 |
| (s2) | Start number of the device in which control data is stored | Refer to Control data ( Page 106) | Word | ANY16_ARRAY (Number of elements: 10) |
| (d) | Start number of the device which turns on for one scan upon completion of the instruction. (d)+1 also turns on when failed. | — | Bit | ANYBIT_ARRAY (Number of elements: 2) |
| EN | Execution condition | — | Bit | BOOL |
| ENO | Execution result | — | Bit | BOOL |

*1 In the case of the ST language and the FBD/LD language, U displays as U0.

*2 Regardless of the program language to be used, the data type is specified by a device. Do not specify a label.

■ Applicable devices

| Operand | Bit | Word | | | Double word | | Indirect specification | Constant | | | Others |
|---------|-----------------------------|---------------------------|-------|---|-------------|----|------------------------|----------|---|----|--------|
| | X, Y, M, L, SM, F, B, SB, S | T, ST, C, D, W, SD, SW, R | U□\G□ | Z | LC | LZ | | K, H | E | \$ | |
| (U) | — | — | — | — | — | — | — | — | — | ○ | — |
| (s1) | — | ○ | — | — | — | — | ○ | ○ | — | — | — |
| (s2) | — | ○ | — | — | — | — | ○ | — | — | — | — |
| (d) | ○ | ○*1 | — | — | — | — | — | — | — | — | — |

*1 T, ST, C cannot be used.

Control data

| Device | Item | Description | Setting range | Set by ^{*1} |
|---------------------|---|---|---|----------------------|
| (s2)+0 | Execution/completion type | Specify which settings are used to open a connection, parameter settings configured by an engineering tool or control data settings (s2) +2 to (s2) +6. 0000H: Connection is opened according to the settings set in "External Device Configuration" of module parameter. 8000H: Connection is opened according to the values specified for control data (s2) +2 to (s2) +6. | 0000H 8000H | User |
| (s2)+1 | Completion status | Completion status is stored 0000H: Completed Other than 0000H: Failed (Error code) Refer to Page 245 Error Codes. | — | System |
| (s2)+2 | Application setting area | <div style="text-align: center;"> $\begin{array}{cccccccc} b15 & b14 & b13 & \text{to} & b11 & b10 & b9 & b8 & b7 & & & \text{to} & & & & & b0 \\ (s2)+2 & [4] & 0 & [3] & [2] & [1] & & & & & & & & & & & 0 \end{array}$ </div> [1] Communication method (protocol) 0: TCP/IP 1: UDP/IP [2] Socket communications function procedure 0: Communication protocol 1: Socket communications (No procedure) [3] Communication protocol setting 0: Do not use the communication protocol support function (use the socket communications function) 1: Use the protocol support function [4] Open method 00: Active open or UDP/IP 10: Unpassive open 11: Fullpassive open | Shown on left side | User |
| (s2)+3 | Host Station Port No. | Specify the port number of the host station. | 1 to 5548, 5570 to 65534 (0001H to 15ACH, 15C2H to FFFE ^H) ^{*3} | |
| (s2)+4 (s2)+5 | Target device IP address ^{*2} | Specify the IP address of the target device. | 1 to 3758096382 (00000001H to DFFFFFFEH) | |
| (s2)+6 | Target device port number ^{*2} | Specify the port number of the target device. | 1 to 65534 (0001H to FFFE ^H) | |
| (s2)+7 to (s2)+9 | — | Use prohibited | — | System |

*1 User: Data to be set before the execution of the instruction. System: The CPU module stores the execution result of the instruction.

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

*3 Because host station port numbers, 1 to 1023 (0001H to 03FFH), are assigned for reserved port numbers and 61440 to 65534 (F000H to FFFE^H) are used for other communication functions, using 1024 to 5548, 5570 to 61439 (0400H to 15ACH, 15C2H to EFFFH) is recommended. Do not specify 5549 to 5569 (15ADH to 15C1H) because these ports are used by the system.

Processing details

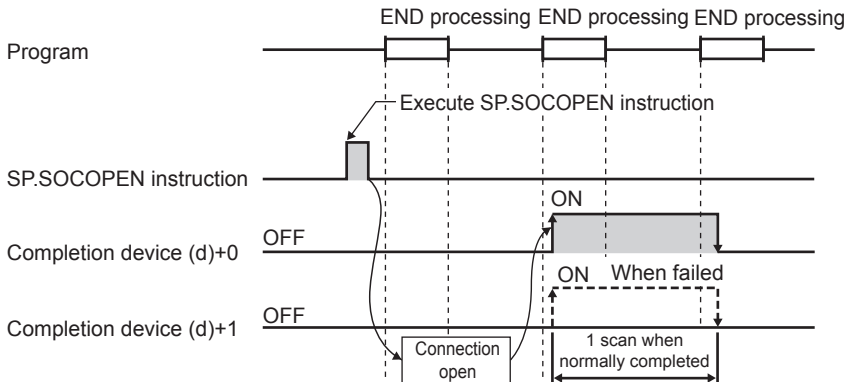
This instruction opens a connection specified in (s1).

The set values used for the open processing is selected in (s2)+0.

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

- Completion device (d)+0: Turns on in the END processing of the scan after completion of the SP.SOCOPEN instruction, and turns off in the next END processing.
- Completion device (d)+1: Turns on or off according to the status at the time of completion of the SP.SOCOPEN instruction.

| Status | Description |
|----------------|--|
| When completed | Remains off. |
| When failed | Turns on in the END processing of the scan after completion of the SP.SOCOPEN instruction, and turns off in the next END processing. |



- The connection in which no protocol is set with the parameter can be opened and used. In this case, specify 8000H in (s2)+0 and the contents of the open processing in (s2)+2 to (s2)+6.

Operation error

| Error code (SD0/SD8067) | Description |
|-------------------------|---|
| 3405H | The connection number specified by (s1) is other than 1 to 8. |
| 2820H | The device number specified by (s2) or (d) is outside the range of the number of device points. |
| 2822H | Device that cannot be specified is specified. |
| 3582H | When an instruction which cannot be used in interruption routine program is used. |

Program example

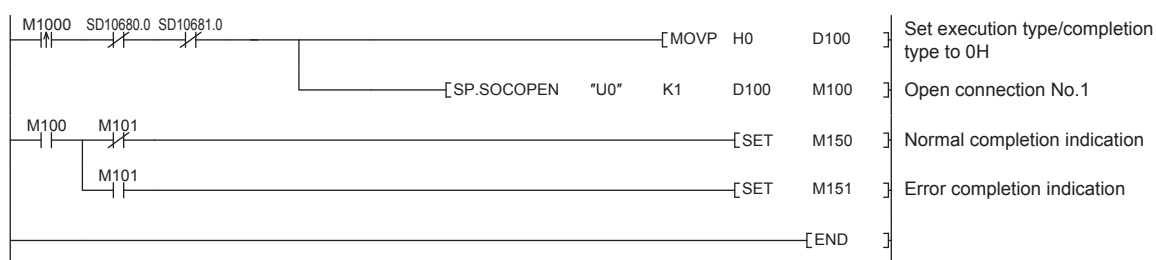
■Opening a connection using parameter settings

When M1000 is turned on, connection No.1 is opened using the parameters set in "External Device Configuration" of module parameter.

- Devices used

| Device No. | Application |
|------------|--|
| SD10680 | Open completion signal |
| SD10681 | Open request signal |
| D100 | SP.SOCOPEN instruction control data |
| M100 | SP.SOCOPEN instruction completion device |
| M101 | SP.SOCOPEN instruction error completion device |
| M150 | Normal completion indication |
| M151 | Error completion indication |
| M1000 | SP.SOCOPEN instruction drive flag |

- Program



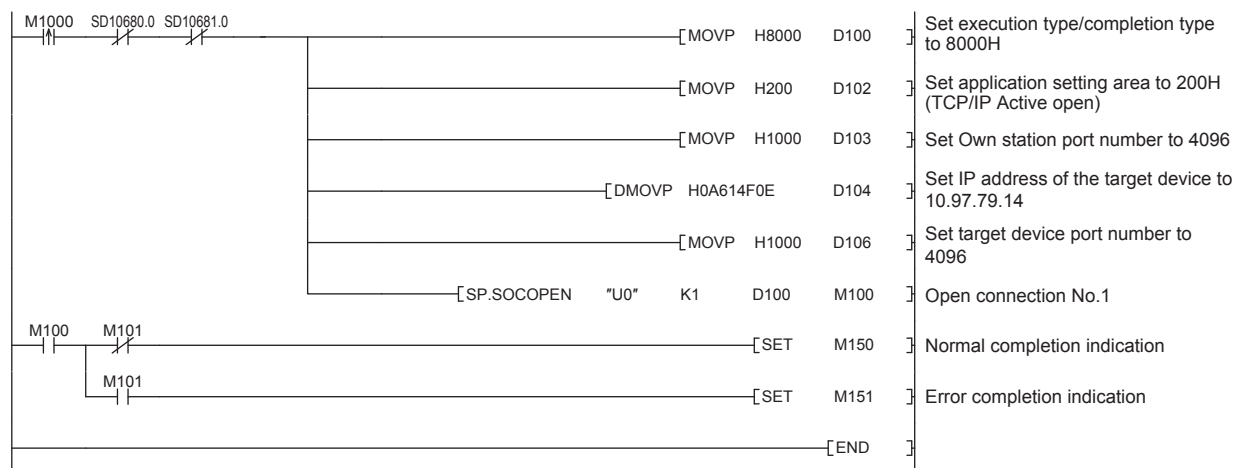
■Opening a connection using control data settings

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

- Devices used

| Device No. | Application |
|--------------------|--|
| SD10680 | Open completion signal |
| SD10681 | Open request signal |
| D100, D102 to D106 | SP.SOCOPEN instruction control data |
| M100 | SP.SOCOPEN instruction completion device |
| M101 | SP.SOCOPEN instruction error completion device |
| M150 | Normal completion indication |
| M151 | Error completion indication |
| M1000 | SP.SOCOPEN instruction drive flag |

- Program

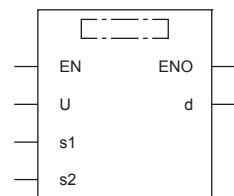


GP.OPEN

Establishes a connection. (Ethernet module dedicated instruction)

| Ladder diagram | Structured text |
|----------------|--|
| | <pre>ENO:=GP_OPEN(EN,U,s1,s2,d);</pre> |

FBD/LD



("GP_OPEN" enters □.)

Setting data

■ Descriptions, ranges, and data types


| Operand | Description | Range | Data type | Data type (Label) |
|---------|---|--|------------------------|---|
| (U) | Position number of the module connected | <ul style="list-style-type: none"> ■ FX5UJ CPU module 1H to 8H ■ FX5U/FX5UC CPU module 1H to 10H | 16-bit unsigned binary | ANY16 |
| (s1) | Connection number | 1 to 32 | 16-bit unsigned binary | ANY16 |
| (s2) | Start number of the device in which control data is stored | Refer to Control data (Page 110) | Word | ANY16_ARRAY (Number of elements: 10) |
| (d) | Start number of the device which turns on for one scan upon completion of the instruction. (d)+1 also turns on when failed. | — | Bit | ANYBIT_ARRAY (Number of elements: 2) |
| EN | Execution condition | — | Bit | BOOL |
| ENO | Execution result | — | Bit | BOOL |

■ Applicable devices

| Operand | Bit | Word | | | Double word | | Indirect specification | Constant | | | Others (U) |
|---------|-----------------------------|---------------------------|-------|---|-------------|----|------------------------|----------|---|----|------------|
| | X, Y, M, L, SM, F, B, SB, S | T, ST, C, D, W, SD, SW, R | U□\G□ | Z | LC | LZ | | K, H | E | \$ | |
| (U) | — | ○ | — | — | — | — | ○ | ○ | — | — | ○ |
| (s1) | ○ | ○ | — | — | — | — | ○ | ○ | — | — | — |
| (s2) | — | ○ | — | — | — | — | ○ | — | — | — | — |
| (d) | ○ | ○*1 | — | — | — | — | — | — | — | — | — |

*1 T, ST, C cannot be used.

■Control data

| Device | Item | Description | Setting range | Set by ^{*1} |
|---------------------|---|--|---|----------------------|
| (s2)+0 | Execution/completion type | Specify which settings are used to open a connection, parameter settings configured by an engineering tool or control data settings (s2) +2 to (s2) +6. 0000H: Connection is opened according to the settings set in "External Device Configuration" of module parameter. 8000H: Connection is opened according to the values specified for control data (s2) +2 to (s2) +6. | 0000H 8000H | User |
| (s2)+1 | Completion status | Completion status is stored 0000H: Completed Other than 0000H: Failed (Error code) Refer to  Page 245 Error Codes. | — | System |
| (s2)+2 | Application setting area | <div style="text-align: center;"> $\begin{array}{ccccccc} \text{b15b14 b13} & \text{to} & \text{b9 b8 b7} & \text{to} & \text{b0} \\ \text{(s2)+2} & \boxed{[2]} & \boxed{0} & \boxed{[1]} & \boxed{0} \end{array}$ </div> <p>[1] Communication method (protocol) 0: TCP/IP 1: UDP/IP [2] Open method 00: Active open or UDP/IP 10: Unpassive open 11: Fullpassive open</p> | Shown on left side | User |
| (s2)+3 | Host Station Port No. | Specify the port number of the host station. | 1 to 5548, 5570 to 65534 (0001H to 15ACH, 15C2H to FFFE ^H) ^{*3} | |
| (s2)+4 (s2)+5 | Target device IP address ^{*2} | Specify the IP address of the target device. | 1 to 3758096382 (00000001H to DFFFFFFEH) | |
| (s2)+6 | Target device port number ^{*2} | Specify the port number of the target device. | 1 to 65534 (0001H to FFFE ^H) | |
| (s2)+7 to (s2)+9 | — | Use prohibited | — | System |

*1 User: Data to be set before the execution of the instruction. System: The Ethernet module stores the execution result of the instruction.

*2 When Unpassive open is selected, the target device IP address and target device port number are ignored.

*3 Of the host station port numbers, 1 to 1023 (0001H to 03FFH) are generally reserved port numbers and 61440 to 65534 (F000H to FFFE^H) are used by other communication functions. Thus, using 1024 to 5548 and 5570 to 61439 (0400H to 15ACH and 15C2H to EFFFH) as the port numbers is recommended. Do not specify 5549 to 5569 (15ADH to 15C1H) since they are used by the system.

Processing details

This instruction opens a connection specified in (s1).

The set values used for the open processing is selected in (s2)+0.

The result of the GP.OPEN instruction can be checked with the completion device, (d)+0 and (d)+1.

- Completion device (d)+0: Turns on in the END processing of the scan after completion of the GP.OPEN instruction, and turns off in the next END processing.
- Completion device (d)+1: Turns on or off according to the status at the time of completion of the GP.OPEN instruction.

| Status | Description |
|----------------|---|
| When completed | Remains off. |
| When failed | Turns on in the END processing of the scan after completion of the GP.OPEN instruction, and turns off in the next END processing. |

- The connection in which no protocol is set with the parameter can be opened and used. In this case, specify 8000H in (s2)+0 and the contents of the open processing in (s2)+2 to (s2)+6.

Operation error

| Error code ((s2)+1) | Description |
|---------------------|--|
| 1D80H | An instruction other than executable dedicated instruction numbers has been specified. |
| 1D83H | The dedicated instruction request data from the CPU module was discarded because the data size was abnormal. |
| 1D84H | The dedicated instruction request data from the CPU module cannot be normally input. |
| 1D85H | A timeout error occurred in the dedicated instruction response data, and the data was discarded. |
| C029H | Description of control data is not correct. |
| C1A6H | A connection No. other than 1 to 32 has been specified. |

Program example

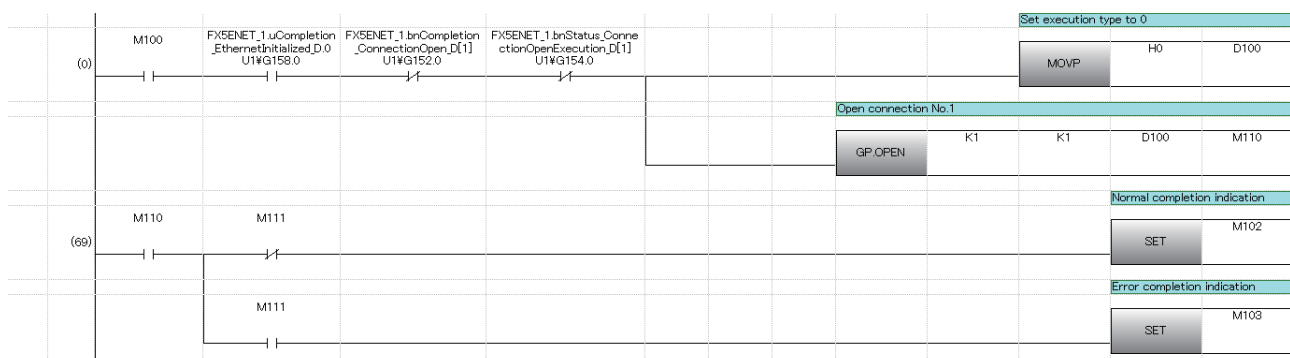
■ Opening a connection using parameter settings

When M100 is turned on, connection No.1 is opened using the parameters set in "External Device Configuration" of module parameter.

- Devices used

| Device No. | Application |
|------------------|---|
| Un\G152, Un\G153 | Open completion signal |
| Un\G154, Un\G155 | Open request signal |
| Un\G158.b0 | Initial normal completion status |
| D100 | GP.OPEN instruction control data |
| M100 | GP.OPEN instruction drive flag |
| M102 | Normal completion indication |
| M103 | Error completion indication |
| M110 | GP.OPEN instruction completion device |
| M111 | GP.OPEN instruction error completion device |

- Program



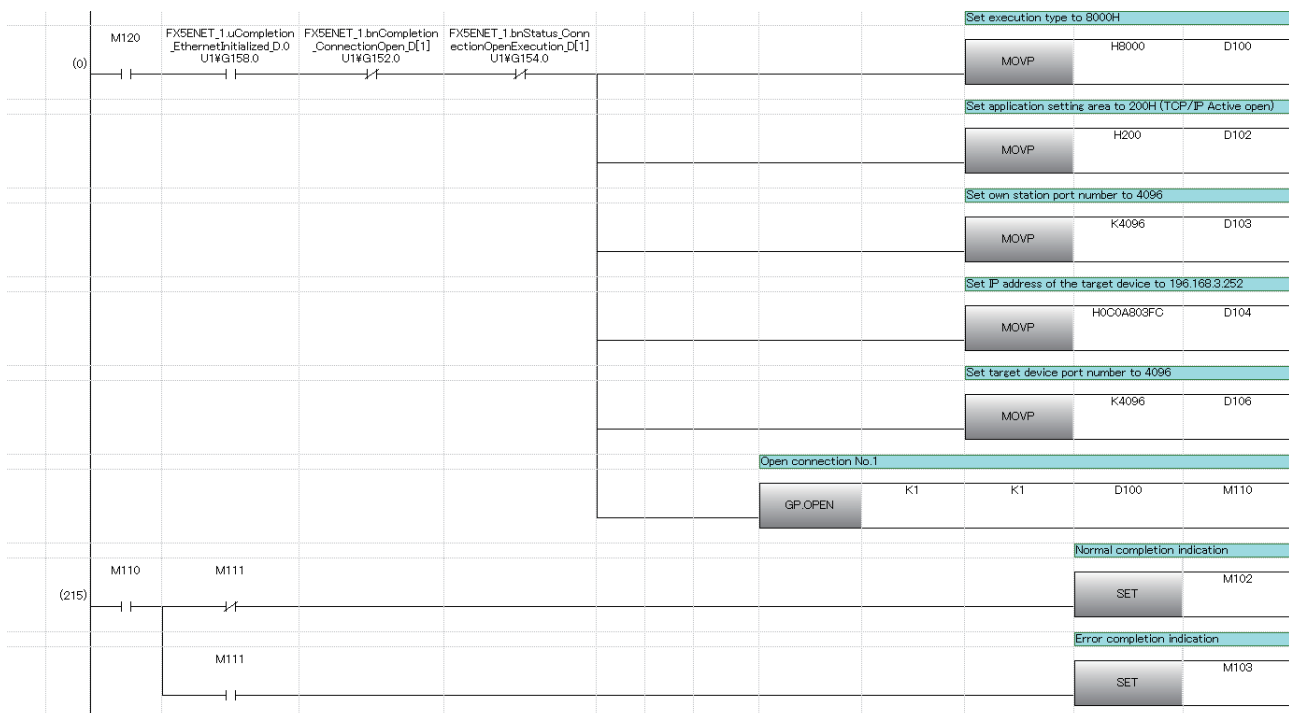
■Opening a connection using control data settings

When M120 is turned on, connection No.1 is opened using control data.

• Devices used

| Device No. | Application |
|--------------------|---|
| Un\G152, Un\G153 | Open completion signal |
| Un\G154, Un\G155 | Open request signal |
| Un\G158.b0 | Initial normal completion status |
| D100, D102 to D106 | GP.OPEN instruction control data |
| M102 | Normal completion indication |
| M103 | Error completion indication |
| M110 | GP.OPEN instruction completion device |
| M111 | GP.OPEN instruction error completion device |
| M120 | GP.OPEN instruction drive flag |

• Program



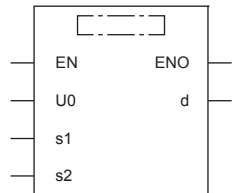
Disconnecting a connection

SP.SOCCLOSE

Closes a connection. (CPU module dedicated instruction)

| Ladder diagram | Structured text |
|----------------|---|
| | <pre>ENO:=SP_SOCCLOSE(EN,U0,s1,s2,d);</pre> |

FBD/LD



("SP_SOCCLOSE" enters □.)

Setting data

■ Descriptions, ranges, and data types

| Operand | Description | Range | Data type | Data type (Label) |
|-------------------|---|--------------------------------------|------------------------|---|
| (U) ^{*1} | Dummy (Input the character string ['U0'].) | — | Character string | __ ^{*2} (ANYSTRING_SINGLE) |
| (s1) | Connection No. | 1 to 8 | 16-bit unsigned binary | ANY16 |
| (s2) | Start number of the device in which control data is stored | Refer to Control data (Page 113) | Word | ANY16_ARRAY (Number of elements: 2) |
| (d) | Start number of the device which turns on for one scan upon completion of the instruction. (d)+1 also turns on when failed. | — | Bit | ANYBIT_ARRAY (Number of elements: 2) |
| EN | Execution condition | — | Bit | BOOL |
| ENO | Execution result | — | Bit | BOOL |

*1 In the case of the ST language and the FBD/LD language, U displays as U0.

*2 Regardless of the program language to be used, the data type is specified by a device. Do not specify a label.

■ Applicable devices

| Operand | Bit | Word | | | Double word | | Indirect specification | Constant | | | Others |
|---------|-----------------------------|---------------------------|-------|---|-------------|----|------------------------|----------|---|----|--------|
| | X, Y, M, L, SM, F, B, SB, S | T, ST, C, D, W, SD, SW, R | U□\G□ | Z | LC | LZ | | K, H | E | \$ | |
| (U) | — | — | — | — | — | — | — | — | — | ○ | — |
| (s1) | — | ○ | — | — | — | — | ○ | ○ | — | — | — |
| (s2) | — | ○ | — | — | — | — | ○ | — | — | — | — |
| (d) | ○ | ○ ^{*1} | — | — | — | — | — | — | — | — | — |

*1 T, ST, C cannot be used.

■ Control data

| Device | Item | Description | Setting range | Set by ^{*1} |
|--------|-------------------|--|---------------|----------------------|
| (s2)+0 | System area | — | — | — |
| (s2)+1 | Completion status | Completion status is stored 0000H: Completed Other than 0000H: Failed (Error code) Refer to Page 245 Error Codes | — | System |

*1 System: The CPU module stores the execution result of the instruction.

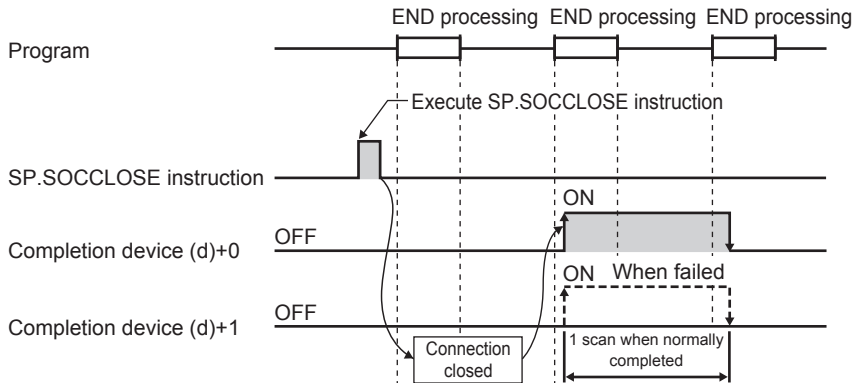
Processing details

This instruction closes a connection specified in (s1). (Disconnecting a connection)

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

- Completion device (d)+0: Turns on in the END processing of a scan after completion of the SP.SOCCLOSE instruction, and turns off in the next END processing.
- Completion device (d)+1: Turns on or off according to the status at the time of completion of the SP.SOCCLOSE instruction.

| Status | 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. |



Operation error

| Error code (SD0/SD8067) | Description |
|-------------------------|---|
| 3405H | The connection number specified by (s1) is other than 1 to 8. |
| 2820H | The device number specified by (s2) or (d) is outside the range of the number of device points. |
| 2822H | Device that cannot be specified is specified. |
| 3582H | When an instruction which cannot be used in interruption routine program is used. |

Point

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

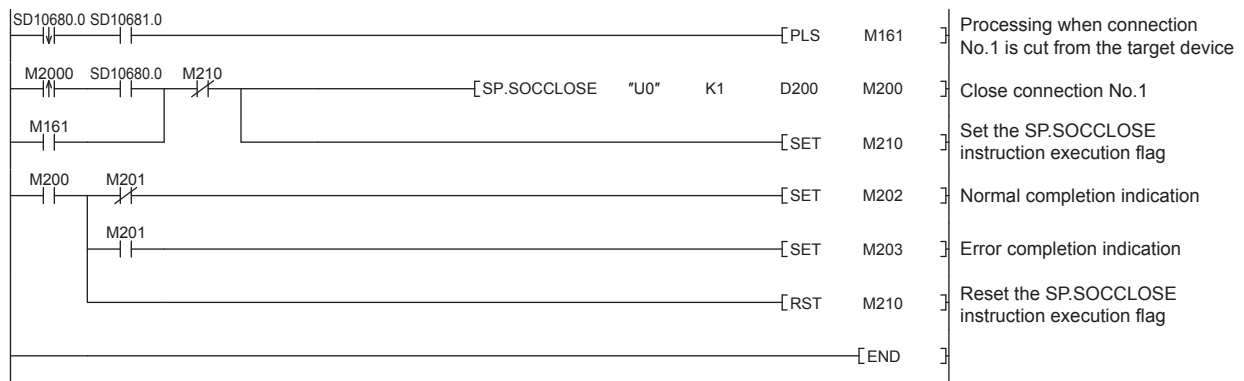
Program example

When M2000 is turned on, or when connection No.1 is disconnected from target device, this program disconnects connection No.1.

• Devices used

| Device No. | Application |
|------------|---|
| SD10680 | Open completion signal |
| SD10681 | Open request signal |
| D200 | SP.SOCCLOSE instruction control data |
| M161 | Connection No.1 cutting flag |
| M200 | SP.SOCCLOSE instruction completion device |
| M201 | SP.SOCCLOSE instruction error completion device |
| M202 | Normal completion indication |
| M203 | Error completion indication |
| M210 | SP.SOCCLOSE instruction execution flag |
| M2000 | SP.SOCCLOSE instruction drive flag |

• Program

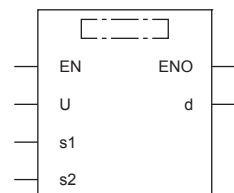


GP.CLOSE

Closes a connection. (Ethernet module dedicated instruction)

| Ladder diagram | Structured text |
|----------------|---|
| | <pre>ENO:=GP_CLOSE(EN,U,s1,s2,d);</pre> |

FBD/LD



("GP_CLOSE" enters □.)

Setting data

■ Descriptions, ranges, and data types

| Operand | Description | Range | Data type | Data type (Label) |
|---------|---|--|------------------------|---|
| (U) | Position number of the module connected | <ul style="list-style-type: none"> ■ FX5UJ CPU module 1H to 8H ■ FX5U/FX5UC CPU module 1H to 10H | 16-bit unsigned binary | ANY16 |
| (s1) | Connection No. | 1 to 32 | 16-bit unsigned binary | ANY16 |
| (s2) | Start number of the device in which control data is stored | Refer to Control data (Page 116) | Word | ANY16_ARRAY (Number of elements: 2) |
| (d) | Start number of the device which turns on for one scan upon completion of the instruction. (d)+1 also turns on when failed. | — | Bit | ANYBIT_ARRAY (Number of elements: 2) |
| EN | Execution condition | — | Bit | BOOL |
| ENO | Execution result | — | Bit | BOOL |

■ Applicable devices

| Operand | Bit | Word | | | Double word | | Indirect specification | Constant | | | Others (U) |
|---------|-----------------------------|---------------------------|-------|---|-------------|----|------------------------|----------|---|----|------------|
| | X, Y, M, L, SM, F, B, SB, S | T, ST, C, D, W, SD, SW, R | U□\G□ | Z | LC | LZ | | K, H | E | \$ | |
| (U) | — | ○ | — | — | — | — | ○ | ○ | — | — | ○ |
| (s1) | ○ | ○ | — | — | — | — | ○ | ○ | — | — | — |
| (s2) | — | ○ | — | — | — | — | ○ | — | — | — | — |
| (d) | ○ | ○*1 | — | — | — | — | — | — | — | — | — |

*1 T, ST, C cannot be used.

■ Control data

| Device | Item | Description | Setting range | Set by*1 |
|--------|-------------------|---|---------------|----------|
| (s2)+0 | System area | — | — | — |
| (s2)+1 | Completion status | Completion status is stored 0000H: Completed Other than 0000H: Failed (Error code) Refer to Page 245 Error Codes | — | System |

*1 System: The Ethernet module stores the execution result of the instruction.

Processing details

This instruction closes a connection specified in (s1). (Disconnecting a connection)

The result of the GP.CLOSE instruction can be checked with the completion device, (d)+0 and (d)+1.

- Completion device (d)+0: Turns on in the END processing of a scan after completion of the GP.CLOSE instruction, and turns off in the next END processing.
- Completion device (d)+1: Turns on or off according to the status at the time of completion of the GP.CLOSE instruction.

| Status | Description |
|----------------|--|
| When completed | Remains off. |
| When failed | Turns on in the END processing of a scan after completion of the GP.CLOSE instruction, and turns off in the next END processing. |

Operation error

| Error code ((s2)+1) | Description |
|---------------------|--|
| 1D80H | An instruction other than executable dedicated instruction numbers has been specified. |
| 1D83H | The dedicated instruction request data from the CPU module was discarded because the data size was abnormal. |
| 1D84H | The dedicated instruction request data from the CPU module cannot be normally input. |
| 1D85H | A timeout error occurred in the dedicated instruction response data, and the data was discarded. |
| C029H | Description of control data is not correct. |
| C1A6H | A connection No. other than 1 to 32 has been specified. |

Point

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

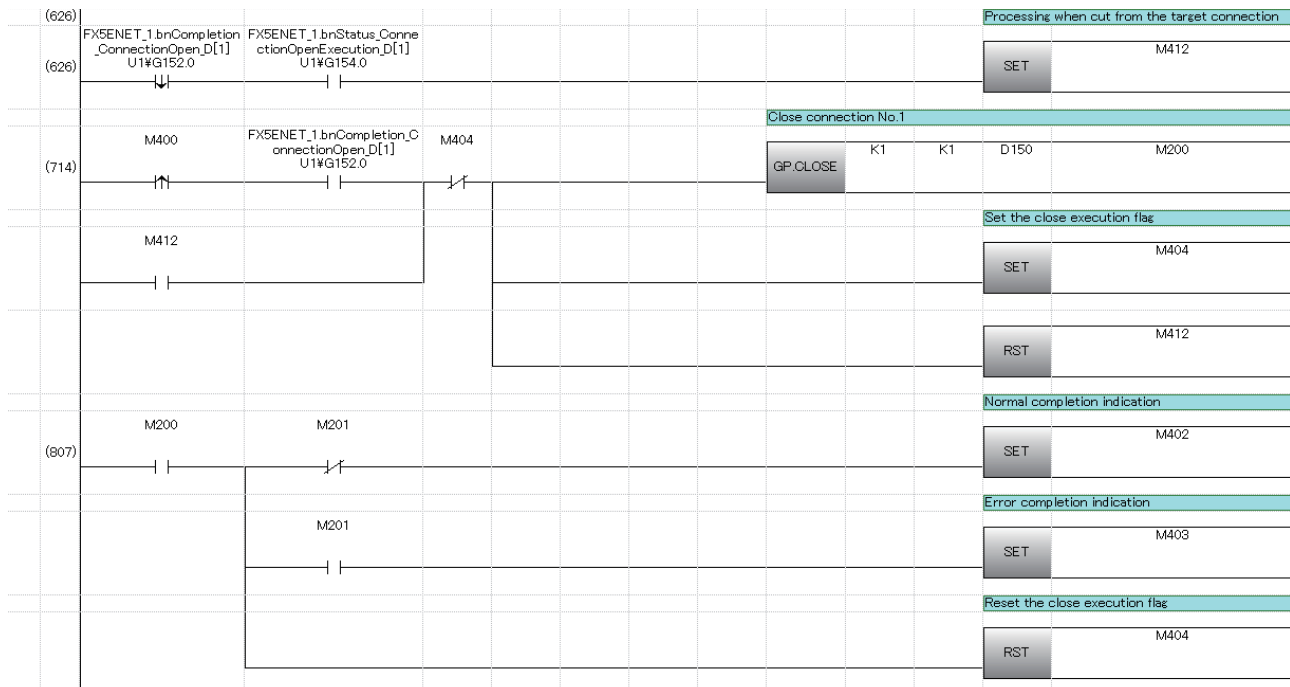
Program example

When M400 is turned on, or when connection No.1 is disconnected from target device, this program disconnects connection No.1.

• Devices used

| Device No. | Application |
|------------------|--|
| Un\G152, Un\G153 | Open completion signal |
| Un\G154, Un\G155 | Open request signal |
| D150 | GP.CLOSE instruction control data |
| M200 | GP.CLOSE instruction completion device |
| M201 | GP.CLOSE instruction error completion device |
| M400 | GP.CLOSE instruction drive flag |
| M402 | Normal completion indication |
| M403 | Error completion indication |
| M404 | GP.CLOSE instruction execution flag |
| M412 | Connection No.1 cutting flag |

• Program



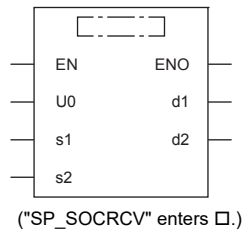
Reading received data in the END processing

SP.SOCRCV

Reads the data received. (Read at END processing) (CPU module dedicated instruction)

| Ladder diagram | Structured text |
|----------------|---|
| | <pre>ENO:=SP_SOCRCV(EN,U0,s1,s2,d1,d2);</pre> |

FBD/LD



Setting data

■ Descriptions, ranges, and data types

| Operand | Description | Range | Data type | Data type (Label) |
|---------|--|--------------------------------------|------------------------|---|
| (U)*1 | Dummy (Input the character string ['U0'].) | — | Character string | _*2 (ANYSTRING_SINGLE) |
| (s1) | Connection No. | 1 to 8 | 16-bit unsigned binary | ANY16 |
| (s2) | Start number of the device where control data is specified | Refer to Control data (Page 120) | Word | ANY16_ARRAY (Number of elements: 2) |
| (d1) | Start number of the device in which received data is stored | — | Word | ANY16 |
| (d2) | Start number of the device which turns on for one scan upon completion of the instruction. (d2)+1 also turns on when failed. | — | Bit | ANYBIT_ARRAY (Number of elements: 2) |
| EN | Execution condition | — | Bit | BOOL |
| ENO | Execution result | — | Bit | BOOL |

*1 In the case of the ST language and the FBD/LD language, U displays as U0.


*2 Regardless of the program language to be used, the data type is specified by a device. Do not specify a label.

■ Applicable devices

| Operand | Bit | Word | | | Double word | | Indirect specification | Constant | | | Others |
|---------|-----------------------------|---------------------------|-------|---|-------------|----|------------------------|----------|---|----|--------|
| | X, Y, M, L, SM, F, B, SB, S | T, ST, C, D, W, SD, SW, R | U□\G□ | Z | LC | LZ | | K, H | E | \$ | |
| (U) | — | — | — | — | — | — | — | — | — | ○ | — |
| (s1) | — | ○ | — | — | — | — | ○ | ○ | — | — | — |
| (s2) | — | ○ | — | — | — | — | ○ | — | — | — | — |
| (d1) | — | ○ | — | — | — | — | ○ | — | — | — | — |
| (d2) | ○ | ○*1 | — | — | — | — | — | — | — | — | — |

*1 T, ST, C cannot be used.

■Control data

| Device | Item | Description | Setting range | Set by ^{*1} |
|------------------|----------------------|---|---------------|----------------------|
| (s2)+0 | System area | — | — | — |
| (s2)+1 | Completion status | Completion status is stored 0000H: Completed Other than 0000H: Failed (Error code) Refer to  Page 245 Error Codes | — | System |
| (d1)+0 | Received data length | The length of the data which was read from the Socket communication receiving data area is stored. (in bytes) | 0 to 2046 | System |
| (d1)+1 to (d1)+n | Received data | The data which was read from the Socket communication receiving data area is stored in order. | — | System |

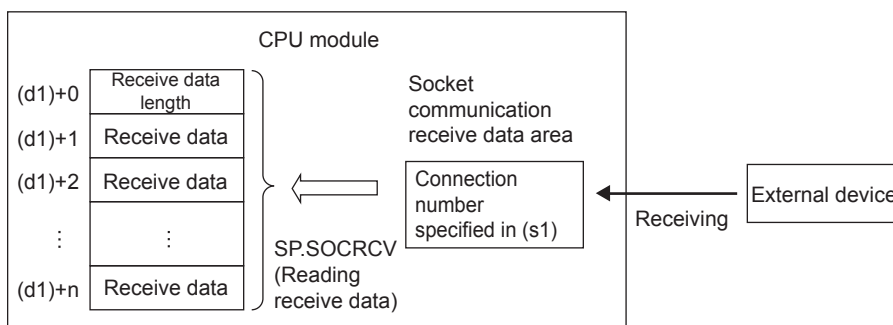
*1 System: The CPU module stores the execution result of the instruction.

Point

- When the SP.SOCRCV instruction is executed, data is read from socket communication receiving data area at END processing. Therefore, executing the SP.SOCRCV instruction will increase the scan time.
- When odd-byte data is received, an invalid byte is stored to the higher byte of the device that stores the last received data.

Processing details

This instruction reads received data of the connection specified in (s1) from the socket communication receive data area in the END processing after execution of the SP.SOCRCV instruction.

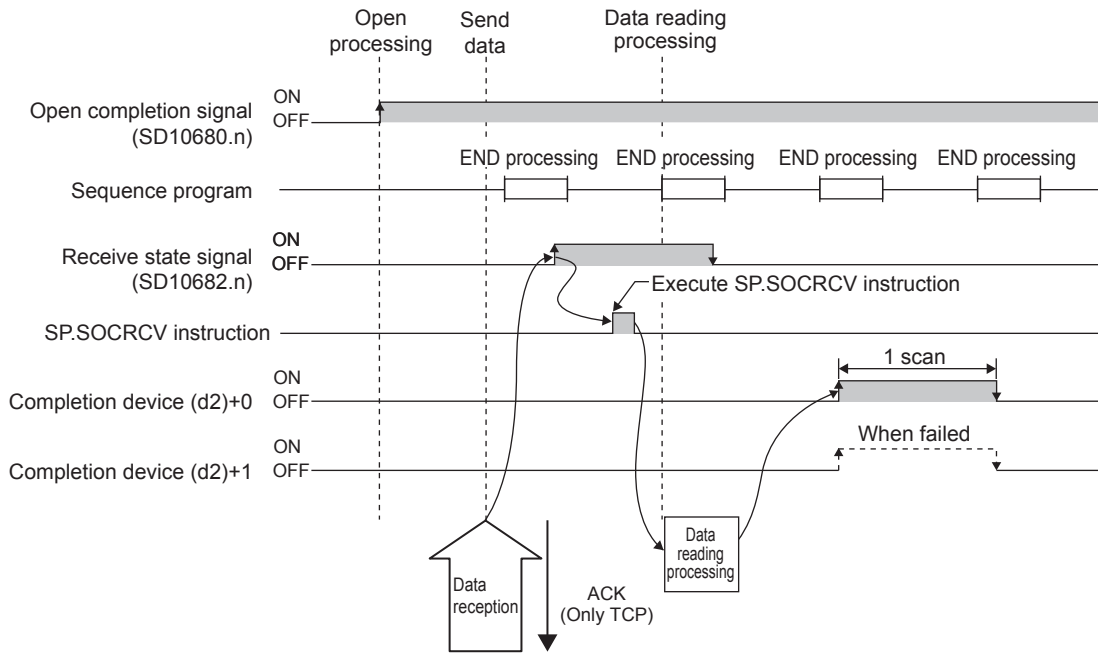


The result of the SP.SOCRCV instruction can be checked with the completion device (d2)+0 and (d2)+1.

- Completion device (d2)+0: Turns on in the END processing of the scan after completion of the SP.SOCRCV instruction, and turns off in the next END processing.
- Completion device (d2)+1: Turns on or off according to the status at the time of completion of the SP.SOCRCV instruction.

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

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



Operation error

| Error code (SD0/SD8067) | Description |
|-------------------------|---|
| 3405H | The connection number specified by (s1) is other than 1 to 8. |
| 2820H | The size of the receive data exceeds the size of the receive data storage device. The device number specified by (s2), (d1) or (d2) is outside the range of the number of device points. |
| 2822H | Device that cannot be specified is specified. |
| 3582H | When an instruction which cannot be used in interruption routine program is used. |

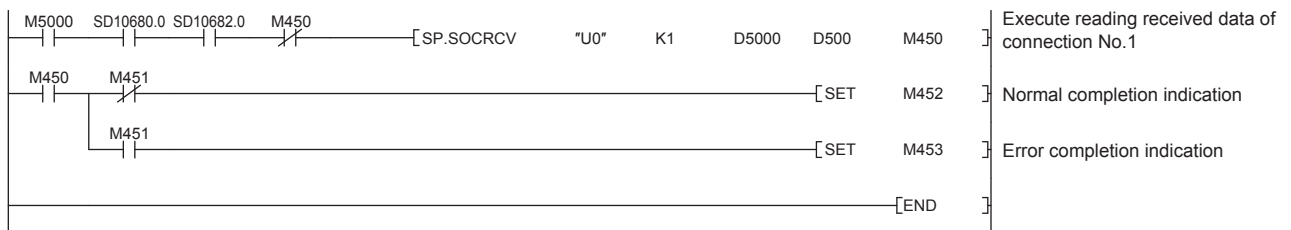
Program example

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

- Devices used

| Device No. | Application |
|------------|---|
| SD10680 | Open completion signal |
| SD10682 | Receive state signal |
| D5000 | SP.SOCRCV instruction control data |
| D500 | Received data length and received data storage location |
| M450 | SP.SOCRCV instruction completion device |
| M451 | SP.SOCRCV instruction error completion device |
| M452 | Normal completion indication |
| M453 | Error completion indication |
| M5000 | SP.SOCRCV instruction drive flag |

- Program



Point

Consecutively sent data can be consecutively read by connecting the completion device of the SP.SOCRCV instruction to the execution command as a normally closed contact.

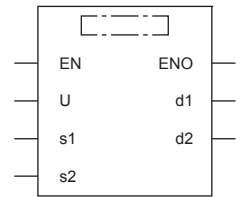
Reading received data in the processing

GP.SOCRCV

Reads the data received.(Ethernet module dedicated instruction)

| Ladder diagram | Structured text |
|----------------|--|
| | <pre>ENO:=GP_SOCRCV(EN,U,s1,s2,d1,d2);</pre> |

FBD/LD



("GP_SOCRCV" enters □.)

Setting data

■ Descriptions, ranges, and data types

| Operand | Description | Range | Data type | Data type (Label) |
|---------|--|--|------------------------|---|
| (U) | Position number of the module connected | ■FX5UJ CPU module 1H to 8H ■FX5U/FX5UC CPU module 1H to 10H | 16-bit unsigned binary | ANY16 |
| (s1) | Connection No. | 1 to 32 | 16-bit unsigned binary | ANY16 |
| (s2) | Start number of the device where control data is specified | Refer to Control data (Page 124) | Word | ANY16_ARRAY (Number of elements: 2) |
| (d1) | Start number of the device in which received data is stored | — | Word | ANY16 ^{*1} |
| (d2) | Start number of the device which turns on for one scan upon completion of the instruction. (d2)+1 also turns on when failed. | — | Bit | ANYBIT_ARRAY (Number of elements: 2) |
| EN | Execution condition | — | Bit | BOOL |
| ENO | Execution result | — | Bit | BOOL |


*1 When specifying setting data by using a label, define an array to secure enough operation area and specify an element of the array label.

■ Applicable devices

| Operand | Bit | Word | | | Double word | | Indirect specification | Constant | | | Others (U) |
|---------|-----------------------------|---------------------------|-------|---|-------------|----|------------------------|----------|---|----|------------|
| | X, Y, M, L, SM, F, B, SB, S | T, ST, C, D, W, SD, SW, R | U□\G□ | Z | LC | LZ | | K, H | E | \$ | |
| (U) | — | ○ | — | — | — | — | ○ | ○ | — | — | ○ |
| (s1) | ○ | ○ | — | — | — | — | ○ | ○ | — | — | — |
| (s2) | — | ○ | — | — | — | — | ○ | — | — | — | — |
| (d1) | — | ○ | — | — | — | — | ○ | — | — | — | — |
| (d2) | ○ | ○ ^{*1} | — | — | — | — | — | — | — | — | — |

*1 T, ST, C cannot be used.

Control data

| Device | Item | Description | Setting range | Set by ^{*1} |
|------------------|----------------------|---|---------------|----------------------|
| (s2)+0 | System area | — | — | — |
| (s2)+1 | Completion status | Completion status is stored 0000H: Completed Other than 0000H: Failed (Error code) Refer to  Page 245 Error Codes | — | System |
| (d1)+0 | Received data length | The length of the data which was read from the Socket communication receiving data area is stored. (in bytes) | 0 to 2046 | System |
| (d1)+1 to (d1)+n | Received data | The data which was read from the Socket communication receiving data area is stored in order. | — | System |

*1 System: The Ethernet module stores the execution result of the instruction.

Point

- When the GP.SOCRCV instruction is executed, data is read from socket communication receiving data area at END processing. Therefore, executing the GP.SOCRCV instruction will increase the scan time.
- When odd-byte data is received, an invalid byte is stored to the higher byte of the device that stores the last received data.

Processing details

This instruction reads received data of the connection specified in (s1) from the socket communication receive data area in the END processing after execution of the GP.SOCRCV instruction.

The result of the GP.SOCRCV instruction can be checked with the completion device (d2)+0 and (d2)+1.

- Completion device (d2)+0: Turns on in the END processing of the scan after completion of the GP.SOCRCV instruction, and turns off in the next END processing.
- Completion device (d2)+1: Turns on or off according to the status at the time of completion of the GP.SOCRCV instruction.

| Status | Description |
|----------------|---|
| When completed | Remains off. |
| When failed | Turns on in the END processing of the scan after completion of the GP.SOCRCV instruction, and turns off in the next END processing. |

Operation error

| Error code ((s2)+1) | Description |
|---------------------|--|
| 1D80H | An instruction other than executable dedicated instruction numbers has been specified. |
| 1D83H | The dedicated instruction request data from the CPU module was discarded because the data size was abnormal. |
| 1D84H | The dedicated instruction request data from the CPU module cannot be normally input. |
| 1D85H | A timeout error occurred in the dedicated instruction response data, and the data was discarded. |
| C029H | Description of control data is not correct. |
| C1A6H | A connection No. other than 1 to 32 has been specified. |

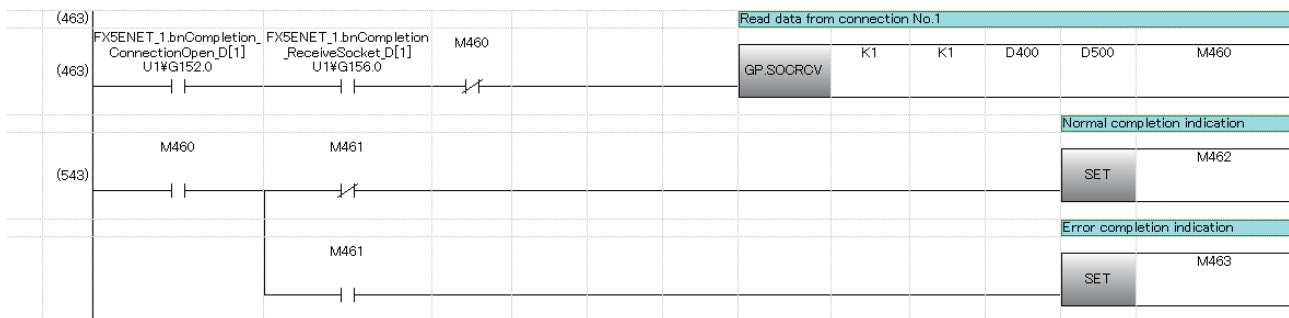
Program example

Data received from the connected device is read.

- Devices used

| Device No. | Application |
|------------------|---|
| Un\G152, Un\G153 | Open completion signal |
| Un\G156, Un\G157 | Socket communications receive status signal |
| D400 | GP.SOCRCV instruction control data |
| D500 or later | Received data length and received data storage location |
| M460 | GP.SOCRCV instruction completion device |
| M461 | GP.SOCRCV instruction error completion device |
| M462 | Normal completion indication |
| M463 | Error completion indication |

- Program



Point

Consecutively sent data can be consecutively read by connecting the completion device of the GP.SOCRCV instruction to the execution command as a normally closed contact.

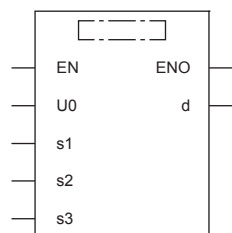
Sending data

SP.SOCSND

Sends data. (CPU module dedicated instruction)

| Ladder diagram | Structured text |
|----------------|--|
| | <pre>ENO:=SP_SOCSND(EN,U0,s1,s2,s3,d);</pre> |

FBD/LD



("SP_SOCSND" enters □.)

Setting data

■ Descriptions, ranges, and data types

| Operand | Description | Range | Data type | Data type (Label) |
|---------|---|---------------------------------------|------------------------|---|
| (U)*1 | Dummy (Input the character string ["U0"].) | — | Character string | ___*2 (ANYSTRING_SINGLE) |
| (s1) | Connection No. | 1 to 8 | 16-bit unsigned binary | ANY16 |
| (s2) | Start number of the device where control data is specified | Refer to Control data (☞ Page 127) | Word | ANY16_ARRAY (Number of elements: 2) |
| (s3) | Start number of the device in which send data is stored | — | Word | ANY16 |
| (d) | Start number of the device which turns on for one scan upon completion of the instruction. (d)+1 also turns on when failed. | — | Bit | ANYBIT_ARRAY (Number of elements: 2) |
| EN | Execution condition | — | Bit | BOOL |
| ENO | Execution result | — | Bit | BOOL |

*1 In the case of the ST language and the FBD/LD language, U displays as U0.


*2 Regardless of the program language to be used, the data type is specified by a device. Do not specify a label.

■ Applicable devices

| Operand | Bit | Word | | | Double word | | Indirect specification | Constant | | | Others |
|---------|-----------------------------|---------------------------|-------|---|-------------|----|------------------------|----------|---|----|--------|
| | X, Y, M, L, SM, F, B, SB, S | T, ST, C, D, W, SD, SW, R | U□\G□ | Z | LC | LZ | | K, H | E | \$ | |
| (U) | — | — | — | — | — | — | — | — | — | ○ | — |
| (s1) | — | ○ | — | — | — | — | ○ | ○ | — | — | — |
| (s2) | — | ○ | — | — | — | — | ○ | — | — | — | — |
| (s3) | — | ○ | — | — | — | — | ○ | — | — | — | — |
| (d) | ○ | ○*1 | — | — | — | — | — | — | — | — | — |

*1 T, ST, C cannot be used.

Control data

| Device | Item | Description | Setting range | Set by ^{*1} |
|------------------|-------------------|--|---------------|----------------------|
| (s2)+0 | System area | — | — | — |
| (s2)+1 | Completion status | Completion status is stored. 0000H: Completed Other than 0000H: Failed (Error code) Refer to  Page 245 Error Codes | — | 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 is specified. | — | User |

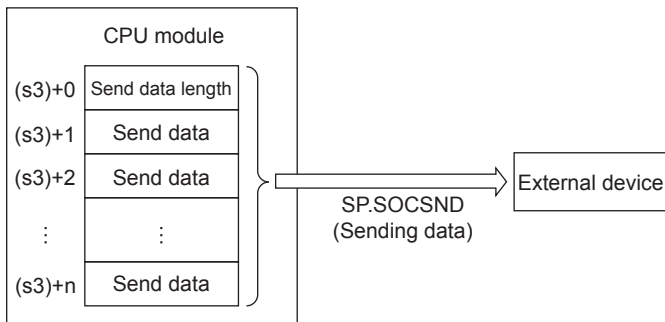
*1 User: Data to be set before the execution of the instruction. System: The CPU module stores the execution result of the instruction.

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.

Processing details

This instruction sends data set in (s3) to the target device of the connection specified by (s1).



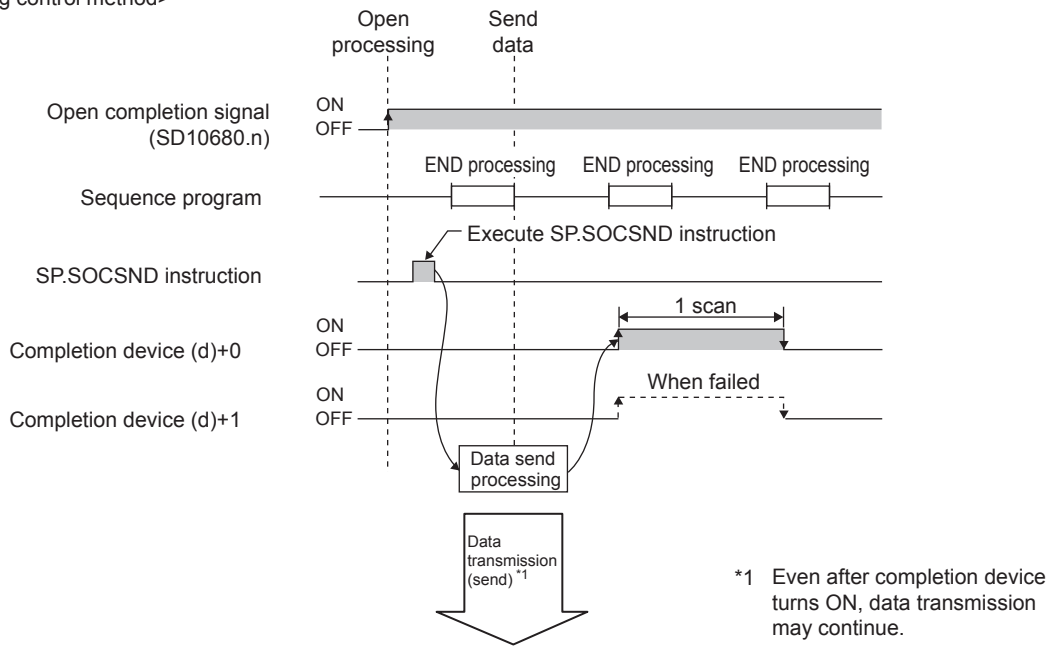
The result of the SP.SOCSND instruction can be checked with the completion device, (d)+0 and (d)+1.

- Completion device (d)+0: Turns on in the END processing of the scan after completion of the SP.SOCSND instruction, and turns off in the next END processing.
- Completion device (d)+1: Turns ON or OFF according to the status at the time of completion of the SP.SOCSND instruction.

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

The following figure shows the timing of send processing with the SP.SOCSND instruction.

<Sending control method>



Operation error

| Error code (SD0/SD8067) | Description |
|-------------------------|---|
| 3405H | The connection number specified by (s1) is other than 1 to 8. |
| 2820H | The device number specified by (s2), (s3) or (d) is outside the range of the number of device points. |
| 2822H | Device that cannot be specified is specified. |
| 3582H | When an instruction which cannot be used in interruption routine program is used. |

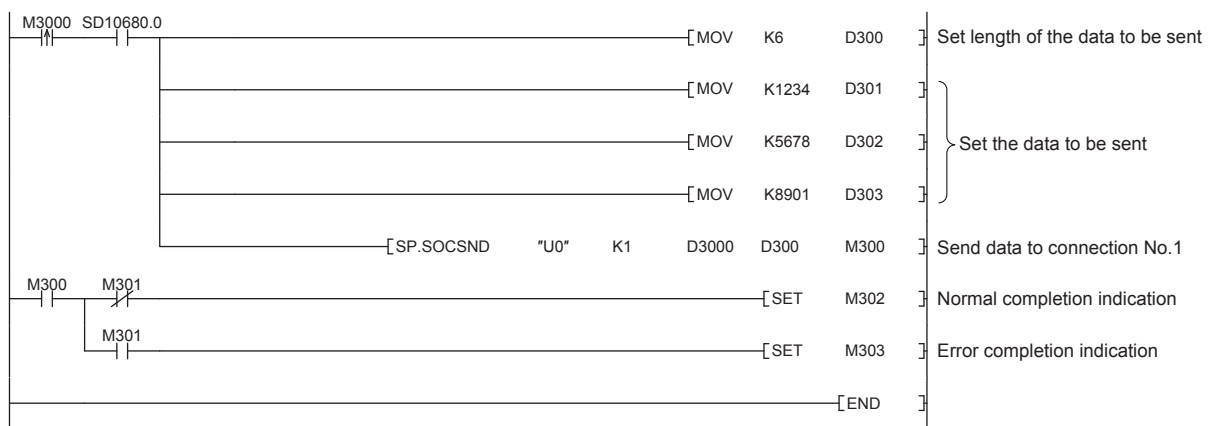
Program example

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

- Devices used

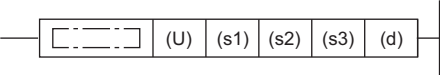
| Device No. | Application |
|--------------|---|
| SD10680 | Open completion signal |
| D3000 | SP.SOCSND instruction control data |
| D300 to D303 | Send data length and send data storage location |
| M300 | SP.SOCSND instruction completion device |
| M301 | SP.SOCSND instruction error completion device |
| M302 | Normal completion indication |
| M303 | Error completion indication |
| M3000 | SP.SOCSND instruction drive flag |

- Program

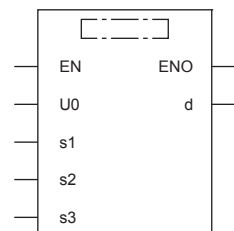


GP.SOCSND

Sends data.(Ethernet module dedicated instruction)

| Ladder diagram | Structured text |
|---|----------------------------------|
|  | ENO:=GP_SOCSND(EN,U,s1,s2,s3,d); |


FBD/LD



("GP_SOCSND" enters □.)

Setting data

■ Descriptions, ranges, and data types

| Operand | Description | Range | Data type | Data type (Label) |
|---------|---|--|------------------------|---|
| (U) | Position number of the module connected | ■FX5UJ CPU module 1H to 8H ■FX5U/FX5UC CPU module 1H to 10H | 16-bit unsigned binary | ANY16 |
| (s1) | Connection No. | 1 to 32 | 16-bit unsigned binary | ANY16 |
| (s2) | Start number of the device where control data is specified | Refer to Control data ( Page 131) | Word | ANY16_ARRAY (Number of elements: 2) |
| (s3) | Start number of the device in which send data is stored | — | Word | ANY16 ^{*1} |
| (d) | Start number of the device which turns on for one scan upon completion of the instruction. (d)+1 also turns on when failed. | — | Bit | ANYBIT_ARRAY (Number of elements: 2) |
| EN | Execution condition | — | Bit | BOOL |
| ENO | Execution result | — | Bit | BOOL |


*1 When specifying setting data by using a label, define an array to secure enough operation area and specify an element of the array label.

■ Applicable devices

| Operand | Bit | Word | | | Double word | | Indirect specification | Constant | | | Others (U) |
|---------|-----------------------------|---------------------------|-------|---|-------------|----|------------------------|----------|---|----|------------|
| | X, Y, M, L, SM, F, B, SB, S | T, ST, C, D, W, SD, SW, R | U□\G□ | Z | LC | LZ | | K, H | E | \$ | |
| (U) | — | ○ | — | — | — | — | ○ | ○ | — | — | ○ |
| (s1) | ○ | ○ | — | — | — | — | ○ | ○ | — | — | — |
| (s2) | — | ○ | — | — | — | — | ○ | — | — | — | — |
| (s3) | — | ○ | — | — | — | — | ○ | — | — | — | — |
| (d) | ○ | ○ ^{*1} | — | — | — | — | — | — | — | — | — |

*1 T, ST, C cannot be used

Control data

| Device | Item | Description | Setting range | Set by ^{*1} |
|------------------|-------------------|--|---------------|----------------------|
| (s2)+0 | System area | — | — | — |
| (s2)+1 | Completion status | Completion status is stored. 0000H: Completed Other than 0000H: Failed (Error code) Refer to  Page 245 Error Codes | — | 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 is specified. | — | User |

*1 User: Data to be set before the execution of the instruction. System: The Ethernet module stores the execution result of the 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.

Processing details

This instruction sends data set in (s3) to the target device of the connection specified by (s1).

The result of the GP.SOCSND instruction can be checked with the completion device, (d)+0 and (d)+1.

- Completion device (d)+0: Turns on in the END processing of the scan after completion of the GP.SOCSND instruction, and turns off in the next END processing.
- Completion device (d)+1: Turns ON or OFF according to the status at the time of completion of the GP.SOCSND instruction.

| Status | Description |
|----------------|---|
| When completed | Remains off. |
| When failed | Turns on in the END processing of the scan after completion of the GP.SOCSND instruction, and turns off in the next END processing. |

Operation error

| Error code ((s2)+1) | Description |
|---------------------|--|
| 1D80H | An instruction other than executable dedicated instruction numbers has been specified. |
| 1D83H | The dedicated instruction request data from the CPU module was discarded because the data size was abnormal. |
| 1D84H | The dedicated instruction request data from the CPU module cannot be normally input. |
| 1D85H | A timeout error occurred in the dedicated instruction response data, and the data was discarded. |
| C020H | A data size exceeding the maximum data length has been specified. |
| C027H | Socket communication send message has failed. |
| C029H | Description of control data is not correct. |
| C1A6H | A connection No. other than 1 to 32 has been specified. |

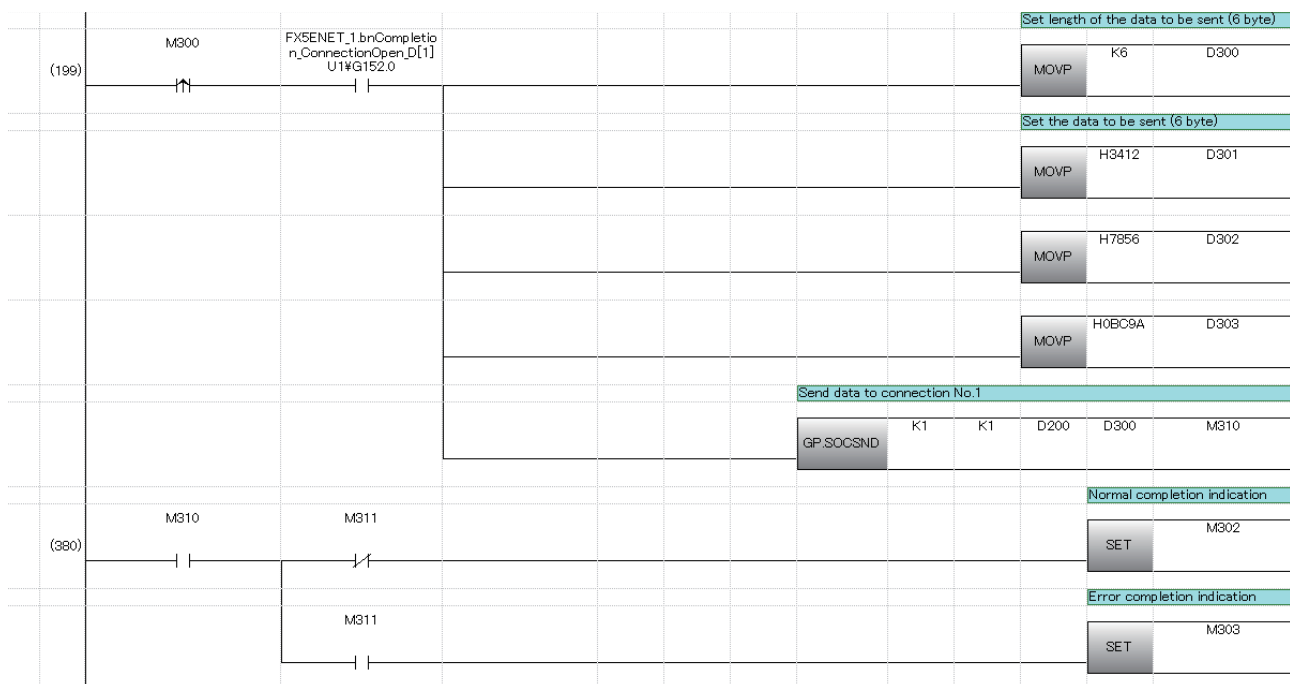
Program example

When M300 is turned on, data (3412H, 7856H, BC9AH) are sent to the target device using the socket communication function.

- Devices used

| Device No. | Application |
|------------------|---|
| Un\G152, Un\G153 | Open completion signal |
| D200 | GP.SOCSND instruction control data |
| D300 to D303 | Send data length and send data storage location |
| M300 | GP.SOCSND instruction drive flag |
| M302 | Normal completion indication |
| M303 | Error completion indication |
| M310 | GP.SOCSND instruction completion device |
| M311 | GP.SOCSND instruction error completion device |

- Program



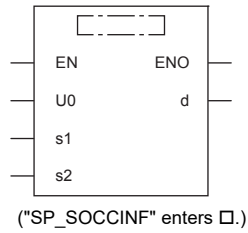
Reading connection information

SP.SOCCINF

Reads connection information. (CPU module dedicated instruction)

| Ladder diagram | Structured text |
|----------------|--|
| | <pre>ENO:=SP_SOCCINF(EN,U0,s1,s2,d);</pre> |

FBD/LD



Setting data

■ Descriptions, ranges, and data types

| Operand | Description | Range | Data type | Data type (Label) |
|-------------------|--|--------------------------------------|------------------------|--|
| (U) ^{*1} | Dummy (Input the character string ['U0'].) | — | Character string | __*2 (ANYSTRING_SINGLE) |
| (s1) | Connection No. | 1 to 8 | 16-bit unsigned binary | ANY16 |
| (s2) | Start number of the device in which control data is stored | Refer to Control data (Page 134) | Word | ANY16_ARRAY (Number of elements: 2) |
| (d) | Start number of the device in which connection information is stored | — | Word | ANY16_ARRAY (Number of elements: 5) |
| EN | Execution condition | — | Bit | BOOL |
| ENO | Execution result | — | Bit | BOOL |


*1 In the case of the ST language and the FBD/LD language, U displays as U0.

*2 Regardless of the program language to be used, the data type is specified by a device. Do not specify a label.

■ Applicable devices

| Operand | Bit | Word | | | Double word | | Indirect specification | Constant | | | Others |
|---------|-----------------------------|---------------------------|-------|---|-------------|----|------------------------|----------|---|----|--------|
| | X, Y, M, L, SM, F, B, SB, S | T, ST, C, D, W, SD, SW, R | U□\G□ | Z | LC | LZ | | K, H | E | \$ | |
| (U) | — | — | — | — | — | — | — | — | — | ○ | — |
| (s1) | — | ○ | — | — | — | — | ○ | ○ | — | — | — |
| (s2) | — | ○ | — | — | — | — | ○ | — | — | — | — |
| (d) | — | ○ | — | — | — | — | ○ | — | — | — | — |

Control data

| Device | Item | Description | Setting range | Set by ^{*1} | | | | | |
|----------------|---------------------------|--|---|----------------------|-----|-----|---|----------------------------------|--|
| (s2)+0 | System area | — | — | — | | | | | |
| (s2)+1 | Completion status | Completion status is stored 0000H: Completed Other than 0000H: Failed (Error code) Refer to  Page 245 Error Codes | — | System | | | | | |
| (d)+0 (d)+1 | Target device IP address | IP address of the target device is stored. | 1 to 3758096382 (00000001H to DFFFFFFEH) ^{*2} | | | | | | |
| (d)+2 | Target device port number | Port number of the target device is stored. | 1 to 65534 (0001H to FFFE ^H) ^{*2} | | | | | | |
| (d)+3 | Host Station Port No. | Port number of the host station is stored. | 1 to 5548, 5570 to 65534 (0001H to 15ACH, 15C2H to FFFE ^H) ^{*2,3} | | | | | | |
| (d)+4 | Application setting area | <div style="text-align: center;"> b15b14b13 to b10 b9 b8 b7 to b0 (d)+4 <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td>[3]</td> <td>0</td> <td>[2]</td> <td>[1]</td> <td>0</td> </tr> </table> </div> [1] Communication method (protocol) 0: TCP/IP 1: UDP/IP [2] Socket communication procedure 1: Non-protocol method [3] Open system 00: Active open or UDP/IP 10: Unpassive open 11: Fullpassive open | [3] | 0 | [2] | [1] | 0 | Shown on left side ^{*2} | |
| [3] | 0 | [2] | [1] | 0 | | | | | |

*1 System: The CPU module stores the execution result of the instruction.

*2 In case of execution for an unopened connection, 0H is returned.

*3 Because host station port numbers, 1 to 1023 (0001H to 03FFH), are assigned for reserved port numbers and 61440 to 65534 (F000H to FFFE^H) are used for other communication functions, using 1024 to 5548, 5570 to 61439 (0400H to 15ACH, 15C2H to EFFFH) is recommended. Do not specify 5549 to 5569 (15ADH to 15C1H) because these ports are used by the system.

Processing details

This instruction reads connection information specified in (s1).

Operation error

| Error code (SD0/SD8067) | Description |
|-------------------------|---|
| 3405H | The connection number specified by (s1) is other than 1 to 8. |
| 2820H | The device number specified by (s2) or (d) is outside the range of the number of device points. |
| 2822H | Device that cannot be specified is specified. |

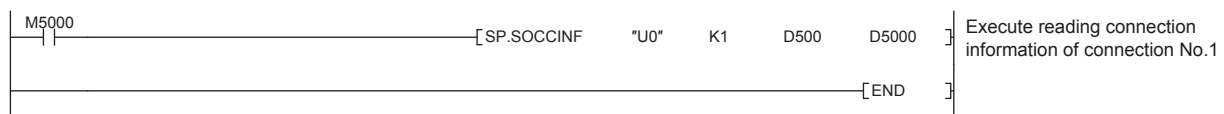
Program example

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

- Devices used

| Device No. | Application |
|------------|--|
| D500 | SP.SOCCINF instruction control data |
| D5000 | Storage location of connection information |
| M5000 | SP.SOCCINF instruction drive flag |

- Program



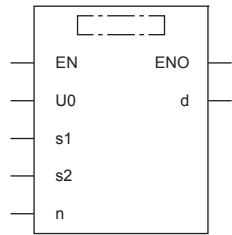
Reading socket communication receive data

S(P).SOCRDATA

Reads data from the socket communication receive data area. (CPU module dedicated instruction)

| Ladder diagram | Structured text |
|----------------|---|
| | <pre>ENO:=S_SOCRDATA(EN,U0,s1,s2,n,d); ENO:=SP_SOCRDATA(EN,U0,s1,s2,n,d);</pre> |

FBD/LD



("S_SOCRDATA", "SP_SOCRDATA" enters □.)

Setting data

■Descriptions, ranges, and data types

| Operand | Description | Range | Data type | Data type (Label) |
|-------------------|--|---------------------------------------|------------------------|--|
| (U) ^{*1} | Dummy (Input the character string ["U0"].) | — | Character string | ___*2 (ANYSTRING_SINGLE) |
| (s1) | Connection No. | 1 to 8 | 16-bit unsigned binary | ANY16 |
| (s2) | Start number of the device in which control data is stored | Refer to Control data (☞ Page 135) | Word | ANY16_ARRAY (Number of elements: 2) |
| (d) | Start number of the device where read data is stored | — | Word | ANY16 |
| (n) | Number of read data (1 to 1024 words) | 1 to 1024 | 16-bit signed binary | ANY16 |
| EN | Execution condition | — | Bit | BOOL |
| ENO | Execution result | — | Bit | BOOL |

*1 In the case of the ST language and the FBD/LD language, U displays as U0.

*2 Regardless of the program language to be used, the data type is specified by a device. Do not specify a label.

■Applicable devices

| Operand | Bit | Word | | | Double word | | Indirect specification | Constant | | | Others |
|---------|-----------------------------|---------------------------|-------|---|-------------|----|------------------------|----------|---|----|--------|
| | X, Y, M, L, SM, F, B, SB, S | T, ST, C, D, W, SD, SW, R | U□\G□ | Z | LC | LZ | | K, H | E | \$ | |
| (U) | — | — | — | — | — | — | — | — | — | ○ | — |
| (s1) | — | ○ | — | — | — | — | ○ | ○ | — | — | — |
| (s2) | — | ○ | — | — | — | — | ○ | — | — | — | — |
| (d) | — | ○ | — | — | — | — | ○ | — | — | — | — |
| (n) | — | ○ | — | — | — | — | ○ | ○ | — | — | — |

■Control data

| Device | Item | Description | Setting range | Set by ^{*1} |
|--------|-------------------|---|---------------|----------------------|
| (s2)+0 | System area | — | — | — |
| (s2)+1 | Completion status | Completion status is stored 0000H: Completed Other than 0000H: Failed (Error code) Refer to ☞ Page 245 Error Codes | — | System |

*1 System: The CPU module stores the execution result of the instruction.

Processing details

This instruction reads the data of the amount specified for n from the socket communication receive data area of connection that is specified in (s1), and stores them in the device specified in (d) or higher. No processing is performed when the number of read data (n) is 0.

Point

The received data length can be read by setting the number of read data to one word. This allows change of the device storing receive data, when executing the SP.SOCRCV instruction.

Precautions

- 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 received data, read the data using the SP.SOCRCV instruction.

Operation error

| Error code (SD0/SD8067) | Description |
|-------------------------|---|
| 3405H | The connection number specified by (s1) is other than 1 to 8. |
| 2820H | The device number specified by (s2), (d), or (n) is outside the range of the number of device points. |
| 2822H | Device that cannot be specified is specified. |

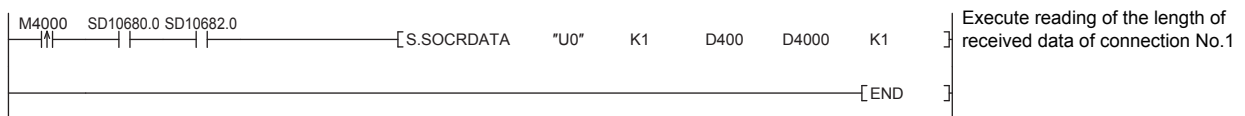
Program example

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

- Devices used

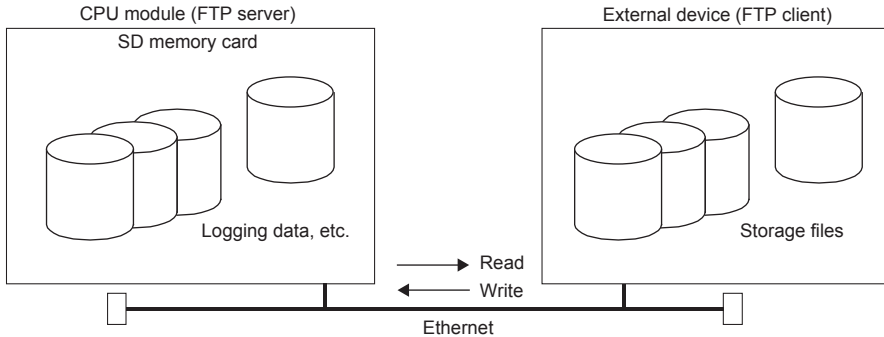
| Device No. | Application |
|------------|-------------------------------------|
| SD10680 | Open completion signal |
| SD10682 | Receive state signal |
| D400 | S.SOCRDATA instruction control data |
| D4000 | Storage location where data is read |
| M4000 | S.SOCRDATA instruction drive flag |
| K1 | Number of read data (one word) |

- Program



8 FILE TRANSFER FUNCTION (FTP SERVER)

The server function of FTP (File Transfer Protocol) used to transfer files to an external device is supported. An external device equipped with the FTP client functions can handle the files (data logging file, etc.) in the SD memory card installed on a CPU module as follows.



- Reading of file from SD memory card (download)
- Writing of file to the SD memory card (upload)
- Browsing of file names in SD memory card

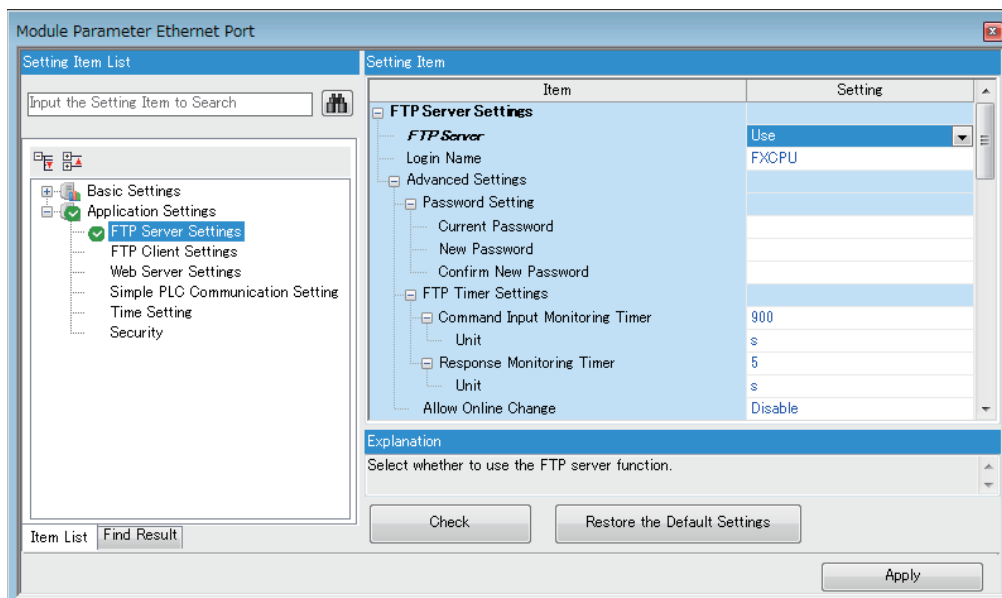
8.1 Data communication procedures

The following section describes the settings for FTP communication.

Setting in the CPU module side

The file transfer function (FTP server) of the CPU module is as follows.

Navigation window ⇒ [Parameter] ⇒ [FX5UJCPU] or [FX5UCPU] ⇒ [Module Parameter] ⇒ [Ethernet Port] ⇒ [Application Settings] ⇒ [FTP Server Settings]



| Item | Description | Setting range |
|------------|---|---|
| FTP Server | Select whether to use the file transfer function (FTP server) of the CPU module. | <ul style="list-style-type: none"> • Not Use • Use (Default: Not Use) |
| Login Name | Set the login name to be used for file transfer request (login) from the external device. | 12 characters maximum (one-byte alphanumeric character) (Default: FXCPU) |

| Item | | Description | Setting range |
|------------------|---------------------|--|--|
| Advanced Setting | Password Setting | Set the password to be used for file transfer request (login) from the external device. | Page 138 Password Setting |
| | FTP Timer Settings | Set the command input monitoring timer and the response monitoring timer used for the file transfer function (FTP server). | Page 138 FTP timer settings |
| | Allow Online Change | Select whether to enable data writing from the external device using the file transfer function (FTP server) while the CPU module is in RUN state. | <ul style="list-style-type: none"> • Disable • Enable (Default: Disable) |

■ Password Setting

- Current password

Enter the current password for login to the CPU module.

Default password (initial setting) is "FXCPU".



Although the default password can be used, it is recommended to change it to another password to prevent unauthorized access.

- New password, confirm new password

Enter the new password in "New Password" and "Confirm New Password" when changing the password.

Set a password within 1 to 32 one-byte characters. Number, alphabet, special character (?,!& \ % # * () [] , etc.) can be used.

■ FTP timer settings

- Command input monitoring timer

Set the monitoring time for the CPU module to monitor the command input time from the FTP client.

It is recommended to use the default value (900 s) for this timer value as much as possible.

When changing the setting value, determine the command input monitoring timer value upon consulting with the administrator of the external device or system.

Set a value within the following range.

| Unit | Setting range |
|------|-----------------|
| s | 1 to 16383 |
| ms*1 | 100 to 16383000 |

*1 Set in increments of 100 ms. Only FX5U/FX5UC CPU module is supported.

The FTP connection is disconnected if there is no command input from the FTP client side within the time of the command input monitoring timer value after the FTP client login.

When restarting the file transfer, start over from the login operation again.

- Response Monitoring Timer

Set the monitoring time for a response from the CPU module after the CPU module receives the request data from the external device.

It is recommended to use the default value (5 s) for this timer value as much as possible.

When changing the setting value, determine the response monitoring timer value upon consulting with the system administrator.

Set a value within the following range.

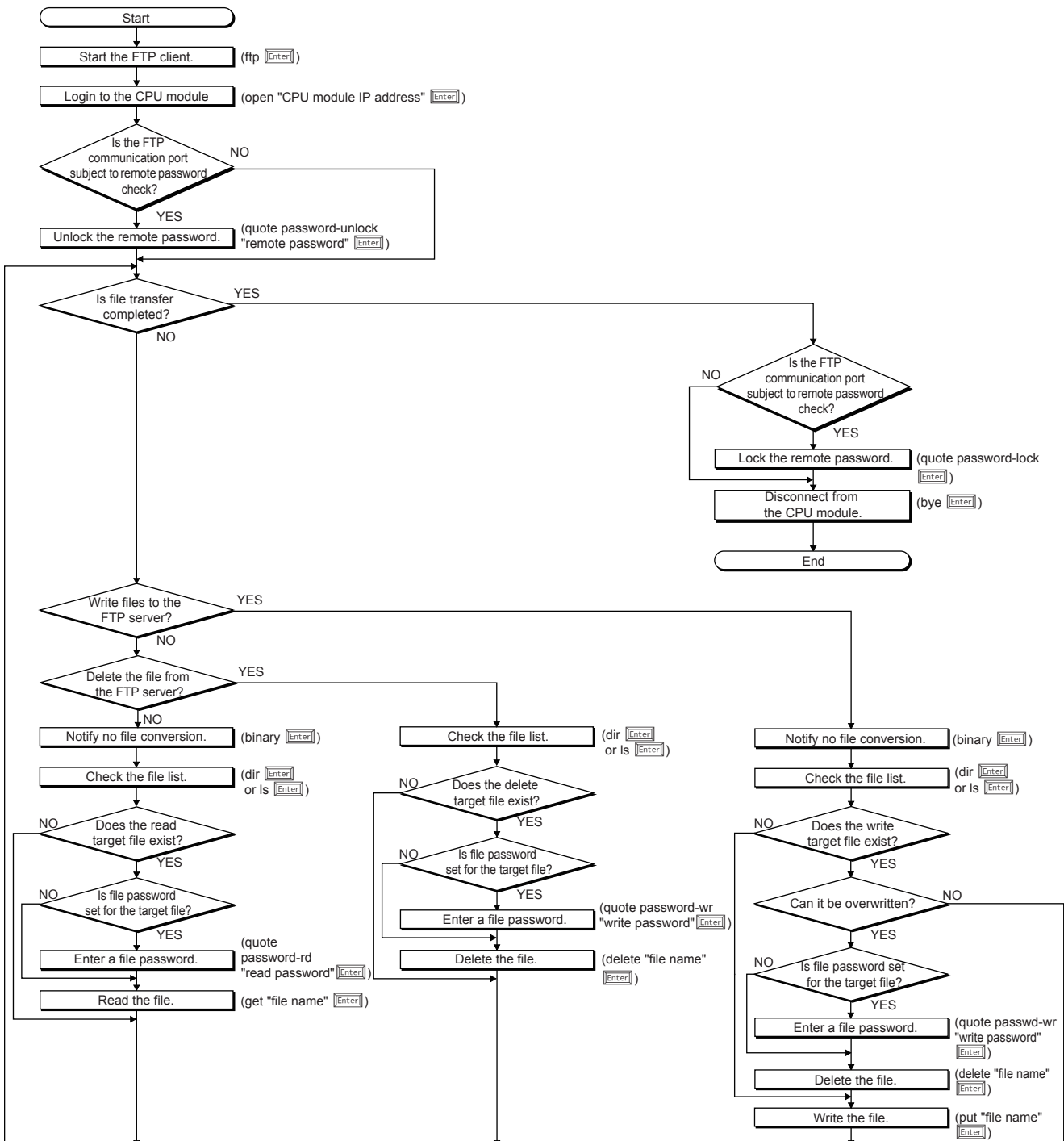
| Unit | Setting range |
|------|-----------------|
| s | 1 to 16383 |
| ms*1 | 100 to 16383000 |

*1 Set in increments of 100 ms. Only FX5U/FX5UC CPU module is supported.

Operations on external device (FTP client) side

This section describes the procedures and required processes on the external device side for using the CPU module's file transfer function (FTP server). The FTP commands and input format used for the operation are shown in the explanation.

("Enter") means to input Enter or the Return key.)



Logging into CPU module

This section describes the steps from starting FTP and logging into the CPU module.

Ex.

Start FTP from the Microsoft® Windows® command prompt.

```
ca. Command Prompt - ftp
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows>ftp
ftp> open 192.168.3.250
Connected to 192.168.3.250.
220 iQ-F FTP server ready.
User (192.168.3.250:(none)): FXCPU
331 Password required.
Password:
230 User logged in.
ftp>
```

- ① FTP start (FTP)
- ② Connect with FTP server (open CPU module IP address)
- ③ Specify login name (login name)
- ④ Specify password (password)

Use the login name and password that are set in "FTP Server Settings" under "Application Settings". When the CPU module (FTP server) receives the login name and password from the external device (FTP client), it checks that the login name and password are correct.

If the login name and password are correct, transfer of the files to the CPU module is permitted. If incorrect, file transfer is not permitted.

Locking and unlocking the remote password

If the FTP communication port is specified as a remote password check target with the remote password setting, unlock the remote password with the following command.

- quote password-unlock remote password

When finished, lock the remote password with the following command.

- quote password-lock

Point

If the FTP communication port is specified as a remote password check target, some commands cannot be used until the remote password is unlocked. For details on the commands that can be used in the locked state, refer to the following.

☞ Page 141 FTP command

Inputting the file password

If a file password is set for the target file, the file password must be input with the following command before the file can be accessed.

- Write password (quote passwd-wr write password)
- Read password (quote passwd-rd read password)

8.2 Files that can be transferred with FTP

The file transfer function (FTP server) enables transfer of files in the SD memory card installed on a CPU module.

For the files that can be transferred (read, write, deleted) with the file transfer function (FTP server), refer to the file operation available which is described in the following manual.

 MELSEC iQ-F FX5 User's Manual (Application)


8.3 FTP command

FTP command list

The FTP client side commands supported by the CPU module are shown below.

○: Executable^{*1}, ×: Not executable^{*2}

| Command | Function | CPU module status | | | Remote password | |
|---------|--|-------------------|----------------------|-----------------------|------------------------|----------------------|
| | | STOP | RUN | | Unlocked ^{*4} | Locked ^{*4} |
| | | | Enable ^{*3} | Disable ^{*3} | | |
| binary | Notifies that the file will be transferred without conversion. | ○ | ○ | ○ | ○ | × |
| bye | Closes and ends the connection with the FTP server. | ○ | ○ | ○ | ○ | ○ |
| cd | Change the CPU module current directory. | ○ | ○ | ○ | ○ | × |
| close | Closes the connection with the FTP server. | ○ | ○ | ○ | ○ | ○ |
| delete | Deletes the CPU module file. | ○ | ○ ^{*5} | × | ○ | × |
| dir | Displays the CPU module file information. | ○ | ○ | ○ | ○ | × |
| get | Reads a file from the CPU module. | ○ | ○ | ○ | ○ | × |
| ls | Displays the CPU module file name. | ○ | ○ | ○ | ○ | × |
| mdelete | Deletes the CPU module file. | ○ | ○ ^{*5} | × | ○ | × |
| mdir | Stores the CPU module file information in the file. | ○ | ○ | ○ | ○ | × |
| mget | Reads a file from the CPU module. | ○ | ○ | ○ | ○ | × |
| mls | Reads a file from the CPU module. | ○ | ○ | ○ | ○ | × |
| mput | Writes the file to the CPU module. | ○ | ○ | × | ○ | × |
| open | Connects to the FTP server. | ○ | ○ | ○ | ○ | ○ |
| put | Writes the 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 the FTP server's subcommand. ^{*6} | ○ | ○ | ○ | ○ | ○ |
| user | Inputs the user name and password for the CPU module. | ○ | ○ | ○ | ○ | ○ |

*1 The command may not be executed depending on the file type. ( Page 141 Files that can be transferred with FTP)

*2 If executed, the process completes abnormally.


*3 Shows the "Allow Online Change" setting in "FTP Server Settings" under "Application Settings". If an illegal command is executed while write is prohibited during RUN, the process completes abnormally.

*4 Shows whether the command can be executed when the FTP communication port performs a remote password check with the remote password setting. For details on the remote password, refer to the following.

 Page 196 Remote Password

*5 The parameter file and program file cannot be deleted when the CPU module is in the RUN state.

*6 Only the subcommands dedicated for the CPU module can be used. For the subcommands that can be used, refer to the following.

 Page 142 Subcommands usable with quote command

■Subcommands usable with quote command

This section describes the CPU module dedicated commands added to the quote command and used.

When executing this command from the FTP client, input the subcommand after the quote command.

("Enter" means to input CR, Enter or the Return key.)

Ex.

Executing the STOP command

Input the following at the command prompt.

quote stop Enter

The following table lists the subcommands can be used.

○: Executable, ×: Not executable*1

| Command | Function | CPU module status | | | Remote password | |
|-----------------|--|-------------------|--------------|----------------|-----------------|--------|
| | | STOP | RUN | | Unlocked | Locked |
| | | | Write enable | Write prohibit | | |
| passwd-rd | Sets, shows or clears the file password (read password). | ○ | ○ | ○ | ○ | × |
| passwd-wr | Sets, shows, or clears the file password (write password). | ○ | ○ | ○ | ○ | × |
| password-lock | Changes the remote password from the unlock state to the lock state. | ○ | ○ | ○ | ○ | ×*2 |
| password-unlock | Changes the remote password from the lock state to the unlock state. | ○ | ○ | ○ | ○ | ○ |

*1 If executed, the process completes abnormally.


*2 Even if the subcommand is executed, the remote password remains locked with no error occurred.

Specifying an FTP command

This section describes the method of specifying the files specified with the FTP command on the FTP client (external device side) supported by the CPU module.

With the CPU module, the drive name and file name are distinguished when specifying the file.

When specifying a file in the CPU module with FTP, specify the target file with the following arrangement.*1

| Item | Description |
|-----------------------|---|
| Specification format | Drive name (drive 2):\Folder name\File name.Extension |
| Example | 2:\LOGGING\LOG01\00000001\LOG01_00000001.BIN |
| Specification details | Refer to the following.  Page 142 Drive name (drive No.),Page 142 Folder name, file name, and extension |

*1 Use "\" as the delimiter.

■Drive name (drive No.)

The destination memory for file transfer is drive 2 (SD memory card) only.

■Folder name, file name, and extension

When using a FTP command that can be used for multiple files, specify the file name and extension with the wild card "*" or "?". (Depending on the FTP client, there may be additional restrictions to the characters that can be used for the file name)

: All files with the optional character string (including none) are targeted from the position specified with "".

?: All files with the optional character string (excluding none) are targeted from the position specified with "?". ("?" can be used multiple times.)

Details of FTP command

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 depending on the client side FTP application, some of the FTP commands may not operate as described in this manual. Refer to the manual for the FTP client, and check the functions, operation methods, and so on.
- The section enclosed in square brackets [] in the specification format can be omitted.

■FTP server support command

| Command name | Description | |
|--------------|----------------------|--|
| binary | Function | Notifies the FTP server that the file will be transferred without conversion. The return code and kanji codes are also not converted. These settings are automatically applied to the CPU module. |
| | Specification format | binary (abbreviated: bin) |
| bye | Function | Closes the connection with the FTP and quits the FTP. |
| | Specification format | bye |
| | Same function | quit |
| cd | Function | Change the current directory. |
| | Specification format | cd [directory path] |
| | Example | cd 2:\LOGGING\ |
| close | Function | Closes the connection with the FTP server. |
| | Specification format | close |
| delete | Function | Deletes files stored in the CPU module. |
| | Specification format | delete "file path name" |
| | Example | When deleting files stored in the SD memory card delete 2:\MAINSEQ1.PRG |
| | Similar command | mdelete |
| dir | Function | Displays the detailed information (file name, date of creation, volume) of the file stored in the CPU module. |
| | Specification format | dir [drive name:\] |
| | Example | 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 files stored in the SD memory card and store with same file name get 2:\LOG01_0000001.BIN |
| | Example 2 | When reading files stored in the SD memory card and store with different file name get 2:\LOG01_0000001.BIN LOG\LOG01_01.B |
| | Caution | <ul style="list-style-type: none"> • If the destination file path name (FTP client side) is not specified, the file is stored in the FTP client side with the same file name as the source file name (CPU module side). • The transfer destination is in the currently connected directly when FTP is started and connected. |
| ls | Function | Displays the names of files stored in the CPU module. |
| | Specification format | ls [drive name:\] |
| | Example | ls 2:\ |
| | Similar command | dir |
| mdelete | Function | Deletes files stored in the CPU module. When deleting multiple files, specify the file name and extension in the file path name with wild cards (*, ?). |
| | Specification format | mdelete "file path name" (abbreviated: mdel) |
| | Example | When deleting all files with "CSV" extension from files stored in SD memory card mdelete 2:*.CSV |
| | Similar command | delete |

| Command name | Description | |
|--------------|----------------------|--|
| mdir | Function | Stores the detailed information (file name, date of creation, volume) of the file 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 file stored in data memory into 20160101.LOG file mdir 2:\20160101.LOG |
| | Caution | <ul style="list-style-type: none"> • 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 in the currently connected directly when FTP is started and connected. |
| | Similar command | mls |
| mget | Function | Reads a file from the CPU module. When reading multiple files, specify the file name and extension in the file path name with wild cards (*, ?). When reading multiple files, receive is confirmed before transferring each file. |
| | Specification format | mget "file path name" |
| | Example | When reading all files with "BIN" extension from files stored in SD memory card mget 2:*.BIN |
| | Caution | <ul style="list-style-type: none"> • The read file is stored with the same file name in the FTP client side. The storage destination is in the current connection directory when the FTP is started and connected. |
| mls | Function | Stores the file 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 file name of file stored in SD memory card into 20160101.LOG file mls 2:\20160101.LOG |
| | Caution | <ul style="list-style-type: none"> • 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 in the currently connected directly when FTP is started and connected. |
| | Similar command | mdir |
| mput | Function | Writes the file to the CPU module. When writing multiple files, specify the file name and extension in the file path name with wild cards (*, ?). When writing multiple files, send is confirmed before transferring each file. |
| | Specification format | mput "source file path name" |
| | Example | When writing all files with "PRG" extension mput*.PRG |
| | Caution | <ul style="list-style-type: none"> • The storage destination file name is the same as the FTP client side. • The transmission destination is the SD memory card (drive 2). |
| 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] <ul style="list-style-type: none"> • 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 (If omitted, port number 21 is used for operation) |
| | 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 192.168.3.250 |
| | Caution | The IP address can be specified to create a connection when starting the FTP. |
| put | Function | Writes the file to the CPU module. |
| | Specification format | put "source file path name" [destination file path name] |
| | Example 1 | When writing the param.PRM file to the SD memory card with the same file name put param.PRM 2:\param.PRM |
| | Example 2 | When writing the param.PRM file to the SD memory card with a different file name put param.PRM 2:\param1.PRM |
| | Caution | If the directory is not specified with the source file path name (FTP client side), the file in the current connection directory when the FTP server is started and connected is written. |
| pwd | Function | Displays current directory name of the CPU module. |
| | Specification format | pwd |

| Command name | Description | |
|-----------------|----------------------|--|
| quit | Function | Closes the connection with the FTP and quits the FTP. |
| | Specification format | quit |
| | Similar command | bye |
| password-lock | Function | Locks the remote password function set for the CPU module. This command is executed when the FTP communication port is specified as a remote password check target port. |
| | Specification format | quote password-lock The following appears as the execution results when the command ends normally. 200 Command Okey |
| | Example | When locking the remote password quote password-lock |
| password-unlock | Function | Specifies the remote password set for the CPU module and unlocks the password. This command is used when FTP communication port is specified as a remote password check target port. |
| | Specification format | quote password-unlock [remote password] <ul style="list-style-type: none"> Remote password: Specifies the remote password set in the CPU module parameters. The following appears as the execution results when the command ends normally. 200 Command Okey The following appears as the execution results when the command ends abnormally. When the remote password is not set 554 Password not Set. When another command is requested before the remote password is unlocked 555 Password Locked When the remote password exceeds the maximum length (32 bytes) 556 Password Error. When the remote password does not match 556 Password Error. When the unlock failed continuously, and the status is unlock prohibited status 556 Password Error. |
| | Example | When specifying a remote password (123456) quote password-unlock 123456 |
| | Caution | <ul style="list-style-type: none"> If the FTP communication port is specified as a remote password check target port when logging in, the password will be locked. The CPU module files can be accessed by executing this command and unlocking before starting the various FTP operation. If the FTP communication port is not specified as a remote password check target port, the processing will complete normally when the remote password is unlocked. |
| | | |
| | | |
| passwd-rd | Function | Sets the read password (file password) registered for the file transfer target file to the CPU module. Shows/clears the read password set in FTP. Use this command only when a read password is registered for the file transfer target file. The CPU module checks the password when accessing the specified file. |
| | Specification format | quote passwd-rd [read password] The following appears as the execution results when the command ends normally. <ul style="list-style-type: none"> 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 state with a read password not set: 200 Read-password is not set. The following appears as the execution results when the command ends abnormally. <ul style="list-style-type: none"> When a read password is outside the following range. Minimum: 6 byte Maximum: 32 byte 501 File, directory not present or syntax error. |
| | Example 1 | When specifying the read password (ABCD1234@efgh) quote passwd-rd ABCD1234@efgh |
| | Example 2 | When clearing the read password currently set in FTP quote passwd-rd c, or quote passwd-rd C |
| | Example 3 | When displaying the read password currently set in FTP quote passwd-rd |
| | Caution | <ul style="list-style-type: none"> One read password can be set for the FTP of the CPU module. When the file transfer target file changes and when a read password is registered for the change destination file, reset the read password for the target file. The read password is initialized (cleared) when logging into the CPU module. |

| Command name | Description | |
|--------------|----------------------|---|
| passwd-wr | Function | Sets the write password (file password) registered for the file transfer target file to the CPU module. Shows/clears the write password set in FTP. Use this command only when a write password is registered for the file transfer target file. The CPU module checks the password when accessing the specified file. |
| | Specification format | quote passwd-wr[write password] The following appears as the execution results when the command ends normally. <ul style="list-style-type: none"> • 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 state with the write password not set: 200 Write-password is not set. The following appears as the execution results when the command ends abnormally. <ul style="list-style-type: none"> • When a write password is outside the following range. Minimum: 6 bytes Maximum: 32 bytes 501 File, directory not present or syntax error. |
| | Example 1 | When specifying the write password (ABCD1234@efgh) quote passwd-wr ABCD1234@efgh |
| | Example 2 | When displaying the write password currently set in the FTP quote passwd-wr |
| | Example 3 | When clearing the write password currently set in the FTP quote passwd-wr c, or quote passwd-wr C |
| | Caution | <ul style="list-style-type: none"> • One write password can be set for the FTP of the CPU module. When the file transfer target file changes and when a write password is registered for the change destination file, reset the write password for the target file. • The write password is initialized (cleared) when logging into the CPU module. |
| | user | Function |
| | Specification format | user "user name" [FTP password] <ul style="list-style-type: none"> • User name: Login name set with CPU module parameters • FTP password: FTP password set with CPU module parameters |
| | Example 1 | When specifying the user name user FXCPU |
| | Example 2 | When specifying the user name and password user FXCPU FXCPU |

8.4 Precautions

Precautions for designing system

Design the system (such as configuration of interlock circuits in the program) so that the entire system always functions properly during file transfer to the operating system and during status control of the programmable controller.

FTP client

- The FTP command specifications may differ from this manual depending on the FTP client. In this case, refer to the manual for the FTP client and check the functions and operation methods.
- FTP operations are not possible from Microsoft® Internet Explorer®. If attempted, Internet Explorer® will issue an error.
- Specify the IP address for the FTP command without zero fill. (Do not set "1" as "001".)

Processing on CPU module side

- You can only access the files in the SD memory card installed on a CPU module.
- Do not power off or reset the CPU module, or insert/eject the SD memory card during file access. The file could be damaged if these are attempted.
- Do not manipulate the files from a peripheral, such as an engineering tool, while accessing the files. (This also applies to online operations such as writing during RUN as the files are manipulated.) If the file is manipulated from another device during the file transfer function (FTP server) operation, the peripheral may issue an error. If the processing has been halted due to an error, re-execute the processing before quitting the FTP function.

Simultaneous use with FTP client function

Do not use the FTP client and FTP server functions simultaneously.

Communication processing

- If a timeout error occurs during file transfer, the TCP/IP connection will be closed. Log into the CPU module with the FTP client again before resuming file transfer.
- The existence of the external device is checked with the FTP connection.
- The file transfer processing time will differ according to the Ethernet line's load rate (line congestion), the number of connections being used simultaneously (other connection's communication processing), and system configuration (distance between FTP server and FTP client, method of accessing CPU module).
- Only one external device (FTP client) can log into the CPU module at one time. If a connection is attempted from another FTP client in the login state, an error will occur without establishing the connection.
- If another communication function is simultaneously executed with UDP/IP during file transfer with FTP, a timeout error and others may occur. Either communicate after the file is transferred, or communicate with TCP/IP.

Writing files

- An existing file cannot be overwritten. Delete an existing file with the file delete command (delete, mdelete) before writing files.
- A read-only file or a file locked by a function other than 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.
- When writing a large file to the SD memory card, set the CPU module to STOP. If writing is performed in the RUN state, a communication error may occur.
- The number of files that can be written is maximum [maximum number of files that can be written] - 1 file. For details on the maximum number of files that can be written to the SD memory card, refer to the following.

📖 MELSEC iQ-F FX5 User's Manual (Application)

Deleting files

- Decide the timing for deleting the files for the entire system including the CPU module and engineering tool.
- Files with read-only attributes and files that are locked by a function other than FTP cannot be deleted. An error will occur if attempted.
- The file cannot be deleted if the SD memory card is protected. An error will occur if attempted.

FTP password

The FTP password can be reset with the following procedure when it is lost.

- 1.** Read the parameters from the CPU module with the engineering tool.
- 2.** Click the [Restore the Default Settings] button in "Application Settings" to return all "Application Settings" to the default values.
- 3.** Set the "FTP Server Settings" and "Application Settings" again.
- 4.** Write the set parameters to the CPU module.
- 5.** Enable the parameters by powering off and on or resetting the CPU module.

Point

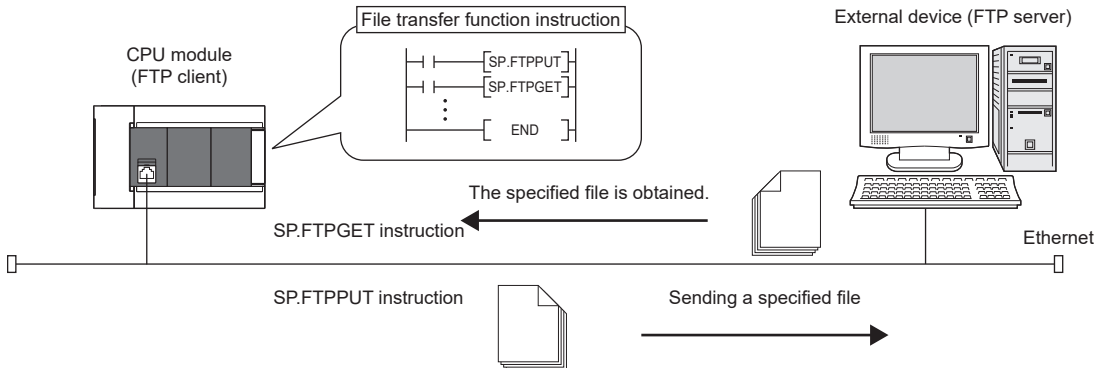
When returning to the default parameters, all items set in "Application Settings" must be reset in addition to the "FTP Server Settings".

Setting a firewall on the FTP client side

If the FTP communication is blocked by a firewall on the FTP client side, data cannot be exchanged from the FTP server. Check the firewall settings, enable FTP communication and then access the FTP server.

9 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 File transfer (transmission/get) in the SD memory card installed on a CPU module is possible.



*1 To execute this function, an FTP server is required. For details on an FTP server, refer to the manual for the server used.

Restriction

File transfer function (FTP client) is supported only for FX5U/FX5UC CPU module.

9.1 File Transfer Specifications of File Transfer Function (FTP Client)

The following table summarizes the file transfer specifications of the file transfer function (FTP client).

| Item | Description |
|---|--|
| FTP server whose operation has been checked by Mitsubishi | Microsoft® Internet Information Services (IIS) Target OS: • Microsoft® Windows® 10 • Microsoft® Windows® 8.1 • Microsoft® Windows® 8 |
| Number of connectable FTP servers | 1 |
| FTP transfer mode | Binary mode |

Transferable file

Sending a file to the FTP server (SP.FTPPUT instruction)

Files (logging files and memory dump files) in the LOGGING folder and DEBUG folder can be specified and transferred. For data logging function and memory dump function, refer to MELSEC iQ-F FX5 User's Manual (Application).

Getting a file from the FTP server (SP.FTPGET instruction)

General data (binary data, CSV data, etc.) can be specified and obtained from the FTP server.

Path specification example

■When a logging file is specified

2:\LOGGING\LOG01\00000001\20200110_00000001.BIN

■When a memory dump file is specified

2:\DEBUG\MEMDUMP\MEMDUMP_00.DPD

■When an FTP server file is specified

\DATABASE\DATA01.CSV




Point

“\” or “/” can be used as a delimiter between folder path and file. However, “\” cannot be used as a delimiter for some FTP servers.

9.2 Procedure for Executing the File Transfer Function (FTP Client)

To execute the file transfer function (FTP client), set an FTP client and an FTP server in parameters, and execute a file transfer function instruction.

Follow the procedures described below.

1. Set the FTP server. ( Page 150 Setting in the external device (FTP server))
2. Set the FTP client. ( Page 151 Setting in the CPU module (FTP client))
3. Transfer a file. ( Page 151 Executing a file transfer function instruction)

Setting in the external device (FTP server)

Set the login name, password, and home directory for the FTP client in the FTP server. In addition, give users a permission to access (read/write) files. For details, refer to manual for the server used.

Setting in the CPU module (FTP client)

The file transfer function (FTP client) of the CPU module is as follows.

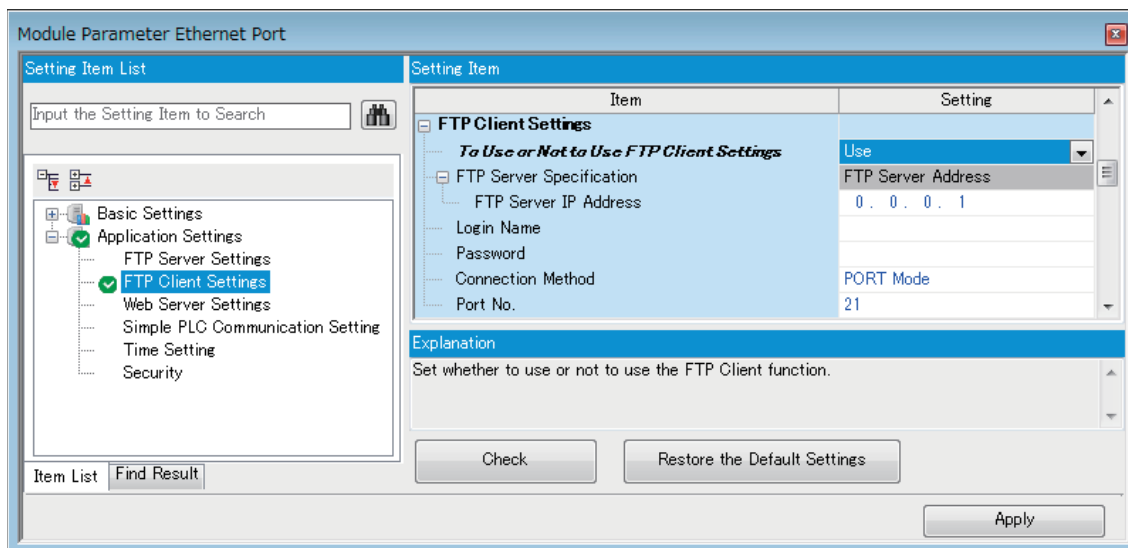
1. Set the IP address of the CPU module to be used as an FTP client. (☞ Page 31 Setting module parameters)
2. Set the file transfer function (the FTP client). (☞ Page 151 FTP Client Setting)

Point

Set an IP address of the same address class as that of the FTP server.

FTP Client Setting

Navigation window⇒[Parameter]⇒[FX5UCPU]⇒[Module Parameter]⇒[Ethernet Port]⇒[Application Settings]⇒[FTP Client Settings]



| Item | Description | Setting range |
|--|---|---|
| To Use or Not to Use FTP Client Settings | Set whether to use the file transfer function (FTP client) or not. | <ul style="list-style-type: none"> • Not Use • Use (Default: Not Use) |
| FTP Server Specification | Set the specification method of the FTP server. | FTP Server Address (Fixed) |
| FTP Server IP Address | Set the IP address of the FTP server of the connection destination in the decimal format. | 0.0.0.1 to 223.255.255.254 |
| Login Name | Set the login name for login to the FTP server. | 1 to 32 characters |
| Password | Set the password for login to the FTP server. | 0 to 32 characters |
| Connection Method | Set the connection method of the FTP server. | <ul style="list-style-type: none"> • PORT Mode • PASV Mode (Default: PORT Mode) |
| Port No. | Set the port number for control of the FTP server. | 1 to 65535 (Default: 21) |

Executing a file transfer function instruction

Use the dedicated instructions to execute the file transfer function (FTP client). For the file transfer function instruction, refer to MELSEC iQ-F FX5 Programming Manual (Instructions, Standard Functions/Function Blocks).

| Instruction | Description |
|-------------|---|
| SP.FTPPUT | This instruction sends files in the CPU module (FTP client) to the folder path of the specified FTP server. |
| SP.FTPGET | Files in the FTP server are obtained in the folder path of the specified CPU module (FTP client). |

9.3 Program Example

Program examples to transfer logging files are shown below.

For details on data logging function, refer to  MELSEC iQ-F FX5 User's Manual (Application).

Program example to specify wild cards

Following is a program example to log 30 word device points and 10 bit device points in the CSV format every 500 ms, and use the FTP client function to transfer logging files to the FTP server every hour. (It is a program to perform logging file switchover every hour and transfer all logging files under a designated folder by using the file switchover as a trigger.)

Setting in the external device (FTP server)


The FTP server's IP address is as follows.

| Item | Setting value |
|-------------|---------------|
| IP address | 192.168.3.101 |
| Subnet mask | 255.255.255.0 |
| Login name | user |
| Password | 1234abcd |

Setting in the CPU module (FTP client)


The FTP client's parameter settings are as follows.

■Own Node Settings

 Navigation window⇒[Parameter]⇒[FX5UCPU]⇒[Module Parameter]⇒[Ethernet Port]⇒[Basic Settings]⇒[Own Node Settings]

| Item | Setting value |
|-------------|---------------|
| IP Address | 192.168.3.250 |
| Subnet Mask | 255.255.255.0 |

■FTP Client Setting

 Navigation window⇒[Parameter]⇒[FX5UCPU]⇒[Module Parameter]⇒[Ethernet Port]⇒[Application Settings]⇒[FTP Client Settings]

| Item | Setting value |
|--|-----------------------|
| To Use or Not to Use FTP Client Settings | Use |
| FTP Server Specification | FTP Server IP Address |
| FTP Server IP Address | 192.168.3.101 |
| Login Name | user |
| Password | 1234abcd |
| Connection Method | PORT Mode |
| Port No. | 21 |

Logging settings

Logging files to be transferred to the FTP server are those files created in accordance with the following logging settings. If any item is not described here, set it optionally.

| Item | | Setting value | | |
|-------------------|--------------------------------|---|--|------|
| Logging type | Logging type | Continuous logging | | |
| | File format | CSV file | | |
| Sampling | Sampling interval | Time specification: 500 ms | | |
| Data | | Data register (D): 30 points Internal relay (M): 10 points | | |
| Save | Logging file save setting | Save destination in the SD memory card | LOG01 | |
| | | File name | Optional setting ^{*1} | |
| | File switching setting | Number of files to be saved | Number of files to be saved | 255 |
| | | | Operation when exceeds the number of files | Stop |
| | File switch timing | Condition specification ^{*2} | | |
| Logging operation | Operation at transition to RUN | Auto Start | | |

*1 The settings are as follows.

- Check "Device value".
- Device: D100
- Data type: String
- Number of digits/characters: 4
- Add date type: Date to establish file switching condition
- Format: YYYYMMDDhh

*2 The settings are as follows.

- Device: Clock data (Hour)
- Conditional formula: During changes
- Data type: Word [signed]

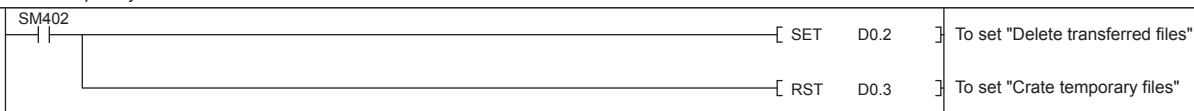
Program example

■Devices used

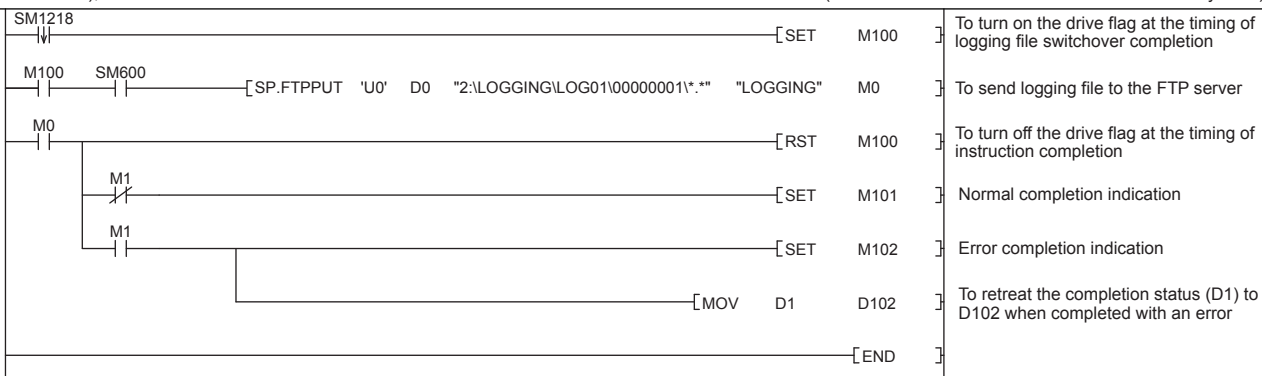
| Device No. | Application |
|------------|--|
| SM402 | After RUN, ON for one scan only |
| SM600 | Memory card usable |
| SM1218 | Data logging setting No.1 Logging data storage file switching in progress |
| D0 | Application setting area (SP.FTPPUT instruction control data) <ul style="list-style-type: none"> • D0.2: Transfer completion file delete setting • D0.3: Temporary file create setting |
| D1 | Completion status (SP.FTPPUT instruction control data) |
| D2 | Total number of files to be transferred (SP.FTPPUT instruction control data) |
| D3 | Number of transferred files (SP.FTPPUT instruction control data) |
| D102 | File transfer completion status |
| M0 | SP.FTPPUT instruction normal completion indication |
| M1 | SP.FTPPUT instruction error completion indication |
| M100 | SP.FTPPUT instruction drive flag |
| M101 | File transfer normal completion indication |
| M102 | File transfer error completion indication |

■Program

Set application to the control data.
b2: Transfer completion file delete: 1 = Delete
b3: Temporary file create: 0 = Create



When the SD memory card of the CPU module is available at the timing of logging file switchover completion of the logging setting No. 1 (at fall of SM1218), all files in "2:\LOGGING\LOG01\00000001\" are sent to "LOGGING" of the FTP server. (The sent files are deleted from the SD memory card)



Program example to transfer files one by one

Following is a program example to log 30 word device points and 10 bit device points in the binary format every 500 ms, and use the FTP client function to transfer files one by one to the FTP server every time 1200 records are logged. This program sequentially transfers the logging files (SP.FTPPUT (S2) operand) in the transfer logging files designated folder one by one. Even if the folder that stores the logging file is switched, it can be transferred continuously.

Setting in the external device (FTP server)


The FTP server's IP address is as follows.

| Item | Setting value |
|-------------|---------------|
| IP address | 192.168.3.101 |
| Subnet mask | 255.255.255.0 |
| Login name | user |
| Password | 1234abcd |

Setting in the CPU module (FTP client)


The FTP client's parameter settings are as follows.

■Own Node Settings

 Navigation window⇒[Parameter]⇒[FX5UCPU]⇒[Module Parameter]⇒[Ethernet Port]⇒[Basic Settings]⇒[Own Node Settings]

| Item | Setting value |
|-------------|---------------|
| IP Address | 192.168.3.250 |
| Subnet Mask | 255.255.255.0 |

■FTP Client Setting

 Navigation window⇒[Parameter]⇒[FX5UCPU]⇒[Module Parameter]⇒[Ethernet Port]⇒[Application Settings]⇒[FTP Client Settings]

| Item | Setting value |
|--|-----------------------|
| To Use or Not to Use FTP Client Settings | Use |
| FTP Server Specification | FTP Server IP Address |
| FTP Server IP Address | 192.168.3.101 |
| Login Name | user |
| Password | 1234abcd |
| Connection Method | PORT Mode |
| Port No. | 21 |

Logging settings

Logging files to be transferred to the FTP server are those files created in accordance with the following logging settings. If any item is not described here, set it optionally.

| Item | | Setting value | |
|-----------------------------|--------------------------------|---|-----------|
| Logging type | Logging type | Continuous logging | |
| | File format | Binary file | |
| Sampling | Sampling interval | Time specification: 500 ms | |
| Data | | Data register (D): 30 points Internal relay (M): 10 points | |
| Save | Logging file save setting | Save destination in the SD memory card | |
| | | LOG01 | |
| | File switching setting | File name | |
| | | Simple setting*1 | |
| Number of files to be saved | | Number of files to be saved | |
| | | 1024 | |
| | | Operation when exceeds the number of files | Overwrite |
| | File switch timing | Number of records: 1200 | |
| Logging operation | Operation at transition to RUN | Auto Start | |

*1 Following all check items are to be omitted.

- Folder name
- Date
- Times

Program example

■Devices used

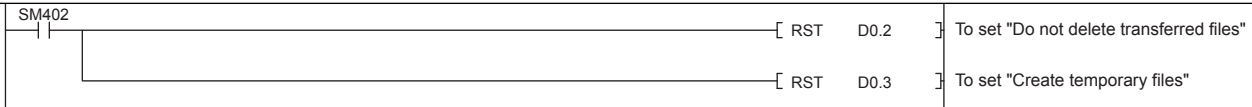
| Device No. | Application |
|----------------|--|
| SD1210 | Data logging setting No.1 Latest storage file number [Low-order] |
| SD1211 | Data logging setting No.1 Latest storage file number [High-order] |
| SM402 | After RUN, ON for one scan only |
| SM600 | Memory card usable |
| SM1218 | Data logging setting No.1 Logging data storage file switching in progress |
| D0 | Application setting area (SP.FTPPUT instruction control data) <ul style="list-style-type: none"> • D0.2: Transfer completion file delete setting • D0.3: Temporary file create setting |
| D1 | Completion status (SP.FTPPUT instruction control data) |
| D2 | Total number of files to be transferred (SP.FTPPUT instruction control data) |
| D3 | Number of transferred files (SP.FTPPUT instruction control data) |
| D102 | File transfer completion status |
| D200 to D238 | File names stored in the CPU module (transfer source) Example: "2:/LOGGING/LOG01/00000001\00000001.BIN"*1 |
| D1000 to D1001 | Calculation area to calculate a folder number |
| D1010 | Character string length of an ASCII code file name (transfer source) |
| D2000 to D2099 | Work area to edit an ASCII cod file name (transfer source) |
| M0 | SP.FTPPUT instruction normal completion indication |
| M1 | SP.FTPPUT instruction error completion indication |
| M100 | SP.FTPPUT instruction drive flag |
| M101 | File transfer normal completion indication |
| M102 | File transfer error completion indication |

*1 "00000001\00000001.BIN" will be updated according to values stored in SD1210 and SD1211.

■ Program

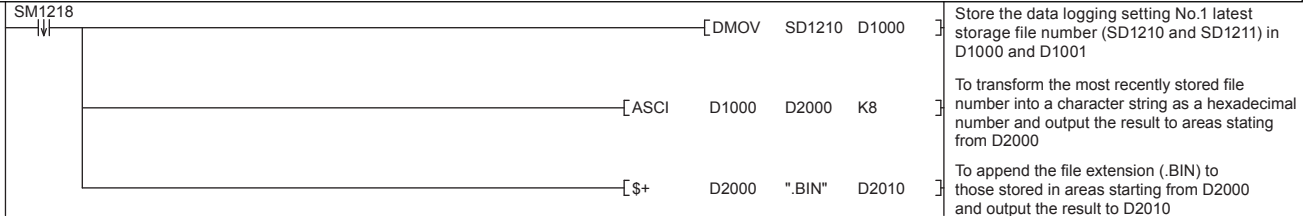
[Control data initialization]

Set application to the control data.
 b2: Transfer completion file delete: 0 = Do not delete
 b3: Temporary file create: 0 = Create

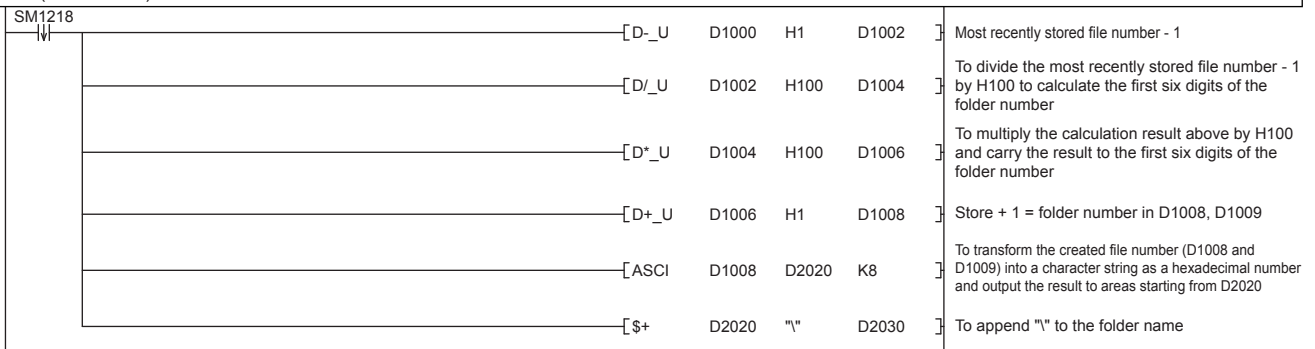


[Create transfer file name]

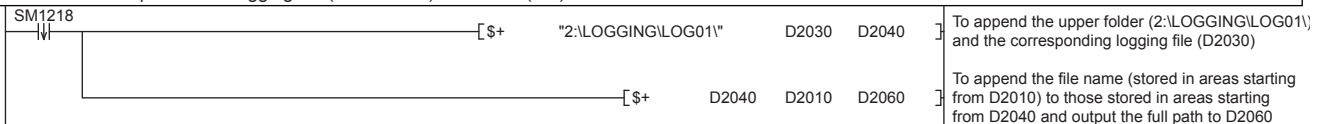
Create the file name (ASCII code) to D2010. (Ex.) "00000001.BIN"



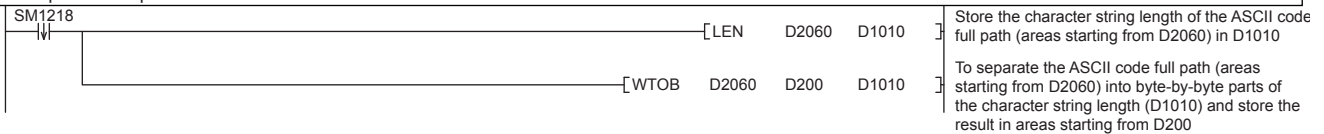
Create the logging file name (ASCII code) to D2030. (Ex.) "00000001"
 • Calculate the most recently stored file number of logging. (D1000 and D1001)
 • Calculation procedure:
 ① (Most recently stored file number - 1) / H100
 ② (Result of ①) × H100 + 1



Create the full path of the logging file (ASCII code) to D2060. (Ex.) "2:\LOGGING\LOG01\00000001\00000001.BIN"

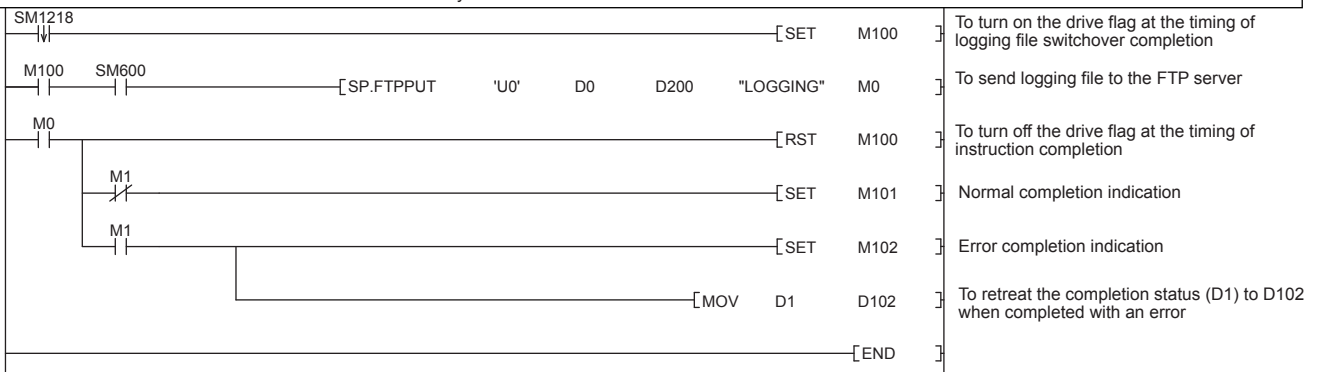


Output the full path in Unicode to D200.




[File transfer processing]

At the timing of logging file switchover completion of the logging setting No. 1 (at fall of SM1218), the files specified to areas starting from D200 are sent to "LOGGING" of the FTP server from the SD memory card of the CPU module.



Example of program for obtaining files from FTP server

Below is shown an example of a program for storing the CSV data obtained from the FTP server in D1000 to D1999 of the CPU module by using the SP.FREAD instruction (file operation instruction). For the details of the file operation instruction, refer to  MELSEC iQ-F FX5 Programming Manual (Instructions, Standard Functions/Function Blocks).

Setting in the external device (FTP server)


The FTP server's IP address is as follows.

| Item | Setting value |
|-------------|---------------|
| IP address | 192.168.3.101 |
| Subnet mask | 255.255.255.0 |
| Login name | user |
| Password | 1234abcd |

Setting in the CPU module (FTP client)


The FTP client's parameter settings are as follows.

■Own Node Settings

 Navigation window⇒[Parameter]⇒[FX5UCPU]⇒[Module Parameter]⇒[Ethernet Port]⇒[Basic Settings]⇒[Own Node Settings]

| Item | Setting value |
|-------------|---------------|
| IP Address | 192.168.3.250 |
| Subnet Mask | 255.255.255.0 |

■FTP Client Setting

 Navigation window⇒[Parameter]⇒[FX5UCPU]⇒[Module Parameter]⇒[Ethernet Port]⇒[Application Settings]⇒[FTP Client Settings]

| Item | Setting value |
|--|-----------------------|
| To Use or Not to Use FTP Client Settings | Use |
| FTP Server Specification | FTP Server IP Address |
| FTP Server IP Address | 192.168.3.101 |
| Login Name | user |
| Password | 1234abcd |
| Connection Method | PORT Mode |
| Port No. | 21 |

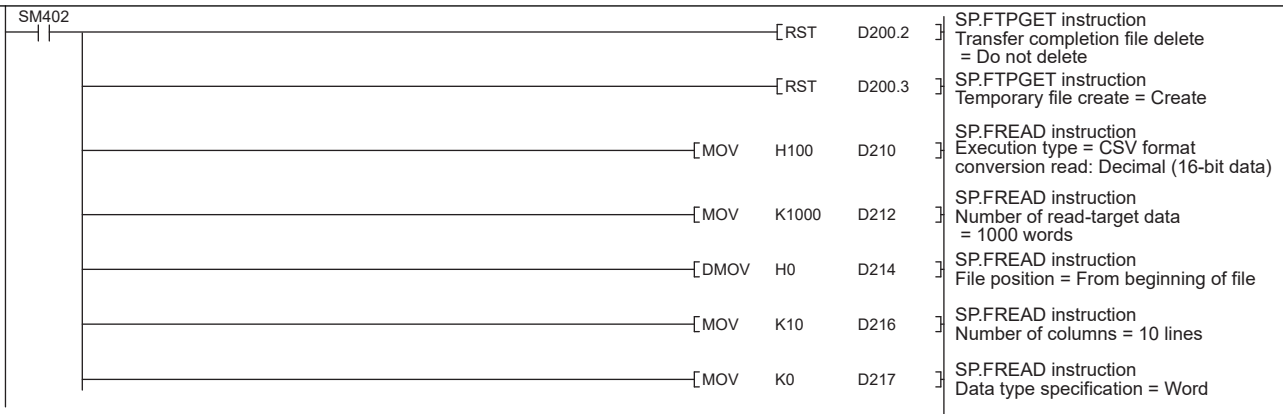
Program example

■Devices used

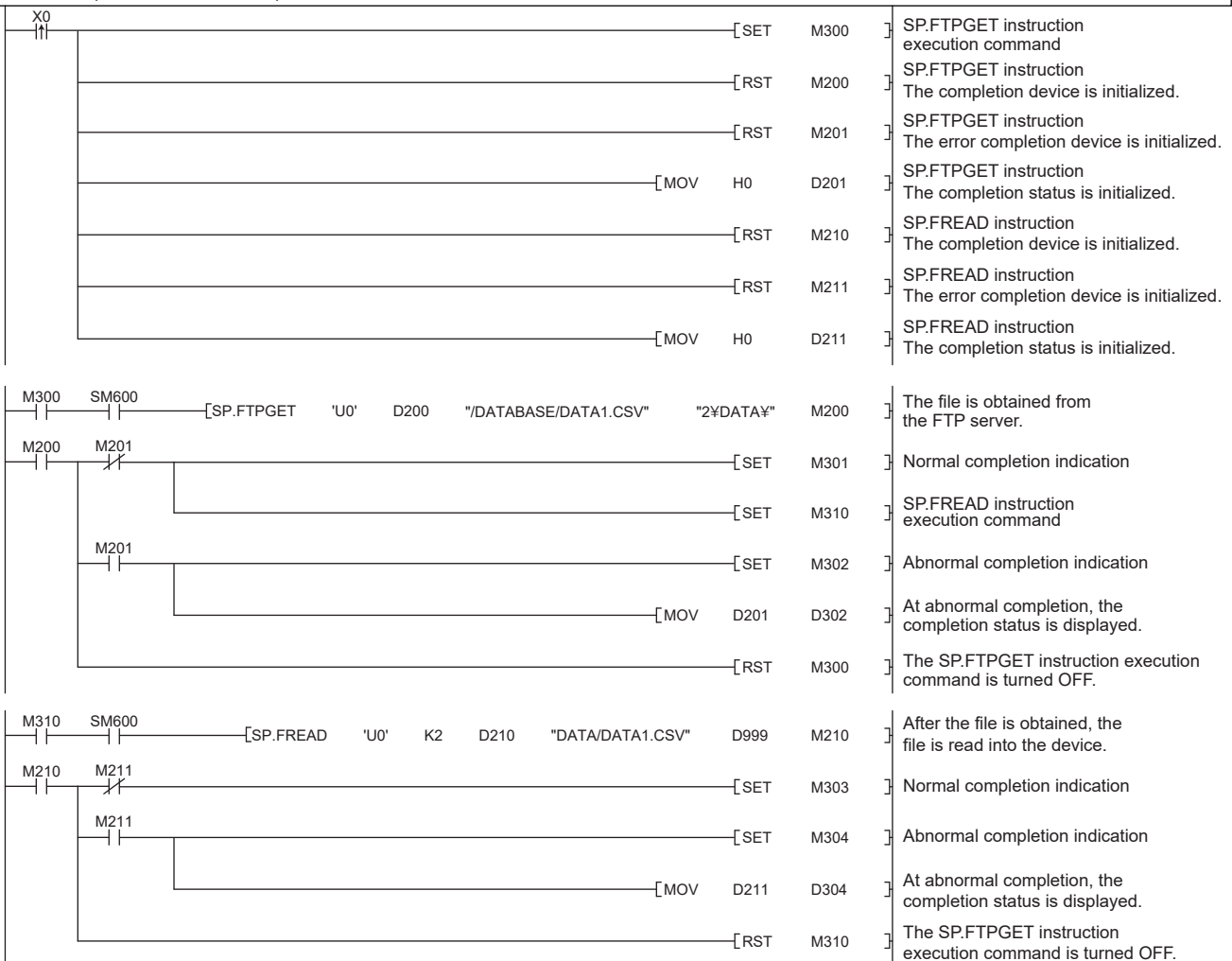
| Device No. | Application |
|---|--|
| SM402 | Only 1 scan is ON after RUN operation. |
| SM600 | Memory card usable |
| ■Devices used for SP.FTPGET instruction | |
| M300 | SP.FTPGET instruction execution command |
| D200 | Use setting area (SP.FTPGET instruction control data) <ul style="list-style-type: none"> • D200.2: Select "0: Do not delete" for the transferred file deletion setting. • D200.3: Select "0: Create" for the setting for creating temporary files when transferring files. |
| D201 | Completion status (SP.FTPGET instruction control data) |
| D202 | Total number of files to be processed (SP.FTPGET instruction control data) |
| D203 | Number of processed files (SP.FTPGET instruction control data) |
| M200 | SP.FTPGET instruction completion device |
| M201 | SP.FTPGET instruction error completion device |
| M301 | Indication of normal completion of file transfer |
| M302 | Indication of abnormal completion of file transfer |
| D302 | File transfer completion status |
| ■Devices used for SP.FREAD instruction | |
| M310 | SP.FREAD instruction execution command |
| D210 | Execution type (SP.FREAD instruction control data) <ul style="list-style-type: none"> • CSV format conversion read: Decimal (16-bit data) |
| D211 | Completion status (SP.FREAD instruction control data) |
| D212 | Number of read-target data (SP.FREAD instruction control data) <ul style="list-style-type: none"> • 1000 words |
| D213 | Maximum number of read data (SP.FREAD instruction control data) <ul style="list-style-type: none"> • 0 (fixed) |
| D214, D215 | File position (SP.FREAD instruction control data) <ul style="list-style-type: none"> • From beginning of file |
| D216 | Number of columns (SP.FREAD instruction control data) <ul style="list-style-type: none"> • 10 lines |
| D217 | Data type specification (SP.FREAD instruction control data) <ul style="list-style-type: none"> • Word |
| D999 | Number of data read by SP.FREAD instruction |
| D1000 to D1999 | Data read by SP.FREAD instruction |
| M210 | SP.FREAD instruction completion device |
| M211 | SP.FREAD instruction error completion device |
| M303 | Indication of normal completion of file reading |
| M304 | Indication of abnormal completion of file reading |
| D304 | File reading completion status |

Program

"DATA1.CSV" file is obtained from "DATABASE" in the FTP server and uncompressed into D1000 and following.
 Set the application for the control data.
 SP.FTPGET instruction
 b2: Delete transfer complete file: 0 = Do not delete
 b3: Create temporary file: 0 = Create
 SP.FREAD instruction
 +0: Execution type: 0100H = Read CSV format conversion, decimal (16-bit data)
 +2: Number of read-target data: K1000 = 1000 words
 +3: Maximum number of read data: K0 (fixed)
 +4: File position: 00000000H = From head of file
 +6: Number of columns: K10 = 10 lines
 +7: Data type specification: K0 = word



Startup of X0 triggers file transfer.
 The completion device and completion status are initialized before execution of the instruction.



9.4 Precautions

Connection with the FTP server

If a response to an alive check request is not returned from the FTP server, the CPU module (FTP client) determines that the FTP server is not alive and disconnects the connection.

The connection may also be disconnected if the FTP server does not support the TCP KeepAlive function (response to a KeepAlive ACK message).

RUN→STOP during file transfer

If the CPU module' status is changed from RUN to STOP during the file transfer, the module continues its operation until the file transfer is completed. (This is also applicable to cases when wild cards (*, ?) are designated)

Execution of the SD memory card forced disable function

If the SD memory card forced disable function is executed, the SD card will be disabled during the file transfer processing, causing an immediate stop of the processing. Check that the transfer processing has been completed, and execute the function.

Execution of a file transfer function instruction during execution of another function

While the backup/restore functions of the CPU module are being executed, the file transfer instruction cannot be performed. Perform the file transfer instruction after having confirmed that the backup/restore functions are not executed.

File operation from the external device during file transfer

Do not manipulate the files from an external device, such as an engineering tool, while transferring the files.

If the file is manipulated from another device while transferring the files, the external device may issue an error. If the processing has been halted due to an error, re-execute the processing after the file transfer operation.

Simultaneous use with FTP server function

Do not use the FTP client and FTP servers simultaneously.

Communication processing

If another communication function (MELSOFT connection or SLMP) is simultaneously executed with UDP/IP during file transfer with FTP, an error such as timeout may occur. Either communicate after the file is transferred, or communicate with TCP/IP.

Port numbers used by other functions

The FTP client function uses the own station port numbers of 62000 to 65534 (F230H to FFFE_H). Do not use the port numbers of 62000 to 65534 (F230H to FFFE_H) for other functions. If the port numbers are used, the file transfer function may not operate normally.

File transfer time

If the FTP client function transfers a file whose size is large, it may take long to transfer the file.

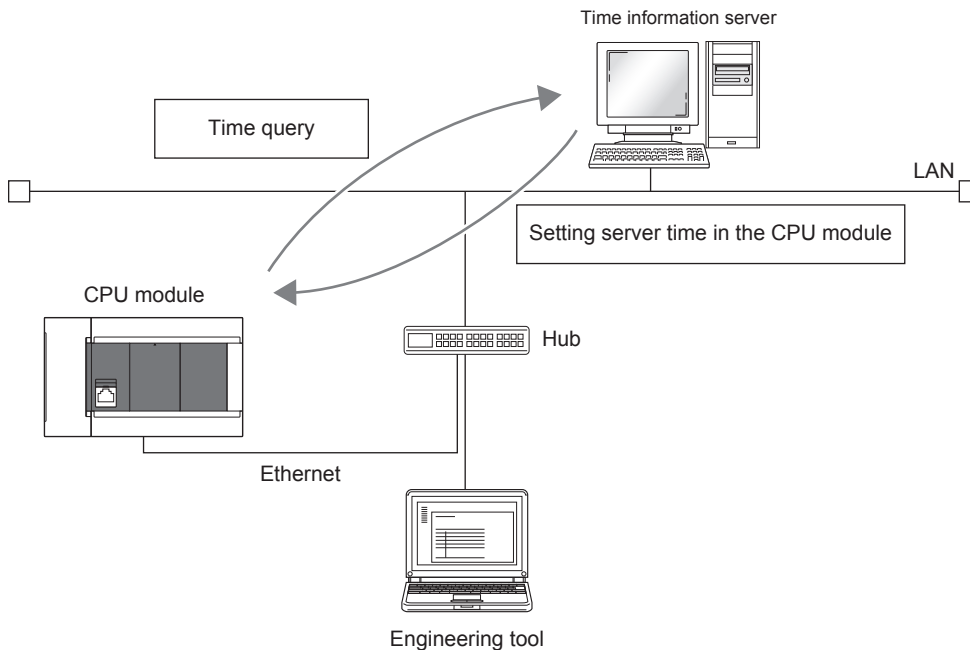
(Example) Scan time: 5 ms, file size: 16MB file transfer time (SP.FTPPUT instruction): Approx. 548 s

Deleting unnecessary files

If a cable is disconnected, system is powered off, or the CPU module is reset during the file transfer, delete unnecessary files (such as a temporary file and undefined files) in the FTP server. Then, transfer files again.

10 TIME SETTING FUNCTION (SNTP CLIENT)

Time information is collected from the time information server (SNTP server) connected on the LAN at the specified timing, and the CPU module's time is automatically set.



Point

An SNTP server (time information server) must be provided on the LAN line to use this function.

Time setting execution timing

Time setting is executed at the following timing.

- Switching power from OFF to ON, and when resetting the CPU module.
- At each set time (periodic execution)
- At set time (execution at set time)
- At programmed optional timing^{*1}

*1 By turning on the time synchronization (SNTP) execution request (SD10299.0), execute the time setting.

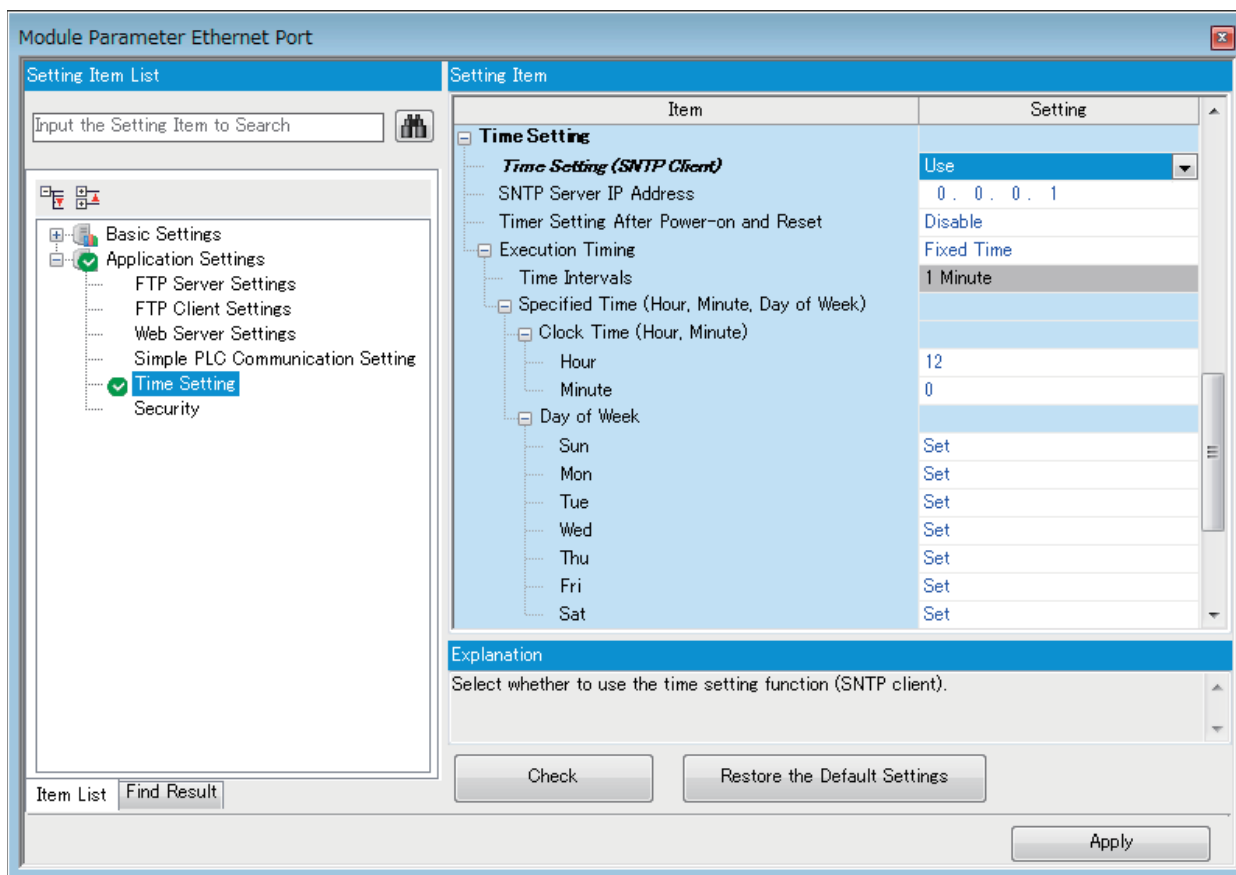
Point

When setting the time during powering on or resetting the CPU module, check the hub or external device connection before setting.

Setting procedure

The following shows time setting function (SNTP client).

Navigation window ⇒ [Parameter] ⇒ [FX5UJCPU] or [FX5UCPU] ⇒ [Module Parameter] ⇒ [Ethernet Port] ⇒ [Application Settings] ⇒ [Time Setting]




| Item | Description | Setting range |
|--|---|--|
| Time Setting (SNTP Client) | Select whether to use the time setting function (SNTP client). | <ul style="list-style-type: none"> • Not Use • Use (Default: Not Use) |
| SNTP Server IP Address | Set the IP address of the SNTP server. | 0.0.0.1 to 223.255.255.254 (Default: 0.0.0.1) |
| Timer Setting After Power-on and Reset | Select whether to execute the time setting function upon power-on or reset. | <ul style="list-style-type: none"> • Disable • Enable (Default: Disable) |
| Execution Timing | Select the execution timing of the time setting. | <ul style="list-style-type: none"> • Fixed Time • Fixed Scan Interval (Default: Fixed Time) |
| | Time Intervals | When "Fixed Scan Interval" is selected, set the time interval (minute) of time setting executions. 1 to 1440 (Default: 1 Minute) |
| | Specified Time (Hour, Minute, Day of Week) | When "Fixed Time" is selected, set the day of the week and the clock time (hour and minute) for the time setting to be executed. <ul style="list-style-type: none"> • Hour: 0 to 23 (Default: 12) • Minute: 0 to 59 (Default: 0) • Day*1 |

*1 To specify the day of the week for the time setting to be executed, set the day, for the time setting not to be executed, under "Day of Week" to "Not Set". (Time setting is set to be executed every day (all the days are set to "Set") by default.)
When specifying the day of the week, set at least one day of the week to "Set". An error occurs when all the days are set to "Not Set".

Point

The SNTP server must be only one in a network. The time to be output is the same even though multiple modules in the same system retrieve time from the same SNTP server.

Confirming the execution results


The time setting execution results can be checked with the following special device. For details, refer to  Page 270 List of Special Device Applications and Assignments.

- Time setting function operation result (SD10290)
- Time setting function execution time (SD10291 to SD10297)
- Time setting function required response time (SD10298)

Precautions

■Communication timeout

If a response is not received from the SNTP server (time information sever) 20 seconds after the time setting is executed, the communication times out. An error does not occur when the communication times out. Instead, the timeout occurrence appears in the event history.

Also, when performing the following setting, the communication times out. ( Page 192 IP Filter Function)

- Not set the SNTP server address to the penetration address of the IP filter.
- Set the SNTP server address to the exclusion address of the IP filter.

■Time information server

SNTP server on a LAN connecting the CPU module is required to use this function.

■Delay by communication time

The time set in the time setting function is calculated according to the SNTP specification, and the CPU module calculates the time in consideration of the communication time with the SNTP server. This calculation method is based on the assumption that the upward and downward communication time are the same, therefore, if there is a great difference between the upward and downward communication time, an error occurs. When setting the time setting in a high accuracy, specify the SNTP server as close as possible to the CPU module on the network.

■Setting the execution time

The execution time can be set in the range of 1980 to 2079.

11 WEB SERVER FUNCTION

For details of Web server function, refer to  MELSEC iQ-R/MELSEC iQ-F Web Server Function Guide Book.

This manual describes the JavaScript objects and CGI objects that can be used in the user Web page (User-customized Web page).

Restriction

The user Web page is available only in the FX5U/FX5UC CPU module.

11

11.1 JavaScript Objects

JavaScript objects in the library of user Web pages enable device data to be read or written easily on the user Web page. The following table lists the objects in the JavaScript object library for the user Web page (FUserWebLib.js).

| Object name (function name) | Description | Reference |
|-----------------------------|---|--|
| Data block (WSDatblk) | Displays the specified device data in a table format. | Page 168 Data block (WSDatblk) |
| Level display (WSLevel) | Displays what percentage the device value accounts for of the whole (the range between the upper and lower limit values). | Page 170 Level display (WSLevel) |
| Figure display (WSFigure) | Displays the specified figure when the device value reaches the set value. | Page 172 Figure display (WSFigure) |
| Image display (WSPicture) | Displays the specified image file when the device value reaches the set value. | Page 174 Image display (WSPicture) |
| Historical graph (WSHstgrp) | Displays the device data in a time-series line graph. | Page 176 Historical graph display (WSHstgrp) |
| Write button (WSWrtBtn) | Writes the specified value to the specified device. | Page 178 Write button (WSWrtBtn) |
| Logout button (WSLogoutBtn) | Performs the logout operation. | Page 179 Logout button (WSLogoutBtn) |

Common information on JavaScript Objects

Style sheet

Descriptions for the styles such as character size, color, line color, the background color of the user Web page can be omitted. If descriptions are omitted or incorrect, the following default display according to the style sheet (UserWebStyle.css) is applied.

| Item | Default |
|-----------------------------|--|
| Character color | Black |
| Background color | White |
| Line color | Black |
| Fill color of level display | Blue |
| Graph line color | Blue |
| Character size | 20 |
| Button character color | According to the setting of the Web browser. |
| Button background color | According to the setting of the Web browser. |
| Button line color | According to the setting of the Web browser. |

The character font differs depending on the terminal used to display the user Web page.

Restriction

When editing the style sheet in the library of the user Web page, do not change the class names in the style sheet. In addition, do not define the style classes with the same name.

Parameter settings of JavaScript objects

- When an optional parameter setting is omitted, the object will be displayed as default. Also, when the optional parameter setting is abnormal, the object will be displayed as default.
- Set parameters using the specified data formats. A parameter error occurs if a parameter is set in the data format other than the specified one (for example, when a character string such as '1' is set in the parameter that should be a numeric value).
- If the X- and Y- coordinates are omitted, the object will be placed at the upper left end (coordinates 0, 0).
- For an RGB value specified for parameters or HTML specifications such as a color name, the range check is not performed. The operation for an abnormal setting differs depending on the browser.
- When specifying a device name which includes \, such as U□\G□, repeat \ as U□\\G□. (\ is an escape sequence.)
- The following table lists formats and positional notations available for the devices which can be set for JavaScript objects.

| Classification | Type | Device name | | Format*1 | | | | |
|------------------------------|---------------------|--------------------------|--------------------|------------------------|------------------------|------------------------------|------|---|
| | | | | 16-bit signed/unsigned | 32-bit signed/unsigned | Single-precision real number | Bit | |
| User device | Bit | Input (X) | | × | × | × | ○*2 | |
| | | Output (Y) | | × | × | × | ○*2 | |
| | | Internal relay (M) | | × | × | × | ○ | |
| | | Latch relay (L) | | × | × | × | ○ | |
| | | Link relay (B) | | × | × | × | ○ | |
| | | Annunciator (F) | | × | × | × | ○ | |
| | | Link special relay (SB) | | × | × | × | ○ | |
| | | Step relay (S) | | × | × | × | ○ | |
| | | Timer (T)*3 | Contact: TS | × | × | × | ○ | |
| | | | Coil: TC | × | × | × | ○ | |
| | | Retentive timer (ST)*3 | Contact: STS | × | × | × | ○ | |
| | | | Coil: STC | × | × | × | ○ | |
| | | Counter (C)*3 | Contact: CS | × | × | × | ○ | |
| | | | Coil: CC | × | × | × | ○ | |
| | Long counter (LC)*3 | Contact: LCS | × | × | × | ○ | | |
| | | Coil: LCC | × | × | × | ○ | | |
| | Word | Timer (T)*3 | | Current value: TN | ○: K/H | × | × | × |
| | | Retentive timer (ST)*3 | | Current value: STN | ○: K/H | × | × | × |
| | | Counter (C)*3 | | Current value: CN | ○: K/H | × | × | × |
| | | Data register (D) | | | ○: K/H | ○: K/H | ○: K | × |
| Link register (W) | | | ○: K/H | ○: K/H | ○: K | × | | |
| Link special register (SW) | | | ○: K/H | ○: K/H | ○: K | × | | |
| Double word | Long counter (LC)*3 | | Current value: LCN | × | ○: K/H | × | × | |
| System device | Bit | Special relay (SM) | | × | × | × | ○ | |
| | Word | Special register (SD) | | ○: K/H | ○: K/H | ○: K | × | |
| Module access device (U□\G□) | Word | Module access device (G) | | ○: K/H | ○: K/H | ○: K | × | |
| Index register | Word | Index register (Z) | | ○: K/H | ○: K/H | ○: K | × | |
| | Double word | Long index register (LZ) | | × | ○: K/H | ○: K | × | |
| File register | Word | File register (R) | | ○: K/H | ○: K/H | ○: K | × | |

○: Can be set, ×: Cannot be set

*1 K: Decimal, H: Hexadecimal

*2 When specifying X and Y, specify in octal.

*3 When T, ST, C, or LC is specified, the device is treated as the device of current value (TN, STN, CN, or LCN).

Refreshing cycle

Set the refreshing cycle of the user Web page to the variable "updateinterval" in HTML. This refreshing cycle applies to all objects in the user Web page. When the variable "updateinterval" is not described in HTML or it is out of range between 5 and 120, 5-second interval (default) applies to the operation.

Ex.

When refreshing the user Web page in 10-second interval

```
<script>
  var updateInterval = 10;
</script>
```

Message display language

Set the message display language to the variable "dspLanguage" in HTML. The following table lists the settings of the variable "dspLanguage" and message display languages.

| dspLanguage | Message display language |
|-------------------------------------|--------------------------|
| No description | English |
| ja-JP | Japanese |
| en-US | English |
| zh-CN | Chinese (Simplified) |
| Other than the above (out of range) | English |

Available files

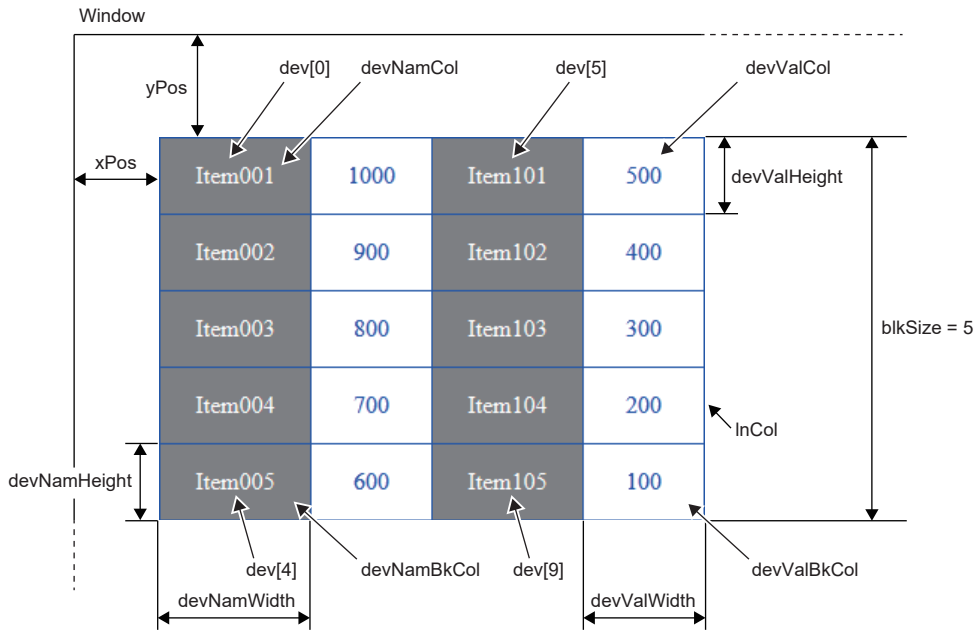
The following files are available in the user Web page.

| File | Extension | MIME type |
|----------------|-------------|-----------------|
| HTML | .html | text/html |
| | .htm | text/htm |
| JavaScript | .js | text/javascript |
| CSS | .css | text/css |
| GIF image | .gif | image/gif |
| PNG image | .png | image/png |
| JPG/JPEG image | .jpg(.jpeg) | image/jpeg |

Data block (WSDatblk)

This object displays the specified device data in a table format. The current values of the devices are displayed as center aligned and updated in a fixed interval.

■ direction = 0 (Vertical)



■ direction = 1 (Horizontal)

| | | | | |
|---------|---------|---------|---------|---------|
| Item001 | Item002 | Item003 | Item004 | Item005 |
| 1000 | 900 | 800 | 700 | 600 |
| Item101 | Item102 | Item103 | Item104 | Item105 |
| 500 | 400 | 300 | 200 | 100 |

blkSize = 5

Parameter

Set dev[n] according to the number of devices to be displayed. (n: 0 to 31)

| Element name | Item | Description | Requirement |
|----------------|--|--|----------------------------|
| dev[0].dsp | Device display name 1 | A name to be displayed with the data block | ○*1 |
| dev[0].name | Device name 1 | Device type + device number | ○ |
| dev[0].base | The positional notation of the device 1 | K: Decimal H: Hexadecimal B: Binary | ○ |
| dev[0].format | The data format of the device 1 | 0: 16-bit signed 1: 16-bit unsigned 2: 32-bit signed 3: 32-bit unsigned 4: Single-precision real number 6: Bit | ○ |
| ⋮ | | | |
| dev[31].dsp | Device display name 32 | A name to be displayed with the data block | — |
| dev[31].name | Device name 32 | Device type + device number | — |
| dev[31].base | The positional notation of the device 32 | K: Decimal H: Hexadecimal B: Binary | — |
| dev[31].format | The data format of the device 32 | 0: 16-bit signed 1: 16-bit unsigned 2: 32-bit signed 3: 32-bit unsigned 4: Single-precision real number 6: Bit | — |
| direction | Display direction | 0: Vertical 1: Horizontal | — (If omitted, it is 0) |
| blkSize | Block size | 1 to 32 (The number of lines is displayed when the display direction is set to vertical, and the number of columns is displayed when the display direction is set to horizontal.) | ○ |
| devNamDisp | Device name display | 0: Not display 1: Display | — (If omitted, it is 1) |
| devNamCol | Character color of device name | The RGB value or color name Example: #FF0000 (RGB value) or red (color name) | — |
| devNamBkCol | Background color of device name | | |
| devNamWidth | Cell width of device name | Positive real numbers (unit: px) (For the horizontal display direction, the setting value of the cell width of device value takes priority, and for the vertical display direction, the setting value of the cell height of device value does.) | ○*1 |
| devNamHeight | Cell height of device name | | |
| devValCol | Character color of device value | The RGB value or color name | — |
| devValBkCol | Background color of device value | | |
| devValWidth | Cell width of device value | Positive real numbers (unit: px) | ○ |
| devValHeight | Cell height of device value | | |
| InCol | Line color | The RGB value or color name | — |
| xPos | X-coordinate | The horizontal coordinate of the data block at the upper left end (unit: px) | — |
| yPos | Y-coordinate | The vertical coordinate of the data block at the upper left end (unit: px) | — |

○: Required, —: Optional

*1 When the device name is not displayed (devNamDisp = 0), a setting is not required.

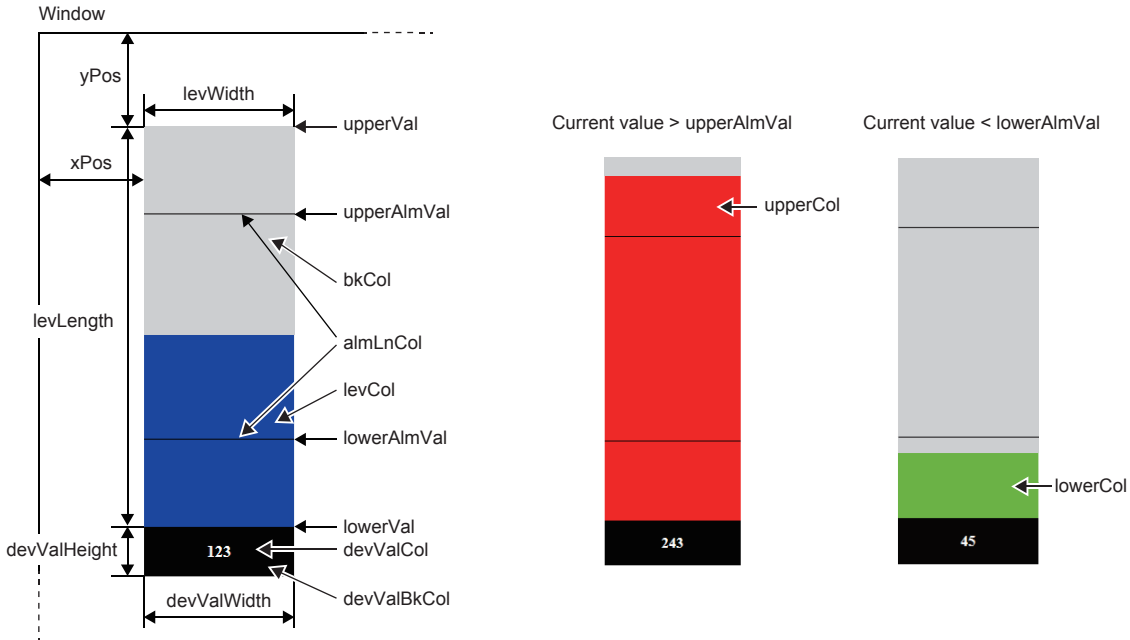
Precautions

- Write all the necessary parameters. If the necessary parameters are not written or the setting value is out of range, an error occurs.
- If the data format is the single-precision real number, the positional notation will be the decimal.

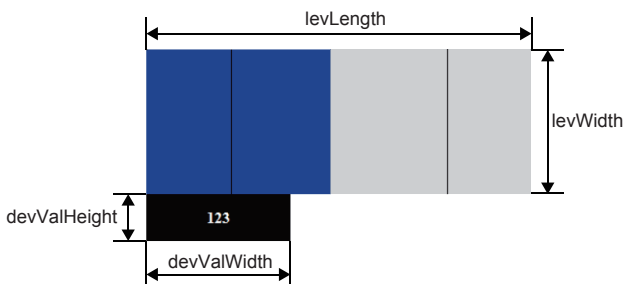
Level display (WSLevel)

This object displays the current value of the specified device as a level for the range between the upper and lower limit values. A different display color can be assigned in each case when the current value is above the upper limit value and when the current value is below the lower limit value. The current values of the devices are updated in the fixed interval.

■ direction = 0 (Vertical)



■ direction = 1 (Horizontal)



Parameter

| Element name | Item | Description | Requirement |
|--------------|--|---|--|
| devName | Device name | Device type + device number | ○ |
| direction | Level direction | 0: Vertical 1: Horizontal | — (If omitted, it is 0) |
| upperVal | Upper limit value | A numeric value (decimal) within the range of the specified data format. | ○ |
| lowerVal | Lower limit value | | |
| upperAlmVal | Upper limit of alarm value | | |
| lowerAlmVal | Lower limit of alarm value | | |
| dspAlmLn | Display of alarm value line | | |
| dspVal | Display of current value | | |
| valFormat | Data format | 0: 16-bit signed 1: 16-bit unsigned 2: 32-bit signed 3: 32-bit unsigned 4: Single-precision real number | — (If omitted, it is 0) |
| levCol | Level display color | The RGB value or color name | — |
| upperCol | Display color when the current value is above the upper limit of alarm value | The RGB value or color name | — (If omitted, it is the same as the level display color) |
| lowerCol | Display color when the current value is below the lower limit of alarm value | | |
| bkCol | Background color | The RGB value or color name | — |
| almLnCol | Line color of alarm value | | |
| levLength | Level length | Positive real numbers (unit: px) | ○ |
| levWidth | Level width | | |
| xPos | X-coordinate | The horizontal coordinate of the level display at the upper left end (unit: px) | — |
| yPos | Y-coordinate | The vertical coordinate of the level display at the upper left end (unit: px) | — |
| devValWidth | Cell width of current value | Positive real numbers (unit: px) | ○ |
| devValHeight | Cell height of current value | | |
| devValCol | Character color of current value | The RGB value or color name | — |
| devValBkCol | Background color of current value | | |

○: Required, —: Optional

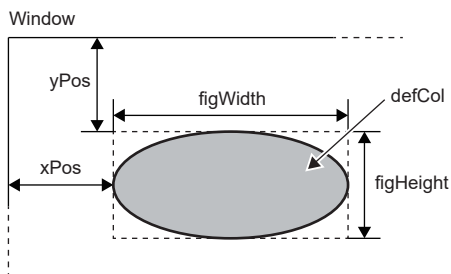
Precautions

Write all the necessary parameters. If the necessary parameters are not written or the setting value is out of range, an error occurs.

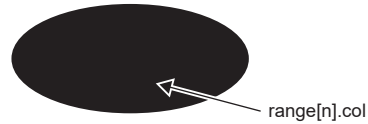
Figure display (WSFigure)

When the device value is within the specified range, a figure is displayed in the specified color. The device value is monitored in the fixed interval.

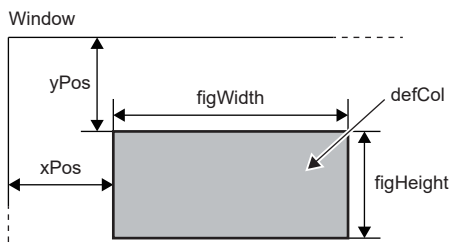
■ figType = 'Oval' (oval)



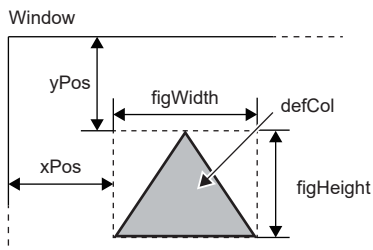
$\text{range}[n].\text{high} \geq \text{Current value} \geq \text{range}[n].\text{low}$



■ figType = 'Rect' (rectangle)



■ figType = 'Tri' (triangle)



When a negative value is set for the figure height (figHeight), the downward triangle is displayed.

Parameter

Set range[n] according to the number of display color ranges to be distinguished by the different colors. (n: 0 to 4)

| Element name | Item | Description | Requirement |
|---------------|---|--|-------------|
| devName | Device name | Device type + device number | ○ |
| devFormat | The data format of the device | 0: 16-bit signed 1: 16-bit unsigned 2: 32-bit signed 3: 32-bit unsigned 4: Single-precision real number 6: Bit | ○ |
| figType | Figure type | Oval: Oval (When specifying a circle, set the figure height = width) Rect: Rectangle (When specifying a square, set the figure height = width) Tri: Triangle | ○ |
| figHeight | Figure height | Real numbers (unit: px) | ○ |
| figWidth | Figure width | When a negative value is set, the value is treated as an absolute value. | |
| defCol | Default display color | The RGB value or color name (This item is displayed when the device value is out of the setting range.) | ○ |
| rangeNum | Number of setting ranges | 1 to 5 | ○ |
| range[0].low | Lower limit of setting range 1 | A numeric value (decimal) within the range of the specified data format | ○ |
| range[0].high | Upper limit of setting range 1 | | |
| range[0].col | Display color when device value is within setting range 1 | The RGB value or color name | ○ |
| range[1].low | Lower limit of setting range 2 | A numeric value (decimal) within the range of the specified data format | — |
| range[1].high | Upper limit of setting range 2 | | |
| range[1].col | Display color when device value is within setting range 2 | The RGB value or color name | — |
| ⋮ | | | |
| range[4].low | Lower limit of setting range 5 | A numeric value (decimal) within the range of the specified data format | — |
| range[4].high | Upper limit of setting range 5 | | |
| range[4].col | Display color when device value is within setting range 5 | The RGB value or color name | — |
| xPos | X-coordinate | The horizontal coordinate of the figure display at the upper left end (unit: px) | — |
| yPos | Y-coordinate | The vertical coordinate of the figure display at the upper left end (unit: px) | |

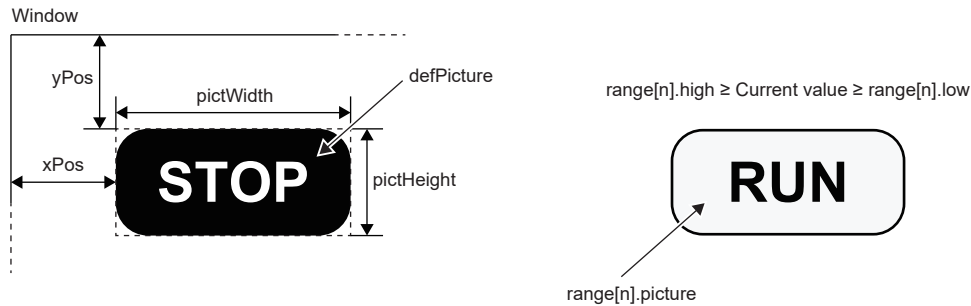
○: Required, —: Optional

Precautions

- Write all the necessary parameters. If the necessary parameters are not written or the setting value is out of range, an error occurs.
- If the setting ranges overlap, the figure with the lower number of the setting range is displayed. For example, when the device value is in both of the setting range 1 and 2, the figure of the setting range 1 is displayed.

Image display (WSPicture)

When the device value is within the specified range, the specified image file is displayed. The device value is monitored in the fixed interval.



Parameter

Set range[n] according to the number of ranges of images to be displayed. (n: 0 to 4)

| Element name | Item | Description | Requirement |
|------------------|--|---|-------------|
| devName | Device name | Device type + device number | ○ |
| devFormat | The data format of the device | 0: 16-bit signed 1: 16-bit unsigned 2: 32-bit signed 3: 32-bit unsigned 4: Single-precision real number 6: Bit | ○ |
| pictHeight | Display range height of image file | Real numbers (unit: px) | ○ |
| pictWidth | Display range width of image file | When a negative value is set, the value is treated as an absolute value. | ○ |
| defPicture | Name of image file to be displayed as default | For a name of image file, only ASCII character string can be used. Extension: jpg, jpeg, gif, png (This item is displayed when the device value is out of the setting range.) | ○ |
| rangeNum | Number of setting ranges | 1 to 5 | ○ |
| range[0].low | Lower limit of setting range 1 | A numeric value (decimal) within the range of the specified data format | ○ |
| range[0].high | Upper limit of setting range 1 | | |
| range[0].picture | Name of image file displayed when device value is within setting range 1 | Extension: jpg, jpeg, gif, png | ○ |
| range[1].low | Lower limit of setting range 2 | A numeric value (decimal) within the range of the specified data format | — |
| range[1].high | Upper limit of setting range 2 | | |
| range[1].picture | Name of image file displayed when device value is within setting range 2 | Extension: jpg, jpeg, gif, png | — |
| ⋮ | | | |
| range[4].low | Lower limit of setting range 5 | A numeric value (decimal) within the range of the specified data format | — |
| range[4].high | Upper limit of setting range 5 | | |
| range[4].picture | Name of image file displayed when device value is within setting range 5 | Extension: jpg, jpeg, gif, png | — |
| xPos | X-coordinate | The horizontal coordinate of the figure display range at the upper left end (unit: px) | — |
| yPos | Y-coordinate | The vertical coordinate of the figure display range at the upper left end (unit: px) | — |

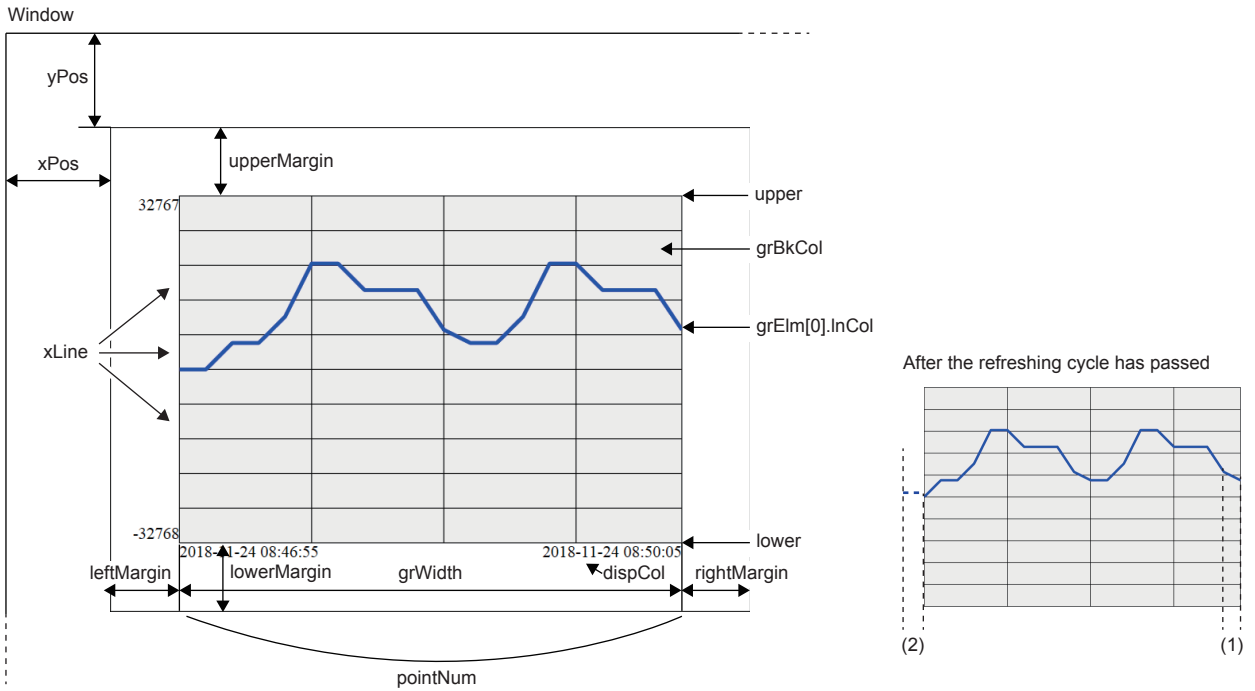
○: Required, —: Optional

Precautions

- Write all the necessary parameters. If the necessary parameters are not written or the setting value is out of range, an error occurs.
- The total size of the image file on one screen should be 100K bytes or smaller.
- If the setting ranges overlap, the image with the lower number of the setting range is displayed. For example, when the device value is in both of the setting ranges 1 and 2, the figure of the setting range 1 is displayed.

Historical graph display (WSHstgrp)

This object displays the device value in a time-series line graph. The device value is read in every refreshing cycle. After the number of records is reached, the display shifts to left and old records are deleted.



- (1) Refreshing cycle
- (2) Data in a refreshing cycle is out of the display range.

Parameter

Set grElm[n] according to the number of devices to be displayed. (n: 0 to 31)

| Element name | Item | Description | Requirement |
|-------------------|---------------------------------|---|----------------------------|
| grElmNum | Number of graph elements | 1 to 32 | ○ |
| devFormat | The data format of the device | 0: 16-bit signed 1: 16-bit unsigned 2: 32-bit signed 3: 32-bit unsigned 4: Single-precision real number 6: Bit | ○ |
| grElm[0].devName | Device name | Device type + device number | ○ |
| grElm[0].InCol | Graph line color | The RGB value or color name Example: #FF0000 (RGB value) or red (color name) | — |
| grElm[1].devName | Device name | Device type + device number | — |
| grElm[1].InCol | Graph line color | The RGB value or color name | — |
| ⋮ | | | |
| grElm[31].devName | Device name | Device type + device number | — |
| grElm[31].InCol | Graph line color | The RGB value or color name | — |
| grBkCol | Graph background color | The RGB value or color name | — |
| dspCol | Character color | | |
| pointNum | Number of records | 5 to 100 | ○ |
| upper | Upper limit value | A numeric value (decimal) within the range of the specified data format | ○ |
| lower | Lower limit value | | |
| yLine | Vertical axis interval (record) | 0 to 99 | ○ |
| xLine | Number of horizontal axes | | |
| grHeight | Graph height | Positive real numbers (unit: px) | ○ |
| grWidth | Graph width | | |
| xPos | X-coordinate | The horizontal coordinate of the historical graph display at the upper left end (unit: px) | — |
| yPos | Y-coordinate | The vertical coordinate of the historical graph display at the upper left end (unit: px) | |
| rightMargin | Right margin | Positive real numbers (unit: px) | — (If omitted, it is 0) |
| leftMargin | Left margin | | |
| upperMargin | Upper margin | | |
| lowerMargin | Lower margin | | |

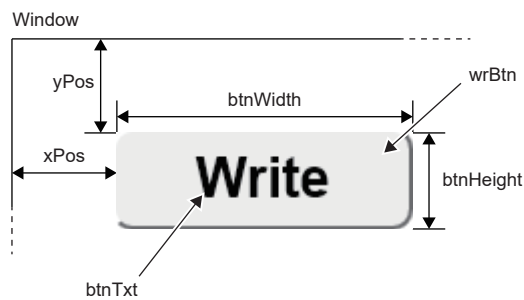
○: Required, —: Optional

Precautions

- Write all the necessary parameters. If the necessary parameters are not written or the setting value is out of range, an error occurs.
- The numerical value used in the historical graph is fixed to decimal.
- If the communication load is high for a while, the device value may be lost.

Write button (WSWrtBtn)

This object writes the specified value to the specified device.



Parameter

| Element name | Item | Description | Requirement |
|--------------|-------------------------------|--|--------------------------------|
| devName | Device name | Device type + device number | ○ |
| devBase | Positional notation of device | K: Decimal H: Hexadecimal B: Binary | ○ |
| devFormat | The data format of the device | 0: 16-bit signed 1: 16-bit unsigned 2: 32-bit signed 3: 32-bit unsigned 4: Single-precision real number 6: Bit | ○ |
| wrVal | Write value | The setting range depends on the positional notation and data format of the device. Specify a string for the input value. Example: wrVal: '1' | ○ |
| wrBtn | Write button | Class element name It is used for the selector of the style sheet. | — |
| btnTxt | Text displayed on the button | Optional | — (If omitted, it is empty) |
| btnWidth | Button width | Positive real numbers (unit: px) | ○ |
| btnHeight | Button height | | |
| wrConfirm | Write confirmation message | 0: Not display 1: Display | — (If omitted, it is 0) |
| language | Message language | 0: Japanese 1: English 2: Chinese (Simplified) | — (If omitted, it is 1) |
| xPos | X-coordinate | The horizontal coordinate of the write button at the upper left end (unit: px) | — |
| yPos | Y-coordinate | The vertical coordinate of the write button at the upper left end (unit: px) | — |

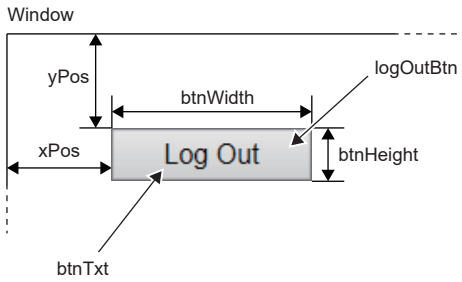
○: Required, —: Optional

Precautions

- Write all the necessary parameters. If the necessary parameters are not written or the setting value is out of range, an error occurs.
- If no parameter is specified for the write button, the default style specified in UserWebStyle.css is applied.

Logout button (WSLogoutBtn)

This object places the button to log out. When the logout button is clicked, the user login page (Log-in_User.html) is displayed.



Parameter

| Element name | Item | Description | Requirement |
|--------------|------------------------------|---|--------------------------------|
| logOutBtn | Logout button | Class element name of button object (It is used for the selector of the style sheet.) | — |
| btnTxt | Text displayed on the button | Optional | — (If omitted, it is empty) |
| btnWidth | Button width | Positive real numbers (unit: px) | ○ |
| btnHeight | Button height | | |
| xPos | X-coordinate | The horizontal coordinate of the logout button at the upper left end (unit: px) | — |
| yPos | Y-coordinate | The vertical coordinate of the logout button at the upper left end (unit: px) | — |

○: Required, —: Optional

Precautions

- Write all the necessary parameters. If the necessary parameters are not written or the setting value is out of range, an error occurs.
- When using the logout button, always store the user login page (Log-in_User.html) in the SD memory card.
- If no parameter is specified for the logout button, the default style specified in UserWebStyle.css is applied.
- Only one logout button can be used in one Web page.

11.2 CGI Object

By using the CGI object, a simple user Web page with small file size can be created.

For an overview of CGI, refer to reference books on the market.

For the user Web page, the following CGI objects dedicated for read/write data from/to devices can be used.

| File name | Item | Description | Reference |
|--------------|------------------|---|---------------------------|
| RdDevRnd.cgi | Device read CGI | Reads the current value of the specified device. | Page 182 Device read CGI |
| WrDev.cgi | Device write CGI | Writes the specified value to the specified device. | Page 187 Device write CGI |

The data acquisition from the Web server or the operation to the Web server is executed by passing it as the CGI request from the client and returning the execution result as a response to the client.

Data specified for CGI object

The data used in the CGI object are listed below.

Device name

The following table lists the devices that can be accessed with the device read CGI and device write CGI.

| Type | Device |
|--------------------|---|
| Bit device | X, Y, M, L, B, F, SB, S, TS, TC, STS, STC, CS, CC, LCS, LCC, SM |
| Word device | T (TN), ST (STN), C (CN), D, W, SW, SD, U□\G□, Z, R |
| Double-word device | LC (LCN), LZ |

Restriction

- When specifying a device name in octal such as X□ or Y□, specify the device name in hexadecimal. (Example: When specifying X20, specify X10 in CGI.)
- When specifying a device name which includes \, such as U□\G□ directly by HTML and JavaScript, repeat \ as U□\\G□. (\ is an escape sequence.)

Device size

Applicable device size differs depending on the device.

For the notation of the device number, specify octal, decimal or hexadecimal depending on the device.

| Device | Notation | Device size | | |
|------------------|---------------------|-------------|------|-------------|
| | | Bit | Word | Double word |
| X | Octal ^{*1} | ○ | × | × |
| Y | Octal ^{*1} | ○ | × | × |
| M | Decimal | ○ | × | × |
| L | Decimal | ○ | × | × |
| B | Hexadecimal | ○ | × | × |
| F | Decimal | ○ | × | × |
| SB | Hexadecimal | ○ | × | × |
| S | Decimal | ○ | × | × |
| T ^{*2} | TS (Contact) | ○ | × | × |
| | TC (Coil) | ○ | × | × |
| | TN (Current value) | × | ○ | ○ |
| ST ^{*2} | STS (Contact) | ○ | × | × |
| | STC (Coil) | ○ | × | × |
| | STN (Current value) | × | ○ | ○ |
| C ^{*2} | CS (Contact) | ○ | × | × |
| | CC (Coil) | ○ | × | × |
| | CN (Current value) | × | ○ | ○ |
| LC ^{*2} | LCS (Contact) | ○ | × | × |
| | LCC (Coil) | ○ | × | × |
| | LCN (Current value) | × | × | ○ |
| D | Decimal | × | ○ | ○ |
| W | Hexadecimal | × | ○ | ○ |
| SW | Hexadecimal | × | ○ | ○ |
| SM | Decimal | ○ | × | × |
| SD | Decimal | × | ○ | ○ |
| G (U□\G□) | Decimal | × | ○ | ○ |
| Z | Decimal | × | ○ | ○ |
| LZ | Decimal | × | × | ○ |
| R | Decimal | × | ○ | ○ |

○: Applicable in both device read CGI and device write CGI, ×: Not applicable

*1 The device number is treated as hexadecimal in CGI.

*2 When T, ST, C, or LC is specified, the device is treated as the device of current value (TN, STN, CN, or LCN).

Device value

The notation of the device value used in the CGI object should be as follows:

- The CGI object treats device values as a string type in hexadecimal notation. When using the decimal number or real number in the Web page, convert decimal or real number by using JavaScript.
- To add 0x to the head is not necessary. Do not perform the 0 interpolation. For example, when reading/writing the value 0x012F, set the string 12F as the device value.
- The case of the hexadecimal alphanumeric character is ignored. For example, the device value can be expressed either as 12F or 12f to read/write the value 0x012F.

Device read CGI



Reads the current value of the specified device.

Access method and access information

| Item | Description |
|--------------------------------------|-------------------|
| Access method | POST |
| Access destination information (URL) | /cgi/RdDevRnd.cgi |

Request specifications

The following table lists the parameters used for the request.

| Parameter name | Data type | Description | Setting range |
|----------------|-----------|---|--|
| NUM | string | Hexadecimal string of the number of read device points (n: 1 to 32) | Set this parameter so that the total number of device points specified to read/write in one Web page is within 32 points. |
| DEV1 | string | Device name of the first point | Up to 10 alphanumeric characters (This parameter is not case-sensitive. The indirect specification, bit specification, digit specification, or index modification cannot be performed.)  Page 180 Device name |
| TYP1 | string | Device size of the first point | B: Bit W: Word D: Double word  Page 181 Device size |
| : | | | |
| DEV(n) | string | Device name of the nth point (n: 1 to 32) | Up to 10 alphanumeric characters (This parameter is not case-sensitive. The indirect specification, bit specification, digit specification, or index modification cannot be performed.) |
| TYP(n) | string | Device size of the nth point | B: Bit W: Word D: Double word |

The format of the request data is in query string. Specify DEV(n) and TYP(n) with the serial number. Otherwise, an error occurs.

Ex.

When reading 10 points of device from D0, M100, ... , SD0

NUM=A&DEV1=D0&TYP1=D&DEV2=M100& ... &DEV10=SD0&TYP10=W

Response specifications

The following table lists the parameters used for the response.

| Parameter name | Data type | Description |
|----------------|-----------|--|
| RET | string | Execution result (hexadecimal string) 0000: Normal 0001: Not login 0005: Incorrect request source (Referer) 4005: Exceeded the number of points 4030: Incorrect device type 4031: Out of device range 4041: Error due to the specified buffer memory number + specified number of transfer points out of buffer memory range 4043: Error due to the specified module not existing 4080: CGI parameter error |
| DATA | string | Reading value (array) Hexadecimal string |

The format of the response data is JSON.

Ex.

Response data of the device read CGI

```
{
  "RET" : "0000",
  "DATA" : [
    "100",
    :
    "FABC"
  ]
}
```

The above response is transferred in the following format on a message.

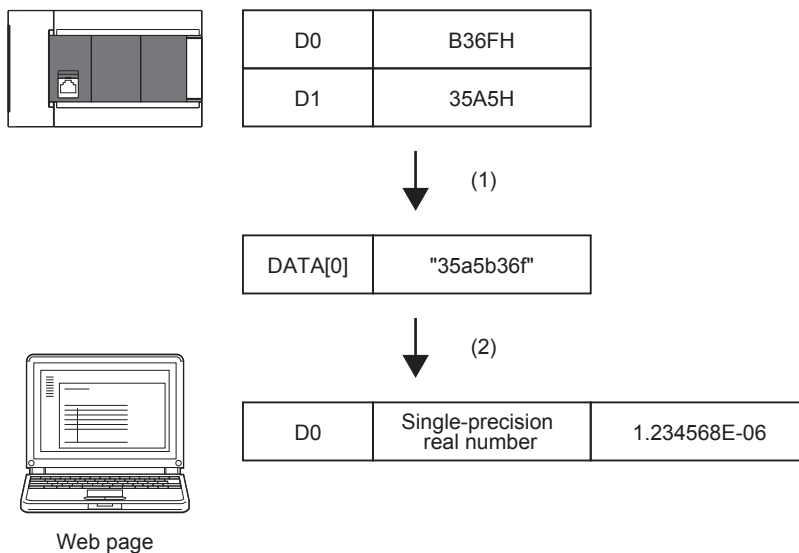
```
{"RET":"0000","DATA":["100", ... ,"FABC"]}
```

When the execution result is abnormal, only RET is transferred.

```
{"RET":"4031"}
```

Point

To display the device value in the real number format on the Web page, specify D: Double word for the device size and execute a request (1). Convert the read data to the real number format by using JavaScript (2).



Application example

An example for reading values by using the device read CGI is shown below.

■Display of HTML

| Device Name | Data Type | Value |
|-------------|----------------|-------|
| D10 | 32-bit Integer | |
| SD0 | 16-bit Integer | |
| M0 | Bit | |

Read

(1) (2) (3) (4)

- (1) Set DEV1
- (2) Set TYP1
- (3) Set VAL1
- (4) Click to call the ReadDeviceRandomTbl function of JavaScript (Set the table ID for the argument)

■HTML example

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <!-- charset setting*Set UTF-8 which is set in the Web server setting -->
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge"/>
    <!-- Set the title name -->
    <title>Device read CGI sample</title>
    <!-- Write the JavaScript of the user here -->
    <script>
      // The function for a CGI request
      function ReadDeviceRandomTbl(devtblid) {
        var devtblitem = document.getElementById(devtblid);
        var i, devitem, typitem;
        var tblrows = devtblitem.rows.length;
        var param;

        // Number of device points setting
        param = "NUM=" + (tblrows - 1) + '&';
        for( i=1; i < tblrows; i++ ) {
          // Parameter setting of the device name
          devitem = document.getElementById(devtblitem.rows[i].cells[0].childNodes[0].id);
          param += devitem.name + "=" + devitem.value + '&';

          // Parameter setting of the device size
          typitem = document.getElementById(devtblitem.rows[i].cells[1].childNodes[0].id);
          if( "Bit" == typitem.value ) {
            param += typitem.name + "=" + 'B';
          }
          else if( "16-bit Integer" == typitem.value ) {
            param += typitem.name + "=" + 'W';
          }
          else if( "32-bit Integer" == typitem.value ) {
            param += typitem.name + "=" + 'D';
          }
          else {
            param += typitem.name + "=" + 'Q';
          }
          if( i < (tblrows - 1) ) param += '&';
        }

        // Request to the CGI
        xhr = new XMLHttpRequest();
        xhr.open("POST", "/cgi/RdDevRnd.cgi", true);
        xhr.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');
        var FUNC = function() { ReadDeviceRandomTbl_Response(xhr, devtblid); }; // Response analysis function setting
        xhr.onreadystatechange = FUNC;
        xhr.send(param);
      }
    </script>
  </head>
  <body>
    <table border="1">
      <thead>
        <tr>
          <th>Device Name</th>
          <th>Data Type</th>
          <th>Value</th>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td>D10</td>
          <td>32-bit Integer</td>
          <td></td>
        </tr>
        <tr>
          <td>SD0</td>
          <td>16-bit Integer</td>
          <td></td>
        </tr>
        <tr>
          <td>M0</td>
          <td>Bit</td>
          <td></td>
        </tr>
      </tbody>
    </table>
    <input type="button" value="Read"/>
  </body>
</html>
```

```

// The function for analyzing a response
function ReadDeviceRandomTbl_Response(xhr, devtblid) {
  // XMLHttpRequest Client status check
  // 0:UNSENT 1:OPENED 2:HEADERS_RECEIVED 3:LOADING 4:DONE
  if( 4 != xhr.readyState ) {
    // End the processing if the status 4 is other than DONE (operation complete).
    return;
  }

  // HTTP Response code check
  if ( 200 != xhr.status ) {
    // Display the error dialog box if the response code is other than "200 OK".
    alert("HTTP STATUS ERROR=" + xhr.status );
    return;
  }

  var i, dataitem;
  var devtblitem = document.getElementById(devtblid);
  var tblrows = devtblitem.rows.length; // Obtain the number of the table lines (including the header).
  var res = JSON.parse( xhr.response ); // Analysis processing of JSON string

  // Judgment from the CGI
  if( res.RET != "0000" ) {
    // Display the error dialog box if the result is abnormal.
    alert("ERROR=" + res.RET);
  }
  else {
    // Reflect the obtained value to the table if the result is normal.
    for ( i = 1, m = 0; i < tblrows; i++, m++) {
      dataitem = document.getElementById(devtblitem.rows[i].cells[2].childNodes[0].id);
      // Set the read result to the value of the table (convert the hexadecimal string to numerical value).
      dataitem.value = parseInt(res.DATA[m], 16);
    }
    alert("read complete");
  }
}
</script>
</head>

```

```

<body>
<form>
  <table id="devtbl" class="devtbl" border="1">
    <tr>
      <th>Device Name</th>
      <th>Data Type</th>
      <th>Value</th>
    </tr>
    <tbody>
      <tr>
        <td><input type="text" id="DEV1" name="DEV1" class="input" value="D10"/></td>
        <td><input type="text" id="TYP1" name="TYP1" class="input" value="16-bit Integer"/></td>
        <td><input type="text" id="DATA1" name="DATA1" class="read-input" /></td>
      </tr>
      <tr>
        <td><input type="text" id="DEV2" name="DEV2" class="input" value="D11"/></td>
        <td><input type="text" id="TYP2" name="TYP2" class="input" value="32-bit Integer"/></td>
        <td><input type="text" id="DATA2" name="DATA2" class="read-input" /></td>
      </tr>
      <tr>
        <td><input type="text" id="DEV3" name="DEV3" class="input" value="M0"/></td>
        <td><input type="text" id="TYP3" name="TYP3" class="input" value="Bit"/></td>
        <td><input type="text" id="DATA3" name="DATA3" class="read-input" /></td>
      </tr>
    </tbody>
  </table>
  <input type="button" value="Read" class="input" onclick="ReadDeviceRandomTbl('devtbl')"/>
</form>
</body>
</html>

```

| Device Name | Data Type | Value |
|-------------|----------------|-------|
| D10 | 16-bit Integer | |
| D11 | 32-bit Integer | |
| M0 | Bit | |

Read

(5) Display the read result.

Point

In the above example, the request parameter is as follows:
 NUM=3&DEV1=D10&TYP1=D&DEV2=SD0&TPY2=W&DEV3=M0&TYP3=B

Device write CGI

Writes the specified value to the specified device.

Access method and access information

| Item | Description |
|--------------------------------------|----------------|
| Access method | POST |
| Access destination information (URL) | /cgi/WrDev.cgi |

Request specifications

The following table lists the parameters used for the request.

| Parameter name | Data type | Description | Setting range |
|----------------|-----------|-----------------------------------|--|
| NUM | string | Number of write device points (1) | Set 1. (Set this parameter so that the total number of device points specified to read/write per Web page is within 32 points.) |
| DEV1 | string | Device name | Up to 10 alphanumeric characters (This parameter is not case-sensitive. The indirect specification, bit specification, digit specification, or index modification cannot be performed.) ☞ Page 180 Device name |
| TYP1 | string | Device size | B: Bit W: Word D: Double word ☞ Page 181 Device size |
| DATA1 | string | Write value | Hexadecimal string |

The format of the request data is the query string.

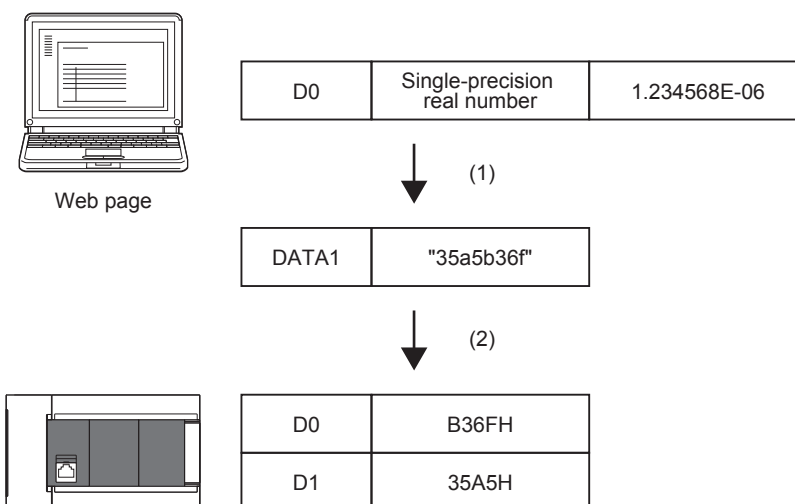
Ex.

Writing FFFFH to D0

NUM=1&DEV1=D0&TYP1=W&DATA1=FFFF

Point

When writing the device value which is input to the Web page in the real number format, the input data needs to be converted to the hexadecimal format by using JavaScript (1). For the device size, D: Double word needs to be specified and performing a request (2) are required as well.



Response specifications

The following table lists the parameters used for the response.

| Parameter name | Data type | Description |
|----------------|-----------|--|
| RET | string | Execution result (hexadecimal string) 0000: Normal 0001: Not login 0002: No permission (A user without device write permission executed the CGI.) 0005: Incorrect request source (Referer) 4005: Exceeded the number of points 4030: Incorrect device type 4031: Out of device range 4041: Specified buffer memory number + specified number in the transfer points is out of buffer memory range error 4043: Error due to the specified module not existing 4080: CGI parameter error |
| DATA | string | The read value of the target device (array) Hexadecimal string |

The format of the response data is JSON.

Ex.

Response data of the device write CGI

```
{
  "RET" : "0",
  "DATA" : [
    "100"
  ]
}
```

The above response is transferred in the following format on a message.

```
{"RET":"0","DATA":["100"]}
```

When the execution result is abnormal, data is not written and only the RET is transferred.

```
{"RET":"4031"}
```

Application example

An example for writing a value by using the device write CGI is shown below.

■ Display of HTML

| Device Name | Data Type | Value |
|-------------|----------------|-------|
| D10 | 32-bit Integer | 3 |

Write

(1) Set DEV1.

(2) Set TYP1.

(3) Set VAL1.

(4) Click it to call the WriteDeviceBlockTbl function of JavaScript. (Set the table ID, the line number where the write starts, and the number of write points 1 for the arguments.)

■HTML example

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <!-- charset setting*Set UTF-8 which is set in the Web server setting -->
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge"/>
    <!-- Set the title name -->
    <title>Device write CGI sample</title>
    <!-- Write the JavaScript of the user here -->
    <script>
      function WriteDeviceBlockTbl(devtblid, row) {
        var dataitem;
        var xhr;
        var devtblitem = document.getElementById(devtblid);
        // Number of device points setting (fixed to one point)
        var param = 'NUM=1&';
        // Parameter setting of the device name
        var devitem = document.getElementById(devtblitem.rows[row].cells[0].childNodes[0].id);
        param += 'DEV1=' + devitem.value + '&';

        // Parameter setting of the device size
        var typitem = document.getElementById(devtblitem.rows[row].cells[1].childNodes[0].id);
        if( 'Bit' == typitem.value ) {
          param += 'TYP1=' + 'B';
        }
        else if( '16-bit Integer' == typitem.value ) {
          param += 'TYP1=' + 'W';
        }
        else if( '32-bit Integer' == typitem.value ) {
          param += 'TYP1=' + 'D';
        }
        else {
          param += 'TYP1=' + 'Q';
        }
        param += '&';

        // Parameter setting of the data
        var dataitem = document.getElementById(devtblitem.rows[row].cells[2].childNodes[0].id);
        param += 'DATA1=' + parseInt(dataitem.value).toString(16)

        // Request to the CGI
        xhr = new XMLHttpRequest();
        xhr.open('POST', "/cgi/WrDev.cgi", true);
        xhr.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');
        var FUNC = function() { WriteDeviceBlockTbl_Response(xhr, typitem, dataitem); }; // Response analysis function setting
        xhr.onreadystatechange = FUNC;
        xhr.send(param);
      }
    </script>
  </head>
  <body>
  </body>
</html>
```

```

// The function for analyzing a response
function WriteDeviceBlockTbl_Response(xhr, typitem, dataitem) {
  // XMLHttpRequest Client status check
  // 0:UNSENT 1:OPENED 2:HEADERS_RECEIVED 3:LOADING 4:DONE
  if( 4 != xhr.readyState ) {
    // End the processing if the status 4 is other than DONE (operation complete).
    return;
  }
  // HTTP Response code check
  if ( 200 != xhr.status ) {
    // Display the error dialog box if the response code is other than "200 OK".
    alert("HTTP STATUS ERROR=" + xhr.status );
    return;
  }

  var value;
  var res = JSON.parse( xhr.response ); // Analysis processing of JSON string
  // Judgment from the CGI
  if( res.RET != "0000" ) {
    // Display the error dialog box if the result is abnormal.
    alert("ERROR=" + res.RET);
  }
  else {
    // Reflect the write result value if the result is normal.
    dataitem.value = parseInt(res.DATA[0],16);
    alert("write complete");
  }
}
</script>
</head>

<body>
<form>
<table id="devtbl" class="devtbl" border="1">
<tr>
<th>Device Name</th>
<th>Data Type</th>
<th>Value</th>
</tr>
<tbody>
<tr>
<td><input type="text" id="DEV1" name="DEV1" class="input" value="D10"/></td>
<td><input type="text" id="TYP1" name="TYP1" class="input" value="16-bit Integer"/></td>
<td><input type="text" id="DATA1" name="DATA1" class="input" value="3"/></td>
<td><input type="button" value=" Write" class="input" onclick="WriteDeviceBlockTbl('devtbl',1)"/></td>
</tr>
<tr>
<td><input type="text" id="DEV2" name="DEV2" class="input" value="D11"/></td>
<td><input type="text" id="TYP2" name="TYP2" class="input" value="32-bit Integer"/></td>
<td><input type="text" id="DATA2" name="DATA2" class="input" value="10"/></td>
<td><input type="button" value=" Write" class="input" onclick="WriteDeviceBlockTbl('devtbl',2)"/></td>
</tr>
<tr>
<td><input type="text" id="DEV3" name="DEV3" class="input" value="M0"/></td>
<td><input type="text" id="TYP3" name="TYP3" class="input" value="Bit"/></td>
<td><input type="text" id="DATA3" name="DATA3" class="input" value="1"/></td>
<td><input type="button" value=" Write" class="input" onclick="WriteDeviceBlockTbl('devtbl',3)"/></td>
</tr>
</tbody>
</table>
</form>
</body>
</html>

```

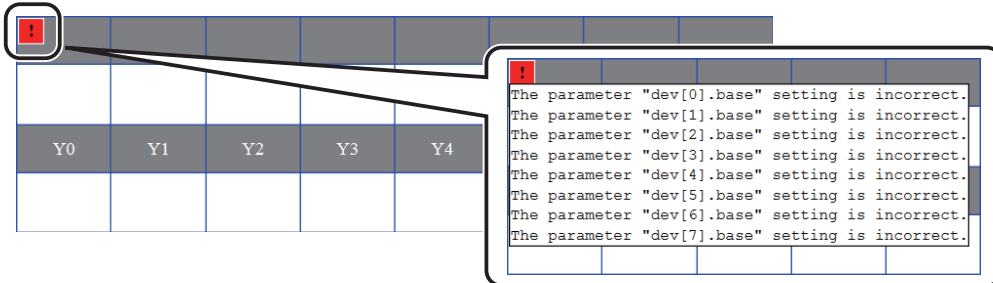
Point

In the above example, the request parameter is as follows:
 NUM=1&DEV1=D10&TYP1=D&DATA1=3

11.3 Error message

Display example of error messages

If an object of the user Web page is used incorrectly, a button to notify the user of an error is displayed at the upper left of the object when the user Web page is displayed. The error message related to the common processing which is not limited to the object is displayed at the upper left of the window.



Click the button to display the detailed information of the error.

List of error messages of the user Web page

The following table lists error messages, causes, and action of the user Web page.

| Description | Cause | Action |
|--|--|--|
| The specified element "XXX" does not exist. | The specified element does not exist. | Check the contents of the HTML file. |
| A required parameter "XXX" has not been set. | A required parameter has not been set. | |
| The setting of the parameter "XXX" is incorrect. | Parameters are out of the setting range. | |
| | Set values of parameters that should be numeric values are specified with other than numeric values. | |
| The parameter "XXX" and the settings of "XXX" are incorrect. | The upper/lower limit values are specified as upper limit value < lower limit value. | |
| The data format of the device "XXX" is incorrect. | The data positional notation of the device and the data format of the device are incorrect. | |
| The number of applicable objects has exceeded the limit. | There are certain objects which exceeds the applicable number in one page. | Check that multiple logout objects are not used in one page. |
| The receive data contains errors. | The receive data contains errors. | Check that the CPU module and the client terminal are properly connected and update the Web page. |
| You were logged off. Log out and try again. | The user has been logged off. (Communications are disconnected.) | Check that the CPU module and the terminal are properly connected and login again. |
| Specified devices are invalid. The following are the possible causes. <ul style="list-style-type: none"> • A device such as pointer and constant • A device which exceeds the device range of device setting of CPU parameters • Unsupported device | Device names are incorrect. | Check parameters of each object to see if any device name is specified incorrectly. |
| Specified device No. is invalid or out of the range. Check the device No. in the device setting of parameters. | Device numbers and buffer memory addresses are out of range or not numeric values. | Check parameters of each object to see if any device number and buffer memory address are specified incorrectly. |
| Specified modules do not exist. Check the modules. | The module numbers are incorrect. An error has occurred in communications with intelligent function modules. | Check parameters of each object to see if any module number is specified incorrectly. Check that there are no power-offs or errors in extension units. |
| The request data is incorrect. | The request parameter of CGI object is incorrect. The size of request parameter of CGI object is too large. | Check the parameter of CGI object. |

For details on the other errors, refer to the following.

- 📖 MELSEC iQ-R/MELSEC iQ-F Web Server Function Guide Book
- 📖 MELSEC iQ-F FX5 User's Manual (Application)

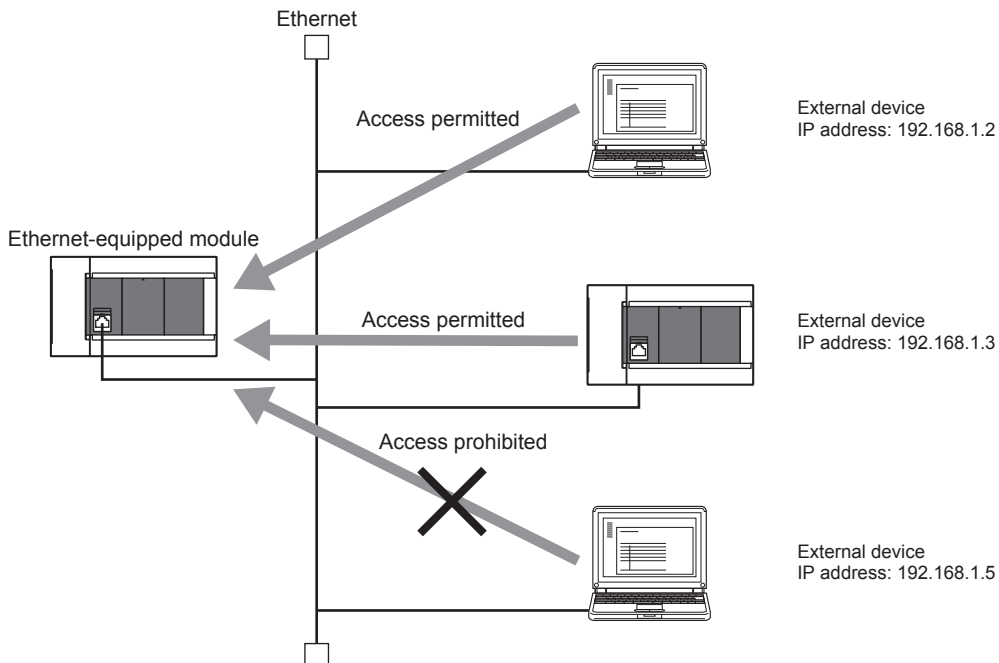
12 SECURITY FUNCTION

12.1 IP Filter Function

Identifies the IP address of the access source, and prevents access from an illegal IP address.

The IP address of the external device to be allowed or denied is set in the parameters, and access from external devices is restricted.

Use of this function is recommended when using in an environment connected to a LAN line.



Point

The IP filter function is one method of preventing illegal access (such as program or data destruction) from an external device. It does not completely prevent illegal access. Incorporate measures other than this function if the programmable controller system's safety must be maintained against illegal access from an external device. Mitsubishi shall not be held liable for any system problems that may occur from illegal access. Examples of measures for illegal access are given below.

- Install a firewall
- Install a personal computer as a relay station, and control the relay of send/receive data with an application program
- Install an external device for which the access rights can be controlled as a relay station (Contact the network provider or equipment dealer for details on the external devices for which access rights can be controlled.)

Setting method

1. Set the IP address to be allowed or denied in "IP Filter Settings" of "Security" under "Application Settings". (Page 195 IP filter settings)
2. Write the module parameters to the CPU module.
3. The IP filter function is enabled when the CPU module power is turned off and on or reset.

Point

Even if the connection is established as set with the Ethernet-equipped module's "External Device Configuration" under "Basic Settings" or the program, access from the external device is either allowed or denied following "IP Filter Settings" of "Security" under "Application Settings".

Therefore, if the IP address set in the CPU module's "External Device Configuration" under "Basic Settings" is set to be denied with "IP Filter Settings" of "Security" under "Application Settings", the IP filter function is enabled and communication with the external device is denied.

Precautions

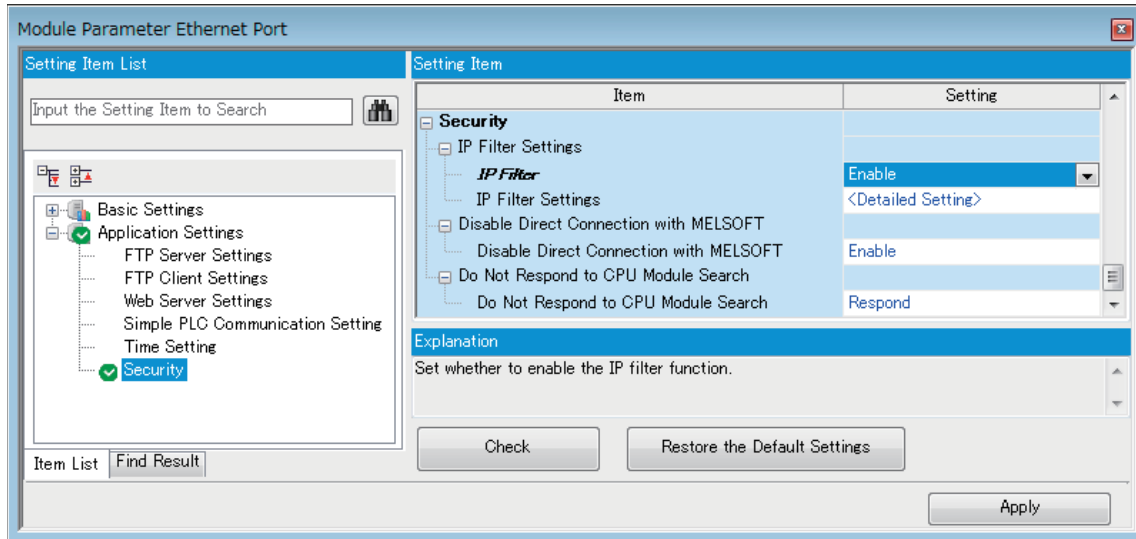
- If there is a proxy server in the LAN line, deny the IP address for the proxy server. If the IP address is allowed, it will not be possible to prevent access from personal computers that access the proxy server.
- When the CPU module is connected to the personal computer by Ethernet, if IP address allowed in this function is forgotten, access to the CPU module cannot be executed.

Setting of the security measures for access to the Ethernet-equipped module

The following shows the setting of the security measures for access to the Ethernet-equipped module.

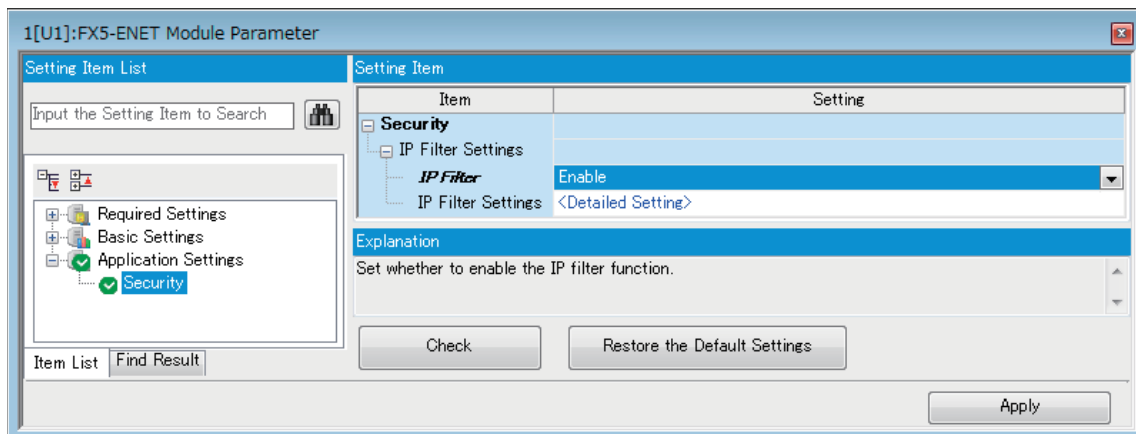
[CPU module]

Navigation window⇒[Parameter]⇒[FX5UJCPU] or [FX5UCPU]⇒[Module Parameter]⇒[Ethernet Port]⇒[Application Settings]⇒ [Security]



[Ethernet module]

Navigation window⇒[Parameter]⇒[Module information]⇒[FX5-ENET] or [FX5-ENET/IP]⇒[Application Settings]⇒ [Security]



| Item | Description | Setting range |
|--|---|---|
| IP Filter Settings | IP Filter | Set whether to enable the IP filter function. • Disable • Enable (Default: Enable) |
| | IP Filter Settings | Set the IP address to be allowed or denied. (☞ Page 195 IP filter settings) |
| Disable Direct Connection with MELSOFT*1 | Set whether to enable or disable direct connection to the engineering tool. | • Disable • Enable (Default: Enable) |
| Do Not Respond to CPU Module Search*1 | Select whether to respond to the CPU module search. | • Do Not Respond • Respond (Default: Respond) |

*1 Only CPU module is supported.

■ IP filter settings

Up to 4 IP addresses can be set as an IP address to be allowed or denied by the IP filter function.

Range specification and specification of the IP addresses to be excluded from the set range are also possible.

| Item | Description | Setting range |
|--------------------------------|---|---|
| Access from IP address below | Select whether to allow or deny the access from the specified IP addresses. | <ul style="list-style-type: none">• Allow• Deny (Default: Allow) |
| Range Setting | Select this item when specifying the IP addresses by range. | (Default: Clear) |
| IP Address | Set the IP address to be allowed or denied. When selecting "Range Setting", enter the start IP address (left field) and end IP address (right field) of the range. | 0.0.0.1 to 223.255.255.254 (Default: Blank) |
| IP Address Excluded from Range | When selecting "Range Setting", set the IP address to be excluded from the set range. Up to 16 IP addresses can be set. | 0.0.0.1 to 223.255.255.254 (Default: Blank) |

12.2 Remote Password

Remote password is checked when a connection is requested for the following.

- Communication using an engineering tool
- Communication using SLMP
- Communication using FTP port

Point

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

However, this function cannot completely prevent unauthorized access.

Other measures should be taken at users' discretion if security of the programmable controller system against unauthorized access from external devices needs to be maintained. Mitsubishi Electric cannot be held responsible 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 an external device that can control access rights as a relay station (For external devices that can control access rights, please consult your network service provider or networking equipment vendors.)

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.

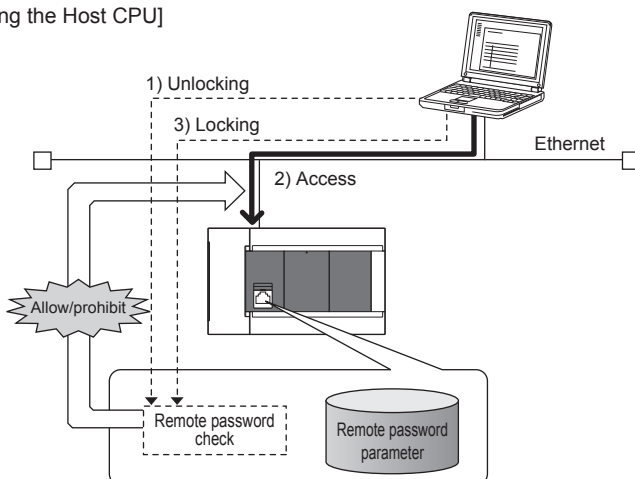
2. Access processing

Access the CPU module after successful 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 access from any other personal computers.

[Accessing the Host CPU]

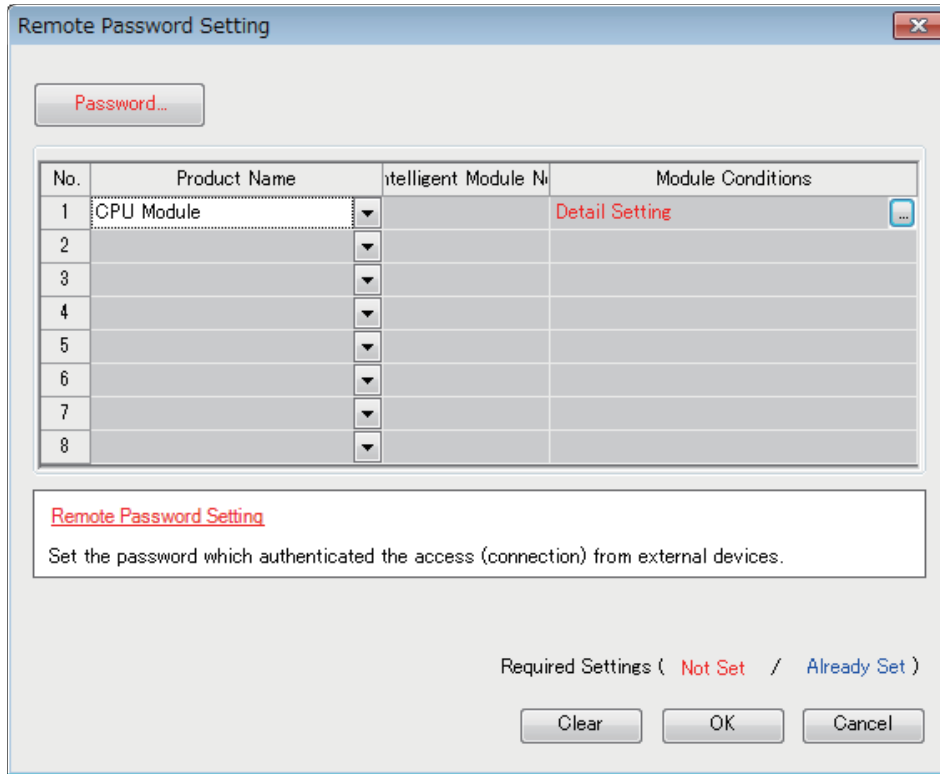


Remote password setting

Setting a remote password

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

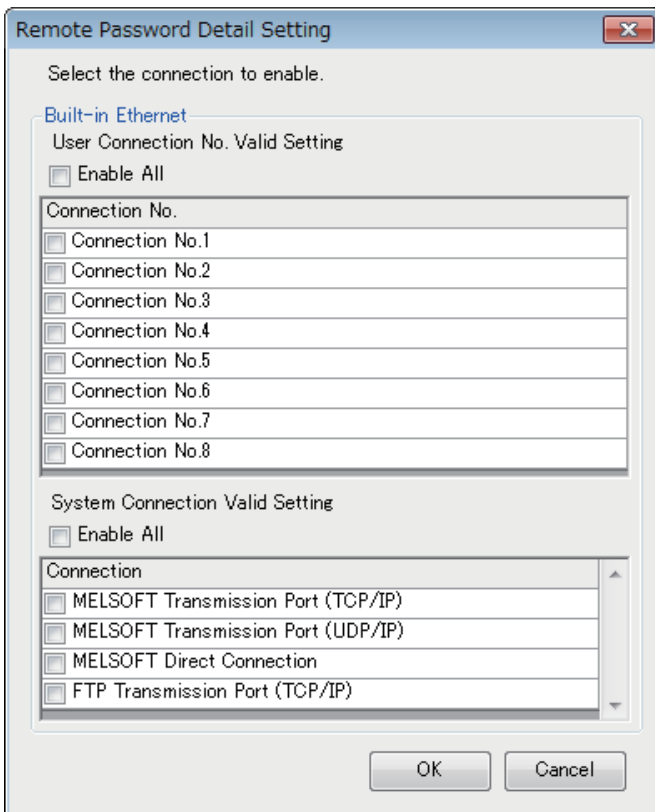
Navigation Window⇒[Parameter]⇒[Remote Password]⇒[Remote Password Setting] Screen



| Item | Description | Setting range |
|------------------------|---|--------------------------------|
| Password | Open "Register Password"/ "Change Password" screen. Enter a remote password to be set for the CPU module.*1 | 6 to 32 Single byte characters |
| Product Name | CPU Module Select "CPU Module" to enable the remote password for the built-in Ethernet port of the CPU module. | Only "CPU Module" |
| Intelligent Module No. | This setting is not required. | — |
| Module Conditions | Detail Setting Click this to display the "Remote Password Detail Setting" screen. | — |

*1 Half-width alphanumeric and special characters can be used for remote password entry. (Case-sensitive)

• Remote password detail setting screen



| Item | | Description | Setting range |
|-------------------|---|---|--|
| Built-in Ethernet | User Connection No. Valid Setting ^{*1} | Connection 1 to 8 | Select whether the remote password is to be enabled for the built-in Ethernet port. (Setting of an unused connection or MELSOFT connection is ignored.) |
| | System Connection Valid Setting ^{*2} | MELSOFT Transmission Port (TCP/IP) ^{*3} MELSOFT Transmission Port (UDP/IP) MELSOFT Direct Connection ^{*4} FTP Transmission Port (TCP/IP) ^{*5} | Select whether the remote password is to be enabled for the built-in Ethernet port. |

*1 User connection is a connection for users for communication such as SLMP communication.

*2 System connection is used by the system for communication such as MELSOFT communications (TCP/IP).


*3 Check this checkbox to enable the remote password for the ports for which the communication system is set to "MELSOFT Connection" in engineering tool.

*4 Check this checkbox to enable the remote password for CPU module direct connection to engineering tool using the built-in Ethernet port. (☞ Page 23 Direct Connection with Engineering Tool)

*5 Check this checkbox to enable the remote password for access by the file transfer function (FTP server). (☞ Page 137 FILE TRANSFER FUNCTION (FTP SERVER))

Writing to the CPU module

Write the set remote password to the CPU module from the "Write to PLC" screen.

 [Online]⇒[Write to PLC]

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

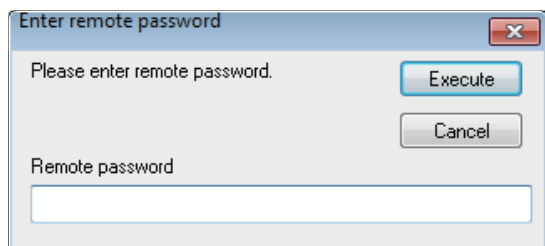
Unlocking or locking the remote password

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

■When using MELSOFT connection

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

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



■In case of SLMP

Use commands dedicated to SLMP. (Lock /unlock of  Page 48 Command list)

■In case of FTP Transmission Port

Use the dedicated FTP command. ( Page 141 FTP command list, password-lock/password-unlock)

Precautions

When a remote password is set for UDP connections

- Determine a target device before data communication. (At the time of SLMP setting, set "Host station port number", "Communication target IP address", "Communication target port number", and limit the communication target.)
- 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 set the parameter to disable direct connection.

When a TCP/IP connection is closed before lock processing

The CPU module automatically performs lock processing.

Further, when protocol is set to TCP, connection is verified by KeepAlive. (Response to KeepAlive ACK message)

An alive check message is sent 5 seconds after reception of the last message from the device with which communication is being done 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 5 seconds. When no response is received for 45 seconds, the connected device is regarded as non-existent and the connection is disconnected.

Therefore, the lock process is automatically executed when the connection is cut.

Valid range of remote password

The remote password is valid only for accessing the module (communication port) for which the parameter is set.


In case of a system configuration that uses multiple modules, set a remote password for each module (communication ports).

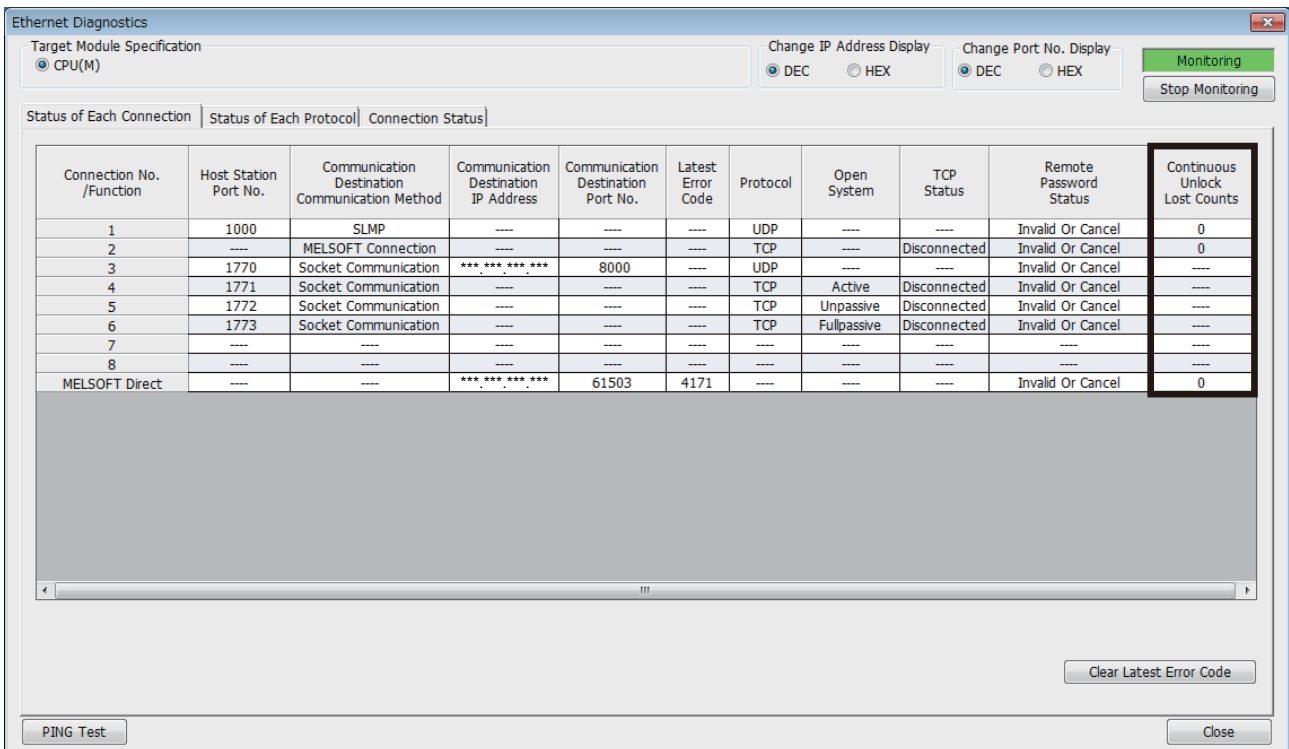
Detection of unauthorized access and actions

When the password mismatch count reaches a fixed count (upper limit) in the unlock process of remote password, access is locked out. If this occurs, unauthorized access from outside the system can be considered as a cause of the error.

Take the following actions as needed.

1. Monitor the unlock failure count (SD10320 to SD10327) and identify the connection in which the mismatch count has reached a fixed count (upper limit) in unlock processing. The continuous unlock lost counts also can be identified on the "Ethernet Diagnostics" screen of GX Works3.

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



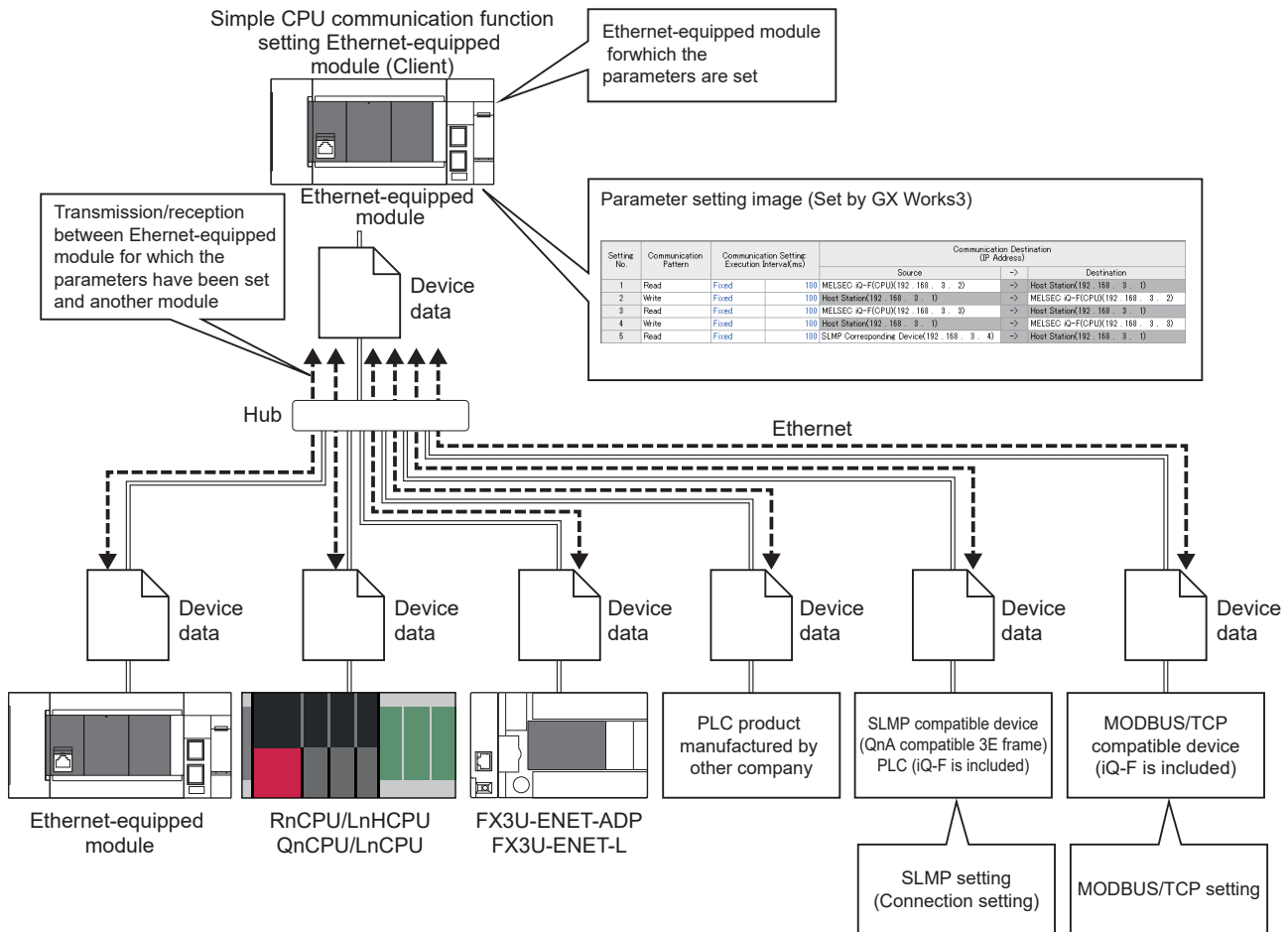
The screenshot shows the 'Ethernet Diagnostics' window with the 'Status of Each Connection' tab selected. The table below represents the data shown in the window:

| Connection No. /Function | Host Station Port No. | Communication Destination Communication Method | Communication Destination IP Address | Communication Destination Port No. | Latest Error Code | Protocol | Open System | TCP Status | Remote Password Status | Continuous Unlock Lost Counts |
|--------------------------|-----------------------|--|--------------------------------------|------------------------------------|-------------------|----------|-------------|--------------|------------------------|-------------------------------|
| 1 | 1000 | SLMP | ---- | ---- | ---- | UDP | ---- | ---- | Invalid Or Cancel | 0 |
| 2 | ---- | MELSOFT Connection | ---- | ---- | ---- | TCP | ---- | Disconnected | Invalid Or Cancel | 0 |
| 3 | 1770 | Socket Communication | ***.***.***.*** | 8000 | ---- | UDP | ---- | ---- | Invalid Or Cancel | ---- |
| 4 | 1771 | Socket Communication | ---- | ---- | ---- | TCP | Active | Disconnected | Invalid Or Cancel | ---- |
| 5 | 1772 | Socket Communication | ---- | ---- | ---- | TCP | Unpassive | Disconnected | Invalid Or Cancel | ---- |
| 6 | 1773 | Socket Communication | ---- | ---- | ---- | TCP | Fullpassive | Disconnected | Invalid Or Cancel | ---- |
| 7 | ---- | ---- | ---- | ---- | ---- | ---- | ---- | ---- | ---- | ---- |
| 8 | ---- | ---- | ---- | ---- | ---- | ---- | ---- | ---- | ---- | ---- |
| MELSOFT Direct | ---- | ---- | ***.***.***.*** | 61503 | 4171 | ---- | ---- | ---- | Invalid Or Cancel | 0 |

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

13 SIMPLE CPU COMMUNICATION FUNCTION

This function allows data communications between specified devices at the specified timing just by setting simple parameters from the engineering tool for the Ethernet-equipped module. Specify one device (transmission source) to one device (transmission destination). The communications are performed between the specified devices.



13

The external devices which can communicate vary depending on the Ethernet-equipped module (client) used. For details, refer to Page 205 Module type list.

Point

- The following maximum number of devices can be connected communication destination.
 - FX5UJ CPU module: 8 devices
 - FX5U/FX5UC CPU module: 16 devices
 - Ethernet module: 32 devices
- Access via routers is also available. For the access, set the subnet mask and default gateway.

Specifications of each Ethernet-equipped module

The CPU module and Ethernet module differ in the following specifications.

| Item | CPU module | Ethernet module |
|----------------------|------------------------------|--|
| Connection | Not occupies any connection. | Occupies 1 connection for 1 communication target device (1 group). |
| Own station port No. | Duplication is not allowed. | Duplication is allowed. |

■Group

The Ethernet module communicates with groups. The groups are assigned to the setting numbers according to the simple CPU communication settings. The module can communicate with up to 32 groups.

When the following simple CPU communication setting items of devices are identical, the devices are regarded as one group.

- Device type
- IP address
- TCP/UDP
- Port No.
- Own station port No.

13.1 Setting Procedure

The following section describes the settings for using this function.

1. Set "To Use or Not to Use Simple PLC Communication Setting" to "Enable".

[CPU module]

Navigation window⇒[Parameter]⇒[FX5UJCPU] or [FX5UCPU]⇒[Module Parameter]⇒[Ethernet Port]⇒[Application Settings]⇒[Simple PLC Communication Setting]

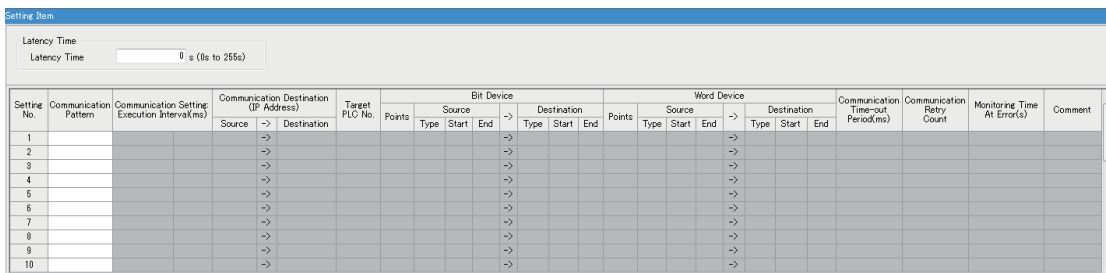
[Ethernet module]

Navigation window⇒[Parameter]⇒[Module Information]⇒[FX5-ENET] or [FX5-ENET/IP]⇒[Application Settings]⇒[Simple CPU Communication Setting]

2. Set the "CPU Response Monitoring Timer". (Only Ethernet module)

| Item | Description | Setting range |
|-------------------------------|---|---|
| CPU Response Monitoring Timer | Set the time for monitoring the response from the CPU module. If the CPU module does not respond within the set time, the response wait state will be canceled. | <ul style="list-style-type: none"> • 1 to 163830 (in sec) • 100 to 1638300 (in ms) (Default: 5) |
| Unit | Select the unit of the CPU response monitor timer. When "ms" is selected, set the time in units of 100 ms. | <ul style="list-style-type: none"> • s • ms (Default: s) |

3. Double-click "<Detailed Setting>" of the "Simple PLC Communication Setting" to display the following window.



4. After setting the items, write the module parameters to the Ethernet-equipped module on the own station, and reset or power off and on the CPU module to start the simple CPU communication.

Number of settings

The setting range for the number of settings is as the following.

| Setting types | Number of setting |
|---------------|---|
| Setting No. | <ul style="list-style-type: none"> • FX5UJ CPU module: 1 to 16 • FX5U/FX5UC CPU module: 1 to 32 • Ethernet module: 1 to 32 |

Communication Pattern

Select the communication pattern from the following items.

| Item | Description | Setting range |
|-----------------------|---|--|
| Communication Pattern | Specify whether to read or write the data. <ul style="list-style-type: none"> • Read: Read the data from the specified destination device (transmission source) to the specified device of the host station (transmission destination). • Write: Write the data of the specified device from the host station (transmission source) to the specified destination device (transmission destination). | <ul style="list-style-type: none"> • Read • Write (Default: Blank) |

Communication Setting and Execution Interval

Select the communication timing from the following items.

| Item | | Description | Setting range |
|--|-----------------------|---|---|
| Communication Setting: Execution Interval (ms) | Communication Setting | Setting for periodic data transmission/receipt. <ul style="list-style-type: none"> Fixed: Device data will be transmitted and received at the specified execution interval. Request: Data are communicated between the devices upon request only. | <ul style="list-style-type: none"> Fixed Request*1 (Default: Fixed) |
| | Execution Interval | Set the execution interval when "Fixed" has been selected in the communication setting. | 10 ms to 65535 ms (in increments of 1 ms) (Default: 100 ms) |

*1 Supported only for Ethernet module.

Point

- Actual time of execution interval may be longer than the value of the setting because the time is affected by the specified communication destinations or Ethernet line congestion. (☞ Page 243 When the simple CPU communication function is used)
- If latency time is set, data communication is started after the latency time has passed. (☞ Page 216 Latency Time)
- The execution interval becomes larger than the setting value when the execution interval is set smaller than the scan time in the execution interval setting because data transmission/receipt is executed by the END processing.

Communication Destination


Set the devices for both transmission source and transmission destination. The following number of devices can be registered as communication destination.

- FX5UJ CPU module: 8 devices
- FX5U/FX5UC CPU module: 16 devices
- Ethernet module: 32 devices

| Item | | Description | Setting range |
|--|-------------|----------------------|---|
| Communication Destination (IP Address) | Source | Module type*1 | Specify the device type of the communication destination. <ul style="list-style-type: none"> MELSEC iQ-R (Built-in Ethernet) MELSEC-Q (Built-in Ethernet) MELSEC-L (Built-in Ethernet) MELSEC iQ-F (Built-in Ethernet) MELSEC iQ-F (Ethernet module) MELSEC iQ-L (Built-in Ethernet) MELSEC-FX3 (Ethernet Block/Adapter) SLMP Compatible Device (QnA Compatible 3E Frame) OMRON (CJ/CP series) KEYENCE (KV series) Panasonic (FP7 series) Panasonic (FP0H series) MODBUS/TCP-compatible device SIEMENS S7 series (Default: Blank) |
| | | IP address | Input the IP address of the communication destination. ☞ Page 206 IP address, TCP/UDP, port number, and own station port number |
| | | TCP/UDP | Set the protocol of the communication destination. ☞ Page 207 Target PLC No. |
| | | Port number | Set the port number of the communication destination. |
| | | Station port number | Set the port number of the own station. |
| | | Option (hexadecimal) | Set the module ID of the communication destination. |
| | | Target PLC No. | Set the target PLC No. of the communication destination. |
| | Destination | Module type | Same as each item of the transmission source. |
| | | IP address | |
| | | TCP/UDP | |
| | | Port number | |
| | | Station port number | |
| | | Option (hexadecimal) | |
| Target PLC No. | | | |

*1 For details, refer to  Page 205 Module type list.

■Module type list

Select the module type of the communication destination. For the compatible versions of each type, refer to  Page 288 Added and Changed Functions.

○: Compatible —: Incompatible

| Module type | Communication destination | Compatibility on client side | | | Communication specification |
|--|---|------------------------------|----------------|-----------------|--|
| | | CPU module | | Ethernet module | |
| | | FX5UJ | FX5U/ FX5UC | | |
| MELSEC iQ-R (Built-in Ethernet) | RnCPU, RnENCPU*1, RnPCPU, RnSFPCPU, RnPSFPCPU | — | ○ | — | MELSOFT connection (protocol: UDP) |
| MELSEC-Q (Built-in Ethernet) | QnUD(P)VCPU, QnUDE(H)CPU | — | ○ | — | |
| MELSEC-L (Built-in Ethernet) | Built-in Ethernet port LCPU | — | ○ | — | |
| MELSEC iQ-F (Built-in Ethernet) | FX5 CPU module | ○ | ○ | ○ | |
| MELSEC iQ-F (Ethernet module) | Ethernet module | — | — | ○ | |
| MELSEC iQ-L (Built-in Ethernet) | LnHCPU | — | ○ | — | |
| MELSEC-FX3 (Ethernet Block/Adapter)*1 | FX3U-ENET-ADP, FX3U-ENET-L | — | ○ | — | MC protocol (A compatible 1E frame) [UDP, Binary] • Read (01H) • Write (03H) |
| SLMP Compatible Device (QnA Compatible 3E Frame) | • SLMP-compatible device (No serial number) • SLMP protocol 3E frame-compatible device | ○ | ○ | — | SLMP protocol (QnA compatible 3E frame) [UDP, Binary] • Read (0401H) • Write (1401H) |
| OMRON (CJ/CP series) | Supported model for SYSMAC CJ/CP series | — | ○ | — | FINS |
| KEYENCE (KV series) | Supported model for KV series | — | ○ | — | SLMP protocol (QnA compatible 3E frame) [UDP, Binary] • Read (0401H) • Write (1401H) |
| Panasonic (FP7 series) | Supported model for FP7 series | — | ○ | — | MEWTOCOL-7 |
| Panasonic (FP0H series) | Supported model for FP0 series | — | ○ | — | MEWTOCOL-COM |
| MODBUS/TCP-compatible device | MODBUS/TCP slave device | — | ○ | ○ | MODBUS/TCP |
| SIEMENS S7 series | Supported model for S7 series | — | ○ | — | S7 Communication |

*1 When the RnENCPU is used for the connection to the Ethernet port of the CPU part.

■IP address, TCP/UDP, port number, and own station port number

Set the IP address of the communication destination, communication protocol, port number, and own port number.

| Module type | IP address | TCP/UDP*1 | Port number | Own station port number | |
|--|----------------------------|-----------|-----------------------------|---|--|
| MELSEC iQ-R (Built-in Ethernet) | 0.0.0.1 to 223.255.255.254 | UDP | 5006 | A port number from 61696 to 65534 will be dynamically used for the own station. When the IP address of the communication destination is the same as that of MELSEC iQ-F (Ethernet module), the same own station port No. is used. | |
| MELSEC-Q (Built-in Ethernet) | | UDP | | | |
| MELSEC-L (Built-in Ethernet) | | UDP | | | |
| MELSEC iQ-F (Built-in Ethernet) | | UDP | 5565 | | |
| MELSEC iQ-F (Ethernet module) | | UDP | 5556 | | |
| MELSEC iQ-L (Built-in Ethernet) | | UDP | 5006 | | |
| MELSEC-FX3 (Ethernet Block/Adapter) | | UDP | 1025 to 5548, 5552 to 65534 | | 1 to 5548, 5570 to 65534 In MELSEC iQ-F (Ethernet module), the own station port number can be duplicated. |
| SLMP Compatible Device (QnA Compatible 3E Frame) | | UDP | 1 to 65534 | | |
| OMRON (CJ/CP series) | | UDP | | | |
| KEYENCE (KV series) | | UDP | | | |
| Panasonic (FP7 series) | | UDP | | | |
| Panasonic (FP0H series) | | UDP | | | |
| MODBUS/TCP-compatible device | | TCP | | | |
| SIEMENS S7 series | TCP | 102 | | | |

*1 Names of protocols supporting individual device types are displayed. (Fixed)

Point

- The communication destination is an SLMP compatible device (QnA compatible 3E frame), a parameter error will occur at the time when the same port number is set as the port number to be used for the own station. Therefore, when using the simple CPU communication function, do not set 61696 to 65534 to the own station port numbers for establishing a connection instruction (SP.SOCOPEN) of the socket communication function. If set, the instruction may not be completed properly.
- Port numbers 1 to 1023 are typically reserved port numbers (WELL KNOWN PORT NUMBERS) and 61440 to 65534 are used by other communication functions, so 1024 to 5548 or 5570 to 61439 should be used for the own station port numbers.
- When specifying the same communication destination in multiple settings, set different own station port numbers.
- Set the own station port number for the simple CPU communication function so that the own station port number and those functions of socket communication, SLMP communication, MODBUS/TCP functions, communication protocol support functions, web server functions, etc. do not overlap with each other.
- When using the simple CPU communication function and the following function simultaneously, do not specify the own station port number for the function to be used.

[For TCP communication]

- File transfer function (FTP server): 20 (14H), 21 (15H)
- Web server function: 80 (50H)*1 *2
- MODBUS/TCP communication (slave station): 502 (1F6H) *1

[For UDP communication]

- Time setting function (SNTP client): 123 (7BH)
- SLMP function: 61440 (F000H), 61441 (F001H)
- CC-Link IE field network Basic: 61450 (F00AH)

*1 Port No. can be changed. (Default: 80 (File transfer function), 502 (MODBUS/TCP communication (slave station)))

*2 When changing the port number by the Web server function, set the port number so that the port number and the own port number to be set by the simple CPU communication function do not overlap each other.

Communication Destination Setting

The settings differ depending on the communication destination.

Specify the protocol and own station port number for each setting to be the same as those set in the simple CPU communication settings.

| Communication destination | Necessary setting |
|--|---|
| <ul style="list-style-type: none"> MELSEC iQ-R (Built-in Ethernet) MELSEC-Q (Built-in Ethernet) MELSEC-L (Built-in Ethernet) MELSEC iQ-F (Built-in Ethernet) MELSEC iQ-F (Ethernet module) MELSEC iQ-L (Built-in Ethernet) | Setting is not required |
| MELSEC-FX3 (Ethernet Block/Adapter) | Configure the settings so that the MC protocol (A-compatible 1E frame) can be used on the external device side. (Refer to manual for the device used) Set the communication data code to "Binary". Set the protocol to "UDP". |
| SLMP Compatible Device (QnA Compatible 3E Frame) | Configure the settings so that the SLMP (QnA-compatible 3E frame) can be used on the SLMP-compatible device side. (Refer to manual for the device used) Set the communication data code to "Binary". Set the protocol to "UDP". |
| OMRON (CJ/CP series) | In the "IP address table," set IP and node addresses which are assigned to the FX5U/5UC CPU module Set "1" to the "node address." For other settings, refer to references of the communication target device's manuals. |
| <ul style="list-style-type: none"> KEYENCE (KV series) Panasonic (FP7 series) Panasonic (FP0H series) SIEMENS S7 series | Refer to manuals of individual devices. |
| MODBUS/TCP-compatible device | Configure the settings so that the MODBUS/TCP can be used on the MODBUS/TCP compatible device. (Refer to manual for the device used) Set the communication data code to "Binary". Set the protocol to "TCP". |

13

Option (hexadecimal)

When the communication destination is "MODBUS/TCP connection device", specify the value to be set in the module ID of MODBUS application header. This item is supported only for Ethernet module.

| Item | Setting range |
|----------------------|--|
| Option (hexadecimal) | <ul style="list-style-type: none"> When communicating with MODBUS/TCP connection device 00H or FFH When communicating with MODBUS RTU/ASCII compatible device through gateway device <ul style="list-style-type: none"> 00H: Broadcast 01H to F7H: Station No. of MODBUS RTU/ASCII compatible device (Default: 00H) |

Target PLC No.

When the communication destination is "MELSEC iQ-R (Built-in Ethernet)" and a multiple CPU system is used, specify the "Target PLC No." of the communication destination.

| Item | Setting range |
|----------------|--|
| Target PLC No. | <ul style="list-style-type: none"> Not Specified CPU No.1 to CPU No.4 (Default: Not Specified) |

Device Setting

Set the device that will transmit and receive data with the communication destination.

| Item | | Description | Setting range |
|-------------|--------|--|--|
| Bit device | Points | Set the bit device type, start number and end number of the "transmission source" and the bit device type and start number of the "transmission destination". The number of points will be automatically displayed after these parameters are input. | The devices that can be specified as transmission source and transmission destination vary depending on the communication destination. Bit device and word device can be set together for each setting number. ^{*1} |
| | Type | | |
| | Start | | |
| | End | | |
| Word device | Points | Set the word device type, start number and end number of the "transmission source" and the word device type and start number of the "transmission destination". The number of points will be automatically displayed after these parameters are input. | |
| | Type | | |
| | Start | | |
| | End | | |

*1 For the devices that can be specified, refer to the following.

- ☞ Page 209 Devices that can be specified on the own station
- ☞ Page 209 When the communication destination is the MELSEC iQ-R (Built-in Ethernet)
- ☞ Page 211 When the communication destination is the MELSEC-Q (Built-in Ethernet)
- ☞ Page 212 When the communication destination is the MELSEC-L (Built-in Ethernet)
- ☞ Page 210 When the communication destination is the MELSEC iQ-F (Built-in Ethernet)
- ☞ Page 211 When the communication destination is the MELSEC iQ-F (Ethernet module)
- ☞ Page 210 When the communication destination is the MELSEC iQ-L (Built-in Ethernet)
- ☞ Page 212 When the communication destination is the MELSEC-FX3 (Ethernet Block/Adapter)
- ☞ Page 212 When the communication destination is an SLMP-compatible device (QnA-compatible 3E frame)
- ☞ Page 213 When the communication destination is the OMRON (CJ/CP series)
- ☞ Page 213 When the communication destination is the KEYENCE (KV series)
- ☞ Page 214 When the communication destination is the Panasonic (FP7 series)
- ☞ Page 214 When the communication destination is the Panasonic (FP0H series)
- ☞ Page 214 When the communication destination is the MODBUS/TCP-compatible device
- ☞ Page 215 When the communication destination is the SIEMENS S7 series

■Number of device points

Set the number of device points to the total number of all settings (8192 words maximum).

Depending on communication target devices, the number of points per setting is as follows.

| Communication destination | Number of points per setting |
|---|---|
| <ul style="list-style-type: none"> • MELSEC iQ-R (Built-in Ethernet) • MELSEC-Q (Built-in Ethernet) • MELSEC-L (Built-in Ethernet) • MELSEC iQ-F (Built-in Ethernet) • MELSEC iQ-F (Ethernet module) • MELSEC iQ-L (Built-in Ethernet) • SLMP Compatible Device (QnA Compatible 3E Frame) • KEYENCE (KV series) | 512 words maximum (bit device 8192 points maximum, word device 512 points maximum) |
| MELSEC-FX3 (Ethernet Block/Adapter) | Read: 96 words maximum (bit device 512 points maximum, word device 64 points maximum) Write: 74 words maximum (bit device 160 points maximum, word device 64 points maximum) |
| OMRON (CJ/CP series) | [When using SYSMAC CJ series] 512 words maximum (bit device 8192 points maximum, word device 512 points maximum) [When using SYSMAC CP series] 496 words maximum (bit device 7936 points maximum, word device 496 points maximum) |
| Panasonic (FP7 series) | 496 words maximum (bit device 7936 points maximum, word device 496 points maximum) |
| Panasonic (FP0H series) | 256 words maximum (bit device 2048 points maximum, word device 128 points maximum) |
| MODBUS/TCP-compatible device | Read: 250 words maximum (bit device 2000 points maximum, word device 125 points maximum) Write: 246 words maximum (bit device 1968 points maximum, word device 123 points maximum) |
| SIEMENS S7 series | [When using S7-1500] 384 words maximum (bit device 2048 points maximum, word device 256 points maximum) [When using S7-1200, S7-200 SMART] Read: 222 words maximum (bit device 1776 points maximum, word device 111 points maximum) Write: 212 words maximum (bit device 1696 points maximum, word device 106 points maximum) |

■Devices that can be specified on the own station

For devices that can be specified, the size of a device is specified in units of 16 points for a bit device and 1 point for a word device. Use 0 or multiples of 16 to specify the device number of a bit device.

Furthermore, described below is the device range when device types with the maximum number of points are used.

| Type | Applicable device | | | | |
|-------------|-------------------|------------------|-----------------------|---|----------------------------|
| | Symbol | Range | | | |
| | | FX5UJ CPU module | FX5U/FX5UC CPU module | Ethernet module | |
| Bit device | X | 0 to 1777 | | The same device range as that of the connected CPU module | |
| | Y | 0 to 1777 | | | |
| | M | 0 to 7679 | 0 to 32767 | | |
| | L | 0 to 7679 | 0 to 32767 | | |
| | B | 0H to 7FFH | 0H to 7FFFH | | |
| | SB | 0H to 7FFH | 0H to 7FFFH | | |
| | SM | 0 to 9999 | | | |
| Word device | D | 0 to 7999 | | | |
| | W | 0H to 3FFH | 0H to 7FFFH | | |
| | R | 0 to 32767 | | | |
| | SW | 0H to 3FFH | 0H to 7FFFH | | |
| | SD | 0 to 11999 | | | |
| | G | — | | | 6400 to 8447 ^{*1} |

*1 Un\G6400 to Un\G8447 can be used as devices for simple CPU communication function.

Point

Set the device within the device range specified in "Device/Label Memory Area Setting." (MELSEC iQ-F FX5 User's Manual (Application))

■When the communication destination is the MELSEC iQ-R (Built-in Ethernet)

For devices that can be specified, the size of a device is specified in units of 16 points for a bit device and 1 point for a word device. Use 0 or multiples of 16 to specify the device number of a bit device.

Furthermore, described below is the device range when device types with the maximum number of points are used.

| Type | Applicable device | | |
|-------------|-------------------|----------------|--|
| | Symbol | Range | Remarks |
| Bit device | X | 0H to 2FFFFH | — |
| | Y | 0H to 2FFFFH | — |
| | M | 0 to 161882111 | A local device cannot be specified. |
| | L | 0 to 32767 | — |
| | B | 0H to 9A61FFFH | — |
| | SB | 0H to 9A61FFFH | — |
| | SM | 0 to 4095 | — |
| Word device | D | 0 to 10117631 | A local device cannot be specified. |
| | W | 0H to 9A61FFFH | — |
| | SW | 0H to 9A61FFFH | — |
| | SD | 0 to 4095 | — |
| | R | 0 to 32767 | Data is read or written following the "File register settings" of the communication destination. |
| | ZR | 0 to 10027007 | Data is read or written following the "File register settings" of the communication destination. |
| | RD | 0 to 1048575 | — |

Point

Set the device within the device range specified in "Device/Label Memory Area Setting."

■When the communication destination is the MELSEC iQ-L (Built-in Ethernet)

For devices that can be specified, the size of a device is specified in units of 16 points for a bit device and 1 point for a word device. Use 0 or multiples of 16 to specify the device number of a bit device.

Furthermore, described below is the device range when device types with the maximum number of points are used.

| Type | Applicable device | | |
|-------------|-------------------|---------------|--|
| | Symbol | Range | Remarks |
| Bit device | X | 0H to 2FFFFH | — |
| | Y | 0H to 2FFFFH | — |
| | M | 0 to 14065663 | A local device cannot be specified. |
| | L | 0 to 32767 | — |
| | B | 0H to D69FFFH | — |
| | SB | 0H to D69FFFH | — |
| | SM | 0 to 4095 | — |
| Word device | D | 0 to 879103 | A local device cannot be specified. |
| | W | 0H to D69FFFH | — |
| | SW | 0H to D69FFFH | — |
| | SD | 0 to 4095 | — |
| | R | 0 to 32767 | Data is read or written following the "File register settings" of the communication destination. |
| | ZR | 0 to 819199 | Data is read or written following the "File register settings" of the communication destination. |
| | RD | 0 to 1048575 | — |



Set the device within the device range specified in "Device/Label Memory Area Setting."

■When the communication destination is the MELSEC iQ-F (Built-in Ethernet)

For devices that can be specified, the size of a device is specified in units of 16 points for a bit device and 1 point for a word device. Use 0 or multiples of 16 to specify the device number of a bit device.

Furthermore, described below is the device range when device types with the maximum number of points are used.

| Type | Applicable device | | | Remarks |
|-------------|-------------------|------------------|-----------------------|-------------------------|
| | Symbol | Symbol | | |
| | | FX5UJ CPU module | FX5U/FX5UC CPU module | |
| Bit device | X | 0 to 1777 | | Octal notation is used. |
| | Y | 0 to 1777 | | Octal notation is used. |
| | M | 0 to 7679 | 0 to 32767 | — |
| | L | 0 to 7679 | 0 to 32767 | — |
| | B | 0H to 7FFH | 0H to 7FFFH | — |
| | SB | 0H to 7FFH | 0H to 7FFFH | — |
| | SM | 0 to 9999 | | — |
| Word device | D | 0 to 7999 | | — |
| | W | 0H to 3FFH | 0H to 7FFFH | — |
| | SW | 0H to 3FFH | 0H to 7FFFH | — |
| | SD | 0 to 11999 | | — |
| | R | 0 to 32767 | | — |



Set the device within the device range specified in "Device/Label Memory Area Setting." (MELSEC iQ-F FX5 User's Manual (Application))


■When the communication destination is the MELSEC iQ-F (Ethernet module)

For devices that can be specified, the size of a device is specified in units of 16 points for a bit device and 1 point for a word device. Use 0 or multiples of 16 to specify the device number of a bit device.

Furthermore, described below is the device range when device types with the maximum number of points are used.

| Type | Applicable device | | | Remarks |
|-------------|-------------------|------------------------------------|---|--|
| | Symbol | Symbol | | |
| | | When FX5UJ CPU module is connected | When FX5U/FX5UC CPU module is connected | |
| Bit device | X | 0 to 1777 | | Octal notation is used. |
| | Y | 0 to 1777 | | Octal notation is used. |
| | M | 0 to 7679 | 0 to 32767 | The same device range as that of the connected CPU module. |
| | L | 0 to 7679 | 0 to 32767 | The same device range as that of the connected CPU module. |
| | B | 0H to 7FFH | 0H to 7FFFH | The same device range as that of the connected CPU module. |
| | SB | 0H to 7FFH | 0H to 7FFFH | The same device range as that of the connected CPU module. |
| | SM | 0 to 9999 | | — |
| Word device | D | 0 to 7999 | | — |
| | W | 0H to 3FFH | 0H to 7FFFH | The same device range as that of the connected CPU module. |
| | SW | 0H to 3FFH | 0H to 7FFFH | The same device range as that of the connected CPU module. |
| | SD | 0 to 11999 | | — |
| | R | 0 to 32767 | | — |

Point

Set the device within the device range specified in "Device/Label Memory Area Setting." ( MELSEC iQ-F FX5 User's Manual (Application))

■When the communication destination is the MELSEC-Q (Built-in Ethernet)

For devices that can be specified, the size of a device is specified in units of 16 points for a bit device and 1 point for a word device. Use 0 or multiples of 16 to specify the device number of a bit device.

Furthermore, described below is the device range when device types with the maximum number of points are used.

| Type | Applicable device | | |
|-------------|-------------------|---------------|--|
| | Symbol | Range | Remarks |
| Bit device | X | 0H to 1FFFH | — |
| | Y | 0H to 1FFFH | — |
| | M | 0 to 61439 | — |
| | L | 0 to 32767 | — |
| | B | 0H to EFFFH | — |
| | SB | 0H to 7FFFH | — |
| | SM | 0 to 2047 | — |
| Word device | D | 0 to 4910079 | The maximum number of points when D (extended data register) or W (extended link register) is set. |
| | W | 0H to 4AEBFFH | The maximum number of points when D (extended data register) or W (extended link register) is set. |
| | SW | 0H to 7FFFH | — |
| | SD | 0 to 2047 | — |
| | R | 0 to 32767 | Data is read or written following the "File register settings" of the communication destination. |
| | ZR | 0 to 4849663 | Data is read or written following the "File register settings" of the communication destination. |

Point

Set the device within the device range specified in "Device/Label Memory Area Setting."

■When the communication destination is the MELSEC-L (Built-in Ethernet)

For devices that can be specified, the size of a device is specified in units of 16 points for a bit device and 1 point for a word device. Use 0 or multiples of 16 to specify the device number of a bit device.

Furthermore, described below is the device range when device types with the maximum number of points are used.

| Type | Applicable device | | |
|-------------|-------------------|--------------|--|
| | Symbol | Range | Remarks |
| Bit device | X | 0H to 1FFFH | — |
| | Y | 0H to 1FFFH | — |
| | M | 0 to 61439 | — |
| | L | 0 to 32767 | — |
| | B | 0H to EFFFH | — |
| | SB | 0H to 7FFFH | — |
| | SM | 0 to 2047 | — |
| Word device | D | 0 to 421887 | The maximum number of points when D (extended data register) or W (extended link register) is set. |
| | W | 0H to 66FFFH | The maximum number of points when D (extended data register) or W (extended link register) is set. |
| | SW | 0H to 6FFFH | — |
| | SD | 0 to 2047 | — |
| | R | 0 to 32767 | Data is read or written following the "File register settings" of the communication destination. |
| | ZR | 0 to 393215 | Data is read or written following the "File register settings" of the communication destination. |



Set the device within the device range specified in "Device/Label Memory Area Setting."

■When the communication destination is the MELSEC-FX3 (Ethernet Block/Adapter)

Applicable devices differ depending on the device. For applicable devices, refer to the manual for the device used.

The accessible device range of the communication destination is the range which can be specified with the commands Read (01H) and Write (03H) of the MC protocol (A compatible 1E frame).

| Type | Applicable device | | |
|-------------|-------------------|-------------------------|-------------------------|
| | Symbol | Range | Remarks |
| Bit device | X | 0 to 377 | Octal notation is used. |
| | Y | 0 to 377 | Octal notation is used. |
| | M | 0 to 7679, 8000 to 8511 | — |
| Word device | D | 0 to 7999, 8000 to 8511 | — |
| | R | 0 to 32767 | — |

■When the communication destination is an SLMP-compatible device (QnA-compatible 3E frame)

Applicable devices differ depending on the device. For applicable devices, refer to the manual for the device used.

The device types that can be set are X, Y, M, L, B, SB, SM, D, W, SW, SD, R, and ZR. The accessible device range of the communication destination is the range which can be specified with the subcommands 0000H of the SLMP (QnA compatible 3E frame).

■When the communication destination is the OMRON (CJ/CP series)

For devices that can be specified, the size of a device is specified in units of 1 point.

| Type | Applicable device | | |
|---|-------------------|------------|---|
| | Symbol | Range | Remarks |
| Bit device (The device is displayed in units of words. One point corresponds to one word.) | .. | 0 to 6143 | — |
| | AR | 0 to 447 | The setting is available only when "Read" is set to "Communication Pattern". |
| | AR | 448 to 959 | — |
| | HR | 0 to 511 | — |
| | WR | 0 to 511 | — |
| Word device | DM | 0 to 32767 | — |
| | TIM | 0 to 4095 | — |
| | CNT | 0 to 4095 | — |
| | DR | 0 to 15 | — |
| | TK | 0 to 31 | The setting is available only when "Read" is set to "Communication Pattern". The upper 8 bits are zero (fixed). (because memory areas are treated byte by byte) |
| | EM | 0 to 32767 | The number of points and whether devices are usable depend on the types and settings of CPU modules used. |
| | EMn_ | 0 to 32767 | The number of points and whether devices are usable depend on the types and settings of CPU modules used. "n" represents 0H to FH or 10H to 18H. (25 devices in total) |

■When the communication destination is the KEYENCE (KV series)

For devices that can be specified, the size of a device is specified in units of 16 points for a bit device and 1 point for a word device.

To format bit devices other than "B," their lower two digits are specified bit by bit and their other upper digits are specified word by word.

| Type | Applicable device | | |
|---|-------------------|-------------|--|
| | Symbol | Range | Remarks |
| Bit device (The device is displayed in units of bits. One point corresponds to one bit.) | R | 0 to 199915 | Specify 00 in the last two digits for the start. |
| | B | 0H to 7FFFH | Specify 0 or a multiple of 16 for the start. |
| | MR | 0 to 399915 | Specify 00 in the last two digits for the start. |
| | LR | 0 to 99915 | Specify 00 in the last two digits for the start. |
| | CR | 0 to 8915 | Specify 00 in the last two digits for the start. |
| | T | 0 to 3999 | — |
| | C | 0 to 3999 | — |
| Word device | CM | 0 to 8999 | — |
| | DM | 0 to 65534 | — |
| | EM | 0 to 65534 | — |
| | FM | 0 to 32767 | — |
| | ZF | 0 to 524287 | — |
| | W | 0H to 7FFFH | — |
| | T | 0 to 3999 | — |
| | C | 0 to 3999 | — |

■When the communication destination is the Panasonic (FP7 series)

For devices that can be specified, the size of a device is specified in units of 1 point.

| Type | Applicable device | | |
|---|-------------------|------------------------|---|
| | Symbol | Range | Remarks |
| Bit device (The device is displayed in units of words. One point corresponds to one word.) | WX | 0 to 511 | — |
| | WY | 0 to 511 | — |
| | WR | 0 to 2047 | — |
| | WL | 0 to 1023 | — |
| | _WX | 001000000 to 468000511 | The upper three digits of the device number shows the program block number, and the lower six digits shows the device number. |
| | _WY | 001000000 to 468000511 | The upper three digits of the device number shows the program block number, and the lower six digits shows the device number. |
| | _WR | 001000000 to 468002047 | The upper three digits of the device number shows the program block number, and the lower six digits shows the device number. |
| | _WL | 001000000 to 468001023 | The upper three digits of the device number shows the program block number, and the lower six digits shows the device number. |
| Word device | LD | 0 to 16383 | — |
| | DT | 0 to 999423 | — |
| | _LD | 001000000 to 468016383 | The upper three digits of the device number shows the program block number, and the lower six digits shows the device number. |
| | _DT | 001000000 to 468065534 | The upper three digits of the device number shows the program block number, and the lower six digits shows the device number. |

■When the communication destination is the Panasonic (FP0H series)

For devices that can be specified, the size of a device is specified in units of 1 point.

| Type | Applicable device | | |
|---|-------------------|------------|--|
| | Symbol | Range | Remarks |
| Bit device (The device is displayed in units of words. One point corresponds to one word.) | WX | 0 to 109 | The setting is available only when "Read" is set to "Communication Pattern". |
| | WY | 0 to 109 | — |
| | WR | 0 to 511 | — |
| | WL | 0 to 127 | — |
| Word device | LD | 0 to 255 | — |
| | DT | 0 to 65532 | — |

■When the communication destination is the MODBUS/TCP-compatible device

For devices that can be specified, the size of a device is specified in units of 16 points for a bit device and 1 point for a word device.

Set individual device ranges by "the lower five digits of target MODBUS devices - 1" to comply with specifications of MODBUS/TCP-compatible devices.

Applicable devices differ depending on the device. For applicable devices, refer to the manual for the device used.

| Type | Applicable device | | |
|-------------|-------------------|------------|--|
| | Type | Range | Remarks |
| Bit device | Coil | 0 to 65535 | — |
| | Input | 0 to 65535 | The setting is available only when "Read" is set to "Communication Pattern". |
| Word device | Input Register | 0 to 65535 | The setting is available only when "Read" is set to "Communication Pattern". |
| | Holding Register | 0 to 65535 | — |

■When the communication destination is the SIEMENS S7 series

For devices that can be specified, the size of a device is specified in units of 1 points for a bit device and 2 point for a word device.

| Type | Applicable device | | |
|--|-------------------|------------------|--|
| | Symbol | Range | Remarks |
| Bit device (One point is equivalent to eight points in word units.) | I | 0 to 255 | Start digit: An even number must be specified. Last digit: An odd number must be specified. |
| | Q | 0 to 255 | Start digit: An even number must be specified. Last digit: An odd number must be specified. |
| | M | 0 to 255 | Start digit: An even number must be specified. Last digit: An odd number must be specified. |
| Word device | DB | 001000 to 255511 | Upper 3 digits: Block No., Lower 3 digits: Address range. An even number must be specified to the start of the lower 3 digits and an odd number must be specified to the last of the lower 3 digits. (The size for 1 point is 1 byte. Two points are required to set for word device.) |

Point

Access is executed in units of words. The upper byte and lower byte of reading/writing value may be interchanged depending on the setting in the communication destination.

13

Precautions

Do not perform settings which cross the DB, such as DB001510 to DB002509.

Operation during the simple CPU communication

■Operation of the own station

Since the own station operates without depending on the operating status of the Ethernet-equipped module, the simple CPU communication is performed even when the CPU module is in the STOP state because of stop error. When an initial communication error occurs, communications of the corresponding setting numbers are stopped and the setting numbers without an error will operate.

If the communication destination sends a response during END processing of the CPU module, the response will be reflected directly in the device by the END processing during which the response is received. In the case of an SLMP compatible device (QnA compatible 3E frame), MELSEC-FX3 (Ethernet Block/Adapter), PLC manufactured by other company, MODBUS/TCP-compatible device data inconsistency may occur.

■Operation of the communication destination

The operation differs depending on the communication destination.

| Communication destination | Data operation in one setting |
|--|---|
| <ul style="list-style-type: none"> MELSEC iQ-R (Built-in Ethernet) MELSEC-Q (Built-in Ethernet) MELSEC-L (Built-in Ethernet) MELSEC iQ-F (Built-in Ethernet) MELSEC iQ-F (Ethernet module) MELSEC iQ-L (Built-in Ethernet) | Since the data in one setting is communicated at the same timing, data inconsistency will not occur. |
| <ul style="list-style-type: none"> MELSEC-FX3 (Ethernet Block/Adapter) SLMP Compatible Device (QnA Compatible 3E Frame) OMRON (CJ/CP series) KEYENCE (KV series) Panasonic (FP7 series) Panasonic (FP0H series) MODBUS/TCP-compatible device SIEMENS S7 series | The data specified with a bit device or a word device in one setting is communicated at the same timing. However, if both a bit device and a word device are set for the same setting number, the data of each device may be communicated at a different timing. |

Communication Time-out Period/Communication Retry Count/Monitoring Time At Error

Set the conditions to cause communication errors.

| Item | Description | Setting range |
|-----------------------------------|---|--|
| Communication Time-out Period(ms) | Set the time after which an error will occur or a retry will start when no response is sent from the communication destination. The set time must satisfy the following condition: "communication time-out period \geq execution interval." | 10 to 65535 ms (in increments of 1 ms) ^{*1} (Default: 1000 ms) |
| Communication Retry Count | Set the number of times the wait operation is repeated for the communication time-out period when an error response is sent from the communication destination or no response is sent within the communication time-out period. | 0 to 255 times ^{*2} (Default: 3 times) |
| Monitoring Time At Error(s) | Set the monitoring time after occurrence of communication error. | 1 to 300 s (Default: 30 s) |

*1 When the communication destination is an "MELSEC-FX3 (Ethernet Block/Adapter)", "SLMP Compatible Device (QnA Compatible 3E Frame)" and "KEYENCE (KV series)", the setting is fixed to 65535 ms.

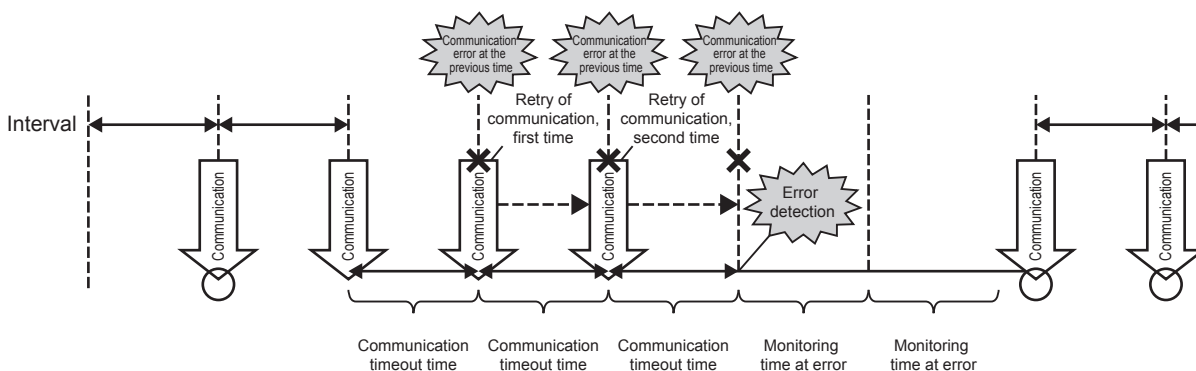
*2 When the communication destination is an "MELSEC-FX3 (Ethernet Block/Adapter)", "SLMP Compatible Device (QnA Compatible 3E Frame)" and "KEYENCE (KV series)", the setting is fixed to 0 time.

When an error response is received from the communication destination or no response is given within the communication time-out period, retry (retransmission) will be attempted. After the communication time-out period has passed since the set number of retries was performed, communication will be made at a regular interval for the Monitoring Time at Error(s).

If the communication destination responds after these communications, the communications at the execution interval specified with the parameters are restarted.

Ex.

Error detection timing when 2 is set for the communication retry count



Point

The error is detected if the communication destination does not respond or communications fail by when the following time has passed: $(\text{communication retry count} + 1) \times \text{communication time-out period}$

Latency Time

Set the wait time before the communication is started.

| Item | Description | Setting range |
|--------------|---|---|
| Latency Time | Set the wait time from completion of startup of the Ethernet-equipped module to start of communication. The time cannot be changed for each setting number because it is a common setting of the simple CPU communication function. | 0 to 255s (in increments of 1 s) (Default: 0s) |

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 communication destination is ready for the communications

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

13.2 Checking the simple CPU communication status

The simple CPU communication status can be checked with the device or diagnostic functions.

Checking with the device

■CPU module

The simple CPU communication status can be checked with the special device or diagnostic functions.

| Item | Device No. | Remarks |
|--|-------------------------------|---|
| Simple PLC Communication execution state | SD10356, SD10357 | SD10356.b0: Setting No.1 to SD10356.b15: Setting No.16 SD10357.b0: Setting No.17 to SD10357.b15: Setting No.32 ^{*1} |
| Simple PLC Communication ready flag | SD10358, SD10359 | SD10358.b0: Setting No.1 to SD10358.b15: Setting No.16 SD10359.b0: Setting No.17 to SD10359.b15: Setting No.32 ^{*1} |
| Simple CPU communication status | 0H: Unset | SD10380 to SD10411 SD10380: Setting No.1 to SD10395: Setting No.16 SD10396: Setting No.17 to SD10411: Setting No.32 ^{*1} |
| | 1H: Preparing | |
| | 3H: Communicating | |
| | 4H: Communication stop | |
| | 5H: Retry being executed | |
| | 6H: Monitoring | |
| | AH: Communications impossible | |
| Simple PLC Communication error code | SD10412 to SD10443 | SD10412: Setting No.1 to SD10427: Setting No.16 SD10428: Setting No.17 to SD10443: Setting No.32 ^{*1} |
| Simple PLC Communication execution interval | SD10444 to SD10475 | SD10444: Setting No.1 to SD10459: Setting No.16 SD10460: Setting No.17 to SD10475: Setting No.32 ^{*1} |
| Simple CPU communication error response code ^{*1} | SD10476 to SD10507 | SD10476: Setting No.1 to SD10591: Setting No.16 SD10492: Setting No.17 to SD10507: Setting No.32 |

*1 Available only in the FX5U/FX5UC CPU module.

■Ethernet module

The simple CPU communication status can be confirmed by checking the status stored in the appropriate setting No. on the following buffer memory.

| Item | Device No. | Remarks |
|------------------------------------|--------------------------|---|
| Execution Status flag | Un\G312, Un\G313 | Un\G312.b0: Setting No.1 to Un\G312.b15: Setting No.16 Un\G313.b0: Setting No.17 to Un\G313.b15: Setting No.32 |
| Ready | Un\G316, Un\G317 | Un\G316.b0: Setting No.1 to Un\G316.b15: Setting No.16 Un\G317.b0: Setting No.17 to Un\G317.b15: Setting No.32 |
| Simple CPU communication status | 0H: Unset | Un\G352 to Un\G383 Un\G352: Setting No.1 to Un\G367: Setting No.16 Un\G368: Setting No.17 to Un\G383: Setting No.32 |
| | 1H: Preparing | |
| | 2H: Waiting for request | |
| | 3H: Communicating | |
| | 4H: Communication stop | |
| | 5H: Retry being executed | |
| | 6H: Monitoring | |
| AH: Communications impossible | | |
| Simple CPU error code | Un\G416 to Un\G447 | Un\G416: Setting No.1 to Un\G431: Setting No.16 Un\G432: Setting No.17 to Un\G447: Setting No.32 |
| Execution interval (current value) | Un\G544 to Un\G575 | Un\G544: Setting No.1 to Un\G559: Setting No.16 Un\G560: Setting No.17 to Un\G575: Setting No.32 |
| Abnormal response code | Un\G480 to Un\G511 | Un\G480: Setting No.1 to Un\G495: Setting No.16 Un\G496: Setting No.17 to Un\G511: Setting No.32 |

■Communication stop/restart request when the communication setting is “Fixed”

[When the own station is an CPU module]

When the Simple CPU communication status (SD10380 to SD10411) of the corresponding setting number in the special device is either of the following, communications can be stopped by turning off and on the Simple PLC Communication stop request (SD10352 or SD10353) of the corresponding setting number. (If the status is other than the following, the stop request is invalid.)

- 3H: Communicating
- 5H: Retry being executed
- 6H: Monitoring

Communication can be restarted by turning OFF and ON the bit of the relevant setting No. in the Simple PLC Communication restart request (SD10354 or SD10355) when the Simple CPU communication status (SD10380 to SD10411) of the relevant setting No. on the special device is “4H: Communication stop” (If the status is other than “4H: Communication stop”, the restart request is invalid.)

[When the own station is an Ethernet module]

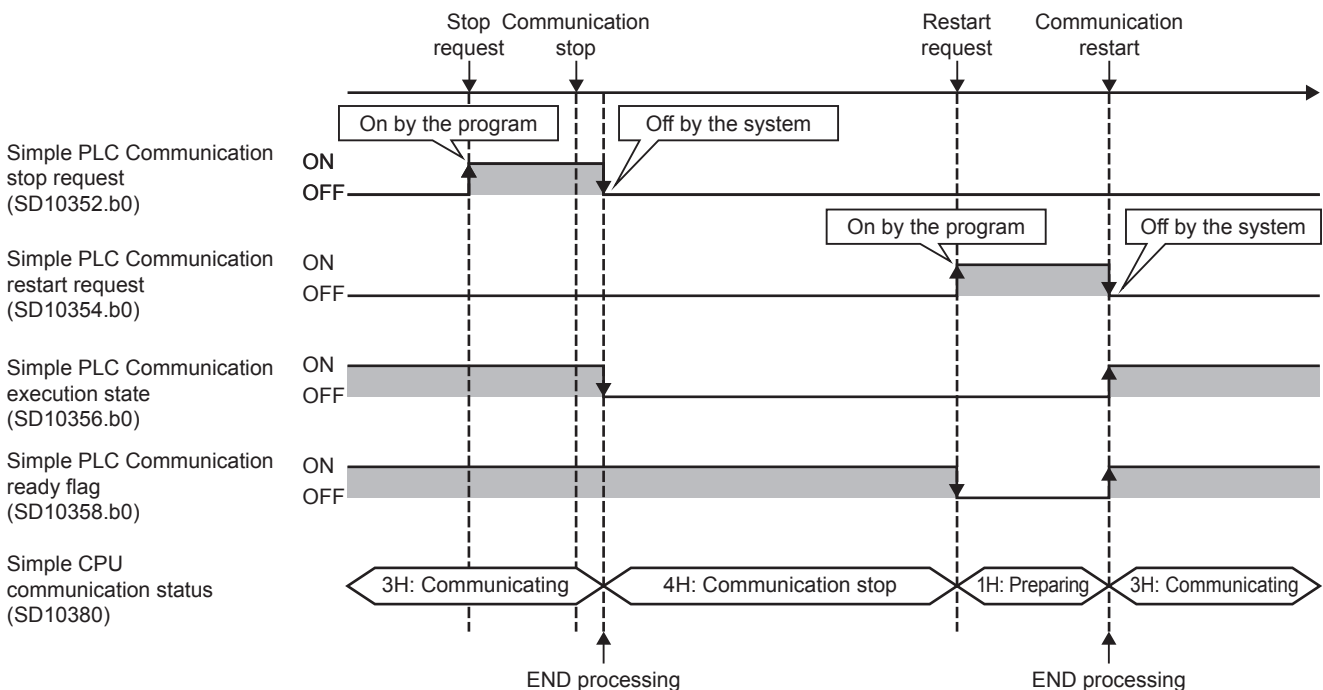
When the Simple CPU communication status (Un\G352 to Un\G383) of the corresponding setting number in the buffer memory is either of the following, communications can be stopped by turning off and on the Periodic communication stop request (Un\G304, Un\G305) of the corresponding setting number. (If the status is other than the following, the stop request is invalid.)

- 3H: Communicating
- 5H: Retry being executed
- 6H: Monitoring

Communication can be restarted by turning OFF and ON the bit of the relevant setting No. in the periodic communication restart request (Un\G308, Un\G309) when the Simple CPU communication status (Un\G352 to Un\G383) of the relevant setting No. on the buffer memory is “4H: Communication stop” (If the status is other than “4H: Communication stop”, the restart request is invalid.)

Ex.

The operation at communication stop and restart of the setting No. 1 (when the own station is a CPU module)



■ Starting communication when the communication setting is “Request”

The communication setting “Request” is applicable only to Ethernet modules.

[When the own station is an Ethernet module]

When the bit of the appropriate setting No. is turned OFF and ON by the “communication start at request” request (Un\G300, Un\G301) in the following case, communication can be started. (In any other cases, the start request will be invalid, or an error will occur.)

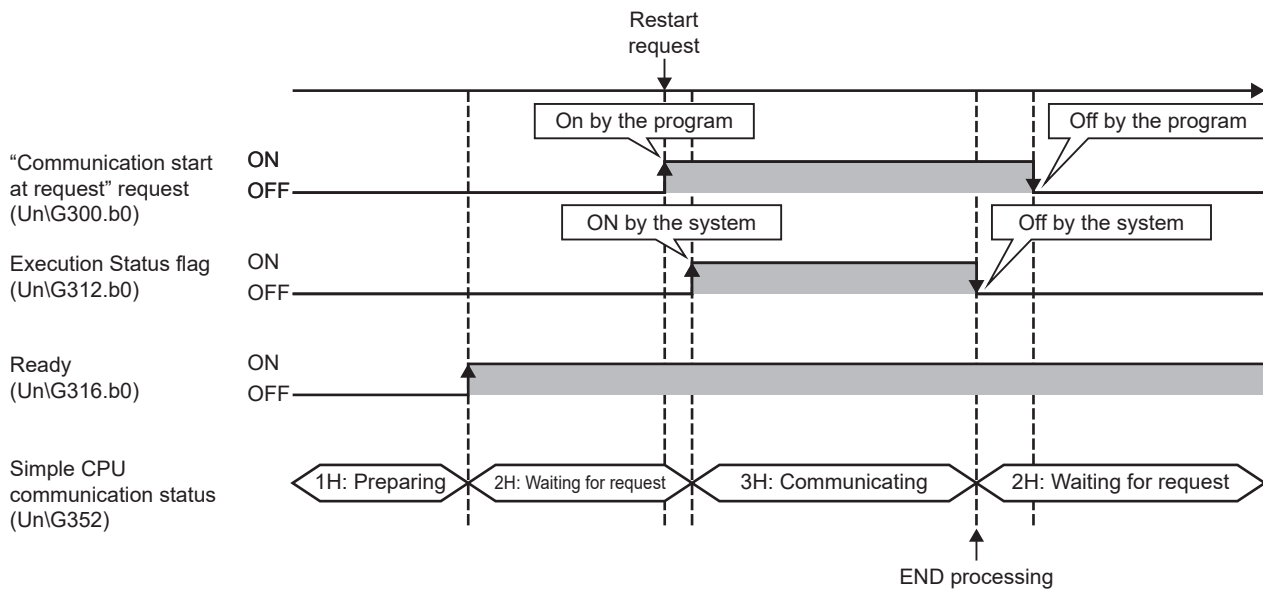
- Ready (Un\G316, Un\G317) of the appropriate setting No. is ON.
- The simple CPU communication status (Un\G352 to Un\G383) of the appropriate setting No. is “2H: Waiting for request.”

However, if the “communication start at request” request (Un\G300, Un\G301) is kept ON when the simple CPU communication status (Un\G352 to Un\G383) is “1H: Preparing,” communication will be started when the simple CPU communication status changes to “2H: Waiting for request.”

After the data is transmitted/received, the communication will be stopped. Next time you start the communication, turn OFF and ON the bit of the appropriate setting No. again by using the “communication start at request” request (Un\G300, Un\G301).

Ex.

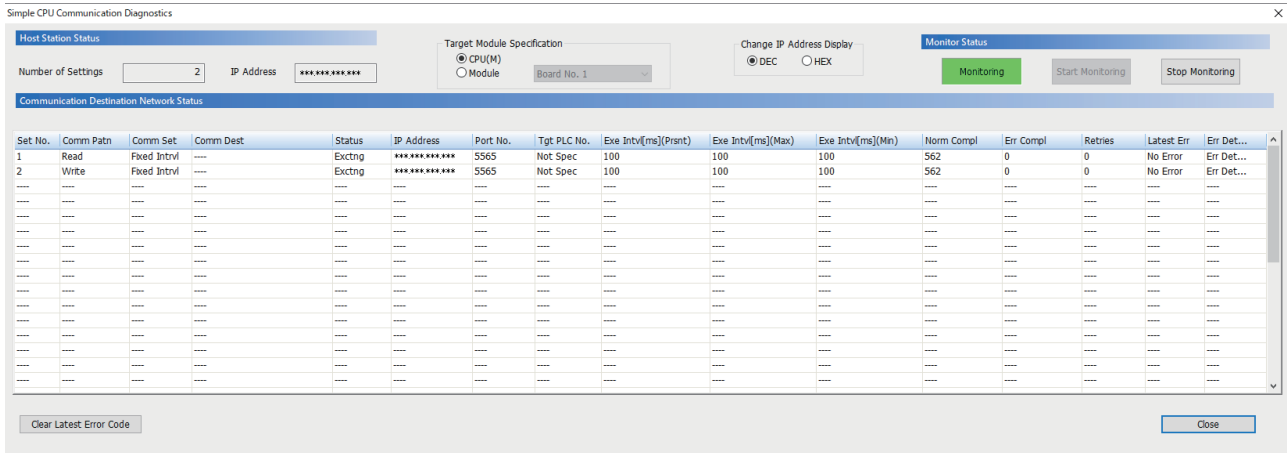
Operation for starting the communication of setting No.1 (when the own station is an Ethernet module)



Checking with "Simple CPU Communication Diagnostics"

The communication status of the simple CPU communication function can be checked with "Simple CPU Communication Diagnostics".

 [Diagnostics] → [Simple CPU Communication Diagnostics]



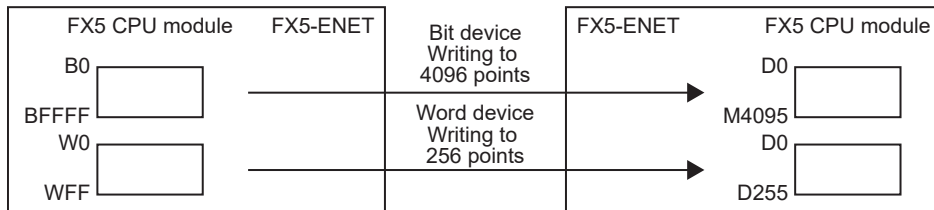
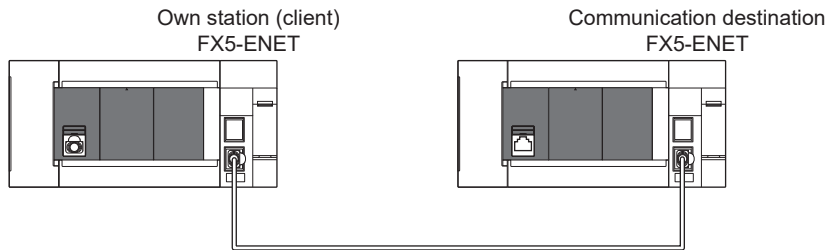
The module to be diagnosed can be switched in "Target Module Specification."

| Item | Description |
|----------------------------------|---|
| Set No. | Displays the setting number of the simple CPU communication. |
| Comm Patn | Displays the communication pattern (read/write) set with the parameters. |
| Comm Set | Displays the communication setting (fixed) set with the parameters. |
| Comm Dest | Displays the communication destination set with the parameters. |
| Status | Displays the simple CPU communication status (preparing, communicating, communication stop, retry being executed, monitoring, or communications impossible). |
| IP Address | Displays the IP address of the communication destination set with the parameters. |
| Port No. | Displays the port number of the communication destination set with the parameters. |
| Tgt PLC No. | Displays the target device No. set with the parameters. |
| Exe Intv [ms] (Prsnt) | Display the current value, the maximum value, and the minimum value of the execution interval in increments of ms. The values will not be updated while the retry is being executed or the error is being monitored. They will be "0" while communication is stopped. |
| Exe Intv [ms] (Max) | |
| Exe Intv [ms] (Min) | |
| Norm Compl | Display the accumulated counts of communications that has been completed successfully, completed with an error, and retried. • 0 to 4294967295: Accumulated count*1 |
| Err Compl | |
| Retries | |
| Latest Err | Displays the error code of the latest error occurred, or "-" when no error has occurred. The error remains displayed even after the communication status has changed to the state in which the communication can be performed normally. When another error occurs, it will be overwritten with a new one. |
| Err Det | Displays the descriptions and corrective actions for the error occurring in the selected setting number. |
| [Clear Latest Error Code] button | Clears the error code. Furthermore, this operation clears the special devices described below. • FX5U/5UC CPU module: Simple PLC Communication error code (SD10412 to SD10443) • FX5UJ CPU module: Simple PLC Communication error code (SD10412 to SD10427) • Ethernet module: Simple CPU error code (Un\G416 to Un\G447) |

*1 When the count exceeds 4294967295, counting is continued from 1 again.

13.3 Program Example

An example program of the simple CPU communication (communication at request) is shown below. FX5-ENET (own station) and FX5-ENET (communication destination) communicate with each other.



Simple CPU communication settings

The own station parameter settings (setting No.1) are shown below. Optionally set other parameters not listed below.

| Item | | Set value | |
|--|-----------------------|----------------|-------------------------------|
| Communication pattern | | Write | |
| Communication setting: Execution interval (ms) | Communication setting | | Request |
| Communication Destination(IP Address) | Transfer destination | Device type | MELSEC iQ-F (Ethernet module) |
| Bit devices | Transfer source | Type/start/end | B0 to BFFF |
| | Transfer destination | Type/start/end | M0 to M4095 |
| Word devices | Transfer source | Type/start/end | W0 to WFF |
| | Transfer destination | Type/start/end | D0 to D255 |

Program example

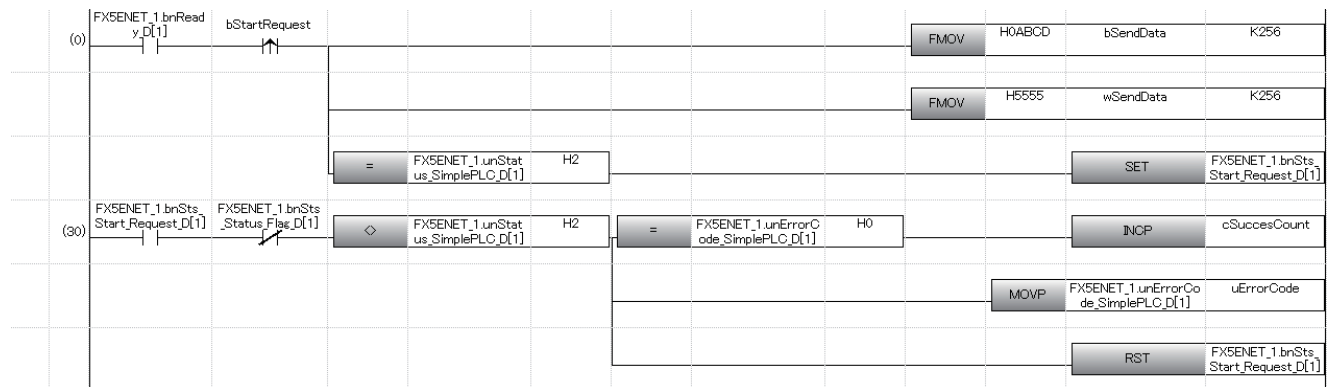
■Devices used

| Classification | Label name | Contents | Device |
|----------------|--------------------------------------|--|-----------|
| Module label | FX5ENET_1.bnSts_Start_Request_D[1] | "Communication start at request" request for simple CPU communication setting No.1 | Un/G300.0 |
| | FX5ENET_1.bnSts_Status_Flag_D[1] | Execution status flag for simple CPU communication setting No.1 | Un/G313.0 |
| | FX5ENET_1.bnReady_D[1] | Ready flag for simple CPU communication setting No.1 | Un/G317.0 |
| | FX5ENET_1.unStatus_SimplePLC_D[1] | Communication status storage location for simple CPU communication setting No.1 | Un/G383 |
| | FX5ENET_1.unErrorCode_SimplePLC_D[1] | Error code storage location for simple CPU communication setting No.1 | Un/G447 |

Global label/local label

| | Label Name | Data Type | Class | Assign (Device/Label) |
|---|---------------|---------------|-------------------|-----------------------|
| 1 | bStartRequest | Bit | VAR_GLOBAL | IM0 |
| 2 | bSendData | Word [Signed] | VAR_GLOBAL | K4B0 |
| 3 | wSendData | Word [Signed] | VAR_GLOBAL | WD |
| 4 | cSuccessCount | Counter | VAR_GLOBAL | CO |
| 5 | uErrorCode | Word [Signed] | VAR_GLOBAL_RETAIN | D1 000 |

■Program



13.4 Precautions

This section provides precautions for the simple CPU communication function.

Incomplete data reception and timeout

Because the load of Ethernet communication becomes heavy during the simple CPU communication, if other communication (MELSOFT connection, SLMP communication, or MODBUS/TCP communication) is simultaneously performed by using UDP, the data reception of UDP may not be completed, resulting in a timeout error. Therefore, to perform other communications during the simple CPU communication, the communication using TCP is recommended.

Execution interval setting

- The execution interval of the simple CPU communication function will differ according to the Ethernet line's load rate, operating status of other communications (such as MELSOFT connection, SLMP communication, socket communication function, CC-Link IE Field Network Basic), scan time of the communication destination, and system configuration. Thus, the set execution interval or time required for completing the communications may be increased.
- In the case of the CPU module, the simple CPU communication function operates at regular intervals. Therefore, if the execution interval is set short, access to the SD memory card may be delayed. The following shows the functions which delay access to the SD memory card.

| Function | Operation which delays access to the SD memory card |
|--|--|
| Access to the SD memory card by peripheral devices | Reading / writing to CPU module, writing during RUN, etc. |
| Data logging function | Data collection, Creating a data logging file, read/write/remove the data logging settings, view the data logging status and operate the data logging file |
| Memory dump function | Save/read of memory dump file |
| Extended file register | Execution of the ERREAD/ERWRITE/ERINIT instructions, data batch reading function, data batch writing function, data batch initialization function |
| Event history function | Logging/viewing/clearing of event history file |
| Data backup/restoration function | Backup processing triggered by turning on SM1351, restoration processing triggered by turning on SM1354 |
| File transfer function (FTP server) | General access to CPU module |
| File transfer function (FTP client) | General access to CPU module |
| Web server function | General access to CPU module, displaying user Web page |

To communicate at the set execution interval, it is recommended to reduce the communication frequency. (such as setting the execution interval of the simple CPU communication, communication timeout time, setting a longer abnormality monitoring time and reducing the number of the settings) If checking the execution interval is required, perform actual communications and check it with the "Simple PLC Communication Diagnostics" window.

Duplication of own station port numbers

The simple CPU communication cannot be performed in the following cases.

- The own station port number set in the simple CPU communication and the own station port number of the Ethernet-equipped module set in the external device configuration are the same.
- The own station port number set in the simple CPU communication and the one set in another function such as the socket communication are the same.

Set different own station port numbers. For the setting numbers other than the one in which a communication error occurs, communications are performed after the preparation processing is completed normally.

IP address and module type of the external device

Check the IP address and module type of the communication destination device before setting the parameters. The simple CPU communication cannot be performed when the communication destination does not exist (an incorrect IP address is specified) or when the preparation processing is not completed.

Device in the communication destination

Check the type or range of the device to be read or written in the communication destination.

Especially, when "Write" is set for the communication pattern, since the control data of the communication destination may be overwritten, communications may result in malfunction.

When the communication destination is a MODBUS/TCP connection device

When specifying 00H (broadcast) for communication between Ethernet module and MODBUS RTU/ASCII compatible device through a gateway device, observe the following instructions.

- Do not specify "Read" for the communication pattern. A timeout error will occur because slave devices do not return responses.
- Do not simultaneously set a bit device and a word device for the same setting No. Set one of them for one setting.
- When setting multiple devices, set different values for all own station port numbers.

Set the communication timeout time, communication retry count and error monitor time as shown below.

- Communication retry count = 0
- Communication timeout time = error monitor time

According to the above settings, the second communication will be performed after a lapse of the communication timeout time, and the third and following communications will be performed at intervals of the communication timeout time 2.

IP filter function

If the IP filter function interrupts the communication target, the simple CPU communication cannot be performed. In addition, if any unique security function of the communication target (other company's PLC) is working, disable the security function of the communication target.

Remote password

The simple CPU communication cannot be performed when a remote password is set for the CPU module which performs the simple CPU communication. Disable the remote password setting of the communication destination.

When using the special relay (SM) and special register (SD)

Do not write any data to the special relay (SM) and the special register (SD) that are set on the system side. Doing so may cause a system failure or communication failure.

When using a file register (R)

When using a file register (R) for the own station device, use a device with latch setting. For any setting No. using a device outside the latch range, simple CPU communication cannot be performed due to an error.

Redundant system

If the communication target is a redundant system, on the process CPU (in a redundant mode) side, use a control system IP address configured in a redundant mode and set the control system IP address as the communication target IP address of the simple CPU communication.

Number of connected communication destinations

The Ethernet module can be connected with up to 32 communication destinations. However, one connection is occupied per communication destination (1 group). Therefore, the following number of communication destinations can be connected.

(When the number of settings per group is 1)

- Max. number of connected devices = 32 – number of connections occupied by other functions (SLMP, etc.)

14 IP ADDRESS CHANGE FUNCTION

14.1 Overview of the IP address change function

This function is provided to change the IP address of the Ethernet-equipped module by setting the desired IP address to special registers from a peripheral unit or another unit and turning ON a special relay.

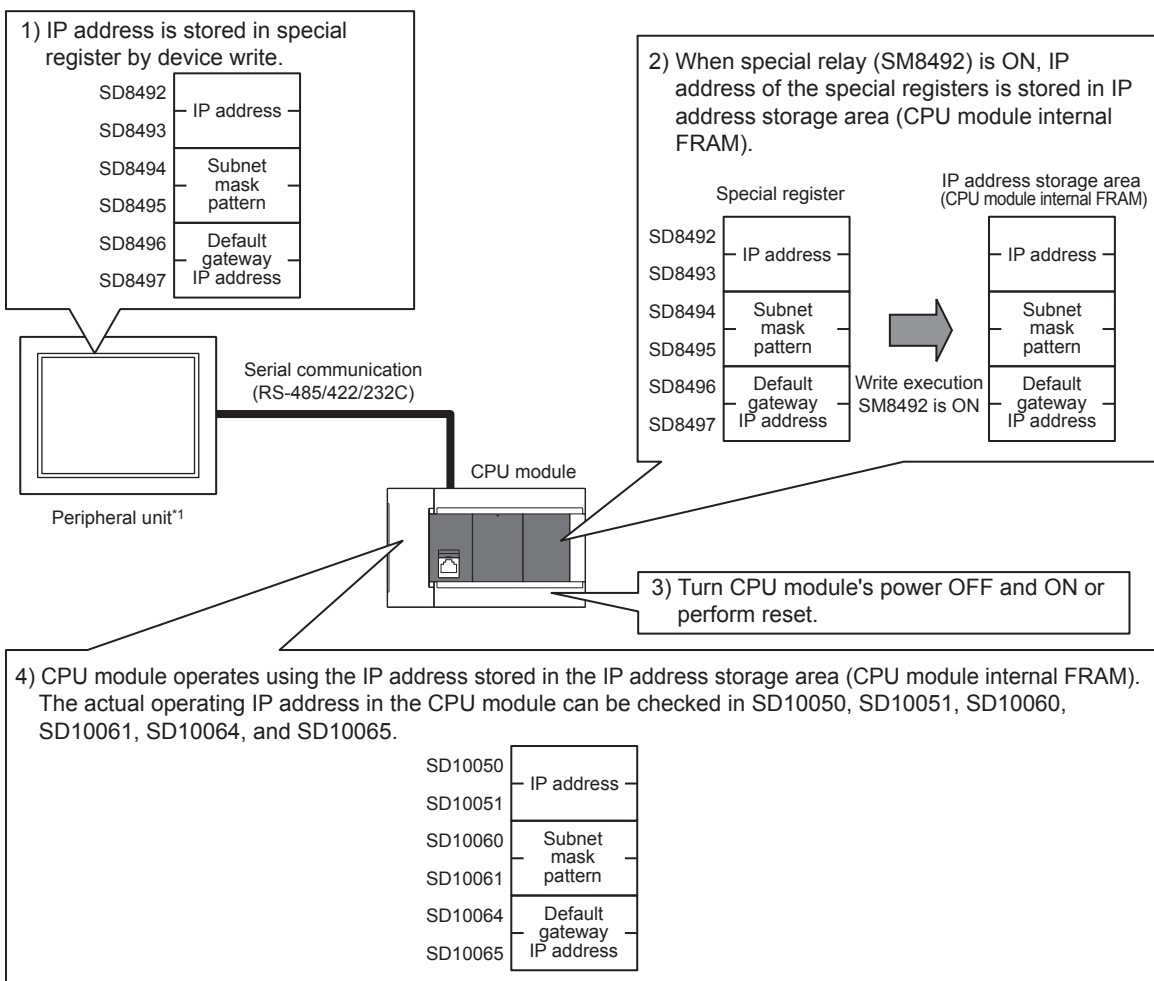
This function changes the IP address of the Ethernet-equipped module even if no settings are made in GX Works3 PLC parameters.

When the IP address change function is used, the IP address stored in the IP address storage area, not the IP address setting of the module parameter Ethernet port in GX Works3, is set to the Ethernet-equipped module.

This function can set three types of data - IP address, subnet mask pattern and default gateway IP address.

Ex.

Case of the CPU module



*1 The IP address change function can be used not only by peripheral units but also by link function, MX Component and MX Sheet by manipulating values of the special devices. For details on link function, refer to the MELSEC iQ-F FX5 User's Manual (Serial Communication). For details on MX Component and MX Sheet, refer to the respective product manual.

- For details on special relays and special register to use IP address change function of CPU module, refer to [Page 270 List of Special Device Applications and Assignments](#).
- For details on the corresponding buffer memory for using the IP address change function of the Ethernet module, refer to [Page 283 List of Buffer Memory Applications and Assignments](#).
- The IP address storage area is different from the storage of the module parameter setting value.
- The IP address storage area is provided in the built-in FRAM of the CPU module and in the built-in flash ROM of the Ethernet module. The IP address is not stored in the SD memory card even if a SD memory card is attached. The setting stored in the IP address storage area is not changed even if the SD memory card is replaced.

14.2 IP address to be set for the Ethernet-equipped module

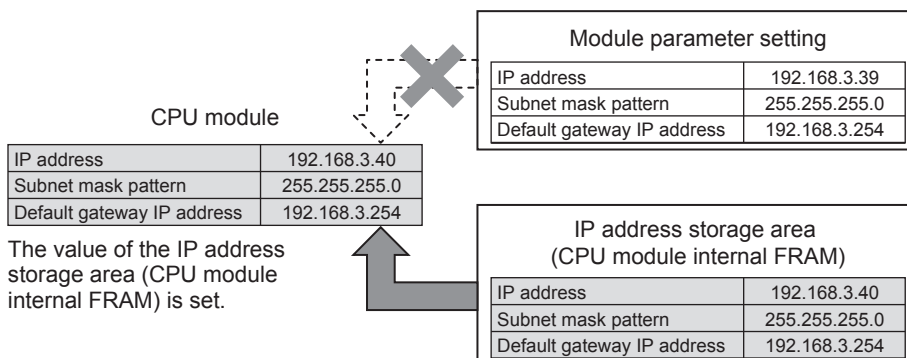
When the IP address change function is used, the IP address stored in the IP address storage area, not the IP address setting of the module parameter in GX Works3, is set to the Ethernet-equipped module.

When the power of the CPU module is turned OFF and ON or reset is performed, the IP address and other data stored in the IP address storage area are reflected on the CPU module, and the IP address change function enable flag (SM8498, Un\G60) turns ON or 0 to 1.

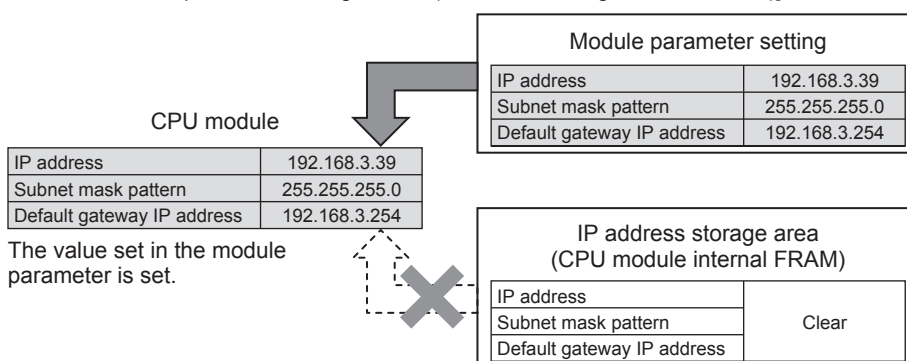
Ex.

Case of the CPU module

[In the case IP address change function is used]



[In the case Module parameter setting is used (IP address storage area is cleared)]

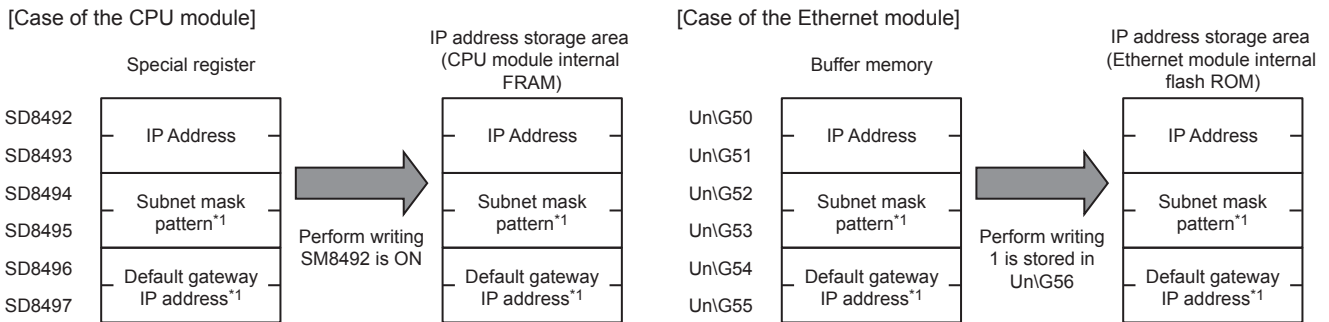


When IP address storage area is cleared (IP address change function enable flag SM8498 is OFF), module parameter setting is enabled.

14.3 Write operation to IP address storage area

Set the IP address and other data to be set to each device, and turn the IP address storage area write request (SM8492, Un\G56) from OFF to ON or 0 to 1 to write the data to the IP address storage area.

Set the IP address, etc. to the each device as hexadecimal values.



*1 It is possible to specify no setting (0.0.0.0) for the subnet mask pattern and default gateway IP address. However, it is necessary to specify no setting (0.0.0.0) for both the subnet mask pattern and the default gateway IP address. If no setting (0.0.0.0) is specified for only one, an error will occur and the IP address will not be changed.

It is possible to write data to the IP address storage area without regard to the CPU module mode (RUN or STOP). To reflect the setting stored in the IP address storage area on the Ethernet-equipped module, it is necessary to turn OFF and ON the power or perform reset of the CPU module.

IP address storage area write procedure

The figure below shows the procedure to write data to the IP address storage area and change the IP address of the Ethernet-equipped module.

Write operation

■Case of the CPU module

1. Store the value to be changed in SD8492 to SD8497 (IP address setting or other).
2. Turn off and on SM8492 (IP address storage area write request).
3. Check the write results with the following special relays and special registers.

| Device No. | Name | At normal completion | At abnormal completion |
|------------|--|----------------------|------------------------|
| SM8492 | IP address storage area write request | On | On |
| SM8493 | IP address storage area write completed | On | On |
| SM8494 | IP address storage area write error | Off | On |
| SD8498 | IP address storage area write error code | — | Stores the error code |

4. When the write is completed normally, power off and on or reset the CPU module.
5. If the IP address stored in the IP address storage area (CPU module internal FRAM) is a valid value, the stored IP address is set as the CPU module's IP address. (If the value is invalid or not set, the value set in the module parameters is set as the IP address of the CPU module.)
6. The IP address or other setting of the CPU module can be checked with the following special register.

| Device No. | Name | Description |
|------------------|----------------------------|---|
| SD10050, SD10051 | IP address | The IP address currently set in the CPU module is stored. |
| SD10060, SD10061 | Subnet mask | The subnet mask currently set in the CPU module is stored. |
| SD10064, SD10065 | Default gateway IP address | The default gateway IP address currently set in the CPU module is stored. |

If the data is not written correctly into the IP address storage area (CPU module internal FRAM), the error code is stored in IP address storage area write error code (SD8498).

| Value of SD8498 | Error details and causes | Action |
|-----------------|---|---|
| 1920H | IP address setting or other (SD8492 to SD8497) value exceeds the setting range. | Correct the IP address setting or other (SD8492 to SD8497) value. |

■Case of the Ethernet module

1. Store the value to be changed in Un\G50 to Un\G55 (IP address setting or other).
2. Store 1 in the IP address storage area write request (Un\G56).
3. Check the write results with the following buffer memories.

| Device No. | Name | At normal completion | At abnormal completion |
|------------|--|----------------------|------------------------|
| Un\G56 | IP address storage area write request | 0 | 0 |
| Un\G57.b0 | IP address storage area write completed | On | On |
| Un\G57.b1 | IP address storage area write error | Off | On |
| Un\G61 | IP address storage area write error code | — | Stores the error code |

4. When the write is completed normally, power off and on or reset the CPU module.
5. If the IP address stored in the IP address storage area (Ethernet module internal flash ROM) is a valid value, the stored IP address is set as the Ethernet module's IP address. (If the value is invalid or not set, the value set in the module parameters is set as the IP address of the Ethernet module.)
6. The IP address or other setting of the Ethernet module can be checked with the following buffer memory.

| Device No. | Name | Description |
|----------------|----------------------------|--|
| Un\G64, Un\G65 | IP address | The IP address currently set in the Ethernet module is stored. |
| Un\G74, Un\G75 | Subnet mask | The subnet mask currently set in the Ethernet module is stored. |
| Un\G76, Un\G77 | Default gateway IP address | The default gateway IP address currently set in the Ethernet module is stored. |

If the data is not written correctly into the IP address storage area (Ethernet module internal flash ROM), the error code is stored in IP address storage area write error code (Un\G61).

| Value of Un\G61 | Error details and causes | Action |
|-----------------|---|---|
| 1920H | IP address setting or other (Un\G50 to Un\G55) value exceeds the setting range. | Correct the IP address setting or other (Un\G50 to Un\G55) value. |

14.4 Clear operation to IP address storage area

When the IP address storage area clear request (SM8495, Un\G58) turns from OFF to ON or 0 to 1, IP address storage area can be cleared. (IP address change function can be disabled.)

IP address storage area clear procedure

IP address storage area clear procedure is described.

Clearing operation

■Case of the CPU module

1. Turn off and on SM8495 (IP address storage area clear request).
2. Check the clear results with the following special relays and special registers.

| Device No. | Name | At normal completion | At abnormal completion |
|------------|--|----------------------|------------------------|
| SM8495 | IP address storage area clear request | On | On |
| SM8496 | IP address storage area clear completed | On | On |
| SM8497 | IP address storage area clear error | Off | On |
| SD8499 | IP address storage area clear error code | — | Stores the error code |

3. When it completed normally, power off and on or reset the CPU module.
4. The IP address or other setting of the CPU module can be checked with the following special register.

| Device No. | Name | Description |
|------------------|----------------------------|---|
| SD10050, SD10051 | IP address | The IP address currently set in the CPU module is stored. |
| SD10060, SD10061 | Subnet mask | The subnet mask currently set in the CPU module is stored. |
| SD10064, SD10065 | Default gateway IP address | The default gateway IP address currently set in the CPU module is stored. |

If the data is not clear correctly into the IP address storage area (CPU module internal FRAM), the error code is stored in IP address storage area clear error code (SD8499).

| Value of SD8499 | Error details and causes | Action |
|-----------------|---|--|
| 1921H | Write request and clear request (SM8492 and SM8495) turned from OFF to ON simultaneously. | Check if write request and clear request (SM8492 and SM8495) turned from OFF to ON simultaneously. |

■Case of the Ethernet module

1. Store 1 in the IP address storage area write request (Un\G58).
2. Check the write results with the following buffer memories.

| Device No. | Name | At normal completion | At abnormal completion |
|------------|--|----------------------|------------------------|
| Un\G58 | IP address storage area clear request | 0 | 0 |
| Un\G59.b0 | IP address storage area clear completed | On | On |
| Un\G59.b1 | IP address storage area clear error | Off | On |
| Un\G62 | IP address storage area clear error code | — | Stores the error code |

3. When it completed normally, power off and on or reset the CPU module.
4. The IP address or other setting of the Ethernet module can be checked with the following buffer memory.

| Device No. | Name | Description |
|----------------|----------------------------|--|
| Un\G64, Un\G65 | IP address | The IP address currently set in the Ethernet module is stored. |
| Un\G74, Un\G75 | Subnet mask | The subnet mask currently set in the Ethernet module is stored. |
| Un\G76, Un\G77 | Default gateway IP address | The default gateway IP address currently set in the Ethernet module is stored. |

If the data is not clear correctly into the IP address storage area (Ethernet module internal flash ROM), the error code is stored in IP address storage area clear error code (Un\G62).

| Value of Un\G62 | Error details and causes | Action |
|-----------------|---|---|
| 1921H | 1 was stored simultaneously in the write request and clear request (Un\G56 and Un\G58). | Check if 1 was not set simultaneously in the write request and clear request (Un\G56 and Un\G58). |

14.5 Precautions

The following section lists the precautions for using the IP address change function.

Power off and reset operation

Do not turn the CPU module power off or execute reset when writing to or clearing the IP address storage area. The values may not be applied to the IP address storage area. Turn off and reset the CPU module power after making sure that the IP address storage area write request (SM8492, Un\G56) and IP address storage area clear request (SM8495, Un\G58) have been changed from ON to OFF or from 1 to 0.

Parameter IP address

For the Ethernet-equipped module IP address, the value in the IP address storage area has precedence over the module parameter value. Whether the IP address change function is enabled can be checked by the IP address change function enable flag (SM8498, Un\G60). When using the IP address specified with the module parameter, clear the IP address storage area.

Write processing and clear processing execution timing

- It may not be possible to execute the write or clear processing to the IP address storage area if an operation that turns off and on, or on and off IP address storage area write request (SM8492, Un\G56), IP address storage area clear request (SM8495, Un\G58) in a short time is executed.
- If IP address storage area write request (SM8492, Un\G56) is turned off and on again while writing to the IP address storage area, the write processing that was executed first will complete normally, and the following write operation will be ignored. (This also applies to the clear operation.)
- If IP address storage area clear request (SM8495, Un\G58) is turned off and on again while writing to the IP address storage area, the clear operation will not be completed. (This also applies if writing is executed during the clear processing.)
- If both IP address storage area write request (SM8492, Un\G56) and IP address storage area clear request (SM8495, Un\G58) are turned off and on, the write operation will take priority, and the clear operation will not be completed.

15 TROUBLESHOOTING

This section contains an explanation of errors that may occur during communication between Ethernet-equipped module and other devices, and troubleshooting for such errors.

The methods for checking the Ethernet-equipped module for errors and the contents of errors are as follows.

In either case, after checking for errors and the contents of the errors, take proper measures to eliminate the error.


Check by display LEDs on the front of the Ethernet-equipped module

You can check whether or not errors currently exist on the Ethernet-equipped module by the status of the display LEDs.

Check using GX Works3

You can check error code corresponding to errors currently occurring, status of the Ethernet-equipped module side, and conduct tests using GX Works3.


■Ethernet diagnostics (Page 233 Ethernet diagnostics)

- Checking error contents by error code ( Page 249 Error codes of the Ethernet communication)


Point

If a line error, etc., occurs when connecting with devices of multiple manufacturers, determine the location of the error by using a line analyzer, etc.

■Module diagnostics (Page 238 Module diagnostics)

- Checking error contents by error code ( Page 245 Error Codes)

■Event history function (Page 240 Event history function)

- Checking error contents by event code ( Page 261 List of Event Code)


15.1 Checking Errors by LEDs

This section contains an explanation of errors that can be checked by LEDs on the front of the Ethernet-equipped module.

Error display check


The following can be checked by LEDs on the front of the Ethernet-equipped module.

<CPU module LED>

| LED name | Check condition | Cause/action |
|----------|---|--|
| PWR | Does not turn on when power of CPU module is turned on | Power source may not be correctly connected to the CPU module. Check the connection status. If there is nothing wrong with connection, the hardware may be faulty. For repair, contact your local Mitsubishi Electric representative. |
| ERR | Turns on when power of CPU module is turned on | In case of built-in Ethernet parameter setting error: • Check/correct built-in Ethernet parameter setting values using GX Works3. In case of CPU module error (hardware error): • For repair, contact your local Mitsubishi Electric representative. |
| | Flashes when power of CPU module is turned on Or flashes temporarily | Check the contents of the error by error code stored when error is detected by the following processing, and eliminate the cause of the error. • Initial process • Open process • SLMP communication process • Other processing (processing wherein error code is stored) For error codes, refer to  Page 245 Error Codes. |


| LED name | Check condition | Cause/action |
|----------|---|---|
| SD/RD | Does not turn on when data is sent/received | <p>If [ERR] is lit:</p> <ul style="list-style-type: none"> Eliminate the cause of [ERR] being lit. <p>If cable connection is faulty:</p> <ul style="list-style-type: none"> Check cable connection. Perform initial processing completion check and check if there is anything wrong with the Ethernet line. <p>In case of own station IP address setting error:</p> <ul style="list-style-type: none"> If there is nothing wrong with cable connection, check the setting values of own station IP address, router setting and subnet mask setting by GX Works3. <p>If there is something wrong with transmission program of other device:</p> <ul style="list-style-type: none"> Check the transmission program of other device. |

<Ethernet module LED>

| LED name | Check condition | Cause/action |
|----------|--|---|
| POWER | Does not turn on when power of CPU module is turned on | <p>The power cable or extension cable may not be correctly connected to the CPU module or Ethernet module. Check the connection status.</p> <p>If there is nothing wrong with connection, the hardware may be faulty. For repair, contact your local Mitsubishi Electric representative.</p> |
| ERROR | Turns on when power of CPU module is turned on | <p>In case of Ethernet parameter setting error:</p> <ul style="list-style-type: none"> Check/correct Ethernet parameter setting values using GX Works3. <p>In case of Ethernet module error (hardware error):</p> <ul style="list-style-type: none"> For repair, contact your local Mitsubishi Electric representative. |
| | Flashes when power of CPU module is turned on Or flashes temporarily | <p>Check the contents of the error by error code stored when error is detected by the following processing, and eliminate the cause of the error.</p> <ul style="list-style-type: none"> Initial process Open process Other processing (processing wherein error code is stored) <p>For error codes, refer to  Page 245 Error Codes.</p> |
| SD/RD | Does not turn on when data is sent/received | <p>If [ERROR] LED is lit:</p> <ul style="list-style-type: none"> Eliminate the cause of [ERROR] being lit. <p>If cable connection is faulty:</p> <ul style="list-style-type: none"> Check cable connection. Perform initial processing completion check and check if there is anything wrong with the Ethernet line. <p>In case of own station IP address setting error:</p> <ul style="list-style-type: none"> If there is nothing wrong with cable connection, check the setting values of own station IP address, router setting and subnet mask setting by GX Works3. <p>If there is something wrong with transmission program of other device:</p> <ul style="list-style-type: none"> Check the transmission program of other device. |

Error information read/clear method

You can read and clear error information by Ethernet diagnostics of GX Works3.

For details concerning Ethernet diagnostics of GX Works3, refer to  Page 233 Checking Errors by GX Works3.

15.2 Checking Errors by GX Works3

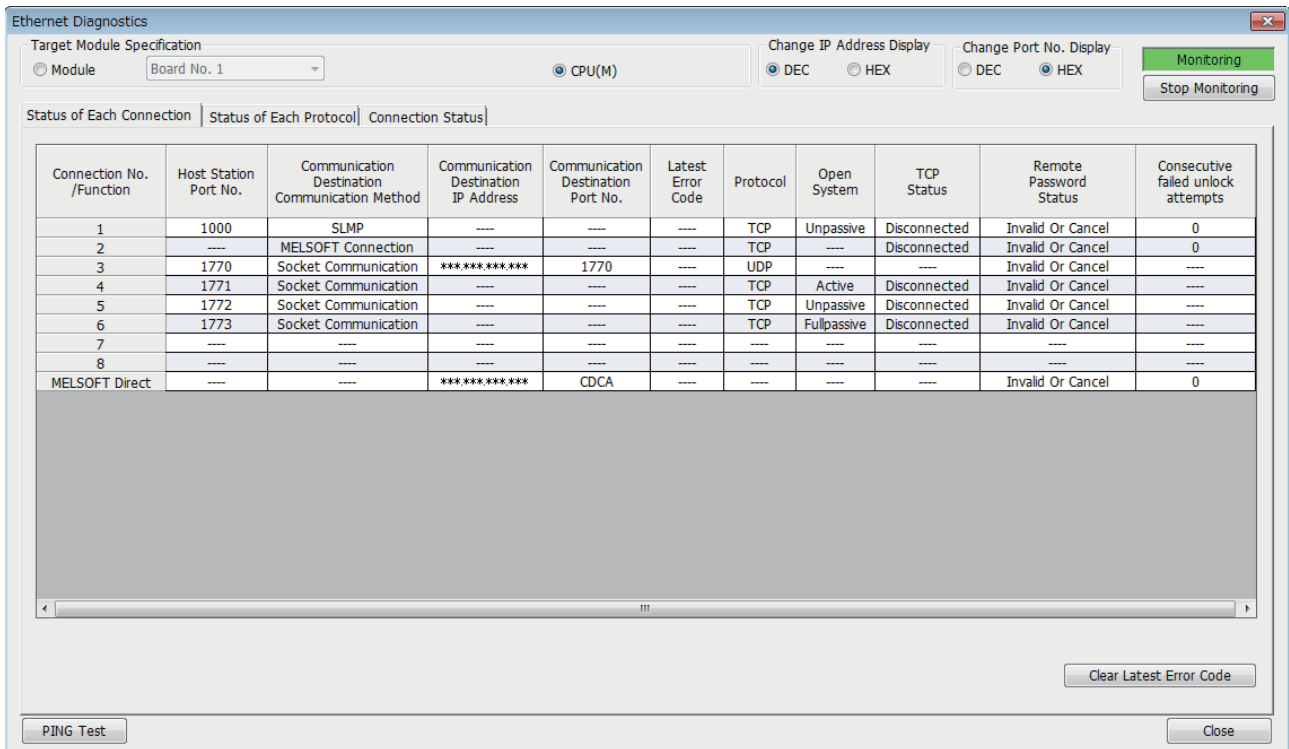
You can check Ethernet status, parameter settings, communication status, etc., with the Ethernet diagnostics function of GX Works3.

Ethernet diagnostics

For details concerning Ethernet diagnostics of GX Works3, refer to GX Works3 Operating Manual.

- Ethernet diagnostics screen

[Diagnostics] ⇒ [Ethernet Diagnostics]



- Ethernet diagnostics item

| Item | Description |
|---------------------------|--|
| Status of Each Connection | Displays information concerning status of each connection. |
| Status of Each Protocol | The total of the send/receive of the packet etc. for each protocol is displayed. |
| Connection Status | Monitors connection status. |

Status of Each Connection

The status of each connection of the Ethernet-equipped module selected.

| Connection No. /Function | Host Station Port No. | Communication Destination Communication Method | Communication Destination IP Address | Communication Destination Port No. | Latest Error Code | Protocol | Open System | TCP Status | Remote Password Status | Consecutive failed unlock attempts |
|--------------------------|-----------------------|--|--------------------------------------|------------------------------------|-------------------|----------|-------------|--------------|------------------------|------------------------------------|
| 1 | 1000 | SLMP | ---- | ---- | ---- | TCP | Unpassive | Disconnected | Invalid Or Cancel | 0 |
| 2 | ---- | MELSOFT Connection | ---- | ---- | ---- | TCP | ---- | Disconnected | Invalid Or Cancel | 0 |
| 3 | 1770 | Socket Communication | ***** | 1770 | ---- | UDP | ---- | ---- | Invalid Or Cancel | ---- |
| 4 | 1771 | Socket Communication | ---- | ---- | ---- | TCP | Active | Disconnected | Invalid Or Cancel | ---- |
| 5 | 1772 | Socket Communication | ---- | ---- | ---- | TCP | Unpassive | Disconnected | Invalid Or Cancel | ---- |
| 6 | 1773 | Socket Communication | ---- | ---- | ---- | TCP | Fullpassive | Disconnected | Invalid Or Cancel | ---- |
| 7 | ---- | ---- | ---- | ---- | ---- | ---- | ---- | ---- | ---- | ---- |
| 8 | ---- | ---- | ---- | ---- | ---- | ---- | ---- | ---- | ---- | ---- |
| MELSOFT Direct | ---- | ---- | ***** | CDCA | ---- | ---- | ---- | ---- | Invalid Or Cancel | 0 |

The following table lists the displayed items in "Status of Each Connection" tab.

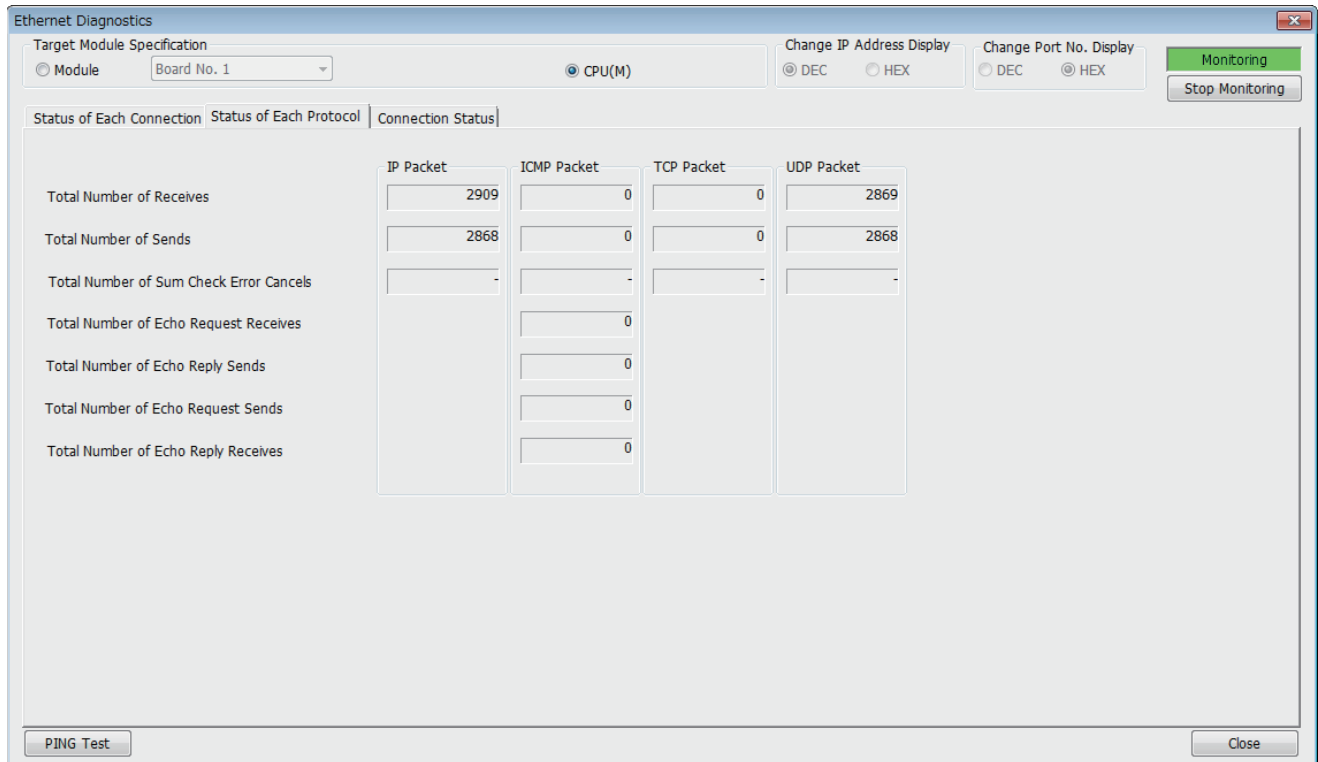
| Item | Description |
|--|--|
| Connection No./Function | Displays the connection number and functions (MELSOFT direct connection). |
| Host Station Port No. | Displays the own station port number used. |
| Communication Destination Communication Method | Displays the communication method. |
| Communication Destination IP Address | Displays the IP address of the sensor/device to be connected, which is set in the parameter settings. |
| Communication Destination Port No. | Displays the port number of the sensor/device to be connected, which is set in the parameter settings. |
| Latest Error Code | Displays the error code that indicates the definition of latest error occurred. |
| Protocol | Displays the protocol (TCP/IP or UDP/IP) |
| Open System | Displays the open method (Active, Unpassive, or Fullpassive) when the protocol of the connection is TCP/IP. |
| TCP Status | Displays the status (open status) of connection with the sensor/device when the protocol of the connection status is TCP/IP. |
| Remote Password Status | Displays the remote password setting status.*1 |
| Continuous Unlock Lost Counts | Displays the total number of continuous failure of remote password unlock. |

*1 Only CPU module is supported.

Click the [Clear Latest Error Code] button to clear all the errors displayed in "Latest Error Code" of each connection.

Status of Each Protocol

The total number of packets sent/received by each protocol of the selected Ethernet-equipped module can be checked.

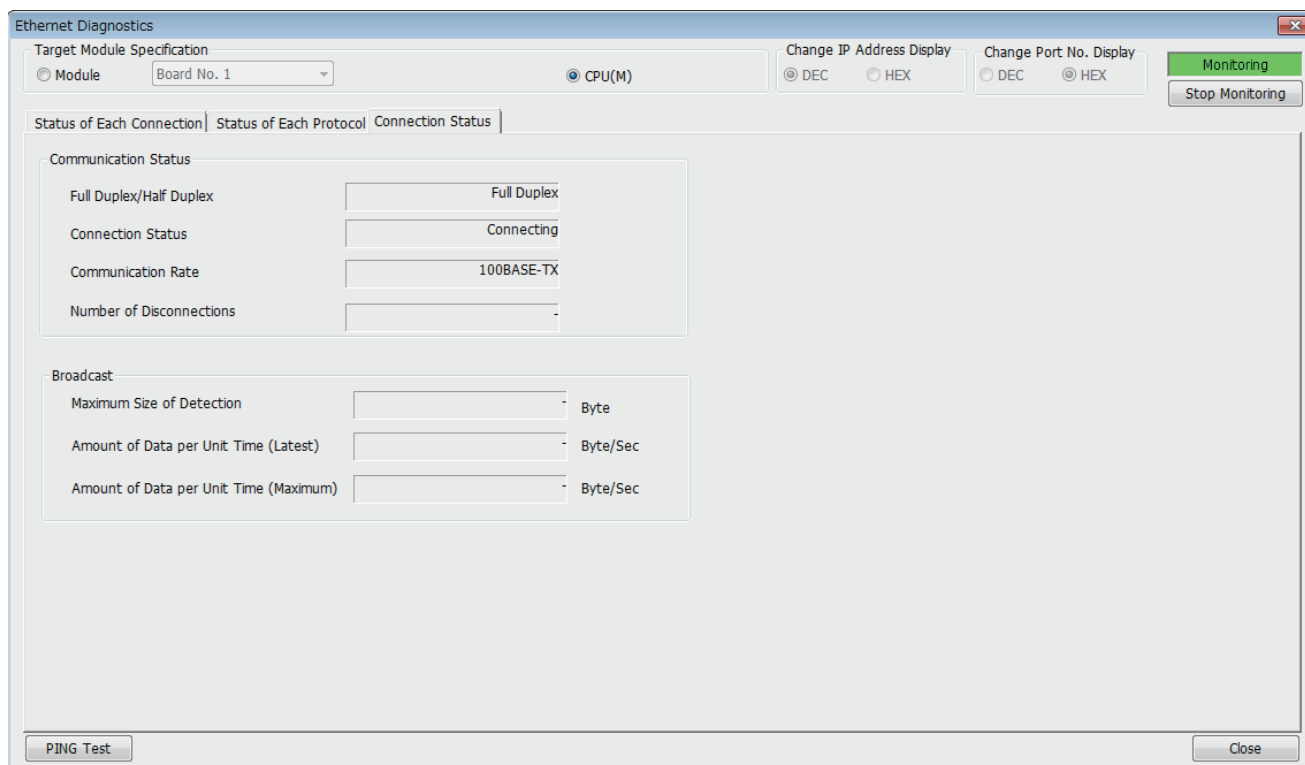


The following table lists the displayed items in "Status of Each Protocol" tab.

| Item | Description | Display range |
|---|--|-----------------|
| Total Number of Receives | Displays the total number of received packets. | 0 to 4294967295 |
| Total Number of Sends | Displays the total number of sent packets. | 0 to 4294967295 |
| Total Number of Sum Check Error Cancels | Not supported. | — |
| Total Number of Echo Request Receives | Displays the total number of received ICMP echo request packets. | 0 to 4294967295 |
| Total Number of Echo Reply Sends | Displays the total number of sent ICMP echo reply packets. | 0 to 4294967295 |
| Total Number of Echo Request Sends | Displays the total number of sent ICMP echo request packets. | 0 to 4294967295 |
| Total Number of Echo Reply Receives | Displays the total number of received ICMP echo reply packets. | 0 to 4294967295 |

Connection Status

The communication status of the Ethernet-equipped module.



The following table lists the displayed items in "Connection Status" tab.

| Item | | Description | Display range |
|----------------------|--|--|---------------|
| Communication Status | Full Duplex/Half Duplex | Displays the communication mode of line. | — |
| | Connection Status | Displays the cable connection status. | — |
| | Communication Rate | Displays the communication speed. | — |
| | Number of Disconnections | Not supported. | — |
| Broadcast | Maximum Size of Detection | Not supported. | — |
| | Amount of Data per Unit Time (Latest) | Not supported. | — |
| | Amount of Data per Unit Time (Maximum) | Not supported. | — |

PING Test

The PING test checks existence of an Ethernet device on the same Ethernet network. Only CPU module is supported. This test is performed on the network of stations connected to the GX Works3 by sending packets for check. If a response returns, the communication can be performed.

☞ "Ethernet Diagnostics" window ⇨ [PING Test] button

The screenshot shows the "PING Test" dialog box. It is titled "PING Test" and has a close button (X) in the top right corner. The dialog is divided into three main sections: "Input Item", "Setting Options", and "Result".

Input Item:

- Address Specification:** There are two radio buttons: "IP Address" (selected) and "IP Address/Host Name".
- IP Address Input Form:** There are two radio buttons: "DEC" (selected) and "HEX".

Setting Options:

- Display the Host Name:** A checkbox that is currently unchecked.
- Default:** A button to reset settings to default.
- Specify the Data Size:** A text box containing "32" followed by "Byte".
- Specify the Communication Time Check:** A text box containing "1" followed by "Seconds".
- Specify the Number of Sends:** A dropdown menu showing "Specify the Count" and a text box containing "4" followed by "Times".

Buttons: "Execute" and "Cancel" buttons are located below the settings.

Result: A large empty text area for displaying the test results.

Status Bar: At the bottom, there is a status bar with the text "Number of Successes/Transmissions = /" and a "Close" button.

■Procedure

Set the required items in "Input Item" and click the [Execute] button to execute the PING test. The test results are displayed in the "Result" box.

■Action for abnormal end

If the test fails, check the following and perform the test again.

- Connection to the Ethernet network
- Parameter settings written in the CPU module
- Operating status of the CPU module (whether or not an error has occurred)
- IP addresses set in GX Works3 and the PING test target station
- Whether the external device has been reset after the CPU module was replaced


Module diagnostics

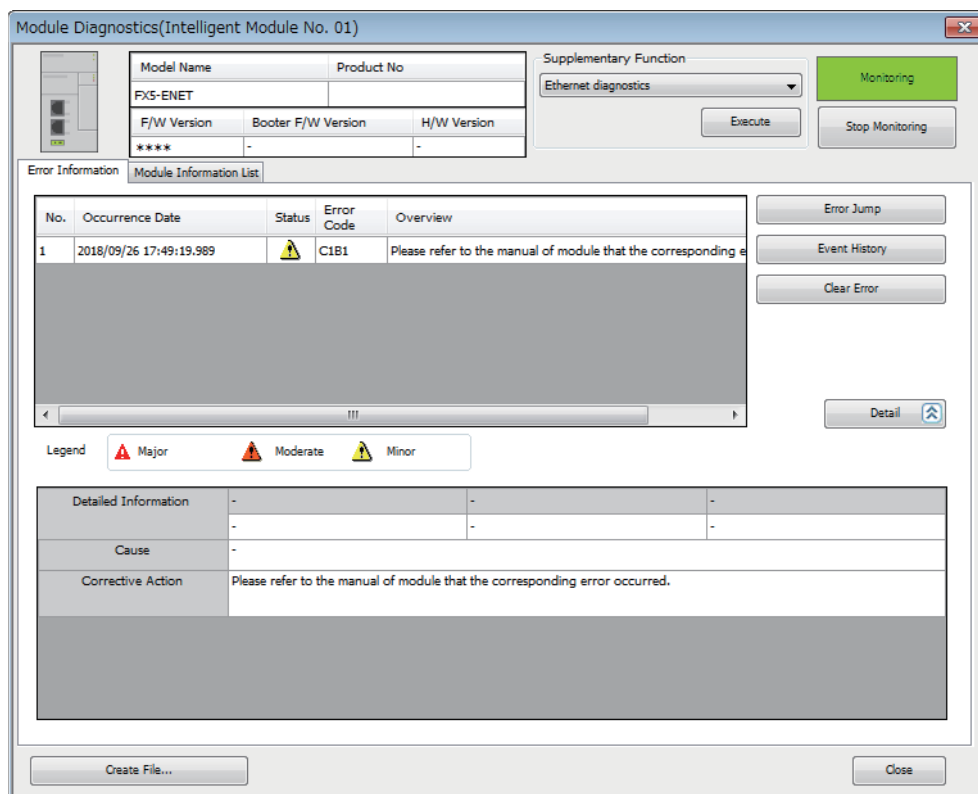
The following functions can be used in the "Module Diagnostics" window for the Ethernet module.

| Function | Application |
|-------------------------|--|
| Error Information | Displays the details of the errors currently occurring. Click the [Event History] button to check the history of errors that have occurred on the Ethernet module, errors detected for each module. |
| Module Information List | Displays various status information of the Ethernet module. |

Error Information

Check the details of the error currently occurring and action to eliminate the error.

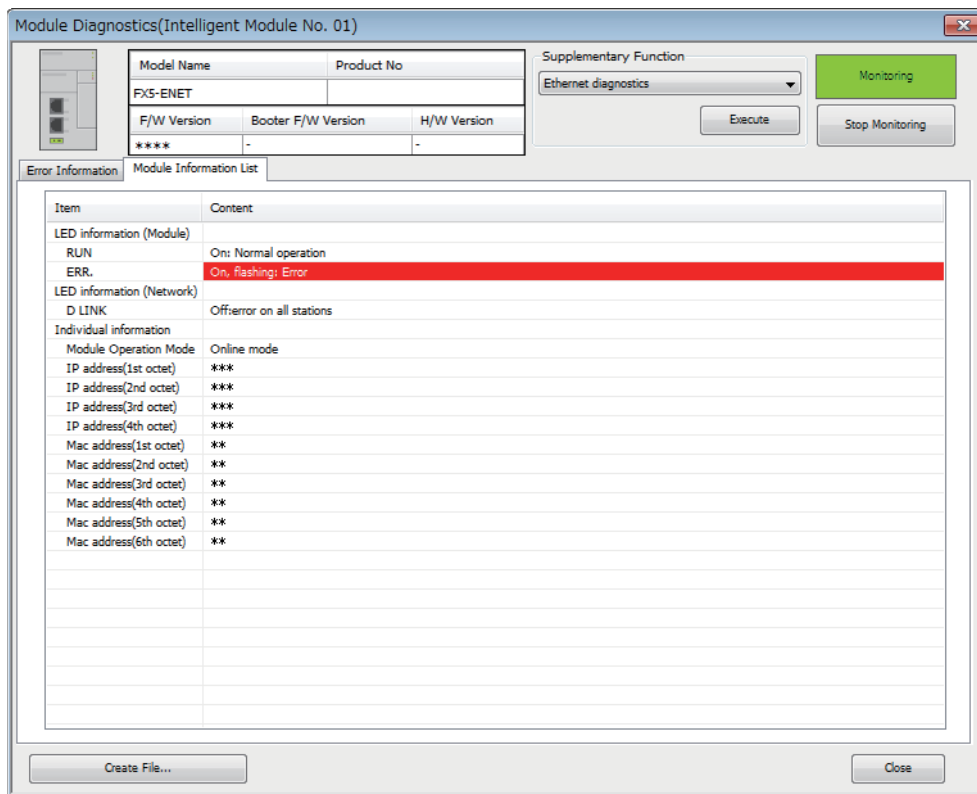
 [Diagnostics]⇒[System Monitor]⇒Double-click "FX5-ENET" or "FX5-ENET/IP"



| Item | Description |
|----------------------|--|
| Status | Major: An error such as hardware failure or memory failure. The module stops operating. |
| | Moderate: An error, such as parameter error, which affects module operation. The module stops operating. |
| | Minor: An error such as communication failure. The module continues operating. |
| Detailed Information | Displays detailed information about each error (maximum of three pieces). |
| Cause | Displays the detailed error causes. |
| Corrective Action | Action Displays the actions to eliminate the error causes. |

Module Information List

Switch to the [Module Information List] tab to check various status information of the Ethernet module.



| Item | Description | |
|---|--|--|
| LED information (Module) | Displays the status of the RUN LED and ERROR LED of the Ethernet module. | |
| LED information (Communication)* ¹ | Displays the status of the D LINK of the Ethernet module. | |
| Setting information* ¹ | IP Address (1st Octet) | Displays the IP address of the Ethernet module. |
| | IP Address (2nd Octet) | |
| | IP Address (3rd Octet) | |
| | IP Address (4th Octet) | |
| | MAC Address (1st Octet) | Displays the MAC address of the Ethernet module. |
| | MAC Address (2nd Octet) | |
| | MAC Address (3rd Octet) | |
| | MAC Address (4th Octet) | |
| | MAC Address (5th Octet) | |
| | MAC Address (6th Octet) | |

*1 An undefined value is stored during hardware test. For hardware test, refer to the following manuals.

📖 MELSEC iQ-F FX5-ENET User's Manual


📖 MELSEC iQ-F FX5-ENET/IP User's Manual

Event history function

This function collects errors from Ethernet-equipped module, and keeps them in the SD memory card, and data memory or battery backed built-in RAM of the CPU module.

The event information collected by the CPU module can be displayed on GX Works3 to check the occurrence history in chronological order.

| Event type | Classification | Description |
|------------|----------------|--|
| System | Error | An error detected by the self diagnostics in Ethernet-equipped module. |

For event code of the Ethernet module, refer to  Page 261 List of Event Code.


Setting procedure

The event history function can be set from the event history setting window of GX Works3. For the setting procedure, refer to the following.

 MELSEC iQ-F FX5 User's Manual (Application)

Displaying event history

Access the menu of GX Works3. For details on the operating procedure and how to view the contents, refer to the following.

 GX Works3 Operating Manual

15.3 Troubleshooting by Symptom

The following for troubleshooting by symptom.

IP address duplication detection

When turning the power from off to on, resetting, or connecting to the network, the Ethernet-equipped module check whether there is any device with duplicated IP address on the same network. When the IP address duplication is detected, an error occurs and appears in the event history. Also, with the following devices, it is possible to check whether there is an IP address duplication, and the MAC address of the device with duplicated IP address can be checked.

| Device No. | | Name |
|--------------------|--------------------|--|
| CPU module | Ethernet module | |
| SD10251 | Un\G201 | Same IP address state storage area |
| SD10252 to SD10254 | Un\G202 to Un\G204 | MAC address of the already connected station |
| SD10255 to SD10257 | Un\G205 to Un\G207 | MAC Address of the station connected later |

After the IP address duplication detection, the contents of the IP address duplication (error information, device, event history) are held until the power is turned off to on, reset, or the error is cleared.

Restriction

If the firmware version of the FX5U/FX5UC CPU module is lower than "1.061 ", when the IP address duplication is detected, the CPU module will stop the processing and the network will be down. Also, the above special devices are not supported.

15

Restoration method of IP address duplication error

The restoration method of IP address duplication error (Error code: 2160H) is shown below.

1. Perform module diagnostics, and check that IP address duplication error does not occur.
2. When the module diagnostics cannot be performed by an Ethernet communication failure, perform direct connection with the other network, and check that IP address duplication error does not occur in the module diagnostics or the event history.
3. When the IP address duplication error occurred, check the device and the device with duplicated IP address.
4. To return to the network, change the IP address of the Ethernet-equipped module to an IP address which is not duplicated in the network, turn the power off to on, or reset, and connect the CPU module to the original network.

When using the file transfer function

Access using the file transfer function (FTP server) cannot be performed

When the CPU module cannot be accessed from the external device (FTP client) using the file transfer function (FTP server), check the following items and take action.

| Check item | Action |
|--|---|
| Is the file transfer function (FTP server) for the CPU module enabled? | Set "FTP Server" under "FTP Server Settings" of "Application Settings" to "Use". (Page 137 Data communication procedures) |
| Are the login name and password correct? | Check the login name and password set in the parameters, and log in again. (Page 137 Data communication procedures) |
| Is the IP address entered at login correct? | Check the IP address of the CPU module set in the parameters, and log in again. (Page 31 Setting module parameters) |
| Is executed command used properly? | Check the instructions of the command and use the command properly. (Page 141 FTP command) |

If the troubleshooting is not successful even though the above measures were taken, troubleshoot individual CPU modules and check their hardware for any error.

Access using the file transfer function (FTP client) cannot be performed

When the CPU module cannot access to the external device (FTP server) during execution of the file transfer function (FTP client), check the following items and take action.

| Check item | Action |
|---|---|
| Is not a value other than 0000H stored in (s1) + 1 (Completion status) of the SP.FTPPUT instruction or SP.FTPGET instruction? | Take action for the error code stored in (s1)+1 (Completion status), and execute the instruction again. (Page 248 Error codes of completion status) |

When the simple CPU communication function is used

Communications cannot be performed with the communication destination.

When communications with the external device cannot be performed using the simple CPU communication function, check the following items and take action.

| Check item | Action |
|---|--|
| Does any error occur in the simple CPU communication diagnostics or simple CPU communication error code in the special device? | Eliminate the error cause. When communications can be performed by increasing the communication retry count and communication time-out period, review the communication retry count and communication time-out period, or check the line status of Ethernet communications with the external device, such as a cable, hub, and router, to check if the line is busy. |
| Is the communication status of the simple CPU communication diagnostics or the simple CPU communication status of the special device "preparing"? | Check the cable between the CPU module and hub and if the hub is operating normally. |
| Have the parameters been written? | <ul style="list-style-type: none"> • Check the error code shown in the simple CPU communication status, and take action. • Rewrite the parameters, and perform communication. |
| Have the parameters been set correctly? | |

If communications cannot be performed despite the check for the above items, the hardware failure may have been occurred. Please consult your local Mitsubishi representative.

Communications with unstable communication destination.

When communications with the communication destination using the simple CPU communication is unstable, check the following items and take action.

| Check item | Action |
|--|---|
| Does any error occur in the simple CPU communication diagnostics or simple CPU communication error code in the special device? | Eliminate the error cause. |
| Is the retry of communications performed? | Check the line status of Ethernet communications with the external device, such as a cable, hub, and router, to check if the line is busy. |
| Are not parameters rewritten during simple CPU communication? | <ul style="list-style-type: none"> • Check the error code shown in the simple CPU communication status, and take action. • Turn off and on or reset the CPU module power. |

If communications cannot be performed despite the check for the above items, the hardware failure may have been occurred. Please consult your local Mitsubishi representative.

Communications cannot be performed at the set execution interval.

When communications cannot be performed at the set execution interval in the simple CPU communication function, check the following items and take action.

| Check item | Action |
|--|--|
| Is not the scanning time long for the CPU module where the simple CPU communication function is set? | <ul style="list-style-type: none"> • Reduce the scanning time. • Increase the setting of the execution interval. |
| Is another Ethernet function executed simultaneously in the CPU module where the simple CPU communication function is set? | Review the Ethernet function executed simultaneously in the CPU module where the simple CPU communication function is set. |
| Is the response from the SLMP-compatible device delayed? (When the SLMP-compatible device (QnA-compatible 3E frame) is used) | <ul style="list-style-type: none"> • Check the function execution status of the communication destination (SLMP-compatible device). • Review the execution interval. |

The communication destination is kept in the monitoring status

If a specific destination is kept in the monitoring status and does not enter the communicating status during communication through the simple CPU communication function, check the following items and take action.

| Check item | Action |
|---|--|
| Is the device range of the destination device appropriate? | <ul style="list-style-type: none">• Check the error code shown in the simple CPU communication status, and take action.• Check the system configuration and parameter settings. |
| Is the specified destination device applicable? | |
| Is the destination device connected? | |
| Is not the power off to the destination device? | |
| Has not a remote password been set? | |
| Has not the setting on the destination device been made? | |
| Is the address set for the destination device correct? | |
| Is the device type set for the destination device appropriate to the actually connected device? | |
| Has not the IP filter been set? | |

A time-out occurs just after the start of communication

If a communication time-out error occurs just after the start of simple CPU communication, check the following items and take action.

| Check item | Action |
|--|--|
| <ul style="list-style-type: none">• Has the Ethernet communication been established?• Does the network environment allow to establish the Ethernet communication? | Set the latency time, and start the simple CPU communication after waiting for the specified time. |

Access to the CPU module cannot be performed

If access to the CPU module from the target device cannot be performed during the simple CPU communication, check the following items and take action.

| Check item | Action |
|--|--|
| Is not the communication time check of the target device short? | Set the communication time check of the target device more than 30 seconds. |
| Are the execution interval of the simple CPU communication, communication timeout time, and monitoring time short? | Set the execution interval of CPU communication and communication timeout time with sufficient execution interval. Set the monitoring time more than 30 seconds. |

15.4 Error Codes

For the error codes (stored in SD0/SD8067) common among CPU modules, refer to the MELSEC iQ-F FX5 User's Manual (Application).

Error codes of the IP address change function

The description and corrective action for error codes generated by the IP address change function are explained.

CPU module

Error codes are stored in SD8498 (IP address storage area write error code) or SD8499 (IP address storage area clear error code).

| Error code (Hexadecimal) | Error details and causes | Action |
|--------------------------|---|--|
| 1920H | IP address setting or other (SD8492 to SD8497) value exceeds the setting range. | Correct the IP address setting or other (SD8492 to SD8497) value. |
| 1921H | Write request and clear request (SM8492 and SM8495) turned from OFF to ON simultaneously. | Check if write request and clear request (SM8492 and SM8495) turned from OFF to ON simultaneously. |

Ethernet module

Each error can be checked with the following.

- 1920H: Un\G61 (IP address storage area write error code)
- 1921H: Un\G62 (IP address storage area clear error code)
- 1810H, 1DC4H: Un\G29 (Latest error code), [Error information] tab in the “Module Diagnostics” (Page 238 Error Information)

| Error code (Hexadecimal) | Error details and causes | Action |
|--------------------------|---|---|
| 1810H | IP address change fails. | Execute the IP address change function again. |
| 1920H | IP address setting or other (Un\G50 to Un\G55) value exceeds the setting range. | Correct the IP address setting or other (Un\G50 to Un\G55) value. |
| 1921H | Write request and clear request (Un\G56 and Un\G58) turned from OFF to ON simultaneously. | Check if write request and clear request (Un\G56 and Un\G58) turned from OFF to ON simultaneously. |
| 1DC4H | IP address change fails. | <ul style="list-style-type: none"> • Execute the IP address change function again. • If the error occurs again even after the above action is taken, the possible cause is a hardware failure of the module on which the error occurred. Please consult your local Mitsubishi representative. |

Error codes of the simple CPU communication function

The description and corrective action for error codes generated by the simple CPU communication function are explained.

Error codes for those detected by the own station

■CPU module

Error codes for those detected by the own station are stored in SD10412 to SD10443 (Simple PLC Communication error code).

| Error code (Hexadecimal) | Error details and causes | Action |
|--------------------------|--|---|
| 4006H | Initial communication has failed. | When using Ethernet communication, shift the communication start timing. |
| 4030H | The specified device name cannot be handled. | Check the specified device name. |
| 4031H | The specified device No. is outside the range. | Check the specified device No. |
| C0B2H | There is no sufficient space in the receive buffer or the send buffer of the relay station or external station for the MELSOFT connection, or SLMP. (Send · receive buffer full error) | <ul style="list-style-type: none">• Increase the request interval (execution interval) and execute the operation.• Do not access through one station using the MELSOFT connection, or SLMP.• Wait for a response to the previous request before sending the next request. |
| C201H | Locked status of the remote password of the port which is used for communication. | Unlock the remote password before data communication. (📖 Page 196 Remote Password) |
| CFB0H | Sending data failed due to a resend timeout during the simple CPU communication. | <ul style="list-style-type: none">• Check the operation of the communication destination.• Check the line status with the communication destination such as a cable, hub, and router.• Check and correct the latency time.• Correct the IP address and Ethernet address of the communication destination.• Check if the communication destination has the ARP function, and communicate with a communication destination that has the ARP function. |
| CFB2H | The same specified own station port number is already used. | Correct the specified port number so that it is not duplicate. |
| CFB4H | An abnormal response was received from the communication destination while the simple CPU communication. | Check the abnormal response code in the special registers. |



If the communication target detects any error in other company's PLC, MODBUS/TCP-compatible device or MELSEC-FX3 (Ethernet Block/Adapter), CFB4H will be stored.

■ Ethernet module

| Error code (Hexadecimal) | Error details and causes | Action |
|--------------------------|---|---|
| CFB0H | Transmission has failed due to retransmission timeout. | <ul style="list-style-type: none"> • Check the operation of the external device. • Check the conditions of the cables, hubs and routers on the lines to the external devices. • Reconsider the communication start wait time. • Reconsider and correct the IP address and Ethernet address of the external device. • Confirm that the external devices have the ARP function, and communicate with an external device that has the ARP function. • When the communication destination is a MODBUS/TCP connection device and the communication pattern is Write, do not specify Input or Input Register for the transfer destination device. |
| CFB1H | The external device cannot be connected or has been disconnected. | <ul style="list-style-type: none"> • Check whether the connection cable is not disconnected. • Check the conditions of the cables, hubs and routers on the lines to the external devices. |
| CFB2H | The specified own station port No. is duplicated. | Reconsider the port No., and prevent duplication. |
| CFB3H | A request to the CPU module has failed. | <ul style="list-style-type: none"> • Reconsider the monitor time of the CPU response monitor timer. • Reconsider the device/label access service processing setting. |
| CFB4H | An abnormal response was received from the external device. | Check the abnormal response code in the buffer memory. |
| CFB5H | The frame received from the external device is abnormal. | <ul style="list-style-type: none"> • Check the operation of the external device. • Check the conditions of the cables, hubs and routers on the lines to the external devices. |
| CFBFH | Simple CPU communication cannot be performed. | <ul style="list-style-type: none"> • Take measures against noise. • If the same error is displayed after retry, Please consult your local Mitsubishi representative. |

Point

If the communication destination detects an error in an MODBUS/TCP connection device, CFB4H will be stored.

Error codes for those detected by the communication target

Error codes for those detected by the communication target are stored in SD10476 to SD10507 (Simple CPU communication error response code). For error codes, refer to individual devices' manuals.

For Simple CPU communication error response code, their error codes are stored only when CFB4H is stored in the Simple PLC Communication error code (SD10412 to SD10443). (Only when any error occurred (with other company's PLC, MODBUS/TCP-compatible devices or MELSEC FX3 (Ethernet Block/Adapter) being the communication target)) CFB4H is stored as the Simple PLC Communication error code in the following cases.

① When an abnormal response is returned

② When an abnormal response message is received (for example, when communicating in binary format, receiving a response message in ASCII format)

In case of ①, an error code is stored in Simple CPU communication error response code. In the case of ②, a value that is not an error code is stored in the Simple CPU communication error response code.

Error codes of completion status

Described here are descriptions and actions for error codes of completion status for those occurred in the file transfer function (the FTP client). Any of the error codes of completion status will be stored in control data (s1)+1 of the file transfer function instruction (the SP.FTPPUT instruction, SP.FTPGET instruction).

| Error code (Hexadecimal) | Error details and causes | Action |
|--------------------------|--|---|
| 41C5H | The specified file does not exist. | Execute again after checking the file. |
| 41C7H | The specified file or drive does not exist. | Re-execute after checking the file or drive (memory). |
| 41CCH | The specified file does not exist. Or, the specified subdirectory does not exist. | Execute again after checking the name of the file and subdirectory. |
| 41CDH | An access to the file is prohibited in the system. | Execute again after checking the file and subdirectory. |
| 41CEH | The file cannot be written because the specified file is read-only. | Execute again after checking the attribute of the specified file. |
| 41CFH | The capacity of the specified drive (memory) is exceeded. | Re-execute after checking the drive (memory) capacity. |
| 41D8H | The specified file is being accessed. | Execute again after a while. |
| 41DFH | <ul style="list-style-type: none"> The FTPPUT instruction (when "Delete" had been specified for the transferred file deletion setting) was executed in the state where the write protect switch of the SD memory card was ON. (The file is transferred, but the file in the SD memory card cannot be deleted.) The FTPGET instruction was executed in the state where the write protect switch of the SD memory card was ON, and file transfer was failed. | <ul style="list-style-type: none"> Turn OFF the write protect switch of the SD memory card. Specify "Not delete" for the transferred file deletion setting. |
| 41E4H | Access to the SD memory card has failed. | <ul style="list-style-type: none"> Execute the operation again after checking that the SD memory card has been inserted. Execute the operation again after replacing the SD memory card. Back up data, and then initialize the memory. |
| 41EBH | The specified file name is incorrect. | <ul style="list-style-type: none"> Re-execute after checking the file name. Reconsider whether the total length of the transfer destination folder path and the temporary file name does not exceed 255 characters when the FTPGET instruction is executed. |
| 41FEH | <ul style="list-style-type: none"> The SD memory card has not been inserted. The SD memory card is disabled. The SD memory card is disabled by SM606 (SD memory card forced disable instruction). | <ul style="list-style-type: none"> Insert the SD memory card or insert it again. Slide down the switch for stopping SD memory card usage. Remove the SD memory card, and insert it again. Cancel the SD memory card forced disable instruction. |
| 4C40H | <ul style="list-style-type: none"> When files are specified by using wild card characters for the file transfer function instruction, the number of files matched exceeds the upper limit of the transferable number of files. When files are specified by using wild card characters for the file transfer function instruction, no files are matched. | <ul style="list-style-type: none"> Check and correct the wild card specification. Check if the specified folder path exists. |
| 4C43H | The number of processing completed files for sending FTP client file is mismatched with the total number of processing files. | Execute the function again. |
| 4C44H | The file transfer function (the FTP client) is performed while the CPU module's backup/restore function is being performed. | Perform it again after having completed the CPU module's backup/restore function. |
| C035H | The existence of the external device could not be confirmed within the response monitoring timer value. | <ul style="list-style-type: none"> Check the movement of the other party equipment. Check whether the connecting cable has come off. |
| C616H | Control port connection to the FTP server failed. | <ul style="list-style-type: none"> Review the IP address setting in the built-in Ethernet port setting. Review the FTP server settings. Check the connection status with the FTP server. Disconnect the user session on the FTP server. Re-execute after some time since there is a possibility that communication preparation is in progress. |
| C618H | Login to FTP server failed. | <ul style="list-style-type: none"> Review the FTP server settings (login user name, login password). Check the settings (login user name, login password) of the FTP server software*1. Check the communication history of the FTP server software*1. |

| Error code (Hexadecimal) | Error details and causes | Action |
|--------------------------|--|---|
| C619H | FTP command execution to the FTP server failed. | <ul style="list-style-type: none"> Review the FTP server settings (folder path, connection method). Check whether the FTP server or the specified file has access authority (write authority, read authority). Check whether the specified folder path exists on the FTP server. Check whether the specified file exists on the FTP server. Review the settings of the FTP server software^{*1}. Check the communication history of the FTP server software^{*1}. Check if the file is being accessed on the FTP server. |
| C620H | Connection of the data transfer port for the FTP server failed. | <ul style="list-style-type: none"> Check the connection status with the FTP server. Review the FTP server settings (connection method). If a firewall or proxy server exists in the connection path, check the settings with the network administrator. |
| C621H | Disconnection of the data transfer port for the FTP server failed. | <ul style="list-style-type: none"> Check the connection status with the FTP server. Review the FTP server settings (connection method). If a firewall or proxy server exists in the connection path, check the settings with the network administrator. |
| C622H | Error occurred during file transfer. | <ul style="list-style-type: none"> Delete unnecessary files from the FTP server to secure free space. Check the connection status with the FTP server. The specified file may be in use by another process, so re-execute after some time. Retry after some time because the line may be congested. |

*1 For the setting and operation method of the FTP server software, check the manual of the FTP server.

Error codes of the Ethernet communication

This section contains an explanation of the contents and method of handling of error codes for errors that occur during various processing for data communication between Ethernet module and other devices, and processing requests from own station.

CPU module

Error codes are stored in built-in Ethernet error code SD10130 (connection 1) to SD10137 (connection 8). However, in case of multiple errors, the error code of the last error that occurred is stored in SD10130 (connection 1) to SD10137 (connection 8).

| Error code (Hexadecimal) | Error details and causes | Action |
|--------------------------|---|--|
| 1120H | Clock setting has failed when the system is powered on or the CPU module is reset. | <ul style="list-style-type: none"> Check if the time settings are correctly set in parameter. Check if the specified SNTP server is operating normally and there is no failure on the network accessing to the SNTP server computer. |
| 112EH | Connection could not be established in open processing. | <ul style="list-style-type: none"> Check the operation of the external device. Check if open processing has been performed in the external device. Correct the port number of the CPU module, IP address/port number of the external device, and opening method. When a firewall is set in the external device, check if access is permitted. Check if the IP address of the external device is not denied in IP Filter Settings. Check if the Ethernet cable is disconnected. |
| 1134H | The external device does not send an ACK response in the TCP/IP communication. | <ul style="list-style-type: none"> Check the operation of the external device. Since there may be congestion of packets on the line, send data after a certain period of time. Check if the connection cable is disconnected. |
| 2160H | Overlapping IP addresses were detected. | Check and correct the IP addresses. |
| 2250H (Stores in SD0) | The protocol setting data stored in the CPU module is not for available modules. | Write the protocol setting data for available modules to the CPU module. |
| C012H | Open processing with the external device failed. (For TCP/IP) | Correct the port numbers of the CPU module and the external device. |
| C013H | Open processing with the external device failed. (For UDP/IP) | Correct the port numbers of the CPU module and the external device. |
| C015H | <ul style="list-style-type: none"> The specified IP address of the external device for the open processing is incorrect. The specified IP address of the external device of the dedicated instruction is incorrect. | Execute the dedicated instruction again after correcting the specified IP address of the external device. |

| Error code (Hexadecimal) | Error details and causes | Action |
|--------------------------|--|---|
| C020H | The send/receive data length exceeds the allowable range. | <ul style="list-style-type: none"> • Correct the data length to be sent. • When the amount of data to be sent exceeds the limit, divide the data into smaller chunks to send it. |
| C024H | Communication using communication protocol is executed in a connection whose connection method is other than communication protocol. | <ul style="list-style-type: none"> • Check that there is no error in the connection number specification of the dedicated instruction. • Correct the communication method of the connection with the external device. |
| C025H | <ul style="list-style-type: none"> • Description of control data is not correct. • Open instruction was executed through open settings parameter even though parameters are not set. | <ul style="list-style-type: none"> • Correct the descriptions of the control data. • Set the open settings parameters. Or, execute the OPEN instruction through control data. |
| C027H | Socket communication send message has failed. | <ul style="list-style-type: none"> • Check the operation of the external device or switching hub. • Since there may be congestion of packets on the line, send data after a certain period of time. • Check if the connection cable is disconnected. • Check that there is no connection failure with the switching hub. • Execute the communication status test, and if the test was completed with an error, take the corrective action. • Execute the module communication test, and check that there is no failure in the module. |
| C029H | <ul style="list-style-type: none"> • Description of control data is not correct. • Open instruction was executed through open settings parameter even though parameters are not set. | <ul style="list-style-type: none"> • Correct the descriptions of the control data. • Set the open settings parameters. Or, execute the OPEN instruction through control data. |
| C035H | The existence of the external device could not be confirmed within the response monitoring timer value. | <ul style="list-style-type: none"> • Check the operation of the external device. • Check if the connection cable is disconnected. |
| C0B2H | There is no sufficient space in the receive buffer or the send buffer of the relay station or external station for the MELSOFT connection, or SLMP. (Send • receive buffer full error) | <ul style="list-style-type: none"> • Increase the request interval (execution interval) and execute the operation. • Do not access through one station using the MELSOFT connection, or SLMP. • Wait for a response to the previous request before sending the next request. |
| C0B6H | The channel specified by the dedicated instruction is out of the range. | Correct the channel to a value within the allowable range of each dedicated instruction. |
| C0DEH | Socket communication receive message has failed. | <ul style="list-style-type: none"> • Check the operation of the external device or switching hub. • Since there may be congestion of packets on the line, send data after a certain period of time. • Check if the connection cable is disconnected. • Check that there is no connection failure with the switching hub. • Execute the communication status test, and if the test was completed with an error, take the corrective action. • Execute the module communication test, and check that there is no failure in the module. |
| C1A2H | A response to the request could not be received. | Check and correct the response waiting time. |
| C1ACH | The specified number of resends is incorrect. | Correct the number of resends. |
| C1ADH | The specified data length is incorrect. | Correct the specified data length. |
| C1AFH | The specified port number is incorrect. | Correct the specified port number. |
| C1B0H | The open processing of the specified connection has been already completed. | <ul style="list-style-type: none"> • Do not perform the open processing to an already opened connection. • When communications with the external device cannot be performed, perform the close processing before the open processing. |
| C1B1H | The open processing of the specified connection has not been completed. | After completion of the open processing, perform the communication. |
| C1B3H | Another send or receive instruction is being executed in the specified channel. | <ul style="list-style-type: none"> • Change the channel number. • Execute again after the send or receive instruction is completed. |
| C1B4H | The specified arrival monitoring time is incorrect. | Set the arrival monitoring time to a value within the allowable range. |
| C1BAH | The dedicated instruction was executed with the initialization not completed. | Execute the dedicated instruction after the initial processing is completed. |
| C1C6H | The execution or error completion type of the dedicated instruction is incorrect. | <ul style="list-style-type: none"> • Execute again after correcting the execution/abnormal end type in the control data. • If the problem cannot be resolved with the above actions, the possible cause is a hardware failure of the module. Please consult your local Mitsubishi representative. |

| Error code (Hexadecimal) | Error details and causes | Action |
|--------------------------|--|---|
| C1CCH | <ul style="list-style-type: none"> A response of the data length that exceeds the allowable range was received by the SLMPSND instruction. The specified request data is incorrect. | <ul style="list-style-type: none"> Execute again after correcting the request data to be within the range. If the error occurs again even after taking the above, please consult your local Mitsubishi representative. |
| C1CDH | Message send of the SLMPSND instruction has failed. | <ul style="list-style-type: none"> Check the operation of the external device or switching hub. Since there may be congestion of packets on the line, send data after a certain period of time. Check if the connection cable is disconnected. Check that there is no connection failure with the switching hub. Execute the communication status test, and if the test was completed with an error, take the corrective action. Execute the module communication test, and check that there is no failure in the module. |
| C1D0H | The requested module I/O No. of the dedicated instruction is incorrect. | <ul style="list-style-type: none"> Execute again after correcting the requested module I/O No. at the request source of the dedicated instruction. If the error occurs again even after taking the above, please consult your local Mitsubishi representative. |
| C1D3H | A dedicated instruction not supported by the communication method of the connection was executed. | <ul style="list-style-type: none"> Check that the dedicated instruction can be executed by the specified communication method. Correct the program if the instruction cannot be executed. Check that there is no error in the connection specification of the dedicated instruction. |
| C400H | The SP.ECPRTCL instruction was executed when "Predefined protocol ready (SD10692)" was "0". | <ul style="list-style-type: none"> Execute the SP.ECPRTCL instruction after "Predefined protocol ready (SD10692)" has become "1". Execute the SP.ECPRTCL instruction after rewriting the protocol setting data to the CPU module. If the error occurs again even after rewriting, replace the CPU module. |
| C401H | <ul style="list-style-type: none"> The control data of the SP.ECPRTCL instruction specified a protocol number not registered in the CPU module. The SP.ECPRTCL instruction was executed while the protocol setting data was not written. | <ul style="list-style-type: none"> Check whether the specified protocol number is correct. Check the presence/absence of protocol registration (SD10722 to SD10725), and then check whether the specified protocol number is registered. Write the protocol setting data, and then execute the SP.ECPRTCL instruction. |
| C404H | The cancel request was received while the protocol was executed, and the SP.ECPRTCL instruction was finished abnormally. | Check the canceled protocol in the control data of the SP.ECPRTCL instruction (execution count result) and eliminate the cause of the cancellation. |
| C405H | The protocol number set value is out of range in the control data of the SP.ECPRTCL instruction. | Correct the protocol number set value. |
| C410H | The receive waiting time timed out. | <ul style="list-style-type: none"> Check if the cable is disconnected. Correct the specified connection number of the external device connection configuration setting, and execute the protocol again. Check that there is no error in the external device. Check that the sending from the external device is not interrupted. Check that there is no data lost due to a receive error. Check that there is no error in the data (packet) sent by the external device. |
| C411H | The received data is larger than 2046 bytes. | <ul style="list-style-type: none"> Check the data sent from the external device. When sending data larger than 2046 bytes from the external device, divide the data into several portions and execute data sending several times. |
| C417H | The data length or data quantity of the received data is out of range. | <ul style="list-style-type: none"> Check the maximum allowable data length and specify the maximum length or less in the data length storage area. Check the maximum allowable data quantity, and specify the maximum quantity or less in the data quantity storage area. |
| C431H | The connection was closed during the SP.ECPRTCL instruction execution. | <ul style="list-style-type: none"> Check the operation of the external device. Check the connection open status with the external device. Open the connection with the external device again and execute the instruction. |
| C614H | The response monitoring timer is timed out. | Since access to the file may take some time, check the setting value of "Response monitoring timer" in "FTP Server Settings" under "Application Settings". |
| CEE0H | The devices supporting iQSS which were detected by the other peripheral device, or other iQSS functions were executed while the automatic detection of connected devices is in process. | Execute the other function after the automatic detection of connected devices is completed. |

| Error code (Hexadecimal) | Error details and causes | Action |
|--------------------------|---|--|
| CEE1H | Incorrect frame is received. | <ul style="list-style-type: none"> • Check the operating status and connection status of the target device. • Check the connection of an Ethernet cable and a hub. • Check the line status of Ethernet. • Reset the CPU module and target device, and retry the operation. If the above actions do not solve the problem, contact the manufacturer of the target device. |
| CEE2H | Incorrect frame is received. | <ul style="list-style-type: none"> • Check the operating status and connection status of the target device. • Check the connection of an Ethernet cable and a hub. • Check the line status of Ethernet. • Reset the CPU module and target device, and retry the operation. If the above actions do not solve the problem, contact the manufacturer of the target device. |
| CF10H | Incorrect frame is received. | <ul style="list-style-type: none"> • Check the operating status and connection status of the target device. • Check the connection of an Ethernet cable and a hub. • Check the line status of Ethernet. • Reset the CPU module and target device, and retry the operation. If the above actions do not solve the problem, contact the manufacturer of the target device. |
| CF20H | <ul style="list-style-type: none"> • The setting value of the communication setting is out of range. • The items of communication setting which cannot be set on the target device are set. • The required setting items have not been set to the target device. | Correct the setting details, and retry the operation. |
| CF30H | The parameter which is not supported by the target device was specified. | Check the version of the target device. |
| CF31H | Incorrect frame is received. | <ul style="list-style-type: none"> • Check the operating status and connection status of the target device. • Check the connection of an Ethernet cable and a hub. • Check the line status of Ethernet. • Reset the CPU module and target device, and retry the operation. If the above actions do not solve the problem, contact the manufacturer of the target device. |
| CF70H | An error occurred on the Ethernet communication path. | <ul style="list-style-type: none"> • Check the operation of the target device. • Check if the connection cable is disconnected. |
| CF71H | A timeout error has occurred. | <ul style="list-style-type: none"> • Check the operation of the target device. Since there may be congestion of packets on the line, perform the operation after a while. • Correct the setting details of when the iQSS function is executed, and retry the operation. • Check the connection of an Ethernet cable and a hub. |
| CFB2H | The same specified own station port number is already used. | Correct the specified port number so that it is not duplicate. |

Ethernet module

The Ethernet communication error codes can be checked in "Status of Each Connection" on the "Ethernet Diagnostics" screen of Ethernet module. (Page 234 Status of Each Connection) The error codes will be stored in 'Error code' (Un\G108 to Un\G139).

| Error code (Hexadecimal) | Error details and causes | Action |
|--------------------------|--|--|
| C012H | <ul style="list-style-type: none"> The port number already used in the open completion connection of TCP/IP is set. The external device port No. set in the external device information is duplicated. | Check and correct the port number of the Ethernet module and the target device. |
| C013H | The port number used in the open completion connection is set during the open processing of UDP/IP. | Check and correct the port number of the Ethernet module. |
| C017H | The connection was not established by the open processing of the TCP connection. | <ul style="list-style-type: none"> Check the operation of the target device. Check the open processing of the target device. Correct the open setting of the communication parameter. Check the port number of the Ethernet module, the IP address/port number of the target device, and the open method. Check if the connection cable is disconnected. |
| C020H | The data length exceeds an allowable range. | <ul style="list-style-type: none"> Correct the data length. If the amount of data to send exceeds the prescribed amount, divide and send the data. |
| C027H | Socket communication send message has failed. | <ul style="list-style-type: none"> Check the operation of the external device or switching hub. Since there may be congestion of packets on the line, send data after a certain period of time. Check if the connection cable is disconnected. Check that there is no connection failure with the switching hub. Execute the communication status test, and if the test was completed with an error, take the corrective action. Execute the module communication test, and check that there is no failure in the module. Check the IP address specified as the target. |
| C029H | Description of control data is not correct. | Correct the descriptions of the control data. |
| C035H | The existence of the external device could not be confirmed within the response monitoring timer value. | <ul style="list-style-type: none"> Check the operation of the external device. Reexamine and change the set values for existence confirmation. Check if the connection cable is disconnected. |
| C04CH | The data cannot be sent since the internal buffer such as IP header buffer has no space. | Send the same data again, and check the receiving of the response. |
| C050H | When the communication data code is set to "ASCII", ASCII code data which cannot be converted to binary is received. | <ul style="list-style-type: none"> For communication, set to "Binary" in the communication data code and restart the Ethernet module. Correct the send data from the target device and send it. |
| C051H | Maximum number of bit devices for which data can be read/written all at once is outside the allowable range. | Correct number of bit devices that can be read or written all at once, and send to Ethernet module again. |
| C052H | Maximum number of word devices for which data can be read/written all at once is outside the allowable range. | Correct number of word devices that can read or write all at once, and send to Ethernet module again. |
| C053H | Maximum number of bit devices for which data can be random read/written all at once is outside the allowable range. | Correct number of bit devices that can be random read or written all at once, and send to Ethernet module again. |
| C054H | Maximum number of word devices for which data can be random read/written all at once is outside the allowable range. | Correct number of word devices that can be random read or written all at once, and send to Ethernet module again. |
| C056H | Read or write request exceeds maximum address. | Correct starting address or number of read and write points, and send to Ethernet module again. (Be careful not to exceed the maximum address.) |
| C057H | The request data length in the SLMP message does not match the number of data in the character section (part of the test). | After reexamining and correcting the content of the text or the request data length in the header, resend the message to the Ethernet module. |
| C058H | Request data length after ASCII-to-binary conversion does not match the number of data in the character section (part of text). | After reviewing and correcting content of text or length of request data in the header, send to Ethernet module again. |
| C059H | <ul style="list-style-type: none"> Error in command or subcommand specification. There is a command or subcommand that cannot be used by the CPU module. | <ul style="list-style-type: none"> Reconsider request contents. Send command or subcommand that can be used by the Ethernet module. |
| C05BH | CPU module cannot read or write from/to specified device. | Reconsider device to read or write. |

| Error code (Hexadecimal) | Error details and causes | Action |
|--------------------------|--|---|
| C05CH | Error in request contents. (Reading or writing by bit unit for word device, etc.) | Correct request content, and send to Ethernet module again. (Subcommand correction, etc.) |
| C05EH | The communication time between the Ethernet module and PLC CPU exceeds the Ethernet monitor timer setting. | <ul style="list-style-type: none"> • Increase the monitor timer setting. • Check the connection between the CPU and Ethernet module. |
| C05FH | There is a request that cannot be executed for the target CPU module. | <ul style="list-style-type: none"> • Correct network No., request station No., request destination module I/O No., or request destination module station No. • Correct contents of write request and/or read request. |
| C060H | Error in request contents. (Error in specification of data for bit device, etc.) | Correct request content, and send to Ethernet module again. (Data correction, etc.) |
| C061H | Request data length does not match the number of data in the character section (part of text). | After reconsidering and correcting content of text or length of request data in the header, send to Ethernet module again. |
| C06FH | When the communication data code is set to "Binary", a request message of ASCII is received. (Error history of this error code is registered but no error response is sent.) | <ul style="list-style-type: none"> • Sent a request message which matches the setting of the communication data code. • Change the communication data code to match the request message. |
| C0D8H | The number of specified blocks exceeds the range. | Correct the specified value of for the number of blocks. |
| C0DEH | Socket communication receive message has failed. | <ul style="list-style-type: none"> • Check the operation of the external device or switching hub. • Since there may be congestion of packets on the line, send data after a certain period of time. • Check if the connection cable is disconnected. • Check that there is no connection failure with the switching hub. • Execute the communication status test, and if the test was completed with an error, take the corrective action. • Execute the module communication test, and check that there is no failure in the module. |
| C1A4H | <ul style="list-style-type: none"> • The operator tried to use the Ethernet diagnosis, CC-Link IEF Basic diagnosis or simple CPU communication diagnosis connecting directly to the Ethernet port of the Ethernet module. • A function not supported for the target device was executed. | Execute the Ethernet diagnosis, CC-Link IEF Basic diagnosis or simple CPU communication diagnosis using the direct connection to the CPU module (built-in Ethernet port). |
| C1A6H | The specification of the connection No. is not correct. | Specify 1 to 32 for the connection No. |
| C1A7H | The specified network number is incorrect. | Correct the specified network number. |
| C1A8H | The specified station number is incorrect. | Correct the specified station number. |
| C1ADH | The specified data length is incorrect. | Correct the specified data length. |
| C1B0H | The open processing of the specified connection has been already completed. | Perform the open processing after completing the close processing. |
| C1B1H | The open processing of the specified connection has not been completed. | Perform the open processing. |
| C1B2H | The specified connection is executing the OPEN/CLOSE instruction. | Execute after the OPEN/CLOSE instruction is completed. |
| C1B3H | The specified channel is being used by another send/receive instruction. | <ul style="list-style-type: none"> • Change the channel number. • Execute after the send/receive instruction is completed. |
| C1D3H | An instruction not conforming to the specifications of the communication method for the connection was executed. | <ul style="list-style-type: none"> • Check that the dedicated instruction can be executed by the specified communication method. Correct the program if the instruction cannot be executed. • Check that there is no error in the connection specification of the dedicated instruction. |
| C709H | A communication error occurred in MELSOFT direct connection. | <ul style="list-style-type: none"> • Do not specify the direct connection when MELSOFT is not directly connected. • In the case of direct connection, do not turn OFF or reset the CPU module or disconnect the cable during communication. |
| CF70H | An error occurred in the Ethernet communication path. | <ul style="list-style-type: none"> • Check the operation of the target device. • Check whether the connection cable is not disconnected. |
| CF71H | A timeout error occurred. | <ul style="list-style-type: none"> • Check the cautions for the executed function. • Check the operation of the target device. • Since there may be congestion of packets on the line, send data after a certain period of time. |

SLMP function error code

Error codes stored when communication ends in error during SLMP are as provided in the following table.

CPU module

3E: 3E frame, 1E: 1E frame, ○: Supported, —: Not supported

| Error code (Hexadecimal) | Error details and causes | Action | SLMP | |
|-----------------------------|--|---|------|----|
| | | | 3E | 1E |
| C035H | The existence of the external device could not be confirmed within the response monitoring timer value. | <ul style="list-style-type: none"> Check the movement of the other party equipment. Check whether the connecting cable has come off. | ○ | ○ |
| C050H | When the communication data code is set to "ASCII", ASCII code data which cannot be converted to binary is received. | <ul style="list-style-type: none"> For communication, set to "Binary" in the communication data code and restart the CPU module. Correct the send data from the target device and send it. | ○ | ○ |
| C051H | Maximum number of bit devices for which data can be read/written all at once is outside the allowable range. | Correct number of bit devices that can be read or written all at once, and send to CPU module again. | ○ | ○ |
| C052H | Maximum number of word devices for which data can be read/written all at once is outside the allowable range. | Correct number of word devices that can read or write all at once, and send to CPU module again. | ○ | ○ |
| C053H | Maximum number of bit devices for which data can be random read/written all at once is outside the allowable range. | Correct number of bit devices that can be random read or written all at once, and send to CPU module again. | ○ | ○ |
| C054H | Maximum number of word devices for which data can be random read/written all at once is outside the allowable range. | Correct number of word devices that can be random read or written all at once, and send to CPU module again. | ○ | ○ |
| C056H | Read or write request exceeds maximum address. | Correct starting address or number of read and write points, and send to CPU module again. (Be careful not to exceed the maximum address.) | ○ | ○ |
| C058H | Request data length after ASCII-to-binary conversion does not match the number of data in the character section (part of text). | After reviewing and correcting content of text or length of request data in the header, send to CPU module again. | ○ | — |
| C059H | <ul style="list-style-type: none"> Error in command or subcommand specification. There is a command or subcommand that cannot be used by the CPU module. | <ul style="list-style-type: none"> Reconsider request contents. Send command or subcommand that can be used by the CPU module. | ○ | ○ |
| C05BH | CPU module cannot read or write from/to specified device. | Reconsider device to read or write. | ○ | ○ |
| C05CH | Error in request contents. (Errors related to device specification, such as reading or writing in bit unit for word device, etc) | Correct request content, and send to CPU module again. (Subcommand correction, etc.) | ○ | ○ |
| C05FH | There is a request that cannot be executed for the target CPU module. | <ul style="list-style-type: none"> Correct network No., request station No., request destination module I/O No., or request destination module station No. Correct contents of write request and/or read request. | ○ | ○ |
| C060H | Error in request contents. (Error in specification of data for bit device, etc.) | Correct request content, and send to CPU module again. (Data correction, etc.) | ○ | ○ |
| C061H | Request data length does not match the number of data in the character section (part of text). | After reconsidering and correcting content of text or length of request data in the header, send to CPU module again. | ○ | ○ |
| C06FH | When the communication data code is set to "Binary", a request message of ASCII is received. (Error history of this error code is registered but no error response is sent.) | <ul style="list-style-type: none"> Sent a request message which matches the setting of the communication data code. Change the communication data code to match the request message. | ○ | — |
| C0D8H | The number of specified blocks exceeds the range. | Correct the specified value of for the number of blocks. | ○ | — |
| C200H | Error in remote password. | Correct remote password, and re-execute remote password lock and unlock. | ○ | — |
| C201H | Locked status of the remote password of the port which is used for communication. | Unlock the remote password before data communication. | ○ | ○ |
| C204H | Different device requested remote password to be unlocked. | Request remote password lock from device that requested unlock of remote password. | ○ | — |
| C810H | Error in remote password. (Authentication failure count is 9 or less.) | Correct remote password, and re-execute remote password unlock. | ○ | — |
| C815H | Error in remote password. (Authentication failure count is 10.) | Re-execute remote password unlock after the specified time elapses. | ○ | — |
| C816H | Remote password authentication is locked out. | Re-execute remote password unlock after the specified time elapses. | ○ | — |

Ethernet module

3E: 3E frame, 1E: 1E frame, ○: Supported, —: Not supported

| Error code (Hexadecimal) | Error details and causes | Action | SLMP | |
|-----------------------------|---|---|------|----|
| | | | 3E | 1E |
| C035H | The existence of the external device could not be confirmed within the response monitoring timer value. | <ul style="list-style-type: none"> • Check the movement of the other party equipment. • Reconsider and change the set values for confirmation of the existence. • Check whether the connecting cable has come off. | ○ | ○ |
| C050H | When the communication data code is set to "ASCII", ASCII code data which cannot be converted to binary is received. | <ul style="list-style-type: none"> • For communication, set to "Binary" in the communication data code and restart the CPU module. • Correct the send data from the target device and send it. | ○ | ○ |
| C051H | Maximum number of bit devices for which data can be read/written all at once is outside the allowable range. | Correct number of bit devices that can be read or written all at once, and send to Ethernet module again. | ○ | ○ |
| C052H | Maximum number of word devices for which data can be read/written all at once is outside the allowable range. | Correct number of word devices that can read or write all at once, and send to Ethernet module again. | ○ | ○ |
| C053H | Maximum number of bit devices for which data can be random read/written all at once is outside the allowable range. | Correct number of bit devices that can be random read or written all at once, and send to Ethernet module again. | ○ | ○ |
| C054H | Maximum number of word devices for which data can be random read/written all at once is outside the allowable range. | Correct number of word devices that can be random read or written all at once, and send to Ethernet module again. | ○ | ○ |
| C056H | Read or write request exceeds maximum address. | Correct starting address or number of read and write points, and send to Ethernet module again. (Be careful not to exceed the maximum address.) | ○ | ○ |
| C057H | The request data length in the SLMP message does not match the number of data in the character section (part of the test). | After reexamining and correcting the content of the text or the request data length in the header, resend the message to the Ethernet module. | ○ | ○ |
| C058H | Request data length after ASCII-to-binary conversion does not match the number of data in the character section (part of text). | After reviewing and correcting content of text or length of request data in the header, send to Ethernet module again. | ○ | — |
| C059H | <ul style="list-style-type: none"> • Error in command or subcommand specification. • There is a command or subcommand that cannot be used by the Ethernet module. | <ul style="list-style-type: none"> • Reconsider request contents. • Send command or subcommand that can be used by the Ethernet module. | ○ | ○ |
| C05BH | Ethernet module cannot read or write from/to specified device. | Reconsider device to read or write. | ○ | ○ |
| C05CH | Error in request contents. (Errors related to device specification, such as reading or writing in bit unit for word device, etc) | Correct request content, and send to Ethernet module again. (Subcommand correction, etc.) | ○ | ○ |
| C05EH | The communication time between the Ethernet module and PLC CPU exceeds the Ethernet monitor timer setting. | <ul style="list-style-type: none"> • Increase the monitor timer setting. • Check the connection between the CPU and Ethernet module. | ○ | ○ |
| C05FH | There is a request that cannot be executed for the target CPU module. | <ul style="list-style-type: none"> • Correct network No., request station No., request destination module I/O No., or request destination module station No. • Correct contents of write request and/or read request. | ○ | ○ |
| C060H | Error in request contents. (Error in specification of data for bit device, etc.) | Correct request content, and send to Ethernet module again. (Data correction, etc.) | ○ | ○ |
| C061H | Request data length does not match the number of data in the character section (part of text). | After reconsidering and correcting content of text or length of request data in the header, send to Ethernet module again. | ○ | ○ |
| C06FH | When the communication data code is set to "Binary", a request message of ASCII is received. (Error history of this error code is registered but no error response is sent.) | <ul style="list-style-type: none"> • Sent a request message which matches the setting of the communication data code. • Change the communication data code to match the request message. | ○ | — |
| C0D8H | The number of specified blocks exceeds the range. | Correct the specified value of for the number of blocks. | ○ | — |

Module error code of Ethernet module

Error codes when a module error occurs are classified into major error, moderate error, and minor error, and can be checked in the [Error Information] tab of the "Module Diagnostics" window of the Ethernet module. (Page 238 Error Information)
The error codes are stored in 'Latest error code' (Un\G29).

| Error code (Hexadecimal) | Error details and causes | Action |
|--------------------------|---|---|
| 0800H | Link-down due to disconnection of the network cable connected to an external device | <ul style="list-style-type: none"> Check the operation of the external device. Check if the connection cable is disconnected. |
| 0904H | Socket communication send message has failed. | <ul style="list-style-type: none"> Check the operation of the external device. Check if the connection cable is disconnected. |
| 0910H | Data cannot be sent to the external device. | <ul style="list-style-type: none"> Check the operation of the external device. Check the conditions of the cables, hubs and routers on the lines to the external devices. The line may be flooded with packets. Retry after a while. The receiving area of the external device may be full (the window size of TCP is small). Confirm whether the receiving processing is performed on the external device side or unnecessary data is not sent from the Ethernet module. Confirm whether the subnet mask pattern and default router IP address are set correctly on the Ethernet module and external device or the IP address class is correct. |
| 0911H | Communication with the external device was interrupted. | <ul style="list-style-type: none"> Check the operation of the external device. Check the conditions of the cables, hubs and routers on the lines to the external devices. This error may occur when the connection under communication is forcibly invalidated. In this case, there is no problem. Re-connect. |
| 0912H | System error or connection connecting error in OS (Malfunction due to noise or hardware trouble may have occurred.) | <ul style="list-style-type: none"> Check the operation of the external device. Check the conditions of the cables, hubs and routers on the lines to the external devices. This error may occur when the connection under communication is forcibly invalidated. In this case, there is no problem. Re-connect. |
| 0913H | The external device cannot be connected or has been disconnected. | <ul style="list-style-type: none"> Check the operation of the external device. Check the conditions of the cables, hubs and routers on the lines to the external devices. If this error occurs during communication, retry after a while. |
| 1080H | The number of writes to ROM exceeded 100000. (Number of writes > 100000) | Replace the module. |
| 1811H, 1812H | An error has occurred during writing of parameters. | Rewrite the EtherNet/IP parameters with EtherNet/IP Configuration Tool for FX5-ENET/IP. |
| 1852H | The property value set in the buffer memory is incorrect. | Confirm that an incorrect value is not stored. |
| 1861H | The data in the transmitted packet exceeded the size that can be transmitted by one packet. | <ul style="list-style-type: none"> Check the settings for the BACnet device of the communication destination. Confirm the causes of increase in packet size. |
| 1870H | The protocol version of NPDU was not 1. | <ul style="list-style-type: none"> Check the specifications of the communication destination device. Check the communication packet. |
| 1871H | Unsupported BVLL (BVLC type) was received. | <ul style="list-style-type: none"> Check the communication packet. Confirm with the manufacturer of the communication destination device. |
| 1872H | Unsupported BVLL (BVLC function) was received. | <ul style="list-style-type: none"> Check the communication packet. Confirm with the manufacturer of the communication destination device. |
| 1873H | 0 was specified for DNET of the received packet. | <ul style="list-style-type: none"> Check the specifications of the communication destination device. Check the communication packet. |
| 1874H | 0 or 65535 was specified for SNET of the received packet. | <ul style="list-style-type: none"> Check the specifications of the communication destination device. Check the communication packet. |
| 1875H | 0 was specified for SLEN of the received packet. | <ul style="list-style-type: none"> Check the specifications of the communication destination device. Check the communication packet. |
| 1876H | An error occurred during packet decoding. | <ul style="list-style-type: none"> Check the communication packet. Please confirm with the manufacturer of the communication destination device. |
| 1877H | SimpleAck, ComplexAck, SegmentAck, Error response, Reject response or Abort response was received. | <ul style="list-style-type: none"> Check the communication packet. Please confirm with the manufacturer of the communication destination device. |

| Error code (Hexadecimal) | Error details and causes | Action |
|--------------------------|--|---|
| 1900H | An memory check error has occurred. | <ul style="list-style-type: none"> Execute the IP address change function again. If the error occurs again even after the above action is taken, the possible cause is a hardware failure of the module on which the error occurred. Please consult your local Mitsubishi representative. |
| 1901H, 1902H | An memory check error has occurred. | <ul style="list-style-type: none"> Execute the hardware test. If the error occurs again even after the above action is taken, the possible cause is a hardware failure of the module on which the error occurred. Please consult your local Mitsubishi representative. |
| 1903H | An memory check error has occurred. | <ul style="list-style-type: none"> Execute Set Attribute for the CIP object with Online Action of EtherNet/IP Configuration Tool for FX5-ENET/IP. If the error occurs again even after the above action is taken, the possible cause is a hardware failure of the module on which the error occurred. Please consult your local Mitsubishi representative. |
| 1904H | An memory check error has occurred. | <ul style="list-style-type: none"> Rewrite the EtherNet/IP parameters (EipConfData.BIN), or configuration information (configuration.apa) with EtherNet/IP Configuration Tool for FX5-ENET/IP. If the error occurs again even after the above action is taken, the possible cause is a hardware failure of the module on which the error occurred. Please consult your local Mitsubishi representative. |
| 1D80H | An instruction other than executable dedicated instruction numbers has been specified. | Check that the dedicated instruction is executable. If it is not executable, correct. |
| 1D83H | The dedicated instruction request data from the CPU module was discarded because the data size was abnormal. | <ul style="list-style-type: none"> Execute the write to program again. If the error occurs again even after the above action is taken, the possible cause is a hardware failure of the module on which the error occurred. Please consult your local Mitsubishi representative. |
| 1D84H | The dedicated instruction request data from the CPU module cannot be normally input. | <ul style="list-style-type: none"> Execute the write to program again. If the error occurs again even after the above action is taken, the possible cause is a hardware failure of the module on which the error occurred. Please consult your local Mitsubishi representative. |
| 1D85H | A timeout error occurred in the dedicated instruction response data, and the data was discarded. | <ul style="list-style-type: none"> Reset the CPU module, and execute again. If the error occurs again even after the above action is taken, the possible cause is a hardware failure of the module on which the error occurred. Please consult your local Mitsubishi representative. |
| 1E10H | An error was detected in the parameters set with EtherNet/IP Configuration Tool for FX5-ENET/IP. | <ul style="list-style-type: none"> Use EtherNet/IP Configuration Tool for FX5-ENET/IP to write the parameters to the module again. If the error occurs again even after the above action is taken, the possible cause is a hardware failure of the module on which the error occurred. Please consult your local Mitsubishi representative. |
| 1E11H | An error was detected in the parameters set with EtherNet/IP Configuration Tool for FX5-ENET/IP. | <ul style="list-style-type: none"> Use EtherNet/IP Configuration Tool for FX5-ENET/IP to write the parameters to the module again. If the error occurs again even after the above action is taken, the possible cause is a hardware failure of the module on which the error occurred. Please consult your local Mitsubishi representative. |
| 1E12H | An error was detected in the parameters set with EtherNet/IP Configuration Tool for FX5-ENET/IP. | <ul style="list-style-type: none"> Use EtherNet/IP Configuration Tool for FX5-ENET/IP to write the parameters to the module again. If the error occurs again even after the above action is taken, the possible cause is a hardware failure of the module on which the error occurred. Please consult your local Mitsubishi representative. |
| 1E13H | It was not possible to stop EtherNet/IP communications. | <ul style="list-style-type: none"> Check the settings of the connection with the external device. If the error occurs again even after the above action is taken, the possible cause is a hardware failure of the module on which the error occurred. Please consult your local Mitsubishi representative. |
| 1E14H | It was not possible to stop EtherNet/IP communications. | <ul style="list-style-type: none"> Check the settings of the connection with the external device. If the error occurs again even after the above action is taken, the possible cause is a hardware failure of the module on which the error occurred. Please consult your local Mitsubishi representative. |
| 1F00H | A communication error occurred in MELSOFT connection. | <ul style="list-style-type: none"> Check the number of connected connections in the whole MELSOFT connection. Check the number of UDP connections in the MELSOFT connection. |
| 2160H | Overlapping IP addresses were detected. | Correct the IP address. |
| 2C80H | A parameter error was detected in the external device configuration. | <ul style="list-style-type: none"> Check the external device configuration settings in the Ethernet-equipped module parameters. If the same error is displayed again, the abnormal module may have a hardware problem. Please consult your local Mitsubishi representative. |

| Error code (Hexadecimal) | Error details and causes | Action |
|--------------------------|--|---|
| 2C81H | A parameter error was detected in the external device configuration. | <ul style="list-style-type: none"> • Check the external device configuration settings in the Ethernet-equipped module parameters. • Set so that the sum of the number of connections in External Device Configuration and the number of settings in the simple CPU communication settings is 32 or less. • If the same error is displayed again, the abnormal module may have a hardware problem. Please consult your local Mitsubishi representative. |
| 2C82H | A parameter error was detected in the external device configuration. | <ul style="list-style-type: none"> • Check the external device configuration settings in the Ethernet-equipped module parameters. • If the same error is displayed again, the abnormal module may have a hardware problem. Please consult your local Mitsubishi representative. |
| 2CF0H | A error of CPU module was detected. | Check the details of the error of the CPU module with the module diagnostics of GX Works3, and take measures. |
| 2DA0H, 2DA1H | A parameter error in external device configuration was detected. | Reexamine and correct the setting of the number of connected units in external device configuration in the Ethernet module parameters. |
| 2DA2H | A parameter error in external device configuration was detected. | Reexamine and correct the setting of the communication method in external device configuration in the Ethernet module parameters. |
| 2DA3H | A parameter error in external device configuration was detected. | Reexamine and correct the setting of the protocol in external device configuration in the Ethernet module parameters. |
| 2DA4H | A parameter error in external device configuration was detected. | Reexamine and correct the setting of the model name in external device configuration in the Ethernet module parameters. |
| 2DA5H | A parameter error in external device configuration was detected. | Reexamine and correct the setting of the fixed buffer transmission in external device configuration in the Ethernet module parameters. |
| 2DA6H | A parameter error in external device configuration was detected. | Reexamine and correct the setting of the IP address in external device configuration in the Ethernet module parameters. |
| 2DA7H | A parameter error in external device configuration was detected. | Reexamine and correct the setting of the IP address (IPv6) in external device configuration in the Ethernet module parameters. |
| 2DA8H | A parameter error in external device configuration was detected. | Reexamine and correct the setting of the existence confirmation in external device configuration in the Ethernet module parameters. |
| 2DA9H, 2DAAH | A parameter error in external device configuration was detected. | Check the setting of External device configuration in the Ethernet module parameters. |
| 2DABH | A parameter error in external device configuration was detected. | Check the own node port number setting in External Device Configuration in Ethernet Module Parameter. |
| 2DB0H | Request and setting data error | Check the content of the specified request data. |
| 3030H | Hardware error | Please consult your local Mitsubishi representative. |
| 3040H | F/W update file version error | For this update, a PLC applicable to the new version is required. Please consult your local Mitsubishi representative. |
| 3041H | F/W update file integrity verification failure error | Replace the update file in the SD memory card with the correct file, and redo update. |
| 3042H | F/W update file acquisition failure | Replace the update file in the SD memory card with the correct file, and redo update. |
| 3056H | The socket communication buffer for receiving has no space. | Read out the received data using the dedicated instruction. |
| 3060H | The total number of connection is outside the range. | Please consult your local Mitsubishi representative. |
| 3061H | The station number is outside the range. | Please consult your local Mitsubishi representative. |
| 3062H | The number of stations occupied is outside the range. | Please consult your local Mitsubishi representative. |
| 3063H | The reserved station specification is outside the range. | Please consult your local Mitsubishi representative. |
| 3064H | The IP address form is outside the range. | Please consult your local Mitsubishi representative. |
| 3065H | The group number is outside the range. | Please consult your local Mitsubishi representative. |
| 3066H | The IP address is outside the range. | Please consult your local Mitsubishi representative. |
| 3067H | The total number of group is outside the range. | Please consult your local Mitsubishi representative. |
| 3068H | The group number is outside the range. | Please consult your local Mitsubishi representative. |
| 306AH | The constant link scan setting is outside the range. | Please consult your local Mitsubishi representative. |
| 306BH | The timeout time to disconnection detection is outside the range. | Please consult your local Mitsubishi representative. |
| 306CH | The consecutive number of time outs to disconnection detection is outside the range. | Please consult your local Mitsubishi representative. |
| 306DH | The IP address form is outside the range. | Please consult your local Mitsubishi representative. |
| 306EH | The IP address (IPv4) is outside the range. | Please consult your local Mitsubishi representative. |

| Error code (Hexadecimal) | Error details and causes | Action |
|--------------------------|--|--|
| 306FH | The subnet mask is outside the range. | Please consult your local Mitsubishi representative. |
| 3073H | The gateway address (IPv4) is outside the range. | Please consult your local Mitsubishi representative. |
| 3074H | The subnet mask is outside the range. | Please consult your local Mitsubishi representative. |
| 3082H | The timer change is outside the specified range. | Please consult your local Mitsubishi representative. |
| 3085H | The destination alive check start interval timer is outside the range. | Please consult your local Mitsubishi representative. |
| 3086H | The start interval timer unit is outside the specified range. | Please consult your local Mitsubishi representative. |
| 3087H | The destination alive check interval timer is outside the range. | Please consult your local Mitsubishi representative. |
| 3088H | The interval timer unit is outside the specified range. | Please consult your local Mitsubishi representative. |
| 3089H | The destination alive check resend count is outside the range. | Please consult your local Mitsubishi representative. |
| 3094H | The number of retry is outside the range. | Please consult your local Mitsubishi representative. |
| 3095H | The number of target IP address setting is outside the range. | Please consult your local Mitsubishi representative. |
| 3096H | The start target IP address setting is outside the range. | Please consult your local Mitsubishi representative. |
| 3097H | The complete target IP address setting is outside the range. | Please consult your local Mitsubishi representative. |
| 3098H | The start target IP address setting is outside the range > the complete target IP address setting is outside the range | Please consult your local Mitsubishi representative. |
| 3099H | The number of excluded IP address setting is outside the range. | Please consult your local Mitsubishi representative. |
| 309AH | The target IP address setting number is outside the range. | Please consult your local Mitsubishi representative. |
| 309BH | The excluded IP address number is outside the range. | Please consult your local Mitsubishi representative. |
| 309CH | The excluded IP address is outside the range. | Please consult your local Mitsubishi representative. |
| 309DH, 309EH | Target module support error | Please consult your local Mitsubishi representative. |
| 309FH | Gateway address (Ipv6) out of range | Please consult your local Mitsubishi representative. |
| 30A8H | The total number of stations occupied is outside the range. | Please consult your local Mitsubishi representative. |
| 30A9H | The master station and slave station IP addresses match | Please consult your local Mitsubishi representative. |
| 30AAH | Simple CPU communication parameter out of range | Please consult your local Mitsubishi representative. |
| 3CF1H | An error of hardware was detected. | Please consult your local Mitsubishi representative. |
| 3E30H to 3E42H | An error of hardware was detected. | Please consult your local Mitsubishi representative. |
| 3E50H to 3E56H | An error of hardware was detected. | Please consult your local Mitsubishi representative. |
| 3E60H to 3E63H | An error of hardware was detected. | Please consult your local Mitsubishi representative. |
| 3F92H | An error was detected in the memory. | <ul style="list-style-type: none"> • Take measures to reduce noise. • Reset the CPU module, and then switch it to RUN mode. If the error occurs again even after the above action is taken, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative. |
| 3FA1H | An error was detected in the memory. | <ul style="list-style-type: none"> • Take measures to reduce noise. • Reset the CPU module, and then switch it to RUN mode. If the error occurs again even after the above action is taken, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative. |
| 4030H | The specified device name cannot be handled. | Check the specified device name. |
| 4031H | <ul style="list-style-type: none"> • The specified device No. is out of the range. • The external device does not correspond to the specified device name. | <ul style="list-style-type: none"> • Check the specified device No. • Check the external device allocation. • Check the specified device name. |
| 4032H | A device name (TS, TC, SS, SC, CS or CC) that cannot be used for SLMP random reading/writing (in word units) was specified. | <ul style="list-style-type: none"> • Check the specified device modification method. • Check the specified device name. |
| 4033H | The specified device is for the system, and data cannot be written into it. | Do not write data into the specified device or turn ON or OFF the device. |
| 4041H | The access range exceeds the buffer memory range of the specified intelligent function module. | Check the starting address and the number of access points, and access within the buffer memory range existing in the intelligent function module. |
| 4043H | The specified intelligent function module does not exist. | Check the module number of the specified intelligent function module. |
| 480CH | The operation cannot be executed because the automatic detection function is being executed. | Re-execute after the completion of the automatic detection function. |
| 480DH | The operation cannot be executed because the communication setting reflection function is being executed. | Re-execute after the completion of the communication setting reflection function. |
| 480EH | The operation cannot be executed because the parameter read/write function is being executed. | Re-execute after the completion of the parameter read/write function. |

15.5 List of Event Code

The following table lists events that occur in the Ethernet module.

| Event code | Event type | Event category | Event status | Detected event | Detailed information | | |
|------------|------------|----------------|--------------|---|---|--|------------------------|
| | | | | | Detailed information 1 | Detailed information 2 | Detailed information 3 |
| 0800 | System | Error | Minor | Link-down | Operation source information | Communication speed and communication mode | — |
| 0904 | System | Error | Minor | Socket communication sending failure | Operation source information | — | — |
| 1080 | System | Error | Major | ROM write count error | Number of times information* ¹ | — | — |
| 1810 | System | Error | Minor | IP address change fails | — | — | — |
| 1811 | System | Error | Minor | Flash memory write error (EtherNet/IP parameter) | — | — | — |
| 1812 | System | Error | Minor | FTP response send timeout (EtherNet/IP parameter transfer) | — | — | — |
| 1852 | System | Error | Minor | Out-of-range setting error | Buffer memory information | — | Failure information |
| 1900 | System | Error | Minor | Flash memory check error (IP address change function sector) | — | — | — |
| 1901 | System | Error | Minor | Flash memory check error (Total number sector of writes to memory Flash ROM) | — | — | — |
| 1902 | System | Error | Minor | Flash memory check error (Flash memory test sector) | — | — | — |
| 1903 | System | Error | Minor | Flash memory check error (TCP Inactivity Timeout sector) | — | — | — |
| 1904 | System | Error | Minor | Flash memory check error (EtherNet/IP parameter sector) | — | — | — |
| 1E10 | System | Error | Minor | EtherNet/IP communication error | — | — | Failure information |
| 1E11 | System | Error | Minor | EtherNet/IP communication error | — | — | — |
| 1E12 | System | Error | Minor | EtherNet/IP communication error | — | — | — |
| 1E13 | System | Error | Minor | EtherNet/IP communication error | — | — | — |
| 1E14 | System | Error | Minor | EtherNet/IP communication error | — | — | — |
| 1F00 | System | Error | Minor | MELSOFT connection error | — | — | — |
| 2160 | System | Error | Moderate | IP address duplication detection | — | — | Failure information |
| 2C80 | System | Error | Moderate | Receive parameter sum value error | — | — | Failure information |
| 2C81 | System | Error | Moderate | Receive parameter data error | — | — | Failure information |
| 2C82 | System | Error | Moderate | Divided parameter receive error | — | — | Failure information |
| 2CF0 | System | Error | Moderate | Ethernet module WDT error occurs | — | — | Failure information |
| 2DA0 | System | Error | Moderate | Connection setting parameter (Error in the number of connection setting) | Parameter information* ² | — | Failure information |
| 2DA1 | System | Error | Moderate | Connection setting parameter (Connection number error) | Parameter information* ² | — | Failure information |
| 2DA2 | System | Error | Moderate | Connection setting parameter (Communication destination communication method error) | Parameter information* ² | — | Failure information |
| 2DA3 | System | Error | Moderate | Connection setting parameter (Protocol system error) | Parameter information* ² | — | Failure information |
| 2DA4 | System | Error | Moderate | Connection setting parameter (Open system error) | Parameter information* ² | — | Failure information |

| Event code | Event type | Event category | Event status | Detected event | Detailed information | | |
|------------|------------|----------------|--------------|--|-------------------------------------|------------------------|------------------------|
| | | | | | Detailed information 1 | Detailed information 2 | Detailed information 3 |
| 2DA5 | System | Error | Moderate | Connection setting parameter (Fixed buffer transmitting method error) | Parameter information* ² | — | Failure information |
| 2DA6 | System | Error | Moderate | Connection setting parameter (IP address (IPv4) error) | Parameter information* ² | — | Failure information |
| 2DA7 | System | Error | Moderate | Connection setting parameter (IP address (IPv6) error) | Parameter information* ² | — | Failure information |
| 2DA8 | System | Error | Moderate | Connection setting parameter (Existence confirmation specification error) | Parameter information* ² | — | Failure information |
| 2DA9 | System | Error | Moderate | Connection setting parameter (Communication data code error) | Parameter information* ² | — | Failure information |
| 2DAA | System | Error | Moderate | Connection setting parameter (Error in specification of writing during running) | Parameter information* ² | — | Failure information |
| 2DAB | System | Error | Moderate | Connection setting parameter own node port number error | Parameter information | — | — |
| 2DB0 | System | Error | Moderate | Request data error | — | — | Failure information |
| 3030 | System | Error | Moderate | Specific code error | — | — | — |
| 3040 | System | Error | Moderate | Firmware update file version error | — | — | — |
| 3041 | System | Error | Moderate | Failure in verification of firmware update file integrity | — | — | — |
| 3042 | System | Error | Moderate | Failure in acquisition of firmware update file | — | — | — |
| 3056 | System | Error | Moderate | Socket communication buffer full | — | — | Failure information |
| 3060 | System | Error | Moderate | The total number of connection is outside the range (Configuration setting parameter) | Parameter information* ² | — | — |
| 3061 | System | Error | Moderate | The station number is outside the range (Configuration setting parameter) | Parameter information* ² | — | — |
| 3062 | System | Error | Moderate | The number of stations occupied is outside the range (Configuration setting parameter) | Parameter information* ² | — | — |
| 3063 | System | Error | Moderate | The reserved station specification is outside the range (Configuration setting parameter) | Parameter information* ² | — | — |
| 3064 | System | Error | Moderate | The IP address form is outside the range (Configuration setting parameter) | Parameter information* ² | — | — |
| 3065 | System | Error | Moderate | The group number is outside the range (Configuration setting parameter) | Parameter information* ² | — | — |
| 3066 | System | Error | Moderate | The IP address is outside the range (Configuration setting parameter) | Parameter information* ² | — | — |
| 3067 | System | Error | Moderate | The total number of group is outside the range (Configuration parameter according to group) | Parameter information* ² | — | — |
| 3068 | System | Error | Moderate | The group number is outside the range (Configuration parameter according to group) | Parameter information* ² | — | — |
| 306A | System | Error | Moderate | The constant link scan setting is outside the range (Configuration parameter according to group) | Parameter information* ² | — | — |
| 306B | System | Error | Moderate | The timeout time to disconnection detection is outside the range (Configuration parameter according to group) | Parameter information* ² | — | — |
| 306C | System | Error | Moderate | The consecutive number of time outs to disconnection detection is outside the range (Configuration parameter according to group) | Parameter information* ² | — | — |

| Event code | Event type | Event category | Event status | Detected event | Detailed information | | |
|------------|------------|----------------|--------------|--|-------------------------------------|------------------------|------------------------|
| | | | | | Detailed information 1 | Detailed information 2 | Detailed information 3 |
| 306D | System | Error | Moderate | The IP address form is outside the range (IP address setting parameter) | Parameter information* ² | — | — |
| 306E | System | Error | Moderate | The IP address (IPv4) is outside the range (IP address setting parameter) | Parameter information* ² | — | — |
| 306F | System | Error | Moderate | The subnet mask is outside the range (IP address setting parameter) | Parameter information* ² | — | — |
| 3073 | System | Error | Moderate | The gateway address (IPv4) is outside the range (Gateway address setting parameter) | Parameter information* ² | — | — |
| 3074 | System | Error | Moderate | The subnet mask is outside the range (Gateway address setting parameter) | Parameter information* ² | — | — |
| 3082 | System | Error | Moderate | The timer change is outside the specified range (Data communication timer setting parameter) | Parameter information* ² | — | — |
| 3085 | System | Error | Moderate | The destination alive check start interval timer is outside the range (Data communication timer setting parameter) | Parameter information* ² | — | — |
| 3086 | System | Error | Moderate | The start interval timer unit is outside the specified range (Data communication timer setting parameter) | Parameter information* ² | — | — |
| 3087 | System | Error | Moderate | The destination alive check interval timer is outside the range (Data communication timer setting parameter) | Parameter information* ² | — | — |
| 3088 | System | Error | Moderate | The interval timer unit is outside the specified range (Data communication timer setting parameter) | Parameter information* ² | — | — |
| 3089 | System | Error | Moderate | The destination alive check resend count is outside the range (Data communication timer setting parameter) | Parameter information* ² | — | — |
| 3094 | System | Error | Moderate | The number of retry is outside the range (Data communication timer setting parameter) | Parameter information* ² | — | — |
| 3095 | System | Error | Moderate | The number of target IP address setting is outside the range (IP filter settings) | Parameter information* ² | — | — |
| 3096 | System | Error | Moderate | IP address 1 is outside the range (IP filter settings) | Parameter information* ² | — | — |
| 3097 | System | Error | Moderate | IP address 2 is outside the range (IP filter settings) | Parameter information* ² | — | — |
| 3098 | System | Error | Moderate | IP address 1 ≥ IP address 2 error (IP filter settings) | Parameter information* ² | — | — |
| 3099 | System | Error | Moderate | The number of excluded IP address setting is outside the range (IP filter settings) | Parameter information* ² | — | — |
| 309A | System | Error | Moderate | The target IP address setting number is outside the range (IP filter settings) | Parameter information* ² | — | — |
| 309B | System | Error | Moderate | The excluded IP address number is outside the range (IP filter settings) | Parameter information* ² | — | — |
| 309C | System | Error | Moderate | The excluded IP address is outside the range (IP filter settings) | Parameter information* ² | — | — |
| 309D | System | Error | Moderate | Target module support error (IP address settings parameter) | Parameter information* ² | — | — |
| 309E | System | Error | Moderate | Target module support error (Gateway address setting parameter) | Parameter information* ² | — | — |
| 309F | System | Error | Moderate | The gateway address (IPv6) is outside the range (Gateway address setting parameter) | Parameter information* ² | — | — |

| Event code | Event type | Event category | Event status | Detected event | Detailed information | | |
|------------|------------|----------------|--------------|--|-------------------------------------|------------------------|------------------------|
| | | | | | Detailed information 1 | Detailed information 2 | Detailed information 3 |
| 30A8 | System | Error | Moderate | The total number of stations occupied is outside the range (Configuration setting parameter) | Parameter information* ² | — | — |
| 30A9 | System | Error | Moderate | The master station and slave station IP addresses match (Configuration setting parameter) | Parameter information* ² | — | — |
| 3CF1 | System | Error | Major | BINT disconnection detection timeout error | — | — | Failure information |
| 3E30 | System | Error | Major | The fixed memory block acquisition/release ID number is invalid. | — | — | Failure information |
| 3E31 | System | Error | Major | Fixed memory block acquisition/release context error | — | — | Failure information |
| 3E32 | System | Error | Major | Error in forced cancellation of waiting for variable memory block acquisition/release | — | — | Failure information |
| 3E33 | System | Error | Major | Error in forced cancellation of waiting due to reset of fixed memory block acquisition/release object | — | — | Failure information |
| 3E34 | System | Error | Major | Fixed memory block acquisition/release parameter error | — | — | Failure information |
| 3E35 | System | Error | Major | The variable memory block acquisition/release ID number is invalid | — | — | Failure information |
| 3E36 | System | Error | Major | Variable memory block acquisition/release context error | — | — | Failure information |
| 3E37 | System | Error | Major | Error in forced cancellation of waiting for variable memory block acquisition/release | — | — | Failure information |
| 3E38 | System | Error | Major | Error in forced cancellation of waiting due to reset of variable memory block acquisition/release object | — | — | Failure information |
| 3E39 | System | Error | Major | Variable memory block acquisition/release parameter error | — | — | Failure information |
| 3E3A | System | Error | Major | E-mail box transmitting/receiving parameter error | — | — | Failure information |
| 3E3B | System | Error | Major | The e-mail box transmitting/receiving ID number is invalid | — | — | Failure information |
| 3E3C | System | Error | Major | E-mail box transmitting/receiving context error | — | — | Failure information |
| 3E3D | System | Error | Major | Error in forced cancellation of waiting for e-mail box transmitting/receiving | — | — | Failure information |
| 3E3E | System | Error | Major | Semaphore acquisition/release parameter error | — | — | Failure information |
| 3E3F | System | Error | Major | Invalid semaphore acquisition/release ID number | — | — | Failure information |
| 3E40 | System | Error | Major | Semaphore acquisition/release context error | — | — | Failure information |
| 3E41 | System | Error | Major | Semaphore acquisition/release queueing overflow | — | — | Failure information |
| 3E42 | System | Error | Major | Error in forced cancellation of waiting for semaphore acquisition/release | — | — | Failure information |
| 3E50 | System | Error | Major | RAM check error | — | — | — |
| 3E51 | System | Error | Major | Sum check code error | — | — | — |
| 3E52 | System | Error | Major | Flash memory test access error | — | — | — |
| 3E53 | System | Error | Major | Flash memory test verification error | — | — | — |
| 3E54 | System | Error | Major | Buffer memory access error | — | — | — |
| 3E55 | System | Error | Major | BusAsic register read error | — | — | — |
| 3E56 | System | Error | Major | Factory test mode error | — | — | — |

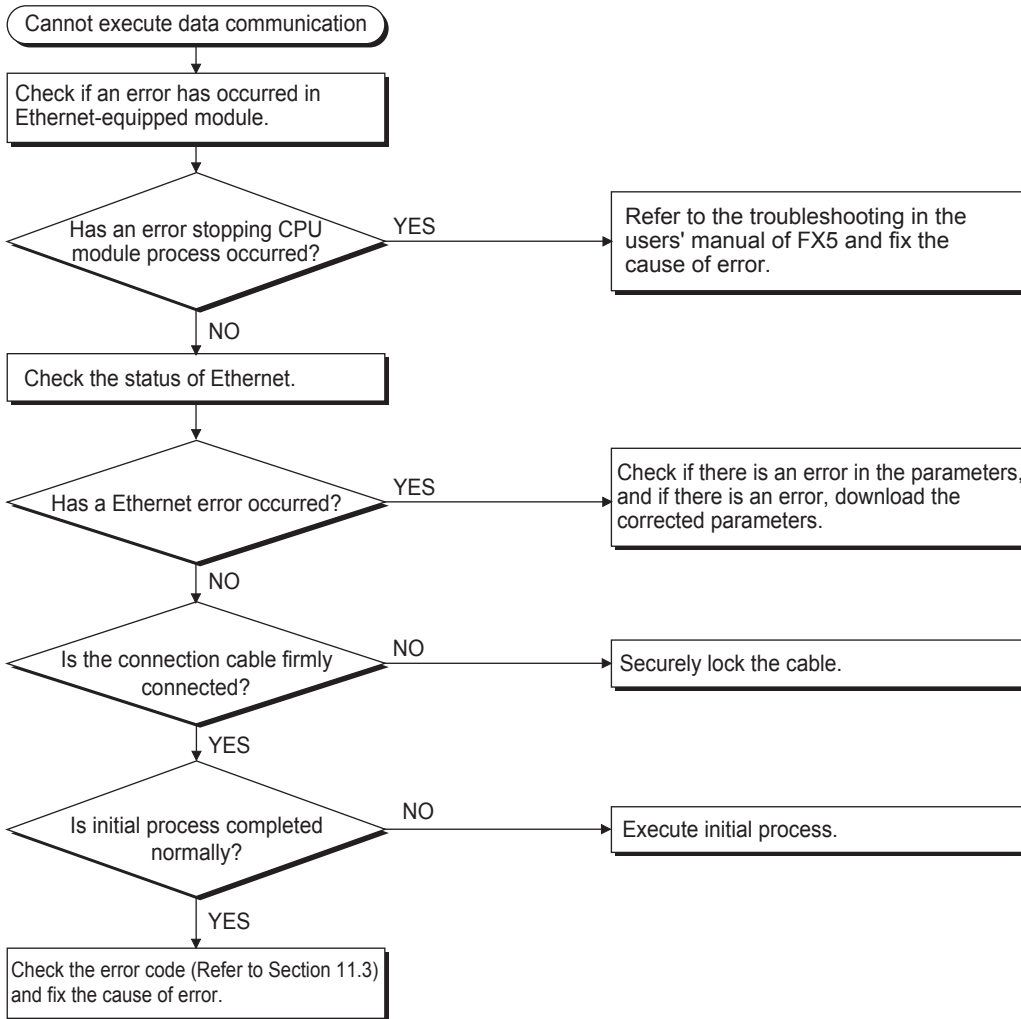
| Event code | Event type | Event category | Event status | Detected event | Detailed information | | |
|--------------|------------|----------------|--------------|------------------------------|------------------------|------------------------|------------------------|
| | | | | | Detailed information 1 | Detailed information 2 | Detailed information 3 |
| 3E60 to 3E63 | System | Error | Major | MPU error | — | — | Failure information |
| 3F92 | System | Error | Major | Memory error | — | — | Failure information |
| 3FA1 | System | Error | Major | Memory error | — | — | Failure information |
| C050 | System | Error | Minor | Ethernet communication error | Parameter information | — | — |
| C051 | System | Error | Minor | Ethernet communication error | Parameter information | — | — |
| C052 | System | Error | Minor | Ethernet communication error | Parameter information | — | — |
| C053 | System | Error | Minor | Ethernet communication error | Parameter information | — | — |
| C054 | System | Error | Minor | Ethernet communication error | Parameter information | — | — |
| C056 | System | Error | Minor | Ethernet communication error | Parameter information | — | — |
| C057 | System | Error | Minor | Ethernet communication error | Parameter information | — | — |
| C058 | System | Error | Minor | Ethernet communication error | Parameter information | — | — |
| C059 | System | Error | Minor | Ethernet communication error | Parameter information | — | — |
| C05B | System | Error | Minor | Ethernet communication error | Parameter information | — | — |
| C05C | System | Error | Minor | Ethernet communication error | Parameter information | — | — |
| C05E | System | Error | Minor | Ethernet communication error | Parameter information | — | — |
| C05F | System | Error | Minor | Ethernet communication error | Parameter information | — | — |
| C060 | System | Error | Minor | Ethernet communication error | Parameter information | — | — |
| C061 | System | Error | Minor | Ethernet communication error | Parameter information | — | — |
| C06F | System | Error | Minor | Ethernet communication error | Parameter information | — | — |
| C0D8 | System | Error | Minor | Ethernet communication error | Parameter information | — | — |

*1 <Number of times information>
 · Number of times (set value)

*2 <Parameter information>
 · Parameter storage location
 · Parameter type
 · (I/O number)
 · (Parameter number)
 · Parameter item number

15.6 Troubleshooting Flowchart

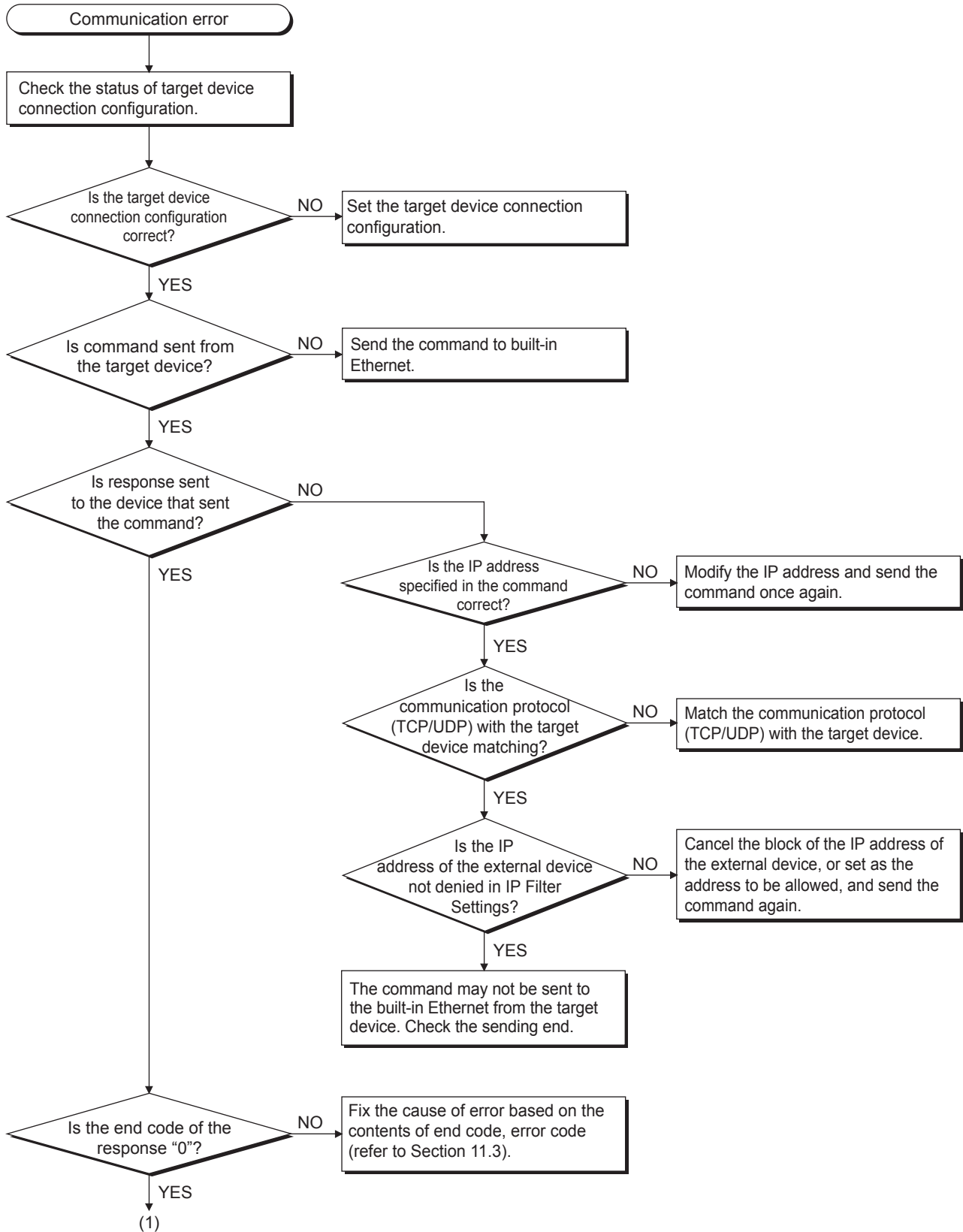
Simple troubleshooting when communication cannot be carried out between Ethernet-equipped module and other device is provided in the form of a flowchart.

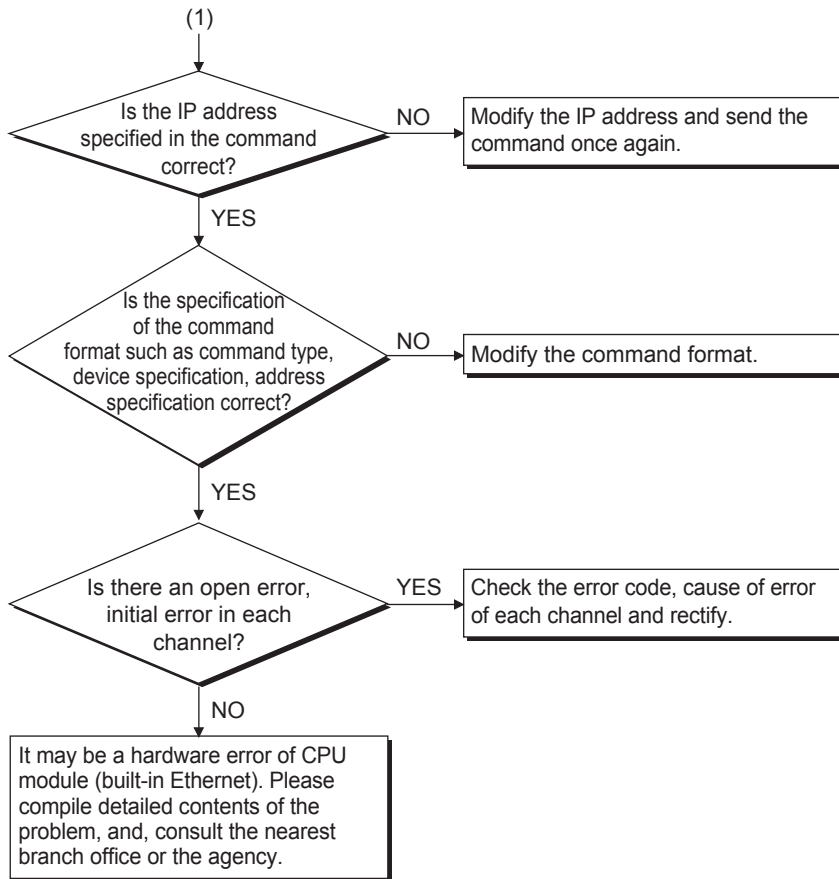


Point

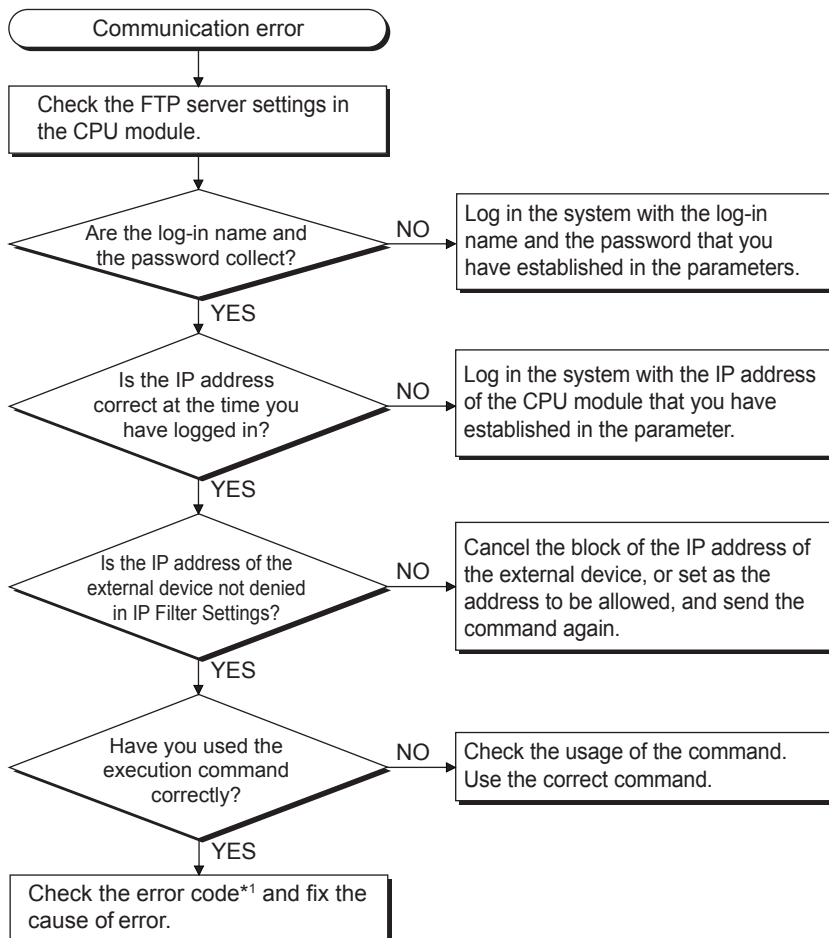
- If trouble occurs while using Ethernet, check the error status with the Ethernet diagnostics function of GX Works3. For error contents, refer to Page 249 Error codes of the Ethernet communication.
- For information concerning when the [ERR] LED or [ERROR] LED is lit, refer to Page 231 Checking Errors by LEDs.
- If Ethernet-equipped module is replaced due to error occurrence, reboot all target devices with which communication was being done, and restart data communication. (If target device retains Ethernet address of communication target, continuation of communication may not be possible in some cases because Ethernet address changes when the Ethernet-equipped module is replaced (includes unit-specific addresses such as MAC address.)
- If target device (such as a personal computer) is replaced, turn CPU module's power OFF→ON and/or perform reset.
- If message transmitted from target device cannot be received by Ethernet (error log is long) frequently, there may be a large load on the Ethernet line due to data transmitted among the various devices connected. In order to reduce load on the Ethernet line, you may have to take measures such as dividing the network or reduce the number of data transmissions. After conferring with the network administrator, reduce the load on the Ethernet line.
- When the ground terminal of the Ethernet-equipped module cannot be grounded, the communication line may be closed due to the effects of noise, making it impossible to communicate with other devices.

Errors during SLMP communication





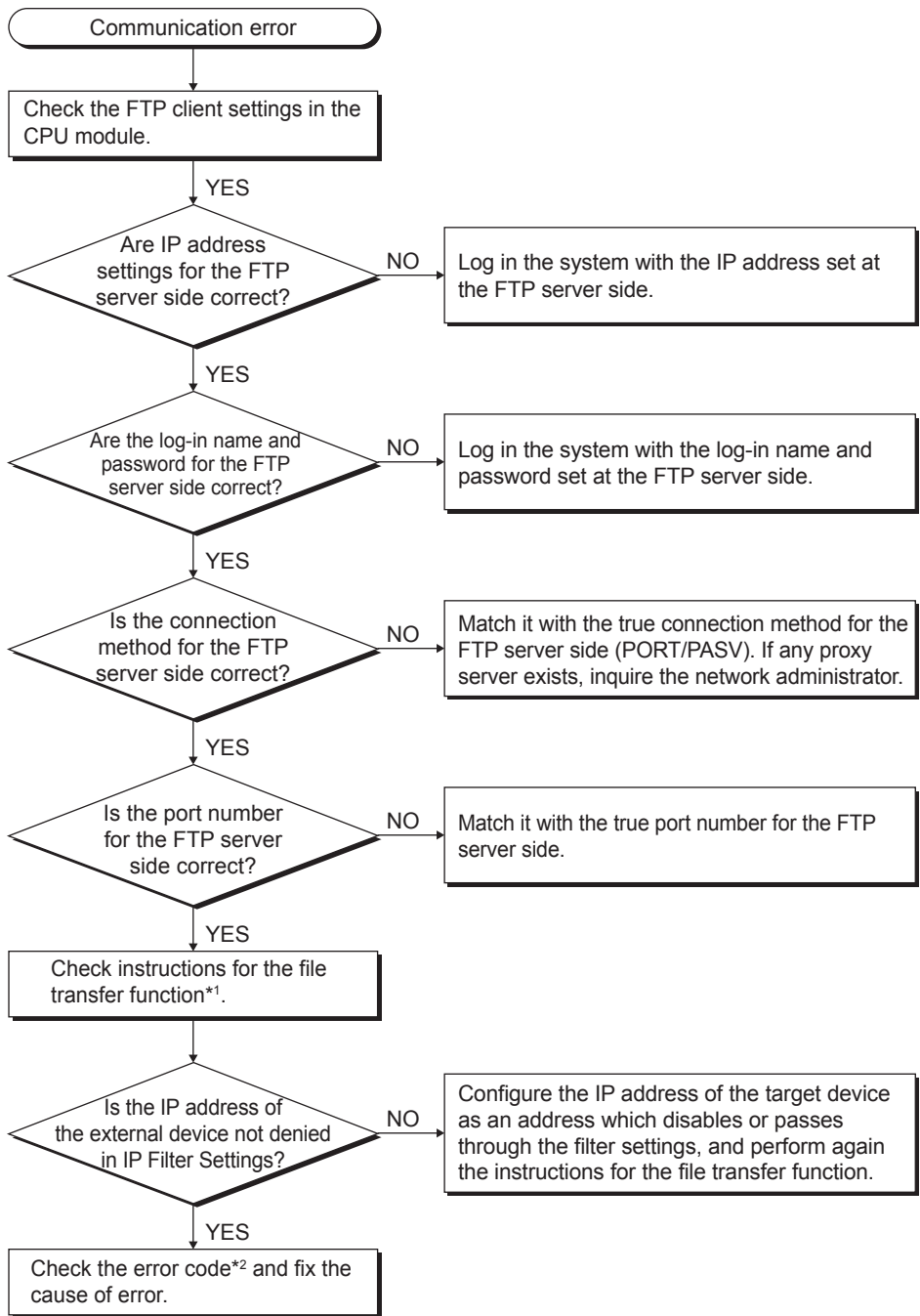
Errors during file transfer function (FTP server)



*1 Refer to Page 245 Error Codes.

Errors during file transfer function (FTP client)

For check items other than those shown below, refer to Page 161 Precautions.




*1 Refer to MELSEC iQ-F FX5 Programming Manual (Instructions, Standard Functions/Function Blocks).

*2 Refer to Page 245 Error Codes.

APPENDIX

Appendix 1 List of Special Device Applications and Assignments

For special relays and special registers other than described below, refer to  MELSEC iQ-F FX5 User's Manual (Application).

Special relays


| Device No. | Name | Description | R/W |
|------------|---|--|-----|
| SM1392*1 | FTP client connection status | This relay turns on when the connection with the FTP server is established. This relay turns off when the connection with the FTP server is cut off. | R |
| SM8492 | IP address storage area write request | Writes IP address setting stored in SD8492 to SD8497 to IP address storage area when this device turns from OFF to ON. | R/W |
| SM8493 | IP address storage area write completed | <ul style="list-style-type: none">• Turns ON when writing to IP address storage area completes or fails.• Turns OFF when IP address storage area write request (SM8492) turns from ON to OFF. | R |
| SM8494 | IP address storage area write error | <ul style="list-style-type: none">• Turns ON when writing to IP address storage area fails.• Turns ON if there is a problem in contents of IP address storage area, when PLC power supply is turned from OFF to ON.• Turns OFF when IP address storage area write request (SM8492) turns from ON to OFF. | R |
| SM8495 | IP address storage area clear request | Contents of IP address storage area are cleared when this device turns from OFF to ON. | R/W |
| SM8496 | IP address storage area clear completed | <ul style="list-style-type: none">• Turns ON when clearing of IP address storage area completes or fails.• Turns OFF when IP address storage area clear request (SM8495) turns from ON to OFF. | R |
| SM8497 | IP address storage area clear error | <ul style="list-style-type: none">• Turns ON when clearing of IP address storage area fails.• Turns OFF when IP address storage area clear request (SM8495) turns from ON to OFF. | R |
| SM8498 | IP address change function enable flag | Turns ON when IP address is changed by IP address change function. | R |

R: Read only, R/W: Read/Write

*1 Only FX5U/FX5UC CPU module is supported.

Special registers

| Device No. | Name | Description | R/W |
|------------|---|--|-----|
| SD8492 | IP address setting (Low-order) | <ul style="list-style-type: none">• Stores IP address (low-order) to be set when using IP address change function.• Becomes 0 when writing to IP address storage area is completed normally. | R/W |
| SD8493 | IP address setting (High-order) | <ul style="list-style-type: none">• Stores IP address (high-order) to be set when using IP address change function.• Becomes 0 when writing to IP address storage area is completed normally. | R/W |
| SD8494 | Subnet mask setting (Low-order) | <ul style="list-style-type: none">• Stores subnet mask (low-order) to be set when using IP address change function.• Becomes 0 when writing to IP address storage area is completed normally. | R/W |
| SD8495 | Subnet mask setting (High-order) | <ul style="list-style-type: none">• Stores subnet mask (high-order) to be set when using IP address change function.• Becomes 0 when writing to IP address storage area is completed normally. | R/W |
| SD8496 | Default gateway IP address setting (Low-order) | <ul style="list-style-type: none">• Stores default gateway IP address (low-order) to be set when using IP address change function.• Becomes 0 when writing to IP address storage area is completed normally. | R/W |
| SD8497 | Default gateway IP address setting (High-order) | <ul style="list-style-type: none">• Stores default gateway IP address (high-order) to be set when using IP address change function.• Becomes 0 when writing to IP address storage area is completed normally. | R/W |
| SD8498 | IP address storage area write error code | Stores error codes if writing to IP address storage area fails. | R |
| SD8499 | IP address storage area clear error code | Stores error codes if clearing of IP address storage area fails. | R |
| SD10050 | IP address (Low-order) | Lower part of the IP address. | R |
| SD10051 | IP address (High-order) | Higher part of the IP address. | R |

| Device No. | Name | Description | R/W |
|--------------------|---|---|-----|
| SD10060 | Subnet mask (Low-order) | Lower part of the subnet mask setting value. | R |
| SD10061 | Subnet mask (High-order) | Higher part of the subnet mask setting value. | R |
| SD10064 | Default gateway IP address (Low-order) | Lower part of the default gateway IP address setting value. | R |
| SD10065 | Default gateway IP address (High-order) | Higher part of the default gateway IP address setting value. | R |
| SD10074 to SD10076 | Host MAC address | MAC address (3 words in total) is stored. | R |
| SD10082 | Communication speed setting | Communication speed setting is stored. 0000H: Automatic Negotiation 0002H: 100Mbps/Half-Duplex 0003H: 100Mbps/Full-Duplex 0004H: 10Mbps/Half-Duplex 0005H: 10Mbps/Full-Duplex | R |
| SD10084 | MELSOFT connection TCP port number | MELSOFT connection TCP port number is stored. | R |
| SD10086 | MELSOFT direct connection port number | MELSOFT direct connection port number is stored. | R |
| SD10130 to SD10137 | Error code | Error code of built-in Ethernet (connection 1 to connection 8) is stored. For details of error code, refer to  Page 245 Error Codes. | R |
| SD10251 | Same IP address state storage area | Same IP address state is stored. [b0]: Same IP address detection flag 0: No same IP address 1: Same IP address | R |
| SD10252 to SD10254 | MAC address of the already connected station | Stores the MAC address of the station, which was connected to the network earlier, in the station with duplicated IP address. SD10252: 5th byte, 6th byte of the MAC address SD10253: 3rd byte, 4th byte of the MAC address SD10254: 1st byte, 2nd byte of the MAC address "FFFFFFFFFFH" is stored in the station that has been already connected to the network. | R |
| SD10255 to SD10257 | MAC address of the station connected later | Stores the MAC address of the station with duplicated IP address in the station which was connected earlier to the network. SD10255: 5th byte, 6th byte of the MAC address SD10256: 3rd byte, 4th byte of the MAC address SD10257: 1st byte, 2nd byte of the MAC address "FFFFFFFFFFH" is stored in the station with duplicated IP address. | R |
| SD10270 | Remote password information remote password locked status (Connection No. 1 to 8) | Locked status of the remote password for each connection [b0] to [b7]: Connection No. 1 to No. 8 0: Unlocked status/No remote password setting 1: Locked status | R |
| SD10271 | Remote password information remote password locked status (System port) | The locked status of the remote password of the system port. [b2]: MELSOFT application communication port (TCP) [b3]: Direct connection with MELSOFT [b4]: FTP transmission port 0: Unlocked status/No remote password setting 1: Locked status | R |
| SD10290 | Time setting function operation result | Stores the operation result of the time setting function. 0000H: Unexecuted 0001H: Success FFFFH: Failure | R |
| SD10291 | Time setting function execution time (Year) | The year (A.D.) which the time setting function is executed is stored in a binary code. When the communication fails, this device is not updated from the value stored previously. | R |
| SD10292 | Time setting function execution time (Month) | The month which the time setting function is executed is stored in a binary code. When the communication fails, this device is not updated from the value stored previously. | R |
| SD10293 | Time setting function execution time (Day) | The day which the time setting function is executed is stored in a binary code. When the communication fails, this device is not updated from the value stored previously. | R |
| SD10294 | Time setting function execution time (Hour) | The hour which the time setting function is executed is stored in a binary code. When the communication fails, this device is not updated from the value stored previously. | R |
| SD10295 | Time setting function execution time (Minute) | The minute which the time setting function is executed is stored in a binary code. When the communication fails, this device is not updated from the value stored previously. | R |
| SD10296 | Time setting function execution time (Second) | The second which the time setting function is executed is stored in a binary code. When the communication fails, this device is not updated from the value stored previously. | R |



| Device No. | Name | Description | R/W |
|--------------------|--|--|-----|
| SD10297 | Time setting function execution time (Day of the week) | The day of the week which the time setting function is executed is stored in a binary code. 0: Sunday 1: Monday 2: Tuesday 3: Wednesday 4: Thursday 5: Friday 6: Saturday When the communication fails, this device is not updated from the value stored previously. | R |
| SD10298 | Time setting function required response time | A time required from sending the message to the SNTP server to receiving the response and setting the time to the CPU module is stored. 0000H to FFFE H (Unit: ms) If the value exceeds the above range, all the values are stored as FFFFH. When the communication fails, this device is not updated from the value stored previously. | R |
| SD10299 | Time setting function (SNTP client) execution | Executes the time setting function when b0 is turned on. (Only when the time setting (SNTP client) is set to "Use" in the GX Works3.) The function is not executed if b0 is turned on during execution of the time setting function. | R/W |
| SD10320 to SD10327 | Unlock failure count | Unlock failure counts are stored. [SD10320] to [SD10327]: Connection No. 1 to No. 8 | R |
| SD10338 | MELSOFT connection TCP port continuous unlock failure count | Unlock failure counts of the MELSOFT connection (via hub) are stored. | R |
| SD10339 | FTP transmission port (TCP/IP) continuous unlock failure count | FTP transmission port (TCP/IP) continuous unlock failure count is stored. | R |
| SD10340 | Direct connection with MELSOFT continuous unlock failure count | Unlock failure counts of the MELSOFT connection (direct connection) are stored. | R |
| SD10352, SD10353 | Simple PLC Communication stop request | Request contact to stop data transmission when the communication setting for the simple CPU communication is "Fixed" SD10352[b0] to [b15]: Setting No.1 to Setting No.16 SD10353[b0] to [b15]: Setting No.17 to Setting No.32*1 0 to 1: Requested (stop request) 1 to 0: Completed (stop completion) | R/W |
| SD10354, SD10355 | Simple PLC Communication restart request | Request contact to restart data transmission when the communication setting for the simple CPU communication is "Fixed" SD10354[b0] to [b15]: Setting No.1 to Setting No.16 SD10355[b0] to [b15]: Setting No.17 to Setting No.32*1 0 to 1: Requested (restart request) 1 to 0: Completed (restart completion) | R/W |
| SD10356, SD10357 | Simple PLC Communication execution state | The data transmission/reception status of the simple CPU communication is stored for each setting number. SD10356[b0] to [b15]: Setting No.1 to Setting No.16 SD10357[b0] to [b15]: Setting No.17 to Setting No.32*1 0: Communication stop (function not used) 1: Communicating | R |
| SD10358, SD10359 | Simple PLC Communication ready flag | The preparation completion status of the simple CPU communication is stored for each setting number. SD10358[b0] to [b15]: Setting No.1 to Setting No.16 SD10359[b0] to [b15]: Setting No.17 to Setting No.32*1 0: Not ready (function not used) 1: Ready | R |
| SD10380 to SD10411 | Simple CPU communication status | The simple CPU communication status is stored. SD10380: Setting No.1 to SD10395: Setting No.16 SD10396: Setting No.17 to SD10411: Setting No.32*1 0H: Unset 1H: Preparing 3H: Communicating 4H: Communication stop 5H: Retry being executed 6H: Monitoring AH: Communications impossible | R |

| Device No. | Name | Description | R/W |
|----------------------------------|---|--|-----|
| SD10412 to SD10443 | Simple PLC Communication error code | The cause of the error detected in the simple CPU communication is stored. SD10412: Setting No.1 to SD10427: Setting No.16 SD10428: Setting No.17 to SD10443: Setting No.32 ^{*1} 0: No error (function not used) CFB4H: An abnormal response was received when the communication target is other company's PLC, a MODBUS/TCP-compatible device or MELSEC FX3 (Ethernet Block/Adapter). Other than above: Error code The value is cleared to 0 with a clear request from the engineering tool. | R |
| SD10444 to SD10475 | Simple PLC Communication execution interval | If "Fixed" is set for communication setting, the current value of the execution interval is stored. SD10444: Setting No.1 to SD10459: Setting No.16 SD10460: Setting No.17 to SD10475: Setting No.32 ^{*1} 0: Unset (function not used), communications impossible Other than 0: Execution interval (unit: ms) | R |
| SD10476 to SD10507 ^{*1} | Simple CPU communication error response code | If the communication target during the simple CPU communication is other company's PLC, a MODBUS/TCP-compatible device or MELSEC-FX3 (Ethernet Block/Adapter), error factors detected will be stored. SD10476: Setting No.1 to SD10591: Setting No.16 SD10492: Setting No.17 to SD10507: Setting No.32 0: when there is no error or the communication target is a MELSEC PLC or a SLMP-compatible device. Other than 0: Error codes (refer to individual devices' manuals) | R |
| SD10680 | Open completion signal | Open completion signal for each connection. [b0] to [b7]: Connection No. 1 to No. 8 0: Closed or not open 1: Open completed | R |
| SD10681 | Open request signal | Open request signal for each connection. [b0] to [b7]: Connection No. 1 to No. 8 0: No open request 1: Requesting open | R |
| SD10682 | Socket communications receive status signal | Socket communication receive state signal for each connection. [b0] to [b7]: Connection No. 1 to No. 8 0: Data not received 1: Data reception completed | R |
| SD10683 | Initial status | Stores the status of the initial processing. Initial normal completion status (b0) 0: — 1: Initialization normal completion Initial abnormal completion status (b1) 0: — 1: Initialization abnormal completion | R |
| SD10692 | Predefined protocol ready | Stores the ready status of the protocol setting data. 0: — 1: Ready | R |
| SD10710 | Predefined protocol setting data check area protocol number | When a protocol setting data error is detected, stores the protocol number where the error was detected. Protocol is checked in order from smallest protocol number. The protocol number where an error was detected first is stored. 0: No error 1 to 64: Protocol number 65535: Cannot identify ^{*2} | R |
| SD10711 | Predefined protocol setting data check area setting type | 0 is stored if an error is detected in the packet setting or element setting. 1 is stored if an error is detected in the protocol detailed setting. (Valid when protocol number value is 1 to 128) 0: Packet setting or element setting 1: Protocol preferences 65535: Cannot identify ^{*2} | R |
| SD10712 | Predefined protocol setting data check area packet number | When an error is detected in the protocol setting data, stores the packet number that detected the error. The packets are checked in order of send packets and then receive packets (expected packets) from smallest number. The packet number where an error was detected first is stored. (Valid when setting type value is 0) 0: Send packet 1 to 16: Receive packet number 65535: Cannot identify ^{*2} | R |



| Device No. | Name | Description | R/W |
|------------|---|--|-----|
| SD10713 | Predefined protocol setting data check area protocol number | When an error is detected in the protocol setting data, stores the element number where the error was detected. The elements are checked in order of smallest element number. The element number where an error was detected first is stored. (Valid when setting type value is 0) 1 to 32: Element number 65535: Cannot identify*2 | R |
| SD10714 | Number of registered predefined protocols | Stores the protocol number of the registered protocol setting data. 0 is stored if the protocol setting data check result is abnormal. 0: No registration 1 to 64: Number of registrations | R |
| SD10722 | Predefined protocol registration (Protocol numbers 1 to 16) | Whether protocol setting data is registered or not is stored. All bits are set to 0 if the protocol setting data check result is abnormal. [b0] to [b15]: Protocol numbers 1 to 16 0: No registration 1: Registered | R |
| SD10723 | Predefined protocol registration (Protocol numbers 17 to 32) | Whether protocol setting data is registered or not is stored. All bits are set to 0 if the protocol setting data check result is abnormal. [b0] to [b15]: Protocol numbers 17 to 32 0: No registration 1: Registered | R |
| SD10724 | Predefined protocol registration (Protocol numbers 33 to 48) | Whether protocol setting data is registered or not is stored. All bits are set to 0 if the protocol setting data check result is abnormal. [b0] to [b15]: Protocol numbers 33 to 48 0: No registration 1: Registered | R |
| SD10725 | Predefined protocol registration (Protocol numbers 49 to 64) | Whether protocol setting data is registered or not is stored. All bits are set to 0 if the protocol setting data check result is abnormal. [b0] to [b15]: Protocol numbers 49 to 64 0: No registration 1: Registered | R |
| SD10740 | Connection No.1 protocol execution status | Stores the status of the protocol being executed at connection No.1. 0: Unexecuted 1: Waiting for transmission 2: Sending 3: Waiting for data reception 4: Receiving 5: Execution completed | R |
| SD10742 | Connection No.1 received data verification result (receive packet No.1) | Stores the verification results of receive packet No.1. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10743 | Connection No.1 received data verification result (receive packet No.2) | Stores the verification results of receive packet No.2. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10744 | Connection No.1 received data verification result (receive packet No.3) | Stores the verification results of receive packet No.3. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10745 | Connection No.1 received data verification result (receive packet No.4) | Stores the verification results of receive packet No.4. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10746 | Connection No.1 received data verification result (receive packet No.5) | Stores the verification results of receive packet No.5. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10747 | Connection No.1 received data verification result (receive packet No.6) | Stores the verification results of receive packet No.6. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10748 | Connection No.1 received data verification result (receive packet No.7) | Stores the verification results of receive packet No.7. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10749 | Connection No.1 received data verification result (receive packet No.8) | Stores the verification results of receive packet No.8. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10750 | Connection No.1 received data verification result (receive packet No.9) | Stores the verification results of receive packet No.9. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |

| Device No. | Name | Description | R/W |
|------------|--|---|-----|
| SD10751 | Connection No.1 received data verification result (receive packet No.10) | Stores the verification results of receive packet No.10. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10752 | Connection No.1 received data verification result (receive packet No.11) | Stores the verification results of receive packet No.11. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10753 | Connection No.1 received data verification result (receive packet No.12) | Stores the verification results of receive packet No.12. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10754 | Connection No.1 received data verification result (receive packet No.13) | Stores the verification results of receive packet No.13. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10755 | Connection No.1 received data verification result (receive packet No.14) | Stores the verification results of receive packet No.14. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10756 | Connection No.1 received data verification result (receive packet No.15) | Stores the verification results of receive packet No.15. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10757 | Connection No.1 received data verification result (receive packet No.16) | Stores the verification results of receive packet No.16. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10758 | Connection No.1 protocol execution count | Stores the number of protocol executions in Connection No.1. 0: Protocol not executed 1 to 65535: Number of executions | R |
| SD10759 | Connection No.1 protocol cancellation specification | Cancels the protocol executed in connection No.1. 0: No cancellation instruction 1: Cancellation request (set by user) 2: Cancellation completed (set by system) | R/W |
| SD10760 | Connection No.2 protocol execution status | Stores the status of the protocol being executed at connection No.2. 0: Unexecuted 1: Waiting for transmission 2: Sending 3: Waiting for data reception 4: Receiving 5: Execution completed | R |
| SD10762 | Connection No.2 received data verification result (receive packet No.1) | Stores the verification results of receive packet No.1. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10763 | Connection No.2 received data verification result (receive packet No.2) | Stores the verification results of receive packet No.2. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10764 | Connection No.2 received data verification result (receive packet No.3) | Stores the verification results of receive packet No.3. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10765 | Connection No.2 received data verification result (receive packet No.4) | Stores the verification results of receive packet No.4. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10766 | Connection No.2 received data verification result (receive packet No.5) | Stores the verification results of receive packet No.5. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10767 | Connection No.2 received data verification result (receive packet No.6) | Stores the verification results of receive packet No.6. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10768 | Connection No.2 received data verification result (receive packet No.7) | Stores the verification results of receive packet No.7. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10769 | Connection No.2 received data verification result (receive packet No.8) | Stores the verification results of receive packet No.8. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10770 | Connection No.2 received data verification result (receive packet No.9) | Stores the verification results of receive packet No.9. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |

| Device No. | Name | Description | R/W |
|------------|--|---|-----|
| SD10771 | Connection No.2 received data verification result (receive packet No.10) | Stores the verification results of receive packet No.10. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10772 | Connection No.2 received data verification result (receive packet No.11) | Stores the verification results of receive packet No.11. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10773 | Connection No.2 received data verification result (receive packet No.12) | Stores the verification results of receive packet No.12. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10774 | Connection No.2 received data verification result (receive packet No.13) | Stores the verification results of receive packet No.13. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10775 | Connection No.2 received data verification result (receive packet No.14) | Stores the verification results of receive packet No.14. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10776 | Connection No.2 received data verification result (receive packet No.15) | Stores the verification results of receive packet No.15. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10777 | Connection No.2 received data verification result (receive packet No.16) | Stores the verification results of receive packet No.16. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10778 | Connection No.2 protocol execution count | Stores the number of protocol executions in connection No.2. 0: Protocol not executed 1 to 65535: Number of executions | R |
| SD10779 | Connection No.2 protocol cancellation specification | Cancels the protocol executed in connection No.2. 0: No cancellation instruction 1: Cancellation request (set by user) 2: Cancellation completed (set by system) | R/W |
| SD10780 | Connection No.3 protocol execution status | Stores the status of the protocol being executed at connection No.3. 0: Unexecuted 1: Waiting for transmission 2: Sending 3: Waiting for data reception 4: Receiving 5: Execution completed | R |
| SD10782 | Connection No.3 received data verification result (receive packet No.1) | Stores the verification results of receive packet No.1. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10783 | Connection No.3 received data verification result (receive packet No.2) | Stores the verification results of receive packet No.2. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10784 | Connection No.3 received data verification result (receive packet No.3) | Stores the verification results of receive packet No.3. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10785 | Connection No.3 received data verification result (receive packet No.4) | Stores the verification results of receive packet No.4. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10786 | Connection No.3 received data verification result (receive packet No.5) | Stores the verification results of receive packet No.5. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10787 | Connection No.3 received data verification result (receive packet No.6) | Stores the verification results of receive packet No.6. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10788 | Connection No.3 received data verification result (receive packet No.7) | Stores the verification results of receive packet No.7. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10789 | Connection No.3 received data verification result (receive packet No.8) | Stores the verification results of receive packet No.8. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10790 | Connection No.3 received data verification result (receive packet No.9) | Stores the verification results of receive packet No.9. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |

| Device No. | Name | Description | R/W |
|------------|--|---|-----|
| SD10791 | Connection No.3 received data verification result (receive packet No.10) | Stores the verification results of receive packet No.10. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10792 | Connection No.3 received data verification result (receive packet No.11) | Stores the verification results of receive packet No.11. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10793 | Connection No.3 received data verification result (receive packet No.12) | Stores the verification results of receive packet No.12. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10794 | Connection No.3 received data verification result (receive packet No.13) | Stores the verification results of receive packet No.13. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10795 | Connection No.3 received data verification result (receive packet No.14) | Stores the verification results of receive packet No.14. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10796 | Connection No.3 received data verification result (receive packet No.15) | Stores the verification results of receive packet No.15. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10797 | Connection No.3 received data verification result (receive packet No.16) | Stores the verification results of receive packet No.16. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10798 | Connection No.3 protocol execution count | Stores the number of protocol executions in connection No.3. 0: Protocol not executed 1 to 65535: Number of executions | R |
| SD10799 | Connection No.3 protocol cancellation specification | Cancels the protocol executed in connection No.3. 0: No cancellation instruction 1: Cancellation request (set by user) 2: Cancellation completed (set by system) | R/W |
| SD10800 | Connection No.4 protocol execution status | Stores the status of the protocol being executed at connection No.4. 0: Unexecuted 1: Waiting for transmission 2: Sending 3: Waiting for data reception 4: Receiving 5: Execution completed | R |
| SD10802 | Connection No.4 received data verification result (receive packet No.1) | Stores the verification results of receive packet No.1. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10803 | Connection No.4 received data verification result (receive packet No.2) | Stores the verification results of receive packet No.2. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10804 | Connection No.4 received data verification result (receive packet No.3) | Stores the verification results of receive packet No.3. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10805 | Connection No.4 received data verification result (receive packet No.4) | Stores the verification results of receive packet No.4. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10806 | Connection No.4 received data verification result (receive packet No.5) | Stores the verification results of receive packet No.5. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10807 | Connection No.4 received data verification result (receive packet No.6) | Stores the verification results of receive packet No.6. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10808 | Connection No.4 received data verification result (receive packet No.7) | Stores the verification results of receive packet No.7. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10809 | Connection No.4 received data verification result (receive packet No.8) | Stores the verification results of receive packet No.8. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10810 | Connection No.4 received data verification result (receive packet No.9) | Stores the verification results of receive packet No.9. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |

| Device No. | Name | Description | R/W |
|------------|--|---|-----|
| SD10811 | Connection No.4 received data verification result (receive packet No.10) | Stores the verification results of receive packet No.10. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10812 | Connection No.4 received data verification result (receive packet No.11) | Stores the verification results of receive packet No.11. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10813 | Connection No.4 received data verification result (receive packet No.12) | Stores the verification results of receive packet No.12. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10814 | Connection No.4 received data verification result (receive packet No.13) | Stores the verification results of receive packet No.13. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10815 | Connection No.4 received data verification result (receive packet No.14) | Stores the verification results of receive packet No.14. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10816 | Connection No.4 received data verification result (receive packet No.15) | Stores the verification results of receive packet No.15. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10817 | Connection No.4 received data verification result (receive packet No.16) | Stores the verification results of receive packet No.16. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10818 | Connection No.4 protocol execution count | Stores the number of protocol executions in connection No.4. 0: Protocol not executed 1 to 65535: Number of executions | R |
| SD10819 | Connection No.4 protocol cancellation specification | Cancels the protocol executed in connection No.4. 0: No cancellation instruction 1: Cancellation request (set by user) 2: Cancellation completed (set by system) | R/W |
| SD10820 | Connection No.5 protocol execution status | Stores the status of the protocol being executed at connection No.5. 0: Unexecuted 1: Waiting for transmission 2: Sending 3: Waiting for data reception 4: Receiving 5: Execution completed | R |
| SD10822 | Connection No.5 received data verification result (receive packet No.1) | Stores the verification results of receive packet No.1. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10823 | Connection No.5 received data verification result (receive packet No.2) | Stores the verification results of receive packet No.2. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10824 | Connection No.5 received data verification result (receive packet No.3) | Stores the verification results of receive packet No.3. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10825 | Connection No.5 received data verification result (receive packet No.4) | Stores the verification results of receive packet No.4. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10826 | Connection No.5 received data verification result (receive packet No.5) | Stores the verification results of receive packet No.5. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10827 | Connection No.5 received data verification result (receive packet No.6) | Stores the verification results of receive packet No.6. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10828 | Connection No.5 received data verification result (receive packet No.7) | Stores the verification results of receive packet No.7. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10829 | Connection No.5 received data verification result (receive packet No.8) | Stores the verification results of receive packet No.8. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10830 | Connection No.5 received data verification result (receive packet No.9) | Stores the verification results of receive packet No.9. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |

| Device No. | Name | Description | R/W |
|------------|--|---|-----|
| SD10831 | Connection No.5 received data verification result (receive packet No.10) | Stores the verification results of receive packet No.10. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10832 | Connection No.5 received data verification result (receive packet No.11) | Stores the verification results of receive packet No.11. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10833 | Connection No.5 received data verification result (receive packet No.12) | Stores the verification results of receive packet No.12. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10834 | Connection No.5 received data verification result (receive packet No.13) | Stores the verification results of receive packet No.13. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10835 | Connection No.5 received data verification result (receive packet No.14) | Stores the verification results of receive packet No.14. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10836 | Connection No.5 received data verification result (receive packet No.15) | Stores the verification results of receive packet No.15. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10837 | Connection No.5 received data verification result (receive packet No.16) | Stores the verification results of receive packet No.16. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10838 | Connection No.5 protocol execution count | Stores the number of protocol executions in connection No.5. 0: Protocol not executed 1 to 65535: Number of executions | R |
| SD10839 | Connection No.5 protocol cancellation specification | Cancels the protocol executed in connection No.5. 0: No cancellation instruction 1: Cancellation request (set by user) 2: Cancellation completed (set by system) | R/W |
| SD10840 | Connection No.6 protocol execution status | Stores the status of the protocol being executed at connection No.6. 0: Unexecuted 1: Waiting for transmission 2: Sending 3: Waiting for data reception 4: Receiving 5: Execution completed | R |
| SD10842 | Connection No.6 received data verification result (receive packet No.1) | Stores the verification results of receive packet No.1. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10843 | Connection No.6 received data verification result (receive packet No.2) | Stores the verification results of receive packet No.2. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10844 | Connection No.6 received data verification result (receive packet No.3) | Stores the verification results of receive packet No.3. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10845 | Connection No.6 received data verification result (receive packet No.4) | Stores the verification results of receive packet No.4. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10846 | Connection No.6 received data verification result (receive packet No.5) | Stores the verification results of receive packet No.5. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10847 | Connection No.6 received data verification result (receive packet No.6) | Stores the verification results of receive packet No.6. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10848 | Connection No.6 received data verification result (receive packet No.7) | Stores the verification results of receive packet No.7. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10849 | Connection No.6 received data verification result (receive packet No.8) | Stores the verification results of receive packet No.8. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10850 | Connection No.6 received data verification result (receive packet No.9) | Stores the verification results of receive packet No.9. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |



| Device No. | Name | Description | R/W |
|------------|--|---|-----|
| SD10851 | Connection No.6 received data verification result (receive packet No.10) | Stores the verification results of receive packet No.10. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10852 | Connection No.6 received data verification result (receive packet No.11) | Stores the verification results of receive packet No.11. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10853 | Connection No.6 received data verification result (receive packet No.12) | Stores the verification results of receive packet No.12. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10854 | Connection No.6 received data verification result (receive packet No.13) | Stores the verification results of receive packet No.13. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10855 | Connection No.6 received data verification result (receive packet No.14) | Stores the verification results of receive packet No.14. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10856 | Connection No.6 received data verification result (receive packet No.15) | Stores the verification results of receive packet No.15. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10857 | Connection No.6 received data verification result (receive packet No.16) | Stores the verification results of receive packet No.16. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10858 | Connection No.6 protocol execution count | Stores the number of protocol executions in connection No.6. 0: Protocol not executed 1 to 65535: Number of executions | R |
| SD10859 | Connection No.6 protocol cancellation specification | Cancels the protocol executed in connection No.6. 0: No cancellation instruction 1: Cancellation request (set by user) 2: Cancellation completed (set by system) | R/W |
| SD10860 | Connection No.7 protocol execution status | Stores the status of the protocol being executed at connection No.7. 0: Unexecuted 1: Waiting for transmission 2: Sending 3: Waiting for data reception 4: Receiving 5: Execution completed | R |
| SD10862 | Connection No.7 received data verification result (receive packet No.1) | Stores the verification results of receive packet No.1. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10863 | Connection No.7 received data verification result (receive packet No.2) | Stores the verification results of receive packet No.2. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10864 | Connection No.7 received data verification result (receive packet No.3) | Stores the verification results of receive packet No.3. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10865 | Connection No.7 received data verification result (receive packet No.4) | Stores the verification results of receive packet No.4. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10866 | Connection No.7 received data verification result (receive packet No.5) | Stores the verification results of receive packet No.5. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10867 | Connection No.7 received data verification result (receive packet No.6) | Stores the verification results of receive packet No.6. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10868 | Connection No.7 received data verification result (receive packet No.7) | Stores the verification results of receive packet No.7. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10869 | Connection No.7 received data verification result (receive packet No.8) | Stores the verification results of receive packet No.8. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10870 | Connection No.7 received data verification result (receive packet No.9) | Stores the verification results of receive packet No.9. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |

| Device No. | Name | Description | R/W |
|------------|--|---|-----|
| SD10871 | Connection No.7 received data verification result (receive packet No.10) | Stores the verification results of receive packet No.10. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10872 | Connection No.7 received data verification result (receive packet No.11) | Stores the verification results of receive packet No.11. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10873 | Connection No.7 received data verification result (receive packet No.12) | Stores the verification results of receive packet No.12. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10874 | Connection No.7 received data verification result (receive packet No.13) | Stores the verification results of receive packet No.13. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10875 | Connection No.7 received data verification result (receive packet No.14) | Stores the verification results of receive packet No.14. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10876 | Connection No.7 received data verification result (receive packet No.15) | Stores the verification results of receive packet No.15. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10877 | Connection No.7 received data verification result (receive packet No.16) | Stores the verification results of receive packet No.16. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10878 | Connection No.7 protocol execution count | Stores the number of protocol executions in connection No.7. 0: Protocol not executed 1 to 65535: Number of executions | R |
| SD10879 | Connection No.7 protocol cancellation specification | Cancels the protocol executed in connection No.7. 0: No cancellation instruction 1: Cancellation request (set by user) 2: Cancellation completed (set by system) | R/W |
| SD10880 | Connection No.8 protocol execution status | Stores the status of the protocol being executed at connection No.8. 0: Unexecuted 1: Waiting for transmission 2: Sending 3: Waiting for data reception 4: Receiving 5: Execution completed | R |
| SD10882 | Connection No.8 received data verification result (receive packet No.1) | Stores the verification results of receive packet No.1. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10883 | Connection No.8 received data verification result (receive packet No.2) | Stores the verification results of receive packet No.2. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10884 | Connection No.8 received data verification result (receive packet No.3) | Stores the verification results of receive packet No.3. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10885 | Connection No.8 received data verification result (receive packet No.4) | Stores the verification results of receive packet No.4. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10886 | Connection No.8 received data verification result (receive packet No.5) | Stores the verification results of receive packet No.5. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10887 | Connection No.8 received data verification result (receive packet No.6) | Stores the verification results of receive packet No.6. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10888 | Connection No.8 received data verification result (receive packet No.7) | Stores the verification results of receive packet No.7. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10889 | Connection No.8 received data verification result (receive packet No.8) | Stores the verification results of receive packet No.8. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10890 | Connection No.8 received data verification result (receive packet No.9) | Stores the verification results of receive packet No.9. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |

| Device No. | Name | Description | R/W |
|------------|--|--|-----|
| SD10891 | Connection No.8 received data verification result (receive packet No.10) | Stores the verification results of receive packet No.10. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10892 | Connection No.8 received data verification result (receive packet No.11) | Stores the verification results of receive packet No.11. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10893 | Connection No.8 received data verification result (receive packet No.12) | Stores the verification results of receive packet No.12. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10894 | Connection No.8 received data verification result (receive packet No.13) | Stores the verification results of receive packet No.13. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10895 | Connection No.8 received data verification result (receive packet No.14) | Stores the verification results of receive packet No.14. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10896 | Connection No.8 received data verification result (receive packet No.15) | Stores the verification results of receive packet No.15. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10897 | Connection No.8 received data verification result (receive packet No.16) | Stores the verification results of receive packet No.16. Element No. where the verification result did not match (b0 to b7) The cause of mismatch (verification result code) (b8 to b15) | R |
| SD10898 | Connection No.8 protocol execution count | Stores the number of protocol executions in connection No.8. 0: Protocol not executed 1 to 65535: Number of executions | R |
| SD10899 | Connection No.8 protocol cancellation specification | Cancels the protocol executed in connection No.8. 0: No cancellation instruction 1: Cancellation request (set by user) 2: Cancellation completed (set by system) | R/W |

R: Read only, R/W: Read/Write

*1 Only FX5U/FX5UC CPU module is supported.

*2 The setting value may be unidentifiable (65535) in the following cases.

- When a setting that cannot be detected by the current CPU module version is written
- When protocol setting data is broken (hardware failure)

Appendix 2 List of Buffer Memory Applications and Assignments

The buffer memory is used to exchange data between the Ethernet module and the CPU module or external devices. Buffer memory values are set to their defaults (initial values) when the system is powered off or the CPU module is reset.

For the details of the listed buffer memories, as well as the buffer memories not described below, refer to following manuals.



📖 MELSEC iQ-F FX5-ENET User's Manual


📖 MELSEC iQ-F FX5-ENET/IP User's Manual

📖 MELSEC iQ-F FX5 User's Manual (BACnet)

| Buffer memory No. | | Name | Description | R/W |
|-------------------|-------------|--|---|-----|
| Decimal | Hexadecimal | | | |
| 29 | 1DH | Latest error code | The latest error code that has occurred in Ethernet module is stored. For error code, refer to 📖 Page 245 Error Codes. | R |
| 30 | 1EH | Module information | This area stores the module information of Ethernet module. • 69A0H: FX5-ENET • 69C1H: FX5-ENET/IP | R |
| 31 | 1FH | Firmware version | This area stores the firmware version of Ethernet module. | R |
| 50 | 32H | IP address setting (Low-order) | • Stores IP address to be set when using IP address (low-order) change function. • Becomes 0 when writing to IP address storage area is completed normally. | R/W |
| 51 | 33H | IP address setting (High-order) | • Stores IP address to be set when using IP address (high-order) change function. • Becomes 0 when writing to IP address storage area is completed normally. | R/W |
| 52 | 34H | Subnet mask setting (Low-order) | • Stores subnet mask (low-order) to be set when using IP address change function. • Becomes 0 when writing to IP address storage area is completed normally. | R/W |
| 53 | 35H | Subnet mask setting (High-order) | • Stores subnet mask (high-order) to be set when using IP address change function. • Becomes 0 when writing to IP address storage area is completed normally. | R/W |
| 54 | 36H | Default router IP address setting (Low-order) | • Stores default router IP address (low-order) to be set when using IP address change function. • Becomes 0 when writing to IP address storage area is completed normally. | R/W |
| 55 | 37H | Default router IP address setting (High-order) | • Stores default router IP address (high-order) to be set when using IP address change function. • Becomes 0 when writing to IP address storage area is completed normally. | R/W |
| 56 | 38H | IP address storage area write request | Specify whether to write the stored values of IP address setting (Un\G50 to Un\G51), Subnet mask pattern setting (Un\G52 to Un\G53) and Default router IP address setting (Un\G54 to Un\G55) to the IP address storage area. 0: Not write 1: Write | R/W |
| 57 | 39H | IP address storage area write status | You can confirm whether or not the values are written to the IP address storage area when executing the IP address change function. [b0]: IP address storage area write completed 0: — 1: Write completed Becomes 1 also in the case of abnormal completion. [b1]: IP address storage area write error 0: — 1: Write fails Becomes 1 also if the data in the IP address storage area is abnormal when the Ethernet module power is turned off and on. | R |
| 58 | 3AH | IP address storage area clear request | Specify whether to clear the data in the IP address storage area. 0: Not clear 1: Clear | R/W |
| 59 | 3BH | IP address storage area clear status | You can confirm whether or not the IP address storage area is cleared. [b0]: IP address storage area clear completed 0: — 1: Clear completed Becomes 1 also in the case of abnormal completion. [b1]: IP address storage area clear error 0: — 1: Clear fails | R |



| Buffer memory No. | | Name | Description | R/W |
|-------------------|-------------|---|---|-----|
| Decimal | Hexadecimal | | | |
| 60 | 3CH | IP address change function enable flag | You can confirm whether or not the IP address change function is enabled. 0: Invalid 1: Enable | R |
| 61 | 3DH | IP address storage area write error code | Stores error codes if writing to IP address storage area fails. 0: Normal (no error) 1920H: Values of IP address setting, etc. (Un\G50 to Un\G55) are out of the setting ranges. | R |
| 62 | 3EH | IP address storage area clear error code | Stores error codes if clearing of IP address storage area fails. 0: Normal (no error) 1921H: IP address storage area write request (Un\G56) and IP address storage area clear request (Un\G58) were simultaneously turned off and on. | R |
| 64 | 40H | IP address (Low-order) | Stores IP address (low-order) on the own station set with engineering tool. | R |
| 65 | 41H | IP address (High-order) | Stores IP address (high-order) on the own station set with engineering tool. | R |
| 74 | 4AH | Subnet mask pattern (Low-order) | Stores subnet mask pattern (low-order) on the own station set with engineering tool. | R |
| 75 | 4BH | Subnet mask pattern (High-order) | Stores subnet mask pattern (high-order) on the own station set with engineering tool. | R |
| 76 | 4CH | Default gateway IP address (Low-order) | Stores default gateway IP address (low-order) on the own station set with engineering tool. | R |
| 77 | 4DH | Default gateway IP address (High-order) | Stores default gateway IP address (high-order) on the own station set with engineering tool. | R |
| 102 to 104 | 66H to 68H | Ethernet address (MAC address) | Stores Ethernet address (MAC address) on the own station. Un\G102: Serial ID Un\G103: Lower one digit of vendor ID, model ID Un\G104: Upper two digits of vendor ID | R |
| 108 to 139 | 6CH to 8BH | Error code | Stores error code of Ethernet. Un\G108: Connection No.1 to Un\G123: Connection No.16 Un\G124: Connection No.1 to Un\G139: Connection No.16 For details on error code, refer to  Page 245 Error Codes. | R |
| 152, 153 | 98H, 99H | Open completion signal | Open completion signal for each connection number. Un\G152[0] to [15]: Connection No.1 to Connection No.16 Un\G153[0] to [15]: Connection No.17 to Connection No.32 0: Closed or not open 1: Open completed | R |
| 154, 155 | 9AH, 9BH | Open request signal | Open request signal for each connection number of socket communication. Un\G154[0] to [15]: Connection No.1 to Connection No.16 Un\G155[0] to [15]: Connection No.17 to Connection No.32 0: No open request 1: Requesting open | R |
| 156, 157 | 9CH, 9DH | Socket communications receive status signal | Socket communication receive state signal for each connection number. Un\G156[0] to [15]: Connection No.1 to Connection No.16 Un\G157[0] to [15]: Connection No.17 to Connection No.32 0: Data not received 1: Data reception completed | R |
| 158 | 9EH | Initial status | Stores the status of the initial processing. [b0]: Initial normal completion status 0: — 1: Initial normal completion [b1]: Initial abnormal completion status 0: — 1: Initial abnormal completion | R |
| 159 | 9FH | Initial error code | The error codes that occur during initialization are stored. For details on error code, refer to  Page 245 Error Codes. | R |
| 201 | C9H | Same IP address state storage area | Same IP address state is stored. [b0]: Same IP address detection flag 0: No same IP address 1: Same IP address | R |

| Buffer memory No. | | Name | Description | R/W |
|-------------------|--------------|--|---|-----|
| Decimal | Hexadecimal | | | |
| 202 to 204 | CAH to CCH | MAC address of the already connected station | Stores the MAC address of the station, which was connected to the network earlier, in the station with duplicated IP address. Un\G202: Serial ID Un\G203: Lower one digit of vendor ID, model ID Un\G204: Upper two digits of vendor ID "FFFFFFFFFFFFH" is stored in the station that has been already connected to the network. | R |
| 205 to 207 | CDH to CFH | MAC address of the station connected later | Stores the MAC address of the station with duplicated IP address in the station which was connected earlier to the network. Un\G205: Serial ID Un\G206: Lower one digit of vendor ID, model ID Un\G207: Upper two digits of vendor ID "FFFFFFFFFFFFH" is stored in the station with duplicated IP address. | R |
| 300, 301 | 12CH, 12DH | "Communication start at request" request | When the communication setting is "requested," start of data transmission in the simple CPU communication is requested. Un\G300[0] to [15]: Setting No.1 to setting No.16 Un\G301[0] to [15]: Setting No.17 to setting No.32 0: Not requested 1: Requested | R/W |
| 304, 305 | 130H, 131H | Periodic communication stop request | When the communication setting is "requested," start of data transmission in the simple CPU communication is requested. Un\G304[0] to [15]: Setting No.1 to setting No.16 Un\G305[0] to [15]: Setting No.17 to setting No.32 0: Not requested 1: Requested | R/W |
| 308, 309 | 134H, 135H | Periodic communication restart request | When the communication setting is "requested," start of data transmission in the simple CPU communication is requested. Un\G308[0] to [15]: Setting No.1 to setting No.16 Un\G309[0] to [15]: Setting No.17 to setting No.32 0: Not requested 1: Requested | R/W |
| 312, 313 | 138H, 139H | Execution Status flag | The data transmission/reception status of the simple CPU communication is stored. Un\G312[0] to [15]: Setting No.1 to setting No.16 Un\G313[0] to [15]: Setting No.17 to setting No.32 0: Unexecuted 1: Communicating | R |
| 316, 317 | 13CH, 13DH | Ready | The preparation completion status of the simple CPU communication is stored. Un\G316[0] to [15]: Setting No.1 to setting No.16 Un\G317[0] to [15]: Setting No.17 to setting No.32 0: Not ready 1: Ready | R |
| 320 to 351 | 140H to 15FH | System area | — | — |
| 352 to 383 | 160H to 17FH | Simple CPU communication status | The simple CPU communication status is stored. Un\G352: Setting No.1 to Un\G367: Setting No.16 Un\G368: Setting No.17 to Un\G383: Setting No.32 0H: Unset 1H: Preparing 2H: Waiting for request 3H: Communicating 4H: Communication stop 5H: Retry being executed 6H: Monitoring AH: Communications impossible | R |
| 416 to 447 | 1A0H to 1BFH | Simple CPU error code | The cause of the error detected in the simple CPU communication is stored. Un\G416: Connection No.1 to Un\G431: Connection No.16 Un\G432: Connection No.17 to Un\G447: Connection No.32 For the details of the error codes, refer to  Page 247 Ethernet module. | R |
| 480 to 511 | 1E0H to 1FFH | Abnormal response code | The abnormal response code detected in the simple CPU communication is stored. Un\G480: Connection No.1 to Un\G495: Connection No.16 Un\G496: Connection No.17 to Un\G511: Connection No.32 | R |
| 544 to 575 | 220H to 23FH | Execution interval (current value) | The execution interval of the simple CPU communication is stored. Un\G544: Setting No.1 to Un\G559: Setting No.16 Un\G560: Setting No.17 to Un\G575: Setting No.32 | R |

| Buffer memory No. | | Name | Description | R/W |
|-------------------|----------------|-----------------------------------|---|-----|
| Decimal | Hexadecimal | | | |
| 6400 to 8447 | 1900H to 20FFH | Area for simple CPU communication | Usable as a device that can be specified for the own station in the simple CPU communication. (Word device in units of 1 point) | R/W |

R: Read only, R/W: Read/Write

Appendix 3 Processing Time

List of performance of simple CPU communication function

The performance of the execution interval of the simple CPU communication function is shown below. The execution interval varies depending on the number of settings, number of communication points and CPU module scan time. In addition, it may vary depending on other factors, such as execution of other functions and Ethernet communication status.

Condition

- Communication setting: “Fixed”
- Own station sequence scan time =: 1 ms (scan time before execution of simple CPU communication function)
- Communication destination: MELSEC iQ-F (built-in Ethernet)
- Communication destination sequence scan time: 1 ms (for any models)
- Device data: Bit device = M Word device = D
- The number of settings is as shown in the table (1, 8, 16, 32), and all must be different communication destinations (not the same communication destination).
- Communication retry has not occurred.

List of communication performance (MELSEC iQ-F (built-in Ethernet))

| Communication pattern | Number of communication points per setting | Processing time depending on number of settings (unit: ms) | | | |
|-----------------------|---|--|-----------------|------------------|------------------|
| | | 1 | 8 ^{*2} | 16 ^{*2} | 32 ^{*2} |
| Read | 32 words each ^{*1} (64 words in all) | 10 | 12 | 26 | 51 |
| | 64 words each ^{*1} (128 words in all) | 10 | 13 | 27 | 56 |
| | 256 words each ^{*1} (512 words in all) | 11 | 22 | 38 | — |
| Write | 32 words each ^{*1} (64 words in all) | 10 | 12 | 25 | 53 |
| | 64 words each ^{*1} (128 words in all) | 10 | 12 | 26 | 63 |
| | 256 words each ^{*1} (512 words in all) | 11 | 18 | 35 | — |

*1 Number of points of each of bit device and word device

*2 The larger the number of settings of the same destination, the longer the execution interval, up to about eight times longer.

Appendix 4 Added and Changed Functions

The functions added or changed with the Ethernet-equipped module and engineering tool, and the supported Ethernet-equipped modules' firmware version and engineering tool software version are given below.

The firmware version of the CPU module can be confirmed by module diagnosis (CPU diagnosis). Refer to the following manuals for details on diagnosing the module (CPU diagnosis).

📖 MELSEC iQ-F FX5UJ User's Manual (Hardware)

📖 MELSEC iQ-F FX5U User's Manual (Hardware)

📖 MELSEC iQ-F FX5UC User's Manual (Hardware)

The firmware version of the Ethernet module can be confirmed in the buffer memory (Un\G30).

Refer to the 📖 GX Works3 Operating Manual for details on the software version.

FX5UJ CPU module

| Add/Change Function | Supported CPU module firmware version | Supported engineering tool software version | Reference |
|--------------------------------|---------------------------------------|---|-----------|
| FX5UJ CPU module is supported. | From the first | "1.060N" and above | — |

FX5U/FX5UC CPU module

| Add/Change Function | Supported CPU module firmware version | Supported engineering tool software version | Reference |
|---|---------------------------------------|---|--|
| The file transfer function is supported. | "1.040" and above*1*2 | "1.030G" and above*3 | Page 137 |
| <ul style="list-style-type: none"> • Automatic detection of connected devices • Communication setting reflection of Ethernet device • Sensor parameter read/write | "1.040" and above | "1.030G" and above | iQ Sensor Solution Reference Manual |
| IP filter function is supported. | "1.050" and above | "1.035M" and above | Page 192 |
| MODBUS/TCP communication is supported. | "1.060" and above | "1.040S" and above | MELSEC iQ-F FX5 User's Manual (MODBUS Communication) |
| Time setting function (SNTP client) is supported. | "1.060" and above | "1.040S" and above | Page 162 |
| Web server function is supported. | "1.060" and above | "1.040S" and above | Page 165 |
| During IP address duplication with the device on the same network, the operation was improved to output the information of the external device with duplicated IP address. | "1.061" and above | — | Page 241 |
| User Web page is supported. | "1.100" and above | "1.047Z" and above | Page 165 |
| Simple CPU communication function is supported. | "1.110" and above | "1.050C" and above | Page 201 |
| FX5-ENET is supported. | "1.110" and above | "1.050C" and above | MELSEC iQ-F FX5-ENET User's Manual |
| FX5-ENET/IP is supported. | "1.110" and above | "1.050C" and above | MELSEC iQ-F FX5-ENET/IP User's Manual |
| MC protocol 1E frame is supported. | "1.210" and above | — | Page 42 |
| File transfer function (FTP client) is supported. | "1.210" and above | "1.065T" and above | Page 149 |
| Compatible with the following communication target devices (the simple CPU communication function) <ul style="list-style-type: none"> • MELSEC iQ-R (Built-in Ethernet) • MELSEC-Q (Built-in Ethernet) • MELSEC-L (Built-in Ethernet) • MELSEC iQ-L (Built-in Ethernet) • MELSEC-FX3 (Ethernet Block/Adapter) • OMRON (CJ/CP series) • KEYENCE (KV series) • Panasonic (FP7 series) • MODBUS/TCP-compatible device • SIEMENS S7 series • Panasonic (FP0H series) | "1.210" and above | "1.065T" and above | Page 201 |
| File acquisition from the FTP server is supported. | "1.210" and above | "1.065T" and above | Page 149 |

| Add/Change Function | Supported CPU module firmware version | Supported engineering tool software version | Reference |
|--|---------------------------------------|---|-----------|
| Compatible with the following communication target device (the simple CPU communication function) • MELSEC iQ-F (Ethernet module) | "1.210" and above | "1.065T" and above | Page 201 |


*1 Supported with CPU module serial No. 16Y**** and above.

The following file transfer function (FTP server) controls are supported by "1.050" and above.

- Write/delete the file to/from the SD memory card
- Unlock/lock the remote password
- Reset the file password
- Change the setting value of the response monitoring timer
- Allow Online Change

*2 The remote password setting to the FTP server is supported by "1.050" and above.

*3 The response monitoring timer setting/Allow Online Change setting of the file transfer function, and the remote password setting to the FTP server are supported by "1.035M" and above.

For the setting method of the remote password, refer to  GX Works3 Operating Manual.

Ethernet module

■When FX5UJ CPU module is used

| Add/Change Function | Supported versions | | | Reference |
|-----------------------------------|-----------------------------|----------------------------------|-----------------------------------|--|
| | CPU module firmware version | Ethernet module firmware version | Engineering tool software version | |
| MELSOFT connection | "1.010" and above | "1.100" and above | "1.075D" and above | Page 23 |
| SLMP communication function | "1.010" and above | "1.100" and above | "1.075D" and above | Page 42 |
| Simple CPU communication function | "1.010" and above | "1.100" and above | "1.075D" and above | Page 231 |
| BACnet function | "1.010" and above | "1.100" and above | "1.075D" and above | MELSEC iQ-F FX5 User's Manual (BACnet) |

■When FX5U/FX5UC CPU module is used

| Add/Change Function | Supported versions | | | Reference |
|-----------------------------------|-----------------------------|----------------------------------|-----------------------------------|---|
| | CPU module firmware version | Ethernet module firmware version | Engineering tool software version | |
| Firmware update function | "1.240" and above | "1.003" and above | "1.075D" and above | MELSEC iQ-F FX5 User's Manual (Application) |
| MELSOFT connection | "1.240" and above | "1.100" and above | "1.075D" and above | Page 23 |
| SLMP communication function | "1.240" and above | "1.100" and above | "1.075D" and above | Page 42 |
| Simple CPU communication function | "1.240" and above | "1.100" and above | "1.075D" and above | Page 231 |
| BACnet function | "1.240" and above | "1.100" and above | "1.075D" and above | MELSEC iQ-F FX5 User's Manual (BACnet) |

A

MEMO

INDEX

A

- Active open 91
- Allow online change 138

D

- Drive name (drive No.) 142

E

- External device 15

F

- FTP client 149
- FTP command 141
- FTP server 137
- Fullpassive 91

I

- IP filter function 192
- IP filter settings 195

L

- Lock processing 196

P

- Passive open 91

R

- Response monitoring timer 138

S

- SLMP 15
- SNTP client 162

U

- Unlock processing 196
- Unpassive 91



MEMO

REVISIONS

| Revision date | Revision | Description |
|---------------|----------|---|
| October 2014 | A | First Edition |
| January 2015 | B | <ul style="list-style-type: none"> ■Added functions Data code of ASCII to SLMP, Predefined protocol support function, Ethernet diagnostics ■Added or modified parts RELEVANT MANUALS, TERMS, Chapter 1, 2, 3, Section 4.1, Chapter 5, 6, Section 7.2, 7.3, 7.4, 10.1, 10.2, Appendix 1 |
| April 2015 | C | A part of the cover design is changed. |
| May 2016 | D | Errors are corrected. |
| October 2016 | E | <ul style="list-style-type: none"> ■Added functions Automatic detection of connected device, communication setting reflection of Ethernet device, and sensor parameter read/write, File transfer function (FTP server) ■Added or modified parts Chapter 1, Section 2.1, Chapter 3, Section 4.1, 4.2, 5.1, 5.2, 5.3, 5.4, 6.5, 7.1, 7.2, 7.3, Chapter 8, Section 11.3, 11.4, Appendix 2 |
| April 2017 | F | <ul style="list-style-type: none"> ■Added functions IP filter function, File transfer function (FTP server) (write/delete the file to/from the SD memory card etc.) ■Added or modified parts RELEVANT MANUALS, TERMS, Chapter 1, 3, 8, 9, Appendix 1, 2 |
| October 2017 | G | <ul style="list-style-type: none"> ■Added functions MODBUS/TCP communication function, Time setting function (SNTP client), Web server function ■Added or modified parts RELEVANT MANUALS, TERMS, Chapter 1, Section 2.1, Chapter 3, 4, Section 5.1, 5.2, 5.4, 6.5, 6.6, Chapter 7, Section 8.1, 8.3, Chapter 9, 10, Section 11.1, 13.3, Appendix 1, 2 |
| April 2018 | H | <ul style="list-style-type: none"> ■Added or modified parts Section 7.4, 10.2, 13.3, Appendix 1, 2 |
| July 2018 | J | <ul style="list-style-type: none"> ■Added functions User Web page ■Added or modified parts Chapter 1, 3, 10, Appendix 2 |
| October 2018 | K | <ul style="list-style-type: none"> ■Added models FX5-ENET ■Added functions Simple CPU communication function ■Added or modified parts INTRODUCTION, RELEVANT MANUALS, TERMS, Chapter 1, 2, 3, 7, Section 11.1, Chapter 12, 13, 14, Appendix 1, 2, 3 |
| December 2018 | L | <ul style="list-style-type: none"> ■Added functions JavaScript Objects of user Web page ■Added or modified parts Section 7.1, 7.2, 7.4, 10.1, 10.3, TRADEMARKS |
| October 2019 | M | <ul style="list-style-type: none"> ■Added models FX5UJ CPU module, FX5-ENET/IP ■Added or modified parts RELEVANT MANUALS, TERMS, Section 2.1, Chapter 3, 4, 5, Section 6.2, 6.5, 7.1, 7.2, 7.4, 8.1, Chapter 9, 10, 11, 12, Section 13.3, 13.4, 14.2, 14.3, 14.4, 14.5, Appendix 1, 2, 3 |
| May 2020 | N | <ul style="list-style-type: none"> ■Added functions SLMP communication function (1E frame), File transfer function (FTP client), Simple CPU communication function (Simple CPU communication function compatible device) ■Added or modified parts RELEVANT MANUALS, TERMS, Chapter 1, Section 2.1, Chapter 3, 4, 5, Section 6.5, 7.1, 7.2, Chapter 9, 13, Section 15.3, 15.4, 15.6, Appendix 1, 3, TRADEMARKS |
| August 2020 | P | <ul style="list-style-type: none"> ■Added or modified parts SAFETY PRECAUTIONS, WARRANTY |

| Revision date | Revision | Description |
|---------------|----------|--|
| April 2021 | Q | <p>■Added functions [FX5U/FX5UC CPU module] File acquisition from FTP server (file transfer function (FTP client)), MELSEC iQ-F (Ethernet module) (simple CPU communication function) [Ethernet module] Firmware update function, MELSOFT connection, SLMP communication function, simple CPU communication function, BACnet function</p> <p>■Added or modified parts RELEVANT MANUALS, TERMS, Chapter 1, Section 2.1, Chapter 3, 4, 5, 9, 13, Section 15.3, 15.4, 15.5, Appendix 2, 3, 4</p> |

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

© 2014 MITSUBISHI ELECTRIC CORPORATION

WARRANTY

Please confirm the following product warranty details before using this product.

1. Gratis Warranty Term and Gratis Warranty Range

If any faults or defects (hereinafter "Failure") found to be the responsibility of Mitsubishi occurs during use of the product within the gratis warranty term, the product shall be repaired at no cost via the sales representative or Mitsubishi Service Company. However, if repairs are required onsite at domestic or overseas location, expenses to send an engineer will be solely at the customer's discretion. Mitsubishi shall not be held responsible for any re-commissioning, maintenance, or testing on-site that involves replacement of the failed module.

[Gratis Warranty Term]

The gratis warranty term of the product shall be for one year after the date of purchase or delivery to a designated place. Note that after manufacture and shipment from Mitsubishi, the maximum distribution period shall be six (6) months, and the longest gratis warranty term after manufacturing shall be eighteen (18) months. The gratis warranty term of repair parts shall not exceed the gratis warranty term before repairs.

[Gratis Warranty Range]

- (1) The range shall be limited to normal use within the usage state, usage methods and usage environment, etc., which follow the conditions and precautions, etc., given in the instruction manual, user's manual and caution labels on the product.
- (2) Even within the gratis warranty term, repairs shall be charged for in the following cases.
 1. Failure occurring from inappropriate storage or handling, carelessness or negligence by the user. Failure caused by the user's hardware or software design.
 2. Failure caused by unapproved modifications, etc., to the product by the user.
 3. When the Mitsubishi product is assembled into a user's device, Failure that could have been avoided if functions or structures, judged as necessary in the legal safety measures the user's device is subject to or as necessary by industry standards, had been provided.
 4. Failure that could have been avoided if consumable parts (battery, backlight, fuse, etc.) designated in the instruction manual had been correctly serviced or replaced.
 5. Relay failure or output contact failure caused by usage beyond the specified life of contact (cycles).
 6. 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.
 7. Failure caused by reasons unpredictable by scientific technology standards at time of shipment from Mitsubishi.
 8. 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.

6. Product application

- (1) In using the Mitsubishi MELSEC programmable controller, the usage conditions shall be that the application will not lead to a major accident even if any problem or fault should occur in the programmable controller device, and that backup and fail-safe functions are systematically provided outside of the device for any problem or fault.
- (2) The Mitsubishi programmable controller has been designed and manufactured for applications in general industries, etc. Thus, applications in which the public could be affected such as in nuclear power plants and other power plants operated by respective power companies, and applications in which a special quality assurance system is required, such as for railway companies or public service purposes shall be excluded from the programmable controller applications.
In addition, applications in which human life or property that could be greatly affected, such as in aircraft, medical applications, incineration and fuel devices, manned transportation, equipment for recreation and amusement, and safety devices, shall also be excluded from the programmable controller range of applications. However, in certain cases, some applications may be possible, providing the user consults their local Mitsubishi representative outlining the special requirements of the project, and providing that all parties concerned agree to the special circumstances, solely at the user's discretion.
- (3) Mitsubishi shall have no responsibility or liability for any problems involving programmable controller trouble and system trouble caused by DoS attacks, unauthorized access, computer viruses, and other cyberattacks.

TRADEMARKS

Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Anywire and AnyWireASLINK are either registered trademarks or trademarks of Anywire Corporation.

Oracle and JavaScript are registered trademarks of Oracle Corporation and/or its affiliates in the United States and other countries.

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.

Manual number: JY997D56201Q

Model: FX5-U-EN-E

Model code: 09R543

When exported from Japan, this manual does not require application to the Ministry of Economy, Trade and Industry for service transaction permission.

MITSUBISHI ELECTRIC CORPORATION

HEAD OFFICE: TOKYO BUILDING, 2-7-3 MARUNOUCHI, CHIYODA-KU, TOKYO 100-8310, JAPAN

Specifications are subject to change without notice.