MITSUBISHI
ELECTRIC

Programmable Controller

MELSEC iQ-F series

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

- FX5S CPU module
- FX5UJ CPU module
- FX5U CPU module
- FX5UC CPU module
- Communication board
- Communication adapter
- FX5-ENET
- FX5-ENET/IP

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.
  - 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) to 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.
- Do not write any data to the "system area" of the buffer memory in an intelligent function module. Doing so may cause malfunction of the programmable controller system.
- When executing control (data change) to a running other station programmable controller by connecting the external device to the SLMP-compatible device or MC protocol-compatible device, configure interlock circuits in the program of the other station programmable controller to ensure that the entire system operates safely at any time.
  For other controls to a running other station programmable controller (such as program modification or operating status change), read the 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 MC protocol-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 or MC protocol-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 873 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 the 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.
- The temperature rating of the cable should be 80℃ or more.
- Make sure to wire the screw terminal block in accordance with the following precautions. Failure to do so may cause electric shock, equipment failures, a short-circuit, wire breakage, malfunctions, or damage to the product.
  - The disposal size of the cable end should follow the dimensions described in the manual.
  - Tightening torque should follow the specifications in the manual.
  - Tighten the screws using a Phillips-head screwdriver No.2 (shaft diameter 6 mm or less). Make sure that the screwdriver does not touch the partition part of the terminal block.
- Make sure to properly wire to the terminal block (European type) in accordance with the following precautions. Failure to do so may cause electric shock, equipment failures, a short-circuit, wire breakage, malfunctions, or damage to the product.
  - The disposal size of the cable end should follow the dimensions described in the manual.
  - Tightening torque should follow the specifications in the manual.
  - Twist the ends of stranded wires and make sure that there are no loose wires.
  - Do not solder-plate the electric wire ends.
  - Do not connect more than the specified number of wires or electric wires of unspecified size.
  - Fix the electric wires so that neither the terminal block nor the connected parts are directly stressed.

[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.
- Make sure to observe the following precautions in order to prevent any damage to the machinery or accidents due to malfunction of the PLC caused by abnormal data written to the PLC due to the effects of noise:
  - 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.
  - Ground the shield of the shield wire or shielded cable at one point on the PLC. However, do not use common grounding with heavy electrical systems.

[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. Read the manual thoroughly and ensure complete safety before executing other controls (for program change, parameter change, forcible output and operation status change) to the PLC in operation. 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 connecting or disconnecting any extension cable. Failure to do so may cause equipment failures or malfunctions.
- 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, bus conversion module and connector 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 MC protocol-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) to 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 communication function, serial communication, MODBUS communication, SLMP function, and MC protocol of the 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.
- Since the examples indicated by this manual, technical bulletin, catalog, etc. are used as a reference, please use it after confirming the function and safety of the equipment and system. Mitsubishi Electric will accept no 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.

CONTENTS

PART 1 Ethernet COMMUNICATION

CHAPTER 7 PREDEFINED PROTOCOL SUPPORT FUNCTION 100

CHAPTER 8 SOCKET COMMUNICATION FUNCTION 129

CHAPTER 9 FILE TRANSFER FUNCTION (FTP SERVER) 178

CHAPTER 10 FILE TRANSFER FUNCTION (FTP CLIENT) 190

PART 2 SERIAL COMMUNICATION

CHAPTER 17 PARALLEL LINK FUNCTION 286

CHAPTER 18 MC PROTOCOL 307

CHAPTER 19 INVERTER COMMUNICATION 329

CHAPTER 20 NON-PROTOCOL COMMUNICATION 411

CHAPTER 21 PREDEFINED PROTOCOL SUPPORT FUNCTION 452

PART 3 MODBUS/TCP COMMUNICATION

CHAPTER 22 OUTLINE 500

CHAPTER 23 SPECIFICATIONS 502

CHAPTER 24 MODBUS/TCP COMMUNICATION SPECIFICATIONS 504

CHAPTER 25 COMMUNICATION SETTING 506

CHAPTER 26 FUNCTION 510

PART 4 MODBUS SERIAL COMMUNICATION

CHAPTER 27 OUTLINE 516

CHAPTER 28 CONFIGURATION 518

CHAPTER 29 SPECIFICATIONS 521

CHAPTER 30 MODBUS COMMUNICATION SPECIFICATIONS 525

CHAPTER 31 WIRING 527

PART 5 SLMP

CHAPTER 38 3E FRAME COMMANDS 602

CHAPTER 39 1E FRAME COMMANDS 666

PART 6 MC PROTOCOL

CHAPTER 40 OUTLINE 684

CHAPTER 41 MC PROTOCOL DATA COMMUNICATION 686

CHAPTER 42 MESSAGE FORMAT 692

CHAPTER 43 COMMANDS 709

CHAPTER 44 COMMUNICATING USING 1C FRAMES 769

PART 7 TROUBLESHOOTING

CHAPTER 45 TROUBLESHOOTING PROCEDURE 794

CHAPTER 46 TROUBLESHOOTING BY SYMPTOM 805

CHAPTER 47 ERROR CODES 816

CHAPTER 48 EVENT CODES 855

APPENDICES 860

RELEVANT MANUALS

Manual name <manual number>	Description
MELSEC iQ-F FX5S/FX5UJ/FX5U/FX5UC User's Manual (Hardware) <SH-082452ENG>	Details of hardware of the CPU module, including I/O 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 (Communication) <SH-082625ENG> (this manual)	Describes the communication function of the built-in CPU module and the Ethernet module
MELSEC iQ-F FX5 Ethernet Module User's Manual <SH-082026ENG>	Describes the FX5-ENET.
MELSEC iQ-F FX5 EtherNet/IP Module User's Manual <SH-082027ENG>	Describes the FX5-ENET/IP.
MELSEC iQ-F FX5 BACnet Reference Manual <SH-082218ENG>	Describes the BACnet functions of the Ethernet module.
GX Works3 Operating Manual <SH-081215ENG>	Describes the system configuration, parameter settings, and online operations of GX Works3.

TERMS

Unless otherwise specified, this manual uses the following terms.

Term	Description
Buffer memory	Memory areas of intelligent function modules and SLMP-compatible devices for storing setting values and monitor values.
Built-in RS-485 port	Built-in RS-485 port of the CPU module
Connected station (host station)	Connected station (host station) indicates a station directly connected to an external device. Other station indicates a station connected to the connected station (host station) on the network.
Other station	
Data logging file	The file which stored data collected by data logging function.
Engineering tool	The product name of the software package for the MELSEC programmable controllers
Relay station	A station that includes two or more network modules. Transient transmission is performed through this station to stations on other networks.

GENERIC TERMS AND ABBREVIATIONS

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

Generic term/abbreviation	Description
Battery	A different name for FX3U-32BL
CLOSE instruction	A generic term for SP.SOCCLOSE instruction and GP.CLOSE instruction
Communication adapter	A generic term for FX5-232ADP and FX5-485ADP
Communication board	A generic term for FX5-232-BD, FX5-485-BD, and FX5-422-BD-GOT
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
Ethernet module	A generic term for FX5-ENET and FX5-ENET/IP
Ethernet-equipped module	A generic term for the following modules when the Ethernet function is used: • CPU module • FX5-ENET • FX5-ENET/IP
Expansion adapter	A generic term for adapter for FX5 CPU module
Expansion board	A generic term for boards for FX5S CPU module, FX5UJ CPU module, and FX5U CPU module
Extension module	A generic term for FX5 extension modules, FX3 extension modules, extension modules (extension cable type), and extension modules (extension connector type)

Generic term/abbreviation	Description
External device	A generic term for personal computers connected by Ethernet for data communication and other Ethernet-equipped modules.
FTP	An abbreviation for File Transfer Protocol. This protocol is used to transfer data files over a network.
FX3	A generic term for FX3S, FX3G, FX3GC, FX3U, and FX3UC programmable controllers
FX5	A generic term for FX5S, FX5UJ, FX5U, and FX5UC programmable controllers
FX5 CPU module	A generic term for FX5S CPU module, FX5UJ CPU module, FX5U CPU module, and FX5UC CPU module
FX5S CPU module	A generic term for FX5S-30MR/ES, FX5S-30MT/ES, FX5S-30MT/ESS, FX5S-40MR/ES, FX5S-40MT/ES, FX5S-40MT/ESS, FX5S-60MR/ES, FX5S-60MT/ES, FX5S-60MT/ESS, FX5S-80MR/ES*1, FX5S-80MT/ES*1, FX5S-80MT/ESS*1, FX5S-30MR/DS, FX5S-30MT/DS, FX5S-30MT/DSS, FX5S-40MR/DS, FX5S-40MT/DS, FX5S-40MT/DSS, FX5S-60MR/DS, FX5S-60MT/DS, FX5S-60MT/DSS, FX5S-80MR/DS*1, FX5S-80MT/DS*1, and FX5S-80MT/DSS*1
FX5U CPU module	A 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	A 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
FX5UJ CPU module	A 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, FX5UJ-60MT/ESS, FX5UJ-24MR/DS, FX5UJ-24MT/DS, FX5UJ-24MT/DSS, FX5UJ-40MR/DS, FX5UJ-40MT/DS, FX5UJ-40MT/DSS, FX5UJ-60MR/DS, FX5UJ-60MT/DS, and FX5UJ-60MT/DSS
GOT	A generic term for Mitsubishi Electric Graphic Operation Terminal GOT1000 and GOT2000 series
GX Works3	The product name of the software package, SWnDND-GXW3, for the MELSEC programmable controllers (The 'n' represents a version.)
Intelligent function module	A generic term for modules that have functions other than input and output
MC protocol	An 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	A 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.
Module access device	A generic term for the module access devices of the MELSEC iQ-R series/MELSEC iQ-F series and intelligent function module devices of the MELSEC-Q/L series
OPEN instruction	A generic term for SP.SOCOPEN instruction and GP.OPEN instruction
Peripheral device	A generic term for engineering tools and GOTs
SD memory card	A generic term for NZ1MEM-2GBSD, NZ1MEM-4GBSD, NZ1MEM-8GBSD, NZ1MEM-16GBSD, L1MEM-2GBSD, and L1MEM-4GBSD SD memory cards. An abbreviation for Secure Digital Memory Card. Device that stores data using flash memory.
Serial port	A generic term for CPU module built-in RS-485 port (CH1), communication board (CH2), communication adapter 1 (CH3), and communication adapter 2 (CH4) (4 ports)
SLMP	An 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	A generic term for devices that can receive SLMP messages.
SOCRCV instruction	A generic term for SP.SOCRCV instruction and GP.SOCRCV instruction
SOCSND instruction	A generic term for SP.SOCSND instruction and GP.SOCSND instruction
TCP	An 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	An 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.)

*1 Area-specific model

MEMO

PART 1

Ethernet COMMUNICATION

This part consists of the following chapters.

1 OUTLINE

The following describes the Ethernet communication function of 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.

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.

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 the following.

☞ Page 500 OUTLINE

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.

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 remote stations using link devices (cyclic transmission).

For details, refer to 📖 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).

For details, refer to 📖 MELSEC iQ-F FX5 EtherNet/IP Module User's Manual.

BACnet communication function

Service can be executed from each device to communicate data.

For details, refer to 📖 MELSEC iQ-F FX5 BACnet Reference 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.

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		100m (length between hub and node)[*2]
	Number of cascade connections	100BASE-TX	2 levels maximum[*3]
		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. When a device such as a GOT and CPU module is directly connected using the Ethernet cable, a cross cable of category 5e or lower can be used as well.

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 the 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

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

For another specifications, refer to the following manuals.

MELSEC iQ-F FX5 Ethernet Module User's Manual

MELSEC iQ-F FX5 EtherNet/IP Module 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		100m (length between hub and node)[2]
	Number of cascade connections	100BASE-TX	2 levels maximum[3]
		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, MQTT[4], SMTP client[4]
		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[5] (Up to 32 external devices can access one Ethernet module at the same time.)
Hub[1]			Hubs with 100BASE-TX or 10BASE-T ports[6] can be used.
IP address			Initial value: 192.168.3.251
Connection cable[7]	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 For the versions compatible with MQTT and the SMTP client (e-mail function), refer to ☞ Page 936 Added and Changed Functions.

*5 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.)

*6 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.

*7 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 the 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

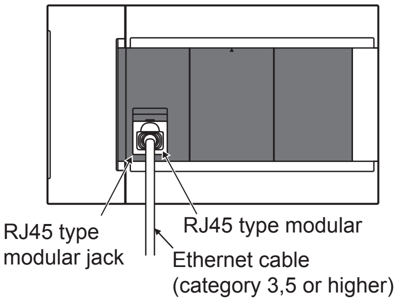
This section describes how to connect the Ethernet to a 100BASE-TX/10BASE-T network.

<Connection procedure>

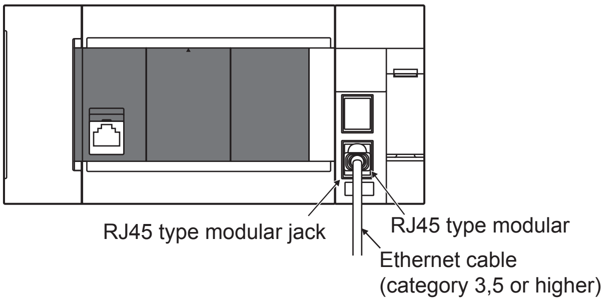
1. Connect the Ethernet cable to a hub.
2. Connect the Ethernet cable to the Ethernet.

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 lists the functions of the built-in Ethernet of the Ethernet-equipped module.

Some functions have restrictions on the firmware version and production information of the CPU module or the version of the engineering tool. (☞ Page 936 Added and Changed Functions)

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

Function	Outline of system	Supported function					Reference
		CPU module			Ethernet module		
		FX5S	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 31 Direct Connection with Engineering Tool
MELSOFT connection	Communication with MELSOFT products (GX Works3, etc.) is done within LAN such as company internal LAN.	○	○	○	○	○	Page 38 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 44 Searching Ethernet-equipped modules on network
MELSOFT diagnosis function	Diagnoses Ethernet port of CPU module and Ethernet module from GX Works3. (Ethernet diagnostics)	○	○	○	○	○	Page 799 Ethernet diagnostics
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 49 SIMPLE CPU COMMUNICATION FUNCTION
SLMP communication function	Reads and writes data from other device.	○	○	○	○	○	Page 90 SLMP FUNCTION
Predefined protocol support function	When the predefined protocol support function is used, data can be exchanged with the external device.	○	○	○	—	—	Page 100 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 129 SOCKET COMMUNICATION FUNCTION
MODBUS/TCP communication	By using sequence program, MODBUS devices of the external devices connected through Ethernet can be read/written.	○	○	○	—	—	Page 500 OUTLINE
File transfer function (FTP server)	Using the dedicated FTP commands enables an external device to read out, write, and delete individual data file.	△[*1]	○	○	—	—	Page 178 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.	△[*1]	○	○	—	—	Page 190 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 203 TIME SETTING FUNCTION (SNTP CLIENT)
Web server function	Monitors and diagnoses the CPU module using a Web browser via connected network.	△[*1]	○	○	—	—	Page 206 WEB SERVER FUNCTION 📖 MELSEC iQ-R/ MELSEC iQ-F Web Server Function Guide Book

Function	Outline of system	Supported function					Reference
		CPU module			Ethernet module		
		FX5S	FX5UJ	FX5U/ FX5UC	FX5-ENET	FX5-ENET/ IP	
IP filter function	This function identifies IP address of the access source and prevents access by unauthorized IP addresses.	○	○	○	○	○	Page 232 IP Filter Function
Remote password	Remote password setting can prevent unauthorized access from the outside and enhance the security of the system.	○	○	○	—	—	Page 235 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.	○	○	○	○	○	Page 240 IP ADDRESS CHANGE FUNCTION
CC-Link IE Field Network Basic	Data is periodically communicated between the master station and remote 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 EtherNet/IP Module 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
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 BACnet Reference Manual
E-mail function	Sends email messages via mail servers over networks to personal computers or smartphones in remote locations.	—	—	—	○	—	📖 MELSEC iQ-F FX5 Ethernet Module User's Manual
DNS setting	Specifies the IP address of the DNS server to set the host name as a domain name in the e-mail function or the MQTT communication function.	—	—	—	○	—	
MQTT communication function	Publishes (submits) information collected on a programmable controller (such as information input from sensors) to an MQTT broker (on the cloud or a local network) or subscribes information from an MQTT broker.	—	—	—	○	—	

*1 An SD memory card module is required.

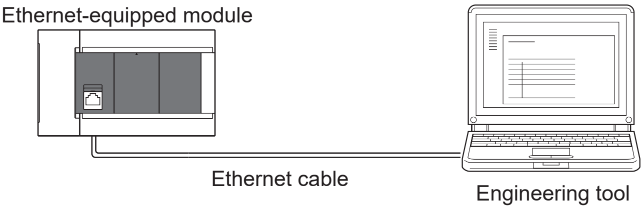
4 CONNECTION WITH MELSOFT PRODUCT AND GOT

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.



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] ⇨ Module model name ⇨ [Module Parameter] ⇨ [Ethernet Port] ⇨ [Application Settings] ⇨ [Security]

[Ethernet module]

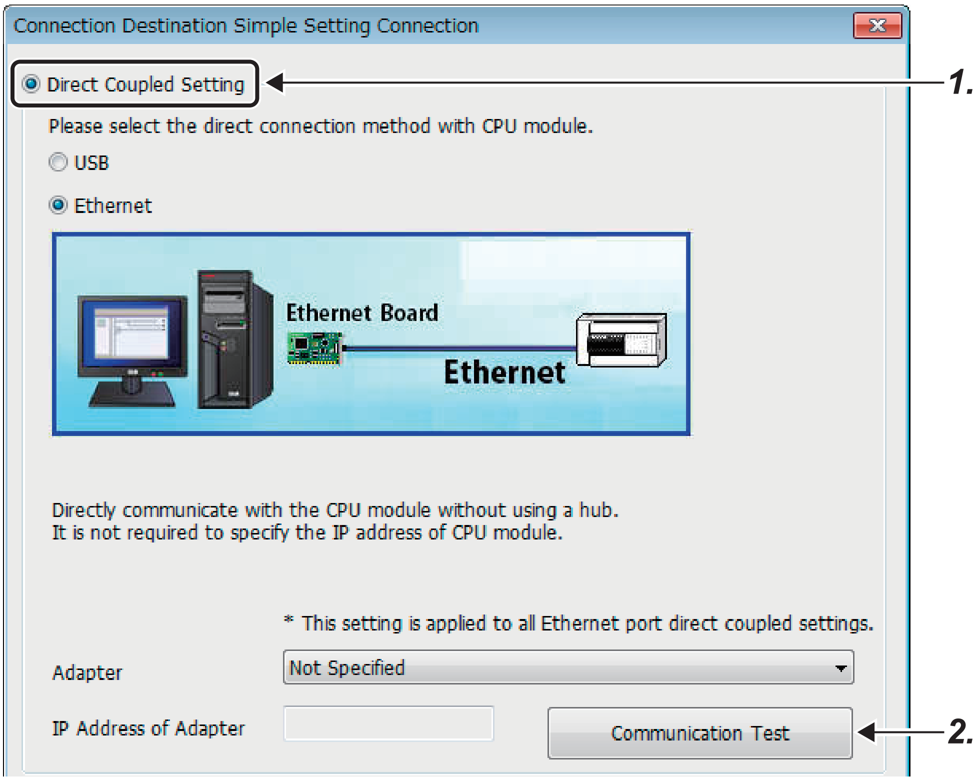
Navigation window ⇨ [Parameter] ⇨ [Module Information] ⇨ [FX5-ENET] or [FX5-ENET/IP] ⇨ [Application Settings] ⇨ [Security]

Setting method

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

[Online] ⇨ [Current Connection Destination]

Simple setting method



Connection Destination Simple Setting Connection

● Direct Coupled Setting

Please select the direct connection method with CPU module.

○ USB

● Ethernet

Directly communicate with the CPU module without using a hub.
It is not required to specify the IP address of CPU module.

* This setting is applied to all Ethernet port direct coupled settings.

Adapter: Not Specified

IP Address of Adapter:

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

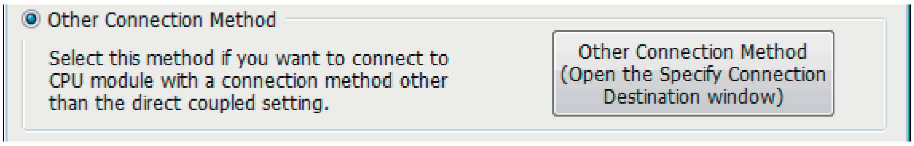
Point

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

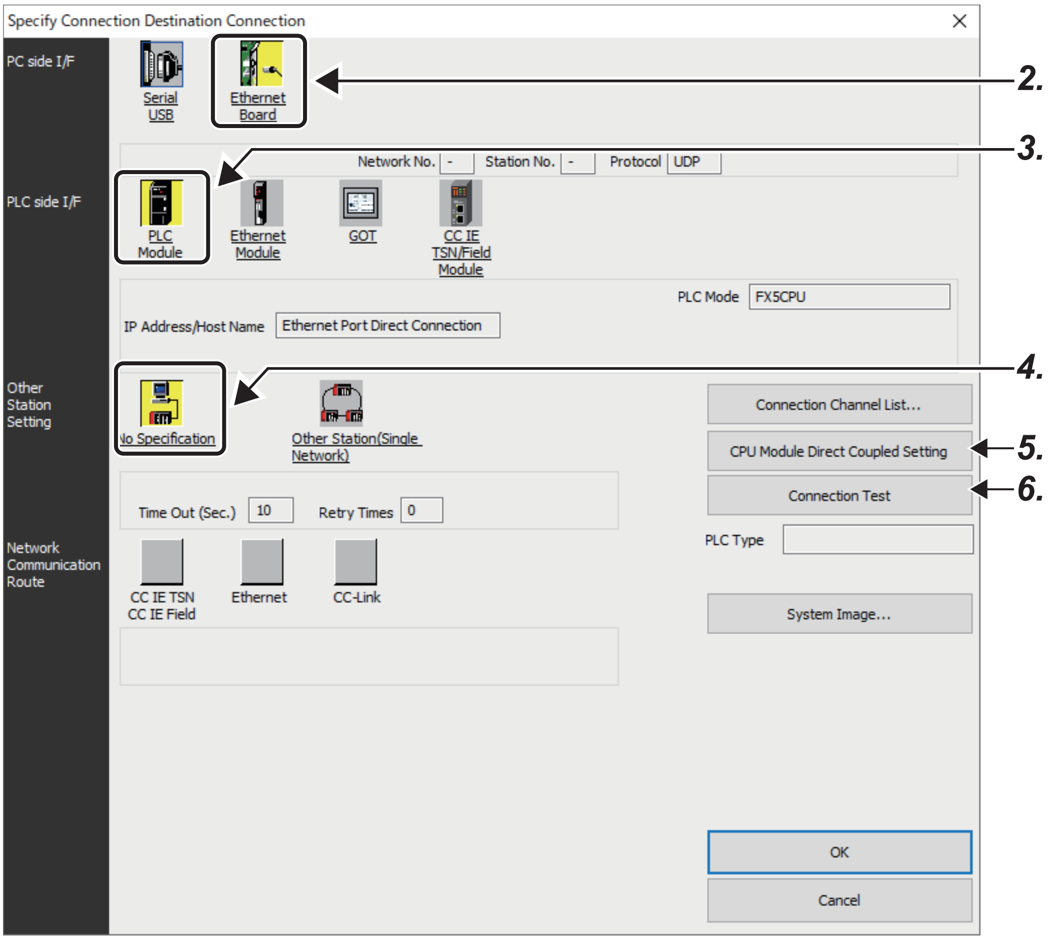
■CPU module



● Other Connection Method

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

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

Network No. -　Station No. -　Protocol UDP

PLC Mode FX5CPU

IP Address/Host Name Ethernet Port Direct Connection

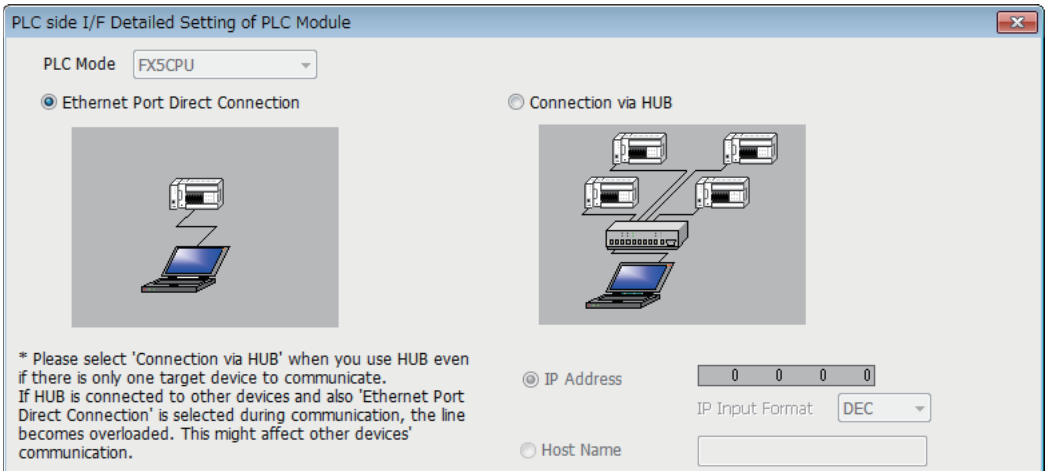
Time Out (Sec.) 10　Retry Times 0

PLC Type

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.



PLC side I/F Detailed Setting of PLC Module

PLC Mode FX5CPU

● Ethernet Port Direct Connection　○ Connection via HUB

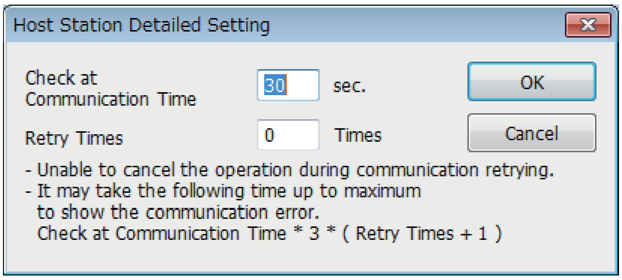
* Please select 'Connection via HUB' when you use HUB even if there is only one target device to communicate.
If HUB is connected to other devices and also 'Ethernet Port Direct Connection' is selected during communication, the line becomes overloaded. This might affect other devices' communication.

● IP Address 0 0 0 0

IP Input Format DEC

○ Host Name

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



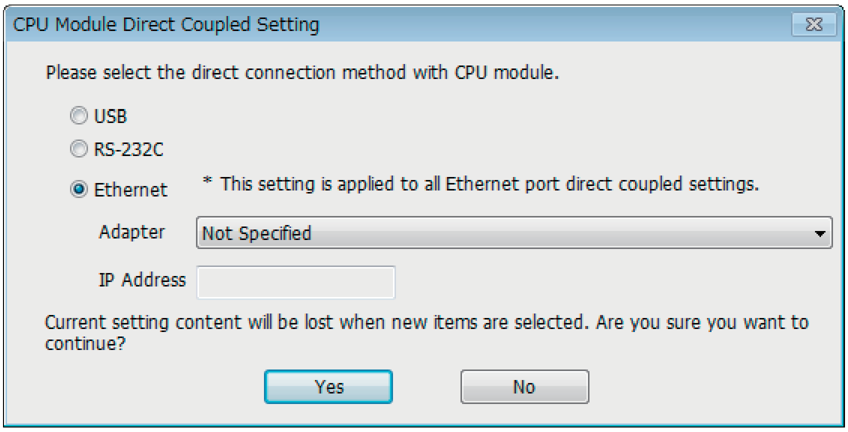
Host Station Detailed Setting

Check at Communication Time: 30 sec.

Retry Times: 0 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)

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



CPU Module Direct Coupled Setting

Please select the direct connection method with CPU module.

○ USB
○ RS-232C
● Ethernet   * This setting is applied to all Ethernet port direct coupled settings.

Adapter: Not Specified

IP Address:

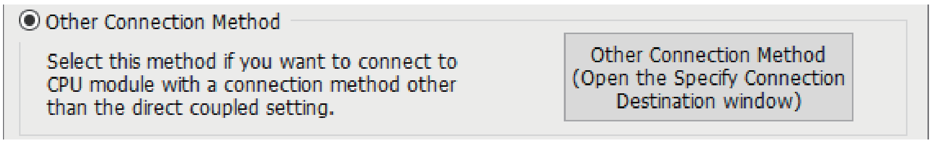
Current setting content will be lost when new items are selected. Are you sure you want to continue?

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.

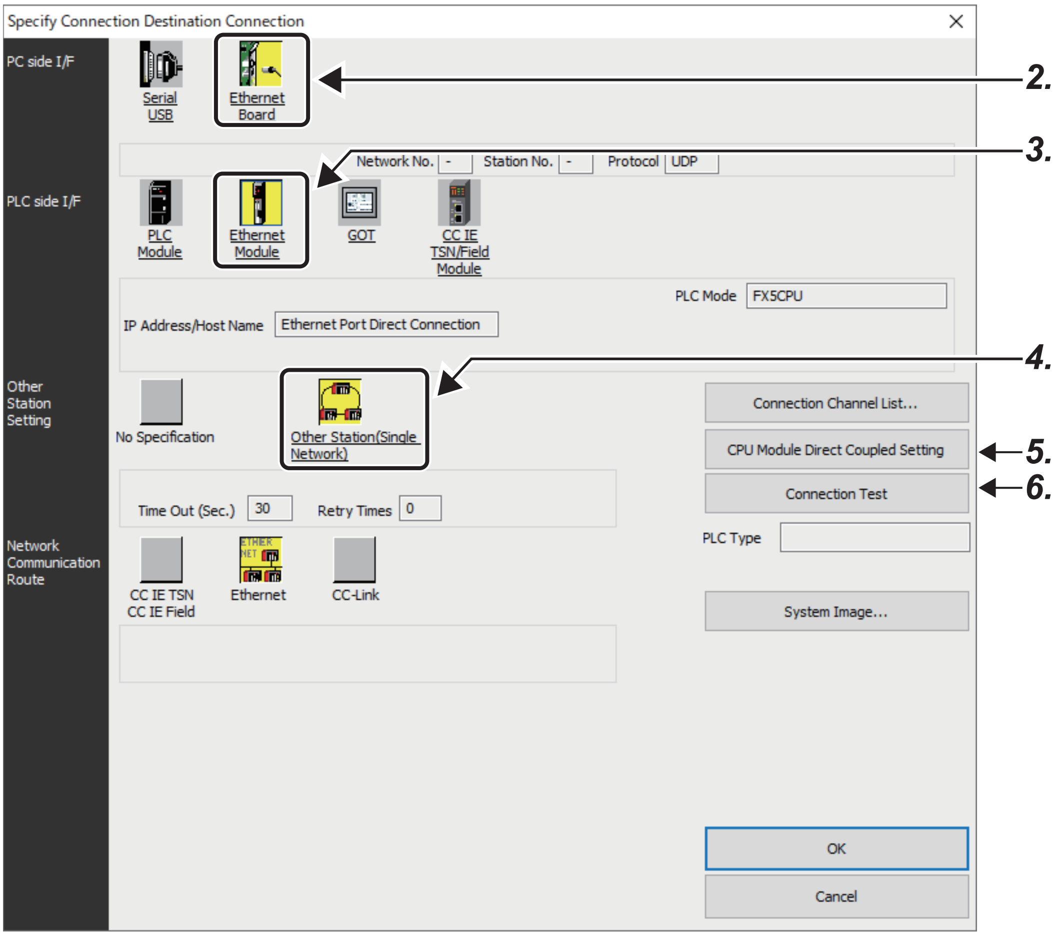
6. Click the [Connection Test] button on the "Specify Connection Destination Connection" screen and check if the connection for the CPU module is possible.

■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.

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.

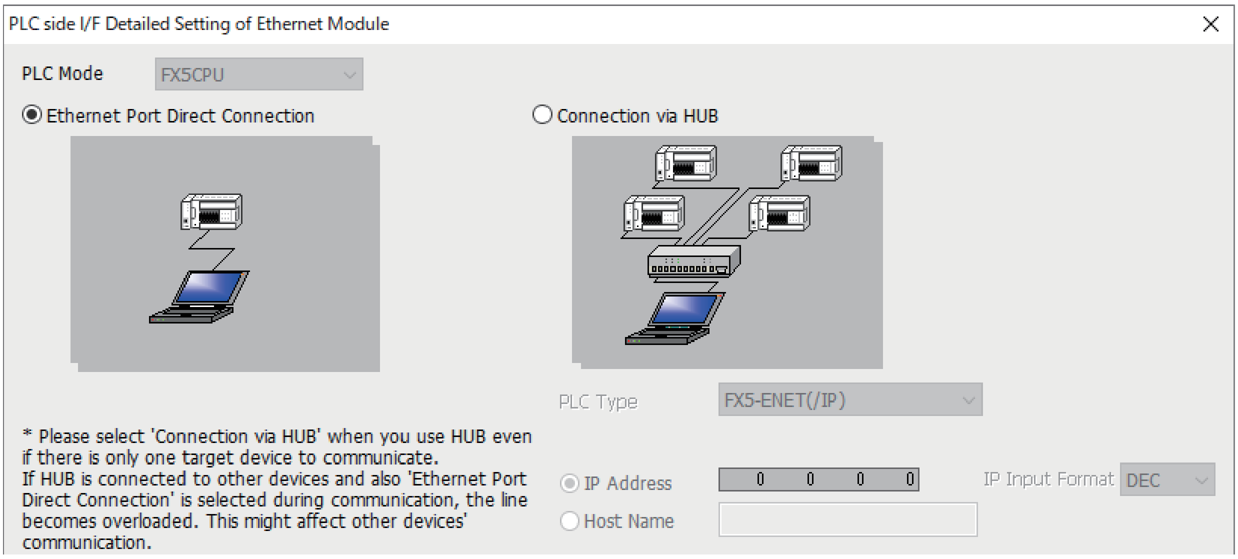


Network No. -
Station No. -
Protocol UDP
PLC Mode FX5CPU
IP Address/Host Name Ethernet Port Direct Connection
Time Out (Sec.) 30
Retry Times 0

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.



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.

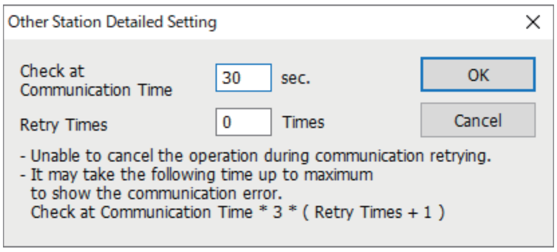
○ IP Address: 0 0 0 0

IP Input Format: DEC

○ Host Name

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

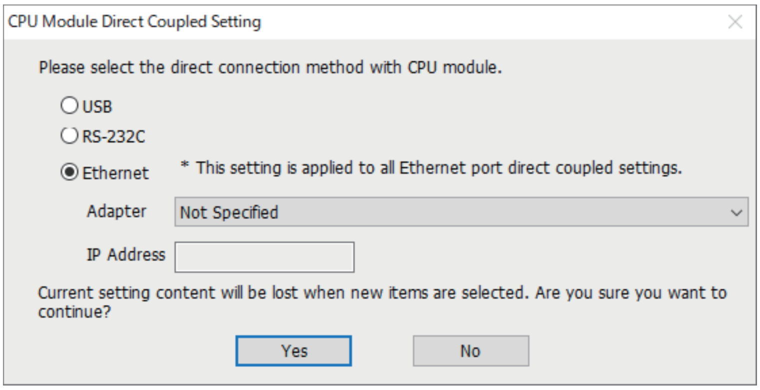
Check at Communication Time: 30 sec.

Retry Times: 0 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)

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

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



CPU Module Direct Coupled Setting

Please select the direct connection method with CPU module.

○ USB
○ RS-232C
● Ethernet

* This setting is applied to all Ethernet port direct coupled settings.

Adapter: Not Specified

IP Address:

Current setting content will be lost when new items are selected. Are you sure you want to continue?

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 module can be connected with the Ethernet module.

Precautions

Connection to LAN line

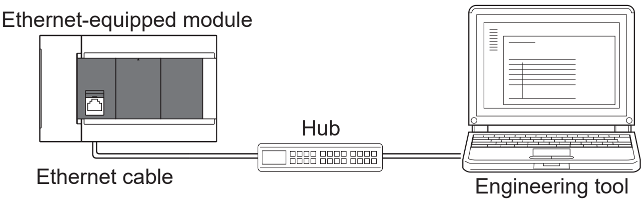
When connecting the CPU module to a LAN line, do not set direct connection. Doing so will apply a load on the LAN line and adversely affect communications with other external devices.

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 an Ethernet-equipped module is connected to an external device via a hub, communication cannot be performed by direct connection. (☞ Page 38 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.

IP address of the Ethernet-equipped module: 64. 64. 255. 255
Personal computer IP address: 64. 64. 1. 1
Personal computer-side 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 ← Class C and the host address is the fourth octet because the IP address is 192.x.x.x.
Personal computer-side subnet mask: 255. 0. 0. 0
IP address of the Ethernet-equipped module: 64.64.255.255 ← Each bit turns ON because 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.

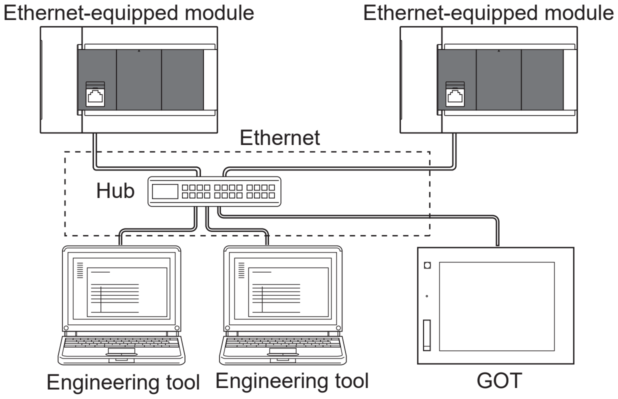
- Class A: 255. 0. 0. 0
- Class B: 255. 255. 0. 0
- Class C: 255. 255. 255. 0

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, configure the 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 module parameters (IP address) with the engineering tool. (☞ Page 39 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 the power OFF → ON or reset the system to enable the parameters. (☞ Page 40 Writing to the Ethernet-equipped module)

3. Connecting cables and external devices

Connect them via a hub to perform Ethernet communication. (☞ Page 28 Connection Specifications)

4. Setting the connection destination

Set connection destination with the engineering tool. (☞ Page 41 Engineering tool settings)

For GOT settings, refer to the following manuals.

📖 GOT2000 Series Connection Manual (Mitsubishi Products)

📖 GOT1000 Series Connection Manual (Mitsubishi Products)

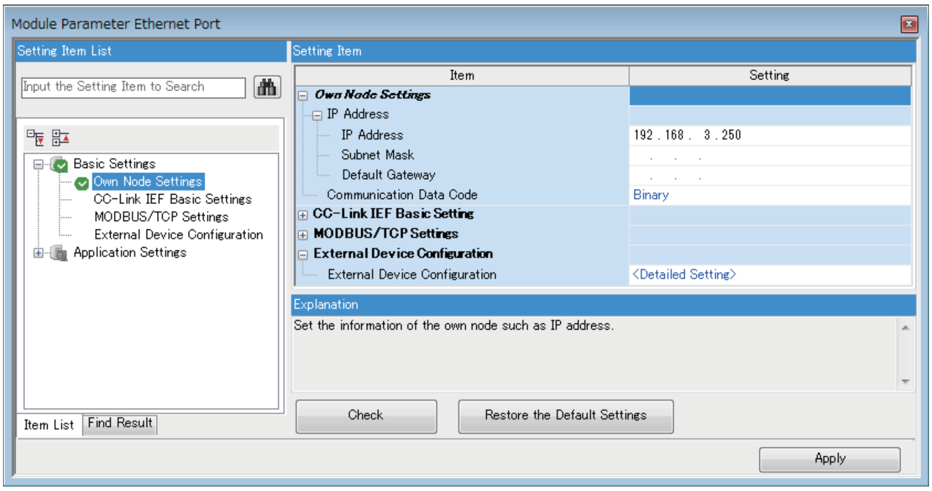
Setting the Ethernet-equipped module

Setting module parameters

■CPU module

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

Navigation window ⇨ [Parameter] ⇨ Module name ⇨ [Module Parameter] ⇨ [Ethernet Port] ⇨ [Basic Settings] ⇨ [Own Node Settings]



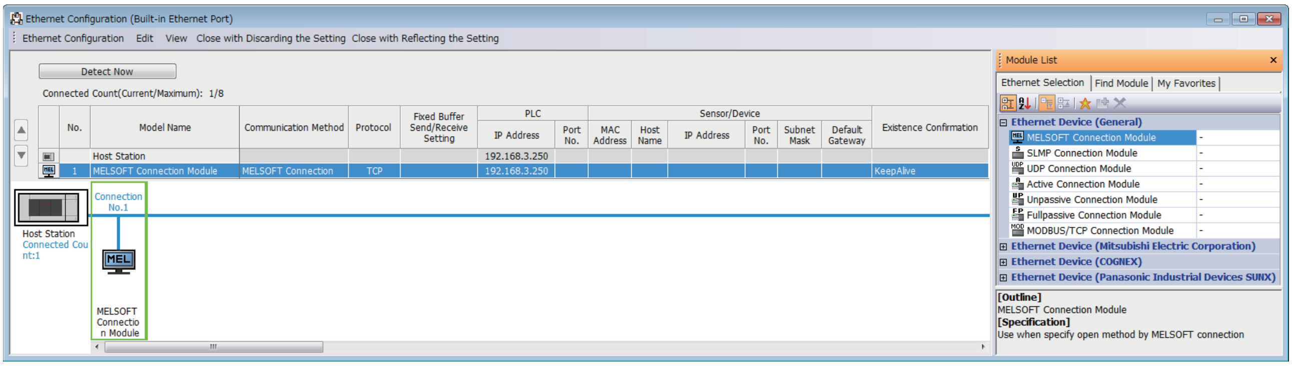
Item	Setting
Own Node Settings	
IP Address	
IP Address	192 . 168 . 3 . 250
Subnet Mask	. . .
Default Gateway	. . .
Communication Data Code	Binary
CC-Link IEF Basic Setting	
MODBUS/TCP Settings	
External Device Configuration	
External Device Configuration	<Detailed Setting>

Explanation: Set the information of the own node such as IP address.

1. Set IP address of the CPU module.

2. Configure the connection settings to perform the MELSOFT connection.

Navigation window ⇨ [Parameter] ⇨ Module model name ⇨ [Module Parameter] ⇨ [Ethernet Port] ⇨ [Basic Settings] ⇨ [External Device Configuration] ⇨ [Detailed Setting] ⇨ [Ethernet Configuration (Built-in Ethernet Port)] screen



Connected Count(Current/Maximum): 1/8

No.	Model Name	Communication Method	Protocol	Fixed Buffer Send/Receive Setting	PLC		Sensor/Device						Existence Confirmation
					IP Address	Port No.	MAC Address	Host Name	IP Address	Port No.	Subnet Mask	Default Gateway	
	Host Station				192.168.3.250								
1	MELSOFT Connection Module	MELSOFT Connection	TCP		192.168.3.250								KeepAlive

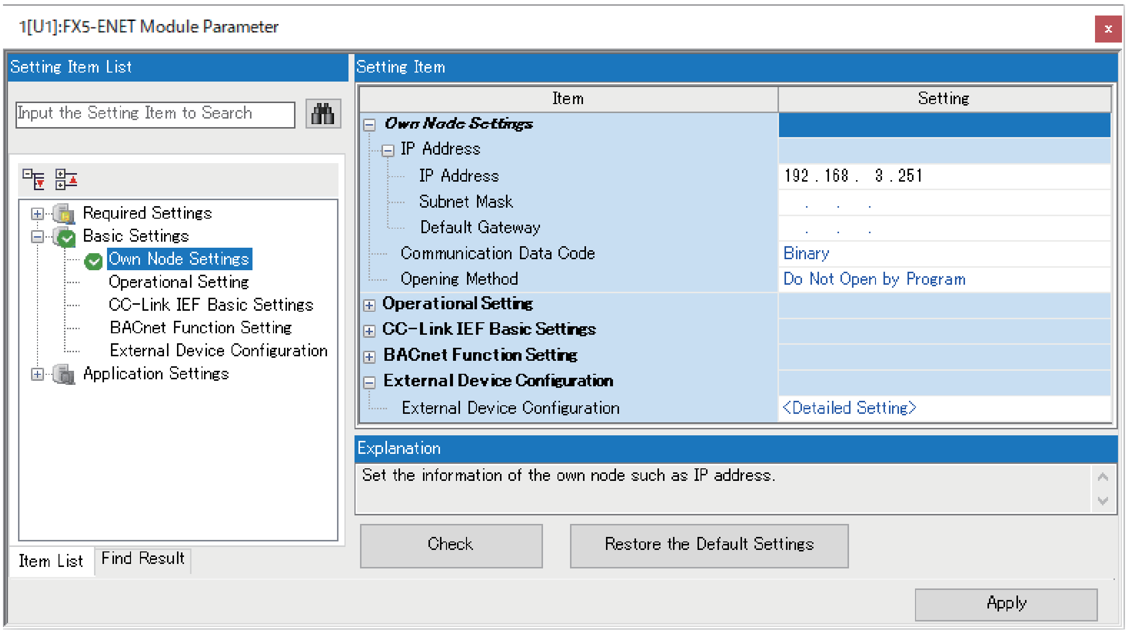
[Outline]
MELSOFT Connection Module
[Specification]
Use when specify open method by MELSOFT connection

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]



Item	Setting
Own Node Settings	
IP Address	
IP Address	192 . 168 . 3 . 251
Subnet Mask	. . .
Default Gateway	. . .
Communication Data Code	Binary
Opening Method	Do Not Open by Program
Operational Setting	
CC-Link IEF Basic Settings	
BACnet Function Setting	
External Device Configuration	
External Device Configuration	<Detailed Setting>

Set the information of the own node such as IP address.

1. Set the IP address of the Ethernet module.

Writing to the Ethernet-equipped module

Write the set parameters to the Ethernet-equipped module.

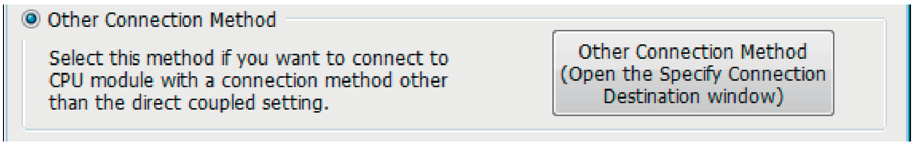
[Online] ⇨ [Write to PLC]

After writing the parameters to the Ethernet-equipped module, turn OFF and ON or reset the power of 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]

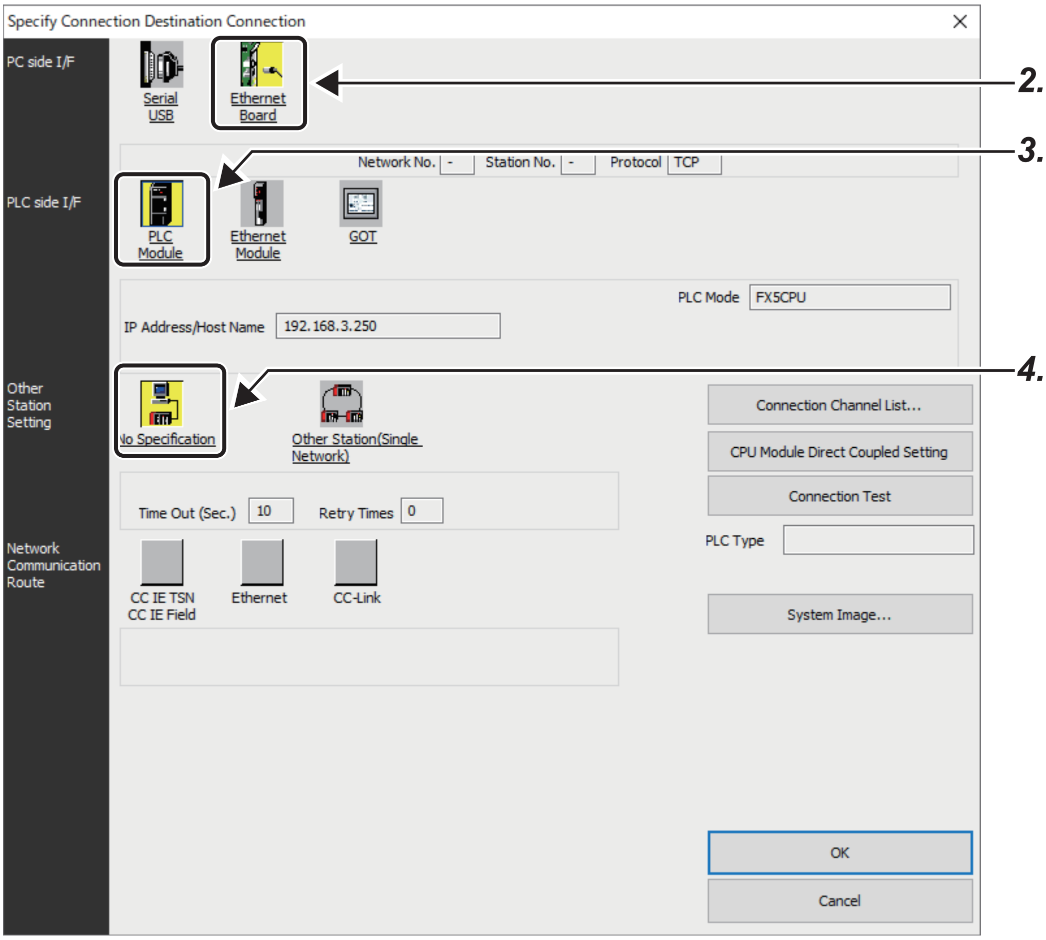


● Other Connection Method

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

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.



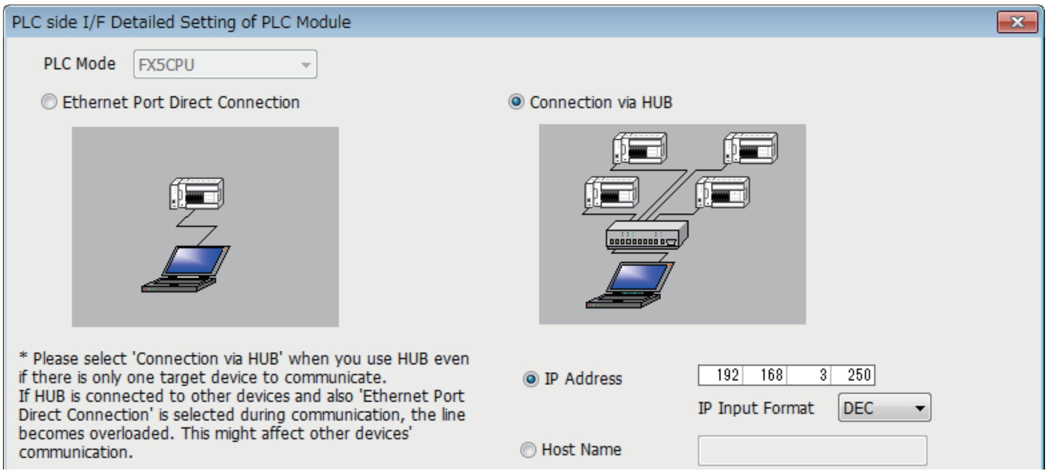
Network No. -
Station No. -
Protocol TCP
PLC Mode FX5CPU
IP Address/Host Name 192.168.3.250
Time Out (Sec.) 10
Retry Times 0

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

3. Select "PLC 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.



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.

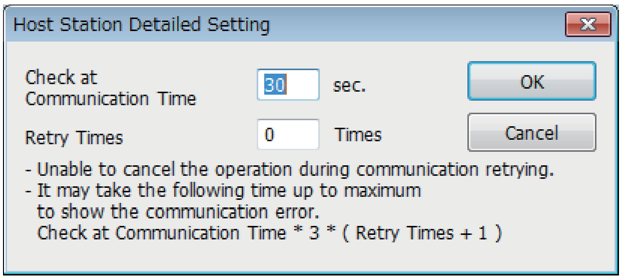
● IP Address: 192.168.3.250

IP Input Format: DEC

○ Host Name:

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

Set the other station specifications according to the operating environment.



Host Station Detailed Setting

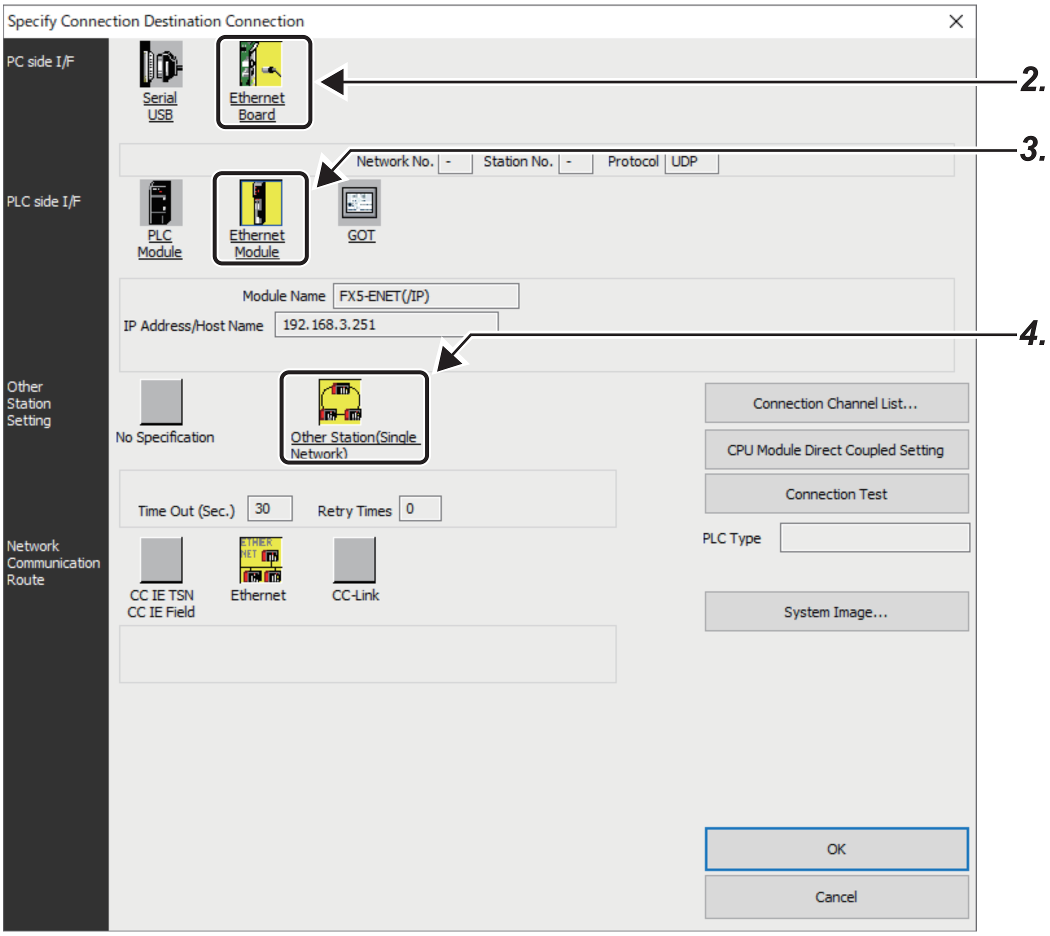
Check at Communication Time: 30 sec.

Retry Times: 0 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)

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.



Specify Connection Destination Connection

PC side I/F: Ethernet Board

PLC side I/F: Ethernet Module

Network No. -

Station No. -

Protocol UDP

Module Name FX5-ENET(/IP)

IP Address/Host Name 192.168.3.251

Other Station Setting: Other Station(Single Network)

Time Out (Sec.) 30

Retry Times 0

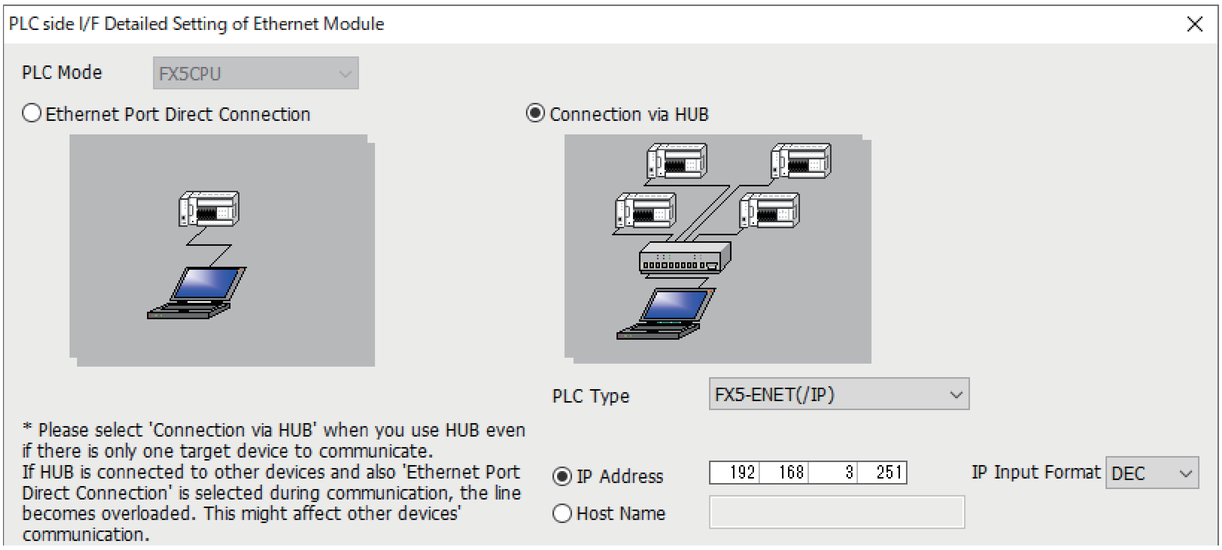
Network Communication Route: Ethernet

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.



PLC side I/F Detailed Setting of Ethernet Module

PLC Mode FX5CPU

○ Ethernet Port Direct Connection

● Connection via HUB

PLC Type FX5-ENET(/IP)

● IP Address 192 168 3 251

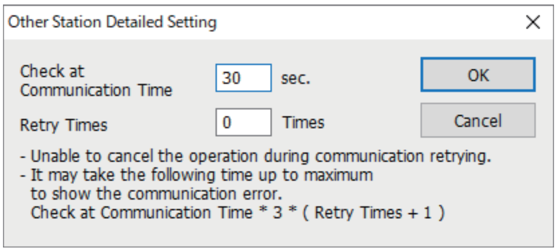
IP Input Format DEC

○ Host Name

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

Set the other station specifications according to the operating environment.



Check at Communication Time: 30 sec.

Retry Times: 0 Times

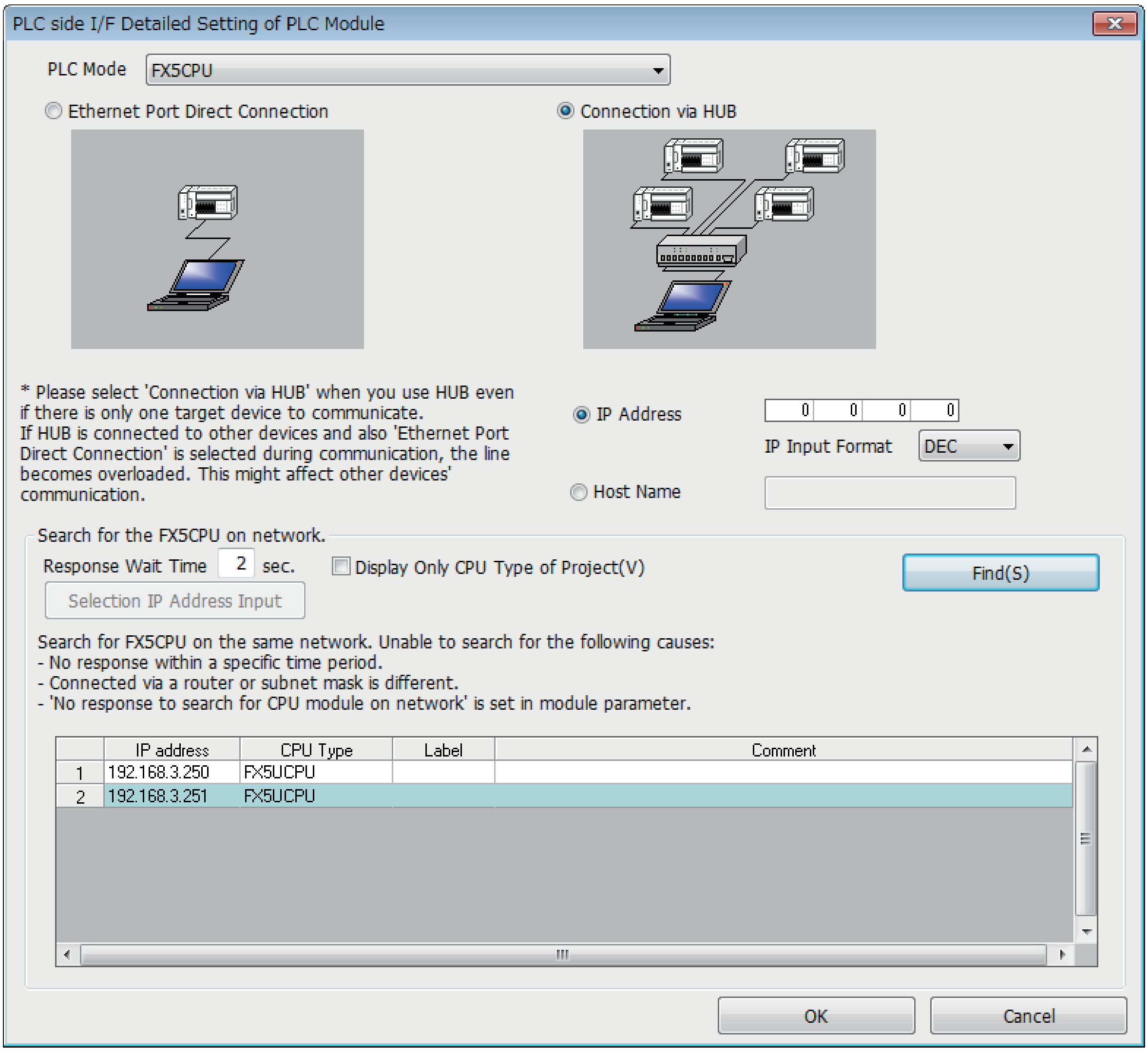
- Unable to cancel the operation during communication retrying.
- It may take the following time up to maximum to show the communication error.
  Check at Communication Time * 3 * (Retry Times + 1)

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 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.

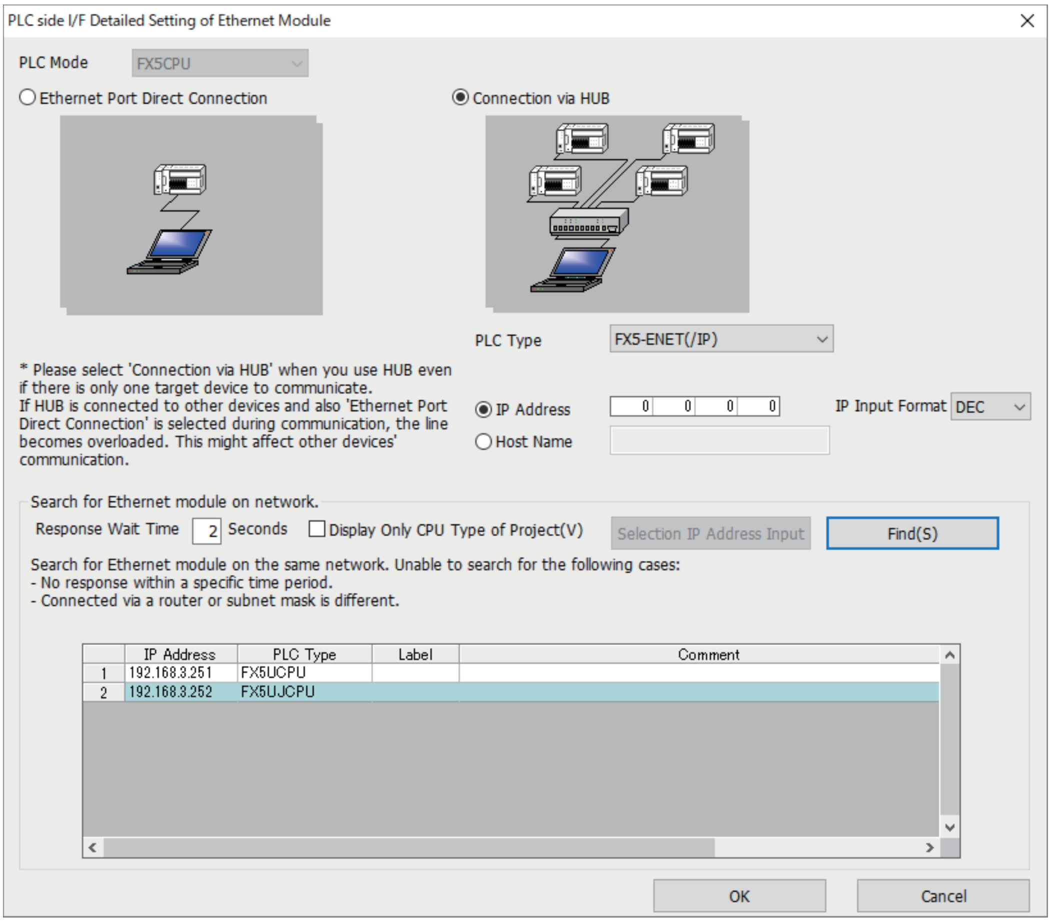
☐ Display Only CPU Type of Project(V)

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		

[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)

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		

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] ➩ Module model name ➩ [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

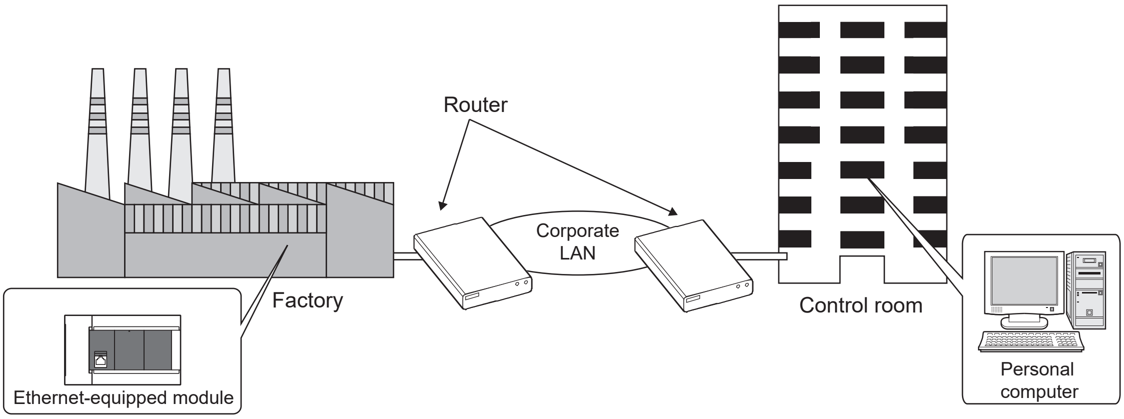
- The Ethernet-equipped module whose address is denied by the IP filter function cannot be searched.
- Ethernet-equipped modules connected via router cannot be searched.
- Some Ethernet-equipped modules connected via wireless LAN may not be found since Ethernet communication may not be stable due to 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]

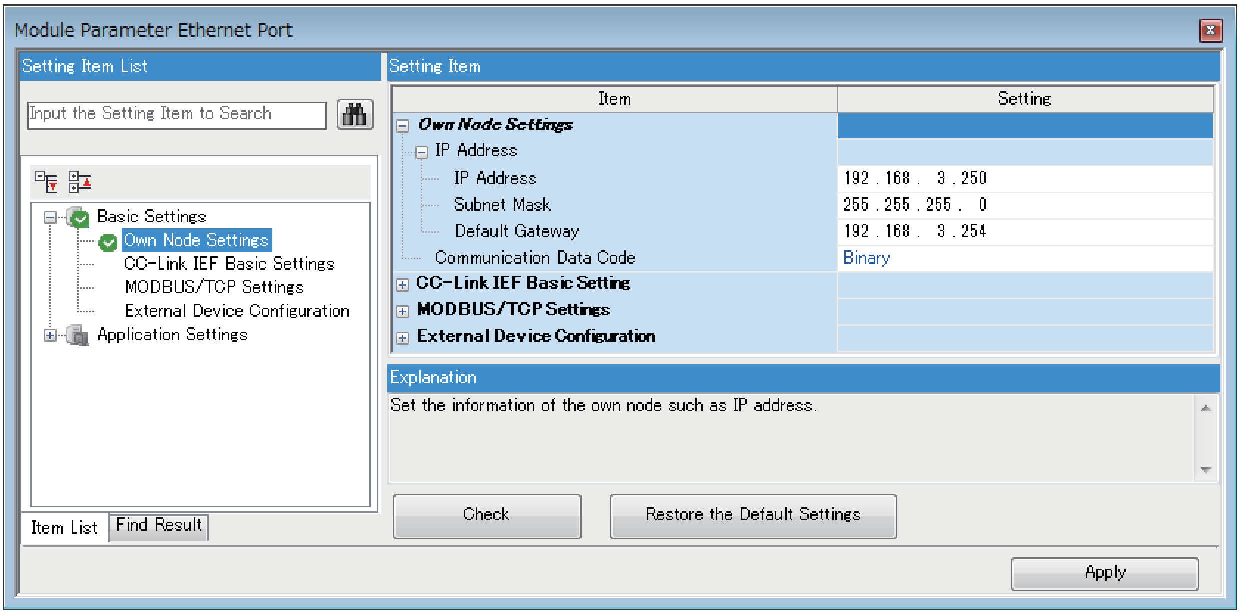


*1 Communication through routers is impossible for some functions. The following functions do not support communication via routers.
· Searching Ethernet-equipped modules on Network

For access via router, set the subnet mask pattern and default gateway IP address in addition to IP address as per ☞ Page 39 Setting module parameters.

[CPU module]

Navigation window ⇨ [Parameter] ⇨ Module name ⇨ [Module Parameter] ⇨ [Ethernet Port] ⇨ [Basic Settings] ⇨ [Own Node Settings]

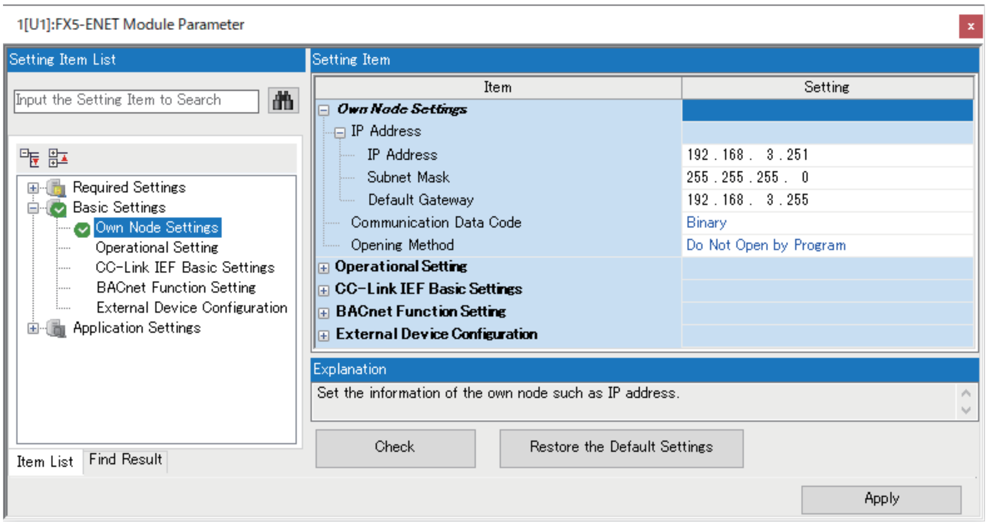


Item	Setting
Own Node Settings	
IP Address	
IP Address	192 . 168 . 3 . 250
Subnet Mask	255 . 255 . 255 . 0
Default Gateway	192 . 168 . 3 . 254
Communication Data Code	Binary
CC-Link IEF Basic Setting	
MODBUS/TCP Settings	
External Device Configuration	

Set the information of the own node such as IP address.

[Ethernet module]

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



Item	Setting
Own Node Settings	
IP Address	
IP Address	192 . 168 . 3 . 251
Subnet Mask	255 . 255 . 255 . 0
Default Gateway	192 . 168 . 3 . 255
Communication Data Code	Binary
Opening Method	Do Not Open by Program
Operational Setting	
CC-Link IEF Basic Settings	
BACnet Function Setting	
External Device Configuration	

Set the information of the own node such as IP address.

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. (Response to 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.

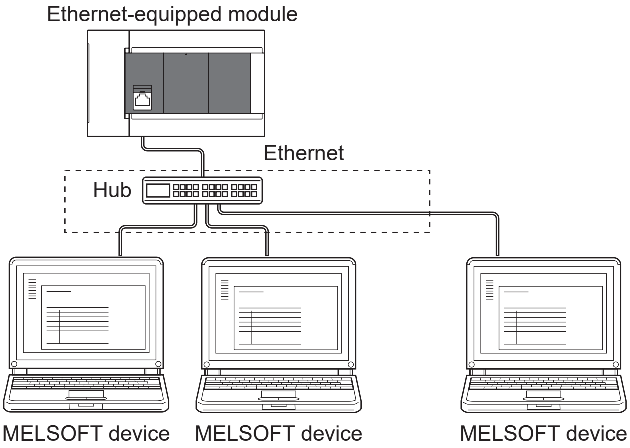
Retransmission in case of TCP connection

In the TCP connection, if no TCP ACK response is returned from the external device in response to a transmission, resending will be performed.

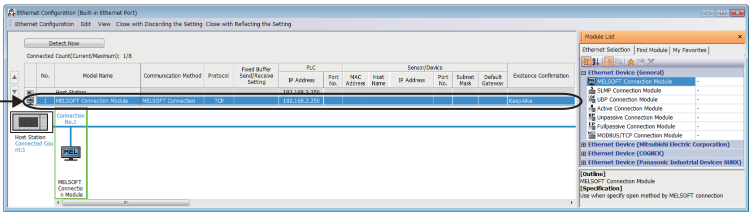
Resending will be performed twelve times, starting 3 seconds after the first transmission, and then 6, 12, 24, 48, 60, and thereafter every 60 seconds. When no TCP ACK response is returned within 60 seconds after the last retransmission, the external device is regarded as faulty and the connection is 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 module parameters.



Set the same number of devices as MELSOFT devices



Connected Count(Current/Maximum): 1/8

No.	Model Name	Communication Method	Protocol	Fixed Buffer Send/Receive Setting	PLC		Sensor/Device						Existence Confirmation
					IP Address	Port No.	MAC Address	Host Name	IP Address	Port No.	Subnet Mask	Default Gateway	
	Host Station				192.168.3.250								
1	MELSOFT Connection Module	MELSOFT Connection	TCP		192.168.3.250								KeepAlive

Host Station Connected Count:1

Ethernet Device (General)

MELSOFT Connection Module	-
SLMP Connection Module	-
UDP Connection Module	-
Active Connection Module	-
Unpassive Connection Module	-
Fullpassive Connection Module	-
MODBUS/TCP Connection Module	-

Ethernet Device (Mitsubishi Electric Corporation)
Ethernet Device (COGNEX)
Ethernet Device (Panasonic Industrial Devices SUNX)

[Outline]
MELSOFT Connection Module
[Specification]
Use when specify open method by MELSOFT connection

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.

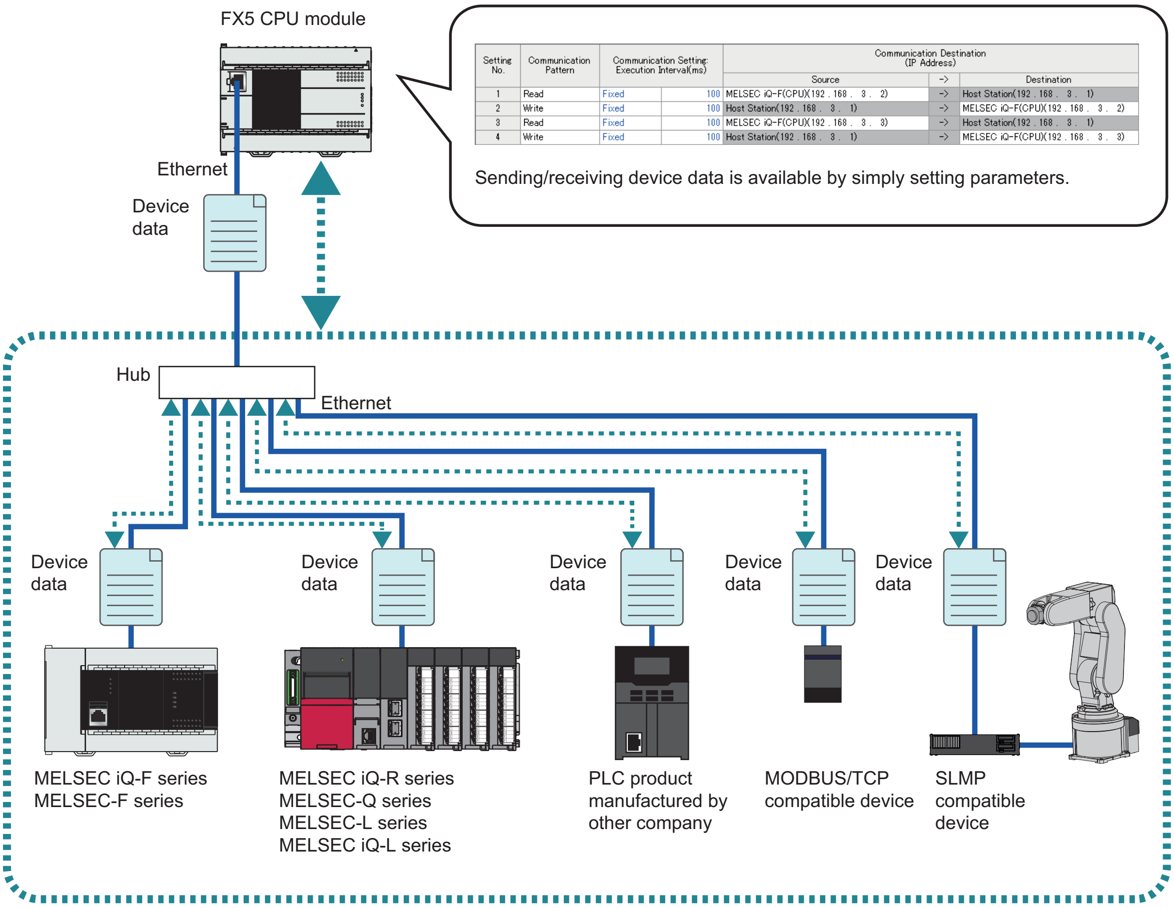
5 SIMPLE CPU COMMUNICATION FUNCTION

This function allows data communications between an Ethernet-equipped module and communication target devices.

The Ethernet port can be used to connect communication target devices and allows data communications between specified devices at the specified timing.

Simply setting the parameters with GX Works3 can build a communication system without programming.

The Ethernet-equipped module can be used to collect data from existing devices, which enables IoT implementation utilizing the existing equipment.



Setting No.	Communication Pattern	Communication Setting: Execution Interval(ms)		Communication Destination (IP Address)		
				Source	->	Destination
1	Read	Fixed	100	MELSEC iQ-F(CPU)(192 . 168 . 3 . 2)	->	Host Station(192 . 168 . 3 . 1)
2	Write	Fixed	100	Host Station(192 . 168 . 3 . 1)	->	MELSEC iQ-F(CPU)(192 . 168 . 3 . 2)
3	Read	Fixed	100	MELSEC iQ-F(CPU)(192 . 168 . 3 . 3)	->	Host Station(192 . 168 . 3 . 1)
4	Write	Fixed	100	Host Station(192 . 168 . 3 . 1)	->	MELSEC iQ-F(CPU)(192 . 168 . 3 . 3)

Sending/receiving device data is available by simply setting parameters.

Point

- The following are the maximum numbers of communication target devices that can be connected.

- FX5S/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. (☞ Page 46 Communication via router)

Connectable devices

Manufacturer	**Applicable model**
Mitsubishi Electric Corporation	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-F (Ethernet block/adapter)
OMRON Corporation	Supported model for SYSMAC CJ/CP series
KEYENCE CORPORATION	Supported model for KV series
Panasonic Corporation	Supported models for FP7 series and FP0H series
Siemens AG	Supported model for S7 series
Others	SLMP-compatible device (QnA-compatible 3E frame)[*1]
	MODBUS/TCP-compatible device

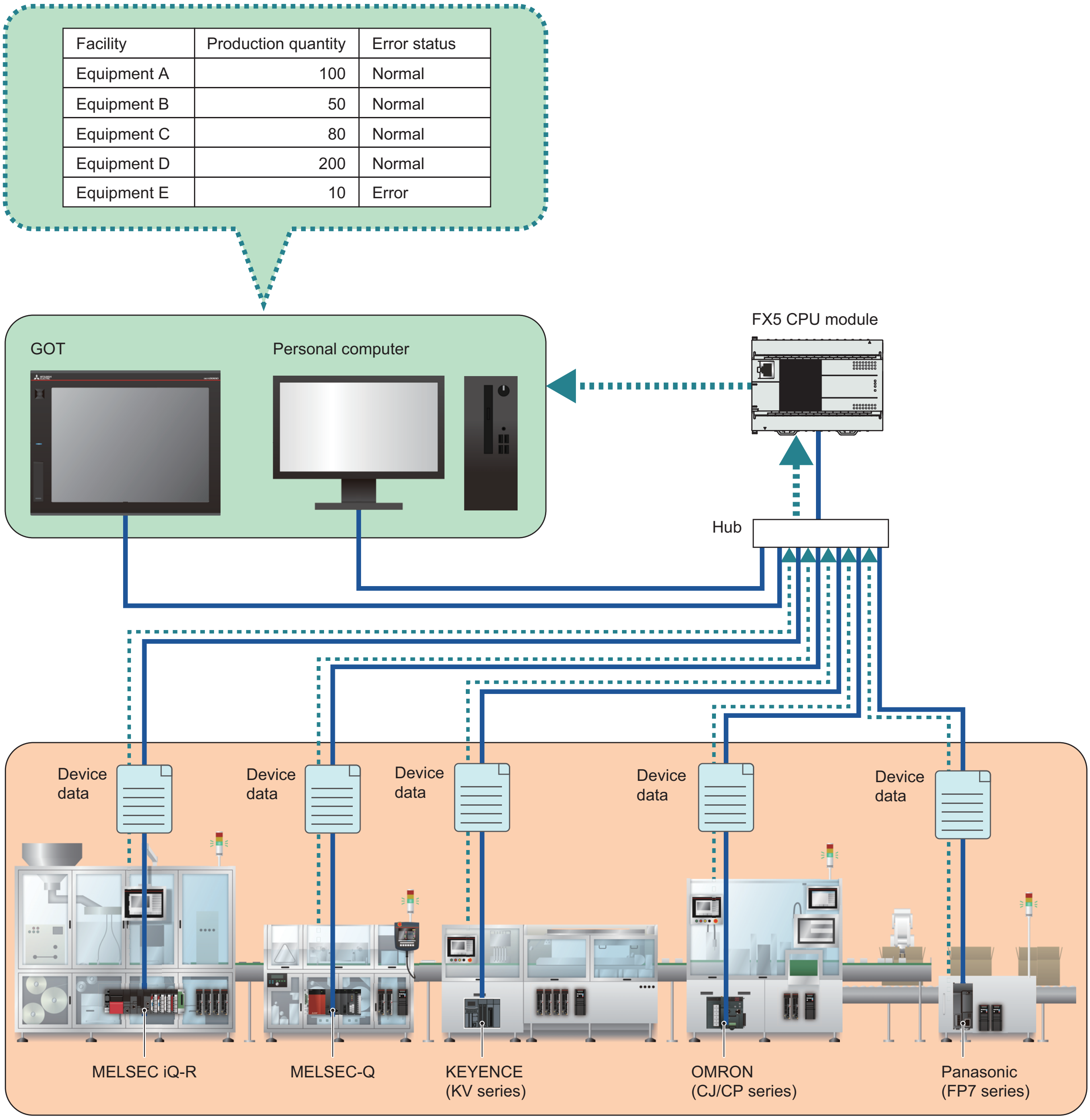
*1 To connect the QJ71E71-100 with the module, set the device type to "SLMP-compatible device (QnA-compatible 3E frame)". (☞ Page 53 Device type)

5.1 Utilization Examples

This section provides utilization examples of the simple CPU communication function.

Enabling operation monitoring of production lines

For a production line using multiple pieces of equipment, the FX5 CPU module can collect data, such as the production status and errors, from supported devices mounted on the equipment. This enables a single FX5 CPU module to monitor the operating status of the production line.

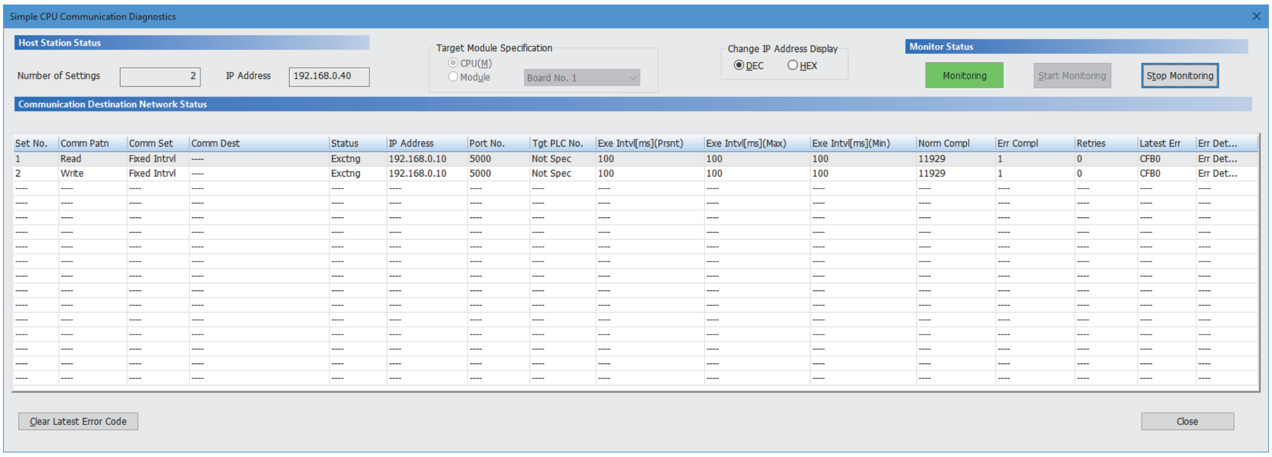


Facility	Production quantity	Error status
Equipment A	100	Normal
Equipment B	50	Normal
Equipment C	80	Normal
Equipment D	200	Normal
Equipment E	10	Error

Batch-monitoring the communication status with the diagnostic function

A dedicated window for the simple CPU communication diagnostics can be used to batch-monitor the communication status of simple CPU communications being executed.

Settings, errors, and other information can be checked and utilized for means such as troubleshooting due to communication errors.



Number of Settings: 2

IP Address: 192.168.0.40

Target Module Specification: ● CPU(M) ○ Module Board No. 1

Change IP Address Display: ● DEC ○ HEX

Set No.	Comm Patn	Comm Set	Comm Dest	Status	IP Address	Port No.	Tgt PLC No.	Exe Intvl[ms](Prsnt)	Exe Intvl[ms](Max)	Exe Intvl[ms](Min)	Norm Compl	Err Compl	Retries	Latest Err	Err Det...
1	Read	Fixed Intrvl	----	Exctng	192.168.0.10	5000	Not Spec	100	100	100	11929	1	0	CFB0	Err Det...
2	Write	Fixed Intrvl	----	Exctng	192.168.0.10	5000	Not Spec	100	100	100	11929	1	0	CFB0	Err Det...
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

5.2 Specifications List

This section describes the specifications for the simple CPU communication function.

Item	Specification		
	FX5S/FX5UJ CPU module	**FX5U/FX5UC CPU module**	**Ethernet module**
Maximum number of connectable modules (per system)	8 devices	16 devices	32 devices
Number of settings[*1]	1 to 16	1 to 32	1 to 32
Devices that can be specified	Bit device, word device		
Number of device points	Maximum of 8192 words (total number for all settings)		
Communication pattern	Read, write		
Communication setting	Fixed interval, On request[*2]		
Execution interval (ms)	10ms to 65535ms		
Connection	Does not occupy any connection.		Occupies one connection per communication target device (per group)
Own station port number	Duplication is not allowed.[*3]		Duplication is allowed.

*1 The number of read and write settings allowed.

*2 For the supported versions of each model, refer to the following.
☞ Page 936 Added and Changed Functions

*3 If a MODBUS/TCP-compatible or SIEMENS S7-series device is used with a CPU module, duplication of the own station port number can be set with conditions. For details, refer to the following.
☞ Page 87 Duplication of own station port numbers

Device type

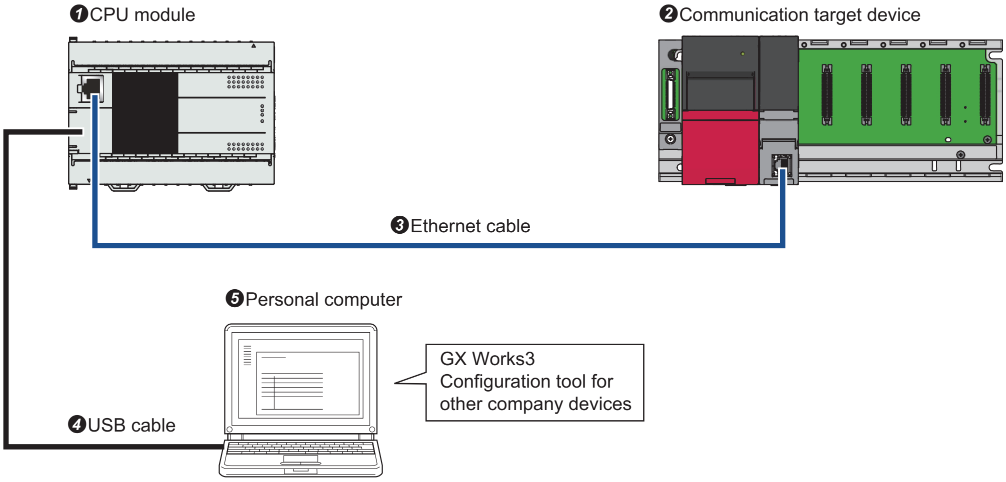
○: Supported, —: Not supported

Communication target device		Own station			Communication specification
Applicable model	Model name	FX5S CPU module	FX5UJ/FX5U/FX5UC CPU module	Ethernet module	
MELSEC iQ-R series (built-in Ethernet)	RnCPU, RnENCPU (CPU part), RnPCPU, RnSFCPU, RnPSFCPU	○	○	—	MELSOFT connection (UDP)
MELSEC-Q series (built-in Ethernet)	QnUD(P)VCPU, QnUDE(H)CPU	○	○	—	
MELSEC-L series (built-in Ethernet)	Built-in Ethernet port LCPU	○	○	—	
MELSEC iQ-F series (built-in Ethernet)	FX5S CPU module, FX5UJ CPU module, FX5U CPU module, FX5UC CPU module	○	○	○	
MELSEC iQ-F series (Ethernet module)	FX5-ENET	○	○	○	
MELSEC iQ-L series (built-in Ethernet)	LnHCPU	○	○	—	
MELSEC-F series (Ethernet block/adapter)	FX3U-ENET-ADP, FX3U-ENET-L	○	○	—	MC protocol (A-compatible 1E frame) [UDP, binary] • Batch read in units of words (01H) • Batch write in units of words (03H)
CJ/CP series	Supported model for SYSMAC CJ/CP series	○	○	—	FINS
KV series	Supported model for KV series	○	○	—	SLMP (QnA-compatible 3E frame) [UDP, binary] • Read (0401H) • Write (1401H)
FP7 series	Supported model for FP7 series	○	○	—	MEWTOCOL-7
FP0H series	Supported model for FP0H series	○	○	—	MEWTOCOL-COM
S7 series	Supported model for S7 series	○	○	—	S7 Communication
SLMP-compatible device (QnA-compatible 3E frame)	SLMP-compatible device (No serial number)	○	○	—	SLMP (QnA-compatible 3E frame) [UDP, binary] • Read (0401H) • Write (1401H)
	SLMP3E frame-compatible device	○	○	—	
MODBUS/TCP-compatible device	MODBUS/TCP slave device	○	○	○	MODBUS/TCP

5.3 Procedures for Use

Devices and software to be used

This section describes devices and software to be used with the simple CPU communication function.

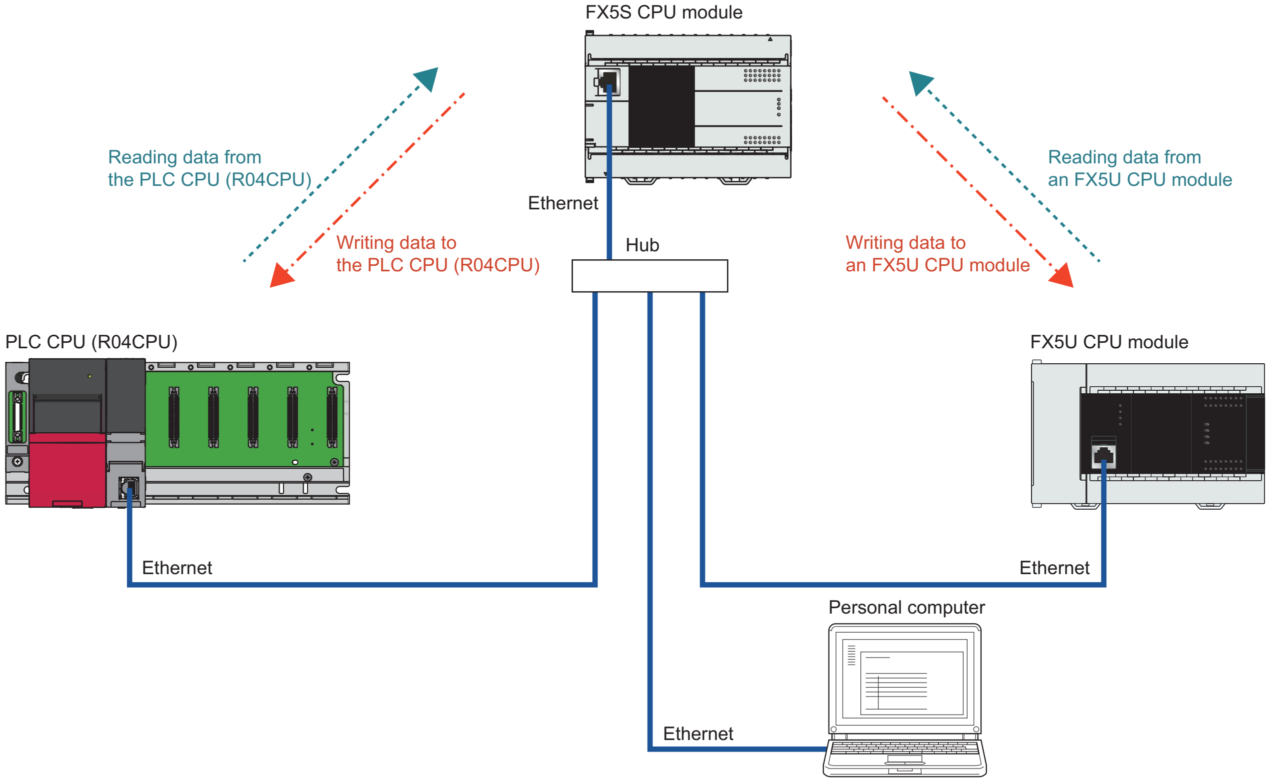


No.	Name	Description
❶	CPU module[*1]	FX5 CPU module or FX5 CPU module + Ethernet module
❷	Communication target device	Connectable device (Page 53 Device type)
❸	Ethernet cable	Standard Ethernet cable
❹	USB cable[*2]	Standard USB cable
❺	Personal computer	Personal computer with the following software installed • GX Works3[*3] • Configuration tool for other company's device (when connecting to other company's device)

*1 Use the latest firmware version on CPU modules.
*2 Usable by the FX5S/FX5UJ CPU module.
*3 Use the latest version.

Setting example 1: Connect three Mitsubishi Electric programmable controllers

This section describes a setting example for connecting the FX5S CPU module (own station) with the FX5U CPU module and a programmable controller CPU (communication target devices) to read and write device data at a fixed interval (100ms).



Setting item	Setting details	
	FX5S CPU module to FX5U CPU module	FX5S CPU module to Programmable controller CPU
Execution Interval	Fixed (100ms)	Fixed (100ms)
Read	Source: M0 to M15 Destination: M400 to M415	Source: M16 to M31 Destination: M416 to M431
Write	Source: D100, D101 Destination: D1100, D1101	Source: D102, D103 Destination: D1102, D1103

Usage flow

Based on the setting example 1, the following shows the flow for using the simple CPU communication function.

1. Set the IP addresses to the CPU modules. (☞ Page 56 Setting the IP address)
2. Set the simple CPU communication setting. (☞ Page 56 Setting the simple CPU communication)
3. Write parameters to the CPU modules.
4. Power off and on or reset the CPU modules to enable the written parameters.
5. Check if data is being read and written correctly. (☞ Page 58 Checking the communication status)

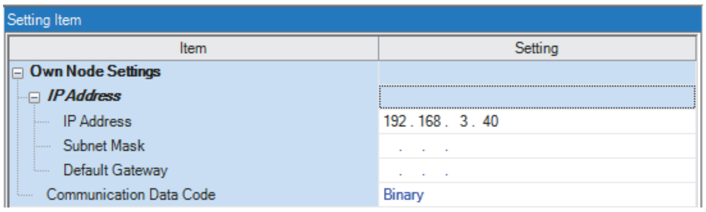
Operating procedure

The following describes the operating procedure for setting example 1.

■Setting the IP address

Create a project for each CPU module and set the IP addresses.

Navigation window ⇨ [Parameter] ⇨ Module name ⇨ [Module Parameter] ⇨ [Ethernet Port] ⇨ [Basic Settings] ⇨ [Own Node Settings]



Setting Item

Item	Setting
Own Node Settings	
IP Address	
IP Address	192 . 168 . 3 . 40
Subnet Mask	. . .
Default Gateway	. . .
Communication Data Code	Binary

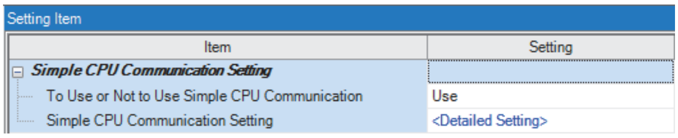
CPU module	IP address
FX5S CPU module	192.168.3.40
FX5U CPU module	192.168.3.250 (default)
Programmable controller CPU	192.168.3.39 (default)

■Setting the simple CPU communication

Set the simple CPU communication for the project of the FX5S CPU module (own station).

1. Select "Use" for "To Use or Not to Use Simple CPU Communication".

Navigation window ⇨ [Parameter] ⇨ [FX5S CPU] ⇨ [Module Parameter] ⇨ [Ethernet Port] ⇨ [Application Settings] ⇨ [Simple CPU Communication Setting]



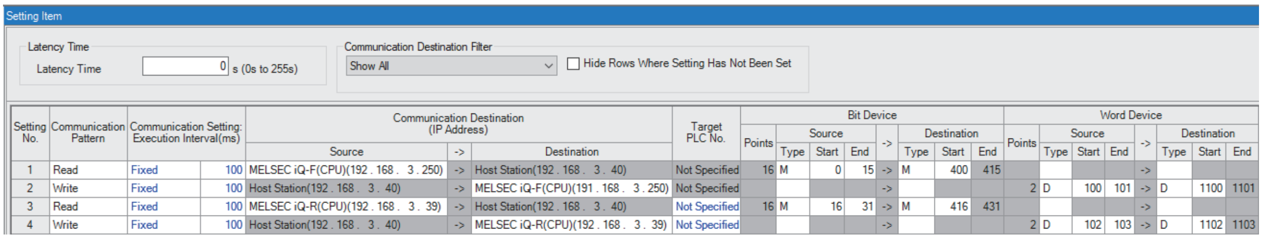
Setting Item

Item	Setting
Simple CPU Communication Setting	
To Use or Not to Use Simple CPU Communication	Use
Simple CPU Communication Setting	<Detailed Setting>

2. Click "Detailed Setting" in "Simple CPU Communication Setting".

3. Configure the settings that are used to read and write device data from the FX5S CPU module (own station) to the communication target devices.

Configure the communication destinations (source and destination) on a one-to-one basis. Also, use the default values for settings that are not listed below.


Setting Item

Latency Time

Latency Time: 0 s (0s to 255s)

Communication Destination Filter: Show All

☐ Hide Rows Where Setting Has Not Been Set

Setting No.	Communication Pattern	Communication Setting: Execution Interval(ms)		Communication Destination (IP Address)			Target PLC No.	Bit Device								Word Device							
				Source	->	Destination		Points	Source			->	Destination			Points	Source			->	Destination		
									Type	Start	End		Type	Start	End		Type	Start	End		Type	Start	End
1	Read	Fixed	100	MELSEC iQ-F(CPU)(192 . 168 . 3 . 250)	->	Host Station(192 . 168 . 3 . 40)	Not Specified	16	M	0	15	->	M	400	415					->			
2	Write	Fixed	100	Host Station(192 . 168 . 3 . 40)	->	MELSEC iQ-F(CPU)(191 . 168 . 3 . 250)	Not Specified					->				2	D	100	101	->	D	1100	1101
3	Read	Fixed	100	MELSEC iQ-R(CPU)(192 . 168 . 3 . 39)	->	Host Station(192 . 168 . 3 . 40)	Not Specified	16	M	16	31	->	M	416	431					->			
4	Write	Fixed	100	Host Station(192 . 168 . 3 . 40)	->	MELSEC iQ-R(CPU)(192 . 168 . 3 . 39)	Not Specified					->				2	D	102	103	->	D	1102	1103

Setting No.	Communication Pattern	Communication Destination (IP Address)		Bit Device		Word Device	
		Source	**Destination**	**Source**	**Destination**	**Source**	**Destination**
1	Read	• Device Type: MELSEC iQ-F (built-in Ethernet) • IP Address: 192.168.3.250	—	• Type: M • Start: 0 • End: 15	Type: M • Start: 400	—	—
2	Write	—	• Device Type: MELSEC iQ-F (built-in Ethernet) • IP Address: 192.168.3.250	—	—	• Type: D • Start: 100 • End: 101	• Type: D • Start: 1100
3	Read	• Device Type: MELSEC iQ-R (built-in Ethernet) • IP Address: 192.168.3.39	—	Type: M Start: 16 • End: 31	Type: M • Start: 416	—	—
4	Write	—	• Device Type: MELSEC iQ-R (built-in Ethernet) • IP Address: 192.168.3.39	—	—	Type: D Start: 102 • End: 103	Type: D • Start: 1102

4. Write the parameters to the FX5S CPU module (own station).

■Starting the simple CPU communication

To start the simple CPU communication, power off and on or reset the FX5S CPU module (own station) to enable the written parameters.

■Checking the communication status

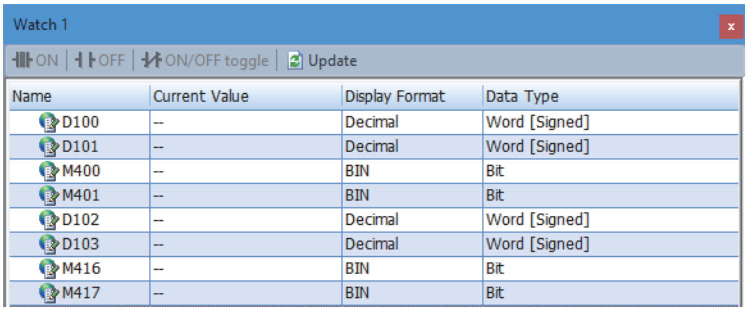
Check if communications are working as specified in the setting example. For checking the communication status of the simple CPU communication, refer to the following.

☞ Page 62 Checking the Simple CPU Communication Status

1. Check that the SD/RD LED flashes on the CPU module.

2. For all projects, register devices in the Watch window.

[View] ⇨ [Docking Window] ⇨ [Watch 1]



Name	Current Value	Display Format	Data Type
D100	--	Decimal	Word [Signed]
D101	--	Decimal	Word [Signed]
M400	--	BIN	Bit
M401	--	BIN	Bit
D102	--	Decimal	Word [Signed]
D103	--	Decimal	Word [Signed]
M416	--	BIN	Bit
M417	--	BIN	Bit

FX5S CPU module	FX5U CPU module	Programmable controller CPU
• D100 • D101 • M400 • M401 • D102 • D103 • M416 • M417	• D1100 • D1101 • M0 • M1	D1102 D1103 M16 M17

3. From the Watch window for each project, perform operations and check the communication status.

- Write check

- Modify the current value on the own station (FX5S CPU module) and then confirm that the current value is changed on the communication target devices (FX5U CPU module and programmable controller CPU).

- Read check

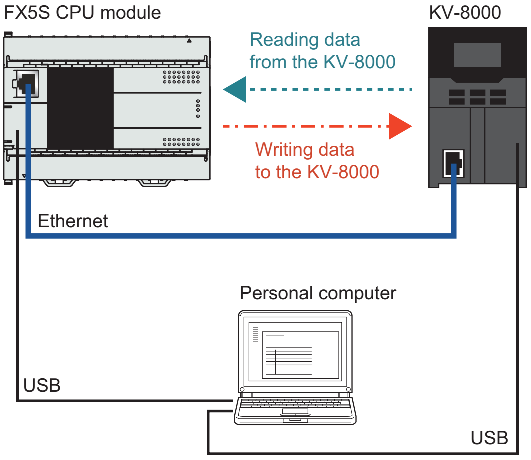
- Modify the current value on the communication target devices (FX5U CPU module and programmable controller CPU) and then confirm that the current value is changed on the own station (FX5S CPU module).

For how to check devices in the Watch window, refer to the following manual.

📖 GX Works3 Operating Manual

Setting example 2: Connect another company's device (KV-8000)

This section provides a setting example for connecting the FX5S CPU module (own station) with the KV-8000 (communication target device) to read and write device data at a fixed interval (100ms).



Setting item	Communication target device
	KV-8000
Execution Interval	Fixed (100ms)
Read (Destination: FX5S CPU module)	Source: MR0 to MR15 Destination: M400 to M415
Write (Source: FX5S CPU module)	Source: D10, D11 Destination: DM100, DM101

Usage flow

The following shows the flow for using the simple CPU communication function. Details of the procedure are described using this setting example.

1. Set the IP addresses to the CPU modules. (☞ Page 60 Setting the IP address)
2. Set the simple CPU communication setting. (☞ Page 60 Setting the simple CPU communication)
3. Write parameters to the CPU modules.
4. On the communication target device, configure the settings for receiving communication.
5. Power off and on or reset the CPU modules to enable the written parameters.
6. Check if data is being read and written correctly. (☞ Page 61 Checking the communication status)

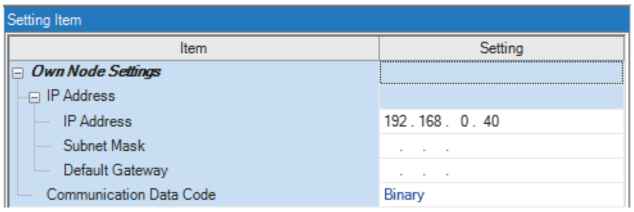
Operating procedure

The following describes the operating procedure for setting example 2.

■Setting the IP address

Create a project for each CPU module and set the IP addresses.

Navigation window ⇨ [Parameter] ⇨ Module name ⇨ [Module Parameter] ⇨ [Ethernet Port] ⇨ [Basic Settings] ⇨ [Own Node Settings]



Setting Item

Item	Setting
Own Node Settings	
IP Address	
IP Address	192 . 168 . 0 . 40
Subnet Mask	. . .
Default Gateway	. . .
Communication Data Code	Binary

CPU module	IP address
FX5S CPU module	192.168.0.40
KV-8000	192.168.0.10 (default)[1]

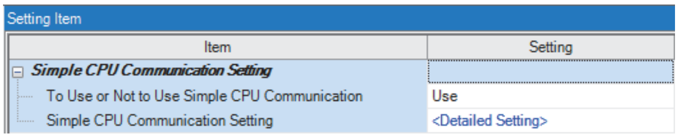
[1] For how to set the IP address, refer to the manual for the communication target device.

■Setting the simple CPU communication

Set the simple CPU communication for the project of the FX5S CPU module (own station).

1. Select "Use" for "To Use or Not to Use Simple CPU Communication".

Navigation window ⇨ [Parameter] ⇨ [FX5S CPU] ⇨ [Module Parameter] ⇨ [Ethernet Port] ⇨ [Application Settings] ⇨ [Simple CPU Communication Setting]



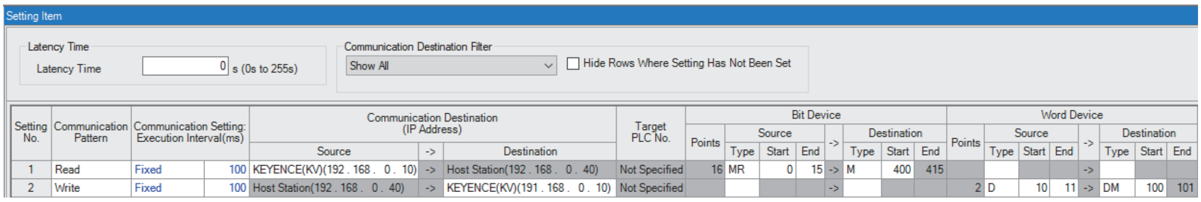
Setting Item

Item	Setting
Simple CPU Communication Setting	
To Use or Not to Use Simple CPU Communication	Use
Simple CPU Communication Setting	<Detailed Setting>

2. Click "Detailed Setting" in "Simple CPU Communication Setting".

3. Configure the settings that are used to read and write device data from the FX5S CPU module (own station) to the communication target devices.

Configure the communication destinations (source and destination) on a one-to-one basis. Also, use the default values for settings that are not listed below.



Setting Item

Latency Time

Latency Time: 0 s (0s to 255s)

Communication Destination Filter: Show All

☐ Hide Rows Where Setting Has Not Been Set

Setting No.	Communication Pattern	Communication Setting: Execution Interval(ms)		Communication Destination (IP Address)			Target PLC No.	Bit Device								Word Device							
				Source	->	Destination		Points	Source			->	Destination			Points	Source			->	Destination		
									Type	Start	End		Type	Start	End		Type	Start	End		Type	Start	End
1	Read	Fixed	100	KEYENCE(KV)(192 . 168 . 0 . 10)	->	Host Station(192 . 168 . 0 . 40)	Not Specified	16	MR	0	15	->	M	400	415					->			
2	Write	Fixed	100	Host Station(192 . 168 . 0 . 40)	->	KEYENCE(KV)(191 . 168 . 0 . 10)	Not Specified					->				2	D	10	11	->	DM	100	101

Setting No.	Communication Pattern	Communication Destination (IP Address)		Bit Device		Word Device	
		Source	Destination	Source	Destination	Source	Destination
1	Read	• Device Type: KEYENCE (KV series) • IP Address: 192.168.0.10 • TCP/UDP: UDP • Port No.: 5000 • Host Station Port No.: 8000	—	• Type: MR • Start: 0 • End: 15	• Type: M • Start: 400	—	—
2	Write	—	• Device Type: KEYENCE (KV series) • IP Address: 192.168.0.10 • TCP/UDP: UDP • Port No.: 5000 • Host Station Port No.: 8500	—	—	• Type: D • Start: 10 • End: 11	• Type: DM • Start: 100

4. Write the parameters to the FX5S CPU module (own station).

■Setting the communication target device

If the method for communicating with the communication target device is not MELSOFT UDP, set up communication via a protocol that matches the simple CPU communication method. For the communication method for each device, refer to the following.

☞ Page 53 Device type

Point

In this example, since the setting configured in "Simple CPU Communication Setting" fits the MC protocol port number (UDP) for KV-8000, the default setting can be used for communication.

■Starting the simple CPU communication

To start the simple CPU communication, power off and on or reset the FX5S CPU module (own station) to enable the written parameters.

■Checking the communication status

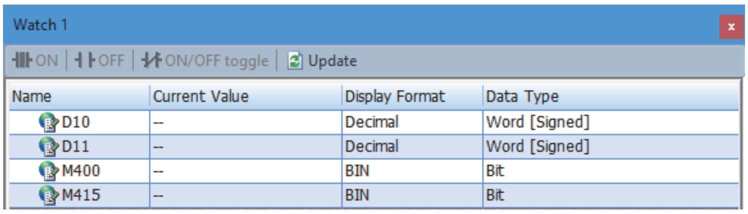
Check if communications are working as specified in the setting example. For checking the communication status of the simple CPU communication, refer to the following.

☞ Page 62 Checking the Simple CPU Communication Status

1. Check that the SD/RD LED flashes on the CPU module.

2. For all projects, register devices in the Watch window.

[View] ⇨ [Docking Window] ⇨ [Watch 1]



Name	Current Value	Display Format	Data Type
D10	--	Decimal	Word [Signed]
D11	--	Decimal	Word [Signed]
M400	--	BIN	Bit
M415	--	BIN	Bit

FX5S CPU module	KV-8000[*1]
• D10 • D11 • M400 • M415	• DM100 • DM101 • MR0 • MR15

[*1] For how to monitor devices, refer to the manual for the communication target device.

3. From the Watch window for each project, perform operations and check the communication status.

- Write check

- Modify the current value on the own station (FX5S CPU module) and then confirm that the current value is changed on the communication target device (KV-8000).

- Read check

- Modify the current value on the communication target device (KV-8000) and then confirm that the current value is changed on the own station (FX5S CPU module).

For how to check devices in the Watch window, refer to the following manual.

📖 GX Works3 Operating Manual

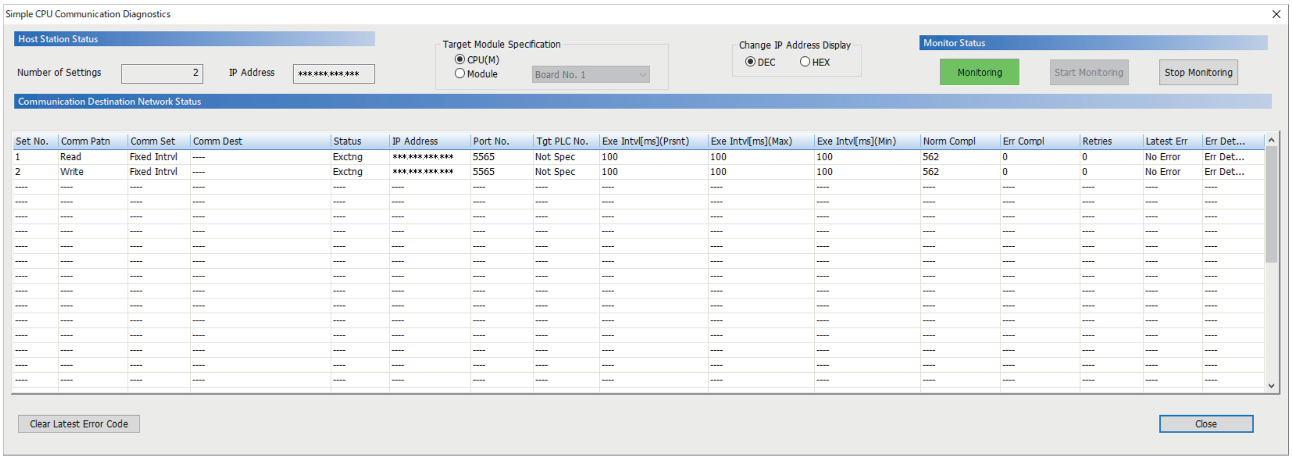
5.4 Checking the Simple CPU Communication Status

The simple CPU communication status can be checked with "Simple CPU Communication Diagnostics" or from the device.

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]



Number of Settings: 2

IP Address: ***.***.***.***

Target Module Specification: ● CPU(M) ○ Module Board No. 1

Change IP Address Display: ● DEC ○ HEX

Set No.	Comm Patn	Comm Set	Comm Dest	Status	IP Address	Port No.	Tgt PLC No.	Exe Intvl[ms](Prsnt)	Exe Intvl[ms](Max)	Exe Intvl[ms](Min)	Norm Compl	Err Compl	Retries	Latest Err	Err Det...
1	Read	Fixed Intrvl	----	Exctng	***.***.***.***	5565	Not Spec	100	100	100	562	0	0	No Error	Err Det...
2	Write	Fixed Intrvl	----	Exctng	***.***.***.***	5565	Not Spec	100	100	100	562	0	0	No Error	Err Det...
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

The module to be diagnosed can be switched with "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/on request[*1]) set with the parameters.
Comm Dest	Displays the communication destination set with the parameters. The communication destination is displayed only when the RJ71EN71, RnENCPU (network part), or FX5-ENET performed simple CPU communications.
Status	Displays the simple CPU communication status (preparing, waiting for request[*1], 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 PLC number set with the parameters.
Exe Intvl [ms] (Prsnt)	Displays the current value, the maximum value, and the minimum value of the execution interval in increments of ms. The values will not be updated during a retry or error monitoring. They will be "0" while communication is stopped.
Exe Intvl [ms] (Max)	
Exe Intvl [ms] (Min)	
Norm Compl	Displays the accumulated count of communications that has been completed successfully, completed with an error, and retried. • 0 to 4294967295: Accumulated count[*2]
Err Compl	
Retries	
Latest Err	Displays the error code of the latest error that 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. • FX5S/FX5UJ CPU module: Simple CPU communication error code (SD10412 to SD10427) • FX5U/5UC CPU module: Simple CPU communication error code (SD10412 to SD10443) • Ethernet module: Simple CPU error code (Un\G416 to Un\G447)

*1 For the supported versions of each model, refer to the following.
☞ Page 936 Added and Changed Functions

*2 When the count exceeds 4294967295, counting is continued from 1 again.

Checking with the device

■CPU module

The simple CPU communication status can be checked with the storage status of the corresponding setting number on the following special devices.

○: Supported, ×: Not supported

Item		Communication setting		Device No.	Remarks
		Fixed interval	On request[*1]		
Simple CPU communication start request (for communication at request)		×	○	SD10350, SD10351	SD10350.b0: Setting No.1 to SD10350.b15: Setting No.16 SD10351.b0: Setting No.17 to SD10351.b15: Setting No.32[*2]
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[*2]
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[*2]
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[*2]
	1H: Preparing	○	○		
	2H: Waiting for request	×	○		
	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[*2]
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[*2]
Simple CPU communication error response code		○	○	SD10476 to SD10507	SD10476: Setting No.1 to SD10491: Setting No.16 SD10492: Setting No.17 to SD10507: Setting No.32[*2]

*1 For the supported versions of each model, refer to the following.
☞ Page 936 Added and Changed Functions

*2 The FX5U/FX5UC CPU modules supports the setting number.
☞ Page 936 Added and Changed Functions

■Ethernet module

The simple CPU communication status can be checked with the storage status of the corresponding setting number in the following buffer memory areas.

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 the 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 Intrvl"

[When the own station is a CPU module]

When the simple CPU communication status (SD10380 to SD10411) of the corresponding setting number on a special device is either of the following, communications can be stopped by turning off and on the bit of the corresponding setting number in the simple CPU communication stop request (SD10352, SD10353). (If the status is other than the following, the stop request is invalid.)

- 3H: Communicating
- 5H: Retry being executed
- 6H: Monitoring

When the simple CPU communication status (SD10380 to SD10411) of the corresponding setting number on a special device is "4H: Communication stop", communications can be restarted by turning off and on the bit of the corresponding setting number in the simple CPU communication restart request (SD10354, SD10355). (If the status is other than "4H: Communication stop", the restart request is invalid.)

[When the own station is an Ethernet module]

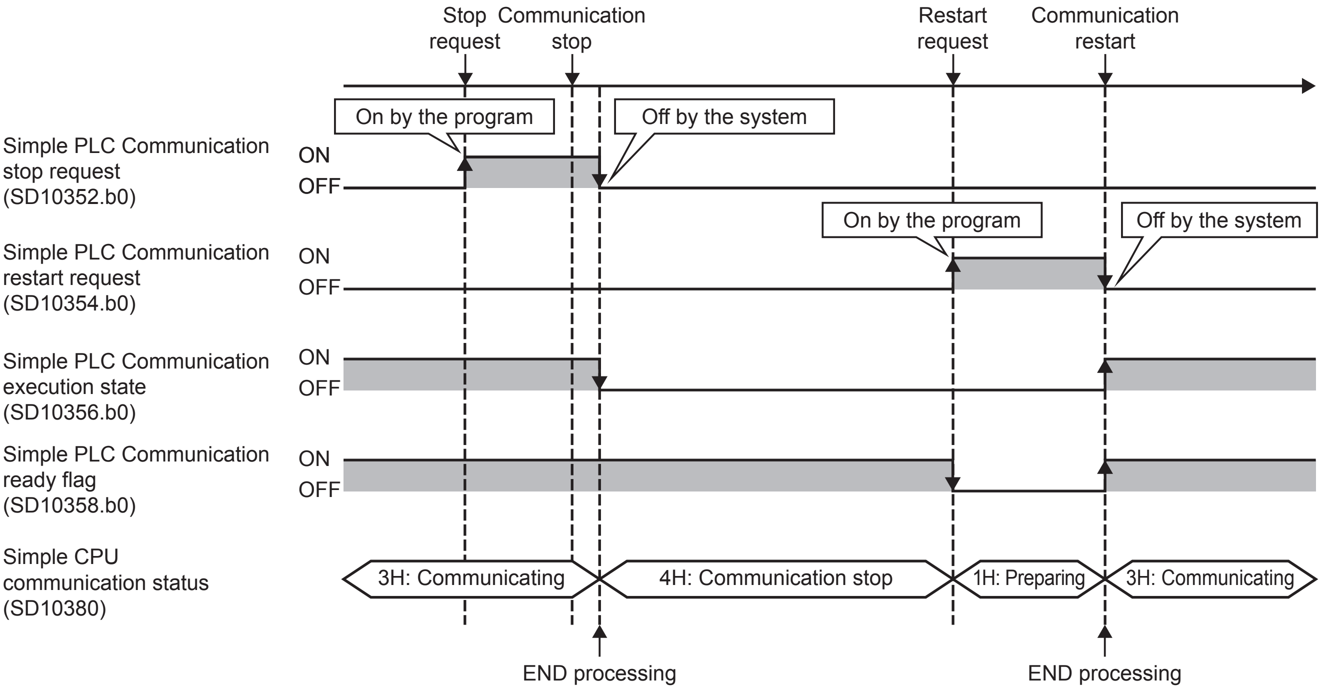
When the simple CPU communication status (Un\G352 to Un\G383) of the corresponding setting number in the buffer memory is either of the following, communications can be stopped by turning off and on the bit of the corresponding setting number in the request to stop fixed interval communication (Un\G304, Un\G305). (If the status is other than the following, the stop request is invalid.)

- 3H: Communicating
- 5H: Retry being executed
- 6H: Monitoring

When the simple CPU communication status (Un\G352 to Un\G383) of the corresponding setting number in the buffer memory is "4H: Communication stop", communications can be restarted by turning off and on the bit of the corresponding setting number in the request to restart fixed interval communication (Un\G308, Un\G309). (If the status is other than "4H: Communication stop", the restart request is invalid.)

Ex.

Operation at communication stop/restart for setting No.1 (when the own station is a CPU module)



■Starting communication when the communication setting is "On Request"

The version supporting "On Request" of the communication setting varies depending on the models. Refer to the following.

☞ Page 936 Added and Changed Functions

[When the own station is a CPU module]

In the following cases, when the bit of the corresponding setting number is turned off and on in Request to start communication at request (SD10350, SD10351), communication can be started. (In cases other than the following, start requests are invalid, or an error occurs.)

- The preparation completion flag (SD10358, SD10359) of the corresponding setting number is on.
- The simple CPU communication status (SD10380 to SD10411) of the corresponding setting number is "2H: Waiting for the request".

However, if Request to start communication at request (SD10350, SD10351) remains on when the simple CPU communication status (SD10380 to SD10411) is "1H: Preparing", communication starts when the simple CPU communication status changes to "2H: Waiting for the request".

After data is transmitted/received, the communication is stopped. When starting communication the next time, turn off and on the bit of the corresponding setting number again in Request to start communication at request (SD10350, SD10351).

[When the own station is an Ethernet module]

In the following cases, when the bit of the corresponding setting number is turned off and on in Request to start communication at request (Un\G300, Un\G301), communication can be started. (In cases other than the following, start requests are invalid, or an error occurs.)

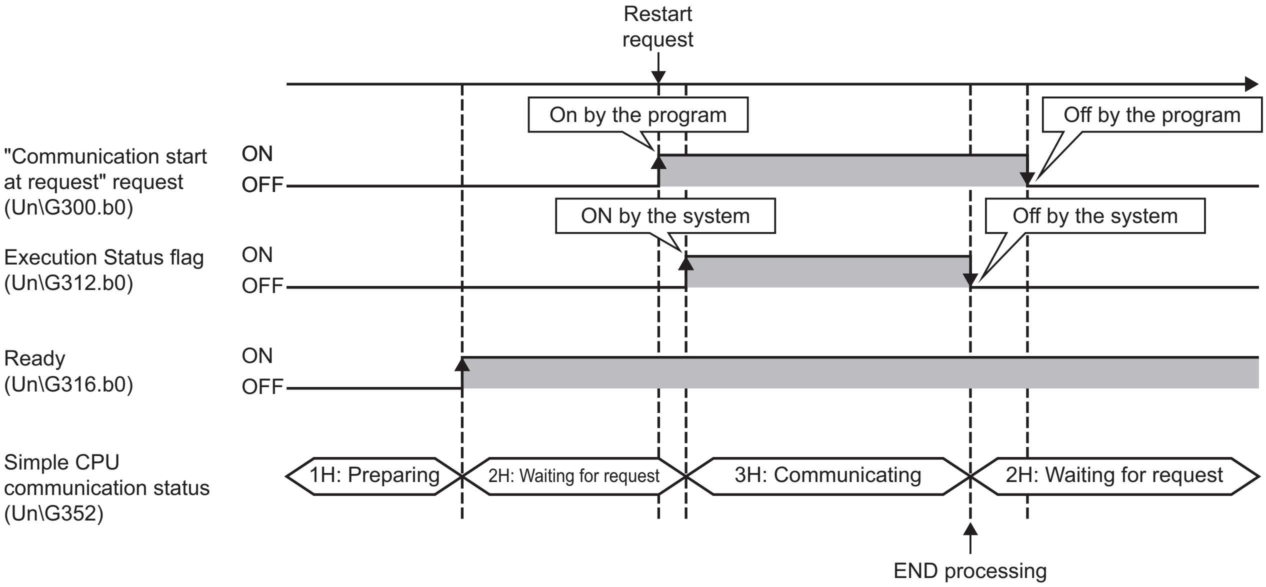
- Ready (Un\G316 or Un\G317) of the corresponding setting number is on.
- The simple CPU communication status (Un\G352 to Un\G383) of the corresponding setting number is "2H: Waiting for the request".

However, if Request to start communication at request (Un\G300 or Un\G301) remains on when the simple CPU communication status (Un\G352 to Un\G383) is "1H: Preparing", communication starts when the simple CPU communication status changes to "2H: Waiting for the request".

After data is transmitted/received, the communication is stopped. When starting communication the next time, turn off and on the bit of the corresponding setting number again in Request to start communication at request (Un\G300, Un\G301).

Ex.

Operation at communication start for setting No.1 (when the own station is an Ethernet module)



5.5 Specification Details

This section describes the details of the simple CPU communication settings.
In the simple CPU communication settings, set the communication destinations on a one-to-one basis. In order from setting No.1, set the communication pattern, communication setting, execution interval, communication destination, device, and other settings.

Number of settings

The number of settings is as follows.

Setting type	Number of settings
Setting No.	• FX5S/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	Set whether to read or write the data. • Read: Read the data of the specified destination device (transmission source) to the specified device of the own station (transmission destination). • Write: Write the data of the specified device of the own station (transmission source) to the specified destination device (transmission destination).	• 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	Set this item to communicate data periodically. • Fixed: Data are communicated between the devices at a specified execution interval. • On Request: Device data are communicated by turning off and on Request to start communication at request of the setting number whose communication setting is set to be on-request.	• Fixed • On Request[*1][*2] (Default: Fixed)
	Execution Interval	Set the execution interval of communications when "Fixed" is set for the communication setting.	10ms to 65535ms (in increments of 1ms) (Default: 100ms)

[*1] For the supported versions of each model, refer to the following.
☞ Page 936 Added and Changed Functions

[*2] When selecting communications on request, programming is required for sending requests. For program examples, refer to the following.
☞ Page 83 Program Examples of Communication on Request

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 812 When the Simple CPU Communication Function is Used)
- If latency time is set, data communication is started after the latency time has passed. (☞ Page 83 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 the communication of data is executed by the END processing.

Registration of communication destination

Set and register devices for both transmission source and transmission destination. The following number of devices can be registered as communication destinations.

- FX5S/FX5UJ CPU module: 8 devices
- FX5U/FX5UC CPU module: 16 devices
- Ethernet module: 32 devices

Item			Description	Setting range
Communication Destination (IP Address)	Source	Device Type	Specify the device type of the communication destination.	• 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	Enter the IP address of the communication destination.	☞ Page 69 IP Address, TCP/UDP, Port No., and Host Station Port No. ☞ Page 71 Target PLC No.
		TCP/UDP	Set the protocol of the communication destination.	
		Port No.	Set the port number of the communication destination.	
		Host Station Port No.	Set the port number of the own station.	
		Option (Hexadecimal)	Set the module ID of the communication destination.	
		Target PLC No.	Set the target number of the communication destination.	
	Destination	Device Type	Same as each item of the transmission source	
		IP Address		
		TCP/UDP		
		Port No.		
		Host Station Port No.		
		Option (Hexadecimal)		
		Target PLC No.		

■Device Type

Select the device type of the communication destination. For details, refer to the following.

☞ Page 53 Device type

■IP Address, TCP/UDP, Port No., and Host Station Port No.

Set the IP address of the communication destination, communication protocol, port number, and own station port number.

Device Type	IP Address	TCP/ UDP*1	Port No.	Host Station Port No.
MELSEC iQ-R (built-in Ethernet)	0.0.0.1 to 223.255.255.254	UDP	5006	A number between 61696 and 65534 is dynamically assigned. When the IP address of the communication destination is the same as that of MELSEC iQ-F (Ethernet module), the same own station port number 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 The own station port number of MELSEC iQ-F (Ethernet module) can be duplicated. For a MODBUS/TCP-compatible device or SIEMENS S7 series device with the same IP address, the own station port number can be duplicated.*3
SLMP-compatible device (QnA-compatible 3E frame)*2		UDP	1 to 65534	
OMRON (CJ/CP series)*4		UDP		
KEYENCE (KV series)		UDP		
Panasonic (FP7 series)		UDP		
Panasonic (FP0H series)		UDP		
MODBUS/TCP-compatible device		TCP		
SIEMENS S7 series		TCP	102	

*1 The protocols that support the device types are displayed. (Fixed)

*2 When the communication destination is the QJ71E71-100, set the port number to enable auto open UDP port (default: 5000).

*3 For the supported versions of each model, refer to the following.
☞ Page 936 Added and Changed Functions

*4 To configure multiple settings (write/read) for the external device, set FINS/UDP with OMRON (CJ/CP series) as follows.
· IP address conversion: Automatic generation method (dynamic)
· Destination IP address dynamic change: The destination IP address is dynamically changed.

Point

- When the communication destination is an SLMP-compatible device (QnA-compatible 3E frame), a parameter error will occur 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 for 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.
- Own station 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.
- When specifying the same communication destination in multiple settings, set different own station port numbers. However, the same own station port number can be set if the conditions are met, such as the device types or IP addresses being identical. (☞ Page 70 Group communications for the CPU module)
- Set the own station port number for the simple CPU communication function so that the own station port number and functions such as socket communication, SLMP communication, MODBUS/TCP functions, communication protocol support functions, and web server functions do not overlap with each other.
- When using the simple CPU communication function and the following functions simultaneously, do not specify the own station port number for the function to be used.

[For TCP communications]

- File transfer function (FTP client/server): 20 (14H), 21 (15H)
- Web server function: 80 (50H)*5*6
- MODBUS/TCP function (slave station): 502 (1F6H)*5

[For UDP communications]

- Time setting function (SNTP client): 123 (7BH)
- SLMP function: 61440 (F000H), 61441 (F001H)
- CC-Link IE Field Network Basic: 61450 (F00AH)

*5 The port number can be changed. (Default: 80 (file transfer function), 502 (MODBUS/TCP function (slave station)))

*6 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.

Specifications of each Ethernet-equipped module

The CPU module and Ethernet module differ in the following specifications.

Item	CPU module	Ethernet module
Connection	Does not occupy any connection.	Occupies one connection per communication target device (per group)
Own station port number	Duplication is not allowed.	Duplication is allowed.

■Group communications for the CPU module

When a CPU module uses the simple CPU communication function, multiple simple CPU communication settings can be combined into one group by satisfying the following conditions for communications that target the same MODBUS/TCP-compatible devices or SIEMENS programmable controllers (S7 series). (When multiple simple CPU communication settings are combined into one group, only a TCP connection will be established for the one group, despite having multiple settings.)

When the settings are made, groups are assigned to each setting number according to the simple CPU communication settings.

The maximum number of groups is 16 for the FX5S/FX5UJ CPU modules and 32 for the FX5U/FX5UC CPU modules.

When the following items of the simple CPU communication settings of devices are identical, the devices are regarded as the same group.

- Device Type
- IP Address
- TCP/UDP
- Port No.
- Host Station Port No.

For the supported versions of each model, refer to the following.

☞ Page 936 Added and Changed Functions

■Group communications for the Ethernet module

The Ethernet module communicates via groups. Groups are assigned to each setting number according to the simple CPU communication settings. The maximum number of groups is 32.

When the following items of the simple CPU communication settings of devices are identical, the devices are regarded as the same group.

- Device Type
- IP Address
- TCP/UDP
- Port No.
- Host Station Port No.

Option (Hexadecimal)

When the communication destination is "MODBUS/TCP-compatible device", set the setting value for the module ID of the MODBUS application header. For the supported versions of each model, refer to the following.

☞ Page 936 Added and Changed Functions

Item	Setting range
Option (Hexadecimal)	■When communicating with a MODBUS/TCP-compatible device 00H or FFH ■When communicating with a MODBUS RTU/ASCII-compatible device via a gateway device • 00H: Broadcast • 01H to F7H: Station number of a 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.	• Not Specified • PLC No.1 to PLC No.4 (Default: Not Specified)

Device setting

Set the device that sends/receives data to/from the communication destination.

Item		Description	Setting range
Bit Device	Points	Set the type, start number, and end number of the "Source" bit device and the type and start number of the "Destination" bit device. When these values are input, the number of points will be displayed automatically.	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 type, start number and end number of the "Source" word device and the type and start number of the "Destination" word device. When these values are input, the number of points will be displayed automatically.	
	Type		
	Start		
	End		

[1] For the devices that can be specified, refer to the following.
☞ Page 73 Devices that can be specified on the own station
☞ Page 73 When the communication destination is the MELSEC iQ-R (built-in Ethernet)
☞ Page 76 When the communication destination is the MELSEC-Q (built-in Ethernet)
☞ Page 76 When the communication destination is the MELSEC-L (built-in Ethernet)
☞ Page 74 When the communication destination is the MELSEC iQ-F (built-in Ethernet)
☞ Page 75 When the communication destination is the MELSEC iQ-F (Ethernet module)
☞ Page 74 When the communication destination is the MELSEC iQ-L (built-in Ethernet)
☞ Page 77 When the communication destination is the MELSEC-FX3 (Ethernet block/adapter)
☞ Page 77 When the communication destination is an SLMP-compatible device (QnA-compatible 3E frame)
☞ Page 77 When the communication destination is OMRON (CJ/CP series)
☞ Page 78 When the communication destination is KEYENCE (KV series)
☞ Page 78 When the communication destination is Panasonic (FP7 series)
☞ Page 79 When the communication destination is Panasonic (FP0H series)
☞ Page 79 When the communication destination is a MODBUS/TCP-compatible device
☞ Page 79 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 the communication destination, the number of points per setting is as follows.

Communication destination	Number of points per setting
• 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, the following table lists the device ranges when device types with the maximum number of points are used.

Type	Applicable device			
	Symbol	Range		
		FX5S/FX5U/FX5UC CPU module	FX5UJ 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 32767	0 to 7679	
	L	0 to 32767	0 to 7679	
	B	0H to 7FFFH	0H to 7FFFH	
	SB	0H to 7FFFH	0H to 7FFFH	
	SM	0 to 9999		
Word Device	D	0 to 7999		
	W	0H to 7FFFH	0H to 3FFH	
	R	0 to 32767		
	SW	0H to 7FFFH	0H to 3FFH	
	SD	0 to 11999		
	G	—		6400 to 8447[1]

[1] Un\G6400 to Un\G8447 can be used as devices for the simple CPU communication function.

Point

Set the devices within the device range specified in the 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, the following table lists the device ranges when device types with the maximum number of points are used.

Type	Applicable device		
	Symbol	Range	Remarks
Bit Device	X	0H to 2FFFH	—
	Y	0H to 2FFFH	—
	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 9A61FFH	—
	SW	0H to 9A61FFH	—
	SD	0 to 4095	—
	R	0 to 32767	Data is read or written following the "File Register Setting" of the communication destination.
	ZR	0 to 10027007	Data is read or written following the "File Register Setting" of the communication destination.
	RD	0 to 1048575	—

Point

Set the devices 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, the following table lists the device ranges when device types with the maximum number of points are used.

Type	Applicable device		
	Symbol	Range	Remarks
Bit Device	X	0H to 2FFFH	—
	Y	0H to 2FFFH	—
	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 D69FFH	—
	SW	0H to D69FFH	—
	SD	0 to 4095	—
	R	0 to 32767	Data is read or written following the "File Register Setting" of the communication destination.
	ZR	0 to 819199	Data is read or written following the "File Register Setting" of the communication destination.
	RD	0 to 1048575	—

Point

Set the devices 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, the following table lists the device ranges when device types with the maximum number of points are used.

Type	Applicable device			
	Symbol	Range		Remarks
		FX5S/FX5U/FX5UC CPU module	FX5UJ CPU module	
Bit Device	X	0 to 1777		Octal notation is used.
	Y	0 to 1777		Octal notation is used.
	M	0 to 32767	0 to 7679	—
	L	0 to 32767	0 to 7679	—
	B	0H to 7FFFH	0H to 7FFH	—
	SB	0H to 7FFFH	0H to 7FFH	—
	SM	0 to 9999		—
Word Device	D	0 to 7999		—
	W	0H to 7FFFH	0H to 3FFH	—
	SW	0H to 7FFFH	0H to 3FFH	—
	SD	0 to 11999		—
	R	0 to 32767		—

Point

Set the devices 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, the following table lists the device ranges when device types with the maximum number of points are used.

Type	**Applicable device**			
	Symbol	**Range**		**Remarks**
		When the FX5S/FX5U/FX5UC CPU module is connected	**When the FX5UJ 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 32767	0 to 7679	The same device range as that of the connected CPU module
	L	0 to 32767	0 to 7679	The same device range as that of the connected CPU module
	B	0H to 7FFFH	0H to 7FFH	The same device range as that of the connected CPU module
	SB	0H to 7FFFH	0H to 7FFH	The same device range as that of the connected CPU module
	SM	0 to 9999		—
Word Device	D	0 to 7999		—
	W	0H to 7FFFH	0H to 3FFH	The same device range as that of the connected CPU module
	SW	0H to 7FFFH	0H to 3FFH	The same device range as that of the connected CPU module
	SD	0 to 11999		—
	R	0 to 32767		—

Point

Set the devices 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, the following table lists the device ranges 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 Setting" of the communication destination.
	ZR	0 to 4849663	Data is read or written following the "File Register Setting" of the communication destination.

Point

Set the devices within the device range specified in "Device 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, the following table lists the device ranges 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 Setting" of the communication destination.
	ZR	0 to 393215	Data is read or written following the "File Register Setting" of the communication destination.

Point

Set the devices within the device range specified in "Device 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 side is the range that can be specified with the following commands of the MC protocol (A-compatible 1E frame): Batch read in units of words (01H) and Batch write in units of words (03H).

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 subcommand 0000H of SLMP (QnA-compatible 3E frame).

■When the communication destination is OMRON (CJ/CP series)

For devices that can be specified, the size of a device is specified in units of one 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 "Communication Pattern" is set to "Read".
	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 "Communication Pattern" is set to "Read". The upper eight bits are 0 (fixed). (because space is handled in units of bytes)
	EM	0 to 32767	The number of points and applicable devices differ depending on the model and settings of the CPU module used.
	EMn_	0 to 32767	The number of points and applicable devices differ depending on the model and settings of the CPU module used. "n" represents 0H to FH or 10H to 18H. (25 devices in total)

Point

In the IP address table on the communication destination side, set the IP address and node address to be assigned to the FX5 CPU module or Ethernet module. (Specify "1" for the node address.)

■When the communication destination is 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.

The format of bit devices other than "B" consists of the last two digits of bit specification and the upper digits of word specification.

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 Panasonic (FP7 series)

For devices that can be specified, the size of a device is specified in units of one 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 show the program block number, and the lower six digits show the device number.
	_WY	001000000 to 468000511	The upper three digits of the device number show the program block number, and the lower six digits show the device number.
	_WR	001000000 to 468002047	The upper three digits of the device number show the program block number, and the lower six digits show the device number.
	_WL	001000000 to 468001023	The upper three digits of the device number show the program block number, and the lower six digits show 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 show the program block number, and the lower six digits show the device number.
	_DT	001000000 to 468065534	The upper three digits of the device number show the program block number, and the lower six digits show the device number.

■When the communication destination is Panasonic (FP0H series)

For devices that can be specified, the size of a device is specified in units of one 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 "Communication Pattern" is set to "Read".
	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 a 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 last five digits of the target MODBUS device number - 1" to comply with the 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 "Communication Pattern" is set to "Read".
Word Device	Input Register	0 to 65535	The setting is available only when "Communication Pattern" is set to "Read".
	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 one point for a bit device and two points for a word device.

Type	Applicable device		
	Symbol	**Range**	**Remarks**
Bit Device (The device is displayed in units of words. One point corresponds to eight points.)	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 number, 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. (Because the size of one point is one byte, specify two points for a 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.

Precautions

Do not perform settings that span the DB, such as DB001510 to DB002509.

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
• 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)	This setting is not required.
MELSEC-FX3 (Ethernet block/adapter)	Configure the following settings so that the MC protocol (A-compatible 1E frame) can be used on the communication destination side. For details, refer to the manual for the communication destination. • Communication data code: "Binary" • Protocol: "UDP"
SLMP-compatible device (QnA-compatible 3E frame)	Configure the following settings so that the SLMP (QnA-compatible 3E frame) can be used on SLMP-compatible device side. For details, refer to the manual for the communication destination. • Communication data code: "Binary" • Protocol: "UDP"
OMRON (CJ/CP series)	In the IP address table, set the IP address and node address to be assigned to the FX5 CPU module. Set "1" for the node address. For other settings, refer to the manual for the communication destination.
• KEYENCE (KV series) • Panasonic (FP7 series) • Panasonic (FP0H series) • SIEMENS S7 series	Refer to the manual for the communication destination.
MODBUS/TCP-compatible device	Configure the following settings so that the MODBUS/TCP can be used on MODBUS/TCP-compatible device. For details, refer to the manual for the communication destination. • Communication data code: "Binary" • Protocol: "TCP"

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 ones of the setting numbers where no error occurs will operate.

If the communication destination responds at the timing of the END processing of the CPU module, the response will be reflected directly on the device by the END processing that received the response. For an SLMP-compatible device (QnA-compatible 3E frame), MELSEC-FX3 (Ethernet block/adapter), programmable controller made by another company, or 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
• 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.
• 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 different timing.

Timeout Time/Communication Retry Count/Monitoring Time At Error

Set the conditions for when communication errors occur.

Item	Description	Setting range
Timeout Time (ms)	Set the time after which an error will occur or a retry will start when no response is sent from the communication destination. Configure the setting so that (communication time-out period) ≥ (execution interval).	10 to 65535ms (in increments of 1ms)[*1] (Default: 1000ms)
Communication Retry Count	Set the number of communication retries to be performed within the communication time-out period when the communication destination returns an error response or does not respond within the communication time-out period.	0 to 255 times[*2] (Default: 3 times)
Monitoring Time At Error (s)	Set the monitoring time after a communication error occurred. When the communication setting is "On Request", this item cannot be set.[*3]	1 to 300s (Default: 30s)

*1 When the communication destination is set to "MELSEC-FX3 (Ethernet block/adapter)", "SLMP-compatible device (QnA-compatible 3E frame)" or "KEYENCE (KV series)", the value is fixed at 65535ms.

*2 When the communication destination is set to "MELSEC-FX3 (Ethernet block/adapter)", "SLMP-compatible device (QnA-compatible 3E frame)" or "KEYENCE (KV series)", the value is fixed at 0 times.

*3 Only FX5U/FX5UC CPU modules are supported. For the supported versions of each model, refer to the following.

☞ Page 936 Added and Changed Functions

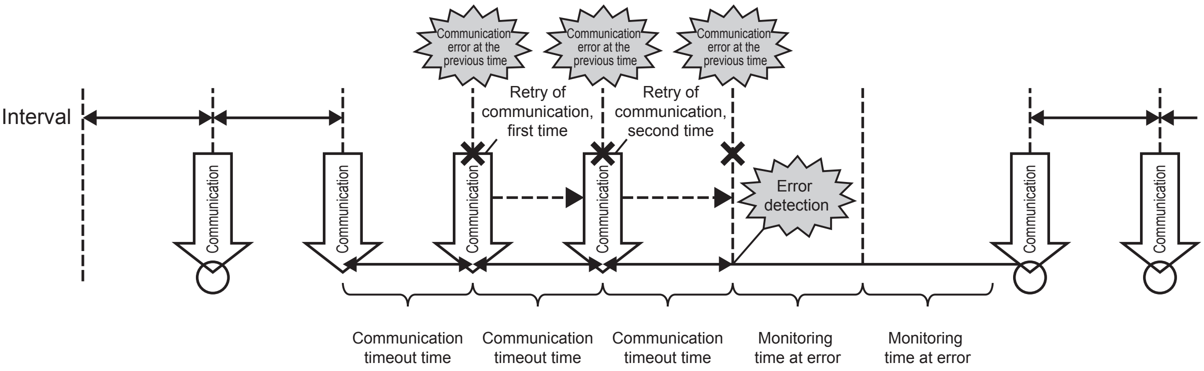
When an error response is received from the communication destination or no response is returned within the communication time-out period, the communication will be retried (resent).

When the communication setting is "Fixed", when the communication time-out period has elapsed after the set number of retries is performed, the fixed interval communication is performed during the monitoring time at error. If the communication destination responds after these communications, the communications at the execution interval specified with the parameters are restarted.

When the communication setting is "On Request", when the communication time-out period has elapsed after the set number of retries is performed, the communication fails.

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: (communication retry count + 1) × communication time-out period.

Latency Time

Set the waiting time until communications are started.

Item	Description	Setting range
Latency Time	Set the time required to start communications after completion of the Ethernet-equipped module startup. The time cannot be changed for each setting number, because the setting is common for the simple CPU communication function.	0 to 255s (in increments of 1s) (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.

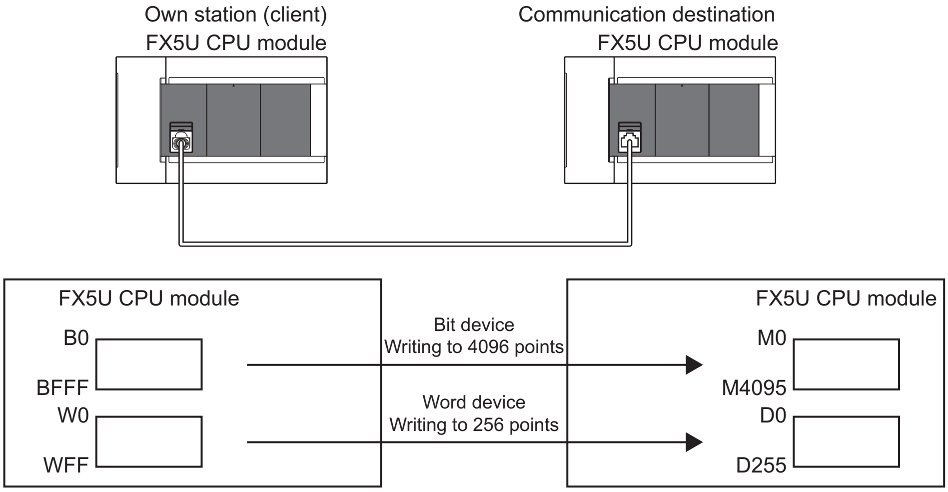
5.6 Program Examples of Communication on Request

When the execution interval in the communication setting is set to "On Request", programming is required for sending requests.

The following is an example of a program for simple CPU communication (communicating on request).

When the own station is a CPU module

The FX5U CPU module (own station) and the FX5U CPU module (communication destination) communicate with each other.



Simple CPU Communication Setting

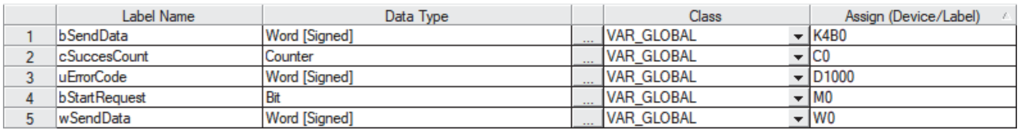
The following table lists the own station parameter settings (setting No.1). Optionally set other parameters not listed below.

Item			Setting value
Communication Pattern			Write
Communication Setting: Execution Interval (ms)	Comm Set		On Request
Communication Destination (IP Address)	Destination	Device Type	MELSEC iQ-F (built-in Ethernet)
Bit Device	Source	Type/Start/End	B0 to BFFF[*1]
	Destination	Type/Start/End	M0 to M4095
Word Device	Source	Type/Start/End	W0 to WFF
	Destination	Type/Start/End	D0 to D255

*1 Change the value in "Memory/Device Setting" in "CPU Parameter" since the device range is B0 to BFF by default.

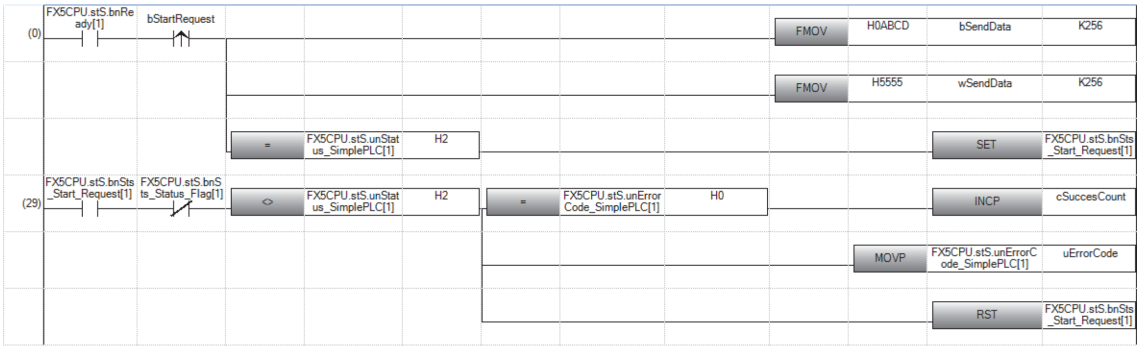
Program example

■Devices used

Classification	Label name	Description	Device
Module label	FX5CPU.stS.bnSts_Start_Request[1]	Communication start request for simple CPU communication setting No.1 (when communication setting is "When requested")	SD10350.0
	FX5CPU.stS.bnSts_Status_Flag[1]	Execution status flag for simple CPU communication setting No.1	SD10356.0
	FX5CPU.stS.bnReady[1]	Ready flag for simple CPU communication setting No.1	SD10358.0
	FX5CPU.stS.unStatus_SimplePLC[1]	Communication status storage location for simple CPU communication setting No.1	SD10380
	FX5CPU.stS.unErrorCode_SimplePLC[1]	Error code storage location for simple CPU communication setting No.1	SD10412
Global label/local label	Define the global labels or local labels as shown below. 		

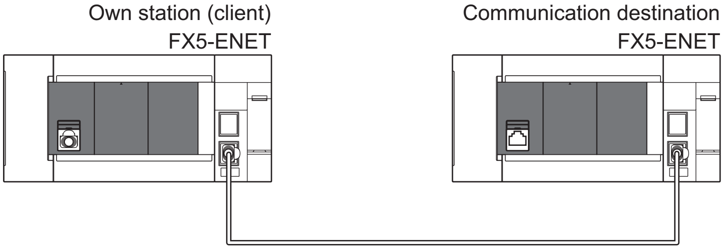
	Label Name	Data Type		Class	Assign (Device/Label)
1	bSendData	Word [Signed]	...	VAR_GLOBAL	K4B0
2	cSuccesCount	Counter	...	VAR_GLOBAL	C0
3	uErrorCode	Word [Signed]	...	VAR_GLOBAL	D1000
4	bStartRequest	Bit	...	VAR_GLOBAL	M0
5	wSendData	Word [Signed]	...	VAR_GLOBAL	W0

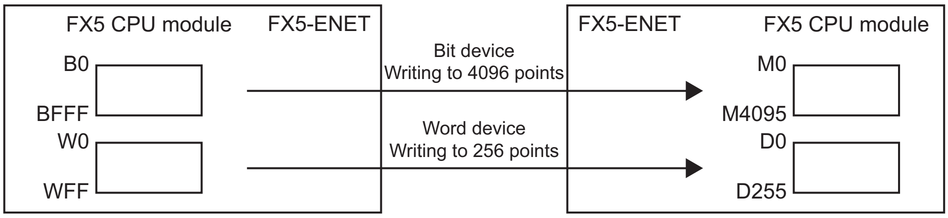
■Program



When the own station is an Ethernet module

The FX5-ENET (own station) and the FX5-ENET (communication destination) communicate with each other.





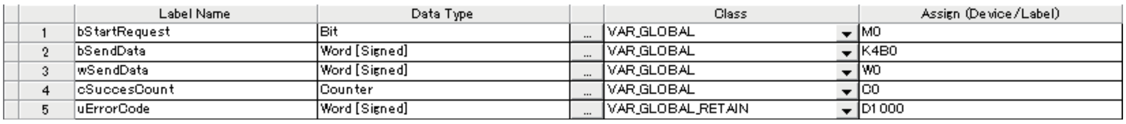
Simple CPU Communication Setting

The following table lists the own station parameter settings (setting No.1). Optionally set other parameters not listed below.

Item			Setting value
Communication Pattern			Write
Communication Setting: Execution Interval (ms)	Comm Set		On Request
Communication Destination (IP Address)	Destination	Device Type	MELSEC iQ-F (Ethernet module)
Bit Device	Source	Type/Start/End	B0 to BFFF
	Destination	Type/Start/End	M0 to M4095
Word Device	Source	Type/Start/End	W0 to WFF
	Destination	Type/Start/End	D0 to D255

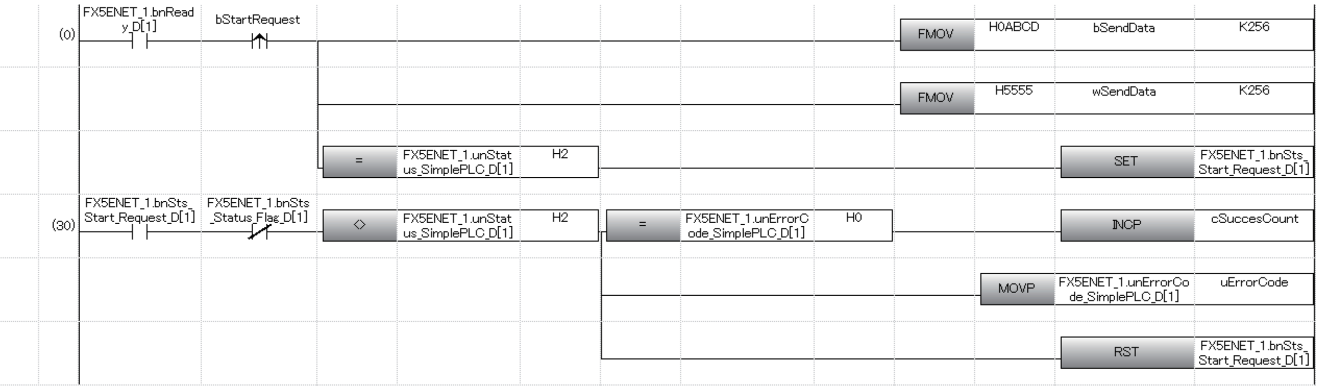
Program example

■Devices used

Classification	Label name	Description	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	Define the global labels or local labels as shown below. 		

	Label Name	Data Type		Class	Assign (Device/Label)
1	bStartRequest	Bit	...	VAR_GLOBAL	M0
2	bSendData	Word [Signed]	...	VAR_GLOBAL	K4B0
3	wSendData	Word [Signed]	...	VAR_GLOBAL	W0
4	cSuccesCount	Counter	...	VAR_GLOBAL	C0
5	uErrorCode	Word [Signed]	...	VAR_GLOBAL_RETAIN	D1000

■Program



5.7 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 the 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 settings

- The execution interval of the simple CPU communication function will differ according to the Ethernet line's load rate, operating status of other communication functions (such as MELSOFT connection, SLMP communication, socket communication function, and 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.
- For a CPU module, the simple CPU communication function operates at regular intervals. Therefore, if a short execution interval is set, access to the SD memory card may be delayed. The following table lists the functions that delay access to the SD memory card.

Function	Operation that delays access to the SD memory card
Access to the SD memory card by peripheral devices	Reading/writing data from/to the CPU module, making online changes, etc.
Data logging function	Collecting data, creating a data logging file, reading/writing/removing the data logging settings, and viewing or manipulating the logging status
Memory dump function	Saving/reading the memory dump file
Extended file register	Executing the ERREAD/ERWRITE/ERINIT instructions, data batch reading function, data batch writing function, data batch initialization function
Event history function	Saving/viewing/clearing the 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)	Generally accessing CPU modules
File transfer function (FTP client)	Generally accessing CPU modules
Web server function	Generally accessing CPU modules, displaying a user Web page

To communicate at the set execution interval, reducing the communication frequency is recommended. (For example, increase the execution interval, communication time-out period, and monitoring time at error for the simple CPU communication, and reduce the number of settings.) If checking the execution interval is required, perform actual communications and check them with the "Simple CPU 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.
- The own station port numbers are the same for modules with different simple CPU communication settings.

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.

Point

When a MODBUS/TCP-compatible device or SIEMENS S7 series device is used, if the communication destinations have the same IP address, the own station port number can be duplicated. For the supported versions of each model, refer to the following.

☞ Page 936 Added and Changed Functions

IP address and device type of communication target devices

Check the IP address and device type of the communication target 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 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-compatible device

When specifying 00H (broadcast) for communication between an Ethernet module and a MODBUS RTU/ASCII-compatible device through a gateway device, pay attention to the following.

- Do not specify "Read" for the communication pattern. Since the slave station does not respond, a timeout error occurs.
- Do not simultaneously set a bit device and a word device with the same setting number. Set only one of them for each setting.
- When setting multiple devices, set different values for all own station port numbers.

For the communication time-out period, communication retry count, and monitoring time at error, set as follows.

- Communication retry count = 0
- Communication Time-out Period = Monitoring Time At Error

According to the above settings, the second communication is executed after a lapse of the communication timeout time, and the third and subsequent communications are executed at intervals of the Communication Time-out Period × 2.

IP filter function

The simple CPU communication is not performed when the IP filter function interrupts the communication destination. In addition, if any unique security function specific to the communication destination (programmable controller manufactured by other companies) is working, disable the function.

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 the file register (R) is used

When using the file register (R) for a device on the own station, use device areas within the setting range. For any setting number using device areas outside the setting range, simple CPU communication cannot be performed.

Redundant system

If the communication destination is a redundant system, on the process CPU (in a redundant mode) side, use a control system IP address with redundant settings, and set the control system IP address as the communication destination IP address of the simple CPU communication.

Number of connected communication destinations

An Ethernet module can be connected with a maximum of 32 communication destinations. However, one connection is occupied per communication destination (per group). Therefore, the following number of communication destinations can be connected. (When the number of settings per group is 1)

- Maximum number of connectable devices = 32 - number of connections occupied by other functions (SLMP and others)

Communication on request

- The time taken from when the Request to start communication at request (SD10350 or SD10351) for Simple CPU communication is turned on to when the communication is completed will differ according to the Ethernet line's load rate, operating status of other communication functions (such as MELSOFT connection, SLMP communication, socket communication function, and 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.
- If the Request to start communication at request (SD10350 or SD10351) is turned off while a communication on request is being executed or Ready flag is off, communication on request is not executed.

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 =: 1ms (scan time before execution of simple CPU communication function)
- Communication destination: MELSEC iQ-F (built-in Ethernet)
- Communication destination sequence scan time: 1ms (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.

6 SLMP FUNCTION

This chapter describes the SLMP function. The following operations can be performed by using the SLMP function.

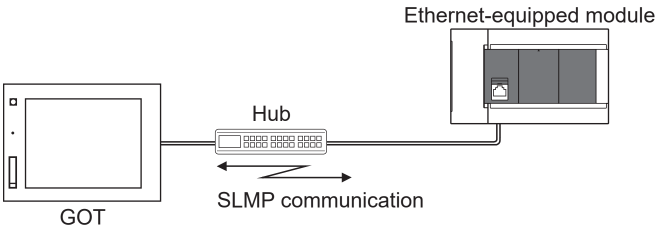
- Ethernet-equipped module device data can be read and written using SLMP (3E/1E frames) from external devices.
- Ethernet-equipped module operation monitoring, data analysis, and production control is possible from external devices by reading and writing device data.
- With the remote password function, unauthorized access from the outside can be prevented. (☞ Page 235 Remote Password)

For details on SLMP, refer to the following.

☞ Page 554 OUTLINE

For the predefined protocol support function and the communications using SLMP (set with the predefined protocol library), refer to the following.

☞ Page 100 PREDEFINED PROTOCOL SUPPORT FUNCTION



Data communication procedures

1. Connecting cables and external devices

Make the connections for SLMP communication. (☞ Page 28 Connection Specifications)

2. Setting parameters

Configure the module parameters with the engineering tool. (☞ Page 93 Setting Method)

3. Writing to the Ethernet-equipped module

Write the parameters set in the Ethernet-equipped module. Turn power OFF → ON or reset the system 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 95 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 46 Communication via router)

6.1 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		100m (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 95 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 become longer depending on the load of the network (how much a line is crowded), window size of each connecting device, number of connections to be used simultaneously, and 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 of the SLMP communication (for batch read or batch write)

Tfs = Ke + (Kdt × Df) + Scr × Number of scans required for processing + ACK processing time of external equipment

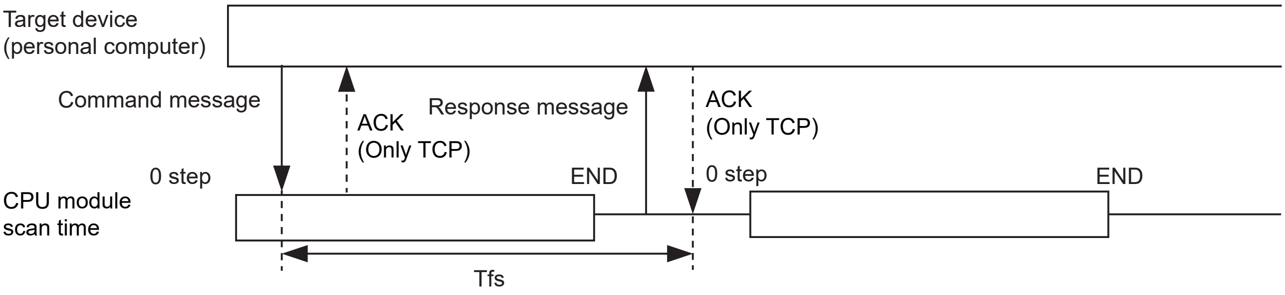
Tfs: The time from when the personal computer receives the request data until the CPU module finishes processing (unit: ms)[*1]

Ke, Kdt: Constant (Refer to the following table.)

Df: Number of words of the request data + Number of words of the response data (application data part)

Scr: Scan time

[*1] The following shows the timing from when the personal computer receives the request data until the CPU module 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 40ms.

Tfs=1+(0.001×32)+40×1+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 40ms.

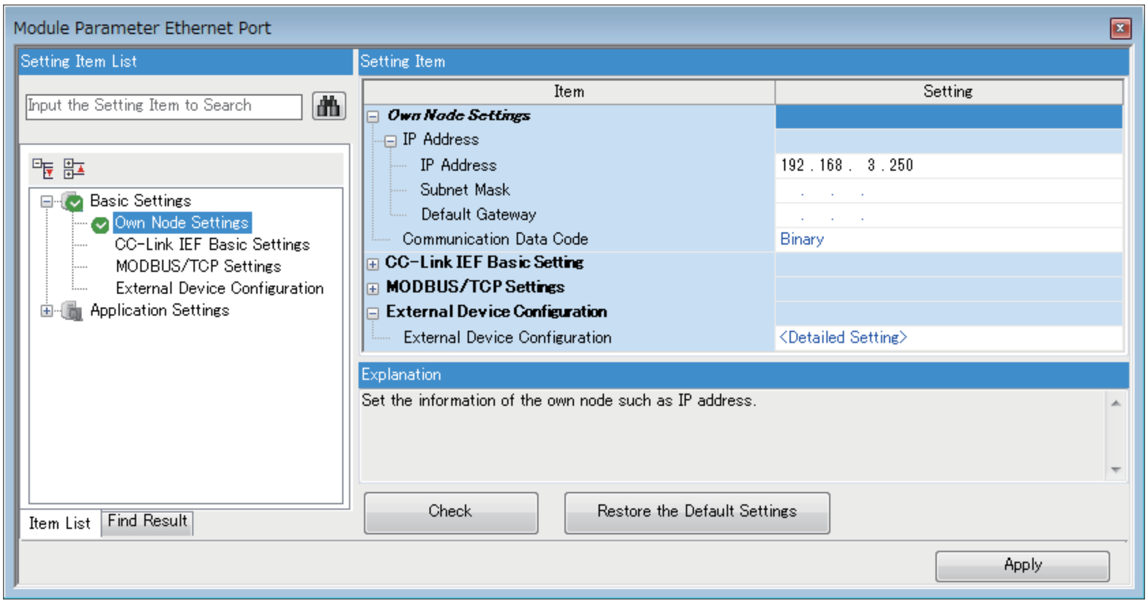
Tfs=1+(0.001×32)+40×1+other device ACK processing time

6.2 Setting Method

The following shows the configuration for communication by SLMP.

CPU module

Navigation window ⇨ [Parameter] ⇨ Module name ⇨ [Module Parameter] ⇨ [Ethernet Port] ⇨ [Basic Settings] ⇨ [Own Node Settings]



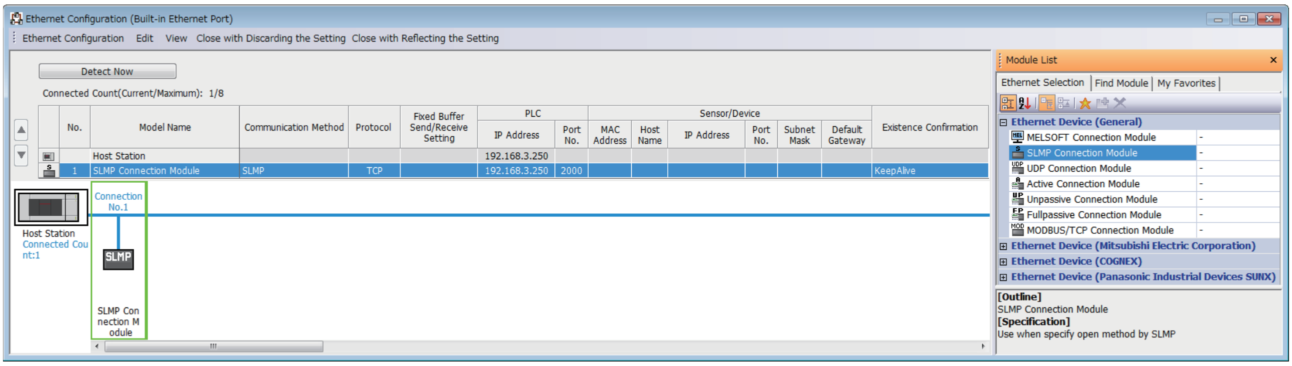
Item	Setting
Own Node Settings	
IP Address	
IP Address	192 . 168 . 3 . 250
Subnet Mask	. . .
Default Gateway	. . .
Communication Data Code	Binary
CC-Link IEF Basic Setting	
MODBUS/TCP Settings	
External Device Configuration	
External Device Configuration	<Detailed Setting>

Set the information of the own node such as IP address.

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 model name ⇨ [Module Parameter] ⇨ [Ethernet Port] ⇨ [Basic Settings] ⇨ [External Device Configuration] ⇨ [Detailed Setting] ⇨ [Ethernet Configuration (Built-in Ethernet Port)] screen



Connected Count(Current/Maximum): 1/8

No.	Model Name	Communication Method	Protocol	Fixed Buffer Send/Receive Setting	PLC		Sensor/Device							Existence Confirmation
					IP Address	Port No.	MAC Address	Host Name	IP Address	Port No.	Subnet Mask	Default Gateway		
	Host Station				192.168.3.250									
1	SLMP Connection Module	SLMP	TCP		192.168.3.250	2000								KeepAlive

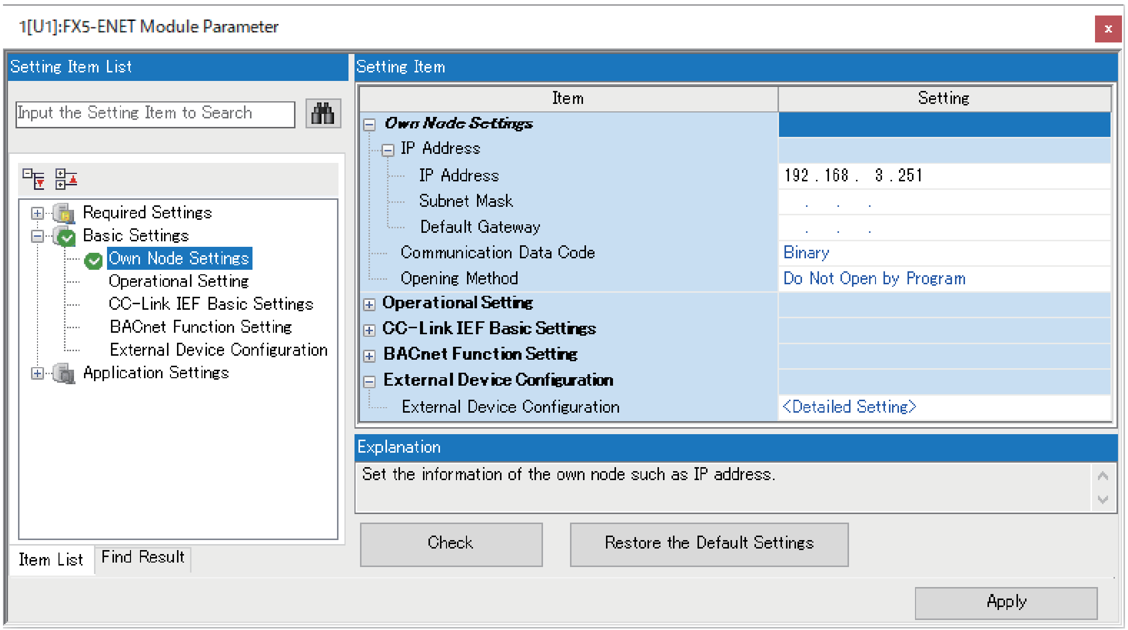
Ethernet Device (General)	
MELSOFT Connection Module	-
SLMP Connection Module	-
UDP Connection Module	-
Active Connection Module	-
Unpassive Connection Module	-
Fullpassive Connection Module	-
MODBUS/TCP Connection Module	-
Ethernet Device (Mitsubishi Electric Corporation)	
Ethernet Device (COGNEX)	
Ethernet Device (Panasonic Industrial Devices SUNX)	

[Outline]
SLMP Connection Module
[Specification]
Use when specify open method by SLMP

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 and 5570 to 65534) for the "Port No." and the IP address of the target device (only UDP). Do not specify the port numbers 5549 to 5569 because these ports are used by the system.

Ethernet module

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



Item	Setting
Own Node Settings	
IP Address	
IP Address	192 . 168 . 3 . 251
Subnet Mask	. . .
Default Gateway	. . .
Communication Data Code	Binary
Opening Method	Do Not Open by Program
Operational Setting	
CC-Link IEF Basic Settings	
BACnet Function Setting	
External Device Configuration	
External Device Configuration	<Detailed Setting>

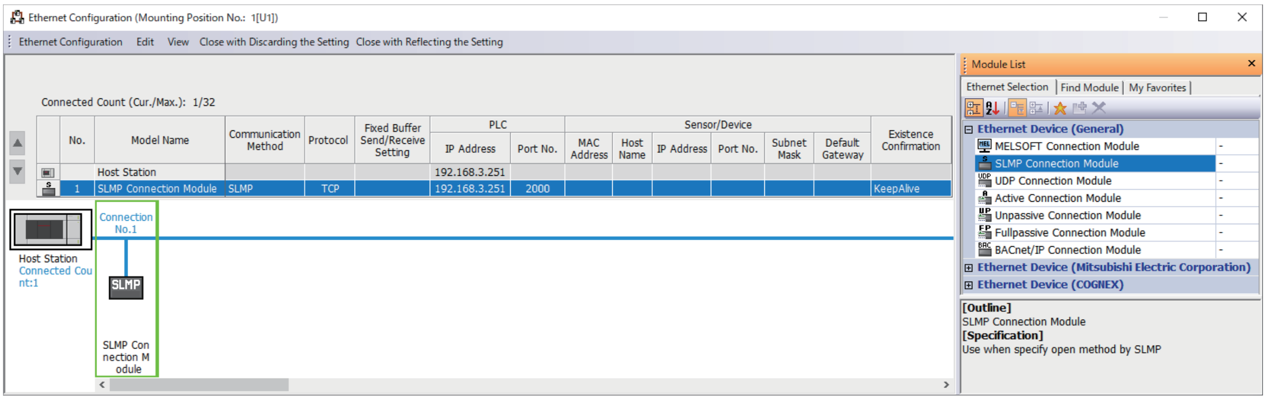
Explanation

Set the information of the own node such as IP address.

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



Connected Count (Cur./Max.): 1/32

No.	Model Name	Communication Method	Protocol	Fixed Buffer Send/Receive Setting	PLC		Sensor/Device						Existence Confirmation
					IP Address	Port No.	MAC Address	Host Name	IP Address	Port No.	Subnet Mask	Default Gateway	
	Host Station				192.168.3.251								
1	SLMP Connection Module	SLMP	TCP		192.168.3.251	2000							KeepAlive

[Outline]
SLMP Connection Module
[Specification]
Use when specify open method by SLMP

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 and 5570 to 65534) in "Port No." Do not specify the port numbers 5549 to 5569 because these ports are used by the system.

6.3 SLMP Commands

For details on the SLMP commands, refer to the following.

☞ Page 602 3E FRAME COMMANDS

☞ Page 666 1E FRAME COMMANDS

6.4 Data Communication Process

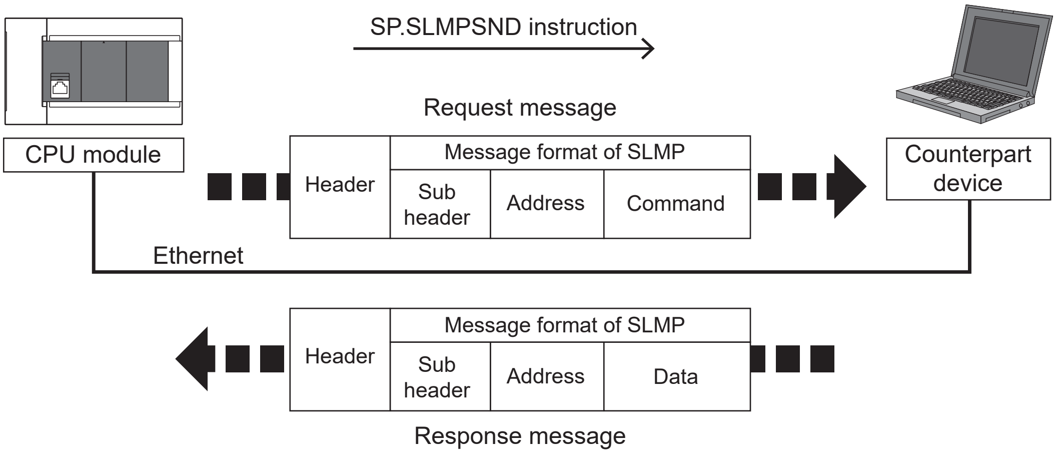
Sending the SLMP frame

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)

6.5 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 display 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

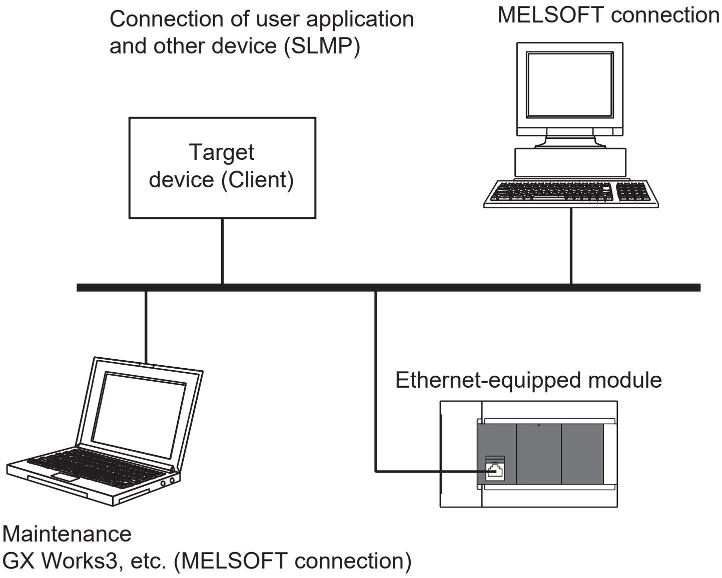
For the error codes stored in the end code when there is an abnormal end of SLMP (3E frame) communication, refer to ☞ Page 826 SLMP function error code.

Number of connectable modules

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 modules
- Ethernet module: Up to 32 modules

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



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 FFFEH) 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 of MC protocol.

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 on the access range, refer to the following.

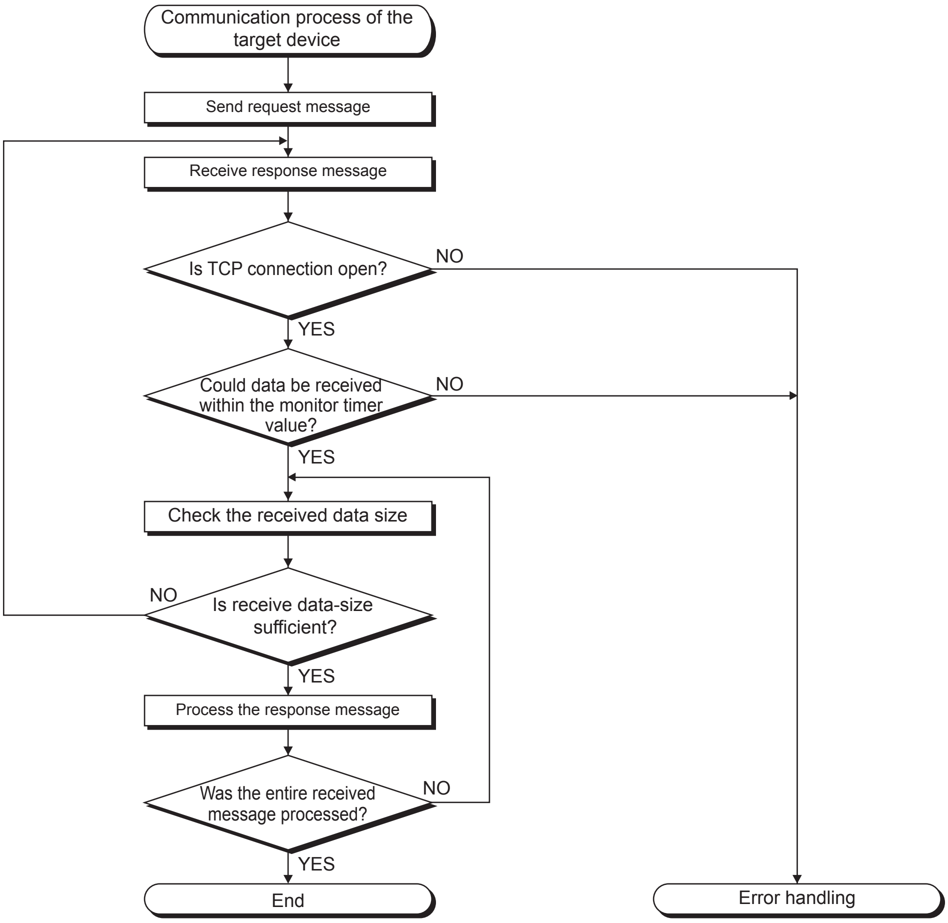
☞ Page 559 Access range

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.



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.

CPU module processing time of SLMP

When accessing the CPU module from an external device using SLMP communication, the following "intervention time to the scan time" and "number of scans for processing" of the CPU module side are required. On the request from the external device using SLMP communication, the CPU module processes a specified number of points during each END processing in case the CPU module is running.

Point

The processing time of the following modules depends on the usage of other intelligent function modules.

- FX5-ENET
- FX5-ENET/IP
- FX5-CCLGN-MS
- FX5-CCLIEF
- FX5-40SSC-G, FX5-80SSC-G

■3E frame

Item		Command	Sub-commands	Access points 1) / 2)	Intervention time [ms]*2 (extension of scan time)						Number of scans required for processing
Type	Operation				Access point 1)			Access point 2)			
					FX5S	FX5UJ	FX5U/ FX5UC	FX5S	FX5UJ	FX5U/ FX5UC	
Device	Read	0401	0001	1/3584	0.04	0.04	0.03	2.30	2.23	1.33	1
			0000	1/960	0.04	0.04	0.03	0.25	0.24	0.14	1
	Write	1401	0001	1/3584	0.04	0.04	0.04	1.46	1.42	1.06	1
			0000	1/960	0.04	0.04	0.03	0.28	0.26	0.20	1
	Read Random	0403	0000	1/192	0.04	0.04	0.03	2.23	1.55	1.48	1
	Write Random	1402	0001	1/188	0.04	0.04	0.03	2.11	1.49	1.41	1
			0000	1/160*1	0.04	0.04	0.04	1.90	1.33	1.26	1
	Read Block	0406	0000	1/960	0.04	0.04	0.03	0.26	0.24	0.14	1
	Write Block	1406	0000	1/770	0.04	0.04	0.03	0.24	0.22	0.17	1
Remote Control	Read Type Name	0101	0000	(one station)	0.04	0.04	0.02	—	—	—	1

*1 This is the processing time when accessing with only word access points specified.

*2 This is the processing time (an average of actual measurements) when 1 is set to "CPU Parameter" - "Service Processing Setting" - "Device/Label Access Service Processing Setting" - "Set Processing Counts" of GX Works3.

■1E frame

- When communicating data in ASCII code

Operation	Command	Access points 1) / 2)	Intervention time [ms]*2 (extension of scan time)		Number of scans required for processing
			Access point 1)	Access point 2)	
Batch Reading	00H	1/256	0.0187	0.0646	1
	01H	1/64	0.0196	0.0232	1
Batch Writing	02H	1/160	0.0206	0.0524	1
	03H	1/64	0.0212	0.0268	1
Test (Random Write)	04H	1/80	0.0221	0.5200	1
	05H	1/10*1	0.0230	0.0788	1
Remote RUN	13H	—	0.0184	—	1
Remote STOP	14H	—	0.0187	—	1
Read PC Type Name	15H	—	0.0129	—	1
Loopback Test	16H	1/254	0.0201	0.0480	1

*1 This is the processing time when accessing with only word access points specified.

*2 This is the processing time (an average of actual measurements) when 1 is set to "CPU Parameter" - "Service Processing Setting" - "Device/Label Access Service Processing Setting" - "Set Processing Counts" of GX Works3.

- When communicating data in binary code

Operation	Command	Access points 1) / 2)	Intervention time [ms]*4 (extension of scan time)		Number of scans required for processing
			Access point 1)	Access point 2)	
Batch Reading	00H	1/256	0.0267	0.1250	1
	01H	1/64	0.0287	0.1050	1
Batch Writing	02H	1/160	0.0290	0.0950	1
	03H	1/64	0.0304	0.0895	1
Test (Random Write)	04H	1/80	0.0308	0.8010	1
	05H	1/10*3	0.0323	0.1210	1
Remote RUN	13H	—	0.0210	—	1
Remote STOP	14H	—	0.0222	—	1
Read PC Type Name	15H	—	0.0240	—	1
Loopback Test	16H	1/254	0.0265	0.2620	1

*3 This is the processing time when accessing with only word access points specified.

*4 This is the processing time (an average of actual measurements) when 1 is set to "CPU Parameter" - "Service Processing Setting" - "Device/Label Access Service Processing Setting" - "Set Processing Counts" of GX Works3.

Point

- Number of scans required for processing

The CPU module processes only one command during an END processing. If GX Works3 or other modules are also accessing the CPU module simultaneously, the number of scans required for processing may increase due to the waiting time.

- Method of reducing the intervention time to the scan time

Adjust the service process execution count of the CPU module in "CPU Parameter" - "Service Processing Setting" - "Device/Label Access Service Processing Setting" to reduce the intervention time to the scan time. (📖 MELSEC iQ-F FX5 User's Manual (Application))

- When extension of scan time affects the control

Access multiple times with less points.

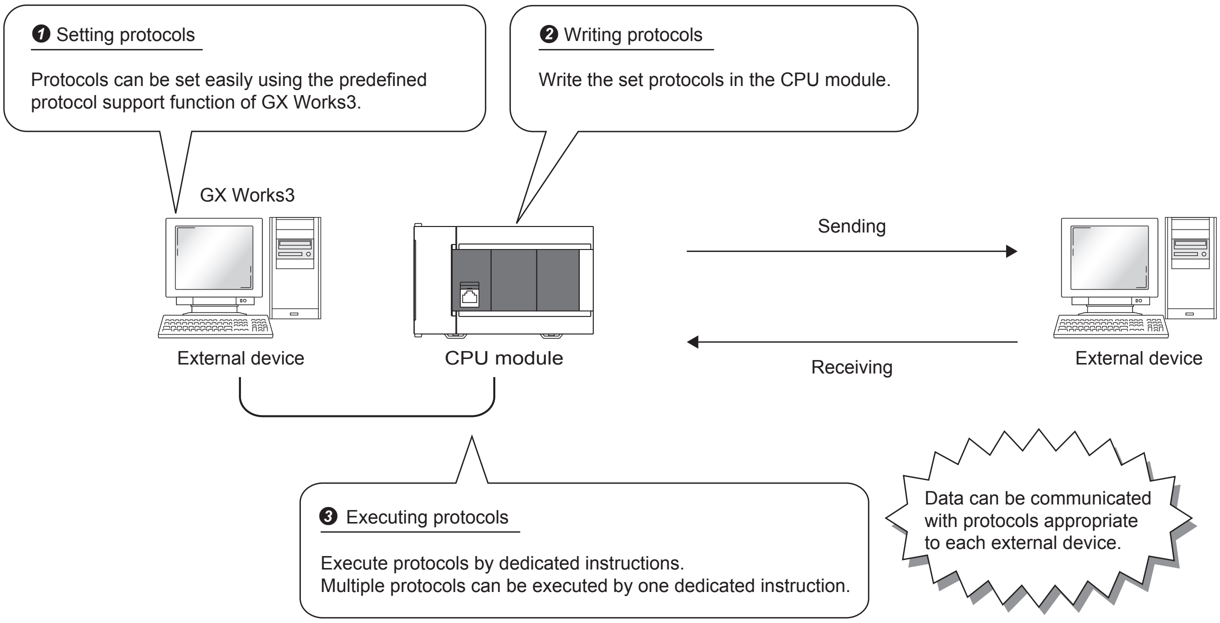
7 PREDEFINED PROTOCOL SUPPORT FUNCTION

This chapter describes the predefined protocol support function (built-in Ethernet).
For details on the predefined protocol support function (serial communication), refer to the following.
☞ Page 452 PREDEFINED PROTOCOL SUPPORT FUNCTION

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 (4E frame), MODBUS/TCP[*1] etc.), or it can be created and edited freely.

[*1] The SLMP (4E frame) and MODBUS/TCP are available only for the client.



❶ Setting protocols

Protocols can be set easily using the predefined protocol support function of GX Works3.

❷ Writing protocols

Write the set protocols in the CPU module.

❸ Executing protocols

Execute protocols by dedicated instructions.
Multiple protocols can be executed by one dedicated instruction.

Data can be communicated with protocols appropriate to each external device.

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.

7.1 Data Communication

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

1. Select the protocol with the predefined protocol support function, create or edit the data, and write the protocol setting data. (☞ Page 101 Creating the protocol setting data)

2. Set the module parameter. (☞ Page 106 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

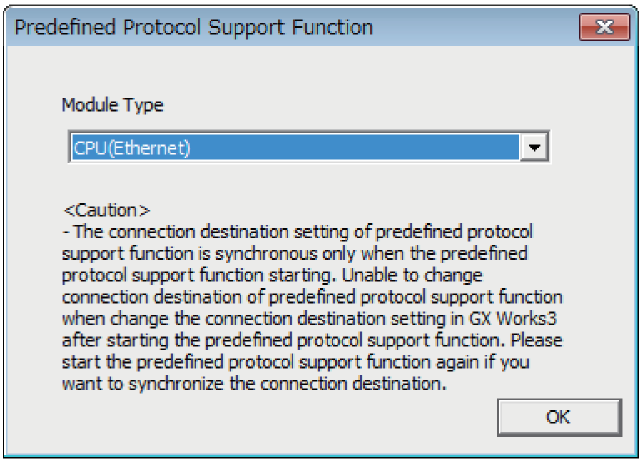
The communication data code is binary code communication regardless of the selected settings.

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.



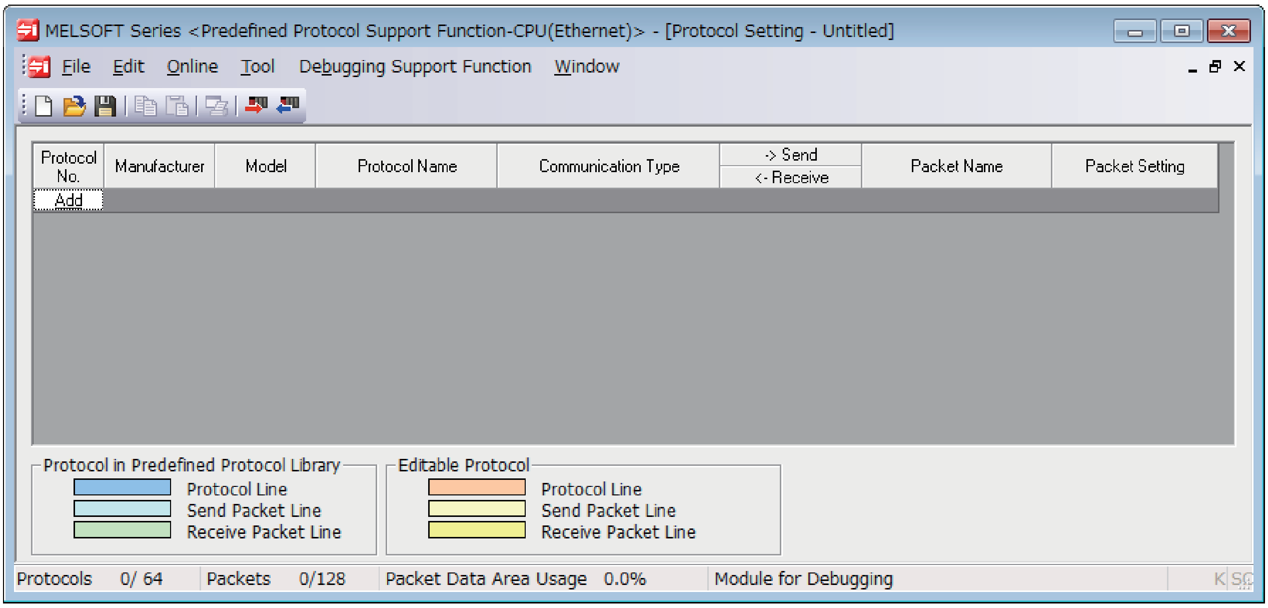
Module Type: CPU(Ethernet)

<Caution>
- The connection destination setting of predefined protocol support function is synchronous only when the predefined protocol support function starting. Unable to change connection destination of predefined protocol support function when change the connection destination setting in GX Works3 after starting the predefined protocol support function. Please start the predefined protocol support function again if you want to synchronize the connection destination.

■Newly creating the protocol setting data

Newly create the protocol setting data.

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

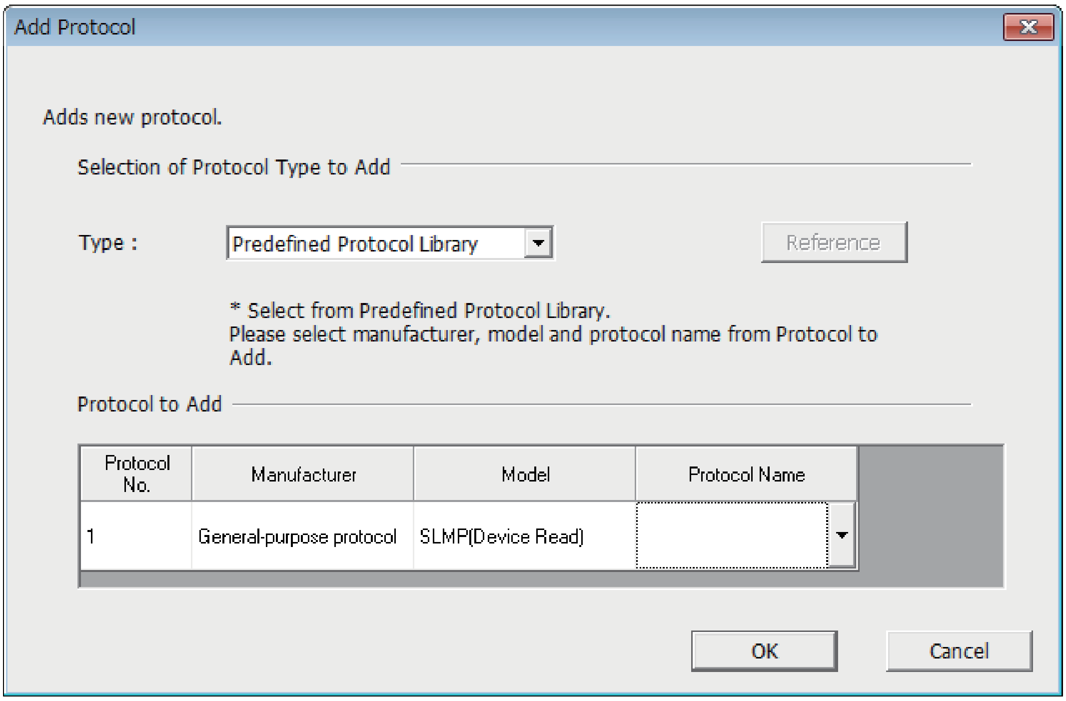


Item	Description
Protocol No.	Displays the protocol number used with the dedicated instruction.
Manufacturer	Displays the name of the manufacturer of the device for which the protocol is being set.
Model	Displays the model of the protocol to be set.
Protocol Name	Displays the name of the protocol to 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 receive packets, it is received. Send & receive: After sending one send packet, if there is a matching packet within up to 16 registered receive 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 : Predefined Protocol Library

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

Protocol to Add

Protocol No.	Manufacturer	Model	Protocol Name
1	General-purpose protocol	SLMP(Device Read)	

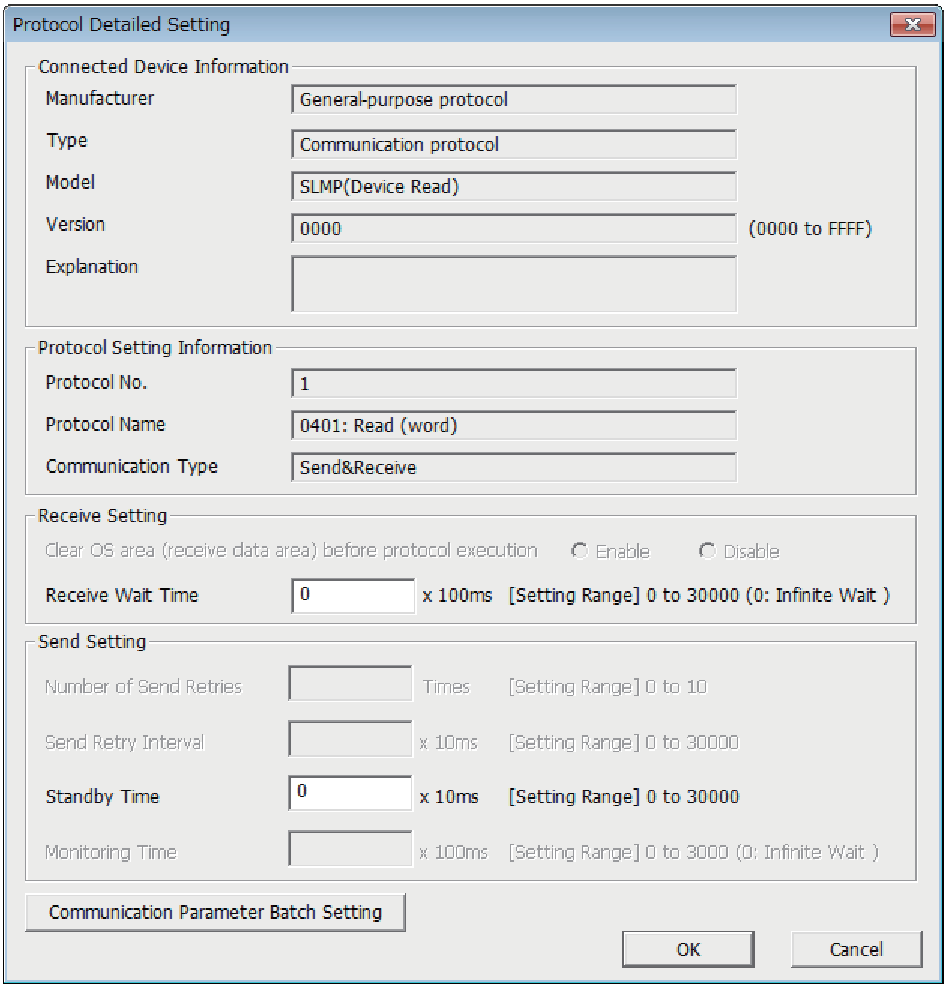
Item	Description	Setting range
Type	Select the type of protocol to be added.	• Predefined Protocol Library • User Protocol Library • Add New
Protocol No.	Select the protocol number to be added.	1 to 64
Manufacturer[*1]	Set the maker of the protocol to be added.	—
Model[*1]	Set the type of protocol to 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" screen ⇨ 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)

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.

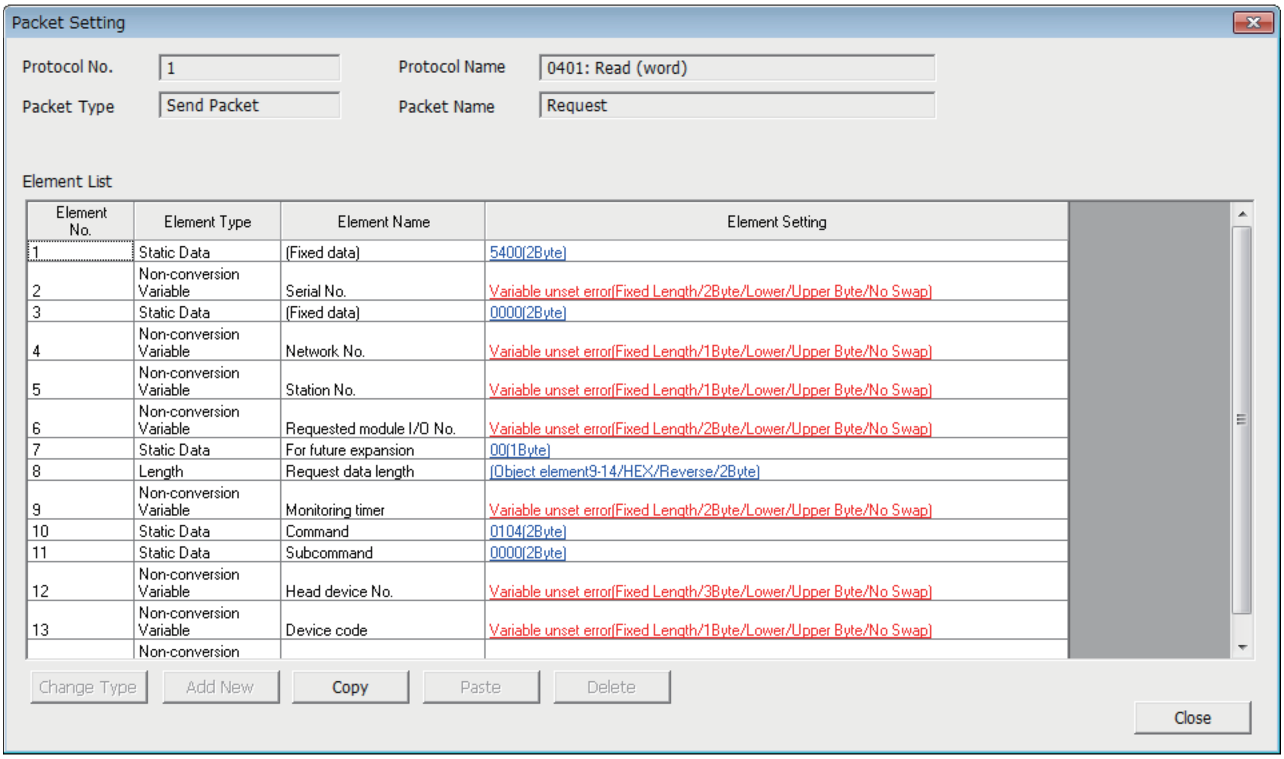
Point

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" screen.

"Protocol Setting" screen ⇨ Packet to be set



Protocol No. 1

Protocol Name 0401: Read (word)

Packet Type Send Packet

Packet Name Request

Element List

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)
	Non-conversion		

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

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

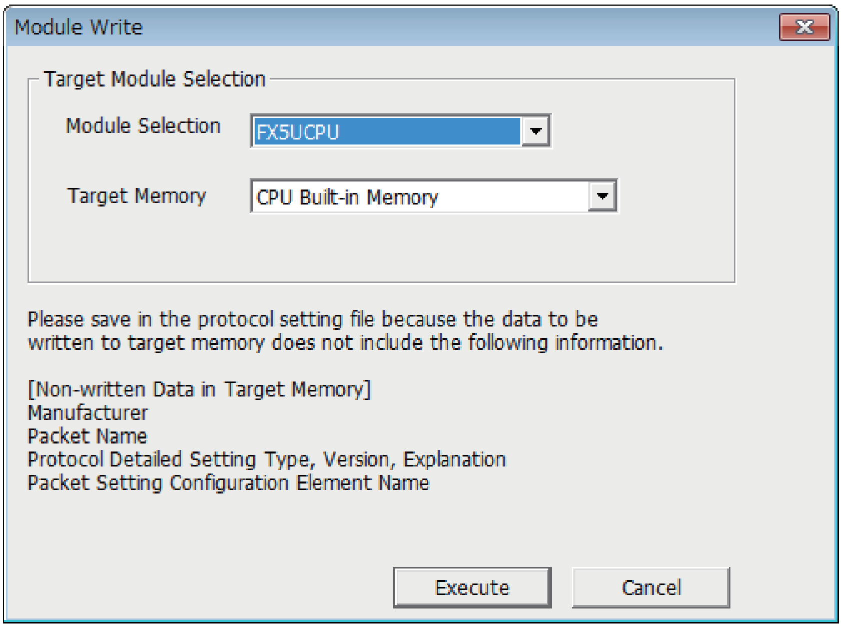
For details on the packet elements, refer to the following.

☞ Page 107 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 FX5UCPU

Target Memory CPU Built-in 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

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.

Point

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 114 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 an 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.

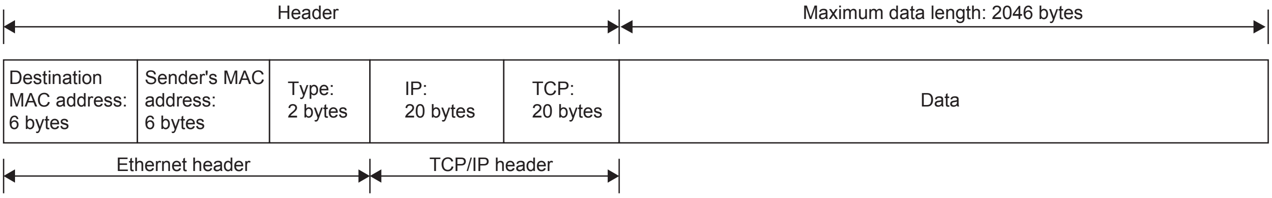
Applicable dedicated instructions

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

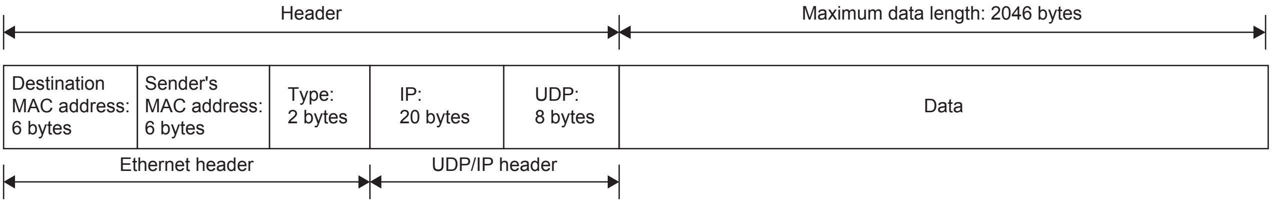
7.2 Protocol Communication Type

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.

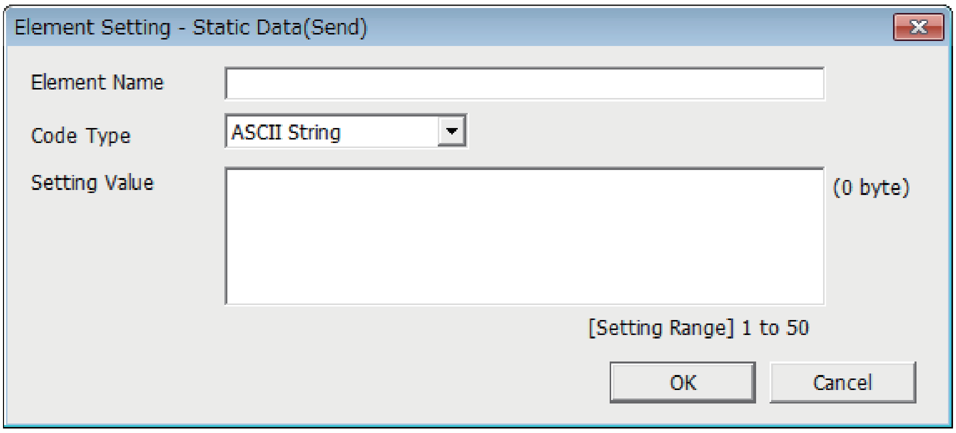
7.3 Packet Elements

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


Element Setting - Static Data(Send)

Code Type: ASCII String

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

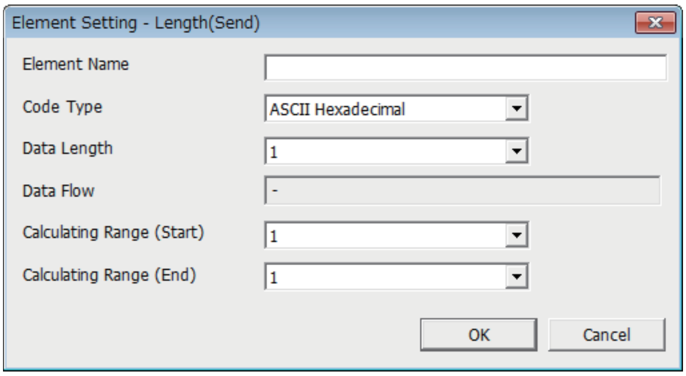
- When sending: The specified code and character string are sent.
- When receiving: The received data is verified.

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: • 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



Element Setting - Length(Send)

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

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.

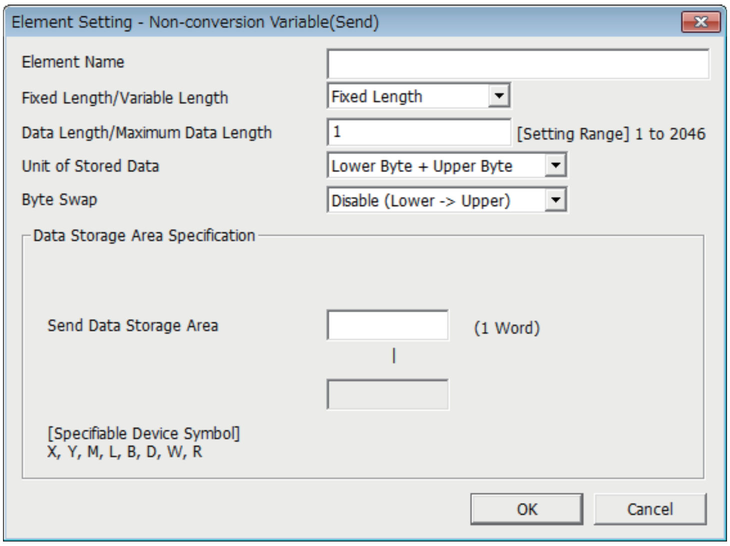
Item		Description		Remarks
Element Name		Set the element name.		—
Code Type		Select a data type of the setting value. ASCII hexadecimal/HEX		—
Data Length		Select the data length on the line. The range is 1 to 4 bytes.		—
Data Flow		Forward Direction (Upper byte → Lower byte)	When sending: Sends a calculated length from the upper byte. When receiving: Receives data from the upper byte.	This cannot be set if the data length is 1 byte.
		Reverse Direction (Lower byte → Upper byte)	When sending: Sends a calculated length from the lower byte. When receiving: Receives data from the lower byte.	
		Byte Swap (by Word)[*1]	When sending: Sends a calculated length swapping the upper byte and lower byte by word. When receiving: Receives data swapping the upper byte and lower byte by word.	
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



Element Setting - Non-conversion Variable(Send)

Element Name

Fixed Length/Variable Length: Fixed Length

Data Length/Maximum Data Length: 1 [Setting Range] 1 to 2046

Unit of Stored Data: Lower Byte + Upper Byte

Byte Swap: Disable (Lower -> Upper)

Data Storage Area Specification

Send Data Storage Area (1 Word)

[Specifiable Device Symbol]
X, Y, M, L, B, D, W, R

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: Specifies the data length at the time of the protocol execution and sends data. When receiving: Receives data whose data length is variable.
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: Sends each one word (2 bytes) data in the data storage area in the order of the lower byte to the upper byte. When receiving: Stores the received data to the data storage area in the order of the lower byte to the upper byte.
	Lower Bytes Only	When sending: Sends only lower byte of data in the data storage area. The CPU module ignores the upper byte data. When receiving: Stores the received data in the only lower byte of 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] • Input (X) • Output (Y) • Internal relay (M) • Latch relay (L) • Link relay (B) • Data register (D) • Link register (W) File register[*1] • File register (R)	

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

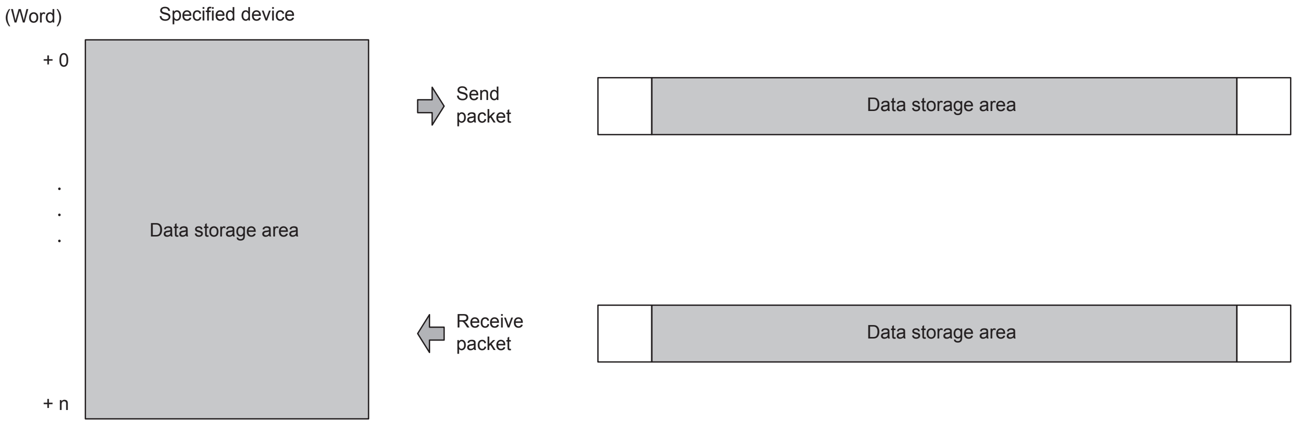
The following figures show the configuration of the data storage area.

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

The area after the device number specified on the "Element Setting" screen 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

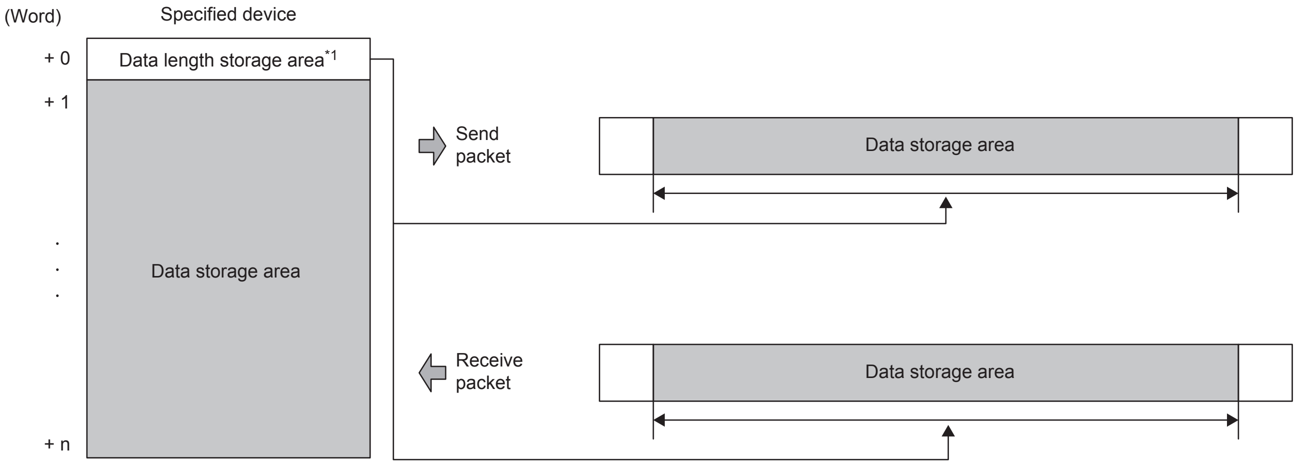
For receive packet: Receive data is stored by the CPU module

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

The area after the device number specified on the "Element Setting" screen + 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

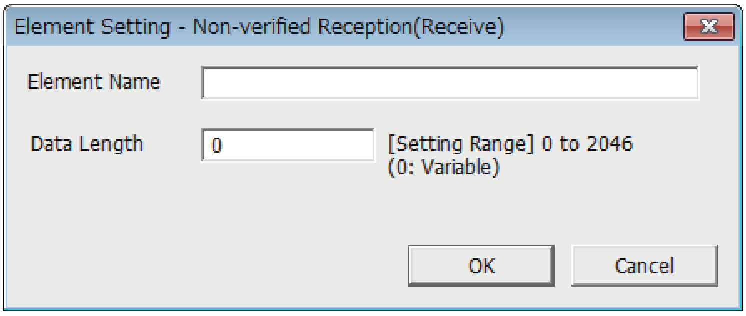
*1 The data length unit is byte fixed

Point

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 non-conversion 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 (variable number of characters)	Set when the number of characters that are not verified differs between each communication session.
	1 to 2046 (specified number of characters)	Set the number of characters that are not verified.

Point

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.

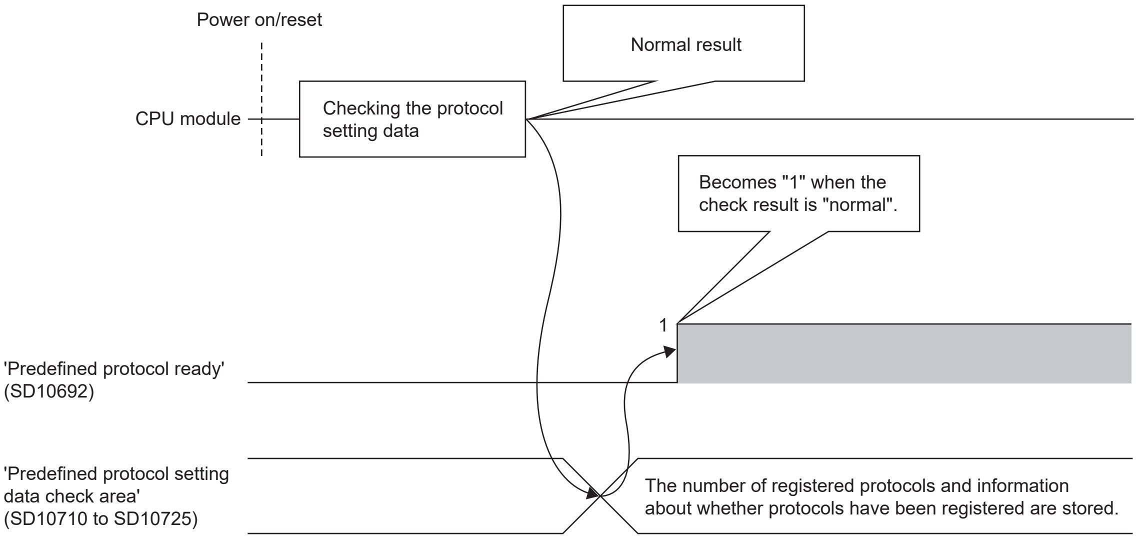
7.4 Execution Conditions of Predefined Protocol Communications

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

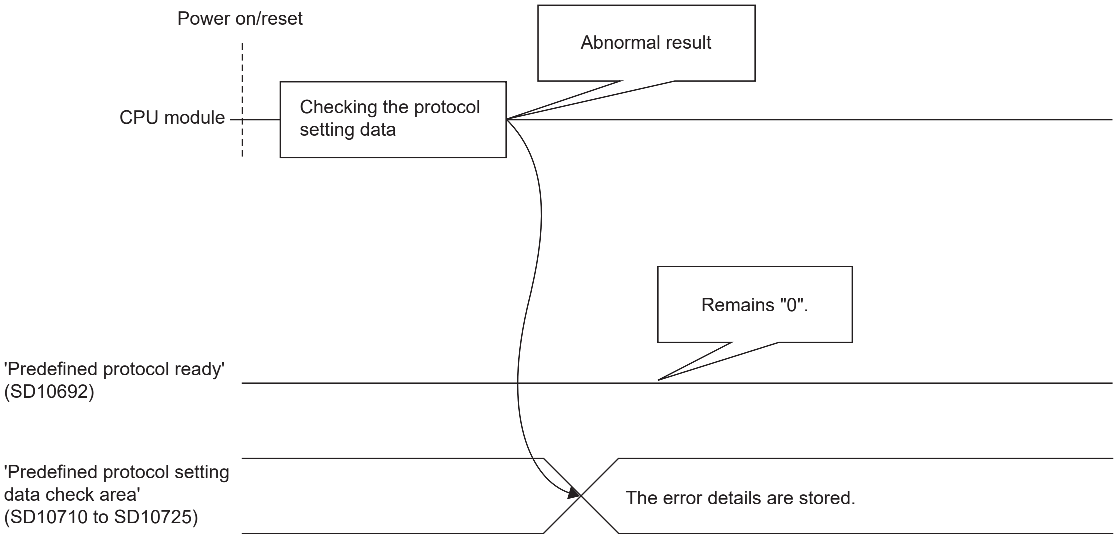
When the system is powered on or reset

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



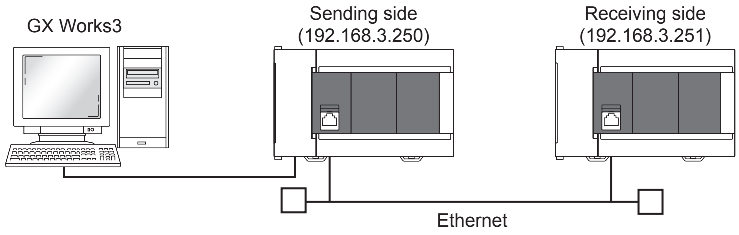
■When protocol setting data is abnormal



7.5 Example of Predefined Protocol Communications

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

System configuration



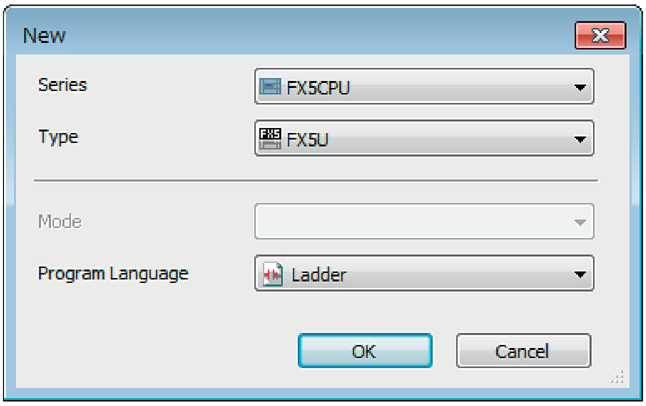
Parameter settings

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

■Sending side

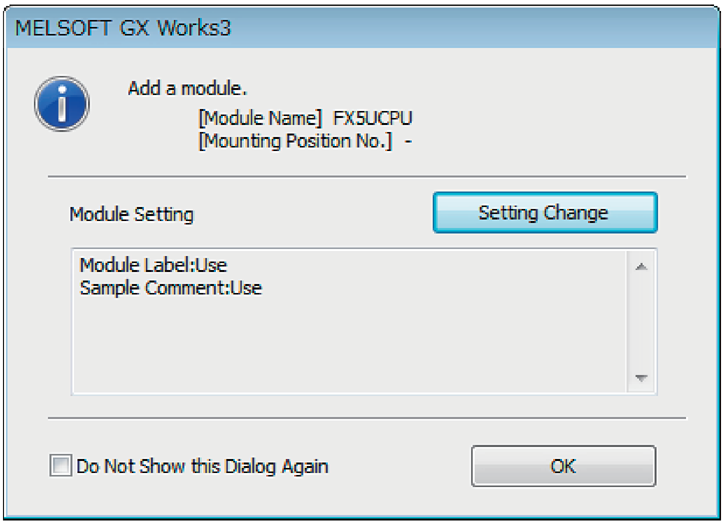
1. Set the CPU module in the following.

[Project] ⇨ [New]



Series: FX5CPU
Type: FX5U
Mode:
Program Language: Ladder

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



Add a module.
[Module Name] FX5UCPU
[Mounting Position No.] -

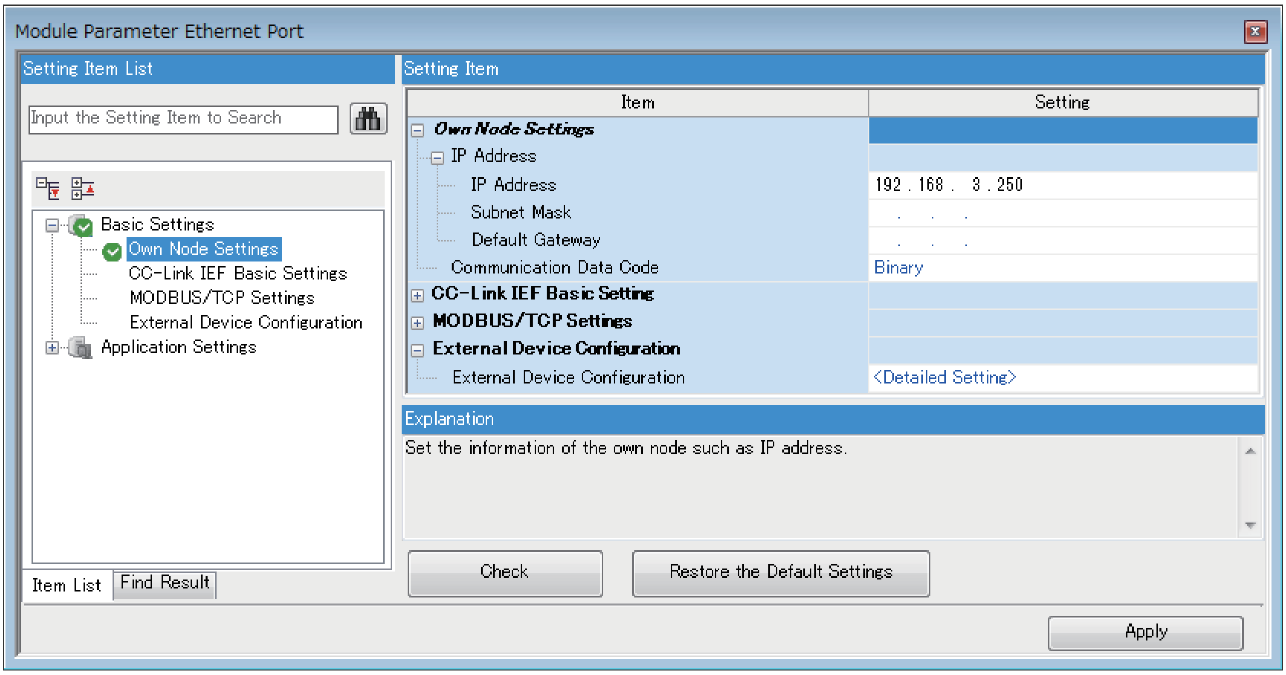
Module Setting

Module Label:Use
Sample Comment:Use

☐ Do Not Show this Dialog Again

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

Navigation window ⇨ [Parameter] ⇨ Module model name ⇨ [Module Parameter] ⇨ [Ethernet Port] ⇨ [Basic Settings]

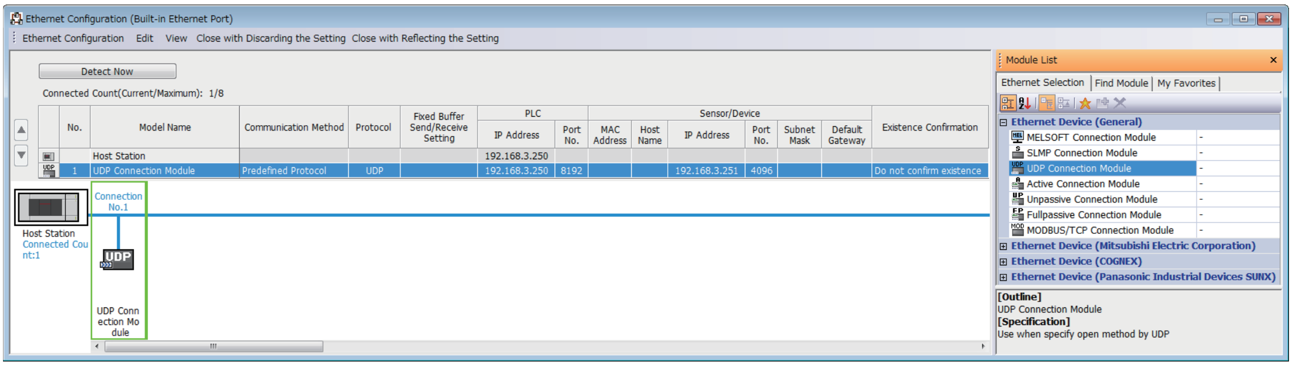


Item	Setting
Own Node Settings	
IP Address	
IP Address	192 . 168 . 3 . 250
Subnet Mask	. . .
Default Gateway	. . .
Communication Data Code	Binary
CC-Link IEF Basic Setting	
MODBUS/TCP Settings	
External Device Configuration	
External Device Configuration	<Detailed Setting>

Set the information of the own node such as IP address.

4. Set the external device configuration in the following.

Navigation window ⇨ [Parameter] ⇨ Module model name ⇨ [Module Parameter] ⇨ [Ethernet Port] ⇨ [Basic Settings] ⇨ [External Device Configuration] ⇨ [Detailed Setting] ⇨ [Ethernet Configuration (Built-in Ethernet Port)] screen



Connected Count(Current/Maximum): 1/8

No.	Model Name	Communication Method	Protocol	Fixed Buffer Send/Receive Setting	PLC		Sensor/Device							Existence Confirmation
					IP Address	Port No.	MAC Address	Host Name	IP Address	Port No.	Subnet Mask	Default Gateway	
	Host Station				192.168.3.250									
1	UDP Connection Module	Predefined Protocol	UDP		192.168.3.250	8192			192.168.3.251	4096			Do not confirm existence

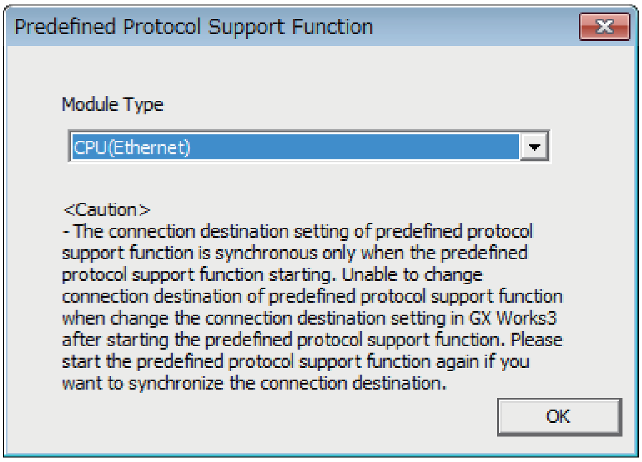
Ethernet Device (General)	
MELSOFT Connection Module	-
SLMP Connection Module	-
UDP Connection Module	-
Active Connection Module	-
Unpassive Connection Module	-
Fullpassive Connection Module	-
MODBUS/TCP Connection Module	-
Ethernet Device (Mitsubishi Electric Corporation)	
Ethernet Device (COGNEX)	
Ethernet Device (Panasonic Industrial Devices SUNX)	

[Outline]
UDP Connection Module
[Specification]
Use when specify open method by UDP

5. Start the predefined protocol support function.

[Tool] ⇨ [Predefined Protocol Support Function]

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



Predefined Protocol Support Function

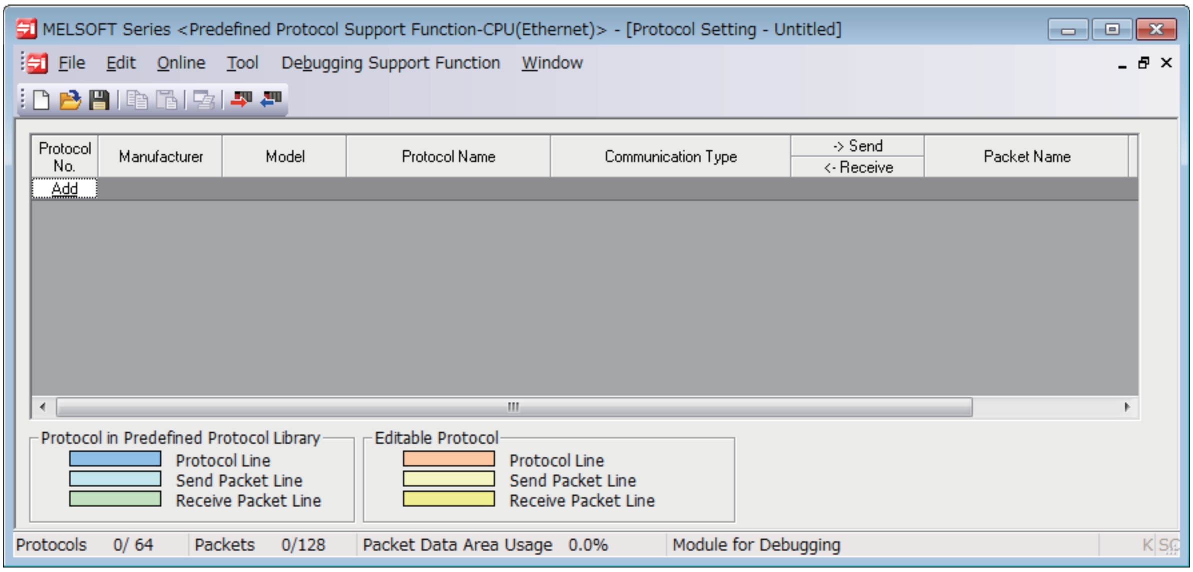
Module Type

CPU(Ethernet)

<Caution>
- The connection destination setting of predefined protocol support function is synchronous only when the predefined protocol support function starting. Unable to change connection destination of predefined protocol support function when change the connection destination setting in GX Works3 after starting the predefined protocol support function. Please start the predefined protocol support function again if you want to synchronize the connection destination.

7. Newly create the protocol setting.

[File] ⇨ [New]

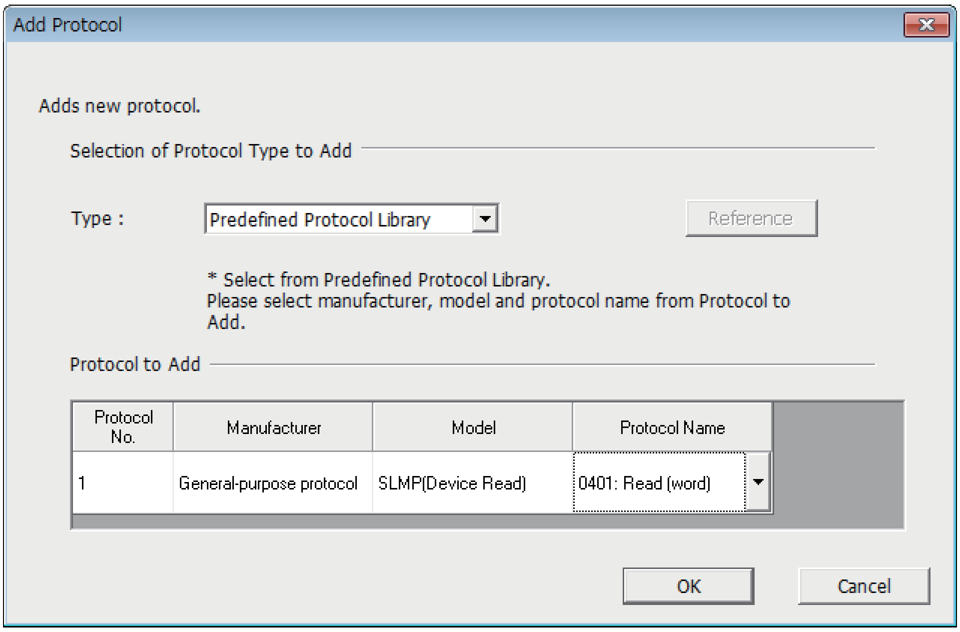


MELSOFT Series <Predefined Protocol Support Function-CPU(Ethernet)> - [Protocol Setting - Untitled]

Protocols 0/ 64 Packets 0/128 Packet Data Area Usage 0.0% Module for Debugging

8. Set a protocol in the following.

[Edit] ⇨ [Add Protocol]



Add Protocol

Adds new protocol.

Selection of Protocol Type to Add

Type : Predefined Protocol Library

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

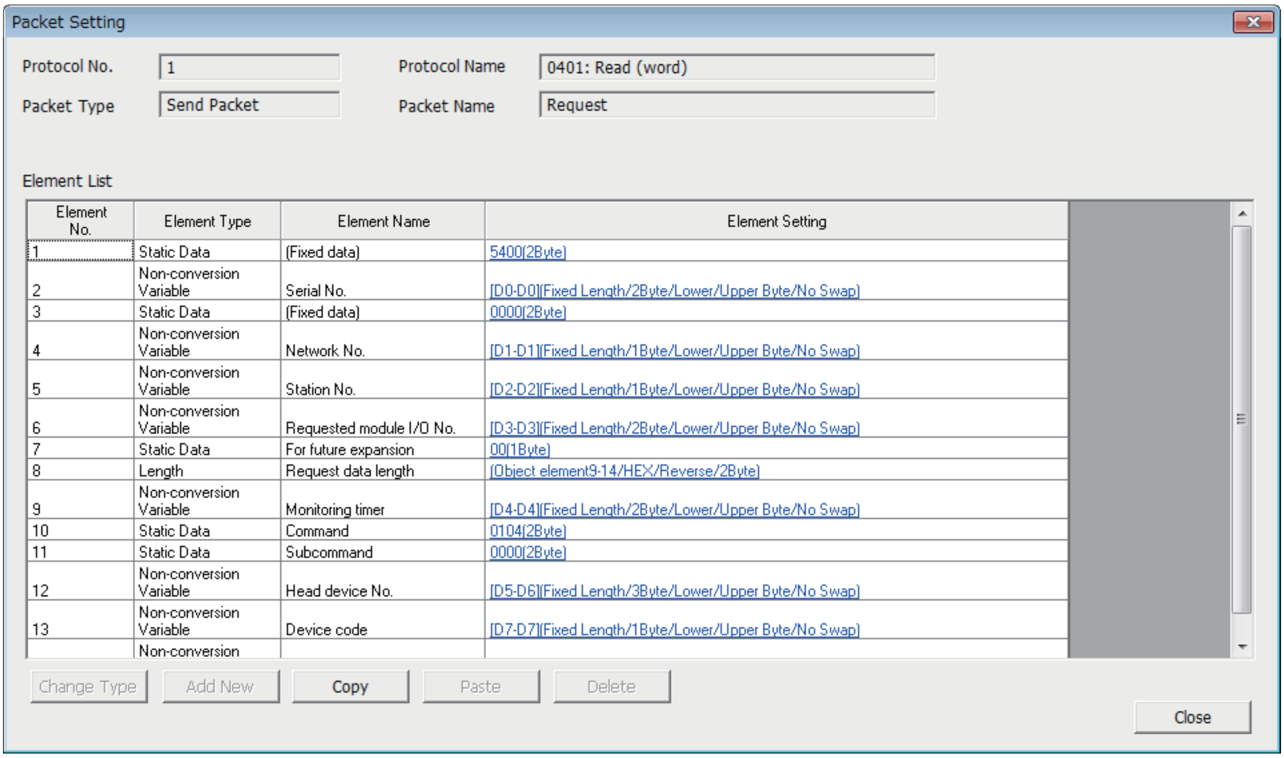
Protocol to Add

Protocol No.	Manufacturer	Model	Protocol Name
1	General-purpose protocol	SLMP(Device Read)	0401: Read (word)

9. Set each packet in the following.

"Protocol Setting" screen ⇨ Packet to be set

- Request

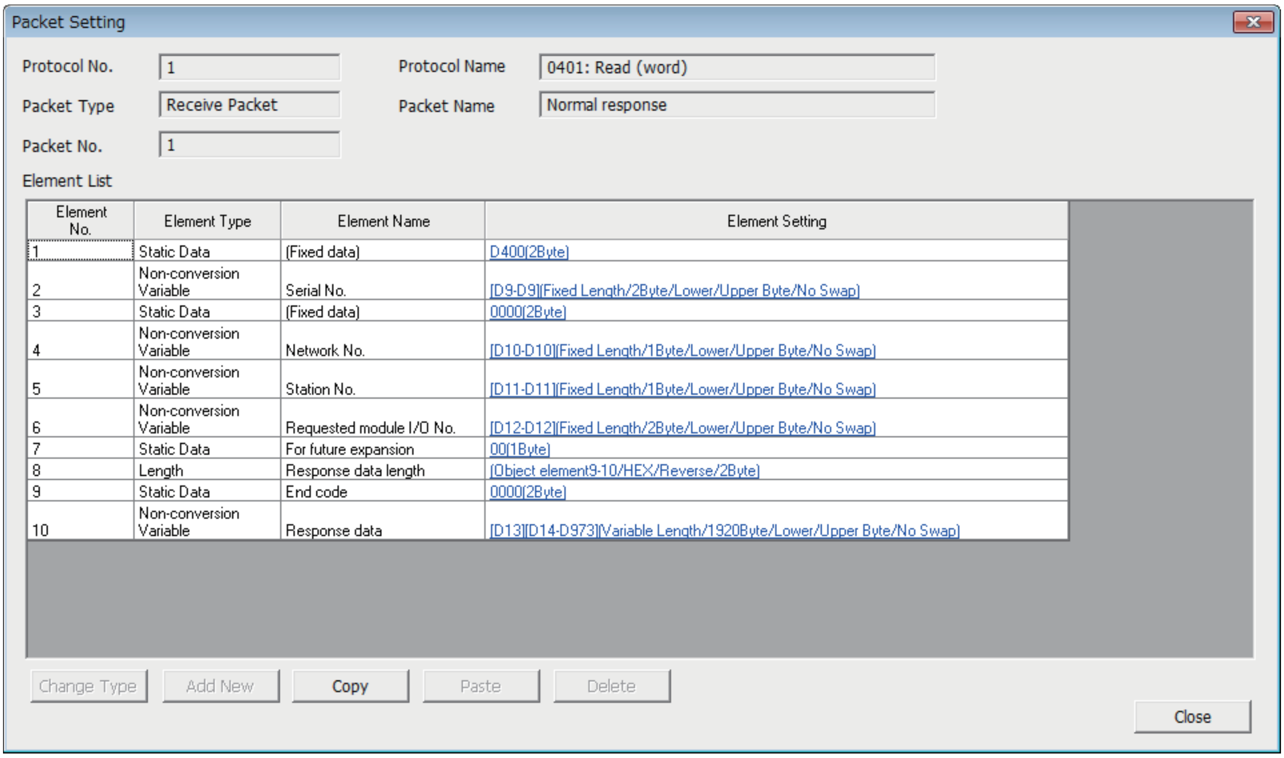


Protocol No.: 1
Protocol Name: 0401: Read (word)
Packet Type: Send Packet
Packet Name: Request

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-D6](Fixed Length/3Byte/Lower/Upper Byte/No Swap)
13	Non-conversion Variable	Device code	[D7-D7](Fixed Length/1Byte/Lower/Upper Byte/No Swap)
	Non-conversion		

- Normal response

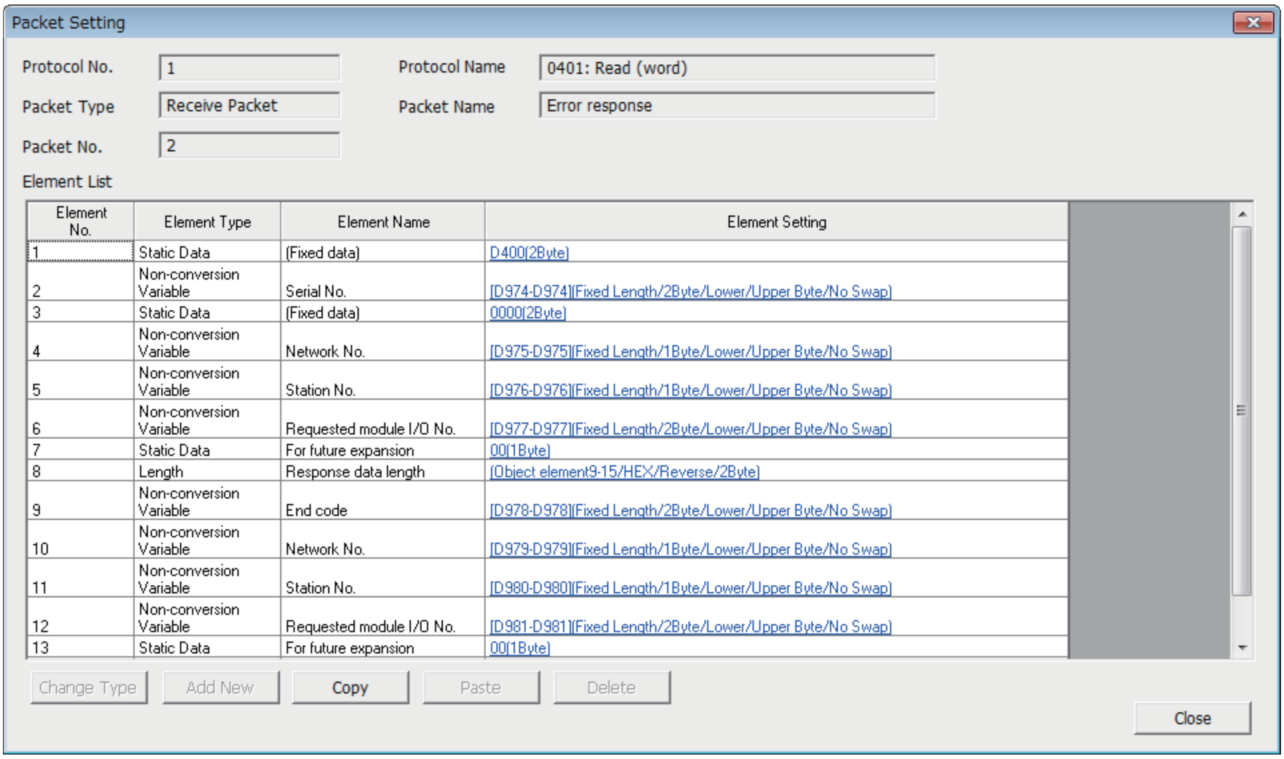


Protocol No.: 1
Protocol Name: 0401: Read (word)
Packet Type: Receive Packet
Packet Name: Normal response
Packet No.: 1

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)

- Error response



Protocol No. 1

Protocol Name 0401: Read (word)

Packet Type Receive Packet

Packet Name Error response

Packet No. 2

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)

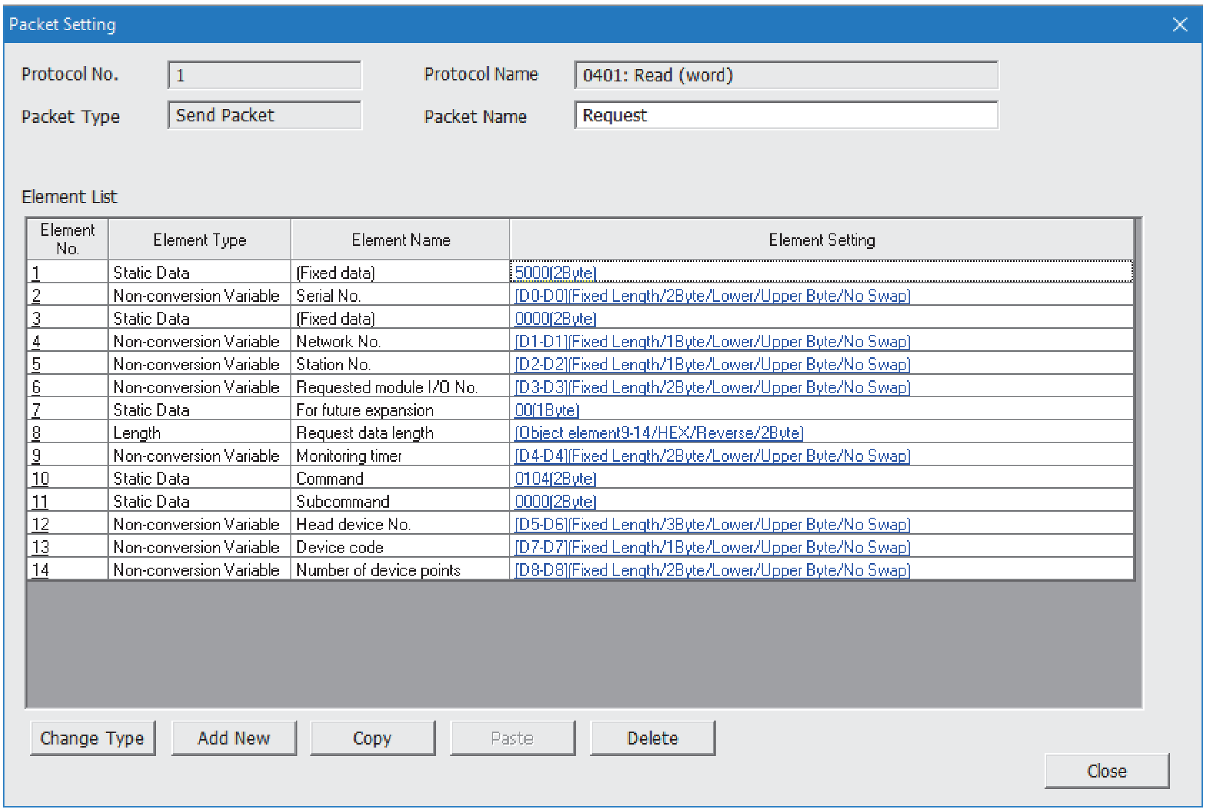
SLMP communication frame for the predefined protocol library is 4E frame in the element. When using 3E frame, configure the settings as below.

- Enable a protocol edit.

[Edit] ⇨ [Editable Protocol]

- Set the element settings for 1 in "Element No." as below.

"Protocol Setting" screen ⇨ Request "Packet Setting" screen



Protocol No. 1

Protocol Name 0401: Read (word)

Packet Type Send Packet

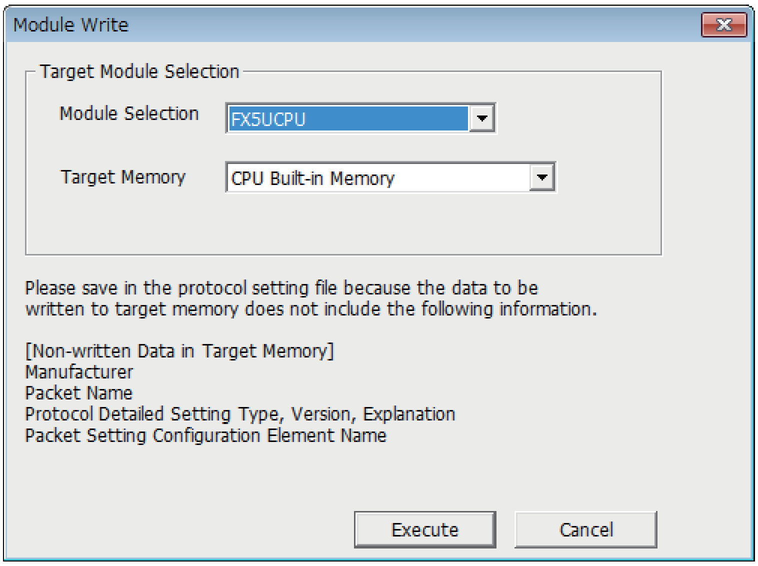
Packet Name Request

Element List

Element No.	Element Type	Element Name	Element Setting
1	Static Data	(Fixed data)	5000(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-D6][Fixed Length/3Byte/Lower/Upper Byte/No Swap]
13	Non-conversion Variable	Device code	[D7-D7][Fixed Length/1Byte/Lower/Upper Byte/No Swap]
14	Non-conversion Variable	Number of device points	[D8-D8][Fixed Length/2Byte/Lower/Upper Byte/No Swap]

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

[Online] ⇨ [Write to Module]



Module Selection: FX5UCPU

Target Memory: CPU Built-in 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

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

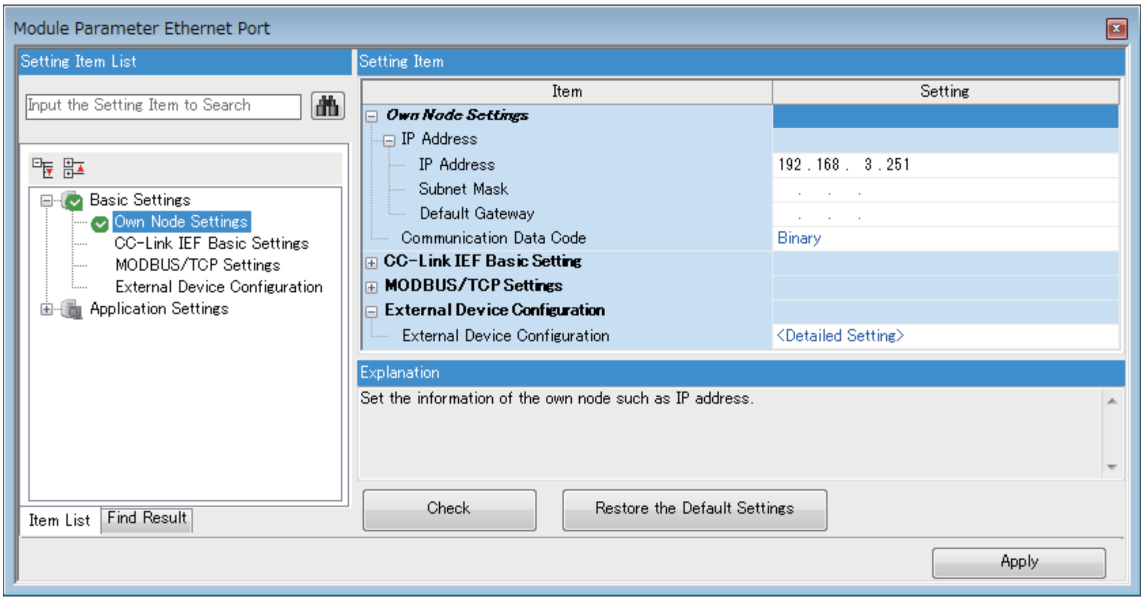
[Online] ⇨ [Write to PLC]

■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 114 Sending side)

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

Navigation window ⇨ [Parameter] ⇨ Module model name ⇨ [Module Parameter] ⇨ [Ethernet Port] ⇨ [Basic Settings]

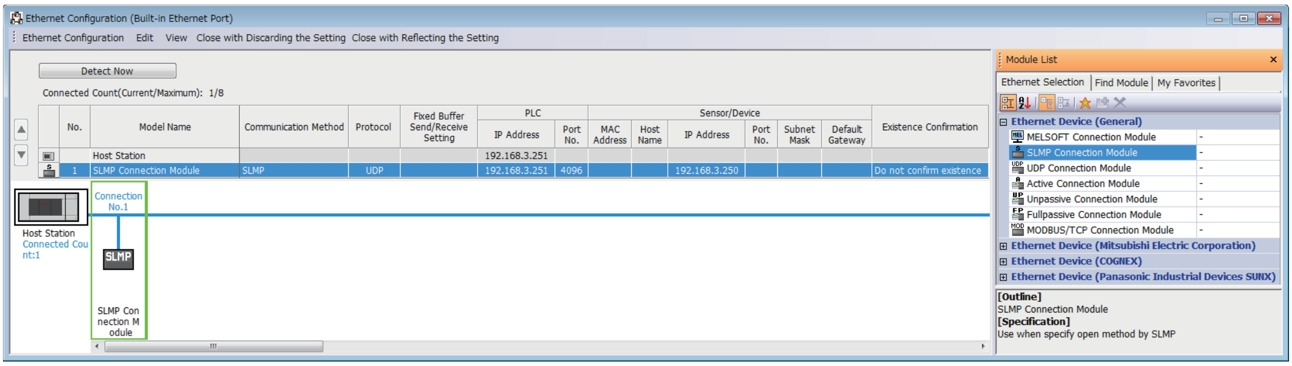


Item	Setting
Own Node Settings	
IP Address	
IP Address	192 . 168 . 3 . 251
Subnet Mask	. . .
Default Gateway	. . .
Communication Data Code	Binary
CC-Link IEF Basic Setting	
MODBUS/TCP Settings	
External Device Configuration	
External Device Configuration	<Detailed Setting>

Set the information of the own node such as IP address.

3. Set the external device configuration in the following.

Navigation window ⇨ [Parameter] ⇨ Module model name ⇨ [Module Parameter] ⇨ [Ethernet Port] ⇨ [Basic Settings] ⇨ [External Device Configuration] ⇨ [Detailed Setting] ⇨ [Ethernet Configuration (Built-in Ethernet Port)] screen



Connected Count(Current/Maximum): 1/8

No.	Model Name	Communication Method	Protocol	Fixed Buffer Send/Receive Setting	PLC		Sensor/Device						Existence Confirmation
					IP Address	Port No.	MAC Address	Host Name	IP Address	Port No.	Subnet Mask	Default Gateway	
	Host Station				192.168.3.251								
1	SLMP Connection Module	SLMP	UDP		192.168.3.251	4096			192.168.3.250				Do not confirm existence

Ethernet Device (General)	
MELSOFT Connection Module	-
SLMP Connection Module	-
UDP Connection Module	-
Active Connection Module	-
Unpassive Connection Module	-
Fullpassive Connection Module	-
MODBUS/TCP Connection Module	-
Ethernet Device (Mitsubishi Electric Corporation)	
Ethernet Device (COGNEX)	
Ethernet Device (Panasonic Industrial Devices SUNX)	

[Outline]
SLMP Connection Module
[Specification]
Use when specify open method by SLMP

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

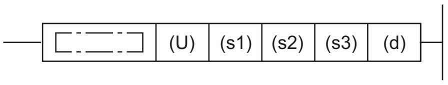
[Online] ⇨ [Write to PLC]

7.6 Predefined Protocol Support Function Instruction

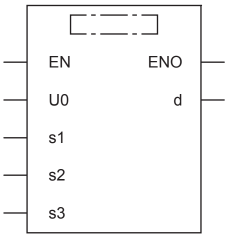
Executing the registered protocols

SP.ECPRTCL

This instruction executes the communication protocol registered using the engineering tool via Ethernet built in the module.

Ladder diagram	Structured text
	ENO: =SP_ECPRTCL (EN, U0, s1, s2, s3, d);

FBD/LD



("SP_ECPRTCL" goes into □.)

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 122)	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 execution result of the SP.ECPRTCL instruction is stored When two or more protocols are executed, the execution result of the protocol executed last is stored. 0: Normal completion Other than 0: Error completion (error code)	—	System
(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	

Device	Item	Description	Setting range	Set by[1]
(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	System

[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 SP.ECPRTCL 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.

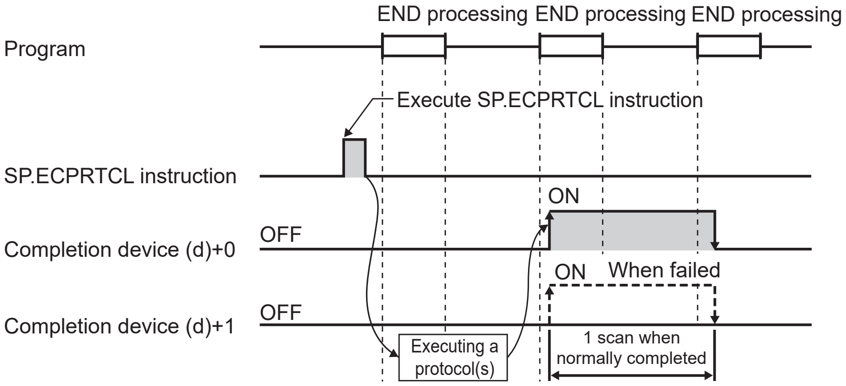
The number of executed protocols is stored in the device specified by (s3)+0.

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

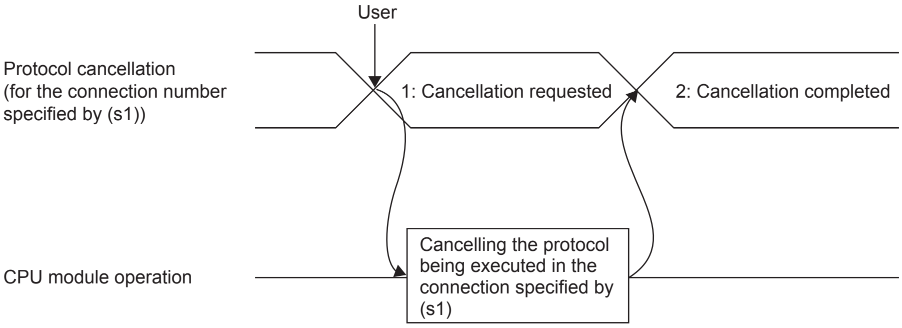
- Completion device (d)+0: The device turns ON during the END processing for the scan in which the SP.ECPRTCL instruction is completed, and turns OFF during the next END processing.
- Completion device (d)+1: Turns ON or OFF depending on the status when the SP.ECPRTCL instruction is completed.

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.



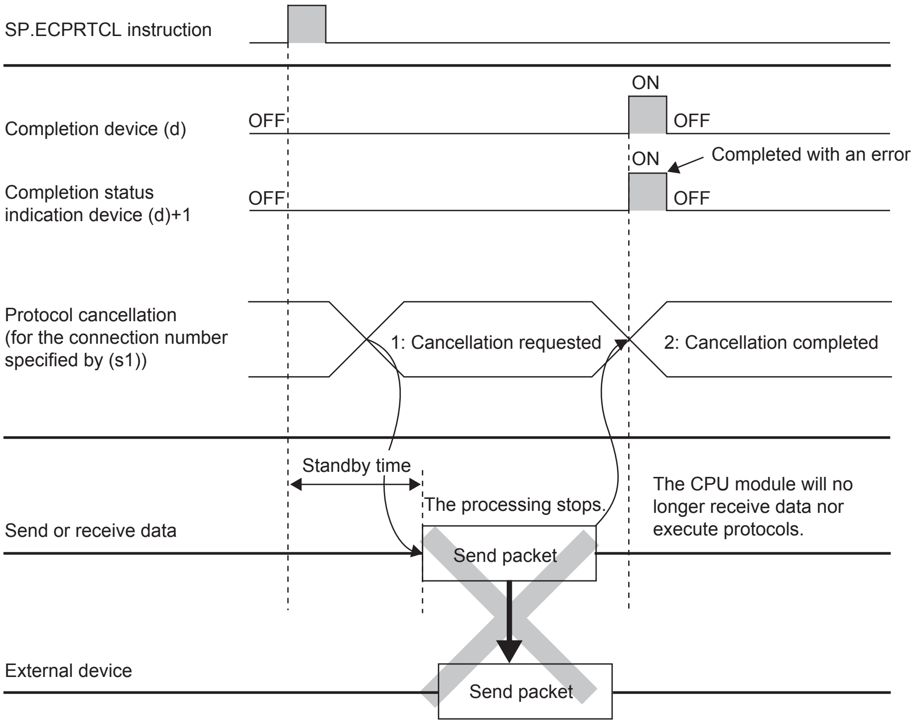
- Protocol execution can be canceled by setting a protocol cancel request. The protocol cancel request is specified in the predefined protocol support function execution status check area (SD10740 to SD10899).



- The following figure shows the protocol cancel operations timing.

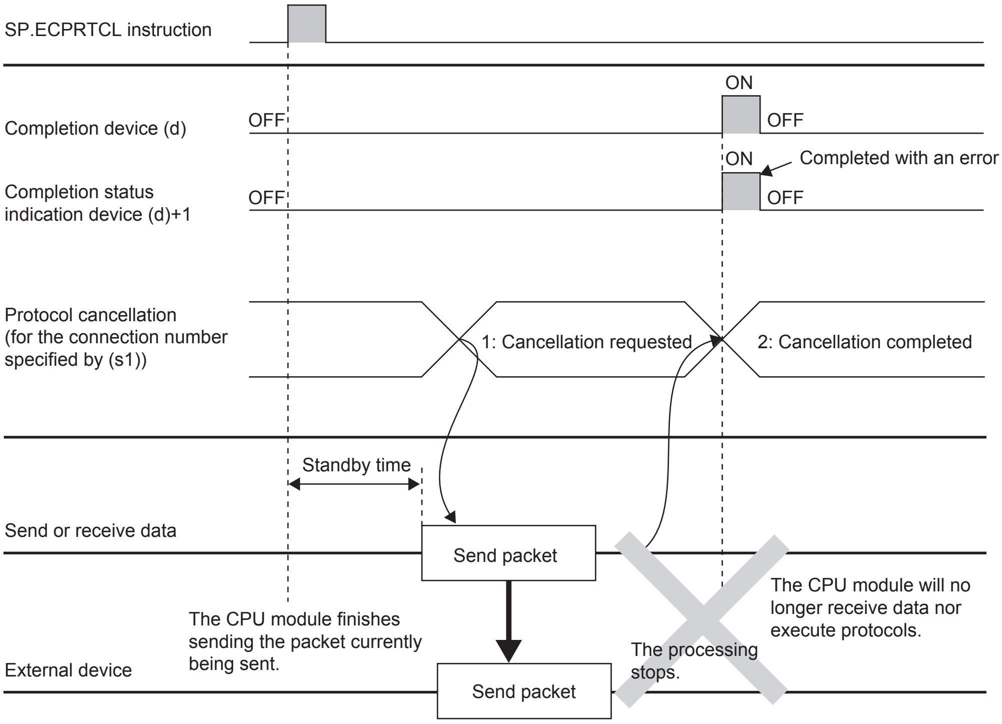
■If a cancel request is issued before transmission

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



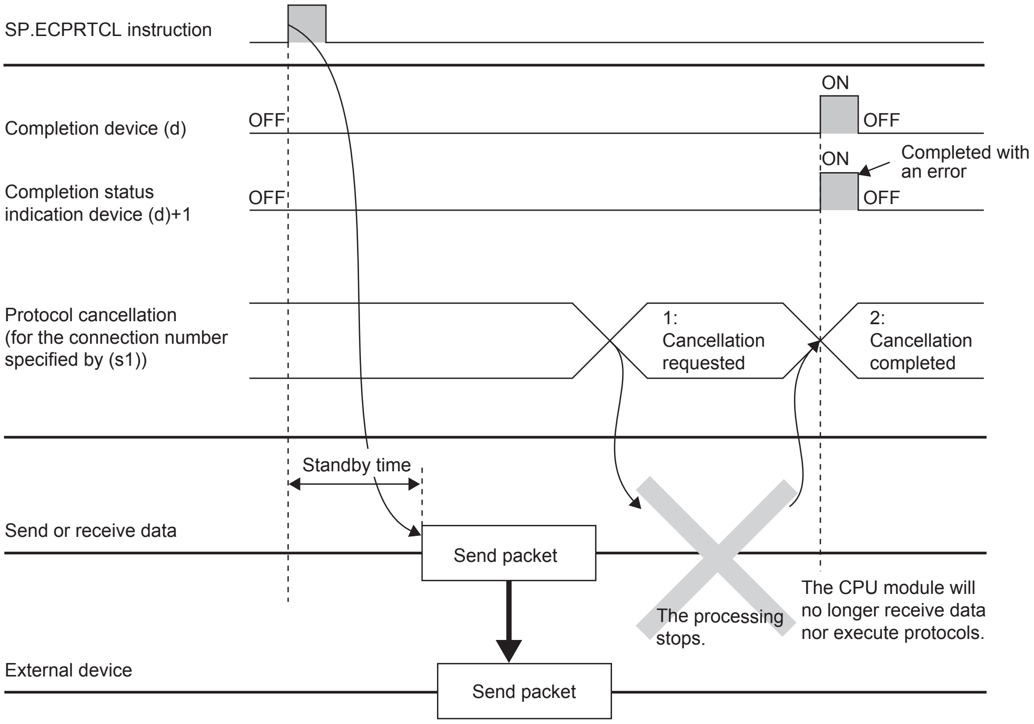
■If a cancel request is issued before completion of transmission

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



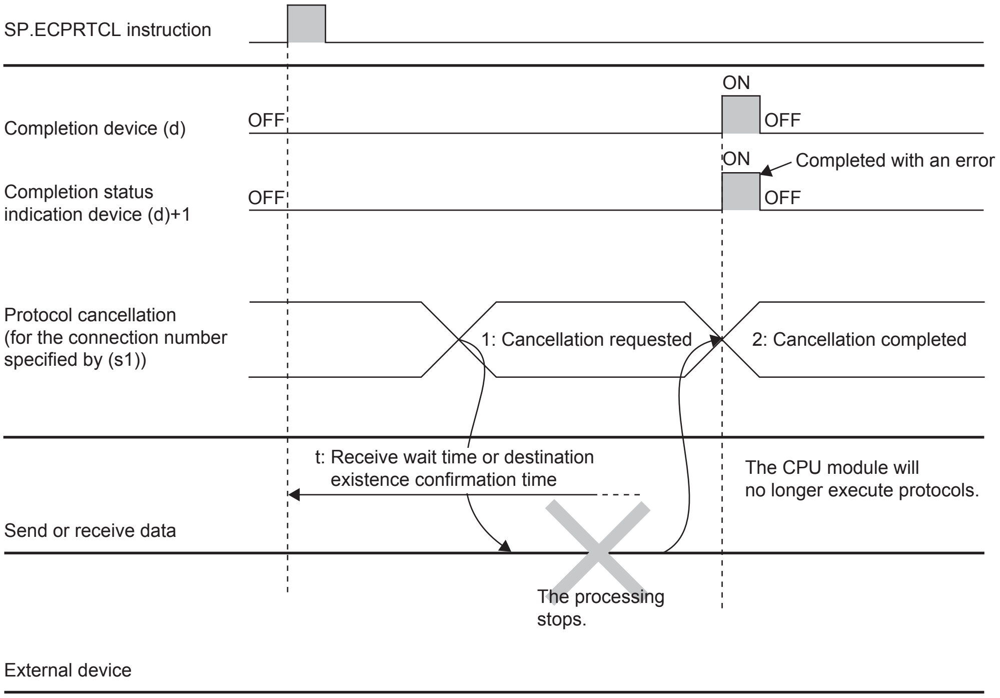
■If a cancel request is issued upon completion of transmission

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



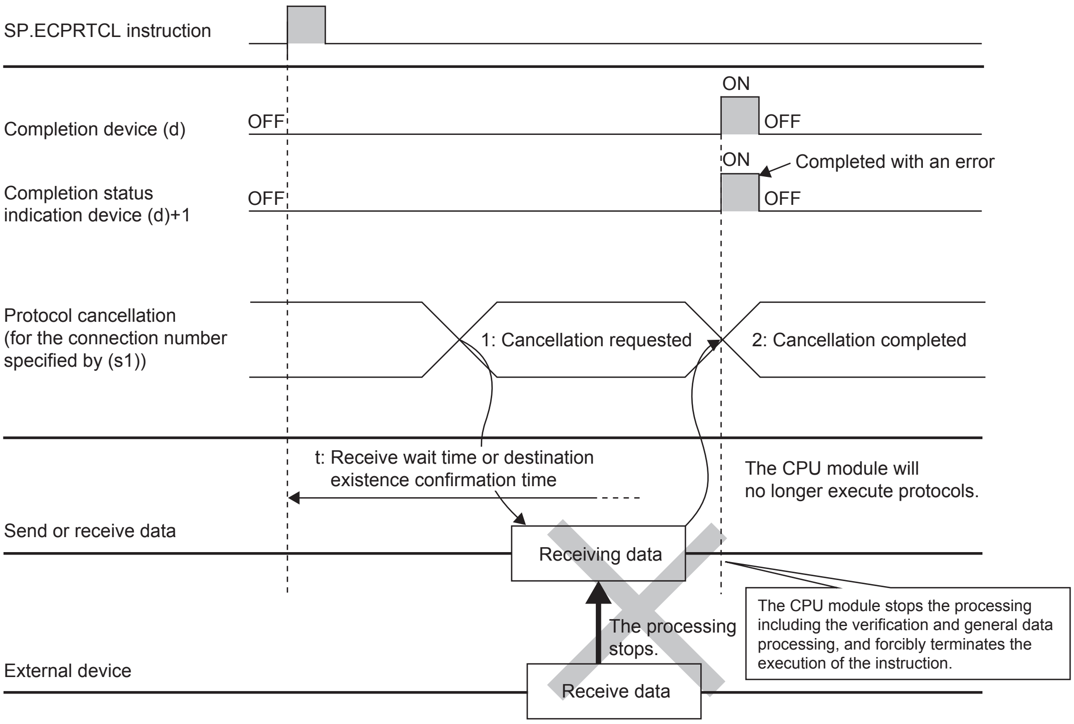
■If a cancel request is issued while waiting for reception

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 following.

☞ Page 143 SP.SOCOPEN

☞ Page 152 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.

7.7 Precautions

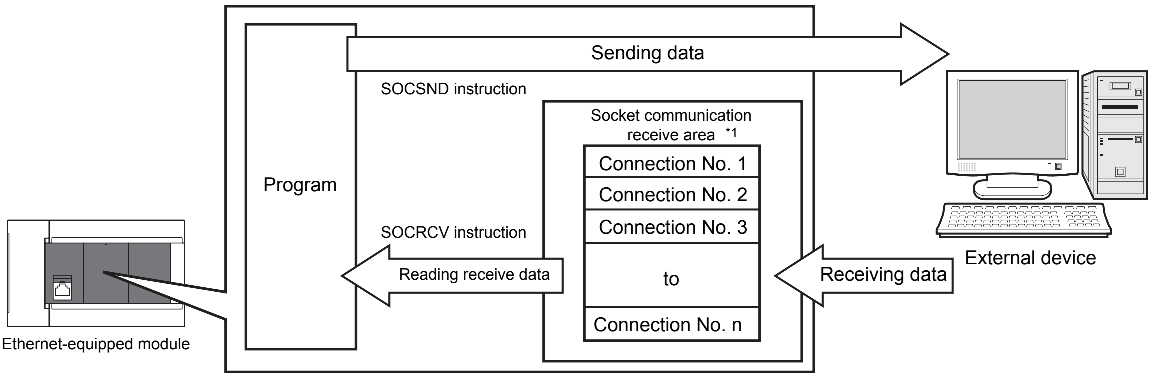
This section provides other precautions for the predefined protocol support function.

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.

8 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

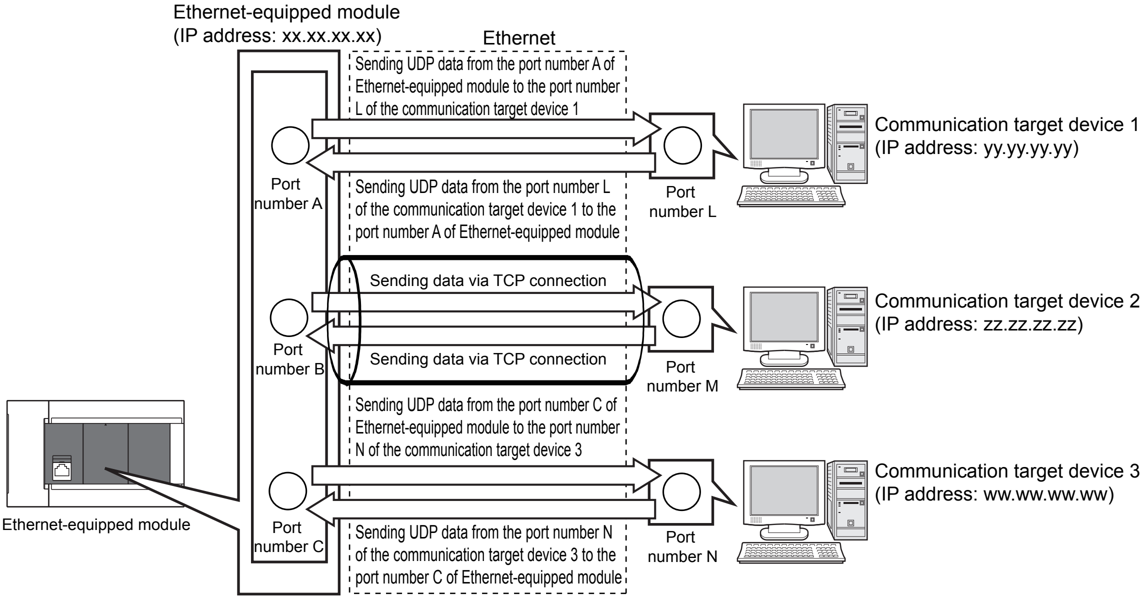
Point

- For dedicated instructions used for the socket communication function, refer to ☞ Page 142 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 46 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.



8.1 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

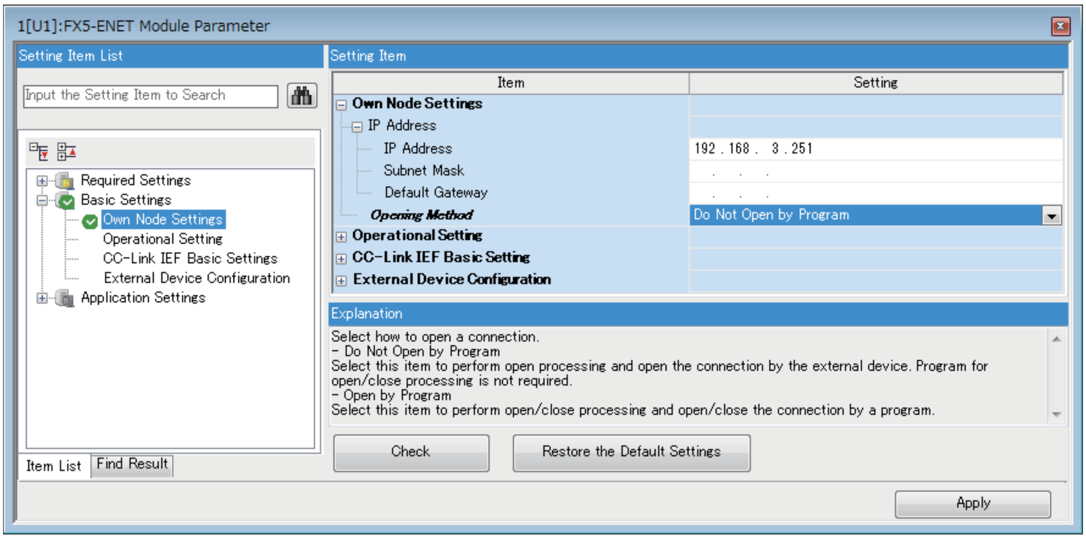
The expressions of Active and Passive opens may vary according to the device.

- 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]



Item	Setting
Own Node Settings	
IP Address	
IP Address	192 . 168 . 3 . 251
Subnet Mask	. . .
Default Gateway	. . .
Opening Method	Do Not Open by Program
Operational Setting	
CC-Link IEF Basic Setting	
External Device Configuration	

Explanation

Select how to open a connection.
- Do Not Open by Program
Select this item to perform open processing and open the connection by the external device. Program for open/close processing is not required.
- Open by Program
Select this item to perform open/close processing and open/close the connection by a program.

■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

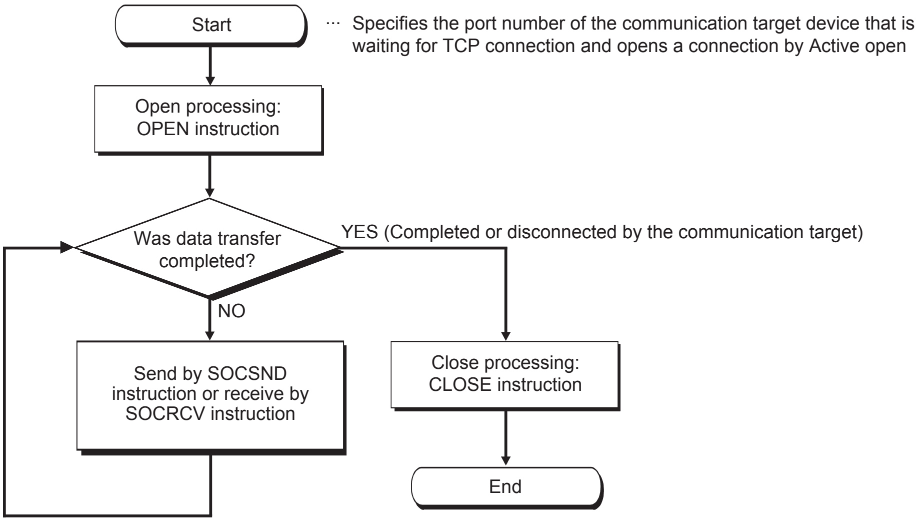
- Change the connection setting before executing the GP.OPEN instruction.
- After the open processing is executed, the open request cannot be canceled until the open processing completes. Execute the close processing (GP.CLOSE instruction) after open completes.

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.

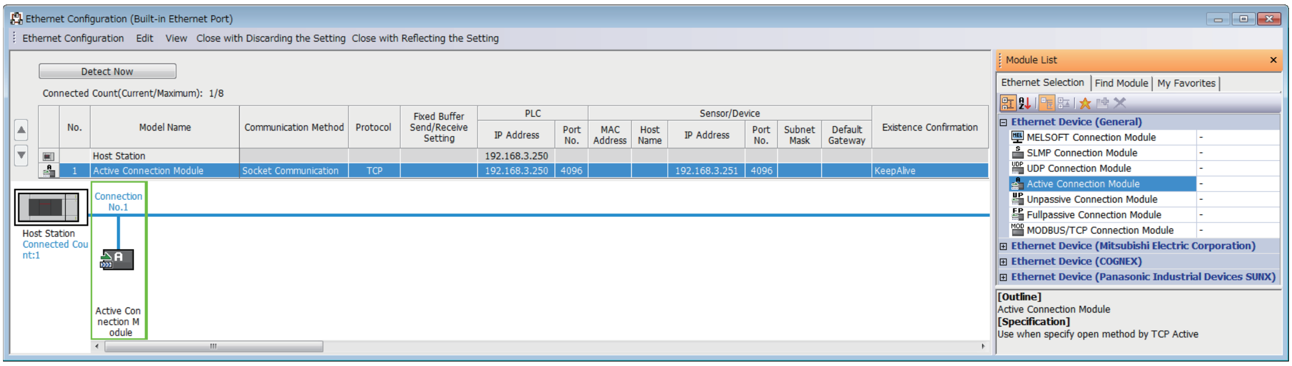


■Parameter setting

The following parameters are set for the sample program.

[CPU module]

Navigation window ⇨ [Parameter] ⇨ Module model name ⇨ [Module Parameter] ⇨ [Ethernet Port] ⇨ [Basic Settings] ⇨ [External Device Configuration] ⇨ [Detailed Setting] ⇨ [Ethernet Configuration (Built-in Ethernet Port)] screen



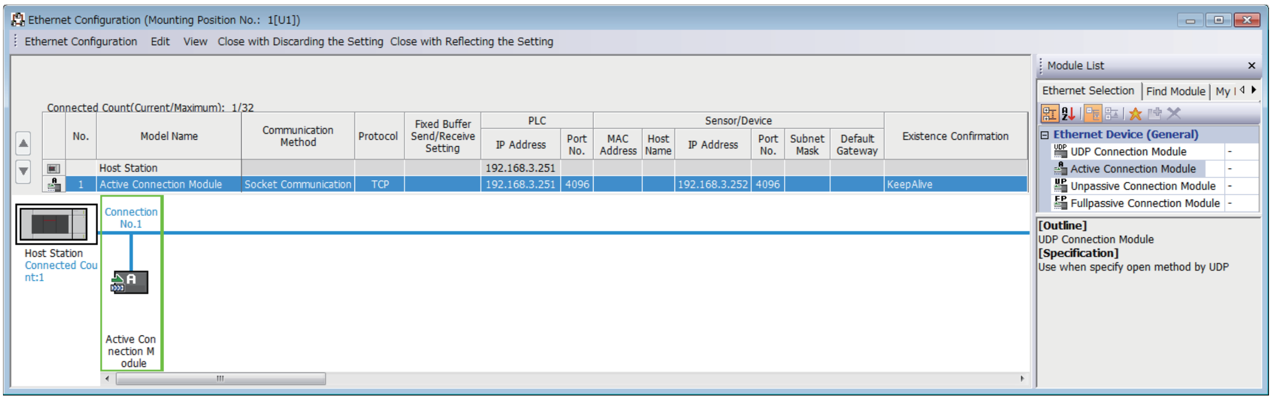
Connected Count(Current/Maximum): 1/8

No.	Model Name	Communication Method	Protocol	Fixed Buffer Send/Receive Setting	PLC IP Address	PLC Port No.	Sensor/Device MAC Address	Sensor/Device Host Name	Sensor/Device IP Address	Sensor/Device Port No.	Sensor/Device Subnet Mask	Sensor/Device Default Gateway	Existence Confirmation
	Host Station				192.168.3.250								
1	Active Connection Module	Socket Communication	TCP		192.168.3.250	4096			192.168.3.251	4096			KeepAlive

[Outline]
Active Connection Module
[Specification]
Use when specify open method by TCP Active

[Ethernet module]

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



Connected Count(Current/Maximum): 1/32

No.	Model Name	Communication Method	Protocol	Fixed Buffer Send/Receive Setting	PLC IP Address	PLC Port No.	Sensor/Device MAC Address	Sensor/Device Host Name	Sensor/Device IP Address	Sensor/Device Port No.	Sensor/Device Subnet Mask	Sensor/Device Default Gateway	Existence Confirmation
	Host Station				192.168.3.251								
1	Active Connection Module	Socket Communication	TCP		192.168.3.251	4096			192.168.3.252	4096			KeepAlive

[Outline]
UDP Connection Module
[Specification]
Use when specify open method by UDP

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

Item		Description
PLC	Port No.	4096 (Setting range: 1 to 5548, 5570 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)

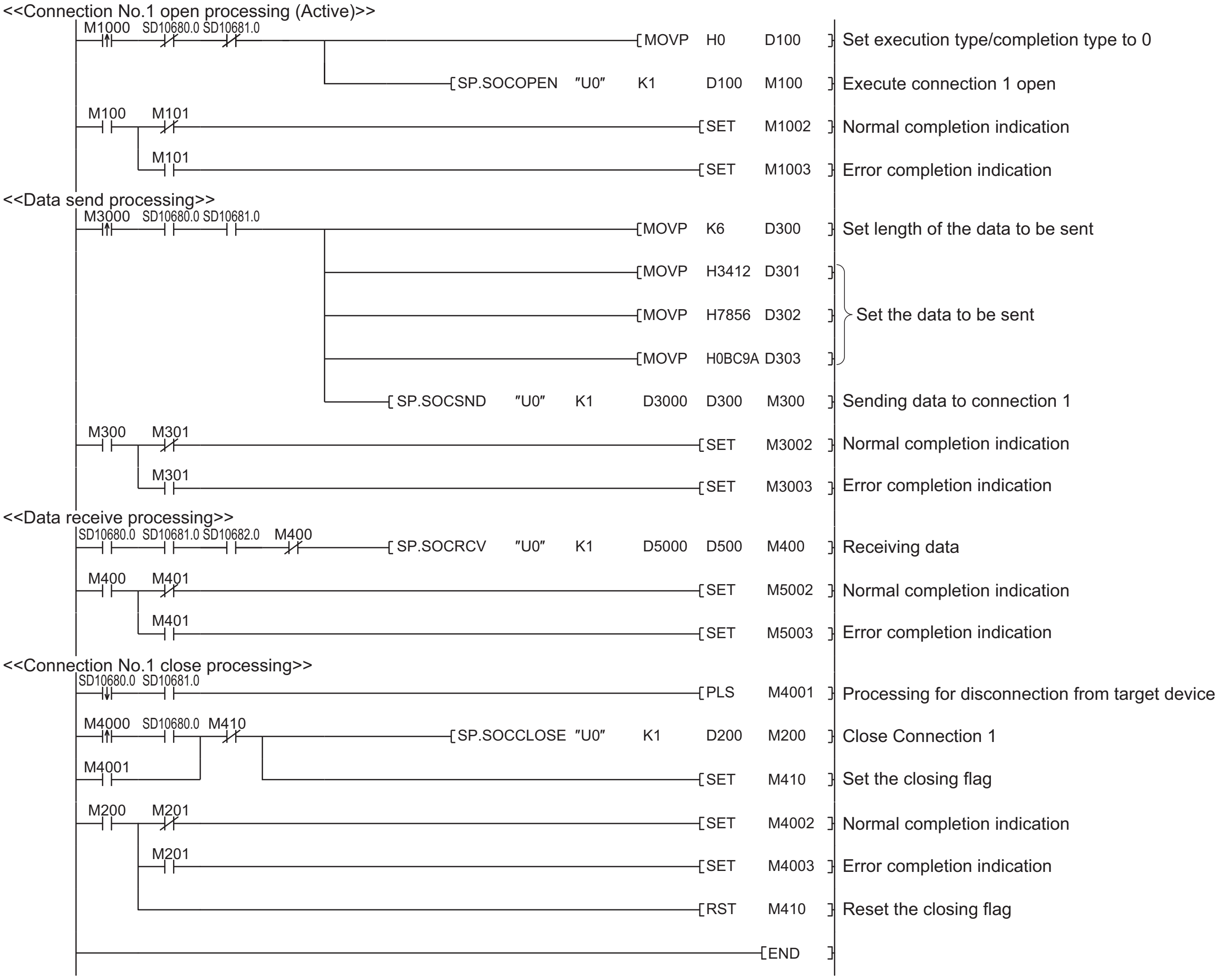
■Devices used in the sample program

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 or later	Receive data length and receive data
M5002	Normal receive indication
M5003	Receive error indication

■Sample program

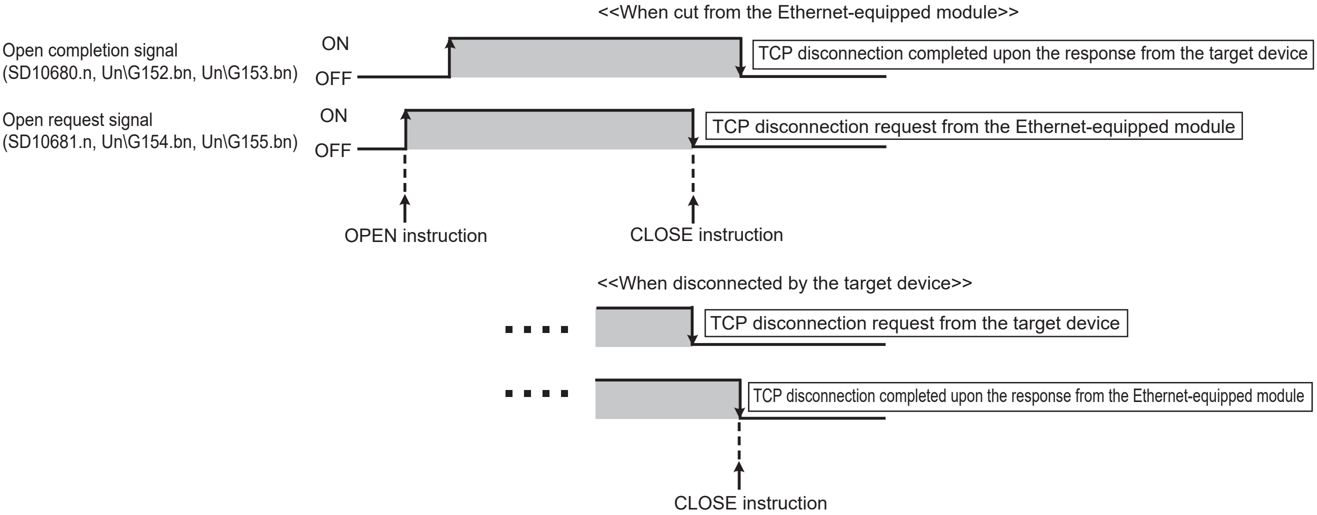
A program example using the CPU module is shown below.



■Precautions for Active open communication

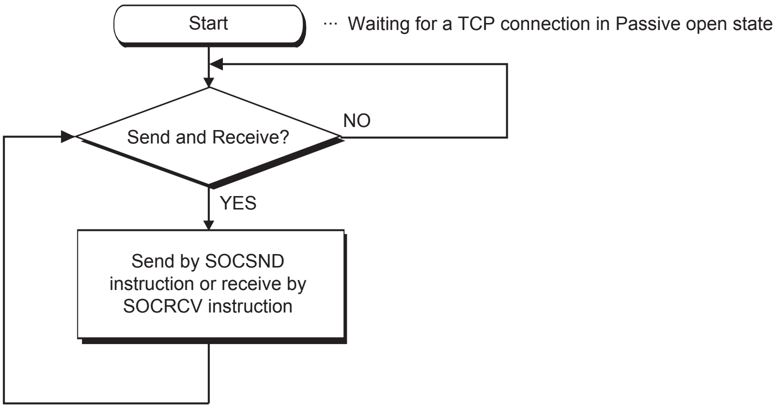
Configure an interlock circuit using the Open completion signal (SD10680.n, Un\G152.bn, Un\G153.bn) and Open request signal (SD10681.n, Un\G154.bn, Un\G155.bn) in the program.

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



Program example for Passive open

The following shows the communication flow of a Passive open.

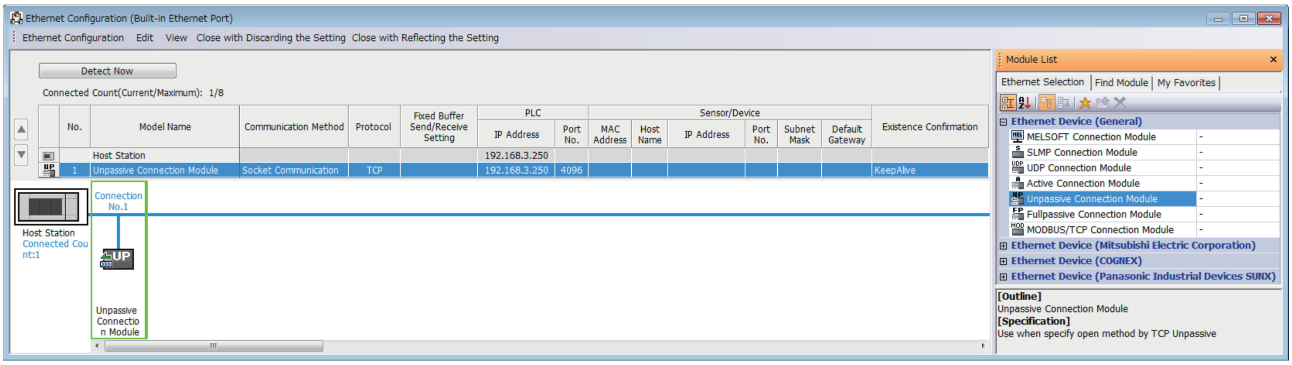


■Parameter setting

The following parameters are set for the sample program.

[CPU module]

Navigation window ⇨ [Parameter] ⇨ Module model name ⇨ [Module Parameter] ⇨ [Ethernet Port] ⇨ [Basic Settings] ⇨ [External Device Configuration] ⇨ [Detailed Setting] ⇨ [Ethernet Configuration (Built-in Ethernet Port)] screen



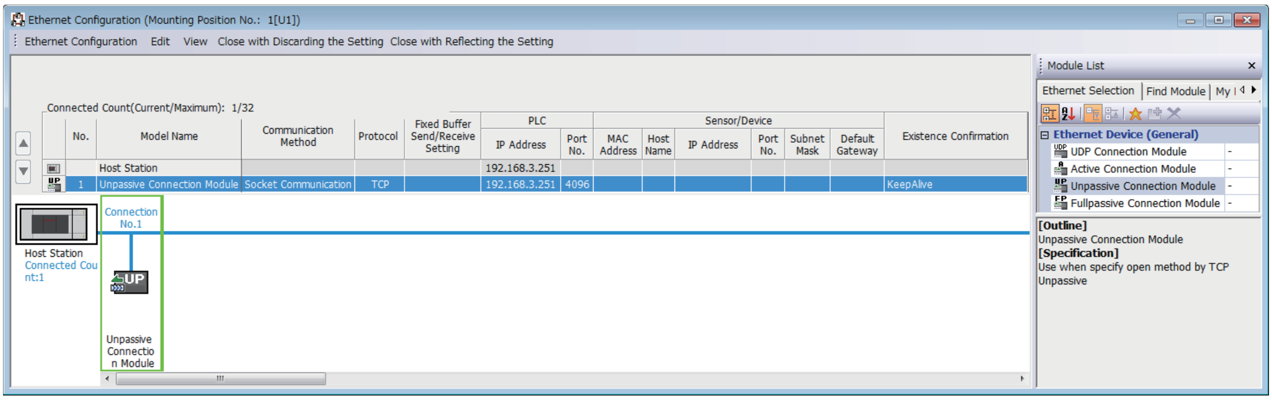
Connected Count(Current/Maximum): 1/8

No.	Model Name	Communication Method	Protocol	Fixed Buffer Send/Receive Setting	PLC		Sensor/Device						Existence Confirmation
					IP Address	Port No.	MAC Address	Host Name	IP Address	Port No.	Subnet Mask	Default Gateway	
	Host Station				192.168.3.250								
1	Unpassive Connection Module	Socket Communication	TCP		192.168.3.250	4096							KeepAlive

[Outline]
Unpassive Connection Module
[Specification]
Use when specify open method by TCP Unpassive

[Ethernet module]

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



Connected Count(Current/Maximum): 1/32

No.	Model Name	Communication Method	Protocol	Fixed Buffer Send/Receive Setting	PLC		Sensor/Device						Existence Confirmation
					IP Address	Port No.	MAC Address	Host Name	IP Address	Port No.	Subnet Mask	Default Gateway	
	Host Station				192.168.3.251								
1	Unpassive Connection Module	Socket Communication	TCP		192.168.3.251	4096							KeepAlive

[Outline]
Unpassive Connection Module
[Specification]
Use when specify open method by TCP Unpassive

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

Item		Description
PLC	Port No.	4096 (Setting range: 1 to 5548, 5570 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)

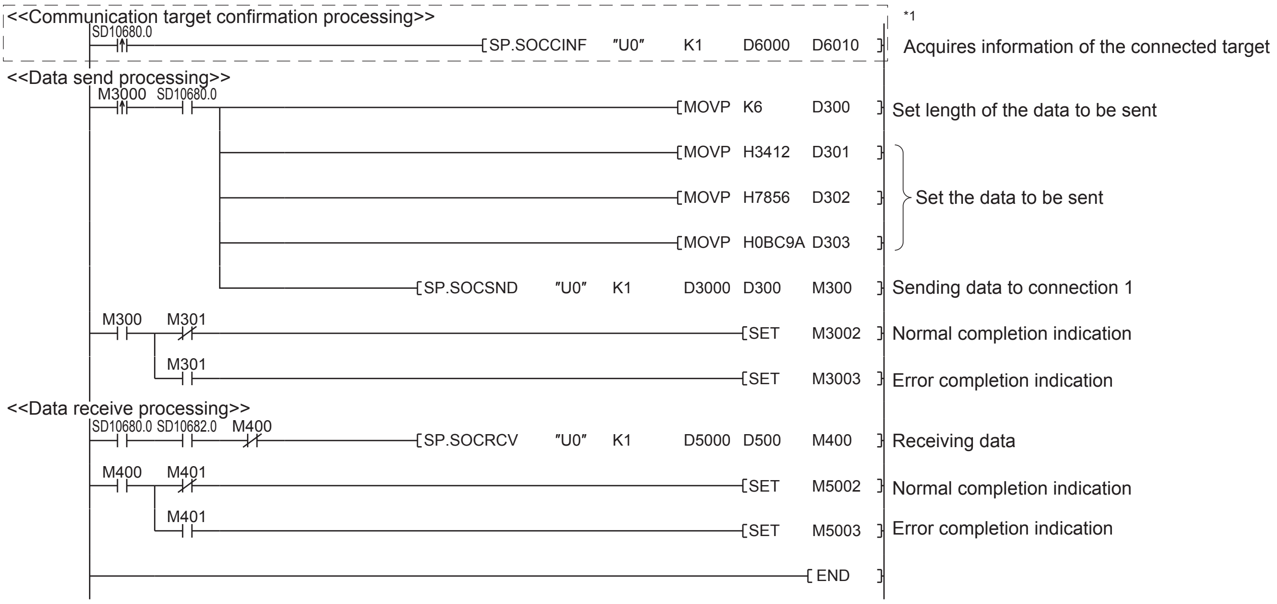
■Devices used in the sample program

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 or later	Receive data length and receive data
M5002	Normal receive indication
M5003	Receive error indication
D6000 and D6001	SP.SOCCINF instruction control data
D6010 to D6014	SP.SOCCINF instruction connection information

■Sample program

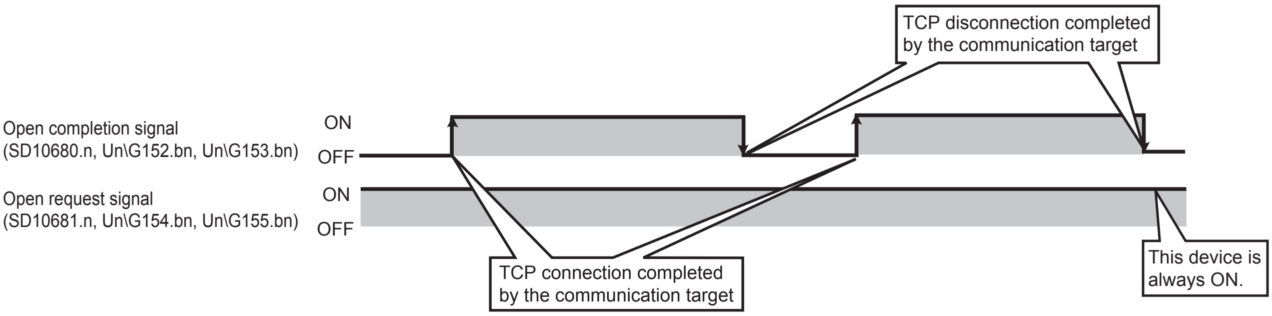
A program example using the CPU module is shown below.



[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.

8.2 Communication Using UDP

UDP (User Datagram Protocol) is a simple protocol that does not perform data sequencing and retransmission.

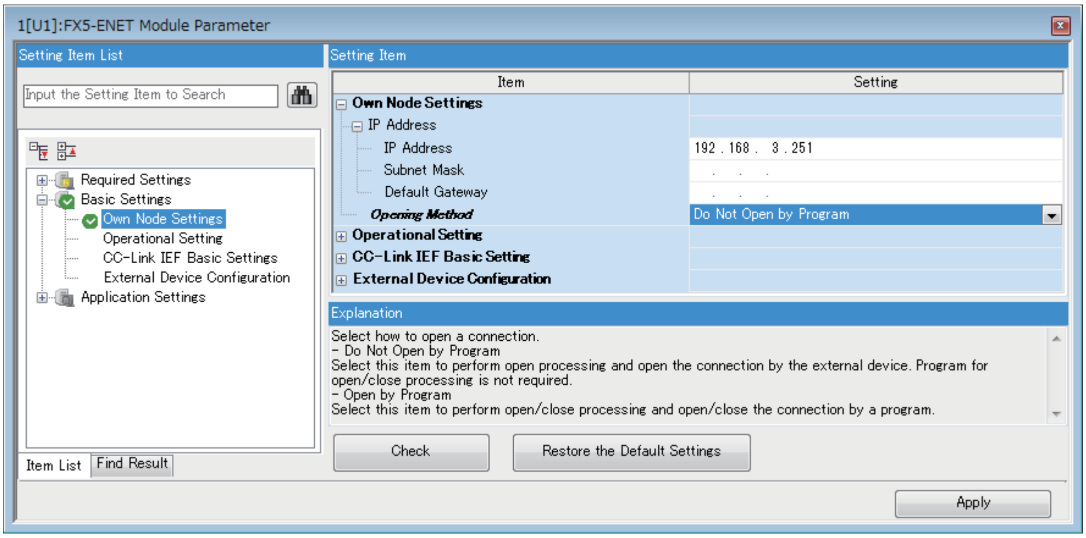
To perform socket communication using UDP, confirm the following in advance.

- 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]



Item	Setting
Own Node Settings	
IP Address	
IP Address	192 . 168 . 3 . 251
Subnet Mask	. . .
Default Gateway	. . .
Opening Method	Do Not Open by Program
Operational Setting	
CC-Link IEF Basic Setting	
External Device Configuration	

Select how to open a connection.
- Do Not Open by Program
Select this item to perform open processing and open the connection by the external device. Program for open/close processing is not required.
- Open by Program
Select this item to perform open/close processing and open/close the connection by a program.

■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.

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 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.

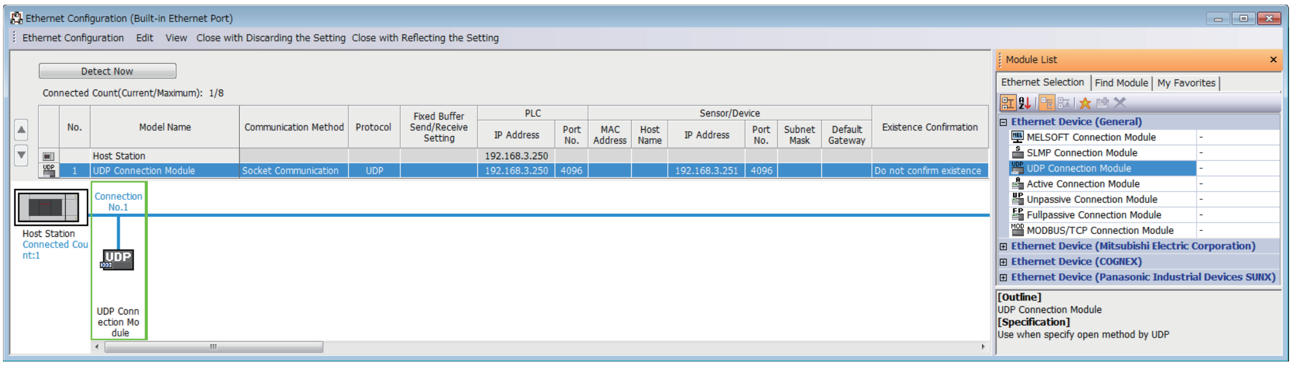
Program example

This section provides a program example for communication using UDP.

Parameter setting

[CPU module]

Navigation window ⇨ [Parameter] ⇨ Module model name ⇨ [Module Parameter] ⇨ [Ethernet Port] ⇨ [Basic Settings] ⇨ [External Device Configuration] ⇨ [Detailed Setting] ⇨ [Ethernet Configuration (Built-in Ethernet Port)] screen



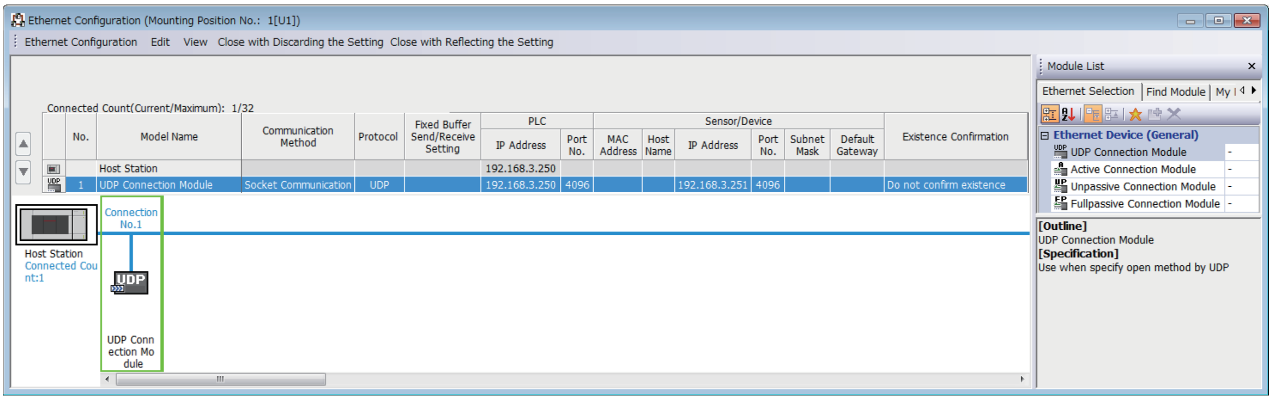
Connected Count(Current/Maximum): 1/8

No.	Model Name	Communication Method	Protocol	Fixed Buffer Send/Receive Setting	PLC IP Address	PLC Port No.	Sensor/Device MAC Address	Sensor/Device Host Name	Sensor/Device IP Address	Sensor/Device Port No.	Sensor/Device Subnet Mask	Sensor/Device Default Gateway	Existence Confirmation
	Host Station				192.168.3.250								
1	UDP Connection Module	Socket Communication	UDP		192.168.3.250	4096			192.168.3.251	4096			Do not confirm existence

[Outline]
UDP Connection Module
[Specification]
Use when specify open method by UDP

[Ethernet module]

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



Connected Count(Current/Maximum): 1/32

No.	Model Name	Communication Method	Protocol	Fixed Buffer Send/Receive Setting	PLC IP Address	PLC Port No.	Sensor/Device MAC Address	Sensor/Device Host Name	Sensor/Device IP Address	Sensor/Device Port No.	Sensor/Device Subnet Mask	Sensor/Device Default Gateway	Existence Confirmation
	Host Station				192.168.3.250								
1	UDP Connection Module	Socket Communication	UDP		192.168.3.250	4096			192.168.3.251	4096			Do not confirm existence

[Outline]
UDP Connection Module
[Specification]
Use when specify open method by UDP

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

Item		Description
PLC	Port No.	4096 (Setting range: 1 to 5548, 5570 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 65534/65535)

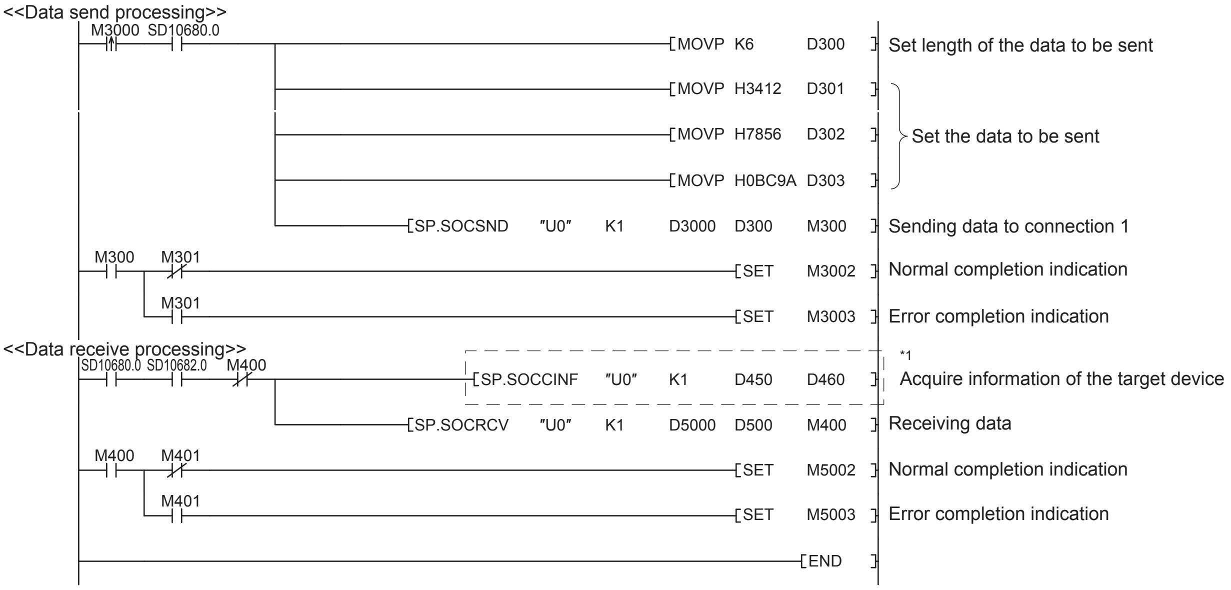
Devices used in the sample program

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
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 or later	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 143 Opening a connection.

8.3 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 143 SP.SOCOPEN
SP.SOCCLOSE	Closes a connection.	Page 152 SP.SOCCLOSE
SP.SOCRCV	Reads the data received. (Read at END processing)	Page 158 SP.SOCRCV
SP.SOCSND	Sends data.	Page 165 SP.SOCSND
SP.SOCCINF	Reads connection information.	Page 172 SP.SOCCINF
S(P).SOCRDATA	Reads data from the socket communication receive data area.	Page 174 S(P).SOCRDATA

Ethernet module dedicated instruction

Instruction	Description	Reference
GP.OPEN	Establishes a connection.	Page 147 GP.OPEN
GP.CLOSE	Closes a connection.	Page 155 GP.CLOSE
GP.SOCRCV	Reads the data received.	Page 162 GP.SOCRCV
GP.SOCSND	Sends data.	Page 169 GP.SOCSND

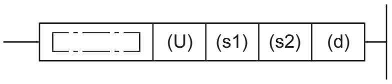
Point

- For configuration of data communication using the socket communication function, refer to ☞ Page 130 Communication Using TCP and ☞ Page 138 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 816 ERROR CODES or 📖 MELSEC iQ-F FX5 User's Manual (Application).

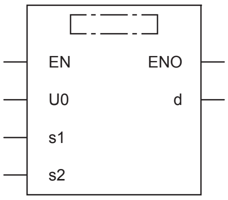
Opening a connection

SP.SOCOPEN

Establishes a connection. (CPU module dedicated instruction)

Ladder diagram	Structured text
	ENO: =SP_SOCOPEN(EN, U0, s1, s2, d);

FBD/LD



("SP_SOCOPEN" goes into □.)

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)	Head device number for storing the control data	Refer to Control data (☞ Page 144)	Word	ANY16_ARRAY (Number of elements: 10)
(d)	Head device number which turns ON when the execution of the instruction is completed and remains on for 1 scan. (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.

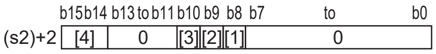
*2 Regardless of the program language to be used, the data type is specified by a device. Do not specify a label.

■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) For the error codes, refer to ☞ Page 816 ERROR CODES.	—	System
(s2) + 2	Application setting area	 [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 system 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 FFFEH)[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 FFFEH)	
(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] 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 FFFEH) 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

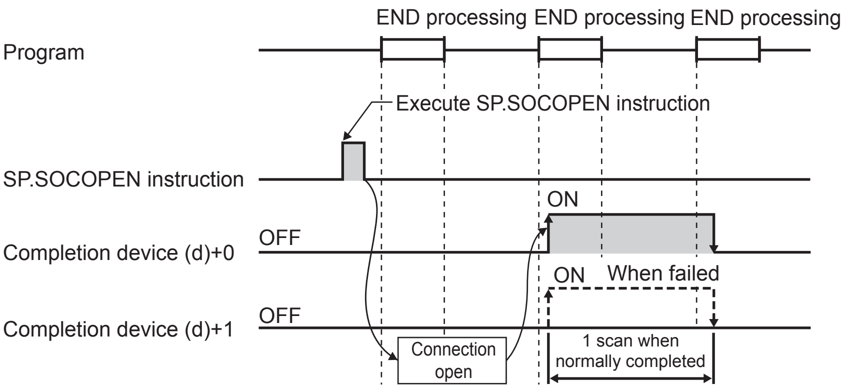
This instruction opens a connection specified in (s1).

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

The result of the 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

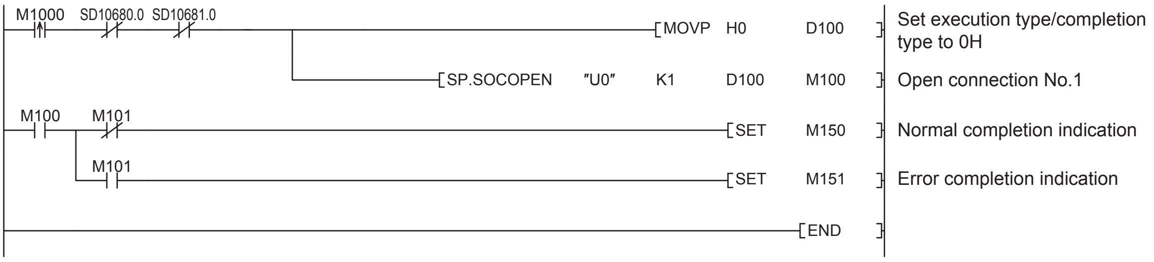
■Opening a connection using parameter settings

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



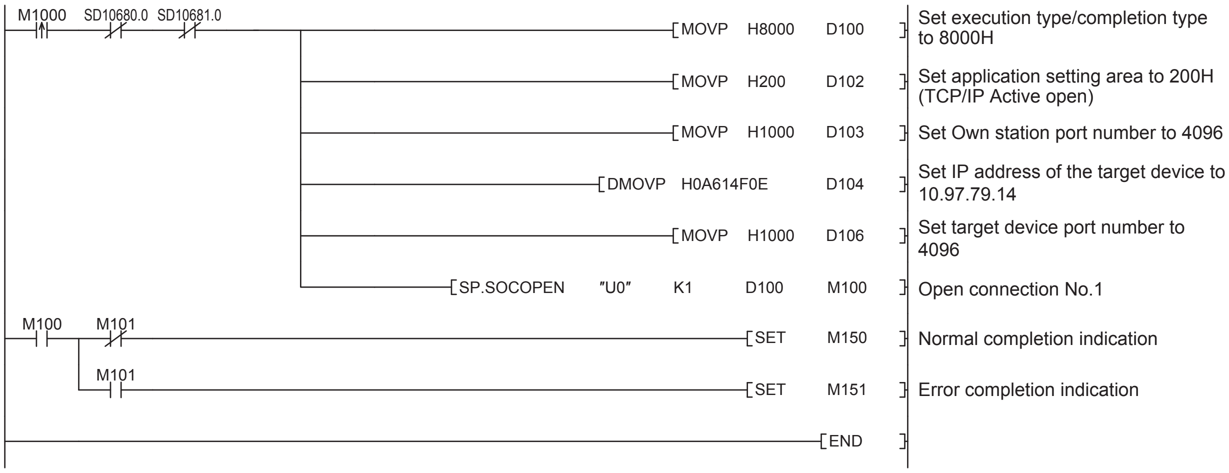
■Opening a connection using control data settings

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 normal completion device
M101	SP.SOCOPEN instruction error completion device
M150	Normal completion indication
M151	Error completion indication
M1000	SP.SOCOPEN instruction drive flag

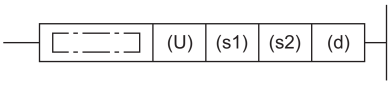
- Program



GP.OPEN

Establishes a connection. (Ethernet module dedicated instruction)

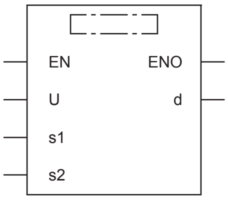
Ladder diagram



Structured text

ENO: =GP_OPEN (EN, U, s1, s2, d);

FBD/LD



("GP_OPEN" goes into □.)

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 number	1 to 32	16-bit unsigned binary	ANY16
(s2)	Head device number for storing the control data	Refer to Control data (☞ Page 148)	Word	ANY16_ARRAY (Number of elements: 10)
(d)	Head device number which turns ON when the execution of the instruction is completed and remains on for 1 scan. (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) For the error codes, refer to ☞ Page 816 ERROR CODES.	—	System
(s2) + 2	Application setting area	b15 b14 b13 to b9 b8 b7 to b0 (s2)+2 [2] | 0 | [1] | 0 [1] Communication method (protocol) 0: TCP/IP 1: UDP/IP [2] Open system 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 FFFEH)[*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 FFFEH)	
(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 FFFEH) 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

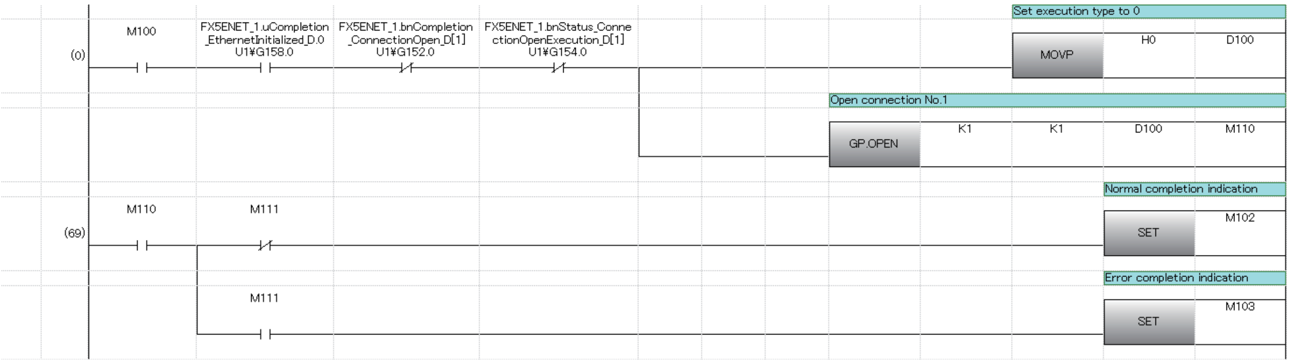
■Opening a connection using parameter settings

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



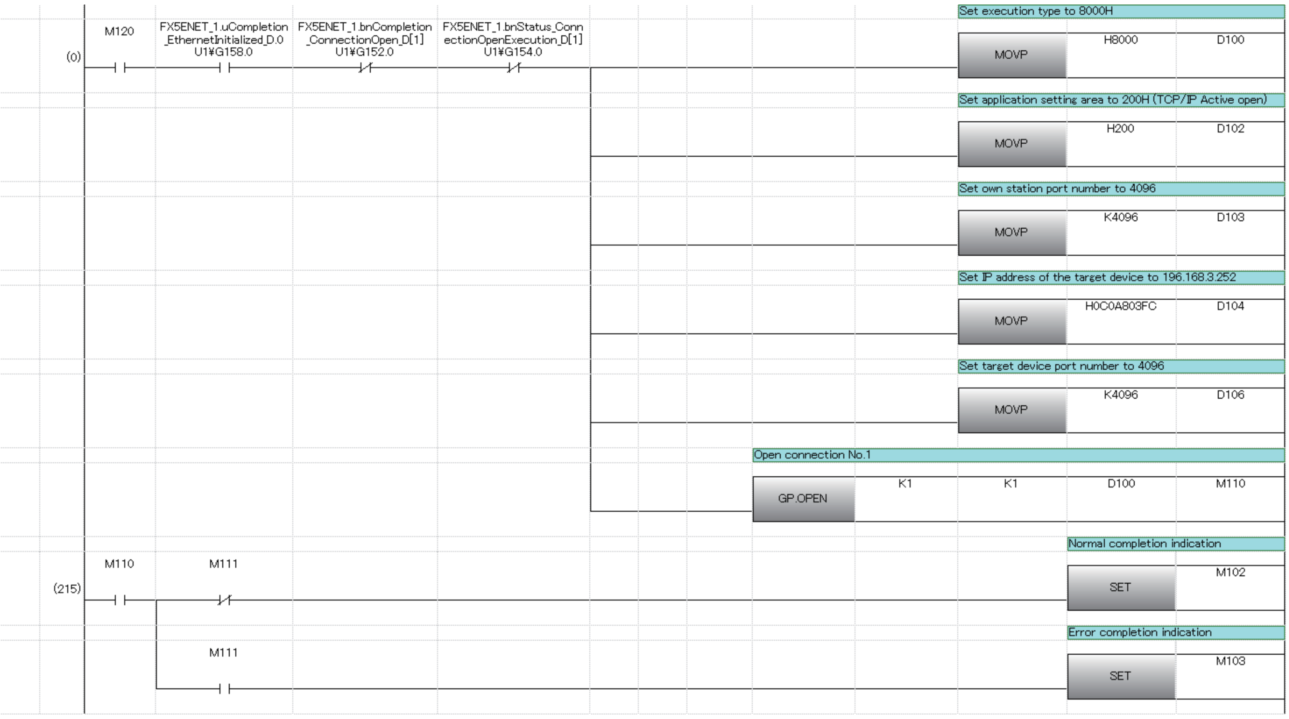
■Opening a connection using control data settings

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

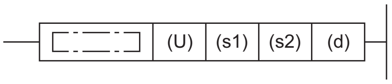


Disconnecting a connection

SP.SOCCLOSE

Closes a connection. (CPU module dedicated instruction)

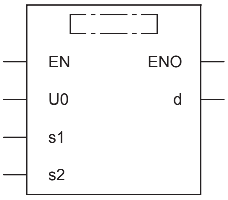
Ladder diagram



Structured text

ENO: =SP_SOCCLOSE (EN, U0, s1, s2, d);

FBD/LD



("SP_SOCCLOSE" goes into □.)

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)	Head device number for storing the control data	Refer to Control data (☞ Page 152)	Word	ANY16_ARRAY (Number of elements: 2)
(d)	Head device number which turns ON when the execution of the instruction is completed and remains on for 1 scan. (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) For the error codes, refer to ☞ Page 816 ERROR CODES.	—	System

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

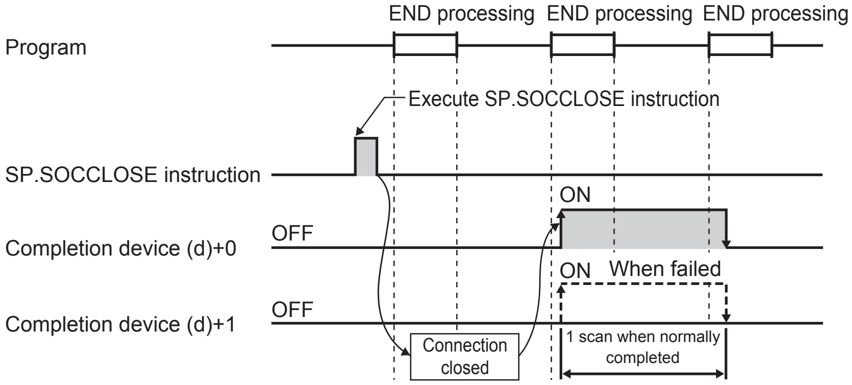
Processing details

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.

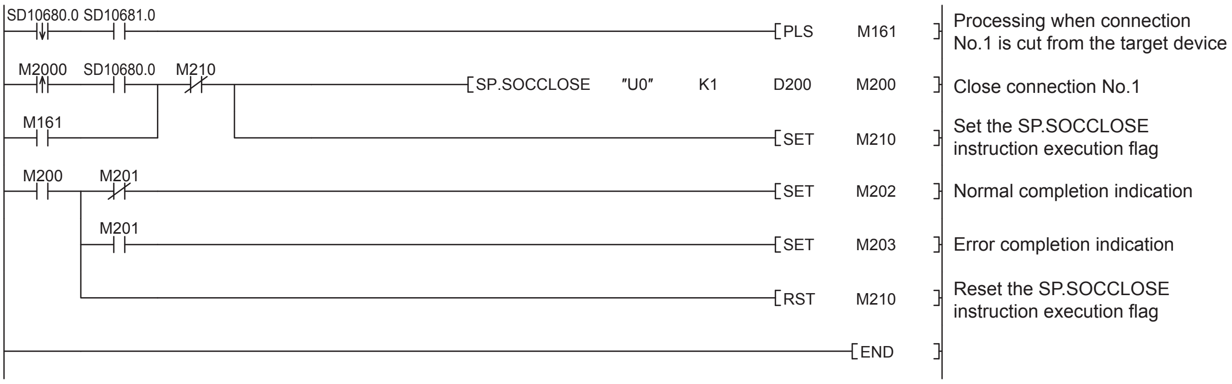
Program example

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

- Devices used

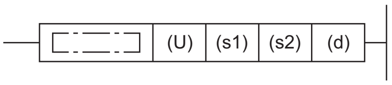
Device No.	Application
SD10680	Open completion signal
SD10681	Open request signal
D200	SP.SOCCLOSE instruction control data
M161	Connection No.1 cutting flag
M200	SP.SOCCLOSE instruction normal completion device
M201	SP.SOCCLOSE instruction error completion device
M202	Normal completion indication
M203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

- Program

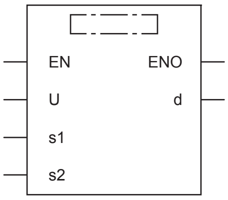


GP.CLOSE

Closes a connection. (Ethernet module dedicated instruction)

Ladder diagram	Structured text
	ENO: =GP_CLOSE (EN, U, s1, s2, d);

FBD/LD



("GP_CLOSE" goes into □.)

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 number	1 to 32	16-bit unsigned binary	ANY16
(s2)	Head device number for storing the control data	Refer to Control data (☞ Page 155)	Word	ANY16_ARRAY (Number of elements: 2)
(d)	Head device number which turns ON when the execution of the instruction is completed and remains on for 1 scan. (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
	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)
(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) For the error codes, refer to ☞ Page 816 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.

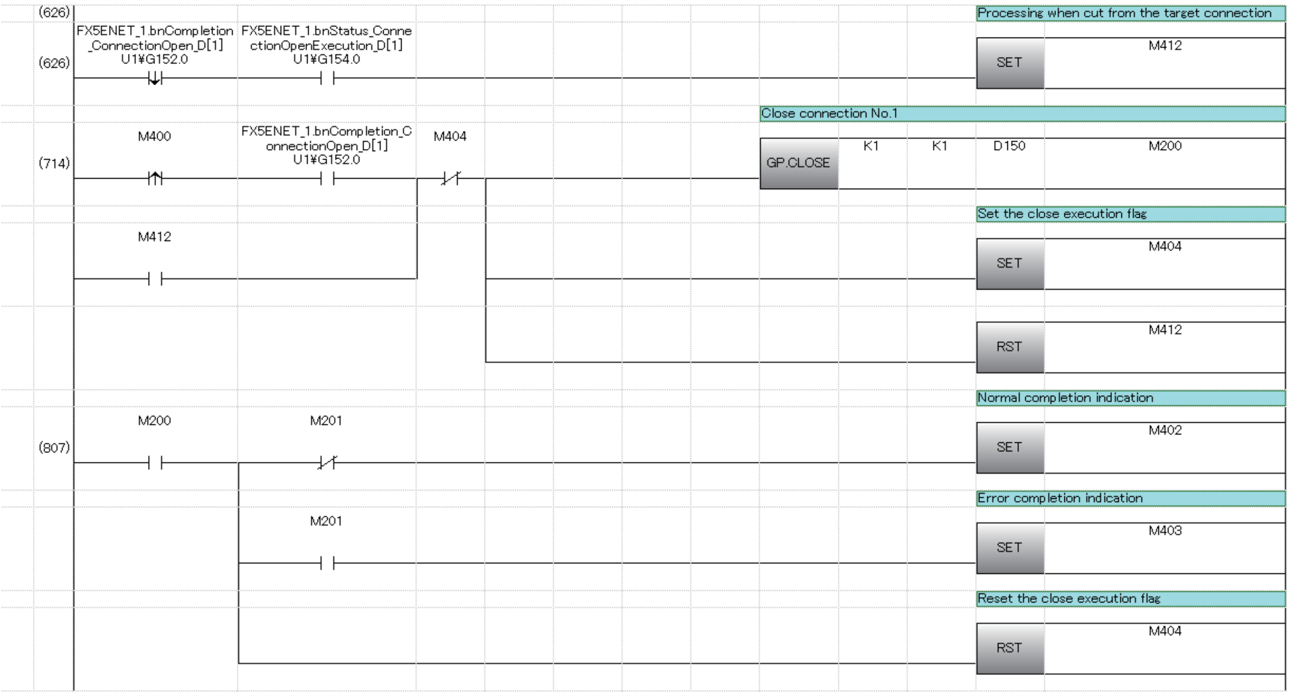
Program example

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

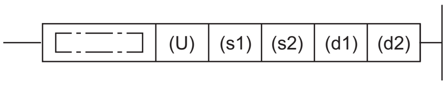
- Program



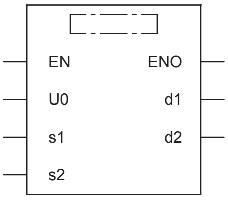
Reading received data in the END processing

SP.SOCRCV

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

Ladder diagram	Structured text
	ENO: =SP_SOCRCV (EN, U0, s1, s2, d1, d2);

FBD/LD



("SP_SOCRCV" goes into □.)

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 where control data is specified	Refer to Control data (☞ Page 159)	Word	ANY16_ARRAY (Number of elements: 2)
(d1)	Start number of the device in which received data is stored	—	Word	ANY16
(d2)	Head device number which turns ON when the execution of the instruction is completed and remains on for 1 scan. (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) For the error codes, refer to ☞ Page 816 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

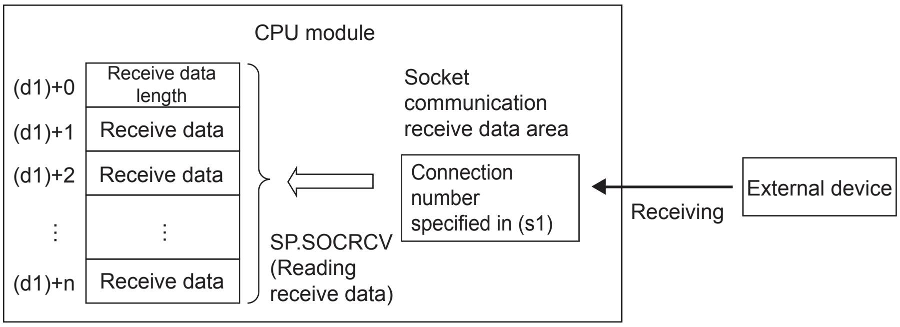
*1 System: The CPU module stores the execution result of the instruction.

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.

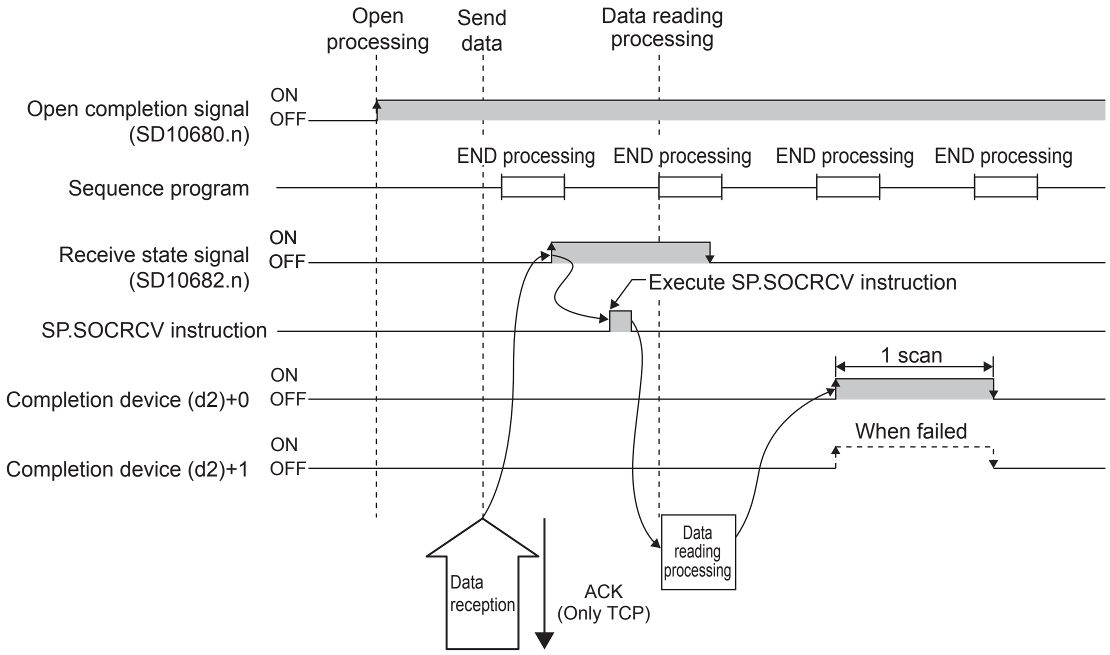


The result of the SP.SOCRCV instruction can be checked with the completion device (d2)+0 and (d2)+1.

- 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.

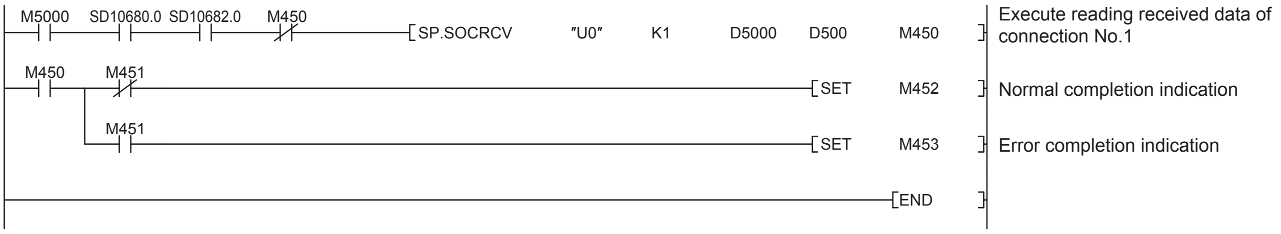
Program example

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 or later	Received data length and received data storage location
M450	SP.SOCRCV instruction normal 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

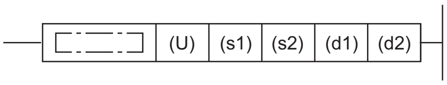
Consecutively sent data can be consecutively read by connecting the completion device of the SP.SOCRCV instruction to the execution command as a normally closed contact.

Reading received data in the processing

GP.SOCRCV

Reads the data received. (Ethernet module dedicated instruction)

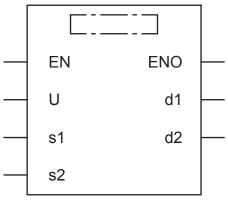
Ladder diagram



Structured text

ENO: =GP_SOCRCV (EN, U, s1, s2, d1, d2);

FBD/LD



("GP_SOCRCV" goes into □.)

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 number	1 to 32	16-bit unsigned binary	ANY16
(s2)	Start number of the device where control data is specified	Refer to Control data (☞ Page 163)	Word	ANY16_ARRAY (Number of elements: 2)
(d1)	Start number of the device in which received data is stored	—	Word	ANY16*1
(d2)	Head device number which turns ON when the execution of the instruction is completed and remains on for 1 scan. (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) For the error codes, refer to ☞ Page 816 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.

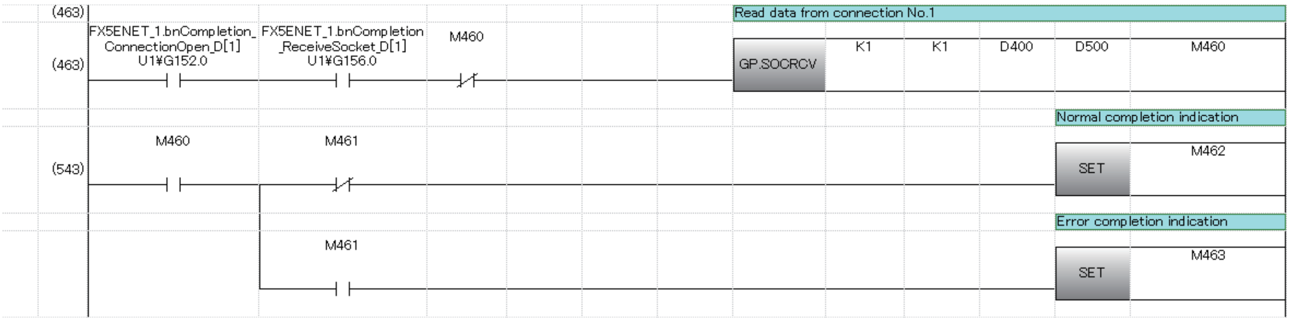
Program example

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



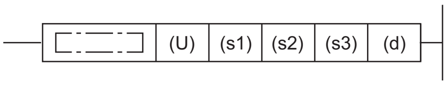
Point

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.

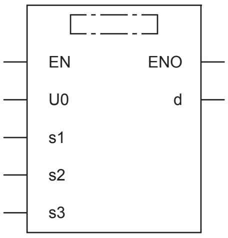
Sending data

SP.SOCSND

Sends data. (CPU module dedicated instruction)

Ladder diagram	Structured text
	ENO: =SP_SOCSND (EN, U0, s1, s2, s3, d);

FBD/LD



("SP_SOCSND" goes into □.)

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 where control data is specified	Refer to Control data (☞ Page 166)	Word	ANY16_ARRAY (Number of elements: 2)
(s3)	Start number of the device in which send data is stored	—	Word	ANY16
(d)	Head device number which turns ON when the execution of the instruction is completed and remains on for 1 scan. (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) For the error codes, refer to ☞ Page 816 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

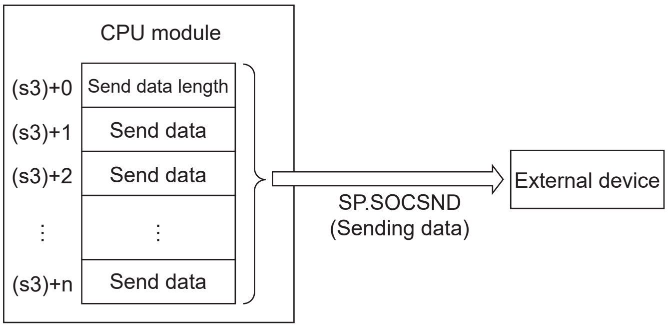
[1] User: Data to be set before the execution of the instruction. System: The CPU module stores the execution result of the instruction.

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).



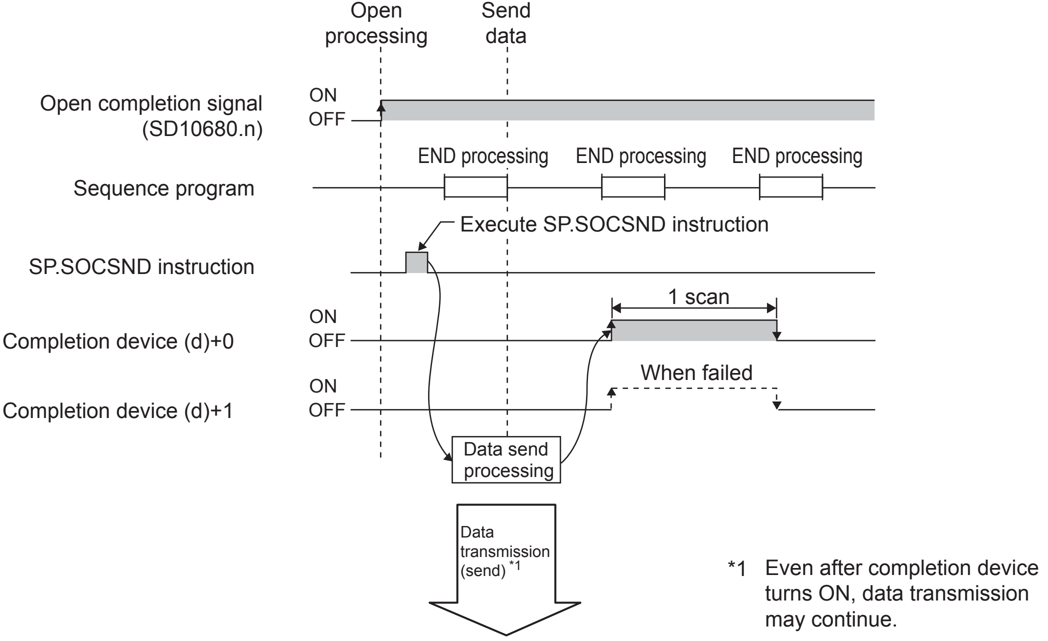
The result of the SP.SOCSND instruction can be checked with the completion device, (d)+0 and (d)+1.

- 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.

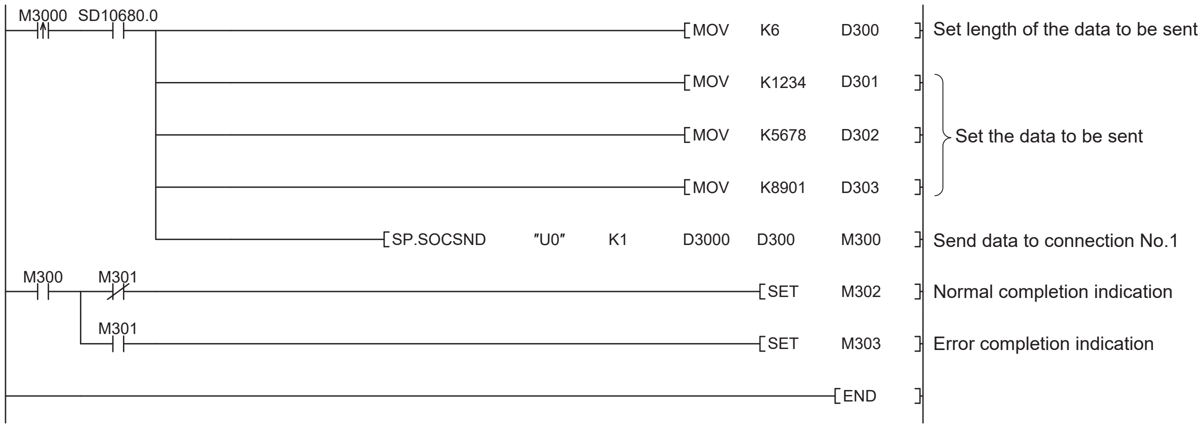
Program example

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

- Devices used

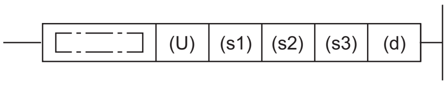
Device No.	Application
SD10680	Open completion signal
D3000	SP.SOCSND instruction control data
D300 to D303	Send data length and send data storage location
M300	SP.SOCSND instruction normal completion device
M301	SP.SOCSND instruction error completion device
M302	Normal completion indication
M303	Error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

- Program

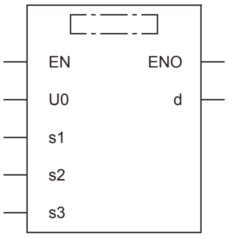

M3000
SD10680.0
MOV K6 D300
Set length of the data to be sent
MOV K1234 D301
MOV K5678 D302
MOV K8901 D303
Set the data to be sent
SP.SOCSND "U0" K1 D3000 D300 M300
Send data to connection No.1
M300
M301
SET M302
Normal completion indication
M301
SET M303
Error completion indication
END

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" goes into □.)

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 number	1 to 32	16-bit unsigned binary	ANY16
(s2)	Start number of the device where control data is specified	Refer to Control data (☞ Page 170)	Word	ANY16_ARRAY (Number of elements: 2)
(s3)	Start number of the device in which send data is stored	—	Word	ANY16[1]
(d)	Head device number which turns ON when the execution of the instruction is completed and remains on for 1 scan. (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) For the error codes, refer to ☞ Page 816 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.

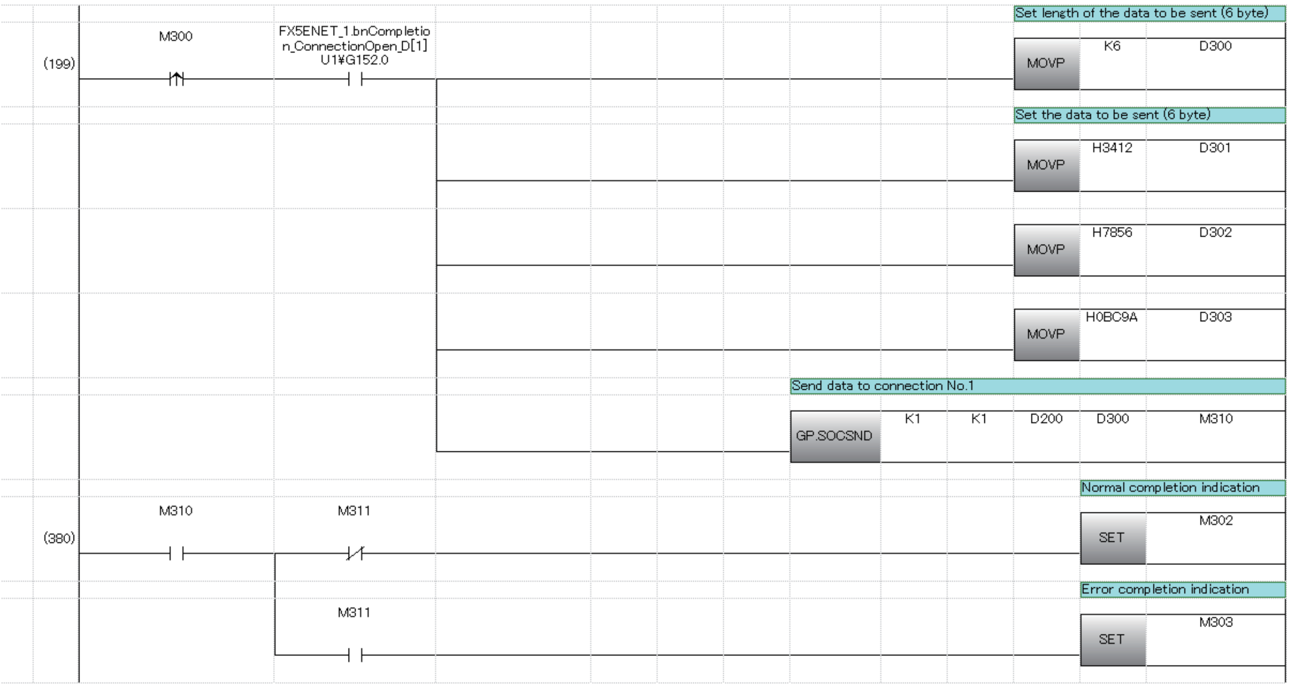
Program example

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

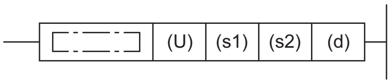
- Program



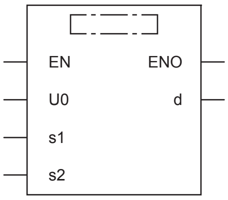
Reading connection information

SP.SOCCINF

Reads connection information. (CPU module dedicated instruction)

Ladder diagram	Structured text
	ENO: =SP_SOCCINF (EN, U0, s1, s2, d);

FBD/LD



("SP_SOCCINF" goes into □.)

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)	Head device number for storing the control data	Refer to Control data (☞ Page 173)	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

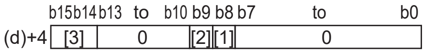
*1 In the case of the ST language and the FBD/LD language, U displays as U0.

*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) For the error codes, refer to ☞ Page 816 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 FFFEH)[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 FFFEH)[2][3]	
(d) + 4	Application setting area	 [1] Communication method (protocol) 0: TCP/IP 1: UDP/IP [2] Socket communications function procedure 1: Non-protocol method [3] Open system 00: Active open or UDP/IP 10: Unpassive open 11: Fullpassive open	Shown on left side[2]	

[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] Of the host station port numbers, 1 to 1023 (0001H to 03FFH) are generally reserved port numbers and 61440 to 65534 (F000H to FFFEH) 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 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

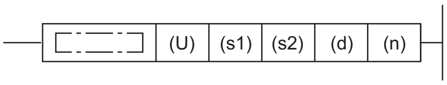
- Program

Execute reading connection information of connection No.1

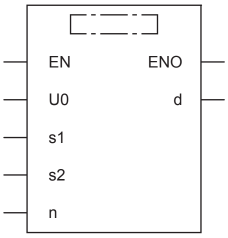
Reading socket communication receive data

S(P).SOCRDATA

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

Ladder diagram	Structured text
	ENO: =S_SOCRDATA (EN, U0, s1, s2, n, d); ENO: =SP_SOCRDATA (EN, U0, s1, s2, n, d);

FBD/LD



("S_SOCRDATA", "SP_SOCRDATA" goes into □.)

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)	Head device number for storing the control data	Refer to Control data (☞ Page 174)	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 unsigned 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) For the error codes, refer to ☞ Page 816 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.

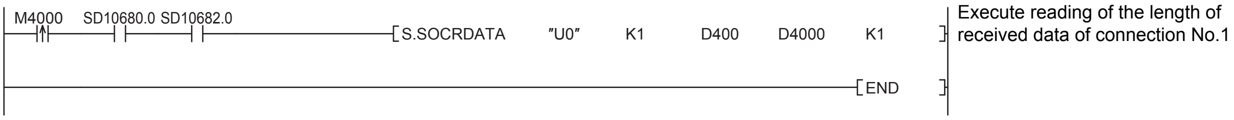
Program example

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



8.4 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 in the processing

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.

Received data is cleared at the status change from RUN to STOP.

Do not execute the CLOSE instruction until reading of all data received from the target device is complete.

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

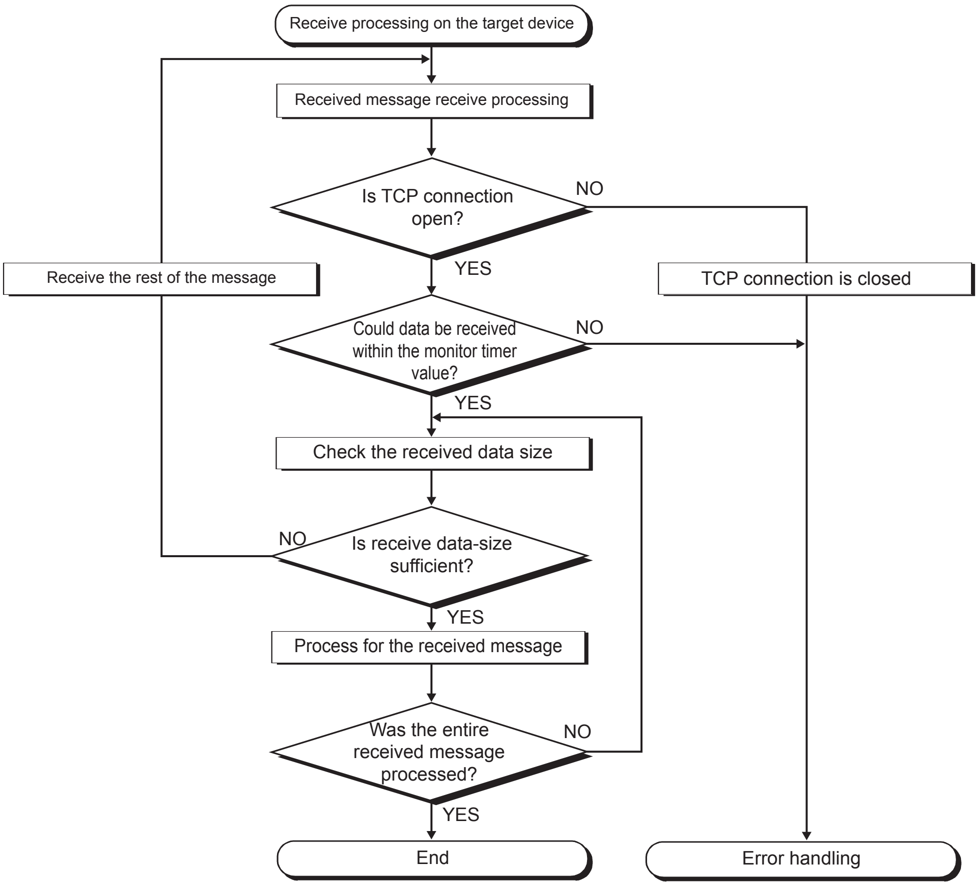
Allow 500ms or more before reestablishing a connection of the same target IP address, the same host station port number, and the same target port number in the TCP communication after closing it.

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

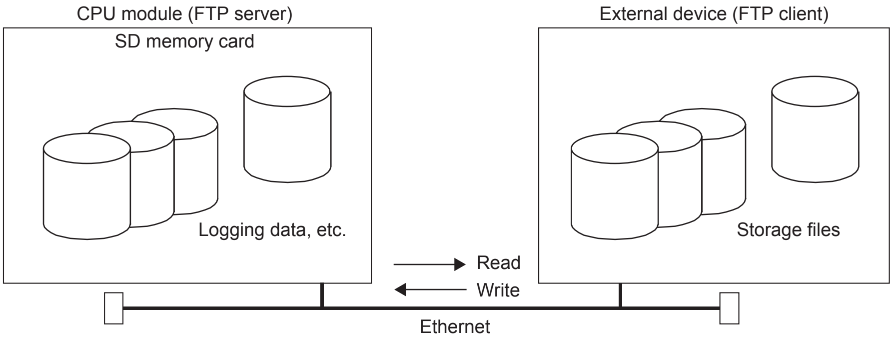
Since no delimiter is provided for TCP communication data, separate data blocks that are sent continuously may be combined, or data sent all at once may be segmented, on the receiving end. The receive data length must be confirmed on the receiving end as necessary.

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



9 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

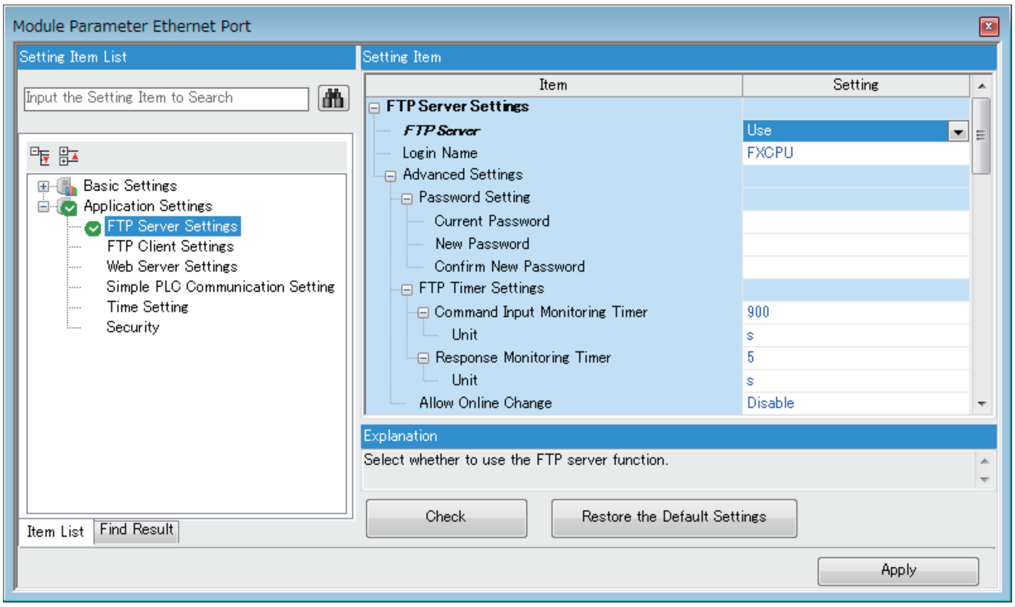
9.1 Data Communication Procedures

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] ⇨ Module model name ⇨ [Module Parameter] ⇨ [Ethernet Port] ⇨ [Application Settings] ⇨ [FTP Server Settings]



Item	Setting
FTP Server Settings	
FTP Server	Use
Login Name	FXCPU
Advanced Settings	
Password Setting	
Current Password	
New Password	
Confirm New Password	
FTP Timer Settings	
Command Input Monitoring Timer	900
Unit	s
Response Monitoring Timer	5
Unit	s
Allow Online Change	Disable

Explanation

Select whether to use the FTP server function.

Item	Description	Setting range
FTP Server	Select whether to use the file transfer function (FTP server) of the CPU module.	• 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 179 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 179 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.	• Disable • Enable (Default: Disable)

■Password Setting

- Current password

Enter the current password for login to the CPU module.

Default password (initial setting) is "FXCPU".

Point

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 (900s) 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.

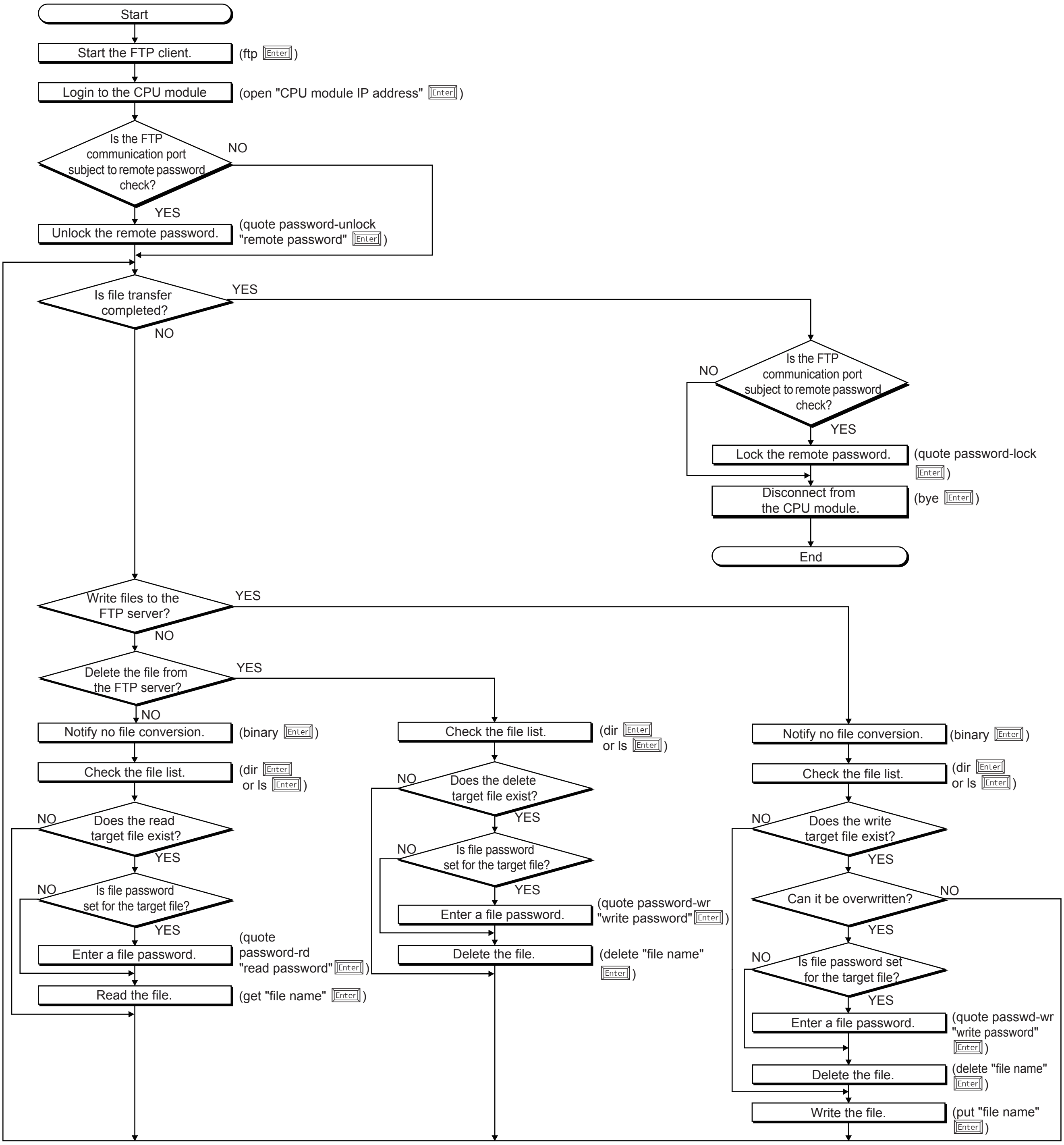
Set a value within the following range.

Unit	Setting range
s	1 to 16383
ms[*2]	100 to 16383000

[*2] 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.)

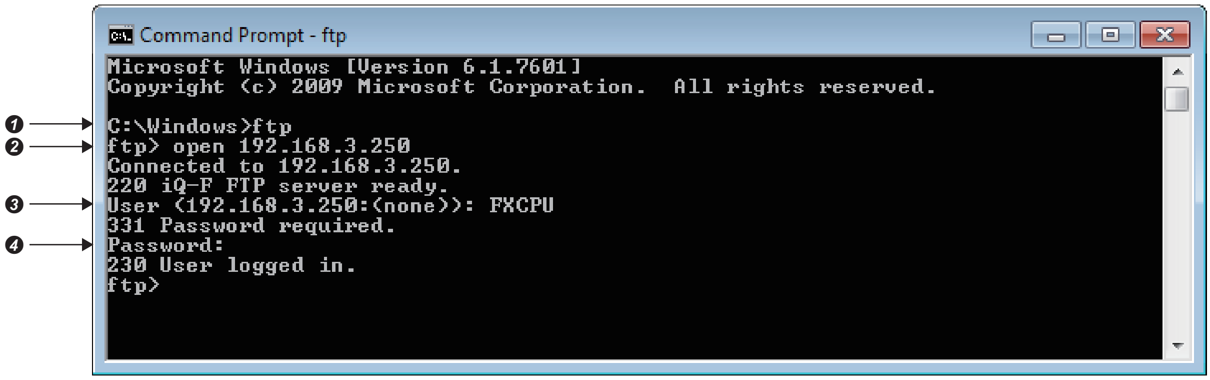


Logging into CPU module

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

Ex.

Start FTP from the Microsoft® Windows® command prompt.



```
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>
```

❶ FTP start (ftp Enter)
❷ Connect with FTP server (open CPU module IP address Enter)
❸ Specify login name (login name Enter)
❹ Specify password (password Enter)

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 Enter

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

- quote password-lock Enter

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 182 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 Enter)
- Read password (quote passwd-rd read password Enter)

9.2 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)

9.3 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 182 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 235 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 183 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 183 Drive name (drive No.),Page 183 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_00000001.BIN
	Example 2	When reading files stored in the SD memory card and store with different file name get 2:\LOG01_00000001.BIN LOG\LOG01_01.B
	Caution	• 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 directory 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 SD memory card into 20160101.LOG file mdir 2:\20160101.LOG
	Caution	• 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 directory 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	• 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	• 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 directory 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	• 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] • 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] • 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	• 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. • When specifying the 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. • 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	• 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. • 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. • When a write 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 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	• 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	Inputs the user name and password for the connected FTP server.
	Specification format	user "user name" [FTP password] • 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

9.4 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

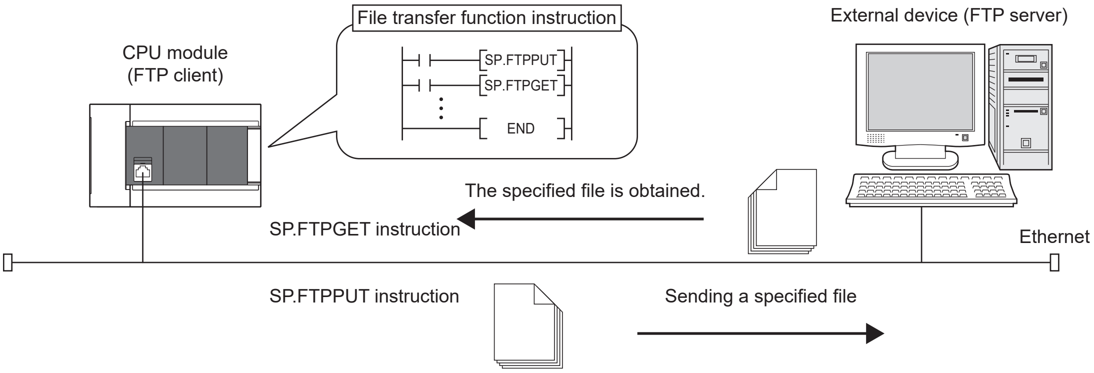
When returning to the default parameters, all items set in "Application Settings" must be reset in addition to the "FTP Server Settings".

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.

10 FILE TRANSFER FUNCTION (FTP CLIENT)

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

The SD memory card module is required to use the FX5S CPU module.

10.1 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.

10.2 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 191 Setting in the external device (FTP server))
2. Set the FTP client. (☞ Page 192 Setting in the CPU module (FTP client))
3. Transfer a file. (☞ Page 192 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 39 Setting module parameters)

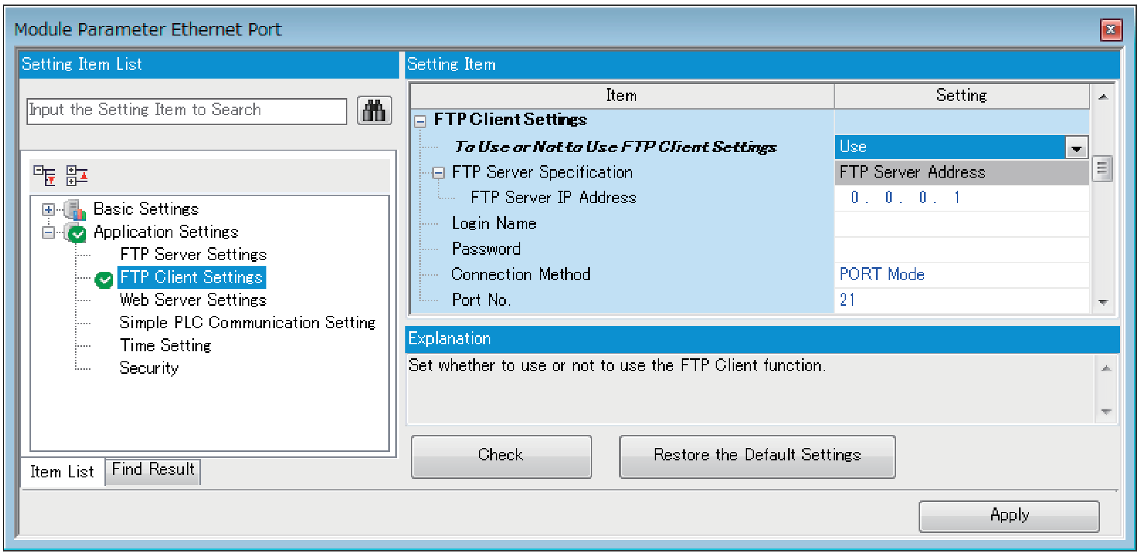
2. Set the file transfer function (the FTP client). (☞ Page 192 FTP Client Setting)

Point

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

FTP Client Setting

Navigation window ⇨ [Parameter] ⇨ Module model name ⇨ [Module Parameter] ⇨ [Ethernet Port] ⇨ [Application Settings] ⇨ [FTP Client Settings]



Item	Setting
FTP Client Settings	
To Use or Not to Use FTP Client Settings	Use
FTP Server Specification	FTP Server Address
FTP Server IP Address	0 . 0 . 0 . 1
Login Name	
Password	
Connection Method	PORT Mode
Port No.	21

Explanation

Set whether to use or not to use the FTP Client function.

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.	• 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.	• 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).

10.3 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] ⇨ Module name ⇨ [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] ⇨ Module model name ⇨ [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: 500ms
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: Character string
· Number of digits/characters: 4
· Add date type: Date to establish file switching condition
· Format: YYYYMMDDhh

*2 The settings are as follows.
· Device: SD213 (Clock Data (Hour))
· Conditional formula: During changes
· Data type: Word [signed]

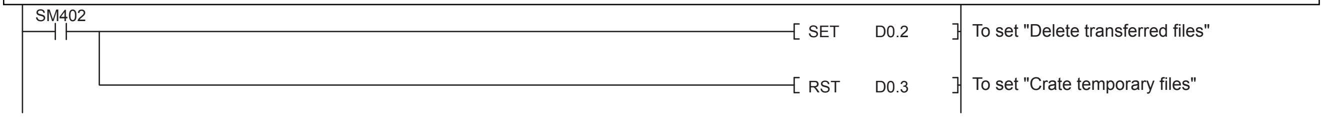
Program example

■Devices used

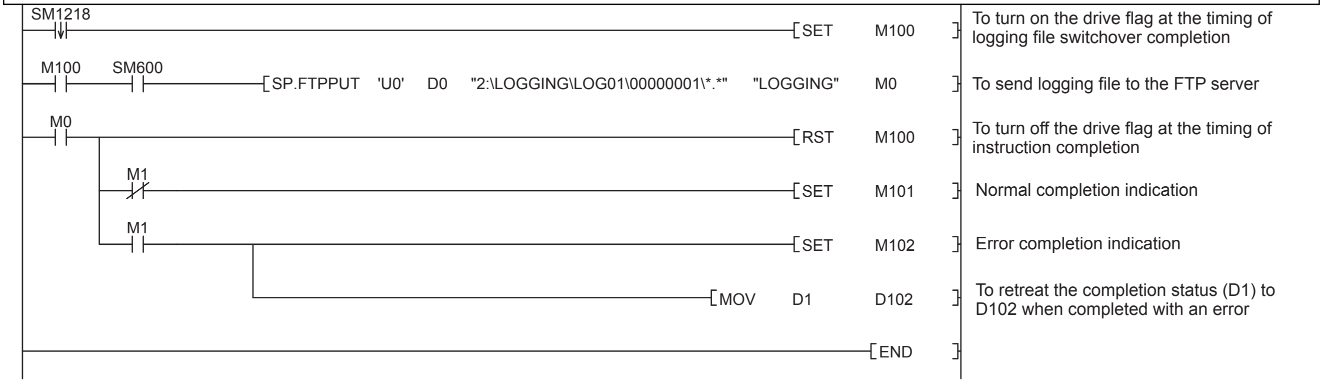
Device No.	Application
SM402	After RUN, ON for one scan only
SM600	Memory card usable
SM1218	Data logging setting No.1 Logging data storage file switching in progress
D0	Application setting area (SP.FTPPUT instruction control data) • D0.2: Transfer completion file delete setting • D0.3: Temporary file 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] ⇨ Module name ⇨ [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] ⇨ Module model name ⇨ [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: 500ms
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		Simple setting[*1]
	File switching setting	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

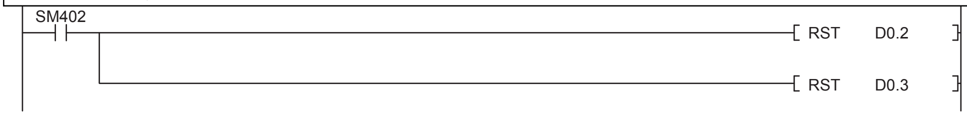
Device No.	Application
SD1210	Data logging setting No.1 Latest storage file number [Low-order]
SD1211	Data logging setting No.1 Latest storage file number [High-order]
SM402	After RUN, ON for one scan only
SM600	Memory card usable
SM1218	Data logging setting No.1 Logging data storage file switching in progress
D0	Application setting area (SP.FTPPUT instruction control data) • 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 code 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

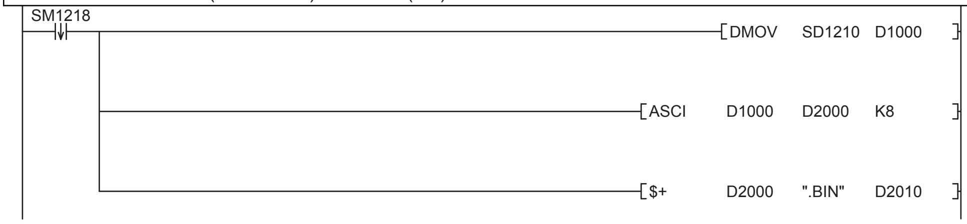


To set "Do not delete transferred files"

To set "Create temporary files"

[Create transfer file name]

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



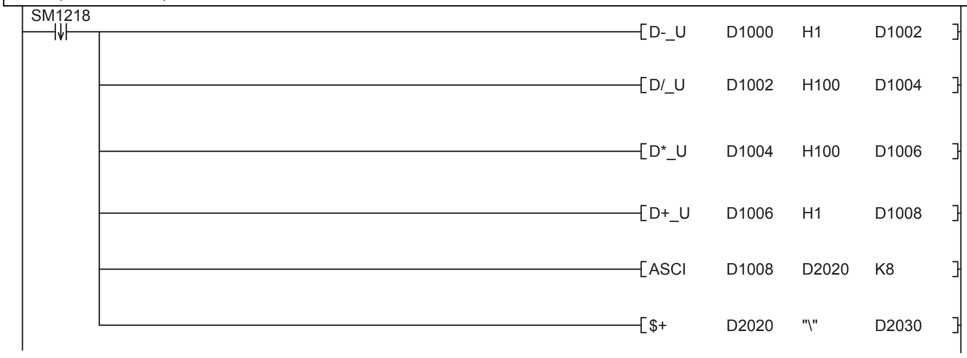
Store the data logging setting No.1 latest storage file number (SD1210 and SD1211) in D1000 and D1001

To transform the most recently stored file number into a character string as a hexadecimal number and output the result to areas stating from D2000

To append the file extension (.BIN) to those stored in areas starting from D2000 and output the result to D2010

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:

❶(Most recently stored file number - 1) / H100
❷(Result of ❶) × H100 + 1



Most recently stored file number - 1

To divide the most recently stored file number - 1 by H100 to calculate the first six digits of the folder number

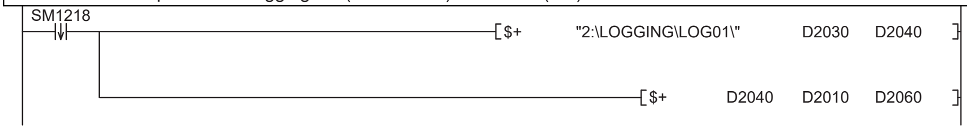
To multiply the calculation result above by H100 and carry the result to the first six digits of the folder number

Store + 1 = folder number in D1008, D1009

To transform the created file number (D1008 and D1009) into a character string as a hexadecimal number and output the result to areas starting from D2020

To append "\" to the folder name

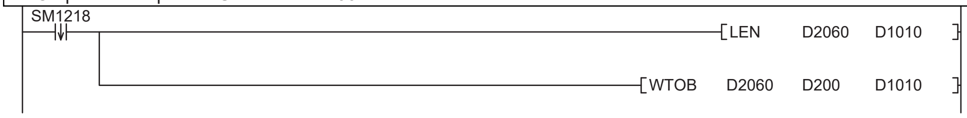
Create the full path of the logging file (ASCII code) to D2060. (Ex.) "2:\LOGGING\LOG01\00000001\00000001.BIN"



To append the upper folder (2:\LOGGING\LOG01\) and the corresponding logging file (D2030)

To append the file name (stored in areas starting from D2010) to those stored in areas starting from D2040 and output the full path to D2060

Output the full path in Unicode to D200.



Store the character string length of the ASCII code full path (areas starting from D2060) in D1010

To separate the ASCII code full path (areas starting from D2060) into byte-by-byte parts of the character string length (D1010) and store the result in areas starting from 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.

To turn on the drive flag at the timing of logging file switchover completion

To send logging file to the FTP server

To turn off the drive flag at the timing of instruction completion

Normal completion indication

Error completion indication

To retreat the completion status (D1) to D102 when completed with an error

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] ⇨ Module name ⇨ [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] ⇨ Module model name ⇨ [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

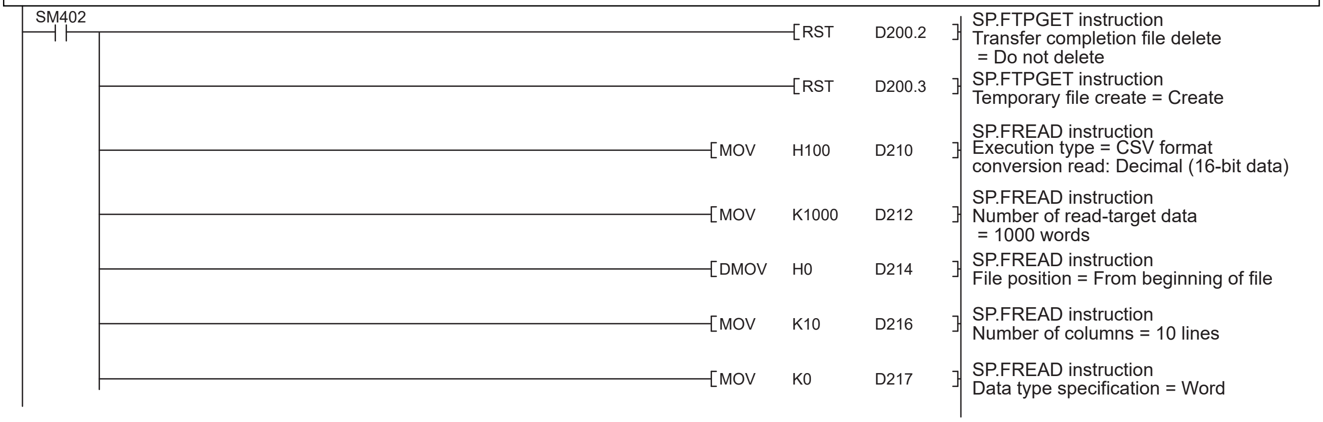
Program example

■Devices used

Device No.	Application
SM402	After RUN, ON for one scan only
SM600	Memory card usable
■Devices used for SP.FTPGET instruction	
M300	SP.FTPGET instruction execution command
D200	Application setting area (SP.FTPGET instruction control data) • 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) • 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) • 1000 words
D213	Maximum number of read data (SP.FREAD instruction control data) • 0 (fixed)
D214, D215	File position (SP.FREAD instruction control data) • From beginning of file
D216	Number of columns (SP.FREAD instruction control data) • 10 lines
D217	Data type specification (SP.FREAD instruction control data) • 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

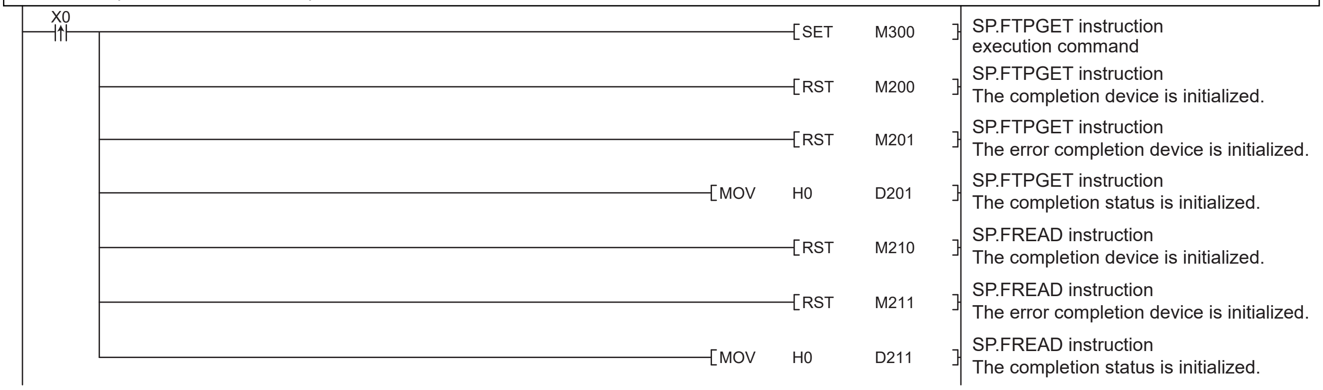
■Program

"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
  +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

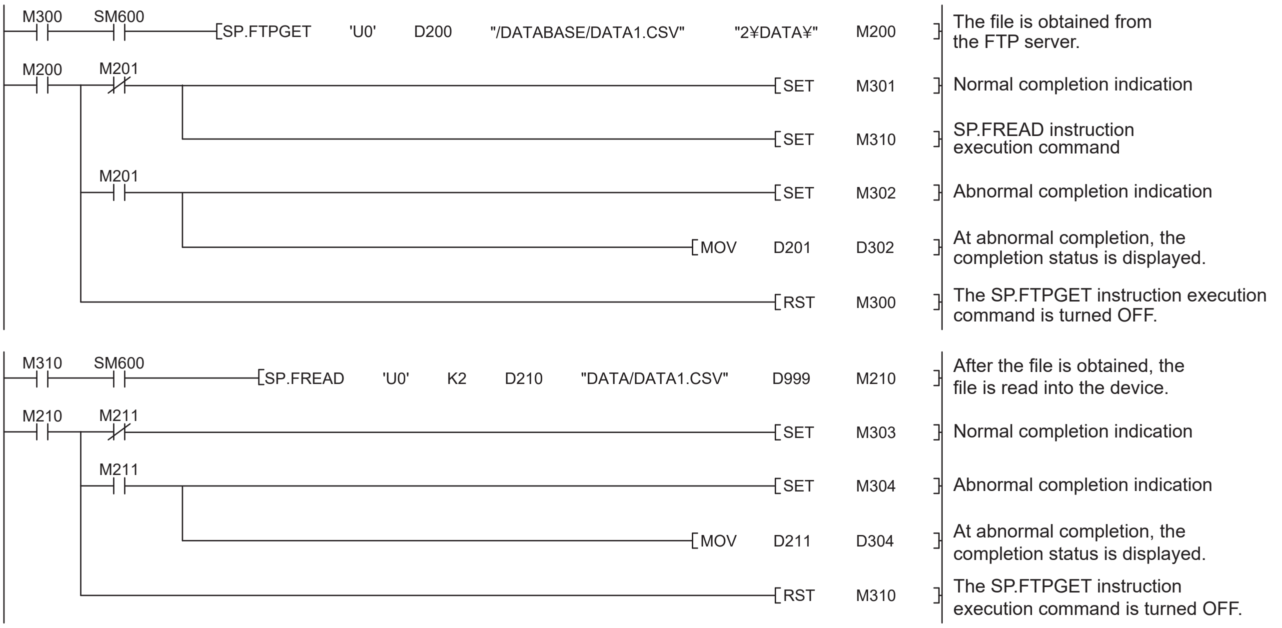


```
SM402
-| |-+-[RST  D200.2 ]   SP.FTPGET instruction
      |                  Transfer completion file delete
      |                   = Do not delete
      +-[RST  D200.3 ]   SP.FTPGET instruction
      |                  Temporary file create = Create
      +-[MOV  H100  D210 ]  SP.FREAD instruction
      |                  Execution type = CSV format
      |                  conversion read: Decimal (16-bit data)
      +-[MOV  K1000  D212 ] SP.FREAD instruction
      |                  Number of read-target data
      |                   = 1000 words
      +-[DMOV  H0  D214 ]   SP.FREAD instruction
      |                  File position = From beginning of file
      +-[MOV  K10  D216 ]   SP.FREAD instruction
      |                  Number of columns = 10 lines
      +-[MOV  K0  D217 ]    SP.FREAD instruction
                         Data type specification = Word
```

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



```
X0
-|↑|-+-[SET  M300 ]      SP.FTPGET instruction
      |                  execution command
      +-[RST  M200 ]      SP.FTPGET instruction
      |                  The completion device is initialized.
      +-[RST  M201 ]      SP.FTPGET instruction
      |                  The error completion device is initialized.
      +-[MOV  H0  D201 ]  SP.FTPGET instruction
      |                  The completion status is initialized.
      +-[RST  M210 ]      SP.FREAD instruction
      |                  The completion device is initialized.
      +-[RST  M211 ]      SP.FREAD instruction
      |                  The error completion device is initialized.
      +-[MOV  H0  D211 ]  SP.FREAD instruction
                         The completion status is initialized.
```



```
M300  SM600
-| |---| |---[SP.FTPGET  'U0'  D200  "/DATABASE/DATA1.CSV"  "2¥DATA¥"  M200 ]
                                         The file is obtained from
                                         the FTP server.
M200  M201
-| |-+--|/|--+-[SET  M301 ]             Normal completion indication
      |       +-[SET  M310 ]             SP.FREAD instruction
      |                                  execution command
      |  M201
      +--| |--+-[SET  M302 ]             Abnormal completion indication
      |       +-[MOV  D201  D302 ]       At abnormal completion, the
      |                                  completion status is displayed.
      +-[RST  M300 ]                     The SP.FTPGET instruction execution
                                         command is turned OFF.

M310  SM600
-| |---| |---[SP.FREAD  'U0'  K2  D210  "DATA/DATA1.CSV"  D999  M210 ]
                                         After the file is obtained, the
                                         file is read into the device.
M210  M211
-| |-+--|/|----[SET  M303 ]             Normal completion indication
      |  M211
      +--| |--+-[SET  M304 ]             Abnormal completion indication
      |       +-[MOV  D211  D304 ]       At abnormal completion, the
      |                                  completion status is displayed.
      +-[RST  M310 ]                     The SP.FTPGET instruction
                                         execution command is turned OFF.
```

10.4 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 FFFEH). Do not use the port numbers of 62000 to 65534 (F230H to FFFEH) 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.

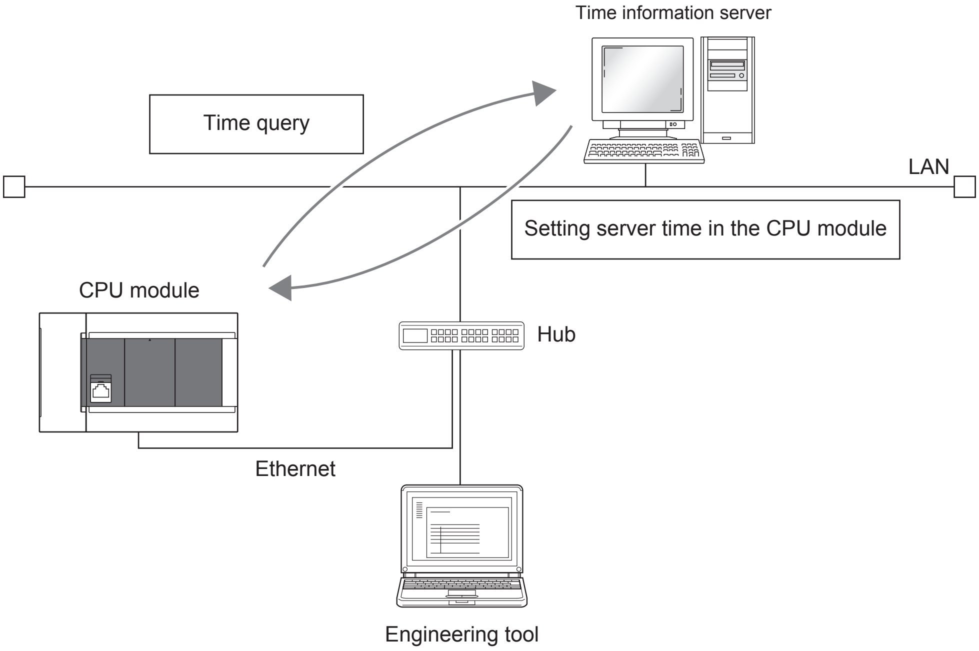
(Example) Scan time: 5ms, file size: 16MB file transfer time (SP.FTPPUT instruction): Approx. 548s

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.

11 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.

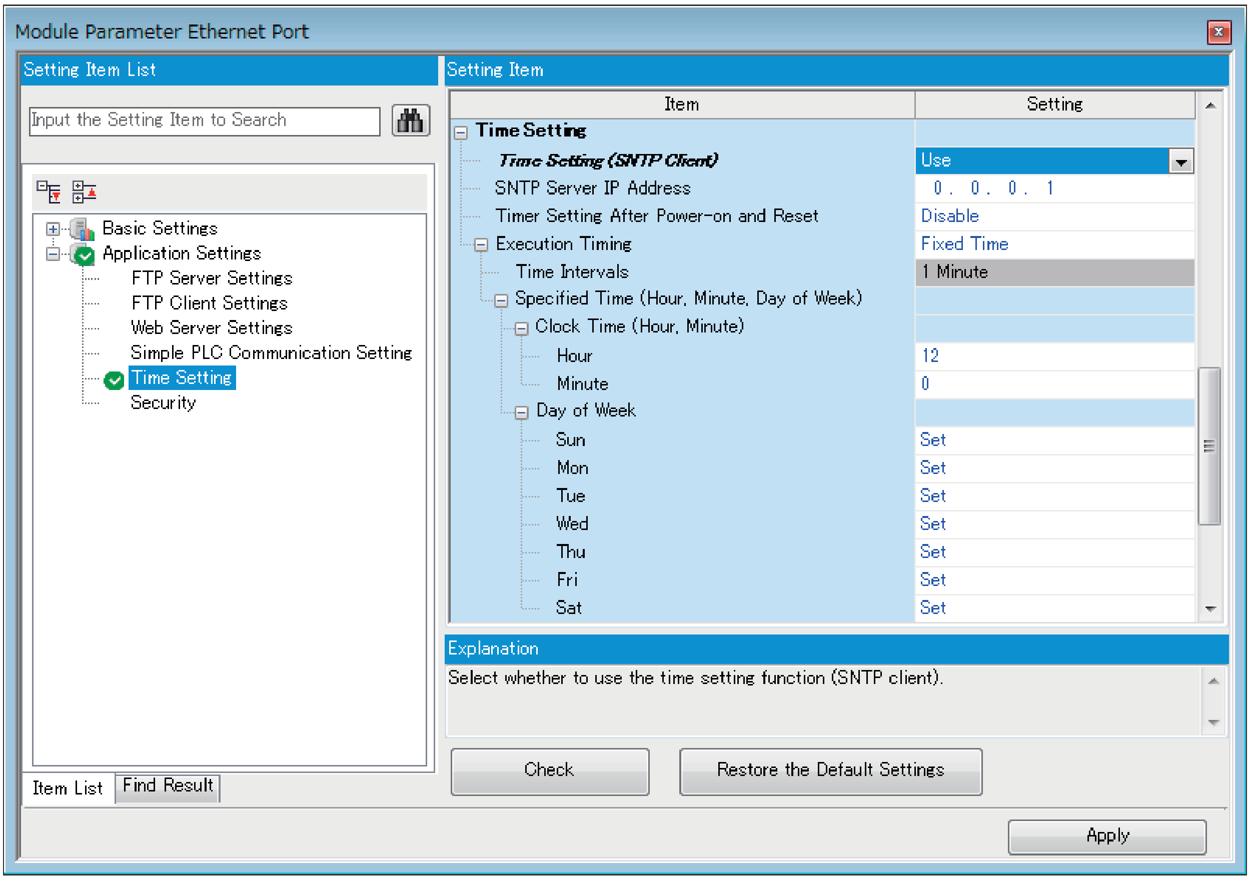
Point

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] ⇨ Module model name ⇨ [Module Parameter] ⇨ [Ethernet Port] ⇨ [Application Settings] ⇨ [Time Setting]



Item	Setting
Time Setting	
Time Setting (SNTP Client)	Use
SNTP Server IP Address	0 . 0 . 0 . 1
Timer Setting After Power-on and Reset	Disable
Execution Timing	Fixed Time
Time Intervals	1 Minute
Specified Time (Hour, Minute, Day of Week)	
Clock Time (Hour, Minute)	
Hour	12
Minute	0
Day of Week	
Sun	Set
Mon	Set
Tue	Set
Wed	Set
Thu	Set
Fri	Set
Sat	Set

Select whether to use the time setting function (SNTP client).

Item		Description	Setting range
Time Setting (SNTP Client)		Select whether to use the time setting function (SNTP client).	• 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.	• Disable • Enable (Default: Disable)
Execution Timing	—	Select the execution timing of the time setting.	• 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.	• 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 860 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) for 20 seconds after the time setting is executed, the communication times out. An error occurs when the communication times out and the event code is registered in the event history.

In addition, the communication times out when the following settings are configured by using the IP filter function. (☞ Page 232 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.

12 WEB SERVER FUNCTION

For details on 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).

12.1 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 209 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 211 Level display (WSLevel)
Figure display (WSFigure)	Displays the specified figure when the device value reaches the set value.	Page 213 Figure display (WSFigure)
Image display (WSPicture)	Displays the specified image file when the device value reaches the set value.	Page 215 Image display (WSPicture)
Historical graph (WSHstgrp)	Displays the device data in a time-series line graph.	Page 216 Historical graph display (WSHstgrp)
Write button (WSWrtBtn)	Writes the specified value to the specified device.	Page 218 Write button (WSWrtBtn)
Logout button (WSLogoutBtn)	Performs the logout operation.	Page 219 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.

○: Can be set, ×: Cannot be set

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	×

[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 1 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

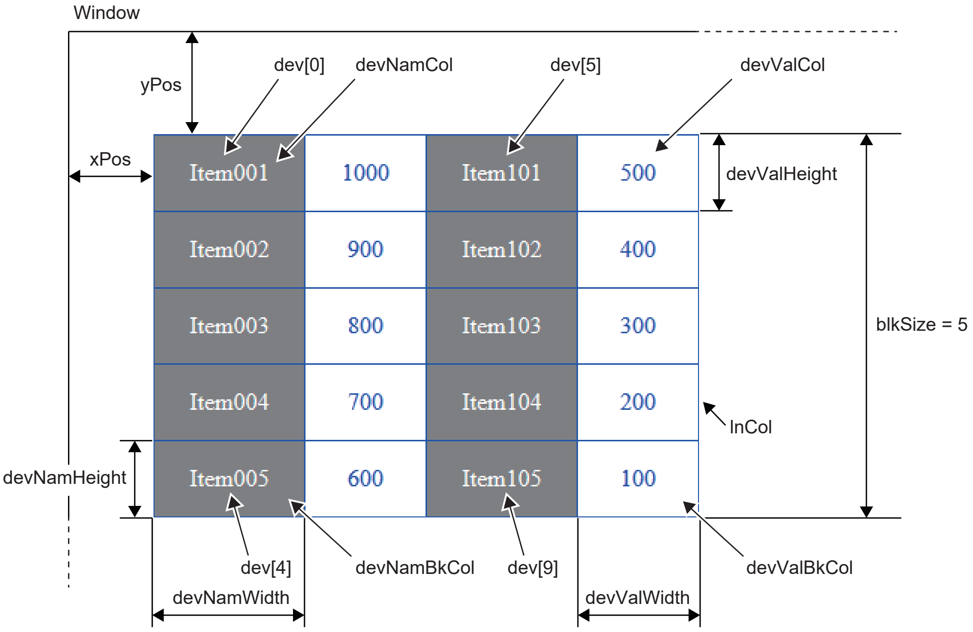
The following files are available in the user Web page.

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)

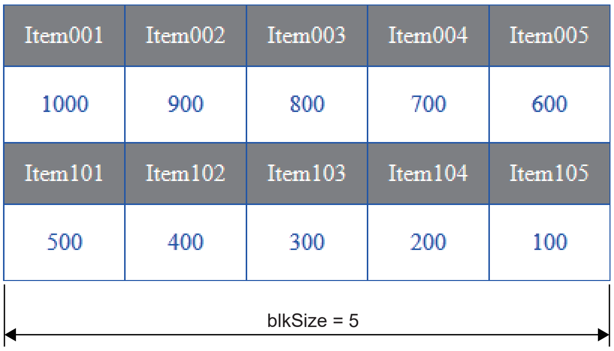
This object displays the specified device data in a table format. The current values of the devices are displayed as center aligned and updated in a fixed interval.

■direction = 0 (Vertical)



Item001	1000	Item101	500
Item002	900	Item102	400
Item003	800	Item103	300
Item004	700	Item104	200
Item005	600	Item105	100

■direction = 1 (Horizontal)



Item001	Item002	Item003	Item004	Item005
1000	900	800	700	600
Item101	Item102	Item103	Item104	Item105
500	400	300	200	100

Parameter

Set dev[n] according to the number of devices to be displayed. (n: 0 to 31)

○: Required, —: Optional

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: Do 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		
lnCol	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)	—

*1 When the device name is not displayed (devNamDisp = 0), a setting is not required.

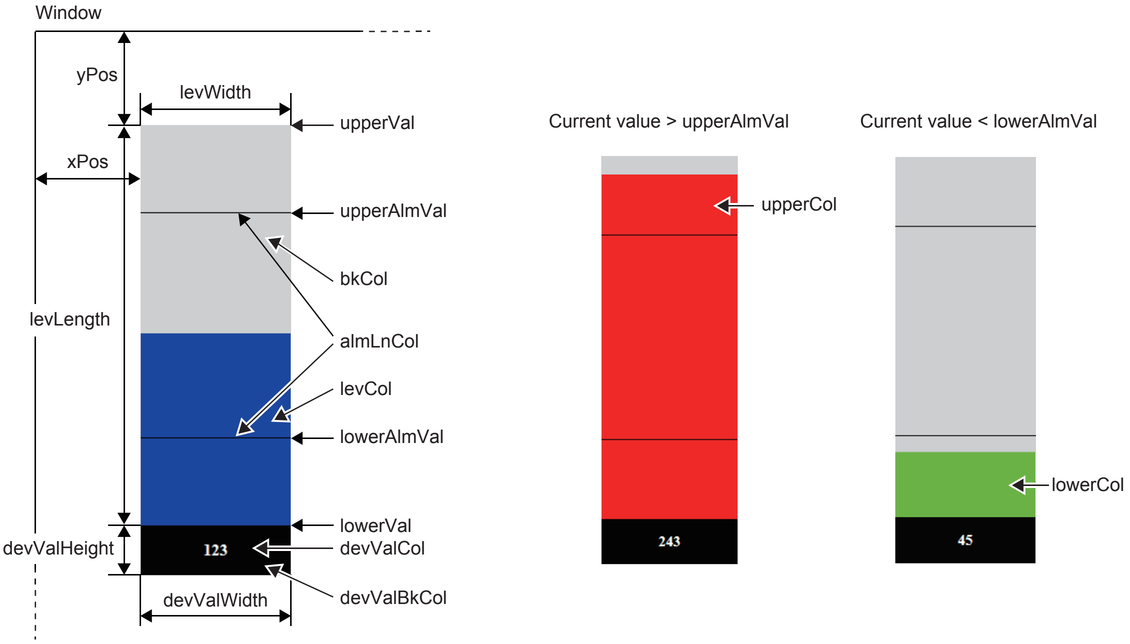
Precautions

- 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.

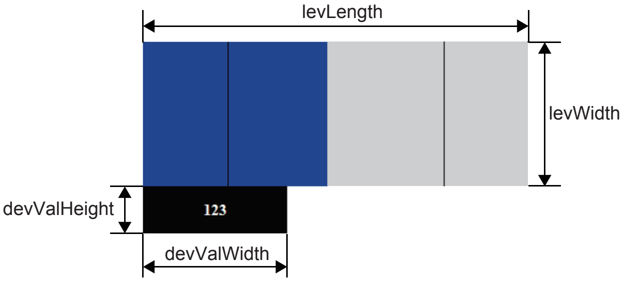
Level display (WSLevel)

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

○: Required, —: Optional

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	0: Do Not Display 1: Display	— (If omitted, it is 1)
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		

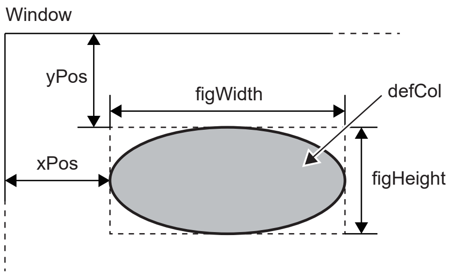
Precautions

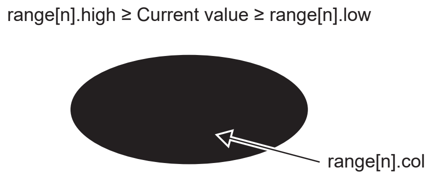
Write all the necessary parameters. If the necessary parameters are not written or the setting value is out of range, an error occurs.

Figure display (WSFigure)

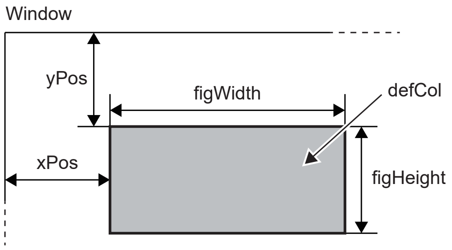
When the device value is within the specified range, a figure is displayed in the specified color. The device value is monitored in the fixed interval.

■figType = 'Oval' (oval)

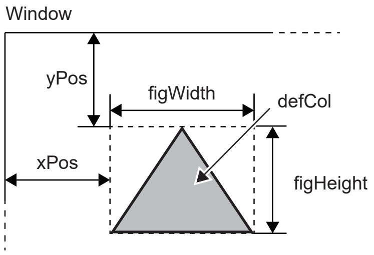




■figType = 'Rect' (rectangle)



■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)

○: Required, —: Optional

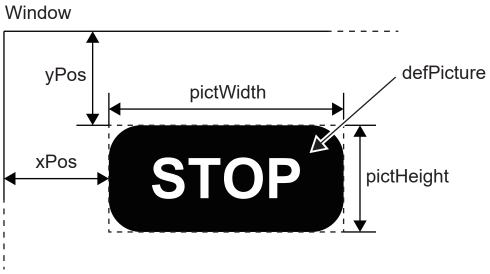
Element name	Item	Description	Requirement
devName	Device name	Device type + device number	○
devFormat	The data format of the device	0: 16-bit signed 1: 16-bit unsigned 2: 32-bit signed 3: 32-bit unsigned 4: Single-precision real number 6: Bit	○
figType	Figure type	Oval: Oval (When specifying a circle, 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) When a negative value is set, the value is treated as an absolute value.	○
figWidth	Figure width		
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)	

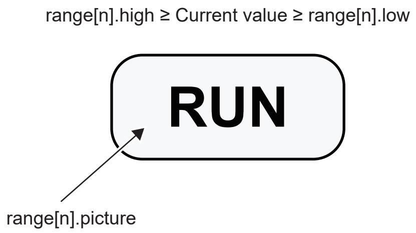
Precautions

- 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)

○: Required, —: Optional

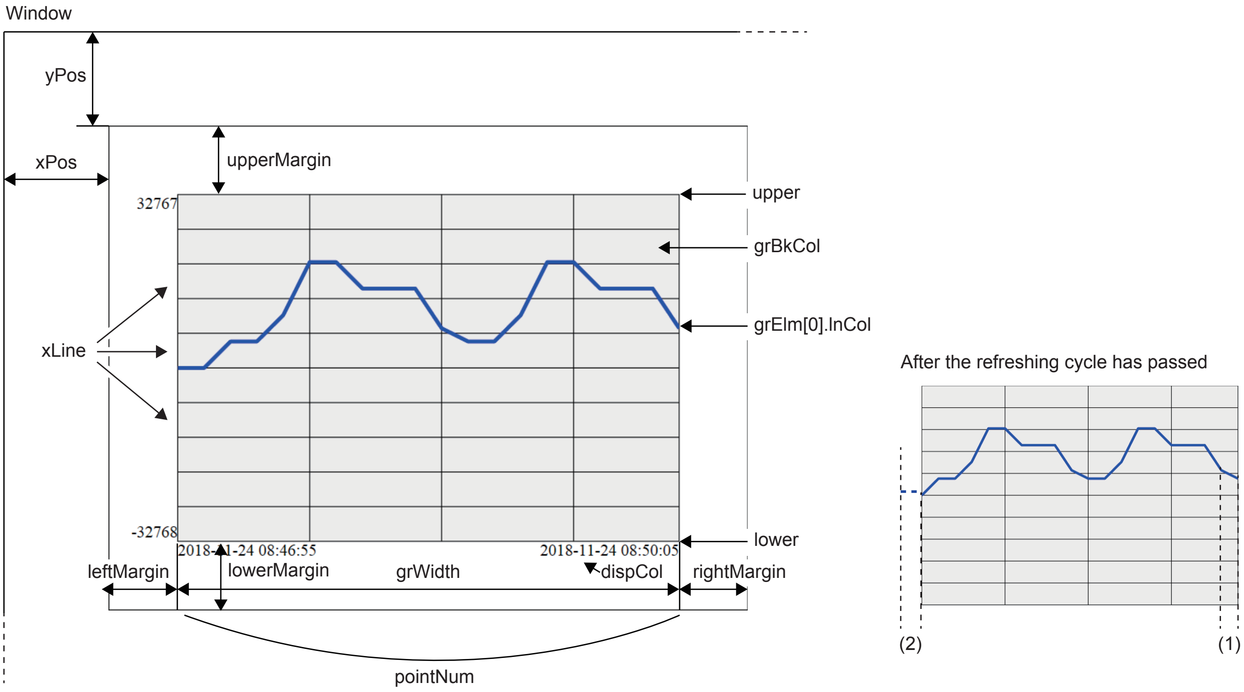
Element name	Item	Description	Requirement
devName	Device name	Device type + device number	○
devFormat	The data format of the device	0: 16-bit signed 1: 16-bit unsigned 2: 32-bit signed 3: 32-bit unsigned 4: Single-precision real number 6: Bit	○
pictHeight	Display range height of image file	Real numbers (unit: px) When a negative value is set, the value is treated as an absolute value.	○
pictWidth	Display range width of image file		
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)	—

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)

○: Required, —: Optional

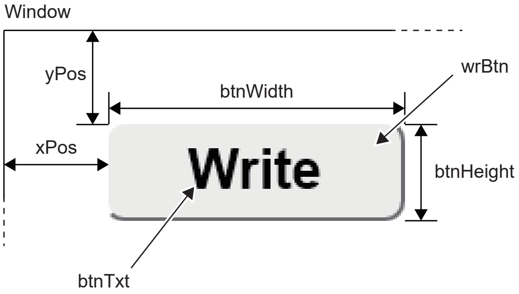
Element name	Item	Description	Requirement
grElmNum	Number of graph elements	1 to 32	○
devFormat	The data format of the device	0: 16-bit signed 1: 16-bit unsigned 2: 32-bit signed 3: 32-bit unsigned 4: Single-precision real number 6: Bit	○
grElm[0].devName	Device name	Device type + device number	○
grElm[0].lnCol	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].lnCol	Graph line color	The RGB value or color name	—
⋮			
grElm[31].devName	Device name	Device type + device number	—
grElm[31].lnCol	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		

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

○: Required, —: Optional

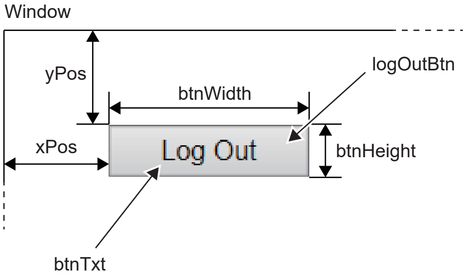
Element name	Item	Description	Requirement
devName	Device name	Device type + device number	○
devBase	Positional notation of device	K: Decimal H: Hexadecimal B: Binary	○
devFormat	The data format of the device	0: 16-bit signed 1: 16-bit unsigned 2: 32-bit signed 3: 32-bit unsigned 4: Single-precision real 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		
wrComfirm	Write confirmation message	0: Do 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)	—

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

○: Required, —: Optional

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)	—

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.

12.2 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 222 Device read CGI
WrDev.cgi	Device write CGI	Writes the specified value to the specified device.	Page 227 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.

○: Applicable in both device read CGI and device write CGI, ×: Not applicable

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)	Decimal	○	×	×
	TC (Coil)	Decimal	○	×	×
	TN (Current value)	Decimal	×	○	○
ST[*2]	STS (Contact)	Decimal	○	×	×
	STC (Coil)	Decimal	○	×	×
	STN (Current value)	Decimal	×	○	○
C[*2]	CS (Contact)	Decimal	○	×	×
	CC (Coil)	Decimal	○	×	×
	CN (Current value)	Decimal	×	○	○
LC[*2]	LCS (Contact)	Decimal	○	×	×
	LCC (Coil)	Decimal	○	×	×
	LCN (Current value)	Decimal	×	×	○
D		Decimal	×	○	○
W		Hexadecimal	×	○	○
SW		Hexadecimal	×	○	○
SM		Decimal	○	×	×
SD		Decimal	×	○	○
G (U□\G□)		Decimal	×	○	○
Z		Decimal	×	○	○
LZ		Decimal	×	×	○
R		Decimal	×	○	○

[*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 220 Device name
TYP1	string	Device size of the first point	B: Bit W: Word D: Double word ☞ Page 221 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 the 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 completion 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.

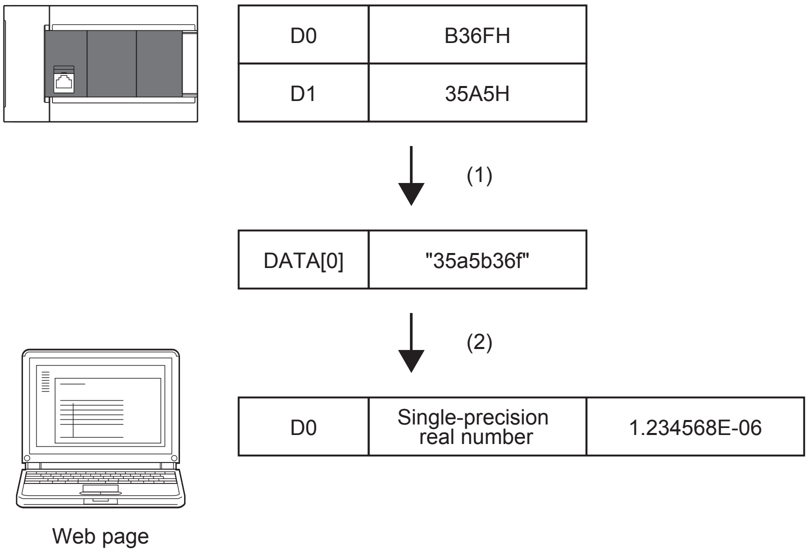
```
{"RET":"0000","DATA":["100", ··· ,"FABC"]}
```

When the execution result is abnormal, only RET is transferred.

```
{"RET":"4031"}
```

Point

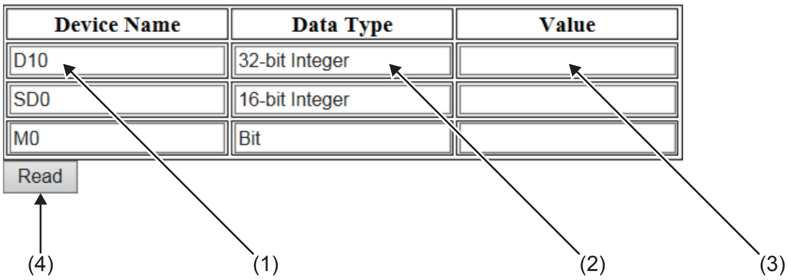
To display the device value in the real number format on the Web page, specify D: Double word for the device size and execute a request (1). Convert the read data to the real number format by using JavaScript (2).



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	

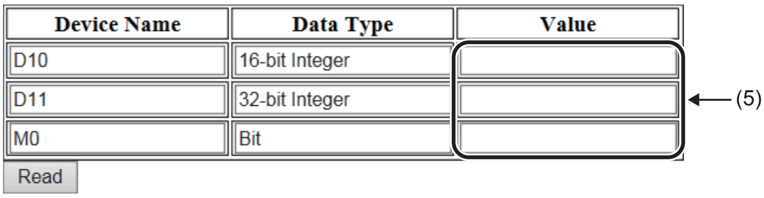
(1) Set DEV1.
(2) Set TYP1.
(3) Set VAL1.
(4) Click to call the ReadDeviceRandomTbl function of JavaScript. (Set the table ID for the argument)

■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);
         }

// 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 ) {
         // End it if the response code is other than "200 OK".
         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	

(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 220 Device name
TYP1	string	Device size	B: Bit W: Word D: Double word ☞ Page 221 Device size
DATA1	string	Write value	Hexadecimal string

The format of the request data is the query string.

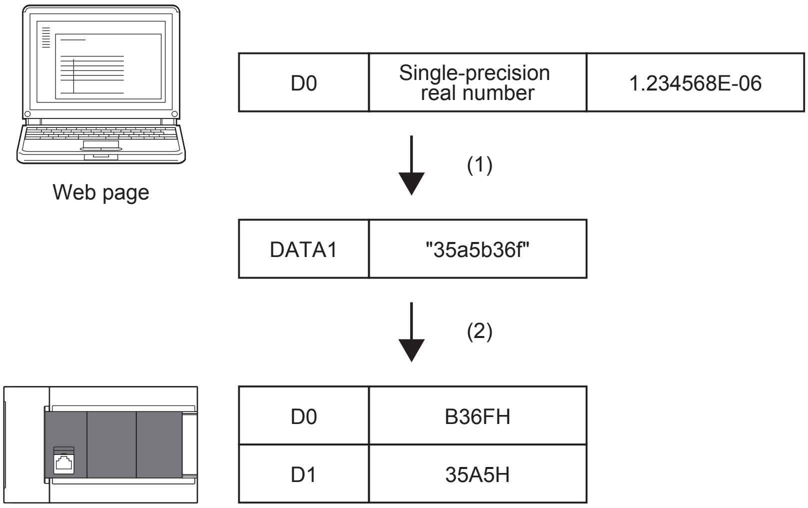
Ex.

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 completion 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: 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	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"]}
```

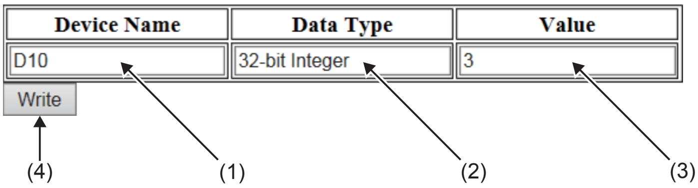
When the execution result is abnormal, data is not written and only the RET is transferred.

```
{"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

(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);
     }
```

```
// 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 ) {
          // End it if the response code is other than "200 OK".
          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

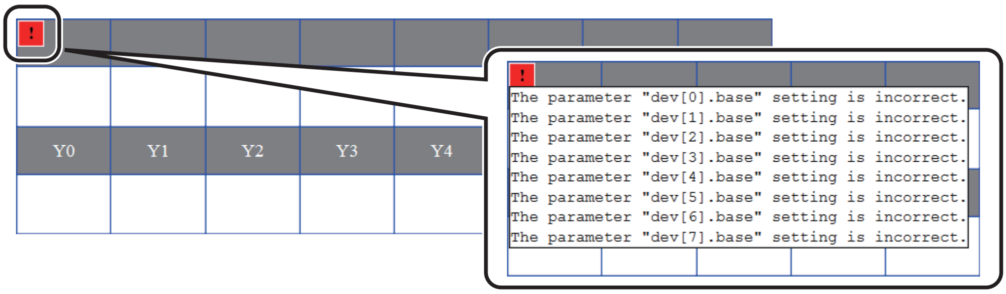
In the above example, the request parameter is as follows:

NUM=1&DEV1=D10&TYP1=D&DATA1=3

12.3 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.



```
The parameter "dev[0].base" setting is incorrect.
The parameter "dev[1].base" setting is incorrect.
The parameter "dev[2].base" setting is incorrect.
The parameter "dev[3].base" setting is incorrect.
The parameter "dev[4].base" setting is incorrect.
The parameter "dev[5].base" setting is incorrect.
The parameter "dev[6].base" setting is incorrect.
The parameter "dev[7].base" setting is incorrect.
```

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. • 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 correctly.
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 correctly.
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 correctly. 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.

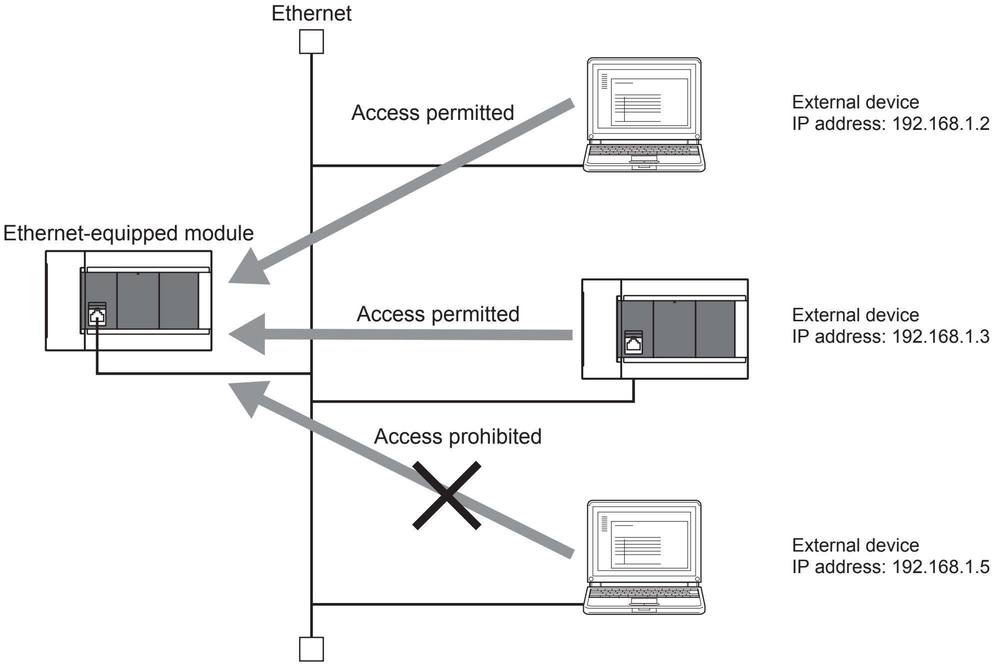
📖 MELSEC iQ-R/MELSEC iQ-F Web Server Function Guide Book

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

13 SECURITY FUNCTION

13.1 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 234 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.

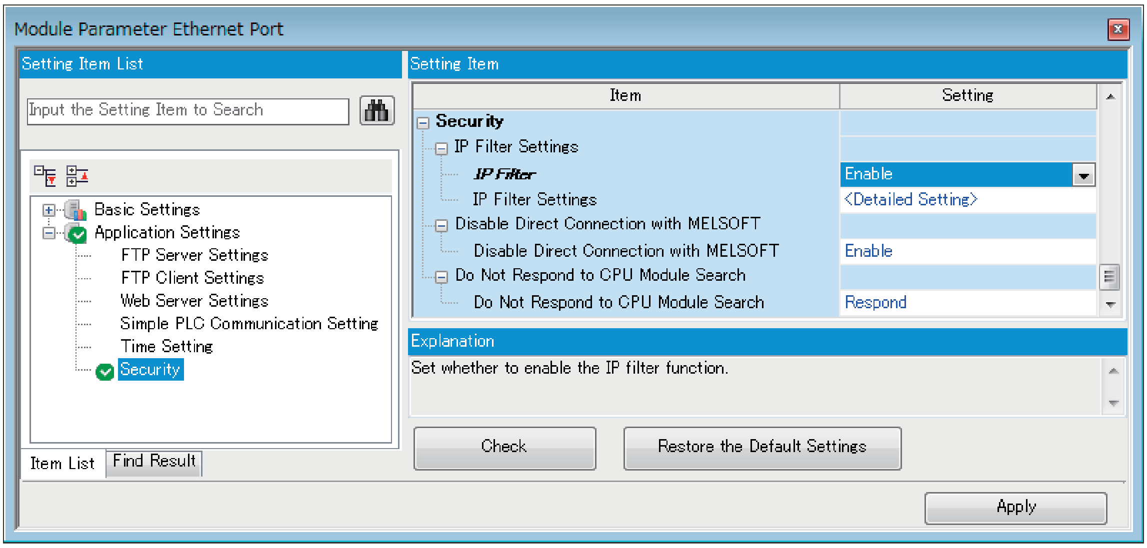
Precautions

- 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] ⇨ Module model name ⇨ [Module Parameter] ⇨ [Ethernet Port] ⇨ [Application Settings] ⇨ [Security]

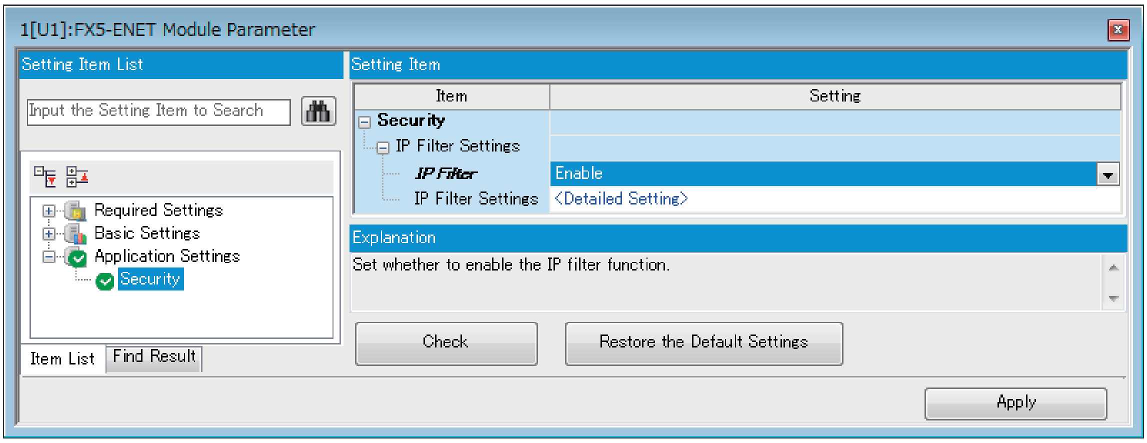


Item	Setting
Security	
IP Filter Settings	
IP Filter	Enable
IP Filter Settings	<Detailed Setting>
Disable Direct Connection with MELSOFT	
Disable Direct Connection with MELSOFT	Enable
Do Not Respond to CPU Module Search	
Do Not Respond to CPU Module Search	Respond

Set whether to enable the IP filter function.

[Ethernet module]

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



1[U1]:FX5-ENET Module Parameter

Item	Setting
Security	
IP Filter Settings	
IP Filter	Enable
IP Filter Settings	<Detailed Setting>

Explanation: Set whether to enable the IP filter function.

Item		Description	Setting range
IP Filter Settings	IP Filter	Set whether to enable the IP filter function.	• Disable • Enable (Default: Disable)
	IP Filter Settings	Set the IP address to be allowed or denied. (☞ Page 234 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.	• 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)

13.2 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.

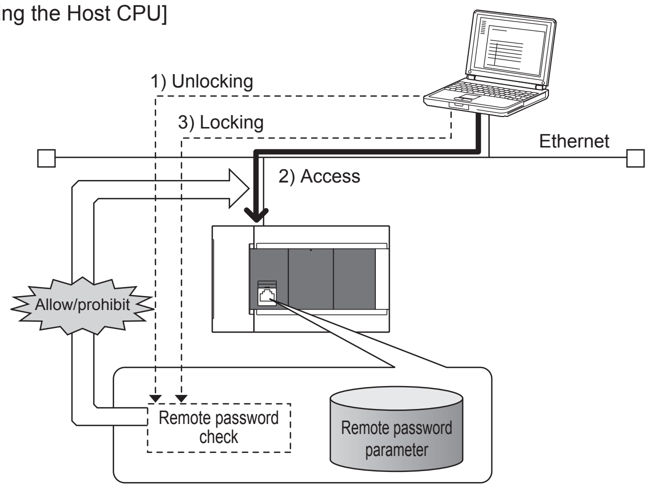
2. Access processing

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]

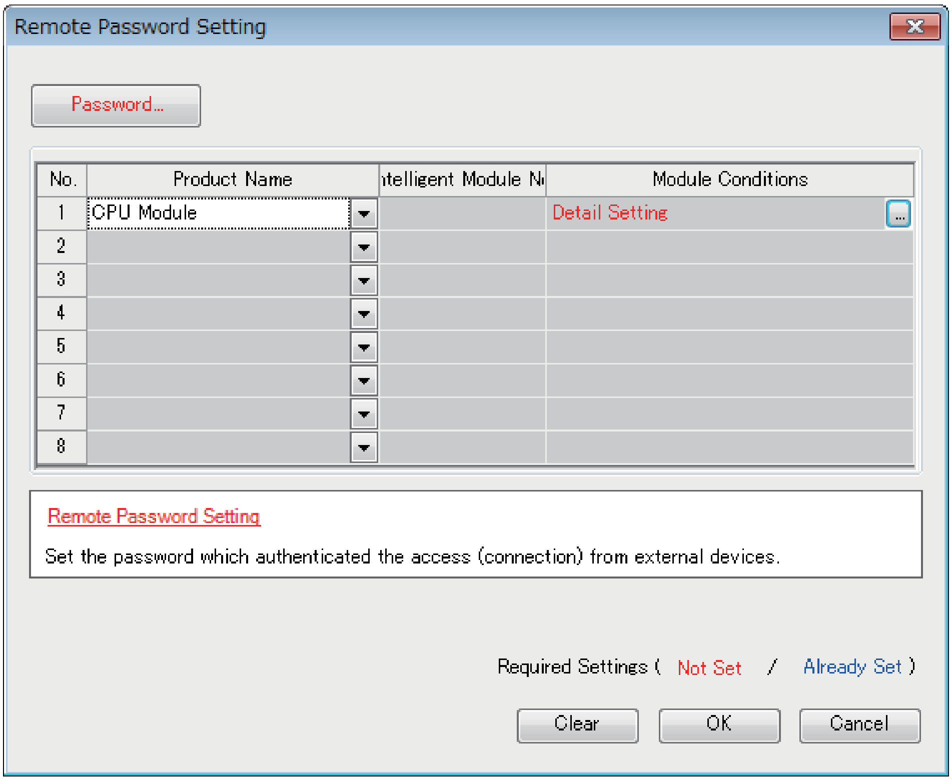


Remote password setting

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



No.	Product Name	ntelligent Module N	Module Conditions
1	CPU Module		Detail Setting
2			
3			
4			
5			
6			
7			
8			

Remote Password Setting

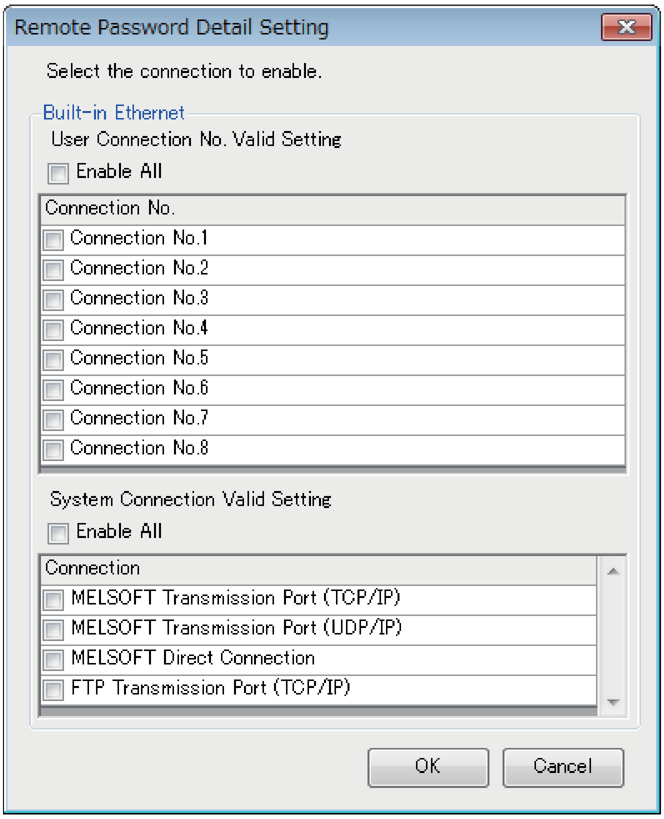
Set the password which authenticated the access (connection) from external devices.

Required Settings (Not Set / Already Set)

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



Remote Password Detail Setting

Select the connection to enable.

Built-in Ethernet

User Connection No. Valid Setting

☐ Enable All

Connection No.
☐ Connection No.1
☐ Connection No.2
☐ Connection No.3
☐ Connection No.4
☐ Connection No.5
☐ Connection No.6
☐ Connection No.7
☐ Connection No.8

System Connection Valid Setting

☐ Enable All

Connection
☐ MELSOFT Transmission Port (TCP/IP)
☐ MELSOFT Transmission Port (UDP/IP)
☐ MELSOFT Direct Connection
☐ FTP Transmission Port (TCP/IP)

Item			Description	Setting range
Built-in Ethernet	User Connection No. Valid Setting[*2]	Connection No.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.)	Check/Do not check checkbox for the target connection
	System Connection Valid Setting[*3]	MELSOFT Transmission Port (TCP/IP)[*4]	Select whether the remote password is to be enabled for the built-in Ethernet port.	Check/Do not check checkbox for the target connection
		MELSOFT transmission port (UDP/IP)		
		MELSOFT Direct Connection[*5]		
		FTP Transmission Port (TCP/IP)[*6]		

*2 User connection is a connection for users for communication such as SLMP communication.

*3 System connection is used by the system for communication such as MELSOFT communications (TCP/IP).

*4 Check this checkbox to enable the remote password for the ports for which the communication system is set to "MELSOFT Connection" in engineering tool.

*5 Check this checkbox to enable the remote password for CPU module direct connection to engineering tool using the built-in Ethernet port. (☞ Page 31 Direct Connection with Engineering Tool)

*6 Check this checkbox to enable the remote password for access by the file transfer function (FTP server). (☞ Page 178 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.

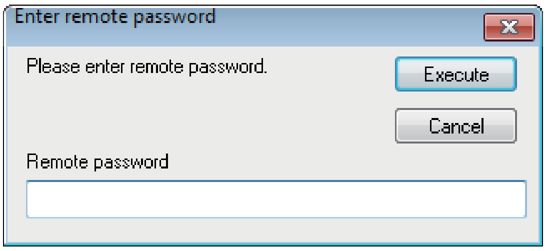
Unlocking or locking the remote password

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 602 List of Commands and Functions)

■In case of FTP Transmission Port

Use the dedicated FTP command. (☞ Page 178 , 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).

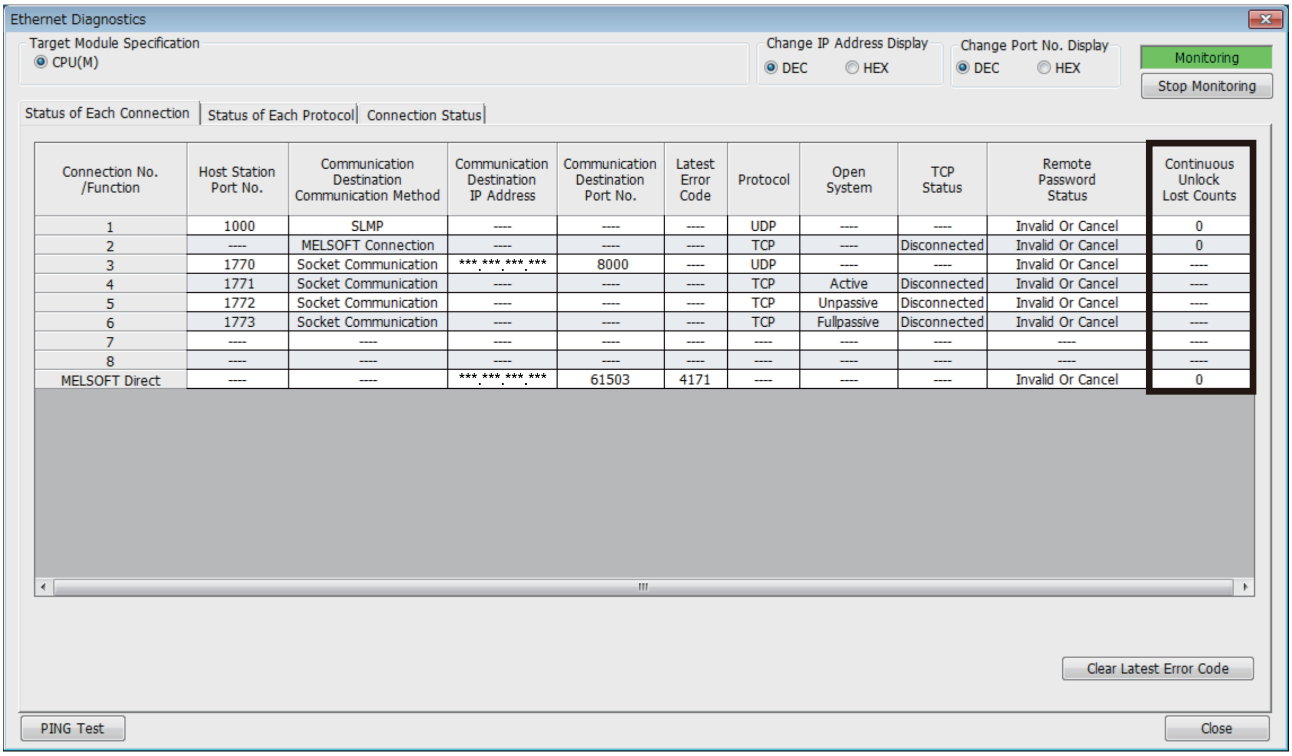
Detection of unauthorized access and actions

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"



Ethernet Diagnostics

Target Module Specification: ● CPU(M)

Change IP Address Display: ● DEC ○ HEX

Change Port No. Display: ● DEC ○ HEX

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.

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.

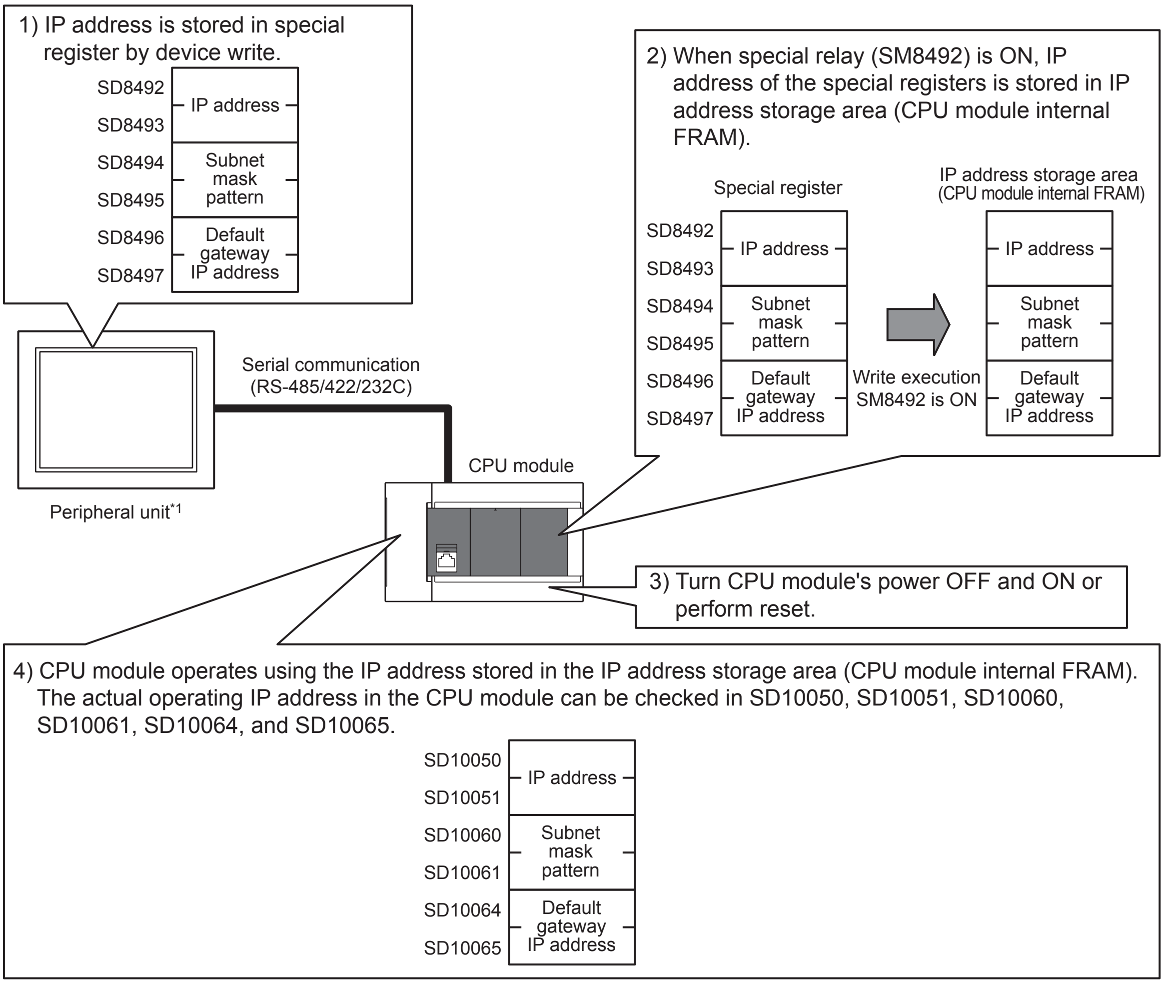
This function changes the IP address of the Ethernet-equipped module even if no settings are made in GX Works3 PLC parameters.

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 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 devices but also by MX Component, MX Sheet and link function by manipulating values of the special devices. For details on MX Component and MX Sheet, refer to the respective product manual. For details on the link function, refer to the following.

☞ Page 248 LIST OF FUNCTIONS

Point

- For details on special relay areas and special register areas to use IP address change function of CPU module, refer to ☞ Page 860 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 873 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.

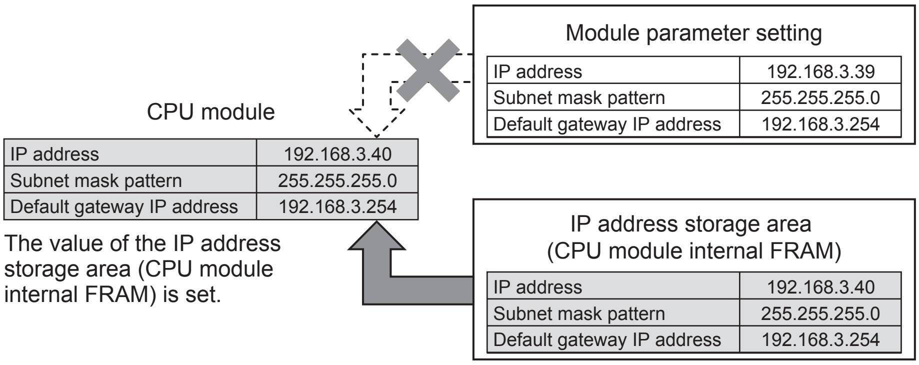
14.2 IP Address to Be Set for the Ethernet-equipped Module

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 CPU module is powered off and on or reset, the IP address and other data stored in the IP address storage area are reflected to the Ethernet-equipped module, and the IP address change function enable flag (SM8498, Un\G60) turns on or changes from 0 to 1.

Ex.

Case of the CPU module

[In the case IP address change function is used]



Module parameter setting

IP address	192.168.3.39
Subnet mask pattern	255.255.255.0
Default gateway IP address	192.168.3.254

CPU module

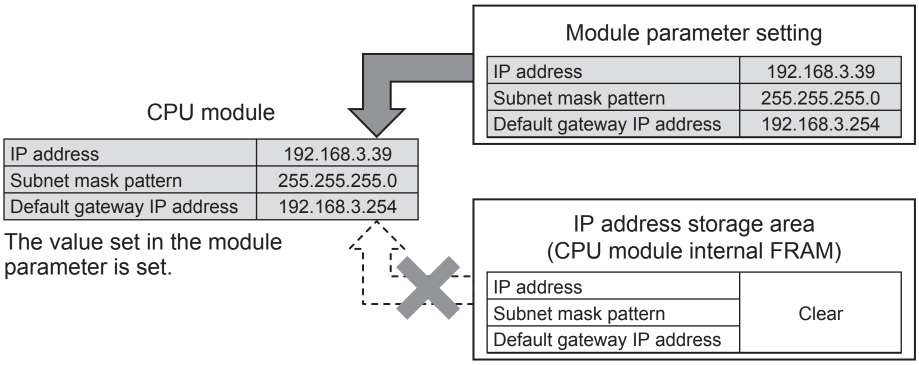
IP address	192.168.3.40
Subnet mask pattern	255.255.255.0
Default gateway IP address	192.168.3.254

The value of the IP address storage area (CPU module internal FRAM) is set.

IP address storage area
(CPU module internal FRAM)

IP address	192.168.3.40
Subnet mask pattern	255.255.255.0
Default gateway IP address	192.168.3.254

[In the case Module parameter setting is used (IP address storage area is cleared)]



Module parameter setting

IP address	192.168.3.39
Subnet mask pattern	255.255.255.0
Default gateway IP address	192.168.3.254

CPU module

IP address	192.168.3.39
Subnet mask pattern	255.255.255.0
Default gateway IP address	192.168.3.254

The value set in the module parameter is set.

IP address storage area
(CPU module internal FRAM)

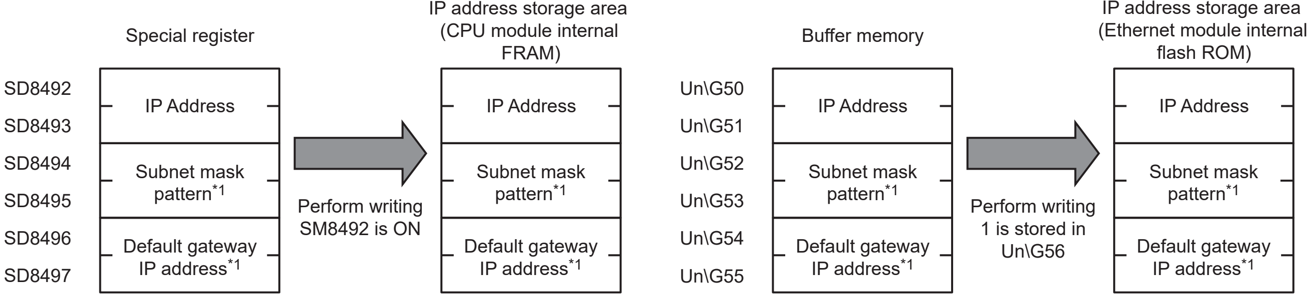
IP address	Clear
Subnet mask pattern	
Default gateway IP address	

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.


[Case of the CPU module] [Case of the Ethernet module]

*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 to the Ethernet-equipped module, powering off and on or resetting the CPU module is required.

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	Values such as the IP address setting (SD8492 to SD8497) are outside the set range.	Recheck the values such as the IP address setting (SD8492 to SD8497).

■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 memory areas.

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.	Verify that write request and clear request (SM8492 and SM8495) do not turn 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 memory areas.

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 power off or reset the CPU module while data is being written to the IP address storage area or is being cleared in the area. The values may not be applied to the IP address storage area.

Power off and on or reset the CPU module 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.

MEMO

PART 2

SERIAL COMMUNICATION

This part consists of the following chapters.

15 LIST OF FUNCTIONS

The table below shows the serial communication types supported by the FX5.

○: Supported, —: Not supported

Communication functions	Function outline	Supported function			Reference section
		FX5S CPU module	FX5UJ CPU module	FX5U/FX5UC CPU module	
N:N Network	Up to eight PLCs are connected, and data is automatically transferred among them.	○	○	○	Page 250
Parallel link	Function for connecting two FX5 PLCs and mutually linking devices.	○	○	○	Page 286
MC protocol	MC protocol is used to access CPU module and external devices (such as a personal computer or an HMI) from equipment that supports MC protocol via Ethernet or serial communication. The serial port of the FX5 is capable of communication with MC protocol 1C/3C/4C frame.	○	○	○	Page 307
Inverter communication	An FX PLC can control up to 16 inverters via RS-485 communication.	○	○	○	Page 329
Non-protocol communication	Non-protocol serial communication is available between an FX PLC and RS-232C/RS-485 interface equipment such as bar code reader, printer, personal computer and measuring instrument.	○	○	○	Page 411
Predefined protocol support	Data can be sent and received between the counterpart device and CPU module with a protocol appropriate to the counterpart device (such as measuring instrument or barcode reader).	○	○	○	Page 452

MEMO

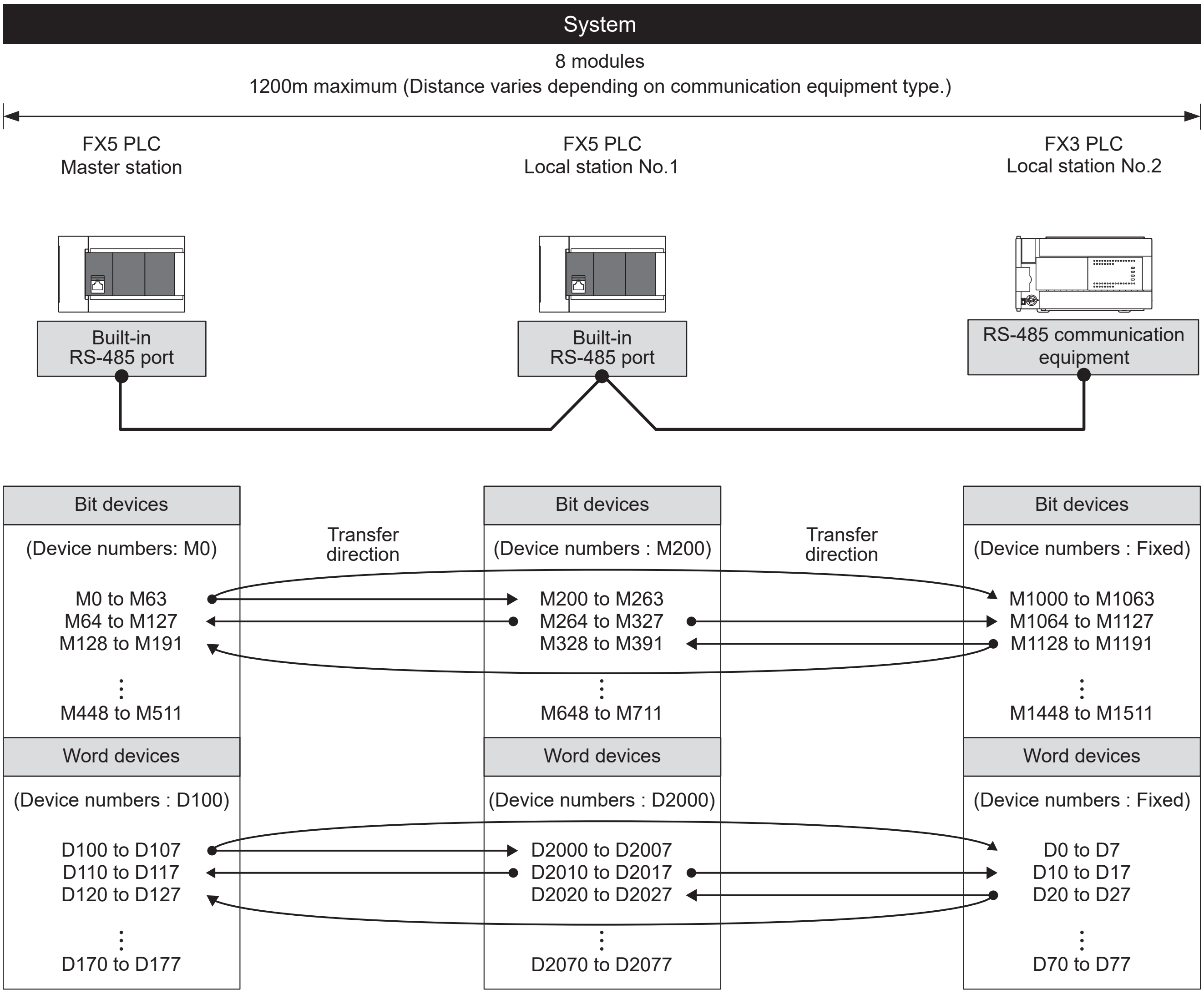
16 N:N NETWORK

This chapter explains the N:N Network.

16.1 Outline of System

N:N Network allows connection of up to eight FX5 PLCs or FX3 PLCs via mutually linked devices through communication in accordance with RS-485.

- One of three patterns can be selected according to the number of devices to be linked.
- Data link is automatically updated between up to eight FX5 PLCs or FX3 PLCs.
- The total extension distance is 1200m maximum. (only when the FX5-485ADP is used in the configuration)
- FX5 PLCs can set head device number for internal relay (M) and data register (D) for link.



The link information can be monitored in the master station and all local stations.

The figure above shows the maximum number of linked devices. There are differences in specification depending on the selected link pattern and FX Series.

16.2 Procedures Before Operation

The flow chart below shows the N:N Network setting procedures up until data link:

1. Check communication specifications

For communication specifications, link specifications, link pattern and number of link points, and link time, refer to ☞ Page 254 Specifications.

2. System configuration and selection

For system configuration and selection of communication devices, refer to ☞ Page 251 System Configuration.

3. Wiring

For wiring with twisted pair cable and wiring example, refer to ☞ Page 257 Wiring.

4. Communication settings[*1]

For communication settings of communication device, refer to ☞ Page 262 Communication Setting.

5. Program creation

For communication test program, master station program, and local station program, refer to ☞ Page 264 Programming.

*1 For more details on the connection and operation methods between the engineering tool (GX Works3) and PLC, refer to the manual below.
📖 GX Works3 Operating Manual

16.3 System Configuration

This section outlines the system configuration required to use the N:N Network.

Restriction

In N:N Network, only one channel is available to use for one CPU module.

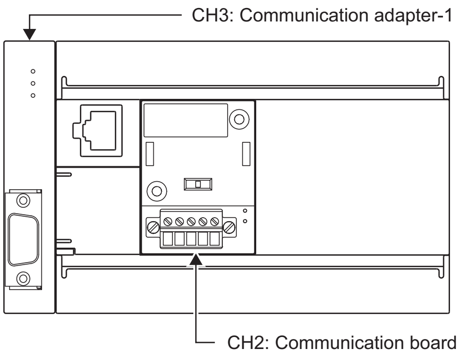
FX5S CPU module

The N:N Network function can be used in the FX5S CPU module by using the communication board and communication adapter.

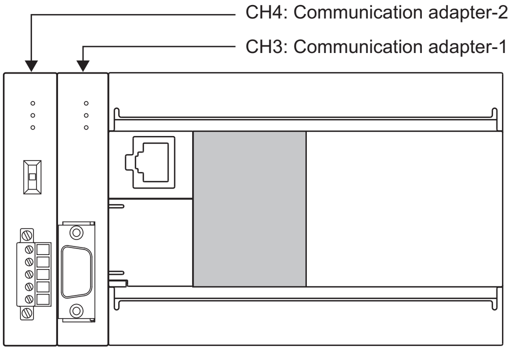
Communication channel assignments are fixed regardless of the system configuration.

The combinations available for the system configurations are shown below.

[Configuration example 1]



[Configuration example 2]



Item		Serial port	Important points in selection	Overall distance
Communication board	FX5-485-BD	CH2	Mount the communication board on the front of the CPU module.	50 m or less
Communication adapter	FX5-485ADP	CH3, CH4[*1]	Mount the communication adapter to the left of the CPU module.	1200 m or less

*1 The adapters are assigned to CH3 and CH4 in the order, from the closest one to the CPU module.

Precautions

Total number of communication boards and communication adapters that can be connected is two.

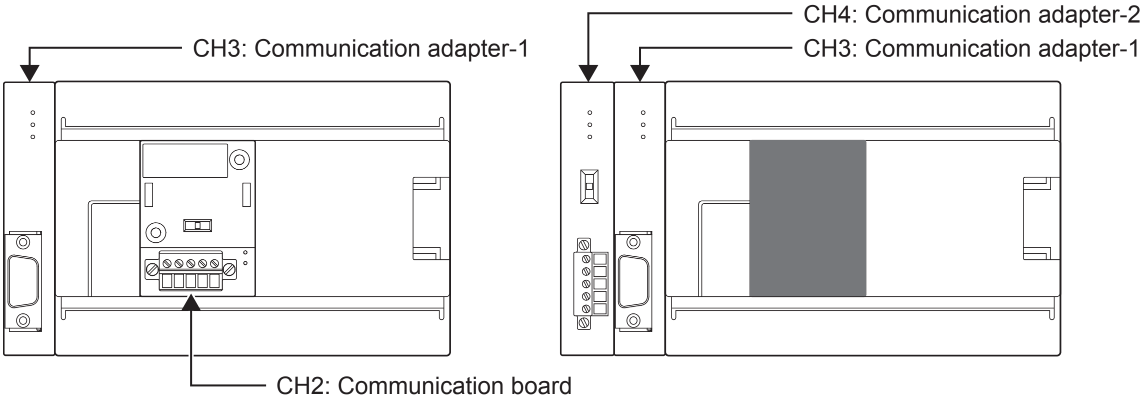
FX5UJ CPU module

The N:N Network function can be used in the FX5UJ CPU module by using the communication board and communication adapter.

Communication channel assignments are fixed regardless of the system configuration.

The combinations which can be configured are shown below.

[Configuration example 1] [Configuration example 2]



Item		Serial port	Important points in selection	Overall distance
Communication board	FX5-485-BD	CH2	Mount the communication board on the front of the CPU module.	50 m or less
Communication adapter	FX5-485ADP	CH3, CH4[*1]	Mount the communication adapter to the left of the CPU module.	1200 m or less

*1 The adapters are assigned to CH3 and CH4 in the order, from the closest one to the CPU module.

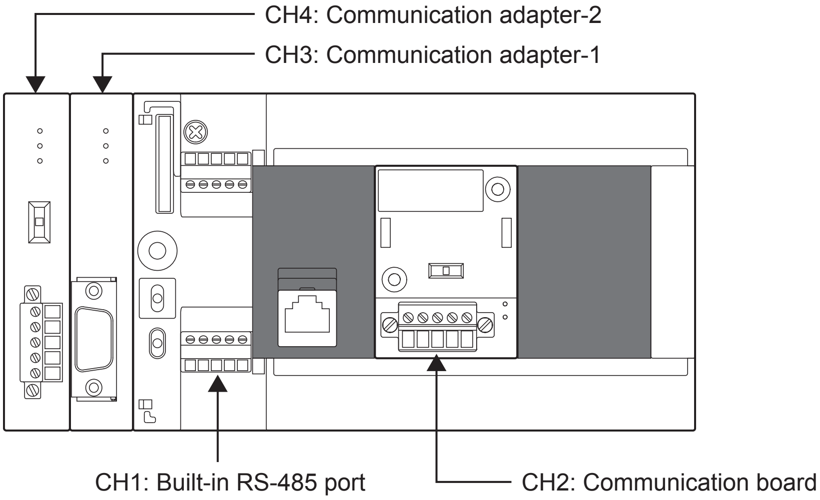
Precautions

Total number of communication boards and communication adapters that can be connected is two.

FX5U CPU module

The N:N Network function can be used in the FX5U CPU module by using the built-in RS-485 port, communication board, and communication adapter.

Communication channel assignments are fixed regardless of the system configuration.



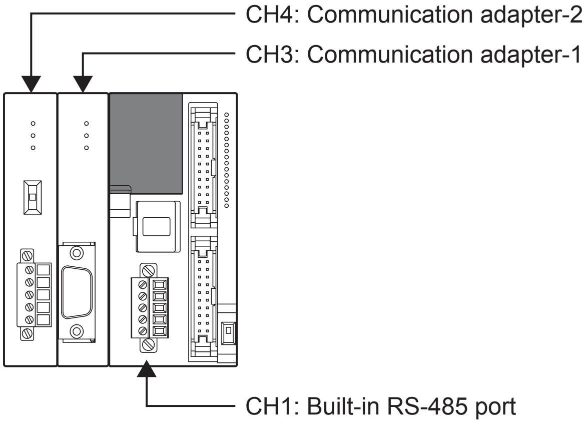
Item		Serial port	Important points in selection	Overall distance
Built-in RS-485 port		CH1	Since it is built-into the CPU module, there is no need to add equipment.	50 m or less
Communication board	FX5-485-BD	CH2	Mount the communication board on the front of the CPU module.	50 m or less
Communication adapter	FX5-485ADP	CH3, CH4[*1]	Mount the communication adapter to the left of the CPU module.	1200 m or less

*1 The adapters are assigned to CH3 and CH4 in the order, from the closest one to the CPU module.

FX5UC CPU module

The N:N Network function can be used in the FX5UC CPU module by using the built-in RS-485 port and communication adapter.

Communication channel assignments are fixed regardless of the system configuration.



Item		Serial port	Important points in selection	Overall distance
Built-in RS-485 port		CH1	Since it is built-into the CPU module, there is no need to add equipment.	50 m or less
Communication adapter	FX5-485ADP	CH3, CH4[*1]	Mount the communication adapter to the left of the CPU module.	1200 m or less

*1 The adapters are assigned to CH3 and CH4 in the order, from the closest one to the CPU module.

16.4 Specifications

This section describes the communication specifications and performance of the N:N Network function.

For the specifications of FX3 PLC, refer to FX Series User's Manual - Data Communication Edition.

Communication specifications

The N:N Network function is executed according to the communication specifications (fixed) shown in the table below.

Specification items such as baud rate cannot be changed.

Items		Specifications	Remarks
Number of connectable modules		Maximum of 8	—
Transmission standard		RS-485 standard	—
Maximum overall distance		When only FX5-485ADP is used in the configuration: 1200 m or less When configuring with FX5-485ADP or FX3U-485ADP: 500 m or less Other than the above configuration: 50 m or less	When the built-in RS-485 port, FX5-485-BD and 485-BD for FX3 Series are all connected: 50 m or less
Protocol type		N:N Network	—
Control procedure		—	—
Communication method		Half-duplex, bi-directional communication	—
Baud rate		38400 bps	—
Character format	Start bit	1 bit	—
	Data length	7 bits	—
	Parity bit	Even	—
	Stop bit	1 bit	—
Header		Fixed	—
Terminator		Fixed	—
Control line		—	—
Sum check		Fixed	—

Link specifications

Link pattern and number of link points

Settings of the link pattern are performed with GX Works3 Serial Communication setting. (☞ Page 262 Communication Setting)

The number of occupied points of the link device differ depending on the number of link patterns and local stations to be used. The devices to be occupied are assigned according to the head number of the link device.

When FX3 PLCs are used, even if the link device numbers are different from that of FX5 PLC (the link device numbers of FX3 PLC are fixed), the link devices corresponding to its station number are linked.

Station number		**Model**	**Pattern 0**		**Pattern 1**		**Pattern 2**	
			Internal relay (M)	**Data register (D)**	**Internal relay (M)**	**Data register (D)**	**Internal relay (M)**	**Data register (D)**
			0 points	**4 in each station**	**32 in each station**	**4 in each station**	**64 in each station**	**8 in each station**
Master station	Station No. 0	FX5	—	D (x) to D (x + 3)	M (y) to M (y + 31)	D (x) to D (x + 3)	M (y) to M (y + 63)	D (x) to D (x + 7)
		FX3		D0 to D3	M1000 to M1031	D0 to D3	M1000 to M1063	D0 to D7
Local stations	Station No. 1	FX5	—	D (x + 10) to D (x + 13)	M (y + 64) to M (y + 95)	D (x + 10) to D (x + 13)	M (y + 64) to M (y + 127)	D (x + 10) to D (x + 17)
		FX3		D10 to D13	M1064 to M1095	D10 to D13	M1064 to M1127	D10 to D17
	Station No. 2	FX5	—	D (x + 20) to D (x + 23)	M (y + 128) to M (y + 159)	D (x + 20) to D (x + 23)	M (y + 128) to M (y + 191)	D (x + 20) to D (x + 27)
		FX3		D20 to D23	M1128 to M159	D20 to D23	M1128 to M1191	D20 to D27
	Station No. 3	FX5	—	D (x + 30) to D (x + 33)	M (y + 192) to M (y + 223)	D (x + 30) to D (x + 33)	M (y + 192) to M (y + 255)	D (x + 30) to D (x + 37)
		FX3		D30 to D33	M1192 to M1223	D30 to D33	M1192 to M1255	D30 to D37
	Station No. 4	FX5	—	D (x + 40) to D (x + 43)	M (y + 256) to M (y + 287)	D (x + 40) to D (x + 43)	M (y + 256) to M (y + 319)	D (x + 40) to D (x + 47)
		FX3		D40 to D43	M1256 to M1287	D40 to D43	M1256 to M1319	D40 to D47
	Station No. 5	FX5	—	D (x + 50) to D (x + 53)	M (y + 320) to M (y + 351)	D (x + 50) to D (x + 53)	M (y + 320) to M (y + 383)	D (x + 50) to D (x + 57)
		FX3		D50 to D53	M1320 to M1651	D50 to D53	M1320 to M1383	D50 to D57
	Station No. 6	FX5	—	D (x + 60) to D (x + 63)	M (y + 384) to M (y + 415)	D (x + 60) to D (x + 63)	M (y + 384) to M (y + 447)	D (x + 60) to D (x + 67)
		FX3		D60 to D63	M1384 to M1415	D60 to D63	M1384 to M1447	D60 to D67
	Station No. 7	FX5	—	D (x + 70) to D (x + 73)	M (y + 448) to M (y + 479)	D (x + 70) to D (x + 73)	M (y + 448) to M (y + 511)	D (x + 70) to D (x + 77)
		FX3		D70 to D73	M1448 to M1479	D70 to D73	M1448 to M1511	D70 to D77

x: Link device head number of data register (D)
y: Link device head number of internal relay (M)

Precautions

When creating a program, do not change the information in devices used by other stations. If such information is changed, the other stations will not operate normally.

Ex.

Refresh range: pattern 1, link device head number: all station D1000/M4000

Master station: FX5 PLC, local station: FX5 PLC × 3

Station number		Master station	Local stations		
		Station No.0 (FX5 PLC)	Station No.1 (FX5 PLC)	Station No.2 (FX5 PLC)	Station No.3 (FX5 PLC)
Master station	Station No.0	D1000 to D1003 M4000 to M4031	D1000 to D1003 M4000 to M4031	D1000 to D1003 M4000 to M4031	D1000 to D1003 M4000 to M4031
Local stations	Station No.1	D1010 to D1013 M4064 to M4095	D1010 to D1013 M4064 to M4095	D1010 to D1013 M4064 to M4095	D1010 to D1013 M4064 to M4095
	Station No.2	D1020 to D1023 M4128 to M4159	D1020 to D1023 M4128 to M4159	D1020 to D1023 M4128 to M4159	D1020 to D1023 M4128 to M4159
	Station No.3	D1030 to D1033 M4192 to M4223	D1030 to D1033 M4192 to M4223	D1030 to D1033 M4192 to M4223	D1030 to D1033 M4192 to M4223

In the example above, devices from D1000 to D1033 (34 points from the head device number of D) and from M4000 to M4223 (224 points from the head device number of M) are occupied.

Other devices can be used as general control devices. Link devices for unconnected local stations can be used as general control devices; however, it is recommended to save the unoccupied link devices for the local stations to be used in the future.

Ex.

Refresh range: pattern 2, link device head number: station 0 (D200/M2000), station 2 (D500/M3000)

Master station: FX5 PLC, local station: FX5 PLC × 1/FX3 PLC × 2

Station number		Master station	Local stations		
		Station No.0 (FX5 PLC)	Station No.1 (FX3 PLC)	Station No.2 (FX5 PLC)	Station No.3 (FX3 PLC)
Master station	Station No.0	D200 to D207 M2000 to M2063	D0 to D7 M1000 to M1063	D500 to D507 M3000 to M3063	D0 to D7 M1000 to M1063
Local stations	Station No.1	D210 to D217 M2064 to M2127	D10 to D17 M1064 to M1127	D510 to D517 M3064 to M3127	D10 to D17 M1064 to M1127
	Station No.2	D220 to D227 M2128 to M2191	D20 to D27 M1128 to M1191	D520 to D527 M3128 to M3191	D20 to D27 M1128 to M1191
	Station No.3	D230 to D237 M2192 to M2255	D30 to D37 M1192 to M1255	D530 to D537 M3192 to M3255	D30 to D37 M1192 to M1255

The link devices are allocated as above and the link device numbers are different according to the station number (for the FX3 series, the numbers are fixed). However, the link device numbers are linked to the corresponding devices.

The link device head number of FX5 PLC can be set individually for each station; however, unifying the numbers in the system is recommended to prevent disordered numbers.

Link time

The link time indicates the cycle time from the start of parameter message editing until link devices are updated and parameter message editing is restarted.

The link time varies depending on the number of linked modules (master station and local stations) and the number of linked devices as shown in the table below.

Number of linked modules	**Pattern 0**	**Pattern 1**	**Pattern 2**
	0 bit-devices (M) **4 word-devices (D)**	**32 bit-devices (M)** **4 word-devices (D)**	**64 bit-devices (M)** **8 word-devices (D)**
2	20 ms	24 ms	37 ms
3	29 ms	35 ms	52 ms
4	37 ms	45 ms	70 ms
5	46 ms	56 ms	87 ms
6	54 ms	67 ms	105 ms
7	63 ms	78 ms	122 ms
8	72 ms	88 ms	139 ms

16.5 Wiring

This section explains about the wiring.

For the wiring of FX3 PLC, refer to FX Series User's Manual - Data Communication Edition.

Wiring procedure

1. Prepare for wiring.

Prepare cables required for wiring. (☞ Page 258 Cable)

2. Turn OFF the PLC power.

Before wiring, make sure that the PLC power is OFF.

3. Wire the cables between the communication equipment.

Connect the RS-485 communication equipment. (☞ Page 261 Connection diagram)

Cable

Select cables using the procedure described below.

Twisted pair cable

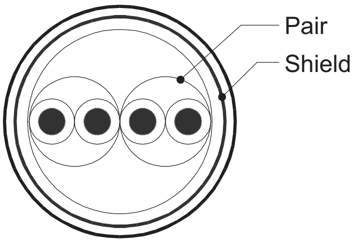
Use shielded twisted pair cables for connecting RS-485 communication equipment.

The specifications of the cables used in wiring are shown.

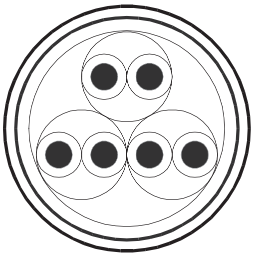
■RS-485 cable specifications

Items	Descriptions
Cable type	Shielded cable
Number of pairs	2p, 3p
Conductor resistance (20℃)	88.0 Ω/km or less
Insulation resistance	10000 MΩ-km or more
Voltage resistance	500VDC, 1 minute
Electrostatic capacitance (1kHz)	60 nF/km or less by an average
Characteristic impedance (100kHz)	110±10 Ω

■Cable structural drawing (reference)



Example of two-pair cable structural drawing



Example of three-pair cable structural drawing

Connecting cables

The table below shows applicable cables and tightening torques.

Item	Number of wires connected per terminal	Wire size		Tightening torque
		Solid wire, Stranded wire	Wire ferrule with insulation sleeve	
FX5U CPU module built-in RS-485 port	One wire	0.2 to 0.5mm² (24 to 20 AWG)	0.2 to 0.5mm² (24 to 20 AWG)	0.22 to 0.25N·m
	Two wires	0.2mm² (24 AWG)	—	
FX5UC CPU module built-in RS-485 port FX5-485-BD FX5-485ADP	One wire	0.3 to 0.5mm² (22 to 20 AWG)	0.3 to 0.5mm² (22 to 20 AWG)	
	Two wires	0.3mm² (22 AWG)	—	

Precautions

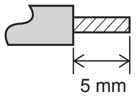
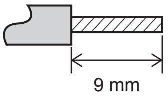
Do not tighten terminal screws exceeding the specified torque range. Otherwise it may cause equipment failure or malfunction.

■Wire end treatment

With regard to the cable end treatment, use a stranded cable or solid cable as is, or use a wire ferrule with insulating sleeve.

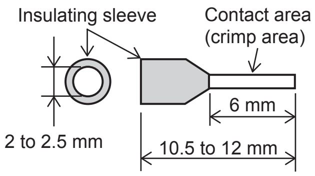
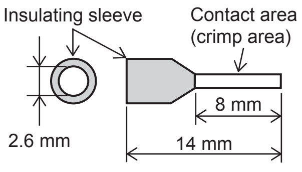
- When using a stranded cable or solid cable as is

- Twist the end of stranded wire and make sure that there are no loose wires.
- Please do not solder plate the ends of the cable.

Dimensions of the cable end	
FX5U CPU module built-in RS-485 port	**FX5UC CPU module built-in RS-485 port, FX5-485-BD, FX5-485ADP**
	

- When using a wire ferrule with insulation sleeve

Because it is difficult to insert a cable into an insulating sleeve depending on the thickness of the cable sheath, select the proper cable according to the outline drawing.

FX5U CPU module built-in RS-485 port	FX5UC CPU module built-in RS-485 port, FX5-485-BD, FX5-485ADP
	

<Reference>

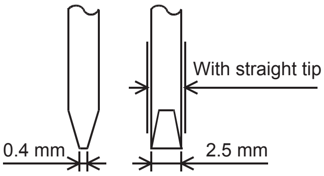
Item	Manufacturer	Model name	Crimping tool
FX5U CPU module built-in RS-485 port	PHOENIX CONTACT GmbH & Co. KG	AI 0.5-6 WH	CRIMPFOX 6 CRIMPFOX 6T-F
FX5UC CPU module built-in RS-485 port FX5-485-BD FX5-485ADP		AI 0.5-8 WH	

- Tool

For tightening the terminal, use a commercially available small screwdriver with straight tip that is not widened toward the end as shown below.

■Precautions

If the diameter of the screwdriver tip is too small, the required tightening torque cannot be achieved. To achieve the appropriate tightening torque shown in the previous page, use the following screwdriver or its equivalent (grip diameter: approximately 25mm).



<Reference>

Manufacturer	Model name
PHOENIX CONTACT GmbH & Co. KG	SZS 0.4×2.5

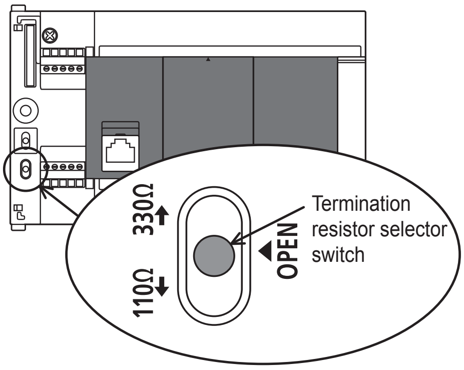
Termination resistor setting

Make sure to provide a termination resistor at both ends of the wire.

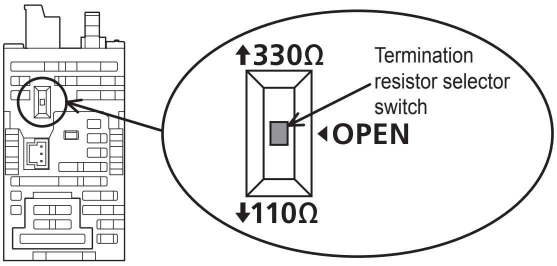
The built-in RS-485 port, FX5-485-BD and FX5-485ADP have a built-in termination resistor.

Set the termination resistor selector switch to 110 Ω.

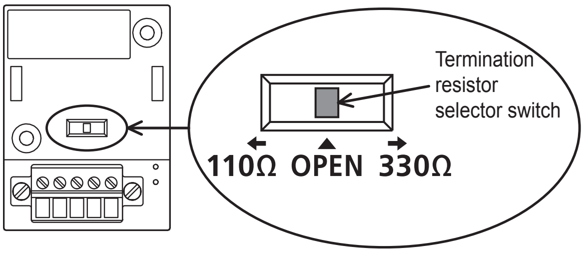
■FX5U CPU module built-in RS-485 port



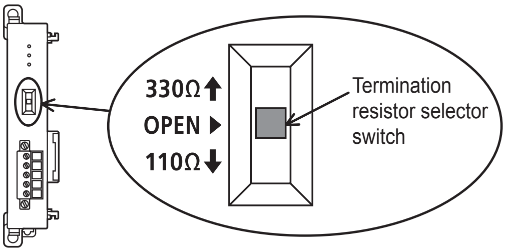
■FX5UC CPU module built-in RS-485 port



■FX5-485-BD

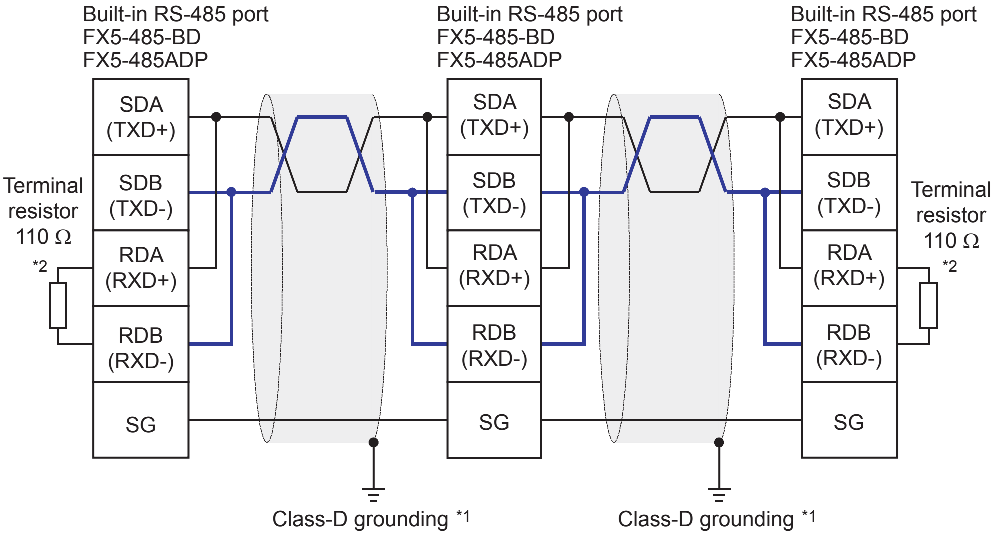


■FX5-485ADP



Connection diagram

Use one-pair wiring for an N:N Network.



[1] Make sure to perform Class-D grounding on the shield of the twisted pair cable to be connected.

[2] Make sure to provide a termination resistor at both ends of the wire. The built-in RS-485 port, FX5-485-BD and FX5-485ADP have built-in termination resistors. Set the termination resistor selector switch to 110 Ω.

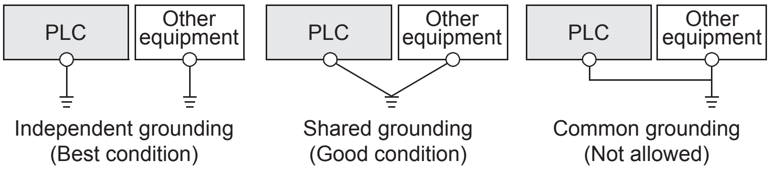
Grounding

Grounding should be performed as stated below.

- Perform D class grounding. (Grounding resistance: 100Ω or less)
- Independent grounding should be performed for best results.

If the PLC cannot be grounded independently, perform the "Shared grounding" shown below.

For details, refer to User's Manual (Hardware) of the CPU module used.



- The grounding wire size should be 14 AWG (2mm²) or larger.
- Bring the grounding point close to the PLC as much as possible so that the ground cable can be shortened.

16.6 Communication Setting

For the FX5 communication settings of this function, parameters are set using GX Works3. For details about GX Works3, refer to 🕮GX Works3 Operating Manual.

Setting of parameter differs according to the module used. The procedure for each module is as follows.

For the communication settings of FX3 PLC, refer to FX Series User's Manual - Data Communication Edition.

Built-in RS-485 port (CH1)

Navigation Window ⇨ Parameter ⇨ FX5UCPU ⇨ Module Parameter ⇨ 485 Serial Port

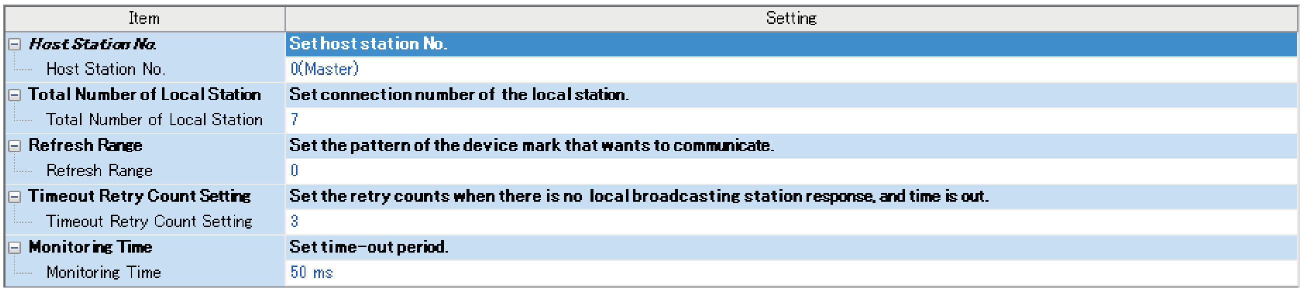
Window

The following screen will be displayed if [N:N Network] is set for the communication protocol type.

■Basic Settings

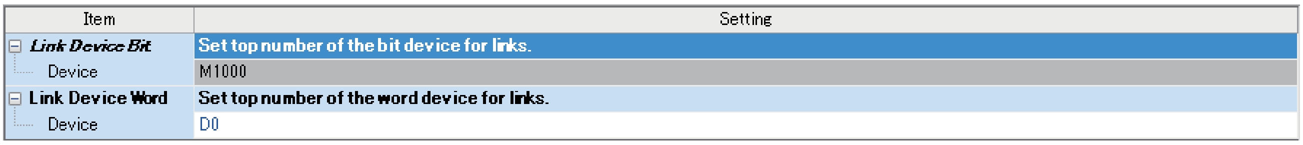
Item	Setting
⊟ *Communication Protocol Type*	Set communication protocol type.
Communication Protocol Type	N:N Network

■Fixed Setting



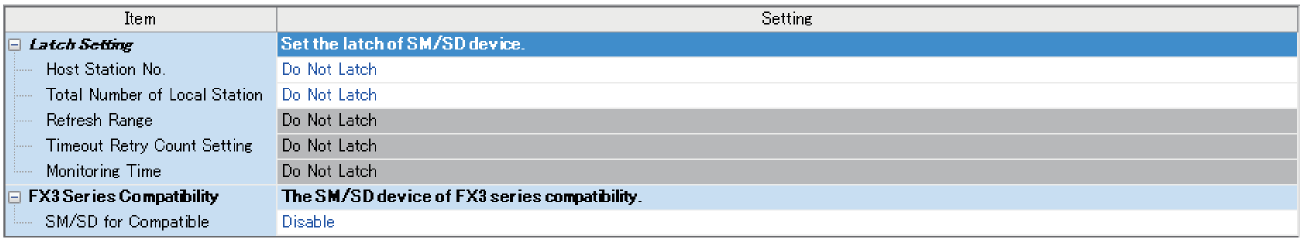
Item	Setting
⊟ *Host Station No.*	Set host station No.
Host Station No.	0(Master)
⊟ Total Number of Local Station	Set connection number of the local station.
Total Number of Local Station	7
⊟ Refresh Range	Set the pattern of the device mark that wants to communicate.
Refresh Range	0
⊟ Timeout Retry Count Setting	Set the retry counts when there is no local broadcasting station response, and time is out.
Timeout Retry Count Setting	3
⊟ Monitoring Time	Set time-out period.
Monitoring Time	50 ms

■Link Device



Item	Setting
⊟ *Link Device Bit*	Set top number of the bit device for links.
Device	M1000
⊟ Link Device Word	Set top number of the word device for links.
Device	D0

■SM/SD Setting



Item	Setting
⊟ *Latch Setting*	Set the latch of SM/SD device.
Host Station No.	Do Not Latch
Total Number of Local Station	Do Not Latch
Refresh Range	Do Not Latch
Timeout Retry Count Setting	Do Not Latch
Monitoring Time	Do Not Latch
⊟ FX3 Series Compatibility	The SM/SD device of FX3 series compatibility.
SM/SD for Compatible	Disable

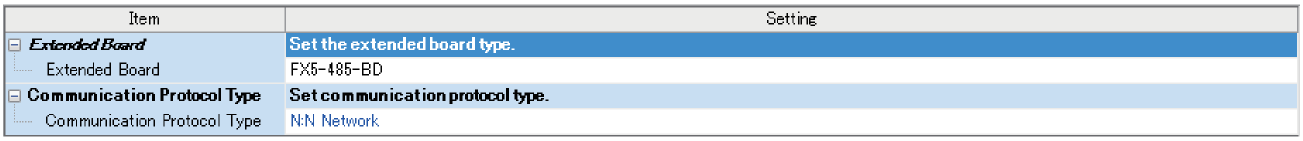
Communication board (CH2)

Navigation Window ⇨ Parameter ⇨ Model name ⇨ Module Parameter ⇨ Extended Board

Window

The following screen will be displayed if [FX5-485-BD] is set for the extended board and [N:N Network] is set for the communication protocol type. Other settings are the same as for the built-in RS-485 port (CH1).

■Basic Settings



Item	Setting
⊟ Extended Board	Set the extended board type.
Extended Board	FX5-485-BD
⊟ Communication Protocol Type	Set communication protocol type.
Communication Protocol Type	N:N Network

Communication adapter (CH3/CH4)

When an expansion adapter is used, add the expansion adapter to Module Information.

Navigation window ⇨ Parameter ⇨ Module Information ⇨ Right-click ⇨ Add New Module

After adding the expansion adapter, make settings on the screen displayed from the following operation.

Navigation window ⇨ Parameter ⇨ Module Information ⇨ ADP1 to ADP6 (Communication adapter) ⇨ Module Parameter

Window

Each setting screen is the same as for the built-in RS-485 port (CH1).

Parameter setting details

Set the following items for the serial ports that use N:N Network. However, only one channel can be set to the N:N Network.

Items			Setting value	Reference section
Basic Settings	Extended Board*1		When using this function, select [FX5-485-BD].	—
	Protocol Type		When using this function, select [N:N Network].	
Fixed Setting	Host Station No.		0(Master) 1 to 7: (Local stations)	Page 255
	Number of local stations		1 to 7	
	Refresh Range		0: Pattern 0 1: Pattern 1 2: Pattern 2	
	Timeout Retry Count Setting		0 to 10	—
	Monitoring Time		50 to 2550 (ms)	
Link Device	Link Device Bit		FX5S CPU module: M0 to M32672 FX5UJ CPU module: M0 to M7584 FX5U/FX5UC CPU module: M0 to M32672	Page 255
	Link Device Word		D0 to D7986	
SM/SD Setting	Latch Setting	Host Station No.	Latch/Do Not Latch	Page 264
		Total Number of Local Station		
	FX3 Series Compatibility	SM/SD for Compatible	Disable/CH1/CH2	

*1 Only for the communication board (CH2).

Setting is not required (fixed value) for the board below.

Item	Descriptions
Data length	7 bits
Parity	Even
Stop bit	1 bit
Start bit	1 bit
Baud rate	38400 bps
Header	Not added
Terminator	Not added
Control mode	None
Sum check code	Not added
Control procedure	Form 1 (CR and LF are not added)

FX3 Series-compatible SM/SD

When using the FX3 Series compatible SM/SD storage area, set to use special devices for either the FX3 Series CH1 or CH2.

FX3 Series compatible devices corresponding to the specified channel can be used.

For details, refer to the following.

☞ Page 273 Related Devices

Latch Setting

Set the necessity of the corresponding SD (special register) latch.

Descriptions	Setting range	Compatible devices
Station number settings	Latch/Do Not Latch	SM9080
Local station quantity setting	Latch/Do Not Latch	SM9081

16.7 Programming

This section explains how to set the N:N Network and how to create programs.

Configuration is only for FX5 PLC configuration (master station + local station × 7 stations).

In N:N Network, pattern 0, 1 or 2 can be set according to the refresh range value.

The number of used devices varies depending on the pattern.

About FX3 PLC program and communication test, refer to FX Series User's Manual - Data Communication Edition.

Communication setting

Serial communication setting are as follows. (☞ Page 262 Communication Setting)

Items		Setting							
		Master station	Local stations						
		Station No. 0	Station No. 1	Station No. 2	Station No. 3	Station No. 4	Station No. 5	Station No. 6	Station No. 7
Protocol type		N:N Network							
Fixed Setting	Host Station No.	0	1	2	3	4	5	6	7
	Total Number of Local Station	7	—	—	—	—	—	—	—
	Refresh Range	1[*1]	—	—	—	—	—	—	—
	Timeout Retry Count Setting	3	—	—	—	—	—	—	—
	Monitoring Time	50	—	—	—	—	—	—	—
Link Device	Link Device Bit[*2]	M4000	M4000	M4000	M4000	M4000	M4000	M4000	M4000
	Link Device Word[*2]	D1000	D1000	D1000	D1000	D1000	D1000	D1000	D1000

[*1] Execute communication test on the following settings: local station quantity setting: 7 (Station No.7), refresh range setting: 0 (pattern 0).

[*2] Devices can be set individually to each station.

Contents of related devices

The device used in the program are shown below. (☞ Page 273 Related Devices)

Devices for determining errors in the N:N Network

These devices are used for determining errors in the N:N Network. Use these devices to output the link errors or use them in interlock sequence programs.

FX5 only				**FX3 Series compatible**		**Name**	**Descriptions**
CH1	**CH2**	**CH3**	**CH4**	**CH1**	**CH2**		
SM8500	SM8510	SM8520	SM8530	SM8063	SM8438	Serial communication error	Turns ON when serial communication error occurred.
SM9040				SM8183		Data transfer sequence error	Turns ON when a data transfer sequence error occurred in the master station.
SM9041 to SM9047[*1]				SM8184 to SM8190[*2]		Data transfer sequence error	Turns ON when a data transfer sequence error occurred in the other local stations. However, data sequence errors that occurred in the host station (local station) cannot be detected.
SM9056				SM8191		Data transfer sequence ON	Remains ON while the N:N Network is operating.

[*1] Station number 1: SM9041, station number 2: SM9042, station number 3: SM9043 ... station number 7: SM9047

[*2] Station number 1: SM8184, station number 2: SM8185, station number 3: SM8186 ... station number 7: SM8190

Link device

The devices are assigned according to the set pattern in the refresh range setting as the following set device number (All stations, bit device (M): 4000, word device (D): 1000) set at the head. (☞ Page 255 Link specifications)

■Pattern 0

Station number	**Master station**	**Local stations**						
	Station No. 0	**Station No. 1**	**Station No. 2**	**Station No. 3**	**Station No. 4**	**Station No. 5**	**Station No. 6**	**Station No. 7**
Word devices (4 points for each station)	D1000 to D1003	D1010 to D1013	D1020 to D1023	D1030 to D1033	D1040 to D1043	D1050 to D1053	D1060 to D1063	D1070 to D1073

■Pattern 1

Station number	**Master station**	**Local stations**						
	Station No. 0	**Station No. 1**	**Station No. 2**	**Station No. 3**	**Station No. 4**	**Station No. 5**	**Station No. 6**	**Station No. 7**
Bit devices (32 points for each station)	M4000 to M4031	M4064 to M4095	M4128 to M4159	M4192 to M4223	M4256 to M4287	M4320 to M4351	M4384 to M4415	M4448 to M4479
Word devices (4 points for each station)	D1000 to D1003	D1010 to D1013	D1020 to D1023	D1030 to D1033	D1040 to D1043	D1050 to D1053	D1060 to D1063	D1070 to D1073

■Pattern 2

Station number	**Master station**	**Local stations**						
	Station No. 0	**Station No. 1**	**Station No. 2**	**Station No. 3**	**Station No. 4**	**Station No. 5**	**Station No. 6**	**Station No. 7**
Bit devices (64 points for each station)	M4000 to M4063	M4064 to M4127	M4128 to M4191	M4192 to M4255	M4256 to M4319	M4320 to M4383	M4384 to M4447	M4448 to M4511
Word devices (8 points for each station)	D1000 to D1007	D1010 to D1017	D1020 to D1027	D1030 to D1037	D1040 to D1047	D1050 to D1057	D1060 to D1067	D1070 to D1077

Precautions

When creating a program, do not change the information in devices used by other stations. If such information is changed, the other stations will not operate normally.

Communication test

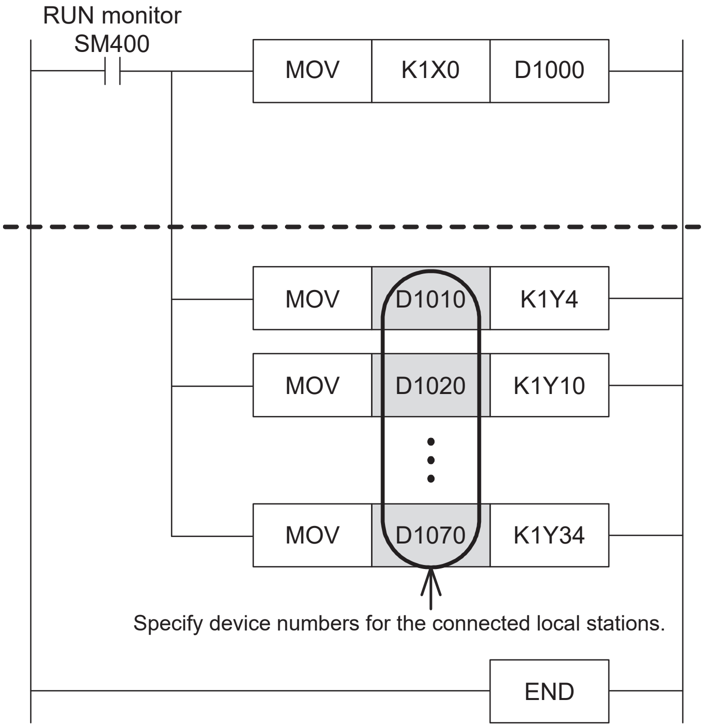
It is recommended to wire the master station and local station, perform communication settings, and then execute the communication test using the following procedure to confirm proper operation.

Communication test program is not required during operation.

Communication test process

1. After configuring the communication setting of the master station and local station, and programming, turn the CPU module power supply OFF→ON or reset.
2. Check if the SD and RD LEDs of the used serial port are flashing. If the LEDs are OFF, refer to troubleshooting. (☞ Page 794 TROUBLESHOOTING PROCEDURE)
3. Set the inputs (X0 to X3) to ON or OFF in the master station, and check that the outputs (Y0 to Y3) turn ON or OFF in each local station.
4. Operate the inputs (X0 to X3) of each local station to check that the master station output (Y0 to Y3) or the local station output (Y4 to Y7, Y10 to Y17...Y34 to Y37) turns ON.

Test program for communication (master station)



Step for writing information from the master station (master station → local station)

The contents of X0 to X3 in the master station are transferred to outputs (Y) in each local station.

Steps for reading information from a local station (local station → master station)

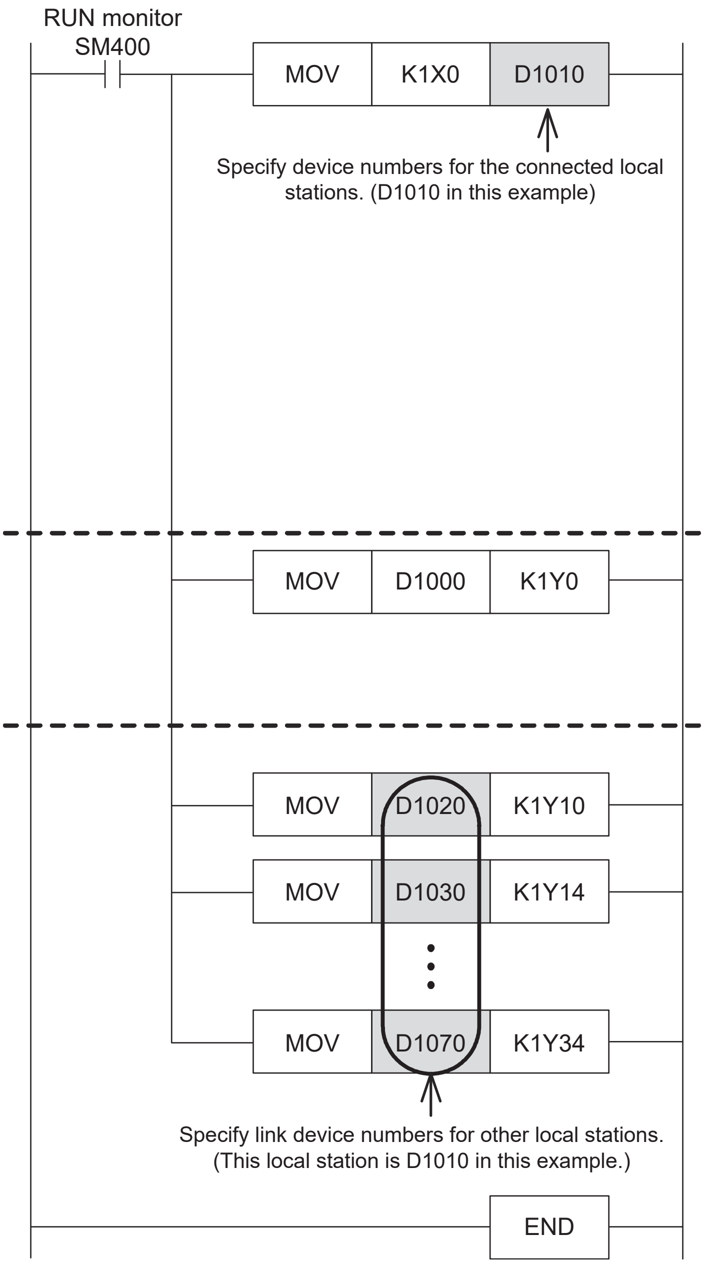
By using link devices, read information from all other local stations.
The table below shows the link devices to be used:

Local station No.	Link device	Output (Y)
1	D1010	Y4 to Y7
2	D1020	Y10 to Y13
3	D1030	Y14 to Y17
4	D1040	Y20 to Y23
5	D1050	Y24 to Y27
6	D1060	Y30 to Y33
7	D1070	Y34 to Y37

Test program for communication (local station)

Determine the station number of each local station, and then transfer a program corresponding to the station number to each local station.

Assign station numbers from "1" in the ascending order. (Use one station number only once. Do not skip station numbers.)



Specify device numbers for the connected local stations. (D1010 in this example)

Specify link device numbers for other local stations. (This local station is D1010 in this example.)

Steps for reading information from a local station (local station → master station)

Transfer the contents of X0 to X3 in this local station to a link device. The link device varies depending on the station number.
The table below shows the link devices to be used:

Local station No.	Link device
1	D1010
2	D1020
3	D1030
4	D1040
5	D1050
6	D1060
7	D1070

Step for writing information from the master station (master station → local station)

The contents of X0 to X3 in the master station are transferred to Y0 to Y3 in this local station.

Steps for reading information from another local station (another local station → this local station)

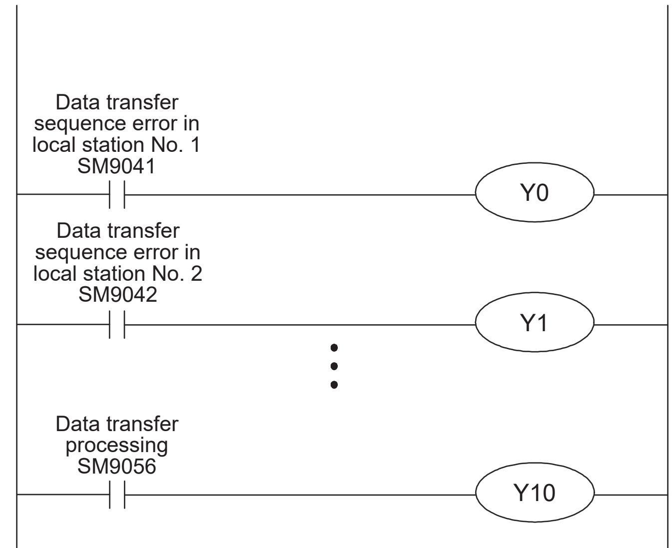
By using link devices, read information from another local station.
The table below shows the link devices to be used:

Local station No.	Link device	Output (Y)
1	D1010	Y4 to Y7
2	D1020	Y10 to Y13
3	D1030	Y14 to Y17
4	D1040	Y20 to Y23
5	D1050	Y24 to Y27
6	D1060	Y30 to Y33
7	D1070	Y34 to Y37

Creating programs for the master station

Create programs for the master station.

Arbitrarily create programs for reading and writing link devices.



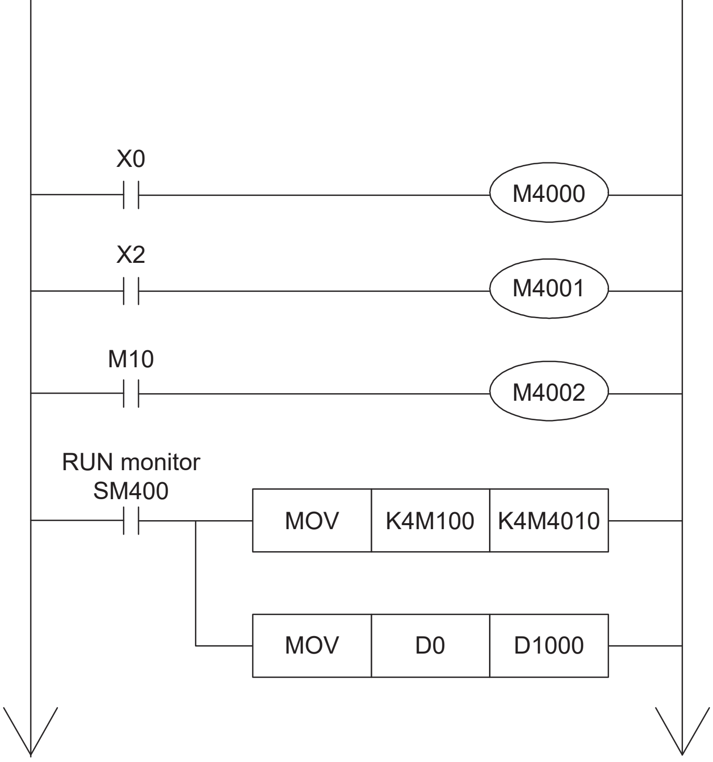
Program for indicating link errors

This program is required to indicate the N:N Network status. Create this step for each connected local station.

When a link error occurs in the station No. 1, Y0 is set to ON.

When a link error occurs in the station No. 2, Y1 is set to ON.

While the N:N Network is executed, Y10 remains ON.



Program for writing link devices (master station → local station)

This program is required to write information from the master station to each local station.

The information from X0 is written to M4000 (link device).

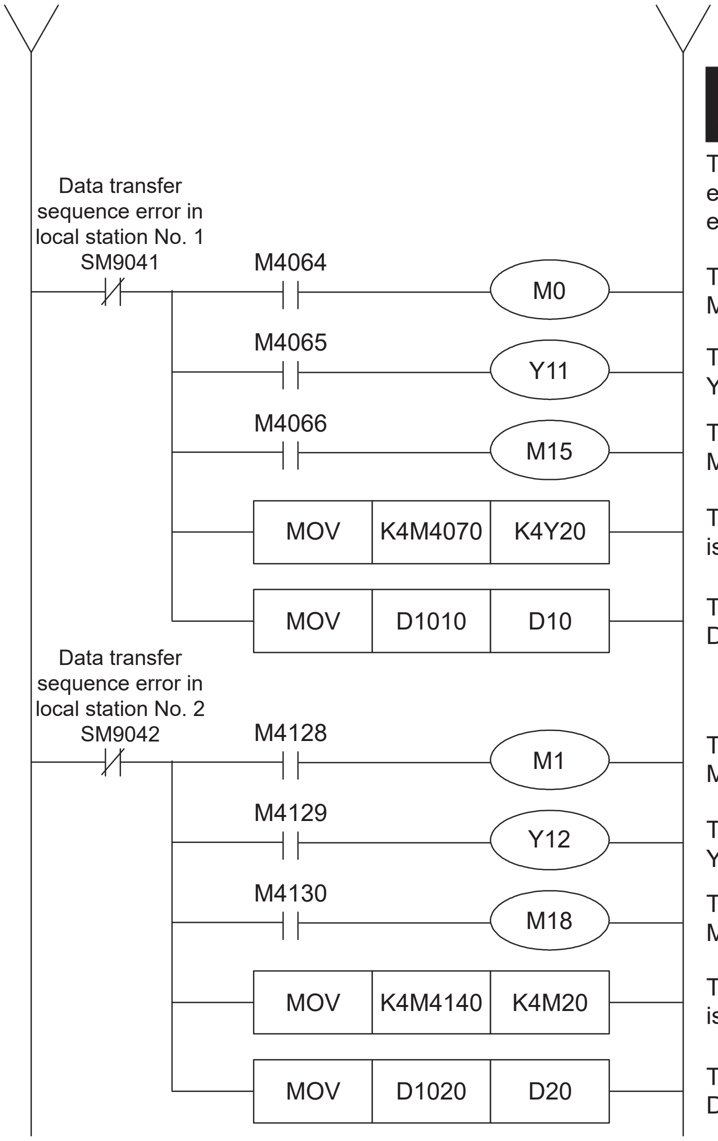
The information from X2 is written to M4001 (link device).

The information from M10 is written to M4002 (link device).

The information from M100 to M115 is written to M4010 to M4025 (link devices).

The information from D0 is written to D1000 (link device).

* In pattern 0, bit devices cannot be used. Use the word devices instead.



Program for reading link devices (local station → master station)

This program is required to read information from each local station to the master station. Monitor link errors in each local station, and read them.

The information from M4064 (link device) is read to M0.

The information from M4065 (link device) is read to Y11.

The information from M4066 (link device) is read to M15.

The information from M4070 to M4085 (link devices) is read to Y20 to Y37.

The information from D1010 (link device) is read to D10.

The information from M4128 (link device) is read to M1.

The information from M4129 (link device) is read to Y12.

The information from M4130 (link device) is read to M18.

The information from M4140 to M4155 (link devices) is read to M20 to M35.

The information from D1020 (link device) is read to D20.

Refer to the above for creating the error monitoring program of station number 3 or later.

For details on the link devices, refer to ☞ Page 265 Contents of related devices.

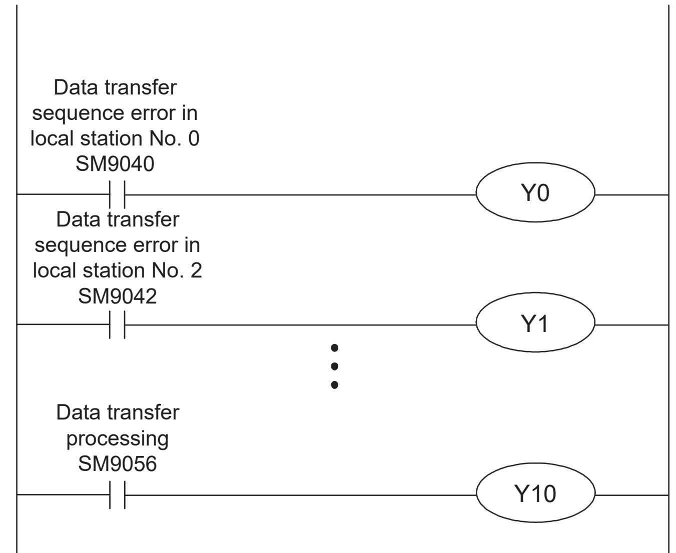
For cautions on program creation, refer to ☞ Page 272 Cautions on program creation.

Creating programs for the local stations

Create programs for the local stations.

Arbitrarily create programs for reading and writing link devices.

The following program is for the local station of station number 1. Program for station number 2 and later are similar.



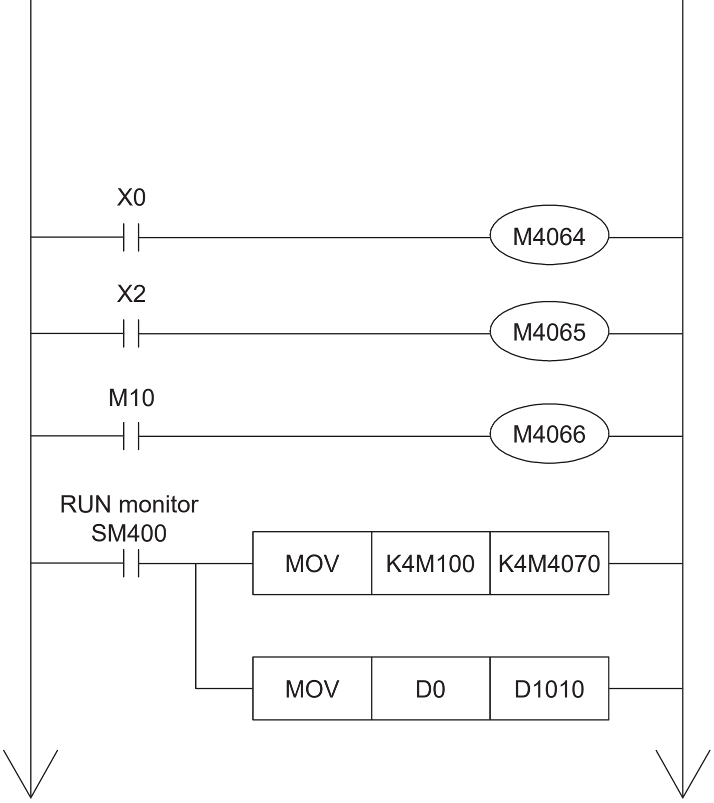
Program for indicating link errors

This program is required to indicate the N:N Network status. Create this step for each connected local station.

When a link error occurs in the station No. 0, Y0 is set to ON.

When a link error occurs in the station No. 2, Y1 is set to ON.

While the N:N Network is executed, Y10 remains ON.



Program for writing link devices (local station → master or local station)

This program is required to write information from a local station to the master station or another local station.

The information from X0 is written to M4064 (link device).

The information from X2 is written to M4065 (link device).

The information from M10 is written to M4066 (link device).

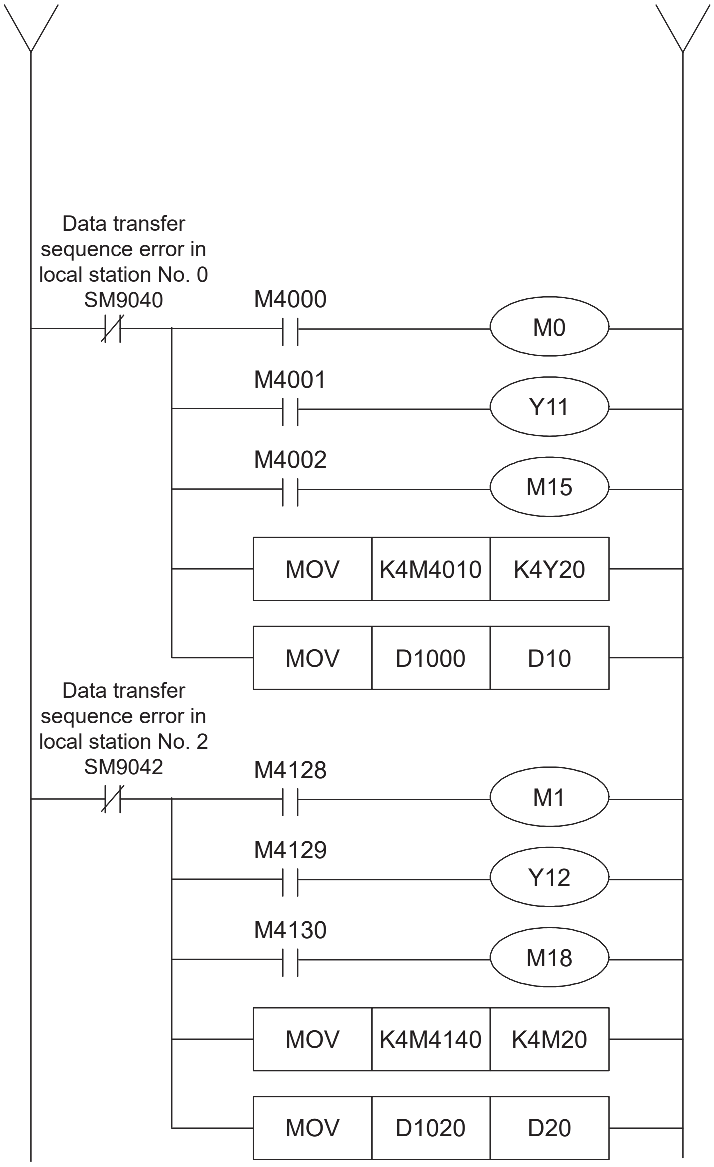
The information from M100 to M115 is written to M4070 to M4085 (link devices).

The information from D0 is written to D1010 (link device).

* In pattern 0, bit devices cannot be used. Use the word devices instead.

Program for reading link devices (local station ← master or local station)

This program is required to read information from the master station or another local station to a local station.
Monitor link errors in each local station, and read them.



The information from M4000 (link device) is read to M0.

The information from M4001 (link device) is read to Y11.

The information from M4002 (link device) is read to M15.

The information from M4010 to M4025 (link devices) is read to Y20 to Y37.

The information from D1000 (link device) is read to D10.

The information from M4128 (link device) is read to M1.

The information from M4129 (link device) is read to Y012.

The information from M4130 (link device) is read to M18.

The information from M4140 to M4155 (link devices) is read to M20 to M35.

The information from D1020 (link device) is read to D20.

For details on the link devices, refer to ☞ Page 265 Contents of related devices.

For cautions on program creation, refer to ☞ Page 272 Cautions on program creation.

Cautions on program creation

- When N:N Network is used, the operation cycle in each PLC becomes longer by about 10% per station.
- Set station numbers consecutively. If a station number that is used twice or more or is skipped, link will not be achieved normally.
- Do not change the contents of link devices at other stations. When a link error (data transfer sequence error) occurs, the link device information is held in the status just before occurrence of the link error. Create program to operate safely when an error occurs.
- Link device update in N:N Network is asynchronous to operations of the sequence program. However, the update processing of the sent data from the local stations are executed by the END process of each station. Therefore, the change of link device value during the ladder operation is not reflected on the communication data until the END process starts.
- Precaution when changing the station number and number of local stations

Only when the station number and number of local stations are set with latch setting can the setting values be written to the corresponding word device from a program or an engineering tool and the setting be changed by powering OFF→ON or a reset. However, be careful not to exceed the range upper limit of device for link device when you set the device head number. When the device range upper limit is exceeded, the operation is as follows.

(1) When the total number of local stations is increased and the master station link device is insufficient

The error code "7705H" is stored in SD8211, SD9061. However, link continues, but it does not update for the device above the range upper limit.

(2) When the total number of local stations is increased and the local station link device is insufficient

The error code "7705H" is stored in SD of the local station of SD8212 to SD8218 and SD9062 to SD9068. However, link continues, but it does not update for the device above the range upper limit.

(3) When the link device of the local station is out of the device range by changing the local station number

The error code "7715H" is stored in SD of the local station of SD8212 to SD8218 and SD9062 to SD9068. However, devices above the range upper limit send all 0 (word: 0, bit: OFF). Also, when the error (7705H) of (2) occurs at the same time, 7715H is stored.

16.8 Related Devices

This section describes the special relay/special register functions used in the N:N Network function.

Point

Available communication channels vary depending on the CPU module and system configuration.
For communication channels, refer to ☞ Page 251 System Configuration.
"FX3 Series compatible" devices operate only on the communication channel specified in the compatible SM/SD for communication settings.
For compatible SM/SD, refer to ☞ Page 262 Communication Setting.

List of related devices

Special relay

■FX5 only

R: Read only, M: Master station (station number 0), L: Local station (station number 1 to 7)

Device No.	Name	Descriptions	Detection	R/W	Available station	
					Master station (Station No.0)	Local stations
SM8500	Serial communication error (CH1)	Turns ON when serial communication error of CH1 occurs.	M, L	R	○	○ (Station 1 to 7)
SM8510	Serial communication error (CH2)	Turns ON when serial communication error of CH2 occurs.	M, L	R	○	○ (Station 1 to 7)
SM8520	Serial communication error (CH3)	Turns ON when serial communication error of CH3 occurs.	M, L	R	○	○ (Station 1 to 7)
SM8530	Serial communication error (CH4)	Turns ON when serial communication error of CH4 occurs.	M, L	R	○	○ (Station 1 to 7)
SM9040	Data communication error (Master station)	Turns ON when data transfer sequence error of master station occurs.	L	R	×	○ (Station 1 to 7)
SM9041	Data transfer sequence error (Local station 1)	Turns ON when data transfer sequence error of local station occurs However, data sequence errors that occurred in the host station (local station) cannot be detected.	M, L	R	○	○ (Except station 1)
SM9042	Data transfer sequence error (Local station 2)					○ (Except station 2)
SM9043	Data transfer sequence error (Local station 3)					○ (Except station 3)
SM9044	Data transfer sequence error (Local station 4)					○ (Except station 4)
SM9045	Data transfer sequence error (Local station 5)					○ (Except station 5)
SM9046	Data transfer sequence error (Local station 6)					○ (Except station 6)
SM9047	Data transfer sequence error (Local station 7)					○ (Except station 7)
SM9056	Data transfer sequence ON	Remains ON while data transfer is being executed.	M, L	R	○	○ (Station 1 to 7)
SM9080	Station number settings SD latch setting valid information	Turns ON when SD latch setting is set for the station number.	M, L	R	○	○ (Station 1 to 7)

Device No.	Name	Descriptions	Detection	R/W	Available station	
					Master station (Station No.0)	Local stations
SM9081	Local station quantity setting SD latch setting valid information	Turns ON when SD latch setting is set for the number of local stations setting.	M, L	R	○	×

■FX3 Series compatible

R: Read only, M: Master station (station number 0), L: Local station (station number 1 to 7)

Device No.	Name	Descriptions	Detection	R/W	Available station	
					Master station (Station No.0)	Local stations
SM8063	Serial communication error (CH1)	Turns ON when error occurs in the serial communication with the SM/SD compatible with FX3 Series being CH1.	M, L	R	○	○ (Station 1 to 7)
SM8438	Serial communication error (CH2)	Turns ON when error occurred in serial communication with the SM/SD compatible with FX3 Series being CH2.	M, L	R	○	○ (Station 1 to 7)
SM8183	Data communication error (Master station)	Turns ON when data transfer sequence error occurs in the master station.	L	R	×	○ (Station 1 to 7)
SM8184	Data transfer sequence error (Local station 1)	Turns ON when a data transfer sequence error occurs in each local station. However, data sequence errors that occurred in the host station (local station) cannot be detected.	M, L	R	○	○ (Except station 1)
SM8185	Data transfer sequence error (Local station 2)					○ (Except station 2)
SM8186	Data transfer sequence error (Local station 3)					○ (Except station 3)
SM8187	Data transfer sequence error (Local station 4)					○ (Except station 4)
SM8188	Data transfer sequence error (Local station 5)					○ (Except station 5)
SM8189	Data transfer sequence error (Local station 6)					○ (Except station 6)
SM8190	Data transfer sequence error (Local station 7)					○ (Except station 7)
SM8191	Data transfer sequence ON	Remains ON while data transfer is being executed.	M, L	R	○	○ (Station 1 to 7)

Special register

■FX5 only

R: Read only, M: Master station (station number 0), L: Local station (station number 1 to 7)

Device No.	Name	Descriptions	Detection	R/W	Available station	
					Master station (Station No.0)	Local stations
SD9040	Station number settings status	Provided to check the station number.	M, L	R	○	○ (Station 1 to 7)
SD9041	Number of local stations setting status	Provided to check the number of local stations.	M, L	R	○	○ (Station 1 to 7)
SD8500	Serial communication error code (CH1)	When an error occurs in the serial communication of CH1, the error code is stored.	M, L	R	○	○ (Station 1 to 7)
SD8502	Serial communication settings (CH1)	Stores the set communication parameter in the PLC of CH1.	M, L	R	○	○ (Station 1 to 7)
SD8503	Serial communication operation mode (CH1)	Stores the communication function that serial communication of CH1 being used.	M, L	R	○	○ (Station 1 to 7)
SD8510	Serial communication error code (CH2)	Stores the error code when an error occurs in the serial communication of CH2.	M, L	R	○	○ (Station 1 to 7)
SD8512	Serial communication settings (CH2)	Stores the set communication parameter in the PLC of CH2.	M, L	R	○	○ (Station 1 to 7)
SD8513	Serial communication operation mode (CH2)	Stores the communication function that serial communication of CH2 being used.	M, L	R	○	○ (Station 1 to 7)
SD8520	Serial communication error code (CH3)	Stores the error code when an error occurs in the serial communication of CH3.	M, L	R	○	○ (Station 1 to 7)
SD8522	Serial communication settings (CH3)	Stores the set communication parameter in the PLC of CH3.	M, L	R	○	○ (Station 1 to 7)
SD8523	Serial communication operation mode (CH3)	Stores the communication function that serial communication of CH3 being used.	M, L	R	○	○ (Station 1 to 7)
SD8530	Serial communication error code (CH4)	Stores the error code when an error occurs in the serial communication of CH4.	M, L	R	○	○ (Station 1 to 7)
SD8532	Serial communication settings (CH4)	Stores the set communication parameter in the PLC of CH4.	M, L	R	○	○ (Station 1 to 7)
SD8533	Serial communication operation mode (CH4)	Stores the communication function that serial communication of CH4 being used.	M, L	R	○	○ (Station 1 to 7)
SD8744	Message waiting time (CH1)	Stores the message waiting time (CH1).	M, L	R	○	○ (Station 1 to 7)
SD8754	Message waiting time (CH2)	Stores the message waiting time (CH2).	M, L	R	○	○ (Station 1 to 7)
SD8764	Message waiting time (CH3)	Stores the message waiting time (CH3).	M, L	R	○	○ (Station 1 to 7)
SD8774	Message waiting time (CH4)	Stores the message waiting time (CH4).	M, L	R	○	○ (Station 1 to 7)
SD9043	Present link scan time	Current value of the network cycle time	M, L	R	○	○ (Station 1 to 7)
SD9044	Maximum link scan time	Maximum value of the network cycle time	M, L	R	○	○ (Station 1 to 7)
SD9045	Number of communication error at master station	Number of data transfer sequence errors occurred in the master station.	L	R	×	○ (Station 1 to 7)

Device No.	Name	Descriptions	Detection	R/W	Available station	
					Master station (Station No.0)	Local stations
SD9046	Number of communication error at local station 1	Number of data transfer sequence errors occurred in other local station. However, data sequence errors that occurred in the host station (local station) cannot be detected.	M, L	R	○	○ (Except station 1)
SD9047	Number of communication error at local station 2					○ (Except station 2)
SD9048	Number of communication error at local station 3					○ (Except station 3)
SD9049	Number of communication error at local station 4					○ (Except station 4)
SD9050	Number of communication error at local station 5					○ (Except station 5)
SD9051	Number of communication error at local station 6					○ (Except station 6)
SD9052	Number of communication error at local station 7					○ (Except station 7)
SD9061	Code of communication error at master station	Stores the error code for the master station.	L	R	×	○ (Station 1 to 7)
SD9062	Data transmission error code (local station 1)	Number of data transfer sequence errors occurred in other local station. However, data sequence errors that occurred in the host station (local station) cannot be detected.	M, L	R	○	○ (Except station 1)
SD9063	Data transmission error code (local station 2)					○ (Except station 2)
SD9064	Data transmission error code (local station 3)					○ (Except station 3)
SD9065	Data transmission error code (local station 4)					○ (Except station 4)
SD9066	Data transmission error code (local station 5)					○ (Except station 5)
SD9067	Data transmission error code (local station 6)					○ (Except station 6)
SD9068	Data transmission error code (local station 7)					○ (Except station 7)
SD9080[*1]	Station number settings	Set the local station number.	M, L	[*2]	○	○ (Station 1 to 7)
SD9081[*1]	Number of local stations setting	Set the number of connected local stations.	M, L	[*2]	○	○ (Station 1 to 7)
SD9082	Refresh range setting status	Refresh range check (same description as SD8175)	M, L	R	○	○ (Station 1 to 7)
SD9083	Number of retries setting status	Check of the number of communication retry times that occur (same description as SD8179)	M, L	R	○	○ (Station 1 to 7)
SD9084	Monitoring time setting status	Monitoring time check However, the local station stores the doubled value of the setting value of the master station.	M, L	R	○	○ (Station 1 to 7)

[*1] When latch setting is set to "latch", change the device value with a program or an engineering tool and powering ON→OFF or reset to operate the corresponding functions with the changed value.

[*2] With latch settings: Read/write
No latch settings: Read only

■FX3 Series compatible

R: Read only, M: Master station (station number 0), L: Local station (station number 1 to 7)

Device No.	Name	Descriptions	Detection	R/W	Available station	
					Master station (Station No.0)	Local stations
SD8173	Station number settings status	Provided to check the station number.	M, L	R	○	○ (Station 1 to 7)
SD8174	Number of local stations setting status	Provided to check the number of local stations.	M, L	R	○	○ (Station 1 to 7)
SD8175	Refresh range setting status	Provided to check the refresh range.	M, L	R	○	○ (Station 1 to 7)
SD8063	Serial communication error code (CH1)	Stores the error code when error occurs in the serial communication with the SM/SD compatible with FX3 Series being CH1.	M, L	R	○	○ (Station 1 to 7)
SD8405	Serial communication settings (CH1)	Stores the set communication parameter with the SM/SD compatible with FX3 Series being CH1.	M, L	R	○	○ (Station 1 to 7)
SD8419	Serial communication operation mode (CH1)	Stores the communication function of the serial communication operation with the SM/SD compatible with FX3 Series CH1.	M, L	R	○	○ (Station 1 to 7)
SD8438	Serial communication error code (CH2)	Stores the error code when error occurs in the serial communication with the SM/SD compatible with FX3 Series being CH2.	M, L	R	○	○ (Station 1 to 7)
SD8425	Serial communication settings (CH2)	Stores the set communication parameter with the SM/SD compatible with FX3 Series being CH2.	M, L	R	○	○ (Station 1 to 7)
SD8439	Serial communication operation mode (CH2)	Store the communication function of the serial communication operation with the SM/SD compatible with FX3 Series in CH2.	M, L	R	○	○ (Station 1 to 7)
SD8201	Present link scan time	Current value of the network cycle time	M, L	R	○	○ (Station 1 to 7)
SD8202	Maximum link scan time	Maximum value of the network cycle time	M, L	R	○	○ (Station 1 to 7)
SD8203	Number of communication error at master station	Number of data transfer sequence errors occurred in the master station.	L	R	×	○ (Station 1 to 7)
SD8204	Number of communication error at local station 1	Number of data transfer sequence errors occurred in other local station. However, data sequence errors that occurred in the host station (local station) cannot be detected.	M, L	R	○	○ (Except station 1)
SD8205	Number of communication error at local station 2					○ (Except station 2)
SD8206	Number of communication error at local station 3					○ (Except station 3)
SD8207	Number of communication error at local station 4					○ (Except station 4)
SD8208	Number of communication error at local station 5					○ (Except station 5)
SD8209	Number of communication error at local station 6					○ (Except station 6)
SD8210	Number of communication error at local station 7					○ (Except station 7)
SD8211	Code of communication error at master station	Stores the error code for the master station.	L	R	×	○ (Station 1 to 7)

Device No.	Name	Descriptions	Detection	R/W	Available station	
					Master station (Station No.0)	Local stations
SD8212	Data transmission error code (local station 1)	Stores the data transfer error code occurred in other local station. However, data sequence errors that occurred in the host station (local station) cannot be detected.	M, L	R	○	○ (Except station 1)
SD8213	Data transmission error code (local station 2)					○ (Except station 2)
SD8214	Data transmission error code (local station 3)					○ (Except station 3)
SD8215	Data transmission error code (local station 4)					○ (Except station 4)
SD8216	Data transmission error code (local station 5)					○ (Except station 5)
SD8217	Data transmission error code (local station 6)					○ (Except station 6)
SD8218	Data transmission error code (local station 7)					○ (Except station 7)

Details of related devices

Serial communication error

Turns ON when an error occurs in serial communication.

R: Read only

FX5 only				FX3 Series compatible		Descriptions	R/W
CH1	CH2	CH3	CH4	CH1	CH2		
SM8500	SM8510	SM8520	SM8530	SM8063	SM8438	Turns ON when an error occurs in serial communication.	R

After the devices above turns ON, the error code is stored in the compatible devices below.

FX5 only				FX3 Series compatible		Name	Descriptions
CH1	CH2	CH3	CH4	CH1	CH2		
SD8500	SD8510	SD8520	SD8530	SD8063	SD8438	Serial communication error code	When a serial communication error occurs, the error code is stored.

Precautions

Do not turn ON or OFF with program or engineering tool.

Serial communication error does not turn OFF even if the communication comes back to normal. The devices turn OFF when power is turned OFF→ON, STOP→RUN, reset or SM50 (Error Detection Reset Completion) is turned ON.

Data transfer sequence error

Turns ON when a data transfer sequence error occurs in the master station or local station.

R: Read only

Station number	Master station	Local stations							R/W
	Station No. 0	Station No. 1	Station No. 2	Station No. 3	Station No. 4	Station No. 5	Station No. 6	Station No. 7	
FX5 only	SM9040	SM9041	SM9042	SM9043	SM9044	SM9045	SM9046	SM9047	R
FX3 Series compatible	SM8183	SM8184	SM8185	SM8186	SM8187	SM8188	SM8189	SM8190	

After the devices above turn ON, the error code is stored in the corresponding devices below (data transfer sequence error).

Station number	Master station	Local stations						
	Station No. 0	Station No. 1	Station No. 2	Station No. 3	Station No. 4	Station No. 5	Station No. 6	Station No. 7
FX5 only	SD9061	SD9062	SD9063	SD9064	SD9065	SD9066	SD9067	SD9068
FX3 Series compatible	SD8211	SD8212	SD8213	SD8214	SD8215	SD8216	SD8217	SD8218

Precautions

Data transfer sequence errors in a station itself cannot be detected.

Do not turn ON with program or engineering tool.

Data transfer sequence ON

Turns ON the data send operation with the master station and also other local stations.

R: Read only

FX5 only	FX3 Series compatible	Descriptions	R/W
SM9056	SM8191	ON: Data transfer sequence ON OFF: Data transfer sequence is stopped	R

Precautions

Do not turn ON with program or engineering tool.

Serial communication error code

This device stores the error code for serial communication (☞ Page 833 Checking absence/presence of N:N Network function errors).

R: Read only

FX5 only				FX3 Series compatible		Descriptions	R/W
CH1	CH2	CH3	CH4	CH1	CH2		
SD8500	SD8510	SD8520	SD8530	SD8063	SD8438	Stores the error code when the serial communication error occurs.	R

Precautions

Do not change the value with program or engineering tool.

Serial communication errors are not cleared even after communication has recovered its normal status.

Devices are cleared when power is turned OFF→ON, STOP→RUN, reset or SM50 (Error Detection Reset Completion) is turned ON.

Serial communication settings

Stores the communication parameters set in the communication settings (☞ Page 262 Communication Setting).

R: Read only

FX5 only				FX3 Series compatible		Descriptions	R/W
CH1	CH2	CH3	CH4	CH1	CH2		
SD8502	SD8512	SD8522	SD8532	SD8405	SD8425	Stores the setting of the communication parameter.	R

The description of the communication parameter is as follows.

Bit No.	Name	Descriptions	
		0 (bit is OFF)	1 (bit is ON)
b0	Data length	7 bits	—
b1 b2	Parity bit	b2, b1 (1, 1): Even	
b3	Stop bit	1 bit	—
b4 b5 b6 b7	Baud rate	b7, b6, b5, b4 (1, 0, 1, 0): 38400bps	
b12	H/W type	—	RS-485

Precautions

Do not change the value with program or engineering tool.

Station number settings status

Stores the station number set in the communication settings (☞ Page 262 Communication Setting). This device is for checking the setting status of station number of the local station.

R: Read only

FX5 only	FX3 Series compatible	Descriptions	R/W
SD9040	SD8173	0: Master station 1 to 7: Local stations	R

- Master station

Reflects the parameter setting value to each device.

- Local stations

Reflects the master station setting value to each device.

Precautions

Do not change the value with program or engineering tool.

Station number settings SD latch setting valid information

Turns ON when SD latch setting is set for the station number settings in the communication settings (☞ Page 262 Communication Setting).

R: Read only

FX5 only	Descriptions	R/W
SM9080	ON: Station number setting is latched OFF: Station number setting is not latched.	R

Precautions

Do not turn ON with program or engineering tool.

Station number settings

Stores the station number set in the communication settings (☞ Page 262 Communication Setting). The stored value is the same as station number settings status (SD9040, SD8173). When the latch setting is set, the station number can be changed with the program.

R: Read only

FX5 only	Descriptions	R/W
SD9080	0: Master station 1 to 7: Local stations	*1

*1 With latch settings: Read/write
No latch settings: Read only

- Master station

The parameter setting value is also reflected to the station number settings status (SD9040, SD8173).

- Local stations

The parameter setting value is also reflected to the station number settings status (SD9040, SD8173).

When the latch setting is set to SD9080, value can be changed with a program or an engineering tool. The values can be changed by powering OFF→ON or reset.

Precautions

When the latch setting is not set, do not change the value with program or engineering tool.

Number of local stations setting status

Stores the number of local stations set in the communication settings (☞ Page 262 Communication Setting). Use this device to check the number of local stations set in the master station.

R: Read only

FX5 only	FX3 Series compatible	Descriptions	R/W
SD9041	SD8174	1 to 7	R

- Master station

Reflects the parameter setting value to each device.

- Local stations

Reflects the master station setting value to each device.

Precautions

Do not change the value with program or engineering tool.

Number of local stations setting SD latch setting valid information

Turns ON when SD latch setting is set for the total number of local station settings in the communication settings (☞ Page 262 Communication Setting).

R: Read only

FX5 only	Descriptions	R/W
SM9081	ON: Number of local stations settings is latched OFF: Number of local stations settings is not latched	R

Precautions

Do not turn ON with program or engineering tool.

Number of local stations setting

Stores the number of local stations set in the communication settings (☞ Page 262 Communication Setting). The stored value is the same as number of local stations setting status (SD9041, SD8174). When there is latch setting, it is possible to change the number of local stations from the program.

R: Read only

FX5 only	Descriptions	R/W
SD9081	1 to 7	[1]

[1] With latch settings: Read/write
No latch settings: Read only

- Master station

The parameter setting value is also reflected to the number of local stations setting status. (SD9041, SD8174).

When SD9081 latch setting is set, it is possible to change the setting value by a program or engineering tool and powering OFF→ON or reset.

- Local stations

Reflects the master station setting value to each device.

Precautions

When the latch setting is not set, do not change the value with program or engineering tool.

Refresh range setting status

Stores the refresh range set in the communication settings (☞ Page 262 Communication Setting). Use this device to check the refresh range set in the master station.

R: Read only

FX5 only	FX3 Series compatible	Descriptions	R/W
SD9082	SD8175	0: Pattern 0 1: Pattern 1 2: Pattern 2	R

- Master station

Reflects the parameter setting value on each device.

- Local stations

Reflects the master station setting value on each device.

Precautions

Do not change the value with program or engineering tool.

Number of retries times setting

Stores the number of retries set in the communication settings (☞ Page 262 Communication Setting). Stores the setting value of the number of retries.

R: Read only

FX5 only	Descriptions	R/W
SD9083	0 to 10 (times)	R

- Master station

Reflects the parameter setting value on each device.

- Local stations

Reflects the master station setting value on each device.

Precautions

Do not change the value with program or engineering tool.

Monitoring time setting

Stores the monitoring time set in the communication settings (☞ Page 262 Communication Setting).

R: Read only

FX5 only	Descriptions	R/W
SD9084	50 to 2550 (ms)	R

- Master station

Reflects the parameter setting value on each device.

- Local stations

Reflects the master station setting value on each device.

Precautions

Do not change the value with program or engineering tool.

Present link scan time

This device stores the current value of the network cycle in the N:N Network.

R: Read only

FX5 only	FX3 Series compatible	Descriptions	R/W
SD9043	SD8201	0 to 32767 (×10ms)	R

Precautions

Do not change the value with program or engineering tool.

Maximum link scan time

This device stores the maximum value of the network cycle in the N:N Network.

R: Read only

FX5 only	FX3 Series compatible	Descriptions	R/W
SD9044	SD8202	0 to 32767 (×10ms)	R

Precautions

Do not change the value with program or engineering tool.

Data transfer sequence error count

These devices store the number of data transfer sequence errors that occurred in the master station and local stations.

R: Read only

Station number	Master station	Local stations							R/W
	Station No. 0	Station No. 1	Station No. 2	Station No. 3	Station No. 4	Station No. 5	Station No. 6	Station No. 7	
FX5 only	SD9045	SD9046	SD9047	SD9048	SD9049	SD9050	SD9051	SD9052	R
FX3 Series compatible	SD8203	SD8204	SD8205	SD8206	SD8207	SD8208	SD8209	SD8210	

Precautions

Data transfer sequence errors in a station itself cannot be detected.

Do not change the value with program or engineering tool.

Data transfer sequence error count does not turn OFF even if the communication comes back to normal. Devices are cleared when power is turned OFF→ON, STOP→RUN, reset or SM50 (Error Detection Reset Completion) is turned ON.

Data transmission error code

These devices store the error code (☞ Page 833 Checking absence/presence of N:N Network function errors) of the master station or the local stations.

R: Read only

Station number	Master station	Local stations							R/W
	Station No. 0	Station No. 1	Station No. 2	Station No. 3	Station No. 4	Station No. 5	Station No. 6	Station No. 7	
FX5 only	SD9061	SD9062	SD9063	SD9064	SD9065	SD9066	SD9067	SD9068	R
FX3 Series compatible	SD8211	SD8212	SD8213	SD8214	SD8215	SD8216	SD8217	SD8218	

Precautions

Data transfer sequence errors in a station itself cannot be detected.

Do not change the value with program or engineering tool.

Data transmission error code does not turn OFF even if the communication comes back to normal. The devices turn OFF when power is turned OFF→ON, STOP→RUN, reset or SM50 (Error Detection Reset Completion) is turned ON.

Serial communication operation mode

Stores the communication function code that serial communication is being used.

R: Read only

FX5 only				FX3 Series compatible		Descriptions	R/W
CH1	CH2	CH3	CH4	CH1	CH2		
SD8503	SD8513	SD8523	SD8533	SD8419	SD8439	0: MELSOFT connection or MC protocol 3: N:N Network Communication 5: Non-Protocol Communication 6: Parallel Link Communication 7: Inverter Communication 9: MODBUS RTU Communication 12: Predefined protocol support Other than above: Not used	R

Precautions

Do not change the value with program or engineering tool.

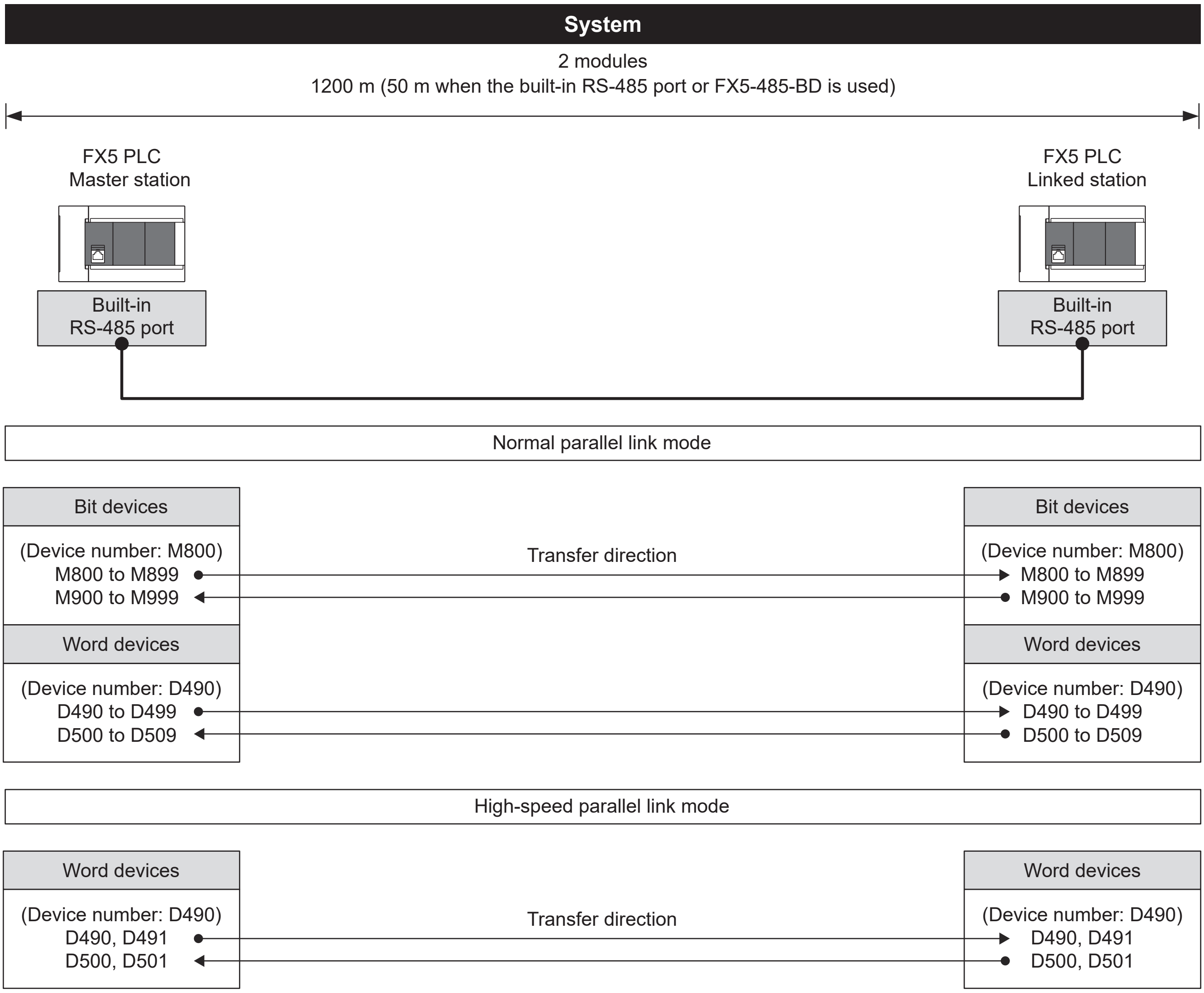
17 PARALLEL LINK FUNCTION

This chapter explains parallel link.

17.1 Function Outline

The parallel link function is designed to connect two FX5 PLCs and mutually link the devices.

- The link mode can be selected from two modes, normal parallel link and high-speed parallel link modes, depending on the number of devices to be linked and link time.
- Data link is automatically updated between two FX5 PLCs.
- The total extension distance is 1200 m maximum. (only when the FX5-485ADP is used in the configuration)
- The head device number can be set for each of the internal relay (M) and data register (D) for link.



The device numbers for link are defaults.

17.2 Procedure Before Operation

The flow chart below shows the procedures from parallel link function setting to data link:

1. Check communication specifications

For the communication specifications, link specifications, link time, link device numbers and number of link points, refer to ☞ Page 290 Specifications.

2. System configuration and selection

For the system configuration and selection of communication devices, refer to ☞ Page 287 System Configuration.

3. Wiring

For the wiring with twisted pair cables and wiring examples, refer to ☞ Page 292 Wiring.

4. Communication setting[*1]

For the communication settings of communication devices, refer to ☞ Page 296 Communication Setting.

5. Program creation

For the communication test program, master station program, and linked station program, refer to ☞ Page 298 Programming.

[*1] For details on operating procedures of GX Works3, refer to the manual below.
📖 GX Works3 Operating Manual

17.3 System Configuration

This section outlines the system configuration required to use the parallel link function.

Restriction

In the parallel link, only one channel is available to use for one CPU module.

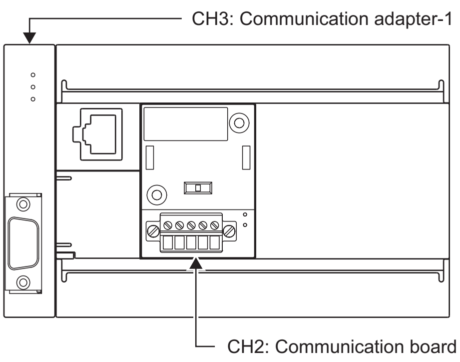
FX5S CPU module

The parallel link function can be used in the FX5S CPU module by using the communication board and communication adapter.

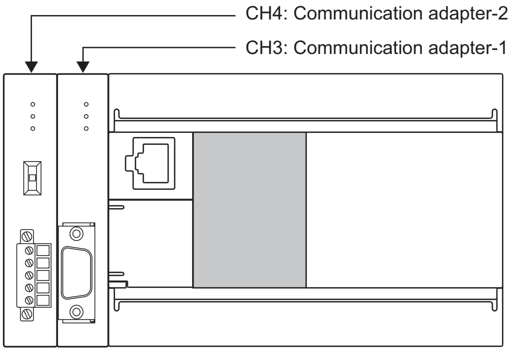
Communication channel assignments are fixed regardless of the system configuration.

The combinations available for the system configurations are shown below.

[Configuration example 1]



[Configuration example 2]



Item		Serial port	Important points in selection	Overall distance
Communication board	FX5-485-BD	CH2	Because the board can be mounted on top of the CPU module, there is no change in the installation space to be required.	50 m or less
Communication adapter	FX5-485ADP	CH3, CH4[*1]	Mount the communication adapter to the left of the CPU module.	1200 m or less

[*1] The adapters are assigned to CH3 and CH4 in the order, from the closest one to the CPU module.

Precautions

Total number of communication boards and communication adapters that can be connected is two.

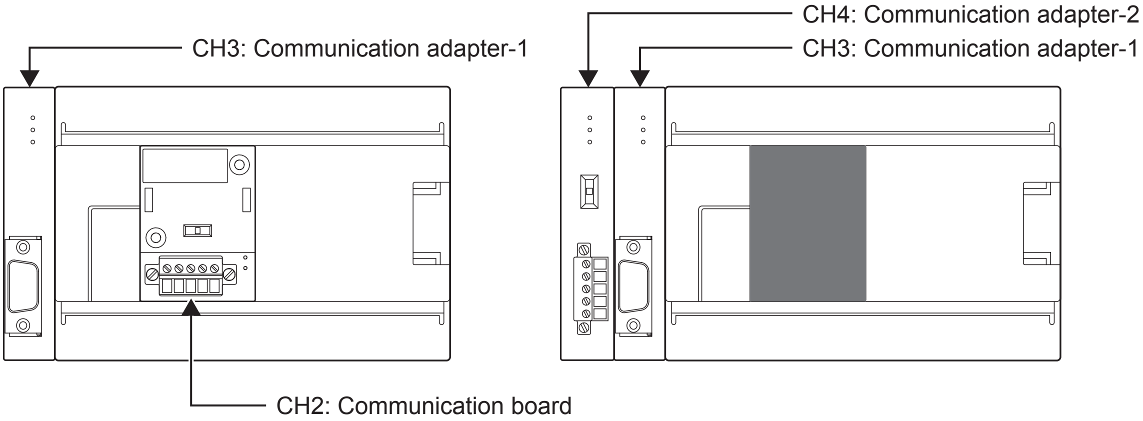
FX5UJ CPU module

The parallel link function can be used in the FX5UJ CPU module by using the communication board and communication adapter.

Communication channel assignments are fixed regardless of the system configuration.

The combinations which can be configured are shown below.

[Configuration example 1] [Configuration example 2]



Item		Serial port	Important points in selection	Overall distance
Communication board	FX5-485-BD	CH2	Mounted on top of the CPU module, there is no change in the installation space requirements.	50 m or less
Communication adapter	FX5-485ADP	CH3, CH4[1]	Mount the communication adapter to the left of the CPU module.	1200 m or less

[1] The adapters are assigned to CH3 and CH4 in the order, from the closest one to the CPU module.

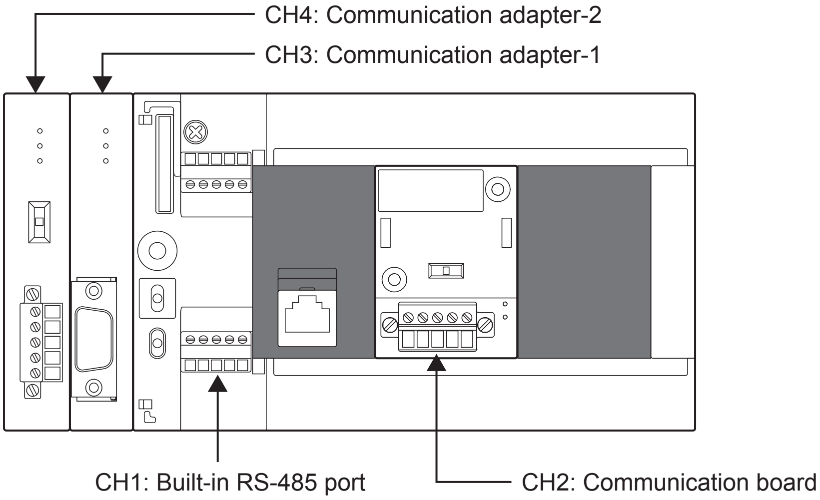
Precautions

Total number of communication boards and communication adapters that can be connected is two.

FX5U CPU module

The parallel link function can be used in the FX5U CPU module by using the built-in RS-485 port, communication board, and communication adapter.

Communication channel assignments are fixed regardless of the system configuration.



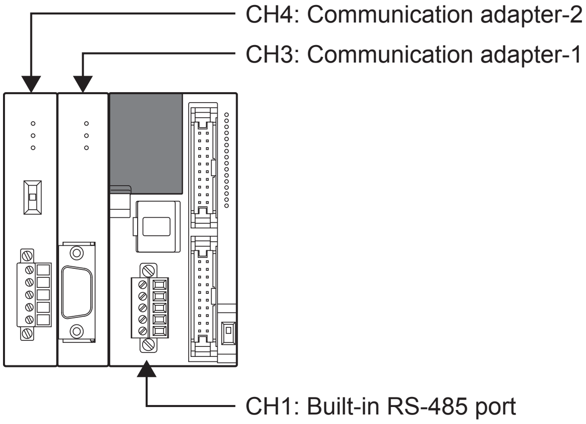
Item		Serial port	Important points for selection	Overall distance
Built-in RS-485 port		CH1	Since it is built-into the CPU module, there is no need to add equipment.	50 m or less
Communication board	FX5-485-BD	CH2	Mounted on top of the CPU module, there is no change in the installation space requirements.	50 m or less
Communication adapter	FX5-485ADP	CH3, CH4[*1]	Mount the communication adapter to the left of the CPU module.	1200 m or less

*1 The adapters are assigned to CH3 and CH4 in the order, from the closest one to the CPU module.

FX5UC CPU module

The parallel link function can be used in the FX5UC CPU module by using the built-in RS-485 port and communication adapter.

Communication channel assignments are fixed regardless of the system configuration.



Item		Serial port	Important points in selection	Overall distance
Built-in RS-485 port		CH1	Since it is built-into the CPU module, there is no need to add equipment.	50 m or less
Communication adapter	FX5-485ADP	CH3, CH4[*1]	Mount the communication adapter to the left of the CPU module.	1200 m or less

*1 The adapters are assigned to CH3 and CH4 in the order, from the closest one to the CPU module.

17.4 Specifications

This section describes the communication specifications and performance of the parallel link function.

Communication specifications

The parallel link function is executed according to the communication specifications (fixed) shown in the table below. Specification items such as baud rate cannot be changed.

Item		Specifications	Remarks
Number of connectable modules		Maximum of 2 (1: 1)	—
Transmission standard		RS-485 standard	—
Maximum overall distance		When only FX5-485ADP is used in the configuration: 1200 m or less Other than the above configuration: 50 m or less	When the built-in RS-485 port or FX5-485-BD is connected: 50 m or less
Protocol Type		Parallel link	—
Control procedure		—	—
Communication method		Half-duplex, bi-directional communication	—
Baud rate		115200 bps	—
Character format	Start bit	1 bit	—
	Data length	7 bits	—
	Parity bit	Even	—
	Stop bit	1 bit	—
Header		Fixed	—
Terminator		Fixed	—
Control line		—	—
Sum check		Fixed	—

Link specifications

Link time

The link time is the cycle time for updating of link device.

The link time varies depending on the link mode as shown in the table below.

Link mode	Time
Normal parallel link mode	15ms + operation cycle of master station (ms) + operation cycle of linked station (ms)
High-speed parallel link mode	5ms + operation cycle of master station (ms) + operation cycle of linked station (ms)

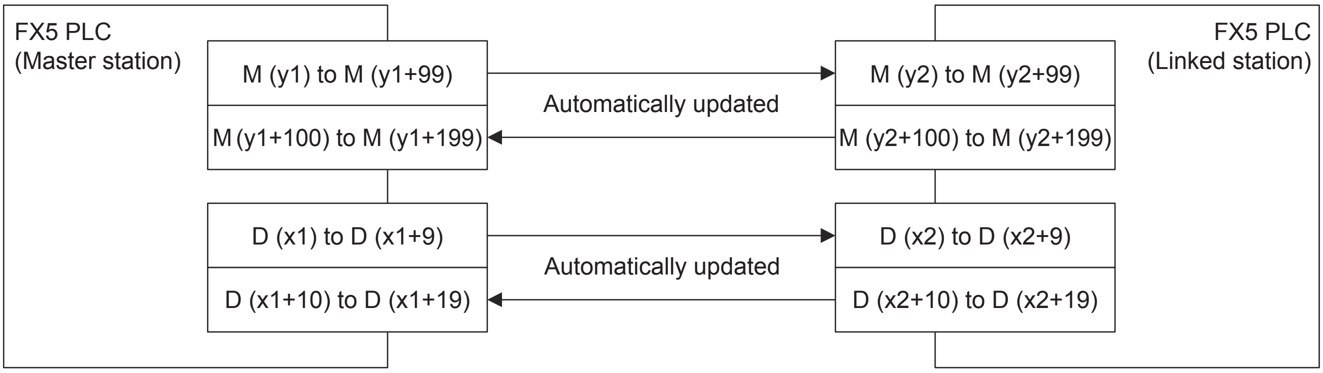
Link device number and number of link points

The devices to be occupied are assigned according to the head number of the link device set by GX Works3. Also the link mode is specified by GX Works3. (☞ Page 296 Communication Setting)

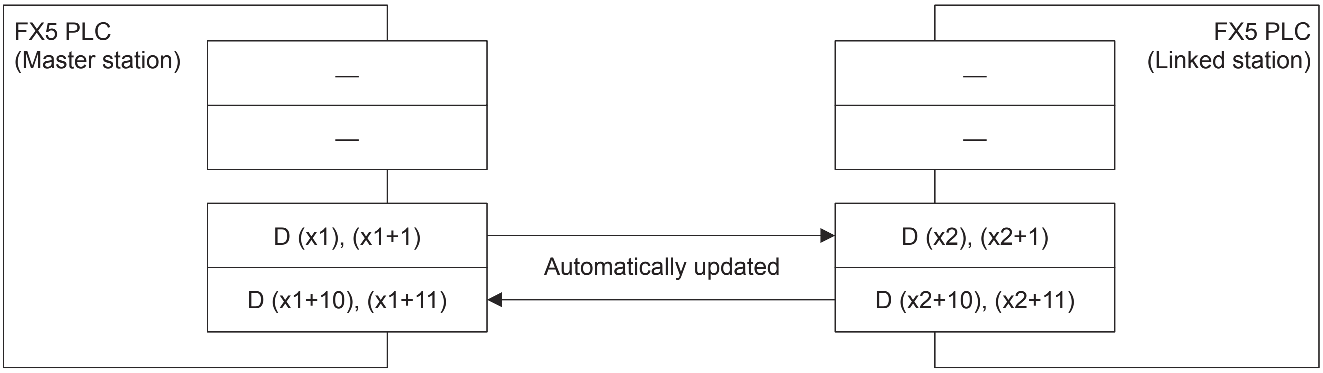
Mode		**Normal parallel link mode**		**High-speed parallel link mode**	
		Internal relay (M)	**Data register (D)**	**Internal relay (M)**	**Data register (D)**
Station number		**100 in each station**	**10 in each station**	**0 points**	**2 in each station**
Master station	For sending	M (y1) to M (y1 + 99)	D (x1) to D (x1 + 9)	—	D (x1), D (x1 + 1)
	For receiving	M (y1 + 100) to M (y1 + 199)	D (x1 + 10) to D (x1 + 19)	—	D (x1 + 10), D (x1 + 11)
Linked station	For receiving	M (y2) to M (y2 + 99)	D (x2) to D (x2 + 9)	—	D (x2), D (x2 + 1)
	For sending	M (y2 + 100) to M (y1 + 199)	D (x2 + 10) to D (x2 + 19)	—	D (x2 + 10), D (x2 + 11)

x1: [Master station] Link device head number of data register (D)
x2: [Linked station] Link device head number of data register (D)
y1: [Master station] Link device head number of internal relay (M)
y2: [Linked station] Link device head number of internal relay (M)

■Normal parallel link mode



■High-speed parallel link mode



Precautions

When creating a program, do not change the information in devices used by other stations. If such information is changed, the other stations will not operate normally.

17.5 Wiring

This section explains about the wiring.

Wiring procedure

1. Preparing for wiring.

Prepare cables required for wiring. (☞ Page 292 Cable)

2. Turn OFF the PLC power.

Before wiring, make sure that the PLC power is OFF.

3. Wire the cables between the communication equipment.

Connect the RS-485 communication equipment. (☞ Page 295 Connection diagram)

Cable

Select cables using the procedure described below.

Twisted pair cable

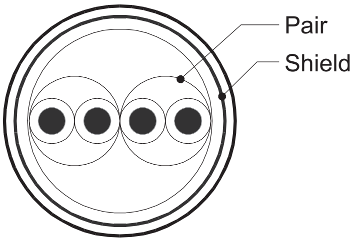
Use shielded twisted pair cables for connecting RS-485 communication equipment.

The specifications of the cables used in wiring are shown.

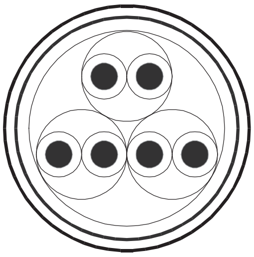
■RS-485 cable specifications

Items	Descriptions
Cable type	Shielded cable
Number of pairs	2p, 3p
Conductor resistance (20℃)	88.0 Ω/km or less
Insulation resistance	10000 MΩ-km or more
Voltage resistance	500 V DC, 1 minute
Electrostatic capacitance (1kHz)	60 nF/km or less as an average
Characteristic impedance (100kHz)	110±10Ω

■Cable structural drawing (reference)



Example of two-pair cable structural drawing



Example of three-pair cable structural drawing

Connecting cables

The table below shows applicable cables and tightening torques.

Item	Number of wires connected per terminal	Wire size		Tightening torque
		Solid wire, Stranded wire	Wire ferrule with insulation sleeve	
FX5U CPU module built-in RS-485 port	One wire	0.2 to 0.5mm^2 (24 to 20 AWG)	0.2 to 0.5mm^2 (24 to 20 AWG)	0.22 to 0.25N·m
	Two wires	0.2mm^2 (24 AWG)	—	
FX5UC CPU module built-in RS-485 port FX5-485-BD FX5-485ADP	One wire	0.3 to 0.5mm^2 (22 to 20 AWG)	0.3 to 0.5mm^2 (22 to 20 AWG)	
	Two wires	0.3mm^2 (22 AWG)	—	

Precautions

Do not tighten terminal screws exceeding the specified torque range. Otherwise it may cause equipment failure or malfunction.

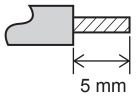
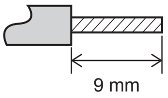
■Wire end treatment

With regard to the cable end treatment, use a stranded cable or solid cable as is, or use a wire ferrule with insulating sleeve.

- When using a stranded cable or solid cable as is

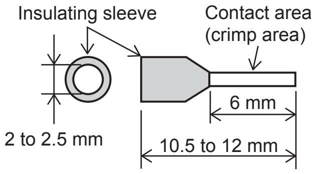
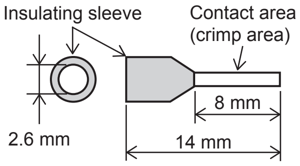
- Twist the end of stranded wire and make sure that there are no loose wires.

- Please do not solder plate the ends of the cable.

Dimensions of the cable end	
FX5U CPU module built-in RS-485 port	FX5UC CPU module built-in RS-485 port, FX5-485-BD, FX5-485ADP
	

- When using a wire ferrule with insulation sleeve

Because it is difficult to insert a cable into an insulating sleeve depending on the thickness of the cable sheath, select the proper cable according to the outline drawing.

FX5U CPU module built-in RS-485 port	FX5UC CPU module built-in RS-485 port, FX5-485-BD, FX5-485ADP
	

<Reference>

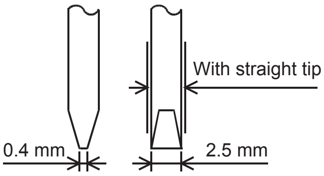
Item	Manufacturer	Model name	Crimping tool
FX5U CPU module built-in RS-485 port	PHOENIX CONTACT GmbH & Co. KG	AI 0.5-6 WH	CRIMPFOX 6 CRIMPFOX 6T-F
FX5UC CPU module built-in RS-485 port FX5-485-BD FX5-485ADP		AI 0.5-8 WH	

- Tool

For tightening the terminal, use a commercially available small screwdriver with straight tip that is not widened toward the end as shown below.

■Precautions

If the diameter of the screwdriver tip is too small, the required tightening torque cannot be achieved. To achieve the appropriate tightening torque shown in the previous page, use the following screwdriver or its equivalent (grip diameter: approximately 25mm).



<Reference>

Manufacturer	Model name
PHOENIX CONTACT GmbH & Co. KG	SZS 0.4×2.5

Termination resistor setting

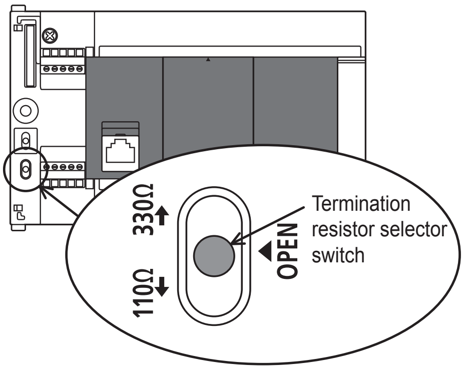
Make sure to provide a termination resistor at both ends of the wire.

The built-in RS-485 port, FX5-485-BD and FX5-485ADP have a built-in termination resistor.

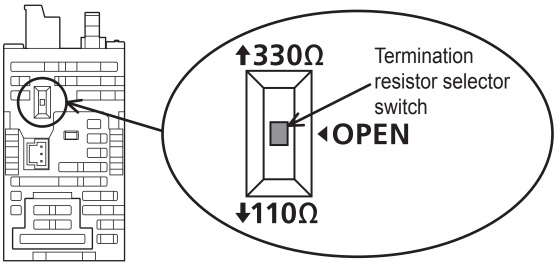
Set the termination resistor selector switch as below.

Wiring	Termination resistor selector switch
Two-pair wiring	330 Ω
One-pair wiring	110 Ω

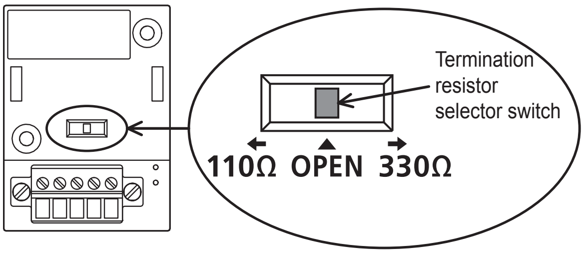
■FX5U CPU module built-in RS-485 port



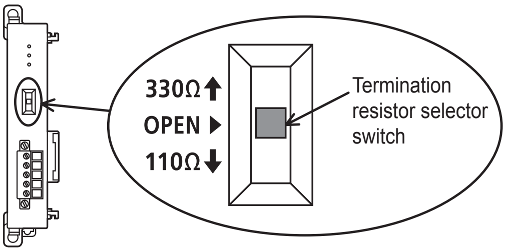
■FX5UC CPU module built-in RS-485 port



■FX5-485-BD



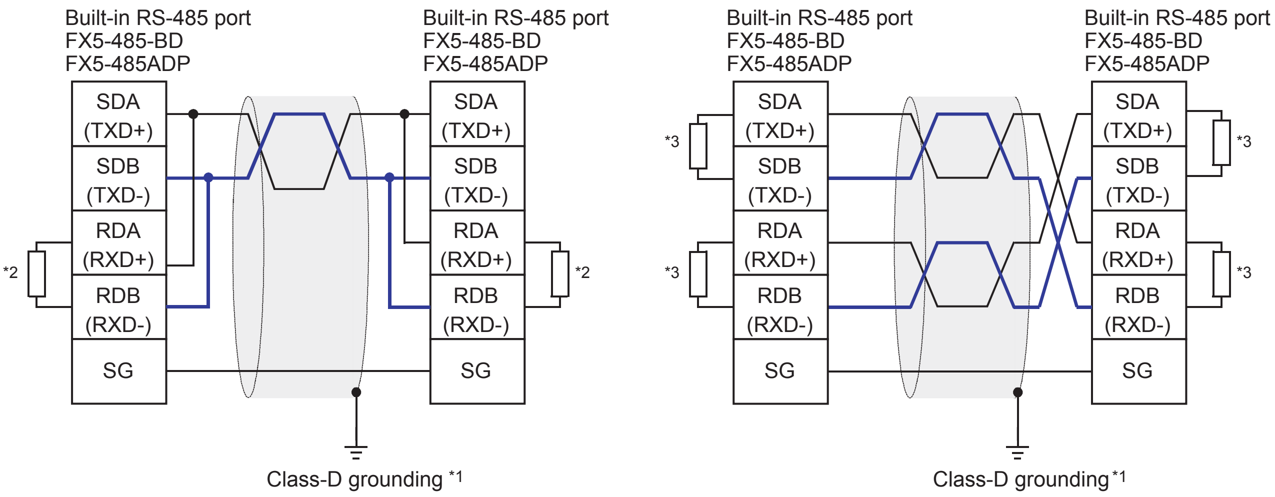
■FX5-485ADP



Connection diagram

■One-pair wiring

■Two-pair wiring



*1 Make sure to perform Class-D grounding on the shield of the twisted pair cable to be connected.

*2 Set the termination resistor selector switch to 110 Ω.

*3 Set the termination resistor selector switch to 330 Ω.

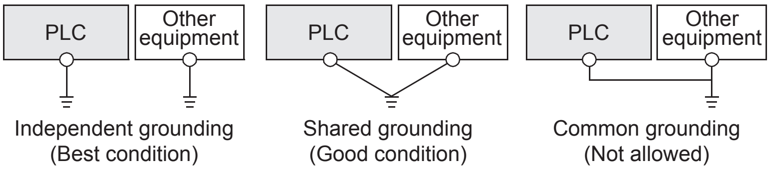
Grounding

Grounding should be performed as stated below.

- Perform D class grounding. (Grounding resistance: 100Ω or less)
- Independent grounding should be performed for best results.

If the PLC cannot be grounded independently, perform the "Shared grounding" shown below.

For details, refer to User's Manual (Hardware) of the CPU module used.



- The grounding wire size should be 14 AWG (2mm²) or larger.
- Bring the grounding point close to the PLC as much as possible so that the ground cable can be shortened.

17.6 Communication Setting

For the FX5 communication settings of this function, parameters are set using GX Works3. For details about GX Works3, refer to 🕮GX Works3 Operating Manual.

Setting of parameter differs according to the module used. The procedure for each module is as follows.

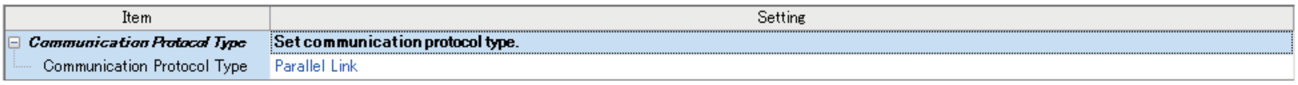
Built-in RS-485 port (CH1)

Navigation Window ⇨ Parameter ⇨ FX5UCPU⇨ Module Parameter ⇨ 485 Serial Port

Window

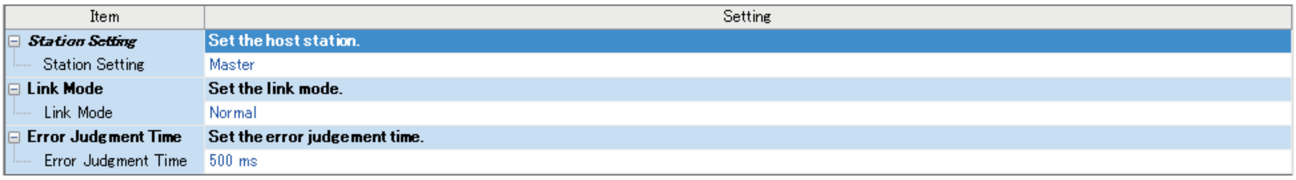
The following screen will be displayed if [Parallel Link] is set for the communication protocol type.

■Basic Settings



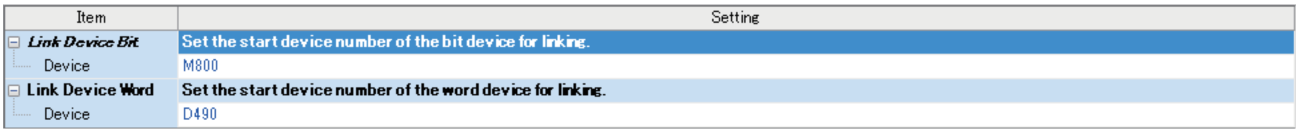
Item	Setting
⊟ *Communication Protocol Type*	Set communication protocol type.
Communication Protocol Type	Parallel Link

■Fixed Setting



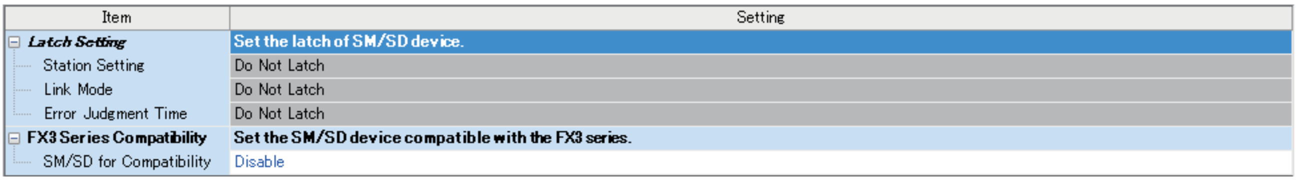
Item	Setting
⊟ *Station Setting*	Set the host station.
Station Setting	Master
⊟ Link Mode	Set the link mode.
Link Mode	Normal
⊟ Error Judgment Time	Set the error judgement time.
Error Judgment Time	500 ms

■Link Device



Item	Setting
⊟ *Link Device Bit*	Set the start device number of the bit device for linking.
Device	M800
⊟ Link Device Word	Set the start device number of the word device for linking.
Device	D490

■SM/SD Setting



Item	Setting
⊟ *Latch Setting*	Set the latch of SM/SD device.
Station Setting	Do Not Latch
Link Mode	Do Not Latch
Error Judgment Time	Do Not Latch
⊟ FX3 Series Compatibility	Set the SM/SD device compatible with the FX3 series.
SM/SD for Compatibility	Disable

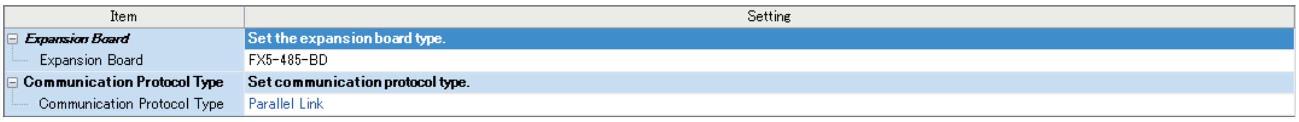
Communication board (CH2)

Navigation Window ⇨ Parameter ⇨ Model name ⇨ Module Parameter ⇨ Extended Board

Window

The following screen will be displayed if [FX5-485-BD] is set for the extended board and [Parallel Link] is set for the communication protocol type. Fixed setting and SM/SD setting are the same as for the built-in RS-485 port (CH1).

■Basic Settings



Item	Setting
⊟ *Expansion Board*	Set the expansion board type.
Expansion Board	FX5-485-BD
⊟ Communication Protocol Type	Set communication protocol type.
Communication Protocol Type	Parallel Link

Communication adapter (CH3/CH4)

When an expansion adapter is used, add expansion adapter to Module Information.

Navigation window ⇨ Parameter ⇨ Module Information ⇨ Right-click ⇨ Add New Module

After adding the expansion adapter, make settings on the screen displayed from the following operation.

Navigation window ⇨ Parameter ⇨ Module Information ⇨ ADP1 to ADP6 (Communication adapter) ⇨ Module Parameter

Window

Each setting screen is the same as for the built-in RS-485 port (CH1).

Parameter setting details

Set the following items for the serial ports that use the parallel link. However, only one channel can be set to the parallel link.

Items			Setting value	Reference section
Basic Settings	Extended Board*1		When using this function, select [FX5-485-BD].	—
	Protocol Type		When using this function, select [Parallel link].	
Fixed Setting	Station Setting		0: Master station 1: Linked station	Page 290
	Link Mode		0: Normal 1: High-Speed	
	Error Judgment Time		10 to 32767 (ms)	—
Link Device	Link Device Bit		FX5S CPU module: M0 to M32568 FX5UJ CPU module: M0 to M7480 FX5U/FX5UC CPU module: M0 to M32568	Page 290
	Link Device Word		Link Mode = Normal: D0 to 7980 Link Mode = High-speed: D0 to 7988	
SM/SD Setting	FX3 Series Compatibility	SM/SD for Compatibility	Disable/CH1/CH2	Page 297

*1 Only for the communication board (CH2).

Setting is not required (fixed value) for the board below.

Item	Descriptions
Start bit	1 bit
Data length	7 bits
Parity	Even
Stop bit	1 bit
Baud rate	115200bps
Header	Not Added
Terminator	Not Added
Control Mode	None
Sum Check	Not Added
Control procedure	Form 1 (CR and LF are not added)

FX3 Series-compatible SM/SD

When using the FX3 Series compatible SM/SD storage area, set to use special devices for either the FX3 Series CH1 or CH2. FX3 Series compatible devices corresponding to the specified channel can be used.

For details, refer to the following.

☞ Page 303 Related Devices

17.7 Programming

This section explains how to set the parallel link and how to create programs.

There are two parallel link modes, normal parallel link mode and high-speed parallel link mode.

The modes vary in applicable devices and number of link points.

When connecting FX5 PLCs through the parallel link, set the master and linked stations in the same link mode.

Communication settings

Serial communication setting are as follows. (☞ Page 296 Communication Setting)

Items		**Setting**	
		Master station	**Linked station**
Communication Protocol Type		Parallel link	
Fixed Setting	Station Setting	Master	Linked station
	Link Mode	Normal	
	Error Judgment Time	500	500
Link Device	Link Device Bit	M4000	M6000
	Link Device Word	D990	D1990

Contents of related devices

The devices used in the program are shown below. (☞ Page 303 Related Devices)

Devices for setting parallel link

These devices are used for determining errors in the parallel link. Use these devices to output the link errors or use them in interlock sequence programs.

Device No.	**Name**	**Descriptions**
SM9090	Parallel link operation	ON during parallel link operation.
SM8500[*1]	Serial communication error	Turns ON when an error occurs in serial communication.

*1 Use the serial communication error (SM8500, SM8510, SM8520 or SM8530) corresponding to each channel executing the parallel link.

Link device

■Device for master station

- Sending device

Device for sending information from master station to linked station

Device No.		Number of link points	Descriptions
■In the case of normal parallel link mode			
Internal relay	M4000 to M4099	100 points	The status of the master station device is automatically updated to the linked station device.
Word devices	D990 to D999	10 points	
■In the case of high-speed parallel link mode			
Word devices	D990, D991	2 points	The status of the master station device is automatically updated to the linked station device.

- Receiving device

Device for transferring information from linked station to master station. Do not change this device information at the master station. If such information is changed, the other station will not operate normally.

Device No.		Number of link points	Descriptions
■In the case of normal parallel link mode			
Internal relay	M4100 to M4199	100 points	The status of the linked station device is received and automatically updated to the master station device.
Word devices	D1000 to D1009	10 points	
■In the case of high-speed parallel link mode			
Word devices	D1000, D1001	2 points	The status of the linked station device is received and automatically updated to the master station device.

■Device for linked station

- Sending device

Device for sending information from linked station to master station

Device No.		Number of link points	Descriptions
■In the case of normal parallel link mode			
Internal relay	M6100 to M6199	100 points	The status of linked station device is automatically updated to the master station device.
Word devices	D2000 to D2009	10 points	
■In the case of high-speed parallel link mode			
Word devices	D2000, D2001	2 points	The status of linked station device is automatically updated to the master station device.

- Receiving device

Device for transferring information from master station to linked station. Do not change this device information at the linked station. If such information is changed, the other station will not operate normally.

Device No.		Number of link points	Descriptions
■In the case of normal parallel link mode			
Internal relay	M6000 to M6099	100 points	The status of the master station device is received and automatically updated to the linked station device.
Word devices	D1990 to D1999	10 points	
■In the case of high-speed parallel link mode			
Word devices	D1990, D1991	2 points	The status of the master station device is received and automatically updated to the linked station device.

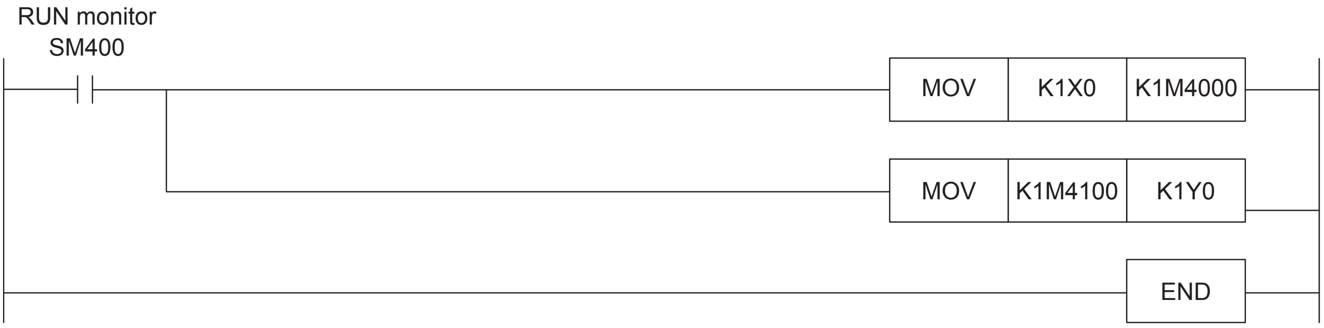
Communication test

It is recommended to wire the master and linked stations, perform communication settings, and then execute the communication test using the following procedure to confirm proper operation.
Communication test program is not required during operation.

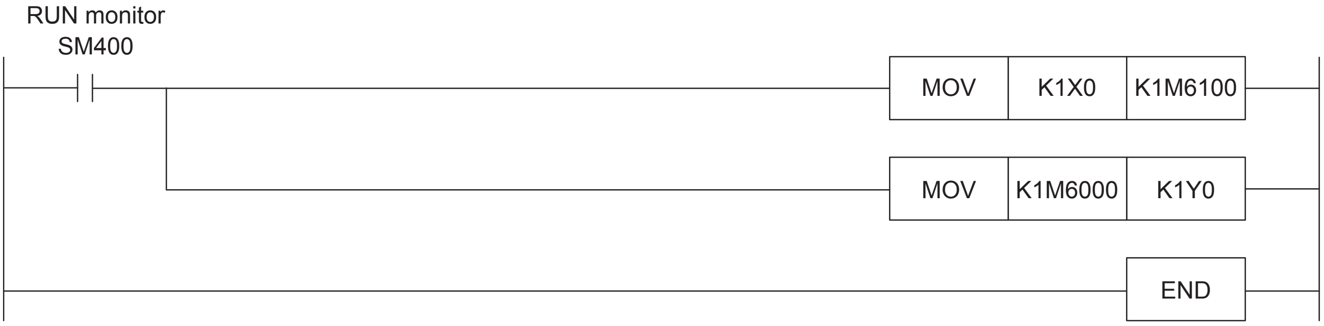
Communication test process

1. After writing the communication configuration of the master and linked stations and the program, turn OFF→ON or reset the power supply to the FX5 PLC.
2. Check if the SD and RD LEDs of the used channel are flashing. If the LEDs are OFF, refer to troubleshooting. (☞ Page 794 TROUBLESHOOTING PROCEDURE)
3. Operate the PLC inputs (X0 to X3) of the master station, and check that the outputs (Y0 to Y3) of the linked station turn ON.
4. Operate the PLC inputs (X0 to X3) of the linked station, and check that the outputs (Y0 to Y3) of the master station turn ON.

Test program for communication (master station)

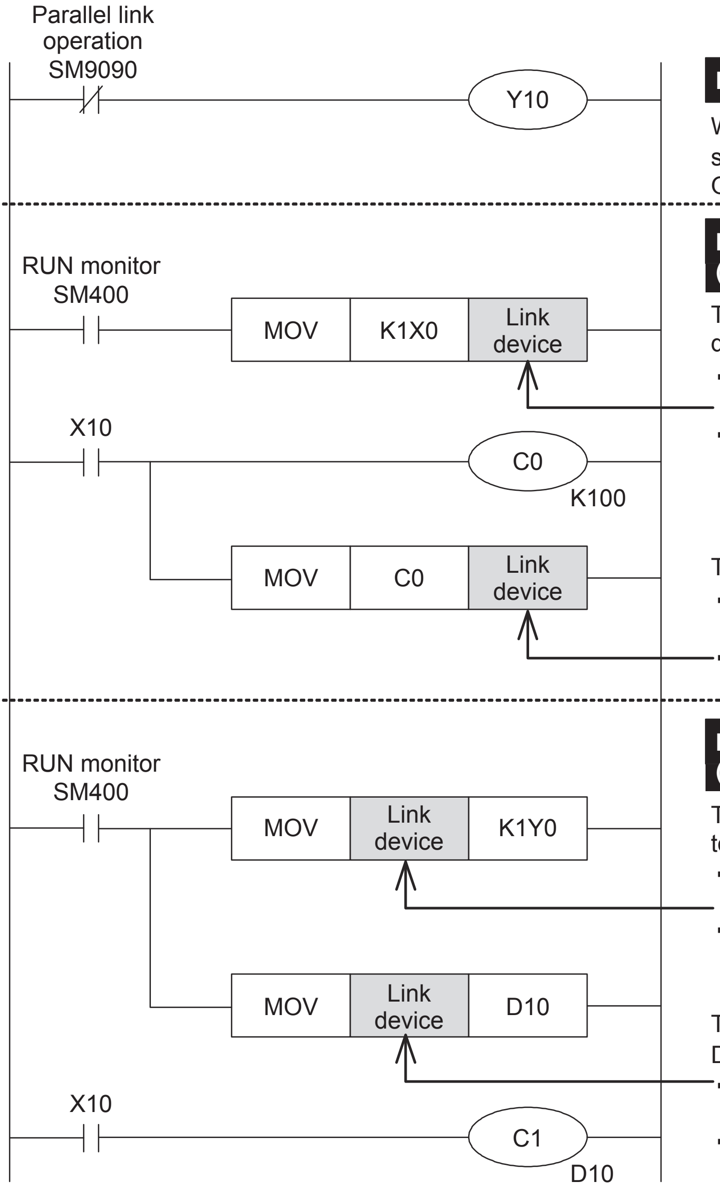


Test program for communication (linked station)



Creating programs for the master station

Create programs for the master station.



Program for indicating link errors

When the parallel link of the master/linked station is suspended or a setting error occurs, Y10 is set to ON.

Program for writing link devices (master station → linked station)

The information from X0 to X3 is written to the link device.

- In the case of normal parallel link mode: M4000 to M4099 ("K1M4000" etc.)
- In the case of high-speed parallel link mode: D990, D991

The current value of C0 is written to the link device.

- In the case of normal parallel link mode: D990 to D999
- In the case of high-speed parallel link mode: D990, D991

Program for reading link devices (linked station → master station)

The information from the link device is output to Y0 to Y3.

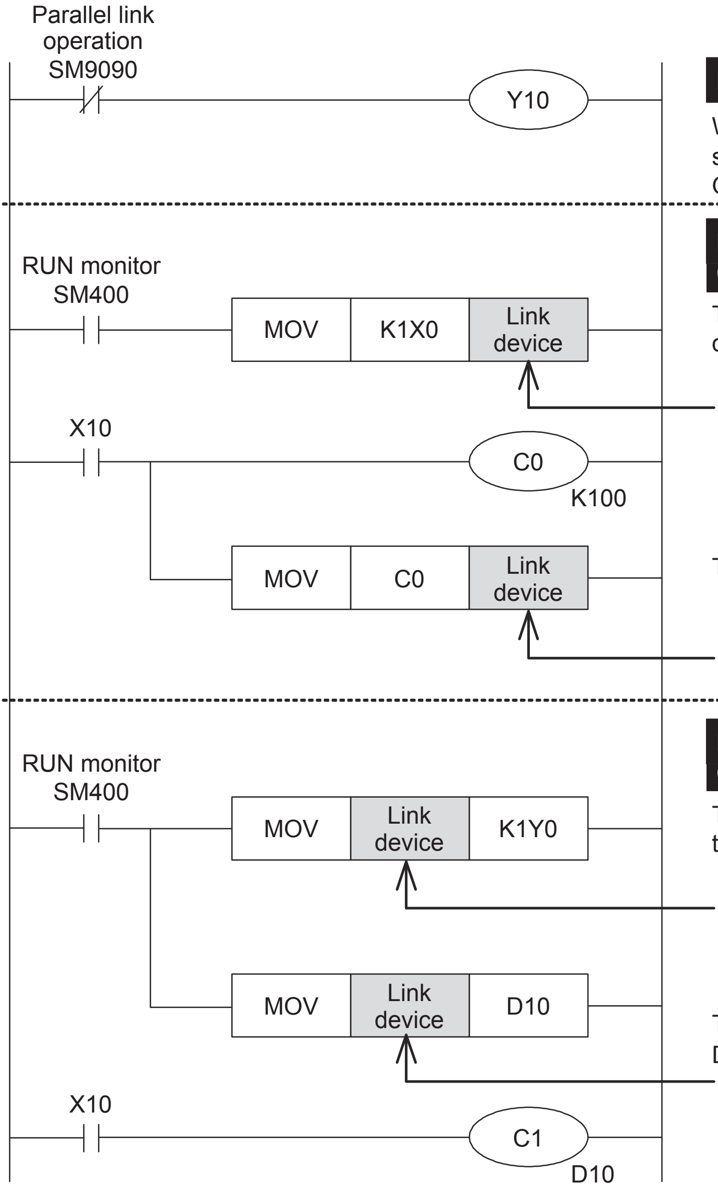
- In the case of normal parallel link mode: M4100 to 4199 ("K1M4100" etc.)
- In the case of high-speed parallel link mode: D1000, D1001

The current value of the link device is read to D10, and set to the setting value of C1.

- In the case of normal parallel link mode: D1000 to D1009
- In the case of high-speed parallel link mode: D1000, D1001

Creating programs for the linked station

Create programs for the linked station.



Program for indicating link errors

When the parallel link of the master/linked station is suspended or a setting error occurs, Y10 is set to ON.

Program for writing link devices (linked station → master station)

The information from X0 to X3 is written to the link device.

- In the case of normal parallel link mode: M6100 to M6199 ("K1M6100" etc.)
- In the case of high-speed parallel link mode: D2000, D2001

The current value of C0 is written to the link device.

- In the case of normal parallel link mode: D2000 to D2009
- In the case of high-speed parallel link mode: D2000, D2001

Program for reading link devices (master station → linked station)

The information from the link device is output to Y0 to Y3.

- In the case of normal parallel link mode: M6000 to M6099 ("K1M6000" etc.)
- In the case of high-speed parallel link mode: D1990, D1991

The current value of the link device is read to D10, and set to the setting value of C1.

- In the case of normal parallel link mode: D1990 to D1999
- In the case of high-speed parallel link mode: D1990, D1991

Cautions on program creation

- Do not change the contents of link devices at other stations. When a link error occurs, the link device information is held in the status just before occurrence of the link error. Create program to operate safely.
- Link device update in the parallel link is asynchronous to operations of the sequence program. The link devices are not updated in the middle of ladder operation.

17.8 Related Devices

This section describes the special relay/special register functions used in the parallel link function.

Point

Available communication channels vary depending on the CPU module and system configuration.
For communication channels, refer to ☞ Page 287 System Configuration.
"FX3 Series compatible" devices operate only on the communication channel specified in the compatible SM/SD for communication settings.
For compatible SM/SD, refer to ☞ Page 296 Communication Setting.

List of related devices

Special relays

■FX5 dedicated

R: Read only, M: Master station, L: Linked station

Device No.				Name	Descriptions	Detection	R/W
CH1	CH2	CH3	CH4				
SM8500	SM8510	SM8520	SM8530	Serial communication error	Turns ON when an error occurs in serial communication.	M, L	R
SM9090				Parallel link operation	ON during parallel link operation.	M, L	R

■FX3 Series compatible

R: Read only, M: Master station, L: Linked station

Device No.		Name	Descriptions	Detection	R/W
CH1	CH2				
SM8063	SM8438	Serial communication error	Turns ON when an error occurs in serial communication.	M, L	R
SM8072		Parallel link operation	ON during parallel link operation.	M, L	R

Special registers

■FX5 dedicated

R: Read only, M: Master station, L: Linked station

Device No.				Name	Descriptions	Detection	R/W
CH1	CH2	CH3	CH4				
SD8500	SD8510	SD8520	SD8530	Serial communication error code	Stores the error code when a serial communication error occurs.	M, L	R
SD8502	SD8512	SD8522	SD8532	Serial communication settings	Stores the communication parameter set in the PLC.	M, L	R
SD8503	SD8513	SD8523	SD8533	Serial communication operation mode	Stores status of the current communication being executed.	M, L	R
SD9090				Master station/linked station setting	Stores the property setting value of the master/linked station.	M, L	R
SD9091				Link mode setting	Stores the property setting value of the link mode.	M, L	R
SD9092				Error determination time setting	Stores the property setting value of the error determination time.	M, L	R

■FX3 Series compatible

R: Read only, M: Master station, L: Linked station

Device No.		Name	Descriptions	Detection	R/W
CH1	CH2				
SD8063	SD8438	Serial communication error code	Stores the error code when a serial communication error occurs.	M, L	R
SD8419	SD8439	Serial communication operation mode	Stores status of the current communication being executed.	M, L	R

Details of related devices

Serial communication error

Turns ON when an error occurs in serial communication.

R: Read only

FX5 dedicated				FX3 Series compatible		Descriptions	R/W
CH1	CH2	CH3	CH4	CH1	CH2		
SM8500	SM8510	SM8520	SM8530	SM8063	SM8438	Turns ON when an error occurs in serial communication.	R

After a device above turns ON, the error code is stored in the corresponding device below.

FX5 dedicated				FX3 Series compatible		Name	Descriptions
CH1	CH2	CH3	CH4	CH1	CH2		
SD8500	SD8510	SD8520	SD8530	SD8063	SD8438	Serial communication error code	Stores the error code when a serial communication error occurs.

Precautions

Do not turn ON or OFF with program or engineering tool.

Serial communication error is not cleared even after communication is restored to normal state. Devices are cleared when power is turned OFF→ON, STOP→RUN, reset, or SM50 (Error Detection Reset Completion) is turned ON.

Parallel link operation

Device for confirming whether the parallel link is running

R: Read only

FX5 dedicated	FX3 Series compatible	Descriptions	R/W
SM9090	SM8072	ON: In normal running state OFF: Stopped	R

Precautions

Do not turn ON or OFF with program or engineering tool.

Serial communication error code

This device stores the error code for serial communication (☞ Page 835 Checking absence/presence of parallel link function errors).

R: Read only

FX5 dedicated				FX3 Series compatible		Descriptions	R/W
CH1	CH2	CH3	CH4	CH1	CH2		
SD8500	SD8510	SD8520	SD8530	SD8063	SD8438	Stores the error code when the serial communication error occurs.	R

Precautions

Do not change the value with program or engineering tool.

Serial communication errors are not cleared even after communication has recovered its normal status. Devices are cleared when power is turned OFF→ON, STOP→RUN, reset, or SM50 (Error Detection Reset Completion) is turned ON.

Serial communication setting

Stores the communication parameters set in the communication settings (☞ Page 296 Communication Setting).

R: Read only

FX5 dedicated				FX3 Series compatible		Descriptions	R/W
CH1	CH2	CH3	CH4	CH1	CH2		
SD8502	SD8512	SD8522	SD8532	SD8405	SD8425	Stores the setting of the communication parameter.	R

The description of the communication parameter is as follows.

Bit No.	Name	Descriptions	
		0 (bit is OFF)	1 (bit is ON)
b0	Data length	7 bits	—
b1 b2	Parity bit	b2, b1 (1, 1): Even	
b3	Stop bit	1 bit	—
b4 b5 b6 b7	Baud rate	b7, b6, b5, b4 (1, 1, 0, 1): 115200bps	
b10 b11 b12	Control mode	b12, b11, b10 (1, 1, 1): RS-485 half-duplex, bi-directional (RS-485)	

Precautions

Do not change the value with program or engineering tool.

Serial communication operation mode

Stores the communication function code of the serial communication being used.

R: Read only

FX5 dedicated				FX3 Series compatible		Descriptions	R/W
CH1	CH2	CH3	CH4	CH1	CH2		
SD8503	SD8513	SD8523	SD8533	SD8419	SD8439	0: MELSOFT connection or MC protocol 3: N:N Network Communication 5: Non-Protocol Communication 6: Parallel Link Communication 7: Inverter Communication 9: MODBUS RTU Communication 12: Predefined protocol support Other than above: Not used	R

Precautions

Do not change the value with program or engineering tool.

Master station/linked station setting

Stores the master/linked station setting value in the serial communication settings.

R: Read only

FX5 dedicated	Descriptions	R/W
SD9090	0: Master station 1: Linked station	R

Precautions

Do not change the value with program or engineering tool.

Link mode setting

Stores the property setting value of the link mode in the serial communication settings.

R: Read only

FX5 dedicated	Descriptions	R/W
SD9091	0: Normal parallel link mode 1: High-speed parallel link mode	R

Precautions

Do not change the value with program or engineering tool.

Error determination time setting

Stores the error determination time setting value in the serial communication settings.

R: Read only

FX5 dedicated	Descriptions	R/W
SD9092	10 to 32767 (ms)	R

Precautions

Do not change the value with program or engineering tool.

18 MC PROTOCOL

This chapter explains MC protocol.

18.1 Function Summary

An MC protocol function is a function to access the equipment compatible with MC protocol from a CPU module and counterpart equipment (such as a personal computer or an HMI) using serial communication.

In case of a serial port of FX5, communication is possible by MC protocol A-compatible 1C frame and QnA-compatible 3C/4C frame.

For details, refer to the following.

☞ Page 684 OUTLINE

18.2 Procedures Before Operation

The flow chart below shows the MC protocol setting procedures up until data link.

1. Checking communication specifications

For communication specifications, link specifications, and link time, refer to ☞ Page 311 Specifications.

2. System configuration and selection

For select communication equipment, refer to ☞ Page 308 System Configuration.

3. Wiring

For wiring with twisted pair cable, and wiring example, refer to ☞ Page 314 Wiring.

4. Communication settings*1

For communication settings of communication device, refer to ☞ Page 320 Communication Setting.

5. Program creation

*1 For details on operating procedure of GX Works3, refer to the manual below.
📖 GX Works3 Operating Manual

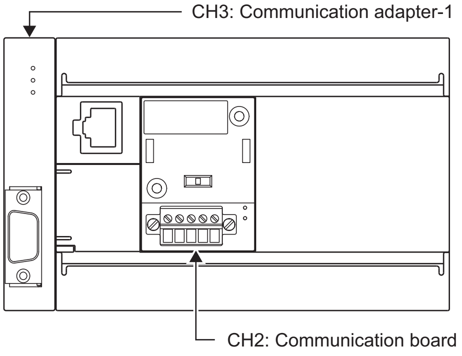
18.3 System Configuration

This section outlines the system configuration required to use the MC protocol.

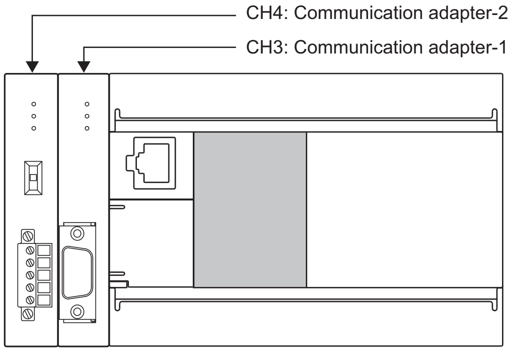
FX5S CPU module

A serial port of up to 2 channels can be connected in the FX5S CPU module by using a communication board and communication adapter.
Communication channel assignments are fixed regardless of the system configuration.
The combinations available for the system configurations are shown below.

[Configuration example 1]



[Configuration example 2]



Item		Serial port	Important points in selection	Overall distance
Communication board	FX5-485-BD	CH2	Because the board can be mounted on top of the CPU module, there is no change in the installation space to be required.	50 m or less
	FX5-422-BD-GOT			Depends on the specifications of GOT*1
	FX5-232-BD			15 m or less
Communication adapter	FX5-485ADP	CH3, CH4*2	Mount the communication adapter to the left of the CPU module.	1200 m or less
	FX5-232ADP			15 m or less

*1 For the maximum transmission distance, refer to the manuals for the models being connected.
*2 The adapters are assigned to CH3 and CH4 in the order, from the closest one to the CPU module.

Precautions

Total number of communication boards and communication adapters that can be connected is two.

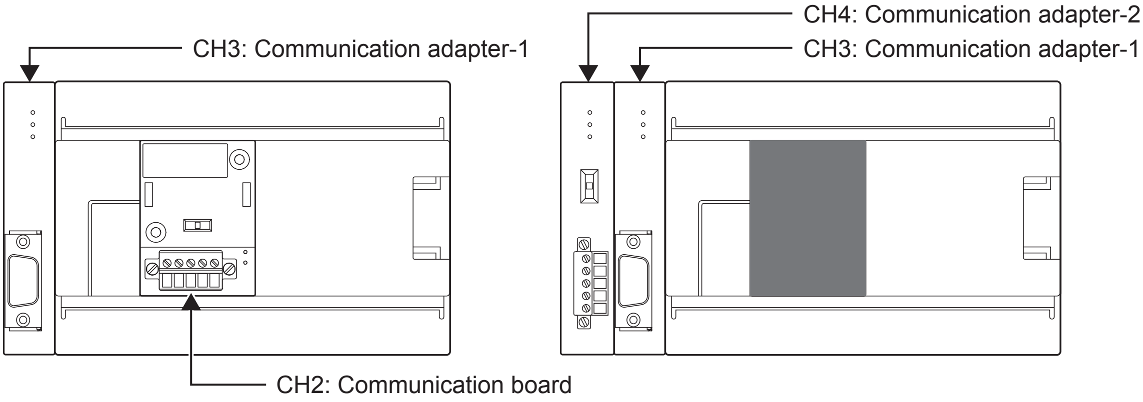
FX5UJ CPU module

A serial port of up to 2 channels can be connected in the FX5UJ CPU module by using a communication board and communication adapter.

Communication channel assignments are fixed regardless of the system configuration.

The combinations which can be configured are shown below.

[Configuration example 1] [Configuration example 2]



Item		Serial port	Important points in selection	Overall distance
Communication board	FX5-485-BD	CH2	Mounted on top of the CPU module, there is no change in the installation space requirements.	50 m or less
	FX5-422-BD-GOT			Depends on the specifications of GOT[1]
	FX5-232-BD			15 m or less
Communication adapter	FX5-485ADP	CH3, CH4[2]	Mount the communication adapter to the left of the CPU module.	1200 m or less
	FX5-232ADP			15 m or less

[1] For the maximum transmission distance, refer to the manuals for the models being connected.

[2] The adapters are assigned to CH3 and CH4 in the order, from the closest one to the CPU module.

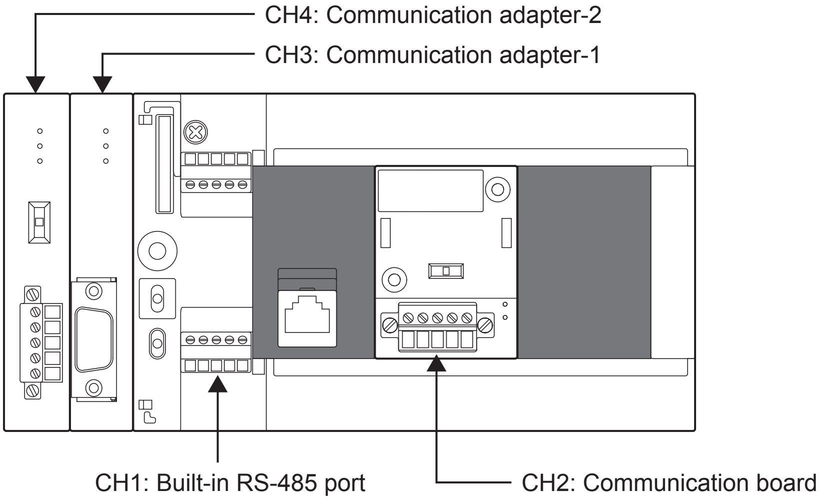
Precautions

Total number of communication boards and communication adapters that can be connected is two.

FX5U CPU module

A serial port of up to 4 channels can be connected in the FX5U CPU module by using the built-in RS-485 port, communication board, and communication adapter.

Communication channel assignments are fixed regardless of the system configuration.



Item		Serial port	Important points for selection	Overall distance
Built-in RS-485 port		CH1	Since it is built-into the CPU module, there is no need to add equipment.	50 m or less
Communication board	FX5-485-BD	CH2	Mounted on top of the CPU module, there is no change in the installation space requirements.	50 m or less
	FX5-422-BD-GOT			Depends on the specifications of GOT[*1]
	FX5-232-BD			15 m or less
Communication adapter	FX5-485ADP	CH3, CH4[*2]	Mount the communication adapter to the left of the CPU module.	1200 m or less
	FX5-232ADP			15 m or less

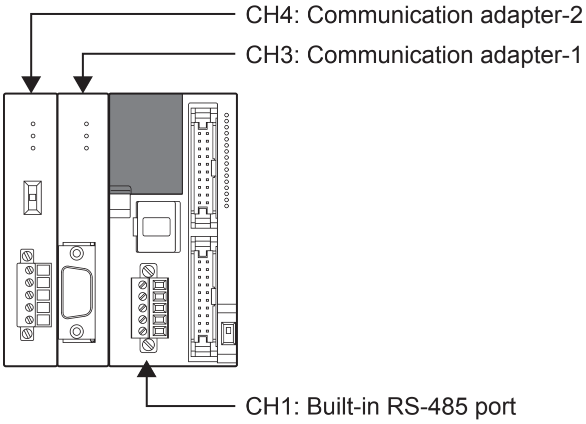
*1 For the maximum transmission distance, refer to the manuals for the models being connected.

*2 The adapters are assigned to CH3 and CH4 in the order, from the closest one to the CPU module.

FX5UC CPU module

A serial port of up to 3 channels can be connected in the FX5UC CPU module by using the built-in RS-485 port and communication adapter.

Communication channel assignments are fixed regardless of the system configuration.



Item		Serial port	Important points for selection	Overall distance
Built-in RS-485 port		CH1	Since it is built-into the CPU module, there is no need to add equipment.	50 m or less
Communication adapter	FX5-485ADP	CH3, CH4[*1]	Mount the communication adapter to the left of the CPU module.	1200 m or less
	FX5-232ADP			15 m or less

*1 The adapters are assigned to CH3 and CH4 in the order, from the closest one to the CPU module.

18.4 Specifications

Communication specifications

Communication is executed within the specifications shown in the table below. Set the baud rate, etc. in the parameter settings of an engineering tool.

Items		Specifications		Remarks
Number of connectable modules		16 maximum		—
Transmission standard		RS-485 or RS-232C standard		—
Maximum total extension distance	RS-485	• When using FX5-485ADP: 1200m or less • When using built-in RS-485 port or FX5-485-BD: 50m or less		Distance varies depending on communication equipment type.
	FX5-422-BD-GOT	Depends on the specifications of GOT[1]		
	RS-232C	15m or less		
Protocol type		MC protocol (dedicated protocol)	4C frame	There are format 1/format 4/format 5.[3]
			3C frame	There are format 1/format 4.[3]
			1C frame[2]	There are format 1/format 4.[3]
Control procedure		—		—
Communication method		Half-duplex, bidirectional communication		—
Baud rate		300/600/1200/2400/4800/9600/19200/38400/57600/115200(bps)		—
Character format	Start bit	1 bit		—
	Data length	7 bits/8 bits		When the protocol type is format5, the data length should be 8 bits.
	Parity bit	None, odd or even		—
	Stop Bit	1 bit/2 bits		—
Header		Fixed		—
Terminator		Fixed		—
Control line		RS-485: None/ RS-232C: Provided		—
Sum check		Provided or not provided		—

[1] For the maximum transmission distance, refer to the manuals for the models being connected.

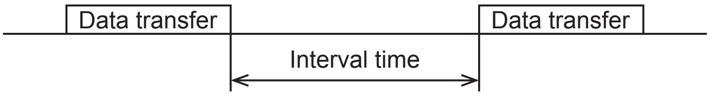
[2] For the FX5U/FX5UC CPU module 1C frame compatible version, refer to ☞ Page 936 Added and Changed Functions.

[3] For the message format of each format, refer to the following.
☞ Page 692 MESSAGE FORMAT

Link specifications

Link time

■Data transfer



■Data transfer time

R: Number of read data points, W: Number of written data points, T: Time to send or receive one character, V: Interval time, S: Max Scan Time of PLC, D: Message waiting time

(1) 1C Frame

Time to read continuous word devices (such as data registers) in one station (ms)

$= (21^{*1} + 4 \times R^{*2}) \times T\ (ms) + V + S\ (SD8012) \times 3 + D$

Time to write continuous word devices (such as data registers) in one station (ms)

$= (20^{*1} + 4 \times W^{*2}) \times T\ (ms) + V + S\ (SD8012) + D$

*1 This is the number of characters when format 1 is used and the sum check is not provided. When format 4 is used, add "4" to this value. Further, when the sum check is provided, add "4" to this value also.

*2 The number of points is counted in 1-word units.

(2) 3C Frame

Time to read continuous word devices (such as data registers) in one station (ms)

$= (43^{*3} + 4 \times R^{*4}) \times T\ (ms) + V + S\ (SD524) \times 3 + D$

Time to write continuous word devices (such as data registers) in one station (ms)

$= (42^{*3} + 4 \times W^{*4}) \times T\ (ms) + V + S\ (SD524) \times 3 + D$

*3 This is the number of characters when format 1 is used and the sum check is not provided at the time of execution of batch read/write command.
When format4 is used, add "4" to this value. Further, when the sum check is provided, add "4" to this value also. Further, when specifying an extension, add "7" to this value also.

*4 The number of points is counted in 1-word units.

(3) 4C Frame: In case of ASCII code (When format 1 to format 4 are used)

Time to read continuous word devices (such as data registers) in one station (ms)

$= (49^{*5} + 4 \times R^{*6}) \times T\ (ms) + V + S\ (SD524) \times 3 + D$

Time to write continuous word devices (such as data registers) in one station (ms)

$= (48^{*5} + 4 \times W^{*6}) \times T\ (ms) + V + S\ (SD524) \times 3 + D$

*5 This is the number of characters when format 1 is used and the sum check is not provided at the time of execution of batch read/write command.
When format4 is used, add "4" to this value. Further, when the sum check is provided, add "4" to this value also. Further, when specifying an extension, add "7" to this value also.

*6 The number of points is counted in 1-word units.

(4) 4C Frame: In case of binary code (When format 5 is used)

Time to read continuous word devices (such as data registers) in one station (ms)

$= (42^{*7} + 4 \times R^{*8}) \times T\ (ms) + V + S\ (SD524) \times 3 + D$

Time to write continuous word devices (such as data registers) in one station (ms)

$= (40^{*7} + 4 \times W^{*8}) \times T\ (ms) + V + S\ (SD524) \times 3 + D$

*7 This is the number of characters when format 5 is used and the sum check is not provided at the time of execution of batch read/write command.
Further, when the sum check is provided, add "4" to this value also. Further, when specifying an extension, add "7" to this value also.
When "10H" exists in the data area, since DLE "10H" is added just before "10H", add "+ "10H numeral".

*8 The number of points is counted in 1-word units.

■Time to send or receive one character

The table below shows the time required to send or receive one character when the start bit is 1-bit, the data length is 7-bit, the parity is 1-bit, and the stop bit is 1-bit (total 10-bits).

Transmission speed (baud rate) (bps)	Time to send or receive 1 character (ms)
300	33.34
600	16.67
1200	8.34
2400	4.17
4800	2.08
9600	1.04
19200	0.52
38400	0.26
57600	0.17
115200	0.08

The tables below show the data transfer times depending on the number of continuously read or written word devices at transmission speeds of 9600 bps and 19200 bps when the message waiting time is 0 ms[1], the maximum scan time is 20 ms, and the interval time is 100 ms.

- When the transmission speed is 9600bps (Unit: Second)

Number of data points	Number of stations		
	Station No. 1	Station No. 8	Station No. 16
10 points	0.3	1.9	3.7
32 points	0.4	2.6	5.2
64 points	0.5	3.7	7.3

- When the transmission speed is 19200bps (Unit: Second)

Number of data points	Number of stations		
	Station No. 1	Station No. 8	Station No. 16
10 points	0.2	1.6	3.2
32 points	0.3	2.0	3.9
64 points	0.4	2.5	5.0

When the types of read or written devices increase, "Data transfer time shown in above table × Number of device types" is required.

When the number of read or written points exceeds "64", the transfer time increases.

Accordingly, for achieving efficient data transfer, it is recommended to decrease the number of types of transferred devices and use as many continuous device numbers as possible.

[1] When RS-485 one-pair wiring using FX-485PC-IF is adopted, the message waiting time (for every exchange) must be 70 to 150ms. When RS-485 two-pair wiring or RS-232C is adopted, the message waiting time becomes 0ms.

18.5 Wiring

This section explains the wiring.

Wiring procedure

1. Select the connection method.

Select the connection method suitable to the application. (☞ Page 314 Selecting connection method)

2. Make arrangements for wiring.

Prepare cables required for wiring. (☞ Page 315 Cable)

3. Turn OFF the PLC power.

Before wiring, make sure that the PLC power is OFF.

4. Wire the cables between the communication equipment.

Connect the RS-485 and RS-232C communication equipment. (☞ Page 318 Connection diagram)

Selecting connection method

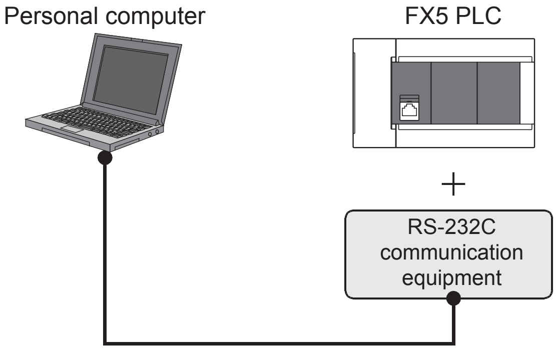
When using MC protocol 1C/3C/4C frame, connection can be achieved in accordance with RS-232C or RS-485 (RS-422) communication.

In FX5, it is possible to use a serial port of up to 4 channels simultaneously by using MC protocol.

Depending on each serial port, the available communication and channel number are decided.

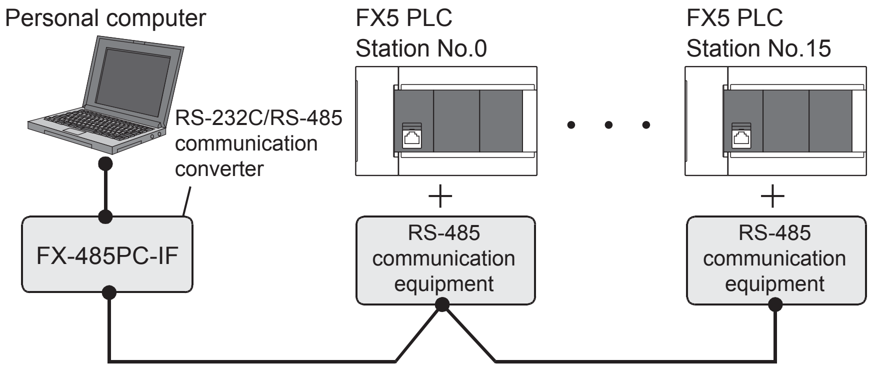
In case of RS-232C communication (1:1)

For communication in accordance with RS-232C, it is 1-to-1 connection. Make sure that the total extension distance is 15 m or less.



In case of RS-485 (RS-422) communication (1:N)

For communication in accordance with RS-485 (RS-422), up to 16 PLC can be connected.
Make sure that the total extension distance is 1200m or less. When using built-in RS-485 port, FX5-485-BD, or FX5-422-BD-GOT, the distance should be 50m or less.



One-pair wiring and two-pair wiring are applicable for communication in accordance with RS-485 (RS-422). The wiring method is determined for each application. Refer to the table below, and perform suitable wiring.
◎: Recommended wiring method, ○: Applicable wiring method, ×: Non-applicable wiring method

Item		One-pair wiring	Two-pair wiring
MC protocol[1]	When the message waiting time should be 70 ms or less	×	○
	When the message waiting time may be more than 70 ms	◎[2]	○

[1] When computer link is added to an existing system, adopt the wiring method used in the existing system.
[2] "Echo" is generated when the FX-485PC-IF is used in one-pair wiring.
Take proper countermeasures in the computer so that the echo can be ignored.

Cable

Select cables using the procedure described below.

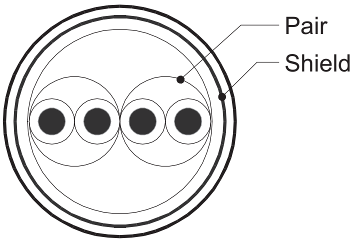
Twisted pair cable (RS-485)

Use shielded twisted pair cables for connecting RS-485 communication equipment.
The specifications of the cables used in wiring are shown.

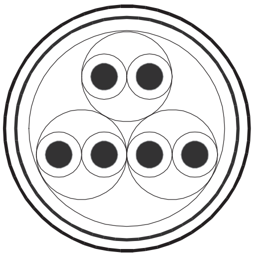
■RS-485 cable specifications

Items	Description
Cable type	Shielded cable
Number of pairs	2p, 3p
Conductor resistance (20℃)	88.0 Ω/km or less
Insulation resistance	10000 MΩ-km or more
Voltage resistance	500VDC, 1 minute
Electrostatic capacitance (1kHz)	60 nF/km or less as an average
Characteristic impedance (100kHz)	110±10 Ω

■Cable structural drawing (reference)



Example of two-pair cable structural drawing



Example of three-pair cable structural drawing

Connecting cables

The table below shows applicable cables and tightening torques.

Item	Number of wires connected per terminal	Wire size		Tightening torque
		Solid wire, Stranded wire	Wire ferrule with insulation sleeve	
FX5U CPU module built-in RS-485 port	One wire	0.2 to 0.5mm² (24 to 20 AWG)	0.2 to 0.5mm² (24 to 20 AWG)	0.22 to 0.25N·m
	Two wires	0.2mm² (24 AWG)	—	
FX5UC CPU module built-in RS-485 port FX5-485-BD FX5-485ADP	One wire	0.3 to 0.5mm² (22 to 20 AWG)	0.3 to 0.5mm² (22 to 20 AWG)	
	Two wires	0.3mm² (22 AWG)	—	

Precautions

Do not tighten terminal screws exceeding the specified torque range. Otherwise it may cause equipment failure or malfunction.

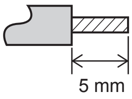
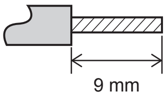
■Wire end treatment

With regard to the cable end treatment, use a stranded cable or solid cable as is, or use a wire ferrule with insulating sleeve.

- When using a stranded cable or solid cable as is

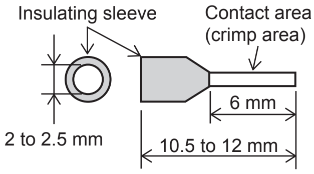
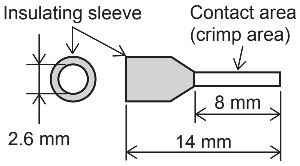
- Twist the end of stranded wire and make sure that there are no loose wires.

- Please do not solder plate the ends of the cable.

Dimensions of the cable end	
FX5U CPU module built-in RS-485 port	**FX5UC CPU module built-in RS-485 port, FX5-485-BD, FX5-485ADP**
	

- When using a wire ferrule with insulation sleeve

Because it is difficult to insert a cable into an insulating sleeve depending on the thickness of the cable sheath, select the proper cable according to the outline drawing.

FX5U CPU module built-in RS-485 port	FX5UC CPU module built-in RS-485 port, FX5-485-BD, FX5-485ADP
	

<Reference>

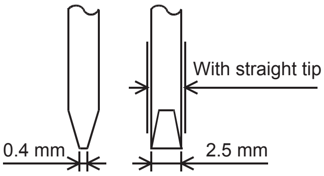
Item	Manufacturer	Model name	Crimping tool
FX5U CPU module built-in RS-485 port	PHOENIX CONTACT GmbH & Co. KG	AI 0.5-6 WH	CRIMPFOX 6 CRIMPFOX 6T-F
FX5UC CPU module built-in RS-485 port FX5-485-BD FX5-485ADP		AI 0.5-8 WH	

- Tool

For tightening the terminal, use a commercially available small screwdriver with straight tip that is not widened toward the end as shown below.

■Precautions

If the diameter of the screwdriver tip is too small, the required tightening torque cannot be achieved. To achieve the appropriate tightening torque shown in the previous page, use the following screwdriver or its equivalent (grip diameter: approximately 25mm).



<Reference>

Manufacturer	Model name
PHOENIX CONTACT GmbH & Co. KG	SZS 0.4×2.5

Termination resistor setting (RS-485)

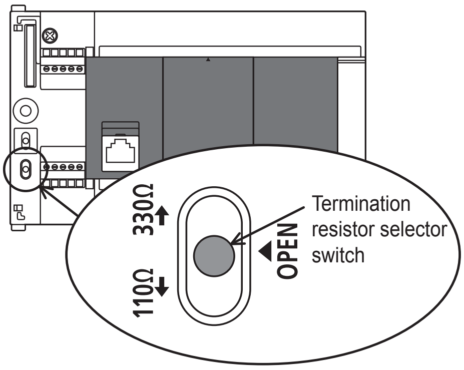
Make sure to provide a termination resistor at both ends of the wire.

The built-in RS-485 port, FX5-485-BD and FX5-485ADP have a built-in termination resistor.

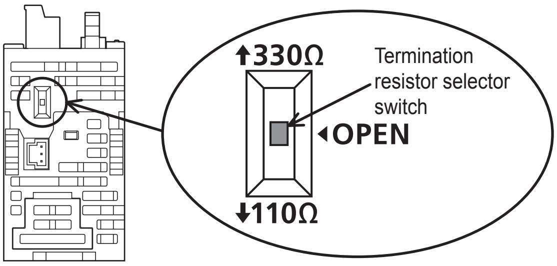
Set the termination resistor selector switch as below.

Wiring	Termination resistor selector switch
Two-pair wiring	330 Ω
One-pair wiring	110 Ω

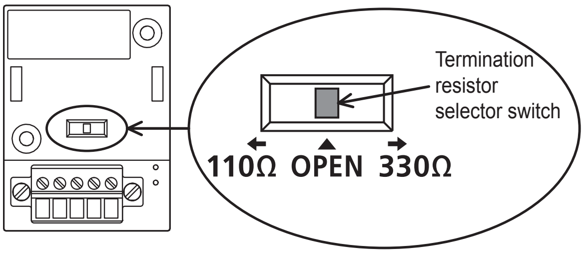
■FX5U CPU module built-in RS-485 port



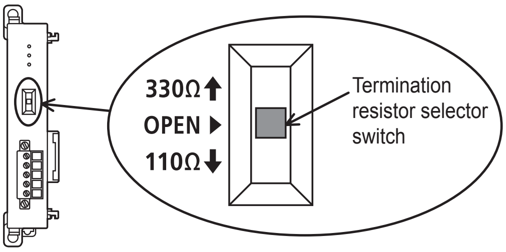
■FX5UC CPU module built-in RS-485 port



■FX5-485-BD



■FX5-485ADP

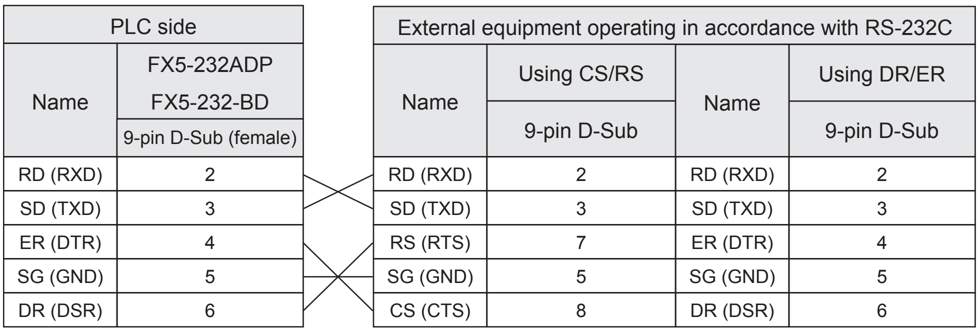


Connection diagram

RS-232C

Representative wiring examples are shown in this section. When pin numbers in the counterpart equipment are different, wire the pins as shown below.

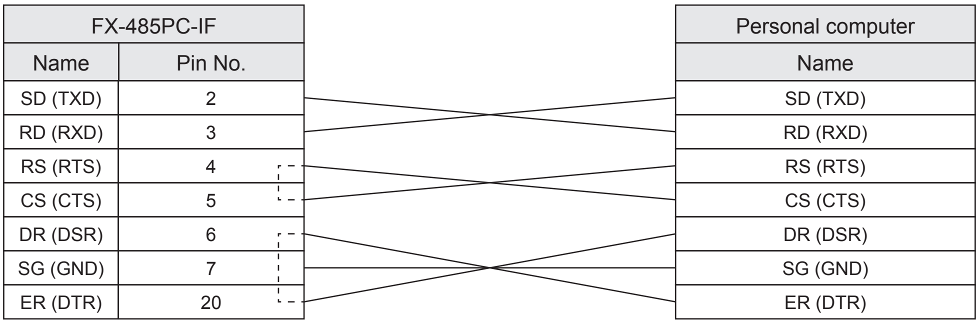
■Connection diagram between FX5 and personal computer



PLC side	
Name	FX5-232ADP FX5-232-BD 9-pin D-Sub (female)
RD (RXD)	2
SD (TXD)	3
ER (DTR)	4
SG (GND)	5
DR (DSR)	6

External equipment operating in accordance with RS-232C			
Name	Using CS/RS 9-pin D-Sub	Name	Using DR/ER 9-pin D-Sub
RD (RXD)	2	RD (RXD)	2
SD (TXD)	3	SD (TXD)	3
RS (RTS)	7	ER (DTR)	4
SG (GND)	5	SG (GND)	5
CS (CTS)	8	DR (DSR)	6

■Connection diagram between FX-485PC-IF and the personal computer

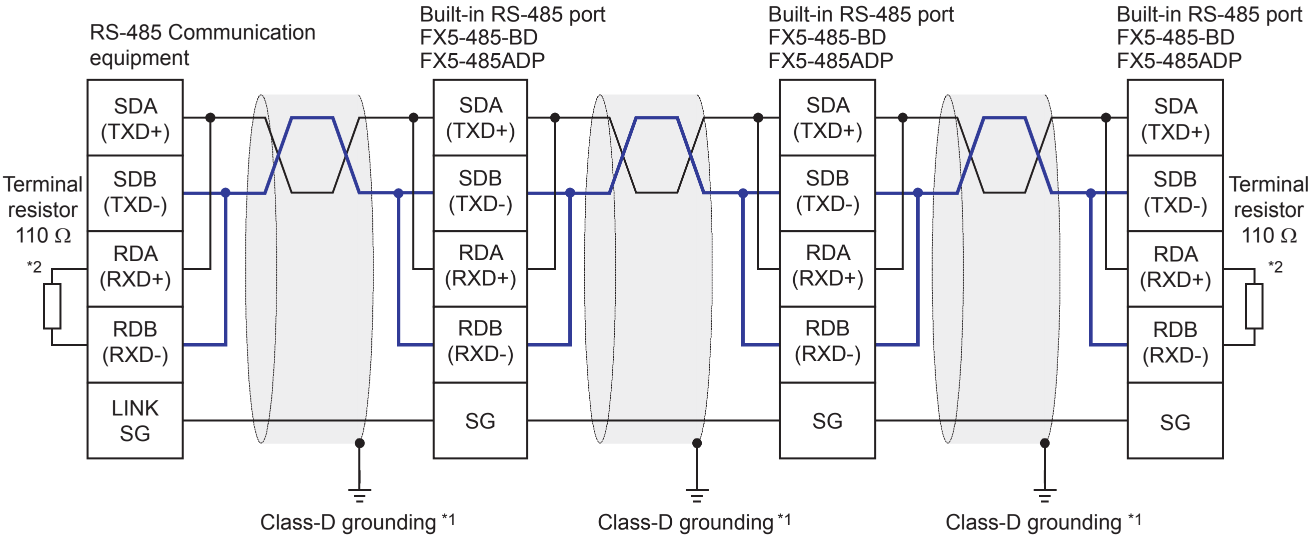


FX-485PC-IF	
Name	Pin No.
SD (TXD)	2
RD (RXD)	3
RS (RTS)	4
CS (CTS)	5
DR (DSR)	6
SG (GND)	7
ER (DTR)	20

Personal computer
Name
SD (TXD)
RD (RXD)
RS (RTS)
CS (CTS)
DR (DSR)
SG (GND)
ER (DTR)

RS-485/RS-422

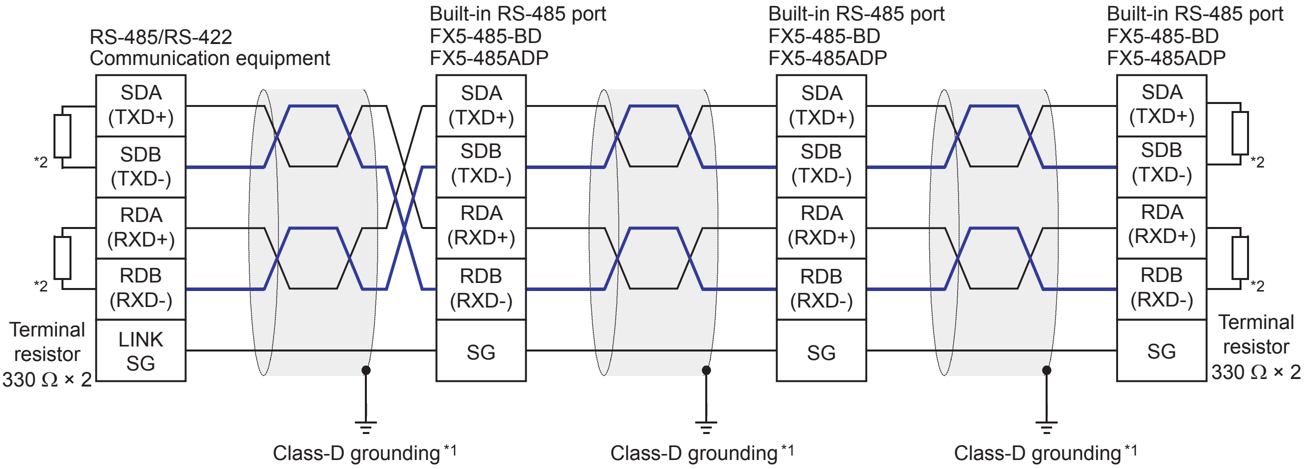
■One-pair wiring



[1] Make sure to perform Class-D grounding on the shield of the twisted pair cable to be connected.

[2] Make sure to provide a termination resistor at both ends of the wire. The built-in RS-485 port, FX5-485-BD and FX5-485ADP have built-in termination resistors. Set the termination resistor selector switch to 110 Ω.

■Two-pair wiring



*1 Make sure to perform Class-D grounding on the shield of the twisted pair cable to be connected.

*2 Make sure to provide a terminal resistor at the end of each line. Set built-in RS-485 port, FX5-485-BD, and FX5-485ADP to 330 Ω depending on the terminal resistor selector switch.

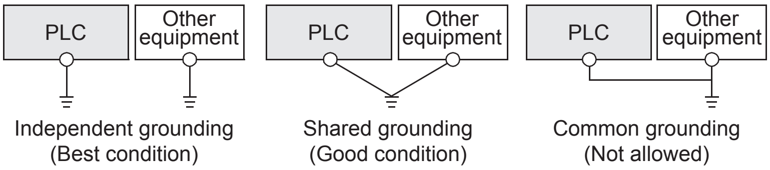
Grounding

Grounding should be performed as stated below.

- Perform D class grounding. (Grounding resistance: 100Ω or less)
- Independent grounding should be performed for best results.

If the PLC cannot be grounded independently, perform the "Shared grounding" shown below.

For details, refer to User's Manual (Hardware) of the CPU module used.



- The grounding wire size should be 14 AWG (2mm²) or larger.
- Bring the grounding point close to the PLC as much as possible so that the ground cable can be shortened.

18.6 Communication Setting

For the FX5 communication settings of this function, parameters are set using GX Works3. For details about GX Works3, refer to 📖GX Works3 Operating Manual.

Setting of parameter differs according to the module used. The procedure for each module is as follows.

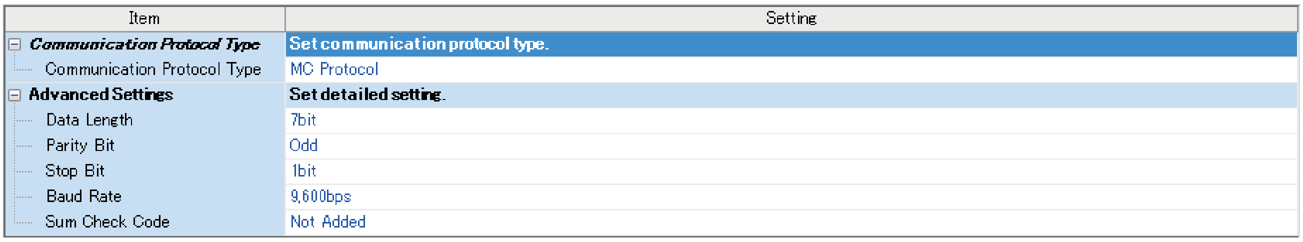
Built-in RS-485 port (CH1)

Navigation Window ⇨ Parameter ⇨ FX5UCPU ⇨ Module Parameter ⇨ 485 Serial Port

Window

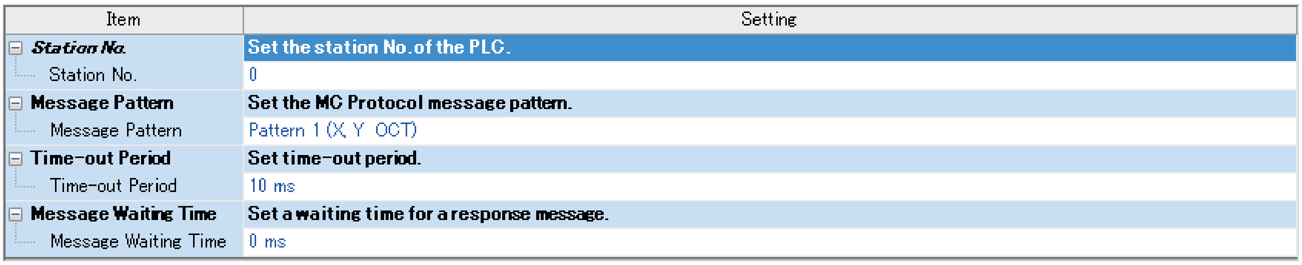
The following screen will be displayed if [MC Protocol] is set for the communication protocol type.

■Basic Settings



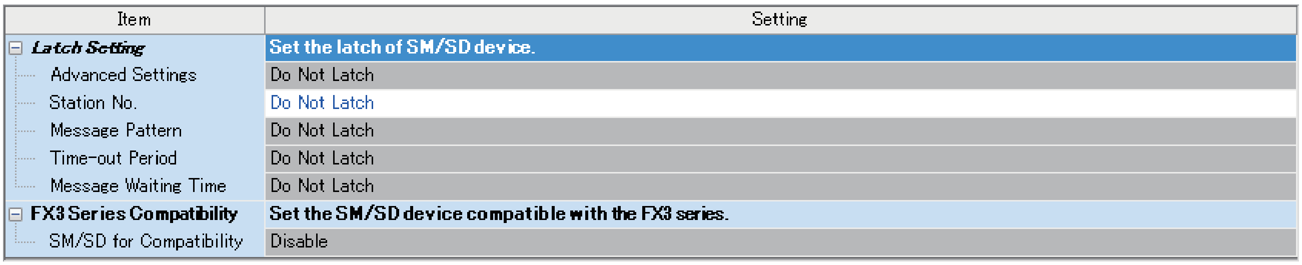
Item	Setting
⊟ ***Communication Protocol Type***	Set communication protocol type.
Communication Protocol Type	MC Protocol
⊟ **Advanced Settings**	**Set detailed setting.**
Data Length	7bit
Parity Bit	Odd
Stop Bit	1bit
Baud Rate	9,600bps
Sum Check Code	Not Added

■Fixed Setting



Item	Setting
⊟ ***Station No.***	Set the station No. of the PLC.
Station No.	0
⊟ **Message Pattern**	**Set the MC Protocol message pattern.**
Message Pattern	Pattern 1 (X, Y OCT)
⊟ **Time-out Period**	**Set time-out period.**
Time-out Period	10 ms
⊟ **Message Waiting Time**	**Set a waiting time for a response message.**
Message Waiting Time	0 ms

■SM/SD Setting



Item	Setting
⊟ ***Latch Setting***	Set the latch of SM/SD device.
Advanced Settings	Do Not Latch
Station No.	Do Not Latch
Message Pattern	Do Not Latch
Time-out Period	Do Not Latch
Message Waiting Time	Do Not Latch
⊟ **FX3 Series Compatibility**	**Set the SM/SD device compatible with the FX3 series.**
SM/SD for Compatibility	Disable

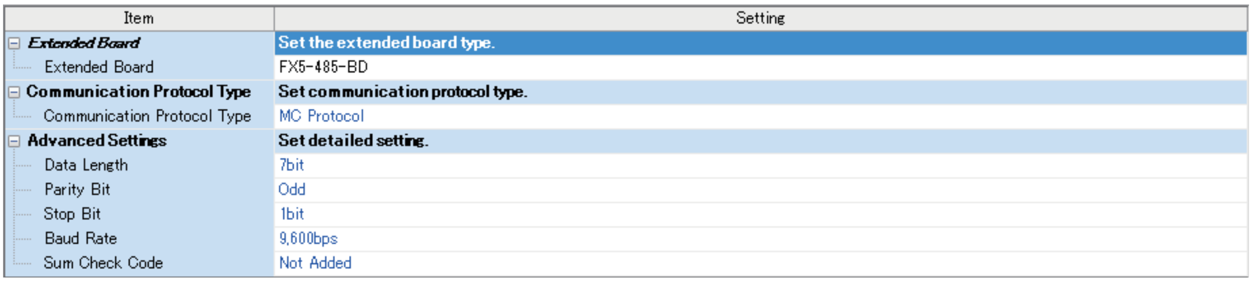
Communication board (CH2)

Navigation Window ⇨ Parameter ⇨ Model name ⇨ Module Parameter ⇨ Extended Board

Window

The following screen will be displayed if [FX5-232-BD], [FX5-485-BD], or [FX5-422-BD-GOT] is set for [Extended Board], and [MC Protocol] is set for [Communication Protocol Type]. Fixed settings and SM/SD settings are the same as for the built-in RS-485 port (CH1).

■Basic Settings



Item	Setting
⊟ ***Extended Board***	**Set the extended board type.**
Extended Board	FX5-485-BD
⊟ **Communication Protocol Type**	**Set communication protocol type.**
Communication Protocol Type	MC Protocol
⊟ **Advanced Settings**	**Set detailed setting.**
Data Length	7bit
Parity Bit	Odd
Stop Bit	1bit
Baud Rate	9,600bps
Sum Check Code	Not Added

Communication adapter (CH3/CH4)

When an expansion adapter is used, add expansion adapter to Module Information.

Navigation window ⇨ Parameter ⇨ Module Information ⇨ Right-click ⇨ Add New Module

After adding the expansion adapter, make settings on the screen displayed from the following operation.

Navigation window ⇨ Parameter ⇨ Module Information ⇨ ADP1 to ADP6 (Communication adapter) ⇨ Module Parameter

Window

Each setting screen is the same as for the built-in RS-485 port (CH1).

Parameter setting details

Set the following items in the channel that uses MC protocol.

Items			Setting value	Reference section
Basic Settings	Extended Board*1		When using this function, select [FX5-232-BD], [FX5-485-BD], or [FX5-422-BD-GOT].	—
	Protocol type		When using this function, select [MC Protocol].	
	Advanced Settings	Data Length	7bit/8bit	
		Parity Bit	None/Odd/Even	
		Stop Bit	1bit/2bit	
		Baud Rate	300bps/600bps/1200bps/2400bps/4800bps/9600bps/19200bps/38400bps/57600bps/115200bps	
		Sum Check Code	Added/Not Added	
Fixed Setting	Station Number		0 to 15	—
	Message system		Pattern 1 (X, Y OCT)/ Pattern 1 (X, Y HEX)*2/ Pattern 4 (X, Y OCT)/ Pattern 4 (X, Y HEX)*2/ Pattern 5*3	
	Time-out Period		1 to 32767 (ms)	
	Message Waiting Time*4		0 to 150 (ms)*5	
SM/SD Setting	Latch Setting	Station No.	Latch/Do Not Latch	☞ Page 322
	FX3 Series Compatibility*6	SM/SD for Compatibility	Disable/Enable	—

*1 Only for the communication board (CH2).
*2 For supported versions of pattern 1 (X, Y HEX) and pattern 4 (X, Y HEX), refer to ☞ Page 936 Added and Changed Functions.
*3 For 1C frame, pattern 5 cannot be used.
*4 For 1C frame, the setting of message waiting time becomes invalid.
*5 The number of the setting must be a multiple of 10.
*6 FX3 Series Compatibility is valid only for 1C frame.

The following settings are unnecessary (fixed values).

Item	Description
Start bit	1 bit
Header	Not added
Terminator	Not added
Control mode	—
Control procedure	None

Latch Setting

Set the necessity of the corresponding SD (special register) latch.

Descriptions	Setting range	Compatible devices
Station number settings	Latch/Do Not Latch	SM8740, SM8750, SM8760, SM8770

18.7 MC Protocol Command

Command list

The following commands can be executed in MC protocol.

1C frame

For details, refer to the following.

☞ Page 774 Command and Function Lists for 1C Frame

3C/4C frame

For details, refer to the following.

☞ Page 710 3C/4C frame

Applicable device

The table below shows devices and device number range that can handled in commands used in communication by MC protocol.

Specify devices and device number range that are there in the targeted unit for performing data reading, writing etc.

1C frame

For details, refer to the following.

☞ Page 776 Accessible device range

3C/4C frame

For details, refer to the following.

☞ Page 716 3C/4C frame

18.8 Related Devices

This section describes the special relay/special register functions used in the MC protocol function.

Point

Available communication channels vary depending on the CPU module and system configuration.
For communication channels, refer to ☞ Page 308 System Configuration.
"FX3 Series compatible" devices operate only on the communication channel specified in the compatible SM/SD for communication settings.
For compatible SM/SD, refer to Page 320 Communication Setting.

List of related devices

Special relay

■FX5 only

R: Read only

Device No.				Name	Description	R/W
CH1	CH2	CH3	CH4			
SM8500	SM8510	SM8520	SM8530	Serial communication error	Turns ON when an error occurs in serial communication.	R
SM8503	SM8513	SM8523	SM8533	Absence/presence of MC protocol	Turns ON when MC protocol is set for serial communication.	R
SM8680	SM8690	SM8700	SM8710	Global ON	Turns ON/OFF when the global command is received.	R
SM8740	SM8750	SM8760	SM8770	Station number setting latch setting valid flag	Turns ON when the SD latch setting of station number setting is valid. ON: SD latch setting of station number setting is valid OFF: SD latch setting of station number setting is invalid	R

■FX3 Series compatible

R: Read only

Device No.		Name	Descriptions	R/W
CH1	CH2			
SM8063	SM8438	Serial communication error	Turns ON when an error occurs in serial communication.	R
SM8126	SM8426	Global ON	Turns ON/OFF when the global command is received in serial communication.	R
SM8419	SM8439	Absence/presence of MC protocol	Turns ON when MC protocol is set for serial communication.	R

Special register

■FX5 only

R: Read only

Device No.				Name	Description	R/W
CH1	CH2	CH3	CH4			
SD8500	SD8510	SD8520	SD8530	Serial communication error code	Stores the error code when the serial communication error occurs.	R
SD8502	SD8512	SD8522	SD8532	Serial communication settings	Stores the setting of the communication parameter.	R
SD8503	SD8513	SD8523	SD8533	Serial communication operation mode	Stores the communication type being used.	R
SD8740	SD8750	SD8760	SD8770	Station number settings[1]	Sets the station number of a PLC.	[2]
SD8741	SD8751	SD8761	SD8771	Message system	Stores the setting of the message system.	R
SD8742	SD8752	SD8762	SD8772	Time-out time setting	These devices set the evaluation time for error when receiving of data from the counterpart equipment is interrupted.	R

[1] When SD latch setting is set to "latch", device value can be changed in the program or an engineering tool, and by turning power supply ON→OFF or by reset, the corresponding function can be operated with the changed value.

[2] With latch settings: Read/write
No latch settings: Read only

■FX3 Series compatible

R: Read only

Device No.		Name	Descriptions	R/W
CH1	CH2			
SD8063	SD8438	Serial communication error code	Stores the error code when a serial communication error occurs.	R
SD8419	SD8439	Serial communication operation mode	Stores status of the current communication being executed.	R

Details of related devices

Serial communication error

Turns ON when an error occurs in serial communication. These flags are for check of the serial communication error.

R: Read only

FX5 only				FX3 Series compatible		Description	R/W
CH1	CH2	CH3	CH4	CH1	CH2		
SM8500	SM8510	SM8520	SM8530	SM8063	SM8438	Turns ON when an error occurs in serial communication. Devices do not go OFF even when normal communication is restored.	R

After the devices above turns ON, the error code is stored in the compatible devices below.

FX5 only				FX3 Series compatible		Name	Description
CH1	CH2	CH3	CH4	CH1	CH2		
SD8500	SD8510	SD8520	SD8530	SD8063	SD8438	Serial communication error code	When a serial communication error occurs, the error code is stored.

Precautions

Do not turn ON or OFF with program or engineering tool.

Serial communication error does not turn OFF even if the communication comes back to normal. The devices turn OFF when power is turned OFF→ON, STOP→RUN, reset or SM50 (Error Detection Reset Completion) is turned ON.

Absence/presence of MC protocol

Turns ON when MC protocol is set for serial communication.

Turns ON by setting MC protocol for the protocol format by the parameters, and by turning power supply ON→OFF or by reset.

R: Read only

FX5 only				**FX3 Series compatible**		**Description**	**R/W**
CH1	**CH2**	**CH3**	**CH4**	**CH1**	**CH2**		
SM8503	SM8513	SM8523	SM8533	SM8419	SM8439	Turns ON when MC protocol is set for serial communication.	R

Global ON

Turns ON/OFF when the global command is received.

R: Read only

FX5 only				**FX3 Series compatible**		**Description**	**R/W**
CH1	**CH2**	**CH3**	**CH4**	**CH1**	**CH2**		
SM8680	SM8690	SM8700	SM8710	SM8126	SM8426	Turns ON/OFF when the global command is received.	R

Station number setting latch setting valid flag

When latch for the station number setting set in the parameter is "latch", these flags turn ON when power supply OFF→ON or when system is reset. In that case, the station number set in SD8740, SD8750, SD8760, SD8770 turns valid.

When latch for the station number setting set in the parameter is "no latch", these flags turn OFF when power supply OFF→ON or when system is reset. In that case, a station number set in the parameter turns valid.

In addition, this device holds its status even when the power is turned OFF→ON or reset.

R: Read only

CH1	**CH2**	**CH3**	**CH4**	**Description**	**R/W**
SM8740	SM8750	SM8760	SM8770	Turned ON when with SD latch setting of station number setting is valid.	R

Precautions

Do not turn ON or OFF with program or engineering tool.

Serial communication error code

This device stores the error code for serial communication (☞ Page 836 Checking error codes of MC protocol function).

R: Read only

FX5 only				**FX3 Series compatible**		**Description**	**R/W**
CH1	**CH2**	**CH3**	**CH4**	**CH1**	**CH2**		
SD8500	SD8510	SD8520	SD8530	SD8063	SD8438	When a serial communication error occurs, the error code is stored.	R

Precautions

Do not change the value with program or engineering tool.

Serial communication errors are not cleared even after communication has recovered its normal status.

Devices are cleared when power is turned OFF→ON, STOP→RUN, reset or SM50 (Error Detection Reset Completion) is turned ON.

Serial communication settings

Stores the set communication parameter in the communication settings when turning OFF→ON the PLC power or resetting the system. (☞ Page 320 Communication Setting)

R: Read only

CH1	CH2	CH3	CH4	Description	R/W
SD8502	SD8512	SD8522	SD8532	Stores the setting of the communication parameter.	R

The descriptions of the communication parameters are as follows.

Bit No.	Name	Description	
		0 (bit is OFF)	1 (bit is ON)
b0	Data length	7 bits	8 bits
b1 b2	Parity bit	b2, b1 (0, 0): N/A (0, 1): Odd (1, 1): Even	
b3	Stop bit	1 bit	2 bits
b4 b5 b6 b7	Baud rate	b7, b6, b5, b4 (0, 0, 1, 1): 300bps (0, 1, 0, 0): 600bps (0, 1, 0, 1): 1200bps (0, 1, 1, 0): 2400bps (0, 1, 1, 1): 4800bps (1, 0, 0, 0): 9600bps (1, 0, 0, 1): 19200bps (1, 0, 1, 0): 38400bps (1, 0, 1, 1): 57600bps (1, 1, 0, 0): 76800bps (1, 1, 0, 1): 115200bps	
b13	Sum check	Not added	Added

Precautions

Do not change the value with program or engineering tool.

Serial communication operation mode

Stores the communication function code that the serial communication is being used.

R: Read only

FX5 only				FX3 Series compatible		Description	R/W
CH1	CH2	CH3	CH4	CH1	CH2		
SD8503	SD8513	SD8523	SD8533	SD8419	SD8439	0: MELSOFT connection or MC protocol 3: N:N Network Communication 5: Non-Protocol Communication 6: Parallel Link Communication 7: Inverter Communication 9: MODBUS RTU Communication 12: Predefined protocol support Other than above: Not used	R

Precautions

Do not change the value with program or engineering tool.

Station number settings

When SD latch setting is set to "no latch", a station number of a local station that is set in the engineering tool and used in MC protocol (station numbers 0 to 15 (00H to 0FH)) is stored when the power supply is turned OFF→ON or reset.
When SD latch setting is set to "latch", the contents of station number settings can be changed by program or engineering tool, and by turning power supply ON→OFF or by reset, it can be operated with the station number stored in the station number settings.

CH1	CH2	CH3	CH4	Description	R/W
SD8740	SD8750	SD8760	SD8770	Stores the station number of a PLC.	[1]

[1] With latch settings: Read/write
No latch settings: Read only

Precautions

Do not change the value with program or engineering tool.

Message system

Stores the settings of the message system (format 1, 4, 5) that is set in the parameter.
R: Read only

CH1	CH2	CH3	CH4	Description	R/W
SD8741	SD8751	SD8761	SD8771	0: Format 1 (X, Y OCT) 3: Format 4 (X, Y OCT) 4: Format 5 5: Format 1 (X, Y HEX)[1] 6: Format 4 (X, Y HEX)[1]	R

[1] For supported versions of pattern 1 (X, Y HEX) and pattern 4 (X, Y HEX), refer to ☞ Page 936 Added and Changed Functions.

Precautions

Do not change the value with program or engineering tool.

Time out judge time

Stores the judgment time until the error when a data reception is stopped from the counterpart equipment, set in the parameter when the power supply is turned OFF→ON or reset.
R: Read only

CH1	CH2	CH3	CH4	Description	R/W
SD8742	SD8752	SD8762	SD8772	1 to 32767 (ms) When "0" or negative value is set for the setting value, the time-out time is set to default value. ☞ Page 328 Default value of the time-out time	R

■Default value of the time-out time

Baud rate (bps)	Default value (ms)	Baud rate (bps)	Default value (ms)
300	50	9600	10
600	30	19200	10
1200	20	38400	10
2400	10	57600	10
4800	10	115200	10

Precautions

Set a longer time-out time to the parameter than the required time to receive one character for the baud rate being used.
Do not change the value with program or engineering tool.

19 INVERTER COMMUNICATION

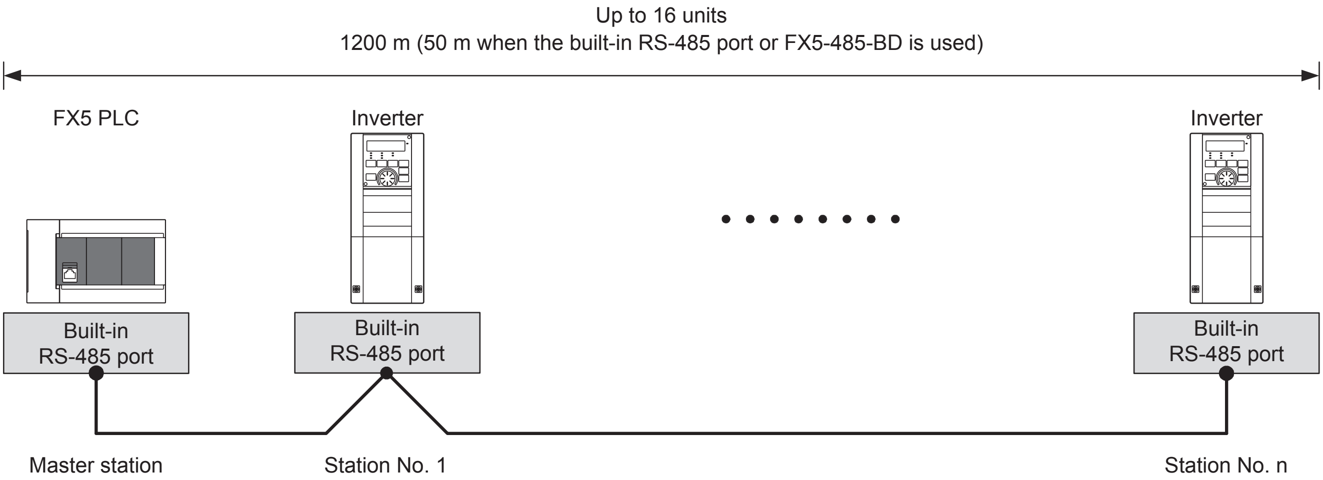
This chapter explains inverter communication.

19.1 Function Outline

Inverter communication function allows connection between an FX5 PLC and up to 16 inverters through RS-485 communication, and monitors operations of inverters, gives various commands to inverters and reads and writes inverter parameters.

- Mitsubishi general-purpose inverters FREQROL-F800/E800/A800/A800 Plus/F700PJ/F700P/A700/E700/E700EX (sensor less servo)/D700/V500 Series can be linked.
- Inverter operations can be monitored, various commands can be given, and parameters can be read or written.
- The overall distance is 1200 m maximum. (Only when configured by FX5-485ADP)

System



19.2 Procedure Before Operation

The flow chart below shows the Inverter Communication setting and sequence programs creation procedure until data link:

1. Check communication specifications

For applicable inverter, operation commands and parameters, and execution time of inverter communication, refer to ☞ Page 334 Specifications.

2. System configuration and selection

For system configuration, and select communication equipment, refer to ☞ Page 330 System Configuration.

3. Wiring

For selection of cables and connection equipment, and wiring example, refer to ☞ Page 342 Wiring.

4. Communication settings[*1]

For communication settings of inverter, refer to ☞ Page 356 Inverter Communication Settings.

For communication settings of communication device, refer to ☞ Page 364 PLC Communication Settings.

5. Program creation

For detailed explanation of related devices, and basic program, refer to ☞ Page 366 Programming.

[*1] For details on operating procedure of GX Works3, refer to the manual below.
📖 GX Works3 Operating Manual

19.3 System Configuration

This section outlines the system configuration required to use inverter communication.

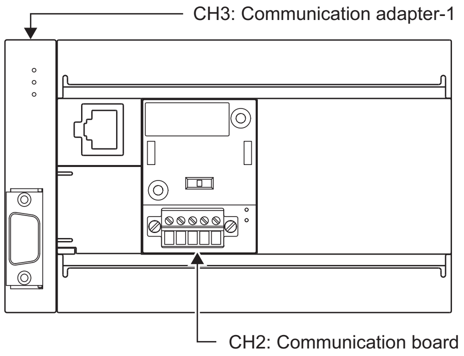
FX5S CPU module

A serial port of up to 2 channels can be connected in the FX5S CPU module by using a communication board and communication adapter.

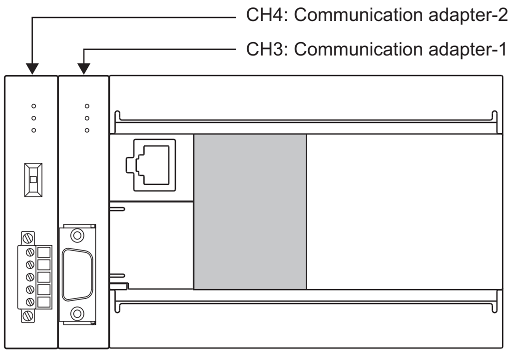
Communication channel assignments are fixed regardless of the system configuration.

The combinations available for the system configurations are shown below.

[Configuration example 1]



[Configuration example 2]



Item		Serial port	Important points in selection	Overall distance
Communication board	FX5-485-BD	CH2	Because the board can be mounted on top of the CPU module, there is no change in the installation space to be required.	50 m or less
Communication adapter	FX5-485ADP	CH3, CH4[1]	Mount the communication adapter to the left of the CPU module.	1200 m or less

[1] The adapters are assigned to CH3 and CH4 in the order, from the closest one to the CPU module.

Precautions

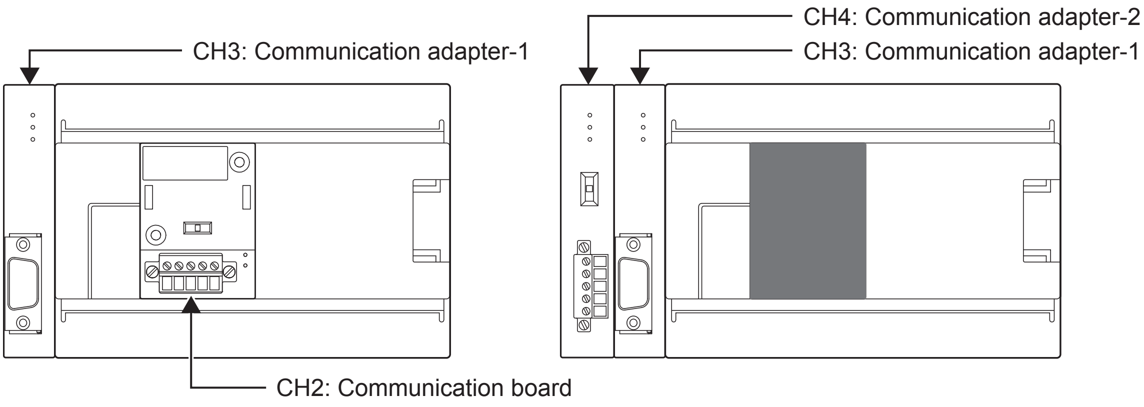
Total number of communication boards and communication adapters that can be connected is two.

FX5UJ CPU module

A serial port of up to 2 channels can be connected in the FX5UJ CPU module by using a communication board and communication adapter.

Communication channel assignments are fixed regardless of the system configuration.

The combinations which can be configured are shown below.



[Configuration example 1]

[Configuration example 2]

Item		Serial port	Important points in selection	Overall distance
Communication board	FX5-485-BD	CH2	Mounted on top of the CPU module, there is no change in the installation space requirements.	50 m or less
Communication adapter	FX5-485ADP	CH3, CH4[*1]	Mount the communication adapter to the left of the CPU module.	1200 m or less

[*1] The adapters are assigned to CH3 and CH4 in the order, from the closest one to the CPU module.

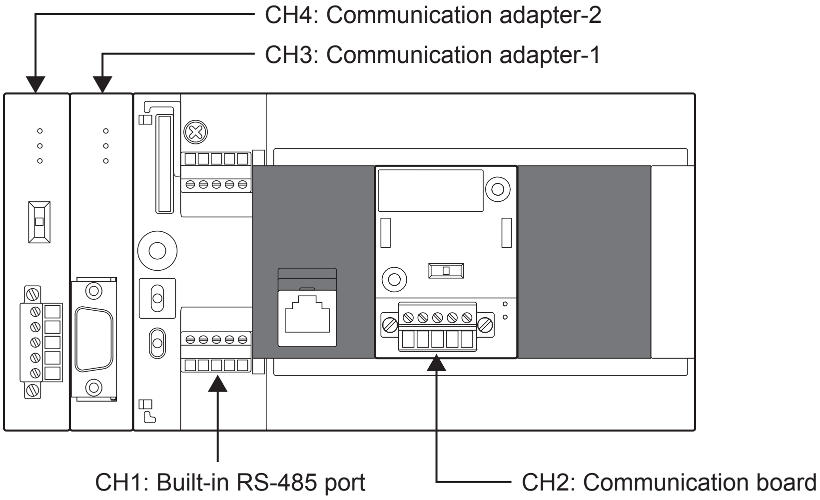
Precautions

Total number of communication boards and communication adapters that can be connected is two.

FX5U CPU module

A serial port of up to 4 channels can be connected in the FX5U CPU module by using the built-in RS-485 port, communication board, and communication adapter.

Communication channel assignments are fixed regardless of the system configuration.



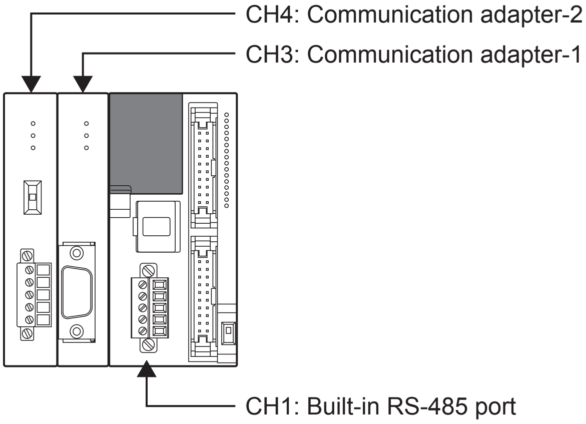
Item		Serial port	Important points in selection	Overall distance
Built-in RS-485 port		CH1	Since it is built-into the CPU module, there is no need to add equipment.	50 m or less
Communication board	FX5-485-BD	CH2	Mounted on top of the CPU module, there is no change in the installation space requirements.	50 m or less
Communication adapter	FX5-485ADP	CH3, CH4[*1]	Mount the communication adapter to the left of the CPU module.	1200 m or less

*1 The adapters are assigned to CH3 and CH4 in the order, from the closest one to the CPU module.

FX5UC CPU module

A serial port of up to 3 channels can be connected in the FX5UC CPU module by using the built-in RS-485 port and communication adapter.

Communication channel assignments are fixed regardless of the system configuration.



Item		Serial port	Important points in selection	Overall distance
Built-in RS-485 port		CH1	Since it is built-into the CPU module, there is no need to add equipment.	50 m or less
Communication adapter	FX5-485ADP	CH3, CH4[1]	Mount the communication adapter to the left of the CPU module.	1200 m or less

[1] The adapters are assigned to CH3 and CH4 in the order, from the closest one to the CPU module.

19.4 Specifications

This section describes the communication specifications and performance of the inverter communication function.

Communication specifications

Items		Specifications	Remarks
Number of connectable modules		Maximum of 16	—
Transmission standard		RS-485 standard	—
Maximum overall distance		When using FX5-485ADP: 1200m or less When using built-in RS-485 port or FX5-485-BD: 50m or less	—
Protocol type		Inverter computer link	—
Control procedure		Asynchronous system	—
Communication method		Half-duplex, bi-directional communication	—
Baud rate		4800/9600/19200/38400/57600/115200(bps)	Only FREQROL-F800/A800 supports 57600 bps or more
Character format	—	ASCII	—
	Start bit	1 bit	—
	Data length	7 bits/8 bits	—
	Parity bit	None, odd or even	—
	Stop Bit	1 bit/2 bits	—

Applicable inverter

Series	Connection destination
FREQROL-F800 Series	Built-in RS-485 terminal
FREQROL-E800 Series	Built-in PU connector
FREQROL-A800 Series	Built-in RS-485 terminal
FREQROL-A800 Plus Series	Built-in RS-485 terminal
FREQROL-F700PJ Series	Built-in PU connector
FREQROL-F700P Series	Built-in RS-485 terminal
FREQROL-A700 Series	Built-in RS-485 terminal
FREQROL-E700 Series	Built-in PU connector, FR-E7TR (optional)
FREQROL-E700EX Series	Built-in PU connector
FREQROL-D700 Series	Built-in PU connector
FREQROL-V500 Series	Built-in PU connector, FR-A5NR (optional)

Inverter instruction codes and parameters

The tables below show the inverter instruction codes and the parameters that can be communicated.

Inverter operation monitoring

The table below shows instruction codes for reading the inverter which can be specified by IVCK instruction (☞ Page 381 Inverter operation monitoring instruction) operand (s2) and their contents. Any instruction codes not shown in the table below may cause communication errors. Use the instruction codes shown below only.

For the instruction codes, refer to the pages explaining computer link in detail in each inverter manual.

Inverter instruction code (hexadecimal)	**Read contents**	**Applicable inverter**										
		F800	**E800**	**A800**	**A800 Plus**	**F700PJ**	**F700P**	**A700**	**E700**	**E700EX**	**D700**	**V500**
H7B	Operation mode	○	○	○	○	○	○	○	○	○	○	○
H6F	Output frequency/ speed	○	○	○	○	○	○	○	○	○	○	○[*1]
H70	Output current	○	○	○	○	○	○	○	○	○	○	○
H71	Output voltage	○	○	○	○	○	○	○	○	○	○	○
H72	Special monitor	○	○	○	○	○	○	○	○	○	○	○
H73	Special monitor selection No.	○	○	○	○	○	○	○	○	○	○	○
H74	Error contents	○	○	○	○	○	○	○	○	○	○	○
H75												
H76												
H77												
H79	Inverter status monitor (expansion)	○	○	○	○	○	○	○	○	○	○	—
H7A	Inverter status monitor	○	○	○	○	○	○	○	○	○	○	○
H6D	Set frequency read (RAM)	○	○	○	○	○	○	○	○	○	○	○[*1]
H6E	Set frequency read (EEPROM)	○	○	○	○	○	○	○	○	○	○	○[*1]
H7F	Link parameter expansion setting	These codes cannot be specified in (s2) in IVCK instruction. They are automatically processed when a "second parameter specification code" is specified in IVRD instruction.										
H6C	Second parameter switching											

[*1] Please set "0" to instruction code HFF (Link parameter expansion setting) just before the IVCK instruction when reading frequency. When "0" is not set, reading of the frequency may not be executed normally.

Inverter operation control

The table below shows instruction codes for writing to the inverter which can be specified by IVDR instruction (☞ Page 383 Inverter operation control instruction) operand (s2) and their contents.

For the instruction codes, refer to the pages explaining computer link in detail in each inverter manual.

Inverter instruction code (hexadecimal)	Write contents	Applicable inverter										
		F800	E800	A800	A800 Plus	F700PJ	F700P	A700	E700	E700EX	D700	V500
HFB	Operation mode	○	○	○	○	○	○	○	○	○	○	○
HF3	Special monitor selection number	○	○	○	○	○	○	○	○	○	○	○
HF9	Run command (expansion)	○	○	○	○	○	○	○	○	○	○	—
HFA	Run command	○	○	○	○	○	○	○	○	○	○	○
HED	Set frequency write (RAM)	○	○	○	○	○	○	○	○	○	○	○[*3]
HEE	Set frequency write (EEPROM)	○	○	○	○	○	○	○	○	○	○	○[*3]
HFD[*1]	Inverter reset[*2]	○	○	○	○	○	○	○	○	○	○	○
HF4	Error contents batch clear	○	○	○	○	○	○	○	○	○	○	○
HFC	All parameter clear	○	○	○	○	○	○	○	○	○	○	○
HFF	Link parameter expansion setting	○	○	○	○	○	○	○	○	○	○	○

*1 The instruction code "HFD (inverter reset)" does not request a response from the inverter. Accordingly, even if inverter reset is executed in a station number to which an inverter is not connected, error does not occur. It takes about 2.2 seconds to complete execution of inverter reset.

*2 When resetting the inverter, please specify H9696 as the operand (s3) of the IVDR instruction.
Do not use H9966.

*3 Please set "0" to instruction code HFF (Link parameter expansion setting) just before the IVDR instruction when writing frequency. When "0" is not set, writing of the frequency may not be executed normally.

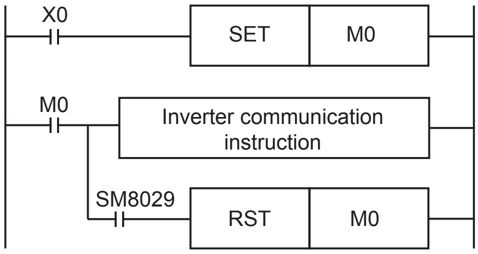
Parameters

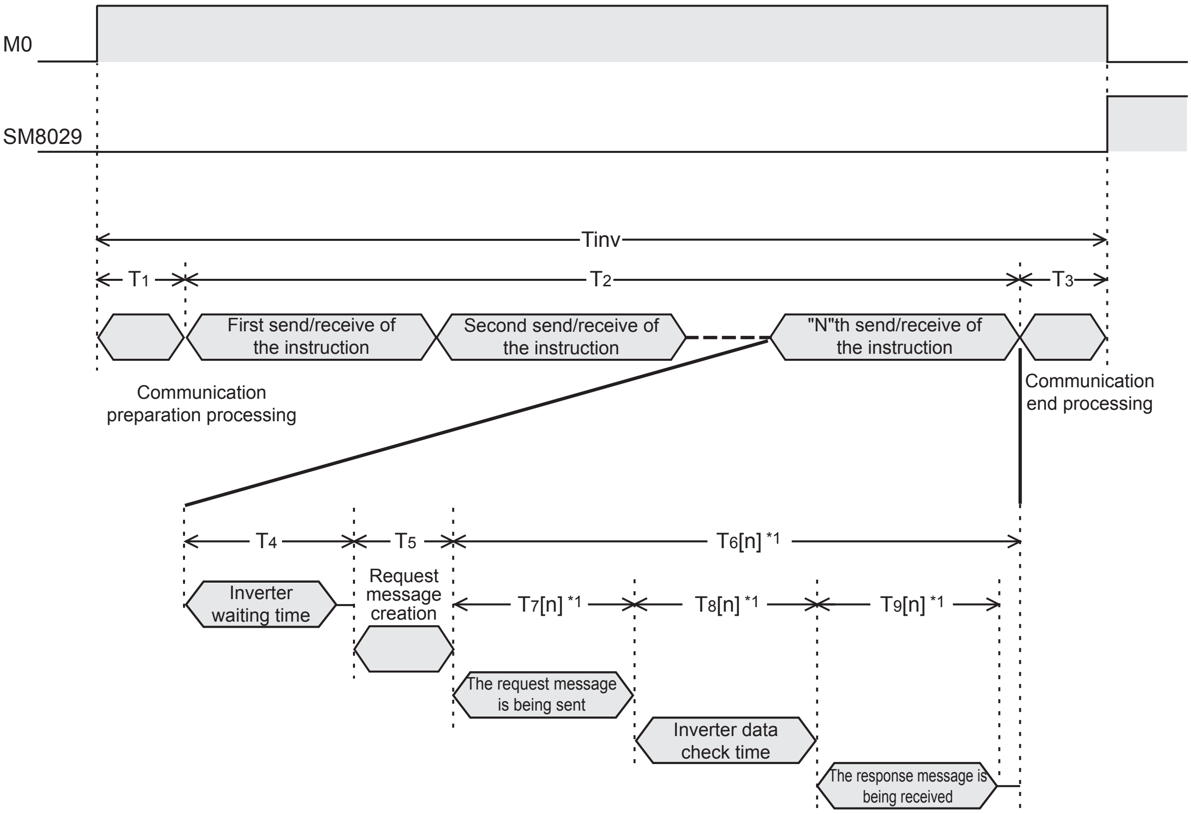
For inverter parameters which can be changed (read/write), refer to the manual for each particular inverter.

Time required for inverter communication instructions

Outline

The period of time after one inverter communication instruction is driven until communication with the inverter is completed (SM8029 turns ON) is considered as the required time for the inverter communication instruction as shown.





*1 The time required for the nth time send/receive of one instruction.

Some inverter communication instructions execute several send/receives. The following table shows the number of send/receives in each instruction.

Inverter communication instruction		Number of send/receives per instruction
IVCK		1
IVDR		1
IVRD	Change of the 2nd parameter is unnecessary.	2
	Change of the 2nd parameter is necessary.*2	3
IVWR	Change of the 2nd parameter is unnecessary.	2
	Change of the 2nd parameter is necessary.*2	3
IVBWR	Number of times of communication (s2)	(s2) × 2 + Number of parameters requiring change of the 2nd parameter*2
IVMC		1

*2 For parameters requiring change of the 2nd parameter, refer to ☞ Page 398 Second parameter specification code.

Calculation method

The required time "Tinv" for inverter communication instruction in units of "ms" is calculated as follows.

"INT(n)" in the calculation formula indicates an integer obtained by truncating decimal places of "n".

■Length of 1 character

In inverter communication, the communication setting is as shown in the following table.

The length of 1 character is as shown in the following expression.

Length of 1 character = Start bit + Data length + Parity bit + Stop bit = 10 [bit]

Name	Setting value	No. of bits
Start bit	—	1
Data length	7	7
Parity bit	Even	1
Stop bit	1	1
Total		10

■Time required for inverter communication instructions

The time required for inverter communication instructions is calculated based on the following expression.

$$\text{Tinv} = T_1 + T_2 + T_3$$

$$T_1 = \text{INT}\left(\frac{1}{\text{Scan time}} + 1\right) \times \text{Scan time [ms]}^{*1}$$

$$T_2 = \{ n \times (T_4 + T_5) \} + \underbrace{\Sigma T_6[n]}_{T_6[1] + T_6[2] + T_6[3] + \cdots} \text{ [ms]} \qquad \text{n: Number of send/receives}$$

$$T_3 = 1 \text{ [ms]}$$

*1 Time required for one instruction.

T_4, T_5 and $T_6[n]$ are calculated as given in the following table.

Communication time	Calculation method
T_4	$= \left(\text{INT}\left(\frac{1}{\text{Scan time}} + 1\right) \times \text{Scan time}\right) + \left(\text{INT}\left(\frac{11}{\text{Scan time}} + 1\right) \times \text{Scan time}\right)$
T_5	$= \text{INT}\left(\frac{1}{\text{Scan time}} + 1\right) \times \text{Scan time}$
$T_6[n]$	• For Scan time $<T_7[n]+T_8[n]+T_9[n]$ $= \left(\text{INT}\left(\frac{T_7[n] + T_8[n] + T_9[n]}{\text{Scan time}}\right) + 1\right) \times \text{Scan time} + \left(\text{INT}\left(\frac{1}{\text{Scan time}} + 1\right) \times \text{Scan time}\right)$
	• For Scan time $\geq T_7[n]+T_8[n]+T_9[n]$ $= \left(\text{INT}\left(\frac{T_7[n] + T_8[n] + T_9[n]}{\text{Scan time}}\right) + 2\right) \times \text{Scan time} + \left(\text{INT}\left(\frac{1}{\text{Scan time}} + 1\right) \times \text{Scan time}\right)$
$T_7[n]+T_9[n]$	$= \left\{\left(\frac{1}{\text{Communication speed [bps]}}\right) \times (\text{No. of characters sent/received}^{*2} \times \text{1-character length})\right\} \times 1000$
$T_8[n]$	= Data check time in inverters[3]

*2 Refer to ☞ Page 339 Number of sent/received characters for the number of sent/received characters.

*3 Refer to ☞ Page 339 Data check time of inverters for the data check time of inverters.

■Number of sent/received characters

Inverter communication instruction	Parameter/instruction code	First			Second			Third		
		Send	Receive	Total	Send	Receive	Total	Send	Receive	Total
IVCK	H73, H7A, H7F, H6C	9	9	18	—	—	—	—	—	—
	Other than above	9	11	20	—	—	—	—	—	—
IVDR	HF3, HFA, HFF	11	4	15	—	—	—	—	—	—
	HFD	13	0[*2]	13	—	—	—	—	—	—
	Other than above	13	4	17	—	—	—	—	—	—
IVRD	Change of the 2nd parameter is unnecessary.	11	4	15	9	11	20	—	—	—
	Change of the 2nd parameter is necessary.	11	4	15	11	4	15	9	11	20
IVWR	Change of the 2nd parameter is unnecessary.	11	4	15	13	4	17	—	—	—
	Change of the 2nd parameter is necessary.	11	4	15	11	4	15	13	4	17
IVBWR[*1]	Change of the 2nd parameter is unnecessary.	11	4	15	13	4	17	—	—	—
	Change of the 2nd parameter is necessary.	11	4	15	11	4	15	13	4	17
IVMC	—	19	19	38	—	—	—	—	—	—

[*1] Indicates the number of sent/received characters required to write 1 parameter. The IVBWR instruction executes parameter writing repeatedly for the number of times of communication (s2).

[*2] There is no response from the inverter because resetting is not complete. The PLC waits for 2.2 s until reset of the inverter is completed, and then completes execution of the inverter communication instruction.

■Data check time of inverters

Make sure to refer to the manual of the inverter used, and confirm the data check time corresponding to the communication command used.

Item	Data check time
Various monitors, operation command, set frequency (RAM)	<12 ms
Parameters read/write, set frequency (EEPROM)	<30 ms
Parameter all clear/all clear	<5 s
Reset instruction	No response (The PLC waits for 2.2s until reset of the inverter is completed, and then completes execution of the inverter communication instruction.)

The IVWR, IVRD and IVBWR instructions automatically change expansion parameters and the 2nd parameter.

The data check time for the last (2nd or 3rd) send/receive in the IVWR and IVRD instructions and the data check time for the last send/receive of each parameter writing in the IVBWR instruction corresponds to parameter reading/writing (<30ms).

The data check time for sending and receiving other than the above (such as expansion parameter change and 2nd parameter change) corresponds to various monitors (<12ms).

Calculation example

This is a calculation example for the following communication settings and scan time when communicating with an inverter.

Communication speed = 19200[bps]

Length of 1 character = 10[bit]

Scan time = 10[ms]

■Calculation example 1

Calculation of required time when Pr. 3 is read by the IVRD instruction

$Tinv=T_1+T_2+T_3=181$[ms]

$T_1=10$[ms], $T_3=1$[ms]

Calculate "T_2" as follows because Pr.3 does not require change of the 2nd parameter.

$$T_2 = \underbrace{2}_{\text{Number of send/receives}} \times (T_4 + T_5) + \underbrace{T_6[1]}_{\text{The first send/receive}} + \underbrace{T_6[2]}_{\text{The second send/receive}} = 2 \times (30 + 10) + 30 + 60 = 170\ [\text{ms}]$$

$$T_4 = (\text{ INT }(\frac{1}{10} +1) \times 10) + (\text{ INT }(\frac{11}{10} +1) \times 10) = 30\ [\text{ms}]$$

$$T_5 = \text{INT }(\frac{1}{10} +1) \times 10 = 10\ [\text{ms}]$$

$$T_6[1] = (\text{ INT }(\frac{T_7[1]+T_8[1]+T_9[1]}{10}) +1) \times 10 + (\text{ INT }(\frac{1}{10} +1) \times 10) = (\text{ INT }(\frac{19.8}{10} +1) \times 10) +10 = 30\ [\text{ms}]$$

$$T_7[1] + T_8[1] + T_9[1] = 7.8 + 12 = 19.8\ [\text{ms}]$$

$$T_7[1] + T_9[1] = ((\frac{1}{19200}) \times (11+4) \times 10) \times 1000 = 7.8\ [\text{ms}]$$

$$T_8[1] = 12\ [\text{ms}]$$

$$T_6[2] = (\text{ INT }(\frac{T_7[2]+T_8[2]+T_9[2]}{10}) +1) \times 10 + (\text{ INT }(\frac{1}{10} +1) \times 10) = (\text{ INT }(\frac{40.4}{10} +1) \times 10) +10 = 60\ [\text{ms}]$$

$$T_7[2] + T_8[2] + T_9[2] = 10.4 + 30 = 40.4\ [\text{ms}]$$

$$T_7[2] + T_9[2] = ((\frac{1}{19200}) \times (9+11) \times 10) \times 1000 = 10.4\ [\text{ms}]$$

$$T_8[2] = 30\ [\text{ms}]$$

$$Tinv = T_1 + T_2 + T_3 = 10 + 170 + 1 = 181\ [\text{ms}]$$

■Calculation example 2

Calculation of required time when Pr.902 is read by the IVRD instruction

$Tinv=T_1+T_2+T_3=251$[ms]

$T_1=10$[ms], $T_3=1$[ms]

Calculate "T_2" as follows because Pr.902 does not require change of the 2nd parameter.

$$T_2 = \underbrace{3}_{\text{Number of send/receives}} \times (T_4+T_5) + \underbrace{T_6[1]}_{\text{The first send/receive}} + \underbrace{T_6[2]}_{\text{The second send/receive}} + \underbrace{T_6[3]}_{\text{The third send/receive}} = 3 \times (30+10) + 30 + 30 + 60 = 240\ [\text{ms}]$$

$$T_4 = (\text{ INT }(\frac{1}{10} +1) \times 10) + (\text{ INT }(\frac{11}{10} +1) \times 10) = 30\ [\text{ms}]$$

$$T_5 = \text{INT }(\frac{1}{10} +1) \times 10 = 10\ [\text{ms}]$$

$$T_6[1] = (\text{ INT }(\frac{T_7[1]+T_8[1]+T_9[1]}{10}) +1) \times 10 + (\text{ INT }(\frac{1}{10} +1) \times 10) = (\text{ INT }(\frac{19.8}{10} +1) \times 10) +10 = 30\ [\text{ms}]$$

$$T_7[1] + T_8[1] + T_9[1] = 7.8 + 12 = 19.8\ [\text{ms}]$$

$$T_7[1] + T_9[1] = ((\frac{1}{19200}) \times (11+4) \times 10) \times 1000 = 7.8\ [\text{ms}]$$

$$T_8[1] = 12\ [\text{ms}]$$

$$T_6[2] = (\text{ INT }(\frac{T_7[2]+T_8[2]+T_9[2]}{10}) +1) \times 10 + (\text{ INT }(\frac{1}{10} +1) \times 10) = (\text{ INT }(\frac{19.8}{10} +1) \times 10) + 10 = 30\ [\text{ms}]$$

$$T_7[2] + T_8[2] + T_9[2] = 7.8 + 12 = 19.8\ [\text{ms}]$$

$$T_7[2] + T_9[2] = ((\frac{1}{19200}) \times (11+4) \times 10) \times 1000 = 7.8\ [\text{ms}]$$

$$T_8[2] = 12\ [\text{ms}]$$

$$T_6[3] = (\text{ INT }(\frac{T_7[3]+T_8[3]+T_9[3]}{10}) +1) \times 10 + (\text{ INT }(\frac{1}{10} +1) \times 10) = (\text{ INT }(\frac{40.4}{10} +1) \times 10 +10 = 60\ [\text{ms}]$$

$$T_7[3] + T_8[3] + T_9[3] = 10.4 + 30 = 40.4\ [\text{ms}]$$

$$T_7[3] + T_9[3] = ((\frac{1}{19200}) \times (9+11) \times 10) \times 1000 = 10.4\ [\text{ms}]$$

$$T_8[3] = 30[\text{ms}]$$

$$Tinv = T_1 + T_2 + T_3 = 10 + 240 + 1 = 251\ [\text{ms}]$$

■Calculation example 3

Calculation of required time when Pr.10 to Pr.14 [(s2) = 5] are written by the IVBWR instruction

$Tinv=T_1+T_2+T_3=851[ms]$

$T_1=10\times(s2)=50[ms]$, $T_3=1[ms]$

Calculate "T_2" as follows because Pr.10 to Pr.14 do not require change of the 2nd parameter and the time required for writing is same in each parameter.

$$T_2 = \underset{\text{Time required to write Pr. 10}}{\underline{(\ 2 \times (T_4+T_5) + T_6[1]+T_6[2]\ )}} + \underset{\text{Time required to write Pr. 11}}{\underline{(\ 2 \times (T_4+T_5) + T_6[3]+T_6[4]\ )}} + \cdots$$

$$= \underset{(s2)}{\underline{5}} \times (\ 2 \times (T_4+T_5) + T_6[1] + T_6[2]\ ) = 5 \times (\ 2 \times (30+10) + 30 + 50\ ) = 800\ [ms]$$

$$T_4 = (\ INT\ (\frac{1}{10} +1\ ) \times 10\ ) + (\ INT\ (\frac{11}{10} +1\ ) \times 10\ ) = 30\ [ms]$$

$$T_5 = INT\ (\frac{1}{10} +1\ ) \times 10 = 10\ [ms]$$

$$T_6[1] = (\ INT\ (\frac{T_7[1]+T_8[1]+T_9[1]}{10})\ +1\ ) \times 10 + (\ INT\ (\frac{1}{10} +1\ ) \times 10\ ) = (\ INT\ (\frac{19.8}{10} +1\ ) \times 10 +10 = 30\ [ms]$$

$$T_7[1] + T_8[1] + T_9[1] = 7.8 + 12 = 19.8\ [ms]$$

$$T_7[1] + T_9[1] = (\ (\frac{1}{19200}\ ) \times (11+4) \times 10\ ) \times 1000 = 7.8\ [ms]$$

$$T_8[1] = 12\ [ms]$$

$$T_6[2] = (\ INT\ (\frac{T_7[2]+T_8[2]+T_9[2]}{10})\ +1\ ) \times 10 + (\ INT\ (\frac{1}{10} +1\ ) \times 10\ ) = (\ INT\ (\frac{38.9}{10} +1\ ) \times 10 +10 = 50\ [ms]$$

$$T_7[2] + T_8[2] + T_9[2] = 8.9 + 30 = 38.9\ [ms]$$

$$T_7[2] + T_9[2] = (\ (\frac{1}{19200}\ ) \times (13+4) \times 10\ ) \times 1000 = 8.9\ [ms]$$

$$T_8[2] = 30\ [ms]$$

$$Tinv = T_1 + T_2 + T_3 = 50 + 800 + 1 = 851\ [ms]$$

19.5 Wiring

This section explains about the wiring.

Wiring procedure

1. Select the connection method

Confirm the inverter connection method. (☞ Page 342 Connection method)

2. Make arrangements for wiring

Prepare cables (☞ Page 346 Cable), distributors (☞ Page 348 Connection devices (RJ45 connector and distributor)) and termination resistors (☞ Page 349 Termination resistor setting) required for wiring.

3. Turn OFF the PLC power

Before wiring, make sure that the PLC power is OFF.

4. Wire the communication equipment.

Connect RS-485 communication equipment of PLC with the serial port of the inverter. (☞ Page 351 Connection diagram)

5. Set or connect termination resistors.

Set or connect termination resistor of the inverter farthest from the PLC. (☞ Page 349 Termination resistor setting)

6. Connect (Ground) a shielding wire (Class-D grounding)

When using a twisted pair cable, connect a shielding wire. (☞ Page 350 Shielded wiring)

Connection method

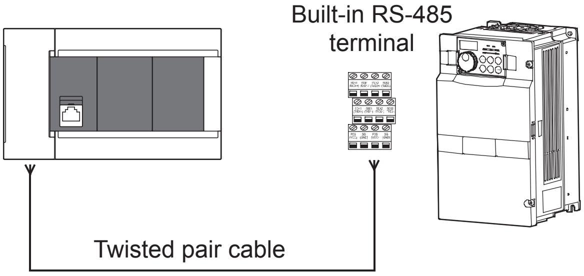
When connecting to RS-485 communication equipment, use the 10BASE-T cable or shielded twisted pair cables, depending on the connection method.

●——	>——	□——
PU connector (RJ45 connector)	Single wire (Use twisted pair cable or 10BASE-T cable for LAN.)	Termination resistor (Which is built in the FX5 PLC, and must be arranged by the user for the inverter, and supplied with or built in for other communication equipment)

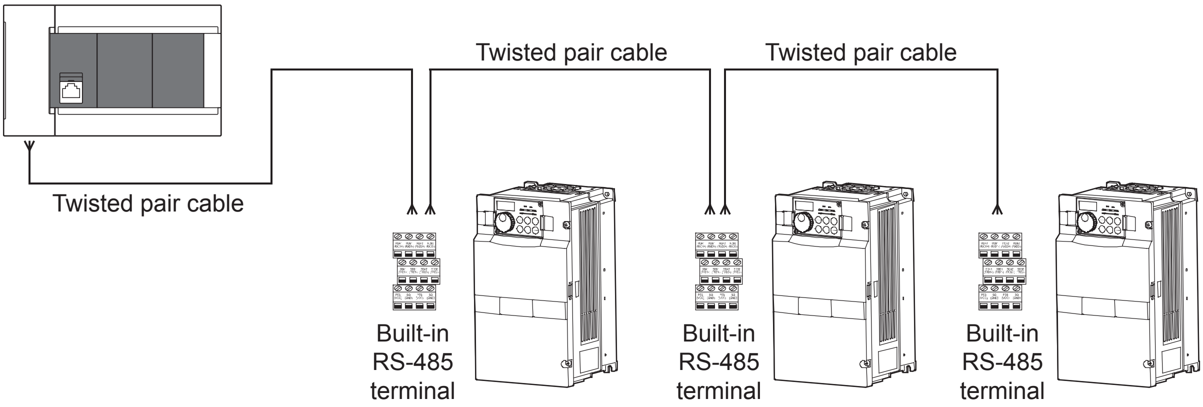
F800/A800/A800 Plus/F700P/A700 series

■Built-in RS-485 terminal

- In the case of 1-to-1 connection



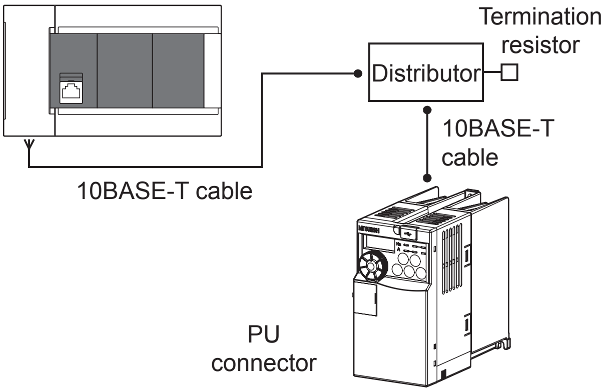
- In the case of 1-to-n connection



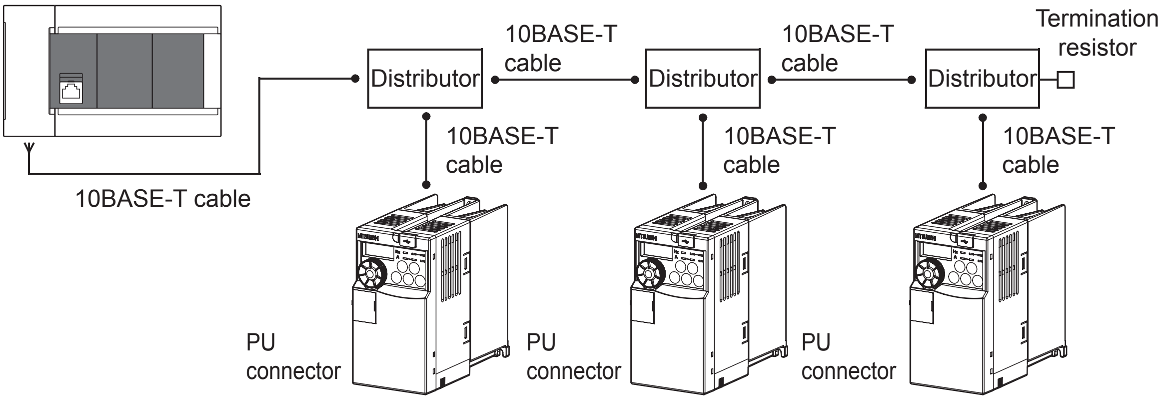
E800/F700PJ/E700/E700EX/D700/V500 series

■PU connector

- In the case of 1-to-1 connection



- In the case of 1-to-n connection



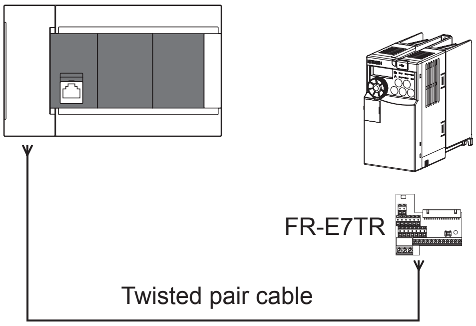
Point

- Because the termination resistor cannot be connected to the inverter, use a distributor.
- Cannot be connected to the built-in Ethernet port of the CPU module.

■FR-E7TR (E700 Series only)

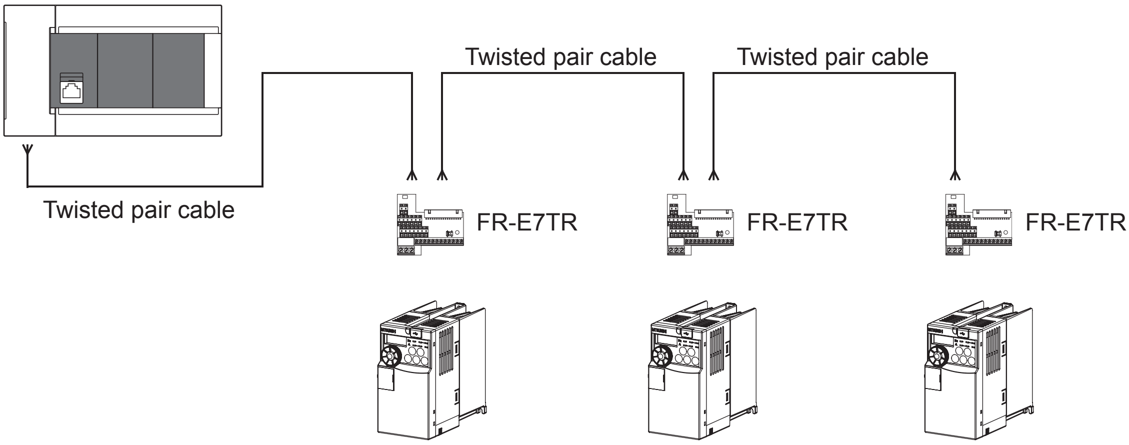
- In the case of 1-to-1 connection

When a distributor is not used



- In the case of 1-to-n connection

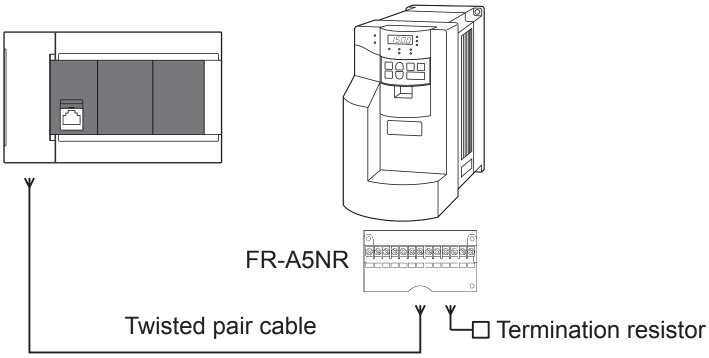
When a distributor is not used



■FR-A5NR (V500 Series only)

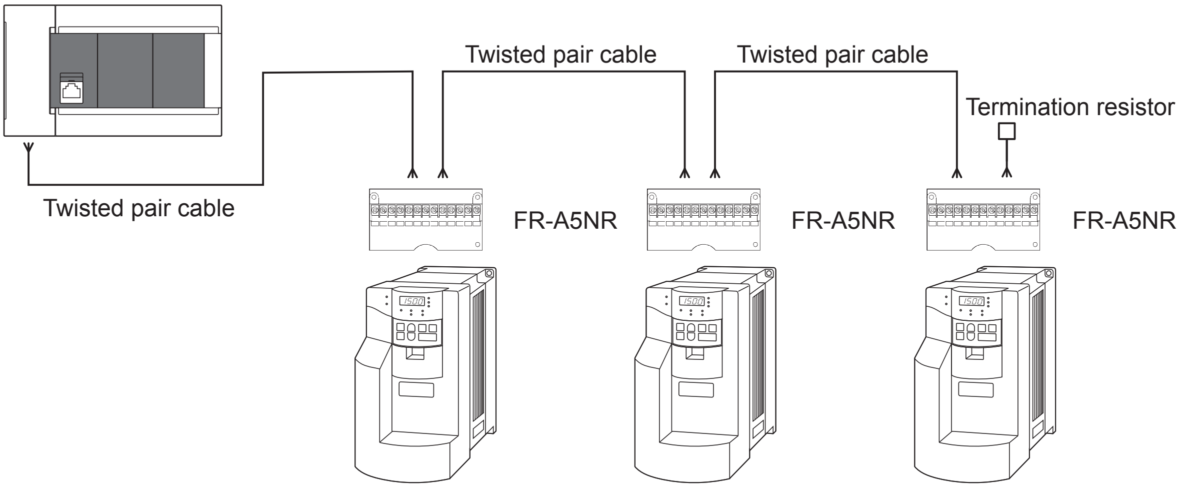
- In the case of 1-to-1 connection

When a distributor is not used



- In the case of 1-to-n connection

When a distributor is not used



Cable

Twisted pair cable

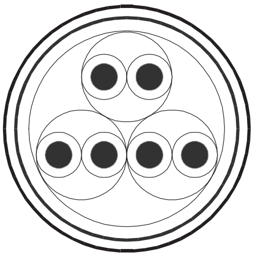
Use commercial three-pair type twisted cables of 0.3 mm² or more.

The specifications of the cables used in wiring are shown.

■RS-485 cable specifications

Items	Description
Cable type	Shielded cable
Number of pairs	3p
Conductor resistance (20℃)	88.0 Ω/km or less
Insulation resistance	10000 MΩ-km or more
Voltage resistance	500VDC, 1 minute
Electrostatic capacitance (1kHz)	60 nF/km or less as an average
Characteristic impedance (100kHz)	110±10 Ω

■Cable structural drawing (reference)



Example of three-pair cable structural drawing

■Point of contact

For details on cables such as specifications, contact each cable manufacturer.

10BASE-T cable

Available 10BASE-T cables for PC LAN wiring can be used.

■Selection procedure when purchasing

Cable type: 10BASE-T cable (Category 3 or higher)

Connection specifications: Straight type

Connector: RJ45 connector

■Precautions for using cables

Pay attention to the following points when purchasing cables.

- 5 V DC is output to the PU connector of the inverters for supplying power to the PU. Please cut pins No. 2 and 8 of commercial cables to prevent wiring to pins No. 2 and 8. ([BMJ-8-28N] distributor is recommended.)

Connecting cables

The table below shows applicable cables and tightening torques.

Item	Number of wires connected per terminal	Wire size		Tightening torque
		Solid wire, Stranded wire	Wire ferrule with insulation sleeve	
FX5U CPU module built-in RS-485 port	One wire	0.2 to 0.5mm^2 (24 to 20 AWG)	0.2 to 0.5mm^2 (24 to 20 AWG)	0.22 to 0.25N·m
	Two wires	0.2mm^2 (24 AWG)	—	
FX5UC CPU module built-in RS-485 port FX5-485-BD FX5-485ADP	One wire	0.3 to 0.5mm^2 (22 to 20 AWG)	0.3 to 0.5mm^2 (22 to 20 AWG)	
	Two wires	0.3mm^2 (22 AWG)	—	

Precautions

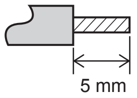
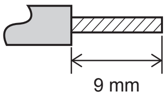
Do not tighten terminal screws with torque beyond the specified range. Otherwise it may cause equipment failure or malfunction.

■Wire end treatment

With regard to the cable end treatment, use a stranded cable or solid cable as is, or use a wire ferrule with insulating sleeve.

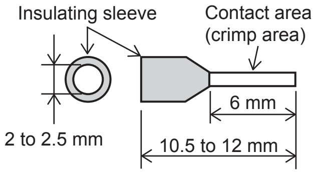
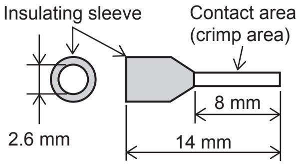
- When using a stranded cable or solid cable as is

- Twist the end of stranded wire and make sure that there are no loose wires.
- Please do not solder plate the ends of the cable.

Dimensions of the cable end	
FX5U CPU module built-in RS-485 port	**FX5UC CPU module built-in RS-485 port, FX5-485-BD, FX5-485ADP**
	

- When using a wire ferrule with insulation sleeve

Because it is difficult to insert a cable into an insulating sleeve depending on the thickness of the cable sheath, select the proper cable according to the outline drawing.

FX5U CPU module built-in RS-485 port	FX5UC CPU module built-in RS-485 port, FX5-485-BD, FX5-485ADP
	

<Reference>

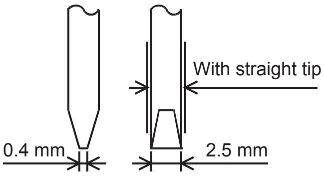
Item	Manufacturer	Model name	Crimping tool
FX5U CPU module built-in RS-485 port	PHOENIX CONTACT GmbH & Co. KG	AI 0.5-6 WH	CRIMPFOX 6 CRIMPFOX 6T-F
FX5UC CPU module built-in RS-485 port FX5-485-BD FX5-485ADP		AI 0.5-8 WH	

- Tool

For tightening the terminal, use a commercially available small screwdriver with straight tip that is not widened toward the end as shown below.

■Precautions

If the diameter of the screwdriver tip is too small, the required tightening torque cannot be achieved. To achieve the appropriate tightening torque shown in the previous page, use the following screwdriver or its equivalent (grip diameter: approximately 25mm).



<Reference>

Manufacturer	Model name
PHOENIX CONTACT GmbH & Co. KG	SZS 0.4×2.5

Connection devices (RJ45 connector and distributor)

Prepare the following devices if necessary.

Product name	Model name	Manufacturer
RJ45 connector	5-554720-3	Tyco Electronics Japan, Ltd.
Distributor	BMJ-8 BMJ-8-28N (no internal connection for pins 2 and 8) (Do not use plug with termination resistor.)	HACHIKO ELECTRIC CO., LTD

Termination resistor setting

Set or connect termination resistor of the inverter farthest from the FX5 PLC.

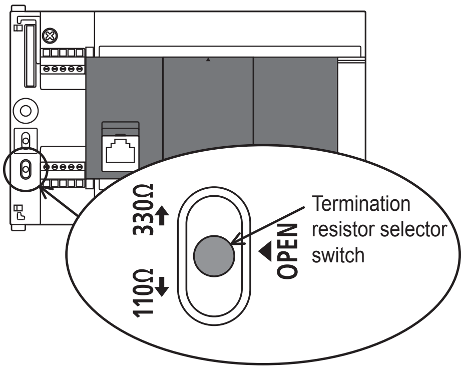
For details on connection, refer to ☞ Page 351 Connection diagram.

At the FX5 PLC

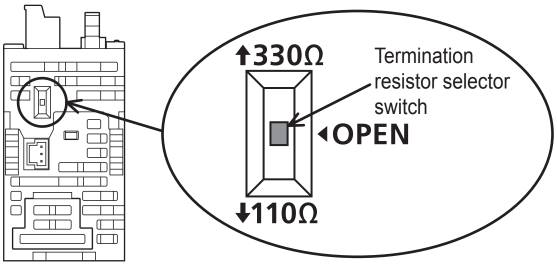
The built-in RS-485 port, FX5-485-BD and FX5-485ADP have a built-in termination resistor.

Set the termination resistor selector switch to 110 Ω.

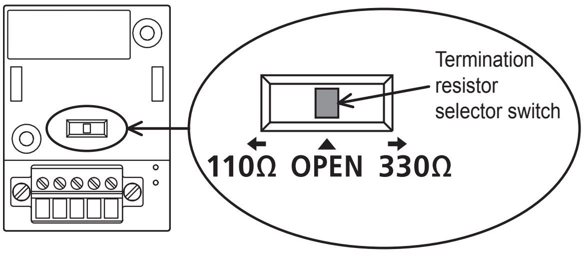
■FX5U CPU module built-in RS-485 port



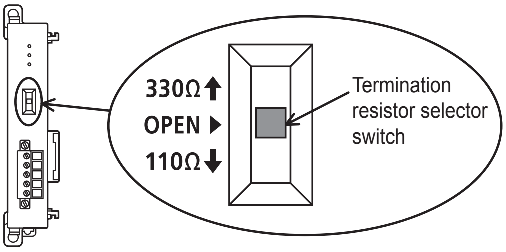
■FX5UC CPU module built-in RS-485 port



■FX5-485-BD



■FX5-485ADP

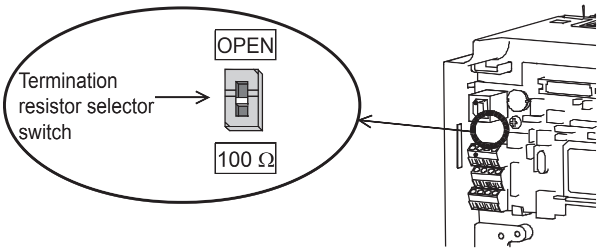


At the inverter

Communication may be affected by noise echo depending on the transmission speed and transmission distance. If communication is hindered by noise echo, connect a termination resistor to the inverter.

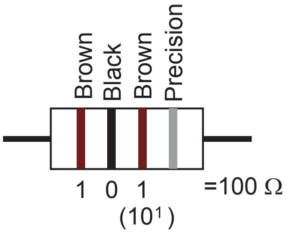
■Built-in RS-485 terminal

A built-in termination resistor is provided. Connect a 100 Ω termination resistor to the inverter farthest from the PLC.



■PU connector

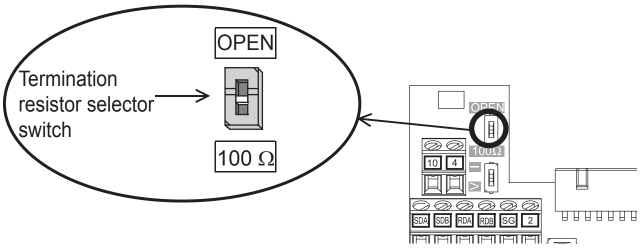
The customer needs to prepare one termination resistor (100 Ω, 1/2 W) such as the following.



- Connect a termination resistor between pin No.3 (RDA) and pin No.6 (RDB).
- Connect a distributor to the PU terminal because termination resistors cannot be connected.
- Connect a termination resistor only to the inverter located furthest away from the PLC.

■FR-E7TR

A built-in termination resistor is provided. Connect a 100 Ω termination resistor to the inverter farthest from the PLC.



■FR-A5NR

Connect a termination resistor chip (which is supplied together with the FR-A5NR) between the RDB and RDR terminals of the inverter farthest from the PLC.

Shielded wiring

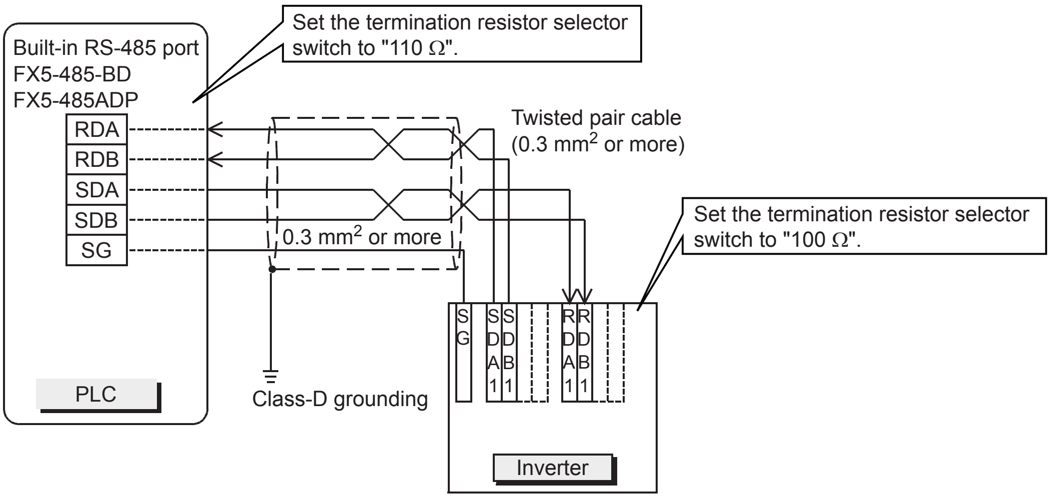
Perform Class-D grounding for the shield of one side of the cable (grounding resistance: 100Ω or less).
For details on connection, refer to ☞ Page 351 Connection diagram.

Connection diagram

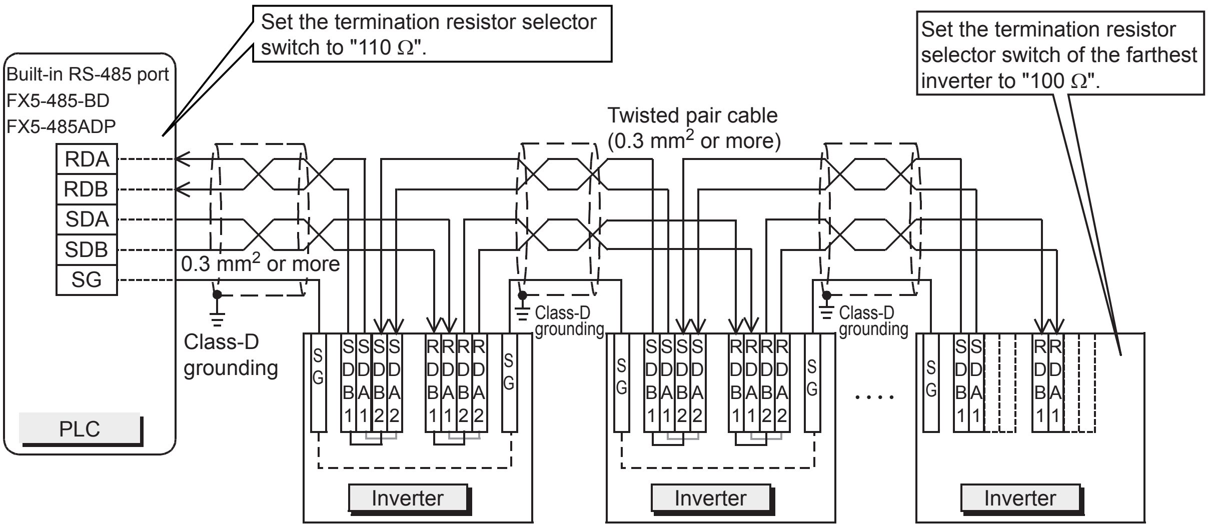
F800/A800/A800 Plus/F700P/A700 series

■Built-in RS-485 terminal

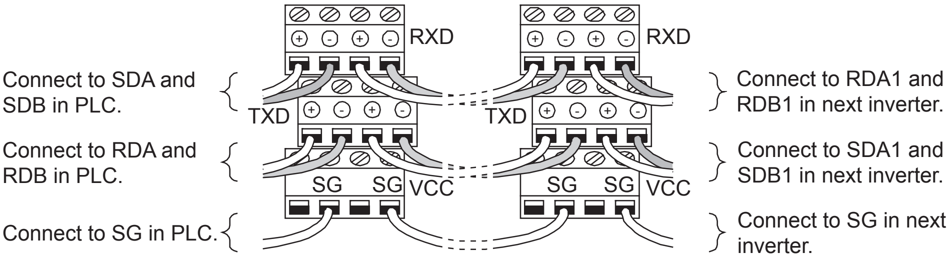
- When one inverter is connected



- When multiple (up to 16) inverters are connected



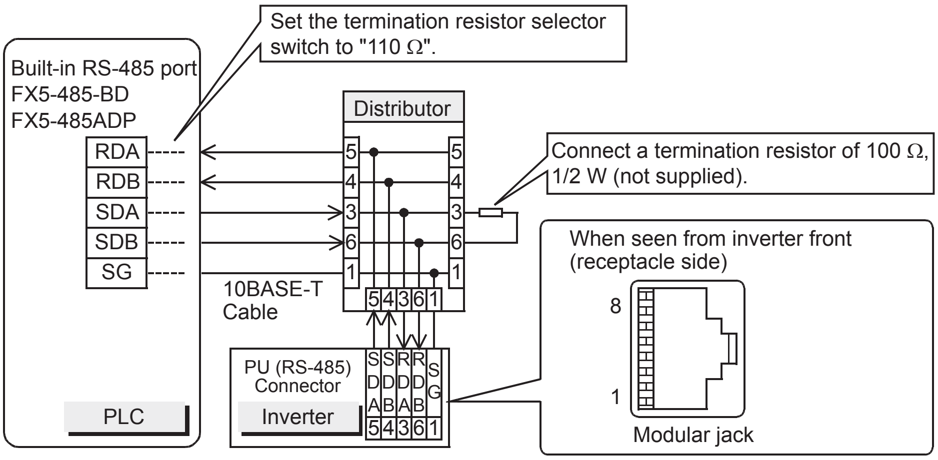
For branching, perform wiring as shown below:



E800/F700PJ/E700/E700EX/D700/V500 series

■PU connector

- When one inverter is connected (4-wire type)

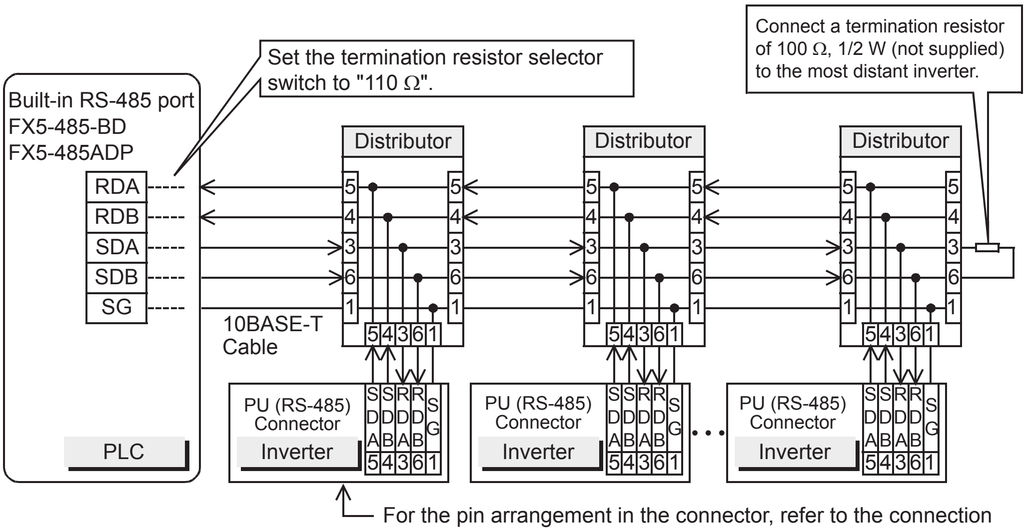


Set the termination resistor selector switch to "110 Ω".

Connect a termination resistor of 100 Ω, 1/2 W (not supplied).

When seen from inverter front (receptacle side)

- When multiple (up to 16) inverters are connected (4-wire type)

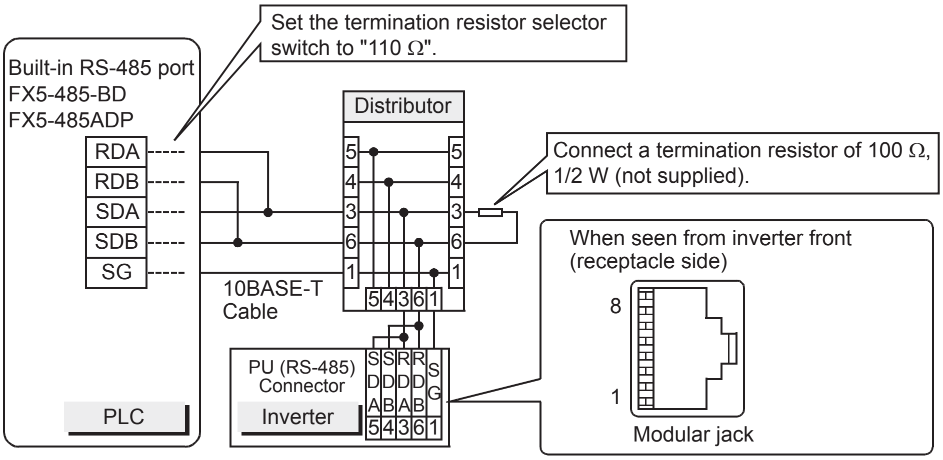


Set the termination resistor selector switch to "110 Ω".

Connect a termination resistor of 100 Ω, 1/2 W (not supplied) to the most distant inverter.

For the pin arrangement in the connector, refer to the connection diagram for one inverter.

- When one inverter is connected (2-wire type, E700 Series only)

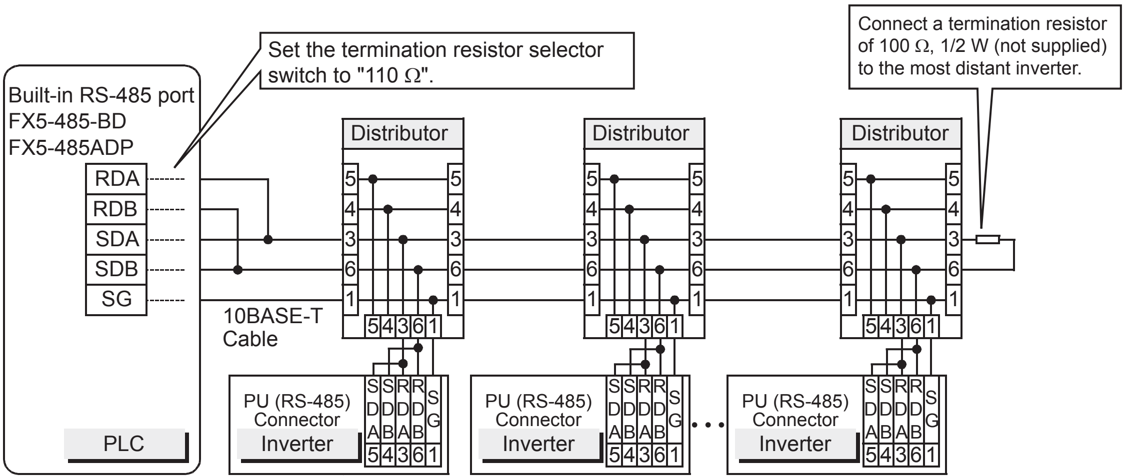


Set the termination resistor selector switch to "110 Ω".

Connect a termination resistor of 100 Ω, 1/2 W (not supplied).

When seen from inverter front (receptacle side)

- When multiple inverters are connected (up to 16) (2-wire type, E700 Series only)



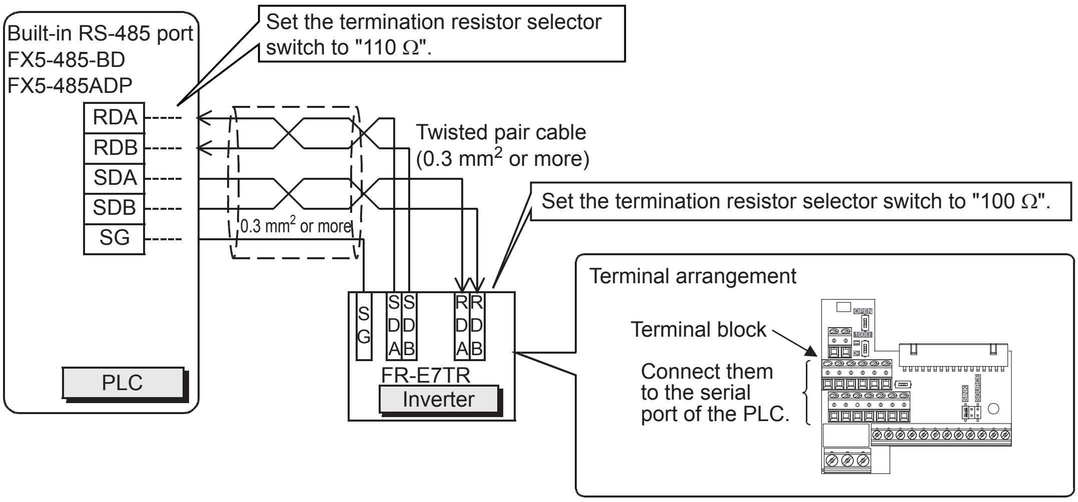
Set the termination resistor selector switch to "110 Ω".

Connect a termination resistor of 100 Ω, 1/2 W (not supplied) to the most distant inverter.

For the pin arrangement in the connector, refer to the connection diagram for one inverter.

■FR-E7TR

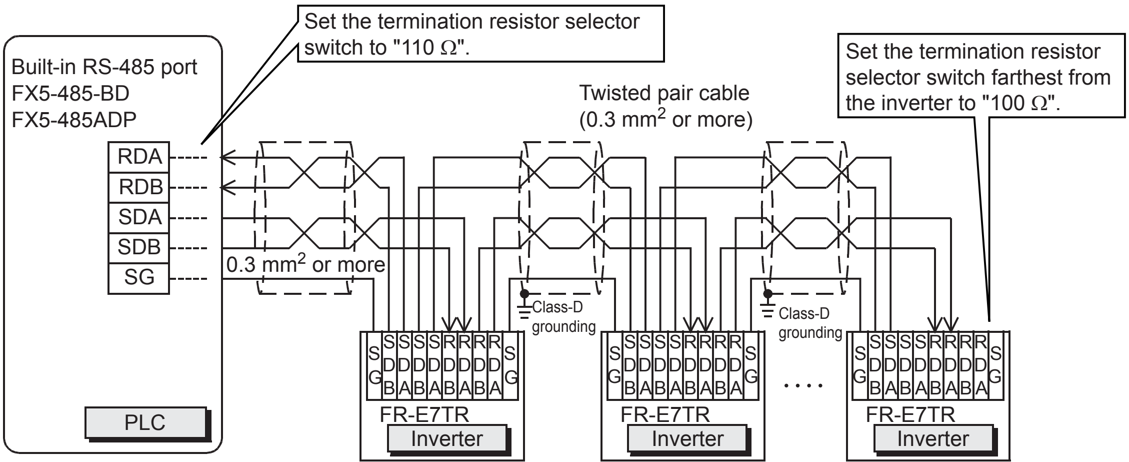
- When one inverter is connected (4-wire type)



Set the termination resistor selector switch to "110 Ω".

Set the termination resistor selector switch to "100 Ω".

- When multiple (up to 16) inverters are connected (4-wire type)

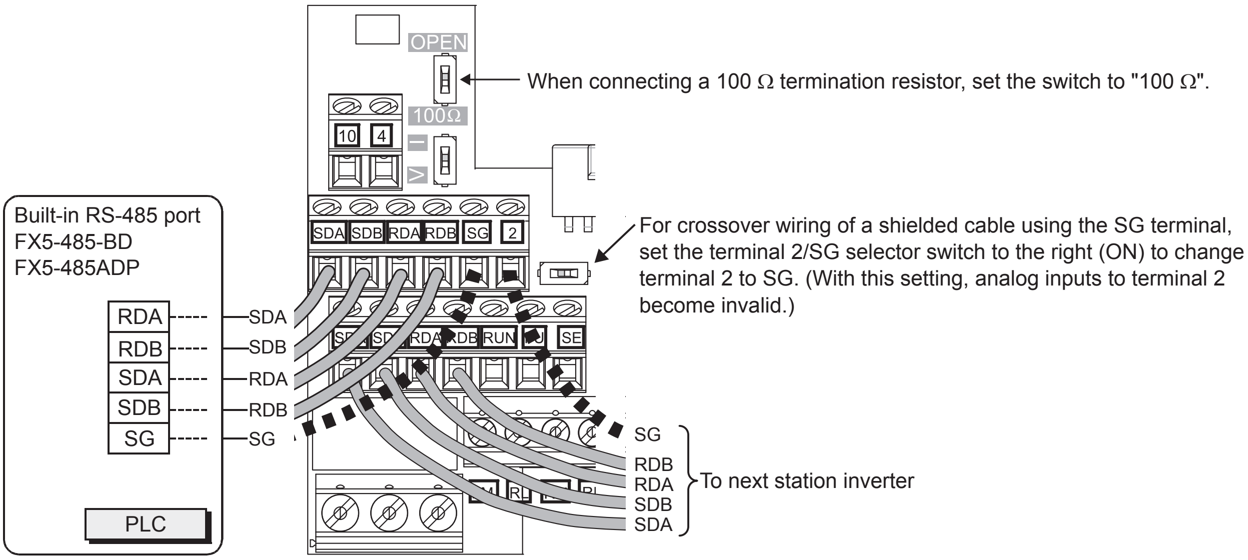


Set the termination resistor selector switch to "110 Ω".

Set the termination resistor selector switch farthest from the inverter to "100 Ω".

For the pin arrangement in the connector, refer to the case of when one inverter is connected.

For branching, perform wiring as shown below: (4-wire type)

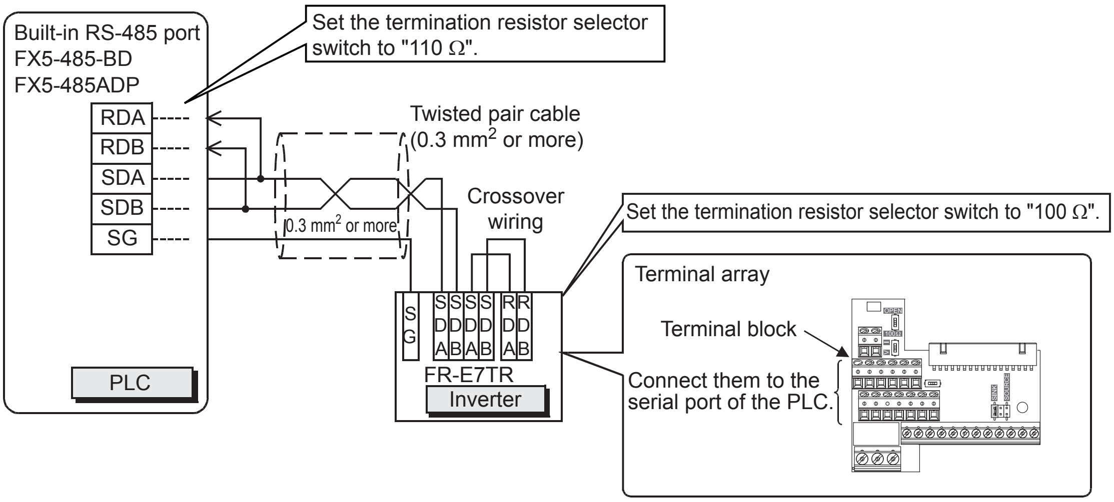


When connecting a 100 Ω termination resistor, set the switch to "100 Ω".

For crossover wiring of a shielded cable using the SG terminal, set the terminal 2/SG selector switch to the right (ON) to change terminal 2 to SG. (With this setting, analog inputs to terminal 2 become invalid.)

SG, RDB, RDA, SDB, SDA: To next station inverter

- When one inverter is connected (2-wire type)

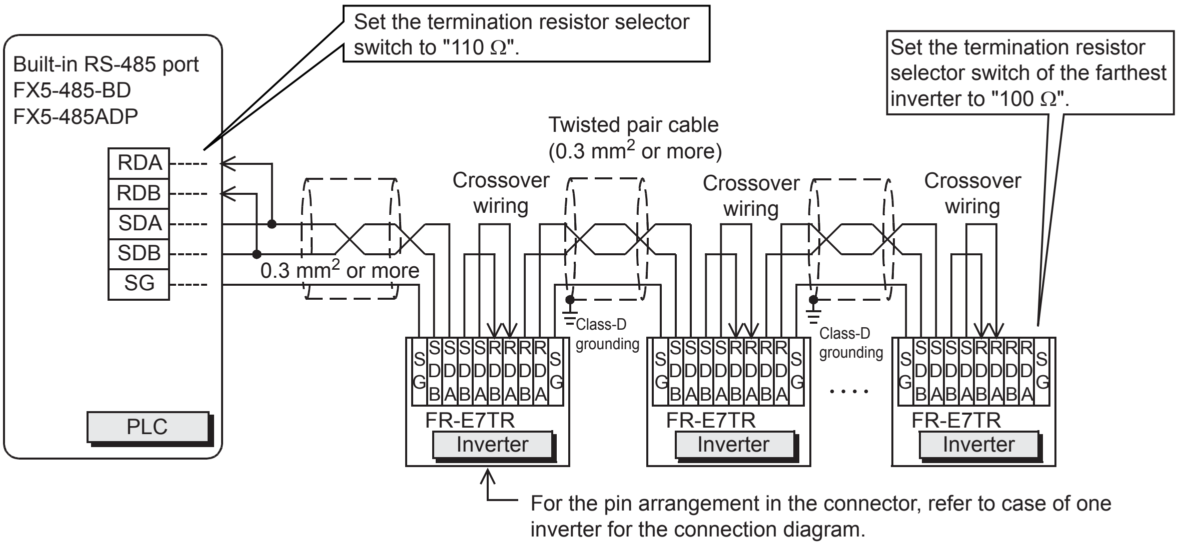


Set the termination resistor selector switch to "110 Ω".

Set the termination resistor selector switch to "100 Ω".

Connect them to the serial port of the PLC.

- When multiple (up to 16) inverters are connected (2-wire type)

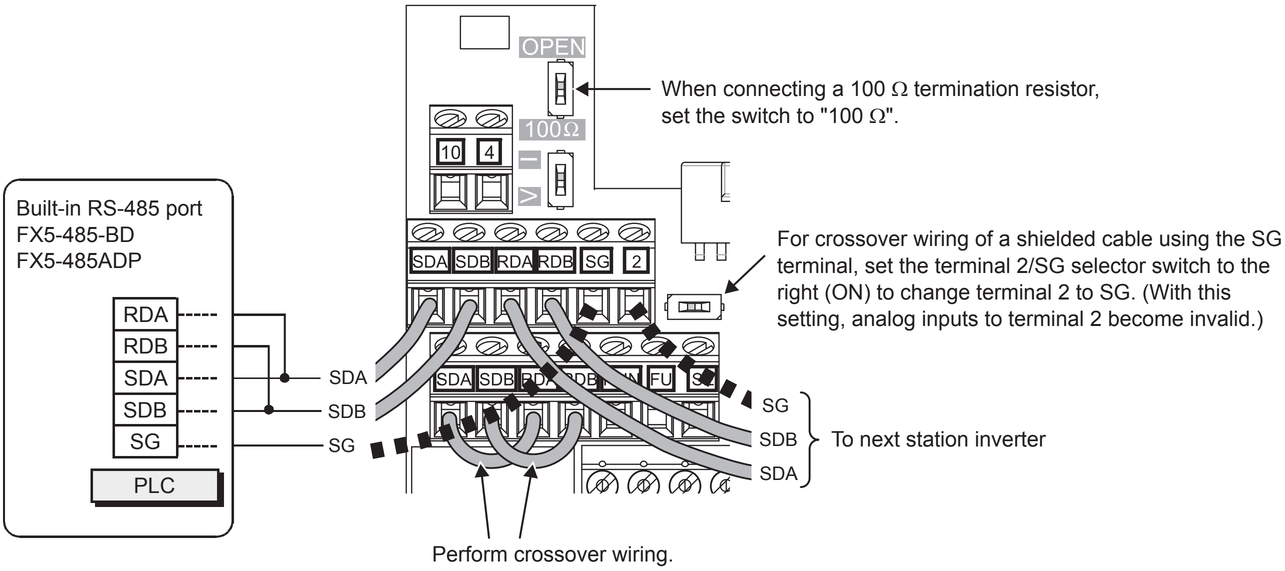


Set the termination resistor selector switch to "110 Ω".

Set the termination resistor selector switch of the farthest inverter to "100 Ω".

For the pin arrangement in the connector, refer to case of one inverter for the connection diagram.

For branching, perform wiring as shown below: (2-wire type)



When connecting a 100 Ω termination resistor, set the switch to "100 Ω".

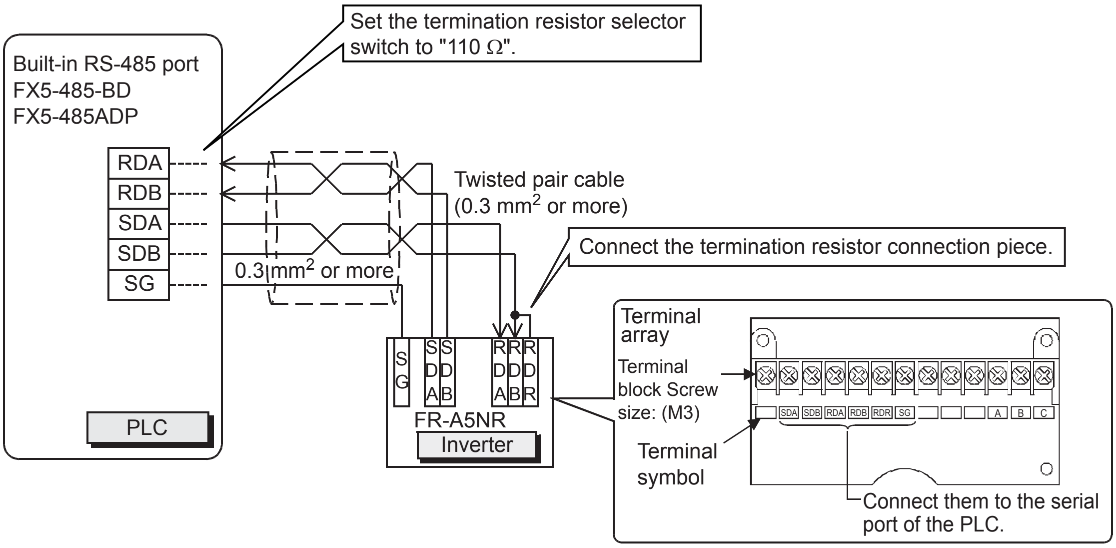
For crossover wiring of a shielded cable using the SG terminal, set the terminal 2/SG selector switch to the right (ON) to change terminal 2 to SG. (With this setting, analog inputs to terminal 2 become invalid.)

To next station inverter

Perform crossover wiring.

■FR-A5NR

- When one inverter is connected



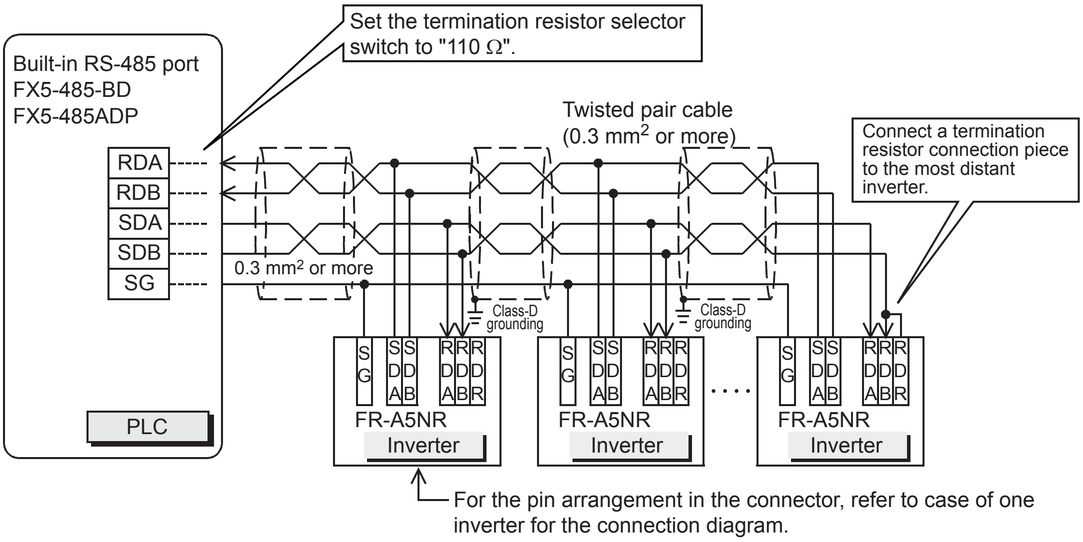
Set the termination resistor selector switch to "110 Ω".

Twisted pair cable (0.3 mm^2 or more)

Connect the termination resistor connection piece.

Connect them to the serial port of the PLC.

- When multiple (up to 16) inverters are connected



Set the termination resistor selector switch to "110 Ω".

Twisted pair cable (0.3 mm^2 or more)

Connect a termination resistor connection piece to the most distant inverter.

For the pin arrangement in the connector, refer to case of one inverter for the connection diagram.

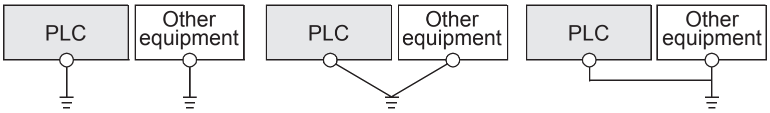
Grounding

Grounding should be performed as stated below.

- Perform Class-D grounding. (Grounding resistance: 100 Ω or less)
- Independent grounding should be performed for best results.

If the PLC cannot be grounded independently, perform the "Shared grounding" shown below.

For details, refer to User's Manual (Hardware) of the CPU module used.



Independent grounding (Best condition)
Shared grounding (Good condition)
Common grounding (Not allowed)

- The grounding wire size should be 14 AWG (2mm^2) or larger.
- Bring the grounding point close to the PLC as much as possible so that the ground cable can be shortened.

19.6 Inverter Communication Settings

Before connecting an inverter to a PLC, set parameters related to communication in the inverter parameter unit (PU) in advance.

For details on each function, refer to the inverter manual for each respective series.

Model	Model classification	Connection destination	Relevant parameters	Reference section
F800 series	Inverter built in	RS-485 terminal	Pr.79, Pr.160, Pr.331 to Pr.342, Pr.549 P.E440, P.N000, P.N001, P.N030 to P.N038, P.D000, P.D001, P.D010, P.D011	Page 357
E800 series	Inverter built in	PU port	Pr.1, Pr.18, Pr.37, Pr.79, Pr.117 to Pr.124, Pr.160, Pr.340, Pr.342, Pr.505, Pr.549	Page 358
A800 series	Inverter built in	RS-485 terminal	Pr.79, Pr.160, Pr.331 to Pr.342, Pr.549 P.E440, P.N000, P.N001, P.N030 to P.N038, P.D000, P.D001, P.D010, P.D011	Page 357
A800 Plus series	Inverter built in	RS-485 terminal	Pr.79, Pr.160, Pr.331 to Pr.342, Pr.549 P.E440, P.N000, P.N001, P.N030 to P.N038, P.D000, P.D001, P.D010, P.D011	Page 357
F700PJ series	Inverter built in	PU port	Pr.79, Pr.117 to Pr.124, Pr.160, Pr.340, Pr.549	Page 361
F700P series	Inverter built in	RS-485 terminal	Pr.79, Pr.160, Pr.331 to Pr.342, Pr.549	Page 360
A700 series	Inverter built in	RS-485 terminal	Pr.79, Pr.160, Pr.331 to Pr.342, Pr.549	
E700 series	Inverter built in	PU port	Pr.79, Pr.117 to Pr.124, Pr.160, Pr.340, Pr.549	Page 361
	Option	FR-E7TR		
E700EX series	Inverter built in	PU port	Pr.79, Pr.117 to Pr.124, Pr.160, Pr.340, Pr.549	
D700 series	Inverter built in	PU port	Pr.79, Pr.117 to Pr.124, Pr.160, Pr.340, Pr.549	
V500 series	Inverter built in	PU port	Pr.79, Pr.117 to Pr.124, Pr.160, Pr.342	Page 363
	Option	FR-A5NR	Pr.79, Pr.160, Pr.331 to Pr.342	

Point

If these parameters are overwritten by the PLC after the inverter is connected, communication will be disabled.

If these parameters are changed by mistake, they should be set again.

FREQROL-F800/A800/A800 Plus series

▶Connection destination: Built-in RS-485 terminal

Contents of communication setting (essential items)

The table below shows parameters (Pr.) which should be set in all cases.

Setting details	Parameter No.	Parameter group	Parameter item	Setting value	Description
Display Setting	160	E440	User group read selection	0	Display simple mode + extended parameters
Communication Setting	331	N030	RS-485 communication station	0 to 31	Up to 16 inverters can be connected.
	332	N031	RS-485 communication speed	48	4800 bps
				96	9600 bps
				192	19200 bps
				384	38400 bps
				576	57600 bps
				1152	115200 bps
	333	—	RS-485 communication stop bit length/data length	10	Data length: 7 bits Stop bit: 1 bit
		N032	RS-485 communication data length	1	Data length: 7 bits
		N033	RS-485 communication stop bit length	0	Stop bit: 1 bit
	334	N034	RS-485 communication parity check selection	2	Even
	337	N037	RS-485 communication waiting time setting	9999	Set in communication data
	341	N038	RS-485 communication CR/LF selection	1	CR: Provided, LF: Not provided
	549	N000	Protocol selection	0	Mitsubishi Electric inverter (computer link) protocol
Operation mode setting	79	D000	Operation mode selection	0	External operation mode is selected when power is turned ON.
	340	D001	Communication startup mode selection	1	Network operation mode

Parameters (test operation, operation)

Parameters that must be adjusted for test operation and operation are as follows.

Parameter No.	Parameter group	Parameter item	Setting value	Set conditions
335	N035	RS-485 communication number of retries	9999	Set the value shown on the left during adjustment, and set a value from 1 to 10 during operation.
336	N036	RS-485 communication check time interval	9999	Set the value shown on the left during adjustment, and set a value in accordance with the system specification during operation.

■Cautions during setting

Caution for setting the RS-485 communication check time interval (Pr.336, P.N036)

Description	Setting value
During adjustment or when communication with the PLC is not executed periodically	9999
When communication with the PLC is not executed	0
Set the communication time in the following cases. • When it is necessary to monitor absence of communication for a certain time and stop the inverter in such a case while communication with the PLC is being executed continuously • When it is necessary to stop the motor at the point when the PLC mode is changed from RUN→STOP	1 to 9998

Parameters (set as needed)

The table below shows parameters to be considered when using system configuration and inverters in various ways.
For the method of use, refer to the respective inverter manual.

Parameter No.	Parameter group	Parameter item	Setting value	Set conditions
342	N001	Communication EEPROM write selection	0 or 1	0: Write to EEPROM and RAM. 1: Write to RAM only.
338	D010	Communication operation command source	0 or 1	0: PLC 1: Outside
339	D011	Communication speed command source	0 or 1	0: PLC 1: Outside

FREQROL-E800 series

▶Connection destination: PU port

Contents of communication setting (essential items)

The table below shows parameters which should be set in all cases.

Set conditions	Parameter No.	Parameter item	Setting value	Description
Display Setting	Pr.160	User group read selection	0, 1, 9999	0: Only simple mode parameters are displayed. 1: Display simple mode + extended parameters 9999: Only parameters registered in user groups are displayed.
Communication Setting	Pr.117	PU communication station number	0 to 31	Up to 16 inverters can be connected.
	Pr.118	PU communication speed	48	4800 bps
			96	9600 bps
			192	19200 bps
			384	38400 bps
	Pr.119	PU communication stop bit length	10	Data length: 7 bits Stop bit: 1 bit
	Pr.120	PU communication parity check	2	Even
	Pr.123	PU communication waiting time setting	9999	Set in communication data
	Pr.124	PU communication CR/LF presence/absence selection	1	CR: Provided, LF: Not provided
	Pr.549	Protocol selection	0	Mitsubishi Electric inverter (computer link) protocol
Operation mode setting	Pr.79	Operation mode selection	0	External operation mode is selected when power is turned ON.
	Pr.340	Communication startup mode selection	1 or 10	1: Network operation mode 10: Network operation mode (Operation mode can be changed between the PU operation mode and network operation mode from the operation panel.)

Parameters (test operation, operation)

Parameters that must be adjusted for test operation and operation are as follows.

Parameter No.	Parameter item	Setting value	Set conditions
Pr.121	Number of PU communication retries	0 to 10, 9999	Set the value shown on the left during adjustment, and set a value from 0 to 10 during operation.
Pr.122	PU communication check time interval	0, 0.1 to 999.8, 9999	Set the value shown on the left during adjustment, and set a value in accordance with the system specification during operation.

■Cautions during setting

Caution for setting the PU communication check time interval (Pr.122)

Description	Setting value
During adjustment or when communication with the PLC is not executed periodically	9999
When communication with the PLC is not executed	0
Set the communication time in the following cases. • When it is necessary to monitor absence of communication for a certain time and stop the inverter in such a case while communication with the PLC is being executed continuously • When it is necessary to stop the motor at the point when the PLC mode is changed from RUN → STOP	0.1 to 999.8

Parameters (set as needed)

The table below shows parameters to be considered when using system configuration and inverters in various ways.

For the method of use, refer to the respective inverter manual.

Parameter No.	Parameter item	Setting value	Set conditions
Pr.1	Maximum frequency	0 to 120	Set the upper limit of the output frequency.
Pr.18	High speed maximum frequency	0 to 590	Set when operating at 120 Hz or higher.
Pr.37	Speed display[*1]	0.01 to 9998	Machine speed at the speed (frequency) set in Pr.505.
Pr.342	Communication EEPROM write selection	0 or 1	0: Write to EEPROM and RAM. 1: Write to RAM only.
Pr.505	Speed setting reference[*1]	1 to 590	Reference speed (frequency) for Pr.37.

*1 The setting ranges of Pr.1 (Pr.18), Pr.37, and Pr.505 are limited so that the following formula is satisfied.
Pr.1 (Pr.18) × Pr.37 / Pr.505 < 8388.607
The setting range of Pr.1 (Pr.18) is not limited when the machine speed display is not selected. To display the machine speed, set values which satisfy the formula.

FREQROL-F700P/A700 series

▶Connection destination: Built-in RS-485 terminal

Contents of communication setting (essential items)

The table below shows parameters which should be set in all cases.

Set conditions	Parameter No.	Parameter item	Setting value	Description
Display Setting	Pr.160	User group read selection	0	Display simple mode + extended parameters
Communication Setting	Pr.331	RS-485 communication station	0 to 31	Up to 16 inverters can be connected.
	Pr.332	RS-485 communication speed	48	4800 bps
			96	9600 bps
			192	19200 bps
			384	38400 bps
	Pr.333	RS-485 communication stop bit length	10	Data length: 7 bits Stop bit: 1 bit
	Pr.334	RS-485 communication parity check selection	2	Even
	Pr.337	RS-485 communication waiting time setting	9999	Set in communication data
	Pr.341	RS-485 communication CR/LF selection	1	CR: Provided, LF: Not provided
	Pr.549	Protocol selection	0	Mitsubishi Electric inverter (computer link) protocol
Operation mode setting	Pr.79	Operation mode selection	0	External operation mode is selected when power is turned ON.
	Pr.340	Communication startup mode selection	1	Network operation mode

Parameters (test operation, operation)

Parameters that must be adjusted for test operation and operation are as follows.

Parameter No.	Parameter item	Setting value	Set conditions
Pr.335	RS-485 communication number of retries	9999	Set the value shown on the left during adjustment, and set a value from 1 to 10 during operation.
Pr.336	RS-485 communication check time interval	9999	Set the value shown on the left during adjustment, and set a value in accordance with the system specification during operation.

■Cautions during setting

Caution for setting the RS-485 communication check time interval (Pr.336)

Description	Setting value
During adjustment or when communication with the PLC is not executed periodically	9999
When communication with the PLC is not executed	0
Set the communication time in the following cases. • When it is necessary to monitor absence of communication for a certain time and stop the inverter in such a case while communication with the PLC is being executed continuously • When it is necessary to stop the motor at the point when the PLC mode is changed from RUN→STOP	1 to 9998

Parameters (set as needed)

The table below shows parameters to be considered when using system configuration and inverters in various ways.
For the method of use, refer to the respective inverter manual.

Parameter No.	Parameter item	Setting value	Set conditions
Pr.342	Communication EEPROM write selection	0 or 1	0: Write to EEPROM and RAM. 1: Write to RAM only.
Pr.338	Communication operation command rights	0 or 1	0: PLC 1: Outside
Pr.339	Communication speed command rights	0 or 1	0: PLC 1: Outside

FREQROL-F700PJ/E700/D700/E700EX series

▶Connection destination: PU port, FR-E7TR

Contents of communication setting (essential items)

The table below shows parameters which should be set in all cases.

Set conditions	Parameter No.	Parameter item	Setting value	Description
Display Setting	Pr.160	Extended function display selection	0	Display simple mode + extended parameters
Communication Setting	Pr.117	PU communication station number	0 to 31	Up to 16 inverters can be connected.
	Pr.118	PU communication speed	48	4800 bps
			96	9600 bps
			192	19200 bps
			384	38400 bps
	Pr.119	PU communication stop bit length	10	Data length: 7 bits Stop bit: 1 bit
	Pr.120	PU communication parity check	2	Even
	Pr.123	PU communication waiting time setting	9999	Set in communication data
	Pr.124	PU communication CR/LF presence/absence selection	1	CR: Provided, LF: Not provided
	Pr.549	Protocol selection	0	Mitsubishi Electric inverter (computer link) protocol
Operation mode setting	Pr.79	Operation mode selection	0	External operation mode is selected when power is turned ON.
	Pr.340	Communication startup mode selection	1 or 10	1: Network operation mode 10: Network operation mode (Operation mode can be changed between the PU operation mode and network operation mode from the operation panel.)

Parameters (test operation, operation)

Parameters that must be adjusted for test operation and operation are as follows.

Parameter No.	Parameter item	Setting value	Set conditions
Pr.121	Number of PU communication retries	9999	Set the value shown on the left during adjustment, and set a value from 1 to 10 during operation.
Pr.122	PU communication check time interval	9999	Set the value shown on the left during adjustment, and set a value in accordance with the system specification during operation.

■Cautions during setting

Caution for setting the PU communication check time interval (Pr.122)

Description	Setting value
During adjustment or when communication with the PLC is not executed periodically	9999
When communication with the PLC is not executed	0
Set the communication time in the following cases. • When it is necessary to monitor absence of communication for a certain time and stop the inverter in such a case while communication with the PLC is being executed continuously • When it is necessary to stop the motor at the point when the PLC mode is changed from RUN → STOP	1 to 9998

Parameters (set as needed)

The table below shows parameters to be considered when using system configuration and inverters in various ways.

For the method of use, refer to the respective inverter manual.

Parameter No.	Parameter item	Setting value	Set conditions
Pr.37	Speed display[*1]	0 or 0.01 to 9998	0: Frequency display, setting 0.01 to 9998: Machine speed at 60Hz.
Pr.146	Built-in potentiometer switching[*2]	0 or 1	0: The built-in frequency setting knob is valid. 1: The built-in frequency setting knob is invalid.
Pr.342	Communication EEPROM write selection	0 or 1	0: Write to EEPROM and RAM. 1: Write to RAM only.

*1 This inverter communication function cannot read or write Pr. 37.
Set Pr.37 to "0" when setting or monitoring the frequency in the PLC.
If any value other than "0" is set and the instruction code HFF is set to "01", the frequency may not be set or monitored normally.

*2 Set Pr.146 to "1" when changing the frequency from the PLC.

FREQROL-V500 series

▶Connection destination: PU port, FR-A5NR

Contents of communication setting (essential items)

The table below shows parameters which should be set in all cases.

Set conditions	Parameter No.	Parameter item	Setting value	Description
Display Setting	Pr.160	Extended function display selection	1	Display simple mode + extended parameters
Communication Setting	Pr.117	Communication station number	0 to 31	Up to 16 inverters can be connected.
	Pr.118	Communication speed	48	4800 bps
			96	9600 bps
			192	19200 bps
	Pr.119	Stop bit length/data length	10	Data length: 7 bits Stop bit: 1 bit
	Pr.120	Parity check provided/not provided	2	Even
	Pr.123	Waiting time setting	9999	Set in communication data
	Pr.124	CR/LF provided/not provided selection	1	CR: Provided, LF: Not provided
Operation mode setting	Pr.79	Operation mode selection	0	External operation mode is selected when power is turned ON.
	Pr.340	Communication startup mode selection	1 or 10	1: Network operation mode 10: Network operation mode (Operation mode can be changed between the PU operation mode and network operation mode from the operation panel.)

Parameters (test operation, operation)

Parameters that must be adjusted for test operation and operation are as follows.

Parameter No.	Parameter item	Setting value	Set conditions
Pr.121	Number of communication retries	9999	Set the value shown on the left during adjustment, and set a value from 1 to 10 during operation.
Pr.122	Communication check time interval	9999	Set the value shown on the left during adjustment, and set a value in accordance with the system specification during operation.

■Cautions during setting

Caution for setting the PU communication check time interval (Pr.122)

Description	Setting value
During adjustment or when communication with the PLC is not executed periodically	9999
When communication with the PLC is not executed	0
Set the communication time in the following cases. • When it is necessary to monitor absence of communication for a certain time and stop the inverter in such a case while communication with the PLC is being executed continuously • When it is necessary to stop the motor at the point when the PLC mode is changed from RUN → STOP	1 to 9998

Parameters (set as needed)

The table below shows parameters to be considered when using system configuration and inverters in various ways.

For the method of use, refer to the V500 series inverter manual.

Parameter No.	Parameter item	Setting value	Set conditions
Pr.342	Communication EEPROM write selection	0 or 1	0: Write to EEPROM. 1: Write to RAM.

19.7 PLC Communication Settings

For the FX5 communication settings of this function, parameters are set using GX Works3. For details about GX Works3, refer to 📖GX Works3 Operating Manual.

Setting of parameter differs according to the module used. The procedure for each module is as follows.

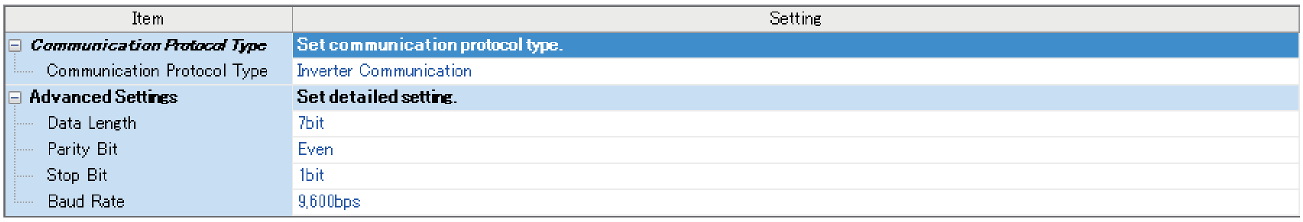
Built-in RS-485 port (CH1)

Navigation Window ⇨ Parameter ⇨ FX5UCPU ⇨ Module Parameter ⇨ 485 Serial Port

Window

The following screen will be displayed if [Inverter Communication] is set for the communication protocol type.

■Basic Settings

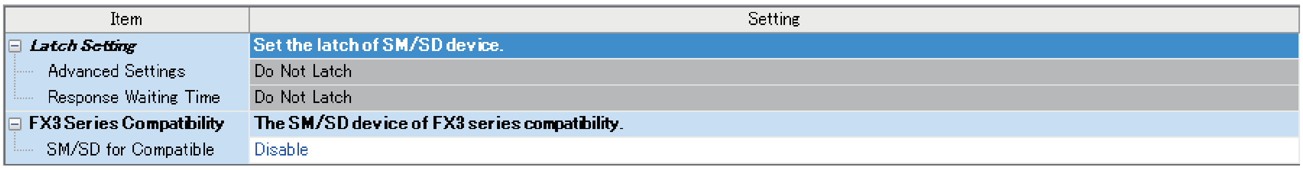


Item	Setting
⊟ *Communication Protocol Type*	Set communication protocol type.
Communication Protocol Type	Inverter Communication
⊟ **Advanced Settings**	**Set detailed setting.**
Data Length	7bit
Parity Bit	Even
Stop Bit	1bit
Baud Rate	9,600bps

■Fixed Setting

Item	Setting
⊟ *Response Waiting Time*	Set response waiting time.
Response Waiting Time	100 ms

■SM/SD Setting



Item	Setting
⊟ *Latch Setting*	Set the latch of SM/SD device.
Advanced Settings	Do Not Latch
Response Waiting Time	Do Not Latch
⊟ **FX3 Series Compatibility**	**The SM/SD device of FX3 series compatibility.**
SM/SD for Compatible	Disable

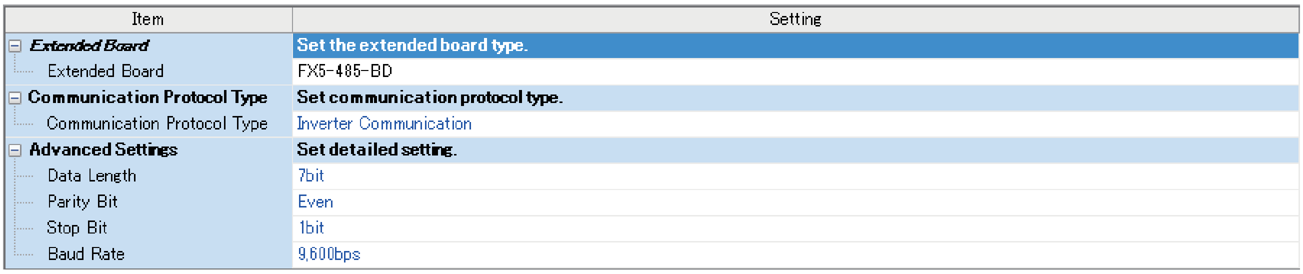
Communication board (CH2)

Navigation Window ⇨ Parameter ⇨ Model name ⇨ Module Parameter ⇨ Extended Board

Window

The following screen will be displayed if [FX5-485-BD] is set for the extended board and [Inverter Communication] is set for the communication protocol type. Fixed setting and SM/SD setting are the same as for the built-in RS-485 port (CH1).

■Basic Settings



Item	Setting
⊟ *Extended Board*	Set the extended board type.
Extended Board	FX5-485-BD
⊟ **Communication Protocol Type**	**Set communication protocol type.**
Communication Protocol Type	Inverter Communication
⊟ **Advanced Settings**	**Set detailed setting.**
Data Length	7bit
Parity Bit	Even
Stop Bit	1bit
Baud Rate	9,600bps

Communication adapter (CH3/CH4)

When an expansion adapter is used, add expansion adapter to Module Information.

Navigation window ⇨ Parameter ⇨ Module Information ⇨ Right-click ⇨ Add New Module

After adding the expansion adapter, make settings on the screen displayed from the following operation.

Navigation window ⇨ Parameter ⇨ Module Information ⇨ ADP1 to ADP6 (Communication adapter) ⇨ Module Parameter

Window

Each setting screen is the same as for the built-in RS-485 port (CH1).

Parameter setting details

Set the following items for serial ports that use inverter communication.

Item			Description	Reference section
Basic Settings	Extended Board[1]		When using this function, select [FX5-485-BD].	—
	Communication Protocol Type		When using this function, select [Inverter Communication].	
	Advanced Settings	Data Length	7bit/8bit	
		Parity Bit	None/Odd/Even	
		Stop Bit	1bit/2bit	
		Baud Rate	4800bps/9600bps/19200bps/38400bps/57600bps/115200bps	
Fixed Setting	Response Waiting Time		1 to 32767 (ms)	—
SM/SD Setting	FX3 Series Compatibility	SM/SD for Compatible	Disable/CH1/CH2	Page 365

[1] Only for the communication board (CH2).

The following settings are unnecessary (fixed values).

Item	Description
Start Bit	1 bit
Header	Not Added
Terminator	Not Added
Control Mode	None
Sum Check Code	Not Added
Control Procedure	None

FX3 Series-compatible SM/SD

When using the FX3 Series compatible SM/SD storage area, set to use special devices for either the FX3 Series CH1 or CH2. FX3 Series compatible devices corresponding to the specified channel can be used.

For details, refer to the following. ☞ Page 404 Related Devices

19.8 Programming

This section explains how to create programs which change parameters of inverters and give operation commands to inverters.

A program example is shown for each inverter communication instruction.

For details on related devices, refer to ☞ Page 404 Related Devices.

Common items in inverter communication instructions

Inverter communication types

The PLC and inverter communicate using inverter communication instructions.

There are six types of inverter communication instructions that differ according to data communication direction and parameter writing/reading direction as follows.

Instruction	Function	Control direction	Reference section
IVCK	Monitors operations of an inverter	PLC←Inverter	Page 381
IVDR	Controls operations of an inverter	PLC→Inverter	Page 383
IVRD	Reads a parameter from an inverter.	PLC←Inverter	Page 385
IVWR	Writes a parameter to an inverter.	PLC→Inverter	Page 388
IVBWR	Writes multiple parameters to an inverter.	PLC→Inverter	Page 391
IVMC	Multiple inverter commands	PLC↔Inverter	Page 394

Function and operation

■Communication start timing

At the rising edge (OFF→ON) of the drive contact of the inverter communication instruction, the PLC starts communication with an inverter.

Even if the drive contact turns OFF during communication with an inverter, the PLC executes communication until the last instruction.

When the drive contact is always ON, the PLC executes communication repeatedly.

■Communication execution state output device

The inverter communication instruction of the FX5 specifies communication execution state in operand (d) or (d2).

Bit device (3 bit occupied) outputs according to state of inverter communication instruction (execution in progress/normal completion/error completion); the state can be checked from specified bit device.

The operations of the specified bit devices (3 bit occupied) and data that can be handled simultaneously are as follows. The following device is not dedicated to a single inverter communication instruction, but is rather shared with other inverter communication instructions. Operation may vary with execution of other instructions. The status of SM8029 is "ON" regardless of whether operation ends normally or abnormally. (d) + 2 or (d2) + 2 is however ON when operation ends abnormally. Therefore, normal operation end and abnormal end can be distinguished.

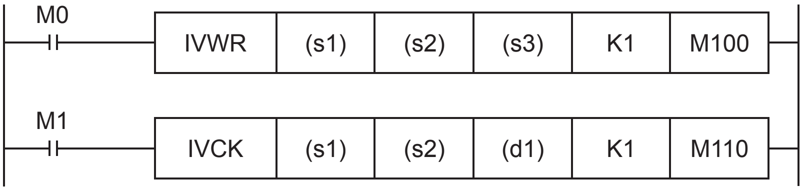
Communication execution state output device	Name	Operation	Device operated simultaneously	
(d) or (d2)	Instruction execution flag	During instruction execution: ON Other than instruction execution: OFF	Inverter communicating	CH1: SM8920 CH2: SM8930 CH3: SM8940 CH4: SM8950
(d) +1 or (d2) +1	Instruction execution normal end flag	When instruction is completed normally: ON When instruction activation contact rises: OFF	Instruction execution completed	SM8029
(d) +2 or (d2) +2	"Instruction abnormal end" flag	When instruction ends in error: ON When instruction activation contact rises: OFF		

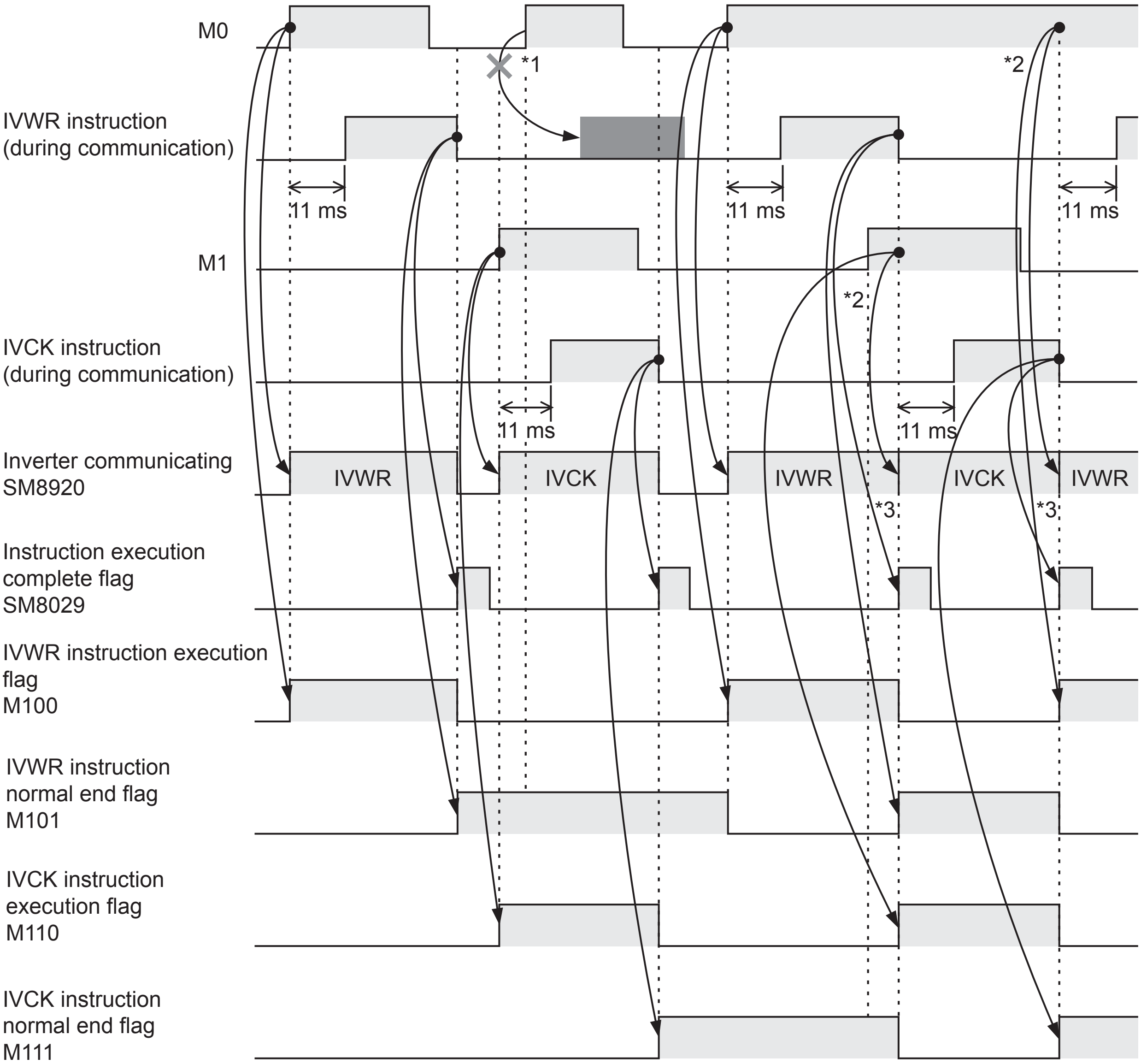
■Simultaneous driving of instructions and communication

(1) Driving instructions at the same time

- Two or more inverter communication instructions can be programmed, and driven at the same time.
- When two or more instructions are driven at the same time for the serial port used for communication, the next inverter communication instruction in the program is executed after the current communication with the inverter has finished.

- Inverter communication instructions standby for 11 ms after acquiring a serial port, and then starts communication. Even if the drive contact turns ON, the inverter communication instruction stands by until (SM8920) turns ON→OFF when the SM8920 inverter communication busy flag has been turned ON by other inverter communication instruction. The PLC frees the serial port, and then executes inverter communication instructions driven in the next step and so on.





*1 If the command contact M1 is OFF and command contact M0 is turned ON when SM8920 (IVWR instruction execution complete) is turned OFF, the IVWR instruction will not be executed because another instruction (IVCK instruction) is being executed.

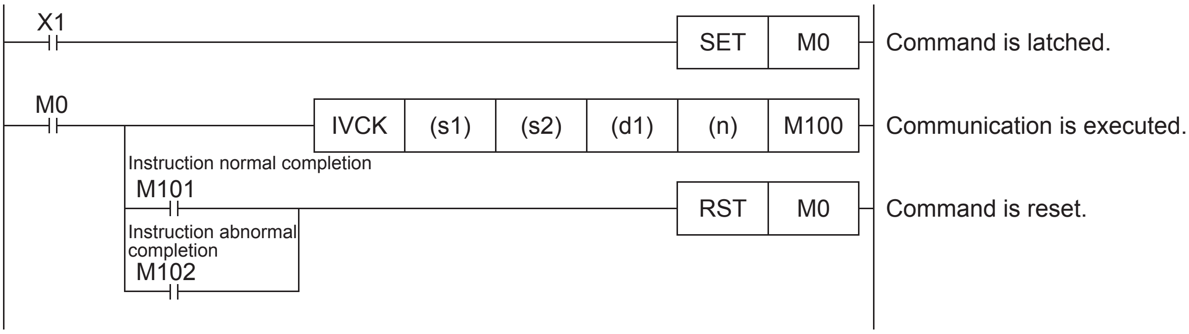
*2 When two or more instructions are driven at the same time during communication, the next inverter communication instruction is executed after the current instruction is completed.

*3 SM8920 remains OFF until the next inverter communication instruction is driven after execution of the current inverter communication instruction is completed.

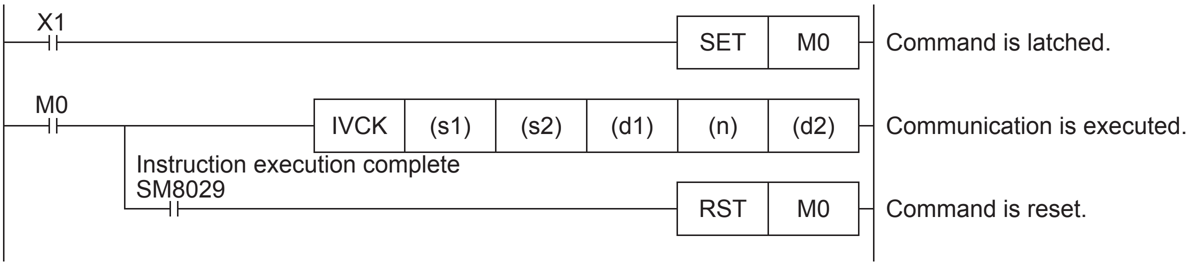
(2) Precautions for programming

When communicating with inverters for multiple items, set the command contact for inverter communication instruction to ON until transmission is complete. Program so that the command contact is turned OFF, when all communication with the inverter has been completed, using the instruction execution normal end flag ([d] +1 or [d2] +1), instruction execution abnormal end flag ([d] +2 or [d2] +2) or instruction execution completed flag (SM8029).

- If using instruction execution normal end flag ([d] +1 or [d2] +1) or instruction execution abnormal end flag ([d] +2 or [d2] +2)



- When using instruction execution complete flag (SM8029)



Instruction completion and error flag operation

When multiple inverter communication instructions are programmed, the following flags turn ON or OFF according to the execution result of each inverter communication instruction.

To acquire the result of each inverter communication instruction, program as necessary just below the inverter communication instruction using these flags.

■Related devices

- Special relays

FX5 dedicated				**Description**
CH1	**CH2**	**CH3**	**CH4**	
SM8029				Instruction execution completed
SM8500	SM8510	SM8520	SM8530	Serial communication error
SM8921	SM8931	SM8941	SM8951	IVBWR instruction error

- Special registers

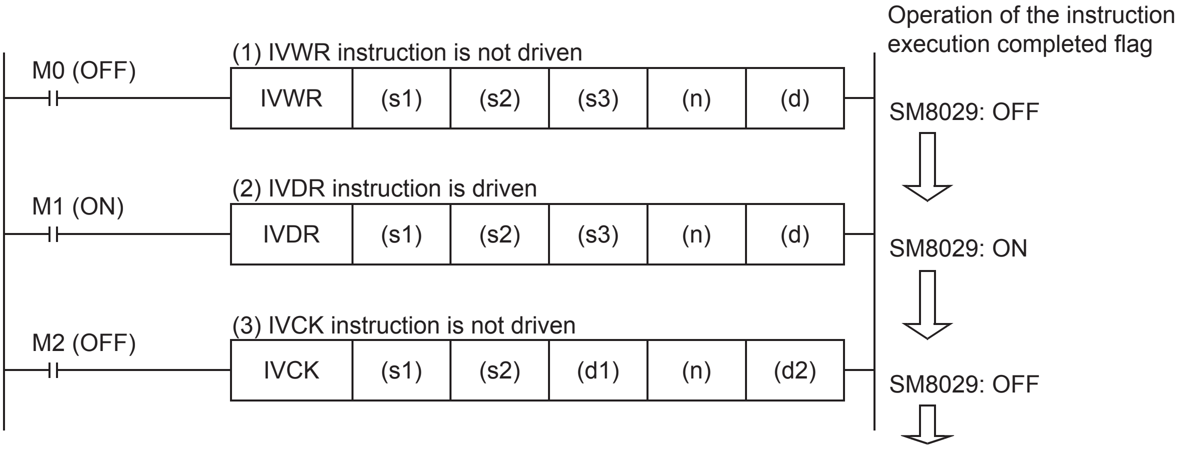
FX5 dedicated				**Description**
CH1	**CH2**	**CH3**	**CH4**	
SD8500	SD8510	SD8520	SD8530	Serial communication error code
SD8921	SD8931	SD8941	SD8951	IVBWR instruction error parameter number

■Operation of instruction execution complete flag

When communication with an inverter is completed, the instruction execution completed flag (SM8029) turns ON, and remains ON for 1 scan.

SM8029 indicates completion of IVDR instruction communication as shown below, when M0, M2 are OFF and M1 is ON.

If inverter communication instruction ends normally, instruction execution normal end flag (☞ Page 366 Communication execution state output device) turns ON simultaneously with SM8029.



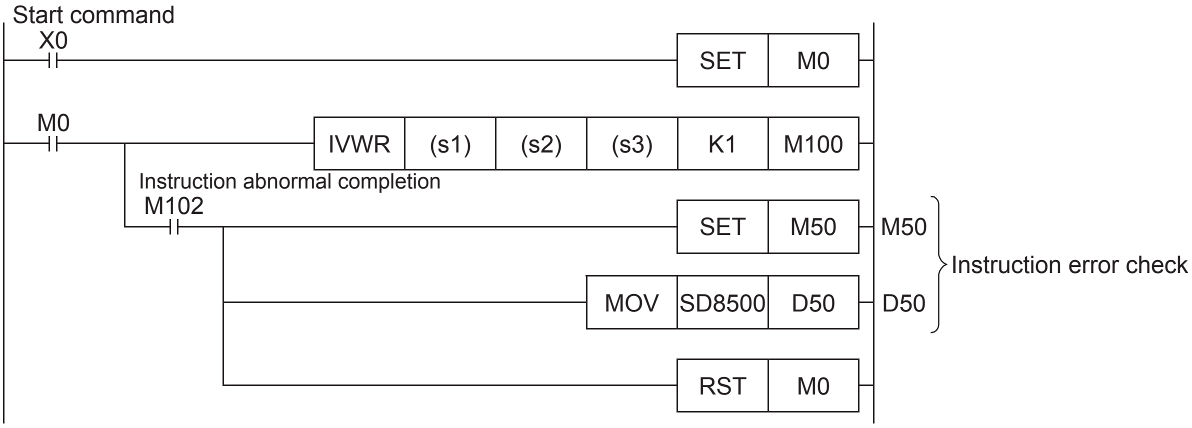
■Processing of communication errors

Communication errors include those that occur for all serial communication functions (parity error, overrun error, framing error) and those that occur during communication with the inverter. A serial communication error results if either occurs.

Item		CH1	CH2	CH3	CH4
Serial communication error	Device No.	SM8500	SM8510	SM8520	SM8530
	Operation	ON (ON maintained)			
Serial communication error code	Device No.	SD8500	SD8510	SD8520	SD8530
	Error code	7010H: Parity error, overrun error or framing error			
		76**H: Inverter communication error (☞ Page 842)			

You can create the following program for the corresponding instruction to check the inverter communication error code.

Program example



Precautions for program creation

■Communication protocol setting

If protocol of communication settings (☞ Page 364 PLC Communication Settings) for the serial port to be used is not set to "Inverter Communication", inverter communication instructions cannot be used.

■Using inverter communication instruction together with another instruction

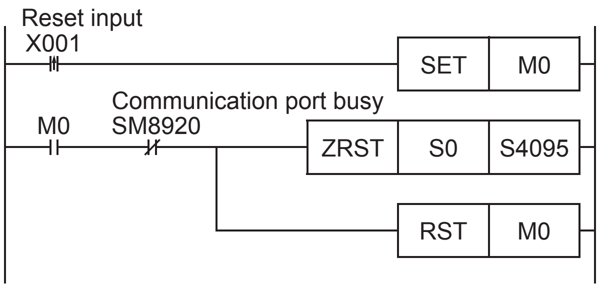
A serial port using any other communication (such as RS2 instruction) cannot use inverter instruction and predefined protocol support instruction.

(☞ Page 877 Combined Use of Serial Communication)

■When programming an inverter communication instruction in a state of an STL instruction

Do not turn state OFF until communication with the inverter is finished. Program in accordance with the following sequence.

- Add SM8029 (instruction execution complete flag) ON condition to the state relay transfer condition, and provide interlock such that the state relay ON/OFF status does not change during communication with an inverter. If the state changes during communication, communication may not be performed normally.
- When resetting many state relays at once using the ZRST instruction or such an instruction, make sure that the SM8920, SM8930, SM8940 or SM8950 (Inverter communicating) condition is OFF.



State relays S0 to S4095 are reset all at once.

■Using an inverter communication instruction in a program flow

An inverter communication instruction cannot be used in the following program flows.

Program flow that cannot use the inverter communication instruction	Remarks
Between CJ and P instructions	Conditional jump
Between FOR and NEXT instructions	Repeat
Between P and RET instructions	Subroutine
Between I and IRET instructions	Interrupt routine

■Precautions when using the password function for inverter

Note the following if using the password function for the inverter. Inverters that support password function are FREQROL-F800, A800, F700PJ, F700P, A700, E700, E700EX and D700.

- When a communication error occurs

When a communication error occurs in an inverter communication instruction, the PLC automatically retries communication up to 3 times.*1

Hence, when a password disable error occurs in the inverter in which "display of the number of times of password disable error"*2 is enabled using Pr.297, please note that the number of times of password disable error displayed in accordance with the setting of Pr.297 may not be the same as the actual number of times of password input error as described below.

Do not execute automatic retry (re-driving of an inverter communication instruction) using a sequence program when writing data to Pr.297.

■Cases in which a password disable error occurs in an inverter communication instruction, and the actual number of times of disable error in such cases.

(1) If an incorrect password is written in Pr.297 due to a typing mistake, number of times of password disable error will be 3 from one execution of inverter instruction.

(2) If the password cannot be written in Pr.297 correctly due to noise, etc., the maximum number of times of password disable error is 3.

- When registering the password

When registering the password in the inverter using an inverter communication instruction, write the password to Pr.297, read Pr.297, and then confirm that registration of the password is completed normally.*3 If writing of the password to Pr.297 is not completed normally due to noise, etc., the PLC automatically retries writing, and the registered password may be reset by the retry.

*1 The PLC executes the first communication, and then retries communication twice (3 times in total).

*2 When "display of the number of times of password disable error" is enabled using Pr.297 and when a password disable error occurs 5 times, the "reading/writing restriction" cannot be disabled even if the right password is input. For recovery from this status, it is necessary to all-clear all parameters.

*3 If the value read from Pr.297 is between 0 to 4, registration of the password is completed normally.

Program example

This program performs operation monitoring, operation control, and parameter control for two inverters (Station number 0 and 1) from the built-in RS-485 port.

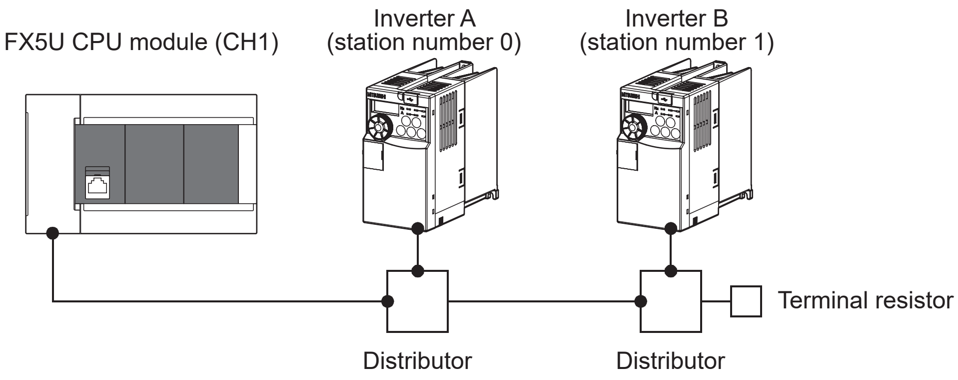
Contents of operation

Operation of the inverter is controlled by the input (X) of the FX5U CPU module, and the speed is changed by the device.

Item	Station number 0	Station number 1
Operation stop	X0	X10
Forward rotation	X1	X11
Reverse rotation	X2	X12
Speed change	D10	D510

System configuration

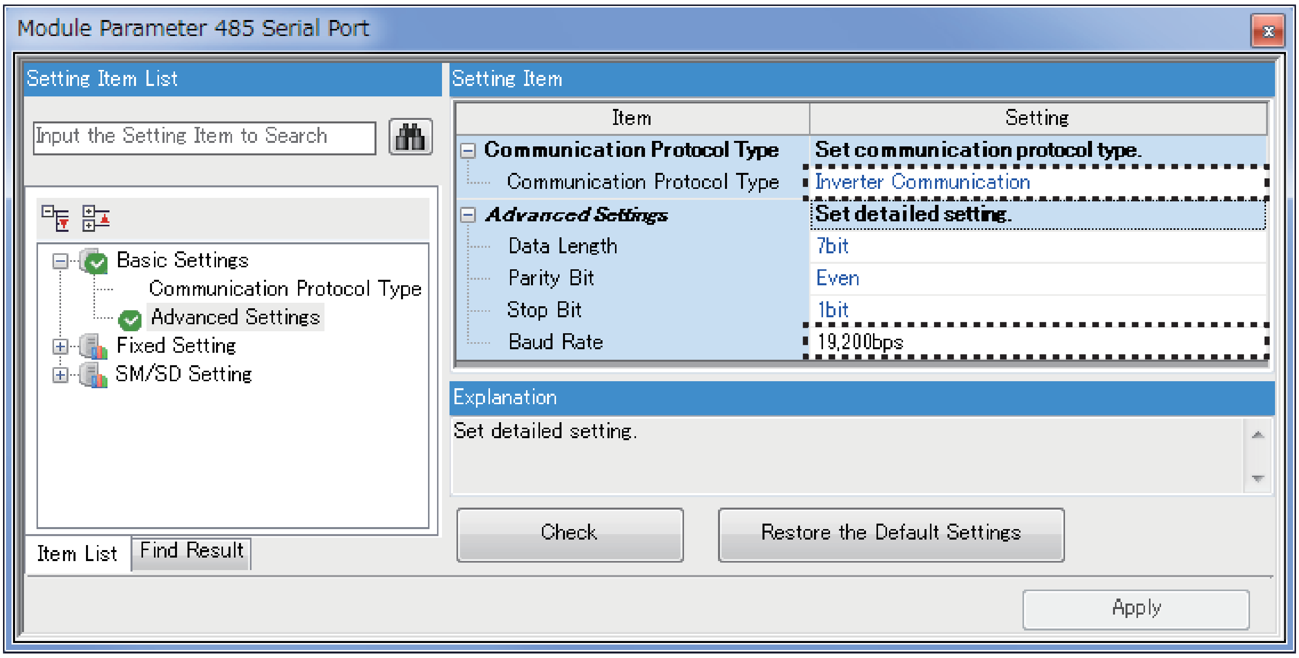
The following shows an example of a system configuration for linking the FX5U CPU module (CH1) with two inverters (Station number: 0 and 1).



Parameter setting

Set the "Communication Protocol Type" in the following parameter to "Inverter Communication", and set "Baud Rate" in the Advanced Settings to "19,200bps". Change the Advanced Settings according to the specifications of the inverter being used.

Navigation Window ⇨ Parameter ⇨ FX5UCPU ⇨ Module Parameter ⇨ 485 Serial Port

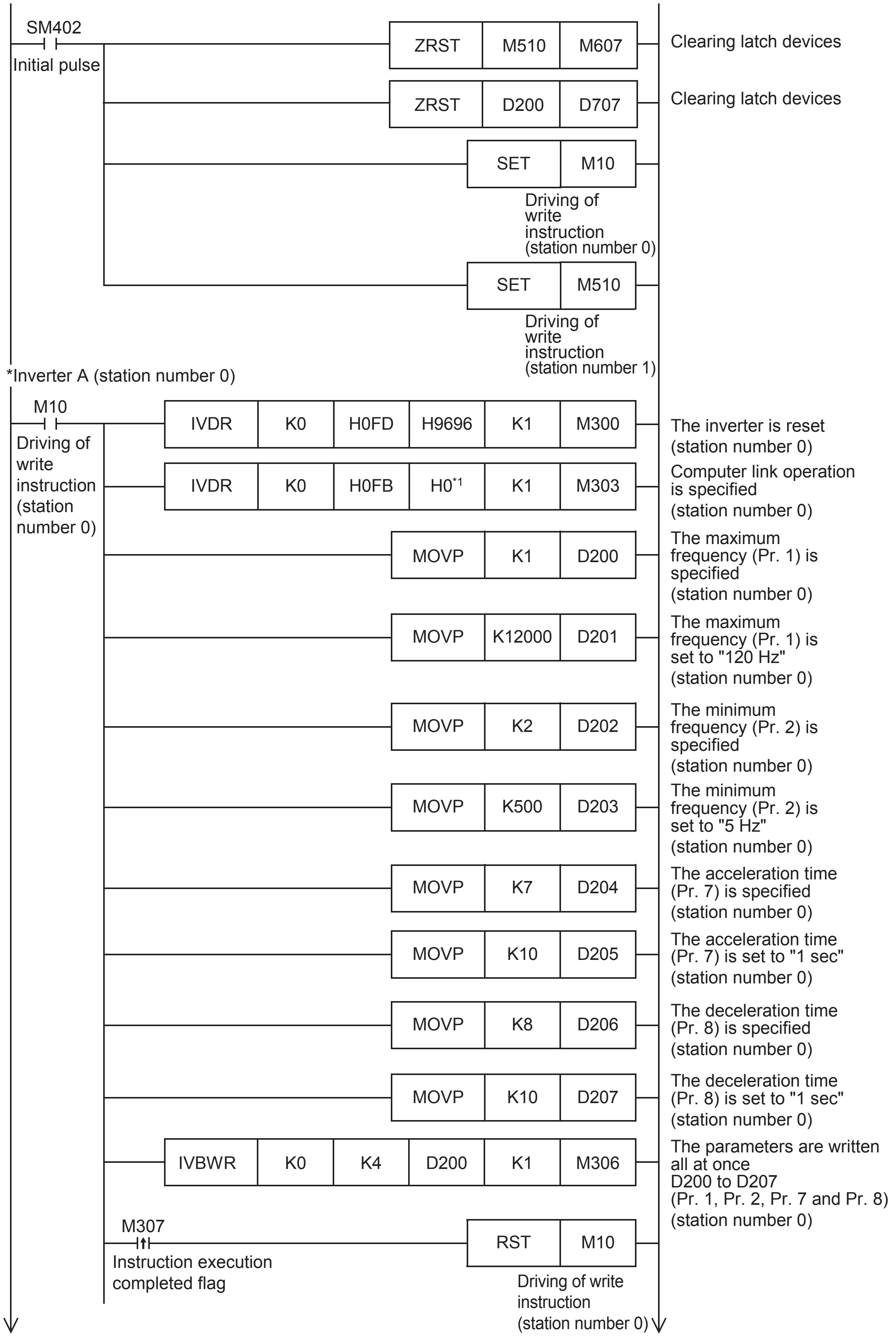


Item	Setting
Communication Protocol Type	**Set communication protocol type.**
Communication Protocol Type	Inverter Communication
Advanced Settings	**Set detailed setting.**
Data Length	7bit
Parity Bit	Even
Stop Bit	1bit
Baud Rate	19,200bps

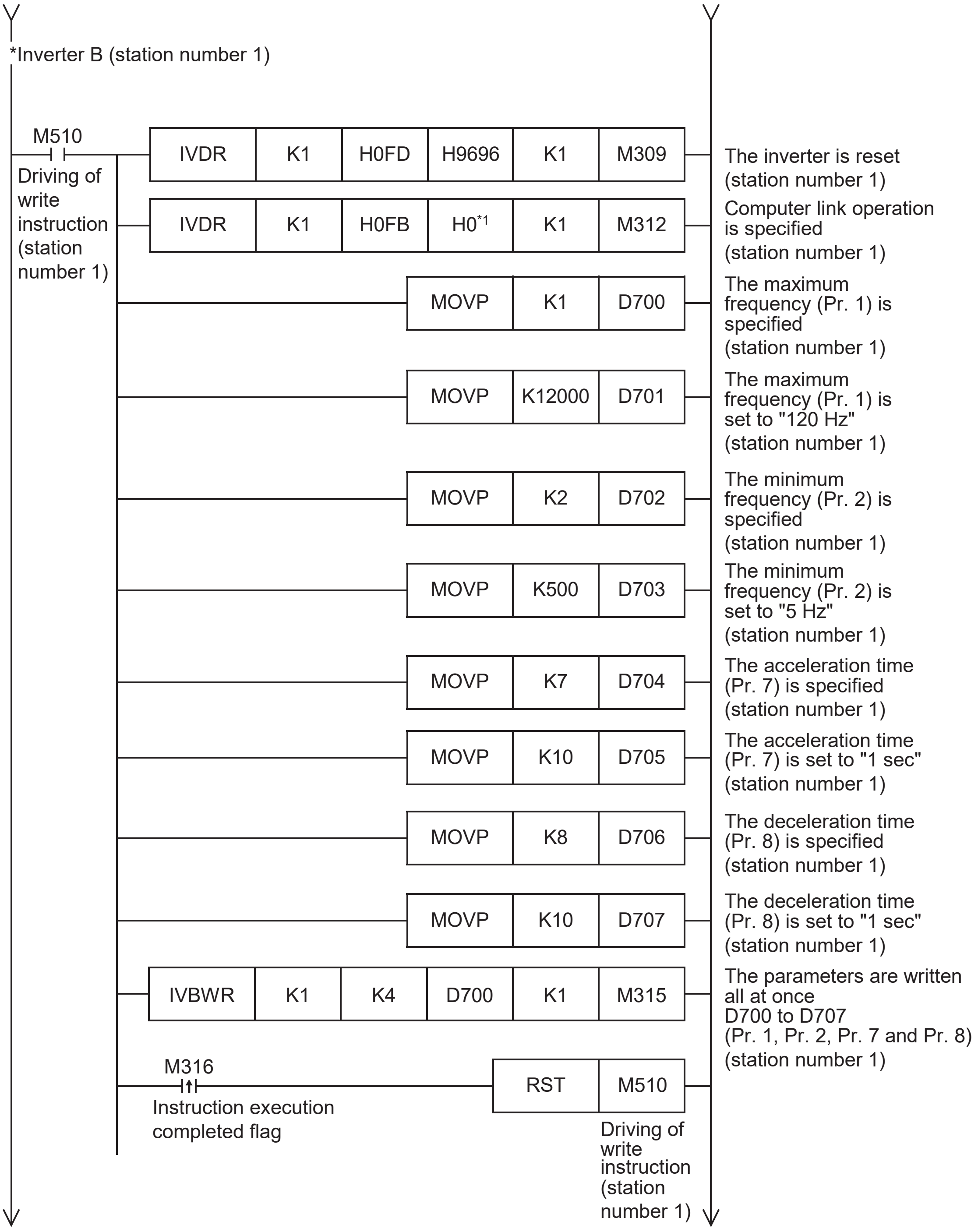
Set detailed setting.

■Writing parameters to an inverter while the PLC is in RUN mode

When M10 (or M510) turns ON, the parameters programmed in the inverter A (or inverter B) are written in. M10 (or M510) turns OFF when writing is completed.



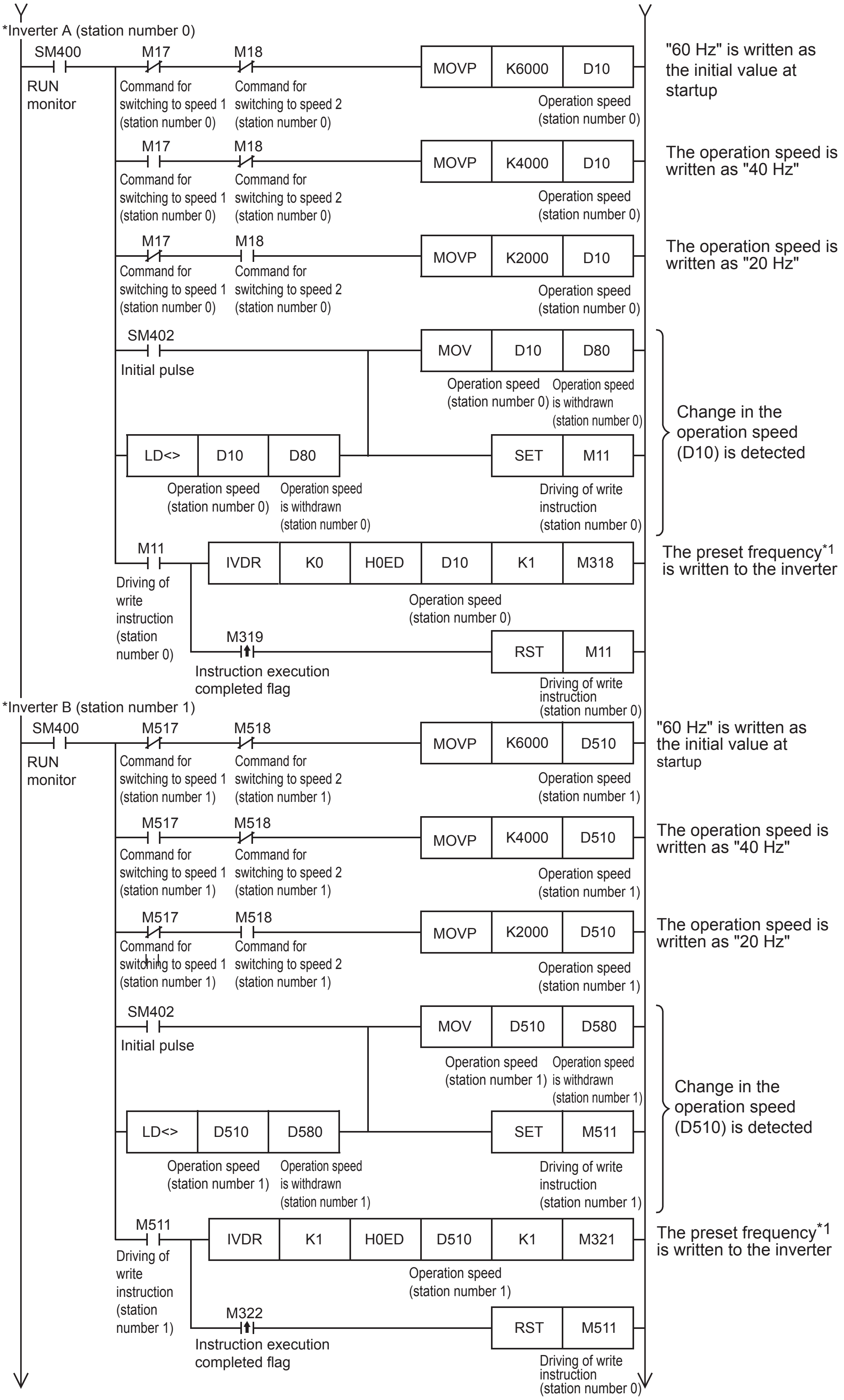
*1 When using an E700 Series inverter, use "H2" to specify computer link operation.



*1 When using an E700 Series inverter, use "H2" to specify computer link operation.

■Changing the speed using a sequence program

When M11 (or M511) turns ON, the operation speed stored in D10 (or D510) is written into the inverter A (or inverter B). M11 (or M511) turns OFF when writing is completed.

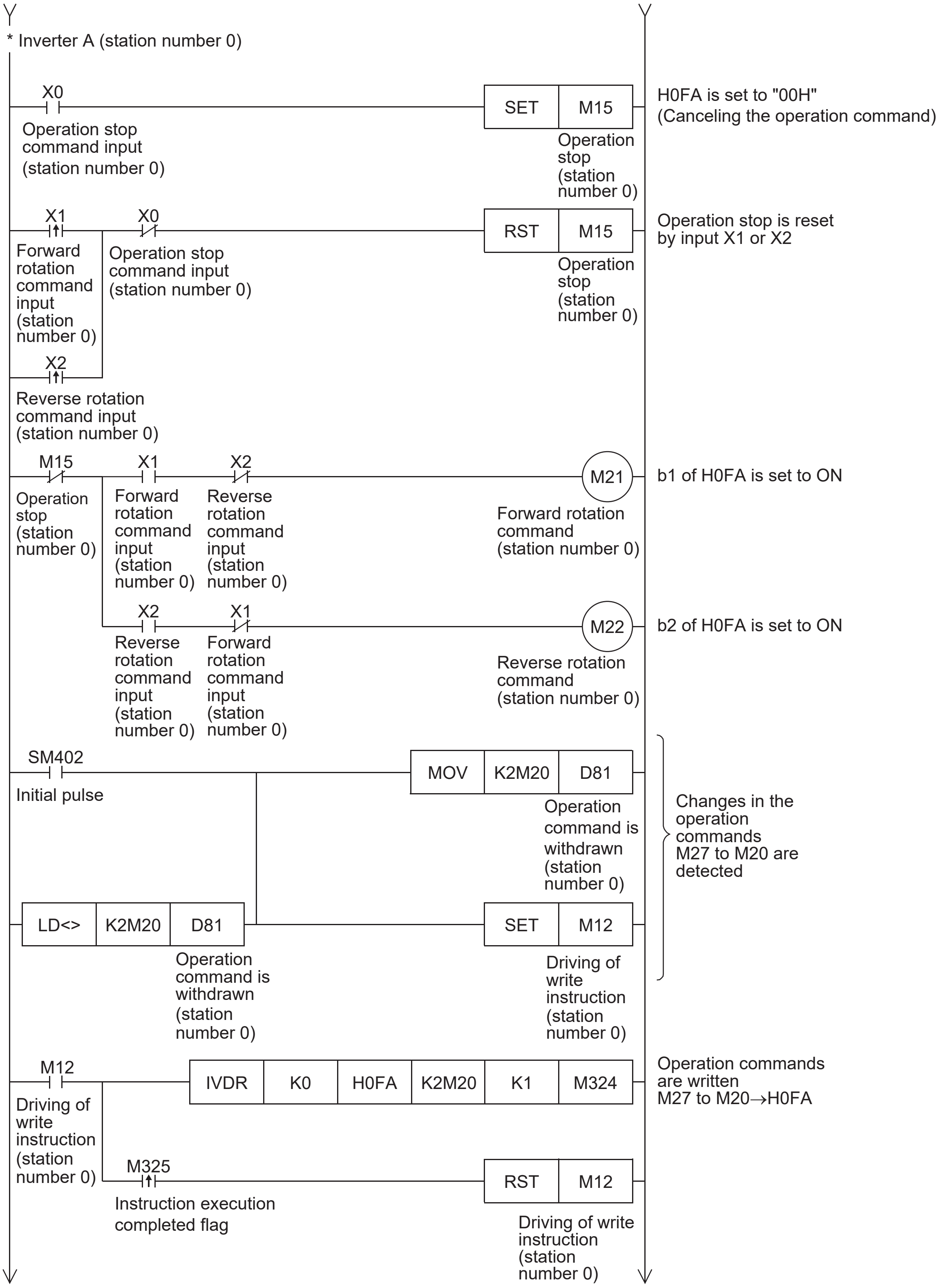


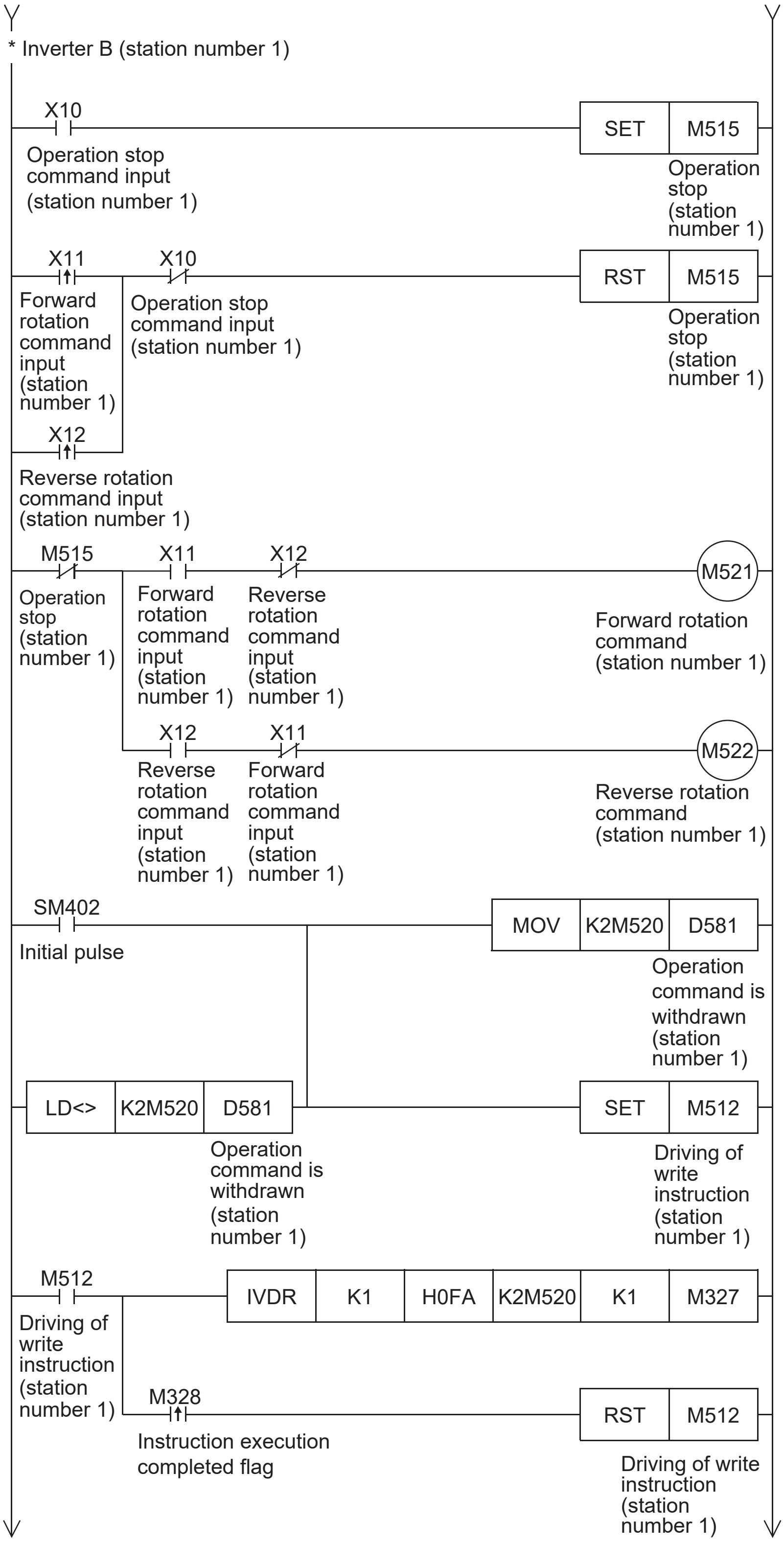
---

*1 For the V500 Series inverter, write "0" to instruction code HFF (Link parameter expansion setting) just before writing the set frequency in the program.

■Controlling operations of an inverter

When M12 (or M512) turns ON, the inverter operation command is written in. M12 (or M512) turns OFF when writing is completed.





H0FA is set to "00H"
(Canceling the operation command)

Operation stop is reset by input X11 or X12

b1 of H0FA is set to ON

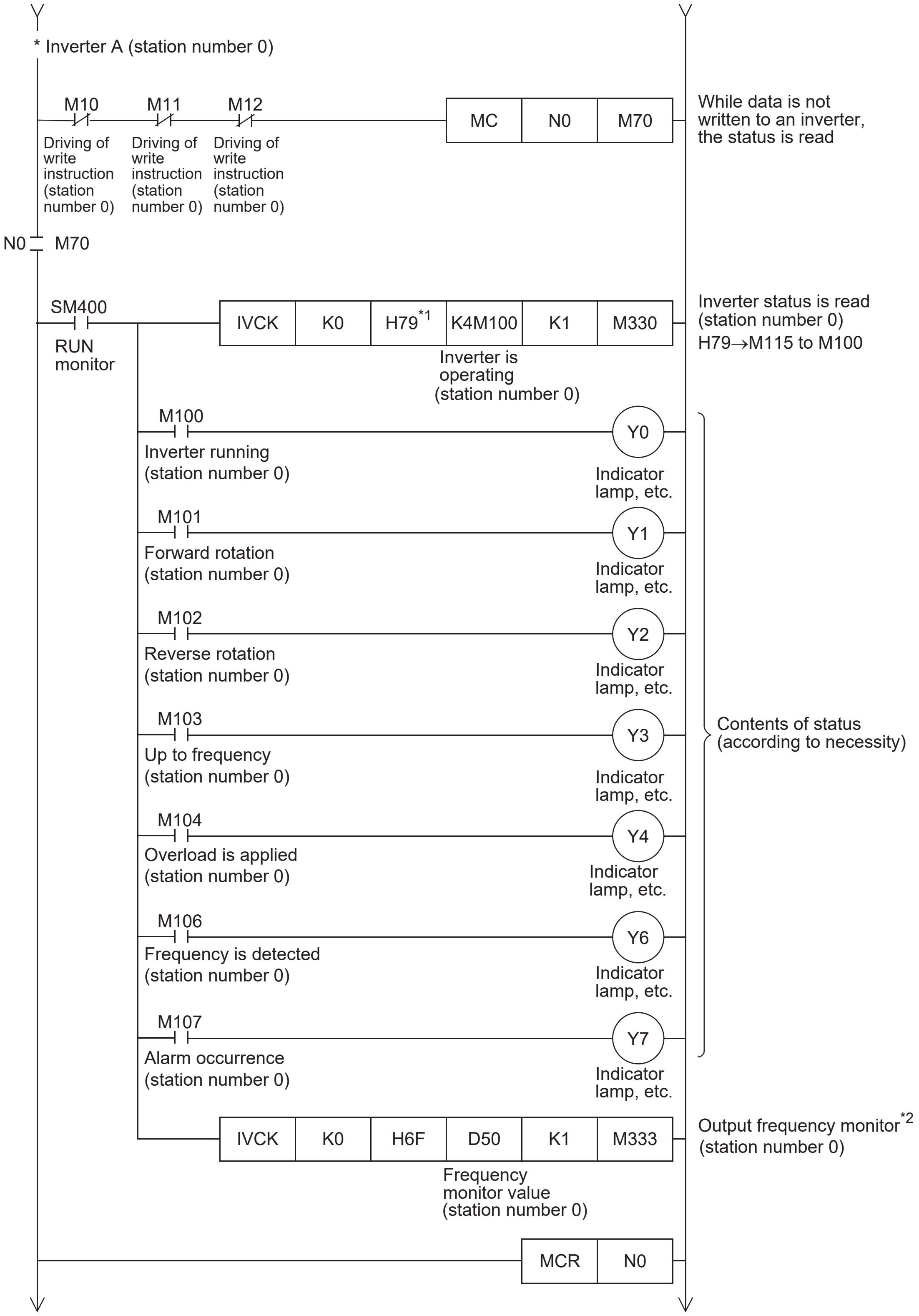
b2 of H0FA is set to ON

Changes in the operation commands M527 to M520 are detected

Operation commands are written M527 to M520→H0FA

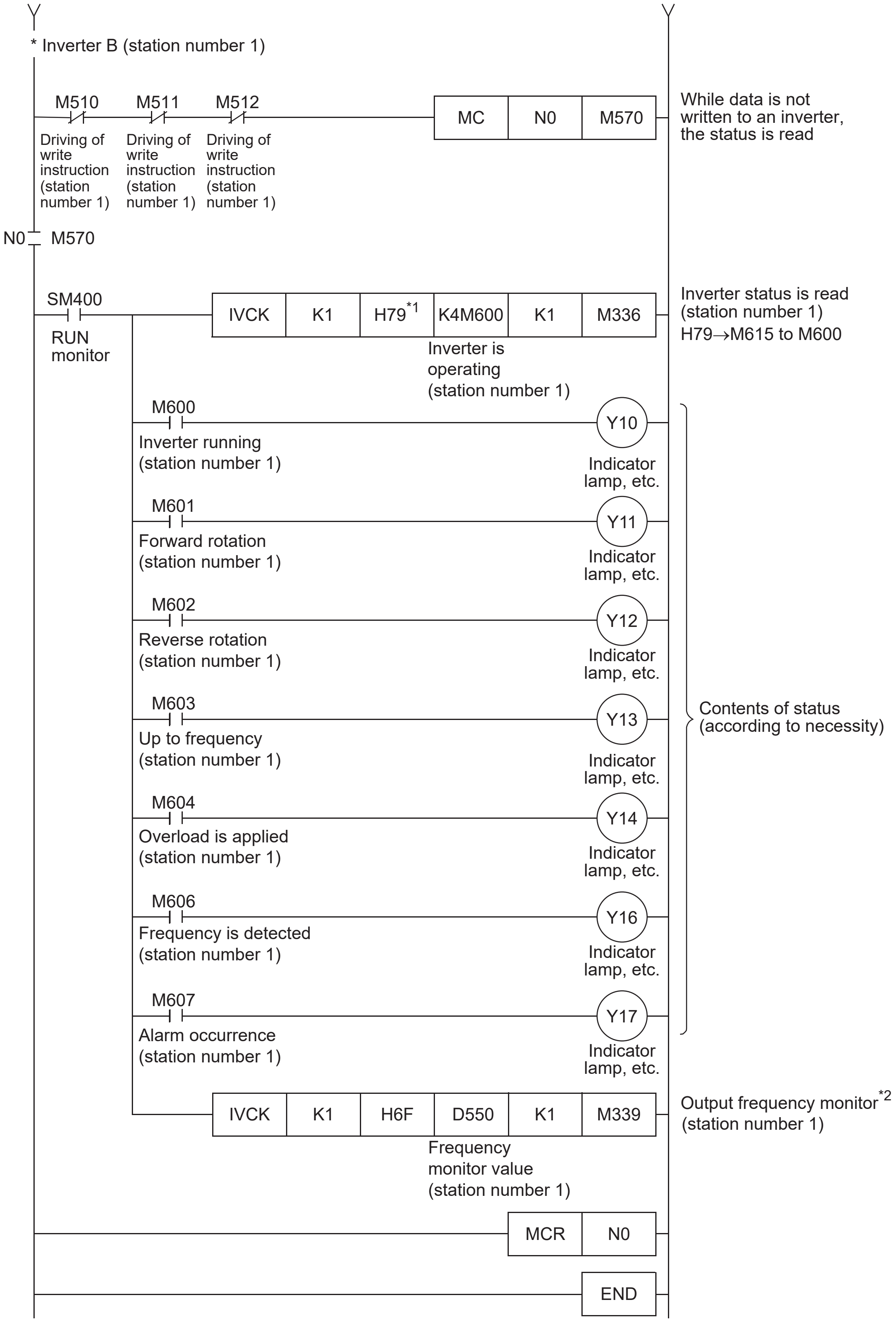
■Monitoring operations of an inverter

This program reads the status and output frequency written into the inverter A (or inverter B).



[1] V500 Series inverters do not support the command code H79 (Inverter status monitor (extended)). Use the command code H7A (Inverter status monitor).

[2] For the V500 Series inverter, write "0" to instruction code HFF (Link parameter expansion setting) just before reading the output frequency monitor in the program.



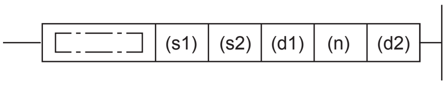
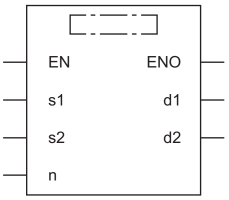
[1] V500 Series inverters do not support the command code H79 (Inverter status monitor (extended)). Use the command code H7A (Inverter status monitor).

[2] For the V500 Series inverter, write "0" to instruction code HFF (Link parameter expansion setting) just before reading the output frequency monitor in the program.

Inverter operation monitoring instruction

This instruction reads the operation status of an inverter.
For information concerning inverter communication instruction expressions and execution format, refer to MELSEC iQ-F FX5 Programming Manual (Instructions, Standard Functions/Function Blocks).

IVCK

Ladder	ST	FBD/LD
	ENO:=IVCK (EN, s1, s2, n, d1, d2);	

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Inverter station number	K0 to 31	16-bit signed binary	ANY16
(s2)	Inverter instruction codes	Refer to the following. ☞ Page 335	16-bit signed binary	ANY16
(d1)	Device number storing the read value	—	16-bit signed binary	ANY16
(n)*1	Communication CH	■FX5S/FX5UJ CPU module K2 to K4 ■FX5U CPU module K1 to K4 ■FX5UC CPU module K1, K3 to K4	16-bit unsigned binary	ANY16_U
(d2)*2	Head bit device number to output the instruction execution status	—	Bit	ANYBIT_ARRAY (Number of elements: 3)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 Specify a channel No. for which communication setting is set for inverter communication.

*2 Make sure not to use devices being used in other control operations. (☞ Page 366 Communication execution state output device)

■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	$	
(s1)	—	○*1	○	○	—	—	○	○	—	—	—
(s2)	—	○*1	○	○	—	—	○	○	—	—	—
(d1)	○	○	○	—	—	—	○	—	—	—	—
(n)	—	—	—	—	—	—	—	○	—	—	—
(d2)	○	○*1	—	—	—	—	—	—	—	—	—

*1 T, ST, C cannot be used.

Processing details

Write the operation status of inverter corresponding to instruction code (☞ Page 335 Inverter operation monitoring) of (s2) for station No. (s1) of the inverter connected to the communication channel (n) in (d1).

Operation error

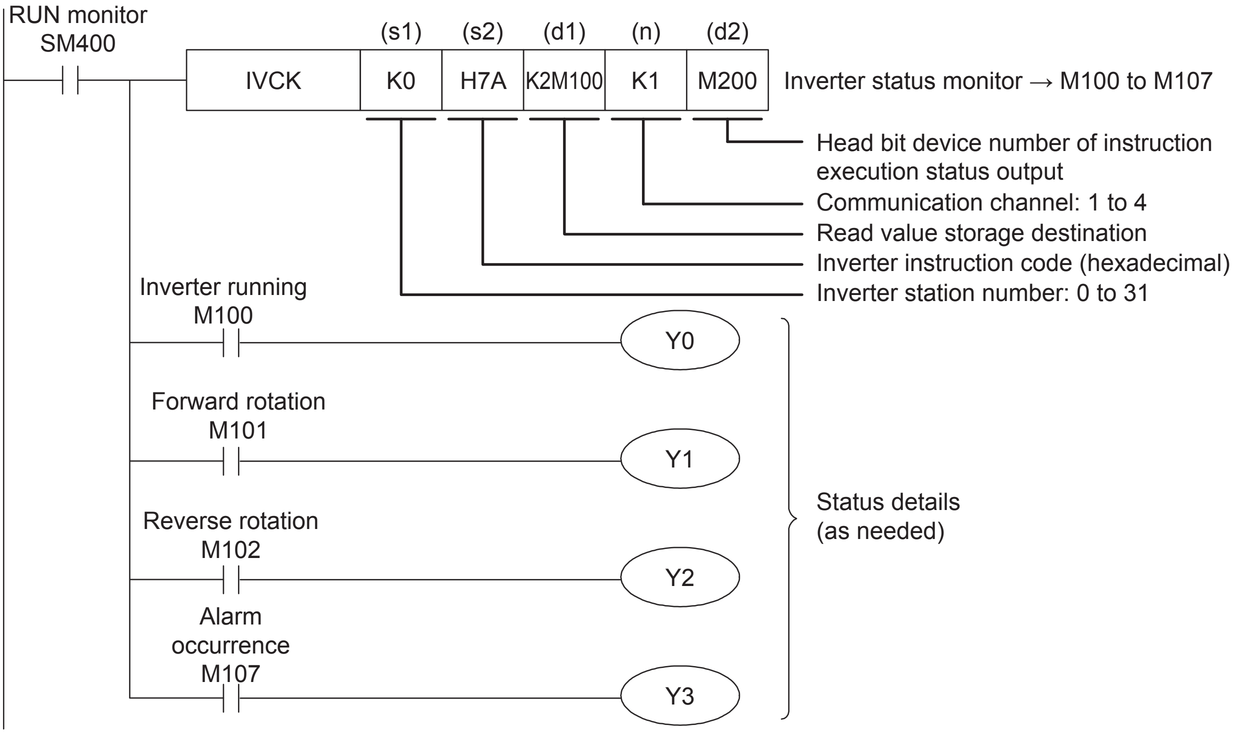
Operation errors pertaining to the instruction are as follows.

Operation error flag				Operation error code		Description
SM0	**SM1**	**SM56**	**SM8067**	**SD0**	**SD8067**	
ON				1810H		When specified channel is used with another instruction.
ON				2820H		When specified device exceeds the device range.
ON				3405H		When the value specified in (s1) is other than K0 to 31.
						When the value specified in (n) is other than the following channel numbers. • FX5S/FX5UJ CPU module K2 to K4 • FX5U/FX5UC CPU module K1 to K4
ON				3600H		When specified channel is not set with parameter.

Program example

Reads status (H7A) of inverter (station No.0) into CPU module (CH1), stores the read value in M100 to M107 and outputs externally (Y0 to Y3).

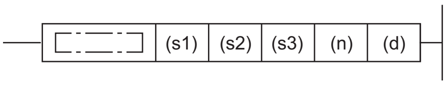
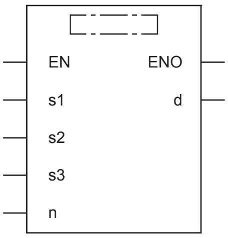
Read contents: Inverter operating = M100, normal rotation = M101, reverse rotation = M102, error occurrence = M107



Inverter operation control instruction

This instruction writes an inverter operation required set value to an inverter.
For information concerning inverter communication instruction expressions and execution format, refer to MELSEC iQ-F FX5 Programming Manual (Instructions, Standard Functions/Function Blocks).

IVDR

Ladder	ST	FBD/LD
	ENO:=IVDR (EN, s1, s2, s3, n, d);	

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Inverter station number	K0 to 31	16-bit signed binary	ANY16
(s2)	Inverter instruction codes	Refer to the following. ☞Page 336	16-bit signed binary	ANY16
(s3)	Set value to be written to the inverter parameter or device number storing the data to be set	—	16-bit signed binary	ANY16
(n)[*1]	Communication CH	■FX5S/FX5UJ CPU module K2 to K4 ■FX5U CPU module K1 to K4 ■FX5UC CPU module K1, K3 to K4	16-bit unsigned binary	ANY16_U
(d)[*2]	Head bit device number to output the instruction execution status	—	Bit	ANYBIT_ARRAY (Number of elements: 3)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

[*1] Specify a channel No. for which communication setting is set for inverter communication.

[*2] Make sure not to use devices being used in other control operations. (☞Page 366 Communication execution state output device)

■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	$	
(s1)	—	○[*1]	○	○	—	—	○	○	—	—	—
(s2)	—	○[*1]	○	○	—	—	○	○	—	—	—
(s3)	○	○	○	—	—	—	○	○	—	—	—
(n)	—	—	—	—	—	—	—	○	—	—	—
(d)	○	○[*1]	—	—	—	—	—	—	—	—	—

[*1] T, ST, C cannot be used.

Processing details

Writes set value of (s3) to the instruction code (☞Page 336 Inverter operation control) of (s2) for station No. (s1) of the inverter connected to the communication channel (n).

Operation error

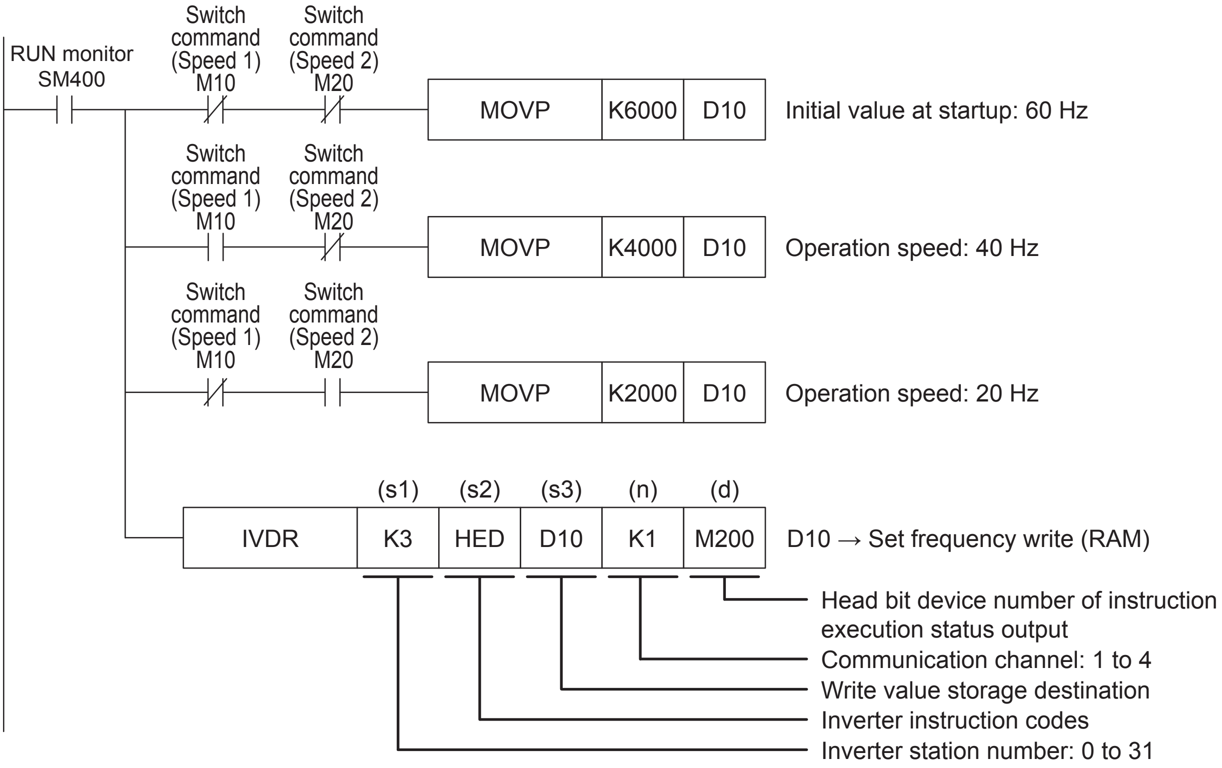
Operation errors pertaining to the instruction are as follows.

Operation error flag				Operation error code		Description
SM0	SM1	SM56	SM8067	SD0	SD8067	
ON				1810H		When specified channel is used with another instruction.
ON				2820H		When specified device exceeds the device range.
ON				3405H		When the value specified in (s1) is other than K0 to 31.
						When the value specified in (n) is other than the following channel numbers. • FX5S/FX5UJ CPU module K2 to K4 • FX5U/FX5UC CPU module K1 to K4
ON				3600H		When specified channel is not set with parameter.

Program example

Assuming that the initial value at startup is 60Hz, the CPU module (CH1) switches the operation speed (HED) of the inverter (Station No.3) between speed 1 (40Hz) and speed 2 (20Hz) by using switch commands.

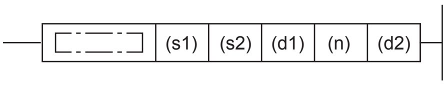
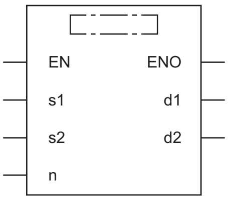
Write contents: D10 = Operation speed (initial value: 60Hz, speed 1: 40Hz, speed 2: 20Hz)



Inverter parameter read

This instruction reads an inverter parameter to the PLC.
For information concerning inverter communication instruction expressions and execution format, refer to MELSEC iQ-F FX5 Programming Manual (Instructions, Standard Functions/Function Blocks).

IVRD

Ladder	ST	FBD/LD
	ENO:=IVRD (EN, s1, s2, n, d1, d2);	

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Inverter station number	K0 to 31	16-bit signed binary	ANY16
(s2)	Inverter parameter number	Refer to the following. Page 336	16-bit signed binary	ANY16
(d1)	Device number storing the read value	—	16-bit signed binary	ANY16
(n)[*1]	Communication CH	■FX5S/FX5UJ CPU module K2 to K4 ■FX5U CPU module K1 to K4 ■FX5UC CPU module K1, K3 to K4	16-bit unsigned binary	ANY16_U
(d2)[*2]	Head bit device number to output the instruction execution status	—	Bit	ANYBIT_ARRAY (Number of elements: 3)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 Specify a channel No. for which communication setting is set for inverter communication.
*2 Make sure not to use devices being used in other control operations. (Page 366 Communication execution state output device)

■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	$	
(s1)	—	○[*1]	○	○	—	—	○	○	—	—	—
(s2)	—	○[*1]	○	○	—	—	○	○	—	—	—
(d1)	○	○	○	—	—	—	○	—	—	—	—
(n)	—	—	—	—	—	—	—	○	—	—	—
(d2)	○	○[*1]	—	—	—	—	—	—	—	—	—

*1 T, ST, C cannot be used.

Processing details

The value of the parameter No. (s2) is read to (d1) from station No. (s1) of inverter connected to communication channel (n).

Operation error

Operation errors pertaining to the instruction are as follows.

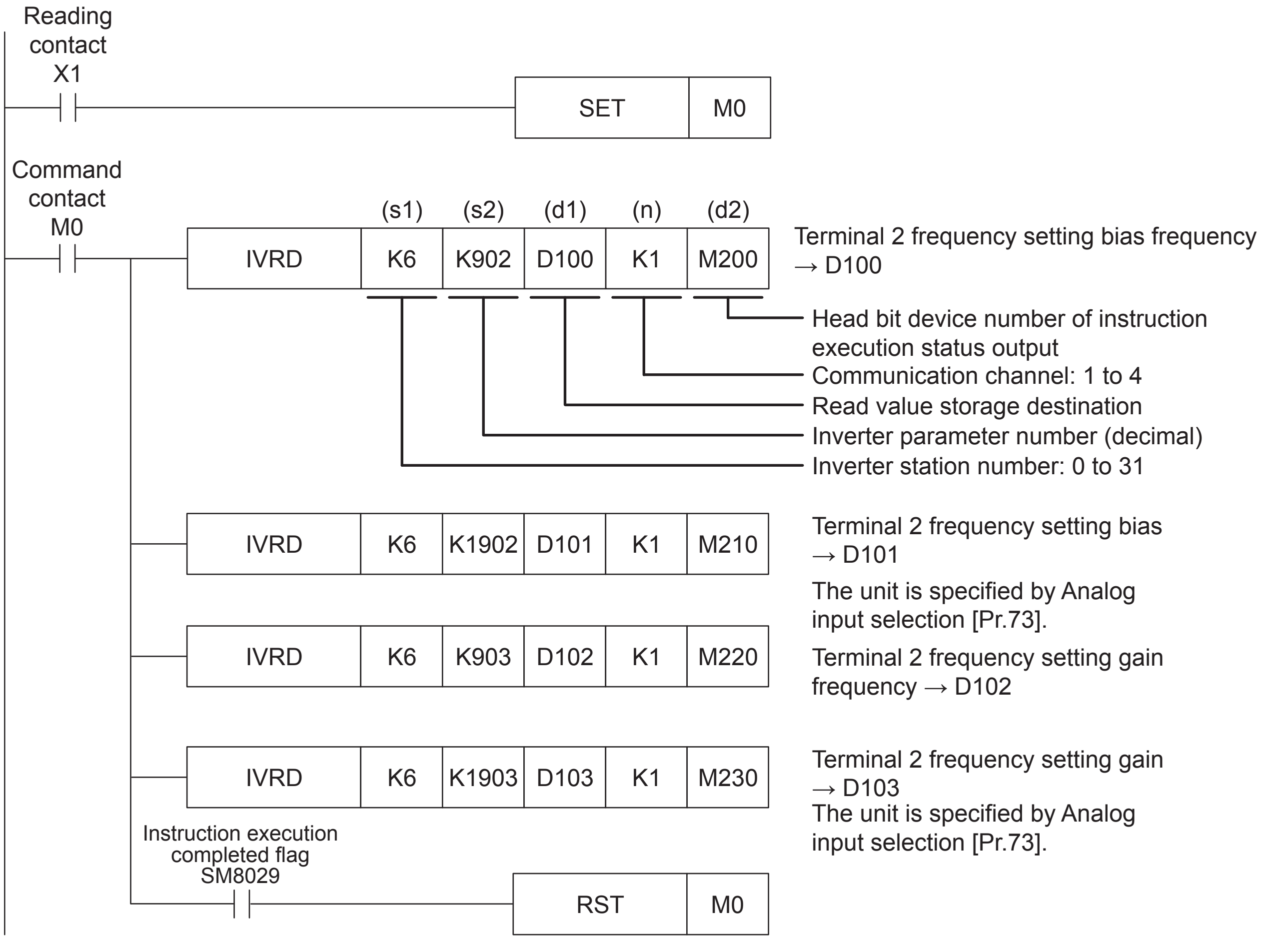
Operation error flag				Operation error code		Description
SM0	**SM1**	**SM56**	**SM8067**	**SD0**	**SD8067**	
ON				1810H		When specified channel is used with another instruction.
ON				2820H		When specified device exceeds the device range.
ON				3405H		When the value specified in (s1) is other than K0 to 31.
						When the value specified in (s2) is outside the range that can be specified. (Less than K0, K3000 to 9999, K13000 to 32767)
						When the value specified in (n) is other than the following channel numbers. • FX5S/FX5UJ CPU module K2 to K4 • FX5U/FX5UC CPU module K1 to K4
ON				3600H		When specified channel is not set with parameter.

Program example

Reads value of the following inverter (station No.6) parameters to storage device for CPU module (CH1).

Example of program using second parameter specification code (☞ Page 398 Second parameter specification code) of FREQROL-F700P series inverter.

Parameter No.	Name	Second parameter specification code	Storage devices
C2	Terminal 2 frequency setting bias frequency	902	D100
C3	Terminal 2 frequency setting bias	1902	D101
125	Terminal 2 frequency setting gain frequency	903	D102
C4	Terminal 2 frequency setting gain	1903	D103



Terminal 2 frequency setting bias frequency → D100

Terminal 2 frequency setting bias → D101

The unit is specified by Analog input selection [Pr.73].

Terminal 2 frequency setting gain frequency → D102

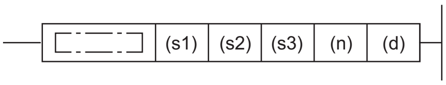
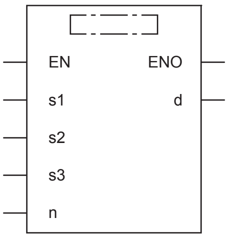
Terminal 2 frequency setting gain → D103

The unit is specified by Analog input selection [Pr.73].

Inverter parameter write

This instruction writes a value from the PLC to a parameter in an inverter.
For information concerning inverter communication instruction expressions and execution format, refer to 📖MELSEC iQ-F FX5 Programming Manual (Instructions, Standard Functions/Function Blocks).

IVWR

Ladder	ST	FBD/LD
	ENO:=IVWR (EN, s1, s2, s3, n, d);	

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Inverter station number	K0 to 31	16-bit signed binary	ANY16
(s2)	Inverter parameter number	Refer to the following. ☞ Page 336	16-bit signed binary	ANY16
(s3)	Set value to be written to the inverter parameter or device number storing the data to be set	—	16-bit signed binary	ANY16
(n)[*1]	Communication CH	■FX5S/FX5UJ CPU module K2 to K4 ■FX5U CPU module K1 to K4 ■FX5UC CPU module K1, K3 to K4	16-bit unsigned binary	ANY16_U
(d)[*2]	Head bit device number to output the instruction execution status	—	Bit	ANYBIT_ARRAY (Number of elements: 3)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 Specify a channel No. for which communication setting is set for inverter communication.
*2 Make sure not to use devices being used in other control operations. (☞ Page 366 Communication execution state output device)

■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	$	
(s1)	—	○[*1]	○	○	—	—	○	○	—	—	—
(s2)	—	○[*1]	○	○	—	—	○	○	—	—	—
(s3)	○	○	○	—	—	—	○	○	—	—	—
(n)	—	—	—	—	—	—	—	○	—	—	—
(d)	○	○[*1]	—	—	—	—	—	—	—	—	—

*1 T, ST, C cannot be used.

Processing details

Writes value of (s3) in parameter No. (s2) of station No. (s1) of inverter connected to communication channel (n).

Operation error

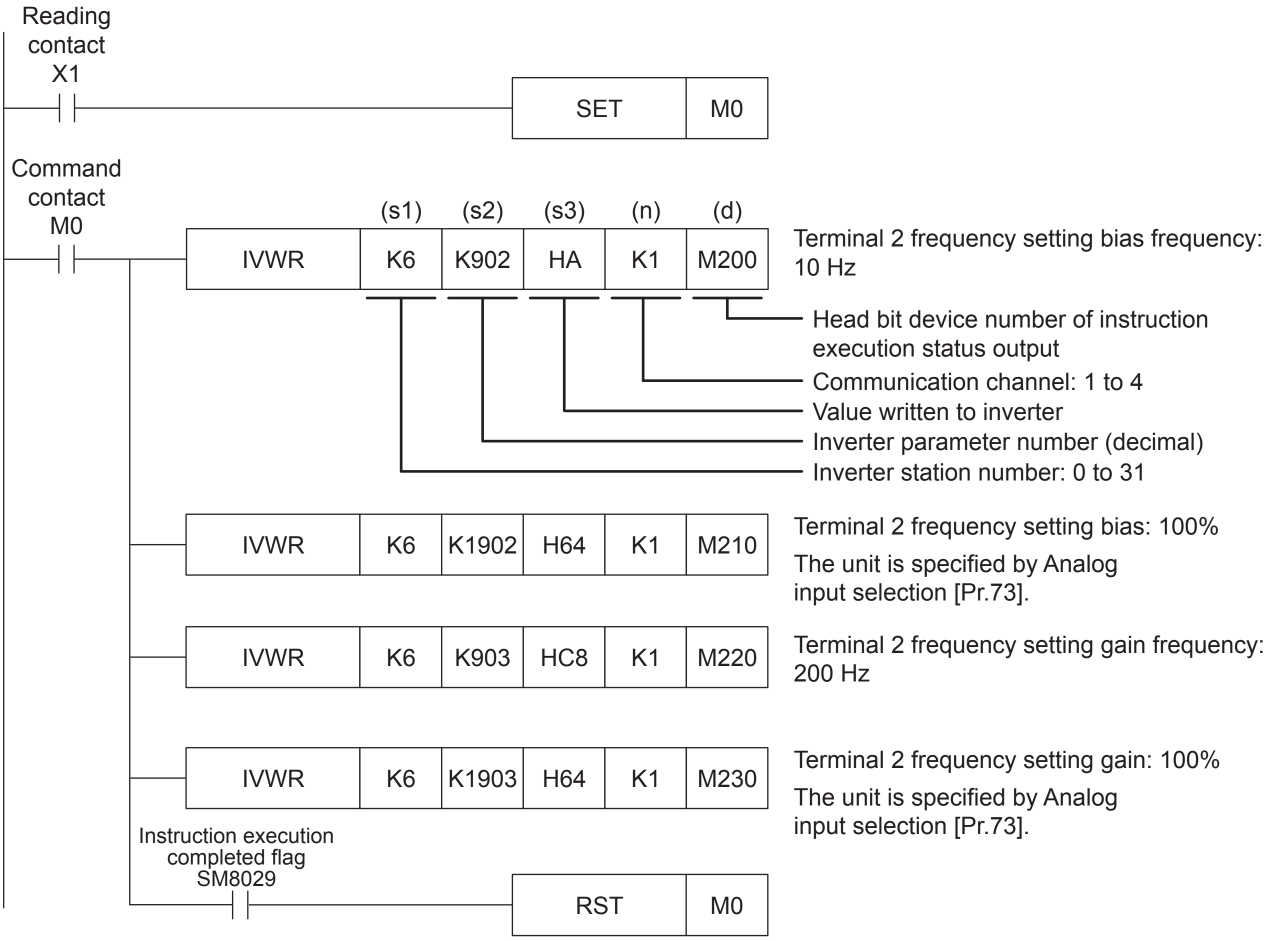
Operation errors pertaining to the instruction are as follows.

Operation error flag				**Operation error code**		**Description**
SM0	**SM1**	**SM56**	**SM8067**	**SD0**	**SD8067**	
ON				1810H		When specified channel is used with another instruction.
ON				2820H		When specified device exceeds the device range.
ON				3405H		When the value specified in (s1) is other than K0 to 31.
						When the value specified in (s2) is outside the range that can be specified. (Less than K0, K3000 to 9999, K13000 to 32767)
						When the value specified in (n) is other than the following channel numbers. • FX5S/FX5UJ CPU module K2 to K4 • FX5U/FX5UC CPU module K1 to K4
ON				3600H		When specified channel is not set with parameter.

Program example

Writes setting values to parameter shown in the following table for inverter (station No.6) from the CPU module (CH1).
Example of program using second parameter specification code (☞ Page 398 Second parameter specification code) of FREQROL-F700P series inverter.

Parameter No.	Name	Second parameter specification code	Setting values to be written
C2	Terminal 2 frequency setting bias frequency	902	10[Hz]
C3	Terminal 2 frequency setting bias	1902	100[%]
125	Terminal 2 frequency setting gain frequency	903	200[Hz]
C4	Terminal 2 frequency setting gain	1903	100[%]

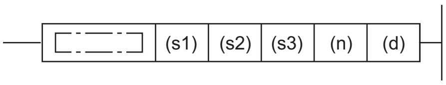
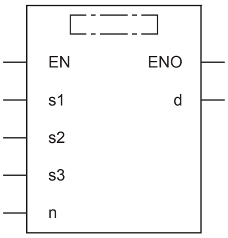


Inverter parameter block write

This instruction writes parameters of an inverter at one time

For information concerning inverter communication instruction expressions and execution format, refer to 🕮MELSEC iQ-F FX5 Programming Manual (Instructions, Standard Functions/Function Blocks).

IVBWR

Ladder	ST	FBD/LD
	ENO:=IVBWR (EN, s1, s2, s3, n, d);	

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Inverter station number	K0 to 31	16-bit signed binary	ANY16
(s2)	Number of parameters in an inverter to be written at one time	—	16-bit signed binary	ANY16
(s3)	Head device number of a parameter table to be written to an inverter	—	16-bit signed binary	ANY16
(n)[*1]	Communication CH	■FX5S/FX5UJ CPU module K2 to K4 ■FX5U CPU module K1 to K4 ■FX5UC CPU module K1, K3 to K4	16-bit unsigned binary	ANY16_U
(d)[*2]	Head bit device number to output the instruction execution status	—	Bit	ANYBIT_ARRAY (Number of elements: 3)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

[*1] Specify a channel No. for which communication setting is set for inverter communication.

[*2] Make sure not to use devices being used in other control operations. (☞ Page 366 Communication execution state output device)

■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	$	
(s1)	—	○[*1]	○	○	—	—	○	○	—	—	—
(s2)	—	○[*1]	○	○	—	—	○	○	—	—	—
(s3)	—	○	○	—	—	—	○	—	—	—	—
(n)	—	—	—	—	—	—	—	○	—	—	—
(d)	○	○[*1]	—	—	—	—	—	—	—	—	—

[*1] T, ST, C cannot be used.

Processing details

Continuously writes parameter numbers and their values (2 words/parameter) for the number of points specified by (s2) starting from word device specified by (s3) for station No. (s1) of inverter connected to communication channel (n) (no limit to number of points that can be written).

Ex.

Writing contents for (s2): K8, (s3): D200

(s3)	D200	Parameter No. 1
(s3)+1	D201	Parameter 1 value
(s3)+2	D202	Parameter No. 2
(s3)+3	D203	Parameter 2 value
⋮	⋮	⋮
(s3)+14	D214	Parameter No. 8
(s3)+15	D215	Parameter 8 value

(s2)×2 = Number of occupied word devices

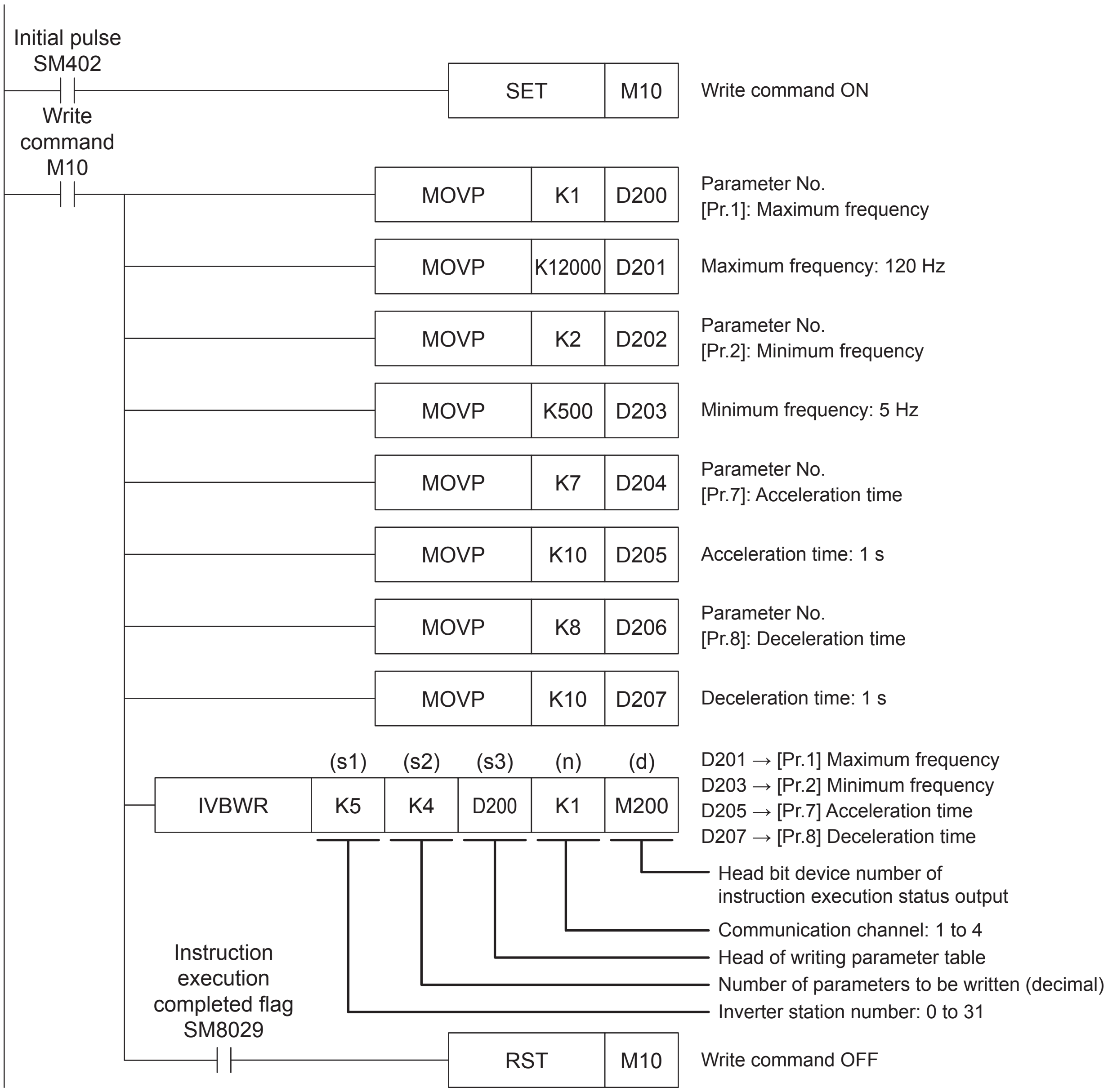
Operation error

Operation errors pertaining to the instruction are as follows.

Operation error flag				Operation error code		Description
SM0	**SM1**	**SM56**	**SM8067**	**SD0**	**SD8067**	
ON				1810H		When specified channel is used with another instruction.
ON				2820H		When specified device exceeds the device range.
ON				3405H		When the value specified in (s1) is other than K0 to 31.
						When the value specified in (s2) is less than K0.
						When the value specified in (s3), ..., (s3) +2 ([s2] -1) is outside the range that can be specified. (Less than K0, K3000 to 9999, K13000 to 32767)
						When the value specified in (n) is other than the following channel numbers. • FX5S/FX5UJ CPU module K2 to K4 • FX5U/FX5UC CPU module K1 to K4
ON				3600H		When specified channel is not set with parameter.

Program example

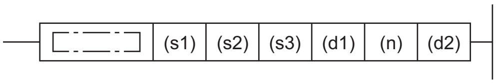
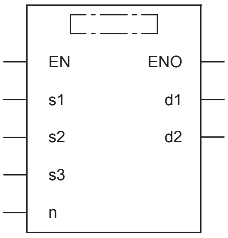
Writes upper limit frequency (Pr.1): 120Hz, lower limit frequency (Pr.2): 5Hz, acceleration time (Pr.7): 1 second, deceleration time (Pr.8):1 second for inverter (station No.5) from CPU module (CH1).
Write contents: Parameter No.1 = D200, 2 = D202, 7 = D204, 8 = D206, upper limit frequency = D201, lower limit frequency = D203, acceleration time = D205, deceleration time = D207



Multiple inverter commands

This instruction writes 2 types of settings (operation command and set frequency) to the inverter, and reads 2 types of data (inverter status monitor, output frequency, etc.) from the inverter at the same time.
For information concerning inverter communication instruction expressions and execution format, refer to 🕮MELSEC iQ-F FX5 Programming Manual (Instructions, Standard Functions/Function Blocks).

IVMC

Ladder	ST	FBD/LD
	ENO:=IVMC (EN, s1, s2, s3, n, d1, d2);	

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Inverter station number	K0 to 31	16-bit signed binary	ANY16
(s2)	Multiple instructions for inverter:	☞ Page 395 Send/ receive data type	16-bit signed binary	ANY16
(s3)[*1]	Head device of a parameter data to be written to an inverter	—	16-bit signed binary	ANY16_ARRAY (Number of elements: 2)
(d1)[*1]	Head device which stores values to be read from the inverter	—	16-bit signed binary	ANY16_ARRAY (Number of elements: 2)
(n)[*2]	Communication CH	■FX5S/FX5UJ CPU module K2 to K4 ■FX5U CPU module K1 to K4 ■FX5UC CPU module K1, K3 to K4	16-bit unsigned binary	ANY16_U
(d2)[*1]	Head bit device number to output the instruction execution status	—	Bit	ANYBIT_ARRAY (Number of elements: 3)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 Specify a channel No. for which communication setting is set for inverter communication. (☞ Page 366 Communication execution state output device)

*2 Specify a channel No. for which communication setting is set for inverter communication.

■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	$	
(s1)	—	○[*1]	○	○	—	—	○	○	—	—	—
(s2)	—	○[*1]	○	○	—	—	○	○	—	—	—
(s3)	—	○	○	—	—	—	○	—	—	—	—
(d1)	—	○	○	—	—	—	○	—	—	—	—
(n)	—	—	—	—	—	—	—	○	—	—	—
(d2)	○	○[*1]	—	—	—	—	—	—	—	—	—

*1 T, ST, C cannot be used.

Processing details

This instruction executes multiple commands to an inverter connected to the communication channel (n) whose station number is specified in (s1). Specify the send/receive data type using (s2), the head device which stores data to be written to the inverter using (s3), and the head device which stores values to be read from the inverter using (d1).

■Send/receive data type

The table below shows valid send data 1 and 2 and receive data 1 and 2 specified by the (s2) send/receive data type.

(s2) Send/receive data type (HEX)	Send data (Written to the Inverter)		Receive data (Read from Inverter)	
	Data 1 (s3)	Data 2 ((s3) +1)	Data 1 (d1)	Data 2 ((d1) +1)
0000H	Run command (expansion)	Set frequency (RAM)	Inverter status monitor (expansion)	Output frequency (speed)
0001H				Special monitor
0010H		Set frequency (RAM, EEPROM)		Output frequency (speed)
0011H				Special monitor

Precautions

- If a device number outside the range due to cases such as indexing is specified in (d1), the receive data from the inverter is not stored in (d1). However, values set in (s3) and (s3)+1 may be written to the inverter.
- If any unspecified value is set in (s2), unexpected data may be written to and read from the inverter, and values of (d1) and (d1)+1 may be updated.
- The IVMC instruction reads the inverter status at the time of communication with the inverter, and stores it in (d1). Accordingly, the inverter status written by the IVMC instruction can be read when the next reading instruction (IVCK, IVMC, etc.) is executed.

Operation error

Operation errors pertaining to the instruction are as follows.

Operation error flag				Operation error code		Description
SM0	SM1	SM56	SM8067	SD0	SD8067	
ON				1810H		When specified channel is used with another instruction.
ON				2820H		When specified device exceeds the device range.
ON				3405H		When the value specified in (s1) is other than K0 to 31.
						When the value specified in (n) is other than the following channel numbers. • FX5S/FX5UJ CPU module K2 to K4 • FX5U/FX5UC CPU module K1 to K4
ON				3600H		When specified channel is not set with parameter.

Applicable inverters

This instruction is applicable to the following inverters:

- FREQROL-E700 series (February 2009 and later)
- FREQROL-F800/A800/F700PJ/F700P/E700EX/D700 Series

Program example

Writes (s3): run command (expansion), (s3) +1: setting frequency (RAM) and reads (d1): inverter status monitor (expansion), (d1) +1: output frequency (rotation speed) for inverter (station No.0) from CPU module (CH1).

Send/receive type code: H0000

- (s3): Run command (expansion)

Runs inverter in forward or reverse by forward rotation command (M21)/reverse rotation command (M22).

Write contents: D10 = run command (M21 = forward rotation, M22 = reverse rotation)

- (s3) + 1: Set frequency (RAM)

Switches speed 1 (40Hz)/speed 2 (20Hz) with 60Hz as the initial value when starting up.

Write contents: D11 = Operation speed (initial value: 60Hz, speed 1: 40Hz, speed 2: 20Hz)

- (d1): Inverter status monitor (expansion)

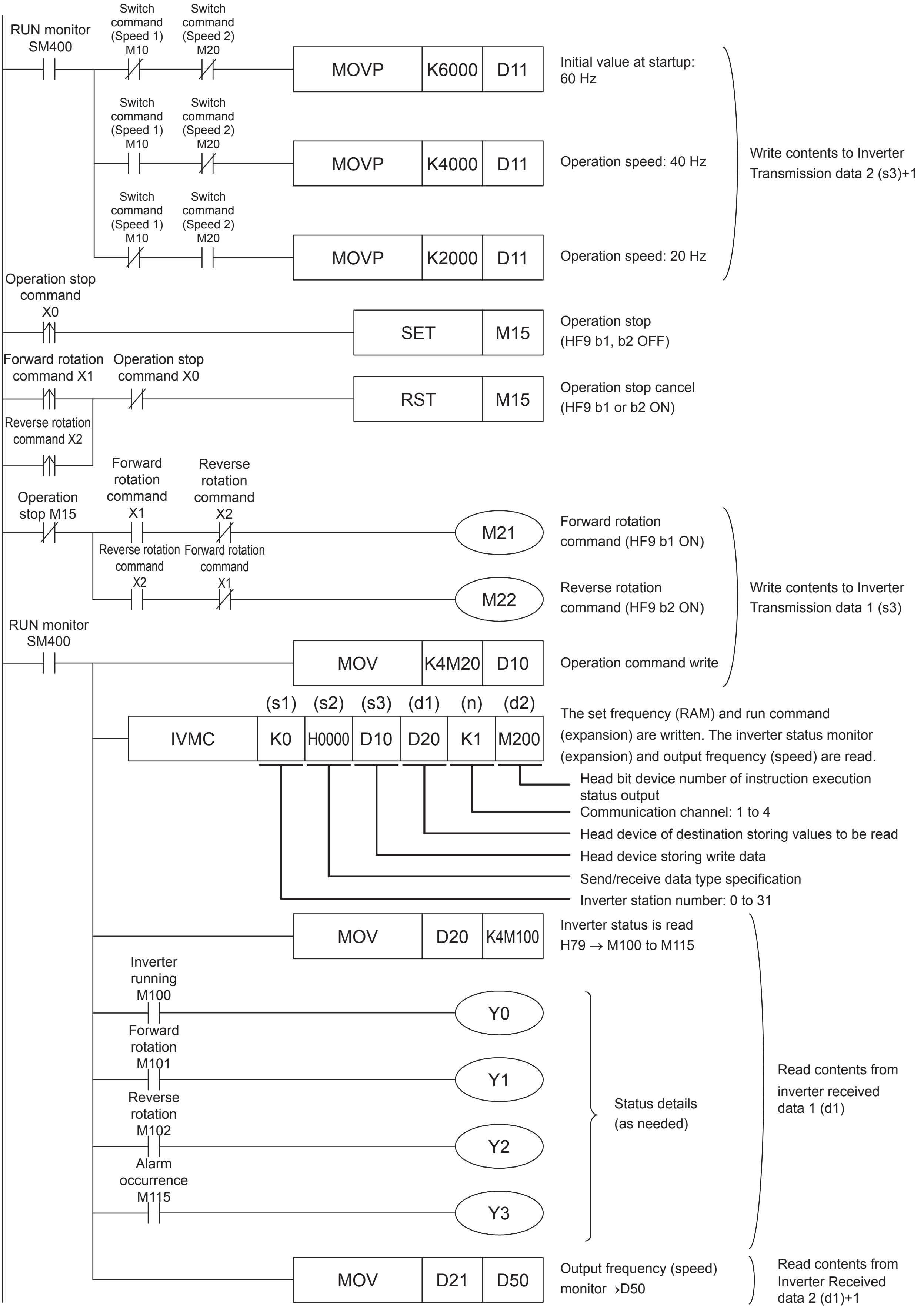
Stores read value in M100 to M115 and outputs externally (Y0 to Y3).

Read contents: D20 = inverter monitor (expansion) (inverter running = M100, forward rotation = M101, reverse rotation = M102, error = M115)

- (d1) + 1: Output frequency (speed)

Reads output frequency (rotation speed).

Read contents: D21 = Output frequency (rotation speed)



Second parameter specification code

When handling the following parameters in inverter communication, it is necessary to select second parameters. In IVRD, IVWR, and IVBWR instructions, when a value shown in the tables below is set in (s2) ((s3) in IVBWR instruction), the extension parameter and second parameter are automatically overwritten, and parameter values are either read or written. For details concerning inverter parameters, refer to the manual for each particular inverter.

FREQROL-F800 series

Second parameter specification codes for parameter numbers Pr.125, Pr.126, C2 to C19 and C38 to C41

Parameter No.	Name	Second parameter specification code
C2	Terminal 2 frequency setting bias frequency	902
C3	Terminal 2 frequency setting bias	1902
125	Terminal 2 frequency setting gain frequency	903
C4	Terminal 2 frequency setting gain	1903
C5	Terminal 4 frequency setting bias frequency	904
C6	Terminal 4 frequency setting bias	1904
126	Terminal 4 frequency setting gain frequency	905
C7	Terminal 4 frequency setting gain	1905
C12	Bias frequency (speed) for terminal No.1	917
C13	Bias (speed) for terminal No.1	1917
C14	Gain frequency (speed) for terminal No.1	918
C15	Gain (speed) for terminal No.1	1918
C16	Bias command (torque) for terminal No.1	919
C17	Bias (torque) for terminal No.1	1919
C18	Gain command (torque) for terminal No.1	920
C19	Gain (torque) for terminal No.1	1920
C8	Current output bias signal	930
C9	Current output bias current	1930
C10	Current output gain signal	931
C11	Current output gain current	1931
C38	Bias command (torque) for terminal No.4	932
C39	Bias (torque) for terminal No.4	1932
C40	Gain command (torque) for terminal No.4	933
C41	Gain (torque) for terminal No.4	1933
C42	PID display bias coefficient	934
C43	PID display bias analog value	1934
C44	PID display gain coefficient	935
C45	PID display gain analog value	1935

FREQROL-E800 series

Second parameter specification codes for parameter numbers Pr.125, Pr.126, C2 to C7 and C38 to C45

Parameter No.	Name	Second parameter specification code
C2	Terminal 2 frequency setting bias (frequency)	902
C3	Terminal 2 frequency setting bias (analog value)	1902
125	Terminal 2 frequency setting gain (frequency)	903
C4	Terminal 2 frequency setting gain (analog value)	1903
C5	Terminal 4 frequency setting bias (frequency)	904
C6	Terminal 4 frequency setting bias (analog value)	1904
126	Terminal 4 frequency setting gain (frequency)	905
C7	Terminal 4 frequency setting gain (analog value)	1905
C3	Terminal 2 frequency setting bias (terminal analog value)	2902
C4	Terminal 2 frequency setting gain (terminal analog value)	2903
C6	Terminal 4 frequency setting bias (terminal analog value)	2904
C7	Terminal 4 frequency setting gain (terminal analog value)	2905
C38	Terminal 4 bias command (torque/ magnetic flux)	932
C39	Terminal 4 bias (torque/magnetic flux)	1932
C40	Terminal 4 gain command (torque/ magnetic flux)	933
C41	Terminal 4 gain (torque/magnetic flux)	1933
C39	Terminal 4 bias (torque/magnetic flux) (terminal analog value)	2932
C41	Terminal 4 gain (torque/magnetic flux) (terminal analog value)	2933
C42	PID display bias coefficient	934
C43	PID display bias analog value	1934
C44	PID display gain coefficient	935
C45	PID display gain analog value	1935
C43	PID display bias analog value (terminal analog value)	2934
C45	PID display gain analog value (terminal analog value)	2935

FREQROL-A800/A800 Plus series

Second parameter specification codes for parameter numbers Pr.125, Pr.126, C2 to C19 and C38 to C41

Parameter No.	Name	Second parameter specification code
C2	Terminal 2 frequency setting bias frequency	902
C3	Terminal 2 frequency setting bias	1902
125	Terminal 2 frequency setting gain frequency	903
C4	Terminal 2 frequency setting gain	1903
C5	Terminal 4 frequency setting bias frequency	904
C6	Terminal 4 frequency setting bias	1904
126	Terminal 4 frequency setting gain frequency	905
C7	Terminal 4 frequency setting gain	1905
C12	Bias frequency (speed) for terminal No.1	917
C13	Bias (speed) for terminal No.1	1917
C14	Gain frequency (speed) for terminal No.1	918
C15	Gain (speed) for terminal No.1	1918
C16	Terminal 1 bias command (torque/magnetic flux)	919
C17	Terminal 1 bias (torque/magnetic flux)	1919
C18	Terminal 1 gain command (torque/magnetic flux)	920
C19	Terminal 1 gain (torque/magnetic flux)	1920
C8	Current output bias signal	930
C9	Current output bias current	1930
C10	Current output gain signal	931
C11	Current output gain current	1931
C38	Terminal 4 bias command (torque/magnetic flux)	932
C39	Terminal 4 bias (torque/magnetic flux)	1932
C40	Terminal 4 gain command (torque/magnetic flux)	933
C41	Terminal 4 gain (torque/magnetic flux)	1933
C42	PID display bias coefficient	934
C43	PID display bias analog value	1934
C44	PID display gain coefficient	935
C45	PID display gain analog value	1935

FREQROL-F700P series

Second parameter specification codes for parameter numbers Pr.125, Pr.126 and C2 to C7

Parameter No.	Name	Second parameter specification code
C2	Terminal 2 frequency setting bias frequency	902
C3	Terminal 2 frequency setting bias	1902
125	Terminal 2 frequency setting gain frequency	903
C4	Terminal 2 frequency setting gain	1903
C5	Terminal 4 frequency setting bias frequency	904
C6	Terminal 4 frequency setting bias	1904
126	Terminal 4 frequency setting gain frequency	905
C7	Terminal 4 frequency setting gain	1905
C42	PID display bias coefficient	934
C43	PID display bias analog value	1934
C44	PID display gain coefficient	935
C45	PID display gain analog value	1935

FREQROL-A700 series

Second parameter specification codes for parameter numbers Pr.125, Pr.126, C2 to C7, C12 to C19 and C38 to C41

Parameter No.	Name	Second parameter specification code
C2	Terminal 2 frequency setting bias frequency	902
C3	Terminal 2 frequency setting bias	1902
125	Terminal 2 frequency setting gain frequency	903
C4	Terminal 2 frequency setting gain	1903
C5	Terminal 4 frequency setting bias frequency	904
C6	Terminal 4 frequency setting bias	1904
126	Terminal 4 frequency setting gain frequency	905
C7	Terminal 4 frequency setting gain	1905
C12	Bias frequency (speed) for terminal No.1	917
C13	Bias (speed) for terminal No.1	1917
C14	Gain frequency (speed) for terminal No.1	918
C15	Gain (speed) for terminal No.1	1918
C16	Terminal 1 bias command (torque/ magnetic flux)	919
C17	Terminal 1 bias (torque/magnetic flux)	1919
C18	Terminal 1 gain command (torque/ magnetic flux)	920
C19	Terminal 1 gain (torque/magnetic flux)	1920
C30	Bias frequency (speed) for terminal No.6	926
C31	Bias (speed) for terminal No.6	1926
C32	Gain frequency (speed) for terminal No.6	927
C33	Gain (speed) for terminal No.6	1927
C34	Bias command (torque) for terminal No.6	928
C35	Bias (torque) for terminal No.6	1928
C36	Gain command (torque) for terminal No.6	929
C37	Gain (torque) for terminal No.6	1929
C38	Terminal 4 bias command (torque/ magnetic flux)	932
C39	Terminal 4 bias (torque/magnetic flux)	1932
C40	Terminal 4 gain command (torque/ magnetic flux)	933
C41	Terminal 4 gain (torque/magnetic flux)	1933

FREQROL-F700PJ/E700/E700EX/D700 series

Second parameter specification codes for parameter numbers Pr.125, Pr.126, C2 to C7 and C22 to C25

Parameter No.	Name	Second parameter specification code
C2	Terminal 2 frequency setting bias frequency	902
C3	Terminal 2 frequency setting bias	1902
125	Terminal 2 frequency setting gain frequency	903
C4	Terminal 2 frequency setting gain	1903
C5	Terminal 4 frequency setting bias frequency	904
C6	Terminal 4 frequency setting bias	1904
126	Terminal 4 frequency setting gain frequency	905
C7	Terminal 4 frequency setting gain	1905
C22	Frequency setting voltage bias frequency (Built-in potentiometer)	922
C23	Frequency setting voltage bias (Built-in potentiometer)	1922
C24	Frequency setting voltage gain frequency (Built-in potentiometer)	923
C25	Frequency setting voltage gain (Built-in potentiometer)	1923

FREQROL-V500 series

Second parameter specification codes for parameter numbers Pr.902 to 905 and 917 to 920

Parameter No.	Name	Second parameter specification code		
		Offset/Gain (H00)	Analog (H01)	Analog value of terminal (H02)
902	Speed setting No. 2 bias	902	1902	2902
903	Speed setting No. 2 gain	903	1903	2903
904	Torque command No. 3 bias	904	1904	2904
905	Torque command No. 3 gain	905	1905	2905
917	No.1 terminal bias (speed)	917	1917	2917
918	No.1 terminal gain (speed)	918	1918	2918
919	No.1 terminal bias (torque/magnetic flux)	919	1919	2919
920	No.1 terminal gain (torque/magnetic flux)	920	1920	2920

Point

Specify inverter parameter number in (s2) ((s3) in IVBWR instruction) as show below.

- In the case of parameter numbers 0 to 999, write the parameter number as is.
- In the case of parameter numbers 1000 to 9999, write parameter number + 10000. (Example: When parameter number is 1234, specify K11234.)

19.9 Related Devices

This section describes the special relay/special register functions used in the inverter communication function.

Point

Available communication channels vary depending on the CPU module and system configuration.
For communication channels, refer to ☞ Page 330 System Configuration.
"FX3 Series compatible" devices operate only on the communication channel specified in the compatible SM/SD for communication settings.
For compatible SM/SD, refer to ☞ Page 364 PLC Communication Settings.

List of related devices

Special relay

■FX5 only

R: Read only

Device No.				Name	Description	R/W
CH1	CH2	CH3	CH4			
SM8029				Instruction execution completed	Turns ON when execution of instruction is completed, and remains ON for 1 scan.	R
SM8500	SM8510	SM8520	SM8530	Serial communication error	Turns ON when an error occurs in serial communication.	R
SM8920	SM8930	SM8940	SM8950	Inverter communicating	Remains ON while inverter communication is being executed.	R
SM8921	SM8931	SM8941	SM8951	IVBWR instruction error	Turns ON when an IVBWR instruction error occurs.	R

■FX3 Series compatible

R: Read only

Device No.		Name	Description	R/W
CH1	CH2			
SM8063	SM8438	Serial communication error	Turns ON when an error occurs in serial communication.	R
SM8151	SM8156	Inverter communicating	Remains ON while inverter communication is being executed.	R
SM8152	SM8157	Inverter communication error	Turns ON when an error occurs in inverter communication.	R
SM8153	SM8158	Inverter communication error latch	Turns ON when an error occurs in inverter communication.	R
SM8154	SM8159	IVBWR instruction error	Turns ON when an IVBWR instruction error occurs.	R

Special registers

■FX5 only

R: Read only

Device No.				Name	Description	R/W
CH1	CH2	CH3	CH4			
SD8500	SD8510	SD8520	SD8530	Serial communication error code	Stores the error code when a serial communication error occurs.	R
SD8502	SD8512	SD8522	SD8532	Serial communication settings	Stores the setting of the communication parameter.	R
SD8503	SD8513	SD8523	SD8533	Serial communication operation mode	Stores the communication type being used.	R
SD8921	SD8931	SD8941	SD8951	IVBWR instruction error parameter number	Stores a parameter number in which an IVBWR instruction error has occurred.	R
SD8981	SD8991	SD9001	SD9011	Response wait time	Stores the communication response wait time.	R

■FX3 Series compatible

R: Read only

Device No.		Name	Description	R/W
CH1	CH2			
SD8063	SD8438	Serial communication error code	Stores the error code when a serial communication error occurs.	R
SD8152	SD8157	Inverter communication error code	Stores the error code when an inverter communication error occurs.	R
SD8154	SD8159	IVBWR instruction error parameter number	Stores a parameter number in which an IVBWR instruction error has occurred.	R
SD8419	SD8439	Serial communication operation mode	Stores status of the current communication being executed.	R

Details of related devices

Instruction execution completed

Turns ON when the execution of inverter communication instruction is completed, and remains ON for 1 scan.

When an error occurs in an inverter communication instruction, remains ON for 1 scan in the same way.

R: Read only

FX5 dedicated	Description	R/W
SM8029	When execution of instruction is completed, turns ON for 1 scan.	R

Precautions

Do not turn ON/OFF by program or engineering tool.

Instruction execution completed is also used as the execution completed flag for other instructions (such as positioning instructions).

When using instruction execution completed, provide the contact just under the instruction whose execution completion is to be checked.

Serial communication error

Turns ON when an error occurs in serial communication. These flags are used to check the serial communication error. Turns ON when a parity error, overrun error or framing error occurs during communication with inverters or when an inverter communication error occurs.

R: Read only

FX5 dedicated				FX3 Series compatible		Description	R/W
CH1	CH2	CH3	CH4	CH1	CH2		
SM8500	SM8510	SM8520	SM8530	SM8063	SM8438	Turns ON when an error occurs in serial communication. Devices do not turn OFF even when normal communication is restored.	R

After a device above turns ON, the error code is stored in the corresponding device below.

FX5 dedicated				FX3 Series compatible		Name	Description
CH1	CH2	CH3	CH4	CH1	CH2		
SD8500	SD8510	SD8520	SD8530	SD8063	SD8438	Serial communication error code	When a serial communication error occurs, the error code is stored.

Precautions

Do not turn ON or OFF with program or engineering tool.

Serial communication error do not turn OFF even if normal communication is restored. The devices turn OFF when power is turned OFF→ON, STOP→RUN, reset or SM50 (Error Detection Reset Completion) is turned ON.

Inverter communicating

These devices remain ON while communicating with inverter by an inverter communication instruction.

R: Read only

FX5 dedicated				FX3 Series compatible		Description	R/W
CH1	CH2	CH3	CH4	CH1	CH2		
SM8920	SM8930	SM8940	SM8950	SM8151	SM8156	Remains ON while communication with an inverter is executed by an inverter communication instruction.	R

Precautions

When [Inverter communicating] is ON, instructions other than the instruction under execution cannot be executed.

Do not turn ON/OFF by program or engineering tool.

Inverter communication error

If FX3 Series compatible SM/SD area settings have been set, the devices turn ON if an error occurs for an inverter communication instruction.

R: Read only

FX3 Series compatible				Description	R/W
CH1		CH2			
SM8152	SM8153[*1]	SM8157	SM8158[*1]	Turns ON when an inverter communication error occurs.	R

*1 Supports latch function.

After the devices above turns ON, the error code is stored in the compatible devices below.

FX3 Series compatible		Name	Description
CH1	CH2		
SD8152	SD8157	Inverter communication error code latch	These devices store an error code when a communication error is caused by an inverter communication instruction.

Storing an error code is only for the first error occurrence, and it is not updated for the second error occurrence onwards.

Precautions

Do not turn ON/OFF by program or engineering tool.

Inverter communication errors are not cleared even after communication is restored to normal state. Devices are cleared when power is turned OFF→ON, STOP→RUN, reset or SM50 (Error Detection Reset Completion) is turned ON.

IVBWR instruction error

The devices turn ON when a parameter number or set value specified in IVBWR instruction is outside the allowable range.

R: Read only

FX5 dedicated				FX3 Series compatible		Description	R/W
CH1	CH2	CH3	CH4	CH1	CH2		
SM8921	SM8931	SM8941	SM8951	SM8154	SM8159	These devices turn ON when an error occurs in IVBWR instruction.	R

When an IVBWR instruction error is ON, the parameter number that could not be set is stored in the following corresponding devices.

FX5 dedicated				FX3 Series compatible		Name	Description
CH1	CH2	CH3	CH4	CH1	CH2		
SD8921	SD8931	SD8941	SD8951	SD8154	SD8159	IVBWR instruction error parameter number	Stores the parameter number which was not written by IVBWR instruction.

Precautions

Do not turn ON/OFF by program or engineering tool.

IVBWR communication errors are not cleared even after communication is restored to normal state. Devices are cleared when power is turned OFF→ON, STOP→RUN, reset or SM50 (Error Detection Reset Completion) is turned ON.

Serial communication error code

Stores the error code (☞ Page 842 List of error codes for inverter communication) when the inverter communication instruction error occurs.

R: Read only

FX5 dedicated				FX3 Series compatible		Description	R/W
CH1	CH2	CH3	CH4	CH1	CH2		
SD8500	SD8510	SD8520	SD8530	SD8063	SD8438	When a serial communication error occurs, the error code is stored.	R

Precautions

Do not change the value with program or engineering tool.

Serial communication error codes are not cleared even after communication is restored to normal state. Devices are cleared when power is turned OFF→ON, STOP→RUN, reset or SM50 (Error Detection Reset Completion) is turned ON.

Serial communication settings

Stores the communication parameters set in the communication settings when PLC power is turned OFF→ON, STOP→RUN, PAUSE→RUN or reset. (☞ Page 364 PLC Communication Settings)

R: Read only

FX5 dedicated				Description	R/W
CH1	CH2	CH3	CH4		
SD8502	SD8512	SD8522	SD8532	Stores the setting of the communication parameter.	R

The descriptions of the communication parameters are as follows.

Bit No.	Name	Description	
		0 (bit is OFF)	1 (bit is ON)
b0	Data length	7 bits	8 bits
b1 b2	Parity bit	b2, b1 (0, 0): N/A (0, 1): Odd (1, 1): Even	
b3	Stop bit	1 bit	2 bits
b4 b5 b6 b7	Baud rate	b7, b6, b5, b4 (0, 1, 1, 1): 4800bps (1, 0, 0, 0): 9600bps (1, 0, 0, 1): 19200bps (1, 0, 1, 0): 38400bps (1, 0, 1, 1): 57600bps (1, 1, 0, 1): 115200bps	

Precautions

Do not change the value with program or engineering tool.

Serial communication operation mode

Stores the communication function code of the serial communication under execution.

R: Read only

FX5 dedicated				FX3 Series compatible		Description	R/W
CH1	CH2	CH3	CH4	CH1	CH2		
SD8503	SD8513	SD8523	SD8533	SD8419	SD8439	0: MELSOFT connection or MC protocol 3: N:N Network Communication 5: Non-Protocol Communication 6: Parallel Link Communication 7: Inverter Communication 9: MODBUS RTU Communication 12: Predefined protocol support Other than above: Not used	R

Precautions

Do not change the value with program or engineering tool.

These devices store "7" while an inverter communication instruction is being driven or not, as long as the communication mode has not changed.

Error code for inverter communication

If FX3 Series compatible SM/SD area settings have been set, error code (☞ Page 842 List of error codes for inverter communication) is stored if communication error occurs by inverter communication instruction.

R: Read only

FX3 Series compatible		Description	R/W
CH1	CH2		
SD8152	SD8157	These devices store an error code when a communication error is caused by an inverter communication instruction.	R

Error code of first error that occurred is maintained if errors occur for multiple instructions.

Precautions

Do not change the value with program or engineering tool.

Inverter communication error code is not cleared even if communication is restored to normal state. Devices are cleared when power is turned OFF→ON, STOP→RUN, reset or SM50 (Error Detection Reset Completion) is turned ON.

IVBWR instruction error parameter number

These devices store the parameter number in which an error has occurred when IVBWR instruction error flag turns ON.

R: Read only

FX5 dedicated				FX3 Series compatible		Description	R/W
CH1	CH2	CH3	CH4	CH1	CH2		
SD8921	SD8931	SD8941	SD8951	SD8154	SD8159	Stores the parameter number which could not be written by IVBWR instruction.	R

Error code of first error that occurred is maintained if multiple errors occur for IVBWR instruction.

Precautions

Do not change the value with program or engineering tool.

Response wait time

Stores the communication response wait time set in the communication settings. (☞ Page 364 PLC Communication Settings)

R: Read only

FX5 dedicated				Description	R/W
CH1	CH2	CH3	CH4		
SD8981	SD8991	SD9001	SD9011	Stores the communication response wait time of inverter communication. 1 to 32767 (ms)	R

Precautions

If the set value is outside the range or less than 100 ms, response waiting time operates as 100 ms.

Do not change the value with program or engineering tool.

20 NON-PROTOCOL COMMUNICATION

This chapter describes non-protocol communication.

20.1 Function Summary

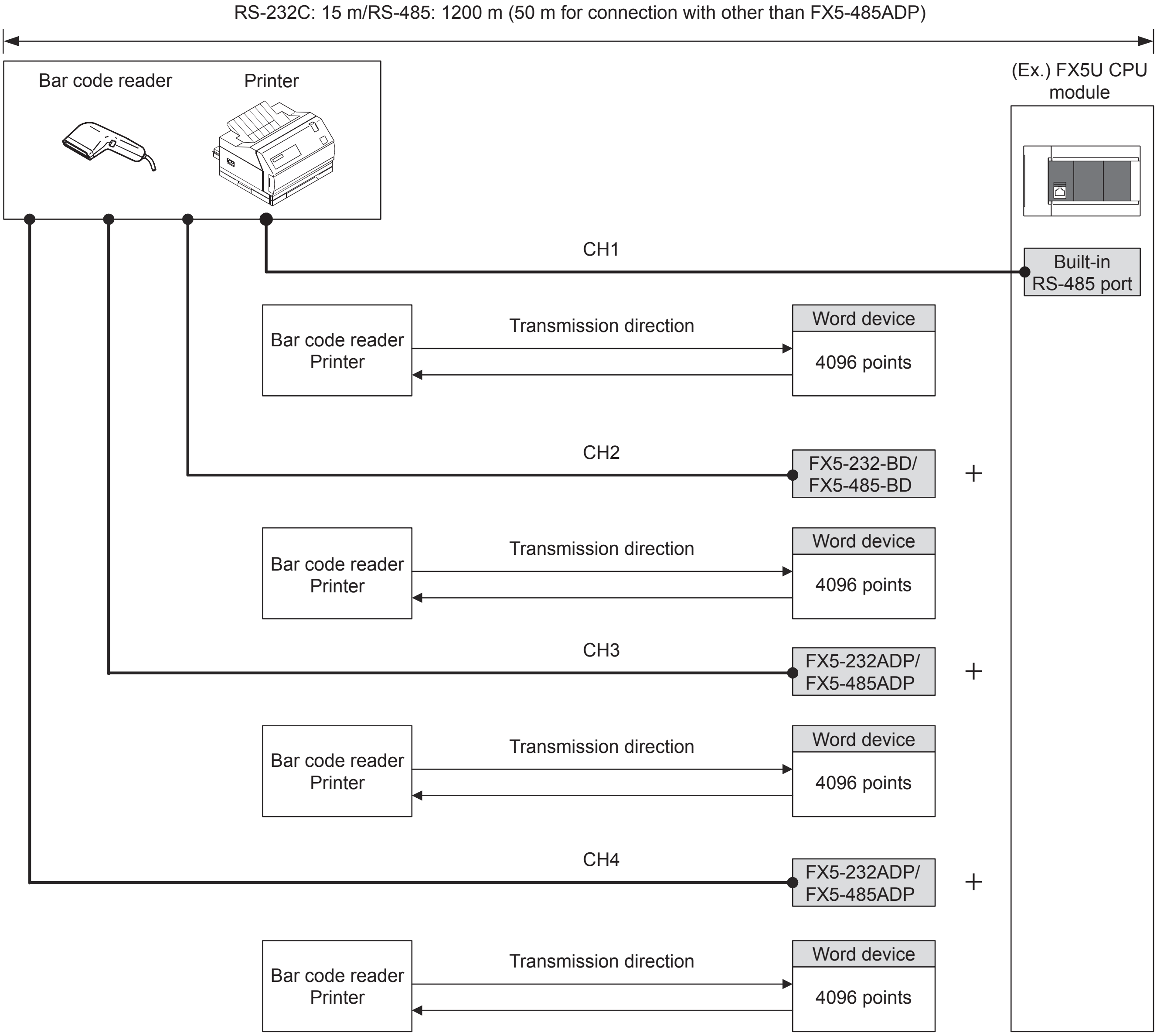
Non-protocol communication exchanges data between a printer, bar code reader, etc. with no protocol.

Non-protocol communication is available using the RS2 instruction.

The RS2 instruction can simultaneously communicate on maximum 4 channels by specifying the channels.[1]

- Up to 4096 points of data can be sent, and up to 4096 points of data can be received.
- Data transfer is enabled when the connected equipment supports non-protocol serial communication.
- The overall distance is 1200 m maximum. (Only when configured by FX5-485ADP)

System



---

[1] The maximum number of channels varies depending on the CPU module. (☞ Page 412 System Configuration)

20.2 Procedure Before Operation

The flow chart below shows the Non-Protocol Communication setting procedure up until data link:

1. Check communication specifications

For communication specifications, and communication applicability, refer to ☞ Page 415 Specifications.

2. System configuration and selection.

For system configuration of each PLCs, refer to ☞ Page 412 System Configuration.

3. Wiring

For selection of cables and connection equipment, and wiring example, refer to ☞ Page 415 Wiring.

4. Communication settings[*1]

For communication settings of communication device, refer to ☞ Page 420 Communication Settings.

5. Program creation

For detailed explanation of related devices, operation of control line, and program, refer to ☞ Page 423 Programming.

*1 For details on operating procedures of GX Works3, refer to the manual below.
📖 GX Works3 Operating Manual

20.3 System Configuration

This section outlines the system configuration required to use Non-Protocol communications.

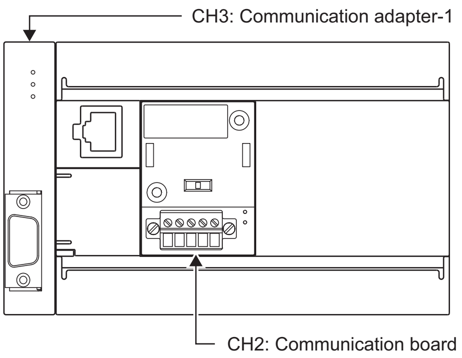
FX5S CPU module

A serial port of up to 2 channels can be connected in the FX5S CPU module by using a communication board and communication adapter.

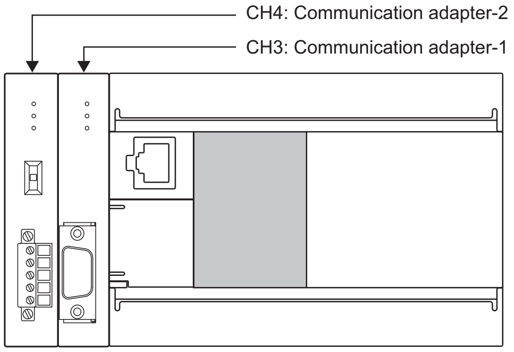
Communication channel assignments are fixed regardless of the system configuration.

The combinations available for the system configurations are shown below.

[Configuration example 1]



[Configuration example 2]



Item		Serial port	Important points in selection	Overall distance
Communication board	FX5-485-BD	CH2	Because the board can be mounted on top of the CPU module, there is no change in the installation space to be required.	50 m or less
	FX5-232-BD			15 m or less
Communication adapter	FX5-485ADP	CH3, CH4[*1]	Mount the communication adapter to the left of the CPU module.	1200 m or less
	FX5-232ADP			15 m or less

*1 The adapters are assigned to CH3 and CH4 in the order, from the closest one to the CPU module.

Precautions

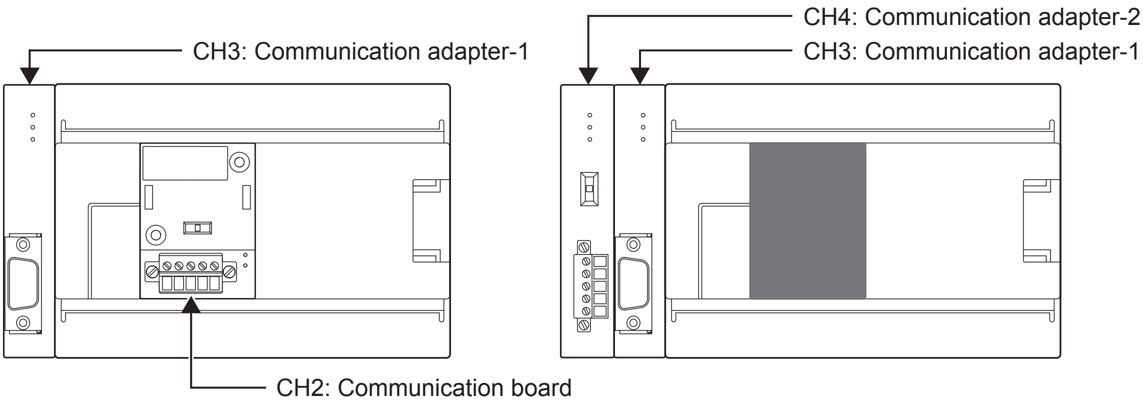
Total number of communication boards and communication adapters that can be connected is two.

FX5UJ CPU module

A serial port of up to 2 channels can be connected in the FX5UJ CPU module by using a communication board and communication adapter.

Communication channel assignments are fixed regardless of the system configuration.

The combinations which can be configured are shown below.


[Configuration example 1]

[Configuration example 2]

Item		Serial port	Important points in selection	Overall distance
Communication board	FX5-485-BD	CH2	Mounted on top of the CPU module, there is no change in the installation space requirements.	50 m or less
	FX5-232-BD			15 m or less
Communication adapter	FX5-485ADP	CH3, CH4[1]	Mount the communication adapter to the left of the CPU module.	1200 m or less
	FX5-232ADP			15 m or less

[1] The adapters are assigned to CH3 and CH4 in the order, from the closest one to the CPU module.

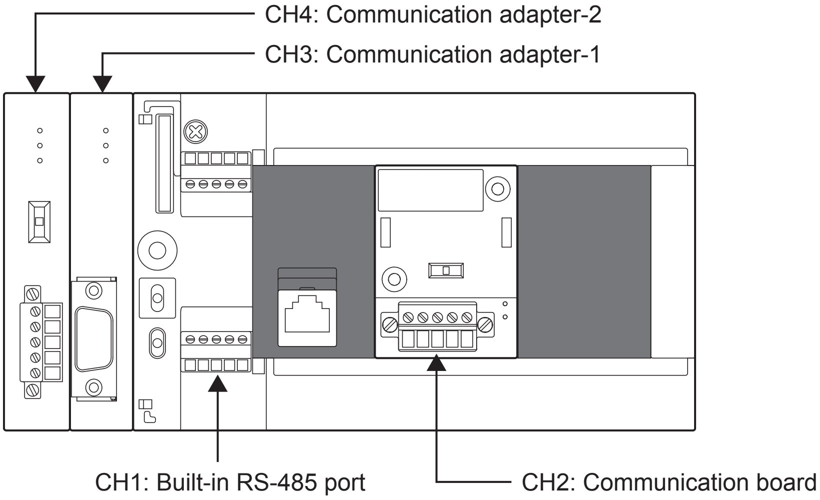
Precautions

Total number of communication boards and communication adapters that can be connected is two.

FX5U CPU module

A serial port of up to 4 channels can be connected in the FX5U CPU module by using the built-in RS-485 port, communication board, and communication adapter.

Communication channel assignments are fixed regardless of the system configuration.



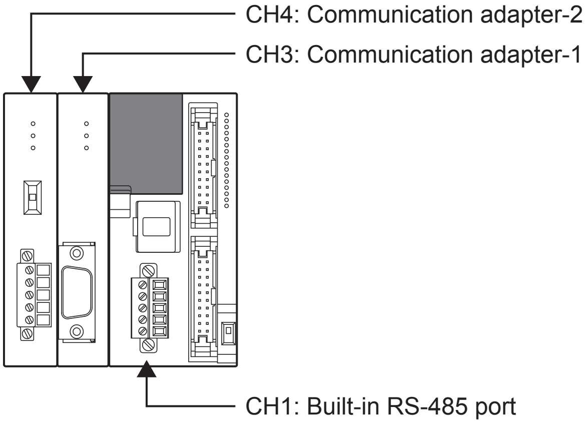
Item		Serial port	Important points for selection	Overall distance
Built-in RS-485 port		CH1	Since it is built-into the CPU module, there is no need to add equipment.	50 m or less
Communication board	FX5-485-BD	CH2	Mounted on top of the CPU module, there is no change in the installation space requirements.	50 m or less
	FX5-232-BD			15 m or less
Communication adapter	FX5-485ADP	CH3, CH4[*1]	Mount the communication adapter to the left of the CPU module.	1200 m or less
	FX5-232ADP			15 m or less

*1 The adapters are assigned to CH3 and CH4 in the order, from the closest one to the CPU module.

FX5UC CPU module

A serial port of up to 3 channels can be connected in the FX5UC CPU module by using the built-in RS-485 port and communication adapter.

Communication channel assignments are fixed regardless of the system configuration.



Item		Serial port	Important points for selection	Overall distance
Built-in RS-485 port		CH1	Since it is built-into the CPU module, there is no need to add equipment.	50 m or less
Communication adapter	FX5-485ADP	CH3, CH4[*1]	Mount the communication adapter to the left of the CPU module.	1200 m or less
	FX5-232ADP			15 m or less

*1 The adapters are assigned to CH3 and CH4 in the order, from the closest one to the CPU module.

20.4 Specifications

This section describes the communication specifications and performance of non-protocol communication.

Items		Specifications	
Transmission standard		RS-485 and RS-422 standard	RS-232C standard
Maximum overall distance		When using FX5-485ADP: 1200m or less When using built-in RS-485 port or FX5-485-BD: 50m or less	15m or less
Number of Transfer Data		0 to 4096	
Protocol type		—	
Control procedure		Format 1 (CR and LF are not added)/Format 4 (CR and LF are added)	
Communication method		Half-duplex, bi-directional communication/full-duplex, bi-directional communication	
Baud rate		300/600/1200/2400/4800/9600/19200/38400/57600/115200(bps)	
Character format	Start bit	1 bit	
	Data length	7 bits/8 bits	
	Parity bit	None, odd or even	
	Stop bit	1 bit/2 bits	
Header		Provided or not provided	
Terminator		Provided or not provided	
Control line		—	Provided or not provided
Sum check		Provided or not provided	

20.5 Wiring

This section explains about the wiring.

Wiring procedure

1. Preparing for wiring

Prepare cables and termination resistors required for wiring. (☞ Page 415 Cable)

2. Turn OFF the PLC power

Before wiring, make sure that the PLC power is OFF.

3. Wire the communication equipment

Connect the RS-485, RS-232C communication equipment. (☞ Page 418 Connection diagram)

Cable

Select cables using the procedure described below.

Twisted pair cable

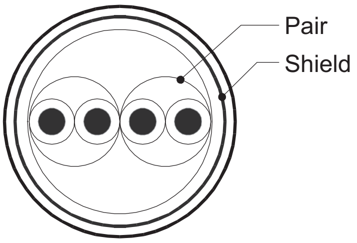
Use shielded twisted pair cables for connecting RS-485 communication equipment.

The specifications of the cables used in wiring are shown.

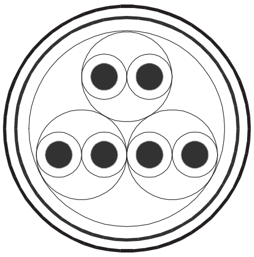
■RS-485 cable specifications

Items	Description
Cable type	Shielded cable
Number of pairs	2p, 3p
Conductor resistance (20℃)	88.0 Ω/km or less
Insulation resistance	10000 MΩ-km or more
Voltage resistance	500VDC, 1 minute
Electrostatic capacitance (1kHz)	60 nF/km or less as an average
Characteristic impedance (100kHz)	110±10 Ω

■Cable structural drawing (reference)



Example of two-pair cable structural drawing



Example of three-pair cable structural drawing

Connecting cables

The table below shows applicable cables and tightening torques.

Item	Number of wires connected per terminal	Wire size		Tightening torque
		Solid wire, Stranded wire	Wire ferrule with insulation sleeve	
FX5U CPU module built-in RS-485 port	One wire	0.2 to 0.5mm^2 (24 to 20 AWG)	0.2 to 0.5mm^2 (24 to 20 AWG)	0.22 to 0.25N·m
	Two wires	0.2mm^2 (24 AWG)	—	
FX5UC CPU module built-in RS-485 port FX5-485-BD FX5-485ADP	One wire	0.3 to 0.5mm^2 (22 to 20 AWG)	0.3 to 0.5mm^2 (22 to 20 AWG)	
	Two wires	0.3mm^2 (22 AWG)	—	

Precautions

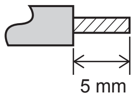
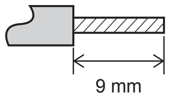
Do not tighten terminal screws with torque beyond the specified range. Otherwise it may cause equipment failure or malfunction.

■Wire end treatment

With regard to the cable end treatment, use a stranded cable or solid cable as is, or use a wire ferrule with insulating sleeve.

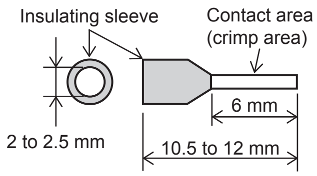
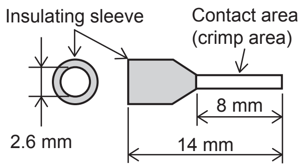
- When using a stranded cable or solid cable as is

- Twist the end of stranded wire and make sure that there are no loose wires.
- Please do not solder plate the ends of the cable.

Dimensions of the cable end	
FX5U CPU module built-in RS-485 port	**FX5UC CPU module built-in RS-485 port, FX5-485-BD, FX5-485ADP**
	

- When using a wire ferrule with insulation sleeve

Because it is difficult to insert a cable into an insulating sleeve depending on the thickness of the cable sheath, select the proper cable according to the outline drawing.

FX5U CPU module built-in RS-485 port	FX5UC CPU module built-in RS-485 port, FX5-485-BD, FX5-485ADP
	

<Reference>

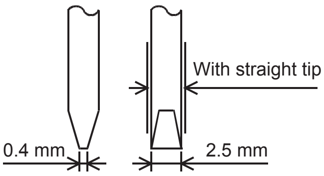
Item	Manufacturer	Model name	Crimping tool
FX5U CPU module built-in RS-485 port	PHOENIX CONTACT GmbH & Co. KG	AI 0.5-6 WH	CRIMPFOX 6 CRIMPFOX 6T-F
FX5UC CPU module built-in RS-485 port FX5-485-BD FX5-485ADP		AI 0.5-8 WH	

- Tool

For tightening the terminal, use a commercially available small screwdriver with straight tip that is not widened toward the end as shown below.

■Precautions

If the diameter of the screwdriver tip is too small, the required tightening torque cannot be achieved. To achieve the appropriate tightening torque shown in the previous page, use the following screwdriver or its equivalent (grip diameter: approximately 25mm).



<Reference>

Manufacturer	Model name
PHOENIX CONTACT GmbH & Co. KG	SZS 0.4×2.5

Termination resistor setting

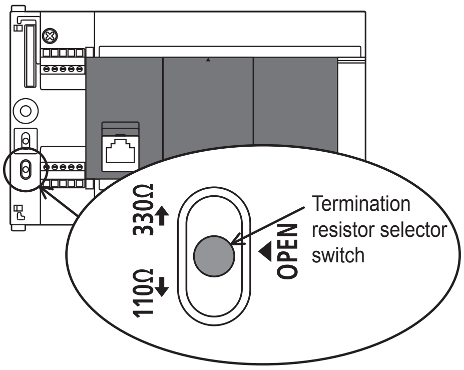
Make sure to provide a termination resistor at both ends of the wire.

The built-in RS-485 port, FX5-485-BD and FX5-485ADP have a built-in termination resistor.

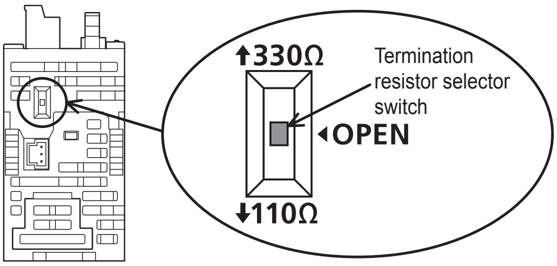
Set the termination resistor selector switch as below.

Wiring	Termination resistor selector switch
Two-pair wiring	330Ω
One-pair wiring	110Ω

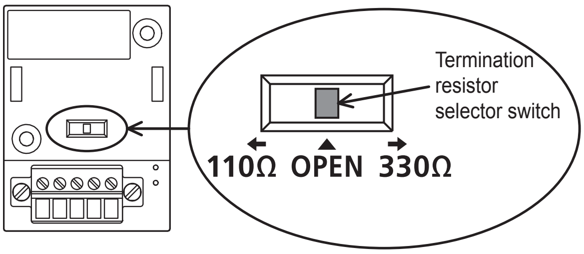
■FX5U CPU module built-in RS-485 port



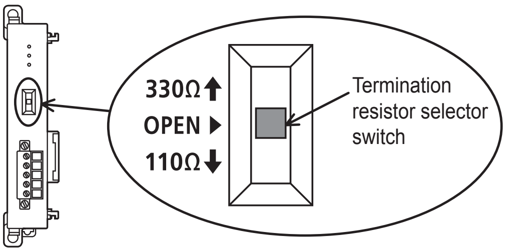
■FX5UC CPU module built-in RS-485 port



■FX5-485-BD



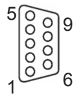
■FX5-485ADP



Connection diagram

RS-232C communication equipment

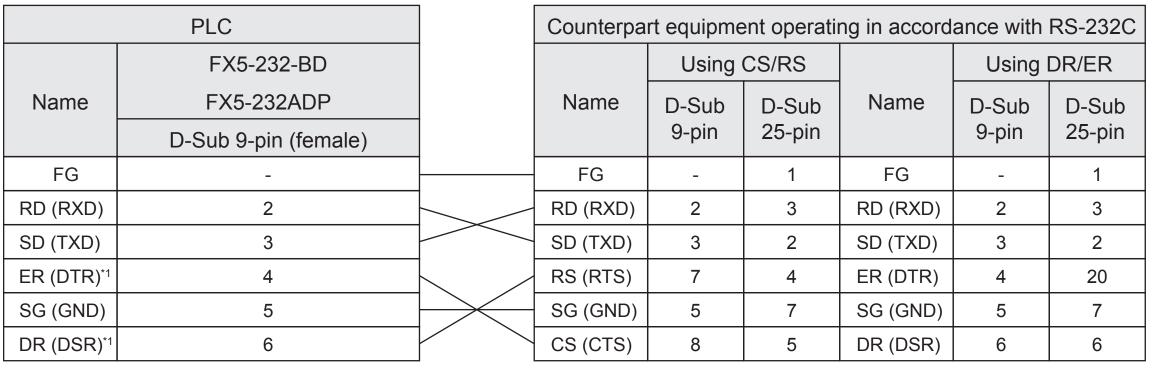
■Connector pin arrangement

FX5-232-BD, FX5-232ADP		Signal name	Function
D-SUB 9-pin (male)	**Pin No.**		
	1	CD (DCD)	Receive carrier detection
	2	RD (RXD)	Receive data
	3	SD (TXD)	Send data
	4	ER (DTR)	Send request
	5	SG (GND)	Signal ground
	6	DR (DSR)	Send enabled
	7, 8, 9	Not used	
	—	FG	Frame ground

■Wiring

Representative wiring examples are shown in this section. When pin numbers in the counterpart equipment are different, wire the pins as shown below.

- When connected equipment is Data Terminal Equipment

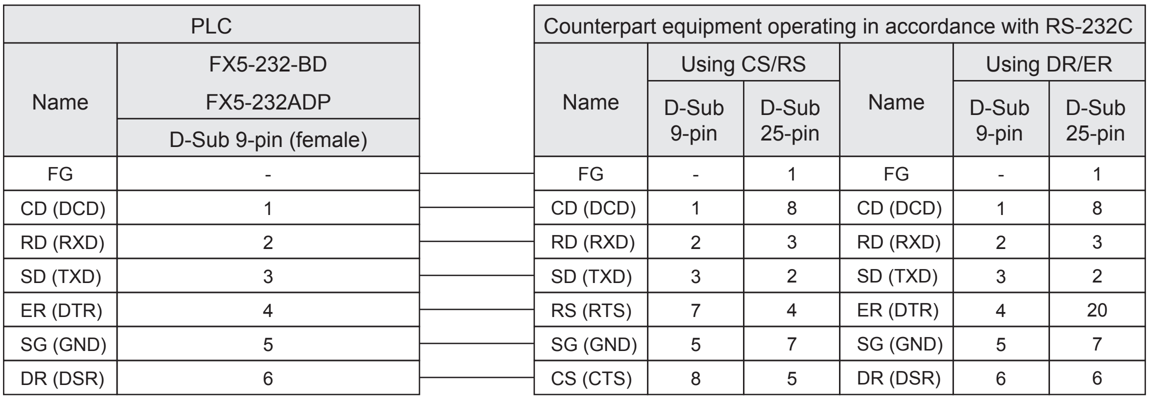


PLC	
Name	FX5-232-BD FX5-232ADP
	D-Sub 9-pin (female)
FG	-
RD (RXD)	2
SD (TXD)	3
ER (DTR)*1	4
SG (GND)	5
DR (DSR)*1	6

Counterpart equipment operating in accordance with RS-232C					
Name	Using CS/RS		Name	Using DR/ER	
	D-Sub 9-pin	D-Sub 25-pin		D-Sub 9-pin	D-Sub 25-pin
FG	-	1	FG	-	1
RD (RXD)	2	3	RD (RXD)	2	3
SD (TXD)	3	2	SD (TXD)	3	2
RS (RTS)	7	4	ER (DTR)	4	20
SG (GND)	5	7	SG (GND)	5	7
CS (CTS)	8	5	DR (DSR)	6	6

*1 When the control line is not used, wiring is not required for this signal.
For the interlink mode, the control line will be used, so wiring for this signal is required.

- When connected equipment is Data Communication Equipment



PLC	
Name	FX5-232-BD FX5-232ADP
	D-Sub 9-pin (female)
FG	-
CD (DCD)	1
RD (RXD)	2
SD (TXD)	3
ER (DTR)	4
SG (GND)	5
DR (DSR)	6

Counterpart equipment operating in accordance with RS-232C					
Name	Using CS/RS		Name	Using DR/ER	
	D-Sub 9-pin	D-Sub 25-pin		D-Sub 9-pin	D-Sub 25-pin
FG	-	1	FG	-	1
CD (DCD)	1	8	CD (DCD)	1	8
RD (RXD)	2	3	RD (RXD)	2	3
SD (TXD)	3	2	SD (TXD)	3	2
RS (RTS)	7	4	ER (DTR)	4	20
SG (GND)	5	7	SG (GND)	5	7
CS (CTS)	8	5	DR (DSR)	6	6

RS-485 communication equipment

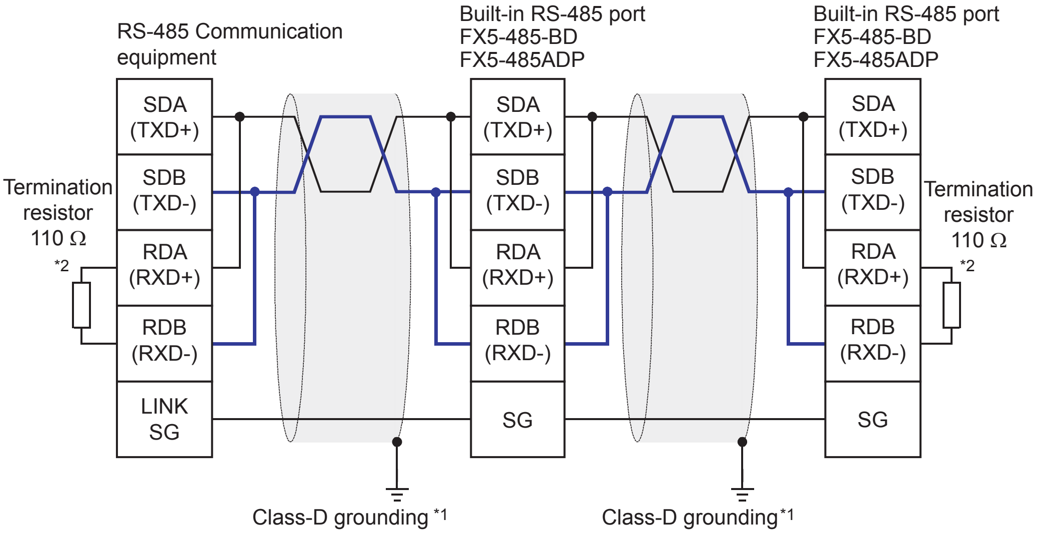
One-pair wiring and two-pair wiring (for RS-422) are applicable for RS-485 communication. Wire according to the counterpart equipment.

The wiring that can be used is limited by the control line setting in the communication settings (☞ Page 420 Communication Settings).

○: Supported, ×: Not supported

Control line	One-pair wiring	Two-pair wiring
Bidirectional half duplex	○	○
Bidirectional full duplex	×	○

■For one-pair wiring



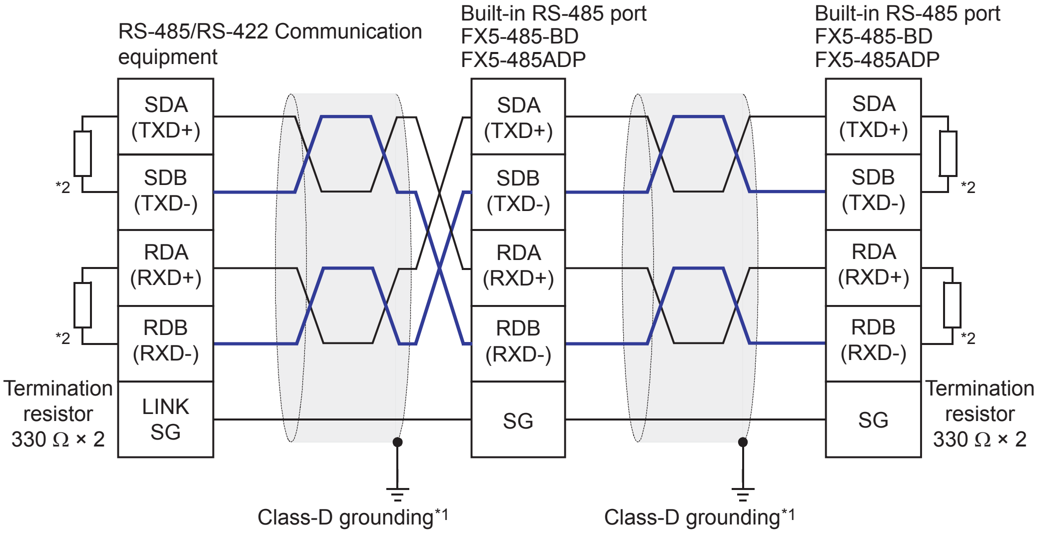
*1 Make sure to perform Class-D grounding on the shield of the twisted pair cable to be connected.

*2 Make sure to provide a termination resistor at both ends of the wire. The built-in RS-485 port, FX5-485-BD and FX5-485ADP have built-in termination resistors. Set the termination resistor selector switch to 110 Ω.

Precautions

When the control line is set to [Bidirectional Full Duplex], the built-in RS-485 port, FX5-485-BD and FX5-485ADP become a full-duplex interface, and echo will occur.

■For two-pair wiring



*1 Make sure to perform Class-D grounding on the shield of the twisted pair cable to be connected.

*2 Make sure to provide a termination resistor at both ends of the wire. The built-in RS-485 port, FX5-485-BD and FX5-485ADP have built-in termination resistors. Set the termination resistor selector switch to 330 Ω.

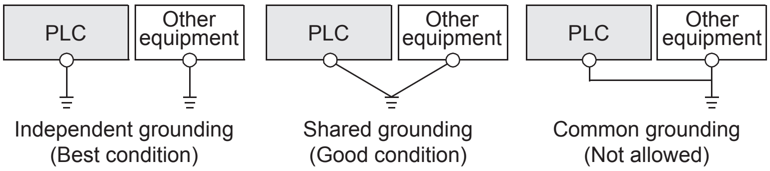
Grounding

Grounding should be performed as stated below.

- Perform Class-D grounding. (Grounding resistance: 100Ω or less)
- Independent grounding should be performed for best results.

If the PLC cannot be grounded independently, perform the "Shared grounding" shown below.

For details, refer to User's Manual (Hardware) of the CPU module used.



- The grounding wire size should be 14 AWG (2mm²) or larger.
- Bring the grounding point close to the PLC as much as possible so that the ground cable can be shortened.

20.6 Communication Settings

For the FX5 communication settings of this function, parameters are set using GX Works3. For details about GX Works3, refer to 📖GX Works3 Operating Manual.

Setting of parameter differs according to the module used. The procedure for each module is as follows.

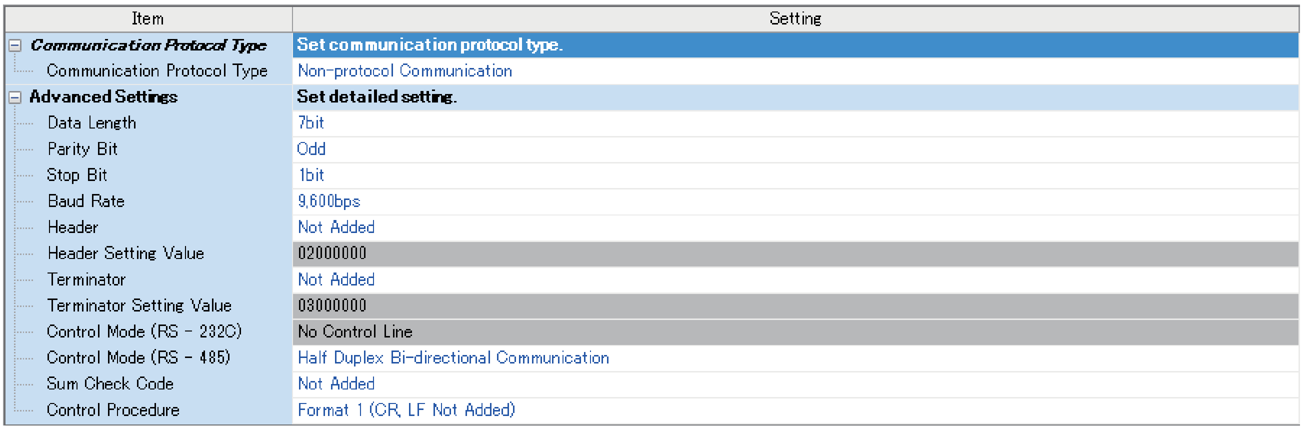
Built-in RS-485 port (CH1)

Navigation Window ⇨ Parameter ⇨ FX5UCPU ⇨ Module Parameter ⇨ 485 Serial Port

Window

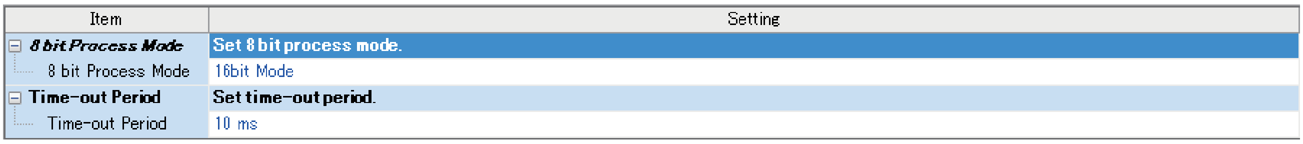
The following screen will be displayed if [Non-protocol Communication] is set for the communication protocol type.

■Basic Settings



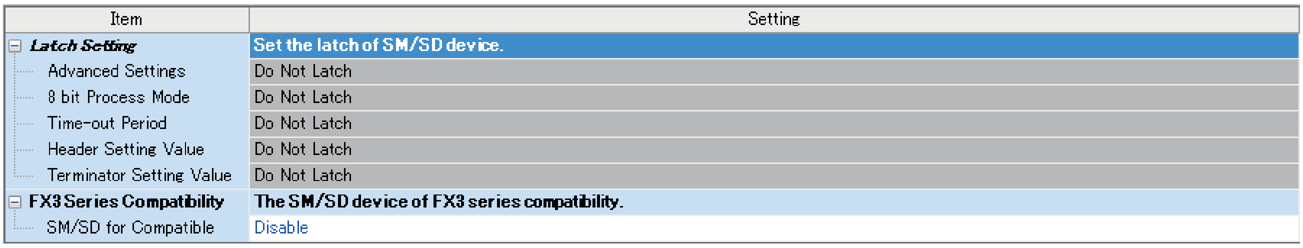
Item	Setting
⊟ **Communication Protocol Type**	**Set communication protocol type.**
Communication Protocol Type	Non-protocol Communication
⊟ **Advanced Settings**	**Set detailed setting.**
Data Length	7bit
Parity Bit	Odd
Stop Bit	1bit
Baud Rate	9,600bps
Header	Not Added
Header Setting Value	02000000
Terminator	Not Added
Terminator Setting Value	03000000
Control Mode (RS - 232C)	No Control Line
Control Mode (RS - 485)	Half Duplex Bi-directional Communication
Sum Check Code	Not Added
Control Procedure	Format 1 (CR, LF Not Added)

■Fixed Setting



Item	Setting
⊟ **8 bit Process Mode**	**Set 8 bit process mode.**
8 bit Process Mode	16bit Mode
⊟ **Time-out Period**	**Set time-out period.**
Time-out Period	10 ms

■SM/SD Setting



Item	Setting
⊟ ***Latch Setting***	**Set the latch of SM/SD device.**
Advanced Settings	Do Not Latch
8 bit Process Mode	Do Not Latch
Time-out Period	Do Not Latch
Header Setting Value	Do Not Latch
Terminator Setting Value	Do Not Latch
⊟ **FX3 Series Compatibility**	**The SM/SD device of FX3 series compatibility.**
SM/SD for Compatible	Disable

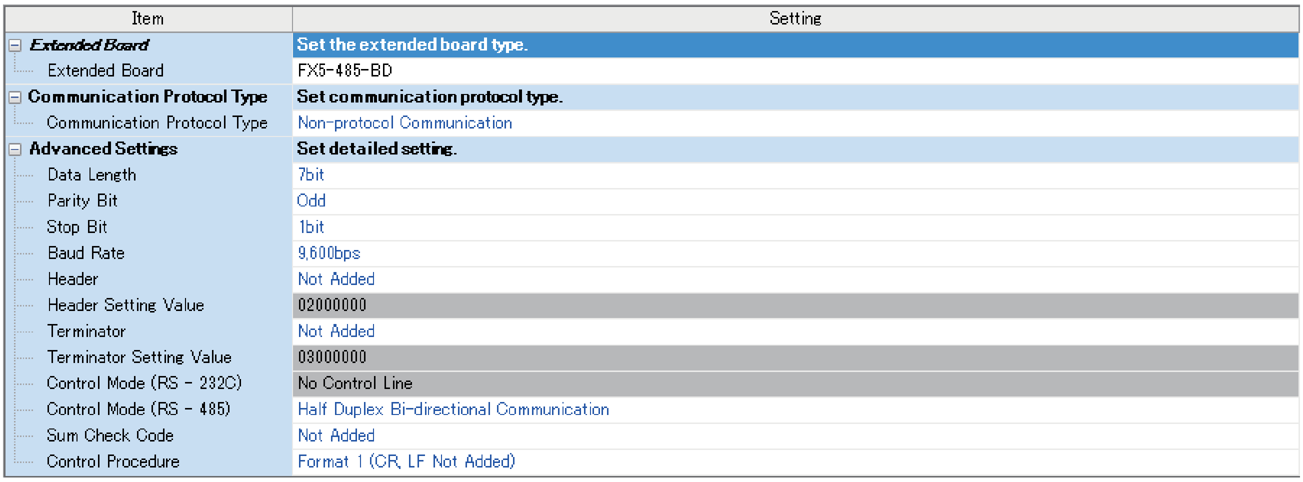
Communication board (CH2)

Navigation Window ⇨ Parameter ⇨ Model name ⇨ Module Parameter ⇨ Extended Board

Window

The following screen will be displayed if [FX5-232-BD] or [FX5-485-BD] is set for the extended board and [Non-protocol Communication] is set for the communication protocol type. Fixed setting and SM/SD setting are the same as for the built-in RS-485 port (CH1).

■Basic Settings



Item	Setting
⊟ ***Extended Board***	**Set the extended board type.**
Extended Board	FX5-485-BD
⊟ **Communication Protocol Type**	**Set communication protocol type.**
Communication Protocol Type	Non-protocol Communication
⊟ **Advanced Settings**	**Set detailed setting.**
Data Length	7bit
Parity Bit	Odd
Stop Bit	1bit
Baud Rate	9,600bps
Header	Not Added
Header Setting Value	02000000
Terminator	Not Added
Terminator Setting Value	03000000
Control Mode (RS - 232C)	No Control Line
Control Mode (RS - 485)	Half Duplex Bi-directional Communication
Sum Check Code	Not Added
Control Procedure	Format 1 (CR, LF Not Added)

Communication adapter (CH3/CH4)

When an expansion adapter is used, add expansion adapter to Module Information.

Navigation window ⇨ Parameter ⇨ Module Information ⇨ Right-click ⇨ Add New Module

After adding the expansion adapter, make settings on the screen displayed from the following operation.

Navigation window ⇨ Parameter ⇨ Module Information ⇨ ADP1 to ADP6 (Communication adapter) ⇨ Module Parameter

Window

Each setting screen is the same as for the built-in RS-485 port (CH1).

Parameter setting details

Set the following items for the serial ports that use non-protocol communication.

Items	Items		Setting value	Reference section
Basic Settings	Extended Board*1		When using this function, select [FX5-232-BD] or [FX5-485-BD].	—
	Protocol type		When using this function, select [Non-protocol Communication].	
	Advanced Settings	Data Length	7bit/8bit	
		Parity Bit	None/Odd/Even	
		Stop Bit	1bit/2bit	
		Baud Rate	300bps/600bps/1200bps/2400bps/4800bps/9600bps/19200bps/38400bps/ 57600bps/115200bps	
		Header	Added/Not Added	Page 424
		Header Setting Value*2	00000000 to FFFFFFFF You can enter a two-digit four headers.	
		Terminator	Added/Not Added	Page 425
		Terminator Setting Value*3	00000000 to FFFFFFFF You can enter a two-digit four terminators.	
		Control Mode (RS-232C)	No Control Line/ Control Line Normal Mode/ Control Line Interlink Mode/ Control Line Modem Mode	Page 436
		Control Mode (RS-485)	Full Duplex Bi-directional Communication/Half Duplex Bi-directional Communication	Page 434 Page 434
		Sum Check Code*3	Added/Not Added	Page 435
		Control Procedure	Format 1 (CR and LF are not added)/Format 4 (CR and LF are added)	Page 425
Fixed Setting	8 bit Process Mode		8 bit Mode/16bit Mode	Page 427
	Time-out Period		1 to 32767 (ms) If the set value is outside the range, it operates as 10ms.	Page 432
SM/SD Setting	FX3 Series Compatibility	SM/SD for Compatible	Disable/CH1/CH2	Page 422

*1 Only for the communication board (CH2).
*2 Valid when the header setting is [Added].
*3 Valid when the terminator setting is [Added].

FX3 Series-compatible SM/SD

When using the FX3 Series compatible SM/SD storage area, set to use special devices for either the FX3 Series CH1 or CH2. FX3 Series compatible devices corresponding to the specified channel can be used.

For details, refer to ☞ Page 442 Related Devices.

20.7 Programming

This section explains how to create programs for non-protocol communication using RS2 instruction and how such programs operate.

For details on related devices, refer to ☞ Page 442 Related Devices.

For communication settings, refer to ☞ Page 420 Communication Settings.

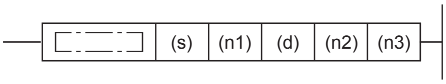
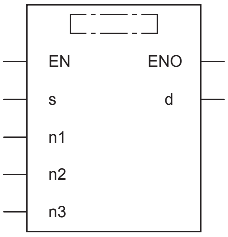
Serial data transmission

This section explains the function, operation and programming method of the RS2 instruction.

For details on the expression and executable format of the RS2 instruction, refer to the 📖MELSEC iQ-F FX5 Programming Manual (Instructions, Standard Functions/Function Blocks).

RS2

This instruction sends and receives data in non-protocol communication using the communication board or communication adapter connected to the CPU module or using the built-in RS-485 port to the CPU module.

Ladder	ST	FBD/LD
	ENO:=RS2 (EN, s, n1, n2, n3, d);	

Setting data

■Content, range, data types

Operand	Description	Range	Data type	Data type (label)
(s)	Head device storing send data	—	16-bit signed binary / character string	ANY16
(n1)	Number of send data	0 to 4096	16-bit unsigned binary	ANY16_U
(d)	Head device storing receive data	—	16-bit signed binary / character string	ANY16
(n2)	Amount of received data	0 to 4096	16-bit unsigned binary	ANY16_U
(n3)	Communication CH	■FX5S/FX5UJ CPU module K2 to K4 ■FX5U CPU module K1 to K4 ■FX5UC CPU module K1, K3 to K4	16-bit unsigned binary	ANY16_U
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■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	$	
(s)	—	○[*1]	—	—	—	—	○	—	—	—	—
(n1)	○	○	○	○	—	—	○	○	—	—	—
(d)	—	○[*1]	—	—	—	—	○	—	—	—	—
(n2)	○	○	○	○	—	—	○	○	—	—	—
(n3)	—	—	—	—	—	—	—	○	—	—	—

[*1] T, ST, C cannot be used.

Processing details

This instruction sends and receives data via non-protocol communication by way of built-in RS-485 port, communication board or communication adapter. This instruction specifies the head device storing the sent data from the CPU module, amount of data, head device storing the received data and the maximum allowable amount of received data.

Precautions

- The RS2 instruction cannot be used in the same serial port as that of the inverter communication instruction and predefined protocol support instruction.
- To change the header, terminator, time-out time or 8-bit processing mode, do so before the RS2 instruction is driven (while off). The change is enabled when the RS2 instruction is drive. While the RS2 instruction is being driven, do not change the header, terminator, time-out time or 8-bit processing mode.

Applicable frames

Message frames used in communication can be selected by setting the communication parameter.

The table below shows message frames applicable to the RS2 instruction.

1		Data			
2		Data	CR+LF		
3		Data	Terminator		
4		Data	Terminator	CR+LF	
5		Data	Terminator	Sum check	
6		Data	Terminator	Sum check	CR+LF
7	Header	Data			
8	Header	Data	CR+LF		
9	Header	Data	Terminator		
10	Header	Data	Terminator	CR+LF	
11	Header	Data	Terminator	Sum check	
12	Header	Data	Terminator	Sum check	CR+LF

■Header

When headers are set in the communication settings (☞ Page 420 Communication Settings), the set values are stored in SD8623 and SD8624 for CH1, SD8633 and SD8634 for CH2, SD8643 and SD8644 for CH3 and SD8653 and SD8654 for CH4.

Up to four headers can be set.

Header	FX5 dedicated			
	First	Second	Third	Fourth
CH1	SD8623 (Low-order byte)	SD8623 (High-order byte)	SD8624 (Low-order byte)	SD8624 (High-order byte)
CH2	SD8633 (Low-order byte)	SD8633 (High-order byte)	SD8634 (Low-order byte)	SD8634 (High-order byte)
CH3	SD8643 (Low-order byte)	SD8643 (High-order byte)	SD8644 (Low-order byte)	SD8644 (High-order byte)
CH4	SD8653 (Low-order byte)	SD8653 (High-order byte)	SD8654 (Low-order byte)	SD8654 (High-order byte)

When data is sent, the stored data in the devices above is added at the head of the specified send data.

When data is received, receiving of the data begins when the stored data in the devices above is received continuously.

If the first header value is 00H, the header settings are not configured. The area before 00H (in 1-byte units) is used to set the headers.

■Terminator

When terminators are set in the communication settings (☞ Page 420 Communication Settings), the set values are stored in SD8625 and SD8626 for CH1, SD8635 and SD8636 for CH2, SD8645 and SD8646 for CH3 and SD8655 and SD8656 for CH4.

Up to 4 terminators can be set.

Terminator	FX5 dedicated			
	First	Second	Third	Fourth
CH1	SD8625 (Low-order byte)	SD8625 (High-order byte)	SD8626 (Low-order byte)	SD8626 (High-order byte)
CH2	SD8635 (Low-order byte)	SD8635 (High-order byte)	SD8636 (Low-order byte)	SD8636 (High-order byte)
CH3	SD8645 (Low-order byte)	SD8645 (High-order byte)	SD8646 (Low-order byte)	SD8646 (High-order byte)
CH4	SD8655 (Low-order byte)	SD8655 (High-order byte)	SD8656 (Low-order byte)	SD8656 (High-order byte)

When data is sent, the stored data in the devices above is added at the end the specified send data.

When data is received, receiving is completed[1] when the stored data in the devices above is received.

If the first terminator value is 00H, the terminator settings are not configured. The area before "00H" (in 1-byte units) is used to set the terminators.

[1] Receiving is completed also when the amount of received data specified by the RS2 instruction is received or when the receiving of data is suspended and the specified time-out occurs.

■Sum check

When the sum check code is selected as [Added] in the communication settings (☞ Page 420 Communication Settings), the sum check is executed for the send and receive data.

When [Added] is selected, always configure the terminators.

When data is sent, the sum of "data" + "terminator" is calculated, and added to the send data.

When data is received, it is checked whether or not the received sum is equivalent to the sum calculated by the PLC.

■CR+LF

When the control procedure is selected as "CR + LF provided" in the communication settings (☞ Page 420 Communication Settings), the character code of "CR + LF" is added at the end of the send data.

When data is received, receiving is completed when "CR + LF" is received continuously.

However, receiving is also completed when the specified amount of received data is received or when data receiving is suspended and the specified time-out occurs.

Make sure that "CR" is not included in the message.

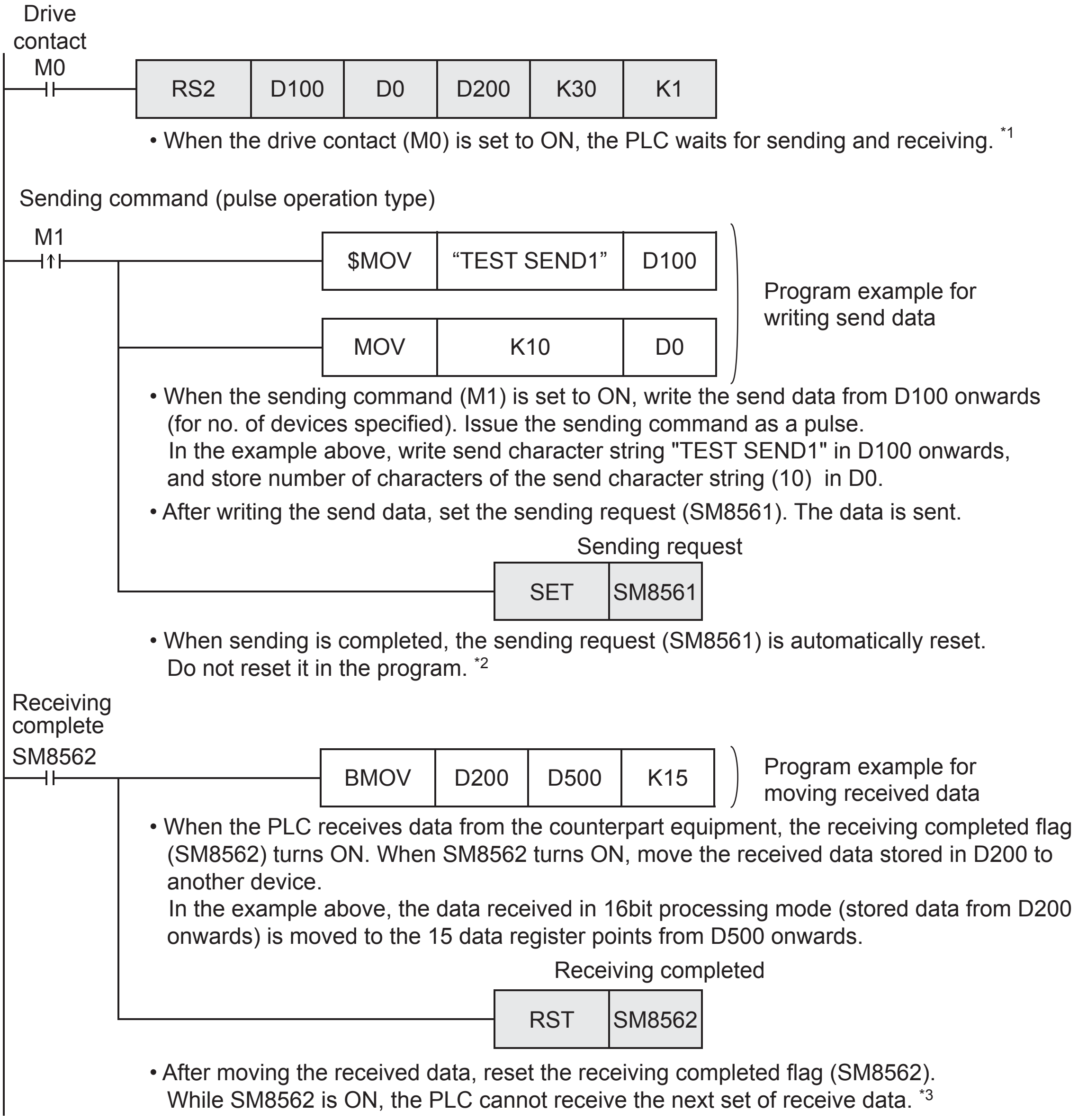
Function and operation

RS2 instruction specifies head device of the send data, no. of data, head device for storing the received data, the maximum no. of data that can be received.

Create a program as shown below.

Ex.

Serial communication settings (CH1)



- When the drive contact (M0) is set to ON, the PLC waits for sending and receiving. [*1]
- When the sending command (M1) is set to ON, write the send data from D100 onwards (for no. of devices specified). Issue the sending command as a pulse.
  In the example above, write send character string "TEST SEND1" in D100 onwards, and store number of characters of the send character string (10) in D0.
- After writing the send data, set the sending request (SM8561). The data is sent.
- When sending is completed, the sending request (SM8561) is automatically reset. Do not reset it in the program. [*2]
- When the PLC receives data from the counterpart equipment, the receiving completed flag (SM8562) turns ON. When SM8562 turns ON, move the received data stored in D200 to another device.
  In the example above, the data received in 16bit processing mode (stored data from D200 onwards) is moved to the 15 data register points from D500 onwards.
- After moving the received data, reset the receiving completed flag (SM8562). While SM8562 is ON, the PLC cannot receive the next set of receive data. [*3]

[*1] For handling of send and receive data, refer to ☞ Page 427 Send/receive data and amount of data.

[*2] For the operation when sending data, refer to ☞ Page 431 Operation during data send.

[*3] For the operation when receiving data, refer to ☞ Page 431 Operation during data receive.

Send/receive data and amount of data

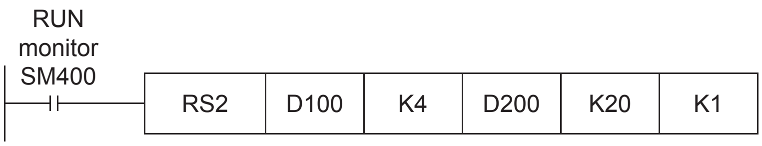
RS2 instruction can handle sent and received data in two modes, 16-bit mode and 8-bit mode. The processing mode is set in the communication settings (☞ Page 420 Communication Settings), and when the RS2 instruction is driven, the mode is switched to the set mode.

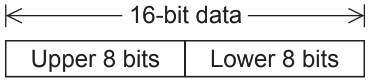
The handling of data is as follows.

Ex.

Serial communication settings (CH1)

■16-bit mode



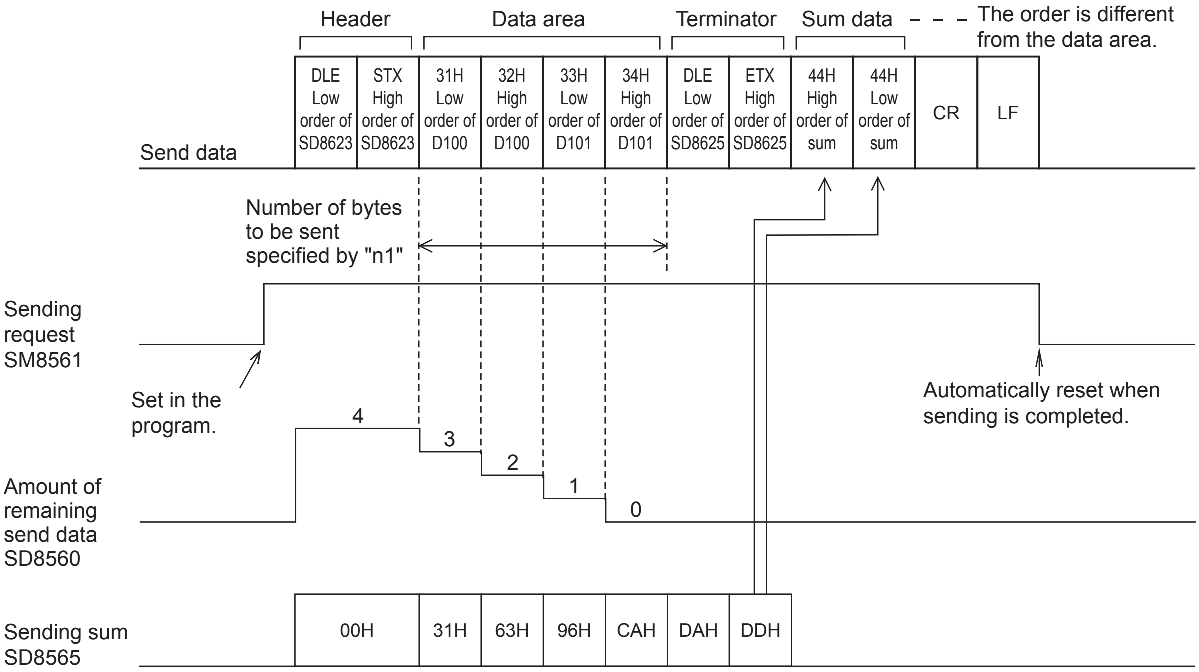


16-bit data is divided into upper 8 bits and lower 8 bits when it is sent or received.

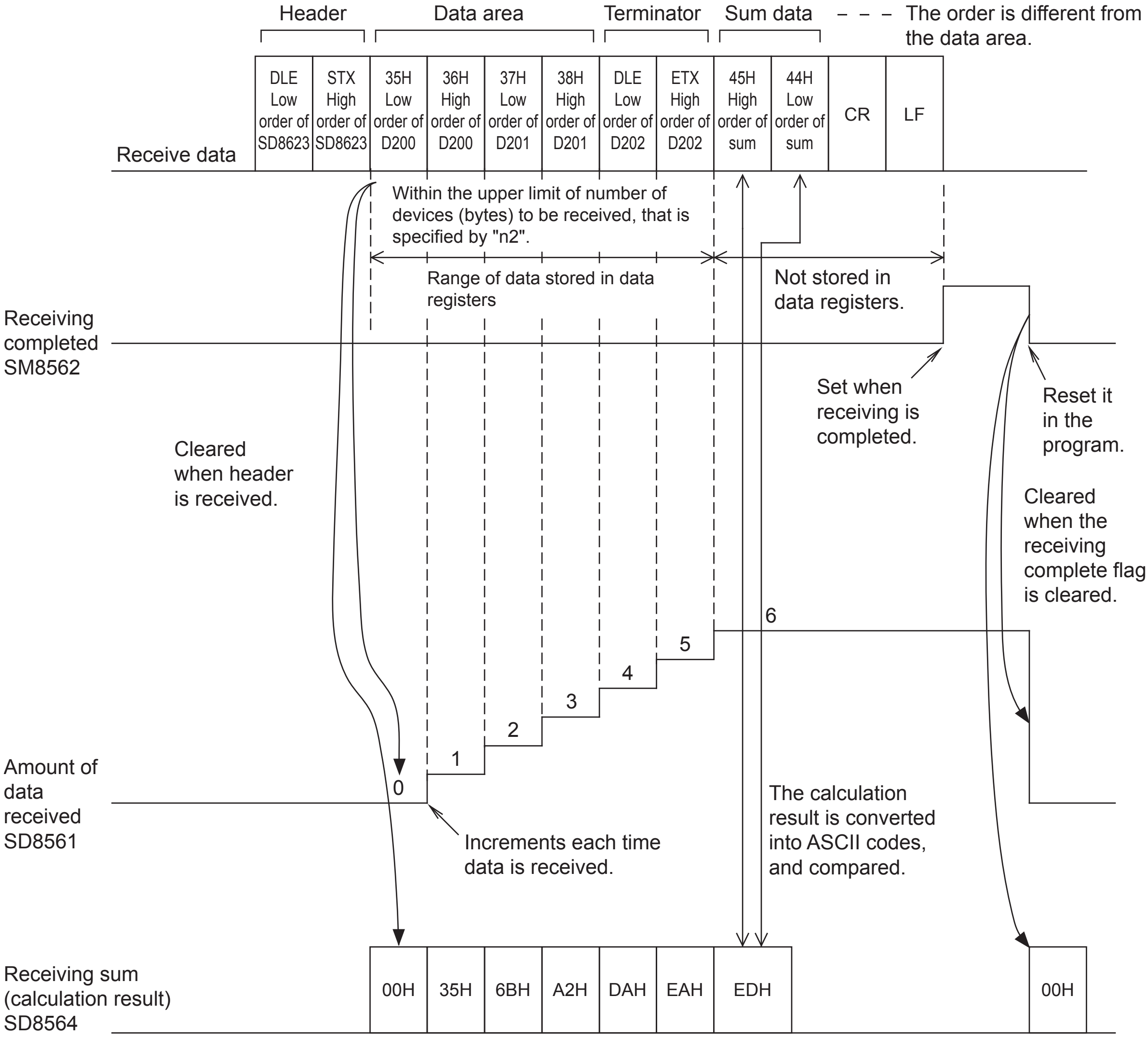
Serial communication setting

- Header : [Added] [DLE + STX (SD8623: 0210H, SD8624: 0000H)]
- Terminator : [Added] [DLE + ETX (SD8625: 0310H, SD8626: 0000H)]
- Sum check code : [Added]
- Control procedure : [CR, LF added]
- Control mode : [None (RS-232C)]

- Send data and amount of remaining send data



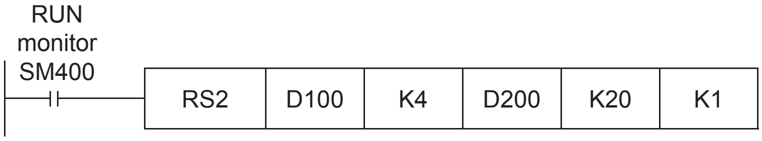
- Receive data and amount of data received

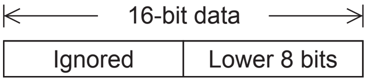


Precautions

When odd data is received, the higher-order 8 bits of data of the last device holds the data received previously.

■8-bit mode



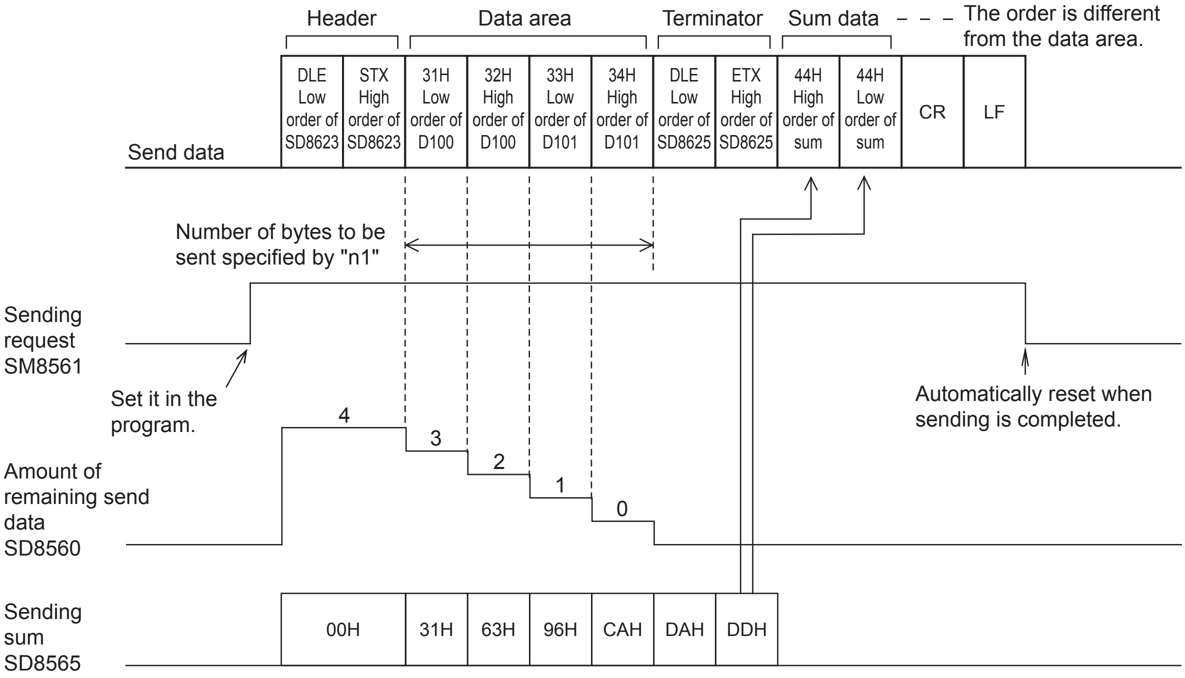


The upper 8 bits are ignored.
Only the lower 8 bits are valid.

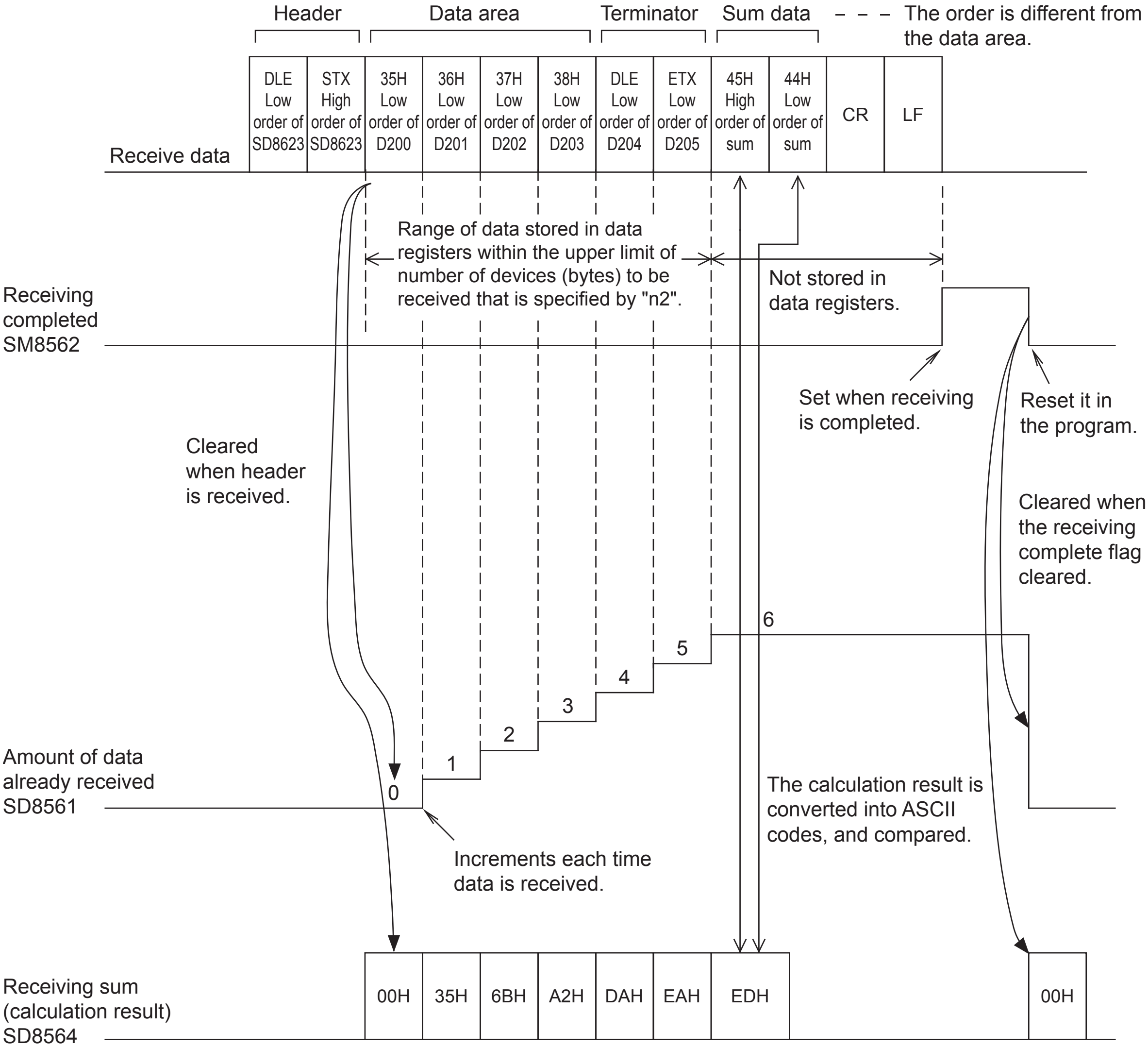
Serial communication setting

- Header : [Added] [DLE + STX (SD8623: 0210H, SD8624: 0000H)]
- Terminator : [Added] [DLE + ETX (SD8625: 0310H, SD8626: 0000H)]
- Sum check code : [Added]
- Control procedure : [CR, LF added]
- Control mode : [None (RS-232C)]

- Send data and amount of remaining send data



- Receive data and amount of data received



Precautions

The higher-order 8 bits of data of the receive data storage destination device holds the data previously received.

Operation during data send

When the sending request flag is set to ON while RS2 instruction is driven, the PLC sends the data stored in (s) to ((s) + (n1) - 1).

Name	FX5 dedicated			
	CH1	CH2	CH3	CH4
Send request flag	SM8561	SM8571	SM8581	SM8591

When sending of the data is completed, the sending request flag is automatically set to OFF.

■Time at which sending is started

When RS2 instruction is executed after the sending request flag is set to ON, the PLC starts to send.

When sending is started, the PLC sends the data specified by RS2 instruction in interrupt processing regardless of the operation cycle.

■Time at which sending is completed

When all send data[1] is sent, sending is completed.

[1] The terminators, sum check and CR+LF configured are also included.

■Precautions for sending

When sending data, please take care about the following.

- While the sending request flag is ON, do not change the amount of send data or the contents of the send data.
- Do not set the sending request flag to OFF in the program. If the send data is changed while the sending request flag (M8122) is ON or if the sending request flag (M8122) is set to OFF in the sequence program, correct data will not be sent.

Operation during data receive

When executing the RS2 instruction, the PLC waits to receive. When the PLC receives data from the counterpart equipment and receiving data is completed, the receiving completed flag turns ON.

Name	FX5 dedicated			
	CH1	CH2	CH3	CH4
Receiving completed flag	SM8562	SM8572	SM8582	SM8592

When the PLC receives data, it stores the received data to (d) to ((d) + (n2) - 1).

While the receiving completed flag (M8123) is ON, the PLC cannot receive new data.

■Time at which receiving is started

When the PLC receives data while it is waiting, it starts receiving data.

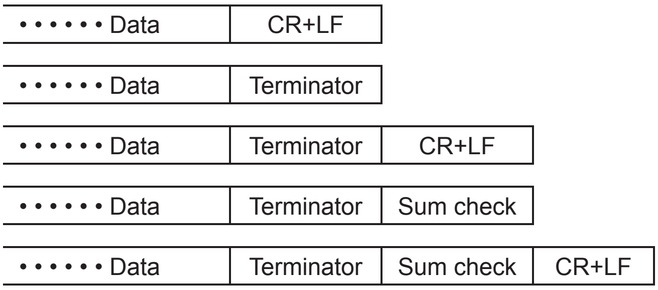
When receiving begins, the PLC stores the received data in interrupt processing regardless of the operation cycle.

However, when the headers are specified in the communication settings, the PLC starts receiving when it continuously receives the codes set in the headers. The PLC stores the received data except the headers.

■Time at which receiving is completed

The time at which receiving is completed falls into the following 3 types. Receiving is completed when one of the following conditions is satisfied.

- When the PLC receives the amount of receive data specified by the RS2 instruction
- The terminators, sum check and CR+LF configured in the communication settings (☞ Page 420 Communication Settings) are received correctly.



- When data receiving is suspended and the PLC does not receive the next data within the time-out time, the time-out flag turns ON.

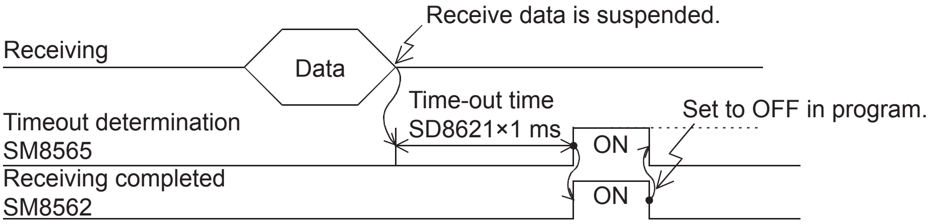
■Operation of time-out flag

When data receiving is suspended, and the PLC does not receive the next within the time-out time, the time-out flag is set to ON.

At this time, the receiving completed flag is also set to ON.

The time-out time can be set within 1 to 32767 (ms). If the set value is outside the range, it operates as 10ms.

Name	**FX5 dedicated**				**FX3 Series compatible**	
	CH1	**CH2**	**CH3**	**CH4**	**CH1**	**CH2**
Time-out flag	SM8565	SM8575	SM8585	SM8595	SM8409	SM8429
Time-out time	SD8621	SD8631	SD8641	SD8651	—	



The time-out flag does not automatically turn OFF. Reset using program. When the receiving completed flag is set to OFF, the time-out flag is also set to OFF.

Using this function, the PLC can receive data from equipment where the amount of send data varies without the terminators.

■When the control line is set to interlink mode

When the interlink mode is set in the communication parameters, the following sequence is adopted from the start of receiving to completion:

1. When the amount of data already received becomes "preset amount of received data -30", the control line ER (DTR) turns OFF. When the control line ER (DTR) turns OFF, the external equipment should suspend data sending. After the control line ER (DTR) turns OFF, the PLC can receive up to 30 characters (bytes).
2. When the external equipment suspends data sending, the PLC sets the time-out check flag and receiving complete flag to ON after the preset time-out time setting. Move the received data in a sequence program, and then set the receiving complete flag and time-out check flag to OFF.
3. When the receiving complete flag is set to OFF, the control line ER (DTR) turns ON. When the control line ER (DTR) turns ON, restart data sending from the external equipment.
4. Repeat steps 1) to 3) until data receiving is completed.

■Precautions for receiving

When receiving data, please take care about the following.

- While the receiving completed flag is ON, the PLC cannot receive the next set of receive data. When the receiving completed flag is set to OFF, the PLC waits to receive.
- If the RS2 instruction is driven while the amount of received data "n2" is 0, the receiving completed flag turns ON. To make the PLC wait to receive, set the amount of received data (n2) to 1 or more, and set the receiving completed flag from ON to OFF.
- Set the amount of received data to a value including "terminators", "sum check" and "CR+LF". If the specified amount of received data is less, the serial communication error flag turns ON. When the terminator is set to [Do Not Add] in the communication settings (☞Page 420 Communication Settings), an error does not occur.

Full-duplex, bi-directional communication operation

■Send completed → start send operation

When sending of the data is completed, the sending request flag is automatically set to OFF.

When RS2 instruction is executed again after the sending request flag is set to ON, the PLC starts to send.

■Send completed → start receive operation

For full-duplex, bi-directional communication, send and receive can operate simultaneously.

For the operation timing of send completed and start receive, refer to the following.

- Operation during data send (☞ Page 431 Operation during data send)
- Operation during data receive (☞ Page 431 Operation during data receive)

■Receive completed → start send operation

For full-duplex, bi-directional communication, send and receive can operate simultaneously.

For the operation timing of send completed and start receive, refer to the following.

- Operation during data send (☞ Page 431 Operation during data send)
- Operation during data receive (☞ Page 431 Operation during data receive)

■Receive completed → start receive operation

Receiving completed flag turns ON when receiving is completed.

While the receiving completed flag is ON, the PLC cannot receive new data.

When the receiving completed flag is reset, the PLC waits to receive (the state where it can receive data).

Half-duplex, bi-directional communication operation

■Send completed → start send operation

When sending of the data is completed, the sending request flag is automatically set to OFF.

When RS2 instruction is executed again after the sending request flag is set to ON, the PLC starts to send.

■Send completed → start receive operation

For the operation timing of send completed and start receive, refer to the following.

- Operation during data send (☞ Page 431 Operation during data send)
- Operation during data receive (☞ Page 431 Operation during data receive)

However, leave 100 μs or more time between send completed → start receive.

■Receive completed → start send operation

For the operation timing of send completed and start receive, refer to the following.

- Operation during data send (☞ Page 431 Operation during data send)
- Operation during data receive (☞ Page 431 Operation during data receive)

■Receive completed → start receive operation

Receiving completed flag turns ON when receiving is completed.

While the receiving completed flag is ON, the PLC cannot receive new data.

When the receiving completed flag is reset, the PLC waits to receive (the state where it can receive data).

Sum check code

The sum check code is a two-digit ASCII code converted from the least significant byte (8-bit) of the sum obtained by adding the corresponding data as hexadecimal numbers.

One can select whether to include the sum check in the message or not, using the communication settings (☞ Page 420 Communication Settings).

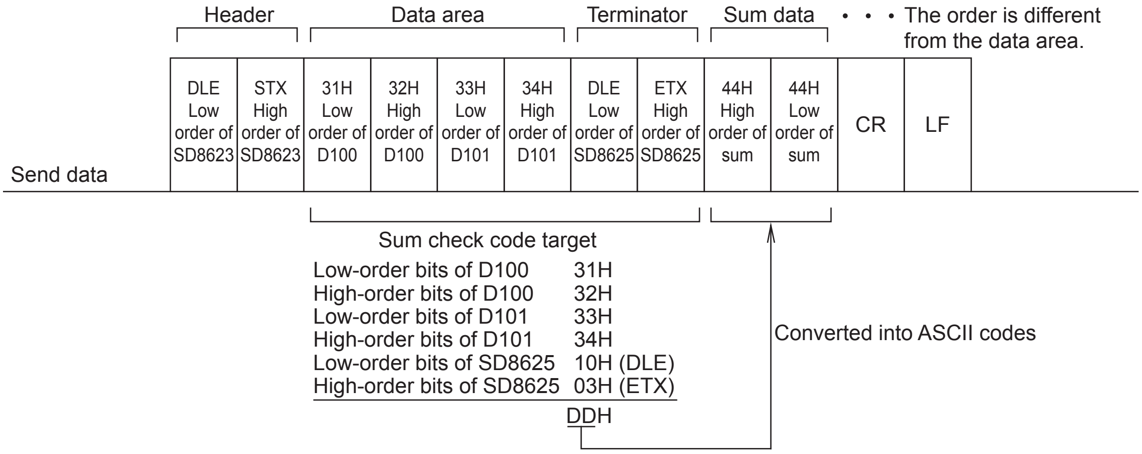
- When "sum check code added" is selected, the sum check code is added in the message during sending. During receiving, the sum check code is compared with the value calculated from the received data to check the received data.
- When "sum check not added" is selected, the sum check code is not added, so the received data is not checked either. A calculation example of the sum check code is shown below.

Ex.

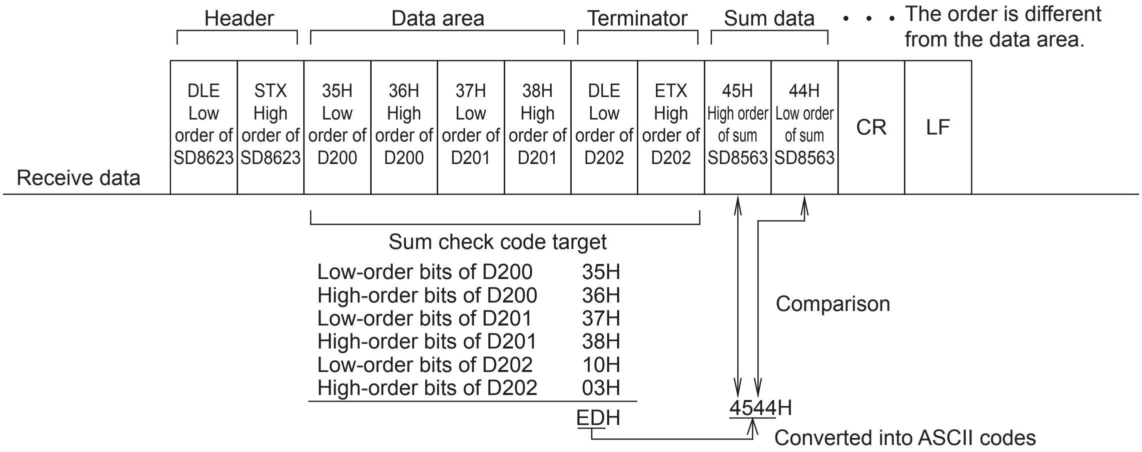
Serial communication settings (CH1)

- Header: [Provided] [DLE+STX (SD8623: 0210H, SD8624: 0000H)]
- Terminator: [Provided] [DLE+ETX (SD8625: 0310H, SD8626: 0000H)]
- Sum Check Code: [Added]
- Control Procedure: [CR, LF Added]
- Control Mode (RS-232C): No Control Line

■For send data



■For receive data



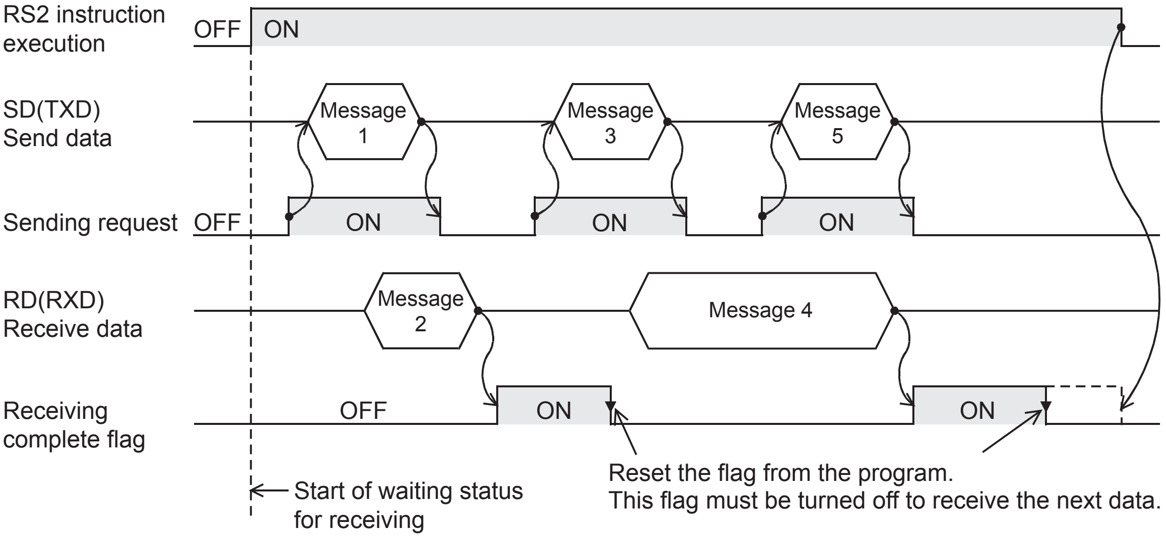
Operation of control line (RS-232C)

RS-232C communication is full-duplex, bi-directional communication. When using half-duplex, bi-directional communication, pay attention not to turn ON the sending request flag while receiving. If set to ON, the PLC starts to send and as a result, the counterpart equipment may not be able to receive data, and the sent and received data may be destroyed.

In full-duplex, bi-directional communication, the sending wait flag does not turn ON.

However, in the case that control is standard mode or interlink mode, the sending wait flag remains ON while DR (DSR) is OFF and the PLC is waiting to send.

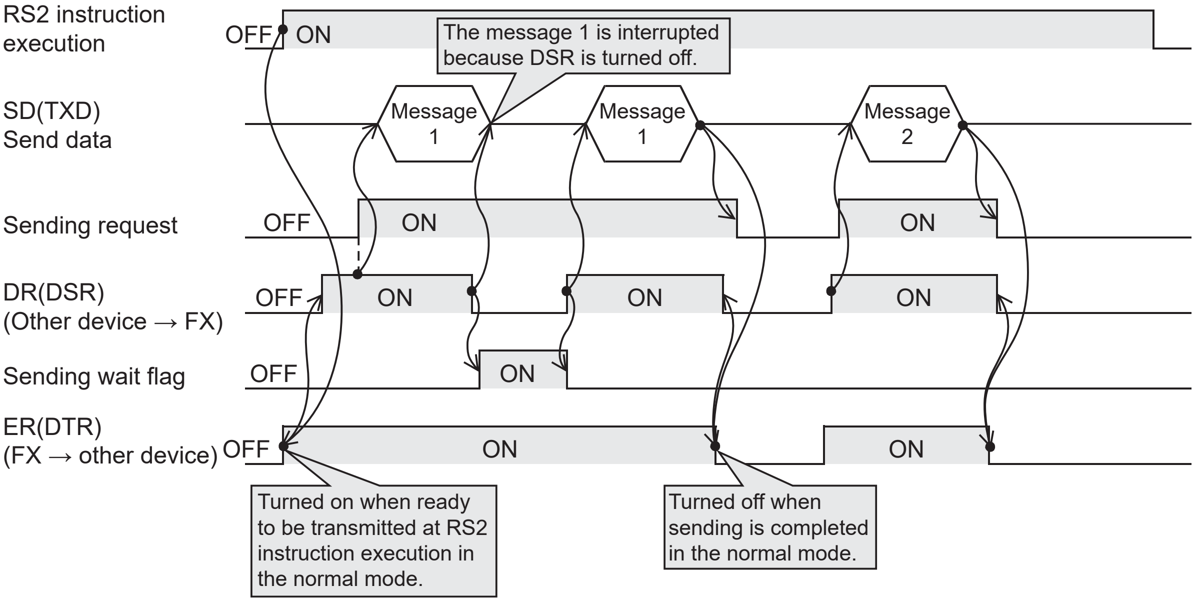
Control line not used



Normal mode

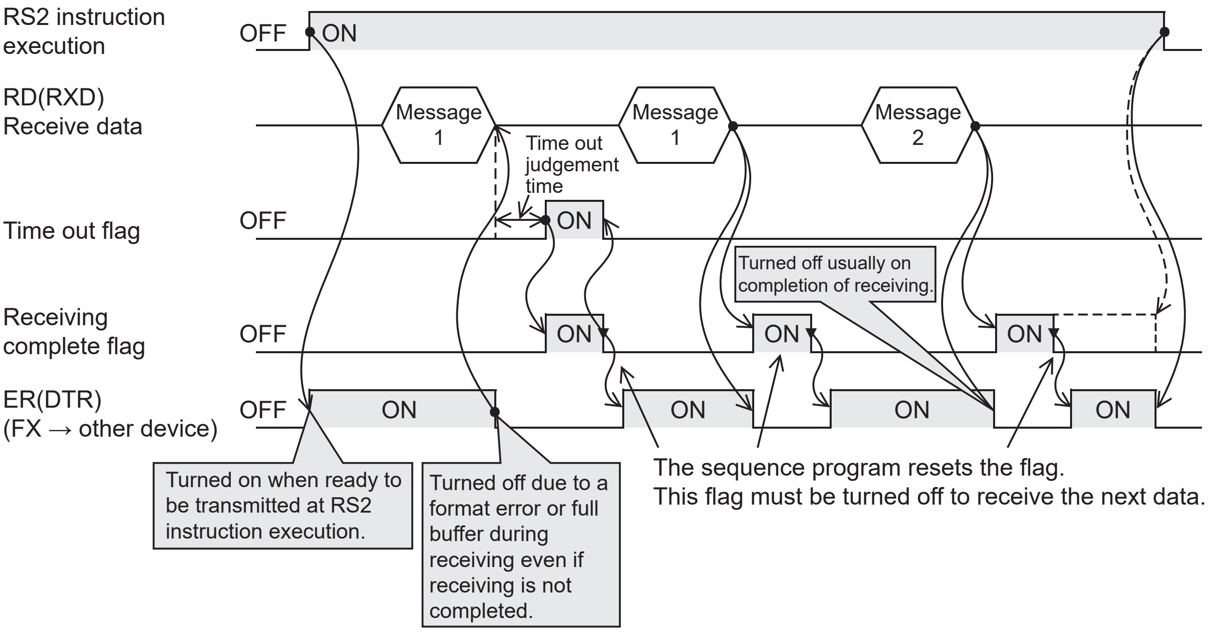
Use this mode for only sending or only receiving.

■Send only

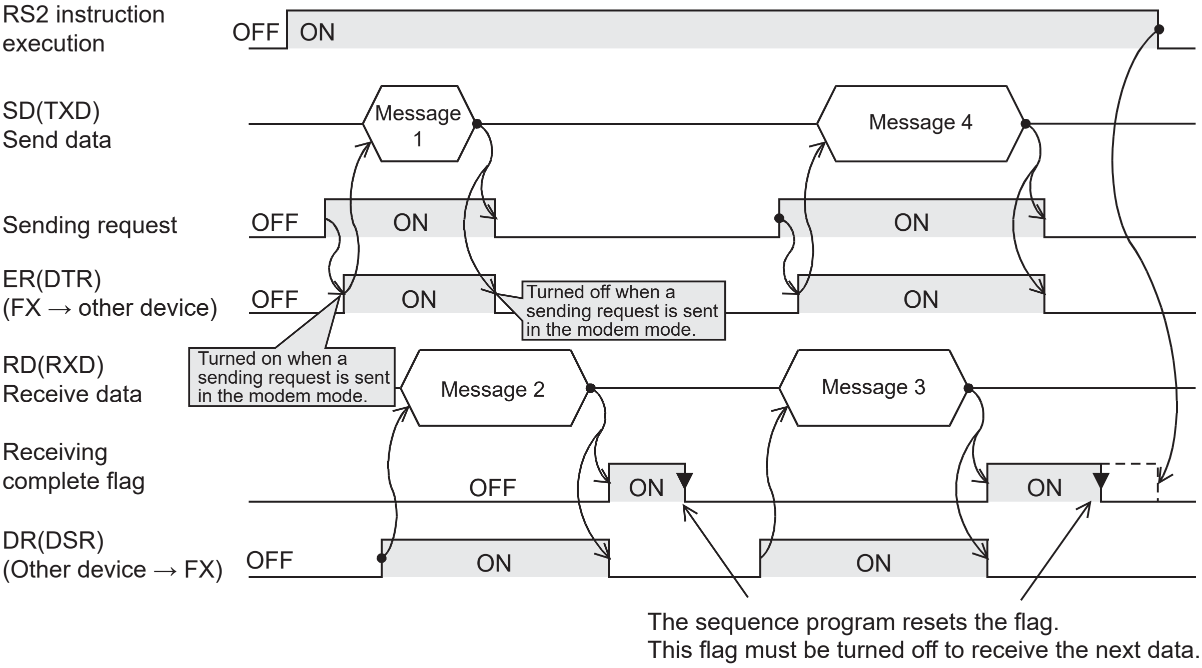


■Receive only

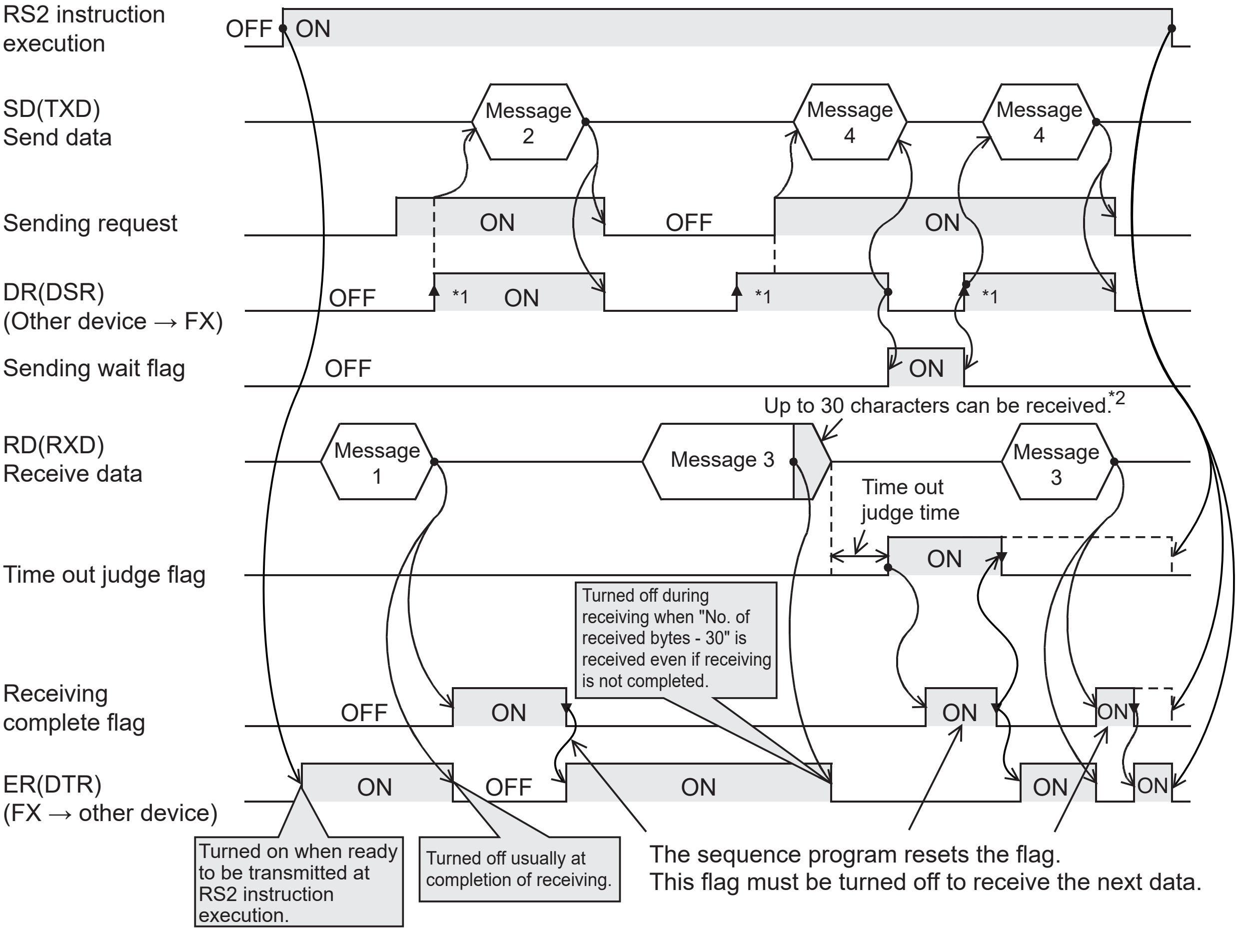
The DR (DSR) signal is not used.



Modem mode



Interlink mode



*1 On the external equipment side, set the DR (DSR) signal to ON when the external equipment is ready to receive.
The FX5 PLC sends the send data when both the DR (DSR) signal and the sending request turn ON.

*2 In the interlink mode, the PLC sets ER (DTR) to OFF when the amount of data already received becomes "Number of bytes to be received -30", and asks the external equipment to stop sending. After this send request, the PLC can only receive up to 30 characters and no more, so the external equipment temporarily stops sending, and then sends the remaining data after the ER (DTR) signal turns ON again.
When sending is stopped, the receiving complete flag turns ON after the time-out check time and receiving completes.
When sending is not stopped, the PLC finishes receiving after it has received the final send data or 30 characters. Accordingly, make sure that the amount of receive data is "30 + α".

Precautions for program creation

Using multiple RS2 instructions

The RS2 instructions can be used as many times as necessary in a program, but make sure that only one RS2 instruction is driven in each serial port at a time. For switching of the RS2 instruction used, make sure the OFF time is one scan time or more.

Communication protocol setting

If protocol of communication settings (☞ Page 420 Communication Settings) for the serial port to be used is not set to "Non-protocol Communication", RS2 instructions cannot be used.

Using RS2 instruction together with another instruction

RS2 instruction cannot be used in the same serial port which is being used for any other communication (such as inverter communication).

(☞ Page 877 Combined Use of Serial Communication)

Control line interlink mode

In the interlink mode, set the amount of received data (n) to 31 or more. If it is set to 30 or less, the control line ER (DTR) is set to OFF as soon as the PLC receives data.

Using RS-485 half-duplex, bi-directional communication

Be sure to check the following.

- Leave 100 μs or more time between send completed → start receive.
- Do not turn ON the sending request flag while receiving.
- Since data cannot be received while sending, do not send data from the counterpart equipment to the PLC.
- When using header or terminator, set before driving the RS2 instruction. Accordingly, do not change the setting while the RS2 instruction is driven.

Example of printing using RS2 instruction

In this example, the set data is sent to a printer connected to an expansion board.

Contents of operation

The send data is sent to the printer by the input (X) of the FX5U CPU module.

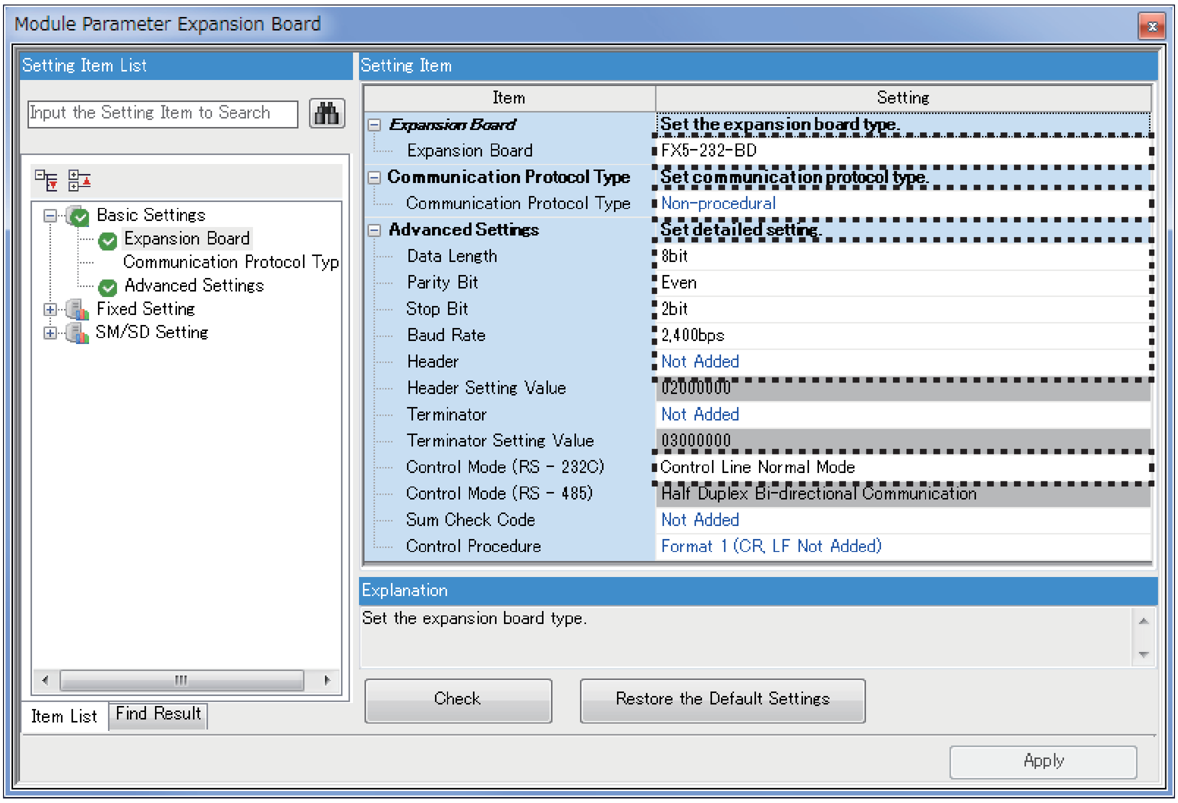
Item	CH2
RS2 instruction execution	X0
Sending of data	X1
Send data	D10 to D15
Send request flag	SM8571

Parameter setting

Set the following parameter.

- Expansion board: FX5-232-BD
- Protocol type: Non-protocol communication
- Data length: 8bit
- Parity bit: Even
- Stop bit: 2bit
- Baud Rate: 2400bps
- Control Mode (RS-232C): Control Line Normal Mode

Navigation Window ⇨ Parameter ⇨ FX5UCPU ⇨ Module Parameter ⇨ Expansion Board

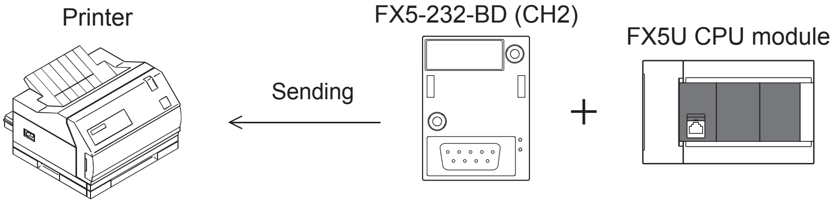


Item	Setting
Expansion Board	Set the expansion board type.
Expansion Board	FX5-232-BD
Communication Protocol Type	Set communication protocol type.
Communication Protocol Type	Non-procedural
Advanced Settings	Set detailed setting.
Data Length	8bit
Parity Bit	Even
Stop Bit	2bit
Baud Rate	2,400bps
Header	Not Added
Header Setting Value	02000000
Terminator	Not Added
Terminator Setting Value	03000000
Control Mode (RS - 232C)	Control Line Normal Mode
Control Mode (RS - 485)	Half Duplex Bi-directional Communication
Sum Check Code	Not Added
Control Procedure	Format 1 (CR, LF Not Added)

Explanation: Set the expansion board type.

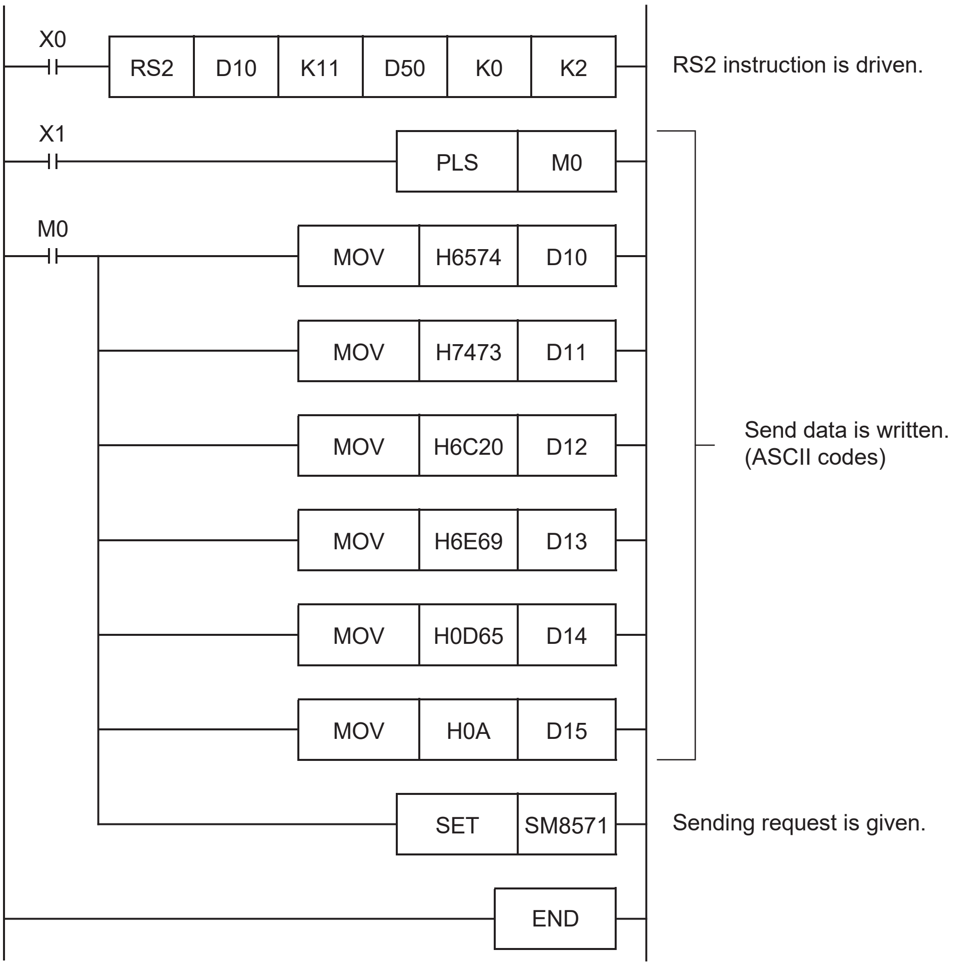
System configuration

The following shows an example of a system configuration in which a printer with an RS-232C interface is connected to the expansion board (CH2) connected to the FX5U CPU module, and the data sent from the FX5U CPU module is printed.



Use a communication cable suitable to the pin arrangement of the connector of the printer used.

■Program example



20.8 Caution on Write During RUN, When RS2 Instruction Is Driven

When the RS2 instruction is eliminated by writing during RUN, non-protocol communication stops immediately.

20.9 Related Devices

This section explains the special relay/special register functions used in the non-protocol communication function.

Point

Available communication channels vary depending on the CPU module and system configuration.
For communication channels, refer to ☞ Page 412 System Configuration.
"FX3 Series compatible" devices operate only on the communication channel specified in the compatible SM/SD for communication settings.
For compatible SM/SD, refer to ☞ Page 420 Communication Settings.

List of related devices

Special relays

■FX5 only

R: Read only, R/W: Read/write

Device No.				Name	Description	R/W
CH1	CH2	CH3	CH4			
SM8500	SM8510	SM8520	SM8530	Serial communication error	Turns ON when an error occurs in serial communication.	R
SM8560	SM8570	SM8580	SM8590	Sending wait flag	This device remains ON while the PLC is waiting to send.	R
SM8561	SM8571	SM8581	SM8591	Send request flag	When this device is set to ON, the PLC starts to send.	R/W
SM8562	SM8572	SM8582	SM8592	Receiving completed flag	This device turns ON when receiving is completed.	R/W
SM8563	SM8573	SM8583	SM8593	Carrier detection flag	This device turns ON in synchronization with the CD (DCD) signal.	R
SM8564	SM8574	SM8584	SM8594	DSR detection flag	This device turns ON in synchronization with the DR (DSR) signal.	R
SM8565	SM8575	SM8585	SM8595	Time-out flag	This device turns ON when data receiving is suspended and the next set of receive data is not given within the time set by the time-out time setting device.	R

■FX3 Series compatible

R: Read only

Device No.		Name	Description	R/W
CH1	CH2			
SM8063	SM8438	Serial communication error	Turns ON when an error occurs in serial communication.	R
SM8401	SM8421	Sending wait flag	This device remains ON while the PLC is waiting to send.	R
SM8404	SM8424	Carrier detection flag	This device turns ON in synchronization with the CD (DCD) signal.	R
SM8405	SM8425	DSR detection flag	This device turns ON in synchronization with the DR (DSR) signal.	R
SM8409	SM8429	Time-out flag	This device turns ON when data receiving is suspended and the next set of receive data is not given within the time set by the time-out time setting device.	R

Special registers

■FX5 only

R: Read only, R/W: Read/write

Device No.				Name	Description	R/W
CH1	CH2	CH3	CH4			
SD8500	SD8510	SD8520	SD8530	Serial communication error code	When a serial communication error occurs, the error code is stored.	R
SD8502	SD8512	SD8522	SD8532	Serial communication settings	Stores the setting of the communication parameter.	R
SD8503	SD8513	SD8523	SD8533	Serial communication operation mode	This device stores the current communication mode being used.	R
SD8560	SD8570	SD8580	SD8590	Amount of remaining send data	These devices store the amount of remaining send data.	R
SD8561	SD8571	SD8581	SD8591	Amount of data already received	This device stores the amount of received data.	R
SD8563	SD8573	SD8583	SD8593	Receive data sum	These devices store the received sum check value.	R
SD8564	SD8574	SD8584	SD8594	Receiving result sum	These devices store the sum check value calculated using the received data.	R
SD8565	SD8575	SD8585	SD8595	Sending sum	These devices store the sum check value added to the send data.	R
SD8621	SD8631	SD8641	SD8651	Time-out time	These devices store the time-out time set in the communication settings.	R/W
SD8622	SD8632	SD8642	SD8652	8-bit processing mode	These devices store the send/receive data processing mode set in the communication settings.	R/W
SD8623 SD8624	SD8633 SD8634	SD8643 SD8644	SD8653 SD8654	Header	These devices store the contents of headers 1 to 4 set in the communication settings.	R/W
SD8625 SD8626	SD8635 SD8636	SD8645 SD8646	SD8655 SD8656	Terminator	These devices store the contents of terminators 1 to 4 set in the communication settings.	R/W

■FX3 Series compatible

R: Read only

Device No.		Name	Description	R/W
CH1	CH2			
SD8063	SD8438	Serial communication error code	When a serial communication error occurs, the error code is stored.	R
SD8402	SD8422	Amount of remaining send data	These devices store the amount of remaining send data.	R
SD8403	SD8423	Amount of data already received	This device stores the amount of received data.	R
SD8405	SD8425	Communication parameter display	Stores the setting of the communication parameter.	R
SD8414	SD8434	Receive data sum	These devices store the received sum check value.	R
SD8415	SD8435	Receiving result sum	These devices store the sum check value calculated using the received data.	R
SD8416	SD8436	Sending sum	These devices store the sum check value added to the send data.	R
SD8419	SD8439	Serial communication operation mode	This device stores the current communication mode being used.	R

Details of related devices

Serial communication error

Turns ON when an error occurs in serial communication. These flags are for check of the serial communication error.

R: Read only

FX5 dedicated				**FX3 Series compatible**		**Description**	**R/W**
CH1	**CH2**	**CH3**	**CH4**	**CH1**	**CH2**		
SM8500	SM8510	SM8520	SM8530	SM8063	SM8438	Turns ON when an error occurs in serial communication.	R

After a device above turns ON, the error code is stored in the corresponding device below.

FX5 dedicated				**FX3 Series compatible**		**Name**	**Description**
CH1	**CH2**	**CH3**	**CH4**	**CH1**	**CH2**		
SD8500	SD8510	SD8520	SD8530	SD8063	SD8438	Serial communication error code	When a serial communication error occurs, the error code is stored.

Precautions

Do not turn ON or OFF with program or engineering tool.

Serial communication error does not turn OFF even if the communication is restored to normal state. The devices turn OFF when power is turned OFF→ON, STOP→RUN, reset or SM50 (Error Detection Reset Completion) is turned ON.

Sending wait flag

These devices remain ON while the PLC is waiting to send.

R: Read only

FX5 dedicated				**FX3 Series compatible**		**Description**	**R/W**
CH1	**CH2**	**CH3**	**CH4**	**CH1**	**CH2**		
SM8560	SM8570	SM8580	SM8590	SM8401	SM8421	These devices turn ON when serial communication is waiting to send.	R

- For RS-232C

While the control line is set to the standard or interlink mode in the communication parameter, when the control line DR (DSR) turns OFF while sending data, the sending wait flag turns ON.

- For RS-485

The sending wait flag does not turn ON.

Precautions

Do not turn ON or OFF with program or engineering tool.

Send request flag

When these devices are set to ON by SET instruction, the PLC starts to send. When sending of the data is completed, the sending request flag is automatically reset.

R/W: Read/write

FX5 dedicated				**Description**	**R/W**
CH1	**CH2**	**CH3**	**CH4**		
SM8561	SM8571	SM8581	SM8591	When set to ON, serial communication data sending starts. Set to ON with the SET instruction.	R/W

Precautions

When setting these devices to ON, set the drive condition with a pulse.

Receiving completed flag

Receiving completed flag turns ON when receiving is completed. When these devices turn ON, data cannot be received.

R/W: Read/write

FX5 dedicated				Description	R/W
CH1	CH2	CH3	CH4		
SM8562	SM8572	SM8582	SM8592	These devices turn ON when serial communication data receiving is complete.	R/W

Receiving is completed in one of the following three conditions.

- When the PLC receives the amount of receive data specified by the RS2 instruction
- When the terminators are set, and the PLC receives the code set by the terminator
- When data receiving is suspended and the PLC does not receive the next set of data within the time set by the time-out time setting device

When the receiving completed flag turns ON, transfer the received data to another storage destination, and then set this flag to OFF.

When this flag is set to OFF, the PLC waits to receive.

Precautions

When the RS2 instruction is driven while the amount of receive data is set to "0", the PLC does not wait to receive. To make the PLC wait to receive, it is necessary to set the amount of receive data to "1" or more and set the receiving completed flag ON→OFF.

Carrier detection flag

These devices turn ON/OFF in synchronization with the CD (DCD) signal.

R: Read only

FX5 dedicated				FX3 Series compatible		Description	R/W
CH1	CH2	CH3	CH4	CH1	CH2		
SM8563	SM8573	SM8583	SM8593	SM8404	SM8424	When the CD (DCD) signal of the counterpart device that is communicating via communication is turned ON, this synchronizes with it and turns ON.	R

While the carrier detection flag is ON, data can be sent and received.

Precautions

Do not turn ON or OFF with program or engineering tool.

DSR detection flag

These devices turn ON/OFF in synchronization with the DR (DSR) signal. The state of DR (DSR) signal can be checked when executing RS2 instruction.

R: Read only

FX5 dedicated				FX3 Series compatible		Description	R/W
CH1	CH2	CH3	CH4	CH1	CH2		
SM8564	SM8574	SM8584	SM8594	SM8405[*1]	SM8425[*1]	When the DR (DSR) signal of the counterpart device that is communicating via communication is turned ON, this synchronizes with it and turns ON.	R

[*1] The device is updated during END processing.

Precautions

Do not turn ON or OFF with program or engineering tool.

Time-out flag

This turns ON when data receiving is suspended, and the next set of receive data is not received within the time set by the time-out time setting device. The receiving completed flag also turns ON.

R: Read only

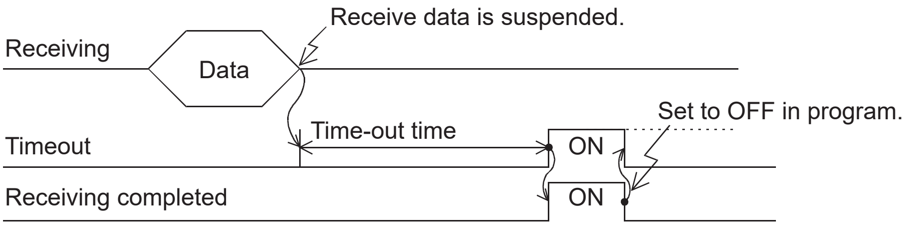
FX5 dedicated				FX3 Series compatible		Description	R/W
CH1	CH2	CH3	CH4	CH1	CH2		
SM8565	SM8575	SM8585	SM8595	SM8409	SM8429	This turns ON when receiving is not restarted within the specified time-out time setting when data receiving is interrupted.	R

The time-out time corresponding to the above devices is stored in the following devices.

FX5 dedicated				FX3 Series compatible		Name	Description
CH1	CH2	CH3	CH4	CH1	CH2		
SD8621	SD8631	SD8641	SD8651	—	—	Time-out time	1 to 32767 (ms) 10 ms when set to a value outside the range.

The time-out flag turns OFF when the PLC waits to receive.

Using this function, the PLC can receive data from equipment where the amount of send data varies, without the terminators.



The time-out check is judged in END processing based on a 1 ms free counter set when the data receiving starts and the time-out flag is turned ON. A lag of maximum 1 scan time occurs due to processing in the END processing.

Precautions

Do not turn ON or OFF with program or engineering tool.

Serial communication error code

When a serial communication error occurs, these devices store the corresponding error codes (☞ Page 844 Checking absence/presence of non-protocol communication function errors).

R: Read only

FX5 dedicated				FX3 Series compatible		Description	R/W
CH1	CH2	CH3	CH4	CH1	CH2		
SD8500	SD8510	SD8520	SD8530	SD8063	SD8438	When a serial communication error occurs, the error code is stored.	R

Precautions

Do not change the value with program or engineering tool.

Serial communication error code is not cleared even after communication is restored to normal state. They are cleared by turning the PLC power OFF→ON, STOP→RUN, reset or by SM50 (Error Detection Reset Completion) is turned ON.

Serial communication settings

The communication parameters set in the communication settings are stored when the power is turned OFF→ON, STOP→RUN or the reset. (☞ Page 420 Communication Settings)

R: Read only

FX5 dedicated				Description	R/W
CH1	**CH2**	**CH3**	**CH4**		
SD8502	SD8512	SD8522	SD8532	Stores the setting of the communication parameter.	R

The descriptions of the communication parameters are as follows.

Bit No.	Name	Description	
		0 (bit is OFF)	**1 (bit is ON)**
b0	Data length	7 bits	8 bits
b1 b2	Parity bit	b2, b1 (0, 0): N/A (0, 1): Odd (1, 1): Even	
b3	Stop bit	1 bit	2 bits
b4 b5 b6 b7	Baud rate	b7, b6, b5, b4 (0, 0, 1, 1): 300bps (0, 1, 0, 0): 600bps (0, 1, 0, 1): 1200bps (0, 1, 1, 0): 2400bps (0, 1, 1, 1): 4800bps (1, 0, 0, 0): 9600bps (1, 0, 0, 1): 19200bps (1, 0, 1, 0): 38400bps (1, 0, 1, 1): 57600bps (1, 1, 0, 1): 115200bps	
b8	Header	Not added	Added
b9	Terminator	Not added	Added
b10 b11 b12	Control mode	b12, b11, b10 (0, 0, 0): Not provided <RS-232C interface> (0, 0, 1): Standard mode <RS-232C interface> (0, 1, 0): Interlink mode <RS-232C interface> (0, 1, 1): Modem mode <RS-232C interface> (1, 1, 0): RS-485 full-duplex, bi-directional <RS-485 interface> (1, 1, 1): RS-485 half-duplex, bi-directional <RS-485 interface>	
b13	Sum check	Not added	Added
b14[*1]	Protocol	Not used	Used
b15	CR, LF	CR and LF are not added	CR/LF added

*1 When using non-protocol communication, 0 (fixed).

Precautions

Do not change the value with program or engineering tool.

Serial communication operation mode

Stores the communication function code used in the serial communication under execution.

R: Read only

FX5 dedicated				**FX3 Series compatible**		**Description**	**R/W**
CH1	**CH2**	**CH3**	**CH4**	**CH1**	**CH2**		
SD8503	SD8513	SD8523	SD8533	SD8419	SD8439	0: MELSOFT connection or MC protocol 3: N:N Network Communication 5: Non-Protocol Communication 6: Parallel Link Communication 7: Inverter Communication 9: MODBUS RTU Communication 12: Predefined protocol support Other than above: Not used	R

Precautions

Do not change the value with program or engineering tool.

These devices store "5" while RS2 instruction is being driven or not, as long as the communication mode has not changed.

Amount of remaining send data

These devices store the amount of remaining send data. These devices store the counted value in 8-bit (1 byte) units.

R: Read only

FX5 dedicated				**FX3 Series compatible**		**Description**	**R/W**
CH1	**CH2**	**CH3**	**CH4**	**CH1**	**CH2**		
SD8560	SD8570	SD8580	SD8590	SD8402	SD8422	These devices store the amount of remaining send data.	R

Precautions

Do not change the value with program or engineering tool.

Amount of data received

This stores the amount of data already received. These devices store the counted value in 8-bit (1 byte) units.

R: Read only

FX5 dedicated				**FX3 Series compatible**		**Description**	**R/W**
CH1	**CH2**	**CH3**	**CH4**	**CH1**	**CH2**		
SD8561	SD8571	SD8581	SD8591	SD8403	SD8423	These devices store the amount of data already received.	R

Precautions

Do not change the value with program or engineering tool.

Communication parameter display

The communication parameter content set in the communication settings (☞ Page 420 Communication Settings) is stored when the power is turned OFF → ON or reset. The stored values are the same as the serial communication property values. (☞ Page 447 Serial communication settings)

R: Read only

FX3 Series compatible		Description	R/W
CH1	CH2		
SD8405	SD8425	Stores the setting of the communication parameter.	R

Precautions

Do not change the value with program or engineering tool.

Receive data sum

These devices store the received sum check value. When the sum check code is set to [Added] in the communication settings (☞ Page 420 Communication Settings), the sum check is executed for the receive data. These devices store the sum added to the received data sent from the counterpart equipment.

R: Read only

FX5 dedicated				FX3 Series compatible		Description	R/W
CH1	CH2	CH3	CH4	CH1	CH2		
SD8563	SD8573	SD8583	SD8593	SD8414	SD8434	These devices store the received sum check value.	R

Precautions

Do not change the value with program or engineering tool.

When the sum check code is set to [Added] in the communication settings, always select [Added] for the terminator.

Receiving result sum

When the sum check code is set to [Added] in the communication settings (☞ Page 420 Communication Settings), the sum check is executed for the receive data. These devices store the sum calculated by the CPU module from the data received from the counterpart equipment.

R: Read only

FX5 dedicated				FX3 Series compatible		Description	R/W
CH1	CH2	CH3	CH4	CH1	CH2		
SD8564	SD8574	SD8584	SD8594	SD8415	SD8435	These devices store the sum check value calculated using the received data.	R

Precautions

Do not change the value with program or engineering tool.

When the sum check code is set to [Added] in the communication settings, always select [Added] for the terminator.

Sending sum

When the sum check code is set to [Added] in the communication settings (☞ Page 420 Communication Settings), the sum check is executed for the send data. These devices store the sum calculated by the CPU module from the send data.

R: Read only

FX5 dedicated				FX3 Series compatible		Description	R/W
CH1	CH2	CH3	CH4	CH1	CH2		
SD8565	SD8575	SD8585	SD8595	SD8416	SD8436	These devices store the sum check value added to the send data.	R

Precautions

Do not change the value with program or engineering tool.

When the sum check code is set to [Added] in the communication settings, always select [Added] for the terminator.

Time-out time

These devices set the evaluation time for error when receiving of data is interrupted. (Unit: ms)

R/W: Read/write

FX5 dedicated				Description	R/W
CH1	CH2	CH3	CH4		
SD8621	SD8631	SD8641	SD8651	1 to 32767 (ms) 10 ms when set to a value outside the range.	R/W

Precautions

Do not change the time-out time setting while the RS2 instruction is being driven. To change the value, do so before the RS2 instruction is driven (while off). The change is enabled when the RS2 instruction is drive.

8-bit processing mode

These devices store the value for the processing mode set in the communication settings (☞ Page 420 Communication Settings).

R/W: Read/write

FX5 dedicated				Description	R/W
CH1	CH2	CH3	CH4		
SD8622	SD8632	SD8642	SD8652	0: 16-bit processing mode 1: 8-bit processing mode	R/W

Precautions

Do not change the 8-bit processing mode while the RS2 instruction is being driven. To change the value, do so before the RS2 instruction is driven (while off). The change is enabled when the RS2 instruction is drive.

Stores the setting value set in the communication settings when the power is turned OFF → ON, STOP → RUN, or reset.

Header

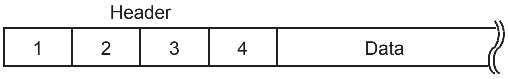
These devices store the contents of headers 1 to 4 set in the communication settings (☞ Page 420 Communication Settings). When the headers are set to [Added] in the communication settings, the headers are added to the send and receive data. Up to four headers can be added in each channel.

R/W: Read/write

Header	FX5 dedicated				R/W
	First	Second	Third	Fourth	
CH1	SD8623 (Low-order byte)	SD8623 (High-order byte)	SD8624 (Low-order byte)	SD8624 (High-order byte)	R/W
CH2	SD8633 (Low-order byte)	SD8633 (High-order byte)	SD8634 (Low-order byte)	SD8634 (High-order byte)	
CH3	SD8643 (Low-order byte)	SD8643 (High-order byte)	SD8644 (Low-order byte)	SD8644 (High-order byte)	
CH4	SD8653 (Low-order byte)	SD8653 (High-order byte)	SD8654 (Low-order byte)	SD8654 (High-order byte)	

When data is sent, the data set in the headers is added at the head of the specified send data.

When data is received, data receiving begins when the data set in the headers is received.



Precautions

Even if the headers are set to [Added] in the communication settings, if the first header value is "00H", the parameter will be [Not Added] state. The area before "00H" (in 1-byte units) is used to set the headers.

Do not change the headers while the RS2 instruction is being driven. To change the value, do so before the RS2 instruction is driven (while off). The change is enabled when the RS2 instruction is drive.

Stores the setting value set in the communication settings when the power is turned OFF → ON, STOP → RUN, or reset.

Terminator

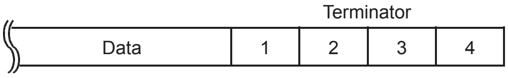
These devices store the contents of terminators 1 to 4 set in the communication settings (☞ Page 420 Communication Settings). When the terminators are set to [Added] in the communication settings, the terminators are added to the send and receive data. Up to four terminators can be added in each channel.

R/W: Read/write

Terminator	FX5 dedicated				R/W
	First	Second	Third	Fourth	
CH1	SD8625 (Low-order byte)	SD8625 (High-order byte)	SD8626 (Low-order byte)	SD8626 (High-order byte)	R/W
CH2	SD8635 (Low-order byte)	SD8635 (High-order byte)	SD8636 (Low-order byte)	SD8636 (High-order byte)	
CH3	SD8645 (Low-order byte)	SD8645 (High-order byte)	SD8646 (Low-order byte)	SD8646 (High-order byte)	
CH4	SD8655 (Low-order byte)	SD8655 (High-order byte)	SD8656 (Low-order byte)	SD8656 (High-order byte)	

When data is sent, the data set in the terminators is added at the end of the specified send data, and data send.

When data is received, receiving is completed when the data set in the terminators is received.



Precautions

Even if the terminators are set to [Added] in the communication settings, if the first terminator value is "00H", the parameter will be [Not Added] state. The area before "00H" (in 1-byte units) is used to set the terminators.

Do not change the terminators while the RS2 instruction is being driven. To change the value, do so before the RS2 instruction is driven (while off). The change is enabled when the RS2 instruction is drive.

Stores the setting value set in the communication settings when the power is turned OFF → ON, STOP → RUN, or reset.

21 PREDEFINED PROTOCOL SUPPORT FUNCTION

This chapter describes predefined protocol support function (serial communication protocol support function).

For details on the predefined protocol support function (built-in Ethernet protocol support), refer to the following.

☞ Page 100 PREDEFINED PROTOCOL SUPPORT FUNCTION

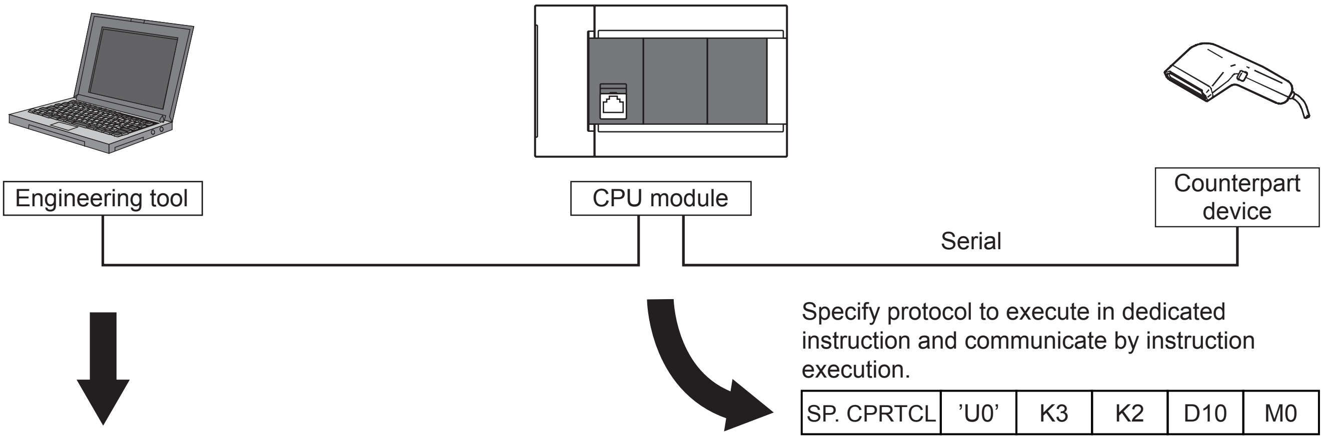
21.1 Function Summary

With predefined protocol support function, data can be sent and received between the counterpart device and CPU module according to the protocol of the counterpart device.

Predefined protocol support function uses the S(P).CPRTCL instruction.

The S(P).CPRTCL instruction can simultaneously communicate on maximum 2 channels by specifying the channels.

- Up to eight protocols can be executed consecutively.
- The protocol can use the general-purpose protocol library, which is pre-registered in the serial communication protocol support tool. Additional protocols can be registered. (1 send packet, 16 receive packets in one protocol)
- Data transfer is enabled when the connected equipment supports non-protocol serial communication.
- The overall distance is 1200 m maximum. (Only when configured by FX5-485ADP)



Specify protocol to execute in dedicated instruction and communicate by instruction execution.

SP. CPRTCL	'U0'	K3	K2	D10	M0

Set the necessary protocol, and write in protocol setting data to the CPU module.

Protocol No.	Manufacturer	Model	Protocol Name	Communication Type	→Send ←Receive	Packet Name	Packet Setting
1	MITSUBISHI ELECTRIC	FREQROL Series	H7B:RD Operation Mode	Send&Receive			
					→	H7B:RD Operation Mode	Variable Set
					←(1)	NOR:RD Data(4 Digits Data)	Variable Set
					←(2)	ERR:NAK Response	Variable Set
2	MITSUBISHI ELECTRIC	FREQROL Series	HFB:WR Operation Mode	Send&Receive			
					→	HFB:WR Operation Mode	Variable Set
					←(1)	ACK:ACK Response	Variable Set
					←(2)	ERR:NAK Response	Variable Set
3	MITSUBISHI ELECTRIC	FREQROL Series	H6F:RD Out Frequency/Speed	Send&Receive			

⋮

21.2 Procedure Before Operation

The flow chart below shows the Predefined Protocol Support Function setting and sequence program creation procedure up until communication with a counterpart device:

1. Check communication specifications

For communication specifications, and predefined protocol specifications, refer to ☞ Page 456 Specifications.

2. System configuration and selection

For system configuration, and select communication equipment, refer to ☞ Page 453 System Configuration.

3. Wiring

For selection of cables and connection equipment, and wiring example, refer to non-protocol communication (☞ Page 415 Wiring).

4. Communication settings

For communication settings of communication device, refer to ☞ Page 457 Communication Settings.

5. Protocol settings

For each protocol settings, refer to ☞ Page 459 Protocol Setting, Page 465 Packet Setting.

6. Program creation

For detailed explanation of related devices, and basic program, refer to ☞ Page 481 Programming.

21.3 System Configuration

This section outlines the system configuration required to use predefined protocol support function.

Restriction

In the predefined protocol support function, only two channels are available to use for one CPU module.

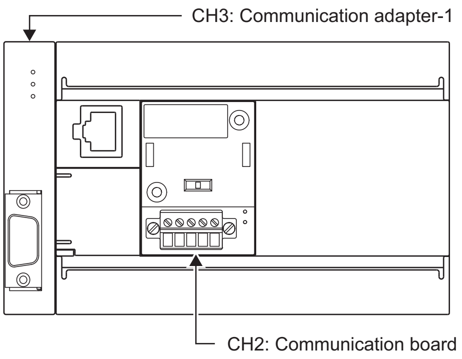
FX5S CPU module

The communication protocol support function can be used in the FX5S CPU module by using the communication board and communication adapter.

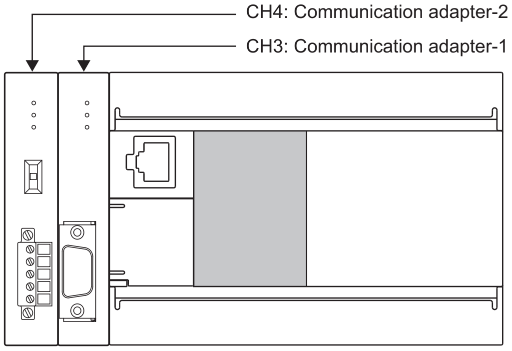
Communication channel assignments are fixed regardless of the system configuration.

The combinations available for the system configurations are shown below.

[Configuration example 1]



[Configuration example 2]



Item		Serial port	Important points in selection	Overall distance
Communication board	FX5-485-BD	CH2	Because the board can be mounted on top of the CPU module, there is no change in the installation space to be required.	50 m or less
	FX5-232-BD			15 m or less
Communication adapter	FX5-485ADP	CH3, CH4[*1]	Mount the communication adapter to the left of the CPU module.	1200 m or less
	FX5-232ADP			15 m or less

[*1] The adapters are assigned to CH3 and CH4 in the order, from the closest one to the CPU module.

Precautions

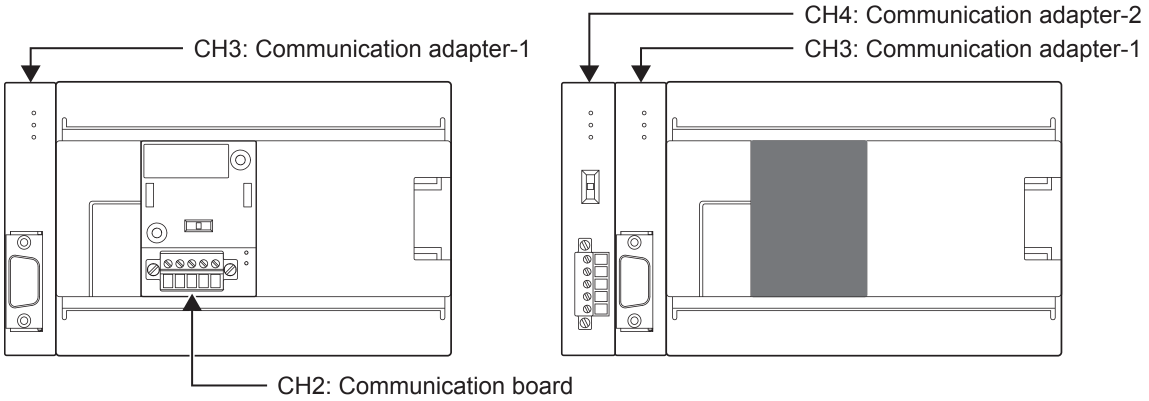
Total number of communication boards and communication adapters that can be connected is two.

FX5UJ CPU module

The communication protocol support function can be used in the FX5UJ CPU module by using the communication board and communication adapter.

Communication channel assignments are fixed regardless of the system configuration.

The combinations which can be configured are shown below.



[Configuration example 1]

[Configuration example 2]

Item		Serial port	Important points in selection	Overall distance
Communication board	FX5-485-BD	CH2	Mounted on top of the CPU module, there is no change in the installation space requirements.	50 m or less
	FX5-232-BD			15 m or less
Communication adapter	FX5-485ADP	CH3, CH4[1]	Mount the communication adapter to the left of the CPU module.	1200 m or less
	FX5-232ADP			15 m or less

[1] The adapters are assigned to CH3 and CH4 in the order, from the closest one to the CPU module.

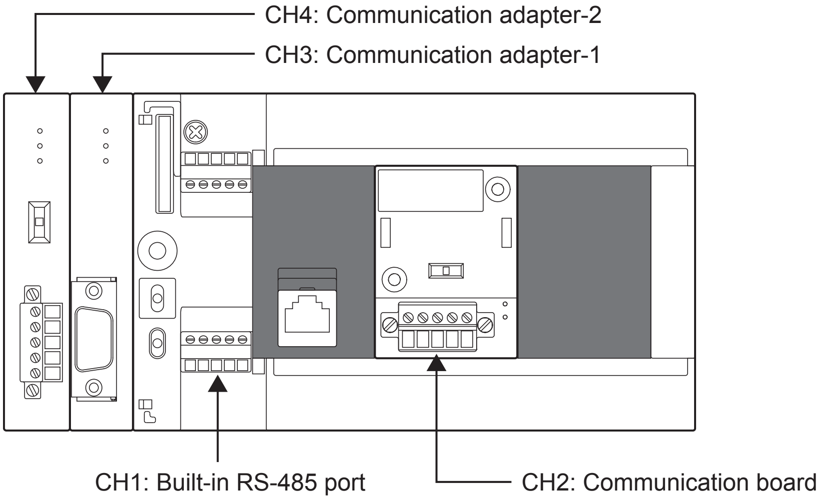
Precautions

Total number of communication boards and communication adapters that can be connected is two.

FX5U CPU module

The communication protocol support function can be used in the FX5U CPU module by using the built-in RS-485 port, communication board, and communication adapter.

Communication channel assignments are fixed regardless of the system configuration.



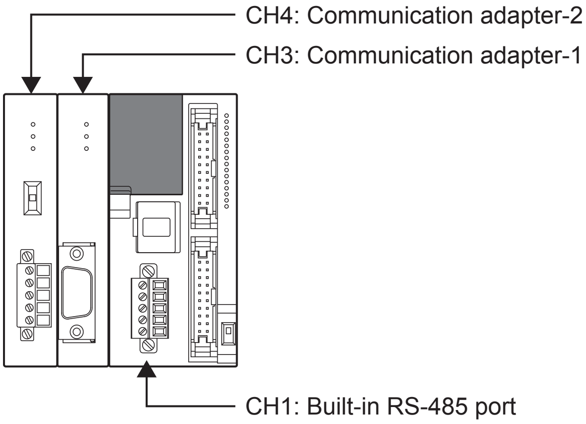
Item		Serial port	Important points for selection	Overall distance
Built-in RS-485 port		CH1	Since it is built-into the CPU module, there is no need to add equipment.	50 m or less
Communication board	FX5-485-BD	CH2	Mounted on top of the CPU module, there is no change in the installation space requirements.	50 m or less
	FX5-232-BD			15 m or less
Communication adapter	FX5-485ADP	CH3, CH4[1]	Mount the communication adapter to the left of the CPU module.	1200 m or less
	FX5-232ADP			15 m or less

*1 The adapters are assigned to CH3 and CH4 in the order, from the closest one to the CPU module.

FX5UC CPU module

The communication protocol support function can be used in the FX5UC CPU module by using the built-in RS-485 port, communication board, and communication adapter.

Communication channel assignments are fixed regardless of the system configuration.



Item		Serial port	Important points for selection	Overall distance
Built-in RS-485 port		CH1	Since it is built-into the CPU module, there is no need to add equipment.	50 m or less
Communication adapter	FX5-485ADP	CH3, CH4[1]	Mount the communication adapter to the left of the CPU module.	1200 m or less
	FX5-232ADP			15 m or less

*1 The adapters are assigned to CH3 and CH4 in the order, from the closest one to the CPU module.

21.4 Specifications

This section describes the communication specifications and performance of predefined protocol support function.

Communication specifications

Items		Specifications	
Transmission standard		RS-485 and RS-422 standard	RS-232C standard
Maximum overall distance		When using FX5-485ADP: 1200m or less When using built-in RS-485 port or FX5-485-BD: 50m or less	15m or less
Protocol type		Predefined protocol support	
Control procedure		—	
Communication method		Half-duplex, bi-directional communication	
Baud rate		9600/19200/38400/57600/115200(bps)	
Character format	Start bit	1 bit	
	Data length	7 bits/8 bits	
	Parity bit	None, odd or even	
	Stop bit	1 bit/2 bits	
Header		Can be set from packet setting.	
Terminator		Can be set from packet setting.	
Control line		—	
Sum check		Can be set from packet setting.	

Predefined protocol specifications

Items	Specifications	Items
Setting method	• Selected from the predefined protocol library offered • Selected from the user protocol library • Creating	By editing the selected protocol, a new protocol can be created.
Number of protocols	Maximum of 64	—
Number of packets	Maximum of 128	One send packet and 16 receive packets can be registered in one protocol.
Number of elements	Maximum of 32	—
Packet data area size	Maximum of 6144 byte	—
Communication type	Send only/receive only/send & receive	☞ Page 457 Communication type
Execution procedure	Protocols can be executed with a dedicated instruction (S(P).CPRTCL).	Up to eight protocols can be executed consecutively for one instruction execution.
Receive wait time	0 to 3000000ms	When set to 0ms, timeout does not occur.
Number of send retries	0 to 10 times	—
Send retry interval	0 to 300000ms	—
Standby time	0 to 300000ms	—
Monitoring time	0 to 300000ms	When set to 0ms, timeout does not occur.
Communication data code	Binary/ASCII	—
The maximum data length that can be sent/received at one time.	2048 byte	—

Communication type

With the predefined protocol support function, communications with counterpart devices are performed using the following procedures (communication types).

Communication type name	Processing
Send Only	Sends a send packet once. One send packet is required.
Receive Only	Receives a packet if it matches any of up to 16 defined receive packets. One or more receive packets are required.
Send & Receive	Sends a send packet, and then receives a packet if it matches any of up to 16 defined receive packets. One send packet and one or more receive packets are required.

For the operation of each communication type, refer to ☞ Page 878 Operation image of each communication type.

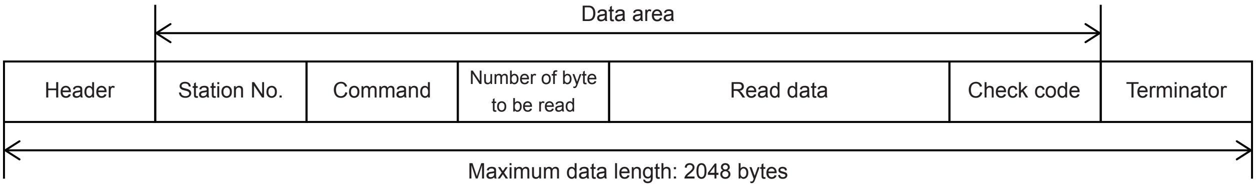
Packet

A send packet to the counterpart devices and a receive packet from the counterpart devices at the time of process execution are registered in a protocol.

For details on packet elements, refer to ☞ Page 465 Packet Setting.

Ex.

Example of a packet configuration



21.5 Communication Settings

For the FX5 Series communication settings of this function, parameters are set using GX Works3. For details about GX Works3, refer to 📖GX Works3 Operating Manual.

Setting of parameter differs according to the module used. The procedure for each module is as follows.

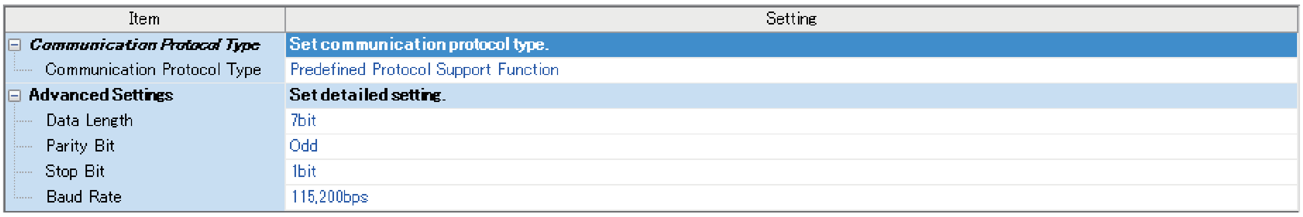
Built-in RS-485 port (CH1)

Navigation Window ⇨ Parameter ⇨ FX5UCPU⇨ Module Parameter ⇨ 485 Serial Port

Window

The following screen will be displayed if [Predefined Protocol Support Function] is set for the communication protocol type.

■Basic Settings



Item	Setting
⊟ *Communication Protocol Type*	Set communication protocol type.
Communication Protocol Type	Predefined Protocol Support Function
⊟ **Advanced Settings**	**Set detailed setting.**
Data Length	7bit
Parity Bit	Odd
Stop Bit	1bit
Baud Rate	115,200bps

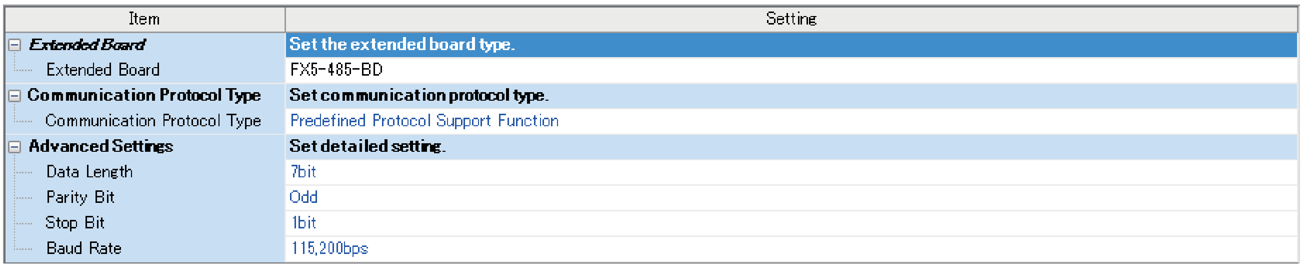
Communication board (CH2)

Navigation Window ⇨ Parameter ⇨ Model name ⇨ Module Parameter ⇨ Extended Board

Window

The following screen will be displayed if [FX5-232-BD] or [FX5-485-BD] is set for the extended board and [Predefined Protocol Support Function] is set for the communication protocol type.

■Basic Settings



Item	Setting
⊟ *Extended Board*	Set the extended board type.
Extended Board	FX5-485-BD
⊟ Communication Protocol Type	Set communication protocol type.
Communication Protocol Type	Predefined Protocol Support Function
⊟ Advanced Settings	Set detailed setting.
Data Length	7bit
Parity Bit	Odd
Stop Bit	1bit
Baud Rate	115,200bps

Communication adapter (CH3/CH4)

When an expansion adapter is used, add expansion adapter to Module Information.

Navigation window ⇨ Parameter ⇨ Module Information ⇨ Right-click ⇨ Add New Module

After adding the expansion adapter, make settings on the screen displayed from the following operation.

Navigation window ⇨ Parameter ⇨ Module Information ⇨ ADP1 to ADP6 (Communication adapter) ⇨ Module Parameter

Window

The setting screen is the same as for the built-in RS-485 port (CH1).

Parameter setting details

Set the following items for the serial ports that use non-protocol communication. However, only two channels can be set for the predefined protocol support function.

Items			Setting value
Basic Settings	Extended Board[1]		When using this function, select [FX5-232-BD] or [FX5-485-BD].
	Protocol type		When using this function, select [Predefined Protocol Support Function].
	Advanced Settings	Data Length	7bit/8bit
		Parity Bit	None/Odd/Even
		Stop Bit	1bit/2bit
		Baud Rate	9600bps/19200bps/38400bps/57600bps/115200bps

[1] Only for the communication board (CH2).

Setting is not required (fixed value) for the board below.

Item	Descriptions
Start bit	1 bit
Header[2]	Not added
Terminator[2]	Not added
Control mode	None
Sum check code[2]	Not added
Control procedure	None

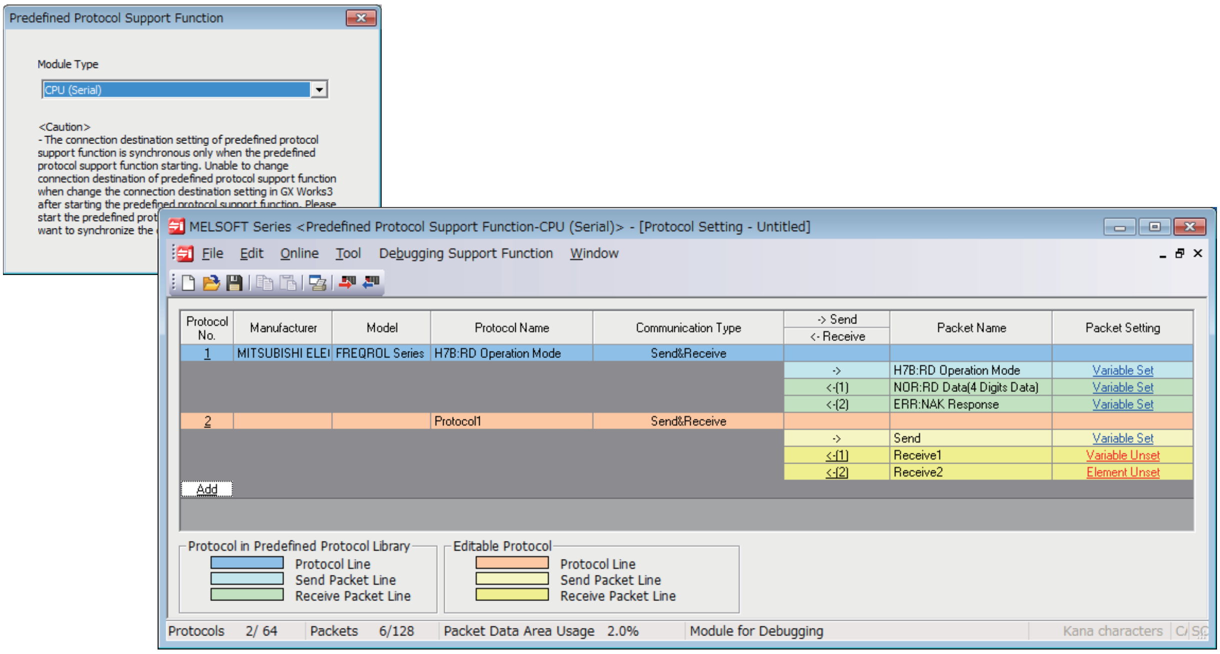
[2] This can be set from packet setting.

21.6 Protocol Setting

Protocols can be selected and edited from the libraries that are pre-registered in GX Works3 or registered by the user.
Execute the protocol setting from a protocol setting window displayed by the following operation in GX Works3.

Tool⇨Predefined Protocol Support Function⇨CPU(Serial)⇨New

Window



Module Type: CPU (Serial)

<Caution>
- The connection destination setting of predefined protocol support function is synchronous only when the predefined protocol support function starting. Unable to change connection destination of predefined protocol support function when change the connection destination setting in GX Works3 after starting the predefined protocol support function. Please start the predefined prot
want to synchronize the

Protocol No.	Manufacturer	Model	Protocol Name	Communication Type	-> Send <- Receive	Packet Name	Packet Setting
1	MITSUBISHI ELE	FREQROL Series	H7B:RD Operation Mode	Send&Receive			
					->	H7B:RD Operation Mode	Variable Set
					<-(1)	NOR:RD Data(4 Digits Data)	Variable Set
					<-(2)	ERR:NAK Response	Variable Set
2			Protocol1	Send&Receive			
					->	Send	Variable Set
					<-(1)	Receive1	Variable Unset
					<-(2)	Receive2	Element Unset

Protocol in Predefined Protocol Library
- Protocol Line
- Send Packet Line
- Receive Packet Line

Editable Protocol
- Protocol Line
- Send Packet Line
- Receive Packet Line

Protocols 2/ 64 | Packets 6/128 | Packet Data Area Usage 2.0% | Module for Debugging | Kana characters | C/ SC

The fields in the protocol setting window are as follows.

Item	Description
Protocol No.	Displays the protocol number to be used for a predefined protocol dedicated instruction.
Manufacturer	Displays the manufacturer's name of the device to which a protocol to be set is applied.
Model	Displays the model name of the device to which a protocol to be set is applied.
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 receive packets, it is received. • Send & receive: After sending one send packet, if there is a matching packet within up to 16 registered receive packets, it is received.
→Send ←Receive	Displays the packet direction. • Send: → • Receive: ←(n) n: Receive packet number (1 to 16)
Packet Name	Displays the packet name.
Packet Setting	Displays the existence or non-existence of variables in an element, and set or unset of the variables. With 'Variable Unset', 'Element Unset', or 'Element error', the settings cannot be written to a CPU module or SD memory card. • 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 • Element Unset: When there are no elements in an editable protocol • Element Error: When elements do not satisfy requirements

The procedure to set a protocol on the protocol setting screen is as follows.

1. Adding protocols

Select or create a protocol from libraries. (☞ Page 460 Adding protocols)

2. Configuring detailed setting of protocols

Set the information and operation of protocols. (☞ Page 461 Configuring detailed setting of protocols)

3. Setting packets

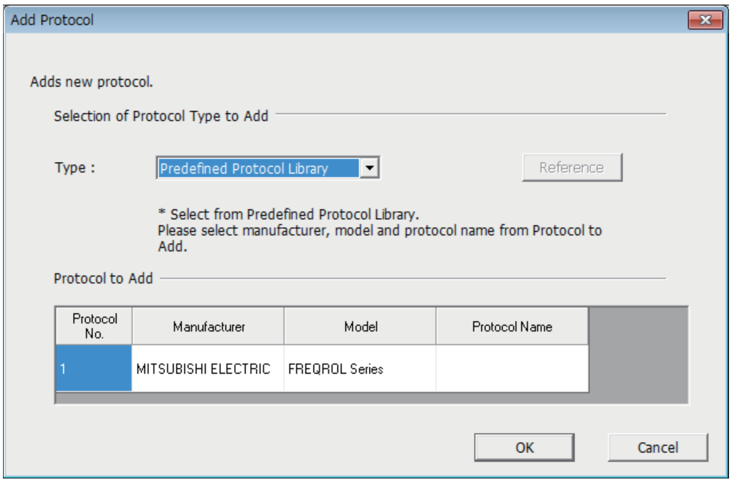
Set the packets of protocols. (☞ Page 465 Packet Setting)

Adding protocols

Add or create a protocol from the predefined protocol library or user protocol library. Add a protocol on the screen displayed by the following operation.

Protocol setting screen ⇨ Edit ⇨ Add Protocol

Window



Protocol No.	Manufacturer	Model	Protocol Name
1	MITSUBISHI ELECTRIC	FREQROL Series	

The setting items of the above screen are as follows.

○: Required, ×: Not required

Item	Description	Setting requirement			Setting range
		Predefined Protocol Library	User Protocol Library	Add New	
Type[*1]	Select a protocol type.	—	—	—	• Predefined Protocol Library • User Protocol Library • Add New
Protocol No.	Set the number of the protocol to be added.	○	○	○	1 to 64
Manufacturer	Set the manufacturer's name of the counterpart device registered in the library.	○	×	×	Manufacturer's name registered in the library.
Model	Set the model of the counterpart device registered in the library of the selected manufacturer.	○	×	×	Model registered in the library of the selected manufacturer.
Protocol Name	Set the protocol name of the counterpart device registered in the library of the selected model.	○	×	×	Protocol name registered with the library of the selected model.

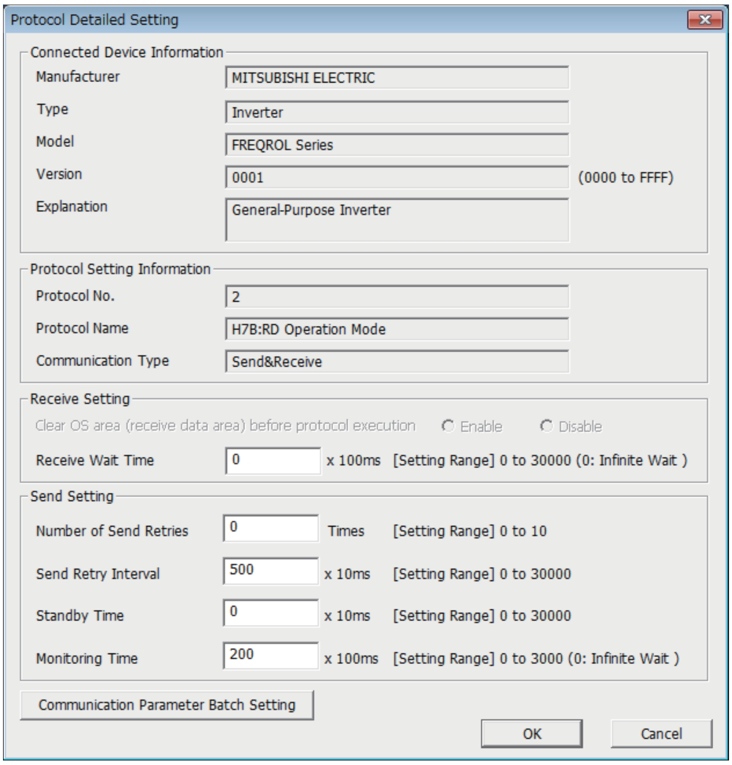
[*1] Predefined Protocol Library: The predefined protocol library offered by GX Works3
User Protocol Library: The protocol library which a user registered

Configuring detailed setting of protocols

Set the send/reception parameters of the added protocol. Set the detailed setting of the protocol on the screen displayed by the following operation.

Protocol setting screen ⇨ Select a row of any protocol on the protocol setting screen ⇨ Edit ⇨Protocol Detailed Setting

Window



Protocol Detailed Setting

Connected Device Information

Manufacturer: MITSUBISHI ELECTRIC

Type: Inverter

Model: FREQROL Series

Version: 0001 (0000 to FFFF)

Explanation: General-Purpose Inverter

Protocol Setting Information

Protocol No.: 2

Protocol Name: H7B:RD Operation Mode

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: 0 Times [Setting Range] 0 to 10

Send Retry Interval: 500 x 10ms [Setting Range] 0 to 30000

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

Monitoring Time: 200 x 100ms [Setting Range] 0 to 3000 (0: Infinite Wait)

The setting items of the above screen are as follows.

○: Required, ×: Not required

Item	Description	Setting requirement			Setting range
		Predefined Protocol Library	User Protocol Library	Add New	
■Connected Device Information					
Manufacturer	Displays the connected device information.	×[1]	×[1]	○	—
Type		×[1]	×[1]	○	—
Model		×[1]	×[1]	○	—
Version		×[1]	×[1]	○	0000H to FFFFH
Explanation		×[1]	×[1]	○	—
■Protocol Setting Information					
Protocol No.	Displays the protocol number of the selected protocol.	—	—	—	—
Protocol Name	Displays the protocol name of the protocol.	×[1]	×[1]	○	Arbitrarily (32 characters)
Communication Type	Displays the communication type of the protocol.	×[1]	×[1]	○	• Send Only • Receive Only • Send & Receive
■Receive Setting (The setting is required when reception is selected for the communication type.)					
Receive Wait Time	Set waiting time after the waiting for reception status. If no matching packet is received within the set time, error occurs.	○	○	○	0 to 30000 (unit: 100ms)[2]
■Send Setting (The setting is required when send is selected for the communication type.)					

Item	Description	Setting requirement			Setting range
		Predefined Protocol Library	User Protocol Library	Add New	
Number of Send Retries	Set the number of times the CPU module retries to send when the sending from the CPU module has not been completed within the set time of monitoring time. Error occurs if sending has not been completed despite the specified number of times of sending retries.	○	○	○	0 to 10
Send Retry Interval	Set the interval between the failure of sending from the CPU module and the retry when the sending from the CPU module has not been completed within the set time of monitoring time.	○	○	○	0 to 30000 (unit: 10ms)
Standby Time	Set standby time between when a protocol set to the CPU module turns to the execution status and when it actually sends the data.	○	○	○	0 to 30000 (unit: 10ms)
Monitoring Time	Set the waiting time between the send start and completion. An error occurs when the send is not completed within the specified time. However, if the send is retried, the data is resent for the number of send retries.	○	○	○	0 to 30000 (unit: 100ms)[2]

[1] These items can be set when the project is changed to an editable project on the protocol setting screen.

[2] When 0 is set, infinite wait is set.

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.

Operations for protocol setting data

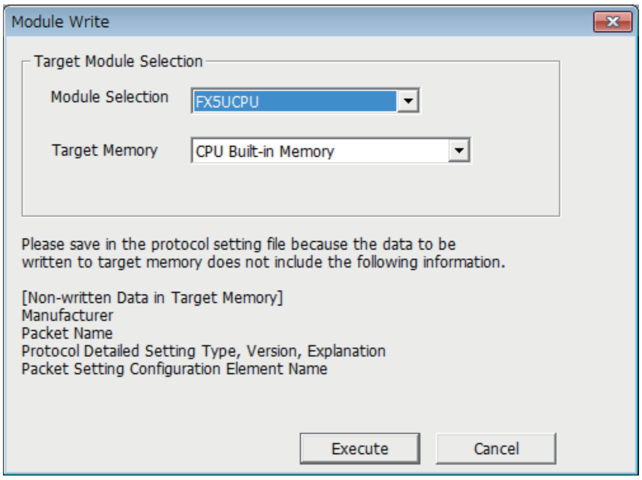
The protocol setting data can be written to the CPU built-in memory or an SD memory card. The written protocol data in the memory can be read and verified. The following describes the operations for the protocol setting data.

Writing protocol setting data

Write the protocol setting data on the screen displayed by the following operation.

Online ⇨ Module Write

Window


Module Write

Target Module Selection

Module Selection: FX5UCPU

Target Memory: CPU Built-in 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

The procedure to write the data is as follows.

1. Select a CPU module to which the protocol setting data is written from Module Selection. Only the CPU module specified in specify connection destination connection of GX Works3 can be specified.
2. Select the memory to which the protocol setting data is to be written from Target Memory.
3. Click [Execute] to execute writing.

Precautions

The written protocol setting data is reflected when the power is turned on or the CPU module is reset.

Point

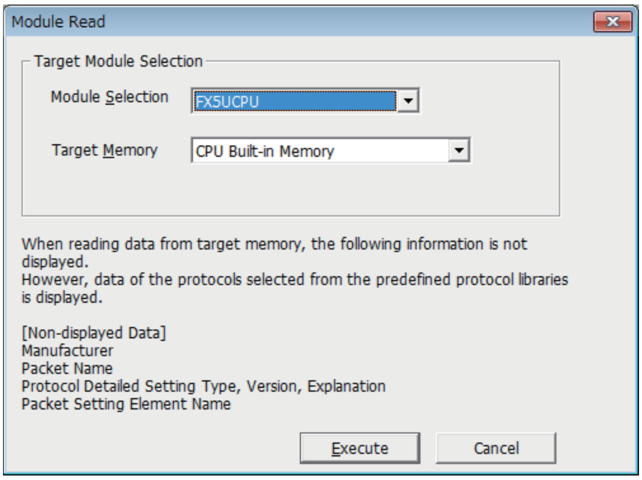
- The protocol setting data is written to module extended parameter (UEX3FF01.PPR).
- When using protocol setting data written to SD memory card, set the protocol setting data transferred to the CPU built-in memory by boot operation. For details, refer to MELSEC iQ-F FX5 User's Manual (Application).

Reading protocol setting data

Read the protocol setting data on the screen displayed by the following operation.

Online ⇨ Module Read

Window



Module Read

Target Module Selection

Module Selection: FX5UCPU

Target Memory: CPU Built-in Memory

When reading data from target memory, the following information is not displayed.
However, data of the protocols selected from the predefined protocol libraries is displayed.

[Non-displayed Data]
Manufacturer
Packet Name
Protocol Detailed Setting Type, Version, Explanation
Packet Setting Element Name

The procedure to read the data is as follows.

1. Select a CPU module from which the protocol setting data is read from Module Selection. Only the CPU module specified in specify connection destination connection of GX Works3 can be specified.
2. Select the memory to which the protocol setting data is to be read from Target Memory.
3. Click [Execute] to execute reading.

Point

The following data is not displayed even when reading from CPU module because it will not be written as a protocol setting data. However, the protocol selected from the predefined library is displayed.

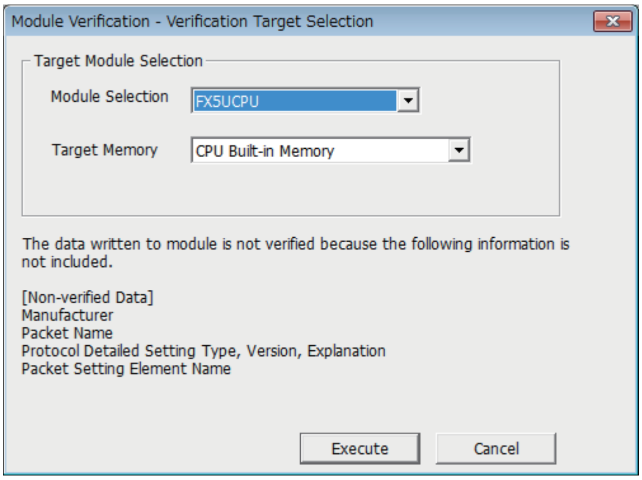
- Manufacturer
- Packet name
- Type, Version, Explanation in the protocol detailed setting
- Element name in the packet setting

Verification protocol setting data

Verify the protocol setting data that is currently set with the protocol setting data that is written to the memory on the screen displayed by the following operation.

Online ⇨ Module Verification

Window



Module Verification - Verification Target Selection

Target Module Selection

Module Selection: FX5UCPU

Target Memory: CPU Built-in Memory

The data written to module is not verified because the following information is not included.

[Non-verified Data]
Manufacturer
Packet Name
Protocol Detailed Setting Type, Version, Explanation
Packet Setting Element Name

The procedure to verify the data is as follows.

1. Select the CPU module of the verification target to which the protocol setting data is written from Module Selection. Only the CPU module specified in specify connection destination connection] of GX Works3 can be specified.
2. Select the memory in which the protocol setting data is written from Target Memory.
3. Click [Execute] to execute verification and display the result.

21.7 Packet Setting

The send packet to the counterpart device and the receive packet from the counterpart device of when process is executed are registered in a protocol. The packet which are set with the predefined protocol support function is created with some elements. Up to 32 elements can be set in a packet.

Ex.

Example of a packet configuration

■Example of format of packet from counterpart device

Header 02H	Station No. “00” to ”99”	Command “WT”	Read data “0000” to ”9999”	Check code	Terminator 03H

■Packet elements setting with the predefined protocol function for the format shown above

Data area

Header	Non-conversion variable	Static data	Conversion variable	Check Code	Terminator

Point

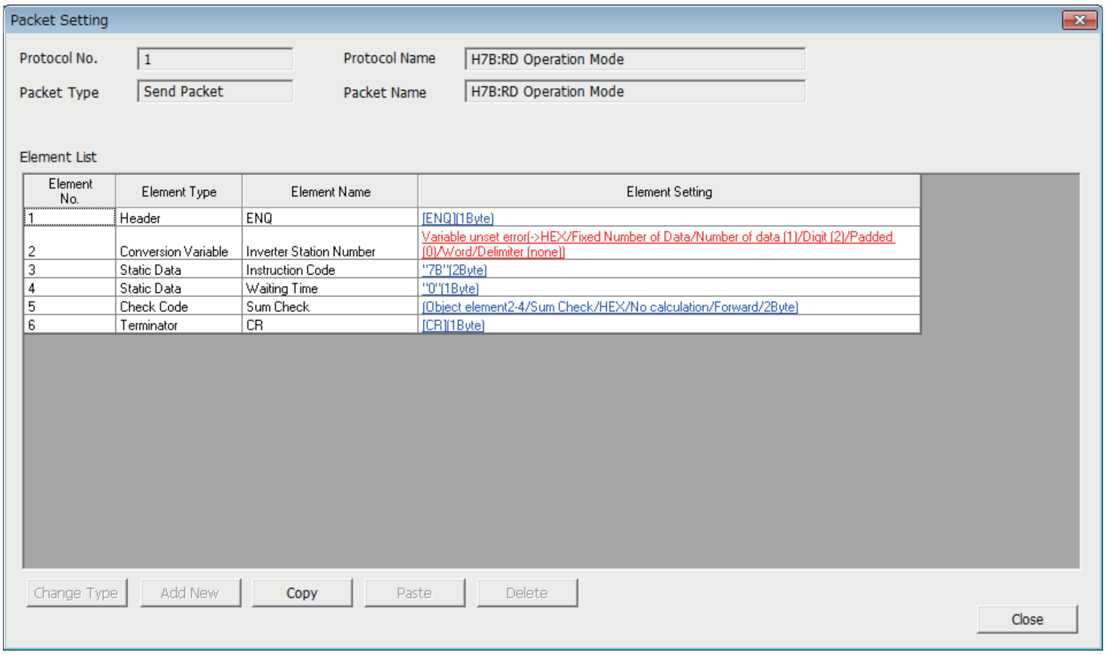
Configure the packet in the order of header, data area[*1], and terminator. Check code and static data can be placed after the terminator.

[*1] Element: Length/static data/non-conversion variable/conversion variable/check code/non-verified reception

Set the packet in the packet setting screen displayed by the following operation of GX Works3.

Protocol setting screen ⇨ Desired packet setting

Window



Protocol No. 1

Packet Type Send Packet

Protocol Name H7B:RD Operation Mode

Packet Name H7B:RD Operation Mode

Element List

Element No.	Element Type	Element Name	Element Setting
1	Header	ENQ	[ENQ](1Byte)
2	Conversion Variable	Inverter Station Number	Variable unset error(->HEX/Fixed Number of Data/Number of data (1)/Digit (2)/Padded (0)/Word/Delimiter (none))
3	Static Data	Instruction Code	"7B"(2Byte)
4	Static Data	Waiting Time	"0"(1Byte)
5	Check Code	Sum Check	(Object element2-4/Sum Check/HEX/No calculation/Forward/2Byte)
6	Terminator	CR	[CR](1Byte)

The setting items of the above screen are as follows.

○: Required, ×: Not required

Item	Description	Setting requirement			Setting range
		Predefined Protocol Library	User Protocol Library	Add New	
Protocol No.	Displays the protocol number of the selected protocol.	—	—	—	—
Protocol Name	The protocol name of the selected protocol.	—	—	—	—

Item	Description	Setting requirement			Setting range
		Predefined Protocol Library	User Protocol Library	Add New	
Protocol Type	The protocol type of the selected protocol. (Send packet or receive packet)	—	—	—	—
Packet Name	The packet name of the selected packet.	×[*2]	○	○	Arbitrarily (32 characters)
■List of elements					
Element No.	These numbers indicate the order of elements. When any element number is changed, the elements are arranged in the numerical order.	×	○	○	1 to 32
Element Type	The type of each element is displayed. For details, refer to the subsequent section of each element. • Header • Terminator ■Data area • Length • Static Data • Non-conversion Variable • Conversion Variable • Check Code • Non-verified reception	×	○	○	—
Element Name	Displays the element name.	×	○	○	Arbitrarily (32 characters)
Element Setting	The setting of each element is displayed. Click an item to display each setting window.	○	○	○	—

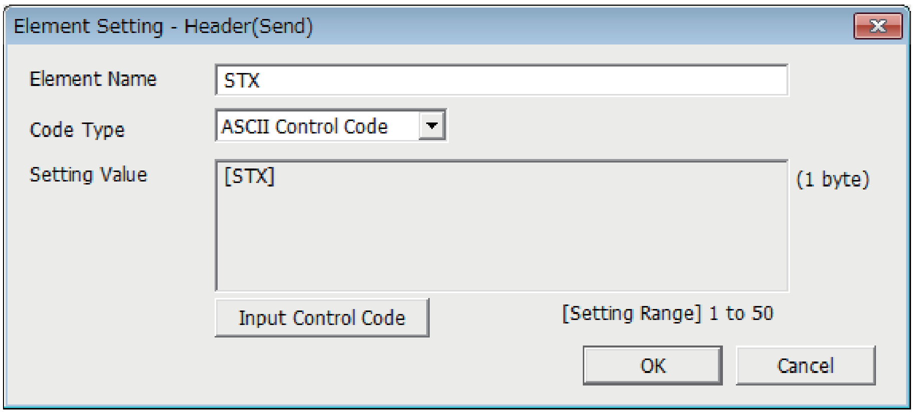
*2 These items can be set when the project is changed to an editable project on the protocol setting screen.

Header

Use this element when a specific code or character string exists at the head of a packet.

- When sending: Sends a specified code and character string.
- When receiving: Verifies a header and received data.

Window



Element Setting - Header(Send)

Element Name: STX

Code Type: ASCII Control Code

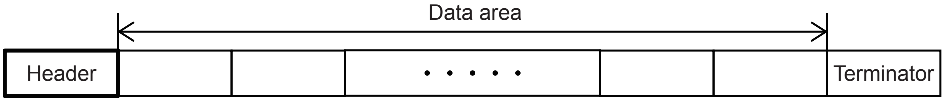
Setting Value: [STX] (1 byte)

[Setting Range] 1 to 50

Item	Descriptions	Remarks
Element Name	Set the name of the element.	—
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. The setting ranges that can be set for each code type are as follows: • 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

Considerations for configuring packet element

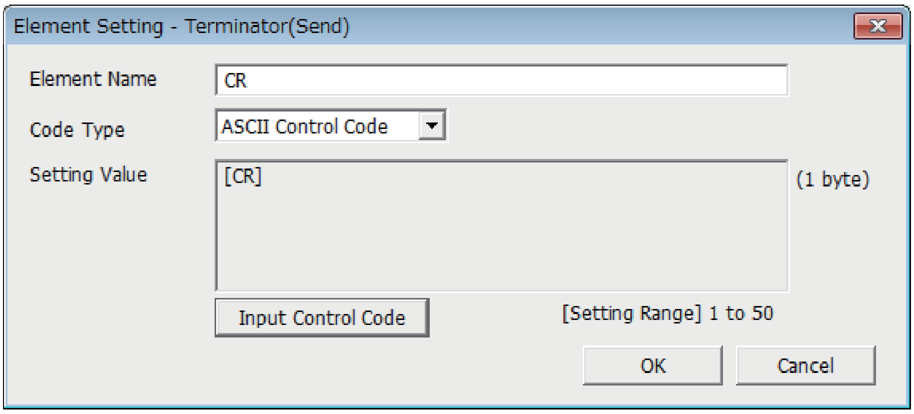
- Only one header can be used in a packet.
- A header can be set only at the head of a packet.



Terminator

Use this element when a code or character string indicating the end of the packet is included.

Window



Element Name: CR

Code Type: ASCII Control Code

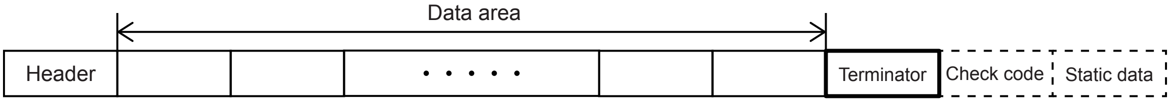
Setting Value: [CR] (1 byte)

[Setting Range] 1 to 50

Item	Description	Remarks
Element Name	Set the name of the element.	—
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. The setting ranges that can be set for each code type are as follows: • 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: ETX • HEX: FFFF

Considerations for configuring packet element

- Only one terminator can be used in a packet.
- A terminator can be set only at the end of a packet. Check code and static data only can be placed after the terminator.



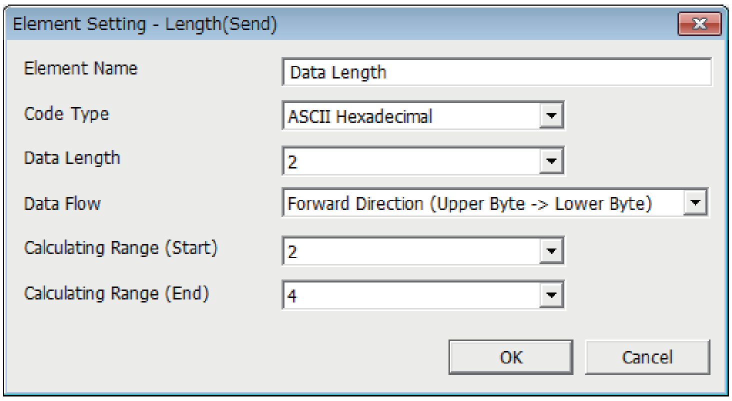
Length

Use this element when an element indicating the data length is included in a 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.

For data examples of length, refer to ☞ Page 883 Data examples of Length.

Window



Element Name: Data Length
Code Type: ASCII Hexadecimal
Data Length: 2
Data Flow: Forward Direction (Upper Byte -> Lower Byte)
Calculating Range (Start): 2
Calculating Range (End): 4

Item		Descriptions			Remarks
Element Name		Set the name of the element.			—
Code Type		Select a data type of the setting value. • ASCII Hexadecimal • ASCII Decimal • HEX			—
Data Length		Select the data length on the line (byte). The range is 1 to 4 bytes.			—
Data Flow		Forward Direction (Upper byte → Lower byte)	When sending	Sends a calculated length from the upper byte.	Data flow setting is invalid when data length is 1 byte.
			When receiving	Receives data from the upper byte.	
		Reverse Direction (Lower byte → Upper byte)	When sending	Sends a calculated length from the lower byte.	
			When receiving	Receives data from the lower byte.	
		Byte Swap (by Word)	When sending	Sends a calculated length swapping the upper byte and lower byte by word.	Data flow setting is invalid when data length is 1 to 3 byte.
			When receiving	Receives data swapping the upper byte and lower byte by word.	
Calculating Range	Start	Select the start packet element numbers for the calculating range. The range is 1 to 32.			Refer to the following. ☞ Page 884
	End	Select the last packet element numbers for the calculating range. The range is 1 to 32.			

Considerations for configuring packet element

- Only one length can be used in a packet.
- To configure the length, one or more other elements are needed.

Caution

- When the number of digits of calculation result is greater than that specified in Data Length, digits greater than the specified digit are omitted (ignored).

Ex.

When 2 bytes is specified in Data Length and the calculation result is 123 bytes, the data length is considered as 23.

- When any Non-conversion variable (Variable length), Conversion variable (Variable number of data), Conversion variable (Fixed number of data/Variable number of digits[*1]), or Non-verified reception (Variable number of characters) is placed behind Length and they are not included in the calculating range of the Length, place any of the following data immediately after the Non-conversion variable (Variable length), Conversion variable (Variable length) or Non-verified reception.

- Static data
- Terminator
- Check code + Static data
- Check code + Terminator

- When Code Type is ASCII Hexadecimal, a corresponding packet is regarded as a mismatch packet if a string other than 0 to 9, A to F, and a to f is received.
- When Code Type is ASCII Decimal, a corresponding packet is regarded as a mismatch packet if a string other than 0 to 9 is received.

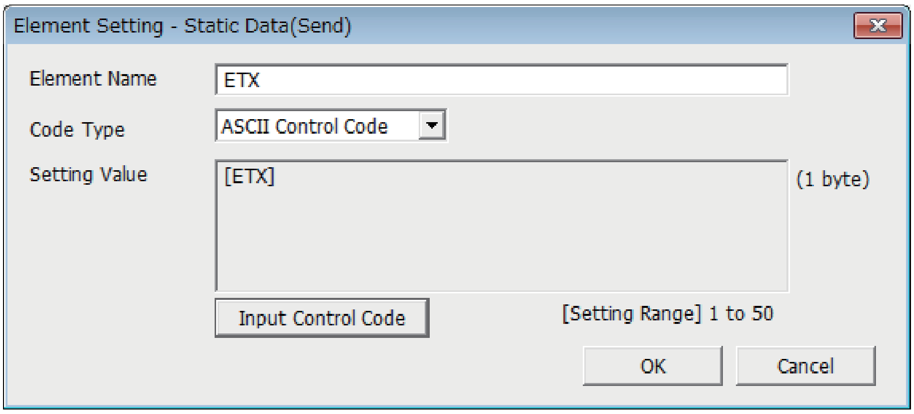
[*1] Excluding the case where number of send data is 1 and delimiter is not no delimiter.

Static Data

Use this element when a specific code or character string such as command exists in a packet.

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

Window



Element Setting - Static Data(Send)

Element Name: ETX

Code Type: ASCII Control Code

Setting Value: [ETX]

(1 byte)

[Setting Range] 1 to 50

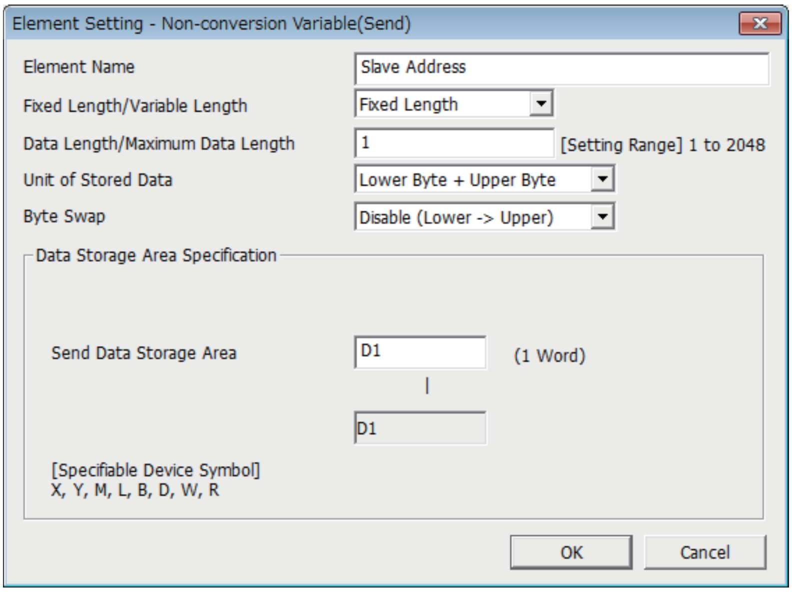
Item	Descriptions	Remarks
Element Name	Set the name of the element.	—
Code Type	Select a data type of the setting value. • ASCII String • ASCII Control Code • HEX	—
Setting Value	Set data within 1 to 50 bytes. The setting ranges that can be set for each code type are as follows: • ASCII String: 20H to 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: US • HEX: FFFF

Point

Multiple static data elements can be placed to desired positions in the data portion.

Non-conversion Variable

Use this element to send the data in the data device as a part of a send packet, or store a part of a receive packet to the device. Use when there is a variable element in the packet format that depends on the system.
For data examples of non-conversion variable, refer to ☞ Page 885 Data examples of Non-conversion Variable.



Element Setting - Non-conversion Variable(Send)

Element Name: Slave Address
Fixed Length/Variable Length: Fixed Length
Data Length/Maximum Data Length: 1 [Setting Range] 1 to 2048
Unit of Stored Data: Lower Byte + Upper Byte
Byte Swap: Disable (Lower -> Upper)

Data Storage Area Specification
Send Data Storage Area: D1 (1 Word)
D1

[Specifiable Device Symbol]
X, Y, M, L, B, D, W, R

Item	Descriptions		
Element Name	Set the name of the element.		
Fixed Length/Variable Length	Fixed Length		Sends and receives the data whose length is fixed.
	Variable Length	When sending	Specifies the data length at the time of the protocol execution and sends data.
		When receiving	Receives data whose data length is variable.
Data Length/Maximum Data Length	Set the length of data to be transmitted and received. (For a variable length, set the maximum data length that can be specified for the data length storage area.) The range is 1 to 2048.		
Unit of Stored Data	Lower Byte + Upper Byte	When sending	Sends each one word (2 bytes) data in the data storage area in the order of the lower byte to the upper byte.
		When receiving	Stores the received data to the data storage area in the order of the lower byte to the upper byte.
	Lower Bytes Only	When sending	Sends only lower byte of data in the data storage area.
		When receiving	Stores the received data in the only lower byte of the data storage area.
Byte Swap	• Disable (Lower → Upper) • Enable (Upper → Lower)	When sending	When Enable is selected, sends data swapping the upper byte and lower byte by word (2 bytes). When Unit of Stored Data is Lower Byte + Upper Byte and Data Length is an odd number of bytes, sends the upper byte at transmission of the last byte. When Unit of Stored Data is Lower Bytes Only and Data Length is an odd number of bytes, sends data without any byte swap at transmission of the last byte.
		When receiving	When Enable is selected, receives data swapping the upper byte and lower byte by word (2 bytes). When Unit of Stored Data is Lower Byte + Upper Byte and Data Length is an odd number of bytes, stores the last byte to the upper byte. When Unit of Stored Data is Lower Bytes Only and Data Length is an odd number of bytes, stores the last byte without any byte swap.
Data Storage Area Specification	Specify a start device*1 to store variable value.	When sending	Specify a head device of the area where send data is stored.
		When receiving*2	Specify a head device of the area where receive data is stored.

*1 Refer to the following.
☞ Page 472 Settable devices

*2 For the variable data length, the data storage area is decided by specifying the data length storage area. (☞ Page 471 When Fixed Length/Variable Length is Variable Length)

Configuration of the data storage area

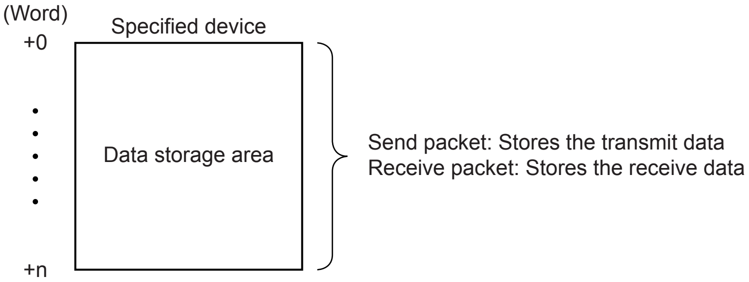
■When Fixed Length/Variable Length is Fixed Length

An area starting from the device number which is specified on the Element Setting screen is considered as the data storage area. The data storage area to be occupied varies depending on the setting of 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, twice the size of the data length is occupied.



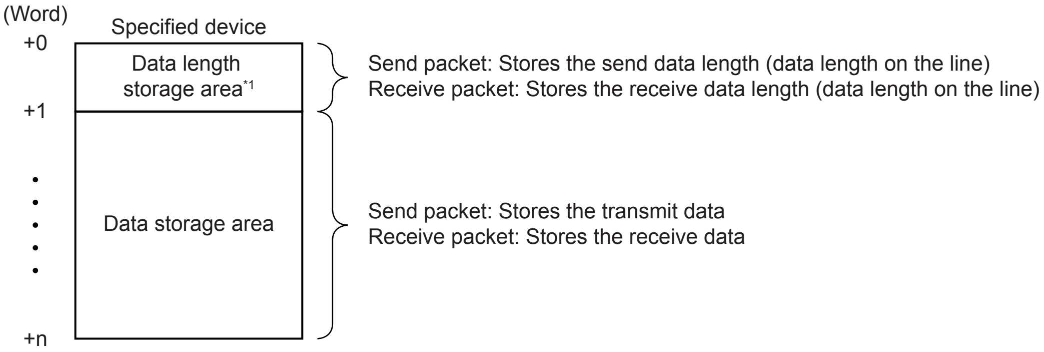
■When Fixed Length/Variable Length is Variable Length

An area starting from the device number which is specified on the Element Setting screen +1 is considered as the data storage area. The data storage area to be occupied varies depending on the setting of 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, twice the size of the data length + one word (length for the data length storage area) are occupied.



*1 The unit of data length is byte. Stores the data length on the line.

■Settable devices

Settable devices to data storage area is follows.

Device		Setting range			Remarks
		FX5S CPU module	FX5UJ CPU module	FX5U/FX5UC CPU module	
Input	X	0 to 1023	0 to 1023	0 to 1023	• For FX5UJ CPU module The assignment cannot be changed. • For FX5S/FX5U/FX5UC CPU module When the assignment has been changed, the devices including the one having the maximum device number after the change can be accessed.
Output	Y				
Internal relay	M	0 to 32767	0 to 7679	0 to 32767	
Latch relay	L				
Link relay	B		0 to 2047		
File register	R		0 to 32767		
Link register	W		0 to 1023		
Data register	D	0 to 7999	0 to 7999	0 to 7999	

Considerations for configuring packet element

Multiple Non-conversion variable (Fixed length) elements can be placed in a packet, and multiple Non-conversion variable (Variable length) elements can also be placed in a send packet. However, only one Non-conversion variable (Variable length) can be placed in a receive packet, and one of the following requirements need to be met.

- Place any of the following items immediately after the non-conversion variable.

- Static data
- Terminator
- Check code + Static data
- Check code + Terminator

- Place length before a non-conversion variable and include the non-conversion variable in the calculating range.

In addition, two or more of the following four elements cannot be placed in the same packet.

- Variable number of data [Conversion variable]
- Fixed number of data and Variable number of digits [Conversion variable] (Excluding the case when the number of data is '1' with delimiter.)
- Variable length [Non-conversion variable]
- Variable number of characters [Non-verified reception]

Caution

- When receiving variable length data whose length exceeds the Maximum data length, CPU module stores data as long as the maximum data length and omits the rest. (The protocol completes successfully.)
- Out of packet data received from counterpart devices, the data corresponding to variable needs to be distinguishable from a terminator or a static data immediately after non-conversion variable. The reception processing may not be performed normally if it cannot distinguished.

Ex.

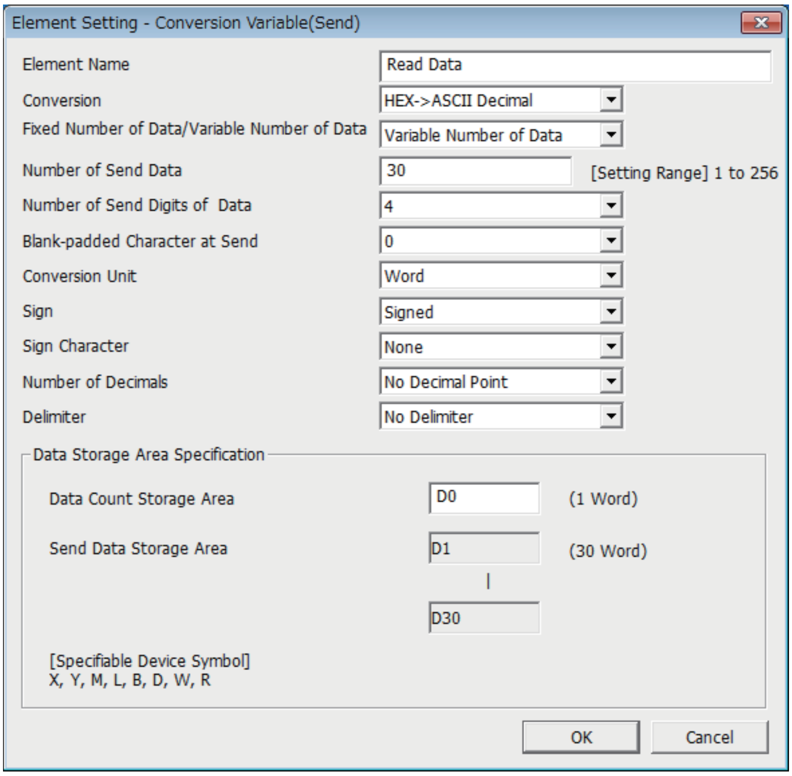
When value of a terminator or value of a static data following a non-conversion variable is used in a non-conversion variable, CPU module recognizes it as data of a terminator or a static data following a non-conversion variable and performs the verification / reception processing.

Conversion Variable

This element converts the numerical data in the value of device to an ASCII string and sends it, or converts the received data (ASCII string) to numerical data and stores it to the device. When a variable element that is dependent on the system exists during packet formatting, use this function.

For data examples of conversion variable, refer to ☞ Page 886 Data examples of Conversion Variable.

Window



Element Setting - Conversion Variable(Send)

Element Name	Read Data	
Conversion	HEX->ASCII Decimal	
Fixed Number of Data/Variable Number of Data	Variable Number of Data	
Number of Send Data	30	[Setting Range] 1 to 256
Number of Send Digits of Data	4	
Blank-padded Character at Send	0	
Conversion Unit	Word	
Sign	Signed	
Sign Character	None	
Number of Decimals	No Decimal Point	
Delimiter	No Delimiter	

Data Storage Area Specification

Data Count Storage Area	D0	(1 Word)
Send Data Storage Area	D1 | D30	(30 Word)

[Specifiable Device Symbol]
X, Y, M, L, B, D, W, R

Item	Descriptions		
Element Name	Set the name of the element.		
Conversion	When sending	HEX→ASCII Decimal	Converts numeric value stored in the data storage area to ASCII decimal.
		HEX→ASCII Hexadecimal	Converts numeric value stored in the data storage area to ASCII hexadecimal.
	When receiving	ASCII Decimal → HEX	Treats received data as ASCII decimal, converts it to numeric value, and stores it to the data storage area.
		ASCII Hexadecimal → HEX	Treats received data as ASCII hexadecimal, converts it to numeric value, and stores it to the data storage area.
Fixed Number of Data/ Variable Number of Data	Fixed Number of Data		Fixes the number of data to be sent and received.
	Variable Number of Data		• When sending: Specifies the number of data to be sent at the time of the protocol execution and sends the data. • When receiving: Receives data of which the number is variable. For Variable number of digits, delimiters are required.
Number of Send Data	Set the number of data to be transmitted and received. (For Variable Number of Data, set the maximum number of data that can be specified to the data quantity storage area.) The range is 1 to 256.		
Number of Send Digits of Data	1 to 10		Set the number of digits per one transmitted and received data. When the number of send digits of data is less than the specified number of digits of data, upper digits are filled with blank-padded characters.
	Variable Number of Digits		• When sending: Sends only the data portion converted to an ASCII string in variable length. • When receiving: Receives only an ASCII string of the data portion in variable length. When the number of send data/maximum data length is set to 2 or greater, blank-padded characters are required.
Blank-padded Character at Send	• 0 • Space		Select a character used to fill upper digits when the number of digits of data is not Variable Number of Digits and the number of digits of transmitted/received data is less than the specified number of send digits of data.

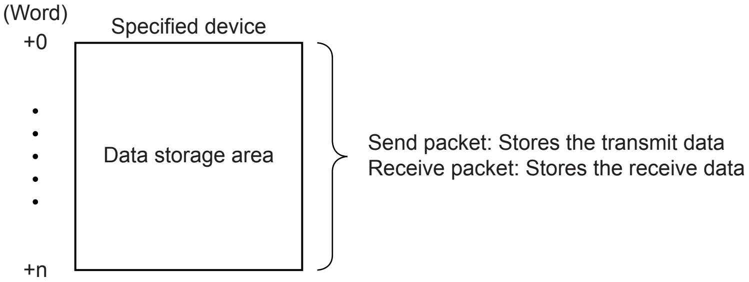
Item	Descriptions	
Conversion Unit	Word	Converts one word as one data in the data storage area.
	Double word	Converts two words as one data in the data storage area.
Sign	• Unsigned • Signed	Select whether to add signs to date in the data storage area. This item can be set when Conversion is HEX → ASCII Decimal or ASCII Decimal → HEX.
Sign Character	• None • + • 0 • Space	Select the sign character for positive value on line. This item can be set when Conversion is HEX → ASCII Decimal or ASCII Decimal → HEX, and Sign is Signed. The sign character for negative value is fixed to "-". For operation of sign character, refer to ☞ Page 887.
Number of Decimals	• No Decimal Point • Variable Point • 1 to 9	Select the decimal point position of data on line. This item can be set when Conversion is HEX → ASCII Decimal or ASCII Decimal → HEX. For operation of number of decimals, refer to ☞ Page 888.
Delimiter	• No Delimiter • Comma • Space	Select the delimiter inserted after one data. A delimiter is not added to the end of data when the number of data is 2 or more. For operation of delimiter, refer to ☞ Page 889.
Data Storage Area Specification	Specify a start device[*1] to store variable value.	

[*1] Refer to the following.
☞ Page 476 Devices that can be specified

Configuration of the data storage area

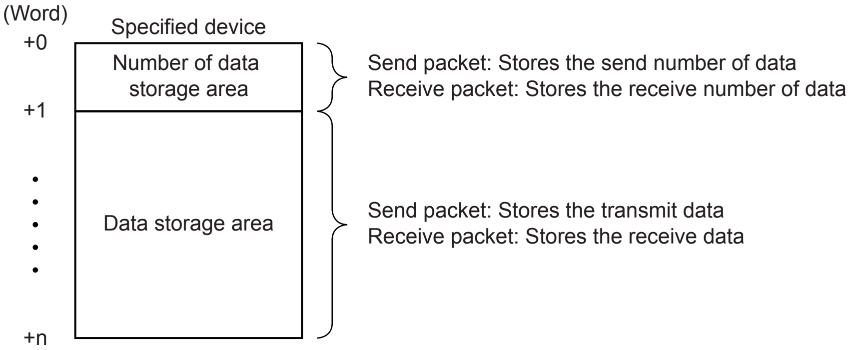
■When Fixed Length/Variable Length is Fixed Length

An area starting from the device number which is specified on the Element Setting screen is considered as the data storage area.



■When Fixed Number of Data/Variable Number of Data is Variable Number of Data

An area starting from the device number which is specified on the element setting screen +1 is considered as the data storage area.



■Occupied size in data storage area

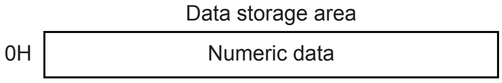
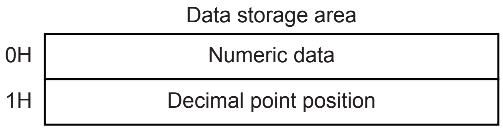
The size occupied in the data storage area varies depending on the settings of Conversion Unit and Number of Decimals.

Setting item		Occupied size in data storage area per one data
Conversion Unit	**Number of Decimals**	
Word	No Decimal point/Fixed point	1 word
	Variable point	2 words
Double Word	No Decimal point/Fixed point	2 words
	Variable point	4 words

■Data storage area configuration per one data

The following shows the data storage area configuration per one data.

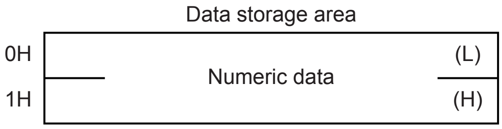
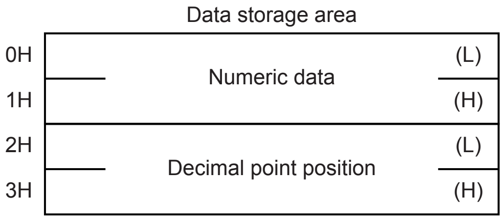
- Case of Conversion Unit is Word

Number of Decimals	
No decimal Point/Fixed point	**Variable point**
	For Variable Point, the decimal point position is set in the data storage area. 

In the decimal point position area, the decimal point position is set as follows.

Send/receive data (number of digits of data is 5)	Numeric data	Decimal point position
12345	12345 (3039H)	1 (1H)
1234.5	12345 (3039H)	10 (0AH)
123.45	12345 (3039H)	100 (64H)
12.345	12345 (3039H)	1000 (3E8H)
1.2345	12345 (3039H)	10000 (2710H)

- Case of Conversion Unit is Double word

Number of Decimals	
No decimal Point/Fixed point	**Variable point**
	For Variable Point, the decimal point position is set in the data storage area. 

In the decimal point position area, the decimal point position is set as follows.

Send/receive data (number of digits of data is 10)	Numeric data	Decimal point position
1234567890	1234567890 (499602D2H)	1 (1H)
123456789.0	1234567890 (499602D2H)	10 (0AH)
12345678.90	1234567890 (499602D2H)	100 (64H)
1234567.890	1234567890 (499602D2H)	1000 (3E8H)
⋮	⋮	⋮
1.234567890	1234567890 (499602D2H)	1000000000 (3B9ACA00H)

■Range of value that can be used in the data storage area

The following table shows the range of value that can be used in the data storage area.

Conversion	Sign	Conversion unit	Range of value
HEX → ASCII decimal ASCII decimal → HEX	Unsigned	Word	0 to 65535 (0H to FFFFH)
		Double word	0 to 4294967295 (0H to FFFFFFFFH)
	Signed	Word	-32768 to 32767 (8000H to FFFFH, 0H to 7FFFH)
		Double word	-2147483648 to 2147483647 (80000000H to FFFFFFFFH, 0H to 7FFFFFFFH)
HEX → ASCII hexadecimal ASCII hexadecimal → HEX	—	Word	0H to FFFFH
		Double word	0H to FFFFFFFFH

■Devices that can be specified

Settable devices to data storage area is follows.

Device		Setting range			Remarks
		FX5S CPU module	FX5UJ CPU module	FX5U/FX5UC CPU module	
Input	X	0 to 1023	0 to 1023	0 to 1023	• For FX5UJ CPU module The assignment cannot be changed. • For FX5S/FX5U/FX5UC CPU module When the assignment has been changed, the devices including the one having the maximum device number after the change can be accessed.
Output	Y				
Internal relay	M	0 to 32767	0 to 7679	0 to 32767	
Latch relay	L				
Link relay	B		0 to 2047		
File register	R		0 to 32767		
Link register	W		0 to 1023		
Data register	D	0 to 7999	0 to 7999	0 to 7999	

Considerations for configuring packet element

To place a Conversion variable in a packet, the following requirements need to be met.

■To place Conversion variable in send packet

Multiple Conversion variable elements can be placed in one packet, and they can be placed in desired positions in the data portion.

■To place Conversion variable in receive packet

Multiple conversion variable elements can be placed in one packet in the cases not corresponding to (1), (2) and (3).

- Variable number of data (Only one Conversion variable can be placed in one packet, and either of the following (1) or (2) needs to be met.)

(1) For discerning the data length of a Conversion variable, any of the following items is placed immediately after the conversion variable.

- Static data
- Terminator
- Check code + Static data
- Check code + Terminator

(2) Length is placed before a Conversion variable. (The Conversion variable needs to be included in the calculating range.)

- Fixed number of data

■Variable number of digits

(3) When the number of data is 2 or more, or the number of data is 1 with no delimiter, only one conversion variable can be placed in a packet and it needs to be placed in the order mentioned in variable number of data.

(4) When a Conversion variable (the number of data is 1, with delimiter, and variable number of digits) and the following four elements are placed in the same packet, these four elements need to be placed behind the Conversion variable (the number of data is 1, with delimiter, and variable number of digits).

- Conversion variable (variable number of data)
- Conversion variable (fixed number of data and variable number of digits) (The case of (3). Excluding the case where Number of Send Data is '1' and Delimiter is not No Delimiter.)
- Non-conversion variable (variable length)
- Non-verified reception (variable number of characters)

Note that two or more of these four elements cannot be placed in the same packet.

■Fixed number of digits (1 to 10)

(5) When a Conversion variable (variable decimal point) and the four elements mentioned in (4) are placed in the same packet, these elements need to be placed behind the Conversion variable (variable decimal point).

(6) When a Conversion variable (unsigned) and the four elements mentioned in (4) are placed in the same packet, these elements need to be placed behind the Conversion variable (unsigned).

Precautions

■A string other than '0'-'9', 'A'-'F', or 'a'-'f' is received.

When Conversion is ASCII Hexadecimal→HEX, an error may occur if a string other than '0'-'9', 'A'-'F', or 'a'-'f' is received.

■A string other than '0'-'9' is received.

When Conversion is ASCII Decimal→HEX, an error may occur if a string other than '0'-'9' is received. However, the error does not occur in the following cases.

Item	CPU module operation
Sign/Sign Character	When Signed is selected, a signed character can be received. However, when a signed character which is not at the head of data is received, an error may occur.
Number of Decimals	When any other than [No Decimal Point] is selected, '.' (a period) can be received. However an error may occur when '.' (a period) which is not at the specified number of digits of data is received. An error may also occur when [Variable Point] is selected and '.' (a period) which is at the head or end of a unit of data is received.
Delimiter	When any other than No delimiter is selected, a delimiter can be received. However an error may occur when a delimiter which is not at a boundary of data is received.

■A data of which number of digits exceeds the upper limit is received

When Number of Send Digits of Data is Variable Number of Digits, an error may occur if the number of digits of received data exceeds the upper limit shown below.

Conversion unit	Conversion content	Upper limit of number data to be received
Word	ASCII Decimal → HEX	5 digits
	ASCII Hexadecimal → HEX	4 digits
Double word	ASCII Decimal → HEX	10 digits
	ASCII Hexadecimal → HEX	8 digits

■A data whose decimal point position is greater than the number of digits is specified

When Number of Decimals is Variable Point in a send packet, a decimal point place specification error (7D21H) may occur if the decimal point position is greater than the number of digits.

■A data whose number of digits is '0' is received

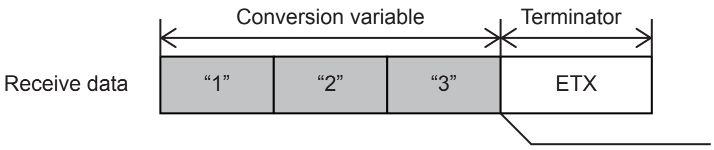
When Number of Send Digits of Data is Variable Number of Digits, an abnormal digit number error (7D19H) may occur if the number of digits of received data is '0'.

■A data whose number of digits are fewer than the digits specified in Number of Send Digits of Data are received

An insufficient digit number error (7D18H) may occur if data whose number of digits are fewer than the digits specified in number of send digits of data are received.

Ex.

When Number of Send Digits of Data is set to 4 digits



An error occurs when conversion variable of the received data ends up with three digits while the setting of Number of Digits of Data is four digits.

■A data whose number exceeds Maximum number of data is received

When receiving data of which the number exceeds the Maximum number of data, CPU module stores data as many as the maximum number of data and omits the rest. (The protocol completes successfully.)

■A data that cannot be distinguished between Terminator or a Static data

In received packet data from counterpart devices, CPU module needs to be able to distinguish data corresponding to a conversion variable from those of a Terminator or a Static data following a Conversion variable. The reception processing may not be performed normally if they cannot be distinguished.

Ex.

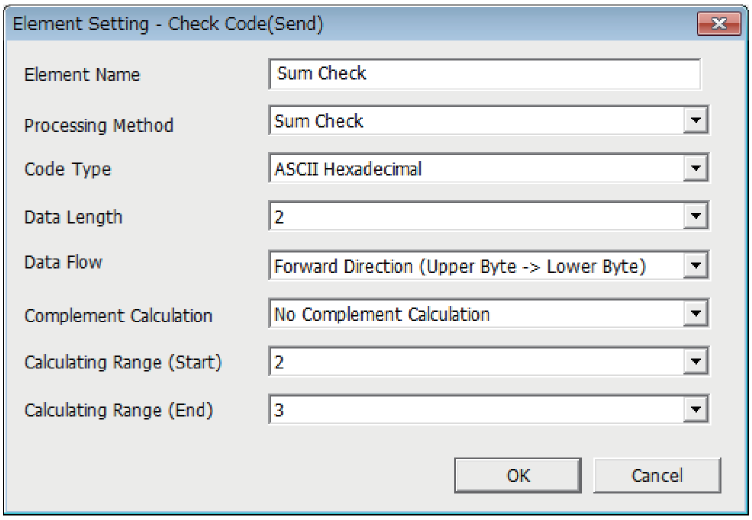
When the value of terminator or a static data following the conversion variable is used as a conversion variable, CPU module recognizes it as a terminator or a fixed data following the conversion variable, verifies, and performs reception processing.

Check Code

Use this element when an element indicating check code data is included in a packet.

CPU module automatically calculates a specified check code at timing of sending/receiving, and adds it to a send packet or detects an error of a receive packet.

Window



Element Setting - Check Code(Send)

Element Name: Sum Check
Processing Method: Sum Check
Code Type: ASCII Hexadecimal
Data Length: 2
Data Flow: Forward Direction (Upper Byte -> Lower Byte)
Complement Calculation: No Complement Calculation
Calculating Range (Start): 2
Calculating Range (End): 3

Item	Description			Remarks
Element Name	Set the name of the element.			—
Processing Method	Select a calculating method. • Horizontal Parity • Sum Check • 16-bit CRC (for MODBUS)			For the calculation procedure, refer to the following. ☞ Page 890 Calculation procedure for horizontal parity ☞ Page 892 Calculation procedure for sum check ☞ Page 894 Calculation procedure for 16-bit CRC (for MODBUS)
Code Type	• ASCII Hexadecimal • ASCII Decimal • HEX	When sending	Select a format in which a calculated check code is sent.	Not settable when Processing Method is 16-bit CRC (for MODBUS).
		When receiving	Select a format in which data are received.	
Data Length	Select the data length on the line (byte). The range is 1 to 4 bytes.			Not settable when Processing Method is 16-bit CRC (for MODBUS).

Item		Description			Remarks
Data Flow		Forward Direction (Upper byte → Lower byte)	When sending	Select a format in which a calculated check code is sent.	Not settable when Processing Method is 16-bit CRC (for MODBUS), or when the Data Length is set to 1 (byte).
			When receiving	Select a format in which data are received.	
		Reverse Direction (Lower byte → Upper byte)	When sending	Sends a calculated check code, from the lower byte.	
			When receiving	Receives data from the lower byte. Valid when the data length is 2 to 4 byte.	
		Byte Swap (by Word)	When sending	Sends a calculated check code by swapping the upper byte and lower byte by word.	Not settable when Processing Method is 16-bit CRC (for MODBUS), or when the Data Length is set to 1 to 3 (byte).
			When receiving	Receives data by swapping the upper byte and lower byte by word, and handles as a check code. Valid when the data length is 4 byte.	
Complement Calculation		Select the complement calculation. • No Complement Calculation • One's Complement • Two's Complement			Not settable when Processing Method is 16-bit CRC (for MODBUS).
Calculating Range	Start	Select the start packet element numbers for the calculating range. The range is 1 to 32.			☞ Page 895 Check code calculation range
	End	Select the last packet element numbers for the calculating range. The range is 1 to 32.			

Considerations for configuring packet element

- Only one check code can be placed in a packet.
- A check code can be placed in any position of the data portion or after a terminator. However, one or more elements are needed before the check code.

Caution

- When Code Type is [ASCII Hexadecimal], an error may occur if a string other than '0' to '9' 'A' to 'F' 'a' to 'f' is received.
- When Code Type is [ASCII Decimal], an error may occur if a string other than '0' to '9' is received.
- When a calculated check code (Sum check/Horizontal parity/16-bit CRC) does not match a received check code, an error may occur.

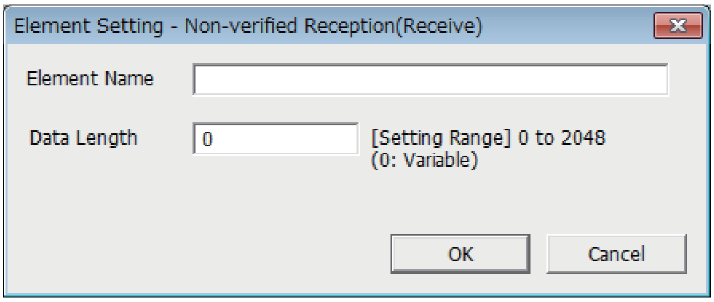
Non-verified Reception

Use this element when received data includes unnecessary data while receiving data.

CPU module skips the specified number of characters if a receive packet includes a non-verified reception.

For data examples of non-verified reception, refer to ☞ Page 896 Data examples of Non-verified reception.

Window



Element Name

Data Length 0 [Setting Range] 0 to 2048
(0: Variable)

Item	Descriptions	
Element Name	Set the name of the element.	
Data Length	0 (variable number of characters)	Set this item when the number of characters not to be verified varies in each communication.
	1 to 2048 (specified number of characters)	Set the number of characters not to be verified.

Considerations for configuring packet element

■When Data Length is '1 to 2048', the following requirements need to be met

Multiple Non-verified receptions can be used in a packet and placed in any position of the data portion.

■When Data Length is '0', the following requirements need to be met

Only one Non-verified reception can be placed in a packet, and one of the following requirements need to be met.

- Place any of the following data immediately after a Non-verified reception.

- Static data
- Terminator
- Check code + Static data
- Check code + Terminator

- Place length before a non-verified reception and include the non-verified reception in the calculating range.

In addition, two or more of the following four elements cannot be placed in the same packet.

- Conversion variable [Variable number of data]
- Fixed number of data and Variable number of digits [Conversion variable] (Excluding the case when the number of data is '1' with delimiter.)
- Non-conversion variable [Variable length]
- Variable number of characters [Non-verified reception]

21.8 Programming

This section explains how to create programs for predefined protocol support function using S(P).CPRTCL instruction and how such programs operate.

For details on related devices, refer to ☞ Page 490 Related Devices.

For communication settings, refer to ☞ Page 457 Communication Settings.

For protocol settings, refer to ☞ Page 459 Protocol Setting.

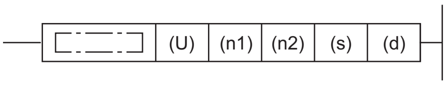
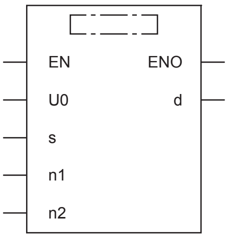
Predefined protocol support instruction

This section explains the function, operation and programming method of the S(P).CPRTCL instruction.

For details on the expression and executable format of the S(P).CPRTCL instruction, refer to 📖MELSEC iQ-F FX5 Programming Manual (Instructions, Standard Functions/Function Blocks).

S(P).CPRTCL

This instruction is used to execute the predefined protocol registered with the engineering tool via RS-485 built in the CPU module, or the communication board and communication adapter installed to the CPU module.

Ladder	ST	FBD/LD
	ENO:=S_CPRTCL (EN, U0, s, n1, n2, d); ENO:=SP_CPRTCL (EN, U0, s, n1, n2, d);	 ("S_CPRTCL", "SP_CPRTCL" enters □.)

Setting data

■Content, range, data types

Operand	Description	Range	Data type	Data type (label)
(U)[*1]	Dummy (Input the character string ['U0'].)	—	Character string	ANYSTRING_SINGLE
(n1)	Communication CH	■FX5S/FX5UJ CPU module K2 to K4 ■FX5U CPU module K1 to K4 ■FX5UC CPU module K1, K3 to K4	16-bit unsigned binary	ANY16_U
(n2)	Number of protocols to be executed continuously	K1 to 8	16-bit unsigned binary	ANY16_U
(s)	Start device containing the control data	Refer to the following. ☞ Page 482	Word	ANY16_ARRAY (Number of elements: 18)
(d)	Head bit device number to output the instruction execution status	—	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)	—	○	—	—	—	—	○	—	—	○	—
(n1)	—	○	—	—	—	—	○	○	—	—	—
(n2)	—	○	—	—	—	—	○	○	—	—	—
(s)	—	○	—	—	—	—	○	—	—	—	—
(d)	○	○[1]	—	—	—	—	—	—	—	—	—

[1] T, ST, C cannot be used.

■Control data

Device	Item	Description	Setting range	Set by[1]
(s)	Result of executed protocols	The result of executed S(P).CPRTCL instruction is stored. When two or more protocols are executed, the execution result of the protocol executed last is stored. • 0: Normal completion • Other than 0: Error completion (error code)	—	System
(s)+1	Resulting number of executed protocols	The number of protocols executed by the S(P).CPRTCL instruction is stored. The execution of the protocol where an error has occurred is also counted in the number of executions. If the setting of setting data or control data contains an error, "0" is stored.	0, 1 to 8	System
(s)+2	Execution protocol number 1	Specify the number of the protocol to be executed first.	1 to 64	User
(s)+3	Execution protocol number 2	Specify the number of the protocol to be executed second.	0, 1 to 64	User
(s)+4	Execution protocol number 3	Specify the number of the protocol to be executed third.	0, 1 to 64	User
(s)+5	Execution protocol number 4	Specify the number of the protocol to be executed fourth.	0, 1 to 64	User
(s)+6	Execution protocol number 5	Specify the number of the protocol to be executed fifth.	0, 1 to 64	User
(s)+7	Execution protocol number 6	Specify the number of the protocol to be executed sixth.	0, 1 to 64	User
(s)+8	Execution protocol number 7	Specify the number of the protocol to be executed seventh.	0, 1 to 64	User
(s)+9	Execution protocol number 8	Specify the number of the protocol to be executed eighth.	0, 1 to 64	User
(s)+10	Verification 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 verification 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
(s)+11	Verification 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 verification 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	System
(s)+12	Verification 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 verification 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 isless than 3, "0" is stored.	0, 1 to 16	System
(s)+13	Verification 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 verification 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 isless than 4, "0" is stored.	0, 1 to 16	System
(s)+14	Verification 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 verification 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	System

Device	Item	Description	Setting range	Set by[1]
(s)+15	Verification 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 verification 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 isless than 6, "0" is stored.	0, 1 to 16	System
(s)+16	Verification 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 verification 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	System
(s)+17	Verification 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 verification 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	System

[1] User: Data to be set by users before the execution of the S(P).CPRTCL instruction.
System: Data to which the CPU module sets the execution result of the S(P).CPRTCL instruction.

Processing details

- Protocols registered with the engineering tool are executed. The communication CH specified by (n1) is used and the protocol to be executed follows the control data of the device specified by (s) or later.
- Protocols are executed consecutively for the number of the protocols specified by (n2) (up to eight protocols) in one instruction execution.
- The number of executed protocols is stored in the device specified by (s)+1.
- The execution of the S(P).CPRTCL instruction and whether it has been completed normally or with an error can be checked with the completion device (d) or resulting of executed protocols device (d)+1.

■Completion device (d)

The completion device turns on in END processing of the scan performed upon completion of the S(P).CPRTCL instruction and turns off in the next END processing.

■Completion status indication device (d)+1

The completion device turns on or off depending on the completion status of the S(P).CPRTCL instruction.

When completed normally:

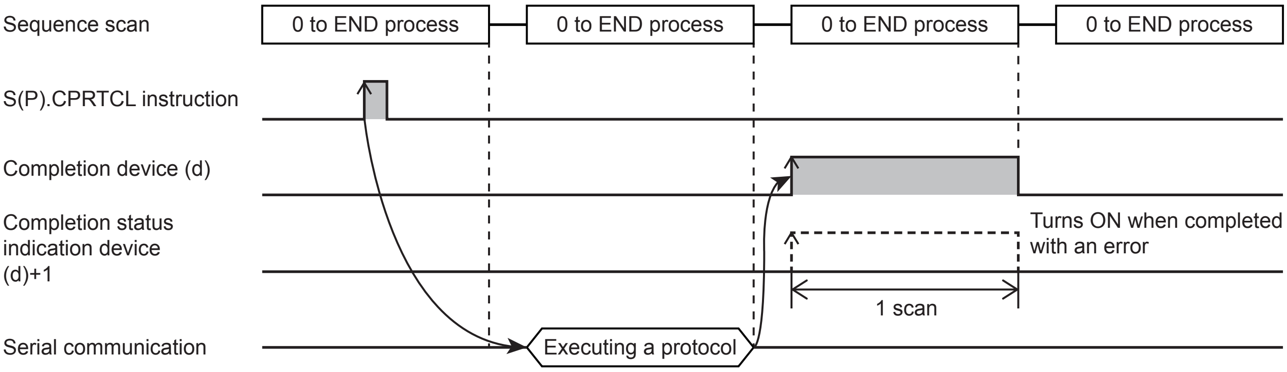
Unchanged from OFF.

When completed with an error:

Turns on in END processing of the scan performed upon completion of the S(P).CPRTCL instruction and turns off in the next END processing. When the processing is completed abnormally, the error code is stored in the result of executed protocols (s).

Instruction execution timing

The following figure shows the S(P).CPRTCL instruction execution timing.



Protocol execution status

The current status of the protocol can be checked with the following devices.

Name	CH1	CH2	CH3	CH4
Protocol execution status	SD9150	SD9170	SD9190	SD9210

The value corresponding to the current protocol execution status is stored in each of the devices above. Any of the values listed in the following table is stored in each of the devices above.

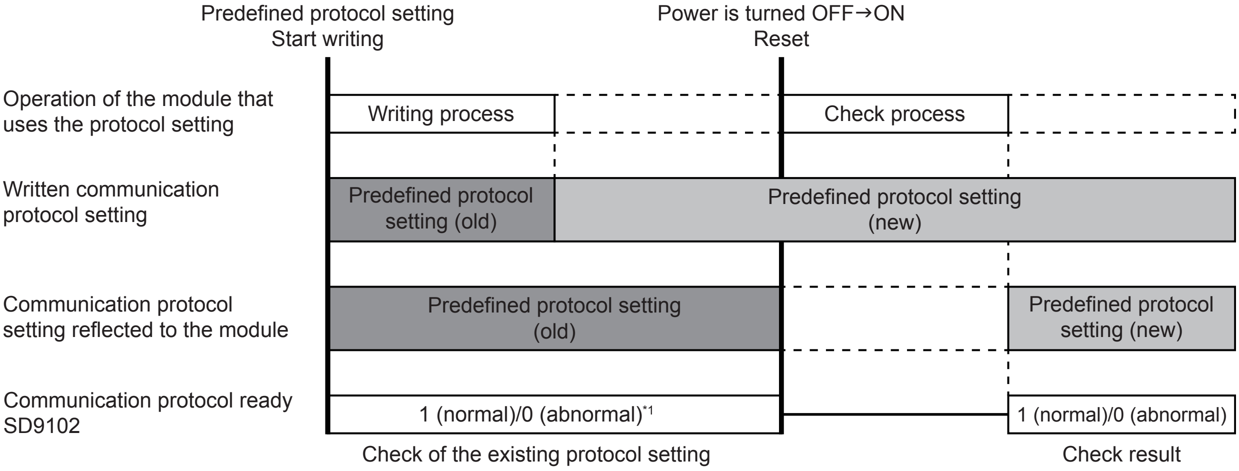
Stored value	Protocol execution status	Description
0	Unexecuted	The protocol has never been executed since the power supply turned on or the reset. Once the protocol has been executed, the status of the protocol does not return to unexecuted until the power supply is turned on or the reset.
1	Waiting for transmission	The status before starting to send a packet to the counterpart device. When the CPU module is preparing or waiting to send a packet, the protocol is in this status.
2	Sending	A packet is being sent to the counterpart device.
3	Waiting for data reception	The CPU module waits to receive data from the counterpart device.
4	Receiving	Data was received from the counterpart device and the receive processing is being executed. Any of the following processing is included in this status. • Data verification processing • General data processing of when a CPU device is specified as a variable
5	Completed	The execution of a protocol has been completed.

Information to judge whether the predefined protocol can be executed or not

Whether the predefined protocol can be executed or not can be judged with the value set in predefined protocol ready (SD9102).

- SD9102 = 0: The protocol cannot be executed because the protocol setting has an error.
- SD9102 = 1: The protocol can be executed because the protocol setting is correct.

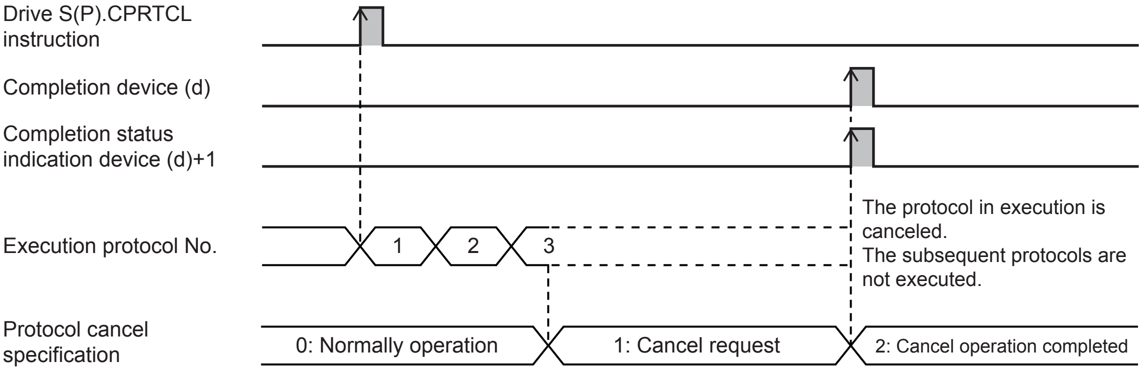
The judgment timing is as follows.



*1 When protocol setting is not written in, it is 0 (error).

Protocol Cancel

Executing protocol can be canceled by the cancel request ("1" is set for the protocol cancel setting value). (☞ Page 496 Protocol cancel specification) Then canceled protocol ends abnormally. When two or more protocols are executed consecutively, executing protocol ends and following protocols are not executed.



The operation differs as follows depending on the protocol communication in execution and the timing of the cancel request (protocol execution status).

The protocol communication in execution.	Cancel timing (Protocol execution status)	Resulting of executed protocols
Send	Executed protocols start - Before data transmission start (Waiting for transmission)	The packet is not sent.
	Data transmission start - Before send completed (Sending)	The packet is not suspended, let the send complete.
Receive	Send completed - Before data reception start (Waiting for data reception)	The packet is not received.
	After data reception start (Receiving)	The packet is suspended, and the packet is discarded.

Precautions

- If a cancel request is issued while no protocol is being executed, the CPU module completes the cancel request without performing any processing.
- The CPU module periodically checks for a cancel request. For this reason, it may take time until cancel processing is performed after a cancel request is issued.

Send/receive data monitoring function

With the send/receive data monitoring function, send/receive data in the communication with a counterpart device can be monitored.

Set the following devices to execute monitoring or specify the storage destination data device for the send/receive data.

CH1	CH2	CH3	CH4	Setting item		Description	Reference
SD9230	SD9240	SD9250	SD9260	Send/receive data monitoring function setting		The monitoring function can be stopped or started. The monitor function status is displayed after monitoring is started.	Page 496
SD9231	SD9241	SD9251	SD9261	Send/receive data monitoring function option setting	Data area full stop specification	Monitoring stops automatically when the number of monitor data size is larger than the specified one.	Page 496
					1-packet stop specification	Monitoring stops automatically after one send packet or receive packet is monitored.	
SD9232	SD9242	SD9252	SD9262	Monitoring data device specification		Specify a type of word device used as the monitor data area.	Page 496
SD9233	SD9243	SD9253	SD9263	Monitoring data start device No. specification		Specify a start device number of word devices used as the monitor data area.	Page 497
SD9234	SD9244	SD9254	SD9264	Monitoring data size specification		Specify the size of word devices used as the monitor data area in word unit.	Page 497

■Monitor start

When 0001H is set for send/receive data monitoring function setting, the device number of monitoring data and the number of monitor data are initialized and monitoring is started. When monitoring is started, monitoring (0002H) is written to send/receive data monitoring function setting.

■Monitoring

After monitoring is started, data is stored in order of receive data, send data, and receive error starting from the monitor data area. The storage timing of the data is as follows.

- Receive data: When receiving data
- Send data: When sending data
- Receive error data: When detecting a receive error

The following shows the operations of when the size of the monitored data exceeds the size of the monitor data area.

- When monitoring continued (bit 0: OFF) is set for data area full stop specification

The old data is overwritten with the new data and the monitoring continues.

- When monitoring stop (bit 0: ON) is set for data area full stop specification

Monitoring stops and monitoring stop (1002H) is written to send/receive data monitoring function settings.

■Monitor stop

Monitoring is stopped by one of the following methods.

- Monitoring is stopped when 0000H is set for send/receive data monitoring function setting.
- When monitoring stop (bit 0: ON) is set for data area full stop specification

Monitoring is stopped when the data is stored for the size of the monitor data area. At this time, monitoring stop (1002H) is written for send/receive data monitoring function setting.

- When 1-packet stop specified (bit 1: ON) is set for 1-packet stop specification

Monitoring is stopped when the sending of one packet send data or the receiving of one packet receive data is completed. If the size of one packet exceeds the size of the monitor data area, monitoring is stopped when the data is stored for the size of the monitor data area. At this time, monitoring stop (1002H) is written to send/receive data monitoring function setting. The type of packets for which monitoring starts or stops differs depending on the communication type of protocols.

Communication type	Stop requirement
Send Only	Monitoring stops when one send packet is monitored.[1]
Receive Only	Monitoring stops when one receive packet is monitored.[1]
Send & Receive	Monitoring stops when one send packet and one receive packet are monitored.[1]

[1] If the size of one packet exceeds the size of the monitor data area, monitoring stops when the data for size of the monitor data area is stored.

■Monitor data

The data stored in the monitor data area is as follows.

Ex.

Monitoring data device specification: D device (0)

Monitoring data start device No. specification: 0

Monitoring data size specification: 100

Device	Data	Description	Program	Explanation of data
D0	0032H	Monitor data device No.	Number of the device in which the oldest data is stored.	The oldest data is D50. (0032H=50)
D1	0064H	Number of monitor data	Number of monitor data stored in the monitor data area	Number of monitor data is 100 (0064H=100)
D2	1000H	Monitor data 1	Monitor data area	Send data/00H
D3	1001H	Monitor data 2		Send data/01H
D4	1002H	Monitor data 3		Send data/02H
⋮	⋮	⋮		⋮
D99	1003H	Monitor data 98		Send data/03H
D100	1004H	Monitor data 99		Send data/04H
D101	1005H	Monitor data 100		Send data/05H

Two words from the start device number are used for the device number of monitoring data and the number of monitor data. The devices with the device numbers after the first two words are used for the monitor data area.

- Monitoring data device No.

Among the data stored in the monitor data area, the number of the device in which the oldest data is stored. The monitor data device number is initialized when the send/receive data monitoring starts.

The storage range is from the start of the monitor data area to (the start of the monitor data area + monitor data size -1).

■When monitoring continued (bit 0: OFF) is set for data area full stop specification

The start device numbers of the monitor data area are stored until the size of the data area where the data is stored and set exceeds the size of the monitor data. When the monitor data area is overwritten because the number of monitor data exceeds the monitor data size, the device in which the oldest data is stored is updated.

■When monitoring stop (bit 0: ON) is set for data area full stop specification

The start device number of the monitor data area is stored.

- Number of monitor data

The number of monitor data stored in the monitor data area is stored. The number of monitor data is initialized when the send/receive data monitoring starts.

The storage range is from 0 to the size of the monitor data.

The number of monitor data does not increase any more after the number of the monitor data reaches the size of the monitor data. (Number of monitor data = Size of the monitor data)

- Monitor data

The monitor data is stored in the following configuration in 1-word units.

FE: Framing error, OVE: Overrun error, PE: Parity error

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	Communication type
0	0	0	0	0	0	0	0	Receiving data								When data receiving
0	0	0	1	0	0	0	0	Sending data								When data sending
0	0	1	1	0	0	0	0	0					FE	OVR	PE	When sending error detection

One word consists of the send/receive data (1 byte) for lower 8 bits and the data classification/signal monitor for upper 8 bits.

Upper high rank 4bit (b15 to b12)	Upper low rank 4bit (b11 to b8)	Lower 8bit (b7 to b0)
Data type	**Signal monitor**	**Send/receive data**
0 (0000b): Receive data 1 (0001b): Send data 3 (0011b): Receive error	b11: 0	—

Precautions

When the monitor data areas overlap for each channel, the data of the smaller channel number is overwritten with the data of the larger channel number because data are stored from the smaller channel number.

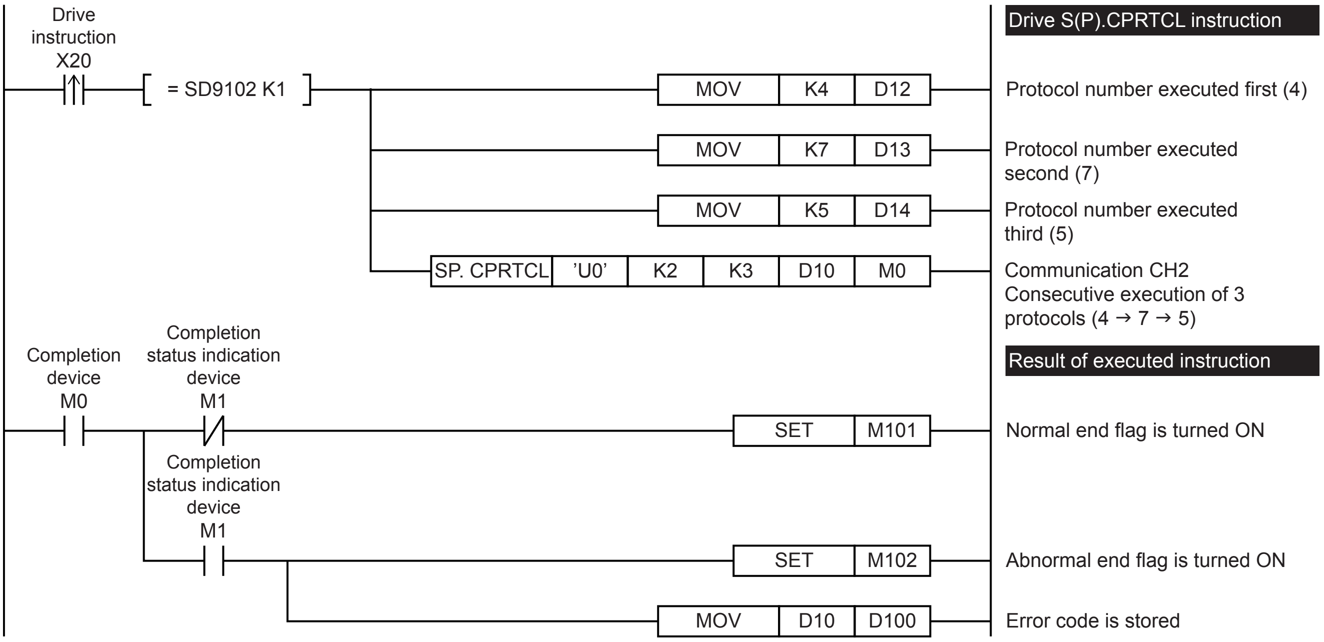
Caution

- If an error occurs in the "n"th protocol while multiple protocols are being executed, the instruction does not execute the "n+1"th protocol and after and is completed with an error.
- If same instructions are executed for the same channel, the subsequent instruction is ignored and is not executed until the preceding instruction is completed.
- If the receive waiting time is set to 0: Infinite wait, the S(P).CPRTCL instruction is not completed until the data specified in the protocol setting is received.

Program example

The following shows an example of a program to execute consecutively the protocols in order of protocol numbers 4, 7, and 5 from the communication board (CH2) with the predefined protocol support function.

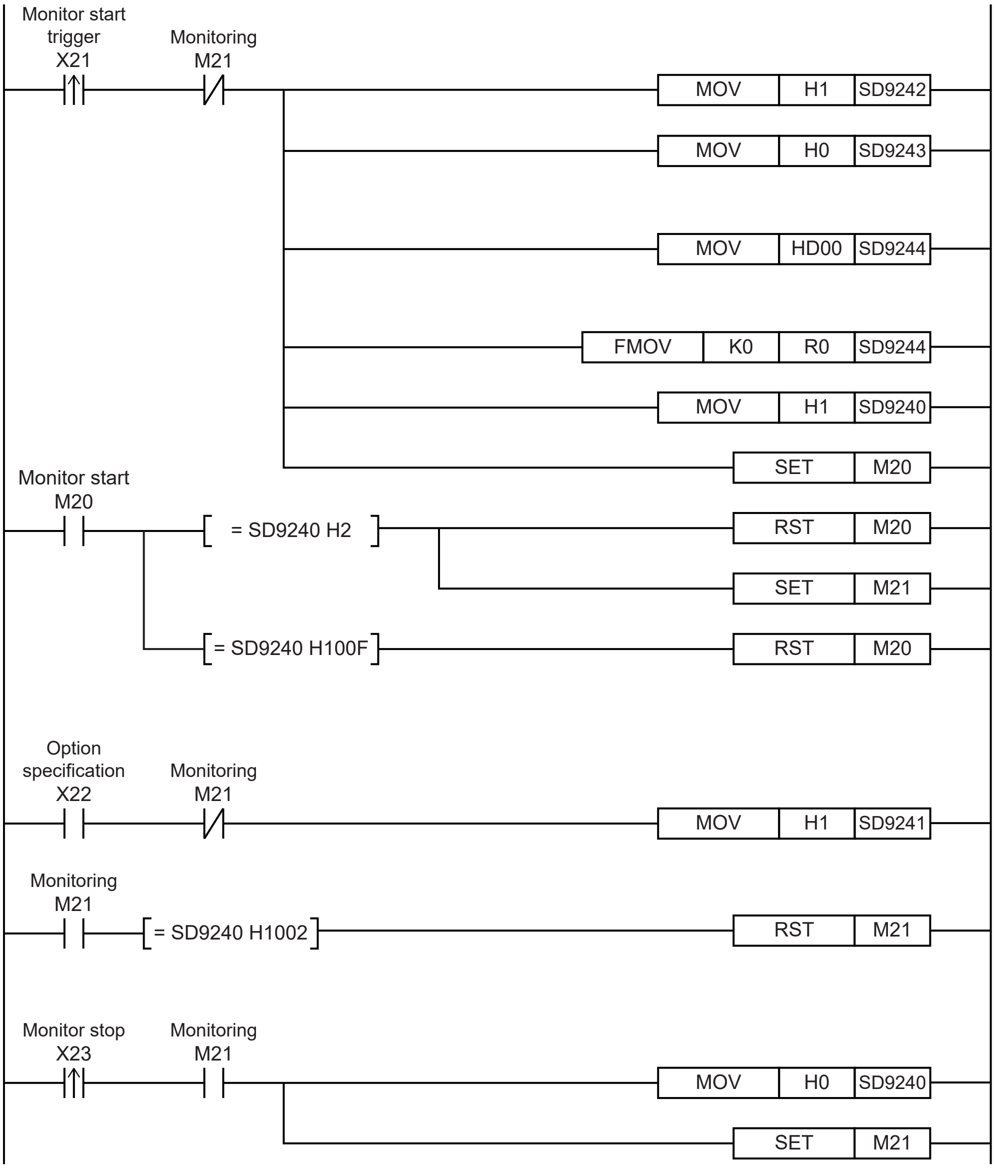
Program example



Device	Description	Device	Description
X20	Trigger of drive S(P).CPRTCL instruction	SD9102	Predefined protocol ready
M0	S(P).CPRTCL instruction completion device	D10	Start device of control data
M1	S(P).CPRTCL instruction completion status indication device	D12	Control data (Protocol number executed first)
M101	Execution normal end flag	D13	Control data (Protocol number executed second)
M102	Abnormal end flag	D14	Control data (Protocol number executed third)

When the send/receive data monitoring function is used, add a program as follows.

Program example



Monitor start

Monitoring data device specification (R device)

Monitoring data start device No. specification (R0)

Monitoring data size specification (3
- Monitoring data device No.: R0
- Number of monitoring data: R1
- Monitoring data 1 to 3328: R2 to R

0 clear monitor data.

Monitor start instruction

Turn on "Monitoring" when the monitoring starts.

Monitoring setting error is detected.
Monitoring is suspended.

Option specification

Data area full stop specification Monitoring stops automatically when the number of monitor data size is larger than the specified one.

Monitor stop instruction

Monitor stop instruction

Device	Description	Device	Description
X21	Monitor start trigger	SD9240	Send/receive data monitoring function setting (CH2)
X22	Option specification trigger	SD9241	Send/receive data monitoring function option setting (CH2) • Data area full stop specification: Monitor stop • Packet stop specification: 1-packet stop specification disabled
X23	Monitor stop trigger		
M20	Monitor start contact		
M21	Monitoring contact	SD9242	Monitoring data device specification (CH2) • Monitoring data device: R device
R0 to R3329	Monitor data area	SD9243	Monitoring data start device No. specification (CH2) • Monitoring data start device No.: 0
		SD9244	Monitoring data size specification (CH2) • Monitoring data size: D00H (= 3328)

21.9 Related Devices

This section describes the special relay/special register functions used in the predefined protocol support function.

Point

Available communication channels vary depending on the CPU module and system configuration.
For communication channels, refer to ☞ Page 453 System Configuration.

List of related devices

Special relays

R: Read only

Device No.				Name	Description	R/W
CH1	CH2	CH3	CH4			
SM8500	SM8510	SM8520	SM8530	Serial communication error	Turns ON when an error occurs in serial communication.	R

Special registers

R: Read only, W: Write only, R/W: Read/write

Device No.				Name	Description	R/W
CH1	CH2	CH3	CH4			
SD8500	SD8510	SD8520	SD8530	Serial communication error code	When a serial communication error occurs, the error code is stored.	R
SD8502	SD8512	SD8522	SD8532	Serial communication settings	Stores the setting of the communication parameter.	R
SD8503	SD8513	SD8523	SD8533	Serial communication operation mode	This device stores the current communication mode being used.	R
SD9102				Predefined protocol ready	The reflected status after the protocol setting data has been written is stored.	R
SD9120				Predefined protocol setting data error information: Protocol No.	When a protocol setting data error was detected, information to identify the error position is stored.	R
SD9121				Predefined protocol setting data error information: Setting type		R
SD9122				Predefined protocol setting data error information: Packet No.		R
SD9123				Predefined protocol setting data error information: Element No.		R
SD9124				Number of registered predefined protocols	The number of registered protocol setting data is stored.	R
SD9132, SD9133, SD9134, SD9135				Predefined protocol registration	The ON/OFF state of the bit corresponding to a protocol number indicates whether the protocol setting data has been registered or not.	R
SD9150	SD9170	SD9190	SD9210	Protocol execution status	The status of a protocol in execution is stored.	R
SD9168	SD9188	SD9208	SD9228	Protocol execution count	The cumulative number of executions of a protocol is stored.	R
SD9169	SD9189	SD9209	SD9229	Protocol cancel specification	The protocol in execution can be cancelled with a value to be stored in this area.	R/W
SD9230	SD9240	SD9250	SD9260	Send/receive data monitoring function setting	The setting of the send/receive data monitoring function is stored.	R/W
SD9231	SD9241	SD9251	SD9261	Send/receive data monitoring function option setting	The option setting of the send/receive data monitoring function is stored.	W
SD9232	SD9242	SD9252	SD9262	Monitoring data device specification	The type of word device used as the monitor data areas is stored.	W

Device No.				Name	Description	R/W
CH1	CH2	CH3	CH4			
SD9233	SD9243	SD9253	SD9263	Monitoring data start device No. specification	The start device number of word devices used as the monitor data areas is stored.	W
SD9234	SD9244	SD9254	SD9264	Monitoring data size specification	The size of word devices used as the monitor data areas is stored in word units.	W

Details of related devices

Serial communication error

Turns ON when an error occurs in serial communication. These flags are for check of the serial communication error.

R: Read only

CH1	CH2	CH3	CH4	Description	R/W
SM8500	SM8510	SM8520	SM8530	Turns ON when an error occurs in serial communication.	R

After a device above turns ON, the error code is stored in the corresponding device below.

CH1	CH2	CH3	CH4	Name	Description
SD8500	SD8510	SD8520	SD8530	Serial communication error code	When a serial communication error occurs, the error code is stored.

Precautions

Do not turn ON or OFF with program or engineering tool.

Serial communication error does not turn OFF even if the communication is restored to normal state. The devices turn OFF when power is turned OFF→ON, STOP→RUN, reset or SM50 (Error Detection Reset Completion) is turned ON.

Serial communication error code

When a serial communication error occurs, these devices store the corresponding error codes (☞ Page 846 Checking absence/presence of predefined protocol support function errors).

R: Read only

CH1	CH2	CH3	CH4	Description	R/W
SD8500	SD8510	SD8520	SD8530	When a serial communication error occurs, the error code is stored.	R

Precautions

Do not change the value with program or engineering tool.

Serial communication error code is not cleared even after communication is restored to normal state. They are cleared by turning the PLC power OFF→ON, STOP→RUN, reset or by SM50 (Error Detection Reset Completion) is turned ON.

Serial communication settings

The communication parameters set in the communication settings are stored when the power is turned OFF→ON, STOP→RUN or the reset. (☞ Page 457 Communication Settings)

R: Read only

CH1	CH2	CH3	CH4	Description	R/W
SD8502	SD8512	SD8522	SD8532	Stores the setting of the communication parameter.	R

The descriptions of the communication parameters are as follows.

Bit No.	Name	Description	
		0 (bit is OFF)	1 (bit is ON)
b0	Data length	7 bits	8 bits
b1 b2	Parity bit	b2, b1 (0, 0): N/A (0, 1): Odd (1, 1): Even	
b3	Stop bit	1 bit	2 bits
b4 b5 b6 b7	Baud rate	b7, b6, b5, b4 (1, 0, 0, 0): 9600bps (1, 0, 0, 1): 19200bps (1, 0, 1, 0): 38400bps (1, 0, 1, 1): 57600bps (1, 1, 0, 1): 115200bps	

Precautions

Do not turn ON or OFF with program or engineering tool.

Serial communication operation mode

Stores the communication function code used in the serial communication under execution.

R: Read only

CH1	CH2	CH3	CH4	Description	R/W
SD8503	SD8513	SD8523	SD8533	0: MELSOFT connection or MC protocol 3: N:N Network Communication 5: Non-Protocol Communication 6: Parallel Link Communication 7: Inverter Communication 9: MODBUS RTU Communication 12: Predefined protocol support Other than above: Not used	R

Precautions

Do not change the value with program or engineering tool.

These devices store "12" while S(P).CPRTCL instruction is being driven or not, as long as the communication mode has not changed.

Predefined protocol ready

The reflected status after the protocol setting data has been written is stored. For details on operation, refer to ☞ Page 484 Information to judge whether the predefined protocol can be executed or not.

R: Read only

Device	Description	R/W
SD9102	0: Error 1: Normal completion	R

Precautions

Do not change the value with program or engineering tool.

The written protocol setting data is reflected when power is turned OFF→ON or reset.

Predefined protocol setting data error

When a protocol setting data error was detected, information to identify the error position is stored.

■Predefined protocol setting data error information: Protocol No.

When a protocol setting data error was detected, the number of the protocol where the error has occurred is stored. Starting from the smallest protocol number, protocols are checked in order, and the number of the protocol where the error has first been detected is stored.

R: Read only

Device	Description	R/W
SD9120	0: No error detected 1 to 64: Protocol No. 65535: Not identifiable[1]	R

[1] Damaged protocol setting data (failures of the memory built in the CPU memory module or the SD memory card) can be a factor that causes a setting value to be not identifiable.

Precautions

Do not change the value with program or engineering tool.

■Predefined protocol setting data error information: Setting type

When a protocol setting data error was detected, the setting type number of the protocol where the error has occurred is stored. This setting is valid when 1 to 64 is set in protocol number (SD9120).

R: Read only

Device	Description	R/W
SD9121	0: Packet setting or element setting 1: Protocol Detailed Setting 65535: Not identifiable[1]	R

[1] Damaged protocol setting data (failures of the memory built in the CPU memory module or the SD memory card) can be a factor that causes a setting value to be not identifiable.

Precautions

Do not change the value with program or engineering tool.

■Predefined protocol setting data error information: Packet No.

When a protocol setting data error was detected, the number of the packet where the error has occurred is stored. Starting from the smallest number of the send packets and receive packets (expected packets), packets are checked in order, and the packet number of the packet where the error has first been detected is stored. This setting is valid when 0 is set in setting type (SD9121).

R: Read only

Device	Description	R/W
SD9122	0: Send packet 1 to 16: Receive packet number 65535: Not identifiable[1]	R

[1] Damaged protocol setting data (failures of the memory built in the CPU memory module or the SD memory card) can be a factor that causes a setting value to be not identifiable.

Precautions

Do not change the value with program or engineering tool.

■Predefined protocol setting data error information: Element No.

When a protocol setting data error was detected, the number of the element where the error has occurred is stored. Starting from the smallest element number, elements are checked in order, and the number of the element where the error has first been detected is stored. This setting is valid when 0 is set in setting type (SD9121).

R: Read only

Device	Description	R/W
SD9123	1 to 32: Element No. 65535: Not identifiable[1]	R

[1] Damaged protocol setting data (failures of the memory built in the CPU memory module or the SD memory card) can be a factor that causes a setting value to be not identifiable.

Precautions

Do not change the value with program or engineering tool.

Number of registered predefined protocols

The number of registered protocol setting data is stored.

R: Read only

Device	Description	R/W
SD9124	1 to 64[1]	R

[1] When a protocol setting data error has occurred, 0 is stored.

Precautions

Do not change the value with program or engineering tool.

Predefined protocol: Protocol registration

The ON/OFF state of the bit corresponding to a protocol number indicates whether the protocol setting data has been registered or not.

R: Read only

Device	Description	R/W
SD9132	A bit corresponding to each protocol number is turned on or off. 1 to 16 (⇔ b0 to b15)	R
SD9133	A bit corresponding to each protocol number is turned on or off. 17 to 32 (⇔ b0 to b15)	R
SD9134	A bit corresponding to each protocol number is turned on or off. 33 to 48 (⇔ b0 to b15)	R
SD9135	A bit corresponding to each protocol number is turned on or off. 49 to 64 (⇔ b0 to b15)	R

The following table lists protocol numbers corresponding to bits of each device.

Device	bit[1]															
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SD9132	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
SD9133	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17
SD9134	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33
SD9135	64	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49

[1] When a protocol setting data error has occurred, 0 is stored in all these areas.

Precautions

Do not change the value with program or engineering tool.

Protocol execution status

The status of a protocol in execution is stored. For details, refer to ☞ Page 484 Protocol execution status.

R: Read only

CH1	CH2	CH3	CH4	Description	R/W
SD9150	SD9170	SD9190	SD9210	0: Unexecuted 1: Waiting for transmission 2: Sending 3: Waiting for data reception 4: Receiving 5: Completed	R

Precautions

Do not change the value with program or engineering tool.

Protocol execution count

The cumulative number of executions of a protocol is stored. Even when a protocol was executed and an error occurred, this execution is also counted and cumulated.

The number of executions is reset to 0 by turning off and on the power supply or resetting the system, and 1 is added every time each protocol is started.

R: Read only

CH1	CH2	CH3	CH4	Description	R/W
SD9168	SD9188	SD9208	SD9228	0 to 65535[1]	R

[1] When cumulated 65535 or more, 65535 is stored.

Precautions

Do not change the value with program or engineering tool.

Protocol cancel specification

The protocol in execution can be cancelled with a value to be stored in this area. For details, refer to ☞ Page 485 Protocol Cancel.

R/W: Read/write

CH1	CH2	CH3	CH4	Description	R/W
SD9169	SD9189	SD9209	SD9229	0: Normal operation (do not cancel) 1: Cancel request 2: Cancel operation completed	R/W

Send/receive data monitoring function setting

The send/receive data monitoring function can be stopped or started by storing a value in hexadecimal. After monitoring has been started, the status of the function is stored. For details, refer to ☞ Page 486 Send/receive data monitoring function.

R/W: Read/write

CH1	CH2	CH3	CH4	Description	R/W
SD9230	SD9240	SD9250	SD9260	0000H: Monitor stop 0001H: Monitor start 0002H: Monitoring (set by system) 1002H: Monitor stop (set by system) 100FH: Monitor setting error (set by system)	R/W

Send/receive data monitoring function option setting

The data area full stop specification and 1-packet stop specification of the send/receive data monitoring function can be enabled or disabled by storing a value. For details, refer to ☞ Page 486 Send/receive data monitoring function.

W: Write only

CH1	CH2	CH3	CH4	Description	R/W
SD9231	SD9241	SD9251	SD9261	The option setting of the send/receive data monitoring function is stored.	W

The following table lists the setting contents of the send/receive data monitoring function option setting.

Bit No.	Name	Description	
		0 (bit is OFF)	1 (bit is ON)
b0	Data area full stop specification	Monitor continued (data is overwritten)	Monitor stop
b1	Packet stop specification	1-packet stop specification disabled	1-packet stop specification enabled
b2 to b15	Not used	—	—

Monitoring data device specification

The type of word device used as the monitor data areas with the send/receive data monitoring function is stored. For details, refer to ☞ Page 487 Monitor data.

W: Write only

CH1	CH2	CH3	CH4	Description	R/W
SD9232	SD9242	SD9252	SD9262	0: D device 1: R device 2: W device 3: SW device	W

Precautions

When a device value is changed after the send/receive data monitoring has been started, the change is not reflected. When monitoring is started after the receive data monitoring was stopped (including a stop due to a receive error), the change is reflected.

When the storage destinations of monitor data overlap for each channel, no error occurs and the values in those areas are overwritten with new values.

Monitoring data start device No. specification

The start device number of word devices used as the monitor data areas with the send/receive data monitoring function is stored. For details, refer to ☞ Page 487 Monitor data.

W: Write only

CH1	CH2	CH3	CH4	Description	R/W
SD9233	SD9243	SD9253	SD9263	0 to 32765	W

Precautions

When a device value is changed after the send/receive data monitoring has been started, the change is not reflected. When monitoring is started after the receive data monitoring was stopped (including a stop due to a receive error), the change is reflected.

When the storage destinations of monitor data overlap for each channel, no error occurs and the values in those areas are overwritten with new values.

Monitoring data size specification

The size of word devices used as the monitor data areas with the send/receive data monitoring function is stored in word units. For details, refer to ☞ Page 487 Monitor data.

W: Write only

CH1	CH2	CH3	CH4	Description	R/W
SD9234	SD9244	SD9254	SD9264	1 to 32765	W

Precautions

When a device value is changed after the send/receive data monitoring has been started, the change is not reflected. When monitoring is started after the receive data monitoring was stopped (including a stop due to a receive error), the change is reflected.

When the storage destinations of monitor data overlap for each channel, no error occurs and the values in those areas are overwritten with new values.

MEMO

PART 3

MODBUS/TCP COMMUNICATION

This part consists of the following chapters.

22 OUTLINE

This chapter explains the FX5 MODBUS/TCP communication (slave station).

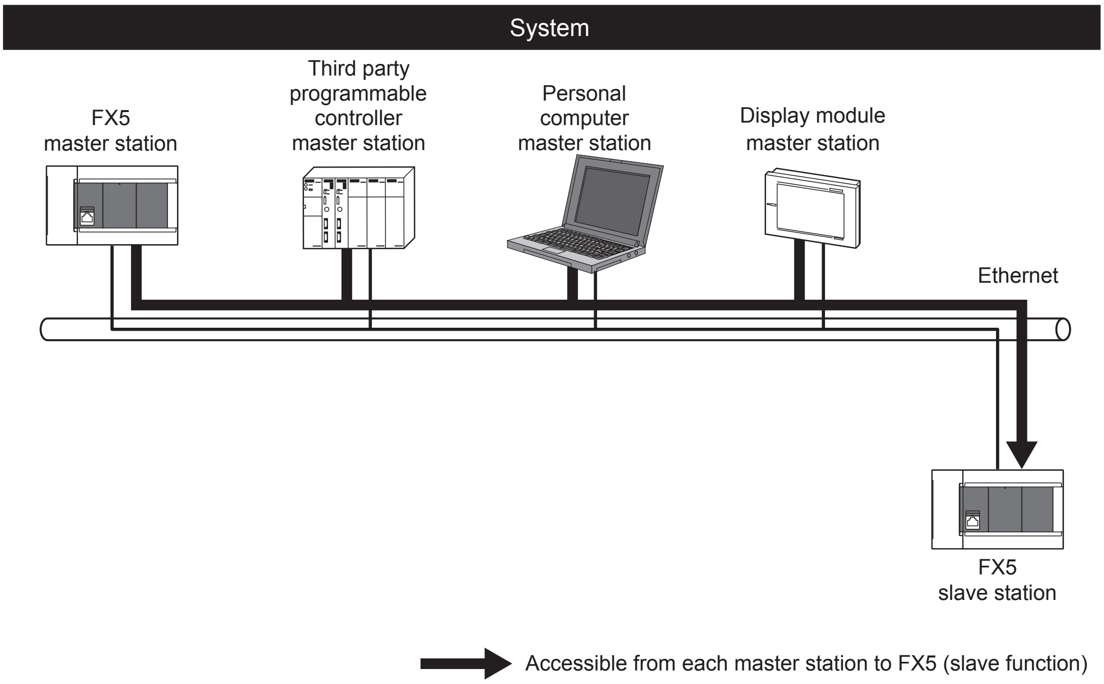
For details on the predefined protocol support function, refer to the following.

☞ Page 100 PREDEFINED PROTOCOL SUPPORT FUNCTION

22.1 Outline of Function

The FX5 MODBUS/TCP communication function allows communication, via Ethernet connection, with various MODBUS/TCP master devices which are connected to FX5 set as the slave station.

- The master function and slave function are supported and the master and slave can be used simultaneously by a single FX5.
- Up to 8 connections can be used for MODBUS/TCP communication function by one CPU module.
- The master uses a predefined protocol support function and controls the slave.



Point

An external device other than FX5 can be used as the master station and slave station.

22.2 Procedure for Operation

The flow chart below shows the procedure for setting up a MODBUS/TCP communication (slave station):

1. Check communication specifications (☞ Page 502 Communication Specifications, ☞ Page 504 MODBUS/TCP COMMUNICATION SPECIFICATIONS)

- Communication specifications
- MODBUS serial communication specifications

Details of MODBUS standard functions

2. Communication settings (☞ Page 506 COMMUNICATION SETTING)

- Communication setting using GX Works3[*1]

Setting parameters

3. Check of function[*2] (☞ Page 510 FUNCTION)

- Slave function

[*1] For details on the connection method or operation method to the PLC of GX Works3, refer to the following manual.
📖 GX Works3 Operating Manual

[*2] The slave station does not need a program.

23 SPECIFICATIONS

This chapter explains the specifications of MODBUS/TCP communication.

23.1 Communication Specifications

MODBUS/TCP communication is executed within the specifications shown in the table below.

For details on communication specifications other than those shown below, refer to the following.

☞ Page 26 SPECIFICATIONS

Item		Specification
Protocol type		MODBUS/TCP (Support only the binary)
Number of connections		Total of 8 connections[*1] (Up to 8 external devices can access one CPU module at the same time.)
Slave function	Number of function	10
	Port station No.	502[*2]

[*1] The number of available connections decreases when the other Ethernet communication function is used. For details on Ethernet function, refer to the following manual.

☞ Page 24 OUTLINE

[*2] The port station No. can be changed by the communication setting.

MEMO

24 MODBUS/TCP COMMUNICATION SPECIFICATIONS

This chapter explains the details of MODBUS Protocol for MODBUS/TCP communication and the MODBUS standard functions supported by the FX5.

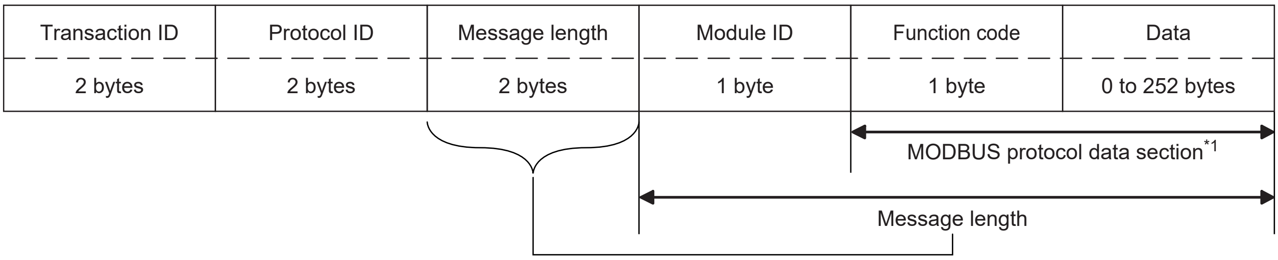
For usage of the supported MODBUS standard functions, refer to the following.

☞ Page 510 FUNCTION

24.1 MODBUS Protocol

Frame specifications

The following figure illustrates the frame specifications for the MODBUS protocol.



Transaction ID	Protocol ID	Message length	Module ID	Function code	Data
2 bytes	2 bytes	2 bytes	1 byte	1 byte	0 to 252 bytes

*1 For details on the MODBUS protocol data section, refer to the following.

☞ Page 900 MODBUS protocol data unit formats

The following table provides details of the frame specifications for the MODBUS Protocol.

Area name	Description
Transaction ID	Used by the master for matching of the response message from the slave.
Protocol ID	Indicates the protocol of the PDU (protocol data unit). Stores 0 in the case of MODBUS/TCP.
Message length	Stores the message size in byte unit. The message length after this field is stored. (See the above figure.)
Module ID	Used to specify the slave connected to the other line, e.g. MODBUS serial protocol. (FX5 is not supported)
Function code	The master specifies the function code to the slave.
Data	[When the master sends a request message to a slave] Stores the requested processing. [When the slave sends a response message to the master] Stores the result of processing execution.

List of supported MODBUS standard functions

The following table lists the MODBUS standard functions supported by the MODBUS/TCP communication of FX5.

Function code	Function Name	Details	Accessible devices per message	Reference
01H	Read coils	Read binary (R/W) devices	1 to 2000 points	Page 901
02H	Read inputs	Read binary (RO) devices	1 to 2000 points	Page 902
03H	Read holding registers	Read 16 bit (R/W) registers	1 to 125 points	Page 903
04H	Read input registers	Read 16 bit (RO) registers	1 to 125 points	Page 904
05H	Write single coil	Write single binary device	1 point	Page 905
06H	Write single register	Write single 16 bit register device	1 point	Page 905
0FH	Write multiple coils	Write multiple binary (R/W) devices	1 to 1968 points	Page 906
10H	Write multiple registers	Write multiple 16 bit (R/W) registers	1 to 123 points	Page 907
16H	Mask write register	Manipulate slave register with AND Mask/OR Mask	1 point	Page 908
17H	Read/write multiple registers	Read/write multiple 16 bit (R/W) registers	Read: 1 to 125 points Write: 1 to 121 points	Page 909

25 COMMUNICATION SETTING

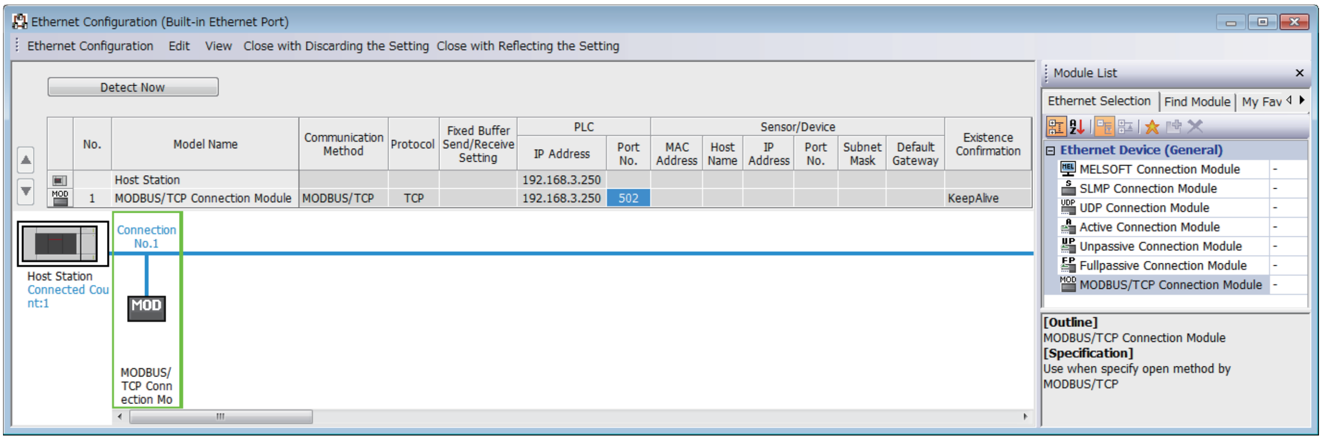
This chapter explains the setup method for using slave station in MODBUS/TCP communication with an FX5.

25.1 Setup Method for Port No.

Set the port No. to identify the communication between the slave station and the master station.

Navigation window ⇨ [Parameter] ⇨ Module model name ⇨ [Module Parameter] ⇨ [Ethernet Port] ⇨ [Basic Settings] ⇨ [External Device Configuration] ⇨ Double click <Detailed Setting> of "External Device Configuration".

Window



No.	Model Name	Communication Method	Protocol	Fixed Buffer Send/Receive Setting	PLC IP Address	PLC Port No.	Sensor/Device MAC Address	Host Name	IP Address	Port No.	Subnet Mask	Default Gateway	Existence Confirmation
	Host Station				192.168.3.250								
1	MODBUS/TCP Connection Module	MODBUS/TCP	TCP		192.168.3.250	502							KeepAlive

[Outline]
MODBUS/TCP Connection Module
[Specification]
Use when specify open method by MODBUS/TCP

Drag and drop the "MODBUS/TCP Connection Module" from "Module List" to the left side on the screen. Execute the settings as described below.

Item		Setting	Remarks
PLC	Port No.	1 to 5548, 5570 to 65534 (Default value: 502) Do not specify 5549 to 5569 because these ports are used by the system.	Set the port No. of slave station.

For setup method for IP address of slave station, refer to the following.

☞ Page 38 Connection via a Hub

Precautions

For host station port number, using 502 is recommended. When changing the port number, use 1024 to 5548 and 5570 to 61439 (0400H to 15ACH and 15C2H to EFFFH).

When using the following functions, do not specify the port No. of the function to be used in MODBUS/TCP 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)

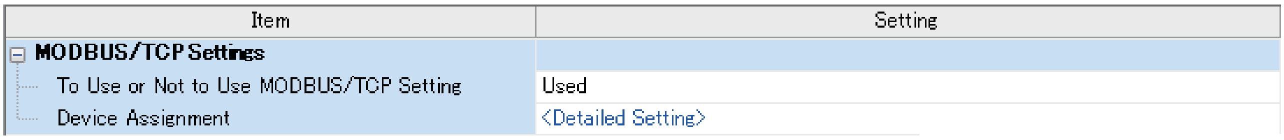
*1 Port No. can be changed. (Default value: 80)

25.2 Setup Method for MODBUS/TCP Communication

For the MODBUS/TCP communication setting of the FX5, set parameters with GX Works3.

Navigation window ⇨ [Parameter] ⇨ Module model name ⇨ [Module Parameter] ⇨ [Ethernet Port] ⇨ [Basic Settings] ⇨ [MODBUS/TCP Settings]

Window



Item	Setting
MODBUS/TCP Settings	
To Use or Not to Use MODBUS/TCP Setting	Used
Device Assignment	<Detailed Setting>

Item	Setting	Remarks
To Use or Not to Use MODBUS/TCP Setting	Used/Not Used	Display the status in External Device Configuration whether MODBUS/TCP connection device is used or not used. (☞ Page 506 Setup Method for Port No.)
Device Assignment	Refer to the following. ☞ Page 508	When To Use or Not to Use MODBUS/TCP Setting is "Used", Device Assignment can be set.

Contents of parameter setting

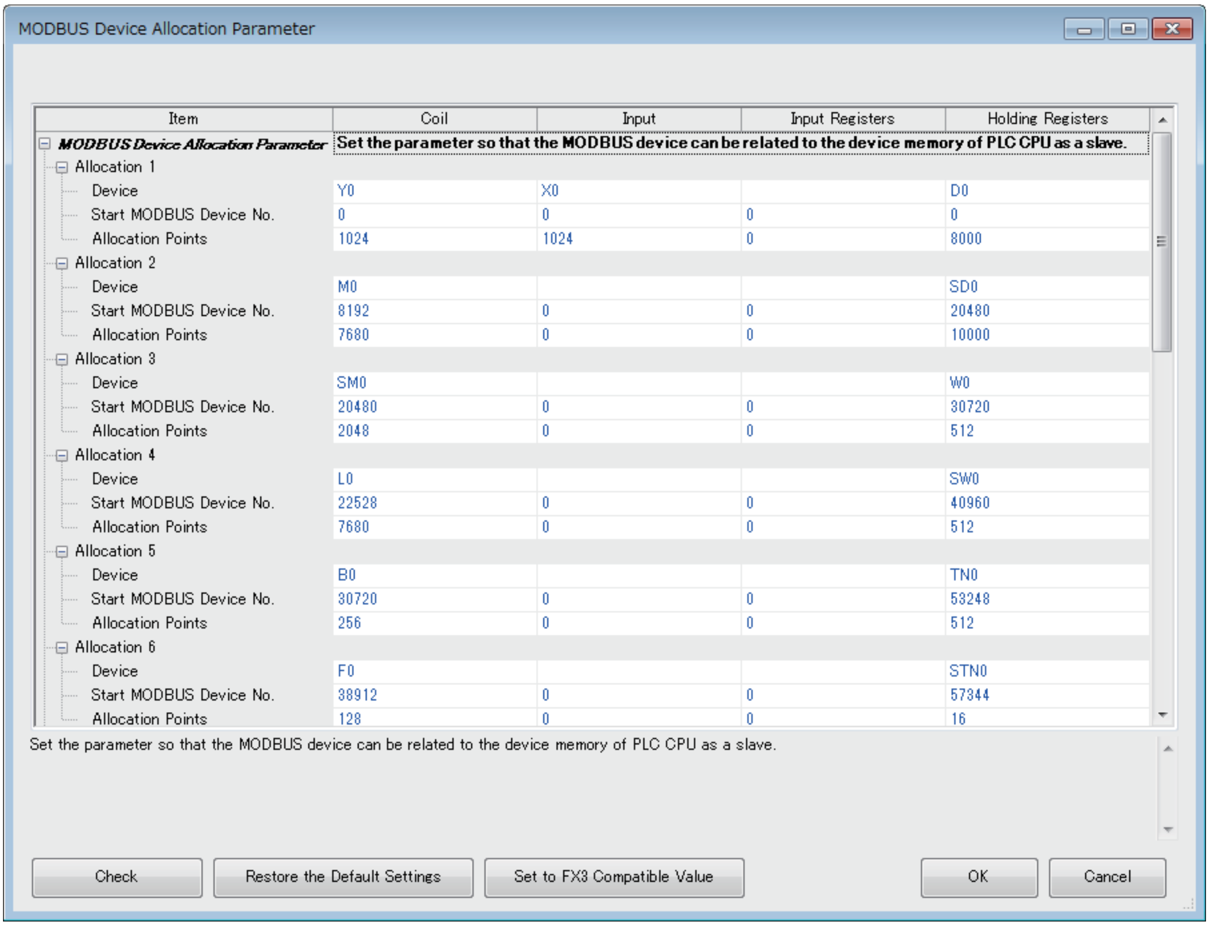
The device allocation used for parameter setting of MODBUS/TCP communication are as follows.

MODBUS device allocation

In the MODBUS device allocation, initial values are set for parameters. (Refer to ☞ Page 910 FX5 dedicated pattern.)
Parameter contents can be changed from the following screen of GX Works3.

Navigation window ⇨ [Parameter] ⇨ Module model name ⇨ [Module Parameter] ⇨ [Ethernet Port] ⇨ [Basic Settings] ⇨ [MODBUS/TCP Settings] ⇨ Double-click <Detailed Setting> of "Device Assignment".

Window



Item	Coil	Input	Input Registers	Holding Registers
MODBUS Device Allocation Parameter	**Set the parameter so that the MODBUS device can be related to the device memory of PLC CPU as a slave.**			
Allocation 1				
Device	Y0	X0		D0
Start MODBUS Device No.	0	0	0	0
Allocation Points	1024	1024	0	8000
Allocation 2				
Device	M0			SD0
Start MODBUS Device No.	8192	0	0	20480
Allocation Points	7680	0	0	10000
Allocation 3				
Device	SM0			W0
Start MODBUS Device No.	20480	0	0	30720
Allocation Points	2048	0	0	512
Allocation 4				
Device	L0			SW0
Start MODBUS Device No.	22528	0	0	40960
Allocation Points	7680	0	0	512
Allocation 5				
Device	B0			TN0
Start MODBUS Device No.	30720	0	0	53248
Allocation Points	256	0	0	512
Allocation 6				
Device	F0			STN0
Start MODBUS Device No.	38912	0	0	57344
Allocation Points	128	0	0	16

Set the parameter so that the MODBUS device can be related to the device memory of PLC CPU as a slave.

Setting item	Description
Allocation 1 to 16	Allocation of each MODBUS device can be set to 1 to 16.
Device	Set the device type and head number of the device to be allocated. (For available devices, refer to ☞ Page 509 Available devices.)
Start MODBUS Device No.	Set the start MODBUS device number.
Allocation Points	Set the number of allocation points.

■Available devices

The following table shows devices that can be set to coil, input, input register, and holding register.

List of devices			Allocable MODBUS device			
Device type		Device	Coil	Input	Input register	Holding register
Special relay		SM	○	○	○[*1]	○[*1]
Special register		SD	—	—	○	○
Input		X	○	○	○[*1]	○[*1]
Output		Y	○	○	○[*1]	○[*1]
Internal relay		M	○	○	○[*1]	○[*1]
Latch relay		L	○	○	○[*1]	○[*1]
Annunciator		F	○	○	○[*1]	○[*1]
Link relay		B	○	○	○[*1]	○[*1]
Data register		D	—	—	○	○
Link register		W	—	—	○	○
Timer	Coil	TC	○	○	○[*1]	○[*1]
	Contact	TS	○	○	○[*1]	○[*1]
	Current value	TN	—	—	○	○
Retentive timer	Coil	STC	○	○	○[*1]	○[*1]
	Contact	STS	○	○	○[*1]	○[*1]
	Current value	STN	—	—	○	○
Counter	Coil	CC	○	○	○[*1]	○[*1]
	Contact	CS	○	○	○[*1]	○[*1]
	Current value	CN	—	—	○	○
Long counter	Coil	LCC	○	○	○[*1]	○[*1]
	Contact	LCS	○	○	○[*1]	○[*1]
	Current value	LCN	—	—	○[*2]	○[*2]
Link special relay		SB	○	○	○[*1]	○[*1]
Link special register		SW	—	—	○	○
Step relay		S	○	○	○[*1]	○[*1]
Index register		Z	—	—	○	○
Index register		LZ	—	—	○[*2]	○[*2]
File register		R	—	—	○	○
Extended file register		ER	—	—	—	—

*1 Set the device number and the allocating points in multiples of 16. In the case that the number of points is not a multiple of 16, a parameter setting error occurs in GX Works3.

*2 Because MODBUS devices are long type devices, 2 points are allocated.

Precautions

- The same device cannot be set for a coil and an input.
- The same device cannot be set for an input register and a holding register.
- A parameter setting error occurs in GX Works3 when the total of selected head device number and the allocated points exceed the valid range for the selected PLC device.

26 FUNCTION

This chapter explains the function of MODBUS/TCP communication.

26.1 Master Function

The master station of FX5 executes communication between the master station and the slave station by using the predefined protocol support function. The CPU module executing the predefined protocol support function will be the master station.

The communication setup method and procedure for program of MODBUS/TCP communication (master) are shown below.

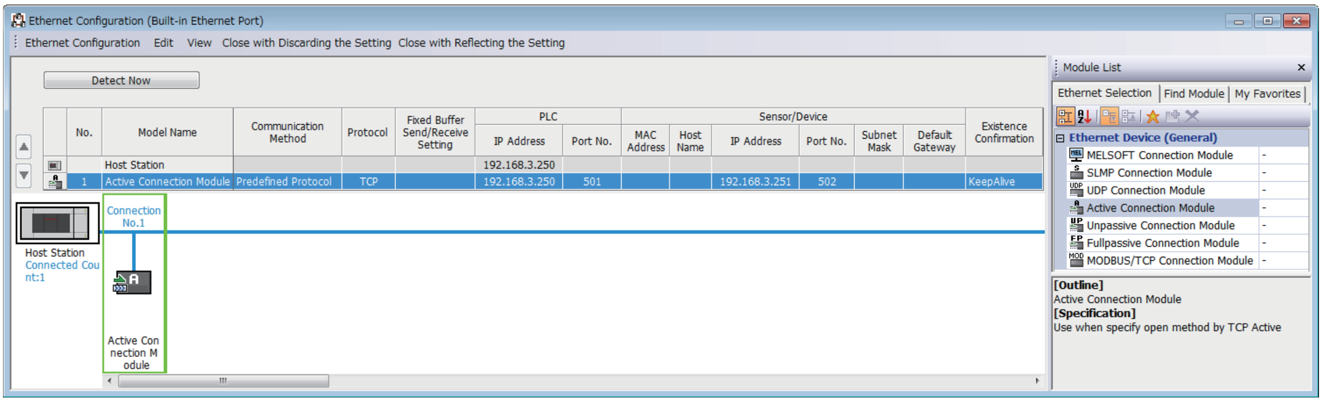
For details on each operation and application instruction, refer to the following.

☞ Page 24 OUTLINE

Connection settings

Navigation window ⇨ [Parameter] ⇨ Module model name ⇨ [Module Parameter] ⇨ [Ethernet Port] ⇨ [Basic Settings] ⇨ [External Device Configuration] ⇨ Double click <Detailed Setting> of "External Device Configuration".

Window



No.	Model Name	Communication Method	Protocol	Fixed Buffer Send/Receive Setting	PLC IP Address	PLC Port No.	Sensor/Device MAC Address	Sensor/Device Host Name	Sensor/Device IP Address	Sensor/Device Port No.	Sensor/Device Subnet Mask	Sensor/Device Default Gateway	Existence Confirmation
	Host Station				192.168.3.250								
1	Active Connection Module	Predefined Protocol	TCP		192.168.3.250	501			192.168.3.251	502			KeepAlive

Ethernet Device (General)

MELSOFT Connection Module	-
SLMP Connection Module	-
UDP Connection Module	-
Active Connection Module	-
Unpassive Connection Module	-
Fullpassive Connection Module	-
MODBUS/TCP Connection Module	-

[Outline]
Active Connection Module
[Specification]
Use when specify open method by TCP Active

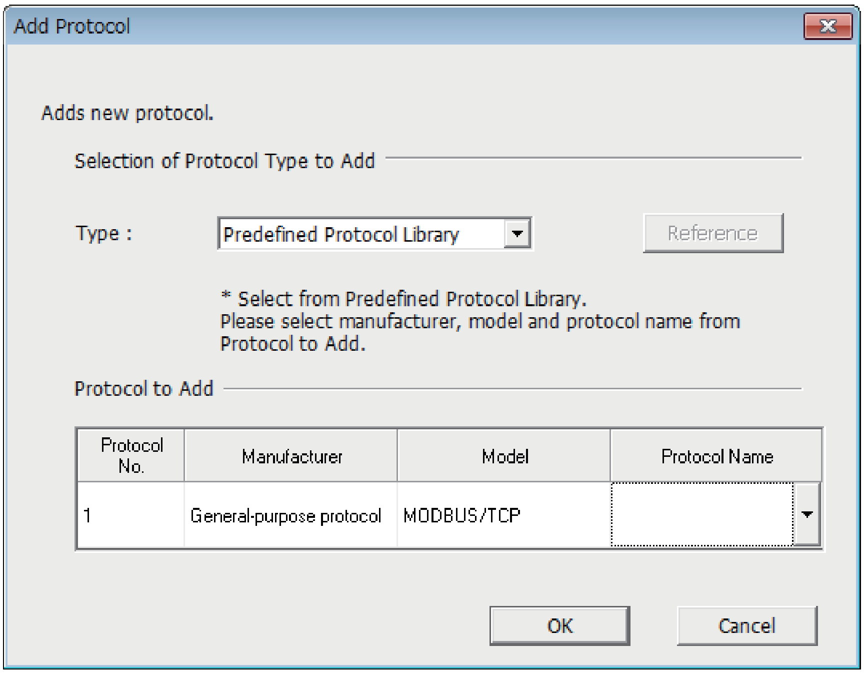
Drag and drop the "Active Connection Module" from "Module List" to the left side on the screen. Execute the settings as described below.

Item		Setting	Remarks
Communication Method		Specify the Predefined Protocol.	—
PLC	Port No.	1 to 5548, 5570 to 65534 Do not specify 5549 to 5569 because these ports are used by the system.	Set the port No. of master station.
Sensor/Device	IP Address	0.0.0.1 to 223.255.255.254	Set the IP Address of slave station.
	Port No.	—	Set the port No. of slave station.

Protocol settings

Send the request message from the master station to the slave station through the predefined protocol support function.
Create the protocol data of MODBUS function requesting to the slave station.

[Tool] ⇨ [Predefined Protocol Support Function] ⇨ [File] ⇨ [New] ⇨ "Protocol Setting" screen ⇨ [Edit] ⇨ [Add Protocol]



Adds new protocol.

Selection of Protocol Type to Add

Type : Predefined Protocol Library

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

Protocol to Add

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

Set the following items, and add the protocol of MODBUS/TCP communication.

Item	Description
Type	Specify the Predefined ProtocolLlibrary.
Manufacturer	Specify the General-purpose protocol.
Model	Specify MODBUS/TCP.
Protocol Name[*1]	01: RD Coils (01H: Read coils)
	02: RD Discrete Inputs (02H: Read discrete inputs)
	03: RD Holding Registers (03H: Read holding registers)
	04: RD IN Registers (04H: Read input registers)
	05: WR Single Coil (05H: Write single coil)
	06: WR Single Registers (06H: Write single register)
	15: WR Multi Coil (0FH: Write multiple coils)
	16: WR Multi Registers (10H: Write multiple registers)
	20: RD File Record (14H: Read file record)[*2]
	21: WR File Record (15H: Write file record)[*2]
	22: Mask WR Registers (16H: Mask write register)
	23: RD/WR Multi Registers (17H: Read/write multiple registers)

*1 ( ) is the MODBUS standard function compatible with each protocol name.
*2 Support only the master station.

Set the packet setting to the added protocol according to the description of MODBUS device read/write.

Point

When setting the packet setting to the FX5 slave station, use the device assigned MODBUS device for the device specified to variable. (☞ Page 508 MODBUS device allocation)

Program

The flow of the program performing the MODBUS/TCP communication by the FX5 master station is shown below.

1. Establishes a connection

Open a connection of the slave station for communication by SP.SOCOPEN instruction.

2. Executes the protocols

Execute the protocol (MODBUS standard function) set to the slave station for communication by SP.ECPRTCL instruction.

3. Closes a connection

Close a connection of the slave station for communication by SP.SOCCLOSE instruction.

26.2 Slave Function

The functions of slave station of FX5 are the followings.

Automatic response

This function performs operations according to the supported function code by communicating with the master station.

For the supported function codes, refer to ☞ Page 505 List of supported MODBUS standard functions.

MODBUS device allocation

This function automatically converts access of the slave station to MODBUS device into access to the CPU module.

MODBUS device is allocated to the device set in the communication setting.

For details on MODBUS device allocation, refer to ☞ Page 508 MODBUS device allocation.

Ethernet diagnostics

This function can execute the troubleshooting when an abnormality occurs.

For details, refer to the following.

☞ Page 799 Ethernet diagnostics

26.3 Related Devices

In this section, the functions of the special registers are described for MODBUS/TCP communication.

The table shows the special registers used for the FX5 MODBUS/TCP communication.

R: Read only

Device number	Name	Valid	Details	R/W
SD10130 to SD10137	Error code	Master/ Slave	Error code of built-in Ethernet (connection 1 to connection 8) is stored.	R

Details of related devices

The following devices are used in MODBUS/TCP communication.

Error code

Stores the current error code of built-in Ethernet. (☞ Page 850 MODBUS Serial Communication)

R: Read only

Device number	Name	Description	R/W
SD10130	Error code (Connection No.1)	Stores the current error code generated during built-in Ethernet.	R
SD10131	Error code (Connection No.2)		
SD10132	Error code (Connection No.3)		
SD10133	Error code (Connection No.4)		
SD10134	Error code (Connection No.5)		
SD10135	Error code (Connection No.6)		
SD10136	Error code (Connection No.7)		
SD10137	Error code (Connection No.8)		

Precautions

Do not change the device value using a program or an engineering tool.

MEMO

PART 4

MODBUS SERIAL COMMUNICATION

This part consists of the following chapters.

27 OUTLINE

This chapter explains the FX5 MODBUS serial communication.

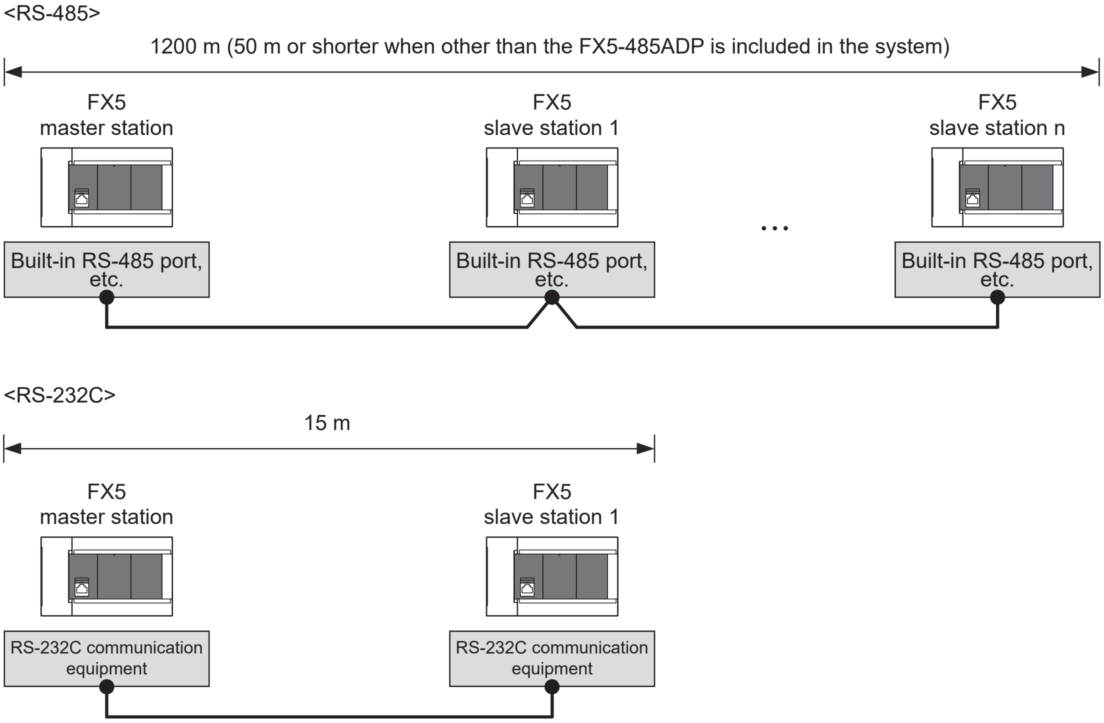
- When predefined protocol support function is used, refer to the following.
  ☞ Page 452 PREDEFINED PROTOCOL SUPPORT FUNCTION

27.1 Outline of Function

The FX5 MODBUS serial communication function can control 32 slaves for RS-485 communication and one slave for RS-232C communication by a single master.

- The master function and slave function are supported and the master and slave can be used simultaneously by a single FX5. (Only one channel for the master)
- Up to 4 channels[1] can be used for MODBUS serial communication function by one CPU module.
- The master uses a PLC command dedicated to MODBUS serial communication and controls the slave.
- The communication protocol supports the RTU mode.

System



Point

The slave station No. can be set from 1 to 247[2]. However, the number of slave station which can be connected to the FX5 master station is 32.

---

[1] Maximum number of channels differs depending on the CPU module. (☞ Page 518 System Configuration)

[2] Slave station No. that can be set in FX5 master station differs depending on the version. (☞ Page 936 Added and Changed Functions)

27.2 Procedure for Operation

The flow chart below shows the procedure for setting up a MODBUS serial communication network:

1. Check communication specifications (☞ Page 521 SPECIFICATIONS, ☞ Page 525 MODBUS COMMUNICATION SPECIFICATIONS)

- Communication Specifications

Link Time

- MODBUS Serial Communication Specifications

MODBUS Serial Communication Protocol, Details of MODBUS Standard Functions

2. System configuration and selection (☞ Page 518 CONFIGURATION)

- System Configuration

Selection of communication equipment

3. Wiring (☞ Page 527 WIRING)

- Wiring procedure

Wiring example

4. Communication settings (☞ Page 533 COMMUNICATION SETTING)

- Communication setting using GX Works3[*1]

[*1] For details on the connection method or operation method to the PLC of GX Works3, refer to the following manual.
📖 GX Works3 Operating Manual

Setting parameters, Related special device

5. Program creation (☞ Page 538 FUNCTION, ☞ Page 550 CREATING PROGRAMS)

- Function

Master function, Slave function, Related special device

- MODBUS serial communication program

Example of creating programs of the master using ADPRW instruction

28 CONFIGURATION

This chapter explains the configuration of RS-485 and RS-232C communication of the FX5.

28.1 System Configuration

This section outlines the system configuration required to use MODBUS serial communication.

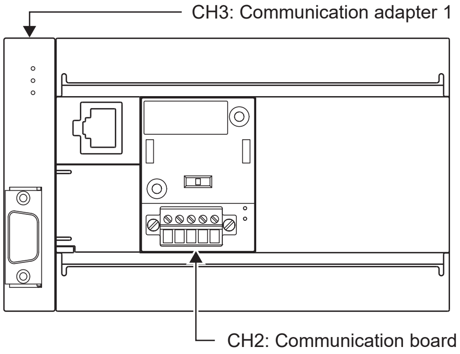
FX5S CPU module

For FX5S CPU modules, up to two channels of communication port can be connected by using communication boards and communication adapters.

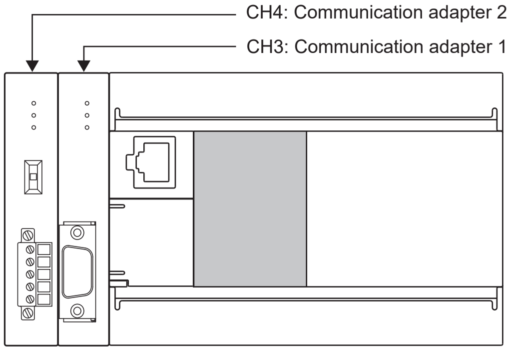
Communication channel assignments are fixed regardless of the system configuration.

The combinations available for the system configurations are shown below.

[Configuration example 1]



[Configuration example 2]



Item		Communication port	Important point in selection	Total extension distance
Communication board	FX5-485-BD	CH2	Because the board can be mounted on top of the CPU module, there is no change in the installation space to be required.	50 m or less
	FX5-232-BD			15 m or less
Communication adapter	FX5-485ADP	CH3, CH4[1]	Mount the communication adapter to the left of the CPU module	1200 m or less
	FX5-232ADP			15 m or less

[1] The adapters are assigned to CH3 and CH4 in the order, from the closest one to the CPU module.

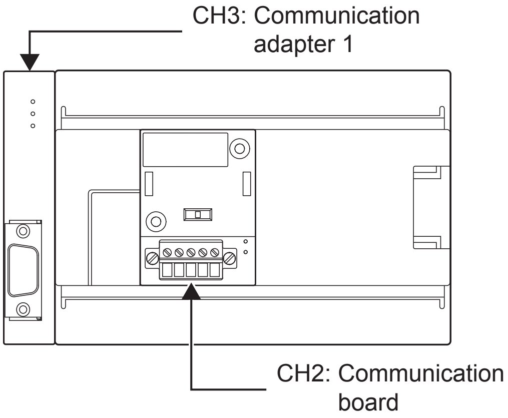
FX5UJ CPU module

In FX5UJ CPU module, up to two communication port channels can be connected to a CPU module using communication board, and communication adapter.

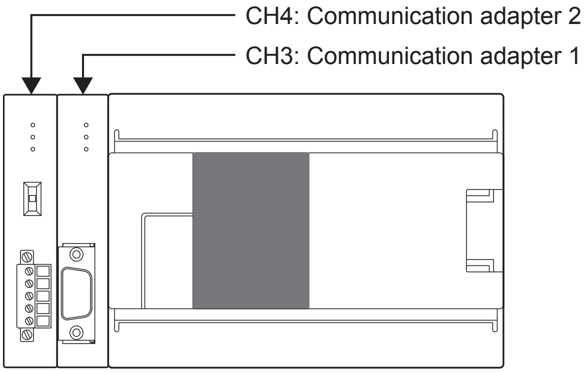
Communication channel assignments are fixed regardless of the system configuration.

The combinations which can be configured are shown below.

[Configuration example 1]



[Configuration example 2]



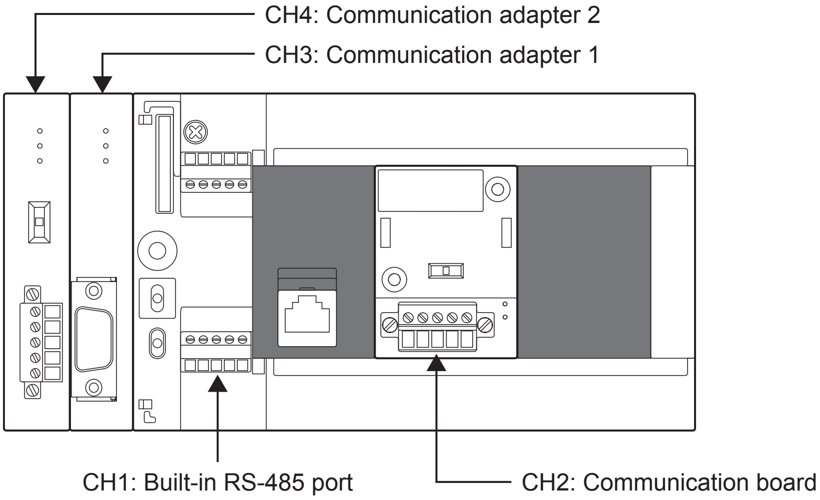
Item		Communication port	Important point in selection	Total extension distance
Communication board	FX5-485-BD	CH2	Mounted on top of the CPU module, there is no change in the installation space requirements	50 m or less
	FX5-232-BD			15 m or less
Communication adapter	FX5-485ADP	CH3, CH4[*1]	Mount the communication adapter to the left of the CPU module	1200 m or less
	FX5-232ADP			15 m or less

*1 The adapters are assigned to CH3 and CH4 in the order, from the closest one to the CPU module.

FX5U CPU module

In FX5U CPU module, up to four communication port channels can be connected to a CPU module using built-in RS-485 port, communication board, and communication adapter.

Communication channel assignments are fixed regardless of the system configuration.



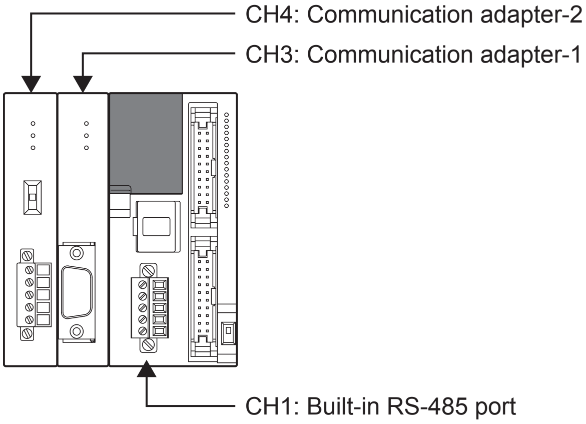
Item		Communication port	Important point in selection	Total extension distance
Built-in RS-485 port		CH1	Since it is built-into the CPU module, there is no need to add equipment	50 m or less
Communication board	FX5-485-BD	CH2	Mounted on top of the CPU module, there is no change in the installation space requirements	50 m or less
	FX5-232-BD			15 m or less
Communication adapter	FX5-485ADP	CH3, CH4[1]	Mount the communication adapter to the left of the CPU module	1200 m or less
	FX5-232ADP			15 m or less

[1] The adapters are assigned to CH3 and CH4 in the order, from the closest one to the CPU module.

FX5UC CPU module

In FX5UC CPU module, up to three communication port channels can be connected to a CPU module using built-in RS-485 port, and communication adapter.

Communication channel assignments are fixed regardless of the system configuration.



Item		Communication port	Important point in selection	Total extension distance
Built-in RS-485 port		CH1	Since it is built-into the CPU module, there is no need to add equipment	50 m or less
Communication adapter	FX5-485ADP	CH3, CH4[1]	Mount the communication adapter to the left of the CPU module	1200 m or less
	FX5-232ADP			15 m or less

[1] The adapters are assigned to CH3 and CH4 in the order, from the closest one to the CPU module.

29 SPECIFICATIONS

This chapter explains the specifications of MODBUS serial communication.

29.1 Communication Specifications

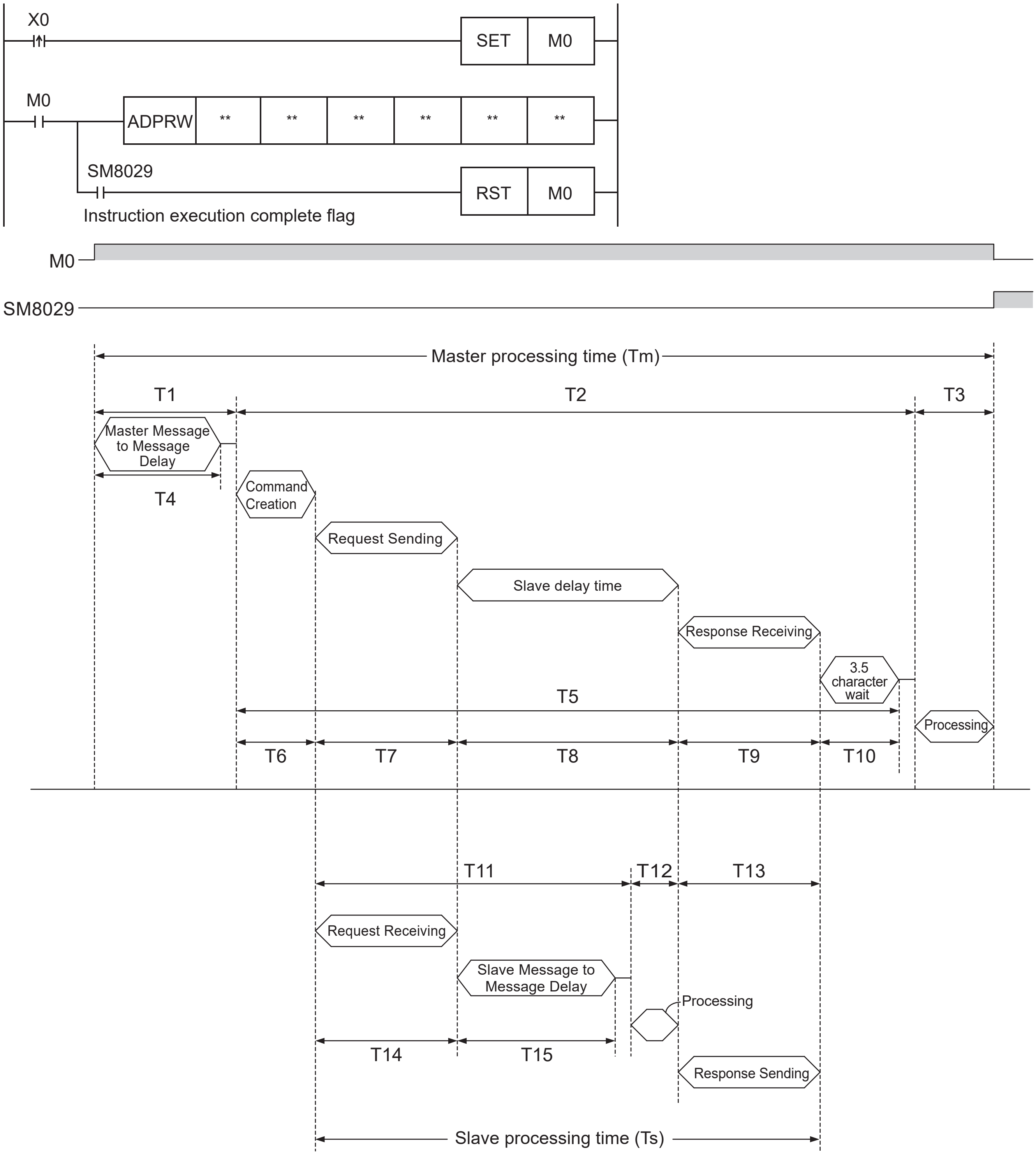
MODBUS serial communication is executed within the specifications shown in the table below. The baud rate, etc. can be changed in the parameter settings of GX Works3.

Item		**Specifications**		**Remarks**
		Built-in RS-485 port **FX5-485-BD** **FX5-485ADP**	**FX5-232-BD** **FX5-232ADP**	
Number of connectable modules		Up to 4 Channels[1] (Only one channel for the master)		The channel can be used as master or slave.
Communication specifications	Communication interface	RS-485	RS-232C	—
	Baud rate	300/600/1200/2400/4800/9600/19200/38400/57600/115200 bps		—
	Data length	8 bit		—
	Parity bit	None, odd or even		—
	Stop bit	1 bit/2 bit		—
	Transmission distance	1200 m or less for the system configured with FX5-485ADP only 50 m or less for the system configured with devices other than above	15 m or less	The transmission distance varies depending on communication equipment type.
	Communication protocol	RTU		—
Master function	Number of connectable slaves	32 slaves	1 slave	The number of slaves varies depending on communication equipment type.
	Number of functions	8 (No diagnosis function)		—
	Number of simultaneous sending messages	1 message		—
	Maximum number of writes	123 words or 1968 coils		—
	Maximum number of reads	125 words or 2000 coils		—
Slave function	Number of functions	8 (No diagnosis function)		—
	Number of messages which can be received simultaneously	1 message		—
	Station number	1 to 247		—

[1] Maximum number of channels differs depending on the CPU module. (☞ Page 518 System Configuration)

29.2 Link Time

The link time indicates the cycle time in which a master module completes a single instruction with a slave, as illustrated by the diagram below.



The master processing time (Tm) can be calculated in milliseconds (ms) as follows. "INT(n)" indicates an integer obtained by truncating decimal places of "n".

Character length (bits):

Start bit (1 bit) + Data Length (8 bit) + Parity bit (0 bit or 1 bit) + Stop bit (1 bit or 2 bit)

Tm = T1 + T2 + T3

$$T1 = \left(INT\left(\frac{T4}{\text{Max. Scan Time}}\right)+1\right) \times \text{Max. Scan Time}$$

T4 = SD8864, SD8874, SD8884, or SD8894 (depending on the communication channel)

$$T2 = \left(INT\left(\frac{T5}{\text{Max. Scan Time}}\right)+1\right) \times \text{Max. Scan Time}$$

T5 = T6 + T7 + T8 + T9 + T10

T6 = less than 1 ms

$$T7 = \frac{\text{Number of Bytes in Request} \times \text{Character Length (bits)}}{\text{Baud Rate (bps)}} \times 1000\text{ (ms)} + 1\text{ ms}$$

T8 = Slave delay time (depending on the slave)

$$T9 = \frac{\text{Number of Bytes in Response} \times \text{Character Length (bits)}}{\text{Baud Rate (bps)}} \times 1000\text{ (ms)} + 1\text{ ms}$$

$$T10 = \frac{\text{3.5 Characters} \times \text{Character Length (bits)}}{\text{Baud Rate (bps)}} \times 1000\text{ (ms)} + 1\text{ ms}$$

T3 = less than 1 ms

The slave processing time (Ts) can be calculated in milliseconds (ms) as follows.

Character length (bits):

Start bit (1 bit) + Data Length (8 bit) + Parity bit (0 bit or 1 bit) + Stop bit (1 bit or 2 bit)

Ts = T11 + T12 + T13

T11 = T14 + T15 + Max. Scan Time

$$T14 = \frac{\text{Number of Bytes in Request} \times \text{Character Length (bits)}}{\text{Baud Rate (bps)}} \times 1000\text{ (ms)} + 1\text{ ms}$$

T15 = SD8864, SD8874, SD8884, or SD8894 (depending on the communication channel)

T12 = less than 1 ms

$$T13 = \frac{\text{Number of Bytes in Response} \times \text{Character Length (bits)}}{\text{Baud Rate (bps)}} \times 1000\text{ (ms)} + 1\text{ ms}$$

Example link time calculations:

Master processing time (Tm)

SD8864 = 5 ms
Max. scan time = 5 ms
Function = Read holding registers 0 to 9 (function code: 03H)
Frame mode = RTU mode
Number of bytes in request = 8 bytes (Address: 1 byte, Frame: 5 bytes, CRC: 2 bytes)
Number of bytes in response = 25 bytes (Address echo: 1 byte, Frame: 22 bytes, CRC: 2 bytes)
Character length = 10 bits (Start bit: 1 bit, Data length: 8 bits, Parity bit: 0 bit, Stop bit: 1 bit)
Baud rate = 19.2 kbps
Slave delay time = 10 ms

$T4 = 5\ \text{ms}$

$$T1 = \left(\text{INT}\left(\frac{5\ \text{ms}}{5\ \text{ms}}\right)+1\right) \times 5\ \text{ms} = (1 + 1) \times 5\ \text{ms} = 10\ \text{ms}$$

$T6 \fallingdotseq 1\ \text{ms}$

$$T7 = \frac{8\ \text{Bytes} \times 10\ \text{Bits}}{19200\ \text{bps}} \times 1000\ (\text{ms}) + 1\ \text{ms} \fallingdotseq 5.2\ \text{ms}$$

$T8 = 10\ \text{ms}$

$$T9 = \frac{25\ \text{Bytes} \times 10\ \text{Bits}}{19200\ \text{bps}} \times 1000\ (\text{ms}) + 1\ \text{ms} \fallingdotseq 14.0\ \text{ms}$$

$$T10 = \frac{3.5\ \text{Characters} \times 10\ \text{Bits}}{19200\ \text{bps}} \times 1000\ (\text{ms}) + 1\ \text{ms} \fallingdotseq 2.8\ \text{ms}$$

$T5 = 1\ \text{ms} + 5.2\ \text{ms} + 10\ \text{ms} + 14.0\ \text{ms} + 2.8\ \text{ms} = 33\ \text{ms}$

$$T2 = \left(\text{INT}\left(\frac{33\ \text{ms}}{5\ \text{ms}}\right)+1\right) \times 5\ \text{ms} = (6 + 1) \times 5\ \text{ms} = 35\ \text{ms}$$

$T3 \fallingdotseq 1\ \text{ms}$

$Tm = 5\ \text{ms} + 35\ \text{ms} + 1\ \text{ms} = \underline{41\ \text{ms}}$

Slave processing time (Ts)

Function = Read holding registers 0 to 9 (function code: 03H)
Frame mode = RTU mode
Number of bytes in request = 8 bytes (Address: 1 byte, Frame: 5 bytes, CRC: 2 bytes)
Number of bytes in response = 25 bytes (Address echo: 1 byte, Frame: 22 bytes, CRC: 2 bytes)
Character length = 10 bits (Start bit: 1 bit, Data length: 8 bits, Parity bit: 0 bit, Stop bit: 1 bit)
Baud rate = 19.2 kbps
SD8864 = 5 ms
Max. scan time = 5 ms

$$T14 = \frac{8\ \text{Bytes} \times 10\ \text{Bits}}{19200\ \text{bps}} \times 1000\ (\text{ms}) + 1\ \text{ms} \fallingdotseq 5.2\ \text{ms}$$

$T15 = 5\ \text{ms}$

$T11 = 5.2\ \text{ms} + 5\ \text{ms} + 5\ \text{ms} = 15.2\ \text{ms}$

$T12 \fallingdotseq 1\ \text{ms}$

$$T13 = \frac{25\ \text{Bytes} \times 10\ \text{Bits}}{19200\ \text{bps}} \times 1000\ (\text{ms}) + 1\ \text{ms} \fallingdotseq 14.0\ \text{ms}$$

$Ts = 15.2\ \text{ms} + 1\ \text{ms} + 14.0\ \text{ms} = \underline{30.2\ \text{ms}}$

30 MODBUS COMMUNICATION SPECIFICATIONS

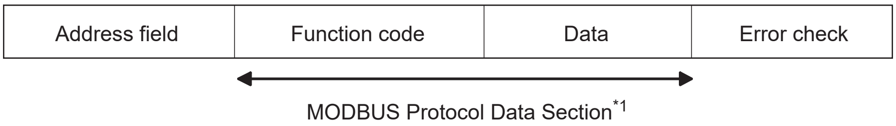
This chapter explains the details of MODBUS Protocol for MODBUS serial communication and the MODBUS standard functions supported by the FX5.

For usage of the supported MODBUS standard functions, refer to the following.

☞ Page 538 FUNCTION

30.1 MODBUS Protocol

The following figure illustrates the frame specifications for the MODBUS protocol.



*1 For details on the MODBUS protocol data section, refer to the following.
☞ Page 900 MODBUS protocol data unit formats

The following table details the frame specifications for the MODBUS Protocol.

Area name	Description
Address field	[When the master sends a request message to a slave] 0: Sends a request message to all the slaves. (Broadcast) 1 to 247: Sends a request to a specific Slave number. Note: 247 is the MODBUS maximum address number.[2] [When the slave sends a response message to the master] The host station number is stored when sending a response message.
Function code	[When the master sends a request message to a slave] The master specifies the function code to the slave. [When the slave sends a response message to the master] The requested function code is stored in the case of normal completion. The most significant bit turns ON in the case of abnormal end.
Data	[When the master sends a request message to a slave] The information needed to execute the action specified by a function code is stored. [When the slave sends a response message to the master] The execution result of the action specified by a function code is stored. An exception code is stored when failed.
Error check	The node (master or slave) adds the check code automatically to all transmitted messages and recalculates the check code for any received message. The received message is discarded if it has an error.

*2 The address number that can be used in FX5 master station differs depending on the version. (☞ Page 936 Added and Changed Functions)

Refer to ☞ Page 526 Frame mode for the data size of each area.

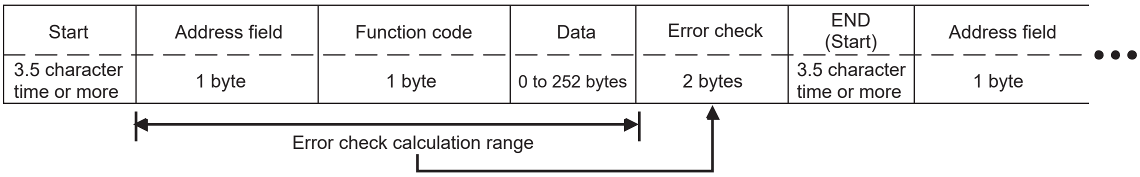
Frame mode

For the FX5, the following frame modes are available. If the frame mode of the FX5 differs from the one of the target device, it cannot be used.

Available frame modes

■RTU mode

In this mode, frames are received or sent in binary codes.
The frame specifications are compliant with the MODBUS protocol specifications.



Point

The error check in the RTU mode is conducted by CRC (Cyclic Redundancy Check).
The CRC field is two bytes, containing a 16 bit binary value. The CRC value is calculated by the transmitting device, which appends the CRC to the message. The device that receives recalculates a CRC during receipt of the message, and compares the calculated value to the actual value it received in the CRC field. If the two values are not equal, an error occurs. (For procedure for generating of CRC, refer to ☞ Page 898 Frame Specifications)

List of supported MODBUS standard functions

The following table lists the MODBUS standard functions supported by the FX5.
○: Supported, ×: Not supported

Function code	Function Name	Details	Accessible devices per message	Broadcast	Reference
01H	Read coils	Read binary (R/W) devices	1 to 2000 points	×	Page 901
02H	Read inputs	Read binary (RO) devices	1 to 2000 points	×	Page 902
03H	Read holding registers	Read 16 bit (R/W) registers	1 to 125 points	×	Page 903
04H	Read input registers	Read 16 bit (RO) registers	1 to 125 points	×	Page 904
05H	Write single coil	Write single binary device	1 point	○	Page 905
06H	Write single register	Write single 16 bit register device	1 point	○	Page 905
0FH	Write multiple coils	Write multiple binary (R/W) devices	1 to 1968 points	○	Page 906
10H	Write multiple registers	Write multiple 16 bit (R/W) registers	1 to 123 points	○	Page 907

31 WIRING

This chapter explains the wiring.

31.1 Wiring Procedure

1. Preparing for wiring

Prepare cables required for wiring. (☞ Page 527 Selecting Connection)

2. Turning off the power to the PLC

Before wiring, make sure that the power of the PLC is off.

3. Wiring communication equipment

Connect RS-485 or RS-232C communication equipment. (☞ Page 531 Connection Diagram)

31.2 Selecting Connection

Select cables using the procedure described below.

For RS-232C

Use an RS-232C cable of the RS-232 standard within 15 m.

For RS-485

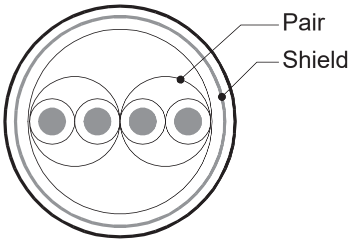
Use shielded twisted pair cables for connecting RS-485 communication equipment.

Twisted pair cable

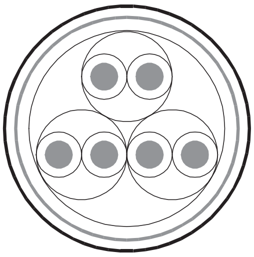
■RS-485 cable specifications

Item	Description
Cable type	Shielded cable
Number of pairs	2p, 3p
Conductor resistance (20℃)	88.0 Ω/km or less
Insulation resistance	10000 MΩ-km or more
Dielectric strength	500VDC, 1 minute
Electrostatic capacitance (1kHz)	60 nF/km or less as an average
Characteristic impedance (100kHz)	110±10 Ω

■Cable structural drawing (reference)



Example of two-pair cable structural drawing



Example of three-pair cable structural drawing

Connecting cables

The table below shows applicable cables and tightening torques.

Item	Number of connected electric wires per terminal	Cable size		Tightening torque
		Solid wire, Stranded wire	Wire ferrule with insulating sleeve	
FX5U CPU module built-in RS-485 port	1-wire connection	0.2 to 0.5㎟ (24 to 20 AWG)	0.2 to 0.5㎟ (24 to 20 AWG)	0.22 to 0.25 N·m
	2-wire connection	0.2㎟ (24 AWG)	—	
FX5UC CPU module built-in RS-485 port FX5-485-BD FX5-485ADP	1-wire connection	0.3 to 0.5㎟ (22 to 20 AWG)	0.3 to 0.5㎟ (22 to 20 AWG)	
	2-wire connection	0.3㎟ (22 AWG)	—	

Precautions

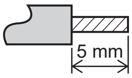
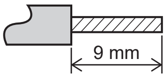
Do not tighten terminal screws with torque beyond the specified range. Otherwise it may cause equipment failure or malfunction.

Wire end treatment

With regard to the cable end treatment, use a stranded cable or solid cable as is, or use a wire ferrule with insulating sleeve.

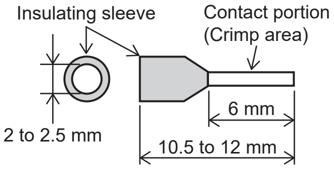
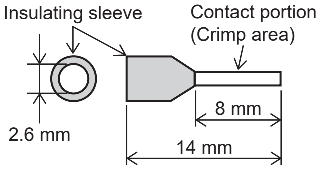
■When using a stranded cable or solid cable as is

- Twist the end of stranded wire and make sure that there are no loose wires.
- Please do not solder plate the ends of the wires.

Dimension of the wire end	
FX5U CPU module built-in RS-485 port	FX5UC CPU module built-in RS-485 port, FX5-485-BD, FX5-485ADP
	

■When using a wire ferrule with insulating sleeve

Because it is difficult to insert a cable into the insulating sleeve depending on the thickness of the cable sheath, select the proper cable according to the outline drawing.

FX5U CPU module built-in RS-485 port	FX5UC CPU module built-in RS-485 port, FX5-485-BD, FX5-485ADP
	

<Reference>

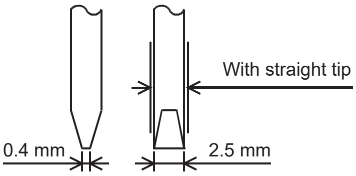
Item	Manufacturer	Model name	Crimping tool
FX5U CPU module built-in RS-485 port	Phoenix Contact GmbH & Co. KG	AI 0.5-6WH	CRIMPFOX 6 CRIMPFOX 6T-F
FX5UC CPU module built-in RS-485 port FX5-485-BD FX5-485ADP		AI 0.5-8WH	

■Tool

For tightening the terminal, use a commercially available small screwdriver with straight tip that is not widened toward the end as shown below.

■Precautions

If the diameter of the screwdriver tip is too small, the required tightening torque cannot be achieved. To achieve the appropriate tightening torque shown in the previous page, use the following screwdriver or its equivalent (grip diameter: approximately 25mm).



<Reference>

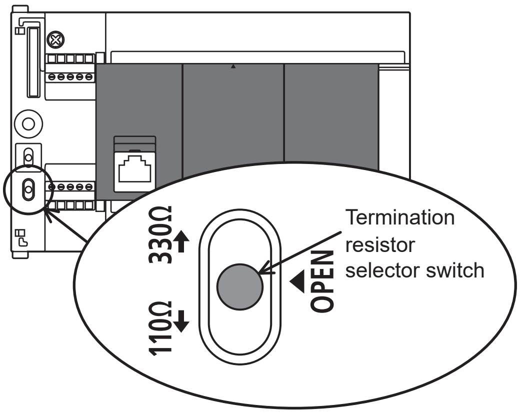
Manufacturer	Model
Phoenix Contact GmbH & Co. KG	SZS 0.4×2.5

Termination resistor setting

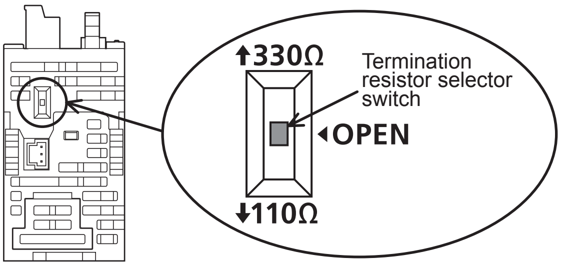
Make sure to provide a termination resistor at each end of a line.
Built-in RS-485 port, FX5-485-BD, and FX5-485ADP have a built-in termination resistor.
Set the termination resistor selector switch accordingly.

Wiring	Termination resistor selector switch
Two-pair wiring	330 Ω
One-pair wiring	110 Ω

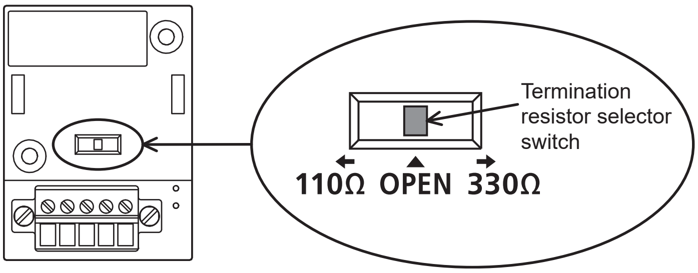
- FX5U CPU module built-in RS-485 port



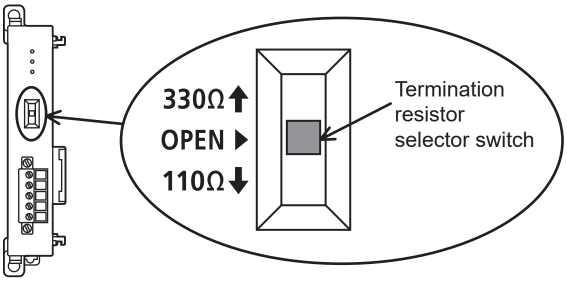
- FX5UC CPU module built-in RS-485 port



- FX5-485-BD



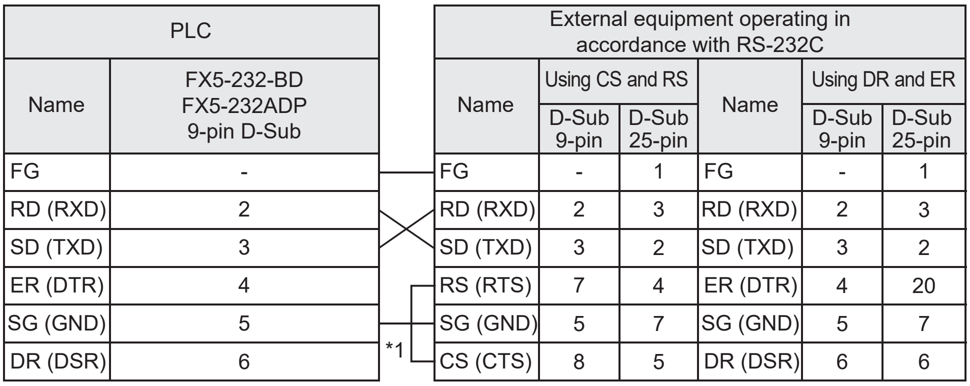
- FX5-485ADP



31.3 Connection Diagram

Representative wiring examples are shown in this section. When pin numbers in the counterpart equipment are different, wire the pins as shown below.

Connection diagram for RS-232C



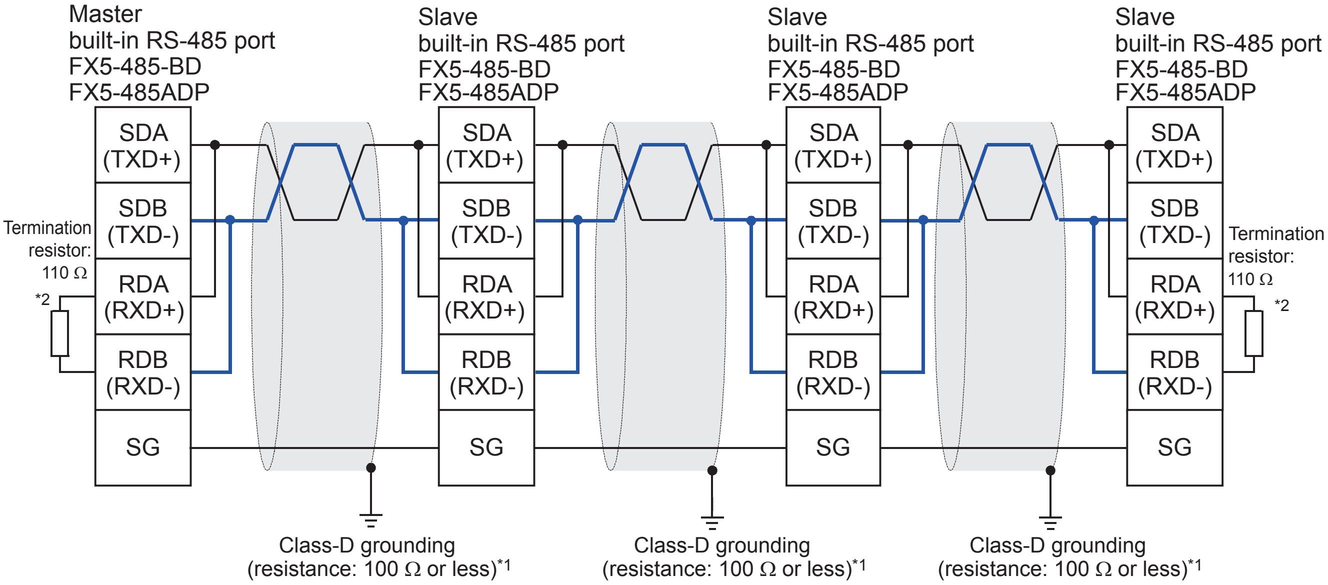
PLC	
Name	FX5-232-BD FX5-232ADP 9-pin D-Sub
FG	-
RD (RXD)	2
SD (TXD)	3
ER (DTR)	4
SG (GND)	5
DR (DSR)	6

External equipment operating in accordance with RS-232C					
Name	Using CS and RS		Name	Using DR and ER	
	D-Sub 9-pin	D-Sub 25-pin		D-Sub 9-pin	D-Sub 25-pin
FG	-	1	FG	-	1
RD (RXD)	2	3	RD (RXD)	2	3
SD (TXD)	3	2	SD (TXD)	3	2
RS (RTS)	7	4	ER (DTR)	4	20
SG (GND)	5	7	SG (GND)	5	7
CS (CTS)	8	5	DR (DSR)	6	6

*1 For third-party external equipment requiring the control signal, connect these pins.
The FX5-232-BD, FX5-232ADP does not require these pins to be connected.

Connection diagram for RS-485

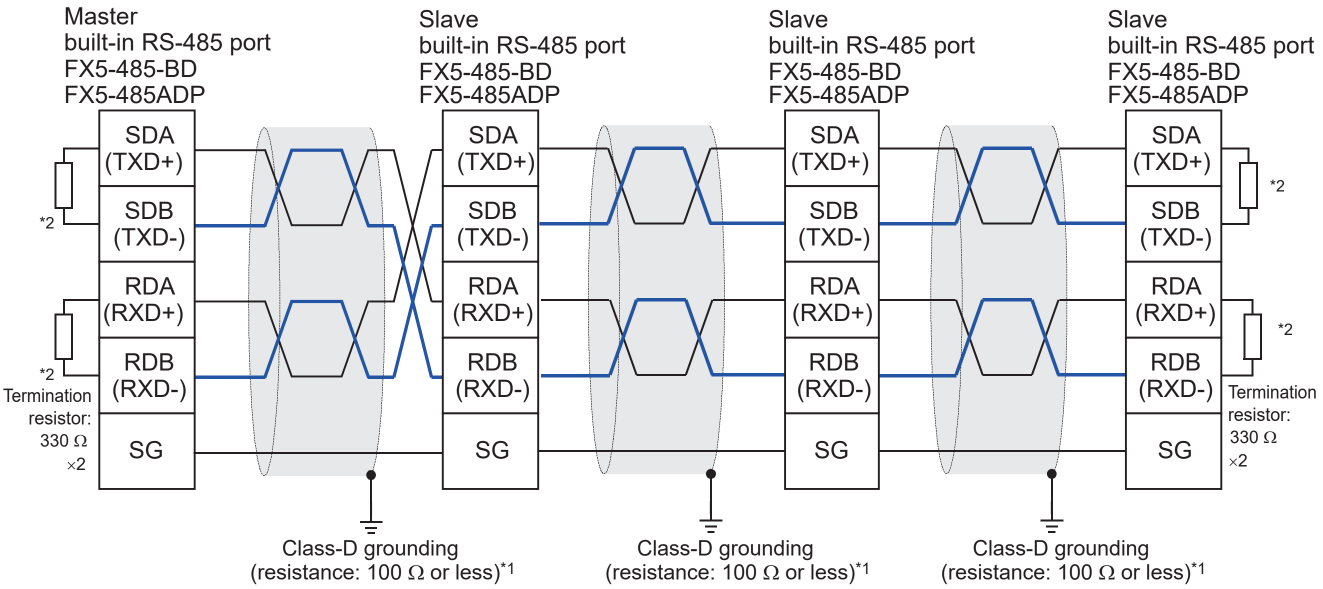
One-pair wiring



*1 Make sure to perform Class-D grounding on the shield of the twisted pair cable to be connected.

*2 Make sure to provide a termination resistor at each end of a line. Set the selector switch to 110 Ω when the termination resistor is built in.

Two-pair wiring



*1 Make sure to perform Class-D grounding on the shield of the twisted pair cable to be connected.

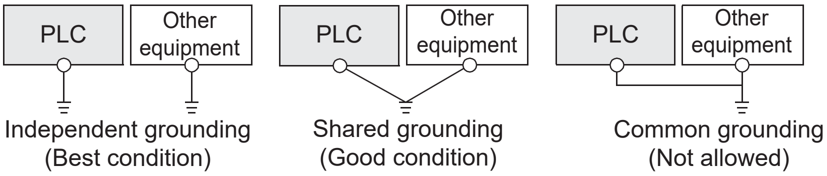
*2 Make sure to provide a termination resistor at each end of a line. Set the selector switch to 330 Ω when the termination resistor is built in.

31.4 Grounding

Grounding should be performed as stated below.

- Perform Class-D grounding. (Grounding resistance: 100Ω or less)
- Independent grounding should be performed for best results. If the PLC cannot be grounded independently, perform the "Shared grounding" shown below.

For details, refer to User's Manual (Hardware) of the CPU module used.



- The grounding wire size should be 14 AWG (2mm²) or larger.
- The grounding point should be close to the PLC, and all grounding wires should be as short as possible.

32 COMMUNICATION SETTING

This chapter explains the setup method for using MODBUS serial communication with an FX5.

32.1 Setup Method for MODBUS Serial Communication

For the MODBUS serial communication setting of the FX5, set parameters with GX Works3.
Setting of parameter differs according to the module used. The procedure for each module is as follows.

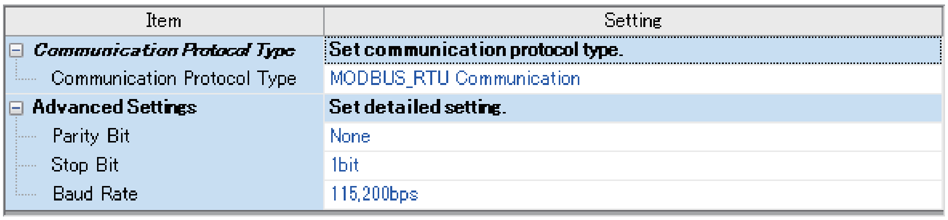
Using the CPU module

Navigation window ⇨ [Parameter] ⇨ Module model name ⇨ [Module Parameter] ⇨ [485 Serial Port]

Window

The following screen will be displayed if "MODBUS_RTU Communication" is set for Communication Protocol Type.

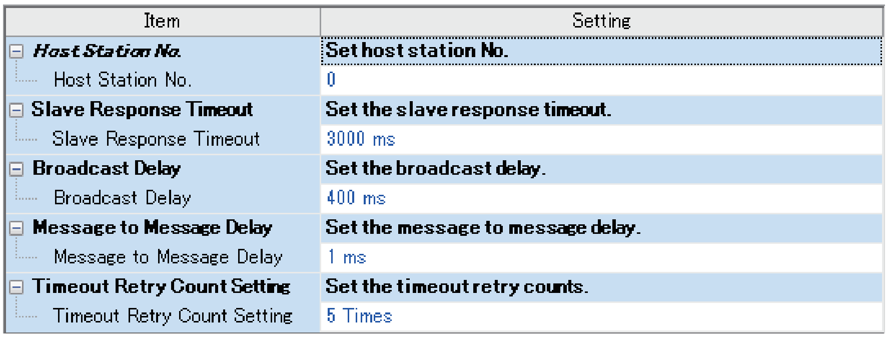
■Basic Settings



Item	Setting
⊟ *Communication Protocol Type*	Set communication protocol type.
Communication Protocol Type	MODBUS_RTU Communication
⊟ Advanced Settings	Set detailed setting.
Parity Bit	None
Stop Bit	1bit
Baud Rate	115,200bps

Item	Setting	Corresponding station
Communication Protocol Type	When using this function, select "MODBUS_RTU Communication".	Master/Slave
Parity Bit	None, Odd, Even	Master/Slave
Stop Bit	1bit, 2bit	Master/Slave
Baud Rate	300bps, 600bps, 1200bps, 2400bps, 4800bps, 9600bps, 19200bps, 38400bps, 57600bps, 115200bps	Master/Slave

■Fixed Setting



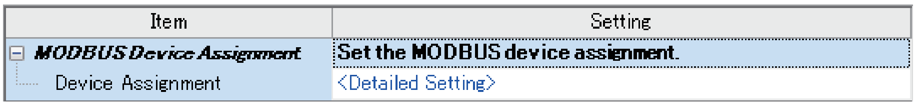
Item	Setting
⊟ *Host Station No.*	Set host station No.
Host Station No.	0
⊟ Slave Response Timeout	Set the slave response timeout.
Slave Response Timeout	3000 ms
⊟ Broadcast Delay	Set the broadcast delay.
Broadcast Delay	400 ms
⊟ Message to Message Delay	Set the message to message delay.
Message to Message Delay	1 ms
⊟ Timeout Retry Count Setting	Set the timeout retry counts.
Timeout Retry Count Setting	5 Times

Item	Setting	Corresponding station
Host Station No.[*1]	0 to 247 (Master station: 0, Slave station: 1 to 247)	Master/Slave
Slave Response Timeout	1 to 32767ms	Master/Slave
Broadcast Delay[*2]	1 to 32767ms	Master/Slave
Message to Message Delay	1 to 16382ms	Master/Slave
Timeout Retry Count Setting	0 to 20 times	Master/Slave

*1 When set to "Latch" by SM/SD Setting, setting values of the Host Station No. can be changed through special registers. (Refer to ☞ Page 537 Latch setting.) When other than 1 is set for the special register of a channel that is already set as a master station (station number: 0) by parameter, the channel does not function as a slave station. In addition, when 0 is set for the special register of a channel that is already set as a slave station (station number: 1 to 247) by parameter, the channel does not function as a master station.

*2 Set master station side broadcast delay as equal to one or more scan times of the slave station.

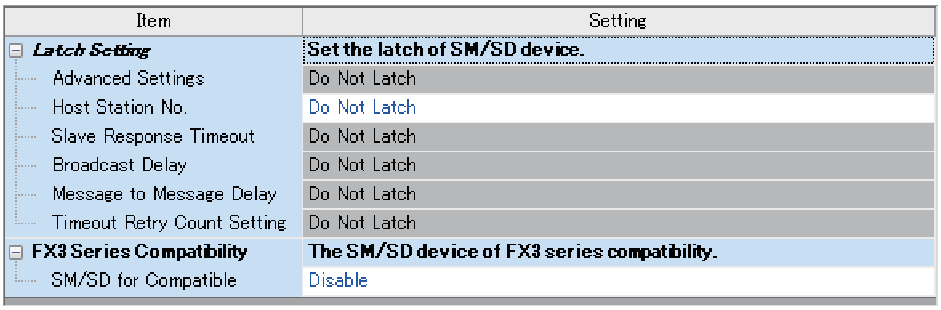
■MODBUS Device Assignment



Item	Setting
⊟ ***MODBUS Device Assignment***	**Set the MODBUS device assignment.**
Device Assignment	<Detailed Setting>

Item	**Setting**	**Corresponding station**
Device Assignment	Refer to the following. ☞ Page 535	Slave

■SM/SD Setting



Item	Setting
⊟ ***Latch Setting***	**Set the latch of SM/SD device.**
Advanced Settings	Do Not Latch
Host Station No.	Do Not Latch
Slave Response Timeout	Do Not Latch
Broadcast Delay	Do Not Latch
Message to Message Delay	Do Not Latch
Timeout Retry Count Setting	Do Not Latch
⊟ **FX3 Series Compatibility**	**The SM/SD device of FX3 series compatibility.**
SM/SD for Compatible	Disable

Item	**Setting**	**Corresponding station**
Advanced Settings	Do Not Latch	—
Host Station No.	Do Not Latch, Latch	Master/Slave
Slave Response Timeout	Do Not Latch	—
Broadcast Delay	Do Not Latch	—
Message to Message Delay	Do Not Latch	—
Timeout Retry Count Setting	Do Not Latch	—
SM/SD for Compatible	Disable/CH1/CH2 (☞ Page 537 SM/SD for FX3 Series compatible)	Master/Slave

For Latch Setting, refer to ☞ Page 537 Latch setting.

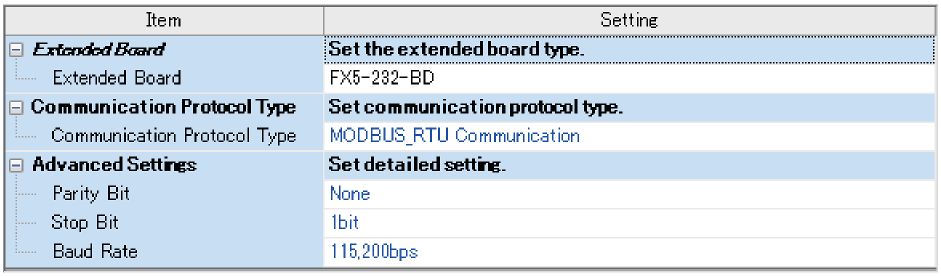
Using an extended board

Navigation window ⇨ [Parameter] ⇨ Module model name ⇨ [Module Parameter] ⇨ [Extended Board]

Window

The following screen will be displayed, if Extended Board to be used is set up and "MODBUS_RTU Communication" is set Communication Protocol Type.

■Basic Settings



Item	Setting
⊟ ***Extended Board***	**Set the extended board type.**
Extended Board	FX5-232-BD
⊟ **Communication Protocol Type**	**Set communication protocol type.**
Communication Protocol Type	MODBUS_RTU Communication
⊟ **Advanced Settings**	**Set detailed setting.**
Parity Bit	None
Stop Bit	1bit
Baud Rate	115,200bps

Item	**Setting**	**Corresponding station**
Extended Board	When using this function, select "FX5-232-BD" or "FX5-485-BD".	Master/Slave
Communication Protocol Type	When using this function, select "MODBUS_RTU Communication".	Master/Slave

All screens and setting fields other than extended board are the same as "Using the CPU module". (Refer to ☞ Page 533 Using the CPU module.)

Using an expansion adapter

When an expansion adapter is used, add expansion adapter to Module Information.

Navigation window ⇨ [Parameter] ⇨ [Module Information] ⇨ Right-click ⇨ [Add New Module]

After adding the expansion adapter, make settings on the screen displayed from the following operation.

Navigation window ⇨ [Parameter] ⇨ [Module Information] ⇨ [ADP1 to ADP6] ⇨ [Module Parameter]

Window

Each setting screen is the same as "Using the CPU module". (Refer to ☞ Page 533 Using the CPU module.)

Contents of parameter setting

The MODBUS device allocation, Latch Setting, SM/SD storage area specification which are used for parameter setting of MODBUS serial communication are as follows.

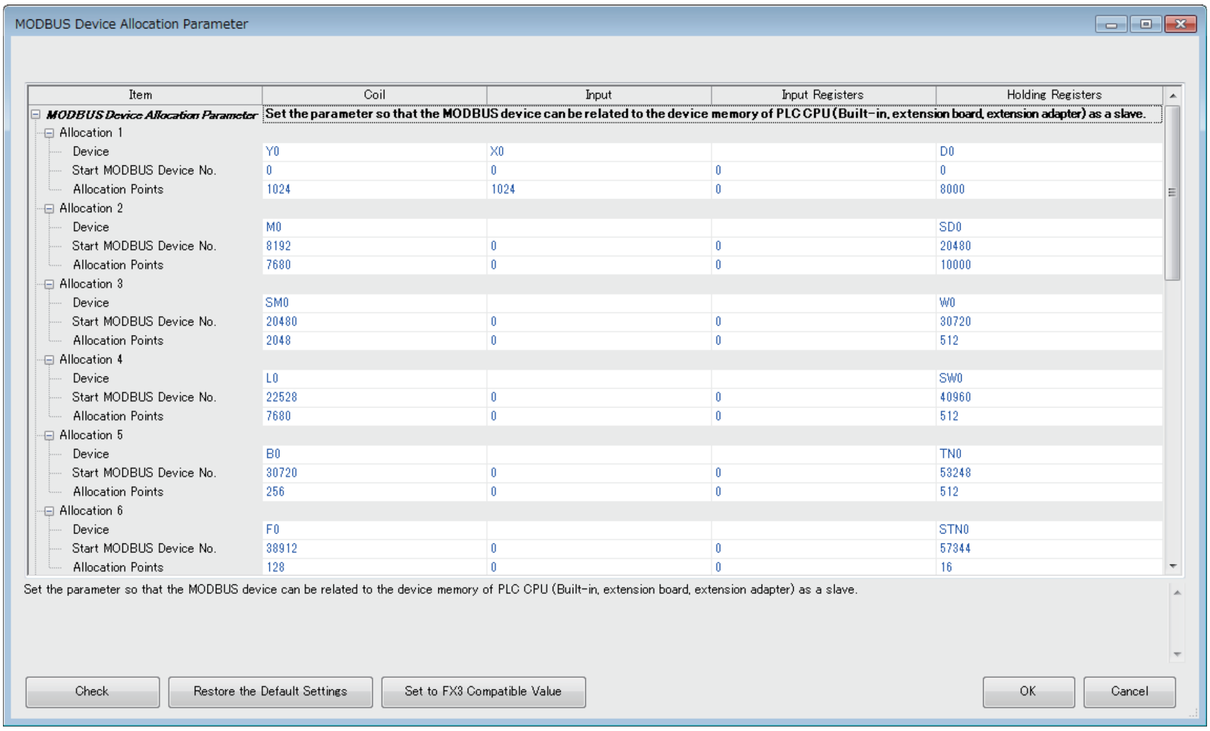
MODBUS device allocation

In the MODBUS device allocation, initial values are set for parameters. (Refer to ☞ Page 910 FX5 dedicated pattern.)

Parameter contents can be changed with following screen of GX Works3.

Window

<Detailed Setting> of "MODBUS Device Assignment" (☞ Page 534 MODBUS Device Assignment) is double-clicked.



Item	Coil	Input	Input Registers	Holding Registers
MODBUS Device Allocation Parameter	**Set the parameter so that the MODBUS device can be related to the device memory of PLC CPU (Built-in, extension board, extension adapter) as a slave.**			
Allocation 1				
Device	Y0	X0		D0
Start MODBUS Device No.	0	0	0	0
Allocation Points	1024	1024	0	8000
Allocation 2				
Device	M0			SD0
Start MODBUS Device No.	8192	0	0	20480
Allocation Points	7680	0	0	10000
Allocation 3				
Device	SM0			W0
Start MODBUS Device No.	20480	0	0	30720
Allocation Points	2048	0	0	512
Allocation 4				
Device	L0			SW0
Start MODBUS Device No.	22528	0	0	40960
Allocation Points	7680	0	0	512
Allocation 5				
Device	B0			TN0
Start MODBUS Device No.	30720	0	0	53248
Allocation Points	256	0	0	512
Allocation 6				
Device	F0			STN0
Start MODBUS Device No.	38912	0	0	57344
Allocation Points	128	0	0	16

Set the parameter so that the MODBUS device can be related to the device memory of PLC CPU (Built-in, extension board, extension adapter) as a slave.

Setting item	Description
Allocation 1 to 16	Allocation of each MODBUS device can be set to 1 to 16.
Device	Set the device type and head number of the device to be allocated. (For available devices, refer to ☞ Page 536 Available devices.)
Start MODBUS Device No.	Set the start MODBUS device number.
Allocation Points	Set the number of allocation points.

■Available devices

The following table shows devices that can be set to coil, input, input register, and holding register.

List of devices			Allocable MODBUS device			
Device type		**Device**	**Coil**	**Input**	**Input register**	**Holding register**
Special relay		SM	○	○	○[*1]	○[*1]
Special register		SD	—	—	○	○
Input		X	○	○	○[*1]	○[*1]
Output		Y	○	○	○[*1]	○[*1]
Internal relay		M	○	○	○[*1]	○[*1]
Latch relay		L	○	○	○[*1]	○[*1]
Annunciator		F	○	○	○[*1]	○[*1]
Link relay		B	○	○	○[*1]	○[*1]
Data register		D	—	—	○	○
Link register		W	—	—	○	○
Timer	Coil	TC	○	○	○[*1]	○[*1]
	Contact	TS	○	○	○[*1]	○[*1]
	Current value	TN	—	—	○	○
Retentive timer	Coil	STC	○	○	○[*1]	○[*1]
	Contact	STS	○	○	○[*1]	○[*1]
	Current value	STN	—	—	○	○
Counter	Coil	CC	○	○	○[*1]	○[*1]
	Contact	CS	○	○	○[*1]	○[*1]
	Current value	CN	—	—	○	○
Long counter	Coil	LCC	○	○	○[*1]	○[*1]
	Contact	LCS	○	○	○[*1]	○[*1]
	Current value	LCN	—	—	○[*2]	○[*2]
Link special relay		SB	○	○	○[*1]	○[*1]
Link special register		SW	—	—	○	○
Step relay		S	○	○	○[*1]	○[*1]
Index register		Z	—	—	○	○
Index register		LZ	—	—	○[*2]	○[*2]
File register		R	—	—	○	○
Extended file register		ER	—	—	—	—

*1 Set the device number and the allocating points in multiples of 16. In the case that the number of points is not a multiple of 16, a parameter setting error occurs in GX Works3.

*2 Because MODBUS devices are long type devices, 2 points are allocated.

Precautions

- The same device cannot be set for a coil and an input.
- The same device cannot be set for an input register and a holding register.
- A parameter setting error occurs in GX Works3 when the total of selected head device number and the allocated points exceed the valid range for the selected PLC device.

Latch setting

In latch setting, it can be set up whether the host number should operate using the GX Works3 parameter or special register.

- In the case of "Do Not Latch", the host number operates using the value set by the parameter setting of GX Works3.
- In the case of "Latch", special relays corresponding to each CH turn on and the host number operates using the value set by special registers. Value of special registers can be changed by program. The following table shows special relays and special registers corresponding for each channel.

CH	Special relays	Special registers	Corresponding parameter
CH1	SM8861	SD8861	Host station number setting
CH2	SM8871	SD8871	
CH3	SM8881	SD8881	
CH4	SM8891	SD8891	

Point

Setting value of special registers or parameter are reflected when the power supply is turned from off to on or reset.

Precautions

When the SD latch setting valid information is set to off due to memory clear and so on, while the parameter is set to "Latch", the parameter setting becomes valid when the power is turned off to on or reset.

SM/SD for FX3 Series compatible

In the FX5, special relays of the FX3 and special registers of the FX3 can be used in the compatibility area. Channel numbers may differ between the FX3 and the FX5 depending on the configuration. Select whether to use special devices for the corresponding channel number (CH1 or CH2) in this setting.

- When this setting is not configured, SM and SD for FX3 series compatibility are not used.
- When this setting is configured, select CH1 or CH2 SM and SD for FX3 series compatibility.

For the device for FX3 series compatible, refer to the following.

☞ Page 541 Related Devices

33 FUNCTION

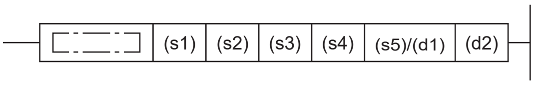
This chapter explains the function of MODBUS serial communication.

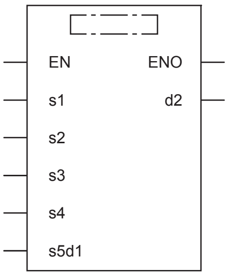
33.1 Master Function

In the FX5 master function, communication is executed with the slave station using the ADPRW instruction.

ADPRW

This instruction allows to communicate (read/write data) with the slave station by the function code which is supported by the master.

Ladder diagram	Structured text
	ENO:=ADPRW (EN, s1, s2, s3, s4, s5d1, d2);

FBD/LD


Setting data

■Descriptions, ranges, data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Slave station No.	0 to F7H[*1]	16-bit signed binary	ANY16
(s2)	Function code (Refer to ☞ Page 539)	01H to 06H, 0FH, 10H	16-bit signed binary	ANY16
(s3)	Function parameter depending on the function code (Refer to ☞ Page 539)	0 to FFFFH	16-bit signed binary	ANY16
(s4)	Function parameter depending on the function code (Refer to ☞ Page 539)	1 to 2000	16-bit signed binary	ANY16
(s5)/(d1)	Function parameter depending on the function code (Refer to ☞ Page 539)	—	Bit/16-bit signed binary	ANY_ELEMENTARY
(d2)[*2]	Start bit device number to which communication execution status is output	—	Bit	ANYBIT_ARRAY (Number of elements: 3)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 Slave station No. that can be used in FX5 master station differs depending on the version. (☞ Page 936 Added and Changed Functions)

*2 Three devices are occupied from the device specified in (d2). Make sure that these devices are not used in other controls.

■Available 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	$	
(s1)	—	○*1	○	○	—	—	○	○	—	—	—
(s2)	—	○*1	○	○	—	—	○	○	—	—	—
(s3)	—	○*1	○	○	—	—	○	○	—	—	—
(s4)	—	○*1	○	○	—	—	○	○	—	—	—
(s5)/(d1)	○	○*1	○	○	—	—	○	○	—	—	—
(d2)	○	○*1	—	—	—	—	—	—	—	—	—

*1 T, ST, C cannot be used.

Processing details

- Function code (s2) is operated on slave station No. (s1) according to parameters (s3), (s4), and (s5)/(d1). Use 0 as the slave station No. (s1) for broadcast commands. (Refer to ☞ Page 539 Function code and function parameters.)
- The communication execution status (d2) is output according to each status of the ADPRW instruction such as communicating/completed normally/completed with an error. (Refer to ☞ Page 540 Communication execution status output device.)

■Function code and function parameters

The following table shows the parameter allocation of (s3), (s4), and (s5)/(d1) for each function code (s2).

(s2): Function code	(s3): MODBUS address	(s4): Access points	(s5)/(d1): Start device storing data	
	Applicable device: ❷ (Refer to the following table of the applicable devices.)			
01H Read coils	MODBUS address: 0000H to FFFFH	Access points: 1 to 2000	Start device storing read data	
			Applicable device	Word device ❶ (Refer to the following table of the applicable devices.) Bit device ❸ (Refer to the following table of the applicable devices.)
			Number of occupied points	Word device ((s4) + 15) ÷ 16 points*1 Bit device (s4) points
02H Read inputs	MODBUS address: 0000H to FFFFH	Access points: 1 to 2000	Start device storing read data	
			Applicable device	Word device ❶ (Refer to the following table of the applicable devices.) Bit device ❸ (Refer to the following table of the applicable devices.)
			Number of occupied points	Word device ((s4) + 15) ÷ 16 points*1 Bit device (s4) points
03H Read holding registers	MODBUS address: 0000H to FFFFH	Access points: 1 to 125	Start device storing read data	
			Applicable device	❶ (Refer to the following table of the applicable devices.)
			Number of occupied points	(s4) points
04H Read input registers	MODBUS address: 0000H to FFFFH	Access points: 1 to 125	Start device storing read data	
			Applicable device	❶ (Refer to the following table of the applicable devices.)
			Number of occupied points	(s4) points

(s2): Function code	(s3): MODBUS address	(s4): Access points	(s5)/(d1): Start device storing data	
	Applicable device: ❷ (Refer to the following table of the applicable devices.)			
05H Write coils	MODBUS address: 0000H to FFFFH	0 (fixed)	Start device storing write data	
			Applicable device*2	Word device ❷ (Refer to the following table of the applicable devices.) Bit device ❸ (Refer to the following table of the applicable devices.)
			Number of occupied points	1 point
06H Write holding registers	MODBUS address: 0000H to FFFFH	0 (fixed)	Start device storing write data	
			Applicable device	❷ (Refer to the following table of the applicable devices.)
			Number of occupied points	1 point
0FH Write multiple coils	MODBUS address: 0000H to FFFFH	Access points: 1 to 1968	Start device storing write data	
			Applicable device	Word device ❷ (Refer to the following table of the applicable devices.) Bit device ❸ (Refer to the following table of the applicable devices.)
			Number of occupied points	Word device ((s4) + 15) ÷ 16 points*1 Bit device (s4) points
10H Write multiple holding registers	MODBUS address: 0000H to FFFFH	Access points: 1 to 123	Start device storing write data	
			Applicable device	❷ (Refer to the following table of the applicable devices.)
			Number of occupied points	(s4) points

*1 Fractions are rounded off.

*2 When the least significant bit is 0, the bit is off. When the least significant bit is 1, the bit is on.

▶ Applicable device table

No.	Applicable device
❶	T, ST, C, D, R, W, SW, SD, label device
❷	T, ST, C, D, R, W, SW, SD, label device, K, H
❸	X, Y, M, L, B, F, SB, S, SM, label device

■Communication execution status output device

The following shows the timing of operation of the communication execution status output device (d2) according to each communication status, and the special relays which operate at the same time.

Operand	Timing of operation	Special relay which operates at the same time
(d2)	Turns on while the instruction is executed and turns off in the status other than while the instruction is executed.	SM8800 (CH1), SM8810 (CH2), SM8820 (CH3), SM8830 (CH4)*1
(d2) + 1*2	Turns on when the instruction is completed normally and turns off when the communication is started.	—
(d2) + 2*2	Turns on when the instruction is completed with an error and turns off when the communication is started.	—

*1 SM8401 (CH1) or SM8421 (CH2) turn on, when SM/SD for FX3 series compatible is set.

*2 (d2) + 1 turns on when the instruction is completed normally and (d2) + 2 turns on when the instruction is completed with an error, so that whether the instruction is completed normally or with an error can be judged.

Precautions

- The channel which uses the ADPRW instruction must be set to the MODBUS master station by GX Works3. (☞ Page 533 Fixed Setting) When it is not set, the device does not operate even though the ADPRW instruction is executed. (Also no error occurs.)
- If the program stops due to error, the device value is turned off if a non-latch device is designated as the communication execution status output device. Specify a latch device to keep the communication state output.

33.2 Slave Function

This function performs operations according to the supported function code by communication with the master station.
For the supported function codes, refer to ☞ Page 526 List of supported MODBUS standard functions.

33.3 Related Devices

In this section, the functions of the special relays and special registers are described for MODBUS serial communication.

Point

Available communication channels differs depending on the CPU module and system configuration.
For communication channels, refer to ☞ Page 518 System Configuration.
The devices of "For FX3 compatibility" operate in the channel which is specified by the SM/SD for compatibility in communication setting.
For compatible SM/SD, refer to ☞ Page 533 COMMUNICATION SETTING.

List of related devices

Special relays

The table shows the special relays used for the FX5 MODBUS serial communication.

■Only for FX5

R: Read only, R/W: Read/write

Device number				Name	Valid	Details	R/W
CH1	CH2	CH3	CH4				
SM8500	SM8510	SM8520	SM8530	Serial communication error	Master/ Slave	Turns on when an error occurs during the serial communication.	R
SM8800	SM8810	SM8820	SM8830	During MODBUS RTU communication	Master	Turns on when the instruction is started until the instruction execution complete flag is turned on during MODBUS serial communication.	R
SM8801	SM8811	SM8821	SM8831	Retry	Master	Turns on while the master sends retries when the slave fails to respond until timeout setting time.	R
SM8802	SM8812	SM8822	SM8832	Timeout	Master	Turns on if a response timeout occurs.	R
SM8861	SM8871	SM8881	SM8891	Host station number latch setting valid	Slave	Turns on when the latch setting is set to "Latch".	[1]

[1] The devices vary depending on the latch setting. The device becomes R when the Latch setting is "Do not Latch", and R/W when the latch setting is "Latch".

■For FX3 series compatibility

R: Read only

Device number		Name	Valid	Details	R/W
CH1	CH2				
SM8029		Instruction execution complete	Master	Turns on if the processing of an instruction is completed.	R
SM8401	SM8421	During MODBUS communication	Master	Turns on when the instruction is started until the instruction execution complete flag is turned on during MODBUS serial communication.	R
SM8402	SM8422	MODBUS communication error	Master	Turns on when an error occurs during MODBUS serial communication.	R
SM8403	SM8423	MODBUS communication error (latched)	Master/ Slave	Turns on once an error occurs during MODBUS serial communication.	R
SM8063	SM8438	Serial communication error	Master/ Slave	Turns on once an error occurs during MODBUS serial communication.	R
SM8408	SM8428	Retry	Master	Turns on while the master sends retries when the slave fails to respond until timeout setting time.	R
SM8409	SM8429	Timeout	Master	Turns on if a response timeout occurs.	R

Special registers

The table shows the special registers used for the FX5 MODBUS serial communication.

■Only for FX5

R: Read only, R/W: Read/write

Device number				Name	Valid	Details	R/W
CH1	CH2	CH3	CH4				
SD8500	SD8510	SD8520	SD8530	Serial communication error code	Master/ Slave	Stores the current error code generated during serial communication.	R
SD8501	SD8511	SD8521	SD8531	Serial communication error details	Master/ Slave	Stores current error details.	R
SD8502	SD8512	SD8522	SD8532	Serial communication setting	Master/ Slave	Stores the communication properties in the CPU module.	R
SD8503	SD8513	SD8523	SD8533	Serial communication operation mode	Master/ Slave	Stores the mode of serial communication being executed.	R
SD8800	SD8810	SD8820	SD8830	Current retry value	Master/ Slave	Stores the current value of retries of when a communication retry is executed due to slave response timeout.	R
SD8861	SD8871	SD8881	SD8891	Host station number	Master/ Slave	Stores the host station number setting value.	[*1]
SD8862	SD8872	SD8882	SD8892	Slave response timeout	Master/ Slave	Stores the slave response timeout setting value.	R
SD8863	SD8873	SD8883	SD8893	Broadcast delay	Master/ Slave	Stores the broadcast delay setting value.	R
SD8864	SD8874	SD8884	SD8894	Message to message delay	Master/ Slave	Stores the message to message delay setting value.	R
SD8865	SD8875	SD8885	SD8895	Timeout retry count	Master/ Slave	Stores the timeout retry count setting value.	R

*1 The devices vary depending on the latch setting. The device becomes R when the Latch setting is "Do not Latch", and R/W when the latch setting is "Latch".

■For FX3 series compatibility

R: Read only

Device number		Name	Valid	Details	R/W
CH1	CH2				
SD8063	SD8438	Serial communication error code	Master/ Slave	Stores the current error code generated during serial communication.	R
SD8402	SD8422	Communication error code	Master/ Slave	Stores the current error code generated during serial communication.	R
SD8403	SD8423	Error details	Master/ Slave	Stores the current error details.	R
SD8405	SD8425	Communication format setting value	Master/ Slave	Stores the communication parameter set in the CPU module.	R
SD8408	SD8428	Current retry value	Master/ Slave	Stores the current value of retries of when a communication retry is executed due to slave response timeout.	R
SD8419	SD8439	Communication operation mode	Master/ Slave	Stores the communication operation mode in the CPU module.	R

Details of related devices

Instruction execution complete

This device checks whether the execution of the instruction is completed.

R: Read only

Only for FX5				For FX3 series compatibility		Description	R/W
CH1	CH2	CH3	CH4	CH1	CH2		
SM8029						Turns on if the processing of the instruction is completed.	R

Point

SM8029 is also used as the execution completed flag for other instructions (such as positioning instructions). When using SM8029, provide the contact just under the instruction whose execution completion is to be checked.

Precautions

Do not turn ON with program or engineering tool.

The device is cleared when the power supply is turned from off to on, reset, set from STOP to RUN, or the next ADPRW instruction is executed.

During MODBUS communication

This device checks if MODBUS serial communication is being executed.

R: Read only

Only for FX5				For FX3 series compatibility		Description	R/W
CH1	CH2	CH3	CH4	CH1	CH2		
SM8800	SM8810	SM8820	SM8830	SM8401	SM8421	Turns on when the instruction is started until the instruction execution complete flag is turned on during MODBUS serial communication.	R

Precautions

Do not turn ON with program or engineering tool.

The device is cleared when the power supply is turned from off to on, reset, or set from STOP to RUN.

MODBUS communication error

This device checks if an error occurs during MODBUS serial communication.

R: Read only

For FX3 series compatibility		Description	R/W
CH1	CH2		
SM8402	SM8422	Turns on when an error occurs during MODBUS serial communication.	R

Precautions

Do not turn ON with program or engineering tool.

This device does not turn OFF even if normal communication is restored. The device is cleared when the power supply is turned from off to on, reset, set from STOP to RUN, SM50 (Error Detection Reset Completion) is turned on, or the next ADPRW instruction is executed.

MODBUS communication error (latched)

This device checks if an error occurs during MODBUS serial communication.

R: Read only

For FX3 series compatibility		Description	R/W
CH1	CH2		
SM8403	SM8423	Turns on once an error occurs during MODBUS serial communication.	R

Precautions

Do not turn ON with program or engineering tool.

The device is cleared when the power supply is turned from off to on, reset, or set from STOP to RUN.

Serial communication error

This device checks if an error occurs during serial communication.

R: Read only

Only for FX5				For FX3 series compatibility		Description	R/W
CH1	CH2	CH3	CH4	CH1	CH2		
SM8500	SM8510	SM8520	SM8530	SM8063	SM8438	Turns on when an error occurs during serial communication.	R

Precautions

Do not turn ON with program or engineering tool.

This device does not turn OFF even if normal communication is restored. The device is cleared when the power supply is turned from off to on, reset, set from STOP to RUN, or SM50 (Error Detection Reset Completion) is turned on.

Retry

This device checks if a retry occurs during MODBUS serial communication.

R: Read only

Only for FX5				For FX3 series compatibility		Description	R/W
CH1	CH2	CH3	CH4	CH1	CH2		
SM8801	SM8811	SM8821	SM8831	SM8408	SM8428	Turns on while the master sends retries when the slave fails to respond in time.	R

Precautions

Do not turn ON with program or engineering tool.

The device is cleared when the power supply is turned from off to on, reset, set from STOP to RUN, SM50 (Error Detection Reset Completion) is turned on, or the next ADPRW instruction is executed.

Timeout

This device checks if a timeout occurs during MODBUS serial communication.

R: Read only

Only for FX5				For FX3 series compatibility		Description	R/W
CH1	CH2	CH3	CH4	CH1	CH2		
SM8802	SM8812	SM8822	SM8832	SM8409	SM8429	Turns on if a response timeout occurs.	R

Precautions

Do not turn ON with program or engineering tool.

The device is cleared when the power supply is turned from off to on, reset, set from STOP to RUN, SM50 (Error Detection Reset Completion) is turned on, or the next ADPRW instruction is executed.

If the number of retries is 1 or more, the error flag (☞ Page 850 Error flags) is not set until the set number of retries fail by timeout (or another failure).

Host station number latch setting valid

The device which set latch valid/invalid of host station number for MODBUS serial communication.

R: Read only, R/W: Read/write

Only for FX5				Description	R/W
CH1	CH2	CH3	CH4		
SM8861	SM8871	SM8881	SM8891	Turns on when host station number setting is "Latch" in MODBUS communication parameter.	[1]

[1] The devices vary depending on the Latch Setting. The device becomes R when the Latch setting is "Do not Latch", and R/W when the Latch setting is "Latch".

Precautions

Do not turn ON with program or engineering tool.

The device is set when the power supply is turned from off to on or reset.

Serial communication error code

This device stores the current error codes during serial communication. (☞ Page 850 MODBUS Serial Communication)

R: Read only

Only for FX5				For FX3 series compatibility		Description	R/W
CH1	CH2	CH3	CH4	CH1	CH2		
SD8500	SD8510	SD8520	SD8530	SD8402 SD8063	SD8422 SD8438	Stores the current error code generated by serial communication.	R

Precautions

Do not change the device value using a program or an engineering tool.

The device is cleared when the power supply is turned from off to on, reset, set from STOP to RUN, or SM50 (Error Detection Reset Completion) is turned on, only in the master.

Serial communication error details

This device stores the current error details during serial communication. (☞ Page 850 MODBUS Serial Communication)

R: Read only

Only for FX5				For FX3 series compatibility		Description	R/W
CH1	CH2	CH3	CH4	CH1	CH2		
SD8501	SD8511	SD8521	SD8531	SD8403	SD8423	Stores the current error details generated during serial communication.	R

Precautions

Do not change the device value using a program or an engineering tool.

The device is cleared when the power supply is turned from off to on, reset, set from STOP to RUN, or SM50 (Error Detection Reset Completion) is turned on, only in the master.

Communication format setting value

This device stores the communication format setting value.

R: Read only

Only for FX5				For FX3 series compatibility		Description	R/W
CH1	CH2	CH3	CH4	CH1	CH2		
SD8502	SD8512	SD8522	SD8532	SD8405	SD8425	Stores the parameters set by an engineering tool. For details, refer to the following table.	R

The following table shows the parameter descriptions of the communication format.

Bit	Name	Description	
		0 (bit = OFF)	1 (bit = ON)
b0	—	—	—
b1, b2	Parity bit	(b2, b1) = (0, 0): None (b2, b1) = (0, 1): Odd (b2, b1) = (1, 1): Even	
b3	Stop bit	1bit	2bit
b4 to b7	Baud rate (bps)	(b7, b6, b5, b4) = (0, 0, 1, 1): 300 (b7, b6, b5, b4) = (0, 1, 0, 0): 600 (b7, b6, b5, b4) = (0, 1, 0, 1): 1200 (b7, b6, b5, b4) = (0, 1, 1, 0): 2400 (b7, b6, b5, b4) = (0, 1, 1, 1): 4800 (b7, b6, b5, b4) = (1, 0, 0, 0): 9600 (b7, b6, b5, b4) = (1, 0, 0, 1): 19200 (b7, b6, b5, b4) = (1, 0, 1, 0): 38400 (b7, b6, b5, b4) = (1, 0, 1, 1): 57600 (b7, b6, b5, b4) = (1, 1, 0, 1): 115200	
b8 to b15	—	—	—

Precautions

Do not change the device value using a program or an engineering tool.

The setting value changes when the power supply is turned from off to on or reset.

Operation mode display

This device stores the operation mode of the serial communication being executed.

R: Read only

Only for FX5				For FX3 series compatibility		Description	R/W
CH1	CH2	CH3	CH4	CH1	CH2		
SD8503	SD8513	SD8523	SD8533	SD8419	SD8439	0: MELSOFT connection or MC protocol 3: N:N Network Communication 5: Non-Protocol Communication 6: Parallel Link Communication 7: Inverter Communication 9: MODBUS RTU Communication 12: Predefined protocol support Other than above: Not used	R

Precautions

Do not change the device value using a program or an engineering tool.

The device is cleared when ADPRW instruction is executed.

Current retry value

This device stores the current retry value.

R: Read only

Only for FX5				For FX3 series compatibility		Description	R/W
CH1	CH2	CH3	CH4	CH1	CH2		
SD8800	SD8810	SD8820	SD8830	SD8408	SD8428	Stores the current value of retries executed by the slave response timeout.	R

Precautions

Do not change the device value using a program or an engineering tool.

The device is cleared when the power supply is turned from off to on, reset, set from STOP to RUN, SM50 (Error Detection Reset Completion) is turned on, or the next ADPRW instruction is executed.

Host station number

This device stores the host station number setting value.

R: Read only, R/W: Read/write

Only for FX5				Description	R/W
CH1	CH2	CH3	CH4		
SD8861	SD8871	SD8881	SD8891	Stores the parameters (host station number) set by a program or an engineering tool. For 0: Master station For 1 to 247: Slave station (station number: 1 to 247)	[1]

[1] The devices vary depending on the Latch Setting. The device becomes R when the Latch setting is "Do not Latch", and R/W when the Latch setting is "Latch".

Precautions

■"Do Not latch"

Do not change the device value using a program or an engineering tool.

■"Latch"

The device value can be changed by the program.

■Change "Do Not Latch" to "Latch"

The setting value is reflected when the power supply is turned from on to off or reset, after parameter write from an engineering tool.

Slave response timeout

This device stores the slave response timeout setting value.

R: Read only

Only for FX5				Description	R/W
CH1	CH2	CH3	CH4		
SD8862	SD8872	SD8882	SD8892	Stores the parameters (slave response timeout) set by an engineering tool.	R

Precautions

The setting value is reflected when the power supply is turned from off to on, reset, or the next ADPRW instruction is executed.

Broadcast delay

This device stores the broadcast delay setting value.

R: Read only

Only for FX5				Description	R/W
CH1	CH2	CH3	CH4		
SD8863	SD8873	SD8883	SD8893	Stores the parameters (broadcast delay) set by an engineering tool.	R

Precautions

The setting value is reflected when the power supply is turned from off to on, reset, or the next ADPRW instruction is executed.

Message to message delay

This device stores the message to message delay setting value.

R: Read only

Only for FX5				Description	R/W
CH1	CH2	CH3	CH4		
SD8864	SD8874	SD8884	SD8894	Stores the parameters (message to message delay) set by an engineering tool.	R

Precautions

The setting value is reflected when the power supply is turned from off to on, reset, or the next ADPRW instruction is executed.

Timeout retry count

This stores the timeout retry count setting value.

R: Read only

Only for FX5				Description	R/W
CH1	CH2	CH3	CH4		
SD8865	SD8875	SD8885	SD8895	Stores the parameters (timeout retry count) set by an engineering tool.	R

Precautions

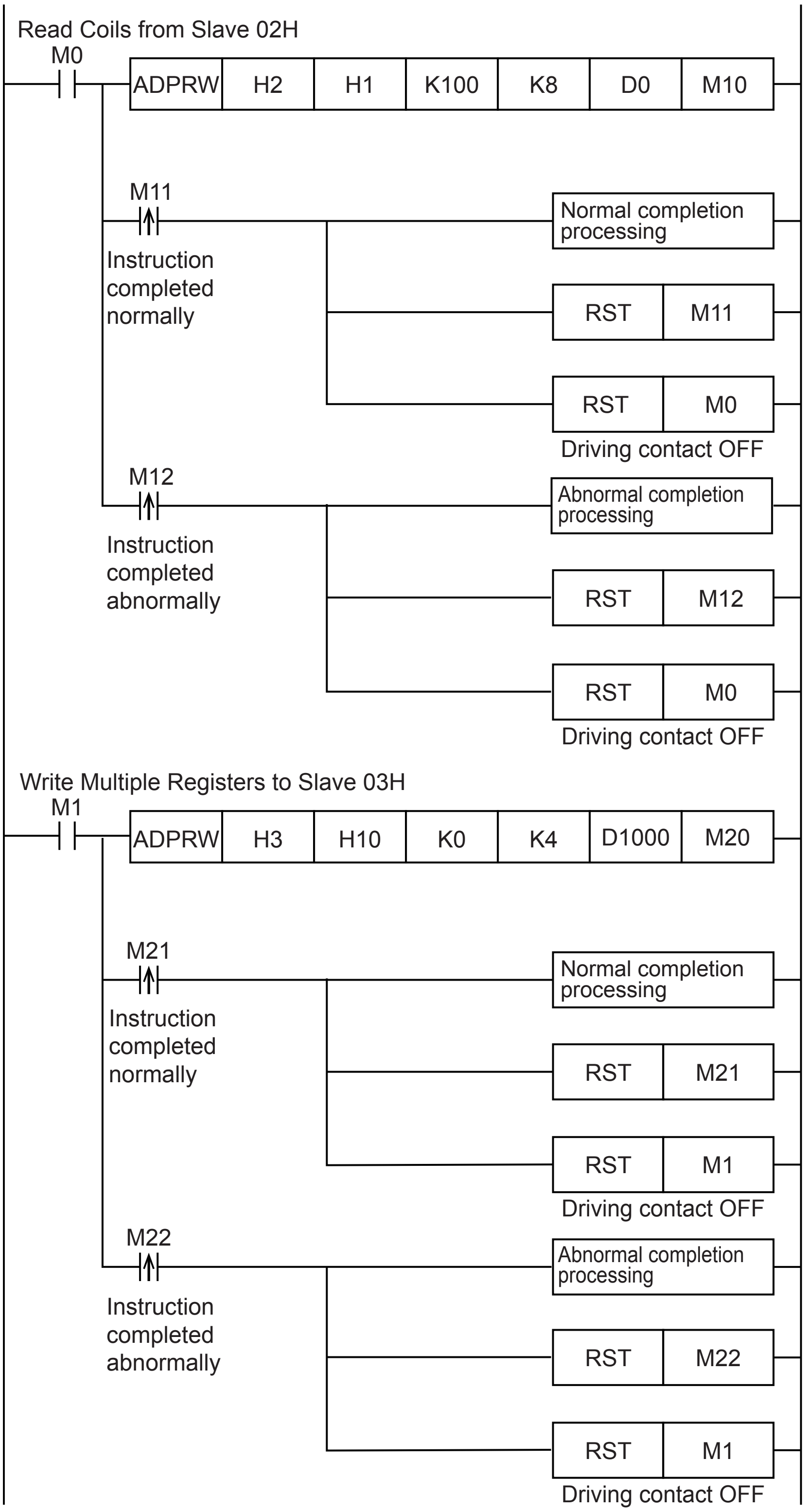
The setting value is reflected when the power supply is turned from off to on, reset, or the next ADPRW instruction is executed.

34 CREATING PROGRAMS

This chapter explains an example of creating programs for the master in MODBUS serial communication.

34.1 Creating Programs for the Master Station

Programs allowing the master station to read and write slave station devices can be created as shown in the example below.
For ADPRW instruction, refer to ☞ Page 538 ADPRW.
For cautions on program creation, refer to ☞ Page 551 Cautions on Program Creation.



Program for Reading Coils

Slave address: 02H
Function code: 01H
MODBUS address: 100
Device count: 8
Start device storing read data: D0

8 coil device values starting at MODBUS Address 100 of Slave 02H are read to the low-order 8 bits in D0 of the Master.

Program for Writing Multiple Registers

Slave address: 03H
Function code: 10H
MODBUS address: 0
Device count: 4
Start device storing write data: D1000

The value of D1000 to D1003 of the Master is written to MODBUS Address 0 and onward for Slave 03H.

34.2 Cautions on Program Creation

- Make sure the driving contact of the ADPRW instruction does not turn off until the instruction has been completed.
- For the ADPRW instruction, a start timing of communication differs depending on the condition at the time of driving. When the ADPRW instruction is driven alone, communication is instantaneously started. When multiple ADPRW instructions are simultaneously driven, after communication by the former driven ADPRW instruction is completed, the communication by the latter driven ADPRW instruction is started. Make sure the driving contact of the ADPRW instruction does not turn off until the instruction has been completed.
- When using the Read coils function or Read inputs function in the Master with a word device (i.e. D, or R) as the destination device, only the number of bits assigned in the device count of the ADPRW instruction will be overwritten. The remaining bits of the word device will not be affected.

MEMO

PART 5 SLMP

This part consists of the following chapters.

35 OUTLINE

This manual describes the compatible devices, access ranges, communication procedures, and message formats of the SLMP.

When transferring data using SLMP, always refer to ☞ Page 558 SLMP DATA COMMUNICATION.

35.1 Outline of SLMP

SLMP is a protocol used for access from an Ethernet-equipped module or an external device (such as a personal computer or an HMI) to an SLMP compatible device through Ethernet.

SLMP communications are available among devices that can transfer messages by SLMP.

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

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

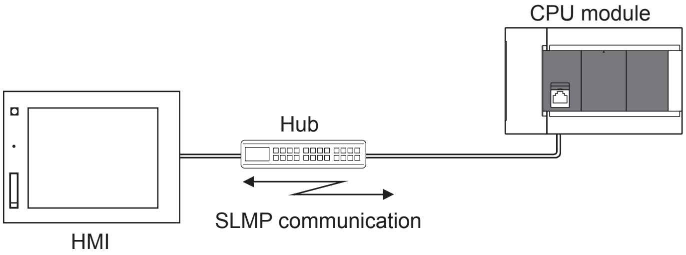
For the versions of modules that are compatible with 1E frames, refer to the following. ☞ Page 936 Added and Changed Functions

[Server function]

The CPU 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 is supported only for the CPU module, and the SLMP frame transmission is supported only for the 3E frame.



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.

For details on MC protocol, refer to the following manual.

📖MELSEC Communication Protocol Reference Manual

Applications

- Device data in an Ethernet-equipped module can be written or read from a personal computer or an HMI by using SLMP.
- Writing and reading the device allows operation monitoring, data analyzing, and production managing of an Ethernet-equipped module by a personal computer or an HMI.
- External illegal access can be prevented by the remote password function.

Data communication procedures

The following shows the flow for starting SLMP communication. For details, refer to the following.

☞ Page 90 SLMP FUNCTION

1. Connecting cables and external devices

Make the connections for SLMP communication.

2. Setting parameters

Configure the module parameters with the engineering tool.

3. Writing to the Ethernet-equipped module

Write the parameters set in the Ethernet-equipped module. Turn the power off and on or perform reset to enable the parameters.

4. Initial process state check

After setting the module parameter, check if initial process of the Ethernet-equipped module is completed normally.

5. SLMP communication[*1]

[Server function]

SLMP messages from external devices are received.

[Client function][*2]

SLMP messages are sent to external devices.

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

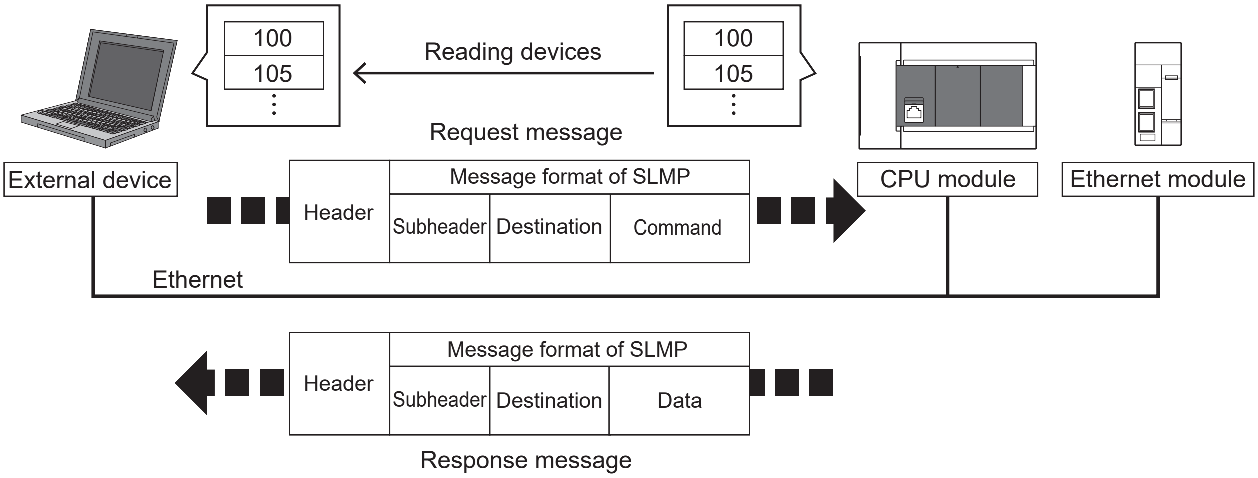
[*2] Only 3E frame of the CPU module is supported.

35.2 Features of SLMP

System monitoring from an external device (such as personal computer, HMI)

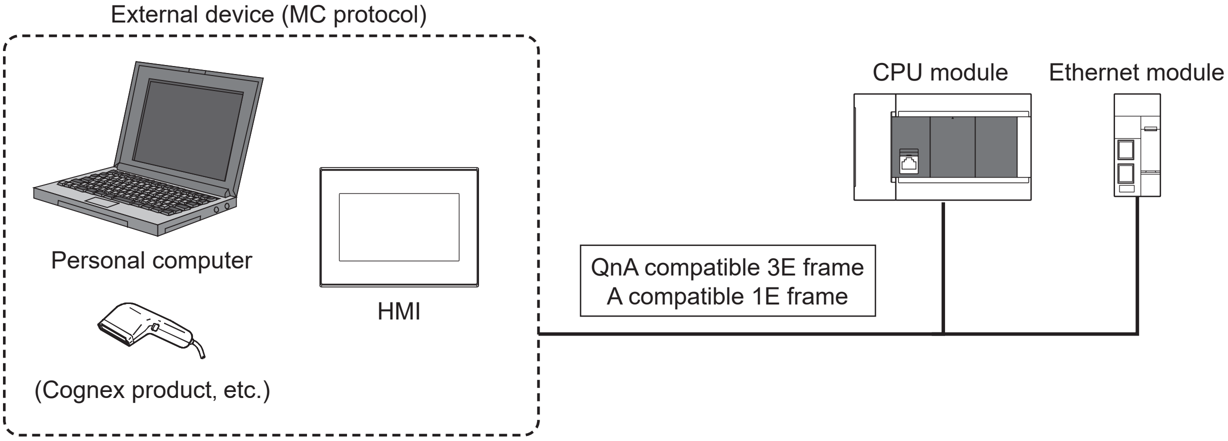
An external device can send a request message in SLMP message format to an Ethernet-equipped module to enable device read, allowing system monitoring.

Using SLMP allows not only device data reading but also device data writing and resetting an Ethernet-equipped module.



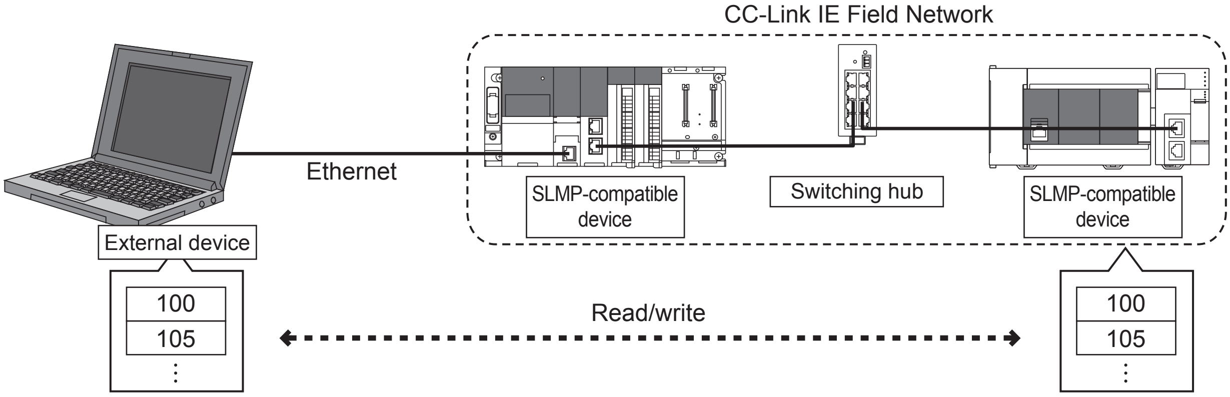
Connecting an external device used with MC protocol

An external device that uses the QnA compatible 3E frame of MC protocol and A compatible 1E frame of MC protocol can be connected to an Ethernet-equipped module directly.



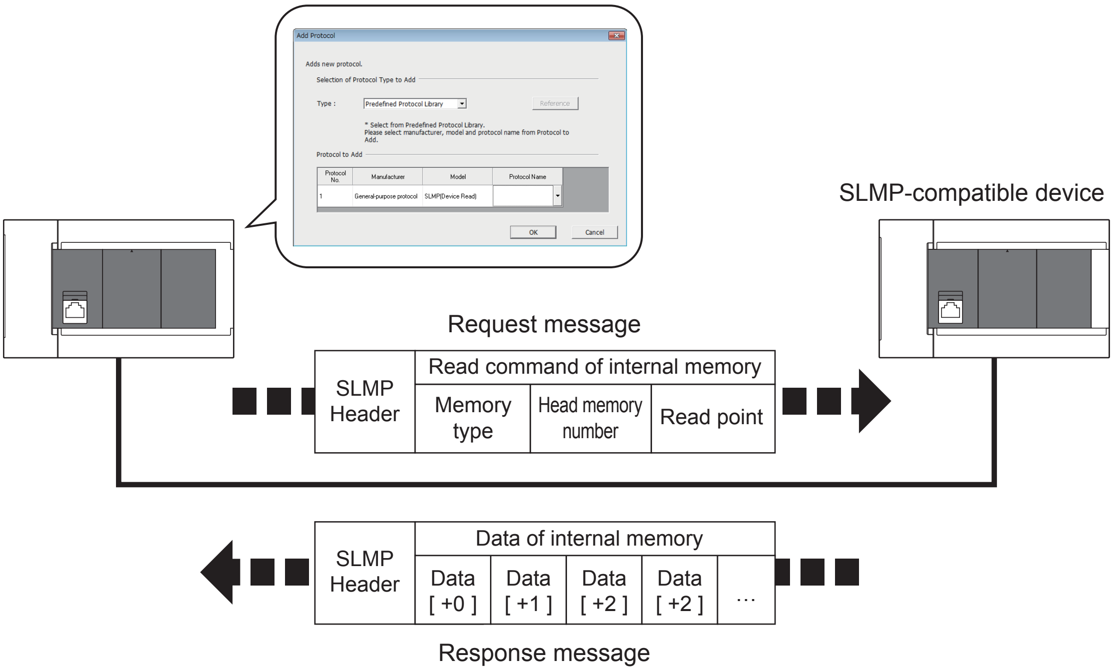
Access via network

SLMP allows an external device to access modules in the same network and other networks seamlessly via an SLMP compatible device.



Easy SLMP communication with the predefined protocol support function

SLMP communication can be easily used with the predefined protocol support function of the engineering tool.
Like external devices communicating on SLMP, CPU modules can control SLMP-compatible devices.



Adds new protocol.

Selection of Protocol Type to Add

Type : Predefined Protocol Library

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

Protocol to Add

Protocol No.	Manufacturer	Model	Protocol Name
1	General-purpose protocol	SLMP(Device Read)	

Request message

SLMP Header	Read command of internal memory		
	Memory type	Head memory number	Read point

SLMP Header	Data of internal memory				
	Data [+0]	Data [+1]	Data [+2]	Data [+2]	…

Response message

36 SLMP DATA COMMUNICATION

This chapter describes the SLMP data communication by which the external equipment reads or writes data to an Ethernet-equipped module.

36.1 Type and Application of the Data Communication Frame

This section describes the type and application of the frame (data communication message) by which the external equipment accesses an Ethernet-equipped module with SLMP.

When the external equipment accesses an Ethernet-equipped module using Ethernet, the data communication is executed by sending or receiving a command message (access request) and response message (response) of the following frame.

Target communication method	Applicable communication frames	Communication data code	Section of control procedure
Ethernet	• 3E frame • 1E frame	ASCII code or binary code	Page 562 MESSAGE FORMAT

Point

There are 2 types of ASCII code: ASCII code (X, Y OCT) and ASCII code (X, Y HEX).

The specification method of the device number for the X (input) and Y (output) to be accessed is different from each other.

- ASCII code (X, Y OCT): octal
- ASCII code (X, Y HEX): hexadecimal

Unless otherwise specified, the both of them are described as ASCII code.

For supported versions of ASCII code (X, Y HEX) for the FX5U/FX5UC CPU module, refer to ☞ Page 936 Added and Changed Functions.

3E frame

- The message format is the same as the QnA compatible 3E frame of MC protocol.
- The main purpose of the frame is to access all the devices of the Ethernet-equipped module from the external equipment.
- The frame enables access to CC-Link IE Controller Network and CC-Link IE Field Network.

Point

When using binary codes, the communication time will decrease since the amount of communication data is reduced by approximately half comparing to using ASCII codes.

For the device range of 3E frame, refer to the following.

☞ Page 611 Device range

1E frame

- The message format is the same as the A compatible 1E frame of MC protocol.
- This frame is designed mainly to facilitate access to Ethernet-equipped module devices from the external equipment.

Point

The communication time of 1E frame is shorter than 3E frame because the amount of communication data by the 1E frame is smaller compared to 3E frame.

For the device range of 1E frame, refer to the following.

☞ Page 668 Device range

36.2 Allowable Access Range of Each Data Communication Frame

The following shows the frame and access range of a message used in SLMP.

SLMP frame

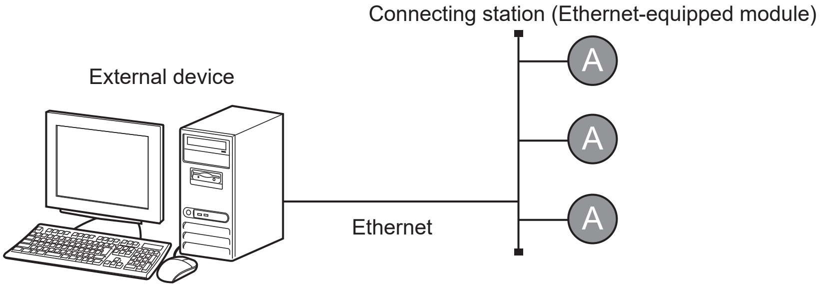
Frame	Type of the network which connects the external device with the connecting stations	Reference
Ethernet communication frame (3E/1E frame)	Ethernet	Page 562 MESSAGE FORMAT

Access range

Ethernet communication frame

■When the external device is connected directly with the Ethernet-equipped module via Ethernet

In the following system configuration, communication with the Ethernet-equipped module is possible using the Ethernet communication frame from the external device.



Assigned symbol	Description
A	Station directly connected to the external device

36.3 Concept of Control Procedure of SLMP

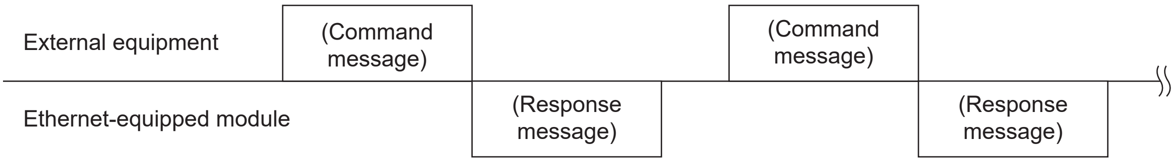
This section describes the concept of the procedure (control procedure) when the external equipment accesses an Ethernet-equipped module with SLMP.

Sending a command message

Data communication using SLMP communication is executed in half-duplex communication.

To access the Ethernet-equipped module, send the next command message after receiving a response message for the preceding command message from the Ethernet-equipped module.

(Until the receiving of the response message is completed, the next command message cannot be sent.)



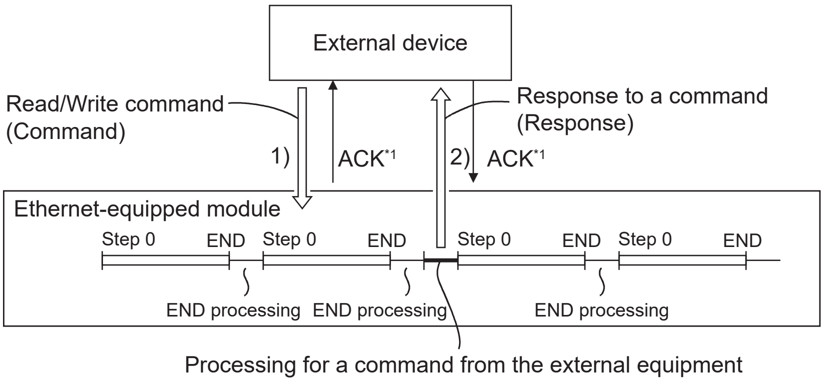
When a response message of completion for a command message cannot be received

When a response message of completion with an error is received, take corrective actions depending on the error code in the response message.

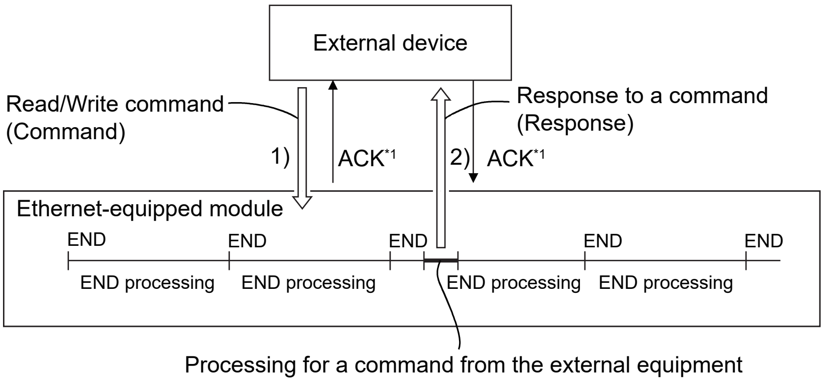
36.4 Access Timing of the Ethernet-equipped Module Side

The following shows the access timing of the Ethernet-equipped module side when the Ethernet port of the Ethernet-equipped module is accessed from the external device.

- RUN



- STOP



*1 ACK shown in the figure is a response which is sent or received between the Ethernet-equipped module and external equipment (a response for receiving a message) when the Ethernet-equipped module is accessed from the external equipment using TCP/IP communication.
This response is not the same as the one for the processing requested from the external equipment by a command message (processing result).
When access is executed using UDP/IP communication via the Ethernet port, an ACK response is not sent.

1. To send a read request or a write request to the Ethernet-equipped module side from the external equipment, a command message is sent.
2. The Ethernet-equipped module reads or writes the data according to the description requested from the external equipment when the END instruction of the CPU module is executed and sends a response message (response) including the processing result to the external equipment of the request source.

Point

- Access between the external equipment and Ethernet-equipped module is processed at each END processing when the Ethernet-equipped module is running for a command request. (The scan time becomes longer by the processing time of the command request.)
- When accesses are requested simultaneously to the Ethernet-equipped module from multiple external equipment, the processing requested from the external device may be on hold until END processing is performed several times depending on the request timing.

36.5 Transfer Time

This section describes the method for calculating the link time of the CPU module.

Point

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

Link time

■Calculation method

Calculate the minimum processing time of the SLMP communication by the following calculation formula.

However, the processing time may become longer depending on the load of the network (how much a line is crowded), window size of each connecting device, number of connections to be used simultaneously, and system configuration. As a guideline, recognize the value calculated by the following calculation formula as the processing time when a communication is executed by only one connection.

- Minimum processing time of the SLMP communication (for batch read or batch write)

Tfs = Ke + (Kdt × Df) + Scr × Number of scans required for processing + ACK processing time of external equipment

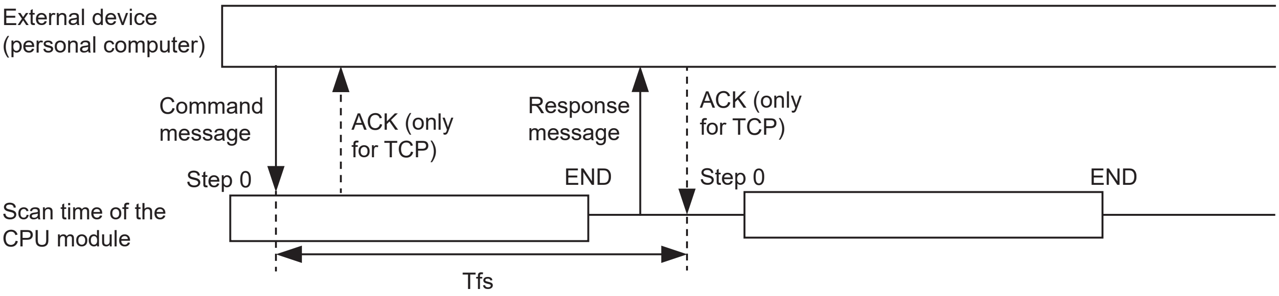
Tfs: Time from when the request data of a personal computer is received until the CPU module completes the processing (Unit: ms)[1]

Ke, Kdt: Constant (Refer to the following table.)

Df: Number of words of the request data + Number of words of the response data (application data part)

Scr: Scan time

[1] The following shows the timing from when the request data of a personal computer is received until the CPU module completes the processing.



Communication description	TCP/IP communication		UDP/IP communication	
	Ke	Kdt	Ke	Kdt
Batch read	1	0.001	1	0.001
Batch write	1	0.001	1	0.001

Ex.

[Calculation example 1]

Time from when the request data of a personal computer is received until the processing is completed, when a TCP/IP communication is executed between personal computers and 32 points data read from the data register (D) of own station by the SLMP communication in binary code (Unit: ms)

The scan time of the mounted station is 40 ms.

Tfs = 1 + (0.001 × 32) + 40 × 1 + ACK processing time of external equipment

[Calculation example 2]

Time from when the request data of a personal computer is received until the processing is completed, when a TCP/IP communication is executed between personal computers and 32 points data written to the data register (D) of own station by the SLMP communication in binary code (Unit: ms)

The scan time of the mounted station is 40 ms.

Tfs = 1 + (0.001 × 32) + 40 × 1 + ACK processing time of external equipment

37 MESSAGE FORMAT

This chapter describes the message data format, the data specification method, and limitations etc. when performing SLMP data communication using the 3E frame to the Ethernet port.

Frame type	Ethernet port	Remarks
3E frame	Communicable	The message format is the same as the QnA compatible 3E frame of MC protocol.
1E frame	Communicable	The message format is the same as the A compatible 1E frame of MC protocol.

37.1 3E Frame

This section describes the message format for each command when performing the data communication using the 3E frame.

Message format and control procedure

This section describes the message format and the control procedures when performing the data communication using the 3E frame.

Data format

The data format consists of header and application data.

■Request message

Header	Application data									
	Subheader	Request destination network No.	Request destination station No.	Request destination module I/O No.	Request destination multidrop station No.	Request data length	Monitoring timer	Command	Subcommand	Request data

■Response message

Header	Application data							
	Subheader	Request destination network No.	Request destination station No.	Request destination module I/O No.	Request destination multidrop station No.	Response data length	End code	Response data

Header

This header is for TCP/IP and UDP/IP.

Add the header for external equipment to Ethernet-equipped module (command message) at the external equipment side before sending the message (normally the header is added automatically).

It is not necessary to set the header for Ethernet-equipped module to external equipment (response message) by the user because the header is added by the Ethernet-equipped module automatically.

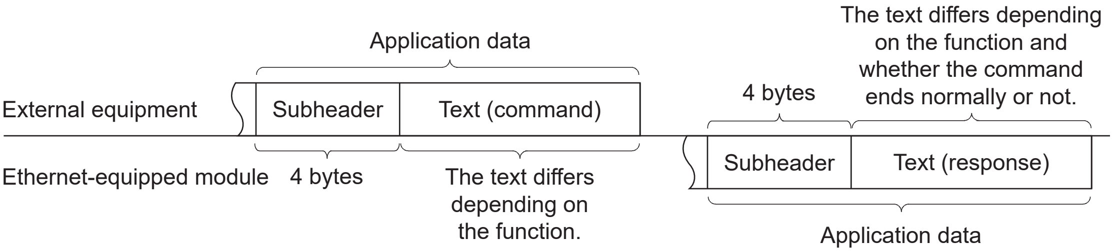
Application data

Application data is divided into subheader and text.

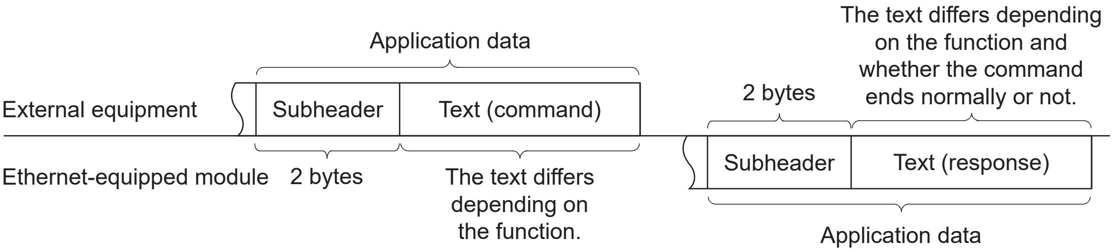
The subheader indicates whether a message is a command message or a response message. (☞ Page 563 Subheader configuration)

Text is the request data (command) and the response date (response) in each function. (☞ Page 602 3E FRAME COMMANDS)

■When communicating data in ASCII code



■When communicating data in binary code



Point

It is not necessary to set the response to a command from the external equipment by the user because the response is created and sent by the Ethernet-equipped module.

Subheader configuration

This section describes the subheader configuration.

■When communicating data in ASCII code

Command message

5	0	0	0
35H	30H	30H	30H

Response message

D	0	0	0
44H	30H	30H	30H

■When communicating data in binary code

Command message

50H	00H

Response message

D0H	00H

Control procedure

This section describes the control procedures and the format of the application data when performing the data communication.

The ☐ (Thick line) part shown in the message explanation diagram (☞ Page 610 Device Access) of this section are items common to all commands and correspond to the * portion of the message explanation diagrams indicated in this chapter.

For the data contents and the data specification method of the ☐ (Thick line) part, refer to ☞ Page 568 Application data specification items.

Point

Data code (ASCII/binary) to be used when communicating, it is determined by the parameters of GX Works3.

[CPU module]

Navigation window ⇨ [Parameter] ⇨ Module model ⇨ [Module Parameter] ⇨ [Ethernet Port] ⇨ [Basic Settings] ⇨ [Own Node Settings] ⇨ "Communication Data Code"

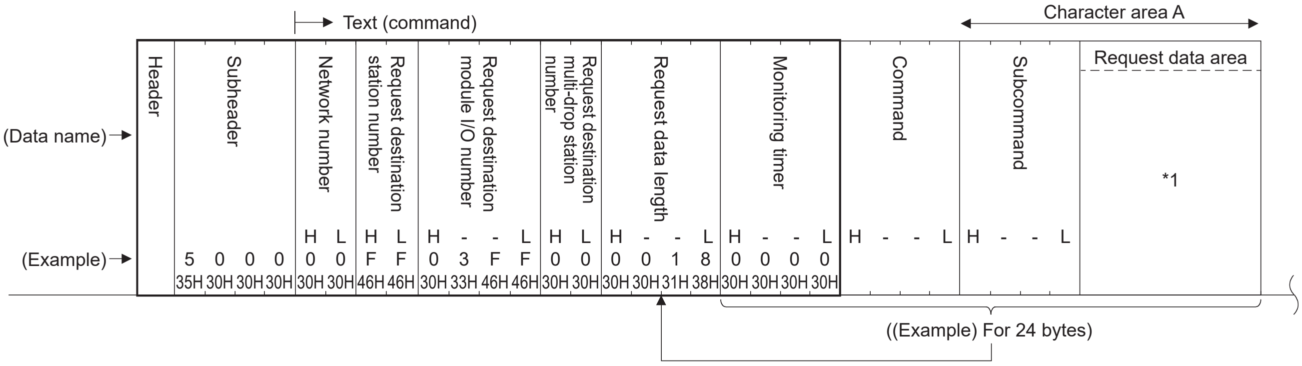
[Ethernet module]

Navigation window ⇨ [Parameter] ⇨ [Module Information] ⇨ [FX5-ENET] or [FX5-ENET/IP] ⇨ [Basic Settings] ⇨ [Own Node Settings] ⇨ "Communication Data Code"

■When communicating data in ASCII code

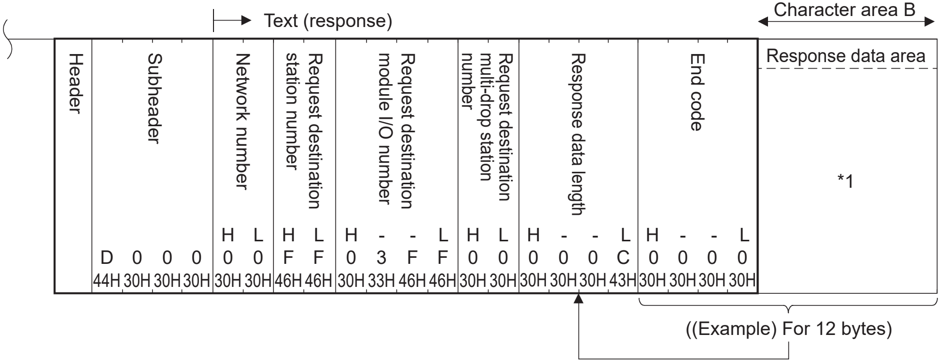
- When data is read from an Ethernet-equipped module by external equipment

External equipment → Ethernet-equipped module (command message)

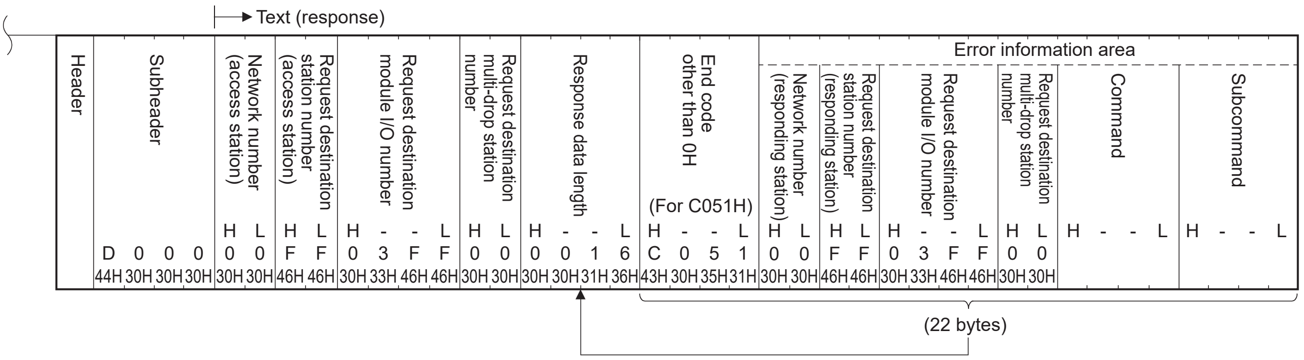


Ethernet-equipped module → External equipment (response message)

(When completed normally)



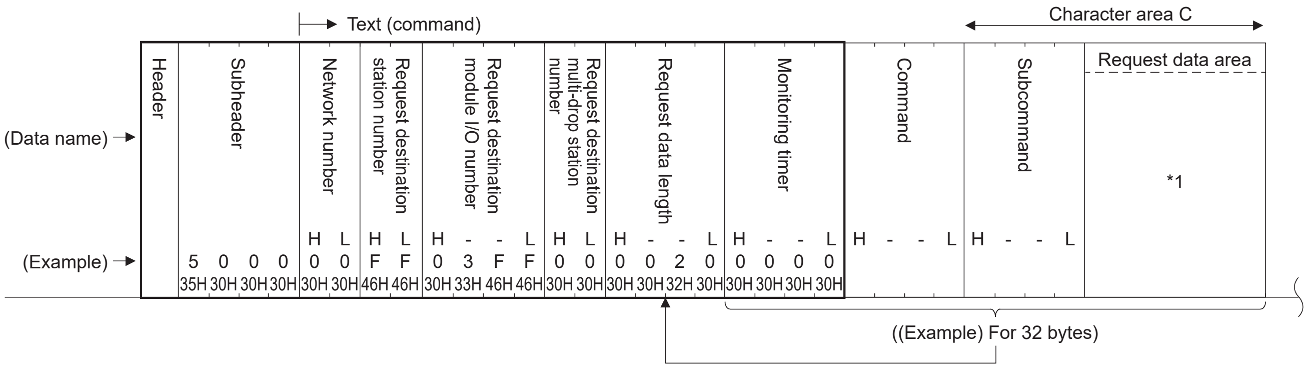
(When completed with error)



*1 The order of data items differs depending on the command or subcommand. For details, refer to ☞ Page 610 Device Access.

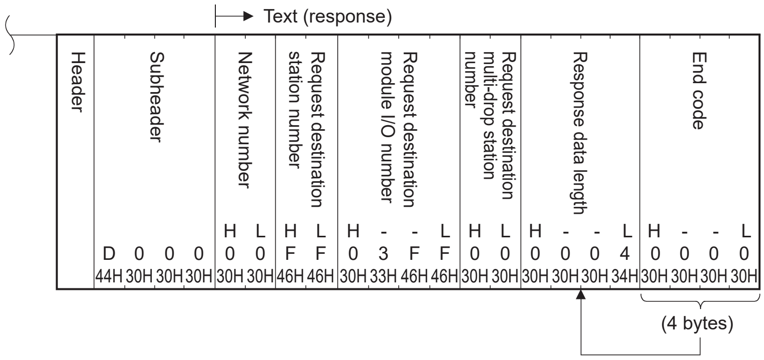
- When data is written from external equipment to an Ethernet-equipped module

External equipment → Ethernet-equipped module (command message)

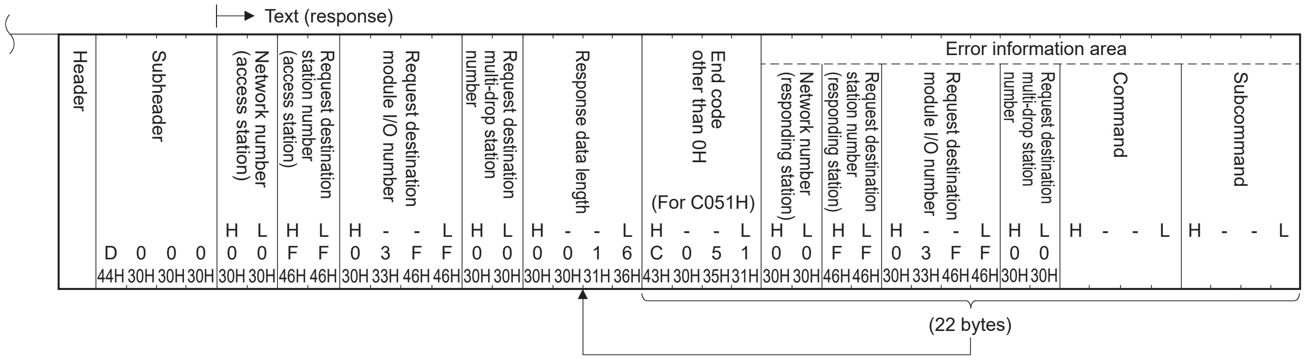


Ethernet-equipped module → External equipment (response message)

(When completed normally)



(When completed with error)

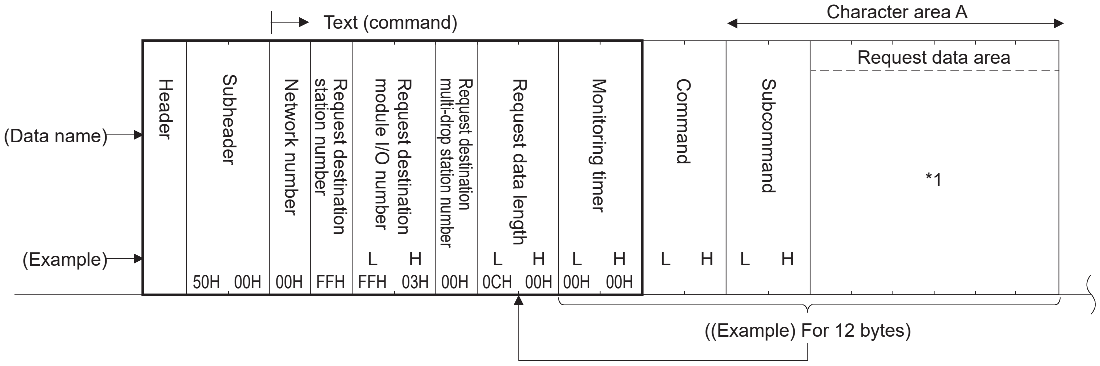


*1 The order of data items differs depending on the command or subcommand. For details, refer to ☞ Page 610 Device Access.

■When communicating data in binary code

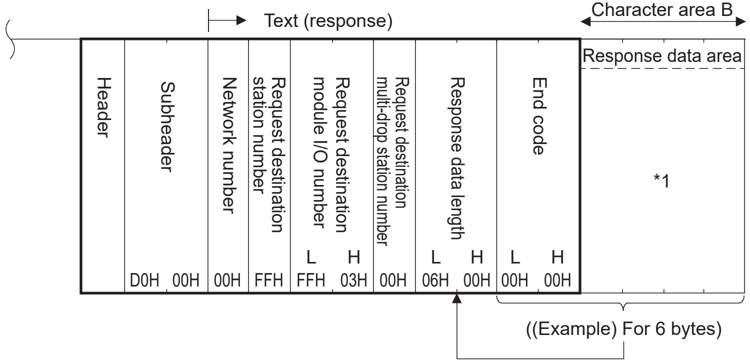
- When data is read from an Ethernet-equipped module by external equipment

External equipment → Ethernet-equipped module (command message)

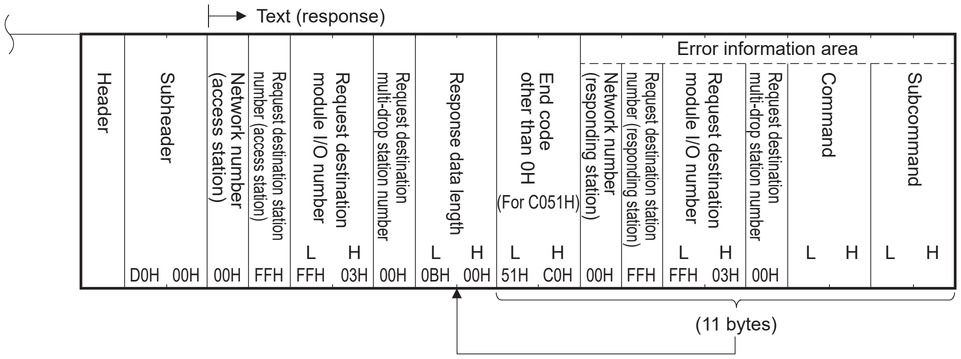


Ethernet-equipped module → External equipment (response message)

(When completed normally)



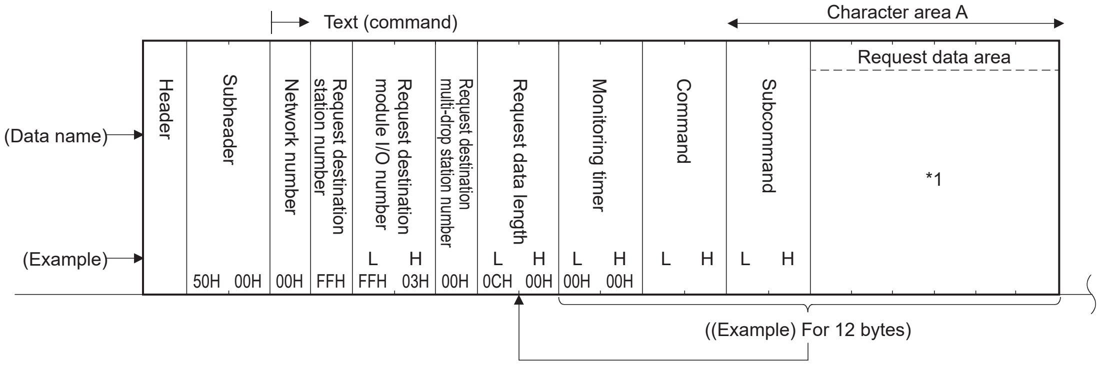
(When completed with error)



*1 The order of data items differs depending on the command or subcommand. For details, refer to ☞ Page 610 Device Access.

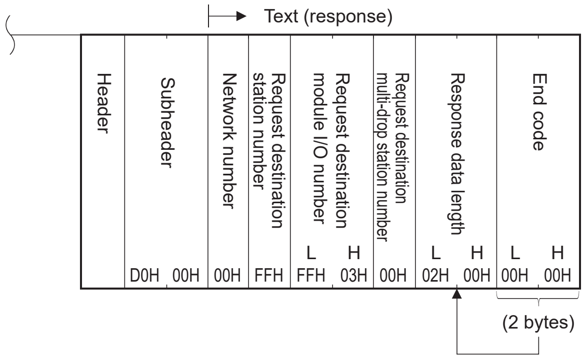
- When data is written from external equipment to an Ethernet-equipped module

External equipment → Ethernet-equipped module (command message)

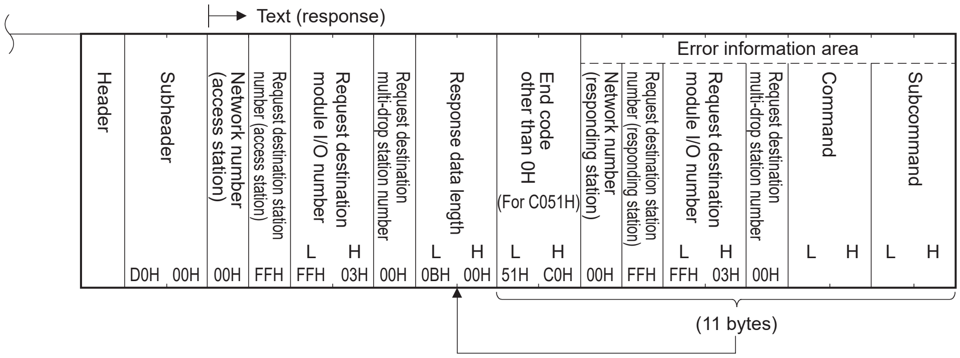


Ethernet-equipped module → External equipment (response message)

(When completed normally)



(When completed with error)



*1 The order of data items differs depending on the command or subcommand. For details, refer to ☞ Page 610 Device Access.

Application data specification items

This section describes the data contents and the specification method of common data items in the application data in each message when performing the data communication using the 3E frame.

Request destination network number and request destination station number

■Request message

Header	Application data									
	Subheader	Request destination network No.	Request destination station No.	Request destination module I/O No.	Request destination multidrop station No.	Request data length	Monitoring timer	Command	Subcommand	Request data

■Response message

Header	Application data							
	Subheader	Request destination network No.	Request destination station No.	Request destination module I/O No.	Request destination multidrop station No.	Response data length	End code	Response data

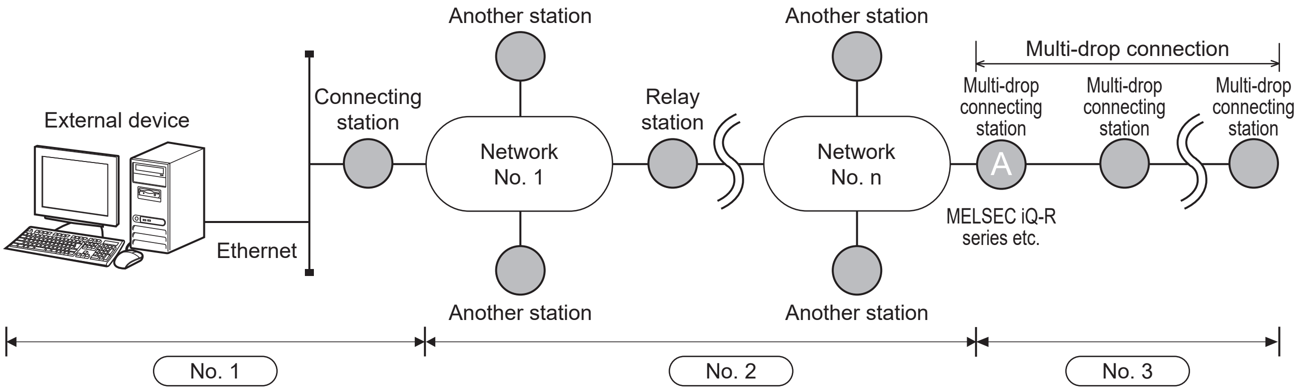
Specify the request destination network number and request destination station number to be used as an access destination in hexadecimal.

Specify the request destination network number and request destination station number according to installation conditions of access destination stations based on the following table.

Data of the response message is a value set in the request message.

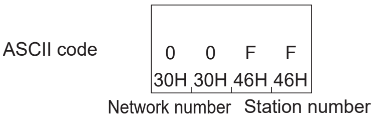
No.	Access destination	Station to be specified	Request destination network number	Request destination station number
1[*1]	Connecting station (Within the range indicated in No. 1 in the figure below)	— (Specify the fixed value indicated on the right)	00H	FFH
2	Other stations or relay station (Within the range indicated in No. 2 in the figure below)	Access destination station	01 to EFH (1 to 239)	01 to 78H (1 to 120): Station number 7DH: Assigned control station/ Master station 7EH: Present control station/ Master station
3	Multi-drop connecting station via network (Within the range indicated in No. 3 in the figure below)	A station on the network where multi-drop connecting stations are connected (In the figure below, [A] is specified)	01 to EFH (1 to 239)	01 to 78H (1 to 120): Station number 7DH: Assigned control station/ Master station 7EH: Present control station/ Master station

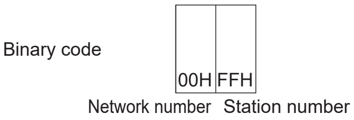
[*1] Please use specification No.1 to access Ethernet-equipped module.



Ex.

When specifying the connecting station (network No.00H, station number FFH)





Precautions

The stations of network number 240 to 255 cannot be accessed.

Ethernet-equipped module cannot perform multi-drop connection.

Ethernet-equipped module cannot perform connection via network.

Request destination module I/O number

■Request message

Header	Application data									
	Subheader	Request destination network No.	Request destination station No.	Request destination module I/O No.	Request destination multidrop station No.	Request data length	Monitoring timer	Command	Subcommand	Request data

■Response message

Header	Application data							
	Subheader	Request destination network No.	Request destination station No.	Request destination module I/O No.	Request destination multidrop station No.	Response data length	End code	Response data

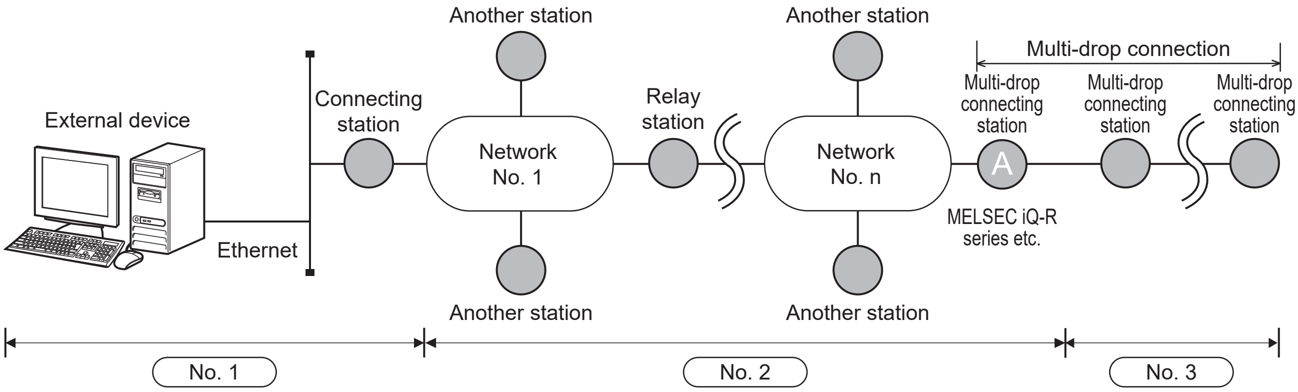
Select the module number of the access destination from the table below.

When the send destination of the request message is a multi-drop connecting station that is connected to the request destination station, set the I/O number (upper 3 digits) of the serial communication module (MELSEC iQ-R series or other series) which is performing the multi-drop connection.

No.	Module to be accessed[*1]	Request destination station Request destination module I/O number
1[*2]	Own station	03FFH
2	Other station (control CPU)	03FFH
3	The module which is performing multi-drop connection with serial communication module ("A" in the figure below), which is connected to the network	0000H to 01FFH

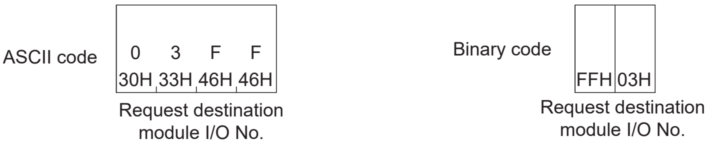
*1 Ethernet-equipped module cannot perform multi-drop connection.

*2 Please use specification No.1 to access Ethernet-equipped module.



Ex.

When specifying the own station (03FFH) as the request destination module I/O number



Request destination multi-drop station number

■Request message

Header	Application data									
	Subheader	Request destination network No.	Request destination station No.	Request destination module I/O No.	Request destination multidrop station No.	Request data length	Monitoring timer	Command	Subcommand	Request data

■Response message

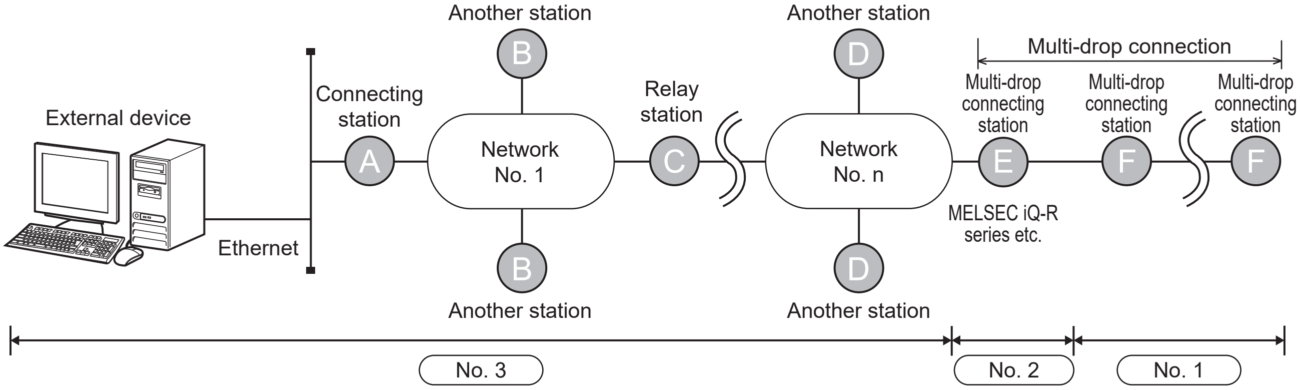
Header	Application data							
	Subheader	Request destination network No.	Request destination station No.	Request destination module I/O No.	Request destination multidrop station No.	Response data length	End code	Response data

Specify the station number of the SLMP compatible device linked by the multi-drop connection in the access destination, within the range shown in the table below.

When not specifying the SLMP compatible device linked by the multi-drop connection, set 00H.

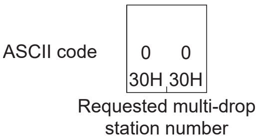
No.	Access station of external equipment	Request destination multi-drop station number
1	Stations on the multi-drop connection ("F" in the figure below)	Set the station number (00H to 1FH (0 to 31)) ("F" in the figure below)
2	A station that relays the network and the multi-drop connection ("E" in the figure below)	00H (0)
3[*1]	Other than above	00H (0)

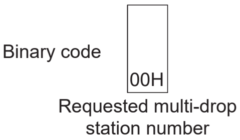
[*1] Please use specification No.3 to access Ethernet-equipped module.



Ex.

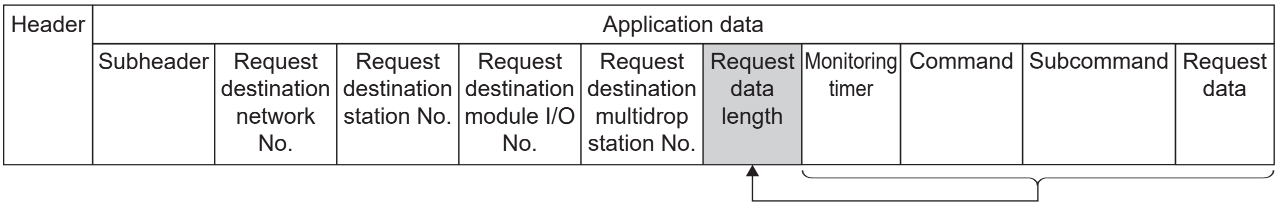
When specifying 00H as the requested multi-drop station number





Request data length

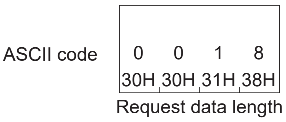
■Request message

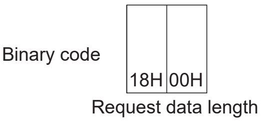


Specify the total data size from the Monitoring timer to the request data in hexadecimal. (Unit: byte)

Ex.

When the request data length is 24 (18H) bytes



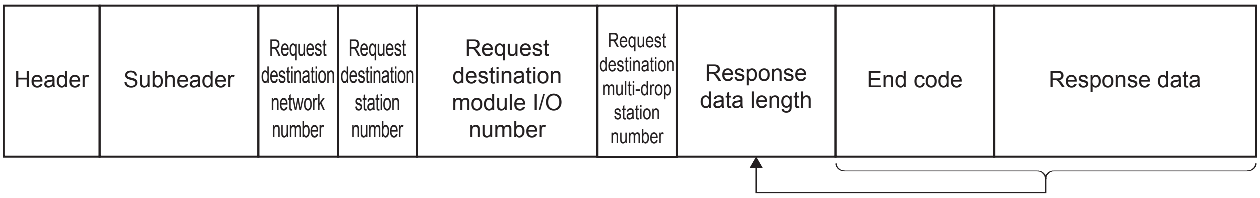


Response data length

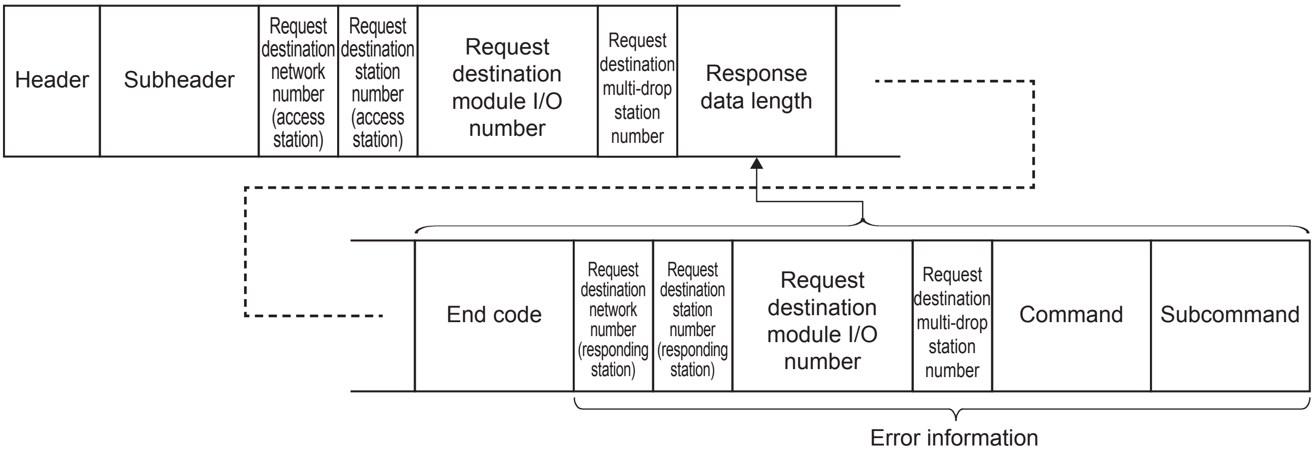
■Response message

When normally completed, the total data size from the end code to the response data is set in hexadecimal. When completed with error, the total data size from the end code to the error information is set in hexadecimal. (Unit: byte)

(When completed normally)



(When completed with error)



Monitoring timer

■Request message

Header	Application data									
	Subheader	Request destination network No.	Request destination station No.	Request destination module I/O No.	Request destination multidrop station No.	Request data length	Monitoring timer	Command	Subcommand	Request data

This timer is used to set the wait time until a response is returned after the SLMP-compatible device that has received a request message from an external device sends a processing request to the accessed device.

- 0000H (0): Indefinitely waiting (waiting until the processing is completed)
- 0001H to FFFFH (1 to 65535)[1]: Wait time (unit: 250ms)

[1] Supported only for Ethernet modules.

For normal data communication, it is recommended to use this message within the setting range shown in the following table depending on the communication destination.

Access destination	Monitoring timer
Other station	01H to 28H (0.25 to 10 seconds)

Ex.

■When communicating data in ASCII code

Sending from the upper byte to the lower byte.

■When communicating data in binary code

Sending from the lower byte to the upper byte.

Precautions

Specify "0000H" (indefinite wait) for the CPU module.

End code

■Response message

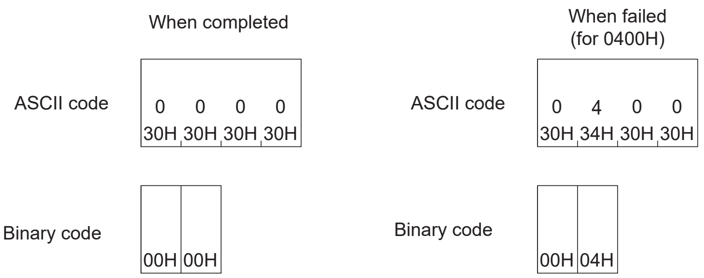
Header	Application data							
	Subheader	Request destination network No.	Request destination station No.	Request destination module I/O No.	Request destination multidrop station No.	Response data length	End code	Response data

The command processing result is stored.

When normally completed, "0000H" is stored. When completed with error, an error code set at the request destination is stored.

(For the set error code and corresponding error contents, refer to ☞ Page 855 EVENT CODES and manuals of the SLMP compatible device of the response station.)

Ex.



Request data

■Request message

Header	Application data									
	Subheader	Request destination network No.	Request destination station No.	Request destination module I/O No.	Request destination multidrop station No.	Request data length	Monitoring timer	Command	Subcommand	Request data

Set a command to be executed and data for the argument of the subcommand.

(Some commands and subcommands do not require the request data specification.)

For details on the request data, refer to the paragraph relating to the command to be executed. (☞ Page 610 Device Access)

Response data

■Response message

Header	Application data							
	Subheader	Request destination network No.	Request destination station No.	Request destination module I/O No.	Request destination multidrop station No.	Response data length	End code	Response data

The processing result of the request data is stored.

(Some commands do not return response messages.)

For details on the response data, refer to the paragraph relating to the command to be executed. (☞ Page 610 Device Access)

Error information

The request destination network number, request destination station number, request destination module I/O number, and request destination multi-drop station number of the station which responded with errors are stored.

Numbers which differ from the requested station specified by the request message may be stored because the information of the station which responded with errors is stored.

The command and the subcommand specified by the request message of the request data are stored.

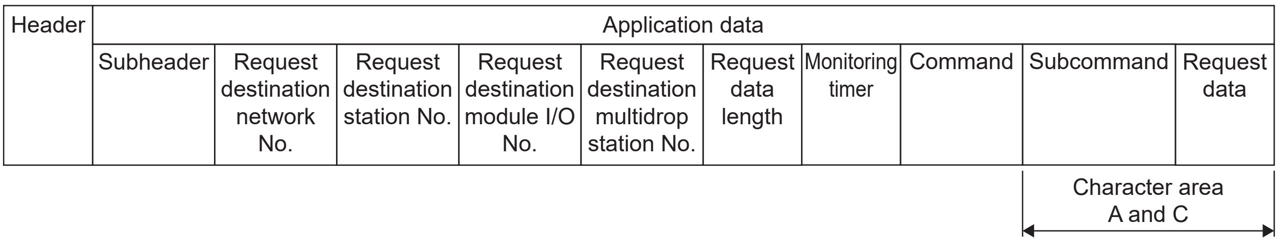
Transfer data in character area

This section describes how to transfer bit device data and word device data and data alignment in the character area sent and received between the external equipment and the Ethernet-equipped module by using each command.

The transfer data explained below is handled as the character area A for reading and monitoring and the character area C for writing, testing, and registering the monitor data are stored.

Character area

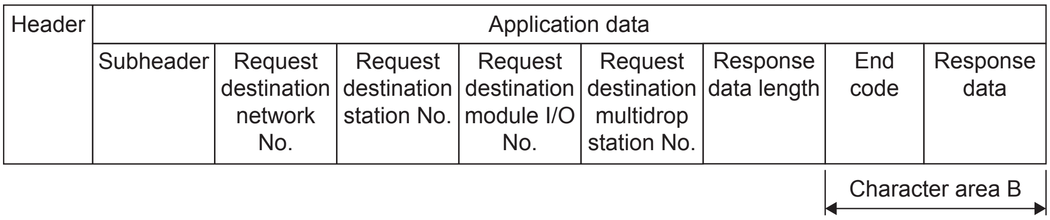
■Request message



Header	Application data									
	Subheader	Request destination network No.	Request destination station No.	Request destination module I/O No.	Request destination multidrop station No.	Request data length	Monitoring timer	Command	Subcommand	Request data

Character area A and C

■Response message



Header	Application data							
	Subheader	Request destination network No.	Request destination station No.	Request destination module I/O No.	Request destination multidrop station No.	Response data length	End code	Response data

Character area B

Communicating data (when communicating in ASCII code)

■When bit device memory is read or written

The bit device memory is handled in 1-bit (1-point) units or in 1-word (16-point) units.

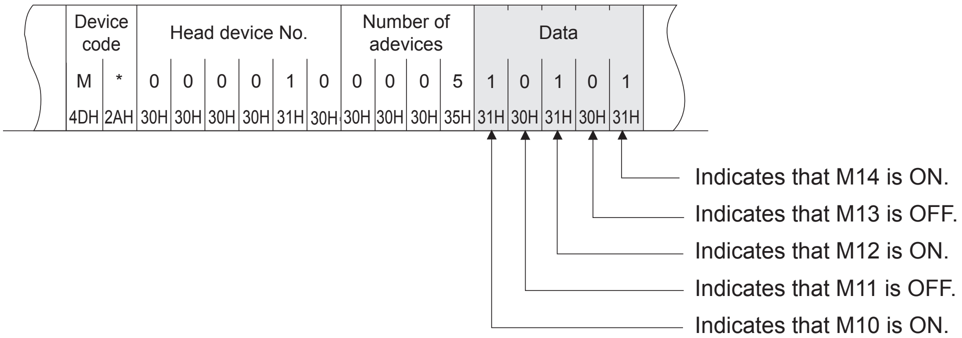
The transfer data in each case is described below.

- In 1-bit (1-point) units

When the bit device memory is handled in 1-bit units, specify 1-point (1-byte) with an ASCII code, and express "1" (31H) for ON and "0" (30H) for OFF. Specify for the number of devices starting from the head device in the order of device numbers.

Ex.

When indicating the on/off status of five devices starting from M10

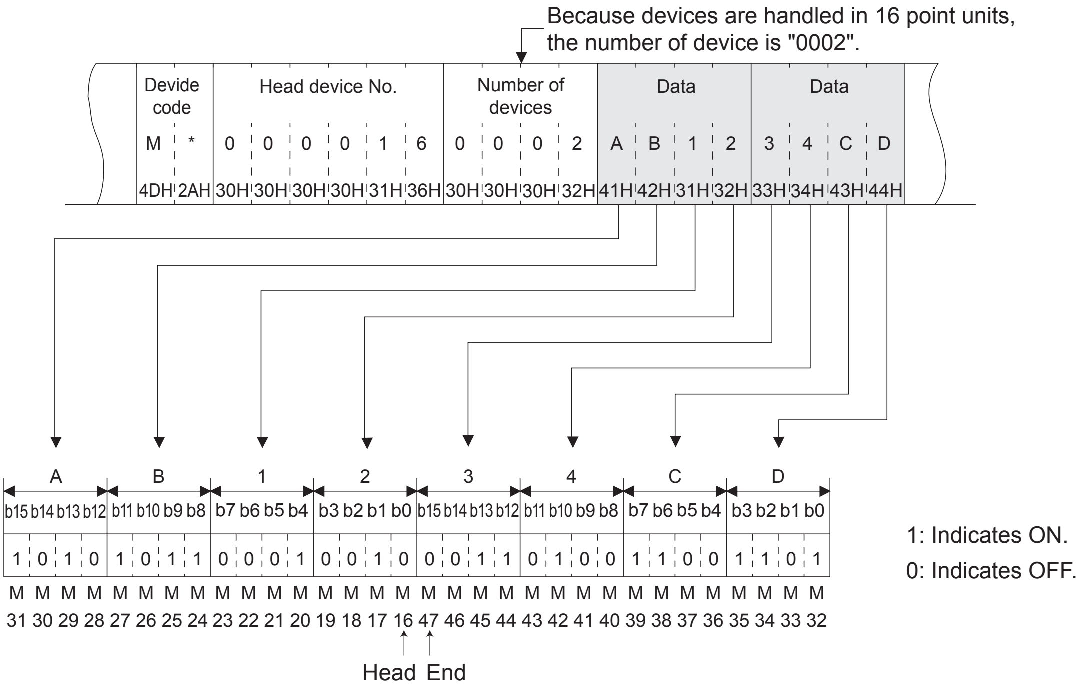


- In 1-word (16-point) units

When the bit device memory is handled in 1-word units, specify one word (16 bits) with a 4-digit ASCII code, and express 1-point with 1-bit ON/OFF. Specify for the number of devices starting from the head device in 1-word units in the order from the most significant bit to the least significant bit (b15 to b0).

Ex.

When indicating ON/OFF status of 32 points from M16

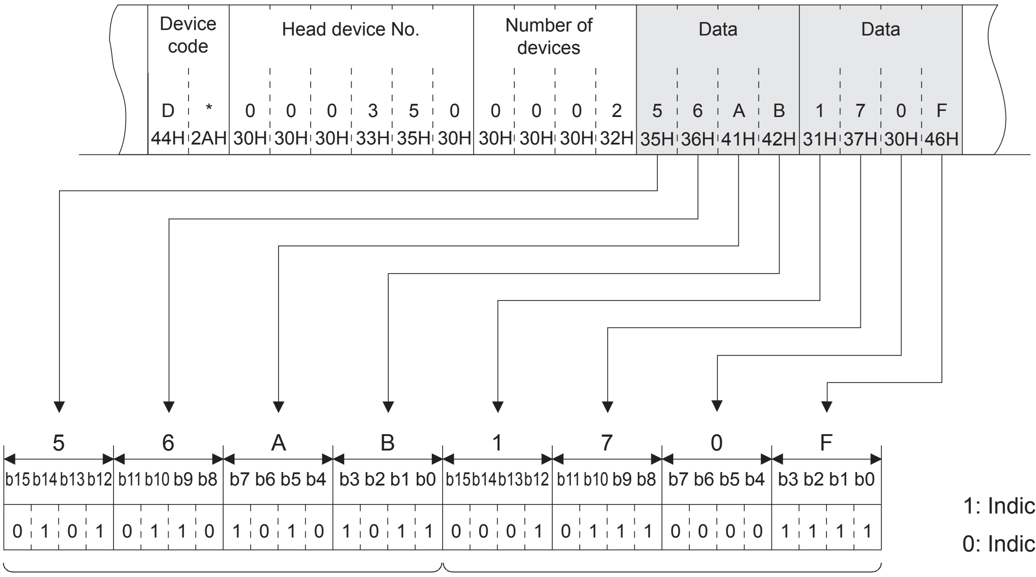


■When word device memory is read or written

One word device is expressed with a 1-word (4-byte) 4-digit ASCII code (hexadecimal). Specify for the number of devices starting from the head device in 1-word units in the order from the most significant byte to the least significant byte (b15 to b0).

Ex.

When indicating the contents stored in the data registers D350 and D351



1: Indicates ON.

0: Indicates OFF.

Indicates that D350 stores "56ABH ("22187" in decimal)".

Indicates that D351 stores "170FH ("5903" in decimal)".

Point

Use capitalized code for alphabetical letter.

When data other than integer value (real number, character string), is stored in the word device memory for reading data, the stored value are read as integer value.

(Example 1) When a real number (0.75) is stored in D0 and D1, the value is read as the following integer value.

- D0 = 0000H, D1 = 3F40H

(Example 2) When a character string (12AB) is stored in D2 and D3, the character string is read as the following integer value.

- D2 = 3231H, D3 = 4241H

Data in word units handled when reading and writing buffer memory areas is expressed in the same way as the word device memory.

Communicating data (When communicating data in binary code)

■When bit device memory is read or written

The bit device memory is handled in 1-bit (1-point) units or in 1-word (16-point) units.

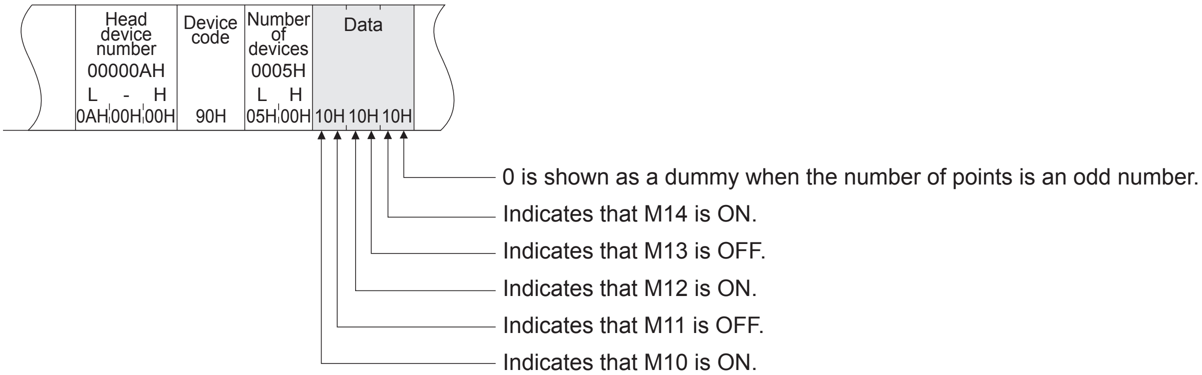
The transfer data in each case is described below.

- In 1-bit (1-point) units

When the bit device memory is handled in 1-bit units, specify 1-point, 4 bits, (two points, 1 byte) with a binary code, "1" for ON and "0" for OFF. Specify for the number of devices starting from the head device in the order of device numbers from the most significant bit.

Ex.

When indicating the on/off status of five devices starting from M10



0 is shown as a dummy when the number of points is an odd number.

Indicates that M14 is ON.

Indicates that M13 is OFF.

Indicates that M12 is ON.

Indicates that M11 is OFF.

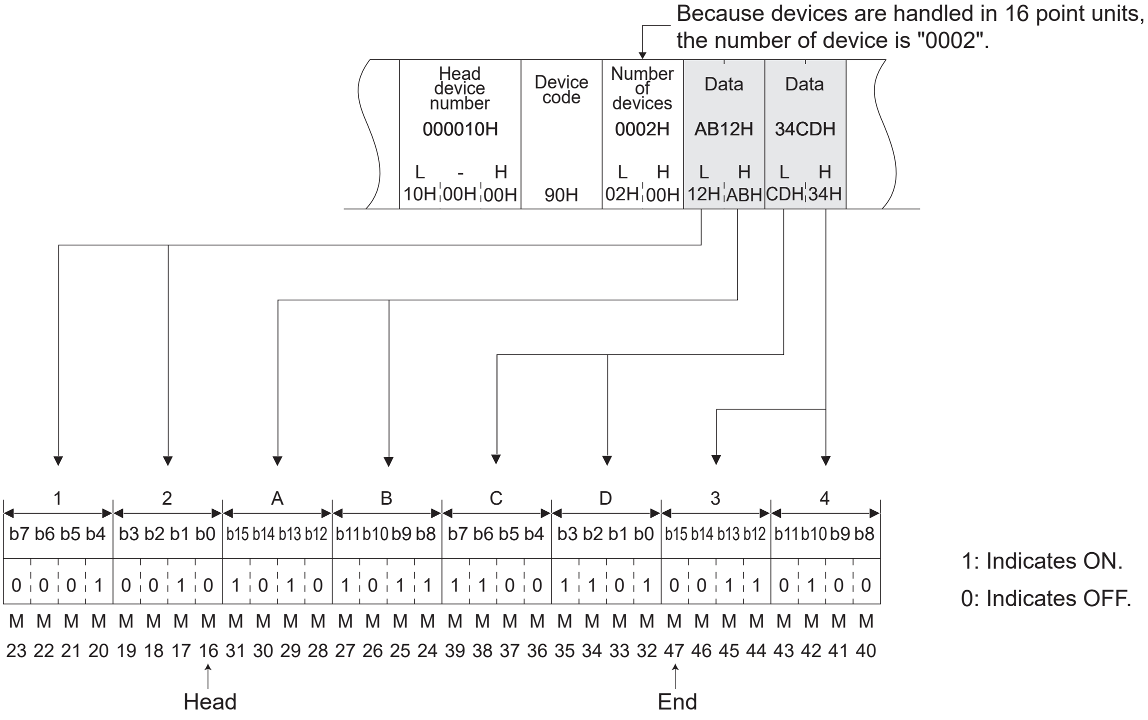
Indicates that M10 is ON.

- In 1-word (16-point) units

When the bit device memory is handled in 1-word units, specify one word (2 bytes) with a binary code, and express 1-point with 1-bit ON/OFF. Specify for the number of devices starting from the head device in 1-word units in the order from the least significant bytes (b7 to b0) to the most significant bytes (b15 to b8).

Ex.

When indicating ON/OFF status of 32 points from M16



1: Indicates ON.

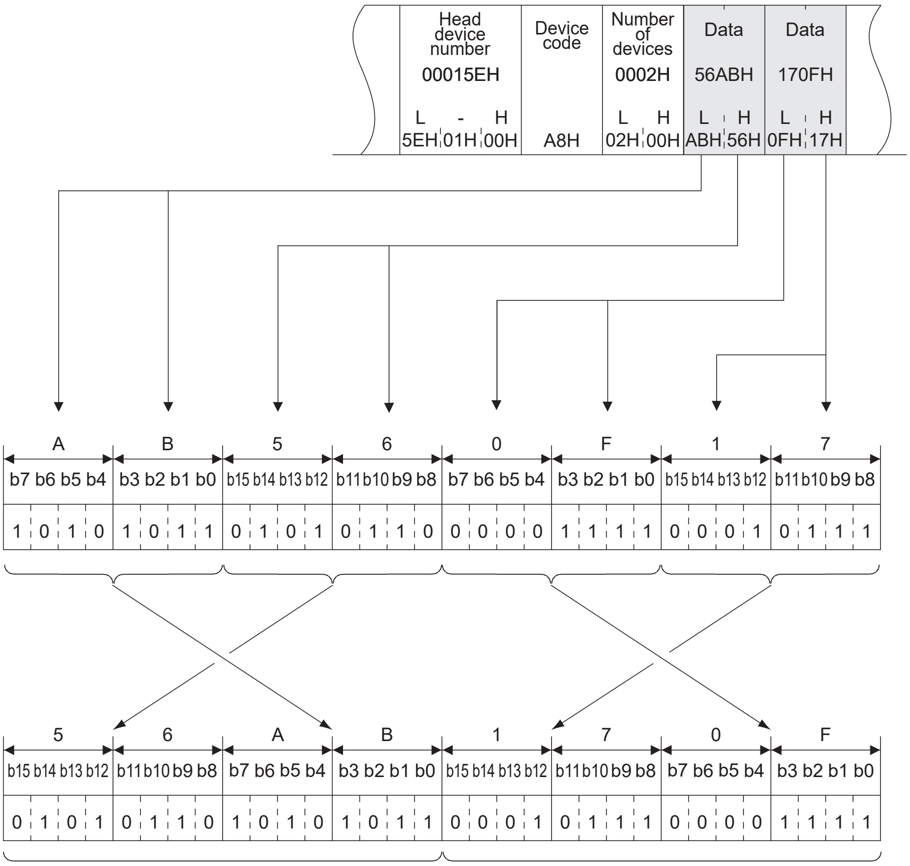
0: Indicates OFF.

■When word device memory is read or written

One word device memory is expressed with a 1-word (2-byte) 4-digit binary code (hexadecimal). Specify for the number of devices starting from the head device in 1-word units in the order from the least significant bytes (b7 to b0) to the most significant bytes (b15 to b8).

Ex.

When indicating the contents stored in the data registers D350 and D351



1: Indicates ON.

0: Indicates OFF.

Indicates that D350 stores "56ABH ("22187" in decimal)".

Indicates that D351 stores "170FH ("5903" in decimal)".

Point

When data other than integer value (real number, character string), is stored in the word device memory for reading data, the stored value are read as integer value.

(Example 1) When a real number (0.75) is stored in D0 and D1, the value is read as the following integer value.

- D0 = 0000H, D1 = 3F40H

(Example 2) When a character string (12AB) is stored in D2 and D3, the character string is read as the following integer value.

- D2 = 3231H, D3 = 4241H

Reading and writing extension file registers and buffer memory areas are performed in the same way as those of the word device memory.

Character areas

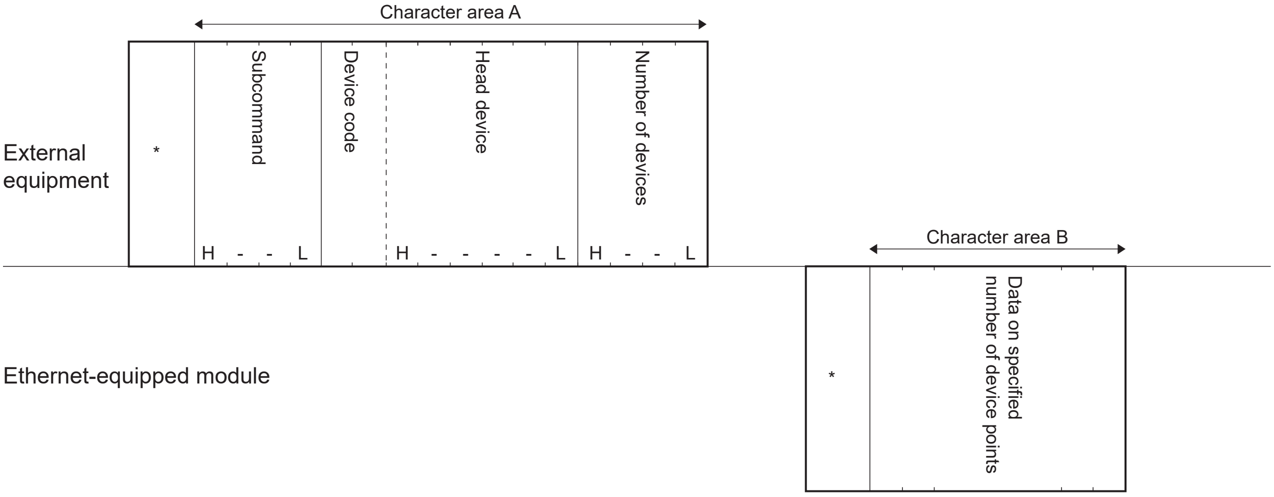
This section explains character areas in the control procedure (data area when communicating in binary code).

- Character areas differ depending on command to be used and contents to be specified. This section explains the data common to the character area when the device memory to be read or written is specified directly.
- Character area data handled only by a certain command and not by others, is explained in the section that explains the corresponding command.

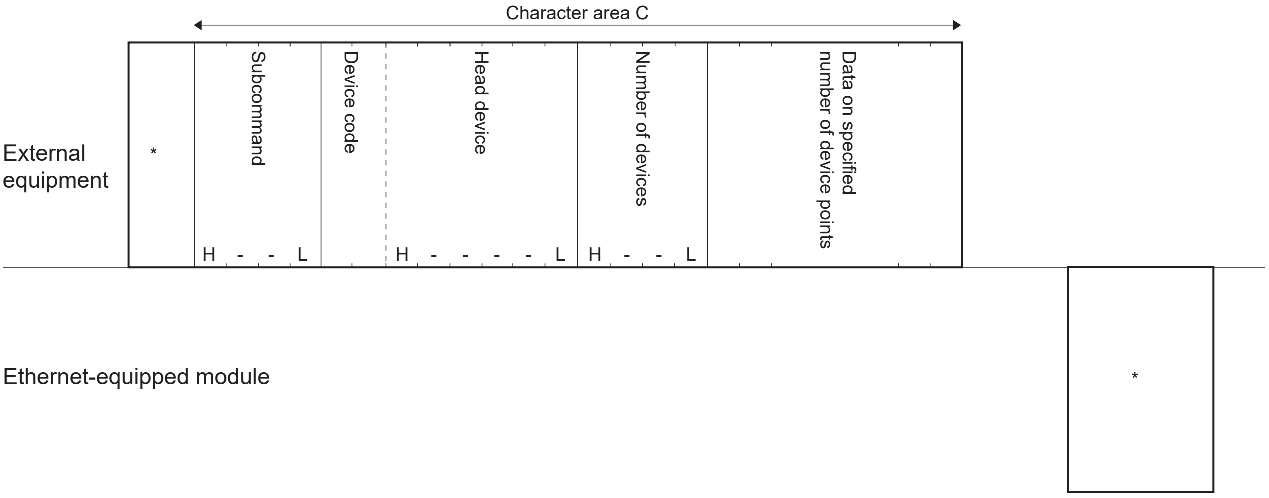
Data of character area (when communicating in ASCII code)

The data order and contents of character areas A, B, and C are identical when the same command is used under the same conditions in the control procedure when communicating using ASCII code.

■In the case of reading



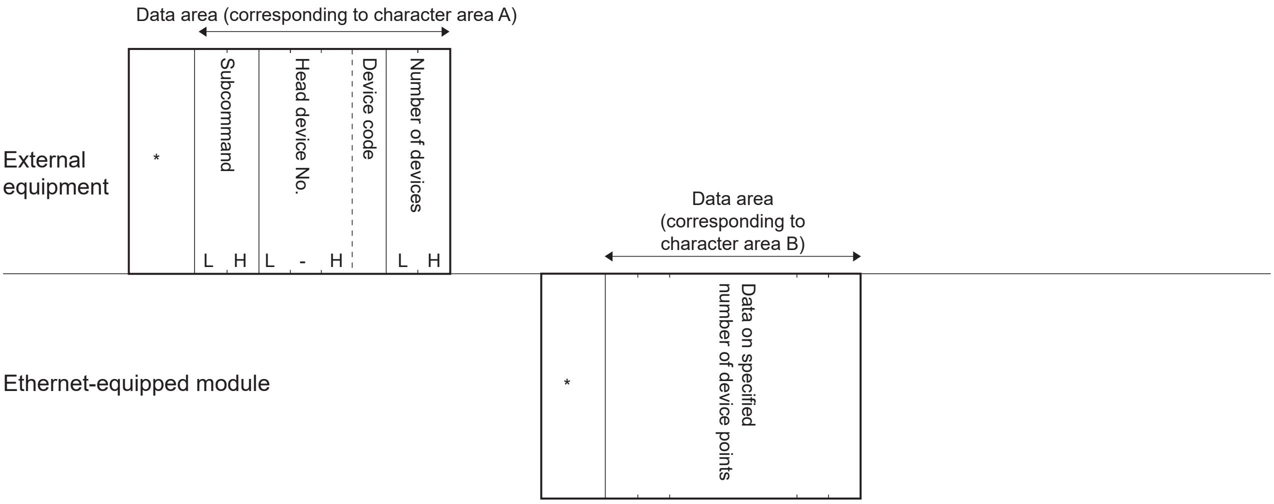
■In the case of writing



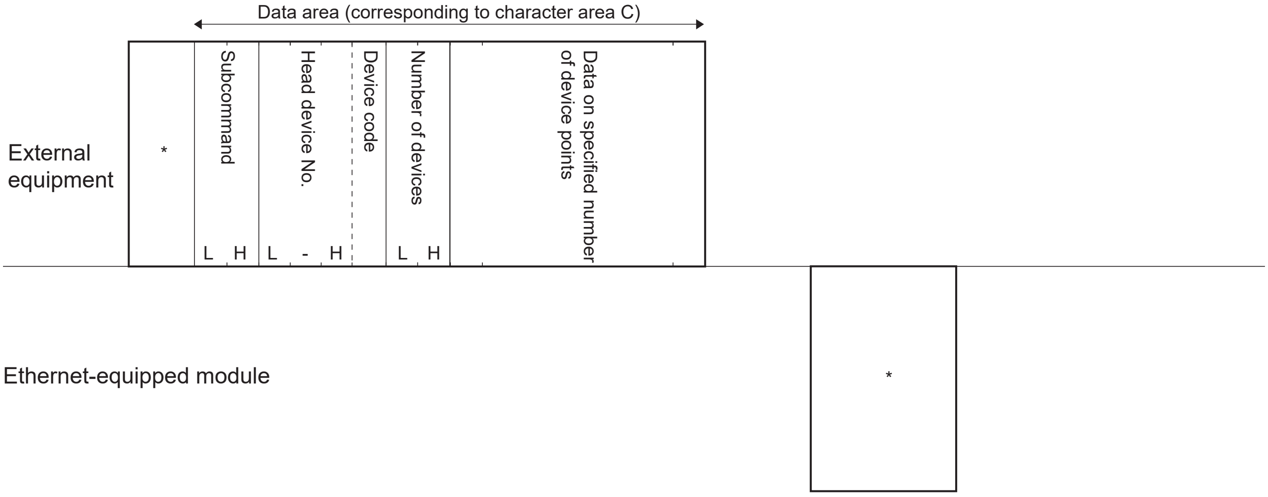
The data array and the data contents marked with * are shown in ☞ Page 562 Message format and control procedure.

Data of data area (when communicating in binary code)

■In the case of reading



■In the case of writing



The data array and the data contents marked with * are shown in ☞ Page 562 Message format and control procedure.

Data contents common to character areas

■Subcommand

Subcommands are data for specifying the unit for reading and writing, device type to be specified, and the data reading condition.

The following table shows the details of setting items.

Setting item		Description
Data size specification	Word units	• The target data is read or written in word units. • Select "0" even when the reading data or writing data does not exist in arguments of the command.
	Bit units	The target data is read or written in bit units.
Device specification format	2 digit code/6 digit number specification	Data or items related to the address specifications are expressed in the following sizes, which are the same as the existing setting. • Device code: 2-digit ASCII code, 1 byte in binary • Device number: 6-digit ASCII code, 3 bytes in binary
	4 digit code/8 digit number specification	Data or items related to the address specifications are extended to the following size. • Device code: 4-digit ASCII code, 2 byte in binary • Device number: 8-digit ASCII code, 4 bytes in binary
Device memory extension specification	Not specified	Set this when specifying devices of an Ethernet-equipped module. * Set this when not using the device memory extension specification.
	Specified	• Set this for the buffer memory specification of the intelligent function module. • This setting corresponds to the buffer memory indirect specification with index register.

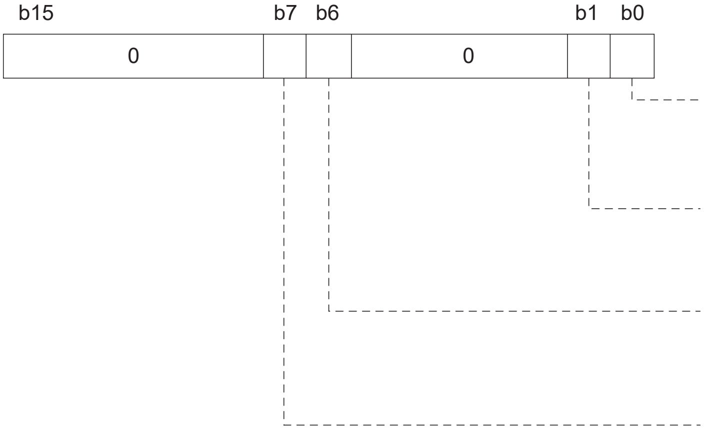
❶When communicating data in ASCII code

The value 0000H (0), or the following value (❸), is converted to a 4 digit (hexadecimal) ASCII code and sequentially transmitted beginning from the most significant digit ("0").

❷When communicating data in binary code

The value 0000H, or the following 2-byte value (❸), is used for transmission.

❸The following figure shows the specification contents of the subcommand.



<<Data size specification>>
0: Word/Not specified
1: Bit

<<Data specification format>>
0: 2 digit code/6 digit number specification
1: 4 digit code/8 digit number specification

<<Monitor condition specification>>[Not used]
0: When a function other than device read random or monitor data registration is used

<<Device memory extension specification>>
0: Device memory extension specification is not used
1: Device memory extension specification is used

❹In the following cases, the subcommand is 0000H or 0001H.

- When neither monitor condition nor device memory extension is specified.
- When using a command that cannot select monitor condition specification and device memory extension specification.

■Device code

Device codes are data for identifying the device memory to be read or written.

Device codes are shown in ☞Page 611 Device range.

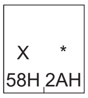
❶When communicating data in ASCII code

Device codes are converted into 2-digit ASCII code (2-digit code/6-digit number specification) or 4-digit ASCII code (4-digit code/8-digit number specification), and the device codes are sequentially sent beginning from the most significant digit. Use capitalized code for alphabetical letter in ASCII code.

Ex.

In the case of input (X)

2 digit code/6 digit number specification



4 digit code/8 digit number specification

The input relay device code "X*" is sequentially sent from "X".

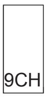
The second character "*" can be specified by a space (code: 20H).

❷When communicating data in binary code

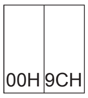
Ex.

In the case of input (X)

2 digit code/6 digit number specification



4 digit code/8 digit number specification



■Head device No. (device No.)

Data for specifying the number of the device to read data from or write data to. When specifying continuous device areas, specify the head number of the device range. Specify the head device number by the expressing method for the relevant device (octal, decimal, or hexadecimal number).

For the device number and expressing method, refer to ☞ Page 611 Device range.

❶When communicating data in ASCII code

The device number is converted to a 6-digit ASCII code (2-digit code/6-digit number specification) or 8-digit ASCII code (4-digit code/8-digit number specification), and sequentially sent beginning from the most significant digit.

The "0" column of the most significant digit (in for example "001234", this refers to "0" of the first two characters) can also be specified by a space (code: 20H).

Ex.

When the device number is "1234"

2 digit code/6 digit number specification

0	0	1	2	3	4
30H	30H	31H	32H	33H	34H

4 digit code/8 digit number specification

0	0	0	0	1	2	3	4
30H	30H	30H	30H	31H	32H	33H	34H

❷When communicating data in binary code

The 3-byte (2 digit code/6 digit number specification) or 4-byte (4 digit code/8 digit number specification) binary code with the device number specified by the device specification format is sequentially sent starting from the low byte. The device with decimal device number is sent after converting to hexadecimal device number.

Ex.

In case of internal relay M1234 and link relay B1234

2 digit code/6 digit number specification

M1234

D2H	04H	00H

B1234

34H	12H	00H

4 digit code/8 digit number specification

M1234

D2H	04H	00H	00H

B1234

34H	12H	00H	00H

Internal relay M1234 becomes 0004D2H and is sent in the order of D2H, 04H, and 00H.

Link relay B1234 becomes 001234H and is sent in the order of 34H, 12H, and 00H.

■Number of device points

This data is for specifying the number of points to be read or written when each command is executed. It must be specified within the limits of the number of points that can be processed per communication. (☞ Page 610 Commands)

❶When communicating data in ASCII code

Points are converted into 4-digit ASCII code (hexadecimal) and sequentially sent from the upper byte to lower byte. Use capitalized code for alphabetical letter.

Ex.

In the case of 5 points and 20 points

5 points

0	0	0	5
30H	30H	30H	35H

20 points

0	0	1	4
30H	30H	31H	34H

❷When communicating data in binary code

Use numerical values in 2 bytes which indicate the number of points to be processed, and send them in order from the lower byte to the upper byte.

Ex.

In the case of 5 points and 20 points

5 points

05H	00H

20 points

14H	00H

■Data on specified number of device points

This field holds the contents of the data written to the specified device, or the contents of the data read from the specified device. The data order changes depending on the processing units (words or bits).

For the data contents and order (transmission order), refer to ☞ Page 575 Transfer data in character area.

■Bit access points

This data is for specifying the number of points to be accessed in units of bits. It must be specified within the limits of the number of points processed per communication. (☞ Page 610 Commands)

❶When communicating data in ASCII code

Points are converted into 2-digit ASCII code (hexadecimal) and sequentially sent from the upper digit to lower digit. Use capitalized code for alphabetical letter in ASCII code.

Ex.

In the case of 5 points and 20 points

5 points

0	0	0	5
30H	30H	30H	35H

20 points

0	0	1	4
30H	30H	31H	34H

❷When communicating data in binary code

The 1-byte value (hexadecimal), which indicates the number of the points, is used for transmission.

Ex.

In the case of 5 points and 20 points

5 points

05H	00H

20 points

14H	00H

■Device memory extension specification (subcommand: bit7)

For details, refer to ☞ Page 912 Device Memory Extension Specification.

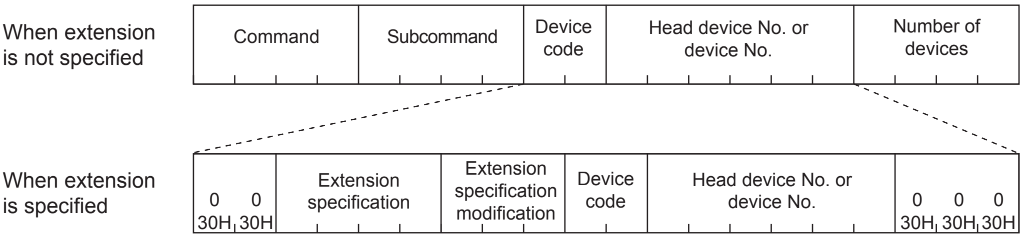
This section explains how to read or write a device from/to module access device areas and how to specify a device indirectly by using index register.

❶Message format

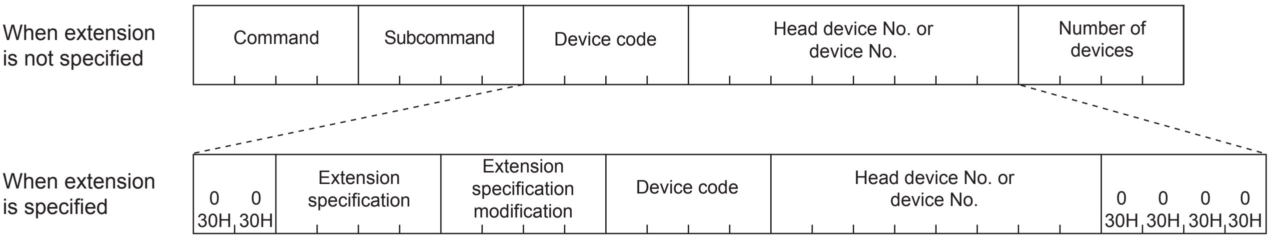
Response messages are extended as well.

- When communicating data in ASCII code

2 digit code/6 digit number specification

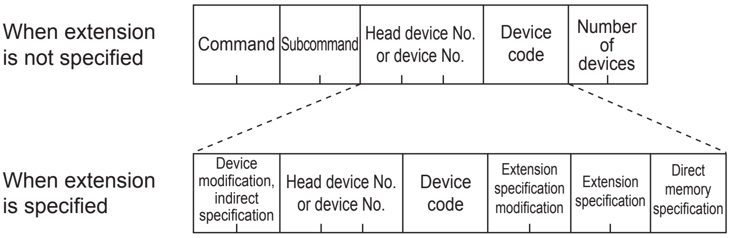


4 digit code/8 digit number specification

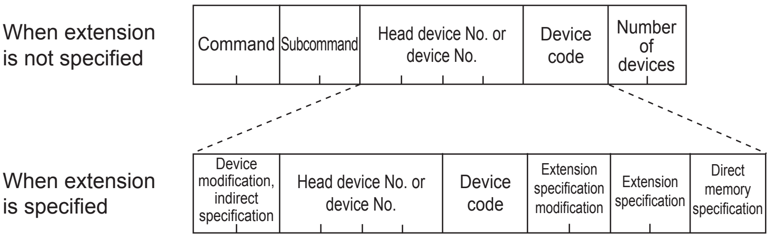


- When communicating data in binary code

2 digit code/6 digit number specification

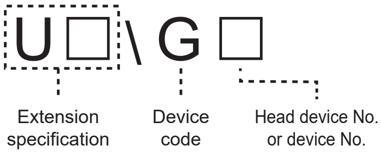


4 digit code/8 digit number specification



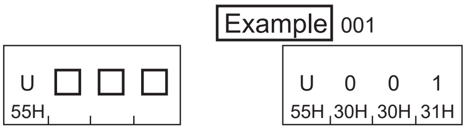
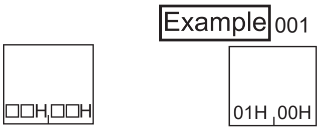
❷Module access device specification

The following shows the approach for module access device specification in programming and request data.



- Extension specification

Specify the start I/O number of intelligent function modules.

<table>
<tr><th>ASCII code</th><th>Binary code</th></tr>
<tr><td>Specify the start I/O number in hexadecimal (3-digit ASCII code). When described with 4-digits, specify the start I/O number with the upper 3-digits.
 </td><td>Specify the start I/O in hexadecimal (2 bytes). When described with 4-digits, specify the start I/O number with the upper 3-digits.
 </td></tr>
</table>

- Device code

Specify the module access device. (☞ Page 611 Device range)

- Head device No. or device No.

The format is the same as the message when extension is not specified.

37.2 1E Frame

This section describes the message format for each command when performing the data communication using the 1E frame.

Message format and control procedure

This section describes the message format and the control procedures when performing the data communication using the 1E frame.

Data format

The data format consists of header and application data.

■Request message

Header	Application data			
	Subheader	PC No.	Monitoring timer	Request data

■Response message

Header	Application data		
	Subheader	End code	Response data

Header

This header is for TCP/IP and UDP/IP.

Add the header for external equipment to Ethernet-equipped module (command message) at the external equipment side before sending the message (normally the header is added automatically).

It is not necessary to set the header for Ethernet-equipped module to external equipment (response message) by the user because the header is added by the Ethernet-equipped module automatically.

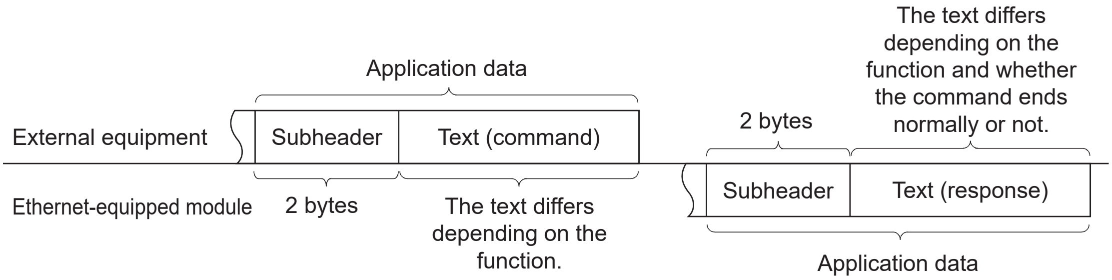
Application data

Application data is divided into subheader and text.

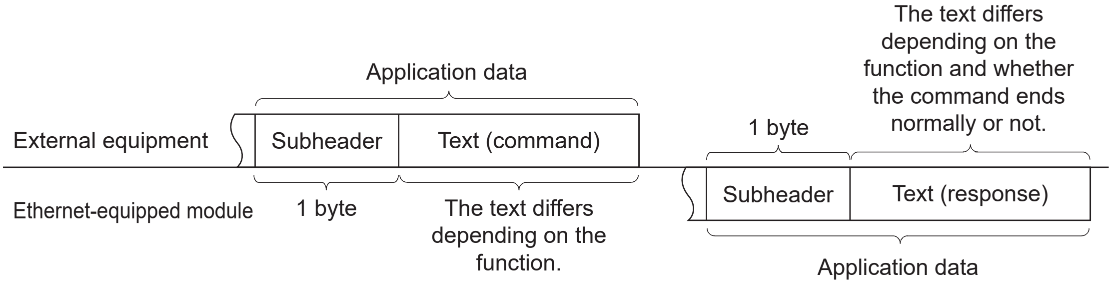
The subheader indicates whether a message is a command message or a response message. (☞ Page 588 Subheader configuration)

Text is the request data (command) and the response date (response) in each function. (☞ Page 666 1E FRAME COMMANDS)

■When communicating data in ASCII code



■When communicating data in binary code



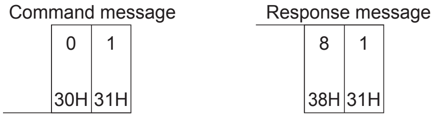
Point

It is not necessary to set the response to a command from the external equipment by the user because the response is created and sent by the Ethernet-equipped module.

Subheader configuration

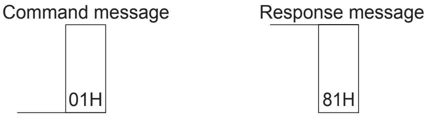
This section describes the subheader configuration.

■When communicating data in ASCII code



[Command message]
Specify the command with a 2-digit ASCII code in the order from the most significant byte to the least significant byte.
[Response message]
A 2-digit ASCII code obtained by adding 8H to the most significant byte of the command from the requesting station is sent in the order from the most significant byte to the least significant byte.

■When communicating data in binary code



[Command message]
Specify the command with a 1-byte binary code.
[Response message]
A 1-byte binary code obtained by adding 80H to the command from the requesting station is sent.

Precautions

When the communication data code is an ASCII code, if data that cannot be binary converted into a subheader is specified, the data cannot be recognized as a 1E frame message, and no response message is returned.

Control procedure

This section describes the control procedures and the format of the application data when performing the data communication.

The ☐ (Thick line) part shown in the message explanation diagram of this section are items common to all commands.

For the data contents and the data specification method of the ☐ (Thick line) part, refer to ☞ Page 591 Application data specification items.

Point

Data code (ASCII/binary) to be used when communicating, it is determined by the parameters of GX Works3.

[CPU module]

Navigation window ⇨ [Parameter] ⇨ Module model ⇨ [Module Parameter] ⇨ [Ethernet Port] ⇨ [Basic Settings] ⇨ [Own Node Settings] ⇨ "Communication Data Code"

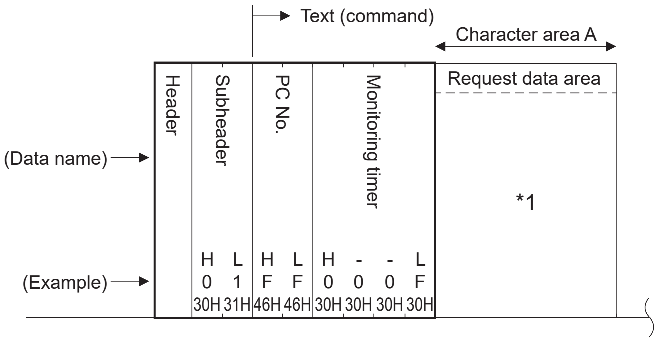
[Ethernet module]

Navigation window ⇨ [Parameter] ⇨ [Module Information] ⇨ [FX5-ENET] or [FX5-ENET/IP] ⇨ [Basic Settings] ⇨ [Own Node Settings] ⇨ "Communication Data Code"

■When communicating data in ASCII code

- When data is read from an Ethernet-equipped module by external equipment

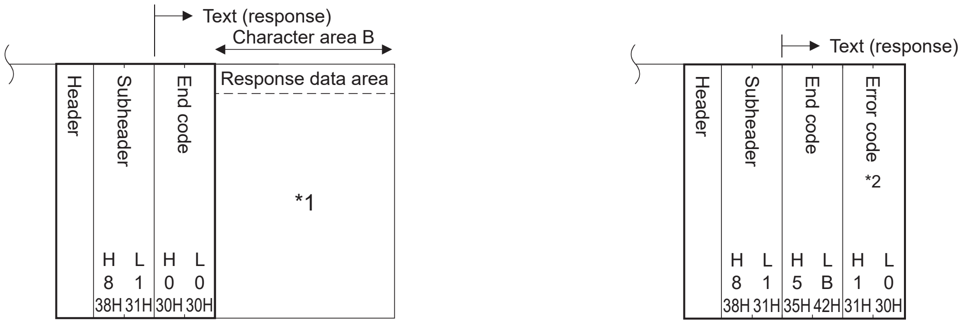
External equipment → Ethernet-equipped module (command message)



Ethernet-equipped module → External equipment (response message)

(When completed normally)

(When completed with error)

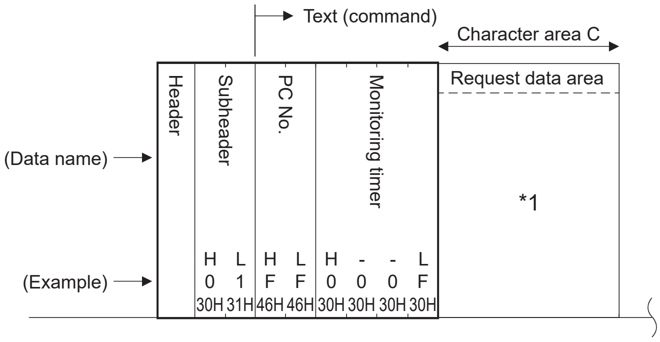


*1 The order of data items differs depending on the command. For details, refer to ☞ Page 667 Device Access.

*2 The error code is stored when the end code is "5BH".

- When data is written from external equipment to an Ethernet-equipped module

External equipment → Ethernet-equipped module (command message)



Ethernet-equipped module → External equipment (response message)

(When completed normally)

(When completed with error)

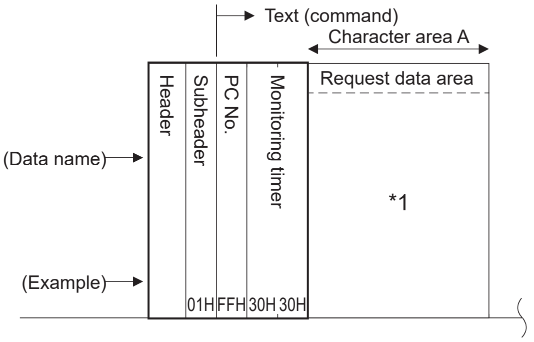
*1 The order of data items differs depending on the command. For details, refer to ☞ Page 667 Device Access.

*2 The error code is stored when the end code is "5BH".

■When communicating data in binary code

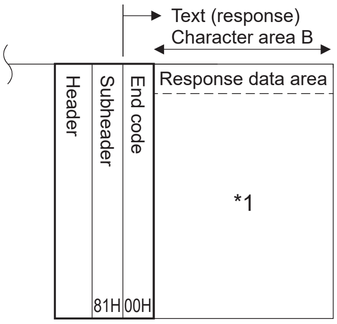
- When data is read from an Ethernet-equipped module by external equipment

External equipment → Ethernet-equipped module (command message)

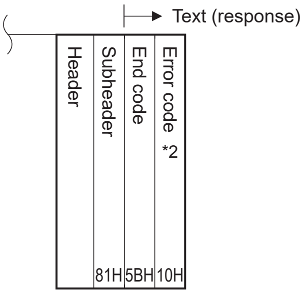


Ethernet-equipped module → External equipment (response message)

(When completed normally)



(When completed with error)

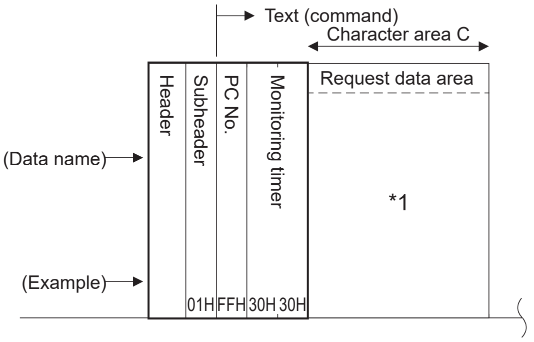


*1 The order of data items differs depending on the command. For details, refer to ☞ Page 667 Device Access.

*2 The error code is stored when the end code is "5BH".

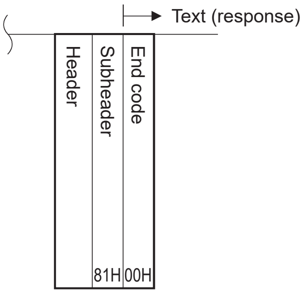
- When data is written from external equipment to an Ethernet-equipped module

External equipment → Ethernet-equipped module (command message)

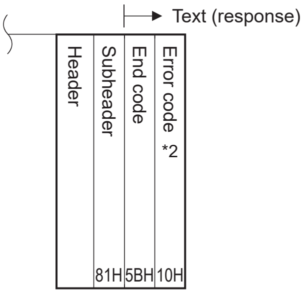


Ethernet-equipped module → External equipment (response message)

(When completed normally)



(When completed with error)



*1 The order of data items differs depending on the command. For details, refer to ☞ Page 667 Device Access.

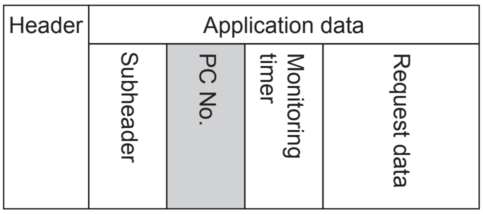
*2 The error code is stored when the end code is "5BH".

Application data specification items

This section describes the data contents and the specification method of common data items in the application data in each message when performing the data communication using the 3E frame.

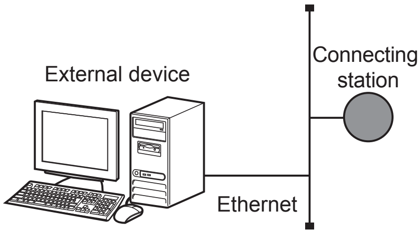
PC No.

■Request message



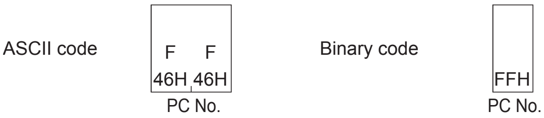
Specify the request destination PC No. to be used as an access destination in hexadecimal.

Only the connecting station (FFH) can be specified as the access destination.



Ex.

When specifying the connecting station (PC No. FFH)

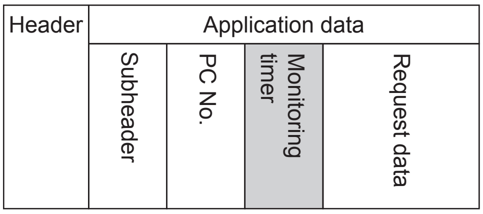


Precautions

When the communication data code is an ASCII code, if data that cannot be binary converted into a PC number is specified, the data cannot be recognized as a 1E frame message, and no response message is returned.

Monitoring timer

■Request message



This timer is used to set the wait time until a response is returned after the SLMP-compatible device that has received a request message from an external device sends a processing request to the accessed device.

- 0000H (0): Indefinitely waiting (waiting until the processing is completed)
- 0001H to FFFFH (1 to 65535)[*1]: Wait time (unit: 250ms)

[*1] Supported only for Ethernet modules.

For normal data communication, it is recommended to use this message within the setting range shown in the following table depending on the communication destination.

Access destination	Monitoring timer
Other station	01H to 28H (0.25 to 10 seconds)

■When communicating data in ASCII code

Sending from the upper byte to the lower byte.

■When communicating data in binary code

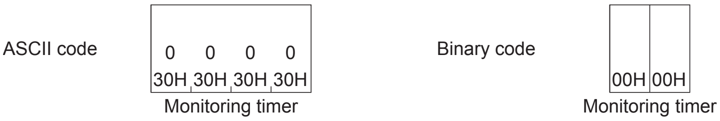
Sending from the lower byte to the upper byte.

Precautions

Specify "0000H" (indefinite wait) for the CPU module.

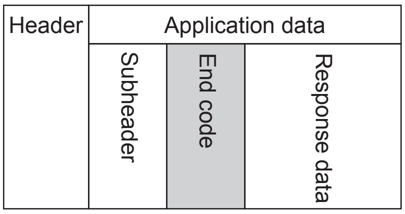
Ex.

When specifying the 0000H (Unlimited wait)



End code

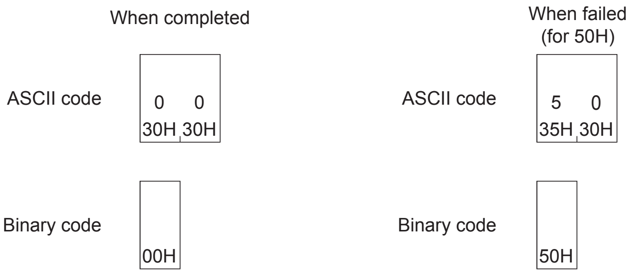
■Response message



The command processing result is stored.

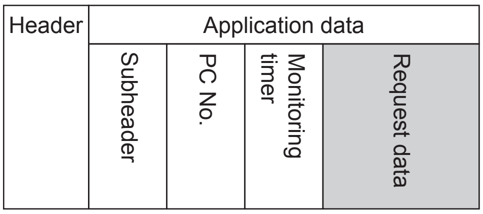
When normally completed, "00H" is stored. When completed with error, an error code set at the request destination is stored. (For the set error code and corresponding error contents, refer to ☞ Page 855 EVENT CODES and manuals of the SLMP compatible device of the response station.)

Ex.



Request data

■Request message



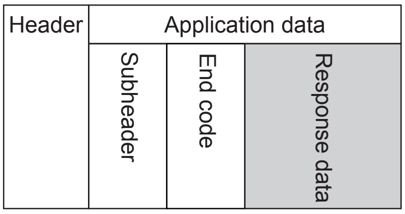
Set the data that is the argument of the command to be executed.

(Some commands do not require the request data specification.)

For details on the request data, refer to the paragraph relating to the command to be executed. (☞ Page 667 Device Access)

Response data

■Response message



The processing result of the request data is stored.

(Some commands do not return response messages.)

For details on the response data, refer to the paragraph relating to the command to be executed. (☞ Page 667 Device Access)

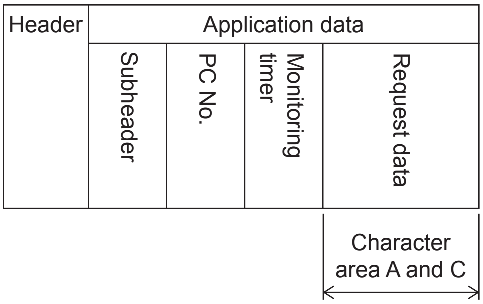
Transfer data in character area

This section describes how to transfer bit device data and word device data and data alignment in the character area sent and received between the external equipment and the Ethernet-equipped module by using each command.

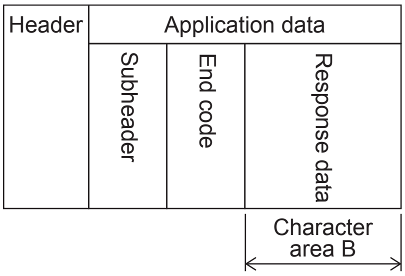
The transfer data explained below is handled as the character area A for reading and monitoring and the character area C for writing, testing, and registering the monitor data are stored.

Character area

■Request message



■Response message



Communicating data (when communicating in ASCII code)

■When bit device memory is read or written

The bit device memory is handled in 1-bit (1-point) units or in 1-word (16-point) units.

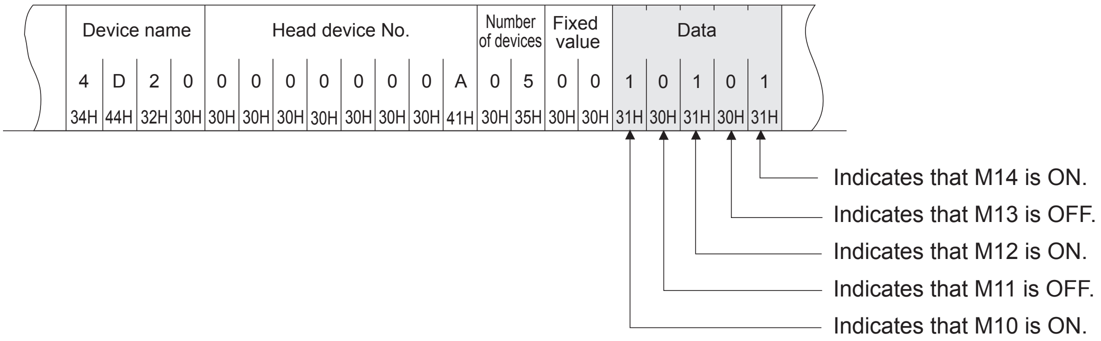
The transfer data in each case is described below.

- In 1-bit (1-point) units

When the bit device memory is handled in 1-bit units, specify 1-point (1-byte) with an ASCII code, and express "1" (31H) for ON and "0" (30H) for OFF. Specify for the number of devices starting from the head device in the order of device numbers.

Ex.

When indicating the on/off status of five devices starting from M10

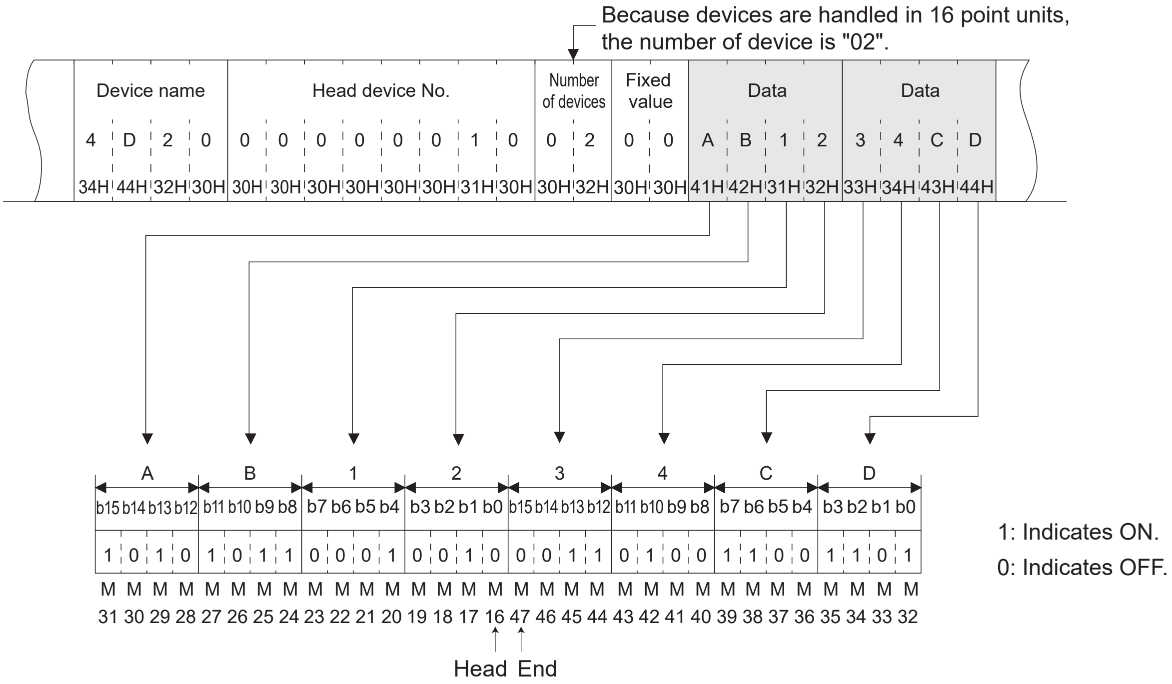


- In 1-word (16-point) units

When the bit device memory is handled in 1-word units, specify one word (16 bits) with a 4-digit ASCII code, and express 1-point with 1-bit ON/OFF. Specify for the number of devices starting from the head device in 1-word units in the order from the most significant bit to the least significant bit (b15 to b0).

Ex.

When indicating ON/OFF status of 32 points from M16

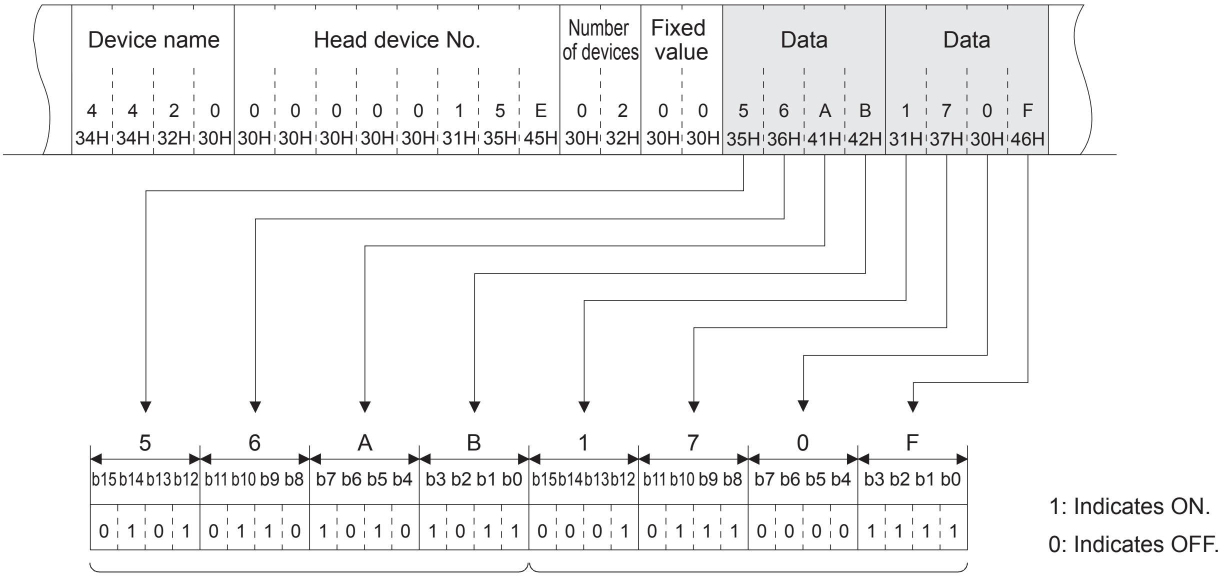


■When word device memory is read or written

One word device is expressed with a 1-word (4-byte) 4-digit ASCII code (hexadecimal). Specify for the number of devices starting from the head device in 1-word units in the order from the most significant byte to the least significant byte (b15 to b0).

Ex.

When indicating the contents stored in the data registers D350 and D351



Indicates that D350 stores "56ABH ("22187" in decimal)".

Indicates that D351 stores "170FH ("5903" in decimal)".

Point

Use capitalized code for alphabetical letter.

When data other than integer value (real number, character string), is stored in the word device memory for reading data, the stored value are read as integer value.

(Example 1) When a real number (0.75) is stored in D0 and D1, the value is read as the following integer value.

- D0 = 0000H, D1 = 3F40H

(Example 2) When a character string (12AB) is stored in D2 and D3, the character string is read as the following integer value.

- D2 = 3231H, D3 = 4241H

Communicating data (When communicating data in binary code)

■When bit device memory is read or written

The bit device memory is handled in 1-bit (1-point) units or in 1-word (16-point) units.

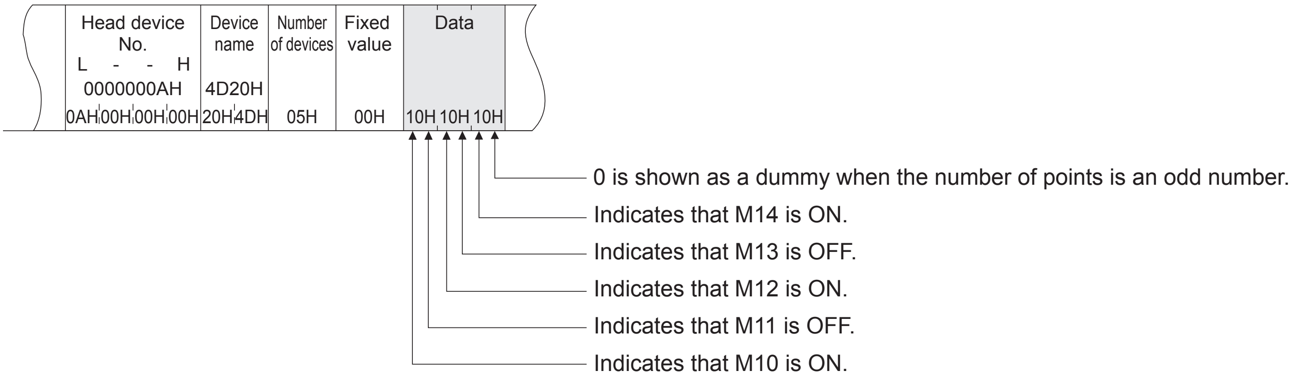
The transfer data in each case is described below.

- In 1-bit (1-point) units

When the bit device memory is handled in 1-bit units, specify 1-point, 4 bits, (two points, 1 byte) with a binary code, "1" for ON and "0" for OFF. Specify for the number of devices starting from the head device in the order of device numbers from the most significant bit.

Ex.

When indicating the on/off status of five devices starting from M10



0 is shown as a dummy when the number of points is an odd number.

Indicates that M14 is ON.

Indicates that M13 is OFF.

Indicates that M12 is ON.

Indicates that M11 is OFF.

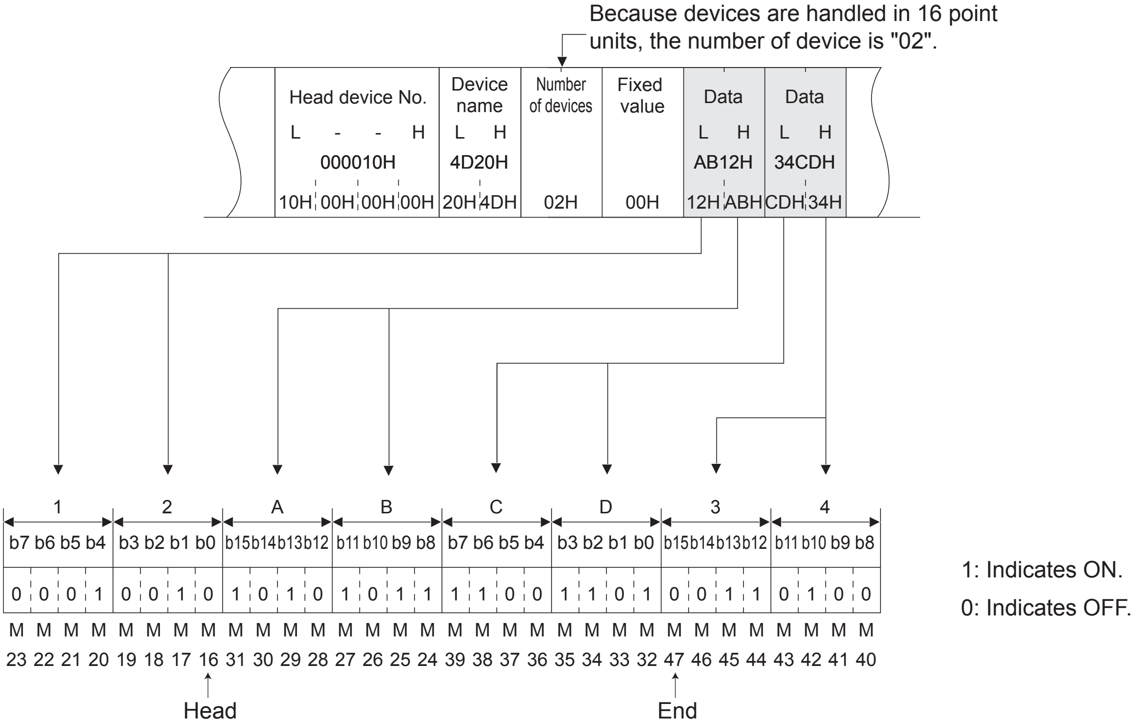
Indicates that M10 is ON.

- In 1-word (16-point) units

When the bit device memory is handled in 1-word units, specify one word (2 bytes) with a binary code, and express 1-point with 1-bit ON/OFF. Specify for the number of devices starting from the head device in 1-word units in the order from the least significant bytes (b7 to b0) to the most significant bytes (b15 to b8).

Ex.

When indicating ON/OFF status of 32 points from M16

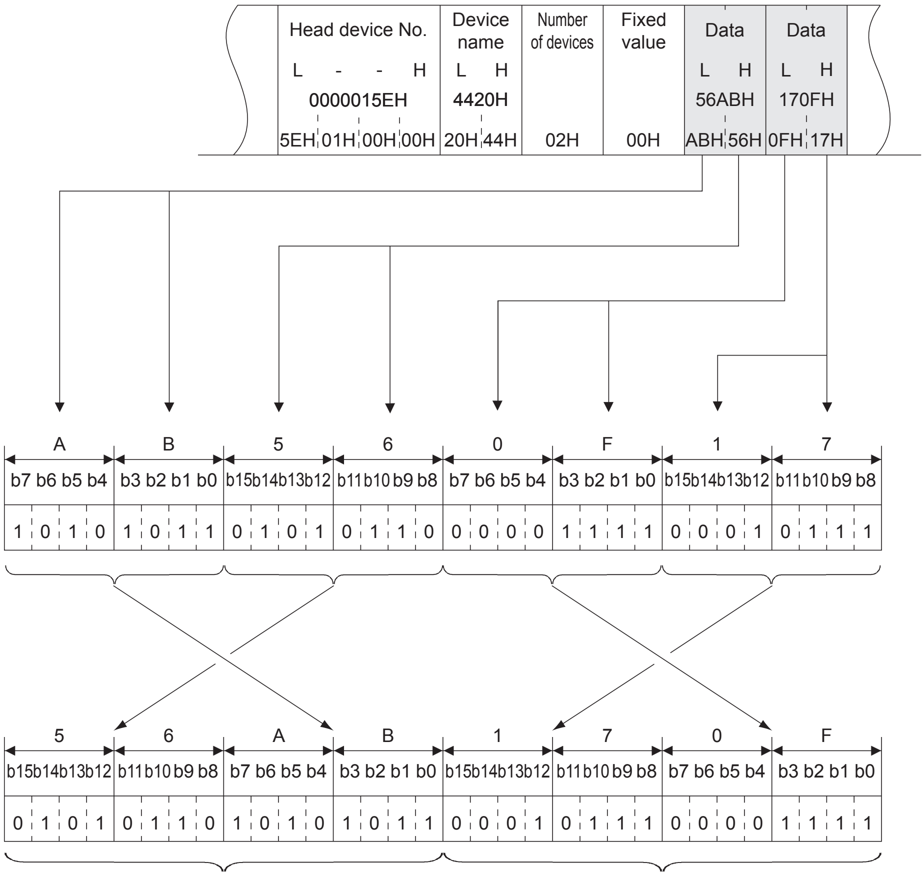


■When word device memory is read or written

One word device memory is expressed with a 1-word (2-byte) 4-digit binary code (hexadecimal). Specify for the number of devices starting from the head device in 1-word units in the order from the least significant bytes (b7 to b0) to the most significant bytes (b15 to b8).

Ex.

When indicating the contents stored in the data registers D350 and D351



Indicates that D350 stores "56ABH ("22187" in decimal)".

Indicates that D351 stores "170FH ("5903" in decimal)".

1: Indicates ON.
0: Indicates OFF.

Point

When data other than integer value (real number, character string), is stored in the word device memory for reading data, the stored value are read as integer value.

(Example 1) When a real number (0.75) is stored in D0 and D1, the value is read as the following integer value.

- D0 = 0000H, D1 = 3F40H

(Example 2) When a character string (12AB) is stored in D2 and D3, the character string is read as the following integer value.

- D2 = 3231H, D3 = 4241H

Character areas

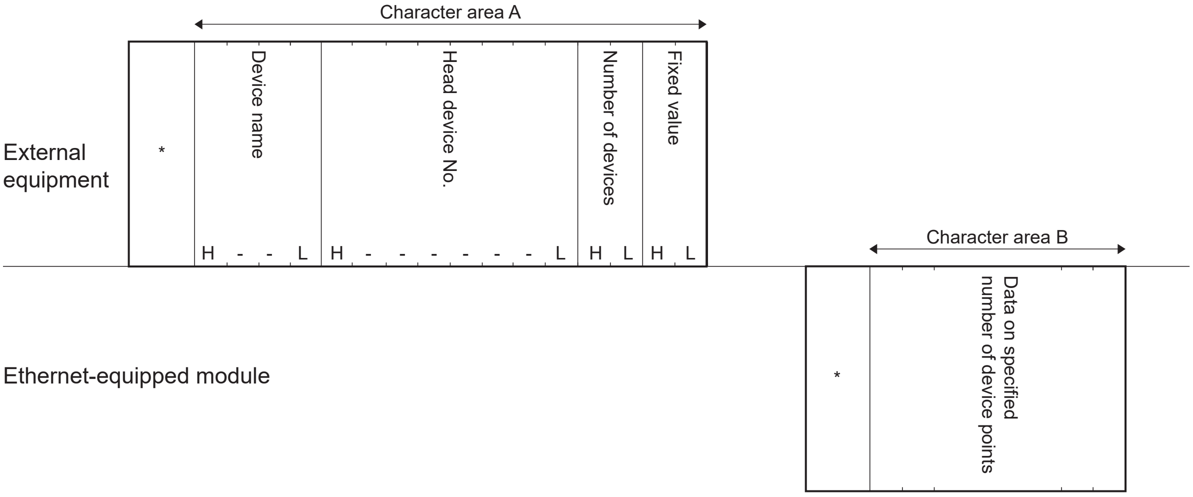
This section explains character areas in the control procedure (data area when communicating in binary code).

- Character areas differ depending on command to be used and contents to be specified. This section explains the data common to the character area when the device memory to be read or written is specified directly.
- Character area data handled only by a certain command and not by others, is explained in the section that explains the corresponding command.

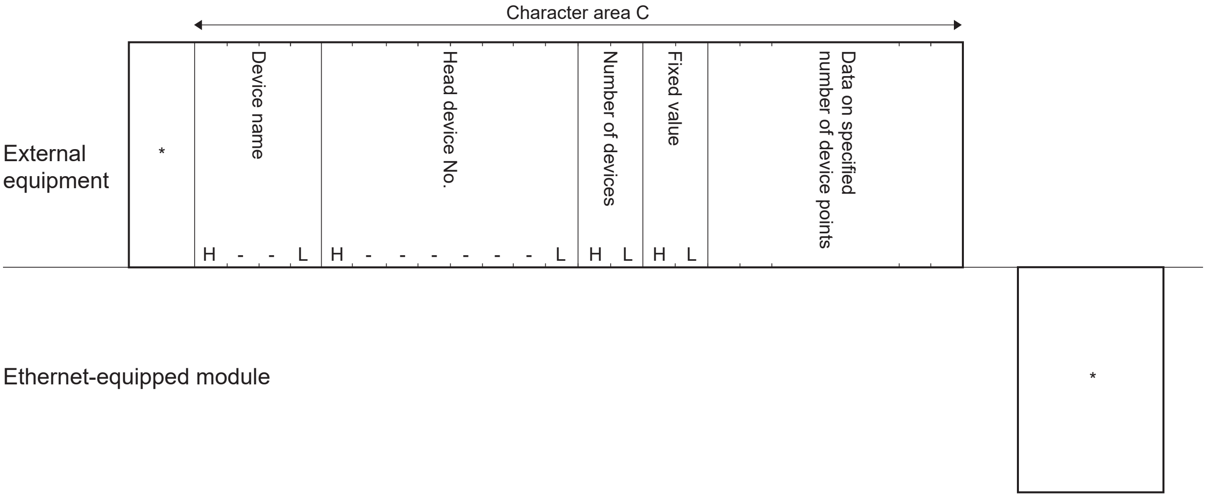
Data of character area (when communicating in ASCII code)

The data order and contents of character areas A, B, and C are identical when the same command is used under the same conditions in the control procedure when communicating using ASCII code.

■In the case of reading



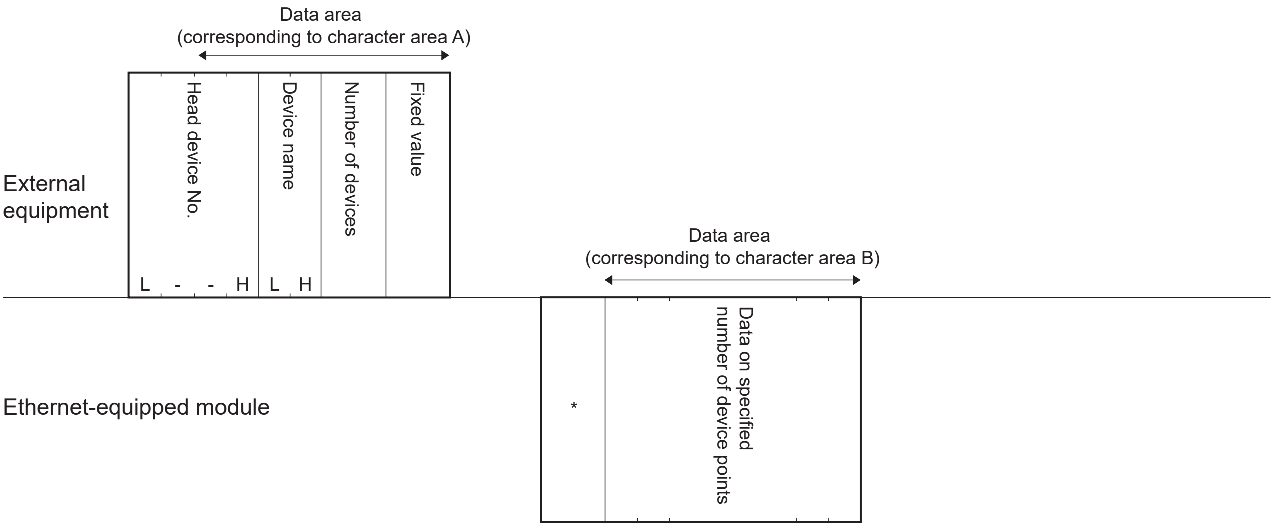
■In the case of writing



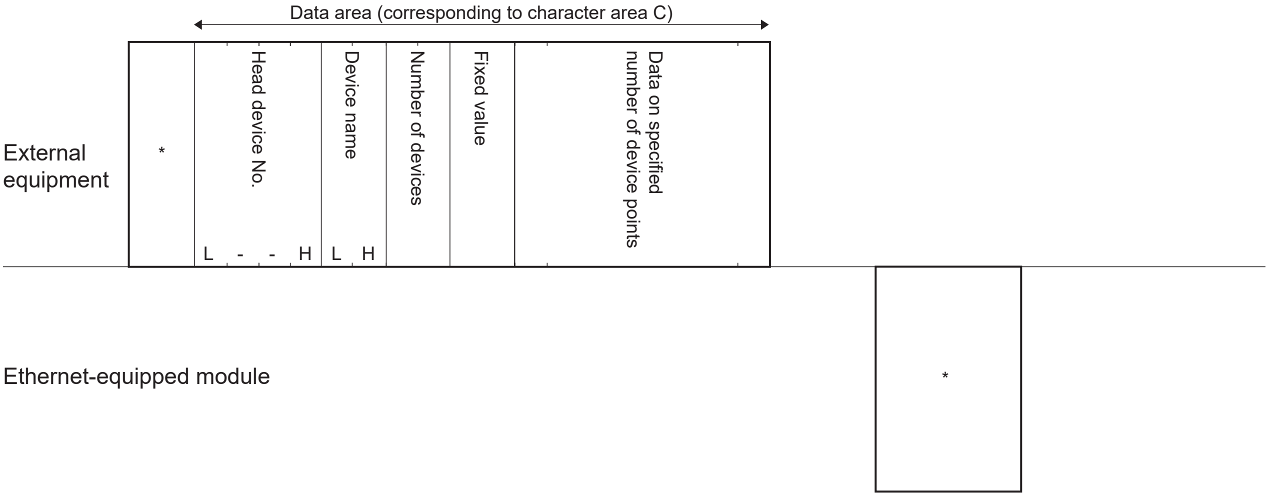
The data array and the data contents marked with * are shown in ☞ Page 587 Message format and control procedure.

Data of data area (when communicating in binary code)

■In the case of reading



■In the case of writing



The data array and the data contents marked with * are shown in ☞ Page 587 Message format and control procedure.

Data contents common to character areas

■Device name (Device code)

Device codes are data for identifying the device memory to be read or written.

Device codes are shown in ☞ Page 668 Device range.

❶When communicating data in ASCII code

Device codes are converted into 4-digit ASCII code (hexadecimal), and the device codes are sequentially sent beginning from the most significant digit. Use capitalized code for alphabetical letter in ASCII code.

Ex.

In the case of input (X)

5	8	2	0
35H	38H	32H	30H

❷When communicating data in binary code

Specify the device code with a 2-byte binary code in the order from the least significant byte to the most significant byte.

Ex.

In the case of input (X)

5820H	
20H	58H

■Head device No. (device No.)

Data for specifying the number of the device to read data from or write data to. When specifying continuous device areas, specify the head number of the device range. The method for specifying the head device number varies depending on the expressing method for the relevant device.

- Octal: Specify in octal notation.
- Decimal: Specify after converting into hexadecimal.
- Hexadecimal: Specify in hexadecimal notation.

Use capitalized code for alphabetical letter.

For the device number and expressing method, refer to ☞ Page 668 Device range.

❶When communicating data in ASCII code

Device codes are converted into 8-digit ASCII code, and the device codes are sequentially sent beginning from the most significant digit.

Ex.

When the device number is "1234 (4D2H)"

0	0	0	0	0	4	D	2
30H	30H	30H	30H	30H	34H	44H	32H

❷When communicating data in binary code

Specify the device number as a 4-byte binary code in the specified device specification format in the order from the least significant byte to the most significant byte after converting into hexadecimal.

Ex.

When the device number is "1234 (4D2H)"

4D2H			
D2H	04H	00H	00H

■Number of device points

This data is for specifying the number of points to be read or written when each command is executed. It must be specified within the limits of the number of points that can be processed per communication. (☞ Page 667 Commands)

❶When communicating data in ASCII code

Points are converted into 4-digit ASCII code (hexadecimal), and the device codes are sequentially sent from the upper byte to lower byte. Use capitalized code for alphabetical letter.

Ex.

In the case of 5 points and 20 points

5 points	20 points
05	1E
30H 35H	31H 45H

❷When communicating data in binary code

Specify the number of points with a 1-byte binary code after converting into hexadecimal.

Ex.

In the case of 5 points and 20 points

5 points	20 points
05H	14H

■Fixed value

A fixed value (00H) must be specified after the number of device points.

❶When communicating data in ASCII code

Specify the fixed value after converting into a 2-digit ASCII code.

Ex.

00
30H 30H

❷When communicating data in binary code

Specify the fixed value with a 1-byte binary code.

Ex.

00H

38 3E FRAME COMMANDS

This chapter explains 3E frame commands of SLMP.

For parts of the transmission message other than the command part, refer to ☞ Page 562 3E Frame.

38.1 List of Commands and Functions

This section describes commands and functions when accessing from the external equipment to the Ethernet-equipped module.

CPU module

Name	Command	Subcommand	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	• 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	• 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	• 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	• 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	• 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	• 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	• 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: 1792 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)

Name	Command	Subcommand	Processing content	Number of points processed per communication
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
		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	• 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 points BIN: (Word access points) × 12 + (double-word access points) × 14 ≤ 1920 points
		0081H	• 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 points BIN: ((Word access points) × 12 + (double-word access points) × 14) × 2 ≤ 1920 points
		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 points BIN: ((Word access points) × 12 + (double-word access points) × 14) × 2 ≤ 1920 points
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 points and (Total points of each block of word device + total points of each block of bit device) × 2 ≤ 960 points BIN: Number of word device blocks + number of bit device blocks ≤ 120 points and Total points of each block of word device + total points of each block of bit device ≤ 960 points

Name	Command	Subcommand	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 points and (Total points of each block of word device + total points of each block of bit device) × 2 ≤ 960 points BIN: (Number of word device blocks + number of bit device blocks) × 2 ≤ 120 points and Total points of each block of word device + total points of each block of bit device ≤ 960 points
		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 points and (Total points of each block of word device + total points of each block of bit device) × 2 ≤ 960 points BIN: (Number of word device blocks + number of bit device blocks) × 2 ≤ 120 points and Total points of each block of word device + total points of each block of bit device ≤ 960 points
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 points and ((Number of word device blocks + number of bit device blocks) × 4 + Total points of each block of word device + total points of each block of bit device) × 2 ≤ 760 points BIN: Number of word device blocks + number of bit device blocks ≤ 120 points and (Number of word device blocks + number of bit device blocks) × 4 + Total points of each block of word device + total points of each block of bit device ≤ 760 points
		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 points and ((Number of word device blocks + number of bit device blocks) × 4 + Total points of each block of word device + total points of each block of bit device) × 2 ≤ 760 points BIN: (Number of word device blocks + number of bit device blocks) × 2 ≤ 120 points and (Number of word device blocks + number of bit device blocks) × 4 + Total points of each block of word device + total points of each block of bit device ≤ 760 points

Name	Command	Subcommand	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 points and ((Number of word device blocks + number of bit device blocks) × 4 + Total points of each block of word device + total points of each block of bit device) × 2 ≤ 760 points BIN: (Number of word device blocks + number of bit device blocks) × 2 ≤ 120 points and (Number of word device blocks + number of bit device blocks) × 4 + Total points of each block of word device + total points of each block of bit device ≤ 760 points
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, FX5-CCLGN-MS, FX5-CCLIEF, FX5-40SSC-G, FX5-80SSC-G

Name	Command	Subcommand	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	• 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	• 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	• 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	• 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	• 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: 1792 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). • 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)

Name	Command	Subcommand	Processing content	Number of points processed per communication
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	• 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	• 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
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	• 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 points BIN: (Word access points) × 18 + (double-word access points) × 20 ≤ 1962 points
		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	• 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: ((Word access points) × 12 + (double-word access points) × 14) × 4 ≤ 1920 points BIN: ((Word access points) × 12 + (double-word access points) × 14) × 2 ≤ 1920 points
		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	• 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: ((Word access points) × 12 + (double-word access points) × 14) × 4 ≤ 1920 points BIN: ((Word access points) × 12 + (double-word access points) × 14) × 2 ≤ 1920 points

Name	Command	Subcommand	Processing content	Number of points processed per communication
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: (Number of word device blocks + number of bit device blocks) × 2 ≤ 120 points and (Total points of each block of word device + total points of each block of bit device) × 2 ≤ 960 points BIN: Number of word device blocks + number of bit device blocks ≤ 120 points and Total points of each block of word device + total points of each block of bit device ≤ 960 points
		0080H	• 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 points and (Total points of each block of word device + total points of each block of bit device) × 2 ≤ 960 points BIN: (Number of word device blocks + number of bit device blocks) × 2 ≤ 120 points and Total points of each block of word device + total points of each block of bit device ≤ 960 points
		0082H	• 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 points and (Total points of each block of word device + total points of each block of bit device) × 2 ≤ 960 points BIN: (Number of word device blocks + number of bit device blocks) × 2 ≤ 120 points and Total points of each block of word device + total points of each block of bit device ≤ 960 points
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 points and ((Number of word device blocks + number of bit device blocks) × 4 + Total points of each block of word device + total points of each block of bit device) × 2 ≤ 770 points BIN: Number of word device blocks + number of bit device blocks ≤ 120 points and (Number of word device blocks + number of bit device blocks) × 4 + Total points of each block of word device + total points of each block of bit device ≤ 770 points
		0080H	• 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 points and ((Number of word device blocks + number of bit device blocks) × 4 + Total points of each block of word device + total points of each block of bit device) × 2 ≤ 770 points BIN: (Number of word device blocks + number of bit device blocks) × 2 ≤ 120 points and (Number of word device blocks + number of bit device blocks) × 4 + Total points of each block of word device + total points of each block of bit device ≤ 770 points

Name	Command	Subcommand	Processing content	Number of points processed per communication
Device Write Block[3]	1406H	0082H	• 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 points and ((Number of word device blocks + number of bit device blocks) × 4 + Total points of each block of word device + total points of each block of bit device) × 2 ≤ 770 points BIN: (Number of word device blocks + number of bit device blocks) × 2 ≤ 120 points and (Number of word device blocks + number of bit device blocks) × 4 + Total points of each block of word device + total points of each block of bit device ≤ 770 points
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.

38.2 Device Access

This section explains the control procedure specification method and shows a specification example when the device memory is read and written.

Commands

This section explains commands when the device memory is read or written.

Commands

Function		Command (Subcommand)	Processing content
Device Read (Batch)	Bit units	0401 (00□1)	Reads bit devices in 1 point units.
	Word units	0401 (00□0)	Reads bit devices in 16 point units.
			Reads word devices in 1 point units.
Device Write (Batch)	Bit units	1401 (00□1)	Writes bit devices in 1-point units.
	Word units	1401 (00□0)	Writes bit devices in 16-point units.
			Writes word devices in 1-point units.
Device Read Random	Word units	0403 (00□0)	Reads bit devices specified randomly in 16-point units or 32-point units.
			Reads word devices specified randomly in 1-point units or 2-point units.
Device Write Random	Bit units	1402 (00□1)	Sets or resets device memory to bit devices specified randomly in 1-point units.
	Word units	1402 (00□0)	Sets or resets device memory to bit devices specified randomly in 16-point units or 32-point units
			Writes device memory to word devices specified randomly in 1-point units or 2-point units.
Device Read Block	Word units	0406 (00□0)	Sets n point(s) in the word device or bit device (one point is specified by 16-bit) as 1 block, specifies multiple blocks randomly and reads the device memory.
Device Write Block	Word units	1406 (00□0)	Sets n point(s) in the word device or bit device (one point is specified by 16-bit) as 1 block, specifies multiple blocks randomly and writes the device memory.

Device range

This section shows accessible CPU module device.
Specify the device and device number range that exist in the module targeted for data read or write.

In the case of Ethernet-equipped module

Classification	Device		Type	Device code*1 (Device specification format: 4 digit code/8 digit number specification)		Device No.		Device compatibility*2
				ASCII code	Binary code			
Internal user device	Input		Bit	X* (X***)	9CH (9C00H)	Specify within the device No. range of the module for access destination.	*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)			—

Classification	Device		Type	Device code*1 (Device specification format: 4 digit code/8 digit number specification)		Device No.		Device compatibility*2
				ASCII code	Binary code			
Internal user device	Counter	Contact	Bit	CS (CS**)	C4H (C400H)	Specify within the device No. range of the module for access destination.	Decimal	○
		Coil	Bit	CC (CC**)	C3H (C300H)			○
		Current value	Word	CN (CN**)	C5H (C500H)			○
	Long counter	Contact	Bit	— (LCS*)	55H (5500H)		Decimal	○
		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)		Hexadecimal	○
System device	Special relay		Bit	SM (SM**)	91H (9100H)		Decimal	○
	Special register		Word	SD (SD**)	A9H (A900H)		Decimal	○
	Function input		Bit	—	—	—	Hexadecimal	—
	Function output			—	—		Hexadecimal	—
	Function register		Word	—	—		Decimal	—
Index register			Word	Z* (Z***)	CCH (CC00H)	Specify within the device No. range of the module for access destination.	Decimal	○
Long index register			Double Word	LZ (LZ**)	62H (6200H)		Decimal	○
File register			Word	R* (R***)	AFH (AF00H)		Decimal	○
				ZR (ZR**)	B0H (B000H)		Decimal	—
Link direct device*4	Link input		Bit	X* (X***)	9CH (9C00H)		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)		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" at 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 It is necessary to make "Device memory extension specification" of the sub-command to ON (1).

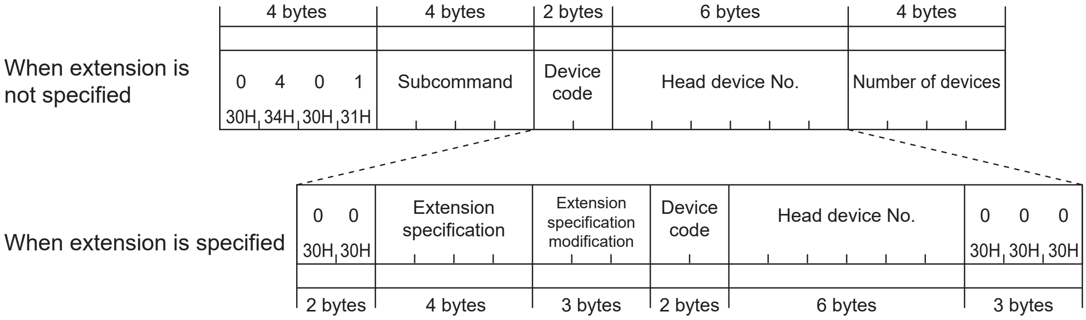
Device Read (Batch)

Data in devices are read in a batch.

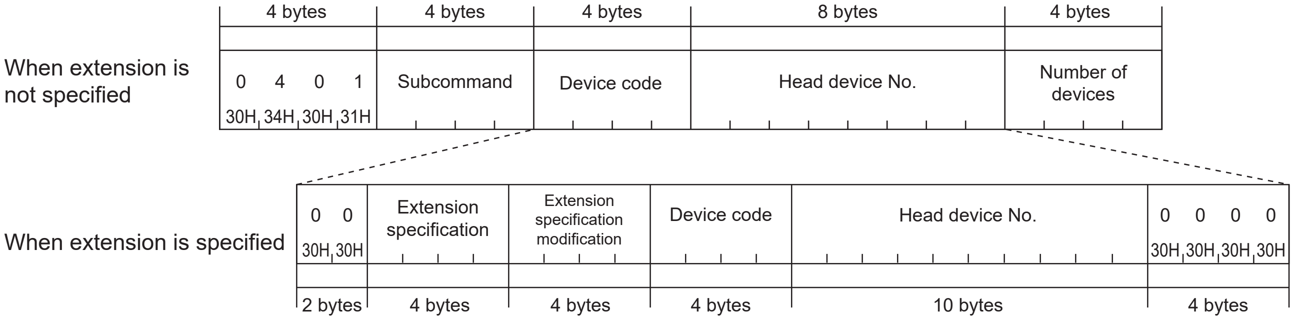
Request data

■When communicating data in ASCII code

2 digit code/6 digit number specification

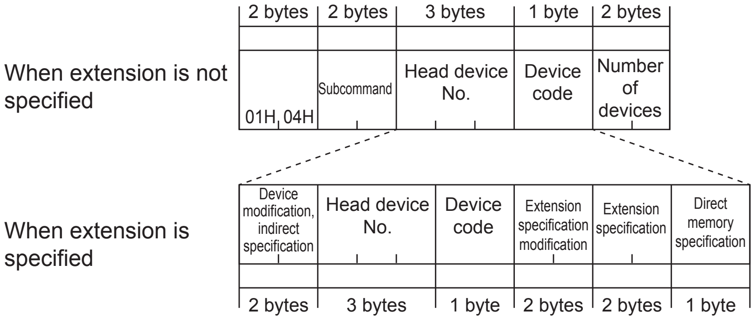


4 digit code/8 digit number specification

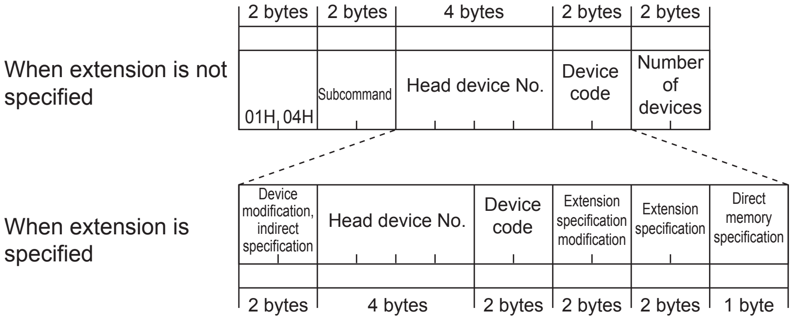


■When communicating data in binary code

2 digit code/6 digit number specification



4 digit code/8 digit number specification



■Subcommand

Specify the subcommand selected from the item.

Item			Subcommand					
Data size specification	**Device specification format**	**Device memory extension specification**	**ASCII code (Upper column: characters, lower column: character code)**				**Binary code**	
Bit units	2 digit code/6 digit number specification	Not specified	0	0	0	1	01H	00H
			30H	30H	30H	31H		
		Specified	0	0	8	1	81H	00H
			30H	30H	38H	31H		
	4 digit code/8 digit number specification	Specified	0	0	8	3	83H	00H
			30H	30H	38H	33H		
Word units	2 digit code/6 digit number specification	Not specified	0	0	0	0	00H	00H
			30H	30H	30H	30H		
		Specified	0	0	8	0	80H	00H
			30H	30H	38H	30H		
	4 digit code/8 digit number specification	Specified	0	0	8	2	82H	00H
			30H	30H	38H	32H		

■Device code

Specify the device code that corresponds to the device type to be read. (☞ Page 611 Device range)

Precautions

Batch Read (0401H) is not applicable to double word devices or long index registers (LZ).

■Device No.

Specify the head number of target device of reading.

■Number of devices

Specify the number of target device points of reading.

Item	Number of devices	
	ASCII code	**Binary code**
When reading data in bit units	1 to 1792 points	1 to 3584 points
When reading data in word units	1 to 480 points	1 to 960 points

Response data

The read device value is stored in hexadecimal. The data order differs depending on the type of code, ASCII code or binary code.

Read data

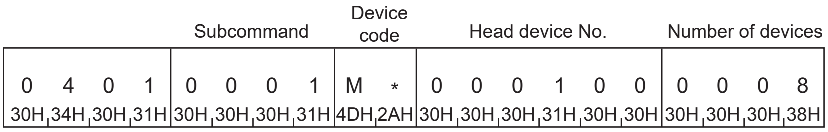
Communication example

■When reading data in bit units

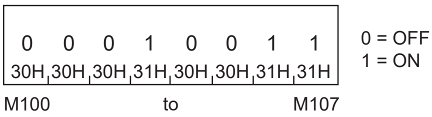
M100 to M107 are read.

- When communicating data in ASCII code

(Request data)

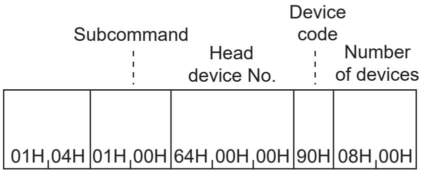


(Response data)

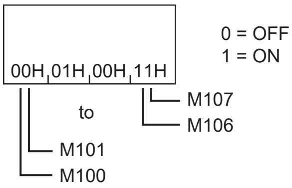


- When communicating data in binary code

(Request data)



(Response data)

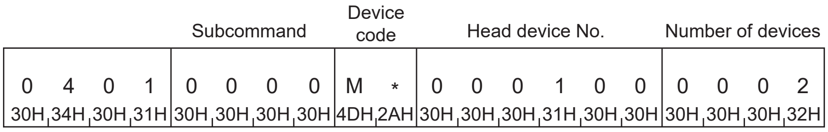


■When reading data in word units (bit device)

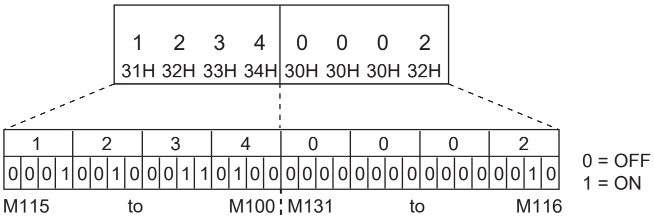
M100 to M131 (2-word) are read.

- When communicating data in ASCII code

(Request data)



(Response data)



- When communicating data in binary code

(Request data)

	Subcommand	Head device No.	Device code	Number of devices
01H,04H	00H,00H	64H,00H,00H	90H	02H,00H

(Response data)

34H 12H 02H 00H

3	4	1	2	0	2	0	0
0 0 1 1	0 1 0 0	0 0 0 1	0 0 1 0	0 0 0 0	0 0 1 0	0 0 0 0	0 0 0 0
M107 to M100		M115 to M108		M123 to M116		M131 to M124	

0 = OFF
1 = ON

■When reading data in word units (word device)

Values in T100 to T102 are read.

It is supposed that 4660 (1234H) is stored in T100, 2 (2H) is stored in T101, and 7663 (1DEFH) is stored in T102.

- When communicating data in ASCII code

(Request data)

	Subcommand	Device code	Head device No.	Number of devices
0 4 0 1 30H,34H,30H,31H	0 0 0 0 30H,30H,30H,30H	T N 54H,4EH	0 0 0 1 0 0 30H,30H,30H,31H,30H,30H	0 0 0 3 30H,30H,30H,33H

(Response data)

1 2 3 4 31H 32H 33H 34H	0 0 0 2 30H 30H 30H 32H	1 D E F 31H 44H 45H 46H
T100	T101	T102

- When communicating data in binary code

(Request data)

	Subcommand	Head device No.	Device code	Number of devices
01H,04H	00H,00H	64H,00H,00H	C2H	03H,00H

(Response data)

34H,12H	02H,00H	EFH,1DH
T100	T101	T102

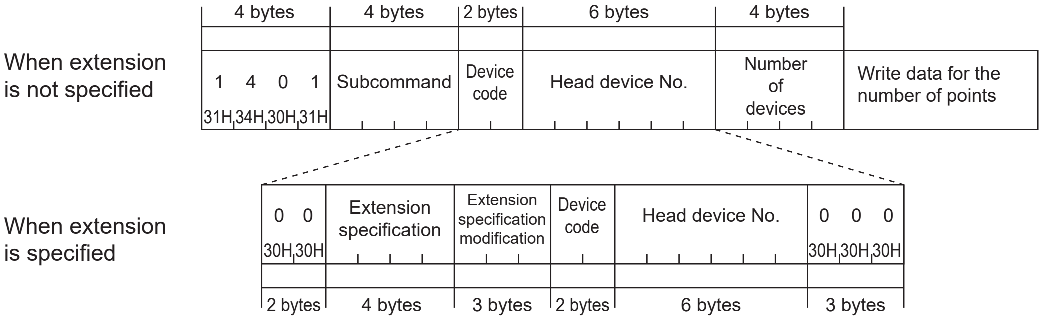
Device Write (Batch)

Data in devices are written in a batch.

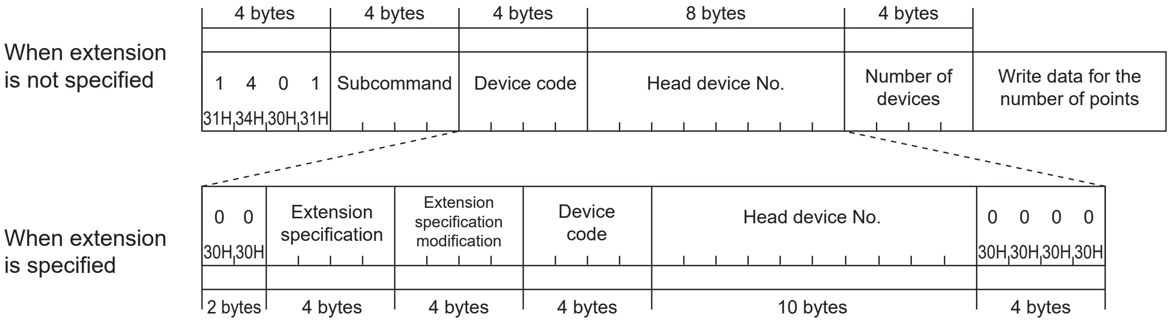
Request data

■When communicating data in ASCII code

2 digit code/6 digit number specification

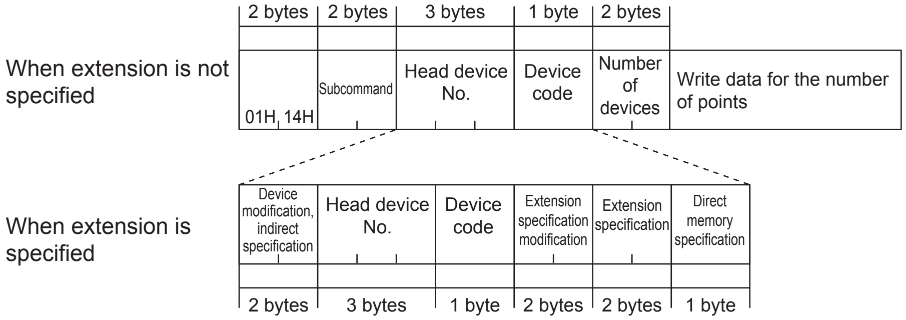


4 digit code/8 digit number specification

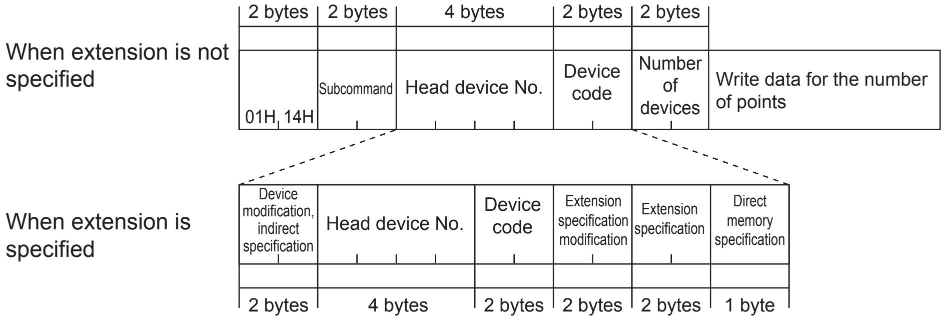


■When communicating data in binary code

2 digit code/6 digit number specification



4 digit code/8 digit number specification



■Subcommand

Specify the subcommand selected from the item.

Item			Subcommand					
Data size specification	Device specification format	Device memory extension specification	ASCII code (Upper column: characters, lower column: character code)				Binary code	
Bit units	2 digit code/6 digit number specification	Not specified	0	0	0	1	01H	00H
			30H	30H	30H	31H		
		Specified	0	0	8	1	81H	00H
			30H	30H	38H	31H		
	4 digit code/8 digit number specification	Specified	0	0	8	3	83H	00H
			30H	30H	38H	33H		
Word units	2 digit code/6 digit number specification	Not specified	0	0	0	0	00H	00H
			30H	30H	30H	30H		
		Specified	0	0	8	0	80H	00H
			30H	30H	38H	30H		
	4 digit code/8 digit number specification	Specified	0	0	8	2	82H	00H
			30H	30H	38H	32H		

■Device code

Specify the device code that corresponds to the device type to be written. (☞ Page 611 Device range)

Precautions

Batch Write (1401H) is not applicable to double word devices or long index registers (LZ).

■Device No.

Specify the head number of target device of writing.

■Number of devices

Specify the number of target device points of writing.

Item	Number of devices		
	ASCII code	Binary code	
		CPU module	Ethernet module, FX5-CCLGNMS, FX5-CCLIEF, FX5-40SSCG, FX5-80SSC-G
When writing data in bit units	1 to 1792 points	1 to 3584 points	
When writing data in word units	1 to 480 points	1 to 960 points	1 to 949 points

■Write data

Specify value to be written to a device for the number of points specified in "Device point".

Response data

There is no response data for the Device Write command.

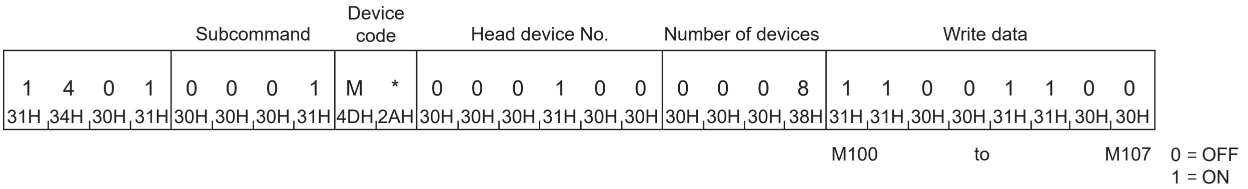
Communication example

■When writing data in bit units

Values are written to M100 to M107.

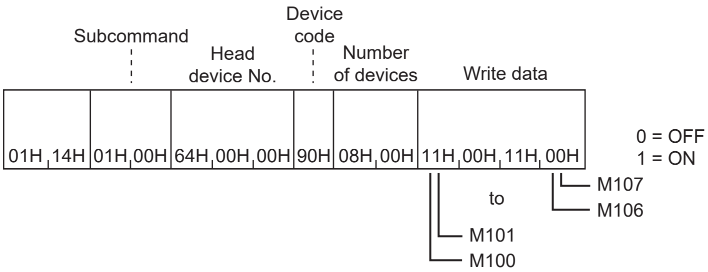
- When communicating data in ASCII code

(Request data)



- When communicating data in binary code

(Request data)

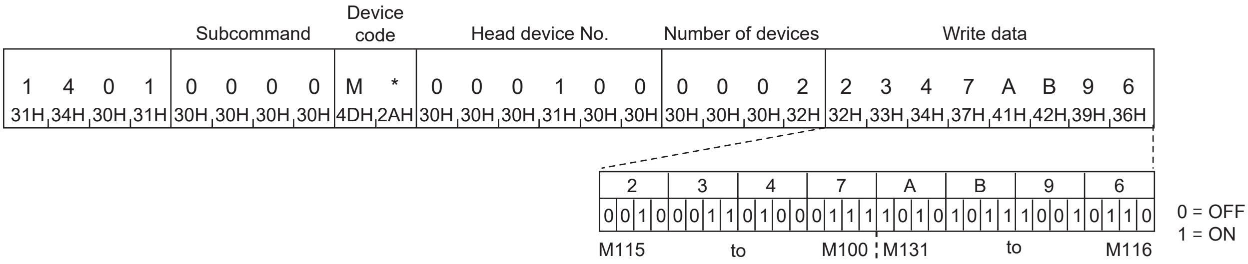


■When writing data in word units (bit device)

Values are written to M100 to M131 (2-word).

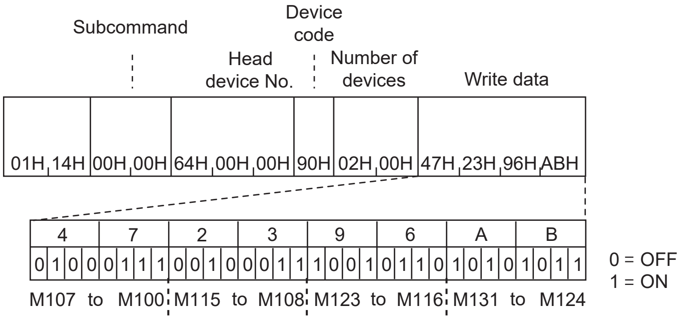
- When communicating data in ASCII code

(Request data)



- When communicating data in binary code

(Request data)



■When writing data in word units (word device)

6549 (1995H) is written in D100, 4610 (1202H) is written in D101, and 4400 (1130H) is written in D102.

- When communicating data in ASCII code

(Request data)

	Subcommand	Device code	Head device No.	Number of devices	Write data
1 4 0 1	0 0 0 0	D *	0 0 0 1 0 0	0 0 0 3	1 9 9 5 1 2 0 2 1 1 3 0
31H 34H 30H 31H	30H 30H 30H 30H	44H 2AH	30H 30H 30H 31H 30H 30H	30H 30H 30H 33H	31H 39H 39H 35H 31H 32H 30H 32H 31H 31H 33H 30H
					D100 D101 D102

- When communicating data in binary code

(Request data)

	Subcommand	Head device No.	Device code	Number of devices	Write data
01H 14H	00H 00H	64H 00H 00H	A8H	03H 00H	95H 19H 02H 12H 30H 11H
					D100 D101 D102

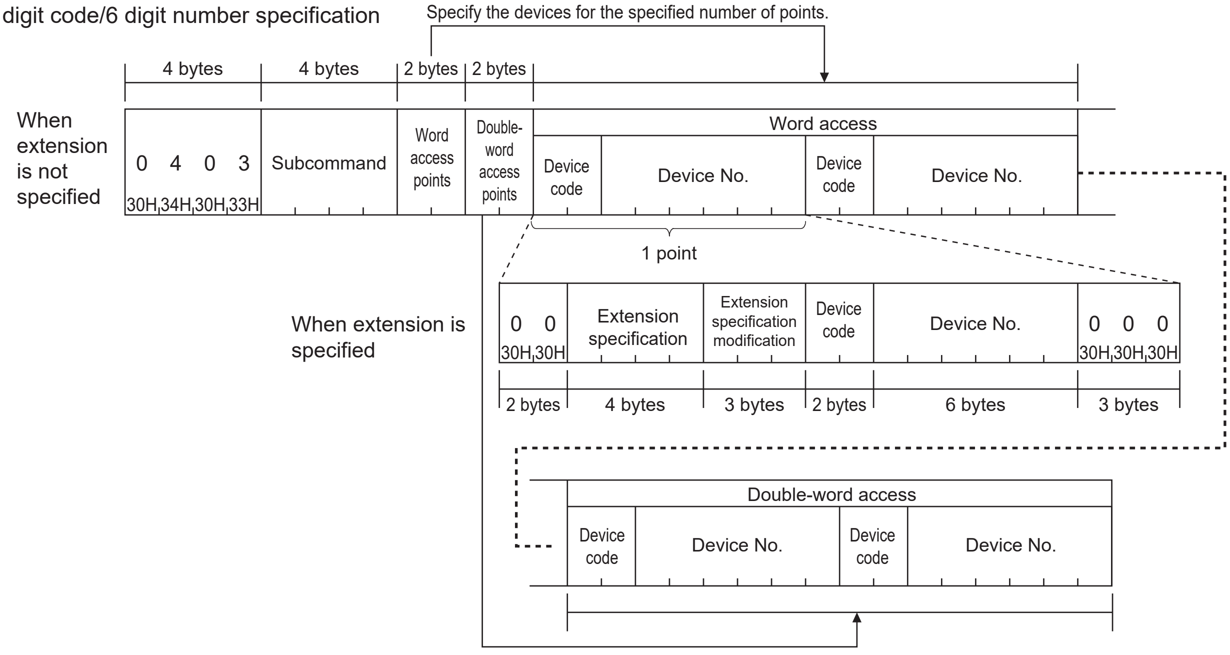
Device Read Random

This command specifies the device No. randomly and reads the device value.

Request data

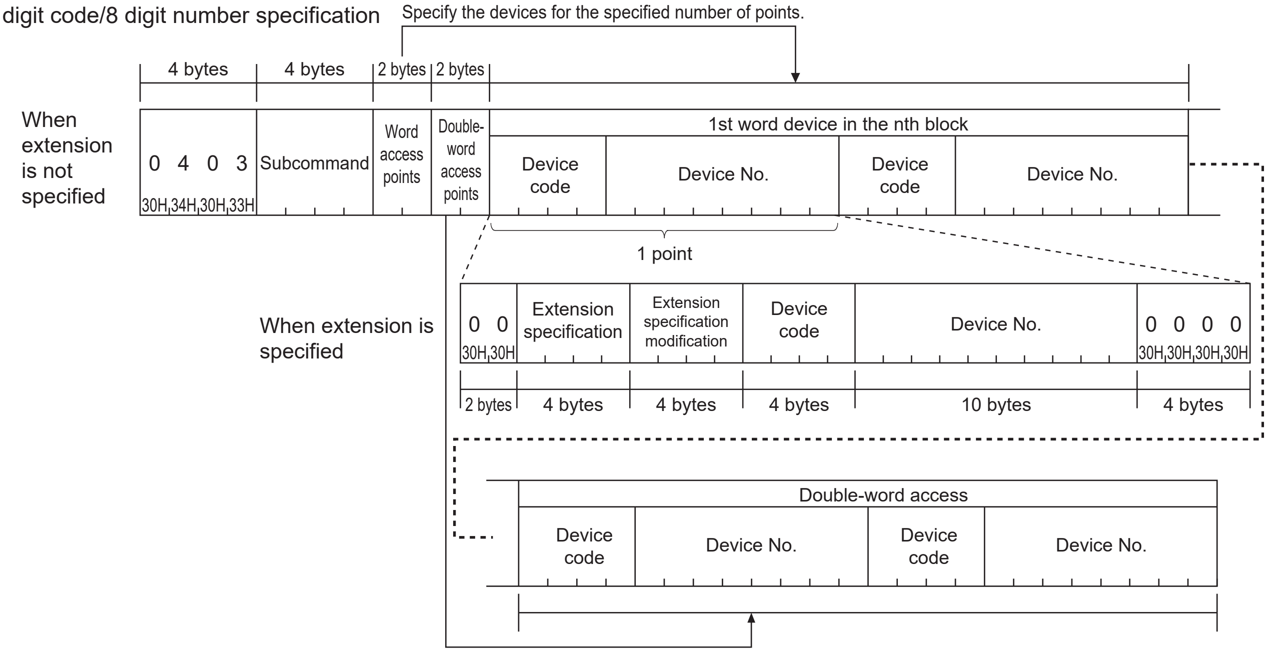
■When communicating data in ASCII code

2 digit code/6 digit number specification



Specify the devices for the specified number of points.

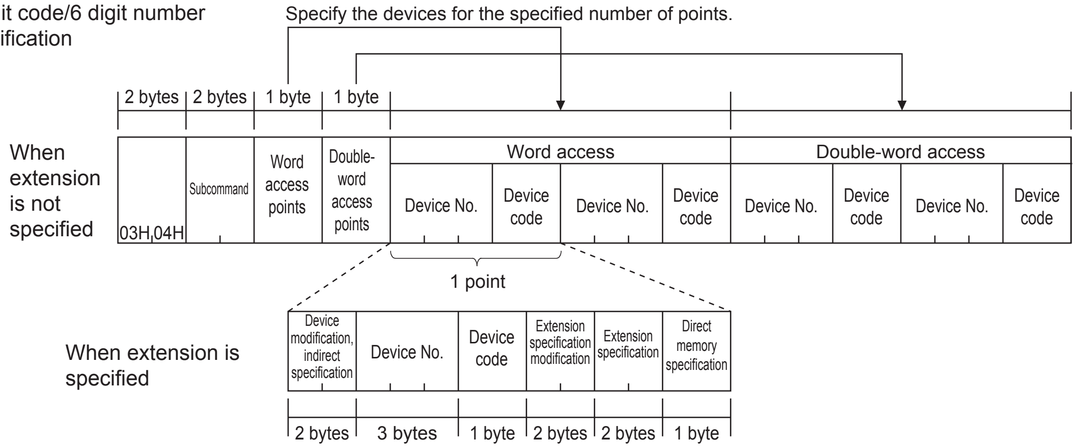
4 digit code/8 digit number specification



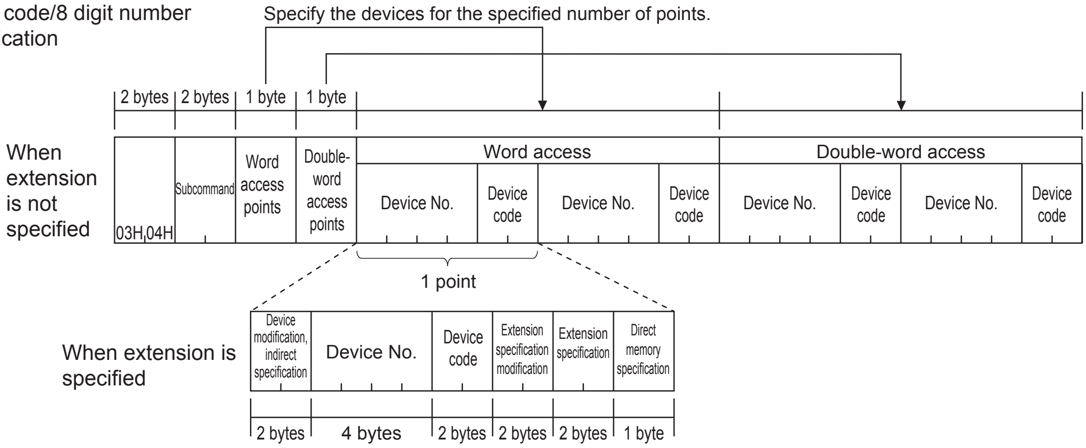
Specify the devices for the specified number of points.

■When communicating data in binary code

2 digit code/6 digit number specification



4 digit code/8 digit number specification



■Subcommand

Specify the subcommand selected from the item.

Item			Subcommand					
Data size specification	**Device specification format**	**Device memory extension specification**	**ASCII code (Upper column: characters, lower column: character code)**				**Binary code**	
Word units	2 digit code/6 digit number specification	Not specified	0	0	0	0	00H	00H
			30H	30H	30H	30H		
		Specified	0	0	8	0	80H	00H
			30H	30H	38H	30H		
	4 digit code/8 digit number specification	Specified	0	0	8	2	82H	00H
			30H	30H	38H	32H		

■Word access points, double-word access points

Specify the number of target device points of reading.

Item	Description	Number of points	
		ASCII code	**Binary code**
Word access points	Specify the number of points to be accessed in one-word units. The bit device is 16-point units, the word device is one-word units.	1 ≤ (word access points + double-word access points) × 2 ≤ 192 When device memory extension specification is used, double the number of the access points.	1 ≤ word access points + double-word access points ≤ 192[*1] When device memory extension specification is used, double the number of the access points.
Double-word access points	Specify the number of points to be accessed in two-word units. The bit device is 32-point units, the word device is two-word units.		

*1 123 points when the subcommand 0000H is specified in the Ethernet module.

■Device code, device No.

Specify the target device of reading.

Item	Description
Word access	Specify the device points specified as word access points. The specification is not necessary when the word access points are zero.
Double-word access	Specify the device points specified as double-word access points. The specification is not necessary when the double-word access points are zero.

Set up in order of word access device → double word access device.

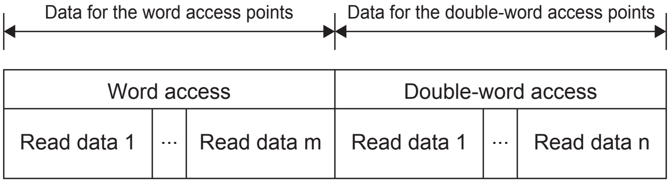
Precautions

Random Read (0403H) is not applicable to the following devices (contacts and coils).

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

Response data

The read device value is stored in hexadecimal. The data order differs depending on the type of code, ASCII code or binary code.



Word access			Double-word access		
Read data 1	···	Read data m	Read data 1	···	Read data n

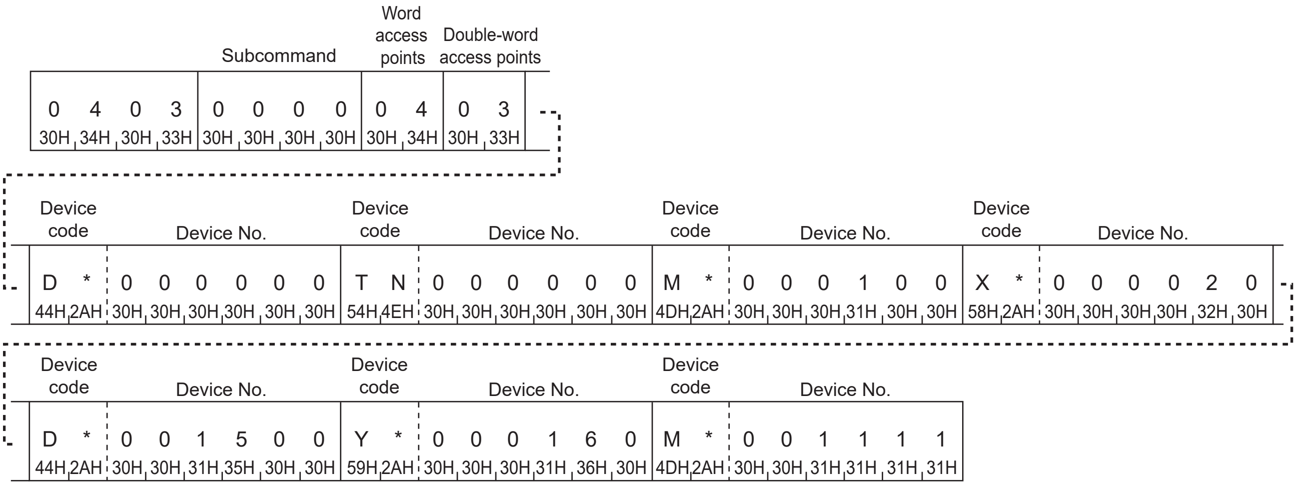
Communication example

Read D0, T0, M100 to M115, X20 to X37 by word access, and D1500 to D1501, Y160 to Y217, M1111 to M1142 by double-word access.

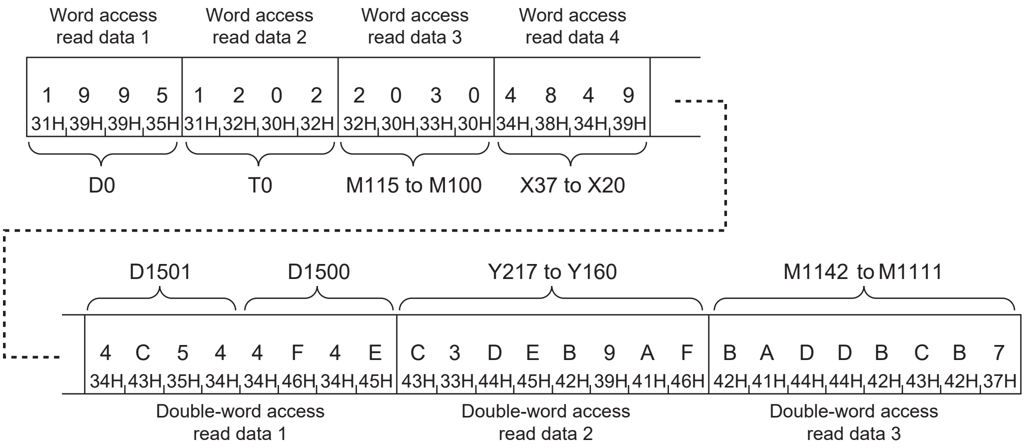
It is supposed that 6549 (1995H) is stored in D0, 4610 (1202H) is stored in T0, 20302 (4F4EH) is stored in D1500, 19540 (4C54H) is stored in D1501.

■When communicating data in ASCII code (X, Y OCT)

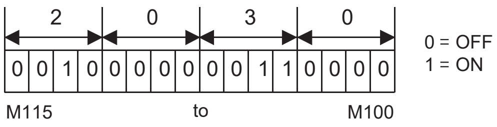
- Request data



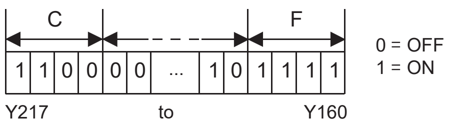
- Response data



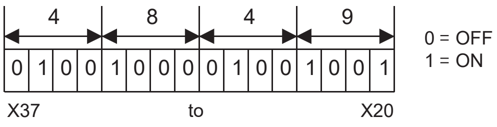
Word access read data 3



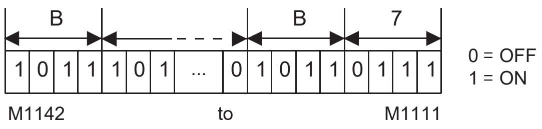
Double-word access read data 2



Word access read data 4

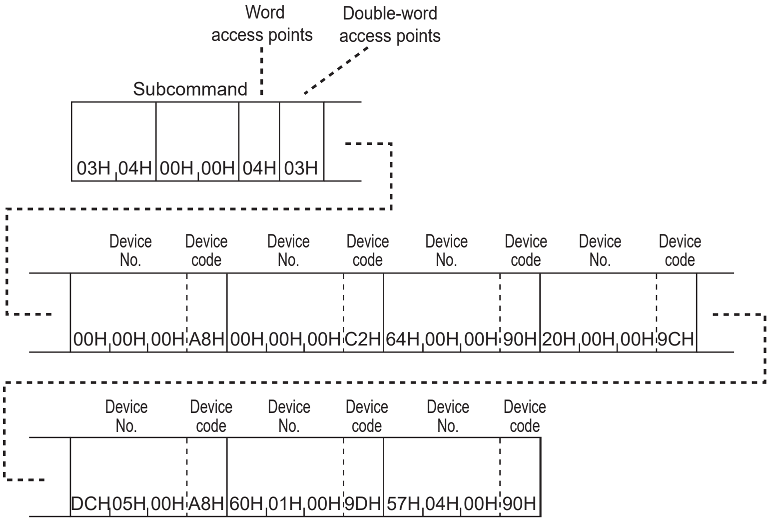


Double-word access read data 3

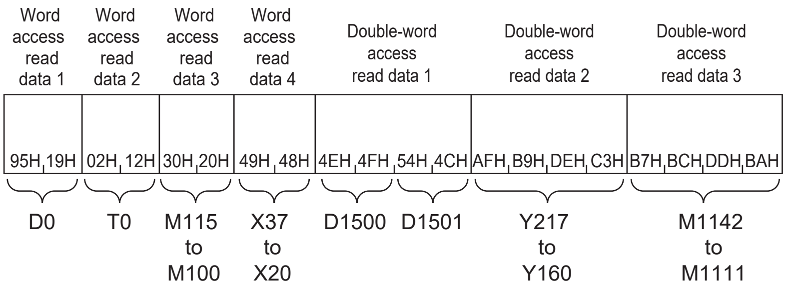


■When communicating data in binary code

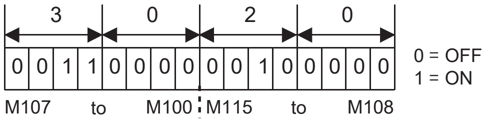
- Request data



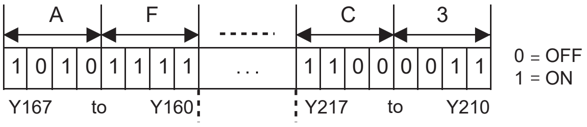
- Response data



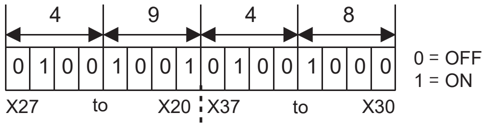
Word access read data 3



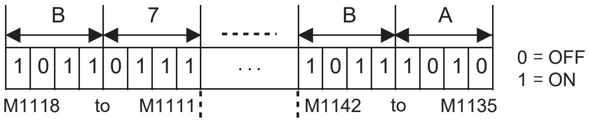
Double-word access read data 2



Word access read data 4



Double-word access read data 3



Device Write Random

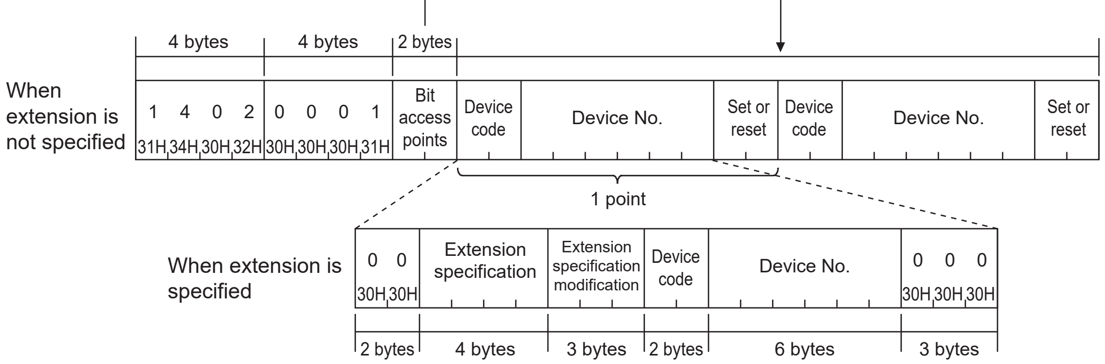
This command specifies the device No. randomly and writes the data.

Request data

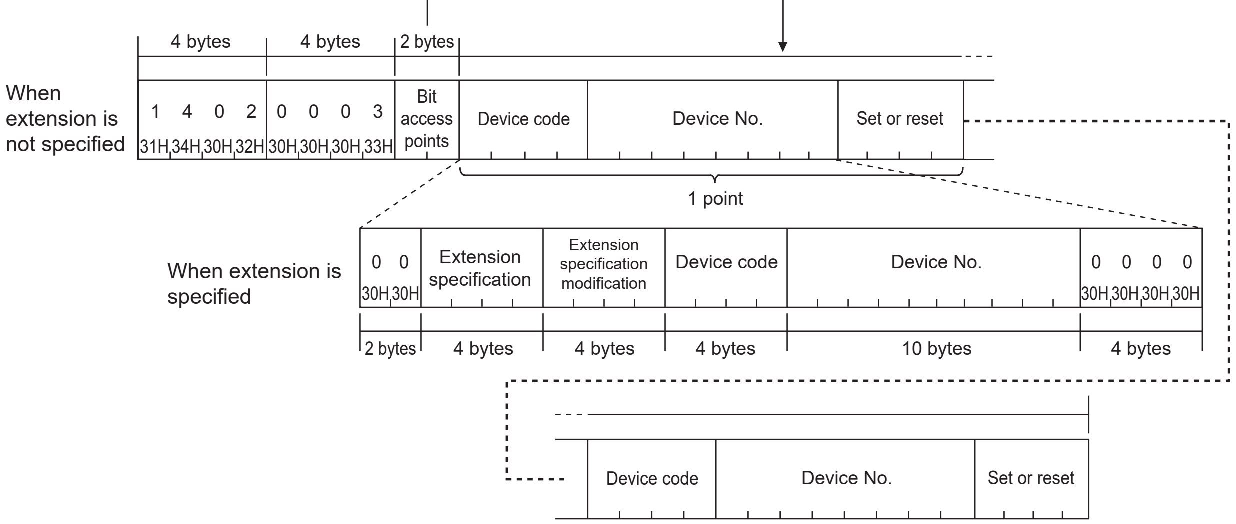
■When writing data in bit units

- When communicating data in ASCII code

2 digit code/6 digit number specification

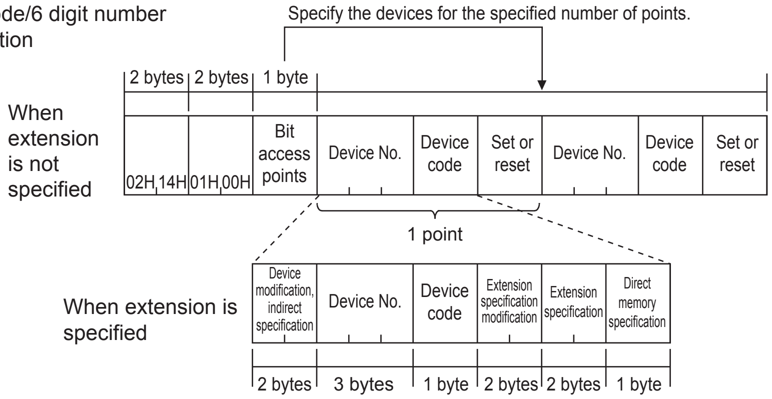


4 digit code/8 digit number specification

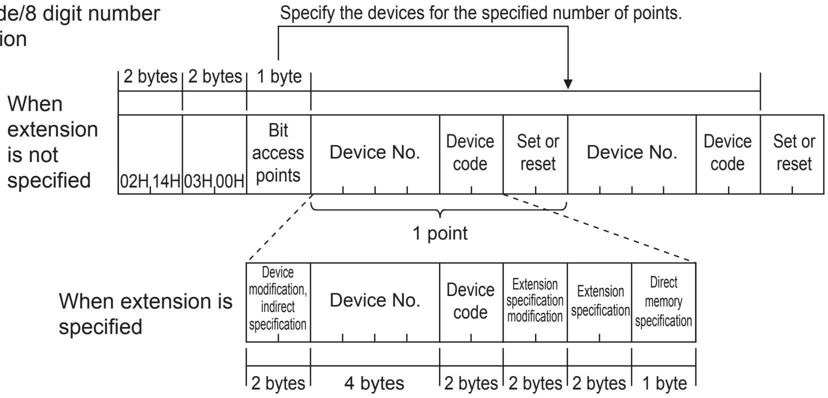


- When communicating data in binary code

2 digit code/6 digit number specification



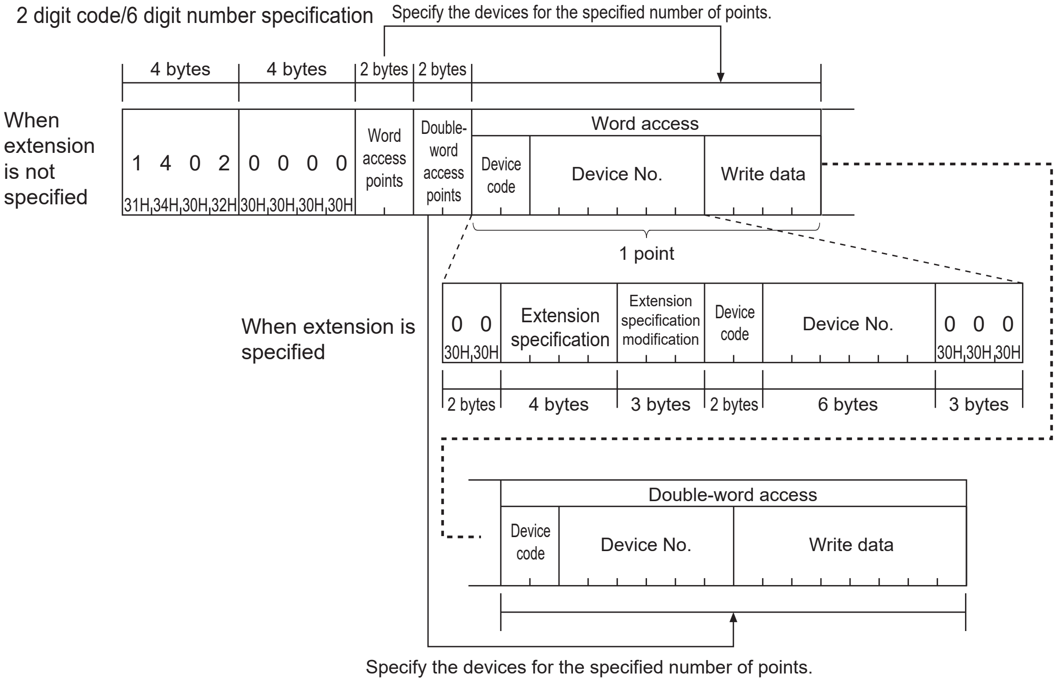
4 digit code/8 digit number specification



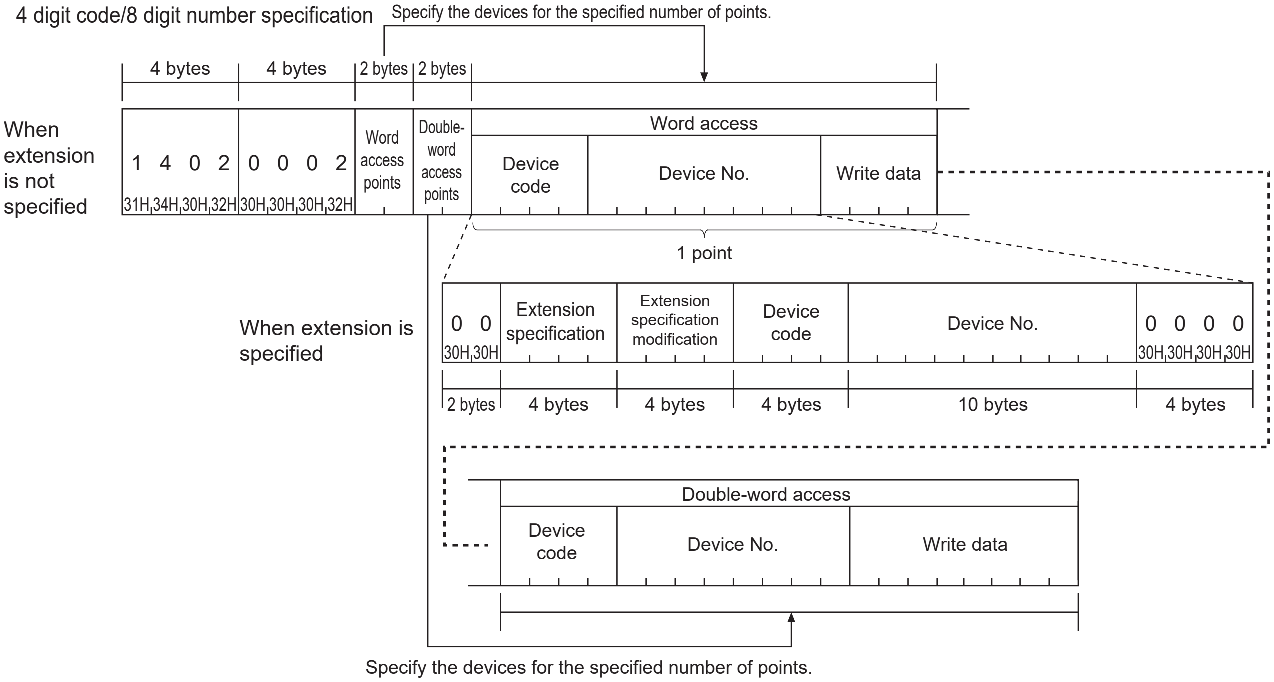
■When writing data in word units

- When communicating data in ASCII code

2 digit code/6 digit number specification

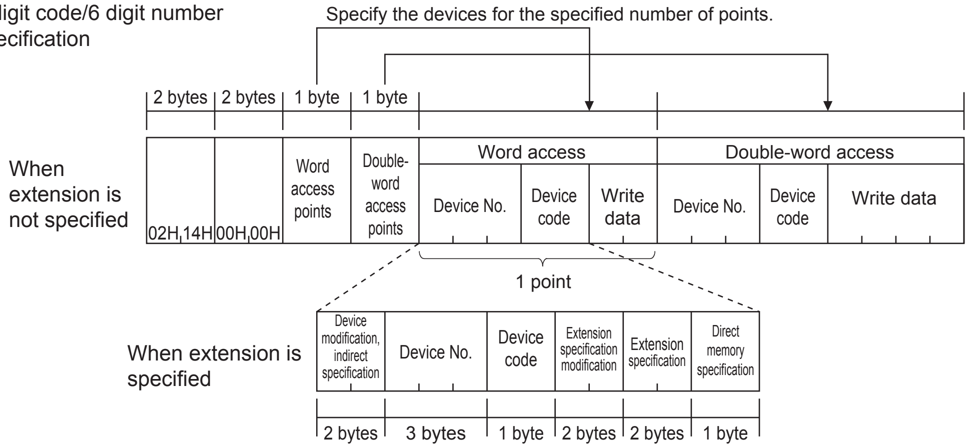


4 digit code/8 digit number specification

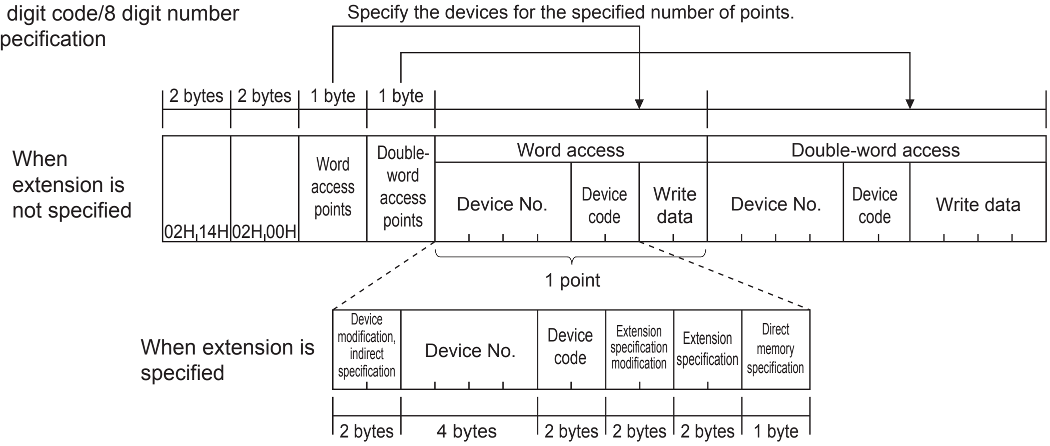


- When communicating data in binary code

2 digit code/6 digit number specification



4 digit code/8 digit number specification



■Subcommand

Specify the subcommand selected from the item.

Item			Subcommand					
Data size specification	**Device specification format**	**Device memory extension specification**	**ASCII code (Upper column: characters, lower column: character code)**				**Binary code**	
Bit units	2 digit code/6 digit number specification	Not specified	0	0	0	1	01H	00H
			30H	30H	30H	31H		
		Specified	0	0	8	1	81H	00H
			30H	30H	38H	31H		
	4 digit code/8 digit number specification	Specified	0	0	8	3	83H	00H
			30H	30H	38H	33H		
Word units	2 digit code/6 digit number specification	Not specified	0	0	0	0	00H	00H
			30H	30H	30H	30H		
		Specified	0	0	8	0	80H	00H
			30H	30H	38H	30H		
	4 digit code/8 digit number specification	Specified	0	0	8	2	82H	00H
			30H	30H	38H	32H		

■Bit access points, word access points, double-word access points

Item	Description	Number of points	
		ASCII code	Binary code
Bit access points	Specify the number of bit device points in one-point units.	1 to 94 When device memory extension specification is used 1 to 47	1 to 188 When device memory extension specification is used 1 to 94
Word access points	Specify the number of points to be accessed in one-word units. The bit device is 16-point units, the word device is one-word units.	1 ≤ (word access points × 12 + double-word access points × 14) × 2 ≤ 1920 When device memory extension specification is used, double the number of the access points.	1 ≤ word access points × 12 + double-word access points × 14 ≤ 1920*1 When device memory extension specification is used, double the number of the access points.
Double-word access points	Specify the number of points to be accessed in two-word units. The bit device is 32-point units, the word device is two-word units.		

*1 When the subcommand 0000H is specified in the Ethernet module, the number of points is:
1 ≤ (number of word access points) × 18 + (number of double-word access points) × 20 ≤ 1962 points

■Device code, device No., write data

Specify the target device of writing.

The data is specified in hexadecimal number.

Item	Description
Word access	Specify the device points specified as word access points. The specification is not necessary when the word access points are zero.
Double-word access	Specify the device points specified as double-word access points. The specification is not necessary when the double-word access points are zero.

Precautions

Random Write (1402H) is not applicable to the following devices (contacts and coils).

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

■Set or reset

Specify ON/OFF of the bit device.

- 2 digit code/6 digit number specification

Item	Data to write		Remark
	ON	OFF	
ASCII code	"01"	"00"	Two characters will be sent in order from "0".
Binary code	01H	00H	The one-byte numerical value shown left will be sent.

- 4 digit code/8 digit number specification

Item	Data to write		Remark
	ON	OFF	
ASCII code	"0001"	"0000"	Four characters will be sent in order from "0".
Binary code	0001H	0000H	The two-byte numerical value shown left will be sent.

Response data

There is no response data for the Write Random command.

Communication example

■When writing data in bit units

Turn off M50 and turn on Y27.

- When communicating data in ASCII code (X, Y OCT)

(Request data)

	Subcommand	Bit access points	Device code	Device No.	Set or reset	Device code	Device No.	Set or reset
1 4 0 2	0 0 0 1	0 2	M *	0 0 0 0 5 0	0 0	Y *	0 0 0 0 2 7	0 1
31H 34H 30H 32H	30H 30H 30H 31H	30H 32H	4DH 2AH	30H 30H 30H 30H 35H 30H	30H 30H	59H 2AH	30H 30H 30H 30H 32H 37H	30H 31H

- When communicating data in binary code

(Request data)

	Subcommand	Bit access points	Device No.	Device code	Set or reset	Device No.	Device code	Set or reset
02H 14H	01H 00H	02H	32H 00H 00H	90H	00H	27H 00H 00H	9DH	01H

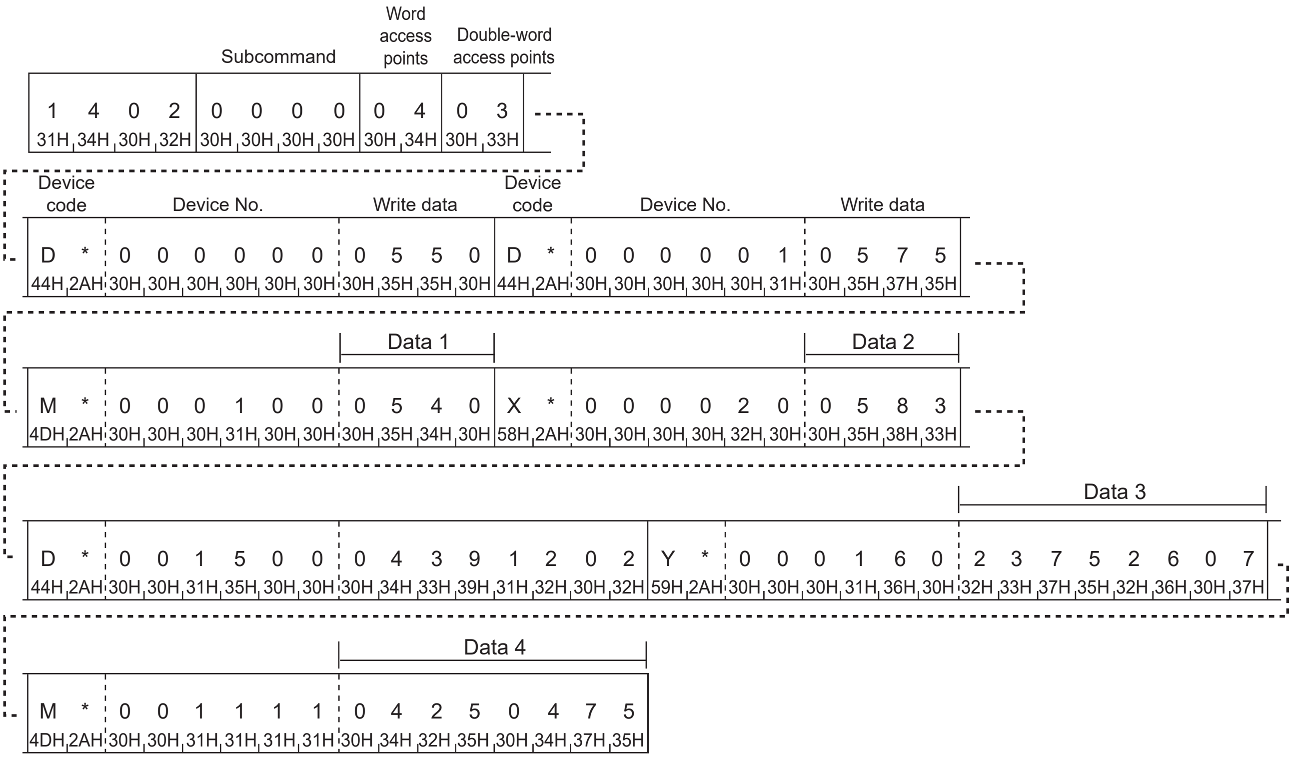
■When writing data in word units

Write the value in a device as follows.

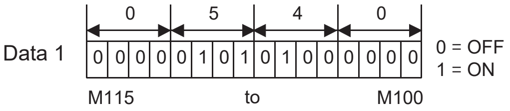
Item	Target device
Word access	D0, D1, M100 to M115, X20 to X37
Double-word access	D1500 to D1501, Y160 to Y217, M1111 to M1142

- When communicating data in ASCII code (X, Y OCT)

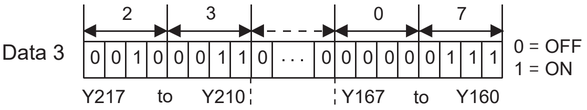
(Request data)



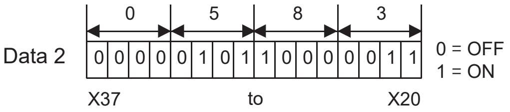
Data 1



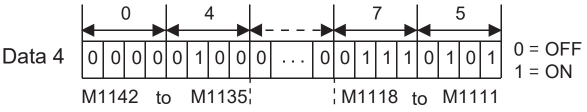
Data 3



Data 2

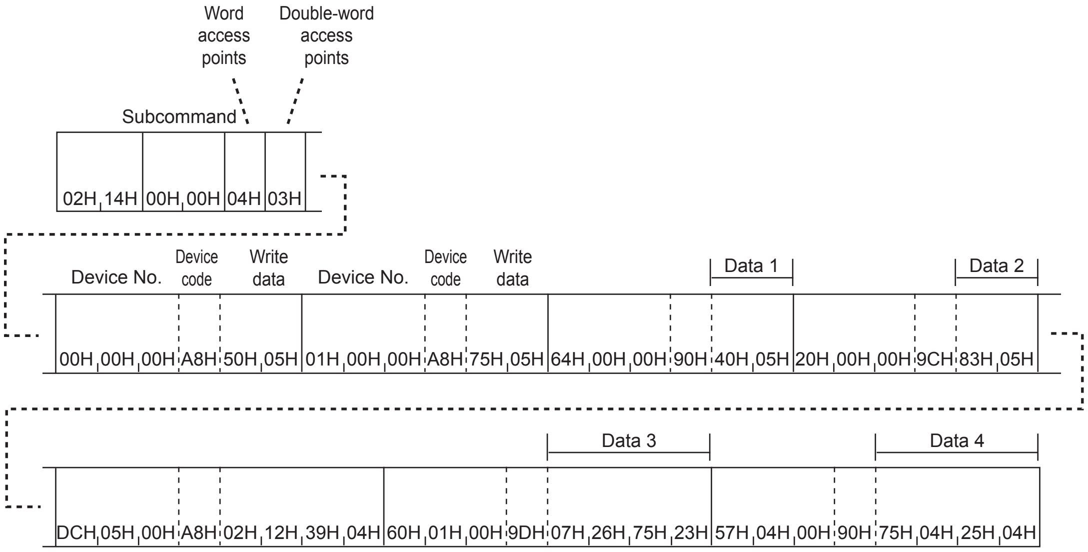


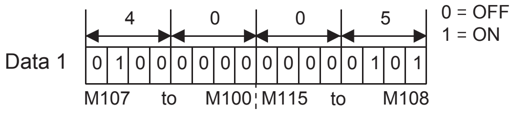
Data 4

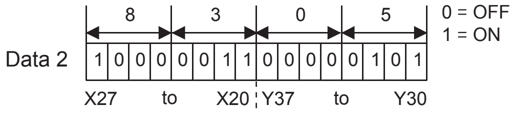


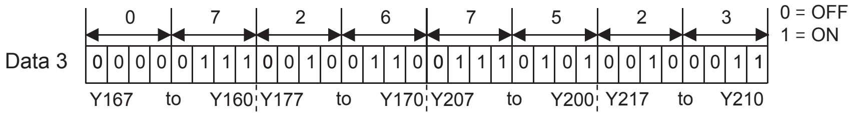
- When communicating data in binary code

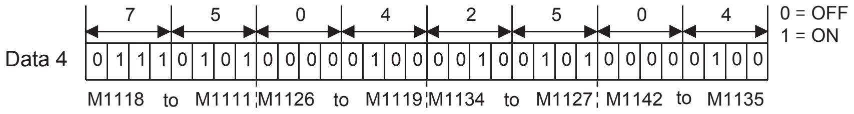
(Request data)











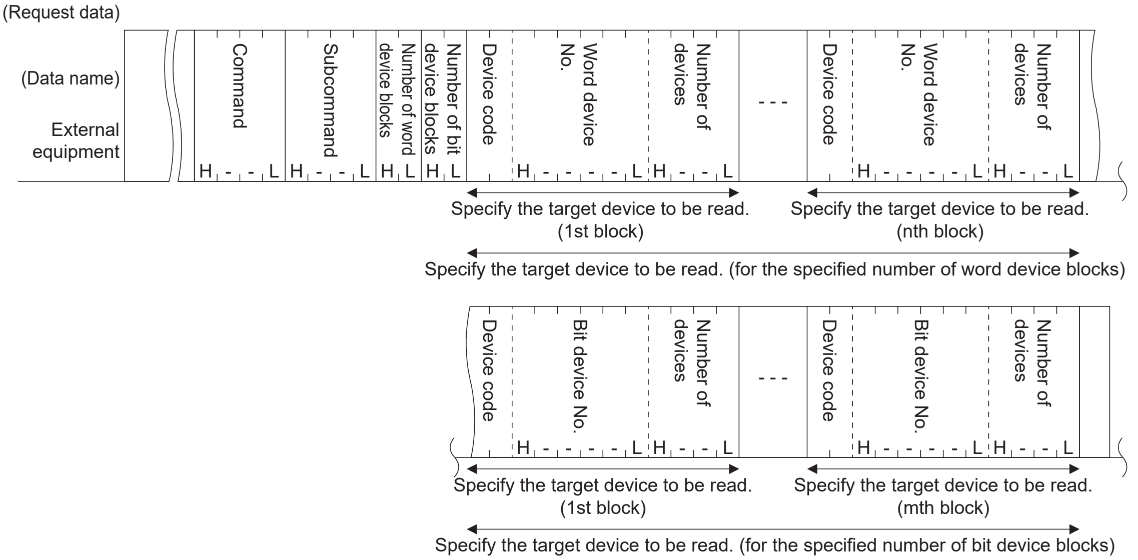
Device Read Block

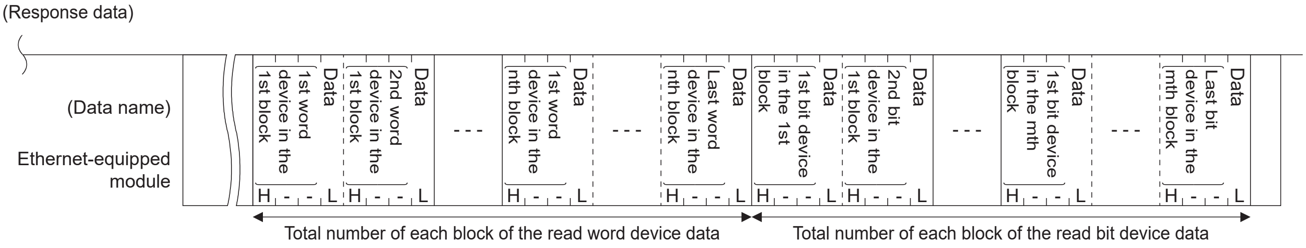
The examples shown in this section explain the control procedure for reading by randomly specifying multiple blocks, where 1 block consists of n point(s) of a bit device memory (one point is specified by 16-bit) and a word device memory (one point is specified by 1-word).

Data array in the character area during the device read block

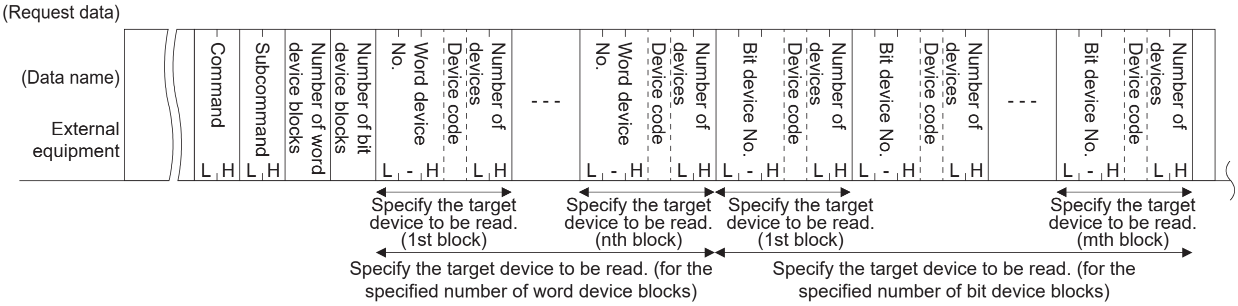
This section explains how data is ordered in the character areas during device read block.

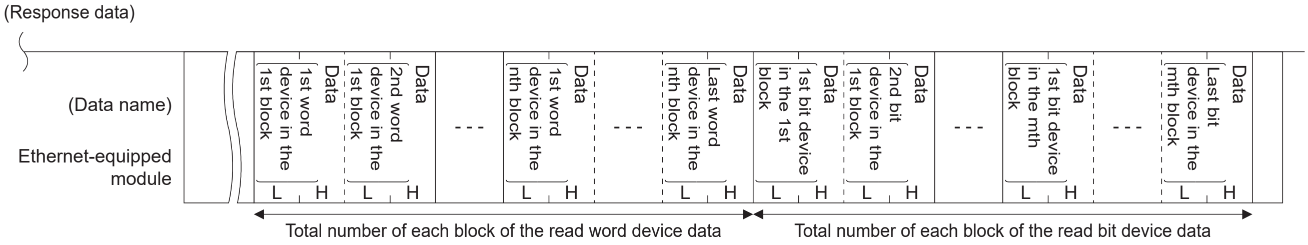
■When communicating data in ASCII code





■When communicating data in binary code





Contents of the character areas during device read block

This section explains what is in the character area when a device read block function is performed.

■Number of word device blocks and number of bit device blocks

This data is for specifying the number of word device blocks or bit device blocks to be sent directly after this data field in the batch read to the word device or bit device, respectively.

- When communicating data in ASCII code

Each number of blocks are converted to 2-digit ASCII code (hexadecimal) and sent.

Ex.

For 5 blocks: Converted to "05", and sent sequentially from "0".

For 20 blocks: Converted to "14", and sent sequentially from "1".

- When communicating data in binary code

1-byte numeric value indicating the number of blocks is transmitted.

Ex.

For 5 blocks: 05H is sent.

For 20 blocks: 14H is sent.

- Specify the number of blocks so the following condition is satisfied:

120 ≥ number of word device blocks + number of bit device blocks

- When setting either number of blocks to 0, the corresponding device number, device code, number of device points, and data specification are not necessary.

■Word device number and bit device number

This data is for specifying the head word device or bit device for each block to which batch read is performed, where continuous word or bit devices are considered one block.

- When communicating data in ASCII code

The head device number of each block is converted to 6-digit ASCII code and sent.

Ex.

Internal relay M1234 and link register W1234:

The internal relay M1234 is converted to "001234" or " 1234", and the link register W1234 is converted to "001234" or "__1234". In both cases, the transmission starts from "0" or " " (space).

- When communicating data in binary code

The head device number of each block is indicated in a 3-byte numeric value and sent.

Ex.

Internal relay M1234 and link register W1234:

Internal relay M1234 becomes 0004D2H and is sent in the order of D2H, 04H, and 00H.

The link register W1234 is converted to 001234H and sent in the order of 34H, 12H, and 00H.

■Device code

This data is for identifying the head device memory for each block for which batch read is performed.

The device code for each device is shown in ☞ Page 611 Device range.

Precautions

Double word devices and long index registers (LZ) cannot be used in the multiple block batch read (0406H).

- When communicating data in ASCII code

Each device code is converted to 2-digit ASCII code (hexadecimal) and sent.

Ex.

Internal relay (M) and link register (W):

The internal relay (M) is converted to "M*" and link register (W) is converted to "W*", and sent from "M" and "W" respectively.

- When communicating data in binary code

1-byte numeric value indicating each device code is sent.

Ex.

Internal relay (M) and link register (W):

90H is transmitted for the internal relay (M) and B4H is sent for the link register (W).

■Number of devices

This data is for specifying the number of points in the continuous device range of each block for which batch read is performed (1 point = 16 bits for bit device memory and 1 point = 1 word for word device memory), where one block consists of continuous word or bit devices.

- When communicating data in ASCII code

The number of points for each block is converted to a 4-digit ASCII code (hexadecimal) and sent.

Ex.

For 5 points: Converted to "0005", and sent sequentially from "0".

For 20 points: Converted to "0014", and sent sequentially from "0".

- When communicating data in binary code

2-byte numeric value indicating the number of points for each block is sent.

Ex.

For 5 points: Converted to 0005H, and sent sequentially from 05H.

For 20 points: Converted to 0014H, and sent sequentially from 14H.

- Specify number of devices so that the appropriate condition is satisfied

960 ≥ total number of points for all word device blocks + total number of points for all bit device blocks

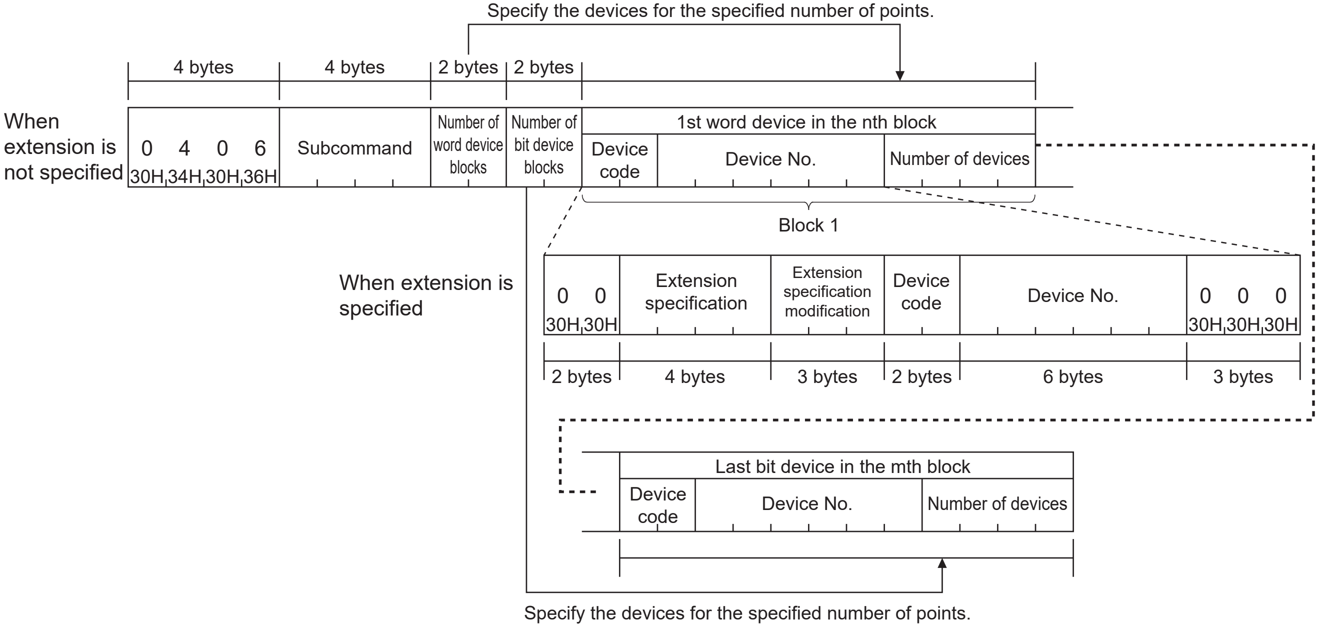
Point

The extension specification is allowed for the device memory being read using the device read block functions.

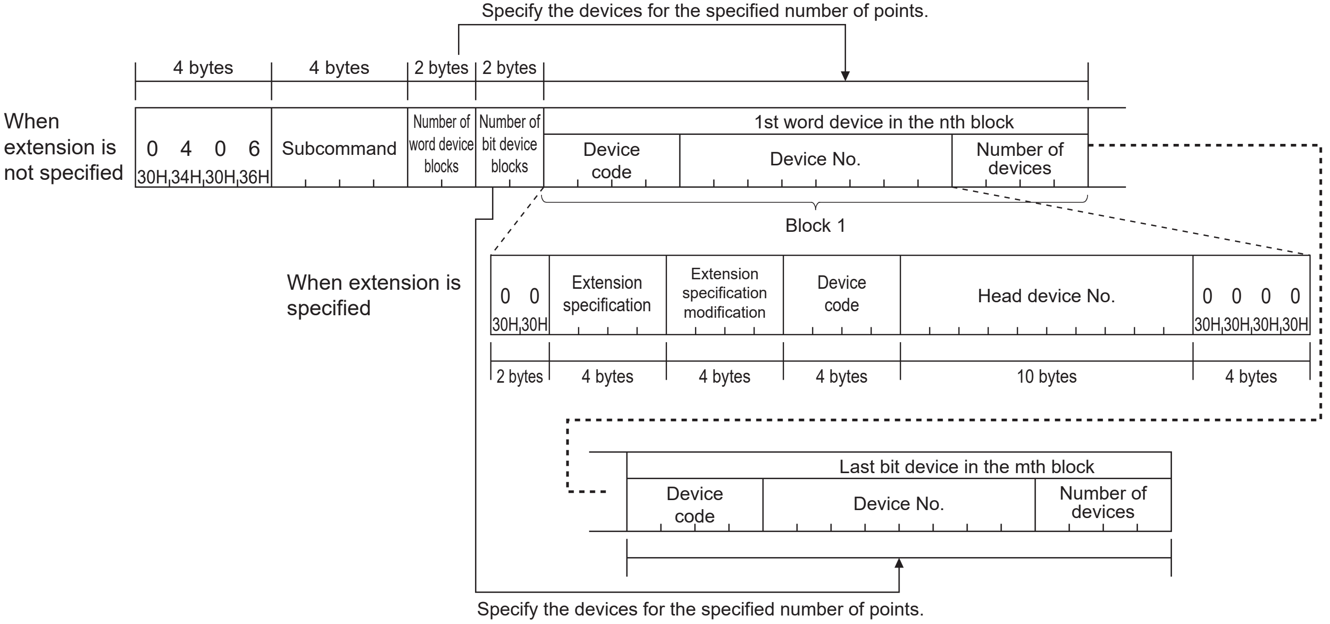
Request data

■When communicating data in ASCII code

2 digit code/6 digit number specification

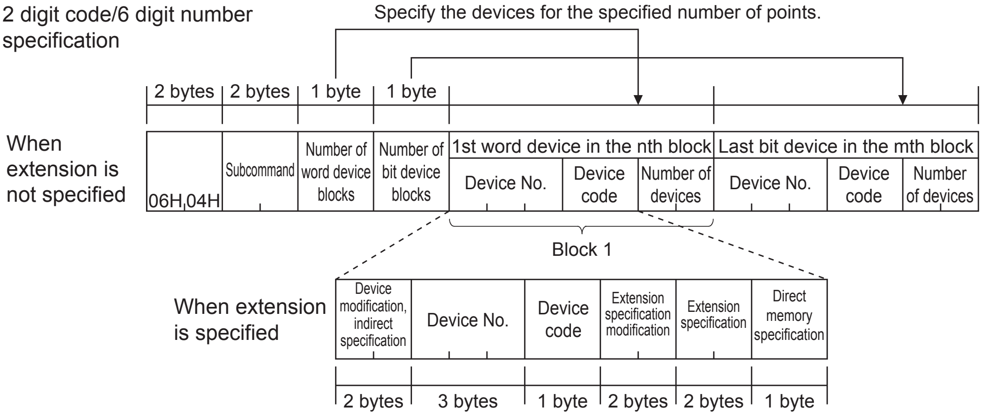


4 digit code/8 digit number specification

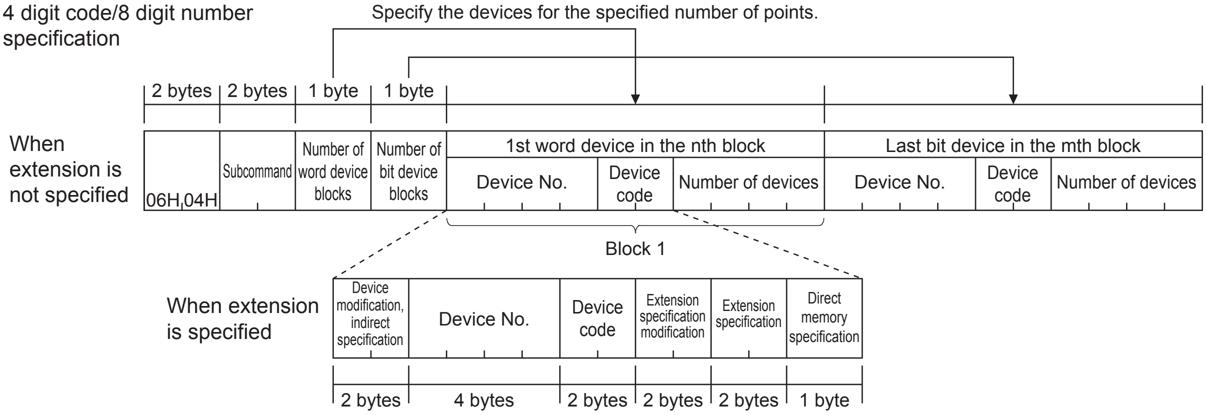


■When communicating data in binary code

2 digit code/6 digit number specification



4 digit code/8 digit number specification



❶Subcommand

Specify the subcommand selected from the item.

Item			Subcommand					
Data size specification	**Device specification format**	**Device memory extension specification**	**ASCII code (Upper column: characters, lower column: character code)**				**Binary code**	
Word units	2 digit code/6 digit number specification	Not specified	0	0	0	0	00H	00H
			30H	30H	30H	30H		
		Specified	0	0	8	0	80H	00H
			30H	30H	38H	30H		
	4 digit code/8 digit number specification	Specified	0	0	8	2	82H	00H
			30H	30H	38H	32H		

❷Number of word device blocks and number of bit device blocks

Specify the number of blocks of the device to be read in hexadecimal.

Item	Description	Number of points	
		ASCII code	**Binary code**
Number of word device blocks	Specify the number of blocks of the word device to be read.	(Number of word device blocks + number of bit device blocks) × 2 ≤ 120 and (Total points of each block of word device + total points of each block of bit device) × 2 ≤ 960 When device memory extension specification is used (Number of word device blocks + number of bit device blocks) × 4 ≤ 120 and (Total points of each block of word device + total points of each block of bit device) × 2 ≤ 960	Number of word device blocks + number of bit device blocks ≤ 120 and Total points of each block of word device + total points of each block of bit device ≤ 960 When device memory extension specification is used (Number of word device blocks + number of bit device blocks) × 2 ≤ 120 and Total points of each block of word device + total points of each block of bit device ≤ 960
Number of bit device blocks	Specify the number of blocks of the bit device to be read.		

❸Device code, device No., number of device points

Specify the device points while satisfying the following conditions:

Total number of points for all word device blocks + total number of points for all bit device blocks ≤ 960

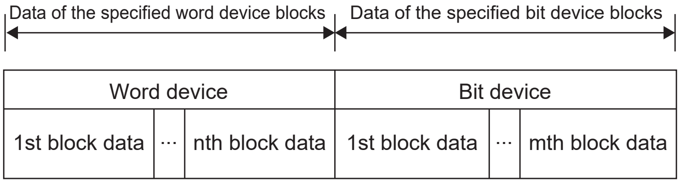
Item	Description
Word device	Specify the device points specified in "Number of word device blocks". When "Number of word device blocks" is set to 0, this specification is unnecessary.
Bit device	Specify the device points specified in "Number of bit device blocks". When "Number of bit device blocks" is set to 0, this specification is unnecessary.

Point

When specifying a contact and a coil of a timer, retentive timer, and counter, use the bit device block.

Set up in order of word device → bit device.

Response data



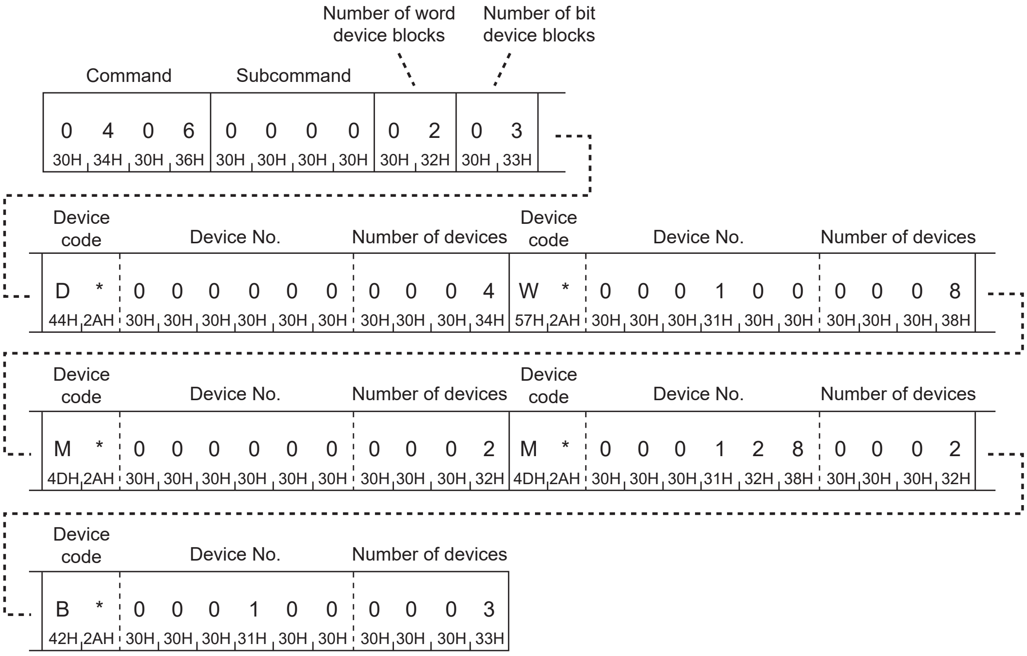
Communication example

Values are read from devices as follows.

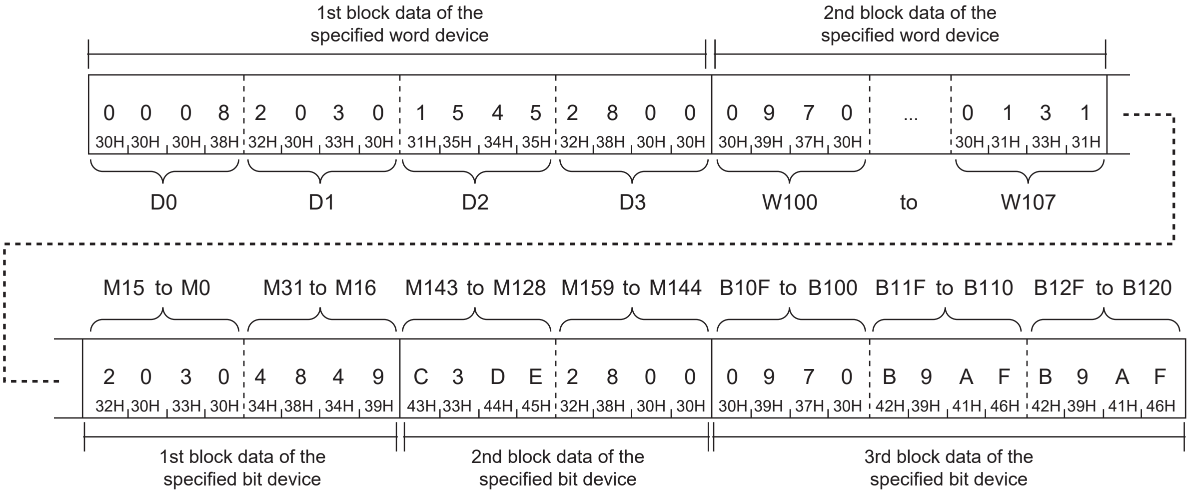
Item	Read contents
Word device	• Block 1: D0 to D3 (4 points) • Block 2: W100 to W107 (8 points)
Bit device	• Block 1: M0 to M31 (2 points) • Block 2: M128 to M159 (2 points) • Block 3: B100 to B12F (3 points)

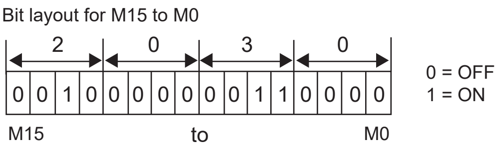
■When communicating data in ASCII code

(Request data)



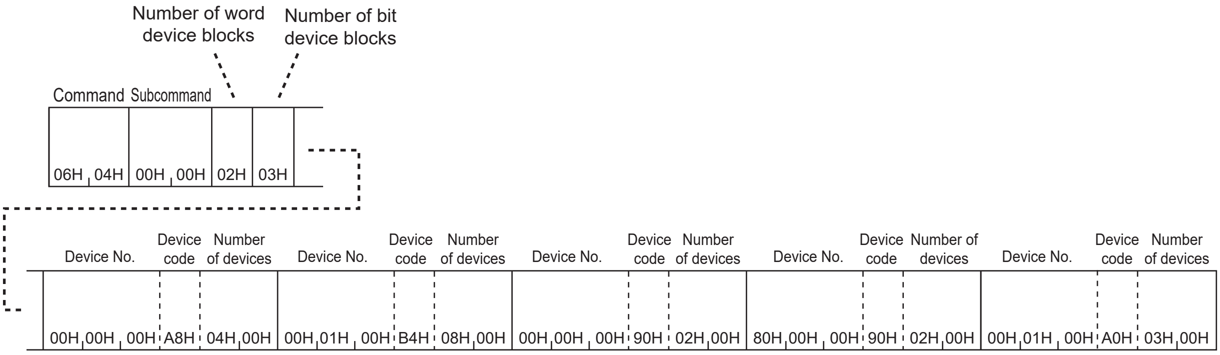
(Response data)



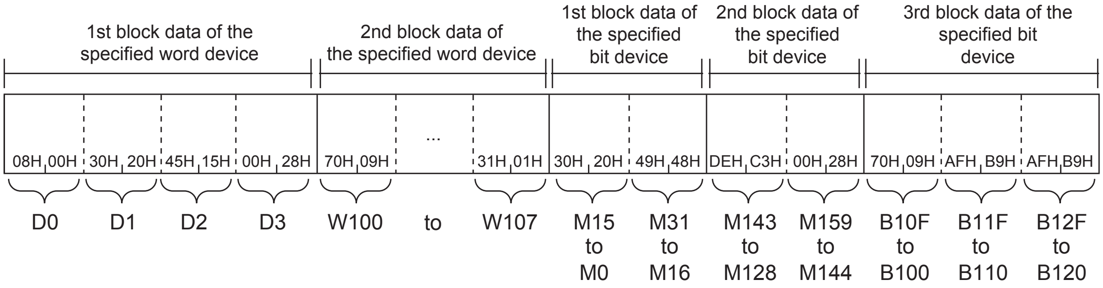


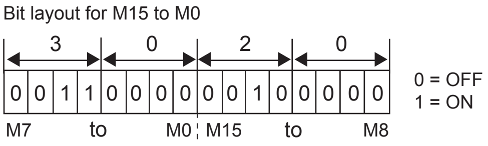
■When communicating data in binary code

(Request data)



(Response data)





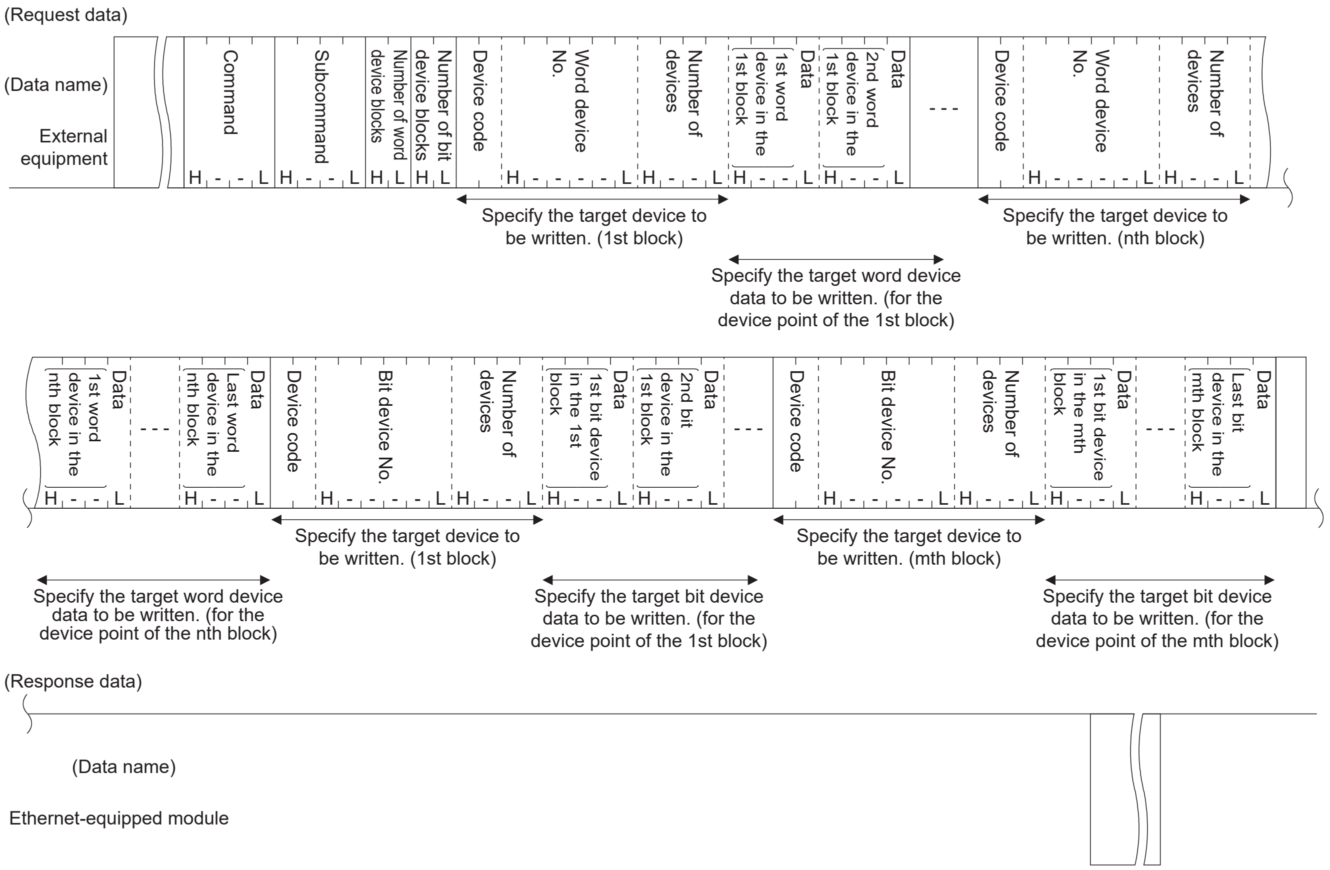
Device Write Block

The examples shown in this section explain the control procedure for writing by randomly specifying multiple blocks, where 1 block consists of n point(s) of a bit device memory (one point is specified by 16-bit) and a word device memory (one point is specified by 1-word).

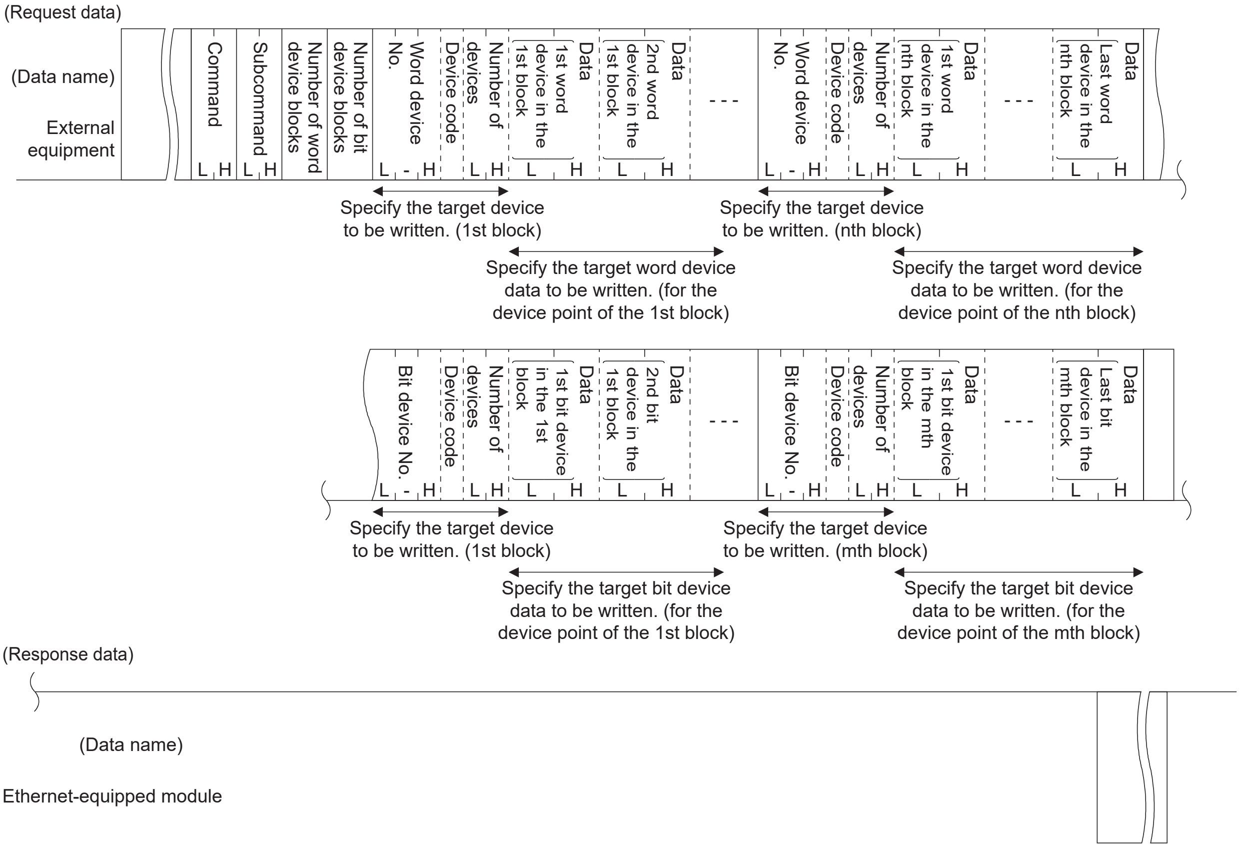
Data array in the character area during the device write block

This section explains how data is ordered in the character areas during device write block.

■When communicating data in ASCII code



■When communicating data in binary code



Contents of the character areas during device write block

This section explains what is in the character area when a device write block function is performed.

■Number of word device blocks and number of bit device blocks

This data is for specifying the number of word device blocks or bit device blocks to be sent directly after this data field in the batch write to the word device or bit device, respectively.

- When communicating data in ASCII code

Each number of blocks are converted to 2-digit ASCII code (hexadecimal) and sent.

Ex.

For 5 blocks: Converted to "05", and sent sequentially from "0".

For 20 blocks: Converted to "14", and sent sequentially from "1".

- When communicating data in binary code

1-byte numeric value indicating the number of blocks is transmitted.

Ex.

For 5 blocks: 05H is sent.

For 20 blocks: 14H is sent.

- Specify the number of blocks so the following condition is satisfied:

120 ≥ number of word device blocks + number of bit device blocks

- When setting either number of blocks to 0, the corresponding device number, device code, number of device points, and data specification are not necessary.

■Word device number and bit device number

This data is for specifying the head word device or bit device for each block to which batch write is performed, where continuous word or bit devices are considered one block.

- When communicating data in ASCII code

The head device number of each block is converted to 6-digit ASCII code and sent.

Ex.

Internal relay M1234 and link register W1234:

The internal relay M1234 is converted to "001234" or " 1234", and the link register W1234 is converted to "001234" or " __1234". In both cases, the transmission starts from "0" or " " (space).

- When communicating data in binary code

The head device number of each block is indicated in a 3-byte numeric value and sent.

Ex.

Internal relay M1234 and link register W1234:

Internal relay M1234 becomes 0004D2H and is sent in the order of D2H, 04H, and 00H.

The link register W1234 is converted to 001234H and sent in the order of 34H, 12H, and 00H.

■Device code

This data is for identifying the head device memory for each block for which batch write is performed.

The device code for each device is shown in ☞ Page 611 Device range.

Precautions

Double word devices and long index registers (LZ) cannot be used in the multiple block batch write (1406H).

- When communicating data in ASCII code

Each device code is converted to 2-digit ASCII code (hexadecimal) and sent.

Ex.

Internal relay (M) and link register (W):

The internal relay (M) is converted to "M*" and link register (W) is converted to "W*", and sent from "M" and "W" respectively.

- When communicating data in binary code

1-byte numeric value indicating each device code is sent.

Ex.

Internal relay (M) and link register (W):

90H is transmitted for the internal relay (M) and B4H is sent for the link register (W).

■Number of devices

This data is for specifying the number of points in the continuous device range of each block for which batch write is performed (1 point = 16 bits for bit device memory and 1 point = 1 word for word device memory), where one block consists of continuous word or bit devices.

- When communicating data in ASCII code

The number of points for each block is converted to a 4-digit ASCII code (hexadecimal) and sent.

Ex.

For 5 points: Converted to "0005", and sent sequentially from "0".

For 20 points: Converted to "0014", and sent sequentially from "0".

- When communicating data in binary code

2-byte numeric value indicating the number of points for each block is sent.

Ex.

For 5 points: Converted to 0005H, and sent sequentially from 05H.

For 20 points: Converted to 0014H, and sent sequentially from 14H.

- Specify number of devices so that the appropriate condition is satisfied

760 ≥ 4 × (number of word device blocks + number of bit device blocks) + total number of points for all word device blocks + total number of points for all bit device blocks

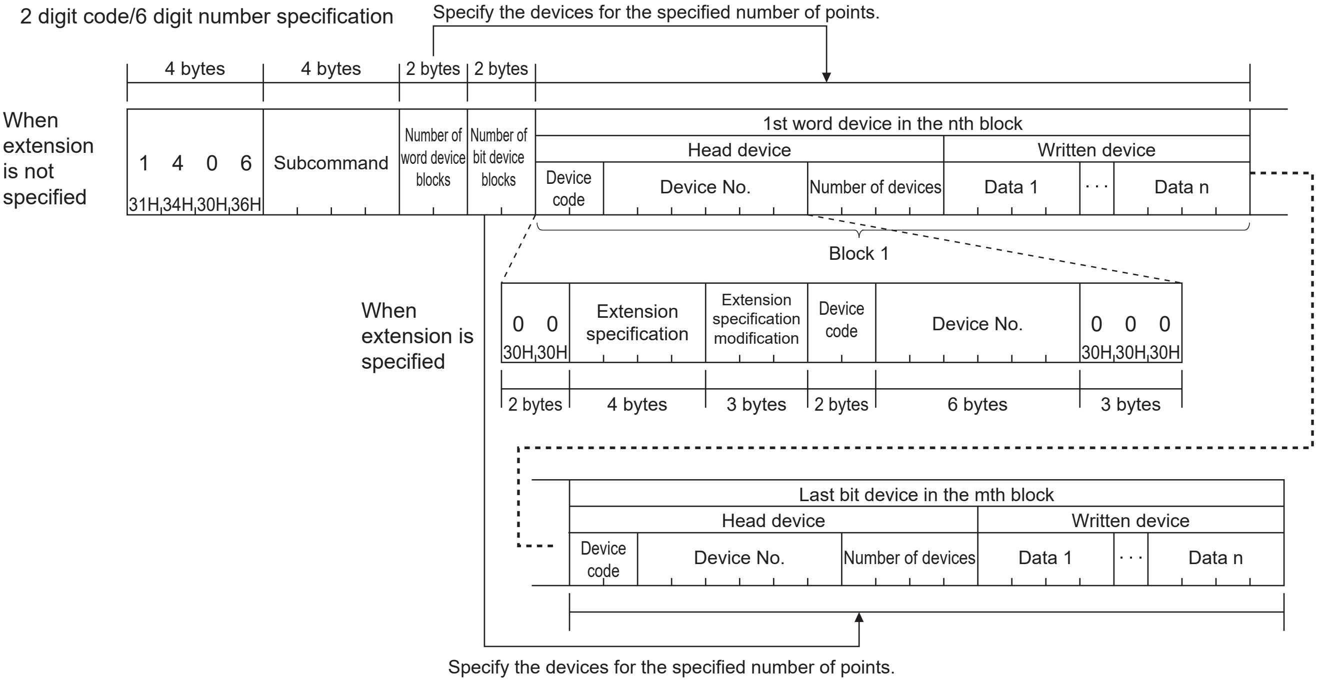
Point

The extension specification is allowed for the device memory being written to using the device write block functions.

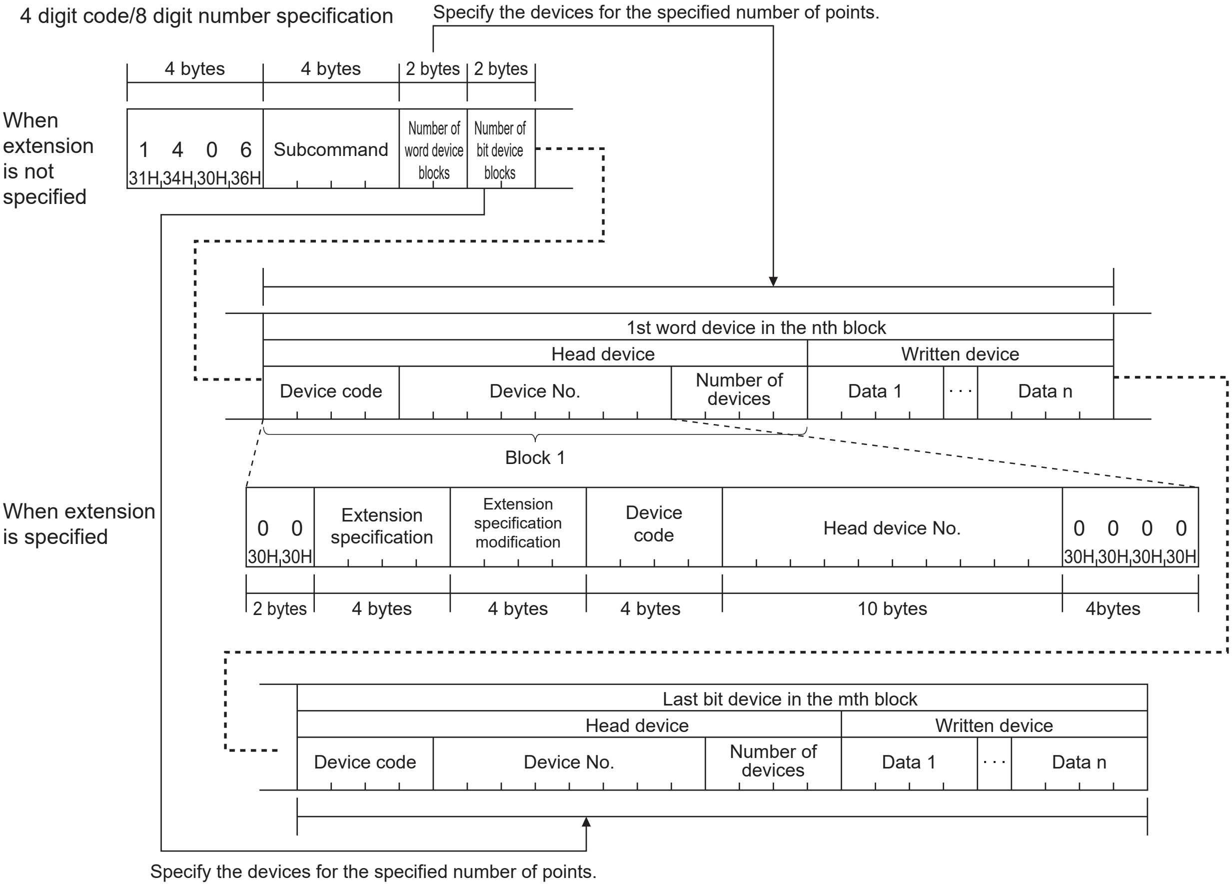
Request data

■When communicating data in ASCII code

2 digit code/6 digit number specification

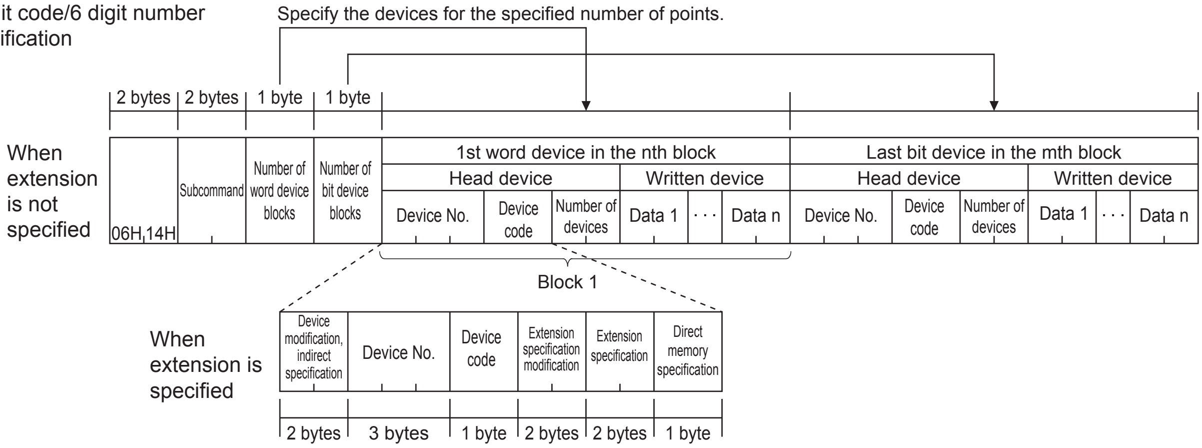


4 digit code/8 digit number specification

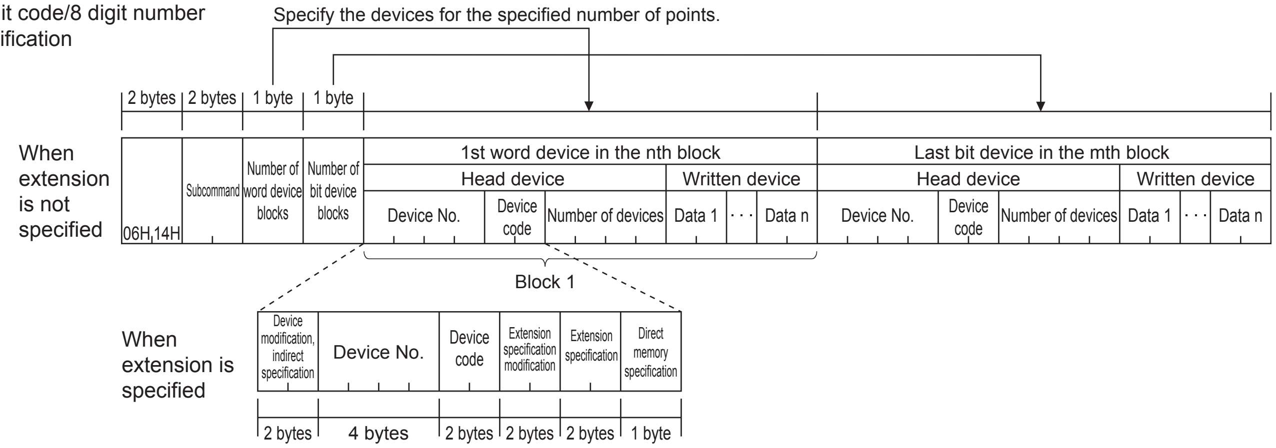


■When communicating data in binary code

2 digit code/6 digit number specification



4 digit code/8 digit number specification



❶Subcommand

Specify the subcommand selected from the item.

Item			**Subcommand**					
Data size specification	**Device specification format**	**Device memory extension specification**	**ASCII code (Upper column: characters, lower column: character code)**				**Binary code**	
Word units	2 digit code/6 digit number specification	Not specified	0	0	0	0	00H	00H
			30H	30H	30H	30H		
		Specified	0	0	8	0	80H	00H
			30H	30H	38H	30H		
	4 digit code/8 digit number specification	Specified	0	0	8	2	82H	00H
			30H	30H	38H	32H		

❷Number of word device blocks and number of bit device blocks

Specify the number of blocks of the device to be write in hexadecimal.

Item	Description	Number of points		
		ASCII code	Binary code	
			CPU module	Ethernet module, FX5-CCLGN-MS, FX5-CCLIEF, FX5-40SSC-G, FX5-80SSC-G
Number of word device blocks	Specify the number of blocks of the word device to be write.	(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 block of word device + total points of each block of bit device) × 2 ≤ 760 When device memory extension specification is used (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 block of word device + total points of each block of bit device) × 2 ≤ 760	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 block of word device + total points of each block of bit device ≤ 760 When device memory extension specification is used (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 block of word device + total points of each block of bit device ≤ 760	
Number of bit device blocks	Specify the number of blocks of the bit device to be write.			

❸Device code, device No., number of device points

Specify the device points while satisfying the following conditions:

(Number of word device blocks + number of bit device blocks) × 4 + Total number of points for all word device blocks + total number of points for all bit device blocks ≤ 760

Item	Description
Word device	Specify the device points specified in "Number of word device blocks". When "Number of word device blocks" is set to 0, this specification is unnecessary.
Bit device	Specify the device points specified in "Number of bit device blocks". When "Number of bit device blocks" is set to 0, this specification is unnecessary.

Point

When specifying a contact and a coil of a timer, retentive timer, and counter, use the bit device block.

Set up in order of word device → bit device.

Response data

There is no response data for the device write block command.

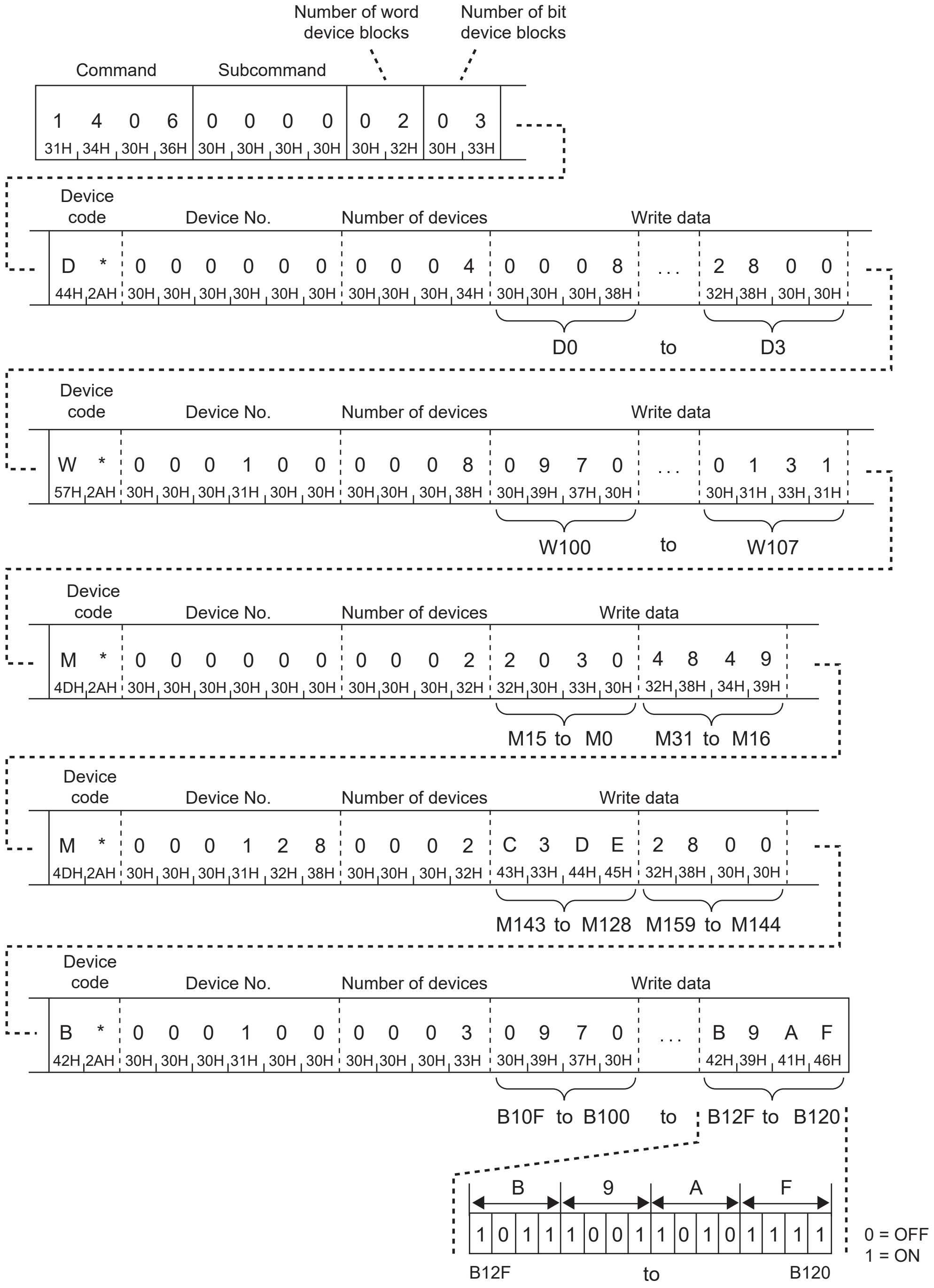
Communication example

Write values from devices as follows.

Item	Write contents
Word device	• Block 1: D0 to D3 (4 points) • Block 2: W100 to W107 (8 points)
Bit device	• Block 1: M0 to M31 (2 points) • Block 2: M128 to M159 (2 points) • Block 3: B100 to B12F (3 points)

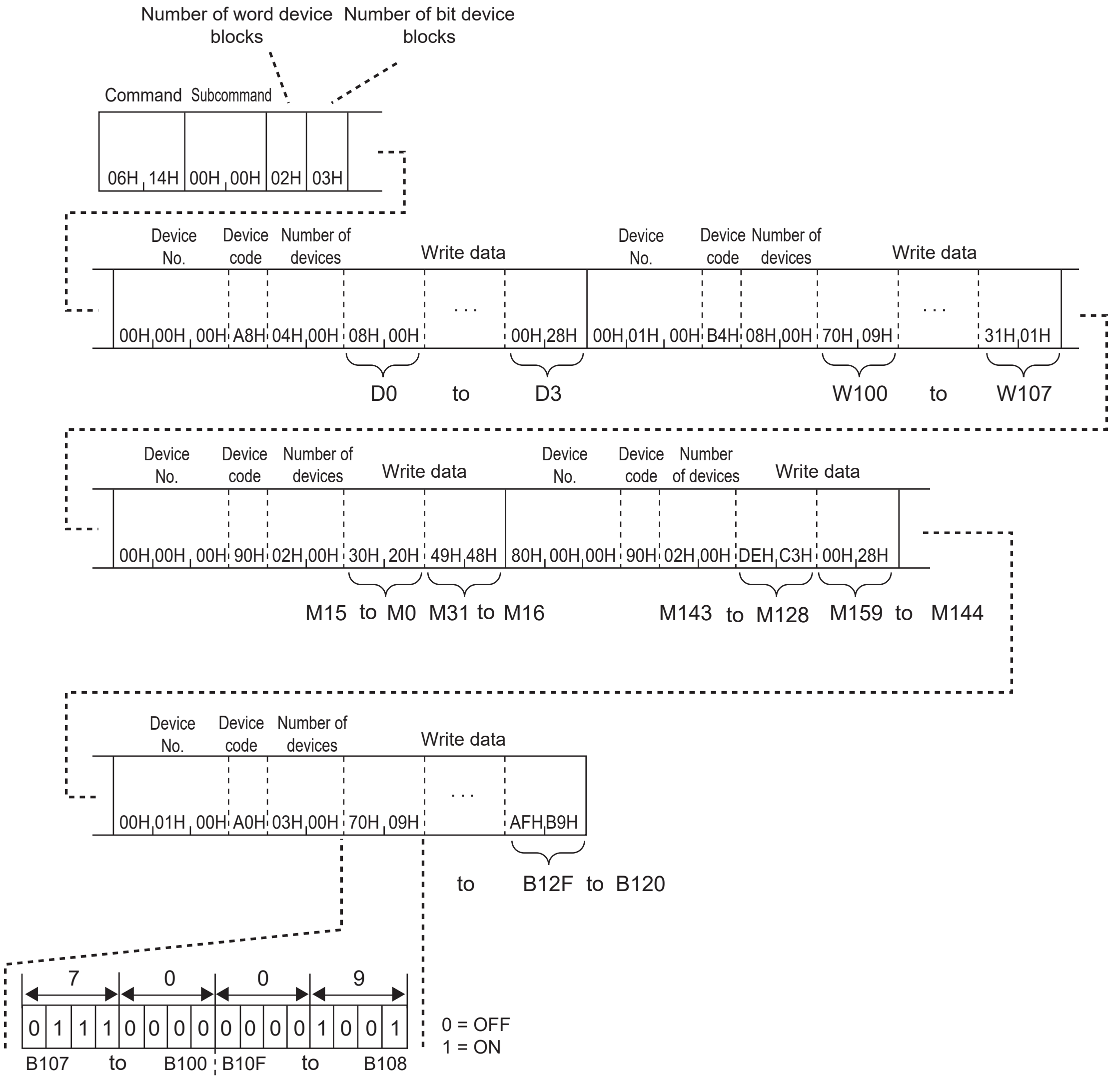
■When communicating data in ASCII code

(Request data)



■When communicating data in binary code

(Request data)



38.3 Remote Control

This section describes the command to set the SLMP compatible device or Ethernet-equipped module to the RUN status or STOP status by a message from the external device.

Before the remote operation

The accessed device or module is turned off and on or reset after remote operation

The information about the remote operation will be deleted.

Ex.

Even if the Remote STOP is executed when the switch of the Ethernet-equipped module is in the RUN status, the switch will return to the RUN status after resetting the module.

A remote password of the Ethernet-equipped module (access destination) is enabled

Remote operation from the external device is not available. An error will occur at the access destination, and an abnormal response will be sent back to the external device. Unlock the remote password of the Ethernet-equipped module side, and resend the request message.

Operable station in one command

Only one station can be operated remotely by one command.

When executing the remote operation to SLMP compatible device

It is recommended to use the UDP protocol for the remote operation. If TCP is used, the connection will be terminated when resetting. Therefore, reestablishing of connection is necessary.

Remote RUN

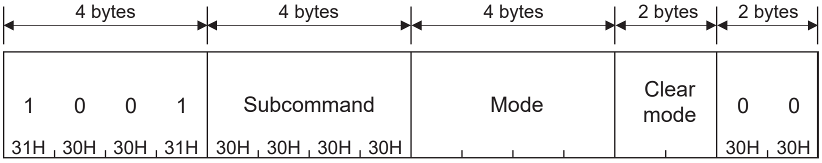
This command executes Remote RUN to the access destination module.

Point

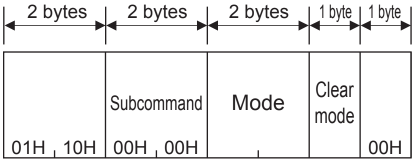
Remote RUN can be executed when the switch of the access destination module is in the RUN status. Even if the switch is in the STOP status, Remote RUN (command: 1001H) will be completed normally. However, the access destination does not change to the RUN status.

Request data

■When communicating data in ASCII code



■When communicating data in binary code



■Mode

This mode specifies whether Remote RUN can be executed forcibly by a device other than the external device which performed Remote STOP or Remote PAUSE. If forced execution is not allowed, Remote RUN can be executed only by the external device which performed Remote STOP or Remote PAUSE.

Forced execution is used when the external device which performed the remote operation cannot execute Remote RUN because of a problem with the device.

Item	**Mode**	
	ASCII code	**Binary code**
Forced execution not allowed (Remote RUN cannot be executed when other device executes Remote STOP or Remote PAUSE.)	0 0 0 1 30H, 30H, 30H, 31H	01H, 00H
Forced execution allowed (Remote RUN can be executed when other device executes Remote STOP or Remote PAUSE.)	0 0 0 3 30H, 30H, 30H, 33H	03H, 00H

■Clear mode

This mode specifies whether the clear (initialization) processing of device is executed when starting the calculation for the Remote RUN.

Only 00H is valid.

Item	**Mode**	
	ASCII code	**Binary code**
Do not clear the device	0 0 30H, 30H	00H

Response data

There is no response data for the Remote RUN command.

Communication example

Set mode to "Forced execution not allowed.", and set clear mode to "Clear all devices including that in the latch range" when executing Remote RUN.

- When communicating data in ASCII code

(Request data)

		Mode	Clear mode	
1 0 0 1	0 0 0 0	0 0 0 1	0 0	0 0
31H, 30H, 30H, 31H	30H, 30H, 30H, 30H	30H, 30H, 30H, 31H	30H, 30H	30H, 30H

- When communicating data in binary code

(Request data)

		Mode	Clear mode	
01H, 10H	00H, 00H	01H, 00H	00H	00H

Remote STOP

This command executes Remote STOP to the access destination module.

Request data

■When communicating data in ASCII code

4 bytes	4 bytes	4 bytes
1 0 0 2	Subcommand	0 0 0 0
31H, 30H, 30H, 32H	30H, 30H, 30H, 30H	30H, 30H, 30H, 30H

■When communicating data in binary code

2 bytes	2 bytes	2 bytes
	Subcommand	
02H, 10H	00H, 00H	00H, 00H

Response data

There is no response data for the Remote STOP command.

Communication example

Send request messages from the external device by using the message format shown in the request data above.

Remote PAUSE

This command executes Remote PAUSE to the access destination module.

Point

Remote PAUSE can be executed when the switch of the access destination module is in the RUN status. Even if the switch is in the STOP status, Remote PAUSE (command: 1003H) will be completed normally. However, the access destination does not change to the PAUSE status.

Request data

■When communicating data in ASCII code

4 bytes	4 bytes	4 bytes
1 0 0 3	Subcommand	Mode
31H, 30H, 30H, 33H	30H, 30H, 30H, 30H	

■When communicating data in binary code

2 bytes	2 bytes	2 bytes
	Subcommand	Mode
03H, 10H	00H, 00H	

■Mode

This mode specifies whether Remote PAUSE can be executed forcibly by a device other than the external device which performed Remote STOP or Remote PAUSE. If forced execution is not allowed, Remote PAUSE can be executed only by the external device which performed Remote STOP or Remote PAUSE.
Forced execution is used when the external device which performed the remote operation cannot execute Remote PAUSE because of a problem with the device.

Item	Mode	
	ASCII code	Binary code
Forced execution not allowed (Remote PAUSE cannot be executed when other device executes Remote STOP or Remote PAUSE.)	0 0 0 1 30H 30H 30H 31H	01H 00H
Forced execution allowed (Remote PAUSE can be executed when other device executes Remote STOP or Remote PAUSE.)	0 0 0 3 30H 30H 30H 33H	03H 00H

Response data

There is no response data for the Remote PAUSE command.

Communication example

Set mode to "Forced execution not allowed" when executing Remote PAUSE.

■When communicating data in ASCII code

(Request data)

		Mode
1 0 0 3 31H 30H 30H 33H	0 0 0 0 30H 30H 30H 30H	0 0 0 1 30H 30H 30H 31H

■When communicating data in binary code

(Request data)

		Mode
03H 10H	00H 00H	01H 00H

Remote latch clear

This command executes remote latch clear to the access destination module.

Point

Before executing the remote latch clear, set the status of the access destination module to STOP.
While the access destination is stopped or paused remotely by request from another external device:
- The remote latch clear cannot be executed. Abnormal completion of the command will occur.
- Cancel the Remote STOP or Remote PAUSE before executing the command.

Request data

■When communicating data in ASCII code

4 bytes	4 bytes	4 bytes
1 0 0 5 31H 30H 30H 35H	Subcommand 30H 30H 30H 30H	0 0 0 0 30H 30H 30H 30H

■When communicating data in binary code

2 bytes	2 bytes	2 bytes
	Subcommand	
05H , 10H	00H , 00H	00H , 00H

Response data

There is no response data for remote latch clear command.

Communication example

Send request messages from the external device by using the message format shown in the request data above.

Remote RESET

This command executes Remote RESET to the access destination module. Remote RESET is used to restore when an error occurred in the SLMP compatible device.

Point

Before executing Remote RESET, perform the following.

- When the access destination module has a Remote RESET enable/disable setting, go to GX Works3 Navigation window ⇨ [Parameter] ⇨ Module model ⇨ [CPU Parameter] ⇨ [Operation Related Setting] ⇨ [Remote Reset Setting], and select "Enable" for "Remote Reset". (Default: Disable)
- Set the status of the access destination module to STOP.

Precautions

- In some cases, Remote RESET cannot be executed because of hardware error, etc.
- The response message when Remote RESET is executed may not be sent back to the external device since the access destination is reset.

Request data

■When communicating data in ASCII code

4 bytes	4 bytes	4 bytes
1 0 0 6	Subcommand	0 0 0 0
31H , 30H , 30H , 36H	30H , 30H , 30H , 30H	30H , 30H , 30H , 30H

■When communicating data in binary code

2 bytes	2 bytes	2 bytes
	Subcommand	
06H , 10H	00H , 00H	00H , 00H

Response data

There is no response data for the Remote RESET command.

Communication example

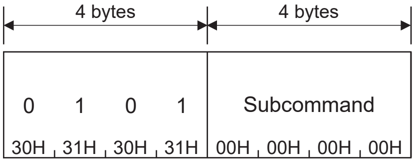
Send request messages from the external device by using the message format shown in the request data above.

Processor type read

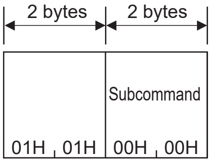
This command reads the processor module name code (processor type) of the access destination module.

Request data

■When communicating data in ASCII code

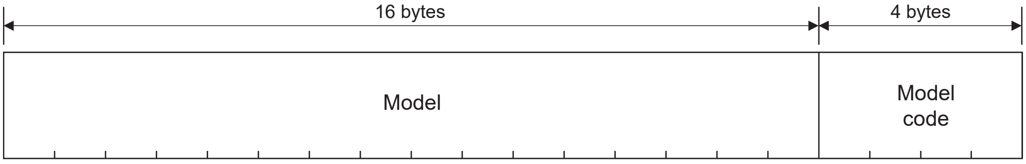


■When communicating data in binary code

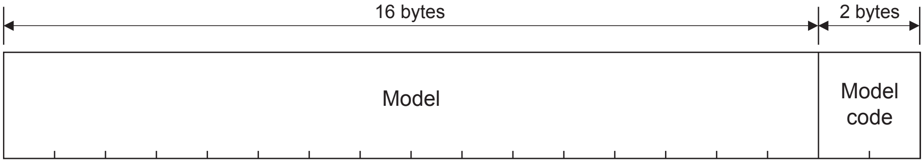


Response data

■When communicating data in ASCII code



■When communicating data in binary code



■Model

The characters of the module model are stored for 16 characters from the upper byte.

If the model to be read is less than 16 characters, space (20H) is stored for the remaining characters. Even when communicating data in binary code, the module model is stored in ASCII code.

■Model code

The following model codes will be stored.

When communicating in ASCII code, the data is stored in order from the upper byte to the lower byte.

When communicating in binary code, the data is stored in order from the lower byte to the upper byte.

Model	Model code (hexadecimal)
FX5U-32MR/ES	4A21H
FX5U-64MR/ES	4A23H
FX5U-80MR/ES	4A24H
FX5U-32MT/ES	4A29H
FX5U-64MT/ES	4A2BH
FX5U-80MT/ES	4A2CH
FX5U-32MT/ESS	4A31H
FX5U-64MT/ESS	4A33H
FX5U-80MT/ESS	4A34H
FX5U-32MR/DS	4A41H
FX5U-64MR/DS	4A43H
FX5U-80MR/DS	4A44H

Model	Model code (hexadecimal)
FX5U-32MT/DS	4A49H
FX5U-64MT/DS	4A4BH
FX5U-80MT/DS	4A4CH
FX5U-32MT/DSS	4A51H
FX5U-64MT/DSS	4A53H
FX5U-80MT/DSS	4A54H
FX5UC-32MT/D	4A91H
FX5UC-64MT/D	4A92H
FX5UC-96MT/D	4A93H
FX5UC-32MT/DSS	4A99H
FX5UC-64MT/DSS	4A9AH
FX5UC-96MT/DSS	4A9BH
FX5UC-32MR/DS-TS	4AA9H
FX5UC-32MT/DS-TS	4AB1H
FX5UC-32MT/DSS-TS	4AB9H
FX5UJ-24MR/ES	4B0DH
FX5UJ-40MR/ES	4B0EH
FX5UJ-60MR/ES	4B0FH
FX5UJ-24MT/ES	4B14H
FX5UJ-40MT/ES	4B15H
FX5UJ-60MT/ES	4B16H
FX5UJ-24MT/ESS	4B1BH
FX5UJ-40MT/ESS	4B1CH
FX5UJ-60MT/ESS	4B1DH
FX5UJ-24MR/DS	4B22H
FX5UJ-40MR/DS	4B23H
FX5UJ-60MR/DS	4B24H
FX5UJ-24MT/DS	4B29H
FX5UJ-40MT/DS	4B2AH
FX5UJ-60MT/DS	4B2BH
FX5UJ-24MT/DSS	4B30H
FX5UJ-40MT/DSS	4B31H
FX5UJ-60MT/DSS	4B32H
FX5S-30MR/ES	4B4EH
FX5S-40MR/ES	4B4FH
FX5S-60MR/ES	4B50H
FX5S-80MR/ES[*1]	4B51H
FX5S-30MT/ES	4B55H
FX5S-40MT/ES	4B56H
FX5S-60MT/ES	4B57H
FX5S-80MT/ES[*1]	4B58H
FX5S-30MT/ESS	4B5CH
FX5S-40MT/ESS	4B5DH
FX5S-60MT/ESS	4B5EH
FX5S-80MT/ESS[*1]	4B5FH
FX5S-30MR/DS	4B63H
FX5S-40MR/DS	4B64H
FX5S-60MR/DS	4B65H
FX5S-80MR/DS[*1]	4B66H
FX5S-30MT/DS	4B6AH
FX5S-40MT/DS	4B6BH
FX5S-60MT/DS	4B6CH
FX5S-80MT/DS[*1]	4B6DH

Model	Model code (hexadecimal)
FX5S-30MT/DSS	4B71H
FX5S-40MT/DSS	4B72H
FX5S-60MT/DSS	4B73H
FX5S-80MT/DSS[*1]	4B74H

*1 Area-specific model

Point

- The model of the CPU module is identified by the model code.
- When the Ethernet module is used, the model code of the connected CPU module is stored.

Communication example

■When communicating data in ASCII code

(Request data)

0	1	0	1	0	0	0	0
30H	31H	30H	31H	30H	30H	30H	30H

(Response data)

F	X	5	U	-	3	2	M	R	/	E	S					4	A	2	1
46H	58H	35H	55H	2DH	33H	32H	4DH	52H	2FH	45H	53H	20H	20H	20H	20H	34H	41H	32H	31H

■When communicating data in binary code

(Request data)

01H	01H	00H	00H

(Response data)

F	X	5	U	-	3	2	M	R	/	E	S						
46H	58H	35H	55H	2DH	33H	32H	4DH	52H	2FH	45H	53H	20H	20H	20H	20H	21H	4AH

38.4 Clear Error

This function turns off ERR LED of the CPU module from the external equipment and/or initializes the communication error information or error code stored in the buffer memory.

This function is used to initialize the current error information due to an abnormal response for a command message and return it to the normal state or initialize the error code storage area of the buffer memory.

The order and description of the data item with * shown in the figure of the control procedure differ depending on the frame and pattern in a communication.

Point

This function can be used only for the CPU module which is connected with the external equipment.
This function cannot be used for the CPU module of another station via the network system.

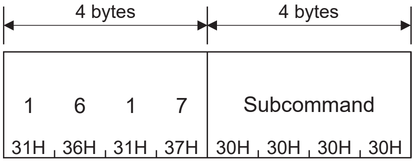
The data part of the command and control procedure when the display LEDs of the CPU module are turned off and the communication error information is initialized from the external equipment is described.

Command

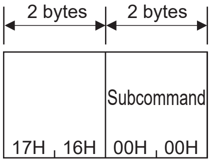
Function	Command (Subcommand)	Processing content
Clear Error	1617 (0000)	Turns off the display LEDs, initializes the error code, and others.

Request data

■When communicating data in ASCII code



■When communicating data in binary code



Response data

There is no response data for the Clear Error command.

Communication example

Send request messages from the external device by using the message format shown in the request data above.

38.5 Self-Test

This function tests whether the communication function between the external equipment and Ethernet-equipped module operates normally or not. The control procedure when this function is used is described with examples.

Point

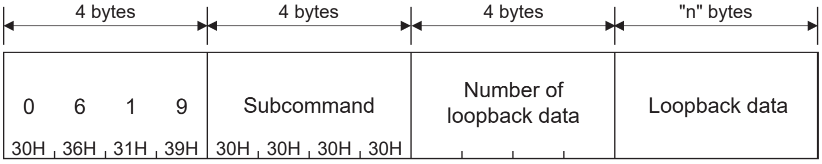
- At the startup of the Ethernet-equipped module or when trouble occurs, this function can check whether the connection between the external equipment and Ethernet-equipped module is correct and/or whether the data communication function operates normally.
- This function can be used only for the Ethernet-equipped module which is connected with the external equipment (including a multi-drop connecting station). This function cannot be used for the Ethernet-equipped module of another station via the network system.

Command

Function	Command (Subcommand)	Processing content
Self-Test	0619 (0000)	Checks whether a data communication is executed normally.

Request data

■When communicating data in ASCII code



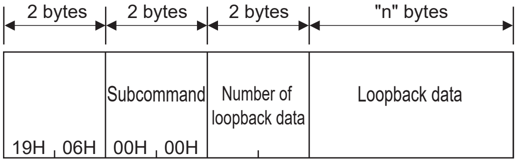
- Number of loopback data (number of bytes)

The number of the bytes is converted into a 4-digit ASCII code (hexadecimal) and data is sent from the upper digit ("0").

- Loopback data

The order of character strings for up to 960 1-byte characters ("0" to "9", "A" to "F") is sent from the head.

■When communicating data in binary code



- Number of loopback data (number of bytes)

The 2-byte numerical value which indicates the number of the bytes is used and data is sent from the low byte (L: bit 0 to 7).

- Loopback data

Data is sent for up to 960 bytes from the head by treating each character code ("0" to "9", "A" to "F") as a 1 byte value.

Response data

The same number of the loopback data and loopback data which the external equipment sent are sent back to the external equipment.

Communication example

Send request messages from the external device by using the message format in the request data. (☞ Page 660 Request data)

Examples of test with loopback data "ABCDE" are given below.

■When executing the Self-Test by communicating in ASCII code

(Request data)

Command	Subcommand	Number of loopback data	Loopback data
0 6 1 9	0 0 0 0	0 0 0 5	A B C D E
30H 36H 31H 39H	30H 30H 30H 30H	30H 30H 30H 35H	41H 42H 43H 44H 45H

(Response data)

Number of loopback data	Loopback data
0 0 0 5	A B C D E
30H 30H 30H 35H	41H 42H 43H 44H 45H

■When executing the Self-Test by communicating in binary code

(Request data)

Command	Subcommand	Number of loopback data	Loopback data
			A B C D E
19H 06H	00H 00H	05H 00H	41H 42H 43H 44H 45H

(Response data)

Number of loopback data	Loopback data
	A B C D E
05H 00H	41H 42H 43H 44H 45H

38.6 Remote Password Unlock or Lock

A remote password prevents illegal access from a user who is not allowed to operate the SLMP compatible device. The following modules support this function.

- FX5 CPU module
- FX5-CCLGN-MS
- FX5-CCLIEF
- FX5-40SSC-G, FX5-80SSC-G

If a remote password is set to the SLMP compatible device, the remote password is checked when the SLMP compatible device is accessed.

The following shows how to use a command to lock or unlock the remote password by the SLMP.

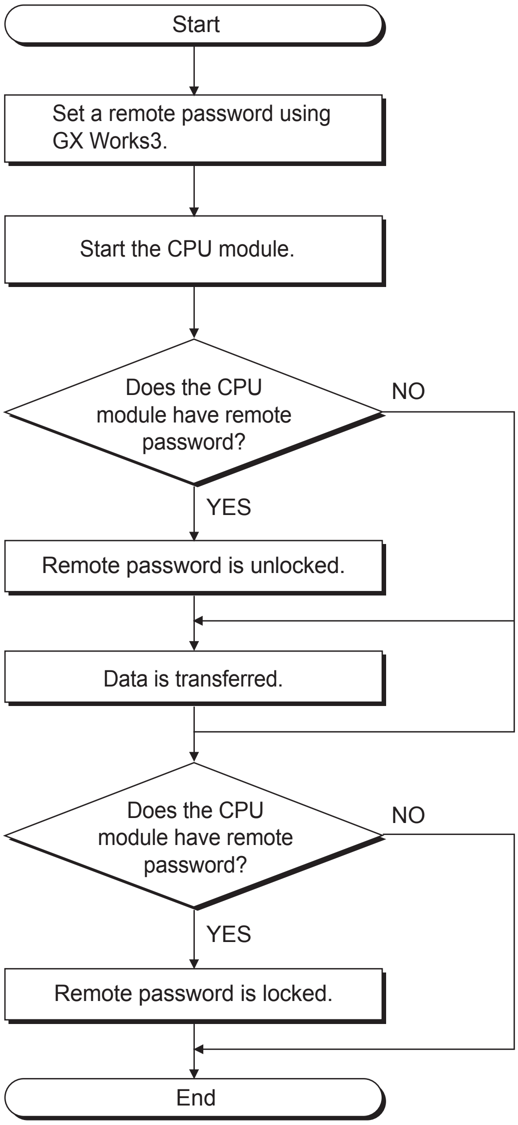
Target of the remote password checking function

When a remote password is set to the SLMP compatible device, unlock the remote password using a command in this section. Then execute data communication.

Control procedure

The following shows the control procedure when a remote password is set to the SLMP compatible device.

■When accessing the FX5CPU



Point

- When the FX5CPU communicating data is set with a remote password, communication is enabled after the completion of the unlock process until the lock process.
- All commands received while the remote password is in locked status will generate an error response. (Execute communication after executing the remote password unlock process.)
- The remote password lock process is automatically performed when the line is disconnected.

Lock

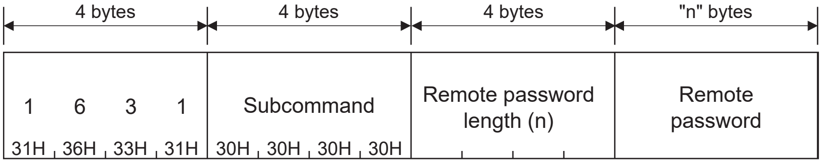
This command changes the remote password from unlocked status to locked status. (Sets the device to the state where communication is not possible.)

Command

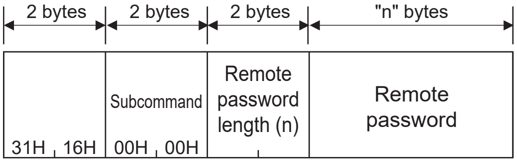
Function		Command (Subcommand)	Processing content
Remote password	Lock	1631 (0000)	Specifies a remote password and changes the unlock status to the lock status. (Communication to the CPU module is disabled.)

Request data

■When communicating data in ASCII code



■When communicating data in binary code



■Subcommand

Item		Subcommand					
		ASCII code				Binary code	
Default	Characters	0	0	0	0	—	—
	Character code	30H	30H	30H	30H	00H	00H

■Remote password length

Remote password length is not used.

■Remote password

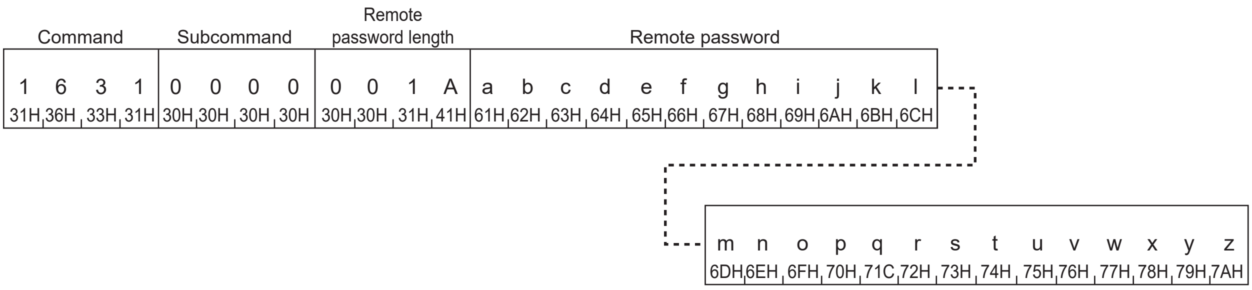
Remote password is not used.

Response data

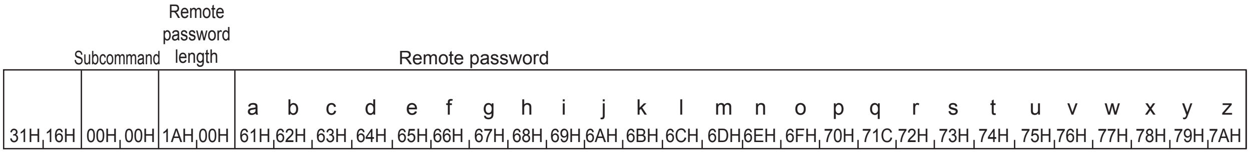
There is no response data for the lock command of the remote password.

Communication example

■When performing the lock process in communication using ASCII code



■When performing the lock process in communication using binary code



Unlock

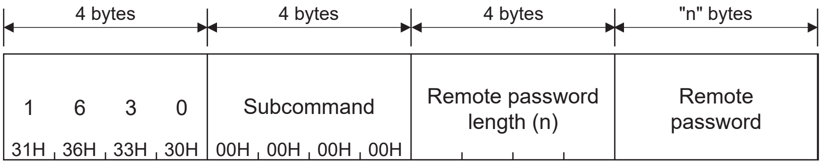
This command changes the remote password from locked status to unlocked status. (Sets the device to the state where communication is possible.)

Command

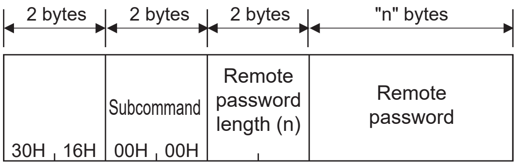
Function		Command (Subcommand)	Processing content
Remote password	Unlock	1630 (0000)	Specifies a remote password and changes the lock status to the unlock status. (Communication to the CPU module is enabled.)

Request data

■When communicating data in ASCII code



■When communicating data in binary code



■Subcommand

Item		Subcommand					
		ASCII code				Binary code	
Default	Characters	0	0	0	0	—	—
	Character code	30H	30H	30H	30H	00H	00H

■Remote password length

Specify the remote password length.

The password length is the specified characters (6 to 32 characters).

Item		Remote password length (when the number of remote password characters is 32)					
		ASCII code				Binary code	
6 to 32 characters	Characters	0	0	2	0	—	—
	Character code	30H	30H	32H	30H	20H	00H

■Remote password

Specify the remote password set for the SLMP-compatible device, CPU module or intelligent function module using GX Works3.

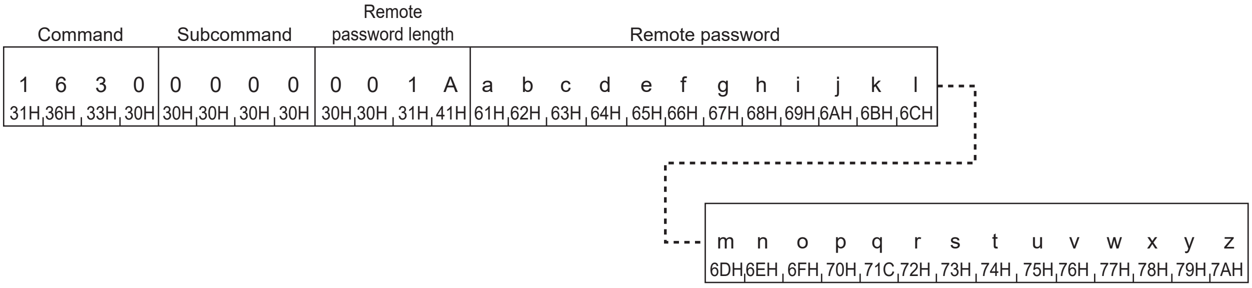
Specify the remote password using ASCII code also when communicating using binary code.

Response data

There is no response data for the unlock command of the remote password.

Communication example

■When performing the unlock process in communication using ASCII code



Command	Subcommand	Remote password length	Remote password
1 6 3 0	0 0 0 0	0 0 1 A	a b c d e f g h i j k l
31H, 36H, 33H, 30H	30H, 30H, 30H, 30H	30H, 30H, 31H, 41H	61H, 62H, 63H, 64H, 65H, 66H, 67H, 68H, 69H, 6AH, 6BH, 6CH
			m n o p q r s t u v w x y z
			6DH, 6EH, 6FH, 70H, 71C, 72H, 73H, 74H, 75H, 76H, 77H, 78H, 79H, 7AH

■When performing the unlock process in communication using binary code

Command	Subcommand	Remote password length	Remote password
			a b c d e f g h i j k l m n o p q r s t u v w x y z
30H, 16H	00H, 00H	1AH, 00H	61H, 62H, 63H, 64H, 65H, 66H, 67H, 68H, 69H, 6AH, 6BH, 6CH, 6DH, 6EH, 6FH, 70H, 71C, 72H, 73H, 74H, 75H, 76H, 77H, 78H, 79H, 7AH

39 1E FRAME COMMANDS

This chapter explains 1E frame commands of SLMP.

For parts of the transmission message other than the command part, refer to ☞ Page 587 1E Frame.

39.1 List of Commands and Functions

This section describes commands and functions when accessing from the external equipment to the Ethernet-equipped module.

Name	**Command**	**Processing content**	**Number of points processed per communication**
Batch Reading	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
Batch 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 word 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	Requests remote RUN/STOP to an Ethernet-equipped module.[*5]	—
Remote STOP	14H		
Read PC Type Name	15H	Reads Ethernet-equipped module model code.	—
Loopback Test	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 of the CPU module is turned 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 CPU module is enabled.

39.2 Device Access

This section explains the control procedure specification method and shows a specification example when the device memory is read and written.

Commands

This section explains commands when the device memory is read or written.

Commands

Name		Command	Processing content
Batch Reading	Bit units	00H	Reads bit devices in 1 point units.
	Word units	01H	Reads bit devices in 16-point units.
			Reads word devices in 1 point units.
Batch Writing	Bit units	02H	Writes bit devices in 1-point units.
	Word units	03H	Writes bit devices in 16-point units.
			Writes word devices in 1-point units.
Test (random Write)	Bit units	04H	Sets or resets device memory to bit devices specified randomly in 1-point units.
	Word units	05H	Sets or resets device memory to bit devices specified randomly in 16-point units.
			Writes device memory to word devices specified randomly in 1-point units.

Device range

This section shows accessible CPU module device.

Specify the device and device number range that exist in the module targeted for data read or write.

In the case of Ethernet-equipped module

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" at 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.

Batch Reading

Data in devices are read in a batch.

Request data

■When communicating data in ASCII code

2 bytes	2 bytes	4 bytes	4 bytes	8 bytes	2 bytes	2 bytes
0 0 30H, 30H	PC No.	Monitoring timer	Device name	Head device No.	Number of devices	Fixed value

■When communicating data in binary code

1 byte	1 byte	2 bytes	4 bytes	2 bytes	1 byte	1 byte
00H	PC No.	Monitoring timer	Head device No.	Device name	Number of devices	Fixed value

■Subheader

Specify the command selected from the data size.

Data size	Command		
	ASCII code (Upper column: characters, lower column: character code)		Binary code
Bit devices in 1 point units	0	0	00H
	30H	30H	
Bit devices in 16 point units Word devices in 1 point units	0	1	01H
	30H	31H	

■PC No.

Specify the "FFH".

■Monitoring timer

Specify the "0000H".

■Device name

Specify the device code that corresponds to the device type to be read. (☞ Page 668 Device range)

■Head device No.

Specify the head number of target device of reading.

■Number of devices

Specify the number of devices of reading.

Item	Number of devices
When reading data in 1-bit units	1 to 256 points
When reading data in 16-bit units	1 to 32 words (16 to 512 points)
When reading data in 1-word units	1 to 64 points

Point

To specify 256 for the number of device points, specify "00H".

■Fixed value

Specify the "00H".

Response data

The read device value is stored in hexadecimal. The data order differs depending on the ASCII code or binary code.

Read data

Communication example

■When reading data in bit units

Values in M100 to M111 are read.

- When communicating data in ASCII code

(Request data)

Subheader	PC No.	Monitoring timer	Device name	Head device No.	Number of devices	Fixed value
0 0	F F	0 0 0 0	4 D 2 0	0 0 0 0 0 0 6 4	0 C	0 0
30H,30H	46H,46H	30H,30H,30H,30H	34H,44H,32H,30H	30H,30H,30H,30H,30H,30H,36H,34H	30H,43H	30H,30H

(Response data)

1 0 0 0 1 0 1 1 0 0 0 0
31H,30H,30H,30H,31H,30H,31H,31H,30H,30H,30H,30H

M100 to M111

0 = OFF
1 = ON

- When communicating data in binary code

(Request data)

Subheader	PC No.	Monitoring timer	Head device No.	Device name	Number of devices	Fixed value
00H	FFH	00H,00H	64H,00H,00H,00H	20H,4DH	0CH	00H

(Response data)

10H,10H,10H,10H,10H,10H

M101
M100
to
M111
M110

0 = OFF
1 = ON

■When reading data in word units (bit device)

Y100 to Y137 (2-word) are read.

- When communicating data in ASCII code (X, Y HEX)

(Request data)

Subheader	PC No.	Monitoring timer	Device name	Head device No.	Number of devices	Fixed value
0 1	F F	0 0 0 0	5 9 2 0	0 0 0 0 0 0 4 0	0 2	0 0
30H,30H	46H,46H	30H,30H,30H,30H	35H,39H,32H,30H	30H,30H,30H,30H,30H,30H,34H,30H	30H,32H	30H,30H

(Response data)

8 2 9 D	5 5 3 E
38H 32H 39H 44H	35H 35H 33H 45H

8	2	9	D	5	5	3	E
1 0 0 0	0 0 1 0	1 0 0 1	1 1 0 1	0 1 0 1	0 1 0 1	0 0 1 1	1 1 1 0

Y117 to Y100 ¦ Y137 to Y120

0 = OFF
1 = ON

- When communicating data in binary code

(Request data)

Subheader	PC No.	Monitoring timer	Head device No.	Device name	Number of devices	Fixed value
01H	FFH	00H,00H	40H,00H,00H,00H	20H,59H	02H	00H

(Response data)

9DH 82H 3EH 55H

9	D	8	2	3	E	5	5
1 0 0 1	1 1 0 1	1 0 0 0	0 0 1 0	0 0 1 1	1 1 1 0	0 1 0 1	0 1 0 1
Y107 to Y100		Y117 to Y110		Y127 to Y120		Y137 to Y130	

0 = OFF
1 = ON

■When reading data in word units (word device)

Values in D100 to D102 are read.

It is supposed that 4660 (1234H) is stored in D100, 5 (5H) is stored in D101, and 26505 (6789H) is stored D102.

- When communicating data in ASCII code

(Request data)

Subheader	PC No.	Monitoring timer	Device name	Head device No.	Number of devices	Fixed value
0 1	F F	0 0 0 0	4 4 2 0	0 0 0 0 0 0 6 4	0 3	0 0
30H,30H	46H,46H	30H,30H,30H,30H	34H,34H,32H,30H	30H,30H,30H,30H,30H,30H,36H,34H	30H,33H	30H,30H

(Response data)

1 2 3 4	0 0 0 5	6 7 8 9
31H 32H 33H 34H	30H 30H 30H 35H	36H 37H 38H 39H
D100	D101	D102

- When communicating data in binary code

(Request data)

Subheader	PC No.	Monitoring timer	Head device No.	Device name	Number of devices	Fixed value
01H	FFH	00H,00H	64H,00H,00H,00H	20H,44H	03H	00H

(Response data)

34H,12H	05H,00H	89H,67H
D100	D101	D102

Batch Writing

Data in devices are written in a batch.

Request data

■When communicating data in ASCII code

2 bytes	2 bytes	4 bytes	4 bytes	8 bytes	2 bytes	2 bytes
0 2 30H, 32H	PC No.	Monitoring timer	Device name	Head device No.	Number of devices	Fixed value

■When communicating data in binary code

1 byte	1 byte	2 bytes	4 bytes	2 bytes	1 byte	1 byte
02H	PC No.	Monitoring timer	Head device No.	Device name	Number of devices	Fixed value

■Subheader

Specify the command selected based on the data size.

Data size	Command		
	ASCII code (Upper column: characters, lower column: character code)		Binary code
Bit devices in 1 point units	0	2	02H
	30H	32H	
Bit devices in 16 point units Word devices in 1 point units	0	3	03H
	30H	33H	

■PC No.

Specify the "FFH".

■Monitoring timer

Specify the "0000H".

■Device name

Specify the device code that corresponds to the device type to be read. (☞ Page 668 Device range)

Precautions

When specifying the command 03H, observe the following instructions.

- When the CPU module uses long counters, specify the number of long counter points × 2 for the number of device points.
- The Ethernet module cannot use long counters.

■Head device No.

Specify the head number of target device of reading.

■Number of devices

Specify the number of devices of reading.

Item	Number of devices
When reading data in 1-bit units	1 to 160 points
When reading data in 16-bit units	1 to 10 words (16 to 160 points)
When reading data in 1-word units	1 to 64 points

■Fixed value

Specify the "00H".

Response data

There is no response data for the Batch Writing command.

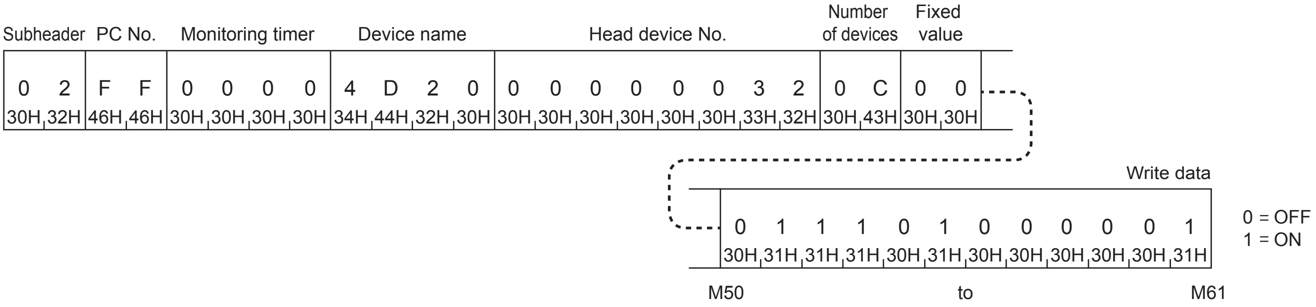
Communication example

■When writing data in bit units

Values are written to M50 to M61.

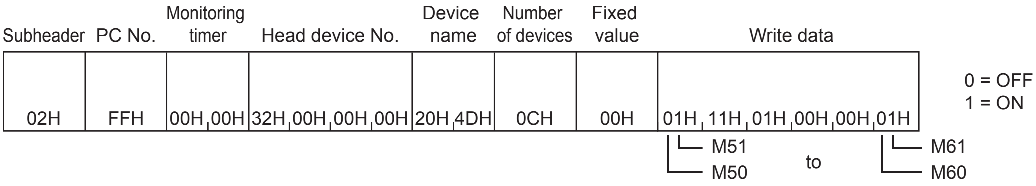
- When communicating data in ASCII code

(Request data)



- When communicating data in binary code

(Request data)

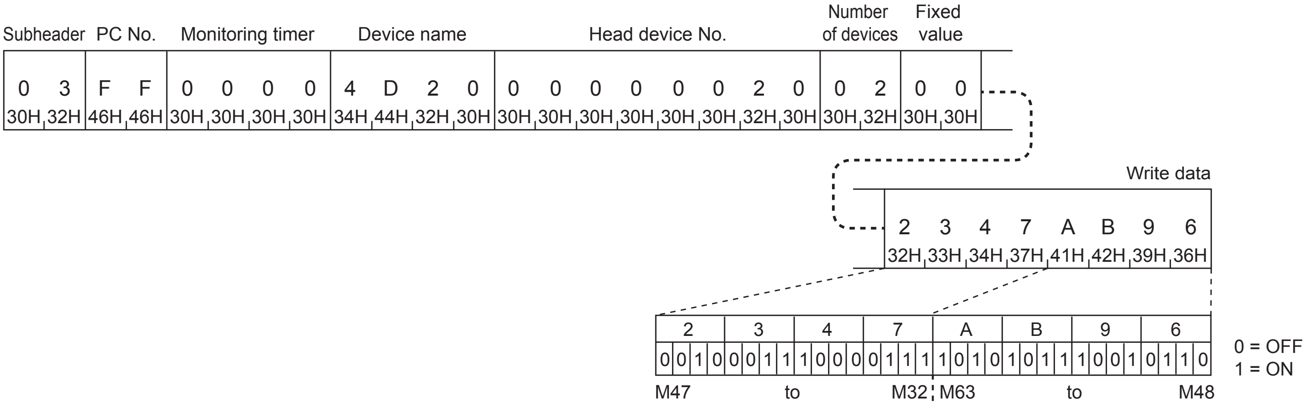


■When writing data in word units (bit device)

Values are written to M32 to M63 (2-word).

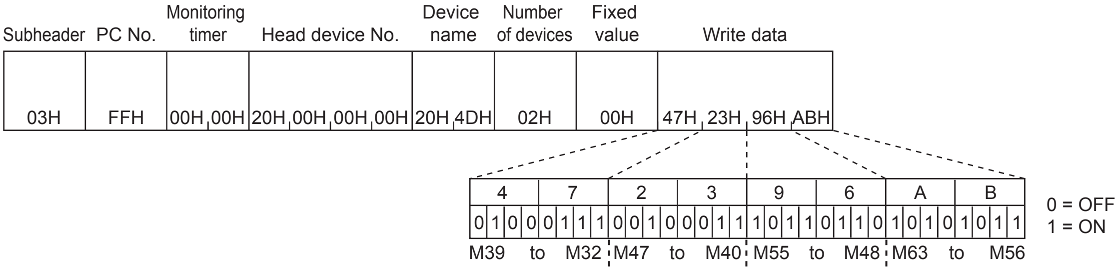
- When communicating data in ASCII code

(Request data)



- When communicating data in binary code

(Request data)

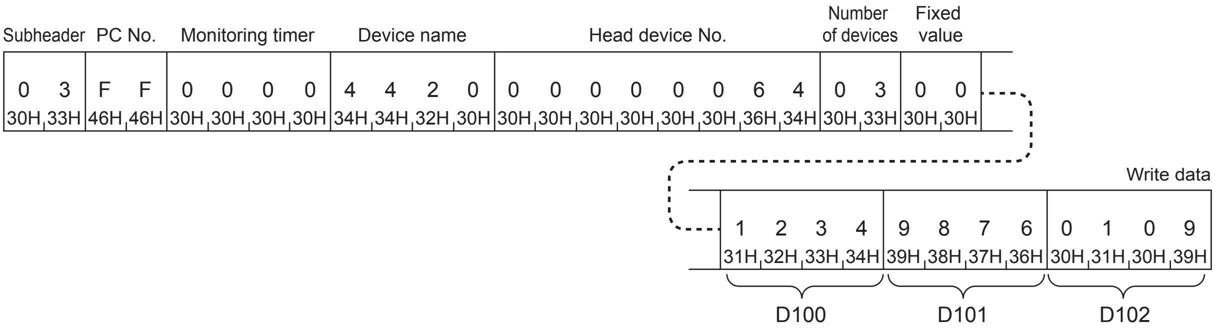


■When writing data in word units (word device)

4660 (1234H) is written in D100, 39030 (9876H) is written in D101, and 265 (109H) is written in D102.

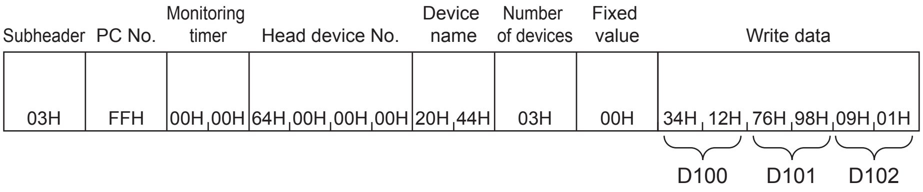
- When communicating data in ASCII code

(Request data)



- When communicating data in binary code

(Request data)



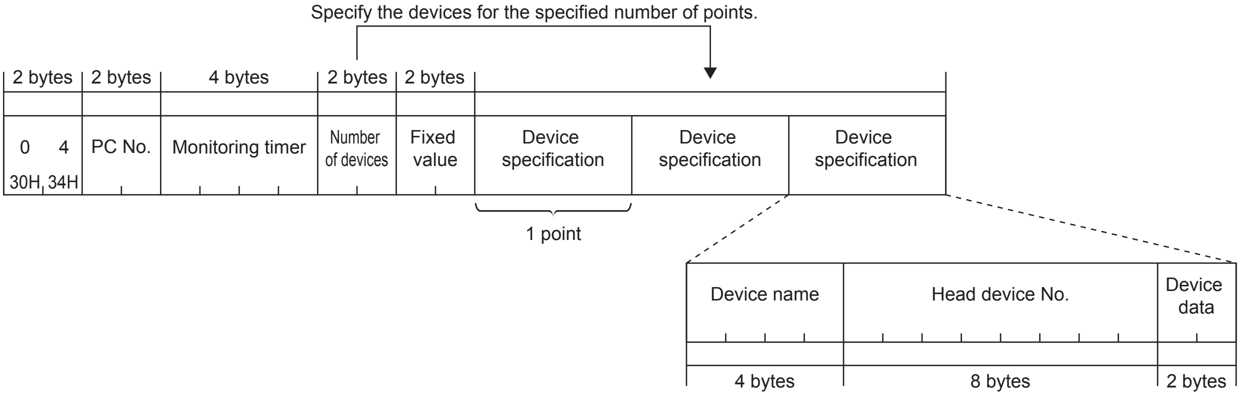
Test (random Write)

This command specifies the device No. randomly and writes the data.

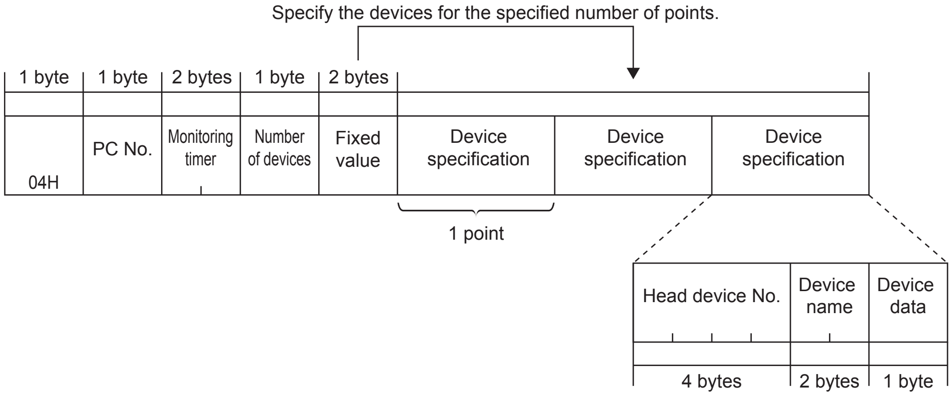
Request data

■When writing data in bit units

- When communicating data in ASCII code

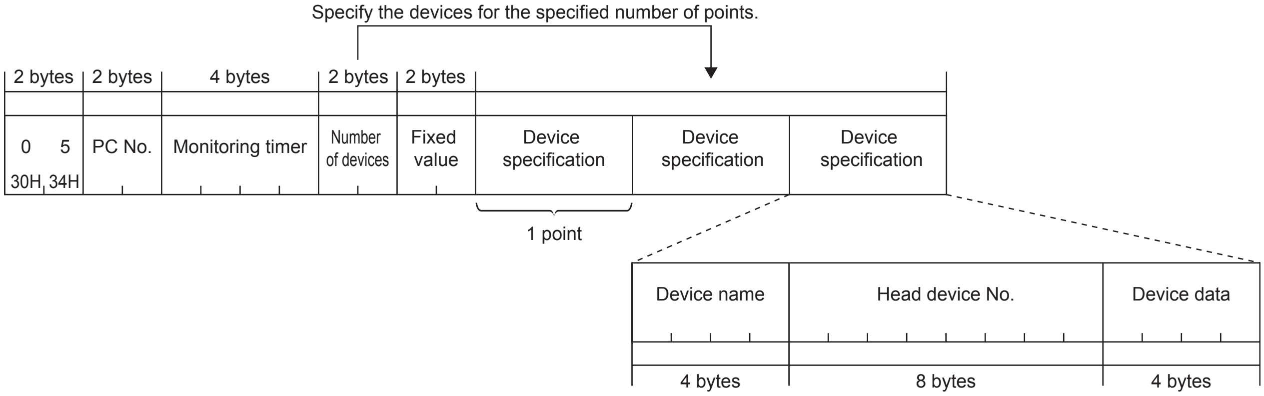


- When communicating data in binary code

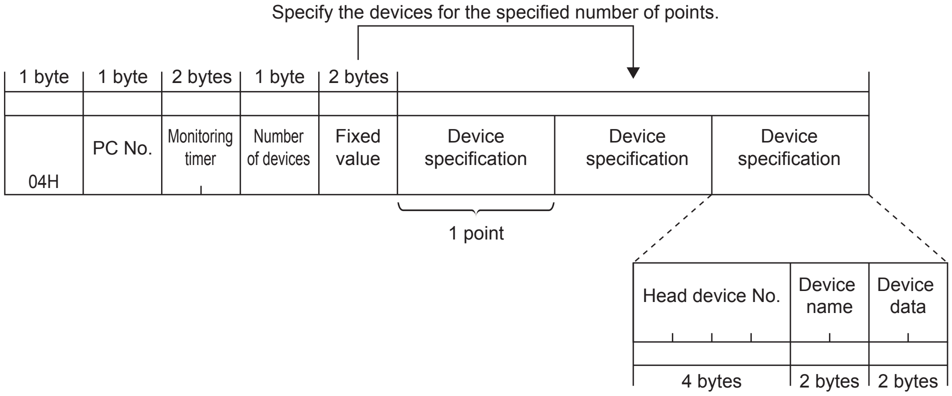


■When writing data in word units

- When communicating data in ASCII code



- When communicating data in binary code



■Subheader

Specify the command selected from the data size.

Data size	Command		
	ASCII code (Upper column: characters, lower column: character code)		Binary code
Bit devices in 1 point units	0	4	04H
	30H	34H	
Bit devices in 16 point units Word devices in 1 point units	0	5	05H
	30H	35H	

■PC No.

Specify the "FFH".

■Monitoring timer

Specify the "0000H".

■Device name

Specify the device code that corresponds to the device type to be read. (☞ Page 668 Device range)

Precautions

When the command 05H is specified in the Ethernet module, TS (timer (contact)), CS (counter (contact)) and LCS (long timer (contact)) cannot be used.

■Head device No.

Specify the head number of target device of reading.

■Number of devices

Specify the number of devices of reading.

Item	Number of devices
When reading data in 1-bit units	1 to 80 points
When reading data in 16-bit units	1 to 10 words (16 to 160 points)
When reading data in 1-word units	1 to 10 points

■Fixed value

Specify the "00H".

■Device data

Specify the value to be written to the device.

Device units	Communicating data	
	ASCII code	Binary code
Bit units (1 point units)	If the specified device is ON, express it as "01". If it is OFF, express it as "00". Convert the code into a 2-digit ASCII code, and specify in the order from the most significant byte to the least significant byte.	If the specified device is ON, express it as "01". If it is OFF, express it as "00". Specify with a 1-byte binary code.
Word units (Bit 16-point units)	Specify one word (16 bits) with a 4-digit ASCII code, and express 1-point with 1-bit ON/OFF. Specify in 1-word units in the order from the most significant bit to the least significant bit (b15 to b0).	Specify one word (2 bytes) with a binary code, and express 1-point with 1-bit ON/OFF. Specify in 1-word units in the order from the least significant byte (b7 to b0) to the least significant byte (b15 to b8).
Word units (Word 1-point units)	Express the value to be written with a 1-word (4-byte) 4-digit ASCII code (hexadecimal). Specify in 1-word units in the order from the most significant byte to the least significant byte.	Express the value to be written with a 1-word (2-byte) 4-digit binary code (hexadecimal). Specify in 1-word units in the order from the least significant byte to the most significant byte.

Response data

There is no response data for the Test (Random Write) command.

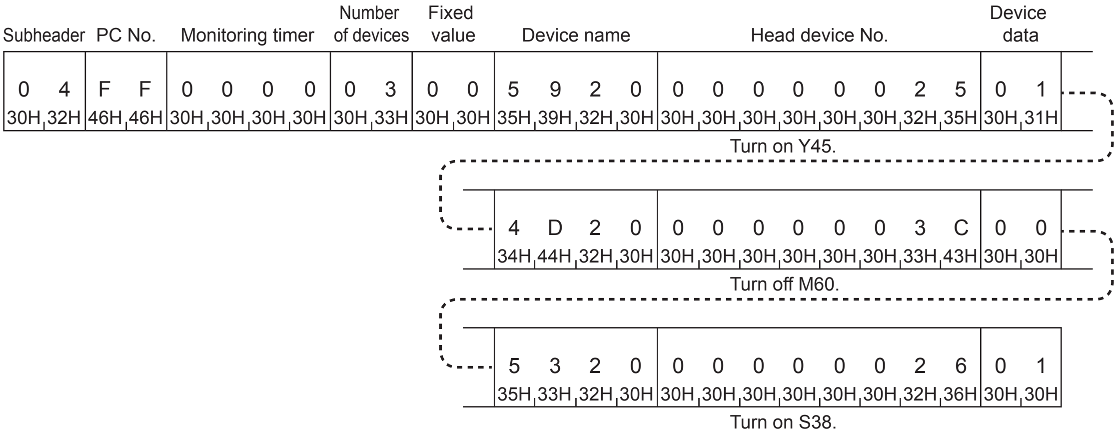
Communication example

■When writing data in bit units

Turn on Y45, turn off M60 and turn on S38.

- When communicating data in ASCII code (X, Y HEX)

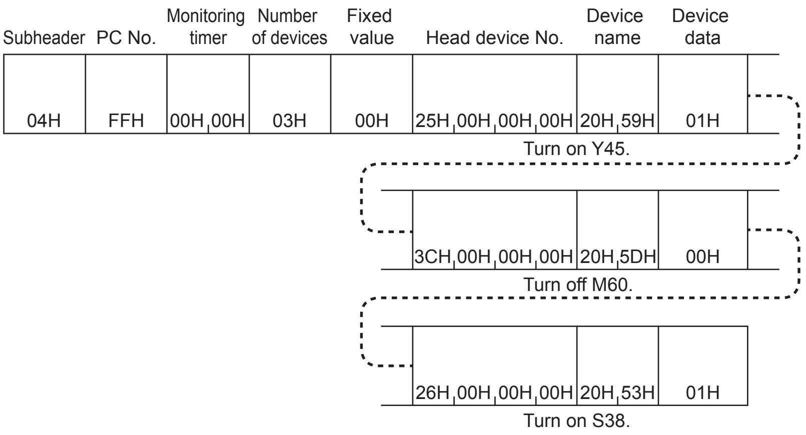
(Request data)



Subheader	PC No.	Monitoring timer	Number of devices	Fixed value	Device name	Head device No.	Device data
0 4 30H,32H	F F 46H,46H	0 0 0 0 30H,30H,30H,30H	0 3 30H,33H	0 0 30H,30H	5 9 2 0 35H,39H,32H,30H	0 0 0 0 0 0 2 5 30H,30H,30H,30H,30H,30H,32H,35H	0 1 30H,31H
					4 D 2 0 34H,44H,32H,30H	0 0 0 0 0 0 3 C 30H,30H,30H,30H,30H,30H,33H,43H	0 0 30H,30H
					5 3 2 0 35H,33H,32H,30H	0 0 0 0 0 0 2 6 30H,30H,30H,30H,30H,30H,32H,36H	0 1 30H,30H

- When communicating data in binary code

(Request data)

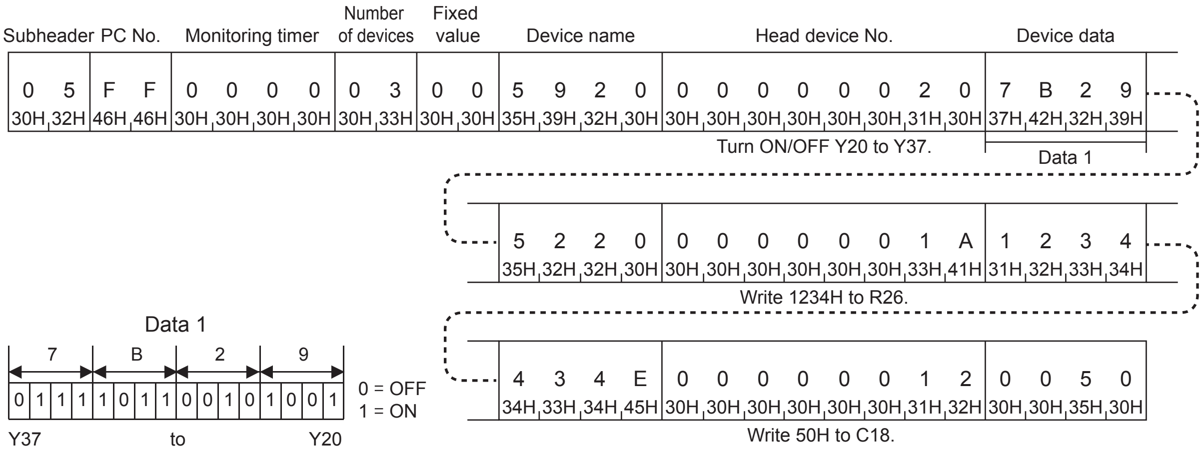


■When writing data in word units

Turn ON/OFF Y20 to Y37, and write 4660 (1234H) to R26 and 80 (50H) to C18.

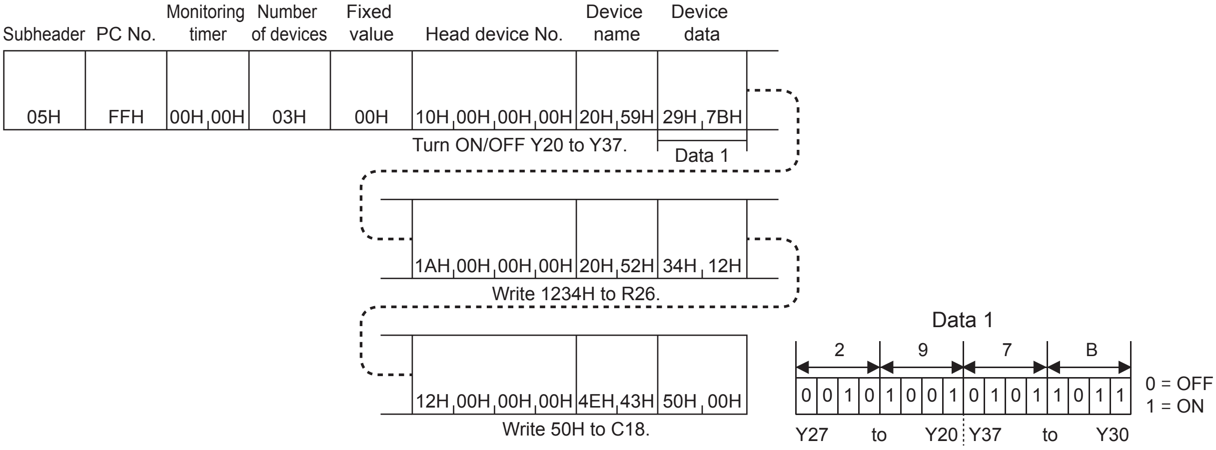
- When communicating data in ASCII code (X, Y HEX)

(Request data)



- When communicating data in binary code

(Request data)



39.3 Remote Control

This section describes the command to set the SLMP compatible device or Ethernet-equipped module to the RUN status or STOP status by a message from the external device.

Before the remote operation

When the Remote STOP operation is performed

Unless the Remote RUN operation is performed by the external device that performed the Remote STOP operation, the Remote RUN/STOP operation by other external devices is invalid.

The accessed device or module is turned off and on or reset after remote operation

The information about the remote operation will be deleted.

Ex.

When the Remote STOP is executed with the Ethernet-equipped module switch in the RUN status, if the Ethernet-equipped module power is turned off and on, the module will return to the RUN status.

A remote password (3E frame) of the CPU module (access destination) is enabled

Remote operation from the external device is not available. An error will occur at the access destination, and an abnormal response will be sent back to the external device. Unlock the remote password of the CPU module side, and resend the request message.

Remote RUN

External equipment executes the Remote RUN operation for the Ethernet-equipped module.

Point

The Remote RUN for the CPU module can be used when the switch of the accessed CPU module is set to RUN. Even if the switch is in the STOP status, Remote RUN (command: 13H) will be completed normally. However, the access destination does not change to the RUN status.

Request data

■When communicating data in ASCII code

2 bytes	2 bytes	4 bytes
1 3 31H 33H	PC No.	Monitoring timer

■When communicating data in binary code

1 byte	1 byte	2 bytes
13H	PC No.	Monitoring timer

■Subheader

Specify the command.

Function	Command		
	ASCII code (Upper column: characters, lower column: character code)		Binary code
Remote RUN	1	3	13H
	31H	33H	

■PC No.

Specify the "FFH".

■Monitoring timer

Specify the "0000H".

Response data

There is no response data for the Remote RUN command.

Remote STOP

External equipment executes the Remote STOP operation for the Ethernet-equipped module.

Request data

■When communicating data in ASCII code

2 bytes	2 bytes	4 bytes
1 4 31H 34H	PC No.	Monitoring timer

■When communicating data in binary code

1 byte	1 byte	2 bytes
14H	PC No.	Monitoring time

■Subheader

Specify the command.

Function	Command		
	ASCII code (Upper column: characters, lower column: character code)		Binary code
Remote STOP	1	4	14H
	31H	34H	

■PC No.

Specify the "FFH".

■Monitoring timer

Specify the "0000H".

Response data

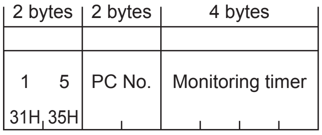
There is no response data for the Remote STOP command.

39.4 Read PC Type Name

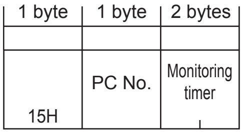
This command reads the type name code of Ethernet-equipped module.

Request data

■When communicating data in ASCII code



■When communicating data in binary code



■Subheader

Specify the command.

Function	Command		
	ASCII code (Upper column: characters, lower column: character code)		Binary code
Read PC Type Name	1	5	15H
	31H	35H	

■PC No.

Specify the "FFH".

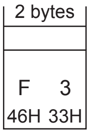
■Monitoring timer

Specify the "0000H".

Response data

The model code "F3H" is stored.

■When communicating data in ASCII code



■When communicating data in binary code

39.5 Loopback Test

This function tests whether the communication function between the external equipment and Ethernet-equipped module operates normally or not. The control procedure when this function is used is described with examples.

Point

At the startup of the Ethernet-equipped module or when trouble occurs, this function can check whether the connection between the external equipment and Ethernet-equipped module is correct and/or whether the data communication function operates normally.

Request data

■When communicating data in ASCII code

(Specified byte length × 2) + 10 bytes						
1 6 31H 36H	PC No.	Monitoring timer	Data length (1 to 254 bytes)	Head transmission data	to	End transmission data

- Data length

Convert the data length to a 2-digit ASCII code (hexadecimal), and specify in the order from the most significant byte.

- Transmission data

Each character code with a sequence of 1-byte characters ("0" to "9" and "A" to "F") is set as a 2-byte value, and up to 254 characters are sent starting from the head.

■When communicating data in binary code

1 byte	1 byte	2 bytes	Specified byte length + 5 bytes			
16H	PC No.	Monitoring timer	Data length (1 to 254 bytes)	Head transmission data	to	End transmission data

- Data length

Specify the data length by 1-byte.

- Transmission data

Data is sent for up to 254 bytes from the head by treating each character code ("0" to "9", "A" to "F") as a 1 byte value.

■Subheader

Specify the command.

Function	Command		
	ASCII code (Upper column: characters, lower column: character code)		Binary code
Loopback Test	1	6	16H
	31H	36H	

■PC No.

Specify the "FFH".

■Monitoring timer

Specify the "0000H".

Response data

The same data length and data as those sent by the external equipment are returned to the external equipment from the Ethernet-equipped module.

Communication example

"ABCDE" is sent.

■When communicating data in ASCII code

(Request data)

Subheader	PC No.	Monitoring timer	Data length	Transmission data				
1 6	F F	0 0 0 0	0 5	A	B	C	D	E
30H 32H	46H 46H	30H 30H 30H 30H	30H 35H	34H 31H	34H 32H	34H 33H	34H 34H	34H 35H

(Response data)

Data length	Transmission data				
0 5	A	B	C	D	E
30H 35H	34H 31H	34H 32H	34H 33H	34H 34H	34H 35H

■When communicating data in binary code

(Request data)

Subheader	PC No.	Monitoring timer	Data length	Transmission data
				A B C D E
16H	FFH	00H 00H	05H	41H 42H 43H 44H 45H

(Response data)

Data length	Transmission data
	A B C D E
05H	41H 42H 43H 44H 45H

PART 6 MC PROTOCOL

This part consists of the following chapters.

40 OUTLINE

This manual describes the method for reading or writing data in a CPU module with the data communication function of the external device using MC protocol (serial communication).
When transferring data using MC protocol, always refer to ☞ Page 686 MC PROTOCOL DATA COMMUNICATION.

40.1 Outline of MC Protocol

MC protocol (MELSEC communication protocol) is a protocol used for access from a CPU module or an external device (such as a personal computer or an HMI) to an MC protocol compatible device.
MC protocol communications are available among devices that can transfer messages by MC protocol.
In case of a serial port of FX5, communication is possible by 1C frame compatible with A and 3C/4C frame compatible with QnA of the MC protocol.
Device data in a CPU module can be written or read from a personal computer or an HMI by using MC protocol.
Writing and reading the device allows operation monitoring, data analyzing, and production managing of a CPU module by a personal computer or an HMI.
The following shows the flow for starting MC protocol communication.

1. Connect cables and external devices.

Configure the connection for the MC protocol communication.
For details, refer to the following.
☞ Page 314 Wiring

2. Set parameters.

Set parameters with engineering tool.

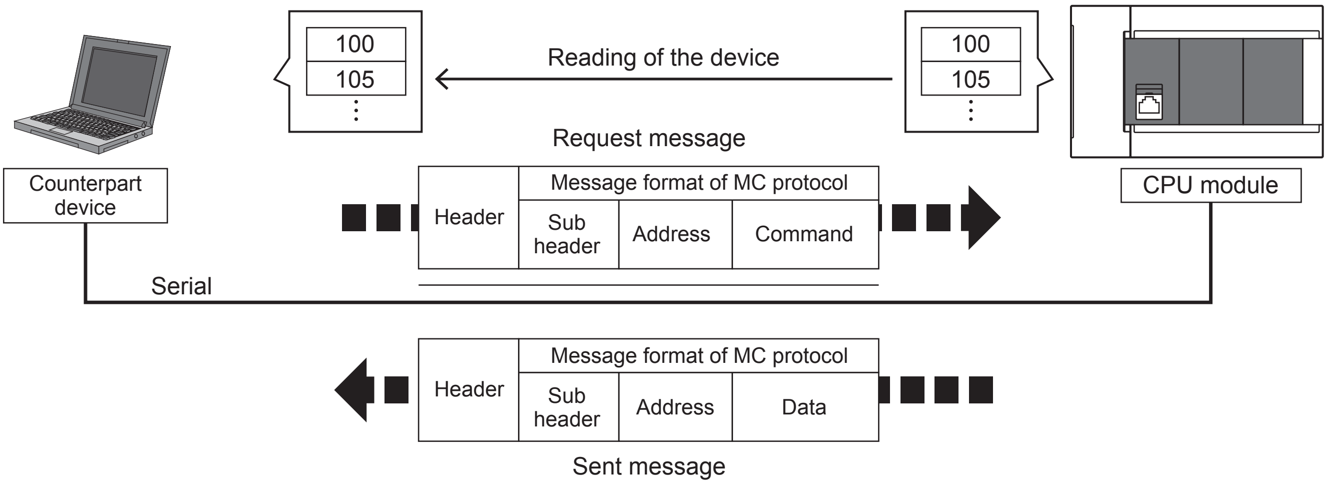
3. Write the set parameters to the CPU module.

Write set parameters to a CPU module. Validate the parameters by turning off to on or resetting the system.

40.2 Features of MC Protocol

System monitoring from an external device (such as personal computer, HMI)

An external device can send a request message in MC protocol message format to a CPU module to enable device read, allowing system monitoring.
Using MC protocol allows not only device data reading but also device data writing and resetting a CPU module.



MEMO

41 MC PROTOCOL DATA COMMUNICATION

This chapter describes the MC protocol data communication by which the external device reads or writes data to a CPU module.

41.1 Type and Application of the Data Communication Frame

This section describes the type and application of the frame (data communication message) by which the external device accesses a CPU module with MC protocol.

When the external device accesses a CPU module using serial communication, the data communication is executed by sending or receiving a request message (access request) and response message of the following frame.

Target communication method	Applicable communication frames	Features and purposes	Communication data code	Reference
serial communication	4C frame	Accessible from external devices with the maximum access range.	ASCII code or binary code	Page 692 MESSAGE FORMAT
	3C frame	These message formats are simplified compared to the 4C frame.	ASCII code	Page 692 MESSAGE FORMAT
	1C frame[*1]	These frames have the same message structures as when accessing the CPU module using an FX3 or MELSEC-A series computer link module.	ASCII code	Page 692 MESSAGE FORMAT

*1 For the FX5U/FX5UC CPU module 1C frame compatible version, refer to ☞ Page 936 Added and Changed Functions.

Point

FX5 CPU module supports 3E/1E frame (Ethernet communication) of MC protocol.

The message format of 3E/1E frame of MC protocol is the same as that of the 3E/1E frame of SLMP.

For details on 3E/1E frame of SLMP, refer to the following.

- ☞ Page 90 SLMP FUNCTION
- ☞ Page 554 OUTLINE

41.2 Concept of Control Procedure of MC Protocol

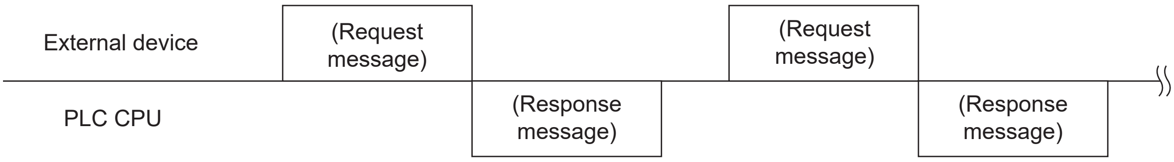
This section describes the concept of the procedure (control procedure) when the external device accesses a CPU module with MC protocol.

Sending a request message

Data communication using MC protocol communication is executed in half-duplex communication.

To access the CPU module, send the next request message after receiving a response message for the preceding request message from the CPU module.

(Until the receiving of the response message is completed, the next request message cannot be sent.)



When a response message of completion for a request message cannot be received

■When a response message of completion with an error is received

Take corrective actions depending on the error code in the response message.

■When a response message or all messages cannot be received

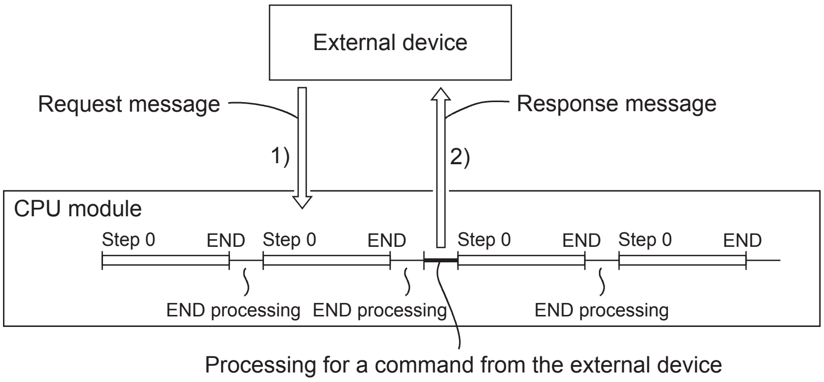
Resend a request message after the monitoring time of the response monitoring timer elapses.

Change the set value of the monitoring time as needed.

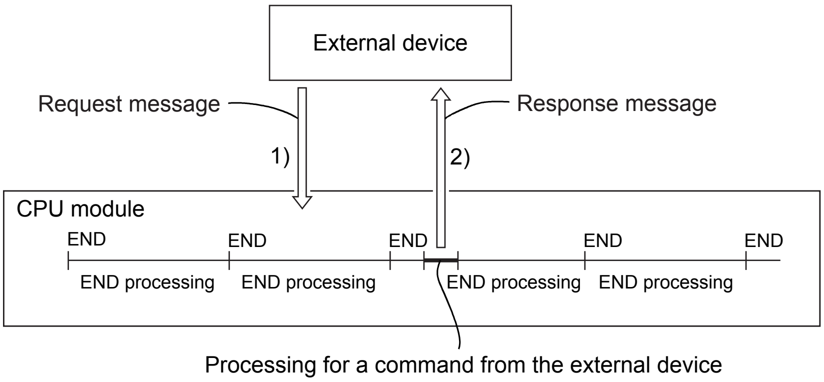
41.3 Access Timing of the CPU Module Side

The following shows the access timing of the CPU module side when the CPU module is accessed from the external device using the serial communication port.

- RUN



- STOP



1. To send a read request or a write request to the CPU module side from the external device, a request message is sent.

2. The CPU module reads or writes the data according to the description requested from the external device when the END instruction of the CPU module is executed and sends a response message including the processing result to the external device of the request source.

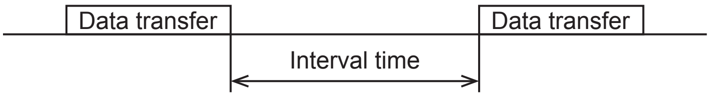
Point

- Access between the external device and CPU module is processed at each END processing when the CPU module is running for a command request. (The scan time becomes longer by the processing time of the command request.)
- When accesses are requested simultaneously to the CPU module from multiple external device, the processing requested from the external device may be on hold until several END processing is performed several times depending on the request timing.

41.4 Transfer Time

Link time

■Data transfer



■Data transfer time

R: Number of read data points, W: Number of written data points, T: Time to send or receive one character, V: Interval time, S: Max Scan Time of PLC, D: Message waiting time

(1) 1C Frame

Time to read continuous word devices (such as data registers) in one station (ms)

= (21[*1] + 4 × R[*2]) × T (ms) + V + S (SD8012) × 3 + D

Time to write continuous word devices (such as data registers) in one station (ms)

= (20[*1] + 4 × W[*2]) × T (ms) + V + S (SD8012) + D

*1 This is the number of characters when format 1 is used and the sum check is not provided. When format 4 is used, add "4" to this value. Further, when the sum check is provided, add "4" to this value also.

*2 The number of points is counted in 1-word units.

(2) 3C Frame

Time to read continuous word devices (such as data registers) in one station (ms)

= (43[*3] + 4 × R[*4]) × T (ms) + V + S (SD524) × 3 + D

Time to write continuous word devices (such as data registers) in one station (ms)

= (42[*3] + 4 × W[*4]) × T (ms) + V + S (SD524) × 3 + D

*3 This is the number of characters when format 1 is used and the sum check is not provided at the time of execution of batch read/write command.
When format 4 is used, add "4" to this value. Further, when the sum check is provided, add "4" to this value also. Further, when specifying an extension, add "+7" to this value also.

*4 The number of points is counted in 1-word units.

(3) 4C Frame: In case of ASCII code (When format 1 to format 4 are used)

Time to read continuous word devices (such as data registers) in one station (ms)

= (49[*5] + 4 × R[*6]) × T (ms) + V + S (SD524) × 3 + D

Time to write continuous word devices (such as data registers) in one station (ms)

= (48[*5] + 4 × W[*6]) × T (ms) + V + S (SD524) × 3 + D

*5 This is the number of characters when format 1 is used and the sum check is not provided at the time of execution of batch read/write command.
When format 4 is used, add "4" to this value. Further, when the sum check is provided, add "4" to this value also. Further, when specifying an extension, add "+7" to this value also.

*6 The number of points is counted in 1-word units.

(4) 4C Frame: In case of binary code (When format 5 is used)

Time to read continuous word devices (such as data registers) in one station (ms)

= (42[*7] + 4 × R[*8]) × T (ms) + V + S (SD524) × 3 + D

Time to write continuous word devices (such as data registers) in one station (ms)

= (40[*7] + 4 × W[*8]) × T (ms) + V + S (SD524) × 3 + D

*7 This is the number of characters when format 5 is used and the sum check is not provided at the time of execution of batch read/write command.
Further, when the sum check is provided, add "4" to this value also. Further, when specifying an extension, add "+7" to this value also.
When "10H" exists in the data area, since DLE "10H" is added just before "10H", add "+ "10H numeral".

*8 The number of points is counted in 1-word units.

■Time to send or receive one character

The table below shows the time required to send or receive one character when the start bit is 1-bit, the data length is 7-bit, the parity is 1-bit, and the stop bit is 1-bit (total 10-bits).

Transmission speed (baud rate) (bps)	Time to send or receive 1 character (ms)
300	33.34
600	16.67
1200	8.34
2400	4.17
4800	2.08
9600	1.04
19200	0.52
38400	0.26
57600	0.17
115200	0.08

The tables below show the data transfer times depending on the number of continuously read or written word devices at transmission speeds of 9600 bps and 19200 bps when the message waiting time is 0 ms[*1], the maximum scan time is 20 ms, and the interval time is 100 ms.

- When the transmission speed is 9600bps (Unit: Second)

Number of data points	Number of stations		
	Station No. 1	Station No. 8	Station No. 16
10 points	0.3	1.9	3.7
32 points	0.4	2.6	5.2
64 points	0.5	3.7	7.3

- When the transmission speed is 19200bps (Unit: Second)

Number of data points	Number of stations		
	Station No. 1	Station No. 8	Station No. 16
10 points	0.2	1.6	3.2
32 points	0.3	2.0	3.9
64 points	0.4	2.5	5.0

When the types of read or written devices increase, "Data transfer time shown in above table × Number of device types" is required.

When the number of read or written points exceeds "64", the transfer time increases.

Accordingly, for achieving efficient data transfer, it is recommended to decrease the number of types of transferred devices and use as many continuous device numbers as possible.

[*1] When RS-485 one-pair wiring using FX-485PC-IF is adopted, the message waiting time (for every exchange) must be 70 to 150ms. When RS-485 two-pair wiring or RS-232C is adopted, the message waiting time becomes 0ms.

41.5 CPU Module Processing Time of MC Protocol

When accessing the CPU module from an external device using MC protocol communication, the following "intervention time to the scan time" and "number of scans for processing" of the CPU module side are required. On the request from the external device using MC protocol communication, the CPU module processes a specified number of points during each END processing in case the CPU module is running.

Item	Command	Subcommand	Access points 1) / 2)	Intervention time [ms]*2 (extension of scan time)				Number of scans required for processing
				Access point 1)		Access point 2)		
				FX5S	FX5UJ/ FX5U/ FX5UC	FX5S	FX5UJ/ FX5U/ FX5UC	
Batch read	0401	0001	1/3584	0.04	0.04	1.17	1.17	1
		0000	1/960	0.04	0.04	1.15	0.96	1
Batch write	1401	0001	1/3584	0.04	0.04	1.26	1.26	1
		0000	1/960	0.05	0.05	1.21	1.16	1
Random read	0403	0000	1/192	0.06	0.06	3.33	2.60	1
Random write	1402	0001	1/188	0.04	0.04	2.86	2.08	1
		0000	1/160*1	0.05	0.04	2.90	2.28	1
Batch read multiple blocks	0406	0000	1/960	0.05	0.05	1.16	0.96	1
Batch write multiple blocks	1406	0000	1/770	0.05	0.04	1.05	0.94	1
CPU model name read	0101	0000	(one station)	0.03	0.03	—	—	1

*1 This is the processing time when accessing with only word access points specified.

*2 This is the processing time when 1 is set to [CPU Parameter] - [Service Processing Setting] - [Device/Label Access Service Processing Setting] - [Set Processing Counts] of GX Works3.

Point

- Number of scans required for processing

The CPU module processes only one command during an END processing. If GX Works3 or other modules are also accessing the CPU module simultaneously, the number of scans required for processing may increase due to the waiting time.

- Method of reducing the intervention time to the scan time

Adjust the service process execution count of the CPU module in [CPU Parameter] - [Service Processing Setting] - [Device/Label Access Service Processing Setting] to reduce the intervention time to the scan time. (For details, refer to 📖MELSEC iQ-F FX5 User's Manual (Application))

- When extension of scan time affects the control

Access multiple times with less points.

42 MESSAGE FORMAT

This chapter describes the message data format, the data specification method, and limitations etc. when performing MC protocol data communication using the 3C/4C frame to the serial communication port.

42.1 Types and Purposes of Messages

The messages of MC protocol can be classified as shown in the following table depending on the supported device and its intended purpose.

Formats and codes

There are five formats for the message that can be used for serial communication module.

○: Supported, ×: Not supported

Format	Code of communication data	Remarks	Reference	Corresponding of FX5	GX Works3 setting
Format 1	ASCII code	—	Page 693 Format 1	○	Message format: format 1 (X, Y OCT), format 1 (X, Y HEX)[*1]
Format 2	ASCII code	Format with block number appended	—	×	—
Format 3	ASCII code	Format enclosed with STX and ETX	—	×	—
Format 4	ASCII code	Format with CR and LF appended at the end	Page 694 Format 4	○	Message format: format 4 (X, Y OCT), format 4 (X, Y HEX)[*1]
Format 5	Binary code	Can be used by 4C frame.	Page 695 Format 5	○	Message format: format 5

[*1] For the version of the FX5U/FX5UC CPU module compatible with each pattern, refer to ☞ Page 936 Added and Changed Functions.

Set the format with the module parameter of GX Works3.

Point

Communication using binary code shorten the communication time since the amount of communication data is reduced by approximately half as compared to the one using ASCII code.

Frame

This section explains the types and purposes of the frames (data communication messages) used by the external device to access the supported devices using MC protocol.

The frames for MC protocol (serial communication) are as follows:

○: Supported, ×: Not supported

Frame	Features and purposes	Compatible message format	Format	Corresponding of FX5
4C frame	Accessible from external devices with the maximum access range.	Dedicated protocols for MELSEC-QnA series serial communication modules (QnA extension frame).	Formats 1 to 5	○ Corresponding to formats 1, 4, and 5.
3C frame	These message formats are simplified compared to the 4C frame. Data communication software for MELSEC-QnA series programmable controllers can be used.	Dedicated protocols for MELSEC-QnA series serial communication modules (QnA frame).	Formats 1 to 4	○ Corresponding to formats 1 and 4.
2C frame		Dedicated protocols for MELSEC-QnA series serial communication modules (QnA simplified frame).		×
1C frame	These frames have the same message structures as when accessing the CPU module using an FX3 or MELSEC-A series computer link module. Data communication software for FX3 or MELSEC-A series programmable controllers can be used.	Dedicated protocols for MELSEC-A series computer link modules		○ Corresponding to formats 1 and 4.

42.2 Message Formats of Each Format

This section explains the message format and setting data per each format.

Format 1

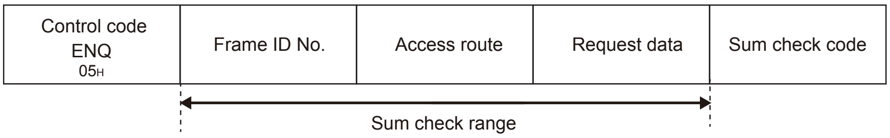
There are 2 types of formats: format 1 (X, Y OCT) and format 1 (X, Y HEX).
The specification method of the device number for the X (input) and Y (output) to be accessed is different from each other. (☞ Page 712 Device Access)

- Format 1 (X, Y OCT): octal
- Format 1 (X, Y HEX): hexadecimal

Unless otherwise specified, the both of them are described as format 1.
Format 1 (X, Y HEX) cannot be used depending on the version. For the version of the FX5U/FX5UC CPU module compatible with this format, refer to ☞ Page 936 Added and Changed Functions.

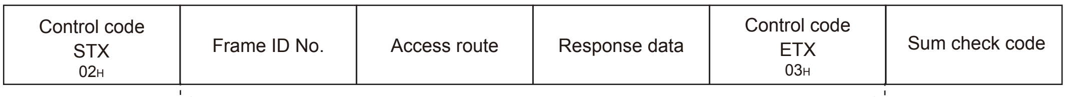
Message format

■Request message



Control code ENQ 05H	Frame ID No.	Access route	Request data	Sum check code

■Response message (Normal completion: Response data)



Control code STX 02H	Frame ID No.	Access route	Response data	Control code ETX 03H	Sum check code

■Response message (Normal completion: No response data)

Control code ACK 06H	Frame ID No.	Access route

■Response message (Abnormal completion)

Control code NAK 15H	Frame ID No.	Access route	Error code

Setting data

Set the following items.

Item	Description	Reference
Control code (ENQ, STX, ACK, NAK, ETX)	A code is defined for control.	Page 696 Control code
Frame ID No.	Specify the frame to be used.	Page 698 Frame ID No.
Access route	Specify the access route.	Page 702 Accessible Ranges and Settable Data for Each Frame
Request data	Set the command that indicates the request content. Refer to "Request data" rows of each command.	Page 709 COMMANDS
Response data	Store the read data for the command. Refer to "Response data" rows of each command.	
Sum check code	The value of the lower one byte (8 bits) of the additional result regarding the data in the sum check target range as a binary data.	Page 699 Sum check code
Error code	Error code indicates the content of occurred error.	Page 701 Error code

Format 4

There are 2 types of formats: format 4 (X, Y OCT) and format 4 (X, Y HEX).

The specification method of the device number for the X (input) and Y (output) to be accessed is different from each other. (☞ Page 712 Device Access)

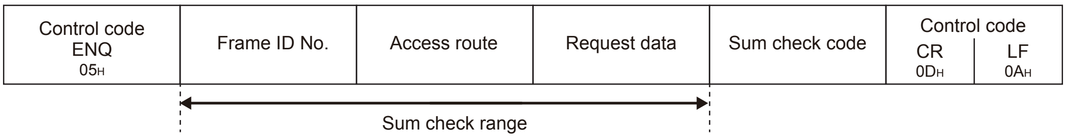
- Format 4 (X, Y OCT): octal
- Format 4 (X, Y HEX): hexadecimal

Unless otherwise specified, the both of them are described as format 4.

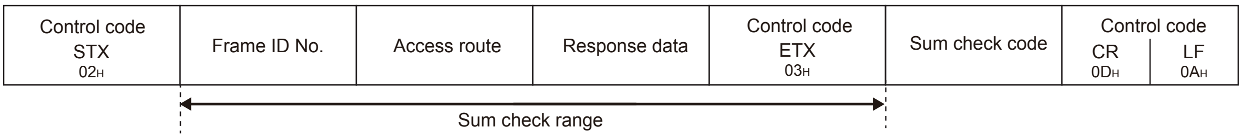
Format 4 (X, Y HEX) cannot be used depending on the version. For the version of the FX5U/FX5UC CPU module compatible with this format, refer to ☞ Page 936 Added and Changed Functions.

Message format

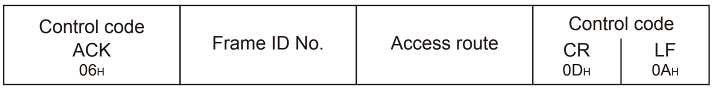
■Request message



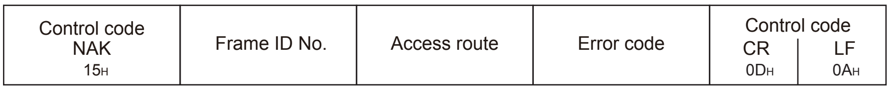
■Response message (Normal completion: Response data)



■Response message (Normal completion: No response data)



■Response message (Abnormal completion)



Setting data

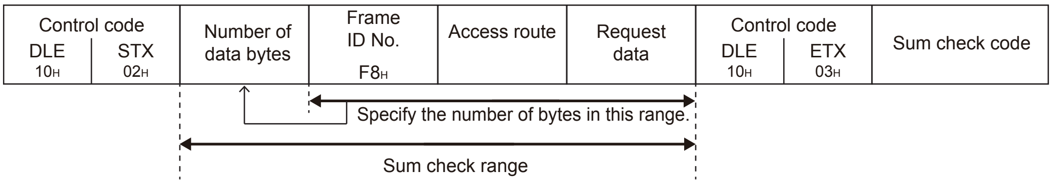
Set the following items.

Item	Description	Reference
Control code (ENQ, STX, ACK, NAK, ETX, CR, LF)	A code is defined for control.	Page 696 Control code
Frame ID No.	Specify the frame to be used.	Page 698 Frame ID No.
Access route	Specify the access route.	Page 702 Accessible Ranges and Settable Data for Each Frame
Request data	Set the command that indicates the request content. Refer to "Request data" rows of each command.	Page 709 COMMANDS
Response data	Store the read data for the command. Refer to "Response data" rows of each command.	
Sum check code	The value of the lower one byte (8 bits) of the additional result regarding the data in the sum check target range as a binary data.	Page 699 Sum check code
Error code	Error code indicates the content of occurred error.	Page 701 Error code

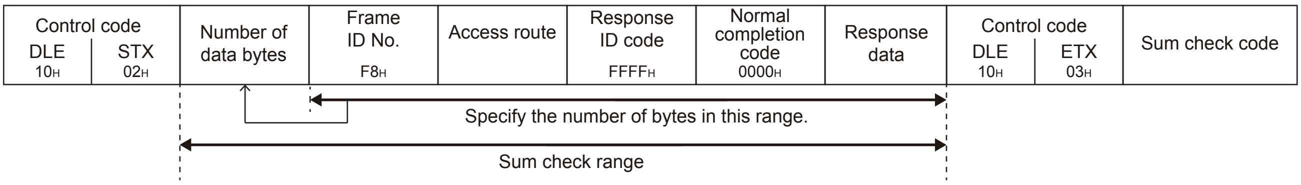
Format 5

Message format

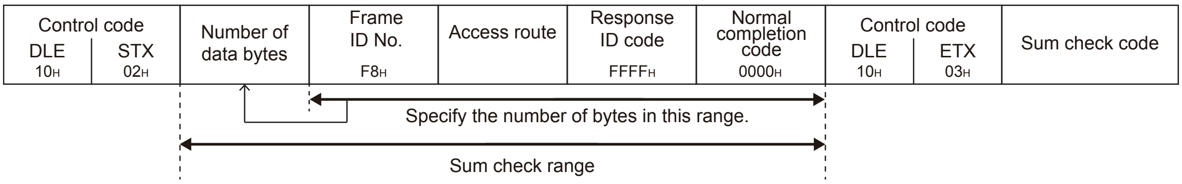
■Request message



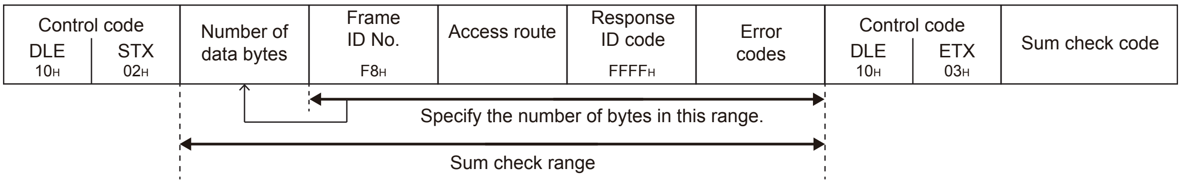
■Response message (Normal completion: Response data)



■Response message (Normal completion: No response data)



■Response message (Abnormal completion)



Setting data

Set the following items.

Item	Description	Reference
Control code (DLE, STX, ETX)	A code is defined for control.	Page 696 Control code
Number of data bytes	The number of bytes from the frame ID No. to control code (DLE, ETX).	Page 697 Number of data bytes
Frame ID No.	Specify the frame to be used.	Page 698 Frame ID No.
Access route	Specify the access route.	Page 702 Accessible Ranges and Settable Data for Each Frame
Request data	Set the command that indicates the request content. Refer to "Request data" rows of each command.	Page 709 COMMANDS
Response data	Store the read data for the command. Refer to "Response data" rows of each command.	
Sum check code	The value of the lower one byte (8 bits) of the additional result regarding the data in the sum check target range as a binary data.	Page 699 Sum check code
Response ID code	This indicates a response message. The 2-byte numerical value, 'FFFFH' is stored.	—
Normal completion code	This indicates the processing is completed normally. The 2-byte value, '0000H' is stored.	—
Error code	Error code indicates the content of occurred error.	Page 701 Error code

42.3 Details of Setting Data (Format)

This section explains how to specify the common data items and their content in each message.

Control code

Control code is a data that has special meaning (such as head data of a message) for transmission control.

Control code used in a message (format 1 to format 4) in ASCII code

The control code used for a message in ASCII code (format 1 to format 4) is shown in the following table.

Symbol name	Description	Code (hexadecimal)
STX	Start of Text	02H
ETX	End of Text	03H
EOT	End of Transmission	04H
ENQ	Enquiry	05H
ACK	Acknowledge	06H
LF	Line Feed	0AH
CL	Clear	0CH
CR	Carriage Return	0DH
NAK	Negative Acknowledge	15H

■EOT (04H), CL (0CH)

EOT and CL are codes for initializing the transmission sequence for data communications in ASCII code using the MC protocol and for placing CPU module into wait state to receive commands from an external device.

The transmission sequence is initialized with the command (command code: 1615) when binary code (format 5) is used.

When performing the following at an external device, send the EOT/CL to the CPU module depending on the format used.

- Canceling a read/write request by command previously sent. (If a write request is issued, the write request cannot be canceled when the data has already written to the CPU module.)
- Placing CPU module into the wait state to receive commands before commands are sent.
- Placing CPU module into the state where it has been started up when data communication cannot be performed normally.

The message structure when sending EOT, CL is shown below.

Only the following data is sent. The station No. and PC No. are not required.

Format	EOT	CL
Format 1	EOT 04_H	CL $0C_H$
Format 4	EOT CR LF 04_H $0D_H$ $0A_H$	CL CR LF $0C_H$ $0D_H$ $0A_H$

When CPU module receives EOT or CL, it proceeds as follows.

- Terminates any read/write processing performed upon request from the external device. In this case, CPU module does not send a response message to the command previously received.
- CPU module initializes the transmission sequence using the MC protocol and placing CPU module into wait state to receive commands from an external device.
- CPU module does not send a response message to the EOT or CL reception. (It does not send anything to external devices.)

Control code used in a message (format 5) in binary code

The control code used for a message in binary code (format 5) is shown in the table below.

Symbol name	Description	Code (hexadecimal)
STX	Start of Text	02H
ETX	End of Text	03H
DLE	Data Link Escape	10H

■Additional code (10H)

The additional code is added to distinguish the data when the control code (10H) is the same as the setting data in frame 5.

When '10H' is included in the data from "Number of data bytes" and "Request data" in the request message, the additional code '10H' is added in front of the data.

When '10H' is included in the data from "Number of data bytes" and "Response data" in the response message, the additional code '10H' is added.

('10H' is transmitted as '10H' + '10H'.)

Point

Calculate the following value except for the additional code.

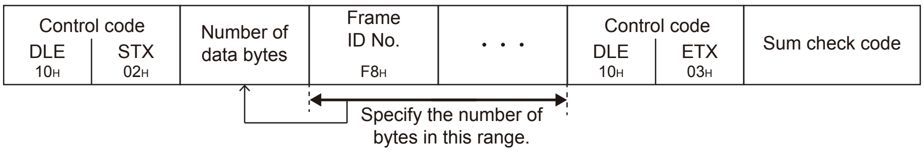
- Number of data bytes (setting item of format 5)
- Sum check code

Number of data bytes

A number of data bytes indicates the total number of bytes from the frame ID No. to control code.

Range

Calculate the data in the range from frame ID No. before DLE (10H) except for the additional code. (☞ Page 697 Additional code (10H))



Setting method

Set the data in binary code (format 5) at data communication.

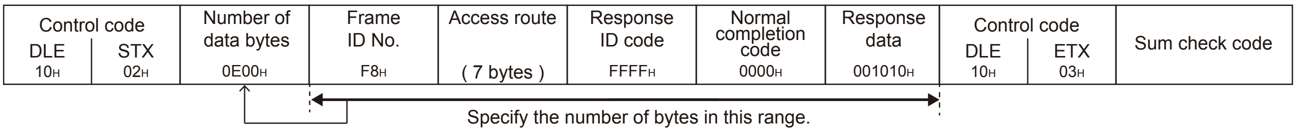
Send 2-byte numerical value from the lower byte (L: bits 0 to 7).

Ex.

Response message (Normal completion: Response data)

- Frame ID No.: 1 byte
- Access route: 7 byte
- Response ID code, normal completion code: 4 bytes
- Response data: 2 bytes + additional code (10H) 1 byte

Number of data bytes = 1 + 7 + 4 + 2 = 14 (0EH)



Block number

Block number is an arbitrary number defined by an external device and used for data defragmentation.

Block number converts data to 2-digit (hexadecimal) ASCII code within the range of '00H' to 'FFH' and sends them from the upper digits.

CPU module only checks if the block number is specified within the correct range. It does not check whether the block numbers are sent in order.

Frame ID No.

Specify the frame to be used.

Type	Setting value
4C frame	F8
3C frame	F9
1C frame	— (Not required)

Setting method

■Data communication in ASCII code

Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.

■Data communication in binary code

Send 1-byte numerical value.

Ex.

For 4C frame (F8)

ASCII code	Binary code
F 8 46_H 38_H	$F8_H$

Sum check code

Set the sum check code when performing sum check.

For sum check code, set the value to be calculated from the data with the range of sum check for error detection.

Sum check

Sum check is a function for detecting error when data changes while data transmission.

Set the sum check existence by Engineering tool.

■When sum check code is set to "Exist"

Attach a sum check code to the request message.

CPU module checks the sum check code. The sum check code is added to the response message.

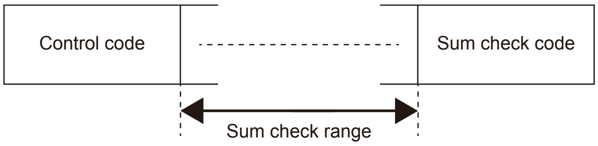
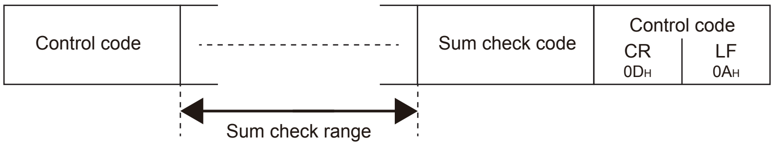
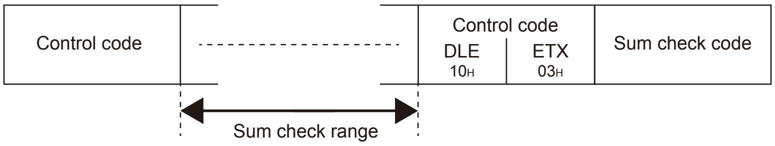
■When sum check code is set to "None"

The sum check code is not required for the request message.

CPU module does not check the sum check code. The sum check code is not added to the response message.

Sum check range

The sum check range of each message format is as follows:

Format	Message structure	Reference
Format 1		Page 693 Format 1
Format 4		Page 694 Format 4
Format 5		Page 695 Format 5

Calculation of a sum check code

For sum check code, set the numerical values of the lower 1 byte (8 bits) of the added result (sum) as binary data within the sum check range.

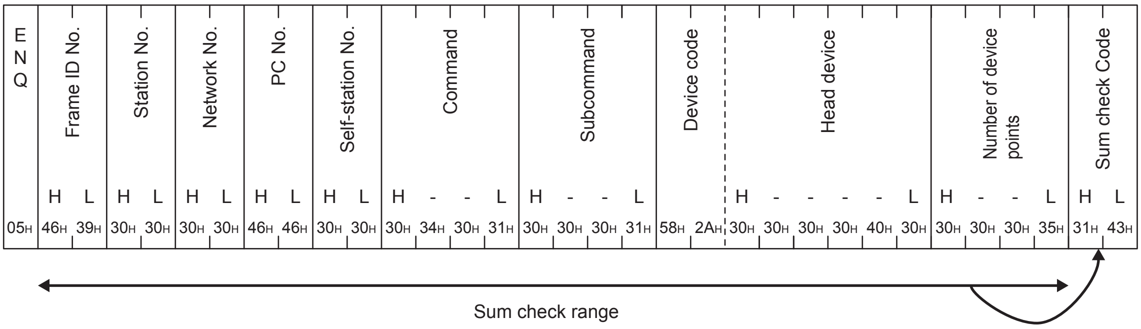
Calculate sum check code except for the additional code. (☞ Page 697 Additional code (10H))

Ex.

In the following case of 3C frame format 1, the sum check code will be '1C'.

Formula: 46H + 39H + 30H + 30H + 30H + 30H + 46H + 46H + 30H + 30H + 30H + 34H + 30H + 31H + 30H + 30H + 30H + 31H + 58H + 2AH + 30H + 30H + 30H + 30H + 34H + 30H + 30H + 30H + 30H + 35H = 61CH

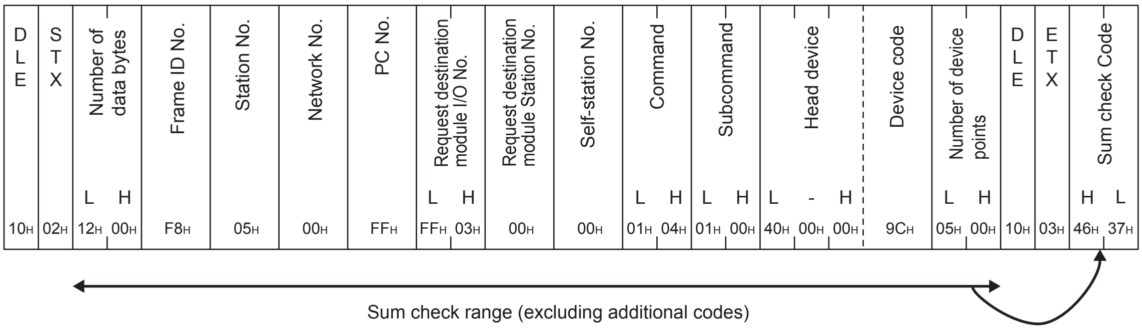
Sum check code: '1C' (ASCII code 31H, 43H)



In the following case of 4C frame format 5, the sum check code will be 'F7'.

Calculation formula: 12H + 00H + F8H + 05H + 00H + FFH + FFH + 03H + 00H + 00H + 01H + 04H + 01H + 00H + 40H + 00H + 00H + 9CH + 05H + 00H = 3F7H

Sum check code: 'F7' (ASCII code 46H, 37H)



Setting method

■Data communication in ASCII code

Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.

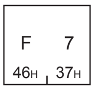
■Data communication in binary code

The same as data communication in ASCII code, use the numerical value converted to 2 digit ASCII code (hexadecimal). Send 2-byte numerical value from the upper byte (H: bits 8 to 15).

Ex.

Sum check code: 'F7' (ASCII code 46H, 37H)

ASCII code, binary code



Error code

Error code indicates the content of error.

If more than one error occurs at the same time, the error code detected first is returned.

For the content of error code and its corrective action, refer to the following.

☞ Page 836 1C frame

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

Setting method

■Data communication in ASCII code

Convert the numerical value to 4-digit ASCII code (hexadecimal) and send it from upper digits.

■Data communication in binary code

Send 2-byte numerical value from the lower byte (L: bits 0 to 7).

Ex.

When error code 7143H is returned

ASCII code	Binary code
7 1 4 3 37_H 31_H 34_H 33_H	43_H 71_H

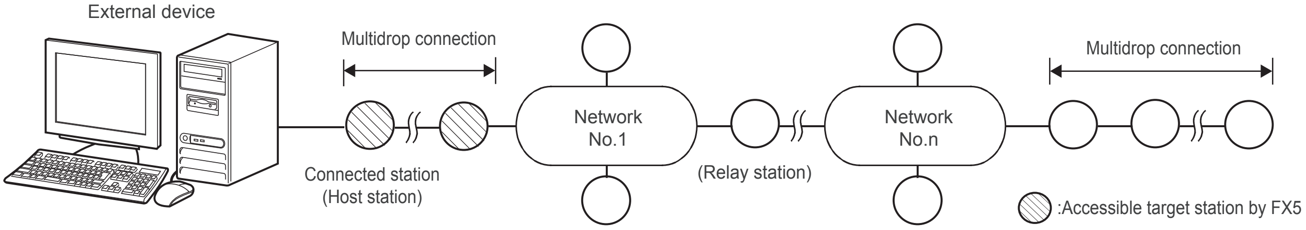
42.4 Accessible Ranges and Settable Data for Each Frame

The accessible range of each frame and the data items to set an access route are as shown below.

4C frame

Accessible range of 4C frame

The following ranges can be accessed.



Message format (Setting example for accessing connected station (host station))

■Data communication in ASCII code (Format 1, Format 4)

Station No.		Network No.		PC No.		Request destination module I/O No.				Request destination module station No.		Self-station No.	
0	0	0	0	F	F	0	3	F	F	0	0	0	0
30_H	30_H	30_H	30_H	46_H	46_H	30_H	33_H	46_H	46_H	30_H	30_H	30_H	30_H

■Data communication in binary code (Format 5)

Station No.	Network No.	PC No.	Request destination module I/O No.	Request destination module station No.	Self-station No.
00_H	00_H	FF_H	FF_H, 03_H	00_H	00_H

Data to be set

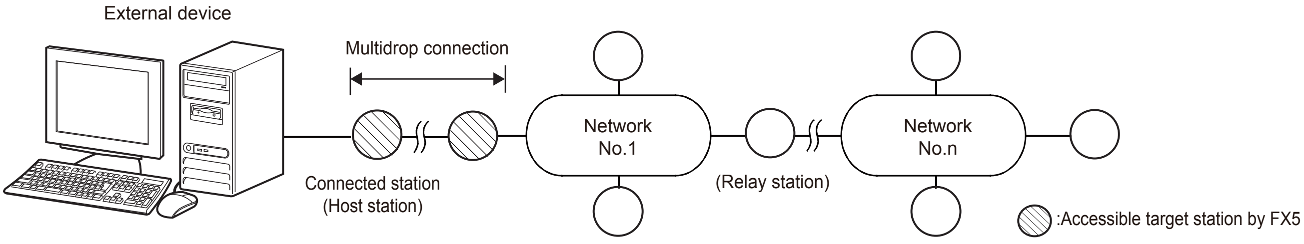
Set the following items.

Item	Description	Reference
Station No.	Specify the station to be connected from an external device.	Page 705 Station No.
Network No.	Specify the access target network No.	Page 706 Network No., PC No.
PC No.	Specify the access target PC No.	
Request destination module I/O No.	Specify the start I/O number of a connection source module (relay station) of multidrop connection via network.	Page 707 Request destination module I/O No., request destination module station No.
Request destination module station No.	Specify the station No. of an access target module of multidrop connection via network.	
Self-station No.	At the time of m:n multidrop connection, specify the station No. of a request source external device.	Page 708 Self-station No.

3C frame

Accessible range of 3C frame

The following ranges can be accessed.



Message format (Setting example for accessing connected station (host station))

Station No.		Network No.		PC No.		Self-station No.	
0	0	0	0	F	F	0	0
30_H	30_H	30_H	30_H	46_H	46_H	30_H	30_H

Data to be set

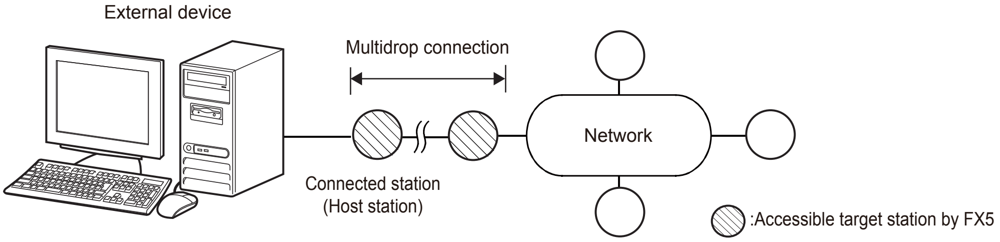
Set the following items.

Item	Description	Reference
Station No.	Specify the station to be connected from an external device.	Page 705 Station No.
Network No.	Specify the access target network No.	Page 706 Network No., PC No.
PC No.	Specify the access target PC No.	
Self-station No.	At m:n multidrop connection, specify the station No. of a request source external device.	Page 708 Self-station No.

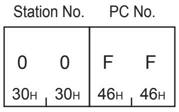
1C frame

Accessible range of 1C frame

The following ranges can be accessed.



Message format (Setting example for accessing connected station (host station))



Data to be set

Set the following items.

Item	Description	Reference
Station No.	Specify the station to be connected from an external device.	Page 705 Station No.
PC No.	Specify the access target PC No.	Page 706 Network No., PC No.

42.5 Details of Setting Data (Frame)

This section explains the content and specification method of the data items to set the access route.

○: Setting required, —: Setting not required

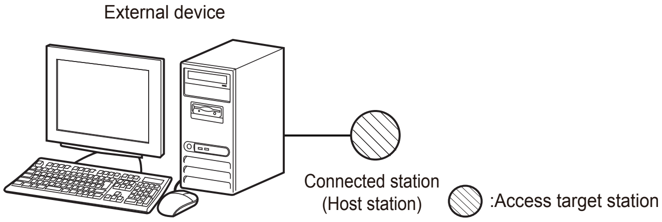
Item	4C frame	3C frame	1C frame	Reference
Station No.	○	○	○	Page 705 Station No.
Network No.	○	○	—	Page 706 Network No., PC No.
PC No.			○	
Request destination module I/O No.	○	—	—	Page 707 Request destination module I/O No., request destination module station No.
Request destination module station No.				
Self-station No.	○	○	—	Page 708 Self-station No.

Station No.

Specify the station accessed from an external device.

Accessing connected station (host station)

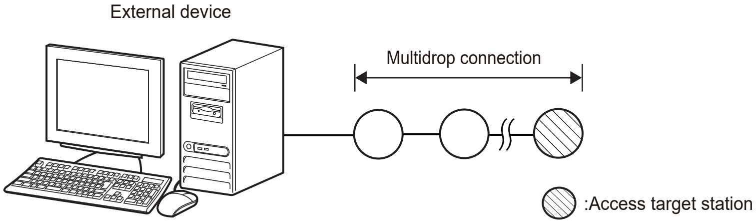
Specify '0' when accessing connected station (host station).



Accessing multidrop connection station

For the multidrop connection, specify the station No. to be accessed from 0 to 15 (00H to 0FH).

When all stations are specified as targets by the global function of 1C frame, specify FFH.



Setting method

The station No. is specified by the following parameter items of engineering tool, and writes the "module parameter" in the CPU module.

- GX Works3: "Station Number Settings" in "Module Parameter"

■Data communication in ASCII code

Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.

■Data communication in binary code

Send 1-byte numerical value.

Ex.

When the station No. setting for CPU module to be accessed is '5'

ASCII code	Binary code
0 5 30_H 35_H	05_H

Network No., PC No.

Specify the network No. and station No. that are set with the parameters for the access target network module.

Specify a fixed value when accessing the connection station.

Network No. setting is not required for 1C frame.

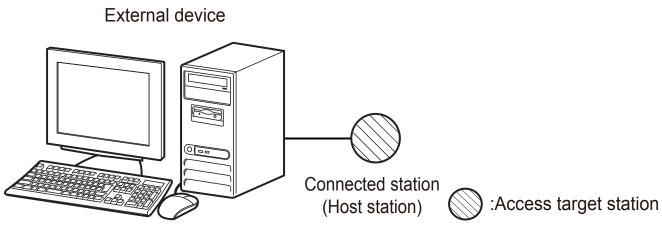
Point

Specify the network No. with the value shown below.

Specifying improper value may result in no response returned.

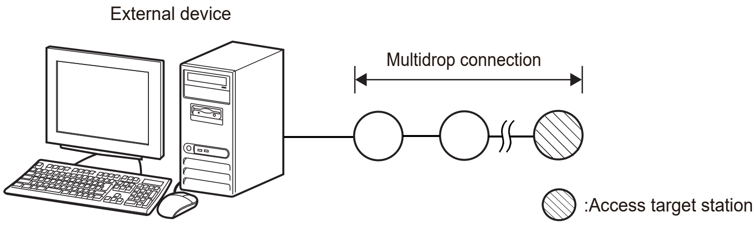
Accessing connected station (host station)

Specify '0' for the network No., and 'FF' for the PC No.



Accessing multidrop connection station

Specify '0' for the network No., and 'FF' for the PC No.



Setting method

■Data communication in ASCII code

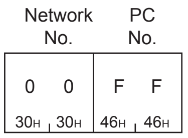
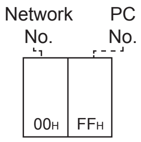
Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.

■Data communication in binary code

Send 1-byte numerical value.

Ex.

Accessing connected station (host station) or multidrop connection station

ASCII code	Binary code
	

Request destination module I/O No., request destination module station No.

The following fixed value is specified in FX5 CPU module.

Request destination module I/O No.	Request destination module station No.
03FFH	00H

Setting method

■Data communication in ASCII code

For the request destination module I/O No., convert the numerical value to 4-digit ASCII code (hexadecimal) and send it from upper digits.

For the request destination module station No., convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.

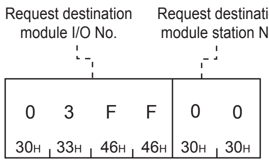
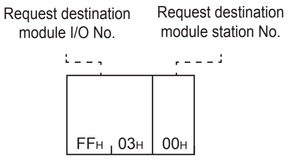
■Data communication in binary code

For the request destination module I/O No., the 2-byte value is sent from the lower byte (L: bit 0 to 7).

For the request destination module station No., the 1-byte value is sent.

Ex.

Accessing connected station (host station)

ASCII code	Binary code
	

Self-station No.

The following fixed value is specified in FX5 CPU module.

Self-station No.
00H

Setting method

■Data communication in ASCII code

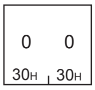
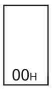
Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.

■Data communication in binary code

Send 1-byte numerical value.

Ex.

When 00H is specified

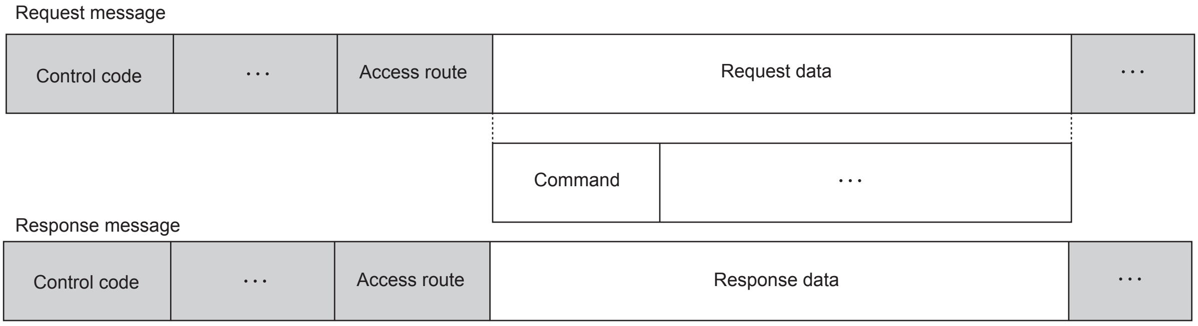
ASCII code	Binary code
	

43 COMMANDS

This chapter explains commands of MC protocol.

43.1 List of Commands and Functions

The functions of a message is defined by each command. The message format for request data and response data varies with commands. Depending on the type of frame to be used, the specific value is assigned to a command. The value of command is specified at the head of a request data.



The explanation of each command in Part 3, the message format of request data and response data are explained.
For the message formats other than request data and response data, refer to the following sections.

☞ Page 692 MESSAGE FORMAT

Command list

1C frame

For the commands for 1C frame, refer to ☞ Page 774 Command and Function Lists for 1C Frame.

3C/4C frame

Name	Command	Subcommand	Contents of processing	Number of points processed in one-time update
Batch read	0401H	0001H	Reads data in 1-point units from bit devices or word devices.	ASCII: 3584 points BIN: 3584 points
		0000H	• Reads data in 16-point units from bit devices. • Reads data in 1-word unit from word devices.	960 words (15360 points)
		0081H	• Reads data in 1-bit unit from buffer memory of intelligent function module. • Reads data in 1-bit unit from a device indirectly specified in the index register.	ASCII: 3584 points BIN: 3584 points
		0080H	• Reads data in 1-word unit from buffer memory of intelligent function module. • Reads data in 1-word unit from a device indirectly specified in the index register.	960 words (15360 points)
		0083H	• Reads data in 1-bit unit from buffer memory of intelligent function module. • Reads data in 1-bit unit from a device indirectly specified in the index register.	ASCII: 3584 points BIN: 3584 points
		0082H	• Reads data in 1-word unit from buffer memory of intelligent function module. • Reads data in 1-word unit from a device indirectly specified in the index register.	960 words (15360 points)
Batch write	1401H	0001H	Writes data in 1-bit units to bit devices.	ASCII: 3584 points BIN: 3584 points
		0000H	• Writes data in 16-bit units to bit devices. • Writes data in 1-word units to bit devices.	960 words (15360 points)
		0081H	• Writes data in 1-bit unit to MC protocol compatible devices or buffer memory of intelligent function module. • Indirectly specifies bit devices, word devices and buffer memory in the index register.	ASCII: 3584 points BIN: 3584 points
		0080H	Writes data in 1 word (16-bits) unit to MC protocol compatible devices or buffer memory of intelligent function module.	960 words (15360 points)
		0083H	Writes data in 1-bit unit to MC protocol compatible devices or buffer memory of intelligent function module.	ASCII: 3584 points BIN: 3584 points
		0082H	Writes data in 1 word (16-bits) unit to MC protocol compatible devices or buffer memory of intelligent function module.	960 words (15360 points)
Random read	0403H	0000H	Reads a word device in 1-word unit or 2-word unit by randomly specifying the device number.	192 points
		0080H	Reads data in 1-word (16-bit) unit from MC protocol compatible devices or buffer memory of intelligent function module.	192 points
		0082H	Reads data in 1-word (16-bit) unit from MC protocol compatible devices or buffer memory of intelligent function module.	192 points

Name	Command	Subcommand	Contents of processing	Number of points processed in one-time update
Random write	1402H	0001H	Writes data in 1-bit unit to a bit device by randomly specifying the device number.	188 points
		0000H	• Writes data in 16-bit unit to a bit device by randomly specifying the device number. • Writes data in 1-word unit or 2-word unit to a word device by randomly specifying the device number.	(Number of word access points) × 12 + (number of double word access points) × 14 ≤ 1920
		0081H	• Writes data in 1-bit unit to MC protocol compatible devices or buffer memory of intelligent function module. • Indirectly specifies a buffer memory in the index register.	188 points
		0080H	Writes data in 1-word (16-bits) or 2-word unit to MC protocol compatible devices or buffer memory of intelligent function module.	(Number of word access points) × 12 + (number of double word access points) × 14 ≤ 1920[*1]
		0083H	Writes data in 1-bit unit to MC protocol compatible devices or buffer memory of intelligent function module.	188 points
		0082H	Writes data in 1-word (16-bits) or 2-word unit to MC protocol compatible devices or buffer memory of intelligent function module.	(Number of word access points) × 12 + (number of double word access points) × 14 ≤ 1920[*1]
Batch read multiple blocks	0406H	0000H	Assumes an n point part of a bit device or word device as 1-block and reads data by randomly specifying the multiple blocks. (When bit devices are specified, 1 point is 16 bits.)	960 points
		0080H	Assumes an n point part of MC protocol compatible devices or buffer memory of intelligent function module as 1-block and reads data by randomly specifying the multiple blocks. (When bit devices are specified, 1 point is 16 bits.)	960 points
		0082H	Assumes an n point part of MC protocol compatible devices or buffer memory of intelligent function module as 1-block and reads data by randomly specifying the multiple blocks.	960 points
Batch write multiple blocks	1406H	0000H	Assumes an n point part of a bit device or word device as 1-block and writes data by randomly specifying the multiple blocks. (When bit devices are specified, 1 point is 16 bits.)	760 points
		0080H	Assumes an n point part of MC protocol compatible devices or buffer memory of intelligent function module as 1-block and writes data by randomly specifying multiple blocks. (When bit devices are specified, 1 point is 16 bits.)	760 points[*1]
		0082H	Assumes an n point part of MC protocol compatible devices or buffer memory of intelligent function module as 1-block and writes data by randomly specifying multiple blocks.	760 points[*1]
Remote RUN	1001H	0000H	Requests remote RUN to a device.	—
Remote STOP	1002H	0000H	Requests remote STOP to a device.	—
Remote PAUSE	1003H	0000H	Requests remote PAUSE to a device.	—
Remote latch clear	1005H	0000H	Requests remote latch clear when a device is in STOP mode.	—
Remote RESET	1006H	0000H	Requests remote reset to cancel error stop mode of a device.	—
Read CPU model name	0101H	0000H	Reads a processor module name code (processor type) of a device.	—
Loopback test	0619H	0000H	Checks if normal communication is possible.	—
LED OFF, error code initialization	1617H	0000H	Clears all errors in batches and turns OFF LED.	—

[*1] With device extension specification, the number of points that can be set become fewer. When using the device extension specification, calculate doubling the number of access points.

43.2 Device Access

This section explains the control procedure specification method and shows a specification example when the device memory is read and written.

Data to be specified in commands

This section explains the contents and specification methods for data items which are set in each command related to device access.

Subcommand

Subcommands are data for specifying the unit for reading and writing, device type to be specified, and the data reading condition.

The following table shows the details of setting items.

Setting item		Description
Data size specification	Word units	• The target data is read or written in word units. • Select "0" even when the reading data or writing data does not exist in arguments of the command.
	Bit units	The target data is read or written in bit units.
Device specification format	2 digit code/6 digit number specification	Data or items related to the address specifications are expressed in the following sizes, which are the same as the existing setting. • Device code: 2-digit ASCII code, 1 byte in binary • Device number: 6-digit ASCII code, 3 bytes in binary
	4 digit code/8 digit number specification	Data or items related to the address specifications are extended to the following size. • Device code: 4-digit ASCII code, 2 bytes in binary • Device number: 8-digit ASCII code, 4 bytes in binary
Device memory extension specification	Not specified	Set this when specifying devices of a CPU module. * Set this when not using the device memory extension specification.
	Specified	• Set this for the buffer memory specification of the intelligent function module. • This setting corresponds to the buffer memory indirect specification with index register.

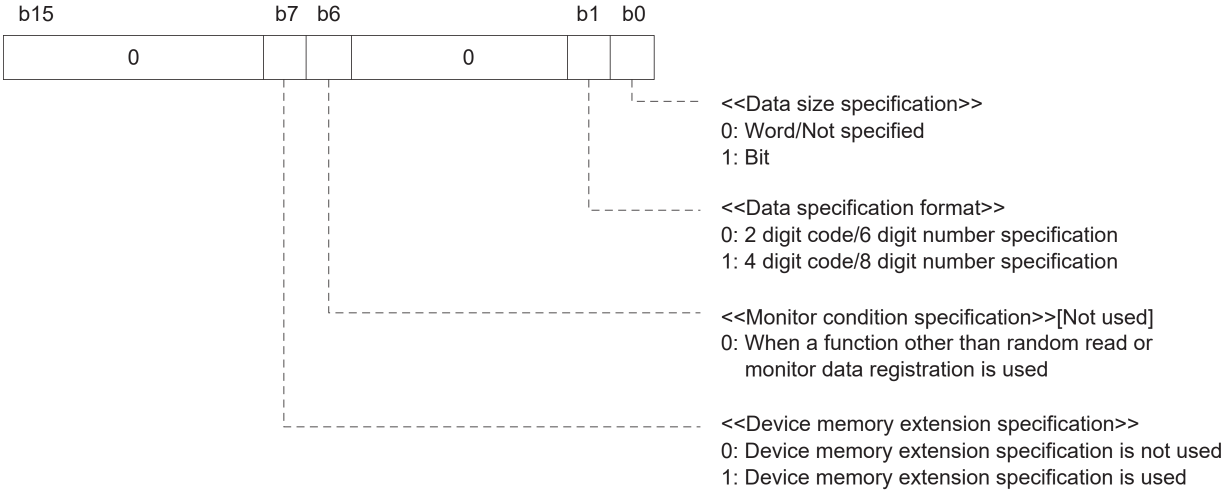
❶When communicating data in ASCII code

The value 0000H (0), or the following value (❸), is converted to a 4-digit (hexadecimal) ASCII code and sequentially transmitted beginning from the most significant digit ("0").

❷When communicating data in binary code

The value 0000H, or the following 2-byte value (❸), is used for transmission.

❸The following figure shows the specification contents of the subcommand.



<<Data size specification>>
0: Word/Not specified
1: Bit

<<Data specification format>>
0: 2 digit code/6 digit number specification
1: 4 digit code/8 digit number specification

<<Monitor condition specification>>[Not used]
0: When a function other than random read or
monitor data registration is used

<<Device memory extension specification>>
0: Device memory extension specification is not used
1: Device memory extension specification is used

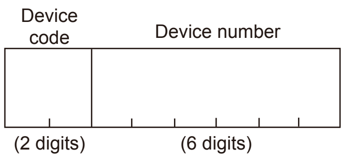
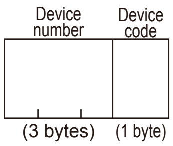
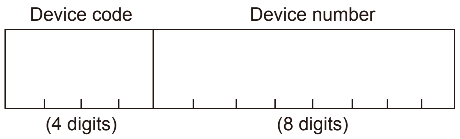
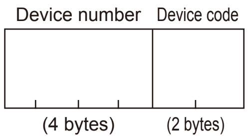
❹In the following cases, the subcommand is 0000H or 0001H.

- When neither monitor condition nor device memory extension is specified.
- When using a command that cannot select monitor condition specification and device memory extension specification.

Devices

Specify the device to be accessed by device code and device number.

- The data order differs between ASCII code and binary code.
- The data size to set up changes with setting of the device specification format of subcommand.

Device specification format of subcommand	ASCII code	Binary code
2 digit code/6 digit number specification		
4 digit code/8 digit number specification		

Point

When accessing any of the following devices, use the device extension specification (subcommand: 008□).

- Module access device

For the message format for device extension specification, refer to the following section.

☞ Page 924 Device Memory Extension Specification

Device codes

Specify the device name to be accessed.

Specify the device within the range of the access target module.

For the values of each device code, refer to the following section.

☞ Page 715 Device code list

■Data communication using ASCII code

Convert the numerical value to 2-digit or 4-digit ASCII code (hexadecimal), and send it from the upper digits.

- For 2 digit code/6 digit number specification: 2-digit ASCII code
- For 4 digit code/8 digit number specification: 4-digit ASCII code

The '*' in a device code can also be specified with a space (code: 20H).

■Data communication in binary code

Send the 1-byte or 2-byte numerical value from the lower byte (L: bits 0 to 7).

- For 2 digit code/6 digit number specification: 1 byte
- For 4 digit code/8 digit number specification: 2 bytes

Ex.

For input (X)

Device specification format of subcommand	ASCII code	Binary code
2 digit code/6 digit number specification	X * 58_H $2A_H$	$9C_H$
4 digit code/8 digit number specification	X * * * 58_H $2A_H$ $2A_H$ $2A_H$	$9C_H$ 00_H

Device number

Specify the number of the device to be accessed.

Specify the device number within the range of the access target module.

■Data communication using ASCII code

Convert the numerical value to 6-digit or 8-digit ASCII code, and sent it from the upper digits.

Specify the device number in octal, decimal, or hexadecimal, depending on the device type. (☞ Page 715 Device code list)

- For 2 digit code/6 digit number specification: 6-digit ASCII code
- For 4 digit code/8 digit number specification: 8-digit ASCII code (10 digits at device extension specification)

The '0' in the upper digits can also be specified with a space (code: 20H).

■Data communication in binary code

Send the 3-byte or 4-byte numerical value in order from the lower byte (L: bit 0 to 7).

For a device of which device number is in decimal, convert it to hexadecimal and specify.

- For 2 digit code/6 digit number specification: 3 bytes[*1]
- For 4 digit code/8 digit number specification: 4 bytes[*1]

*1 The additional code may be added. (☞ Page 697 Additional code (10H))

Ex.

For link relay (B) 1234 (a device of which device number is in hexadecimal)

Device specification format of subcommand	ASCII code	Binary code
2 digit code/6 digit number specification	0 0 1 2 3 4 30H 30H 31H 32H 33H 34H	34H 12H 00H
4 digit code/8 digit number specification	0 0 0 0 1 2 3 4 30H 30H 30H 30H 31H 32H 33H 34H	34H 12H 00H 00H

For internal relay (M) 1234 (a device of which device number is in decimal)

For binary code, convert the device number to hexadecimal. '1234' (decimal) → '4D2' (hexadecimal)

Device specification format of subcommand	ASCII code	Binary code
2 digit code/6 digit number specification	0 0 1 2 3 4 30H 30H 31H 32H 33H 34H	D2H 04H 00H
4 digit code/8 digit number specification	0 0 0 0 1 2 3 4 30H 30H 30H 30H 31H 32H 33H 34H	D2H 04H 00H 00H

For internal relay (M) 16 (with additional code)

For CPU module binary code, specify '10H' as '10H + 10H'. (☞ Page 697 Additional code (10H))

Device specification format of subcommand	Binary code
2 digit code/6 digit number specification	DLE 10H 10H 00H 00H
4 digit code/8 digit number specification	DLE 10H 10H 00H 00H 00H

Device code list

The table below shows devices and device number range that can handled in commands used in communication by MC protocol.

Specify devices and device number range that are there in the targeted unit for performing data reading, writing etc.

1C frame

For device code used in 1C frame, refer to ☞ Page 775 Data to be specified in command.

3C/4C frame

In 3C/4C frame, in the following "Device Code", specify a device of the access point.

Division	Device		Type	Device Code*1 (Device specification format: Long)		Device No.		FX5 device available*2
				ASCII code	Binary code			
Internal user devices	Input		Bit	X* (X***)	9CH (9C00H)	Specify within the device No. range of the module for access destination.	*3	○
	Output			Y* (Y***)	9DH (9D00H)		*3	○
	Internal relay			M* (M***)	90H (9000H)		Decimal	○
	Latch 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		TC (TC**)	C0H (C000H)			○
		Current value	Word	TN (TN**)	C2H (C200H)			○
	Long timer	Contact	Bit	— (LTS*)	51H (5100H)		Decimal	—
		Coil		— (LTC*)	50H (5000H)			—
		Current value	Double word	— (LTN*)	52H (5200H)			—
	Retentive timer	Contact	Bit	SS (STS*)	C7H (C700H)		Decimal	○
		Coil		SC (STC*)	C6H (C600H)			○
		Current value	Word	SN (STN*)	C8H (C800H)			○
	Long retentive timer	Contact	Bit	— (LSTS)	59H (5900H)		Decimal	—
		Coil		— (LSTC)	58H (5800H)			—
		Current value	Double word	— (LSTN)	5AH (5A00H)			—
	Counter	Contact	Bit	CS (CS**)	C4H (C400H)		Decimal	○
		Coil		CC (CC**)	C3H (C300H)			○
		Current value	Word	CN (CN**)	C5H (C500H)			○
	Long counter	Contact	Bit	— (LCS*)	55H (5500H)		Decimal	○
		Coil		— (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)		Hexadecimal	○
System device	Special relay		Bit	SM (SM**)	91H (9100H)		Decimal	○
	Special register		Word	SD (SD**)	A9H (A900H)		Decimal	○
	Command input		Bit	—	—	—	Hexadecimal	—
	Command output			—	—		Hexadecimal	—
	Function register		Word	—	—		Decimal	—
Index register			16 bit	Z* (Z***)	CCH (CC00H)	Specify within the device No. range of the module for access destination.	Decimal	○
			32 bit	LZ (LZ**)	62H (6200H)		Decimal	○
File register			Word	R* (R***)	AFH (AF00H)		Decimal	○
				ZR (ZR**)	B0H (B000H)		Decimal	—
Module access device*4	Link register		Word	W* (W***)	B4H (B400H)		Hexadecimal	—
	Link special register			SW (SW**)	B5H (B500H)		Hexadecimal	—
	Module access device			G* (G***)	ABH (AB00H)		Decimal	○

*1 [ASCII code]
When a device code is less than the specified number of characters, add "*" (ASCII code: 2AH), or <space> (ASCII code: 20H) at the end of the device code.
[Binary code]
When a device code is less than the specified size, add "00H" at the end of the device code.

*2 ○: FX5 device
—: No FX5 device

*3 Depends on the message format. See below.
Format 1 (X, Y OCT), format 4 (X, Y OCT): octal
Format 1 (X, Y HEX), format 4 (X, Y HEX), format 5: hexadecimal

*4 It is necessary to make "Device memory extension specification" of the subcommand to ON (1).

Number of device points

Specify the number of device points to be read or written.

Setting method

■Data communication using ASCII code

Convert the numerical value to 4-digit ASCII code (hexadecimal), and send it from the upper digits.

Use capitalized code for alphabetical letter.

■Data communication using binary code

Send the 2-byte numerical value[*1] in order from the lower byte (L: bit 0 to 7).

*1 The additional code may be added. (☞ Page 697 Additional code (10H))

Ex.

In the case of 5 points and 20 points

Number of device points	ASCII code	Binary code
5 points	0 0 0 5 30H 30H 30H 35H	05H 00H
20 points	0 0 1 4 30H 30H 31H 34H	14H 00H

Access points

Specify the number of device points to be accessed in word unit, double word unit, or bit unit.

Specify it within the number of points that can be processed in one communication. The number of points is shown in the table of the command list (☞ Page 710 Command list).

Setting method

■Data communication using ASCII code

Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.

Use capitalized code for alphabetical letter.

■Data communication using binary code

Send the 1-byte[1] numerical value (hexadecimal).

[1] The additional code may be added. (☞ Page 697 Additional code (10H))

Ex.

In the case of 5 points and 20 points

Number of device points	ASCII code	Binary code
5 points	0 5 30_H 35_H	05_H
20 points	1 4 31_H 34_H	14_H

Number of bit access points

Specify the number of device points to be accessed in bit units.

Number of word access points, number of double word access points

Specify the number of device points to be accessed in word unit or double word unit.

Number of blocks

Specify the number of blocks of the device to be accessed in hexadecimal.

Set each number of blocks within the following range.

- Number of word device blocks + Number of bit device blocks ≤ 120

Point

In the following case, calculate it as number of blocks × 2.

- When accessing by setting device extension specification (subcommand: 008□)

Setting method

■Data communication using ASCII code

Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.

Use capitalized code for alphabetical letter.

■Data communication using binary code

Send the 1-byte[1] numerical value (hexadecimal).

[1] The additional code may be added. (☞ Page 697 Additional code (10H))

Ex.

In the case of 5 points and 20 points

Number of device points	ASCII code	Binary code
5 points	0 5 30_H 35_H	05_H
20 points	1 4 31_H 34_H	14_H

Number of word device blocks

Specify the number of blocks of the word device.

Number of bit device blocks

Specify the number of blocks of the bit device.

Read data, write data

The read device value is stored for reading. The data to be written is stored for writing.

The data order differs between bit units or word units.

For bit units

The following shows the data to be read and written in bit units.

■Data communication using ASCII code

The ON/OFF status of each device are represented with single-digit ASCII code.

- For ON: '1' (31H)
- For OFF: '0' (30H)

■Data communication in binary code

Represent the ON/OFF status of each device in 4-bit per 1 point.

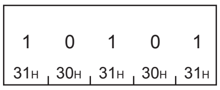
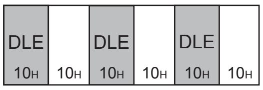
- For ON: '1'
- For OFF: '0'

When the number of points is odd, the lowest 4 bits are set to '0'.

Ex.

When indicating ON/OFF status of five points from M10

M10	M11	M12	M13	M14
ON	OFF	ON	OFF	ON

ASCII code	Binary code*1
	

*1 The additional code may be added. (☞ Page 697 Additional code (10H))

For word units (16-point unit for bit device)

The following shows the data to be read and written in word units.

When handling data other than bit data, refer to the following section.

☞ Page 725 Considerations for handling real number data and character string data

■Data communication using ASCII code

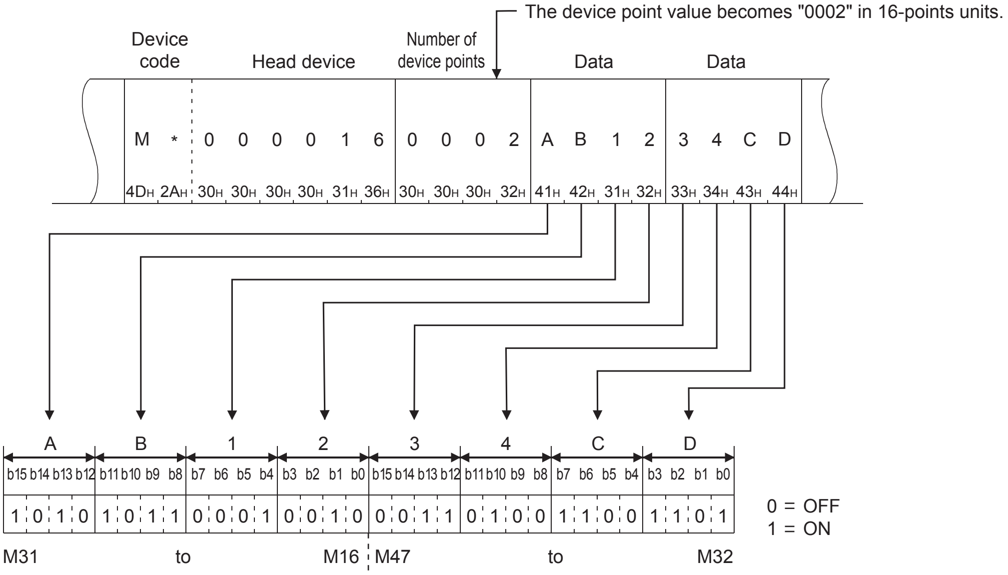
Convert the 1-word (16 points of bit device) numerical value to 4-digit ASCII code (hexadecimal), and send it from the upper digits.

Use capitalized code for alphabetical letter.

The ON/OFF status of bit device is a value of hexadecimal 1-digit in 4-point units.

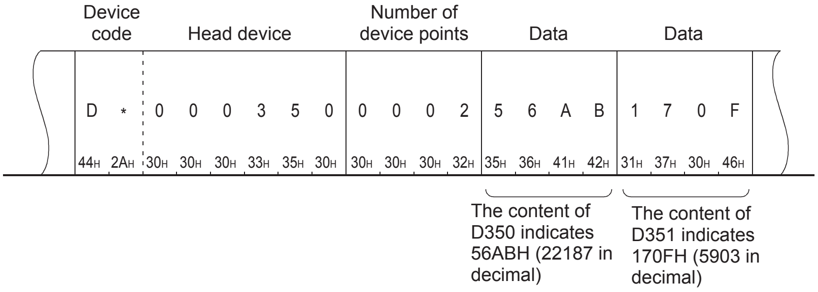
Ex.

When indicating ON/OFF status of 32 points from M16



Ex.

When indicating the stored data of D350 and D351

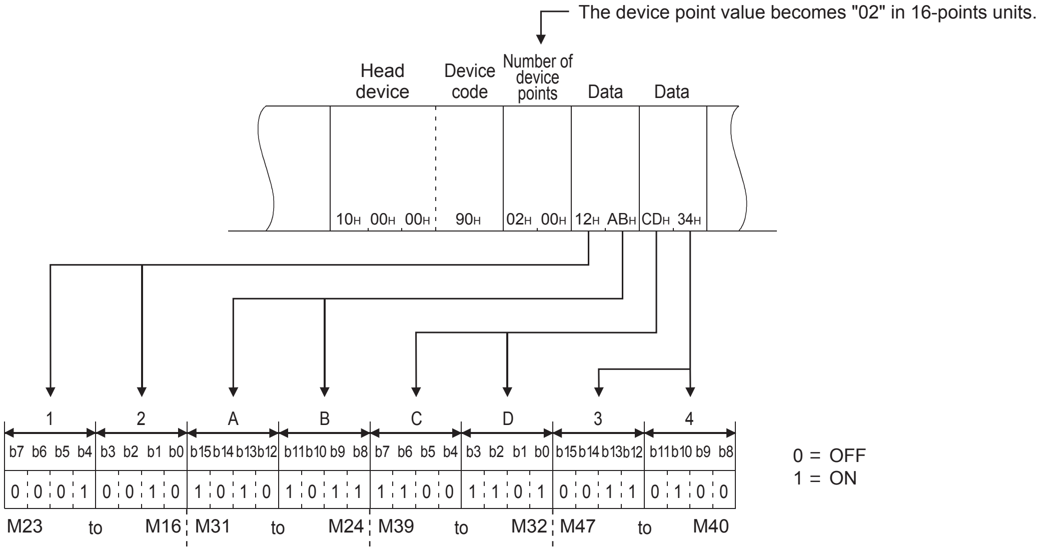


■Data communication using binary code

Send the numerical value in order from the lower byte (L: bit 0 to 7) by handling 16 points unit as 2 bytes.

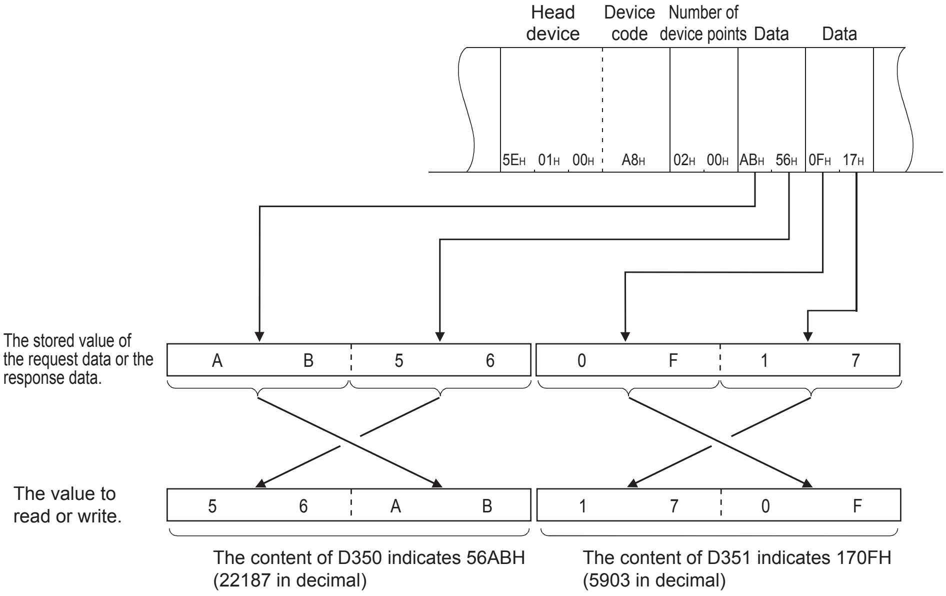
Ex.

When indicating ON/OFF status of 32 points from M16



Ex.

When indicating the stored data of D350 and D351



For double word unit (32-point unit for bit device)

The following shows the data to be read and written in double word units.

■Data communication using ASCII code

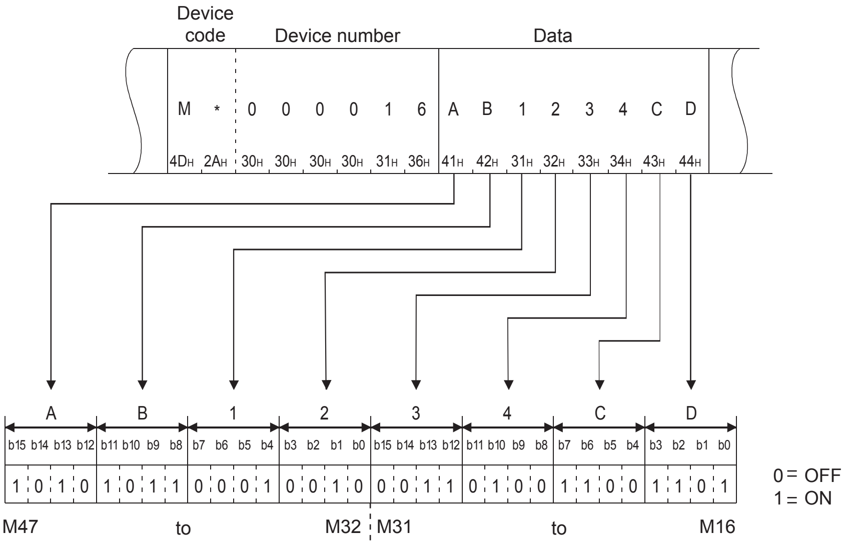
Convert the 2-word numerical value (32 points of bit device) to 8-digit ASCII code (hexadecimal), and send it from the upper digits.

Use capitalized code for alphabetical letter.

The ON/OFF status of the bit device is 1-digit hexadecimal value in 4-point units.

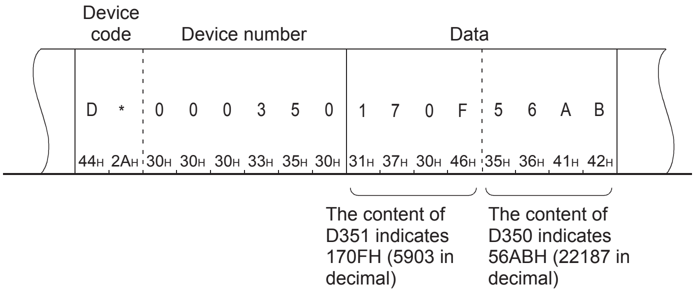
Ex.

When indicating ON/OFF status of 32 points from M16



Ex.

When indicating the stored data of D350 (D351)



The content of D351 indicates 170FH (5903 in decimal)

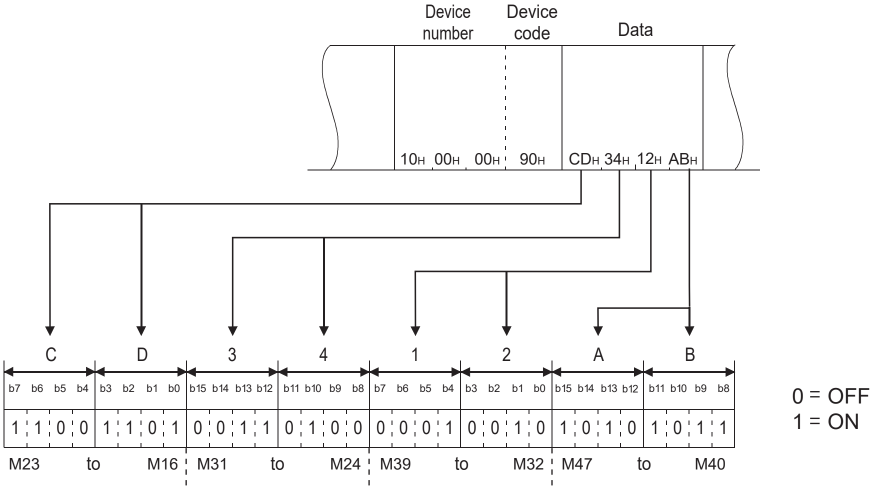
The content of D350 indicates 56ABH (22187 in decimal)

■Data communication using binary code

Send the numerical value in order from the lower byte (L: bit 0 to 7) by handling 32 points unit as 4 bytes.

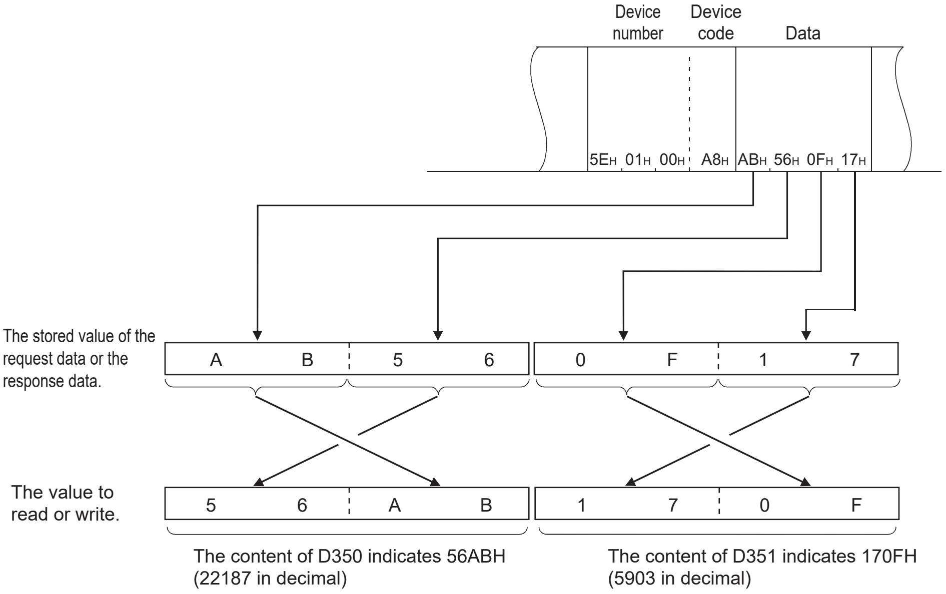
Ex.

When indicating ON/OFF status of 32 points from M16



Ex.

When indicating the stored data of D350 (D351)



Considerations for handling real number data and character string data

The word data and double word data are handled as integer value (16-bit data or 32-bit data).

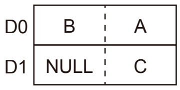
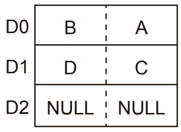
When data other than integer (real number, character string) is stored in a device, the stored value is read as integer value.

- When real number (0.75) is stored in D0 and D1: D0 = 0000H, D1 = 3F40H
- When character string ('12AB') is stored in D2 and D3: D2 = 3231H, D3 = 4241H

For data to be used as real number or character string data in the instructions of the programmable controller, write it to the device/label according to the defined data specification method. For more details on how to specify data used in instructions, refer to the 📖MELSEC iQ-F FX5 Programming Manual (Instructions, Standard Functions/Function Blocks).

■For character string data

The following shows how character string data is stored.

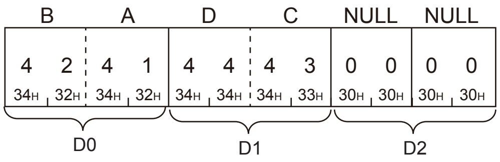
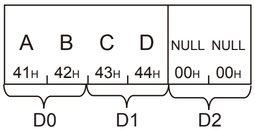
Item	For ASCII code character string	
Character string to be stored	'ABC'	'ABCD'
Character code	'41H', '42H', '43H'	'41H', '42H', '43H', '44H'
Image when character string data is stored from D0	NULL indicates 00H. 	NULL indicates 00H. 

Ex.

Write ASCII code character string data used in the instructions which handle character strings to word device

Store the character string ('ABCD') to D0 and D1: D0=4241H ('BA'), D1=4443H ('DC')

Specify the following data for write data.

ASCII code	Binary code
	

Point

When communicating ASCII code character string data in ASCII code, data is rearranged every two characters and stored.

Device memory extension specification (subcommand: bit7)

For details, refer to ☞ Page 924 Device Memory Extension Specification.

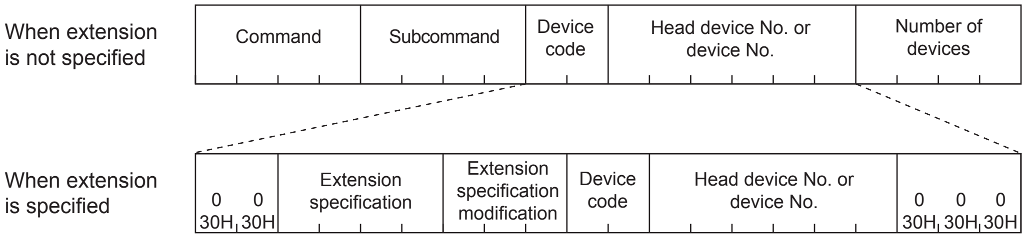
This section explains how to read or write a device from/to module access device areas and how to specify a device indirectly by using index register.

❶Message format

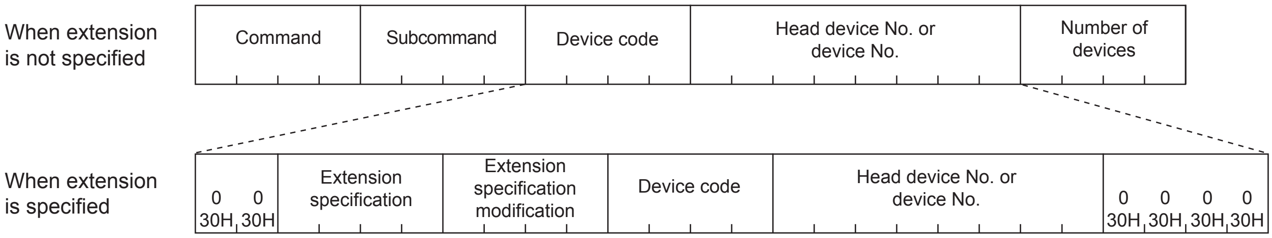
Response messages are extended as well.

- When communicating data in ASCII code

2 digit code/6 digit number specification

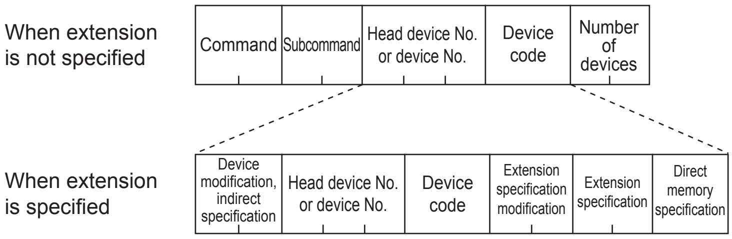


4 digit code/8 digit number specification

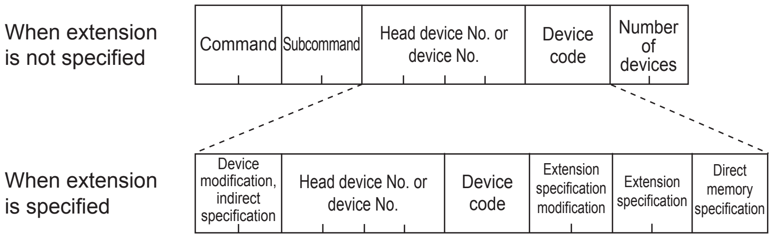


- When communicating data in binary code

2 digit code/6 digit number specification

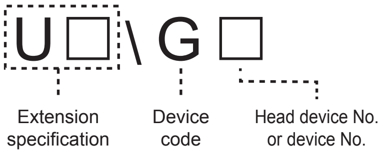


4 digit code/8 digit number specification



❷Module access device specification

The following shows the approach for module access device specification in programming and request data.



- Extension specification

Specify the start I/O number of intelligent function modules.

ASCII code	Binary code
Specify the start I/O number in hexadecimal (3-digit ASCII code). When described with 4-digits, specify the start I/O number with the upper 3-digits. Example 001 U □ □ □ 55H U 0 0 1 55H 30H 30H 31H	Specify the start I/O number in hexadecimal (2 bytes). When described with 4-digits, specify the start I/O number with the upper 3-digits. Example 001 □□H □□H 01H 00H

- Device code

Specify the module access device in the device code list (☞ Page 715 Device code list).

- Head device No. or device No.

The format is the same as the message when extension is not specified.

- Direct memory specification (only when communicating in binary code)

The type (intelligent function module device) of access device is specified.

Module access device: Specify F8H

Set/reset

Specify the ON/OFF status of bit device.

- For ON: '1'

Device specification format of subcommand	ASCII code	Binary code
2 digit code/6 digit number specification	0 1 30H 31H	01H
4 digit code/8 digit number specification	0 0 0 1 30H 30H 30H 31H	01H 00H

- For OFF: '0'

Device specification format of subcommand	ASCII code	Binary code
2 digit code/6 digit number specification	0 0 30H 30H	00H
4 digit code/8 digit number specification	0 0 0 0 30H 30H 30H 30H	00H 00H

Batch read

Data in devices are read in a batch.

Request data

■When communicating data in ASCII code

2 digit code/6 digit number specification

	4 bytes	4 bytes	2 bytes	6 bytes	4 bytes
When extension is not specified	0 4 0 1 30H 34H 30H 31H	Subcommand	Device code	Head device No.	Number of devices

	2 bytes	4 bytes	3 bytes	2 bytes	6 bytes	3 bytes
When extension is specified	0 0 30H 30H	Extension specification	Extension specification modification	Device code	Head device No.	0 0 0 30H 30H 30H

4 digit code/8 digit number specification

	4 bytes	4 bytes	4 bytes	8 bytes	4 bytes
When extension is not specified	0 4 0 1 30H 34H 30H 31H	Subcommand	Device code	Head device No.	Number of devices

	2 bytes	4 bytes	4 bytes	4 bytes	10 bytes	4 bytes
When extension is specified	0 0 30H 30H	Extension specification	Extension specification modification	Device code	Head device No.	0 0 0 0 30H 30H 30H 30H

■When communicating data in binary code

2 digit code/6 digit number specification

	2 bytes	2 bytes	3 bytes	1 byte	2 bytes
When extension is not specified	01H 04H	Subcommand	Head device No.	Device code	Number of devices

	2 bytes	3 bytes	1 byte	2 bytes	2 bytes	1 byte
When extension is specified	Device modification, indirect specification	Head device No.	Device code	Extension specification modification	Extension specification	Direct memory specification

4 digit code/8 digit number specification

	2 bytes	2 bytes	4 bytes	2 bytes	2 bytes
When extension is not specified	01H 04H	Subcommand	Head device No.	Device code	Number of devices

	2 bytes	4 bytes	2 bytes	2 bytes	2 bytes	1 byte
When extension is specified	Device modification, indirect specification	Head device No.	Device code	Extension specification modification	Extension specification	Direct memory specification

■Subcommand

Specify the subcommand selected from the item.

Item			Subcommand					
Data size specification	**Device specification format**	**Device memory extension specification**	**ASCII code (Upper column: characters, lower column: character code)**				**Binary code**	
Bit units	2 digit code/6 digit number specification	Not specified	0	0	0	1	01H	00H
			30H	30H	30H	31H		
		Specified	0	0	8	1	81H	00H
			30H	30H	38H	31H		
	4 digit code/8 digit number specification	Specified	0	0	8	3	83H	00H
			30H	30H	38H	33H		
Word units	2 digit code/6 digit number specification	Not specified	0	0	0	0	00H	00H
			30H	30H	30H	30H		
		Specified	0	0	8	0	80H	00H
			30H	30H	38H	30H		
	4 digit code/8 digit number specification	Specified	0	0	8	2	82H	00H
			30H	30H	38H	32H		

■Device code

Specify the device code that corresponds to the device type to be read. Refer to the device code list (☞ Page 715 Device code list).

The double word device and the long index register (LZ) are not supported.

■Device No.

Specify the head number of target device of reading.

■Number of devices

Specify the number of target device points of reading.

Item	Number of devices	
	ASCII code	**Binary code**
When reading data in bit units	1 to 3584 points	1 to 3584 points
When reading data in word units	1 to 960 points	1 to 960 points

Response data

The read device value is stored in hexadecimal. The data order differs depending on the type of code, ASCII code or binary code.

Read data

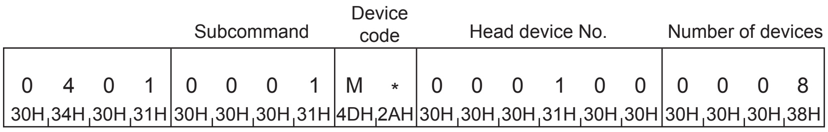
Communication example

■When reading data in bit units

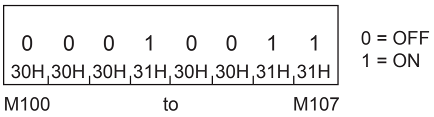
M100 to M107 are read.

- When communicating data in ASCII code

(Request data)

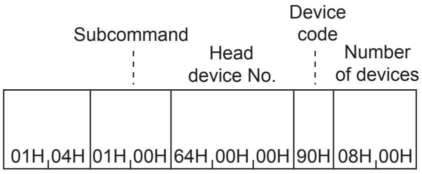


(Response data)

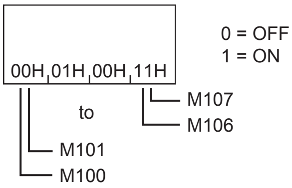


- When communicating data in binary code

(Request data)



(Response data)

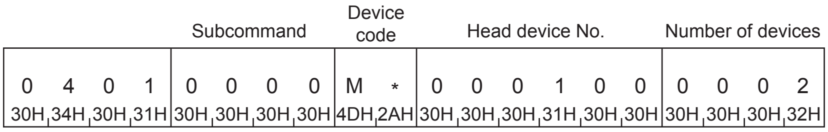


■When reading data in word units (bit device)

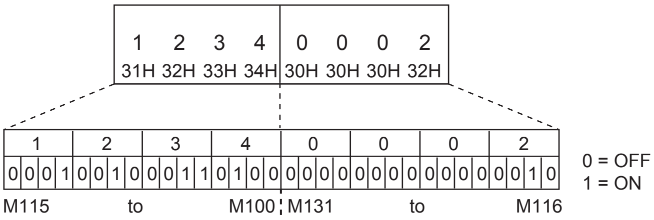
M100 to M131 (2-word) are read.

- When communicating data in ASCII code

(Request data)

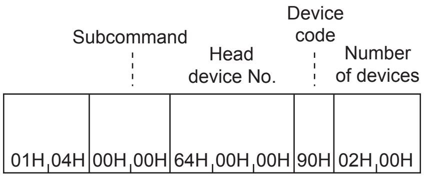


(Response data)

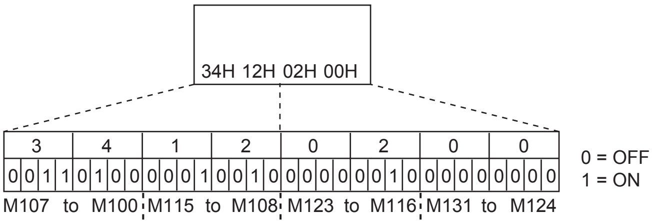


- When communicating data in binary code

(Request data)



(Response data)



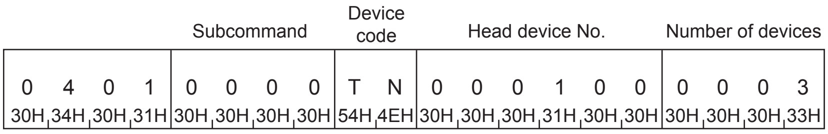
■When reading data in word units (word device)

Values in T100 to T102 are read.

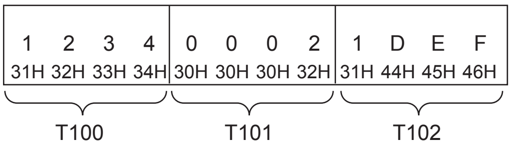
It is supposed that 4660 (1234H) is stored in T100, 2 (2H) is stored in T101, and 7663 (1DEFH) is stored in T102.

- When communicating data in ASCII code

(Request data)

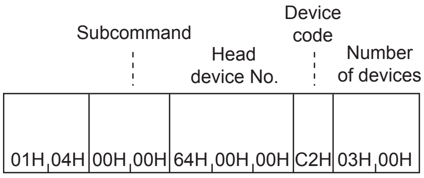


(Response data)

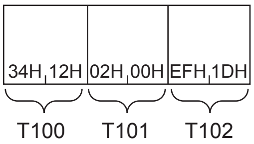


- When communicating data in binary code

(Request data)



(Response data)



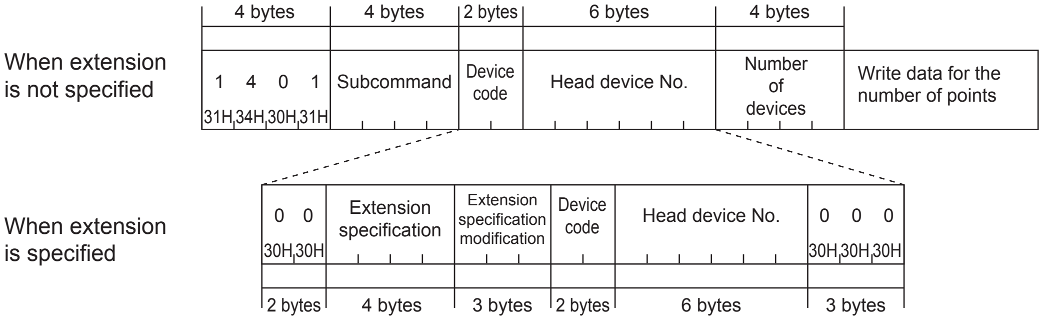
Batch write

Data in devices are written in a batch.

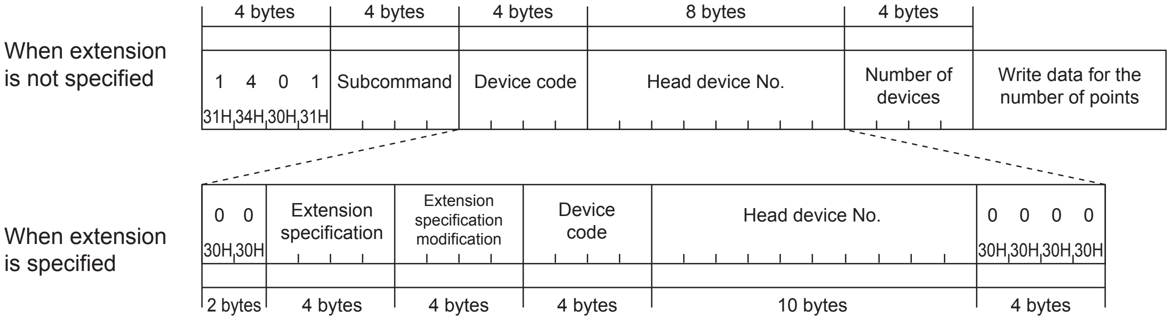
Request data

■When communicating data in ASCII code

2 digit code/6 digit number specification

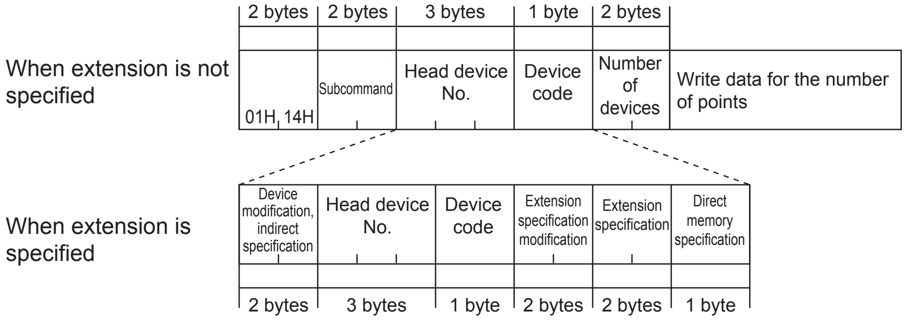


4 digit code/8 digit number specification

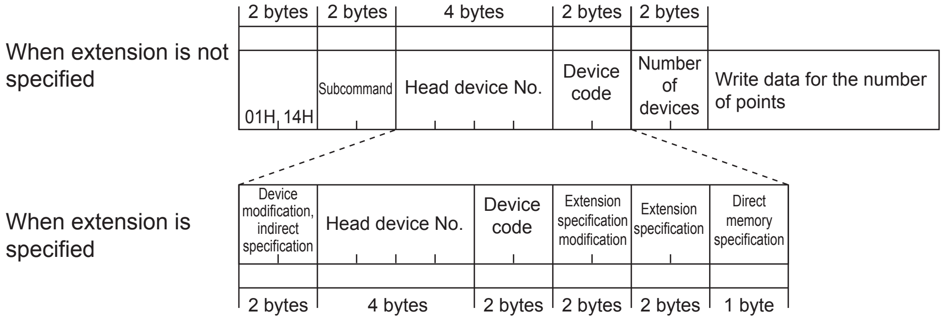


■When communicating data in binary code

2 digit code/6 digit number specification



4 digit code/8 digit number specification



■Subcommand

Specify the subcommand selected from the item.

Item			Subcommand					
Data size specification	**Device specification format**	**Device memory extension specification**	**ASCII code (Upper column: characters, lower column: character code)**				**Binary code**	
Bit units	2 digit code/6 digit number specification	Not specified	0	0	0	1	01H	00H
			30H	30H	30H	31H		
		Specified	0	0	8	1	81H	00H
			30H	30H	38H	31H		
	4 digit code/8 digit number specification	Specified	0	0	8	3	83H	00H
			30H	30H	38H	33H		
Word units	2 digit code/6 digit number specification	Not specified	0	0	0	0	00H	00H
			30H	30H	30H	30H		
		Specified	0	0	8	0	80H	00H
			30H	30H	38H	30H		
	4 digit code/8 digit number specification	Specified	0	0	8	2	82H	00H
			30H	30H	38H	32H		

■Device code

Specify the device code that corresponds to the device type to be written. Refer to the device code list (☞ Page 715 Device code list).

The double word device and the long index register (LZ) are not supported.

■Device No.

Specify the head number of target device of writing.

■Number of devices

Specify the number of target device points of writing.

Item	Number of devices	
	ASCII code	**Binary code**
When writing data in bit units	1 to 3584 points	1 to 3584 points
When writing data in word units	1 to 960 points	1 to 960 points

■Write data

Specify value to be written to a device for the number of points specified in "Device point".

Response data

There is no response data for the Device Write command.

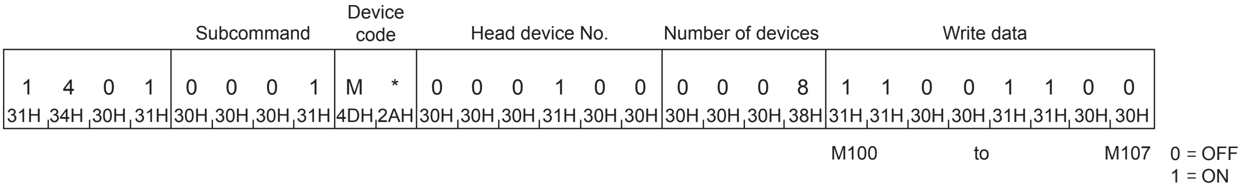
Communication example

■When writing data in bit units

Values are written to M100 to M107.

- When communicating data in ASCII code

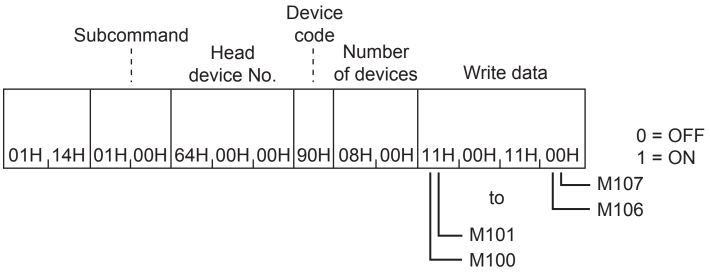
(Request data)



	Subcommand	Device code	Head device No.	Number of devices	Write data
1 4 0 1	0 0 0 1	M *	0 0 0 1 0 0	0 0 0 8	1 1 0 0 1 1 0 0
31H 34H 30H 31H	30H 30H 30H 31H	4DH 2AH	30H 30H 30H 31H 30H 30H	30H 30H 30H 38H	31H 31H 30H 30H 31H 31H 30H 30H

- When communicating data in binary code

(Request data)



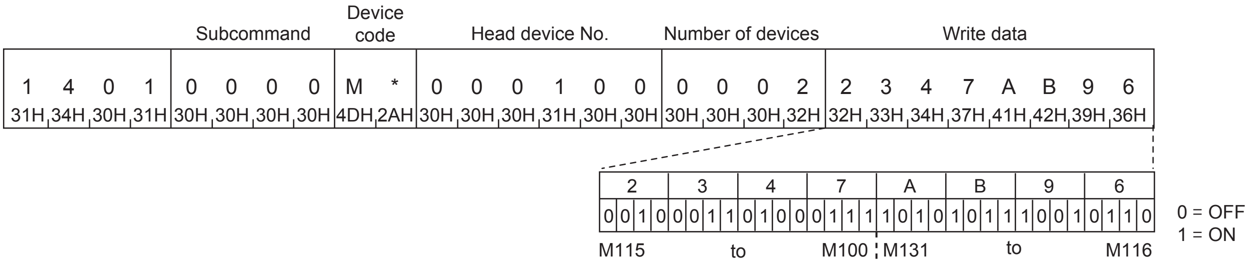
	Subcommand	Head device No.	Device code	Number of devices	Write data
01H 14H	01H 00H	64H 00H 00H	90H	08H 00H	11H 00H 11H 00H

■When writing data in word units (bit device)

Values are written to M100 to M131 (2-word).

- When communicating data in ASCII code

(Request data)

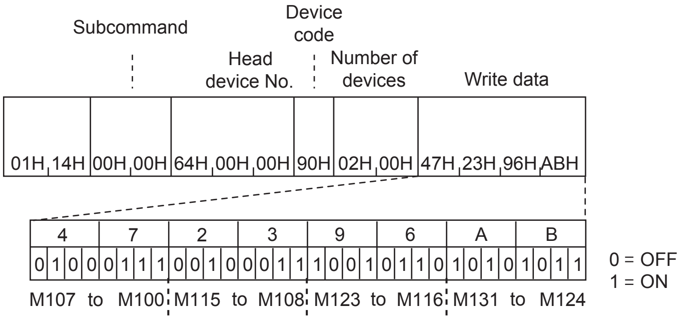


	Subcommand	Device code	Head device No.	Number of devices	Write data
1 4 0 1	0 0 0 0	M *	0 0 0 1 0 0	0 0 0 2	2 3 4 7 A B 9 6
31H 34H 30H 31H	30H 30H 30H 30H	4DH 2AH	30H 30H 30H 31H 30H 30H	30H 30H 30H 32H	32H 33H 34H 37H 41H 42H 39H 36H

2	3	4	7	A	B	9	6
0 0 1 0	0 0 1 1	0 1 0 0	0 1 1 1	1 0 1 0	1 0 1 1	1 0 0 1	0 1 1 0

- When communicating data in binary code

(Request data)



	Subcommand	Head device No.	Device code	Number of devices	Write data
01H 14H	00H 00H	64H 00H 00H	90H	02H 00H	47H 23H 96H ABH

4	7	2	3	9	6	A	B
0 1 0 0	0 1 1 1	0 0 1 0	0 0 1 1	1 0 0 1	0 1 1 0	1 0 1 0	1 0 1 1

■When writing data in word units (word device)

6549 (1995H) is written in D100, 4610 (1202H) is written in D101, and 4400 (1130H) is written in D102.

- When communicating data in ASCII code

(Request data)

	Subcommand	Device code	Head device No.	Number of devices	Write data		
1 4 0 1	0 0 0 0	D *	0 0 0 1 0 0	0 0 0 3	1 9 9 5	1 2 0 2	1 1 3 0
31H 34H 30H 31H	30H 30H 30H 30H	44H 2AH	30H 30H 30H 31H 30H 30H	30H 30H 30H 33H	31H 39H 39H 35H	31H 32H 30H 32H	31H 31H 33H 30H
					D100	D101	D102

- When communicating data in binary code

(Request data)

	Subcommand	Head device No.	Device code	Number of devices	Write data		
01H 14H	00H 00H	64H 00H 00H	A8H	03H 00H	95H 19H	02H 12H	30H 11H
					D100	D101	D102

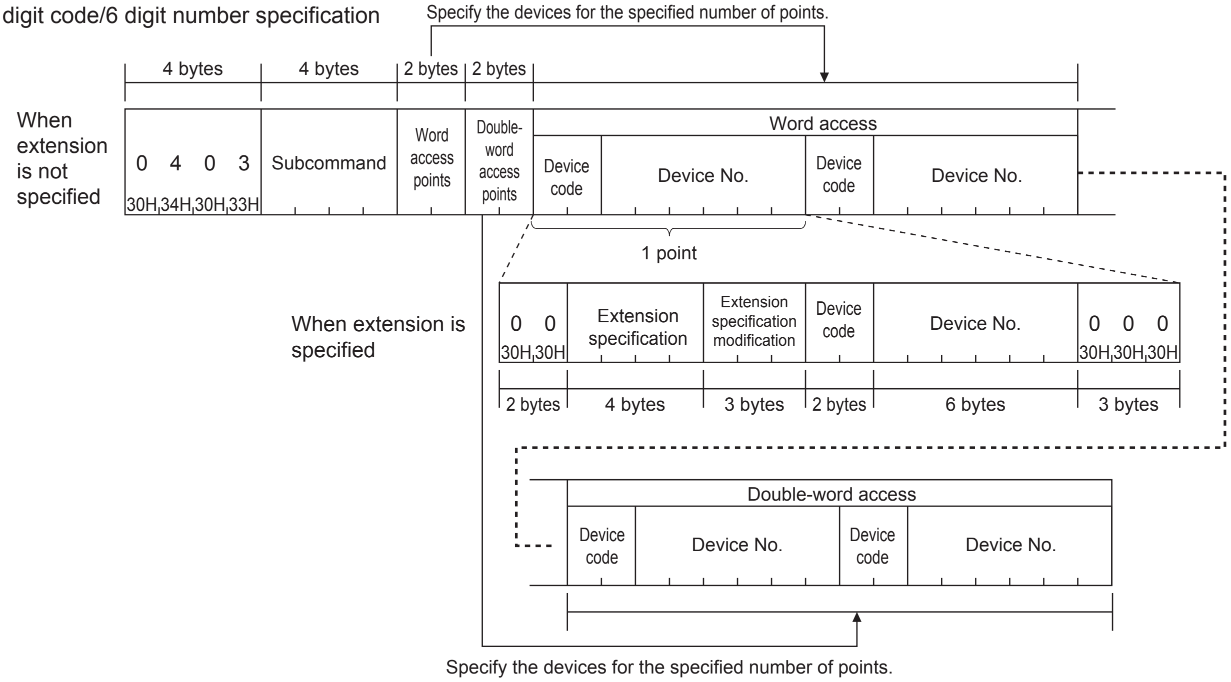
Random read

This command specifies the device No. randomly and reads the device value.

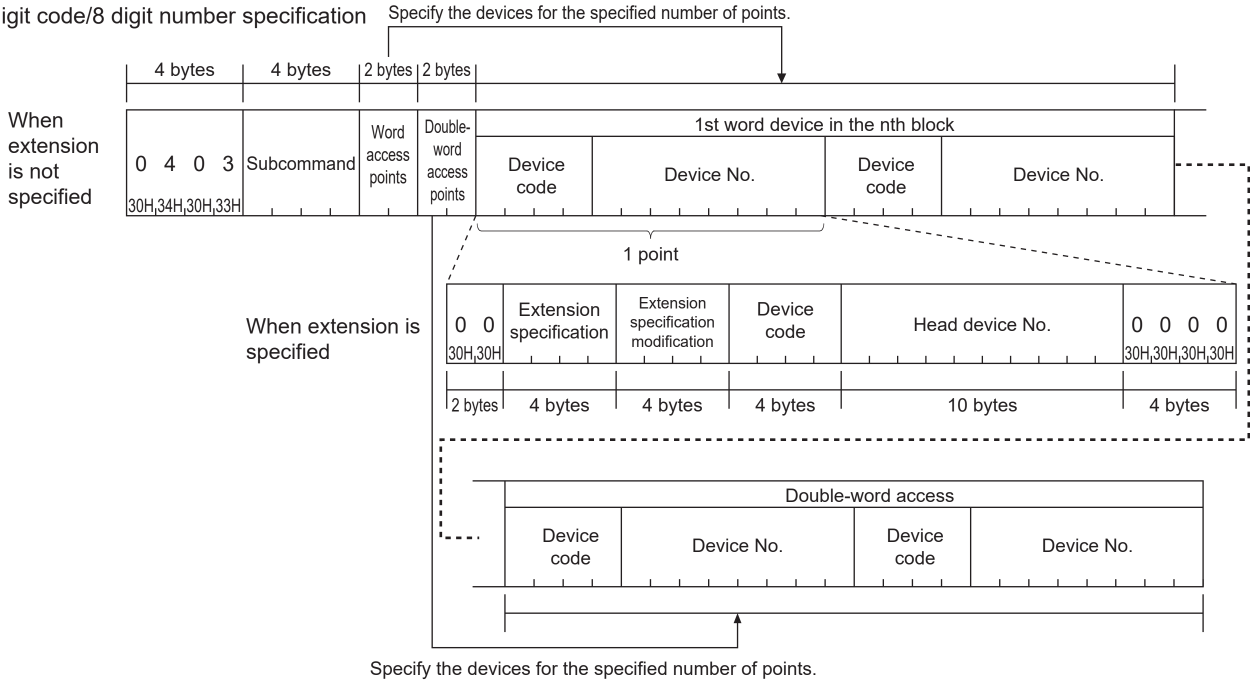
Request data

■When communicating data in ASCII code

2 digit code/6 digit number specification

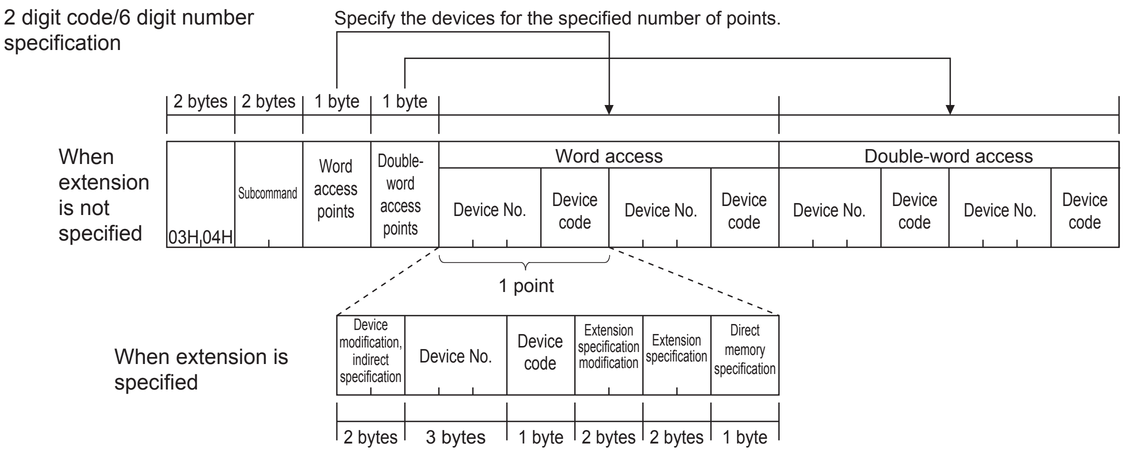


4 digit code/8 digit number specification

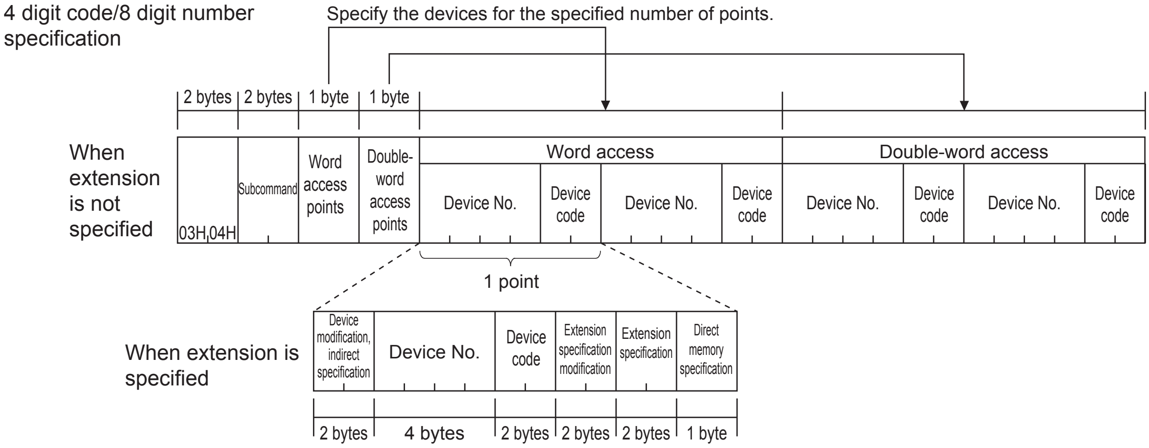


■When communicating data in binary code

2 digit code/6 digit number specification



4 digit code/8 digit number specification



■Subcommand

Specify the subcommand selected from the item.

Item			Subcommand					
Data size specification	**Device specification format**	**Device memory extension specification**	**ASCII code (Upper column: characters, lower column: character code)**				**Binary code**	
Word units	2 digit code/6 digit number specification	Not specified	0	0	0	0	00H	00H
			30H	30H	30H	30H		
		Specified	0	0	8	0	80H	00H
			30H	30H	38H	30H		
	4 digit code/8 digit number specification	Specified	0	0	8	2	82H	00H
			30H	30H	38H	32H		

■Word access points, double-word access points

Specify the number of target device points of reading.

Item	Description	Number of points	
		ASCII code	**Binary code**
Word access points	Specify the number of points to be accessed in one-word units. The bit device is 16-point units, the word device is one-word units.	1 ≤ word access points + double-word access points ≤ 192 When device memory extension specification is used, double the number of the access points.	
Double-word access points	Specify the number of points to be accessed in two-word units. The bit device is 32-point units, the word device is two-word units.		

■Device code, device No.

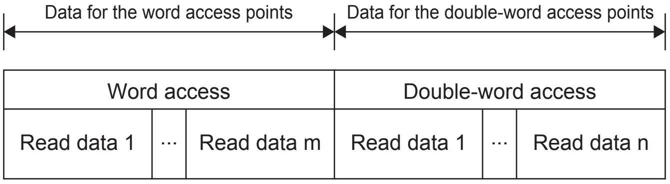
Specify the target device of reading.

Item	Description
Word access	Specify the device points specified as word access points. The specification is not necessary when the word access points are zero.
Double-word access	Specify the device points specified as double-word access points. The specification is not necessary when the double-word access points are zero.

Set up in order of word access device → double word access device.

Response data

The read device value is stored in hexadecimal. The data order differs depending on the type of code, ASCII code or binary code.



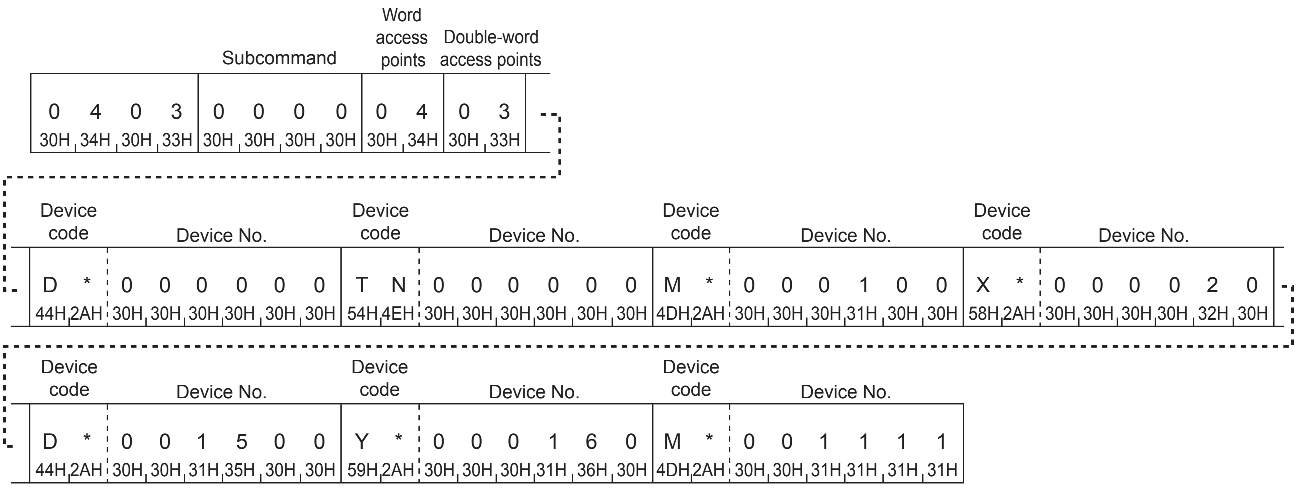
Communication example

Read D0, T0, M100 to M115, X20 to X37 by word access, and D1500 to D1501, Y160 to Y217, M1111 to M1142 by double-word access.

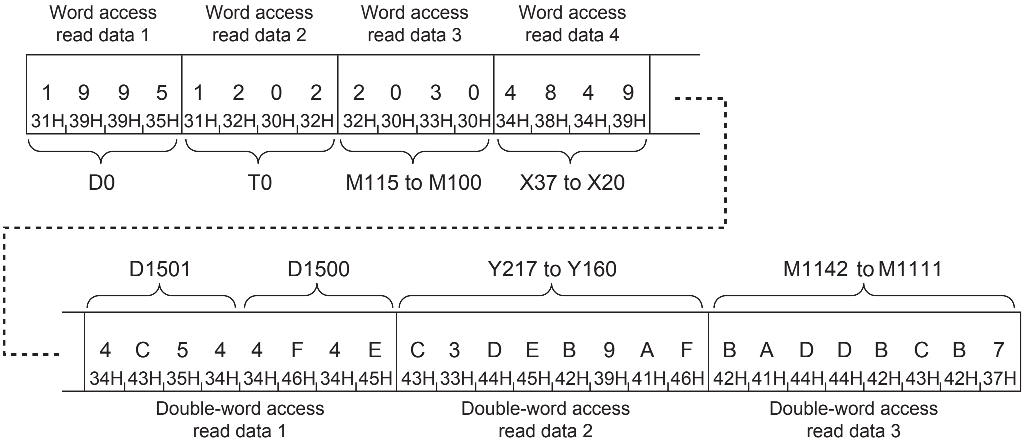
It is supposed that 6549 (1995H) is stored in D0, 4610 (1202H) is stored in T0, 20302 (4F4EH) is stored in D1500, 19540 (4C54H) is stored in D1501.

■When communicating data in ASCII code (X, Y OCT)

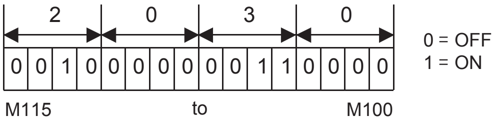
- Request data



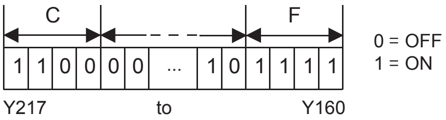
- Response data



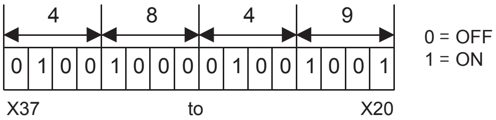
Word access read data 3



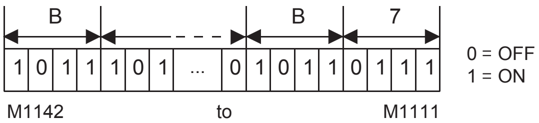
Double-word access read data 2



Word access read data 4

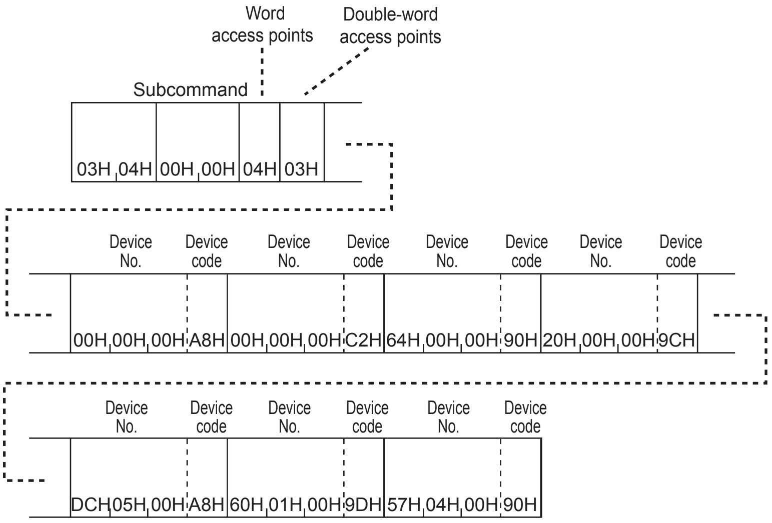


Double-word access read data 3

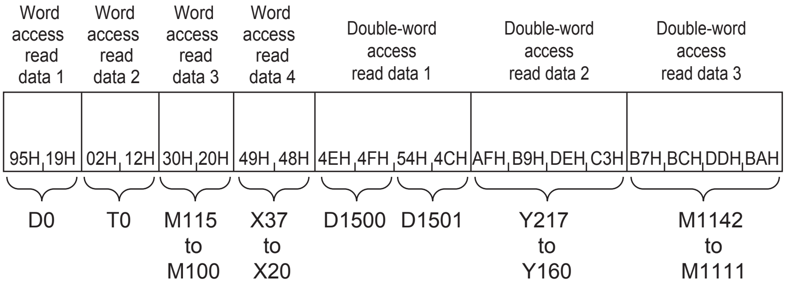


■When communicating data in binary code

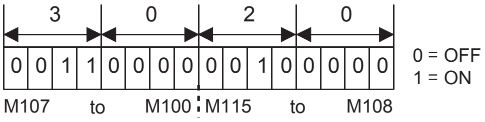
- Request data



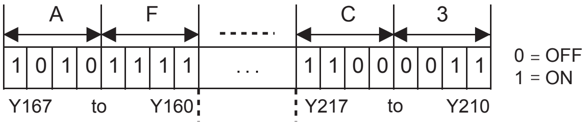
- Response data



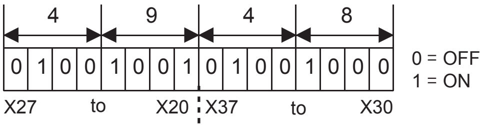
Word access read data 3



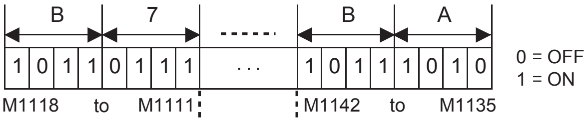
Double-word access read data 2



Word access read data 4



Double-word access read data 3



Random write

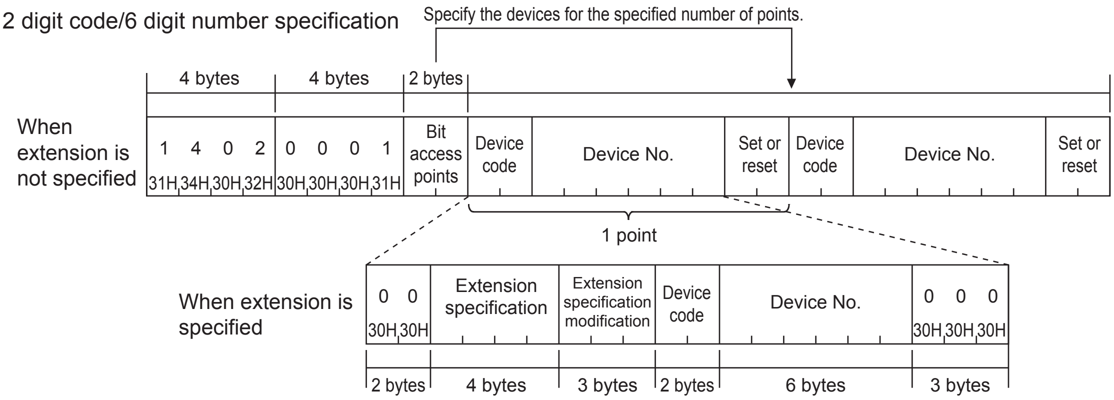
This command specifies the device No. randomly and writes the data.

Request data

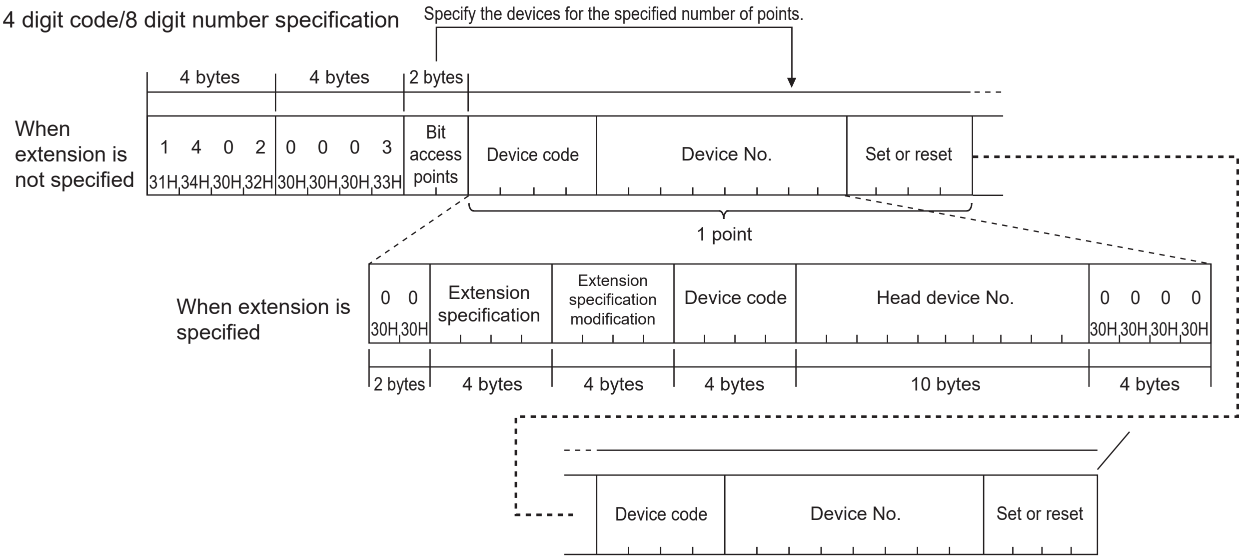
■When writing data in bit units

- When communicating data in ASCII code

2 digit code/6 digit number specification

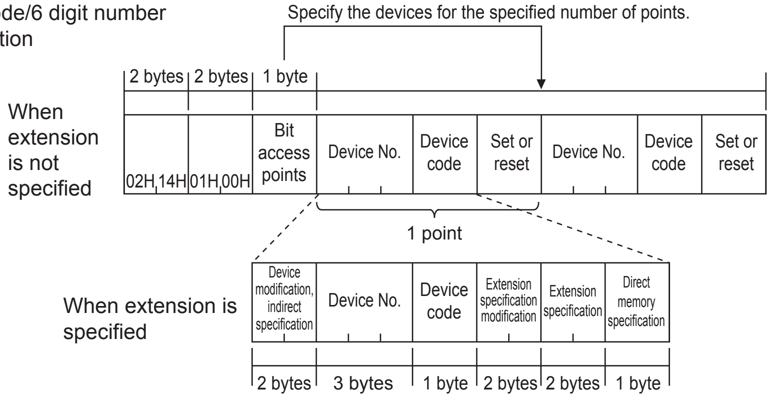


4 digit code/8 digit number specification

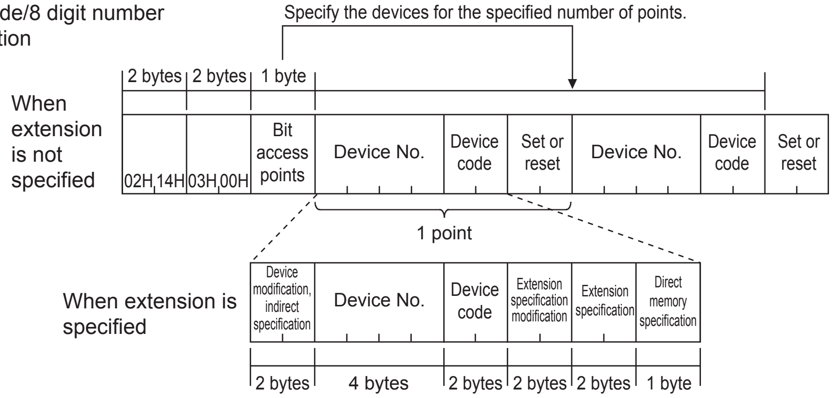


- When communicating data in binary code

2 digit code/6 digit number specification



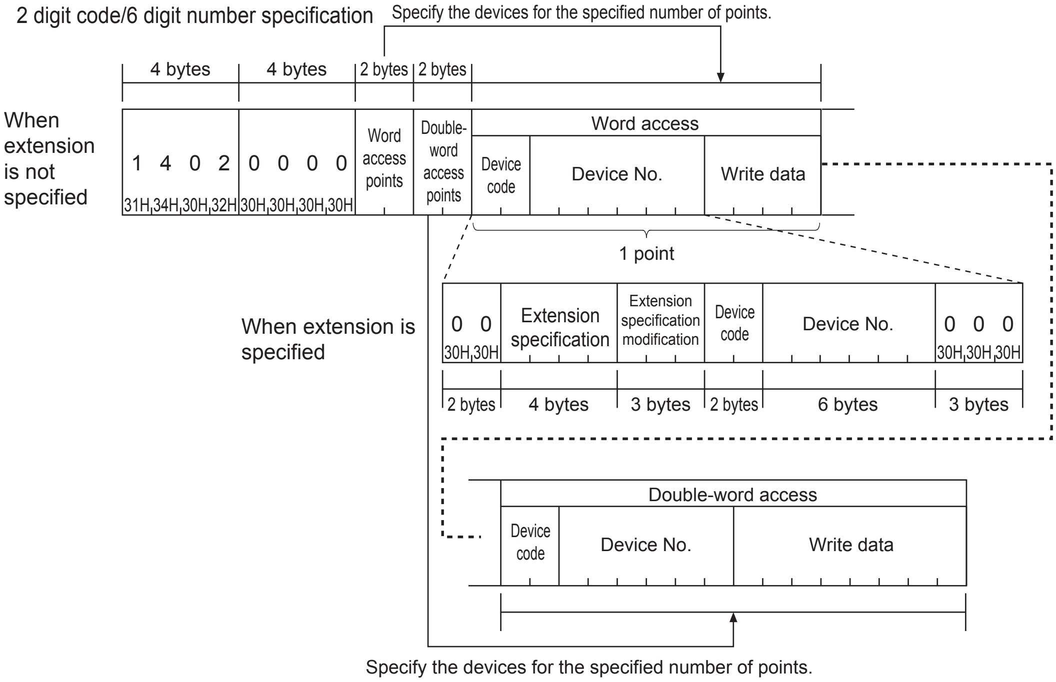
4 digit code/8 digit number specification



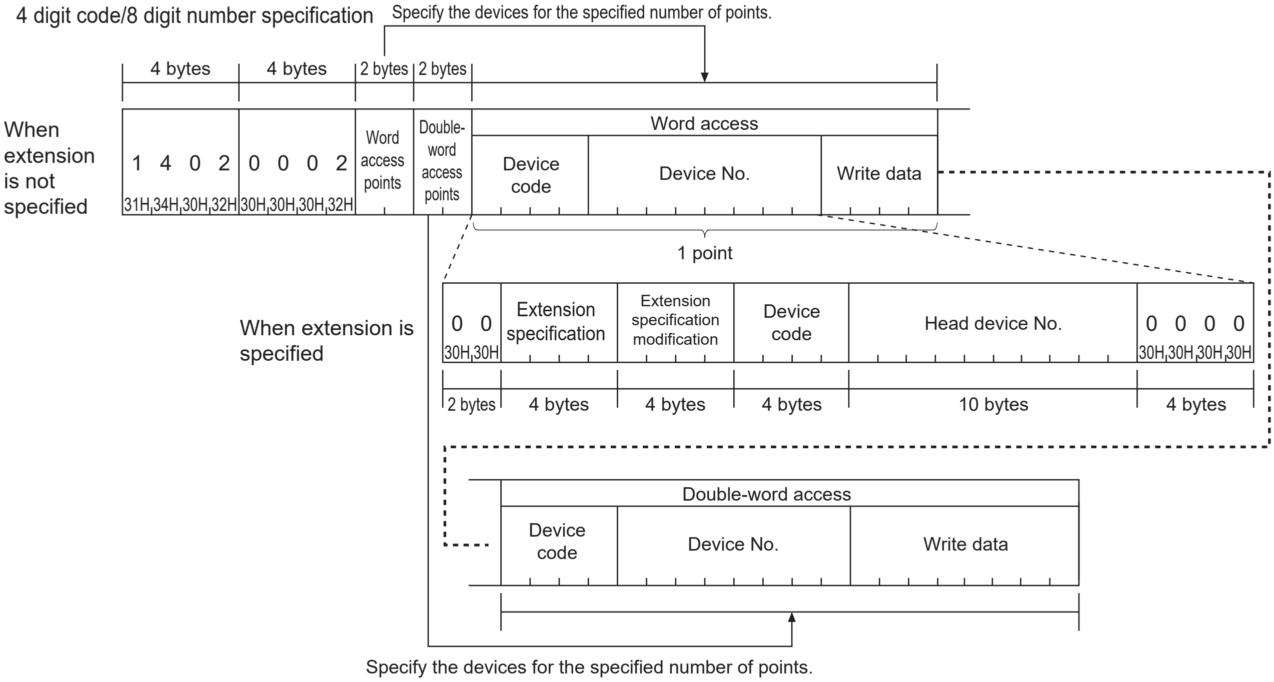
■When writing data in word units

- When communicating data in ASCII code

2 digit code/6 digit number specification

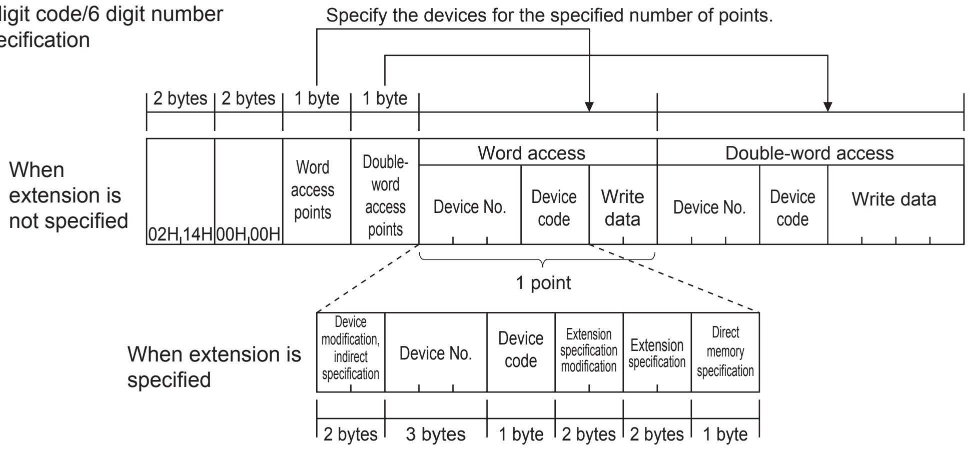


4 digit code/8 digit number specification

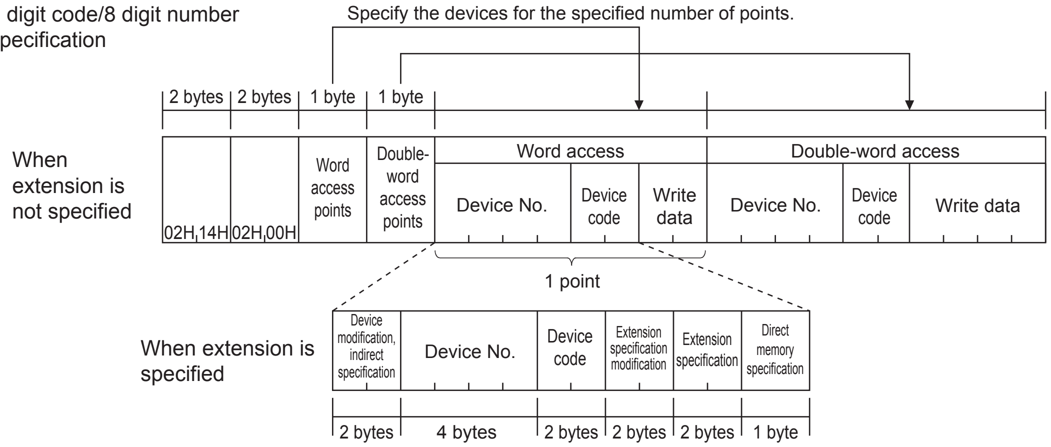


- When communicating data in binary code

2 digit code/6 digit number specification



4 digit code/8 digit number specification



■Subcommand

Specify the subcommand selected from the item.

Item			Subcommand					
Data size specification	**Device specification format**	**Device memory extension specification**	**ASCII code (Upper column: characters, lower column: character code)**				**Binary code**	
Bit units	2 digit code/6 digit number specification	Not specified	0	0	0	1	01H	00H
			30H	30H	30H	31H		
		Specified	0	0	8	1	81H	00H
			30H	30H	38H	31H		
	4 digit code/8 digit number specification	Specified	0	0	8	3	83H	00H
			30H	30H	38H	33H		
Word units	2 digit code/6 digit number specification	Not specified	0	0	0	0	00H	00H
			30H	30H	30H	30H		
		Specified	0	0	8	0	80H	00H
			30H	30H	38H	30H		
	4 digit code/8 digit number specification	Specified	0	0	8	2	82H	00H
			30H	30H	38H	32H		

■Bit access points, word access points, double-word access points

Item	Description	Number of points	
		ASCII code	Binary code
Bit access points	Specify the number of bit device points in one-point units.	1 to 188 When device memory extension specification is used 1 to 94	
Word access points	Specify the number of points to be accessed in one-word units. The bit device is 16-point units, the word device is one-word units.	1 ≤ word access points × 12 + double-word access points × 14 ≤ 1920 When device memory extension specification is used, double the number of the access points.	
Double-word access points	Specify the number of points to be accessed in two-word units. The bit device is 32-point units, the word device is two-word units.		

■Device code, device No., write data

Specify the target device of writing.

The data is specified in hexadecimal number.

Item	Description
Word access	Specify the device points specified as word access points. The specification is not necessary when the word access points are zero.
Double-word access	Specify the device points specified as double-word access points. The specification is not necessary when the double-word access points are zero.

■Set or reset

Specify ON/OFF of the bit device.

- 2 digit code/6 digit number specification

Item	Data to write		Remark
	ON	OFF	
ASCII code	"01"	"00"	Two characters will be sent in order from "0".
Binary code	01H	00H	The one-byte numerical value shown left will be sent.

- 4 digit code/8 digit number specification

Item	Data to write		Remark
	ON	OFF	
ASCII code	"0001"	"0000"	Four characters will be sent in order from "0".
Binary code	0001H	0000H	The two-byte numerical value shown left will be sent.

Response data

There is no response data for the Random write command.

Communication example

■When writing data in bit units

Turn off M50 and turn on Y27.

- When communicating data in ASCII code (X, Y OCT)

(Request data)

	Subcommand	Bit access points	Device code	Device No.	Set or reset	Device code	Device No.	Set or reset
1 4 0 2	0 0 0 1	0 2	M *	0 0 0 0 5 0	0 0	Y *	0 0 0 0 2 7	0 1
31H 34H 30H 32H	30H 30H 30H 31H	30H 32H	4DH 2AH	30H 30H 30H 30H 35H 30H	30H 30H	59H 2AH	30H 30H 30H 30H 32H 37H	30H 31H

- When communicating data in binary code

(Request data)

	Subcommand	Bit access points	Device No.	Device code	Set or reset	Device No.	Device code	Set or reset
02H 14H	01H 00H	02H	32H 00H 00H	90H	00H	27H 00H 00H	9DH	01H

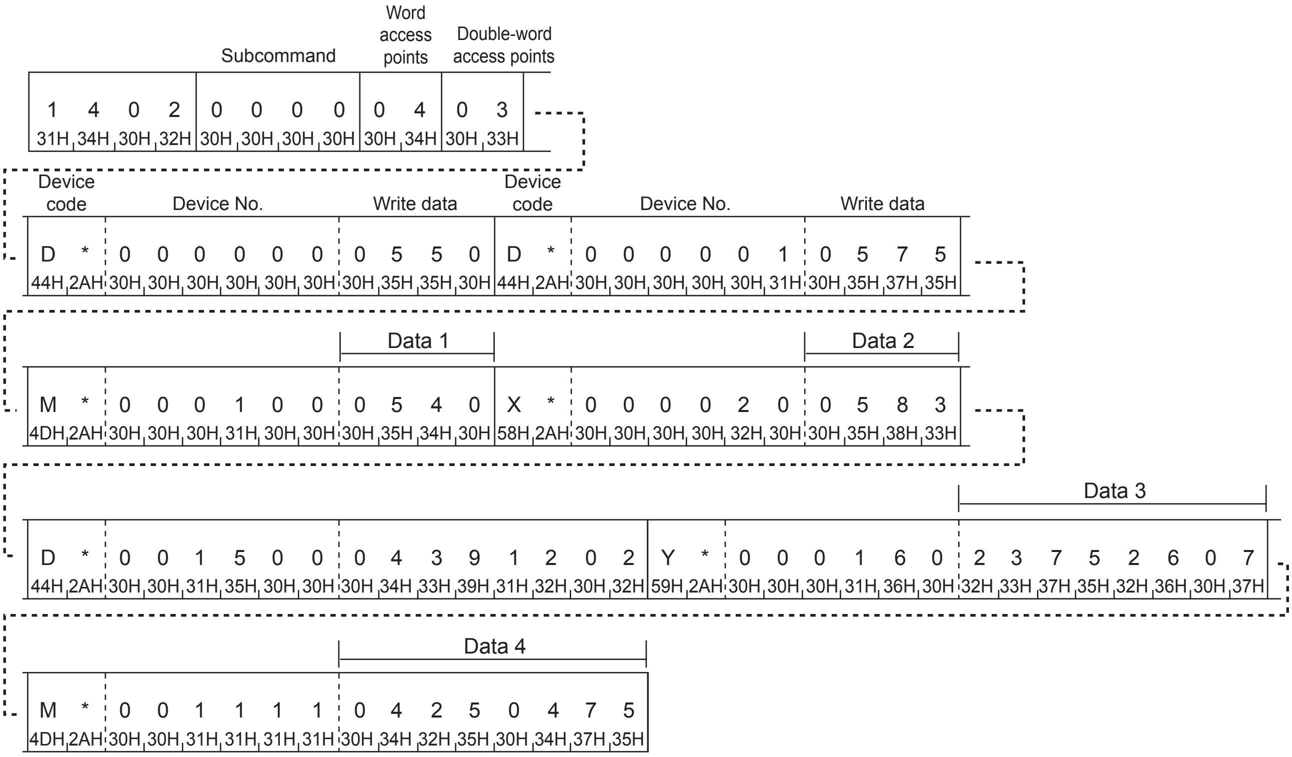
■When writing data in word units

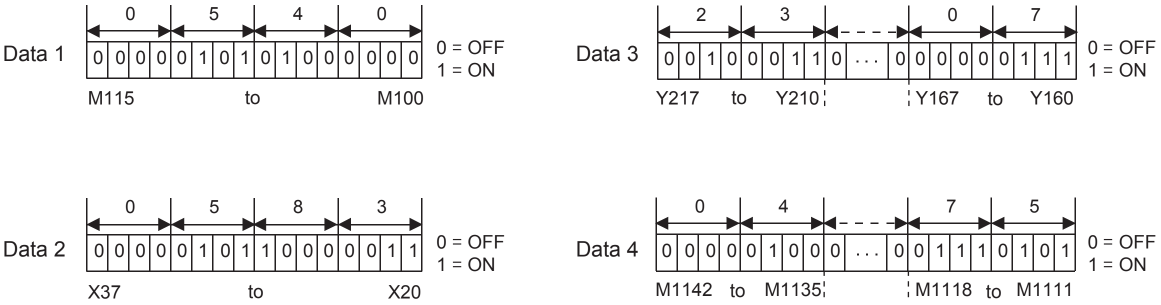
Write the value in a device as follows.

Item	Target device
Word access	D0, D1, M100 to M115, X20 to X37
Double-word access	D1500 to D1501, Y160 to Y217, M1111 to M1142

- When communicating data in ASCII code (X, Y OCT)

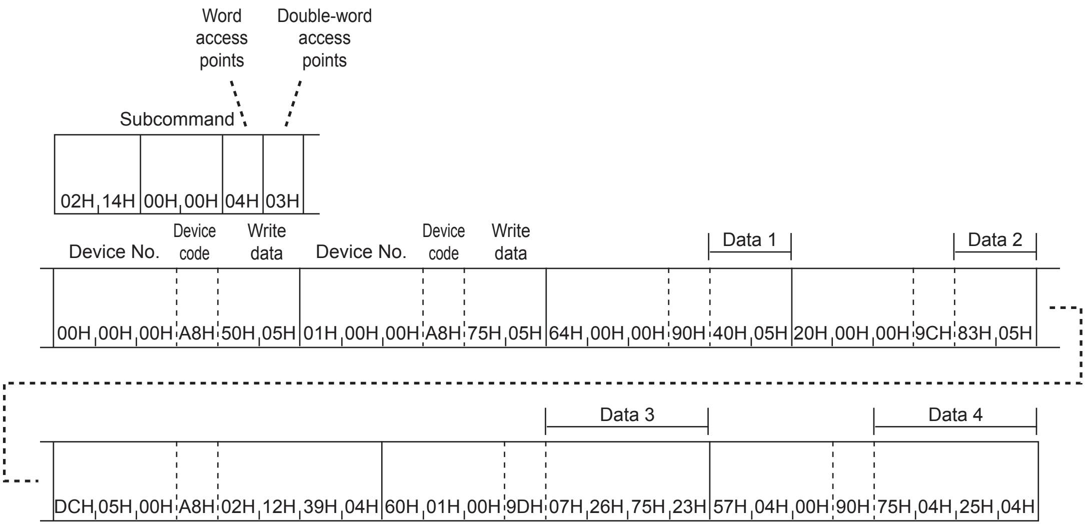
(Request data)

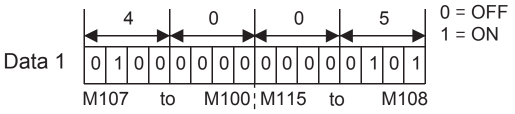


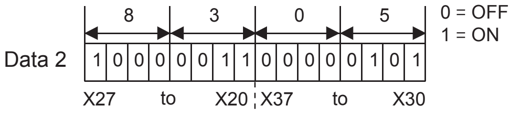


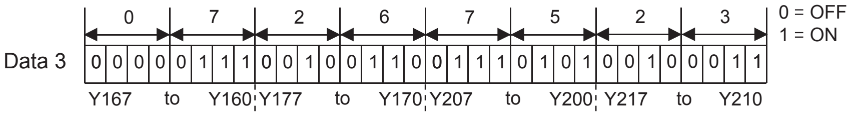
- When communicating data in binary code

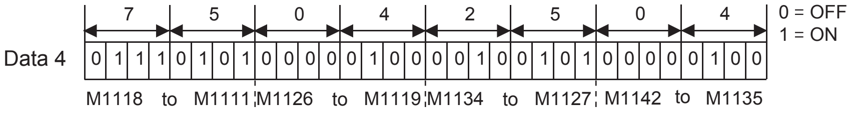
(Request data)











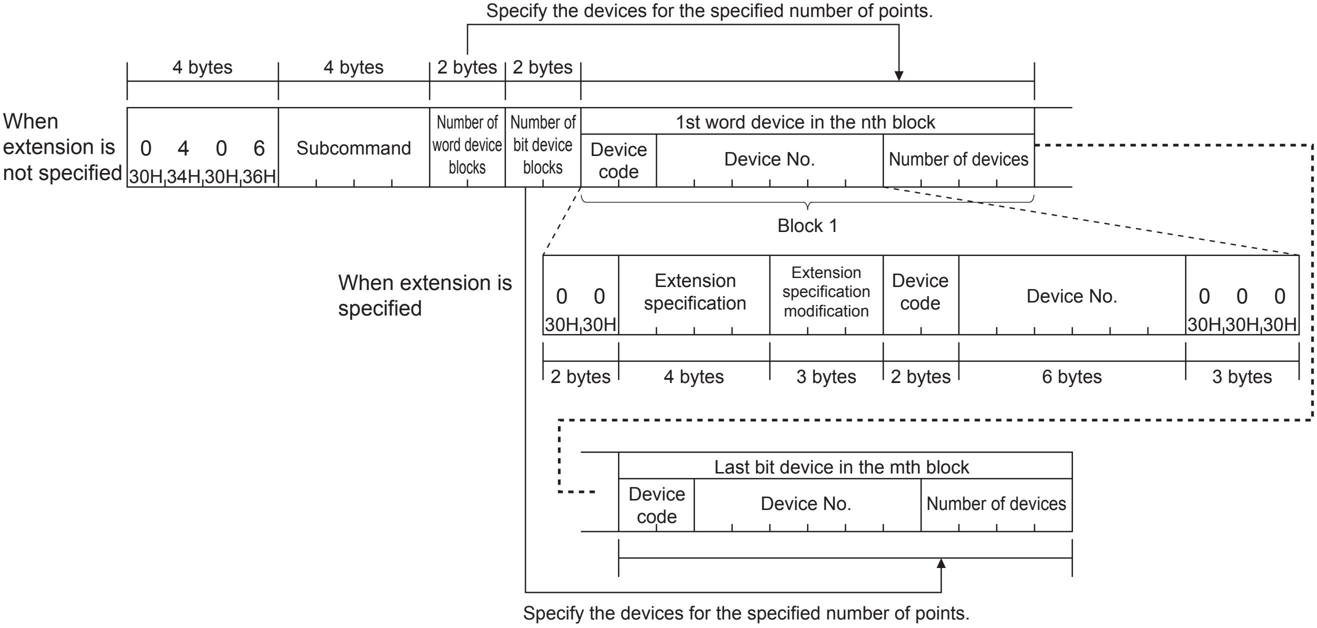
Batch read multiple blocks

The examples shown in this section explain the control procedure for reading by randomly specifying multiple blocks, where 1 block consists of n point(s) of bit device memory (one point is specified by 16-bit) or word device memory (one point is specified by 1-word).

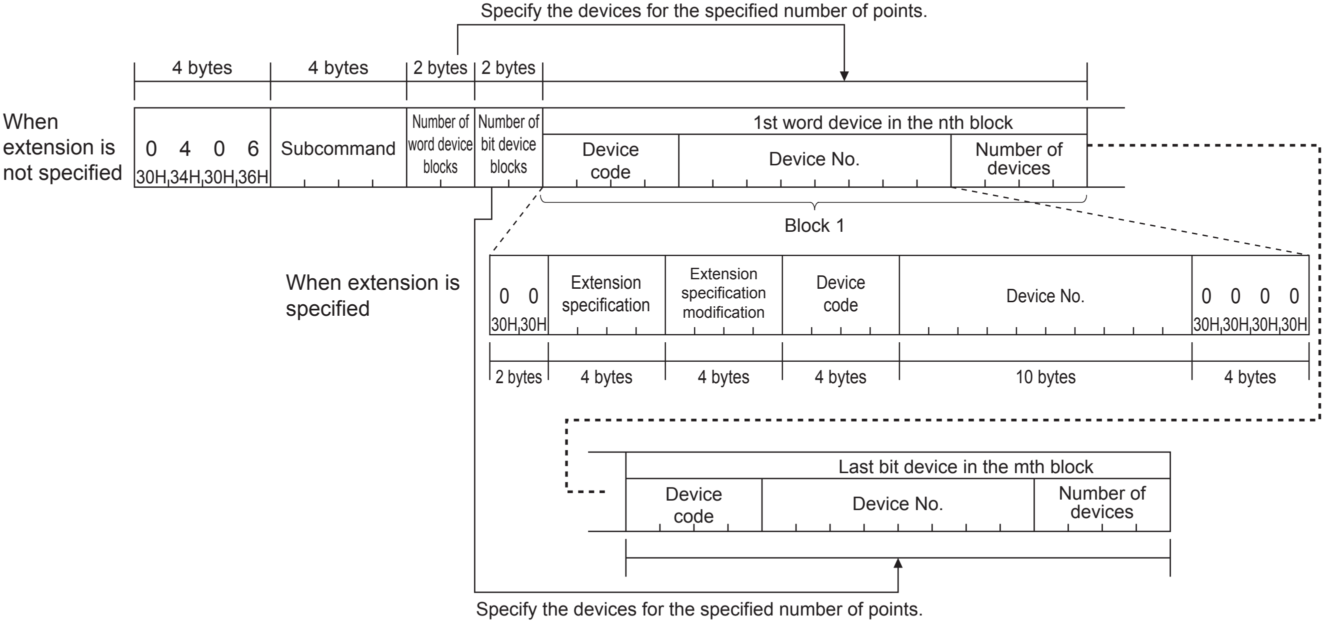
Request data

■When communicating data in ASCII code

2 digit code/6 digit number specification

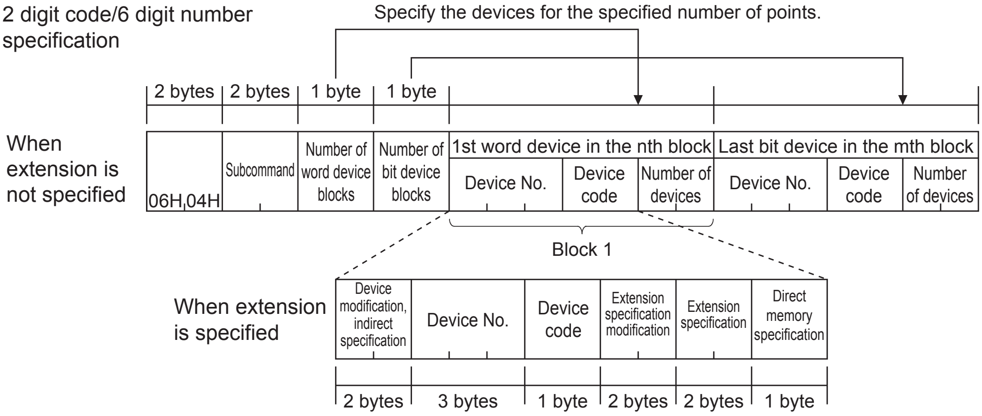


4 digit code/8 digit number specification

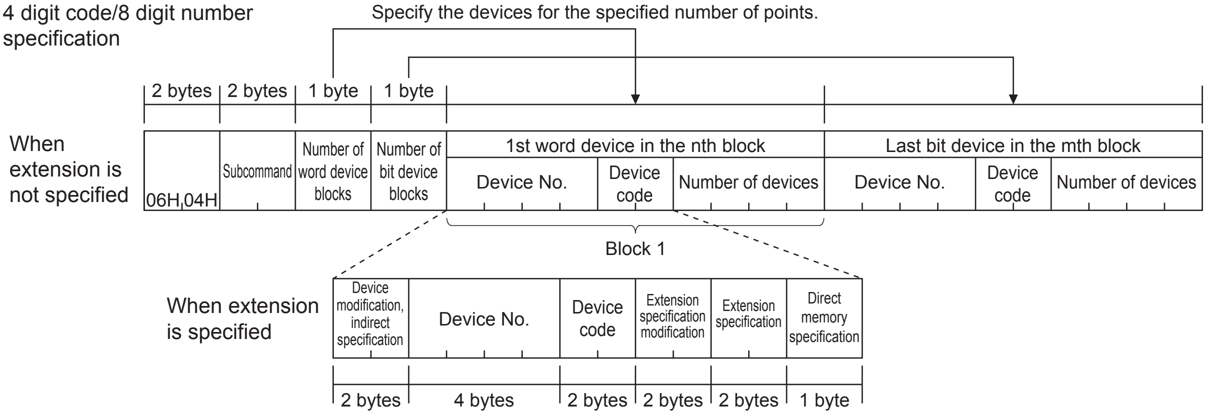


■When communicating data in binary code

2 digit code/6 digit number specification



4 digit code/8 digit number specification



①Subcommand

Specify the subcommand selected from the item.

Item			Subcommand					
Data size specification	**Device specification format**	**Device memory extension specification**	**ASCII code (Upper column: characters, lower column: character code)**				**Binary code**	
Word units	2 digit code/6 digit number specification	Not specified	0	0	0	0	00H	00H
			30H	30H	30H	30H		
		Specified	0	0	8	0	80H	00H
			30H	30H	38H	30H		
	4 digit code/8 digit number specification	Specified	0	0	8	2	82H	00H
			30H	30H	38H	32H		

②Number of word device blocks and number of bit device blocks

Specify the number of blocks of the device to be read in hexadecimal.

Item	Description	Number of points	
		ASCII code	**Binary code**
Number of word device blocks	Specify the number of blocks of the word device to be read.	Number of word device blocks + Number of bit device blocks ≤ 120 When device memory extension specification is used, double the number of the block points.	
Number of bit device blocks	Specify the number of blocks of the bit device to be read.		

❸Device code, device No., number of device points

Specify the device points while satisfying the following conditions:

Total number of points for all word device blocks + total number of points for all bit device blocks ≤ 960

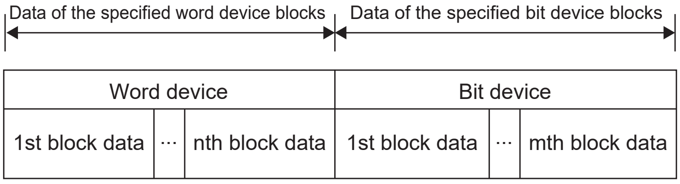
Item	Description
Word device	Specify the device of the points specified in "Number of word device blocks". When "Number of word device blocks" is set to 0, this specification is unnecessary.
Bit device	Specify the device of the points specified in "Number of bit device blocks". When "Number of bit device blocks" is set to 0, this specification is unnecessary.

Point

When specifying a contact or coil of a timer, retentive timer, or counter, use the bit device block.

Set up in order of word device → bit device.

Response data



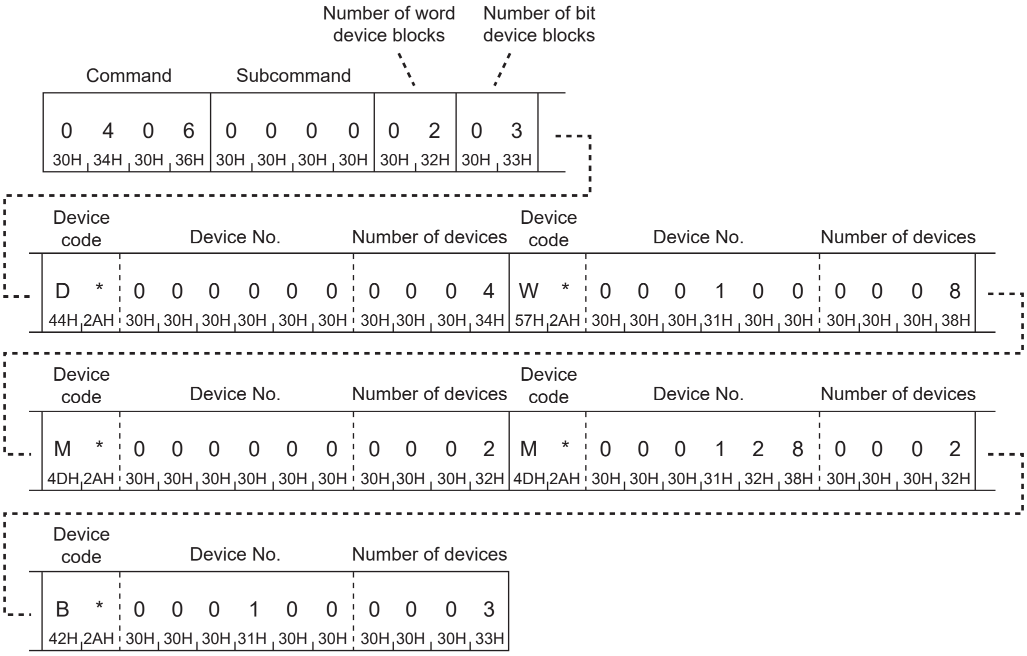
Communication example

Values are read from devices as follows.

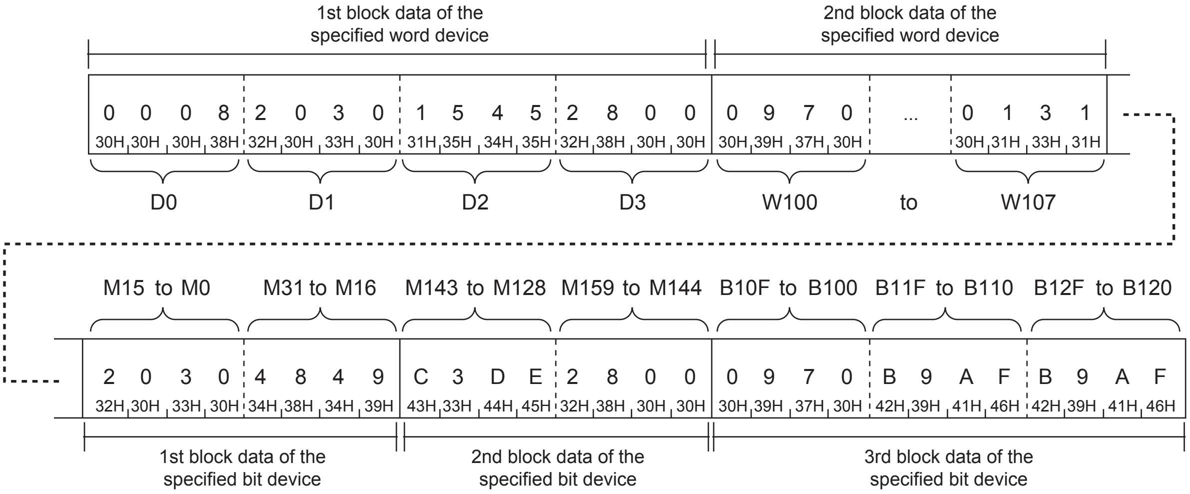
Item	Read contents
Word device	• Block 1: D0 to D3 (4 points) • Block 2: W100 to W107 (8 points)
Bit device	• Block 1: M0 to M31 (2 points) • Block 2: M128 to M159 (2 points) • Block 3: B100 to B12F (3 points)

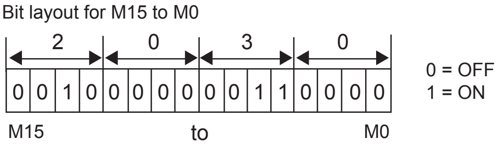
■When communicating data in ASCII code

(Request data)



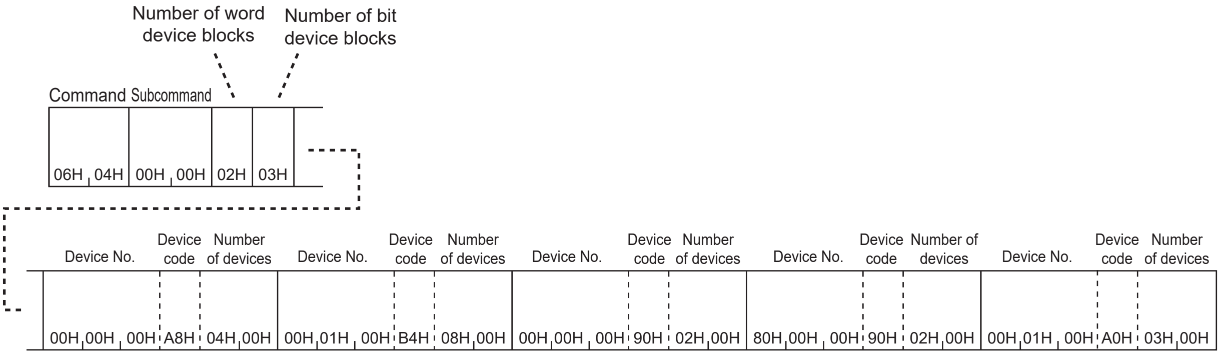
(Response data)



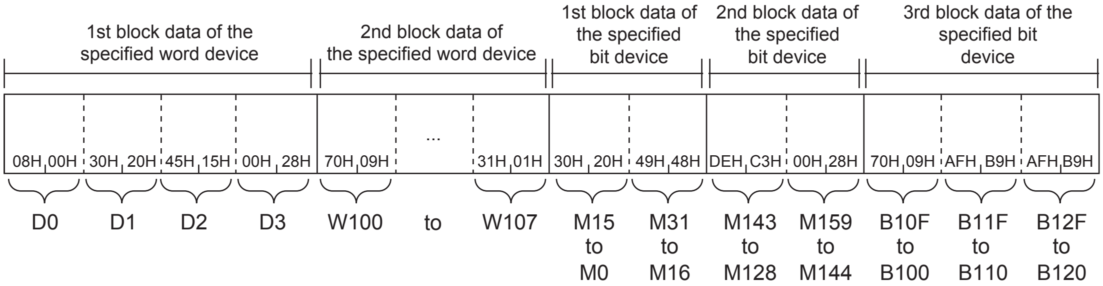


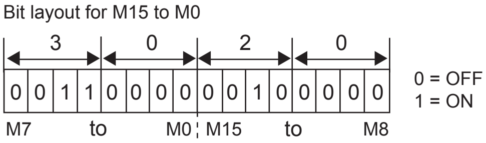
■When communicating data in binary code

(Request data)



(Response data)





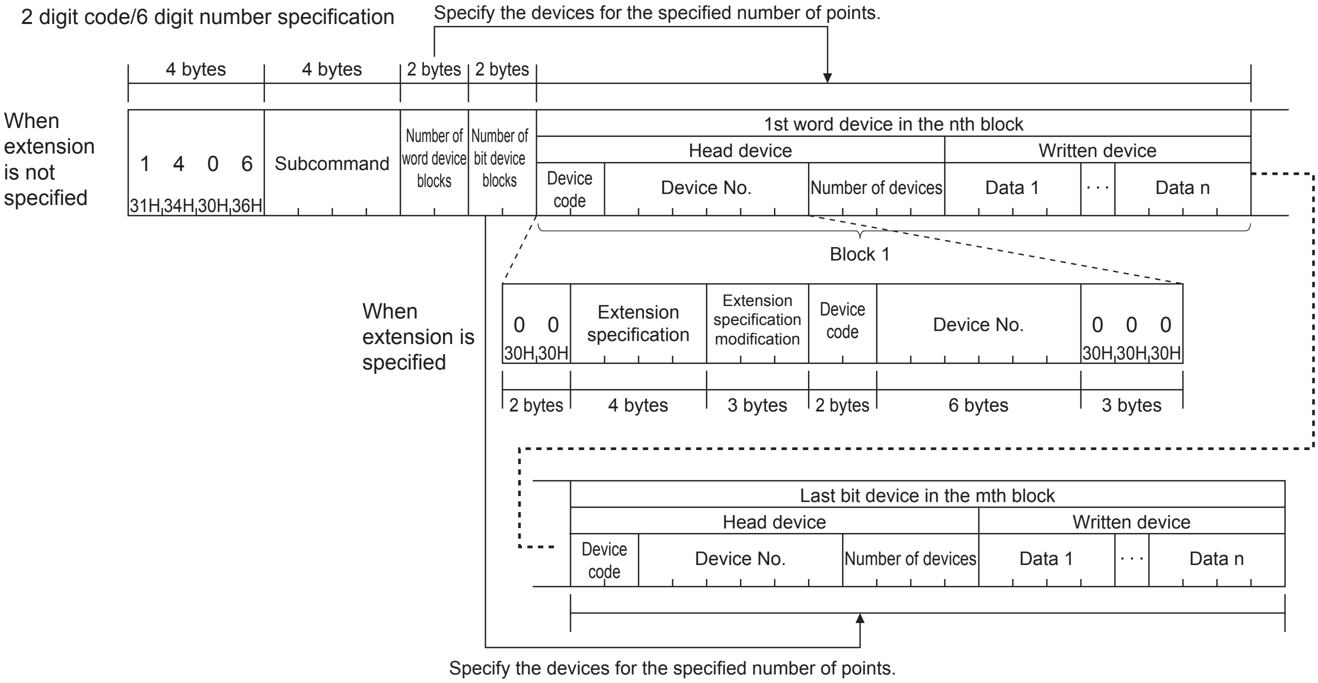
Batch write multiple blocks

The examples shown in this section explain the control procedure for writing by randomly specifying multiple blocks, where 1 block consists of n point(s) of a bit device memory (one point is specified by 16-bit) and a word device memory (one point is specified by 1-word).

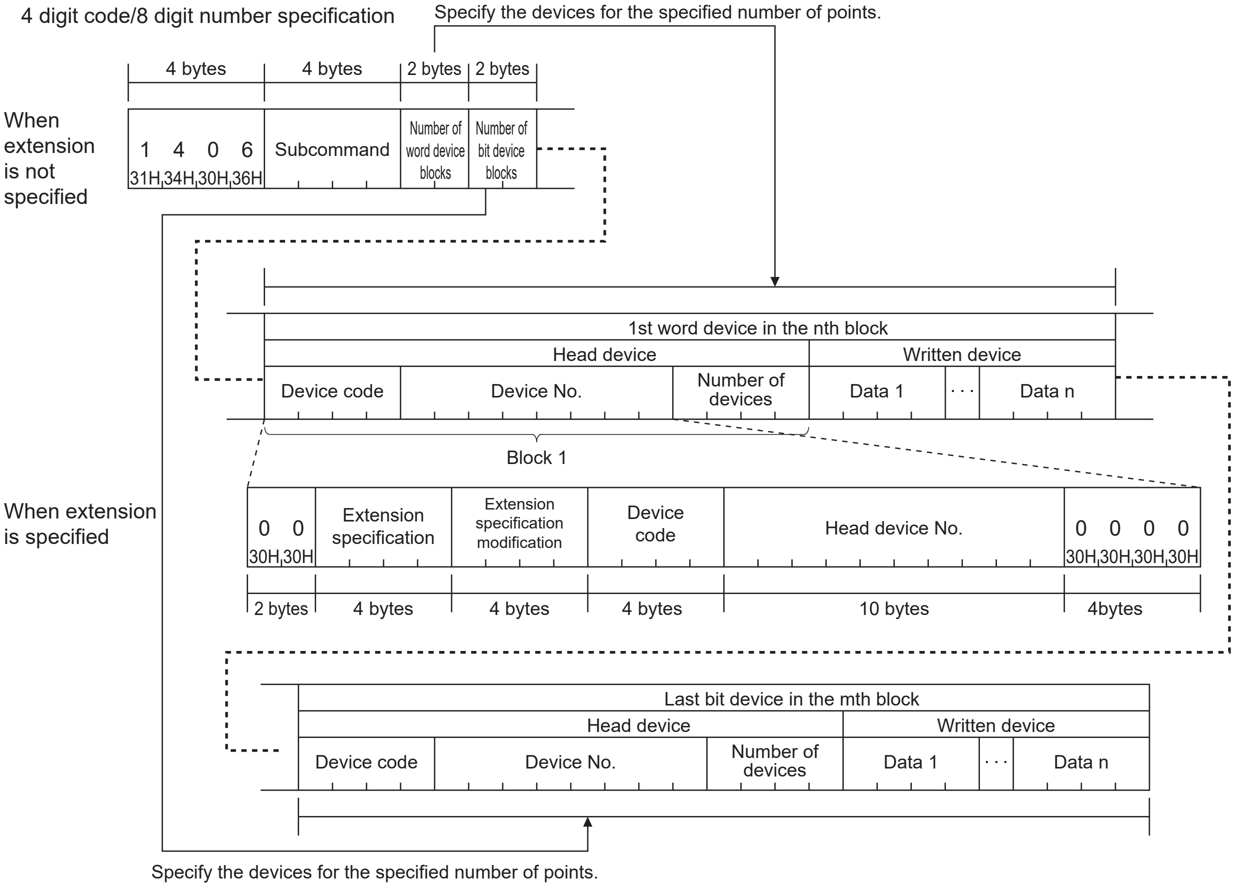
Request data

■When communicating data in ASCII code

2 digit code/6 digit number specification

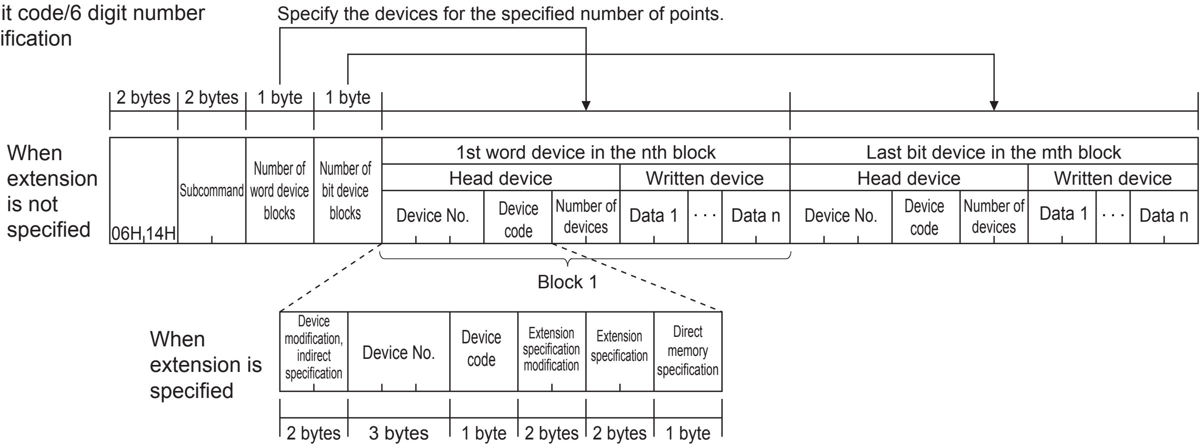


4 digit code/8 digit number specification

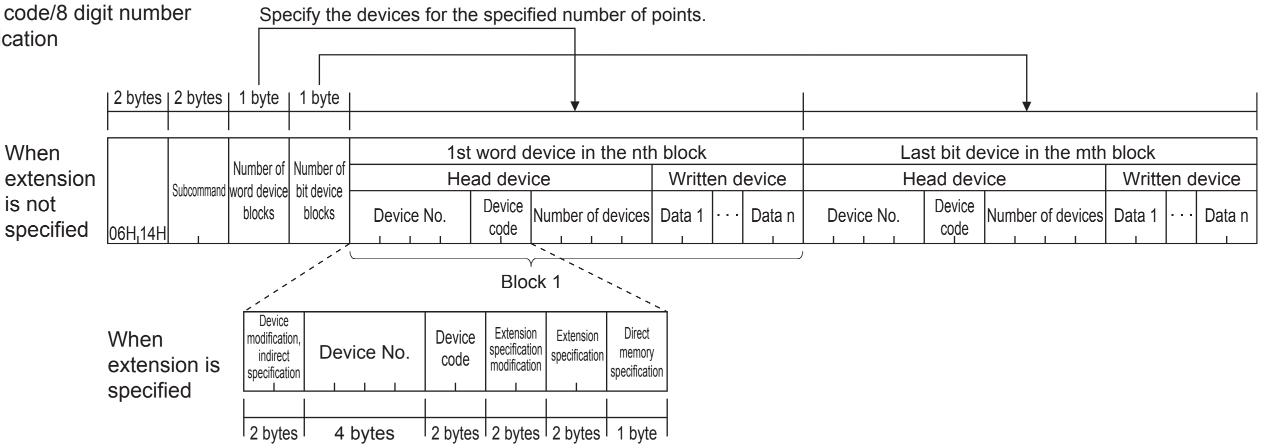


■When communicating data in binary code

2 digit code/6 digit number specification



4 digit code/8 digit number specification



❶Subcommand

Specify the subcommand selected from the item.

Item			Subcommand					
Data size specification	**Device specification format**	**Device memory extension specification**	**ASCII code (Upper column: characters, lower column: character code)**				**Binary code**	
Word units	2 digit code/6 digit number specification	Not specified	0	0	0	0	00H	00H
			30H	30H	30H	30H		
		Specified	0	0	8	0	80H	00H
			30H	30H	38H	30H		
	4 digit code/8 digit number specification	Specified	0	0	8	2	82H	00H
			30H	30H	38H	32H		

❷Number of word device blocks and number of bit device blocks

Specify the number of blocks of the device to be written in hexadecimal.

Item	Description	Number of points	
		ASCII code	Binary code
Number of word device blocks	Specify the number of blocks of the word device to be written.	Number of word device blocks + Number of bit device blocks ≤ 120 When device memory extension specification is used, double the number of the block points.	
Number of bit device blocks	Specify the number of blocks of the bit device to be written.		

❸Device code, device No., number of device points

Specify the device points while satisfying the following conditions:

(Number of word device blocks + number of bit device blocks) × 4 + Total number of points for all word device blocks + total number of points for all bit device blocks ≤ 760

Item	Description
Word device	Specify the device of the points specified in "Number of word device blocks". When "Number of word device blocks" is set to 0, this specification is unnecessary.
Bit device	Specify the device of the points specified in "Number of bit device blocks". When "Number of bit device blocks" is set to 0, this specification is unnecessary.

Point

When specifying a contact or coil of a timer, retentive timer, or counter, use the bit device block.

Set up in order of word device → bit device.

Response data

There is no response data for the batch write multiple blocks command.

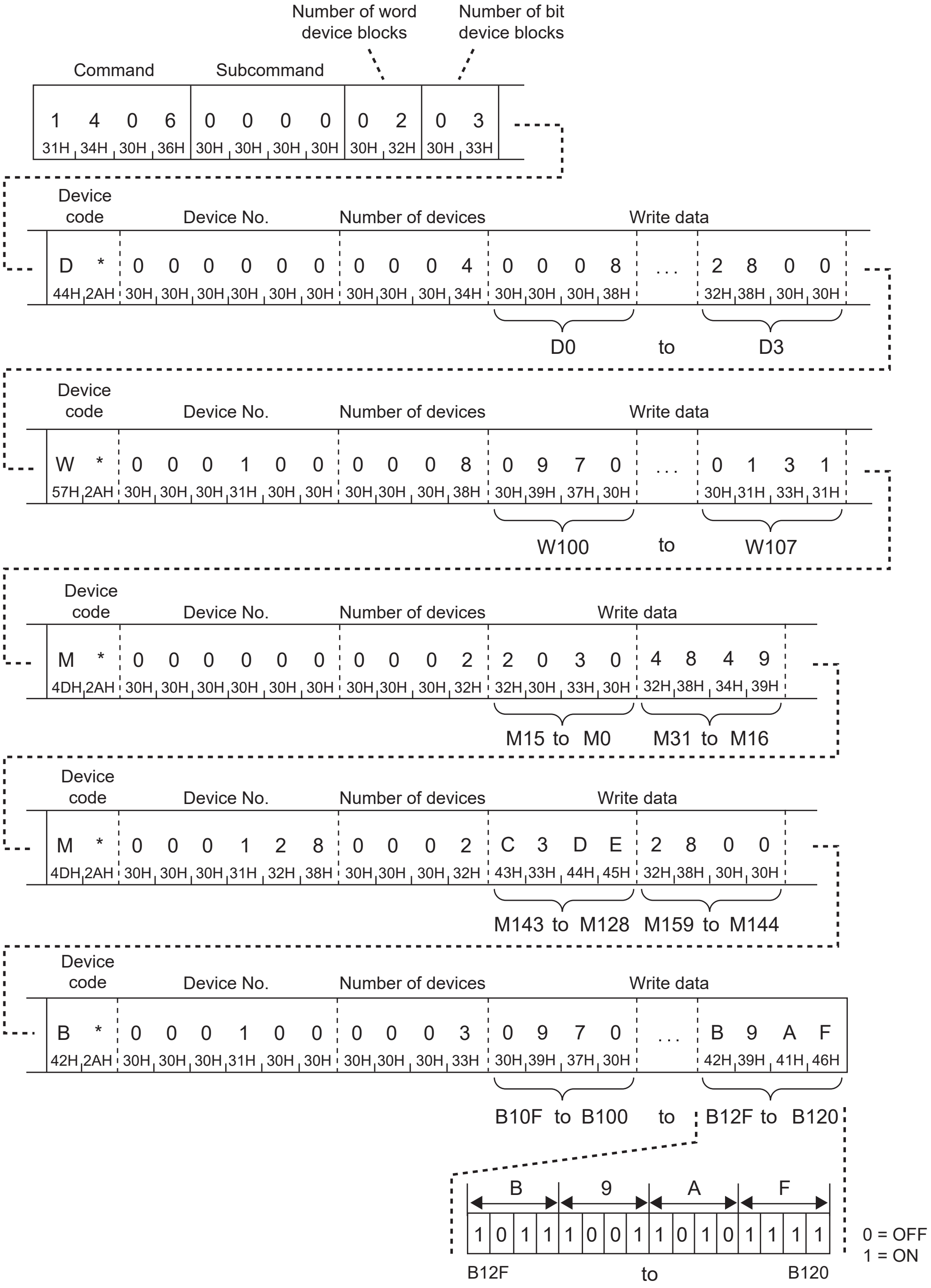
Communication example

Write values from devices as follows.

Item	Write contents
Word device	• Block 1: D0 to D3 (4 points) • Block 2: W100 to W107 (8 points)
Bit device	• Block 1: M0 to M31 (2 points) • Block 2: M128 to M159 (2 points) • Block 3: B100 to B12F (3 points)

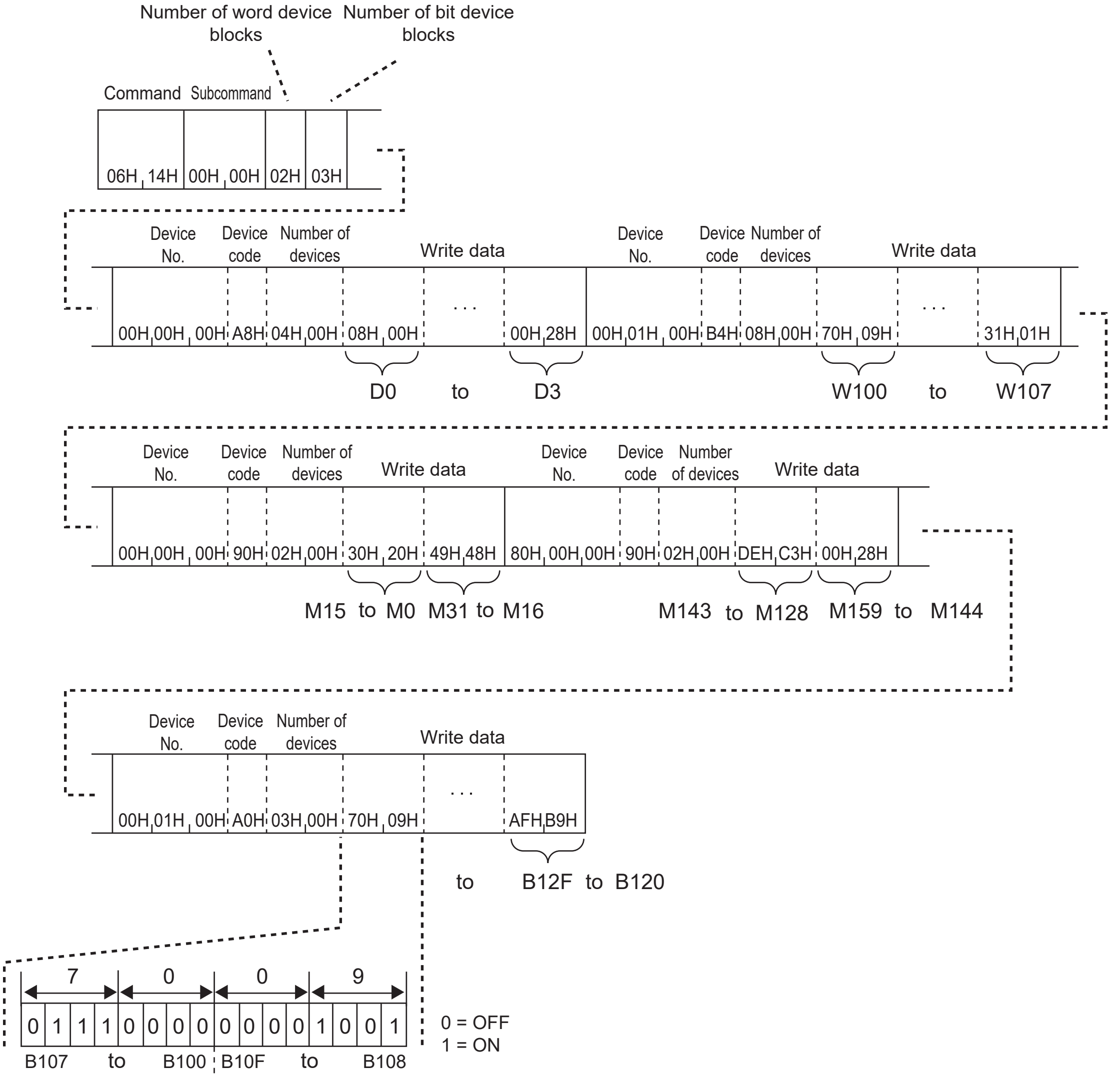
■When communicating data in ASCII code

(Request data)



■When communicating data in binary code

(Request data)



43.3 Remote Control

This section describes the command to set the MC protocol compatible device or CPU module to the RUN status or STOP status by a message from the external device.

Before the remote operation

Access destination is turned OFF → ON or system is reset after remote operation

The information about the remote operation will be deleted.

Ex.

Even if the Remote STOP is executed when the switch of the CPU module is in the RUN status, the operation will return to the RUN status after resetting the module.

When a remote password of the CPU module of the access destination is enabled

Remote operation from the external device is not available. An error will occur at the access destination, and an abnormal response will be sent back to the external device. Unlock the remote password of the CPU module side, and resend the request message.

Operable station in one command

Only one station can be operated remotely by one command.

Remote RUN

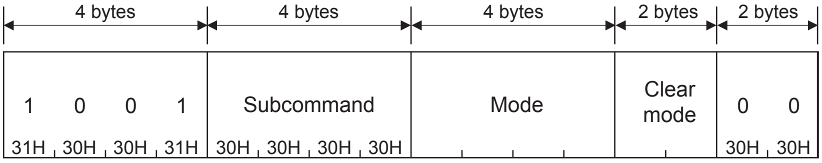
This command executes Remote RUN to the access destination module.

Point

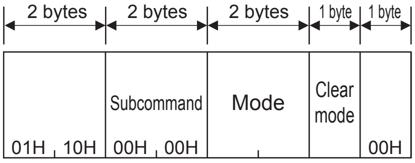
Remote RUN can be executed when the switch of the access destination module is in the RUN status. Even if the switch is in the STOP status, Remote RUN (command: 1001H) will be completed normally. However, the access destination does not change to the RUN status.

Request data

■When communicating data in ASCII code



■When communicating data in binary code



■Mode

This mode specifies whether Remote RUN can be executed forcibly by a device other than the external device which performed Remote STOP or Remote PAUSE. If forced execution is not allowed, Remote RUN can be executed only by the external device which performed Remote STOP or Remote PAUSE.

Forced execution is used when the external device which performed the remote operation cannot execute Remote RUN because of a problem with the device.

Item	**Mode**	
	ASCII code	**Binary code**
Forced execution not allowed (Remote RUN cannot be executed when other device executes Remote STOP or Remote PAUSE.)	0 0 0 1 30H, 30H, 30H, 31H	01H, 00H
Forced execution allowed (Remote RUN can be executed when other device executes Remote STOP or Remote PAUSE.)	0 0 0 3 30H, 30H, 30H, 33H	03H, 00H

■Clear mode

This mode specifies whether the clear (initialization) processing of device is executed when starting the calculation for the Remote RUN.

Only 00H is valid.

Item	**Mode**	
	ASCII code	**Binary code**
Do not clear device	0 0 30H, 30H	00H

Response data

There is no response data for the Remote RUN command.

Communication example

Set mode to "Forced execution not allowed", and set clear mode to "Clear all devices including that in the latch range" when executing Remote RUN.

- When communicating data in ASCII code

(Request data)

			Mode	Clear mode
1 0 0 1 31H, 30H, 30H, 31H	0 0 0 0 30H, 30H, 30H, 30H	0 0 0 1 30H, 30H, 30H, 31H	0 2 30H, 32H	0 0 30H, 30H

- When communicating data in binary code

(Request data)

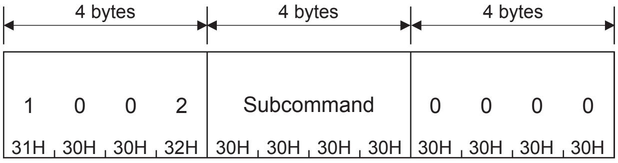
		Mode	Clear mode	
01H, 10H	00H, 00H	01H, 00H	02H	00H

Remote STOP

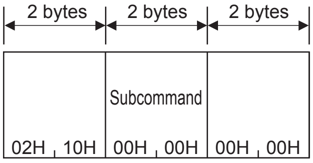
This command executes Remote STOP to the access destination module.

Request data

■When communicating data in ASCII code



■When communicating data in binary code



Response data

There is no response data for the Remote STOP command.

Communication example

Send request messages from the external device by using the message format shown in the request data above.

Remote PAUSE

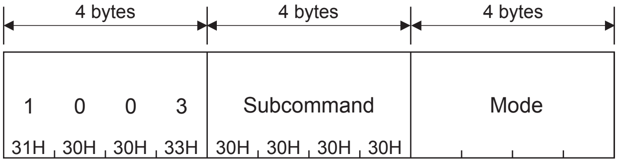
This command executes Remote PAUSE to the access destination module.

Point

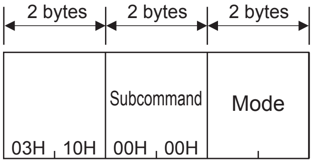
Remote PAUSE can be executed when the switch of the access destination module is in the RUN status. Even if the switch is in the STOP status, Remote PAUSE (command: 1003H) will be completed normally. However, the access destination does not change to the PAUSE status.

Request data

■When communicating data in ASCII code



■When communicating data in binary code



■Mode

This mode specifies whether Remote PAUSE can be executed forcibly by a device other than the external device which performed Remote STOP or Remote PAUSE. If forced execution is not allowed, Remote PAUSE can be executed only by the external device which performed Remote STOP or Remote PAUSE.

Forced execution is used when the external device which performed the remote operation cannot execute Remote PAUSE because of a problem with the device.

Item	Mode	
	ASCII code	Binary code
Forced execution not allowed (Remote PAUSE cannot be executed when other device executes Remote STOP or Remote PAUSE.)	0 0 0 1 30H,30H,30H,31H	01H,00H
Forced execution allowed (Remote PAUSE can be executed when other device executes Remote STOP or Remote PAUSE.)	0 0 0 3 30H,30H,30H,33H	03H,00H

Response data

There is no response data for the Remote PAUSE command.

Communication example

Set mode to "Forced execution not allowed" when executing Remote PAUSE.

■When communicating data in ASCII code

(Request data)

		Mode
1 0 0 3 31H,30H,30H,33H	0 0 0 0 30H,30H,30H,30H	0 0 0 1 30H,30H,30H,31H

■When communicating data in binary code

(Request data)

		Mode
03H,10H	00H,00H	01H,00H

Remote latch clear

This command executes remote latch clear to the access destination module.

Point

Before executing the remote latch clear, set the status of the access destination module to STOP.

While the access destination is stopped or paused remotely by request from another external device:

- The remote latch clear cannot be executed. Abnormal completion of the command will occur.
- Cancel the Remote STOP or Remote PAUSE before executing the command.

Request data

■When communicating data in ASCII code

4 bytes	4 bytes	4 bytes
1 0 0 5 31H,30H,30H,35H	Subcommand 30H,30H,30H,30H	0 0 0 0 30H,30H,30H,30H

■When communicating data in binary code

2 bytes	2 bytes	2 bytes
	Subcommand	
05H, 10H	00H, 00H	00H, 00H

Response data

There is no response data for remote latch clear command.

Communication example

Send request messages from the external device by using the message format shown in the request data above.

Remote RESET

This command executes Remote RESET to the access destination module. Remote RESET is used to restore when an error occurred in the MC protocol compatible device.

Point

Before executing Remote RESET, perform the following.

- When the access destination module has a Remote RESET enable/disable setting, go to GX Works3 Navigation window ⇨ [Parameter] ⇨ Module model ⇨ [CPU Parameter] ⇨ [Operation Related Setting] ⇨ [Remote Reset Setting], and select "Enable" for "Remote Reset". (Default: Disable)
- Set the status of the access destination module to STOP.

Precautions

- In some cases, Remote RESET cannot be executed because of hardware error, etc.
- The response message when Remote RESET is executed may not be sent back to the external device since the access destination is reset.

Request data

■When communicating data in ASCII code

4 bytes	4 bytes	4 bytes
1 0 0 6	Subcommand	0 0 0 0
31H, 30H, 30H, 36H	30H, 30H, 30H, 30H	30H, 30H, 30H, 30H

■When communicating data in binary code

2 bytes	2 bytes	2 bytes
	Subcommand	
06H, 10H	00H, 00H	00H, 00H

Response data

There is no response data for the Remote RESET command.

Communication example

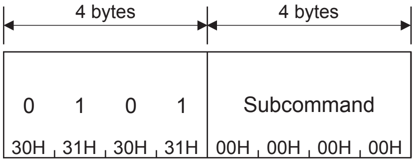
Send request messages from the external device by using the message format shown in the request data above.

Read CPU model name

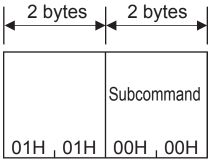
This command reads the processor module name code (processor type) of the access destination module.

Request data

■When communicating data in ASCII code

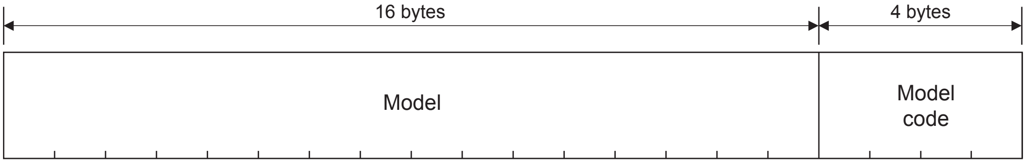


■When communicating data in binary code

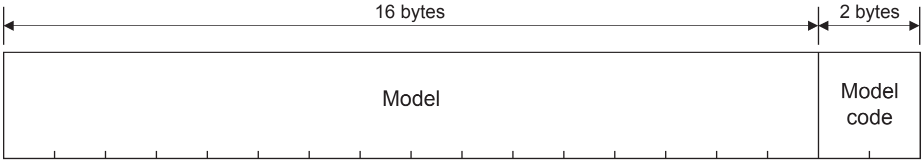


Response data

■When communicating data in ASCII code



■When communicating data in binary code



■Model

The characters of the module model are stored for 16 characters from the upper byte.

If the model to be read is less than 16 characters, space (20H) is stored for the remaining characters. Even when communicating data in binary code, the module model is stored in ASCII code.

■Model code

The following model codes will be stored.

When communicating in ASCII code, the data is stored in order from the upper byte to the lower byte.

When communicating in binary code, the data is stored in order from the lower byte to the upper byte.

Model	Model code (hexadecimal)
FX5U-32MR/ES	4A21H
FX5U-64MR/ES	4A23H
FX5U-80MR/ES	4A24H
FX5U-32MT/ES	4A29H
FX5U-64MT/ES	4A2BH
FX5U-80MT/ES	4A2CH
FX5U-32MT/ESS	4A31H
FX5U-64MT/ESS	4A33H
FX5U-80MT/ESS	4A34H
FX5U-32MR/DS	4A41H
FX5U-64MR/DS	4A43H
FX5U-80MR/DS	4A44H
FX5U-32MT/DS	4A49H
FX5U-64MT/DS	4A4BH
FX5U-80MT/DS	4A4CH
FX5U-32MT/DSS	4A51H
FX5U-64MT/DSS	4A53H
FX5U-80MT/DSS	4A54H
FX5UC-32MT/D	4A91H
FX5UC-64MT/D	4A92H
FX5UC-96MT/D	4A93H
FX5UC-32MT/DSS	4A99H
FX5UC-64MT/DSS	4A9AH
FX5UC-96MT/DSS	4A9BH
FX5UC-32MR/DS-TS	4AA9H
FX5UC-32MT/DS-TS	4AB1H
FX5UC-32MT/DSS-TS	4AB9H
FX5UJ-24MR/ES	4B0DH
FX5UJ-40MR/ES	4B0EH
FX5UJ-60MR/ES	4B0FH
FX5UJ-24MT/ES	4B14H
FX5UJ-40MT/ES	4B15H
FX5UJ-60MT/ES	4B16H
FX5UJ-24MT/ESS	4B1BH
FX5UJ-40MT/ESS	4B1CH
FX5UJ-60MT/ESS	4B1DH
FX5UJ-24MR/DS	4B22H
FX5UJ-40MR/DS	4B23H
FX5UJ-60MR/DS	4B24H
FX5UJ-24MT/DS	4B29H
FX5UJ-40MT/DS	4B2AH
FX5UJ-60MT/DS	4B2BH
FX5UJ-24MT/DSS	4B30H
FX5UJ-40MT/DSS	4B31H
FX5UJ-60MT/DSS	4B32H
FX5S-30MR/ES	4B4EH
FX5S-40MR/ES	4B4FH
FX5S-60MR/ES	4B50H

Model	Model code (hexadecimal)
FX5S-80MR/ES[*1]	4B51H
FX5S-30MT/ES	4B55H
FX5S-40MT/ES	4B56H
FX5S-60MT/ES	4B57H
FX5S-80MT/ES[*1]	4B58H
FX5S-30MT/ESS	4B5CH
FX5S-40MT/ESS	4B5DH
FX5S-60MT/ESS	4B5EH
FX5S-80MT/ESS[*1]	4B5FH
FX5S-30MR/DS	4B63H
FX5S-40MR/DS	4B64H
FX5S-60MR/DS	4B65H
FX5S-80MR/DS[*1]	4B66H
FX5S-30MT/DS	4B6AH
FX5S-40MT/DS	4B6BH
FX5S-60MT/DS	4B6CH
FX5S-80MT/DS[*1]	4B6DH
FX5S-30MT/DSS	4B71H
FX5S-40MT/DSS	4B72H
FX5S-60MT/DSS	4B73H
FX5S-80MT/DSS[*1]	4B74H

[*1] Area-specific model

Point

The model of the CPU module is identified by the model code.

Communication example

■When communicating data in ASCII code

(Request data)

0 1 0 1	0 0 0 0
30H,31H,30H,31H	30H,30H,30H,30H

(Response data)

F X 5 U - 3 2 M R / E S	4 A 2 1
46H,58H,35H,55H,2DH,33H,32H,4DH,52H,2FH,45H,53H,20H,20H,20H,20H	34H,41H,32H,31H

■When communicating data in binary code

(Request data)

01H,01H	00H,00H

(Response data)

F X 5 U - 3 2 M R / E S	
46H,58H,35H,55H,2DH,33H,32H,4DH,52H,2FH,45H,53H,20H,20H,20H,20H	21H,4AH

43.4 Clear Error

This function turns off ERR LED of the FX5CPU from the external device and/or initializes the communication error information or error code stored in the buffer memory.

This function is used to initialize the current error information due to an abnormal response for a request message and return it to the normal state or initialize the error code storage area of the buffer memory.

The order and description of the data item differ depending on the frame and pattern in a communication.

Point

This function can be used only for the FX5CPU which is connected with the external device.

This function cannot be used for the FX5CPU of another station via the network system.

The data part of the command and control procedure when the display LEDs of the CPU module are turned off and the communication error information is initialized from the external device is described.

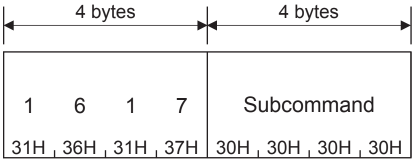
Command

○: The function can be executed.

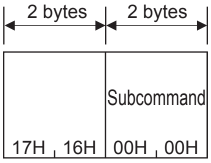
Function	Command (Subcommand)	Contents of processing	CPU module status		
			STOP	RUN	
				Write allow setting	Write prohibit setting
LED OFF, error code initialization	1617 (0000)	Turns off the display LEDs, initializes the error code, and others.	○	○	○

Request data

■When communicating data in ASCII code



■When communicating data in binary code



Response data

There is no response data for the Clear Error command.

Communication example

Send request messages from the external device by using the message format shown in the request data above.

43.5 Loopback Test

This function tests whether the communication function between the external device and FX5CPU operates normally or not. The control procedure when this function is used is described with examples.

Point

- At the startup of the FX5CPU or when trouble occurs, this function can check whether the connection between the external device and FX5CPU is correct and/or whether the data communication function operates normally.
- This function can be used only for the FX5CPU which is connected with the external device (including a multi-drop connection station). This function cannot be used for the FX5CPU of another station via the network system.

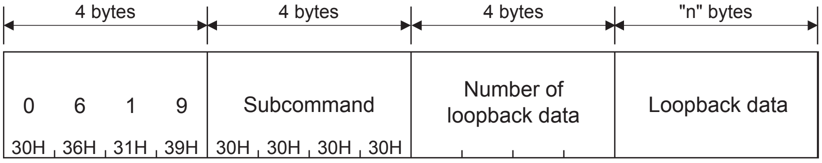
Command

○: The function can be executed.

Function	Command (Subcommand)	Contents of processing	CPU module status		
			STOP	RUN	
				Write allow setting	Write prohibit setting
Loopback test	0619 (0000)	Checks whether data communication is executed normally.	○	○	○

Request data

■When communicating data in ASCII code



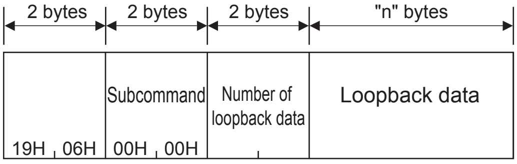
- Number of loopback data (number of bytes)

The number of the bytes is converted into a 4-digit ASCII code (hexadecimal) and data is sent from the upper digit ("0").

- Loopback data

The order of character strings for up to 960 1-byte characters ("0" to "9", "A" to "F") is sent from the head.

■When communicating data in binary code



- Number of loopback data (number of bytes)

The 2-byte numerical value which indicates the number of the bytes is used and data is sent from the low byte (L: bit 0 to 7).

- Loopback data

Data is sent for up to 960 bytes from the head by treating each character code ("0" to "9", "A" to "F") as a 1 byte value.

Response data

The same number of the loopback data and loopback data which the external device sent are sent back to the external device.

Communication example

Send request messages from the external device by using the message format shown in the request data (☞ Page 767 Request data).

The following shows an example when loopback data is "abcdefghijkl".

■When executing the Self-Test by communicating in ASCII code

(Request data)

Command	Subcommand	Number of loopback data	Loopback data
0 6 1 9	0 0 0 0	0 0 1 2	a b c d e f g h i j k l
30H 36H 31H 39H	30H 30H 30H 30H	30H 30H 31H 32H	61H 62H 63H 64H 65H 66H 67H 68H 69H 6AH 6BH 6CH

(Response data)

Number of loopback data	Loopback data
0 0 1 2	a b c d e f g h i j k l
30H 30H 31H 32H	61H 62H 63H 64H 65H 66H 67H 68H 69H 6AH 6BH 6CH

■When executing the Self-Test by communicating in binary code

(Request data)

Command	Subcommand	Number of loopback data	Loopback data
			a b c d e f g h i j k l
19H 06H	00H 00H	12H 00H	61H 62H 63H 64H 65H 66H 67H 68H 69H 6AH 6BH 6CH

(Response data)

Number of loopback data	Loopback data
	a b c d e f g h i j k l
12H 00H	61H 62H 63H 64H 65H 66H 67H 68H 69H 6AH 6BH 6CH

44 COMMUNICATING USING 1C FRAMES

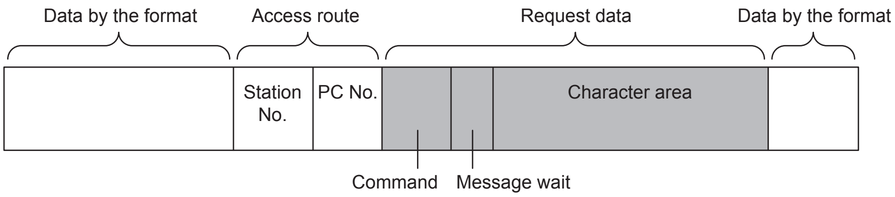
This chapter explains the functions when accessing using 1C frame and their message format.
1C frame is compatible with the communication function of the dedicated protocols supported by computer link of FX3 and A series computer link modules.
Only the commands for 1C frame explained in this chapter can be used for 1C frame.

44.1 Message Format

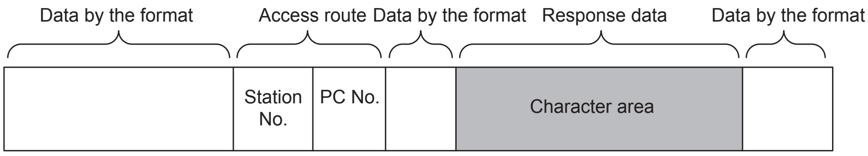
This section explains the message format when communicating data using 1C frame.

Message format

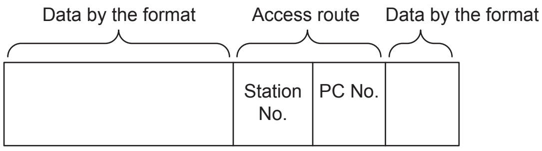
■Request message



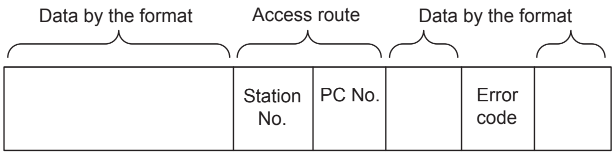
■Response message (Normal completion: Response data)



■Response message (Normal completion: No response data)



■Response message (Abnormal completion)



Setting data

Set the following items.

Item		Description	Reference
Data by format		The message formats differ depending on the set format (Format 1, Format 4).	Page 693 Message Formats of Each Format
Access route	Station No.	Specify the station to be connected from an external device.	Page 705 Station No.
	PC No.	Specify the network station No. of the access target.	Page 706 Network No., PC No.
Request data	Command	Specify the function to request such as read or write.	Page 771 Command
	Message wait	A data to generate a delay time for response transmission. Specify the wait time within the range of 0 to 150ms in 10ms units.	Page 771 Message wait
	Character area	A data that instructs the FX5 CPU module to execute a request specified by command. The content of the character areas differs depending on the command.	Page 772 Character area
Response data	Character area	A data that FX5 CPU module returns to a request specified by a command. The content of the character areas differs depending on the command.	
Error code		Error code indicates the content of occurred error.	Page 773 Error code

44.2 Details of Setting Data

This section explains how to specify the common data items and their content in each message.

Command

Set the command type. (☞ Page 774 Command and Function Lists for 1C Frame)

The setting values for each command are as follows.

Function	Command		Reference
	Symbol	ASCII code	
Device memory read and write	BR	42H, 52H	Page 778 Batch read (bit units) (command: BR)
	WR	57H, 52H	Page 779 Batch read (word units) (command: WR, QR)
	QR	51H, 52H	
	BW	42H, 57H	Page 781 Batch write (bit units) (command: BW)
	WW	57H, 57H	Page 782 Batch write (word units) (command: WW, QW)
	QW	51H, 57H	
	BT	42H, 54H	Page 784 Test (random write) (bit units) (command: BT)
	WT	57H, 54H	Page 785 Test (random write) (word units) (command: WT, QT)
	QT	51H, 54H	
Remote RUN, Remote STOP	RR	52H, 52H	Page 787 Remote RUN, remote STOP (command: RR, RS)
	RS	52H, 53H	
Read CPU model name	PC	50H, 43H	Page 788 Read CPU model name (command: PC)
Global	GW	47H, 57H	Page 790 Global signal ON/OFF (command: GW)
Loopback test	TT	54H, 54H	Page 791 Loopback test (Command: TT)

Setting method

Use the commands by converting to 2-digit (hexadecimal) ASCII codes.

Ex.

Device memory batch read (BR) in bit unit

Message wait

Message wait is a data to generate a delay time for response transmission.

Some external devices may take time to become receiving status after sending a command.

Specify the minimum wait time to send the result after FX5 CPU module is received a command from an external device.

Specify the wait time in accordance with the specifications of the external device.

Setting method

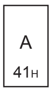
Specify the wait time within the range of 0 to 150ms in 10ms units.

Convert 0H to FH (0 to 15) to 1-digit (hexadecimal) ASCII codes regarding 10ms as 1H.

Ex.

When the message wait time is 100 ms

If the following value is set to message wait in request message, after passing 100 ms or more, transmission of a response message will be started.



Character area

The content of the character areas differs depending on the command.

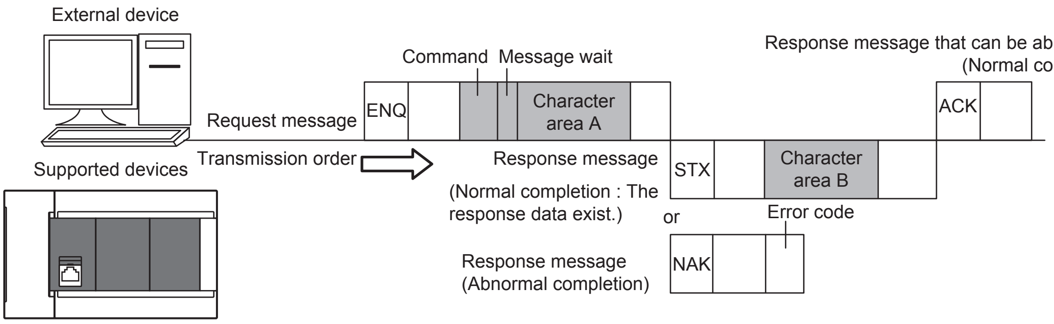
The character area of request data is equivalent to the character A area and the character C area of the dedicated protocols for computer link of FX3 and A series computer link module. The character are of response data is equivalent to the character B area of a dedicated protocols.

- Character area A: Data to instruct the FX5 CPU module to perform the read request specified by a command.
- Character area B: Data that the FX5 CPU module returns to a request specified by a command.
- Character area C: Data to instruct the FX5 CPU module to perform the write request specified by a command.

When reading data (Response data)

The following shows the image when the response data (character B area of the dedicated protocol) is included in the response message.

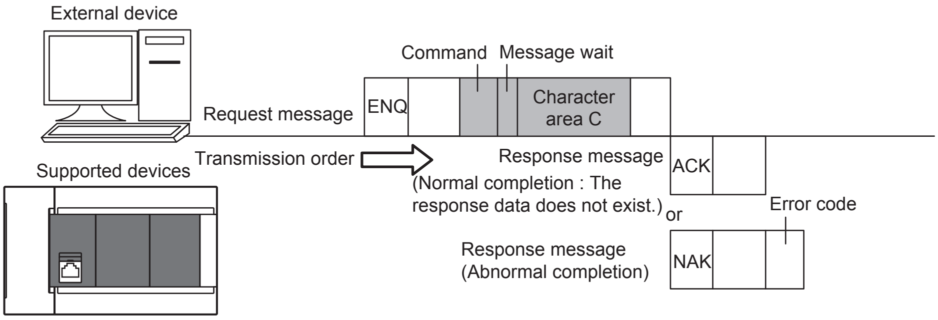
(The head of the message data in the figure is a control code of format 1. ☞ Page 693 Message Formats of Each Format)



When writing data (No response data)

The following shows the image when the response data is not included in the response message.

(The head of the message data in the figure is a control code of format 1. ☞ Page 693 Message Formats of Each Format)



Error code

An error code indicates the content of error.

If more than one error occurs at the same time, the error code detected first is returned.

For the content of error code and its corrective action, refer to the following.

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

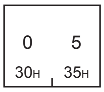
☞ Page 836 Checking error codes of MC protocol function

Setting method

Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.

Ex.

For error code 05H



44.3 Command and Function Lists for 1C Frame

Use the following commands for data communication using 1C frame.

Name			Command		Contents of processing	Number of points processed in onetime update
			Symbol	ASCII code		
Device memory	Batch reading	Bit units	BR	42H, 52H	Reads data in 1-point units from bit devices.	256 points
		Word units	WR	57H, 52H	Reads data in 16-point units from bit devices.	32 words (512 points)
					Reads data in 1-word unit from word devices.	64 points
			QR	51H, 52H	Reads data in 16-point units from bit devices.	32 words (512 points)
					Reads data in 1-word unit from word devices.	64 points
	Batch writing	Bit units	BW	42H, 57H	Writes data in 1-point units to bit devices.	160 points
		Word units	WW	57H, 57H	Writes data in 16-points units to bit devices.	10 words (160 points)
					Writes data in 1-point units to word devices.	64 points
			QW	51H, 57H	Writes data in 16-points units to bit devices.	10 words (160 points)
					Writes data in 1-point units to word devices.	64 points
	Test (random write)	Bit units	BT	42H, 54H	Writes data in 1-bit unit to a bit device by randomly specifying the device number.	20 points
		Word units	WT	57H, 54H	Writes data in 16-bit unit to a bit device by randomly specifying the device number.	10 words (160 points)
					Writes data in 1-word unit to a word device by randomly specifying the device number.	10 points
			QT	51H, 54H	Writes data in 16-bit unit to a bit device by randomly specifying the device number.	10 words (160 points)
					Writes data in 1-word unit to a word device by randomly specifying the device number.	10 points
PLC	Remote RUN		RR	52H, 52H	Requests remote RUN to a device.	—
	Remote STOP		RS	52H, 53H	Requests remote STOP to a device.	
	CPU model name reading		PC	50H, 43H	Reads PLC model name.	—
Global			GW	47H, 57H	Turns global signal ON or OFF (SM8126 in FX5) in all PLCs connected in computer link.	1 point
Loopback test			TT	54H, 54H	Returns received characters back to the computer as they are.	254 bytes

Command

Command is a communication function issued by MC protocol, and the command is accessible.

When accessing modules other than FX5 CPU module, there is a restriction for the accessible device range.

☞ Page 775 Considerations when accessing devices other than FX5 CPU module

44.4 Device Memory Read and Write

This section explains the specification content and examples of the control procedure when reading from/writing to the device memory are as shown below.

For the message formats other than request data and response data, refer to the following sections.

☞ Page 769 Message Format

☞ Page 771 Details of Setting Data

Considerations

The considerations when reading/writing device memory using the commands described in this section.

Considerations when accessing devices other than FX5 CPU module

■Accessible devices

Only the devices with the same names that exist in FX5 CPU can be accessed within the device range.

☞ Page 776 Accessible device range

■Special relays and special registers

Special relays and special registers can be accessed within the following range.

- Access SM8000 to SM8511 by specifying M8000 to M8511.
- Access SD8000 to SD8511 by specifying D8000 to D8511.
- Access LC0 to LC55 by specifying CS200 to CS255 or CN200 to CN255.

Data to be specified in command

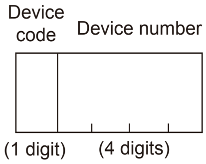
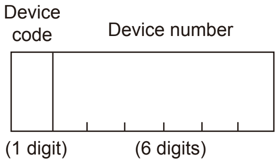
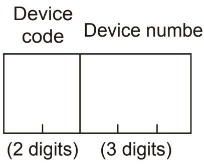
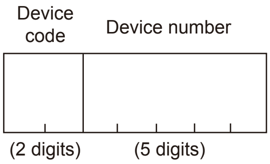
Device codes, device numbers

The settings of each device when reading/writing device memory can be performed using device code and device number as shown in the following figure.

Specify the device to be accessed by a device code and a device number.

The data size to be set differs depending on the command.

The setting data size differ when the device type is timer or counter.

Device type	BR, BW, BT, WR, WW, WT command	QR, QW, QT command
Other than timer and counter		
Timer, counter		

■BR, BW, BT, WR, WW, WT commands

- Device code: Convert the device name to 1-digit ASCII code (2-digits for timer or counter), and send it from the upper digits.
- Device number: Convert the numerical value to 4-digit ASCII code (3-digits for timer or counter), and send it from the upper digits.

■QR, QW, QT command

- Device code: Convert the device name to 1-digit ASCII code (2-digits for timer or counter), and send it from the upper digits.
- Device number: Convert the numerical value to 6-digit ASCII code (5-digits for timer or counter), and send it from the upper digits.

Ex.

Current value of input (X) 40 and timer (T) 10

Device	BR, BW, BT, WR, WW, WT command	QR, QW, QT command
Input (X) 40	X 0 0 4 0 58_H 30_H 30_H 34_H 30_H	X 0 0 0 0 4 0 58_H 30_H 30_H 30_H 30_H 34_H 30_H
Current value of timer (T) 10	T S 0 1 0 54_H $4E_H$ 30_H 31_H 30_H	T S 0 0 0 1 0 54_H $4E_H$ 30_H 30_H 30_H 31_H 30_H

Accessible device range

The following table shows the devices and device number range that can be specified when accessing the device memory. Access the CPU module within the range of device number that can be used by commands and the range of the device number that can be used in the access target CPU. (☞ Page 775 Considerations when accessing devices other than FX5 CPU module)

Device		Type	Device code	Device number	Representation	Available commands
Input relay		Bit	X	X0000 to X0377	Octal/Hexadecimal[1]	BR, BW, BT, WR, WW, WT
				X000000 to X000377	Octal/Hexadecimal[1]	QR, QW, QT
Output relay		Bit	Y	Y0000 to Y0377	Octal/Hexadecimal[1]	BR, BW, BT, WR, WW, WT
				Y000000 to Y000377	Octal/Hexadecimal[1]	QR, QW, QT
Internal relay		Bit	M	M0000 to M7679	Decimal	BR, BW, BT, WR, WW, WT
				M000000 to M007679	Decimal	QR, QW, QT
Step relay		Bit	S	S0000 to S4095	Decimal	BR, BW, BT, WR, WW, WT
				S000000 to S004095	Decimal	QR, QW, QT
Timer	Contact	Bit	TS	TS000 to TS511	Decimal	BR, BW, BT, WR, WW, WT
				TS00000 to TS00511	Decimal	QR, QW, QT
	Present value	Word	TN	TN000 to TN511	Decimal	WR, WW, WT
				TN00000 to TN00511	Decimal	QR, QW, QT
Counter	Contact	Bit	CS[2]	CS000 to CS255	Decimal	BR, BW, BT, WR, WW, WT
				CS00000 to CS00255	Decimal	QR, QW, QT
	Present value	Word	CN[3]	CN000 to CN255	Decimal	WR, WW, WT
				CN00000 to CN00255	Decimal	QR, QW, QT
Data register		Word	D	D0000 to D7999	Decimal	WR, WW, WT
				D000000 to D007999	Decimal	QR, QW, QT
Extended register		Word	R	R0000 to R9999	Decimal	WR, WW, WT
				R000000 to R032767	Decimal	QR, QW, QT
Special relay		Bit	M[4]	M8000 to M8511	Decimal	BR, BW, BT, WR, WW, WT
				M008000 to M008511	Decimal	QR, QW, QT
Special Register		Word	D[5]	D8000 to D8511	Decimal	WR, WW, WT
				D008000 to D008511	Decimal	QR, QW, QT

[1] It depends on the message format. See below.
Format 1 (X, Y OCT), format 4 (X, Y OCT): octal
Format 1 (X, Y HEX), format 4 (X, Y HEX): hexadecimal

*2 Access LCS0 to LCS55 by specifying CS200 to CS255 (CS00200 to CS00255).
*3 Access LCN0 to LCN55 by specifying CN200 to CN255 (CN00200 to CN00255).
*4 Access SM8000 to SM8511 by specifying M8000 to M8511 (M008000 to M008511).
*5 Access SD8000 to SD8511 by specifying D8000 to D8511 (D008000 to D008511).

Precautions

- When using the command specifying the word unit (WR, WW, WT, QR, QW, QT), make sure to set a multiple of 8 as the head device address of the bit device.
- Special relays and special registers are divided into three, the ones for read only, write only, and the system. If the write is performed outside the writable range, an error may occur in the PLC. For details on the special relays and special registers, refer to 🕮MELSEC iQ-F FX5 User's Manual (Application).
- Do not set the setting which overlaps the data register and the special register.
- In the FX5 CPU module, even if the device value is changed, the device initial value is not changed.

Number of device points

Specify the number of device points to be read or written.

Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.

Specify the number of device points in one command within the device points that can be processed in one communication.

☞ Page 774 Command and Function Lists for 1C Frame

Specify '00' for 256 points.

Ex.

5 points, 20 points, 256 points

Number of device points	ASCII code
5 points	0 5 30_H 35_H
20 points	1 4 31_H 34_H
256 points	0 0 30_H 30_H

Read data, write data

The data storage method is the same as reading/writing data with device access of 4C/3C frame. (☞ Page 720 Read data, write data)

Batch read (bit units) (command: BR)

Reads bit devices in batch.

Message format

The following shows the message format of the request data and response data of the command.

■Request data

Command	Message wait	Head device	Number of device points

■Response data

The value of read device is stored in bit units. (☞ Page 720 Read data, write data)

Data specified by request data

■Command

BR command

B	R
42_H	52_H

■Message wait

Specify the delayed time of the response transmission. (☞ Page 771 Message wait)

■Head device

Specify the head device. (☞ Page 775 Device codes, device numbers)

■Number of device points

Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.

Specify the number of device points within the following range:

- 1 ≤ Number of device points ≤ 256 (for 256 points, specify '00H')
- Head device No. + Number of device points -1 ≤ Maximum device No.

Communication example

Read data in bit units under the following conditions.

- Message wait: 100ms
- Head device: X40
- Number of device points: 5 points

(Request data)

■When using BR

		Message wait	Head device					Number of device points	
B	R	A	X	0	0	4	0	0	5
42_H	52_H	41_H	58_H	30_H	30_H	34_H	30_H	30_H	35_H

(Response data)

Data read

0	1	1	0	1
30_H	31_H	31_H	30_H	31_H
(X40)	(X41)	(X42)	(X43)	(X44)

Batch read (word units) (command: WR, QR)

Reads bit devices in 16-point units.

Reads word devices in 1-point units

Message format

The following shows the message format of the request data and response data of the command.

■Request data

Command	Message wait	Head device	Number of device points (Number of words)

■Response data

The value of read device is stored in word units. (☞ Page 720 Read data, write data)

Data specified by request data

■Command

WR command	QR command
W R 57_H 52_H	Q R 51_H 52_H

■Message wait

Specify the delayed time of the response transmission. (☞ Page 771 Message wait)

■Head device

Specify the head device. (☞ Page 775 Device codes, device numbers)

■Number of device points

Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.

Specify the number of device points within the following range:

- For bit device: 1 ≤ Number of device points ≤ 32
- For bit device: Head device No. + Number of device points × 16 - 1 ≤ Maximum device No.
- For word device: 1 ≤ Number of device points ≤ 64
- For word device: Head device No. + Number of device points -1 ≤ Maximum device No.

When reading CN200 to CN255 (CN00200 to CN00255), since 1 device point is 2 words, the number of device points becomes 32 points.

Point

- When specifying bit devices, set the head device No. in multiples of 8.

Communication example (Reading bit device memory)

Read bit devices in 16-point units under the following conditions.

- Message wait: 0ms
- Head device: X40
- Number of device points: 32 points (2 words)

(Request data)

■When using WR

	Message wait	Head device	Number of device points (Number of words)
W R	0	X 0 0 4 0	0 2
57_H 52_H	30_H	58_H 30_H 30_H 34_H 30_H	30_H 32_H

■When using QR

	Message wait	Head device	Number of device points (Number of words)
Q R	0	X 0 0 0 0 4 0	0 2
51_H 52_H	30_H	58_H 30_H 30_H 30_H 30_H 34_H 30_H	30_H 32_H

(Response data)

Data read

0001 (Hexadecimal : 1) 31_H	0010 (Hexadecimal : 2) 32_H	0011 (Hexadecimal : 3) 33_H	0100 (Hexadecimal : 4) 34_H	1010 (Hexadecimal : A) 41_H	1011 (Hexadecimal : B) 42_H	1100 (Hexadecimal : C) 43_H	1101 (Hexadecimal : D) 44_H
(X57) to (X54)	(X53) to (X50)	(X47) to (X44)	(X43) to (X40)	(X77) to (X74)	(X73) to (X70)	(X67) to (X64)	(X63) to (X60)

Communication example (Reading word device memory)

Read word devices in 1-point units under the following conditions.

- Message wait: 0ms
- Head device: Current value of T123
- Number of device points: 2 points (2 words)

(Request data)

■When using WR

	Message wait	Head device	Number of device points (Number of words)
W R	0	T N 1 2 3	0 2
57_H 52_H	30_H	54_H $4E_H$ 31_H 32_H 33_H	30_H 32_H

■When using QR

	Message wait	Head device	Number of device points (Number of words)
Q R	0	T N 0 0 1 2 3	0 2
51_H 52_H	30_H	54_H $4E_H$ 30_H 30_H 31_H 32_H 33_H	30_H 32_H

(Response data)

Data read

7 B C 9	1 2 3 4
37_H 42_H 43_H 39_H	31_H 32_H 33_H 34_H
(T123)	(T124)

Batch write (bit units) (command: BW)

Writes bit devices in batch.

Message format

The following shows the message format of the request data and response data of the command.

■Request data

Command	Message wait	Head device	Number of device points	Write data for the number of device points

■Response data

There is no response data for this command.

Data specified by request data

■Command

BW command

B	W
42_H	57_H

■Message wait

Specify the delayed time of the response transmission. (☞ Page 771 Message wait)

■Head device

Specify the head device. (☞ Page 775 Device codes, device numbers)

■Number of device points

Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.

Specify the number of device points within the following range:

- 1 ≤ Number of device points ≤ 160
- Head device No. + Number of device points -1 ≤ Maximum device No.

■Write data for the number of device points

Store the data to be written in batch. (☞ Page 777 Read data, write data)

Communication example

Write data in bit units in batch under the following conditions.

- Message wait: 0ms
- Head device: M903
- Number of device points: 5 points

(Request data)

■When using BW

Message wait

Number of device points (Number of words)

Head device

Write data for the number of device points

B	W	0	M	0	9	0	3	0	5	0	1	1	0	1
42_H	57_H	30_H	$4D_H$	30_H	39_H	30_H	33_H	30_H	35_H	30_H	30_H	30_H	31_H	30_H
										(M903)	(M904)	(M905)	(M906)	(M907)

Batch write (word units) (command: WW, QW)

Write data to bit devices in 16-point units.

Write data to word devices in 1-point units.

Message format

The following shows the message format of the request data and response data of the command.

■Request data

Command	Message wait	Head device	Number of device points	Write data for the number of device points

■Response data

There is no response data for this command.

Data specified by request data

■Command

WW command	QW command
W W 57_H 57_H	Q W 51_H 57_H

■Message wait

Specify the delayed time of the response transmission. (☞ Page 771 Message wait)

■Head device

Specify the head device. (☞ Page 775 Device codes, device numbers)

■Number of device points

Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.

Specify the number of device points within the following range:

- For bit device: 1 ≤ Number of device points ≤ 10
- For bit device: Head device No. + Number of device points × 16 - 1 ≤ Maximum device No.
- For word device: 1 ≤ Number of device points ≤ 64
- For word device: Head device No. + Number of device points -1 ≤ Maximum device No.

When reading CN200 to CN255 (CN00200 to CN00255), since 1 device point is 2 words, the number of device points becomes 32 points.

■Write data for the number of device points

Store 4-digit data per one device point. (☞ Page 777 Read data, write data)

Point

When specifying bit devices, set the head device No. in multiples of 8.

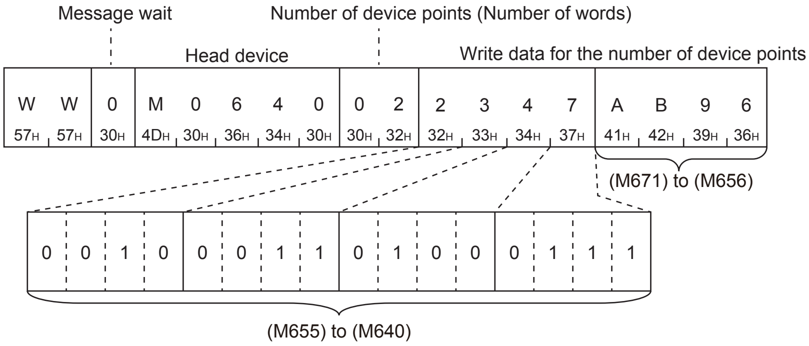
Communication example (Writing to word bit memory)

Write data to bit devices in 16-point units under the following conditions.

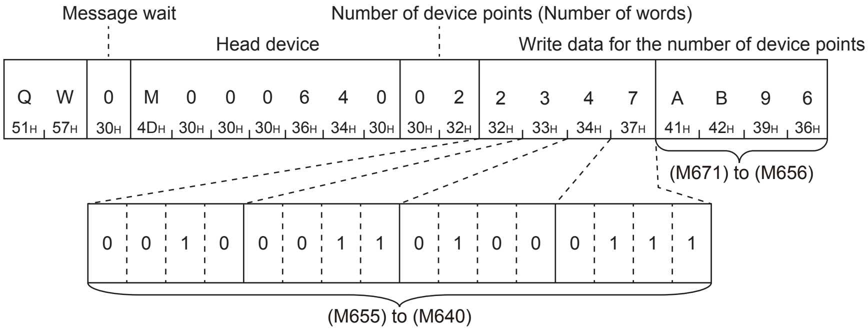
- Message wait: 0ms
- Head device: M640
- Number of device points: 32 points (2 words)

(Request data)

■When using WW



■When using QW



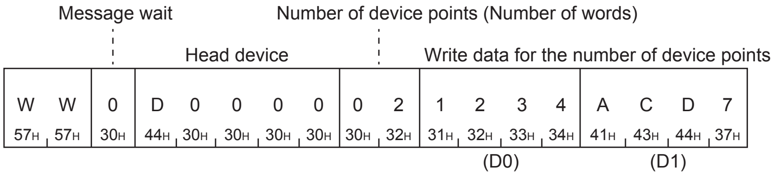
Communication example (Writing to word device memory)

Write data to word devices in 1-point units under the following conditions.

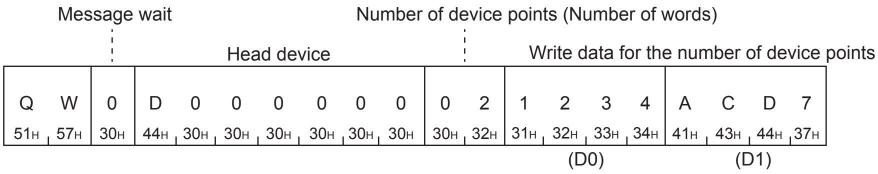
- Message wait: 0ms
- Head device: D0
- Number of device points: 2 points (2 words)

(Request data)

■When using WW



■When using QW



Test (random write) (bit units) (command: BT)

Set/reset devices and device numbers to bit devices by specifying them randomly in 1 point units.

Message format

The following shows the message format of the request data and response data of the command.

■Request data

Command	Message wait	Number of device points (n points)	Device (first point)	Set/reset (first point)	...	Device (nth point)	Set/reset (nth point)

■Response data

There is no response data for this command.

Data specified by request data

■Command

BT command

B	T
42H	54H

■Message wait

Specify the delayed time of the response transmission. (☞ Page 771 Message wait)

■Number of device points

Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.

Specify the number of device points within the following range:

- 1 ≤ Number of device points ≤ 20

■Device

Specify the device to test. (☞ Page 775 Device codes, device numbers)

■Set/Reset

- 0 (30H): Reset (OFF)
- 1 (31H): Set (ON)

Communication example

Perform the test in bit units under the following conditions.

- Message wait: 0ms
- Number of device points: 3 points
- Device: Turn ON M50, turn OFF S100, and turn ON Y1

(Request data)

■When using BT

	Message wait	Number of device points (Number of words)	Device	Set/ reset	Device	Set/ reset	Device	Set/ reset
B T	0	0 3	M 0 0 5 0	1	S 0 1 0 0	0	Y 0 0 0 1	1
42H 54H	30H	30H 33H	4DH 30H 30H 35H 30H	31H	53H 30H 31H 30H 30H	31H	59H 30H 30H 30H 31H	31H

Test (random write) (word units) (command: WT, QT)

Set/reset devices and device numbers to bit devices by specifying them randomly in 16 point units.

Write devices and device numbers to word devices by specifying them randomly in 1 point units.

A mixture of word devices and bit devices (16 bit units) can be specified. However, LC0 to LC55 (CN200 to CN255, and CN00200 to CN00255) cannot be used for the word device.

Message format

The following shows the message format of the request data and response data of the command.

■Request data

Command	Message wait	Number of device points (n points)	Device (first point)	Write data (first point)	...	Device (nth point)	Write data (nth point)

■Response data

There is no response data for this command.

Data specified by request data

■Command

WT command	QT command
W T 57_H 54_H	Q T 51_H 54_H

■Message wait

Specify the delayed time of the response transmission. (☞ Page 771 Message wait)

■Number of device points

Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.

Specify the number of device points within the following range:

- 1 ≤ Number of device points ≤ 10 (for bit device: 10 (16 points are designated as 1))

■Device

Specify the device to test. (☞ Page 775 Device codes, device numbers)

■Write data

Store 4-digit data per one device point. (☞ Page 777 Read data, write data)

Point

When specifying bit devices, set the head device No. in multiples of 8.

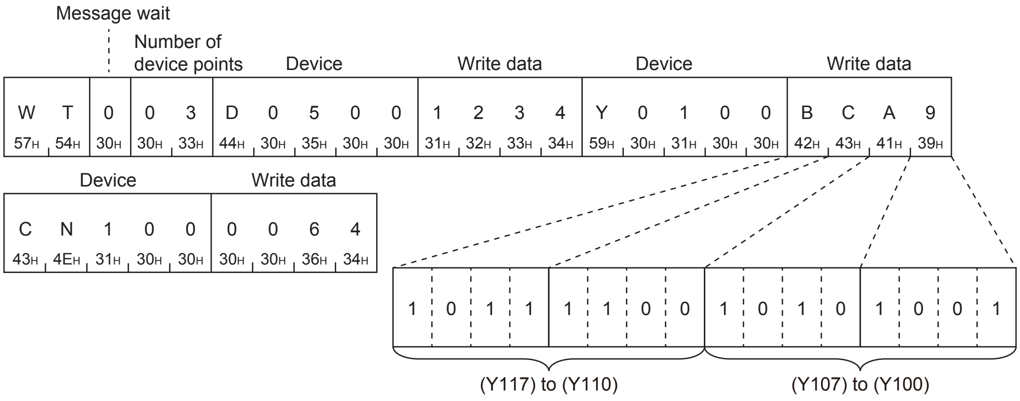
Communication example

Write data with mixture specification of word devices and bit devices (16-point unit) under the following conditions.

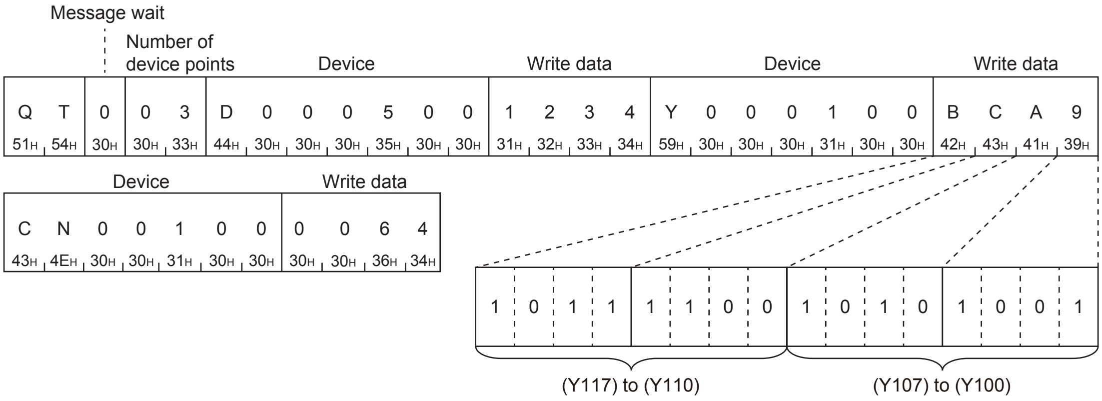
- Message wait: 0ms
- Number of device points: 3 points (3 words)
- Device: Set 1234H to D500, BCA9H to 'Y100 to Y117', and 64H to current values of C100

(Request data)

■When using WT



■When using QT



44.5 Remote Operation

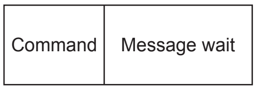
Remote RUN, remote STOP (command: RR, RS)

Perform remote RUN (RR) and remote STOP (RS) of the FX5 PLC from the target device.

Message format

The following shows the message format of the request data and response data of the command.

■Request data



■Response data

There is no response data for RR and RS command.

Data specified by request data

■Command

RR command	RS command
	

■Message wait

Specify the delayed time of the response transmission. (☞ Page 771 Message wait)

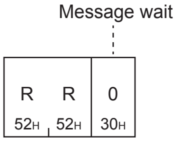
Communication example

Perform remote RUN or remote STOP of the PLC under the following conditions.

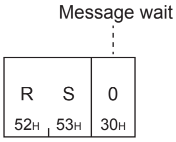
- Message wait: 0ms

(Request data)

■When using RR



■When using RS



Read CPU model name (command: PC)

Read the model name of the PLC which links to the target device.

Message format

The following shows the message format of the request data and response data of the command.

■Request data

P 50_H	C 43_H	Message wait

■Response data

Model name

The read contents can be interpreted as the model names of PLC according to the following.

PLC model (CPU module)	Model code (hexadecimal)	PLC model (CPU module)	Model code (hexadecimal)
FX1S	F2H	A2ACPU-S1	93H
FX0N	8EH	A2CCPU	9AH
FX2(FX), FX2C	8DH	A2USCPU	82H
FX1N, FX1NC	9EH	A2CPU-S1, A2USCPU-S1	83H
FX2N, FX2NC	9DH	A3CPU, A3NCPU	A3H
FX3S	F5H	A3ACPU	94H
FX3G, FX3GC	F4H	A3HCPU, A3MCPU	A4H
FX3U, FX3UC	F3H	A3UCPU	84H
FX5S, FX5UJ, FX5U, FX5UC	F3H	A4UCPU	85H
A0J2HCPU	98H	A52GCPU	9AH
A1CPU, A1NCPU	A1H	A73CPU	A3H
A1SCPU, A1SJCPU	98H	A7LMS-F	A3H
A2CPU(-S1), A2NCPU(-S1), A2SCPU	A2H	AJ72P25/R25	ABH
A2ACPU	92H	AJ72LP25/BR15	8BH

Data specified by request data

■Command

PC command

P 50_H	C 43_H

■Message wait

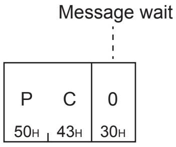
Specify the delayed time of the response transmission. (☞ Page 771 Message wait)

Communication example

Read the model name of the PLC under the following conditions.

- Message wait: 0ms

(Request data)



Model code setting

The model code read by the PC command can be changed freely using a special relay or register.

■Corresponding devices

The following devices are used for setting the model code.

Device number	Description	Default
SM8125	ON: The lower 8 bits of SD8126 are used as the model name. OFF: F3H (fixed value) is used as the model name.	OFF
SD8126	Store the model code to be changed to the lower 8 bits.	0

■Setting method

The procedure for changing the model code is shown below.

1. Set the model code in the lower 8 bits of SD8126.
2. Turn on SM8125.

The lower 8 bits of SD8126 are set in the response data.

44.6 Global Function

Global signal ON/OFF (command: GW)

Turn on and off the special relay of all stations or the specified station of the PLC at the multi drop link from the target device.

Message format

The following shows the message format of the request data and response data of the command.

■Request data

G 47H	W 57H	Message wait	Factor number

■Response data

There is no response data for this command.

Data specified by request data

■Command

GW command

G 47H	W 57H

■Message wait

Specify the delayed time of the response transmission. (☞ Page 771 Message wait)

■Factor number

For the special relay corresponding to the following CH, in the case of 1 (31H), turn on the special relay, and in the case of 0 (30H), turn off the special relay.

CH	Special relay corresponding to CH
1	SM8680
2	SM8690
3	SM8700
4	SM8710

Communication example

Turn on the special relay of the target station under the following conditions.

- Message wait: 0ms

(Request data)

Message wait

Factor number

G 47H	W 57H	0 30H	1 31H

Point

When specifying all stations, specify FFH to the station number.

If the value other than FFH is specified, the special relay of the specified station number is turned on and off.

44.7 Loopback Test

A loopback test checks whether the communication function between an external device and FX5 CPU module operates normally.

Loopback test (Command: TT)

Return the characters received from an external device to the external device unchanged.

Message format

The following shows the message format of the request data and response data of the command.

■Request data

T T 54_H 54_H	Message wait	Character length	Loopback data

■Response data

Character length	Loopback data

Data specified by request data

■Command

TT command

T T 54_H 54_H

■Message wait

Specify the delayed time of the response transmission. (☞ Page 771 Message wait)

■Character length

Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.

Specify the character length within the following range:

- 1 ≤ Character length ≤ 254

■Loopback data

Store the loopback data for character length.

Data stored by response data

■Character length

The same data as request data is stored.

■Loopback data

The same data as request data is stored.

Point

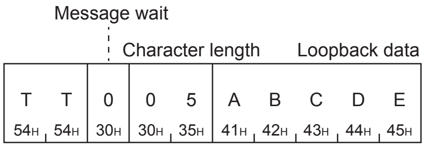
Specify 'FF' for PC No.

Communication example

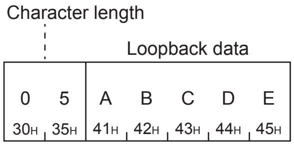
Return 5-digit data received from an external device to the external device unchanged under the following conditions.

- Message wait: 0ms
- Character length: 5
- Loopback data: 'ABCDE'

(Request data)



(Response data)



PART 7 TROUBLESHOOTING

This part consists of the following chapters.

45 TROUBLESHOOTING PROCEDURE

This section contains an explanation of errors that may occur during communication between the FX5 CPU module or Ethernet-equipped module and other devices, and troubleshooting for such errors.
For module-specific troubleshooting, refer to the manual of each module.

Point

When an error occurs, saving the program or device status will be useful for clarifying the cause of the error.
For details on reading from a programmable controller, refer to the following manual.
📖 GX Works3 Operating Manual

When an error occurs, perform troubleshooting in the following order.

1. Check the LEDs on the CPU module. (☞ Page 794 Checking with LEDs)

2. Check the LEDs on each intelligent function module. (📖 User's manual of each module)

3. Check that each module is correctly mounted and wired.

4. Connect the engineering tool and start the system monitor. The module that generated the error can be checked. (☞ Page 796 Checking with Engineering Tool)

5. Select the module that generated the error and start the module diagnostics. The cause of the error and the action to eliminate it can be checked. (☞ Page 797 Module diagnostics)

6. If the cause cannot be determined by module diagnostics, check the history of operations and errors in the event history of the engineering tool to determine the cause. (☞ Page 798 Event history)

7. Check that the communication settings are correct.

8. If the cause cannot be identified through steps 1 to 6, perform troubleshooting by symptom. (☞ Page 805 TROUBLESHOOTING BY SYMPTOM)

45.1 Checking with LEDs

This section contains an explanation of errors that can be checked by LEDs on the front of the module.
Check the status of the LED indications of [RD] and [SD] on the module or the communication board/communication adapter.

LED status		Operation status
RD	SD	
On	On	Data is being transmitted and received.
On	Off	Data is being received, but not being transmitted.
Off	On	Data is being transmitted but not being received.
Off	Off	Data is neither being transmitted nor being received.

When communication is being performed normally, the LEDs both flash brightly.
If the LEDs are not flashing, check the wiring and the communication settings or the master station or other local station setting.

<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 the error codes, refer to ☞ Page 820 Error codes of the Ethernet communication.
SD/RD	Does not turn on when data is sent/received	If [ERR] is lit: • Eliminate the cause of [ERR] being lit. If cable connection is faulty: • Check cable connection. • Perform initial processing completion check and check if there is anything wrong with the Ethernet line. In case of own station IP address setting error: • 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. If there is something wrong with transmission program of other device: • 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	The power cable or extension cable may not be correctly connected to the CPU module or Ethernet 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.
ERROR	Turns on when power of CPU module is turned on	In case of Ethernet parameter setting error: • Check/correct Ethernet parameter setting values using GX Works3. In case of Ethernet 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 • Other processing (processing wherein error code is stored) For the error codes, refer to ☞ Page 816 ERROR CODES.
SD/RD	Does not turn on when data is sent/received	If [ERROR] LED is lit: • Eliminate the cause of [ERROR] being lit. If cable connection is faulty: • Check cable connection. • Perform initial processing completion check and check if there is anything wrong with the Ethernet line. In case of own station IP address setting error: • 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. If there is something wrong with transmission program of other device: • 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 796 Checking with Engineering Tool.

45.2 Checking with Engineering Tool

Use the engineering tool to check the errors that have occurred and their history, and identify the cause(s) of the errors. This method will yield more detailed information, the cause(s) of the errors, and the actions to eliminate them, compared with checking with LEDs.

The engineering tool has the following functions to support troubleshooting.

Function	Action
System monitor	Function that displays the module configuration, detailed information of each module, and error status
Module diagnostics	Function for diagnosing the target module (checking currently occurring errors and their detailed information)
Event history	Function that displays event information such as errors that occurred in the CPU module, expansion board, expansion adapter, intelligent function modules, and errors on the network

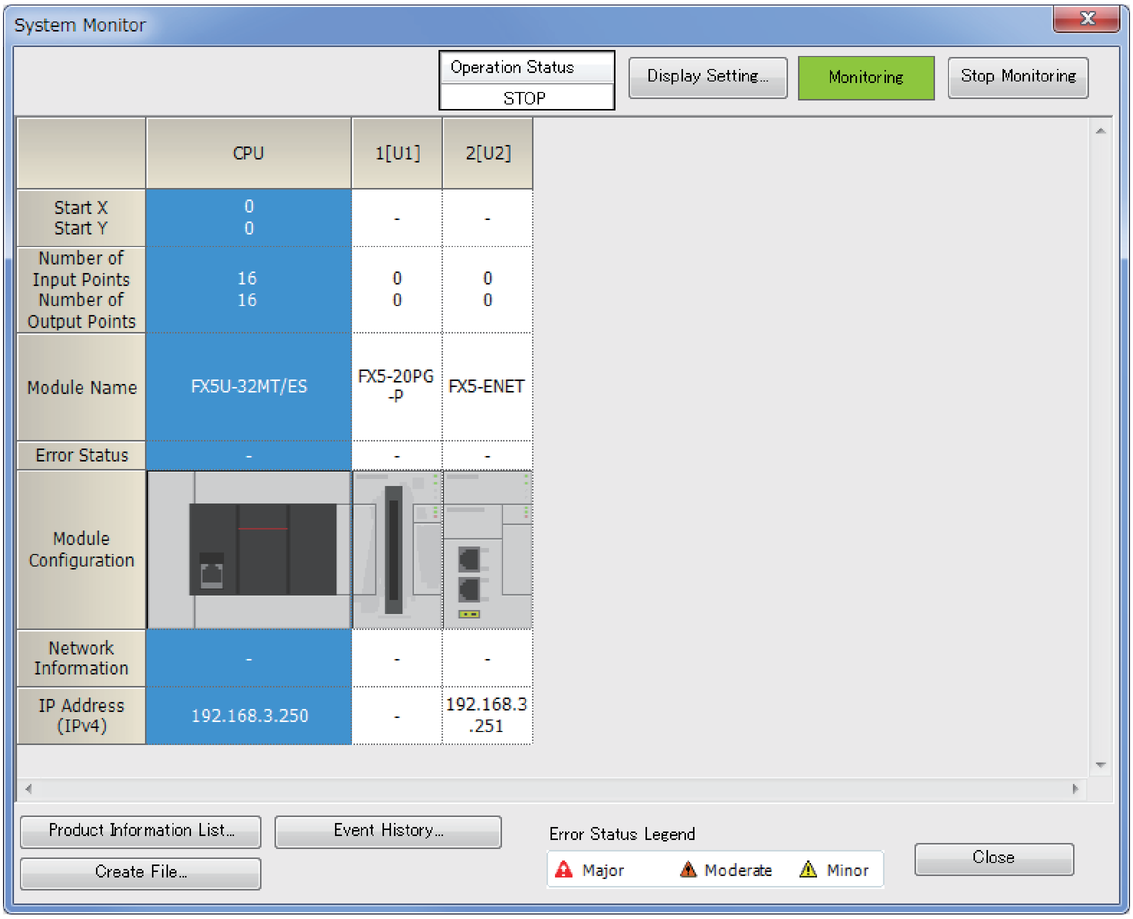
For details on the functions of each engineering tool, refer to the following manual.

📖 GX Works3 Operating Manual

System monitor

System monitor is a function that displays the module configuration, detailed information of each module, and error status. Module diagnostics can be started for modules where an error is occurring.

🖱 [Diagnostics] ⇨ [System Monitor]



Operation Status
STOP

	CPU	1[U1]	2[U2]
Start X Start Y	0 0	-	-
Number of Input Points Number of Output Points	16 16	0 0	0 0
Module Name	FX5U-32MT/ES	FX5-20PG-P	FX5-ENET
Error Status	-	-	-
Module Configuration			
Network Information	-	-	-
IP Address (IPv4)	192.168.3.250	-	192.168.3.251

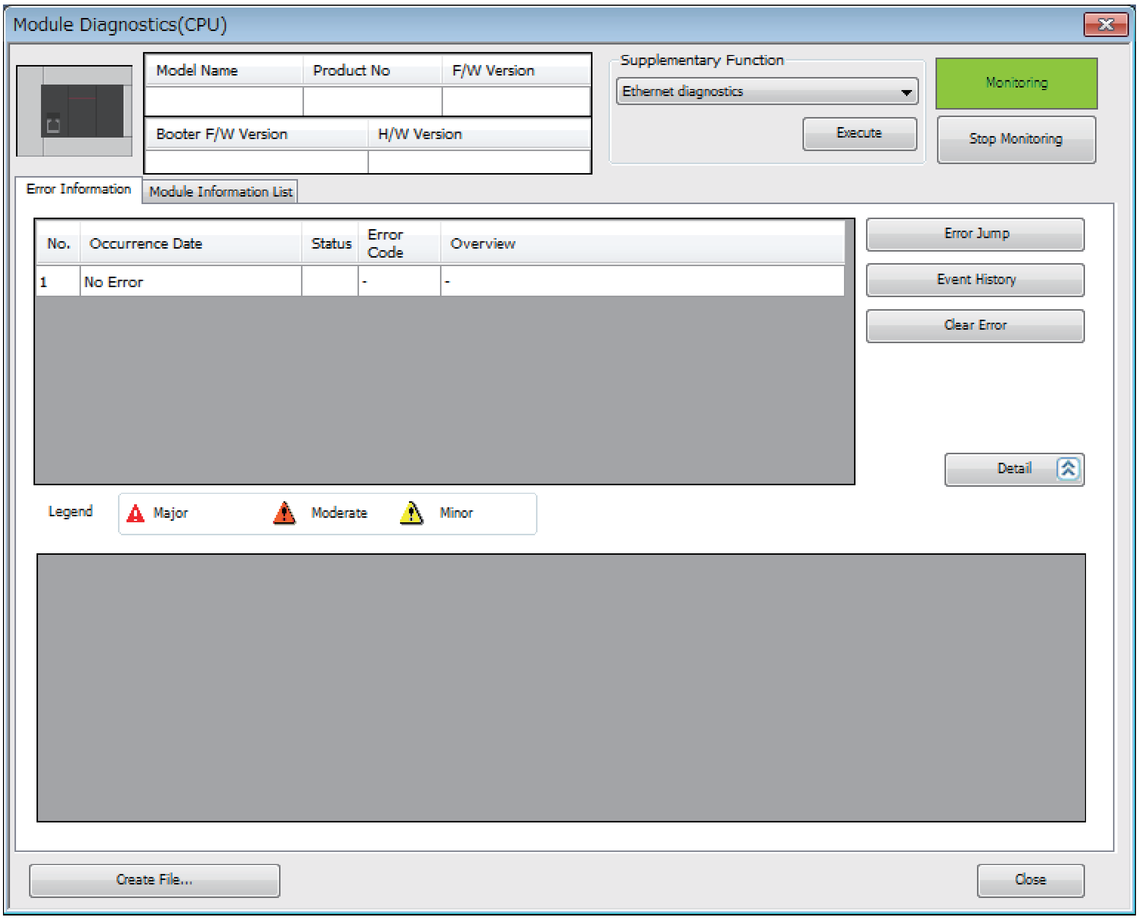
Module diagnostics

Module diagnostics is a function for diagnosing the target module (checking currently occurring errors and their detailed information).

It displays the error that occurred, its detailed information, the cause, and the action, so the information necessary for troubleshooting can be checked. In addition, the location of the parameter or program error can be identified by selecting the error and clicking the [Error Jump] button.

In the [Module Information List] tab, the current LED information or switch information of the corresponding module can be checked.

[Diagnostics] ⇨ [System Monitor] ⇨ Double-click the corresponding module



No.	Occurrence Date	Status	Error Code	Overview
1	No Error		-	-

Restriction

In a multiple CPU system, if the I/O assignment setting of the system parameters is different from the actual configuration of the installed module, the correct information may not be displayed in the module information list.

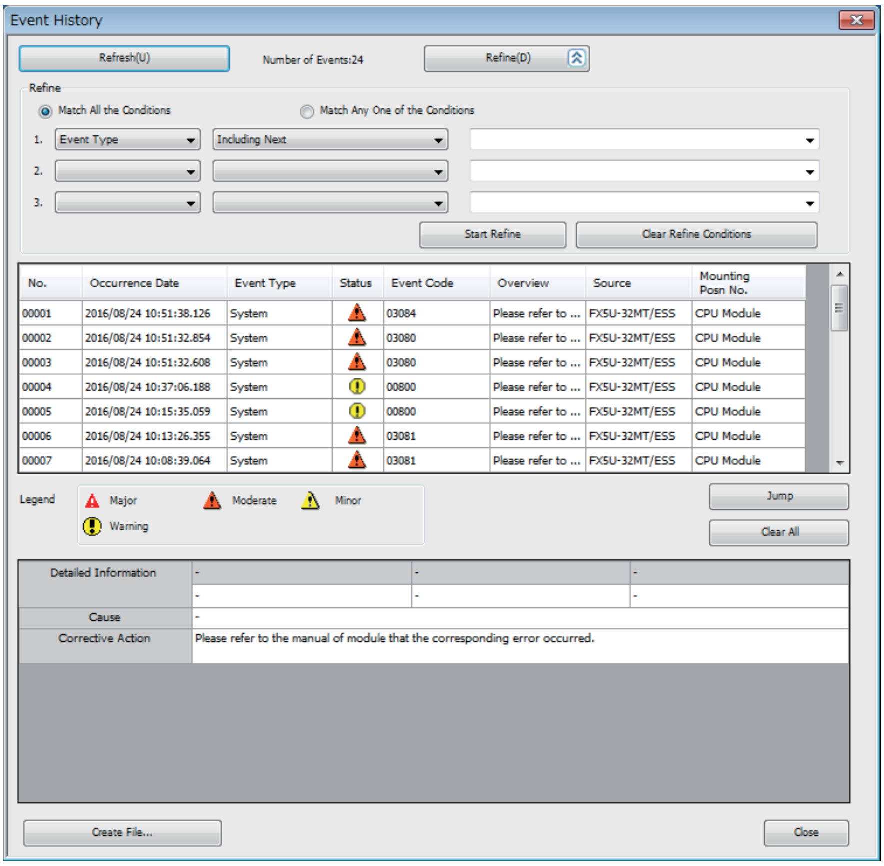
Event history

Event history is a function that displays event information such as errors that occurred in each module, operations that were executed, and errors on the network.

Information collected before the power was turned off or reset can also be displayed. This allows the user to identify the cause of an error based on past operations and error occurrence trends.

The displayed information can also be saved in CSV file format.

[Diagnostics] ⇨ [System Monitor] ⇨ [Event History] button



Number of Events:24

● Match All the Conditions ○ Match Any One of the Conditions

1. Event Type / Including Next

2.

3.

No.	Occurrence Date	Event Type	Status	Event Code	Overview	Source	Mounting Posn No.
00001	2016/08/24 10:51:38.126	System	⚠	03084	Please refer to ...	FX5U-32MT/ESS	CPU Module
00002	2016/08/24 10:51:32.854	System	⚠	03080	Please refer to ...	FX5U-32MT/ESS	CPU Module
00003	2016/08/24 10:51:32.608	System	⚠	03080	Please refer to ...	FX5U-32MT/ESS	CPU Module
00004	2016/08/24 10:37:06.188	System	❗	00800	Please refer to ...	FX5U-32MT/ESS	CPU Module
00005	2016/08/24 10:15:35.059	System	❗	00800	Please refer to ...	FX5U-32MT/ESS	CPU Module
00006	2016/08/24 10:13:26.355	System	⚠	03081	Please refer to ...	FX5U-32MT/ESS	CPU Module
00007	2016/08/24 10:08:39.064	System	⚠	03081	Please refer to ...	FX5U-32MT/ESS	CPU Module

Detailed Information	-	-	-
	-	-	-
Cause	-		
Corrective Action	Please refer to the manual of module that the corresponding error occurred.		

Point

The event history function is used for purposes such as the following.

- To check the error occurrence status and investigate the cause of faults that occurred in the system or device
- To check when and where the program or parameters of the programmable controller were changed
- To check whether there is unauthorized access by a third party

Ethernet diagnostics

The communication status between an Ethernet-equipped module and external device can be checked by using Ethernet diagnostics.

Select the Ethernet-equipped module to be diagnosed in "Target Module Specification".

- Ethernet diagnostics item

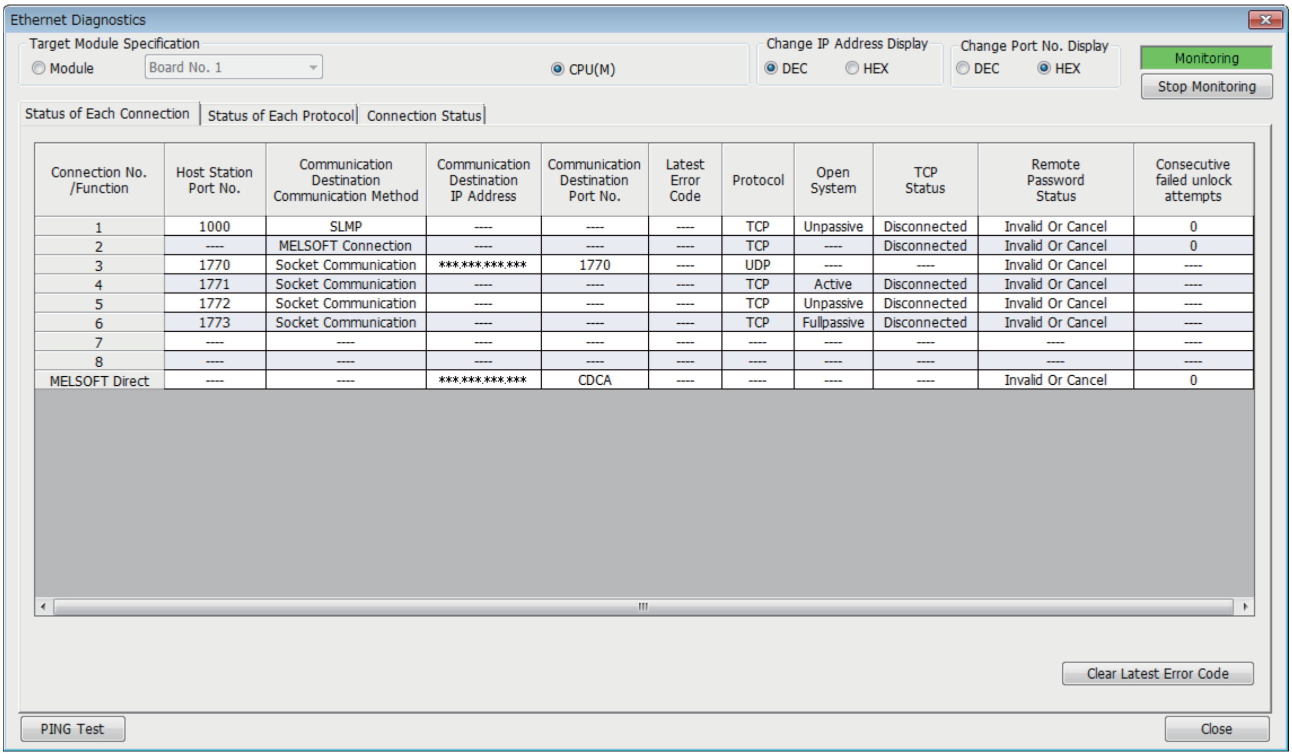
Item	Description
Status of Each Connection	Displays information concerning status of each connection.
Status of Each Protocol	The total of the send/receive of the packet etc. for each protocol is displayed.
Connection Status	Monitors connection status.

[Diagnostics] ⇨ [Ethernet Diagnostics]

Status of Each Connection

The status of each connection of the Ethernet-equipped module selected.

The following table lists the displayed items in "Status of Each Connection" tab.



Target Module Specification: ○ Module Board No. 1 ● CPU(M)

Change IP Address Display: ● DEC ○ HEX

Change Port No. Display: ○ DEC ● HEX

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

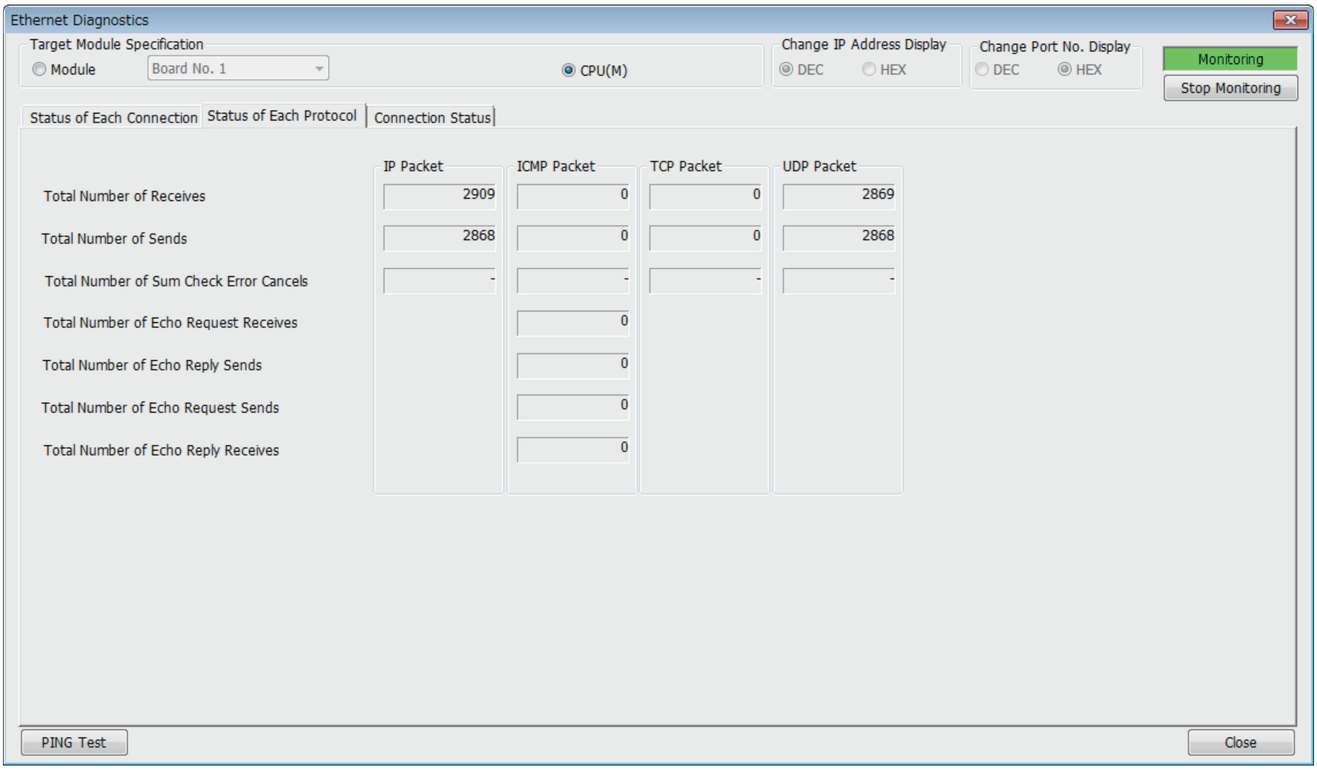
Item	Description
Connection No./Function	Displays the connection number and functions (MELSOFT direct connection).
Host Station Port No.	Displays the own station port number used.
Communication Destination Communication Method	Displays the communication method.
Communication Destination IP Address	Displays 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]
Consecutive failed unlock attempts	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.



Target Module Specification: ○ Module (Board No. 1) ● CPU(M)

Change IP Address Display: ● DEC ○ HEX

Change Port No. Display: ○ DEC ● HEX

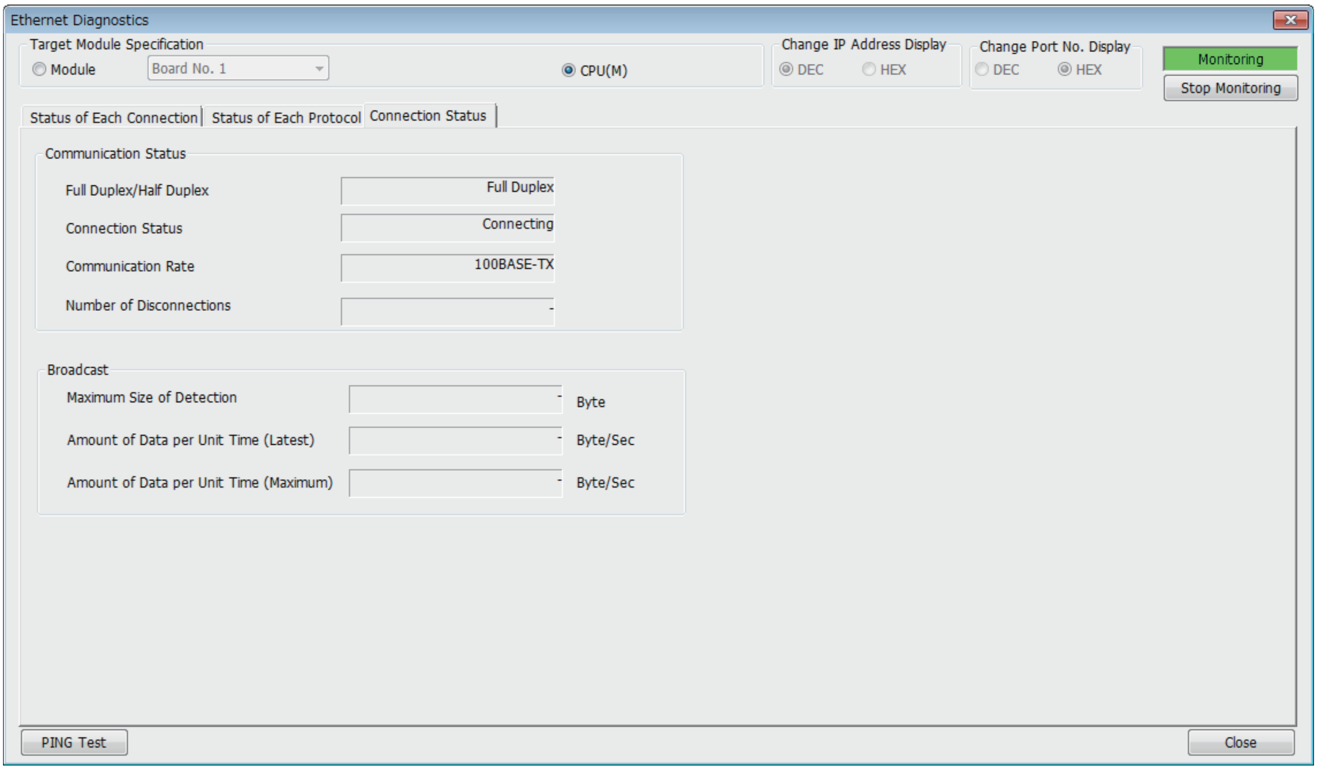
	IP Packet	ICMP Packet	TCP Packet	UDP Packet
Total Number of Receives	2909	0	0	2869
Total Number of Sends	2868	0	0	2868
Total Number of Sum Check Error Cancels	-	-	-	-
Total Number of Echo Request Receives		0		
Total Number of Echo Reply Sends		0		
Total Number of Echo Request Sends		0		
Total Number of Echo Reply Receives		0		

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.



Ethernet Diagnostics

Target Module Specification: ○ Module Board No. 1 ● CPU(M)

Change IP Address Display: ● DEC ○ HEX

Change Port No. Display: ○ DEC ● HEX

Communication Status

Full Duplex/Half Duplex: Full Duplex

Connection Status: Connecting

Communication Rate: 100BASE-TX

Number of Disconnections: -

Broadcast

Maximum Size of Detection: - Byte

Amount of Data per Unit Time (Latest): - Byte/Sec

Amount of Data per Unit Time (Maximum): - Byte/Sec

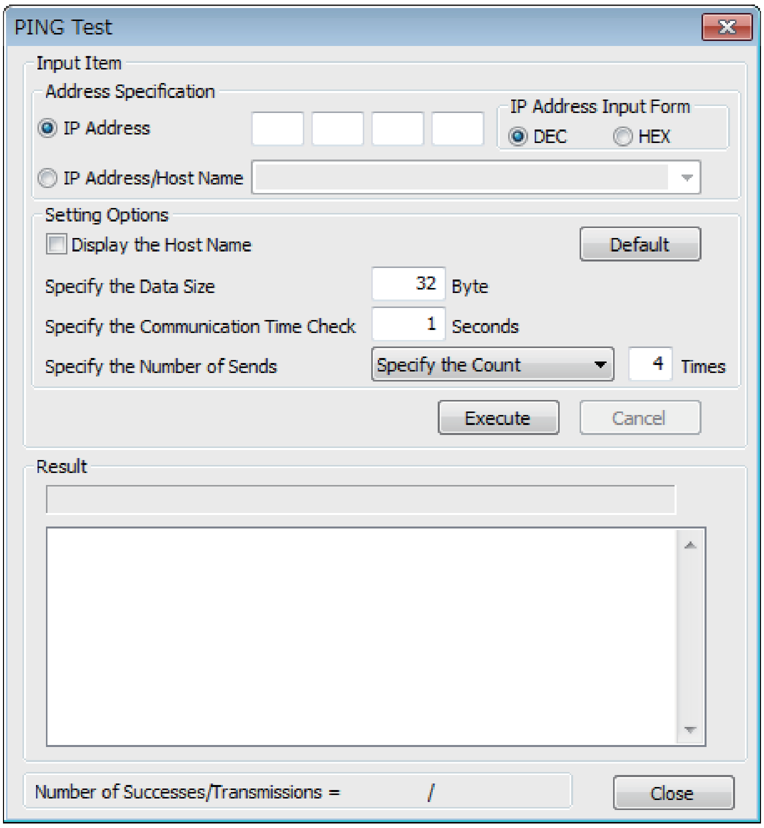
The following table lists the displayed items in "Connection Status" tab.

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" screen ⇨ [PING Test] 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

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.

CC-Link IE Field Network Basic diagnostics

Perform troubleshooting by checking the network status and error details with GX Works3.

For details, refer to the following manual.

📖 CC-Link IE Field Network Basic Reference Manual

Simple CPU communication diagnostics

The communication status of the simple CPU communication function can be checked with GX Works3.

For details, refer to the following.

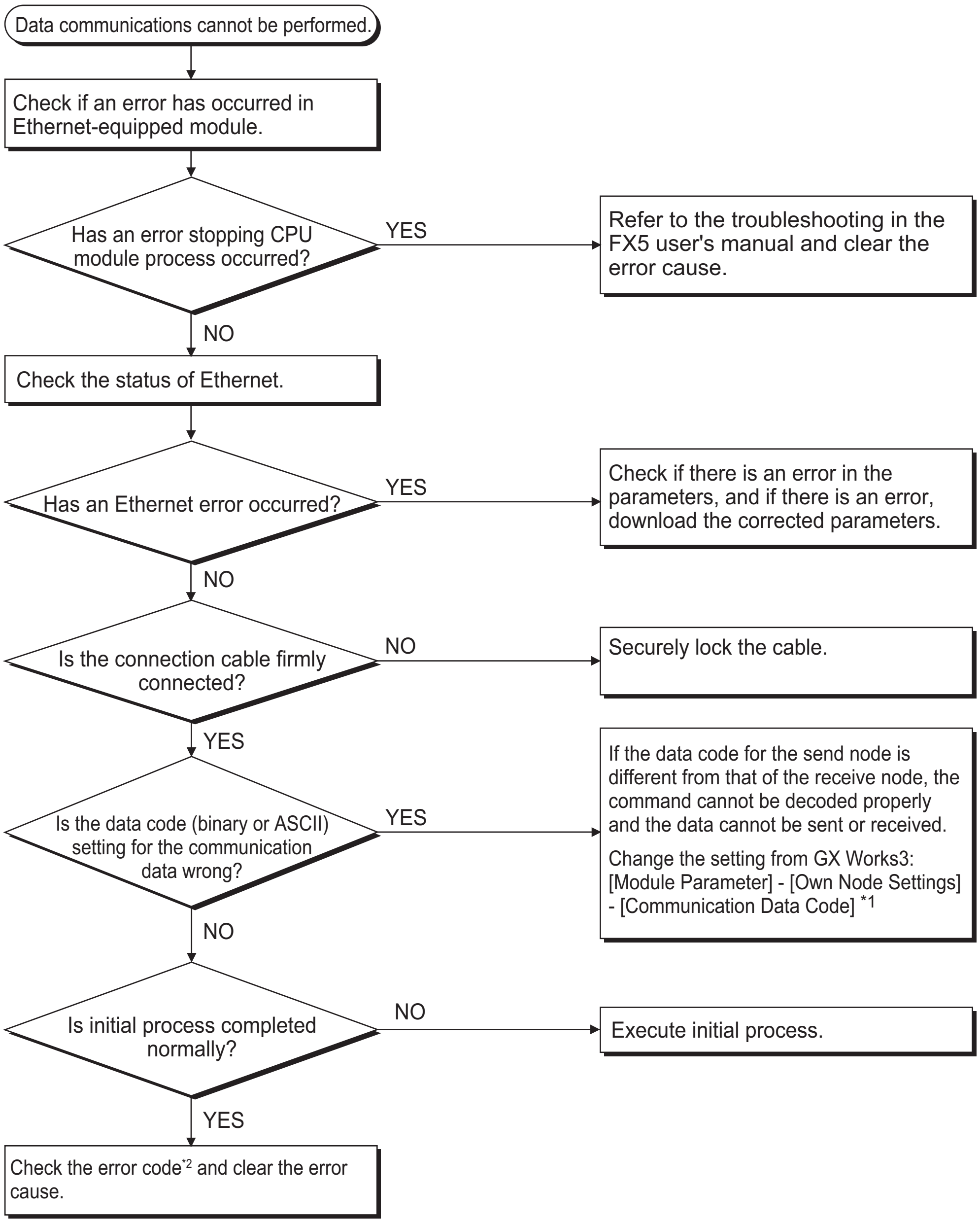
☞ Page 62 Checking the Simple CPU Communication Status

46 TROUBLESHOOTING BY SYMPTOM

The following shows troubleshooting by symptom in flowchart format. For details on corrective action, refer to ☞ Page 811 IP Address Duplication Detection onward.

46.1 Troubleshooting Flowchart

Error during communication between Ethernet-equipped module and other device



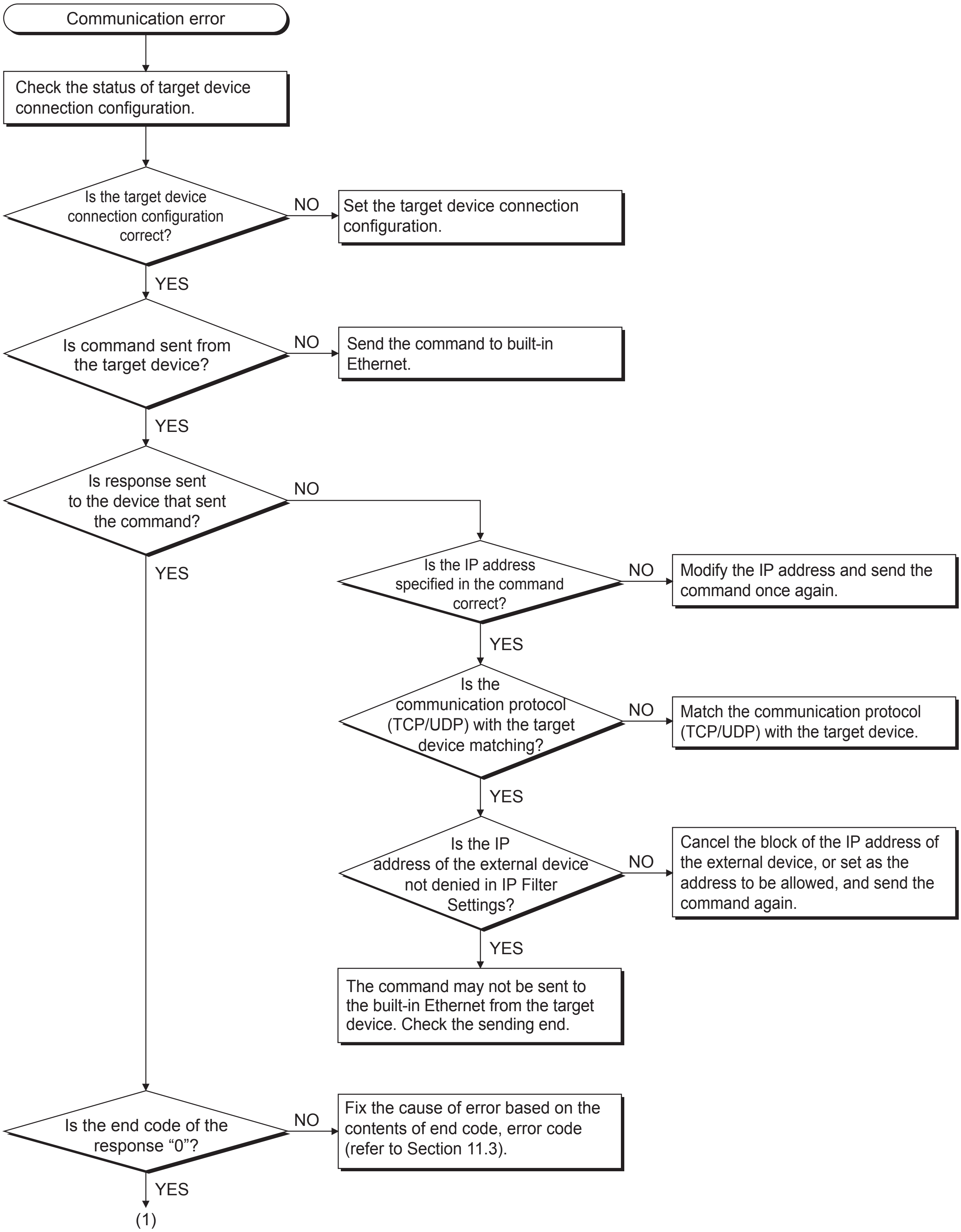
[1] When the communication data code setting for the Ethernet-equipped module is different from the data code setting for the external device, an error code not listed in the error code list may be returned to the external device. In addition, if data with a different data code is received, the command cannot be decoded properly. The Ethernet-equipped module will return an error response according to the communication data code setting.

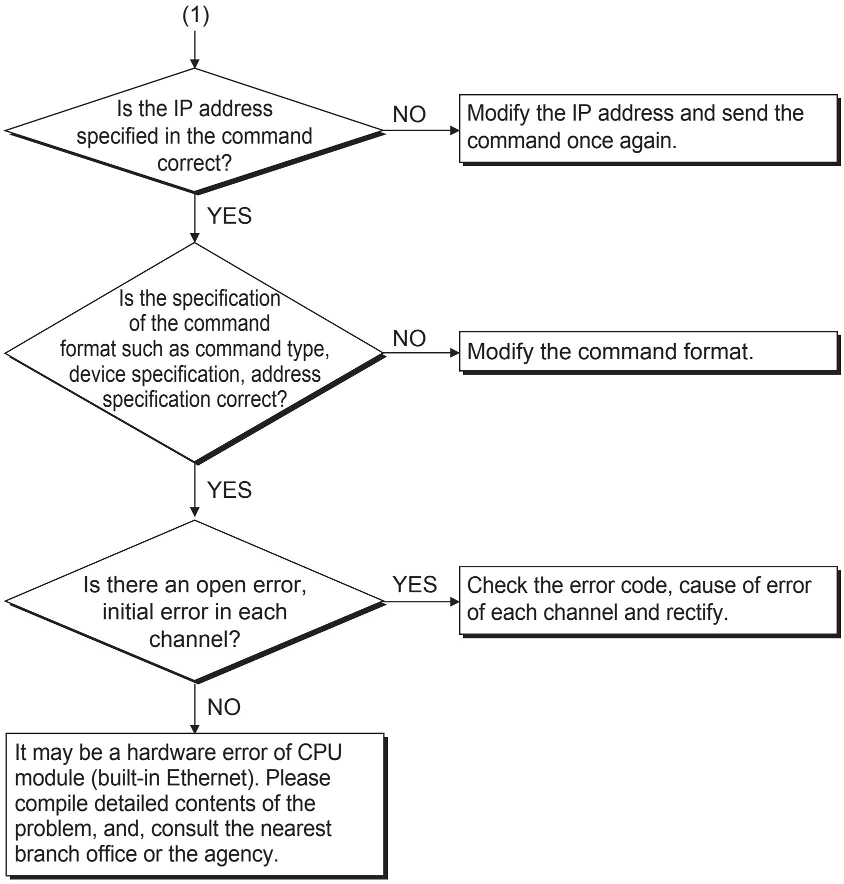
[2] Refer to the following.
☞ Page 816 ERROR CODES

Point

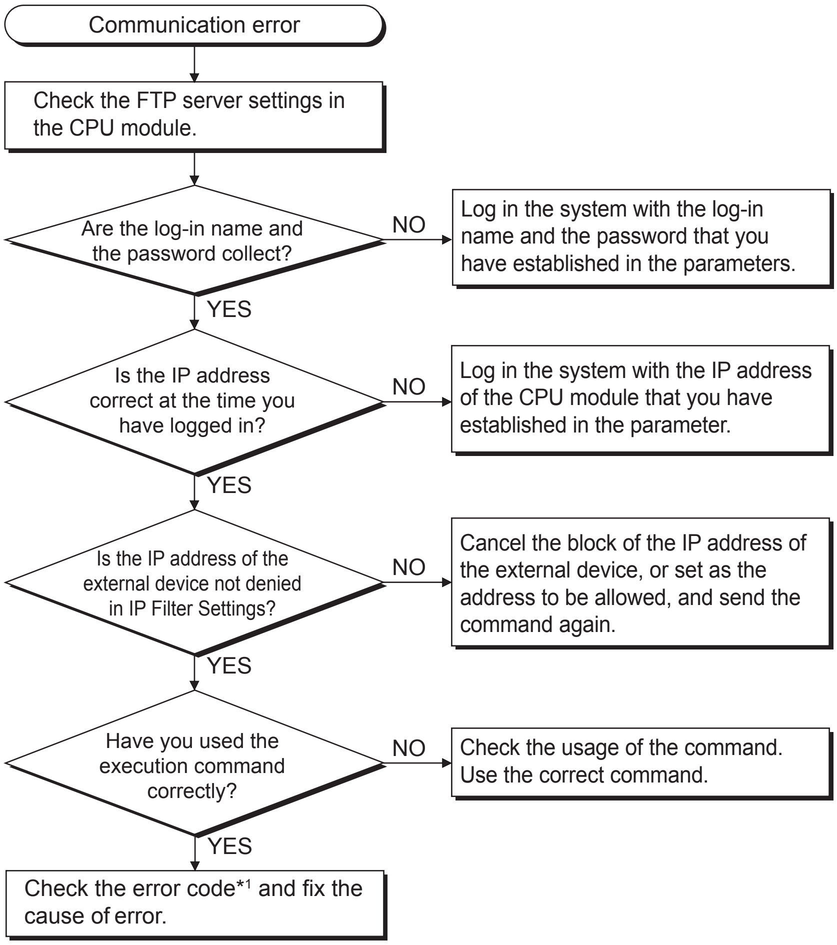
- If trouble occurs while using Ethernet, check the error status with the Ethernet diagnostics function of GX Works3. For error contents, refer to ☞ Page 820 Error codes of the Ethernet communication.
- For information concerning when the [ERR] LED or [ERROR] LED is lit, refer to ☞ Page 794 Checking with 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




(1)
Is the IP address specified in the command correct?
NO
Modify the IP address and send the command once again.
YES
Is the specification of the command format such as command type, device specification, address specification correct?
NO
Modify the command format.
YES
Is there an open error, initial error in each channel?
YES
Check the error code, cause of error of each channel and rectify.
NO
It may be a hardware error of CPU module (built-in Ethernet). Please compile detailed contents of the problem, and, consult the nearest branch office or the agency.

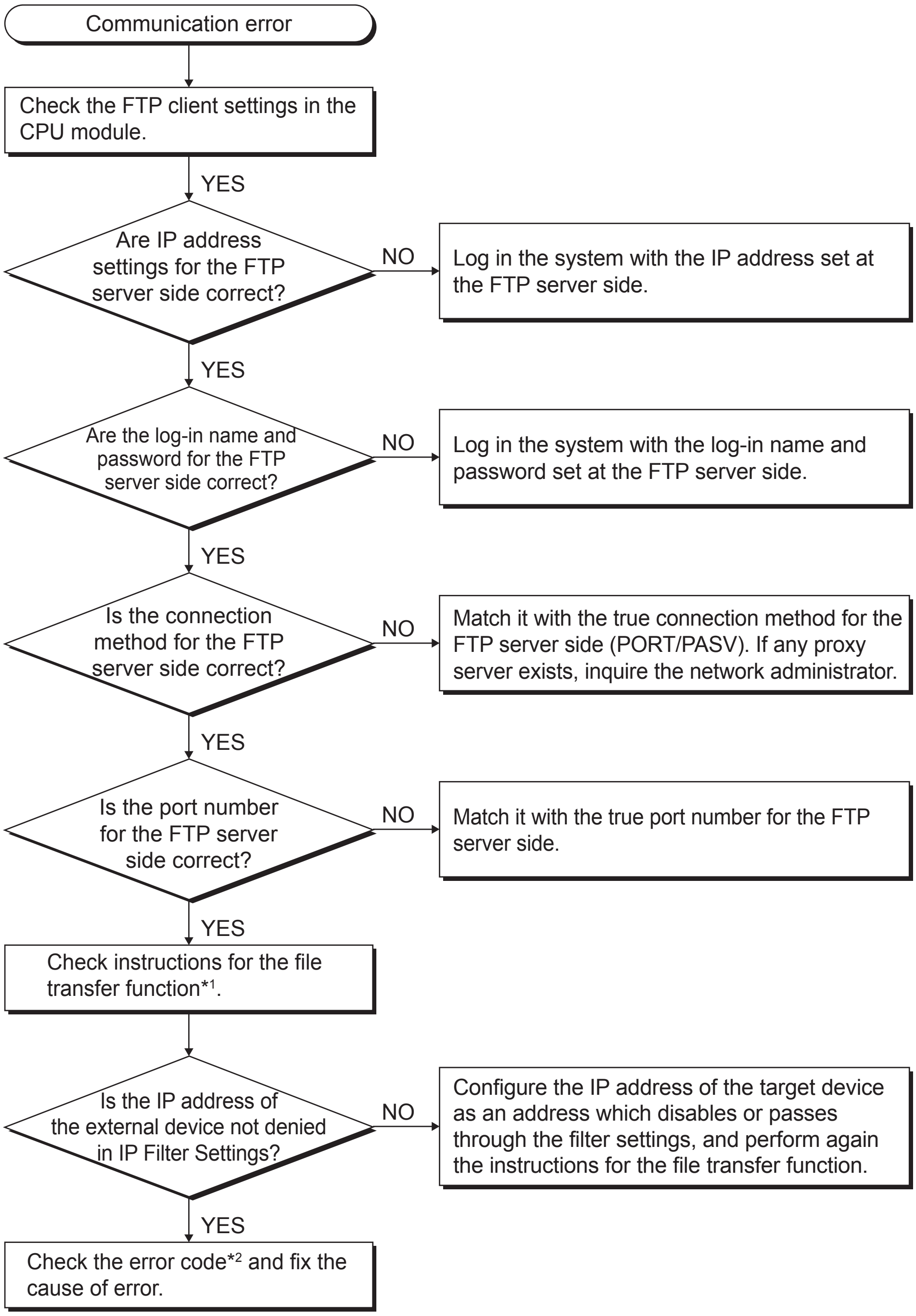
Errors during file transfer function (FTP server)



*1 Refer to the following.
☞ Page 816 ERROR CODES

Errors during file transfer function (FTP client)

For check items other than those shown below, refer to ☞ Page 202 Precautions.



[1] Refer to the following.
📖 MELSEC iQ-F FX5 Programming Manual (Instructions, Standard Functions/Function Blocks)

[2] Refer to the following.
☞ Page 816 ERROR CODES

46.2 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.

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.

46.3 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.
Is the communication status of the simple CPU communication diagnostics or the simple CPU communication status of the special device "Communications impossible"?	Check the own station port number of the simple CPU communication setting with "Communications impossible" to check if the same port number is already used for other functions.
Have the parameters been written?	• 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?	• 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?	• 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)	• 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?	• 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
• 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.

When MODBUS/TCP-compatible devices are connected, an error code is not cleared.

If the error code (CFB0H) is not cleared, check the following item and take action.

Check item	Action
Is 00H (Broadcast) specified for communications between an Ethernet module and MODBUS RTU/ASCII-compatible devices via a gateway device?	• If 00H (Broadcast) is specified, the error code (CFB0H) is detected at each communication. Specify the station No. of MODBUS RTU/ASCII-compatible devices in Option (hexadecimal). (☞ Page 70 Option (Hexadecimal)) • Check the conditions of the cables, hubs and routers on the lines to the external devices. • Check and correct the IP address and station No. of the external devices.

46.4 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 178)
Are the login name and password correct?	Check the login name and password set in the parameters, and log in again. (☞ Page 178)
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 39 Setting module parameters)
Is executed command used properly?	Check the instructions of the command and use the command properly. (☞ Page 178)

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 819 Error codes of completion status)

46.5 When Using the Inverter Communication Function

1. When the operation mode of an inverter is not changed to computer link operation mode

- Verify that the inverter is set to the external operation mode.
- Verify that no signal is being input to the external terminals STF and STR.
- Verify that the correct change operation mode program is being executed.

2. When an inverter cannot be started even in computer link operation mode

- Verify that the program for starting the inverter is executed correctly.
- Verify that the operation command rights and speed command rights are set correctly.
- Verify that the allowable communication time interval is set correctly.

3. When an inverter is stopped by an alarm during operation due to defective communication

- Verify that a communication cable is connected correctly between the PLC and the inverter. (Check for poor contact or wire breakage.)
- Verify that a sequence program is created so that communication is executed with each inverter within a constant cycle. Set the communication check time interval to a large value, and check the communication status.
- Verify that the allowable communication time interval is set correctly.
- Verify that termination resistors are set correctly.

46.6 MODBUS/TCP Communication

When an error occurs in MODBUS/TCP communication, check the error details with Ethernet diagnostics. (☞ Page 799 Ethernet diagnostics)

47 ERROR CODES

47.1 Ethernet Communication

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	Values such as the IP address setting (SD8492 to SD8497) are outside the set range.	Recheck the values such as the IP address setting (SD8492 to SD8497).
1921H	Write request and clear request (SM8492 and SM8495) turned from OFF to ON simultaneously.	Verify that write request and clear request (SM8492 and SM8495) do not turn 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 797 Module diagnostics)

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.	• 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 detected in the own station are stored in Simple PLC Communication error code (FX5S/FX5UJ CPU module: SD10412 to SD10427, FX5U/FX5UC CPU module: SD10412 to SD10443).

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)	• 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 235 Remote Password)
CFB0H	Sending data failed due to a resend timeout during the simple CPU communication.	• 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.
CFB1H	The external device cannot be connected or has been disconnected.	• Check whether the connection cable is not disconnected. • Check the conditions of the cables, hubs and routers on the lines to the external devices. • 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.
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.
CFB5H	The frame received from the external device is abnormal.	• Check the operation of the external device. • Check the conditions of the cables, hubs and routers on the lines to the external devices.

Point

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.	• 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.	• Check whether the connection cable is not disconnected. • Check the conditions of the cables, hubs and routers on the lines to the external devices. • 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.
CFB2H	The same specified own station port number is already used.	Correct the specified port number so that it is not duplicate.
CFB3H	A request to the CPU module has failed.	• 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.	• 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.	• 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 a MODBUS/TCP connection device, CFB4H will be stored.

Error codes for those detected by the communication target

Error codes detected in the communication target are stored in Simple CPU communication error response code (FX5S/FX5UJ CPU module: SD10476 to SD10491, FX5U/FX5UC CPU module: SD10476 to SD10507). For error codes, refer to the manual of each device.

An error code is stored in Simple CPU communication error response code only when CFB4H is stored in Simple PLC Communication error code (FX5S/FX5UJ CPU module: SD10412 to SD10427, FX5U/FX5UC CPU module: SD10412 to SD10443). (Only when the communication target is other company's PLC, MODBUS/TCP-compatible devices, or MELSEC FX3 (Ethernet Block/Adapter) and an error occurs)

CFB4H is stored as the Simple PLC Communication error code in the following cases.

❶When an abnormal response is returned

❷When an abnormal response message is received (for example, when communicating in binary format, receiving a response message in ASCII format)

In case of ❶, an error code is stored in Simple CPU communication error response code. In the case of ❷, 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	• 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.	• 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.	• 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.	• 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	• The SD memory card has not been inserted. • The SD memory card is disabled. • The SD memory card turned to disable status by SM606 (SD memory card forcibly disable command).	• Insert the SD memory card or insert it again. • Slide down the switch for stopping SD memory card usage. • Cancel the SD memory card forced disable instruction.
4C40H	• 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.	• 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.	• Check the operation of the external device. • Check whether the connection cable is not disconnected.
C616H	Control port connection to the FTP server failed.	• 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.	• 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.	• 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.	• 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.	• 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.	• 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.	• 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.
1134H	The external device does not send an ACK response in the TCP/IP communication.	• 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 whether the connection cable is not disconnected.
2160H	Overlapping IP addresses were detected.	Check and correct the IP addresses.
2250H (Stored 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	• 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.
C020H	The send/receive data length exceeds the allowable range.	• 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.	• 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	• Description of control data is not correct. • Open instruction was executed through open settings parameter even though parameters are not set.	• Correct the descriptions of the control data. • Set the open settings parameters. Or, execute the OPEN instruction through control data.

Error code (hexadecimal)	Error details and causes	Action
C027H	Socket communication send message has failed.	• 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 whether the connection cable is not 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	• Description of control data is not correct. • Open instruction was executed through open settings parameter even though parameters are not set.	• 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.	• Check the operation of the external device. • Check whether the connection cable is not 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)	• 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.	• 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 whether the connection cable is not 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.	• 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.	• 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.	• 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.
C1CCH	• A response of the data length that exceeds the allowable range was received by the SLMPSND instruction. • The specified request data is incorrect.	• 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.	• 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 whether the connection cable is not 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.

Error code (hexadecimal)	Error details and causes	Action
C1D0H	The requested module I/O No. of the dedicated instruction is incorrect.	• 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.	• 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".	• 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	• 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.	• 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.	• 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.	• 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.	• Check the maximum data length that can be set to the data length storage area, and specify a value that is equal to or smaller than the maximum data length. • 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.	• 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.
CEE1H	Incorrect frame is received.	• 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.	• 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.

Error code (hexadecimal)	Error details and causes	Action
CF10H	Incorrect frame is received.	• 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	• 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.	• 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.	• Check the operation of the target device. • Check whether the connection cable is not disconnected.
CF71H	A timeout error has occurred.	• 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 800 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	• 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.	• 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 whether the connection cable is not disconnected.
C020H	The data length exceeds an allowable range.	• 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.	• 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 whether the connection cable is not 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.

Error code (hexadecimal)	Error details and causes	Action
C035H	The existence of the external device could not be confirmed within the response monitoring timer value.	• Check the operation of the external device. • Reexamine and change the set values for existence confirmation. • Check whether the connection cable is not 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.	• 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 text).	After reviewing and correcting the content of text or length of request data in the header, send to Ethernet module again.
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 the content of text or length of request data in the header, send to Ethernet module again.
C059H	• Error in command or subcommand specification. • There is a command or subcommand that cannot be used by the Ethernet module.	• Reconsider request contents. • Send command or subcommand that can be used by the Ethernet module.
C05BH	Ethernet module cannot read or write data from/to specified device.	Reconsider device to read or write.
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.	• 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.	• 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 reviewing and correcting the 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 returns.)	• Send 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.	• 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 whether the connection cable is not 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	• 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 number.

Error code (hexadecimal)	Error details and causes	Action
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.	• 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.	• 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.	• 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.	• Check the operation of the target device. • Check whether the connection cable is not disconnected.
CF71H	A timeout error occurred.	• 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.	• Check the operation of the external device. • Check whether the connection cable is not disconnected.	○	○
C050H	When the communication data code is set to "ASCII", ASCII code data which cannot be converted to binary is received.	• 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 the content of text or length of request data in the header, send to CPU module again.	○	—
C059H	• Error in command or subcommand specification. • There is a command or subcommand that cannot be used by the CPU module.	• 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. (Error related to device specification, such as reading or writing by 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.	• 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 reviewing and correcting the 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 returns.)	• Send 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 lock and 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.	• Check the operation of the external device. • Reexamine and change the set values for existence confirmation. • Check whether the connection cable is not disconnected.	○	○
C050H	When the communication data code is set to "ASCII", ASCII code data which cannot be converted to binary is received.	• 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 text).	After reviewing and correcting the content of text or length of request data in the header, send to Ethernet module again.	○	○
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 the content of text or length of request data in the header, send to Ethernet module again.	○	—
C059H	• Error in command or subcommand specification. • There is a command or subcommand that cannot be used by the CPU module.	• Reconsider request contents. • Send command or subcommand that can be used by the Ethernet module.	○	○
C05BH	Ethernet module cannot read or write data from/to specified device.	Reconsider device to read or write.	○	○
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.	• 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.	• 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 reviewing and correcting the 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 returns.)	• Send 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" screen of the Ethernet module. (☞ Page 797 Module diagnostics)
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	• Check the operation of the external device. • Check if the connection cable is disconnected.
0904H	Socket communication send message has failed.	• Check the operation of the external device. • Check if the connection cable is disconnected.
0910H	Data cannot be sent to the external device.	• 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.	• 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.)	• 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.	• 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 to 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.	• 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.	• Check the specifications of the communication destination device. • Check the communication packet.
1871H	Unsupported BVLL (BVLC type) was received.	• Check the communication packet. • Confirm with the manufacturer of the communication destination device.
1872H	Unsupported BVLL (BVLC function) was received.	• Check the communication packet. • Confirm with the manufacturer of the communication destination device.
1873H	0 was specified for DNET of the received packet.	• Check the specifications of the communication destination device. • Check the communication packet.
1874H	0 or 65535 was specified for SNET of the received packet.	• Check the specifications of the communication destination device. • Check the communication packet.
1875H	0 was specified for SLEN of the received packet.	• Check the specifications of the communication destination device. • Check the communication packet.
1876H	An error occurred during packet decoding.	• 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.	• Check the communication packet. • Please confirm with the manufacturer of the communication destination device.

Error code (hexadecimal)	Error details and causes	Action
1900H	A memory check error has occurred.	• 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	A memory check error has occurred.	• 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	A memory check error has occurred.	• 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	A memory check error has occurred.	• 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.	• 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.	• 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.	• 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.	• 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.	• 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.	• 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.	• 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.	• 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.	• 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.	• Check the external device configuration settings in the Ethernet 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.	• Check the external device configuration settings in the Ethernet 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.	• Check the external device configuration settings in the Ethernet module parameters. • If the same error is displayed again, the abnormal module may have a hardware problem. Please consult your local Mitsubishi representative.
2CF0H	An 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 external device configuration settings 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 connections 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 groups 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	The gateway address (Ipv6) is outside the 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	IP addresses of the master station and the remote station 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.	• 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.	• 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	• The specified device No. is out of the range. • The external device does not correspond to the specified device name.	• 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.	• 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.

Error code (hexadecimal)	Error details and causes	Action
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.

47.2 Serial Communication

Checking absence/presence of N:N Network function errors

Verify that link errors have not occurred in the master station and local stations. The error can be checked using the flags specified below.

Serial communication error

■Error flags

If a communication error occurs in the N:N Network, the serial communication error flag turns ON.

Verify that the device specified in the table below is ON.

FX5 only				FX3 Series compatible		Name	Descriptions
CH1	CH2	CH3	CH4	CH1	CH2		
SM8500	SM8510	SM8520	SM8530	SM8063	SM8438	Serial communication error	Turns ON when an error occurs in serial communication.

■Error codes

When the serial communication error flag turns ON, the error code is stored in the device specified in the table below.

FX5 only				FX3 Series compatible		Name	Descriptions
CH1	CH2	CH3	CH4	CH1	CH2		
SD8500	SD8510	SD8520	SD8530	SD8063	SD8438	Serial communication error code	When a serial communication error occurs, the error code is stored.

For error code above is stored, refer to ☞ Page 834 Error codes.

Precautions

Serial communication errors are not cleared even after communication has recovered its normal status.

Devices are cleared when power is turned OFF→ON, STOP→RUN, reset or SM50 (Error Detection Reset Completion) is turned ON.

Data transfer sequence error

■Check while data transfer sequence is being executed

Verify that the device specified in the table below is ON.

While N:N Network is being executed, the data transfer sequence ON flag remains ON.

FX5 only	FX3 Series compatible	Name	Descriptions
SM9056	SM8191	Data transfer sequence ON	Remains ON while data transfer is being executed.

■Error flags

When a link error occurs in the master station or local station of the N:N Network, the data transfer sequence error flag turns ON.

Verify that the device specified in the table below is ON.

Station number	Master station	Local stations						
	Station No. 0	Station No. 1	Station No. 2	Station No. 3	Station No. 4	Station No. 5	Station No. 6	Station No. 7
FX5 only	SM9040	SM9041	SM9042	SM9043	SM9044	SM9045	SM9046	SM9047
FX3 Series compatible	SM8183	SM8184	SM8185	SM8186	SM8187	SM8188	SM8189	SM8190

■Error codes

When the data transfer sequence error turns ON, the error code is stored in the following device.

Station number	Master station	Local stations						
	Station No. 0	Station No. 1	Station No. 2	Station No. 3	Station No. 4	Station No. 5	Station No. 6	Station No. 7
FX5 only	SD9061	SD9062	SD9063	SD9064	SD9065	SD9066	SD9067	SD9068
FX3 Series compatible	SD8211	SD8212	SD8213	SD8214	SD8215	SD8216	SD8217	SD8218

For error code above is stored, refer to ☞ Page 834 Error codes.

Error codes

The error codes (hexadecimal) stored in the devices for error code are as follows. When an error occurs, refer to the error code list shown below to confirm the check points.

M: Master station, L: Local station

Error codes (Hexadecimal)	Error name	Station in which error occurred	Station which detected error	Description of error	Check point
2221H[*1]	Parameter error	M, L	M, L	The parameter read from file is incorrect.	Parameter setting by GX Works3
7701H	Send monitoring timeout in local station	L	M	Local station did not respond to the request from the master station within the monitoring time.	• Wiring • Power
7702H	Station number error	L	M	An unexpected local station responded to the request by the master station.	• Station number settings
7703H	Counter error	L	M	The counter value included in the parameter data is different from the counter value given by a local station.	• Wiring
7704H	Message format error in local station	L	M	The message given by the local station is incorrect.	• Wiring • Power • Station number settings
7705H	Insufficient device points for local stations error	M, L	M[*2], L[*2]	The link device point number corresponding to the number of local stations cannot be secured.	• Number of local stations • Station number settings • Link device head number
7711H	Send monitoring timeout in master station	M	L	The master station did not send request to the next local station within the monitoring timer.	• Wiring • Power
7714H	Message format error in master station	M	L	The message from the master station is incorrect.	• Wiring • Power • Station number settings
7715H	Insufficient link device points for local station error	L	L[*2]	The link device corresponding to the local station number cannot be secured (If the error occurs at the same as 05H, 15H is stored).	• Number of local stations • Station number settings • Link device head number
7721H	Local station no response error	L	L[*3]	The specified local station does not exist.	• Wiring • Power • Station number settings
7722H	Station number error in local station	L	L[*3]	An unexpected local station responded to the request by the master station.	• Station number settings
7723H	Counter error in local station	L	L[*3]	The counter value included in the parameter data is different from the count value given by a local station.	• Wiring
7731H	Parameter not received	L	L[*3]	Before parameters had been received, sending request was received from the master station.	• Wiring • Power

[*1] Not supported for data transfer sequence error.
[*2] Station in which error occurred
[*3] Any local station other than the local station in which error occurred

Checking absence/presence of parallel link function errors

Confirm whether an error has not occurred in the PLCs at the master or linked station. To confirm, check the following flags.

Parallel link operation

"Parallel link operation" flag is ON while the parallel link is running.

Verify that the device specified in the table below is ON.

FX5 only	FX3 Series compatible	Name	Descriptions
SM9090	SM8072	Parallel link operation	On: In normal running state Off: In stopped state

Serial communication error

■Error flags

If a communication error occurs in the parallel link, the serial communication error flag turns ON.

Verify that the device specified in the table below is ON.

FX5 only				FX3 Series compatible		Name	Descriptions
CH1	CH2	CH3	CH4	CH1	CH2		
SM8500	SM8510	SM8520	SM8530	SM8063	SM8438	Serial communication error	Turns ON when an error occurs in serial communication.

■Error codes

When the serial communication error flag turns ON, the error code is stored in the device specified in the table below.

FX5 only				FX3 Series compatible		Name	Descriptions
CH1	CH2	CH3	CH4	CH1	CH2		
SD8500	SD8510	SD8520	SD8530	SD8063	SD8438	Serial communication error code	Stores the error code when the serial communication error occurs.

Error code

The error codes (hexadecimal) stored in the devices for error code are as follows. When an error occurs, check the following error codes.

Error code (Hexadecimal)	Error name	Error handling
2220H	Parameter not detected	Write the project again.
2221H	Parameter error	Correct the setting value of the parameter and write the project again.
7001H	Communication CH parameter setting duplication error	Confirm whether the serial port is not used for other communication. (☞ Page 877)
7010H	Parity error, overrun error or framing error	Check the wiring. (☞ Page 292)
7812H	Parallel link character error	
7813H	Parallel link sum error	
7814H	Parallel link format error	

Checking error codes of MC protocol function

1C frame

■Error codes when NAK is sent

The table below shows error codes (hexadecimal) and contents of errors when NAK is sent in communication between the counterpart equipment and a CPU module. As an error code, a 2-digit ASCII code (hexadecimal) within the range from 00H to FFH is sent. When two or more errors occur at the same time, priority is given to the error code which occurred first, and the error code are sent in chronological order. Further, in case of an occurrence of the following errors, initialize the entire transmission sequence.

Error code (hexadecimal)	Error item	Description of error	Action
02H	Sum check error	Sum check error has occurred. The sum check code included in the received data does not match the sum check created from the received data.	Check the data sent from the computer and the contents of the sum check. Modify either one, and then execute communication again.
03H	Protocol error	The communication protocol is abnormal. A control procedure set using parameters was ignored, and a different control procedure was adopted in communication. Or the adopted control procedure was partially different from the preset control procedure. Or a command specified in the preset control procedure does not exist.	1) Check the contents of parameters and the contents of control procedure. Modify either contents, and then execute communication again. 2) Refer to the command details, modify the specified command etc., and then execute communication again. (☞ Page 684)
06H	Character area error	An error occurred in the character area A, B or C. 1) The control procedure set using parameters is different. 2) A specified device number does not exist in the target PLC. 3) A device number is not set with the specified number of characters (5 or 7 characters).	1) Check the contents of the character areas A, B and C, modify the contents if necessary, and then execute communication again. 2) Modify the device range, and then execute communication again. (☞ Page 323)
07H	Character error	ASCII code data to be written to a device is not hexadecimal.	Check the data to be written to the device, modify it if necessary, and then execute communication again.
10H	PLC number error	A station with the corresponding PLC number does not exist.	Check the PLC number included in the message, modify it if necessary, and then execute communication again. The PLC number should be "FFH" in all FX5 PLCs.
18H	Remote control error	Remote control to set the RUN or STOP mode is disabled. Remote STOP is operated by other engineering tools or peripheral devices.	Operate Remote RUN in the engineering tools or peripheral devices where Remote STOP was operated and operate Remote RUN/STOP again.

■Error codes in CPU

When an error is included in a message from the counterpart equipment in the communication between counterpart equipment and CPU unit, a serial communication error occurs in the CPU unit.

- Error flags

If a communication error occurs in MC protocol, the serial communication error flag turns ON.

Verify that the device specified in the table below is ON.

CH1	CH2	CH3	CH4	Name	Descriptions
SM8500	SM8510	SM8520	SM8530	Serial communication error	Turns ON when an error occurs in serial communication.

- Error codes

When the serial communication error turns ON, the error code is stored in the device specified in the table below.

CH1	CH2	CH3	CH4	Name	Descriptions
SD8500	SD8510	SD8520	SD8530	Serial communication error code	Stores the error code when the serial communication error occurs.

List of error codes is as follows.

Error code (hexadecimal)	Error item	Description of error	Action
7001H	Serial communication common error	Communication CH parameter setting duplication error	Confirm whether the serial port is not used for other communication.
7010H	Serial communication common error	Parity error, overrun error or framing error	Check the following. • Wiring • Communication setting
7143H	Device number error	Head device address is outside the range.	Check the sent message of the counterpart equipment, modify it if necessary, and then execute communication again.
7164H	Request contents error	There is an error in the request contents or the device specification method.	Check the sent message/request contents of the counterpart equipment, modify it if necessary, and then execute communication again.
7E40H	Command error	A non-existing command or sub-command is specified.	Check the sent message of the counterpart equipment, modify it if necessary, and then execute communication again.
7E41H	Data length error	The number of points that can be exchanged at the time of random reading/writing are exceeded and specified.	Check the sent message of the counterpart equipment, modify it if necessary, and then execute communication again.
7E42H	Data count error	The number of points requested are exceeding the range for the command.	Check the sent message of the counterpart equipment, modify it if necessary, and then execute communication again.
7E43H	Device error	• Non-existing device is specified. • A device that cannot be specified in the pertinent command is specified. • Head device address is outside the range.	Check the sent message of the counterpart equipment, modify it if necessary, and then execute communication again.
7F21H	Receiving header part error	• There is a specification error in the command (frame) part. • An ASCII code that cannot be converted to binary code is received.	Check the sent message of the counterpart equipment, modify it if necessary, and then execute communication again.
7F22H	Command error	• Non-existing command or device is specified. • There is an error in the remote password length.	Check the sent message of the counterpart equipment, modify it if necessary, and then execute communication again.
7F23H	MC protocol message error	There is no data (ETX, CR-LF etc.) at the end of character part or there is an error in the specification.	Check the sent message of the counterpart equipment, modify it if necessary, and then execute communication again.
7F24H	Sum check error	• The received sum check code does not match with the calculated sum check code. • The received horizontal parity code does not match with the calculated horizontal parity code. • The received error check code does not match with the calculated error check code.	• Check the sum check code of the counterpart equipment. • Check the horizontal parity code of the counterpart equipment. • Check CRC-16 of the counterpart equipment.
7F26H	Command error	At the time of registering a remote password, a different command is received before unlock process.	After the unlock process ends normally, execute communication.
7F40H	Timeout error	A timeout occurred because the received message was insufficient.	• Verify that there is no data omission in the reception error. • Verify that the reception is not stopped due to DTR control etc. • Check the transmission program on the counterpart device for any missing messages.
7F67H	Overrun error	The following data is received before FX5 PLC completed receiving process.	Reduce the communication speed and then execute communication again.
7F68H	Framing error	• A disturbance occurred in the line due to power supply ON/OFF of the target station. • There is noise on the line. • At the time of multi drop connection, the data is sent simultaneously from multiple devices.	• Take counter measures for noise. • At the time of multi drop connection, have an interlock so that the data is not sent simultaneously from multiple devices.
7F69H	Parity error	• The parity bit settings do not match. • A disturbance occurred in the line due to power supply ON/OFF of the target station. • There is noise on the line. • At the time of multi drop connection, the data is sent simultaneously from multiple equipments.	• Match the settings of FX5 PLC and counterpart equipment. • Take counter measures for noise. • At the time of multi drop connection, have an interlock so that the data is not sent simultaneously from multiple devices.

3C/4C frame

■Error codes when NAK is sent

The table below shows error codes (hexadecimal) and contents of errors when NAK is sent in communication between the counterpart equipment and a CPU module. When two or more errors occur at the same time, priority is given to the error code with the smallest number, and the error code with the smallest number is sent. Further, in case of an occurrence of the following errors, initialize the entire transmission sequence.

Error code (hexadecimal)	Error item	Description of error	Action
4000H to 4FFFH	—	An error detected by the CPU unit. (An error occurring outside communication by MC protocol)	Refer to following manual and take measures. 📖MELSEC iQ-F FX5S/FX5UJ/FX5U/FX5UC User's Manual (Hardware)
7143H	Device number error	Head device address is outside the range.	Check the sent message of the counterpart equipment, modify it if necessary, and then execute communication again.
7164H	Request contents error	There is an error in the request contents or the device specification method.	Check the sent message/request contents of the counterpart equipment, modify it if necessary, and then execute communication again.
7167H	Disabled in RUN mode	Write command is specified when write not possible in RUN mode is set.	Change the settings to make writing in RUN mode possible, and then execute communication again.
7168H		A command that cannot be executed in RUN mode is specified.	STOP the CPU and then execute communication again.
7E40H	Command error	A non-existing command or sub-command is specified.	Check the sent message of the counterpart equipment, modify it if necessary, and then execute communication again.
7E41H	Data length error	The number of points that can be exchanged at the time of random reading/writing are exceeded and specified.	Check the sent message of the counterpart equipment, modify it if necessary, and then execute communication again.
7E42H	Data count error	The number of points requested are exceeding the range for the command.	Check the sent message of the counterpart equipment, modify it if necessary, and then execute communication again.
7E43H	Device error	• Non-existing device is specified. • A device that cannot be specified in the pertinent command is specified.	Check the sent message of the counterpart equipment, modify it if necessary, and then execute communication again.
7F21H	Receiving header part error	• There is a specification error in the command (frame) part. • An ASCII code that cannot be converted to binary code is received.	Check the sent message of the counterpart equipment, modify it if necessary, and then execute communication again.
7F22H	Command error	There is an error in the remote password length.	Check the sent message of the counterpart equipment, modify it if necessary, and then execute communication again.
7F23H	MC protocol message error	There is no data (ETX, CR-LF etc.) at the end of character part or there is an error in the specification.	Check the sent message of the counterpart equipment, modify it if necessary, and then execute communication again.
7F24H	Sum check error	• The received sum check code does not match with the calculated sum check code. • The received horizontal parity code does not match with the calculated horizontal parity code. • The received error check code does not match with the calculated error check code.	• Check the sum check code of the counterpart equipment. • Check the horizontal parity code of the counterpart equipment. • Check CRC-16 of the counterpart equipment.
7F26H	Command error	At the time of registering a remote password, a different command is received before unlock process.	After the unlock process ends normally, execute communication.
7F40H	Timeout error	A timeout occurred because the received message was insufficient.	• Verify that there is no data omission in the reception error. • Verify that the reception is not stopped due to DTR control etc. • Check the transmission program on the counterpart device for any missing messages.
7F67H	Overrun error	The following data is received before FX5 PLC completed receiving process.	Reduce the communication speed and then execute communication again.

Error code (hexadecimal)	Error item	Description of error	Action
7F68H	Framing error	• A disturbance occurred in the line due to power supply ON/ OFF of the target station. • There is noise on the line. • At the time of multi drop connection, the data is sent simultaneously from multiple devices.	• Take counter measures for noise. • At the time of multi drop connection, have an interlock so that the data is not sent simultaneously from multiple devices.
7F69H	Parity error	• The parity bit settings do not match. • A disturbance occurred in the line due to power supply ON/ OFF of the target station. • There is noise on the line. • At the time of multi drop connection, the data is sent simultaneously from multiple devices.	• Match the settings of FX5 PLC and counterpart equipment. • Take counter measures for noise. • At the time of multi drop connection, have an interlock so that the data is not sent simultaneously from multiple devices.
7FE6H	Remote password error	Remote password does not match.	Check the remote password error, and then execute communication again.

■Error codes in CPU

When an error is included in a message from the counterpart equipment in the communication between counterpart equipment and CPU unit, a serial communication error occurs in the CPU unit.

- Error flags

If a communication error occurs in MC protocol, the serial communication error flag turns ON.

Verify that the device specified in the table below is ON.

CH1	CH2	CH3	CH4	Name	Descriptions
SM8500	SM8510	SM8520	SM8530	Serial communication error	Turns ON when an error occurs in serial communication.

- Error codes

When the serial communication error turns ON, the error code is stored in the device specified in the table below.

CH1	CH2	CH3	CH4	Name	Descriptions
SD8500	SD8510	SD8520	SD8530	Serial communication error code	Stores the error code when the serial communication error occurs.

The following table shows the error code (hexadecimal) stored in each device.

Error code (hexadecimal)	Description
0000H	No error detected
7F40H	Monitoring time out
7F67H	Overrun error
7F68H	Framing error
7F69H	Parity error

When an error code above is stored, check the following items:

Name	Description of error	Action
Parity, overrun or framing error	The transfer data is abnormal.	Check the transfer specifications set using parameters, and execute communication again.
Monitoring time out	The received message was insufficient. Because normal message was not received within the time-out time setting, the transfer sequence was initialized.	The message is insufficient. Modify the transfer program in the counterpart equipment to execute the communication again.

Point

The serial communication error flags and serial communication error codes are not cleared even when communication is restored.

They are cleared by turning the PLC power OFF→ON, STOP→RUN or by resetting the system.

Operation error

■Error flags

The operation error flag turns ON when an operation error occurs in the communication settings.

Verify that the device specified in the table below is ON.

FX5 only				Name	Descriptions
CH1	CH2	CH3	CH4		
SM0				Latest self diagnostics error (including annunciator ON)	Turns ON when an operation error occurs.
SM1				Latest self diagnostics error (not including annunciator ON)	
SM56, SM8067				Operation error	

■Error codes

When the operation error flag turns ON, the error code (hexadecimal) is stored in the operation error code (SD0/SD8067).

The error codes stored in the device are shown as follows.

Error code (hexadecimal)	Name	Description of error	Error handling
2222H	Parameter error	The parameter setting value exceeded the available range.	Correct the setting value of the parameter and write the project again.

Checking absence/presence of inverter communication function errors

If an inverter communication error occurs, it is processed as a serial communication error. (☞ Page 370 Processing of communication errors)

Serial communication error

■Error flags

The following devices turn ON when an error occurs during serial communication.

FX5 dedicated				**FX3 Series compatible**		**Name**	**Descriptions**
CH1	**CH2**	**CH3**	**CH4**	**CH1**	**CH2**		
SM8500	SM8510	SM8520	SM8530	SM8063	SM8438	Serial communication error	Turns ON when an error occurs in serial communication.

■Error codes

When the serial communication error turns ON, the error code is stored in the corresponding device specified in the table below.

FX5 dedicated				**FX3 Series compatible**		**Name**	**Descriptions**
CH1	**CH2**	**CH3**	**CH4**	**CH1**	**CH2**		
SD8500	SD8510	SD8520	SD8530	SD8063	SD8438	Serial communication error code	When a serial communication error occurs, the error code is stored.

The error code (hexadecimal) is below.

Error code (hexadecimal)	Description
7010H	Parity error, overrun error or framing error
76**H	Inverter communication error (☞ Page 842)

■List of error codes for inverter communication

The error codes shown below are stored when a communication error is caused by an inverter communication instruction.

Error code (hexadecimal)	Description of error		Inverter operation
0000H	Normal end	(No Errors)	—
7601H	—	No response from the inverter.	
7602H	Timeout error	Sending from inverter was aborted midway.	
7603H	Station number error	An unspecified station gave response.	
7604H	Sum check error	The sum of data sent back by the inverter did not match.	
7608H	Sending timeout error	Sending to the inverter was not completed within the specified time.	
7609H	Receive data error	Wrong data was received from the inverter.	
7620H	Computer NAK error	Inverter sent the error code H0. The number of retries exceeded the allowable number because of an error in the transfer request data.	When errors have occurred consecutively beyond the allowable number of retries, inverter is brought to an alarm stop.
7621H	Parity error	Inverter sent the error code H1. The contents are different from the specified parity.	
7622H	Sum check error	Inverter sent the error code H2. The sum check code in the computer is different from the sum value calculated from the data received by the inverter.	
7623H	Protocol error	Inverter sent the error code H3. There is a syntax error in data received by the inverter. Or, data reception is not completed within a certain amount of time. CR/LF does not match parameter setting.	
7624H	Framing error	Inverter sent the error code H4. The stop bit length is different from the initial set value.	
7625H	Overrun error	Inverter sent the error code H5. Before receiving the completed data in the inverter, the computer sent the next set of data.	
7626H	Undefined	Inverter sent the error code H6. Not defined currently in inverter.	—
7627H	Character error	Inverter sent the error code H7. An unused character (other than 0 to 9, A to F and control codes) was received.	Inverter does not accept the received data, but is not brought to an alarm stop.
7628H	Undefined	Inverter sent the error code H8. Not defined currently in inverter.	—
7629H	Undefined	Inverter sent the error code H9. Not defined currently in inverter.	
762AH	Mode error	Inverter sent the error code HA. A parameter was written in a mode other than computer link operation mode, or while the inverter was operating.	Inverter does not accept the received data, but is not brought to an alarm stop.
762BH	Instruction code error	Inverter sent the error code HB. Non-existing instruction code was specified.	
762CH	Data range error	Inverter sent the error code HC. In writing a parameter or operation frequency, data outside the allowable range was specified.	
762DH	Undefined	Inverter sent the error code HD. Not defined currently in inverter.	—
762EH	Undefined	Inverter sent the error code HE. Not defined currently in inverter.	
762FH	Undefined	Inverter sent the error code HF. Not defined currently in inverter.	

■IVMC instruction error codes

When errors occur in the send data of the IVMC instruction, the following error codes are stored.

Error code (hexadecimal)	Description of error	Inverter operation
0000H	No error has occurred in both send data 1 and send data 2. Normal end	—
7640H	IVMC instruction send data 1: Mode error A parameter was written in a mode other than computer link operation mode or while inverter was operating.	The inverter accepts data corresponding to send data 2. It is not brought to an alarm stop.
7641H	IVMC instruction send data 1: Instruction code error Non-existing instruction code was specified.	
7642H	IVMC instruction send data 1: Data range error Data outside the allowable range was specified.	
7643H	IVMC instruction send data 2: Mode error A parameter was written in a mode other than computer link operation mode or while inverter was operating.	The inverter accepts data corresponding to send data 1. It is not brought to an alarm stop.
7644H	IVMC instruction send data 2: Instruction code error Non-existing instruction code was specified.	
7645H	IVMC instruction send data 2: Data range error Data outside the allowable range was specified.	
7646H	IVMC instruction send data 1: Mode error IVMC instruction send data 2: Mode error	The inverter does not accept data corresponding to send data 1 and 2, but is not brought to an alarm stop.
7647H	IVMC instruction send data 1: Mode error IVMC instruction send data 2: Instruction code error	
7648H	IVMC instruction send data 1: Mode error IVMC instruction send data 2: Data range error	
7649H	IVMC instruction send data 1: Instruction code error IVMC instruction send data 2: Mode error	
764AH	IVMC instruction send data 1: Instruction code error IVMC instruction send data 2: Instruction code error	
764BH	IVMC instruction send data 1: Instruction code error IVMC instruction send data 2: Data range error	
764CH	IVMC instruction send data 1: Data range error IVMC instruction send data 2: Mode error	
764DH	IVMC instruction send data 1: Data range error IVMC instruction send data 2: Instruction code error	
764EH	IVMC instruction send data 1: Data range error IVMC instruction send data 2: Data range error	

Operation error

■Error flags

The operation error flag turns ON when an operation error occurs in the inverter communication instruction. Verify that the device specified in the table below is ON.

FX5 dedicated				Name	Descriptions
CH1	CH2	CH3	CH4		
SM0				Latest self-diagnostic error (Including the annunciator ON)	Turns ON when an operation error occurs.
SM1				Latest self-diagnostic error (Not including the annunciator ON)	
SM56, SM8067				Operation error	

■Error codes

When the operation error flag turns ON, the error code (hexadecimal) is stored in the operation error code (SD0/SD8067). For the error codes stored in the device, refer to the operation error of each inverter communication instruction.

Checking absence/presence of non-protocol communication function errors

Serial communication error

■Error flags

If a communication error occurs in the non-protocol communication, the serial communication error flag turns ON.

Verify that the device specified in the table below is ON.

FX5 only				FX3 Series compatible		Name	Descriptions
CH1	CH2	CH3	CH4	CH1	CH2		
SM8500	SM8510	SM8520	SM8530	SM8063	SM8438	Serial communication error	Turns ON when an error occurs in serial communication.

■Error codes

When the serial communication error turns ON, the error code is stored in the corresponding device specified in the table below.

FX5 dedicated				FX3 Series compatible		Name	Descriptions
CH1	CH2	CH3	CH4	CH1	CH2		
SD8500	SD8510	SD8520	SD8530	SD8063	SD8438	Serial communication error code	When a serial communication error occurs, the error code is stored.

The error codes (hexadecimal) stored in devices are shown as follows.

Error code (hexadecimal)	Name	Description of error	Error handling
7010H	Parity error, overrun error or framing error	A parity error, overrun error or framing error occurred during communication.	When an error code is stored, check the following items: • Wiring • Communication settings (☞ Page 420)
7200H	Communication data sum mismatch	Receive data sum mismatch.	
7201H	Incorrect data format	Incorrect data format • Character other than LF was received after CR in control procedure [CR, LF added]. • Character other than CR was received after the check sum in sum check code [Added] and control procedure [CR, LF added]. • The amount of data was greater than the set value. • CR, LF was received before the terminator.	

Operation error

■Error flags

The operation error flag turns ON when an operation error occurs in the RS2 instruction.

Verify that the device specified in the table below is ON.

FX5 only				Name	Descriptions
CH1	CH2	CH3	CH4		
SM0				Latest self-diagnostic error (Including the annunciator ON)	Turns ON when an operation error occurs.
SM1				Latest self-diagnostic error (Not including the annunciator ON)	
SM56, SM8067				Operation error	

■Error codes

When the operation error flag turns ON, the error code (hexadecimal) is stored in the operation error code (SD0/SD8067).

The error codes stored in the device are shown as follows.

Error code (hexadecimal)	Name	Description of error	Error handling
2822H	A device was specified that cannot be specified with the instruction.	Wrong operand set for the RS2 instruction.	When an error code is stored, check the following items: • Program • Communication settings (☞ Page 420)
3405H	Data was entered outside of the range that can be specified.	The operand device number range or data value set for the RS2 instruction is out of range.	
2820H	A device or label specified with the instruction was entered outside of the range that can be used.	The operand device number range or data value set for the RS2 instruction is out of range.	
1810H	Operation error	The same channel is used for more than one RS2 instruction.	
3600H	Operation error	Specified channel is not set in parameter.	

Checking absence/presence of predefined protocol support function errors

Serial communication error

■Error flags

If a communication error occurs in the predefined protocol support function, the serial communication error flag turns ON. Verify that the device specified in the table below is ON.

CH1	CH2	CH3	CH4	Name	Descriptions
SM8500	SM8510	SM8520	SM8530	Serial communication error	Turns ON when an error occurs in serial communication.

■Error codes

When the serial communication error turns ON, the error code is stored in the corresponding device specified in the table below. And be stored in resulting of executed S(P).CPRTCL instruction (operand (s)).

CH1	CH2	CH3	CH4	Name	Descriptions
SD8500	SD8510	SD8520	SD8530	Serial communication error code	When a serial communication error occurs, the error code is stored.

The error codes (hexadecimal) stored in devices are shown as follows.

Error code (hexadecimal)	Name	Description of error	Error handling
7D00H	Protocol No. setting error	An out-of-range value is set as the number of protocols to be executed continuously, which is set by an argument of the S(P).CPRTCL instruction.	Check the set protocol No.
7D02H	Protocol not ready error	• The S(P).CPRTCL instruction was executed while the predefined protocol is not ready (SD9102 = 0). • The S(P).CPRTCL instruction was executed while the protocol setting data has an error.	• Execute the S(P).CPRTCL instruction while the predefined protocol is ready (SD9102 = 1). • Write the protocol setting data again to the CPU module, and execute the S(P).CPRTCL instruction. • If this error persists even after the data is rewritten, replace the module.
7D10H	Protocol unregistration error	• The control data of the S(P).CPRTCL instruction specified a protocol number not registered in the CPU module. • The S(P).CPRTCL instruction was executed without the protocol setting data written.	• Check the specified protocol No. • Check that the protocol number specified to Protocol registration (SD9132 to SD9135) is registered. • Write the protocol setting data, and execute the S(P).CPRTCL instruction.
7D12H	Send monitoring time timeout error	• Send monitoring time has timed out. • Although the transmission was retried for the specified number of retries, it did not succeed.	Check if the cables are disconnected.
7D13H	Receive wait timeout error	Receive wait time has timed out.	• Check if the cable is disconnected. • Check if any errors occur on the counterpart device side.
7D16H	Protocol cancel request error	A cancel request was accepted during protocol execution, and the S(P).CPRTCL instruction was completed abnormally.	Check the canceled protocol in the S(P).CPRTCL instruction's control data (number of executions), and eliminate the cause of the cancellation.
7D17H	Packet size error	A packet of more than 2048 bytes was received.	• Check the data sent from the counterpart device. • Divide the packet data into several parts and send them separately.
7D18H	Insufficient digit number error	When data is received using a protocol including a packet that contains conversion variable (variable number of data), the corresponding data is insufficient.	• Check the data sent from the counterpart device. • Check the packet format of the counterpart device to confirm that the number of digits is set correctly.
7D19H	Abnormal digit number error	When data is received using a protocol including a packet that contains conversion variable (fixed number of data and variable number of digits), the corresponding data corresponding is 0 bytes (0 digits) or exceeds the maximum number of digits.	

Error code (hexadecimal)	Name	Description of error	Error handling
7D1AH	Data length error	In the data received from the counterpart device, the Length value does not match the data length of the conversion variable(s).	• Check the packet format of the counterpart device to confirm that the conversion variable is set correctly. ■Check the data sent from the counterpart device to confirm that; • The Length value is correct. • Any of the data corresponding to the conversion variable is not missing.
7D1BH	Value range error	When data is received using a protocol including a packet that contains conversion variable, the corresponding data exceeds the range of values allowed for the CPU module.	• Check the data sent from the counterpart device. ■Check the packet format of the counterpart device to confirm that the conversion size is set correctly. • If the size exceeds the range for word data, change the conversion size setting to double word. • If the size exceeds the range for double word data, change the element to non-conversion variable.
7D20H	Data length size error, data count size error	The value set in the data length storage area or the data count storage area is out of range.	• Check the maximum data length that can be set to the data length storage area, and specify a value that is equal to or smaller than the maximum data length. • Check the maximum allowable data quantity, and specify the maximum quantity or less in the data quantity storage area.
7D21H	Decimal point place specification error	• An out-of-range decimal point position is set when the number of decimals is set to variable point. • The number of decimals is larger than the number of digits for each data.	• Check the set decimal point position. • Check the number of digits setting, and set the decimal point position so that the number of decimals is less than the number of digits.
7F20H	ASCII → BIN conversion error	• When data is received by conversion variable, the data corresponding to the conversion variable cannot be converted to binary numbers. • When data is received using a protocol including a packet of check code (ASCII hexadecimal or ASCII decimal), the data corresponding to the check code cannot be converted to binary numbers.	• Check the sent message of the counterpart device, modify it if necessary, and then execute communication again. • If the protocol was edited, check the packet format of the counterpart device to check that the set values of the conversion, sign character, number of decimals, delimiter, and number of digit for conversion variable are correct. • If the protocol was edited, check the packet format of the counterpart device to check that the setting for the code type and data length of check code are correct.
7F24H	Sum check error	• The received sum check does not match with the calculated sum check. • The received horizontal parity code does not match with the calculated horizontal parity code. • The received error check code does not match with the calculated error check code.	• Check the sum check of the counterpart device. • Check the horizontal parity code of the counterpart device. • Check CRC-16 of the counterpart device. ■If the protocol was edited, check if the following values match the packet format of the counterpart device: • Processing method or code type of the check code • Data length • Data flow • Complement calculation • Calculating range
7F67H	Overrun error	The following data was received before the CPU module completes receiving process.	• Reduce the communication speed and then execute communication again. • Check if momentary power failure has occurred on the station with the CPU module (using the special register SD1005). If momentary power failure has occurred, remove the cause.

Error code (hexadecimal)	Name	Description of error	Error handling
7F68H	Framing error	• The stop bit settings do not match. • A disturbance occurred in the line due to power supply ON/OFF of the target station. • There is noise on the line. • At the time of multi drop connection, the data is sent simultaneously from multiple devices.	• Match the settings of the CPU module with that of the counterpart device. • Take measures against noise. • In the multi drop connection, have an interlock so that data is not sent simultaneously from multiple devices.
7F69H	Parity error	• The parity bit settings do not match. • A disturbance occurred in the line due to power supply ON/OFF of the target station. • There is noise on the line. • At the time of multi drop connection, the data is sent simultaneously from multiple devices.	
7F6AH	Buffer full error	The receive buffer overflowed and received data was skipped.	Perform a predefined protocol including data reception to clear the receive buffer.
7FF2H	Predefined protocol setting error	The instruction cannot be executed with the current predefined protocol setting.	Correct the setting value of the predefined protocol.

Operation error

■Error flags

The operation error flag turns ON when an operation error occurs in the S(P).CPRTCL instruction.

Verify that the device specified in the table below is ON.

FX5 only				Name	Descriptions
CH1	CH2	CH3	CH4		
SM0				Latest self-diagnostic error (Including the annunciator ON)	Turns ON when an operation error occurs.
SM1				Latest self-diagnostic error (Not including the annunciator ON)	
SM56, SM8067				Operation error	

■Error codes

When the operation error flag turns ON, the error code (hexadecimal) is stored in the operation error code (SD0/SD8067).

The error codes stored in the device are shown as follows.

Error code (hexadecimal)	Name	Description of error	Error handling
2220H	Parameter not detected	The parameter contents are corrupted.	Write the project again.
2221H	Parameter error	The parameter setting value exceeded the available range.	Correct the setting value of the parameter and write the project again.
2222H			
2250H	Module extended parameter error	Any of the following errors exists in the written module extended parameter setting: • The predefined protocol setting has an error. • The data for the set item is corrupted.	• Check the predefined protocol setting in protocol setting data error (SD9120 to 9123), and correct the setting. Then write the setting to the CPU module and execute the S(P).CPRTCL instruction. • If the error occurs again even after rewriting, replace the CPU module.
2820H	Invalid device designation	The device used for operands of the instruction exceeded the device range.	Check the device range and correct the program.
3100H	Instruction code error	The program contains instructions that cannot be used or interpreted.	• Perform a module diagnosis on the engineering tool to check the details (program location). Check and modify the program (step) indicated by the error jump. • Take measures against noise. • Rewrite the program, reset the CPU module, and perform RUN. If the same error persists, it may be due to a hardware failure of the CPU module. Please consult your local Mitsubishi representative.
3405H	Operation error	The operand device number range or data value set for the S(P).CPRTCL instruction is out of range.	Check the data specified by the S(P).CPRTCL instruction.
3582H	Operation error	The instruction that is being used cannot be used in the interruption routine program.	Correct the program so that the instruction that cannot be used in the interruption routine program is not used.

47.3 MODBUS Serial Communication

Communication error

■Error flags

If a communication error occurs in the serial communication, the serial communication error flag turns ON.

Verify that the device specified in the table below is ON.

FX5 only				FX3 Series compatible		Name	Descriptions
CH1	CH2	CH3	CH4	CH1	CH2		
SM8500	SM8510	SM8520	SM8530	SM8063	SM8438	Serial communication error	Turns ON when an error occurs in serial communication.
—				SM8402	SM8422	MODBUS communication error	Turns ON when an error occurs in MODBUS communication.
				SM8403	SM8423	MODBUS communication error (latched)	Turns on once an error occurs during MODBUS serial communication.

■Error codes

When the serial communication error turns ON, the error code and error detail will be stored in the relevant device.

FX5 only				FX3 Series compatible		Name	Descriptions
CH1	CH2	CH3	CH4	CH1	CH2		
SD8500	SD8510	SD8520	SD8530	SD8402 SD8063	SD8422 SD8438	Serial communication error code	Stores the error code when a serial communication error occurs.
SD8501	SD8511	SD8521	SD8531	SD8403	SD8423	Serial communication error details	When a serial communication error occurs, the error detail is stored.

When an error occurs in Ethernet communication, the error code is stored in the device corresponding to each connection.

Device No.	Name	Descriptions
SD10130	Error code (Connection No.1)	Stores the current error code generated during built-in Ethernet.
SD10131	Error code (Connection No.2)	
SD10132	Error code (Connection No.3)	
SD10133	Error code (Connection No.4)	
SD10134	Error code (Connection No.5)	
SD10135	Error code (Connection No.6)	
SD10136	Error code (Connection No.7)	
SD10137	Error code (Connection No.8)	

The error codes (hexadecimal) and error details stored in devices are shown as follows.

○: Supported, ×: Not supported

Error code	Error details	Description of error	MODBUS communication		Station in which error occurred
			Serial	TCP	
7001H	—	When used by the communication other than MODBUS RTU communication	○	×	Slave
7010H	—	Parity, overrun (rx register) or framing error	○	×	Master/Slave
7304H	—	CRC error	○	×	Master/Slave
7305H	—	Bus character overrun The received data amount is 256 bytes or more	○	○	Master/Slave[*1]
7306H	—	Data length mismatch The number of bytes received does not match with the specified number of bytes	○	○	Master/Slave[*1]
7307H	—	Unsupported function code error	○	○	Slave
7308H	—	Invalid device address	○	○	Slave

Error code	Error details	Description of error	MODBUS communication		Station in which error occurred
			Serial	TCP	
7309H	—	Slave response timeout A slave does not respond within the time set in the time-out time setting of the communication parameter	○	×	Master
730AH	The following "response message formats" are set. Exception function code: High-order byte Exception code: Low-order byte (Refer to ☞ Page 851 When the processing is completed in error at the slave)	Exception response error Slave answers by exception response	○	×	Master
730BH	The following "response message formats" are set. Request station number: High-order byte Response station number: Low-order byte	Slave station No. mismatch The slave station No. of the response does not match the slave station No. of the request	○	×	Master
730CH	The following "response message formats" are set. Request function code: High-order byte Response function code: Low-order byte	Function code mismatch The function code of the response does not match the function code of the request	○	×	Master
730DH	Response function code is stored.	Illegal broadcast command The read command is a broadcast command.	○	×	Slave
730EH	—	Illegal data value in request A value outside the valid range is included in a request message	○	×	Slave
7311H	—	Port No. duplication The port No. to be used and the port No. of the other function are duplicated	×	○	Slave

*1 In MODBUS/TCP communication, an error occurs only in the slave station.

■When the processing is completed in error at the slave

When a master receives an exception response from a slave, the following "response message formats" are set.

b15 to b8	b7 to b0
Exception function code (high-order byte)	Exception code (low-order byte)

The following table outlines the exception function codes (high-order byte).

Exception function code	Function name	Details
01H	Read coils	Completed with an error by read binary (R/W) devices
02H	Read inputs	Completed with an error by read binary (RO) devices
03H	Read holding registers	Completed with an error by read 16 bit (R/W) register
04H	Read input registers	Completed with an error by read 16 bit (RO) register
05H	Write single coil	Completed with an error by write single binary device
06H	Write single register	Completed with an error by write single 16 bit register device
0FH	Write multiple coils	Completed with an error by write multiple binary (R/W) devices
10H	Write multiple registers	Completed with an error by write multiple 16 bit (R/W) registers
16H	Mask Write Register	Completed with an error by manipulate slave register with AND Mask/OR Mask
17H	Read/Write Multiple Registers	Completed with an error by read/write multiple 16 bit (R/W) registers

The following table outlines the exception codes (low-order byte).

Exception code	Exception code name	Details
01H	Illegal function code	Unsupported function code was received
02H	Illegal device address	MODBUS address to which device is not allocated was accessed
03H	Illegal data value	An error occurred in the data area of the request message

Operation error

■Error flags

The operation error flag turns ON when an operation error occurs in the ADPRW instruction.

Verify that the device specified in the table below is ON.

FX5 only				Name	Descriptions
CH1	CH2	CH3	CH4		
SM0				Latest self-diagnostic error (Including the annunciator ON)	Turns ON when an operation error occurs.
SM1				Latest self-diagnostic error (Not including the annunciator ON)	
SM56, SM8067				Operation error	

■Error codes

When the operation error flag turns ON, the error code (hexadecimal) is stored in the operation error code (SD0/SD8067).

The error codes stored in the device are shown as follows.

Error code	Description	Details	Station in which error occurred
1810H	Channel double use	Channel used by the instruction is used by another instruction.	Master/Slave
3600H	Invalid parameter setup	The instruction is used in Slave mode.	Slave
2822H	A device was specified that cannot be specified with the instruction.	A device that cannot be used for the instruction is specified.	Master
3405H	Data was entered outside of the range that can be specified.	The operand device number range or data value set for the instruction is out of range.	Master
2820H	A device or label was beyond the range that can be specified with the instruction.	The operand corresponding device set for the instruction is out of range.	Master

47.4 SLMP

Error codes stored when communication ends in error during SLMP are as provided in the following table.

For the troubleshooting on the SLMP-compatible device side, refer to the SLMP-compatible device manual.

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.	• Check the operation of the external device. • Check whether the connection cable is not disconnected.	○	○
C050H	When the communication data code is set to "ASCII", ASCII code data which cannot be converted to binary is received.	• 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.)	○	○

Error code (hexadecimal)	Error details and causes	Action	SLMP	
			3E	1E
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 the content of text or length of request data in the header, send to CPU module again.	○	—
C059H	• Error in command or subcommand specification. • There is a command or subcommand that cannot be used by the CPU module.	• 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. (Reading or writing by 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.	• 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 reviewing and correcting the 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 returns.)	• Send 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 lock and 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.	• Check the operation of the external device. • Reexamine and change the set values for existence confirmation. • Check whether the connection cable is not disconnected.	○	○
C050H	When the communication data code is set to "ASCII", ASCII code data which cannot be converted to binary is received.	• 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.)	○	○

Error code (hexadecimal)	Error details and causes	Action	SLMP	
			3E	1E
C057H	The request data length in the SLMP message does not match the number of data in the character section (part of the text).	After reviewing and correcting the content of text or length of request data in the header, send to Ethernet module again.	○	○
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 the content of text or length of request data in the header, send to Ethernet module again.	○	—
C059H	• Error in command or subcommand specification. • There is a command or subcommand that cannot be used by the Ethernet module.	• Reconsider request contents. • Send command or subcommand that can be used by the Ethernet module.	○	○
C05BH	Ethernet module cannot read or write data from/to specified device.	Reconsider device to read or write.	○	○
C05CH	Error in request contents. (Error related to device specification, such as 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.	• 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.	• 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 reviewing and correcting the 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 returns.)	• Send 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.	○	—

48 EVENT CODES

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

Event code	Event type	Event category	Event status	Detected event	Detailed information		
					Detailed information 1	Detailed information 2	Detailed information 3
2DA4	System	Error	Moderate	Connection setting parameter (Open system error)	Parameter information[*2]	—	Failure information
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 occupied stations 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]	—	—

Event code	Event type	Event category	Event status	Detected event	Detailed information		
					Detailed information 1	Detailed information 2	Detailed information 3
306C	System	Error	Moderate	The consecutive number of timeouts to disconnection detection is outside the range. (Configuration parameter according to group)	Parameter information[*2]	—	—
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 retries 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]	—	—

Event code	Event type	Event category	Event status	Detected event	Detailed information		
					Detailed information 1	Detailed information 2	Detailed information 3
309F	System	Error	Moderate	The gateway address (IPv6) is outside the range. (Gateway address setting parameter)	Parameter information[*2]	—	—
30A8	System	Error	Moderate	The total number of occupied stations is outside the range. (Configuration setting parameter)	Parameter information[*2]	—	—
30A9	System	Error	Moderate	The master station and remote 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	—	—	—

Event code	Event type	Event category	Event status	Detected event	Detailed information		
					Detailed information 1	Detailed information 2	Detailed information 3
3E56	System	Error	Major	Factory test mode error	—	—	—
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 No.)
- · (Parameter number)
- · Parameter item number

APPENDICES

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

R: Read only, R/W: Read/write

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	• 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	• 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	• 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	• 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

*1 Only FX5S/FX5U/FX5UC CPU module is supported.

Special registers

R: Read only, R/W: Read/write

Device No.	Name	Description	R/W
SD8492	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
SD8493	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
SD8494	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
SD8495	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
SD8496	Default gateway IP address setting (Low-order)	• 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)	• 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
SD10060	Subnet mask (Low-order)	Lower part of the subnet mask setting value.	R

Device No.	Name	Description	R/W
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 on error code, refer to ☞ Page 816 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 "FFFFFFFFFFFFH" 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 "FFFFFFFFFFFFH" 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 FFFEH (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	Continuous unlock failure counts are stored. [SD10320] to [SD10327]: Connection No.1 to No.8	R
SD10338	MELSOFT connection TCP port continuous unlock failure count	Continuous 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 counts are stored.	R
SD10340	Direct connection with MELSOFT continuous unlock failure count	Continuous unlock failure counts of the MELSOFT connection (direct connection) are stored.	R
SD10350, SD10351	Simple CPU communication start request (for communication at request)	Request contact to start data transmission when the communication setting for the simple CPU communication is "When requested" SD10350[b0] to [b15]: Setting No.1 to Setting No.16*1 SD10351[b0] to [b15]: Setting No.17 to Setting No.32*2 0→1: Requested (start request) This device does not automatically turn off at the communication completion, and thus, to start the communications again, the device must be turned off and on.	R/W
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*2 0→1: Requested (stop request) 1→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*2 0→1: Requested (restart request) 1→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*2 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*2 0: Not ready (function not used) 1: Ready	R

Device No.	Name	Description	R/W
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[*2] ■When the communication setting is "Fixed" 0H: Unset 1H: Preparing 3H: Communicating 4H: Stopped 5H: Retry being executed 6H: Monitoring AH: Communications impossible ■When the communication setting is "When requested" 0H: Unset 1H: Preparing 2H: Waiting for request 3H: Communicating 5H: Retry being executed	R
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[*2] 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 and 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[*2] 0: Unset (function not used), communications impossible Other than 0: Execution Interval (unit: ms)	R
SD10476 to SD10507	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 SD10491: Setting No.16 SD10492: Setting No.17 to SD10507: Setting No.32[*2] 0: When there is no error or the communication target is a MELSEC PLC or an 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. [b0]: Initial normal completion status 0: — 1: Initialization normal completion [b1]: Initial abnormal completion status 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 detected 1 to 64: Protocol No. 65535: Not identifiable[*3]	R

Device No.	Name	Description	R/W
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 64) 0: Packet setting or element setting 1: Configuring detailed setting of protocols 65535: Not identifiable[*3]	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: Not identifiable[*3]	R
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 No. 65535: Not identifiable[*3]	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. 0: No registration 1: Registered Protocol numbers 1 to 16: b0 to b15	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. 0: No registration 1: Registered Protocol numbers 17 to 32: b0 to b15	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. 0: No registration 1: Registered Protocol numbers 33 to 48: b0 to b15	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. 0: No registration 1: Registered Protocol numbers 49 to 64: b0 to b15	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: 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

Device No.	Name	Description	R/W
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
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: Cancel request 2: Cancellation completed	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: 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

Device No.	Name	Description	R/W
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
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: Cancel request 2: Cancellation completed	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: 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

Device No.	Name	Description	R/W
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
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: Cancel request 2: Cancellation completed	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: 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

Device No.	Name	Description	R/W
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
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: Cancel request 2: Cancellation completed	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: 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

Device No.	Name	Description	R/W
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
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: Cancel request 2: Cancellation completed	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: 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

Device No.	Name	Description	R/W
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
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: Cancel request 2: Cancellation completed	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: 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

Device No.	Name	Description	R/W
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
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: Cancel request 2: Cancellation completed	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: 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

Device No.	Name	Description	R/W
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
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: Cancel request 2: Cancellation completed	R/W

*1 Only FX5UJ/FX5U/FX5UC CPU module is supported. For the compatible versions of each type, refer to ☞ Page 936 Added and Changed Functions.

*2 Only FX5U/FX5UC CPU module is supported. For the compatible versions of each type, refer to ☞ Page 936 Added and Changed Functions.

*3 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 details on the listed buffer memories, as well as the buffer memories not described below, refer to following manuals.

📖 MELSEC iQ-F FX5 Ethernet Module User's Manual

📖 MELSEC iQ-F FX5 EtherNet/IP Module User's Manual

📖 MELSEC iQ-F FX5 BACnet Reference Manual

R: Read only, R/W: Read/write

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 details on error code, refer to ☞ Page 816 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 pattern setting (Low-order)	• Stores subnet mask pattern (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 pattern setting (High-order)	• Stores subnet mask pattern (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: Do 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: Disabled 1: Enabled	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: IP address setting or other (Un\G50 to Un\G55) value exceeds the setting range	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)	This 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 816 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 816 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
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

Buffer memory No.		Name	Description	R/W
Decimal	Hexadecimal			
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: Stopped 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 details on error code, refer to ☞ Page 816 ERROR CODES.	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
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

Appendix 3 Port Numbers Used by Ethernet-equipped Modules

The following port numbers are used by the system and cannot be specified.

This applies to FX5U/FX5UC CPU modules.

Port No.		Applications	Protocol	Default port status
Decimal	Hexadecimal			
20	14H	For FTP server function (data port)	TCP/IP	Close
21	15H	For FTP server function (control port)	TCP/IP	Close
80	50H	For Web server function[*1]	TCP/IP	Close
123	7BH	For clock setting function (SNTP client)	UDP/IP	Close
502	1F6H	For MODBUS/TCP function[*1]	TCP/IP	Close
5560	15B8H	For MELSOFT direct connection	UDP/IP	Open[*1]
5561	15B9H	For searching modules on the network	UDP/IP	Open[*1]
5562	15BAH	MELSOFT transmission port (TCP/IP)	TCP/IP	Open[*1]
5565	15BDH	MELSOFT transmission port (UDP/IP)	UDP/IP	Open[*1]
61440	F000H	For sending data for device supporting iQ Sensor Solution	UDP/IP	Open[*1]
61441	F001H	For SLMPSND instruction	UDP/IP	Open[*1]
61450	F00AH	For CC-Link IE Field Network Basic function	UDP/IP	Close
61696 to 61999	F100H to F22FH	For simple CPU communication function	UDP/IP	Close
62000 to 65534	F230H to FFFEH	For file transfer function (FTP client)	TCP/IP	Close

*1 This can be changed with the module parameters.

Appendix 4 Combined Use of Serial Communication

Channel specification

The same serial port cannot be used for more than one communication function.

Duplicate channel numbers cannot be specified in communication settings. If the channel number specified by the following instruction and the channel number of another communication function are specified twice, an error may occur.

Inverter communication	Non-protocol communication	Predefined protocol support function
IVCK, IVDR, IVRD, IVWR, IVBWR, IVMC instruction	RS2 instruction	S(P).CPRTCL instruction

If a communications error occurs, check if commands above are using the same channel.

If so, first either delete unnecessary command(s) or correct the channel number, then reset the system by turning the power OFF→ON.

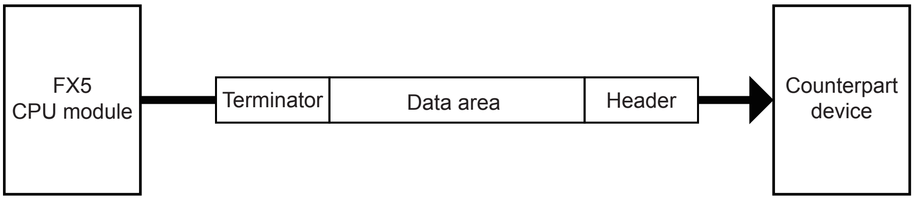
Appendix 5 Operation Image and Data Structure of Predefined Protocol

Operation image of each communication type

Operation image of each communication type of protocol as shown below.

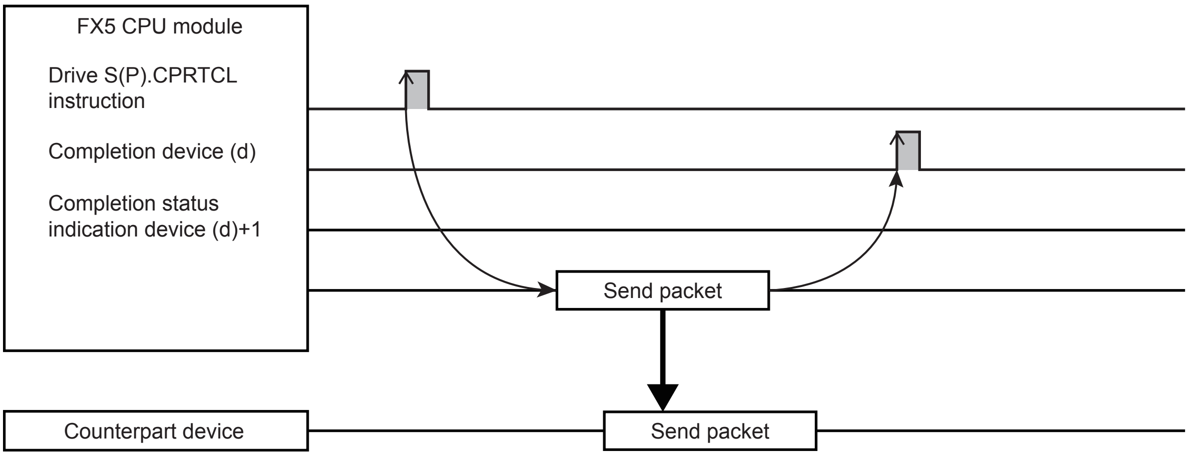
Send only

The specified packet is transmitted once.

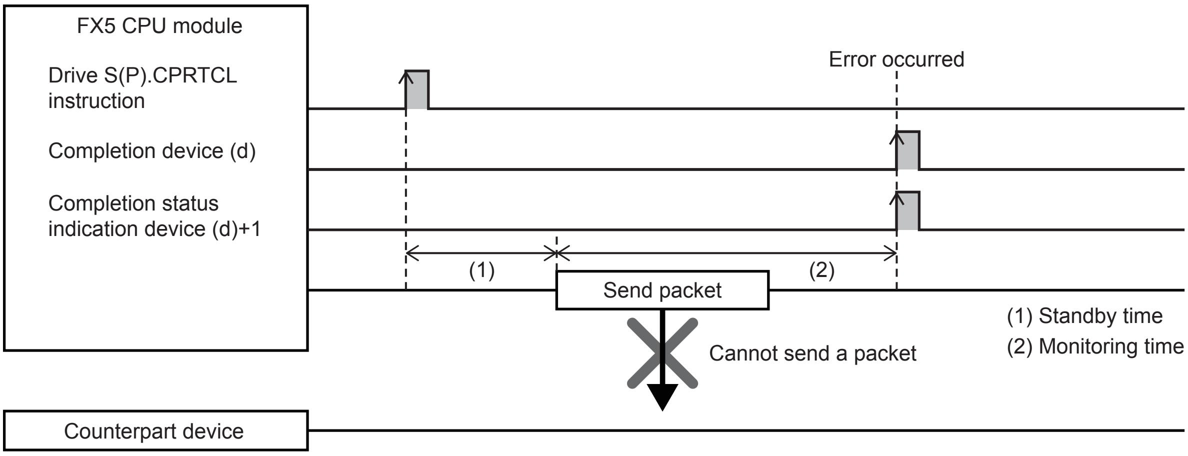


The operation is as follows.

■Normal completion



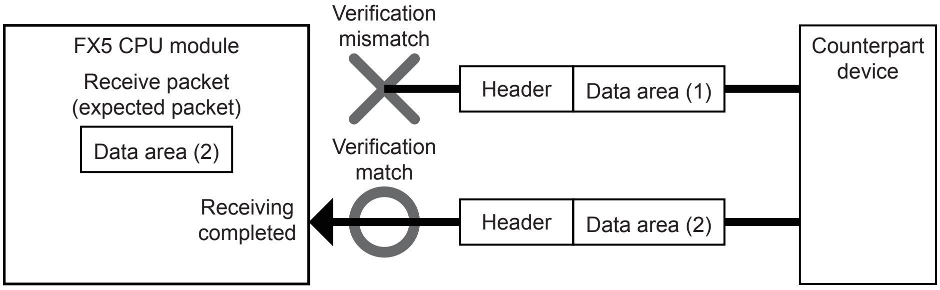
■Error completion (transmission monitoring timeout error)



Receive only

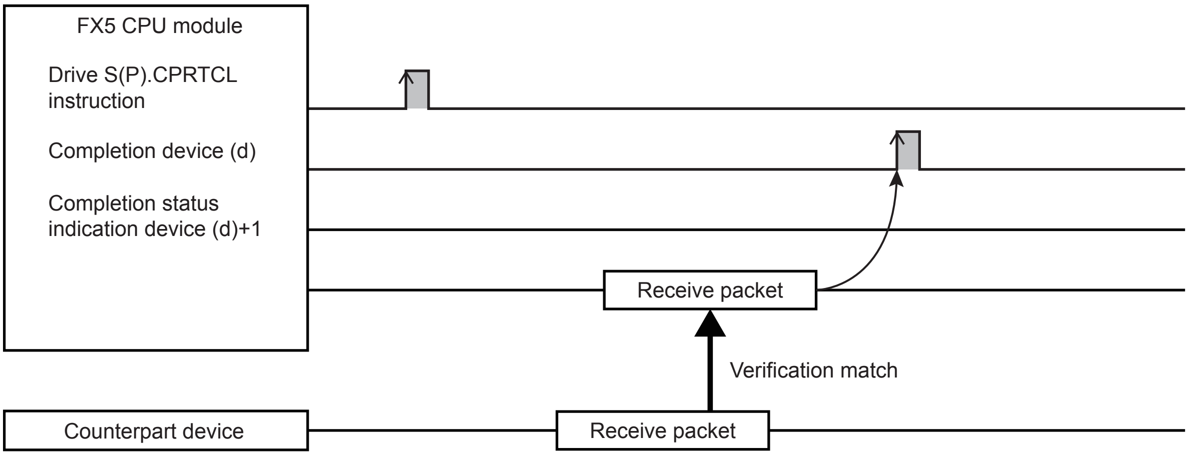
When data are received from the counterpart device, the process completes when the received data matches the receive packet and the receiving process is performed. For verification mismatch, the receive data is discarded, and the CPU unit waits for the next receive data. (☞ Page 882 Verification operation)

Settable receive packet (expected packet) is up to 16.

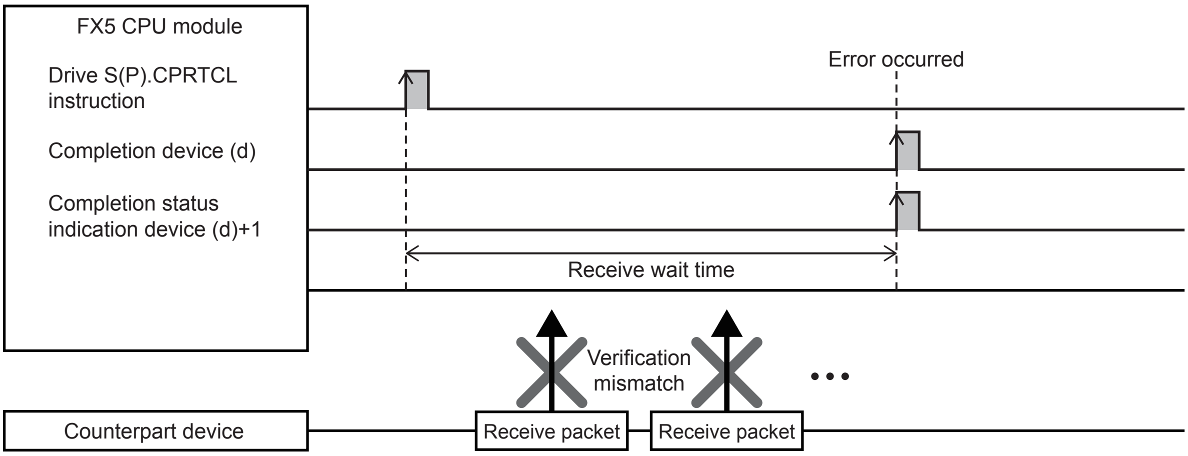


The operation is as follows.

■Normal completion



■Error completion (receive wait timeout error)



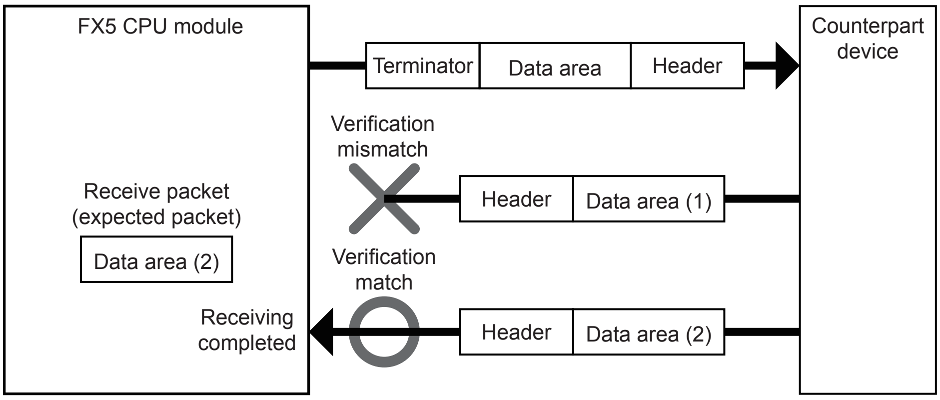
Point

- When variables are included in receive packet elements, variable parts are not verified.
- When more than one receive packet is specified, received data is verified with the receive packet information of the first registered packet in the order of registration. The receive processing is performed once received data match one of the receive packet number, and further verification is not performed.
- The receive packet number which is matched as the result of the verification is stored in the control data of the S(P).CPRTCL instruction.

Send & receive

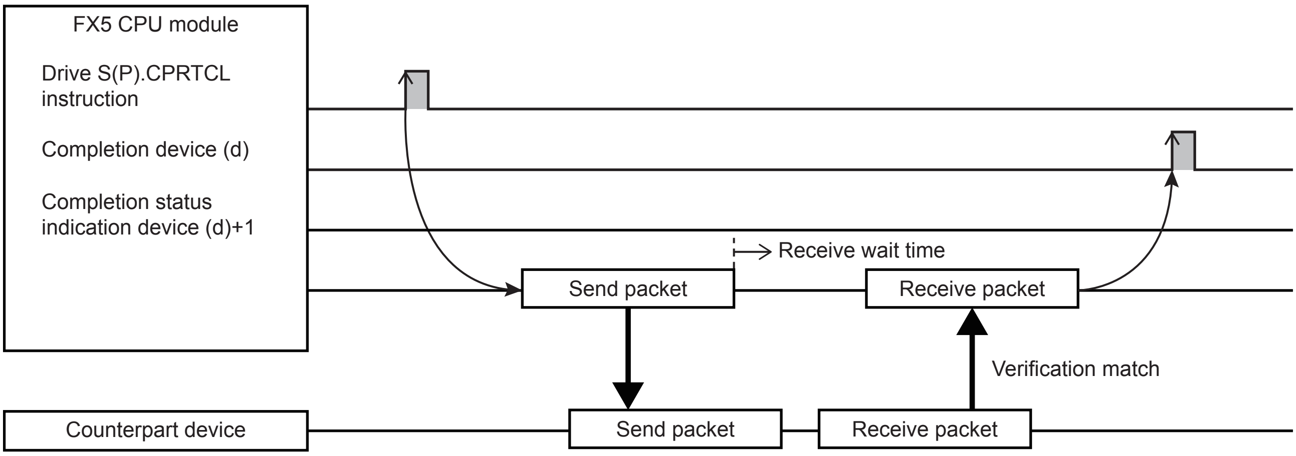
A packet is sent once, and the status changes to the data receive wait status after the transmission completes normally. Then, data is received from the counterpart device, and the process completes when the received data matches the receive packet and the receiving process is performed. (☞ Page 882 Verification operation)

Settable receive packet (expected packet) is up to 16.

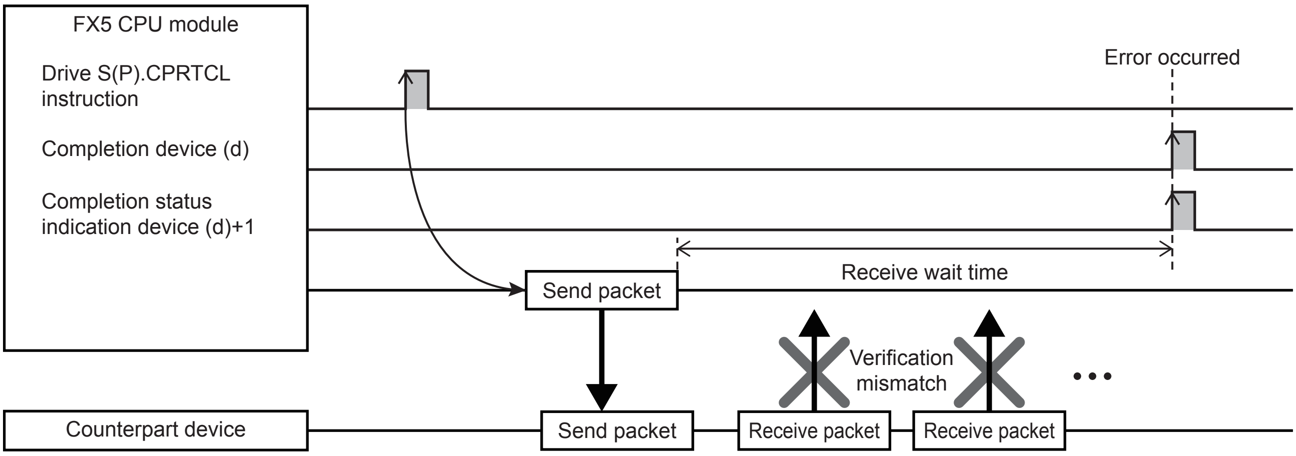


The operation is as follows.

■Normal completion



■Error completion (transmission monitoring timeout error)



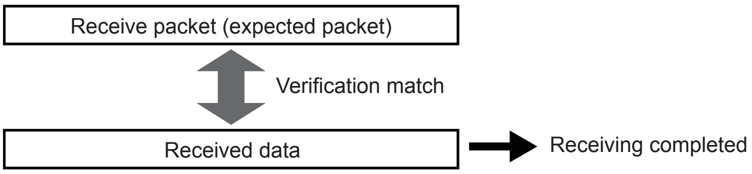
Point

- When variables are included in receive packet elements, variable parts are not verified.
- When more than one receive packet is specified, received data is verified with the receive packet information of the first registered packet in the order of registration. The receive processing is performed once received data match one of the receive packet number, and further verification is not performed.
- The receive packet number which is matched as the result of the verification is stored in the control data of the S(P).CPRTCL instruction.

Verification operation

The following shows the verification operations of receive packets (expected packets) of when receive is included in the selected communication type.

Receive data and receive packets are compared in the receive processing, and the processing is completed when the verification result is "matched". Note that the data section of a packet is handled as receive data in the same way as it is handled at send operation.



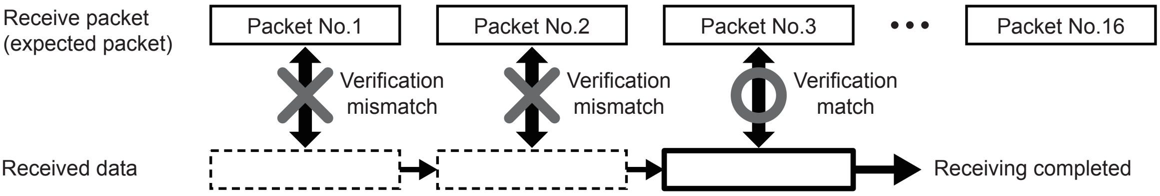
Verification standards

In the verification of receive packets, each element is judged if it is "matched" or not. The standard to judge each element as "matched" depends on each element type. The following table lists verification standards for each element type.

Element type	Verification standards
Header	Judged as "matched" when the data contents are the same.
Terminator	
Static data	
Length	Judged as "matched" when the number of digits is the same.
Non-conversion variable	
Conversion variable	
Check code	Judged as "matched" when the result calculated based on the settings (horizontal parity, sum check, CRC-16) is the same.
Non-verified reception	Judged as "matched" when the number of digits is the same.

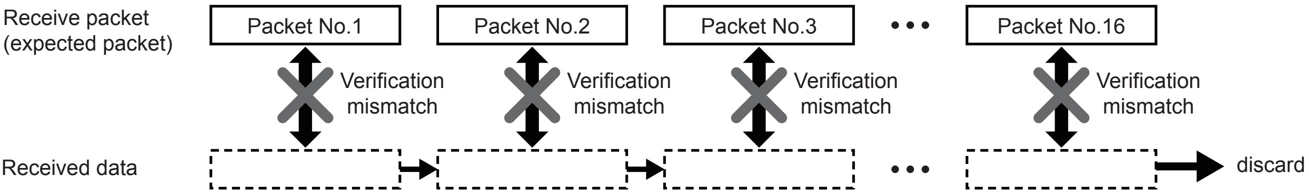
Operation performed when multiple receive packets are registered

When data is received, the verification operation is performed in order starting from the receive packet that was first registered. The receive processing is completed when the receive packets and receive data match.



Operation performed for verification mismatch

When receive data does not match with all receive packets, the receive data is discarded. Note that discarding receive data does not cause an error.



Data examples of Length

Shows the data examples of element length that can be placed in a packet.

Data example

The following is an example in the case where the calculated value of length is 258 bytes in decimal (hexadecimal: 102H).

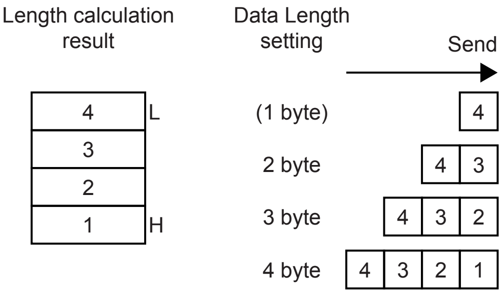
■Data flow is Forward Direction

Code type	**Data length**[*1]			
	1 byte	**2 byte**	**3 byte**	**4 byte**
ASCII hexadecimal	2 (32H)	02 (30H 32H)	102 (31H 30H 32H)	0102 (30H 31H 30H 32H)
ASCII decimal	8 (38H)	58 (35H 38H)	258 (32H 35H 38H)	0258 (30H 32H 35H 38H)
HEX	02H	0102H	000102H	00000102H

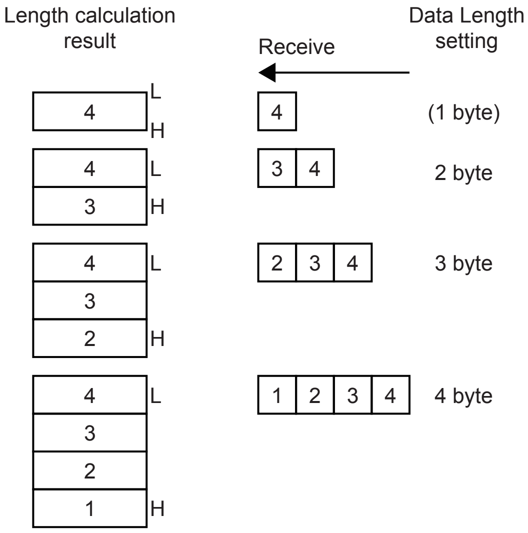
*1 Values in ( ) indicate ASCII code.

The send/receive image is as follows.

■When sending



■When receiving



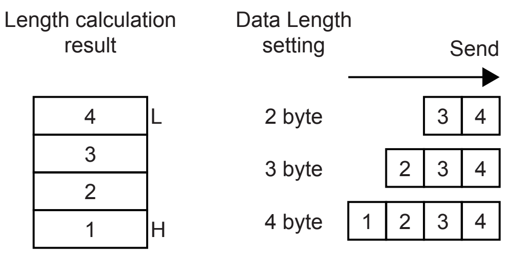
■Data flow is Reverse Direction

Code type	**Data length**[*1]			
	1 byte	**2 byte**	**3 byte**	**4 byte**
ASCII hexadecimal	—	20 (32H 30H)	201 (32H 30H 31H)	2010 (32H 30H 31H 30H)
ASCII decimal	—	85 (38H 35H)	852 (38H 35H 32H)	8520 (38H 35H 32H 30H)
HEX	—	0201H	020100H	02010000H

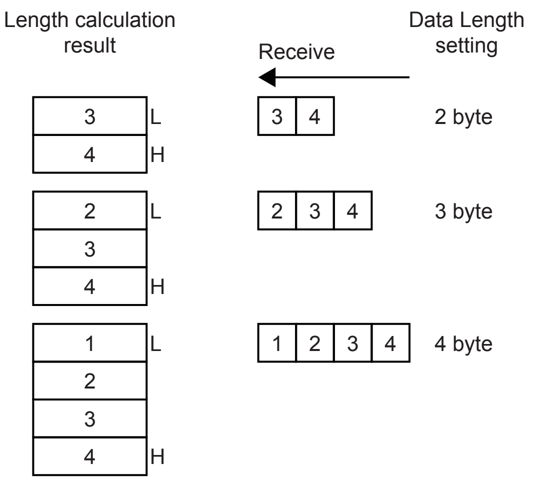
*1 Values in ( ) indicate ASCII code.

The send/receive image is as follows.

■When sending



■When receiving



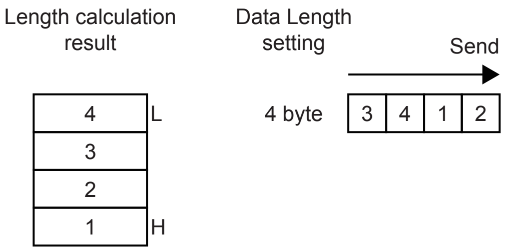
■Data flow is Byte Swap

Code type	Data length[*1]			
	1 byte	2 byte	3 byte	4 byte
ASCII hexadecimal	—	—	—	1020 (31H 30H 32H 30H)
ASCII decimal	—	—	—	2085 (32H 30H 38H 35H)
HEX	—	—	—	00000201H

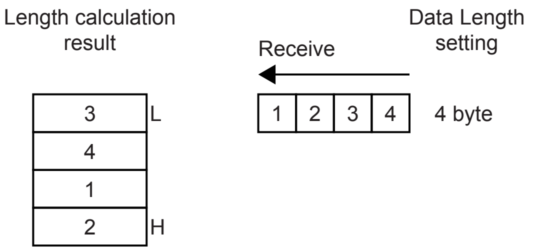
[*1] Values in ( ) indicate ASCII code.

The send/receive image is as follows.

■When sending

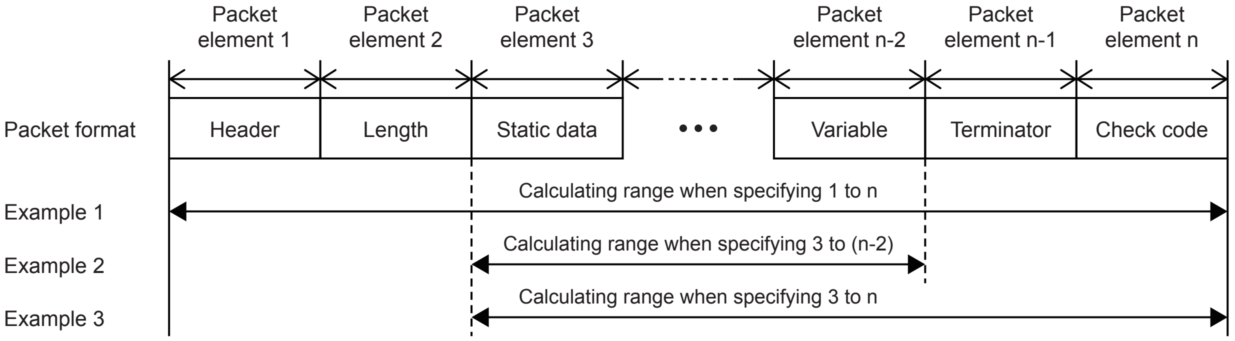


■When receiving



Calculating range

The following shows specification examples of the calculating range of Length.



Example 1: Calculating range when 1 and n are specified as the start and the end respectively

Example 2: Calculating range when 3 and n-2 are specified as the start and the end respectively

Example 3: Calculating range when 3 and n are specified as the start and the end respectively

Data examples of Non-conversion Variable

Shows the data examples of element non-conversion variable that can be placed in a packet.

Data example 1

The following table shows data to be stored in the data storage area when the string of send data is 'ABCD' (ASCII code: A = 41H, B = 42H, C = 43H, and D = 44H).

Item	Description			
Fixed Length/Variable Length	Fixed Length			
Data Length/Maximum Data Length	4 bytes			
Unit of Stored Data	Lower Byte + Upper Byte		Lower Bytes Only	
Byte Swap	Disable	Enable	Disable	Enable
Data Storage Area Specification	D0			
Data to be stored in data storage area	D0=4241H D1=4443H	D0=4142H D1=4344H	D0=0041H D1=0042H D2=0043H D3=0044H	D0=0042H D1=0041H D2=0044H D3=0043H

Data example 2

The following table shows data to be stored in the data storage area when the string of send data is 'EFG' (ASCII code: E = 45H, F = 46H, and G = 47H).

Item	Description			
Fixed Length/Variable Length	Fixed Length			
Data Length/Maximum Data Length	3 bytes			
Unit of Stored Data	Lower Byte + Upper Byte		Lower Bytes Only	
Byte Swap	Disable	Enable	Disable	Enable
Data Storage Area Specification	D0			
Data to be stored in data storage area	D0=4645H D1=0047H	D0=4546H D1=4700H	D0=0045H D1=0046H D2=0047H D3= (any data)	D0=0046H D1=0045H D2=0047H D3= (any data)

Data examples of Conversion Variable

Shows the data examples of element conversion variable that can be placed in a packet.

Data example

The following table shows send data when a packet consists of [Header], [Conversion variable], [Terminator] and data stored in the data storage area is D0=837 (0345H), D1=18 (0012H). (Reference: decimal: 120345H = 1180485)

■Data example 1

Item	Description		
Conversion	HEX→ASCII Decimal	HEX→ASCII Decimal	HEX→ASCII Decimal
Fixed Number of Data/Variable Number of Data	Fixed Number of Data	Fixed Number of Data	Fixed Number of Data
Number of Send Data	1	1	1
Number of Send Digits of Data	5	5	Variable Number of Digits
Blank-padded Character at Send	0	Space	— (Not applicable)
Conversion Unit	Word	Word	Word
Sign	Unsigned	Signed	Signed
Sign Character	— (Not applicable)	+	+
Number of Decimals	No Decimal Point	2	No Decimal Point
Delimiter	No Delimiter	Comma	Comma
Data Storage Area Specification	D0	D0	D0
Send data[*1]	[Header]00837[Terminator]	[Header]+_ _8.37,[Terminator]	[Header]+837,[Terminator]

*1 The "_" indicates a space.

■Data example 2

Item	Description		
Conversion	HEX→ASCII Decimal	HEX→ASCII Decimal	HEX→ASCII Decimal
Fixed Number of Data/Variable Number of Data	Fixed Number of Data	Fixed Number of Data	Fixed Number of Data
Number of Send Data	1	2	2
Number of Send Digits of Data	10	5	5
Blank-padded Character at Send	0	Space	0
Conversion Unit	Double word	Word	Word
Sign	Signed	Unsigned	Signed
Sign Character	+	— (Not applicable)	+
Number of Decimals	8	No Decimal Point	2
Delimiter	No Delimiter	No Delimiter	Comma
Data Storage Area Specification	D0	D0	D0
Send data[*1]	[Header]+00.01180485[Terminator]	[Header]_ _ 837 _ _ _18[Terminator]	[Header]+008.37,+000.18[Terminator]

*1 The "_" indicates a space.

■Data example 3

Item	Description		
Conversion	HEX→ASCII Hexadecimal	HEX→ASCII Hexadecimal	HEX→ASCII Hexadecimal
Fixed Number of Data/Variable Number of Data	Fixed Number of Data	Fixed Number of Data	Fixed Number of Data
Number of Send Data	1	2	2
Number of Send Digits of Data	Variable Number of Digits	4	Variable Number of Digits
Blank-padded Character at Send	— (Not applicable)	Space	— (Not applicable)
Conversion Unit	Word	Word	Word
Sign	— (Not applicable)	— (Not applicable)	— (Not applicable)
Sign Character	— (Not applicable)	— (Not applicable)	— (Not applicable)
Number of Decimals	— (Not applicable)	— (Not applicable)	— (Not applicable)
Delimiter	Comma	Comma	Comma
Data Storage Area Specification	D0	D0	D0
Send data[*1]	[Header]345,[Terminator]	[Header]_ 345,_ _ 12[Terminator]	[Header]345,12[Terminator]

*1 The "_" indicates a space.

Operation of Sign Character

■Sending

Ex.

Conversion: HEX→ASCII Decimal Number of Send Data: 1 Conversion Unit: Word Number of Send Digits of Data: 5

- Unsigned

- Signed (Sign Character: +)

- Signed (Negative value)

■Receiving

Ex.

Conversion: ASCII Decimal→HEX Number of Receive Data: 1 Conversion Unit: Word Number of Receive Digits of Data: 5

- Unsigned

- Signed (Sign Character: +)

- Signed (Negative value)

Operation of Number of Decimals

■Sending

- Fixed point

Ex.

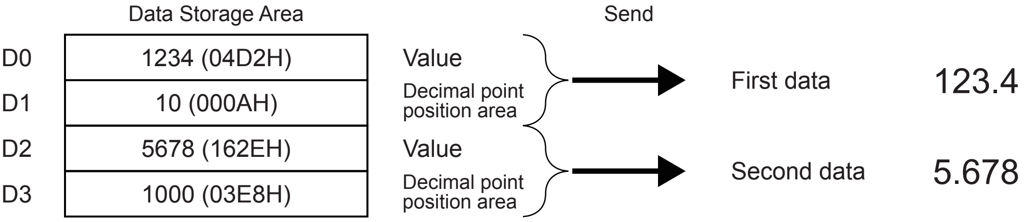
Conversion: HEX→ASCII Decimal Number of Send Data: 2 Conversion Unit: Word Number of Send Digits of Data: 4 Number of Decimals: 1

D0	1234 (04D2H)
D1	5678 (162EH)

- Variable point

Ex.

Conversion: HEX→ASCII Decimal Number of Send Data: 2 Conversion Unit: Word Number of Send Digits of Data: 4



D0	1234 (04D2H)
D1	10 (000AH)
D2	5678 (162EH)
D3	1000 (03E8H)

Precautions

An error occurs when a decimal point position is larger than the number of digits for variable point.

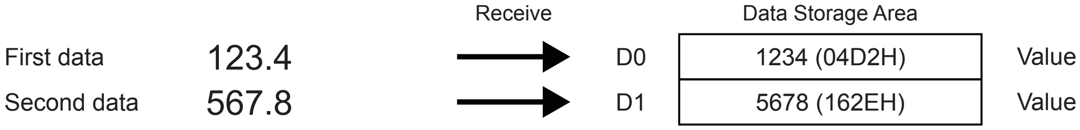
(Example) When the number of digits is 3 and the decimal point position is 1000 or larger, an error occurs.

■Receiving

- Fixed point

Ex.

Conversion: ASCII Decimal→HEX Number of Receive Data: 2 Conversion Unit: Word Number of Receive Digits of Data: 4 Number of Decimals: 1

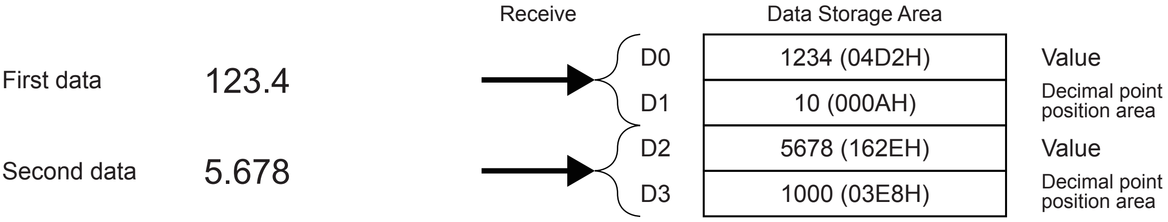


D0	1234 (04D2H)
D1	5678 (162EH)

- Variable point

Ex.

Conversion: ASCII Decimal→HEX Number of Receive Data: 2 Conversion Unit: Word Number of Receive Digits of Data: 4



D0	1234 (04D2H)
D1	10 (000AH)
D2	5678 (162EH)
D3	1000 (03E8H)

Operation of Delimiter

■Sending

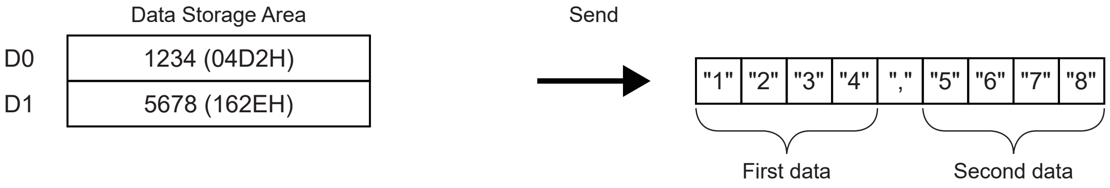
- Comma (Space too)

Ex.

Conversion: HEX→ASCII Decimal Number of Send Data: 1 Conversion Unit: Word Number of Send Digits of Data: 4

Ex.

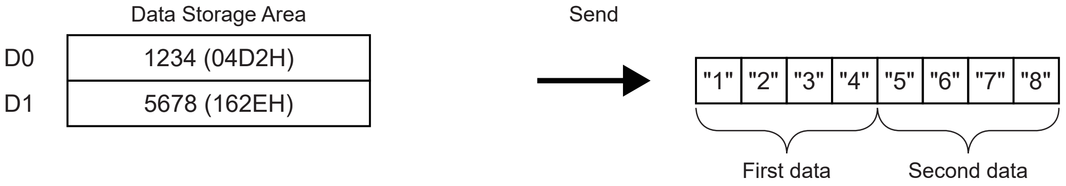
Conversion: HEX→ASCII Decimal Number of Send Data: 2 Conversion Unit: Word Number of Send Digits of Data: 4



- No Delimiter

Ex.

Conversion: HEX→ASCII Decimal Number of Send Data: 2 Conversion Unit: Word Number of Send Digits of Data: 4



■Receiving

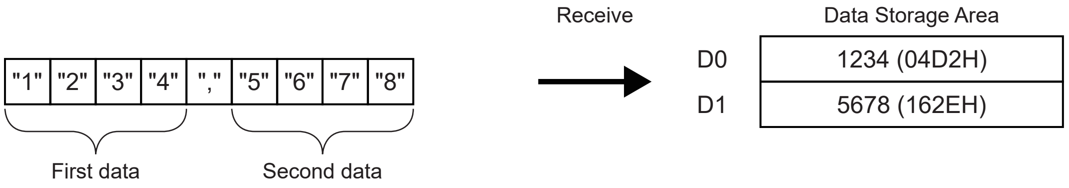
- Comma (Space too)

Ex.

Conversion: ASCII Decimal→HEX Number of Receive Data: 1 Conversion Unit: Word Number of Receive Digits of Data: 4

Ex.

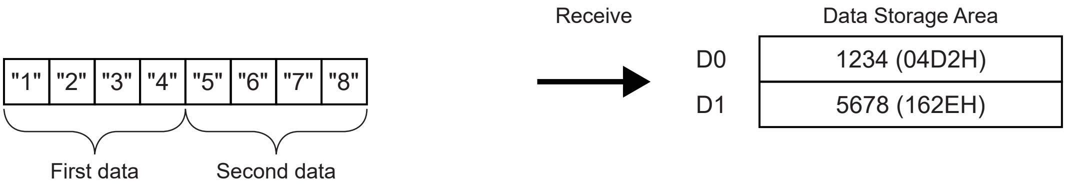
Conversion: ASCII Decimal→HEX Number of Receive Data: 2 Conversion Unit: Word Number of Receive Digits of Data: 4



- No Delimiter

Ex.

Conversion: ASCII Decimal→HEX Number of Receive Data: 2 Conversion Unit: Word Number of Receive Digits of Data: 4

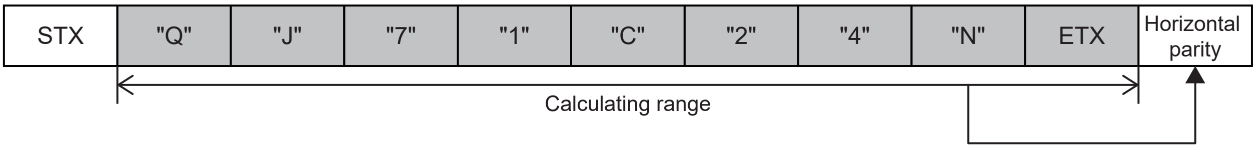


Data examples of Check Code

Shows the data examples of element check code that can be placed in a packet.

Calculation procedure for horizontal parity

The following show procedures for calculating horizontal parities using the following sample data.



For the packet shown above

"Q" (51H) 0101 0001
XOR
"J" (4AH) 0100 1010 = 0001 1011
XOR
"7" (37H) 0011 0111 = 0010 1100
XOR
"1" (31H) 0011 0001 = 0001 1101
XOR
"C" (43H) 0100 0011 = 0101 1110
XOR
"2" (32H) 0011 0010 = 0110 1100
XOR
"4" (34H) 0011 0100 = 0101 1000
XOR
"N" (4EH) 0100 1110 = 0001 0110
XOR
ETX (03H) 0000 0011 = 0001 0101 (Binary)
1 5 (Hexadecimal)

■Data Flow: Forward Direction (Data Length: 1)

- No complement calculation is designated

(Reference) Hexadecimal: 15H, Decimal: 21

Code type	Data length (Values in ( ) indicate ASCII code.)			
	1 byte	2 byte	3 byte	4 byte
ASCII hexadecimal	5 (35H)	15 (31H 35H)	015 (30H 31H 35H)	0015 (30H 30H 31H 35H)
ASCII decimal	1 (31H)	21 (32H 31H)	021 (30H 32H 31H)	0021 (30H 30H 32H 31H)
HEX	15H	0015H	000015H	00000015H

- One's complement is designated (One's complement for 0000 0015H: FFFF FFEAH).

When Code Type is ASCII Decimal, the lower one word is extracted and converted from hexadecimal to decimal.

(Hexadecimal: FFEAH, Decimal: 65514)

Code type	Data length (Values in ( ) indicate ASCII code.)			
	1 byte	2 byte	3 byte	4 byte
ASCII hexadecimal	A (41H)	EA (45H 41H)	FEA (46H 45H 41H)	FFEA (46H 46H 45H 41H)
ASCII decimal	4 (34H)	14 (31H 34H)	514 (35H 31H 34H)	5514 (35H 35H 31H 34H)
HEX	EAH	FFEAH	FFFFEAH	FFFFFFEAH

- Two's complement is designated (Two's complement for 0000 0015H: FFFF FFEBH).

When Code Type is ASCII Decimal, the lower one word is extracted and converted from hexadecimal to decimal. (Hexadecimal: FFFBH, Decimal: 65515)

Code type	Data length (Values in ( ) indicate ASCII code.)			
	1 byte	2 byte	3 byte	4 byte
ASCII hexadecimal	B (42H)	EB (45H 42H)	FEB (46H 45H 42H)	FFEB (46H 46H 45H 42H)
ASCII decimal	5 (35H)	15 (31H 35H)	515 (35H 31H 35H)	5515 (35H 35H 31H 35H)
HEX	EBH	FFEBH	FFFFEBH	FFFFFFEBH

■Data Flow: Reverse Direction

- No complement calculation is designated

(Reference) Hexadecimal: 15H, Decimal: 21

Code type	Data length (Values in ( ) indicate ASCII code.)			
	1 byte	2 byte	3 byte	4 byte
ASCII hexadecimal	—	51 (35H 31H)	510 (35H 31H 30H)	5100 (35H 31H 30H 30H)
ASCII decimal	—	12 (31H 32H)	120 (31H 32H 30H)	1200 (31H 32H 30H 30H)
HEX	—	1500H	150000H	15000000H

- One's complement is designated (One's complement for 0000 0015H: FFFF FFEAH).

When Code Type is ASCII Decimal, the lower one word is extracted and converted from hexadecimal to decimal. (Hexadecimal: FFEAH, Decimal: 65514)

Code type	Data length (Values in ( ) indicate ASCII code.)			
	1 byte	2 byte	3 byte	4 byte
ASCII hexadecimal	—	AE (41H 45H)	AEF (41H 45H 46H)	AEFF (41H 45H 46H 46H)
ASCII decimal	—	41 (34H 31H)	415 (34H 31H 35H)	4155 (34H 31H 35H 35H)
HEX	—	EAFFH	EAFFFFH	EAFFFFFFH

- Two's complement is designated (Two's complement for 0000 0015H: FFFF FFEBH).

When Code Type is ASCII Decimal, the lower one word is extracted and converted from hexadecimal to decimal. (Hexadecimal: FFFBH, Decimal: 65515)

Code type	Data length (Values in ( ) indicate ASCII code.)			
	1 byte	2 byte	3 byte	4 byte
ASCII hexadecimal	—	BE (42H 45H)	BEF (42H 45H 46H)	BEFF (42H 45H 46H 46H)
ASCII decimal	—	51 (35H 31H)	515 (35H 31H 35H)	5155 (35H 31H 35H 35H)
HEX	—	EBFFH	EBFFFFH	EBFFFFFFH

■Data Flow: Byte Swap

- No complement calculation is designated

(Reference) Hexadecimal: 15H, Decimal: 21

Code type	Data length (Values in ( ) indicate ASCII code.)			
	1 byte	2 byte	3 byte	4 byte
ASCII hexadecimal	—	—	—	0051 (30H 30H 35H 31H)
ASCII decimal	—	—	—	0012 (30H 30H 31H 32H)
HEX	—	—	—	00001500H

- One's complement is designated (One's complement for 0000 0015H: FFFF FFEAH).

When Code Type is ASCII Decimal, the lower one word is extracted and converted from hexadecimal to decimal. (Hexadecimal: FFEAH, Decimal: 65514)

Code type	Data length (Values in ( ) indicate ASCII code.)			
	1 byte	2 byte	3 byte	4 byte
ASCII hexadecimal	—	—	—	FFAE (46H 46H 41H 45H)
ASCII decimal	—	—	—	5541 (35H 35H 34H 31H)
HEX	—	—	—	FFFFEAFFH

- Two's complement is designated (Two's complement for 0000 0015H: FFFF FFEBH).

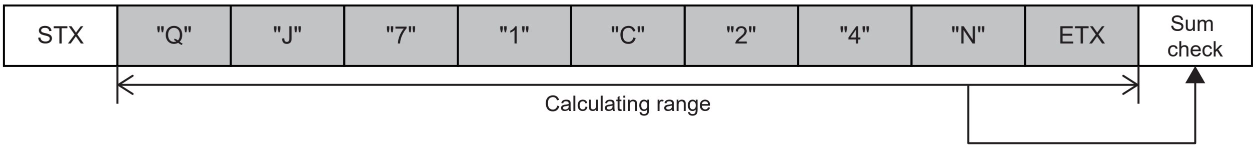
When Code Type is ASCII Decimal, the lower one word is extracted and converted from hexadecimal to decimal.

(Hexadecimal: FFFBH, Decimal: 65515)

Code type	Data length (Values in ( ) indicate ASCII code.)			
	1 byte	2 byte	3 byte	4 byte
ASCII hexadecimal	—	—	—	FFBE (46H 46H 42H 45H)
ASCII decimal	—	—	—	5551 (35H 35H 35H 31H)
HEX	—	—	—	FFFFEBFFH

Calculation procedure for sum check

The following show procedures for calculating sum check codes using the following sample data.



For the data shown above

Sum check value = 51H + 4AH + 37H + 31H + 43H + 32H + 34H + 4EH + 03H = 1FDH

■Data Flow: Forward Direction (Data Length: 1)

- No complement calculation is designated

(Reference) Hexadecimal: 1FDH, Decimal: 509

Code type	Data length (Values in ( ) indicate ASCII code.)			
	1 byte	2 byte	3 byte	4 byte
ASCII hexadecimal	D (44H)	FD (46H 44H)	1FD (31H 46H 44H)	01FD (30H 31H 46H 44H)
ASCII decimal	9 (39H)	09 (30H 39H)	509 (35H 30H 39H)	0509 (30H 35H 30H 39H)
HEX	FDH	01FDH	0001FDH	000001FDH

- One's complement is designated (One's complement for 0000 01FDH: FFFF FE02H).

When Code Type is ASCII Decimal, the lower one word is extracted and converted from hexadecimal to decimal.

(Hexadecimal: FE02H, Decimal: 65026)

Code type	Data length (Values in ( ) indicate ASCII code.)			
	1 byte	2 byte	3 byte	4 byte
ASCII hexadecimal	2 (32H)	02 (30H 32H)	E02 (45H 30H 32H)	FE02 (46H 45H 30H 32H)
ASCII decimal	6 (36H)	26 (32H 36H)	026 (30H 32H 36H)	5026 (35H 30H 32H 36H)
HEX	02H	FE02H	FFFE02H	FFFFFE02H

- Two's complement is designated (Two's complement for 0000 01FDH: FFFF FE03H).

When Code Type is ASCII Decimal, the lower one word is extracted and converted from hexadecimal to decimal.

(Hexadecimal: FE03H, Decimal: 65027)

Code type	Data length (Values in ( ) indicate ASCII code.)			
	1 byte	2 byte	3 byte	4 byte
ASCII hexadecimal	3 (33H)	03 (30H 33H)	E03 (45H 30H 33H)	FE03 (46H 45H 30H 33H)
ASCII decimal	7 (37H)	27 (32H 37H)	027 (30H 32H 37H)	5027 (35H 30H 32H 37H)
HEX	03H	FE03H	FFFE03H	FFFFFE03H

■Data Flow: Reverse Direction

- No complement calculation is designated

(Reference) Hexadecimal: 1FDH, Decimal: 509

Code type	Data length (Values in ( ) indicate ASCII code.)			
	1 byte	2 byte	3 byte	4 byte
ASCII hexadecimal	—	DF (44H 46H)	DF1 (44H 46H 31H)	DF10 (44H 46H 31H 30H)
ASCII decimal	—	90 (39H 30H)	905 (39H 30H 35H)	9050 (39H30H 35H 30H)
HEX	—	FD01H	FD0100H	FD010000H

- One's complement is designated (One's complement for 0000 01FDH: FFFF FE02H).

When Code Type is ASCII Decimal, the lower one word is extracted and converted from hexadecimal to decimal.
(Hexadecimal: FE02H, Decimal: 65026)

Code type	Data length (Values in ( ) indicate ASCII code.)			
	1 byte	2 byte	3 byte	4 byte
ASCII hexadecimal	—	20 (32H 30H)	20E (32H 30H 45H)	20EF (32H 30H 45H 46H)
ASCII decimal	—	62 (36H 32H)	620 (36H 32H 30H)	6205 (36H 32H 30H 35H)
HEX	—	02FEH	02FEFFH	02FEFFFFH

- Two's complement is designated (Two's complement for 0000 01FDH: FFFF FE03H).

When Code Type is ASCII Decimal, the lower one word is extracted and converted from hexadecimal to decimal.
(Hexadecimal: FE03H, Decimal: 65027)

Code type	Data length (Values in ( ) indicate ASCII code.)			
	1 byte	2 byte	3 byte	4 byte
ASCII hexadecimal	—	30 (30H 33H)	30E (33H 30H 45H)	30EF (33H 30H 45H 46H)
ASCII decimal	—	72 (37H 32H)	720 (37H 32H 30H)	7205 (37H 32H 30H 35H)
HEX	—	03FEHH	03FEFFH	03FEFFFFH

■Data Flow: Byte Swap

- No complement calculation is designated

(Reference) Hexadecimal: 1FDH, Decimal: 509

Code type	Data length (Values in ( ) indicate ASCII code.)			
	1 byte	2 byte	3 byte	4 byte
ASCII hexadecimal	—	—	—	10DF (31H 30H 44H 46H)
ASCII decimal	—	—	—	5090 (35H 30H 39H 30H)
HEX	—	—	—	0000FD01H

- One's complement is designated (One's complement for 0000 01FDH: FFFF FE02H).

When Code Type is ASCII Decimal, the lower one word is extracted and converted from hexadecimal to decimal.
(Hexadecimal: FE02H, Decimal: 65026)

Code type	Data length (Values in ( ) indicate ASCII code.)			
	1 byte	2 byte	3 byte	4 byte
ASCII hexadecimal	—	—	—	EF20 (45H 46H 32H 30H)
ASCII decimal	—	—	—	0562 (30H 35H 36H 32H)
HEX	—	—	—	FFFF02FEH

- Two's complement is designated (Two's complement for 0000 01FDH: FFFF FE03H).

When Code Type is ASCII Decimal, the lower one word is extracted and converted from hexadecimal to decimal.
(Hexadecimal: FE03H, Decimal: 65027)

Code type	**Data length (Values in ( ) indicate ASCII code.)**			
	1 byte	**2 byte**	**3 byte**	**4 byte**
ASCII hexadecimal	—	—	—	EF30 (45H 46H 33H 30H)
ASCII decimal	—	—	—	0572 (30H 35H 37H 32H)
HEX	—	—	—	FFFF03FEH

Calculation procedure for 16-bit CRC (for MODBUS)

This is an error check method to be used when data are transmitted/received with the RTU mode of the MODBUS protocol. The data length of CRC is fixed to 2 bytes (16 bits), and the CRC is calculated every 1 byte (8 bits) from the start of the calculating range according to the following procedure.

1. Load a 16-bit register whose bits are all '1'.
2. Exclusive OR (XOR) the first 1 byte (8 bits) of the calculating range with 8 bits in above 1.
3. Shift the result of step 2 for one bit right.
4. If the least significant bit in above 2 is '1', exclusive OR (XOR) the result of 3 with the generator polynomial (A001H). If the least significant bit is '0', shift the result of step 3 one bit right (operation described in 3) without the exclusive OR (XOR) operation.
5. Repeat steps 3 and 4 for 8 times.
6. Exclusive OR (XOR) the result of the above 5 with the next 1 byte (8 bits).
7. Repeat step 3 through 6 until all bytes have been processed. The final result is CRC value.
8. The CRC value is stored in a packet in the order of lower 8 bits → upper 8 bits.

Ex.

The calculation example of 16-bit CRC (for MODBUS)

■Packet example

Station number	**Function code**	**16-bit CRC**	
02H	07H	41H	12H

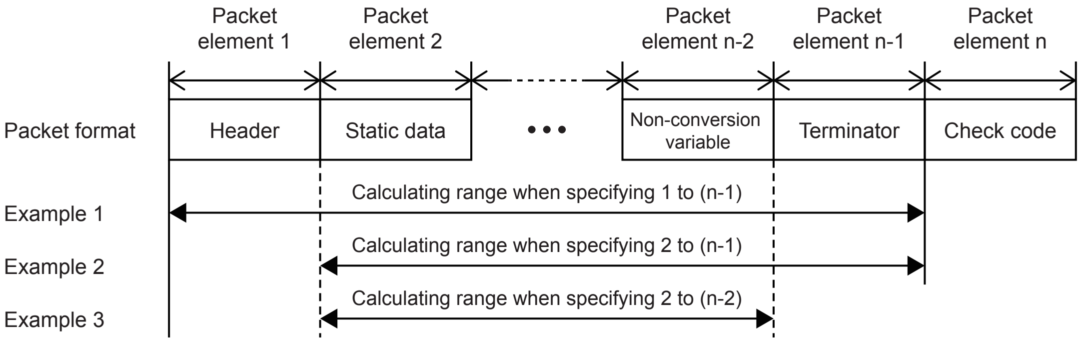
■Calculation method of 16-bit CRC (for MODBUS) for the above packet example

CRC error checking procedure	**16-bit register (MSB)**		**Flag**	**Calculating procedure**
(Load a 16-bit register whose bits are all '1'.) 02H (Station number) Exclusive OR (XOR)	1111 1111 — 1111 1111	1111 1111 0000 0010 1111 1101		1 to 2
Shift 1 Generator polynomial Exclusive OR (XOR)	0111 1111 1010 0000 1101 1111	1111 1110 0000 0001 1111 1111	1	3 to 4
Shift 2 Generator polynomial Exclusive OR (XOR)	0110 1111 1010 0000 1100 1111	1111 1111 0000 0001 1111 1110	1	5
Shift 3 Shift 4 Generator polynomial Exclusive OR (XOR)	0110 0111 0011 0011 1010 0000 1001 0011	1111 1111 1111 1111 0000 0001 1111 1110	0 1	
Shift 5 Shift 6 Generator polynomial Exclusive OR (XOR)	0100 1001 0010 0100 1010 0000 1000 0100	1111 1111 1111 1111 0000 0001 1111 1110	0 1	
Shift 7 Shift 8 Generator polynomial Exclusive OR (XOR)	0100 0010 0010 0001 1010 0000 1000 0001	0111 1111 0011 1111 0000 0001 0011 1110	0 1	

CRC error checking procedure	16-bit register (MSB)		Flag	Calculating procedure
07H (Function code) Exclusive OR (XOR)	— 1000 0001	0000 0111 0011 1001		6
Shift 1 Generator polynomial Exclusive OR (XOR)	0100 0000 1010 0000 1110 0000	1001 1100 0000 0001 1001 1101	1	7
Shift 2 Generator polynomial Exclusive OR (XOR)	0111 0000 1010 0000 1101 0000	0100 1110 0000 0001 0100 1111	1	
Shift 3 Generator polynomial Exclusive OR (XOR)	0110 1000 1010 0000 1100 1000	0010 0111 0000 0001 0010 0110	1	
Shift 4 Shift 5 Generator polynomial Exclusive OR (XOR)	0110 0100 0011 0010 1010 0000 1001 0010	0001 0011 0000 1001 0000 0001 0000 1000	0 1	
Shift 6 Shift 7 Shift 8	0100 1001 0010 0100 0001 0010	0000 0100 1000 0010 0100 0001	0 0 0	
CRC value	12H	41H		8

Check code calculation range

The following shows specification examples of the check code calculation code.



Example 1: Calculating range when 1 and n-1 are specified as the start and the end respectively

Example 2: Calculating range when 2 and n-1 are specified as the start and the end respectively

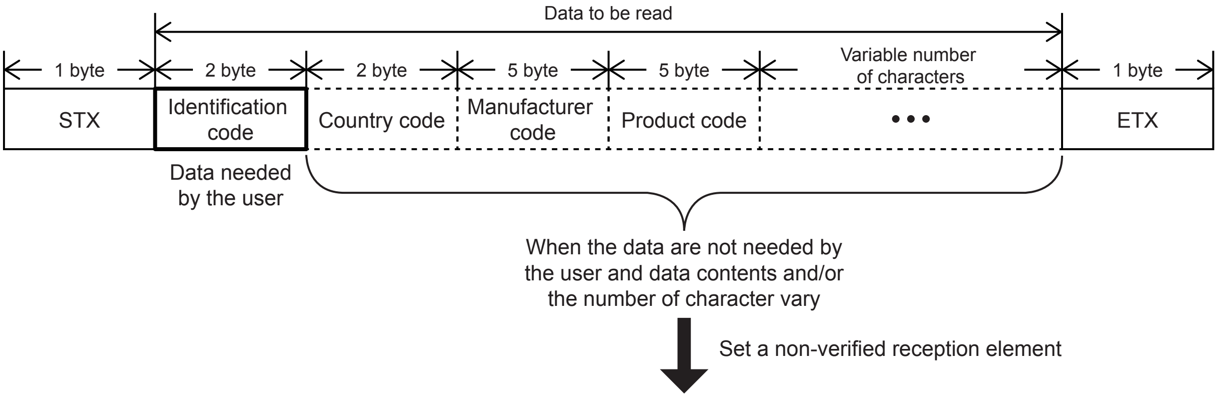
Example 3: Calculating range when 2 and n-2 are specified as the start and the end respectively

Data examples of Non-verified reception

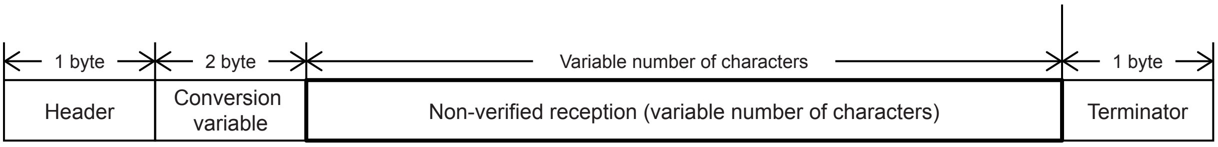
Shows the data examples of element non-verified reception that can be placed in a packet.

The usage example for non-verified reception is shown below.

■Example of format of packet from other device



■Example of packet setting



Using a non-verified reception element has the following advantages in the case of the packet format shown above.

- Only the necessary data can be stored in the device memory of a CPU module and buffer memory.
- A single protocol (packet) can handle receive packets that includes data whose contents vary each time.

Appendix 6 ASCII Code Table

ASCII code table (8-bit code expressed in hexadecimal)

Hexadecimal value	0	1	2	3	4	5	6	7
0		DLE	SP	0	@	P	`	p
1	SOH	DC1	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3	#	3	C	S	c	s
4	EOT	DC4	$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[	k	{
C	FF	FS	,	<	L	\	l	|
D	CR	GS	-	=	M	]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	DEL

Appendix 7 Frame Specifications

Procedure for generating of CRC

The error check in the MODBUS serial communication (RTU mode) is conducted by CRC (Cyclic Redundancy Check).
A procedure for generating a CRC is:

1. Load the register whose 16 bits are all "1" (FFFFH). Call this the CRC register.
2. Exclusive OR the first 8 bit byte of the message with the low-order byte of the 16 bit CRC register, putting the result in the CRC register.
3. Shift the CRC register one bit to the right (toward the Least Significant Bit), zero-filling the MSB (Most Significant bit).
4. Check the carry flag.

- If the carry flag was 0: Repeat Step 3 (another shift).
- If the carry flag was 1: Exclusive OR the CRC register with the value A001H (1010 0000 0000 0001).

5. Repeat Steps 3 and 4 until 8 shifts have been executed. When this is done, a complete 8 bit byte will have been processed.
6. Repeat Steps 2 through 5 for the next 8 bit byte of the message. Continue doing this until all bytes have been processed.
7. The final content of the CRC register is the CRC value.
8. When the CRC is placed in the message, the upper 8 bits is put in after the lower 8 bits.

The following is a calculation example in the case where function code 05H is sent to station No. (address field) 2.

CRC error check procedure	16-bit register				Carry Flag
(Load the register whose 16 bits are all "1")	1111	1111	1111	1111	
02H (Station No.)	0000	0000	0000	0010	
Exclusive OR (XOR)	1111	1111	1111	1101	
Shift 1	0111	1111	1111	1110	1
Generator value	1010	0000	0000	0001	
Exclusive OR (XOR)	1101	1111	1111	1111	
Shift 2	0110	1111	1111	1111	1
Generator value	1010	0000	0000	0001	
Exclusive OR (XOR)	1100	1111	1111	1110	
Shift 3	0110	0111	1111	1111	0
Shift 4	0011	0011	1111	1111	1
Generator value	1010	0000	0000	0001	
Exclusive OR (XOR)	1001	0011	1111	1110	
Shift 5	0100	1001	1111	1111	0
Shift 6	0010	0100	1111	1111	1
Generator value	1010	0000	0000	0001	
Exclusive OR (XOR)	1000	0100	1111	1110	
Shift 7	0100	0010	0111	1111	0
Shift 8	0010	0001	0011	1111	1
Generator value	1010	0000	0000	0001	
Exclusive OR (XOR)	1000	0001	0011	1110	
05H (Function code)	0000	0000	0000	0101	
Exclusive OR (XOR)	1000	0001	0011	1011	
Shift 1	0100	0000	1001	1101	1
Generator value	1010	0000	0000	0001	
Exclusive OR (XOR)	1110	0000	1001	1100	
Shift 2	0111	0000	0100	1110	0
Shift 3	0011	1000	0010	0111	0
Shift 4	0001	1100	0001	0011	1
Generator value	1010	0000	0000	0001	
Exclusive OR (XOR)	1011	1100	0001	0010	
Shift 5	0101	1110	0000	1001	0
Shift 6	0010	1111	0000	0100	1
Generator value	1010	0000	0000	0001	
Exclusive OR (XOR)	1000	1111	0000	0101	
Shift 7	0100	0111	1000	0010	1
Generator value	1010	0000	0000	0001	
Exclusive OR (XOR)	1110	0111	1000	0011	
Shift 8	0111	0011	1100	0001	1
Generator value	1010	0000	0000	0001	
Exclusive OR (XOR)	1101	0011	1100	0000	
CRC value	D3H		C0H		

Address field	Function code	CRC (Error check)	
(02H)	(05H)	(C0H)	(D3H)

MODBUS protocol data unit formats

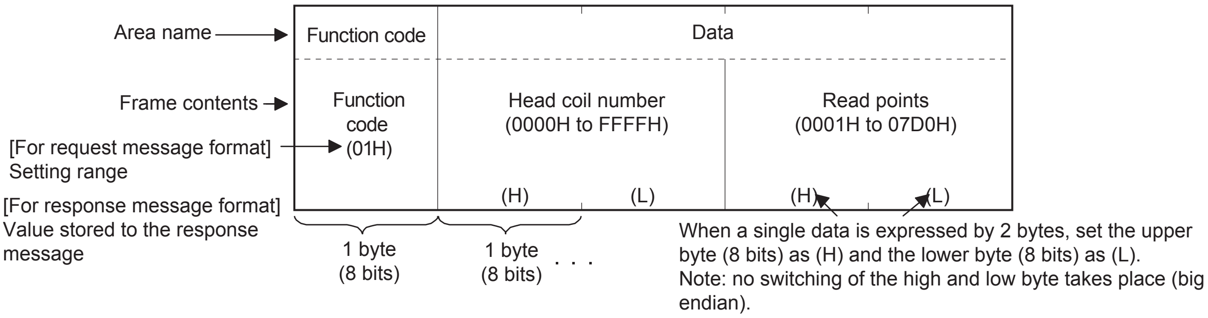
This section explains MODBUS protocol data unit formats of the MODBUS standard functions.

The MODBUS protocol data unit contains request messages sent from the master to a slave and response messages sent from the slave to the master.

How to see the request/response message formats

■Request/Response message format diagram

The following shows how to see the request/response message format diagrams provided in ☞ Page 901 Read coils to ☞ Page 909 Read/Write multiple registers.



When a single data is expressed by 2 bytes, set the upper byte (8 bits) as (H) and the lower byte (8 bits) as (L).
Note: no switching of the high and low byte takes place (big endian).

■Response message format

The response message formats issued from the slave to the master differs depending on whether the slave has normally completed or failed to handle the requested processing (read/write).

The formats for normal and error completions are shown in ☞ Page 901 Read coils to ☞ Page 909 Read/Write multiple registers.

Precautions

■When the slave receives a broadcast request message

Although the processing requested by the request message is performed, no response message is sent to the master.

■When the processing is completed with error at the slave

When the processing (read/write) requested by the request message is completed with error, an exception code is sent to the master.

Refer to "Response message formats (when completed with an error)" in ☞ Page 901 Read coils to ☞ Page 909 Read/Write multiple registers.

■Storage location of exception code and error code

Refer to the following sections for the storage location, confirmation methods, and other detailed contents.

MODBUS serial communication: ☞ Page 541 Related Devices

MODBUS/TCP communication: ☞ Page 513 Related Devices

Read coils

Reads the status (ON/OFF) of one or more coils.
Supports the MODBUS serial communication and MODBUS/TCP communication.

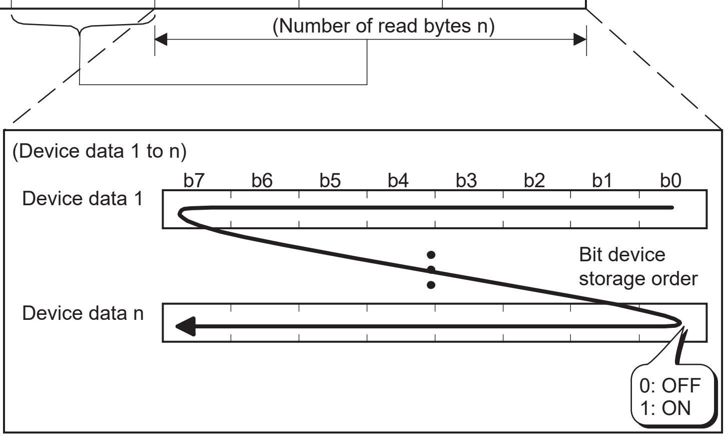
■Request message format (Master → Slave)

Function code	Data			
Function code (01H)	Head coil number (0000H to FFFFH)		Read points (0001H to 07D0H)	
	(H)	(L)	(H)	(L)

■Response message format (Slave → Master)

(When completed normally)

Function code	Data			
Function code (01H)	Number of read bytes n	Device data 1	· · ·	Device data n



- The read coil statuses are stored in order from low-order to high-order bits.

(When completed with an error)

Function code	Data
Function code (81H)	Exception code[*1]

---

*1 Exception and error codes are stored in special registers in the case of error completion. Refer to the following sections for the storage location, confirmation methods, and other detailed contents.
MODBUS serial communication: ☞ Page 541 Related Devices
MODBUS/TCP communication: ☞ Page 513 Related Devices

Read inputs

Reads the status (ON/OFF) of one or more inputs.

Supports the MODBUS serial communication and MODBUS/TCP communication.

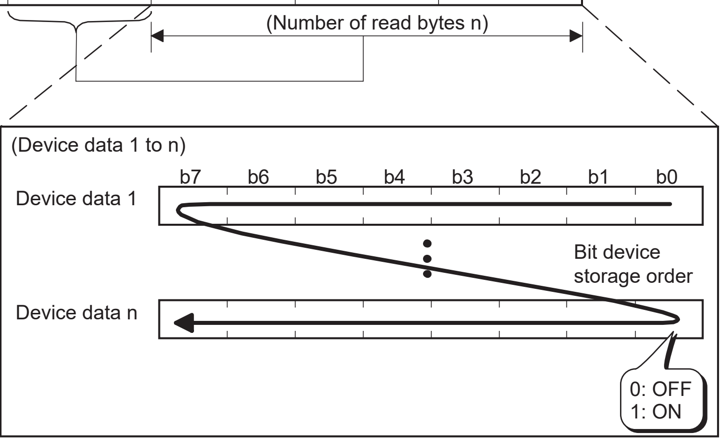
■Request message format (Master → Slave)

Function code	Data			
Function code (02H)	Head input number (0000H to FFFFH)		Read points (0001H to 07D0H)	
	(H)	(L)	(H)	(L)

■Response message format (Slave → Master)

(When completed normally)

Function code	Data			
Function code (02H)	Number of read bytes n	Device data 1	· · ·	Device data n



- The read input statuses are stored in order from low-order to high-order bits.

(When completed with an error)

Function code	Data
Function code (82H)	Exception code[1]

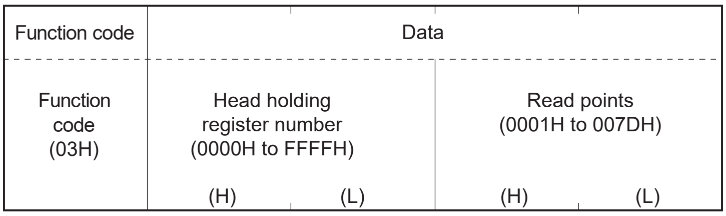
[1] Exception and error codes are stored in special registers in the case of error completion. Refer to the following sections for the storage location, confirmation methods, and other detailed contents.
MODBUS serial communication: ☞ Page 541 Related Devices
MODBUS/TCP communication: ☞ Page 513 Related Devices

Read holding registers

Reads the values of one or more holding registers.

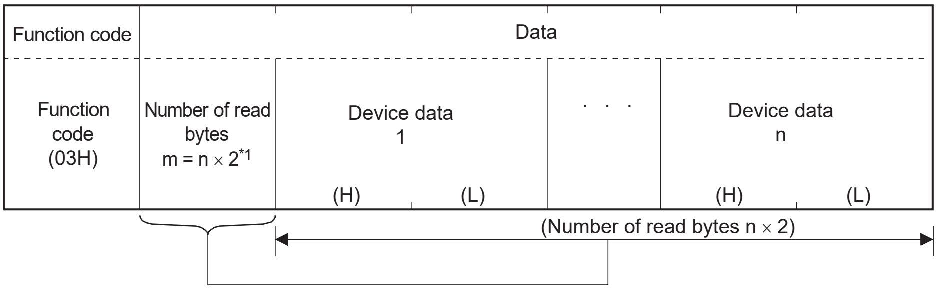
Supports the MODBUS serial communication and MODBUS/TCP communication.

■Request message format (Master → Slave)



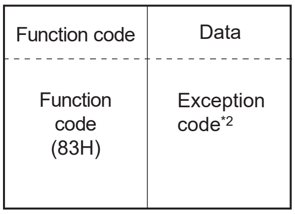
■Response message format (Slave → Master)

(When completed normally)



*1 For example, if n = 4, the number of read bytes is calculated as 4 × 2 = 8 bytes.

(When completed with an error)



*2 Exception and error codes are stored in special registers in the case of error completion. Refer to the following sections for the storage location, confirmation methods, and other detailed contents.

MODBUS serial communication: ☞ Page 541 Related Devices

MODBUS/TCP communication: ☞ Page 513 Related Devices

Read input registers

Reads the values of one or more input registers.

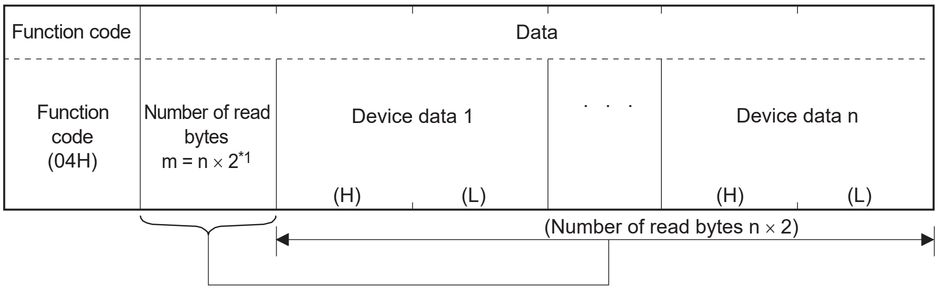
Supports the MODBUS serial communication and MODBUS/TCP communication.

■Request message format (Master → Slave)

Function code	Data			
Function code (04H)	Head input register number (0000H to FFFFH)		Read points (0001H to 007DH)	
	(H)	(L)	(H)	(L)

■Response message format (Slave → Master)

(When completed normally)



*1 For example, if n = 4, the number of read bytes is calculated as 4 × 2 = 8 bytes.

(When completed with an error)

Function code	Data
Function code (84H)	Exception code*2

*2 Exception and error codes are stored in special registers in the case of error completion. Refer to the following sections for the storage location, confirmation methods, and other detailed contents.

MODBUS serial communication: ☞ Page 541 Related Devices

MODBUS/TCP communication: ☞ Page 513 Related Devices

Write single coil

Writes a value (ON/OFF) to one coil.

Supports the MODBUS serial communication and MODBUS/TCP communication.

■Request message format (Master → Slave)

Function code	Data			
Function code (05H)	Coil number (0000H to FFFFH)		ON/OFF specification 0000H: OFF FF00H: ON	
	(H)	(L)	(H)	(L)

■Response message format (Slave → Master)

(When completed normally)

The slave returns the request message received from the master without change.

(When completed with an error)

Function code	Data
Function code (85H)	Exception code[*1]

[*1] Exception and error codes are stored in special registers in the case of error completion. Refer to the following sections for the storage location, confirmation methods, and other detailed contents.
MODBUS serial communication: ☞ Page 541 Related Devices
MODBUS/TCP communication: ☞ Page 513 Related Devices

Write single register

Writes a value to one holding register.

Supports the MODBUS serial communication and MODBUS/TCP communication.

■Request message format (Master → Slave)

Function code	Data			
Function code (06H)	Holding register number (0000H to FFFFH)		Write data (0000H to FFFFH)	
	(H)	(L)	(H)	(L)

■Response message format (Slave → Master)

(When completed normally)

The slave returns the request message received from the master without change.

(When completed with an error)

Function code	Data
Function code (86H)	Exception code[*1]

[*1] Exception and error codes are stored in special registers in the case of error completion. Refer to the following sections for the storage location, confirmation methods, and other detailed contents.
MODBUS serial communication: ☞ Page 541 Related Devices
MODBUS/TCP communication: ☞ Page 513 Related Devices

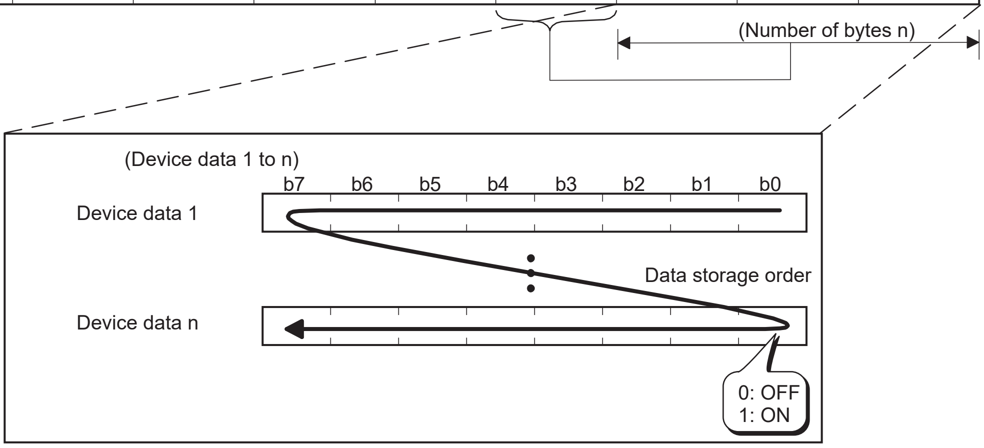
Write multiple coils

Writes values (ON/OFF) to multiple coils.

Supports the MODBUS serial communication and MODBUS/TCP communication.

■Request message format (Master → Slave)

Function code	Data					
Function code (0FH)	Head coil number (0000H to FFFFH) (H) (L)	Write points*1 (0001H to 07B0H) (H) (L)	Number of bytes n*1 (0001H to 00F6H)	Device data 1	. . .	Device data n



The values (ON/OFF) stored into the device data 1 to n are written to the coils in order from low-order to high-order bits of the device data.

*1 Since the number of bytes is automatically calculated, no settings are required from ADPRW instruction and the predefined protocol support function tool.

■Response message format (Slave → Master)

(When completed normally)

Function code	Data	
Function code (0FH)	Head coil number (The same head coil number value as in the request message is stored.) (H) (L)	Write points (The same write points value as in the request message is stored.) (H) (L)

(When completed with an error)

Function code	Data
Function code (8FH)	Exception code*1

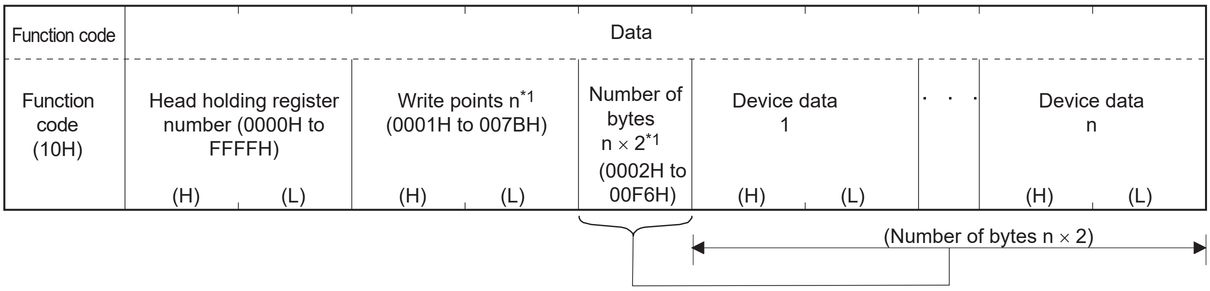
*1 Exception and error codes are stored in special registers in the case of error completion. Refer to the following sections for the storage location, confirmation methods, and other detailed contents.
MODBUS serial communication: ☞ Page 541 Related Devices
MODBUS/TCP communication: ☞ Page 513 Related Devices

Write multiple registers

Writes values to multiple holding registers.

Supports the MODBUS serial communication and MODBUS/TCP communication.

■Request message format (Master → Slave)



Function code	Data									
Function code (10H)	Head holding register number (0000H to FFFFH)		Write points n[*1] (0001H to 007BH)		Number of bytes n × 2[*1] (0002H to 00F6H)	Device data 1		· · ·	Device data n	
	(H)	(L)	(H)	(L)		(H)	(L)		(H)	(L)

*1 Since the number of bytes is automatically calculated, no settings are required from ADPRW instruction and the predefined protocol support function tool.

■Response message format (Slave → Master)

(When completed normally)

Function code	Data			
Function code (10H)	Head holding register number (The same head holding register value as in the request message is stored.)		Write points (The same write points value as in the request message is stored.)	
	(H)	(L)	(H)	(L)

(When completed with an error)

Function code	Data
Function code (90H)	Exception code[*1]

*1 Exception and error codes are stored in special registers in the case of error completion. Refer to the following sections for the storage location, confirmation methods, and other detailed contents.
MODBUS serial communication: ☞ Page 541 Related Devices
MODBUS/TCP communication: ☞ Page 513 Related Devices

Mask write register

Masks the values stored in a single holding register with AND or OR and writes the value. The masked values written to the holding register are as shown below.

- Value to be written = (Register current value ∧ AND mask value) ∨ (OR mask value ∧ $\overline{\text{AND mask value}}$)

Supported only by the MODBUS/TCP communication.

■Request message format (Master → Slave)

Function code	Data					
Function code (16H)	Target holding register number (0000H to FFFFH)		AND mask value (0000H to FFFH)		OR mask value (0000H to FFFFH)	
	(H)	(L)	(H)	(L)	(H)	(L)

■Response message format (Slave → Master)

(When completed normally)

Function code	Data
Function code (96H)	Exception code*1

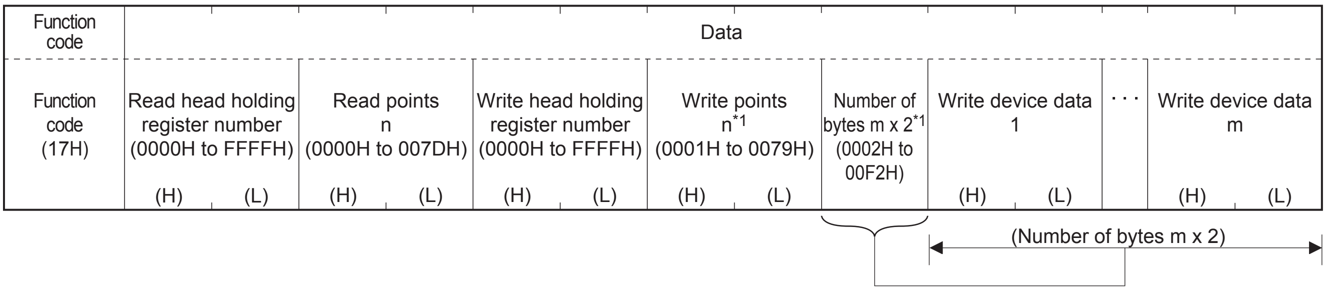
*1 Exception and error codes are stored in special registers in the case of error completion. Refer to ☞ Page 513 Related Devices for the storage location, confirmation methods, and other detailed contents.

Read/Write multiple registers

Reads from and writes to multiple holding registers. Writing is executed first and reading is then executed.

Supported only by the MODBUS/TCP communication.

■Request message format (Master → Slave)

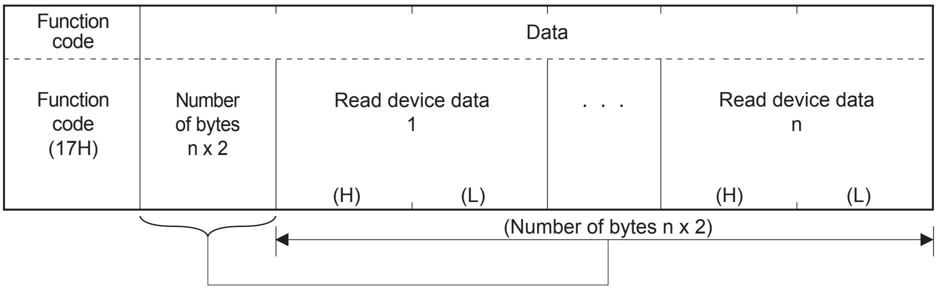


Function code	Data							
Function code (17H)	Read head holding register number (0000H to FFFFH) (H) (L)	Read points n (0000H to 007DH) (H) (L)	Write head holding register number (0000H to FFFFH) (H) (L)	Write points n[1] (0001H to 0079H) (H) (L)	Number of bytes m x 2[1] (0002H to 00F2H)	Write device data 1 (H) (L)	. . .	Write device data m (H) (L)

[1] Since the number of bytes is automatically calculated, no settings are required from the predefined protocol support function tool.

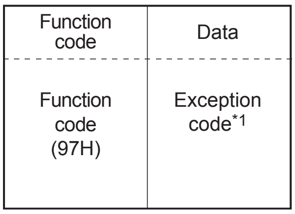
■Response message format (Slave → Master)

(When completed normally)



Function code	Data			
Function code (17H)	Number of bytes n x 2	Read device data 1 (H) (L)	. . .	Read device data n (H) (L)

(When completed with an error)



Function code	Data
Function code (97H)	Exception code[1]

[1] Exception and error codes are stored in special registers in the case of error completion. Refer to ☞ Page 513 Related Devices for the storage location, confirmation methods, and other detailed contents.

Appendix 8 Initial Values of MODBUS Device Allocation

Parameter initial value of MODBUS device allocation

The FX5 dedicated pattern and the FX3 compatible pattern are provided for the parameter initial values.

■FX5 dedicated pattern

MODBUS address <Bit device>		FX5 device			
		Coil (read/write)		Input (read only)	
FX5S/FX5UJ	FX5U/FX5UC	FX5S/FX5UJ	FX5U/FX5UC	FX5S/FX5UJ	FX5U/FX5UC
0000H to 03FFH	0000H to 03FFH	Y0 to 1023	Y0 to 1023	X0 to 1023	X0 to 1023
0400H to 1FFFH	0400H to 1FFFH	—	—	—	—
2000H to 3DFFH	2000H to 3DFFH	M0 to 7679	M0 to 7679	—	—
3E00H to 4FFFH	3E00H to 4FFFH	—	—	—	—
5000H to 57FFH	5000H to 57FFH	SM0 to 2047	SM0 to 2047	—	—
5800H to 75FFH	5800H to 75FFH	L0 to 7679	L0 to 7679	—	—
7600H to 77FFH	7600H to 77FFH	—	—	—	—
7800H to 7FFFH	7800H to 78FFH	B0 to 2047	B0 to 255	—	—
8000H to 97FFH	7900H to 97FFH	—	—	—	—
9800H to 987FH	9800H to 987FH	F0 to 127	F0 to 127	—	—
9880H to 9FFFH	9880H to 9FFFH	—	—	—	—
A000H to A7FFH	A000H to A0FFH	SB0 to 2047	SB0 to 255	—	—
A800H to AFFFH	A100H to AFFFH	—	—	—	—
B000H to BFFFH	B000H to BFFFH	S0 to 4095	S0 to 4095	—	—
C000H to CFFFH	C000H to CFFFH	—	—	—	—
D000H to D1FFH	D000H to D1FFH	TC0 to 511	TC0 to 511	—	—
D200H to D7FFH	D200H to D7FFH	—	—	—	—
D800H to D9FFH	D800H to D9FFH	TS0 to 511	TS0 to 511	—	—
DA00H to DFFFH	DA00H to DFFFH	—	—	—	—
E000H to E00FH	E000H to E00FH	STC0 to 15	STC0 to 15	—	—
E010H to E7FFH	E010H to E7FFH	—	—	—	—
E800H to E80FH	E800H to E80FH	STS0 to 15	STS0 to 15	—	—
E810H to EFFFH	E810H to EFFFH	—	—	—	—
F000H to F0FFH	F000H to F0FFH	CC0 to 255	CC0 to 255	—	—
F100H to F7FFH	F100H to F7FFH	—	—	—	—
F800H to F8FFH	F800H to F8FFH	CS0 to 255	CS0 to 255	—	—
F900H to FFFFH	F900H to FFFFH	—	—	—	—

MODBUS address <Word device>		FX5 device			
		Input register (read only)		Holding register (read/write)	
FX5S/FX5UJ	FX5U/FX5UC	FX5S/FX5UJ	FX5U/FX5UC	FX5S/FX5UJ	FX5U/FX5UC
0000H to 1F3FH	0000H to 1F3FH	—	—	D0 to 7999	D0 to 7999
1F40H to 4FFFH	1F40H to 4FFFH	—	—	—	—
5000H to 770FH	5000H to 770FH	—	—	SD0 to 9999	SD0 to 9999
7710H to 77FFH	7710H to 77FFH	—	—	—	—
7800H to 7BFFH	7800H to 79FFH	—	—	W0 to 1023	W0 to 511
7C00H to 9FFFH	7A00H to 9FFFH	—	—	—	—
A000H to A3FFH	A000H to A0FFH	—	—	SW0 to 1023	SW0 to 511
A400H to CFFFH	A100H to CFFFH	—	—	—	—
D000H to D1FFH	D000H to D1FFH	—	—	TN0 to 511	TN0 to 511
D200H to DFFFH	D200H to DFFFH	—	—	—	—
E000H to E00FH	E000H to E00FH	—	—	STN0 to 15	STN0 to 15
E010H to EFFFH	E010H to EFFFH	—	—	—	—

MODBUS address <Word device>		FX5 device			
		Input register (read only)		Holding register (read/write)	
FX5S/FX5UJ	FX5U/FX5UC	FX5S/FX5UJ	FX5U/FX5UC	FX5S/FX5UJ	FX5U/FX5UC
F000H to F0FFH	F000H to F0FFH	—	—	CN0 to 255	CN0 to 255
F100H to FFFFH	F100H to FFFFH	—	—	—	—

■FX3 compatible pattern

MODBUS address <Bit device>	FX5 device	
	Coil (read/write)	Input (read only)
0000H to 1DFFH	M0 to 7679	—
1E00H to 1FFFH	SM8000 to 8511	—
2000H to 2FFFH	S0 to 4095	—
3000H to 31FFH	TS0 to 511	—
3200H to 32FFH	CS0 to 255	—
3300H to 33FFH	Y0 to 377	—
3400H to 34FFH	—	X0 to 377
3500H to FFFFH	—	—

MODBUS address <Word device>	FX5 device	
	Input register (read only)	Holding register (read/write)
0000H to 1F3FH	—	D0 to 7999
1F40H to 213FH	—	SD8000 to 8511
2140H to A13FH	—	R0 to 32767
A140H to A33FH	—	TN0 to 511
A340H to A407H	—	CN0 to 199
A408H to A477H	—	LCN0 to 55
A478H to A657H	—	M0 to 7679
A658H to A677H	—	SM8000 to 8511
A678H to A777H	—	S0 to 4095
A778H to A797H	—	TS0 to 511
A798H to A7A7H	—	CS0 to 255
A7A8H to A7B7H	—	Y0 to 377
A7B8H to A7BCH	X0 to 377	LCS0 to 63
A7BDH to A7C7H		—
A7C8H to FFFFH	—	—

Appendix 9 Device Memory Extension Specification

The following accesses are available by setting the subcommand of request data to 008□.

- Access to module access device
- Access with indirect specification of the device No. by using index register or long index register
- Access with indirect specification of the device No. by using values stored in word device

Restriction

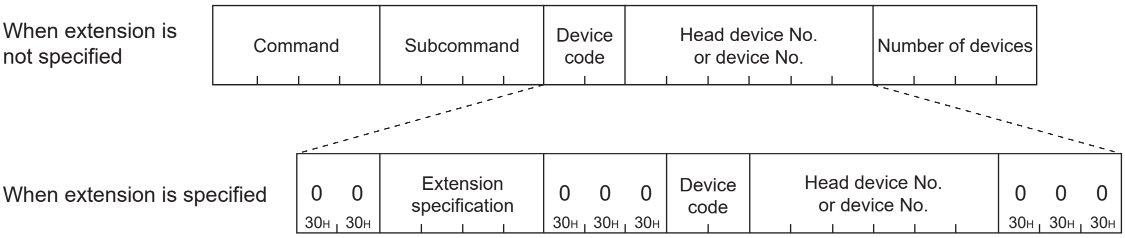
Device memory extension specification is available only for 3E frame.

Access to module access device

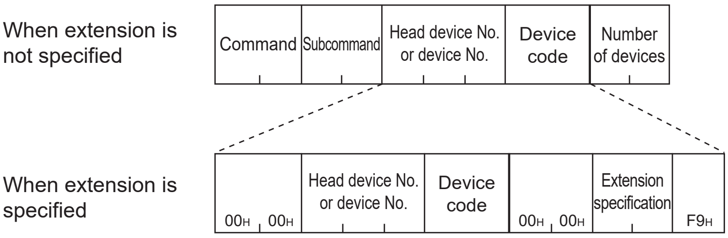
Access to the buffer memory of SLMP compatible devices or intelligent function modules.

Request data

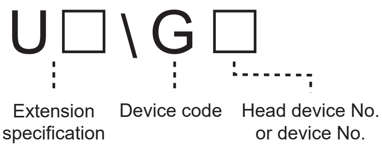
ASCII



Binary



The following shows the module access device and request data.



Point

Devices of FX5 CPU module can be accessed by specifying 0 in "extension specification" of commands which can specify multiple devices. (☞ Page 611 Device range) However, when specifying 008□ in "subcommand", specify the device in the message format shown above. Message formats when extension is not specified and message formats when extension is specified cannot coexist in the same message.

■Command

The following commands can be used for accessing.

Item		Command
Type	Operation	
Device	Read	0401
	Write	1401
	Read Random	0403
	Write Random	1402
	Read Block	0406
	Write Block	1406

■Subcommand

Subcommand	
ASCII code	Binary code
0 0 8 0 30H 30H 38H 30H	80H 00H
0 0 8 2 30H 30H 38H 32H	82H 00H

■Extension specification

Specify the start I/O number of intelligent function modules.

ASCII code	Binary code
Specify the start I/O number in hexadecimal (3-digit ASCII code). When described with 4-digits, specify the start I/O number with the upper 3-digits.	Specify the start I/O number in hexadecimal (2 bytes). When described with 4-digits, specify the start I/O number with the upper 3-digits.
U □ □ □ 55H Example 001 U 0 0 1 55H 30H 30H 31H	□□H □□H Example 001 01H 00H

Point

- Specify 0 when accessing buffer memory of modules other than intelligent function modules, such as CC-Link IE Field Network Ethernet adapter module.

■Device code

Specify the following device codes.

Type	Device code				Device No. range	
	ASCII code[1]		Binary code			
	2 digit code/6 digit number specification	4 digit code/8 digit number specification	2 digit code/6 digit number specification	4 digit code/8 digit number specification		
Word	G*	G***	ABH	AB00H	Specify within the device No. range of the module for access destination.	Decimal

[1] If the device text is one character only, add "*" (ASCII code: 2AH) or a space (ASCII code: 20H) after the device text.

■Head device or device No.

Specify the head device or device No. in decimal, with the same format as the message when extension is not specified.

Point

Indirect specification of the access target device No. can be performed by using the CPU module index register (Z) or long index register (LZ). (☞ Page 915 Access with indirect specification of the device No. by using index register or long index register)

Response data

The same as when extension is not specified.

Communication example

Access to the buffer memory (Address: 1) of the intelligent function module whose start I/O number is 0030H.

- When communicating data in ASCII code

(Request data)

Subcommand		Extension specification		Device code	Head device No. or device No.	
0 0 8 0	0 0	U 0 0 3	0 0 0	G *	0 0 0 0 0 1	0 0 0
30H 30H 38H 30H	30H 30H	55H 30H 30H 33H	30H 30H 30H	47H 2AH	30H 30H 30H 30H 30H 31H	30H 30H 30H

- When communicating data in binary code

(Request data)

Subcommand		Head device No. or device No.	Device code		Extension specification	
80H 00H	00H 00H	01H 00H 00H	ABH	00H 00H	03H 00H	F8H

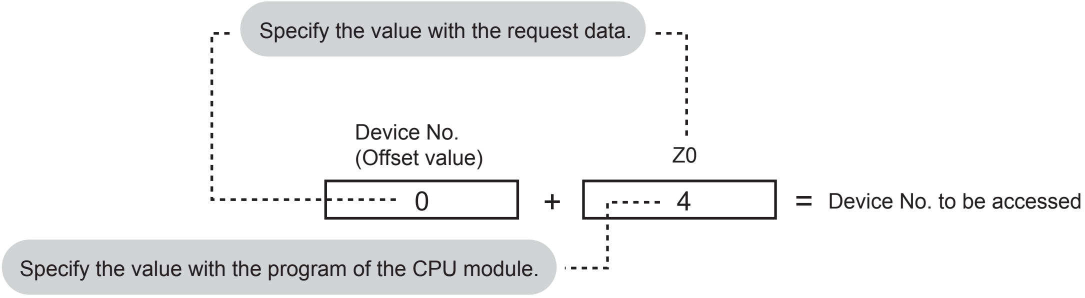
Access with indirect specification of the device No. by using index register or long index register

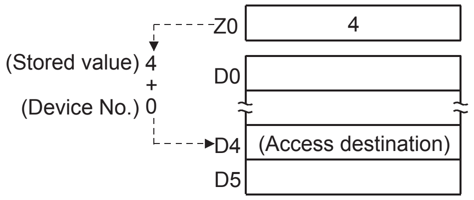
Indirect specification of the device No. can be performed by using the index register or long index register when accessing the device.

The access destination can be switched with one message, by changing the value of the index register or long index register in CPU module programs.

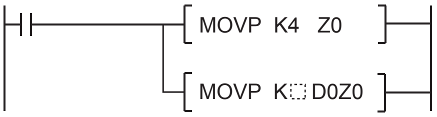
Ex.

When accessing D4 with D0 and Z0 specifications



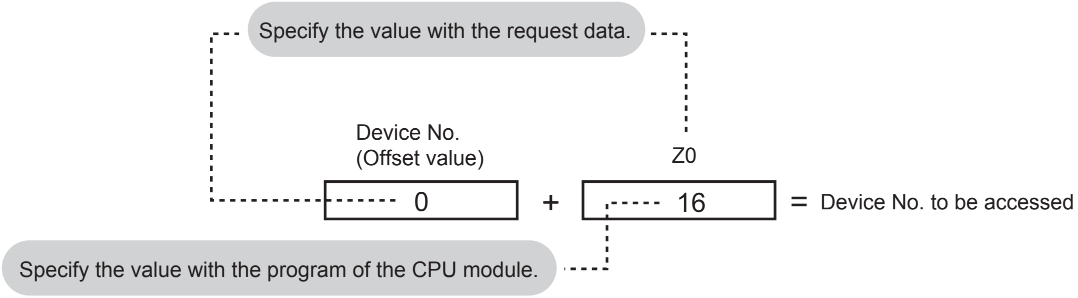


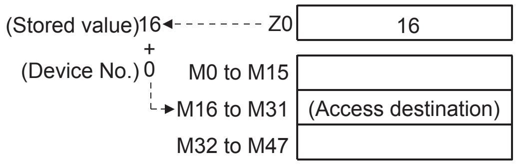
Equivalent to accessing D4 using the following program.



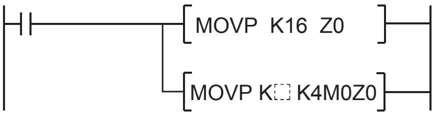
Ex.

When accessing M16 to M31 with M0 and Z0 specifications (Word units)



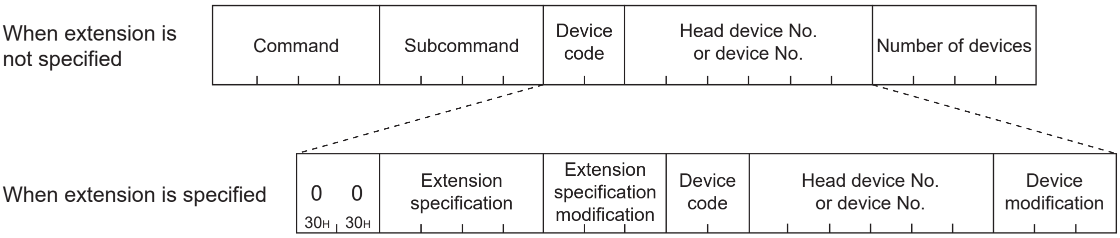


Equivalent to accessing M16 to M31 using the following program.

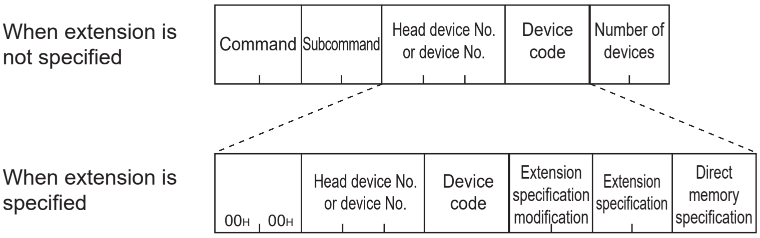


Request data

ASCII

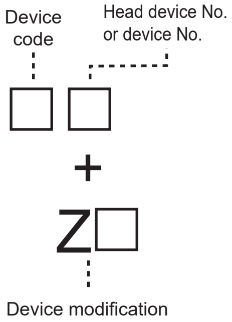


Binary

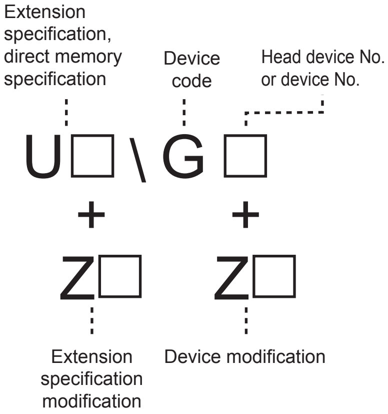


The following shows the approach for devices, index registers, long index registers and request data.

- Other than the module access device



- Module access device



Point

When specifying 008□ in "subcommand", specify the device with the message format shown above.

Message formats when extension is not specified and message formats when extension is specified cannot coexist in the same message.

■Command

The following commands can be used for accessing.

Item		Command
Type	Operation	
Device	Read Random	0403
	Write Random	1402

■Subcommand

Item	Subcommand	
	ASCII code	Binary code
When accessing in bit units	0 0 8 1 30H 30H 38H 31H	81H 00H
	0 0 8 3 30H 30H 38H 33H	83H 00H
When accessing in word units	0 0 8 0 30H 30H 38H 30H	80H 00H
	0 0 8 2 30H 30H 38H 32H	82H 00H

■Extension specification

Specify the module number.

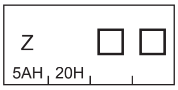
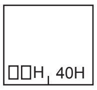
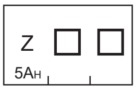
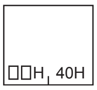
The values specified in this item turn to the offset value when performing indirect specification of the module number in "extension specification modification".

Item	ASCII code	Binary code
Module access device	Specify the module number in hexadecimal (3-digit ASCII code). U □ □ □ 55H Example 001 U 0 0 1 55H 30H 30H 31H	Specify the module number in hexadecimal (2 bytes). □□H □□H Example 001 01H 00H
Devices other than the above	Specify 0. 0 0 0 0 30H 30H 30H 30H	Specify 0. 00H 00H

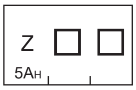
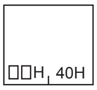
■Extension specification modification

Treat the value specified in "extension specification" as the offset value. Specify the index register or long index register number when performing indirect specification of the module number with index register or long index register.

The following value is specified when the access point is a module of the MELSEC iQ-R/iQ-F Series.

Subcommand	ASCII code	Binary code
0083 0082	Specify the number of the index register (Z) in decimal (2-digit ASCII code). • FX5S/FX5U/FX5UC CPU module: 0 to 23 • FX5UJ CPU module: 0 to 19 • Ethernet module: 0 to 24 	Specify the number of the index register (Z) in hexadecimal. • FX5S/FX5U/FX5UC CPU module: 00H to 17H • FX5UJ CPU module: 00H to 13H • Ethernet module: 00H to 18H 
0081 0080	Specify the number of the index register (Z) in decimal (2-digit ASCII code). • FX5S/FX5U/FX5UC CPU module: 0 to 23 • FX5UJ CPU module: 0 to 19 • Ethernet module: 0 to 24 	Specify the number of the index register (Z) in hexadecimal. • FX5S/FX5U/FX5UC CPU module: 00H to 17H • FX5UJ CPU module: 00H to 13H • Ethernet module: 00H to 18H 

The following value is specified when the access point is a module of the MELSEC-Q/L Series.

ASCII code	Binary code
Specify the number of the index register in decimal (2-digit ASCII code). (Specification range: 0 to 15) 	Specify the number of the index register in hexadecimal. (Specification range: 00H to 0FH) 

Point

The long index register (LZ) cannot be used in the extension specification modification.

■Device code

Specify the code of the device to be accessed. (☞ Page 611 Device range)

Specify the following device code when accessing the module access device.

Type	Device code				Device No. range	
	ASCII code[*1]		Binary code			
	2 digit code/6 digit number specification	4 digit code/8 digit number specification	2 digit code/6 digit number specification	4 digit code/8 digit number specification		
Word	G*	G***	ABH	AB00H	Specify within the device No. range of the module for access destination.	Decimal

1 If the device text is one character only, add "" (ASCII code: 2AH) or a space (ASCII code: 20H) after the device text.

■Head device or device No.

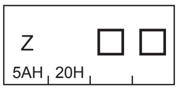
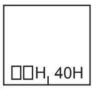
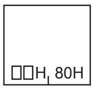
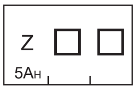
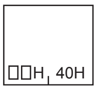
Specify the head device or device No. with the same format as the message when extension is not specified.

The values specified in this item turn to the offset value when performing indirect specification of the device No. in "device modification".

■Device modification

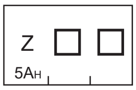
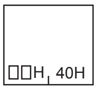
Treat the value specified in "Head device or device No." as the offset value. Specify the index register or long index register number when performing indirect specification of the device No. with index register or long index register.

The following value is specified when the access point is a module of the MELSEC iQ-R/iQ-F Series.

Subcommand	ASCII code	Binary code
0083 0082	Specify the number of the index register (Z) in decimal (2-digit ASCII code).[1] • FX5S/FX5U/FX5UC CPU module: 0 to 23 • FX5UJ CPU module: 0 to 19 • Ethernet module: 0 to 24 Specify the number of the long index register (LZ) in decimal (2-digit ASCII code). • FX5S/FX5U/FX5UC CPU module: 0 to 11 • FX5UJ CPU module: 0, 1 • Ethernet module: 0 to 12  	Specify the number of the index register (Z) in hexadecimal.[1] • FX5S/FX5U/FX5UC CPU module: 00H to 17H • FX5UJ CPU module: 00H to 13H • Ethernet module: 00H to 18H Specify the number of the long index register (LZ) in hexadecimal. • FX5S/FX5U/FX5UC CPU module: 00H to 0BH • FX5UJ CPU module: 00H and 01H • Ethernet module: 00H to 0CH  
0081 0080	Specify the number of the index register (Z) in decimal (2-digit ASCII code). • FX5S/FX5U/FX5UC CPU module: 0 to 23 • FX5UJ CPU module: 0 to 19 • Ethernet module: 0 to 24 	Specify the number of the index register (Z) in hexadecimal. • FX5S/FX5U/FX5UC CPU module: 00H to 17H • FX5UJ CPU module: 00H to 13H • Ethernet module: 00H to 18H 

[1] The device modification range of the index register (Z) is -32768 to 32767. When the device modification range is not within -32768 to 32767, use the long index register (LZ).

The following value is specified when the access point is a module of the MELSEC-Q/L Series.

ASCII code	Binary code
Specify the number of the index register in decimal (2-digit ASCII code). (Specification range: 0 to 15) 	Specify the number of the index register in hexadecimal. (Specification range: 00H to 0FH) 

■Direct memory specification (only when communicating in binary code)

Specify the device type when accessing the module access device.

Item	Binary code
Module access device	Specify F8H.
Other than the above	Specify 00H.

Response data

The same as when extension is not specified.

Communication example

Accessing the device of D100 + Z4.

- When communicating data in ASCII code

(Request data)

Subcommand		Extension specification	Extension specification modification	Device code	Head device No. or device No.	Device modification
0 0 8 0	0 0	0 0 0 0	0 0 0	D *	0 0 0 1 0 0	Z 0 4
30H, 30H, 38H, 30H	30H, 30H	30H, 30H, 30H, 30H	30H, 30H, 30H	44H, 2AH	30H, 30H, 30H, 31H, 30H, 30H	5AH, 30H, 34H

- When communicating data in binary code

(Request data)

Subcommand	Device modification	Head device No. or device No.	Device code	Extension specification modification	Extension specification	Direct memory specification
80H, 00H	04H, 40H	64H, 00H, 00H	A8H	00H, 00H	00H, 00H	00H

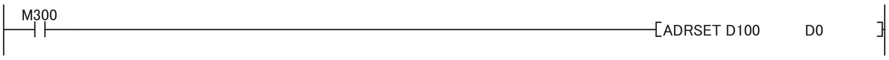
Access with indirect specification of the device No. by using the values stored in word device

Access the device corresponding to the address stored in word device (for 2 points).

Ex.

When storing the address of D100 in D0, and trying to access D100 from external devices by accessing "@D0"

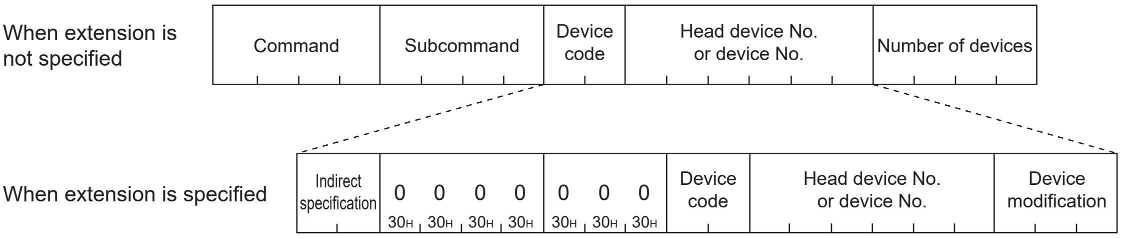
The ADRSET instruction is used on the Ethernet-equipped module side and the address of D100 is stored in D0.



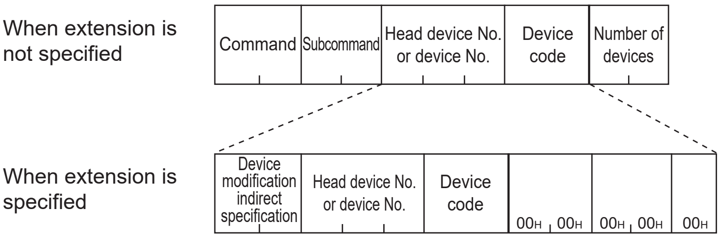
D100 can be indirectly accessed by specifying "@D0" with the request data.

Request data

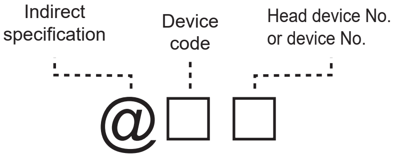
ASCII



Binary



The following shows the indirect specification devices and request data.



Point

- When specifying 008□ in "subcommand", specify the device with the message format shown above. Message formats when extension is not specified and message formats when extension is specified cannot coexist in the same message.
- The indirect specification and the device modification using index registers cannot be set simultaneously.

■Command

The following commands can be used for accessing.

Item		Command
Type	Operation	
Device	Read Random	0403
	Write Random	1402

■Subcommand

ASCII code	Binary code
0 0 8 0 30H 30H 38H 30H	80H 00H

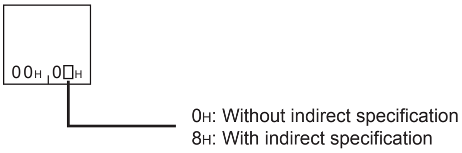
■Indirect specification, Device modification

Specify the "@" part of the indirect specification device. Indirect specification can be specified only for word devices.

When communicating data in ASCII code

0	@
30H	40H

When communicating data in binary code



0H: Without indirect specification
8H: With indirect specification

■Device code (Only word device codes can be specified at indirect specification)

Specify the code of the device to be accessed. (☞ Page 611 Device range)

■Head device or device No.

Specify the head device or device No. with the same format as the message when extension is not specified.

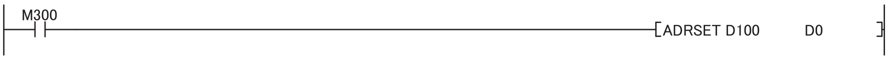
Response data

The same as when extension is not specified.

Communication example

Access to @D0. (Consider @D0 indirect specification of D100.)

At command execution, store the D100 address in D0 with the following programs.



- When communicating data in ASCII code

(Request data)

Subcommand	Indirect specification			Device code	Head device No. or device No.	Device modification
0 0 8 0 30H 30H 38H 30H	0 @ 30H 40H	0 0 0 0 30H 30H 30H 30H	0 0 0 30H 30H 30H	D * 44H 2AH	0 0 0 0 0 0 30H 30H 30H 30H 30H 30H	0 0 0 30H 30H 30H

- When communicating data in binary code

(Request data)

Subcommand	indirect specification	Head device No. or device No.	Device code			
80H 00H	00H 08H	00H 00H 00H	A8H	00H 00H	00H 00H	00H

Appendix 10 Command Comparison between MC Protocol and SLMP

The correspondence table of MC protocol and SLMP is shown below. When connecting an external device which uses MC protocol to an SLMP compatible device, check if replacement of command is required.

Applicable 3E frame command list

The message format of 3E frame of the SLMP is the same as that of the QnA compatible 3E frame of MC protocol.
The commands shown in the following table need not be replaced with SLMP commands.

MC protocol			**SLMP**	
Item	**Command**	**Subcommand**	**Type**	**Operation**
Batch read in bit units	0401	00□1	Device	Read
Batch read in word units		00□0		
Batch write in bit units	1401	00□1		Write
Batch write in word units		00□0		
Random read in word units	0403	00□0		Read Random
Random write in bit units (Test)	1402	00□1		Write Random
Random write in word units (Test)		00□0		
Multiple block batch read	0406	00□0		Read Block
Multiple block batch write	1406	00□0		Write Block
Remote RUN	1001	0000	Remote Control	Remote Run
Remote STOP	1002	0000		Remote Stop
Remote PAUSE	1003	0000		Remote Pause
Remote latch clear	1005	0000		Remote Latch Clear
Remote RESET	1006	0000		Remote Reset
CPU model name read	0101	0000		Read Type Name
Loopback test	0619	0000	Self-Test	
COM.ERR.LED off	1617	0000	Clear Error	
Remote password unlock	1630	0000	Password Unlock	
Remote password lock	1631	0000	Password Lock	

Applicable 1E frame command list

The message format of 1E frame of the SLMP is the same as that of the A compatible 1E frame of MC protocol.
The commands shown in the following table need not be replaced with SLMP commands.

MC protocol			**SLMP**	
Item	**Command**		**Operation**	**Command**
Batch Reading	BR	4252H	Batch Reading	00H
	WR	5752H		01H
	QR	5152H		
Batch Writing	BW	4257H	Batch Writing	02H
	WW	5757H		03H
	QW	5157H		
Test (Random Write)	BT	4254H	Test (Random Write)	04H
	WT	5754H		05H
	QT	5154H		
Remote RUN	RR	5252H	Remote RUN	13H
Remote STOP	RS	5253H	Remote STOP	14H
Read CPU Model Name	PC	5043H	Read PC Type Name	15H
Global	GW	4757H	—	—
Loopback Test	TT	5454H	Loopback Test	16H

Appendix 11 Device Memory Extension Specification

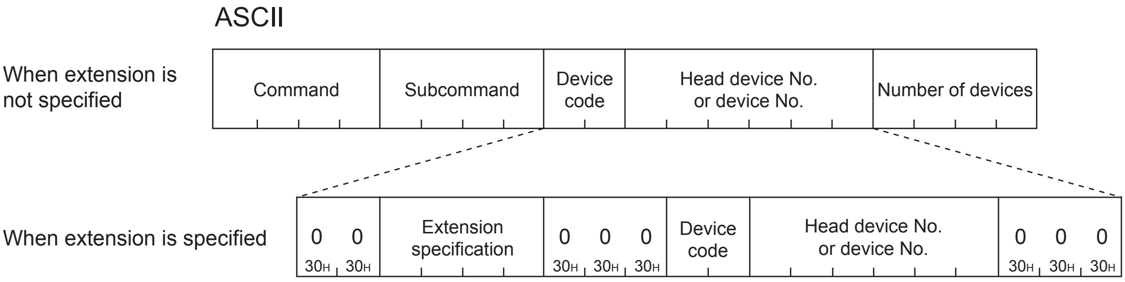
The following accesses are available by setting the subcommand of request data to 008□.

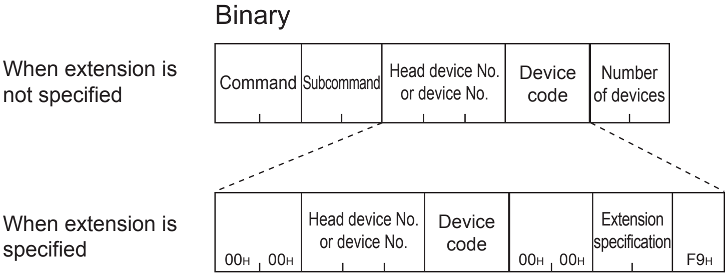
- Access to module access device
- Access with indirect specification of the device No. by using index register or long index register
- Access with indirect specification of the device No. by using values stored in word device

Access to module access device

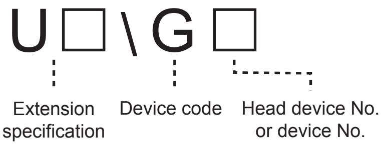
Access to the buffer memory of MC protocol compatible devices or intelligent function modules.

Request data





The following shows the module access device and request data.



Point

Devices described in ☞ Page 715 Device code list can be accessed by specifying 0 in "extension specification" of commands which can specify multiple devices. However, when specifying 008□ in "subcommand", specify the device in the message format shown above. Message formats when extension is not specified and message formats when extension is specified cannot coexist in the same message.

■Command

The following commands can be used for accessing.

Function	Command
Batch read	0401
Batch write	1401
Random read	0403
Random write	1402
Multiple block batch read	0406
Multiple block batch write	1406

■Subcommand

Subcommand	
ASCII code	**Binary code**
0 0 8 0 30H 30H 38H 30H	80H 00H
0 0 8 2 30H 30H 38H 32H	82H 00H

■Extension specification

Specify the start I/O number of intelligent function modules.

ASCII code	Binary code
Specify the start I/O number in hexadecimal (3-digit ASCII code). When described with 4-digits, specify the start I/O number with the upper 3-digits. U □ □ □ 55H Example 001 U 0 0 1 55H 30H 30H 31H	Specify the start I/O number in hexadecimal (2 bytes). When described with 4-digits, specify the start I/O number with the upper 3-digits. □□H □□H Example 001 01H 00H

■Device code

Specify the following device codes.

Type	Device code				Device No. range	
	ASCII code[1]		Binary code			
	2 digit code/6 digit number specification	4 digit code/8 digit number specification	2 digit code/6 digit number specification	4 digit code/8 digit number specification		
Word	G*	G***	ABH	AB00H	Specify within the device No. range of the module for access destination.	Decimal

[1] If the device text is one character only, add "*" (ASCII code: 2AH) or a space (ASCII code: 20H) after the device text.

■Head device or device No.

Specify the head device or device No. in decimal, with the same format as the message when extension is not specified.

Point

Indirect specification of the access target device No. can be performed by using the CPU module index register (Z) or long index register (LZ). (☞ Page 927 Access with indirect specification of the device No. by using index register or long index register)

Response data

The same as when extension is not specified.

Communication example

Access to the buffer memory (Address: 1) of the intelligent function module whose start I/O number is 0030H.

- When communicating data in ASCII code

(Request data)

Subcommand		Extension specification		Device code	Head device No. or device No.	
0 0 8 0	0 0	U 0 0 3	0 0 0	G *	0 0 0 0 0 1	0 0 0
30H 30H 38H 30H	30H 30H	55H 30H 30H 33H	30H 30H 30H	47H 2AH	30H 30H 30H 30H 30H 31H	30H 30H 30H

- When communicating data in binary code

(Request data)

Subcommand		Head device No. or device No.	Device code	Extension specification		
80H 00H	00H 00H	01H 00H 00H	ABH	00H 00H	03H 00H	F8H

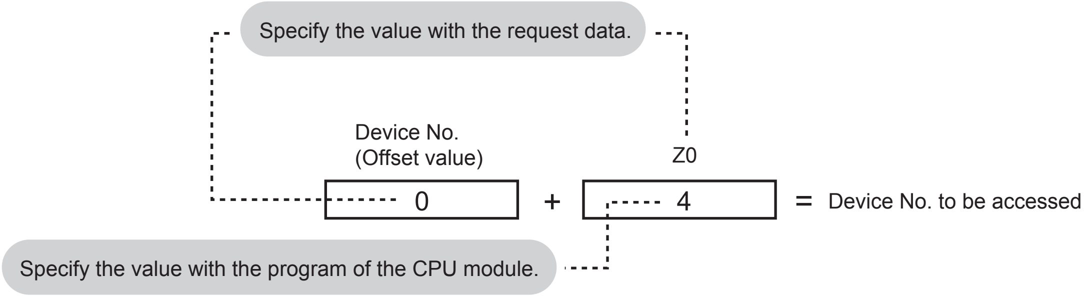
Access with indirect specification of the device No. by using index register or long index register

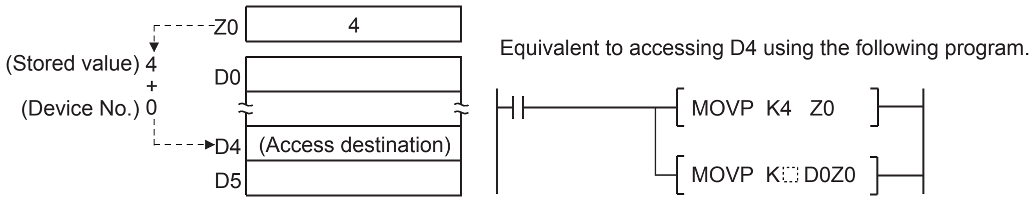
Indirect specification of the device No. can be performed by using the index register or long index register when accessing the device.

The access destination can be switched with one message, by changing the value of the index register or long index register in CPU module programs.

Ex.

When accessing D4 with D0 and Z0 specifications

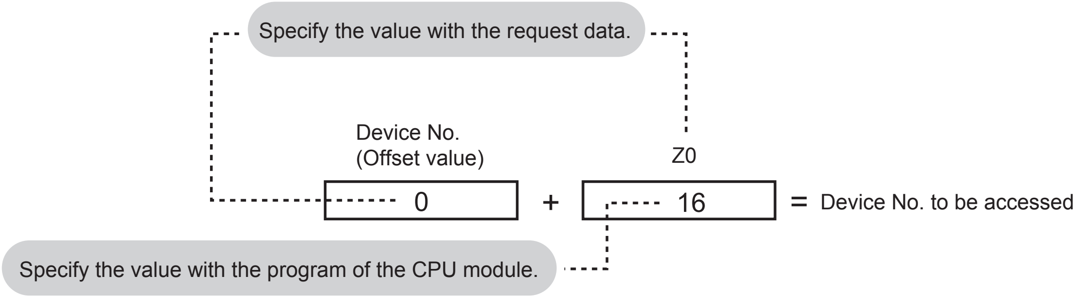


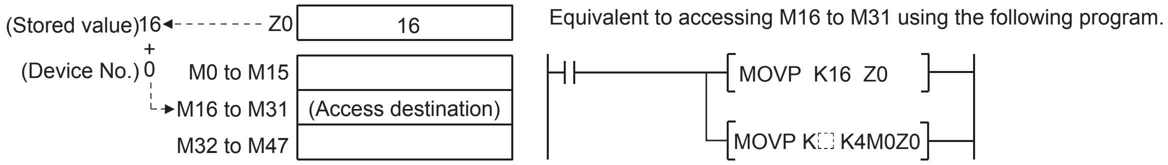


Equivalent to accessing D4 using the following program.

Ex.

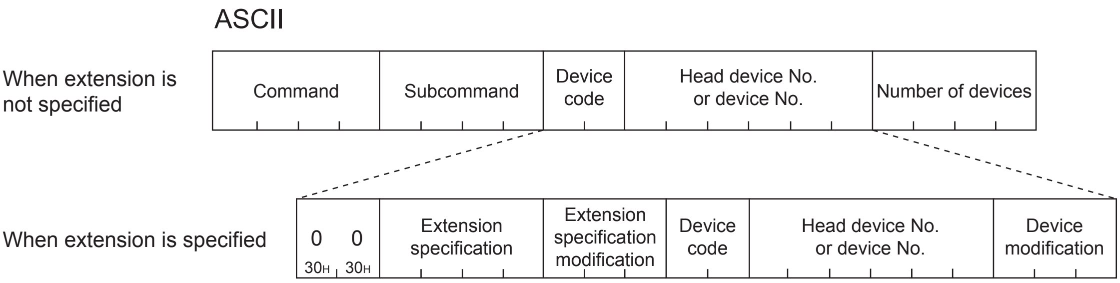
When accessing M16 to M31 with M0 and Z0 specifications (Word units)

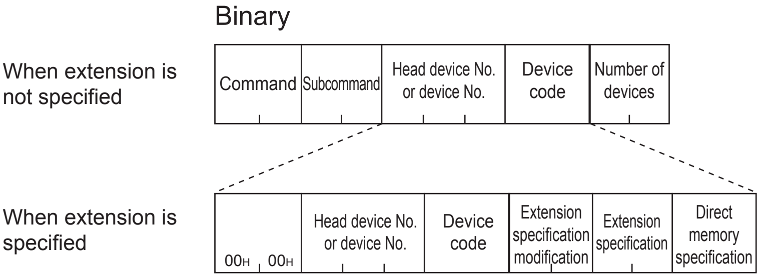




Equivalent to accessing M16 to M31 using the following program.

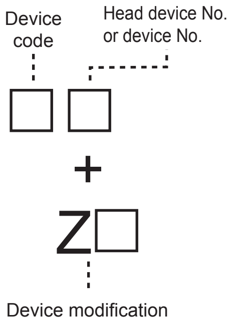
Request data



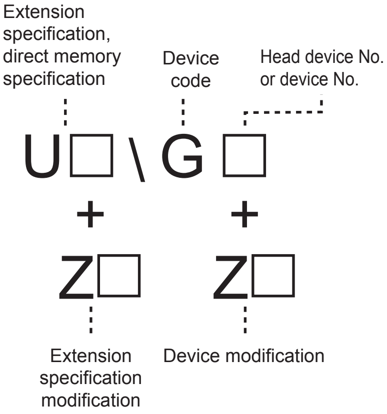


The following shows the approach for devices, index registers, long index registers and request data.

- Other than the module access device



- Module access device



Point

When specifying 008☐ in "subcommand", specify the device with the message format shown above.
Message formats when extension is not specified and message formats when extension is specified cannot coexist in the same message.

■Command

The following commands can be used for accessing.

Function	Command
Random read	0403
Random write	1402

■Subcommand

Item	Subcommand	
	ASCII code	Binary code
When accessing in bit units	0 0 8 1 30H 30H 38H 31H	81H 00H
	0 0 8 3 30H 30H 38H 33H	83H 00H
When accessing in word units	0 0 8 0 30H 30H 38H 30H	80H 00H
	0 0 8 2 30H 30H 38H 32H	82H 00H

■Extension specification

Specify the module number.

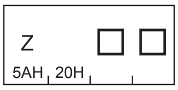
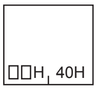
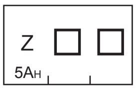
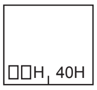
The values specified in this item turn to the offset value when performing indirect specification of the module number in "extension specification modification".

Item	ASCII code	Binary code
Module access device	Specify the module number in hexadecimal (3-digit ASCII code). Example 001 U □ □ □ 55H U 0 0 1 55H 30H 30H 31H	Specify the module number in hexadecimal (2 bytes). Example 001 □□H □□H 01H 00H
Devices other than the above	Specify 0. 0 0 0 0 30H 30H 30H 30H	Specify 0. 00H 00H

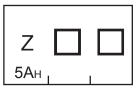
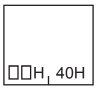
■Extension specification modification

Treat the value specified in "extension specification" as the offset value. Specify the index register or long index register number when performing indirect specification of the module number with index register or long index register.

The following value is specified when the access point is a module of the MELSEC iQ-R/iQ-F Series.

Subcommand	ASCII code	Binary code
0083 0082	Specify the number of the index register (Z) in decimal (2-digit ASCII code). • FX5S/FX5U/FX5UC CPU module: 0 to 23 • FX5UJ CPU module: 0 to 19 	Specify the number of the index register (Z) in hexadecimal. • FX5S/FX5U/FX5UC CPU module: 00H to 17H • FX5UJ CPU module: 00H to 13H 
0081 0080	Specify the number of the index register (Z) in decimal (2-digit ASCII code). • FX5S/FX5U/FX5UC CPU module: 0 to 23 • FX5UJ CPU module: 0 to 19 	Specify the number of the index register (Z) in hexadecimal. • FX5S/FX5U/FX5UC CPU module: 00H to 17H • FX5UJ CPU module: 00H to 13H 

The following value is specified when the access point is a module of the MELSEC-Q/L Series.

ASCII code	Binary code
Specify the number of the index register in decimal (2-digit ASCII code). (Specification range: 0 to 15) 	Specify the number of the index register in hexadecimal. (Specification range: 00H to 0FH) 

Point

The long index register (LZ) cannot be used in the extension specification modification.

■Device code

Specify the code of the device to be accessed. (☞ Page 715 Device code list)

Specify the following device code when accessing the module access device.

Type	Device code				Device No. range	
	ASCII code[*1]		Binary code			
	2 digit code/6 digit number specification	4 digit code/8 digit number specification	2 digit code/6 digit number specification	4 digit code/8 digit number specification		
Word	G*	G***	ABH	AB00H	Specify within the device No. range of the module for access destination.	Decimal

1 If the device text is one character only, add "" (ASCII code: 2AH) or a space (ASCII code: 20H) after the device text.

■Head device or device No.

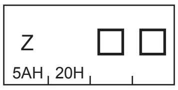
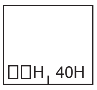
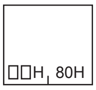
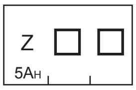
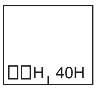
Specify the head device or device No. with the same format as the message when extension is not specified.

The values specified in this item turn to the offset value when performing indirect specification of the device No. in "device modification".

■Device modification

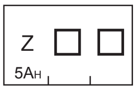
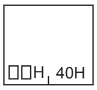
Treat the value specified in "Head device or device No." as the offset value. Specify the index register or long index register number when performing indirect specification of the device No. with index register or long index register.

The following value is specified when the access point is a module of the MELSEC iQ-R/iQ-F Series.

Subcommand	ASCII code	Binary code
0083 0082	Specify the number of the index register (Z) in decimal (2-digit ASCII code).[1] • FX5S/FX5U/FX5UC CPU module: 0 to 23 • FX5UJ CPU module: 0 to 19 Specify the number of the long index register (LZ) in decimal (2-digit ASCII code). • FX5S/FX5U/FX5UC CPU module: 0 to 11 • FX5UJ CPU module: 0, 1  	Specify the number of the index register (Z) in hexadecimal.[1] • FX5S/FX5U/FX5UC CPU module: 00H to 17H • FX5UJ CPU module: 00H to 13H Specify the number of the long index register (LZ) in hexadecimal. • FX5S/FX5U/FX5UC CPU module: 00H to 0BH • FX5UJ CPU module: 00H and 01H  
0081 0080	Specify the number of the index register (Z) in decimal (2-digit ASCII code). • FX5S/FX5U/FX5UC CPU module: 0 to 23 • FX5UJ CPU module: 0 to 19 	Specify the number of the index register (Z) in hexadecimal. • FX5S/FX5U/FX5UC CPU module: 00H to 17H • FX5UJ CPU module: 00H to 13H 

[1] The device modification range of the index register (Z) is -32768 to 32767. When the device modification range is not within -32768 to 32767, use the long index register (LZ).

The following value is specified when the access point is a module of the MELSEC-Q/L Series.

ASCII code	Binary code
Specify the number of the index register in decimal (2-digit ASCII code). (Specification range: 0 to 15) 	Specify the number of the index register in hexadecimal. (Specification range: 00H to 0FH) 

■Direct memory specification (only when communicating in binary code)

Specify the device type when accessing the module access device.

Item	Binary code
Module access device	Specify F8H.
Other than the above	Specify 00H.

Response data

The same as when extension is not specified.

Communication example

Accessing the device of D100 + Z4.

- When communicating data in ASCII code

(Request data)

Subcommand		Extension specification	Extension specification modification	Device code	Head device No. or device No.	Device modification
0 0 8 0 30_H 30_H 38_H 30_H	0 0 30_H 30_H	0 0 0 0 30_H 30_H 30_H 30_H	0 0 0 30_H 30_H 30_H	D * 44_H $2A_H$	0 0 0 1 0 0 30_H 30_H 30_H 31_H 30_H 30_H	Z 0 4 $5A_H$ 30_H 34_H

- When communicating data in binary code

(Request data)

Subcommand	Device modification	Head device No. or device No.	Device code	Extension specification modification	Extension specification	Direct memory specification
80_H 00_H	04_H 40_H	64_H 00_H 00_H	$A8_H$	00_H 00_H	00_H 00_H	00_H

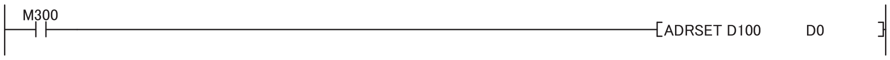
Access with indirect specification of the device No. by using the values stored in word device

Access the device corresponding to the address stored in word device (for 2 points).

Ex.

When storing the address of D100 in D0, and trying to access D100 from external devices by accessing "@D0"

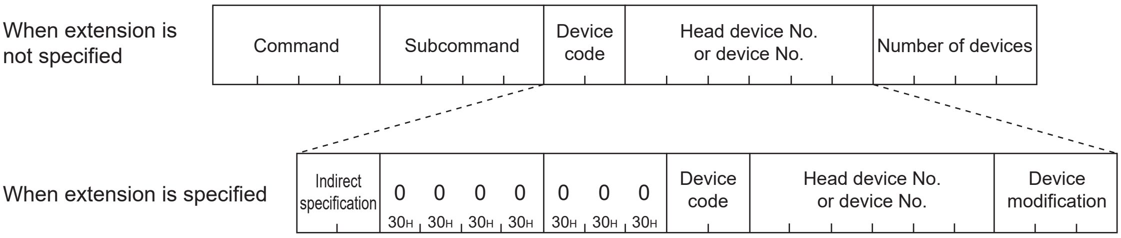
The ADRSET instruction is used on the CPU module side and the address of D100 is stored in D0.



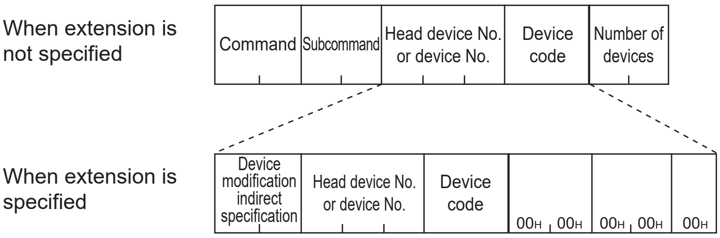
D100 can be indirectly accessed by specifying "@D0" with the request data.

Request data

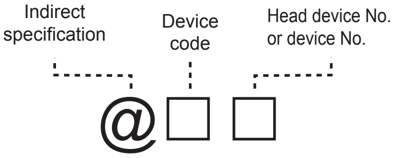
ASCII



Binary



The following shows the indirect specification devices and request data.



Point

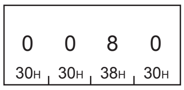
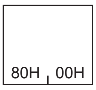
- When specifying 008□ in "subcommand", specify the device with the message format shown above. Message formats when extension is not specified and message formats when extension is specified cannot coexist in the same message.
- The indirect specification and the device modification using index registers cannot be set simultaneously.

■Command

The following commands can be used for accessing.

Function	Command
Random read	0403
Random write	1402

■Subcommand

ASCII code	Binary code
	

■Indirect specification, Device modification

Specify the "@" part of the indirect specification device. Indirect specification can be specified only for word devices.

When communicating data in ASCII code

0	@
30H	40H

When communicating data in binary code

00H	0□H

0H: Without indirect specification
8H: With indirect specification

■Device code (Only word device codes can be specified at indirect specification)

Specify the code of the device to be accessed. (☞ Page 715 Device code list)

■Head device or device No.

Specify the head device or device No. with the same format as the message when extension is not specified.

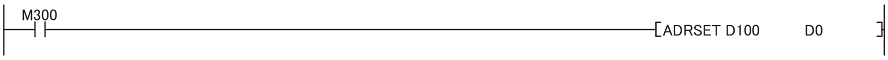
Response data

The same as when extension is not specified.

Communication example

Access to @D0. (Consider @D0 indirect specification of D100.)

At command execution, store the D100 address in D0 with the following programs.



- When communicating data in ASCII code

(Request data)

Subcommand	Indirect specification			Device code	Head device No. or device No.	Device modification
0 0 8 0	0 @	0 0 0 0	0 0 0	D *	0 0 0 0 0 0	0 0 0
30H, 30H, 38H, 30H	30H, 40H	30H, 30H, 30H, 30H	30H, 30H, 30H	44H, 2AH	30H, 30H, 30H, 30H, 30H, 30H	30H, 30H, 30H

- When communicating data in binary code

(Request data)

Subcommand	indirect specification	Head device No. or device No.	Device code			
80H, 00H	00H, 08H	00H, 00H, 00H	A8H	00H, 00H	00H, 00H	00H

Appendix 12 Software Licenses and Copyrights

This section describes the licenses and copyrights of the software used in this product.

MD5 Message-Digest Algorithm

This product includes code that was developed by RSA Data Security, Inc.
Copyright (C) 1991-2, RSA Data Security, Inc.

License to copy and use this software is granted provided that it is identified as the "RSA Data Security, Inc. MD5 Message-Digest Algorithm" in all material mentioning or referencing this software or this function.

License is also granted to make and use derivative works provided that such works are identified as "derived from the RSA Data Security, Inc. MD5 Message-Digest Algorithm" in all material mentioning or referencing the derived work.

RSA Data Security, Inc. makes no representations concerning either the merchantability of this software or the suitability of this software for any particular purpose. It is provided "as is" without express or implied warranty of any kind.

These notices must be retained in any copies of any part of this documentation and/or software.

Appendix 13 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 checked with module diagnosis (CPU diagnosis). Refer to the following manuals for details on diagnosing the module (CPU diagnosis).

📖 MELSEC iQ-F FX5S/FX5UJ/FX5U/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.

FX5S CPU module

Add/Change Function	Supported CPU module firmware version	Supported engineering tool software version	Reference
FX5S CPU module is supported.	From the first	"1.080J" or later	—
MC protocol 1E frame is supported.	From the first	"1.080J" or later	—
The following communication target device (simple CPU communication function) is supported. • MELSEC iQ-F (Ethernet module)	"1.010" or later	"1.095Z" or later	Page 52
MODBUS/TCP-compatible device optional setting (simple CPU communication function) is supported.	"1.010" or later	"1.095Z" or later	Page 70
Port number duplication (simple CPU communication function) in the following devices is supported. • MODBUS/TCP-compatible device • SIEMENS S7 series	"1.010" or later	"1.095Z" or later	Page 69
Communication at request (simple CPU communication function) is supported.	"1.010" or later	"1.095Z" or later	Page 67

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" or later	—
FREQROL-E800/A800 Plus inverters are supported.	From the first	"1.080J" or later	Page 329
User Web page is supported.	"1.020" or later	"1.080J" or later	Page 206
MC protocol 1E frame is supported.	"1.030" or later	"1.085P" or later	Page 90
File transfer function (FTP client) is supported.	"1.030" or later	"1.085P" or later	Page 190
The following communication target device (simple CPU communication function) is supported. • MELSEC iQ-R (Built-in Ethernet) • MELSEC-Q (Built-in Ethernet) • MELSEC-L (Built-in Ethernet) • MELSEC iQ-F (Ethernet module) • MELSEC iQ-L (Built-in Ethernet) • MELSEC-FX3 (Ethernet Block/ Adapter) • OMRON (CJ/CP series) • KEYENCE (KV series) • Panasonic (FP7 series) • Panasonic (FP0H series) • MODBUS/TCP-compatible device • SIEMENS S7 series	"1.030" or later	"1.085P" or later	Page 53
MODBUS/TCP-compatible device optional setting (simple CPU communication function) is supported.	"1.040" or later	"1.090U" or later	Page 70

Add/Change Function	Supported CPU module firmware version	Supported engineering tool software version	Reference
Port number duplication (simple CPU communication function) in the following devices is supported. • MODBUS/TCP-compatible device • SIEMENS S7 series	"1.040" or later	"1.090U" or later	Page 69
Communication at request (simple CPU communication function) is supported.	"1.040" or later	"1.090U" or later	Page 67
The following communication target device (simple CPU communication function) is supported. • MELSEC iQ-F (Ethernet module)	"1.040" or later	"1.090U" or later	Page 53

FX5U/FX5UC CPU module

Add/Change Function	Supported CPU module firmware version	Supported engineering tool software version	Reference
FREQROL-F800 inverter is supported.	From the first	—	Page 329
The file transfer function (FTP server) is supported.	"1.040" or later[*1][*2]	"1.030G" or later[*3]	Page 178
• Automatic detection of connected devices • Communication setting reflection of Ethernet device • Sensor parameter read/write	"1.040" or later	"1.030G" or later	iQ Sensor Solution Reference Manual
Message format: format 1 (X, Y HEX), format 4 (X, Y HEX) are supported.	"1.040" or later	"1.030G" or later	Page 320 Page 692
Communication data code: ASCII code (X, Y HEX) is supported.	"1.040" or later	"1.030G" or later	Page 558
IP filter function is supported.	"1.050" or later	"1.035M" or later	Page 232
Parallel link function is supported.	"1.050" or later	"1.035M" or later	Page 286
MODBUS/TCP communication is supported.	"1.060" or later	"1.040S" or later	Page 499
Time setting function (SNTP client) is supported.	"1.060" or later	"1.040S" or later	Page 203
Web server function is supported.	"1.060" or later	"1.040S" or later	Page 206
Slave station No. (MODBUS address number) that can be set in FX5 master station was changed from 32 to 247.	"1.060" or later	"1.040S" or later	Page 516 Page 525 Page 538
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" or later	—	Page 805
User Web page is supported.	"1.100" or later	"1.047Z" or later	Page 206
Simple CPU communication function is supported.	"1.110" or later	"1.050C" or later	Page 49
FX5-ENET is supported.	"1.110" or later	"1.050C" or later	MELSEC iQ-F FX5 Ethernet Module User's Manual
FX5-ENET/IP is supported.	"1.110" or later	"1.050C" or later	MELSEC iQ-F FX5 EtherNet/IP Module User's Manual
1C frame is supported.	"1.110" or later	"1.050C" or later	Page 686 Page 769
MC protocol 1C frame support is supported.	"1.110" or later	"1.050C" or later	Page 311
FREQROL-E800/A800 Plus inverters are supported.	From the first	"1.080J" or later	Page 329
MC protocol 1E frame is supported.	"1.210" or later	—	Page 90
File transfer function (FTP client) is supported.	"1.210" or later	"1.065T" or later	Page 190

Add/Change Function	Supported CPU module firmware version	Supported engineering tool software version	Reference
The following communication target devices (simple CPU communication function) are supported. • 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" or later	"1.065T" or later	Page 49
File acquisition from the FTP server is supported.	"1.210" or later	"1.065T" or later	Page 190
The following communication target device (simple CPU communication function) is supported. • MELSEC iQ-F (Ethernet module)	"1.270" or later	"1.085P" or later	Page 67
Communication at request (simple CPU communication function) is supported.	"1.270" or later	"1.085P" or later	Page 67
MODBUS/TCP-compatible device optional setting (simple CPU communication function) is supported.	"1.270" or later	"1.085P" or later	Page 70
Port number duplication (simple CPU communication function) in the following devices is supported. • MODBUS/TCP-compatible device	"1.270" or later	"1.085P" or later	Page 69
Port number duplication (simple CPU communication function) in the following devices is supported. • SIEMENS S7 series	"1.280" or later	"1.090U" or later	Page 69

*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" or later	"1.100" or later	"1.075D" or later	Page 31
SLMP communication function	"1.010" or later	"1.100" or later	"1.075D" or later	Page 90
Simple CPU communication function	"1.010" or later	"1.100" or later	"1.075D" or later	Page 49
BACnet communication function	"1.010" or later	"1.100" or later	"1.075D" or later	MELSEC iQ-F FX5 BACnet Reference Manual
MQTT communication function	"1.040" or later	"1.200" or later[*1]	"1.095Z" or later	MELSEC iQ-F FX5 Ethernet Module User's Manual
E-mail function	"1.040" or later	"1.200" or later[*1]	"1.095Z" or later	MELSEC iQ-F FX5 Ethernet Module User's Manual

*1 Supported with FX5-ENET serial No.234**** and above.

■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" or later	"1.003" or later	"1.075D" or later	MELSEC iQ-F FX5 User's Manual (Application)
MELSOFT connection	"1.240" or later	"1.100" or later	"1.075D" or later	Page 31
SLMP communication function	"1.240" or later	"1.100" or later	"1.075D" or later	Page 90
Simple CPU communication function	"1.240" or later	"1.100" or later	"1.075D" or later	Page 49
BACnet communication function	"1.240" or later	"1.100" or later	"1.075D" or later	MELSEC iQ-F FX5 BACnet Reference Manual
MQTT communication function	"1.280" or later	"1.200" or later[*1]	"1.095Z" or later	MELSEC iQ-F FX5 Ethernet Module User's Manual
E-mail function	"1.280" or later	"1.200" or later[*1]	"1.095Z" or later	MELSEC iQ-F FX5 Ethernet Module User's Manual

*1 Supported with FX5-ENET serial No.234**** and above.

CC-Link IE TSN master/local module

Add/Change Function	Supported versions			Reference
	CPU module firmware version	FX5 intelligent function module firmware version	Engineering tool	
SLMP communication function	"1.210" or later	From the first	"1.065T" or later	—

Motion module

Add/Change Function	Supported versions			Reference
	CPU module firmware version	FX5 intelligent function module firmware version	Engineering tool	
SLMP communication function	"1.230" or later	From the first	"1.072A" or later	—

INDEX

REVISIONS

Revision date	Revision	Description
June 2023	SH(NA)-082625ENG-A	First Edition
July 2023	SH(NA)-082625ENG-B	■Added or modified part GENERIC TERMS AND ABBREVIATIONS
October 2023	SH(NA)-082625ENG-C	■Added or modified parts RELEVANT MANUALS, GENERIC TERMS AND ABBREVIATIONS, Chapter 1, Section 2.1, Chapter 3, 5, Section 16.3, 20.7, Chapter 24, Section 28.1, Chapter 30, Section 36.1, 38.2, 42.4, 43.2, 46.1, Appendix 2, 5, 13
April 2024	SH(NA)-082625ENG-D	■Added or modified parts Section 5.4, 8.3, Chapter 18
July 2024	SH(NA)-082625ENG-E	■Added or modified parts GENERIC TERMS AND ABBREVIATIONS, Section 8.1, 38.3, 43.3

Japanese manual number: SH-082624-E

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.

INFORMATION AND SERVICES

For further information and services, please contact your local Mitsubishi Electric sales office or representative.
Visit our website to find our locations worldwide.

MITSUBISHI ELECTRIC Factory Automation Global Website
Locations Worldwide
www.MitsubishiElectric.com/fa/about-us/overseas/

TRADEMARKS

Microsoft and Windows are trademarks of the Microsoft group of companies.
PROFIBUS is a trademark of PROFIBUS Nutzerorganisation e.V.
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.
Unicode is either a registered trademark or a trademark of Unicode, Inc. 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: SH(NA)-082625ENG-E
MODEL: FX5-U-COMMU-E
MODEL CODE: 09R741

MITSUBISHI ELECTRIC CORPORATION

HEAD OFFICE: TOKYO BLDG., 2-7-3, MARUNOUCHI, CHIYODA-KU, TOKYO 100-8310, JAPAN
NAGOYA WORKS: 1-14, YADA-MINAMI 5-CHOME, HIGASHI-KU, NAGOYA 461-8670, JAPAN