MITSUBISHI ELECTRIC

PROGRAMMABLE CONTROLLERS

MELSEC iQ-F series

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

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 medium or slight personal injury or physical damage.

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

It is important to follow all precautions for personal safety.

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

[DESIGN PRECAUTIONS]

⚠WARNING

- Make sure to set up the following safety circuits outside the PLC to ensure safe system operation even during external power supply problems or PLC failure. Otherwise, malfunctions may cause serious accidents.
  (1) Note that when the PLC CPU 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 PLC CPU occurs in an input/output control block, output control may be disabled. External circuits and mechanisms should be designed to ensure safe machine operation in such a case.
- Do not write any data into the "system area" of the buffer memory in the intelligent function module. Executing data writing to the "system area" may cause malfunction of the programmable controller alarm.
- When executing control (data change) to a running other station programmable controller by connecting the external device to the SLMP compatible device, configure interlock circuits in the program of the other station programmable controller to ensure that the entire system operates safely at any time.
  For other controls to a running other station programmable controller (such as program modification or operating status change), read relevant manuals carefully and ensure the safety before the operation. Especially, in the case of a control from an external device to a remote other station programmable controller, immediate action cannot be taken for a problem on the programmable controller due to a communication failure.
  Determine the handling method as a system when communication failure occurs along with configuration of interlock circuit on other station PLC program, by considering external equipment and other station PLC.
- Do not write any data into the "system area" or "write protect area" of the buffer memory in the SLMP compatible device or intelligent function module. Also, do not output (ON) any "use prohibited" signals among the signals which are output to the SLMP compatible device and intelligent function device. Executing data writing to the "system area" or "write protect area", or outputting "use prohibited" signals may cause malfunction of the programmable controller alarm.

[STARTUP AND MAINTENANCE PRECAUTIONS]

⚠WARNING

- 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

- Read relevant manuals carefully and ensure the safety before performing online operations (operation status change) with peripheral devices connected to the running SLMP compatible device or CPU modules of other stations. Improper operation may damage machines or cause accidents.

INTRODUCTION

This manual explains the specifications and settings related to the SLMP function of the MELSEC iQ-F Series.
It should be read and understood before attempting to install or use the module.
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 to 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

RELEVANT MANUALS

User's manuals for the applicable modules

Manual name <manual number>	Description
MELSEC iQ-F FX5 User's Manual (Startup) <JY997D58201>	Performance specifications, procedures before operation, and troubleshooting of the CPU module.
MELSEC iQ-F FX5U User's Manual (Hardware) <JY997D55301>	Describes the details of hardware of the FX5U CPU module, including input/output specifications, wiring, installation, and maintenance.
MELSEC iQ-F FX5UC User's Manual (Hardware) <JY997D61401>	Describes the details of hardware of the FX5UC CPU module, including input/output specifications, wiring, installation, and maintenance.
MELSEC iQ-F FX5 User's Manual (Application) <JY997D55401>	Describes 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 specifications of ladders, ST, FBD/LD, and other programs and labels.
MELSEC iQ-F FX5 Programming Manual (Instructions, Standard Functions/Function Blocks) <JY997D55801>	Describes specifications of instructions and functions that can be used in programs.
MELSEC iQ-F FX5 User's Manual (Serial Communication) <JY997D55901>	Describes N:N network, MELSEC Communication protocol, inverter communication, non-protocol communication, and predefined protocol support.
MELSEC iQ-F FX5 User's Manual (MODBUS Communication) <JY997D56101>	Describes MODBUS serial communication.
MELSEC iQ-F FX5 User's Manual (Ethernet Communication) <JY997D56201>	Describes the functions of the built-in Ethernet port communication function.
MELSEC iQ-F FX5 User's Manual (SLMP) <JY997D56001> (This manual)	Explains methods for the device that is communicating with the CPU module by SLMP to read and write the data of the CPU module.
MELSEC iQ-F FX5 User's Manual (Positioning Control) <JY997D56301>	Describes the built-in positioning function.
MELSEC iQ-F FX5 User's Manual (Analog Control) <JY997D60501>	Describes the analog function.
GX Works3 Operating Manual <SH-081215ENG>	System configuration, parameter settings, and online operations of GX Works3.

TERMS

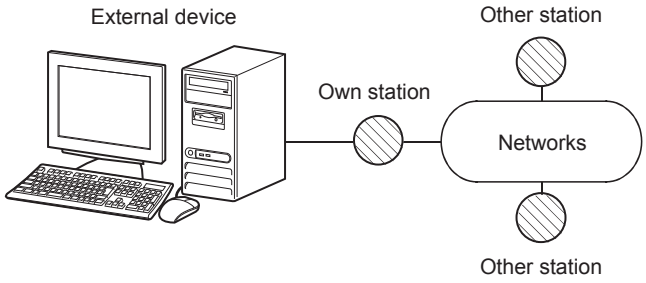
Unless otherwise specified, this manual uses the following terms.

- ☐ indicates a variable part to collectively call multiple models or versions.

(Example) FX5U-32MR/ES, FX5U-32MT/ES ⇨ FX5U-32M☐/ES

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

Terms	Description
■Devices	
FX5	Generic term for FX5U and FX5UC PLCs
FX3	Generic term for FX3S, FX3G, FX3GC, FX3U, and FX3UC PLCs
FX5 CPU module	Generic term for FX5U CPU module and FX5UC CPU module
FX5U CPU module	Generic term for FX5U-32MR/ES, FX5U-32MT/ES, FX5U-32MT/ESS, FX5U-64MR/ES, FX5U-64MT/ES, FX5U-64MT/ESS, FX5U-80MR/ES, FX5U-80MT/ES, and FX5U-80MT/ESS
FX5UC CPU module	Generic term for FX5UC-32MT/D and FX5UC-32MT/DSS
Extension module	Generic term for FX5 extension modules and FX3 function modules
• FX5 extension module	Generic term for I/O modules, FX5 extension power supply module, and FX5 intelligent function module
• FX3 extension module	Generic term for FX3 extension power supply module and FX3 special function blocks
Extension module (extension cable type)	Input modules (extension cable type), Output modules (extension cable type), Bus conversion module (extension cable type), and Intelligent function modules
Extension module (extension connector type)	Input modules (extension connector type), Output modules (extension connector type), Input/output modules, Bus conversion module (extension connector type), and Connector conversion module (extension connector type)
I/O module	Generic term for input modules, output modules, Input/output modules, and powered input/output modules
Input module	Generic term for Input modules (extension cable type) and Input modules (extension connector type)
• Input module (extension cable type)	Generic term for FX5-8EX/ES and FX5-16EX/ES
• Input module (extension connector type)	Generic term for FX5-C32EX/D and FX5-C32EX/DS
Output module	Generic term for output modules (extension cable type) and output modules (extension connector type)
• Output module (extension cable type)	Generic term for FX5-8EYR/ES, FX5-8EYT/ES, FX5-8EYT/ESS, FX5-16EYR/ES, FX5-16EYT/ES, and FX5-16EYT/ESS
• Output module (extension connector type)	Generic term for FX5-C32EYT/D and FX5-C32EYT/DSS
Input/output modules	Generic term for FX5-C32ET/D and FX5-C32ET/DSS
Powered input/output module	Generic term for FX5-32ER/ES, FX5-32ET/ES, and FX5-32ET/ESS
Extension power supply module	Generic term for FX5 extension power supply module and FX3 extension power supply module
• FX5 extension power supply module	Different name for FX5-1PSU-5V
• FX3 extension power supply module	Different name for FX3U-1PSU-5V
Intelligent module	The abbreviation for intelligent function modules
Intelligent function module	Generic term for FX5 intelligent function modules and FX3 intelligent function modules
• FX5 intelligent function module	Generic term for FX5 intelligent function modules
• FX3 intelligent function module	Generic term for FX3 special function blocks
Simple motion module	Different name for FX5-40SSC-S
Expansion board	Generic term for board for FX5U CPU module
• Communication board	Generic term for FX5-232-BD, FX5-485-BD, and FX5-422-BD-GOT
Expansion adapter	Generic term for adapter for FX5 CPU module
• Communication adapter	Generic term for FX5-232ADP and FX5-485ADP
• Analog adapter	Generic term for FX5-4AD-ADP and FX5-4DA-ADP
Bus conversion module	Generic term for Bus conversion module (extension cable type) and Bus conversion module (extension connector type)
• Bus conversion module (extension cable type)	Different name for FX5-CNV-BUS
• Bus conversion module (extension connector type)	Different name for FX5-CNV-BUSC
Battery	Different name for FX3U-32BL
Peripheral device	Generic term for engineering tools and GOTs
GOT	Generic term for Mitsubishi Graphic Operation Terminal GOT1000 and GOT2000 series

Terms	Description
■Software packages	
Engineering tool	The product name of the software package for the MELSEC programmable controllers
GX Works3	The product name of the software package, SWnDND-GXW3, for the MELSEC programmable controllers (The 'n' represents a version.)
■Manuals	
User's manual	Generic term for separate manuals
• User's manual (Startup)	Abbreviation of MELSEC iQ-F FX5 User's Manual (Startup)
• FX5 User's manual (Hardware)	Generic term for MELSEC iQ-F FX5U User's Manual (Hardware) and MELSEC iQ-F FX5UC User's Manual (Hardware)
• FX5U User's manual (Hardware)	Abbreviation of MELSEC iQ-F FX5U User's Manual (Hardware)
• FX5UC User's manual (Hardware)	Abbreviation of MELSEC iQ-F FX5UC User's Manual (Hardware)
• User's manual (Application)	Abbreviation of MELSEC iQ-F FX5 User's Manual (Application)
Programming manual (Program Design)	Abbreviation of MELSEC iQ-F FX5 Programming Manual (Program Design)
Programming manual (Instructions, Standard Functions/Function Blocks)	Abbreviation of MELSEC iQ-F FX5 Programming Manual (Instructions, Standard Functions/Function Blocks)
Communication manual	Generic term for MELSEC iQ-F FX5 User's Manual (Serial Communication), MELSEC iQ-F FX5 User's Manual (MODBUS Communication), MELSEC iQ-F FX5 User's Manual (Ethernet Communication), and MELSEC iQ-F FX5 User's Manual (SLMP)
• Serial communication manual	Abbreviation of MELSEC iQ-F FX5 User's Manual (Serial Communication)
• MODBUS communication manual	Abbreviation of MELSEC iQ-F FX5 User's Manual (MODBUS Communication)
• Ethernet communication manual	Abbreviation of MELSEC iQ-F FX5 User's Manual (Ethernet Communication)
• SLMP manual	Abbreviation of MELSEC iQ-F FX5 User's Manual (SLMP)
Positioning manual	Abbreviation of MELSEC iQ-F FX5 User's Manual (Positioning Control)
Analog manual	Abbreviation of MELSEC iQ-F FX5 User's Manual (Analog Control)
■Communication-related	
Built-in RS-485 port	Built-in RS-485 port of the CPU module.
Serial port	Generic term for the four ports consisting of the FX5 Series built-in RS-485 port (CH1), communication board (CH2), communication adapter 1 (CH3), and communication adapter 2 (CH4).
SLMP	The abbreviation for Seamless Message Protocol. A protocol for accessing SLMP-compatible devices and PLCs that are connected to SLMP-compatible devices from external devices.
SLMP-compatible device	Generic term for devices that can receive SLMP messages.
MC protocol	The abbreviation of the MELSEC communication protocol. A protocol for accessing MC protocol-compatible devices and PLCs that are connected to MC protocol-compatible devices from external devices.
MC protocol-compatible device	Generic term for devices that can receive MC protocol messages.
External device	Generic term for devices of communication target (such as personal computer, HMI)
Own station	Own station indicates the station directly connected to external device. Other station indicates a station connected to the own station on the network. 
Other station	
Relay station	A station that includes two or more network modules. Transient transmission is performed through this station to stations on other networks.
Module access device	A generic term for the module access device of the MELSEC iQ-R series/MELSEC iQ-F series and intelligent function module device of the MELSEC-Q/L series
Buffer memory	Memory areas of Intelligent function modules and SLMP-compatible devices for storing setting values and monitor values.

1 OUTLINE

This manual describes the method for reading or writing data in a CPU module with the data communication function of the external equipment using SLMP.
When transferring data using SLMP, always refer to ☞ Page 11 SLMP DATA COMMUNICATION.

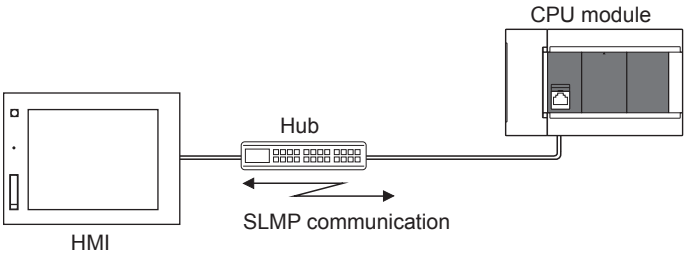
1.1 Outline of SLMP

SLMP is a protocol used for access from a CPU 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.

Point

The message format of 3E frame of SLMP is the same as that of the QnA compatible 3E frame of MC protocol.
Therefore, external devices used with MC protocol can be connected to an SLMP compatible device directly.
For details on MC protocol, refer to the following manual.
• 📖MELSEC-Q/L MELSEC Communication Protocol Reference Manual

Device data in a CPU 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 a CPU module by a personal computer or an HMI.
In addition, external illegal access can be prevented by the remote password function.



The following shows the flow for starting SLMP communication.

1. Connect cables and external devices.

Configure the connection for the SLMP communication.

2. Set parameters.

Set parameters with engineering tool.

3. Write the set parameters to the CPU module.

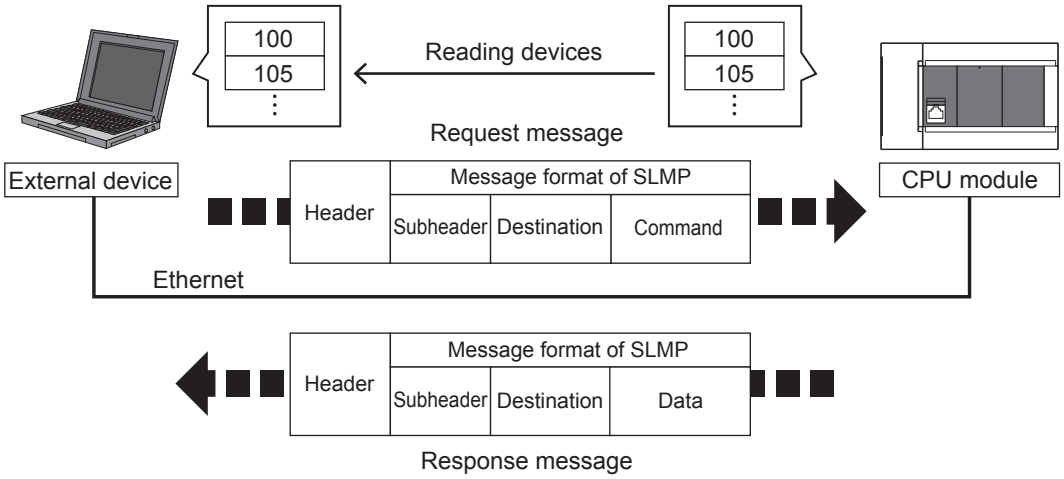
Write set parameters to a CPU module. Validate the parameters by turning off to on or resetting the system.

1.2 Features of SLMP

System monitoring from an external device (such as personal computer, HMI)

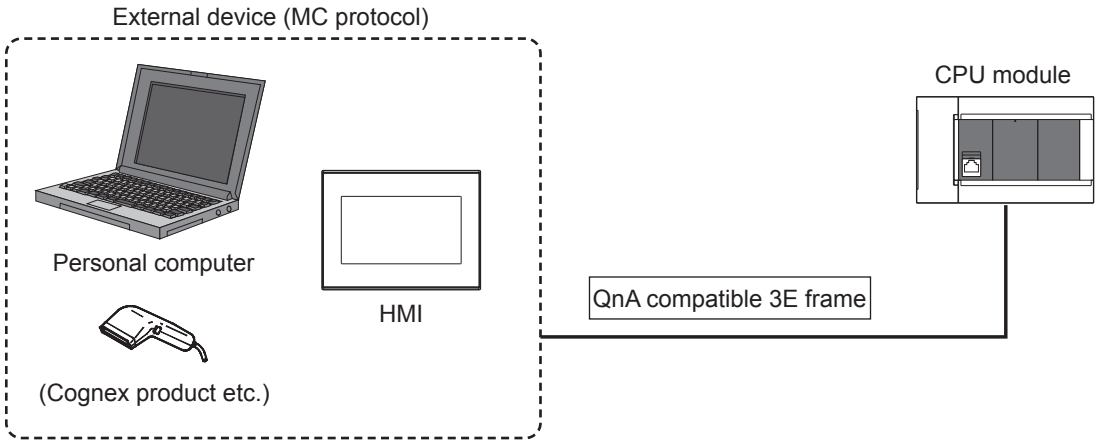
An external device can send a request message in SLMP message format to an CPU module to enable device read, allowing system monitoring.

Using SLMP allows not only device data reading but also device data writing and resetting an CPU module.



Connecting an external device used with MC protocol

An external device that uses the QnA compatible 3E frame of MC protocol can be connected to an CPU module directly.



2 SLMP DATA COMMUNICATION

This chapter describes the SLMP data communication by which the external equipment reads or writes data to a CPU module.

2.1 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 a CPU module with SLMP.
When the external equipment accesses a CPU 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	binary code	Page 15 MESSAGE FORMAT

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 CPU module from the external equipment.
- The frame enables access to the device of the MELSEC iQ-R/L/Q/A Series PLC CPUs via the CC-Link IE controller network, CC-Link IE field network, or Ethernet.

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.

2.2 Allowable Access Range of Each Data Communication Frame

The following shows the frame and access range of a message used in SLMP.

SLMP frame

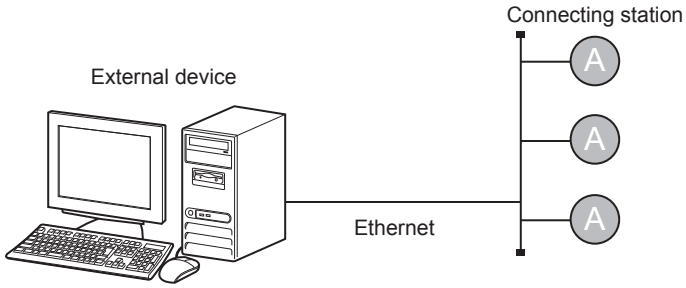
Frame	Type of the network which connects the external device with the connecting stations	Reference
Ethernet communication frame (3E frame)	Ethernet	Page 15 MESSAGE FORMAT

Access range

Ethernet communication frame

■When the external device is connected directly with the CPU module via Ethernet

In the following system configuration, communication with the CPU module is possible using the Ethernet communication frame from the external device.



Assigned symbol	Description
A	Station directly connected to the external device

2.3 Concept of Control Procedure of SLMP

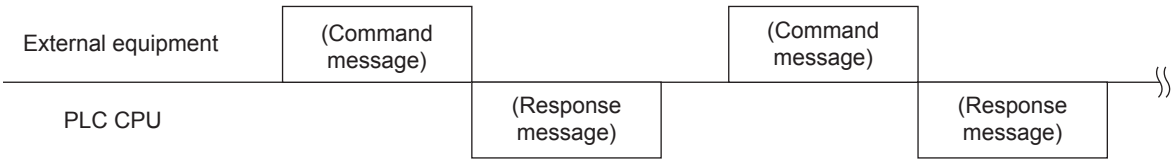
This section describes the concept of the procedure (control procedure) when the external equipment accesses a CPU module with SLMP.

Sending a command message

Data communication using SLMP communication is executed in half-duplex communication.

To access the CPU module, send the next command message after receiving a response message for the preceding command message from the CPU 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.

■When a response message or all messages cannot be received

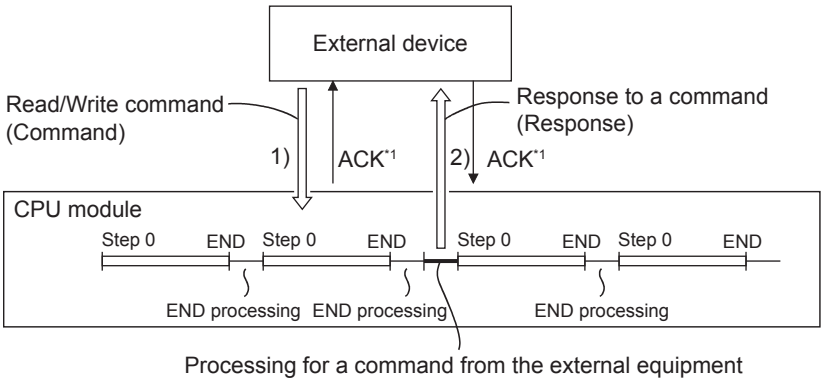
Resend a command message after the monitoring time of the response monitoring timer elapses.

Change the set value of the monitoring time as needed.

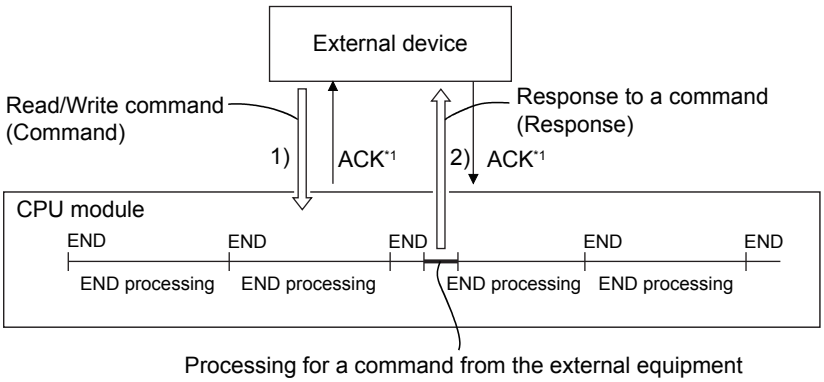
2.4 Access Timing of the CPU Module Side

The following shows the access timing of the CPU module side when the CPU module is accessed from the external equipment using the built-in Ethernet port.

• RUN



• STOP



*1 ACK shown in the figure is a response which is sent or received between the CPU module and external equipment (a response for receiving a massage) when the CPU 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 built-in Ethernet port, an ACK response is not sent.

1. To send a read request or a write request to the CPU module side from the external equipment, a command message is sent.

2. The CPU 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 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 equipment, the processing requested from the external device may be on hold until several END processings take place depending on the request timing.

2.5 Transfer Time

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

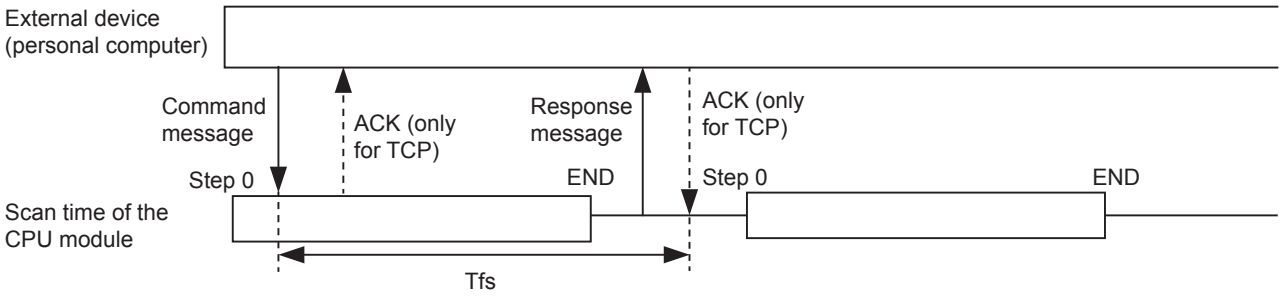
Tfs: Time from when the request data of a personal computer is received until the CPU module completes the processing (Unit: ms) *1

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

3 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 built-in Ethernet port.

Frame type	Built-in Ethernet port	Remark
3E frame	Communicable	The message format is the same as the QnA compatible 3E frame

3.1 Message Format

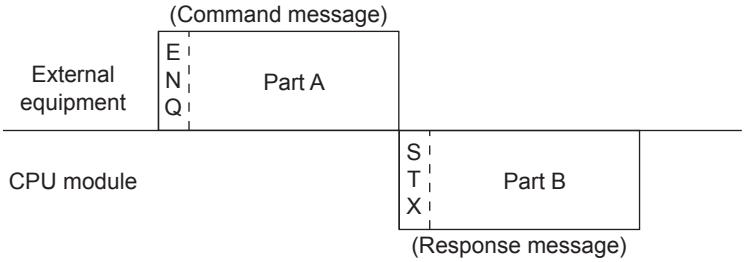
This section describes the message format for each command when performing the data communication using the 3E frame.

How to understand command descriptions

This section describes how to understand message diagrams in each command description shown in ☞ Page 43 Device Access and after.

The following example shows how to understand message diagrams in command descriptions for each control procedure when communicating with the built-in Ethernet port.

When data is read from a CPU module by external equipment

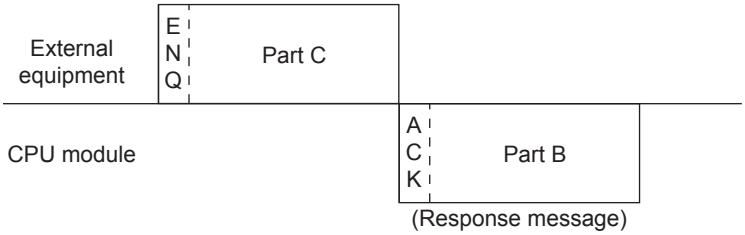


- Part A indicates transfer from the external equipment to the CPU module.
- Part B indicates transfer from the CPU module to the external equipment.
- Create a program in the external equipment so that each data is transferred sequentially from the left to the right.

Ex.

In part A, data is sent sequentially starting from ENQ. In part B, data is received sequentially starting from STX.

When data is written from external equipment to a CPU module



- Part C indicates transfer from the external equipment to the CPU module.
- Part B indicates transfer from the CPU module to the external equipment.
- Create a program in the external equipment so that each data is transferred sequentially from the left to the right.

Ex.

In part C, data is sent sequentially starting from ENQ. In part B, data is received sequentially starting from ACK.

Point

After receiving a command message from the external equipment, the CPU module completes the processing for part A and part C in the message and sends a response message part B, then starts the receiving wait status (neutral status).

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 for communicating between the built-in Ethernet port and the external device 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	Reserve	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 CPU 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 CPU module to external equipment (response message) by the user because the header is added by the CPU automatically.

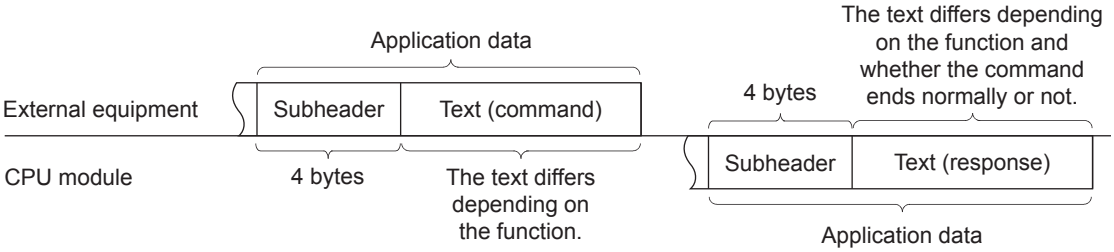
Application data

Application data is divided into subheader and text.

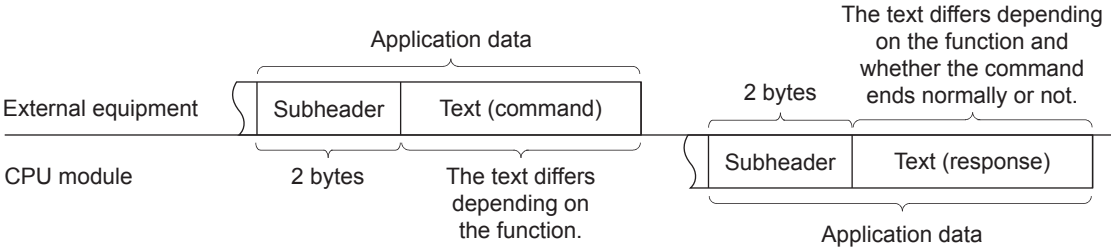
The subheader indicates whether a message is the command message or the response message. (Refer to ☞ Page 17 Subheader configuration.)

Text is the request data (command) and the response date (response) in each function. (For details on each function, refer to ☞ Page 39 List of Commands and Functions or and after.)

■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 CPU 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 of this section are items common to all commands and correspond to the * portion of the message explanation diagrams indicated in ☞ Page 72 Device Write Block or after in this chapter.

For the data contents and the data specification method of the ☐ (Thick line) part, refer to ☞ Page 22 Application data specification items.

Point

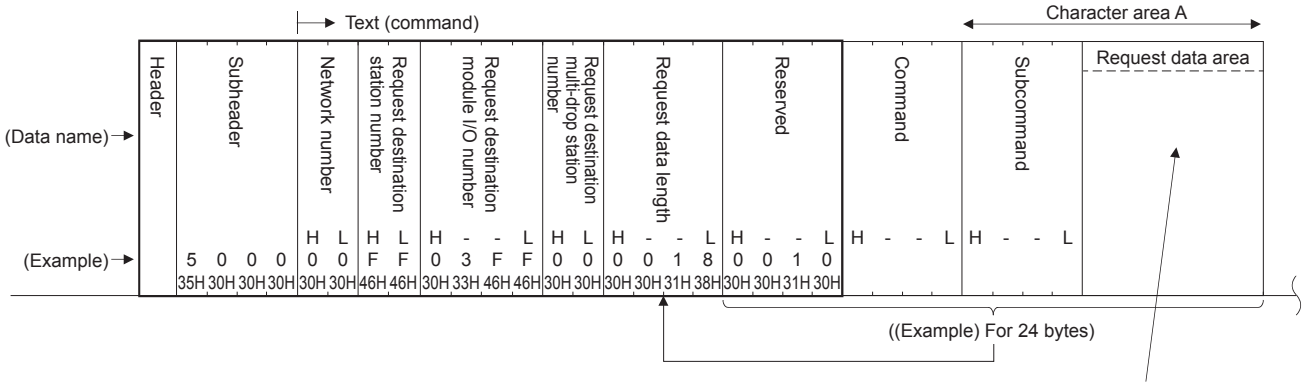
Data code (ASCII/binary) to be used when communicating, it is determined by the parameters of the GX Works3.

[Module Parameter] → [Ethernet Port] → [Communication Data Code]

■When communicating data in ASCII code

• When data is read from a CPU module by external equipment

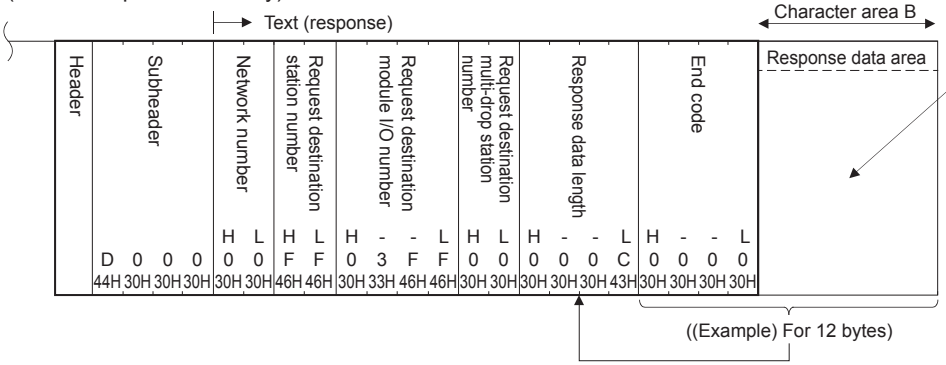
External equipment → CPU module (command message)



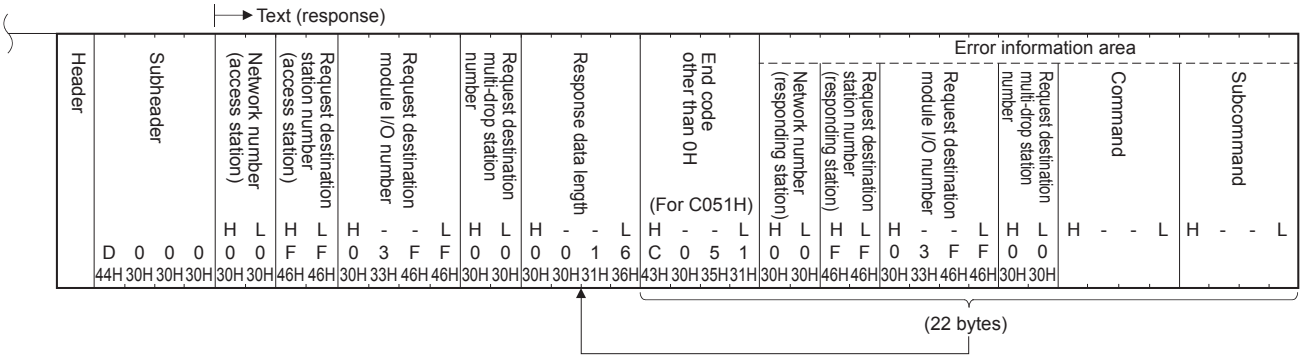
The order of data items differs depending on the command or subcommand. For details, refer to the description on command details in Section 4.2 or later.

CPU module → External equipment (response message)

(When completed normally)

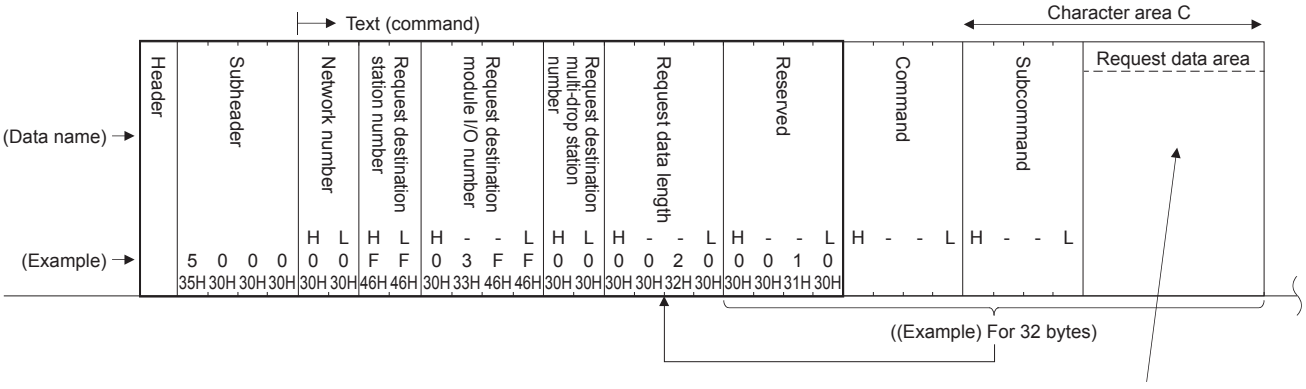


(When completed with error)



• When data is written from external equipment to a CPU module

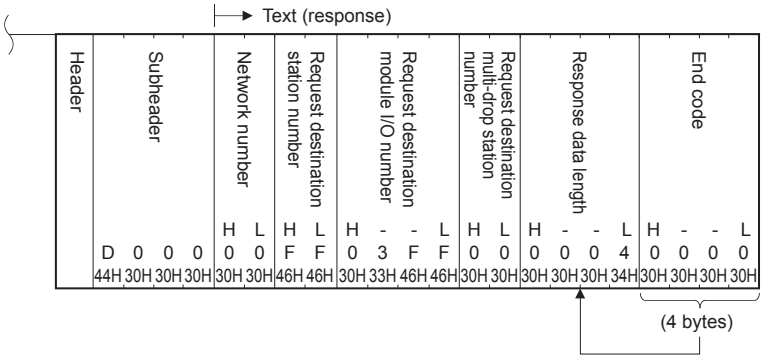
External equipment → CPU module (command message)



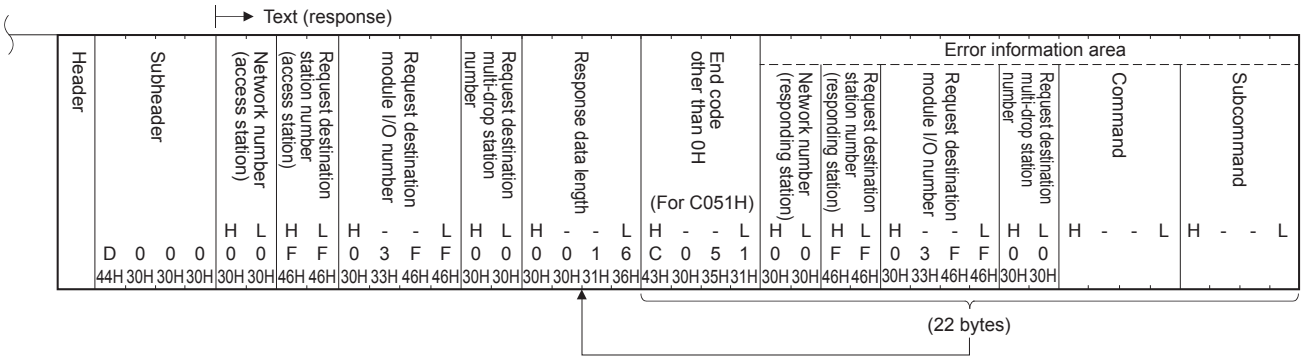
The order of data items differs depending on the command or subcommand. For details, refer to the description on command details in Section 4.2 or later.

CPU module → External equipment (response message)

(When completed normally)



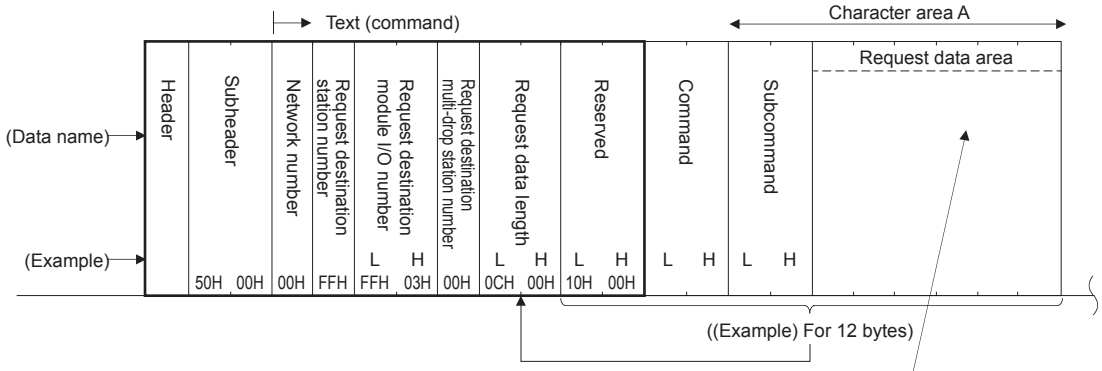
(When completed with error)



■When communicating data in binary code

• When data is read from a CPU module by external equipment

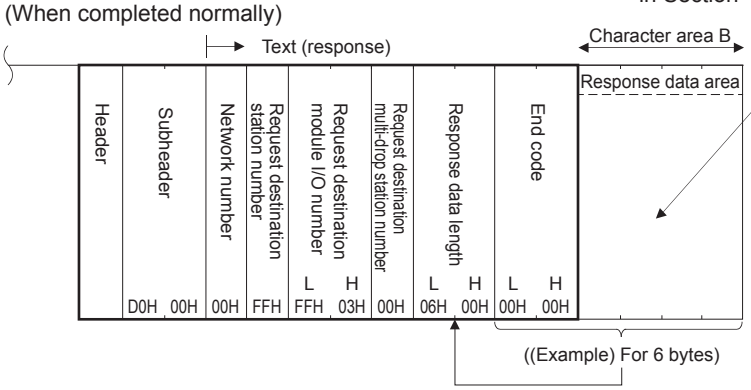
External equipment → CPU module (command message)



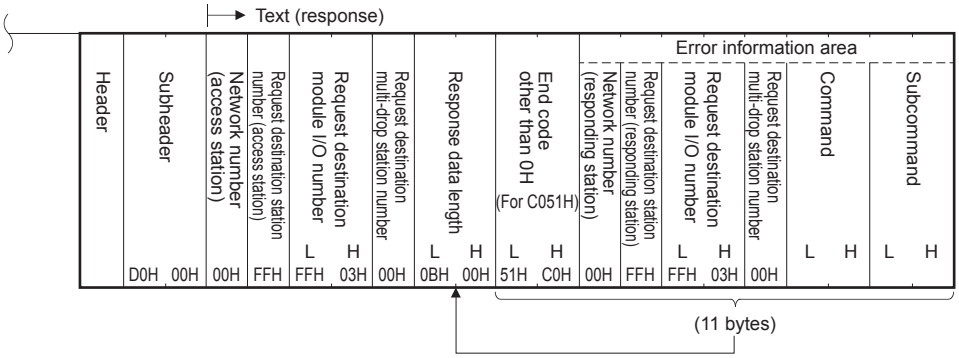
The order of data items differs depending on the command or subcommand. For details, refer to the description on command details in Section 4.2 or later.

CPU module → External equipment (response message)

(When completed normally)

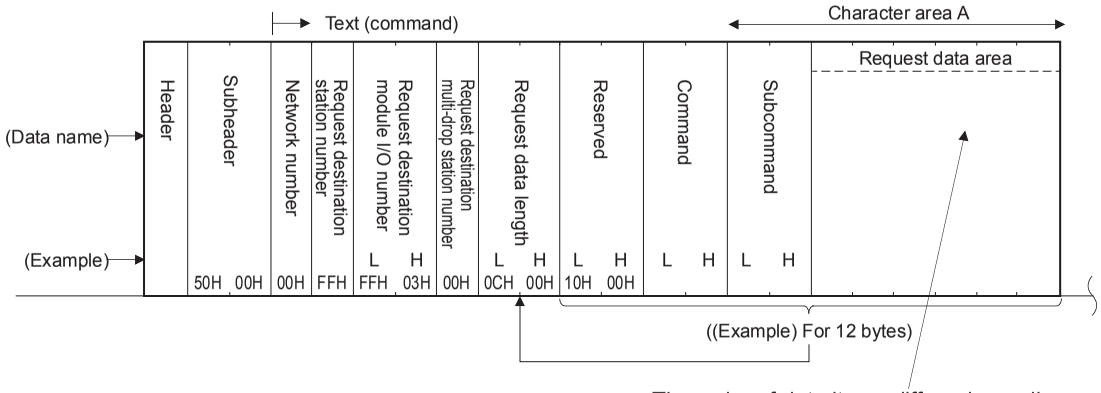


(When completed with error)



• When data is written from external equipment to a CPU module

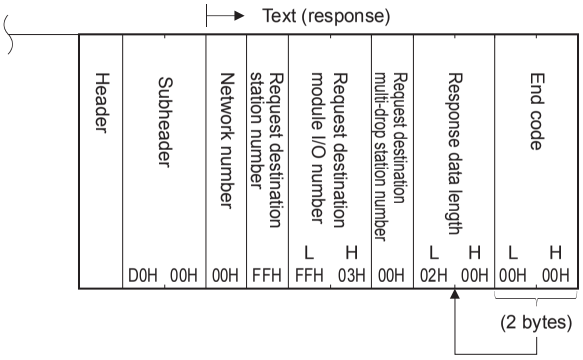
External equipment → CPU module (command message)



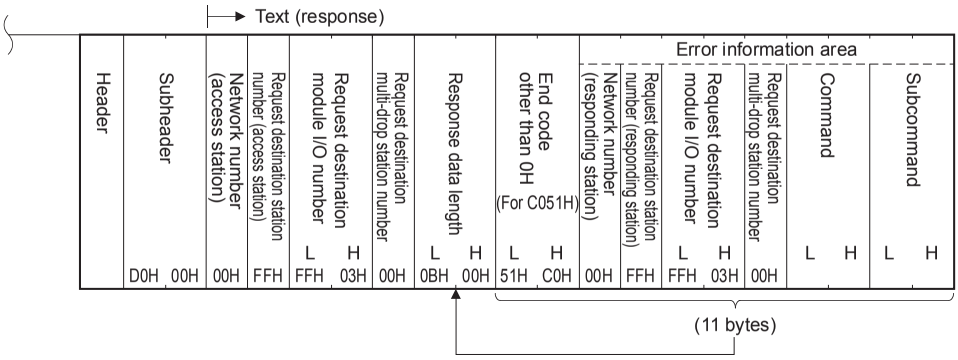
The order of data items differs depending on the command or subcommand. For details, refer to the description on command details in Section 4.2 or later.

CPU module → External equipment (response message)

(When completed normally)



(When completed with error)



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	Reserve	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

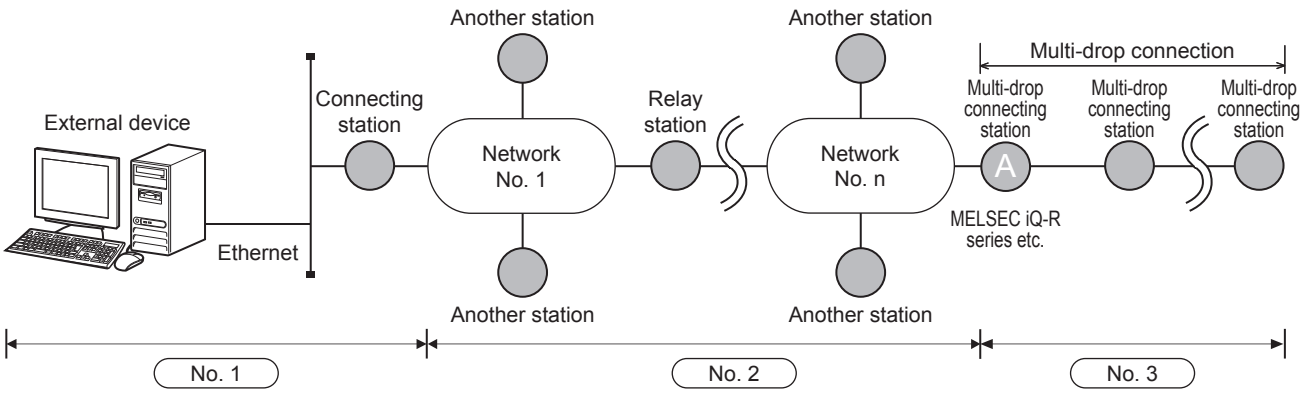
Specify the request destination network number and request destination station number to be used as an access destination in hexadecimal.

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 FX5CPU.



Ex.

When specifying 26 (1AH) as the station number n and 16 (10H) as the station number of station A

ASCII code

1	A	1	0
31H	41H	31H	30H

Network number Station number

Binary code

1AH	10H

Network number Station number

Precautions

The stations of network number 240 to 255 cannot be accessed.

FX5CPU cannot perform multi-drop connection.

FX5CPU 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	Reserve	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

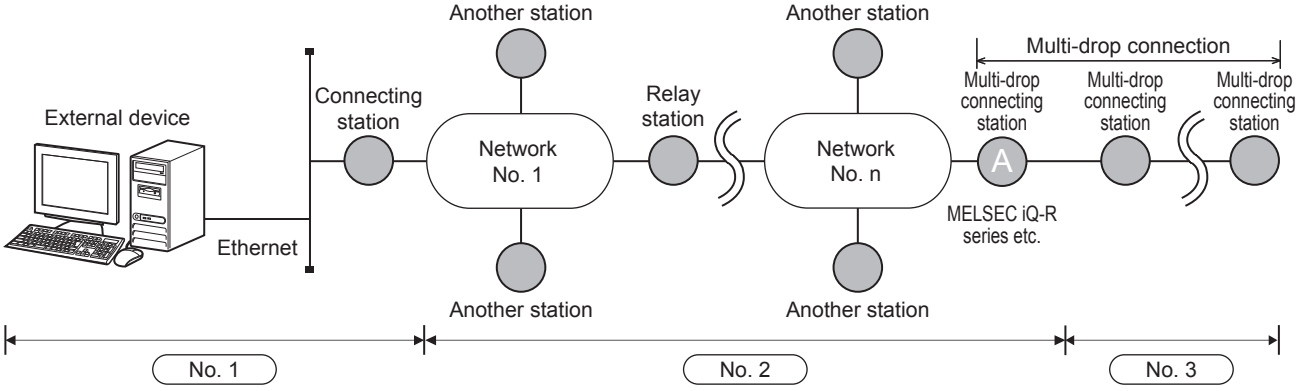
Select the module number of the access destination from the table below.

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 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 FX5CPU cannot perform multi-drop connection.
FX5CPU cannot perform connection via network.

*2 Please use specification No.1 to access FX5CPU.



Ex.

When specifying the default processor (0005H) as the request destination module I/O number

ASCII code

0	0	0	5
30H	30H	30H	35H

Request destination module I/O No.

Binary code

05H	00H

Request destination module I/O No.

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	Reserve	Command	Subcommand	Request data

■Response message

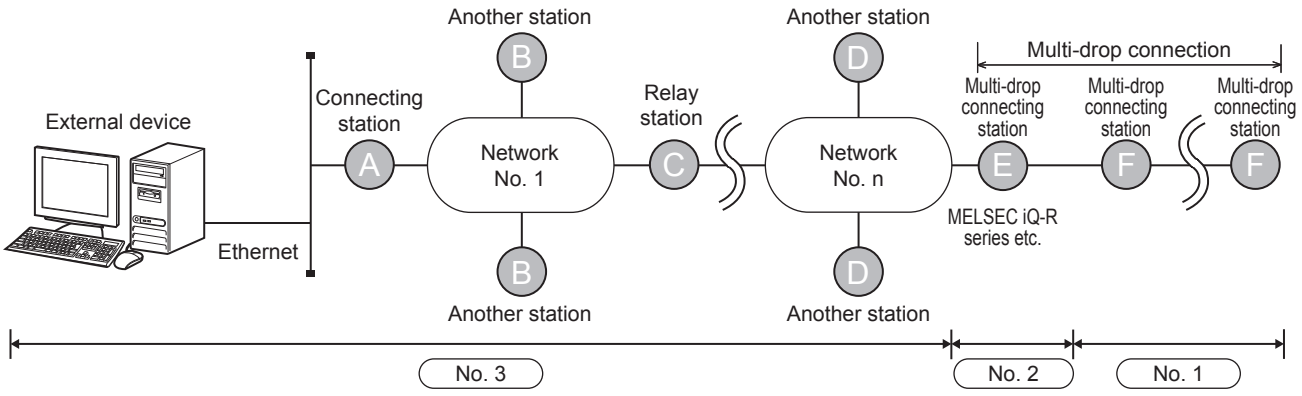
Header	Application data							
	Subheader	Request destination network No.	Request destination station No.	Request destination module I/O No.	Request destination multidrop station No.	Response data length	End code	Response data

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 FX5CPU.



Ex.

When specifying 00H as the requested multi-drop station number

ASCII code

0	0
30H	30H

Requested multi-drop station number

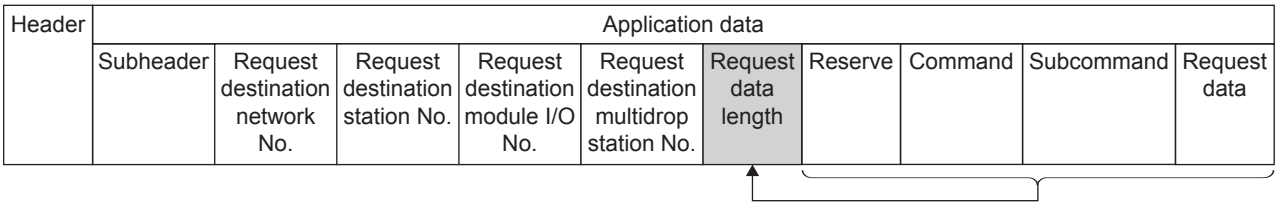
Binary code

00H

Requested multi-drop station number

Request data length

■Request message



Specify the total data size from the reserve to the request data in hexadecimal. (Unit: byte)

Ex.

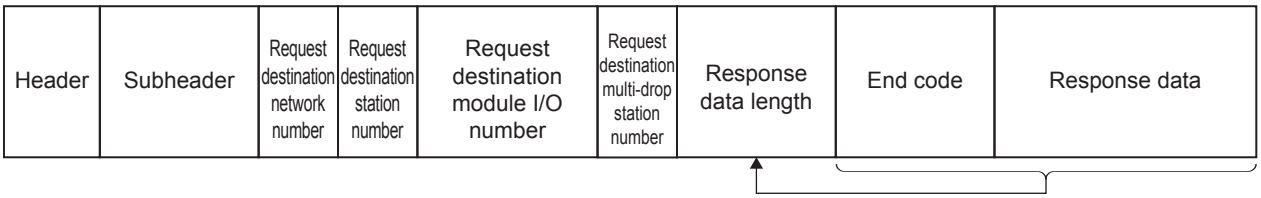
When the request data length is 24 (18H) bytes

Response data length

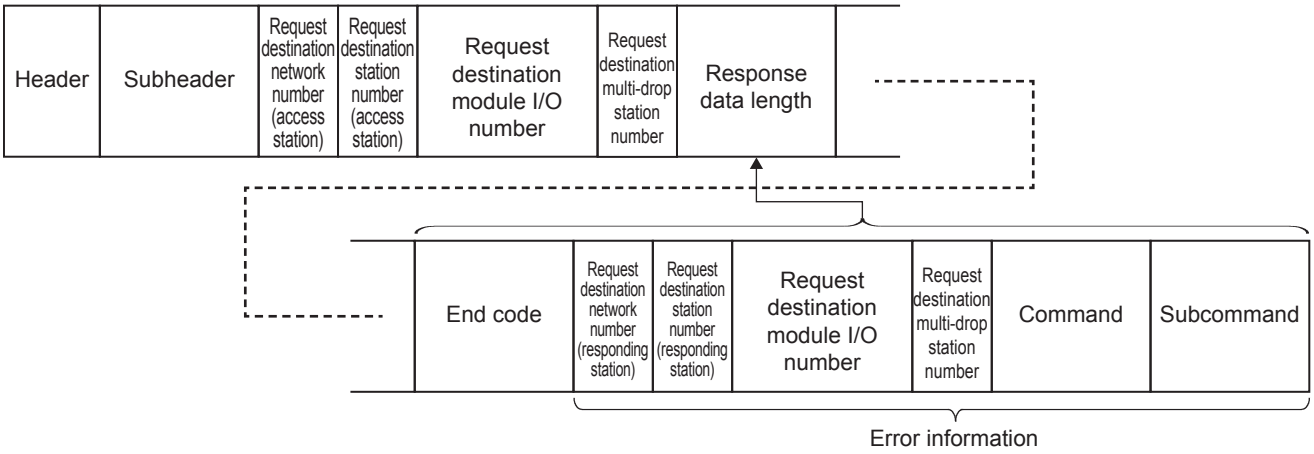
■Response message

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)



Reserved

■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	Reserve	Command	Subcommand	Request data

Setting range:

- Set 0000H (0).

Ex.

ASCII code

0	0	0	0
30H	30H	30H	30H

Binary code

00H	00H

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, "0" 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 manuals of the SLMP compatible device of the response station.)

Ex.

When completed

ASCII code

0	0	0	0
30H	30H	30H	30H

Binary code

00H	00H

When failed (for 0400H)

ASCII code

0	4	0	0
30H	34H	30H	30H

Binary code

00H	04H

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	Reserve	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 of the request data, refer to ☞ Page 43 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 of the response data, refer to ☞ Page 43 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 CPU module by using each command.
The transfer data explained below is handled as the character area B 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	Reserve	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.

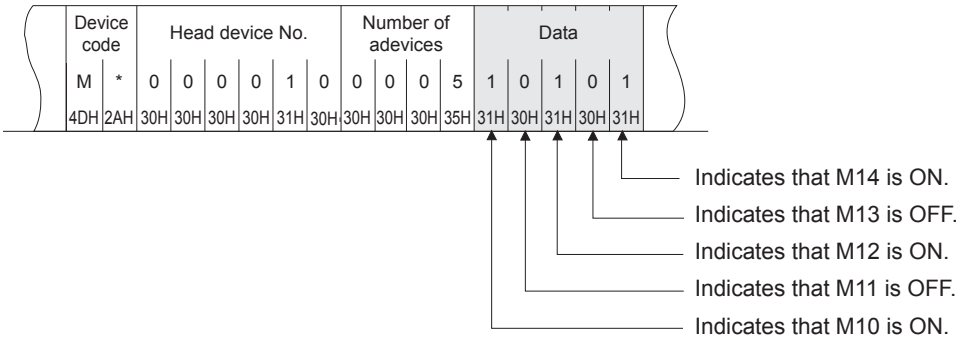
The transfer data in each case is described below.

• In 1-bit (1-point) units

When the bit device memory is handled in 1-bit (1-point) units, a specified number of devices starting from the specified start device are expressed in turn from the left end in "1 (31H)" for the on status or "0 (30H)" for the off status.

Ex.

When indicating the on/off status of five devices starting from M10

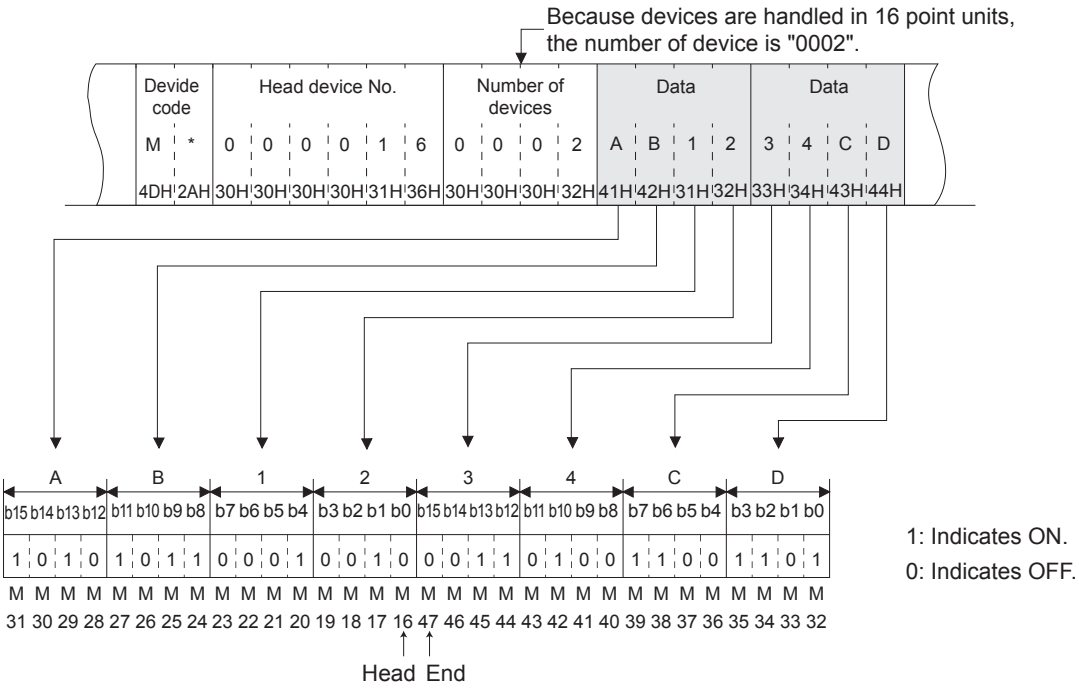


• In 1-word (16-point) units

When the bit device memory is handled in 1-word units, one word is expressed in 4-bit units in turn from the most significant bit in hexadecimal.

Ex.

When indicating the on/off status of 32 devices starting from M16

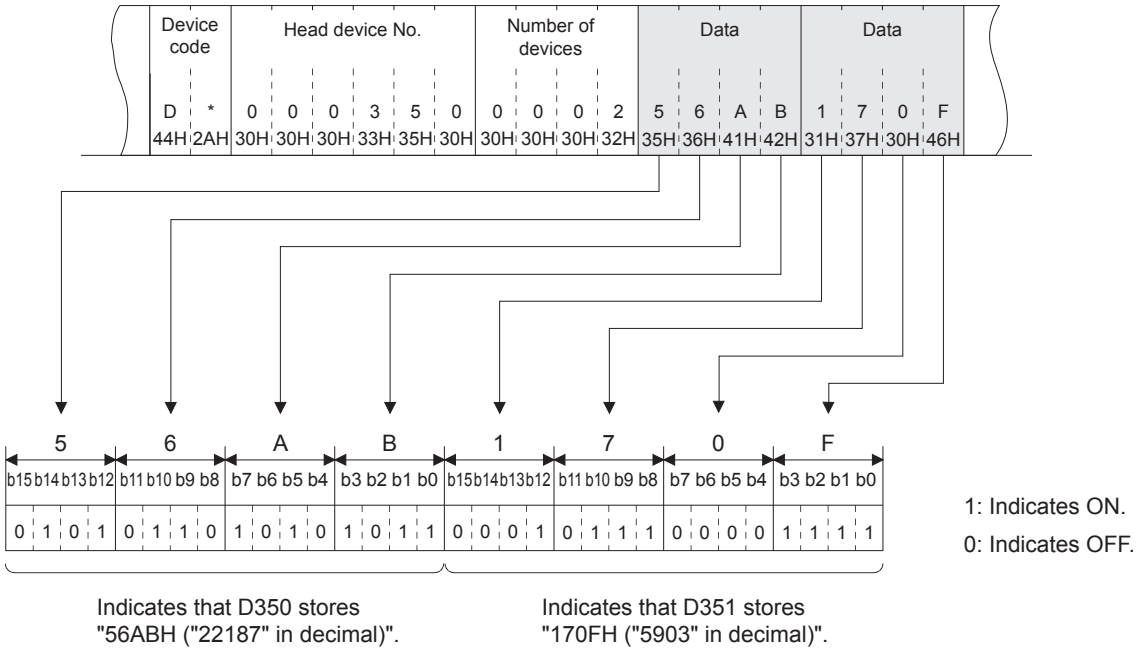


■When word device memory is read or written

In the case of word device memory, one word is expressed in 4-bit units in turn from the most significant bit in hexadecimal.

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 to 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 to 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.

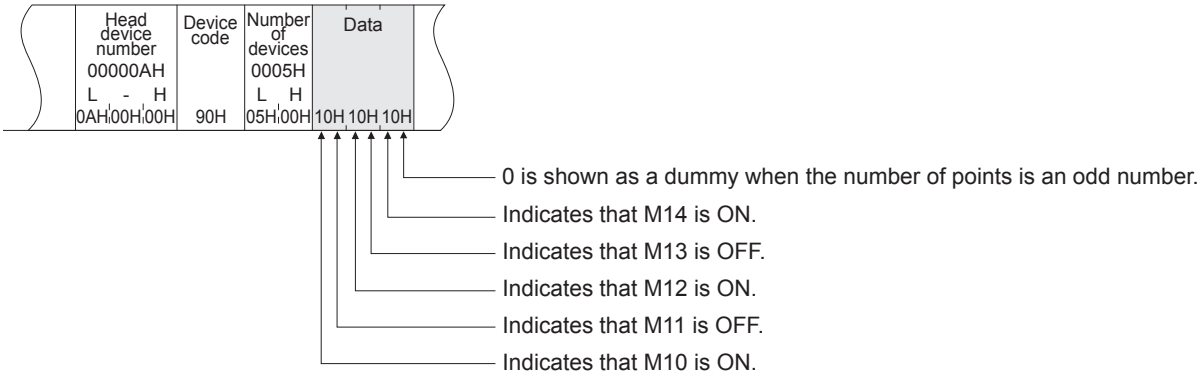
The transfer data in each case is described below.

• In 1-bit (1-point) units

When the bit device memory is handled in 1-bit (1-point) units, one point is specified by 4-bits and a specified number of devices starting from the specified start device are expressed in turn from the most significant bit as "1" for the on status or "0" for the off status.

Ex.

When indicating the on/off status of five devices starting from M10

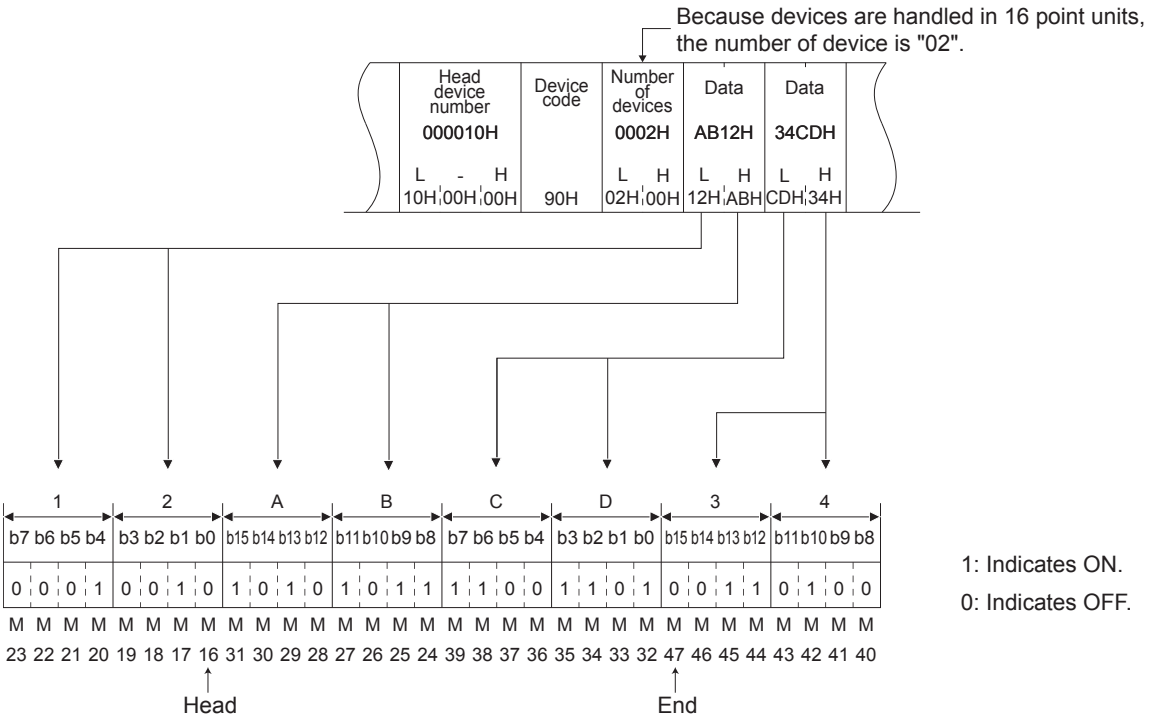


• In 1-word (16-point) units

When the bit device memory is handled in 1-word (16-point) units, one point is specified by 1-bit and a specified number of devices starting from specified start device are expressed in 16-point units in turn from Low byte (L: bit 0 to 7) to High byte (H: bit 8 to 15).

Ex.

When indicating the on/off status of 32 devices starting from M16

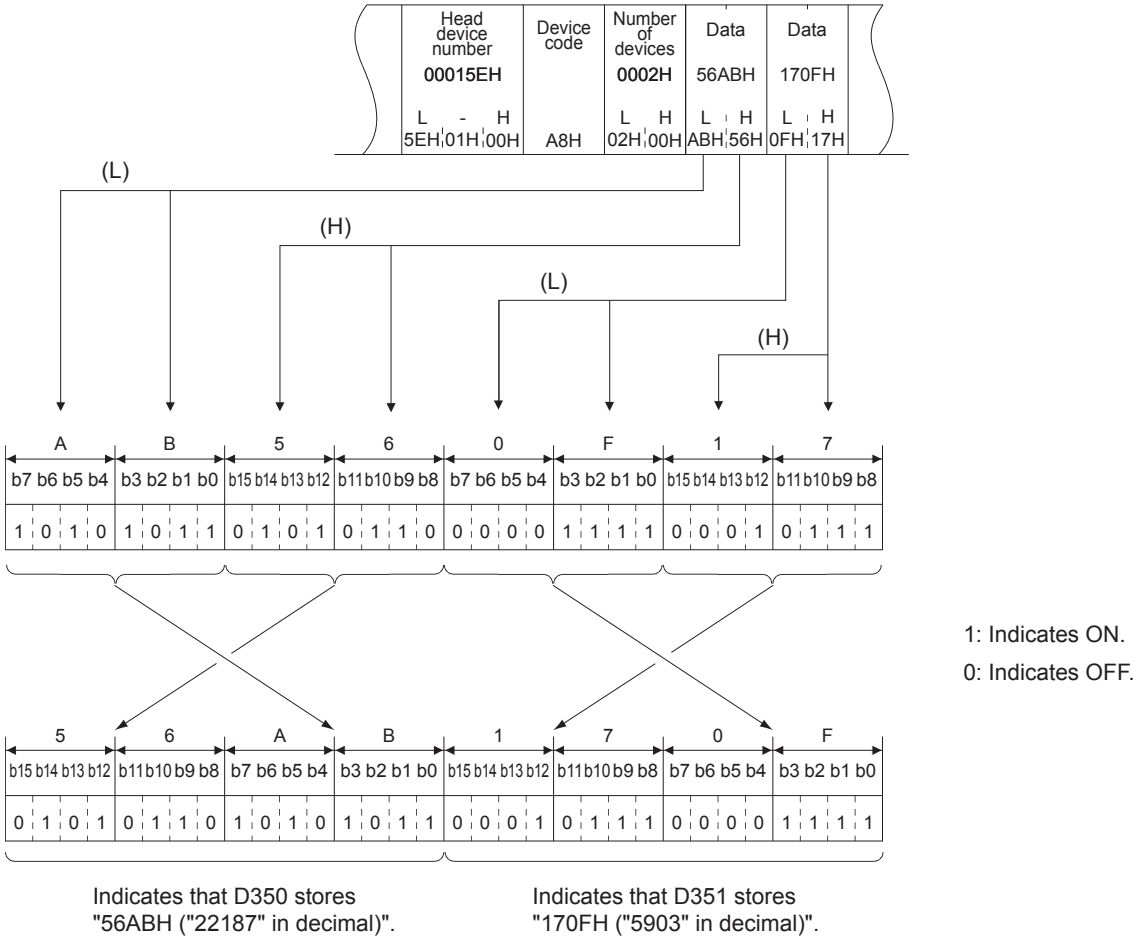


■When word device memory is read or written

In the word device memory, one word is specified by 16-bit and a specified number of devices starting from specified start device are expressed in 1-point units in turn from Low byte (L: bit 0 to 7) to High byte (H: bit 8 to 15).

Ex.

When indicating the contents stored in the data registers D350 and D351



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 to 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 to 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

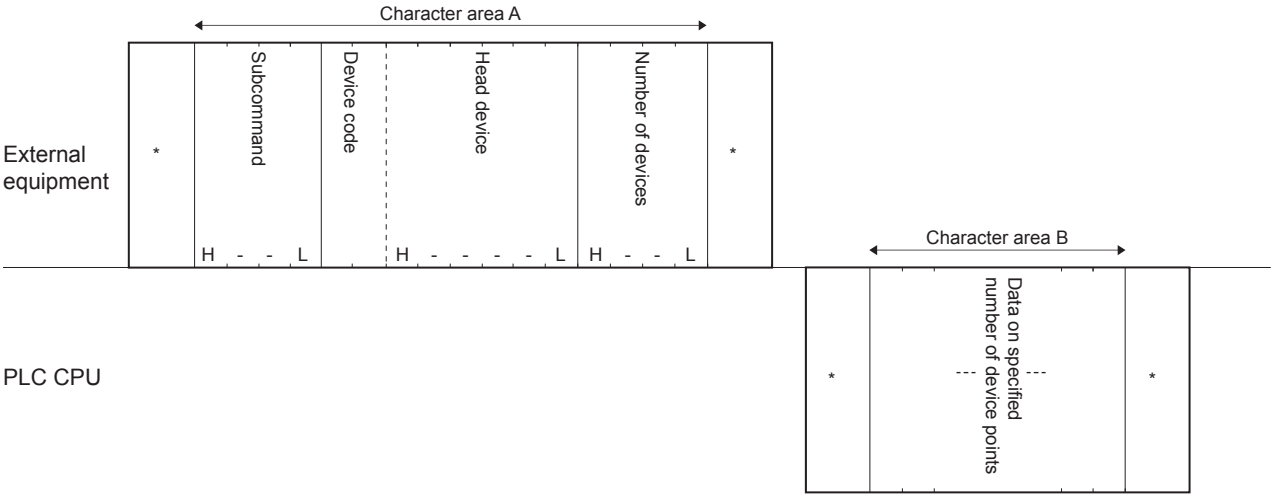
This section explains character areas in the control procedure (data area when communicating in binary code).

- 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.

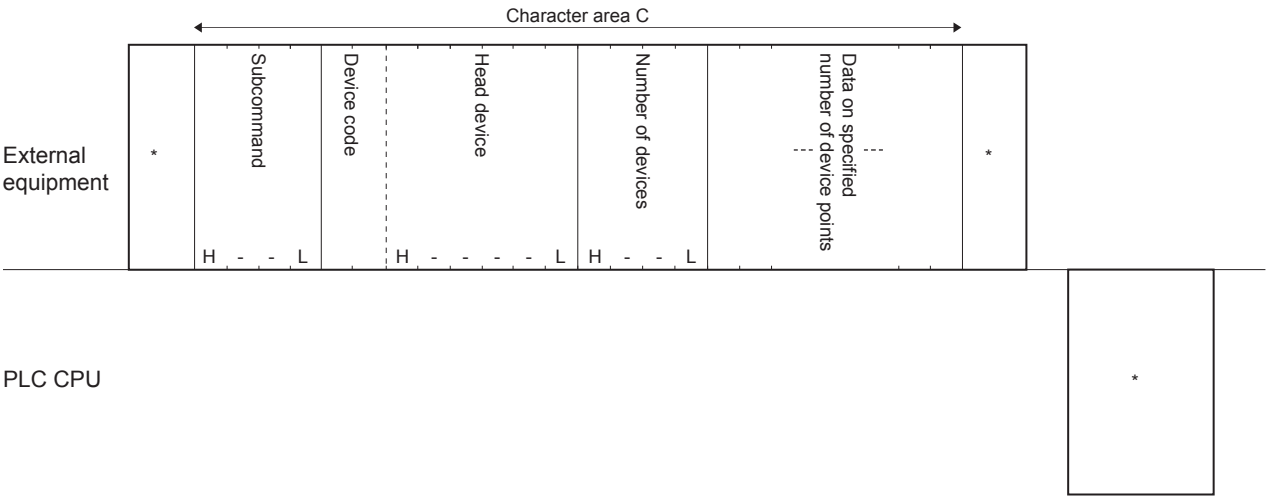
Data of character area (when communicating in ASCII code)

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



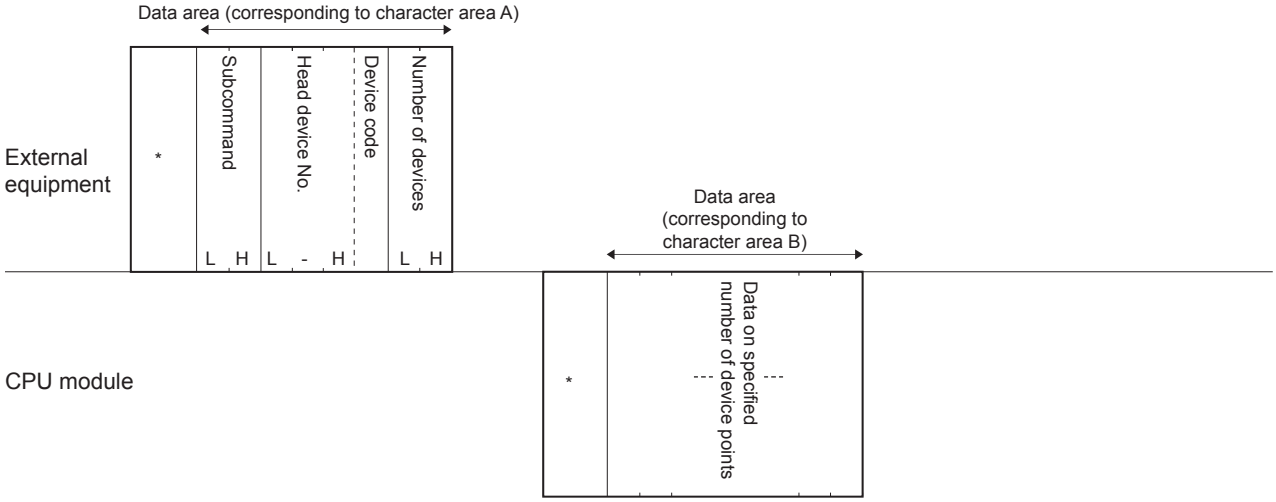
■In the case of writing



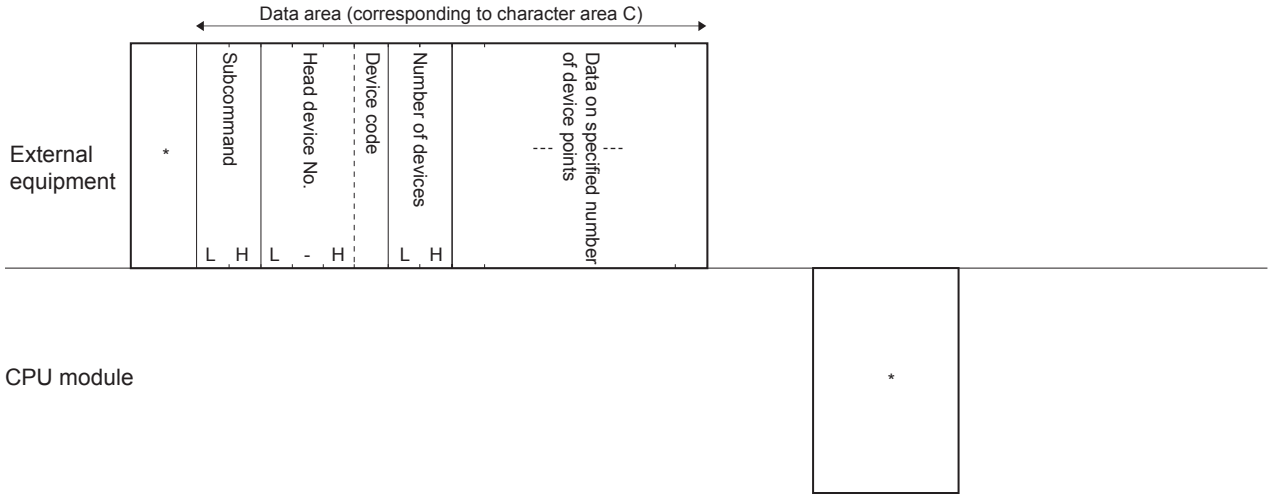
The data array and the data contents marked with * are shown in ☞ Page 15 Message Format.

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 15 Message Format.

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: 1 byte in binary • Device number: 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: 2 byte in binary • Device number: 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.

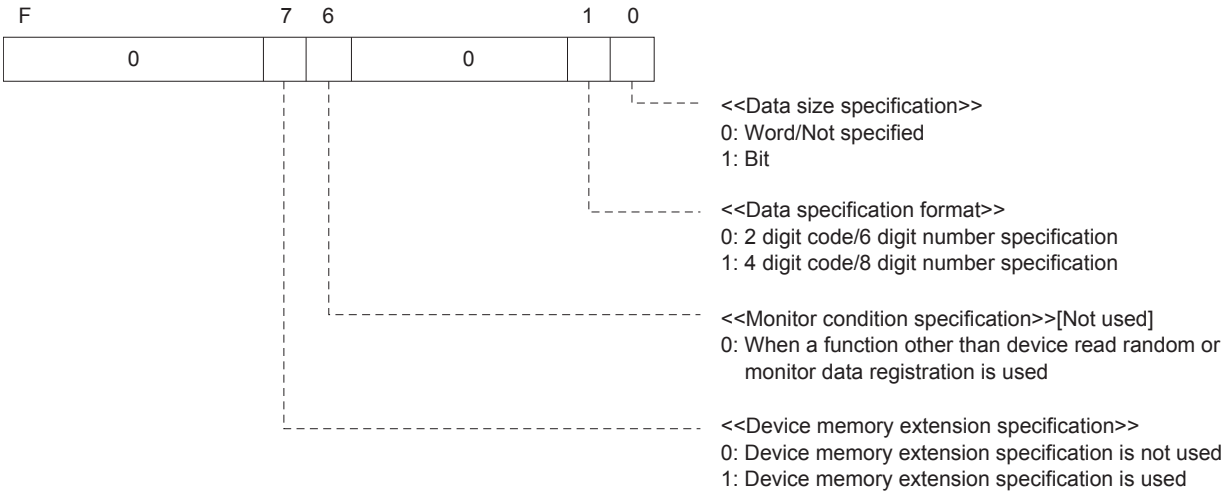
❶When communicating data in ASCII code

The value 0000H(0), or the following value, 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, is used for transmission.

❸The following figure shows the specification contents of the subcommand.



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.

❶Device codes are shown in the table in ☞ Page 44 Device range.

❷When communicating data in ASCII code

Device codes are converted into 2-digit ASCII code (when word device is specified) or 4-digit ASCII code (when long device is specified), 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

X	*
58H	2AH

4 digit code/8 digit number specification

X	*	*	*
58H	2AH	2AH	2AH

The input relay device code "X*" is sequentially sent from "X".

The second character "*" can be specified by a space (code: 20H).

❸When communicating data in binary code

The data is sent with the binary codes shown in ☞ Page 44 Device range are used.

Ex.

In the case of input (X)

2 digit code/6 digit number specification

9CH

4 digit code/8 digit number specification

00H	9CH

■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 head device number is specified by the data expression (decimal or hexadecimal) shown in the "Device number" column of the table shown in ☞ Page 44 Device range according to the corresponding device.

❶When communicating data in ASCII code

The device number shown in the table (☞ Page 44) is converted to a 6-digit ASCII code (when word device is specified) or 8-digit ASCII code (when long device is specified), 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.

In the case of 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			B1234		
D2H	04H	00H	34H	12H	00H

4 digit code/8 digit number specification

M1234				B1234			
D2H	04H	00H	00H	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 devices

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 to the number of points that can be processed per communication shown in the table in ☞ Page 43 Commands.

❶When communicating data in ASCII code

Points are converted into 4-digit hexadecimal ASCII code (when word device is specified) or 8-digits ASCII code (when long device is specified) with the device number that specified by the device specification format and sequentially sent beginning from the most significant 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

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	20 points
05H 00H	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 15 Message Format.

■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 to the number of points processed per communication shown in the table in ☞ Page 43 Commands.

❶When communicating data in ASCII code

The number of the bytes is converted into 2-digit ASCII code (hexadecimal) and sequentially sent beginning from the most significant digit. Use capitalized code for alphabetical letter in ASCII code.

Ex.

In the case of 5 points and 20 points

5 points	20 points
0 0 0 5 30H 30H 30H 35H	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	20 points
05H 00H	14H 00H

■Device memory extension specification (subcommand: bit7)

For details, refer to ☞ Page 94 Device Memory Extension Specification.

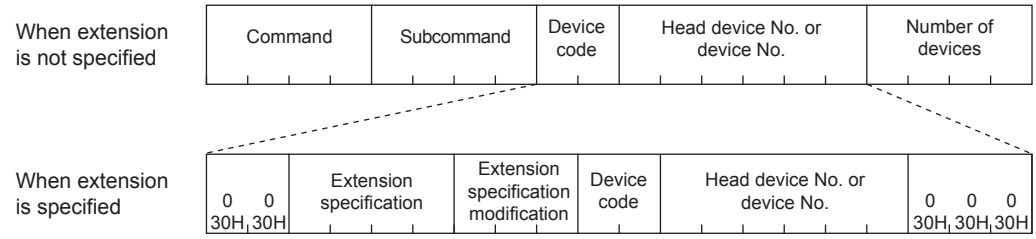
This section explains how to read or write from/to a device to/from module access device areas and how to specify a device indirectly by using index register.

❶Message format

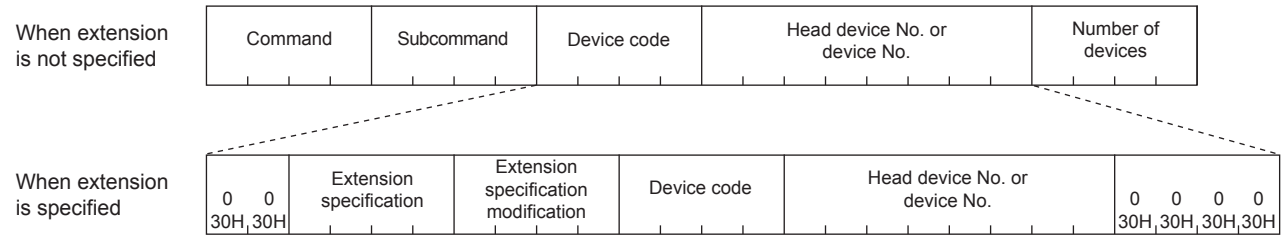
Response messages are extended as well.

■When communicating data in ASCII code

2 digit code/6 digit number specification

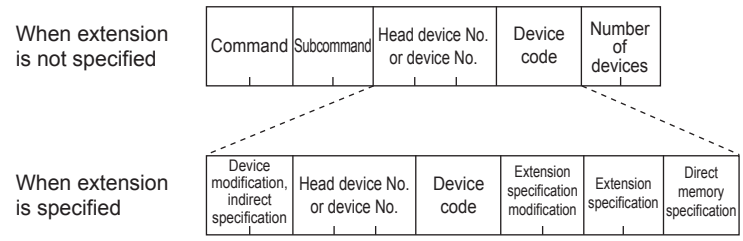


4 digit code/8 digit number specification

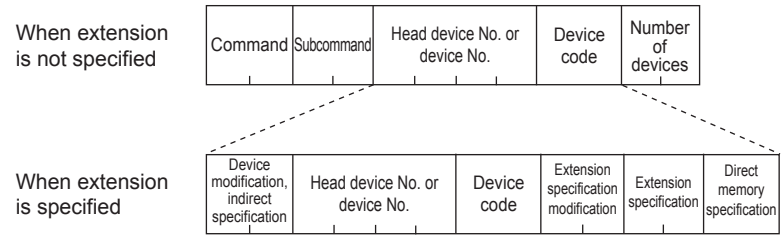


■When communicating data in binary code

2 digit code/6 digit number specification

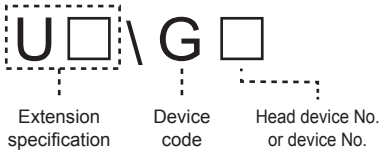


4 digit code/8 digit number specification



❷Module access device specification

The following shows the approach for module access device specification in programming and request data.



■Extension specification

Specify the module 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 module number in hexadecimal (2 bytes). When described with 4-digits, specify the module number with the upper 3-digits. □□H □□H Example 001 01H 00H

■Device code

Specify the module access device in the device code list.

■Head device No. or device No.

The format is the same as the message when extension is not specified.

4 COMMANDS

This chapter explains commands of SLMP.

For parts of the transmission message other than the command part, refer to ☞ Page 15 MESSAGE FORMAT

4.1 List of Commands and Functions

This section describes commands and functions when accessing from the external equipment to the CPU module.

Name	Command	Sub-commands	Processing content	Number of points processed per communication
Device Read (Batch)	0401H	0001H	This command reads data from a bit device or word device in units of 1 bit.	ASCII: 1792 points BIN: 3584 points
		0000H	• 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 link direct devices in units of 1 bit. • 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 link direct devices in units of 1 word. • 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 link direct devices in units of 1 bit. • 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 link direct devices in units of 1 word. • 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: 1972 points BIN: 3584 points
		0082H	This command writes data to the buffer memory in intelligent function modules and SLMP-compatible devices in units of 1 word (16 bits).	ASCII: 480 words (7680 points) BIN: 960 words (15360 points)
Device Read Random	0403H	0000H	This command reads data from word devices in units of 1 word or 2 words by randomly specifying device numbers.	ASCII: (Word access points + double-word access points) ×2 ≤192 BIN: Word access points + double-word access points ≤192

Name	Command	Sub-commands	Processing content	Number of points processed per communication
Device Read Random	0403H	0080H	This command reads data from the buffer memory in intelligent function modules and SLMP-compatible devices in units of 1 word (16 bits).	ASCII: (Word access points + double-word access points) ×4 ≤192 BIN: Word access points + double-word access points ≤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 ≤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 BIN: (Word access points)×12+ (double-word access points) ×14 ≤1920
		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 BIN: ((Word access points)×12+ (double-word access points) ×14) ×2 ≤1920
		0083H	This command writes data to the buffer memory in intelligent function modules and SLMP-compatible devices in units of 1 bit.	ASCII: 47 points BIN: 94 points
		0082H	This command writes data to the buffer memory in intelligent function modules and SLMP-compatible devices in units of 1 word (16 bits) or 2 words.	ASCII: ((Word access points) ×12+ (double-word access points) ×14) ×4 ≤1920 BIN: ((Word access points)×12+ (double-word access points) ×14) ×2 ≤1920
Device Read Block	0406H	0000H	With n points of bit devices and word devices as 1 block, this command reads data by randomly specifying multiple blocks. (When bit devices are specified, 1 point is 16 bits.)	ASCII: (Number of word device blocks + number of bit device blocks) ×2 ≤120 and (Total points of each blocks of word device + total points of each blocks of bit device) ×2 ≤960 BIN: Number of word device blocks + number of bit device blocks ≤120 and Total points of each blocks of word device + total points of each blocks of bit device ≤960

Name	Command	Sub-commands	Processing content	Number of points processed per communication
Device Read Block	0406H	0080H	With n points of buffer memory in intelligent function modules and SLMP-compatible devices as 1 block, this command reads data by randomly specifying multiple blocks. (When bit devices are specified, 1 point is 16 bits.)	ASCII: (Number of word device blocks + number of bit device blocks) ×4 ≤120 and (Total points of each blocks of word device + total points of each blocks of bit device) ×2 ≤960 BIN: (Number of word device blocks + number of bit device blocks) ×2 ≤120 and Total points of each blocks of word device + total points of each blocks of bit device ≤960
		0082H	With n points of buffer memory in intelligent function modules and SLMP-compatible devices as 1 block, this command reads data by randomly specifying multiple blocks.	ASCII: (Number of word device blocks + number of bit device blocks) ×4 ≤120 and (Total points of each blocks of word device + total points of each blocks of bit device) ×2 ≤960 BIN: (Number of word device blocks + number of bit device blocks) ×2 ≤120 and Total points of each blocks of word device + total points of each blocks of bit device ≤960
Device Write Block	1406H	0000H	With n points of bit devices and word devices as 1 block, this command writes data by randomly specifying multiple blocks. (When bit devices are specified, 1 point is 16 bits.)	ASCII: (Number of word device blocks + number of bit device blocks) ×2 ≤120 and ((Number of word device blocks + number of bit device blocks) ×4 + Total points of each blocks of word device + total points of each blocks of bit device) ×2 ≤770 BIN: Number of word device blocks + number of bit device blocks ≤120 and (Number of word device blocks + number of bit device blocks) ×4 + Total points of each blocks of word device + total points of each blocks of bit device ≤770
		0080H	With n points of buffer memory in intelligent function modules and SLMP-compatible devices as 1 block, this command writes data by randomly specifying multiple blocks. (When bit devices are specified, 1 point is 16 bits.)	ASCII: (Number of word device blocks + number of bit device blocks) ×4 ≤120 and ((Number of word device blocks + number of bit device blocks) ×4 + Total points of each blocks of word device + total points of each blocks of bit device) ×2 ≤770 BIN: (Number of word device blocks + number of bit device blocks) ×2 ≤120 and (Number of word device blocks + number of bit device blocks) ×4 + Total points of each blocks of word device + total points of each blocks of bit device ≤770

Name	Command	Sub-commands	Processing content	Number of points processed per communication
Device Write Block	1406H	0082H	With n points of buffer memory in intelligent function modules and SLMP-compatible devices as 1 block, this command writes data by randomly specifying multiple blocks.	ASCII: (Number of word device blocks + number of bit device blocks) ×4 ≤120 and ((Number of word device blocks + number of bit device blocks) ×4 + Total points of each blocks of word device + total points of each blocks of bit device) ×2 ≤770 BIN: (Number of word device blocks + number of bit device blocks) ×2 ≤120 and (Number of word device blocks + number of bit device blocks) ×4 + Total points of each blocks of word device + total points of each blocks of bit device ≤770
Remote Run	1001H	0000H	This command performs a remote RUN request for a device.	—
Remote Stop	1002H	0000H	This command performs a remote STOP request for a device.	—
Remote Pause	1003H	0000H	This command performs a remote PAUSE request for a device.	—
Remote Latch Clear	1005H	0000H	This command performs a remote latch clear request when the device is in the STOP state.	—
Remote Reset	1006H	0000H	This command performs a remote reset request to reset the device error stop state.	—
Read Type Name	0101H	0000H	This command reads the processor module name code (processor type) of a device.	—
Global	1618H	0000H	Turns off the global signal.	—
		0001H	Turns on the global signal.	—
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.)	—

4.2 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	CPU module status		
				STOP	RUN	
					Write allow setting	Write prohibit setting
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.	○	○	×

○: Available, ×: Unavailable

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 FX5CPU

Classification	Device		Type	Device code[*1] (Device specification format: Long)		Device No.		Applicable FX5CPU device[*2]
				ASCII code	Binary code			
Internal user device	Input		Bit	X* (X***)	9CH (9C00H)	Specify in the range of device numbers of the module to access.	Octal	○
	Output			Y* (Y***)	9DH (9D00H)		Octal	○
	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)	Specify in the range of device numbers of the module to access.	Decimal	○
	Link register			W* (W***)	B4H (B400H)		Hexadecimal	○
	Timer	Contact	Bit	TS (TS**)	C1H (C100H)		Decimal	○
		Coil	Bit	TC (TC**)	C0H (C000H)			○
		Current value	Word	TN (TN**)	C2H (C200H)			○
	Long timer	Contact	Bit	— (LTS*)	51H (5100H)		Decimal	—
		Coil	Bit	— (LTC*)	50H (5000H)			—
		Current value	Double Word	— (LTN*)	52H (5200H)			—
	Retentive timer	Contact	Bit	SS (STS*)	C7H (C700H)		Decimal	○
		Coil	Bit	SC (STC*)	C6H (C600H)			○
		Current value	Word	SN (STN*)	C8H (C800H)			○
	Long retentive timer	Contact	Bit	— (LSTS)	59H (5900H)		Decimal	—
		Coil	Bit	— (LSTC)	58H (5800H)			—
		Current value	Double Word	— (LSTN)	5AH (5A00H)			—
	Counter	Contact	Bit	CS (CS**)	C4H (C400H)		Decimal	○
		Coil	Bit	CC (CC**)	C3H (C300H)			○
		Current value	Word	CN (CN**)	C5H (C500H)			○

Classification	Device		Type	Device code*1 (Device specification format: Long)		Device No.		Applicable FX5CPU device*2
				ASCII code	Binary code			
Internal user device	Long counter	Contact	Bit	— (LCS*)	55H (5500H)	Specify in the range of device numbers of the module to access.	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)	Specify in the range of device numbers of the module to access.	Decimal	○
	Special register		Word	SD (SD**)	A9H (A900H)		Decimal	○
	Function input		Bit	—	—	—	Hexadecimal	—
	Function output			—	—		Hexadecimal	—
	Function register		Word	—	—		Decimal	—
Index register			Word	Z* (Z***)	CCH (CC00H)	Specify in the range of device numbers of the module to access.	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*3	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*3	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]
If the device code is less than the specified character number, add "*" (ASCII code: 2AH) or a space (ASCII code: 20H) after the device code.
[Binary code]
When "Device code" is less than the size specified add "00H" to the end of the device code.

*2 ○: An FX5CPU device exists
—: No FX5CPU device

*3 "Device memory extension specification" for sub-commands must be turned ON (1).

Device Read (Batch)

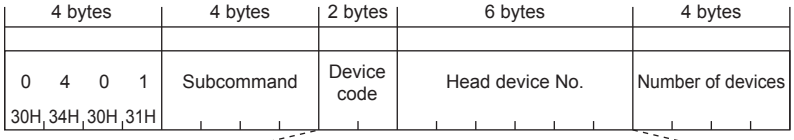
Data in devices are read in a batch.

Request data

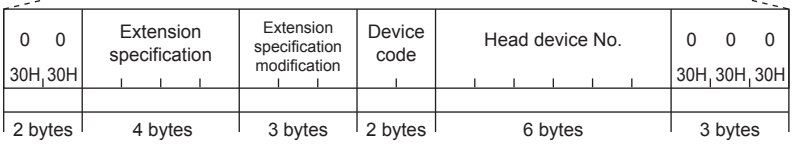
■When communicating data in ASCII code

2 digit code/6 digit number specification

When extension is not specified

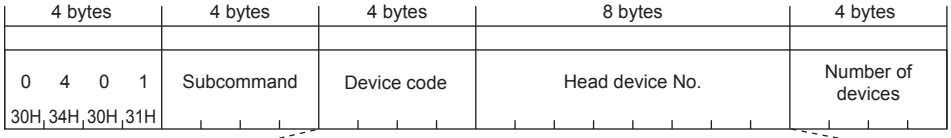


When extension is specified

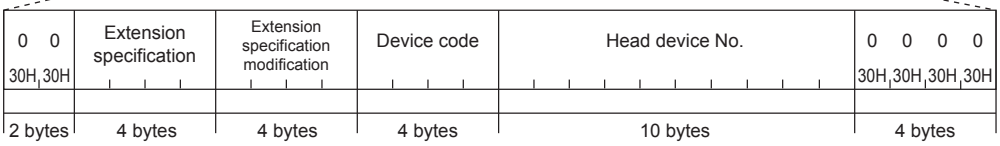


4 digit code/8 digit number specification

When extension is not specified



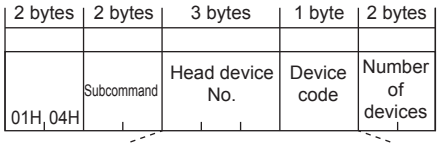
When extension is specified



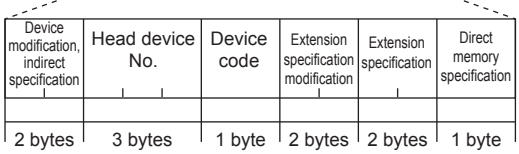
■When communicating data in binary code

2 digit code/6 digit number specification

When extension is not specified

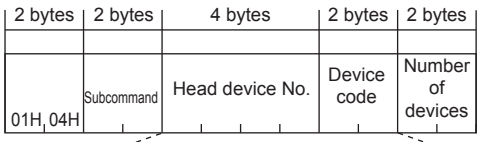


When extension is specified

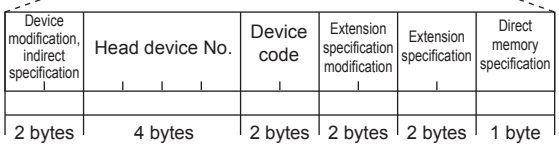


4 digit code/8 digit number specification

When extension is not specified



When extension is specified



■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	30H	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.)

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 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)

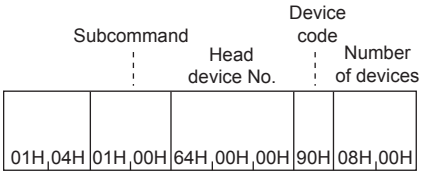
	Subcommand	Device code	Head device No.	Number of devices
0 4 0 1	0 0 0 1	M *	0 0 0 1 0 0	0 0 0 8
30H 34H 30H 31H	30H 30H 30H 31H	4DH 2AH	30H 30H 30H 31H 30H 30H	30H 30H 30H 38H

(Response data)

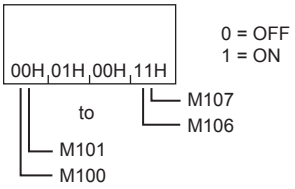
0 0 0 1 0 0 1 1	0 = OFF 1 = ON
30H 30H 30H 31H 30H 30H 31H 31H	
M100 to M107	

• When communicating data in binary code

(Request data)



(Response data)

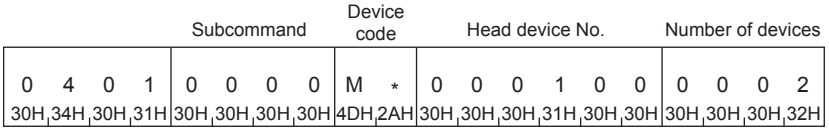


■When reading data in word units (bit device)

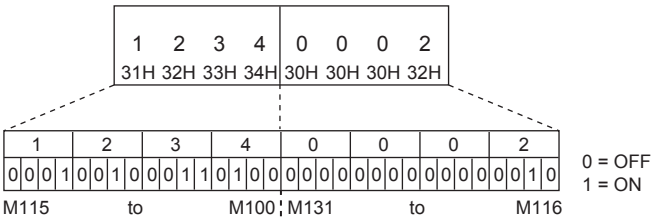
M100 to M131 (2-word) are read.

• When communicating data in ASCII code

(Request data)

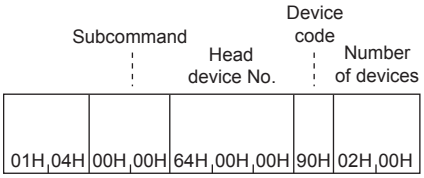


(Response data)

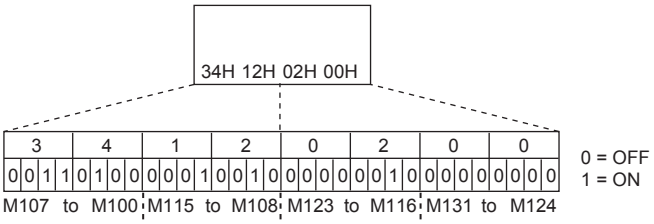


• When communicating data in binary code

(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 T102.

• When communicating data in ASCII code

(Request data)

	Subcommand	Device code	Head device No.	Number of devices
0 4 0 1	0 0 0 0	T N	0 0 0 1 0 0	0 0 0 3
30H 34H 30H 31H	30H 30H 30H 30H	54H 4EH	30H 30H 30H 31H 30H 30H	30H 30H 30H 33H

(Response data)

1 2 3 4	0 0 0 2	1 D E F
31H 32H 33H 34H	30H 30H 30H 32H	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)

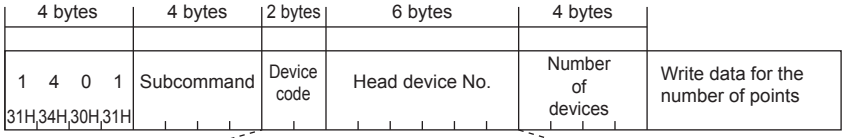
Data in devices are written in a batch.

Request data

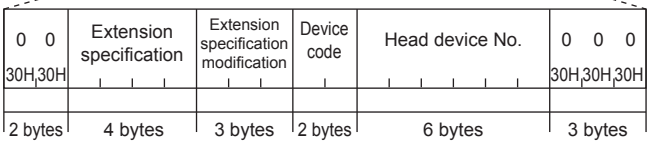
■When communicating data in ASCII code

2 digit code/6 digit number specification

When extension is not specified

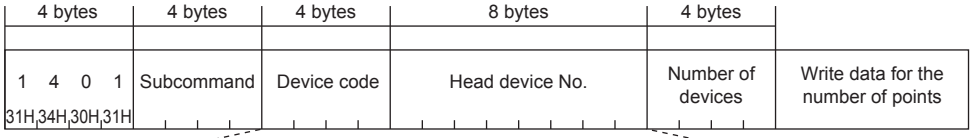


When extension is specified

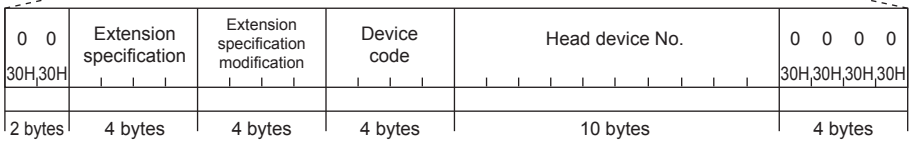


4 digit code/8 digit number specification

When extension is not specified



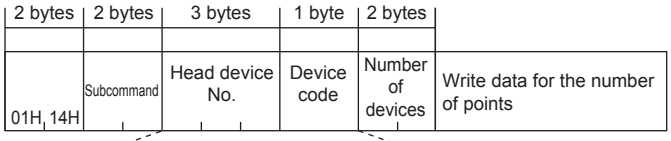
When extension is specified



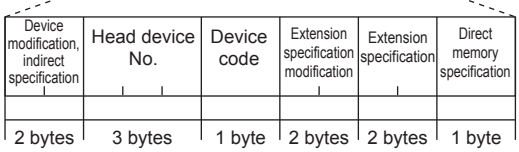
■When communicating data in binary code

2 digit code/6 digit number specification

When extension is not specified

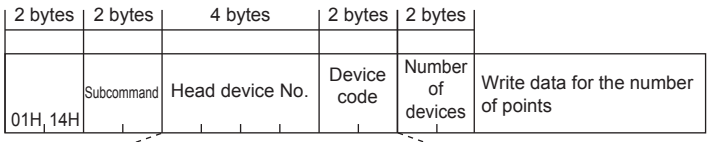


When extension is specified

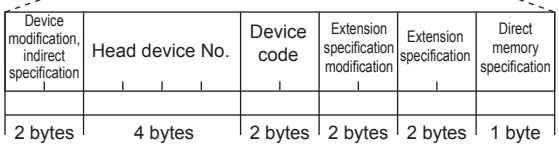


4 digit code/8 digit number specification

When extension is not specified



When extension is specified



■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.)

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 1792 points	1 to 3584 points
When writing data in word units	1 to 480 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

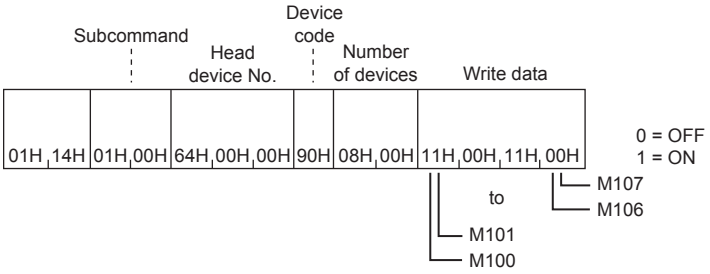
(Request data)

	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

M100 to M107 0 = OFF, 1 = ON

• When communicating data in binary code

(Request data)

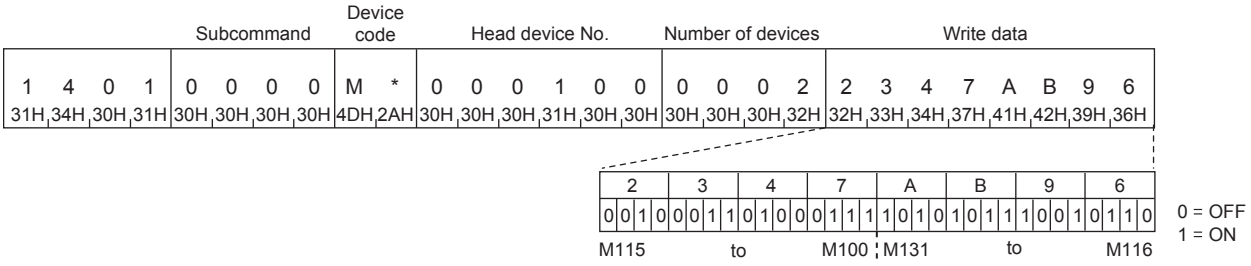


■When writing data in word units (bit device)

Values are written to M100 to M131 (2-word).

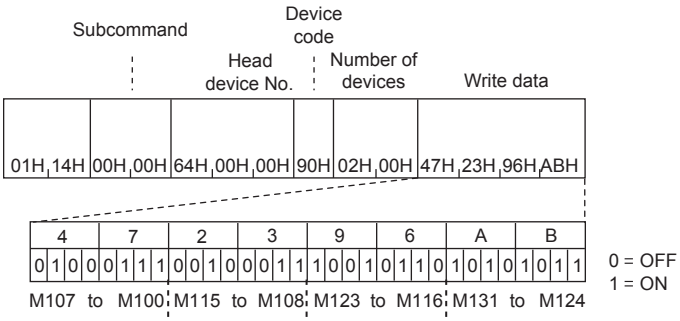
• When communicating data in ASCII code

(Request data)



• When communicating data in binary code

(Request data)

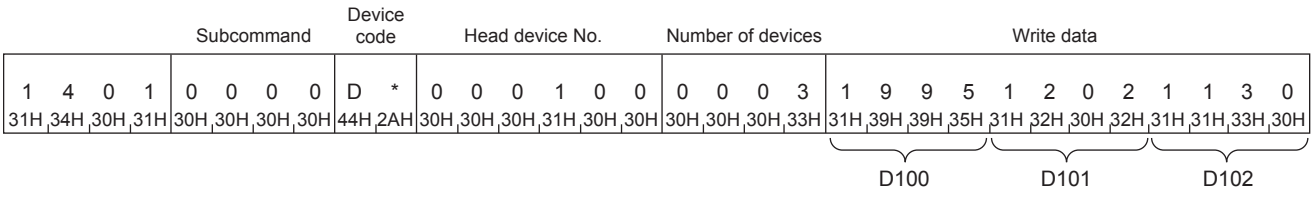


■When writing data in word units (word device)

6549(1995H) is written in D100, 4610(1202H) is written in D101, and 4400(1130H) is written in D102.

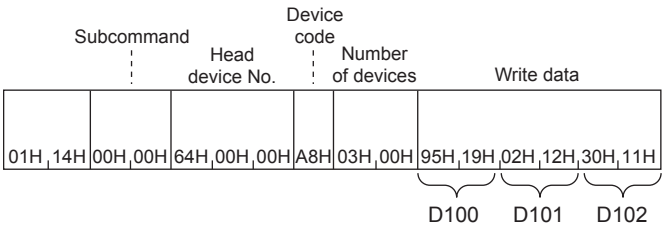
• When communicating data in ASCII code

(Request data)



• When communicating data in binary code

(Request data)



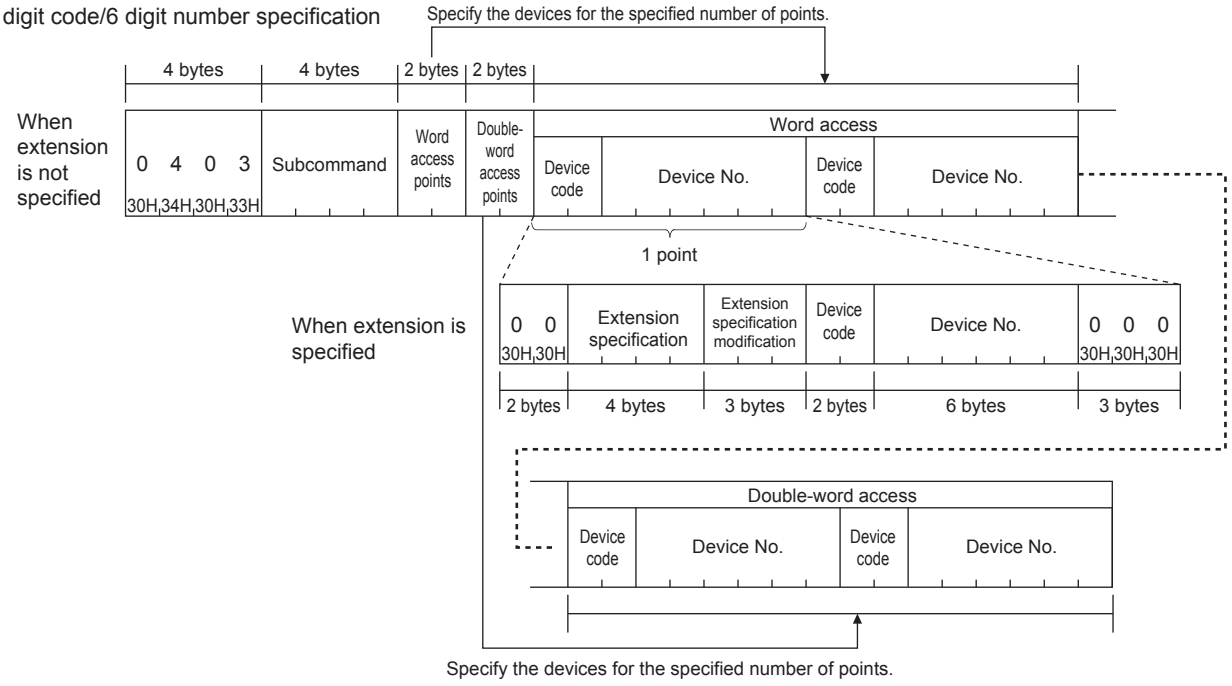
Device Read Random

This command specifies the device No. randomly and reads the device value.

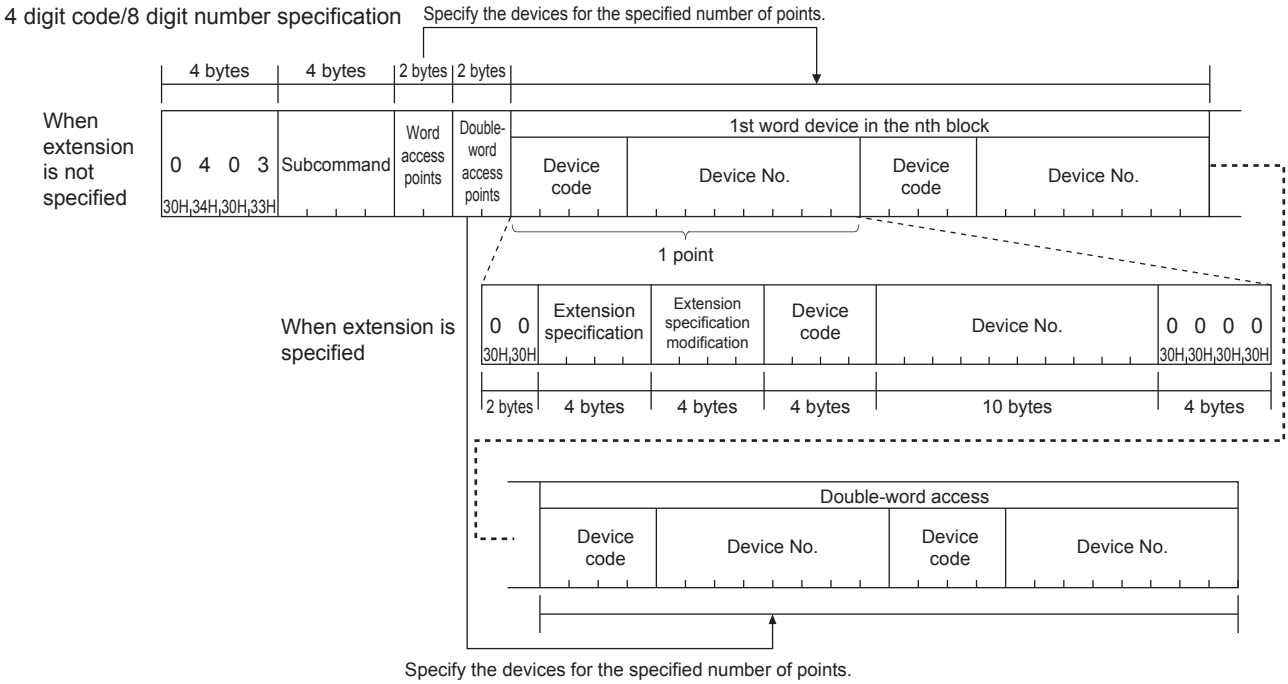
Request data

■When communicating data in ASCII code

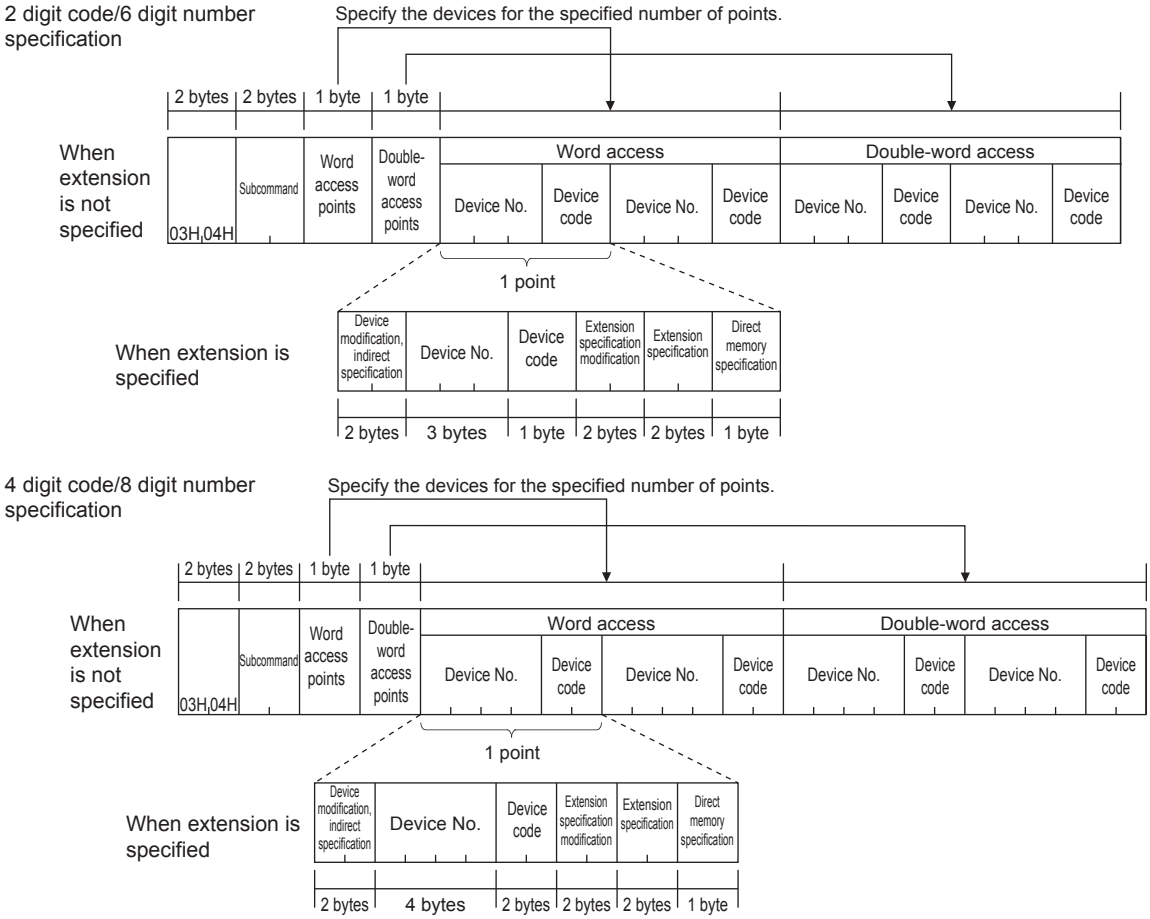
2 digit code/6 digit number specification



4 digit code/8 digit number specification



■When communicating data in binary code



■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
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.

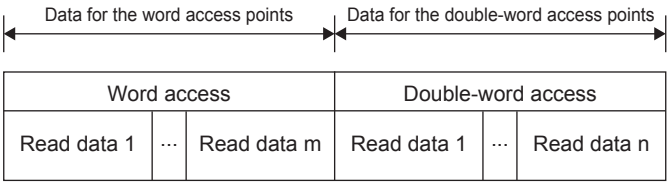
Specify the target device of reading.

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.



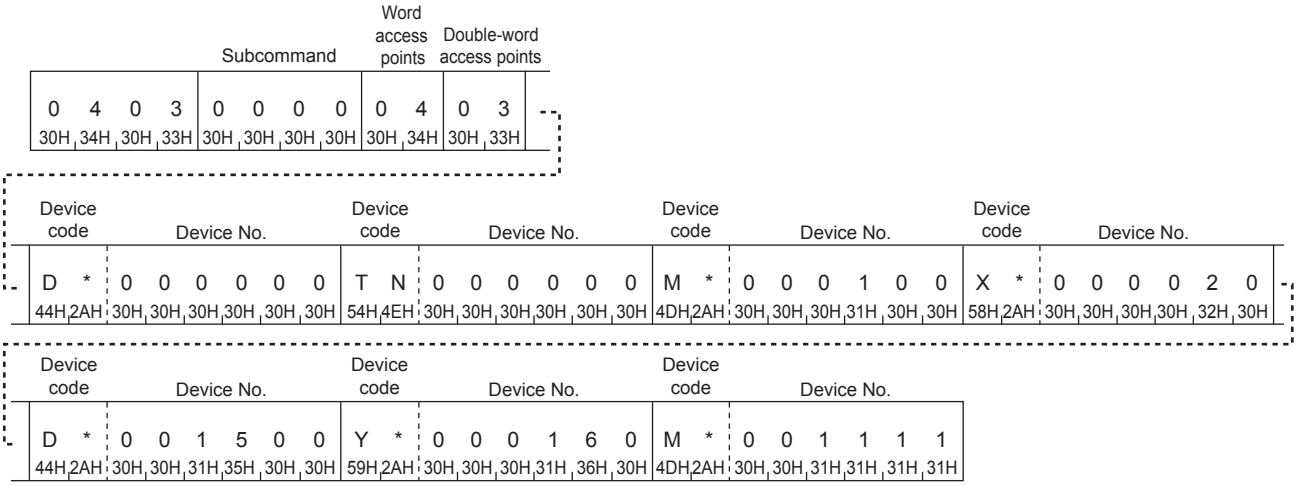
Communication example

Read D0, T0, M100 to M115, X20 to X2F by word access, and D1500 to D1501, Y160 to Y17F, M1111 to M1142 by double-word access.

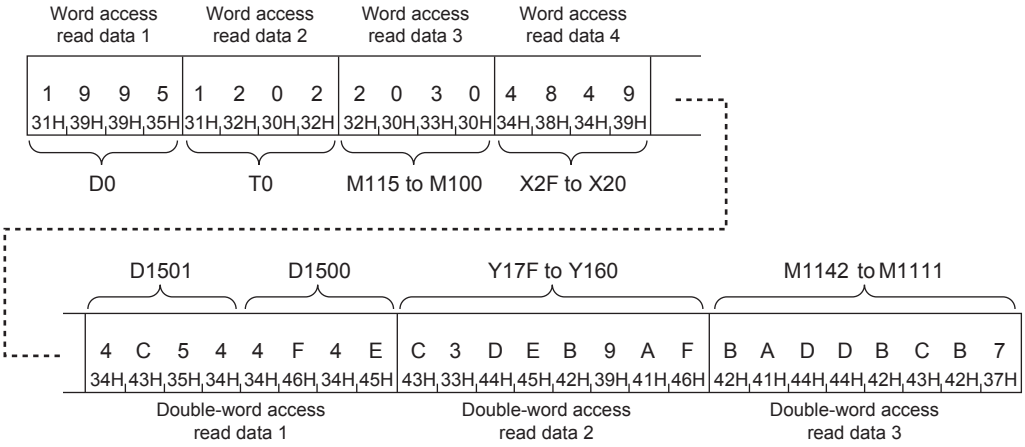
It is supposed that 6549(1995H) is stored in D0, 4610(1202H) is stored in T0, 20302(4F4EH) is stored in D1500, 19540(4C54H) is stored in D1501.

■When communicating data in ASCII code

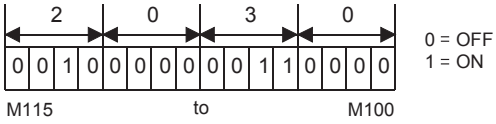
• Request data



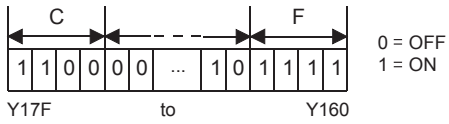
• Response data



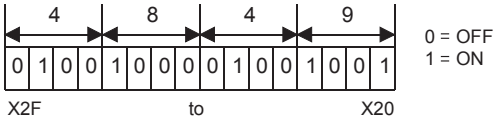
Word access read data 3



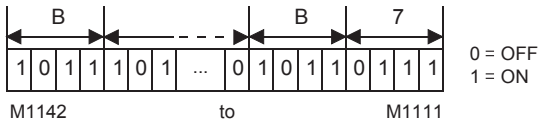
Double-word access read data 2



Word access read data 4

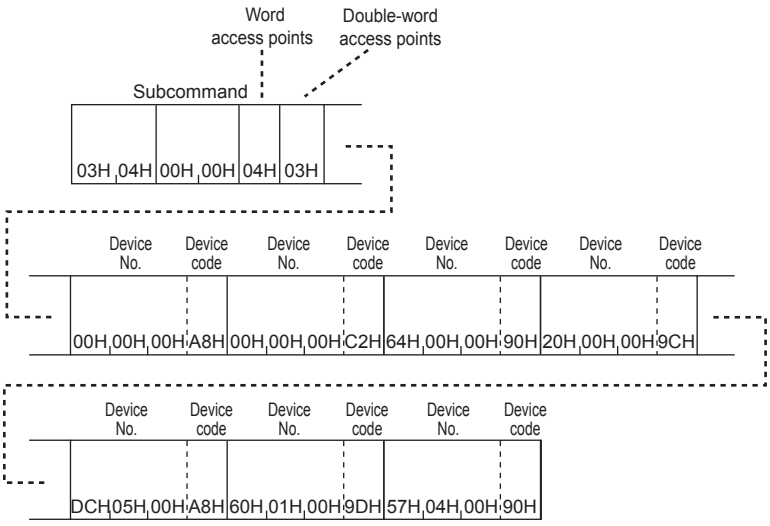


Double-word access read data 3

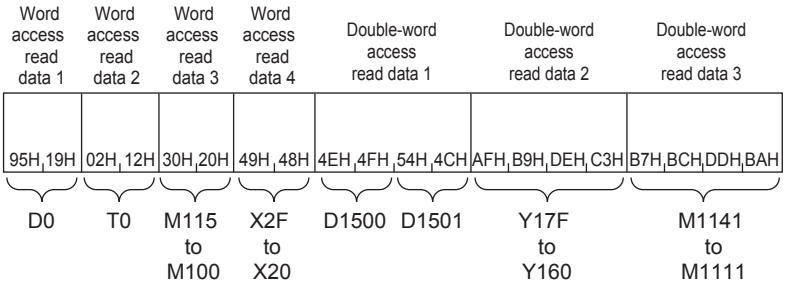


■When communicating data in binary code

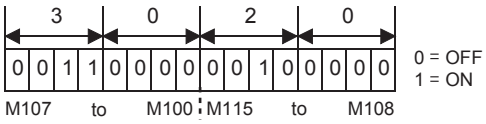
• Request data



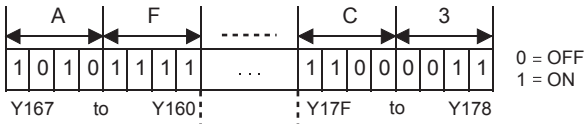
• Response data



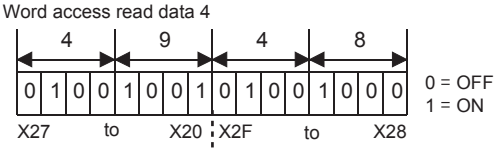
Word access read data 3



Double-word access read data 2



Word access read data 4



Double-word access read data 3

Device Write Random

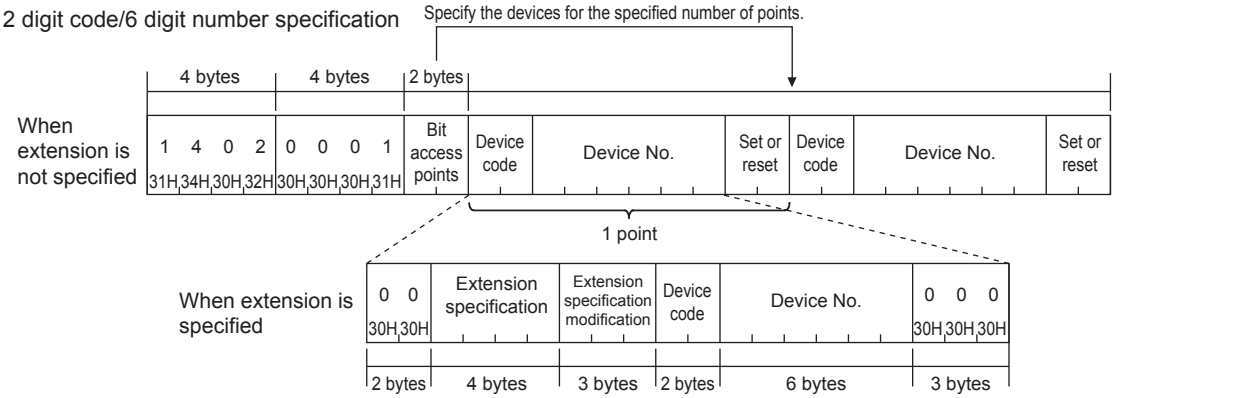
This command specifies the device No. randomly and writes the data.

Request data

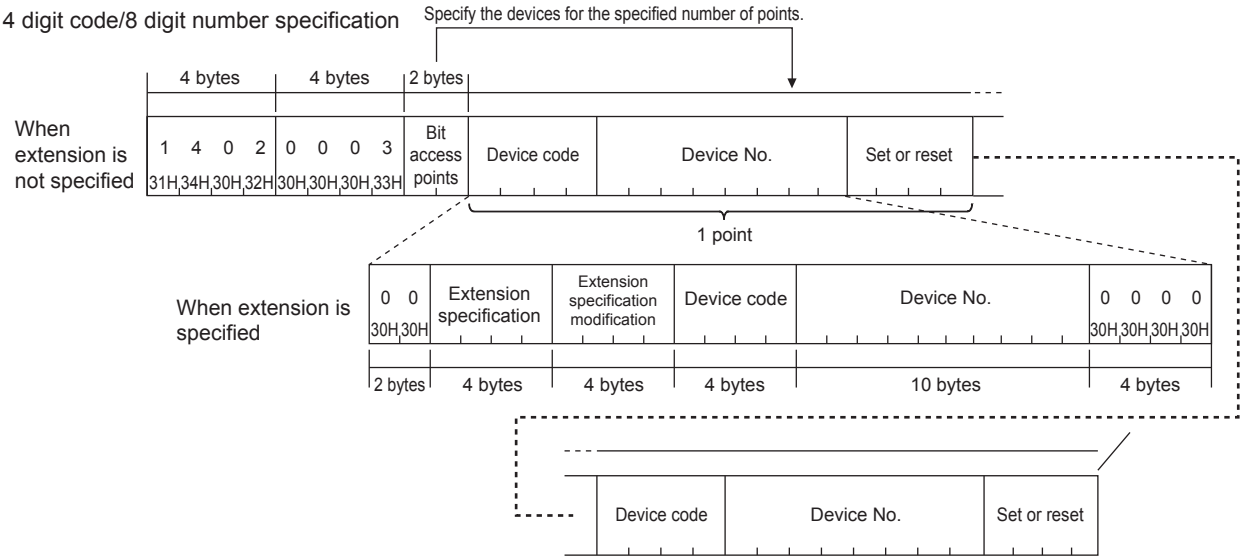
■When writing data in bit units

• When communicating data in ASCII code

2 digit code/6 digit number specification

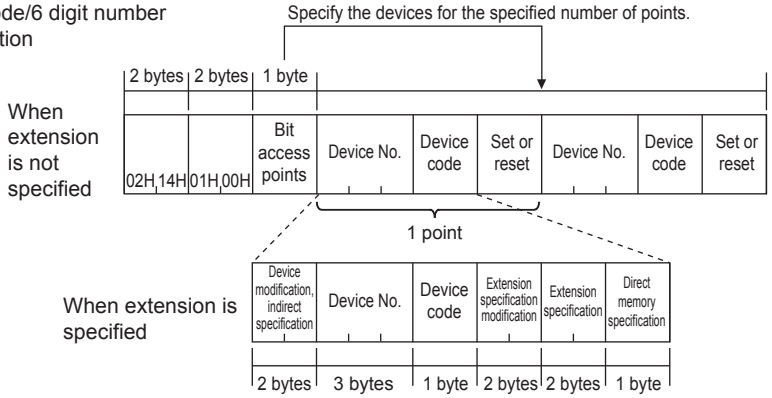


4 digit code/8 digit number specification

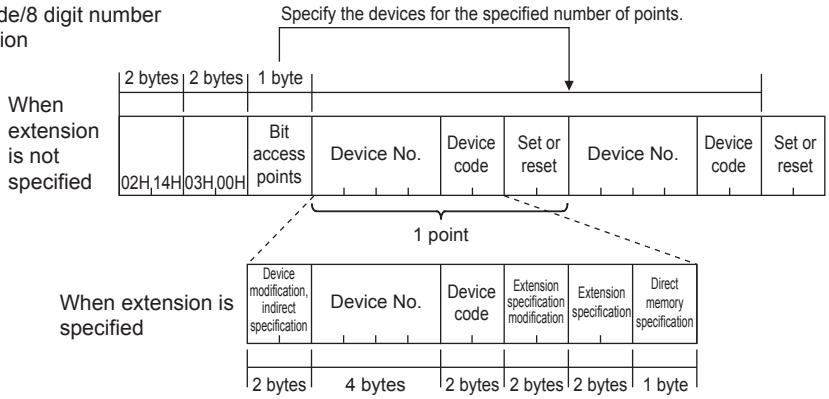


• When communicating data in binary code

2 digit code/6 digit number specification

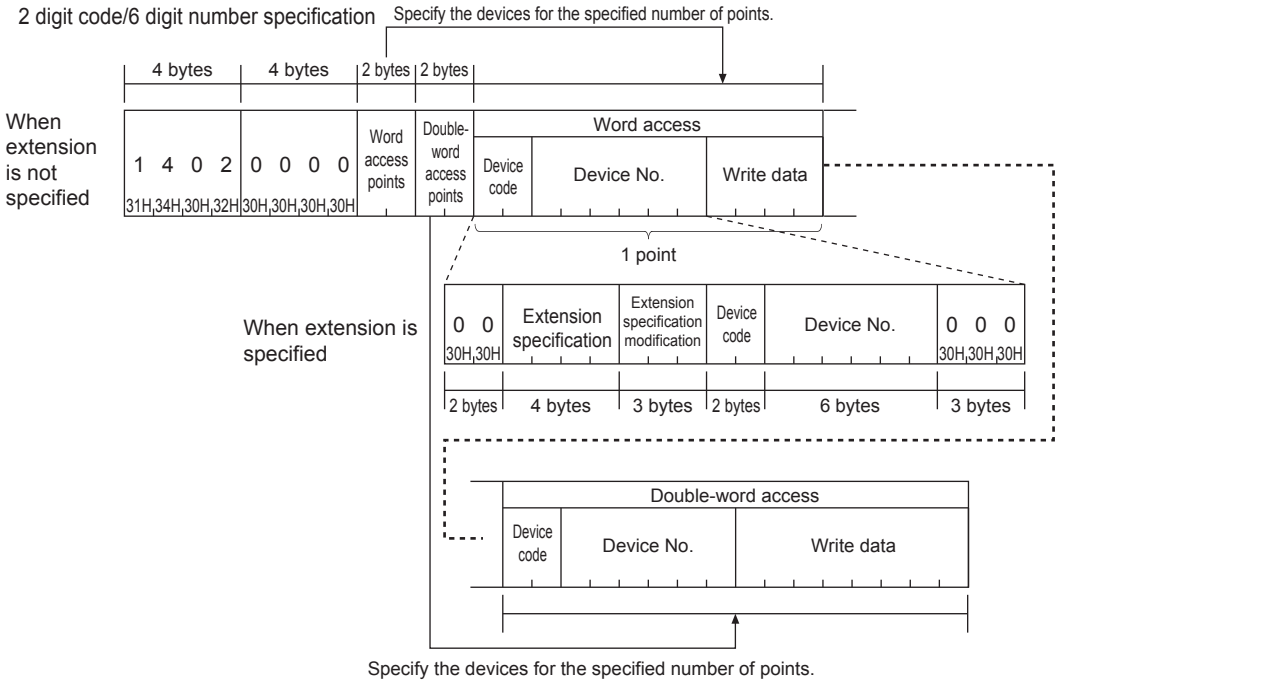


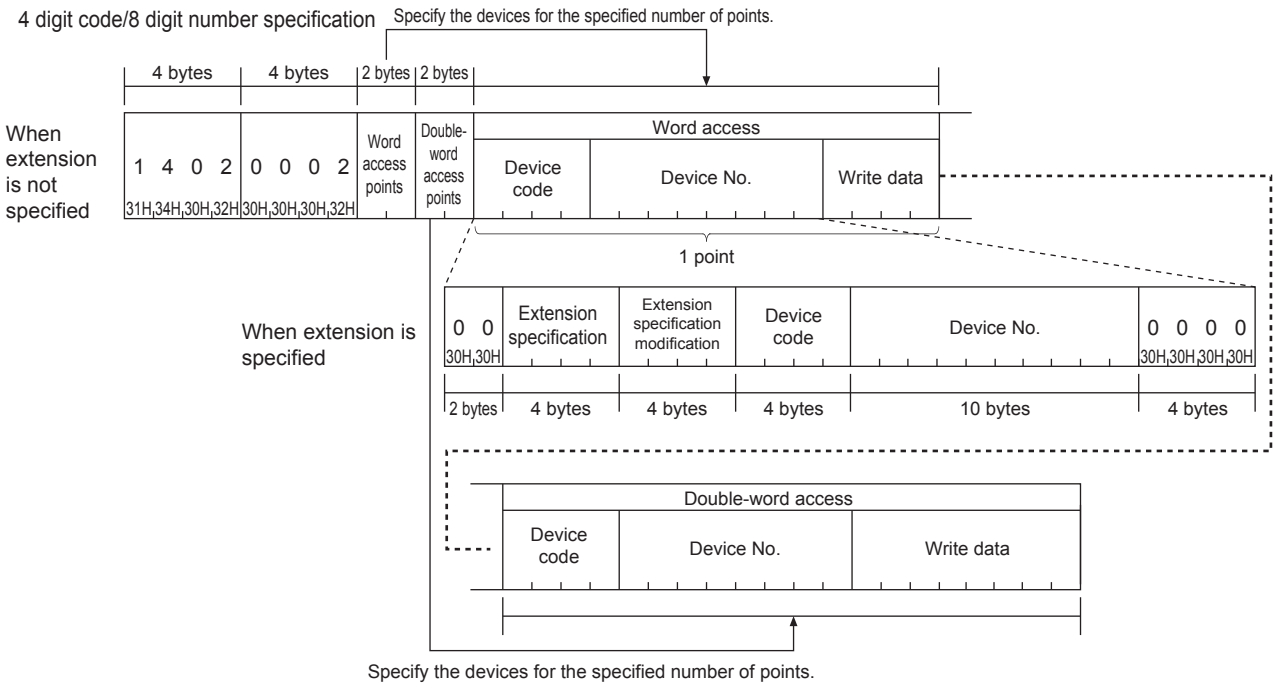
4 digit code/8 digit number specification



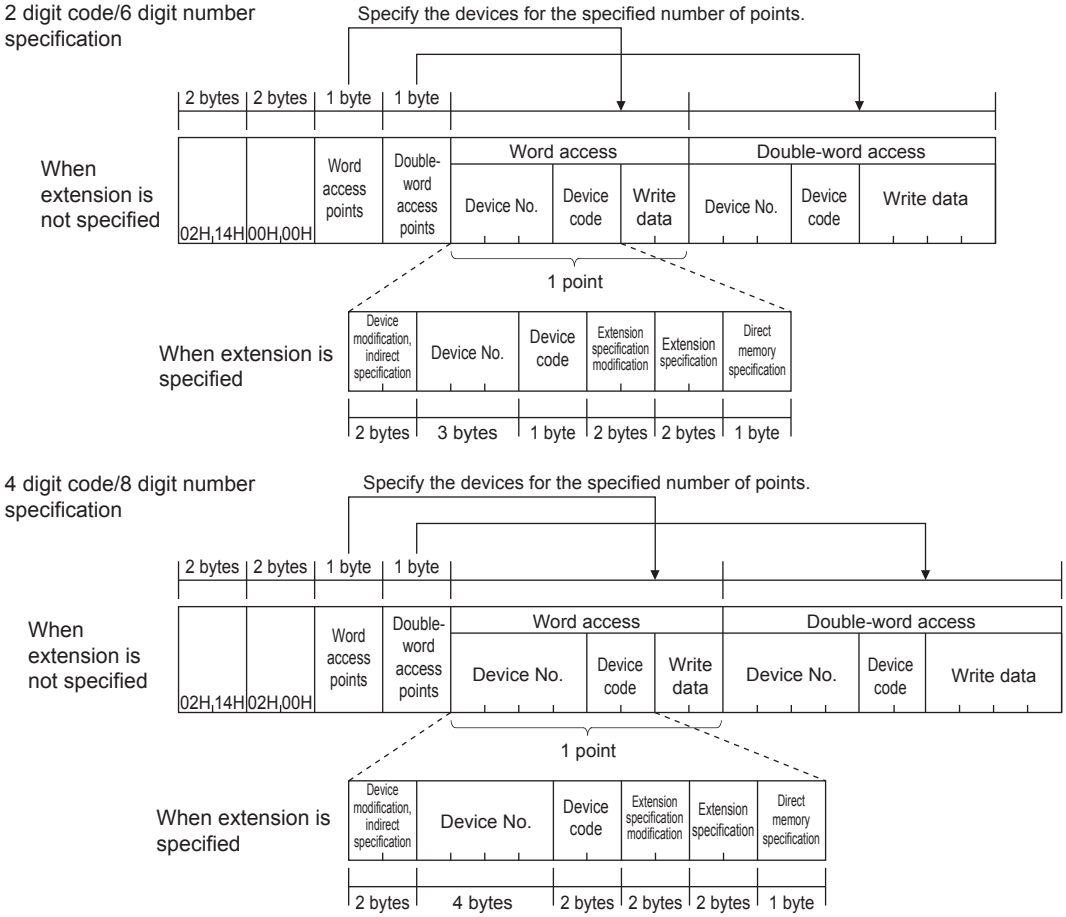
■When writing data in word units

• When communicating data in ASCII code





• When communicating data in binary code



■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

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 × 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 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 Write Random command.

Communication example

■When writing data in bit units

Turn off M50 and turn on Y2F.

• When communicating data in ASCII code

(Request data)

	Subcommand	Bit access points	Device code	Device No.	Set or reset	Device code	Device No.	Set or reset
	1402	0001	02	M*	000050	00	Y*	000002F	01
	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,46H	30H,31H

• When communicating data in binary code

(Request data)

	Subcommand	Bit access points	Device code	Device No.	Set or reset	Device code	Device No.	Set or reset
	0214	01	00	02	32,00,00	90	00	2F,00,00	9D	01
	02H,14H	01H,00H	02H	32H,00H,00H	90H	00H	2FH,00H,00H	9DH	01H

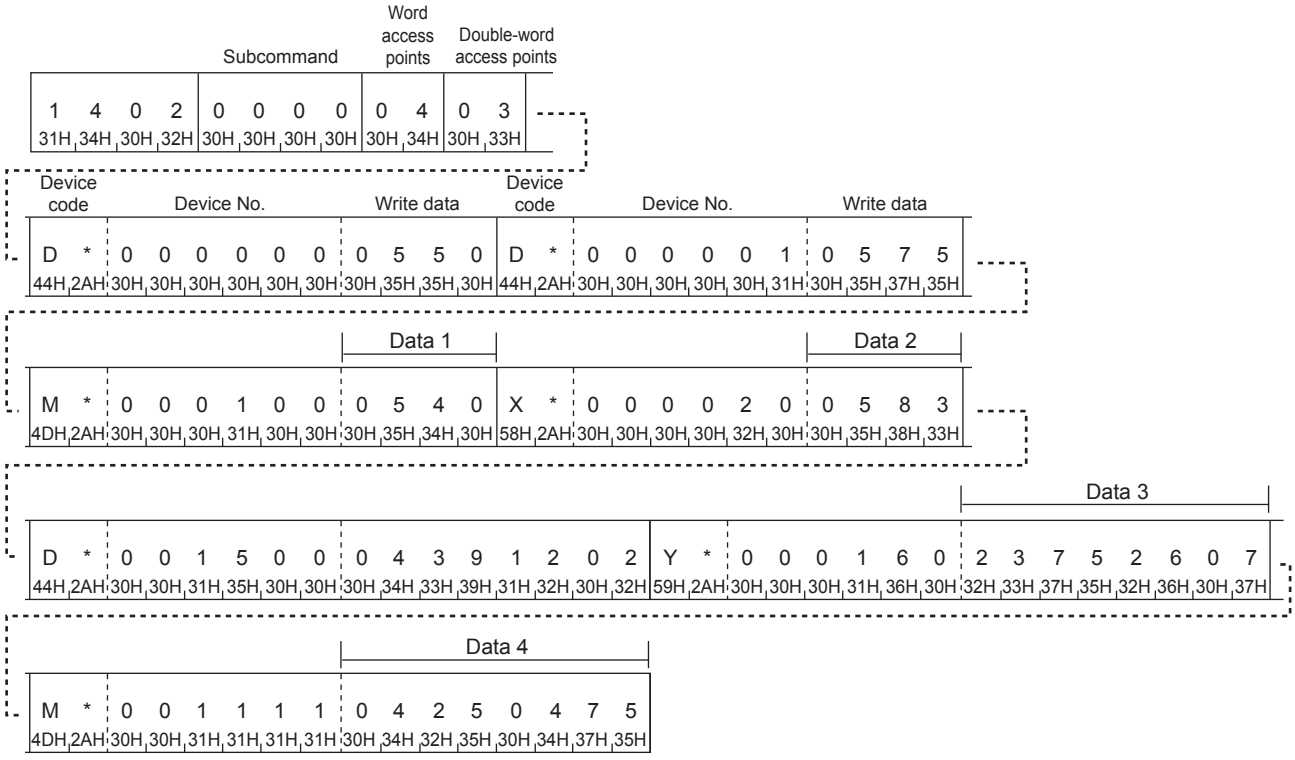
■When writing data in word units

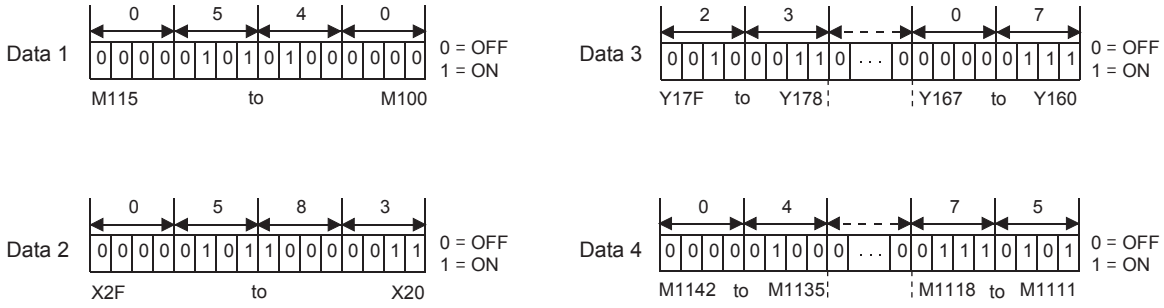
Write the value in a device as follows.

Item	Target device
Word access	D0, D1, M100 to M115, X20 to X2F
Double-word access	D1500 to D1501, Y160 to Y17F, M1111 to M1142

• When communicating data in ASCII code

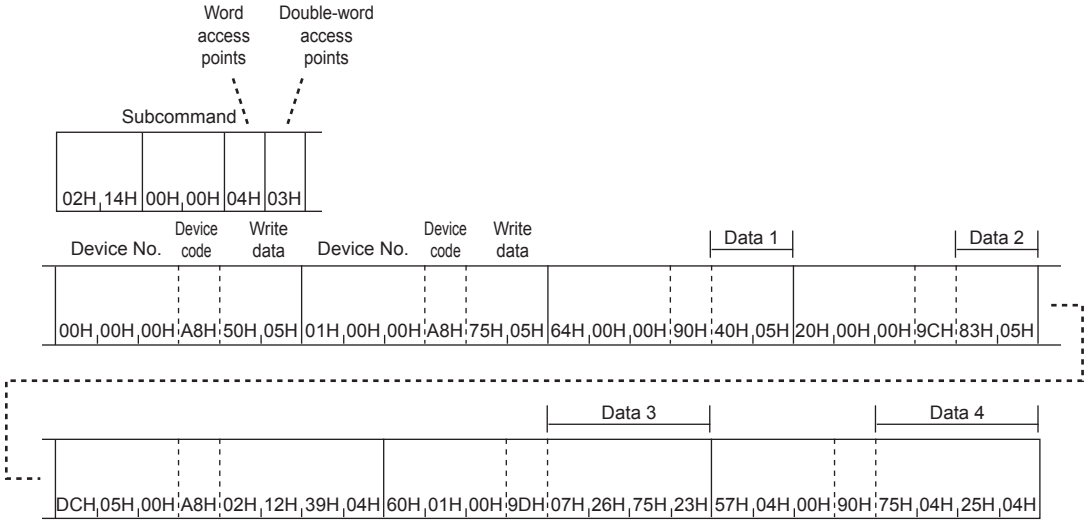
(Request data)

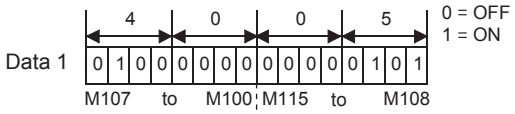


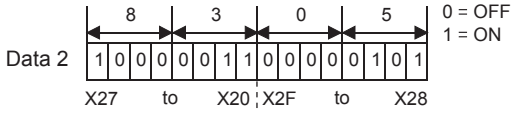


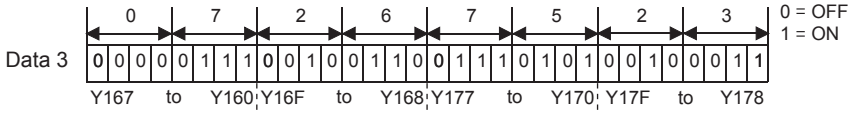
• When communicating data in binary code

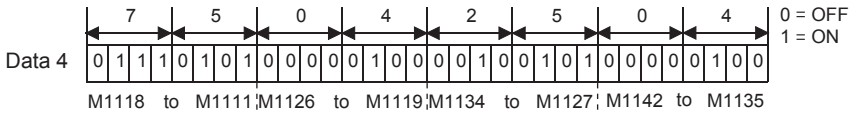
(Request data)











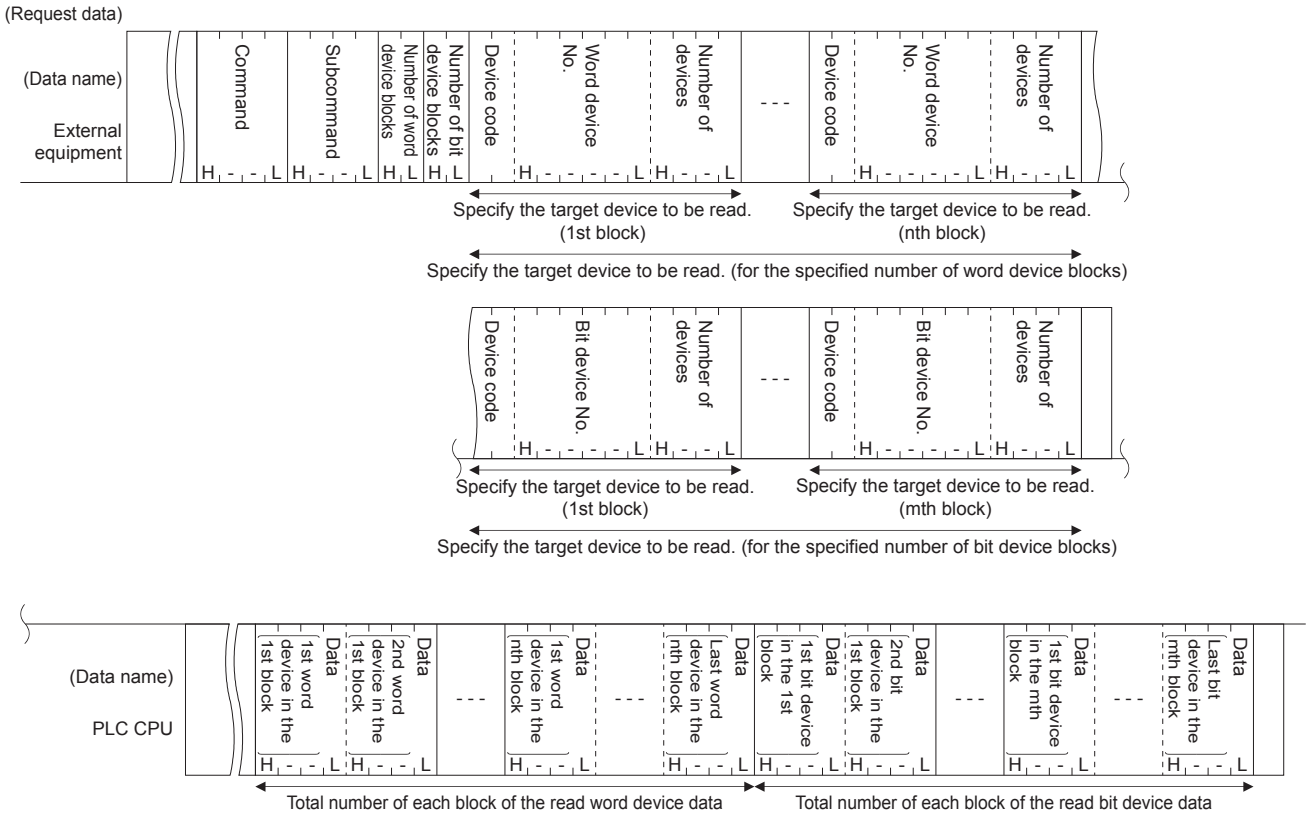
Device Read Block

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).

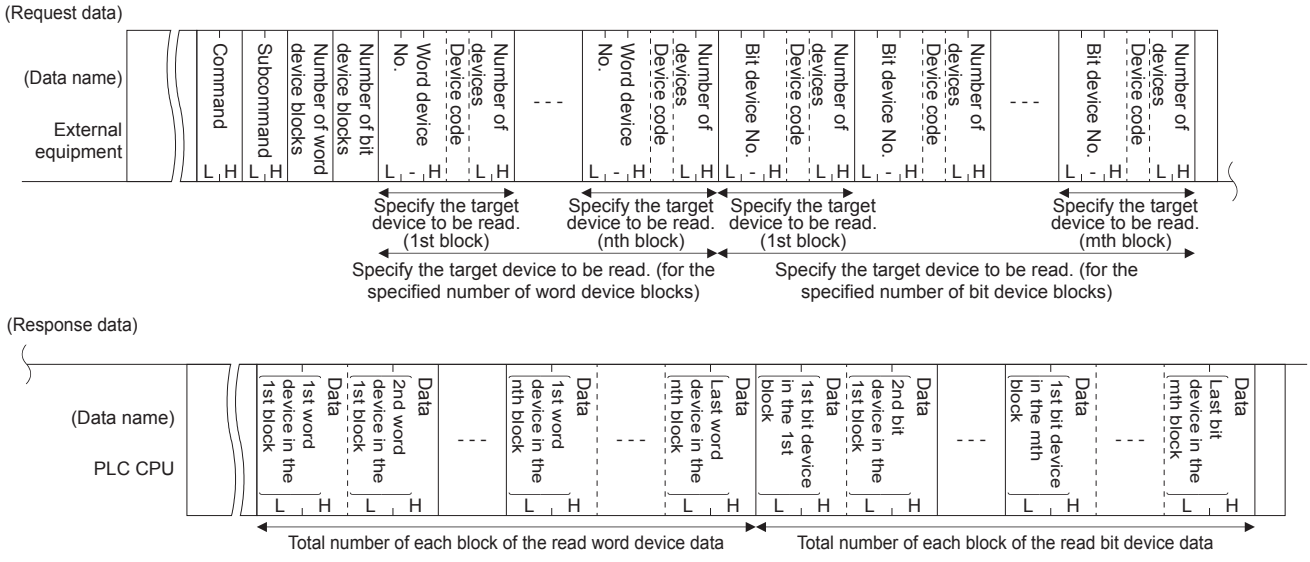
Data array in the character area during the device read block

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 44 Device range.

The double word device and the long index register (LZ) are not supported.

• 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

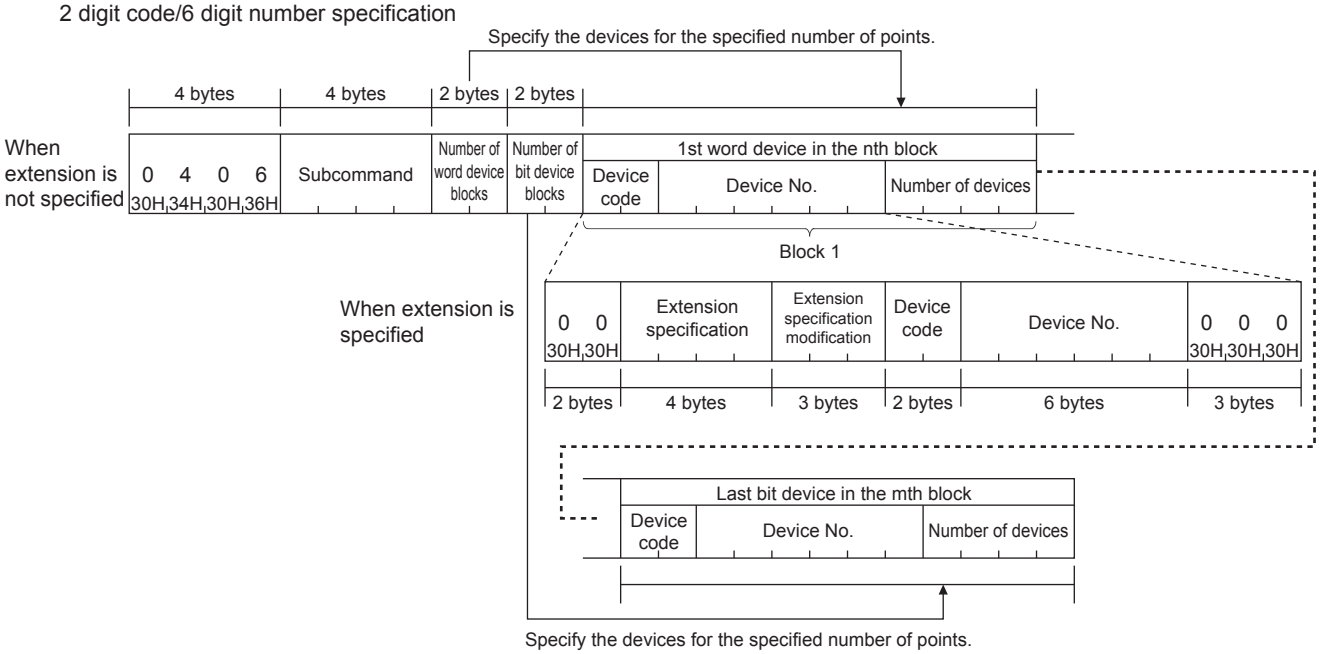
Point

The extension specification is allowed for the device memory being read using the device read block functions.

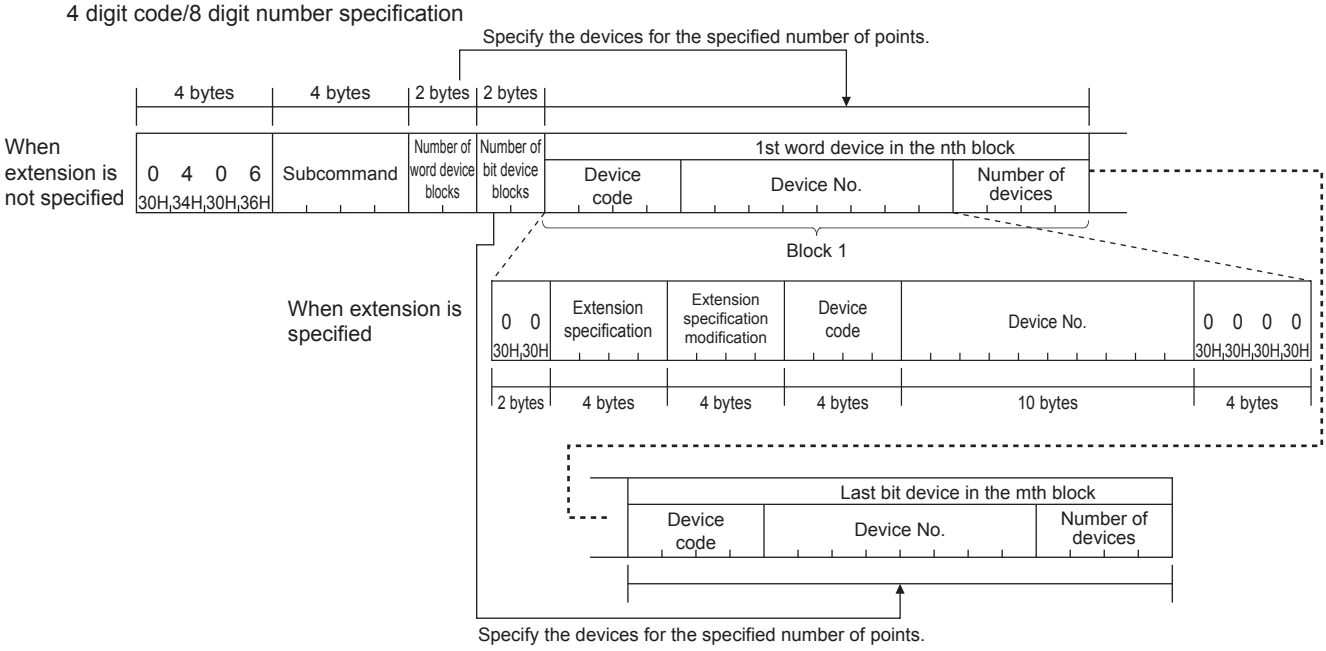
Request data

■When communicating data in ASCII code

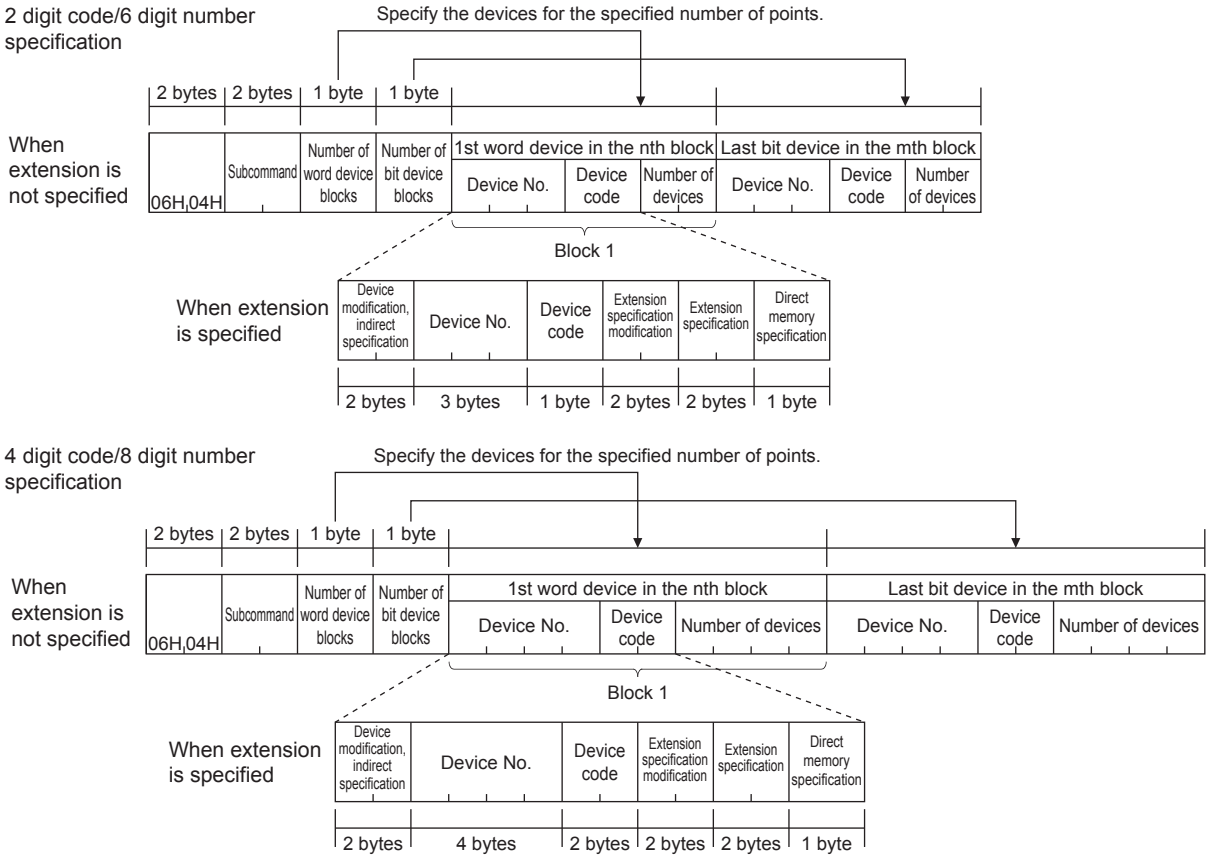
2 digit code/6 digit number specification



4 digit code/8 digit number specification



■When communicating data in binary code



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 blocks of word device + total points of each blocks 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 blocks of word device + total points of each blocks of bit device) ×2 ≤960	Number of word device blocks + number of bit device blocks ≤120 and Total points of each blocks of word device + total points of each blocks of bit device ≤960 When device memory extension specification is used (Number of word device blocks + number of bit device blocks) ×2 ≤120 and Total points of each blocks of word device + total points of each blocks of bit device ≤960
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

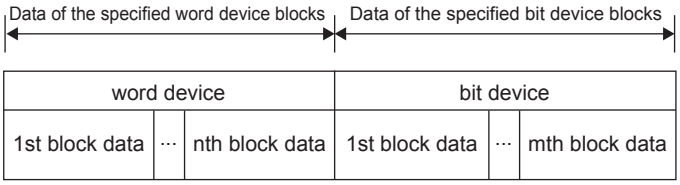
Item	Description
Word device	Specify the device points specified in "Number of word device blocks". When "Number of word device blocks" is set to 0, this specification is unnecessary.
Bit device	Specify the device points specified in "Number of bit device blocks". When "Number of bit device blocks" is 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



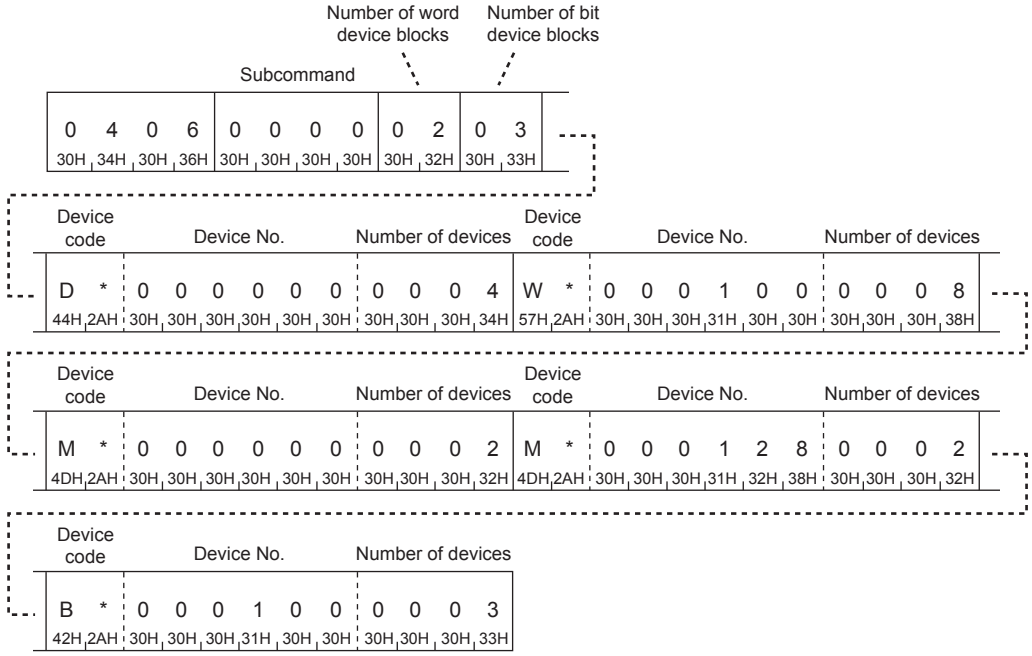
Communication example

Values are read from devices as follows.

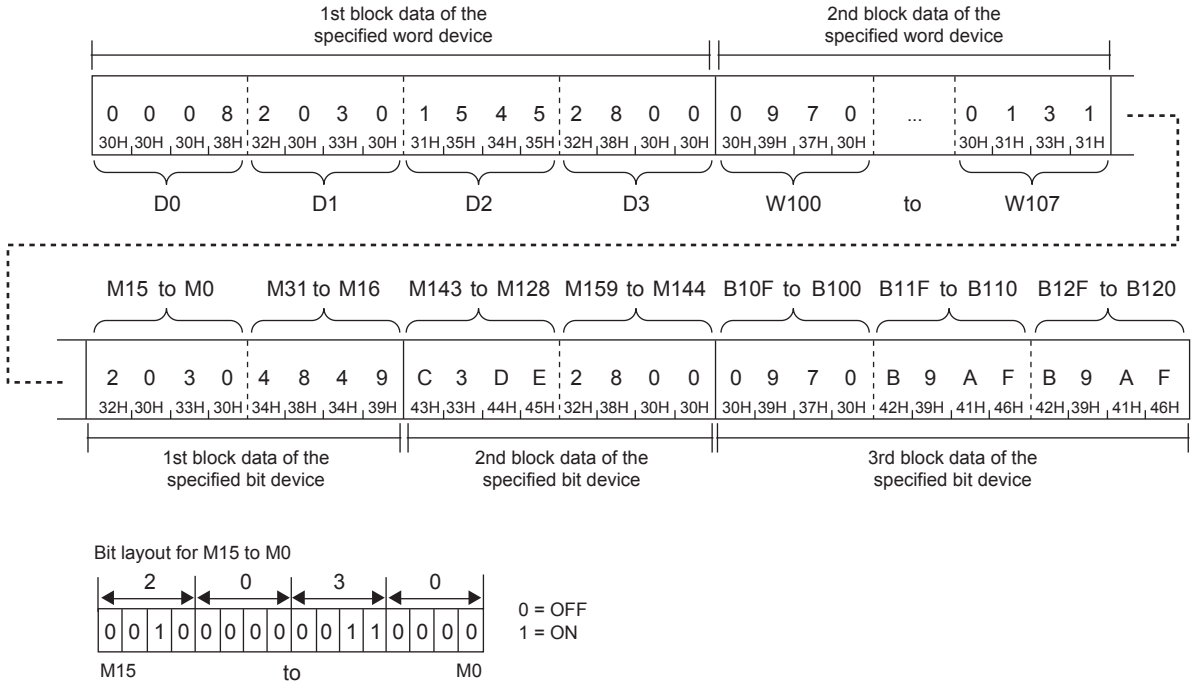
Item	Read contents
Word device	• Block 1: D0 to D3 (4 points) • Block 2: W100 to W107 (8 points)
Bit device	• Block 1: M0 to M31 (2 points) • Block 2: M128 to M159 (2 points) • Block 3: B100 to B12F (3 points)

■When communicating data in ASCII code

(Request data)

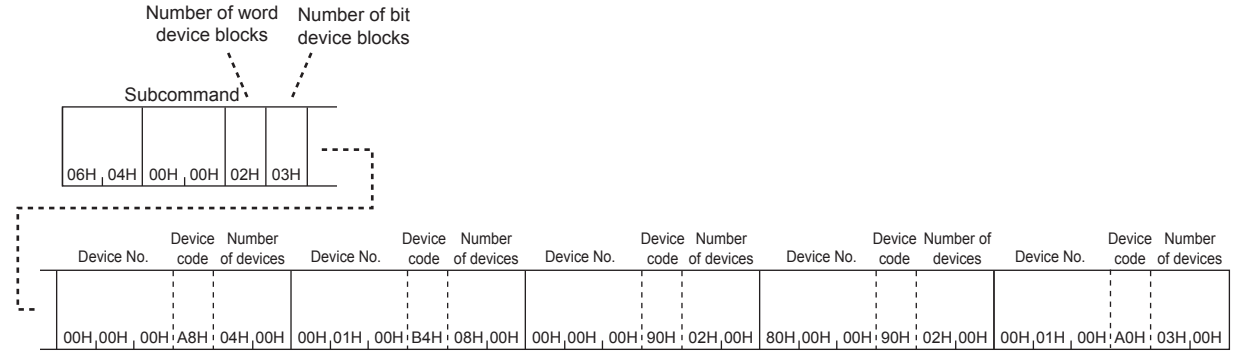


(Response data)

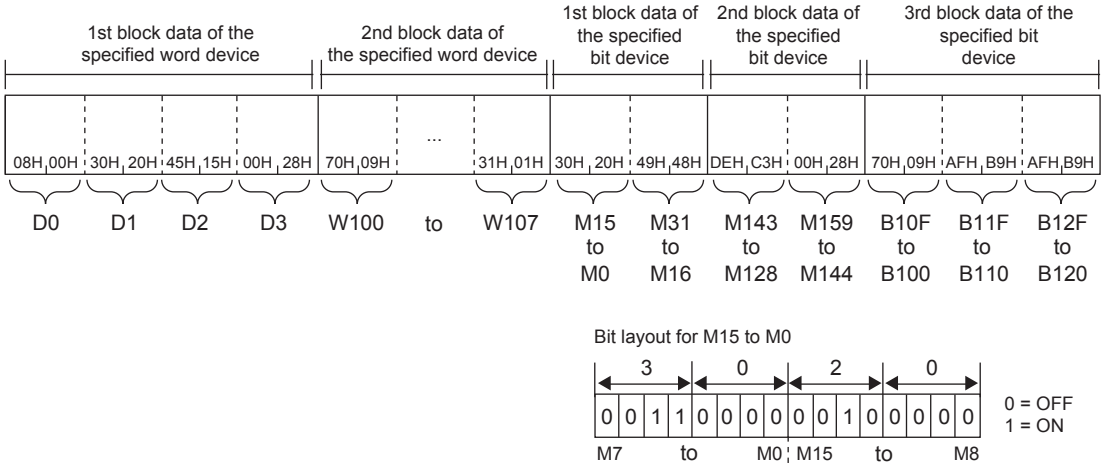


■When communicating data in binary code

(Request data)



(Response data)



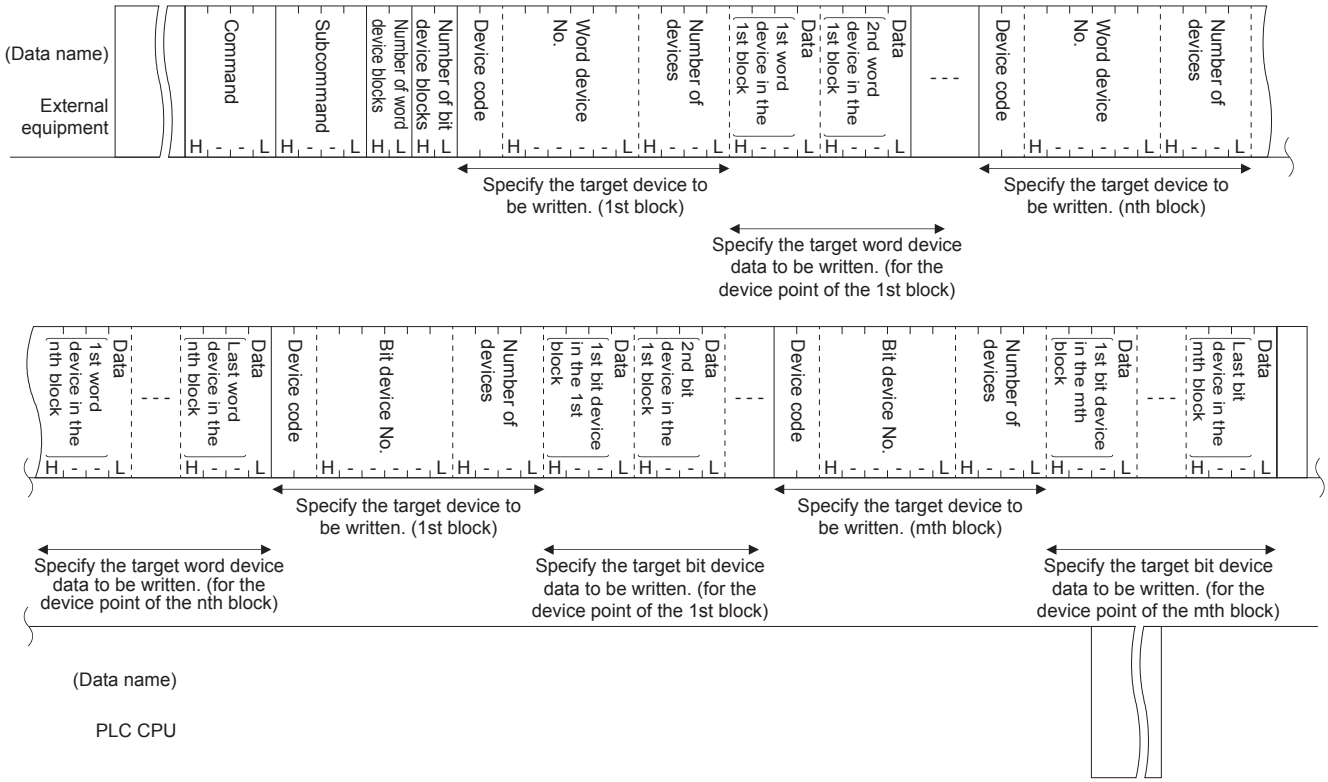
Device Write Block

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).

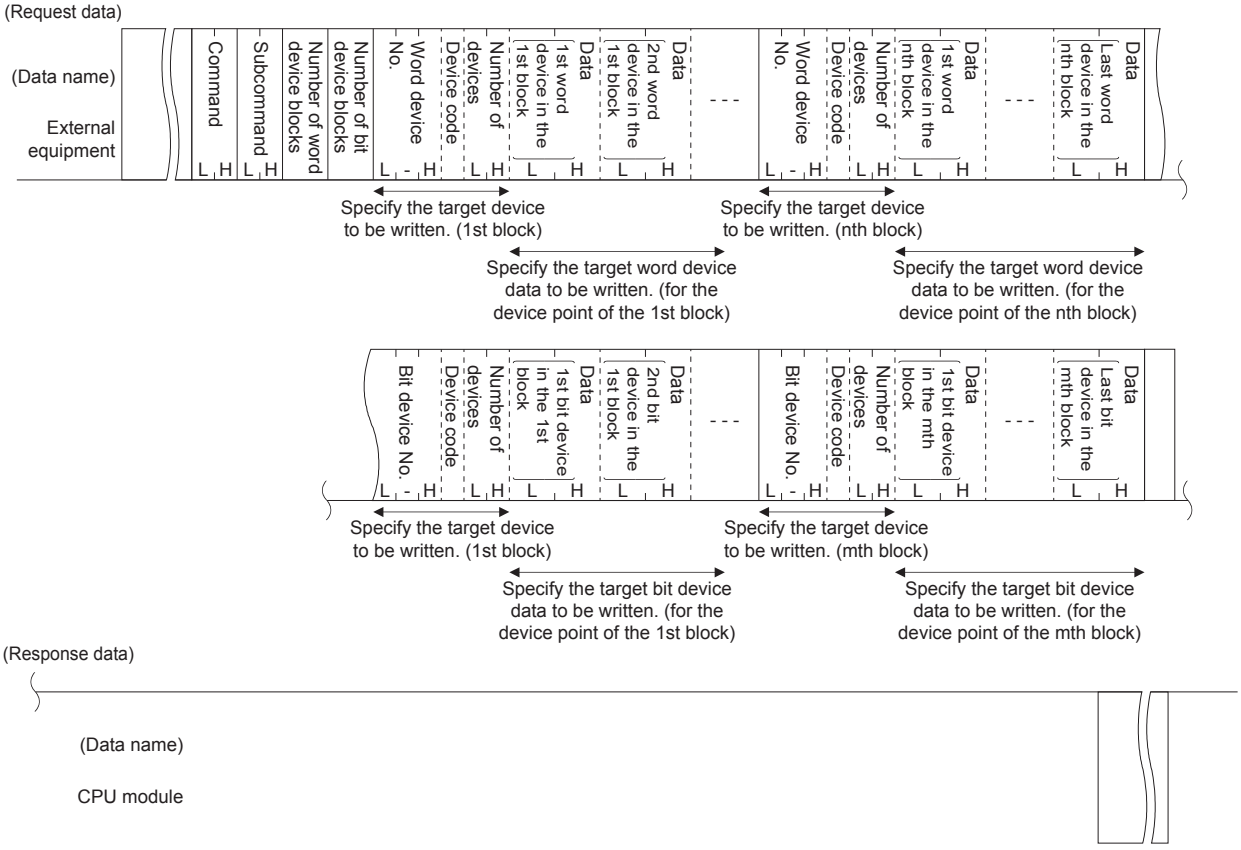
Data array in the character area during the device write block

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 44 Device range.

The double word device and the long index register (LZ) are not supported.

• 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

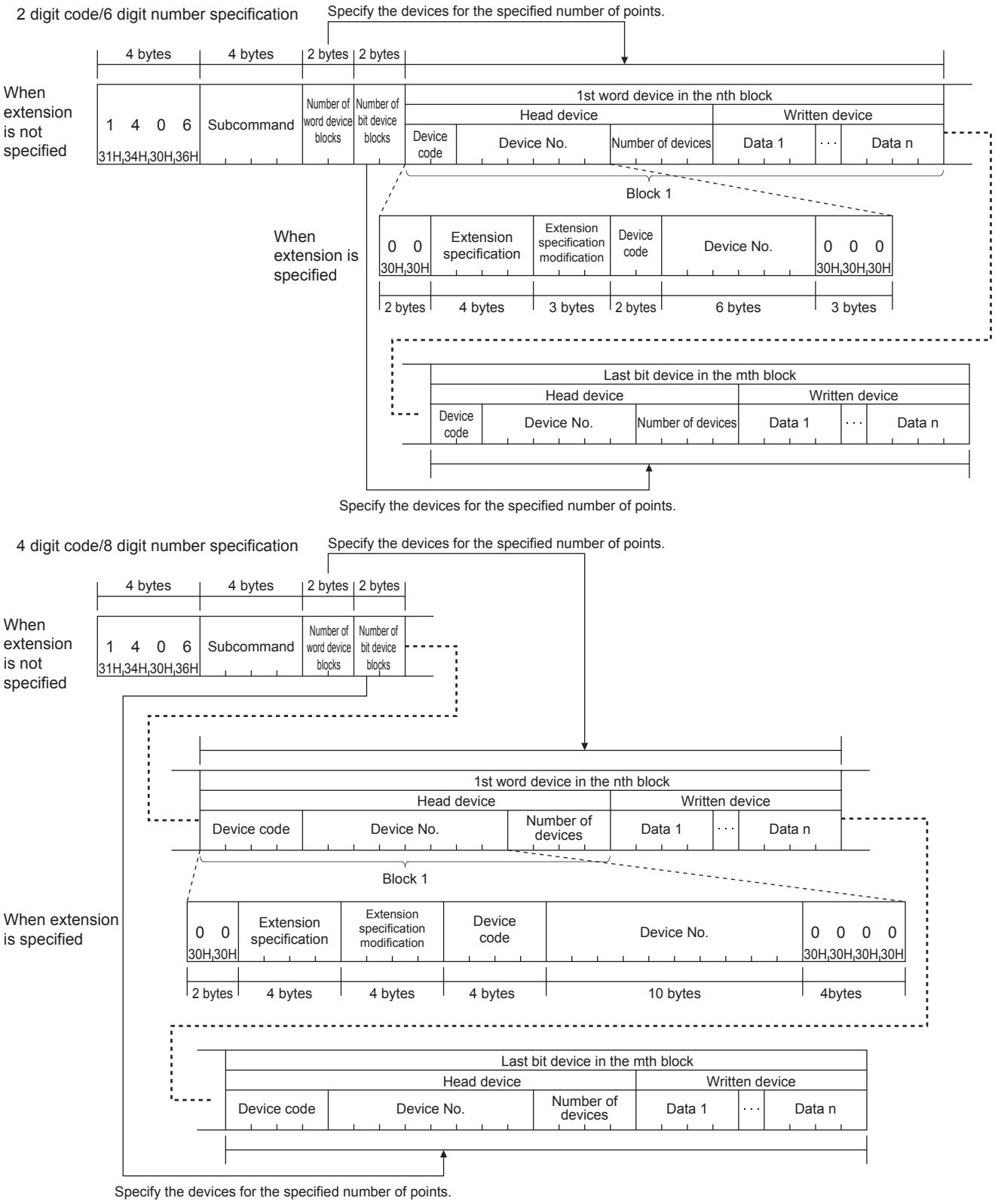
$770 \geq 4 \times$ (number of word device blocks + number of bit device blocks) + total number of points for all word device blocks + total number of points for all bit device blocks

Point

The extension specification is allowed for the device memory being written to using the device write block functions.

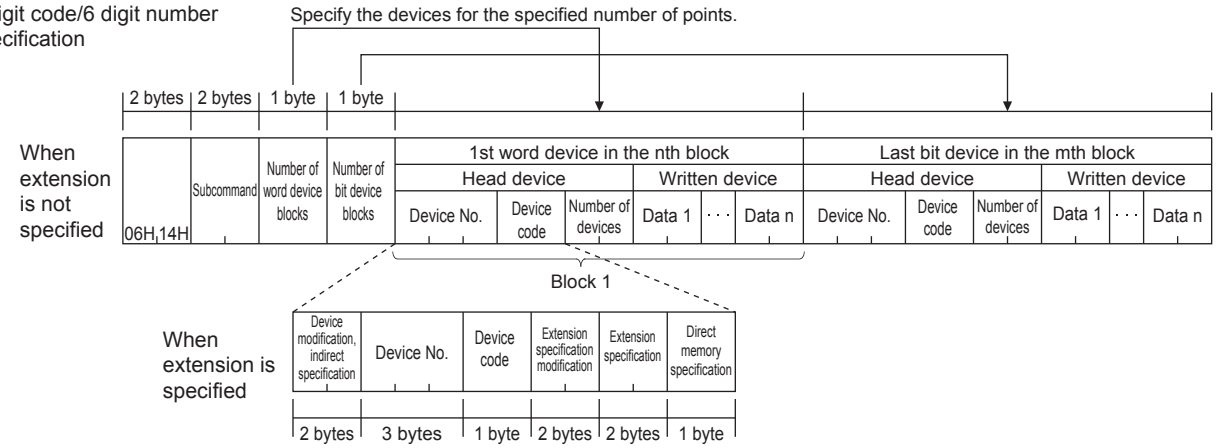
Request data

■When communicating data in ASCII code

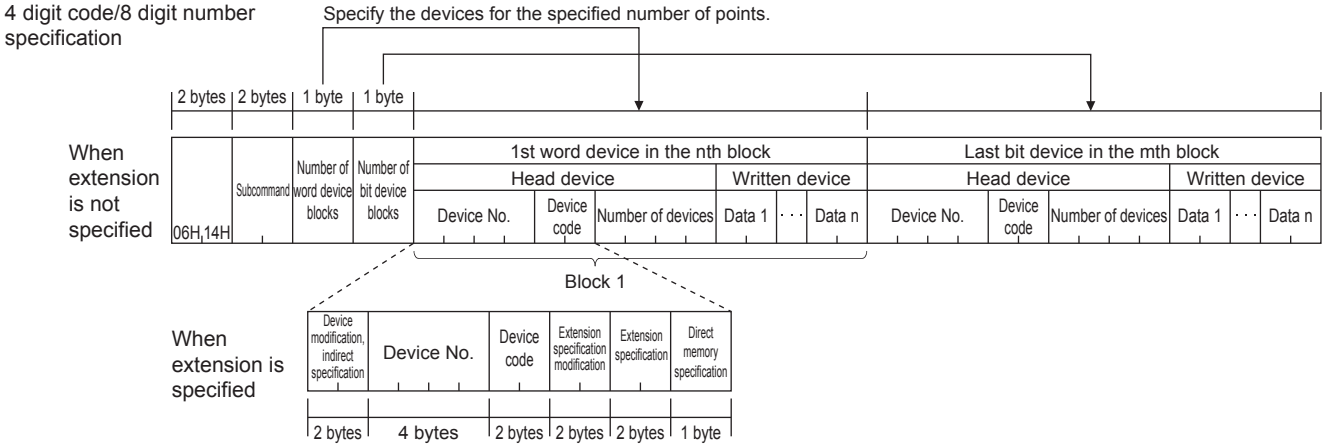


■When communicating data in binary code

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
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 blocks of word device + total points of each blocks of bit device) ×2 ≤770 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 blocks of word device + total points of each blocks of bit device) ×2 ≤770	Number of word device blocks + number of bit device blocks ≤120 and (Number of word device blocks + number of bit device blocks) ×4 + Total points of each blocks of word device + total points of each blocks of bit device ≤770 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 blocks of word device + total points of each blocks of bit device ≤770
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 ≤ 770

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.

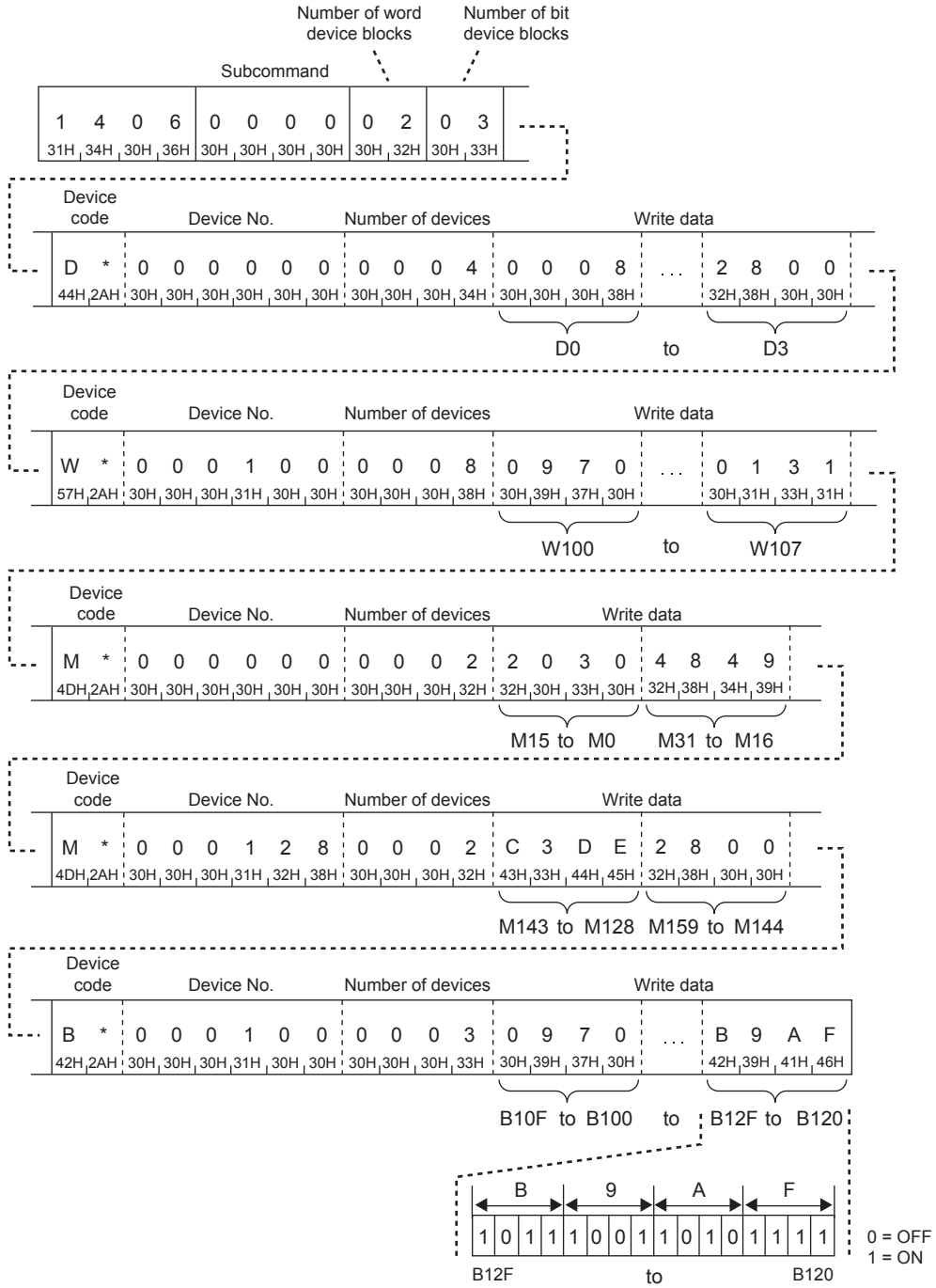
Communication example

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)

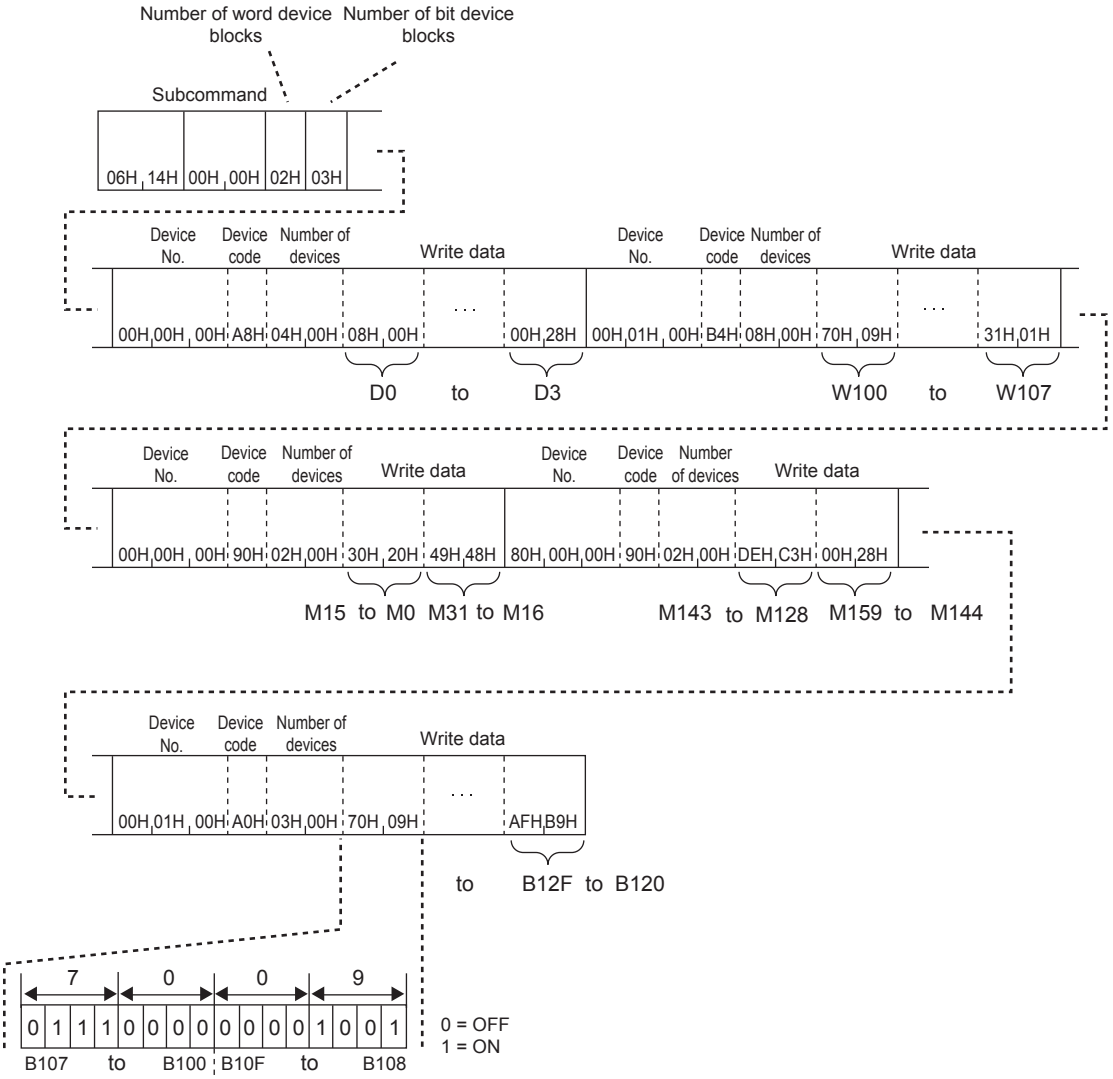
■When communicating data in ASCII code

(Request data)



■When communicating data in binary code

(Request data)



4.3 Remote Control

This section describes the command to set the SLMP compatible device or CPU module to the RUN status or STOP status by a message from the external device.

Before the remote operation

When the accessed device or module is turned from off to on or the system is reset after the 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 switch 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.

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

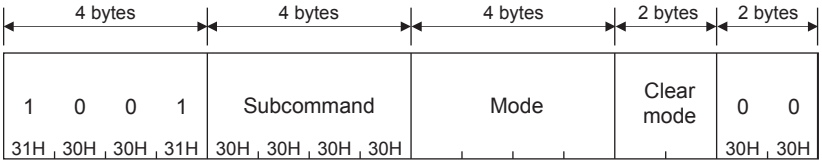
This command executes Remote RUN to the access destination module.

Point

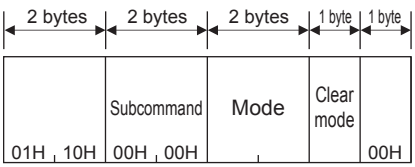
Remote RUN can be executed when the switch of the access destination module is in the RUN status. Even if the switch is in the STOP status, Remote RUN (command: 1001H) will be completed normally. However, the access destination does not change to the RUN status.

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	2	0	0				
31H	30H	30H	31H	30H	30H	30H	30H	30H	30H	30H	31H	30H	32H	30H	30H				

• When communicating data in binary code

(Request data)

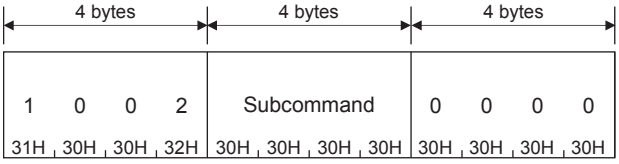
		Mode	Clear mode	
01H 10H	00H 00H	01H 00H	02H	00H

Remote STOP

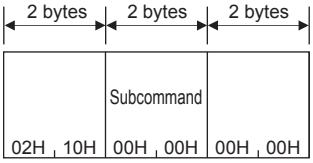
This command executes Remote STOP to the access destination module.

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

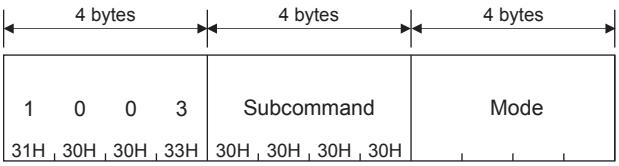
This command executes Remote PAUSE to the access destination module.

Point

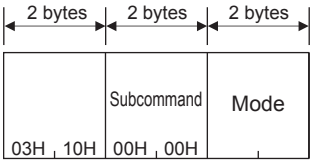
Remote PAUSE can be executed when the switch of the access destination module is in the RUN status. Even if the switch is in the STOP status, Remote PAUSE (command: 1003H) will be completed normally. However, the access destination does not change to the PAUSE status.

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 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

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	0	0	0	0	0	0	0	1	
31H	30H	30H	33H	30H	30H	30H	30H	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:

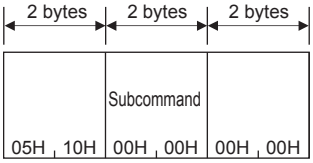
- The remote latch clear cannot be executed. Abnormal completion of the command will occur.
- Cancel the Remote STOP or Remote PAUSE before executing the command.

Request data

■When communicating data in ASCII code

4 bytes				4 bytes				4 bytes			
1	0	0	5	Subcommand				0	0	0	0
31H	30H	30H	35H	30H	30H	30H	30H	30H	30H	30H	30H

■When communicating data in binary code



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

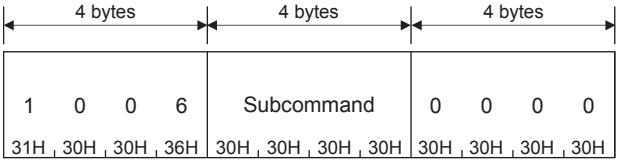
This command executes Remote RESET to the access destination module. Remote RESET is used to restore when an error occurred in the SLMP compatible device.

Point

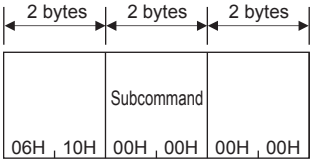
- Before executing Remote RESET, enable Remote RESET if there is a setting of Remote RESET enable or disable in the parameter of the access destination
- Before executing Remote RESET, set the status of the access destination module to STOP.
- In some cases, Remote RESET cannot be executed because of 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



■When communicating data in binary code



Response data

There is no response data for the Remote RESET command.

Communication example

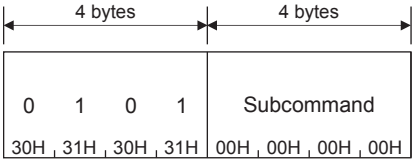
Send request messages from the external device by using the message format shown in the request data above.

Processor type read

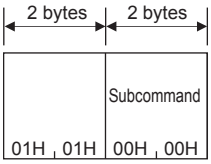
This command reads the processor module name code (processor type) of the access destination module.

Request data

■When communicating data in ASCII code

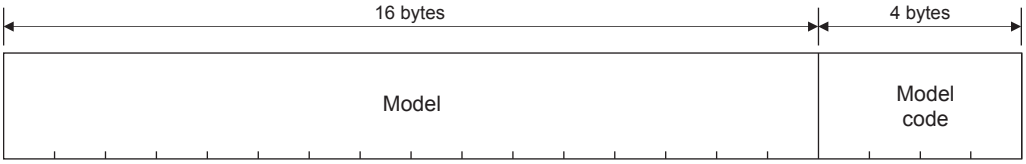


■When communicating data in binary code

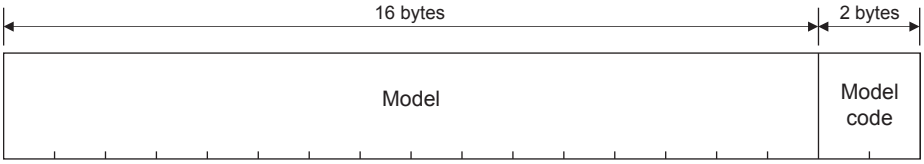


Response data

■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 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
FX5UC-32MT/D	4A91H
FX5UC-32MT/DSS	4A99H

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

4.4 Clear Error

This function turns off ERR LED of the FX5CPU 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 FX5CPU which is connected with the external equipment.
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 FX5CPU are turned off and the communication error information is initialized from the external equipment is described.

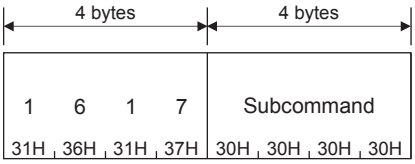
Command

Function	Command (Subcommand)	Processing content	CPU module status		
			STOP	RUN	
				Write allow setting	Write prohibit setting
Clear Error	1617(0000)	Turns off the display LEDs, initializes the error code, and others.	○	○	○

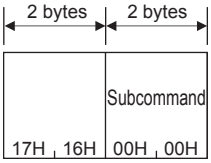
○: The function can be executed.

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.

4.5 Self-Test

This function tests whether the communication function between the external equipment 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 equipment 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 equipment (including a multi-drop connecting station). This function cannot be used for the FX5CPU of another station via the network system.

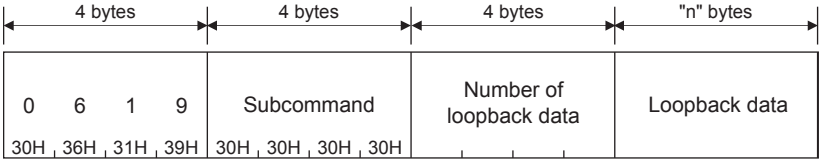
Command

Function	Command (Subcommand)	Processing content	CPU module status		
			STOP	RUN	
				Write allow setting	Write prohibit setting
Self-Test	0619(0000)	Checks whether a data communication is executed normally.	○	○	○

○: The function can be executed.

Request data

■When communicating data in ASCII code



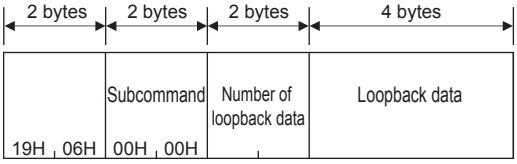
- Number of loopback data (number of bytes)

The number of the bytes is converted into a four-digit ASCII code (hexadecimal) and data is sent from the upper digit ("0").

- Loopback data (user 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 two-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 (user 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 shown in the request data (☞ Page 88).

■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

4.6 Remote Password Unlock or Lock

This function prevents illegal access from a user who is not allowed to operate the SLMP compatible device.
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.

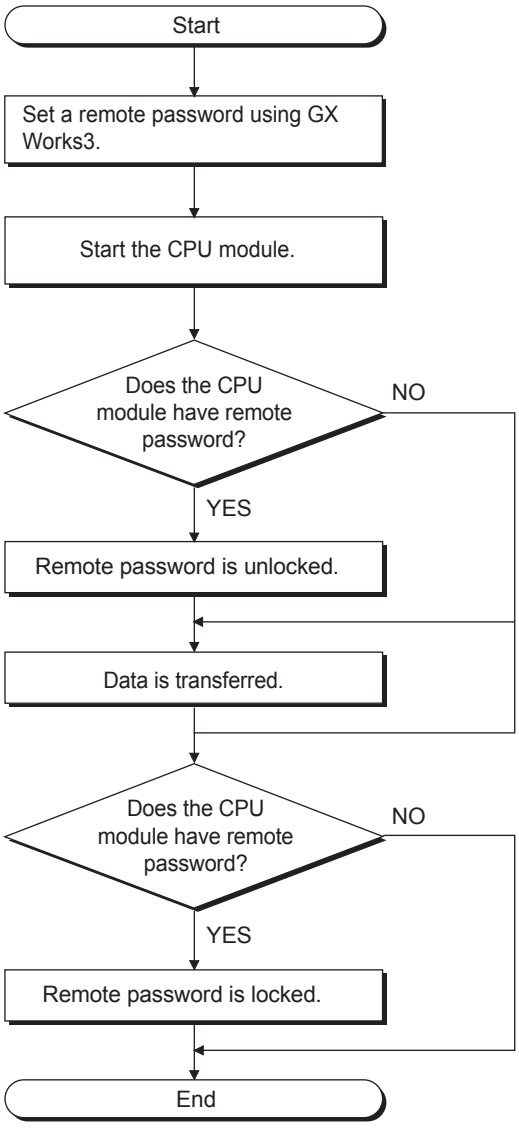
Target of the remote password checking function

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. (Communication to the device is disabled.)

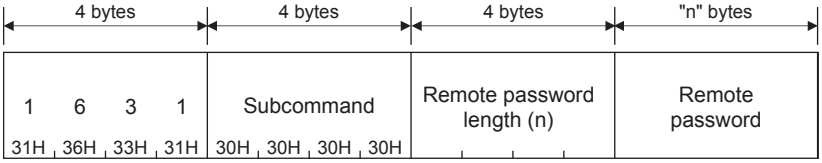
Command

Function		Command (Subcommand)	Processing content	CPU module status		
				STOP	RUN	
					Write allow setting	Write prohibit setting
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.)	○	○	○

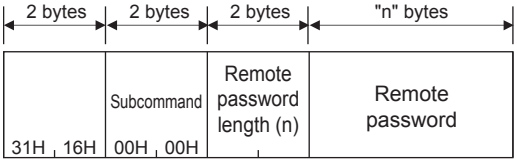
○: The function can be executed.

Request data

■When communicating data in ASCII code



■When communicating data in binary code



■Subcommand

Item		Subcommand	
		Binary code	
Default	Characters	—	—
	Character code	00H	00H

■Remote password length

Remote password length is not used.

■Remote password

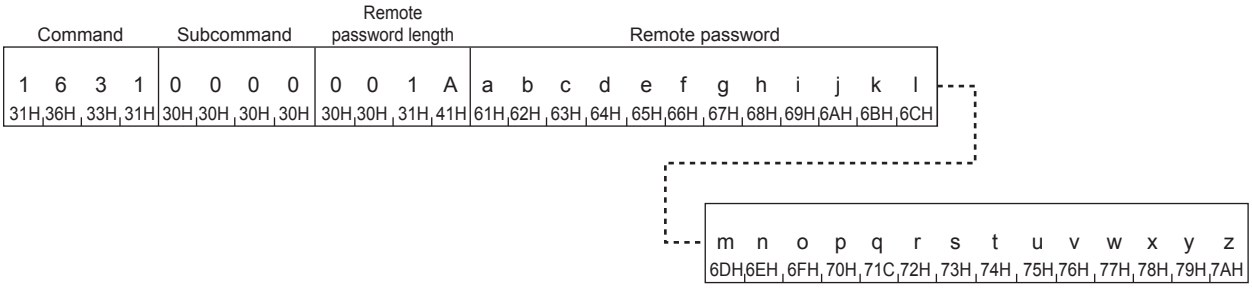
Remote password is not used.

Response data

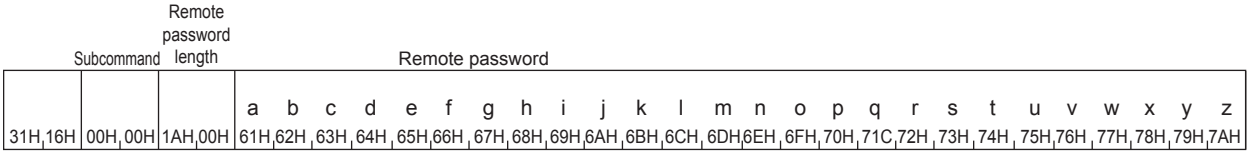
There is no response data for the lock command of the remote password.

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. (Enables communication to the device.)

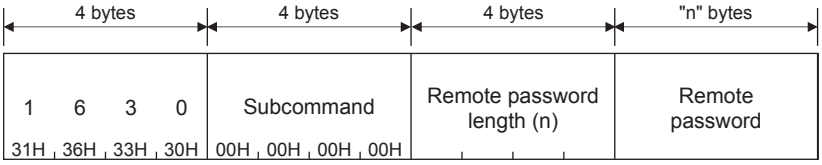
Command

Function		Command (Subcommand)	Processing content	CPU module status		
				STOP	RUN	
					Write allow setting	Write prohibit setting
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.)	○	○	○

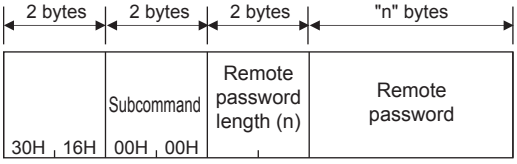
○: The function can be executed.

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.

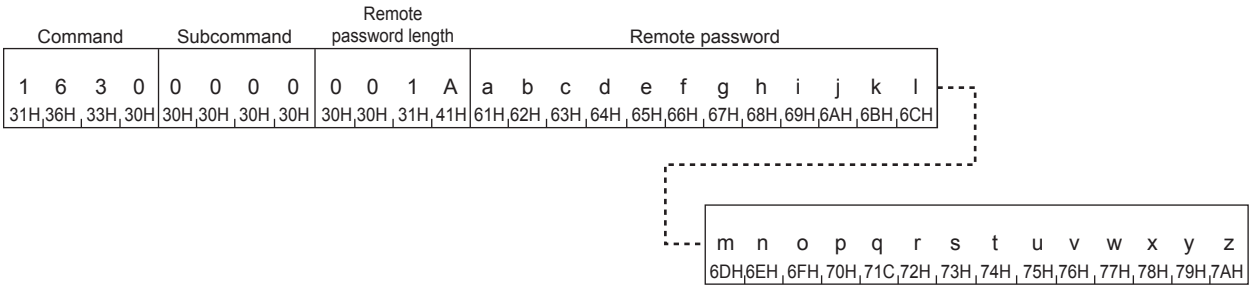
Specify the remote password using ASCII code also when communicating using binary code.

Response data

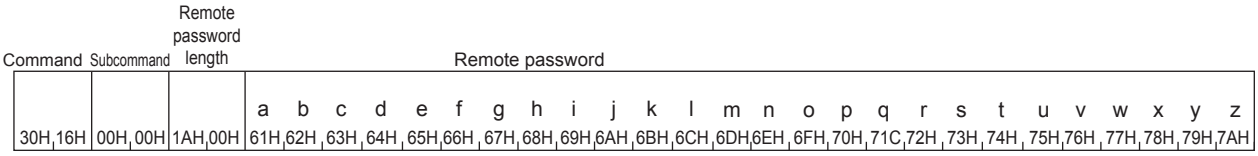
There is no response data for the unlock command of the remote password.

Communication example

■When performing the unlock process in communication using ASCII code



■When performing the unlock process in communication using binary code



APPENDIX

Appendix 1 Device Memory Extension Specification

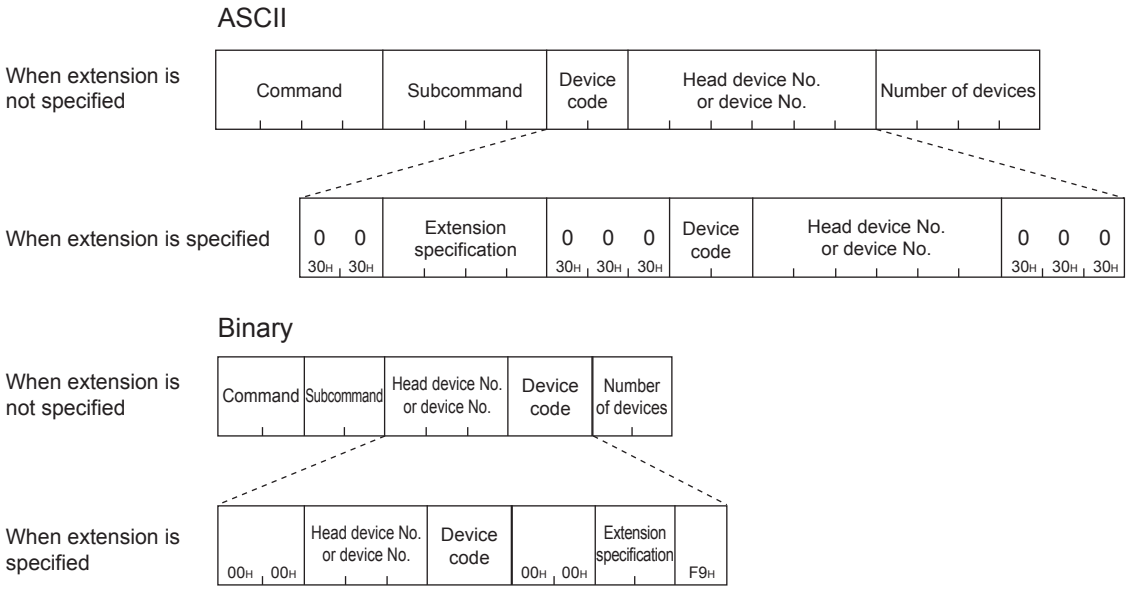
The following accesses are available by setting the subcommand of request data to 008☐.

- 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

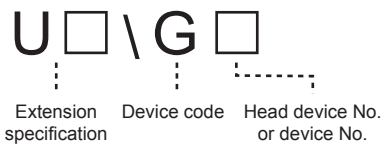
Access to module access device

Access to the buffer memory of SLMP compatible devices or intelligent function modules.

Request data



The following shows the module access device and request data.



Point

Devices described in Page 44 In the case of FX5CPU 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.

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 module number of intelligent function modules.

ASCII code	Binary code
Specify the module number in hexadecimal (ASCII code 3-digits). When described with 4-digits, specify the module number with the upper 3-digits. U □ □ □ 55H Example 001 U 0 0 1 55H 30H 30H 31H	Specify the module number in hexadecimal (2 bytes). When described with 4-digits, specify the module number with the upper 3-digits. □□H □□H Example 001 01H 00H

Point

• Access to buffer memories of modules other than intelligent function modules

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 For ASCII codes, the device code is specified with 2 characters. 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 97 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 module number is 003H.

• 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

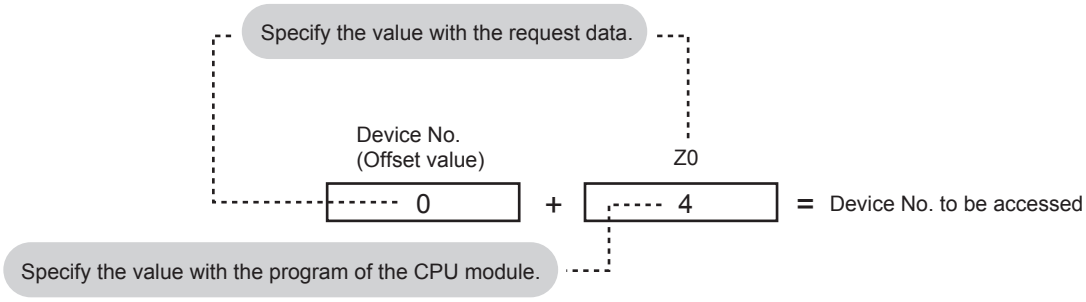
Access with indirect specification of the device No. by using index register or long index register

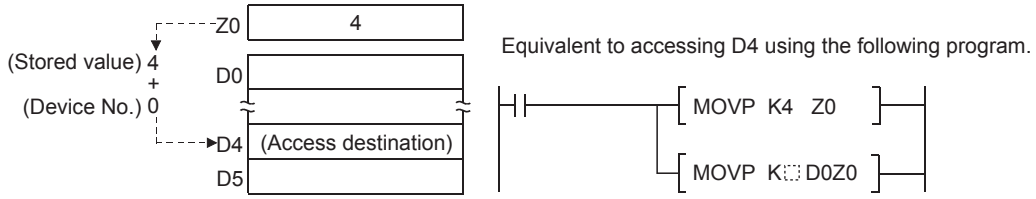
Indirect specification of the device No. can be performed by using the index register or long index register when accessing the device.

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.

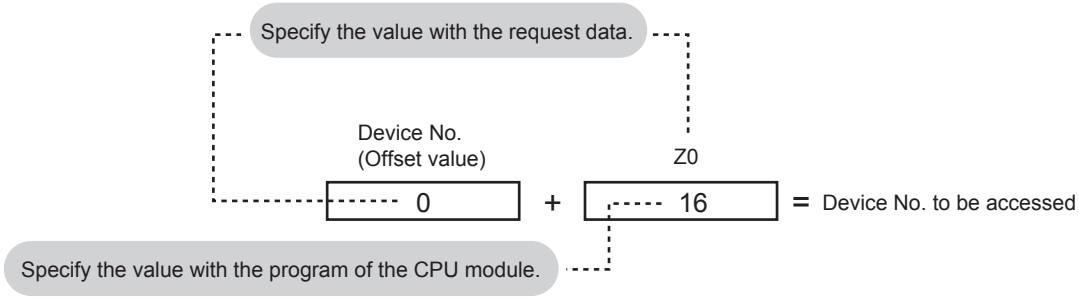
When accessing D4 with D0 and Z0 specifications

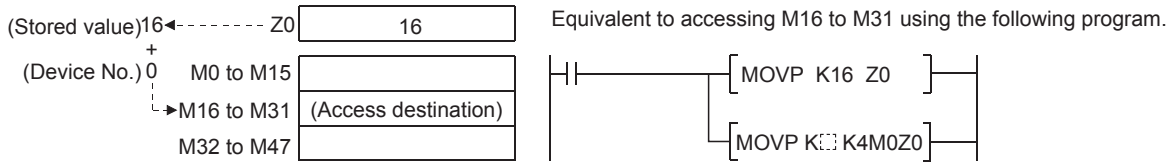




Ex.

When accessing M16 to M31 with M0 and Z0 specifications (Word units)

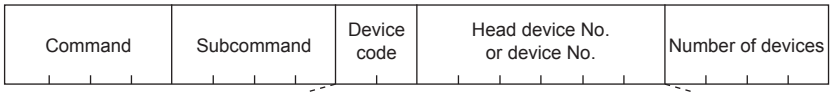




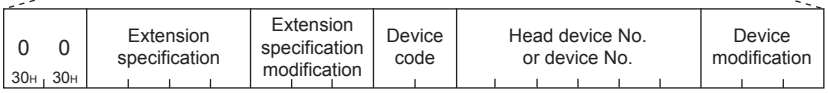
Request data

ASCII

When extension is not specified

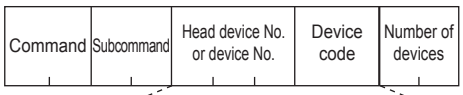


When extension is specified

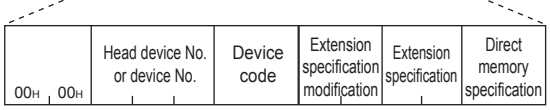


Binary

When extension is not specified

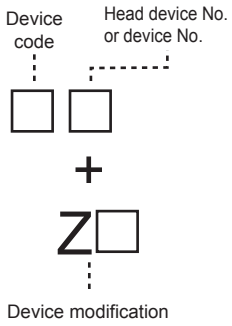


When extension is specified

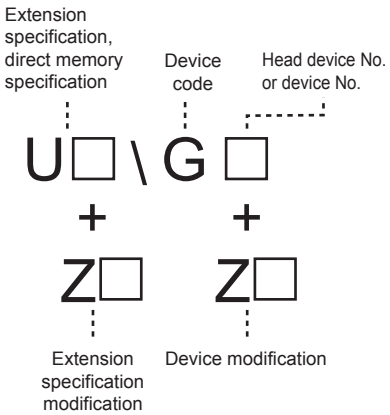


The following shows the approach for devices, index registers, long index registers and request data.

- 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 (2 bytes). 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 in decimal (2-digit ASCII code). (Specification range: 0 to 24) Z □ □ 5AH 20H	Specify the number of the index register (Z) in hexadecimal. (Specification range: 00H to 18H) □□H 40H
0081 0080	Specify the number of the index register in decimal (2-digit ASCII code). (Specification range: 0 to 24) Z □ □ 5AH	Specify the number of the index register (Z) in hexadecimal. (Specification range: 00H to 18H) □□H 40H

• 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: 0 to F) 

Point

The long index register (LZ) can not be used in the extension specification modification.

■Device code

Specify the code of the device to be accessed. (☞ Page 44 In the case of FX5CPU)

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 For ASCII codes, the device code is specified with 2 characters. 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.

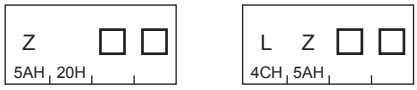
Specify the head device or device No. with the same format as the message when extension is not specified.

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 in decimal (2-digit ASCII code). (Specification range: 0 to 24)*1 Specify the number of the long index register (LZ) in decimal (2-digit ASCII code). (Specification range: 0 to 12) 	Specify the number of the index register (Z) in hexadecimal. (Specification range: 00H to 18H)*1 Specify the number of the long index register (LZ) in hexadecimal. (Specification range: 00H to 0CH) 
0081 0080	Specify the number of the index register in decimal (2-digit ASCII code). (Specification range: 0 to 24) 	Specify the number of the index register (Z) in hexadecimal. (Specification range: 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: 0 to F) 

■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	0	0	1	0	0	Z	0	4
30H	30H	38H	30H	30H	30H	30H	30H	30H	30H	30H	30H	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.

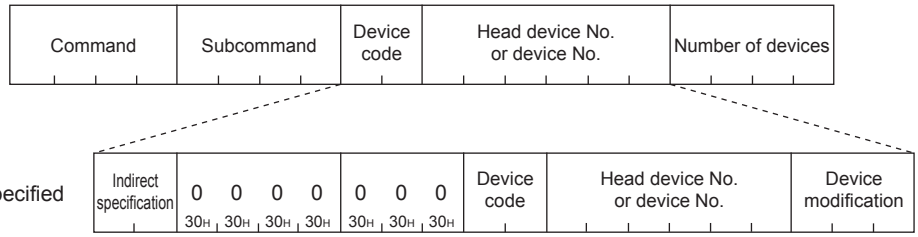
When storing the address of D100 in D0, and trying to access D100 from external devices by accessing "@D0"

The ADRSET instruction is used on the CPU module side and the address of D100 is stored in D0.

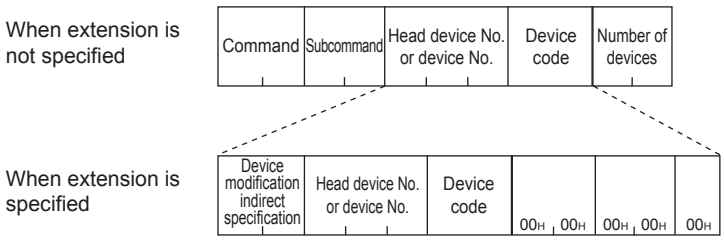
D100 can be indirectly accessed by specifying "@D0" with the request data.

Request data

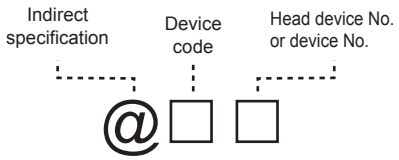
ASCII



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 can not be set simultaneously.

■Command

The following commands can be used for accessing.

Item		Command
Type	Operation	
Device	Read Random	0403
	Write Random	1402

■Subcommand

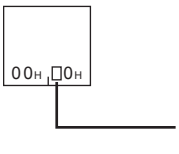
ASCII code	Binary code
0 0 8 0 30H 30H 38H 30H	80H 00H

■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

When communicating data in binary code



■Device code (Only word device codes can be specified at indirect specification)

Specify the code of the device to be accessed. (☞ Page 44 In the case of FX5CPU)

■Head device or device No.

Specify the head device or device No. with the same format as the message when extension is not specified.

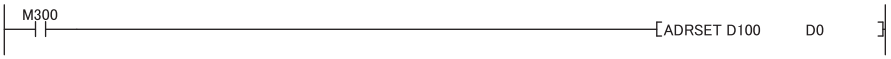
Response data

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	80H	00H	00H	00H	A8H	00H	00H	00H	00H	00H

Appendix 2 Command Comparison between MC Protocol and SLMP

The message format of 3E frame of the SLMP is the same as that of the QnA compatible 3E frame of MC protocol. The correspondence table of MC protocol and SLMP is shown below. When connecting an external device which uses MC protocol to a SLMP compatible device, check if replacement of command is required.

Applicable command list

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	

Appendix 3 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.

Item		Command	Subcommand	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)	
Device	Read	0401	0001	1/3584	0.03	1.51	1
			0000	1/960	0.03	0.21	1
	Write	1401	0001	1/3584	0.03	1.46	1
			0000	1/960	0.03	0.21	1
	Read Random	0403	0000	1/192	0.03	1.71	1
	Write Random	1402	0001	1/188	0.03	1.53	1
			0000	1/160*1	0.03	1.43	1
	Read Block	0406	0000	1/960	0.03	0.21	1
	Write Block	1406	0000	1/960	0.03	0.22	1
Remote Control	Read Type Name	0101	0000	(one station)	0.02	—	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. (MELSEC iQ-F FX5 User's Manual (Application))

• When extension of scan time affects the control

Access multiple times with less points.

INDEX

MEMO

REVISIONS

Revision date	Revision	Description
November 2014	A	First Edition
January 2015	B	■Added function Data code of ASCII

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.
   a) Failure occurring from inappropriate storage or handling, carelessness or negligence by the user. Failure caused by the user's hardware or software design.
   b) Failure caused by unapproved modifications, etc., to the product by the user.
   c) 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.
   d) Failure that could have been avoided if consumable parts (battery, backlight, fuse, etc.) designated in the instruction manual had been correctly serviced or replaced.
   e) Relay failure or output contact failure caused by usage beyond the specified life of contact (cycles).
   f) 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.
   g) Failure caused by reasons unpredictable by scientific technology standards at time of shipment from Mitsubishi.
   h) Any other failure found not to be the responsibility of Mitsubishi or that admitted not to be so by the user.

2. Onerous repair term after discontinuation of production

1) Mitsubishi shall accept onerous product repairs for seven (7) years after production of the product is discontinued.
   Discontinuation of production shall be notified with Mitsubishi Technical Bulletins, etc.
2) Product supply (including repair parts) is not available after production is discontinued.

3. Overseas service

Overseas, repairs shall be accepted by Mitsubishi's local overseas FA Center. Note that the repair conditions at each FA Center may differ.

4. Exclusion of loss in opportunity and secondary loss from warranty liability

Regardless of the gratis warranty term, Mitsubishi shall not be liable for compensation of damages caused by any cause found not to be the responsibility of Mitsubishi, loss in opportunity, lost profits incurred to the user or third person by failure of Mitsubishi products, special damages and secondary damages whether foreseeable or not, compensation for accidents, and compensation for damages to products other than Mitsubishi products, replacement by the user, maintenance of on-site equipment, start-up test run and other tasks.

5. Changes in product specifications

The specifications given in the catalogs, manuals or technical documents are subject to change without prior notice.

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.

TRADEMARKS

Microsoft® and Windows® are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
Ethernet is a trademark of Xerox Corporation.
The company name and the product name to be described in this manual are the registered trademarks or trademarks of each company.

Manual number: JY997D56001B
Model: FX5-U-SL-E
Model code: 09R541

MITSUBISHI ELECTRIC CORPORATION

HEAD OFFICE: TOKYO BUILDING, 2-7-3 MARUNOUCHI, CHIYODA-KU, TOKYO 100-8310, JAPAN
HIMEJI WORKS: 840, CHIYODA MACHI, HIMEJI, JAPAN